J. Miguel Nóbrega · Hrvoje Jasak *Editors*

OpenFOAM®

Selected Papers of the 11th Workshop

Springer

OpenFOAM®

J. Miguel Nóbrega · Hrvoje Jasak
Editors

OpenFOAM®

Selected Papers of the 11th Workshop

Springer

Editors
J. Miguel Nóbrega
Department of Polymer Engineering
University of Minho
Guimarães, Portugal

Hrvoje Jasak
University of Zagreb
Zagreb, Croatia

and

Wikki Ltd.
London, UK

ISBN 978-3-319-60845-7 ISBN 978-3-319-60846-4 (eBook)
https://doi.org/10.1007/978-3-319-60846-4

Library of Congress Control Number: 2018962914

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Group photograph taken at the 11th OpenFOAM Workshop

OpenFOAM® (Open source Field Operation And Manipulation) is a free, open-source computational toolbox that has a large user base across most areas of engineering and science, from both industrial and academic organizations. As a technology, OpenFOAM provides an extensive range of features to solve anything from complex fluid flows involving chemical reactions, turbulence, and heat transfer to solid dynamics and electromagnetics, among several others. Additionally, the OpenFOAM technology offers complete freedom to customize and extend its functionalities, which is one of the major contributions for its current growth rate.

The **OpenFOAM Workshop** provides a forum for researchers, industrial users, software developers, consultants, and academics working with OpenFOAM technology. The meeting is held at different locations around the world and is usually attended by OpenFOAM technology users and developers from all continents. The central part of the workshop is the 2-day conference, where presentations and posters on industrial applications and academic research are given.

The **11th OpenFOAM Workshop** was held in Guimarães, Portugal, and joined circa 300 individuals, from more than 30 countries and 180 institutions, from all regions of the globe. The meeting comprised more than 140 presentations covering the latest developments and applications related to OpenFOAM technology.

The book *OpenFOAM®—Selected Papers of the 11th Workshop*—aimed to join in a document self-contained information about the most prominent contributions presented at the 11th OpenFOAM Workshop and thus provides useful information both for experienced and novice users that intend to have a clear idea of the current trends and capabilities of OpenFOAM technology. This project would not be possible without the contribution of the authors of the 37 chapters accepted for publication and the support of more than 50 reviewers, which assured que quality of the book contents.

Guimarães, Portugal — J. Miguel Nóbrega
Zagreb, Croatia/London, UK — Hrvoje Jasak

Contents

Added Mass Partitioned Fluid–Structure Interaction Solver Based on a Robin Boundary Condition for Pressure

Željko Tuković, Martina Bukač, Philip Cardiff, Hrvoje Jasak and Alojz Ivanković

Abstract This paper describes a self-contained, partitioned fluid–structure interaction solver based on a finite volume discretisation. The incompressible fluid flow is described by the Navier–Stokes equations in the arbitrary Lagrangian–Eulerian form and the solid deformation is described by the St. Venant-Kirchhoff hyperelastic model in the total Lagrangian form. Both fluid and solid are discretised in space using the second-order accurate cell-centred finite volume method, and temporal discretisation is performed using the first-order accurate implicit Euler scheme. Coupling between fluid and solid is performed using a Robin-Neumann partitioned procedure based on a new Robin boundary condition for pressure. The solver has been tested on the wave propagation in an elastic tube test case characterised by a low solid-to-fluid density ratio. The first-order temporal accuracy is shown and the stability of the method is demonstrated for both the strongly coupled and loosely coupled versions of the solution procedure. It is also shown that the proposed methodology can efficiently handle FSI cases in which the fluid domain is entirely enclosed by Dirichlet boundary conditions, even for the case of geometrically nonlinear elastic deformation.

Ž. Tuković (✉) · H. Jasak
Faculty of Mechanical Engineering and Naval Architecture,
University of Zagreb, Zagreb, Croatia
e-mail: Zeljko.Tukovic@fsb.hr

H. Jasak
e-mail: Hrvoje.Jasak@fsb.hr

M. Bukač
Department of Applied and Computational Mathematics and Statistics,
University of Notre Dame, South Bend, IN, USA
e-mail: mbukac@nd.edu

P. Cardiff · A. Ivanković
School of Mechanical and Materials Engineering, University College Dubin,
Bekaert University Technology Centre, Dublin, Ireland
e-mail: Philip.Cardiff@ucd.ie

A. Ivanković
e-mail: Alojz.Ivankovic@ucd.ie

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_1

1 Introduction

Fluid–structure interaction (FSI) problems can be solved in a monolithic or partitioned way. Loosely or strongly coupled partitioned solution procedures solve the fluid problem separately from the structure problem; consequently, this approach is popular due to its modularity and simplicity of implementation. In classical partitioned approaches, known as Dirichlet-Neumann (DN) schemes, the fluid problem is solved with a Dirichlet boundary condition (structure velocity) at the fluid–structure interface, while the structure problem is solved with a Neumann boundary condition (fluid stress) at the interface. Loosely coupled partitioned DN schemes are stable only if the structure's density is much larger than the fluid's density. This requirement is easily achieved in some applications like aerodynamics, but not, for example, in hemodynamics, in which the density of blood is on the same order of magnitude as the density of arterial walls. In these cases, the energy of the discrete problem in the DN partitioned algorithm does not accurately approximate the energy of the continuous problem, introducing numerical instabilities known as the added mass effect. A possible solution to this problem is to sub-iterate between the fluid and structure sub-problems at each time step until the energy at the fluid–structure interface is balanced. Schemes that require sub-iterations, also known as strongly coupled partitioned schemes, may suffer from slow convergence issues, however, this can be mitigated using Aitken's dynamic relaxation method [1] or reduced order models [2].

Loosely coupled partitioned schemes that can efficiently cope with the added mass effect are rare. One such scheme is the kinematically coupled β-scheme [3, 4]. The scheme is based on Lie operator splitting, in which the fluid and the structure sub-problems are fully decoupled and communicate only via the initial conditions. The fluid and structure equations are split in such a way that the fluid problem is solved with a Robin-type boundary condition (BC) including the structural inertia; this is the main ingredient of the scheme to ensure unconditional stability. In the original implementation of this scheme, the second-order accurate finite element method (FEM) is used for discretisation of both fluid and solid models in space and the overall temporal accuracy of the scheme is first order.

The main goal of this study is to implement the kinematically coupled β-scheme in a new computational framework, in which both fluid and structure sub-problems are discretised using a second-order accurate cell-centred finite volume method (FVM); the flow sub-problem is solved using a pressure-based solver and a SIMPLE-like solution procedure. In this work, the first step in the fulfilment of the above-mentioned goal will be presented, namely the implementation of the Robin BC for the fluid sub-problem. In the context of the pressure-based fluid flow solver, the proposed Robin BC is applied on the pressure field during the solution of pressure equation. The derivative of the pressure at the interface in the normal direction is defined by the simplified momentum equation, and at the same time, the value of the pressure is limited by the structural inertia. The Robin BC, derived in such a way, is very similar to the Robin BC proposed by Banks, Henshaw and Schwendeman [5]. Based on the Robin BC for pressure, a Robin-Neumann (RN) partitioned FSI scheme is proposed

in which the fluid sub-problem is solved with the Robin BC at the interface, while the structure sub-problem is still solved with the Neumann BC. Although the proposed FSI scheme does not contain all of the ingredients of the kinematically coupled β-scheme, it already shows good performance in terms of stability and accuracy.

The article is organised as follows: Sect. 2 outlines the mathematical model for the fluid and structure and shows the derivation of the Robin BC for pressure at the fluid–structure interface. The discretisation of the mathematical model and FSI solution procedure are given in Sects. 3 and 4 presents the application of the method to two benchmark test cases, in which the stability and order of accuracy of the method are demonstrated.

2 Mathematical Model

In the current article, the interaction between an incompressible Newtonian fluid and a hyperelastic solid is considered, in which an FV discretisation and partitioned approach are employed. Mathematical models governing mechanical behaviour of the fluid and solid are solved separately from each other, and coupling is achieved through enforcement of proper boundary conditions at the interface.

2.1 Fluid Governing Equations

Fluid flow is considered on a spatial domain whose shape is changing in time, due to deformation of the fluid–solid interface. The isothermal flow of an incompressible Newtonian fluid, inside an arbitrary volume V bounded by a closed moving surface S, is governed by the mass and linear momentum conservation laws

$$\oint_S \mathbf{n} \cdot \mathbf{v} \, \mathrm{d}S = 0, \tag{1}$$

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t} \int_V \mathbf{v} \, \mathrm{d}V + \oint_S \mathbf{n} \cdot (\mathbf{v} - \mathbf{v}_s) \mathbf{v} \, \mathrm{d}S \\ &= \oint_S \mathbf{n} \cdot (\nu_F \nabla \mathbf{v}) \, \mathrm{d}S - \frac{1}{\rho_F} \int_V \nabla p \, \mathrm{d}V, \end{aligned} \tag{2}$$

where $\mathbf{n}$ is the outward pointing unit normal on S, $\mathbf{v}$ is the fluid velocity, $\mathbf{v}_s$ is the velocity of surface S, ν_F is the fluid kinematic viscosity, p is the fluid pressure and ρ_F is the fluid density. The relationship between the rate of change of the volume V and the velocity $\mathbf{v}_s$ is defined by the *geometric (space) conservation law* (GCL, see [6, 7])

$$\frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{V} \mathrm{d}V - \oint\limits_{S} \mathbf{n}\cdot\mathbf{v}_s \,\mathrm{d}S = 0. \tag{3}$$

The mathematical model presented in the form above is usually referred to as the arbitrary Lagrangian–Eulerian (ALE) formulation.

2.2 Solid Governing Equations

The deformation of the solid, assumed to be elastic and compressible, can be described by the linear momentum conservation law in the total Lagrangian form

$$\int\limits_{V_0} \rho_{S,0}\frac{\partial}{\partial t}\left(\frac{\partial \mathbf{u}}{\partial t}\right)\mathrm{d}V = \int\limits_{S_0} \mathbf{n}\cdot\left(\Sigma\cdot\mathbf{F}^{\mathrm{T}}\right)\mathrm{d}S + \int\limits_{V_0}\rho_{S,0}\mathbf{b}\,\mathrm{d}V, \tag{4}$$

where subscript 0 indicates quantities related to the initial (undeformed) configuration, ρ_S is the solid density, $\mathbf{u}$ is the displacement vector, $\mathbf{F} = \mathbf{I} + (\nabla\mathbf{u})^{\mathrm{T}}$ is the deformation gradient tensor, $\mathbf{I}$ is the second-order identity tensor and Σ is the second Piola-Kirchhoff stress tensor, which is related to the Cauchy stress tensor σ by the following expression:

$$\sigma = \frac{\mathbf{1}}{\det\mathbf{F}}\mathbf{F}\cdot\Sigma\cdot\mathbf{F}^{\mathrm{T}}. \tag{5}$$

The St. Venant-Kirchhoff constitutive material model is assumed, which relates the second Piola-Kirchhoff stress tensor to the Green-Lagrange strain tensor as follows:

$$\Sigma = 2\mu_S\mathbf{E} + \lambda_S\mathrm{tr}(\mathbf{E})\,\mathbf{I}, \tag{6}$$

where μ_S and λ_S are the Lamé's coefficients. The Green-Lagrange strain tensor is defined as

$$\mathbf{E} = \frac{1}{2}\left[\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}} + \nabla\mathbf{u}\cdot(\nabla\mathbf{u})^{\mathrm{T}}\right]. \tag{7}$$

Substituting the constitutive relation, Eq. (6), into the governing equation, Eq. (4), one obtains the linear momentum conservation equation for a St. Venant-Kirchhoff hyperelastic solid in the total Lagrangian form, where the displacement vector $\mathbf{u}$ is the primitive variable

$$\rho_{S0}\int\limits_{V_0}\frac{\partial}{\partial t}\left(\frac{\partial\mathbf{u}}{\partial t}\right)\mathrm{d}V - \oint\limits_{S_0}\mathbf{n}\cdot(2\mu_S+\lambda_S)\nabla\mathbf{u}\,\mathrm{d}S = \oint\limits_{S_0}\mathbf{n}\cdot\mathbf{q}\,\mathrm{d}S + \rho_{S0}\int\limits_{V_0}\mathbf{b}\,\mathrm{d}V, \tag{8}$$

where

$$\begin{aligned}\mathbf{q} &= \mu_S(\nabla\mathbf{u})^{\mathrm{T}} + \lambda_S \mathrm{tr}(\nabla\mathbf{u})\,\mathbf{I} - (\mu_S + \lambda_S)\nabla\mathbf{u} \\ &+ \mu_S\,\nabla\mathbf{u}\cdot(\nabla\mathbf{u})^{\mathrm{T}} + \frac{1}{2}\lambda_S \mathrm{tr}[\nabla\mathbf{u}\cdot(\nabla\mathbf{u})^{\mathrm{T}}]\mathbf{I} + \Sigma\cdot\nabla\mathbf{u}.\end{aligned} \tag{9}$$

Tensor $\mathbf{q}$ consists of nonlinear and displacement component coupling terms that are treated explicitly after the discretisation, allowing the resulting linear system to be solved using a segregated algorithm. The diffusivity $(2\mu_S + \lambda_S)$ in the Laplacian on the left-hand side of Eq. (8) is used to maximise the implicit part of the discretised equation [8].

2.3 Conditions at the Fluid–Solid Interface

The fluid and solid models are coupled by kinematic and dynamic conditions, which must be satisfied at the fluid–solid interface. The kinematic condition states that the velocity and displacement must be continuous across the interface

$$\mathbf{v}_{F,i} = \mathbf{v}_{S,i}, \tag{10}$$

$$\mathbf{u}_{F,i} = \mathbf{u}_{S,i}, \tag{11}$$

where subscripts F and S represent quantities corresponding to the fluid and solid regions of the model, respectively, and subscript i represents quantities at the fluid–solid interface. When a DN coupling scheme is used, displacement and velocity are calculated at the solid side of the interface and applied as a Dirichlet boundary condition at the fluid side of the interface.

The dynamic condition follows from the linear momentum conservation law, which, as was discussed by Batchelor [9], reduces to the force equilibrium equation at the fluid–solid (material) interface

$$\mathbf{n}_i \cdot \sigma_{\mathbf{F,i}} = \mathbf{n_i} \cdot \sigma_{\mathbf{S,i}}, \tag{12}$$

where $\mathbf{n}_i$ is the unit normal vector at the interface.

In the DN partitioned computational approach, force (traction) is calculated on the fluid side of the interface and applied as a boundary condition on the solid side of the interface. The traction is calculated by using the fluid stress tensor, consisting of the isotropic and viscous components. The viscous (deviatoric) component, defined by Newton's law of viscosity, is

$$\tau_F = \mu_F \left[\nabla\mathbf{v} + \nabla\mathbf{v}^{\mathrm{T}}\right], \tag{13}$$

where $\mu_F = \rho_F \nu_F$ is the fluid dynamic viscosity. Hence, the stress tensor for an incompressible Newtonian fluid reads as follows:

$$\sigma_{F,i} = -p\mathbf{I} + \tau_F = -p\mathbf{I} + \mu_F \left[\nabla\mathbf{v} + \nabla\mathbf{v}^{\mathrm{T}}\right], \tag{14}$$

and the traction at the interface reads as

$$\mathbf{t}_{F,i} = \mathbf{n}_i \cdot \sigma_{F,i} = -\mathbf{n}_i p + \mu_F \mathbf{n}_i \cdot \nabla\mathbf{v}_t - 2\mu_F \left(\nabla_s \cdot \mathbf{v}\right)\mathbf{n}_i + \mu_F \nabla_s v_n, \tag{15}$$

where $\mathbf{v}_t = (\mathbf{I} - \mathbf{nn}) \cdot \mathbf{v}$ is the tangential velocity component, $\nabla_s = \nabla - \mathbf{nn} \cdot \nabla$ is the surface tangential gradient operator and $v_n = \mathbf{n} \cdot \mathbf{v}$ is the normal velocity component. The third and fourth terms on the right-hand side of Eq. (15) can usually be omitted; otherwise, these terms can be calculated directly on the solid side of the interface, taking that the kinematic condition is valid.

2.4 Robin Boundary Condition for Pressure

In order to solve the fluid sub-problem, one also has to specify the boundary condition for pressure; this is required to calculate the value of pressure at the spatial domain boundary and to solve the pressure equation when a pressure-based collocated FV solver is used (as is the case in this work). The value of the pressure at the FSI interface (moving wall) is usually extrapolated from the interior of the spatial domain using first- or second-order practice. In this work, an alternative approach will be presented, which is required in order to derive a Robin BC for pressure. Namely, the boundary condition for pressure can also be derived from the momentum Eq. (2), which, in its differential non-conservative form, reads as

$$\frac{\partial\mathbf{v}}{\partial t} + (\mathbf{v} - \mathbf{v}_s) \cdot \nabla\mathbf{v} = \nabla \cdot (\nu\nabla\mathbf{v}) - \frac{1}{\rho}\nabla p. \tag{16}$$

Assuming a linear distribution of the velocity field near the non-permeable moving walls, one can derive an expression for the derivative of the pressure in the normal direction from Eq. (16)

$$\mathbf{n} \cdot \nabla p = -\rho \frac{\partial v_n}{\partial t}, \tag{17}$$

where v_n is the normal component of the fluid velocity. Equation (17) can be used as the boundary condition for the pressure at the FSI interface (and other walls), in which the temporal derivative of the velocity can be assumed as specified when the DN partitioned scheme is used.

DN partitioned schemes suffer from stability and/or convergence issues when the densities of the fluid and the structure are comparable, i.e., when strong added mass effects are present. For example, it has been shown in [10] that in the presence of a large added mass effect, the DN procedure needs a strong relaxation and displays slow convergence.

Partitioned schemes that can efficiently cope with the added mass effect are rare. One such scheme with good performance is the so-called kinematically coupled β-scheme [3, 4] based on the Lie operator splitting, in which the fluid and the structure sub-problems communicate via the initial conditions. The fluid and structure equations are split in such a way that the fluid problem is solved with a Robin-type BC including the structural inertia; this is the main ingredient of the scheme ensuring its unconditional stability.

Using the idea of the kinematically coupled β-scheme, one can state that fluid pressure at the interface is limited (or can be approximated) by the solid inertia

$$p \approx \rho_S h_S \frac{\partial v_n}{\partial t}, \tag{18}$$

where ρ_S is the structure density and h_S is the thickness of the structure (a procedure to calculate this thickness in the case of a thick structure will be defined later). Combining Eqs. (17) and (18) through the $\frac{\partial v_n}{\partial t}$ term, one can derive the Robin-type boundary condition for pressure:

$$p + \frac{\rho_S h_S}{\rho_F} \frac{\partial p}{\partial n} = 0, \tag{19}$$

where ρ_F is the density of the fluid. The Robin BC (19) is equivalent to the Robin BC obtained when the kinematically coupled β-scheme is applied with $\beta = 0$ to calculate the interaction between the inviscid Stokes fluid and the thin elastic structure [11].

The coupling scheme proposed in this work departs from the kinematically coupled β-scheme in such a way that, during the solid sub-problem solution, there is no communication with the fluid sub-problem through the initial condition, but only through the Neumann BC (specified traction). In order to preserve the accuracy of the overall scheme in the new computational framework, the Robin BC (19) is reformulated such that, in the context of the iterative solution procedure (strong coupling), it reduces to the exact boundary condition defined by Eq. (17) when the converged solution is reached

$$p^k + \frac{\rho_S h_S}{\rho_F} \left(\frac{\partial p}{\partial n} \right)^k = p^{(k-1)} - \rho_S h_S \left(\frac{\partial v_n}{\partial t} \right)^{(k-1)}, \tag{20}$$

where k is the iteration counter. One can notice that when the converged solution is reached ($p^k = p^{(k-1)}$), Eq. (20) becomes equivalent to Eq. (17).

In the case when a shell model is used to define the structure deformation, the thickness parameter h_S is a specified value. For a thick structure, the quantity h_S represents the virtual thickness, which is calculated in this work using the propagation speed of waves in an elastic media as follows [5]:

$$h_S = a_p \Delta t, \tag{21}$$

where Δt is the time step size and a_p is the propagation speed of the p-wave defined by

$$a_p = \sqrt{\frac{\lambda_S + 2\mu_S}{\rho_S}}. \tag{22}$$

As one can note, the virtual thickness parameter h_S depends only on the time step size and structure material properties.

3 Numerical Model

The discretisation procedure is separated into two distinct parts: discretisation of the computational domain and discretisation of the governing equations.

3.1 Discretisation of the Computational Domain

The time interval is split into a finite number of time steps Δt and the equations are solved in a time-marching manner. Both sub-problems are discretised in time using a first-order accurate implicit Euler scheme [12]. In general, the computational space is divided into a finite number of convex polyhedral control volumes (CVs) or cells bounded by convex polygons. Cells do not overlap and fill the spatial domain completely. Figure 1 shows a polyhedral control volume V_P with the computational point P located in its centroid, f is an arbitrary face with area S_f and unit normal vector $\mathbf{n}_f$, and it is shared with the neighbouring CV with the centroid N. The geometry of the CV is fully determined by the position of its vertices.

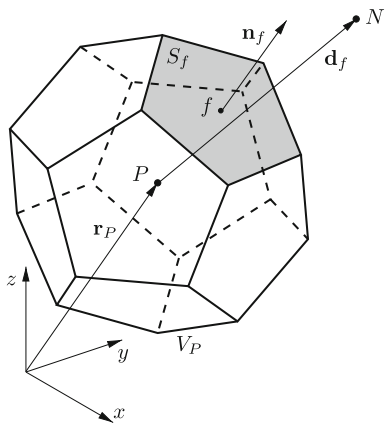


Fig. 1 Polyhedral control volume (cell)

While the fluid flow mesh is changing in time due to deformation of the fluid–solid interface, the solid mesh is always in its initial (undeformed) configuration due to the use of the total Lagrangian formulation.

3.2 Discretisation of the Governing Equations

3.2.1 Discretisation of the Fluid Model

The discretised fluid mathematical model with moving polyhedral mesh consists of the discretised momentum equation and the discretised pressure equation, in which the pressure equation is derived from the discretised continuity equation using the Rhie-Chow momentum interpolation method [13]. The FV discretisation of the fluid model is described in [14]. However, it should be noted that in [14], emphasis was on the moving interface between two fluid phases, and boundary conditions on moving walls were not considered. A comprehensive description of moving wall boundary conditions for the DN partitioned scheme can be found in [15]. Here, the implementation of the Robin BC for pressure will be briefly discussed and the influence on the velocity BC and flux calculation procedure are outlined.

The Robin BC (20) can be written in contracted form as follows:

$$c_0 p + c_1 \frac{\partial p}{\partial n} = r, \tag{23}$$

where $c_0 = 1$, $c_1 = \rho_S h_S / \rho_F$ and r represents the right-hand side of Eq. (20). Equation (23), discretized at the interface boundary face bi (Fig. 2), reads as

$$c_0 p_{bi} + c_1 \frac{p_{bi} - p_P}{\delta_{bn}} = r_{bi}, \tag{24}$$

where p_{bi} is the pressure at the interface boundary face centre, p_P is the pressure in the centre of the corresponding neighbouring cell P and δ_{bn} is the normal distance defined in Fig. 2. Equation (24) can now be used to define an expression for calculation of the pressure value at the interface boundary face centre

$$p_{bi} = \frac{\delta_n r_{bi}}{\delta_{bn} c_0 + c_1} + \frac{c_1}{\delta_{bn} c_0 + c_1} p_P. \tag{25}$$

The derivative of the pressure in the normal direction can now be expressed for the interface boundary face centre as follows:

$$\left(\frac{\partial p}{\partial n} \right)_{bi} = n_{bi} \cdot (\nabla p)_{bi} = \frac{r_{bi}}{\delta_{bn} c_0 + c_1} - \frac{c_0}{\delta_{bn} c_0 + c_1} p_P. \tag{26}$$

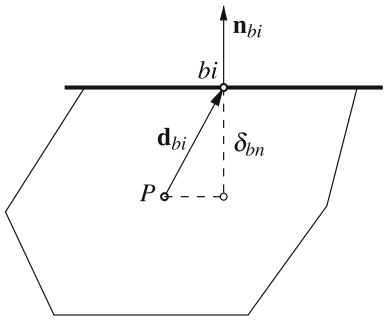


Fig. 2 Interface boundary face bi and its neighbouring cell P

Application of the proposed Robin BC for pressure also requires modification of the discretized pressure equation. As is shown in [14], the discretized pressure equation for cell P reads as

$$\sum_f \left(\frac{1}{a_p}\right)_f \mathbf{n}_f \cdot (\nabla p)_f S_f = \sum_f \mathbf{n}_f \cdot \left(\frac{\mathbf{H}_P}{a_P}\right)_f S_f, \tag{27}$$

where the subscript f represents the face-centre values and the summation is performed over all faces enclosing the considered cell P. Taking into account the first-order accurate Euler implicit temporal discretisation of the fluid model, terms $(1/a_P)_f$ and $(\mathbf{H}_P/a_P)_f$ are calculated at the boundary faces coinciding with the fluid side of the interface as follows:

$$\left(\frac{1}{a_p}\right)_{bi} = \Delta t, \tag{28}$$

$$\left(\frac{\mathbf{H}_P}{a_P}\right)_{bi} = \mathbf{v}_{bi}^{[m-1]}, \tag{29}$$

where $\mathbf{v}_{bi}^{[m-1]}$ is the fluid velocity at the interface in the previous time step. After the solution of the pressure equation, the volume flow rate at the interface boundary face is calculated as follows:

$$\dot{V}_{bi} = \mathbf{n}_{bi} \cdot \left(\frac{\mathbf{H}_P}{a_P}\right)_{bi} S_{bi} - \left(\frac{1}{a_p}\right)_{bi} \mathbf{n}_{bi} \cdot (\nabla p)_{bi} S_{bi}, \tag{30}$$

where it is assumed that the derivative of the pressure in the normal direction satisfies Eq. (17).

The boundary condition for velocity (momentum equation) at the interface is of the Dirichlet type, in which the tangential component of velocity is set to the value calculated at the solid side of the interface; the normal component is calculated using the volume flow rate $\dot{V}_{bi}$ obtained after the solution of the pressure equation

$$\mathbf{v}_{bi} = (\mathbf{I} - \mathbf{n}_{bi}\mathbf{n}_{bi}) \cdot \mathbf{v}_{S,bi} + \mathbf{n}_{bi} \frac{\dot{V}_{bi}}{S_{bi}}, \tag{31}$$

where $\mathbf{v}_{S,bi}$ is the velocity at the centre of the interface boundary face transferred from the solid to the fluid side of the interface.

Fluid mesh deformation is performed using the Laplace mesh motion equation with variable diffusivity, as described in [14, 16]. Instead of using an FE discretisation for the Laplace mesh motion equation, this work discretises the motion equation using the cell-centred FV method, in which vertex displacements are obtained using a least squares reconstruction procedure.

3.2.2 Discretization of the Solid Model

In terms of the solid model, the second-order FV discretisation of the integral conservation equation, Eq. (8), transforms the surface integrals into sums of face integrals; the face integrals and volume integrals are then approximated using the second-order accurate mid-point rule. The spatially discretised counterpart of the momentum equation, Eq. (8), for the control volume V_P reads as

$$\begin{aligned}(\rho_{S,0})_P \left[\frac{\partial}{\partial t}\left(\frac{\partial \mathbf{u}}{\partial t}\right)\right]_P V_P - \sum_f (2\mu_{S,f} + \lambda_{S,f})\mathbf{n}_f \cdot (\nabla \mathbf{u})_f S_f = \\ \sum_f \mathbf{n}_f \cdot \mathbf{q}_f S_f + (\rho_{S,0})_P \mathbf{b}_P V_P,\end{aligned} \tag{32}$$

where the subscript P represents the cell-centre value and subscript f represents a face-centre value.

The temporal discretisation of Eq. (32) is performed using the first-order accurate implicit Euler scheme. All terms in Eq. (32) are evaluated at the new time instance $t^{[m]} = t^{[m-1]} + \Delta t$. The acceleration $\frac{\partial}{\partial t}\left(\frac{\partial \mathbf{u}}{\partial t}\right)$ and the velocity $\frac{\partial \mathbf{u}}{\partial t}$ are discretised at the new time instance, using the two-time-level finite difference formula as follows:

$$\left[\frac{\partial}{\partial t}\left(\frac{\partial \mathbf{u}}{\partial t}\right)\right]_P^{[m]} = \frac{\left(\frac{\partial \mathbf{u}}{\partial t}\right)_P^{[m]} - \left(\frac{\partial \mathbf{u}}{\partial t}\right)_P^{[m-1]}}{\Delta t}, \tag{33}$$

$$\left(\frac{\partial \mathbf{u}}{\partial t}\right)_P^{[m]} = \frac{\mathbf{u}_P^{[m]} - \mathbf{u}_P^{[m-1]}}{\Delta t}. \tag{34}$$

The face normal gradient of displacement $\mathbf{n}_f \cdot (\nabla \mathbf{u})_f$ is discretised using the central scheme with non-orthogonal and skewness correction (see Fig. 1)

$$\mathbf{n}_f \cdot (\nabla \mathbf{u})_f = \frac{\mathbf{u}_N - \mathbf{u}_P}{d_{fn}} + \frac{\mathbf{k}_N \cdot (\nabla \mathbf{u})_N - \mathbf{k}_P \cdot (\nabla \mathbf{u})_P}{d_{fn}}, \tag{35}$$

where d_{fn} is the normal distance between points N and P,

$$d_{fn} = \mathbf{n}_f \cdot \mathbf{d}_f = \mathbf{n}_f \cdot (\mathbf{r}_N - \mathbf{r}_P), \tag{36}$$

and $\mathbf{k}_P$ and $\mathbf{k}_N$ are the correction vectors calculated as follows:

$$\mathbf{k}_P = (\mathbf{I} - \mathbf{n}_f \mathbf{n}_f) \cdot (\mathbf{r}_f - \mathbf{r}_P), \tag{37}$$

$$\mathbf{k}_N = (\mathbf{I} - \mathbf{n}_f \mathbf{n}_f) \cdot (\mathbf{r}_N - \mathbf{r}_f). \tag{38}$$

The first term on the right-hand side of Eq. (35) is treated implicitly, while the correction term is explicit. The cell-centre displacement gradient used for the calculation of the correction term in Eq. (35) is calculated using the vertex-based Gauss-Green method, which gives a gradient of second-order accuracy irrespective of the mesh quality. Here, the face-centre displacement is calculated by averaging corresponding vertex displacements.

The face-centre gradient needed for the evaluation of tensor $\mathbf{q}_f$ in the first term on the right-hand side of Eq. (32) is calculated separately in the normal and tangential directions. The face-centre displacement gradient in the normal direction is calculated using Eq. (35), while the face-centre gradient in the tangential direction is calculated by applying the vertex-based Gauss-Green method on a flat polygonal face. The vertex displacements are reconstructed from the cell-centre displacements of the cells surrounding the vertex using the weighted least-squares method and linear fitting function.

When Eqs. (33), (34) and (35) are substituted into Eq. (32), the fully discretised form of the linear momentum conservation law, Eq. (8), can be written in the form of a linear algebraic equation, which, for cell P, reads as

$$a_P \, \mathbf{u}_P^{[m]} + \sum_N a_N \, \mathbf{u}_N^{[m]} = \mathbf{R}_P, \tag{39}$$

where the diagonal coefficient a_P, the neighbour coefficient a_N and the source term $\mathbf{R}_P$ are defined by the following expressions:

$$a_P = \frac{(\rho_{S,0})_P V_P}{\Delta t^2} + \sum_f (2\mu_{S,f} + \lambda_{S,f}) \frac{S_f}{d_{fn}}, \tag{40}$$

$$a_N = -(2\mu_{S,f} + \lambda_{S,f}) \frac{S_f}{d_{fn}}, \tag{41}$$

$$\begin{aligned}\mathbf{R}_P =&(\rho_{S,0})_P V_P \left[\frac{\mathbf{u}_P^{[m-1]}}{\Delta t^2} + \frac{1}{\Delta t}\left(\frac{\partial \mathbf{u}}{\partial t}\right)_P^{[m-1]}\right] \\ &+ \sum_f (2\mu_{S,f} + \lambda_{S,f}) \frac{\mathbf{k}_N \cdot (\nabla \mathbf{u})_N^{[m]} - \mathbf{k}_P \cdot (\nabla \mathbf{u})_P^{[m]}}{d_{fn}} S_f \\ &+ \sum_f \mathbf{n}_f \cdot \mathbf{q}_f^{[m]} S_f + \sum_f (\rho_{S,0})_P \mathbf{b}_P^{[m]} V_P .\end{aligned} \tag{42}$$

3.3 Solution Procedure for Fluid and Solid Models

The incompressible fluid flow model, which is discretised on a moving finite volume mesh, is solved for velocity and pressure using the PISO algorithm [17]. A detailed description of the fluid model solution procedure is given in [14].

For solution of the discretised solid model, Eq. (39) is assembled for all CVs in the computational mesh, resulting in the following system of linear algebraic equations:

$$[A] \cdot \{\mathbf{u}\} = \{\mathbf{R}\}, \tag{43}$$

where $[A]$ is a sparse matrix, with coefficients a_P on the diagonal and a_N off the diagonal, $\{\mathbf{u}\}$ is the solution vector consisting of displacements $\mathbf{u}_P$ for all CVs and $\{\mathbf{R}\}$ is the right-hand side vector consisting of source terms $\mathbf{R}_P$ for all CVs.

The system of Eq. (43) is solved using a segregated algorithm, in which the three components of the displacement vector are temporarily decoupled and solved separately. Since nonlinear and coupling terms depending on the unknown displacement vector are placed in the right-hand side vector, the system is solved in an iterative manner, in which the right-hand side vector $\{\mathbf{R}\}$ is updated at the beginning of each outer iteration through use of the displacement vector from the previous iteration. When the solution changes less than some predefined tolerance, the system is considered to be solved. This is performed for every time step of the transient simulation.

The sparse matrix $[A]$ from Eq. (43) is symmetric and weakly diagonally dominant, and the corresponding system of equations is solved using the preconditioned conjugate gradient iterative solver [18, 19]. There is no need to solve the system to a fine tolerance, since the right-hand side vector is only an approximation based on the displacement vector from the previous iteration; reduction of the residuals by an order of magnitude normally suffices.

3.4 Solution Procedure for Fluid–Structure Interaction

As mentioned earlier, a partitioned approach is adopted for the fluid–structure interaction solution procedure, in which the flow model and the structural model are

solved separately using different solvers in the FV framework. The coupled fluid–structure interaction problem is decomposed using the Robin-Neumann procedure, in which the structural model is solved for a given force on the interface, while the fluid model is solved using the Robin BC for pressure and corresponding BC for velocity. The equilibrium of the force and velocity (or displacement) on the fluid–structure interface is enforced at each time step using a strongly coupled procedure by performing iterations between the fluid and solid solvers.

The fluid–structure interaction solution procedure is summarised in Algorithm 1. The procedure is mostly self-explanatory, except perhaps for the interface residual calculation, defined as the difference between the solid side interface displacement, obtained by solving the structural model, and the fluid side interface displacement, used to move the fluid mesh before solving the fluid model

$$\{\mathbf{r}\}_i^k = \{\mathbf{u}\}_{S,i}^k - \{\mathbf{u}\}_{F,i}^k, \tag{44}$$

where the superscript k represents the iteration number, $\{\mathbf{u}\}_{F,i}$ is the vector consisting of vertex displacements on the fluid side of the interface and $\{\mathbf{u}\}_{S,i}$ is the vector consisting of displacements of the solid side of the interface mapped to the vertices on the fluid side of the interface.

The vertex displacements on the fluid side of the interface, used in the next iteration to move the fluid mesh, are calculated as follows:

$$\{\mathbf{u}\}_{F,i}^{k+1} = \{\mathbf{u}\}_{F,i}^k + \{\mathbf{r}\}_i^k, \tag{45}$$

where the full residual is applied without under-relaxation.

At the end of each fluid–structure interaction iteration, the L^2-norm of the interface residual vector is calculated and it is checked whether a converged solution has been reached.

Algorithm 1 Fluid–structure interaction iterative solution procedure

1: Switch to the next time step.
2: Predict the interface displacement: solve the solid model using the fluid force from the previous time step, transpose the displacement and acceleration from the solid to the fluid side of the interface, solve the mesh motion equation and move the fluid mesh.
3: Start the FSI strongly coupled iterative procedure.
4: Switch to the next iteration.
5: Solve the fluid model using the proposed Robin BC for pressure at the interface.
6: Transpose the force from the fluid to the solid side of the interface and solve the solid model.
7: Transpose the displacement and acceleration from the solid to the fluid side of the interface, solve the mesh motion equation and move the fluid mesh.
8: Calculate the interface residual at the fluid side of the interface.
9: **if** converged **then**
10: Go to next time step (line 1)
11: **else**
12: Go to next iteration (line 4)
13: **end if**

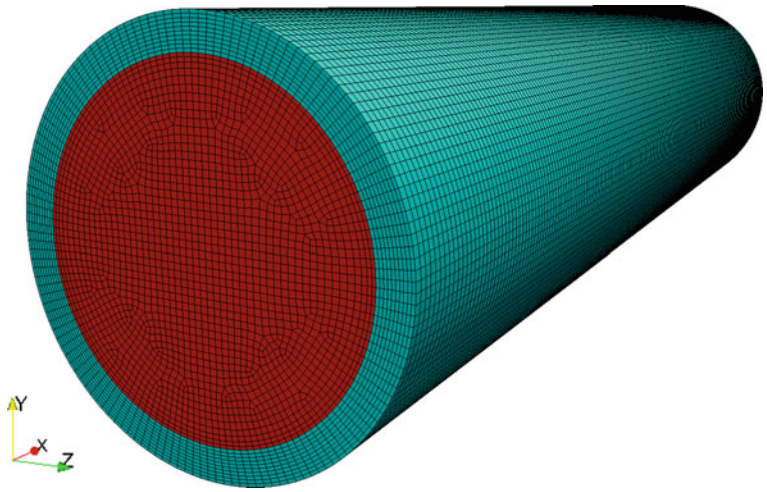


Fig. 3 Discretised spatial domain for elastic tube test case

4 Numerical Results

4.1 Wave Propagation in an Elastic Tube

This is a standard FSI test case intended to demonstrate the capability of the numerical model to predict blood flow in large arteries [2].[1] The spatial computational domain consists of a straight flexible tube with radius $r = 0.005$ m, length $L = 0.05$ m and wall thickness $\delta = 0.001$ m. The tube's wall is a St. Venant-Kirchhoff material with density $\rho_S = 1200\,\mathrm{kg/m^3}$, Young's modulus $E = 3 \times 10^5\,\mathrm{N/m^2}$ and Poisson's ratio $\nu = 0.3$. The tube is clamped in all directions at the inlet and outlet. The fluid is incompressible with a density of $\rho_F = 1000\,\mathrm{kg/m^3}$ and a dynamic viscosity of $\mu = 0.003$ pas. Both the fluid and the structure are initially at rest. During the first 0.003 s, a uniform overpressure of $1333.2\,\mathrm{N/m^2}$ is applied at the inlet.

Figure 3 shows the spatial domain discretised by a hexahedral finite volume mesh. The fluid part of the mesh consists of 449, 600 finite volumes, while the solid part consists of 288, 000 finite volumes. The numerical solution presented here is obtained using the time step size $\Delta t = 2.5 \times 10^{-5}$ s, which corresponds to the virtual solid thickness $h_S = 0.00046$ m, and the maximum Courant number based on p-wave propagation speed $Co = 4.6$. The strongly coupled solution procedure was applied, in which four sub-iterations (on average) were required to reduce the L^2-norm of the interface residual by six orders of magnitude.

[1] It is assumed that for the considered test case (large artery and small pressure change), the blood can be considered as an incompressible Newtonian fluid.

Figure 4 shows the velocity field in both the fluid and solid parts of the spatial domain at the time instance 0.005 s, while Fig. 5 shows the pressure field in the fluid part and equivalent stress in the solid part of the spatial domain at the same time instance.

Pressure and axial velocity along the pipe axis for a few time instances are shown in Figs. 6 and 7. Using the procedure described in [20], one can obtain the simulated pressure wave speed of 4.54 m/s. The analytical solution of the pressure wave speed, c_F, can be found in [21] and, for incompressible fluid, reads as

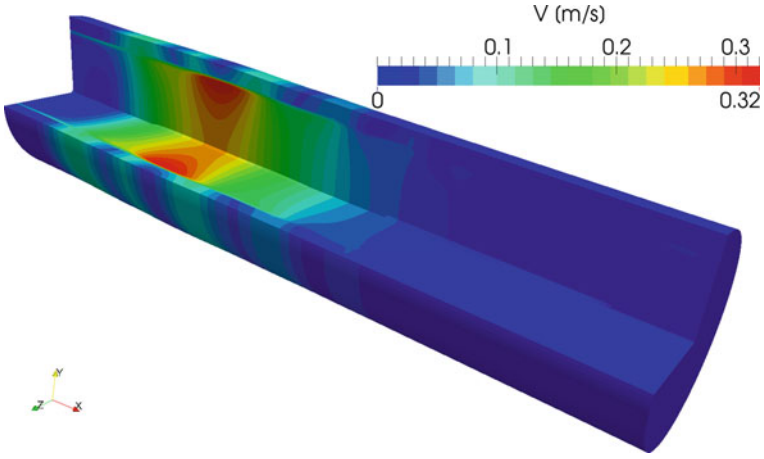


Fig. 4 Velocity field in the fluid and solid parts of the spatial computation domain at the time instance 0.005 s for the elastic tube test case

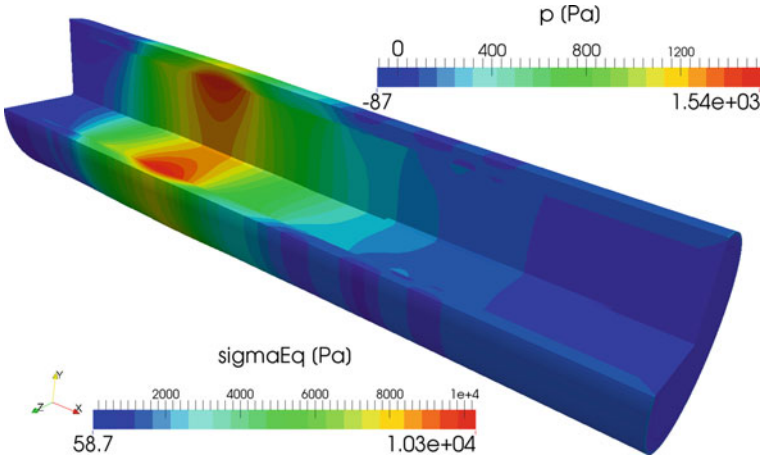


Fig. 5 Pressure field in the fluid and equivalent stress in the solid part of the spatial computational domain at the time instance 0.005 s for the elastic tube test case

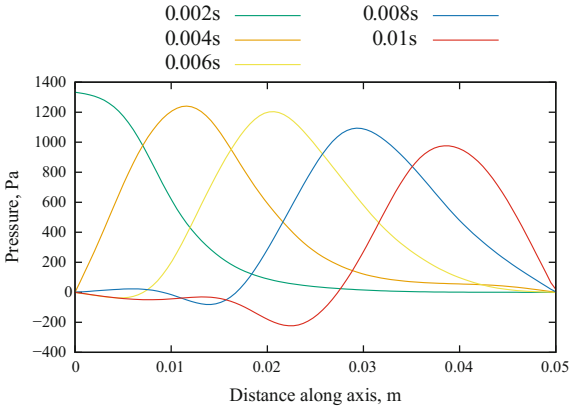


Fig. 6 Pressure variation along the tube axis as a function of time for the elastic tube test case

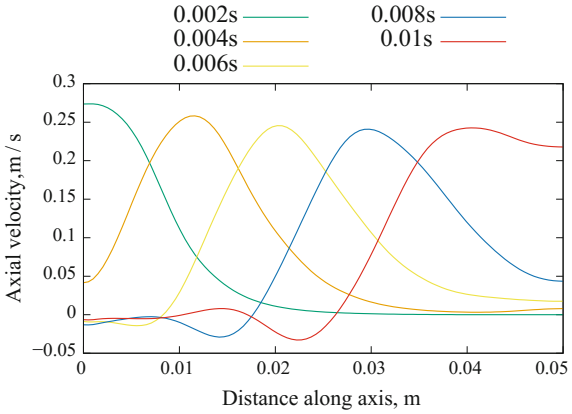


Fig. 7 Axial velocity variation along the tube axis as a function of time for the elastic tube test case

$$c_F = \sqrt{\frac{E\delta}{2\rho_F r}\left[\frac{\delta}{r}(1+\nu)+\frac{2r}{2r+\delta}\right]^{-1}}. \tag{46}$$

Taking into account the axial stress waves in the tube wall, the final expression for analytical wave speed can be written as [20]:

$$\tilde{c_F} = c_F \sqrt{1-\frac{\nu^2}{1-\frac{\delta}{2r}\frac{\rho_F}{\rho_S}}}. \tag{47}$$

Using the current case data, a value of 4.81 m/s can be obtained, which is in good agreement with the simulation results.

In order to evaluate the temporal accuracy of the method, the calculation is carried out for four different time step sizes 5×10^{-5} s, 2.5×10^{-5} s, 1.25×10^{-5} s and 6.25×10^{-6} s, and the radial displacement is monitored at the inner side of the

tube's midsection. The results are shown in Fig. 8. Temporal accuracy is evaluated for the time instance when the radial displacement reaches its maximum. The order of temporal accuracy is calculated as [22]

$$p = \frac{\ln\left[\frac{\phi_3-\phi_2}{\phi_2-\phi_1}\right]}{\ln(r)}, \tag{48}$$

where ϕ_1, ϕ_2 and ϕ_3 are the numerical solutions obtained using the three consecutive time step sizes, refined using constant refinement ratio $r = \Delta t_1/\Delta t_2 = \Delta t_2/\Delta t_3 = 2$. The resulting calculated order of temporal accuracy is 1.099, i.e., above the theoretical order of accuracy for the implicit Euler temporal discretisation scheme.

In order to show that the proposed partitioned FSI coupling scheme is stable even if it is used as a loosely coupled scheme, the calculation is performed with only one and two FSI iterations and the calculated solution is compared with the corresponding fully converged solution. Results of this analysis are shown in Fig. 9. Although the solution obtained after one FSI iteration substantially departs from the fully converged solution, it can be concluded that the applied loosely coupled solution procedure is stable for the considered test case. The solution obtained by performing only one additional FSI iteration (two in total) agrees well with the fully converged solution.

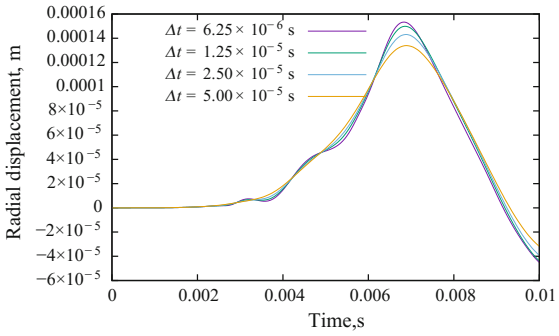


Fig. 8 Calculating temporal accuracy – radial displacement at the inner wall at the midsection of the tube

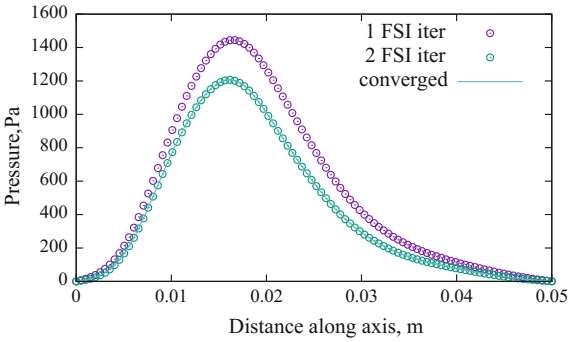


Fig. 9 Pressure variation along the tube axis as a function of time for the elastic tube test case. Solutions obtained with one and two FSI iterations are compared with the fully converged solution

4.2 Enclosed Domain: A Balloon-Type Problem

The purpose of this example is to demonstrate the capabilities of the proposed fluid–solid coupling procedure to handle FSI cases with geometrically nonlinear solid deformation and in which the fluid domain is entirely enclosed by Dirichlet boundary conditions, i.e., prescribed velocities. The test case represents a three-dimensional variant of a similar test case used in [23], in which fluid and solid properties and spatial domain dimensions are modified in such a way as to be within the range of hemodynamics applications. A similar approach is used in [24] to test a balloon-type problem. The fluid domain is a cube of size $[-0.025, 0.025] \times [-0.025, 0.025] \times [-0.025, 0.025]$ m, enclosed by a solid wall of thickness 0.005 mm. The fluid and solid properties are the same as in the previous example. The fluid enters into the cavity through a channel of square cross section (0.016×0.016 m) positioned in the middle of the cube side at the plane $x = -0.025$ m. At the inlet of the channel, a uniform velocity changing in time is specified according to the following expression: $\mathbf{v}(t) = [\sin \frac{2\pi t}{0.2}, 0, 0]$ m/s. Due to the symmetry of the geometry, only one quarter of the spatial domain is simulated. The initial shape of the discretized spatial domain is shown in Fig. 10. The spatial domain is discretised using a uniform structured mesh (element size 0.005 m) and the calculation is performed using a constant time step size $\Delta t = 10^{-4}$ s. The strongly coupled solution procedure was applied, in which

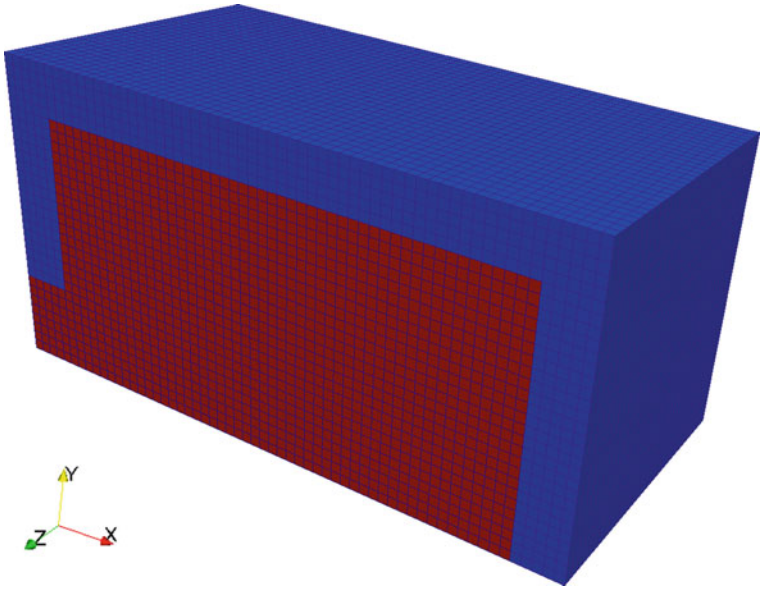


Fig. 10 Initial shape of the spatial domain for the balloon-type problem. Due to symmetry, only one quarter of the whole domain is used

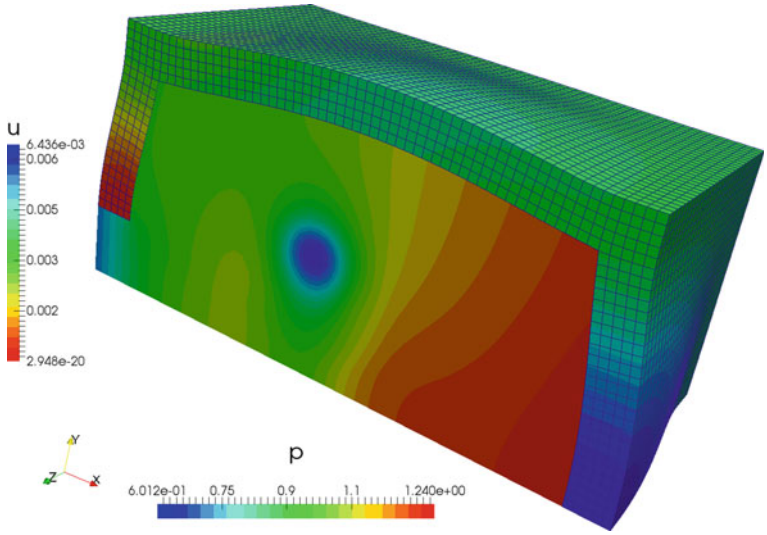


Fig. 11 Deformed shape of the spatial domain for the balloon-type problem at the time instance $t = 0.1$ s. The solid domain is coloured by the displacement field (u), while the fluid domain is coloured by the kinematic pressure field (p)

only three sub-iterations (on average) were required to reduce the L^2-norm of the interface residual by six orders of magnitude.

According to the specified inlet boundary condition, at the time instance $t = 0.2$ s, the initial volume of the fluid domain must be recovered, and at the time instance $t = 0.1$ s (end of the filling phase), the maximal fluid domain volume should be reached. Figure 11 shows the shape of the spatial domain at the time instance $t = 0.1$ s, in which the colours represent kinematic pressure (p/ρ) and displacement. The relative change of the fluid domain volume as a function of time is shown in Fig. 12. One can notice that the volume of fluid entering/leaving the cavity closely follows the change in the fluid domain volume, proving that the applied numerical method conserves fluid mass in the case of a pure Dirichlet domain. At the end of the calculated time interval ($t = 0.2$ s), the volume of the fluid domain is close to the initial volume. The same figure also shows temporal variation of the average pressure in the fluid domain.

5 Conclusions

A self-contained fluid–structure interaction solver based on an FV discretisation method and partitioned solution procedure is presented. Both fluid and solid models are discretised in space using the second-order accurate cell-centred finite volume method, while numerical integration of the models in time is performed using a first-

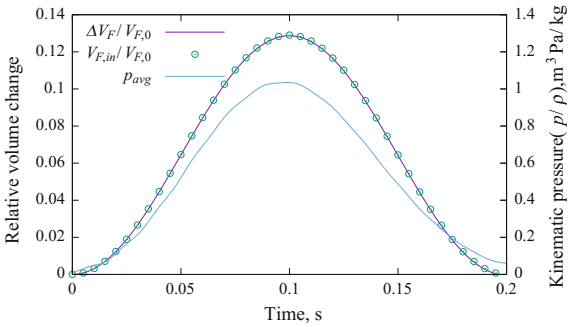


Fig. 12 Change of the fluid domain volume ($^{\Delta V_F}/_{V_{F,0}}$) and fluid volume entering/leaving the fluid domain ($^{V_{F,in}}/_{V_{F,0}}$) as a function of time, where $V_{F,0}$ is the initial volume of the fluid domain. On the secondary axis, the temporal variation of the volume average pressure over the fluid domain is shown

order accurate implicit Euler scheme. Coupling between the fluid and structure is performed using a Robin-Neumann partitioned procedure based on a new Robin BC for pressure derived from the coupled β-scheme. The main ingredient of the scheme, ensuring its stability, is the inclusion of the structural inertia in the Robin BC for pressure. In such a way, there is no need for any under-relaxation during information transfer between fluid and solid, which ensures good convergence of the solution procedure.

The solver has been tested on the wave propagation in an elastic tube test case characterised by a low solid-to-fluid density ratio. The temporal consistency of the numerical model is shown by proving the first-order accuracy in time, which is in agreement with the theoretical order of accuracy of the applied implicit Euler scheme. The stability of the method is demonstrated for both strongly coupled and the loosely coupled versions of the solution procedure. It is shown that the proposed methodology can efficiently handle FSI cases in which the fluid domain is entirely enclosed by Dirichlet boundary conditions, even for the case of geometrically nonlinear elastic deformation.

In future work, temporal accuracy will be increased to second-order and the complete kinematically coupled β-scheme will be implemented.

References

1. Küttler, U., Wall, W.A.: Fixed–point fluid–structure interaction solvers with dynamic relaxation. Computational mechanics **43**(1), 61–72 (2008)
2. Degroote, J., Bathe, K.J., Vierendeels, J.: Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction. Computers and structures **87**, 793–801 (2009)
3. Bukač, M., Čanić, S., Glowinski, R., Tambača, J., Quaini, A.: Fluid-structure interaction in blood flow capturing non-zero longitudinal structure displacement. Journal of Computational Physics **235**, 515–541 (2013)
4. Guidoboni, G., Glowinski, R., Cavallini, N., Čanić, S.: Stable loosely-coupled-type algorithm for fluid-structure interaction in blood flow. Journal of Computational Physics **228**(18), 6916–6937 (2009)

5. Banks, J., Henshaw, W., Schwendeman, D.: An analysis of a new stable partitioned algorithm for fsi problems. part i: Incompressible flow and elastic solids. Journal of Computational Physics **269**, 108–137 (2014)
6. Demirdžić, I., Perić, M.: Space conservation law in finite volume calculations of fluid flow. International journal for numerical methods in fluids **8**(9), 1037–1050 (1988)
7. Thomas, P.D., Lombard, C.K.: Geometric conservation law and its application to flow computations on movining grids. AIAA Journal **17**, 1030–1037 (1979)
8. Jasak, H., Weller, H.G.: Application of the finite volume method and unstructured meshes to linear elasticity. International journal for numerical methods in engineering **48**(2), 267–287 (2000)
9. Batchelor, F.R.: An introduction to fluid dynamics. Cambridge University Press, Cambridge (1967)
10. Causin, P., Gerbeau, J., Nobile, F.: Added-mass effect in the design of partitioned algorithms for fluid?structure problems. Computer Methods in Applied Mechanics and Engineering **194**((42-44)), 4506–4527 (2005)
11. Čanić, S., Muha, B., Bukač, M.: Stability of the kinematically coupled β-scheme for fluid-structure interaction problems in hemodynamics. International Journal of Numerical Analysis and Modeling **12**(1), 54–80 (2015)
12. Ferziger, J.H., Perić, M.: Computational methods for fluid dynamics. Springer Verlag, Berlin-New York (1995)
13. Rhie, C.M., Chow, W.L.: A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. AIAA Journal **21**, 1525–1532 (1983)
14. Tuković, Ž., Jasak, H.: A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow. Computers and fluids **55**, 70–84 (2012)
15. Gillebaart, T., Blom, D.S., van Zuijlen, A.H., Bijl, H.: Time consistent fluid structure interaction on collocated grids for incompressible flow. Computer Methods in Applied Mechanics and Engineering **298**(0), 159–182 (2016)
16. Jasak, H., Tuković, Ž.: Automatic mesh motion for the unstructured finite volume method. Transactions of FAMENA **30**(2), 1–20 (2006)
17. Issa, R.I.: Solution of the implicitly discretised fluid flow equations by operator-splitting. Journal of computational physics **62**(1), 40–65 (1986)
18. Jacobs, D.A.H.: Preconditioned Conjugate Gradient methods for solving systems of algebraic equations. Tech. Rep. RD/L/N193/80, Central Electricity Research Laporatories (1980)
19. Kershaw, D.: The incomplete cholesky-conjugate gradient method for the iterative solution of systems of linear equations. Journal of Computational Physics **26**(1), 43–65 (1978)
20. Kanyanta, V., Ivankovic, A., Karac, A.: Validation of a fluid-structure interaction numerical model for predicting flow transients in arteries. Journal of Biomechanics **42**(11), 1705–1712 (2009)
21. Wylie, E.B., Streeter, V.L.: Fluid Transients in Systems. Englewood Cliffs, New York (1993)
22. Roache, P.J.: Quantification of uncertainty in computational fluid dynamics. Annual Review of Fluid Mechanics **29**, 123–160 (1997)
23. Küttler, U., Förster, C., Wall, W.A.: A solution for the incompressibility dilemma in partitioned fluid–structure interaction with pure dirichlet fluid domains. Computational mechanics **38**, 417–429 (2006)
24. Badia, S., Nobile, F., Vergara, C.: Robin-robin preconditioned krylov methods for fluid-structure interaction problems. Computer Methods in Applied Mechanics and Engineering **198**, 2768–2784 (2009)

CAD-Based Parameterization for Adjoint Optimization

Marios Damigos and Eugene De Villiers

Abstract Manipulating CAD geometry using primitive components rather than the originating software is typically a challenging prospect. The parameterisation used to define the geometry of a model is often integral to the efficiency of the design. Even more crucial are the relations (constraints) between those parameters that do not allow the model to be under-defined. However, access to these parameters is lost when making the CAD model portable. Importing a standard CAD file gives access to the Boundary Representation (BRep) of the model and consequently its boundary surfaces which are usually trimmed patches. Therefore, in order to connect Adjoint optimization and Computational Fluid Dynamics to the industrial design framework (CAD) in a generic manner, the BRep must be used as a starting point to produce volume meshes and as a means of changing a model's shape. In this study, emphasis is given firstly, to meshing (triangulation) of a BRep model as a precursor to volume meshing and secondly, to the use of techniques similar to Free Form Deformation for changing the model's shape.

1 Introduction

One of the biggest challenges in modern-day CFD and optimization is to establish the missing link to industrial design. Initially, CFD software acts upon a discrete geometry, whereas a designer provides that geometry in CAD format. For example, in OpenFOAM®, the native hex dominant mesher requires a triangulated surface as input (usually in STL format) to create a volume mesh. A tool for the reliable creation of surface meshes from CAD input is thus a necessity.

M. Damigos (✉)
ENGYS S.R.L.,Trieste, Italy
e-mail: mariosg.damigos@gmail.com

E. De Villiers
ENGYS Ltd., London, UK
e-mail: e.devilliers@engys.com

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_2

Secondly, each CAD software has its own formatted parameterization tools about which its vendors are sensitive. Due to the closed source nature of the most popular CAD packages, the above-mentioned formats cannot be accessed. The two most popular options employed thus far are

1. Request access to a specific CAD package and make its format available. This is obviously a very limiting option. Furthermore, if one wants CAD to be included in an optimization routine, then, obviously (for the adjoint technique), the CAD package has to be differentiated, which is seldom possible.
2. Employ feature recognition algorithms and extract a feature tree from the Boundary Representation (BRep) [1–3]. This type of algorithm, is not commonly successful, and thus, this option lacks automation capabilities.

Due to the setbacks of these two options, one must use an alternative way to access a model's information. A common choice is to use standard CAD formats such as STEP or IGES [4], which contain the BRep of a model (see Sect. 1.1). BRep model is most commonly described by NURBS (see Sect. 1.2), and therefore this leads to the use of free–form surfaces as a means to change a CAD model's shape.

Both of these challenges are tackled by introducing the open source CAD package OpenCascade Technology (OCCT) [5] into the OpenFOAM® framework. The following application has been developed by using the OpenFOAM® and OCCT libraries in concert.

1.1 Boundary Representation

Boundary Representation [6, 7] is a method for representing shapes in solid modeling. A solid is described by the BRep format using surface elements defining the interface between solid and non-solid volumes (Fig. 1).

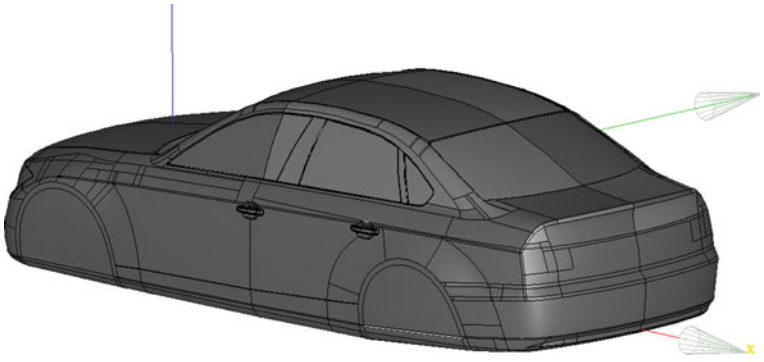


Fig. 1 Boundary representation model of the DrivAER car model [8]

The BRep format is composed of two parts: Topological data and Geometry. The topology of a BRep is created using vertices, edges, faces, shells and ultimately solids. Regarding their underlying geometry

1. Vertices: The underlying geometry of vertices is simply 3D points.
2. Edges: Edges are curves bounded by the points describing their boundary vertices. In the general case, the geometry of the edge is only a segment of its underlying curve, since the topological bounds of an edge and the geometrical bounds of a curve are not strictly identical.
3. Faces: Similarly to edges, faces are described by surfaces bounded by a closed loop of edges. Moreover, the geometry of the face is, in general, a part of its underlying surface, since its boundary loop of edges does not coincide with the natural bounds of the surface.
4. Shells: A shell is composed of multiple faces connected to each other and has no particular underlying geometry
5. Solids: A solid, similarly to a shell, does not have an underlying geometry and is practically the volume bounded by a collection of shells.

1.2 NURBS Curves and Surfaces

The geometry of a BRep is most commonly described by Non-Uniform Rational B-Splines or simply NURBS [9]. NURBS geometry can be defined using three things: a segmented parameter space for each parametric direction, a degree for each parametric direction, and a set of weighted control points.

1.2.1 Uni-variate NURBS – Curves

The formula that gives ys points on a NURBS curve is as follows:

$$\mathbf{C}(u) = \frac{\sum_{i=1}^{nCp} N_i^p(u) \cdot w_i \mathbf{P}_i}{\sum_{i=1}^{nCp} N_i^p(u) \cdot w_i}, \tag{1}$$

where nCp is the number of control points in the control polygon, $\mathbf{P}_i$ is the vector containing the coordinates of control point i, w_i is the corresponding weight and $N_i^p(u)$ is the i–th basis function of degree p, evaluated at parameter u. In order to evaluate a basis function, a non-decreasing vector of real values is required, also called a knot vector. The knot vector will be of the form

$$\mathbf{U} = [U_1, U_2, \ldots U_m]. \tag{2}$$

The parameter u spans between the values U_1 and U_m ($U_1 < U_m$) and every internal value of the vector defines a break point that segments the parameter span. The i-th basis function is then calculated as:

$$N_i^0(u) = \begin{cases} 1 & \text{if } U_i \leq u < U_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_i^p(u) = \frac{u - U_i}{U_{i+p} - U_i} N_i^{p-1}(u) + \frac{U_{i+p+1} - u}{U_{i+p+1} - U_{i+1}} N_{i+1}^{p-1}(u). \tag{3}$$

1.2.2 Bi-variate NURBS – Surfaces

The formula giving the points of a NURBS surface is the following:

$$\mathbf{S}(u, v) = \frac{\sum_{i=1}^{nCp} \sum_{j=1}^{mCp} N_i^p(u) N_j^q(v) \cdot w_{i,j} \mathbf{P}_{i,j}}{\sum_{i=1}^{nCp} \sum_{j=1}^{mCp} N_i^p(u) N_j^q(v) \cdot w_{i,j}}. \tag{4}$$

Similarly to NURBS Curves, $\mathbf{P}_{i,j}$ is the (i, j)-th control point of a $nCp \times mCp$ control grid and $w_{i,j}$ is the corresponding weight. Parameters u and v come with knot vectors U, V defined as in Eq. 2. Then, the basis functions $N_i^p(u)$ and $N_j^q(v)$ are calculated using Eq. 3 .

1.3 Connecting CAD to CFD

In order to start a CFD simulation or to proceed with an adjoint-based optimization [10, 11], a CAD model has to be imported (in our case, from a STEP or IGES file) in the form of a BRep. In Fig. 2 the flow of an optimization cycle is shown. This study deals with the surface meshing and the CAD-based shape optimization

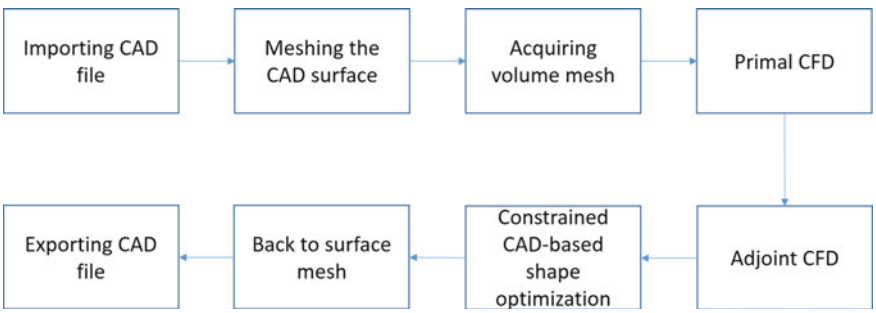


Fig. 2 The work flow of an adjoint optimization cycle

step. After the meshing of the surface, an STL file can be exported, on which the OpenFOAM® native hex dominant mesher can build the volume mesh. Afterwards, a method for changing the shape of the CAD model is shown, suitable for optimization applications.

2 Meshing of the CAD Surfaces

The need to develop a triangulation tool arises from the fact that most open source surface meshers, require a high number of parameters as input. Netgen [12], for instance, requires 15 parameters as input. Furthermore, these parameters depend on the dimensions of each model, and thus the surface meshing sequence cannot become automated. Another challenge that arises when meshing a CAD model comes from various defects resulting from the model's possibly poor design or information loss during export. When reading a CAD file, for example, the topology inside it is not always consistent. This problem makes it difficult to produce a unified background mesh across the whole model.

The proposed surface mesher will require two parameters (both dimensionless) as input and will depend on a global 3D background mesh based on an octree.

2.1 Using Dimensionless Parameters

At each point on the CAD model's surfaces, a required mesh size should be defined. That size should be approximately equal to the size of the surface mesh edges at that location. Any triangle created during the meshing process should fulfill two criteria

1. The distance of the center of the triangle should be smaller than a prescribed distance, which will depend on a dimensionless parameter.
2. The allowed size difference between two neighboring triangles should be smaller than a value, which will also be dimensionless.

First, regarding the size, a parameter α, calculated in degrees or radians, is chosen to be used.

If a point on a curve is taken, a curvature radius K_r can be defined locally. A mesh size can then be calculated to be equal to the distance between the starting and ending points of a circular arc of radius K_r and angle α. Therefore, the higher the curvature, the smaller the size. This idea can easily be used for surfaces as well, if, instead of circles, spheres are used as a measure.

Second, regarding the difference between two neighboring triangles, a growth ratio is defined as

$$GR = \frac{The\ maximum\ allowed\ change\ in\ the\ size\ of\ two\ triangles}{The\ distance\ between\ them} \cdot \frac{m}{m}. \quad (5)$$

Examples of the effects of the deflection and growth ratio parameters can be seen in Figs. 4 and 5, respectively.

2.2 Using an Octree Mesh as a Background Mesh

The BRep format, as stated above, can have certain discontinuities due to portability. If that is the case, the mesh size of a face will not be propagated to its neighbors, which can cause discontinuities in the surface mesh. To overcome this challenge, a global background mesh is generated in the form of an Octree mesh [13]. The mesh size is stored at the center of each leaf. The sizes of the leaves have to be roughly equal to the mesh size they store and the sizes of neighbor leaves have to respect the prescribed growth ratio. The algorithm goes as follows:

1. Define the axis-aligned bounding box of the model to be meshed.
2. Sample the points on the edges and faces of the model.
3. Take a point and calculate the mesh size.
4. Find the leaf that the point belongs to, and if its size is bigger than the mesh size, divide it and find the new leaf it belongs to.
5. Repeat 4 until the leaf's size is smaller or equal to the mesh size.
6. Store the mesh size at its leaf. If the growth ratio condition is not satisfied, update the mesh sizes of its neighbors, so that the smaller size is always respected.
7. Repeat 3 until all points have been used.
8. Balance the Octree mesh [14].

An example octree mesh for a cylinder is shown in Fig. 3, and the resulting triangular mesh is shown in Fig. 6.

2.3 Using the Advancing Front Method for Meshing the Surfaces

Once a proper and consistent background mesh is created and the mesh size is calculated and stored, the algorithm can proceed with the actual surface meshing. The Advancing Front algorithm [15–17] is chosen, because of the high quality meshes it can produce.

The algorithm starts by meshing the model face by face. For each face, its bounding edges are first meshed to create the initial front. Then, the triangulation of the face is done in its 2D parametric space, to ensure convergence and to enable easier quality checks.

In order to improve quality even more, Laplacian smoothing [18] and edge flipping [19] are performed.

The algorithm was tested in the Drivaer model, and the results are shown in Figs. 7 and 8.

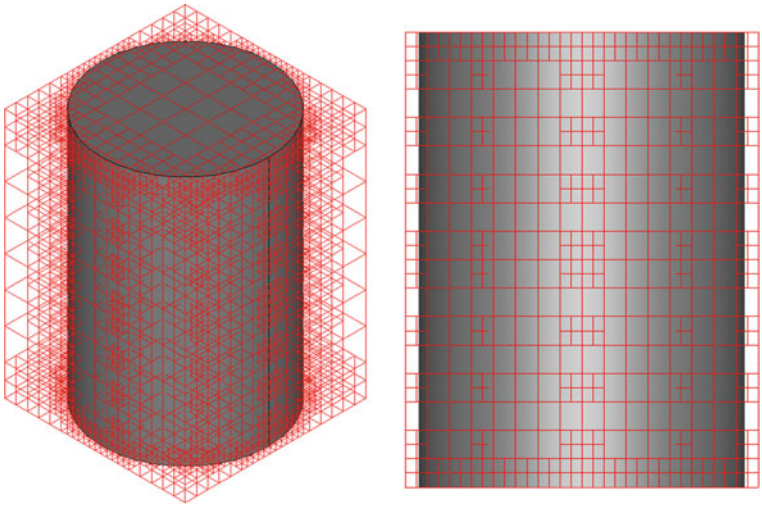


Fig. 3 The Octree background mesh around a cylinder with $\alpha = 10°$, $GR = 0.1$ m/m

3 Changing the Shape of BRep Models

In order to insert CAD inside the optimization loop, there has to be access to its parameterization. This must be done through the BRep format and its differentiated version. The alternative would be to have access to differentiated versions of various (commercial) CAD packages.

The independent differentiation with respect to the geometric parameters (i.e., control points, weights) of each face and edge, is not an option, as, after an optimization cycle, the continuity of the model would not be respected. Thus, a more general way to change the shape of the BRep model has to be found.

Geometric Morphing is a method that combines the morphing of surface meshes and Reverse Engineering [20, 21]. It starts by morphing a surface using discrete points on it, and then it attempts to fit these displaced points using the same surface that described them initially.

Therefore, a surface mesh is generated on the faces and edges of the BRep model. While it is generated, the mapping between each point and its surface parameters (u, v) is kept. Then, the mesh is displaced using a Free Form Deformation Technique [22] and a new surface mesh is acquired. Finally, that newly acquired mesh is fitted by changing the initial BRep geometry.

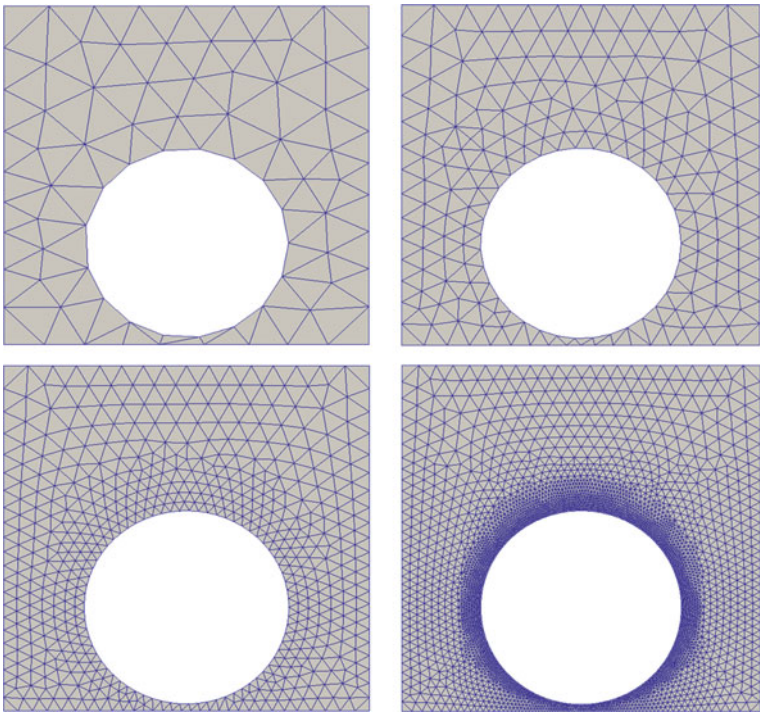


Fig. 4 Meshing of a single-face BRep model, under different angular distances. Top Left: $\alpha = 20°$, $GR = 0.1$ m/m. Top Right: $\alpha = 10°$, $GR = 0.1$ m/m. Bottom Left: $\alpha = 5°$, $GR = 0.1$ m/m. Bottom Right: $\alpha = 1°$, $GR = 0.1$ m/m

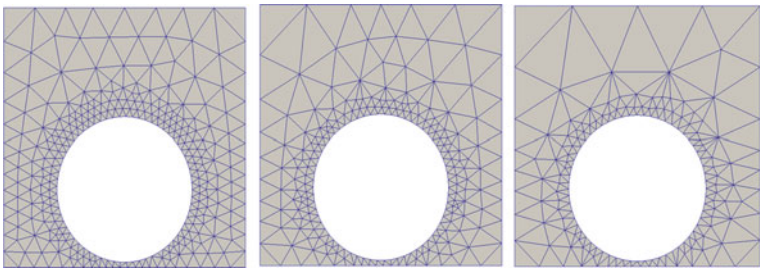


Fig. 5 Meshing of a single-face BRep model, under different growth ratios. Left: $\alpha = 5°$, $GR = 0.25$ m/m. Middle: $\alpha = 5°$, $GR = 0.4$ m/m. Right: $\alpha = 5°$, $GR = 0.5$ m/m

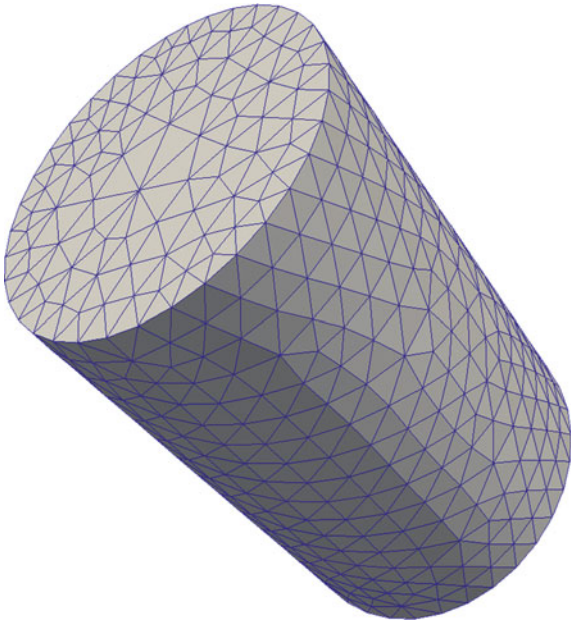


Fig. 6 The resulting surface mesh for the cylinder and its background mesh shown in Fig. 3

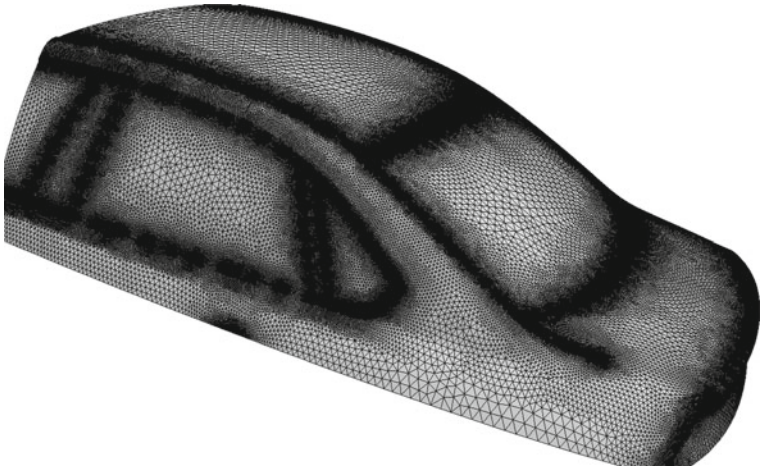


Fig. 7 Meshing of the DrivAER model shown in Fig. 1

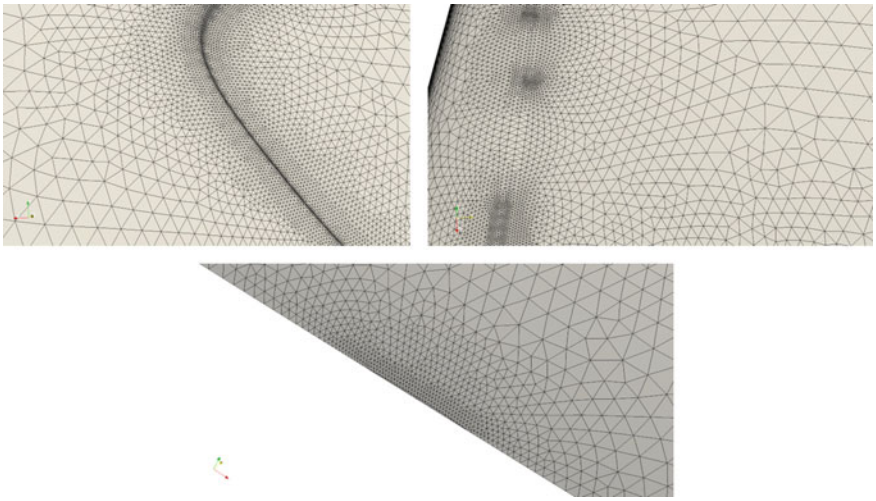


Fig. 8 Meshed areas of the DrivAER model of Fig. 7. Top Left: Back right window. Top Right: Roof. Bottom: Left back door handle

3.1 Adjoint-Based Optimization and the Continuous Adjoint Technique

During recent years, in CFD gradient-based optimization, the Adjoint Technique [23–25] has received much attention, due to the fact that the cost of computing sensitivity derivatives of an objective function J is independent of the number of design variables. Therefore, the Adjoint technique, in its discrete [26, 27] or continuous [23, 28–32] form, is excellent for large-scale optimization problems. In this article, the continuous Adjoint formulation is used to calculate the sensitivity derivatives.

3.1.1 Primal Equations

The primal problem governed by the incompressible Reynolds-averaged Navier–Stokes equations can be written as

$$R^p = -\frac{\partial u_i}{\partial x_i} = 0, \tag{6}$$

$$R_i^u = u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}\left[(\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] = 0. \tag{7}$$

In Eqs. 6 and 7, u denotes the components of the primal velocity and p is the primal pressure.

3.1.2 Adjoint Equations

Let J be a function to be minimized by the computation of an optimal set of design variables $b_n, n \in [1, N]$. The starting point of the continuous Adjoint formulation is the formulation of the augmented objective function J_{aug}. Assuming a computational domain Ω and its boundary S,

$$J_{aug} = J + \int_{\Omega} vi R_i^u d\Omega + \int_{\Omega} q R^p d\Omega. \tag{8}$$

In Eq. 8 v_i is the i-th component of the adjoint velocity and q is the adjoint pressure. It is obvious, since the primal state equations must hold, that $J_{aug} = J$. Minimization of J, therefore becomes minimization of J_{aug}.

$$\frac{\delta J_{aug}}{\delta b_n} = \frac{\delta J}{\delta b_n} + \int_{\Omega} vi \frac{\partial R_i^u}{\partial b_n} d\Omega + \int_{\Omega} q \frac{\partial R^p}{\partial b_n} d\Omega + \int_{S} (vi R_i^u + q R^p) n_k \frac{\delta x_k}{\delta b_n} dS. \tag{9}$$

In Eq. 9, two differential operators can be seen $\delta()/\delta b_n$ and $\partial()/\partial b_n$. The first operator denotes the total derivative and the second denotes the partial derivative. For a given quantity Φ, these operators are connected through

$$\frac{\delta \Phi}{\delta b_n} = \frac{\partial \Phi}{\partial b_n} + \frac{\partial \Phi}{\partial x_k} \frac{\delta x_k}{\delta b_n}. \tag{10}$$

The field Adjoint equations are then formulated so as to make Eq. 9 independent of variations in the primal state variables. These are written as

$$R^q = -\frac{\partial v_j}{\partial x_j} = 0, \tag{11}$$

$$R_i^v = v_j \frac{\partial u_j}{\partial x_i} + \frac{\partial q}{\partial x_i} - \frac{\partial (u_j v_i)}{\partial x_j} - \frac{\partial}{\partial x_j}\left[(\nu + \nu_t)\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right] = 0. \tag{12}$$

The time required to solve the Adjoint equations is equivalent to the time required to solve the primal problem. This makes apparent the strength of the Adjoint technique: the time required for the sensitivity calculation is always two equivalent flow solutions. The formula for the calculation of the sensitivity derivatives is omitted for the sake of space. The presentation of the adjoint formulation of turbulence models is omitted as well, for the sake of simplicity.

3.2 Volumetric NURBS Free Form Deformation

In Sect. 1.2, the basics of Uni-variate and Bi-variate NURBS were shown. Here, Tri-variate NURBS will be used in a similar way. The formula that gives us a point inside a NURBS volume is

$$\mathbf{V}(u,v,w) = \frac{\sum_{i=1}^{nCp}\sum_{j=1}^{mCp}\sum_{k=1}^{lcp} N_i^p(u)N_j^q(v)N_k^r(w)\cdot w_{i,j,k}\mathbf{P}_{i,j,k}}{\sum_{i=1}^{nCp}\sum_{j=1}^{mCp}\sum_{k=1}^{lcp} N_i^p(u)N_j^q(v)N_k^r(w)\cdot w_{i,j,k}}, \tag{13}$$

which is the formula of surface NURBS, extended to three parametric directions.

The bounds of the control box are usually set to the bounding box. Then, any point inside the bounding box can be mapped from x, y, z coordinates to u, v, w coordinates using a point-inversion technique. The method proposed in [33] is suggested because of its robustness. The point-inversion ensures that any point inside the control box can be mapped and reproduced by the Volumetric NURBS with machine accuracy.

Assuming that an optimization is performed, after solving the primal and adjoint CFD equations, a sensitivity vector is acquired. Let us denote as J an objective function to be minimized that depends on the shape of an aerodynamic object. Furthermore, let $\mathbf{X}_{pi}$ with $pi \in [1, n]$ be the boundary mesh point coordinates. Each $\mathbf{X}_{pi}$ can be mapped to parametric coordinates (u_{pi}, v_{pi}, w_{pi}) such that $\mathbf{V}(u_{pi}, v_{pi}, w_{pi}) = \mathbf{X}_{pi}$. The n sensitivity vectors are going to be of the form $\frac{dJ}{d\mathbf{X}_{pi}}$.

Finally, the derivatives with respect to the control box points, at a specific set of parameters, are

$$\frac{d\mathbf{V}(u_{pi}, v_{pi}, w_{pi})}{d\mathbf{P}_{i,j,k}} = \frac{N_i^p(u_{pi})N_j^q(v_{pi})N_k^r(w_{pi})\cdot w_{i,j,k}\mathbf{P}_{i,j,k}}{\sum_{i=1}^{nCp}\sum_{j=1}^{mCp}\sum_{k=1}^{lcp} N_i^p(u_{pi})N_j^q(v_{pi})N_k^r(w_{pi})\cdot w_{i,j,k}}. \tag{14}$$

By the chain rule, this leads to the following:

$$\frac{dJ}{d\mathbf{P}_{i,j,k}} = \sum_{pi=1}^{n}\frac{dJ}{d\mathbf{X}_{pi}}\cdot\frac{N_i^p(u_{pi})N_j^q(v_{pi})N_k^r(w_{pi})\cdot w_{i,j,k}}{\sum_{i=1}^{nCp}\sum_{j=1}^{mCp}\sum_{k=1}^{lcp} N_i^p(u_{pi})N_j^q(v_{pi})N_k^r(w_{pi})\cdot w_{i,j,k}}. \tag{15}$$

Equation 15 is used to transform the sensitivity vectors to control box sensitivities. An optimizer can then be employed, which will relocate the control box.

3.3 Fitting the Displaced Surface Mesh

After relocating the control box by moving its control points, every surface mesh point previously mapped inside the box will now be relocated as well. Therefore, the

final task is to change the control point positions of the surface and curve NURBS that make the BRep model best fit the relocated points. This is done using linear regression [34]. For NURBS curves, Eq. 1 can be written as

$$\mathbf{C}(u) = \sum_{i=1}^{nCp} R_i(u) \cdot \mathbf{P}_i \quad \text{with} \quad R_i(u) = \frac{N_i^p(u) \cdot w_i}{\sum_{i=1}^{nCp} N_i^p(u) \cdot w_i}. \tag{16}$$

For multiple grid points $\mathbf{X}_{pi}$, $pi \in [1, n]$, Eq. 16 can be written multiple times and set equal to the grid points

$$\begin{aligned}
\mathbf{C}(u_1) &= \sum_{i=1}^{nCp} R_i(u_1) \cdot \mathbf{P}_i = \mathbf{X}_1 \\
\mathbf{C}(u_2) &= \sum_{i=1}^{nCp} R_i(u_2) \cdot \mathbf{P}_i = \mathbf{X}_2 \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
\mathbf{C}(u_n) &= \sum_{i=1}^{nCp} R_i(u_n) \cdot \mathbf{P}_i = \mathbf{X}_n,
\end{aligned}$$

or in matrix form

$$\begin{bmatrix} R_1(u_1) & R_2(u_1) & \dots & R_{nCp}(u_1) \\ R_1(u_2) & R_2(u_2) & \dots & R_{nCp}(u_2) \\ \cdots & \cdots & \cdots & \cdots \\ R_1(u_n) & R_2(u_n) & \dots & R_{nCp}(u_n) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_{nCp} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_n \end{bmatrix} \tag{17}$$

If we set the right-hand side of Eq. 17 as equal to the displaced surface points and then solve the over-determined system, for each edge, the control points acquired will give the best fit to the displaced grid points in the least square sense. The above methodology is applied to all edges and surfaces for which it can be easily generalized. The CAD model will follow the change of a control box as accurately as possible, while still being described by the same NURBS geometry. An application is shown in Fig. 9.

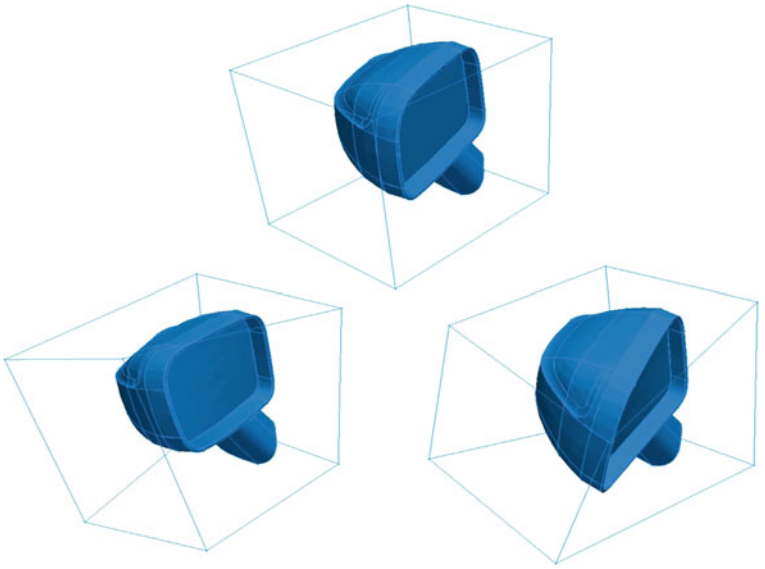


Fig. 9 Morphing of a side mirror, courtesy of Trabant NT. Top: The initial geometry of the mirror, along with the initial control box. Bottom: The mirror deformations, after its control box has been subjected to two different twists

4 Conclusions

In this study, methods for connecting CFD and the Adjoint framework to the CAD framework were shown.

Initially, a surface triangulator was built on BRep models. The triangulator can overcome BRep discontinuities, as it depends on a global background mesh (Octree) and is controlled by parameters that are dimensionless. Defaults can be easily located for these parameters, and ultimately, a scale-invariant mesh will be created, depending only on curvature. Thanks to the use of the Advancing Front method, the resulting mesh will be of high quality and suitable for use as a basis for volume meshing.

Second, the Geometric Morphing Technique was shown, combining the aspects of Free Form Deformation and Reverse Engineering. By performing an optimization using this technique, the resulting model keeps its initial geometric properties (basis degree, number of control points). Furthermore, because the model is described by NURBS, it can be subjected to fine tuning and be manufacturable. Moreover, no post-processing tool is required to reexport it to a CAD software.

Both methods were designed and programmed using a hybrid framework between OCCT and OpenFOAM®, ensuring the open source nature and the portability of the final applications.

Acknowledgements The work shown here is part of the IODA (Industrial Optimal Design using Adjoint CFD) Project. Research topic: *Intuitive interfaces for optimisation parameterisation, constraint definition and automated mesh-to-CAD conversion.*
The project leading to this application, has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 642959.

References

1. Han, J., Requicha, A.: Feature recognition from CAD models. IEEE Comput. Graph., **18(2)**, 80–94 (1998)
2. Shah, J. J., Anderson, D., Kim, Y. S., Joshi, S.: A Discourse on Geometric Feature Recognition From CAD Models. J. Comput. Inf. Sci. Eng., **1(1)**, 41 (2001)
3. Subrahmanyam, S., Wozny, M.: An overview of automatic feature recognition techniques for computer-aided process planning. Comput. Ind., **26(1)**, 1–21 (1995)
4. Nowacki, H., Dannenberg, L.: Approximation Methods Used in the Exchange of Geometric Information via the VDA/VDMA Surface Interface. Product Data Interfaces in CAD/CAM Applications, 150–159 (1986)
5. https://www.opencascade.com/
6. Mantyla, M. (1988). An introduction to solid modeling. Rockville, MD: Computer Science Press.
7. Stroud, I. (2006). Boundary representation modelling techniques. London: Springer.
8. TUM Department of Mechanical Engineering, Chair of Aerodynamics and Fluid mechanics. Retrieved from: www.aer.mw.tum.de/en/research-groups/automotive/drivaer
9. Piegl, L., Tiller, W. (1997). The NURBS book. Berlin: Springer.
10. Papadimitriou, D., Giannakoglou, K.: A continuous adjoint method with objective function derivatives based on boundary integrals, for inviscid and viscous flows. Comput. Fluids, **36(2)**, 325–341 (2007)
11. Papoutsis-Kiachagias, E., Zymaris, A., Kavvadias, I., Papadimitriou, D., Giannakoglou, K.: The continuous adjoint approach to the k turbulence model for shape optimization and optimal active control of turbulent flows. Eng. Optimiz., **47(3)**, 370–389 (2014)
12. Schberl, J.: NETGEN An advancing front 2D/3D-mesh generator based on abstract rules. Computing and Visualization in Science, **1(1)** 41–52 (1997)
13. Meagher, D.: Geometric modeling using octree encoding. Comput. Vision Graph, **19(2)**, 129–147 (1982)
14. Isaac, T., Burstedde, C., Ghattas, O. (2012) : Low-Cost Parallel Algorithms for 2:1 Octree Balance. 2012 IEEE 26th International Parallel and Distributed Processing Symposium
15. Frey, P. J., George, P. L. (2008). Mesh generation: Application to finite elements. London: ISTE.
16. Mavriplis, D. J.: An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness. J. Comput. Phys., **117(1)**, 90–101 (1995)
17. Tristano, J. R., Owen, S. J., Canann, S. A.: Advancing Front Surface Mesh Generation in Parametric Space Using a Riemannian Surface Definition (1998), https://doi.org/10.1.1.36.630
18. Herrmann, L. R.: Laplacian-isoparametric grid generation scheme. J. Eng. Mech. Div-ASCE, **102(5)**, 749–756 (1976)
19. Cheng, S., Jin, J.: Edge Flips in Surface Meshes. Discrete Comput. Geom., **54(1)**, 110–151 (2015)
20. Piegl, L., Tiller, W.: Parametrization for surface fitting in reverse engineering. Comput. Aided Design, **33(8)**, 593–603 (2001)

21. Varady, T., Martin, R. R., Cox, J.: Reverse engineering of geometric modelsan introduction. Comput. Aided Design, **29(4)**, 255–268 (1997)
22. Sederberg, T. W., Parry, S. R.: Free-form deformation of solid geometric models. Comp. Graph., **20(4)**, 151–160 (1986)
23. Jameson, A.: Aerodynamic design via control theory. J. Sci. Comput. **3(3)**, 233–260 September (1988).
24. Pironneau, O.: On optimum design in fluid mechanics. Journal of Fluid Mechanics **64(1)**, 97–110 (1974).
25. Thvenin, D. and Janiga, G.: Optimization and Computational Fluid Dynamics. Springer Publishing Company, Incorporated, 1st edition, (2008)
26. Giles, M. B., Duta, M. C., Mller, J.-D., and Pierce, N. A.: Algorithm developments for discrete adjoint methods, (2001).
27. Vishnampet, R., Bodony, D. J., and Freund, J. B.: A practical discrete-adjoint method for high-fidelity compressible turbulence simulations. J. Comput. Phys. 285(C), 173–192 March (2015).
28. Giannakoglou, K. C. and Papadimitriou, D. I.: Adjoint Methods for Shape Optimization, 79–108. Springer Berlin Heidelberg, Berlin, Heidelberg (2008).
29. Giannakoglou, K. C., Papadimitriou, D. I., Papoutsis-Kiachagias, E. M., and Kavvadias, I. S.: Aerodynamic Shape Optimization Using Turbulent Adjoint And Robust Design in Fluid Mechanics, 289–309. Springer International Publishing, Cham (2015).
30. Othmer, C.: Implementation of a continuous adjoint for topology optimization of ducted flows, 2007 (2007).
31. Othmer, C.: A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. International Journal for Numerical Methods in Fluids **58(8)**, 861–877 (2008).
32. Papoutsis-Kiachagias, E., Kyriacou, S., and Giannakoglou, K.: The continuous adjoint method for the design of hydraulic turbomachines. Computer Methods in Applied Mechanics and Engineering 278(Complete), 621–639 (2014).
33. Martin, M. J., Andres, E., Lozano, C., Valero, E.: Volumetric b-splines shape parametrization for aerodynamic shape design. Aerosp. Sci. Techno., **37**, 26–36 (2014)
34. Yan, X., Su, X. G.: Linear Regression Analysis (2009), https://doi.org/10.1142/6986

Cavitating Flow in a 3D Globe Valve

Daniel Rodriguez Calvete and Anne Gosset

Abstract The efficiency of control valves operating with liquids is highly conditioned by the occurrence of cavitation when they undergo large pressure drops. For severe service control valves, the subsequent modification of their performance can be crucial for the safety of an installation. In this work, the solver interPhaseChangeFoam, implemented in OF v2.3, is used to characterize the flow in a globe valve, with the objective to evaluate its capability in solving cavitating flows in complex 3D geometries. An Homogeneous Equilibrium approach is adopted, and phase change is modeled using the Schnerr and Sauer cavitation model. Confrontation with experimental data is carried out in order to validate the numerical results. It is found that the solver predicts correctly the location of vapor cavities, but tends to underestimate their extension. The flow rate is correctly calculated, but in strong cavitating regimes, it is affected by the underprediction of vapor cavities. The force acting on the stem is found to be more sensitive to the computation parameters.

1 Introduction

In nuclear power plants and petrochemical installations, certain specific control valves play a critical role in the functioning of the plants. Therefore, these severe service valves have to be responsive, precise and perfectly reliable. The efficiency of control valves operating with liquids is highly conditioned by the occurrence of cavitation when they undergo large pressure drops. In the vena contracta that develops in the restriction region, the fluid is accelerated such that the local pressure may decrease below the vapor pressure and generate cavitation. When cavitation is fully developed, it can modify the performance of the valve and even limit the flow rate close to chocked flow conditions. In practice, the occurrence of cavitation is difficult

D. R. Calvete (✉) · A. Gosset (✉)
Grupo Integrado de Ingenieria da Universidad da Coruña,
C/Mendizabal s/n Ferrol, A Coruña, Spain
e-mail: daniel.rcalvete@udc.es

A. Gosset
e-mail: agosset@udc.es

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_3

to detect, because of the harsh and noisy conditions under which the valves operate. This is why CFD has become an important tool for the design and characterization of severe service control valves. However, it has scarcely been used up to now for cavitating flows, due to the important computational times involved and the sensitivity of the solvers to empirical parameters. Chen and Stoffel [4] investigate the transient effects of cavitation in a poppet valve during closure, but their computational domain is axisymmetric and limited to the vicinity of the restriction region. Bernad et al. [1] extend the analysis of cavitation to a 3D poppet valve. Beune et al. [2] were the first to publish a validated CFD study with a safety relief valve. They show that taking into account phase change in CFX allows for reducing the overestimation of the mass flow rate in the valve and the force exerted by the fluid on the stem in a spectacular way. Couzinet et al. [5] also use a mixture model in CFX to predict the flow capacity of a safety relief valve in cavitating regimes, and they propose a correction of the liquid vapor pressure to take into account the effect of turbulence. The topology of the flow field is very well validated with experimental data obtained through Particle Image Velocimetry. Ferrari and Leutwyler [6] also propose a single phase numerical study of the flow in a globe valve with Fluent. But more importantly, they perform an extensive experimental study, gathering unsteady measurements of flow forces on the stem and flow visualizations on a transparent mock-up.

Up to now, all the numerical studies were conducted with commercial codes. In this paper, we propose evaluating the capabilities of the open source CFD code OpenFoam® (OF) in predicting the unsteady cavitating flow in a 3D globe valve geometry. For that, we validate the performances of the solver interPhaseChangeFoam using the experimental data of Ferrari and Leutwyler [6] to validate the results.

2 Numerical Approach

2.1 Governing Equations

To simulate cavitating flows, the two phases, liquid (l) and vapor (v), have to be taken into account in the governing equations, and the phase transition mechanism due to evaporation-condensation has to be modeled. In the interPhaseChangeFoam solver, implemented in OpenFoam® v2.3, the two phases are assumed to be homogeneously mixed and in mechanical equilibrium, following the Homogeneous Equilibrium approach. Hence, only one set of momentum and continuity equations is solved for the mixture. It is assumed that there is no interaction and no slip between vapor bubbles. The VOF (Volume of Fluid) technique is used to track the interface between liquid and vapor.

Since liquid and vapor are assumed to be perfectly mixed within each cell of the mesh, the density and viscosity of the mixture are expressed as a function of the liquid and vapor volume fractions, α_l and α_v, respectively

$$\rho = \alpha_l \rho_l + \alpha_v \rho_v \tag{1}$$

$$\mu = \alpha_l \mu_l + \alpha_v \mu_v \tag{2}$$

The subscripts l and v stand for the properties of pure liquid and pure vapor, respectively. The constraint condition to fullfill is

$$\alpha_l + \alpha_v = 1. \tag{3}$$

To close the system, a transport equation for α_l is needed. Using Eq. (1) in the continuity equation, and considering the mass transfer between phases due to cavitation, the transport equation for α_l can be derived as [11]

$$\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \mathbf{U}) + \nabla \cdot (\alpha_l (1 - \alpha_l) \mathbf{U}_r) = \frac{\dot{m}^- - \dot{m}^+}{\rho_l} \tag{4}$$

where $\mathbf{U}$ is the velocity vector, $\mathbf{U}_r$ is the relative velocity vector at the interface between the two phases, $\dot{m}^+$ is the mass transfer rate by vaporization and $\dot{m}^-$ the one by condensation.

2.2 Cavitation Model

The source term of mass transfer (RHS of Eq. 4) requires an appropriate cavitation model. Different cavitation models can be found in the literature. In this work, the cavitation model of Sauer and Schnerr [11] is chosen. This model considers an initial amount of micro-spherical vapor bubbles with a radius R, which constitute nucleation sites for cavitation. They grow and collapse according to the bubble pressure dynamics governed by the first-order Rayleigh Plesset equation.

In the model of Sauer and Schneer [11], the vapor fraction is calculated based on the volume of the spherical nuclei with radius R, and their number per cubic meter of liquid, n_0, as

$$\alpha_v = \alpha_l \cdot n_0 \frac{4}{3} \pi R^3 \tag{5}$$

The combination of Reyleigh Plesset equation and Eqs. (4) and (5) gives the final expression for the mass transfer source terms

$$\dot{m}^+ = C_v (1 - \alpha_v) \frac{3\alpha_v}{R} \frac{\rho_v \rho_l}{\rho} \sqrt{\frac{2}{3} \frac{|max(0, p_v - p)|}{\rho_l}} \tag{6}$$

$$\dot{m}^- = C_c (1 - \alpha_v) \frac{3\alpha_v}{R} \frac{\rho_v \rho_l}{\rho} \sqrt{\frac{2}{3} \frac{|max(0, p - p_v)|}{\rho_l}} \tag{7}$$

where R is derived from Eq. (5).

As inputs, the model therefore requires the volumetric concentration of nuclei n_0, their initial radius R, and the empirical coefficients C_v and C_c. The latter depend on the state of deaeration of the liquid and the mean flow. To quantify the influence of those parameters in the OpenFoam® solver, a sensitivity study was conducted for a sharp orifice by Gosset et al. [7], who concluded that there was no significant difference between results for $n_0 \leq 10^{10}\ \mathrm{m}^{-3}$ and $10^{-7} \leq R \leq 10^{-5}$ m. In this work, the value of the parameters are set at $R = 5 \times 10^{-6}$ m, $n_0 = 10^{14}\ \mathrm{m}^{-3}$, $C_v = 1$ and $C_c = 2$ ($C_c = 2$ because condensation is considered faster than vaporization). Note that C_v and C_c are set empirically, and the latter values are known as being valid for a wide range of flows.

Originally, in the model by Sauer and Schneer, the phase change threshold pressure (p_v) is assumed to be equal to the saturation pressure (p_{vap}) in the absence of dissolved gases. In this study, the value of p_{vap} is set at 3540 Pa, which is estimated with the Rankine formula for water at 26 °C. Nevertheless, several investigations have shown significant effects of turbulence on cavitating flows e.g., [9]. Several authors, including Singhal et al. [12] and Bouziad [3], have suggested taking it into account by integrating in time the contributions of $\dot{m}_v^+$ and $\dot{m}_c^-$ assuming a probability density function of the pressure fluctuations due to turbulence. Bouziad [3] proposes a simple approach based on a correction of p_v using the shear strain to modify the bubble pressure. The corrected value of p_v becomes

$$p_v = p_{vap} + (\mu + \mu_t)S \tag{8}$$

where S is the shear strain and μ_t is the turbulent viscosity.

Couzinet et al. [5] and Rodriguez Calvete et al. [10] evaluate the effect of using this approach in a safety relief valve and globe valve flows, respectively. They show that turbulence effects contribute to an increase of up to 500% of the vapor threshold pressure (p_v), which directly influences the location and extension of cavitation. It must be noted that this kind of approach is highly dependent on the quality of turbulence prediction, therefore an appropriate turbulence model should be chosen.

2.3 Turbulence Model

For this 3D pressure-driven flow, a URANS model for turbulence is adopted instead of Large Eddy Simulation, disregarded in this study due to its time cost. In this study, the $k - \omega - SST$ has been chosen for all the computations. Default parameters of the turbulence model were used. The Reynolds number varies between 1.81 and 3.3×10^5 for all of the cases in this study.

On the other hand, the strong pressure gradients expected in the flow make it necessary to solve the boundary layer accurately up to the viscous sublayer, especially close to the stagnation zone. This means that a high resolution mesh at the wall is

needed, so that the use of standard wall functions is avoided. $k-\omega-SST$ adapts to the local mesh density in the boundary layer region using the Spalding wall function (nutSpaldingWallFunction). The y^+ values range from 0.1 to 15 on the piston, with an average of 5.

2.4 Computational Domain

The computational domain is defined from the geometrical model sketched in Fig. 1 which matches the geometry of a 2″ commercial globe valve. It corresponds exactly to the mock-up built by Ferrari and Leutwyler [6] to visualize the flow inside the valve. Under cavitating conditions, Couzinet et al. [5] find that the flow is substantially asymmetric in a safety relief valve, which is why the whole domain is simulated here.

The mesh is designed with the meshing software Ansys-ICEM, using a fully structured topology and only hexahedral cells. In order to preserve low y^+ at the wall, a special refinement in the restriction is made, as shown in Fig. 2. A focus on the 6 mm lift of the stem is proposed in this study. A mesh independence study is performed using three different grid sizes. Since cavitation develops not only in the restriction region, but also downstream of the valve body, in the outlet duct, special attention is given to the refinement of this region (Fig. 2). The results in terms of volumetric flow rate (Q), transversal and axial force acting on the stem (F_{trans} and F_{axial}) are presented in Table 1 for the different meshes. Slight differences are found between the coarse and the medium mesh, which become negligible between the medium and the fine mesh. In addition, the extension of vapor cavities is very similar for the two finest grids. Therefore, the medium mesh of 1.65 million cells is chosen as the base mesh.

For the boundary conditions, a total absolute pressure is fixed at the inlet ($p_{t,in}$) and a static pressure is fixed at the outlet (p_{out}) (Fig. 1). For velocity, corresponding Neuman conditions are set (i.e., `pressureVelocityInlet` at the inlet and `zeroGradient` at the outlet).

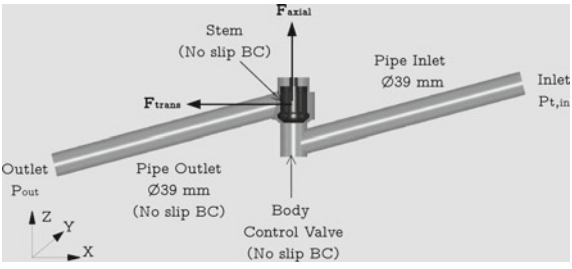


Fig. 1 Valve model and computational domain

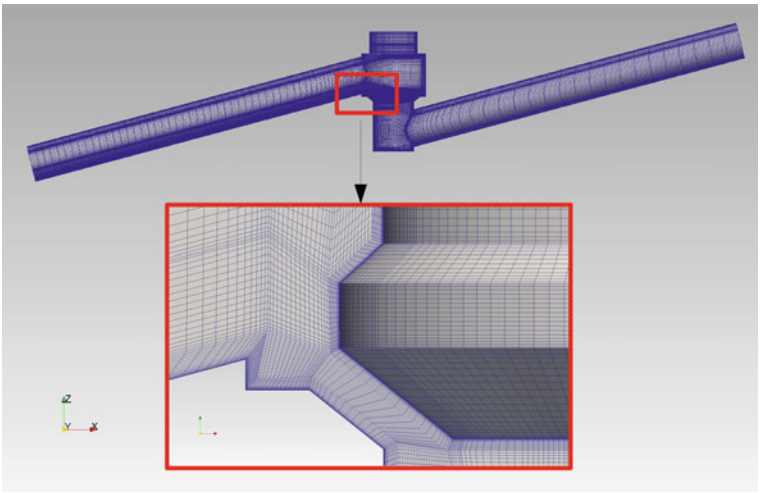


Fig. 2 Mesh restriction detail

Table 1 Results of the mesh sensitivity study

Cells $\times 10^6$	Δp (bar)	Q (m^3/h)	F_{axial} (N)	F_{trans} (N)
1.55	1.810	32.28	108.8	69.39
1.65	1.810	32.53	103.6	71.33
1.96	1.803	32.32	103.2	71.27

2.5 Numerical Methodology

The algorithm PIMPLE, which combines both SIMPLE and PISO for unsteady simulations, is used in order to keep the computational time within reasonable limits. It should be underlined that even if the time step can be set to guarantee the stability of the solution, the phase change process due to cavitation is very fast, and the time step must be sufficiently small so as to capture the relevant phenomena and control the non linearities generated by the mass transfer term.

Regarding time schemes, the second-order backward and Crank Nicholson schemes both result in numerical instabilities and divergence of the solution, even with $C_{o,max} < 1$. Therefore, a first order Euler implicit scheme is adopted. An adaptive time step based on a maximum Courant number, $C_{o,max} = 3$, is set. For the spatial discretization, second-order bounded schemes (`limitedLinear 1.0`) are used for the divergence terms related to U, k and ω . For divergence of alpha `div(phi,alpha)` and for the compression of the interface `div(phirb, alpha)`, the vanLeer and interfaceCompression schemes are respectively chosen. Regarding gradient schemes, `cellMDLimited Gauss linear 0.777` is set as default. To avoid unboundedness issues in k and ω, the `edgeCellsLeast`

`Squares 1.0` scheme is set. For Laplacian schemes, `Gauss linear limited 0.333` is used to overcome non-orthogonal features, which are difficult to avoid in such a complex 3D geometry.

3 Results

3.1 Operating Conditions

The numerical study is based on the experimental conditions described in [6]. The authors run experiments for a set of cavitating conditions, varying the pressure drop across the valve Δp and the valve opening (lift). To relate the pressure drop with the intensity of cavitation, the cavitation number σ is defined as

$$\sigma = \frac{p_{in} - p_{out}}{p_{in} - p_{vap}} \tag{9}$$

The simulations are performed for valve openings of 6 and 4 mm. Table 2 reports the conditions corresponding to the different computations.

The outlet pressure p_{out} is set at values of 0.4 and $0.8bar$, in order to induce earlier cavitation. Cases Dp07 and Dp08, with $p_{out} = 0.8$ bar, correspond to non-cavitating conditions, in agreement with the experimental observations. In contrast, cases Dp15-18-23, with $p_{out} = 0.4$ bar, correspond to fully developed cavitating conditions, as shown in Fig. 3.

Table 2 Experimental and numerical conditions

Cases		Δp (bar)	$p_{t,in}$ (bar)	p_{in} (bar)	p_{out}	Q (m^3/h)	σ (−)
Dp07	Exp	0.67	1.62	1.47	0.8	20.98	0.47
	OF	0.71	1.62	1.51	0.8	19.95	0.48
Dp08	Exp	0.83	1.74	1.63	0.8	22.90	0.52
	OF	0.83	1.74	1.63	0.8	21.90	0.52
Dp15	Exp	1.49	2.10	1.89	0.4	28.61	0.80
	OF	1.46	2.10	1.86	0.4	29.25	0.80
Dp18	Exp	1.74	2.50	2.13	0.4	30.78	0.83
	OF	1.81	2.50	2.21	0.4	32.53	0.83
Dp23	Exp	2.33	3.00	2.72	0.4	34.71	0.86
	OF	2.22	3.00	2.62	0.4	36.24	0.86

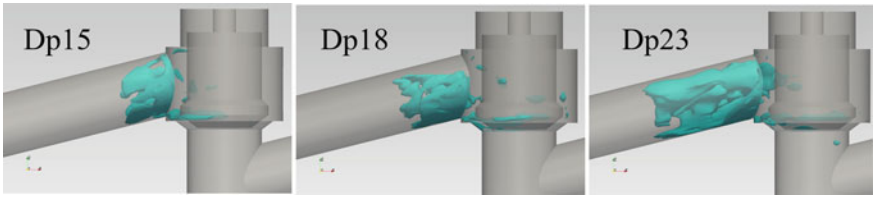


Fig. 3 Isosurface $\alpha_v = 0.5$ at cases $\Delta p = 1.5$ bar ($\sigma = 0.80$) , $\Delta p = 1.8$ bar ($\sigma = 0.83$) and $\Delta p = 2.3$ bar ($\sigma = 0.86$)

3.2 *Influence of Turbulence on p_v*

As mentioned above, the correction of p_v (8) accounts for the influence of turbulence on cavitation. Figure 4 shows the ratio of the corrected pressure threshold (P_v) and the saturation pressure (P_{vap}), which confirms how the correction affects the areas where cavitation is developed. Therefore, this correction is implemented into the interPhaseChangeFoam solver to account for turbulence effects on the prediction of cavitation.

3.3 *Flow Topology*

Figure 3 shows the extension of vapor cavities at 3 different pressure drops. It can be seen how an increase of σ from 0.80 ($\Delta p = 0.71$ bar) to 0.86 ($\Delta p = 2.22$ bar) leads to an increase of the vapor extension towards the valve outlet.

Finally, Fig. 5 illustrates the unsteady behavior of cavitation by comparing the experimental observations with the predictions of OpenFoam®, both at $\Delta p = 1.54$ bar (lift 4 mm). Two isosurfaces, at $\alpha_v = 0.5$ and $\alpha_v = 0.1$, are superimposed to

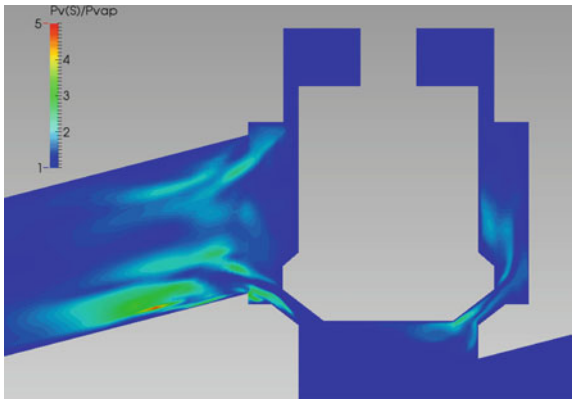


Fig. 4 Vapor phase iso-surface ($\alpha_v = 0.5$) with and without p_v correction in OpenFoam®

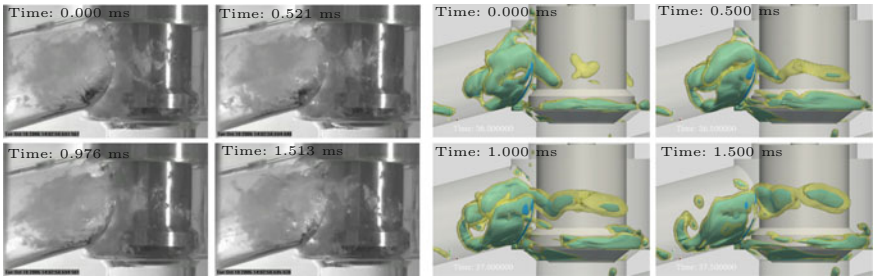


Fig. 5 Left: Experimental high-speed camera visualization (Lift 4 mm, Δp = 1.54 bar, Q = 23 m^3/h); Right: Numerical cavitation sequences of isosurfaces $\alpha_v = 0.5$ (Blue), $\alpha_v = 0.1$ (Yellow), with OpenFoam® (Lift 4 mm, Δp = 1.54 bar, Q = 26.4 m^3/h)

illustrate the extent of vapor–liquid mixing. Experimentally, the sequence of images is captured by a high-speed camera at a frequency of 13 KHz, while the vapor fraction fields are sampled at 2 KHz. The sequences are synchronized in time to compare the behavior of vapor cavities qualitatively. In this close-up, it can be clearly seen how the vapor cavities grow in the restriction region around the stem, and extend downstream the valve body in both cases. The location of vapor cavities is well predicted, and a certain synchronization of the bubble growth and collapse can be seen as well.

3.4 Flow Curve

For single phase turbulent flows through control valves, the flow rate Q in m^3/h is a linear function of the square root of the pressure drop Δp in bar across the valve [8]. This relation is expressed as

$$Q = K_v \sqrt{\Delta p} \tag{10}$$

where K_v is the flow coefficient, and the slope of the characteristic curve of the valve. In fact, K_v represents the volumetric flow rate of water circulating in a valve under a 1 bar pressure drop, at a given valve aperture. Under cavitating conditions, the characteristic flow curve deviates from this linear behavior, until it reaches the chocked flow condition. The latter occurs when Q no longer varies with Δp due to a vapor blockage at the valve outlet.

The volumetric flow rates obtained numerically are compared to the experimental data [6] in Fig. 6. The prediction by OpenFoam® presents a good agreement with the experiments, with an error of 1.5% on the prediction of K_v. However, it can be noticed that at the highest pressure drops, the flow rates predicted numerically remain on a linear curve, while the experimental values deviate slightly from their linear evolution. This indicates that the solver tends to underestimate the influence of vapor cavities on the head loss across the valve.

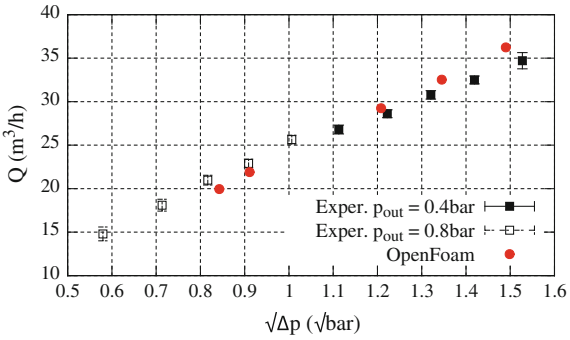
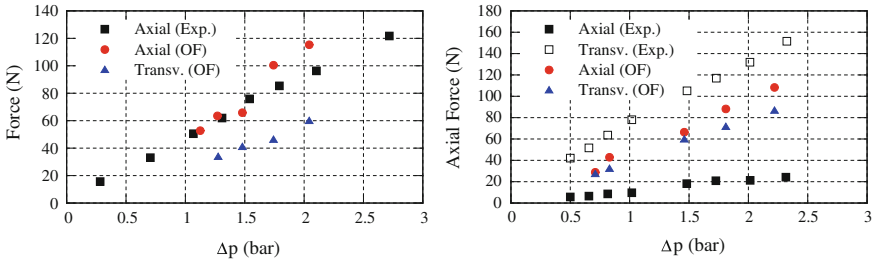


Fig. 6 Characteristic flow curve for a valve opening of 6 mm

3.5 *Forces on the Stem*

The experimental study [6] focuses on the components of the force acting on the stem. Figure 7a, b compare the axial and transversal force components found experimentally and numerically, at 4 and 6 mm lift, respectively. It can be seen that the code strongly underestimates the transversal forces and overestimates the axial forces at 6 mm lift (error up to 350%), while at 4 mm lift the error on the axial forces is less than 20%. Unfortunately, the experimental data of transversal forces at 4 mm are not available. This behavior, where the axial force is dominant, normally corresponds to small valve apertures [6]. It is at larger apertures that the force ratio is inversed, possibly due to the acceleration of the fluid under the disk, which generates lower pressures. According to experimental results [6], transversal forces are already dominant at a valve opening of 6 mm. Probably, the transition between the two behaviors (axial/transversal dominant force) is located within this range of lift (4–6 mm). Axial force measurements are also known to be affected by the friction of the stem in its seating, which may cause a bias in the measurement of the hydrodynamic force.



(a) At 4mm lift.

(b) At 6mm lift.

Fig. 7 Forces on the stem

4 Conclusions

In this work, the interPhaseChangeFoam solver is assessed for the simulation of a 3D cavitating flow in a globe valve that was characterized experimentally. The code succeeds reasonably well in predicting the flow rate through the valve, but it tends to underestimate the mass transfer by cavitation, which explains why the quality of the predictions decreases slightly under conditions of fully developed cavitation. Qualitatively, OpenFoam® correctly calculates the location of cavitation, and the highly unsteady behavior of cavities is well reproduced. Regarding the force exerted by the fluid on the stem, at 6 mm lift, it predicts an axial force larger than the transversal one, in contrast with the experiments, possibly due to measurement errors. A similar behavior was found with a coupled solver in Ansys-CFX [10]. The simulations at 4mm lift are in much better agreement with the experiments in terms of axial force prediction. Probably, the transition between the two behaviors (axial/transversal dominant force) is located within this range of lift (4–6 mm). The influence of the treatment of the vapor phase is also under investigation, to try to improve the prediction of vapor extension.

Acknowledgements The authors wish to thank Jérôme Ferrari from EDF R&D for providing us with additional experimental data. They also acknowledge the financial support of Xunta de Galicia through grant EM2013-009.

References

1. Bernad SI, Susan-Resiga R (2012) Numerical model for cavitational flow in hydraulic poppet valves. Modelling and Simulation in Engineering 2012:742,162
2. Beune A, Kuerten J, Schmidt J (2011) Numerical calculation and experimental validation of safety valve flows at pressures up to 600 bar. AIChE Journal 57(12):3285–3298
3. Bouziad Y (2006) Physical modelling of leading edge cavitation: computational methodologies and application to hydraulic machinery. PhD thesis, Ecole Polytechnique Fédérale de Lausanne
4. Chen Q, Stoffel B (2004) Cfd simulation of a hydraulic conical valve with cavitation and poppet movement. In: 4th Int. Fluid Power Conf., pp 331–339
5. Couzinet A, Gros L, Pinho J, Chabane S, Pierrat D (2014) Numerical modeling of turbulent cavitation flows in safety relief valves. In: Proc. 2014 ASME Pressure Vessels and Piping Conference. Anaheim (USA), p 28384
6. Ferrari J, Leutwyler Z (2008) Fluid flow force measurement under various cavitation state on a globe valve model. In: Proc. 2008 ASME Pressure Vessels and Piping Conference. Chicago (IL, USA), ASME, pp 157–165
7. Gosset A, Lamas Galdo I, Lema Rodrguez M (2015) Cavitating flow in a globe valve: Calibration of mass transfer models and simulations with OpenFOAM®. In: Proc. of Workshop on Safety and Control Valves
8. IEC (2007) EN-60534-2-1: Industrial process control valves. Flow equations for sizing control valves
9. Keller A, Rott H (1997) The effect of flow turbulence on cavitation inception. In: Proc. of 1997 ASME Fluids Eng. Summer Conf. Vancouver (Canada)

10. Rodriguez Calvete D, Gosset A, Pierrat D, Couzinet A (2016) Numerical simulation of cavitating flow in a globe valve: Confrontation of OpenFOAM® and CFX. In: Proc. of the ASME 2016 Pressure Vessels and Piping Conference. Vancouver, British Columbia, Canada, ASME
11. Sauer J, Schnerr G (2000) Unsteady cavitating flow: a new cavitation model based on a modified front capturing method and bubble dynamics. In: Proc. of 2000 ASME Fluids Eng. Summer Conf. Boston (MA, USA), pp 11–15
12. Singhal AK, Athavale MM, Li H, Jiang Y (2002) Mathematical basis and validation of the full cavitation model. J of Fluids Engineering 124(3):617–624

CFD Analysis and Optimisation of Tidal Turbine Arrays Using OpenFOAM®

G. R. Tabor

Abstract Tidal estuaries represent a significant and accessible source of renewable energy for modern society. The regular nature of the tides makes this a valuable resource to exploit. Tidal turbines that extract energy from tidal currents would be one way to do this. The shallow nature of estuaries suggests that these would need to be low power units linked together in large farms, and the modelling and optimisation of such farms of turbines is a significant challenge. In this chapter, I report on a major research effort to model a possible turbine, the AquaScientific Lift/Drag turbine, and the development from this of a simplified CFD model, the Immersed Body Force method, to allow for simulation of small arrays. Following from this, I report on the development of surrogate modelling techniques to allow the prediction of outputs from larger arrays and optimisation using Genetic Algorithm techniques.

1 Introduction

Several factors combine to drive forward the search for renewable and decarbonised forms of energy, in particular, the finite availability of non-renewable sources such as oil (with the imminent arrival of 'peak oil') and the global warming effects of our current carbon-intensive fuels such as coal. Several renewable energy sources are now competitive in the market place with minimal subsidy, such as wind and solar energy. However, these do suffer from inherent deficiencies as they are, by nature, intermittent—no wind turbine will generate power if the wind is not blowing, and the output from a solar panel is seriously reduced when the day is cloudy. Tidal energy has significant advantages over these other sources in that the tides, driven by the motion of the moon, are entirely regular and predictable, with two high and two low tides over the course of (slightly over) 1 day, and thus significant tidal currents occur 4 times in this period. Although the flow speeds are lower than for wind, the much higher density of water means that significant energy is contained in these flows.

G. R. Tabor (✉)
CEMPS, University of Exeter, Harrison Building,
North Park Road, Exeter EX4 4QF, UK
e-mail: g.r.tabor@ex.ac.uk

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_4

1.1 Esturine Tidal Energy

Marine tidal energy has been extensively researched in recent years, with numerous generating devices being proposed, including several of an 'underwater windmill' type. The marine environment, however, is a very challenging environment to work in, with other forces such as wave impact to contend with, as well as the corrosive nature of seawater; and there is also the challenge of transporting any generated electrical power to shore. In many ways, esturine tidal energy may represent a better route to harnessing the power of the tides. River estuaries are, of course, tidal, with significant head changes and flow speeds associated with, for example, the Severn estuary—an estimate of the UK tidal resource suggests that the top 10 sites around the UK could contribute up to 36 TWh/year, about 10% of UK energy needs [1]. River estuaries are more benign environments than deep marine ones, being sheltered from the most powerful wave surges. They are also more accessible, both in terms of cabling to deliver the electrical power to shore, and also for installation and maintenance of any devices.

Numerous approaches have been proposed to harness tidal power in estuaries. The best known are barrages which block the estuary, holding back the tide on one side until sufficient difference in head has been created to drive a low-head turbine. The La Rance installation in France is the only current commercial installation [2] although various variants of this idea have been and continue to be considered for the Severn Estuary in the UK. A variant of the barrage idea is a tidal pool or lagoon, in which a portion of the estuary is isolated by a closed loop of barrage, once again allowing the exploitation of head differences generated by the tides. All barrage schemes, however, suffer from the problem of high initial capital outlay; the whole scheme has to be completed before any electricity can be generated. In the case of the Severn barrage, this could be an investment of tens of billions of £ up front. Full barrages also block the estuary, leading to environmental changes such as silting, and challenges to accommodate other users of the estuary such as shipping.

An alternative approach is to capture energy not from head differences, but from tidal currents through some form of turbine. These could be similar in design to proposed marine current turbines (so standard horizontal axis turbines would be possible) or they could be more innovative. The shallow nature of estuaries suggests that cross-flow turbines that intercept the flow through a cylindrical region immediately below the surface may have advantages. Examples include the Transverse Horizontal Axis Water Turbine (THAWT, see Fig. 1), vertical axis Darrieus turbines, as well as AquaScientific's Lift/Drag turbine studied in this work (also illustrated in Fig. 1). Most significantly, such turbines could be built and installed singly, and a site could begin generating power as soon as a single turbine was in place, providing a less steep path to commercialisation. However, the shallow nature of the estuary also implies that individual turbines are likely to be low power (10's of MW each), and so a commercial scale installation is likely to involve hundreds of turbines working together. The interaction of such turbines must be studied and techniques developed to predict and optimise power generation for such farm arrays.

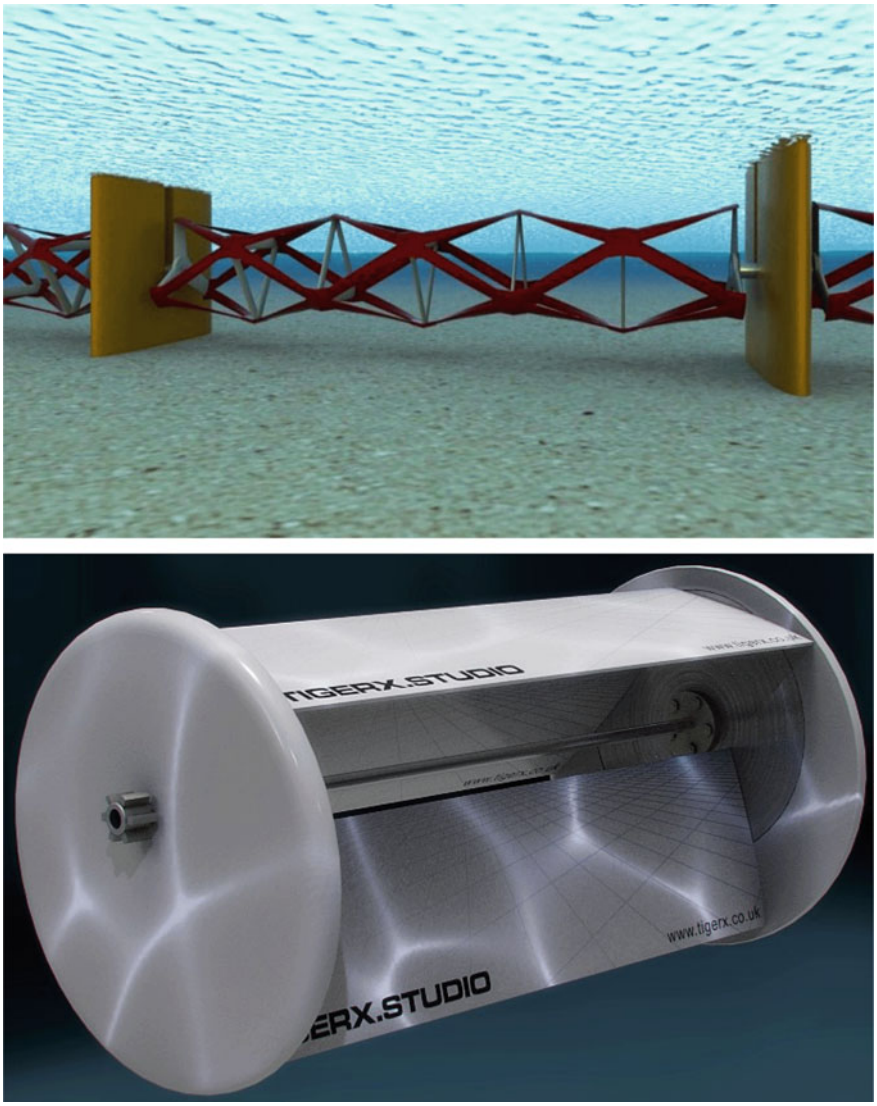


Fig. 1 Some illustrative tidal cross-flow turbines. Top: Transverse Horizontal Axis Water Turbine (THAWT). Image credit: Kepler Energy Ltd. Bottom: AquaScientific Lift/Drag turbine

1.2 Lift/Drag Turbine

The Lift/Drag turbine was proposed a few years ago by the startup company Aqua-Scientific as an esturine tidal turbine. It is a cross-flow turbine similar in action to a Voigt-Schneider impeller, consisting of individual rectangular turbine blades that

can rotate around their long axis, mounted on an armature rotating on a parallel axis at 1/2 the rotational speed. The blades act in drag mode on one side of the turbine; as the armature rotates, the blades reorientate to develop lift on the other side of the turbine, with the net effect that the blades generate a consistent torque throughout much of their rotation. The unit is bidirectional, i.e. it will generate power from tidal flow in either direction and is also self-starting. It operates as a cross-flow turbine, with a relatively high blockage factor; it could be installed to operate just below the water surface and intercept a large fraction of the upper (high-energy) flow. Moreover, energy is extracted through the swept volume (a cylindrical volume) as against a swept disk for a standard HAWT arrangement, resulting in a high efficiency of extraction.

1.3 Project Aims

The objective of the project[1] was to develop techniques for modelling and optimising arrays of tidal turbines, concentrating on the Lift/Drag design proposed by AquaScientific, but capable of being extended to other specific designs. A multilevel approach was adopted to accomplish this. Detailed CFD (and experimental) work was undertaken to understand the complex behaviour of flow around the turbine, using RANS modelling and sliding mesh in OpenFOAM® [3–6]. Although highly accurate, this was computationally extremely expensive, even in 2D, and so a larger scale model was developed and implemented in OpenFOAM®. This was an empirical model of the turbine behaviour using body forces to represent the effects of the blades, referred to as the Immersed Body Force (IBF) model, similar to the Actuator Disk/Actuator Line models commonly used for HAWT turbines, and this allowed the LES simulation of single and small groups of turbines. Whilst computationally less costly than the detailed CFD, the model was still too expensive to run a full installation of hundreds of individual turbines, or to run multiple cases to optimise the farm layout or operation. For the largest scale, therefore, a surrogate modelling approach was adopted; a limited number of farm simulations were run using the IBF model and appropriate symmetry boundary conditions to represent larger (technically infinite) farms, and various approaches were used to create surrogate models. Surrogate models are computationally inexpensive models which are fit to the CFD results to cheaply model the inputs/outputs relation for the system; this could then be used in a genetic algorithm optimisation system to explore the optimal configuration.

The rest of the chapter divides into three main sections. In Sect. 2, I present the results from the detailed CFD model using RANS and sliding mesh (GGI). In Sect. 3, results are shown from the development of the IBF method. Finally, in Sect. 4, the surrogate model derivation and array optimisation work is presented.

[1]Optimal Design of Very Large Tidal Stream Farms; for Shallow Esturine Applications, EPSRC project EP/J010138/1.

2 Detailed CFD

In parallel with the computational programme, a series of tank-scale tests were carried out with a turbine of radius $R = 0.055$ m and blade chord lengths between 0.05 and 0.095 m. This was adopted as a standard turbine, and all CFD simulations in this paper refer to turbines of this radius, with blade chord length 0.05 m. The blade profiles were arcs of circles of radius 0.2 m.

For the detailed CFD calculations, transient simulation with `pisoFoam` was used, with an adapted k–ω-SST model wall resolved to $y^+ \sim 1$ to 2. The mesh was 2D, constructed using the commercial meshing code Pointwise. Because of the complex nature of the blade motion, five independent mesh regions were established; one circular domain for each blade rotating within a larger circular domain representing the armature, itself rotating within an extended rectangular domain representing the exterior flow. The mesh structure is shown in Fig. 2. The cell count was 100,000 cells with gradual mesh inflation. The turbulence model adaption was required as initial simulations demonstrated a significant dependence of the results on the turbulence level at the inlet (a result observed independently elsewhere, see [7]). As a result, the standard OpenFOAM® k–ω implementation was adapted to include the sustain terms developed by Spalart and Rumsey [8] to keep the turbulence level at an appropriate level.

The results are shown in Figs. 3, 4 and 5. The wake region behind the turbine is captured well, and indeed the details of the vortex shedding from the blades and blade/vortex interactions are all resolved. The physical processes being modelled

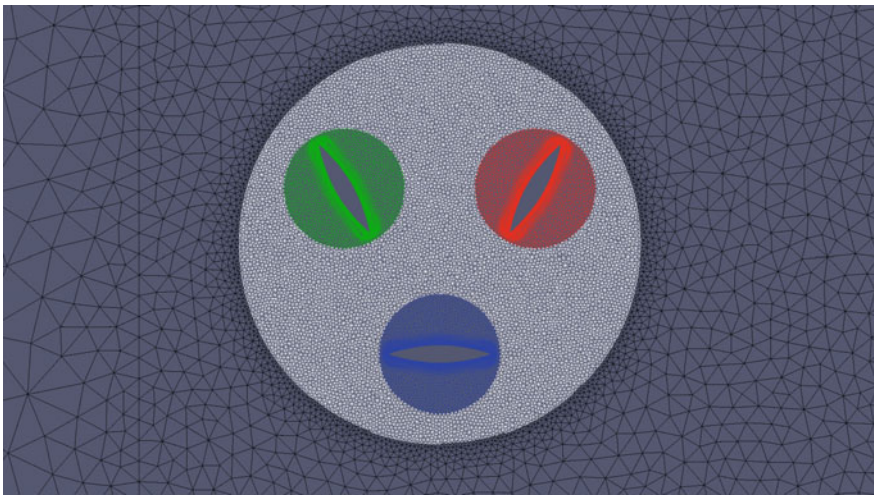


Fig. 2 Computational mesh for the rotor for GGI sliding mesh calculations. The 5 independent mesh regions are shown in different colours; each blade requires its own rotating mesh region within the armature region, which itself rotates within the outside mesh

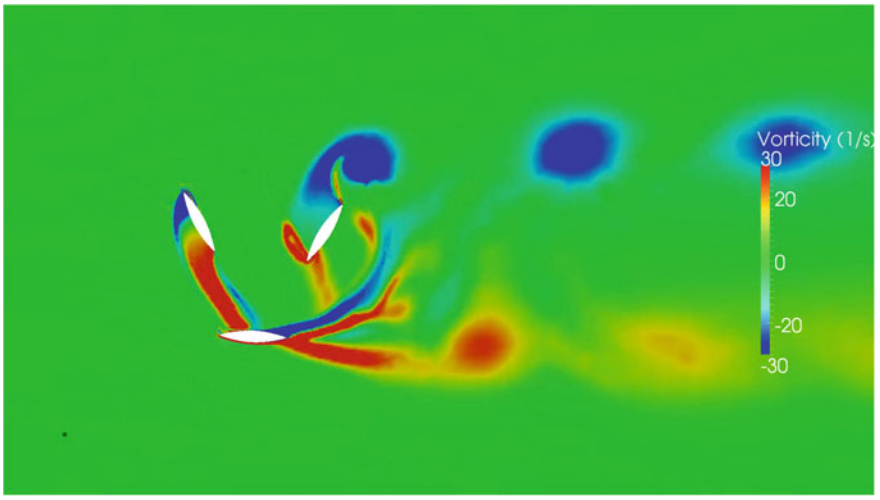


Fig. 3 Instantaneous vortex shedding behind the turbine

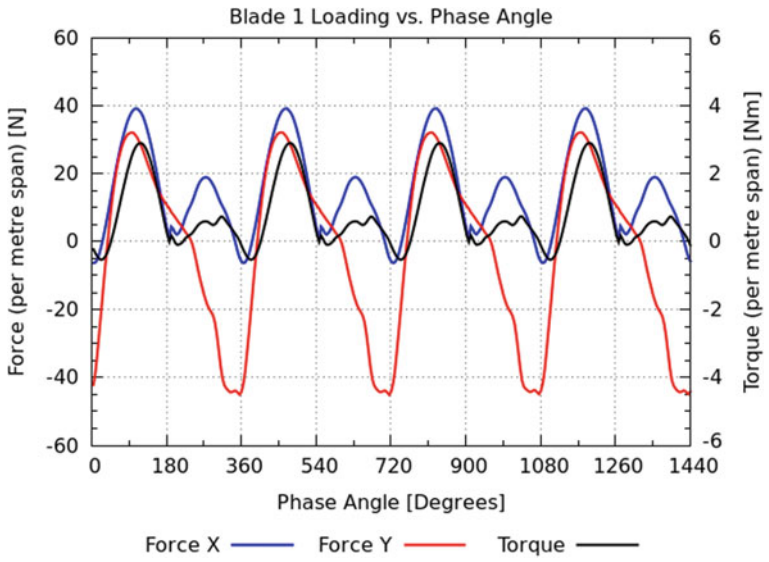


Fig. 4 Time-resolved forces/torques on the Lift/Drag turbine blades

here are extremely complex, with aspects of transient stall being resolved as the blades pass through a full 360° rotation. Individual blade forces and torques were calculated (Figs. 4 and 5), enabling time series data on these as shown in the graphs (phase angle is equivalent to time in the computation). In addition, wake recovery was also plotted. This was compared with the limited amount of experimental data available at the time and discrepancies noted; the rate of wake recovery was somewhat

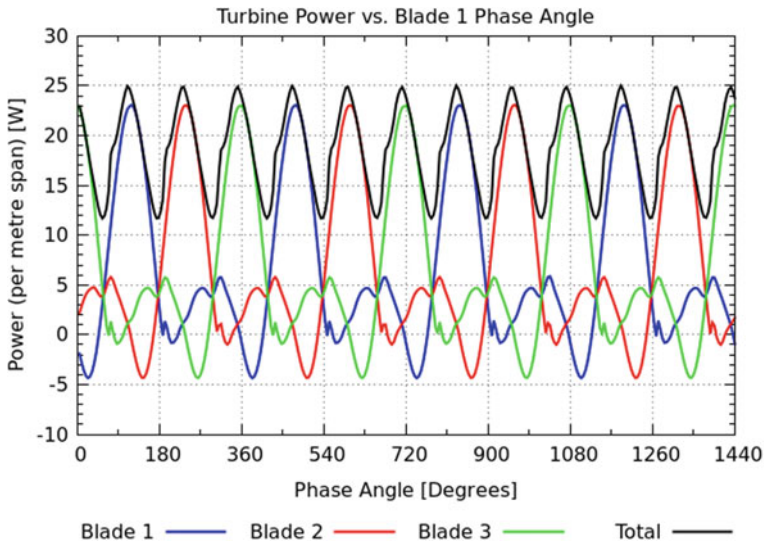


Fig. 5 Time-resolved power for the Lift/Drag turbine

reduced compared to the experimental data, probably due to the lack of 3D effects in the simulation. Further details can be found in [9]. However, the detailed CFD simulations were already computationally expensive (around 5 days on 16 cores of a parallel machine, see Table 1) and the costs of moving this to full 3d simulation were thought to be prohibitive. Accordingly, a new method had to be developed to represent the turbine in a less costly fashion, as discussed in the next section.

3 Immersed Body Force Method

The sliding mesh approach detailed in Sect. 2 gives detailed results on the motion of an individual turbine but is extremely expensive; given limited computational resources, it was only possible to run simulations in 2D. Thus, a new model for the turbines was developed, referred to as the Immersed Body Force technique [10]. This is similar to the actuator disk/actuator line models used for HAWT turbines (wind and water) in that it represents the turbine in terms of spatially constrained body forces within the domain, as shown in Fig. 6. The development of this model not only alleviated the requirement to resolve the rotational motion of the blades but it also massively reduced the meshing requirement close to the blades, enabling the problem to be calculated on a coarser, more uniform mesh. This enabled computational resources to be redirected into other physical modelling issues; the calculations were run in 3D, with LES modelling using the `oneEqEddy` SGS model, and VOF to represent the presence of the free surface.

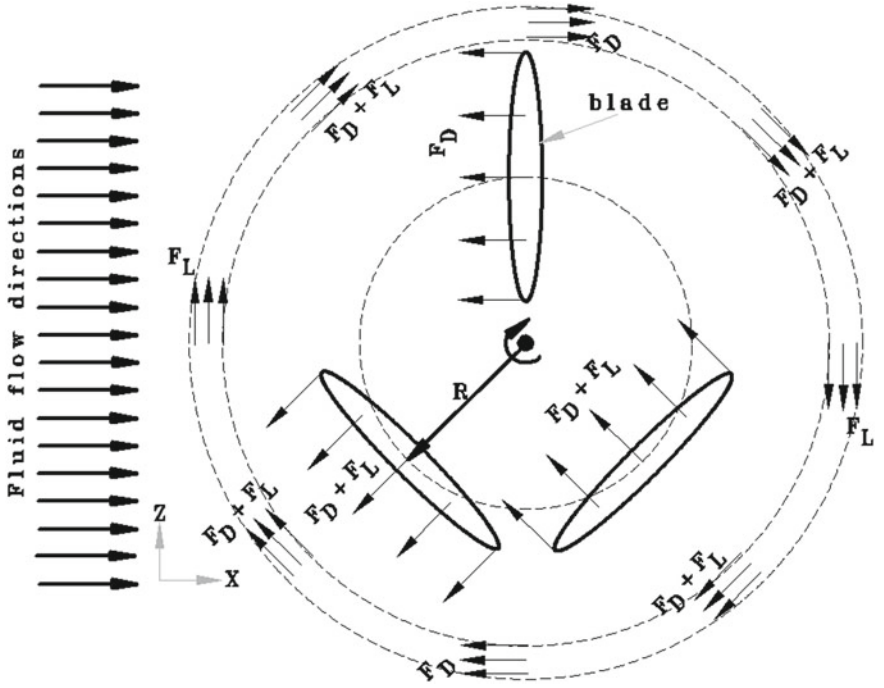


Fig. 6 Body forces for the IBF method. There are two sets of forces involved; one set representing the blades and the other the rotational motion of the system

In the IBF approach, two sets of body forces were introduced to represent the turbine; one set representing the blades themselves and a second set as a 'vortex ring' in an attempt to mimic the effect of the rotation of the overall system on the armature.

$$\overline{\mathbf{F}} = \overline{\mathbf{F}}_D + \overline{\mathbf{F}}_L \tag{1}$$

Since the motion of the individual blades in this case is not explicitly simulated, we cannot evaluate the mechanical power directly; however, a standard technique in actuator disk theory is to relate this to the momentum change across a control volume around the turbine.

3.1 Validation

Results from this are shown in Figs. 7 and 8. Laboratory testing was carried out for a single turbine in a flow channel, with flow rate and turbine rotational speed being measured under a range of imposed mechanical torque conditions. The rotational rate was recorded optically using a light gate; the flow rate was measured with a rotorme-

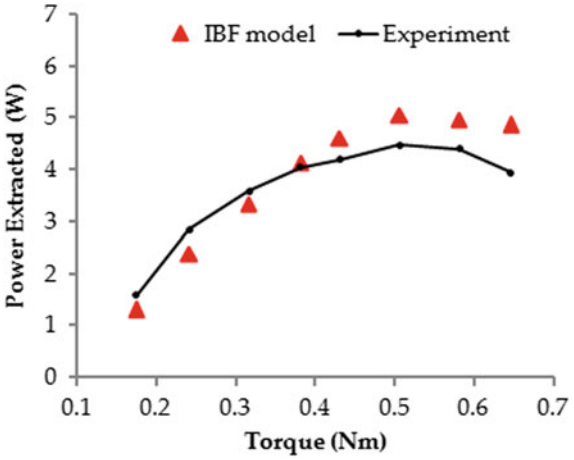


Fig. 7 Calibration of the IBF model against experimental results for the power curve

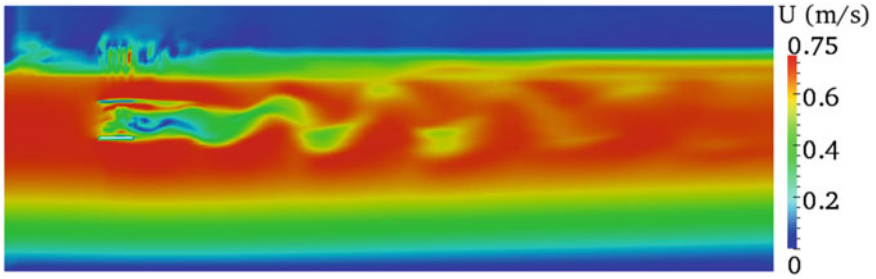


Fig. 8 Results for a single IBF turbine. A vertical section through the midline of the turbine is shown, featuring vortex shedding very similar to that seen in the GGI simulations

ter, and the imposed torque varied mechanically. The resulting data was compared with functionally equivalent CFD simulations (Fig. 7) to calibrate the coefficients in the model (for further details see [10]). Simulations of individual turbines (Fig. 8) show vortex shedding consistent with the resolved CFD shown above (Sect. 2) and a wake recovery consistent with more detailed experimental results [10, 11]. Computational costs were considerably reduced, with a calculation for an individual turbine taking 17 h on 12 cores with a mesh resolution of 148,000 cells, as shown in Table 1. Further details of the calibration and results from comparison with measured wake profiles are provided in [12].

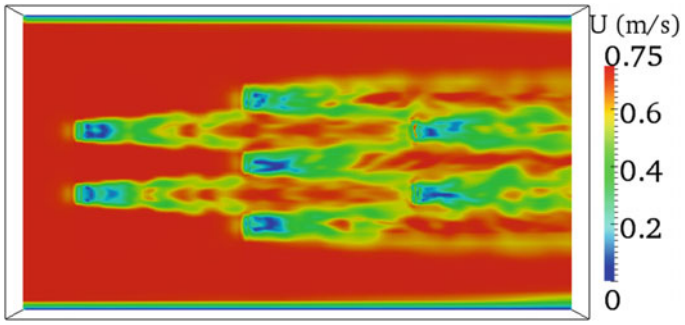


Fig. 9 Results for a farm array of 7 turbines, plan view through the turbine centre. Turbine-wake interaction can be studied with these arrangements

3.2 Farm Modelling

The lowered costs of the IBF model made it conceivable to undertake simulation of multiple turbines in farm arrays. Farms of 7 and 15 turbines were investigated with both inline and staggered configurations [13, 14], and the effects of turbine–turbine interaction investigated . Some typical results are shown in Fig. 9 for a 7-turbine array demonstrating the wake interaction with downstream turbines [14, 15]. The mesh size for this calculation was $O(10^6)$ cells and typical calculations took 44 h for this, as listed in Table 1.

4 Optimisation

The overall objective of the project was to investigate optimisation of the tidal arrays. A tidal array is likely to consist of several rows of turbines, creating an optimisation problem of determining the streamwise and spanwise distances between the individual turbines, with the additional possibility of separations being different between rows. This optimisation problem would be faced at installation, requiring the designers to find the best possible siting for the different turbines. Additionally, it may be desirable to vary the loadings so that the first row of turbines does not necessarily extract the maximum power; the optimum power for the farm may be achieved by optimising power takeup in rows further downstream. This part of the optimisation might change according to the current flow conditions, and so might be variable throughout the tidal cycle, throughout the year (spring and neep tides) or even possibly throughout the turbine's lifespan. However, a straightforward power maximisation from the farm is not the whole issue either, since the cost of the installation must also be considered. With any renewable energy source, the efficiency of the system is of lesser importance (since the energy is ultimately free) than the cost per unit of power generated, and so we are also interested in minimising the

cost of the farm. For a fixed number of turbines, the costs will scale with the cost of the cabling, and so the area of the farm was thought to be quite a good metric of the cost of construction. Hence we have a multi-parameter, multi-objective optimisation problem to investigate.

Genetic Algorithms are a known approach for multi-parameter optimisation. In a GA, a genome representing the values for all the input parameters is defined, and an initial population representing a spread of parameter values is created. The fitness of each individual in the population is evaluated through calculating some form of cost function (here, calculating the power generated from the farm and its total area). Then, less-fit individuals are eliminated from the population, and a new generation of individuals created from the survivors through a combination of genetic recombination and mutation. The process then repeats through several generations (typically 30–50). The process is advantageous as it thoroughly explores the parameter space, thus reducing the likelihood of getting trapped in a local optima, and at the same time it reliably identifies the global optima for the system. For a multi-objective problem, as this one, the result is a set of *non-dominated* solutions known as the *pareto front*, which represents the best possible tradeoff between the different objectives. In this case, these will be the best possible solutions for any given combination of power output and area; for any given non-dominated solution there is no solution possible which is better in both power and area; any increase in power will be at the expense of a larger farm area (and vice versa). Such information would be enormously valuable for a tidal farm designer.

However, there is still a problem. As shown in Table 1, the cost of simulating a modest array of 7 turbines was 44 h on 12 cores. Scaling this to represent the cost of simulating, say, 30 individuals in 30 generations gives 39,600 h, or around 4.5 years' calculation. Thus, it was decided that an intermediate stage would be needed in the form of a *surrogate model*, a simplified mathematical model that can be fit to a limited subset of core data and then used to extrapolate the results for any set of input parameters. Various approaches for the surrogate model have been investigated. Kriging, in which the value of the surrogate function at any given point is predicted from a weighted average of the known values in the neighbourhood of the point, is a statistical technique closely related to regression analysis, which provides an explicit functional evaluation of the input–output relationship. Other techniques used in the literature include Artificial Neural Networks, which can be trained to provide a learned relation between inputs and outputs, although this is only an implied relationship from the ANN, not an explicit mathematical formula. Whichever form of surrogate model is used, the number of CFD evaluations necessary to derive it is considerably smaller, in our case a few 10's of evaluations suffice, against the 900 necessary for the direct optimisation (Fig. 10).

To simplify the problem, a three-row tidal farm was investigated, using three individual turbines and symmetry boundary conditions as shown in Fig. 11 to represent infinite arrays of turbines in the spanwise directions. The longitudinal and lateral distances between the turbines and their loadings were varied in discrete steps, representing six different parameters to explore, or 592,704 different discrete layouts. Both staggered (as in Fig. 11) and inline arrangements were investigated. The stan-

Table 1 Table showing typical computational times for the different simulations run on 'Callisto', a CFD computer cluster at Exeter. Nodes on the cluster comprise 4xIntel Xeon E5-4620 CPU's with 8 cores each running at 2.20 GHz, and 132 GB RAM per node

Case	Details	Spec	Time
GGI	160k cells, 30 revolutions	16 cores	5 days
IBF	1 turbine, 148k cells	12 cores	17 h
IBF	7 turbines, 1M cells	12 cores	44 h

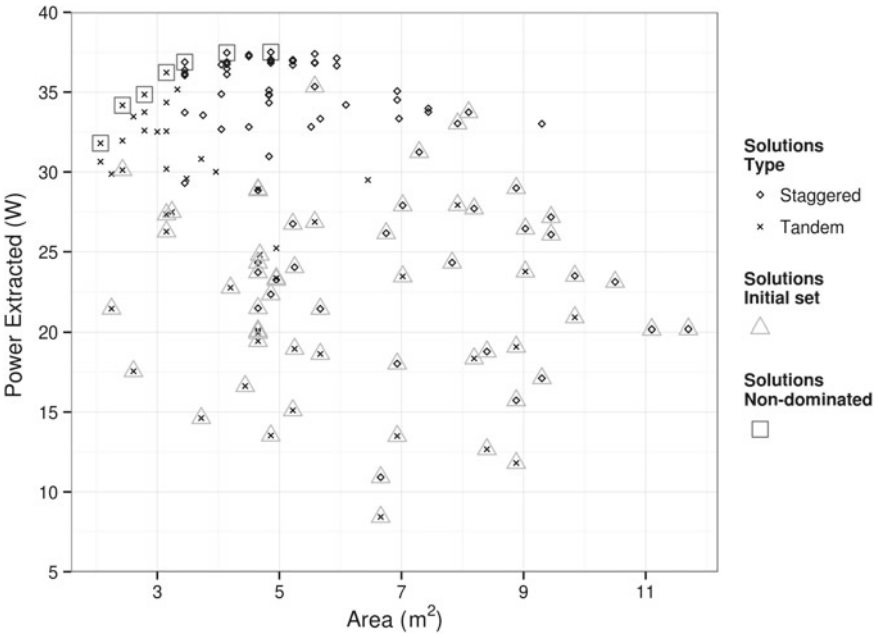


Fig. 10 Results for the farm optimisation, showing both inline (tandem) and staggered configurations. The Pareto front (non-dominated solutions) is marked with square boxes

dard practice was adopted of using a Latin hypercube technique to sample the initial parameter space, with 30 evaluations of the CFD model to generate the surrogate model, together with further evaluations to refine the model as the optimisation process proceeded. Various surrogate models were investigated [16]; the results shown here use Kriging. Figure 4 plots the initial and final population sets for both staggered and tandem (inline) arrangements, plotting power generation against farm area. The non-dominated solutions are shown in square boxes and from a Pareto set in the top left-hand part of the figure. This seems to saturate at a power extracted of around 37 W and area of about 4 m^2, suggesting that above this area, the turbine wakes do not interfere, and thus there is no benefit to an increased turbine spacing beyond this.

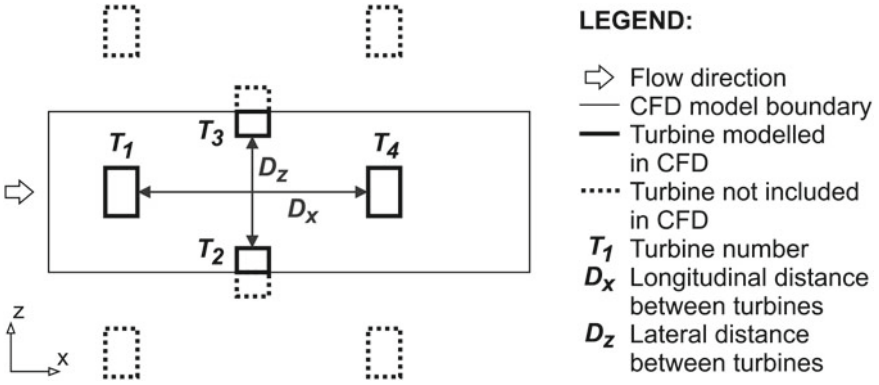


Fig. 11 Unit cell layout for the staggered configuration

5 Conclusions

In this chapter, I report on a significant research programme in tidal renewable energy. This has involved CFD simulation with OpenFOAM® at all levels, from detailed simulation of a novel cross-flow turbine (the AquaScientific Lift/Drag turbine) using GGI methods, to the development of a cheaper Immersed Body Force (IBF) representation which has allowed the simulation of these turbines in 3D (including detailed resolution of the wake and free surface) and of small groups of turbines. From this, the project was able to develop a methodology for multi-objective optimisation of turbine farms using a combination of surrogate modelling and Genetic Algorithms. Whilst this was demonstrated on a small idealised array, with sufficient (and not unachievable) additional computational resources, the methodology could be expanded to cover realistically sized farms.

Acknowledgements This work has involved a significant number of collaborators, too many to list as co-authors on the paper. On the CFD side, this has involved Dr. Mulualem Gebreslassie, Dr. Matt Berry and Mr. Ben Ashby; the optimisation involved Prof. Dragan Savic, Dr. Michele Guidolin, Dr. Helena Mala Jetmarova and Mr. Ogaday Willers Moore, and the project led by P.I. Prof. Mike Belmont. Experimental work was overseen by Prof. Ian Bryden and Dr. Tom Bruce at the FloWave facility at Edinburgh University. The work was funded by the EPSRC (Optimal Design of Very Large Tidal Stream Farms; for Shallow Esturine Applications) and the Laing Foundation.

References

1. House of Commons Select Committee on Science and Technology. Wave and tidal energy – seventh report 2000/01 (HC291), 2001.
2. R. H. Charlier. Forty candles for the Rance River TPP tides provide renewable and sustainable power generation. *Renewable and Sustainable Energy Reviews*, 11:2032–2057, 2007.

3. M. Beaudoin and Jasak. H. Development of an arbitrary mesh interface for turbomachinery simulations with OpenFOAM®. In *Open Source CFD International Conference, Berlin*, December 2008.
4. M. Beaudoin, H. Nilsson, M. Page, R. Magnan, and H. Jasak. Evaluation of an improved mixing plane interface for OpenFOAM®. 26th Symposion on Hydraulic Machinery and Systems, August 2014.
5. H. Jasak. Dynamic mesh handling in OpenFOAM®. 48th AIAA Aerospace Sciences Meeting, AIAA Paper, January 2009. Orlando, FL.
6. H. Jasak and M. Beaudoin. OpenFOAM® turbo tools: From general purpose CFD to turbomachinery simulations. In *Proceedings of ASME-JSME-KSME Joint Fluids Engineering Conference*. ASME-JSME-KSME, July 2011. Paper No. AJK2011-05015.
7. T. Blackmore, W. Batten, and A. Bahaj. Influence of turbulence on the wake of a marine current turbine simulator. *Proc. Roy. Soc. A*, 470(2170):20140331, 2014.
8. P. R. Spalart and C.L. Rumsey. Effective inflow conditions for turbulence models in aerodynamic calculations. *AIAA.J.*, 45(10):2544–2553, 2007.
9. M. J. Berry. *Hydrodynamic Analysis of the Momentum-Reversal and Lift Tidal Turbine*. PhD thesis, CEMPS, University of Exeter, 2017.
10. M. G. Gebreslassie, G. R. Tabor, and M. R. Belmont. Numerical simulation of a new type of cross flow tidal turbine using OpenFOAM® - part i: Calibration of energy extraction. *Renewable Energy*, 50:994–1004, 2013.
11. M.G.Gebreslassie, G.R.Tabor, and M.R.Belmont. CFD simulations for sensitivity analysis of different parameters to the wake characteristics of a tidal turbine. *Open Journal of Fluid Dynamics*, 2:56–64, 2012.
12. M. G. Gebreslassie, S. O. Sanchez, G. R Tabor, M. R. Belmont, T. Bruce, G. S. Payne, and I Moon. Experimental and cfd analysis of the wake characteristics of tidal turbines. *Int. J. Marine Energy*, 16:209–219, 2016.
13. M. G. Gebreslassie, G. R. Tabor, and M. R. Belmont. Investigation of the performance of a staggered configuration of tidal turbines using CFD. *Renewable Energy*, 80:690–698, 2015.
14. M. G. Gebreslassie, G. T. Tabor, and M. R. Belmont. Numerical simulation of a new type of cross flow tidal turbine using OpenFOAM® - part ii: Investigation of turbine-to-turbine interaction. *Renewable Energy*, 50:1005–1013, 2013.
15. M.G. Gebreslassie, M.R. Blemont, and G.R. Tabor. Comparison of analytical and CFD modelling of the wake interactions of tidal turbines. In *IWTEC2013, Aalborg, Denmark, 2–5th Sept 2013*, 2013.
16. W.O.Willers Moore, H. Mala-Jetmarova, M.G. Gebreslassie, G. R. Tabor, M. R. Belmont, and D. Savic. Comparison of multiple surrogates for 3d CFD model in tidal farm optimisation. In *Procedia Engineering*. 12th International Conference on Hydroinformatics, HIC 2016, 2016.

Combining an OpenFOAM®-Based Adjoint Solver with RBF Morphing for Shape Optimization Problems on the RBF4AERO Platform

E. M. Papoutsis-Kiachagias, K. C. Giannakoglou, S. Porziani, C. Groth, M. E. Biancolini, E. Costa and M. Andrejašič

Abstract This chapter presents a combination of an OpenFOAM®-based continuous adjoint solver and a Radial Basis Function (RBF)-based morpher forming a software suite able to tackle shape optimization problems. The adjoint method provides a fast and accurate way for computing the sensitivity derivatives of the objective functions (here, drag and lift forces) with respect to the design variables. The latter control a group of RBF control points used to deform both the surface and volume mesh of the CFD domain. The use of the RBF-based morpher provides a fast and robust way of handling mesh and geometry deformations with the same tool. The coupling of the above-mentioned tools is used to tackle shape optimization problems in automotive and aerospace engineering. This work was funded by the RBF4AERO "*Innovative benchmark technology for aircraft engineering design and efficient design phase optimisation*" project funded by the EU 7th Framework Programme (FP7-AAT, 2007-2013) under Grant Agreement No. 605396 and the presented methods are available for use through the RBF4AERO platform (www.rbf4aero.eu).

E. M. Papoutsis-Kiachagias (✉) · K. C. Giannakoglou
Parallel CFD & Optimization Unit,
National Technical University of Athens (NTUA), Athens, Greece
e-mail: vaggelisp@gmail.com

K. C. Giannakoglou
e-mail: kgianna@central.ntua.gr

S. Porziani · E. Costa
D'Appolonia S.p.A., Rome, Italy
e-mail: stefano.porziani@dappolonia.it

E. Costa
e-mail: emiliano.costa@dappolonia.it

C. Groth · M. E. Biancolini
University of Rome Tor Vergata (UTV), Rome, Italy
e-mail: corrado.groth@uniroma2.it

M. E. Biancolini
e-mail: biancolini@ing.uniroma2.it

M. Andrejašič
PIPISTREL d.o.o. Ajdovščina, R&D, Department of Aerodynamics, Ajdovščina, Slovenia
e-mail: matej.andrejasic@pipistrel.si

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_5

Nomenclature

RBF	Radial basis functions
SD	Sensitivity derivatives
FD	Finite differences
PDE	Partial differential equation
SI	Surface integrals
FI	Field integrals
E-SI	Enhanced surface integrals

1 Introduction

During recent years, CFD-based aerodynamic shape optimization has been attracting the interest of both academia and industry. The constituents needed for executing an automated shape optimization loop include the flow solver, the geometry parameterization (the parameters of which act as the design variables), an optimization method capable of computing the optimal values of the design variables and a way to adapt (or regenerate) the computational mesh to each candidate solution.

In the studies presented herein, the steady-state flow solver of the open-source CFD toolbox OpenFOAM® is used to numerically solve the Navier-Stokes equations for incompressible, turbulent flows.

Shape parameterization techniques can be divided into two categories, i.e., those parameterizing only the surface to be optimized and those that simultaneously also deform the surrounding nodes of the interior mesh. The great advantage of the latter is that the interior of the mesh is also deformed, thus avoiding costly re-meshing and allowing for the initialization of the flow field from the solution of the previous optimization cycle, since the mesh topology is preserved. Here, a number of parameters controlling the positions of groups of RBF control points are used as the design variables, using technology and methods of the RBF Morph software [2].

Gradient-based optimization methods require a high degree of effort to develop but can have a cost per optimization cycle that does not scale with the number of design variables, if the adjoint method is used to compute the sensitivity derivatives (SD). In this work, a continuous adjoint method that takes into consideration the differentiation of the turbulence model PDE is used to compute the SD of the force objective function w.r.t. the design variables [7]. The adjoint solver has been implemented in-house, based on the OpenFOAM® software.

The above-mentioned tools are combined in order to form an automated optimization loop. Two applications are presented, namely the drag minimization of the DrivAer car model [3], and the lift-to-drag ratio maximization of a glider plane, developed by Pipistrel, a partner of the RBF4AERO project.

2 Continuous Adjoint Formulation

The derivation of the adjoint equations, their boundary conditions and the SD expression, concerning flows governed by the Navier–Stokes equations coupled with the Spalart–Allmaras model, starts with the definition of the augmented objective function L,

$$L = J + \int_{\Omega} u_i R_i^v d\Omega + \int_{\Omega} q R^p d\Omega + \int_{\Omega} \widetilde{\nu}_a R^{\widetilde{\nu}} d\Omega, \tag{1}$$

where J is the objective function. We assume that J is defined only along the boundary S of the flow domain Ω, so $J = \int_S J_{S,i} n_i dS = \int_{S_W} J_{S_W,i} n_i dS + \int_{S_O} J_{S_O,i} n_i dS$, where $S = S_W \cup S_O$, S_W is the controlled solid wall, S_O any other boundary of Ω and n_i the outward unit normal vector to S. Apart from J, L also includes the integrals of the residuals of the momentum ($R_i^v = 0$), continuity ($R^p = 0$) and turbulence model ($R^{\widetilde{\nu}} = 0$) equations, multiplied by the fields of the adjoint velocities u_i, adjoint pressure q and adjoint turbulence model variable $\widetilde{\nu}_a$, [8]. Dropping the last integral on the r.h.s. of Eq. 1 results in the so-called "frozen turbulence" assumption, which neglects variations in the eddy viscosity field and leads to reduced SD accuracy, possibly even to wrong sensitivity signs [8]. A review of continuous adjoint methods for turbulent flows can be found in [7].

A literature survey shows that continuous adjoint can be formulated in two different ways, which both give the same adjoint field equations and boundary conditions, yet different expressions for the gradient of J with respect to (w.r.t.) b_n, where b_n, $n = 1, \ldots, N$ are the design variables.

The first formulation leads to SD expressions with Field Integrals (FI), including the variations in the spatial coordinates $\mathbf{x}$ w.r.t. $\mathbf{b}$, a.k.a. grid sensitivities. A typical way to compute $\delta\mathbf{x}/\delta\mathbf{b}$ is through finite differences (FD) at a cost that scales linearly with N. The *FI*approach starts by formulating

$$\begin{aligned}\frac{\delta L}{\delta b_n} = \frac{\delta J}{\delta b_n} + \int_{\Omega} \left(u_i \frac{\delta R_i^v}{\delta b_n} + q \frac{\delta R^p}{\delta b_n} + \widetilde{\nu}_a \frac{\delta R^{\widetilde{\nu}}}{\delta b_n} \right) d\Omega \\ + \int_{\Omega} \left(u_i R_i^v + q R^p + \widetilde{\nu}_a R^{\widetilde{\nu}} \right) \frac{\delta (d\Omega)}{\delta b_n},\end{aligned} \tag{2}$$

where the last integral vanishes, since $R_i^v = R^p = R^{\widetilde{\nu}} = 0$ in Ω. By developing Eq. 2, [4], integrals of expressions multiplied by $\delta v_i/\delta b_n$, $\delta p/\delta b_n$ and $\delta\widetilde{\nu}/\delta b_n$ arise. By zeroing those expressions in Ω, the field adjoint equations are formulated [7, 8]:

$$R^q = -\frac{\partial u_j}{\partial x_j} = 0 \tag{3a}$$

$$R_i^u = u_j \frac{\partial v_j}{\partial x_i} - \frac{\partial (v_j u_i)}{\partial x_j} - \frac{\partial \tau_{ij}^a}{\partial x_j} + \frac{\partial q}{\partial x_i} + \widetilde{\nu_a} \frac{\partial \widetilde{\nu}}{\partial x_i} - \frac{\partial}{\partial x_l} \left(\widetilde{\nu_a} \widetilde{\nu} \frac{\mathscr{C}_Y}{Y} e_{mjk} \frac{\partial v_k}{\partial x_j} e_{mli} \right) = 0, \quad i = 1, 2, 3 \tag{3b}$$

$$R^{\widetilde{\nu_a}} = -\frac{\partial (v_j \widetilde{\nu_a})}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(\frac{\nu + \widetilde{\nu}}{\sigma} \right) \frac{\partial \widetilde{\nu_a}}{\partial x_j} \right] + \frac{1}{\sigma} \frac{\partial \widetilde{\nu_a}}{\partial x_j} \frac{\partial \widetilde{\nu}}{\partial x_j} + 2 \frac{c_{b2}}{\sigma} \frac{\partial}{\partial x_j} \left(\widetilde{\nu_a} \frac{\partial \widetilde{\nu}}{\partial x_j} \right) + \widetilde{\nu_a} \widetilde{\nu} \mathscr{C}_{\widetilde{\nu}} + \frac{\partial v_t}{\partial \widetilde{\nu}} \frac{\partial u_i}{\partial x_j} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) + (-P(\widetilde{\nu}) + D(\widetilde{\nu}))\, \widetilde{\nu_a} = 0, \tag{3c}$$

where $P(\widetilde{\nu})$ and $D(\widetilde{\nu})$ are the production and dissipation terms of the Spalart–Allmaras RANS turbulence model, τ_{ij}^a are the components of the adjoint stress tensor and functions $\mathscr{C}_Y$, $\mathscr{C}_{\widetilde{\nu}}$ can be found in [8]. Adjoint boundary conditions are derived by zeroing the expressions multiplying $\delta p/\delta b_n$, $\delta v_i/\delta b_n$, $\delta \widetilde{\nu}/\delta b_n$ and $\delta \tau_{ij}/\delta b_n$ in the surface integrals of $\delta L/\delta b_n$ [7]. The remaining terms in $\delta L/\delta b_n$ yield the SD expression

$$\left. \frac{\delta J}{\delta b_n} \right|_{FI} = \int_\Omega \left(-u_i v_j \frac{\partial v_i}{\partial x_k} - u_j \frac{\partial p}{\partial x_k} - \tau_{ij}^a \frac{\partial v_i}{\partial x_k} + u_i \frac{\partial \tau_{ij}}{\partial x_k} + q \frac{\partial v_j}{\partial x_k} + A_{jk}^t \right) \frac{\partial}{\partial x_j} \left(\frac{\delta x_k}{\delta b_n} \right) d\Omega + W(0), \tag{4}$$

where

$$\begin{aligned} W(\phi) = &- \int_{S_W} \left(-u_k n_k + \frac{\partial J_{S_W,k}}{\partial \tau_{lz}} n_k n_l n_z \right) \left(\tau_{ij} \frac{\delta (n_i n_j)}{\delta b_n} + \phi \frac{\partial \tau_{ij}}{\partial x_k} \frac{\delta x_k}{\delta b_n} n_i n_j \right) dS \\ &- \int_{S_W} \frac{\partial J_{S_W,k}}{\partial \tau_{lz}} n_k t_l^I t_z^I \left(\tau_{ij} \frac{\delta (t_i^I t_j^I)}{\delta b_n} + \phi \frac{\partial \tau_{ij}}{\partial x_k} \frac{\delta x_k}{\delta b_n} t_i^I t_j^I \right) dS + \int_{S_W} J_{S_W,i} \frac{\delta (n_i dS)}{\delta b_n} \\ &- \int_{S_W} \left(\frac{\partial J_{S_W,k}}{\partial \tau_{lz}} n_k (t_l^{II} t_z^I + t_l^I t_z^{II}) \right) \left(\tau_{ij} \frac{\delta (t_i^{II} t_j^I)}{\delta b_n} + \phi \frac{\partial \tau_{ij}}{\partial x_k} \frac{\delta x_k}{\delta b_n} t_i^{II} t_j^I \right) dS \\ &- \int_{S_W} \frac{\partial J_{S_W,k}}{\partial \tau_{lz}} n_k t_l^{II} t_z^{II} \left(\tau_{ij} \frac{\delta (t_i^{II} t_j^{II})}{\delta b_n} + \phi \frac{\partial \tau_{ij}}{\partial x_k} \frac{\delta x_k}{\delta b_n} t_i^{II} t_j^{II} \right) dS \end{aligned} \tag{5}$$

and t_i^I, t_i^{II} are the components of the tangential-to-the-surface unit vectors (in 3D shapes). In Eq. 4, one should notice the presence of the field integral containing the spatial gradient of the grid sensitivities.

The alternative SI formulation, with an SD expression comprised of Surface Integrals, is based on the Leibniz theorem for integral variations, namely

$$\frac{\delta L}{\delta b_n} = \frac{\delta J}{\delta b_n} + \int_{\Omega} \left(u_i \frac{\partial R_i^v}{\partial b_n} + q \frac{\partial R^p}{\partial b_n} + \widetilde{\nu}_a \frac{\partial R^{\widetilde{\nu}}}{\partial b_n} \right) d\Omega + \int_S \left(u_i R_i^v + q R^p + \widetilde{\nu}_a R^{\widetilde{\nu}} \right) n_k \frac{\delta x_k}{\delta b_n} dS. \quad (6)$$

The last integral in Eq. 6 is usually ignored [6], by making the debatable assumption that the flow PDEs are satisfied along the boundary. The *SI* formulation (by excluding the last integral in Eq. 6) in shape optimization, leads to the SD expression

$$\left. \frac{\delta J}{\delta b_n} \right|_{SI} = \int_{S_W} \left[- \left(\tau_{ij}^a n_j - q n_i + \frac{\partial J_{S_W,l}}{\partial v_i} n_l \right) \frac{\partial v_i}{\partial x_k} + \frac{\partial J_{S_W,i}}{\partial x_k} n_i \right] \frac{\delta x_k}{\delta b_n} dS + W(1). \quad (7)$$

The *SI* formulation is, by far, less expensive than the *FI* formulation in problems with many design variables. However, because of the elimination of the last surface integral in Eq. 6, the accuracy of the *SI* formulation is not guaranteed. In contrast, the *FI* formulation provides accurate SD.

A new, third formulation, referred to as the E-*SI* (Enhanced-*SI*) adjoint formulation, was recently proposed in [4] by the NTUA group and is intended to alleviate the accuracy issue of the *SI* formulation, while having almost the same computational cost. Since the E-*SI* formulation was developed after the commencement of the RBF4AERO project, it is not included in this software suite. In Fig. 1, the sensitivity derivatives computed by the three different adjoint formulations are compared in an indicative turbulent flow problem, in which the loss of accuracy caused by the utilization of the *SI* approach is showcased.

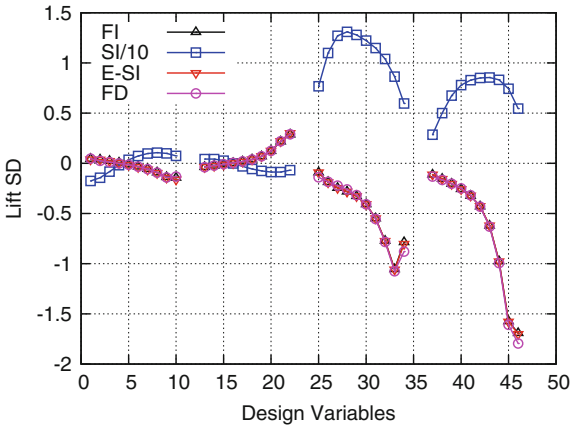


Fig. 1 Turbulent flow around the NACA0012 airfoil ($Re = 10^6$, $a_{\text{inf}} = 3°$, average $y^+ = 0.2$): Comparison of the lift SD computed by the *FI*, *SI*, *E-SI* and FD methods. The SD are computed w.r.t. the x (first half points in the abscissa) and y (second half) coordinates of 24 NURBS control points parameterizing the pressure and suction sides. For scaling reasons, the *SI* results have been divided by 10

3 RBF-Based Morphing

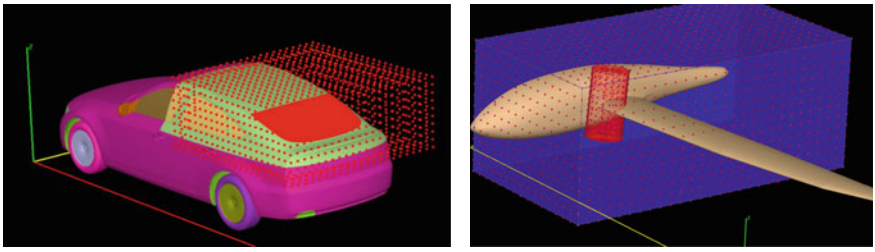
In this section, the mesh morphing algorithm based on RBFs is described. RBFs are mathematical functions able to interpolate data defined at discrete (source) points in an n-dimensional space. The interpolation quality depends on the chosen RBF.

In general, solving the RBF problem requires the solution of a linear system of size M, where M is the source points number. The RBF system solution is computed after defining a set of source points and their displacement. Once the solution has been computed, the displacement of an arbitrary node of the mesh can be expressed as the sum of contributions from all source points. Hence, a desired modification of the mesh nodes position can be rapidly applied preserving mesh topology. RBFs can be classified by the type of support (global or compact) they have, meaning the domain where the chosen RBF is non zero-valued. The interpolation function $s(x) = \sum_{i=1}^{M} \gamma_i \varphi \left(\left\| x - x_{k_i} \right\| \right) + h(x)$ composed of an RBF φ and a polynomial h of order $m-1$, where m is the order of φ, guarantees the compatibility with rigid motions. The degree of the polynomial has to be chosen depending on the kind of the RBF adopted. A radial basis fit exists if the coefficients γ_i and the weights of the polynomial can be found such that the desired function values are obtained at source points and the polynomial terms gives no contributions at source points, i.e., $s(x_{k_i}) = g_i, 1 \leq i \leq M$, $\sum_{i=1}^{M} \gamma_i q(x_{k_i}) = 0$ for all polynomials q with a degree less than or equal to that of polynomial h. The minimal degree of h depends on the choice of the RBF. A unique interpolant exists if the basis function is a conditionally positive definite function [5]. If the RBFs are conditionally positive definite of order $m \leq 2$ [1], a linear polynomial can be used $h(x) = \beta_1 + \beta_2 x + \beta_3 y + \beta_4 z$. The subsequent development assumes that the aforementioned hypothesis is valid. The γ and β coefficients are obtained by solving the system for each of the three spatial directions

$$\begin{pmatrix} \mathbf{M} & \mathbf{P} \\ \mathbf{P}^{\mathrm{T}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \gamma \\ \beta \end{pmatrix} = \begin{pmatrix} \mathbf{g} \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 1 & x_{k_1} & y_{k_1} & z_{k_1} \\ 1 & x_{k_2} & y_{k_2} & z_{k_2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{k_M} & y_{k_M} & z_{k_M} \end{pmatrix}, \tag{8}$$

where g are the known values at the source points and $\mathbf{M}$ is the interpolation matrix defined by computing all the radial interactions between source points, $M_{ij} = \varphi \left(\left\| x_{k_i} - x_{k_j} \right\| \right), 1 \leq i \leq M, 1 \leq j \leq M$. P is a constraint matrix that arises to balance the polynomial contribution.

The RBF method has several advantages that make it very attractive for mesh smoothing. The key point is that, being a meshless method, only grid points are moved, regardless of which elements are connected to them; this makes the method suitable for parallel implementation. Though meshless, the method is able to exactly prescribe known deformations onto the surface mesh: this is achieved by using all the mesh nodes as RBF centres with prescribed displacements, including the zero field to guarantee that a surface is left untouched by the morphing action.



(a) DrivAer rear window (b) wing-fuselage junction

Fig. 2 Example of RBF points arrangement for the definition of two shape parameters: **a** the height of the rear window of the DrivAer car model is controlled by a cluster of RBF control points; a Box Encapsulation is used to limit the effect of the displacement in the vicinity of the windows; **b** a similar set-up is used to define the deformation of the wing-fuselage junction close to the leading edge

The industrial implementation of RBF morphing poses two challenges: the numerical complexity related to the solution of the RBF problem for a large number of centres and the definition of suitable paradigms to effectively control shapes. RBF Morph deals with both, as it comes with a fast RBF solver capable of fitting large datasets (hundreds of thousands of points in a few minutes) and with a suite of modeling tools allowing the user to set-up shape modifications in an expressive and flexible way. This performance is due to iterative solutions, the Fast Multipole Method and Partition of Unity, as well as shared memory parallelism, the efficiency of which depend on the problem size and has been proven up to 48 cores. RBF Morph allows to extract control points from surfaces and edges, to put points on primitive shapes (boxes, spheres, and cylinders) or to specify them directly by individual coordinates. Two shape modifications used in this study are represented in Fig. 2.

Once the adjoint fields are available, it is possible to compute the SD w.r.t. shape parameters defined by the morphing tool. To take into account the nonlinear fashion of the morphing field, the grid sensitivities are generated through second-order FD of the morphing field around the current design point. It is worth noting that in case the *FI* formulation is used, grid sensitivities are required for the entire grid while, for the *SI* or *E-SI* formulations, these are needed only at the deformable boundaries.

4 Optimization Algorithm

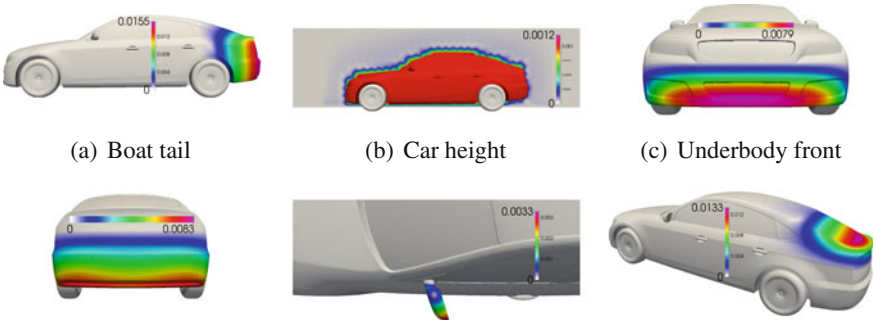
The gradient-based algorithm used to minimize the objective function consists of the following steps: (1) Define the shape modification parameters, and compute the grid sensitivities through FD. These are kept fixed during the entire optimization loop. (2) Solve the flow equations. (3) Compute J. (4) Solve the adjoint equations. (5) Compute the SD. (6) Update the design variables by using a descent method. (7)

Morph the parameterized surface and displace the interior mesh nodes. (8) Unless the stopping criterion is met, go to step 2.

Apart from step 1, which runs in serial using the RBF Morph tools, the rest of the steps are executed within a single OpenFOAM®-based executable, which performs all tasks in parallel. Steps 2 and 4 are the most costly parts of the algorithm, since they require the solutions of PDEs, and have approximately the same cost. If the *FI* formulation is chosen, SD computation can become expensive as well, in case of many (of the order of hundreds) design variables. In case the *E-SI* formulation is used, the cost of computing SD is negligible, as is the cost of the remaining steps.

5 Applications

In the first application, the drag minimization of the DrivAer car model, developed by the Institute of Aerodynamics and Fluid Mechanics of TU Munich [3], is studied. Specifically, the fast-back configuration with a smooth underbody, with mirrors and wheels (F_S_wm_ww) is used. Following the standard practice of the automotive industry, wall functions are used to effect closure on a grid of about 3.8 million cells. Six design variables are defined in total. The part of the car surface parameterized by each of them and the corresponding grid sensitivities are depicted in Fig. 3. The convergence of the optimization algorithm using the *FI* and *SI* adjoint formulations, along with a comparison of the pressure fields between the initial and optimized geometries, is illustrated in Fig. 4. Lowering the rear windshield, creating a spoiler at the end of the trunk and a boat-tail-shaped rear side led to increased pressure on the rear part of the car and, thus, lower drag.



(a) Boat tail (b) Car height (c) Underbody front

(d) Underbody back (e) Mirror rotation (f) Rear window

Fig. 3 DrivAer optimization: Six design parameters are used to morph different parts of the car, by controlling: **a** the boat tail, **b** the car height, **c** the front bumper, **d** the rear bumper, **e** the mirror shape, **f** the rear window shape. In color, one may see $|\delta \mathbf{x}/\delta b_n|$. By computing SD on the initial geometry, the variables depicted in (**a**), (**c**), (**d**) and (**f**) are identified as the ones with the highest optimization potential

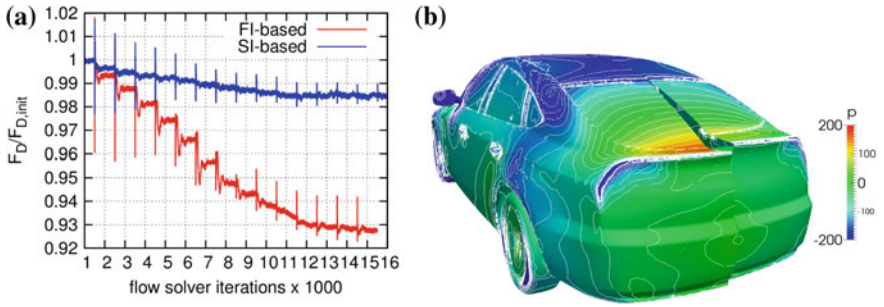


Fig. 4 DrivAer optimization: Left: Evolution of the normalized drag in terms of the number of iterations of the flow solver, following the *FI* and *SI* formulations. In each optimization cycle, the flow solver runs for 1000 iterations. Kinks in the drag value indicate the first iterations after each shape update. With the *FI* formulation, a drag reduction of 7% was achieved, whereas the *SI* gave no more than 1.5% at approximately the same CPU cost. Right: initial (starboard) and optimized (port) (with the *FI* formulation) geometries, colored based on the surface pressure

The second application is concerned with the shape optimization of a glider plane targeting the maximization of the lift-to-drag ratio. The Reynolds number is $Re = 1.55 \times 10^6$ based on the wing chord, the Spalart–Allmaras turbulence model is used, the mesh consists of about 4.7 million cells and the far-field flow angle is 10°. The geometry is parameterized using four RBF-based design variables, depicted in Fig. 5. The FI adjoint formulation is used, the convergence of the optimization

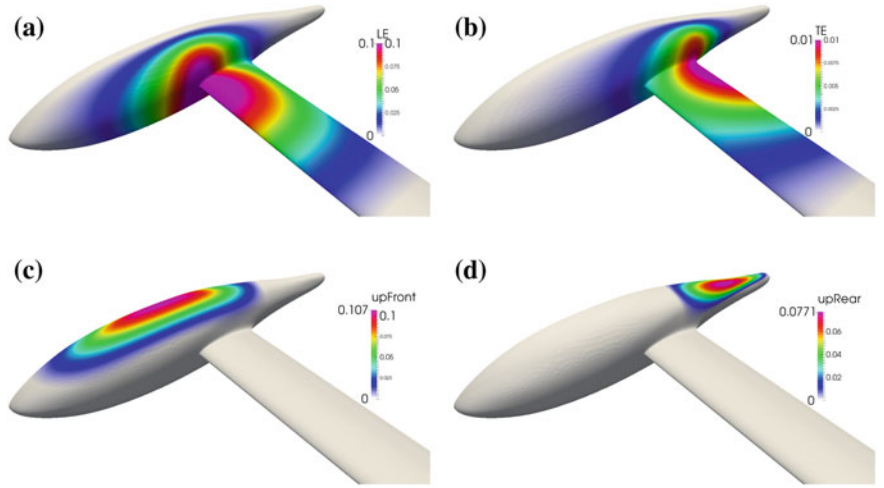


Fig. 5 Glider optimization: the grid sensitivities magnitude for the four design variables. (**a**) and (**b**) parameterize the wing-fuselage junction close to the leading and trailing edges, while (**c**) and (**d**) affect the upper glider surface. All design variables are bounded in order to prevent the generation of non-manufacturable solutions

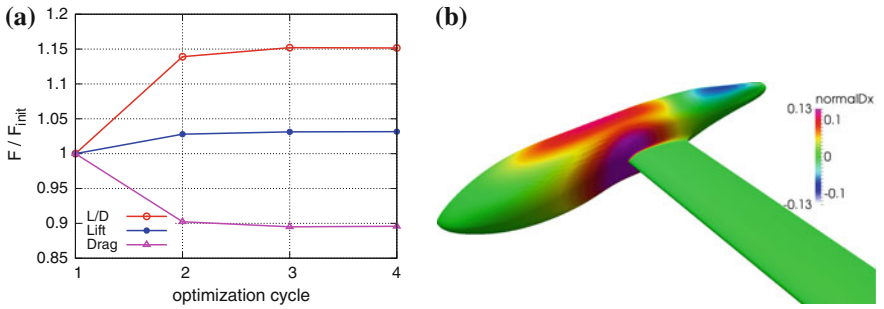


Fig. 6 Glider optimization: **a** convergence of the lift-to-drag ratio (L/D), along with the lift and drag values, normalized with the ones obtained using the initial geometry. A 15% lift-to-drag increase is observed in four optimization cycles by mainly reducing the drag value and slightly increasing lift, **b** the optimized glider geometry, colored based on the projection of the cumulative surface displacement to the surface normal vectors of the initial geometry. Positive numbers indicate an inward displacement while negative ones, an outward movement

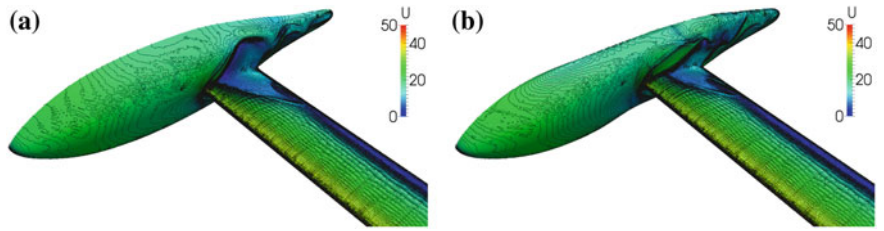


Fig. 7 Glider optimization: near-wall velocity isolines, plotted on the glider surface for the **a** initial and **b** optimized geometries. It can be observed that the low velocity area close to the trailing edge has been considerably reduced

algorithm is shown in Fig. 6a and the optimized geometry is illustrated in Fig. 6b. In Fig. 7, the near-wall velocity isolines are plotted on the glider surface for the initial and optimized geometries.

6 Conclusions

An in-house developed OpenFOAM®-based continuous adjoint solver and an RBF-based morpher, combined into an automated optimization software in the context of a research project funded by the EU, were used as the constituents of a gradient-based optimization algorithm. Two optimization problems of automotive and aerospace engineering were studied, giving a more than 7% drag reduction in the DrivAer case and a 15% lift-to-drag ratio increase in the glider case.

References

1. A. Beckert and H. Wendland. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Constructive Approximation*, 5(2):125–134, 2011.
2. M.E. Biancolini. Mesh morphing and smoothing by means of radial basis functions (RBF): A practical example using Fluent and RBF Morph. In *Handbook of Research on Computational Science and Engineering: Theory and Practice (2 vol)*, pages 347–380, 2011.
3. A. Heft, T. Indinger, and N. Adams. Experimental and numerical investigation of the DrivAer model. In *ASME 2012, Symposium on Issues and Perspectives in Automotive Flows*, pages 41–51, Puerto Rico, USA, 8-12 July 2012.
4. I.S Kavvadias, E.M. Papoutsis-Kiachagias, and K.C. Giannakoglou. On the proper treatment of grid sensitivities in continuous adjoint methods for shape optimization. *Journal of Computational Physics*, 301:1–18, 2015.
5. C. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2(1):11–22, 1986.
6. D.I. Papadimitriou and K.C. Giannakoglou. A continuous adjoint method with objective function derivatives based on boundary integrals for inviscid and viscous flows. *Journal of Computers & Fluids*, 36(2):325–341, 2007.
7. E.M. Papoutsis-Kiachagias and K.C. Giannakoglou. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: Industrial applications. *Archives of Computational Methods in Engineering*, 23(2):255–299, 2016.
8. A.S. Zymaris, D.I. Papadimitriou, K.C. Giannakoglou, and C. Othmer. Continuous adjoint approach to the Spalart-Allmaras turbulence model for incompressible flows. *Computers & Fluids*, 38(8):1528–1538, 2009.

Development of a Combined Euler-Euler Euler-Lagrange Slurry Model

Alasdair Mackenzie, M. T. Stickland and W. M. Dempster

Abstract There has been a significant amount of work on modelling erosion caused by slurries, however, these studies are normally focused on low concentrations. The reason for this is usually that dense slurries are too computationally expensive to model in the Euler-Lagrange frame. This presentation suggests a novel solution for reducing computational effort using OpenFOAM to combine two solvers. The two phases of the bulk flow are modelled, partially in the Eulerian-Eulerian reference frame, and partially in the Eulerian-Lagrangian frame. The method aims to increase computational efficiency, but still keep the necessary particle impact data at the wall required for erosion modelling. The new model consists of splitting the domain into two regions and using patch interpolation to couple them together. The particles are then injected into the second region by using the values of the second Eulerian phase from the first region. The values of the second Eulerian phase are written at every time step to a lookupTable, enabling the solver to be used in conjunction with geometry changes, etc., as in Lopez's work (Lopez in LPT for erosion modelling in OpenFOAM 2014, [1]). If the process can be validated, it provides a promising step towards modelling dense slurry erosion.

A. Mackenzie (✉)
Weir Advanced Research Centre, University of Strathclyde, Glasgow G1 1RD, UK
e-mail: alasdair.mackenzie.100@strath.ac.uk

M. T. Stickland · W. M. Dempster
Mechanical and Aerospace Engineering Department,
University of Strathclyde, Glasgow G1 1RD, UK
e-mail: matt.stickland@strath.ac.uk

W. M. Dempster
e-mail: william.dempster@strath.ac.uk

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_6

1 Introduction

Erosion caused by dense slurry (liquid/solid) flows is a common problem, and a significant amount of work has been published by various authors [2–7]. The aim of this piece of work is to find a more efficient way of modelling the fluid and particles, so that the erosion modelling times can be improved without sacrificing accuracy.

The modelling of erosion of slurries using computational fluid dynamics (CFD) can be grouped into two categories: Euler-Euler (EE) and Euler-Lagrange (EL). The Euler-Euler technique models both the particles and fluid as continuous phases, and since it is volume-averaged it is not computationally expensive. Euler-Lagrange, on the other hand, models particles as individual entities (or parcels), having their own velocity vectors. This means the particles have individual vector values at the wall, which is substantially better for erosion modelling than an averaged cell centre value. The downside of Euler-Lagrange is that it is more computationally demanding, with varying degrees; depending on one-, two- or four-way coupling. This can lead to very large modelling times, which are unsuitable for most engineering applications. A similar method of using a hybrid model for erosion modelling has been proposed by Messa et. al [8], and by other authors to varied applications [9–11]. However, it seems that all these papers view the addition of particles as a post-processing step, rather than a runtime feature. This means that the models are one-way instead of two-way coupled, as they are in this piece of work. One exception is the very recent paper by Yu [12], in which the volume of fluid (VOF) method is coupled with Lagrangian particles in the application of diesel spray modelling.

The method presented here proposes combining the two types of modelling approaches into a hybrid model, which should enable shorter computational times, but also, crucially, give the same results required for erosion calculations.

Outline of paper

The following will describe the creation of a new hybrid model. The mesh/domain used was for development purposes, and therefore had large cells to keep solution times down. Realistic results were not a major consideration at this development stage, however, the mesh is good enough to compare the results of the various models with each other.

Despite there being an abundance of research and work into erosion modelling with CFD [6, 13–15], OpenFOAM is rarely used in the published literature. This is not due to OpenFOAM's results being less reliable; as the paper by Mackenzie [16] shows, OpenFOAM gives similar results to other commercial CFD packages. OpenFOAM was used in this study because the authors recognise its potential for further unrestricted development in the future, a potential that would not be possible with commercial alternatives.

It should also be mentioned that there is an online tutorial on the Chalmers University website [17] explaining in depth how the solver in this work was created.

2 Current OpenFOAM Models

OpenFOAM 3.0.x was chosen to develop the hybrid model for its ease of development, and for the fact that it already has multiple multiphase CFD models available. Solvers were chosen according to their suitability for modelling dispersed two-phase (liquid/solid) flows, with the outlook to model dense slurries. For the EL solver, Lagrangian particles were used from the `DPMFoam` solver (governing equations can be read in the paper by Li et al. [18]), which has been successfully used in other papers for erosion modelling [19]. The discrete particle method (DPM) has successfully been applied in various other applications for dispersed particulate flows [20–25].

For the EE solver, `reactingTwoPhaseEulerFoam` was chosen (a detailed tutorial on this solver by Phanindra can be read in [26] and the governing equations are in the paper by Bhusare [27]). This is a solver for two fluids with a common pressure but otherwise separate properties, and it was chosen because the energy equation can be turned off, making the problem incompressible with no heat transfer, which is as close to a slurry as is possible with a standard model. It should be noted that this approach could be applied to any other multiphase model, like `twoPhaseEulerFoam`, for example. The major drawback of the Eulerian models in regard to erosion modelling is that there are no particles modelled, and therefore no impact data at the wall.

3 Solver Development

To begin the second Eulerian phase conversion into particles, an interface or a region was required, as in Yu's paper [12]. An interface was chosen for this project, as it seems to be the simplest option for the design engineer to implement in a practical application. This interface should not affect the first phase, and not significantly affect the overall resulting flow field. Therefore, the first step of development was to create a domain, split it into two parts with the interface connecting the two, and then make it run as if it was one single domain. This was done with the EE model running through the two regions first. Once this was complete, the second (solid) EE phase was turned off in the second region and Lagrangian particles added instead.

There are also some conceptual issues that should be highlighted. The main one is that there is a transfer from a volume-averaged model to a particulate model, or in other words, a transfer from less information to more information. The Euler cell has one velocity vector and a volume fraction, which are both used to calculate the number of particles to be injected with a certain velocity. For example, in a real-life problem, two particles could be travelling through the Euler cell parallel to each other, but at opposite sides of the cell. In the Euler CFD model, this would be represented by *one* velocity vector pointing in the direction of travel; thus, when the two Lagrangian particles are added, they will originate from the cell centre with the same speed and

direction. There is also a potential problem with mass continuity due to rounding errors, since the solver cannot inject a fraction of a particle.

A new solver was created by using the EE solver as a template. The following attributes were added to the main .C file to create the hybrid model.

3.1 Mesh/Baffles/Regions

A simple 3D domain was created with `blockMesh`, and is shown in Fig. 1. A square pipe bend is simple enough to converge, but complex enough to allow the user to see if the solver is working as it should. The inlet is at the top, outlet on the bottom right, and the other external faces are all walls. The bottom wall had a post-processing patch on it called 'bottom' to allow for particle data acquisition.

Baffles are useful because boundary conditions can be set on them, providing a suitable injection site for the particle injector and region interface. An .stl surface

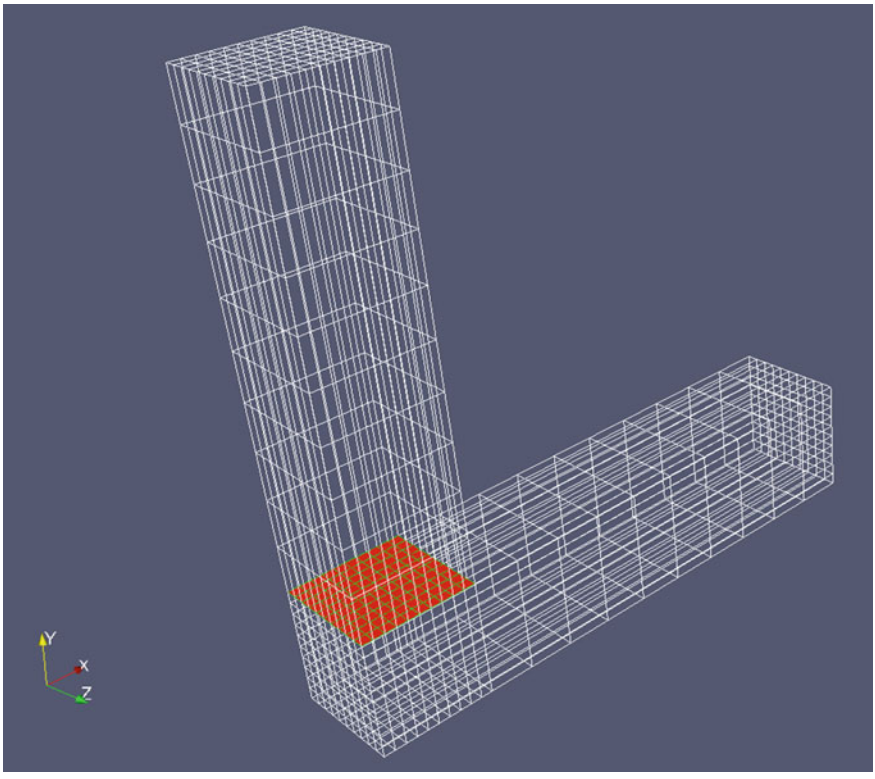


Fig. 1 Domain showing the mesh and the baffle. The pipe's dimensions are 10 × 10 mm, with each piece being 100 mm long. The location of the baffle is shown in red

was drawn and the `createBaffles` command was used to create the red baffle, which can be seen in Fig. 1.

The hybrid model also has the potential to be two-way coupled between EE and EL, and therefore regions were created with the `splitMeshRegions` command. Regions are created above and below the baffle's location, with the aim of Region 0 having EE and Region 1 having EL.

3.2 Interpolation

`patchToPatchInterpolation`, an OpenFOAM utility, was used to interpolate the results from Region 0 to Region 1. In this simple test case, the topology above and below is exactly the same, so the utility is superfluous here, however, the function is added to facilitate future developments in which topologies might not be identical. When baffles are created, a 'master' and 'slave' patch are formed, master on top (Region 0) and slave on bottom (Region 1). Figure 2 shows the iterative loop, and this process repeats every timestep.

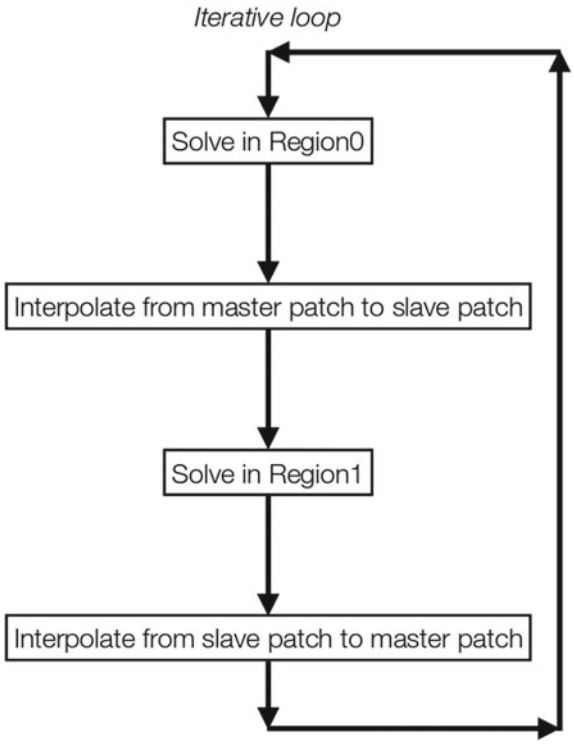


Fig. 2 Iterative loop using patchToPatchInterpolation [28]

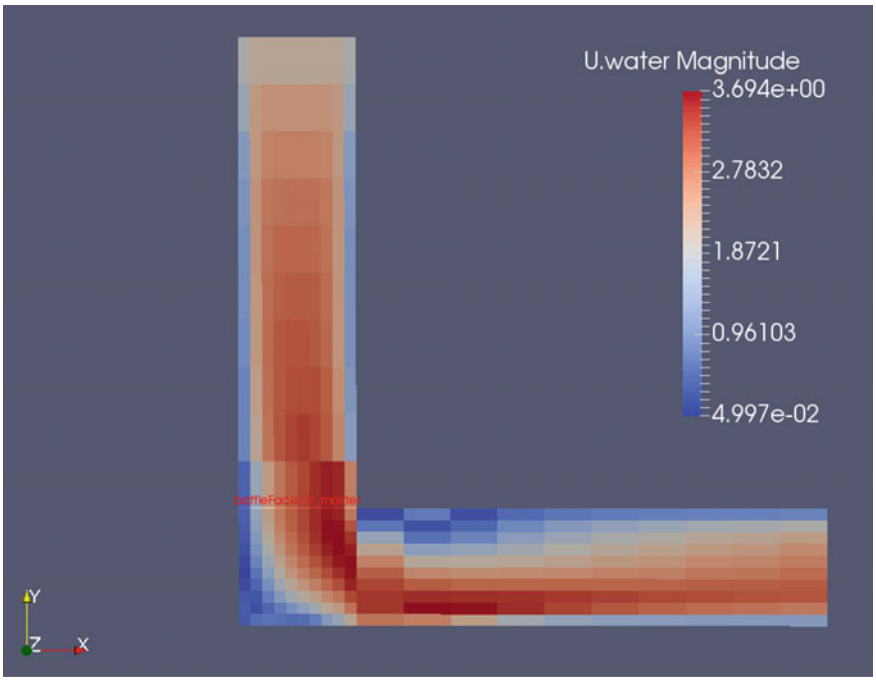


Fig. 3 Velocity contours of 1st Eulerian phase: cell centre values

The code for solving one region at a time was taken from `chtMultiRegionFoam`, and modified to suit the case. This is a common method for developing in OpenFOAM: taking an existing code and modifying it to suit. One major benefit is that the user does not need to be fluent in the programming language to be able to modify or manipulate solvers.

Before any particles were added, the solver was tested as a pure EE model to find out if the interpolating routines were working as they should be. Figure 3 shows the velocity of water cell centre values before any particles have been added. The red writing indicates the location of the interpolating routines and the baffles, and it can be seen that the contour values of the cells on either side of the baffle are the same: this is discussed further in Sect. 4.1. This is due to the interpolating code, and steps should be taken to make the cells as small as possible when located next to the baffles so as to reduce errors downstream. The contour plot is comparable to the simulation result when the baffles/regions were not in place, therefore the baffles are not having a significant negative effect on the results.

3.3 Addition of Particles to the Solver

Now that there is verification that the regions and interpolating routines are working, Lagrangian particles can be added to Region 1, and the second EE phase can be turned off (if the second phase was not turned off, there would be a mass continuity problem). The second phase is removed from Region 1 by setting the volume fraction equal to zero on the slave patch (the 'inlet') in the main .C file, ensuring that the interpolating routines do not overwrite it. The other boundary conditions should be set to 'fixed value' on the slave patch, so that they can be overwritten by the interpolating routines.

There are multiple standard injection methods available in OpenFOAM. A slightly modified version of `kinematicLookupTableInjection` was used, as it allows the user to specify particle locations and velocities. The lookupTable does not allow the user to control how many particles are injected from each site, therefore the injection model was edited to allow this. Another column was added to the lookupTable, which was called `numParcels`. This is defined in Eq. 1 as

$$Num_p = \frac{(\alpha_p * A_{cell} * V_{normal})}{(Vol_p * N_{pp} * \Delta T^{-1})}, \tag{1}$$

where α_p is the volume fraction of the particulate phase, A_{cell} is the area of one cell, V_{normal} is the normal velocity component of the particulate phase, Vol_p is the volume of one particle, N_{pp} is the number of particles/parcel (user-defined), and ΔT^{-1} is the number of timesteps per second. Therefore, the number of parcels injected is based on the volume flowrate from the first region.

The lookupTable is written at every timestep using the `OFStream` application, with the values in the table taken from the master patch. Each cell on the master patch has one row; since there are 100 cells in this case, there are 100 rows.

(x, y, z) coordinates are written from the cell centre locations; (u, v, w) velocity vectors are written from the Eulerian velocity field; diameter, density and mass flow rate are constant (mass flow rate is unused); and the number of parcels to inject is based on the volume flow rate and the phase fraction of each cell. It is here that rounding to integers could contribute to mass continuity issues. As a way to save on computational resources, *Parcels*, which are groups or clusters of particles, are injected instead of individual particles; the size of these is user-defined.

With the lookupTable defined, the '`kinematicCloud.evolve();`' command from `DPMFoam` is added and the second phase of EE is turned off in Region 1 to preserve mass continuity. Figure 4 shows the working model with a 2D slice of velocity magnitude of the 2nd Eulerian phase, and the Lagrangian particles coloured according to their velocity magnitude.

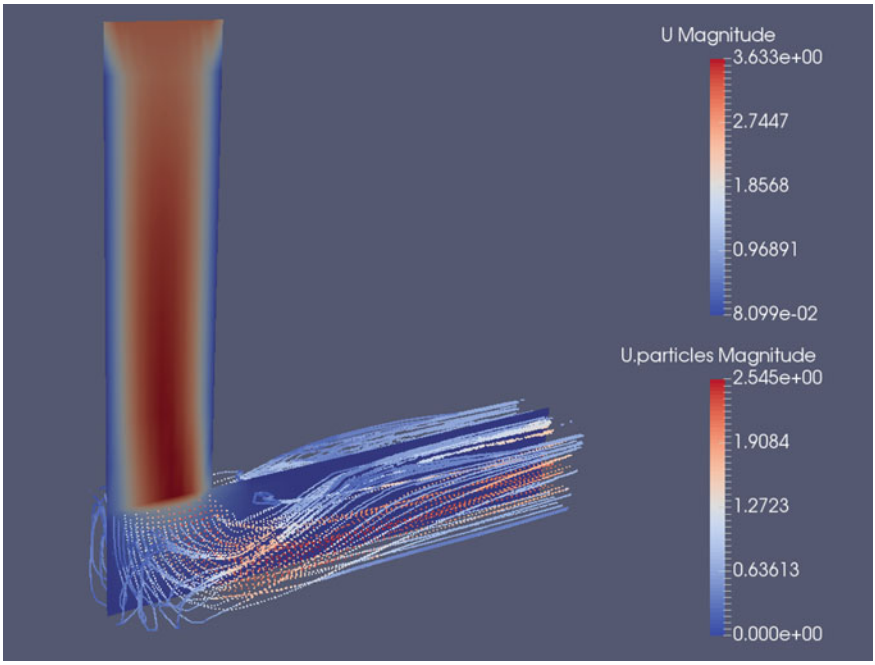


Fig. 4 Hybrid model showing second phase velocity magnitude in the first and second regions as a Eulerian fluid (z-plane slice) and as Lagrangian particles (particle tracks)

4 Initial Test of Model

To test the performance of the hybrid model, it was compared with both a pure EL model, and a pure EE model. All three models were set up as a transient case in the same manner, with the same boundary conditions as far as possible (inlet velocity = 2m/s, diameter of particles = 55 microns, 3 particles/parcel). The injection parameters for the EL case could not be identical to the other two models, as there are no particles in Region 0. Therefore, the particles were injected uniformly over the face of the inlet plane, using the same number of particles as the hybrid model and the same cell centre positions (but on the inlet plane).

Each solver was run from 0 to 0.39 s with the start of injection (SOI) set at 0.29 seconds for the models with particles. With 1% mass concentration, 540 parcels were added to each injection, with a total of around three hundred thousand parcels added. A second test was carried out with 2% mass concentration, this time having 1100 parcels per injection.

Table 1 shows the execution time of the three solvers, giving expected results: the hybrid model is faster than the EL, but slower than the EE. The hybrid model is almost 50% faster than the pure EL model in both cases.

Table 1 Execution time from 0 to 0.39 s: % mass concentration (MC)

Model	Execution time (s)	
	1% MC	2% MC
Hybrid model	225	298
Euler-Lagrange	420	585
Euler-Euler	102	105

Although the chief aim of the model is to be more computationally efficient, there cannot be a large sacrifice of accuracy, therefore other parameters were compared. The first phase (water) was compared between the three models, and the particle impacts were compared among the hybrid and the EL model.

4.1 First Phase Velocity Comparison

The hybrid model was used as a standard and compared against the EE and EL models. The cell centre value contour plots below represent the difference as a percentage, as expressed in Eq. 2, where ΔV_{h-ee} is the difference between the velocity magnitude of the hybrid model and the EE model for the first (fluid) phase

$$\Delta V_{h-ee} = \frac{|U_{hf} - U_{eef}|}{|U_{hf}|} \times 100, \tag{2}$$

where U_{hf} is the velocity of the hybrid model fluid phase and U_{eef} is the EE model's first phase of velocity. The same equation is used for the EL model, but with EL substituted for EE.

Figures 5 and 6 have the percentages clipped to $+/-$ 100%, as there are two cells that are very high/low, causing the contours to be difficult to read for the rest of the domain (see captions for max/min values). The contours show that the majority of the domain is the same among the three models, only varying by around 5%. The positive values indicate where the hybrid model over-predicts, and the negative values where it under-predicts relative to the standard models.

Both errors can be explained by the presence of the baffle/interpolating routine. Figures 7 and 8 show the location of the baffle (with red writing and a thin white line) and the contours of the neighbouring cells. With the EL simulation, which has no baffle present, the values of the cells over the baffle are different, whereas with the hybrid model, they are the same. This is due to the way in which the interpolation works, as the cells that neighbour the baffle always have the same value.

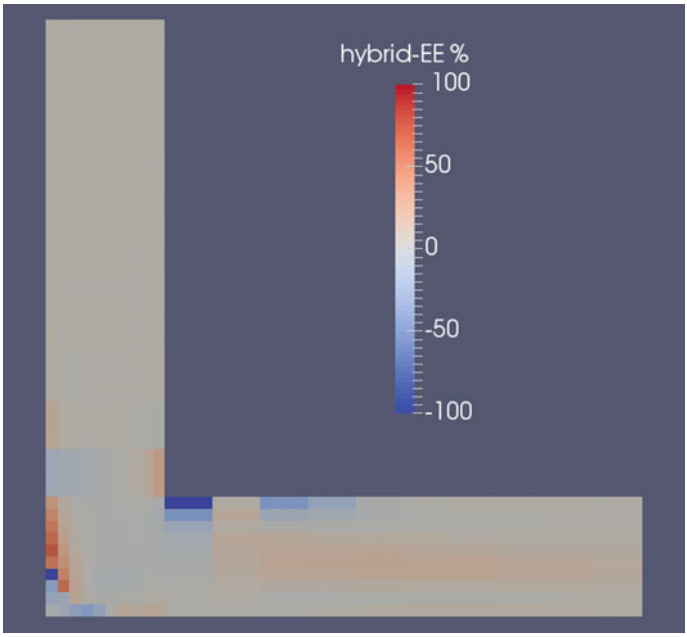


Fig. 5 Hybrid-EE: Contours of Eq. 1. Min = −300%, Max = 70%

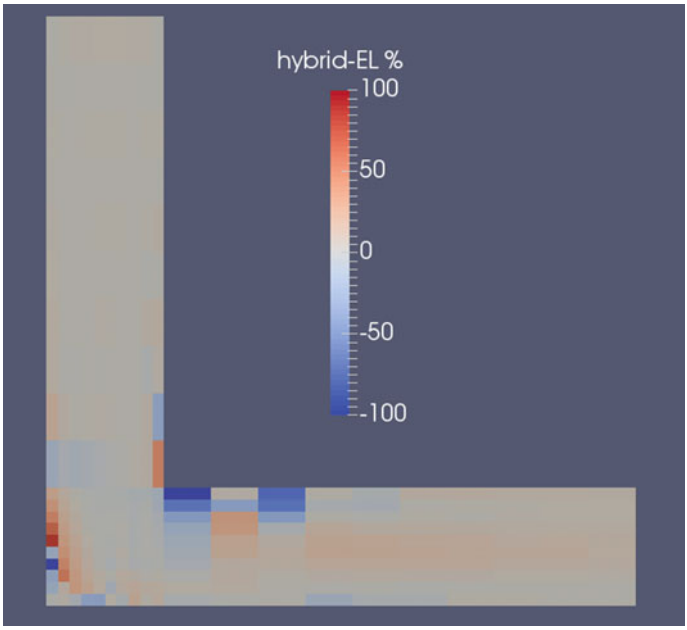


Fig. 6 Hybrid-EL: contours of Eq. 1. Min = −460%, Max = 80%

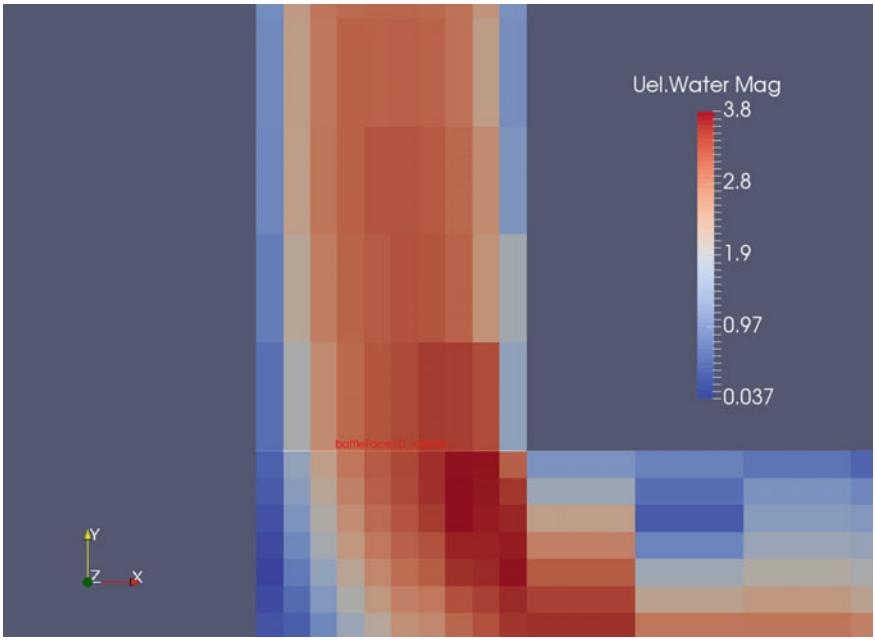


Fig. 7 EL model: no baffle

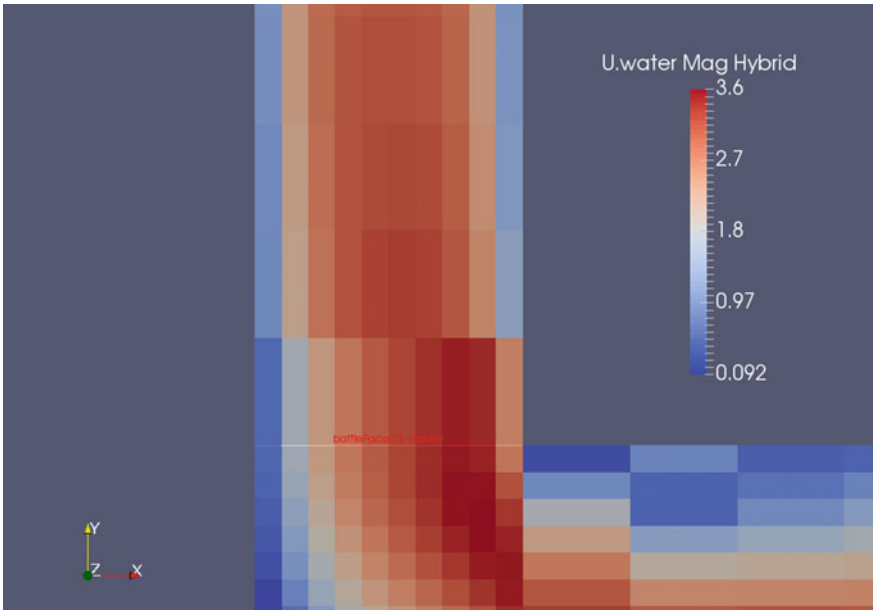


Fig. 8 Hybrid model: with baffle

In a real problem, the error could be reduced by minimising the size of cells close to the baffle, and also by choosing a more suitable location for the baffle. Considering that the largest error is due to these two factors and the remainder of the domain is almost the same as that of the original models, the results are deemed acceptable.

4.2 Particle Comparison

As mentioned, the particle impacts were compared between the pure EL model and the hybrid model.

Figures 9 and 10 show the bottom patch particle impacts; −0.005 m in the Z-direction is the centre of the pipe, and the X-axis only goes up to 0.02 m, as only the first impacts are considered. Both models were run with the dispersion model switched off, and there are 5000 data points in each graph.

The differences in the graphs can be explained by a number of things. Firstly, the hybrid model scatter looks less random, which could be due to the injection site taking place only 10 mm above the surface, as opposed to 100 mm for the EL. Another issue that has been described previously is the attempt to compare similar injections. The hybrid model forces particles to be injected from the cell centres each time, whereas by the time the particles get to the baffle in the EL simulation from the inlet, they could be going through the cells on the baffle at any location. Another consideration is that these graphs are also comparing OpenFOAM's EE and EL models: since the hybrid injection is based on the EE region, if the EE model is wrong, then the injector will be wrong.

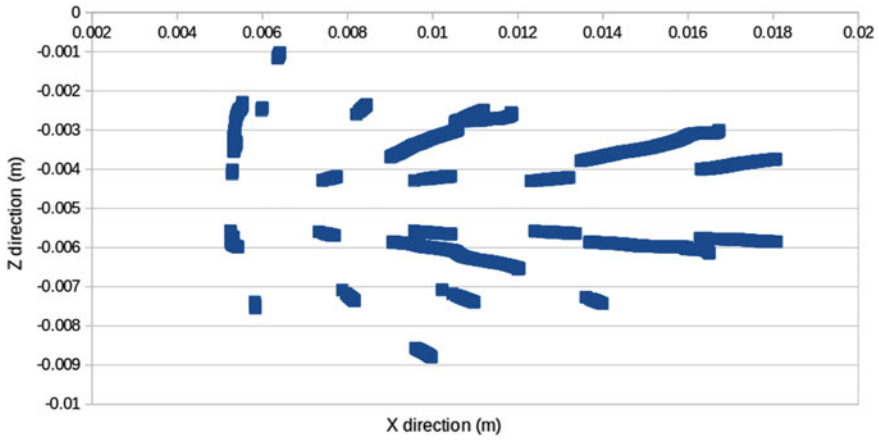


Fig. 9 EL particle impacts

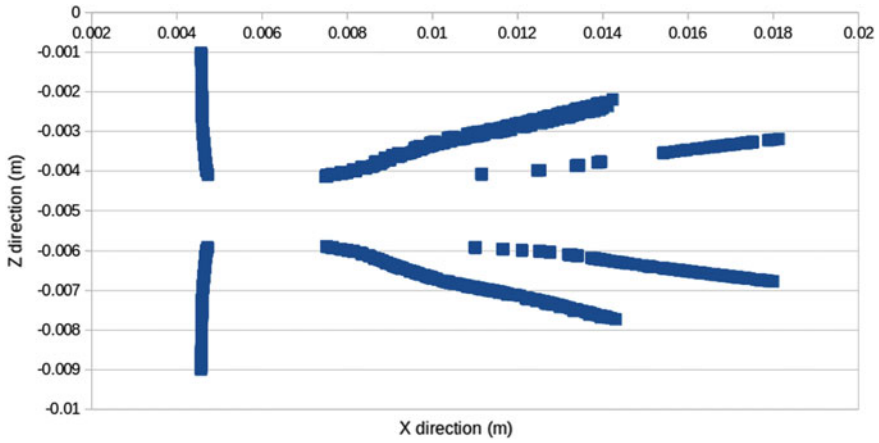


Fig. 10 Hybrid model particle impacts

5 Future Development and Conclusion

Although still in the development stage, this model shows a promising outlook for the future of slurry modelling, as it was twice as fast as the standard EL solver. As mentioned earlier, the lookupTable is written t for every timestep, however, it is only read at the start of the simulation. Although not necessary, there is potential here for future work, as the injector would then be able to cope with large changes in the flow field, e.g., mesh deformation caused by erosion.

The model currently only goes from EE to EL, but in a real application, it would be desirable to go back from EL to EE. Particles remain active in the domain until they exit; however, transferring them back to a Eulerian phase would save on computational time.

The previous section has also mentioned the difficulty in comparing the hybrid model to an EL simulation. There is scope within which to improve the injection model so that it can inject from locations other than the cell centres, eliminating any dependency on mesh density.

This study has shown the potential for the future of dense slurry flow modelling, by creating a new hybrid model that reduces computational time without severely affecting particle impact data, which is required for successful erosion modelling.

References

1. A Lopez. LPT for erosion modeling in OpenFOAM. http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2013/AlejandroLopez/LPT_for_erosionModelling_report.pdf, 2014.
2. Hao Zhang, Yuanqiang Tan, Dongmin Yang, Francesc Xavier Trias, Shengqiang Jiang, Yong Sheng, and Assensi Oliva. Numerical investigation of the location of maximum erosive wear

damage in elbow: Effect of slurry velocity, bend orientation and angle of elbow. *Powder Technology*, 217:467–476, 2012.
3. Chong Y. Wong, Christopher Solnordal, Anthony Swallow, Steven Wang, Lachlan Graham, and Jie Wu. Predicting the material loss around a hole due to sand erosion. *Wear*, 276-277:1–15, February 2012.
4. D.R. Kaushal, T. Thinglas, Yuji Tomita, Shigeru Kuchii, and Hiroshi Tsukamoto. CFD modeling for pipeline flow of fine particles at high concentration. *International Journal of Multiphase Flow*, 43:85–100, July 2012.
5. L J W Graham, D Lester, and J Wu. SLURRY EROSION IN COMPLEX FLOWS : EXPERIMENT AND CFD. (December):1–6, 2009.
6. a. Mansouri, H. Arabnejad, S.a. Shirazi, and B.S. McLaury. A combined CFD/experimental methodology for erosion prediction. *Wear*, 332-333:1090–1097, 2015.
7. Amir Mansouri, Hadi Arabnejad, Soroor Karimi, Siamack A. Shirazi, and Brenton S. McLaury. Improved CFD modeling and validation of erosion damage due to fine sand particles. *Wear*, 338-339:339–350, 2015.
8. Gianandrea Vittorio Messa, Giacomo Ferrarese, and Stefano Malavasi. A mixed Euler-Euler/Euler-Lagrange approach to erosion prediction. *Wear*, 342-343:138–153, 2015.
9. Elke Deux and Martin Sommerfeld. MODELING OF TURBULENT ATOMIZATION COMBINING A TWO-FLUID AND A STRUCTURE FUNCTION APPROACH Elke Deux and Martin Sommerfeld. 16, 2006.
10. Oscar J. Soriano-Palao, Martin Sommerfeld, and Axel Burkhardt. Modelling the influence of the nozzle geometry on the primary breakup of diesel jets. *International Journal of Spray and Combustion Dynamics*, 6(2):113–146, 2014.
11. Rasmus Gjesing, Jesper Hattel, and Udo Fritsching. Coupled Atomization and Spray Modelling in the Spray Forming Process using Open Foam. *Engineering Applications of Computational Fluid Mechanics*, 3(4):471–486, 2009.
12. H Yu, L Goldsworthy, M Ghiji, P A Brandner, and V Garaniya. A parallel Volume of Fluid-Lagrangian Parcel Tracking coupling procedure for diesel spray modelling. 150:46–65, 2017.
13. a. Gnanavelu, N. Kapur, a. Neville, and J.F. Flores. An integrated methodology for predicting material wear rates due to erosion. *Wear*, 267(11):1935–1944, October 2009.
14. a. Gnanavelu, N. Kapur, a. Neville, J.F. Flores, and N. Ghorbani. A numerical investigation of a geometry independent integrated method to predict erosion rates in slurry erosion. *Wear*, 271(5-6):712–719, June 2011.
15. A Mansouri, S A Shirazi, and B S Mclaury. Experimental and numerical investigation of the effect of viscosity and particle size on erosion damage caused by solid particles. *ASME*, pages 1–10, 2015.
16. Alasdair Mackenzie, Alejandro Lopez, Konstantinos Ritos, Matthew Stickland, William Dempster,. A COMPARISON OF CFD SOFTWAREPACKAGES' ABILITY TO MODEL A SUBMERGED JET In *Eleventh International Conference on CFD in the Minerals and Process Industries CSIRO*, pages 1–4, 2015.
17. Alasdair Mackenzie. A hybrid slurry cfd model:euler-euler to euler-lagrange. http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2016/AlasdairMackenzie/tutorial1.pdf, 2017.
18. Linmin Li, Baokuan Li, and Zhongqiu Liu. Modeling of spout-fluidized beds and investigation of drag closures using OpenFOAM. *Powder Technology*, 305:364–376, 2017.
19. Alejandro López, William Nicholls, Matthew T. Stickland, and William M. Dempster. CFD study of Jet Impingement Test erosion using Ansys Fluent® and OpenFOAM®. *Computer Physics Communications*, 197:88–95, 2015.
20. Ordinance Factory Estate, National Mineral, Development Corporation, and Uppal Road. CFD STUDY ON THE EFFECT OF NEAR GRAVITY MATERIAL ON DMC TREATING COAL USING DPM AND ASM MULTIPHASE MODEL. (December):1–7, 2015.
21. Roza Tarpagkou and Asterios Pantokratoras. CFD methodology for sedimentation tanks: The effect of secondary phase on fluid phase using DPM coupled calculations. *Applied Mathematical Modelling*, 37(5):3478–3494, March 2013.

22. Halima Hadžiahmetović, Nedim Hodžić, Damir Kahrimanović, and Ejub Džaferović. COMPUTATIONAL FLUID DYNAMICS (CFD) BASED EROSION PREDICTION MODEL IN ELBOWS. 3651:275–282, 2014.
23. Amir Mansouri, Hadi Arabnejad Khanouki, Siamack A Shirazi, and Brenton S Mclaury. PARTICLE TRACKING VELOCIMETRY (PTV) MEASUREMENT OF ABRASIVE MICROPARTICLE IMPACT SPEED AND ANGLE IN BOTH AIR-SAND AND SLURRY EROSION TESTERS. *Proceedings of the ASME 2016 Fluids Engineering Division Summer Meeting*, pages 1–9, 2016.
24. Jukai Chen, Yueshe Wang, Xiufeng Li, Renyang He, Shuang Han, and Yanlin Chen. Erosion prediction of liquid-particle two-phase flow in pipeline elbows via CFDDEM coupling method. *Powder Technology*, 275:182–187, 2015.
25. Wojciech P. Adamczyk, PaweKozoub, Gabriel Wcel, Adam Klimanek, Ryszard a. Biaecki, and Tomasz Czakiert. Modeling oxy-fuel combustion in a 3D circulating fluidized bed using the hybrid Euler-Lagrange approach. *Applied Thermal Engineering*, 71(1):266–275, 2014.
26. P.Thummala Phanindra. Description of reactingtwophaseeulerfoam solver with a focus on mass transfer modeling terms. http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2016/PhanindraPrasadThummala/reactingTwoPhaseEulerFoam.pdf, 2017.
27. V.H. Bhusare, M.K. Dhiman, D.V. Kalaga, S. Roy, and J.B. Joshi. CFD simulations of a bubble column with and without internals by using openfoam. *Chemical Engineering Journal*, 317:157 – 174, 2017.
28. Alasdair Mackenzie, MT Stickland, WM Dempster. A Combined Euler-Euler Euler-Lagrange Slurry Model. In *Eleventh International Conference on CFD in the Minerals and Process Industries CSIRO*, 2016.

Development of Data-Driven Turbulence Models in OpenFOAM®: Application to Liquid Fuel Nuclear Reactors

M. Tano-Retamales, P. Rubiolo and O. Doche

Abstract The following chapter presents a new approach for the development of turbulent models, with potential application to the design of liquid fuel nuclear reactors. To begin the chapter, the work being carried out at LPSC (Grenoble) for validating the modeling of molten salt coolants is presented, alongside a Backward-Facing Step (BFS) geometry, which will be studied throughout this work. In the subsequent section, various turbulence models are evaluated in the BFS and their advantages and limitations are analyzed, with the conclusion that some improvements in the turbulence modeling are necessary. Therefore, the next section introduces a methodology for developing a nonlinear closure for turbulence models by means of Symbolic Regression via Genetic Evolutionary Programming (GEATFOAM). Then, this new methodology is implemented for direct numerical simulation data of the BFS, obtaining a new nonlinear closure for the standard k–ε model. Finally, the new model is compared against classical turbulence models for the BFS, and, then, the extrapolability of this model is analyzed for available experimental data of an axial expansion in a pipe. Encouraging results are obtained in both cases.

This project has received funding from the Euratom research and training program 2014–2018 under grant agreement No. 661891.
The content of this article does not reflect the official opinion of the European Union. Responsibility for the information and/or views expressed in the article lies entirely with the authors.

M. Tano-Retamales (✉) · P. Rubiolo
University Grenoble Alpes, CNRS, Grenoble INP, LPSC, 38000 Grenoble, France
e-mail: retamales@lpsc.in2p3.fr

P. Rubiolo
e-mail: rubiolo@lpsc.in2p3.fr

O. Doche
University Grenoble Alpes, CNRS, Grenoble INP,
SIMAP, 38000 Grenoble, France
e-mail: olivier.doche@phelma.grenoble-inp.fr

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_7

1 Introduction

Molten salt nuclear reactors are an auspicious new concept in the nuclear industry, because of the innovative design and safety possibilities opened up by the use of a liquid nuclear fuel [1]. In particular, among these reactors, the Molten Salt Fast Reactor (MSFR) is a type of fourth-generation liquid fuel nuclear reactor, which is presently being studied in the framework of the H2020 European project SAMOFAR (2015–2019) [2]. Among the undertakings of this project, the SWATH (Salt at Walls: Thermal excHanges) experiment is being carried out at LPSC in Grenoble. This experiment intends to improve the understanding and predictions of the thermal-hydraulic behavior of high-temperature molten salt internal flows over diverse geometries. The foreseen outcomes of the experiment are validated mathematical models for describing the heat exchange phenomena in such flows.

The experimental layout of SWATH (shown in Fig. 1) includes two tanks filled with a molten salt (FLiNaK), a test section hosted inside a glovebox and instrumentation for performing the measurements. A pressure difference between the two tanks generates a molten salt flow through the test section. No pump is therefore present in the system, reducing the risks of component failure.

The high complexity of molten salt flows demands a parsimonious validation of the proposed mathematical models. Therefore, the substantiation process for the models is executed in two steps: first, the fluid mechanics models are validated without thermal exchanges in the SWATH-W facility (using water), and, second, the thermal exchange models are tested in the SWATH-S facility (using molten FLiNaK). The SWATH-W experiment is a one-to-one scale mockup operating with water, for the purpose of validating the fluid mechanics models. The water flow in the mockup can be generated either by the pressure difference between two tanks or by a centrifugal pump. Precise Particle Image Velocimetry [3] measurements are performed on the test sections (SWATH-W), allowing us to validate the proposed fluid mechanics

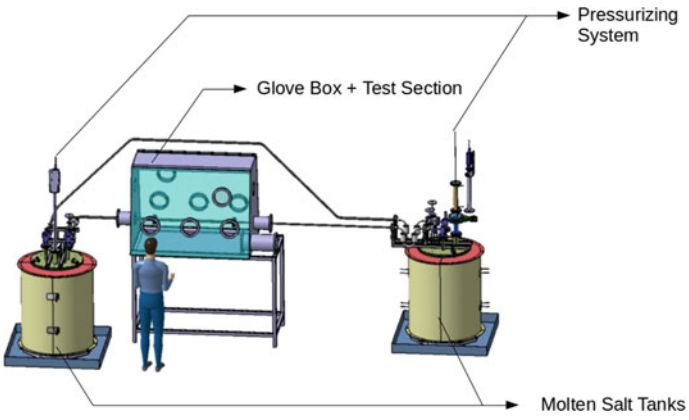


Fig. 1 The SWATH experiment

models (without heat transfer). Afterward, the validated models are completed with the modeling of the thermal exchange phenomena. Finally, by preserving the dynamic similarity, the complete fluid mechanics and thermal exchange models are compared against the data obtained in the main molten salt experiment (SWATH-S).

Computational Fluid Dynamics (CFD) is the name given to the simulation of fluid dynamics phenomena by means of a computer. Different CFD techniques have been proposed for resolving the mean and fluctuating components of the velocity in a turbulent flow. The Direct Numerical Simulation (DNS) technique aims for a complete resolution of both mean and fluctuating components at an expensive computational cost [4]. For the purpose of reducing the computational cost while resolving turbulence fields, Reynolds Average Navier Stokes (RANS) and Large Eddy Simulations (LES) techniques apply *integral filters* to the velocity and model the extra term appearing in the Navier–Stokes as a result of this filtering process [5]. In the RANS method, the velocity is time-filtered, whereas in the LES method, the filters are spatiotemporal, allowing us to control the size of the resolved scales [6].

In order to perform the required multiphysics studies in the MSFR, thermal-hydraulic models should be coupled to neutronics and thermal-mechanics models. Given the reactor geometry and the complexity of the phenomena, only RANS models present a good compromise between computational cost and accuracy [7].

A Backward-Facing Step (BFS) geometry has been selected as one of the sections to be studied in the SWATH experiment, since the flow phenomena in this geometry are representative of the entrance region of the MSFR. This section is interesting, since standard RANS models cannot fully predict the richness of the turbulent structures generated past the BFS [8]. The ultimate modeling objective is to produce an accurate RANS turbulence model, able to predict the bulk velocities in the BFS with an error of less than 5%. Accurate predictions of the molten salt bulk velocities in the MSFR are needed for similarly accurate predictions of the coupled phenomena of the reactor. Currently, Particle Image Velocimetry (PIV) measurements are being carried out for a BFS in the SWATH-W experiment. As the results are not yet available, the methodological approach of the current work is developed according to the precise DNS simulation performed by [9, 10] on a BFS, referred to from now on as DNS data.

2 Application of State-of-the-Art Turbulence Models for the BFS

The studied BFS section is shown in Fig. 2. It consists of a 2D section, with an expansion rate of 2. The mean inlet velocity U is fixed so that the Reynolds number in the inlet throat is $Re = \frac{Uh}{\nu} = 9000$, where $\nu = 10^{-6}\,\frac{\mathrm{m}^2}{\mathrm{s}}$ is the kinematic viscosity (of water at 20 °C).

The BFS geometry was discretized in two dimensions (assuming plane symmetry) with a regular structured quadrilateral mesh, having a maximum aspect ratio of 5 and a maximum skewness ratio of 10^{-6}. The mesh was refined until the predictions in

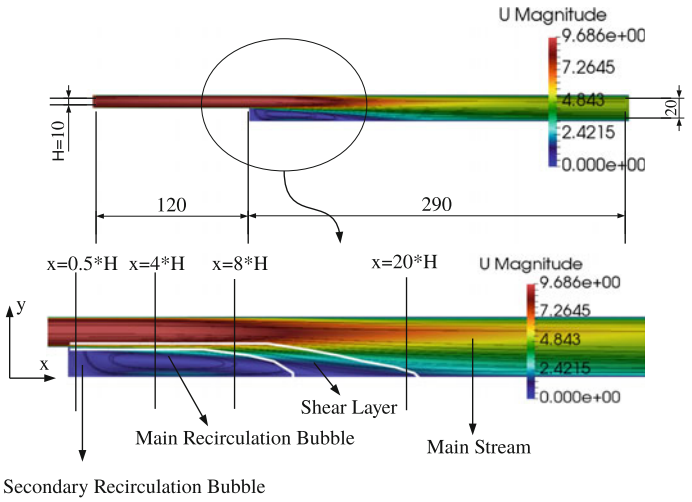


Fig. 2 Top: Dimensions of the BFS section in mm; the stream-wise velocity profile is completely developed before the sudden expansion and in the exit of the domain. Bottom: Three flow zones in the BFS section are distinguished: a main shear layer, a recirculation zone, and a mainstream zone. Also, the vertical lines over which the stream-wise component of the velocity will be studied during the present study are displayed

the bulk velocities were unchanged by further refinements for each turbulence model tested in the section. This standard mesh convergence procedure [11] allows us to obtain results that are independent of the mesh employed, depending only on the applied turbulence model. As expected, the mesh in the region next to the expansion of the BFS (see Fig. 2) was found to be key for an accurate description of the turbulence phenomena in the bulk region of the BFS. This is a consequence of the tripping of the boundary layer arising in this zone. In addition, the predictions in the tripping of the boundary layer were found to be very sensitive to the adopted wall functions. Therefore, since the goal is to make the results only dependent on the turbulence model used, an enhanced wall treatment procedure was imposed in the simulations, i.e., no wall functions were implemented. In this regard, the centers of the mesh cells along the walls of the BFS were adjusted, using a uniform inflation ratio boundary layer, obtaining a dimensionless wall distance (y^+) that varied between 0.5 and 1.5 in these cells. The meshes were generated using the snappyHexMesh utility in OpenFOAM® [12].

An incompressible and isothermal flow in a backward-facing step can be mathematically described by the incompressible Navier–Stokes equations, asserting the conservation of mass and linear momentum. Assuming a description in an Eulerian inertial reference frame, without exterior surface or body forces applied to the fluid, these equations can be read as

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

$$\frac{\partial}{\partial t}\mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla \frac{p}{\rho} + \nu \nabla^2 \mathbf{u}, \tag{2}$$

where $\mathbf{u}(\mathbf{x}, t)$ is the velocity, $p(\mathbf{x}, t)$ is the pressure, ρ is the density, and ν is the kinematic viscosity. When measuring the velocities at a point in a turbulent flow, it is found that the velocity field can be decomposed into a mean and fluctuating component $\mathbf{u}(\mathbf{x}, t) = \overline{\mathbf{u}(\mathbf{x}, t)} + \mathbf{u}(\mathbf{x}, t)'$ [13].

By applying these filtering techniques to the Navier–Stokes equations (2), the following set of filtered equations is obtained:

$$\nabla \cdot \overline{\mathbf{u}} = 0, \tag{3}$$

$$\frac{\partial}{\partial t}\overline{\mathbf{u}} + \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} = -\nabla \frac{\overline{p}}{\rho} + \nu \nabla \cdot (\nabla \overline{\mathbf{u}} - \boldsymbol{\tau}^{\mathbf{f}}). \tag{4}$$

Comparing Eqs. 2 and 4, besides the overbar in the variables indicating its mean component, the difference arrives in a newly introduced term $\boldsymbol{\tau}^{\mathbf{f}}$ that results from the filtering process. For RANS models, $\boldsymbol{\tau}^{\mathbf{f}} = \overline{\mathbf{u}'\mathbf{u}'}$, and this is called the Reynolds Shears Stress (RSS) Tensor. For LES models, $\boldsymbol{\tau}^{\mathbf{f}} = \boldsymbol{\tau}^{\mathbf{r}}$, and this is usually called the residual stress tensor or Sub-Grid Scale (SGS) stress tensor.

For RANS models, the RSS tensor is symmetric and has six independent components. The simpler models for the RSS tensor are the linear eddy viscosity models, wherein it is assumed that $\overline{\mathbf{u}'\mathbf{u}'} \approx \nu_t \mathbf{S}$, where $\mathbf{S} = \frac{1}{2}(\nabla \overline{\mathbf{u}} + \nabla \overline{\mathbf{u}}^T)$ is the strain rate tensor and ν_t is the turbulent viscosity (assumed to be a function of the solved turbulence variables k and ε). One of the models studied in the present work is the k–ε model [14] in which $\nu_t = C_\mu \frac{k^2}{\varepsilon} f_\mu \mathbf{S}$, where $k = \frac{1}{2} tr(\overline{\mathbf{u}'\mathbf{u}'})$ is the turbulent kinetic energy, ε is the specific dissipation rate of turbulent kinetic energy $\varepsilon = \nu \overline{\nabla \mathbf{u}' \cdot \nabla \mathbf{u}'}$, and f_μ is a Van Driest-like damping function that reduces the turbulent viscosity value close to the walls [15]. Specific transport equations are derived for k and for ε, introducing closure coefficients when approximations are done (for specific details, see [16]). The introduction of f_μ into the model, as well as the corrections in the modeling of k and ε close to solid surfaces, is known as *standard wall functions*. However, when performing enhanced wall treatment, wall corrections are not introduced, since the turbulent phenomena next to the walls is considered to be readily resolved by the fine mesh next to the walls (i.e., $f_\mu = 1$). The k–ε model is not unique, since several other *linear eddy viscosity* models exist. In particular, the Wilcox k–ω model [17], available in OpenFOAM®, was also considered during the present analysis.

An important limitation of the linear eddy viscosity models is that the tensorial character of the RSS is solely dependent on the instantaneous strain rate and, therefore, the history of the development of turbulent stresses in the fluid field is not resolved. Consequently, the assumption of linear viscosity is misled in the BFS case, wherein the richness of turbulent structures in the flow is generated by a vortex rolling mechanism in the shear layer soon after the tripping of the boundary layer [18].

Therefore, the turbulence field (and hence the RSS tensor) at a given point will be strongly dependent on the upstream flow or *fluid history*.

An immediate improvement is proposed by the nonlinear eddy viscosity models. First, the RSS tensor is divided into its isotropic and non-isotropic components $\overline{\mathbf{u}'\mathbf{u}'} = \frac{2}{3}k\mathbf{I} + k\mathbf{a}$, where the tensor describing the non-isotropic component $\mathbf{a}$ is known as the anisotropy tensor. Following the previous definition, the anisotropy tensor is traceless $tr(\mathbf{a}) = 0$ and symmetrical $\mathbf{a} = \mathbf{a}^T$. Based on these arguments, and introducing the vorticity tensor as $\mathbf{W} = \frac{1}{2}(\nabla\overline{\mathbf{u}} - \nabla\overline{\mathbf{u}}^T)$, the modeling of the anisotropy tensor based on the k–ε model can be constructed to third orders as follows:

$$\mathbf{a} = \sum_{i_1}^{n} C_i \mathbf{T_i}, \tag{5}$$

$$\texttt{First-order terms:}\mathbf{T_1} = C_1 \frac{\nu_t}{k}\mathbf{S}, \tag{6}$$

$$\texttt{Second-order terms:}\mathbf{T_2} = C_2 \frac{\nu_t}{\varepsilon}(\mathbf{SW} - \mathbf{WS}) \tag{7}$$

$$\mathbf{T_3} = C_3 \frac{\nu_t}{\varepsilon}(\mathbf{S}\cdot\mathbf{S} - \frac{1}{3}\mathbf{S}:\mathbf{S})\mathbf{I}\mathbf{T_4} = C_4 \frac{\nu_t}{\varepsilon}(\mathbf{W}\cdot\mathbf{W} - \frac{1}{3}\mathbf{W}:\mathbf{W}\mathbf{I}),$$

$$\texttt{Third-order closure:}\mathbf{T_5} = C_5 \frac{\nu_t k}{\varepsilon^2}(\mathbf{S}\cdot\mathbf{SW} - \mathbf{WS}\cdot\mathbf{S})$$

$$\mathbf{T_6} = C_6 \frac{\nu_t k}{\varepsilon^2}(\mathbf{W}\cdot\mathbf{WS} + \mathbf{SW}\cdot\mathbf{w} - \frac{2}{3}\mathbf{S}:\mathbf{W}\cdot\mathbf{W})$$

$$\mathbf{T_7} = C_7 \frac{\nu_t k}{\varepsilon^2}(\mathbf{SS}:\mathbf{S}), \mathbf{T_8} = C_8 \frac{\nu_t k}{\varepsilon^2}(\mathbf{SW}:\mathbf{W}). \tag{8}$$

Since the model for the anisotropic tensor includes terms up to the third order, it is said to be a cubic order closure. During the solution process, the k–ε equations are solved at each iteration and, subsequently, ν_t and $\mathbf{a}$ are computed. The near wall treatment in the simulations could be done identically to that of the k–ε model. However, for the reasons explained earlier in the text, no special wall treatment is introduced when doing enhanced wall treatment. For closing the set of equations, this model requires the empirical fitting of 13 coefficients. The nonlinear cubic model proposed by Craft et al. [19] was implemented in OpenFOAM®, along with the proposed fitting coefficients.

The Reynolds Shear Stress (RSS) modeling technique consists in solving a set of seven transport equations, derived from the six independent components of the RSS tensor and the turbulent kinetic energy k. This set of equations is known as the RSS transport equations. In these equations, the pressure–strain correlation, the pressure and turbulence diffusion, and the specific turbulent kinetic energy dissipation terms require specific modeling. In the present work, the RSS model of [20], already implemented in OpenFOAM®, was tested in the BFS.

In the LES technique, for the accurately modeling of the wide range of anisotropic turbulent structures of the BFS, the multi-gradient model for the Sub-Grid Scale

(SGS) stress tensor developed by [21] was implemented in OpenFOAM®. The multi-gradient equations for the SGS stress tensor are

$$\boldsymbol{\tau}^{\mathbf{r}} = \frac{8\bar{\Delta}^2}{C_\varepsilon^2}\left(-\frac{\mathbf{G}}{tr(\mathbf{G})}\mathbf{S}\right)^2\left(\frac{\mathbf{G}}{tr(\mathbf{G})}\right)H(P),$$
$$\mathbf{G} = \frac{1}{12}[(\mathbf{D}\cdot(\nabla\mathbf{u}))(\mathbf{D}\cdot(\nabla\mathbf{u})^T)], \tag{9}$$

where $\mathbf{D} = (\Delta_x, \Delta_y, \Delta_z)$ is the grid filter vector, $\bar{\Delta} = (\Delta_x\Delta_y\Delta_z)^{1/3}$ is the grid filter length, and C_ε is a coefficient that depends on whether the local equilibrium or global equilibrium hypothesis is used [22]. This last coefficient was taken as equal to 1 in the present case. Furthermore, $H(P)$ is a Heaviside function that turns off the model whenever the turbulent production term $P = -\boldsymbol{\tau}^{\mathbf{r}}\cdot\mathbf{S}$ becomes negative. The filter length was taken as twice the mesh size ($\bar{\Delta} = 2\Delta$) in the simulations.

The results for the steady-state stream-wise component of the velocity, for the y-lines displayed in Fig. 2, are shown in Fig. 3. Simulations were performed for each of the above turbulence models (k–ε, k–ω, RSS, cubic nonlinear and multi-gradient LES). As previously discussed, all simulations were performed in a converged mesh.

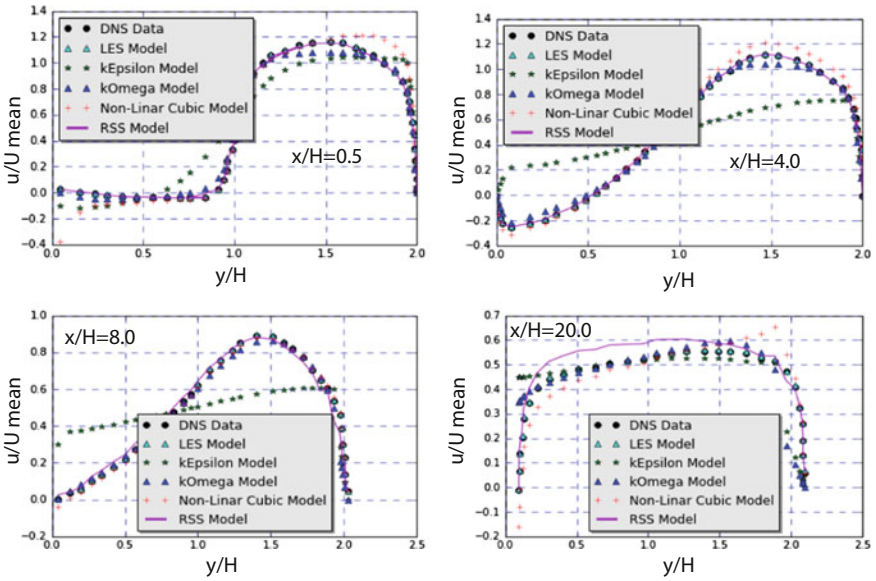


Fig. 3 Top-left: Results for the normalized steady-state stream-wise component of the velocity for the lines $\frac{x}{H} = 0.5$. Top-right: $\frac{x}{H} = 4.0$. Bottom-left: $\frac{x}{H} = 8.0$. Bottom-right: $\frac{x}{H} = 20.0$. In the above results, the LES model corresponds to a multi-gradient SGS model [21] implemented in OpenFOAM® with no near wall treatment, the k–ε [14], and k–ω [17] models are the standard models used in OpenFOAM®, the nonlinear cubic model corresponds to the one developed by Craft [19] and implemented in OpenFOAM® and the RSS model is the SSG model available in OpenFOAM® developed by Speziale [20]

The results of these models are compared against the DNS data, which is taken as the reference. The general performance of the k–ε model for the BFS is judged to be poor. This is a result of the overestimation of the turbulent dissipation ε, as well as the turbulent kinetic energy k. Thereupon, the k–ε model predicts an early development of the velocity profile past the BFS expansion. The results predicted by the k–ω model are, however, in much better agreement with the DNS data. This is because, when calculating $\omega = \frac{\varepsilon}{k}$, both overpredictions compensate remarkably well. The nonlinear viscosity model has a good agreement with the DNS data close to the middle stream, but it performs badly next to the walls. Since the closure coefficients of this model were fitted by Craft [19] for the development of a boundary layer over a flat plate, it is not surprising that they are misled for the highly anisotropic internal flow of the BFS (in which the flow velocity perpendicular to the walls may be important). The RSS transport model shows good results in the region immediately after the detachment of the boundary layer. However, it underestimates the dissipation of the turbulent kinetic energy in the stream-wise direction, causing an over diffusion of the shear layer and significant errors in the predicted velocities downstream. Finally, the multi-gradient LES model shows an almost perfect agreement with the DNS data.

To quantitatively analyze the error of the turbulence models, the mis-prediction in the axial velocity on the y-lines displayed in Fig. 2 is weighted by the *importance* of this mis-prediction. Therefore, the average weighted quadratic error is computed for each y-line l ($\frac{x}{H} = 0.5, 4, 8, 20$) with n_l data points as

$$L_w = \frac{1}{4}\sum_{l=1}^{4}\frac{1}{n_l}\sum_{i=1}^{n_l} w(y_{il})\left(\frac{u_{DNS_{il}} - u_{MODEL_{il}}}{u_{DNS_{il}}}\right)^2,$$
$$w(y) = 1 - \left(1 - \frac{y}{h}\right)^2, \tag{10}$$

where $u_{DNS_{il}}$ is the velocity stream-wise component of the DNS data for the line l at point y_i and $u_{MODEL_{il}}$ is the predicted stream-wise velocity for the turbulence model under consideration. The function $w(y)$ is a *Poiseuille*-like weighting function that maximizes the importance of the errors in the middle stream of the section. This function is introduced regarding the expansion of the MSFR reactor, in which the middle stream region in the flow has a larger *importance* than the region close to the walls, due to neutronic considerations. The errors obtained and the CPU time to convergence to steady state in the simulations are shown in Table 1. The only model that allows to obtain an error smaller than 5% is the multi-gradient LES model, but its computational cost is prohibitively expensive when considering its application to the scale of the MSFR. In addition, the nonlinear stress model is close to the targeted error. A good attempt might have been made to improve the closure coefficients for this model, but due to its complexity, further improvements are difficult to develop. Therefore, a new nonlinear eddy viscosity model based on the k–ε model, developed by means of symbolic regression using the GEATFOAM tool, is discussed in the following section.

Table 1 Relative errors and computational cost for the different turbulence models

Model	Relative error (%)	CPU time@16 × 1.2 GHz(s)
k–ε	14.32	674
k–ω	8.14	725
Nonlinear cubic	5.64	1243
RSS	7.61	1785
LES	0.73	74,513

3 Optimization of a k–ε Model with GEATFOAM

The Genetic Evolutionary Algorithms for Turbulence modeling tool (GEATFOAM) is a library developed in C++ that can be compiled with the OpenFOAM® libraries. The library implements symbolic regression and gradients optimization techniques for constructing a mathematical expression for the anisotropy tensor.

Symbolic regression techniques are a subset of regression techniques, which allow you to search the space of mathematical expressions for a model that best fits the data [23, 24] , which, in the present case, is the stream-wise velocity over the *y-lines* displayed in Fig. 2. In symbolic regression analysis, new expressions are generated by randomly combining mathematical building blocks (such as constants or tensor, or some of their more complicated functions). The expressions are then tested against the data and catalogued according to an error, determined by some previously defined measure [25]. Subsequently, new expressions are constructed by recombining (more or less randomly in genetic algorithms) the previously obtained ones. The core concept beneath the recombination process is to prioritize the recombination of the better-fitting expressions, decreasing the mean error of the newly obtained expressions, and obtaining better-fitting ones for each subsequent iteration. Several techniques can be applied for performing the recombination process, namely, genetic programming [26], neural networks [25], and support vector machines [25]. However, genetic programming techniques have been demonstrated to outperform the others on problems in which simple and interpretable expressions should be obtained [27]. For the recombination process, GEATFOAM implements a symbolic tensorial version of the NSGA-II evolutionary algorithm proposed in [28].

The flowchart of the library is presented in Appendix. The initial step consists in the generation of the mathematical blocks (constants and tensors) on which the expressions for the anisotropy tensor are built on. In mathematical terms, these building blocks belong to two different sets: a variable set and an operation set.

$$\begin{aligned} \varphi &= \{\mathbf{T_i}, C_i, I_2(\mathbf{S}), I_3(\mathbf{S}), I_4(\mathbf{S}), I_2(\mathbf{W}), I_3(\mathbf{W}), I_4(\mathbf{W})\}, \\ \vartheta &= \{*_{\Re^2,\Re^0}, *_{\Re^0,\Re^0}, /_{\Re^0,\Re^0}, +_{\Re^2,\Re^2}, +_{\Re^0,\Re^0}, -_{\Re^2,\Re^2}, -_{\Re^0,\Re^0}\}. \end{aligned} \tag{11}$$

The functional set φ contains the variables for describing the nonlinear anisotropy tensor. In the present case, due to the rank of the anisotropy tensor and the geometric

considerations explained in Sect. 2, the variables are the $\mathbf{T}_i$ tensors in Eqs. 6–8, the fitting constants and the second, third, and fourth invariants [29] of the shear rate and the vorticity tensors. The operator set ϑ contains the set of operations allowed over the set. For the present case, they are scalar-matrix multiplication ($*_{\Re^2,\Re^0}$), scalar multiplication ($*_{\Re^0,\Re^0}$) and division ($/_{\Re^0,\Re^0}$), matrix addition ($+_{\Re^2,\Re^2}$) and subtraction ($-_{\Re^2,\Re^2}$), and scalar addition ($+_{\Re^0,\Re^0}$) and subtraction ($-_{\Re^0,\Re^0}$).

Then, an initial number of ten expressions for the anisotropy tensor (referred to as *individuals* in genetic programming) are randomly generated, combining the variables from the allowed operations (e.g., $\mathbf{a} = C_1\mathbf{T_1} + C_3 I_2(\mathbf{S})\mathbf{T_2} + C_3\mathbf{T_3}$). In GEATFOAM, each individual is represented by an open reading frame (ORF) [30]. Each expression may contain a set of fitting constants C_i, which are not necessarily optimally fitted to the data. Therefore, the next step is to optimize the fitting constants in each expression so as to minimize the value of the fitness function defined by Eq. 10. In proper mathematical terms, this is an unconstrained optimization problem for a fitness function $L_w : R^n \rightarrow R$, where n is the number of C_i coefficients in the optimization model, a number that may vary between individuals. Defining $\mathbf{C}$ as the vector of coefficients to be optimized for the fitness problem, $\mathbf{g} = g_i = \frac{\partial L_w}{\partial C_i} \approx \frac{\Delta L_w}{\Delta C_i}$ as the finite difference approximated gradient of the fitness function at certain values of C_i, and $\mathbf{H} = H_{ij} = \frac{\partial^2 L_w}{\partial C_i \partial C_j} \approx \frac{\Delta L_w/\Delta C_i}{\Delta C_j}$ as the finite difference approximated Hessian, the iterative quasi-Newton optimization problem is implemented as follows:

$$\begin{aligned} \mathbf{H}^{(n)}\mathbf{p}^{(n)} &= -\mathbf{g}^{(n)}, \\ \mathbf{C}^{(n+1)} &= \mathbf{C}^{(n)} + \mathbf{p}^{(n)}. \end{aligned} \tag{12}$$

At this point, a limitation is discovered, the optimization problem is not necessarily convex or, equivalently, the Hessian matrix is not necessarily positive-defined. This implies that the previous system may not converge to a minimum. To solve this issue, the method proposed by Forsgren et al. [31] is applied, in which the Hessian is factorized by a Cholesky algorithm with row pivoting. The Hessian factorization becomes $\mathbf{H} = \mathbf{L}\mathbf{D}\mathbf{L}^T$, replacing the negative values in the matrix diagonal with small positive values during the factorization process. Therefore, a descendent direction toward a minimum can be found independently of the convexity of the problem. This technique is usually referred to as the *active set method*.

Once the coefficients for each model have been optimized, the obtained expressions are arranged in a stack list that confers its fitting error to the data. Additionally, an extra error proportional to the size of the individual ORF is added for to prioritize simple expressions for the anisotropy models. Then, the recombination process takes place, in which a set of random genetic operations is applied to the ORFs prioritizing the reproduction of the better-fitting individuals. The genetic operations supported by GEATFOAM consist in changing terms in the tail of the ORF by introducing terms of other ORF (recombination), by introducing terms of the head of the same ORF (transposition) or changing it through random fragments of operators and/or variables (mutation). Once the genetic operations have taken place, a new set of averagely more fit expressions is introduced back into the loop. When an expression

arrives to satisfy the error criteria imposed on the fitness function (less than 5% of error in the stream-wise velocity over the *y-lines*), the iteration process finishes and the individual is extracted.

4 A Nonlinear Quadratic Closure for the Anisotropy Tensor Developed with the GEATFOAM Tool

The model obtained for the anisotropy tensor, resulting from the iteration process of the GEATFOAM process with a randomly generated initial population, is

$$\mathbf{a} = 0.074\mathbf{T_1} + 0.521\mathbf{T_2} + 0.071\frac{I_2(\mathbf{S})I_3(\mathbf{S})}{I_2(\mathbf{W})I_3(\mathbf{W})}\mathbf{T_3} - 0.180\mathbf{T_4}. \tag{13}$$

The output of GEATFOAM for the anisotropy tensor is a nonlinear quadratic model, since large extra weights have been assigned to big expressions during the optimization process. In order to assess the repeatability of the evolutionary process, different sets of random initial populations have been tested, obtaining a success ratio [32] for the model proposed in Eq. 13 of 85%, meaning that the model was obtained in 85% of the evolutionary lines studied. None of the other models resulting from offsprings in the evolutionary process had a smaller error than Eq. 13, according to the definition of error by Eq. 10.

The results obtained for the optimized nonlinear quadratic model (Eq. 13) are compared against the DNS data, the standard k–ε [14] and the nonlinear cubic model (Eqs. 5–8 [19]) in Fig. 4, for the stream-wise velocity over the *y-lines* shown in Fig. 2. Both nonlinear models outperform the standard k–ε model. The model that best fits the DNS data is the optimized nonlinear quadratic model (13), which presents a uniform accuracy over all *y-lines*. The nonlinear cubic model (5–8) performs well on the first three lines, where the turbulence production is important, but have important errors downstream, where the turbulent dissipation should compensate the production.

The values for the fitness function L_w of these models are compared in Table 2; it is observed that the optimized nonlinear quadratic model (13) satisfies the originally imposed error criteria. In terms of the turbulent kinetic energy and specific turbulent dissipation, the nonlinear modeling of the viscosity allows us to account for the turbulent structures produced by the shear stress history in the flow. This, in turn, allows to avoid the overprediction of k and ε after the step expansion, accurately modeling the production and dissipation in the main recirculation bubble (graphical results are presented in the Appendix).

In order to evaluate the extrapolability of the constructed anisotropic model, the results are applied to an axisymmetrical pipe expansion with a different Reynolds number. In principle, since the underlying physics of the axisymmetric expansion should be similar to that of the BFS, the developed nonlinear quadratic model (13) should perform well on this new geometry. Measured data for the axial velocities is

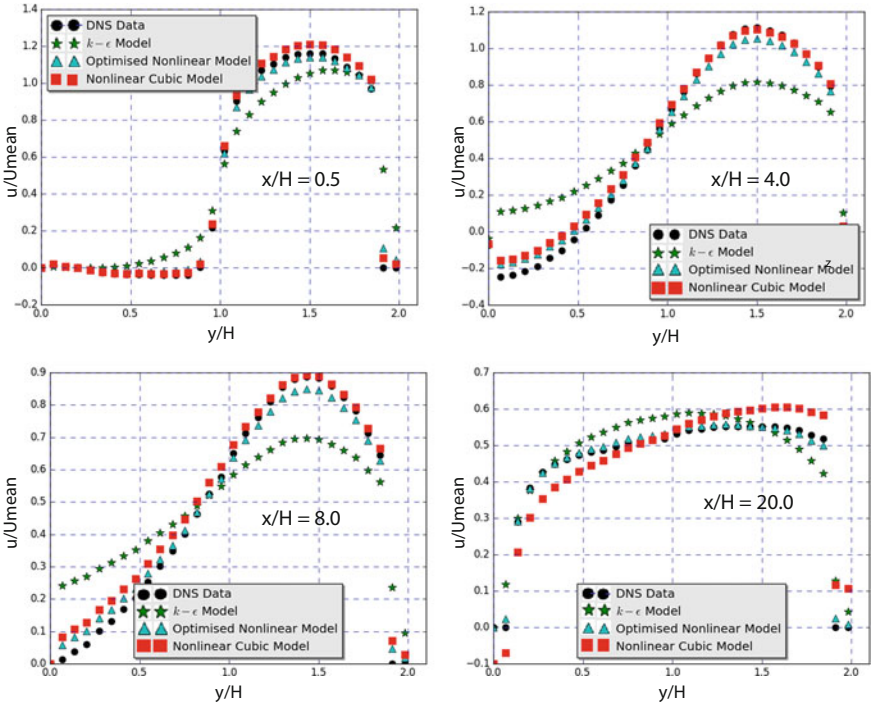


Fig. 4 Top-left: Results for the normalized stream-wise steady-state velocity for the lines $\frac{x}{H} = 0.5$. Top-right: $\frac{x}{H} = 4.0$. Bottom-left: $\frac{x}{H} = 8.0$. Bottom-right: $\frac{x}{H} = 20.0$. The DNS data points are taken as reference values. The results were obtained using the standard k–ε model [14], the optimized nonlinear model (13), and Craft's cubic nonlinear model [19]

Table 2 Assessment of the optimized nonlinear model against the standard k–ε model and Craft's nonlinear cubic model

Model	Relative error (%)	CPU time@16×1.2 GHz(s)
Standard k–ε	14.32	674
Optimized Nonlinear Model	2.97	1124
Nonlinear Cubic Model	5.64	1243

provided in the work done by [33] at $x = 0.05D$ and $x = 0.25D$ after the expansion, and is taken as reference in the present case. The section has an expansion rate of 1.94 and the mean inlet velocity is fixed so that $Re = \frac{UD}{\nu} = 2 \times 10^5$, taking the value of the kinematic viscosity as $\nu = 10^{-5}\,\frac{\mathrm{m}^2}{\mathrm{s}}$. The standard k–ε model [14] and the optimized nonlinear quadratic model (13) were tested on this geometry. Steady-state and 2D axisymmetrical simulations were performed, converging the mesh in each case. The results are shown in Fig. 5. A good agreement is observed between the nonlinear quadratic model (13) and the experimental results [33].

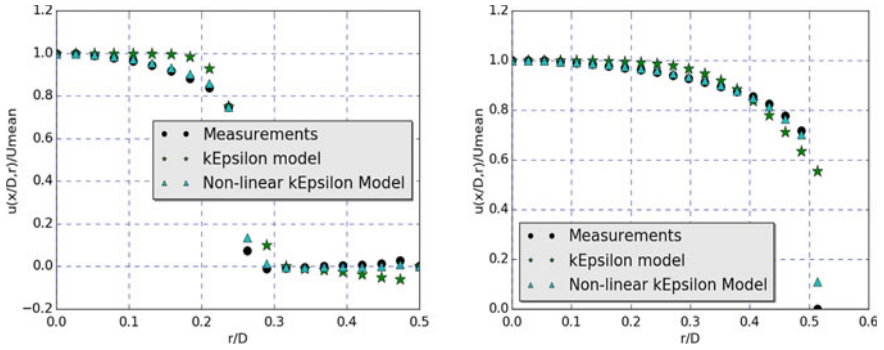


Fig. 5 Results for the normalized axial steady-state velocity for the lines Left: $\frac{x}{D} = 0.05$ and Right: $\frac{x}{D} = 0.025$. The results were obtained using the standard k–ε [14] model and the evolved nonlinear quadratic model (13) model by means of nonlinear modeling of the anisotropy. The measured velocities are taken as reference values. For the k–ε turbulent model, $L_w = 11.51\%$, and for the modified model, $L_w = 4.05\%$

5 Conclusions

In the present work, different turbulence models have been evaluated for a BFS geometry, with the objective of validating a model to be applied in the design of a molten salt fast reactor. On the RANS models side, the Wilcox k–ω model [17], the implemented cubic nonlinear viscosity model [19] and the SSG Reynolds Shear Stress transport model [20] mispredict the stream-wise components of the bulk velocities by 5–10%. In addition, the standard k–ε model [14] presents an error of 15% in the prediction of these velocities. As a consequence, they all fail to attain the required error of less than 5% of mis-prediction in the stream-wise velocities. An LES multi-gradient [21] model was also tested in the BFS, having less than 1% of error in the prediction of the stream-wise velocities. However, it consumes excessive computational resources and its convergence is complicated for complex geometries, making it inadequate to use for design purposes and for its application to the MSFR. Therefore, the development of an adapted turbulent model was necessary for the BFS section. A new nonlinear quadratic model for the anisotropy tensor, based on the k–ε equations, was developed by means of symbolic regression through genetic evolutionary programming (the GEATFOAM tool).

To the current extent, the tool still presents the limitation that the results found are sensible to the initial population proposed in the evolutionary process, and different results for the anisotropy tensor are obtained between successive runs of the tool. However, by increasing the cost of high-order nonlinear models, an efficient quadratic model for the anisotropy tensor was found, having an error of less than 5% for the analyzed BFS. Furthermore, the extrapolability of this new model was successfully evaluated for a pipe expansion case.

Appendix

See Figs. 6 and 7

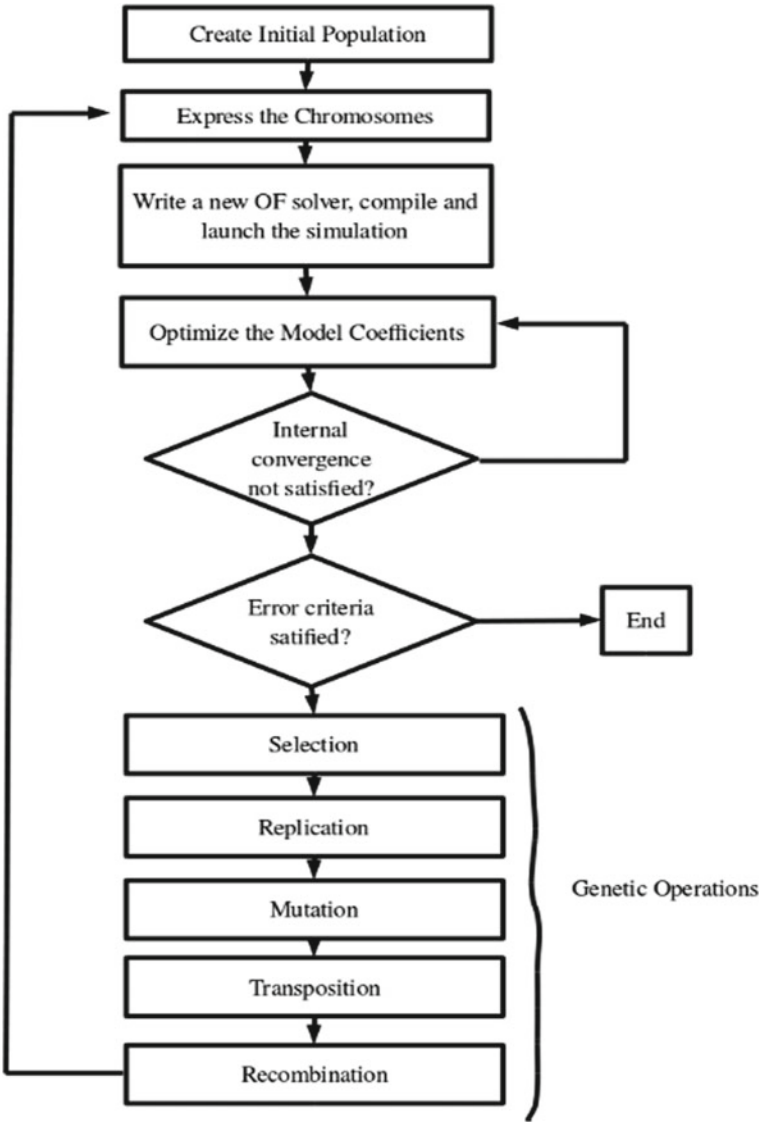


Fig. 6 Flowchart of the GEATFOAM tool. First, an initial population is proposed and its chromosomes are *expressed* in ORFs. A first internal loop optimizes the free scalar coefficients of the ORF through a conjugate gradient method. Then, a set of genetic operations is introduced, changing the population for one with a better fit and the iterations are repeated. The process ends when one individual attains the error criteria

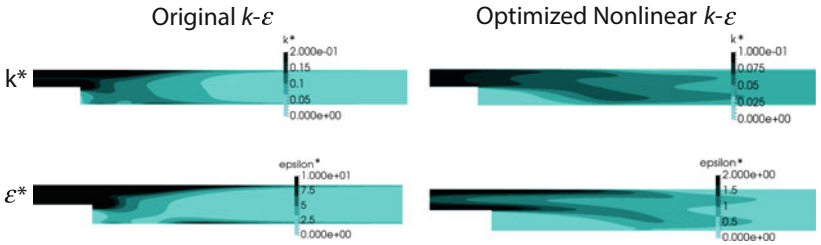


Fig. 7 Results obtained for the normalized turbulent kinetic energy $k^* = \frac{k}{U_{mean}^2}$ and the normalized specific turbulent dissipation $\varepsilon^* = \frac{\varepsilon v}{U_{mean}^4}$. The standard k–ε model overpredicts k and ε after the expansion, causing an overprediction in the turbulent viscosity nu_T and a rapid development of the velocity profile after the step. The optimized nonlinear k–ε model correctly models the turbulent viscosity as a function of the history of the stresses in the fluid, allowing us to avoid the initial burst of the standard k–ε turbulent model and being able to resolve the evolution of k and ε in the main recirculation bubble of the BFS

References

1. Lund, Henrik, et al. "4th Generation District Heating (4GDH): Integrating smart thermal grids into future sustainable energy systems." Energy 68 (2014): 1–11.
2. Heuer, D., Merle-Lucotte, E., Allibert, M., Brovchenko, M., Ghetta, V., and Rubiolo, P. (2014). Towards the thorium fuel cycle with molten salt fast reactors. Annals of Nuclear Energy, 64, 421–429.
3. Raffel, Markus, Christian E. Willert, Steven Wereley, and Jrgen Kompenhans. Particle image velocimetry: a practical guide. Springer, 2013.
4. Moser, Robert D., John Kim, and Nagi N. Mansour. "Direct numerical simulation of turbulent channel flow up to Re= 590." Phys. Fluids 11, no. 4 (1999): 943–945.
5. Pope, Stephen B. "Turbulent flows." (2001): 2020.
6. Germano, M. "Turbulence: the filtering approach." Journal of Fluid Mechanics 238 (1992): 325–336.
7. Rubiolo, P. R., et al. "Overview of the Salt at WAll Thermal ExcHanges (SWATH) Experiment." 2016 International Topical Meeting on High Temperature Reactor Technology (HTR2016). 2016.
8. Thangam, S., Speziale, C. G. (1991). Turbulent separated flow past a backward-facing step: a critical evaluation of two-equation turbulence models INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING HAMPTON VA.(No. ICASE-91-23).
9. Kopera M., Kerr R., Blackburn H. and Barkley D. (2014). Direct numerical simulation of turbulent flow over a backward-facing step. Approved to appear in the Journal of Fluid Mechanics.
10. Le, H., Moin, P., Kim, J. (1997). Direct numerical simulation of turbulent flow over a backward-facing step. Journal of fluid mechanics, 330(1), 349–374.
11. LIM, H., et al. Chaos, transport and mesh convergence for fluid mixing. Acta Mathematicae Applicatae Sinica (English Series), 2008, vol. 24, no 3, p. 355–368.
12. https://openfoamwiki.net/index.php/SnappyHexMesh .
13. Kwon, Y. S., N. Hutchins, and J. P. Monty. "On the use of the Reynolds decomposition in the intermittent region of turbulent boundary layers." Journal of Fluid Mechanics 794 (2016): 5–16.
14. Launder, B. E., and B. I. Sharma. "Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc." Letters in heat and mass transfer 1, no. 2 (1974): 131–137.

15. Van Driest, Edward R. “On turbulent flow near a wall.” Journal of the Aeronautical Sciences (2012).
16. Chen, Y-S., and S-W. Kim. “Computation of turbulent flows using an extended k-epsilon turbulence closure model.” (1987).
17. Wilcox, David C. “Formulation of the kw turbulence model revisited.” AIAA journal 46, no. 11 (2008): 2823–2838.
18. Schfer, F., M. Breuer, and F. Durst. “The dynamics of the transitional flow over a backward-facing step.” Journal of Fluid Mechanics 623 (2009): 85–119.
19. Craft, T. J., B. E. Launder, and K. Suga. “Development and application of a cubic eddy-viscosity model of turbulence.” International Journal of Heat and Fluid Flow 17, no. 2 (1996): 108–115.
20. Speziale, Charles G., Sutanu Sarkar, and Thomas B. Gatski. “Modelling the pressurestrain correlation of turbulence: an invariant dynamical systems approach.” Journal of Fluid Mechanics 227 (1991): 245–272.
21. Lu, Hao, and Fernando Port-Agel. “A modulated gradient model for scalar transport in large-eddy simulation of the atmospheric boundary layer.” Physics of Fluids (1994-present) 25, no. 1 (2013): 015110.
22. Ghaisas, Niranjan S., and Steven H. Frankel. “Dynamic gradient models for the sub-grid scale stress tensor and scalar flux vector in large eddy simulation.” Journal of Turbulence 17, no. 1 (2016): 30–50.
23. Billard, Lynne, and Edwin Diday. “Symbolic regression analysis.” Classification, Clustering, and Data Analysis. Springer Berlin Heidelberg, 2002. 281–288.
24. Keijzer, Maarten. “Scaled symbolic regression.” Genetic Programming and Evolvable Machines 5.3 (2004): 259–269.
25. Smits, Guido, and Mark Kotanchek. “Pareto-front exploitation in symbolic regression.” Genetic programming theory and practice II (2005): 283–299.
26. Vladislavleva, Ekaterina J., Guido F. Smits, and Dick Den Hertog. “Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming.” IEEE Transactions on Evolutionary Computation 13.2 (2009): 333–349.
27. Yu, Tina, Rick Riolo, and Bill Worzel, eds. Genetic programming theory and practice III. Vol. 9. Springer Science and Business Media, 2006.
28. Deb, Kalyanmoy, Amrit Pratap, Sameer Agarwal, and T. A. M. T. Meyarivan. “A fast and elitist multiobjective genetic algorithm: NSGA-II.” IEEE transactions on evolutionary computation 6, no. 2 (2002): 182–197.
29. Cantwell, Brian J. “On the behavior of velocity gradient tensor invariants in direct numerical simulations of turbulence. Physics of Fluids A: Fluid Dynamics 5.8 (1993): 2008–2013.
30. Takahashi, Hiro, Anna Takahashi, Satoshi Naito, and Hitoshi Onouchi. “BAIUCAS: a novel BLAST-based algorithm for the identification of upstream open reading frames with conserved amino acid sequences and its application to the Arabidopsis thaliana genome.” Bioinformatics 28, no. 17 (2012): 2231–2241.
31. Forsgren, Anders, Philip E. Gill, and Elizabeth Wong. “Active-set methods for convex quadratic programming.” arXiv preprint ArXiv:1503.08349 (2015).
32. Gen, Mitsuo, and Runwei Cheng. Genetic algorithms and engineering optimization. Vol. 7. John Wiley and Sons, 2000.
33. Khezzar, L., J. H. Whitelaw, and M. Yianneskis. “An experimental study of round sudden-expansion flows.” In 5th Symposium on Turbulent Shear Flows, vol. 1, p. 5. 1985.

Differential Heating as a Strategy for Controlling the Flow Distribution in Profile Extrusion Dies

Ananth Rajkumar, Luís L. Ferrás, Célio Fernandes, Olga S. Carneiro, Alberto Sacramento and J. Miguel Nóbrega

Abstract This work presents a simple procedure for balancing the flow in extrusion dies. The method consists in using different temperatures on the different sides of the extrusion die surface, in this way altering the local viscosity of the polymer melt, and thus the melt flow distribution. The design methodology follows a numerical trial-and-error procedure (implemented in OpenFOAM®), which was assessed with an industrial case study (swimming pool cover profile). The results obtained show that the support of computational tools is an excellent design aid, and a much better alternative to the experimental trial-and-error procedure commonly used in industry.

1 Introduction

Extruded thermoplastic profiles find application in different fields, such as automotive, household, civil construction, electrical, and health, among many others. One of the main concerns in profile extrusion is the flow balance at the die flow channel outlet. Most of the time, the geometries of the extruded profiles are complex and induce differential flow restrictions, thus promoting an unbalanced flow distribution.

A. Rajkumar · L. L. Ferrás · C. Fernandes · O. S. Carneiro · J. M. Nóbrega (✉)
Institute for Polymers and Composites/i3N, University of Minho,
Campus de Azurém, 4800-058 Guimarães, Portugal
e-mail: mnobrega@dep.uminho.pt

A. Rajkumar
e-mail: rajkananth@gmail.com

L. L. Ferrás
e-mail: luis.ferras@dep.uminho.pt

C. Fernandes
e-mail: cbpf@dep.uminho.pt

O. S. Carneiro
e-mail: olgasc@dep.uminho.pt

A. Sacramento
Soprefa - Componentes Industriais SA, Mosteiró, Portugal
e-mail: sacramento@soprefa.com

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_8

In this case, the melt flow average velocity varies along the die outlet cross section. This happens because fluids flow easily in less restrictive (thicker) regions [1, 2]. Several methodologies have been proposed in the literature to solve this problem, the experimental trial-and-error procedure being the most popular method among the profile extrusion companies. The major drawbacks of this method are the fact that it is time-consuming and expensive, and relies on the designer's know-how. More recently, with the huge increase in computational power, the numerical simulation of these processes became a reality, and some approaches were proposed regarding a numerical optimization of the extrusion process [1]. These studies either rely on numerical trial-and-error procedures (inexpensive but still dependent on the know-how of the user) or more autonomous numerical techniques that perform the optimization automatically [1–6].

Independently of the current situation, there is a huge gap between industry (which, in general, still relies on resource-consuming experimental-based approaches) and numerical tools, which can provide a very useful support for them. In this work, we bridge the gap between numerical simulation and industry, by presenting, in detail, the optimization of an existing extrusion die. The optimization was performed in cooperation with a profile extrusion company, and we show that numerical simulations can be used to reduce costs and the time to market.

All the computational tools used in this work are open-source, meaning that no special licensing is needed. The numerical modeling code is developed using the OpenFOAM® computational library [7], a free open-source and expandable set of numerical tools, which includes routines for creating the geometry and generating the computational mesh, and also allows for parallel computing. A new solver was developed to model the steady non-isothermal flow of incompressible generalized Newtonian fluids. The optimization of the extrusion die was performed by imposing different temperatures on the different surfaces of the extrusion die (which adjusts locally the viscosity of the polymer melt).

The remainder of this work is organized as follows. In Sect. 2, the developed design methodology is presented and discussed, explaining in detail the main stages involved. In Sect. 3, we present the governing equations of the new solver developed in the OpenFOAM® framework. In Sect. 4, we illustrate the usefulness of the developed numerical tools in an industrial case study, which comprises the improvement of the flow distribution in an existing die. The work ends with the main conclusions.

2 Die-Design Methodology

The design procedure encompasses a series of stages involving the use of different software packages, as illustrated in Fig. 1 and detailed in [8]. The procedure starts with the material rheological characterization, to determine the relevant rheological parameters for modeling purposes. Then, the die outlet cross section is divided into Elemental Sections (ESs) and Intersection Sections (ISs). These subsections will be used to monitor the flow distribution (or flow rate) at the die outlet. The generation

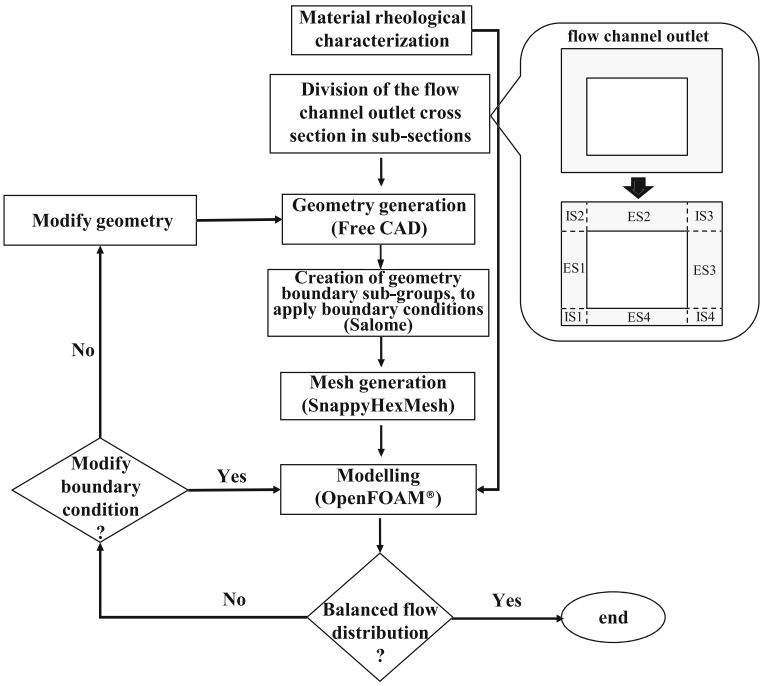


Fig. 1 Proposed die-design methodology

of the flow channel geometry is done using a CAD software (Free CAD [9]). The generated geometry is imported to Salome [10], and the geometry boundary is divided into different groups to allow for setting different boundary conditions. The defined regions are exported as STL ASCII files from Salome and used as input for the mesh generation code. The computational mesh is generated using snappyHexMesh [11], a mesh generation utility included in OpenFOAM®. Finally, the numerical simulation is performed in the OpenFOAM® framework using the developed solver. A standard post-processing utility from OpenFOAM®, PatchIntegrate, is used to compute the flow rate in each subsection of the flow channel outlet (see Fig. 1).

The numerically obtained flow rate is then used to calculate the flow distribution's objective function ($F_{obj,i}$) at each ES and IS, using the following equation:

$$F_{obj,i} = \frac{\frac{U_i}{U_{av}} - 1}{\max\left(\frac{U_i}{U_{av}}; 1\right)}, \tag{1}$$

where U_i and U_{av} are the ESi (or ISi) and the global average velocities, respectively. These velocities are obtained by dividing the computed flow rate by the respective subsection area. The objective function ($F_{obj,i}$) was defined in such a way so that the ratio U_i/U_{av} has the equivalent contribution if it is half or double of the value

required for flow balancing. In the case of perfect flow balancing, the $F_{obj,i}$ value will be zero. The performance of each iteration is quantified using

$$F_{obj} = \sum_{i=1}^{N_{ES}} \left\| F_{obj,i} \right\| + \sum_{i=1}^{N_{IS}} \left\| F_{obj,i} \right\|, \tag{2}$$

where N_{ES} and N_{IS} represent the total number of ESs and ISs, respectively. Notice that, in this way, the die performance increases with the reduction of the F_{obj} absolute value.

The iterative process is repeated by adjusting the geometry and/or boundary conditions until a balanced flow distribution is achieved (see [8] for more details). In this work, only the boundary conditions will be adjusted to improve the flow distribution.

3 Numerical Modeling

The numerical code should be able to model the flow of an incompressible and inelastic fluid under non-isothermal conditions. This requirement led to the implementation of a new solver, in the OpenFOAM® computational library, able to solve the relevant governing equations. This new solver is briefly described later (see [8] for more details).

3.1 Governing Equations

The equations used to model the flow are, the mass conservation,

$$\nabla \cdot \mathbf{u} = 0 \tag{3}$$

and the linear momentum conservation

$$\nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau}. \tag{4}$$

The energy conservation equation must also be considered to predict the temperature distribution:

$$\nabla \cdot \left(\rho c_p \mathbf{u} T\right) - \nabla \cdot (k \nabla T) = \boldsymbol{\tau} : \nabla \mathbf{u}. \tag{5}$$

In the above equations, ρ is the density, $\mathbf{u}$ is the velocity vector, p is the pressure, $\boldsymbol{\tau}$ is the deviatoric stress tensor, T is the temperature, c_p is the specific heat, and k is the thermal conductivity. The last term on the RHS of Eq. 5, $\boldsymbol{\tau} : \nabla \mathbf{u}$, accounts for the viscous dissipation contribution.

For generalized Newtonian fluids, the deviatoric stress tensor is

$$\boldsymbol{\tau} = 2\eta(\dot{\gamma}, T)\mathbf{D}, \tag{6}$$

where $\mathbf{D}$ is the rate of strain tensor, given by

$$\mathbf{D} = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right), \tag{7}$$

where $\nabla\mathbf{u}$ is the velocity gradient tensor, and $\eta(\dot{\gamma}, T)$ is the shear viscosity that depends both on temperature (T) and shear rate ($\dot{\gamma}$), being a function of the second invariant of the rate of the deformation tensor,

$$\dot{\gamma} = 2\sqrt{tr\mathbf{D}^2}. \tag{8}$$

To take into account the effect of both temperature and shear rate on the shear viscosity, the Bird-Carreau model was coupled to an Arrhenius law, leading to

$$\eta(\dot{\gamma}, T) = a_T\eta_\infty + \frac{a_T\left(\eta_0 - \eta_\infty\right)}{\left(1 + (a_T\lambda\dot{\gamma})^2\right)^{(1-n)/2}}, \tag{9}$$

where η_0 and η_∞ are the zero and infinite shear rate viscosities, respectively, λ is a constant with units of time (defining the value of the shear rate at which the shear thinning effect begins), and n is the power-law exponent. In Eq. 9, the temperature shift factor, a_T, is given by

$$a_T = \exp\left(\frac{E}{R}\left(\frac{1}{T} - \frac{1}{T_0}\right)\right), \tag{10}$$

with R the universal gas constant, E the activation energy, and T and T_0 the temperature and its reference value (both in Kelvin), respectively.

The implementation of the solver is explained in detail in the Ph.D. thesis by the first author [12].

4 Case Study

The new numerical code will now be used to improve the flow balance of an existing extrusion die used for the production of a swimming pool cover profile. In this case, the extrusion die, already tested in the company, showed an unsatisfactory flow distribution. This die had four heaters, one on each face of the tool, all controlled by the same thermocouple. Since the polymer melt shear viscosity is substantially affected by temperature, the local resistance to flow can be adjusted through controlled variations of the material temperature at specific locations. Thus, the main purpose of this

study was to verify if it is worth having the four heaters controlled by independent thermocouples, in order to improve the flow distribution. If approved, this would comprise an economic alternative to the redesign and manufacture of a new properly balanced die. The stages of the design procedure (illustrated in Fig. 1) are followed, by modifications of the boundary conditions, i.e., the boundary temperatures, in this case.

4.1 Material Characterization

The material used by the extrusion company to produce the profiles under study is a polycarbonate extrusion grade (TRIREX 3027U(M1)). The shear viscosity flow curves were obtained using both parallel plate (ARG2, from TA Instruments) and capillary (Rosand RH7, from Malvern Instruments) rheometers, at three different temperatures (230, 250 and 270 °C). The intermediate temperature (250 °C) was chosen as the reference one, and the experimental viscosity data obtained for 230 and 270 °C were shifted to that temperature using the corresponding shift factor a_T (Eq. 10). A fit to the master flow curve was made with a Bird-Carreau model, resulting in the following parameters values: $\eta_0 = 11{,}731$ Pa s, $\eta_\infty = 0$ Pa s, $n = 0.524$, $\lambda = 0.1572$ s and $E/R = 16{,}296$ K (see [8] for more details).

4.2 Geometry and Mesh

The exact flow channel geometry of the existing die was replicated in the CAD software, and the flow channel boundary was divided into pre-parallel zone (PPZ), top, bottom, right, left, and inner walls, as indicated in Fig. 2a. The division in four outer regions allows for imposing different temperatures in each zone.

The flow channel outlet cross section and its division into ESs and ISs are shown in Fig. 2b. The $F_{obj,i}$ will only be shown for the ESs, due to the large number of subsections involved, and also because the direct control of flow in the ISs is not possible. For modeling purposes, the boundary conditions employed in this case are as follows. The inlet velocity was determined from the required extrusion line speed, 1.28 m/min, considering a zero gradient boundary condition at the outlet, and no slip ($\mathbf{u} = \mathbf{0}$) at the walls. Concerning the pressure field, the outlet pressure was assumed to be zero and a zero gradient was imposed at the inlet and walls. Regarding the temperature field, different sets of values will be used in the die walls (top, bottom, right, and left). The inlet and the converging walls were assumed to have a value of 245 °C (as prescribed by the company), and we have imposed a zero gradient at the outlet and inner walls.

A mesh sensitivity analysis was performed to identify the mesh refinement level required to obtain accurate results. Three different mesh refinements with 432,653, 3,336,519 and 26,254,523 computational cells were used to discretize the flow chan-

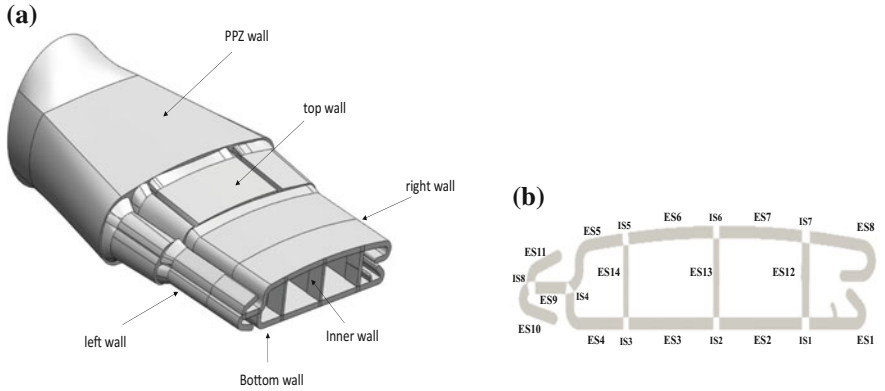


Fig. 2 Swimming pool cover: **a** flow channel; **b** division of the die outlet into ESs and ISs subsections

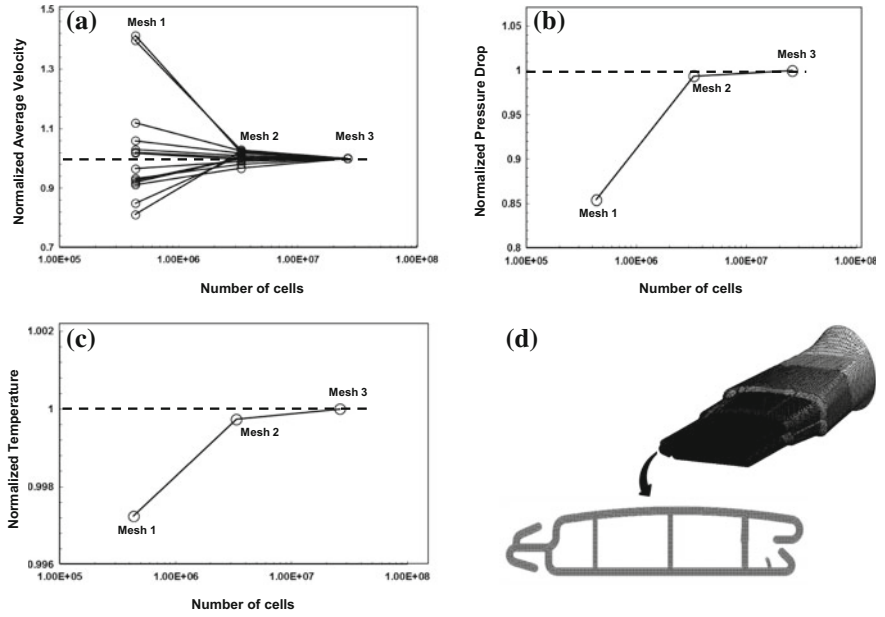


Fig. 3 Results of the mesh refinement study, for the swimming pool cover profile flow channel, normalized with those obtained with the finer mesh: **a** average velocity for all ESs; **b** pressure drop; **c** maximum temperature in the domain; **d** selected mesh (Mesh 2)

nel. The results (average velocity, pressure drop, and temperature) were normalized with the reference field value obtained for the most refined mesh, and are shown in Fig. 3. As one can see from the average velocity field (Fig. 3a), the predicted values for all the ESs, using Mesh2, converge to the value of the most refined one (Mesh3).

The same happens with temperature and pressure drop. Based on these results, and since Mesh3 is computationally/time expensive, Mesh2 (see Fig. 3d) was chosen for the numerical studies.

4.3 Numerical Trials and Results

The initial numerical trial (Trial1) was performed setting 245 °C for all the heaters, to mimic the previous extrusion runs performed at the company. The results obtained, shown in Fig. 4, allow us to conclude that the flow in ES10 and ES11 (located on the left side of the profile; see Fig. 2b) is substantially higher than that required, whereas the opposite happens in ES12 to ES14 (the inner profile walls).

In Trial2, and to compensate for the excessive flow obtained on the left side of the profile, the temperature of this side heater was decreased from 245 to 230 °C (see Fig. 5a). The results show an improvement of the flow distribution, namely a decrease in the flow in ES10 and ES11, and an increase in ES12 to ES14, supporting the potential of the strategy employed. However, ES11 is still the most unbalanced region, with the highest flow rate (with an average velocity 1.8 times higher than the global average one). Therefore, the temperature of the left side was further decreased. The temperatures on the other sides were also decreased to promote an increase in the flow rate in sections ES13 and ES14.

Trial3 was performed with 225 °C at the top and bottom heaters, 230 °C at the right heater and 227 °C at the left one, as indicated in Fig. 5b. The results (see Fig. 4) show a significant improvement, with a decrease in the flow of ES10 and ES11 and

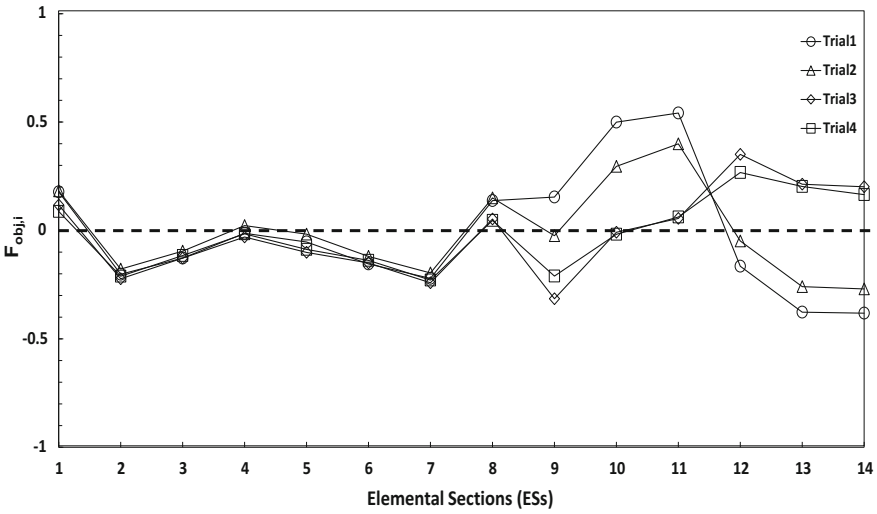


Fig. 4 $F_{obj,i}$ predicted at the die outlet, for four different trials (different combinations of temperatures at the boundaries of the extrusion die)

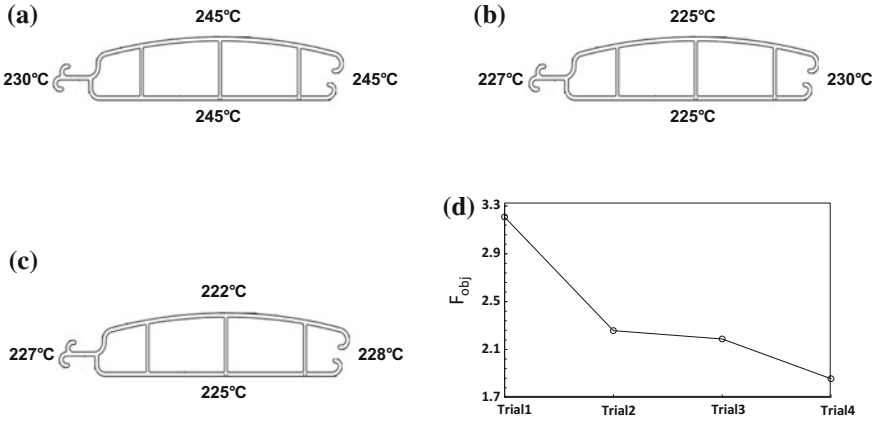


Fig. 5 Die heaters temperature distribution: **a** Trial2; **b** Trial3; **c** Trial4. **d** F_{obj} determined for the different numerical trials performed

an increase in ES12, ES13, and ES14, as expected. However, this trial negatively affected the flow in ES9 (which is now 1.5 times lower than the global average) and in ES12 (which is now slightly higher).

Subsequent numerical trials were performed, following the same procedure, by adjusting the temperatures of the heaters. Trial4 (the last trial), in which the temperatures illustrated in Fig. 5c were used, provided a satisfactory flow distribution. The $F_{obj,i}$ results for this last trial are also represented in Fig. 4. Accordingly, the lowest value for the global objective function F_{obj} (Eq. 2) is also obtained for this trial, as illustrated in Fig. 5d.

Based on these results, the company decided to modify the extrusion die control heating system, enabling the proposed solution to be tested. This modification allowed them to extrude the profile with all the dimensions within the required tolerances. The production line and the extruded profile samples of the swimming pool cover obtained with the proposed solution (Trial4) are shown in Fig. 6.

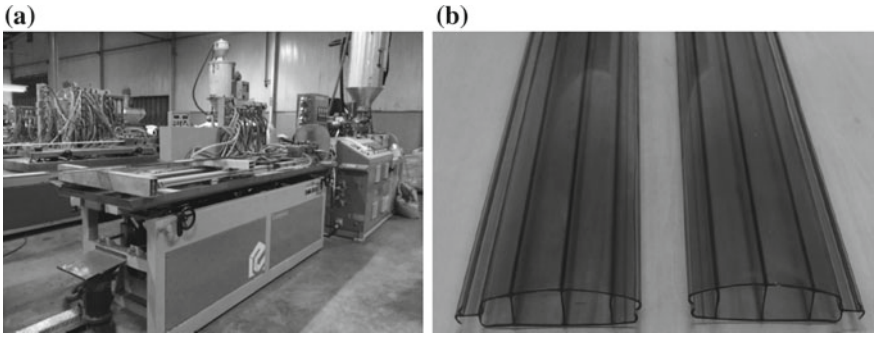


Fig. 6 Extrusion of the swimming pool cover profile: **a** extrusion line in service; **b** extruded samples

4.4 Experimental Assessment

For assessment purposes, the cross section of the profile was imprinted and its soft copy image was used to perform the measurements of each ES and IS, following a protocol described in a previous work [6], using a digital microscope (Leica DMS 1000) and the proper software (Leica Application Suite), see Fig. 7. Then, these relative areas were compared with the numerical predictions. The experimental relative areas and the numerical predictions (shown in Table 1) are in very good agreement, the largest difference being on the order of 2%, for ES7, the thickest section.

Table 1 Relative area ratios for the numerical predictions and experimental results (average and standard deviation measured values) for the swimming pool cover

Subsections	Relative area (%)	Relative area (%)
	Numerical	Experimental
ES1	6.521	6.987 ± 0.309
ES2	10.502	8.968 ± 0.236
ES3	10.456	10.251 ± 0.230
ES4	8.624	9.074 ± 0.335
ES5	8.644	8.796 ± 0.354
ES6	10.562	10.212 ± 0.165
ES7	10.577	8.310 ± 0.268
ES8	8.282	7.655 ± 0.449
ES9	3.321	2.735 ± 0.084
ES10	2.634	2.703 ± 0.072
ES11	1.923	1.757 ± 0.131
ES12	4.782	5.582 ± 0.119
ES13	4.879	5.717 ± 0.233
ES14	4.258	4.846 ± 0.395
IS1	0.532	0.673 ± 0.046
IS2	0.482	0.875 ± 0.054
IS3	0.482	1.004 ± 0.022
IS4	0.298	0.439 ±0.018
IS5	0.506	0.912 ± 0.134
IS6	0.484	0.843 ± 0.013
IS7	0.530	0.743 ± 0.062
IS8	0.722	0.916 ± 0.069

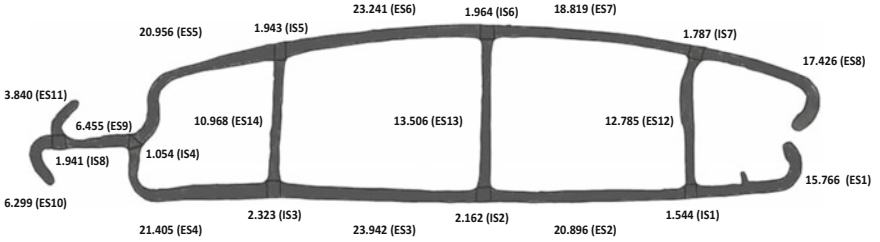


Fig. 7 Cross-sectional areas of the ESs and ISs of one of the extruded samples of the swimming pool cover profile, produced with the temperatures adopted in Trial4 (all the dimensions are in mm^2)

5 Conclusions

In this work, we propose a new numerical approach for balancing extrusion dies, which consists in using differential temperature boundary conditions to control the polymer melt flow distribution. We tested this procedure in a poorly designed existing die, and, by changing the temperature on its different faces, we successfully balanced the flow, avoiding the machining of a new tool. By using the same conditions as those in the numerical simulations, the company was able to produce a profile with all the dimensions within the required tolerances.

The codes used in the numerical trial-and-error procedure were Free CAD (for geometry generation), Salome (for dividing the geometry boundary into subregions), snappyHexMesh (for mesh generation), and OpenFOAM® (for numerical calculations).

In conclusion, the developed computational framework and the proposed die-design methodology proved to be a much better alternative to the usual experimental-based trial-and-error procedure, currently employed by many companies, reducing development costs and the time to market.

Acknowledgements The authors would like to thank for the funding by FEDER through the COMPETE 2020 Programme, the National Funds through FCT—Portuguese Foundation for Science and Technology under the project UID/CTM/50025/2013. L.L. Ferrás would also like to thank for the funding from FCT through the scholarship SFRH/BPD/100353/2014. The authors would also like to acknowledge the Minho University cluster under the project Search-ON2: Revitalization of HPC infrastructure of UMinho, (NORTE-07-0162-FEDER- 000086), co-funded by the North Portugal Regional Operational Programme (ON.2-0 Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF).

References

1. Michaeli W (2003) Extrusion Dies for Plastics and Rubber Design and Engineering Computations, Carl Hanser Verlag GmbH & Co. KG, Munich.

2. Carneiro OS, Nóbrega JM (2012) Design of Extrusion forming Tools, Smithers Rapra Technology Ltd, Shawbury.
3. Gonçalves ND, Carneiro OS, Nóbrega JM (2013) Design of complex profile extrusion dies through numerical modeling. J. Non-Newton. Fluid. Mech. 200:103–110.
4. Nóbrega JM, Carneiro OS, Oliveira PJ, Pinho FT (2003) Flow Balancing in Extrusion Dies for Thermoplastic Profiles Part I: Automatic Design. Int. Polym. Proc. 18:298–306.
5. Nóbrega JM, Carneiro OS, Oliveira PJ, Pinho FT (2003) Flow balancing in extrusion dies for thermoplastic profiles Part II: influence of the design strategy. Int. Polym. Proc. 18:307–312.
6. Nóbrega JM, Carneiro OS, Oliveira PJ, Pinho FT (2004) Flow balancing in extrusion dies for thermoplastic profiles Part III: Experimental assessment. Int. Polym. Proc. 19:225–235.
7. Weller HG, Tabor G, Jasak H, Fureby C (1998) A tensorial approach to computational continuum mechanics using object-oriented techniques. Comput. Phys. 12:620–631.
8. Rajkumar A, Ferrás LL, Fernandes C, Carneiro OS, Sacramento A, Nóbrega JM (2017) Design Guidelines to Balance the Flow Distribution in Complex Profile Extrusion Dies, Int. Polym. Proc. 32:58–71.
9. Falck B, Falck D, Collette B (2012) Freecad [How-To], Packt Publishing Ltd, United Kingdom.
10. Ribes A, Caremoli C (2007) Salome platform component model for numerical simulation. Computer Software and Applications Conference (COMPSAC 2007). 2:553–564.
11. Gisen D (2014) Generation of a 3-D Mesh Using SnappyHexMesh Featuring Anisotropic Refinement and Near-Wall Layers. ICHE Conference Proceedings.
12. Rajkumar A (2017) Improved methodologies for the design of extrusion forming tools (Ph.D. thesis), University of Minho, Portugal.

Drag Model for Coupled CFD-DEM Simulations of Non-spherical Particles

Rolf Lohse and Ulrich Palzer

Abstract The production and handling of non-spherical granular products plays an important role in many industries. It is often necessary to consider the real particle shape of the real particles as an essential prerequisite for modeling these processes reliably. This work presents a new approach for approximating the drag coefficient of non-spherical particles during simulation. This is based on the representation of the particle shape as a clump of multiple spheres, as it is often used in the Discrete Element Method (DEM). The paper describes the calculation of the drag coefficient based on the arrangement of the spheres within the clump depending on the Reynolds number and the flow direction. Numerical simulations of the flow around regularly- and irregularly shaped particles, as well as experiments in a wind tunnel, are used as the basis of model development. The new drag model is able to describe the drag coefficient for irregularly shaped particles within a wide range of Reynolds numbers. It has been implemented in the toolbox CFDEM® coupling. The new drag model is tested within CFD-DEM simulations of particle behavior in a spouted bed.

1 Introduction

Particle-laden multiphase flows are relevant in various areas of process technology. Typical examples are fluidized or spouted beds, pneumatic conveying of granular media and mixing and separation processes. The accurate prediction of the physical behavior of the particles and the continuous fluid phase is important for the three-dimensional modeling of these processes. At higher particle volume concentrations, the interactions among the particles and between the fluid and dispersed phase become more decisive. Also, the shape of the particles has an essential influence on the

R. Lohse (✉)
Weimar Institute of Applied Construction Research,
Über der Nonnenwiese 1, 99428 Weimar, Germany
e-mail: r.lohse@iab-weimar.de

U. Palzer
e-mail: u.palzer@iab-weimar.de

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_9

particle behavior. For this reason, the simulation with a coupling of the Discrete Element Method (DEM) and CFD methods is the preferred choice, if the computational expense is justified.

2 Modeling of Non-spherical Particles

The shape of a non-spherical particle is often characterized by its sphericity, which is the ratio between the surface of a volume-equivalent sphere and the surface of the real particle. This method is used in many drag models [1, 2]. Other possibilities are the Corey shape factor used by Swamee et al. [3] and the lengthwise and crosswise sphericity used by Leith [4] and Hölzer et al. [5].

In common DEM software (EDEM [6], PFC [7]), a multi-sphere approach is used to model non-spherical particles for bulk mechanics, wherein an arbitrarily shaped particle is approximated by a clump of different spheres (Fig. 1). In the presented work, this multi-sphere arrangement is also used to calculate the drag forces of the particles resulting from the fluid flow in CFD-DEM simulations. Thus, for every sphere within the clump, a partial drag force $F_{D,i}$ is calculated based on the position of the sphere within the clump, the overlap with other particles and the flow direction. The sum of these values over all spheres within a clump gives the drag force of the non-spherical particles F_D, which depends on the orientation of the particle to the fluid flow and the Reynolds number.

2.1 Drag Forces on Non-spherical Particles

There are many studies on the drag force coefficient of spheres and non-spherical particles. These include experimental investigations based on settling experiments and wind tunnel tests [8–10]. The influence of the alignment of the particle to the flow is not studied in detail. It is only distinguished in a lengthwise or crosswise flow. Numerical investigations are available for selected particle shapes. Ellipsoids and discs are studied in [11–13], which also include the influence of the angle of

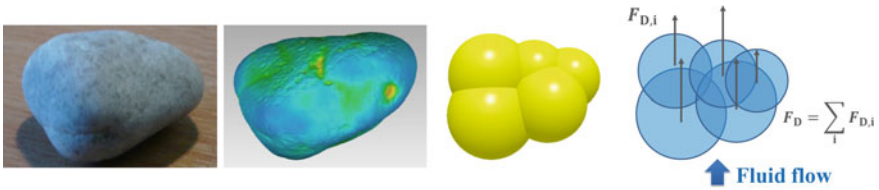


Fig. 1 Transition from a real particle over a 3D scan model to a multi-sphere clump in DEM to the approximation used for the calculation of the drag forces of a non-spherical particle

attack. There are some reliable correlations for the drag coefficient of shaped particles [1–3, 14], which contains the Reynolds number and the sphericity as parameters. Newer formulations by Hölzer and Sommerfeld [5] also capture the influence of the alignment of the particle to the flow. This is done through the usage of the lengthwise and crosswise sphericity of the particles. In the present work, the influence of the particle shape, the Reynolds number and the angle of attack is studied within a wider range. The data from the literature were supplemented with additional CFD calculations and experimental investigation. The approach used is explained below.

2.1.1 Numerical Investigations

Computational fluid dynamics (CFD) is used to calculate the drag of non-spherical particles. The investigations are carried out based on steady state and transient simulations on a stationary particle, with the exact particle shape in an incompressible, turbulent air flow. The SST k–ω model is used for the modeling of the turbulence. The CFD toolbox OpenFOAM® 2.3.x [15] is used to solve the Reynolds-averaged Navier–Stokes (RANS) equation for the flow around the investigated particle shapes, which are shown in Fig. 2. These include both regularly shaped particles, such as cylinders and ellipsoids, and irregularly shaped particles, which were obtained through a 3D scanner. First, simulations for the flow around a sphere were used to validate the computational model. It was found that the drag forces were calculated in good

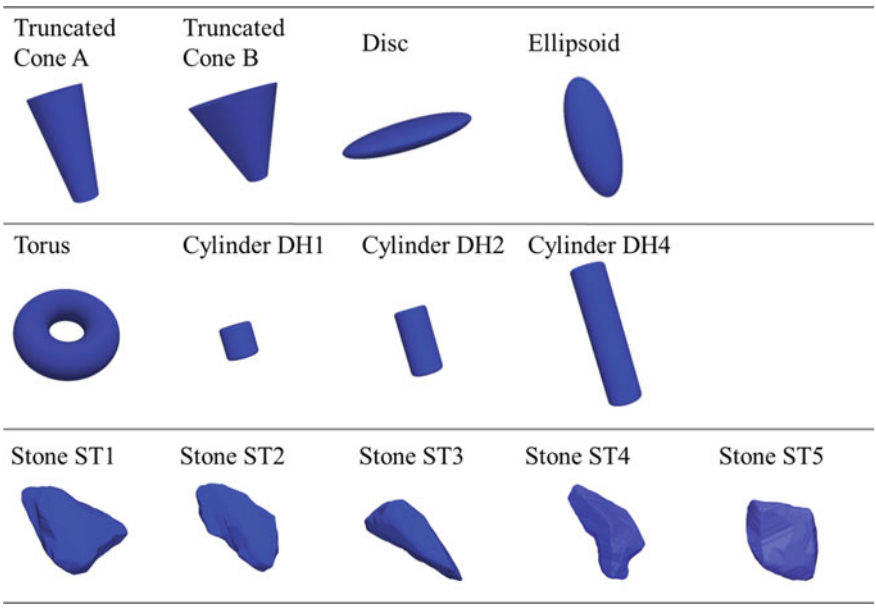


Fig. 2 Overview of the investigated particle shapes

accordance with the known models from the literature [16, 17] up to a Reynolds number of about 14,000. For the simulation of the flow around the arbitrarily shaped particle, the numerical investigations are carried out for Reynolds numbers ranging from 1 to 5000.

2.1.2 Experimental Investigations

A wind tunnel was used for experimental studies of the drag force on non-spherical particles. This Göttinger-Type wind tunnel, build by Westenberg Engineering, has a rectangular-shaped nozzle of 300 × 300 mm and an open test section of 600 mm length (Fig. 3). The possible air velocities are between 0.5 and 40 m/s. The alignment of the particle to the flow can be defined by the mounting device, which is connected to a six-axis force sensor so as to measure the drag force onto the investigated object. The validation of the measurement device was performed with different-sized spheres at the full velocity range of the wind tunnel. Further trials with ellipsoids showed that a minimal air velocity of 8 m/s was sufficient for the investigated particle sizes to ensure a good reproducibility and accuracy of the measurements.

The experiments were performed for selected particle shapes from Fig. 2. This involved truncated cone A, an ellipsoid and different cylinders. The particles were produced with a 3D printer, wherein an adapter for the connection with the measuring device was integrated. The study with the wind tunnel included 80 measurement series of five–eight variants of the particle Reynolds number. The drag forces were measured at flow rates from 5 to 35 m/s. The resulting particle Reynolds numbers were between 15,000 and 40,000. The angle of attack was changed stepwise with ϕ = 0, 30, 45, 75 and 90° (ellipsoid, cylinder) and ϕ = 0, 45, 90, 135 and 180° (truncated cone).

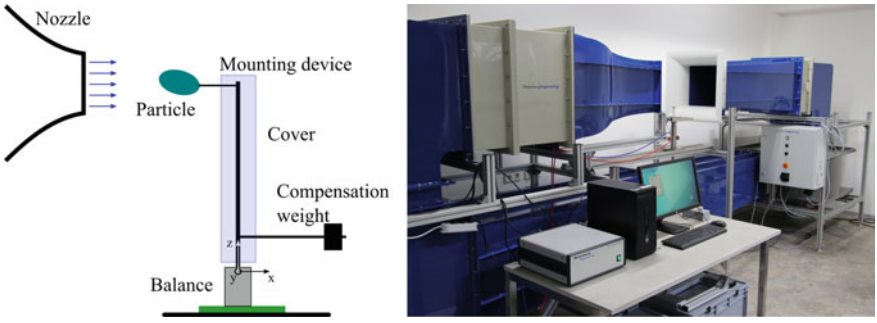


Fig. 3 Schematic of the measurement arrangement (left) and closed wind tunnel test facility (right)

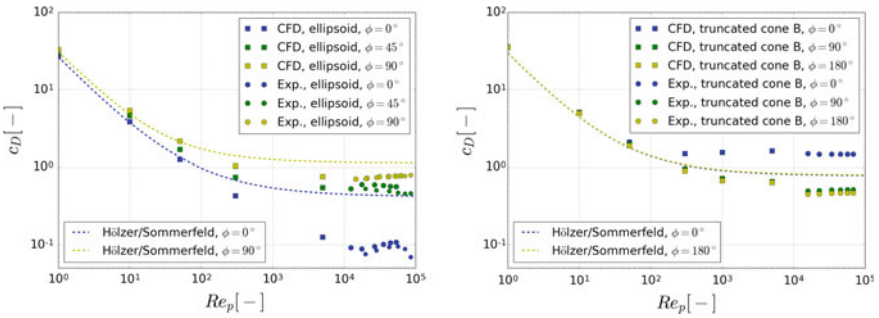


Fig. 4 Results of the numerical simulations and wind tunnel test for the drag coefficient of the ellipsoid (left) and the truncated cone (right) for different angles of attack and Reynolds numbers

2.1.3 Results

The results of the investigation from the wind tunnel tests and the numerical simulation are shown in Fig. 4. In addition, the data are compared with the drag model by Hölzer and Sommerfeld [5]. It can be seen that the numerical simulations fit well with the experimental results for the ellipsoid. Both sets of data are below the model by Hölzer and Sommerfeld, especially at higher Reynolds numbers. The results of the simulation and the experiments show a small jump for the truncated cone around $Re = 10{,}000$. It has not yet been fully clarified as to whether the reason is to be found in the measurement or in the simulation approach.

The simulation and the experimental data for the truncated cone show a clear influence of the angle of attack on the drag coefficient. This behavior cannot be reproduced by the drag model by Hölzer and Sommerfeld. There is no difference shown between the alignment to the flow for 0° and 180°. It can be recognized from the wind tunnel tests that the drag coefficient is nearly independent of the flow velocity for the investigated Reynolds numbers above 10,000. The known drag models from the literature confirm this behavior. Only the particle shape and the angle of attack influence the drag coefficient value.

3 Drag Model Development

The non-spherical particle shape is represented by a clump of multiple spheres. The spheres can have different diameters and can overlap each other. The idea of this model approach is to estimate the drag coefficient in two steps. First, the acting drag force for each sphere within the clump is approximated in dependence on the flow direction, position, overlapping and shading of each sphere. With the sum of the drag forces on each sphere, the drag coefficient $c_{D,S}$ can be determined for the particle. In the literature, many models exist for the drag of single spheres. Morrison's model

[16] covers a very wide range of Reynolds numbers and is used as the basis for the new drag model.

$$c_{D,S} = \frac{24}{Re_S} + \frac{2.6\left(\frac{Re_S}{5.0}\right)}{1+\left(\frac{Re_S}{5.0}\right)^{1.52}} + \frac{0.411\left(\frac{Re_S}{263000}\right)^{-7.94}}{1+\left(\frac{Re_S}{263000}\right)^{-8.00}} + \frac{{Re_S}^{0.80}}{461000}\,. \tag{1}$$

A sphere Reynolds number is calculated for each sphere within the clump in dependence on the diameter and the relative velocity between particle and fluid flow. The drag force of the sphere $F_{D,S}$ is calculated including a weight factor W

$$F_{D,S} = W\frac{c_{D,S}\,\rho}{2}{U_{rel}}^2\frac{\pi}{4}{d_S}^2, \tag{2}$$

where U_{rel} is the local relative velocity, d_S the diameter of the sphere and ρ the fluid density. With the sum of the drag forces of all spheres, the drag coefficient of the particle c_D can be determined

$$c_D = \frac{\sum F_{D,S}}{\frac{\rho}{2}{U_{rel}}^2\frac{\pi}{4}{d_V}^2}\,, \tag{3}$$

where d_V is the diameter of the volume-equivalent sphere. The weighting factor W has to be defined as a function of the position of the sphere within the clump, the influence of the particle Reynolds number and the alignment of the clump to the flow. Different combinations of several parameters that describe the position and the size of the spheres inside the clump are investigated for the definition of the weighting factor. Some examples of these parameters are listed below and explained in more detail in Fig. 5.

- Volume of a sphere, which is not overlapped by another sphere
- Relative free surface a_{fs}, which is the ratio of the surface area, having direct contact with the surrounding fluid A_{fs} and sphere surface A_S ($a_{fs} = A_{fs}/A_S$)
- Incident flow surface ratio of the sphere ($a_{sh} = A_{sh}/A_S$)
- Incident flow surface ratio of the sphere with reversed flow direction ($a_{bs} = A_{bs}/A_S$)
- Diameter of a sphere d_S
- Distance to the leading sphere s (parallel to the flow vector).

Different functional combinations from these parameters were examined for the weighting factor, including the sphere Reynolds number. The best variant so far is given by

$$W = A\ {Re_S}^B\left(C\ {a_{sh}}^D + E\ {a_{bs}}^F\right) + G\ {Re_S}^H e^{\left(I\ {a_{sh}}^J + K\ {a_{bs}}^L\right)} + M\ {Re_S}^N\left(O\ {a_{fs}}^P\right), \tag{4}$$

with A to P as model coefficients.

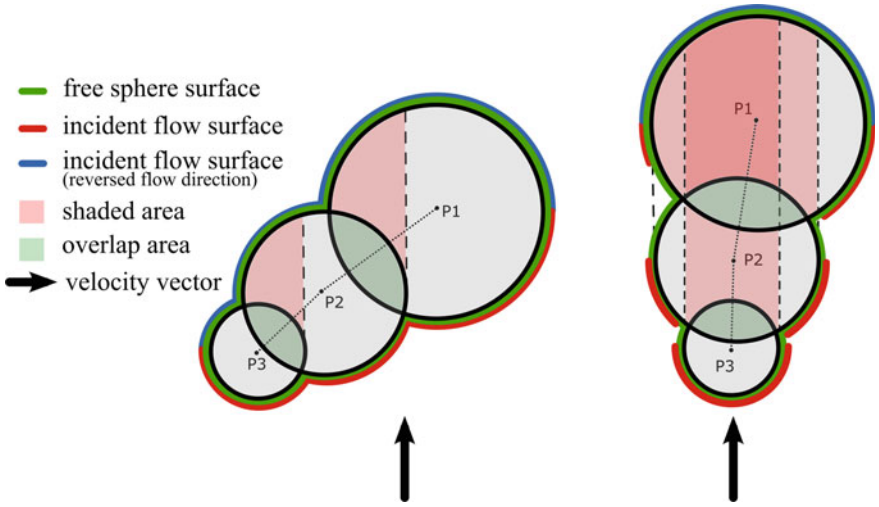


Fig. 5 Visualization of the parameters for the calculation of the weighting factor W

The values of the coefficients contained in the model functions for the weighting factor were determined using a fitting algorithm. The definition is based on the results of CFD simulations for selected particle shapes. These particles were discretized by a clump of spheres with various sphere numbers. The weighting factor is determined by Eq. 4 for each of these spheres and the appropriate drag coefficient is calculated by Eq. 3. The coefficients in Eq. 4 are adjusted to fit best the values from the CFD calculation for a wide range of Reynolds numbers and angles of attack by an optimization algorithm. Thereby, a particle swarm approach is used. The problem is implemented in combination with the python program deap [18], which processes the optimization on a multi-processor platform in parallel.

The coefficients used in Eq. 4 are listed in Table 1. The resulting drag coefficient for a truncated cone and an irregularly shaped particle are shown in Fig. 6 in dependence on the Reynolds number and the angle of attack. The new drag model gives a better agreement with the CFD results in comparison to the model by Hölzer and Sommerfeld.

Table 1 Shape-independent parameters used for calculation of the weighting factor in Eq. 4

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
A	0.272444	E	0.860548	I	2.993242	M	1.689610
B	0.179754	F	0.711180	J	2.077819	N	0.127088
C	−0.380534	G	0.034872	K	0.216286	O	−0.004679
D	5.650548	H	−0.021070	L	0.256386	P	−0.097940

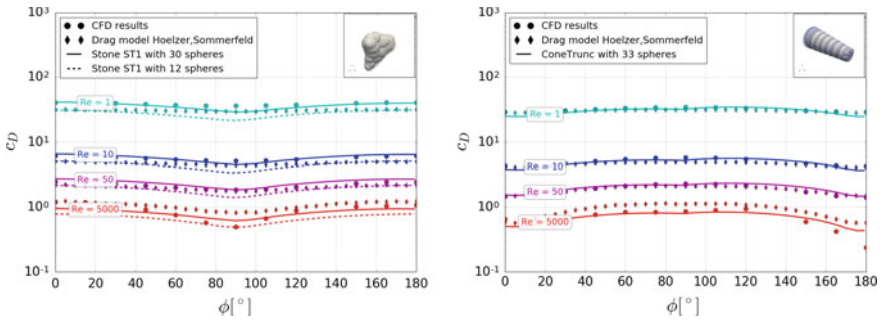


Fig. 6 Drag coefficient for the new model depending on the Reynolds number and particle orientation in comparison to CFD data and the drag model by Hölzer and Sommerfeld [5]; truncated cone (left) and stone ST1 (right)

4 Application

The new drag model will be tested on a complex flow configuration. Therefore, experimental investigations on a spouted bed configuration are processed. The toolbox CFDEM® coupling is used to calculate the behavior of the fluid and the granular phase. This is done through a coupling of the CFD code OpenFOAM® [15] and the DEM code LIGGGHTS [19]. The momentum conservation equation for the fluid phase uses the incompressible formulations according to a pressure gradient force model (PGF or model A [20, 21]) that is implemented in CFDEM® coupling as follows:

$$\frac{\partial \varepsilon_f U_i}{\partial t} + \frac{\partial \varepsilon_f U_i U_j}{\partial x_i} = -\varepsilon_f \frac{\partial p}{\partial x_i} + \frac{\partial \varepsilon_f \tau}{\partial x_i} + \varepsilon_f g - F_{pf}, \tag{5}$$

where ε_f is the fluid phase fraction, U the fluid velocity, τ the shear tensor, and F_{pf} the sum of the volume forces for the coupling between the fluid and the granular phase. The simulations are performed with the cfdemSolverPiso solver of CFDEM® coupling [19] using a four-way coupling approach that considers the particle–particle and particle–fluid interactions. The new drag model is valid for the flow around a single particle. In order to take into account effects resulting from locally high particle volume concentrations (swarm effects), the drag model is combined with a correlation by DiFelice [22]. This approach was also used by Hilton et al. [23] and Oschmann et al. [24]. The drag force F_D also depends on the local particle volume concentration ε_f and the model parameters χ

$$F_D = \frac{c_D \rho}{2} U^2 \frac{\pi}{4} {d_V}^2 {\varepsilon_f}^{(1-\chi)}, \tag{6}$$

where

$$\chi = 3.7 - 0.65 exp\left(-\frac{(1.5 - log(Re))^2}{2}\right). \tag{7}$$

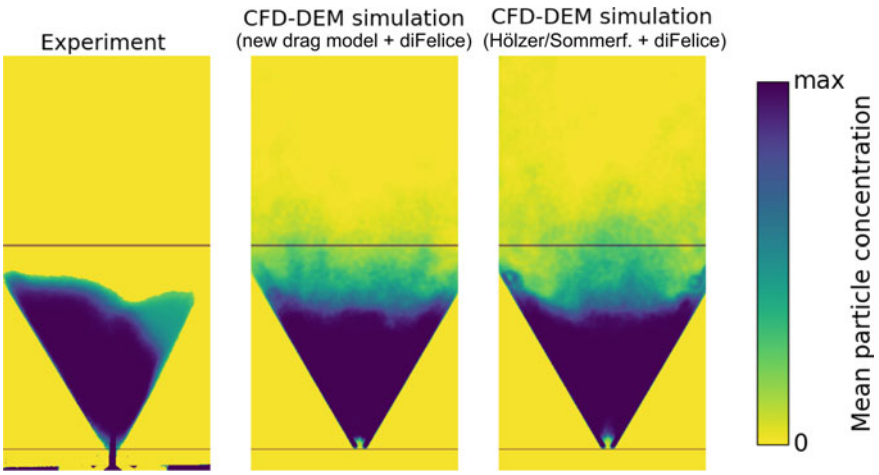


Fig. 7 Mean particle distribution within the spouted bed for the experimental and numerical investigations with 1000 tori: experiments (left), CFD-DEM calculation with the new drag model (middle) and the drag model by Hölzer and Sommerfeld [5] (right)

The combination with DiFelice's approach was implemented and tested, both for the newly developed drag model and the drag model by Hölzer and Sommerfeld. The grid resolution for spouted bed geometry was chosen such that the flow field is still well-discretized and most grid cells are larger than the diameter of the Clump spheres. In the area of the wall and the inlet region, the cell size is much smaller than the spheres used. So, the local particle concentration may not be calculated correctly in these regions. A resolved CFD-DEM method presented in [25] could lead to better results, but requires higher computational costs and was not tested within this work. The simulation was performed for a real time of 30 s.

The experimental setup uses a camera in combination with a backlight to describe the particle behavior. A time series of 300 images over 5 min is used to determine the average bed dimension. The results of the simulations for the motion of tori in the spouted bed are shown in Fig. 7. All performed CFD-DEM simulations show a similar fluidization of particles compared to the experiment. The average calculated bed height is similar to the experiments. The particle velocities close to the walls are higher, so that particles in this region are transported well above the average bed height, a phenomenon that has not been observed experimentally. The differences between experiments and simulations in the inlet region are due to the experimental setup, whereas this region was not sufficiently well-lit.

There are only minor differences between the results with the new drag model and those with the model by Hölzer and Sommerfeld. However, the whirling up of a few particles at the wall is more pronounced with the model by Hölzer and Sommerfeld. The results with the new drag model show a slightly better agreement with the experimental results.

5 Conclusions

A new model is presented to approximate the drag coefficient of non-spherical particles using a clump of multiple spheres to represent the shape of the particle. Numerical simulation and wind tunnel experiments are performed to determine the drag forces of regularly and irregularly shaped particles for a wide range of Reynolds numbers and particle orientations. These data are used to determine the necessary model parameters. The drag model shows a good fit with the CFD results within a wide range of particle shapes, Reynolds numbers and particle orientations. It gives, in some cases, better results than the widely used model by Hölzer and Sommerfeld [5]. The new drag model is implemented within the CFDEM® coupling environment. It is used in combination with the correlation by DiFelice [22] to capture the influence of high particle concentrations on the drag values. The application of the drag model for the simulation of non-spherical particles in spouted beds by means of CFD-DEM calculations gives a good agreement with experimental results.

Acknowledgements This work was funded by the German Federal Ministry of Economy and Technology (BMWi) in the framework of the INNO-KOM-Ost project under grant VF130034.

References

1. Haider, A., Levenspiel, O.: Drag coefficient and terminal velocity of spherical and nonspherical particles. Powder Technology **58**, 63–70, (1989)
2. Loth, E.: Drag of non-spherical solid particles of regular and irregular shape. Powder Technology **182**, 342–353, (2008)
3. Swamee, P., Ojha, C.: Drag Coefficient and Fall Velocity of nonspherical particles. Journal of Hydraulic Engineering **117**, 660–667, (1991)
4. Leith, D.: Drag on Nonspherical Objects. Aerosol Science and Technology **6**, 153–161, (1987)
5. Hölzer, A., Sommerfeld, M.: New simple correlation formula for the drag coefficient of non-spherical particles. Powder Technology **184**, 361–365, (2008)
6. DEM-Solutions, EDEM version 2.7, Edinburgh (2015)
7. Itasca Consultings GmbH, PFC3D version 4.0, Gelsenkirchen, (2015)
8. Becker, H.: The Effects of Shape and Reynolds Number on Drag in the Motion of a Freely Oriented Body in an Infinite Fluid. Can. J. Chem. Eng. **37**, 85–91, (1959)
9. Kasper, G., Niida, T., Yang, M.: Measurements of Viscous Drag on Cylinders and Chains of Spheres with Aspect Ratios Between 2 and 50. Journal of Aerosol Science **16**, 535–556, (1985)
10. Pettyjohn, E. S., Christiansen, E. B.: Effect of Particle Shape on Free-Settling Rates of Isometric Particles. Chemical Engineering Progress **44**, 157–172, (1948)
11. Dwyer, H. A., Dandy, D. S.: Some influences of particle shape on drag and heat transfer. Physics of Fluids A: Fluid Dynamics **2**, 2110–2118, (1990)
12. Hölzer, A.: Bestimmung des Widerstandes, Auftriebs und Drehmoments und Simulation der Bewegung nichtsphrischer Partikel in laminaren und turbulenten Strmungen mit dem Lattice-Boltzmann-Verfahren. Martin-Luther-Universitt Halle-Wittenberg, 2007
13. Zastawny, M., Mallouppas, G., Zhao, F., van Wachem, B.: Derivation of drag and lift force and torque coefficients for non-spherical particles in flows. International Journal of Mineral Processing **39**, 227–239, (2012)
14. Ganser, G. H.: A rational approach to drag prediction of spherical and nonspherical particles. Powder Technology **77**, 143–152, (1993)

15. OpenCFD Ltd. (ESI-Group), OpenFOAM® version 2.3, Bracknell, UK (2014)
16. Morrison, F. A.: Data Correlation for Drag Coefficient for Sphere. Department of Chemical Engineering, Michigan Technological University, Houghton, MI (2013) Available via http://www.chem.mtu.edu/~fmorriso/DataCorrelationForSphereDrag2013.pdf
17. Schiller, L., Naumann, A.: Über die grundlegenden Berechnungen bei der Schwerkraftaufbereitung. Zeitschrift Verein Deutscher Ingenieure **77**, 318–320, (1933)
18. Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., Gagn, C.: DEAP: Evolutionary Algorithms Made Easy. Journal of Machine Learning Research **13**, 2171–2175, (2012)
19. Goniva, C., Kloss, C., Deen, N., Kuipers, J., Pirker, S.: Influence of Rolling Friction Modelling on Single Spout Fluidized Bed Simulations. Particuology **5**, 582–591, (2012)
20. Kafui, K. D., Thornton, C., Adams, M. J: Reply to comments by Feng and Yu on "Discrete particle-continuum fluid modelling of gas-solid fluidised beds" by Kafui et al. Chemical Engineering Science **59**, 719–722, (2004)
21. Zhou, Z. Y., Kuang, S. B., Chu, K. W., Yu, A. B.: Discrete particle simulation of particle–fluid flow: model formulations and their applicability. Journal of Fluid Mechanics **661**, 482–510, (2010)
22. Di Felice, R.: The voidage function for fluid–particle interaction systems. International Journal of Multiphase Flow **20**, 153–159, (1994)
23. Hilton, J., Cleary, P. (2009) The Role of Particle Shape in Pneumatic Conveying. In: Proceedings of the Seventh International Conference on CFD in the Minerals and Process Industries (CSIRO)
24. Oschmann, T., Vollmari, K., Kruggel-Emden, H., Wirtz, S.: Numerical investigation of the mixing of non-spherical particles in fluidized beds and during pneumatic conveying. Procedia Engineering **102**, 976–985, (2015)
25. Hager, A., Kloss, K., Pirker, S., Goniva, C. (2012) Parallel Open Source CFD-DEM for Resolved Particle-Fluid Interaction. In: Proceedings of the Ninth International Conference on CFD in the Minerals and Process Industries (CSIRO)

Effects of Surface Textures on Gravity-Driven Liquid Flow on an Inclined Plate

Martin Isoz

Abstract Even though free surface flows are of high importance in a number of engineering areas, they still pose a challenging problem from the point of view of Computational fluid dynamics (CFD) modeling. In the present work, a Volume-of-fluid (VOF) method-based open source CFD solver, interFoam, is used to study the properties of a gravity-driven liquid flow on an inclined plate with respect to different plate textures. At first, the proposed model was validated against the available experimental data. Then, the effects of three different types of texture on the specific wetted area of the plate were evaluated.

1 Introduction

The multiphase flow of thin films, rivulets and drops is of key importance throughout many areas of chemical engineering, including mass transfer [2], trickle bed reactors [8], heat exchangers [14] and various coating processes [7]. However, on an industrial scale, most of these processes are performed in complex apparatuses, the hydrodynamics of which still remains rather mysterious.

Examples of such apparatuses are the separation columns that are used in chemical engineering to perform mass transfer operations on large scales. The fluid flow in these columns is multiphase and it occurs in a geometrically complex domain (see Fig. 1).

Due to the geometrical complexity of the column packing and the size of the columns themselves, a direct simulation of its hydrodynamics has yet to be developed. However, even models of the liquid flow on simplified geometries may shed some light on the behavior of the original system.

M. Isoz (✉)
University of Chemistry and Technology in Prague,
Technicka 5, 166 28 Prague 6, Dejvice, Czech Republic
e-mail: isozm@vscht.cz

M. Isoz
Institute of Thermomechanics of CAS,
Dolejskova 5, 182 00 Prague 8, Ladvi, Czech Republic

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_10

Fig. 1 Example of separation column packing – Mellapak 250X. An overall view of the packing is depicted on the left. On the right, a detail of one element of a dismantled packing is shown. Structured packing usually consists of corrugated, perforated and textured steel plates. The standard mode of operation is counter-current: the liquid phase flows down and the gas phase up

For example, for the modeling of separation columns, Raynal and Royon-Lebeaud [11] proposed an appproach based on a segmentation of the problems into three ranges of spatial scale: (i) the micro-scale represented by the liquid film flow on a small subsection of the structured packing, (ii) the meso-scale corresponding to one element of the packing material, and (iii) the macro-scale, which coincides with the full column. In terms of Raynal and Royal-Lebeaud's classification, we are interested in the micro-scale. Despite the overall geometric complexity of the structured packing, if we choose a small enough section of it, it is possible to represent such a packing as an inclined plate.

The properties of various types of liquid flows down inclined plates have been studied both experimentally and theoretically since, 1960s [13]. However, the recent developments in experimental techniques, numerical mathematics and available computing power have enabled a truly rapid advances in the area. The gravity-driven flow of various liquids on an inclined plate was studied by Hoffmann et al. [5, 6], who performed a series of experiments and numerical simulations in order to establish the specific wetted area $a_S = a_W/a_T \in [0, 1]$ (the ratio of the wetted and total packing area) of an inclined plate in dependence on the liquid properties and the liquid volumetric flow rate.

The specific wetted area is a key parameter for the mass transfer calculations, as the full film ($a_S = 1$) has significantly better mass transfer characteristics than rivulets or sliding drops. Hence, the work of Hoffmann et al. has led to a substantial amount of subsequent studies aimed at examination and characterization of the effects of different textures on the gravity-driven liquid flow (see [2, 3, 12, 15] and references therein).

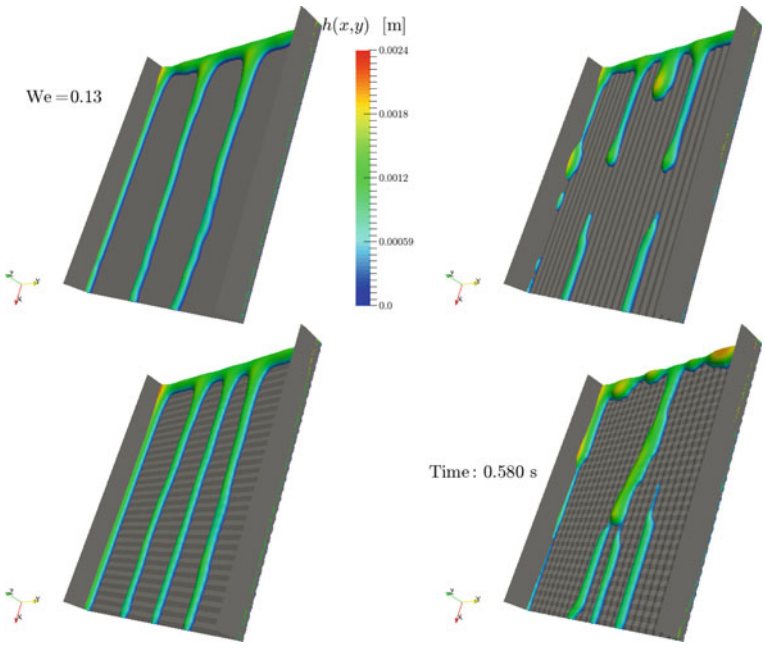


Fig. 2 Examples of a liquid flowing down a smooth plate (upper left) and plates equipped with longitudinal (upper right), transversal (lower left) and pyramidal (lower right) textures. Texture roughness is 0.2 mm (∼10% of the film thickness) and the width of one texture fold is 2 mm

In the present study, we aim to provide characteristics for a gravity-driven free surface flow of a liquid on an inclined plate. We consider four different surface treatments of the investigated plate; namely, no surface treatment, i.e., a smooth plate, a plate with either a longitudinal or transversal texture, and a plate with a pyramidal type of surface texture. Examples of the studied geometries may be found in Fig. 2. The studied flow parameter is the specific wetted area, a_S, of the plate. Because of the complex nature of the simulated flow, we perform a three-dimensional VOF simulation of an isothermal liquid film. The solver used was interFoam, an open-source solver from the OpenFOAM® framework [9, 10].

At first, we compare our simulations for the case of a smooth plate with existing experimental data published by Hoffman et al. [5, 6] and with the published CFD results [2, 3, 12, 15]. Then, we proceed to the simulations of a liquid flow on the plate equipped with either longitudinal, transversal or pyramidal texture. The study is completed by an evaluation of the influence of the pyramidal texture parameters on the liquid wetting properties.

2 Numerical Model

As was mentioned above, the interFoam solver from the OpenFOAM® toolbox [9, 10] was used for the purpose of the present investigation. OpenFOAM® is a set of open-source C++ libraries used to solve different problems of computational continuum mechanics within the finite volume method (FVM) framework. Nonetheless, the main focus of the toolbox lies within the field of CFD. Bellow, we first describe the basic principles of the interFoam solver, and then present the computational domain and the simulation set-up.

2.1 Model Equations

The interFoam solver is an FVM solver for multiphase problems. The solver is based on the volume-of-fluid (VOF) method [4]. The VOF method can be used to solve the multiphase flow of incompressible isothermal and immiscible fluids. The method is based on solving a single set of the Navier-Stokes equations for a hypothetical liquid. The properties of the given liquid, namely its density, ρ, and dynamic viscosity, μ, are calculated as a weighed average of the corresponding properties of the individual fluids. If we consider two immiscible fluids, A and B, the final hypothetical fluid properties would be calculated as,

$$\begin{aligned} \rho &= \alpha\rho_A + (1-\alpha)\rho_B\,, \\ \mu &= \alpha\mu_A + (1-\alpha)\mu_B\,. \end{aligned} \tag{1}$$

In the VOF methods, the weight in the above equations, α, corresponds to the volume fraction of the reference liquid, A, in the individual mesh cells,

$$\alpha = \begin{cases} 1, & \text{cell contains only phase } A \\ 0, & \text{cell contains only phase } B \\ a \in (0,1) & \text{cell contains gas-liquid interface}\,. \end{cases} \tag{2}$$

As a result of the introduction of the weight function, α, it is necessary to complete the Navier-Stokes equations by an advection equation for it. Furthermore, if we assume the velocity field, $\mathbf{U}$, to have the form of

$$\mathbf{U} = \alpha\mathbf{U}_A + (1-\alpha)\mathbf{U}_B\,, \tag{3}$$

it is possible to derive a Eulerian two-fluid model, in which the advection equation for α is,

$$\partial_t\alpha + \nabla\cdot(\mathbf{U}\alpha) + \nabla\cdot((\mathbf{U}_r\alpha(1-\alpha)) = 0\,, \tag{4}$$

where $\mathbf{U}_r = \mathbf{U}_A - \mathbf{U}_B$ is the interface compression velocity. The corresponding Navier-Stokes equations are, as implemented in the interFoam solver,

$$\begin{gathered} \partial_t (\rho \mathbf{U}) + \nabla \cdot (\rho \mathbf{U} \otimes \mathbf{U}) - \nabla \cdot (\mu \nabla \mathbf{U}) = -\nabla p_d - \mathbf{g} \cdot \mathbf{h} \nabla \rho + \mathbf{F}_s \\ \nabla \cdot \mathbf{U} = 0 \,, \end{gathered} \tag{5}$$

where $\mathbf{g}$ is the gravitational acceleration, $\mathbf{h} = \mathbf{x} - h_{ref} |\mathbf{g}|/||\mathbf{g}||$ and $p_d = p - \rho \mathbf{g} \cdot \mathbf{h}$ is the dynamic pressure in the liquid and $\mathbf{F}_s$ are the surface forces exerted on the liquid.

The surface forces, $\mathbf{F}_s$, are evaluated via the continuous surface force (CSF) model [1],

$$\mathbf{F}_s = \gamma \, \mathbf{n} \cdot \kappa(\mathbf{x}) \,, \tag{6}$$

where γ is the surface tension coefficient, $\mathbf{n}$ is the outer unit normal to the gas-liquid interface and κ is the mean surface curvature. Moreover, it is possible to evaluate the interface normal vector, $\mathbf{n}$, and the interface curvature, κ, based on the α function as follows:

$$\begin{aligned} \mathbf{n} &= \tfrac{\nabla \alpha}{||\nabla \alpha||} \\ \kappa(x) &= \nabla \cdot \mathbf{n} \,. \end{aligned} \tag{7}$$

In the cells adjacent to a wall boundary, $\mathbf{n}$ has to be adjusted in order to take into account the liquid dynamic contact angle, θ,

$$\mathbf{n}_{wall} = \mathbf{n}_w \cos\theta + \mathbf{t}_w \sin\theta \,, \tag{8}$$

where $\mathbf{n}_w$ and $\mathbf{t}_w$ are, in order, the unit vectors normal and tangent to the wall.

For the purpose of the present study, a dynamic contact angle model was applied. This model is available in the core OpenFOAM® installation as dynamicAlphaContactAngle. In the model, the dynamic contact angle, θ, is defined as,

$$\theta = \theta_0 + (\theta_A - \theta_R) \tanh\left(\frac{u_w}{u_\theta}\right) , \tag{9}$$

where θ_0, θ_A and θ_R are the equilibrium, advancing and receding contact angles, respectively, u_w is the speed of the moving contact line relative to the speed of the adjacent wall and u_θ is an appropriate scale for u_w.

2.2 Computational Domain and Simulation Set-Up

The computational domain in the present investigation was, in agreement with the available data [2, 5, 6], a steel plate of length $L = 0.06\,\text{m}$ and width $W = 0.05\,\text{m}$, which corresponds to a small section of the packing material. The height of the computational domain was specified as $H = 0.007\,\text{m}$. The liquid inlet had height

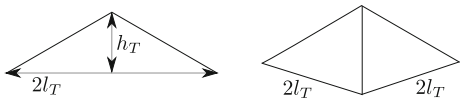


Fig. 3 The surface texture parameters: texture density, l_T, and roughness, h_T. For the case of the pyramidal texture, a square base pyramid, as depicted in the right side of the figure, was used. For the cases of longitudinal and transversal texture, the texture had the shape of a prism, with one base dimension specified by $2l_T$ and the other by the corresponding size of the geometry (L for the longitudinal texture and W for the transversal texture). The height of the prism was h_T

$H_I = 0.4\,\text{mm}$ and spanned across the width of the plate. The detailed structure of the computational domain varied depending on the parameters of the texture used. These parameters were the texture's overall structure, where we used the above specified four different types of texture, and the texture's density, l_T, and roughness, h_T, as specified in Fig. 3.

The simulations were performed with the plate inclination angle φ fixed at $\pi/3$ to the horizontal.[1] The investigated pair of fluids was water and air, which we denote as A and B, respectively. The densities of the fluids used were $\rho_A = 997\,\text{kg}\,\text{m}^{-3}$ and $\rho_B = 1.18\,\text{kg}\,\text{m}^{-3}$. The dynamic viscosities of the constituent phases were, $\mu_A = 8.899 \times 10^{-3}\,\text{Pa}\,\text{s}$ and $\mu_B = 1.831 \times 10^{-5}\,\text{Pa}\,\text{s}$. The surface tension between the two phases was specified as $\gamma = 0.0728\,\text{N}\,\text{m}^{-1}$. The parameters of the dynamic contact angle model defined by (9) were fixed at $\theta_0 = 70°$, $\theta_A = 75°$, $\theta_R = 65°$ and $u_\theta = 1\,\text{m}\,\text{s}^{-1}$.

The changed parameter was the flow rate of the phase A (liquid), which we quantified via the Reynolds and Weber numbers, defined as

$$\text{Re} = \frac{\rho_A ||U_I|| H_I}{\mu_A}, \qquad \text{We} = \frac{\rho_A ||U_I||^2 H_I}{\gamma}, \tag{10}$$

where H_I is the above-introduced liquid inlet height and U_I is the uniform velocity prescribed at the liquid inlet. The coordinate system was chosen in such a way that $U_I = (u_I, 0, 0)$, $u_I > 0$, and thus $||U_I|| = u_I$. The gas phase, B, was assumed to be stationary in order to enable the validation of the simulations against the experimental data available in the literature.

The meshes used for the purpose of the present study were hexahedral and structured (see Fig. 4). The number of cells in the respective meshes was determined based on the mesh independence study.

[1] The selected plate inclination corresponds to the channel inclination of the Mellapak 250X commercial packing.

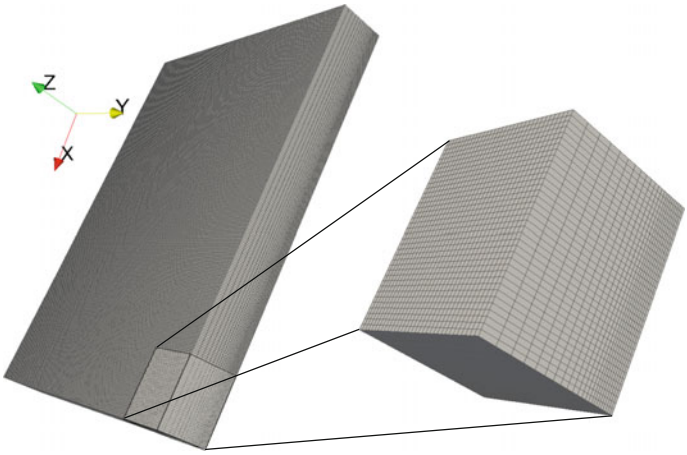


Fig. 4 Example of a structured hexahedral mesh used for the study of the flow properties on the smooth plate. The domain dimensions are $6 \times 5 \times 0.7$ cm (length $\times$ width $\times$ height)

In order to perform the above-mentioned mesh independence study, we defined the variables,

$$\delta_\alpha^i = \frac{\left\| \alpha^F(t_i) - \tilde{\alpha}^C(t_i) \right\|}{\texttt{nCells}_\alpha} \quad \text{and} \quad \overline{\delta}_\alpha = \frac{1}{\texttt{nTimes}} \sum_{(i)} \delta_\alpha^i \,, \tag{11}$$

where α^F is the solution for the phase volume fraction field, α, obtained on the finer of the compared meshes, $\tilde{\alpha}^C$ denotes the solution obtained on the coarser mesh mapped on the finer mesh, $\texttt{nCells}_\alpha$ is the number of cells in the finer mesh fulfilling the condition $\alpha^F \geq 0.5$ and nTimes is the number of the stored solution vectors. For the case of a mesh-independent solution, the variables δ_α^i, $i = 1, \ldots,$ nTimes, and hence also the variable $\overline{\delta}_\alpha$, should have the order of magnitude of the interpolation error introduced by the mapping of α^C on the finer mesh. An example of the time evolution of δ_α^i for different mesh sizes and the case of the transversal texture is shown in Fig. 5.

The evolution of $\overline{\delta}_\alpha$ with the mesh refinement for the case of transversal texture is depicted in Fig. 6. In an ideal case, the $\overline{\delta}_\alpha$ function should converge monotonically towards zero with the increasing number of cells in the mesh. Any jumps in the function values indicate a qualitative change in the solution obtained for two consecutive levels of mesh refinement. Furthermore, the time evolution of δ_α^i helps to indicate the time at which such a deviation of the two solutions occurs. Finally, the mesh sizes used in the present study were 1,104,000 cells for the smooth plate and the plate equipped with the transversal texture, 1,725,000 cells for the plate equipped with the longitudinal texture, and 2,136,000 cells for the case of the pyramidal texture.

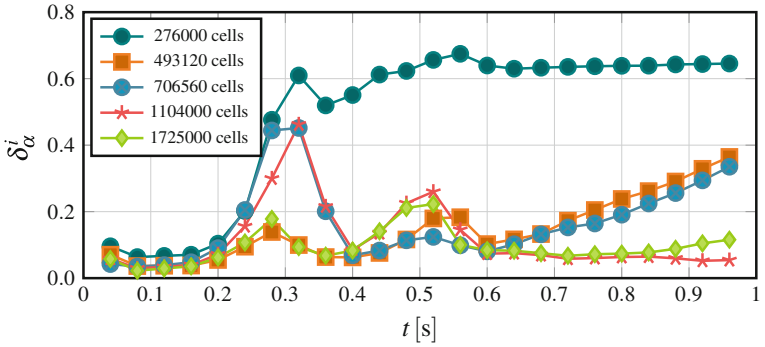


Fig. 5 Example of the time evolution of δ_α^i for different mesh sizes. A case of the transversal texture with $2l_T = 1$ mm and $h_T = 0.2$ mm is depicted. The numbers of cells in the plot legend correspond to the number of cells in the finer of the two compared meshes

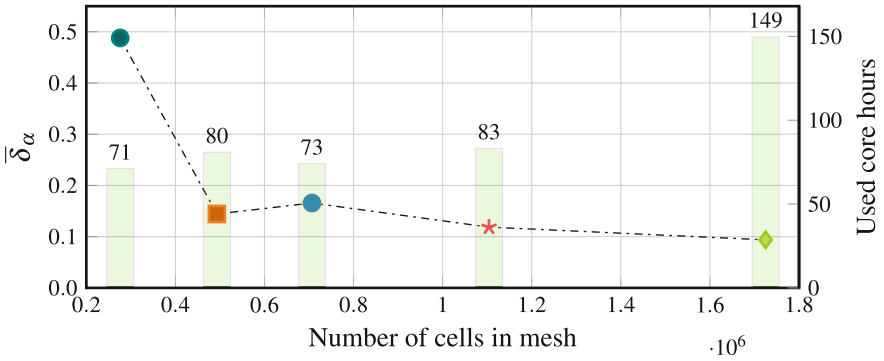


Fig. 6 Example of the evolution of $\overline{\delta}_\alpha$ with mesh refinement. The second axis is used to depict the number of core hours needed to perform the simulation in parallel on 4 cores of the Intel Xeon E5-4627v2@3.3GHz processor (In case of simulations with number of cells lower than 1 million, the communication between the cores was responsible for a significant portion of the simulation time. Thus, the first 4 simulations depicted in Fig. 6 used up roughly the same number of core hours). The picture shows the same case as in Fig. 5. Furthermore, the same notation is employed

3 Results and Discussion

At first, we validated the proposed model against the experimental data and simulation results available in the literature. The resulting comparison is depicted in Fig. 7. Based on the available data, it would seem, that the CFD simulations underestimate the specific wetted area. On the other hand, all the experiments are based on an optical technique; hence, it is necessary to add some tracer to the liquid. Furthermore, it is almost impossible to maintain the absolute purity of the water during the measurements. The presence of both the impurities and the tracer influence the physical properties of the liquid. For the specific case of water, both the impurities and the

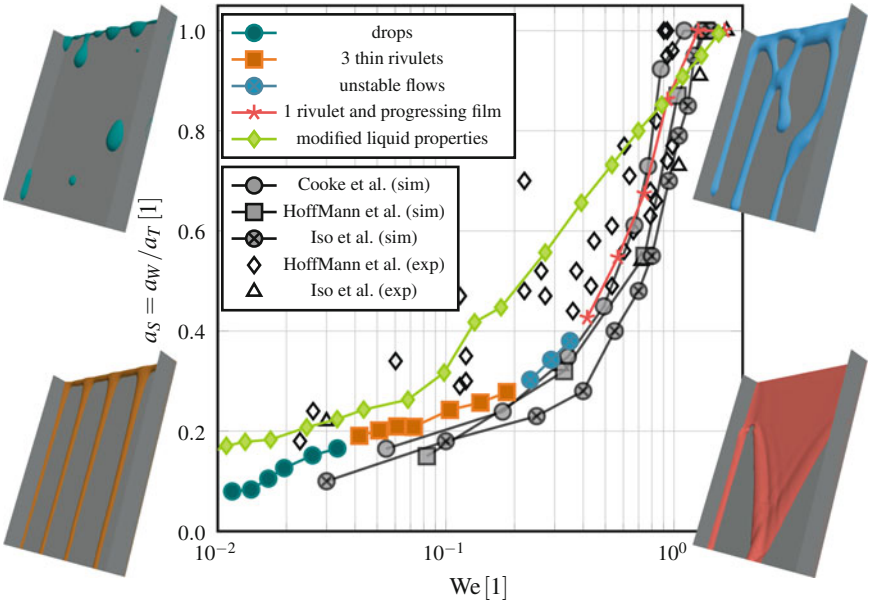


Fig. 7 The specific wetted area, a_S, in dependence on the Weber number. The data from literature are depicted in grayscale. The results of the model introduced in Section 2 are depicted with colors. Examples of each type of the identified flow are depicted in image borders and colored accordingly to the plot legend

tracer tend to lower its surface tension and contact angles. If we decrease the surface tension coefficient, γ, as well as the parameters of the dynamic contact angle model, θ_0, θ_A and θ_R, by a mere 10% and rerun the simulations, the obtained a_S are mostly overestimated (consult Fig. 7).

During the model validation, we tested different flow rates in the range approximately corresponding to We $\in (0.01, 1)$ or Re $\in (20, 220)$. For these flow rates, we identified, in total, four qualitatively different types of flow (see the borders of Fig. 7). For low flow rates (We < 0.03), the liquid flows down the smooth plate in the form of isolated droplets. With the increase in flow rate, those droplets merge and form thin, stable rivulets. For We ≥ 0.2, those rivulets lose the stability and the flow becomes chaotic. At last, with a further increase of the liquid flow rate beyond We $\doteq 0.3$, a region with a fully formed thin film emerges on the top of the plate. The film tends to break and form a stable rivulet further along the plate. However, the area occupied by the film smoothly increases with the increase in the intensity of the flow.

After validation of the model for the case of the smooth plate, we performed a series of numerical experiments with the different types of texture. The specific wetted area, a_S, for the four studied cases is depicted in Fig. 8. The texture density was fixed at $l_T = 2$ mm and the texture roughness at $h_T = 0.2$ mm.

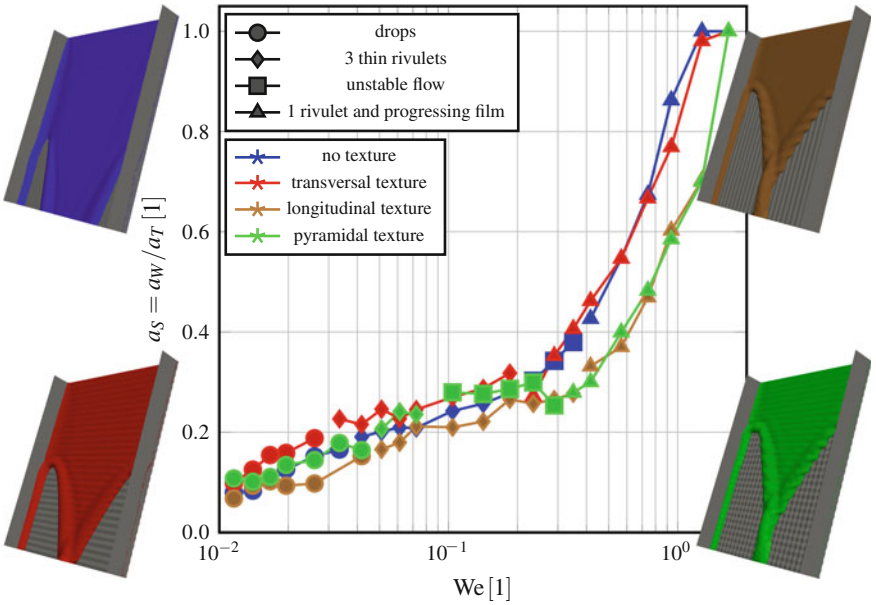


Fig. 8 The specific wetted area, a_S, in dependence on the Weber number. The results for different textures are distinguished by different colors. Furthermore, the different flow regimes are identified by the symbols used. Examples of the flow on each of the studied textures for $\mathrm{Re} = 185$ and $\mathrm{We} = 0.94$ are depicted in the image borders and colored according to the plot legend. Another example of the flow on different textures may be found in Fig. 2

From the comparison of a_S for the four tested types of texture, it is possible to conclude that the transversal texture increased the wetted area for lower and medium liquid flow rates, i.e., for $\mathrm{We} \in (0.01, 0.2)$. The used pyramidal texture seems to increase the wetted area for the medium liquid flow rates, roughly corresponding to $\mathrm{We} \in (0.08, 0.2)$. Furthermore, in this region, the pyramidal texture also highly increased the instability of the flow. The longitudinal texture did not seem to improve the liquid wetting properties for any of the tested liquid flow rates.

In general, the textures tried did not seem to improve the liquid wetting properties significantly. During the numerical experiments, it was observed that the liquid was slowed down by the texture, and thus lost its kinetic energy. This resulted in the pinning of the contact lines and stopping of the wetting process. On the other hand, in the literature [2, 12], there are data available on textures that caused a significant increase of the specific wetted area of the plate. Furthermore, a large number of the commercially available structured packings is equipped with some kind of a surface texture.

It is interesting that the industrial textures used in practice do not correspond to those documented in the literature. For example, the most widely employed type of texture used for commercial packings is close to the pyramidal texture presented in this work (see Fig. 1). Yet, besides the present study, all the authors concentrate on

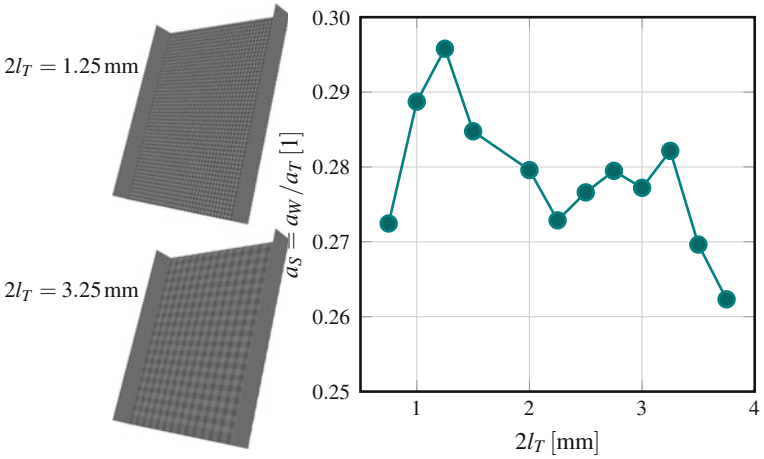


Fig. 9 The specific wetted area for the case of the pyramidal texture with different $2l_T$ varying from 0.75 to 3.75 mm. The Reynolds number was fixed at Re = 62, which corresponds to We = 0.1

either longitudinal or transversal types of texture. Use of the pyramid structures in industrial applications might be promoted by the fact that in the packing, it is not possible to estimate the angle of attack on the texture. Thus, on the scale of the full column, the pyramidal texture may perform better than the unidirectional textures.

A dependence of the specific wetted area, a_S, on the texture density, l_T, is depicted in Fig. 9. For the purpose of the study, we selected the case from the range of the liquid flow rates in which the default pyramidal texture was performing the best (We $\in$ (0.08, 0.2), consult Fig. 8). We fixed the liquid flow rate at We = 0.1 and tested the flow behavior for different texture densities. From the data, it seems that the best for the given case should be the textures with $2l_T$ roughly between 1 and 1.5 mm. Such textures best promote the instabilities in the flow and coincidently increase the amount of the liquid on the plate.

However, only a larger dataset would enable us to specify if the texture with $2l_T \doteq 1.25$ mm would perform best for a wider range of liquid flow rates. Furthermore, a similar study should be provided for the texture roughness, h_T.

4 Conclusion

Despite the ever-growing computing capacity of modern computers, multiphase modeling of free surface flows in complex geometries of industrial apparatuses is still beyond the possibilities of the most of the computers available to CFD engineers. Hence, modeling of such devices has to be split into different scales, with these tasks then being solved separately, whereby information is shared between them. In the

present paper, we provided a detailed micro-scale investigation of the gravity-driven flow of a liquid on four different types of textures. We identified four qualitatively different types of flow and provided estimates for the ranges of the Weber and Reynolds numbers for which these flows might occur. Finally, we studied effects of the four different types of textures on the flow behavior. Unfortunately, we were not able to confirm a significant improvement of the plate wetting properties associated with the presence of the texture. However, we plan to further evaluate the performance of the pyramidal texture in dependence on the parameters h_T and l_T. Furthermore, as water systems are not the ones most widely used in the industrial separation columns, it would be useful to complete the analysis by testing other types of liquids as well.

Acknowledgements This work was supported by the Centre of Excellence for nonlinear dynamic behaviour of advanced materials in engineering CZ.02.1.01/0.0/0.0/15_003/0000493 (Excellent Research Teams) in the framework of Operational Programme Research, Development and Education.

References

1. Brackbill J, Kothe D, Zemach C (1992) A continuum method for modeling surface tension. J Comp Phys 100:335–354
2. Cooke JJ, Gu S, Armstrong LM, Luo KH (2012) Gas-liquid flow on smooth and textured inclined planes. World Ac of Sc, Eng and Technol 68:1712–1719
3. Haroun Y, Raynal L, Alix P (2014) Prediction of effective area and liquid hold-up in structured packings by CFD. Chem Eng Res and Des 92:2247–2254
4. Hirt CW, Nichols BD (1981) Volume of fluid (VOF) method for thy dynamics of free boundaries. J Comp Phys 39:201–225
5. Hoffmann A, Ausner I, Repke JU, Wozny G (2005) Fluid dynamics in multiphase distillation processes in packed towers. Comp & Chem Eng 29(6):1433–1437
6. Hoffmann A, Ausner I, Repke JU, Wozny G (2006) Detailed investigation of multiphase (gas-liquid and gas-liquid-liquid) flow behaviour on inclined plates. Chem Eng Res and Des 84(A2):147–154
7. Kistler SF, Schweizer PM (eds) (1997) Liquid film coating., 1st edn. Chapman and Hall, London
8. Maiti R, Khanna R, Nigam KPD (2006) Hysteresis in trickle-bed reactors: A review. Ind Eng Chem Res 45:5185–5198
9. OpenCFD (2016a) OpenFOAM®: The Open Source CFD Toolbox. Programmer's Guide. OpenCFD Ltd, UK
10. OpenCFD (2016b) OpenFOAM®: The Open Source CFD Toolbox. User Guide. OpenCFD Ltd, UK
11. Raynal L, Royon-Lebeaud A (2007) A multi-scale approach for CFD calculations of gas-liquid flow within large size column equipped with structured packing. Chem Eng Sci 62(24):7196–7204
12. Sebastia-Saez D, Gu S, Ranganathan P (2013) 3D modeling of hydrodynamics and physical mass transfer characteristics of liquid flom flows in structured packing elements. Int J of Greenhouse Gas Cont 19:492–502
13. Towell GD, Rothfeld LB (1966) Hydrodynamics of rivulet flow. AIChE J 9:972–980
14. Vlasogiannis P, Karagiannis G, Argyropoulos P, Bontozoglou V (2002) Air-water two-phase flow and heat transfer in a plate heat exchanger. Int J Multiph Flow 25:757–772
15. Xu Y, Yuan J, Repke JU, Wozny G (2012) CFD study on liquid flow behavior on flat plate focusing on effect of flow rate. Eng Appl of Comp Fluid Mech 6(2):186–194

Enhanced Turbomachinery Capabilities for Foam-Extend: Development and Validation

Ilaria De Dominicis, Gregor Cvijetić, Mark Willetts and Hrvoje Jasak

Abstract Turbomachinery simulations represent one of the most challenging fields in Computational Fluid Dynamics (CFD). In recent years, the general CFD capabilities of foam-extend have been extended by introducing and maintaining additional features specifically needed for turbomachinery applications, with the aim of offering a high-quality CFD tool for the study of rotating machinery. This work presents the implementation and validation of new capabilities for turbomachinery with foam-extend, a community-driven fork of OpenFOAM®. The formulation of an energy equation more convenient for compressible turbomachinery applications has resulted in the rothalpy equation. Rothalpy is a physical quantity conserved over a blade row, stator or rotor, but not over a stage, both stator and rotor. It is fundamental to take into account that the value of rothalpy is not continuous across the rotor–stator interface, due to the change of rotational speed between zones. The rothalpy equation has been derived for both relative and absolute frames of reference, showing that additional terms appear in the absolute frame of reference. Moreover, additional functionality has been added to the rotor–stator interface boundary conditions' General Grid Interface (GGI), partial Overlap GGI and Mixing Plane Interface, in order to account for the rothalpy jump. The development of these new capabilities and their validation are shown, as well as industrial applications of compressible turbomachinery flows.

I. De Dominicis (✉)
General Electric Power, Methodologies for Tools, Newbold Road, Rugby CV21 2NH, UK
e-mail: ila.dedominicis@gmail.com

G. Cvijetić · H. Jasak
Department of Energy, Power Engineering and Environment,
Faculty of Mechanical Engineering and Naval Architecture,
University of Zagreb, Ivana Lučića 5, 10000 Zagreb, Croatia
e-mail: gregor.cvijetic@fsb.hr

H. Jasak
e-mail: hrvoje.jasak@fsb.hr

M. Willetts
General Electric Power, Tools & Methods Integration Group,
Newbold Road, Rugby CV21 2NH, UK
e-mail: mark.willetts@ge.com

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_11

Nomenclature

e	Energy
h_0	Total enthalpy
i	Rothalpy
ρ	Density
p	Static pressure
R	Radius
S_H	Source term of enthalpy
T	Temperature
t	Time
$\bar{\bar{\tau}}$	Viscous stress tensor
$\mathbf{U}$	Rotational velocity vector
$\mathbf{V}$	Absolute velocity vector
V_θ	Component of absolute velocity in the tangential direction
$\mathbf{W}$	Relative velocity vector
ω	Rotor angular velocity

1 Introduction

The role of turbomachinery studies is fundamental, nowadays, in a large number of industrial CFD applications. Rotating machinery presents particular challenges, such as complex geometries, the presence of an adverse pressure gradient and the relative motion of multiple rotors and stators, which require the modification of standard CFD tools with turbomachinery-specific capabilities. Jasak and Beaudoin [1] have described the implementation of turbo-specific features in the open-source numerical simulation software foam-extend, emphasizing the basic functionalities of turbo tools, the software layout in foam-extend and the numerical formulation of interfaces between stages.

The Open-Source platform foam-extend is a community-driven fork of OpenFOAM®, consisting of C++ libraries for Computational Continuum Mechanics, with extended CFD capabilities. The object-oriented architecture of foam-extend allows users to efficiently modify specific libraries and take advantage of the existing parts of the toolbox. The full source code of foam-extend is released under the GPL license and can be used at no cost.

The main challenges that turbomachinery simulations impose are linked to the need to study the relative motion of multiple rotors and stators. Depending on the transient or steady-state approach, the rotation can be handled in two ways: directly, by moving the mesh, or indirectly, by using a static mesh and modifying the equations to take into account the rotation. The former is only appropriate for transient simulations, whereas the latter is suitable for steady flow and may be formulated either for a single body in rotation, using the Single Rotating Frame of Reference

method (SRF), or for different rotating regions, using the Multiple Rotating Frame of Reference method (MRF). SFR and MRF can be implemented choosing either the absolute or relative velocity formulation. Recently, a new method for studying turbomachinery has also been developed and validated in foam-extend: the Harmonic Balance method for nonlinear, temporally periodic and incompressible flows [2].

The interface between stationary and rotating parts can be treated in several ways: for transient simulations with topological changes, the sliding mesh technique is commonly used, whereas for steady-state simulations, the methods available are the General Grid Interface (GGI), in which a weighted interpolation is performed to evaluate and transmit flow values across a pair of conformal or non-conformal coupled patches [3]; the partial Overlap GGI for cases in which some of the interface faces are not physically covered by their counterpart; and the Mixing Plane Interface, which consists of circumferential averaging of the solution at the rotor–stator interface [4].

The focus of this paper is the implementation of the rothalpy equation in foam-extend, as well as the implementation of the related boundary conditions and rotor–stator interfaces in order to properly handle this physical quantity. The validation is performed on steady simulations, in which the mesh is static and the rotation is handled by means of the MRF approach.

The paper is organized as follows: Sect. 2 contains a theoretical discussion of the mathematical model, Sect. 3 reports the validation cases and discussion, ending with the conclusion in Sect. 4.

2 Mathematical Model

The energy equation recommended for complex turbomachinery applications are the conservation of rothalpy equation. This is due to the rothalpy's property of being conserved over a blade row, stator or rotor, under the conditions investigated by Lyman [5]: isentropic flow, steadiness in the rotor frame, constant rotor angular velocity and no work done by the net viscous and body forces on the relative flow.

The rothalpy is a physical quantity, defined as [6]

$$i = h_0 - \omega R V_\theta. \tag{1}$$

The steady conservation equation for rothalpy is strictly dependent on the MRF formulation used. When the MRF is formulated in relative velocity, it is

$$\nabla\cdot(\rho i \mathbf{W}) = \nabla\cdot(k\nabla T + \bar{\bar{\tau}}\cdot\mathbf{V}) + S_H. \tag{2}$$

When the absolute velocity formulation is used for MRF, the steady conservation equation for rothalpy is

$$\nabla\cdot(\rho i \mathbf{W}) = -\nabla\cdot(\rho\omega R V_\theta \mathbf{W}) - \nabla\cdot(p\mathbf{U}) + \nabla\cdot(k\nabla T + \bar{\bar{\tau}}\cdot\mathbf{V}) + S_H. \tag{3}$$

Equations 2 and 3 differ because of the two terms $-\nabla\bullet(\rho\omega R V_\theta \mathbf{W})$ and $-\nabla\bullet(p\mathbf{U})$, which are only present in Eq. 3. These are, respectively, the convection of the quantity $\omega R V_\theta$ and the work done by the pressure forces. At the rotor–stator interfaces, the value of the rothalpy is not equal on both the rotor and stator sides, as the rotational velocity of the stator is zero. This requires modifications of existing rotor–stator interface methods: GGI, partial Overlap GGI and Mixing Plane Interface, in order to take into account the rothalpy jump at the rotor–stator interface. The value of the jump has been calculated by observing that on the stator, the rothalpy is equal to total enthalpy, being the angular velocity of rotor null, whereas on the rotor, both components of the rothalpy are non-null.

$$\begin{cases} i = h_0 & \text{stator} \\ i = h_0 - \omega R V_\theta & \text{rotor} \end{cases} \tag{4}$$

Therefore, the value of the rothalpy jump at the rotor–stator interface is $-\omega R V_\theta$. Accounting for the possibility of reverse flow, the jump has to be added to each face where the flux is going from the stator to the rotor and has to be subtracted from each face where the flux is going from the rotor to the stator.

3 Validation and Discussion

Validation of the new compressible solver specialized for turbomachinery applications has been carried out comparing the global turbomachinery parameters and the flow fields achieved with foam-extend and with a CFD commercial solver.

A 1.5-stage axial turbine has been chosen as the validation case. Steady simulations have been performed, accounting for only one rotor position. In order to shorten the CPU time, periodic boundary conditions have been adopted and only 1 blade passage has been simulated, with whole geometry having 36 blades in each rotor and stator. The mesh used is structured and consists of 405,600 hexahedral cells. The rotational speed is set to 3501 rpm and the inlet velocity is 60 m/s, with the kinematic viscosity being 1.8×10^{-5} m^2/s. The interaction between the rotor and stator has been resolved by taking advantage of the new implemented boundary conditions, which handle the rothalpy jump over the rotor–stator interfaces. At the rotor–stator interfaces, all of the variables are transported continuously, except for rothalpy.

In this work, two different methodologies of interface treatment have been considered: the partial Overlap GGI and Mixing Plane approaches.

3.1 Aachen Test Case: Partial Overlap GGI Approach

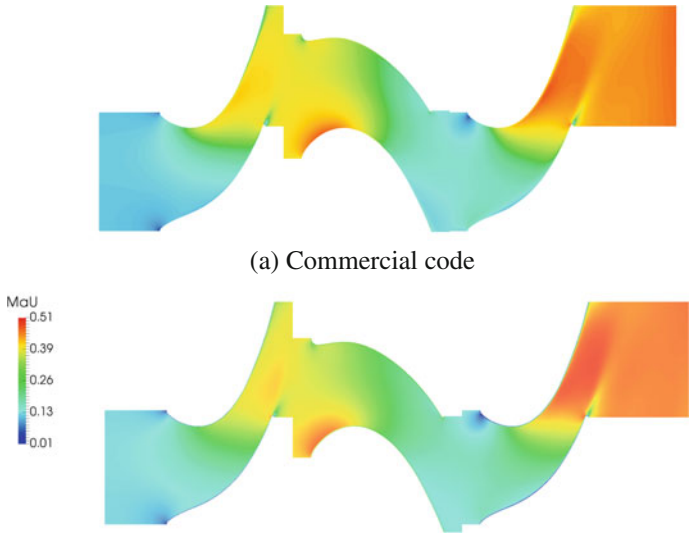
First, the partial Overlap GGI approach has been used at the rotor–stator interfaces. It interpolates the variables from one interface patch to another, allowing the use of a non-conformal mesh and the implementation of jump conditions over the interface before the interpolation is effected.

In Fig. 1, a comparison of the Mach number field obtained with a commercial code and the one obtained with foam-extend is shown. It can be noticed that the Mach number is continuous over the interface in both cases, and that the stator and rotor wakes are successfully resolved using the rothalpy jump boundary condition.

Figure 2 shows the static pressure fields comparison, whereas the temperature fields can be seen in Fig. 3. Since the temperature is calculated from the rothalpy, the effect of the rothalpy jump can be observed, as the temperature is continuous over the interface, rather than showing a jump.

3.2 Aachen Test Case: Mixing Plane Approach

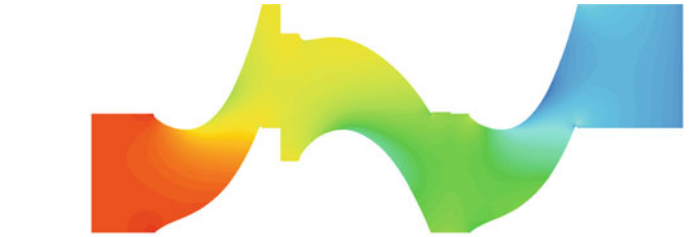
The Mixing Plane approach is usually employed when rotor–stator interaction is not of primary importance, but temporally averaged flow fields are considered. The flow fields calculated using the Mixing Plane Interface are comparable to those obtained by averaging the solution achieved in the transient simulation over time.



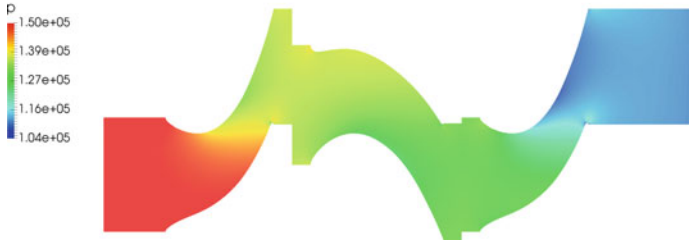
(a) Commercial code

(b) foam-extend

Fig. 1 Mach number field comparison



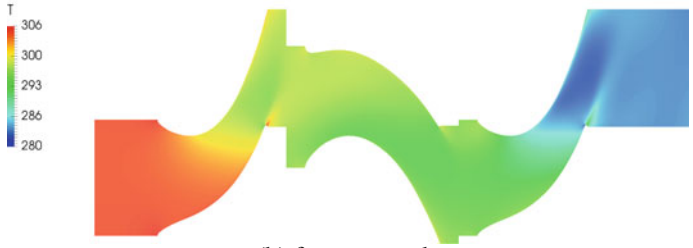
(a) Commercial code



(b) foam-extend

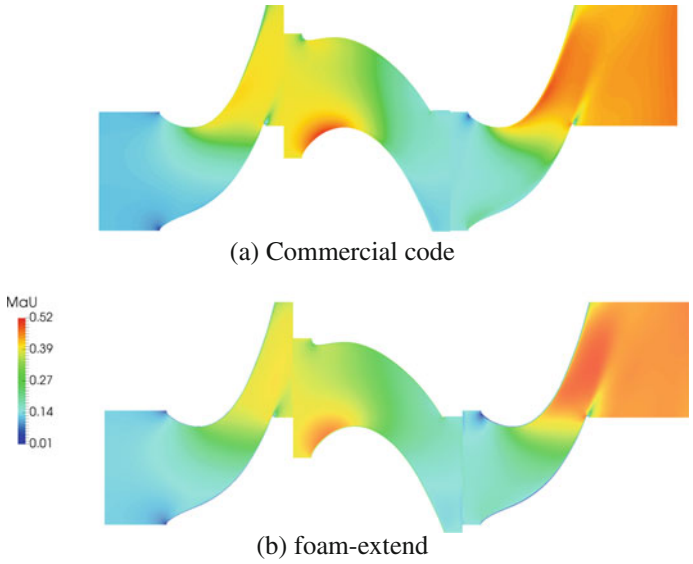
Fig. 2 Static pressure field comparison

(a) Commercial code



(b) foam-extend

Fig. 3 Temperature field comparison



(a) Commercial code

(b) foam-extend

Fig. 4 Mach number field comparison

Figure 4 presents a comparison of the Mach number field achieved with a commercial software and the one achieved with foam-extend by using the Mixing Plane approach at the rotor–stator interfaces.

Compared to the partial Overlap GGI approach, the interfaces are visible in this case due to the fact that the flow field has been averaged before being passed to the opposite side of the interface. By doing this, the variables are no longer continuous, but the result can still be considered physical, as it resembles the temporally averaged solution. It should be noted that, regardless of rothalpy jump, the mass conservation over the Mixing Plane Interface is preserved. Mass flow relative differences between the rotor and stator interfaces are on the order of 10^{-3}%.

Figures 5 and 6, respectively, present the static pressure and temperature fields.

3.3 Global Pump Parameters Comparison

The global pump parameters of isentropic efficiency and torque related to the first stage are presented in Table 1. Two different approaches of interface treatment, partial Overlap GGI and Mixing Plane Interface, and two CFD softwares, foam-extend and a commercial code, have been employed in this study. The last column presents the differences between the two codes, while the third row presents the differences between the two approaches. The differences comparing the approaches are below 1%, showing that, in the case of steady-state simulations, the Mixing Plane approach

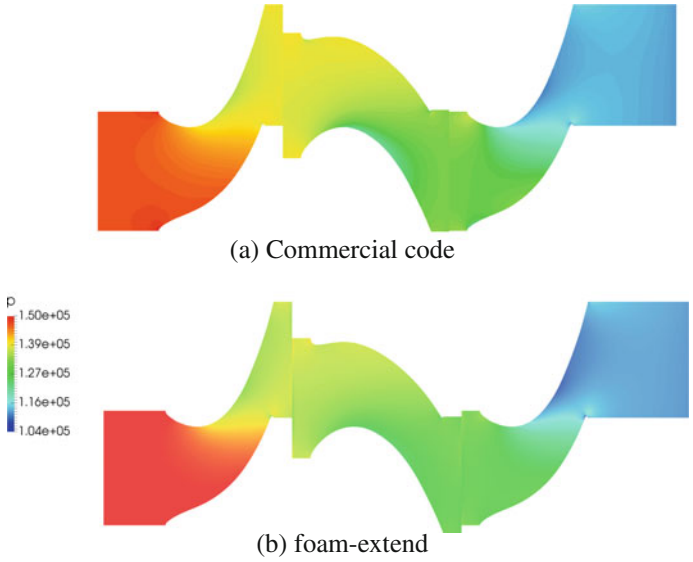


Fig. 5 Static pressure field comparison

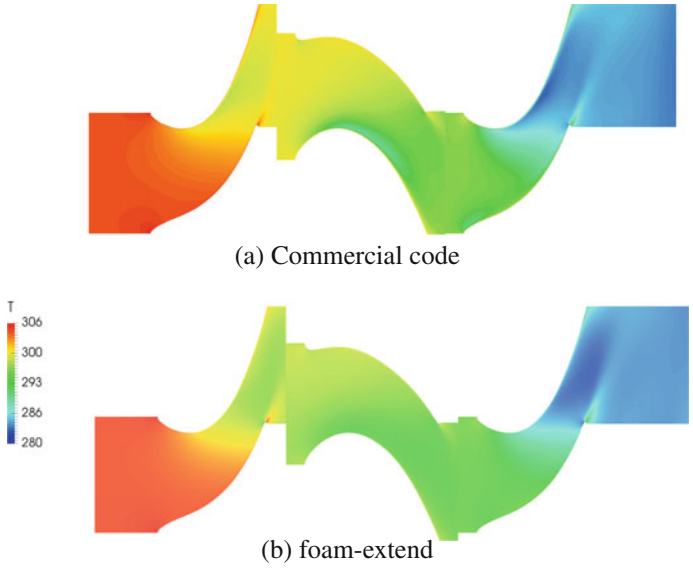


Fig. 6 Temperature field comparison

Table 1 Global pump parameters comparison

		Foam-extend	Commercial code	Difference (%)
Isentropic efficiency (–)	Partial Overlap GGI	0.8405	0.8553	1.78
	Mixing plane interface	0.8435	0.8499	0.76
	Difference (%)	0.37	0.63	
Torque (N/m)	partial Overlap GGI	5.2343	5.2428	0.16
	Mixing plane interface	5.2176	5.1986	0.36
	Difference (%)	0.32	0.84	

can be used as an accurate alternative to partial Overlap GGI if rotor–stator interaction is not needed. This is especially significant for cases with different pitches between the rotor and stator rows, as partial Overlap GGI is not suitable for this application. Comparing the results between foam-extend and the commercial code, the highest error is 1.78%, while other errors are below 1%.

Being an important aspect in turbomachinery simulations and design, efficiency differences show that foam-extend's rothalpy approach gives results comparable to a commercial CFD software.

4 Conclusion

In this work, the implementation and validation of additional turbomachinery functionality in foam-extend have been presented. The energy equation has been solved for the physical quantity rothalpy, which is convenient for turbomachinery applications, as it is conserved over a blade row, stator or rotor. The use of rothalpy has required the implementation of specialized boundary conditions for treating the rothalpy jump over the rotor–stator interface in order to obtain a continuous temperature field. The jump occurs due to rothalpy taking into account the rotational velocity, which is null for the stator. Regardless of the interface treatment chosen, partial Overlap GGI or Mixing Plane Interface, the rothalpy jump is performed in the interpolation part of interface communication. For validation, a 1.5-stage axial turbine has been used and the results achieved with foam-extend and with a commercial CFD software have been compared, showing good agreement. Foam-extend can, therefore, be efficiently used as a part of the development process with full turbomachinery capabilities.

Acknowledgements Marie Curie Initial Training Network (ITN) AeroTraNet2 of the European Community's Seventh Framework Programme (FP7).

Appendix

The MRF may be formulated using the relative or the absolute velocity formulation. In the first case, the conservation of rothalpy in a moving reference frame is derived directly from the conservation of energy formulated in terms of relative internal energy. It is

$$\frac{\partial(\rho i)}{\partial t} + \nabla\cdot(\rho i \mathbf{W}) = \frac{\partial p}{\partial t} + \nabla\cdot(k\nabla T + \bar{\bar{\tau}}\cdot\mathbf{V}) + S_H. \tag{5}$$

When the absolute formulation is used for the MRF, the conservation of rothalpy equations for a steadily moving frame is

$$\begin{aligned}\frac{\partial(\rho i)}{\partial t} + \nabla\cdot(\rho i \mathbf{W}) = \frac{\partial p}{\partial t} - \frac{\partial(\rho\omega R V_\theta)}{\partial t} \\ - \nabla\cdot(\rho\omega R V_\theta \mathbf{W}) - \nabla\cdot(p\mathbf{U}) + \nabla\cdot(k\nabla T + \bar{\bar{\tau}}\cdot\mathbf{V}) + S_H.\end{aligned} \tag{6}$$

This equation has been derived starting from the well-known conservation of energy equation in MRF for absolute velocity formulation [7]:

$$\frac{\partial(\rho e)}{\partial t} + \nabla\cdot(\rho e \mathbf{W}) = -\nabla\cdot(p\mathbf{V}) + \nabla\cdot(k\nabla T + \bar{\bar{\tau}}\cdot\mathbf{V}) + S_H. \tag{7}$$

And substituting the energy expression

$$e = h_0 - \frac{p}{\rho} = i + \omega R V_\theta - \frac{p}{\rho} \tag{8}$$

leads to

$$\begin{aligned}\frac{\partial(\rho i)}{\partial t} + \frac{\partial(\rho\omega R V_\theta)}{\partial t} - \frac{\partial p}{\partial t} + \nabla\cdot(\rho i \mathbf{W}) + \nabla\cdot(\rho\omega R V_\theta \mathbf{W}) - \nabla\cdot(p\mathbf{W}) = \\ - \nabla\cdot(p\mathbf{V}) + \nabla\cdot(k\nabla T + \bar{\bar{\tau}}\cdot\mathbf{V}) + S_H.\end{aligned} \tag{9}$$

Reorganizing the final terms, Eq. 6 is obtained.

References

1. Jasak, H. and Beaudoin, M., "OpenFOAM® turbo tools: From general purpose CFD to turbomachinery simulations," *ASME-JSME-KSME 2011 Joint Fluids Engineering Conference*, American Society of Mechanical Engineers, 2011, pp. 1801–1812.
2. Cvijetic, G., Jasak, H., and Vukcevic, V., "Finite Volume Implementation of the Harmonic Balance Method for Periodic Non–Linear Flows," *54th AIAA Aerospace Sciences Meeting*, 2016, p. 0070.
3. Beaudoin, M. and Jasak, H., "Development of a generalized grid interface for turbomachinery simulations with OpenFOAM®," *Open source CFD International conference*, Vol. 2, 2008.

4. Beaudoin, M., Nilsson, H., Page, M., Magnan, R., and Jasak, H., "Evaluation of an improved mixing plane interface for OpenFOAM®," *IOP Conference Series: Earth and Environmental Science*, Vol. 22, IOP Publishing, 2014, p. 022004.
5. Lyman, F. A., "On the conservation of rothalpy in turbomachines," *ASME 1992 International Gas Turbine and Aeroengine Congress and Exposition*, American Society of Mechanical Engineers, 1992, pp. V001T01A078–V001T01A078.
6. Ingram, G., *Basic concepts in turbomachinery*, Bookboon, 2009.
7. Poinsot, T. and Veynante, D., *Theoretical and numerical combustion*, RT Edwards, Inc., 2005.

Evaluation of Energy Maximising Control Systems for Wave Energy Converters Using OpenFOAM®

Josh Davidson, Christian Windt, Giuseppe Giorgi, Romain Genest and John V. Ringwood

Abstract Wave energy conversion is an active field of research, aiming to harness the vast amounts of energy present in ocean waves. An essential development trajectory towards an economically competitive wave energy converter (WEC) requires early device experimentation and refinement using numerical tools. OpenFOAM® is proving to be a useful numerical tool for WEC development, having been increasingly employed in recent years to simulate and analyse the performance of WECs. This chapter reviews the latest works employing OpenFOAM® in the field of wave energy conversion, and then presents the new application, of evaluating energy maximising control systems (EMCSs) for WECs, in an OpenFOAM® numerical wave tank (NWT). The advantages of using OpenFOAM® for this application are discussed, and implementation details for simulating a controlled WEC in an OpenFOAM® NWT are outlined. An illustrative example is given, and results are presented, highlighting the value of evaluating EMCSs for WECs in an OpenFOAM® NWT.

1 Introduction

Ocean waves represent an enormous renewable energy resource, however, economically harvesting this energy is a challenging problem. Developing a cost-effective WEC requires early optimisation and refinement of the device's design and operation using numerical tools, before considering the expense of physical prototype construction, deployment and experimentation. An EMCS can greatly improve the performance of a WEC, without any substantial increase in capital costs. Therefore, optimising and refining a WEC design and operation, requires evaluating EMCSs using numerical tools.

Numerically analysing and simulating the fluid–structure interaction (FSI) between a WEC and its environment, requires solving the Navier–Stokes equations, a problem computationally infeasible for historic computers. Therefore, the equations were linearised to obtain results using boundary element methods (BEMs).

J. Davidson (✉) · C. Windt · G. Giorgi · R. Genest · J. V. Ringwood
Centre for Ocean Energy Research, Maynooth University, Maynooth, Ireland
e-mail: josh.davidson@nuim.ie

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_12

The underlying hydrodynamics are based on linear potential theory, with assumptions of small wave and body motion amplitudes, inviscid fluid and an irrotational flow. However, these linearising assumptions are challenged by realistic WEC operation under *controlled* conditions. An EMCS effectively tunes the WEC dynamics to resonate with the incident waves, resulting in increased amounts of absorbed energy due to larger WEC motions. The large amplitude motions result in viscous drag, flow separation, vortex shedding and other nonlinear hydrodynamic effects.

Simulating a WEC under *controlled* conditions, therefore requires a realistic simulation environment, such as computational fluid dynamics (CFD). The discrepancy between the linear hydrodynamic model and CFD simulations, for WEC motions when the input wave frequencies are in the vicinity of the WEC resonant frequency, is shown by [48]. The output power estimations from the linear hydrodynamic model simulations, for frequencies around the WEC resonance, were shown to be considerably larger than the CFD estimations, due to the absence of viscous damping effects. The rigorous numerical treatment of the Navier–Stokes equations provided by CFD, enables a realistic, high- fidelity simulation environment for assessing WEC operation. However, the inclusion of nonlinear terms, neglected by linear hydrodynamic models, comes at the expense of massively larger computational requirements. Yet, the continuous improvement in performance and reduction in the cost of high-performance computers (HPCs), opens the way for CFD-simulated WEC experiments with reasonable computation times.

1.1 Outline of Chapter

This chapter focuses on the role OpenFOAM® can play in the evaluation of an EMCS for a WEC. OpenFOAM® provides open-source CFD solvers, whose application towards numerical experimentation on WEC devices has been rapidly growing in recent years. Section 2 reviews the usage of OpenFOAM® in wave energy research, showing a broad range of different WEC devices, simulated for a wide variety of research purposes. The new application of EMCS evaluation is then outlined in Sect. 3. A case study highlighting the importance of using a fully nonlinear simulation, such as OpenFOAM®, when evaluating the performance of an EMCS, is then presented in Sect. 4. The illustrative example in the case study, provides a comparison between the simulated motions and energy output of a WEC, under both *controlled* and *uncontrolled* conditions, calculated with a traditional linear hydrodynamic model and an OpenFOAM® simulation. An EMCS is used to drive a WEC into resonance with an incident wave field, and a divergence between the calculated linear model response and the OpenFOAM® simulation is observed.

2 OpenFOAM® in Wave Energy Applications

An extensive literature review of OpenFOAM®'s application in wave energy-related studies was presented in [6]. However, in the relatively short time since this review was composed (2015), numerous further studies have been published, demonstrating the growing usage of OpenFOAM® in wave energy research. This section builds upon [6], to provide an updated review detailing the broad range of WEC types and analysis purposes of OpenFOAM® in wave energy applications.

Oscillating water column (OWC)-type WECs operate by converting wave energy into pneumatic energy, whereby wave oscillations change the water levels inside of a chamber to force entrapped air through a turbine. OpenFOAM® is a useful tool for this type of WEC, able to model both the water and the air components of the OWC. Iturrioz et al. [24] validates an OpenFOAM® model for an OWC against physical experiments, showing excellent agreement among the free surface elevation (FSE), air pressure and velocity measurements. Likewise, [44–46] validate OpenFOAM® models of fixed OWCs against experimentally measured FSE and pressure data. To investigate the causes of damage to the Mutriku OWC plant, [29] use OpenFOAM® experiments to calculate flows, pressures and resulting loads at critical positions within the OWC.

An operating principle similar to that of an OWC is the *Blow Jet* WEC, which uses a horizontally oriented funnel to reproduce the hydraulic behaviour of a blowhole, turning relatively small waves into very strong air–water jets to drive an impulse turbine. Mendoza et al. [30] used OpenFOAM® to analyse different *Blow Jet* WEC configurations, validating results against measured pressure data. The *Bombora* WEC is comprised of submerged flexible membranes that use the force of incoming waves to drive air through a unidirectional air turbine. King et al. [26] uses an OpenFOAM® framework to model the FSI in the submerged flexible membranes of the *Bombora* WEC, coupling a simplified Finite Element model for the membrane and a thermodynamic model of the air ducting and turbine, with a CFD model for the water.

An oscillating wave surge converter (OWSC) is a flap-type WEC that rotates around a fixed axis in response to forcing from the incident waves. This type of WEC presents a particular meshing challenge in CFD, due to the large rotational displacements of the oscillating flap. A method for modelling this type of WEC in OpenFOAM® is presented in [42], along with a comparison of simulation results against experiments. The OpenFOAM® model developed in [42] is then used in [41] to optimise the power take-off (PTO) damping torque for a generic flap-type OWSC. Loh et al. [28] model a specific OWSC device, the *WaveRoller*, at a 1:24 scale under operational wave conditions to validate the numerical data with experiments. Ferrer et al. [16] model the *Oyster*, OWSC device, in extreme sea states to investigate slamming events, using both compressible and incompressible solvers, and compare the results against experiments. Akimoto et al. [1] proposes a new concept of rotational WEC, for capturing the orbital fluid particle motion of a wave. The preliminary CFD analysis demonstrates the rotating WEC and the wave flow field can keep the suitable position for torque generation in all the phases of orbital motion. Similarly, [13] implements

a 2D OpenFOAM® model for a quantitative investigation of the conversion performance of the *Seaspoon* WEC, which uses the same rotational operating principles.

Point absorber WECs are wave-activated bodies that are physically small compared to the typical wavelengths. OpenFOAM® was employed in [27, 31] to examine two-body self-reacting point absorber-type WECs, and in [35] to study the 'damper plate' component of self-reacting point absorbers. Devolder et al. [11] validates a heaving point absorber against free decay, and regular wave, experiments in a wave flume. Palm et al. [33] analyses a moored point absorber, by coupling a solver for the mooring system dynamics with OpenFOAM®, presenting the formulation for the coupled mooring analysis and validation results against physical experiments. Rafiee and Fiévez [34] uses OpenFOAM®, as well as traditional linear hydrodynamic models, to simulate the performance of the *CETO* point absorber WEC, under moderate and extreme wave conditions. The results in [34] were compared against physical experiments, showing a good agreement with the OpenFOAM® simulations but not by the linear models. Eskilsson et al. [14] investigates simulating the wave-induced motions of point absorber-type WECs, comparing the results of approximate but computationally efficient hydrodynamic models against the more complete but time-consuming OpenFOAM® simulations. Similarly, [17] investigate the difference among the performances of various linear and nonlinear hydrodynamic models compared with OpenFOAM® results, for the case of a heaving point absorber-type WEC.

The Ph.D. theses [4, 38], and the resulting papers, focus on the OpenFOAM® modelling of WECs. In [38], the use of OpenFOAM® to simulate WEC and mooring performance under survival sea conditions is investigated. The thesis presents several case studies; a fixed truncated cylinder, a moored buoy [36], the Wavestar [37] and Seabased WEC prototypes, including validation against physical wave tank experiments. Chen [4] implements wave generation and absorption by modifying the *interDyMFOAM* solver, and validates wave propagation and impact cases. The modified solver is then used to simulate and analyse the wave-induced roll motion of a rectangular barge and the hydrodynamic performance of an OWSC [5].

OpenFOAM® has been used for system identification of WEC models. The general concept of identifying mathematical models describing the dynamical behaviour of WECs from recorded data, using OpenFOAM® simulations as examples, is given in [40]. The types of identification tests available in an OpenFOAM® NWT are investigated in [10], and are used in [22] to identify the parameters of nonlinear hydrodynamic models. Giorgi and Ringwood [20] investigates the identification of hydrodynamic drag coefficients from OpenFOAM® experiments, the drag coefficients for the CETO WEC are identified by [34] using prescribed motion tests, and [2] determine nonlinear damping coefficients for a flap-type OSWC using free decay tests. Davidson et al. [7] uses system identification techniques to adapt the parameters of a linear control model online from measured responses of the WEC behaviour, so as to ensure the best linear model representation of the nonlinear conditions in the OpenFOAM® simulation.

Devolder et al. [12] review the *interDyMFoam* solver for the application of simulating a heaving buoy, outlining the importance of the fluid and body solver coupling

for wave energy applications and describe some pitfalls in the implemented methodology. Windt et al. [47] outlines an assessment methodology for the different numerical wavemakers available in an OpenFOAM® NWT for wave energy experiments, showcasing evaluation tests and metrics for their wave generation and absorption capabilities.

3 Evaluating Energy Maximisation Control Systems

By increasing the energy capture of a WEC, across changing sea states, EMCSs can improve the economic viability of the WEC. In addition to maximising energy output, EMCSs can also enforce constraints on the WECs operation. Maximum displacements and PTO forces can be constrained below desired values, so as to decrease device damage and fatigue and ensure efficient PTO sizing. A review of EMCSs for wave energy conversion is given in [39].

Evaluating the performance of an EMCS classically relied on linear model simulations. However, the increased amplitude of the WECs dynamics under controlled conditions challenges the validity of the linearising assumptions such models are built upon. Consistent with the observations in [48], the results in [9] show that increasing the amplitude of the WECs operation away from its zero-amplitude equilibrium state, leads to a divergence between the linear hydrodynamic model and CFD simulations. Specifically, the levels of hydrodynamic damping experienced by a WEC are seen to increase as the amplitude of operation increases. Therefore, evaluating an EMCS with a linear model will likely result in predictions of unrealistically large WEC motions and energy capture, due to an underestimation of the hydrodynamic damping on the WEC. CFD, on the other hand, has a greater range of validity when simulating large amplitude WEC motions. The treatment of nonlinear effects, such as viscosity or a time-varying wetted body surface area, enables CFD to provide a higher fidelity simulation, compared to a linear hydrodynamic model, at these operational amplitudes.

A strong advantage in choosing OpenFOAM® for the CFD simulation platform is the open-source nature of the software. The cost of commercial licenses can be prohibitive for university-based researchers, and WEC developers in small companies, with limited funds, which could be better spent purchasing HPC hardware or computing time. The open-source nature of OpenFOAM® often results in useful toolboxes being freely shared, a prime example being the wave generation and absorption toolboxes: waves2FOAM [25] and IHFOAM [23]. Of the papers reviewed in Sect. 2, waves2FOAM is used by [1, 10, 13, 19, 27, 30, 31, 34, 40, 43] and IHFOAM by [12, 24].

The complete access to the source code, provided by OpenFOAM®, allows modifications to be made. For example, mooring forces are applied to a WEC in [33] by modifying the *restraints* function in the *sixDoFRigidBodyMotion* solver, following the procedure outlined in [32]. The same function is modified in [10] to apply generic PTO forces to a WEC, and then is coupled with MATLAB in [7] to calculate opti-

mal control of the PTO force, as well as online system identification for the control model. Giorgi and Ringwood [19] implements latching control for a heaving sphere in regular waves where the WEC is 'latched' stationary during certain instants of the wave cycle, and then released at a later time when the phase of the incident wave is more favourable for increased energy capture. To implement the latching control, the source code was modified as detailed in [18].

4 Illustrative Example

An illustrative example is given, demonstrating the influence of the chosen simulation environment on the evaluation of an EMCS. Consider the WEC shown in Fig. 1, comprising a spherical buoy that acts as a point absorber. The WEC reacts against the inertia of the seafloor (or stationary damper plate) to extract power through a PTO force, F_{PTO}. Simulation of the WEC operation in an irregular sea state is performed by both an OpenFOAM® NWT and a linear hydrodynamic model, to compare the calculated wave-induced heave motion, $x(t)$, and energy capture

$$E(t) = \int_0^T F_{PTO}(t)\dot{x}(t)dt. \tag{1}$$

An *uncontrolled* case shall be used as a reference, in which the PTO acts as a simple linear damper, applying a purely resistive force proportional and opposite to the WEC velocity

$$F_{PTO}(t) = -d\dot{x}(t), \tag{2}$$

where d is the PTO damping parameter.

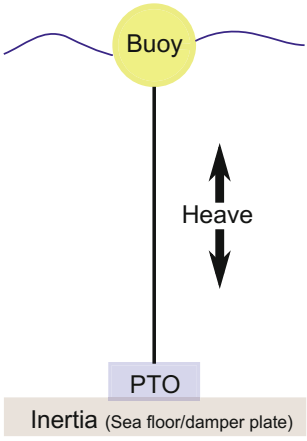


Fig. 1 WEC device considered in the illustrative example

The EMCS to be evaluated is PI control, which also applies a resistive PTO force proportional and opposite to the WEC velocity to absorb power, however, an additional reactive PTO force, proportional to the WECs displacement, is applied

$$F_{PTO}(t) = -d\dot{x}(t) - cx(t), \tag{3}$$

where c is the PTO spring parameter. PI control uses the reactive force to drive the WEC into resonance with the input waves, leading to increased WEC motions and energy capture. The value of c required to align the resonant period of the WEC, T_{WEC}, with the peak period of the input wave spectrum, T_p, is estimated here using linear oscillation theory, [15]

$$c = \frac{kT_{WEC}^2}{T_p^2} - k, \tag{4}$$

where k is the hydrodynamic restoring force coefficient.

The PTO damping parameter, d, is chosen as equal to the value of the WECs hydrodynamic radiation damping parameter at T_p, representing impedance matching at the peak wave period T_p [15].

4.1 Implementation

The illustrative example evaluates an EMCS, using both OpenFOAM® NWT and classical linear hydrodynamic model simulations. Here, implementation details, for the OpenFOAM® NWT, linear hydrodynamic model and EMCS, are given.

4.1.1 OpenFOAM® NWT

The implementation of the OpenFOAM® NWT is presented in [6]. The present example considers a WEC whose buoy has a radius of 0.1m, that floats 50% submerged at equilibrium, in the middle of a 100 m^2 square tank, with a 3 m water depth.

A cross-sectional view of the NWT mesh is depicted in Fig. 2a. Wave generation and absorption is implemented using the *waves2FOAM* toolbox and the wave creation and absorption zones are also depicted in Fig. 2a. In Fig. 2b, the dynamic pressure fields are seen to be generated in the wave creation zone, propagate through the central zone, interact with the WEC, and then be absorbed in the leeward side absorption zone. A unidirectional input wave spectrum, with a peak period of 1 s, is generated. The input waves are initially simulated without the WEC in the NWT, to allow the free surface elevation (FSE) to be measured at the centre of the tank. The FSE measurement is then used by the linear hydrodynamic model so that both simulations have the same input waves.

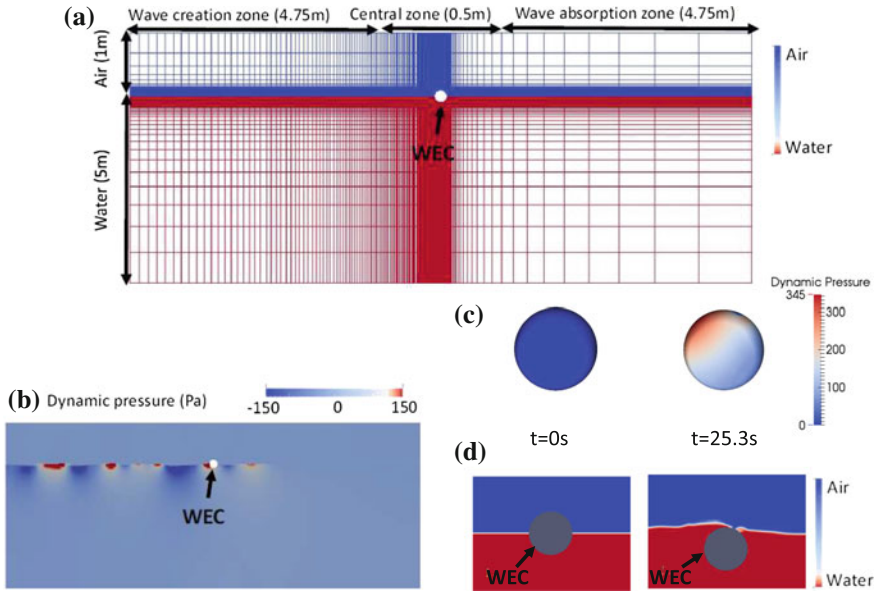


Fig. 2 Cross-sectional view of **a** the mesh and fluid volume fractions (water = red, air = blue) at time = 0 s, **b** the dynamic pressure at time = 25.3 s, **c** the dynamic pressure on the WEC at time = 0 s and 25.3 s, and **d** the fluid volume fractions around the WEC at time = 0 s and 25.3 s

4.1.2 Linear Model

The linear model, uses a fourth-order Runge–Kutta scheme to solve Cumin's equation, as described in [15], with the hydrodynamic parameters obtained from the open-source linear potential theory BEM software Nemoh [3].

4.1.3 Energy Maximising Controller

The PI controller is relatively easy to implement in OpenFOAM®, as it does not require any modifications to the source code. The *linearSpring* or *linearDamper* functions inside the *restraints* function of the *sixDoFRigidBodySolver* can be used directly. The functions require a *stiffness* and a *damping* value, which represent the PTO spring and damping parameters, c and d, in Eq. 3, respectively.

To determine the value for the PTO spring parameter, c, Eq. 4 can be used, once the values of T_{WEC} and k are known. To identify T_{WEC}, a free decay experiment is performed, Fig. 3a, and its spectral content is obtained, Fig. 3b, following the system identification techniques described in [10]. The peak of the spectrum in Fig. 3b, indicates a WEC resonant period of 0.61 s. To identify, k, the methods in [9] can be followed, using measurements of the hydrostatic force from the free decay experiment, Fig. 3c, to obtain the hydrostatic force versus displacement graph

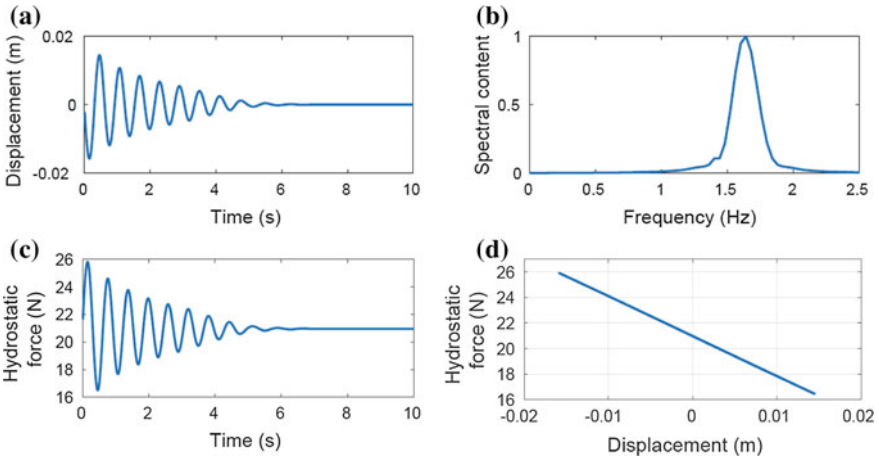


Fig. 3 **a** Simulated heave free decay test for WEC, **b** the spectral content of the signal indicating a resonant heave period of 0.61 s, **c** the hydrodstatic force versus displacement data used to identify the linear restoring force coefficient, and **d** system identification for the linear model's hydrostatic restoring force coefficient (method detailed in [9])

in Fig. 3d. The slope of the graph at $x = 0$ m, gives a linearised restoring force coefficient around the WEC equilibrium. A k value of 314 N/m can be identified from the results in Fig. 3d. Therefore, a PTO spring parameter, c, with a value of -197 N/m is obtained from Eq. 4. The PTO damping parameter, d, is set as equal to the linear hydrodynamic radiation damping at T_p, with a value of 6.22 Ns/m calculated using Nemoh.

4.2 Results

The generated input wave series is shown in Fig. 4a, the WEC heave motion for the uncontrolled and the PI control simulations are shown in Fig. 4b, c, respectively. The resulting heave motion for WECs using PI control can be seen to be considerably larger than for the uncontrolled cases. The absorbed energy is plotted in Fig. 4d, showing the effect of the reactive power applied by the PI controller, when during certain periods of time, the absorbed energy decreases, flowing back from the PTO to the WEC. However, over time, the PI-controlled WECs are seen to absorb considerably more energy than the uncontrolled WECs, highlighting the benefit of using control.

The results also show that the linear model and OpenFOAM® simulations agree well with each other in the uncontrolled case. However, in the controlled case, the linear model significantly overpredicts the WEC motion and absorbed energy compared to the higher fidelity OpenFOAM® simulation. At these larger amplitudes, nonlinear

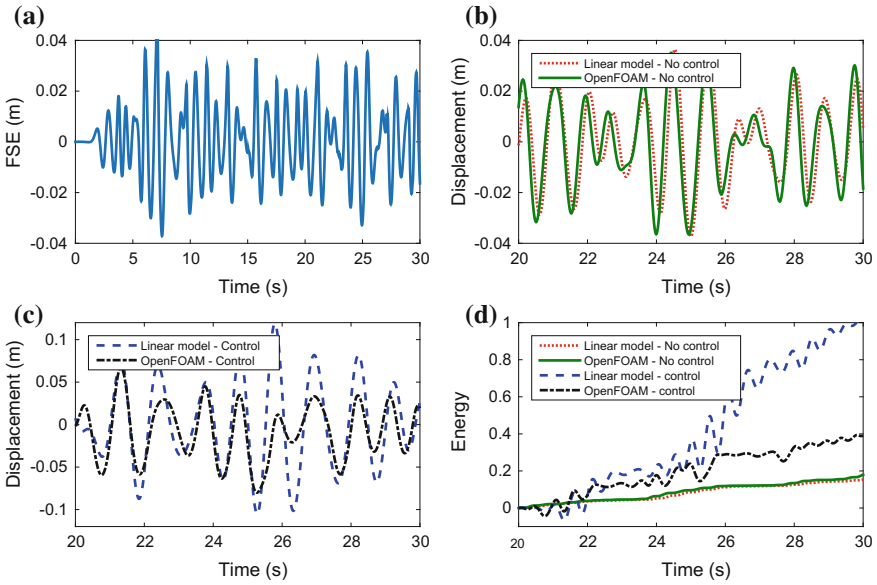


Fig. 4 **a** The measured FSE; **b** the heave displacement for the case of a passive damping PTO; **c** the heave displacement for the case of PI-controlled PTO; **d** normalised energy absorbed by the WEC

hydrodynamic effects begin to influence the device motion, and the predictions made by the linear model and the OpenFOAM® simulations diverge.

The operational space, in the displacement–velocity plane, spanned by the WEC motion is pictured in Fig. 5. The maximum WEC displacements and velocities from the four simulations in Fig. 4b, c are plotted. The operational space for the linear model and OpenFOAM® simulations of the uncontrolled WEC are very similar, and are much smaller than for the controlled WEC simulations. The linear model is seen to perform well compared to the more realistic OpenFOAM® simulation in the low amplitude operational space of the uncontrolled WEC. However, for the controlled WEC, the extended amplitude of the operational space diminishes the validity of the linear model, as nonlinear effects become relevant. Figure 5 shows that the operational space of the WEC motion in the OpenFOAM® simulation is much less than that predicted by the linear model simulation, likely due to the neglect of viscous drag effects by the linear model. The background of Fig. 5 displays a contour plot of the power absorbed by the PTO at each point in the operational space. The overprediction of absorbed energy made by the linear model for the controlled WEC, Fig. 4d, results from the WECs trajectory unrealistically spanning regions of large power absorption.

The amplitude of the relative displacement and relative velocity, between the WEC and the water, has a large effect on the presence of nonlinear hydrodynamic effects. For example, if the relative displacement between the WEC and the FSE

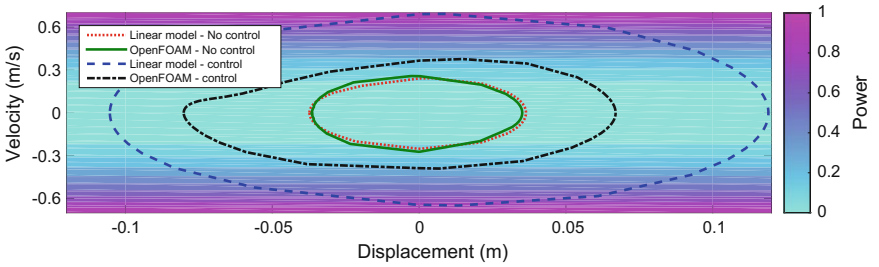


Fig. 5 The operational space in the displacement–velocity plane spanned by the WECs trajectory (lines), and the power absorbed by the PTO at each point in the operational space (contour)

exceeds the WEC radius, then the WEC will either be fully submerged or airborne. An example of that occurs at $t = 25.3$ s of the *controlled* OpenFOAM® simulation (shown in the snapshot of the WEC and the fluid in Fig. 2d). For a WEC geometry with a non-uniform horizontal cross section, such as the sphere, increasing the relative displacement amplitude increases the nonlinearity of the hydrodynamic restoring and Froude–Krylov forces, as shown in [8, 21], respectively. Viscous damping forces are dependent on the relative motion between the WEC and water, whereby viscous drag is often modelled as proportional to the square of the relative velocity.

The relative displacement between the WEC and the FSE is plotted in Fig. 6a, b, and the operational space, in the relative displacement–velocity plane, in Fig. 6c. The increase of different nonlinear hydrodynamic effects, for increasing amplitudes, are also indicated in Fig. 6c, and are seen to be more prevalent for a controlled WEC. Therefore, a realistic simulation environment, capable of modelling these nonlinear hydrodynamic effects, should be used when analysing the wave-induced motions of a WEC under *controlled* conditions.

While the illustrative example here utilised CFD, to capture the relevant nonlinear hydrodynamic effects evoked by the resulting large amplitude motions of a controlled WEC, other nonlinear hydrodynamic modelling techniques may also give improved results compared to the classical linear models, but with less computational requirements than CFD. A hierarchical approach to WEC hydrodynamic modelling is detailed in [14], examining the trade-off between model fidelity and computational requirements. Similarly, a comparison of different nonlinear hydrodynamic modelling techniques against the performance of an OpenFOAM® simulation is given in [17], for both an uncontrolled and a controlled WEC. Like the present illustrative example, the results in [17] also display a similar increase in operational space spanned by the uncontrolled and controlled WECs, and highlight the need for a high-fidelity nonlinear simulation environment for evaluating a controlled WEC.

The illustrative example shown, herein demonstrates the discrepancy between classical linear models and CFD when simulating a controlled WEC. To ensure confidence in the accuracy of the CFD results, the simulation should be validated against experimental data. Validating against a full-scale WEC in the open ocean is problematic, therefore a more common approach is to validate against a scaled-down

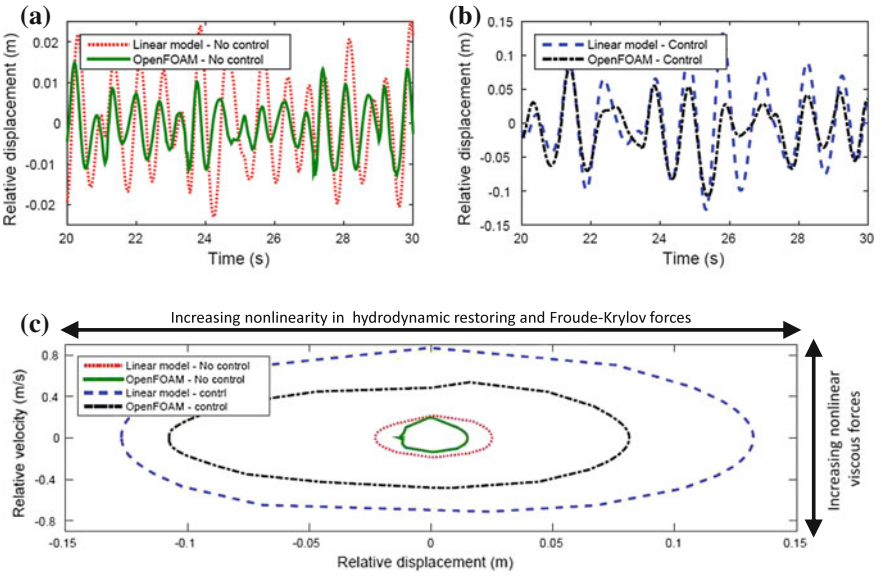


Fig. 6 The relative displacement between the WEC and FSE for **a** the uncontrolled WECs, and **b** the controlled WECs; **c** the relative displacement–relative velocity operational space spanned by the WEC trajectory

version of the WEC in an experimental wave tank facility, and then extrapolate that the validation holds true for full-scale conditions. The results from the illustrative example suggest that a CFD simulation validated under uncontrolled conditions will not extrapolate well to a simulation involving a controlled WEC, due the prevalence of nonlinear effects for the controlled WEC operation absent in the uncontrolled case.

5 Conclusion

Evaluating EMCSs for WECs requires an environment of realistic numerical simulation, capable of representing nonlinear hydrodynamic conditions. To maximise the absorbed energy, an EMCS will drive the WEC motion into resonance with an incident wave field, and the resulting FSI conditions challenge the validity of linear models. The example results shown in this chapter revealed that the energy capture evaluated by a linear model was more than double the energy predicted by the CFD simulation for a PI-controlled WEC. The increased amplitudes of the WEC displacement and velocity, and the relative WEC-water displacement and velocity, for a *controlled* WEC extend the operational space of the WEC dynamics far from the region where linear hydrodynamic assumptions are valid. The nonlinear FSI simula-

tions of CFD, on the other hand, are shown to more realistically handle the resonant conditions experienced when evaluating an EMCS for a WEC. OpenFOAM® is shown to be a useful simulation tool for the evaluation of an EMCS for a WEC.

Acknowledgements This chapter is based upon work supported by Science Foundation Ireland under Grant No. 13/IA/1886.

References

1. Akimoto, H., Kim, Y., Tanaka, K.: Configuration of the single-bucket wave turbine for the direct utilization of orbital fluid motion. In: Grand Renewable Energy (2014)
2. Asmuth, H., Schmitt, P., Elsaesser, B., Henry, A.: Determination of non-linear damping coefficients of bottom-hinged oscillating wave surge converters using numerical free decay tests. In: Proceedings of the 1st International Conference on Renewable Energies Offshore, Lisbon, Portugal, pp. 24–26 (2014)
3. Babarit, A., Delhommeau, G.: Theoretical and numerical aspects of the open source BEM solver NEMOH. In: 11th European Wave and Tidal Energy Conference (EWTEC2015) (2015)
4. Chen, L.: Modelling of marine renewable energy. Ph.D. thesis, University of Bath (2015)
5. Chen, L., Zang, J., Hillis, A.J., Plummer, A.R., et al.: Hydrodynamic performance of a flap-type wave energy converter in viscous flow. In: The Twenty-fifth International Offshore and Polar Engineering Conference. International Society of Offshore and Polar Engineers (2015)
6. Davidson, J., Cathelain, M., Guillemet, L., Le Huec, T., Ringwood, J.: Implementation of an OpenFOAM® numerical wave tank for wave energy experiments. In: Proceedings of the 11th European Wave and Tidal Energy Conference (EWTEC 2015), Nantes (2015)
7. Davidson, J., Genest, R., Ringwood, J.V.: Adaptive control of a wave energy converter simulated in a numerical wave tank. In: Proceedings of the 12th European Wave and Tidal Energy Conference (EWTEC 2017), Cork (2017)
8. Davidson, J., Giorgi, S., Ringwood, J.: Numerical wave tank identification of nonlinear discrete-time hydrodynamic models. In: 1st Int. Conf. on Renewable Energies Offshore (Renew 2014), Lisbon (2014)
9. Davidson, J., Giorgi, S., Ringwood, J.V.: Linear parametric models for ocean wave energy converters identified from numerical wave tank experiments. Ocean Engineering **103** (2015)
10. Davidson, J., Giorgi, S., Ringwood, J.V.: Identification of wave energy device models from numerical wave tank datapart 1: Numerical wave tank identification tests. IEEE Transaction on Sustainable Energy (2016)
11. Devolder, B., Rauwoens, O., Troch, P.: Numerical simulation of a single floating point absorber wave energy converter using OpenFOAM®. In: Proceedings of the 2nd International Conference on Renewable Energies Offshore (2016)
12. Devolder, B., Schmitt, P., Rauwoens, P., Elsaesser, B., Troch, P.: A review of the implicit motion solver algorithm in OpenFOAM® to simulate a heaving buoy. In: NUTTS conference 2015: 18th Numerical Towing Tank Symposium, pp. 1–6 (2015)
13. Di Fresco, L., Traverso, A., Barberis, S., Guglielmino, E., Garrone, M.: Off-shore wave energy harvesting: A wec-microturbine system: Harvesting and storing energy for off-shore applications. In: OCEANS 2015-Genova, pp. 1–6. IEEE (2015)
14. Eskilsson, C., Palm, J., Engsig-Karup, A., Bosi, U., Ricchiuto, M.: Wave induced motions of point-absorbers: a hierarchical investigation of hydrodynamic models. In: 11th European Wave and Tidal Energy Conference (EWTEC). Nantes, France (2015)
15. Falnes, J.: Ocean Waves and Oscillating Systems : linear interactions including wave-energy extraction. Cambridge University Press (2002)

16. Ferrer, P.M., Causon, D.M., Qian, L., Mingham, C.G., Ma, Z.H.: Numerical simulation of wave slamming on a flap type oscillating wave energy device. In: Proceedings of the Twenty-sixth (2016) International Ocean and Polar Engineering Conference (2016)
17. Giorgi, G., Retes, M., Ringwood, J.: Nonlinear hydrodynamic models for heaving buoy wave energy converters. In: 3rd Asian Wave and Tidal Energy Conference (2016)
18. Giorgi, G., Ringwood, J.: NWT Latching Control User Manual. Available at: http://www.eeng.nuim.ie/coer/doc/NWTLatchingControlUserManual.pdf
19. Giorgi, G., Ringwood, J.V.: Implementation of latching control in a numerical wave tank with regular waves. Journal of Ocean Engineering and Marine Energy **2**(2), 211–226 (2016)
20. Giorgi, G., Ringwood, J.V.: Consistency of viscous drag identification tests for wave energy applications. In: Proceedings of the 12th European Wave and Tidal Energy Conference (EWTEC 2017), Cork (2017)
21. Giorgi, S., Davidson, J., Ringwood, J.V.: Identification of nonlinear excitation force kernals using numerical wave tank experiments. In: EWTEC (2015)
22. Giorgi, S., Davidson, J., Ringwood, J.V.: Identification of wave energy device models from numerical wave tank data - part 2: Data-based model determination. IEEE Transaction on Sustainable Energy (2016)
23. Higuera, P., Lara, J.L., Losada, I.J.: Simulating coastal engineering processes with OpenFOAM®. Coastal Engineering **71**, 119–134 (2013)
24. Iturrioz, A., Guanche, R., Lara, J., Vidal, C., Losada, I.: Validation of OpenFOAM® for oscillating water column three-dimensional modeling. Ocean Engineering **107**, 222–236 (2015)
25. Jacobsen, N.G., Fuhrman, D.R., Fredsøe, J.: A wave generation toolbox for the open-source CFD library: OpenFOAM®. International Journal for Numerical Methods in Fluids **70**, 1073–1088 (2012)
26. King, A., Algie, C., Ryan, S., Ong, R.: Modelling of fluid structure interactions in submerged flexible membranes for the bombora wave energy converter. In: 20th Australasian Fluid Mechanics Conference, Perth, Australia (2016)
27. Li, L., Tan, M., Blake, J., et al.: Numerical simulation of multi-body wave energy converter. In: The Twenty-fifth International Offshore and Polar Engineering Conference. International Society of Offshore and Polar Engineers (2015)
28. Loh, T.T., Greaves, D., Maeki, T., Vuorinen, M., Simmonds, D., Kyte, A.: Numerical modelling of the WaveRoller device using OpenFOAM®. In: Proceedings of the 3rd Asian Wave & Tidal Energy Conference (2016)
29. Medina-Lopez, E., Allsop, W., Dimakopoulos, A., Bruce, T.: Conjectures on the failure of the OWC breakwater at Mutriku. In: Coastal Structures (2015)
30. Mendoza, E., Chávez, X., Alcérreca-Huerta, J.C., Silva, R.: Hydrodynamic behavior of a new wave energy convertor: The blow-jet. Ocean Engineering **106**, 252–260 (2015)
31. Mishra, V., Beatty, S., Buckham, B., Oshkai, P., Crawford, C.: Application of an arbitrary mesh interface for CFD simulation of an oscillating wave energy converter. In: Proc. 11th Eur. Wave Tidal Energy Conf, pp. 07B141–07B1410 (2015)
32. Palm, J.: Connecting OpenFOAM® with matlab. Online: http://www.tfd.chalmers.se/hani/kurser/OSCFD2012/ (2012)
33. Palm, J., Eskilsson, C., Paredes, G.M., Bergdahl, L.: Coupled mooring analysis for floating wave energy converters using CFD: Formulation and validation. International Journal of Marine Energy **16**, 83–99 (2016)
34. Rafiee, A., Fiévez, J.: Numerical prediction of extreme loads on the CETO wave energy converter. 11th European Wave and Tidal Energy Conference (EWTEC). Nantes, France (2015)
35. Rajagopalan, K., Nihous, G.: Study of the force coefficients on plates using an open source numerical wave tank. Ocean Engineering **118**, 187–203 (2016)
36. Ransley, E., Greaves, D., Raby, A., Simmonds, D., Hann, M.: Survivability of wave energy converters using CFD. Renewable Energy **109**, 235–247 (2017). https://doi.org/10.1016/j.renene.2017.03.003
37. Ransley, E., Greaves, D., Raby, A., Simmonds, D., Jakobsen, M., Kramer, M.: RANS-VOF modelling of the wavestar point absorber. Renewable Energy **109**, 49–65 (2017). https://doi.org/10.1016/j.renene.2017.02.079

38. Ransley, E.J.: Survivability of wave energy converter and mooring coupled system using CFD. Ph.D. thesis, Plymouth University, UK (2015)
39. Ringwood, J.V., Bacelli, G., Fusco, F.: Energy-maximizing control of wave-energy converters: the development of control system technology to optimize their operation. IEEE Control Systems **34**(5), 30–55 (2014)
40. Ringwood, J.V., Davidson, J., Giorgi, S.: Numerical Modeling of Wave Energy Converter: State-of-the-art techniques for single WEC and converter arrays, chap. Identifying models using recorded data. Elsevier (2016)
41. Schmitt, P., Asmuth, H., Elsäßer, B.: Optimising power take-off of an oscillating wave surge converter using high fidelity numerical simulations. International Journal of Marine Energy **16**, 196–208 (2016)
42. Schmitt, P., Elsaesser, B.: On the use of OpenFOAM® to model oscillating wave surge converters. Ocean Engineering **108**, 98–104 (2015)
43. Simonetti, I., Cappietti, L., El Safti, H., Oumeraci, H.: 3d numerical modelling of oscillating water column wave energy conversion devices: current knowledge and OpenFOAM® implementation. In: 1st International Conference on Renewable Energies Offshore (2014)
44. Simonetti, I., Cappietti, L., El Safti, H., Oumeraci, H.: Numerical modelling of fixed oscillating water column wave energy conversion devices: Toward geometry hydraulic optimization. In: ASME 2015 34th International Conference on Ocean, Offshore and Arctic Engineering, pp. V009T09A031–V009T09A031. American Society of Mechanical Engineers (2015)
45. Simonetti, I., Crema, I., Cappietti, L., El Safti, H., Oumeraci, H.: Site-specific optimization of an OWC wave energy converter in a Mediterranean area. In: Progress in Renewable Energies Offshore, pp. 343–350. CRC Press (2016)
46. Vyzikas, T., Deshoulieres, S., Giroux, O., Barton, M., Greaves, D.: Numerical Study of fixed Oscillatin Water Column with RANS-type two-phase CFD model. Renewable Energy **102**, 294–305 (2017)
47. Windt, C., Davidson, J., Schmitt, P., Ringwood, J.V.: Assessment of numerical wave makers. In: Proceedings of the 12th European wave and tidal energy conference (EWTEC 2017), Cork (2017)
48. Yu, Y.H., Li, Y.: Reynolds-averaged navier–stokes simulation of the heave performance of a two-body floating-point absorber wave energy system. Computers & Fluids **73**, 104–114 (2013)

Floating Potential Boundary Condition in OpenFOAM®

Nils Lavesson and Tor Laneryd

Abstract In OpenFOAM®, the powerful CFD solver can be combined with an electrostatic solver, allowing multiphysics analysis on the same mesh within the same numerical framework, which offers advantages for applications in high voltage power devices. One important piece of missing functionality in OpenFOAM® is a boundary condition for the electrostatic solver that can apply a floating potential to conducting parts that are not connected to any fixed potential and should be treated as an equipotential surface of undefined potential. This chapter describes the theoretical background and an OpenFOAM® implementation of a numerical algorithm that can accurately solve the electrostatic problem for a domain that includes several floating potentials.

1 Introduction

In power transmission and distribution systems, high voltage bushings are essential components for insulating conductors that carry high voltage and current through a grounded enclosure, such as the transformer tank shown in Fig. 1. For low voltage applications, a homogeneous layer of an insulating material, such as porcelain, glass, cast resin, or paper, provides sufficient insulation, but for higher voltages, this solution becomes too bulky to be practical. An alternative solution is to intersperse the insulation material with metallic conductive layers at appropriate intervals, designed to control the electric field distribution by the partial capacitances so as to generate a more uniform radial and axial stress distribution within the limits of the insulation material [2]. This is typically achieved by using tightly wound paper as the insulating material and applying screens of aluminum foil or ink during the winding process [5]. The paper is later impregnated with oil or resin to improve the dielectric strength. A schematic view of the condenser core of a bushing is shown in Fig. 2.

N. Lavesson · T. Laneryd (✉)
ABB Corporate Research, Västerås, Sweden
e-mail: tor.laneryd@se.abb.com

N. Lavesson
e-mail: nils.lavesson@se.abb.com

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_13

Fig. 1 600 kV HVDC transformer bushings

Simulating the electric stress in a condenser core requires a solver that can handle floating potentials. For similar studies based on other solvers, see [1, 3, 6].

As with all high voltage components, the reliability of the bushing is of extreme importance. The degradation of insulation material is a slow process that depends both on temperature and electric field strength. A proper description thus requires a multiphysics approach that includes the numerically challenging natural convection cooling. By using OpenFOAM®, the powerful CFD solver can be combined with an electrostatic solver, allowing the problem to be solved on the same mesh within the same numerical framework. The missing functionality in OpenFOAM® is a boundary condition for the electrostatic solver that can apply a floating potential to the metallic conductors of the bushing condenser core, which the present work aims to address. Here, a floating potential is defined as a single piece of highly (assumed infinite) conductive material, which is insulated from potential sources. The electric potential of the floating potential depends on its charge, as well as the geometry of the problem, including any fixed potentials that are specified as boundary conditions.

2 Theoretical Background

All conductors, including those placed at floating potentials, are assumed to have an infinitely high conductivity. Infinite conductivity means that all charge is accumulated on the surface in such a configuration that the electric field inside the conductor is

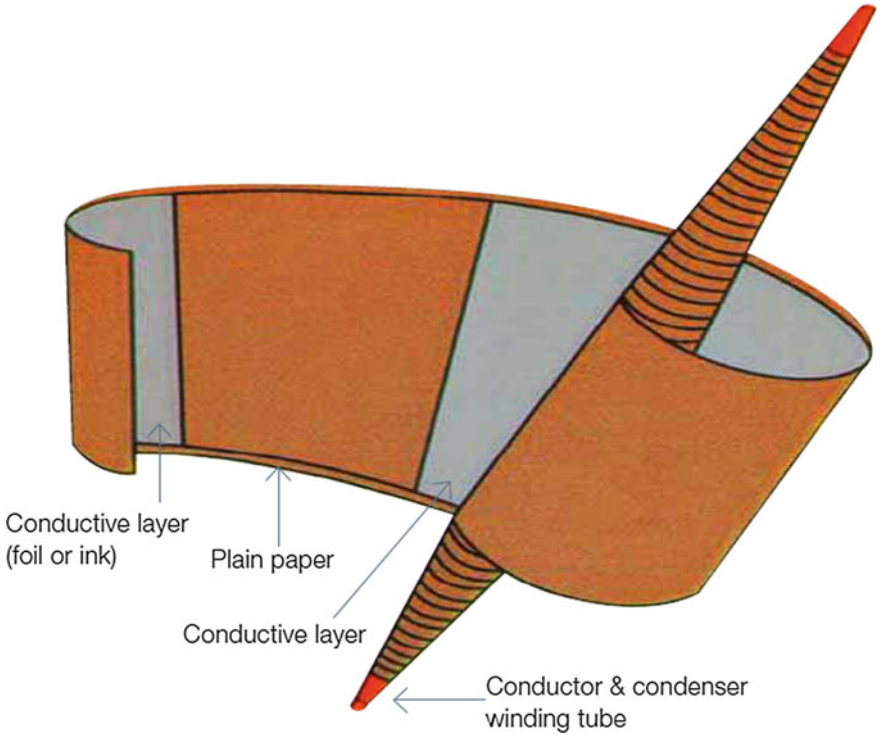


Fig. 2 Condenser bushing design principle (from [5])

zero. Hence, the surface of the conductor becomes an equipotential. If transients fast enough to create a potential difference along the conductor are of interest, then a more advanced model has to be considered. Analytical models are available for a few geometries such as when the floating potential is an infinite plane or a sphere. A mapping of space charge and potential sources can then be applied in order to create equipotential surfaces. For more general problems, numerical methods provide the only viable option.

The task is to numerically calculate the electric potential distribution for a geometry with several floating potentials, which may also include boundaries at fixed potential and space charge. It is assumed that the total charge of each floating potential is known and denoted Q_k for the k:th floating potential. The electric potential is calculated using Poisson's equation

$$\begin{aligned} \nabla \cdot (\varepsilon_r \varepsilon_0 \nabla \Phi) &= -\rho \\ \mathbf{E} &= -\nabla \Phi, \end{aligned} \tag{1}$$

where Φ denotes the electric potential, $\mathbf{E}$ the electric field, ρ the charge density, ε_r the relative permittivity, and ε_0 the permittivitty of vacuum. The floating potentials are treated as an external boundary with a fixed electric potential.

The proposed method is to calculate Eq. (1) iteratively, modifying the potentials so that the calculated charge approaches the actual charge. By doing the calculations in this way, we avoid having to calculate the surface charge of the conducting parts explicitly. The calculation starts from an initial guess of the electric potential provided by the user. Assuming at step n, a known electric potential Φ_k^n for each conductor at floating potential, Eq. (1) is solved, and then the charge can be evaluated using Gauss's law by integrating the electric field along the surface

$$Q_k^n = \oint \mathbf{D}^n \cdot \mathrm{d}\mathbf{S}_k = \oint \varepsilon_r \varepsilon_0 \mathbf{E}^n \cdot \mathrm{d}\mathbf{S}_k, \tag{2}$$

where $\mathbf{D}$ is the electric displacement field. The total charge of each conductor at floating potential is then compared to the actual charge and the electric potentials assigned to the floating potentials are adjusted to approach the correct solution.

To implement the proposed method for one floating potential is straightforward. Any optimization algorithm will quickly find the value of the electric potential that gives the correct charge. However, when several floating potentials are present, the problem becomes a lot more complicated. Frequently, one floating potential screens another, which leads to strong cross-couplings. Optimizing a strongly coupled system of floating potentials is possible, but a preferred way is to calculate the cross-couplings and utilize this information.

By properly calculating all cross-coupling between the floating potentials, the optimization of the electric potentials can be performed globally. Given a system of n conductors placed at floating potential, there exists a relation between the electric potentials and the charges of the conductors

$$Q_i^n = C_{ij} \Phi_j^n, \tag{3}$$

where C is a matrix describing the capacitances [4]. If conductors placed at fixed potentials or space charge are included, the charge of the floating potentials would no longer be zero if the potentials were all set to zero. Hence, an offset is needed, leading to the equation

$$Q_i^n = Q_{0i} + C_{ij} \Phi_j^n. \tag{4}$$

By changing the value of one potential and solving the electrostatic problem, it is possible to calculate the values in one column of the C matrix. If the potential Φ_j^n is changed by $\Delta\Phi_j$ and the corresponding change in the charges are denoted by ΔQ_i, then

$$C_{ij} = \frac{\Delta Q_i}{\Delta \Phi_j}. \tag{5}$$

The process is then repeated for all the potentials to calculate the full capacitance matrix. A new value for the floating potentials is calculated by

$$\Phi^{n+1} = \Phi^n + C^{-1}\left(Q - Q^n\right). \tag{6}$$

This equation is applied iteratively, and typically only a few iterations are required before the numerical errors due to the finite mesh size in Poisson's solver start to dominate and no further improvements are possible.

The entire algorithm is summarized in the following steps:

1. Calculate an initial guess for the electric field and charge of each floating potential (Q_k^0) using Eqs. (1) and (2) and the initial guess for the electric potentials (Φ_k^0) provided by the user.
2. For each floating potential, modify the electric potential and recalculate the electric field and charge of the floating potentials using Eqs. (1) and (2).
3. Calculate the capacitance matrix according to Eq. (5) using the results from step 2.
4. Calculate an improved value for the electric potentials for all floating potentials according to Eq. (6) using the initial value Q^0 from step 1 and the capacitance matrix calculated in step 3. Use the new electric potentials (Φ^1) to calculate improved values for the charge (Q^1) using Eqs. (1) and (2).
5. Repeat the procedure in step 4 iteratively until no further improvements are noted in the difference between the calculated charge Q^n and the actual charge Q.

The model can easily be adapted to work with time-dependent problems. A separate condition describing how much charge is injected or absorbed from the surface of the floating potential is then required, typically by integrating the current along the surface of the floating potential according to

$$\frac{dQ_i}{dt} = \oint \mathbf{J} \cdot \mathrm{d}\mathbf{S}_i. \tag{7}$$

This is then used to update the values of the charge of each floating potential, and the new values for the charge are then used the next time the electrostatic problem is solved.

The procedure for solving an electrostatic problem with floating potentials is numerically many times more expensive compared to solving a similar problem without floating potentials. Given k floating potentials, Poisson's equation has to be solved $k + 1$ times to calculate the capacitance matrix and another few times to achieve a stable solution. In OpenFOAM®, the Poisson solver has to be iterated in order to remove errors from nonorthogonal contributions. In addition, a system of linear equations has to be solved, which is typically a problem with the complexity of order k^3.

For a static solution, this increases the computational time substantially, but this can usually be accommodated. For a typical time-dependent problem, Poisson's equation is already computed a large number of times and the additional computational

work for adding floating potentials is usually small. This can be achieved, since it is usually enough to calculate the capacitance matrix once, and the matrix (and matrix factorization) can then be reused for the rest of the simulation. In exceptional cases, the capacitance matrix may require recalculation, but if this can be done after a substantial number of time steps, the additional increase in simulation time will be minor.

3 Implementation in OpenFOAM®

A boundary condition type for OpenFOAM® has been implemented in a library containing two modules. The first module is a new boundary condition derived from the fixed value boundary condition. Functions have been added to calculate the total charge of the floating potential and to update the charge according to the electric current in time-dependent problems. The attributes of the boundary condition include the current value of potential and charge and references to the electric current and displacement field.

The second part of the floating potential implementation is a class that takes care of calculating the capacitances and adjusting the values of the potentials. This class is intended to be instantiated and called directly from the main solver. The class contains functions for updating the capacitance matrix and the floating potentials. The capacitance matrix is decomposed using LU factorization from the GNU scientific library, which was chosen because the main solvers in OpenFOAM® are built for sparse matrices. In this implementation, the GSL has to be linked to the main solver if floating potential is to be used. We have subsequently discovered that there is an LU solver available in the code (used, for instance, in the class simpleMatrix), which may also be used for the factorization.

To utilize the floating potential functionality in a solver, a floating potential object should be created before the time integration loop. The floating potential constructor initializes and calculates the capacitance matrix. Finally, a line calling the update function should be added after the electric displacement field has been calculated inside the time integration loop.

A set of boundaries describing the conductors at floating potential needs to be included in the geometry and given unique labels. Two main attributes should be specified: the initial charge and an initial guess for the potential. If an initial guess for the potential is not specified, a default value of zero can be used, but this may lead to convergence problems for a complicated geometry.

4 Examples

For validation, the OpenFOAM® implementation of floating potentials is initially tested for the problem of electrostatic field distribution between concentric spheres, which has an analytical solution. The inner sphere has a radius of $r_1 = 5$ cm and is

given a charge of 1 μC, whereas the outer sphere has a radius of r_2 = 10 cm and is connected to ground. The medium between the spheres is considered to be oil with a relative permittivitty of ε_r = 2.2. The geometry is implemented using cylindrical symmetry by constructing a wedge. The analytical solution is given by

$$\Phi = \frac{Q}{\left(\frac{4\pi\varepsilon r_1 r_2}{r_2 - r_1}\right)}. \tag{8}$$

For the parameters given above, this expression yields Φ = 40,853 V. For a sufficient number of mesh elements, the numerical solution in OpenFOAM® approaches the analytical solution, as demonstrated in Table 1. The electric potential is plotted in Fig. 3.

To test a more complicated geometry pertinent to the case of a condenser bushing, three floating potentials are introduced inside a piece of insulating solid material. The geometry has cylindrical symmetry and consists of a 10 × 10 cm domain outside a 5 cm radius electrode. The electrode is placed to the left and ground to the right, with symmetry boundary conditions at the top and bottom. A structured rectangular mesh

Table 1 Electric potential of the inner sphere as a function of the number of mesh elements

Mesh elements between inner and outer spheres	Electric potential of inner sphere (V)
50	40,860
100	40,854
200	40,853

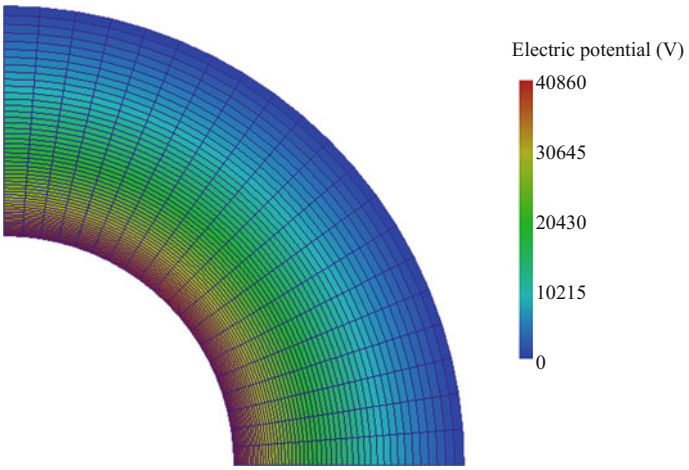


Fig. 3 Electric potential for two concentric spheres

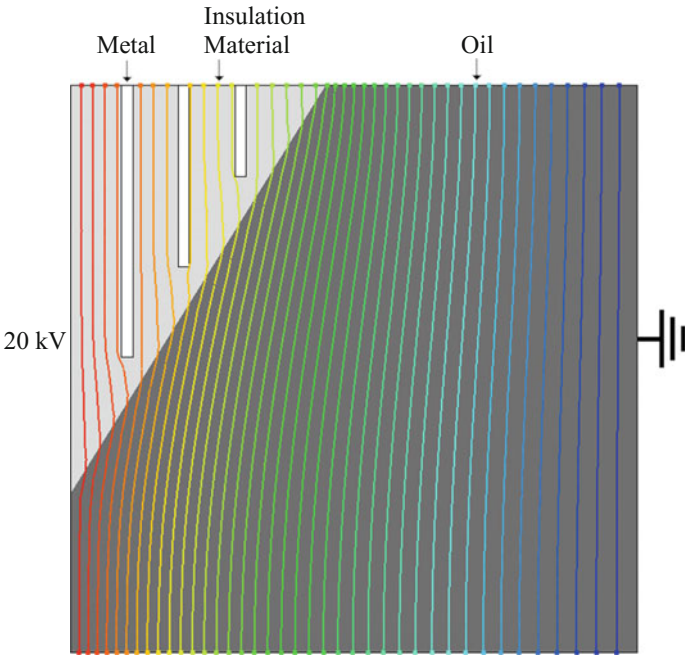


Fig. 4 Test problem with three floating potentials

is applied near the floating potentials and the material boundary and an unstructured triangular mesh is used for the rest of the geometry. The mesh size is around 1 mm in the entire geometry. The results are shown in Fig. 4.

5 Conclusions

The proposed algorithm seems stable and not overly expensive, assuming the capacitance matrix can be reused. With the floating potential BC, we can use OpenFOAM® for multiphysics analysis of high voltage bushings on a single mesh in a single numerical framework. The BC will also be useful for time-dependent simulations.

References

1. Chakravorti S, Steinbigler H (1998) Capacitive-resistive field calculation on HV bushings using the boundary-element method. IEEE Trans Dielectr Electr Insul 5(2):237–244
2. Graham JS (2001) High Voltage Bushings. In: Ryan HM (ed) High voltage engineering and testing, 2nd edn. The Institution of Electrical Engineers, London

3. Hesamzadeh MR, Hosseinzadeh N, Wolfs P (2008) An advanced optimal approach for high voltage AC bushing design. IEEE Trans Dielectr Electr Insul 15(2):461–466
4. Jackson JD (1999) Classical Electrodynamics. 3rd edn. John Wiley & Sons, New York, p 40–43
5. Jonsson L, Johansson R (2009) High-voltage bushings. ABB review 3:66–70
6. Li N et al (2015) Electric field analysis and optimal design of inner electrodes for HVDC wall bushing. Paper presented at the 2015 IEEE 11th International Conference on the Properties and Applications of Dielectric Materials (ICPADM), UNSW, Sydney, Australia, 19–22 Jul 2015

Fluid Dynamic and Thermal Modeling of the Injection Molding Process in OpenFOAM®

Jozsef Nagy and Georg Steinbichler

Abstract For the description of the filling, packing, and cooling phases of the injection molding process, a simulation framework of a compressible two-phase fluid model with polymer-specific material models is established and validated with experimental results. With this approach, it is possible to describe the fluid dynamic, the rheological, and the thermal behavior of the material during the production process. The main focus of this work is on the description of the standard injection molding process of common thermoplastic materials for industrial application, with special focus on process relevant quantities, e.g., pressure, temperature, as these values are of utmost importance for understanding the underlying phenomena and comparing the results to experimentally measured values.

1 Introduction

Computational Fluid Dynamics (CFD) has been successfully utilized in a variety of fields in chemical and mechanical engineering, e.g., combustion simulation [1–3], high velocity flows [4–6], and the polymerization of thermoplastic polymer materials [7, 8]. For the understanding of the underlying physical phenomena, it is of great significance for the optimization of geometries and processes governed by local flow phenomena, as well as global processing conditions.

In the polymer processing industry, especially in injection molding, the reduction of the development time of tools and machines has been one of the most important issues in recent decades. A large variety of processed materials are available with varying properties. Most importantly, the material's behavior differs from that of usual Newtonian fluids in the non-Newtonian behavior and the order of magnitude of the viscosity (≈10–10000 Pas). The value can vary, depending on the shear rate, the

J. Nagy (✉) · G. Steinbichler
Institute of Polymer Injection Moulding and Process Automation,
Johannes Kepler University, Altenbergerstrasse 69, Linz, Austria
e-mail: jozsef.nagy@jku.at

G. Steinbichler
e-mail: georg.steinbichler@jku.at

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_14

temperature, and the pressure. With the addition of the compressibility of the liquid polymer, machines have to be able to handle considerable changes in processing conditions.

A progressive trend has been emerging during the recent years, in which computer-aided optimization has been proven to be one of the key steps in the development of machines and processes in injection molding. In simulations, the process can be visualized three-dimensionally, as opposed to experiments, in which processing conditions can only be monitored at certain locations [9, 10]. Additionally, all quantities (e.g., density, viscosity, shear rate, velocity, etc.) needed for the simulation can be evaluated, as opposed to only the pressure or temperature values in experiments.

For a serious and general optimization methodology of development, the correctness of the results has to be guaranteed for all possible geometries, processing conditions, and materials. OpenFOAM® offers an excellent foundation for the development of a tool for the description of polymer processing [7, 8], and especially the injection molding process [9, 10].

Due to the complexity and interdependency of quantities of fluid dynamic and thermal processes during the discontinuous process, it has to be guaranteed that a wide variety of phenomena can be modeled correctly. It is important to focus on dominant phenomena in the process in order to reduce calculation time for the industrial application.

With the proposed models, experimental validation shows promising agreement in both pressure and temperature during the entire discontinuous process of injection molding. The good agreement achieved promises the possibility of using OpenFOAM® as an intrinsic part of the development stage of injection molding tools and machines, as well as of a certain machine optimization simulation methodology of an intelligent, self-regulating injection molding machine.

2 Governing Equations

The physics of the injection molding process is governed by the standard compressible equations of Computational Fluid Dynamics (CFD), with the addition of material-specific models that describe the complex material behavior of polymers.

2.1 Fluid Dynamic Equations

For the description of the fluid movement, a set of coupled nonlinear partial differential equations is employed. The continuity and the momentum equations are solved in the compressible form [11].

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{1}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \nabla \cdot \tau + \mathbf{F}. \tag{2}$$

In Eqs. (1) and (2), ρ is the mass density of the fluid, t is the time, $\mathbf{u}$ is the vector of velocity, p is the pressure, τ is the stress tensor, and $\mathbf{F}$ is a certain source term (e.g., surface tension).

Equations (1) and (2) are not solved directly, but are rather transformed into a Laplacian equation of the pressure. This approach of velocity–pressure coupling is commonly used in CFD [12] and is based on the approach taken in the OpenFOAM® solver compressibleInterFoam [11, 13]. The discretization schemes utilized for all partial differential equations are of second-order accuracy.

2.2 Thermal Modeling

In order to describe the thermal phenomena of the process, convection, heat diffusion, and shear heating have to be considered in a certain energy equation.

$$\frac{\partial \rho T}{\partial t} + \nabla \cdot (\rho \mathbf{u} T) = \Delta \left(\bar{k} T\right) + \tau : \nabla \mathbf{u} \cdot \left(\frac{\alpha}{c_{\mathrm{v_l}}}\right) + [\nabla \cdot (p\mathbf{u})] \left(\frac{\alpha}{c_{\mathrm{v_l}}} + \frac{(1-\alpha)}{c_{\mathrm{v_g}}}\right), \tag{3}$$

where T is the transported temperature, c_v is the constant heat capacity, and $\bar{k}$ is the thermal conductivity of the fluid divided by the heat capacity and weighted by the phase fraction (see details in Sect. 2.3). In addition to heat convection and diffusion, two terms are included in the balance equation. The first term is a dissipation function modeling the rate of work irreversibly converted into heat. Here, the dominant contribution to injection modeling is coming from shear phenomena, and this term is commonly referred to as "shear heating." The last term describes the rate of work for volume change [14].

Owing to the inexact surface of molds in injection molding, during processing, microscopic air entrapments can appear between the liquid polymer and the wall of the mold. Due to the heat resistance of the air, the temperature at the wall often shows a non-negligible jump [15]. Usually, the thickness of the gaseous layer is unknown. For this reason, the temperature distribution along the wall has to be modeled explicitly. This is similar to the modeling of turbulent phenomena near walls in CFD [16, 17]. Therefore, a certain heat transfer is assumed from the polymer melt into the wall with a certain heat transfer coefficient. This global "heat transfer coefficient" (HTC) models the unknown temperature distribution close to the wall, as well as the heat resistance of a possible thin gaseous layer. The coefficient is determined empirically, similar to the approach often taken in chemical engineering

[18]. With this coefficient, a temperature gradient can be calculated with both a spatial distribution and a temporal evolution.

$$\nabla T = \frac{-HTC\,(T_{\text{melt}} - T_{\text{wall}})}{k_{\text{l}}}. \tag{4}$$

Here, T_{melt} represents the temperature value in the center of the first cell next to the wall, T_{wall} the mold temperature, and k_l the thermal conductivity of the liquid polymer.

2.3 Multiphase Modeling

Multiple phases (liquid polymer and gaseous air) are implemented using the Volume-of-Fluid (VOF) method [19–21], in which a scalar quantity α is used for the liquid phase fraction that is transported with the velocity $\mathbf{u}$. Here, $\alpha = 1$ will denote the liquid polymer (index l) and $\alpha = 0$ will represent the gaseous air (index g). Equation (6) describes the transport of α.

$$\rho = \alpha\rho_{\text{l}} + (1 - \alpha)\,\rho_{\text{g}}, \tag{5}$$

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{u}) + \nabla \cdot [\alpha\,(1 - \alpha)\,\mathbf{u}_r] = S_p + S_u. \tag{6}$$

Simple discretization schemes might create a region of a diffuse phase interface between $\alpha = 1$ and $\alpha = 0$. The surface compression term with the compression velocity $\mathbf{u}_r$ helps in maintaining a sharp liquid–gas interface (details can be found in [19, 21]). The terms S_u and S_p are source terms introduced by the compressibility of the material [11].

For calculation of the material properties, Eq. (7) is applied.

$$\begin{pmatrix} \rho \\ \nu \\ c_{\text{v}} \\ \overline{k} \end{pmatrix} = \alpha \begin{pmatrix} \rho_{\text{l}} \\ \nu_{\text{l}} \\ c_{\text{v}_{\text{l}}} \\ k_{\text{l}}/c_{\text{v}_{\text{l}}} \end{pmatrix} + (1 - \alpha) \begin{pmatrix} \rho_{\text{g}} \\ \nu_{\text{g}} \\ c_{\text{v}_{\text{g}}} \\ k_{\text{g}}/c_{\text{v}_{\text{g}}} \end{pmatrix}, \tag{7}$$

where $\overline{k}$ is the weighted thermal conductivity divided by the respective specific heat capacity in Eq. (3). The dynamic viscosity of the fluid is calculated out of the density, as well as the kinematic viscosity, with $\mu = \alpha\rho_{\text{l}}\nu_{\text{l}} + (1 - \alpha)\rho_{\text{g}}\nu_{\text{g}}$.

2.4 Material Models

Given the complex behavior of the material, polymer-specific models have to be employed in order to correctly describe the production process.

2.4.1 Viscosity—Cross WLF Model

The Cross WLF model combines the shear-thinning effect of polymers [22] with the William-Landel-Ferry (WLF) model [23] for the description of the temperature and pressure dependence of the viscous behavior of polymers.

$$\nu_1(\dot{\gamma}, T, p) = \frac{\nu_0(T, p)}{1 + \left(\frac{\nu_0(T,p)\dot{\gamma}}{D_4}\right)^{1-n}}. \tag{8}$$

Here, D_4 defines the transition region from constant viscosity to the shear-thinning region, where the slope of the viscosity curve is given by the exponent n [22]. These constants are material-specific and have to be determined in rheological measurements. $\nu_0(T, p)$ is the projected viscosity at zero shear rate ($\dot{\gamma} = 0$) and is defined as

$$\nu_0(T, p) = D_1 \exp\left(\frac{(-A_1) \cdot (T - D_2 - D_3 \cdot p)}{A_2 + T - D_2 - D_3 \cdot p}\right). \tag{9}$$

The constants D_1, D_2, D_3, A_1, and A_2 are also material-specific and have to be determined in a manner similar to that of the constants in Eq. (8). The viscosity of air is considered to be constant.

2.4.2 Compressibility—Tait Model

The specific volume v of amorphous and semi-crystalline polymers behaves differently under the change of temperature and pressure [24]. In order to describe both material classes, the Tait model [25] is used, in which the dependence of the specific volume below a given transition temperature $T < T_{\text{trans}}$

$$T_{\text{trans}} = b_5 + b_6\, p \tag{10}$$

is given by

$$v_{\text{polymer}}(p, T) = \left\{ v_s(T) \left[1 - C \ln\left(1 + \frac{p}{B_s(T)}\right)\right] + W_s(T) \right\}, \tag{11}$$

$$v_{\mathrm{s}}(T) = b_{1\mathrm{s}} + b_{2\mathrm{s}}\,(T - b_5)\,, \tag{12}$$

$$B_{\mathrm{s}}(T) = b_{3\mathrm{s}} \exp\left(-b_{4\mathrm{s}}\,(T - b_5)\right), \tag{13}$$

$$W_{\mathrm{s}}(T) = b_7 \exp\left(b_8\,(T - b_5) - b_9\,p\right), \tag{14}$$

and above the transition temperature, the dependence is defined by

$T \geq T_{\mathrm{trans}}$

$$v_{\mathrm{polymer}}(p, T) = \left\{ v_{\mathrm{m}}(T) \left[1 - C \ln\left(1 + \frac{p}{B_{\mathrm{s}}(T)} \right) \right] \right\}, \tag{15}$$

$$v_{\mathrm{s}}(T) = b_{1\mathrm{m}} + b_{2\mathrm{m}}\,(T - b_5)\,, \tag{16}$$

$$B_{\mathrm{m}}(T) = b_{3\mathrm{m}} \exp\left(-b_{4\mathrm{m}}\,(T - b_5)\right), \tag{17}$$

where $C = 0.0894$. Here, all the constants are material-specific and have to be determined in, e.g., high-pressure-capillary experiments. With the specific volume, the density is calculated $\rho_{\mathrm{l}} = 1/v_{\mathrm{polymer}}(p, T)$. The air is considered to be an ideal gas.

2.5 Modeling Processing Steps of Injection Molding

Injection molding is a discontinuous process consisting of several steps. Depending on the definition of the starting point, the following main steps have to be considered [26, 27]:

1. Plastication: In the plasticating unit, a screw executes a rotational movement, thus moving solid polymer pellets in the direction of the mold, which has the shape of the final product. Due to friction and external heating, the polymer is melted.
2. Filling: The plasticating unit injects a preset amount of melt into the mold with a constant velocity. Simultaneously, the injected melt begins to cool down in the mold. Typically, a thin solid layer is formed immediately along the walls of the mold.
3. Packing: A constant holding pressure is applied by the screw of the plasticating unit to compensate for the volumetric shrinkage of the part during the cooling phase.
4. Cooling: The residual cooling time begins, the plasticating unit disconnects from the mold, and the screw prepares the next shot for the new cycle.
5. Part ejection: Once the part is sufficiently solidified and a predefined temperature is reached, the clamping unit opens and the mold ejects the part.

In this work, the injection process is considered, and thus the filling, packing, and cooling phases (steps 2–4) are considered as well. The processing conditions are changed between the filling and the packing phases from constant velocity to constant

Table 1 Boundary conditions of velocity and pressure during the phases of the process at the inlet

Phase	Velocity	Pressure
Filling	Time-dependent velocity profile	Zero gradient
Packing	Zero gradient	Fixed value
Cooling	Inlet–outlet	Fixed at 1 bar

pressure. During the cooling phase, the thermal phenomena are dominant and fluid dynamic processes can be neglected [26, 27]. In order to model these changes within one simulation run, the boundary conditions have to be changed during the runtime.

For this, two switches are used in the simulations. The first switch from the filling phase to the packing phase is given by an integral volumetric value. Once the liquid phase fraction α reaches this value (typically 0.98–0.995), the simulation changes the boundary condition required by the packing phase (see Table 1). The second temporal switch gives the time of the end of the packing phase, when the simulation changes the conditions to those in the cooling phase. These switches mimic common settings of injection molding machines [26, 27].

3 Experiments

3.1 Processing Conditions

Experiments with a standard hydraulic injection molding machine (Engel VC 200/50 tech with a screw diameter of 30 mm) are used to estimate the magnitude of errors arising due to assumptions made in the presented models (see Sect. 2). For this reason, typical processing conditions in injection molding are used.

- material: PP HD120MO,
- volume flux: 50 cm^3/s,
- melt and nozzle temperature: 240 °C,
- tool temperature: 30 °C,
- packing pressure: 338 bar,
- packing time: 26 s, and
- process time: 40 s.

The parts considered in these analyses are two shouldered test bars, for which the mold, the distribution channel, the sprue, the nozzle, and the first part of the screw chamber are considered in the simulations (see Fig. 1). In front of the screw chamber, a sensor (Kistler 4021B) monitors the pressure in order to completely describe the pressure drop throughout the entire volume. A second pressure sensor

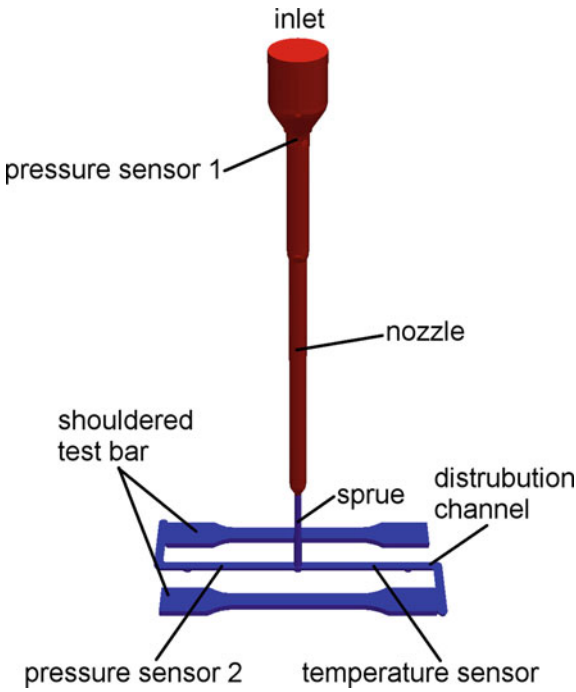


Fig. 1 Geometry of the modeled domain with the locations of the sensors. The red region is filled with melt and is also heated to a constant processing temperature (here, 240 °C)

(Kistler 6157BA) is located in the distribution channel so as to obtain the information about the pressure drop in the mold of the test bars. A symmetrically placed infrared temperature sensor (FOS MTS 408-IR-STS) measures the melt temperature.

3.2 Measurement Errors

It is important to know the experimental setup, particularly in order to quantify systematic errors. Both the pressure sensors and the infrared temperature sensors are calibrated for liquid polymers. However, a solid layer with a certain thickness arises during the injection process along the walls of the mold. This solid layer distorts and reduces both the pressure and the temperature signals. Figure 2 shows schematic sketches of the influence of the solid layer on the measurement signals. The force applied by the fluid pressure is redistributed by the solid layer in an undefined way, and similarly, the infrared radiation of the liquid polymer is modified by the solid layer distorting the temperature signal. In the experiments, a pressure reduction of up to 40 bars could be quantified at the end of the filling and the beginning of the packing phases and a temperature reduction of 15–20 °C was observed during the cooling phase. These values are inaccurate and have to be analysed in more detail; however, they do give a first estimation during the evaluation of the experimental results.

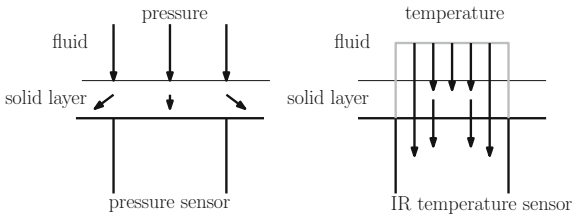


Fig. 2 Schematic sketch of the influence of the solid layer on the measurement signals

4 Validation

4.1 Filling Phase

In Fig. 3, the pressure evolution at the two sensor locations in both the experiment and the simulation is shown. At the first sensor location in front of the screw chamber, the polymer is heated to a constant processing temperature. At this location, the polymer is completely in a liquid state, avoiding measurement issues given by a solid layer (see Sect. 3.2). Thus, a good quantitative agreement between simulation and experiment can be found (see Fig. 3). At the second pressure sensor, the thin solid layer distorts the experimental pressure signal, thus giving the impression of a bigger deviation between experiment and simulation. However, the deviation is given mostly by the described systematic error within the experimental setup. In order to quantify the deviations of these values, three process quantities are compared here.

The first quantity is the location of the flow front. For that, the time of the first increase at the second pressure sensor is evaluated and referred to the filling phase.

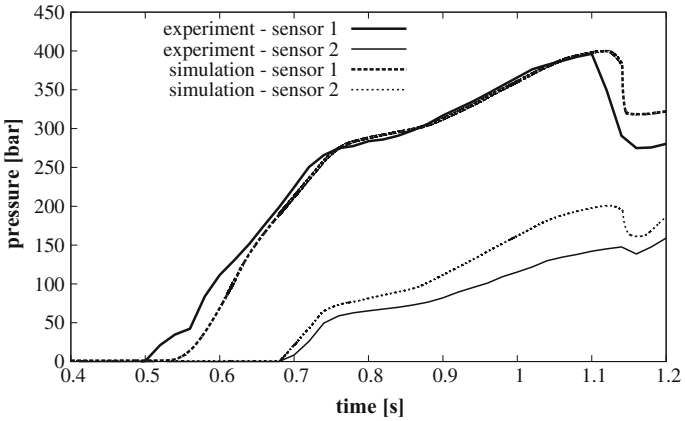


Fig. 3 Time evolution of the pressure at two sensor locations in the experiment and simulation during the filling phase of the process

$$\Delta t_{\text{fill}} = \frac{t_{\text{s2,sim}} - t_{\text{s2,exp}}}{t_{\text{switchover}} - t_{\text{start}}} = \frac{0.67\,\text{s} - 0.66\,\text{s}}{1.16\,\text{s} - 0.5\,\text{s}} = 1.5\%. \tag{18}$$

The second quantity is the maximum pressure in front of the screw chamber at the first sensor.

$$\Delta p_{\text{max}} = \frac{p_{\text{s1,sim}} - p_{\text{s1,exp}}}{p_{\text{max}} - p_{\text{start}}} = \frac{398.7\,\text{bar} - 396.3\,\text{bar}}{396.3\,\text{bar} - 1\,\text{bar}} = 0.6\%. \tag{19}$$

The last quantity in the filling phase is the maximum pressure in the cavity at the second pressure sensor.

$$\Delta p_{\text{cav}} = \frac{p_{\text{s2,sim}} - p_{\text{s2,exp}}}{p_{\text{max}} - p_{\text{start}}} = \frac{199.8\,\text{bar} - 147.7\,\text{bar}}{396.3\,\text{bar} - 1\,\text{bar}} = 13.2\%. \tag{20}$$

Considering this, the simulation calculates reasonable values for important values of the process (deviation $\leq$15%).

4.2 Packing Phase

During the packing phase, the pressure is kept at a constant level in the screw chamber, but the pressure distribution in the mold changes, due to the cooling of the material and the change in viscosity (see Fig. 4). Here, the deviation between experiment and simulation is increasing, due to the fact that the thickness of the solid layer increases with time. The change of the pressure slope given by the freezing of the material at this point is seen in both the experiment and the simulation at approximately 20–22 s. Although the quantitative validation of the pressure in this phase cannot be done due to the previously mentioned insufficiently quantified inaccuracies in the experiments, other process parameters, like the freezing of the material, can be derived out of the change of the slope of pressure.

In order to quantify the deviation between experiment and simulation, the time of freezing is evaluated. For this, the point of time when the first derivative of the pressure curves changes abruptly is utilized.

$$\Delta t_{\text{freeze}} = \frac{t_{\text{fr,sim}} - t_{\text{fr,exp}}}{t_{\text{filling+packing}} - t_{\text{start}}} = \frac{20.6\,\text{s} - 21.3\,\text{s}}{26\,\text{s} - 0.5\,\text{s}} = -2.7\%. \tag{21}$$

4.3 Cooling Phase

During the process, the temperature is reduced from processing temperature to values at which the part can be ejected (see Fig. 5). Here, the temporal evolution of the temperature in the simulation and the experiment is very similar. A certain deviation

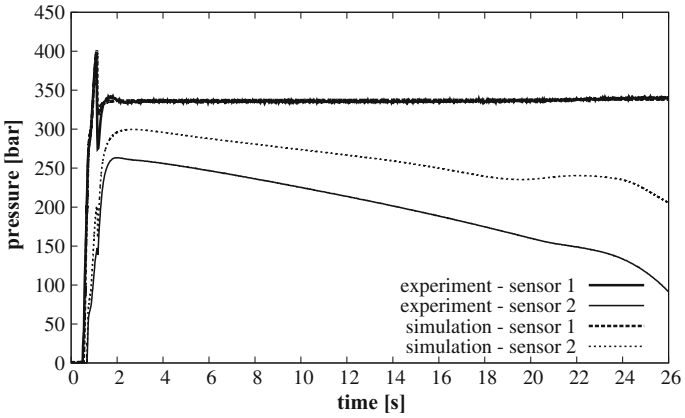


Fig. 4 Time evolution of the pressure at two sensor locations in the experiment and simulation during the filling and packing phases of the process

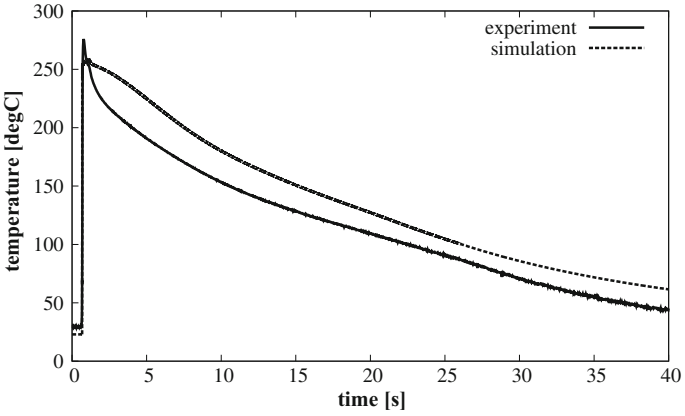


Fig. 5 Time evolution of the temperature in the experiment and simulation during the entire process

can be found during the cooling phase of approximately 20 °C, mostly arising from the fact, that the sensor is calibrated for liquid polymers and the solid thin layer distorts the signal (see Sect. 3.2).
Here, the temperature after 40 s is evaluated in order to quantify the deviation.

$$\Delta T_{\text{end}} = \frac{T_{\text{end,sim}} - T_{\text{end,exp}}}{T_{\max} - T_{\text{tool}}} = \frac{61.5\,^{\circ}\text{C} - 43.7\,^{\circ}\text{C}}{275\,^{\circ}\text{C} - 30\,^{\circ}\text{C}} = 7.3\%. \tag{22}$$

Table 2 Deviations (in %) between experiment and simulation with different volume flux values

Volume flux cm^3/s	Δt_{fill}	Δp_{max}	Δp_{cav}	Δt_{freeze}	ΔT_{end}
5	1.1	−4.3	9.4	0.33	10.9
30	−0.9	1.5	14.0	−2.3	8.6
50	1.5	0.6	13.1	−2.7	7.3
70	1.9	0.3	10.9	−3.9	6.6
90	2.0	2.5	12.6	−4.3	6.7

4.4 Parameter study

In order to check for consistency of the quality of the results, the volume flux is changed during the process ($5, 30, 50, 70, 90\,cm^3/s$). With this, the order of magnitude of the deviation between experiment and simulation should not change.

Table 2 shows the deviations with regard to the location of the flow front Δt_{fill}, the maximum pressure Δp_{max}, the mold pressure Δp_{cav}, the time of freezing Δt_{freeze}, and the temperature at the end of the process T_{end}. Independently from the changes in processing conditions, the same order of error is found supporting the idea that the dominant deviations are arising systematically in the experiments. Simulation results seem to correctly describe important quantities for processing.

5 Conclusion

The suggested approach of modeling the injection molding process with the compressible form of the continuity, the Navier–Stokes and the energy equations with polymer-specific material models, as well as the dynamic change of boundary conditions during runtime, promise a good agreement between simulations and experiments. However, it is of utmost importance to understand which quantities can be compared and at which points possible systematic errors are occurring.

With this, it is possible to analyze geometries, as well as materials, and use the implemented collection of utilities for the optimization of the injection molding process.

Acknowledgements The founding of the Österreichische Forschungsförderungsgesellschaft FFG within the project "Neuentwicklung von Spritzaggregaten und neue Ansätze für den Spritzgieprozess der Zukunft" is acknowledged here. The authors also give thank for the kind support of the ENGEL Austria GmbH.

References

1. Zhubrin, S.: Discrete reaction model for composition of sooting flames. Int. J. Heat Mass Transf. **52** (17–18), pp. 4125–4133 (2009)
2. Li, Y., Kong, S.-C.: Coupling conjugate heat transfer with in-cylinder combustion modeling for engine simulation. Int. J. Heat Mass Transf. **54** (11–12), pp. 24672478 (2011)
3. Nagy, J., Jordan, C., Harasek, M.: Optimization of an industrial high temperature furnace. In: Proceedings of the Third Open Source CFD International Conference Paris-Chantilly, France (2011)
4. Theofanous, T., Mitkin, V., Ng, C., Chang, C., Deng, X., Sushchikh, S.: The physics of aero-breakup. Part II: viscous liquids. Phys. Fluids. **24** (2), pp. 022104 (2012)
5. Nagy, J., Jordan, C., Harasek, M.: Numerical and experimental investigation of the role of asymmetric gas flow in the breakup of liquid droplets. In: Proceedings of the Fourth Open Source CFD International Conference London The Tower Hotel, ondon,Great-Britain (2012)
6. Nagy, J., Jordan, C., Harasek, M.: Turbulent phenomena in the aerobreakup of liquid droplets. CFD Letters. **3**, pp. 112–126 (2012)
7. Nagy, J., Reith, L., Fischlschweiger, M., Steinbichler, G.: Modeling the influence of flow phenomena on the polymerization of ε-Caprolactam. Chemical Engineering Science. **111**, pp. 85–93 (2014)
8. Nagy, J., Reith, L., Fischlschweiger, M., Steinbichler, G.: Influence of fiber orientation and geometry variation on flow phenomena and reactive polymerization of ε-caprolactam. Chemical Engineering Science. **128**, pp. 1–10 (2015)
9. Nagy, J., Kobler, E., Steinbichler G.: Influence of complex material behavior of polymer materials on the roduction process. In: Proceedings of the Tenth OpenFOAM® Workshop, Ann Arbor, USA (2015)
10. Nagy, J., Kobler, E., Wuschko, S., Steinbichler G.: Modeling and optimization of the injection molding process in OpenFOAM®. In: Proceedings of the Eleventh OpenFOAM® Workshop, Guimaraes, Portugal (2016)
11. Miller, S.T., Jasak, H., Boger, D.A., Paterson, E.G., Nedungadi, A.: A pressure-based, compressible, two-phase flow finite volume method for underwater explosions. Computers & Fluids. **87**, pp. 132–143 (2013)
12. Jasak, H.: Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows. PhD thesis. Imperial College of Science, Technology and Medicine (1996)
13. *OpenFOAM® source code*, OpenCFD Ltd. (ESI Group). http://www.openfoam.com and http://www.openfoam.org visited on 10.11.2016
14. Winter, H.H.: Viscous dissipation term in energy equations. AIChEMI Modular Instructions. Series C: Transport, Volume **7**: Calculation and Measurement Techniques for Momentum, Energy and Mass Transfer. pp. 27–34 (1987)
15. Brunotte, R.: Die thermodynamischen und verfahrenstechnischen Abläufe der in-situ-Oberflächenmodifizierung beim Spritzgiessen. PhD thesis. Technische Universit Chemnitz (2006)
16. Prandl, L.: Bericht über Untersuchungen zur ausgebildeten Turbulenz. Zeitschr. Für Angewandte Math. Und Mech.pp. 136–147 (1925)
17. von Karman, T., L.: Mechanische Ähnlichkeit und Turbulenz. In: Proceedings of the Third International Congress on Applied Mechanics, Stockholm, Sweden, (1930)
18. Gnielinski, V.: Wrmebertragung im konzentrischen Ringspalt und im ebenen Spalt. In VDI-Wrmeatlas. Springer Verlag Heidelberg Berlin (2013)
19. Rusche, H.: Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions. PhD thesis. Imperial College of Science, Technology and Medicine (2002)
20. Raessi, M., Mostaghimi, J., Bussmann, M.: A volume-of-fluid interfacial flow solver with advected normals. Computers & Fluids. **39** (8), pp. 1401–1410 (2013)
21. Nagy, J.: Untersuchung von mehrphasigen, kompressiblen Strömungen durch Simulation und Experiment. PhD thesis. Technische Universitt Wien (2012)

22. Cross, M.M.: Rheology of non-newtonian fluids a new flow equation for pseudoplastic systems. J. colloid Sci. **20**, pp. 417–437 (1965)
23. Williams, M.L., Landel, R.F., Ferry, J.D.: Mechanical properties of substances of high molecular weight. 19. The temperature dependence of relaxation mechanisms in amorphous polymers and other glass-forming liquids. J. Am. Chem. Soc. **77**, pp. 3701-3707 (1955)
24. Zoller, P., Fakhreddine, Y.A.: Pressure-volume-temperature studies of semicrystalline polymers. Thermochimica Acta **238**, pp. 397–415 (1994)
25. Wang, J.: PVT Properties of Polymers for Injection Molding, Some Critical Issues for Injection Molding. InTech. 2012 http://cdn.intechopen.com/pdfs/33643/InTech-Pvt_properties_of_polymers_for_injection_molding.pdf cited 07 Nov 2016
26. Bonten, C.: Kunststofftechnik: Einführung und Grundlagen. Hanser (2014)
27. Zheng, R., Tanner, R. I., Fan, X.-J.: Injection Molding. In VDI-Wrmeatlas. Springer Verlag Heidelberg Berlin (2011)

Free-Surface Dynamics in Induction Processing Applications

Pascal Beckstein, Vladimir Galindo and Gunter Gerbeth

Abstract Induction processing technology is widely applied in the metallurgical and crystal growth industry where conducting or semi-conducting material is involved. In many applications, alternating magnetic fields, which are used to generate heat and force, occur together with a free-surface flow. The numerical analysis of such three-dimensional, multi-physical phenomena on the industrial scale is still a big challenge. We present an overview of a novel multi-mesh model for addressing these kinds of coupled problems by means of computational simulations. It is based on the Finite Volume Method (FVM) of the software *foam-extend* (http://www.foam-extend.org)—an extended version of *OpenFOAM*® (Weller et al. in Computational Physics 12(6):620–631, 1998, [15]). Our development is motivated by the desire to investigate the so-called Ribbon Growth on Substrate (RGS) process. RGS is a crystallisation technique that allows for the production of silicon wafers and advanced metal silicide alloys Schönecker et al. (Solid State Phenomena 95-96:149-158 2004, [12]) with high volume manufacturing and outstanding material yield.

1 Introduction

In our devotion to obtaining a better understanding of the Ribbon Growth on Substrate (RGS) process [12], we were confronted with the question of how to realise simulations of free-surface flows under the influence of (strong) magnetic fields. Due to the spatial redistribution of conducting material in the proximity of the excitation coil, a two-way coupling exists between magnetodynamic and hydrodynamic effects.

Three-dimensional electromagnetic (eddy-current) simulations are computationally very expensive. They are usually formulated on the basis of the Finite Element Method (FEM). In contrast to that, in Computational Fluid Dynamics (CFD), FVM is normally the preferred vehicle for discretisation. Bringing together both methods,

P. Beckstein (✉) · V. Galindo · G. Gerbeth
Institute of Fluid Dynamics / Magnetohydrodynamics,
Helmholtz-Zentrum Dresden-Rossendorf,
Bautzner Landstraße 400, 01328 Dresden, Germany
e-mail: p.beckstein@hzdr.de

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM*®,
https://doi.org/10.1007/978-3-319-60846-4_15

even though it has been done successfully using commercial software (e.g. Spitans et al. [13]), produces lots of computational overhead due to recurring data exchange, including extraction, interpolation and re-meshing.

In the process of investigating the RGS process [2, 3], we have developed a new multi-mesh model. This relies only on FVM and comprises two major milestones. The first one is that, until recently, to the best of the authors' knowledge, there are only a small number of researchers dealing with eddy-current simulations based on FVM [1, 5]. Moreover, existing publications are either designed specifically for structured grids or limited with respect to performance and accessibility of code. We have created a similar, but very efficient, implementation in the freely available *OpenFOAM®* toolbox. Our second achievement is the design of a multi-physics framework for coupling already existing tools for CFD with our new electromagnetic solver.

With this paper, we endeavour to give a brief overview of the whole concept behind our recent development.

2 Magnetodynamics

For the description of the electromagnetic fields, we restrict ourselves to highly conducting materials, in which the local rate of change of electric charges and displacement currents may be neglected. Under the additional assumption of purely non-magnetic materials, the electromagnetic fields are governed within an unbounded domain Ω by the following simplification of Maxwell's equations, including current conservation:

$$\Omega \; : \quad \nabla \times \mathbf{B} = \mu_0 \mathbf{j}; \quad \nabla \cdot \mathbf{B} = 0; \quad \nabla \times \mathbf{E} = -\partial_t \mathbf{B}; \quad \nabla \cdot \mathbf{E} = \rho_\mathrm{E}/\epsilon_0, \tag{1}$$

$$\Omega \; : \quad \nabla \cdot \mathbf{j} = 0. \tag{2}$$

Given the physical time t, μ_0, ϵ_0, $\mathbf{B}$ and $\mathbf{E}$ denote the constant vacuum permeability and permittivity, magnetic flux density and the electric field intensity, respectively. The density of electrical charges is represented by ρ_E and the total current density $\mathbf{j} = \mathbf{j}_0 + \mathbf{j}'$ consists of two parts: a purely induced part $\mathbf{j}'$ and a part that arises only from an externally applied current source $\mathbf{j}_0$.

The domain Ω may be split into a non-conducting Ω_0 and enclosed conducting domain Ω_C, as sketched in Fig. 1. For media with an isotropic electrical conductivity σ and cases with low Magnetic Reynolds Numbers $\mathrm{R_m} = \mu_0 \sigma U L \ll 1$ (characteristic fluid velocity U and length L), the induced current density $\mathbf{j}'$ is proportionally related to $\mathbf{E}$ via Ohm's law

$$\Omega_\mathrm{C}: \quad \mathbf{j}' = \sigma \mathbf{E}. \tag{3}$$

The FVM on unstructured meshes, as it is used in *OpenFOAM®*, shows some limitations compared to, e.g. FEM regarding implicit discretisation of the curl-operator

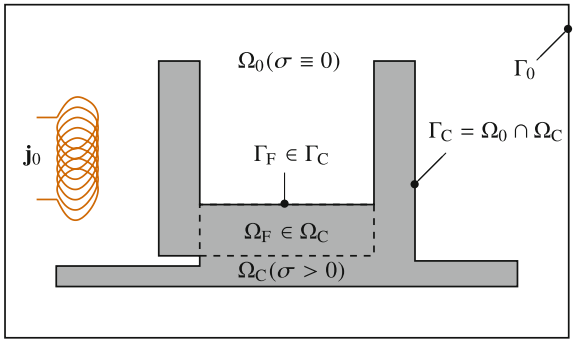


Fig. 1 Typical domain decomposition for an application like RGS: Conducting material Ω_C, including a liquid part Ω_F, is surrounded by a non-conducting region Ω_0. The latter contains an excitation coil, which is driven by the source current density $\mathbf{j}_0$. The far-field boundary Γ_0 represents a numerical truncation. The conductor boundary Γ_C includes the free-surface Γ_F of the fluid region

$\nabla \times ()$. We therefore use an alternative description for the system of Eqs. (1) and (2), which relies more on differential operators that are typical for CFD and are known from the Navier–Stokes equations. Based on the Coulomb-gauged magnetic vector potential $\mathbf{A}$ and the electric scalar potential ϕ

$$\mathbf{B} = \nabla \times \mathbf{A}; \quad \nabla \cdot \mathbf{A} = 0, \tag{4}$$

$$\mathbf{E} = -\left(\partial_t \mathbf{A} + \nabla\phi\right), \tag{5}$$

the reformulated system, including (3), reads as

$$\Omega : \quad \nabla^2 \mathbf{A} = \mu_0 \sigma \left(\partial_t \mathbf{A} + \nabla\phi\right) - \mu_0 \mathbf{j}_0, \tag{6}$$

$$\Omega_C: \quad \nabla \cdot (\sigma \nabla\phi) = -\partial_t \mathbf{A} \cdot \nabla\sigma. \tag{7}$$

The unbounded domain Ω is numerically truncated (cf. Fig. 1) such that the resulting cut boundary Γ_0 is sufficiently far away from any current sources. With this assumption, a homogeneous Dirichlet-type boundary condition may be used for $\mathbf{A}$ at Γ_0. On the conductor boundary Γ_C, the gradient of ϕ depends on the time derivative of $\mathbf{A}$ in terms of an inhomogeneous Neumann-type boundary condition to obtain a zero current density flux. Numerical treatment of Eq. (7) requires adequate discretisation schemes in case of possible jumps in the distribution of σ in Ω_C.

The induced current $\mathbf{j}'$ in this region causes a Lorentz force according to

$$\Omega_C: \quad \mathbf{F}_L = \mathbf{j}' \times (\nabla \times \mathbf{A}). \tag{8}$$

This includes a volume force within the fluid region. In most industrial applications, the source current $\mathbf{j}_0$ originates from an excitation coil with a time-harmonic oscillation at frequency f. If this is the case, a complex, quasi-steady formulation of

Eqs. (6) and (7) is more convenient, where all time derivatives are replaced with a complex-valued angular frequency ($\partial_t = i\omega$). If the timescale of magnetodynamic effects is much smaller compared to the coupled physics, it is sufficient to consult the time-averaged Lorentz force:

$$\Omega_C: \quad \langle \mathbf{F}_L \rangle_t = f \int_0^{f^{-1}} \mathbf{F}_L \, \mathrm{d}t. \tag{9}$$

3 Hydrodynamics

The dynamic behaviour of the liquid part Ω_F of the conductor Ω_C (cf. Fig. 2) is governed by the incompressible Navier–Stokes equations. For the sake of simplicity, we make the assumption of an isothermal system and constant material properties with respect to time. The resulting equation system yields

$$\Omega_F: \quad \rho \left[\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u}\right] = \nabla \cdot \boldsymbol{\tau}' + \langle \mathbf{F}_L \rangle_t; \quad \nabla \cdot \mathbf{u} = 0, \tag{10}$$

$$\boldsymbol{\tau}' = \eta \left[\nabla \mathbf{u} + \{\nabla \mathbf{u}\}^T\right] - p' \mathbf{I}, \tag{11}$$

$$p' = p - \rho \, (\mathbf{g} \cdot \mathbf{x}) \, . \tag{12}$$

The equations contain the fluid velocity $\mathbf{u}$ and the physical pressure p. $\boldsymbol{\tau}'$ is the stress tensor based on a modified diagonal fluid pressure p' without the hydrostatic part of p according to the spatial coordinate $\mathbf{x}$ and the gravitational vector $\mathbf{g}$.

This system has been set up in accordance with the model presented by Tukovic and Jasak [14], in which the free-surface boundary is represented by means of a surface-tracking method. Boundary conditions for $\mathbf{u}$ and p are derived from the dynamic condition (force balance) at the interface Γ_F. The normal force balance leads to a Dirichlet-type condition for the pressure, depending on the mean curvature κ and (constant) surface tension γ. A Neumann-type boundary condition for the

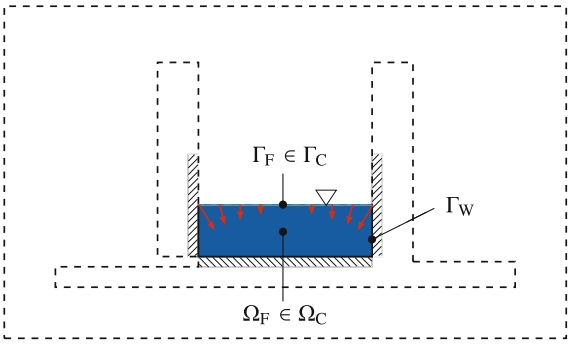


Fig. 2 Hydrodynamics inside the conducting region: The domain Ω_F contains conducting liquid, and is thus part of Ω_C. As a result of induced currents in Ω_C, a Lorentz force is acting on Ω_F, too. This is symbolised by small arrows. The fluid is being held in a solid frame. This wall boundary is denoted as Γ_W

gradient of velocity $\mathbf{u}$ in normal direction is the result of a tangential force balance. For both conditions, we neglect drag forces from the atmosphere at the opposite side of the free-surface, as its viscosity is several orders of magnitude smaller compared to, e.g. liquid metals. Further details can be found in the above-mentioned article by Tukovic and Jasak [14]. The main difference in our description is that we have an additional Lorentz force term in the momentum balance. Here, the force density has to be taken into account for wall boundary conditions (at Γ_W), as the pressure gradient is given by the normal component of $\langle \mathbf{F}_L \rangle_t$. At the contact line $\Gamma_F \cap \Gamma_W$, we specify an equilibrium wall contact angle of $\alpha_{\Gamma_F \cap \Gamma_W} = 90°$. Furthermore, we have extended our model with support for the turbulence and sub-grid-scale models available in *OpenFOAM®*.

As the free-surface is moving with respect to its kinematic condition, the mesh points representing Γ_F need to be adjusted. To maintain an acceptable mesh quality inside the domain, the inner mesh points also need to be updated over time.

We have preferred the interface-tracking method over a more flexible interface-capturing method because of several advantages. First of all, surface breaking is an undesirable effect for many industrial applications. Second, a sharp, high-quality representation of the free-surface is essential. There may be a huge jump in the electrical conductivity at Γ_F that, due to the so-called skin effect, may coincide with a strong concentration of most of the induced currents and forcing for higher frequencies. Finally, the moving mesh allows us to transport the time-averaged Lorentz force distribution $\langle \mathbf{F}_L \rangle_t$ with the mesh for a defined and reasonably small simulation time of the flow. This reduces the computational effort required to solve the whole coupled system.

4 Mesh Motion

In accordance with [14], a motion solver is used to calculate the mesh deformation in Ω_F from the time-varying shape of Γ_F. A Laplacian equation

$$\Omega_F: \quad \nabla \cdot (\gamma_M \nabla \mathbf{u}_M) = \mathbf{0}, \tag{13}$$

is solved for the mesh velocity $\mathbf{u}_M$ to achieve a preferably smooth mesh point distribution. We found that the artificial diffusion coefficient γ_M needs to be tailored for each test case (e.g. inversely proportional to the closest distance from Γ_F). Boundary conditions for Eq. (13) are calculated by interpolating the fluid velocity $\mathbf{u}$ at Γ_F to the corresponding mesh points. On Γ_W, suitable boundary conditions depend on the topology of the test case (e.g. slip/no-slip).

In order to compensate the boundary movement of Γ_F at the outside of Ω_F, which is necessary for a re-calculation of the electromagnetic Lorentz force, we have introduced a buffer region Ω_B, which is shown in Fig. 3. The size and shape of the buffer region are case-dependent and may comprise parts of Ω_C and Ω_0. Its introduction avoids a mesh movement of the whole domain Ω. However, Ω_B needs to be large

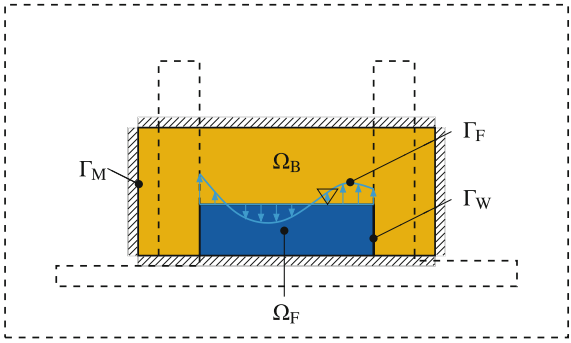


Fig. 3 Regions of dynamic mesh movement: The mesh points in Ω_F need to be re-adjusted according to the flow- and time-dependent shape of the free-surface Γ_F (and parts of Γ_W). A buffer region Ω_B is used to compensate the mesh motion in a sufficiently large area around the fluid domain. The mesh of the buffer region is fixed at its outer boundary Γ_M

enough to allow for a sufficiently smooth mesh deformation without too much degradation of the mesh quality. The same Laplacian equation as given in (13) is solved in Ω_B to redistribute the inner mesh points. Boundary conditions are calculated from a direct mapping of $\mathbf{u}_M$ at the shared part of Γ_W and Γ_F, whereas homogeneous Dirichlet-type conditions are used at Γ_M.

Equation (10) is strictly valid only for fixed meshes. For a dynamically moving mesh, a correction needs to be incorporated to address the translation from the Eulerian (absolute) to a Lagrangian (relative) frame of reference. This is known as the Arbitrary Lagrangian–Eulerian (ALE) method. The transformation itself results in a modified convection term $((\mathbf{u} - \mathbf{u}_M) \cdot \nabla)\mathbf{u}$ in Eq. (10), where $\mathbf{u} - \mathbf{u}_M$ represents the fluid velocity relative to the mesh velocity. For more information about the theory and implementation of the dynamic mesh motion in *OpenFOAM*®, the reader is referred to Jasak and Tukovic [8].

5 Multi-mesh Multi-physics

The backbone of our numerical model for solving a system like (6), (7) and (10) is a multi-region representation of all involved domains based on superposed meshes. This is illustrated in Fig. 4. A tailored use of multiple meshes facilitates a flexible description of different physical effects (multi-physics) in their (possibly overlapping) domains with corresponding boundary conditions, as shown in the last section. One-to-one mapping of the individual sub-mesh typologies onto a global, shared base-mesh allows us to propagate adjustments of the mesh geometry and connected field data without bulk (domain-internal) signed-distance interpolation. On this basis, we have extended *OpenFOAM*® by flexible multi-region fields with functionality for fast, bi-direct mapping and interpolation/extrapolation at shared boundaries.

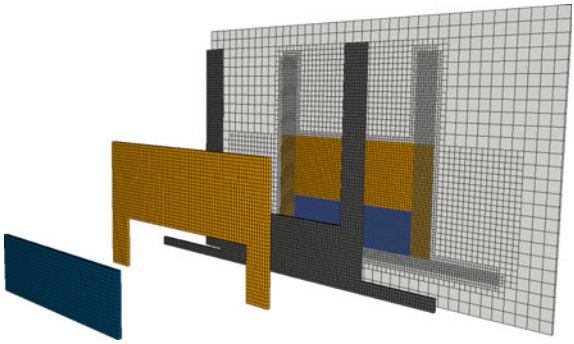


Fig. 4 Illustration of the multi-mesh concept: for each sub-mesh (left), there exists a corresponding one-to-one projection onto the base-mesh (rightmost)

5.1 Parallelisation

For computer simulations comprising a large number of cells, it is essential to be able to distribute a given problem over several processors while balancing the associated workload. Each sub-mesh of our multi-mesh representation (Fig. 4) is related to one or more governing equations from the last section, and is thus connected to a substantial part of the total computational effort.

To achieve an acceptable load balance, the most flexible approach to decompose the problem is to balance each mesh individually. We have realised this by means of the multi-constraint feature of the graph-partitioning software *METIS* (http://glaros.dtc.umn.edu/gkhome/metis/metis/overview), which is explained, e.g. in Karypis [9].

After an extension of *OpenFOAM*'s® interface for *METIS*, we are now able to perform fully parallelised simulations, in which each processor holds overlapping parts of all corresponding sub-meshes. This is depicted in Fig. 5. An advantageous side effect of this decomposition is that bulk data mapping does not involve undesired parallel communication.

5.2 Magnetohydrodynamic Solution

The solution procedure for the coupled system (6), (7) and (10) is split according to physical effects. The coupling is, in general, addressed by means of iterative data exchange and mesh topology updates.

A novel, efficient solution algorithm with a solver called *eddyCurrentFoam* has been developed to solve electromagnetic (eddy-current) problems in *OpenFOAM*®. This currently involves a quasi-static, semi-coupled multi-mesh approach to solve Maxwell's equations for non-magnetic materials, only. The major steps of the scheme are arranged in Fig. 6. For two-dimensional cases, only the magnetic vector potential $\mathbf{A}$ from (4) is necessary. According to Eq. (6), $\mathbf{A}$ is solved on the base-mesh, which represents Ω. In three dimensions, the electric scalar potential V from (5) serves as a

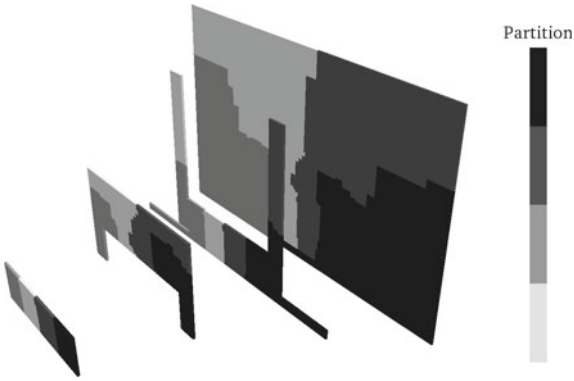


Fig. 5 Multi-mesh decomposition for parallel computations: each colour represents all parts of the base- and sub-meshes which belong to one partition (processor). Overlapping parts are held by the same processor. Bulk data mapping between different meshes does not involve parallel communication

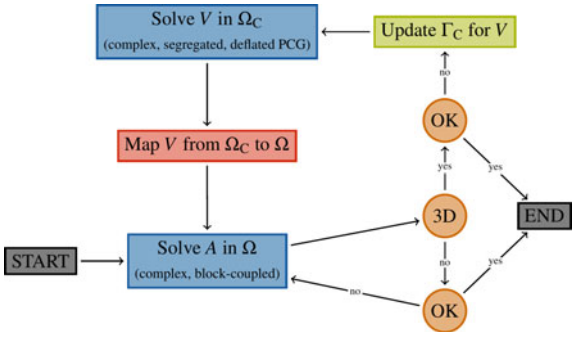


Fig. 6 *eddyCurrentFoam* algorithm: the round nodes with captions 'OK' and '3D' symbolise a convergence and dimensional check, respectively

tool for enforcing charge conservation (7) in the sub-mesh for conducting parts Ω_C. Both potentials are solved in their dedicated domain, while each solution is iteratively used to update explicit coupling terms in the respective other domain. The complex vector components of $\mathbf{A}$ in the quasi-static formulation are strongly coupled. This coupling is numerically handled by means of block matrices, as they are used for the coupled solution of $\mathbf{u}$ and p (cf. [7]). All details about *eddyCurrentFoam* will be published in a subsequent paper.

Fluid flow, surface tracking and fluid mesh motion are numerically solved on the basis of a modified *interTrackFoam*-solver algorithm, which was derived from the version originally published and implemented by Tukovic and Jasak [14]. Its sequence is sketched in Fig. 7. The small, upper right loop corresponds to an extended PISO-loop including surface tracking and the outer one represents the time loop including mesh update for Ω_F.

Embedding algorithms from both *eddyCurrentFoam* and *interTrackFoam* into an even broader multi-mesh framework constitutes our concept of coupling magneto-dynamics and hydrodynamics in the form of another solver called *interTrackEddyCurrentFoam*. Its algorithm is detailed in Fig. 8. On the one hand, the upper left loop basically shows an extended version of *interTrackFoam*, where the mesh is additionally adjusted in the buffer region Ω_B. On the other hand, the lower right loop

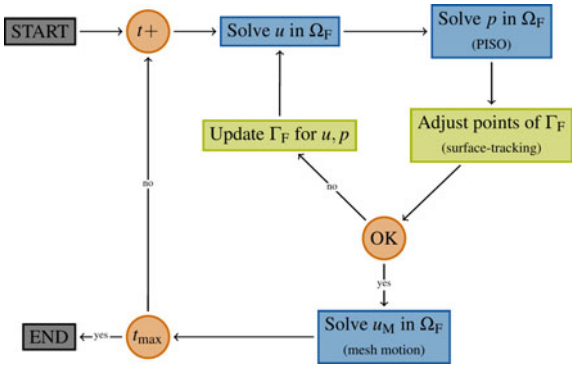


Fig. 7 *interTrackFoam* algorithm: the round nodes containing '$t+$', '$t_{\max}$' and 'OK' symbolise marching in time, a maximum-time check and a convergence check, respectively

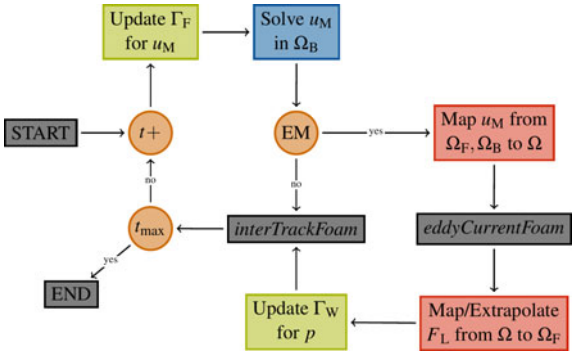


Fig. 8 *interTrackEddyCurrentFoam* algorithm: the round nodes showing '$t+$', '$t_{\max}$' and 'EM' symbolise marching in time, a maximum-time check and a check as to whether a magnetic update is due. For the meaning of the two nodes labelled '*interTrackFoam*' and '*eddyCurrentFoam*' the reader is referred to Fig. 7 and Fig. 6

comes into play if an electromagnetic update is due. As we have mentioned before, a quasi-static form of Maxwell's equations is being used. The Lorentz force $\langle \mathbf{F}_\mathrm{L} \rangle_t$, once calculated for a certain fluid time, is kept and transported with the dynamic mesh until its quality is no longer justifiable. Subsequently, a re-calculation of the electromagnetic problem is triggered. Prior to that, the current sub-mesh topologies of Ω_F and Ω_B need to be mapped onto the base-mesh (Ω) and onto the sub-mesh of the conducting region Ω_C.

5.3 *Improved Surface-Tracking Method*

As already mentioned in Sect. 3, the core surface-tracking algorithm of our model corresponds to Tukovic and Jasak [14]. Concerning one particular algorithm, the adjustment of the mesh points based on the current normal velocity at the free-surface Γ_F, it is therein referred to Muzaferija and Peric [10]. Following these references, the position of the free-surface points is updated in two steps using additionally introduced control points: firstly, the control points are shifted according to the current face-normal velocity (flux). Secondly, the new location of interface mesh points is

reconstructed by means of least-squares planes [14] or linear interpolation [10]. Due to the intentionally local design, both cited reconstruction methods are prone to a numerical instability mechanism, which will be being explained in the following.

For induction processing applications, it is not uncommon that a relatively high tangential flow velocity may occur close to the free-surface. In numerical simulations of such cases, we have repeatedly observed a strong tendency for the results to produce a creased surface shape in regions of high tangential velocity gradients.

We propose the following mechanism as a possible explanation: Small numerical errors in approximating the tangential gradient of the velocity may give rise to an initial evolution of local Kelvin–Helmholtz vortices. This may shift the control points alternatively in opposite directions from face to face along Γ_F. Normally, the resulting curvature and corresponding surface tension would oppose this effect immediately. But due to a lack of the above-mentioned (local) reconstruction methods for representing the subsequent curvature, this counter-force fails to appear. Figure 9 tries to visualise this destabilising principle for an idealised straight surface shape. It is obvious that the depicted part of the interface would remain flat by evenly shifting the control points further away from the surface in an alternate manner. Regarding the linear approximation, this checkerboard effect is reminiscent of the checkerboard pressure addressed by the interpolation of Rhie and Chow [11].

To affirm our assumption, we have successfully improved the surface-tracking method proposed by Tukovic and Jasak [14], so that a triangulated sub-mesh now serves to calculate the curvature. The triangulation of free-surface faces is done using mesh and control points, as displayed in Fig. 10. With this small change, the creasing disappeared and the overall numerical stability was greatly enhanced. In some simulated test cases, the number of pressure–velocity iterations could even be reduced by a factor of two.

Finally, we found that for the solution of the Laplacian (13) in Ω_F, a deformation in relation to the original mesh geometry (instead of an incremental one) helps in maintaining the mesh quality in the long term.

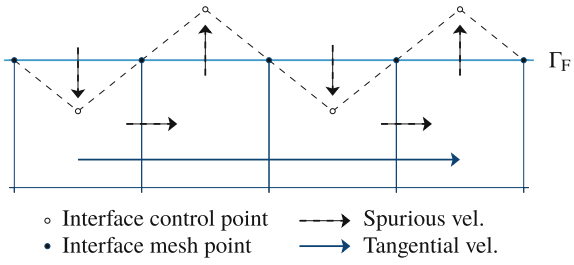


Fig. 9 Possible numerical instability mechanism in presence of a tangential velocity: evenly shifting the control points of the free-surface Γ_F further away from the flat interface in an alternate manner does not change the discretised shape. Piecewise linear reconstruction fails to represent the corresponding curvature

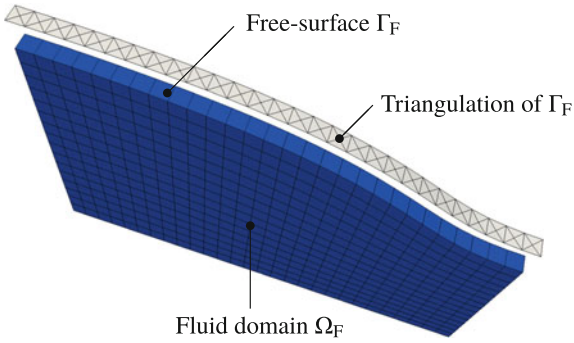


Fig. 10 Triangulated sub-mesh: for the calculation of the surface curvature, we use a triangulated mesh based on the original surface mesh points in combination with the control points (shifted face-centroids)

6 Application Examples

Since a comprehensive presentation of results would go far beyond the scope of this paper, we provide only two examples as demonstration of the capabilities of our development.

The first exemplary case is a simplified three-dimensional RGS process (cf. [3]) with a lower external source current density. A typical snapshot of the instantaneous fluid velocity magnitude (vectors) and time-averaged Lorentz force density (surface plot) is illustrated in Fig. 11. It represents a view towards one half of a central cut through the core process assembly. The latter consists of a casting frame, liquid silicon as fluid, a substrate and a rectangular excitation coil. Only the conducting region Ω_C and the fluid region Ω_F (as part of the former) are visible from the base-mesh. Based on these results, we were able to gain insight into global flow structures, quantify the surface deformation and capture the dynamic behaviour of it. In this case, the entire mesh has the shape of a box and consists of roughly 1×10^6 cells. With the help of turbulence modelling, the simulation can be run using only one CPU of a modern PC. More detailed results regarding RGS can be found in Beckstein et al. [2, 3].

As a second example, we have selected a simulation of an electromagnetically levitated (EML) drop of silicon. The simulation and its parameters comply with the experimental setup from [6]. Figure 12 shows the levitated drop centred in a coil arrangement, which is modelled as five independent, toroidal loops. Due to the more complex coil geometry and the large relative motion of the drop, this simulation was realised using a reduced (induced) magnetic vector potential [4], where the impressed (source) magnetic vector potential is calculated on the basis of a very efficient integral usage of Biot–Savart's law. The base-mesh is spherically shaped and also contains approximately 1×10^6 cells. In this case, we have used sub-grid-scale modelling for the hydrodynamic part.

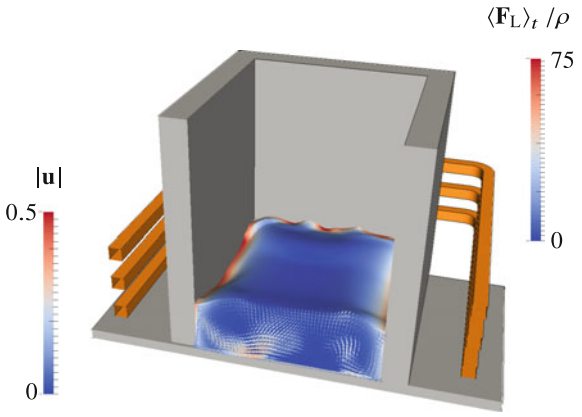


Fig. 11 Simplified 3D-RGS model: liquid silicon is being held between a graphite casting frame and substrate. A large excitation coil introduces heat and drives a turbulent flow. The instantaneous amplitude of the flow velocity $|\mathbf{u}|$ is shown as vectors in a central cutting plane. The contour plot demonstrates the skin effect and the dependency of the time-averaged Lorentz force $\langle \mathbf{F}_\mathrm{L} \rangle_t$ on the current surface shape

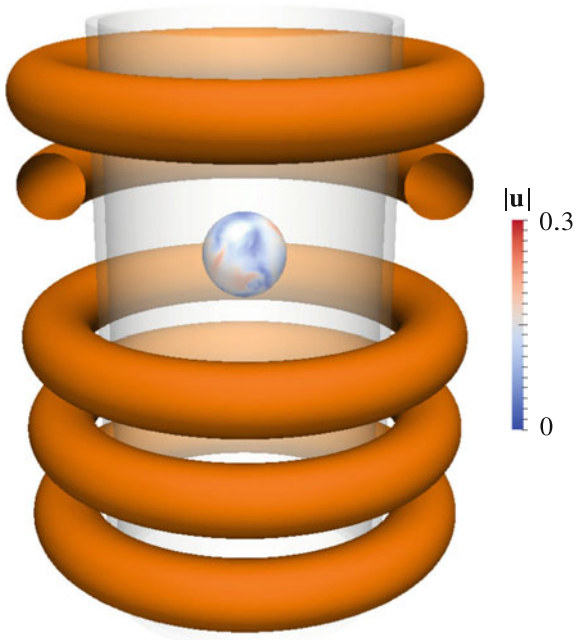


Fig. 12 Electromagnetic levitation (EML): a drop of liquid silicon, enclosed in a non-conducting glass cylinder, is floating in the centre of a modelled coil (one loop is cut to provide a clear view inside). The electric current in the lower three windings flows in the opposite direction of the upper two. The surface contour of the drop shows the (turbulent) instantaneous amplitude of the flow velocity $|\mathbf{u}|$

7 Conclusion

A novel multi-physics method has been implemented in *OpenFOAM®* for the simulation of free-surface flows under the influence of strong, alternating magnetic fields. Magnetodynamic and hydrodynamic effects are addressed in one single framework called *interTrackEddyCurrentFoam*. The formulation of the underlying multi-mesh description is versatile, fully parallelised and may be used for other multi-physics applications, too. Development of all presented elements is, however, not yet completely finished and optimised. In the whole context, a new solver for eddy-current problems called *eddyCurrentFoam* was developed and validated on the basis of the FVM. The existing implementation of the surface-tracking algorithm of *interTrackFoam* has been improved and extended. Even though an interface-tracking method is less flexible than interface-capturing methods, it allows us to reduce the amount of recurring electromagnetic calculations. The downside of this is that topological changes are not supported. Another limitation, which is visually outlined, e.g. in Jasak and Tukovic [8, Figs. 5–8], arises from the dynamic mesh. A moving mesh may suffer from degradation of its quality over time. This is especially true if large deformations are involved. Using adaptive re-meshing techniques is, however, quite challenging in combination with our multi-mesh approach. Therefore, our focus currently lies in finding suitable strategies for mesh-quality preservation of long simulation runs, stability improvements and an extensive validation based on numerical and experimental data.

Acknowledgements Financial support for this research from the German Helmholtz Association in frame of the Alliance Liquid Metal Technologies (LIMTECH) is gratefully acknowledged. We also would like to thank all the teachers and students involved in the NUMAP-FOAM Summer School 2015 at the Faculty of Mechanical Engineering and Naval Architecture (FAMENA) in Zagreb for numerous fruitful discussions, shared code and helpful comments.

References

1. Aruliah DA, Ascher UM, Haber E, Oldenburg D (2001) A method for the forward modelling of 3-d electromagnetic quasi-static problems. Mathematical Models and Methods in Applied Sciences 11(1):1–21
2. Beckstein P, Galindo V, Gerbeth G (2015a) Electromagnetic flow control in the ribbon growth on substrate (RGS) process. Magnetohydrodynamics 51(2):385–396
3. Beckstein P, Galindo V, Gerbeth G (2015b) Free-surface dynamics in the ribbon growth on substrate (RGS) process. In: Proceedings of the 8th International Conference on Electromagnetic Processing of Materials, Cannes, France, pp 167–170
4. Biro O, Preis K (2000) An edge finite element eddy current formulation using a reduced magnetic and a current vector potential. IEEE Transactions on Magnetics 36(5):3128–3130
5. Djambazov G, Bojarevics V, Pericleous K, Croft N (2015) Finite volume solutions for electromagnetic induction processing. Applied Mathematical Modelling 39(16):4733–4745
6. Gao L, Shi Z, Li D, Yang Y, Zhang G, McLean A, Chattopadhyay K (2016) Dimensionless analysis and mathematical modeling of electromagnetic levitation (EML) of metals. Metallurgical and Materials Transactions B 47(1):67–75

7. Jareteg K, Vukcevic V, Jasak H (2014) pUCoupledFoam - An open source coupled incompressible pressure-velocity solver based on foam-extend. In: 9th OpenFOAM® Workshop
8. Jasak H, Tukovic Z (2007) Automatic mesh motion for the unstructured finite volume method. Transactions of FAMENA 30(2):1–18
9. Karypis G (2003) Multi-constraint mesh partitioning for contact/impact computations. In: Proceedings of the 2003 ACM/IEEE Conference on Supercomputing
10. Muzaferija S, Peric M (1997) Computation of free-surface flows using the finite-volume method and moving grids. Numerical Heat Transfer, Part B: Fundamentals 32(4):369–384
11. Rhie CM, Chow WL (1983) A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. AIAA Journal 21:1525–1532
12. Schönecker A, Geerligs LJ, Müller A (2004) Casting technologies for solar silicon wafers: Block casting and ribbon-growth-on substrate. Solid State Phenomena 95–96:149–158
13. Spitans S, Baake E, Nacke B, Jakovics A (2014) A numerical model for coupled free surface and liquid metal flow calculation in electromagnetic field. International Journal of Applied Electromagnetics and Mechanics 44:171–182
14. Tukovic Z, Jasak H (2012) A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow. Computers & Fluids 55:70 – 84
15. Weller H, Tabor G, Jasak H, Fureby C (1998) A tensorial approach to computational continuum mechanics using object oriented techniques. Computational Physics 12(6):620–631

GEN-FOAM: An OpenFOAM®-Based Multi-physics Solver for Nuclear Reactor Analysis

Carlo Fiorina

Abstract A multi-physics solver for nuclear reactor analysis, named GeN-Foam (Generalized Nuclear Foam), has been developed by the Laboratory for Reactor Physics and System Behavior at the EPFL and at the Paul Scherrer Institut (Switzerland). The developed solver couples: a multigroup neutron diffusion or SP3 sub-solver; a thermal-hydraulics sub-solver based on the standard k-ε turbulence model, but extended to coarse-mesh applications through the use of a porous medium approach for user-selected cell zones; a displacement-based thermal-mechanics sub-solver to evaluate thermal deformations of structures; and a finite-difference subscale fuel model that can be used in coarse-mesh simulations of the core to evaluate the local temperature profile in fuel and cladding. A first-order implicit Euler scheme with an adaptive time step is used for time integration, and the coupling between equations is semi-implicit, using the Picard iteration. Three different meshes are used for thermal-hydraulics, thermal-mechanics and neutron diffusion, and fields are projected between different meshes through a standard volume-averaging technique. GeN-Foam features a general applicability to pin- or plate-fuel, or homogeneous nuclear reactors. Its application in several cases of interest has shown stable numerical behavior, the possibility of obtaining reliable results for traditional reactor types, as well as the possibility of investigating non-conventional reactors, whose analysis cannot be easily carried out using nuclear legacy codes.

1 Introduction

Nuclear reactors are highly complex engineering systems normally consisting of a reactor core, where heat is produced via nuclear fission, and multiple cooling loops that transfer the heat to a conventional power production unit (typically based on Rankine or Joule-Bryton power production cycles). The reactor core is most typically composed of several thousand tightly packed pins, spheres or plates containing the

C. Fiorina (✉)
École Polytechnique Fédérale de Lausanne, LRS, 1015 Lausanne, Switzerland
e-mail: carlo.fiorina@epfl.ch

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®* ,
https://doi.org/10.1007/978-3-319-60846-4_16

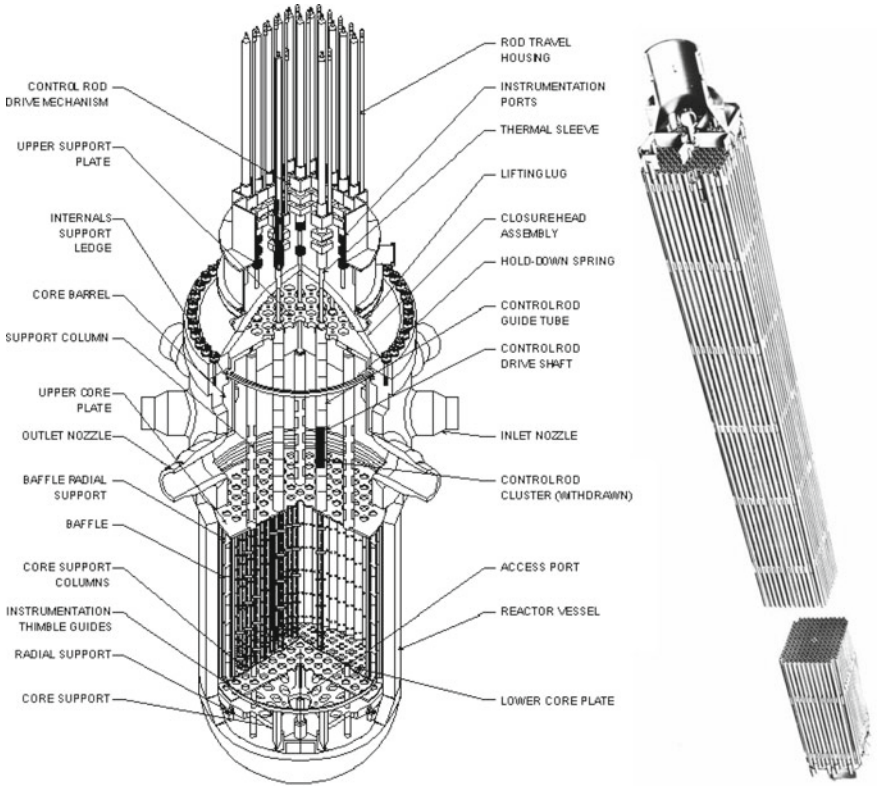


Fig. 1 Schematic representation of a PWR and of a fuel assembly (Photographs by United States Department of Energy and PD-USGov, distributed as public domain)

nuclear fuel (uranium, thorium and/or transuranic elements). These fuel elements operate at temperatures ranging from a few hundred to two thousand degrees Celsius and are typically cooled by water, helium or liquid metals. Figure 1 shows a schematic representation of a Pressurized Water Reactor (PWR), currently the most widely deployed nuclear reactor in the world [12].

The geometrical complexity of these systems adds to the high degree of complexity of the physical phenomena involved, including: the transport of neutrons; the interaction of these neutrons with matter and the consequent degradation of the materials in the core; the fuel burn-up; the highly complex chemical and thermomechanical behavior of the nuclear fuel; the thermomechanical behavior of the core-supporting structures; the complex one- or two-phase flow of the core coolant; the heat transfer from the fuel to the coolant; and the chemistry of the coolant and its interaction with the core structures.

When investigating a reactor's steady-state and transient behavior, some of these phenomena, like the evolution of the fuel and structural materials, can be neglected,

thanks to the longer time scales they act upon. These analyses can then be restricted to the neutron transport, the thermal-hydraulics of the fuel and coolant, and, for certain reactors, the thermal deformations of the core structures. Despite this simplification, the complexity and specificity of the phenomena involved have led the nuclear industry towards the development of quite unique numerical methodologies and computer codes. The core's thermal-hydraulics is generally investigated with 1-D codes or so-called sub-channel codes based on a very coarse radial discretization [2]. Neutron transport is most typically investigated using the so-called diffusion theory, which assimilates the behavior of neutrons to that of a gas and simulates it using a set of diffusion-reaction partial differential equations. These legacy codes take advantage of the repetitive multi-scale geometry of most nuclear reactors and operate on selected fixed geometries, which allows for very efficient solution algorithms based on structured meshes.

In recent years, the need has emerged for simulation tools that are both more accurate and more flexible. This follows from (1) the reliability needed for life extension of operating nuclear reactors; and (2) the design of some innovative nuclear reactors for which legacy codes cannot (easily) be employed. Several efforts have been undertaken in recent years for the modernization of the numerical toolset available in the nuclear engineering community. This can be witnessed in such important initiatives as the CASL project [3] in the US and the NURESAFE project [13] in Europe. Some initiatives have also been undertaken to aim at a paradigm shift in the development of nuclear codes, the main idea being a code development based on modern numerical libraries and methodologies that disregards legacy codes. Two main examples in this sense are the MOOSE project in the US [11] and the several initiatives dedicated to the use of OpenFOAM®. In this framework, the Laboratory for Reactor Physics and Systems Behaviour at PSI and EPFL has started developing new tools for reactor analysis and has decided to privilege the use of OpenFOAM®, mainly due to the available CFD solvers and to its open-source philosophy, which is expected to promote collaborative work and support educational efforts. One main result of this research work has been the development of a multi-physics tool for steady-state and transient analysis of nuclear reactors named GeN-Foam [6–8, 10], which represents the main subject of this chapter.

2 The GeN-Foam Multi-physics Solver

The GeN-Foam multi-physics solver is built upon four main components, namely: a neutron transport sub-solver; a thermal-mechanics sub-solver; a thermal-hydraulics sub-solver; and a 1-D sub-solver that evaluates temperatures in the fuel.

2.1 Neutron Transport

Power generation in a nuclear reactor is based on the energy released by fission of heavy metals, most typically, uranium and plutonium. The fission process is triggered by absorption of neutrons in these heavy metals, and is accompanied by the release of other neutrons, which allows for a self-sustaining fission chain. A prediction of density and energy distribution of neutrons is thus essential for predicting the spatial power distribution and its evolution over time.

Neutron behavior is governed by a Boltzmann transport equation [17]. Such an equation can be solved without major approximations by using stochastic Monte Carlo methodologies [17]. However, these methodologies have been prohibitive in the past given their computational requirements, and even nowadays, they can still hardly be used, e.g., for transient analyses. Many deterministic approximations of the Boltzmann equation have thus been proposed [17], the most common being the diffusion approximation. In such an approximation, neutrons are treated as a gas with isotropic neutron energy. A slightly more complex approximation is the so-called SP3 model (simplified third-order spherical harmonics), which consists of an expansion of the angular dependence of the neutron flux in spherical harmonics. The term "simplified" refers to the fact that the expansion is performed in 1-D, with extension to three dimensions obtained by simple substitution of derivatives with the corresponding 3-D operators. Both the diffusion and the SP3 model predict the neutron behavior in terms of neutron flux and typically rely on the subdivision of neutrons into energy groups. From an implementation perspective, an interesting aspect of the SP3 model is that the commonly used diffusion approximation can be derived as a special case in which only the first moment of the flux expansion is not zero. For this reason, implementation of the SP3 model was selected for GeN-Foam [10].

In the SP3 approximation, four equations can be obtained for the four flux moments φ_0 to φ_3, and substitution of odd moment into even moment equations then allows us to reduce the number of equations to be solved to 2. One additional approximation is typically included by considering the scattering of neutrons between energy groups as isotropic, thus obtaining the so-called within-group form of the SP3 equations. In addition, a new variable is typically defined as

$$\hat{\varphi}_0 = \varphi_0 + 2\varphi_2, \tag{1}$$

which allows us to obtain, for each energy group i, two equations in the form

$$\frac{1}{v_i}\frac{\partial \hat{\varphi}_{0,i}}{\partial t} = \nabla \cdot D_i \nabla \hat{\varphi}_{0,i} + \frac{\upsilon \sum_{f,i} (1-\beta_t)\chi_{p,i}}{k_{\text{eff}}} \hat{\varphi}_{0,i} - \sum_{r,i} \hat{\varphi}_{0,i} + \frac{S_{n,i}(1-\beta_t)\chi_{p,i}}{k_{\text{eff}}} + S_d \chi_{d,i} + S_{s,i} + 2\sum_{r,i} \varphi_{2,i} + 2\frac{1}{v_i}\frac{\partial \varphi_{2,i}}{\partial t}, \tag{2}$$

$$\frac{3}{v_i}\frac{\partial \varphi_{2,i}}{\partial t} = \frac{3}{7}\nabla \cdot \frac{1}{\Sigma_{t,i}} \nabla \varphi_{2,i} - \left(\frac{5}{3}\Sigma_t + \frac{4}{3}\Sigma_{r,i}\right)\varphi_{2,i} + \frac{2}{3} +_{r,i} \hat{\varphi}_{0,i} - \frac{2}{3}\frac{\upsilon \Sigma_{f,i}(1-\beta_t)\chi_{p,i}}{k_{\text{eff}}}\hat{\varphi}_{0,i} - \frac{2}{3}\frac{S_{n,i}(1-\beta_t)\chi_{p,i}}{k_{\text{eff}}} - \frac{2}{3}S_d \chi_{d,i} - \frac{2}{3}S_{s,i} + \frac{2}{3}\frac{1}{v_i}\frac{\partial \hat{\varphi}_{0,i}}{\partial t}, \tag{3}$$

where $S_{n,i}$, $S_{d,i}$ and $S_{s,i}$ are the neutron sources from fission, delayed neutron precursors, and scattering from other energy groups, and the standard notations for nuclear data are used (see, e.g., [10, 17]). The number of energy groups is arbitrary and can be set by the user depending on the specific application.

Discretization and solution of the equations are achieved based on the standard finite-volume discretization methodology provided by OpenFOAM® (i.e., Gauss discretization for Laplacians), normally choosing a harmonic interpolation for the diffusion coefficient at the cell faces. The sub-solver is designed both for time-dependent and eigenvalue (criticality) calculations. In the case of eigenvalue calculations, a traditional power iteration algorithm [17] is used. In the case of transient calculations, the equations for the different energy groups are solved sequentially and the explicit terms are resolved at each time step via the Picard iteration. Control rod movement is dealt with by the code via a modification of the nuclear data in selected cells. The possibility is included of deforming the mesh based on a given displacement field to account for the thermal deformation of a core, which allows for an accurate prediction of expansion-related reactivity feedbacks [6, 7]. Further details about the neutron transport sub-solver can be found in Fiorina et al. [7, 8, 10], including cross-sectional preparation and parametrization, use of discontinuity factors, acceleration techniques for transient analysis, and boundary conditions.

2.2 Thermal-Mechanics

The thermal-mechanic solver named solidDisplacementFoam available in the OpenFOAM® 4.1 version has been used as the basis for the solution of the thermal-mechanics equations. The possibility has been included of using meshes subdivided into multiple cell zones, and associating a set of properties with each of them through an input file. The thermomechanic sub-solver can be used to evaluate the temperature-

induced deformation of the main core structures that support the reactor core. Axial deformation of the neutronic mesh in the active core (where the actual fuel is) is calculated independently, based on the fuel expansion coefficient and temperatures. Specifically, an equation is solved for the displacement D_f in the form

$$\boldsymbol{v_f} \cdot \nabla \cdot D_f = \alpha(T - T_{\text{ref}}), \tag{4}$$

where $\boldsymbol{v_f}$ is the user-selected axial orientation of the fuel and α its linear expansion coefficient. An equation like Eq. (4) is also used to determine the axial displacement of the control rod driveline. The three displacement fields are then combined to deform the neutronic mesh.

2.3 Thermal-Hydraulics

A main objective of GeN-Foam is to analyze the full core or the primary circuit of nuclear reactors. Standard turbulence modeling [e.g., Reynolds Averaged Navier–Stokes (RANS)] is currently too demanding for routine calculations at the core level. A standard engineering approach to investigating complex structures using a coarse mesh is to treat these structures as porous media, with dedicated models employed to simulate the interaction of the fluid with the subscale structures [18]. One advantage of this treatment is that the equations employed revert back to standard RANS equations in cases of clear fluid (i.e., fluid regions not treated as porous), so that the same set of equations can be discretized and solved on the same mesh while de facto treating different zones of the geometry with two different approaches (detailed RANS or a coarse-mesh porous medium). Equations for the turbulent single-phase flow of a fluid in a porous medium can be rigorously derived from standard Navier–Stokes equations via time and volume averages (see Vafai [18] for details). The resulting equations of mass, momentum, and energy conservation are [4, 7, 16]

$$\frac{\partial \gamma \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u_D}) = 0, \tag{5}$$

$$\frac{\partial \rho \boldsymbol{u_D}}{\partial t} + \frac{1}{\gamma} \nabla \cdot (\rho \boldsymbol{u_D} \otimes \boldsymbol{u_D}) = \nabla \cdot (\mu_T \nabla \boldsymbol{u}) - \gamma \nabla p + \gamma \boldsymbol{F_g} + \gamma \boldsymbol{F_{ss}} - (\rho \boldsymbol{u_D} \cdot \boldsymbol{u_D}) \nabla \frac{1}{\gamma}, \tag{6}$$

$$\frac{\partial \gamma \rho e}{\partial t} + \nabla \cdot (\boldsymbol{u_D}(\rho e + p)) = \gamma \nabla \cdot (k_T \nabla \mathrm{T}) + \boldsymbol{F_{ss}} \cdot \boldsymbol{u_D} + \gamma \dot{Q}_{ss}, \tag{7}$$

where the Darcy velocity $\boldsymbol{u_D}$ is defined as

$$\boldsymbol{u_D} \triangleq \gamma \boldsymbol{u}. \tag{8}$$

It can be observed that Eqs. (5)–(7) closely resemble the traditional RANS equations. A primary main difference is the appearance of a porosity term γ, which takes into account the fact that only part of the volume is occupied by the fluid. Moreover, two additional terms $\boldsymbol{F}_{ss}$ and $\dot{Q}_{ss}$ appear in the equations, representing the effect of the subscale structures on the fluid flow. In case of a clear fluid, these terms are equal to zero and the porosity equal to one, so that, as mentioned, Eqs. (5)–(7) revert back to the traditional RANS equations.

The last term in Eqs. (7) represents the Bernoulli effect. As a matter of fact, in most applications in nuclear engineering, the porosity does not change inside of a porous zone, but only at the interface between two different zones. The last term in Eq. (7) can thus be neglected (at least for low Mach number flows), provided its effect is properly taken into account at interfaces between different cell zones. This can be done via the introduction of a pressure baffle [14]. In fact, such a pressure baffle is necessary to take into account localized pressure drops at the entrance of a porous zone. An alternative approach is to include a thin buffer region at the interface between two different porous zones and to define the term $\boldsymbol{F}_{ss}$ there so that it models both the concentrated pressure drops and the Bernoulli effect.

The terms $\boldsymbol{F}_{ss}$ and Q_{ss} can be obtained from experiments or by making use of standard correlations for pressure drops and heat transfer coefficients. The terms μ_T and k_T in Eqs. (7) and (8) are modeled in a clear fluid zone using the standard k-ε model. For porous zones, the problem of predicting k and ε becomes extremely complex. As a matter of fact, pressure drops and heat transfer in porous zones are typically dominated by interaction with the subscale structure, so that an accurate modeling of μ_T and k_T becomes inessential. On the other hand, it is relatively important to predict a correct value of k and ε at the entrance of a clear fluid zone, which asks for a good prediction of k and ε at the outlet of the porous zone where the fluid comes from. Following these considerations, GeN-Foam avoids solving the standard k-ε model inside the porous zones but instead forces the values of k and ε to converge to user-selected values [7]. A special treatment is also required for wall treatment in porous regions. In particular, a zero gradient can generally be assumed for velocity, but appropriate heat transfer conditions have to be set. This has been achieved via a modified thermal baffle that employs available heat transfer correlations. Some details can be found in Radman et al. [15].

2.4 Fuel Temperatures

When performing a coarse-mesh analysis of a reactor core, it is necessary to use a dedicated model so as to accurately predict the temperature distribution in the fuel, which, in turn, is essential for the correct prediction of the heat transfer between the fuel and coolant, as well as the interaction of the neutrons with the fuel. To this purpose, GeN-Foam solves the exact form of the equations depending on the actual shape of the fuel (spheres, plates or pins) for a 1-D heat conduction problem. The solution is achieved via a second-order finite-difference spatial discretization with

implicit Euler time stepping. The coupling with the solution for coolant temperatures is achieved by calculating the heat flux on the surface of fuel elements and by transferring it to the coolant in the form of a volumetric heat source via multiplication with the fuel volumetric area. Conversely, the coolant temperature and the heat transfer coefficient between coolant and fuel elements are used to set up a convective boundary condition for the 1-D conduction problem. Details can be found in Fiorina et al. [7].

2.5 Coupling Strategy

The four sub-solvers described above need to be tightly coupled for an accurate solution. The fields calculated on one mesh but needed on a second mesh are projected based on the standard OpenFOAM® cell-volume-weighted algorithm. The coupling strategy is depicted in Fig. 2. The initial time step is given by the user, while subsequent time steps are selected based on the most stringent requirement between Courant number condition and a maximum power variation at each time step (default 2.5%). After selection of the time step, the solver follows the structure of a standard PIMPLE loop for solving velocity, pressure, and energy. After achieving the coolant flow and temperature fields, GeN-Foam solves for the temperatures in the subscale structures, in the fuel, and again for the energy equation. It then goes through the thermal-mechanic sub-solver and the neutronic sub-solver, each of which includes its own Picard iterations to solve for the explicit terms. Iteration between pressure-velocity-energy solution and the other equations is typically not necessary, and such a possibility has not been shown in Fig. 2. However, the possibility of a user-selected number of "outer iteration" steps is included in GeN-Foam.

3 Discussion and Conclusions

In the last few years, GeN-Foam has been applied to the analysis of a number of different reactor concepts, which allows us to draw some general conclusions on its performance and capabilities [9]. In view of its unique flexibility, GeN-Foam has been used most frequently for the analysis of nontraditional reactors, whose features often challenge the use of nuclear legacy codes.

The use of unstructured meshes and non-problem-specific finite-volume discretization schemes for neutronics tends to significantly increase the required computational resources compared to legacy codes that employ structured meshes and purpose-specific methodologies [10]. On the other hand, use of unstructured meshes makes possible the investigation of experimental or innovative reactors with unconventional geometries. As an example, Fig. 3 shows the thermal neutron flux distribution for the experimental Argonaut reactor at the University of Florida. In addition,

the possibility of parallelizing the computation generally allows for an acceptable computing time.

The possibility of deforming the mesh based on a detailed displacement field provides notably improved predictions of a reactivity feedback coefficient in fast-spectrum reactors. Such a capability has been used in [6] to predict the axial fuel

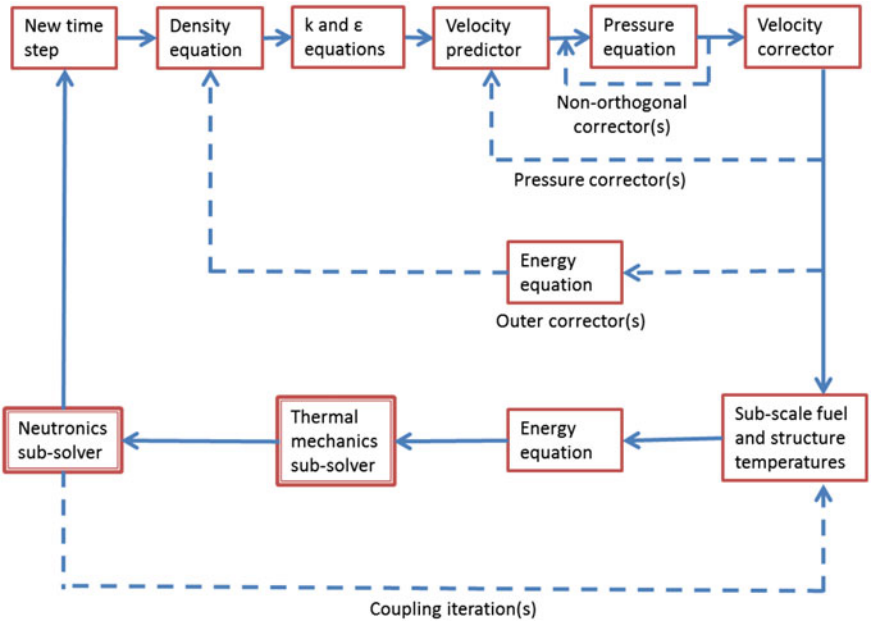


Fig. 2 GeN-Foam coupling strategy [7]

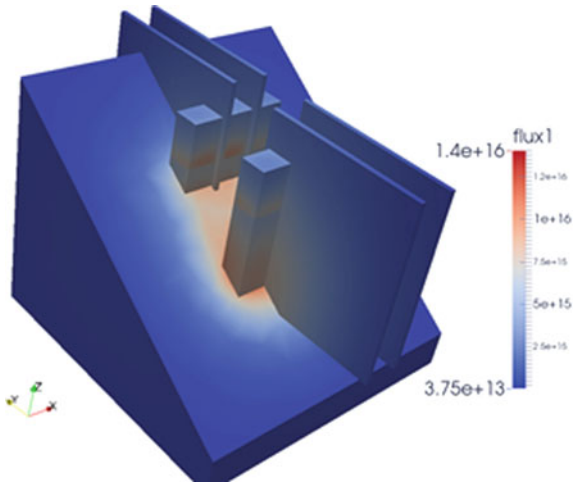


Fig. 3 Thermal neutron flux distribution for the Argonaut reactor at the University of Florida

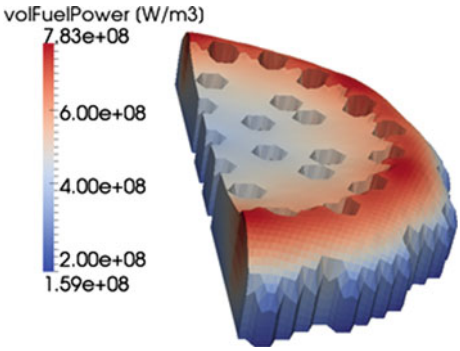


Fig. 4 Fuel volumetric power in the core region of the European Sodium Fast Reactor [6]. Deformation is magnified by 100 times

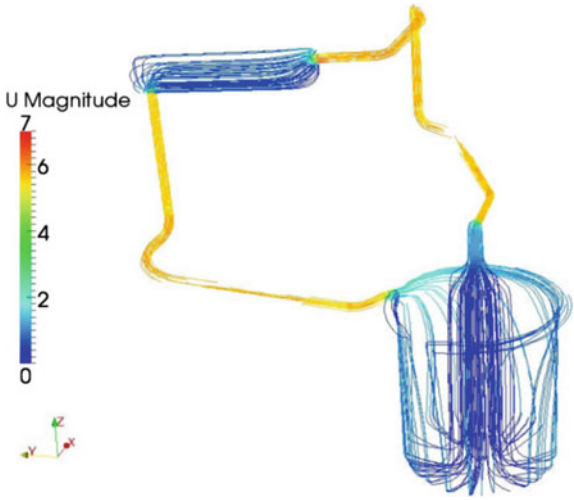


Fig. 5 Coarse-mesh simulation of the Molten Salt Reactor Experiment [1]

expansion coefficient for the European Sodium Fast Reactor (Fig. 4), showing that a 50% under-prediction is obtained via the frequently employed evaluation based on a uniform core axial expansion.

Use of a mixed coarse/fine-mesh approach with a porous medium approximation for thermal-hydraulics is still a field of active research and would require further testing and development. However, the work performed to date has highlighted the actual possibility of achieving full core or full primary circuit solutions in a reasonable time frame. Figure 5 shows an example of a coarse-mesh solution for the Molten Salt Reactor Experiment [1]. Drawbacks are in the preparation of the mesh, which can be time-consuming (compared to 1-D system codes) and requires good expertise. In addition, the use of coarse-mesh approaches defies the definition of a mesh-independent solution and introduces some small degrees of user discretion into the results.

In general, the use of a single environment for preprocessing, computing and postprocessing the different "physics" involved in the analysis of nuclear reactors

proved to be extremely convenient if compared, e.g., to the external coupling of independent codes. On the other hand, using a code like GeN-Foam requires previous knowledge about OpenFOAM®, and more generally, about CFD calculations and pre- and postprocessing techniques involving unstructured meshes. Such complexity has been shown to be a significant challenge for users.

References

1. Bao J (2016) Development of the model for the multi-physics analysis of Molten Salt Reactor Experiment using GeN-Foam code. Master Thesis, Paul Scherrer Institut, Switzerland.
2. Bestion D (2008) System Code Models and Capabilities, In: THICKET 2008, Pisa, Italy, May 5–9.
3. CASL, Consortium for Advanced Simulation of LWRs (2011) A Project Summary, CASL-2010-0022-001-P. U.S. Department of Energy, Nuclear Energy. www.casl.gov (Published: 07/18/2011).
4. Clifford I (2013) A hybrid coarse and fine mesh solution method for prismatic high temperature gas-cooled reactor thermal-fluid analysis. PhD Thesis. PennState, US.
5. Encyclopedia Britannica, https://www.britannica.com/technology/nuclear-reactor/images-videos.
6. Fiorina C, Mikityuk K. (2015a) Application of the new GeN-Foam multi-physics solver to the European Sodium Fast Reactor and verification against available codes, In: proceedings of the ICAPP 2015 Conference, Nice, France, May 03–06.
7. Fiorina C, Clifford I D, Aufiero M, Mikityuk K (2015b) GeN-Foam: a novel OpenFOAM® based multi-physics solver for 2D/3D transient analysis of nuclear reactors, Nuclear Engineering and Design 294:24–37.
8. Fiorina C, Kerkar N, Mikityuk K, Pautz A (2016) Development and verification of the neutron diffusion solver for the GeN-Foam multi-physics platform, Annals of Nuclear Energy 96:212–222.
9. Fiorina C, Pautz A (2016) Status and ongoing developments for the GeN-Foam multi-physics solver, In: proceedings of the 2016 ANS Winter meeting, November 6–10, Las Vegas, US.
10. Fiorina C, Hursin M, Pautz A (2017) Extension of the GeN-Foam neutronic solver to SP3 analysis and application to the CROCUS experimental reactor, Annals of Nuclear Energy 101:419–428.
11. Gaston D, Newman C, Hansen G, Lebrun-Grandié D (2009) MOOSE: A parallel computational framework for coupled systems of nonlinear equations, Nuclear Engineering and Design 239:1768–1778.
12. IAEA (2017) https://www.iaea.org/PRIS/WorldStatistics/OperationalReactorsByType.aspx.
13. NURESAFE, NUclear REactor SAFEty Simulation Platform (2013). Project presentation. http://www.nuresafe.eu/.
14. OpenFOAM® pressure baffle (2017) https://github.com/OpenFOAM/OpenFOAM-4.x/blob/master/src/TurbulenceModels/turbulenceModels/derivedFvPatchFields/porousBafflePressure/porousBafflePressureFvPatchField.H.
15. Radman S, Fiorina C, Pautz A (2017) Investigation of partial coolant flow blockage in a Sodium Fast Reactor assembly with coarse-mesh methodologies, In: NENE 2017, September 11–14, Bled, Slovenia.
16. Saurel R, Abgrall R (1999) A Multiphase Godunov Method for Compressible Multifluid and Multiphase Flows, Journal of Computational Physics 150:425–467.
17. Stacey W M (2007) Nuclear Reactor Physics. WILEY-VCH Verlag Gmbh & Co. KGaA.
18. Vafai K, (2005) Handbook of Porous Media, CRC Press, Taylor & Francis, New York.

Harmonic Balance Method for Turbomachinery Applications

Gregor Cvijetić and Hrvoje Jasak

Abstract The Harmonic Balance Method for nonlinear periodic flows is presented in this paper. Assuming a temporally periodic flow, a Fourier transformation is deployed in order to formulate a transient problem as a multiple quasi-steady-state problem. A solution of the obtained equations yields flow fields at discrete instants of time throughout a representative harmonic period, while still capturing the transient effect. The method is implemented in foam-extend, a community-driven fork of OpenFOAM® and developed for multi-frequential use in turbomachinery applications. For validation, a 2D turbomachinery test case is used. Pump head, efficiency, and torque obtained with Harmonic Balance will be compared to a transient and steady-state simulation. Furthermore, pressure contours on rotor blades will be compared. And finally, in order to present the method's efficiency along with its accuracy, a CPU time comparison will also be presented.

Nomenclature

$\mathscr{Q}$	Dimensionless passive scalar in the time domain
$\mathscr{R}$	Convection–diffusion transport operator for a passive scalar in the time domain
t	Time, s
$\mathbf{u}$	Velocity field, m/s
γ	Diffusion coefficient, m^2/s
$S_{\mathscr{Q}}$	Source terms for a passive scalar, 1/s
ω	Base radian frequency, rad/s
$\underline{\underline{A}}$	Discrete Fourier expansion matrix
$\underline{Q}$	Vector of Fourier harmonics for $\mathscr{Q}$

G. Cvijetić (✉) · H. Jasak
Department of Energy, Power Engineering and Environment,
Ivana Lučića 5, 10000 Zagreb, Croatia
e-mail: gregor.cvijetic@gmail.com

H. Jasak
e-mail: hrvoje.jasak@fsb.hr

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_17

$\underline{R}$	Vector of Fourier harmonics for $\mathscr{R}$
$\underline{\mathscr{Q}}$	Vector of discrete time instant values for $\mathscr{Q}$
$\underline{\mathscr{R}}$	Vector of discrete time instant values for $\mathscr{R}$
T	Base period, s
$\underline{\underline{E}}$	Forward DFT matrix
$\underline{\underline{E}}^{-1}$	Backward (inverse) DFT matrix
$\underline{\underline{P}}_{i-j}$	Coupling coefficient for t_i and t_j time instants
P_l	Coupling coefficient equivalent to P_{i-j}
ν	Kinematic viscosity, m^2/s
ρ	Density, kg/m^3
p	Pressure, Pa
f	Base frequency, Hz
A, B	Wave amplitudes
ϕ	Phase shift, s

Subscripts

S	Sine part
C	Cosine part
i	Harmonic index
t_j	Discrete time instant

1 Introduction

In order to accurately model unsteady flows while still maintaining a reasonable computational cost, the Harmonic Balance method is developed. Instead of using an original, transient mathematical model, one solves a specified number of coupled steady-state problems, each representing an equation for a unique time instant. The transformation of the mathematical model is achieved through a Fourier transform of unknown fields, in which the accuracy of the resulting model is controlled by a specified number of harmonics. With an increasing number of harmonics, higher order flow effects are captured. As opposed to conventional steady-state methods, the benefit of Harmonic Balance is the ability to capture transient flow features, but at a cost of longer CPU time. However, compared to conventional transient simulations, Harmonic Balance offers a significant CPU time reduction [1] with comparable accuracy.

Harmonic Balance was initially developed as a periodic boundary condition by He [2]. He and Ning [3] extended its application to solving the two-dimensional Navier–Stokes equations and presented the efficiency improvement compared to nonlinear time-marching methods. Recently, the Harmonic Balance method has been extensively developed in numerous areas of application. Oscillating airfoils were presented

by Thomas et al. [4], and Dufour et al. presented the 3D oscillating wings [5]. Limit cycle oscillations [6] were the topic of Harmonic Balance research by Ekici et al. Hall et al. [7] used complex geometries such as turbines to extend and demonstrate the Harmonic Balance capabilities. After the development of Harmonic Balance for turbomachinery, the need for multiple frequencies was stressed by Gopinath et al. [8]. In the case of multistage turbines, the rotor frequency changes in each stage due to a different number of blades. A similar multiple-frequency approach by Guédeney et al. [9] relies on a uniform time sampling of the longest period of interest, therefore reconstructing the flow field between stages to match the time instants. Additionally, Guédeney et al. [10] presented the two algorithms for nonuniform time sampling in a multiple-frequency approach of Harmonic Balance.

He [11] addressed the stability and convergence issues of the Harmonic Balance method. Huang and Ekici [12] introduced time spectral viscosity as an additional stabilization factor. The addition of time spectral viscosity eliminates aliasing errors and ensures convergence toward the physical solution. In order to avoid stability issues in more complex geometric configurations, as reported by Hall et al. [13], strategies for implementation of the implicit Harmonic Balance algorithm have been investigated [14]. Sicot et al. [15], Woodgate and Badcock [16], and Su and Yuan [17] have proposed different implicit techniques for the Harmonic Balance method that involves the development of new implicit algorithms. The more favorable approach that does not require the development of a new algorithm was presented by Thomas et al. [14] using a two-step approximate factorization approach. Antheaume and Corre [18] implemented an implicit Harmonic Balance method using numerical implicit strategies, block relaxation and matrix-free approaches.

The numerical model is based on a conventional, second-order accurate, polyhedral Finite Volume Method (FVM) in the open-source software foam-extend. The harmonic balance operator is implemented in generic form, allowing its use within the foam-extend toolkit on various equation sets without intervention in the nonlinear flux-source model. The coupling of temporal source terms was initially implemented in a segregated manner and was then further developed for implicit calculation of time instants and source terms.

Validation of the Harmonic Balance method will be performed using a 2D turbomachinery test case. The Harmonic Balance method will be compared with conventional transient simulation and steady-state simulation with Multiple Reference Frames (MRF) accounting for rotor rotation. Global pump parameters will be compared: head, torque, and efficiency. In order to present the local accuracy of the Harmonic Balance method, pressure contours on rotor blades will be presented for different number of harmonics. To present the efficiency of the Harmonic Balance method, CPU time comparison will also be given.

2 Mathematical Model

An overview of the mathematical model of the Harmonic Balance (HB) method is presented in this section. HB treatment, transforming the time-derivative term into a source term, is done on incompressible flow equations. The mathematical model presented here is general and valid for any number of harmonics. A full derivation can be found in [1].

2.1 Passive Scalar Transport

The convection–diffusion equation for the passive scalar transport of scalar $\mathscr{Q}$ reads as

$$\frac{\partial \mathscr{Q}}{\partial t} + \mathscr{R} = 0\,, \tag{1}$$

where $\mathscr{R}$ stands for convection, diffusion, and source/sink terms:

$$\mathscr{R} = \nabla \cdot (\mathbf{u}\mathscr{Q}) - \nabla \cdot (\gamma \nabla \mathscr{Q}) - S_{\mathscr{Q}}\,, \tag{2}$$

$\mathbf{u}$ is the transport velocity and γ is diffusivity. Expanding $\mathscr{Q}$ into a Fourier series with n harmonics reads as

$$\mathscr{Q}(t) = Q_0 + \sum_{i=1}^{n} Q_{S_i} \sin(i\omega t) + Q_{C_i} \cos(i\omega t). \tag{3}$$

Scripture characters, $\mathscr{Q}$, are used to denote time-domain variables, while Q denotes the frequency domain field. The Fourier expansion for $\mathscr{R}$ is analogous to the one in Eq. (3), with Q replaced with R. Inserting Eq. (3) into the transport equation, Eq. (1), yields sine, cosine and mean terms. Grouping the terms gives $2n + 1$ equations: n for sine and cosine and 1 for the mean value. Thus, an HB scalar transport equation becomes a set of $2n + 1$ equations, written in matrix form as follows:

$$\omega \underline{\underline{A}}\, \underline{Q} + \underline{R} = \underline{0}\,, \tag{4}$$

where $\underline{\underline{A}}$ is a $(2n + 1) \times (2n + 1)$ coefficients matrix, $\underline{Q}$ and $\underline{R}$ are column matrices containing Fourier sine Q_{S_i}/R_{S_i} and cosine Q_{C_i}/R_{C_i} coefficients and ω is a base radian frequency.

Introducing the matrix representation of a Discrete Fourier Transform (DFT) for conversion from the time-domain vector $\underline{\mathscr{Q}}$ to the frequency domain vector $\underline{Q}$ yields

$$\underline{Q} = \underline{\underline{E}}\, \underline{\mathscr{Q}}\,, \tag{5}$$

where $\underline{\underline{E}}$ is a DFT matrix and $\underline{\mathscr{Q}}$ is a discrete time-domain vector needed for unique one-to-one mapping:

$$\underline{\mathscr{Q}}^T = \left[\mathscr{Q}_{t_1}\ \mathscr{Q}_{t_2}\ \mathscr{Q}_{t_3}\ \cdots\ \mathscr{Q}_{t_{2n+1}}\right], \tag{6}$$

where t_i stands for

$$t_i = \frac{iT}{2n+1}, \text{for } i = 1 \ldots 2n+1. \tag{7}$$

Multiplying Eq. (5) with $\underline{\underline{E}}^{-1}$ from the left, one obtains a mapping from the frequency domain to the time domain:

$$\underline{\mathscr{Q}} = \underline{\underline{E}}^{-1}\,\underline{Q}. \tag{8}$$

Using the DFT matrices $\underline{\underline{E}}$ and $\underline{\underline{E}}^{-1}$, the frequency domain scalar transport equation, Eq. (4), is formulated using the time-domain vector $\underline{\mathscr{Q}}$:

$$\omega \underline{\underline{A}}\,\underline{\underline{E}}\,\underline{\mathscr{Q}} + \underline{\underline{E}}\,\underline{\mathscr{R}} = \underline{0}, \tag{9}$$

where the same transformation has been applied to $\underline{R}$ and $\underline{Q}$. Even though equations could be solved in this form, evaluating sources and fluxes in the frequency domain is computationally expensive and inconvenient [13]. Therefore, the equation is transformed back to the time domain, multiplying Eq. (9) with $\underline{\underline{E}}^{-1}$ from the left:

$$\omega \underline{\underline{E}}^{-1}\,\underline{\underline{A}}\,\underline{\underline{E}}\,\underline{\mathscr{Q}} + \underline{\mathscr{R}} = \underline{0}. \tag{10}$$

The resulting equation represents a temporally coupled set of $2n+1$ steady-state problems. It can be noticed that $\mathscr{R}$ has been replaced with its discrete counterpart $\underline{\mathscr{R}}$, indicating that the solution is sought at a fixed number of discrete time instants only. The number of discrete time instants is defined with a specified number of harmonics n, as indicated in Eq. (6). The time-derivative term has been replaced by terms coupling the solutions at different time steps. This is equivalent to evaluating the time derivative of a harmonic signal via $2n+1$ uniformly spaced temporal snapshots, including a mean (steady) solution.

The expanded form of the coupled HB scalar transport equations may be written in a more convenient form:

$$\nabla \cdot (\mathbf{u}\mathscr{Q}_{t_j}) - \nabla \cdot \left(\gamma \nabla \mathscr{Q}_{t_j}\right) - S_{\mathscr{Q}_{t_j}} = -\frac{2\omega}{2n+1}\left(\sum_{i=1}^{2n+1} P_{i-j}\mathscr{Q}_{t_i}\right), \tag{11}$$
$$\text{for } j = 1 \ldots 2n+1,$$

where P_{i-j} is defined as

$$P_{i-j} = P_l = \sum_{k=1}^{n} k \sin(lk\omega\Delta t), \text{for } l = -2n \ldots 2n \tag{12}$$

and

$$\Delta t = \frac{T}{2n+1}. \tag{13}$$

The coupling of solutions at different time instants t_j is achieved through the P_{i-j} matrix, modeling the time-derivative term as additional source terms. Hence, a single transient equation given by Eq. (1) is transformed into a set of $2n+1$ coupled steady-state problems, Eq. (11). It should be noted that no less then $2n+1$ time instants should be solved, as that could cause aliasing errors. A set of $2n+1$ equations is the smallest required number of time instants for a chosen number of harmonics, n, for which an accurate solution can be obtained.

2.2 Incompressible Fluid Flow

An incompressible, turbulent, single-phase flow is modeled with the continuity and the momentum equation:

$$\nabla \cdot \mathbf{u} = 0\,, \tag{14}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) - \nabla \cdot (\nu \nabla \mathbf{u}) = -\frac{\nabla p}{\rho}\,, \tag{15}$$

where ν denotes kinematic viscosity, ρ fluid density, and p the pressure field.

As previously presented, HB treatment transforms the time derivative term into a set of temporally coupled source terms, leaving convection, diffusion, and additional source terms in their original form. The continuity equation remains the same, as it does not contain a time-derivative term:

$$\nabla \cdot \mathbf{u}_{t_j} = 0. \tag{16}$$

Equation (16) suggests that the incompressible continuity equation must hold in each time instant t_j, as expected. The HB form of the momentum equation reads as follows:

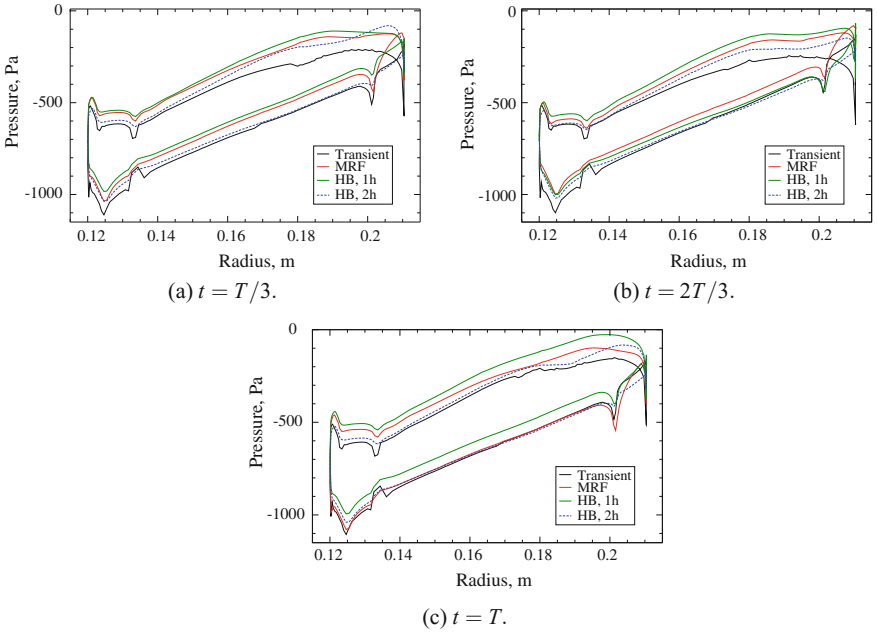
$$\nabla \cdot (\mathbf{u}_{t_j}\mathbf{u}_{t_j}) - \nabla \cdot \left(\nu \nabla \mathbf{u}_{t_j}\right) = -\nabla p_{t_j} - \tfrac{2\omega}{2n+1}\left(\textstyle\sum_{i=1}^{2n} P_{i-j}\mathbf{u}_{t_i}\right), \quad \text{for } j = 1 \ldots 2n+1. \tag{17}$$

Equations (16) and (17) represent $2n+1$ coupled pressure–velocity systems with enforced periodic behavior defined using the base frequency ω and number of harmonics n.

3 Results

The HB method is presented using a 2D ERCOFTAC centrifugal pump test case. Due to different number of blades in the rotor and the stator, a multiple-frequency approach is adopted in order to account for their separate frequencies. In this way, the part of the domain consisting of the rotor is solved using the frequency corresponding to that at which periodic instabilities occur in the rotor, while the part of the domain consisting of the stator is solved using the frequency corresponding to that at which periodic instabilities occur in the stator. For comparison, global pump parameters will be used and compared with a transient simulation and a steady-state simulation with Multiple Reference Frame (MRF). At the end, a CPU time comparison is presented.

Figure 1 shows the pressure contours on one rotor blade in time instants $t = T/3$, $t = 2T/3$, and $t = T$. The HB solution with 1 and 2 harmonics is compared with the MRF and transient solutions. The two harmonics solution agrees with the transient one more closely than the one harmonic solution. Compared to the transient simulation, the MRF contours are mainly between the one and two harmonics contours. Although the MRF pressure contours agree well with the transient one, in most cases the MRF cannot predict the local instabilities accurately, as the mesh is not moving. On the other hand, the transient simulation and HB deploy a moving mesh in order to account for rotation. In each time instant, HB is calculated using the corresponding mesh position for that time instant.



(a) $t = T/3$. (b) $t = 2T/3$.

(c) $t = T$.

Fig. 1 Pressure contours on the rotor blade at different time instants

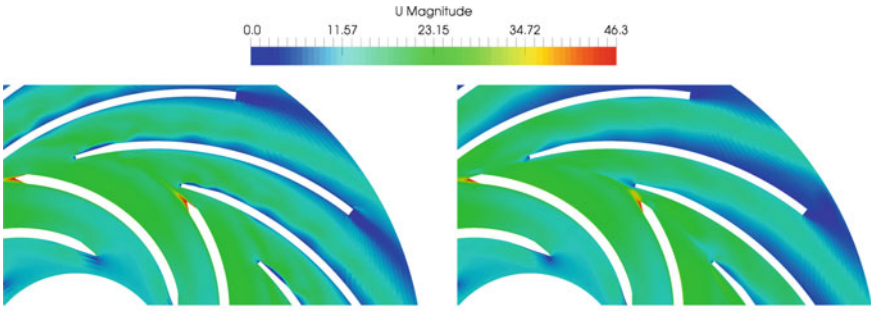
Table 1 Global pump parameters comparison

		Transient solver	HB, 1h	Error, %	HB, 2h	Error, %	MRF	Error, %
$t = \frac{T}{3}$	Efficiency	89.72	88.80	1.0	89.76	0.0	89.65	0.1
	Head	81.48	81.80	0.4	80.45	1.3	84.12	3.1
	Torque	0.0297	0.0302	1.7	0.0294	0.9	0.0308	3.6
$t = \frac{2T}{3}$	Efficiency	89.92	88.78	1.3	89.81	0.1	89.65	0.3
	Head	81.48	81.85	0.4	80.6	1.1	84.12	3.2
	Torque	0.0296	0.0302	2.0	0.0295	0.4	0.0308	4.1
$t = T$	Efficiency	89.83	88.85	1.1	89.71	0.1	89.65	0.2
	Head	81.49	81.79	0.4	80.39	1.3	84.12	3.2
	Torque	0.0297	0.0302	1.6	0.0294	1.0	0.0308	3.7

To ensure that the global prediction is accurate, in Table 1 the comparison of global pump parameters is presented. The Harmonic Balance solution with 1 and 2 harmonics, as well as the MRF, are compared to the transient simulation using pump head, efficiency and torque. For Harmonic Balance, the highest error is 1.7% for the one harmonics case and 1.3% for the two harmonics case, while for the MRF the highest error is 4.1%. This shows that HB with 1 harmonic can be used as an accurate tool for prediction of global pump parameters, while still obtaining transient effects. In HB simulation with 2 harmonics, most errors are below 1%, demonstrating a significant level of accuracy.

The ability of each method to capture local transient effects is shown in Fig. 2, and the presented scale is valid for all four cases. Comparing the HB and MRF velocity fields with the transient one, wakes in the stator blade passage appear only in HB and the transient simulation. The MRF is unable to resolve the blade passage transients due to a static mesh and a steady-state approach. Neither HB simulation resolves wakes completely, suggesting that a higher number of harmonics should be used in that case.

A CPU time comparison is presented in Table 2, showing the HB speed-up compared to the transient simulation. The ERCOFTAC centrifugal pump test case consists of 93,886 hexahedral cells and was simulated on an Intel Core I5-3570K, 3.4 GHz computer. One period of transient simulation lasted nearly 5 h, with a maximum Courant–Friedrichs–Lewy number equal to 0.5 and 600 time steps per period. It was noticed that 8 periods are needed in order to achieve a fully periodic steady-state solution. Taking 8 periods into account, the entire transient simulation took nearly 40 h of CPU time. On the other hand, the HB simulation with one harmonic took nearly 52 min of CPU time and the HB simulation with 2 harmonics took nearly 78 min. Comparing the transient simulation and the HB with two harmonics, the HB simulation shows CPU time speed-up by a factor of 30. The decrease in the number of iterations with an increase in the number of harmonics can be noticed, as 3000 iterations were needed to reach convergence in the 1 harmonic case, as opposed to



(a) Transient simulation.

(b) Harmonic Balance with 1 harmonic.

(c) Harmonic Balance with 2 harmonics.

(d) MRF simulation.

Fig. 2 Local transient effects comparison, velocity field

Table 2 CPU time comparison for the transient simulation, MRF and HB

Transient	1 period = ∼ 5 h, 8 periods = 40 h
MRF	3100 iterations, ∼ 20 min
HB, 1 h	3000 iterations, ∼ 52 min
HB, 2 h	2400 iterations, ∼ 78 min

2400 in the 2 harmonics case. This is considered to be the result of stronger coupling between time instants as larger number of time instants are solved.

Although the presented cases were run in serial using a single core, no significant change in CPU time ratios is expected if it were to be run in parallel.

4 Conclusion

This paper presents a Harmonic Balance method for nonlinear periodic flows. In order to present the Harmonic Balance capabilities, a comparison with conventional transient and steady-state solvers is given. The pressure contours of the rotor blades agree well with the transient simulation. A two harmonics solution agrees with the

transient contour more closely than for a one harmonic case, demonstrating the increase in accuracy with a higher number of harmonics used. The overall results presented show that Harmonic Balance is an accurate and efficient tool for tackling periodic problems, as it demonstrates good flow prediction with significantly lower CPU time compared to a transient simulation. Average errors of global pump parameters are 1.1% for the one harmonic case, while a higher number of harmonics should be used if a more reliable prediction of transient instabilities is wanted. The validation of the model using a 3D test case shall be done for future work, as well as further development of the Harmonic Balance method. Two main aspects of development are to be considered: a compressible flow solver and improvement of accuracy by solving only for the dominant frequencies.

References

1. Cvijetic, G., Jasak, H., and Vukcevic, V., "Finite Volume Implementation of the Harmonic Balance Method for Periodic Non–Linear Flows," *54th AIAA Aerospace Sciences Meeting*, 2016, p. 0070.
2. He, L., "Method of simulating unsteady turbomachinery flows with multiple perturbations," *AIAA Journal*, Vol. 30, 11 1992, pp. 2730–2735.
3. He, L. and Ning, W., "Efficient approach for analysis of unsteady viscous flows in turbomachines," *AIAA Journal*, Vol. 36, No. 11, 1998, pp. 2005–2012.
4. Thomas, J., Custer, C., Dowell, E., and Hall, K., "Unsteady flow computation using a harmonic balance approach implemented about the OVERFLOW 2 flow solver," *19th AIAA Computational Fluid Dynamics Conference*, 2009.
5. Dufour, G., Sicot, F., Puigt, G., Liauzun, C., and Dugeai, A., "Contrasting the Harmonic Balance and Linearized Methods for Oscillating-Flap Simulations," *AIAA Journal*, Vol. 48, No. 4, 2010, pp. 788–797.
6. Ekici, K. and Hall, K. C., "Harmonic Balance Analysis of Limit Cycle Oscillations in Turbomachinery," *AIAA Journal*, Vol. 49, No. 7, 2011, pp. 1478–1487.
7. Hall, K., Thomas, J., Ekici, K., and Voytovych, D., "Frequency domain techniques for complex and nonlinear flows in turbomachinery," Vol. 3998, 2003, p. 2003.
8. Gopinath, A., Van Der Weide, E., Alonso, J., Jameson, A., Ekici, K., and Hall, K., "Three-dimensional unsteady multi-stage turbomachinery simulations using the harmonic balance technique," *45th AIAA Aerospace Sciences Meeting and Exhibit*, Vol. 892, 2007.
9. Guédeney, T., Gomar, A., and Sicot, F., "Multi-frequential harmonic balance approach for the computation of unsteadiness in multi-stage turbomachines," AFM, Maison de la Mécanique, 39/41 rue Louis Blanc, 92400 Courbevoie, France (FR), 2013.
10. Guédeney, T., Gomar, A., Gallard, F., Sicot, F., Dufour, G., and Puigt, G., "Non-uniform time sampling for multiple-frequency harmonic balance computations," *Journal of Computational Physics*, Vol. 236, 2013, pp. 317–345.
11. He, L., "Fourier methods for turbomachinery applications," *Progress in Aerospace Sciences*, Vol. 46, No. 8, 2010, pp. 329–341.
12. Huang, H. and Ekici, K., "Stabilization of High-Dimensional Harmonic Balance Solvers Using Time Spectral Viscosity," *AIAA Journal*, Vol. 52, No. 8, 2014, pp. 1784–1794.
13. Hall, K. C., Thomas, J. P., and Clark, W. S., "Computation of unsteady nonlinear flows in cascades using a harmonic balance technique," *AIAA Journal*, Vol. 40, No. 5, 2002, pp. 879–886.
14. Thomas, J. P., Custer, C. H., Dowell, E. H., Hall, K. C., and Corre, C., "Compact Implementation Strategy for a Harmonic Balance Method Within Implicit Flow Solvers," *AIAA Journal*, Vol. 51, No. 6, 2013, pp. 1374–1381.

15. Sicot, F., Puigt, G., and Montagnac, M., "Block-Jacobi Implicit Algorithms for the Time Spectral Method," *AIAA Journal*, Vol. 46, No. 12, 2008, pp. 3080–3089.
16. Woodgate, M. A. and Badcock, K. J., "Implicit Harmonic Balance Solver for Transonic Flow with Forced Motions," *AIAA Journal*, Vol. 47, No. 4, 2009, pp. 893–901.
17. Su, X. R. and Yuan, X., "Implicit solution of time spectral method for periodic unsteady flows," *International Journal for Numerical Methods in Fluids*, Vol. 63, No. 7, 2010, pp. 860–876.
18. Antheaume, S. and Corre, C., "Implicit Time Spectral Method for Periodic Incompressible Flows," *AIAA Journal*, Vol. 49, No. 4, 2011, pp. 791–805.

Implementation of a Flexible and Modular Multiphase Framework for the Analysis of Surface-Tension-Driven Flows Based on a Hybrid LS-VOF Approach

Paolo Capobianchi, Marcello Lappa and Mónica S. N. Oliveira

Abstract The mathematical modelling and numerical simulation of multiphase flows are both demanding and highly complex. In typical problems with industrial relevance, the fluids are often in non-isothermal conditions, and interfacial phenomena are a relevant part of the problem. A number of effects resulting from the presence of temperature differences must be adequately taken into account to make the results of numerical simulations consistent and realistic. Moreover, in general, gradients of surface tension at the interface separating two liquids are a source of numerical issues that can delay (and in some circumstances even prevent) the convergence of the solution algorithm. Here, we propose a fundamental and concerted approach for the simulation of the typical dynamics resulting from the presence of a dispersed phase in an external matrix under non-isothermal conditions based on the modular computer-aided design, modelling and simulation capabilities of the OpenFOAM® environment. The resulting framework is tested against the migration of a droplet induced by thermocapillary effects in the absence of gravity. The simulations are fully three-dimensional and based on an adaptive mesh refinement (AMR) strategy. We describe in detail the countermeasures taken to circumvent the problematic issues associated with the simulation of this kind of flow.

1 Introduction

The motion of gas bubbles and liquid droplets in fluid media is a widespread phenomenon in nature and a subject of great relevance to many engineering and material processing applications.

A fluid particle can move under the influence of different driving forces of a distinct nature. As an example, a falling raindrop or a bubble rising in a denser liquid are put in motion because of the gravity force exerted on it by Earth. Other body forces

P. Capobianchi (✉) · M. Lappa · M. S. N. Oliveira
James Weir Fluids Laboratory, Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow G1 1XJ, UK
e-mail: paolo.capobianchi@strath.ac.uk

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®* ,
https://doi.org/10.1007/978-3-319-60846-4_18

such as magnetic and ultrasonic fields can also be used to induce or "control" the motion of droplets in some specific circumstances (see, for instance, [7, 16]). Even in the absence of body forces, a similar phenomenon can be induced by another fundamental mechanism, the so-called *thermal Marangoni* (or *thermocapillary*) *effect*. This process, which occurs every time a gradient of temperature or concentration is present at the interface separating two fluid phases, becomes particularly important in all of those situations in which body forces (such as gravity) do not play a major role, e.g., in the microgravity environment provided by orbiting platforms, or when either the densities of the fluid pair are similar and/or the typical size of the dispersed phase is very *small* (e.g., atomised droplets).

The first pioneering study on the thermocapillary migration of droplets was conducted in the late 50s by Young et al. [26], who, under some limiting assumptions (perfectly spherical drop of radius R moving in an infinitely extended fluid domain under Stokes flow conditions), derived a landmark solution (in analytical form) of the governing equations. With such an approach, velocity and temperature fields were considered to be fully established at every moment of time, while neglecting the inertia and convective effects. Under this approximation, the temperature field can be inferred independently of the flow field, and this greatly simplifies the derivation of an analytical solution to the problem, yielding a precise relationship between the asymptotic (steady) droplet migration velocity and the properties of the considered fluids and characteristics of the driving force.

Over subsequent years, especially because of the advent of space programs and the possibility of executing experiments in the field of fluids and materials in space (sounding rockets, Space Shuttle, and most recently, the International Space Station, see, e.g., [12]), the subject gained increasing popularity, extending to multidisciplinary fields.

Balasubramaniam and Subramanian [2] extended the analytical study of Young et al. [26] to the case in which the Reynolds number tends to infinity. They analysed the steady migration of a spherical drop in a continuous phase when subjected to a temperature gradient under conditions such that inertial terms in the momentum equation and convective-transport terms in the energy equation dominate over the corresponding molecular-transport terms (i.e., $Ma \to \infty$ and $Re \to \infty$, for which they were able to derive analytical solutions partially based on the earlier mathematical model of Harper and Moore [11]). The migration velocity of the drop was obtained by these authors in the framework of a potential-flow theory by equating the rate at which work is done by the thermocapillary stress to the rate of viscous dissipation of energy; the method of matched asymptotic expansions was employed to solve the conjugate heat-transfer problem in the two phases (characterised by the presence of thin thermal boundary layers both outside and within the drop). In physical terms, they found that in the limit of $Ma \to \infty$, the velocity of a drop is proportional to the square of the temperature gradient and the cube of the radius of the drop, whereas in the opposite limit ($Ma \to 0$), both dependencies are linear.

In addition to theoretical studies, a number of experimental works have also appeared over the last two decades (see, e.g., [3, 9, 23] for experiments under weightless conditions). More recently, attempts have been made to approach the

problem directly in the framework of moving boundary methods such as Volume of Fluid (VOF) or Level Set (LS) techniques, by which the typical simplifications of past analytical approaches can be removed and the typical difficulties (and cost) of microgravity-based experimentation avoided.

Despite valuable numerical advances over the years [6, 10, 13, 15, 21, 25, 27], the numerical simulation of droplets migrating under the influence of surface-tension gradients at finite values of the Reynolds and Marangoni numbers can still be regarded as an "open task". In the present chapter, we lay the general foundation of a possible theoretical and mathematical treatment of the subject based on the modular capabilities of the OpenFOAM® environment. Starting from an already existing algorithm [24], we undertake all of the steps necessary to expand the range of treatable physical effects. Furthermore, the code is validated against the analytical solution of Young et al. [26] and used to study the impact of the geometrical configuration on the drop migration pattern.

2 Mathematical Formulation

The ingredients of our overall conceptual architecture are provided and discussed in a step-by-step approach, with the aim of defining each of the sub-models as simply as possible and building and growing the framework "organically" by progressive integration of the various components. The class of such sub-parts or sub-models is highly diverse, including typical moving boundary methods and energy transport models in synergy with non-dimensional and asymptotic analyses.

2.1 Governing Equations

We consider the Marangoni migration of a liquid drop surrounded by an immiscible liquid under the effect of a constant temperature gradient $\nabla_\infty T$. The most common way to describe such a flow is based on the consideration of two distinct phases, each with its own set of governing equations, and appropriate interface stress jump conditions to guarantee proper phase coupling (see, e.g., [22]). However, the problem can also be approached in terms of "interface capturing methods", such as the Level Set or the Volume of Fluid. These techniques are based on a different strategy known as the "single-fluid" or "one-fluid" approach (see, e.g., [14], and references therein). The underlying idea is that the system might be considered as if composed of one single fluid with variable material properties (undergoing discontinuities across the fluid–fluid boundaries). In terms of momentum, the presence of the interfacial stresses is accounted for by adding "extra" forces to the transport equation.

More precisely, if we assume that the effect of gravity and any other external body force is negligible, the conservation of momentum can be written as

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)=-\nabla p+\nabla\cdot\left[\mu\left(\nabla\mathbf{u}+(\nabla\mathbf{u})^{T}\right)\right]+\mathbf{f}_{\sigma}, \tag{1}$$

where t represents time, ρ, μ are the fluid density and viscosity, respectively, p is the pressure and $\mathbf{u}$ the velocity vector. The last term $\mathbf{f}_{\sigma}$ is a force accounting for the capillary ($\mathbf{f}_{\sigma,\mathbf{n}}$) and thermocapillary ($\mathbf{f}_{\sigma,\tau}$) forces at the interface

$$\mathbf{f}_{\sigma}=\mathbf{f}_{\sigma,n}+\mathbf{f}_{\sigma,\tau}=\sigma(T_0)k\mathbf{n}\delta_{\mathrm{S}}+\nabla_{\|}\sigma(T)\delta_{\mathrm{S}} \tag{2}$$

Here, k and $\mathbf{n}$ are the curvatures and normal unit vector at the interface, respectively, $\mathbf{I}$ is the identity tensor and the operator $\nabla_{\|}=(\mathbf{I}-\mathbf{nn})\nabla$ accounts for the projection of the surface-tension gradient along the direction tangent to the interface. The term δ_{S} represents a distribution function that takes values one at the interface and zero elsewhere [14]. Since the interfacial tension σ depends on the temperature T, we have explicitly included the related dependence in Eq. 2. Closure of the mathematical model requires consideration of the conservation of mass for incompressible flows (Eq. 3) and the temperature transport equation (Eq. 4)

$$\nabla\cdot\mathbf{u}=0, \tag{3}$$

$$\rho c_p\left(\frac{\partial T}{\partial t}+\mathrm{u}\cdot\nabla T\right)=\nabla\cdot(\kappa\nabla T), \tag{4}$$

where c_p is the specific heat and κ the thermal conductivity of the fluid. Following common practice for this kind of problem (see, e.g., [25]), all material properties are assumed to be constant in each phase and are evaluated at a suitable reference temperature. The dependence on temperature, however, is retained for the surface tension σ via a linear relationship

$$\sigma(T)=\sigma_0+\sigma_T(T-T_0), \tag{5}$$

where $\sigma_T=-\partial\sigma(T)\big/\partial T$ is negative for most known fluids [18] and T_0 is the reference temperature.

2.2 The Simplified LS-VOF Method

Our solver relies on a simplified coupled LS-VOF code (based on the hybrid formulation originally developed by Albadawi et al. [1], see also Sussman and Puckett [20]) implemented into the framework of OpenFOAM® [24] as an extension of the standard VOF solver "interFoam". The simplified coupled LS-VOF for an isothermal system is based on the solution of Eqs. 6–10. The equation for the volume fraction reads as

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{u}) + \nabla \cdot (\alpha(1-\alpha)\mathbf{u}_c) = 0, \tag{6}$$

where α is the volume fraction and $\mathbf{u}_c$ is an artificial "compressive velocity" [4]. Although there was a consistent improvement in terms of accuracy and reduction of the so-called "parasite" currents with respect to the original two-phase solver, we had to take additional countermeasures to fix typical "algorithm stability" issues at the interface (where Marangoni stresses of thermal nature are produced). This was accomplished by "proper" smoothing, both of the level set and the volume of fluid phase functions, as further described in Sect. 2.3.

The resulting time-marching procedure can be outlined as follows: in order to calculate the level set function φ, we first calculate the field $\varphi_0 = (2\alpha - 1)\Gamma$, where $\Gamma = 0.75\,\Delta x$ and Δx is the grid resolution (see [1]). Subsequently, a re-initialisation equation is solved (see, e.g., [19]):

$$\frac{\partial \varphi}{\partial \tau} = \mathrm{Sgn}(\varphi_0)(1 - |\nabla \varphi|), \tag{7}$$

with the initial condition $\varphi(\mathbf{x}, 0) = \varphi_0(\mathbf{x})$ and where $\mathrm{Sgn}(\varphi_0) = \varphi_0 \big/ |\varphi_0|$ and τ is a fictitious time. Once the scalar field φ is known at each point, it is possible to evaluate the curvature at the interface

$$k(\varphi) = -\nabla \cdot \mathbf{n}(\varphi), \tag{8}$$

with $\mathbf{n}(\varphi) = \nabla\varphi \big/ |\nabla\varphi|$ being the unit vector perpendicular to the interface. Finally, the term described by Eq. 2 is evaluated, leading to the momentum equation cast in compact form as

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) = -\nabla p + \nabla \cdot \left[\mu\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}}\right)\right] + \sigma k(\varphi)\eta(\varphi)\nabla\varphi + \sigma_T \nabla_{\parallel} T |\nabla \alpha|, \tag{9}$$

where

$$\eta(\varphi) = \begin{cases} 0 & \text{if } |\varphi| > \varepsilon \\ \frac{1}{2\varepsilon}\left(1 + \cos\left(\frac{\pi\varphi}{\varepsilon}\right)\right) & \text{if } |\varphi| \leq \varepsilon \end{cases} \tag{10}$$

and $2\varepsilon = 3\Delta x$.

The reader is referred to Lappa [14] for additional information about the mathematical manipulations required to turn the surface force seen in Eqs. 1 and 2 into a corresponding volume force spread over a region of finite thickness, which no longer relies on the use of the delta function. Additional details on the dependences on φ and α present in Eq. 9 are provided in Sect. 2.3.

2.3 Implementation of the Thermal Marangoni Migration Method in OpenFOAM®

We used properly mollified variables to increase algorithm stability and avoid non-physical effects at the interface. More precisely, the smoothing was applied to each "relevant variable" χ (various variables required by the LS and VOF implementation in different parts of the solver, as needed) using a "pure diffusive" evolution equation $\chi_{\text{mol}}^{n+1} = \chi_{\text{mol}}^{n} + \left(\nabla^2 \chi_{\text{mol}}^{n}\right)\Delta\tau_{\text{mol}}$, where τ_{mol} represents an artificial or fictitious time, to be solved with the initial condition for a prefixed number of cycles n (the condition $n = 0$ corresponding to the recovery of the original non-smoothed function). $\Delta\tau_{\text{mol}}$ is defined according to the following well-known numerical stability criterion (see, e.g., [8]):

$$\Delta\tau_{\text{mol}} = \frac{0.5^2}{(1/\Delta x)^2 + (1/\Delta y)^2 + (1/\Delta z)^2}. \tag{11}$$

We used mollified quantities to evaluate the new curvature at each time step, i.e.,

$$k_{\varphi\text{mol}} = -\nabla \cdot \mathrm{n}_{\varphi\text{mol}} = -\nabla \cdot \frac{\nabla\varphi_{\text{mol}}}{|\nabla\varphi_{\text{mol}}|}, \tag{12}$$

where φ_{mol} is the smoothed version of φ.

As discussed in Sect. 2.1, accounting for surface-tension effects requires two additional source terms in the momentum equation (see Eq. 2). In the framework of an optimisation strategy based on a trial-and-error approach, we could obtain the best results using the mollified level set function to determine the unit vector perpendicular to the interface (and the corresponding tangent unit vector) and retaining a non-mollified volume fraction in the gradient appearing in the expression of the thermal contribution (see Eq. 13). The level set function was also used accordingly to determine the curvature.

$$\mathbf{f}_{\sigma,\tau} = \sigma_T \nabla_{\parallel} T |\nabla\alpha| = \sigma_T \left(\mathbf{I} - \mathbf{n}_{\varphi\text{mol}}\mathbf{n}_{\varphi\text{mol}}\right)\nabla T |\nabla\alpha|. \tag{13}$$

The portion of the code in which we have included the thermocapillary force is shown in Fig. 1.

Following common practice in the literature [5], the smoothing philosophy has also been applied to the fluid properties (assumed to be constant in each phase) in order to prevent the algorithm from developing spurious oscillations due to the discontinuity established at the liquid–liquid interface. In our hybrid implementation, we decided to rely on a standard VOF approach, expressing each property as

$$\gamma = \alpha_{\text{mol}}\gamma_1 + (1 - \alpha_{\text{mol}})\gamma_2. \tag{14}$$

Special care has also been devoted to the solution of the energy equation. Some mathematical manipulations were indeed necessary to increase algorithm stability

```
fvVectorMatrix UEqn
(
    fvm::ddt(rho, U)
  + fvm::ddt(rhoPhi, U)
  + turbulence->divDevRhoReff(rho, U)
  + ddtSigma*TangentialGradT*mag(fvc::grad(alpha1))
==
    sources(rho, U)
);
```

Fig. 1 Snippet of code from the UEqn.H file. The line highlighted in boldface represents the stresses due to the thermocapillary effect, where ddtSigma represents the interfacial tension coefficient σ_T, `TangentialGradT` is the projection $\nabla_{\|} T$ of the temperature gradient (implemented in the code as `gradT` – (`gradT` & `nMoll`)*`nMoll`, where `gradT` is the temperature gradient and `nMoll` corresponds to $n_{\varphi mol}$) and the last term is the magnitude of the gradient of the volume fraction $|\nabla \alpha|$

```
fvScalarMatrix TEqn
(
    fvm::ddt(T)
  + fvm::ddt(phi, T)
  - fvm::laplacian(D, T)
  - 1.0/(rho*cp)*(fvc::grad(k) & fvc::grad(T))
  + fvc::grad(D) & fvc::grad(T)
);
```

Fig. 2 Implementation of the temperature equation. The name of the variables has direct correspondence to the symbolism adopted in Eq. 15

and its related ability to reproduce available test cases in the literature (as discussed later in this chapter). We rearranged the equation as follows: by introducing the thermal diffusivity $D = \kappa/\rho c_p$ and considering that all the fluid material properties can, in general, change across the interface, after some algebraic manipulations, we obtained the following equivalent expression for the energy equation:

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \nabla \cdot (D \nabla T) + \frac{1}{\rho c_p} \nabla \kappa \cdot \nabla T - \nabla D \cdot \nabla T. \tag{15}$$

The snippet of code displayed in Fig. 2 shows the corresponding implementation.

The different macro-steps in which our algorithm has been articulated can ultimately be sketched as follows:

1. Solve the re-initialization equation (Eq. 7);
2. Solve the volume fraction equation using the MULES algorithm (which is essentially an explicit method, see, e.g., the OpenFOAM® user guide [17] to guarantee the boundedness of the scalar field α;
3. Solve the temperature equation;
4. Compute the thermal Marangoni force $\mathbf{f}_{\sigma,\tau}$;
5. Calculate the velocity and pressure field using a projection method (PISO algorithm);
6. Go back to step 1 or end of calculation.

Before the validation and discussion of the results in the next section, we will list here the independent non-dimensional parameters governing the physics of the flow under discussion. These are the fluid property ratios $\widetilde{\rho} = \rho_\mathrm{d}/\rho_\mathrm{m}$, $\widetilde{\mu} = \mu_\mathrm{d}/\mu_\mathrm{m}$, $\tilde{c}_\mathrm{p} = c_{\mathrm{p,d}}/c_{\mathrm{p,m}}$ and $\tilde{k} = k_\mathrm{d}/k_\mathrm{m}$, the Capillary number $Ca = \sigma_T(\nabla_\infty T)R/\sigma_0$, the Marangoni number $Ma = \sigma_T(\nabla_\infty T)R^2/\alpha_\mathrm{m}\mu_\mathrm{m}$ and either the Reynolds number $Re = \rho_\mathrm{m}\sigma_T(\nabla_\infty T)R^2/\mu_\mathrm{m}^2$ or the Prandtl number $Pr = \mu_\mathrm{m}/\rho_\mathrm{m}\alpha_\mathrm{m}$, since $Ma = Re\,Pr$. The subscripts "m" and "d" stand for matrix and drop, respectively.

3 Solver Validation

As indicated at the end of Sect. 2, our overall framework has been built via the integration of self-contained modules, which could be individually tested. However, because it is crucial that the entire numerical architecture be tested as a single integrated unit, we considered available solutions in the literature for comparison. In order to validate our code, in particular, we focused on the thermocapillary motion of a spherical Newtonian droplet of radius R in a constant temperature gradient $\nabla_\infty T$ embedded in an unconfined Newtonian matrix in the limiting case of $(Ma, Re) \to 0$ and negligible buoyancy effects. As discussed in the introduction, in such a case, an analytical solution exists for the velocity of the droplet [26] (YGB theory), which, in dimensional form, reads as

$$U_\mathrm{YGB} = \frac{2|\sigma_T|(\nabla_\infty T)R/\mu_\mathrm{m}}{\left(2+\frac{k_\mathrm{d}}{k_\mathrm{m}}\right)\left(2+3\frac{\mu_\mathrm{d}}{\mu_\mathrm{m}}\right)}, \tag{16}$$

where the temperature gradient $\nabla_\infty T$ is defined as $\nabla_\infty T = (T_\mathrm{hot} - T_\mathrm{cold})/H$, where H is the height of the channel along the direction of the motion (between the hot wall and the cold wall) and T_hot and T_cold correspond to the temperature at the hot wall and cold wall, respectively (cf. Fig. 3a). In our simulations, we assumed conditions corresponding to the following set of (non-dimensional) characteristic numbers: $Pr = 0.1$, $Re = 1.0 \times 10^{-4}$, $Ma = 1.0 \times 10^{-5}$ and $Ca = 2.0 \times 10^{-1}$ (with the capillary number being sufficiently small to guarantee negligible deformations, see, e.g.,

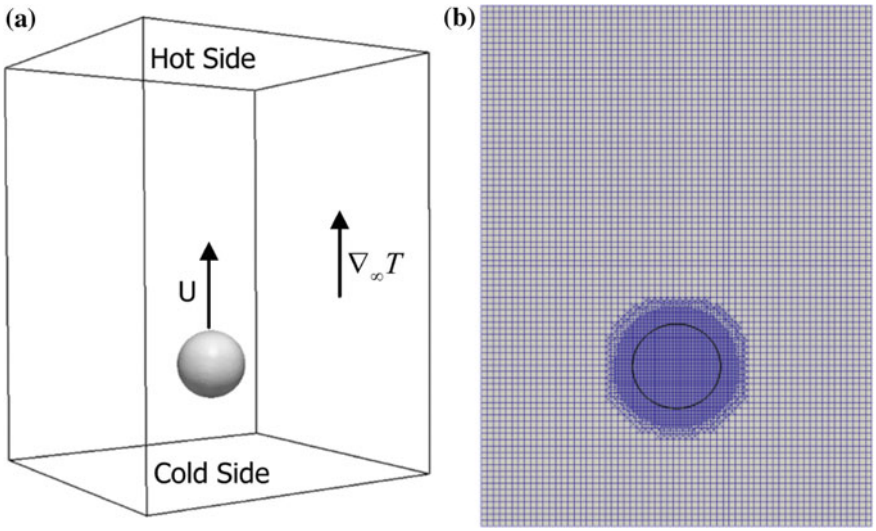


Fig. 3 **a** Initial conditions used for the simulation. At the initial instant, the droplet is at rest. The velocity arrow has been included to show the droplet migration direction when a temperature gradient $\nabla_\infty T$ is imposed. **b** Particular of the mesh in a plane parallel to the motion of the droplet. Note the area of refinement in the region of the droplet

[27]). For simplicity, we also considered the fluid material parameters to be the same for both fluids (i.e., unit fluid property ratios). Assuming the radius of the droplet to be $R = 0.5\,\text{cm}$, we fixed the size of the external container to $(6 \times 4.5 \times 4.5)\,\text{cm}^3$ (shown in Fig. 3a), corresponding to a confinement ratio $C = R/L = 0.22$, where L represents the distance between the centre of the drop and the wall. This size is intended to mimic the effective geometry of the container used in microgravity experiments by Hadland et al. [9]. As shown in Fig. 3b, a structured mesh with $85 \times 64 \times 64$ elements adaptively refined in the region of the drop is employed. For the boundary conditions, we have applied no-slip conditions for the velocity and `"zeroGradient"` for the pressure at each wall of the container (a reference pressure `"pRefValue = 0"` has been set at the centre of the "cold" wall). For the temperature, we set constant values at the "cold" and "hot" sides and adiabatic (`"zeroGradient"`) conditions in the rest of the boundaries of the domain.

All simulations were executed applying two ($n = 2$) cycles of smoothing for the Level Set function φ.

Figure 4 shows a snapshot of the flow pattern and the temperature distribution in a plane parallel to the direction of migration in the case of $C = 0.22$. As expected, a toroidal roll is formed as a result of the thermocapillary effect. The intersection of such a roll with the considered visualisation plane clearly shows regions of recirculating fluid surrounding the droplet. As time increases, such vortices move with the droplet.

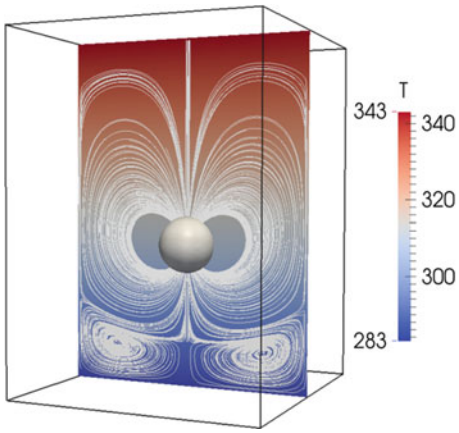


Fig. 4 Snapshot of the flow pattern and temperature field in a plane parallel to the direction of migration at instant $t' = 10$. The droplet is moving from the bottom (cold side) toward the top (hot side)

Figure 5 shows the dimensionless migration velocity as a function of the dimensionless time $t' = \mu_m/\sigma_T(\nabla_\infty T)$ for the same case. As evident in this figure, after a given transient, the droplet reaches a steady state in which its (final) migration velocity is in excellent agreement with the predictions of the YGB theory.

We also studied the effect of the geometric "confinement", considering narrower containers with $C = R/L > 0.22$. We found that the migration velocity decreases nontrivially with the degree of confinement. More specifically, when the height of the geometry is halved ($C = 0.44$), the resulting steady-state migration velocity is about 12% smaller than the limit predicted by the YGB theory. Such results, summarised in Fig. 5, clearly indicate that some care should be taken in the choice of geometry if wall effects are not intended to be a relevant aspect of the analysis.

The last test considered was the simulation of a droplet migrating into a convergent channel, resorting to the same set-up adopted for the previous simulations. We used a convergent duct having a "cold side" cross-sectional area equal to the case of $C = 0.44$, with the cross-sectional area of the "hot side" being half of the cold one.

It is worth emphasising that in such a case, the droplet could not reach a steady state (its velocity increases monotonically, until it suddenly decreases when the droplet is close to the hot wall). Such interesting behaviour, which would require further investigation, might perhaps be explained by the presence of two different counteracting contributions: as the droplet migrates along the converging region, the degree of confinement increases, and on the basis of the previous findings, one should expect the velocity to decrease; however, since the temperature field distribution is no longer linear, and the Marangoni stresses increase accordingly, one should expect the droplet to accelerate. Further studies are in progress along these lines to assess the role played in such dynamics by the relative importance of molecular and inertial transport terms in the governing equations.

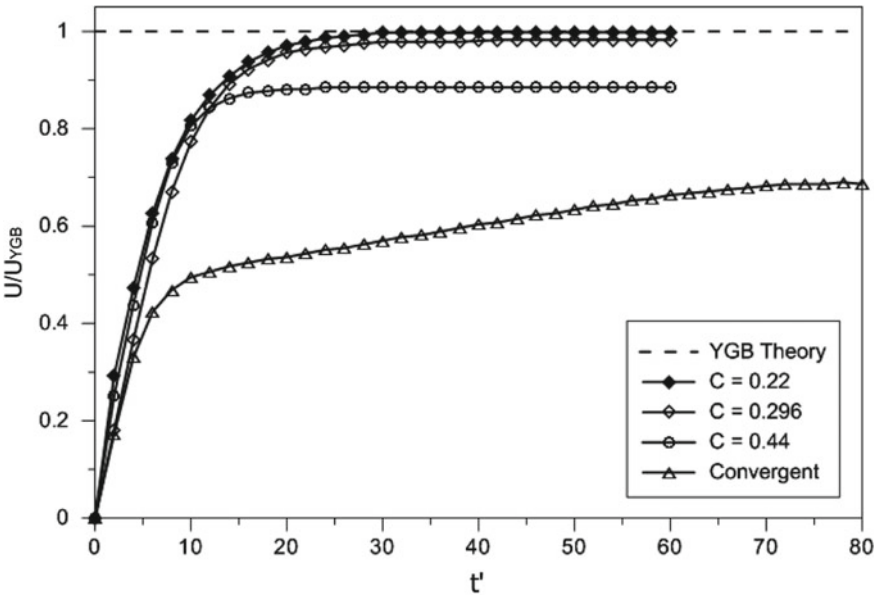


Fig. 5 Velocity history for various geometry and confinements C. The results are normalised using the YGB theoretical velocity. Notice the difference between the two cases $C = 0.296$ (still in good agreement with the YGB theory) and the case $C = 0.44$

4 Conclusions and Future Directions

In this work, we addressed the question of how a typical numerical framework for isothermal multiphase flows can be adequately extended to make it suitable for the simulation of phenomena in which surface-tension gradients act as the main flow or pattern driver. In particular, starting from existing implementations in OpenFOAM® of moving boundary methods, some effort has been put into strengthening the used approach by incorporating the possibility of accounting for thermal effects of different nature in the algorithm. Special care has been devoted to numerical stability issues that are typical of such problems (in which the phenomena occurring in a limited neighbourhood of the interface separating the two liquids play a "crucial role").

The framework has been successfully tested under restricted conditions, but future work shall be devoted to the application and testing of the resulting approach to more complex problems in which inertial terms in the momentum equation and convective-transport terms in the energy equation dominate over the corresponding molecular-transport terms.

References

1. Albadawi A., Donoghue D. B., Robinson A. J., Murray D. B., Delauré Y. M. C., (2013), Influence of surface tension implementation in volume of fluid and coupled Volume of Fluid with Level Set methods for bubble growth and detachment, International Journal of Multiphase Flow, 53, 11–28.
2. Balasubramaniam R. and Subramanian R. S, (2000), The migration of a drop in a uniform temperature gradient at large Marangoni numbers, *Physics of Fluids*, 12(4): 733–743.
3. Balasubramaniam R., Lacy C.E., Wozniak G., Subramanian R.S., (1996), Thermocapillary migration of bubbles and drops at moderate values of the Marangoni number in reduced gravity, *Physics of Fluids*, 8(4): 872–880.
4. Berberović, van Hinsberg N. P., Jarkirlić S., Roisman L. V., Tropea C., (2009), Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution, *Phyical. Review E,* 79, 036306.
5. Brackbill J.U., Kothe D.B., Zemach C., (1992), A Continuum Method for Modeling Surface Tension, *Journal of Computational Physics*, 100 (2): 335–354.
6. Brady P. T., Hermann M., Lopez J. M., (2011), Confined thermocapillary motion of a three-dimensional deformable drop, *Physics of Fluids,* 23, 022101.
7. Brunet P., Baudoin M., Bou Matar O., Zoueshtiagh F., (2010), Droplet displacement an doscillation induced by ultrasonic surface acoustic waves: A quantitative study, *Physical Review E* 81, 036315.
8. Fletcher C. A. J., Compuational techniques for fluid-dynamics (Springer Verlag, Berlin, 1991).
9. Hadland P.H., Balasubramaniam R., Wozniak G., Subramanian R.S., (1999), Thermocapillary migration of bubbles and drops at moderate to large Marangoni number and moderate Reynolds number in reduced gravity, *Experiments in Fluids*, 26: 240–248.
10. Haj-Hariri H., Shi Q., Borhan A., (1997), Thermocapillary motion of deformable drops at finite Reynolds and Marangoni numbers, *Physics of Fluids* 9 (4):845–855.
11. Harper J.F. and Moore D.W., (1968), The motion of a spherical liquid drop at high Reynolds number*, Journal of Fluid Mechanics*, 32(2): 367–391.
12. Lappa M., (2004), Fluids, Materials and Microgravity: Numerical Techniques and Insights into the Physics, 538 pages, Elsevier Science (2004, Oxford, England).
13. Lappa M., (2005a) Assessment of VOF Strategies for the analysis of Marangoni Migration, Collisional Coagulation of Droplets and Thermal wake effects in Metal Alloys under Microgravity conditions, *Computers, Materials & Continua*, 2(1), 51–64.
14. Lappa M., (2005b), Coalescence and non-coalescence phenomena in multi-material problems and dispersed multiphase flows: Part 2, a critical review of CFD approaches, *Fluid Dynamics & Materials Processing*, 1(3): 213–234.
15. Ma X., Balasubramanian R., Subramanian R. S., (1999), Numerical simulation of thermocapillary drop motion with internal circulation, *Numerical Heat Transfer,* 35, 291–309.
16. Nguyen N., Ng K. M., Huang X., (2006), Manipulation of ferrofluid droplet using planar coils, *Applied Physics Letters* 89, 052509.
17. OpenFOAM® User Guide, 2008.
18. Subramanian R. S and Balasubramaniam R., (2001), The motion of bubbles an drops in reduced gravity, *Cambridge University Press.*
19. Sussman, M. and Fatemi, E., (1999): An efficient, interface-Preserving Level Set Redistancing Algorithm and its application to Interfacial Incompressible Fluid Flow. *SIAM Journal of. Scientific Computing*, 20, 1165–1191.
20. Sussman, M., Puckett, E., (2000), A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, vol. 162, pp. 301–337.
21. Tryggvason G., Bunner B., Esmaeeli A., Juric D., Al-Rawahi N., Tauber W., Han J., Nas S., and Jan Y.-J., (2001), A Front Tracking Method for the Computations of Multiphase Flow, *Journal of Computational Physics*, 169: 708–759.

22. Tryggvason G., Scardovelli R., Zaleski S., (2011), Direct numerical simulations of gas-liquid multiphase flows, *Cambridge University Press.*
23. Wozniak G., (1991), On the thermocapillary motion of droplets under reduced gravity, *Journal of Colloid and Interface Science*, 141(1): 245–254.
24. Yamamoto T., Okano Y., Dost S., (2016), Validation of the S-CLSVOF method with the density-scaled balanced continuum surface force model in multiphase systems coupled with thermo-capillary flows, *International Journal of Numerical Methods in Fluids*.
25. Yin Z., Chang L., Hu W., Li Q., and Wang H., (2012), Numerical simulations on thermocapillary migrations of nondeformable droplets with large Marangoni numbers, *Physics of Fluids,* 24, 092101.
26. Young N.O., Goldstein J.S., Block M.J., (1959), The motion of bubbles in a vertical temperature gradient*, Journal of Fluid Mechanics*, 6, 350–360.
27. Zhao J., Zhang L., Li Z., Qin W., (2011), Topological structure evolvement of flow and temperature fields in deformable drop marangoni migration in microgravity, *International Journal* of *Heat and Mass Transfer,* 54, 4655–4663.

Implicitly Coupled Pressure–Velocity Solver

Tessa Uroić, Hrvoje Jasak and Henrik Rusche

Abstract Formulation of implicitly coupled incompressible and compressible pressure–velocity solvers is presented in this paper. The formulation is an alternative to commonly used segregated solvers, in which inter-equation coupling is resolved by Picard iterations. In the coupled solver, the momentum and continuity (pressure) equations are solved simultaneously, in a single block matrix. Turbulence model equations and energy equation in compressible flow are solved in a segregated manner. The formulation is based on deriving the pressure equation as a Schur complement, including the Rhie–Chow correction. A new formulation of the compressible pressure-based solver is proposed by assuming an isentropic compression/expansion, resulting in consistent reduction of the compressible form to incompressible form.

1 Introduction

The second decade of the twenty-first century has brought a new direction in the use of numerical methods in industrial product development and optimisation. Over the last 20 years, the capabilities of numerical modelling tools have proved to be well established: physics models and numerical solution algorithms have been examined and validated for a variety of applications. While the aspects of numerical accuracy and physics model development still remain, the most challenging aspect of modern

T. Uroić (✉) · H. Jasak
Faculty of Mechanical Engineering and Naval Architecture,
Ivana Lučića 5, 10000 Zagreb, Croatia
e-mail: tessa.uroic@fsb.hr

H. Jasak
Wikki Ltd., 459 Southbank House, London SE1 7SJ, UK
e-mail: hrvoje.jasak@fsb.hr; h.jasak@wikki.co.uk

H. Rusche
Wikki Gesellschaft für numerische Kontinuumsmechanik mbH,
Görgesstraße 24, 38118 Braunschweig, Germany
e-mail: henrik.rusche@wikki-gmbh.de

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_19

Computational Fluid Dynamics (CFD) is the need for fast and efficient large-scale computations.

The dominant numerical solution techniques originate from work done in 1970s, ’80s and ’90s, when the balance between computational cost and mere ability to produce the solution were critical. With the rapid development of modern computers and access to robust and powerful High-Performance Computing (HPC) clusters with substantial memory and storage capacities, compromises made in the past can now be revisited.

Today, CFD tools are used for complex, coupled and nonlinear heat and mass transfer problems. In many cases, inter-equation coupling terms dominate the system of equations, and most algorithms still use the segregated, sequential algorithms in their search for the solution. The most notable class of solution techniques relies on linearisation of inter-equation coupling terms and the solution in a coupled, implicit manner. However, the memory efficiency compromise implied by such algorithms is significantly different than it is in weakly coupled segregated algorithms. The benefit is the prospect of significantly reduced time-to-solution.

The considered set of equations consists of the momentum equation, with velocity as the primitive variable, and the continuity equation, with pressure as the primitive variable. SIMPLE [13] and PISO [8] and the variants of these algorithms are the most popular methods for dealing with inter-equation coupling in the pressure–velocity system. The equations are solved in a sequential manner: the momentum equation is solved with the available (old or guessed) values of pressure. The pressure field is then calculated by inserting the momentum equation with the calculated velocity value into the continuity equation, i.e. velocity is being solved with the guessed values of pressure, and vice versa, until convergence. The decoupling of these linearly coupled variables causes slow convergence. Attempts at implicit coupling have been limited by the technology of the time, but a brief overview of the efforts will be given.

Raithby and Schneider [14] reported on new methods of solving the pressure–velocity system, mainly adding improvements to the existing SIMPLE algorithm. It was concluded that keeping more of the effects of the coupling between the velocity and pressure significantly increases the convergence rates.

Zedan and Schneider [19] assembled a pressure equation by replacing the velocity components in the continuity equation with their respective values from the momentum equations and keeping the pressure terms as unknowns. This resulted in a more implicit formulation, which gave better solver performance but was still solved in a segregated manner.

Zedan and Schneider [20] continued this effort presenting a Coupled Strongly Implicit Procedure in which continuity and momentum equations for each point in the solution domain were written together by keeping their primitive form in a submatrix. However, the procedure performed poorly, due to the inversion of the block coefficients. All these algorithms were implemented and validated for 2D problems using the finite difference method.

Implicit coupling has recently been gaining popularity, and new work has been published by Mazhar [10, 11] and Darwish [3, 4] in the finite volume framework. However, segregated methods are still the most widely used.

In the following section, a block implicit formulation of the continuity and momentum equations implemented in the finite volume framework will be presented.

2 Mathematical and Numerical Model

In this section, the mathematical and numerical models for incompressible and compressible flow will be presented.

2.1 Incompressible Formulation

The mathematical model of the steady state, incompressible, single-phase and turbulent flow consists of the continuity equation

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

and the momentum equation

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) - \nabla \cdot (\nu \nabla \mathbf{u}) = -\nabla p. \tag{2}$$

In the incompressible flow formulation, p is defined as

$$p = \frac{P}{\rho}, \tag{3}$$

where P is the pressure field and ρ is the density field. The momentum equation is nonlinear, due to the convective term. In order to eliminate the nonlinearities, the term is linearised as follows:

$$\nabla \cdot (\mathbf{u}\mathbf{u}) = \nabla \cdot (\mathbf{u}^o \mathbf{u}^n), \tag{4}$$

where superscript o denotes the values available from the previous iteration and n the new value to be obtained from the solution of the linear system. The linearised system can now be written in block form:

$$\begin{bmatrix} [A_{\mathbf{u}}] & [\nabla(.)] \\ [\nabla \cdot (.)] & [0] \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{5}$$

The problem that arises in Eq. 5 is the presence of the zero-block on the diagonal of the system. This limits the choice of linear system solvers to those that are saddle-point

specific. However, the problem can be remedied by deriving the pressure equation in the form of the Schur complement.

Consider a general block matrix system M, consisting of four block matrices, A, B, C and D, which are, respectively, $p \times p$, $p \times q$, $q \times p$ and $q \times q$ matrices, and assume A is invertible

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}. \tag{6}$$

This structure will arise naturally when trying to solve a block system of equations

$$\begin{aligned} Ax + By &= a, \\ Cx + Dy &= b. \end{aligned}$$

The Schur complement arises when we try to eliminate x from the system using partial Gaussian elimination (multiplying the first row with A^{-1})

$$A^{-1}Ax + A^{-1}By = A^{-1}a, \tag{7}$$

then expressing x as

$$x = A^{-1}a - A^{-1}By. \tag{8}$$

Substituting x with Eq. 8 in the second of the system row yields

$$(D - CA^{-1}B)y = b - CA^{-1}a. \tag{9}$$

Identifying the submatrices A, B, C, D and vectors x, y, a, b in the pressure–velocity system, Eq. 5, produces the following form of the pressure equation:

$$[\nabla\bullet(.)][A_{\mathbf{u}}^{-1}][\nabla(.)][p] = 0. \tag{10}$$

Inverting a sparse momentum matrix $A_{\mathbf{u}}$ will likely produce a dense matrix. Thus, Eq. 10 is impractical and the following modification is introduced: the momentum matrix $A_{\mathbf{u}}$ is decomposed into the diagonal and off-diagonal parts before the Schur complement, Eq. 11.

$$[A_{\mathbf{u}}] = [D_{\mathbf{u}}] + [LU_{\mathbf{u}}], \tag{11}$$

where $D_{\mathbf{u}}$ only contains block diagonal entries and $LU_{\mathbf{u}}$ the lower and the upper triangle of the momentum matrix $A_{\mathbf{u}}$. Substituting the decomposition Eq. 11 into Eq. 5 and moving the off-diagonal component $LU_{\mathbf{u}}$ onto the right-hand side yields

$$\begin{bmatrix} [D_{\mathbf{u}}] & [\nabla(.)] \\ [\nabla\bullet(.)] & [0] \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} -[LU_{\mathbf{u}}][\mathbf{u}] \\ 0 \end{bmatrix}. \tag{12}$$

Using Eq. 9, the pressure equation is formulated as

$$[\nabla\bullet(.)][D_{\mathbf{u}}^{-1}][\nabla(.)][p] = [\nabla\bullet(.)][D_{\mathbf{u}}^{-1}][LU_{\mathbf{u}}][\mathbf{u}]. \tag{13}$$

In this way, Eq. 13 can be used in the block-coupled pressure–velocity system, Eq. 5, removing the need for saddle-point solvers. The final form of the incompressible block-coupled system reads as

$$\begin{bmatrix} [A_{\mathbf{u}}] & [\nabla(.)] \\ [\nabla\bullet(.)] & [\nabla\bullet(D_{\mathbf{u}}^{-1}\nabla(.))] \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} 0 \\ \nabla\bullet(D_{\mathbf{u}}^{-1}\overline{\nabla(p^o)}) \end{bmatrix}, \tag{14}$$

where p^o is the previously available pressure solution and is only used in the Rhie–Chow correction, [15]. The overline indicates a face-interpolated cell-centred pressure gradient.

2.2 Compressible Formulation

In this section, the conventional and the novel coupled formulation of the compressible pressure–velocity solver will be introduced, as implemented in foam-extend, a community-driven fork of OpenFOAM® software.

The starting point is the compressible continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla\bullet(\rho\mathbf{u}) = 0, \tag{15}$$

where ρ is the density field.

The compressible form of the momentum equation, assuming a Newtonian fluid reads as:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\bullet(\rho\mathbf{u}\mathbf{u}) - \nabla\bullet\left[\mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\right] = -\nabla\left(P + \frac{2}{3}\mu\nabla\bullet\mathbf{u}\right), \tag{16}$$

where μ is the dynamic viscosity.

The internal energy equation reads as

$$\frac{\partial(\rho e)}{\partial t} + \nabla\bullet(\rho e\mathbf{u}) - \nabla\bullet(\lambda\nabla T) = \rho\mathbf{g}\bullet\mathbf{u} - \nabla\bullet(P\,\mathbf{u}) - \nabla\bullet\left(\frac{2}{3}\mu(\nabla\bullet\mathbf{u})\,\mathbf{u}\right) + \nabla\bullet\left[\mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\bullet\mathbf{u}\right], \tag{17}$$

where T is the temperature and λ is the thermal conductivity.

The relation between ρ, P and T is defined using the equation of state. For an ideal gas

$$\rho = \frac{P}{RT} = \psi P, \tag{18}$$

where ψ is compressibility

$$\psi = \frac{1}{R\,T}. \tag{19}$$

The first step in the derivation of the compressible pressure equation is the transformation of the rate-of-change term. Using the chain rule on $\rho = \rho(P, T)$, it follows that

$$\frac{\partial \rho}{\partial t} = \frac{\partial \rho}{\partial P}\frac{\partial P}{\partial t} + \frac{\partial \rho}{\partial T}\frac{\partial T}{\partial t}. \tag{20}$$

The second term introduces a source term dependent on the rate of change of temperature, allowing for a general polytrophic state change. However, the formal derivation relies on the substitution of the equation of state into the rate-of-change term. From the ideal gas law, it follows that

$$\frac{\partial \rho}{\partial P} = \psi. \tag{21}$$

The discretised form of the momentum equation can be written in the following form:

$$a_P^{\mathbf{u}}\mathbf{u}_P = \mathbf{H}(\mathbf{u}) - \nabla p. \tag{22}$$

The velocity from Eq. 22 is substituted into the divergence term in Eq. 15 yielding

$$\nabla\cdot(\rho\mathbf{u}) = \nabla\cdot\left[\rho(a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})\right] - \nabla\cdot\left[\rho(a_P^{\mathbf{u}})^{-1}\nabla P\right]. \tag{23}$$

Using $\rho = \psi P$, it follows that

$$\nabla\cdot\left[\rho(a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})\right] = \nabla\cdot\left[\psi P(a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})\right] = \nabla\cdot(\mathbf{F}_p\,P), \tag{24}$$

where $\mathbf{F}_p$ is the convective flux for the pressure

$$\mathbf{F}_p = \psi(a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u}). \tag{25}$$

Note that, unlike the divergence of the mass flux, the divergence of $\mathbf{F}_p$ is not zero, as no conservation law applies ($\nabla\cdot\mathbf{F}_p \neq \mathbf{0}$).

Looking at the terms in Eq. 24, the structure of the compressible pressure equation can be established. The first term, $\nabla\cdot\left[\psi P(a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})\right]$, represents the convective effects and is responsible for the appearance of shocks, while the second term, $\nabla\cdot\left[\rho(a_P^{\mathbf{u}})^{-1}\nabla P\right]$, leads to a pressure Laplacian equivalent to the one seen in the incompressible pressure equation, Eq. 29.

The final form of the compressible pressure equation reads as

$$\frac{\partial(\psi P)}{\partial t} + \nabla\cdot\left[\psi(a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})\,P\right] - \nabla\cdot\left[\rho^o\,(a_P^{\mathbf{u}})^{-1}\nabla P\right] = 0. \tag{26}$$

2.2.1 Segregated Compressible Flow Formulation

The segregated solver operates on the equation set consisting of the discretised momentum equation, Eq. 22, the discretised compressible pressure equation, Eq. 26 and the appropriate form of the discretised energy equation. Here, only the pressure–velocity part of the system is presented

$$a_P^{\mathbf{u}}\mathbf{u}_P + \sum_N a_N^{\mathbf{u}}\mathbf{u}_N = \mathbf{r} - \nabla p, \tag{27}$$

$$\frac{\partial(\psi P)}{\partial t} + \nabla\cdot\left[\psi (a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})\, P\right] - \nabla\cdot\left[\rho^o\, (a_P^{\mathbf{u}})^{-1}\nabla P\right] = 0. \tag{28}$$

The coupled block solver was formulated using Eqs. 27 and 28; however, this form of the discretised equations resulted in instabilities that led to non-physical temperatures, which could be resolved via multiple outer iterations over the p-U-h equation set. However, such an outer iteration loop defeats the purpose of the coupled solver, which is to provide a robust and accurate solver for compressible flows with high Courant–Friedrich–Levy numbers.

2.2.2 Coupled Compressible Flow Formulation

In this section, an alternative formulation of the compressible pressure equation will be presented. The advantage of pressure-based compressible flow solvers over their density-based counterparts are that a pressure-based system correctly reduces to the incompressible formulation as the compressibility ψ approaches zero [6]. This is important for the stability and accuracy of the algorithm, especially when there exists a zone in the flow field where the effective Mach number approaches zero.

The discretized incompressible pressure equation has the following form:

$$\nabla\cdot\left[(a_P^{\mathbf{u}})^{-1}\nabla p\right] = \nabla\cdot((a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})), \tag{29}$$

and its compressible counterpart is

$$\frac{\partial(\psi P)}{\partial t} + \nabla\cdot\left[\psi (a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})\, P\right] - \nabla\cdot\left[\rho^o\, (a_P^{\mathbf{u}})^{-1}\nabla P\right] = 0. \tag{30}$$

It appears that the divergence term on the left-hand side of Eq. 29 has its counterpart in the last term on the left-hand side of Eq. 30, while the explicit term on the right-hand side of Eq. 29 is analogous to the implicit convection term on the left-hand side of Eq. 30, indicating consistency.

However, inserting the incompressibility condition $\psi = 0$ into Eq. 30 does not yield the incompressible formulation, as the implicit convection term on the left-hand side of the equation does not convert into the right-hand side term of Eq. 29.

The cause of this is the fact that, at $\psi \rightarrow 0$, the relationship between ρ and P is lost, rather than being correctly converted into the incompressible form for $\rho = const.$ This is a fundamental failure of the compressible pressure equation derivation by substituting the equation of state $\rho = \psi P$ into the continuity equation, Eq. 18, in the same manner as for density-based solvers.

Based on the previous analysis, a new derivation of the compressible pressure equation based on an alternative linearisation of the $P - \rho$ relation is proposed.

The linearisation of the density change between the two states, formally associated with the old-time-level and a new-time-level in the discretisation reads as

$$\frac{\partial \rho}{\partial t} = \frac{\partial \rho}{\partial P}\frac{\partial P}{\partial t} + \frac{\partial \rho}{\partial T}\frac{\partial T}{\partial t} = \psi \frac{\partial P}{\partial t} + \frac{\partial \rho}{\partial T}\frac{\partial T}{\partial t}, \tag{31}$$

where the first term already exists in the compressible pressure equation, Eq. 30 and the second term accounts for arbitrary polytrophic compression/expansion. Since the second term in Eq. 31 is implicit in T, it cannot reasonably be made implicit in the new pressure equation.

The original derivation, Eq. 30, neglects the term containing $\delta T/\delta t$, implying that the compression or expansion process is isothermal, which may involve a substantial amount of heat exchange. In most cases, a rapid change in pressure indicates a poor pressure field guess in the momentum predictor and is mostly numerical in nature. The actual form of heat exchange is governed by the energy equation, rather than the pressure (correction)—which is, in turn, associated with numerical instability.

Based on this argument, it can be assumed that the nature of compression/expansion described in Eq. 31 should be isentropic, i.e. the change in pressure required to ensure that the velocity field produced by the predictor should not require substantial heat exchange.

Therefore, isentropic compression/expansion is prescribed between P^o and P^n in the rate-of-change term

$$\frac{\partial \rho}{\partial t} = \psi_s^o \frac{P^n - P^o}{\Delta T}, \tag{32}$$

where ψ_s is the isentropic compression/expansion coefficient

$$\psi_s = \frac{1}{\gamma RT} \tag{33}$$

and $\gamma = \frac{C_p}{C_v}$.

Linearisation of the implicit convection term follows from Eq. 23. Using

$$\rho^n = \rho^o + \psi_s (P^n - P^o), \tag{34}$$

the convection term can be rewritten as follows:

$$\nabla\cdot\left[\rho(a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})\right] = \nabla\cdot\left[(\rho^o - \psi_s P^o)(a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})\right] + \nabla\cdot\left[\psi_s(a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})P^n\right], \tag{35}$$

where the first term on the right-hand side of Eq. 35 is explicit and the second one is implicit.

The new form of the compressible pressure equation, accounting for the isentropic compression/expansion and new linearisation of the convection term reads as

$$\begin{aligned}\frac{\partial(\psi_s P)}{\partial t} &+ \nabla\cdot\left[\psi_s(a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})P^n\right] \\ &+ \nabla\cdot\left[(\rho^o - \psi_s P^o)(a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})\right] \\ &- \nabla\cdot\left[\rho^o\,(a_P^{\mathbf{u}})^{-1}\nabla P\right] \\ &= 0.\end{aligned} \tag{36}$$

Unlike Eqs. 30 and 36 correctly reduces to its incompressible form for $\psi_s = 0$, which was the objective of rederivation.

Assembly of the pressure-based block-coupled compressible flow solver follows the procedure already presented in Sect. 2.1, Eq. 14, with two changes:

- The incompressible pressure equation is replaced with its reformulated compressible form, Eq. 36;
- The $\nabla\cdot\mathbf{u}$ block in Eq. 14 is replaced by $\nabla\cdot(\rho^o\mathbf{u}^n)$ and made implicit in $\mathbf{u}$ in the off-diagonal.

The final form of the steady compressible block-coupled system reads as

$$\begin{bmatrix} [A_{\mathbf{u}}] & [\nabla(.)] \\ [\nabla\cdot(\rho^o .)] & \nabla\cdot\left[(\psi_s(a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u}))(.)\right] - [\nabla\cdot(\rho^o D_{\mathbf{u}}^{-1}\nabla(.))] \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ P \end{bmatrix} = \begin{bmatrix} 0 \\ \nabla\cdot\left[\psi_s P^o(a_P^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})\right] - \nabla\cdot[D_{\mathbf{u}}^{-1}\overline{\nabla(P^o)}] \end{bmatrix}, \tag{37}$$

where the overline indicates face interpolation of the cell-centre pressure gradient, in line with the Rhie–Chow procedure.

3 Validation and Benchmarking

In this section, validation of the incompressible and compressible coupled pressure–velocity solver is presented. The incompressible solver is benchmarked versus its segregated counterpart based on the SIMPLE algorithm.

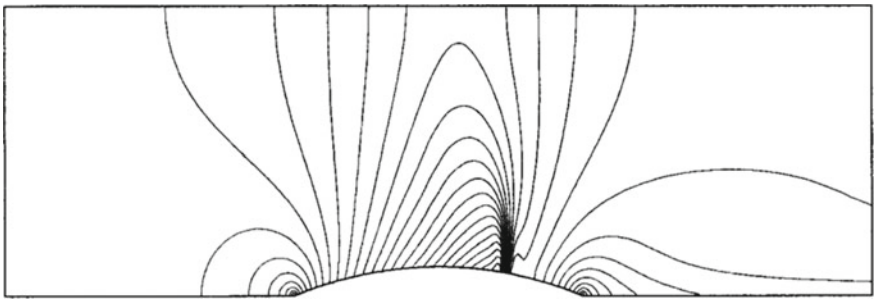


Fig. 1 Mach number contours from the literature for transonic flow [6]

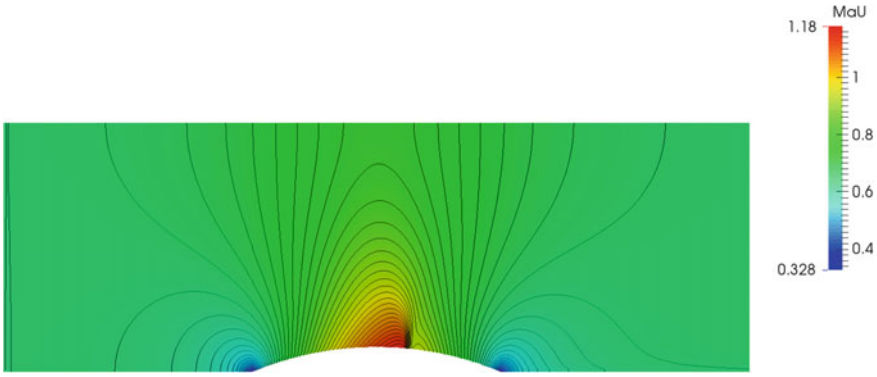


Fig. 2 Calculated Mach number contours for transonic flow

3.1 Validation of the Compressible Coupled Solver

The compressible coupled formulation was validated for flow over a circular arc bump, which is described in [6]. The viscosity was set to zero with slip boundary conditions on the upper and lower wall. For the transonic case, a uniform inlet Mach number was prescribed ($Ma = 0.68$). Isentropic boundary conditions of total temperature and total pressure were set at the inlet and static values at the outlet. The height-to-chord ratio of the circular arc is 10% for the transonic case. Figures 1 and 2, respectively, show the Mach number contours from the literature and obtained from simulation. One shock is visible on the lower wall. A comparison between Mach number values on the lower and upper walls from simulation and literature are shown in Fig. 3. Two mesh densities were tested, 80,000 control volumes and a denser mesh consisting of 100,000 control volumes.

The supersonic case was set with a uniform inlet Mach number, $Ma = 1.65$. The height-to-chord ratio of the circular arc is 4%. Figures 4 and 5, respectively, show the Mach number contours from the literature and obtained from the simulation. Two shocks can be observed on the lower wall: one as the flow reaches the bump and one

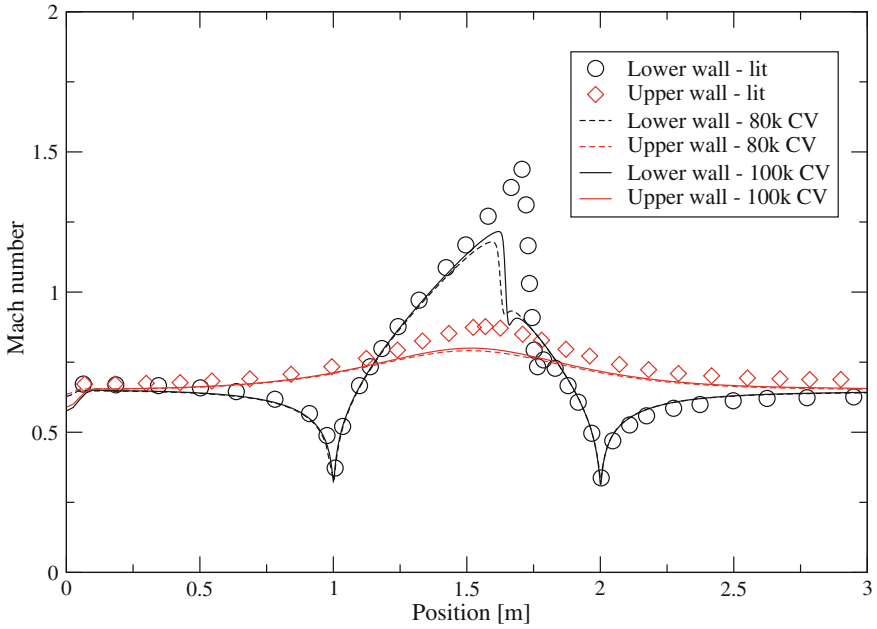


Fig. 3 Comparison of Mach number values on the upper and lower wall for transonic flow

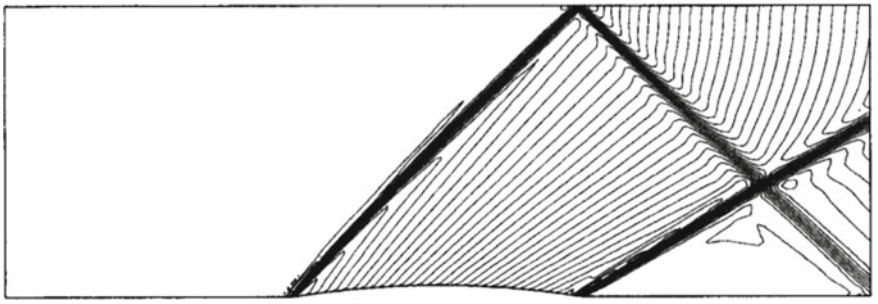


Fig. 4 Mach number contours from the literature for supersonic flow [6]

at the end of the bump. The first shock reflects from the upper wall and crosses the other shock wave. A comparison between the Mach numbers on the upper and lower walls with data from the literature is shown in Fig. 6. The results are shown for two mesh densities: 87,000 control volumes and 200,000 control volumes.

In both the transonic and supersonic cases, the Mach number contours agree well with data from the literature. However, some improvements can be observed with mesh refinement in the transonic flow case: the shock becomes more asymmetrical with the increase in mesh density, and it becomes more similar to Fig. 1. This can also be observed in Fig. 3. It was concluded that further mesh refinement and

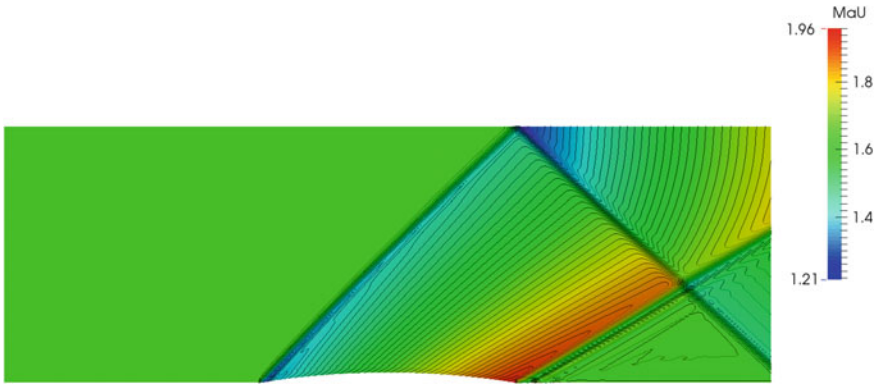


Fig. 5 Calculated Mach number contours for supersonic flow

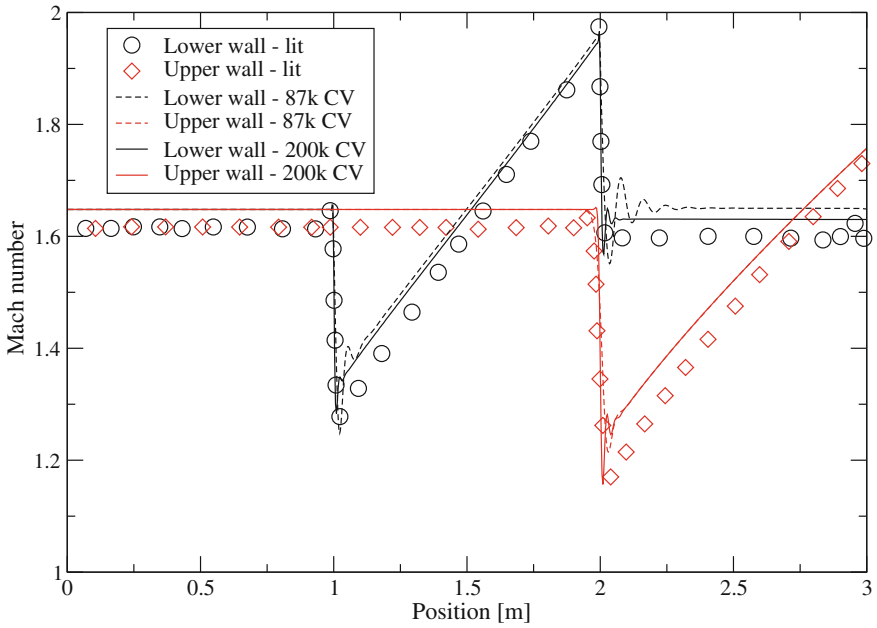


Fig. 6 Comparison of Mach number values on the upper and lower wall for supersonic flow

higher order accurate convection schemes would help in resolving the shock position and steepness. This was confirmed with the supersonic flow case Fig. 6. Mach number oscillations can be observed on the coarser mesh, and the oscillations were reduced with mesh refinement in combination with second-order accurate convection schemes, which contributed to the shock being captured more sharply.

3.2 Validation and Benchmarking of the Incompressible Coupled Solver

The incompressible coupled formulation has been validated for two test cases from the automotive industry. New regulations, which were recently introduced in Formula 1 [1], limit the testing in wind tunnels, as well as CPU time spent for CFD simulations. The coupled formulation can minimise the CPU time while preserving the stability and robustness of the solver, which makes it attractive to such industries. The cases selected for validation represent the most important components of a Formula 1 racing car—the diffuser that generates the negative lift (downforce), and a rotating wheel in contact with the ground that has massive effects on the overall aerodynamic design of the car.

The first case is a bluff body equipped with an upswept aft section, which operates in close proximity to the ground, [17]. The upswept surface of the body and the ground form an expansion/diffuser. This is used to increase the negative lift (downforce). The distance from the ground (ride height) is equal to 40 mm and the free stream velocity is 20 m/s. The half-width of the diffuser is $d = 157$ mm, length $l = 1315$ mm, height $h = 326$ mm and the diffuser angle is 17°.

Two mesh densities were tested, an unstructured mesh with 2.3 million cells and one with 9.5 million cells. Turbulence was calculated in a segregated manner, using the k-ω SST model [12] with wall functions. Turbulence intensity at the inlet was set to 2% and values of the turbulence kinetic energy and dissipation were calculated with a turbulent viscosity ratio equal to 10.

The lift coefficient obtained from experiments is equal to -1.90, while the drag coefficient varies depending on the source, from 0.49 [17] to 0.53 [16]. A comparison of the force coefficients obtained from the simulation is shown in Table 1.

A comparison of the pressure coefficient corresponding to four measurement planes on the upswept surface is shown in Fig. 7. The locations of the measurement planes are nondimensional, calculated as x/d, where x is the distance from the tip of the bluff body in the x-direction (lengthwise) and d the half-width of the diffuser. The results from the coarse mesh better coincide with experimental data. It was found that $y+$ values on the upswept surface influence the values of the pressure coefficient. The best agreement was obtained for larger $y+$ values, ranging from 12 to 30.

Table 1 Comparison of the force coefficients for the bluff body case

ID	C_d	Rel. error (%)	C_l	Rel. error (%)
Experiment [17]	0.49	–	–1.90	–
Simulation (2.3 m CV)	0.52	+6.1	−1.89	+0.5
Simulation (9.5 m CV)	0.47	−4.1	−1.88	+1.1

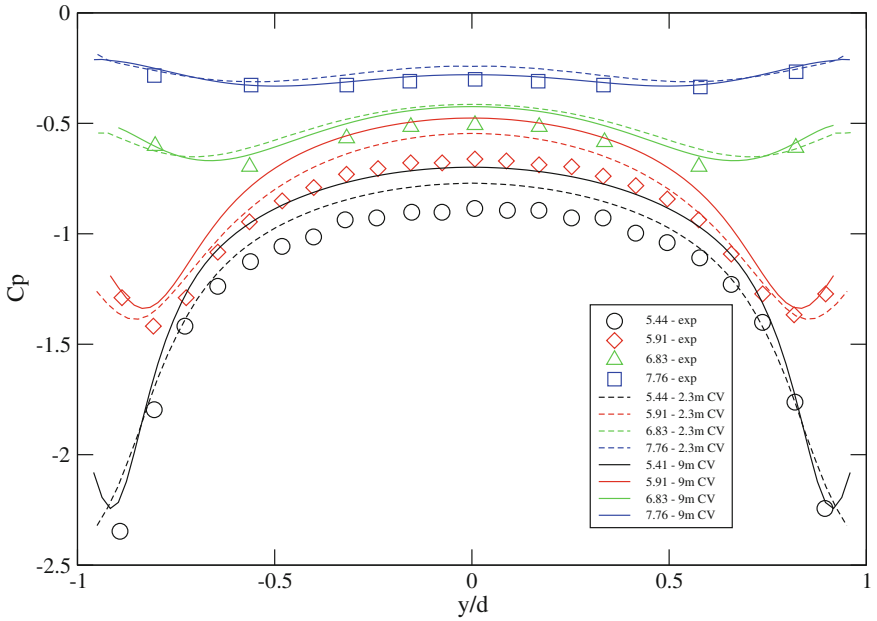


Fig. 7 Pressure coefficient distribution on the corresponding measurement planes of the bluff body: $x/d = 5.44$, $x/d = 5.91$, $x/d = 6.83$, $x/d = 7.76$

The second case is a rotating wheel in contact with the ground. The diameter of the wheel is $d = 314$ mm, width $w = 173$ mm and the camber angle is 2.4°. The computational mesh is block-structured with 11.7 million cells. The freestream velocity is 30 m/s and the rotation of the wheel is modelled using the Multiple Reference Frames (MRF) approach [9].

The experimental value of the drag coefficient is 0.598 [5], while the calculated drag coefficient is equal to 0.60 (+0.3% relative error). Flow characteristics were compared to descriptions found in the literature [5]:

> (The near wake of the tire) is dominated by two large counterrotating vortices. Looking from the back of the wind tunnel, the left vortex is larger and more persistent than the right vortex, and this is due to the combined effect of the wheel camber angle and strut.
> ...
> (There is) a region of strong downward velocity between the vortex cores in the centerplane of the tire.

These flow features can be observed in the results of the simulation Figs. 8 and 9.

For all cases, an algebraic multigrid [18] linear solver was used for the coupled variables. The solver converges in fewer iterations than its segregated counterpart Fig. 11. The coupled solver allows higher underrelaxation factors for velocity and no underrelaxation for pressure, as the connection between velocity and pressure is linear. The coupled solver converges in fewer iterations, but each iteration is more computationally expensive than the segregated solver, which results in similar CPU

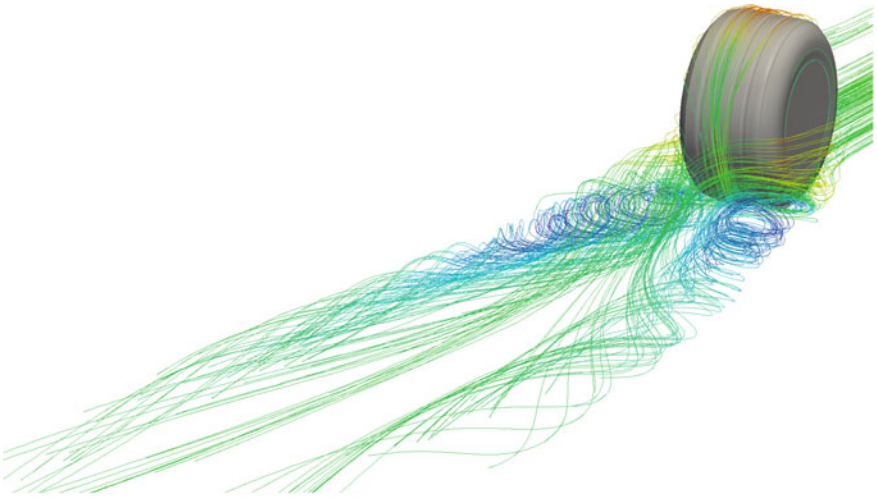


Fig. 8 Streamlines illustrating the flow around a rotating wheel: two counterrotating vortices and a strong downward velocity between the vortex cores

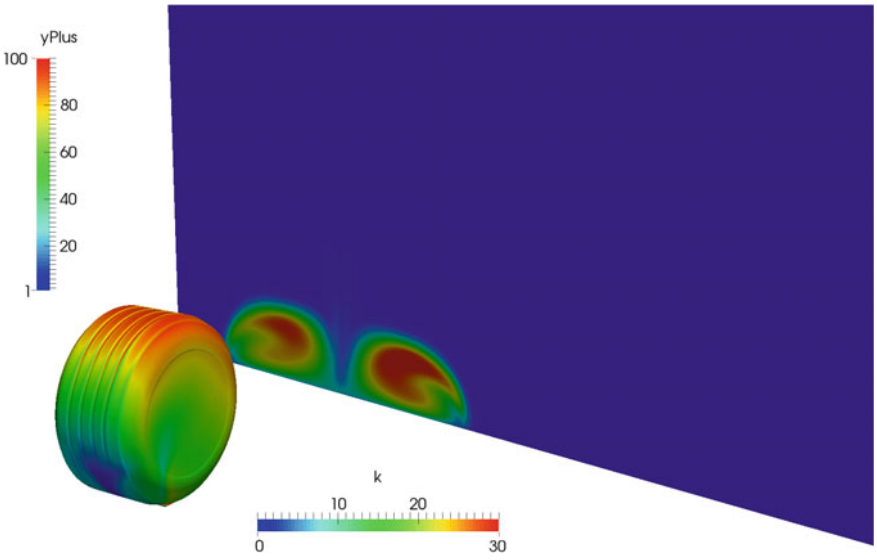


Fig. 9 Slice behind the wheel showing the turbulent kinetic energy of the two vortices. Looking from the back, the left vortex is stronger

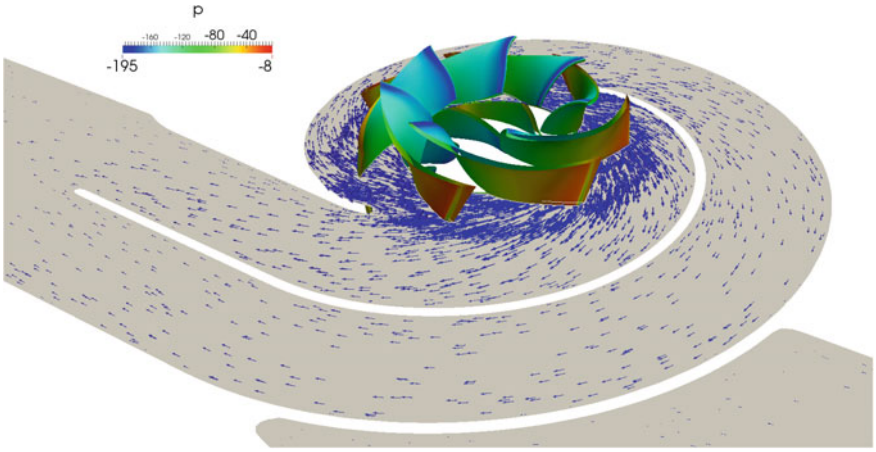


Fig. 10 Centrifugal pump: pressure distribution on the rotor and flow direction

times for larger cases. It was also noticed that the algebraic multigrid solver improves the convergence of the coupled solver, but is slower per iteration compared to Krylov subspace solvers (BiCGStab).

The incompressible coupled solver was extended for turbomachinery applications by adding the Generalised Grid Interface (GGI) boundary condition [2], and MRF zones [9]. The MRF approach is used to simulate a fixed position of the rotor, which is computationally less demanding than the transient simulation with the moving rotor mesh. The model was used for the simulation of a centrifugal pump. In Fig. 10, components of the pump can be seen: inlet pipe, spiral channel, diagonal rotor (inlet is axial, outlet is radial with an axial component) with seven blades and outlet pipe. The outer diameter of the rotor is 300 mm, blade height is 86 mm and the operating speed is 1300 rpm. GGI interface was used for communication between the rotor and stator because the meshes for these components were made separately.

The results of the steady-state simulation were compared to results of the transient simulation [7]: the relative error calculated for the pump head equals 5.6%, for efficiency η 1.41%, and for power 7.2%. Figure 11 shows the comparison of convergence: the coupled solver converges in 400 iterations, while the segregated solver solution still oscillates after 1000 iterations. The convergence of the pump head value is shown in Fig. 12: the behaviour demonstrates the instability of the pressure equation solution of the segregated solver and the stable convergence of the coupled solver (Table 2).

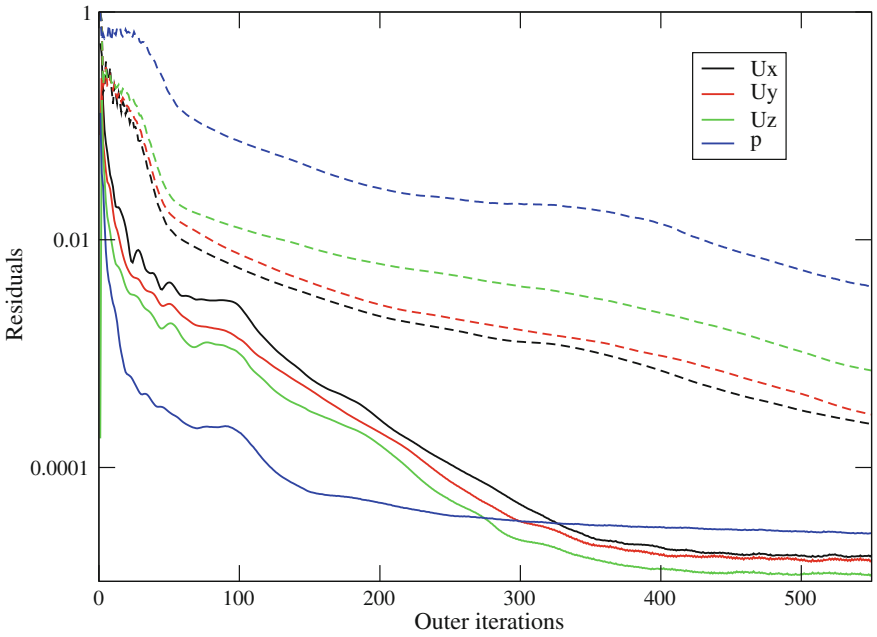


Fig. 11 Convergence of the centrifugal pump case. The solid line corresponds to the coupled solver, the dashed line to the segregated

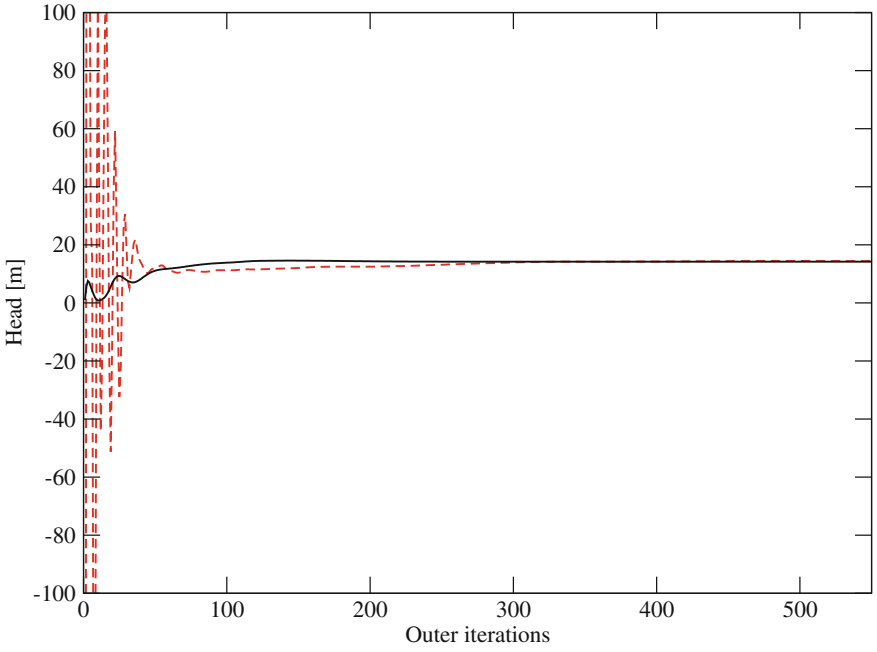


Fig. 12 Convergence of the centrifugal pump head. The solid line corresponds to the coupled solver, the dashed line to the segregated

Table 2 Centrifugal pump simulation data

Mesh	Cell count		
Hybrid: block-structured rotor, unstructured stator	9,054,517		
Coupled solver		Segregated solver	
Underrelax	Iterations	Underrelax	Iterations
$p \rightarrow 1$	400	$p \rightarrow 0.3$	>1000
$\mathbf{u} \rightarrow 0.8$		$\mathbf{u} \rightarrow 0.7$	

4 Conclusion

An implicitly coupled formulation of the pressure–velocity system for incompressible and compressible flow is presented. A new formulation of the compressible pressure-based solver is proposed by assuming an isentropic compression/expansion, resulting in consistent reduction of the compressible form to the incompressible form.

The compressible formulation of the implicitly coupled solver was validated for two test cases and compared to results from the literature. The results of the simulations coincided well with available data, but mesh refinement and higher order accurate numerical schemes have a significant impact on the position and steepness of shock waves.

The incompressible formulation was validated for two industrial test cases and showed good agreement with experimental data. Furthermore, the solver was benchmarked against a segregated formulation. The results have shown that the coupled solver converges in fewer iterations, but at a higher computational cost per iteration. The most notable improvement can be observed in the convergence and stability of the pressure equation. It was noticed that multigrid algorithms contribute to faster convergence and stability of the coupled algorithm.

The bottleneck of the coupled algorithm is the memory requirements for solving all equations simultaneously and, in some cases, the properties of the matrix, which require special preconditioning techniques.

References

1. 2016 Formula One Sporting Regulations. Federation Internationale de l'Automobile http://www.fia.com/regulation/category/110. Cited 29 Sep 2016
2. Beaudoin, M., Jasak, H.: Development of a general grid interface for turbomachinery simulations with OpenFOAM. Proceedings of Open Source CFD International Conference, Berlin (2008)
3. Darwish, M., Sraj, I., Moukalled, F.: A coupled finite volume solver for the solution of incompressible flows on unstructured grids. Journal of Computational Physics 228, 180–201 (2009)

4. Darwish, M., Moukalled, F.: A fully coupled Navier-Stokes solver for fluid flow at all speeds. Numerical Heat Transfer Fundamentals 65, 410–444 (2014)
5. Fackrell, J.E.: The aerodynamics of an isolated wheel rotating in contact with the ground. Faculty of Engineering, University of London, London (1974)
6. Ferziger, J.H., Peric, M.: Computational methods for fluid dynamics. Springer, Germany (2002)
7. Horvat, M.: A study of rotor-stator interaction models for numerical performance prediction of centrifugal pumps. Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Zagreb (2016)
8. Issa, R.I.: Solution of the implicitly discretized fluid flow equations by operator-splitting. Journal of Computational Physics 62, 40–65 (1986)
9. Jasak, H., Beaudoin, M.: OpenFOAM turbo tools: from general purpose CFD to turbomachinery simulations. Proceedings of ASME-JSME-KSME Joint Fluid Engineering Conference, Hamamatsu (2011)
10. Mazhar, Z.: A procedure for the treatment of the velocity-pressure coupling problem in incompressible fluid flow. Numerical Heat Transfer 39, 91–100 (2001)
11. Mazhar, Z.: An enhancement to the block implicit procedure for the treatment of the velocity-pressure coupling problem in incompressible fluid flow. Numerical Heat Transfer 41, 493–500 (2002)
12. Menter, F.R.: Zonal two equation k-ω turbulence models for aerodynamic flows. Proceedings of 24th Fluid Dynamics Conference, Orlando (1993)
13. Patankar, S.V., Spalding, D.B.: A calculation procedure for heat, mass and momentum transfer in three dimensional parabolic flows. International Journal of Heat and Mass Transfer 15, 1787–1806 (1972)
14. Raithby, G.D., Schneider, G.E.: Numerical solution of problems in incompressible fluid flow: treatment of the velocity-pressure coupling. Numerical Heat Transfer 2, 417–440 (1979)
15. Rhie, C.M., Chow, W.L.: A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. AIAA Journal 21, 1525–1532 (1983)
16. Ruhrmann, A., Zhang, X.: Influence of diffuser angle on a bluff body in ground effect. Journal of Fluids Engineering 125, 332–338 (2003)
17. Senior, A.E.: The aerodynamics of a diffuser equipped bluff body in ground effect. School of Engineering Sciences, University of Southampton, Southampton (2002)
18. Trottenberg, U., Oosterlee, C., Schueller, A.: Multigrid. Academic Press, London (2001)
19. Zedan, M., Schneider, G.E.: A strongly implicit simultaneous variable solution procedure for velocity and pressure in fluid flow problems. 18th Thermophysics Conference, Montreal, Canada (1983)
20. Zedan, M., Schneider, G.E.: A coupled strongly implicit procedure for velocity and pressure computation in fluid flow problems. Numerical Heat Transfer 8, 537–557 (1985)

Improving the Numerical Stability of Steady-State Differential Viscoelastic Flow Solvers in OpenFOAM®

Célio Fernandes, Manoel S. B. Araujo, Luís L. Ferrás and J. Miguel Nóbrega

Abstract This work reports the developments made in improving the numerical stability of the viscoelastic solvers available in the open-source finite volume computational library $OpenFOAM^{®}$. For this purpose, we modify the usual both-side diffusion (BSD) technique, using a new approach to discretize the explicit diffusion operator. Calculations performed with the new solver, for two benchmark 2D case studies of an upper-convected Maxwell (UCM) fluid, are presented and compared with literature results, namely the 4:1 planar contraction flow and the flow around a confined cylinder. In the 4:1 planar contraction flow, the corner vortex size predictions agree well with the literature, and a relative error below 5.3% is obtained for $De \leq 5$. In the flow around a confined cylinder, the predictions of the drag coefficient on the cylinder are similar to reference data, with a relative error below 0.16% for $De \leq 0.9$.

1 Introduction

Viscoelastic fluids are present in many engineering applications, namely in the form of polymer solutions, gels, surfactants, emulsions, or colloidal solutions. To describe the rheological behavior of the material as a relation between the stress and deformation history, suitable constitutive equations must be used. These equations are

C. Fernandes · L. L. Ferrás · J. M. Nóbrega (✉)
Institute for Polymers and Composites/i3N, University of Minho,
Campus de Azurém, 4800-058 Guimarães, Portugal
e-mail: mnobrega@dep.uminho.pt

C. Fernandes
e-mail: cbpf@dep.uminho.pt

L. L. Ferrás
e-mail: luis.ferras@dep.uminho.pt

M. S. B. Araujo
Instituto de Ciências Exatas e Naturais, Faculdade de Matemática,
Universidade Federal do Pará, 66075-110 Belem, PA, Brazil
e-mail: silvino@ufpa.br

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_20

usually of differential or integral form, and can be employed in numerical codes to describe the flow of viscoelastic fluids.

The use of computational fluid dynamics (CFD) has increased significantly over the recent decades, mainly due to the development of better and faster computers that allow for the use of more realistic models, and due to the development of more accurate and efficient numerical methods. One of the more widely used approaches is the finite volume method (FVM), which can be advantageous in terms of computer memory and time requirements [1], as well as in terms of numerical stability. The use of simple constitutive equations, such as the upper-convected Maxwell (UCM) and Oldroyd-B models [2] or the White–Metzner [3] model, is very challenging from the numerical point of view. Some characteristic difficulties found when using these models are associated with stress singularities near sharp corners or at stagnation points. Hence, these constitutive models are widely employed to test the accuracy and robustness of new numerical methods for viscoelastic fluid flows. In regard to that which concerns to the development of viscoelastic solvers in the $OpenFOAM^{®}$ computational library, [4] implemented several viscoelastic differential models, using the discrete elastic-viscous stress-splitting formulation. The major drawback of the numerical algorithm implemented by [4] is that it is prone to developing numerical instabilities, when there is no solvent viscosity, as it happens in the UCM model. More recently, [5] presented a new formulation for the discretization of the divergence of the viscoelastic stress that allows for a semi-implicit handling of the constitutive equation and improves the coupling between the velocity and stress fields. The verification was done using planar and square–square contraction flows of a simplified Phan–Thien–Tanner (SPTT) fluid.

In this work, we propose a simple methodology for improving the stability of the *viscoelasticFluidFoam* solver available in $OpenFOAM^{®}$, aiming to increase the numerical stability when dealing with complex fluid flows, as is the case with the UCM model. In the context of the FVM approach, the newly developed methodology uses a special second-order discretization operator for the explicit term added by the both-side diffusion technique, in contrast to the usual Laplacian operator. To verify the developed code, the results are compared with the benchmark problems of the flow in a 4:1 planar sudden contraction and the flow around a confined cylinder.

The remaining sections of this chapter are organized as follows: Sect. 2 presents the governing equations to be solved and the numerical procedure adopted. In Sect. 3, the results obtained with the newly developed numerical code for the two benchmark case studies are presented, discussed, and compared with results from the literature. The chapter ends with the main conclusions.

2 Governing Equations and Numerical Procedure

This study concerns the incompressible flow of a viscoelastic fluid, following the UCM model, and hence the governing equations that have to be solved are those of conservation of mass,

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

and of momentum,

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau}_P, \tag{2}$$

together with an appropriate constitutive equation for the polymeric extra stress tensor $\boldsymbol{\tau}_P$. In the previous equations, $\mathbf{u}$ is the velocity vector, ρ the fluid density, t the time, and p the pressure. In the current study, the polymeric extra stress tensor $\boldsymbol{\tau}_P$ is computed using the UCM differential model as

$$\boldsymbol{\tau}_P + \lambda \left(\frac{\partial \boldsymbol{\tau}_P}{\partial t} + \nabla \cdot (\mathbf{u}\boldsymbol{\tau}_P) \right) = \eta_P \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) + \lambda \left((\nabla \mathbf{u})^T \cdot \boldsymbol{\tau}_P + \boldsymbol{\tau}_P \cdot \nabla \mathbf{u} \right), \tag{3}$$

where λ is the fluid relaxation time and η_P the polymer viscosity. Notice that, the UCM model simplifies to the Newtonian fluid model when $\lambda = 0$.

In order to increase the stability of the numerical method, namely to avoid the decoupling between velocity and extra stress tensor fields, an improved both-side diffusion technique (BSD) was newly implemented in the $OpenFOAM^{®}$ computational library. Inspired by the work of [6] for the finite element method (FEM), an additional diffusive term proportional to η_P is introduced on both sides of the momentum equation, Eq. (2), to obtain

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) - \eta_P \nabla^2 \mathbf{u} = -\nabla p + \nabla \cdot \boldsymbol{\tau}_P - \eta_P \nabla \cdot (\nabla \mathbf{u}), \tag{4}$$

in which the terms on the left-hand side are discretized implicitly (incorporated into the coefficients of the algebraic equations) and those on the right-hand side are discretized explicitly (incorporated into the source term of the algebraic equations). The improvement of our BSD technique is the special second-order discretization applied to the explicit diffusion operator, $\nabla \cdot (\nabla \mathbf{u})$, added by the BSD technique on the right-hand side of Eq. (4), instead of the usual Laplacian, $\nabla^2 \mathbf{u}$, discretization.

For illustration purposes, Fig. 1 shows the solution of an UCM fluid in a 4:1 contraction flow, obtained before and after the code modifications. As can be easily concluded from those results, the new treatment of the explicit diffusion term, assured the coupling between stress and velocity fields and consequently, the prediction's accuracy.

The finite volume method was used to solve the governing equations. Hence, the governing equations are integrated in space over control volumes (cells) forming the computational mesh, and in time over a time step Δt, so that sets of linearized algebraic equations are obtained. The linear sets of equations are solved initially for velocity and pressure fields, and subsequently for the extra stress tensor. The SIMPLE algorithm [7] is used to obtain a velocity field satisfying the continuity equation. To

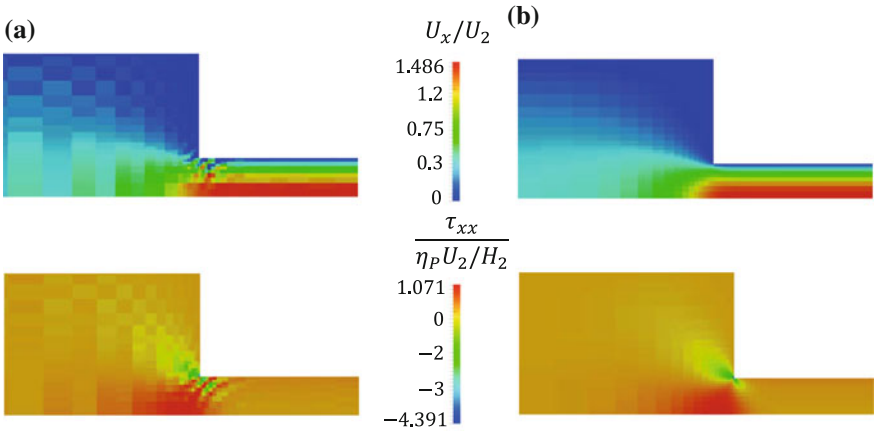


Fig. 1 Normalized velocity and stress profiles obtained for the 4:1 contraction flow of an UCM fluid with the **a** original and **b** developed formulations. H_2 and U_2 are the downstream channel height and average velocity, respectively

discretize the convective fluxes, the method uses the MINMOD and SMART high-resolution schemes, as in the works of [8, 9]. For the time-step discretization, the implicit first-order Euler method was used, since only steady-state solutions are of interest for the case studies addressed in this work. Notice that, the transient term used in the momentum equation is only retained for time marching purposes, but it vanishes when steady-state conditions are achieved. The Poisson-type equation for pressure is solved with a conjugate gradient method with a Cholesky preconditioner and the velocity and stress linear systems are solved using BiCGstab with an incomplete lower-upper (ILU) preconditioning [10–12]. The absolute tolerance for pressure, velocity and stress fields was set as 10^{-20}. Under-relaxation is applied with a value of 0.3 for pressure and stress and 0.7 for velocity.

3 Case Studies

3.1 Flow in a 4:1 Planar Sudden Contraction

A planar sudden contraction with contraction ratio H_1/H_2 of 4:1 (upstream thickness of $2H_1 = 4$ cm and downstream thickness of $2H_2 = 1$ cm) was chosen as the first test geometry (Fig. 2), because of the availability of benchmark numerical data in the literature [8]. The flow is assumed to have a plane of symmetry along the centerline ($y = 0$), and, thus, only half of the domain is considered for modeling purposes. The figure shows the five structured blocks used to generate the five consecutively refined meshes, which will be considered in the mesh refinement study that follows.

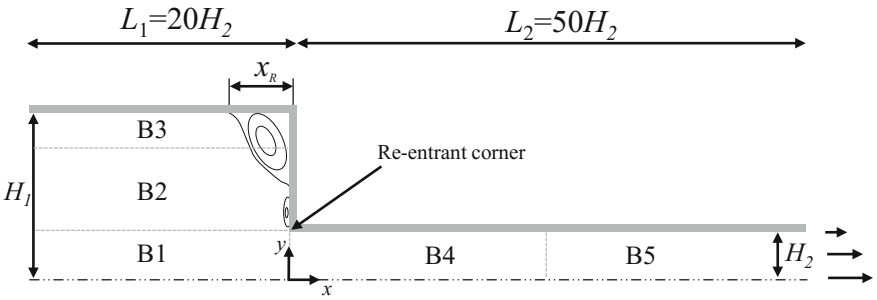


Fig. 2 Schematic representation of the 4:1 planar contraction: geometry and blocks used to generate the mesh

Table 1 Characteristics of the five meshes used for the mesh convergence analysis in the 4:1 planar sudden contraction flow

Block	Mesh 1			Mesh 5		
	NX × NY	f_x	f_y	NX × NY	f_x	f_y
Block I	24 × 10	0.8210	0.8475	376 × 160	0.9877	0.9897
Block II	24 × 13	0.8210	1.2091	376 × 200	0.9877	1.0119
Block III	24 × 5	0.8210	0.7384	376 × 68	0.9877	0.9812
Block IV	20 × 10	1.2179	0.8475	320 × 160	1.0124	0.9897
Block V	7 × 10	1.3782	0.8475	100 × 160	1.0202	0.9897
NC	942			228,128		
$\delta x_{\min} = \delta y_{\min}$	$0.04H_2$				$0.0025H_2$	

NX and NY are the number of cells along the x and y directions, respectively, for each block
f_x and f_y are the expansion/contraction ratios for each block
NC is the number of cells for each mesh
$\delta x_{\min}$ and $\delta y_{\min}$ are the minimum cell size in each direction

The characteristics of the coarsest and finest hexahedral meshes used are presented in Table 1. The expansion or contraction geometrical factors are defined for each direction as the ratio of the lengths of two consecutive cells ($f_x = \delta_{x_{i+1}}/\delta_{x_i}$ with δ_{x_i} being the length of the cell i in the x-direction). With this procedure, the minimum normalized cell size at the corner was $\delta x_{\min}/H_2 = \delta y_{\min}/H_2 = 0.0025$ for the finest mesh, Mesh 5. For each mesh refinement, the number of cells along each direction (NX and NY) was doubled, and the corresponding expansion/contraction ratios (f_x and f_y) inside each subblock were root-squared.

The Reynolds and Deborah numbers are defined on the basis of downstream channel quantities

$$Re = \frac{\rho U_2 H_2}{\eta_P}, \tag{5}$$

$$De = \frac{\lambda U_2}{H_2}, \tag{6}$$

and Re was fixed at 0.01 (representative of creeping flow) while De was varied. The downstream velocity U_2 was fixed as 50 cm/s, the fluid density ρ as 100 kg/m^3, and the polymer viscosity η_P as 0.25 Pa.s. The Deborah number was varied by changing the parameter λ.

The boundary conditions employed in this case study are as follows. At the inlet boundary, the streamwise velocity and stress components were set as equal to fully developed profiles, and a zero gradient was imposed for pressure. At the outlet boundary, zero gradients are assumed for the velocity and stress components and the pressure value was set to zero. Across the symmetry plane, the convective and diffusive fluxes are forced to vanish for all variables. At the walls' boundary, the usual zero no-slip condition was applied to the velocity field, the zero gradient for the pressure and the extra-stress tensor components were linearly extrapolated to the wall face.

The quantitative comparison of the size of the corner vortex is made by measuring its dimensionless length $X_R = x_R/H_2$ (see Fig. 2). The mesh refinement technique allowed us to apply Richardson's extrapolation for the X_R value, using the three finest meshes. Table 2 compares the results obtained with the developed code and the ones provided by [8]. The X_R results in the most refined mesh (Mesh 5), obtained with the developed code, have a difference below 5.3% when compared to the extrapolated

Table 2 Dimensionless length of primary vortex (X_R) as a function of the Deborah number and the mesh for UCM fluid. Comparison between the developed code and the results from [8]

Developed code							
De	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5	Extrapolated	Difference (%)[a]
0	1.436	1.475	1.477	1.479	1.479	1.479	0.0003
1	1.374	1.378	1.349	1.330	1.321	1.314	0.6
2	1.301	1.285	1.176	1.113	1.083	1.056	2.5
3	1.305	1.290	1.054	0.928	0.881	0.854	3.2
4	1.402	1.396	1.014	0.803	0.735	0.702	4.7
5	1.530	1.524	1.037	0.709	0.622	0.591	5.3
[8]							
De	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Extrapolated	Difference (%)[b]	
0	1.472	1.488	1.494	1.495	1.496	0.1	
1	1.349	1.371	1.349	1.339	1.335	0.3	
2	1.631	1.259	1.154	1.118	1.105	1.2	
3	1.517	1.266	1.014	0.946	0.923	2.5	
4	1.644	1.337	0.987	c	0.87	13.4	
5	1.687	1.517	1.127	c	0.997	13	

[a]Calculated between Mesh 5 and the extrapolated values
[b]Calculated between Mesh 4 (or 3) and the extrapolated values
[c]Convergence criterion not attained (solution oscillates)

value for all De numbers tested. In general, the developed code predictions are aligned with the literature results, but it is worth noting that, for larger De, we were able to clearly improve the accuracy of the results available in the literature.

Figure 3 shows the detailed first normal stress difference behavior and pressure profiles at normalized y/H_2 locations progressively closer to the wall (normalized distances of 0.9875, 0.9925, and 0.9975), obtained in Mesh 5 for $De = 0$, $De = 3$, and $De = 5$. In accordance with [8], the peaks in stress and pressure increase and become narrower as we approach the singular point.

3.2 Flow Around a Confined Cylinder

The second case study refers to the plane flow past a circular cylinder placed at the centerline of a channel (see Fig. 4). The blockage ratio in this study, defined as the ratio of cylinder radius R to channel half height h, is $\beta = 0.5$.

The computational domain extends from $x = -20R$, where a streamwise uniform velocity U is imposed for the Newtonian computations ($\lambda = 0$) and a fully developed profile is imposed for the UCM case, and leaves at $x = 60R$. In this study, the cylinder radius is fixed as $R = 1$ m and the uniform velocity $U = 1$ m/s. Similarly to the first case study, here, the flow is assumed to have a plane of symmetry at $y = 0$, and thus only half of the domain is considered in the numerical studies. Three meshes, with different degrees of refinement, were used for mesh convergence analysis, and are presented in Table 3. The numbers in the mesh names indicate the number of radial cells between the cylinder surface and the channel wall.

In this case study, the Reynolds and Deborah numbers are defined on the basis of the inlet bulk velocity, U, and the cylinder radius, R:

$$Re = \frac{\rho U R}{\eta_P}, \tag{7}$$

$$De = \frac{\lambda U}{R}, \tag{8}$$

and Re was fixed at 0.01 (representative of creeping flow) while De was varied. The inlet bulk velocity was kept constant and equal to 1 m/s, the fluid density ρ as 100 kg/m^3, and the polymer viscosity η_P as 10000 Pa.s. The Deborah number was varied by changing the value of the relaxation time, λ. The boundary conditions employed for this case study are of the same type as those used in the 4:1 planar sudden contraction case study.

The results that are shown in this section are based on the dimensionless drag coefficient C_D, resulting from surface integration of the stress and pressure fields around the cylinder

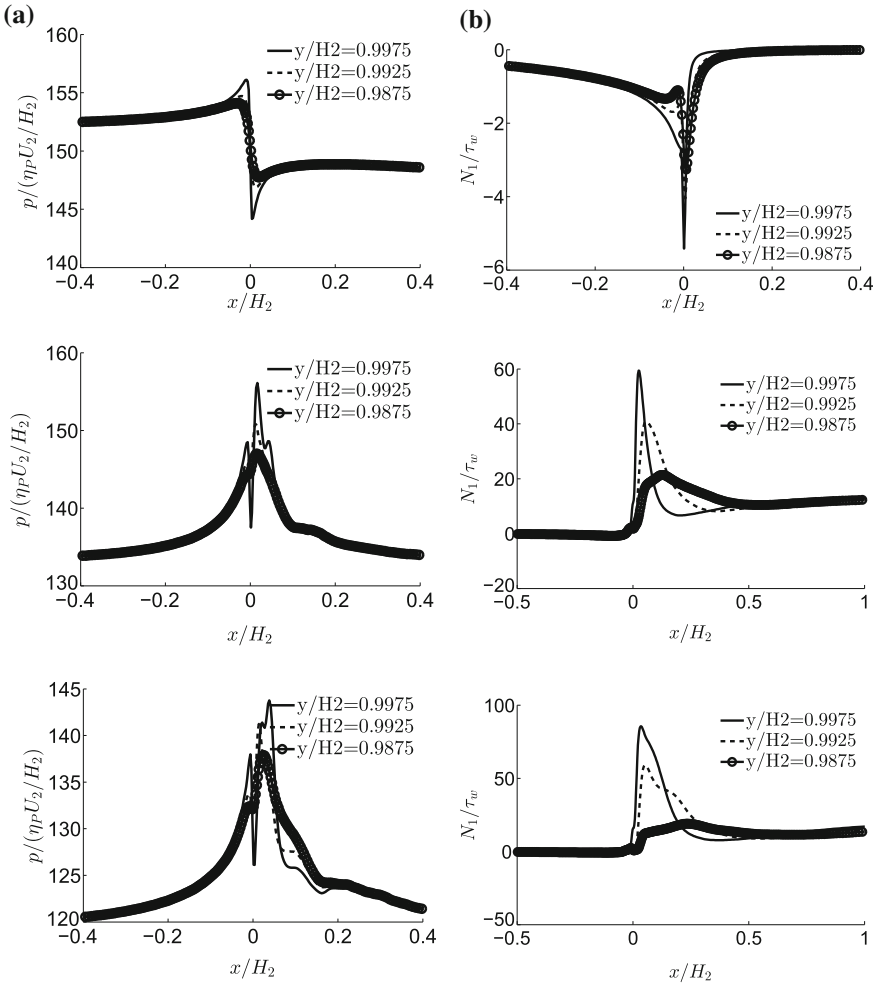


Fig. 3 Longitudinal distributions of **a** pressure and **b** first normal stress difference in the vicinity of the downstream wall and corner, obtained in Mesh 5 for $De = 0$ (top), $De = 3$ (middle) and $De = 5$ (bottom). The first normal stress difference is normalized with $\boldsymbol{\tau}_w = 3\eta_P U_2/H_2$ (for Newtonian fluid flows, η_P is substituted by the Newtonian dynamic viscosity μ)

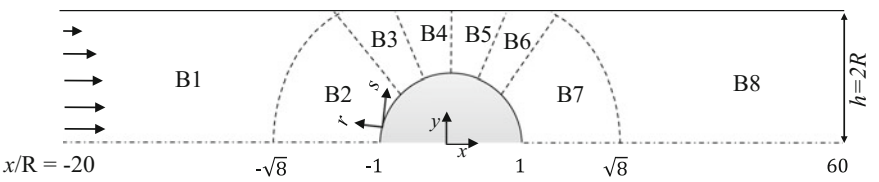


Fig. 4 Schematic representation of the flow past a circular cylinder: geometry and blocks used to generate the mesh

Table 3 Characteristics of the three meshes used for mesh convergence analysis in the flow around a confined cylinder case study

Block	M30		M60		M120	
	NS × NR	f_r	NS × NR	f_r	NS × NR	f_r
Block I	25 × 24	1.1029	50 × 48	1.0502	100 × 96	1.0248
Block II	25 × 30	1.1075	50 × 60	1.0524	100 × 120	1.0259
Block III	13 × 30	1.1075	25 × 60	1.0524	50 × 120	1.0259
Block IV	13 × 30	1.1075	25 × 60	1.0524	50 × 120	1.0259
Block V	13 × 30	1.1075	25 × 60	1.0524	50 × 120	1.0259
Block VI	13 × 30	1.1075	25 × 60	1.0524	50 × 120	1.0259
Block VII	25 × 30	1.1075	50 × 60	1.0524	100 × 120	1.0259
Block VIII	25 × 30	1.1323	50 × 60	1.0641	100 × 120	1.0316
NCV	4, 410		17,400		69, 600	
$(NS)_{tot}$	102		200		400	
$(\Delta r/R)_{min}$	0.00963		0.00481		0.00238	
$(\Delta s/R)_{min}$	0.0302		0.0157		0.00785	

NS, NR: number of cells in the tangential and radial directions
NCV: total number of control volumes
$(NS)_{tot}$: number of tangential cells around the half cylinder
f_r is the expansion/contraction ratio inside each block in the radial direction. For the tangential direction, the mesh is uniform
Δr and Δs are the minimum cell size around the cylinder surface in the radial and tangential directions, respectively

$$C_D = \frac{1}{\eta_P U L} \int_S (\boldsymbol{\tau}_P - p\mathbf{I}) \cdot \mathbf{n} \cdot \mathbf{i} dS, \tag{9}$$

where $\mathbf{I}$ is the unitary tensor, $\mathbf{n}$ the unit normal vector to the cylinder surface S, $\mathbf{i}$ is the unit vector in the $x-$direction, and L is the depth of the cylinder in the neutral direction. A unitary depth was used in the neutral direction.

The C_D values predicted by the developed solver are listed in Table 4, and some data from the literature are given for comparison purposes. Accurate results were obtained with the developed code, as the most refined mesh (M120) produced a C_D value with a difference less than or equal to 0.16% from the extrapolated values for all De numbers.

Normal stress profiles around the cylinder and along the centerline are shown in Fig. 5. It can be seen that two maximums occur, the first one within the thin boundary layer over the cylinder surface and the second in the wake, after the rear stagnation point.

Finally, Fig. 6 shows the pressure variation around the cylinder wall and centerline for different De numbers. As shown, the pressure on the cylinder's rear surface increases smoothly with De, due to the shifted flow pattern. In the rear stagnation point region, for low De numbers, the pressure level increases steadily with De. For a De number between 0.3 and 0.6, the behavior of the pressure distribution reverses,

Table 4 Drag coefficient (C_D) as a function of the Deborah number and the mesh for the UCM fluid. Comparison between the developed code and the results from [9]

Developed code					
De	M30	M60	M120	Extrapolated	Difference (%)[a]
0	131.998	132.397	132.484	132.508	0.02
0.3	109.519	109.005	108.900	108.873	0.03
0.6	94.027	92.962	92.703	92.620	0.09
0.9	89.206	88.128	87.802	87.661	0.16
[9]					
De	M30	M60	M120	Extrapolated	Difference (%)[a]
0	132.23	132.342	132.369	132.378	0.01
0.3	–	108.515	108.614	108.647	0.03
0.6	–	92.277	92.298	92.305	0.01
0.9	–	87.395	87.218	87.16	0.07

[a]Calculated between M120 and extrapolated values

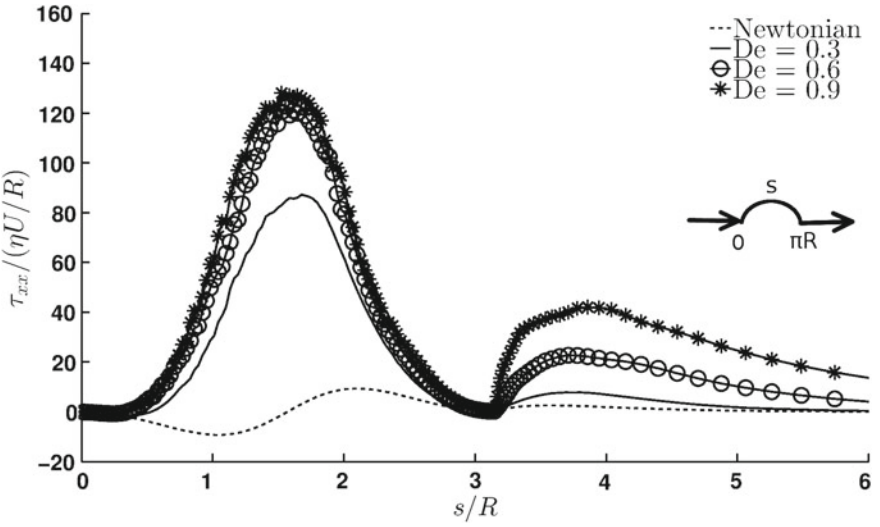


Fig. 5 Profiles of longitudinal normal stress τ_{xx} along the cylinder wall and wake centerline for the UCM fluid at increasing *De* (mesh M120)

and ultimately, a highly negative peak occurs for *De* equal to 0.9. As can be seen in Figs. 5 and 6, some oscillations in the profiles are present, but they can be removed if finer meshes are used.

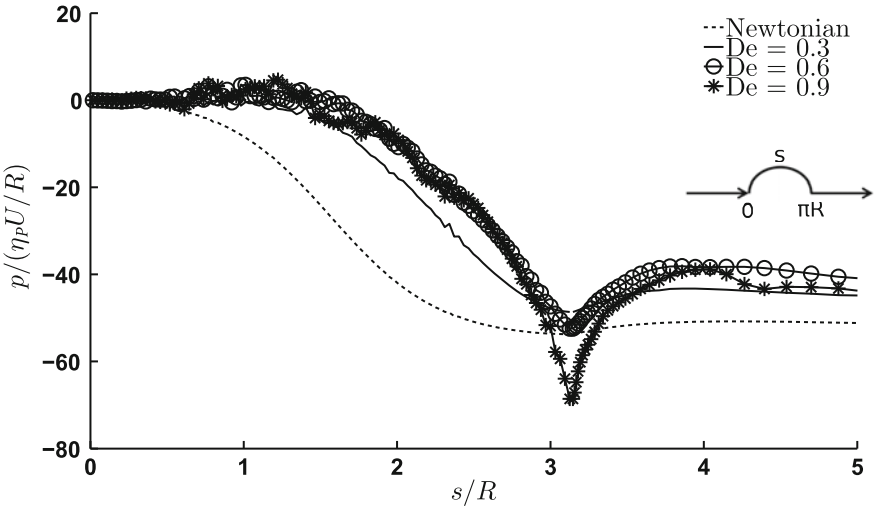


Fig. 6 Profiles of pressure around the cylinder surface and along the wake centerline for UCM fluid at increasing De

4 Conclusions

This work presents a general finite volume methodology (FVM) for the computation of the flow of viscoelastic fluids described by differential-type constitutive equations. The approach was implemented with the open-source computational fluid dynamics library $OpenFOAM^{®}$. The proposed methodology consists of an improvement of the both-side diffusion (BSD) technique widely used in FVM, which comprises modification of the discretization of the explicit diffusion term. The developed approach can be applied to orthogonal or non-orthogonal meshes and was found to accurately predict steady-state solutions of the upper-convected Maxwell (UCM) fluid, which is known to be challenging from a numerical point of view.

The code was verified with two widely used benchmark case studies: the 4:1 planar sudden contraction flow and flow around a confined cylinder. In both cases, the simulations were performed at $Re = 0.01$, representing creeping flow conditions. For the former case, the Deborah number varied from 0 to 5, and for the latter, it was varied from 0 to 0.9. The results obtained in both test cases were accurately predicted, in the sense that the vortex length size and drag coefficient have less than 5.3% and 0.12% of differences, respectively, when compared with the finest mesh result and the extrapolated value.

In summary, the results obtained show that the newly improved viscoelastic solver using an open-source code can accurately predict the flow patterns of the UCM fluid in the two benchmark problems tested.

Acknowledgements This work is funded by FEDER funds through the COMPETE 2020 Programme and National Funds through FCT—Portuguese Foundation for Science and Technology under the project UID/CTM/50025/2013 and under the scholarship SFRH/BPD/100353/2014. The author M.S.B. Araujo acknowledges funding from CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) proc. BEX 1902-14-8. The authors would like to acknowledge the Minho University cluster under the project Search-ON2: Revitalization of HPC infrastructure of UMinho, (NORTE-07-0162-FEDER-000086), co-funded by the North Portugal Regional Operational Programme (ON.2-0 Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF).

References

1. H. Versteeg and W. Malalasekra. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method Approach*. Prentice Hall, London, 1996.
2. J.G. Oldroyd. On the formulation of rheological equations of state. *Proc. Roy. Soc. A*, 200: 523–541, 1950.
3. J.L. White and A.B. Metzner. Development of constitutive equations for polymeric melts and solutions. *J. App. Polym. Sci.*, 7: 1867–1889, 1963.
4. J.L. Favero, A.R. Secchi, N.S.M. Cardozo, and H. Jasak. Viscoelastic flow analysis using the software OpenFOAM and differential constitutive equations. *J. Non-Newton. Fluid Mech.*, 165: 1625–1636, 2010.
5. F. Habla, A. Obermeier, and O. Hinrichsen. Semi-implicit stress formulation for viscoelastic models: Application to three-dimensional contraction flows. *J. Non-Newton. Fluid Mech.*, 199: 70–79, 2013.
6. R. Guénette and M. Fortin. A new mixed finite element method for computing viscoelastic flows. *J. Non-Newton. Fluid Mech.*, 60: 27–52, 1995.
7. S.V. Patankar and D.B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. Heat Mass Transf.*, 115: 1787–1803, 1972.
8. M.A. Alves, F.T. Pinho, and P.J. Oliveira. Effect of a high-resolution differencing scheme on finite-volume predictions of viscoelastic flows. *J. Non-Newtonian Fluid Mech.*, 93: 287–314, 2000.
9. M.A. Alves, F.T. Pinho, and P.J. Oliveira. The flow of viscoelastic fluids past a cylinder: finite-volume high-resolution methods. *J. Non-Newtonian Fluid Mech.*, 97: 207–232, 2001.
10. J. Lee, J. Zhang, and C.-C. Lu. Incomplete LU preconditioning for large scale dense complex linear systems from electromagnetic wave scattering problems. *J. Non-Newtonian Fluid Mech.*, 185: 158–175, 2003.
11. D.A.H. Jacobs. Preconditioned conjugate gradient methods for solving systems of algebraic equations. Technical Report RD/L/N193/80, Central Electricity Research Laboratories, 1980.
12. M.A. Ajiz and A. Jennings. A robust incomplete Cholesky-conjugate gradient algorithm. *J. Numer. Meth. Eng.*, 20: 949–966, 1984.

IsoAdvector: Geometric VOF on General Meshes

Johan Roenby, Henrik Bredmose and Hrvoje Jasak

Abstract In a recent publication, we presented a novel geometric VOF interface advection algorithm, denoted isoAdvector (Roenby et al. in R Soc Open Sci 3:160405 2016, [1]). The OpenFOAM® implementation of the method was publicly released to allow for more accurate and efficient two-phase flow simulations in OpenFOAM® (Roenby in isoAdvector www.github.com/isoadvector, [2]). In the present paper, we give a brief outline of the isoAdvector method and test it with two pure advection cases. We show how to modify interFoam so as to use isoAdvector as an alternative to the currently implemented MULES limited interface compression method. The properties of the new solver are tested with two simple interfacial flow cases, namely the damBreak case and a steady stream function wave. We find that the new solver is superior at keeping the interface sharp, but also that the sharper interface exacerbates the well-known spurious velocities in the air phase close to an air–water interface. To fully benefit from the accuracy of isoAdvector, there is a need to modify the pressure–velocity coupling algorithm of interFoam, so it more consistently takes into account the jump in fluid density at the interface. In our future research, we aim to solve this problem by exploiting the subcell information provided by isoAdvector.

J. Roenby (✉)
Stromning, Luftmarinegade 62, 1432 København K, Denmark
e-mail: johan@stromning.com

J. Roenby
DHI, Agern Alle 5, 2970 Hørsholmh, Denmark

H. Bredmose
DTU Wind Energy, Nils Koppels Alle, 2800 Kgs. Lyngby, Denmark
e-mail: hbre@dtu.dk

H. Jasak
Faculty of Mechanical Engineering and Naval Architecture,
Hrvoje Jasak University of Zagreb, Ivana Lucica 5, Zagreb, Croatia
e-mail: hrvoje.jasak@fsb.hr

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_21

1 The Interfacial Flow Equations

We start by writing the equations of motion governing the flow of two incompressible, immiscible fluids. To keep things simple, we will ignore viscous effects and surface tension. What remains are the passive advection equation,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{1}$$

the incompressibility equation,

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

and the Euler equations,

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \rho \mathbf{g}. \tag{3}$$

Here, ρ is the fluid density field taking the constant value, ρ_1, in the reference fluid and the constant value, ρ_2, in the other fluid, $\mathbf{u}$ is the velocity field, p is the fluid pressure and $\mathbf{g}$ is the constant downward pointing gravity vector. In the interFoam solver of OpenFOAM®, these equations are discretized in the finite volume framework and advanced in time in a segregated manner. Within a time step, Eq. 1 is used to update the density field in time, followed by a procedure for solving Eqs. 2 and 3 to update the pressure and velocity field in time. The details of the implementation are well-described in the paper [3], which also gives an overview of the challenge faced by the interfacial CFD community in keeping the density field sharp and bounded, with spurious velocities at the interface, and with handling of large density ratios. The development of the isoAdvector interface advection method is a first step in our efforts to solve these problems and increase the general performance and accuracy of interfacial flow simulations. In the following, we will briefly explain how isoAdvector works.

2 IsoAdvector for Interface Advection

The basic equation that we will solve is Eq. 1, recast in the volume-of-fluid formulation. For this recasting, we need a number of definitions: First, we divide the computational domain into cells, $\mathscr{C}_1, \mathscr{C}_2, \ldots$, and define the notation for the cell-averaged value of a field, $f(\mathbf{x}, t)$, at time t,

$$\langle f \rangle_i(t) \equiv \frac{1}{V_i} \int_{\mathscr{C}_i} f(\mathbf{x}, t) dV, \tag{4}$$

where V_i is the volume of cell i. Defining the indicator field

$$H(\mathbf{x}, t) \equiv \frac{\rho(\mathbf{x}, t) - \rho_2}{\rho_1 - \rho_2}, \tag{5}$$

the volume fraction (of fluid 1) in cell i is then defined as

$$\alpha_i(t) \equiv \langle H \rangle_i. \tag{6}$$

We will denote the mesh faces, $\mathscr{F}_1, \mathscr{F}_2, \ldots$, and the list of labels of faces on the boundary of cell i will be denoted B_i. On the time axis, the times, $t_1 < t_2 < \cdots$ define the time intervals (or steps), $[t_n, t_{n+1}]$, over which the governing equations are integrated. We will use superscripts to denote a function evaluated at one of these times, $f^n = f(t_n)$.

With these definitions in place, we can now rewrite Eq. 1 in terms of H and integrate it over the volume of cell i and over the time interval $\left[t_n, t_{n+1}\right]$. This converts the equation into an evolution equation for the volume fraction in cell i. Unfortunately, space does not allow a full derivation here (the reader is referred to [1] for more details), but the form of the equation is

$$\alpha_i^{n+1} = \alpha_i^n - \frac{1}{V_i} \sum_{j \in B_i} \Delta V_{ij}^n, \tag{7}$$

where the quantity ΔV_{ij}^n is the total volume of fluid 1 flowing from cell i during the time interval $[t_n, t_{n+1}]$ into the neighbour cell with which it shares face j. This important quantity is defined by

$$\Delta V_{ij}^n \equiv \int_{t_n}^{t_{n+1}} \int_{\mathscr{F}_j} H(\mathbf{x}, \tau) \mathbf{u}(\mathbf{x}, \tau) \cdot d\mathbf{S}_{ij} d\tau. \tag{8}$$

Here, $d\mathbf{S}_{ij}$ is the infinitesimal surface element of face j, oriented out of cell i, so if cell k is the other cell of face j, then $d\mathbf{S}_{kj} = -d\mathbf{S}_{ij}$ and $\Delta V_{kj}^n = -\Delta V_{ij}^n$.

The art of constructing a volume-of-fluid algorithm is all about coming up with the best possible approximation of ΔV_{ij}^n given the incomplete available data. In our collocated finite volume framework, the available data consists of the volume fractions α_i^n, the cell- averaged velocities, $\langle \mathbf{u} \rangle_i^n$, and the volumetric face fluxes,

$$\phi_{ij}^n \equiv \int_{\mathscr{F}_j} \mathbf{u}(\mathbf{x}, t_n) \cdot d\mathbf{S}_{ij}. \tag{9}$$

In the following, we show how isoAdvector uses these data and a number of geometric considerations to come up with an approximation for ΔV_{ij}^n.

2.1 Interface Reconstruction

We start by noting that most cells will normally be fully immersed in either fluid 1 or fluid 2 during the time interval, and for such a cell, the advection problem is trivial, since there is only one fluid fluxed through all its faces. The surface cells requiring special treatment are those containing both fluid 1 and fluid 2. We will define a surface cell as one with $\varepsilon < \alpha_i^n < 1 - \varepsilon$, where we typically set $\varepsilon = 10^{-8}$ in our calculations. The first step in finding α_i^{n+1} for such cells is to reconstruct the fluid interface inside the cell from the available data, α_i^n at time t_n. In the isoAdvector method, this is done by calculating an isosurface inside the cell. For this purpose, we need to first interpolate the volume fractions from the cell centres to the vertices. This process is illustrated in Fig. 1. This interpolation can be done in various ways. For convenience, we have chosen the inverse distance weighting provided by the `volPointInterpolation` class.

With volume fractions interpolated to all vertices of cell i, we can choose an isovalue, α_0, and construct the α_0-isosurface inside the cell. This we do by going through all the cell's edges and determining whether they are cut by the isosurface. An edge is cut, if the interpolated volume fraction at one end is larger than α_0 and the value at the other end is smaller than α_0. If that is the case, we calculate the intersection point along the edge by linear interpolation. Connecting these intersection points across the cell faces, we construct the cell–isosurface intersection, as illustrated in Fig. 2. The representation of this intersection will be called an *isoface*, because it is really just an internal face cutting the cell into two subcells. We can calculate the face centre, $\mathbf{x}_S$, and face unit normal vector, $\hat{\mathbf{n}}_S$, for this isoface as for any other mesh face (black dot and vector in Fig. 2).

If we imagine sweeping the isovalue, α_0, from the lowest to the highest cell vertex α value, the isoface will pass through the cell. Which isovalue in this interval should we choose for a particular surface cell? Our answer is the isovalue that makes the isoface

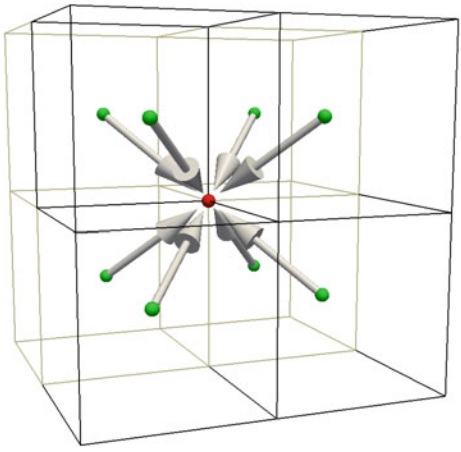


Fig. 1 Interpolation of volume fraction to a vertex from all surrounding cells

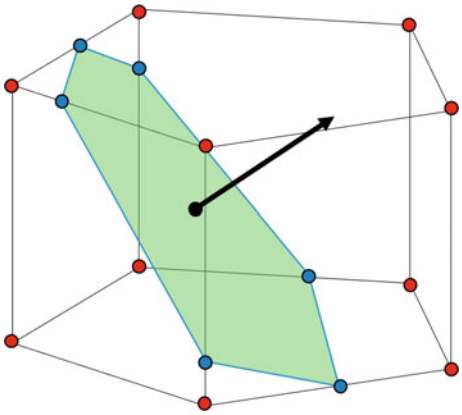


Fig. 2 Construction of the isoface inside a surface cell

cut the cell into subcells of subvolumes in accordance with the cell's volume fraction, α_i^n. To find this isovalue (different in each surface cell), we have implemented an efficient root-finding algorithm that exploits the fact that the volume fraction is a piecewise cubic polynomial in α_0. The details of this algorithm are further described in [1]. This concludes our description of the interface reconstruction at time t_n.

2.2 *Interface Advection*

The next step is to exploit our new knowledge about the interface position inside surface cells at time t_n to estimate how much of the total fluid volume transported across a face during a time step, $[t_n, t_{n+1}]$, is fluid 1 and how much is fluid 2. We will first make the assumption that $\mathbf{u}(\mathbf{x}, \tau)$ in Eq. 8 can be replaced by an appropriately chosen constant vector $\tilde{\mathbf{u}}_j^n$, which is representative of the velocity on the face during the whole time interval. We also assume that we can write

$$d\mathbf{S}_{ij} = \hat{\mathbf{n}}_{ij}(\mathbf{x})dA \approx \frac{\mathbf{S}_{ij}}{|\mathbf{S}_{ij}|}dA, \tag{10}$$

where $\hat{\mathbf{n}}_{ij}$ is the (for a non-planar face spatially varying) unit normal vector and $\mathbf{S}_{ij}$ is the mean normal vector of face j pointing out of cell i. Then, the volumetric face flux can be defined as $\tilde{\phi}_{ij}^n \equiv \tilde{\mathbf{u}}_j^n \cdot \mathbf{S}_{ij}$, and ΔV_{ij}^n in Eq. 8 can be approximated by

$$\Delta V_{ij}^n \approx \frac{\tilde{\phi}_{ij}^n}{|\mathbf{S}_{ij}|} \int_{t_n}^{t_{n+1}} \int_{\mathscr{F}_j} H(\mathbf{x}, \tau) dA d\tau. \tag{11}$$

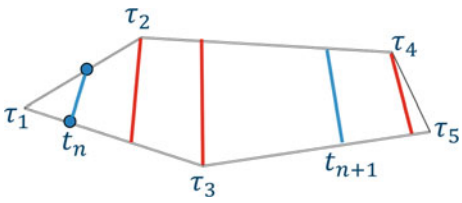


Fig. 3 Face–interface intersection line sweeping the face

In the current implementation, we simply use the volumetric face fluxes, ϕ_{ij}^n, at the beginning of the time step for $\tilde{\phi}_{ij}^n$.[1] The remaining area integral in Eq. 11 is just the area of face j that is submerged in the reference fluid. This area we will denote by

$$A_j(\tau) \equiv \int_{\mathscr{F}_j} H(\mathbf{x}, \tau) dA. \tag{12}$$

If we want to be able to take time steps in which the interface moves a substantial fraction of a cell size, we should come up with an estimate of how A_j varies with time *within* a time step. The topmost face of the polygonal prism cell in Fig. 2 is reproduced in Fig. 3 with the initial *face–interface intersection line* at t_n shown in blue.

To estimate how this line sweeps over the face as the isoface moves in the velocity field, we first interpolate the velocity field to the initial isoface centre, $\mathbf{x}_s$, shown with a black dot in Fig. 2. We can then take the dot product of the interpolated velocity with the isoface unit normal, $\hat{\mathbf{n}}_S$, to obtain the speed of the isoface motion perpendicular to itself, U_S. For a vertex, $\mathbf{x}_v$, in Fig. 3, we can also estimate the perpendicular distance to the isoface by $d_v = (\mathbf{x}_v - \mathbf{x}_S) \cdot \hat{\mathbf{n}}_S$. With the calculated isoface normal speed and vertex-to-isoface distance, we can then estimate the time of arrival at vertex $\mathbf{x}_v$ to be $\tau_v = d_v / U_S$. In this way, we obtain the "vertex arrival times", $\tau_1, \tau_2, \ldots$, shown in Fig. 3. As illustrated, some of these will generally be outside the integration interval, $[t_n, t_{n+1}]$ and some will be inside. The crucial point is now, that between two such times, say, τ_2 and τ_3 in Fig. 3, the face–interface intersection line sweeps a quadrilateral. If we assume the line sweeps this quadrilateral steadily, we can come up with an analytical expression for the way in which A_j depends on τ on this sub-interval. This expression is a quadratic polynomial in τ and its coefficients depend only on the shape of the quadrilateral. The resulting time variation of $A_j(\tau)$ as the line sweeps the face is illustrated in Fig. 4.

With a piecewise quadratic polynomial for $A_j(\tau)$ in Eq. 12, its time integral in Eq. 11 is a piecewise cubic polynomial, and ΔV_{ij}^n is finally obtained as the sum of the contributions from these sub-intervals. We note that a face of a surface cell may initially be fully immersed in fluid 1 or 2, and then become intersected during the time

[1] An idea could be to use ϕ_{ij}^{n-1} and ϕ_{ij}^n to obtain an estimate, $\overline{\phi}_{ij}^{n+1}$, of ϕ_{ij}^{n+1}, and then use this to estimate $\tilde{\phi}_{ij}^n \approx 0.5(\phi_{ij}^n + \overline{\phi}_{ij}^{n+1})$ in Eq. 11. Also, if using more than one outer corrector, the value from the previous iteration could be used for $\overline{\phi}_{ij}^{n+1}$ in a similar manner (for all but the first iteration).

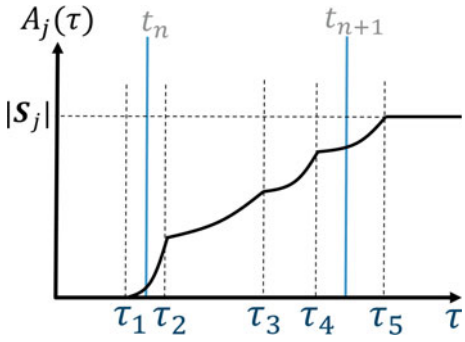


Fig. 4 Submerged face are, $A_j(\tau)$, as a piecewise quadratic polynomial

interval $[t_n, t_{n+1}]$. With the calculated vertex arrival times, this situation corresponds to $\tau_1 > t_n$ (the t_n line in Fig. 4 would then be further to the left), and it is treated by fluxing pure fluid 1 or 2 through the face in the sub-interval $[t_n, \tau_1]$. Similarly, if the interface leaves the face during $[t_n, t_{n+1}]$, we will have $\tau_5 < t_{n+1}$ for a pentagonal face (the t_{n+1} line would then be further to the right in Fig. 4), and we must flux pure fluid 1 or 2 through the face during the last sub time interval $[\tau_5, t_{n+1}]$.

There is one final decision we must make before our advection routine is complete: for a face j, both its owner and neighbour cell may be surface cells with their isofaces not coinciding exactly on face j due to the different isovalues used in the two cells and not moving with exactly the same velocity due to the spatial variations in the velocity field. Which cell should be used to calculate ΔV_{ij}^n for this face? In our current implementation, we have chosen to let ΔV_{ij}^n be determined by the upwind cell, i.e., the owner if $\phi_j^n > 0$ and the neighbour if $\phi_j^n < 0$.

2.3 Bounding

The test cases provided with the released isoAdvector code [2] show that the method outlined above generally gives very good estimates of ΔV_{ij}^n and leads to accurate interface advection, as long as the interface is well resolved by the mesh and time step size is limited to CFL < 1. In some situations, there may, however, arise small inaccuracies, which can build up over time and lead to intolerable levels of unboundedness. To prevent the gradual build up of unboundedness, we have introduced a bounding step that detects unboundedness and tries to adjust the ΔV_{ij}^n's of unbounded cells with a procedure that is described in detail in [1]. If this pure redistribution step fails, the provided code also gives the option of brute-force non-volume preserving chopping α_i^{n+1} after each advection step to guarantee boundedness. Activating this will ruin the machine precision volume conservation, but our experience so far indicates that, in many situations, the resulting volume conservation error is very small.

In the current isoAdvector implementation, we assume that there is only a single isoface inside a cell. There are several occasions when one would expect more iso-

faces inside a cell. One such situation is when a planar interface passes a non-planar mesh face with which it is close to parallel. During the passage, the face and interface will intersect at more than two points and the face–interface intersection cannot be represented by a single straight line. Proper treatment of such an event can be implemented by on-the-fly decomposition of the non-planar face into triangular subfaces sharing the face centre as their common apex. A face–interface intersection line can then be calculated for each triangle separately (a triangular face can, at most, have two intersection points with the interface).

Another situation with more than one isoface inside a cell is when two separate volumes of fluid approach each other and collide inside a cell. Then, there will be a time interval just before the collision during which each volume has its own separate piece of interface within the cell. In this event, the solution could be to decompose the whole cell into tetrahedra sharing the cell centre as their common apex and separately reconstruct the interface in each subcell. This solution has not yet been implemented.

We have experienced that, due to these shortcomings of the current implementation, the bounding errors can be substantial, e.g., on polyhedral meshes with many highly non-planar faces of the type obtained by generating the dual mesh of a random tetrahedral mesh. The method still works on such meshes if switching on the brute-force chopping described above, but one may then experience substantial loss of volume conservation. We plan to implement the fixes described above in a future release of the code.

3 Pure Advection Tests

In this section, we compare the performance of isoAdvector and MULES with two standard pure advection test cases with a predefined velocity field.

3.1 Notched Disc in Solid Body Rotation

Our first test case is the notched disc in solid body rotation, which has become a standard test case since its introduction in [4]. The domain is the unit square, the velocity field is the solid body rotation around the point (0.5, 0.5):

$$u = -2\pi(y - 0.5), \quad v = 2\pi(x - 0.5). \tag{13}$$

The initial volume fraction field is 1 within the disc of radius 0.15 centred at (0.5, 0.75), except in a slit of width 0.06 going up to y=0.85. The disc rotates around (0.5, 0.5) and returns to its original position at time t = 1. The resulting interface shape after such a rotation with isoAdvector and MULES is shown in Fig. 5 for three different mesh types with square, triangular and polygonal cells. All simulations have been performed with CFL = 0.1 and 0.5, but since the isoAdvector simulations with

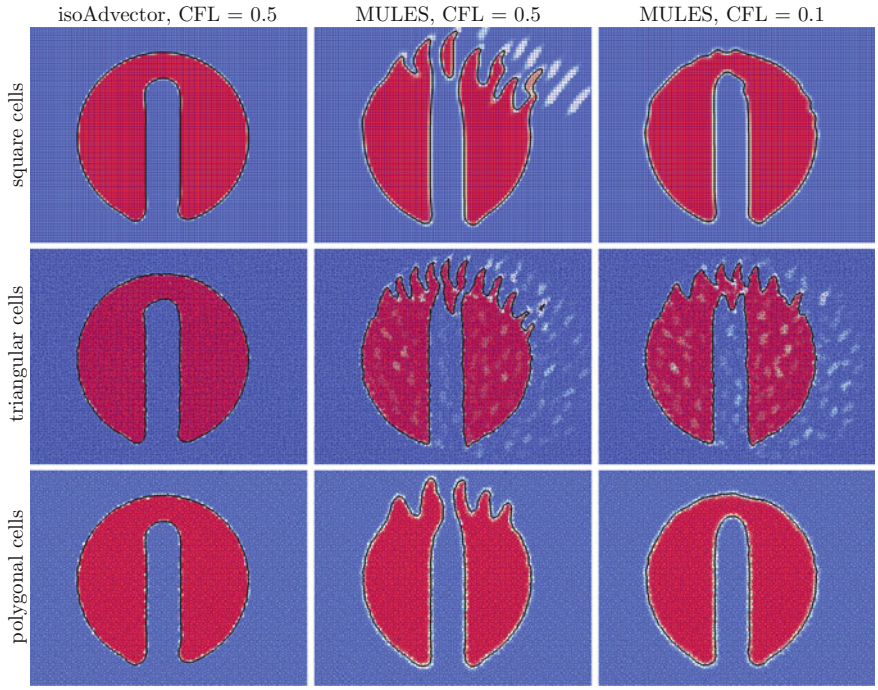


Fig. 5 Notched disc advection test. Volume fraction field shown after one full rotation. 0.5-contour shown with a black curve

Table 1 Error for the notched disc: $E_1 \equiv \sum_i |\alpha_i - \alpha_i^{\text{exact}}| V_i / \sum_i \alpha_i^{\text{exact}} V_i$ (sums are over all cells)

Mesh type	isoAdvector CFL = 0.5	MULES CFL = 0.5	MULES CFL = 0.1
Square	0.014	0.21	0.062
Triangular	0.022	0.17	0.13
Polygonal	0.022	0.14	0.064

CFL = 0.1 and 0.5 are almost indistinguishable, we only show the latter. Tables 1 and 2 show the error compared to the exact solution and the calculation time for the 9 simulations. From the figure and Tables 1 and 2, we remark that:

- IsoAdvector with CFL = 0.5 performs better than MULES with both CFL = 0.5 and 0.1 on square, triangular and polygonal meshes.
- MULES severly distorts the shape on all mesh types with CFL = 0.5.
- On square and polygonal meshes, MULES improves dramatically when going from CFL = 0.5 to 0.1, but not on the triangular mesh.
- isoAdvector is ∼3 times faster than MULES with CFL = 0.5.

Table 2 Calculation times in seconds for notched disc simulations on a single processor

Mesh type	isoAdvector CFL = 0.5	MULES CFL = 0.5	MULES CFL = 0.1
Square	25	175	437
Triangular	232	639	1929
Polygonal	85	278	803

Table 3 E_1 error (left) and calculation times (right) for a sphere in a reversed 3D shear flow on a polyhedral mesh

CFL	isoAdvector	MULES
0.5	0.1	0.24
0.1	0.11	0.15
0.5	146 s	439 s
0.1	513 s	1622 s

3.2 Sphere in Shear Flow

Our second pure advection case is from [5]. The domain is the unit box and the initial volume fraction field is 1 within the sphere of radius 0.15 centred at (0.5, 0.75, 0.25) and 0 elsewhere. The velocity field in which the interface is advected is

$$\mathbf{u}(\mathbf{x}, t) = \cos\left(\frac{2\pi t}{T}\right)\left[\sin(2\pi y)\sin^2(\pi x), -\sin(2\pi x)\sin^2(\pi y), (1-2r)^2\right], \tag{14}$$

where $T = 6$ and $r = \sqrt{(x-0.5)^2 + (y-0.5)^2}$. In this flow, the initial spherical interface is sheared into a thin spiralling sheet until, at $t = 1.5$, it has reached its maximum deformation and flows back to its initial shape and position at time $t = 3$. We run the case on a polyhedral mesh of the type generated with the pMesh tool of cfMesh [6]. In Fig. 6, we show the results obtained with isoAdevector and MULES using CFL = 0.5 and 0.1.

From Fig. 6 and Tables 3, we see that with this test case, isoAdvector is more accurate and approximately three times faster than MULES, but also note that MULES does a decent job, even with CFL = 0.5. We also observe that the E_1 error with isoAdvector actually increases slightly with smaller time steps.

4 Using isoAdvector in interFoam

In interFoam, the MULES explicit solver code does not only provide the updated volume fractions, α_i^{n+1}, but also provides the quantity `rhoPhi`, which is used in the convective term, `fvm::div(rhoPhi, U)`, in the momentum matrix equation,

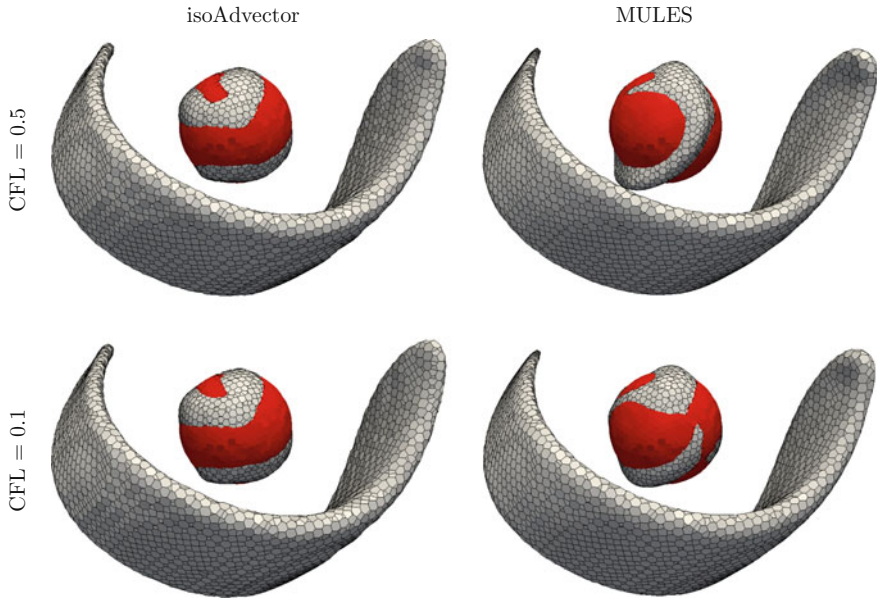


Fig. 6 Sphere deformed in a 3D shear flow with isoAdvector and MULES for CFL = 0.5 and 0.1. The initial sphere is shown in red. The interface shape at maximum deformation ($t = 1.5$) and at the time of return to the spherical shape ($t = 3$) is shown in grey

`UEqn`. To understand the way in which we should construct `rhoPhi` from ΔV_{ij}^n, let us start by looking at the convective term in the Euler equations in Eq. 3, formally integrated over a small time interval and over a cell:

$$\langle \rho \mathbf{u} \rangle_i^{n+1} = \langle \rho \mathbf{u} \rangle_i^n - \frac{1}{V_i} \int_{t_n}^{t_{n+1}} \int_{\mathscr{C}_i} \nabla \cdot [\rho(\mathbf{x}, \tau)\mathbf{u}(\mathbf{x}, \tau)\mathbf{u}(\mathbf{x}, \tau)] \, dV d\tau + \cdots \quad (15)$$

We will denote this integrated convective term by C_i^n and use Gauss's theorem to write it as

$$C_i^n = \frac{1}{V_i} \sum_{j \in B_i} \int_{t_n}^{t_{n+1}} \int_{\mathscr{F}_j} \rho(\mathbf{x}, \tau)\mathbf{u}(\mathbf{x}, \tau)\mathbf{u}(\mathbf{x}, \tau) \cdot d\mathbf{S} d\tau. \quad (16)$$

We now approximate $\mathbf{u}(\mathbf{x}, \tau)$ with a constant representative velocity vector, $\tilde{\mathbf{u}}_j^n$ and use $\mathbf{u} \cdot d\mathbf{S} \approx \tilde{\phi}_{ij}^n / |\mathbf{S}_j| dA$ as described at the beginning of Sect. 2. This allows us to write

$$C_i^n \approx \frac{1}{V_i} \sum_{j \in B_i} \tilde{\mathbf{u}}_j^n \frac{\tilde{\phi}_{ij}^n}{|\mathbf{S}_{ij}|} \int_{t_n}^{t_{n+1}} \int_{\mathscr{F}_j} \rho(\mathbf{x}, \tau) dA d\tau. \quad (17)$$

Now, using the definition of $H(\mathbf{x}, t)$ in terms of $\rho(\mathbf{x}, t)$ in Eq. 5 and the definition of the submerged face area, A_j, in terms of H in Eq. 12, we can write Eq. 17 as

$$C_i^n \approx \frac{1}{V_i} \sum_{j \in B_i} \tilde{\mathbf{u}}_j^n \frac{\tilde{\phi}_{ij}^n}{|\mathbf{S}_{ij}|} \int_{t_n}^{t_{n+1}} \left[\rho_2 + (\rho_1 - \rho_2) A_j(\tau) \right] d\tau. \tag{18}$$

With the definition of ΔV_{ij}^n in Eq. 8, we can finally write the convective term as

$$C_i^n \approx \frac{1}{V_i} \sum_{j \in B_i} \tilde{\mathbf{u}}_j^n \left[\rho_2 \tilde{\phi}_{ij}^n \Delta t_n + (\rho_1 - \rho_2) \Delta V_{ij}^n \right]. \tag{19}$$

Here, the content of the square brackets is exactly the desired expression for `rhoPhi`. The specific expression for $\tilde{\mathbf{u}}_j^n$ in terms of the cell-averaged velocities is determined by the settings in `fvSchemes` for the convective term.

5 The damBreak Case

For an initial investigation of the behaviour of our new interFoam solver using isoAdvector instead of MULES, we run a refined version of the standard dam break tutorial shipped with OpenFOAM®-4.0. The domain is a box of width and height 0.584 m, with a small rectangular obstacle on the bottom and the water initially placed in a rectangular column on the left side of the domain. The case is run with an adaptive time step using `maxAlphaCo` = `maxCo` = 0.5 . In Fig. 7, we show snapshots of the volume fraction field at two times. In the top panels, just after impact with the obstacle on the floor, we clearly see how isoAdvector—in contrast to MULES—is capable of keeping the interface sharp, even for droplets of only a few cells' width. In the bottom panels, the water is starting to settle, and we see how the interface produced with isoAdvector is only one cell wide, whereas the interface produced with MULES covers many cells. Calculation times are similar for the two runs.

This is to be thought of as a kind of "Hello World!" case for our new solver, and caution should be taken in drawing quantitative conclusions from this setup. In future work, we will conduct more quantitative investigations based, e.g. on the experimental data provided in [7].

We remark that a razor-sharp interface is not always the best representation of the physical water distribution on the given mesh. If the encounter with the obstacle in a real physical damBreak experiment causes the interface to explode into a cloud of subcell-sized droplets, then a smeared representation, together with an appropriate dispersed flow model, may be a better representation of the physical reality. To prevent nonphysical sharpening of the interface in regions with clouds of subcell droplets and bubbles, one could introduce a quantitative criterion for detecting such regions and replace the isoAdvector interface treatment for surface cells in these regions with a dispersed treatment.

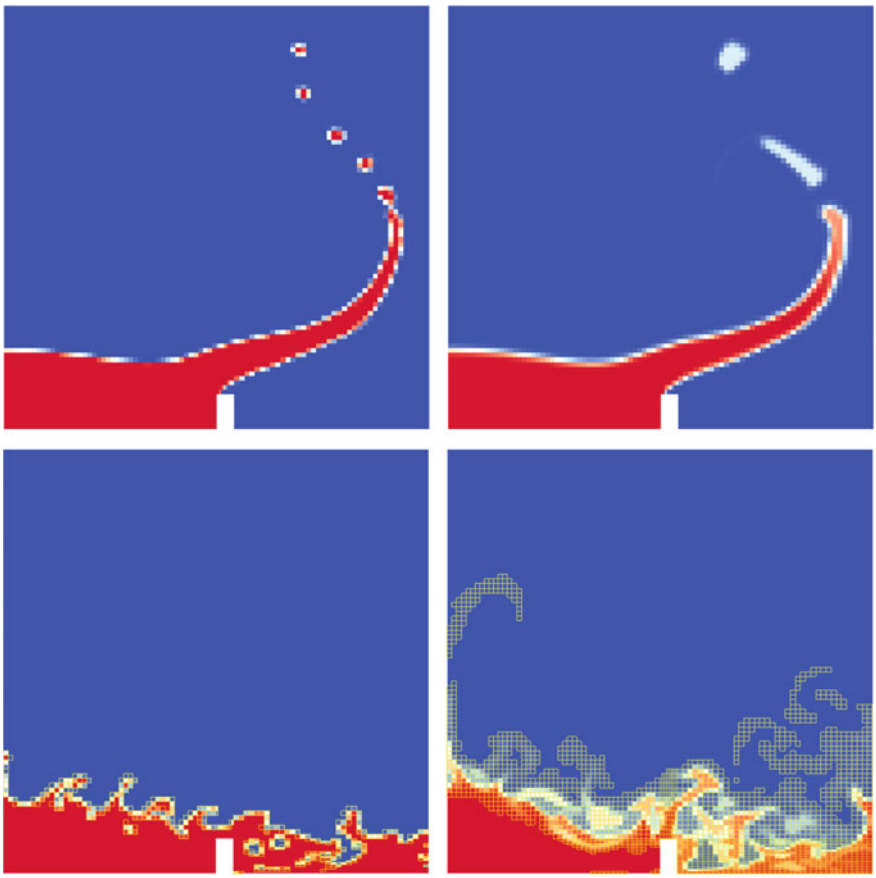


Fig. 7 Dam break at times $t = 0.32$ s (top) and 1.1 s (bottom), run with isoAdvector (left) and MULES (right). Cells with $0.001 < \alpha_i^n < 0.999$ at $t = 1.1$ s are coloured yellow in the lower panels

6 Steady Stream Function Wave

The purpose of our final test case is to investigate the effect of replacing MULES with isoAdvector on the propagation of a steady stream function wave. The initial surface elevation, velocity and pressure fields are calculated using a Fourier approximation method, which is well described in [8]. The derivation is based on potential flow theory with vacuum above the wave. This corresponds to $\rho_2 = 0$, which is not practically possible with the interFoam solver, because it involves division by ρ in the PISO loop. We, therefore, use $\rho_2 = 0.1$ kg/m^3 and $\rho_1 = 1000$ kg/m^3. As in [9], we use a wave with height $H = 10$ m and period $T = 14$ s on depth $D = 20$ m. This gives rise to a wavelength of $L = 193.23$ m, which we choose as the length of our rectangular domain with, `cyclic` boundary conditions on the sides. The cells in the

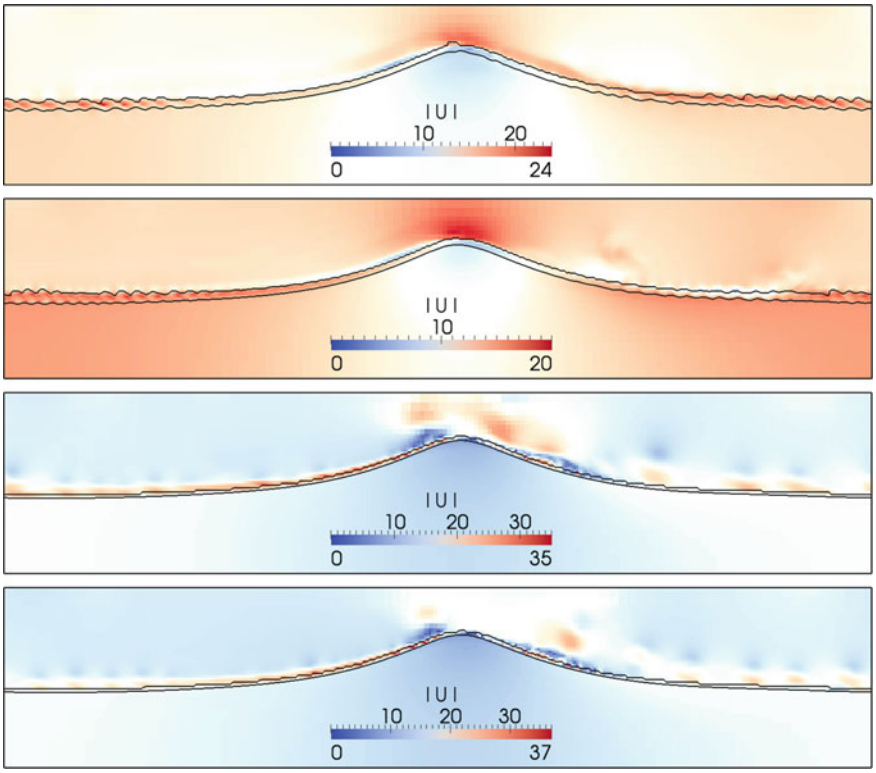


Fig. 8 Stream function wave with $H = 10$ m, $D = 20$ m and $T = 14$ s at time $t = 10$ s. From top: MULES-Euler, MULES-Crank–Nicolson, isoAdvector-Euler, isoAdvector-Crank–Nicolson

interface region are squares of side length 0.5 m corresponding to 20 cells per wave height and ∼386 cells per wavelength. Above and below the interface region, the mesh is coarser, with cell size up to 2 m. For `div(rho*phi,U)`, we use `Gauss limitedLinearV 1`, as opposed to the `upwind` scheme used in [9]. Another difference is that in our setup, we initialise the wave in a co-moving frame, and so the surface elevation and velocity field should ideally not change throughout the transient simulation. Also, we use a fixed time step of 0.002 s, which based on the theoretical crest particle velocity of 5.95 m/s, gives a CFL number of 0.0238. We run the setup with MULES and isoAdvector, using both `Euler` and `crankNicolson 0.5` for time integration. The resulting wave shape and velocity magnitudes for the four combinations a short time after the simulations have been started ($t = 10$ s) are shown in Fig. 8. The interface thickness is shown by plotting the 0.5 and 0.0001 contours of the volume fraction data. We see that the interface is sharper and smoother in the isoAdvector simulations than in the MULES simulations. But we also observe that the air speed in a narrow band just above the surface takes values almost twice as high in the isoAdvector simulations (note the different colour scales in the different

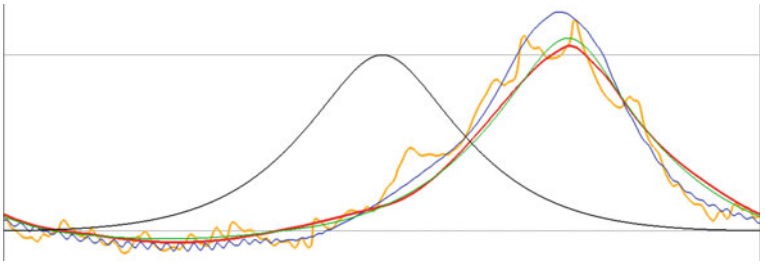


Fig. 9 Stream function wave with $H = 10$ m, $D = 20$ m and $T = 14$ s at time 105 s. The x-axis has been compressed by a factor of 4. Black: exact surface elevation (0.5 contour). Orange: MULES-Euler. Blue: MULES-Crank–Nicolson. Red: isoAdvector-Euler. Green: isoAdvector-Crank–Nicolson

panels). These larger air velocities do not seem to affect the interface significantly, as illustrated in Fig. 9, where we show the surface elevations from the four runs after 105 s corresponding to 7.5 wave periods.

All simulations have a celerity that is slightly too high, causing all waves to have drifted almost a quarter of a wavelength to the right compared to the theoretical steady profile crest centred in the middle of the domain. The MULES-Euler wave (orange) has broken, giving rise to a jerky profile. The MULES-Crank–Nicolson wave crest (blue) has grown significantly and the trough is wrinkled. The isoAdvector waves with Euler (red) and Crank–Nicolson (green) also have slight overshoots in wave height, but these are much smaller and with much smoother profiles. This is a preliminary test and since many things can change if the case setup is adjusted, one should not jump to conclusions before a more thorough study has been conducted. It is, however, safe to conclude that using isoAdvector instead of MULES in surface wave propagation simulations does have a clear effect on the solution. It is also safe to conclude, that the spurious tangential velocities observed at the interface for large density ratios are not caused by MULES alone. More likely, the problem is also associated with the PISO loop implementation in interFoam not taking the large density jump properly into account, e.g., when interpolating density-dependent quantities between cell centres and face centres. We expect that an improved density jump treatment in this part of the code can be achieved by using the information provided by isoAdvector about the interface position inside surface cells.

7 Summary

We have given a brief description of the isoAdvector algorithm for advection of a sharp interface across general meshes. We added two pure advection cases to the suite of test cases already presented in [1], demonstrating the superior behaviour of isoAdvector compared to MULES with respect to accuracy and efficiency. We derive an expression for the convective term in the momentum equation so that

isoAdvector can be used in interFoam instead of MULES. The resulting solver is tested using the damBreak case and a steady stream function wave in a periodic domain. From these tests, we conclude that using isoAdvector in interFoam is feasible and leads to a sharper interface. Using isoAdvector for the tested wave propagation case leads to significantly higher spurious tangential velocities in the lighter phase, but nevertheless, the quality of the solution is improved. In future work, we will exploit the isoface data to impose consistent physical boundary conditions at the interface in the PISO loop.

References

1. J. Roenby, H. Bredmose, and H. Jasak, "A computational method for sharp interface advection," *Royal Society Open Science*, vol. 3, p. 160405, 2016.
2. J. Roenby, "isoAdvector." www.github.com/isoadvector.
3. S. S. Deshpande, L. Anumolu, and M. F. Trujillo, "Evaluating the performance of the two-phase flow solver interFoam," *Computational Science & Discovery*, vol. 5, no. 1, p. 014016, 2012.
4. S. T. Zalesak, "Fully multidimensional flux-corrected transport algorithms for fluids," *Journal of Computational Physics*, vol. 31, no. 3, pp. 335–362, 1979.
5. J. López, C. Zanzi, P. Gómez, F. Faura, and J. Hernández, "A new volume of fluid method in three dimensions–part II: Piecewise-planar interface reconstruction with cubic-bézier fit," *International Journal for Numerical Methods in Fluids*, vol. 58, no. 8, pp. 923–944, 2008.
6. "cfMesh." http://cfmesh.com/. Accessed: 2016-12-09.
7. L. Lobovsk, E. Botia-Vera, F. Castellana, J. Mas-Soler, and A. Souto-Iglesias, "Experimental investigation of dynamic pressure loads during dam break," *Journal of Fluids and Structures*, vol. 48, pp. 407–434, 2014.
8. J. D. Fenton, "Numerical methods for nonlinear waves," in *Advances in Coastal and Ocean Engineering*, vol. 5, pp. 241–324, World Scientific, July 1999.
9. B. T. Paulsen, H. Bredmose, H. Bingham, and N. Jacobsen, "Forcing of a bottom-mounted circular cylinder by steep regular water waves at finite depth," *Journal of Fluid Mechanics*, vol. 755, pp. 1–34, 2014.

Liquid Atomization Modeling in OpenFOAM®

J. Anez, S. Puggelli, N. Hecht, A. Andreini, J. Reveillon and F. X. Demoulin

Abstract Several approaches have been developed to simulate liquid-jet atomization phenomena. Despite recent developments in numerical methods and computer performance, direct numerical simulation of the atomization process remains inaccessible for practical applications. Therefore, to carry out numerical simulations of the injected liquid from the internal flow within flow as far as the final dispersed spray, a modeling strategy has been developed. It is composed of a set of models implemented within the open-source software `OpenFOAM®`. First, the so-called Euler–Lagrange Spray Atomization (ELSA) approach is introduced. This is Eulerian formulation dedicated to jet atomization that is based on the analogy of turbulent mixing in a flow with variable density in the limit of infinite Reynolds and Weber numbers. Second, ELSA's extension to a Quasi-Multiphase Eulerian (QME) approach is proposed. This method solves the problem of a second-order closure in modeling the turbulent liquid flux, hence solving the slip velocity between the phases. Third, an enhanced version of ELSA coupling with an Interface Capturing Method (ICM) and a Lagrangian approach for the final spray are introduced.

J. Anez · J. Reveillon (✉) · F. X. Demoulin
CORIA-UMR CNRS 6614, Normandy University, Rouen, France
e-mail: julien.reveillon@coria.fr

J. Anez
e-mail: javier.anez@coria.fr

F. X. Demoulin
e-mail: francois-xavier.demoulin@coria.fr

S. Puggelli · A. Andreini
Department of Industrial Engineering (DIEF), University of Florence, Florence, Italy
e-mail: stefano.puggelli@htc.de.unifi.it

A. Andreini
e-mail: antonio.andreini@htc.de.unifi.it

N. Hecht
LOMC UMR CNRS 6294, Normandy University, Le Havre, France
e-mail: nicolas.hecht@univ-lehavre.fr

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_22

1 Introduction

Several strategies can be found in the literature for modeling fuel injection and coping with the multiphase/multi-scale nature of the flow. A full resolution of the interface is possible thanks to Direct Numerical Simulation (DNS), with interface capturing and/or reconstruction methods [1–4]. However, it is not applicable as far as industrial applications are concerned, mostly because of an unfeasible computational cost. Moreover, the notion of DNS when an interface is considered, must be approached with special care.

Therefore, atomization modeling is required. Many approaches are based on kinetic theory, in which the spray is described through a number density function that verifies the Williams–Boltzmann Equation (WBE) [5] containing terms for spatial transport, evaporation and fluid drag. A widely used approach to solving the WBE is the Lagrangian Monte Carlo method [6], in which the liquid is tracked with a Lagrangian description and the gas is solved in a Eulerian framework. Its advantage lies in a straightforward implementation of physical processes such as evaporation and secondary breakup, even if its computational cost is generally challenging, especially in unsteady configurations. Indeed, a high number of parcels is required in each cell of the numerical domain in order to reach statistical convergence.

Another approach to solve the WBE is based on a Euler–Euler (EE) formulation, in which both phases are treated as a continuum. This solution is very attractive for describing the evolution of the spray's characteristics. Moreover, reduced computational costs in terms of parallel computing are among the advantages of this formulation. Nonetheless, despite the efficiency of EE methods on actual supercomputers, the direct resolution of WBE is generally unfeasible since the dimension of the problem is multiplied by the number of spray characteristics (position, velocity, size, temperature, etc) retained to describe the physics. This constrains EE methods to addressing a limited description of these properties. Numerous possible hypotheses have led to an abundant research in this framework. For instance, in Multi-Fluids models [7, 8], the droplet geometry information is discretized in sections to represent the spray distribution. Another solution [9] is a smooth reconstruction based on a sum of the kernel of the density functions, with a quadrature method of moments being employed for this purpose. Other approaches, such as entropy maximization [10] or moments with interpolative closure [11], can also be found in literature, but their description goes beyond the scope of the present work.

All these methods based on WBE generally assume that the spray is composed of numerous individual spherical droplets with such well-defined features as position or diameter. However, this is far from the case with the atomizers generally employed in an industrial framework. Indeed, the liquid phase is initially a continuum (i.e., liquid jet or film), and it is not possible to define such well-defined characteristics until the end of primary breakup. Therefore, multi-scale models have been developed in this work. In these models, the interface is considered, depending of the resolution of

the interface, as a mixing zone, so that both the liquid and gas phases coexist at the same macroscopic position with an occupied portion of the volume defined by the liquid volume fraction (α_l). In this context, two families of equilibrium models have been developed. The first possibility [12] is to use the liquid volume fraction as the unique variable allowing the description of the interface. Another set of approaches is based on a transport equation for the liquid/gas interface density [1, 13]. In this second group of models, the Eulerian–Lagrangian Spray Atomization (ELSA) models are of major importance [1, 13]. The purpose of the present work is, therefore, to formulate models based on the ELSA concept and to describe their implementation in the open-source software `OpenFOAM®`.

The following section is devoted to the description of a Eulerian–Eulerian solver, directly derived from the ELSA formulation. Then, the Quasi-Multiphase Eulerian (QME) approach, which introduces the effects of a relative velocity between phases, is presented, with an application that considers a jet in a cross-flow test case. Finally, an innovative coupling between an Interface Capturing Method (ICM) and a complete ELSA approach is applied.

2 ELSA-Base

In the ELSA-base model, a two-phase flow is considered as a single phase composed of two species with highly variable density. Several features are summarized such as large-scale properties (penetration length and angle of dispersion of the liquid core) and small-scale characteristics (mean droplet diameter and their size distribution). These features are modeled based on the analogy between atomization and turbulent mixing of a liquid jet with a high-density ratio within the surroundings [13]. In this section, starting from this complete approach, governing equations of the Eulerian–Eulerian solver are presented, together with its implementation in `OpenFOAM®`.

A balance equation for the liquid volume fraction α_l, which represents the proportion of liquid in a considered volume, is considered with an additional term, namely the *turbulent diffusion liquid flux*, similar to the turbulent diffusion flux defined for a turbulent single-phase flow. Additionally, the small-scale characteristics such as droplet size distribution and mean droplet diameter, can be represented by means of the liquid gas interface density, namely Σ (greek letter on the text), which represents the liquid/gas surface interface per unit of volume. The concept of interface density, is more general than droplet diameter or Sauter Mean Diameter (SMD). Indeed, liquid shapes are not always spherical, thus Σ (greek letter on text) is a more generalized quantity able to quantify any type of interface. The mean mixture velocity $\boldsymbol{U}$ follows a classical transport equation with the deviatoric stress tensor $\boldsymbol{T}$ related to the viscous stress, plus any additional term coming from turbulent fluctuations. ρ is the density and $\boldsymbol{f}_b$ is the body forces per unit of mass. The treatment of $\boldsymbol{T}$ may require a dedicated modeling to account for density fluctuations, and previous studies have shown that the usual single-phase flow models are not satisfactory [14]. The resulting system of equations, assuming the two phases have constant density, gives us

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{U}) = 0 \\ \dfrac{\partial \rho \boldsymbol{U}}{\partial t} + \nabla \cdot (\rho \boldsymbol{U} \otimes \boldsymbol{U}) = -\nabla p + \nabla \cdot \boldsymbol{T} + \rho \boldsymbol{f}_b + \nabla \cdot \boldsymbol{R}_U \\ \dfrac{\partial \alpha_l}{\partial t} + \nabla \cdot (\boldsymbol{U} \alpha_l) = \nabla \cdot (\boldsymbol{R}_{\alpha_l}) \\ \dfrac{\partial \Sigma}{\partial t} + \nabla \cdot (\boldsymbol{U} \Sigma) = \nabla \cdot (\boldsymbol{R}_\Sigma) + S_\Sigma. \end{cases} \tag{1}$$

The source term included in the Σ equation is S_Σ, which comes from the complete modeling approach [1, 13]. Vaporization has not been considered in this work so far. Weber and Reynolds numbers, characterizing the flow, are supposed to be high enough to consider that the main source term is the production/destruction of surface density due to the turbulence. In this case, from the work of Duret [15], the remaining source term is

$$S_\Sigma = \frac{C_\Sigma}{\tau_t} \Sigma \left(1 - \frac{\Sigma}{\Sigma^*}\right), \tag{2}$$

with $C_\Sigma = 0.2$.

The first term on the RHS of the Σ-equation accounts instead for dispersion of the interface by turbulence ($\boldsymbol{R}_\Sigma$). Additionally, $\boldsymbol{R}_{\alpha_l}$ is the *turbulent liquid flux* that represents the transport of the liquid volume fraction induced by velocity fluctuations. These terms must be correctly modeled. A widely exploited approach is based on a gradient closure approximation, which, as explained in Sect. 3, is only valid in the absence of slip velocity between the phases. Then, the following modeling strategy is used:

$$\begin{cases} \boldsymbol{R}_U \approx \dfrac{\nu_t}{Sc_t} \nabla U + \nabla U^t \\ \boldsymbol{R}_{\alpha_l} \approx \dfrac{\nu_t}{Sc_t} \nabla \alpha_l \\ \boldsymbol{R}_\Sigma \approx \dfrac{\nu_t}{Sc_t} \nabla \Sigma, \end{cases} \tag{3}$$

where ν_t is the turbulent viscosity (or sub-grid stress in LES framework) and Sc_t is the turbulent Schmidt number.

Implementation of `ElsaBase` within the `OpenFOAM®` framework is based on the `twoLiquidMixingFoam` solver, which solves the mixing of two incompressible isothermal fluids by introducing the turbulent diffusion liquid flux within the liquid volume fraction equation. The implementation of the balance equation describing Σ with source terms for coalescence and breakup is as follows:

```
    fvScalarMatrix elsaSigmaEqn
    (
        fvm::ddt(elsaSigma)
      + fvm::div(phi,elsaSigma)
      - fvm::laplacian
        (
             dab + alphaTab*turbulence->nut(), elsaSigma,
            "laplacian(dab,elsaSigma)"
        )
        ==
(cTurbElsaSigma* (turbulence->k()/(turbulence->nut()))*
(elsaSigmaRatio))*(elsaSigmaEquil)
 +fvm::Sp(-cTurbElsaSigma* (turbulence->k()/(turbulence->nut()))*
(elsaSigmaRatio),elsaSigma)
```

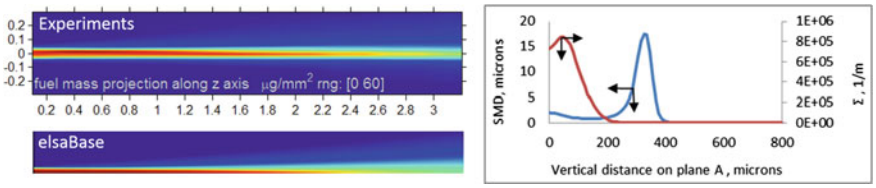


Fig. 1 Distributions of projected mass density (left) and interface density and SMD (right) for 2D axisymmetric test case

Moreover, by integrating the interface density, it is possible to calculate the mean size of the liquid-gas parcels that corresponds, for a dilute spray, to the Sauter Mean Diameter (SMD) $D_{32} = 6\alpha_l/\Sigma$. Nevertheless, to have a definition of D_{32} valid in both limits of the liquid and gas volume fraction tending to zero, the following modification is introduced:

$$D_{32} = 6\alpha_l(1-\alpha_l)/\Sigma. \tag{4}$$

To demonstrate the potential of the ELSA approach, a 2D axisymmetric injector coming from the Engine Combustion Network website (https://ecn.sandia.gov/) has been considered using a simple computational domain (a 5 degrees wedge, 2.2 mm length, and 0.8 mm radius). Boundary conditions are shown in Fig. 1. On the left of the figure is shown the liquid volume fraction field. It should be pointed out that the penetration of liquid is immediately subjected to a strong diffusion imposed by turbulent diffusion liquid flux closure. On the right of the figure is reported the interface density and the SMD on a vertical plane A. Σ is zero in the gaseous phase (above 200 μ) and has some higher values where the phases are being mixed, which it is mainly in the centerline of the liquid jet due to diffusivity effects. Therefore, SMD also experiences low values in the centerline, which is mathematically correct, and higher values when Σ decreases. This is representative of the presence of an intact liquid core, which then generates a cloud of droplets. From the contour plots, it is also possible to point out that such evolution is in line with experiments in terms of projected mass density.

3 Quasi-Multiphase Eulerian Approach

In the previous section, one of the modeling strategies behind ELSA is the use of the gradient closure for the turbulent liquid flux ($\boldsymbol{R}_{\alpha_l}$); this approach is correct only if the mean slip velocity between the liquid and the gas is zero. Indeed, if both phases are strictly nonmiscible, it is possible to consider the following definition for the turbulent liquid flux:

$$\boldsymbol{R}_{\alpha_l} = \overline{\boldsymbol{u'\alpha_l'}} = -\alpha_l \left(\boldsymbol{U} - \boldsymbol{U}_l\right) = \alpha_l \left(1 - \alpha_l\right) \boldsymbol{V}_{rlg}. \tag{5}$$

This shows the strong link between $\boldsymbol{R}_{\alpha_l}$ and the local relative velocity $\boldsymbol{V}_{rlg}$[14] that can be rearranged as

$$\boldsymbol{V}_{rlg} = \left(\boldsymbol{U}_l - \boldsymbol{U}_g - \boldsymbol{V}_D\right) = \left(\boldsymbol{U}_{slg} - \boldsymbol{V}_{Dlg}\right), \tag{6}$$

following the analysis reported in [16, 17]. It leads to the following decomposition of the turbulent liquid flux:

$$\boldsymbol{R}_{\alpha_l} = \alpha_l \left(1 - \alpha_l\right) \left(\boldsymbol{U}_{slg} - \boldsymbol{V}_{Dlg}\right) = \boldsymbol{\Upsilon}_{slg} + \boldsymbol{\Phi}_{Dlg}. \tag{7}$$

This expression shows the two main constituents of the turbulent liquid flux, $\boldsymbol{R}_{\alpha_l}$. They may be detailed as follows:

$\boldsymbol{U}_{slg}$ is the average relative velocity between the particle and the surrounding flow in the vicinity of the interface that is directly related to the drag force acting on the liquid. The transport of liquid associated with this velocity is denoted $\boldsymbol{\Upsilon}_{slg}$.
$\boldsymbol{V}_{Dlg}$ is the drift velocity. It is the conditional average of the fluid turbulent velocity fluctuations with respect to the particle distribution. The liquid transport associated with this velocity is denoted $\boldsymbol{\Phi}_{Dlg}$ [17].

For the sake of clarity, the slip velocity $\boldsymbol{U}_{slg}$ is now denoted $\boldsymbol{U}_s$ and the drift becomes $\boldsymbol{U}_D$. Similar notations apply in terms of fluxes. The drift velocity components account for the dispersion mechanism due to the particle transport by the fluid turbulent motion. It is related to a turbulent agitation that promotes homogenization of phase concentration, leading to the generation of a mean average velocity. Hence, the usual gradient models developed in the framework of single-phase flows can be adapted, such as

$$\boldsymbol{V}_D = \frac{D_{gl,t}}{\alpha_l \left(1 - \alpha_l\right)} \nabla \alpha_l, \tag{8}$$

where $D_{gl,t}$ represents the gas/liquid turbulent dispersion coefficient.

Therefore, using Eqs. 5 and 6, if the spray dynamic relaxation time τ_p and the mean effective slip velocity $\boldsymbol{U}_s$ are negligible (i.e., in the case of droplets with small Stokes numbers) the turbulent liquid flux is only due to the drift velocity, and the approach detailed in Sect. 1 can be correctly employed. However, considering the engineering applications of interest, in the near injection region, the shear stresses between the two phases are usually very high. They have a strong impact on the liquid distribution and on the development of the atomization process.

Hence, the Quasi-Multiphase Eulerian approach, which introduces into ELSA the effects of the most important physical interactions between liquid and gas phases in terms of drag, pressure gradient and body forces, has been developed. For the sake of brevity, the mathematical modeling behind QME is not detailed here, but

the reader interested in its derivation is addressed to reference [18], in which the model is thoroughly presented. Starting from the ELSA-base system, in the QME framework, the liquid volume fraction equation is formulated in order to highlight the flux contribution related to the slip ($\boldsymbol{\Upsilon}_s$) and the drift ($\boldsymbol{\Phi}_d$) fluxes, as reported below

$$\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\boldsymbol{U}\alpha_l) = -\nabla \cdot \boldsymbol{R}_{\alpha_l} = -\nabla \cdot (\boldsymbol{\Upsilon}_s + \boldsymbol{\Phi}_d) = -\nabla \cdot \left(\boldsymbol{\Upsilon}_s - \frac{\nu_t}{Sc_t} \nabla \alpha_l \right). \tag{9}$$

The drift flux is modeled using a gradient closure (Eq. 8), while for $\boldsymbol{\Upsilon}_s$, an innovative conservation equation is proposed that considers the physical interactions between the two phase flows under investigation [18]

$$\frac{\partial \rho \boldsymbol{\Upsilon}_s}{\partial t} + \nabla \cdot (\rho \boldsymbol{U} \otimes \boldsymbol{\Upsilon}_s) = \underbrace{\nabla \cdot \left(\frac{\mu_t}{Sc_t} \boldsymbol{\Upsilon}_s \right)}_{S_{turbDiff}} + \boldsymbol{S}_{\boldsymbol{\Upsilon}_s}, \tag{10}$$

where $\boldsymbol{S}_{\Upsilon_s}$ represents additional source terms such as drag, pressure gradient, and body forces. For the sake of brevity, the mathematical modeling behind these terms is not detailed here (see [18]). From a numerical point of view, the solution algorithm is the same as that presented in the previous section, with the addition of Eq. 10 just before the resolution of the liquid volume fraction equation. The resulting equation for $\boldsymbol{\Upsilon}_s$ (i.e., `reliqflux`) is reported below

```
fvVectorMatrix turbulentRelativeFluxEqn
(

   fvm::ddt(rho, reliqFlux)
 + fvm::div(rhoPhi, reliqFlux)
 - fvm::laplacian
    (
     (rho*dab + rho*alphaTab*turbulence->nut()), reliqFlux,
     "laplacian(dab,reliqflux)"
    )
 ==
 //Source term due to drag
 - fvm::Sp((rho*dragSource), reliqFlux)
 - rho*alpha1*fvc::ddt(U)
 - fvm::Sp((rho*dragSourceAlpha*fvc::ddt(alpha1)), reliqFlux)
 //Source term due to pressure gradient
 + alpha1*(1.0-alpha1)*gradP*rho/rho1
 //Source term due to body forces
 + (rho1-rho2)*alpha1*g*liqMassFrac*(1.0-liqMassFrac)
 );
```

Hence, once $\boldsymbol{\Upsilon}_s$ is calculated, it is introduced inside the α_l equation using the `MULES` structure

```
MULES::explicitSolve(alpha1, (phi+phiSlip), phiAlpha, 1, 0);
```

where `phiSlip` is the flux related to $\boldsymbol{\Upsilon}_s$, whereas `phiAlpha` is now defined as follows:

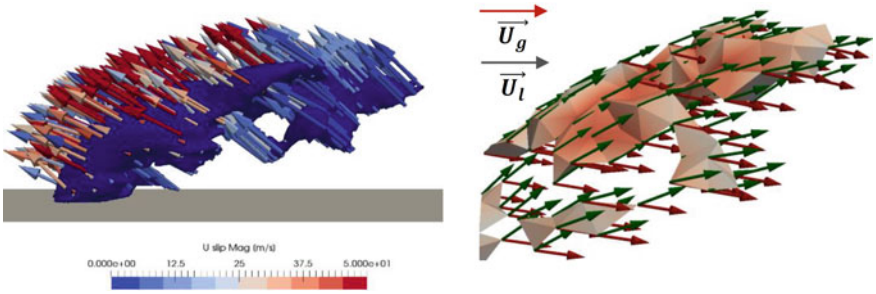


Fig. 2 Effect of Υ_s on liquid distribution (left) and vectorial comparison between liquid and gas velocity (right)

```
surfaceScalarField phiAlpha
    (
        fvc::flux
        (
            phi,
            alpha1,
            alphaScheme
        )+fvc::flux
        (
            phiSlip,
            alpha1,
            alphaSchemeSlip
        )
    );
```

The solver is now able to predict the generated slip velocity and the liquid and gas velocities, as well as their impact on the atomization process. In Fig. 2, an example of the effect of Υ_s on LES simulations of a liquid jet in an air cross-flow is shown in order to highlight that the difference in velocity between both phases tends to lift up the jet. Such an effect is mainly generated in the region where the two jets collide due to the inertia of the liquid phase.

4 ELSA-ICM Approach

Considering the known shortcomings of diffusive interface approaches in the dense spray region, and for the purpose of developing a model that is also suitable in the dilute spray region, a coupling technique between ELSA-base and an Interface Capturing Method (ICM) is proposed. Based on the `interFoam` solver, the ICM consists in defining a supplementary velocity field $\boldsymbol{U}_r$ in the vicinity of the interface, in such a way that the local flow steepens the gradient of the volume fraction, and thus the interface resolution is improved [19]. Hence, starting from the system reported (Eq. 1), the liquid volume fraction equation has been modified considering C_α as a pondering coefficient between ELSA-base and an ICM approach.

$$\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\boldsymbol{U}\alpha_l) + \underbrace{\nabla \cdot (C_\alpha \boldsymbol{U}_r \alpha_l (1-\alpha_l))}_{ICM} = (1-C_\alpha)\nabla \cdot (\boldsymbol{R}_{\alpha_l}). \tag{11}$$

The advantages of the proposed solver is to determine a resolution of the interface with ICM within a limited region, whereas it is disabled when $\boldsymbol{R}_{\alpha_l}$ prevails (i.e., when the interface fluctuations become significant at sub-grid-scale, for instance, in LES framework). Clearly, $\boldsymbol{R}_{\alpha_l}$ can be modeled using first or second order closures, as explained in previous sections (Eq. 3 or Eq. 7). The switching strategy is introduced through C_α and two different criteria, based on the interface resolution and the curvature of the interface, have been proposed to determine its value. Such a parameter is set to zero when the interface is poorly resolved (secondary atomization) and set to one otherwise (primary region). For the sake of brevity, such criteria are not detailed here, but the reader interested in this topic is addressed to reference [20]. This model is thus able to take advantage of a full interface resolution to recover a DNS formulation with ICM and to switch to a sub-grid approach when necessary. Furthermore, when the spray is formed and diluted, it is more accurate to use a regular method dedicated to WBE resolution, and therefore a Lagrangian formulation is initiated. The Lagrangian coupling has already been implemented in `OpenFOAM®`. However, the description of its theoretical and numerical details goes beyond the scope of the present paper. Further details on this point can be found in [21].

The implementation of this approach into `OpenFOAM®` is based on the well-established ICM solver for two isothermal, immiscible, incompressible phases (i.e., `interFoam`). Inside the solver, the proposed variable C_α is multiplied by the compressive term, which is implemented in the solver as follows:

```
phic = min(cAlphaField*phic , max(phic));
```

Then, its complementary value is multiplied by the Laplacian term inside the α_l equation:

```
fvScalarMatrix alpha1Eqn
  (
          - fvm::laplacian
          (
           (dTurb)*(1-cAlphaField)*(1-coeffAlphaField),
           alpha1,
           "laplacian(dab,alpha1)"
           )
   );
```

leading to the following complete formulation of the fluxes for the α_l equation that is solved through MULES:

```
      fvc::flux
      (
          phiAlphaGaz,
          alpha1,
          alphaScheme
      )
    + fvc::flux
      (
         -fvc::flux(-phir, alpha2, alpharScheme),
```

```
        alpha1,
        alpharScheme
    )
```

The capabilities of the whole approach (ICM, ELSA-base for sub-grid scales, and Lagrangian tracking) have been tested on the commercial injector available on the Engine Combustion Network website (https://ecn.sandia.gov/). In Fig. 3, an iso-surface of 0.5 of liquid volume fraction showing a C_α field is displayed. It shows the difference between the resolved scale (red color) and the poorly resolved scales downstream of the flow where the sub-grid fluctuations become important (blue color). At the inlet, there is a clear definition of the interface, thanks to the interface capturing method. On the other hand, downstream of the flow, the turbulent diffusion liquid

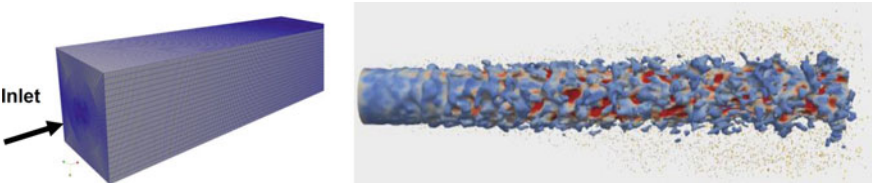


Fig. 3 3D mesh of the injection chamber (left) and iso-surface of 0.5 liquid volume fraction showing a C_α field (right)

Table 1 Nomenclature

Symbol	OpenFOAM®	Definition
α_l	alpha1	Liquid volume fraction
ρ	rho	Mixture density
$D_{gl,t}$	–	Liquid/gas turbulent dispersion coefficient
ν_t	nut	Kinematic turbulent viscosity
P	p	Pressure
Sc_t	alphaTab	Inverse Schmidt number
$\boldsymbol{S}_{\boldsymbol{\Upsilon}_s}$	-	Source terms for the $\boldsymbol{\Upsilon}_s$ equation
$\boldsymbol{R}_{\alpha_l}$	–	Turbulent liquid flux
$\boldsymbol{U}$	U	Mixture velocity
$\boldsymbol{U}_l$	Uliquid	Liquid phase velocity
$\boldsymbol{U}_g$	Ugas	Gas phase velocity
$\boldsymbol{V}_D$	–	Drift velocity
$\boldsymbol{V}_{rlg}$	–	Local relative velocity between phases
$\boldsymbol{U}_{slg}$	–	Local relative velocity between phases due to slip
$\boldsymbol{V}_{Dlg}$	–	Local relative velocity between phases due to drift
$\boldsymbol{\Upsilon}_{slg}$	reliqflux	Turbulent flux due to slip between phases
$\boldsymbol{\Phi}_{Dlg}$	–	Turbulent flux due to drift between phases
$\boldsymbol{U}_r$	phic	Local velocity used for the Compressive term
C_{α_l}	cAlphaField	Switching parameter between the ICM and sub-grid scales

flux dominates, and it is perceived as the beginning of the mixing of the phases, along with a cone-shaped cloud of particles around the liquid jet due to the generation of Lagrangian particles as the liquid volume fraction decreases.

In Table 1, the most important variables shown in the present work, together with their definitions in `OpenFOAM®` are reported.

Acknowledgements We acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources and support. A large part of the results reported here have been also obtained using TGCC-Curie, CRIHAN, and GENCI (IDRIS) supercomputers. They are gratefully acknowledged.

References

1. R. Lebas, T. Menard, P.A. Beau, A. Berlemont, and F.X. Demoulin. Numerical simulation of primary break-up and atomization: DNS and modelling study. *International Journal of Multiphase Flow*, 35(3):247–260, 2009.
2. T. Menard, S. Tanguy, and A. Berlemont. Coupling level set/vof/ghost fluid methods: Validation and application to 3d simulation of the primary break-up of a liquid jet. *International Journal of Multiphase Flow*, 33(5):510–524, 2007.
3. J. Shinjo and A. Umemura. Simulation of liquid jet primary breakup: Dynamics of ligament and droplet formation. *International Journal of Multiphase Flow*, 36(7):513–532, 2010.
4. J. Shinjo and A. Umemura. Surface instability and primary atomization characteristics of straight liquid jet sprays. *International Journal of Multiphase Flow*, 37(10):1294–1304, 2011.
5. F. A. Williams. Spray combustion and atomization. *Physics of Fluids*, 1(6):541–545, 1958.
6. G.A. Bird. *Molecular gas dynamics and the direct simulation of gas flows*. Oxford University Press, 1994.
7. Frédérique Laurent and Marc Massot. Multi-fluid modelling of laminar polydisperse spray flames: origin, assumptions and comparison of sectional and sampling methods. *Combustion Theory and Modelling*, 5(4):537–572, 2001.
8. Frédérique Laurent, Alaric Sibra, and François Doisneau. Two-size moment Eulerian multi-fluid model: a flexible and realizable high-fidelity description of polydisperse moderately dense evaporating sprays. June 2015.
9. C. Yuan, F. Laurent, and R.O. Fox. An extended quadrature method of moments for population balance equations. *Journal of Aerosol Science*, 51:1–23, 2012.
10. D. Kah, F. Laurent, M. Massot, and S. Jay. A high order moment method simulating evaporation and advection of a polydisperse liquid spray. *Journal of Computational Physics*, 231(2):394–422, 2012.
11. Michael Frenklach. Method of moments with interpolative closure. *Chemical Engineering Science*, 57(12):2229–2239, 2002. Population balance modelling of particulate systems.
12. Stephen L. Passman Donald A. Drew. *Theory of Multicomponent Fluids*. Springer-Verlag New York, 1999.
13. Ariane Vallet and Roland Borghi. Modélisation eulerienne de l'atomisation d'un jet liquide. *Comptes Rendus de l'AcadÃl'mie des Sciences - Series IIB - Mechanics-Physics-Astronomy*, 327(10):1015–1020, 1999.
14. F.X. Demoulin, P.A. Beau, G. Blokkeel, A. Mura, and R. Borghi. A new model for turbulent flows with large large density fluctuations: application to liquid atomization. *Atomization and Sprays*, 17(4):315–345, 2007.
15. B Duret, J Reveillon, T Menard, and FX Demoulin. Improving primary atomization modeling through dns of two-phase flows. *International Journal of Multiphase Flow*, 55:130–137, 2013.

16. E. Deutsch and O. Simonin. Large eddy simulation applied to the motion of particles in stationary homogeneous fluid turbulence. *Turbulence Modification in Multiphase Flows-ASME FED*, 110(35), 1991.
17. O. Simonin. Statistical and continuum modelling of turbulent reactive particulate flows. *Lecture Series 1996–02, Von Karman Institute for Fluid Dynamics*, 2000.
18. Antonio Andreini, Cosimo Bianchini, Stefano Puggelli, and F.X. Demoulin. Development of a turbulent liquid flux model for eulerian eulerian multiphase flow simulations. *International Journal of Multiphase Flow*, 81:88–103, 2016.
19. F Raees, DR Van der Heul, and Cornelis Vuik. Evaluation of the interface-capturing algorithm of OpenFOAM® for the simulation of incompressible immiscible two-phase flow. Technical report, Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft Institute of Applied Mathematics, 2011.
20. Nicolas Hecht. *Simulation aux grandes échelles des écoulements liquide-gaz : application á l'atomisation*. PhD thesis, University of Rouen, 2014.
21. Romain Lebas, Pierre-Arnaud Beau, Gregory Blokkeel, and Francois-Xavier Demoulin. Elsa model for atomization: To benefit of the eulerian and lagrangian descriptions of the liquid phase. In *Proceedings of ASME Fluids Engineering Division Summer Meeting*, volume 2006, 2006.

Lubricated Contact Model for Cold Metal Rolling Processes

Vanja Škurić, Peter De Jaeger and Hrvoje Jasak

Abstract A numerical method for calculating lubricated contact pressures and friction in cold metal rolling is presented in this study. In order to have a good representation of the contact phenomena in lubricated metal rolling processes, the interaction between the surface roughness and lubricant flow has to be taken into account. Due to the changes in lubricant thickness during the rolling process, the lubricant flows in four local regimes: hydrodynamic thick film, hydrodynamic thin film, mixed and boundary lubrication regimes. The ability to treat all four lubrication regimes is required. Surface roughness effects, lubrication regimes treatment and lubricant property variations are all implemented within the present model. In order to calculate contact pressures and frictional forces, the Greenwood-Williamson model with the modified Reynolds lubrication equation is used. The Finite Area Method is used to discretize the Reynolds lubrication equation over a curved surface mesh. The implemented model is used as a solid contact boundary condition for a large strain hyperelastoplastic deformation solver developed in the `foam-extend` framework. The model is tested on wire and sheet rolling cases, and the results are presented here.

1 Introduction

Lubrication and friction are important factors in metal forming processes, since unoptimized frictional parameters can result in lower productivity of the rolling machinery and deteriorate the surface quality of the product [3]. Lubrication is a

V. Škurić (✉) · H. Jasak
Faculty of Mechanical Engineering and Naval Architecture,
Ivana Lučića 5, Zagreb, Croatia
e-mail: vanja.skuric@fsb.hr

H. Jasak
e-mail: hrvoje.jasak@fsb.hr

P. De Jaeger
NV Bekaert SA, Bekaertstraat 2, 8550 Zwevegem, Belgium
e-mail: peter.dejaeger@bekaert.com

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_23

complex problem in wire rolling and drawing processes, due to the necessary coupling between the surface asperity contact and lubricant flow [13].

The Finite Element Method (FEM) has been widely used for solving wire and sheet rolling problems. Liu et al. [14] use an elastoplastic finite element technique in the arbitrary Lagrangian–Eulerian form for simulating the cold rolling of a strip. The friction layer technique is employed to model the frictional effects. Boman and Ponthot [3] use an extended hydrodynamic lubrication procedure and the arbitrary Lagrangian–Eulerian formulation for strip rolling simulations. Khan et al. [13] use the Finite Element Method with the equivalent interfacial layer in order to capture the contact pressure and traction during a lubricated sheet rolling simulation by using a statistical Hertz-based asperity contact model and modified Reynolds equation. Wu et al. [19] use a similar approach as in [13] and perform a three-dimensional sheet rolling simulation. A comparison of contact and hydrodynamic pressures for different lubricant viscosities and rolling speeds is given, as well as a comparison with the experimental results.

In the past 20 years, the Finite Volume Method (FVM) has become a noteworthy alternative to the Finite Element Method in the field of solid mechanics. Examples of FVM applications to problems of solid mechanics are linear elasticity [11], crack propagation [8], fluid–structure interaction [18], finite strain elastoplasticity [2], etc. Recently, a finite volume large strain hyperelastoplastic deformation solver was developed by Cardiff et al. [4]. The solver was tested on several cases and showed very good agreement with the FEM results. Implementation and application of a lubricated rough surface contact model using the Finite Volume Method are not reported in the literature to the authors' knowledge.

In this study, a finite volume model for lubricated rough surface contact is presented. The model is implemented as a solid contact boundary condition inside the `foam-extend`, a community-driven fork of `OpenFOAM`® [12]. Sheet and wire rolling simulations are performed and the results are presented.

2 Mathematical Model

The thin-film lubricant flow between two rough surfaces in contact can be divided into four local regimes [3]: hydrodynamic thick film, hydrodynamic thin film, mixed and boundary lubrication regimes (Fig. 1). In the hydrodynamic regime, two surfaces are completely separated by the lubricant and the total contact pressure is equal to the hydrodynamic pressure of the lubricant. The difference between the thick and thin hydrodynamic regimes are that in the thick film regime, surface roughness does not influence the film's flow, while in the thin film regime, the roughness has a significant influence on the flow characteristics. In the mixed lubrication regime, contact pressure is shared between the asperities in contact and the lubricant. In the boundary regime, almost, the whole contact pressure is carried by the asperities, while the lubricant is found in traces in asperity valleys. In order to take into account all four lubrication regimes, the total contact pressure is divided into the asperity

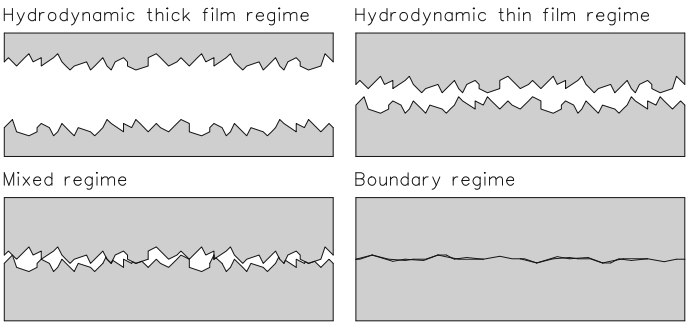


Fig. 1 The four lubrication regimes

contact pressure and the lubricant hydrodynamic pressure [19]

$$P_n = A_r P_a + (1 - A_r) P_f, \tag{1}$$

where P_n is the total contact pressure, P_a is the asperity contact pressure, P_f is the lubricant hydrodynamic pressure and A_r represents the ratio of asperity contact area to the nominal area. Correspondingly, the shear traction can be described by [13, 19]

$$P_t = A_r \tau_a + (1 - A_r) \tau_f, \tag{2}$$

where P_t is the total shear traction, τ_a is the asperity shear traction and τ_f is the lubricant shear stress.

In order to calculate the asperity contact pressure and the lubricant hydrodynamic pressure, appropriate models should be utilized. Since surface roughness properties are important factors during the metal rolling and drawing processes [3], both the asperity and lubricant models have to take them into account.

2.1 Asperity Contact Model

The real surfaces are rough on the microscopic scale [13], having a large number of different-sized asperities. In order to represent the interaction between asperities of two surfaces in contact during a wire rolling simulation, an asperity contact model is required. If the deformation of asperities is considered to be elastic, a widely used Greenwood-Williamson (GW) [6] model can be utilized as a contact model for rough surfaces [9]. The Greenwood-Williamson model is a statistical contact model based on the Hertzian theory [6], in which the surface roughness is modelled as a cluster of hemispherical asperities. The GW model has the following assumptions [5]:

- The rough surface is considered isotropic.
- Asperities have a spherical shape near their summits.

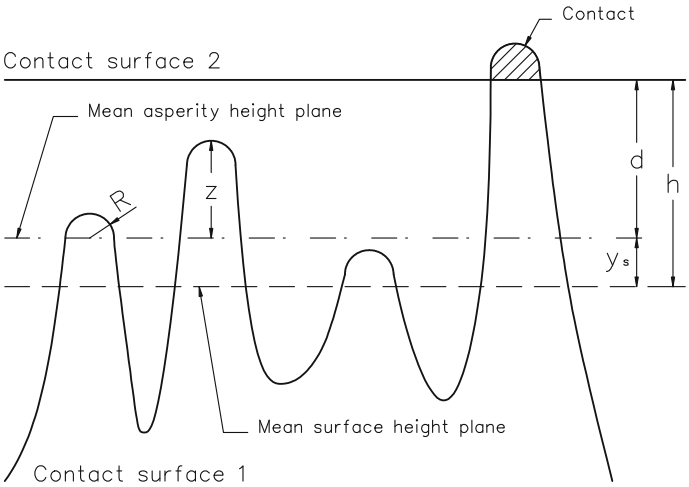


Fig. 2 Greenwood-Williamson rough surface contact geometry

- All asperity summits have the same radius R, but their heights vary based on the asperity heights distribution function.
- There is no interaction between asperities.
- Bulk deformation is not considered in the model, only asperity deformation.
- Deformation of the asperities is purely elastic.

Figure 2 depicts the geometry of a GW rough surface contact. The two rough surfaces in contact are modelled as one smooth surface and one equivalent rough surface with the Root Mean Squared (RMS) roughness is defined by

$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}, \tag{3}$$

where σ_1 and σ_2 are the roughnesses of surface 1 and 2, respectively.

Two reference planes are defined: the mean asperity height plane and the mean surface height plane. The former is usually used in the asperity-based contact models and the latter is more practically obtained experimentally [5]. Asperity height z and separation d are measured from the mean asperity height plane, while distance h is measured from the plane defined by surface heights. The distance between the two respective planes is denoted by y_S.

In order to simplify a complex surface topography, Greenwood and Williamson [6] used three surface parameters: the asperity density η (the number of asperities per unit area), the standard deviation of asperity heights σ_s and the radius of asperity summit R. Determining respective parameters from surface roughness profiles was introduced in [15], in which parameters were expressed by three spectral moments: m_0, m_2, m_4. The spectral moments are determined from a surface profile as

$$m_0 = \frac{1}{n}\sum_{i=1}^{n} y^2(x), \tag{4a}$$

$$m_2 = \frac{1}{n}\sum_{i=1}^{n} \left(\frac{\mathrm{d}y(x)}{\mathrm{d}x}\right)^2, \tag{4b}$$

$$m_4 = \frac{1}{n}\sum_{i=1}^{n} \left(\frac{\mathrm{d}^2y(x)}{\mathrm{d}x^2}\right)^2, \tag{4c}$$

where $y(x)$ is a height deviation from the mean surface plane at location x on the examined surface and n is the total number of measured points. The surface parameters are defined by [15]

$$\sigma = \sqrt{m_0}, \tag{5a}$$

$$R = 0.375\sqrt{\frac{\pi}{m_4}}, \tag{5b}$$

$$\eta = \frac{m_4}{6\pi\sqrt{3}\,m_2}. \tag{5c}$$

Using the surface parameters, the standard deviation of asperity heights is defined by [5]

$$\sigma_s = \sqrt{\sigma^2 - \frac{3.717 \times 10^{-4}}{(\eta R)^2}}, \tag{6}$$

and the distance y_s between the two mean planes is defined by

$$y_s = \frac{0.25}{\pi\sqrt{3}R\eta}. \tag{7}$$

Contact area and force between a single hemispherical asperity and a flat surface is defined by the Hertz elastic solution [10]

$$\overline{A} = \pi R(z - d), \tag{8a}$$

$$\overline{F} = \frac{4}{3}E'\sqrt{R}(z - d)^{1.5}, \tag{8b}$$

where the equivalent Young's modulus is defined as

$$E' = \frac{1 - \nu_1^2}{E_1} + \frac{1 - \nu_2^2}{E_2}. \tag{9}$$

E_1, E_2 and ν_1, ν_2 are Young's modulus and Poisson's ratio of surfaces 1 and 2, respectively.

Following the GW model, the total contact area and force between two surfaces is found by integrating a single asperity contact area and force over the entire range of asperities' heights

$$A = \eta A_n \int_d^\infty \overline{A}\phi(z)\mathrm{d}z, \tag{10a}$$

$$F = \eta A_n \int_d^\infty \overline{P}\phi(z)\mathrm{d}z, \tag{10b}$$

where A_n is the nominal contact area and $\phi(z)$ is the asperity height distribution function. If the asperity height distribution is Gaussian, the distribution function is defined by [10]

$$\phi(z) = \frac{1}{\sigma_s\sqrt{2\pi}} \exp\left(-0.5\left(\frac{z}{\sigma_s}\right)^2\right). \tag{11}$$

By inserting Eqs. 8a and 8b into Eqs. 10a and 10b, respectively, the final expressions for a Greenwood-Williamson rough surface contact area and force are given by

$$A = \eta A_n \pi R \int_d^\infty (z-d)\phi(z)\mathrm{d}z, \tag{12a}$$

$$F = \frac{4}{3}\eta A_n E'\sqrt{R} \int_d^\infty (z-d)^{1.5}\phi(z)\mathrm{d}z. \tag{12b}$$

Up until recently, the contact area and force integrals (Eqs. 12a and 12b) need to be numerically integrated, however, analytical solutions to the corresponding integrals were found in [9], which eliminated the need for numerical integration. Finally, the ratio of asperity contact area and unit nominal area A_r and the asperity contact pressure P_a (Eq. 1) are defined by

$$A_r = \frac{A}{A_n}, \tag{13a}$$

$$P_a = \frac{F}{A_n}. \tag{13b}$$

If the Coulomb friction law is used, the asperity shear traction is defined by [19]

$$\tau_a = \mu_a P_a, \tag{14}$$

where μ_a is the Coulomb friction coefficient.

2.2 Lubricant Flow Model

In order to calculate the lubricant hydrodynamic pressure and the shear stress between two rough surfaces in contact and in relative motion to each other, an appropriate thin-film flow model has to be utilized. Patir and Cheng [16] developed the modified Reynolds equation, which takes into account the surface roughness during the thin-film flow. The main assumptions of the Reynolds equation are [7]:

- Fluid is considered to be Newtonian.
- Fluid viscous forces are dominant, while the fluid body, inertia and surface tension forces can be neglected.
- Fluid film curvature can be neglected, since the thickness of the fluid is much smaller than the width and length of the film.
- The variation of pressure across the fluid film is negligibly small.

The modified Reynolds equation is a 2D partial differential pressure equation defined by [3, 16]

$$\nabla_s \bullet \left(\phi_{xy} \frac{\rho h^3}{12\mu} \nabla_s p \right) = \nabla_s \bullet \left[\rho h_T \frac{\mathbf{U_1} + \mathbf{U_2}}{2} \right] + \nabla_s \bullet \left[\rho \sigma \phi_s \frac{\mathbf{U_1} - \mathbf{U_2}}{2} \right] + \frac{\partial (\rho h)}{\partial t}, \tag{15}$$

where ∇_s, $\nabla_s \bullet$ are the surface gradient and divergence operators, respectively, p is the hydrodynamic pressure, h is the surface separation (Fig. 3), h_T is the film flow thickness, $\mathbf{U_1}$, $\mathbf{U_2}$ are the tangential velocity vectors of surface 1 and 2, respectively, σ is the equivalent RMS surface roughness of the two surfaces in contact, ϕ_{xy} and ϕ_s are the pressure and shear flow factors, respectively, ρ is the density and μ is the viscosity of the fluid.

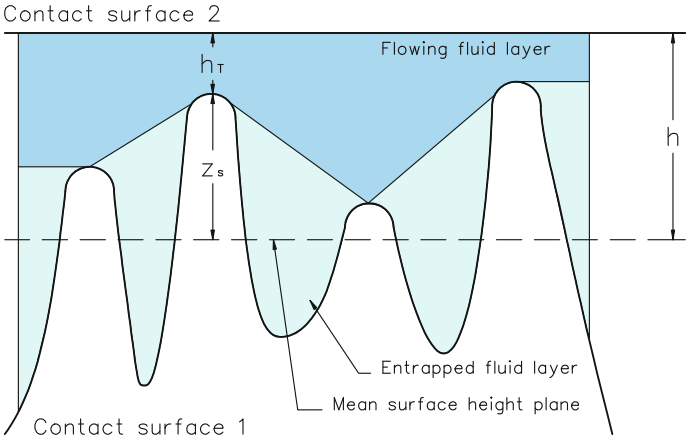


Fig. 3 Rough surface film flow geometry

There are two layers of fluid film between the surfaces [13]: the entrapped fluid layer and the flowing fluid layer (Fig. 3). Since only the fluid between asperity tips and the opposite surface can flow (the flowing fluid layer), the film flow thickness (Fig. 3) is defined by [19]

$$h_T = \int_{-\infty}^{h} (h - z_s)\phi(z_s)\mathrm{d}z_s, \tag{16}$$

where $\phi(z_s)$ is the surface height distribution defined by

$$\phi(z_s) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-0.5\left(\frac{z_s}{\sigma}\right)^2\right). \tag{17}$$

The pressure flow factor ϕ_{xy} represents the influence of surface roughness on the hydrodynamic pressure [16]. If the surface is isotropic, the pressure flow factor is defined by

$$\phi_{xy} = 1 - 0.9\exp\left(-0.56\frac{h}{\sigma}\right). \tag{18}$$

The shear flow factor ϕ_s represents the influence of rough surface sliding effects on the film fluid flow, and is defined in [17].

Following [17], the fluid film shear stress vector acting on surfaces contact is defined by

$$\boldsymbol{\tau}_f = \pm\frac{\mu(\mathbf{U_2} - \mathbf{U_1})}{h}\left[\phi_f - \phi_{fs}\left(1 - 2\frac{\sigma_1}{\sigma}\right)\right], \tag{19}$$

where ϕ_f is a factor resulting from the averaging of the sliding velocity component of the shear stress, ϕ_{fs} is the shear stress factor and σ_1 is the RMS roughness of surface 1. The plus sign is relative to the shear stress vector acting on surface 1, while the minus sign refers to surface 2.

2.3 Implementation of Numerical Models

The asperity contact model and lubricant film flow model, described in Sects. 2.1 and 2.2, are implemented as a solid contact boundary condition for a large strain hyper-elastoplastic deformation solver developed by Cardiff et al. [4] in `foam-extend`, a community-driven fork of `OpenFOAM`® [12].

Calculation of the contact pressures (Eq. 13b) and contact areas (Eq. 13a) is performed by a written utility for a selected number (2000 in this study) of surface separations in a range from 0 to 6σ before the start of the simulation. The calculated values, with their respective separations, are written into a table. During the simulation runtime, the contact pressure and area values are interpolated from the table data based on the current surface separation values. This procedure reduces the

required CPU time compared to the run-time calculations of contact pressures and areas, especially in the case of contact models in which analytical solutions are not available, thus requiring numerical integration of the pressure and area equations. In this study, the Greenwood-Williamson contact model (Sect. 2.1) is implemented using the analytical solutions from Jackson and Green [9]. The required model inputs are surface roughness σ, asperity radius R and asperity density η.

The modified Reynolds equation (Sect. 2.1) is discretized over a curved surface mesh using the Finite Area Method, and is solved for the hydrodynamic pressure. The model inputs are surface roughness σ, lubricant density ρ and viscosity μ. Local separation between the surfaces and the film flow thickness is calculated during the runtime.

The distance between two surfaces in contact, i.e. two contact patches of the computational mesh, is calculated using the `GGI` interface (General Grid Interface [1]) for every patch point. The point values are then interpolated to the face centres of the contact patch.

During the runtime, at the end of every inner iteration, the total contact pressure (Eq. 1) and the total shear traction (Eq. 2) are calculated for every face of the contacting patches, and are used as boundary conditions for the deformation solver. The number of inner iterations, i.e. iterations inside a single time-step, is limited by a deformation convergence criteria (10^{-7} in this study).

3 Results and Discussion

The implemented models were tested in a three-dimensional sheet and wire rolling cases. The mechanical and lubricant properties are specified in Table 1. The rollers are considered rigid in both cases. The simulations were performed for several different roller speeds, reduction ratios and lubricant viscosities. Cases are set up as axisymmetric, in which quarter of a sheet (or wire) and one half of a single roller are simulated.

3.1 Sheet Rolling

A 3D case of metal sheet rolling is presented in this section. Sheet dimensions are length 100 mm, width 16 mm and height 8 mm. Roller diameter is 158 mm and thickness 19.2 mm. The computational mesh of the sheet (Fig. 4) consists of 25600 hexahedral cells, while the roller is considered rigid. Simulations were run with three different lubricant viscosities (0.5, 2 and 5 Pas), three sheet thickness reductions (10, 20 and 30%) and three roller speeds (60, 120 and 240 RPM). The pressure variations in Figs. 5, 6 and 7 are measured along the axisymmetric plane.

Figure 5 shows the significant increase of maximum hydrodynamic pressure at the inlet of the rolling bite when using higher viscosity lubricant. By increasing the

Table 1 Mechanical and lubricant properties

Name	Symbol	Value	Units
Initial material density	ρ_m	7800	kg/m^3
Young's modulus	E	177	GPa
Poisson's ratio	ν	0.3	–
Initial yield stress [4]	σ_Y	1.3	GPa
Surface roughness [19]	σ	8.0	μm
Asperity radius [19]	R	6.0	μm
Asperity density [19]	η	1.6×10^{10}	m^{-2}
Coulomb friction coeff.	μ_a	0.3	–
Lubricant viscosity	μ_L	0.5, 2.0, 5.0	Pas
Lubricant density	ρ_L	878	kg/m^3
Hardening [4]	Plastic strain	Yield stress (GPa)	
	0.00	1.30	
	0.01	1.50	
	0.10	1.69	
	0.50	1.64	
	0.88	2.11	

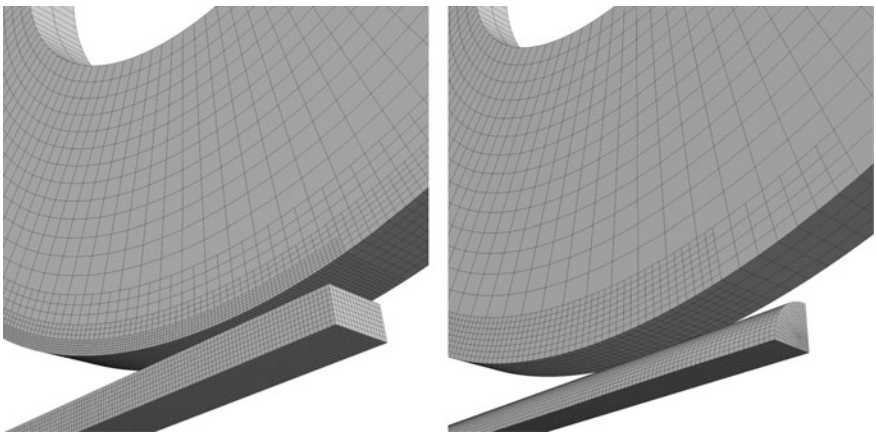


Fig. 4 Sheet and wire rolling meshes

viscosity from 0.5 to 2 Pas, the maximum value of hydrodynamic pressure increases from 1 to 5 MPa (400%). By increasing the lubricant viscosity to 5 Pas, the maximum hydrodynamic pressure further increases by 140%. Increasing the viscosity, and subsequently, the hydrodynamic pressure, has almost no effect on the asperity contact pressure (Fig. 5), due to the fact that the asperity contact pressure is three orders of magnitude larger than the lubricant hydrodynamic pressure.

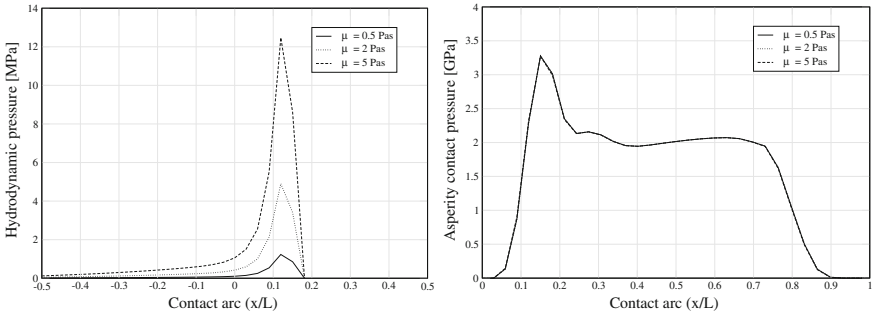


Fig. 5 Sheet case: variations of hydrodynamic and asperity contact pressures due to viscosity changes (roller speed 120 RPM, 20% reduction)

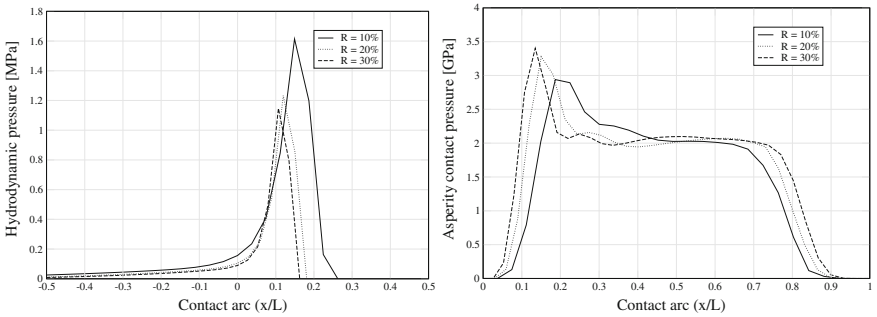


Fig. 6 Sheet case: variations of hydrodynamic and asperity contact pressures due to sheet thickness reduction changes (viscosity 0.5 Pas, roller speed 120 RPM)

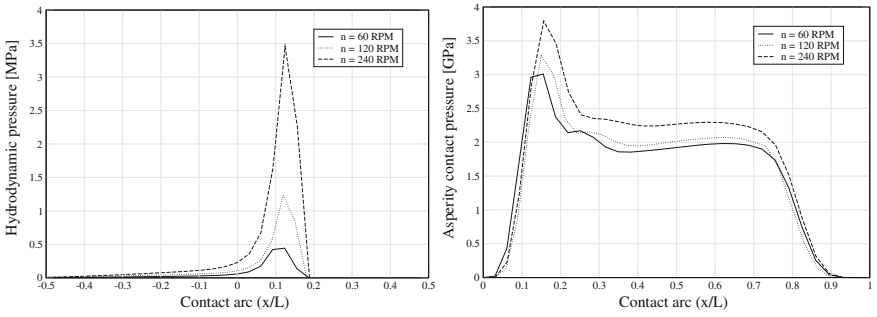


Fig. 7 Sheet case: variations of hydrodynamic and asperity contact pressures due to roller speed changes (viscosity 0.5 Pas, thickness reduction 20%)

Figure 6 shows the influence of the thickness reduction parameter on the hydrodynamic and asperity contact pressure. By increasing reduction from 10 to 20%, the hydrodynamic pressure increases from 1.18 to 1.22 MPa (3.5%). A further increase of reduction to 30% similarly increases hydrodynamic pressure to 1.6 MPa (31%). By increasing the reduction, the high-pressure area expands, with the location of maximum hydrodynamic pressure moving into the rolling bite. By increasing the reduction, the area of the high asperity contact pressure expands (Fig. 6), with the maximum value of pressure increasing by 17% with 30% reduction, compared to the 10% reduction.

The maximum value of hydrodynamic pressure increases from 0.4 to 1.25 MPa (310%) with an increase in the roller speed from 60 to 120 RPM (Fig. 7). The asperity contact pressure also increases from 5 to 8% along the whole area of asperity contact. By further increasing the roller speed to 240 RPM, the maximum hydrodynamic pressure increases by 280%, and the asperity contact pressure by 12–15% on the whole contact area.

3.2 Wire Rolling

A 3D case of metal wire rolling is presented in this section. The wire is 100 mm long and 8 mm in diameter. The roller's diameter is 158 mm and its thickness 12 mm. The computational mesh of the wire (Fig. 4) consists of 6400 hexahedral cells, while the roller is considered rigid. Simulations were run with three different lubricant viscosities (0.5, 2 and 5 Pas), two sheet thickness reductions (20 and 30%) and two roller speeds (120 and 240 RPM).

Figure 8 shows a significant increase in the maximum hydrodynamic pressure with the higher viscosity lubricant. The maximum hydrodynamic pressure with lubricant viscosity of 5 Pas is 4.7 MPa, which is an increase of 147% compared to the 2 Pas viscosity case. Similar to the sheet rolling case, viscosity changes have negligible influence on the asperity contact pressure (Fig. 8). The high asperity contact area has a well-known shape of a horseshoe [4].

The influence of the wire thickness reduction is presented in Fig. 9. The maximum hydrodynamic pressure decreases from 0.468 to 0.371 MPa (21%) when the thickness reduction increases from 20 to 30%. The maximum asperity contact pressure has the same trend as the hydrodynamic pressure, decreasing from 5.68 to 5.48 GPa (3.5%). These pressure decreases can be explained by the fact that, with the higher thickness reduction, the asperity contact area becomes longer and wider (Fig. 9), so contact forces act over a larger area, resulting in lower maximum pressure values.

By increasing the rolling speed (Fig. 10) from 120 to 240 RPM, the maximum value of hydrodynamic pressure increases from 0.477 to 0.952 MPa (100%). Two areas of high hydrodynamic pressure can be observed at the rolling speed of 240 RPM: the first one at the inlet of the rolling bite and the second one in the middle of the bite. The maximum value of the asperity contact pressure also increases from 5.73 to 6.73 GPa (17%).

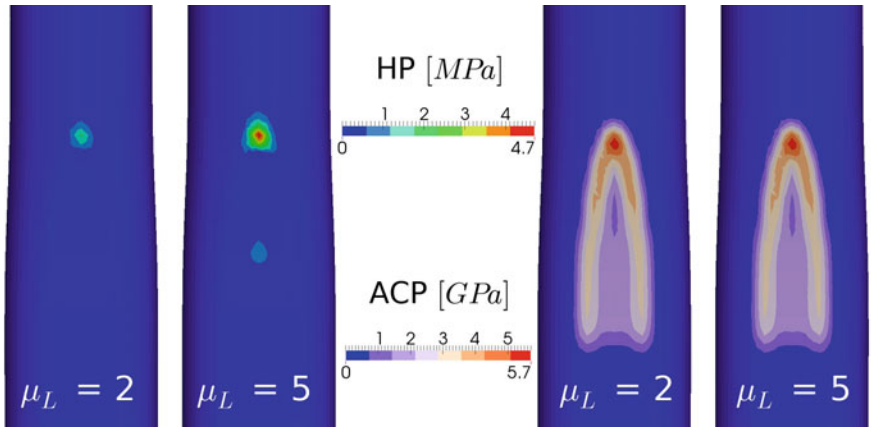


Fig. 8 Wire case: variations of hydrodynamic and asperity contact pressures due to viscosity changes (roller speed 120 RPM, thickness reduction 20%)

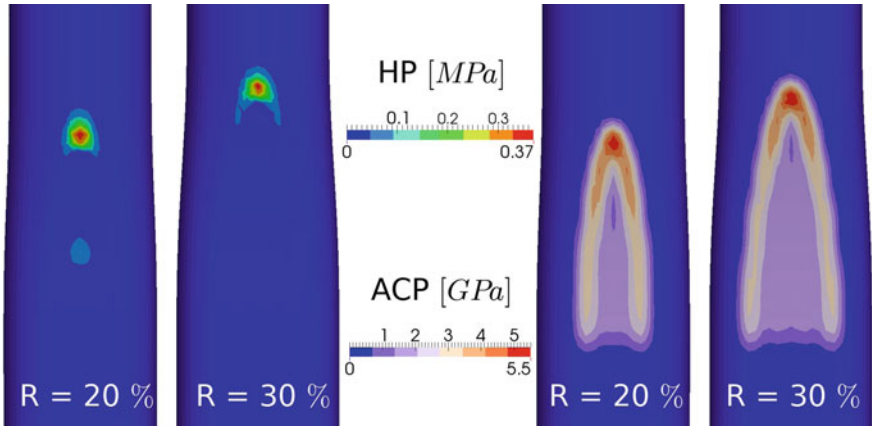


Fig. 9 Wire case: variations of hydrodynamic and asperity contact pressures due to thickness reduction changes (viscosity 0.5 Pas, roller speed 120 RPM)

4 Conclusion

A rough surface contact model with lubricant flow was presented in this study. The model was implemented as a solid contact boundary condition for a large strain hyperelastoplastic deformation solver [4] in the `foam-extend` framework. The Greenwood-Williamson model was used as an asperity contact model, and the modified Reynolds equation for the lubricant flow.

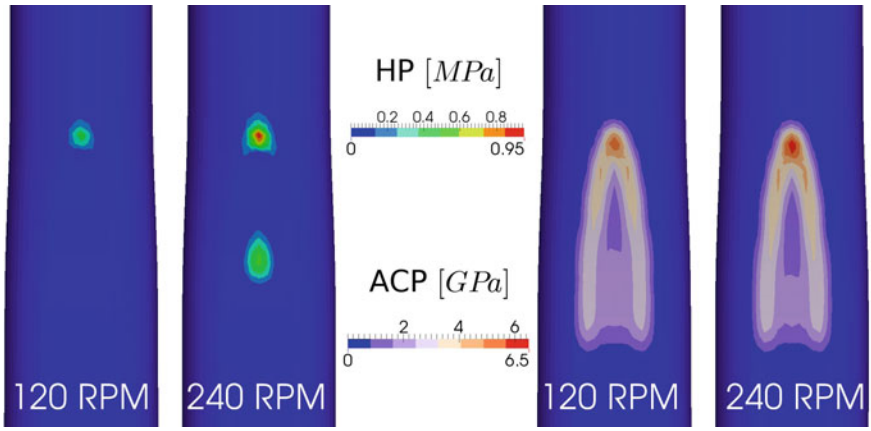


Fig. 10 Wire case: variations of hydrodynamic and asperity contact pressures due to roller speed changes (viscosity 0.5 Pas, thickness reduction 20%)

The model was tested in two cases: sheet rolling and wire rolling. The influence of viscosity, roller speed and thickness reduction changes on the hydrodynamic and asperity contact pressures was presented. By increasing the lubricant viscosity and roller speed, the hydrodynamic pressure increases significantly. Changes in the hydrodynamic pressure have negligible influence on the asperity contact pressure, due to a significant difference in their maximum values. With the thickness reduction increase in the wire rolling case, the contact area also increases, resulting in a decrease in the maximum pressure value. The shape of asperity contact pressure in the sheet rolling simulations shows a similar trend to that of the finite element simulations and experimental data [19], while the contact pressure in the wire rolling simulations has a well-known shape of a horseshoe [4]. The area of the highest hydrodynamic pressure, both in the wire and sheet rolling simulations, is at the inlet of the rolling bite, which is expected, due to the sudden reduction of film thickness in that area (converging part of the contact). In order to determine the accuracy of the model, a detailed validation will be performed in future work, in which the important characteristics of wire rolling and drawing processes (roller forces and torques, wire thickness, film temperatures, etc.) will be compared to the experimental data.

In order to possibly improve the contact physics of the model, an appropriate elastoplastic contact method should be implemented, along with a heat transfer model taking into account the lubricant flow.

Acknowledgements Financial support via Ph.D. funding is gratefully acknowledged from Peter De Jaeger and Bekaert.

References

1. M. Beaudoin and H. Jasak. Development of a generalized grid interface for turbomachinery simulations with OpenFOAM®. In *Open Source CFD International Conference 2008*, Berlin, 2008.
2. I. Bijelonja, I. Demirdžić, and S. Muzaferija. A finite volume method for large strain analysis of incompressible hyperelastic materials. *International Journal for Numerical Methods in Engineering*, 64(12):1594–1609, 2005.
3. R. Boman and J.P. Ponthot. Finite element simulation of lubricated contact in rolling using the arbitrary lagrangian-eulerian formulation. *Comput. Methods Appl. Mech. Engrg.*, 193:4323–4353, 2004.
4. P. Cardiff, Ž. Tuković, P. De Jaeger, M. Clancy, and A. Ivanković. A lagrangian cell-centred finite volume method for metal forming simulation. *International Journal for Numerical Methods in Engineering*, pages n/a–n/a, 2016. nme.5345.
5. W.R. Chang, I.I. Etsion, and D.B. Bogy. An elastic-plastic model for the contact of rough surfaces. *Journal of Tribology*, 109(2):257–263, 1987.
6. J. A. Greenwood and J. B. P. Williamson. Contact of nominally flat surfaces. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 295(1442):300–319, 1966.
7. B.J. Hamrock, S.R. Schmid, and B.O. Jacobson. *Fundamentals of Fluid Film Lubrication*. Marcel Dekker, 2004.
8. A. Ivanković and G.P. Venizelos. Rapid crack propagation in plastic pipe: predicting full-scale critical pressure from s4 test results. *Engineering Fracture Mechanics*, 59(5):607–622, 1998.
9. R. L. Jackson and I. Green. On the modeling of elastic contact between rough surfaces. *Tribology Transactions*, 54(2):300–314, 2011.
10. Robert L. Jackson and Itzhak Green. A statistical model of elasto-plastic asperity contact between rough surfaces. *Tribology International*, 39(9):906–914, 2006.
11. H. Jasak and H. Weller. Finite volume methodology for contact problems of linear elastic solids. In *Proceedings of 3rd International Conference of Croatian Society of Mechanics*, Dubrovnik, Croatia, 2000.
12. Hrvoje Jasak. OpenFOAM®: Open source cfd in research and industry. *International Journal of Naval Architecture and Ocean Engineering*, 1(2):89–94, 2009.
13. M.N. Khan, H. Ruan, L.C. Zhang, X.M. Zhao, and X.M. Zhang. A new approach to the investigation of mixed lubrication in metal strip rolling. In *Proc. 7 Australasian Congress on Applied Mechanics, ACAM 7*, Adelaide, Australia, 2012.
14. C. Liu, P. Hartley, C.E.N. Sturgess, and G.W. Rowe. Simulation of the cold rolling of strip using an elastic-plastic finite element technique. *International Journal of Mechanical Sciences*, 27(11):829–839, 1985.
15. J.I. McCool. Relating profile instrument measurements to the functional performance of rough surfaces. *Journal of Tribology*, 109(2):264–270, 1987.
16. N. Patir and H.S. Cheng. An average flow model for determining effects of three-dimensional roughness on partial hydrodynamic lubrication. *J. of Lubrication Tech.*, 100(1):12–17, 1978.
17. N. Patir and H.S. Cheng. Application of average flow model to lubrication between rough sliding surfaces. *J. of Lubrication Tech.*, 102(2):220–229, 1979.
18. Ž. Tuković, P. Cardiff, A. Karač, H. Jasak, and A. Ivanković. OpenFOAM® library for fluid-structure interaction. In *9th OpenFOAM® Workshop*, Zagreb, 2014.
19. C. Wu, L. Zhang, S. Li, Z. Jiang, and P. Qu. A novel multi-scale statistical characterization of interface pressure and friction in metal strip rolling. *International Journal of Mechanical Sciences*, 89:391–402, 2014.

Modeling of Turbulent Flows in Rectangular Ducts Using OpenFOAM®

Raquel Faria, Almerindo D. Ferreira, A. M. G. Lopes and Antonio C. M. Sousa

Abstract The present work aims to verify the applicability of *Irwin* probes and *Preston* tubes to turbulent incompressible flow in smooth rectangular ducts. For the experimental apparatus, an aspect ratio of 1:2 is considered and tests are conducted for *Reynolds* number values within the range of 10^4 to 9×10^4. Local friction coefficients are determined based on the pressure measurements obtained using pressure taps and a *Preston* tube. A linear relation is established between the local wall shear stress (τ_{w_x}) and the pressure difference (Δp_I) measured with the *Irwin* probes. The numerical simulations implemented in the open source OpenFOAM® 2.4.0 CFD toolbox are benchmarked against ANSYS CFX results, and the two sets are compared against the experimental results. A viscous sub-layer formulation was used, with $y^+ \approx 1$ for the mesh. Although the focus of the present study is to investigate constant section ducts, some preliminary results for variable section ducts are also presented. Two representative cases—convergent with 1° slope (**C1**) and divergent with 1° slope (**D1**)—were selected. The *Preston* tube measurements are in good agreement with the numerical results and within the expected accuracy of the experimental results obtained under adverse pressure gradient conditions.

R. Faria (✉)
Department of Mechanical Engineering—Rua Pedro Nunes—Quinta da Nora, Polytechnic Institute of Coimbra, ISEC, 3030-199 Coimbra, Portugal
e-mail: rfaria@isec.pt
URL: http://www.isec.pt

R. Faria · A. D. Ferreira · A. M. G. Lopes
Department of Mechanical Engineering, ADAI-LAETA, University of Coimbra—Polo II, 3030-788 Coimbra, Portugal
e-mail: almerindo.ferreira@dem.uc.pt

A. M. G. Lopes
e-mail: antonio.gameiro@dem.uc.pt
URL: http://www.uc.pt

A. C. M. Sousa
Department of Mechanical Engineering, University of New Brunswick, 4400, Fredericton, NB E3B 5A3, Canada
e-mail: asousa@unb.ca
URL: http://www.unb.ca/fredericton/engineering/depts/mechanical/people/sousa.html

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®* ,
https://doi.org/10.1007/978-3-319-60846-4_24

1 Introduction

The increasing interest in turbulent flows in noncircular ducts of current use in many technical applications such as heat exchangers, air conditioning systems, and rotary machinery has led to a considerable research effort aimed at the understanding and characterization of these flows. The most common configurations are the triangular and rectangular cross-section shapes [5, 8, 16, 17].

In engineering applications, it is quite common to consider the fully developed flow characteristics in noncircular ducts to be similar to those flowing in pipes of circular cross section [1, 3, 5, 6, 8]. Under these conditions, the characteristic length for noncircular ducts is formulated in terms of the hydraulic (or equivalent) diameter (D_h); therefore, the Reynolds number (Re) is determined as follows:

$$Re = \frac{U D_\mathrm{h}}{\nu}, \tag{1}$$

where U is the axial mean velocity (m/s) and ν the kinematic viscosity of the fluid (m^2/s).

It is well-established that, in a fully developed flow, the pressure drops linearly in the stream-wise direction. Pressure is commonly expressed in terms of the dimensionless pressure coefficient (C_p), namely

$$C_\mathrm{p} = \frac{p - p_\mathrm{ref}}{\frac{1}{2}\rho U^2}, \tag{2}$$

where p (Pa) is the static pressure at the section of interest, p_ref (Pa) is the reference static pressure, ρ ($\mathrm{kg/m}^3$) is the density of the fluid, and U (m/s) is the axial mean velocity.

Assuming a fully developed flow in ducts of constant cross section, the pressure drop between two sections can be used to evaluate the mean friction coefficient (C_f) [5]. Moreover, the C_f value is constant along the section and can be evaluated by using, e.g., *Preston* tubes. The friction factor is defined as

$$C_\mathrm{f} = \frac{\tau_w}{\frac{1}{2}\rho U^2}. \tag{3}$$

The numerical simulations performed within this work were done in the open source OpenFOAM® 2.4.0 (2015) CFD toolbox (CFD: Computational Fluid Dynamics) (henceforth designated as OF), which is attracting increasing interest from practitioners in industrial engineering, as well as in the scientific community. Several reasons led to the choice of this code, such as no limitations for parallel computing and openness for adaptation and development, bringing great flexibility and suitability for the users, which is not the case with commercial CFD codes.

Verification of the model is performed through a mesh dependency study with several grids of increasing resolution, i.e., decreasing of y^+ values, defined as follows:

$$y^+ = \frac{u_* y}{v}, \tag{4}$$

where y (m) is the distance to the nearest wall and u_* is the friction velocity (m/s), which is related to the wall shear stress (τ_w), as follows:

$$u_* = \sqrt{\frac{\tau_w}{\rho}}. \tag{5}$$

This work is conducted so as to evaluate the applicability of *Irwin* [7] probes and *Preston* [14] tubes to the study of turbulent incompressible flow through rectangular ducts and, in particular, the determination of the wall shear stress under pressure gradient conditions. In the present work, the numerical results obtained with OF are benchmarked against ANSYS CFX results and the two sets are compared against the experimental results.

2 Experimental Setup

For the present study, a rectangular duct with a constant width of 0.12 m was built; although the experimental setup allows for four different aspect ratios (AR) [4], for the present work, only the 1:2 case ($H = 0.06$ m) was considered.

Figure 1 schematically depicts the experimental apparatus. A centrifugal fan was used to blow the air into the 5-m-long plywood rectangular duct, and between the blower and the duct entrance, a *Venturi* flow meter was placed to measure the mean flow velocity. The turbulence intensity was obtained with a hot film probe placed at the duct's exit. The duct has an initial flow-developing section, followed by the so-called test section; both sections have the same length (2.5 m). Fifteen pressure taps were initially considered in the test section, aiming for the identification of four averaging sections. At each averaging location (indicated by the dashed lines in Fig. 1), the mean pressure was obtained by using four pressure taps, one on each side of the duct.

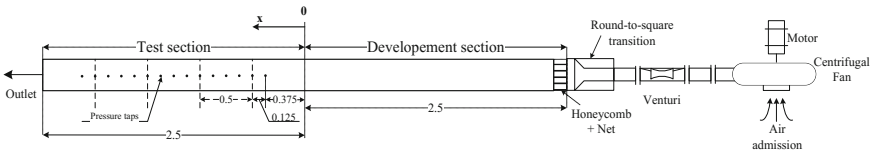


Fig. 1 Schematic of the experimental apparatus—top view (Dimensions in meters)

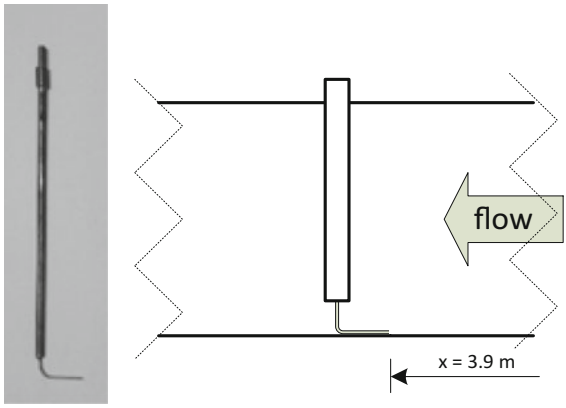


Fig. 2 Placement of the *Preston* tube

2.1 Preston Tube

The pressure difference between the *Preston* tube reading and the local static pressure (pressure tap) (Δp) is used to quantify the local wall shear stress (τ_{w_x}). The calibration of Patel [12], simplified by Bechert [2] later on, is used in the present work to compute τ_{w_x} as follows:

$$\tau^+ = \left[28.44\Delta p^{+^2} + \left(6.61 \times 10^{-6}\right)\Delta p^{+^{3.5}}\right]^{\frac{1}{4}}. \tag{6}$$

Normalization of the quantities presented in Eq. [5] leads to

$$\Delta p^+ = \Delta p d^2/\rho\nu^2 \qquad \tau^+ = \tau_{w_x} d^2/\rho\nu^2, \tag{7}$$

where d is the outside diameter of the *Preston* tube.

Figure 2 presents the *Preston* tube used to perform the measurements and the respective placement.

2.2 Irwin Probes

Several equal *Irwin*-type [7] pressure probes, projected and built as described in Faria et al. [4], were used in the present work. The dimensions of the probes are presented in Fig. 3.

The pressure difference (Δp_I) between the two pressure taps of an *Irwin* probe can be related to the local wall shear stress (τ_{w_x}) by calibration, which will be discussed in the ***Results and Discussion*** section. After calibration, the wall's shear stress can be used in conjunction with Eq. (3) to determine the friction coefficient (C_f).

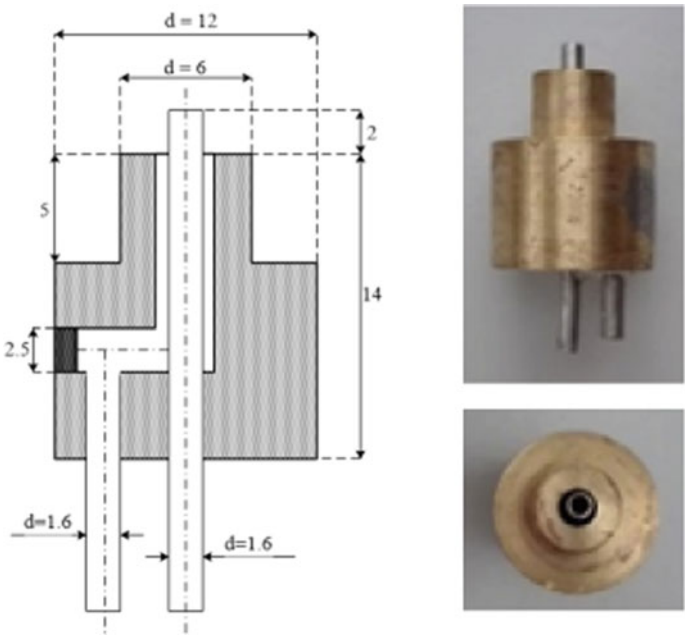


Fig. 3 Geometry of the *Irwin* probes used (dimensions in millimeters)

The *Irwin* probes are surface-mounted on the rectangular duct, with one of the taps placed level with the surface and the other one 2 mm away from the same; the probes are located along the bottom centerline on positions of the test section that satisfy the fully developed flow requirement.

3 Numerical Setup

The numerical simulations were implemented in the open source OpenFOAM® CFD toolbox [11], as already stated. Figure 4 depicts the three-dimensional computational domain employed. Due to the symmetry of the geometry under consideration, only a quarter of the physical domain was simulated.

The OF *blockMesh* utility was used to generate several structured grids with increasing resolution so as to analyze the mesh dependency. A simple grid expansion (*simpleGrading*) away from the walls was employed (as shown in Fig. 5) with expansion factors between 1 (no expansion) and 1.15 (Table 1).

The standard solver *simpleFOAM* was employed to solve the incompressible, steady-state RANS equations based on the finite-volume discretization method (FVM). A parallelized computation with six processors was employed, and numer-

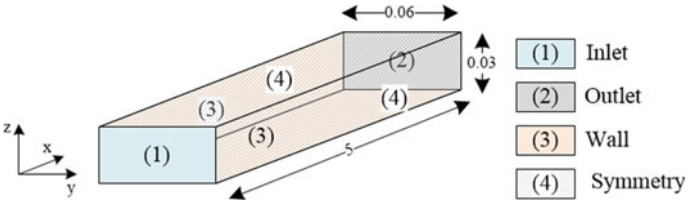


Fig. 4 Computational domain (dimensions in meters)

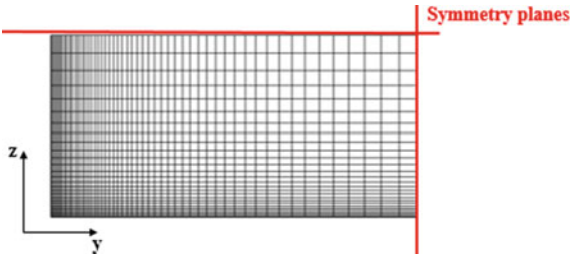


Fig. 5 Typical mesh with simpleGrading for one-quarter of the cross section

ical convergence was assumed satisfied when all the normalized residuals have a value lower than 1×10^{-5}.

The k-ω SST (Shear Stress Transport) [9] turbulence model was used with the automatic near-wall treatment [10]. Symmetry conditions were imposed on two boundaries, as shown in Fig. 4. The boundary conditions at the inlet were as follows: zero pressure gradient for pressure and fixed values for the other flow properties; the turbulence intensity took a value of 5% to match the experimental conditions. At the outlet, the zero-gradient condition was taken for all variables, except pressure, which was set as equal to the experimental value. Walls were treated as smooth, no-slip surfaces. Similar conditions and identical mesh sizes were adopted for the ANSYS CFX simulations.

4 Results and Discussion

In this section, the measurements taken in the experimental setup and their comparison with the values predicted by OF are presented. The results discussed here are for the duct aspect ratio 1:2, which is representative of all other cases tested.

In addition, preliminary results for rectangular ducts with variable cross section are also presented.

Table 1 Structured mesh characteristics

Mesh	Cells			Expansion factor			Elements	Iterations	Mean y^+
	x	y	Z	x	y	z			
M1	201	12	6	1.008	1.000	1.000	14,472	192	139.72
M2	201	20	10	1.008	1.000	1.000	40,200	308	84.37
M3	201	24	12	1.008	1.018	1.038	57,888	346	56.72
M4	201	33	16	1.008	1.035	1.076	106,128	453	28.27
M5	201	43	21	1.008	1.044	1.094	181,503	632	14.00
M6	201	70	34	1.008	1.051	1.109	478,380	1460	2.41
M7	201	82	40	1.008	1.052	1.111	659,280	1038	1.18
M8	347	75	37	1.004	1.071	1.151	962,925	1030	0.53

4.1 Experimental Results

To determine the entrance length required to obtain a fully developed flow, the static pressure was measured along the test section of the duct. Independently of the Reynolds number (*Re*), the results reported in Fig. 6 clearly indicate that the pressure decreases linearly towards the exit, with a mean r-square coefficient of 0.9969 (within the range of 0.9940–0.9985). In Fig. 6, x/L_{max} refers to the distance (x) normalized by L_{max} (L_{max} being the length of the test section, $L_{max} = 2.5$ m).

It can be observed in Fig. 6 that the higher pressure values occur for the lower Re numbers, which is in accordance with the findings of several authors, e.g., Rochlitz et al. [15].

The local friction was calculated using Eq. (6) for several flow velocities; its dependence on *Re* is depicted in Fig. 7.

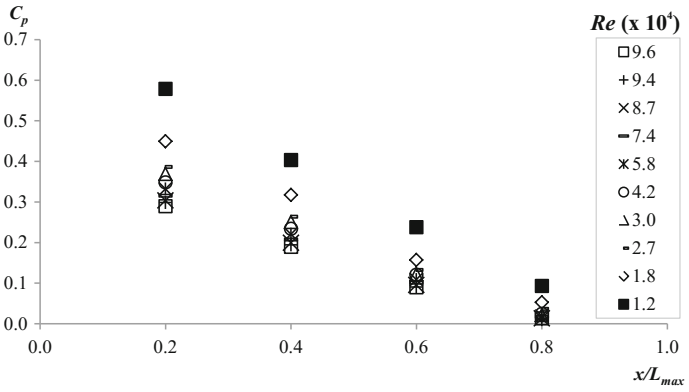


Fig. 6 Pressure coefficient along the duct for several *Reynolds* numbers

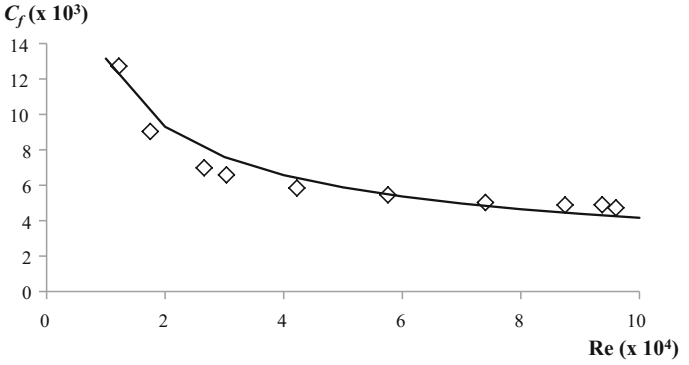


Fig. 7 Local friction coefficient for different *Reynolds* numbers at $x/L_{max} = 0.56$

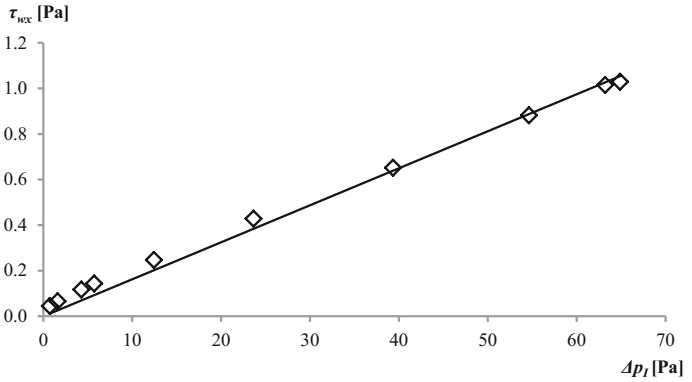


Fig. 8 Calibration curve for the *Irwin* probes located at $x/L_{\max} = 0.56$

The wall friction tends towards a constant value as *Re* increases (Fig. 7); there is clear evidence that, with increasing values of *Re,* the *Reynolds* independence condition occurs in what concerns the local friction coefficient. To support this observation, a square root fit, $C_f \alpha Re^{-0.5}$, which is represented by the solid line, was added to Fig. 7. Several authors, such as Peysson et al. [13], reported similar findings.

4.2 *Calibration of the* Irwin *Probes*

For each selected velocity, the calibration of the *Irwin* probes is established by relating the values of the local wall shear stress (τ_{w_x}) determined using Eq. (6) with the pressure difference measured with the *Irwin* probes (Δp_I), as depicted in Fig. 8. The empirical relation for the calibration of the *Irwin* probes can be found by fitting a curve through the measured data points.

For the data presented in Fig. 8, the local wall shear stress (τ_{w_x}) can be correlated to the pressure difference (Δp_I) by the following linear approximation, with an *r*-square coefficient of 0.9915:

$$\tau_{w_x} = 0.0162 \Delta p_I. \tag{8}$$

4.3 *Numerical Results*

Mesh independency tests were performed prior to validation of the OF model against the experimental results.

The first step was to verify the behavior and sensitivity of the numerical code and turbulence model to the increasing mesh resolution. Two test cases of of tutorials

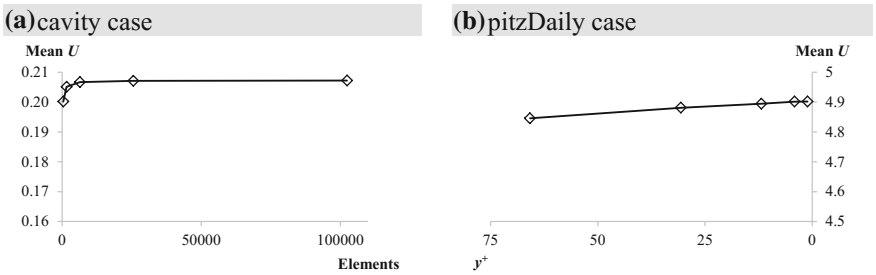


Fig. 9 Mesh independency test—cross-section mean velocity. **a** Cavity case; **b** pitzDaily case

were chosen for this purpose, one case with laminar flow (cavity case) and the other one with the turbulence model k-ω SST (pitzDaily case). It can be observed in Fig. 9 that, in both cases, beyond a certain degree of refinement, the results were not affected, yielding differences among the meshes lower than 1%; therefore, mesh independency is satisfied. Furthermore, the turbulence model k-ω SST is shown to be practically insensitive to the value of y^+, since differences smaller than 4% were obtained within the entire range, $0.23 < y^+ < 65$, in line with the findings reported by Menter and Esch [10].

However, those results were only achieved for the properties far from the wall, like the cross-section mean velocity. When considering wall-bounded quantities, like the local wall shear stress, some discrepancies among the meshes were found. Similar behavior occurred in the test performed for the meshes presented in Table 1.

As already mentioned, the k-ω SST turbulence model was used with the automatic near-wall treatment, which implies that the boundary layer is resolved in the fine grids (viscous sub-layer formulation: $y^+ < 5$) and modeled in the coarse grids (wall functions: $30 < y^+ < 100$–300). Therefore, when performing mesh independency tests, besides the spatial discretization inaccuracy, the different wall treatment approaches also contribute to the errors; this can be observed when comparing the near-wall velocity profile for the various meshes tested (Table 1), as depicted in Fig. 10.

To avoid this problem, the mesh independency study should typically be conducted within the y^+ validity range of a particular procedure. Considering that it is intended to evaluate wall-bounded flow characteristics, namely the wall pressure distribution and the local wall shear stress, a viscous sub-layer formulation approach was used, i.e., a mesh with a value of $y^+ < 5$. The value of $y^+ \approx 1$, as indicated by the CFD community, is the best one for predicting the wall-bounded quantities, especially when employing the k-ω SST turbulence model. Under the circumstances, the mesh $201 \times 82 \times 40$ (M7 in Table 1) is used in the present work to conduct the validation of the model against the experimental results and, in the process, add further credibility to these experimental results.

Furthermore, according to the selected near-wall treatment, the applied boundary conditions must be chosen in accordance. A study on this subject was conducted,

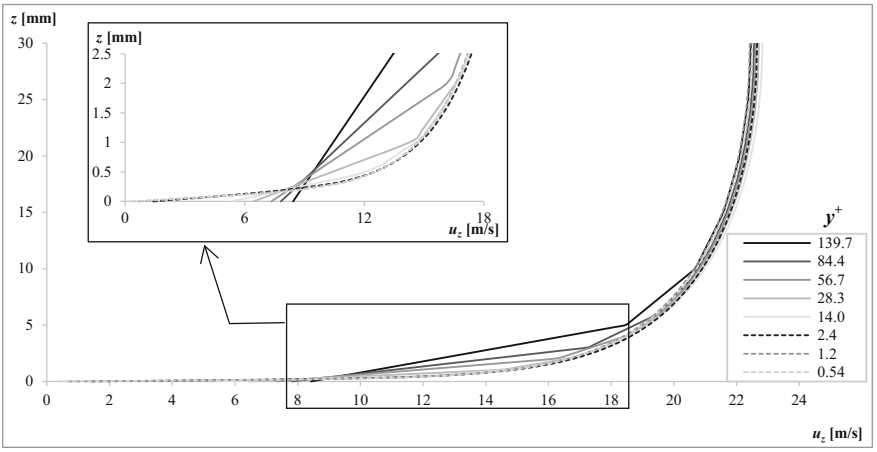


Fig. 10 Mesh independency study—near-wall velocity profile

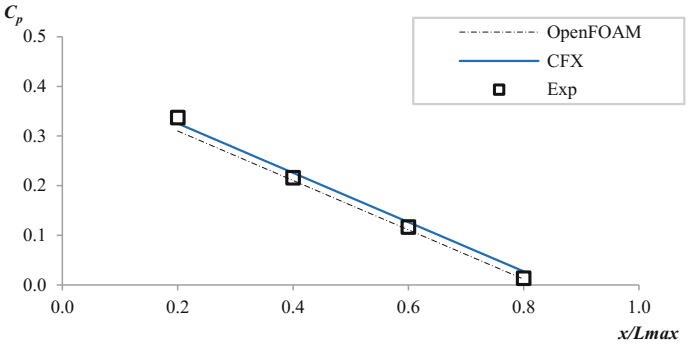


Fig. 11 Comparison of the experimental wall pressure distribution along the rectangular duct with OF and CFX predicted values

leading to the use of the boundary conditions presented in the ***Numerical Setup*** section.

The results shown in this section were obtained for the maximum velocity flow in the duct ($Re = 9.6 \times 10^4$).

Figure 11 depicts the comparison of the wall pressure distribution along the rectangular duct (only for the four averaging locations of mean pressure defined in Fig. 1) with the values predicted by OF and the CFX code.

It can be observed in Fig. 11 that good agreement was observed in the predictions of both OF and CFX, with a deviation lower than 5%. However, it should be mentioned that, although the same conditions were adopted for both the CFX and OF simulations (in particular, computational domain, boundary conditions, and y^+), the calculation time and required number of iterations showed marked differences. To reach the converged solution, the OF solution required 1038 iterations (Table 1), while CFX

needed only 83. Regarding the calculation time, OF took about six hours, while CFX required only 30 min. In spite of the fact that the mesh arrangement is different for these software packages, as well as the procedure for calculation of the residuals, the main factor for the calculation speed difference may be related to the solver algorithm: CFX uses a coupled solver, while OF adopts a segregated approach. Nevertheless, due to its great flexibility and the ease of use associated with its open source nature, OF has been generating considerable interest among CFD practitioners in the scientific community.

The local wall friction was compared against the experimental measurements of the *Preston* tube, with a maximum deviation of less than 2%.

A comparison between numerically and experimentally derived pressure and friction coefficients was made in order to validate the numerical model. The differences obtained were less than 5%; therefore, it can be concluded that the OF predictions match well with the experimental measurements.

5 Velocity Influence

As already mentioned, the previous numerical results were obtained for the maximum flow velocity (U_{max}). However, the velocity influence (Fig. 12) was also analyzed and two other velocities were simulated, namely 60 and 30% of the maximum velocity. The M7 mesh was again used in the numerical simulations to determine the wall pressure distribution and the local wall shear stress reported in Fig. 12 and Table 2, respectively.

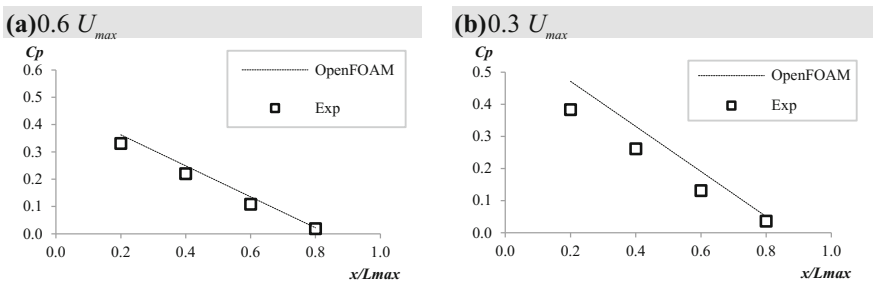


Fig. 12 Wall pressure distribution. **a** Velocity of 0.6 U_{max}, **b** velocity of 0.3 U_{max}

Table 2 Local friction coefficient ($x/L_{max} = 0.56$) predicted by OF and deviations from *Preston* measurements for two different velocities (0.6 U_{max} and 0.3 U_{max})

Case	*Preston* measurements	OF	Deviation (%)
0.6 U_{max}	21.51	21.95	10.97
0.3 U_{max}	27.17	27.67	10.80

In regard to that which concerns the wall pressure distribution, the results obtained with the velocity of 0.6 U_{max} are in better agreement with the experimental data than those for velocity of 0.3 U_{max}, as presented in Fig. 12. This can happen because the measured pressure values in the velocity of 0.3 U_{max} are very low and close to the limit of the sensitivity range of the measuring equipment. Table 2 presents the local friction coefficient, and in both cases, deviations between the numerical and experimental results are below 2%.

5.1 *Preliminary Results of the Rectangular Duct with Variable Section*

A few changes in the experimental setup (Fig. 1) allowed for the study of the rectangular duct with variable section, which is obtained by the variation of the angle of the top wall of the duct. Both convergent and divergent longitudinal sections were tested, as can be seen in Fig. 13.

The results presented in this section were obtained for two representative cases, namely convergent with 1° slope (**C1**) and divergent with 1° slope (**D1**).

The measurements of the pressure distribution along the test section of the duct and its comparison against the predicted values by OF are depicted in Fig. 14.

The location of the hinge is at $x/L_{max} = 0.375$ and it is indicated in both graphs of Fig. 14 by the dotted lines.

The experimental and numerical results are in good agreement for the convergent case (**C1**), with a deviation around 2%. The deviation for the divergent case is around

Fig. 13 Schematic of the new experimental apparatus—Side View (dimensions in meters)

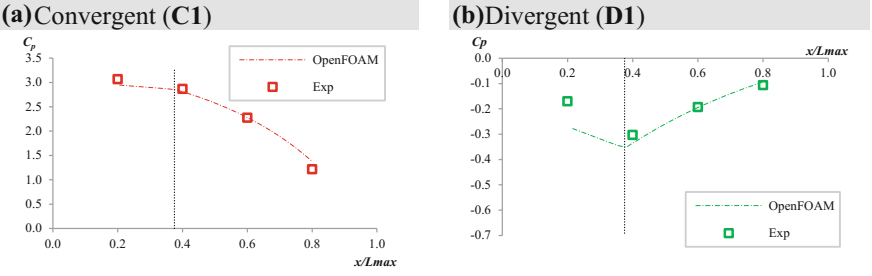


Fig. 14 Pressure coefficient results. **a** Convergent geometry, **b** divergent geometry

Table 3 Local friction coefficient ($x/L_{max} = 0.56$) predicted by OF and deviations from *Preston* measurements for three representative cases: constant section (**H**), convergent with 1° slope (**C**1) and divergent with 1° slope (**D**1)

Case	*Preston* measurements	OF	Deviation (%)
H	18.68	19.04	1.93
C1	31.61	26.53	16.07
D1	15.45	14.72	4.73

20%; however, if only the divergent longitudinal section is considered, the deviation for this case (**D1**) drops to 8%. It should be noted that the mesh adopted for these simulations is identical to that of the constant section tests.

In regard to that which concerns the friction coefficient, the comparison of the *Preston* tube measurements with the numerical results is reported in Table 3.

For **C1** geometry, the deviation is approximately 16%, while for the **D1** geometry, the mean deviation is below 5%.

Some caution is required in these comparisons, considering that the use of *Preston* tubes under adverse pressure gradient conditions has severe limitations. Patel [12] reports that, under both favorable and adverse pressure gradients, the *Preston* tube tends to overestimate the skin friction, as observed in Table 3. Using the interval of the parameter Δ (severity of the pressure gradient) proposed by Patel [12], the prescribed error range leads to a decrease in the indicated deviations, to 10 and 2% for the convergent and divergent cases, respectively.

6 Conclusions

The applicability of *Irwin* probes and *Preston* tubes to the measurement of the local wall shear stress in incompressible flow through rectangular ducts of constant and variable longitudinal section is the scope of the present work.

The tests for the rectangular duct experimental setup were conducted for a cross section with aspect ratio 1:2 and *Reynolds* number values ranging from 10^4 to 9.6×10^4; measurements were obtained for the fully developed flow condition for all velocities tested.

The occurrence of *Reynolds* independence conditions is observed for the local friction coefficient, which is derived using data obtained with a *Preston* tube and local static pressure.

The *Irwin* probes' calibration was obtained through the fitting of a power function to the experimental data. The local wall shear stress (τ_{w_x}) is related to the pressure difference in the *Irwin* probes (Δp_I) by a linear approximation, with an r-square coefficient of 0.9915.

In the present work, simulations performed using OF and ANSYS CFX are compared against experimental results. A range of y^+ from 0.5 to around 140 was used to perform the mesh dependency study in OF.

A viscous sub-layer formulation was the approach selected with $y^+ \approx 1$ for the mesh based on CFD best practice, which indicates that, with the k-ω SST turbulence model, it is the appropriate value for predicting the wall-bounded quantities.

A comparison between numerically and experimentally derived pressure and friction coefficients was conducted to validate the numerical model and, in the process add credibility to the experimental data; deviations below 5% were obtained.

Therefore, considering the good agreement with the numerical results, it was concluded that *Irwin* probes and *Preston* tubes are applicable to the measurement of the local wall shear stress in turbulent incompressible flows through smooth rectangular ducts.

The influence of the velocity was also measured, and two other velocities were considered ($0.6U_{\max}$ and $0.3U_{\max}$). The difference between predictions and experimental data for the wall pressure distribution increases for the lower velocity. However, for both cases, the difference between predicted and experimental local wall shear stress is lower than 2%.

Some preliminary results were also reported for the rectangular duct with variable cross section. For both configurations (**C1** and **D1**), good agreement between experimental and numerical results was obtained for the pressure coefficient, namely 2% for the convergent geometry and 8% if only the divergent section is considered for the **D1** geometry. In regard to that which concerns the local friction coefficient, comparison between predicted and experimental results indicates better agreement for the **D1** geometry, with a mean deviation of 5% against 16% for the **C1** geometry. However, this comparison should be made with considerable caution, taking into consideration the limitations of a Preston tube under pressure gradient conditions. By introducing the prescribed error range proposed by Patel [12], the deviations will drop to 10 and 2% for the convergent and divergent cases, respectively.

Acknowledgements The authors acknowledge all those involved in the organization of OFW11 and all the contributors who made this event a very stimulating one. The authors are also indebted for the experimental facilities made available by ADAI (Association for the Development of Industrial Aerodynamics).

References

1. Ahmed S, Brundrett E (1971) Turbulent flow in non-circular ducts. Part 1. Int J Heat Mass Transf 14:365–375. https://doi.org/10.1016/0017-9310(71)90156-6.
2. Bechert DW (1996) Calibration of Preston tubes. AIAA J 34:205–206. https://doi.org/10.2514/3.13048.
3. Duan Z, Yovanovich MM, Muzychka YS (2012) Pressure Drop for Fully Developed Turbulent Flow in Circular and Noncircular Ducts. ASME J Fluids Eng 134:61201. https://doi.org/10.1115/1.4006861.

4. Faria R, Ferreira AD, Lopes AMG, Sousa ACM (2016) On the use of Irwin probes to measure wall shear stress in rectangular ducts of constant section. CYTEF2016 - Proc - Adv Refrig Sci Technol - VIII 9.
5. Hartnett JP, Koh JCY, McComas ST (1962) A Comparison of Predicted and Measured Friction Factors for Turbulent Flow Through Rectangular Ducts. J Heat Transfer 84:82. https://doi.org/10.1115/1.3684299.
6. He S, Gotts JA (2005) Calculation of Friction Coefficients for Noncircular Channels. ASME J Fluids Eng 126:1033–1038. https://doi.org/10.1115/1.1845479.
7. Irwin HPAH (1981) A simple omnidirectional sensor for wind-tunnel studies of pedestrian-level winds. J Wind Eng Ind Aerodyn 7:219–239. https://doi.org/10.1016/0167-6105(81)90051-9.
8. Melling A, Whitelaw JH (1976) Turbulent flow in a rectangular duct. J Fluid Mech 78:289. https://doi.org/10.1017/s0022112076002450.
9. Menter FR (1994) Two-equation eddy-viscosity turbulence models for engineering applications. AIAA J 32:1598–1605. https://doi.org/10.2514/3.12149.
10. Menter FR, Esch T (2001) Elements of Industrial Heat Transfer Predictions. 16th Brazilian Congr Mech Eng 117–127. https://doi.org/10.1017/cbo9781107415324.004.
11. OpenFOAM® (2015) The Open Source CFD Toolbox. User Guide Version 3.0.1.
12. Patel VC (1965) Calibration of the Preston tube and limitations on its use in pressure gradients. J Fluid Mech 23:185. https://doi.org/10.1017/s0022112065001301.
13. Peysson Y, Ouriemi M, Medale M, Aussillous P, Guazzelli É (2009) Threshold for sediment erosion in pipe flow. Int J Multiph Flow 35:597–600. https://doi.org/10.1016/j.ijmultiphaseflow.2009.02.007.
14. Preston JH (1954) The Determination of Turbulent Skin Friction by Means of Pitot Tubes. J R Aeronaut Soc 58:109–121. https://doi.org/10.1017/s0368393100097704.
15. Rochlitz H, Scholz P, Fuchs T (2015) The flow field in a high aspect ratio cooling duct with and without one heated wall. Exp Fluids 56:1–13. https://doi.org/10.1007/s00348-015-2071-y.
16. Rokni M, Olsson C, Sundén B (1998) Numerical and experimental investigation of turbulent flow in a rectangular duct. Int J Numer Methods Fluids 28:225–242. https://doi.org/10.1002/(sici)1097-0363(19980815)28:2%3c225::aid-fld710%3e3.3.co;2-k.
17. Yao J, Zhao Y, Fairweather M (2015) Numerical simulation of turbulent flow through a straight square duct. Appl Therm Eng 91:800–811. https://doi.org/10.1016/j.applthermaleng.2015.08.065.

Numerical Approach for Possible Identification of the Noisiest Zones on the Surface of a Centrifugal Fan Blade

Tenon Charly Kone, Yann Marchesse and Raymond Panneton

Abstract This paper examines the capability of both the Proper Orthogonal Decomposition (POD) and the Singular Value Decomposition (SVD) to identify the zones on the surface blades of a centrifugal fan that contribute the most to the sound power radiated by moving blades. The Computational Fluid Dynamics (CFD) OpenFOAM® source code is used as a first step to evaluate the pressure field at the surface of the blade moving in a subsonic regime. The fluctuating component of this pressure field makes it possible to directly estimate both the loading noise and the sound power that is radiated by the blade based on an acoustic analogy of Ffowcs Williams and Hawkings (FW&H). In the second step, the estimated loading noise is then employed to evaluate the radiated sound power using the POD and SVD approaches. It may be noted that the sound power reconstructed by the two latter approaches, when relying solely on the most important acoustic modes, is similar to the one predicted by the FW&H analogy. It is also noted that the contribution of the modes in the radiated sound power does not necessarily appear in ascending order in the decomposition (i.e., in descending order of energy). Moreover, the highest radiating SVD modes are mapped onto the blade surface so as to highlight the zones that contribute the most to the noise. It is then expected that this identification will be used as a guide in the design of the blade surface to reduce the radiated noise.

T. C. Kone (✉) · R. Panneton
GAUS Department of Mechanical Engineering, Université de Sherbrooke,
2500 bvd. Université, Sherbrooke, QC J1K2R1, Canada
e-mail: Tenon.Charly.Kone@USherbrooke.ca

R. Panneton
e-mail: Raymond.Panneton@USherbrooke.ca

Y. Marchesse
Université de Lyon, ECAM Lyon, 40 montée Saint-Barthélemy,
69321 Lyon Cedex 05, France
e-mail: yann.marchesse@ecam.fr

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_25

1 Introduction

When considering centrifugal fans, the acoustic noise due to the interaction between the turbulent flow and the moving surfaces cannot be ignored, since the radiated noise level is generally a quality criterion. Furthermore, reducing production costs, while maintaining good performance and low noise, is a major challenge for the engineers. Unfortunately, there is no general solution to this compromise, and each fan configuration needs an in-depth design analysis. Therefore, experimental or numerical analyses are usually employed to reduce the sound radiated by fans and compressors [1, 2]. In this context, it is difficult to identify and characterize the zones on the blade surface that contribute the most to the radiated sound power. The present investigation aims to study the ability of one particular method to overcome this difficulty. The latter is based on the combination of the acoustic analogy of Ffowcs Williams and Hawkings (FW&H) [3] and the use of the Proper Orthogonal Decomposition (POD) [4] or the Singular Value Decomposition (SVD) [5].

The POD method is a particularly efficient data analysis method for studying complex physical systems. It was introduced in fluid mechanics by Lumley in 1967 to identify and extract coherent turbulent flow structures [4]. Its use in acoustics began in 1974, with Arnt and George [6], with a view toward linking the radiated noise to the fluctuating velocity modes in the Lighthill tensor [6]. Since then, only a few studies have been conducted [7–11], and most of them dealt with flow noise based on fluctuating velocity decomposition (e.g., jet noise). Few of these studies addressed the noise generated by the interaction between turbulent flow and solid walls. As for SVD, it is generally used to search for propagation operators, such as the Green function [5, 12], or to solve inverse problems in acoustics [13–16]. To the author's knowledge, SVD has not yet been used to locate radiating areas on a moving surface due to its interaction with the flow.

In light of the previous works, this research applies the POD and SVD methods to the problem of aeroacoustic noise generated by the interaction of a moving blade and turbulent flow in a centrifugal fan. The objective is to establish a link between the decomposition modes (of POD or SVD) and the noisiest areas of the blade. The methodology consists of three steps: (i) Modeling, using foam-extend-3.2 [19–21] software, the internal flows of the centrifugal fan by the Large Eddy Simulation (LES) [17, 18] method. The objective is to estimate the wall pressure fluctuation over the blade. (ii) Using the previous wall pressure fluctuation over the blade to estimate the loading noise from an FW&H analogy [3, 22] adapted to a moving source. (iii) Finally, using the POD and SVD to extract the most important acoustic modes and visualize them, so as to identify the highest radiating zones on the blade surface.

The paper is organized as follows. First, the estimation of the acoustic field based on the FW&H analogy is introduced. Both the POD and SVD methods are then described. Finally, a simulation of a flow passing through centrifugal blades is made as an application of this approach.

2 Theory

2.1 Geometry of the Problem

The problem under consideration is schematically depicted in Fig. 1. It consists of a centrifugal fan composed with a rotor holding 22 identical blades, an inlet stator, and an outlet stator. When the fan rotates, the interaction between the flow and the blades generates noise. Since the geometry of the fan is periodic, a periodic model with only one blade will be studied in the forthcoming developments and calculations. Moreover, it will be assumed that the radiated sound power is calculated only for the blade placed in a free field (i.e., without any wall in the vicinity). This will simplify the analysis since acoustic reflections with the stator's walls will not be taken into account in the calculated sound power.

2.2 Estimation of the Acoustic Field (FW&H Analogy)

As discussed previously, the configuration studied here is a rotating blade radiating noise in free space. The tip tangential velocity is such that the flow is subsonic. The thickness of the blades is considered null, and only the loading noise (p_L) generated by the pressure fluctuation over the blade surface is therefore investigated. To alleviate the computation time problem, formulation 1A proposed by Farassat [22] can be used. In this approach, the receiver time derivative in formulation 1 [22] is transformed into a retarded time derivative. This also has the great advantage of permuting the time derivative and the space integration. For the case in which both the source and the receiver are not moving, and the propagation medium is at rest, the Farassat formulation 1A [22] in a far field (i.e., when $r \gg \lambda \gg D_{ext}$, where λ is the wavelength and D_{ext} the characteristic size of the fan) becomes

$$p_L(\mathbf{x}, t) \simeq \frac{1}{4\pi} \int_{f(\mathbf{y},\tau)=0} \left[\frac{\dot{\ell}_r}{cr(1-M_r)^2} + \frac{\ell_r \dot{M}_r}{r(1-M_r)^3} \right]_\tau dS, \tag{1}$$

where S is the source surface (i.e., blade surface), $r = \|\mathbf{r}\| = \|\mathbf{y} - \mathbf{x}\|$ is the distance between the source position $\mathbf{y}$ on the blade surface and the receiver position $\mathbf{x}$, c is the speed of sound of the acoustic medium at rest, $\ell_r = -p\mathbf{n}.\mathbf{r}/r$, where $\mathbf{n}$ is the unit normal vector to the blade surface, p is the pressure fluctuation over the blade surface obtained by CFD calculation, $\mathbf{M}$ is the Mach vector number with $M_r = \mathbf{M}.\mathbf{r}/r$, and $[\bullet]_\tau$ indicates that all the integrands should be evaluated at the retarded time $\tau = t - r/c$, where t is the reception time.

As mentioned previously, formulation 1, proposed by Farassat, has the main advantage of avoiding the spatial derivatives. However, the receiver time derivative is maintained on ℓ_r and M_r. The implementation of this operation is complex and the computation time increases. To evaluate Eq. (1), two computational approaches are

available in the literature [23, 24], using either retarded time or advanced time. The latter is usually chosen when the aerodynamic data comes from CFD computations, as in this study. The received acoustic pressure must then be determined, while an irregular receiving time discretization appears, despite the regular emission time. An interpolation is thus necessary to obtain the received sound pressure at a regular time step.

The advanced time approach and the Lagrange interpolation are used in this paper. The calculation of the sound pressure at receiver $\mathbf{x}$ in the far field and in a free medium allows for the calculation of root-mean-square sound pressure, p_L. When considering the loading noise (Eq. 1), this reads as

$$p_L^2(\mathbf{x}) = \left\langle p_L(\mathbf{x}, t)\, p_L(\mathbf{x}, t) \right\rangle_{T_0}, \tag{2}$$

where $\langle\ \rangle_{T_0}$ is the temporal average over the time period T_0. Thus, the radiated sound power estimated from far-field microphones (i.e., receivers) located on a spherical surface encompassing the source writes

$$\mathscr{P} = \int_{S_\mathbf{x}} \frac{p_L^2(\mathbf{x})}{\rho c} dS_\mathbf{x} \simeq \frac{1}{\rho c} \sum_{\mathbf{x}} p_L^2(\mathbf{x}) \Delta S_\mathbf{x}, \tag{3}$$

where $S_\mathbf{x}$ is the receiving surface, $\Delta S_\mathbf{x}$ is the elementary surface associated with receiver $\mathbf{x}$, and ρ is the density of the surrounding fluid medium.

2.3 Proper Orthogonal Decomposition

Generally, POD used in aeroacoustics is not based on acoustic analogies. However, in this work, the developed approach is a combination of the POD theory and the FW&H acoustic analogy limited to the dipole term (i.e., loading noise).

If one considers a dipole located at $\mathbf{y}_i$, the sound pressure received at point $\mathbf{x}$ at time t according to Eq. (2) can be written as

$$p_L(\mathbf{x}, \mathbf{y}_i, t) = \frac{1}{4\pi} \left[\frac{\dot{\ell}_r}{cr(1 - M_r)^2} + \frac{\ell_r \dot{M}_r}{r(1 - M_r)^3} \right]_\tau \Delta S_{\mathbf{y}_i}, \tag{4}$$

where $\Delta S_{\mathbf{y}_i}$ is the ith elementary surface of the source located at $\mathbf{y}_i$. Here, this elementary surface comes from the discretization of the blade surface for the LES calculation.

Since the sound pressure received at one point corresponds to the contribution of all the dipoles located on the blade surface, two matrices $\mathbf{A}$ and $\mathbf{W}_{obs}$ are defined by

$$\mathbf{W}_{obs} = \frac{1}{N} \mathbf{A}\mathbf{A}^T, \tag{5}$$

where

$$\mathbf{A} = \begin{pmatrix} p_L(\mathbf{x}, \mathbf{y}_0, t_{e_0}) & p_L(\mathbf{x}, \mathbf{y}_0, t_{e_1}) & \cdots & p_L(\mathbf{x}, \mathbf{y}_0, t_{e_{N-1}}) \\ p_L(\mathbf{x}, \mathbf{y}_1, t_{e_0}) & p_L(\mathbf{x}, \mathbf{y}_1, t_{e_1}) & \cdots & p_L(\mathbf{x}, \mathbf{y}_1, t_{e_{N-1}}) \\ \vdots & \vdots & \ddots & \vdots \\ p_L(\mathbf{x}, \mathbf{y}_{m-1}, t_{e_0}) & p_L(\mathbf{x}, \mathbf{y}_{m-1}, t_{e_1}) & \cdots & p_L(\mathbf{x}, \mathbf{y}_{m-1}, t_{e_{N-1}}) \end{pmatrix}$$

and where t_{e_j} represents the emission time for the dipole source located at $\mathbf{y}_i (i = 0, 1, \ldots, m-1)$ at time step j $(j = 0, 1, \ldots, N-1)$. Each column vector of the matrix $\mathbf{A}$ is the sound contribution of all dipoles at a given reception time, while each row vector represents the sound pressure of a single dipole source during the receiving time.

Matrix $\mathbf{W}_{obs}$ in Eq. (5) is the correlation matrix of the sources for receiver $\mathbf{x}$. It is symmetric, real, positive definite, and spatial. Its eigenvalues (modes) are thus real, positive, and space-dependent. For developing a modal basis for matrix $\mathbf{W}_{obs}$, let $\boldsymbol{\lambda} = diag\,(\lambda_0, \lambda_1, \ldots, \lambda_{m-1})$ and $\boldsymbol{\phi} = \left[\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{m-1}\right]$ be, respectively, the diagonal matrix of eigenvalues and the matrix of eigenvectors at receiver $\mathbf{x}$. Each column $\boldsymbol{\phi}_i$ is the eigenvector associated with the eigenvalue λ_i at receiver $\mathbf{x}$. Then, for every receiver $\mathbf{x}$, the problem to be solved is the following eigenvalue problem:

$$\mathbf{W}_{obs}\boldsymbol{\phi} = \boldsymbol{\lambda}\boldsymbol{\phi}. \tag{6}$$

Multiplying each member of Eq. (6) from the right by the transpose of the matrix of eigenvectors, if normalized modes are considered so that $\boldsymbol{\phi}\boldsymbol{\phi}^T = \mathbf{I}$, the expression for the correlation matrix is given by

$$\mathbf{W}_{obs} = \sum_{k=0}^{m-1} (\lambda_k \boldsymbol{\phi}_k)\boldsymbol{\phi}_k^T. \tag{7}$$

The correlation matrix $\mathbf{W}_{obs}$ is then written as a sum of independent matrices defined as spatial autocorrelation patterns with proper modes as components. Since the eigenvectors form an orthonormal basis of the source space, the sound pressure produced by the ith elementary surface of the source located at $\mathbf{y}_i$ can be written as

$$p_L(\mathbf{x}, \mathbf{y}_i, t) = \sum_{k=0}^{m-1} \alpha_k(t)\Phi_{k,i}, \tag{8}$$

where $\alpha_k(t) = \alpha_k(\mathbf{x}, t) = \sum_{i=0}^{m-1} p_L(\mathbf{x}, \mathbf{y}_i, t)\Phi_{k,i}$ are the temporal modal amplitudes or the projection coefficients on the modal basis. $\Phi_{k,i} = \phi_k(\mathbf{x}, y_i)$ is the ith component of the kth eigenvector $(\boldsymbol{\phi}_k)$. Thus, coefficients are the root-mean-square of the acoustic pressure projected on the $\Phi_k(\mathbf{y})$ axis in the source space (i.e., the blade). According to the theory of Merces [25], the projection coefficients form an orthogonal basis

for the temporal space and its root-mean-square $\langle \alpha_k(t)\alpha_k(t)\rangle_{T_0}$ corresponds to the eigenvalue λ_k. The eigenvalues represent the square of the sound pressure projected onto the $\Phi_k(\mathbf{y})$ axis in the source space. The total sound pressure from the loading noise radiated to the receiver $\mathbf{x}$ at time t is the contribution of individual sources (Eq. 8), and therefore Eq. (1) reads as

$$p_L(\mathbf{x}, t) = \sum_{k=0}^{m-1} \left(\alpha_k(t) \sum_{i=0}^{m-1} \Phi_{k,i} \right). \tag{9}$$

Substituting the loading pressure p_L of Eq. (2) with Eq. (9), and considering the orthogonality of the eigenvectors, the mean quadratic sound pressure at receiver $\mathbf{x}$ yields

$$p_L^2(\mathbf{x}) = \sum_{k=0}^{m-1} \left(\sum_{i=0}^{m-1} \Phi_{k,i} \right)^2 \lambda_k. \tag{10}$$

In this proper orthogonal decomposition, not every mode contributes equally to the total quadratic pressure. The latter can then be evaluated only by taking into account the modes that contribute the most. This is reported by the accumulated acoustic energy of the first q modes, E_{Ac_q}, divided by the total acoustic energy of all modes

$$E_{Ac_q} = \frac{\sum_{k=0}^{q-1} \lambda_k}{\sum_{k=0}^{m-1} \lambda_k}. \tag{11}$$

Once the highest contributing modes are identified, the summations in Eqs. (3) and (10) are limited up to $q-1$ instead of $m-1$. Thus, Eqs. (9) and (10) becomes, respectively,

$$p_L(\mathbf{x}, t) \simeq \sum_{k=0}^{q-1} \left(\alpha_k(t) \sum_{i=0}^{m-1} \Phi_{k,i} \right), \tag{12}$$

$$p_L^2(\mathbf{x}) \simeq \sum_{k=0}^{q-1} \left(\sum_{i=0}^{m-1} \Phi_{k,i} \right)^2 \lambda_k. \tag{13}$$

In the previous equations, the spatial eigenvectors $\Phi_{k,i}$ give information on the acoustic radiation of all dipole sources distributed over the surface S. Thus, considering Eq. (13), the acoustic power defined by Eq. (3) becomes

$$\mathscr{P} \simeq \frac{1}{\rho c} \sum_{\ell=0}^{N_{obs}-1} \left(\Delta S_\ell \sum_{k=0}^{q-1} \lambda_{\ell k} \left(\sum_{i=0}^{m-1} \Phi_{\ell k,i} \right)^2 \right), \tag{14}$$

where $\Delta S_\ell = \Delta S_{\mathbf{x}}$ is the elementary surface of the ℓth receiver, N_{obs} is the number of receivers, $\lambda_{\ell k}$ is the kth eigenvector of the ℓth receiver, and $\Phi_{\ell k,i}$ is the ith component of the eigenvector associated with the kth eigenvalue $\lambda_{\ell k}$.

2.4 Singular Value Decomposition (SVD)

The SVD makes the sound generation investigation (i.e., the eigenvalues and the eigenvectors resulting from the POD) independent of the receiver $\mathbf{x}$. A global $(N_{obs} \times m) \times m$ matrix $\mathbf{W}_{SVD}$ gathering all the correlation matrices $\mathbf{W}_k$ $(0 \leq k < N_{obs})$ of the N_{obs} receivers is built

$$\mathbf{W}_{SVD} = \begin{bmatrix} \mathbf{W}_0 \\ \mathbf{W}_1 \\ \vdots \\ \mathbf{W}_{N_{obs}-1} \end{bmatrix}. \tag{15}$$

Here, the receivers are distributed over a sphere of radius R around the source. The radius is large enough that the far-field assumption is verified. The position of the receivers over the sphere is done according to ISO 3745 standard [26].

According to the SVD method, the $\mathbf{W}_{SVD}$ matrix is decomposed in the following form:

$$\mathbf{W}_{SVD} = \mathbf{U}\boldsymbol{\sigma}\mathbf{V}^T, \tag{16}$$

where $\boldsymbol{\sigma}$ is the diagonal matrix of singular eigenvalues and $\mathbf{U}$ and $\mathbf{V}$ are the matrices of the left and right eigenvectors, respectively. $\mathbf{U}$ accounts for the difference in the acoustic radiation of the sources between receivers and $\mathbf{V}$ provides the average information of the acoustic radiation of the sources for all receivers. One can show that $\mathbf{U}$ and $\mathbf{V}$ can be obtained by application of the POD on matrices $\mathbf{W}_{SVD}^T\mathbf{W}_{SVD}$ and $\mathbf{W}_{SVD}\mathbf{W}_{SVD}^T$, respectively. By retaining only the first q energetic modes, the expression of the sound power (Eq. 3) based on the SVD can finally be expressed as

$$\mathscr{P} \simeq \frac{1}{\rho c} \sum_{k=0}^{q-1} \left(\sigma_k \sum_{i=0}^{m-1} V_{k,i} \sum_{\ell=0}^{N_{obs}-1} \left(\Delta S_\ell \sum_{j=m*\ell}^{m(\ell+1)-1} U_{k,j} \right) \right), \tag{17}$$

where σ_k is the kth eigenvalue of the matrix $\mathbf{W}_{SVD}$, and $V_{k,i}$ and $U_{k,i}$, respectively, represent the ith component of the right and left eigenvectors associated with to the eigenvalue σ_k from the POD method.

3 Application

3.1 Geometry, Spatial Discretization, and Boundary Conditions

As introduced earlier, the geometry under consideration consists of a centrifugal fan made of two stators and one rotor. More specifically, the rotor holds 22 identical blades with a constant $\pi/11$ angular distance placed around the rotor axis (Fig. 1a). The blades' profile is derived from a NACA airfoil profile.

The numerical model to be solved by CFD is simplified due to the periodicity of the fan's geometry. As a consequence, only 1/11th of the geometry is studied as shown in Fig. 1c. The rotor domain considers one blade and is delimited on either side by the wall of the next blade. The centered blade is the principal acoustic source of the problem; the other sources are supposed to be negligible.

The inner and outer radii of the rotor equal $D_{int} = 0.105$ m and $D_{ext} = 1.66 D_{int}$, respectively. The thickness of the profile is 0.0017 m, the length of its chord is 0.056 m, and the skeleton shape follows a logarithmic function. The height of the rotor at the inlet equals $b_0 = 0.04$ m, while the heights of the leading edge and trailing edge are $b_1 = 0.034$ m and $b_2 = 0.0177$ m, respectively.

A hybrid mesh of the numerical domain is built using the free software Salome. A structured mesh is used for both stators and near the central blade (acoustic source). An unstructured mesh is used elsewhere. The first layer of cells of the structured mesh at the wall of the central blade is chosen so that the dimensionless wall parameter y^+ nearly equals 1. The first grid points are thus located in the laminar part of the boundary layer. The influence of the size of the cells, located in the rotor domain, on the solution is investigated in the direction of the flow (Δx) and the transverse direction (Δz). For this, three meshes are built: coarse, medium, and fine meshes, namely cases 0, 1, and 2. All information about these meshes is shown in Table 1.

For the boundary conditions, the velocity at the inlet stator is fixed at 8.127 m/s, and a zero pressure is imposed at the exit of the outlet stator (Fig. 1c). The no-slip

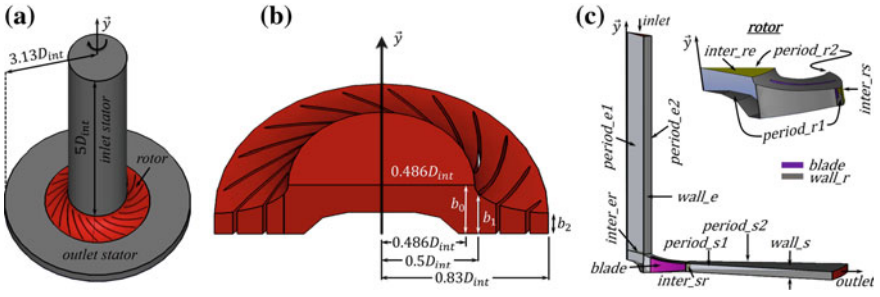


Fig. 1 Geometry of the centrifugal fan problem under consideration. The **y**-axis is the axis of rotation. **a** Full geometry. **b** Rotor. **c** 1/11th periodic geometry and boundary conditions

Table 1 Discretization parameters of each mesh of the blade and the rest of the rotor

	Discretization parameters			Number of cells	
	Blade (source)		Rotor		
Case	Δx (m)	Δz (m)	Max size (m)	Blade (source)	Total
0	0.0010	0.00025	0.00123	22,813	462,910
1	0.0010	0.00025	0.00064	22,813	1,118,233
2	0.0007	0.00002	0.00064	44,972	1,381,229

condition is imposed on both the rotor and stator wall surfaces. Periodic boundary conditions are imposed at all periodic coupled boundary faces, i.e., (period_e1, period_e2), (period_r1, period_r2) and (period_s1, period_s2), using cyclicGgi boundary conditions with nCopies = 11 following foam-extend-3.2 notation. Furthermore, interface conditions are defined on coupled surfaces (inter_re, inter_er) and (inter_rs, inter_sr) based on overlapGgi boundary conditions [19, 21] with a periodicity angle of $\pm\pi/11$, so that the air flow passes through the three domains. The angular velocity of the rotor equals $\Omega = 2800$ rpm.

3.2 Governing Equations and Time Discretization

The fluid is pure air and its physical properties are estimated at 25 °C ($\rho = 1.2$ kg/m^3 and $\mu = 1.831 \times 10^{-5}$ Pa s). The Large Eddy Simulation method is used to simulate the internal flow channel with the free software foam-extend-3.2 [19]. The one−equation sub-grid model [18] for turbulent kinetic energy is used. For the pressure field, the linear system is solved with the iterative preconditioned conjugate gradient method with Diagonal-Based Incomplete Cholesky (DIC) preconditioning. For the velocity field and turbulent kinetic energy, the stabilized biconjugate gradient method with preconditioning Diagonal Incomplete-LU (DILU) is chosen. The backward second-order scheme is used for the temporal resolution. The time step value depends on the mesh that is used in the simulation: $\Delta t_{CFD} = 6.62 \times 10^{-7}$ s for case 0; 5×10^{-7} s for case 1; and 3.5×10^{-7} s for case 2 (Table 1). These values lead to a CFL number lower than 0.2 in the three simulations. The Gauss linear scheme is used to calculate the terms of the spatial derivatives. Two months were usually necessary to reach convergence using 48 processors on the Compute Canada-Sherbrooke (supercomputer) [27]. 3.5 revolutions of the blade are needed for the solution to be converged here. Once this convergence is reached, the wall pressure fluctuations (i.e., p) over the blade surface (source) are saved periodically, so that $\Delta t_a = 8\Delta t_{CFD}$ during a full rotation of the rotor. The number of time samples equals $N = 4046$, 5357, and 7653, respectively, for cases 0, 1, and 2. For each case, the pressure fluctuations

p are filtered by a band-pass filter to reject frequencies outside the 50 Hz–10 kHz range. The sound that will be estimated from the FW&H acoustic analogy will then remain within the audible range.

3.3 POD Analysis and Interpretation

As was mentioned earlier the pressure fluctuations over the blade surface are estimated using the CFD approach. An acoustic analogy then leads to a filtered acoustic pressure (i.e., filtered pressure signal) ultimately employed for both the POD and SVD methods. Thus, the radiated sound pressure, the correlation matrix, and the modes are calculated for the receivers placed on a sphere around the source, per ISO 3745 standard [26], as discussed in Sect. 2.4. For this investigation, $N_{obs} = 20$ receivers are placed on a sphere of radius $R = 6$ m. Figure 2 represents the projection of the receivers in the plane passing through the origin and normal to the rotor axis. In order to investigate the influence of the number of modes in the estimation of the loading noise, Eq. (9) is first used with all the m POD modes to evaluate p_L. Second Eq. (12) is used to evaluate p_L with only the first 10 dominant modes. The comparison is shown in Fig. 3 for four different receivers. It is noticed, as expected, that both the amplitude and the time at which maxima occur depend on the receiver location, whatever the value of q. Furthermore, using $q = 10$ seems sufficient to reconstruct the signal obtained when $q = $ m (Fig. 3), since the relative error between both calculations remains less than 12.5%. Moreover, what is not shown here is the fact that the mesh refinement does not significantly influence the loading noise evaluation. From these results, calculations show that 60% of the acoustic energy is reconstructed with the first 10 POD modes. When the radiated sound power is evaluated from Eqs. (3) or (14), the comparison is even better, since the relative error drops below 1.1% for the three cases.

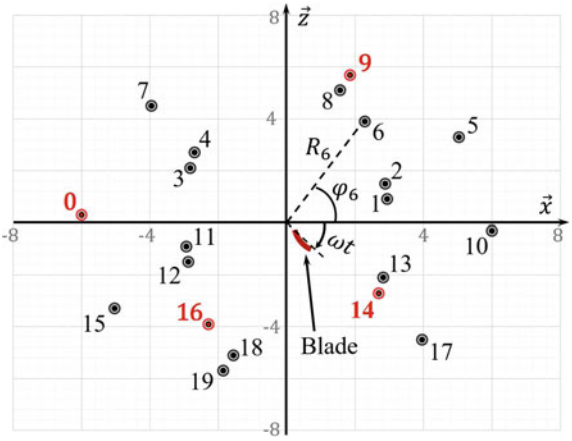


Fig. 2 Position of the receivers in the plane passing through the origin and normal to the rotor axis **y** with $\omega = 2\pi\,\Omega/60$, as per ISO 3745 standard [26]

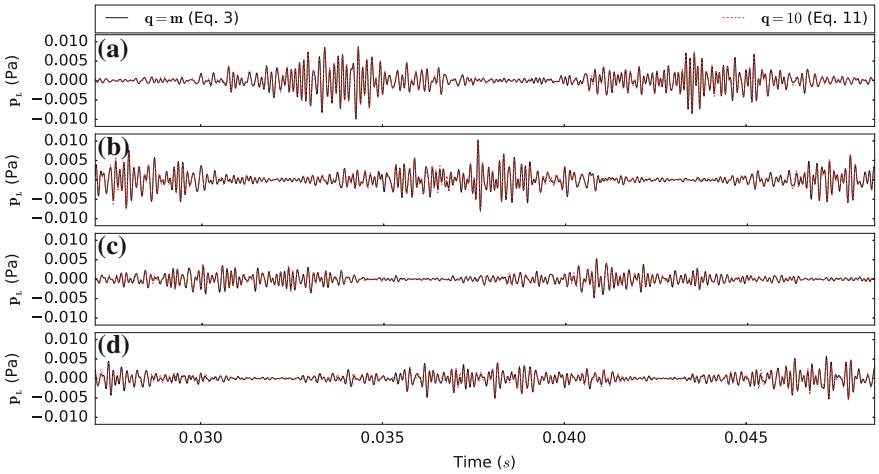


Fig. 3 Load noise pressure with all POD modes (Eq. 9) and with the first 10 dominant modes (Eq. 12) at receivers **a**: 0, **b**: 9, **c**: 14, and **d**: 16

To conclude on POD, it has been shown that both the radiated sound pressure and the sound power using only the first few POD modes may be satisfactorily reconstructed. As a consequence, instead of using all the sources $\mathbf{x}$ on the blade surface (i.e., all m POD modes), only the first 10 dominant POD modes were sufficient (for information, here, $m = 22813$ for cases 0 and 1, and $m = 44972$ for case 2).

3.4 SVD Analysis and Interpretation

Figure 4 shows the contribution of each SVD mode in the evaluation of the acoustic power from Eq. (17). It appears that SVD modes 0 and 1 for cases 0 and 1, and 0 and 2 for case 2, make a significant contribution to the acoustic power. While the mode number associated with the second most significant mode differs for case 2 in comparison with the two others, the shapes of the modes are similar for all of them, as will be shown later in the mapping of the modes on the blade surface. In contrast with the results obtained from POD, the major part of the acoustic energy is now captured by only two modes. As a consequence, when estimating the radiated sound power using Eqs. (3) and (17), based on all SVD modes and only the above 2 modes, respectively, relative error equals 0.62% for cases 0 and 2, and 1.1% for case 1.

From the singular value decomposition of the aeroacoustic problem under consideration, it is possible to map the right eigenvectors $\mathbf{V}_k$ on the blade surface, as shown in Fig. 5. Here, this is done in order to visualize the zones that are mostly responsible for the radiated sound pressure independently of the receiver location. Depending on the mode number and the side of the blade, the highest values appear

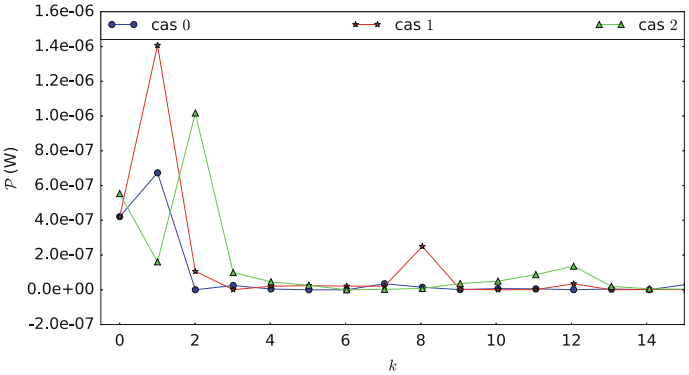


Fig. 4 Radiated sound power of each SVD mode (Eq. 17) for different mesh cases (0, 1, and 2)

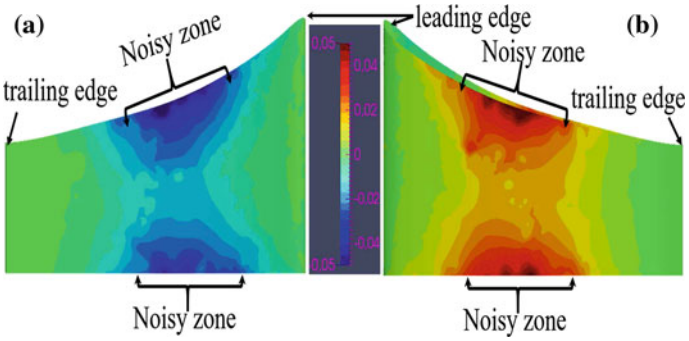


Fig. 5 Mapping of the highest radiating SVD eigenvector for each side of the blade: **a** suction side, eigenvector 1 for cases 0 and 1, and eigenvector 2 for case 2; **b** pressure side, eigenvector 0 for the three cases

mainly at 50% of the chord and near the edge (see Fig. 5). These regions are located near the borders, where the fluctuations of the velocity and pressure in the boundary layer are strong. As a consequence, a proper modification of the blade geometry, or surface treatment, in these regions would reduce the radiated sound power. This conclusion has yet to be validated, but maybe in the future work.

The same mapping and identification of the highest radiating region could also be done using the POD approach. However, since POD is done per receiver, the mapping would only indicate the potentially noisiest zone for a given receiver. It would not lead to a global reduction of the radiated sound power. However, if the reduction is to be obtained at a particular location, the POD approach would be most preferable.

3.5 Conclusion

Two decomposition methods based on CFD calculations and acoustic analogies were applied to identify the zones on centrifugal fan blades that contribute the most to the sound power radiated in the subsonic regime. Here, the CFD calculations were performed using foam-extend-3.2 to obtain the pressure fluctuations on the blade surface. These pressure signals were filtered to retain frequency components between 50 Hz and 10 kHz. Based on these filtered pressure signals, the loading noise was evaluated based on the FW&H acoustic analogy and decomposed using POD and SVD. From POD and SVD, the radiated sound power and pressure can be estimated using only the first few modes. For the geometry studied, 10 POD modes or 2 SVD modes were sufficient to recover the radiated sound pressure. The highest contributing eigenvectors found by POD or SVD can be mapped onto the surface of the blade to visualize the zone that contributes the most to sound radiation. Since the eigenvectors are dependent on the receiver location, the POD approach is preferred when dealing with unidirectional acoustic treatments (e.g., the acoustics of pipes). As for the SVD approach, it is more general and enables acoustic treatment in all directions in space. It is mostly appropriate when dealing with the reduction of radiated sound power of a source. For the centrifugal fan studied, the mapping of the most radiating SVD eigenvectors showed that the noisiest area of the blade is located at 50% of the chord near the edge. This is an interesting insight in regard to modifying the blade properly for possible reduction of sound power.

This iteration has yet to be validated (although it may be in future work) to prove the efficiency of the redesign method. This study also demonstrated the usefulness of foam-extend in assisting in the realization of complex aeroacoustic projects. The next step is to implement the FW&H acoustic analogy in foam-extend.

Acknowledgements This work was supported by the Natural Sciences and Engineering Research Council of Canada (N.S.E.R.C.). The authors wish to thank Compute Canada-Sherbrooke for their help.

References

1. Embleton, T. F.: Experimental study of noise reduction in centrifugal blowers. The Journal of the Acoustical Society of America **35**, 700–705 (1963)
2. Guedel, A.: Bruit des ventilateurs–Partie 2 (Fan noise–Part 2) Technical engineering BM4178 23p (2002)
3. Ffowcs Williams, J. E., Hawkings, D. L.: Sound generation by turbulence and surfaces in arbitrary motion. Proceedings of the Royal Society of London, **264**, 21–342 (1969)
4. Lumley, J. L.: The structure of inhomogeneous turbulent flows in atmospheric turbulence and radio wave propagation. In proceeding international colloque. Publisher House NAUK, Moscow, june 15–22, 166–177 (1967)

5. Borgiotti, G. V.: The power radiated by a vibrating body in an acoustic field and its determination from boundary measurements. Journal of the Acoustical Society of America. **88**, 1884–1893 (1990)
6. Arndt, R. E. A., George, W. K.: Investigation of the large scale coherent structure in a jet and its relevance to jet noise (technical report). Pennsylvania State Univ.; University Park, PA, United States, 20 p. (1974)
7. Arndt, R. E. A., Long, D. F., Glauser, M. N.: The proper orthogonal decomposition of pressure fluctuations surrounding a turbulent jet. Journal of Fluid Mechanics, **234**, 1–33 (1997)
8. Druault, P., Yu, M., Sagaut, P.:Quadratic stochastic estimation of far-field acoustique pressure with coherent structure event in a 2d compressible plane mixing layer. International Journal for Numerical Method in Fluids, **62**, 906–926 (2010)
9. Druault, P., Hekmati, A., Ricot, D.: Discrimination of acoustic and turbulent components from aeroacoustic wall pressure field. Journal of Sound and Vibration, **332**, 7257–7278 (2013)
10. Glegg, S. A. L., Devenport, W. J.: Proper orthogonal decomposition of turbulent flows for aeroacoustic and hydroacoustic applications. Journal of Sound and Vibration. **239**, 767–784 (2001)
11. Hekmati, A., Ricot, D.: Aeroacoustic analysis of th automotive ventilation outlets using extended proper orthogonal decomposition. In Proceedings of the 15th AIAA/CEAS Aeroacoustics Conference. 1–11 (2009)
12. Photiadis, D. M.: The relationship of singular value decomposition to wave–vector filtering in sound radiation problems. Journal of the Acoustical Society of America. **88**, 1152–1159 (1990)
13. Grace, S. P., Atassi, H. M., Blake, W. K.: Inverse aeroacoustic problem for a streamlined body, part 1. basic formulation. American Institute of Aeronautics and Astronautics Journal. **34**, 2233–2240 (1996)
14. Grace, S. P., Atassi, H. M., Blake, W. K.: Inverse aeroacoustic problem for a streamlined body, part 2. accuracy of solutions. American Institute of Aeronautics and Astronautics Journal. **34**, 2241–2246 (1996)
15. Nelson, P. A., Yoon, S. H.: Estimation of acoustic source strength by inverse methods : Part i. conditioning of the inverse problem. Journal of Sound and Vibration. **233**, 643–668 (2000)
16. Nelson, P. A., Yoon, S. H.: Estimation of acoustic source strength by inverse methods : Part ii. experimental investigation of methods for choosing regularisation parameters. Journal of Sound and Vibration. **233**, 669–705 (2000)
17. Da Silva, C. B., Pereira, J. C. F.: Analysis of the gradient diffusion hypothesis in large eddy simulations based on transport equations. Physics of Fluids **19** (2007) https://doi.org/10.1063/1.2710284
18. Eugene, d. V.: The Potential of large eddy simulation for the modeling of wall bounded flows. Phd, Imperial College of Science, Technology and Medicine, Londre-UK (2006) http://powerlab.fsb.hr/ped/kturbo/OpenFOAM/docs/EugeneDeVilliersPhD2006.pdf
19. OpenFoam®: The open source CFD toolbox https://openfoamwiki.net/index.php/News/Release_of_foam--extend_3.2
20. OpenFoam®: User Guide https://www.openfoam.com/documentation/user--guide.
21. Martin Beaudoin, Hrvoje Jasak: Development of a Generalized Grid Interface for Turbomachinery simulations with OpenFOAM®. Open Source CFD International Conference, Berlin, Germany (2008)
22. Farassat, F.: Derivation of Formulations 1 and 1A of Farassat (Rapport technique). NASA Langley Technical Report Server (2007) http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20070010579.pdf
23. Casalino, D.: An advanced time approach for acoustic analogy predictions. Journal of Sound and Vibration, **261**, 583–612 (2003)
24. Fedela, D., Smaine, K., Farid, B., Robert, R.: Modelling of broadband noise radiated by an airfoil-application to an axial fan. International Journal of Vehicle Noise and Vibration **3**, p. 106–117 (2007) https://doi.org/10.1504/IJVNV.2007.014400
25. Mercer, J.: Functions of positive and negative type and their connection with the theory of integral equations. Philosophical Transactions of the Royal Society A, volume **209**, 415–446 (1909) https://doi.org/10.1098/rsta.1909.0016

26. ISO 3745: Acoustics – Determination of sound power levels and sound energy levels of noise sources using sound pressure – Precision methods for anechoic rooms and hemi-anechoic rooms (2012) https://www.iso.org/obp/ui/#iso:std:iso:3745:ed-3:v1:en
27. Compute–Canada: https://www.computecanada.ca

Numerical Modeling of Flame Acceleration and Transition from Deflagration to Detonation Using OpenFOAM®

Reza Khodadadi Azadboni, Jennifer X. Wen and Ali Heidari

Abstract The present numerical investigation aims to study the dynamics of deflagration-to-detonation transition (DDT) in inhomogeneous and homogeneous mixtures. Modeling discontinuities, such as shocks and contact surfaces, in high-speed compressible flows require numerical schemes that can capture these features while avoiding spurious oscillations. For the numerical model, two different solution approaches, i.e., the pressure-based and density-based methods, have been adopted using the OpenFOAM® CFD toolbox. A reactive density-based solver using the Harten–Lax–van Leer-contact (HLLC) scheme has been developed within the frame of OpenFOAM®. The predictions are in reasonably good qualitative and quantitative agreement with the experiments (Boeck et al. in The GraVent DDT Database, 2015 [3]). The DDT phenomena have two major stages; flame acceleration (FA), during which the flow is in the subsonic regime, and the transition-to-detonation stage, in which the combustion wave undergoes a transition to the supersonic state. The present study indicates that it is viable to use the pressure-based algorithm for studying FA, but a density-based method is required for modeling DDT.

R. Khodadadi Azadboni · A. Heidari
Fire, Explosion and Fluid Dynamics Research Team, School of Mechanical & Automotive Engineering, Kingston University London, SW15 3DW Kingston, UK
e-mail: R.khodadadiazadboni@kingston.ac.uk

A. Heidari
e-mail: A.Heidari@kingston.ac.uk

J. X. Wen (✉)
Warwick FIRE, School of Engineering, University of Warwick, CV4 7AL Coventry, UK
e-mail: Jennifer.wen@warwick.ac.uk

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®* ,
https://doi.org/10.1007/978-3-319-60846-4_26

1 Introduction

Fire and explosion in combustible mixtures have been widely studied, both experimentally and numerically. Gas explosions inside tubes have been investigated for a long time. Most of these studies were carried out for industrial safety and with a desire to describe general mechanisms of flame propagation. Therefore, most of these works concern understanding the flame acceleration (FA) phenomena and transition from deflagration to detonation (DDT) in tubes [1].

One of the main hazards in hydrogen energy applications is the formation of flammable vapor clouds, which can propagate kilometers away from the source, with the possibility of igniting and resulting in fire and explosions [1]. Explosions in homogenous reactive mixtures have been widely studied, both experimentally and numerically. However, in practice, combustible mixtures are usually inhomogeneous and subject to both vertical and horizontal concentration gradients.

Thomas [2] has given a comprehensive overview of various forms of DDT and differentiates the terminology between the macroscopic DDT and the microscopic DDT. The large-scale macroscopic DDT includes the process from accelerating deflagration-to-detonation propagation. The small-scale microscopic DDT governs the actual onset of detonation at the point where the combustion process changes from diffusion controlled to shock heating controlled [2]. In this work, the term "DDT" is used in the larger definition and includes both acceleration and the onset of detonation. Thomas also features a discussion on understanding of the weak DDT, in which it is not onset by a strong reflected shock wave, but rather non-isotropic and nonequilibrium turbulence accelerates a deflagration and creates small hot spots, which, in turn, generate transverse waves and add up to strong pressure waves capable of forming the required shock/reaction complex known as the detonation [2], the importance of which is emphasized. Gas explosions inside tubes have been studied for a relatively long time.

The combustible mixtures pose a risk, especially when an ignition source is available or when the pressure and/or temperature exceed the self-ignition limits [1]. In the past, very few studies considered the effect of mixture inhomogeneity on the behavior of DDT.

Recently, Boeck et al. [3] investigated flame acceleration and DDT in a channel with vertically variable hydrogen concentrations. They showed that the flame accelerated faster when the mixture has concentration gradients. DDT was also observed as reflected shock waves interacting with the deflagration front.

The present study aims to provide an appropriate numerical method for modeling DDT phenomena in horizontal obstructed channels with two different blockage ratios, i.e., 60 and 30%. The tube is filled with a hydrogen/air mixture with an average of 30 percent hydrogen by volume.

Two different solvers developed in OpenFOAM® have been used. A pressure-based solver using the flame-wrinkling combustion model [4] has been developed and named RMXiFoam. The solver is used for uniform hydrogen/air mixture DDT modeling, as well as a prediction of baroclinic torque and Richtmyer–Meshkov insta-

bilities. According to Eq. (1), baroclinic torque is generated as a result of strong misalignment of the density and pressure gradients. In high-velocity reacting flows, such as deflagration and detonation waves, hydrodynamic instabilities are one of the key factors in enhancing turbulence through shock–flame interaction. Therefore, baroclinic vorticities have been predicted using this solver to examine the Richtmyer–Meshkov (RM) instability.

$$\text{Baroclinic torque} = (\nabla\rho \times \nabla p)/\rho^2. \tag{1}$$

For mixtures with concentration gradients, the FA, a stage that occurs at relatively low Mach numbers, has been modeled using a pressure-based algorithm, and the transition stage, which is supersonic and includes strong shock waves, has been modeled with a density-based solver. To evaluate the contribution of the convective fluxes, and accurate shock capturing, the Harten–Lax–van Leer-contact (HLLC) [5] scheme is used. The compressible Navier–Stokes equations with a single step Arrhenius reaction are solved. For turbulence modeling, the large eddy simulation (LES) technique has been used. The solver and numerical schemes were initially tested by solving the Sod's shock tube problem [6]. For numerical validation, an experimental inhomogeneous DDT test case [3] has been selected.

2 Governing Equations

The standard governing equations for solving the flow field in a Eulerian framework can be listed as below:

Mass conservation:

$$\frac{\partial\rho}{\partial t} + \nabla \cdot (\rho U) = 0. \tag{2}$$

Conservation of momentum (neglecting body forces):

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot [u(\rho u)] + \nabla p + \nabla \cdot \tau = 0. \tag{3}$$

Conservation of total energy:

$$\frac{\partial(\rho E)}{\partial t} + \nabla \cdot [u(\rho E)] + \nabla \cdot [up] + \nabla \cdot (\tau \cdot u) + \nabla \cdot j = 0, \tag{4}$$

where, in Eq. (2), ρ is the density, U the velocity vector, and p the pressure, and in Eq. (3), τ is the viscous stress tensor, which can be defined using Eq. (5) [7].

$$\tau = -2\mu\,\text{dev}(D), \tag{5}$$

where, in Eq. (5), μ is the dynamic viscosity and D is the deformation gradient tensor, which is defined as Eq. (6),

$$D \equiv \frac{1}{2}\left[\nabla u + (\nabla u)^{\tau}\right]. \tag{6}$$

In Eq. (4), j is the diffusive flux of heat and E is the total energy density, which can be defined as

$$E = e + |u|^2/2, \tag{7}$$

where e is the specific internal energy.

2.1 Solution Algorithms

Simulating discontinuities, such as shocks and contact surfaces, in high-speed compressible flows require numerical schemes that can capture these features while avoiding spurious oscillations. There are two main major solution algorithms for solving the flow fields: the pressure-based approach and the density-based approach. Both approaches are examined in the present study.

In the pressure-based approach of OpenFOAM® solvers (such as sonicFoam [8]), a non-iterative method for handling the coupling of implicitly discretized time-dependent fluid flow equations is utilized. The method is known as PISO (for pressure implicit with the splitting of operators). It is based on applying the pressure and velocity as dependent variables and applies to both the incompressible and compressible forms of the transport equations. The main feature of the technique is the splitting of the solution process into a series of steps in which operations on pressure are decoupled from those on velocity. With the split, sets of equations amenable to solution by standard techniques are intended to be produced [9].

The fields obtained after each PISO step are closer predictions of the analytical solution of the equations with a proper order of accuracy, depending on the number of operation-splittings used. The errors' rapid decay, together with the fact that the stability of the overall scheme is little impaired by the splitting procedure, should allow for getting rid of iterations while retaining the advantage of implicit differencing, namely, the ability to cope with large time steps [9]. In this work, the PISO methodology is outlined.

In some methods that are effective in producing accurate non-oscillatory solutions, for capturing shock and discontinuities, the generation of numerical fluxes typically involves Riemann solvers, characteristic decomposition, and Jacobian evaluation, making them complex and difficult to implement on a mesh of polyhedral cells that have an arbitrary number of faces [7]. However, an alternative approach exists that does not involve Riemann solvers and can also provide accurate non-oscillatory solutions using the so-called central schemes. The central schemes that are developed

Table 1 Initial condition of Sod's problem

Compartment	$X > 0.5$ Left (driven)	$X < 0.5$ Right (driven)
Pressure	$P_L = 1$	$P_R = 0.1$
Density	$\rho_L = 1$	$\rho_R = 0.125$
Velocity	$U_L = 0$	$U_R = 0$

by Kurganov and Tadmor [10], Kurganov et al. [11] and Greenshields et al. [7] are available in OpenFOAM®. Khodadadi et al. [12] carried out a shock-capturing study using different OpenFOAM® solvers to solve Sod's problem [6] and concluded that the available central scheme density-based solver in OpenFOAM® can provide the most accurate shock capturing. They also mentioned that the proposed density-based solution could generate some oscillations on the shock contact surface. These oscillations may be linked to the numerical schemes. Hence, to provide the most appropriate numerical scheme, for shock capturing, a similar shock tube modeling using Sod's condition has been conducted in this study.

The shock tube problem that was used by Sod [6] to test a number of methods for solving the equations of compressible flow has become a standard test problem. The initial conditions for this problem consist of two semi-infinite states separated by a diaphragm at time $t = 0$ [12]. The left and right states are set to the following conditions (Table 1). For the numerical modeling of Sod's problem, 100 cells have been considered for comparison with the analytical solution.

The existing central scheme density-based solver in OpenFOAM® is called "rhoCentralFoam," and the pressure-based solver adopted is called sonicFoam in the OpenFOAM® package [8, 10].

Sod's problem was modeled using the rhoCentralFoam solver by applying three different interpolation schemes as follows:

1. Using Upwind schemes for density, temperature, and velocity.
2. Using the VanLeer TVD method for density and temperature, and Upwind for the velocity.
3. Using VanLeer for all of the parameters.

For the solution of the pressure-based solver, sonicFoam has also been provided for better comparison between the numerical methods and analytical solution.

The results in Fig. 1 show a shock wave moving to the right and a rarefaction wave (expansion fan) moving to the left; the contact surface discontinuity separating the shock and rarefaction waves is moving to the right [6]. These results demonstrate that density-based solutions can provide better shock capturing than the pressure-based solver (dashed blue line) when compared with the analytical solution. Moreover, it is seen that using the Upwind method in the solution of the density-based solver can provide an adequate estimation of the shock front, but it will not produce an accurate shock contact surface and expansion fan modeling. In order to improve discontinuity modeling, TVD-type schemes [13] are used. Figure 1 also shows that the VanLeer

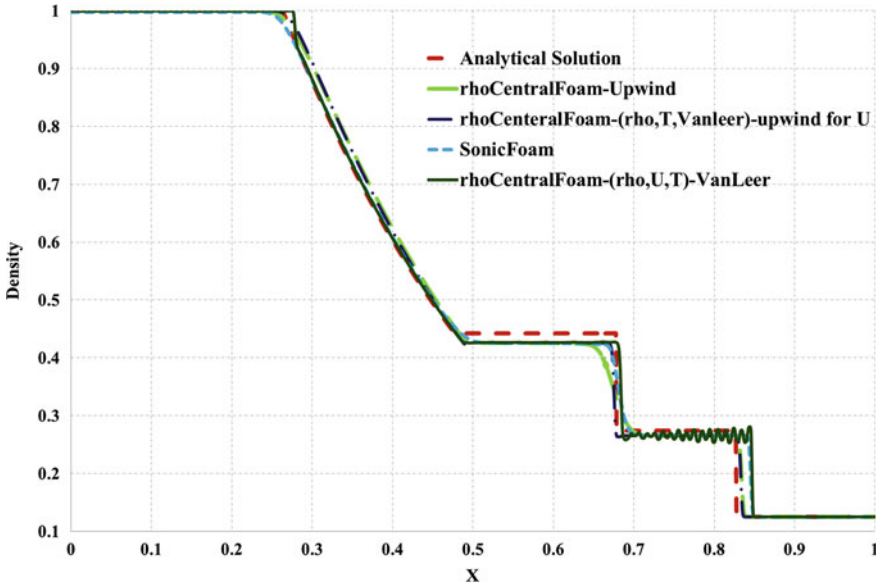


Fig. 1 Density distribution in the shock tube using 100 cells

scheme [13] can produce an accurate shock contact surface modeling, but there are some oscillations on the leading shock contact surface.

Additionally, Fig. 1 shows that by changing the interpolation scheme of the velocity from VanLeer to Upwind, and using VanLeer for interpolation of density and temperature in the rhoCentralFoam solver, the shock front, contact surface, and expansion fan predictions will closely match the analytical solution without having any oscillation on the shock contact surface.

Borm et al. [14] implemented a Godunov-type scheme to the density-based solver in OpenFOAM®, which is called dbnsTurbFoam [14]. This method uses the Harten–Lax–van Leer-contact (HLLC) [5] scheme, and for the time discrete schemes, which include the dual time scheme and the physical time step, it uses the Runge–Kutta scheme [14]. This is used as the basis for the presently assembled density-based solver, VCEFoam (vapor cloud explosion Foam) [15]. In addition, a reaction, as well as an adaptive refinement mesh (AMR), has been implemented. VCEFoam can simulate high Mach number reactive flows, by solving chemical reactions and HLLC schemes.

In order to include the chemical reaction and species transport terms from OpenFOAM®'s combustion library, the energy equation needs to be changed from using the total energy to using the sensible enthalpy, as in Eq. (8) as given below:

$$\frac{\partial(\rho h_s)}{\partial t} + \nabla \cdot (\rho U h_s) - \frac{Dp}{Dt} = \nabla \cdot \left[\alpha \nabla h_s + \sum_{i=1}^{n} h_i J_i \right] + \nabla \cdot (\tau \cdot U) + S_h, \quad (8)$$

where U, p, h_s, T, S_h are the density, velocity, pressure, sensible enthalpy, temperature and enthalpy source, respectively, α is the ratio between k and the thermal conductivity, and c_p the specific heat at constant pressure. The viscous stress tensor is defined in Eq. (5).

Species transport and the diffusion coefficient are added, so that the species conservation equation is

$$\frac{\partial(\rho Y_i)}{\partial t} + \nabla \cdot (\rho u Y_i) = \nabla \cdot J_i + R_i, \tag{9}$$

where Y_i is the mass fraction and J_i is the diffusion flux of species i, defined as Eq. (10) given below:

$$J_i = -\rho D_{i,m} \nabla Y_i, \tag{10}$$

where the binary diffusion coefficient, $D_{i,m}$, for species i in the mixture can be derived according to Wilke's equation [16].

2.2 *Transition from Low Mach Number to High Mach Number Flows*

The transition mechanism in this work is due to the interaction of reflected shock from the obstacles to the flame front. Therefore, the shock-capturing capability of the model has a substantial effect on the accuracy of the predictions. Moreover, the DDT phenomena can be divided into two main stages; first, flame acceleration (deflagration) with the subsonic flow, and then the transition-to-detonation supersonic flow. Since the density-based solver is not suitable for modeling low Mach number problems, it is better to use a pressure-based approach for the flame acceleration stage, and then subsequently switch to the density-based solution for the supersonic flow.

3 Case Study

The experiments of Boeck et al. [3] involving inhomogeneous and homogeneous DDT in a hydrogen–air mixture are simulated. The experiments were conducted in a horizontal obstructed channel with 30 and 60% blockage ratios. It was initially filled with an inhomogeneous hydrogen–air mixture, which was, on average, 30% hydrogen by volume (Fig. 2).

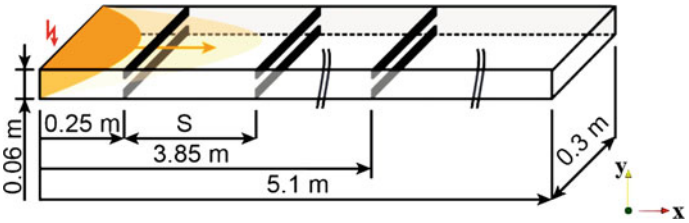


Fig. 2 Schematic of the computational domain (Reproduced from Boeck et al. [3])

4 Results and Discussion

The DDT experiments [3] are modeled using both the assembled pressure-based and density-based solvers. The pressure-based RMXiFoam solver (which is based on the XiFoam solver) uses the flame-wrinkling model [4] for combustion. Several numerical test cases revealed that although the present model can reproduce reasonable results for low speed and to some extent, fairly high-speed deflagration waves, the models suffer from considerable deviations when applied to highly turbulent and fast deflagrations that are about to undergo a transition to detonation. This is mainly because many of the underlying assumptions are not valid in near-DDT combustion regimes. For example, due to flame thickening, the standard flamelet assumption is not valid near a wall when a highly turbulent deflagration wave is interacting with an obstacle. This could significantly alter the prediction of hot spot formation when the deflagration waves are hitting an obstacle [17]. Therefore, it is decided from this point onwards, in the developed density-based solver (VCEFoam) that the efforts toward simulating DDT will be carried out without using the traditional flamelet-based combustion model, but rather through use of Arrhenius-type reactions and full instantaneous Navier–Stokes equations involving monotone-integrated large eddy simulation, a.k.a. the MILES approach [18]. The solver and numerical schemes are initially tested by solving Sod's shock tube problem [6].

4.1 Predictions Using the Pressure-Based Solver

The pressure-based RMXiFoam solver provides a reasonably good prediction of flame acceleration in the obstructed channel, but the accelerated flame did not undergo a transition to detonation.

Figure 3 shows the density counter of the flame while it propagates through obstacles 6 and 7. In the first frame, it can be seen that a weak shock wave has been generated in the flame front, subsequently interacting with the obstacle and leading to a reflected shock moving upstream of the flow. At 8.83 ms, the reflected shocks interact in the middle of the tube, and shock-focusing phenomena occur, generating a stronger shock wave. Then, at 8.9 ms, the reflected shock interacts with the flame

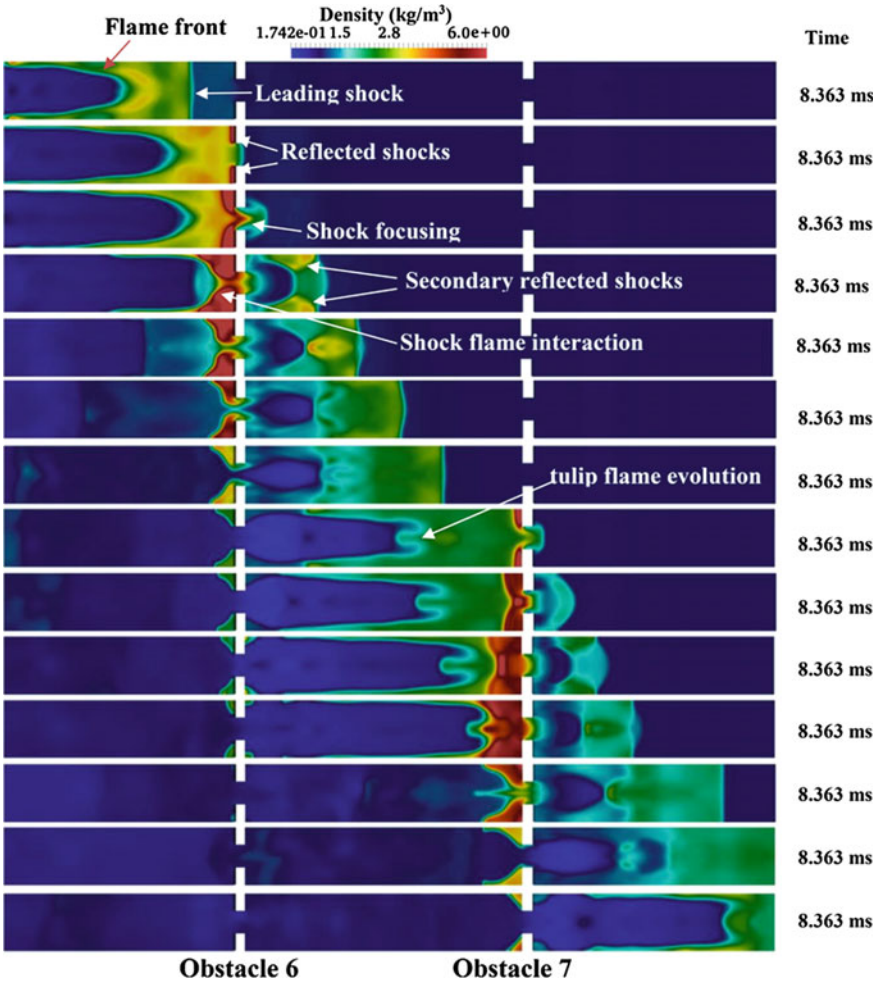


Fig. 3 Density distribution contours of deflagration of a homogenous hydrogen/air mixture of 30% Vol and BR60, with a pressure-based solution

front and the first local explosion appears, increasing the pressure up to 20 bar. However, within a period of 9 ms, the detonation fails and the leading shock wave and flame front are decoupled. The predicted baroclinic torque in the tube is not strong enough to trigger Richtmyer–Meshkov instability.

Figure 4 shows the baroclinic torques during the flame acceleration stage. It contains different directions of the baroclinic vorticities (red: out of the plane, and blue: into the plane). Xiao et al. [19] presented the baroclinic torque field at different times during the initiation of the tulip flame and predicted the magnitude of baroclinic torque to be within the range of ($-2e + 8$ to $+2e + 8(1/s^2)$). The present predictions are consistent with their findings, as shown in Fig. 4a. They also predicted that the

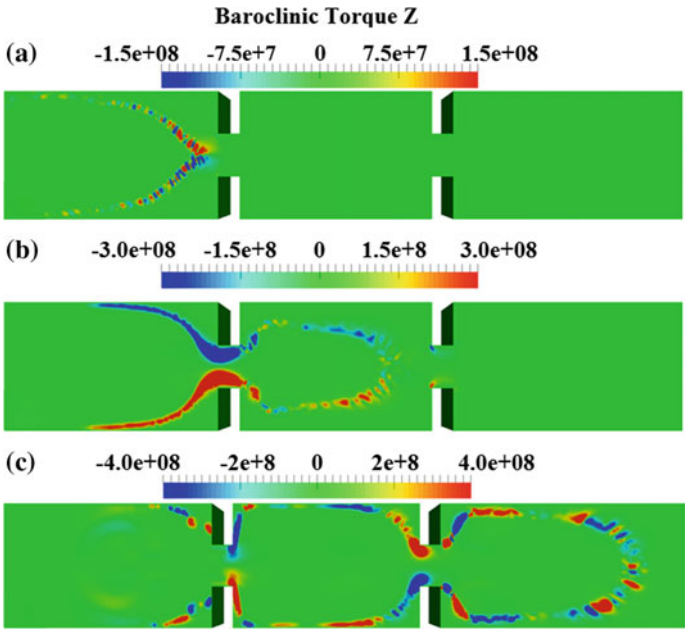


Fig. 4 Baroclinic torque contour in the Z-direction of the flame acceleration of a hydrogen flame simulation with an RMXiFoam solver, for BR = 60% at **a** time = 1.86 ms, **b** time = 2.16 ms, and **c** time = 2.409 ms. The unit of baroclinic torque is $1/s^2$

baroclinic torque would increase with time (around $-1e + 9$ to $+1e + 9(1/s^2)$). The present predictions are, however, smaller, and as a result, no DDT or RM instability occurred.

4.2 Predictions Using the Density-Based Solver

For better shock and detonation capturing, the newly assembled density-based solver is used.

Figure 5 shows that the predicted flame position and flame tip speed are in reasonably good quantitative agreement between the measurement [3] and the present predictions of the VCEFoam solver. It can be seen (Fig. 5) that the flame velocity rises continuously in the obstructed part of the channel (around the seventh obstacle, $x \leq 2.05$ m) due to flame interaction with the obstacles, resulting in combustion-induced expansion and turbulence generation.

Figure 6 shows a good qualitative agreement between the experimental observations and the predicted OH distribution of homogeneous hydrogen flame DDT with the density-based solver. In Fig. 6a, it can be seen that the flame front is planer and moving toward the obstacle (the seventh obstacle in the tube), and that two strong

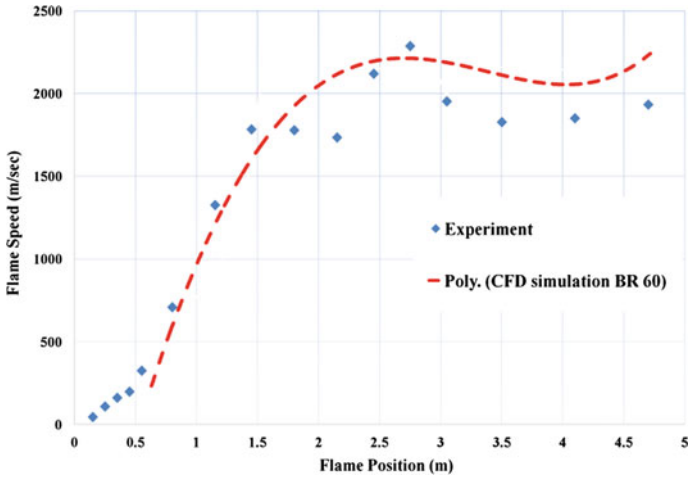


Fig. 5 Comparison between the predicted and measured flame positions [18] for BR60% and an inhomogeneous mixture of 30% Vol. hydrogen on average

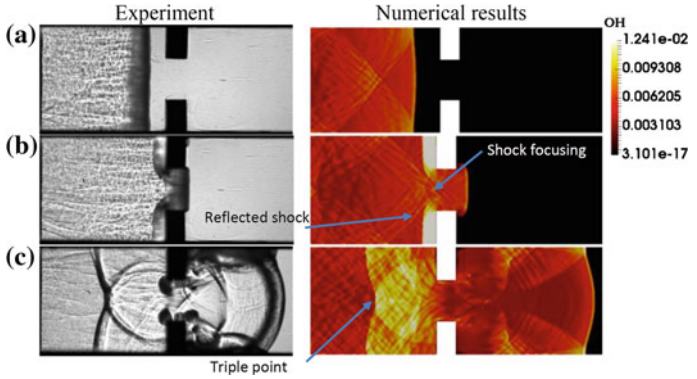


Fig. 6 Comparison of experimental results [3] of DDT of a homogenous 30% Vol. hydrogen in BR 60%, and $x = 2$ m, (obstacle 7), with numerical results of the OH distribution from VCEFoam in; **a** time = 9.42 ms, **b** time = 9.44 ms, **c** time = 9.48 ms

shock waves are generated behind the top and bottom obstacles. Figure 6b shows that these two reflected shock waves are interacting in the middle of the tube and, as the result of shock-focusing phenomena, a refraction shock wave will be generated in the downstream of flow. Then, the generated shock wave interacts with the flame front and triggers a transition to detonation (Fig. 6c). Following DDT, the shock-detonation structure can be seen, including a strong reflected shock wave, as well as a leading shock ahead of the flame front. Moreover, pressure gradient frames show the mechanism of shock and explosion generation (Fig. 7).

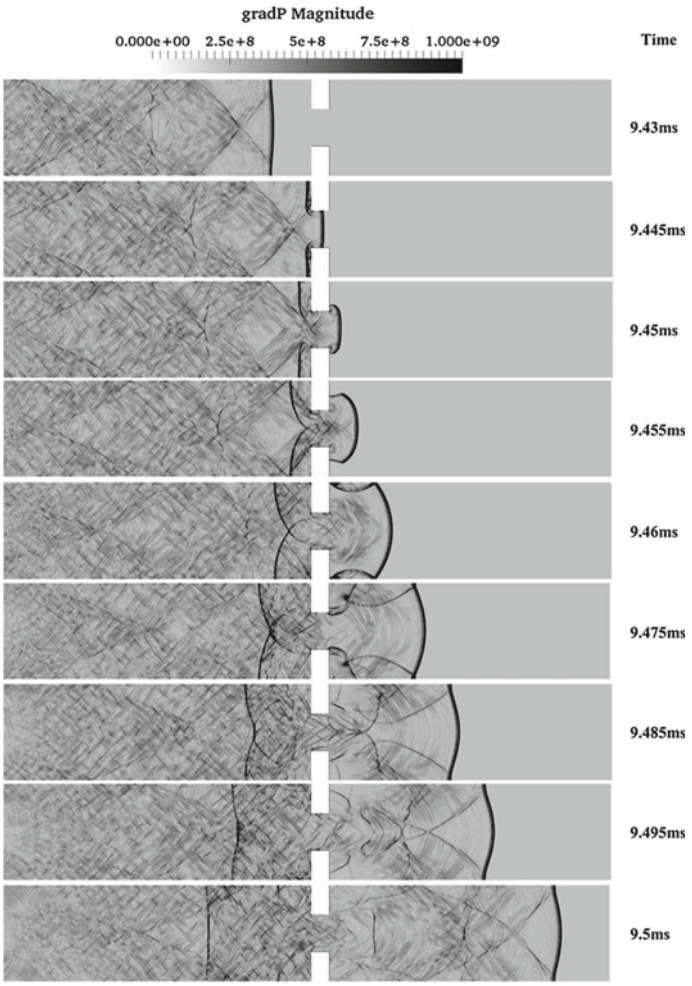


Fig. 7 The magnitude of pressure gradient contours of deflagration to detonation for a homogenous hydrogen/air mixture of 30% Vol and BR60. The unit of the pressure gradient is Pa/m

Figure 7 shows the distribution of pressure gradient magnitude in the tube around obstacle 7. These results illustrate the evolution of the shock waves. At time 9.43 ms, a planer flame is moving toward the obstacle with a strong shock wave ahead of the flame front, as well as some weaker shock waves in the upstream of the flow field. Moreover, it can be seen that the reflected shock waves from the upper and lower walls are interacting in the middle of the tube. By marching in time, it can be seen that the leading shock is interacting with the obstacle, and as a result, a strong reflected shock is generated in the flow. The reflected shocks from the upper and lower obstacles interact in the middle of the tube, which causes a shock focusing

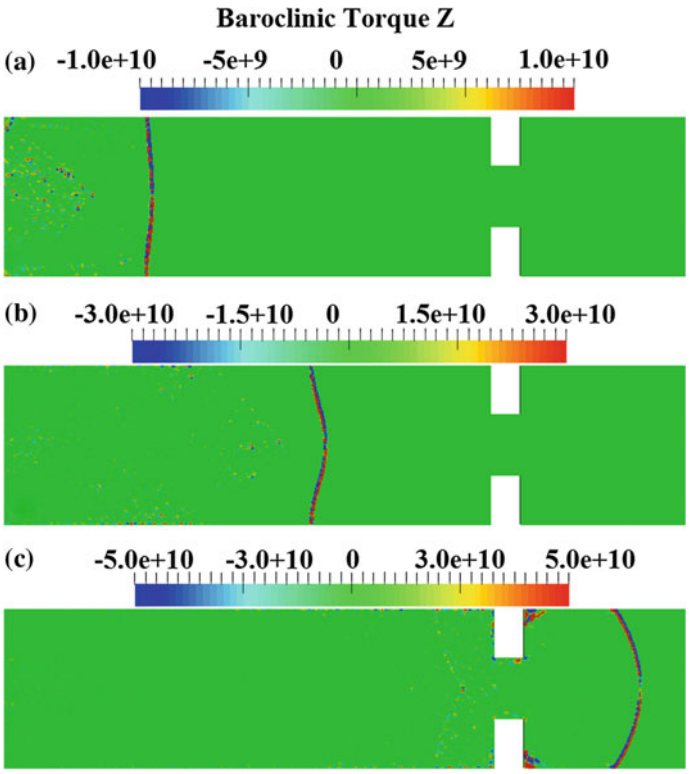


Fig. 8 Baroclinic torque contour in the *Z*-direction of DDT of a homogenous hydrogen flame simulation with the density-based solver, for BR = 60, 30% H2 at **a** time = 9.0 ms, **b** time = 9.14 ms, and **c** time = 9.18 ms. The unit of baroclinic torque is $1/s^2$

(Fig. 7, time = 9.455 ms). The reflected shock will generate a triple point in the detonation structure. Some part of the leading shock passes through the obstacle, slowly growing with the flame until it reaches the upper and lower walls of the tube (Fig. 7, time = 9.46 ms). At 9.475 ms, it can be seen that the reflected shock from the wall causes a secondary triple point in the shock–flame structure. At 9.495 ms, the reflected shock waves from the upper and lower walls meet in the middle of the tube, and it results in shock-focusing phenomena in the downstream of the flow field. These results also illustrate that the obstacles and wall in the tube have a significant effect on shock generation ahead of the flame front.

Figure 8 shows the predicted baroclinic torque distribution in the *Z*-direction with the density-based solution. It can be found that baroclinic torque is increasing with time (Fig. 8a–c), so that the pressure gradient has been, respectively, increased. Moreover, as shown in Fig. 8c, the magnitude of the baroclinic torque (−5e + 10 to +5e + 10 ($1/s^2$)) is considerably higher than in the predictions of the pressure-based

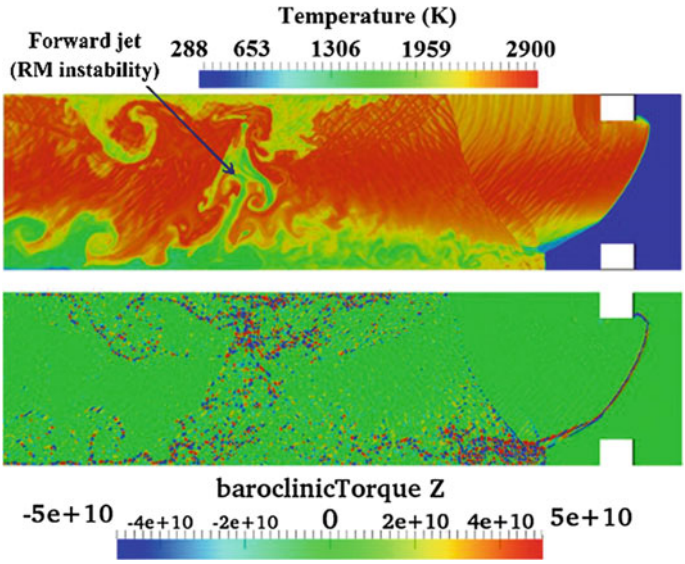


Fig. 9 Inhomogeneous hydrogen–air with 30% Vol. in the average mixture in the case with BR 30% and time = 4.7 ms; Top: temperature contour, Bottom: the predicted baroclinic torque in the *Z*-direction

model for the uniform mixture (as shown in Fig. 4). The developed density-based solver has improved the capability to model shock-detonation phenomena.

Figure 9 shows that DDT has occurred in the accelerated nonuniform hydrogen flame. The temperature contour indicates the formation of mushroom-shaped forward jets on the flame surface, which are known as Richtmyer–Meshkov (RM) instabilities [15, 20]. Furthermore, the predictions demonstrate that the overpressure at the DDT stage is higher in the nonuniform mixtures compared to the homogeneous mixtures under similar conditions. The results also provide evidence that the baroclinic torque and resulting RM instability have a substantial effect on flame acceleration and DDT. The results further suggest that higher baroclinic torque in the flow field would induce more RM instability, and hence increase the possibility of DDT.

5 Conclusion

The dynamics of deflagration-to-detonation transition (DDT), in both inhomogeneous and homogeneous mixtures, has been studied using the newly assembled pressure-based and density-based solvers within the frame of the OpenFOAM® CFD toolbox. The first stage of DDT involving flame acceleration at a subsonic level has been modeled using the pressure-based algorithm while the DDT is modeled by the density-based solver. To evaluate the contribution of the convective fluxes, the

Harten–Lax–van Leer-contact scheme is used for accurate shock capturing. For validation, the experimental data of Boeck et al. [3] is used. The predictions are in reasonably good qualitative and quantitative agreement with the experimental observations and measurements [3]. In addition, the baroclinic torque and the resulting Richtmyer–Meshkov instability are studied. These studies show that the overpressure at the transition stage is higher in the mixture with concentration gradients in comparison with the homogeneous mixtures under similar test conditions. The results of the present study recommend that, for DDT, the pressure-based solver, which is also computationally faster, can be used for the subsonic flame acceleration stage, while the density-based solver, through use of the MILES approach, should be used for the transition process. The results of the present work can be used in the context of safety to assess the potential risks of explosions in the energy industry.

Acknowledgements This work is a part of the SafeLNG project funded by the Marie Curie Action of the 7th Framework Programme of the European Union.

References

1. Ersen, K. K. (2004). Gas explosions in process pipes. Dissertation, Telemark University College, Norway
2. Thomas, G.O. (2012). Some observations on the initiation and onset of detonation. *J Phil. Trans. R. Soc.* 370(1960):715–739
3. Boeck, L. R., et al. (2015). The GraVent DDT Database. In: 25th International Colloq. on the Dynamics of Explosions and Reactive Systems (ICDERS), Leeds, UK
4. Weller, H.G., Tabor, G., Gosman, A.D. & Fureby, C. (1998). Application of a flame-wrinkling LES combustion model to a turbulent mixing layer. *Proc. Combust. Inst.* 27:899–907
5. Batten, P., Leschziner, M.A., Goldberg, U.C. (1997). Average-state Jacobians and implicit methods for compressible viscous and turbulent flows. *J. Comput. Phys*. 137(1):38–78
6. Sod, G. A. (1978). A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws. *J. Comput. Phys*. 27:1–31
7. Greenshields, C., Weller, H., Gasparini, L., and Reese, J. (2010). Implementation of semi-discrete, non-staggered central schemes in a colocated, polyhedral, finite volume framework, for high speed viscous flows. *Int J Numer Meth Fl*, 63(1):1–21
8. OpenFOAM® Ltd. http://www.openfoam.com Accessed 2015
9. Issa, R. (1985). Solution of the implicitly discretised fluid flow equations by operator-splitting. *J. Comput. Phys*. 62(1):40–65
10. Kurganov, A., and Tadmor, E. (2000). New high-resolution central schemes for nonlinear conservation laws and convection – diffusion equations. *J. Comput. Phys*. 160(1):241–282
11. Kurganov, A., Noelle, S., and Petrova, G. (2000). Semidiscrete central-upwind schemes for hyperbolic conservation laws and hamilton – jacobi equations. *J. Comput. Phys*, 160:720–742
12. Khodadadi Azadboni, R., et al. (2013). Evaluate shock capturing capability with the numerical methods in OpenFOAM®. *J Therm Sci*. 17(4):1255–1260
13. Van Leer, B. (1974). Towards the ultimate conservative difference scheme, II: monotonicity and conservation combined in a second order scheme. *J. Comput. Phys*. 17:361–370
14. Borm, O., Jemcov, A., Kau, H.P. (2011). Density based Navier Stokes solver for transonic flows. In: Proceedings of 6th OpenFOAM® workshop, penn state university, USA
15. Khodadadi Azadboni, R., et al. (2017). Numerical modeling of deflagration to detonation transition in inhomogeneous hydrogen/air mixtures. *J. Loss Prev. Process Ind.*, 49: 722–730.

16. Wilke, C. R. (1950). Diffusional properties of multicomponent gases. *J Chem. Eng. Prog*. 46(2):95–104
17. Heidari, A. and Wen, J. X. (2017). Numerical simulation of detonation failure and re-initiation in bifurcated tubes. *Int. J. Hydrogen Energy*, 42 (11):7353–7359
18. Grinstein, F., Margolin, L., Rider, W. (Eds.), (2007). Implicit Large Eddy Simulation: Computing Turbulent Flow Dynamics. Cambridge University Press, Cambridge, UK
19. Xiao, H.H, Houim, R.W., Oran, E.S. (2015). Formation and evolution of distorted tulip flames. *J. Combust. Flame*, 162(11):4084–4101
20. Khodadadi Azadboni, R., Heidari, A., Wen, X. J. (2018). A computational fluid dynamic investigation of inhomogeneous hydrogen flame acceleration and transition to detonation. *J. Flow Turbulence Combust.*, 1–13. https://doi.org/10.1007/s10494-018-9977-4

Open-Source 3D CFD of a Quadrotor Cyclogyro Aircraft

Louis Gagnon, Giuseppe Quaranta and Meinhard Schwaiger

Abstract This chapter provides a detailed method for building an unsteady 3D CFD model with multiple embedded and adjacent rotating geometries. This is done relying solely on open-source software from the OpenFOAM® package. An emphasis is placed on interface meshing and domain decomposition for parallel solutions. The purpose of the model is the aerodynamic analysis of a quadrotor cyclogyro. The challenging features of this aircraft consist of a series of pairwise counterrotating rotors, each consisting of blades that oscillate by roughly 90° about their own pivot point. The task is complicated by the presence of solid features in the vicinity of the rotating parts. Adequate mesh tuning is required to properly decompose the domain, which has two levels of sliding interfaces. The favored decomposition methods are either to simply divide the domain along the vertical and longitudinal axes or to manually create sets of cell faces that are designated to be held in a single processor domain. The model is validated with wind tunnel data from a past and finished project for a series of flight velocities. It agrees with the experiment in regard to the magnitude of vertical forces, but only in regard to the trend for longitudinal forces. Comparison of past wind tunnel video footage and CFD field snapshots validates the features of the flow. The model uses the laminar Euler equations and gives a nearly linear speedup on up to four processors, requiring 1 day to attain periodic stability.

L. Gagnon (✉) · G. Quaranta
DAER, Politecnico di Milano, Milan, Italy
e-mail: louis.gagnon@polimi.it

G. Quaranta
e-mail: giuseppe.quaranta@polimi.it

M. Schwaiger
Innovative Aeronautics Technologies GmbH, Traun, Austria
e-mail: amxgmbh@gmail.com

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_27

1 Background

The cyclogyro is an aircraft that uses cycloidal rotors to generate propulsion. Cycloidal rotors are still largely unexplored by the aeronautic world. As opposed to conventional propellers, they produce forces that can change direction almost instantly on a 360° plane. Various studies have relied on these rotors to propel aircraft [23, 26, 29], micro-aircraft [2–4, 24] and airships [16, 17, 20, 30]. They are also used commercially to propel boats such as water tractors [1] and drillships [22]. Furthermore, they have been studied for wind [7, 8, 19, 25] and water [19, 25] turbines.

A cycloidal rotor, as illustrated in Fig. 1a, is an arrangement of symmetric blades of constant cross section that rotate about a central drum. That drum transmits the spinning motion to the blades through a series of pivot rods. Each blade pitches individually about its intersection point with its pivot rod. The blade pitching motion is transmitted through the pitch rods, which are themselves offset by a central mechanism within the drum. Consequently, the total thrust generated by the rotor is the sum of individual blade lift and drag forces. The D-Dalus, shown in Fig. 1b, is a four-rotor cyclogyro aircraft prototype developed by IAT21 [27, 28]. It relies solely on cycloidal rotors for thrust generation and is the object of this study. The actual aircraft prototype is shown in Fig. 1c. The rotor blades have 6 cm chords, while their span and the rotor diameter are both 24 cm. The pivot rods are attached to the blades slightly in front of the chord midpoint and allow pitching from −37° to 35°. The endplates have a 1 cm thickness and a 29 cm diameter.

The main purpose of the developed 3D CFD model is to observe the aerodynamic rotor–aircraft–rotor interaction. A better understanding of flow interaction arises [13] from the use of this model and a more informed aircraft design process can be conducted. CFD models for this type of aircraft have not been published before.

(a) Cycloidal rotor sketch. (b) Aicraft concept render. (c) Actual aircraft.

Fig. 1 Rotor and aircraft for which the CFD model is created

2 CFD Model

The CFD model is tridimensional and uses the finite volume method to solve the PIMPLE algorithm, which consists of a merger of the PISO and SIMPLE algorithms. In OpenFOAM® 2.4.x, which is the version used for this project, this is achieved by using the *pimpleDyMFoam* solver. One pressure correction step is used and the pressure–momentum coupling is calculated twice. A bounded first-order implicit discretization scheme is used on the time derivative. A Gaussian integration with linear interpolation is used for the derivative terms of pressure and velocity, with bounding on velocity. A second-order upwind interpolation scheme is used for the advection of velocity. Linear interpolation is used for the Laplacian terms, with an underrelaxed face gradient corrected for mesh non-orthogonality. The convergence tolerance on the residual is 10^{-6} for both pressure and velocity. Prior to the end of the iteration loop, fields are also considered converged if the pressure and velocity residuals become 1 and 10% of their initial residuals, respectively. Although the solver is designed for Navier–Stokes equations, the Euler laminar equations are instead solved by setting the viscosity to zero. Air density is 1.204 kg/m^3. The main motivator for ignoring the effects of viscosity is to reduce the required computer time. The omission of viscosity is justified because the dynamics of rotors are dominated by pressure contributions and dynamic effects. This is also demonstrated by a study [21] that showed marginally small differences between experimental, Euler, and Navier–Stokes results for a helicopter rotor. A total of 14 moving meshes use sliding interfaces of interpolation. They are solved by the Arbitrary Mesh Interface (AMI) algorithm [9] and are shown in Fig. 2. Each rotor blade is inserted into a double AMI. The outer AMIs rotate and the inner ones strongly oscillate. The model relies on an embedded moving mesh algorithm [14] and an accompanying moving wall slip boundary condition [15] that were previously created [10] and publicly released. The embedded moving mesh is based on a regular oscillating mesh method, called the *oscillatingRotatingMotion* class in OpenFOAM®. It incorporates a new origin vector $\mathbf{o_c}$,

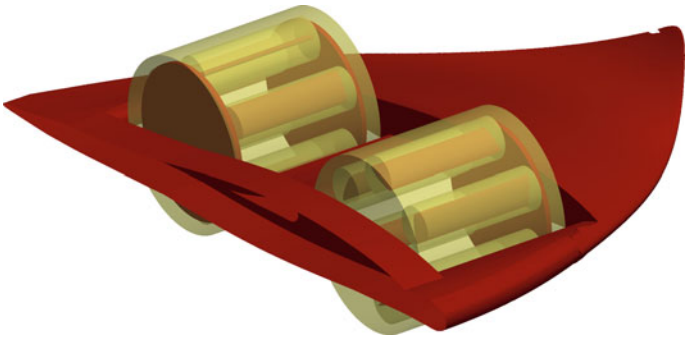


Fig. 2 The 14 AMI cylinders used for the cyclogyro aircraft, of which 12 are embedded

$$\mathbf{o_c} = \mathbf{o_o} + r_o\{\text{sign}(\omega_o)\cos(|\omega_o|t + \phi_o\pi), \sin(|\omega_o t| + \phi_o\pi), 0\}, \tag{1}$$

where $\mathbf{o_o}$, r_o, ω_o, and ϕ_o are the outer AMIs origin, radius, angular velocity, and initial offset, respectively, and t is the time. The $\mathbf{o_c}$ vector is applied as the new origin of the transformation septernion.[1] The moving wall slip is based on a moving wall boundary condition, called the *movingWallVelocityFvPatchVectorField* class in OpenFOAM®, and imposes the normal velocity vectors of the field as

$$\mathbf{n}\left(\mathbf{n}\cdot\left(\mathbf{u_p} + \mathbf{n}(u_n - (\mathbf{n}\cdot\mathbf{u_p}))\right)\right), \tag{2}$$

where $\mathbf{u_p}$ is the wall velocity, $\mathbf{n}$ is the unit normal to the wall and u_n is the wall mesh flux per area. The tangent velocity is taken as the planar vector component of the velocity adjacent to the wall.

The single rotor meshes with and without endplates have roughly 1 million and 300,000 cells, respectively. The endplates make the flow more two-dimensional, both in experiment and simulation. A first harmonic sinusoidal pitching schedule is imposed on the blades of the rotor. The blade angle, θ, with respect to a line perpendicular to the pivot rod, is

$$\theta = \theta_o + \theta_s \sin(\omega t + \phi), \tag{3}$$

where θ_o is the fixed pitch angle offset, θ_s is the magnitude of the pitch angle oscillations, ϕ is the imposed phase angle, and ω is the constant rotor angular velocity. The purpose of θ_o is to increase the pitch angle on the bottom part of the rotation cycle to counter the stronger inflow. The position of the maximum pitch is anticipated by ϕ with respect to the bottommost angular position in order to counter the aerodynamic delay.

The aircraft is fixed in space, and thus the model disregards the inertial effects of gravity and aerodynamic forces. Careful tuning of the mesh interfaces allows us to keep the actual geometry of the aircraft. The only change is that the spanwise distance between the endplates and the rotor blades is slightly increased, to roughly one-tenth of the chord length. A gap of this size has the same effect as if the endplate were attached to the foil [5], and is thus negligible. The space available in the physical aircraft between the rotor blades' pivot points and the airframe allows us to have an AMI cylinder radius at least equal to the maximum distance between the pivot point and any edge of the blade. The blades can thus pitch at any angle.

[1] The septernion is a seven component array used in OpenFOAM® composed of a translation vector and a rotation quaternion.

2.1 Mesh Generation

The *snappyHexMesh* hexahedral meshing tool is used to generate the mesh. A description of this mesher is given to introduce concepts that clarify the generation of embedded AMI interfaces in a very narrow space. The mesher inserts imported CAD geometries into a structured volume mesh. It then refines the volume mesh near and on surfaces; in portions of surfaces that are close to other surfaces; and inside user-defined regions. Refinement is applied as a user-specified number of subdivisions to the original structured mesh. Cells are refined either inside or within a specified distance of a given region or when intersecting a given surface. Once the mesh is refined, the cell faces are moved so as to smoothly adhere to the boundaries, which can be wall boundaries or simple reference geometries. This last option allows us to create the sliding interfaces of the blade oscillating zones and the rotor spinning zones. The size of the mesh and the time it takes to generate are controlled by quality and iteration options. While this chapter focuses on the aspects critical for the cycloidal rotor aircraft application, a detailed guide to the mesher is available online [18].

2.2 Isolated Airframe Mesh

Before initiating the actual modeling of the cyclogyro, a separate mesh quality evaluation campaign is conducted for the airframe taken alone without the rotors. The impact of mesh refinement on the airframe alone is studied in order to obtain the smallest possible grid while having a mostly mesh-independent solution with low discretization error. Eight different meshes are generated and evaluated. The tests are all done at a 15° airframe angle of attack and a 30 m/s flow velocity. Different meshing techniques are also studied and the influence of different mesh parameters on the force results is examined. These parameters are the value of the surface feature angles that trigger mesh refinement, the refinement level of the mesh in the wake zone, and the increase of the overall mesh density. Table 1 shows the mesh attempts, along with the parameters of interest and the number of cells, which go from 293,000 to 1.7 million. The first three cases, *baseline*, *halfSize*, and *thirdSize*, are meshes created with identical parameters. Their only difference is that the initial structured volume mesh cells of *halfSize* and *thirdSize* are 1/2 and 1/3 the sizes of those of *baseline*, respectively. The table also shows the normalized mean longitudinal, vertical, thrust, and moment forces obtained for each mesh over a period equivalent to one rotation at 3750 rpm. One period takes 1300–7800 timesteps, depending on the refinement level of the mesh. The numbers in the surface column of Table 1 are the minimum and maximum number of divisions to apply to the structured mesh cells that encounter the airframe surface. The numbers in the *Distance* column are the respective number of divisions to apply to the cells that are located at 1, 20, and 50 cm from the airframe.

The forces obtained using any of these meshes reach a fairly constant value after 10 periods. At this point, the longitudinal and vertical forces oscillate by less than

Table 1 Airframe mesh refinement tests

Test	Surface	Distance	Wake zone	kCells	100Fx	Fy	T	M
baseline	5 8	6 3 1	No	293	−3.93	3.31	3.31	−1.47
halfSize	5 8	6 3 1	No	747	−6.22	3.15	3.15	−1.36
thirdSize	5 8	6 3 1	No	1729	5.50	2.75	2.75	−1.20
angles[a]	5 9	5 3 1	No	507	−4.53	3.22	3.22	−1.39
surf	6 9	6 3 1	No	524	−4.80	3.26	3.26	−1.40
surf2	7 8	6 3 1	No	423	−1.98	3.19	3.19	−1.41
wake	4 8	5[b]	Yes[c]	507	−5.61	3.12	3.12	−1.40
noWake	4 8	5[b]	No	492	−4.56	3.14	3.14	−1.41

[a]Attempt at changing the *featureAngle* value (surface feature angles that trigger refinement)
[b]At a 25 cm distance
[c]3 levels of refinement inside and 2 levels within 1 m of the wake zone

2 and 9% of the thrust, respectively. These oscillations are caused by the vortex shedding that occurs on the airframe at a 15° angle of attack. The magnitudes of these oscillations are not linked to the refinement level of the mesh, but the most refined mesh does take longer to stabilize. Table 1 also indicates that for the case studied, the level of refinement on the airframe influences the thrust by 8% of its maximum value. That value drops to 6% when the average thrust is measured over more periods. Finally, the coarsest case is run for 3 s, which is equivalent to 200 periods, in order to see the long-term tendency of the flow. It is shown that the average forces remain almost constant over time, with an oscillation in the mean lift generated of roughly 1%.

2.3 Rotor Model

The rotor model is initiated by enhancing a simulation from a previous project [11, 12], which had been validated against experimental data [30] for a larger rotor without endplates. That prior CFD simulation had been shown to yield more accurate quantitative results than its analogous 2D version. It had also shown that the size and velocity of the inlet and outlet boundary conditions have little influence on the rotor flow features and forces. The difference between simulation and experiment was below 20% for the power, with a much better agreement at a low pitch angle and at low angular velocities. For thrust, the errors were contained at low angular velocities but reached problematic magnitudes at higher angular velocities. That existing simulation had been used for proof of concept simulations and had not been tested for stability. It is thus reconfigured to match the new geometry, which is roughly proportional, 3 times smaller, and has different pivot points for the blades. The mesh is tweaked to allow for locating an oscillating sliding interface between the rotor and its blades. Mesh tweaking also ensures validity over a range of rotor angular velocities,

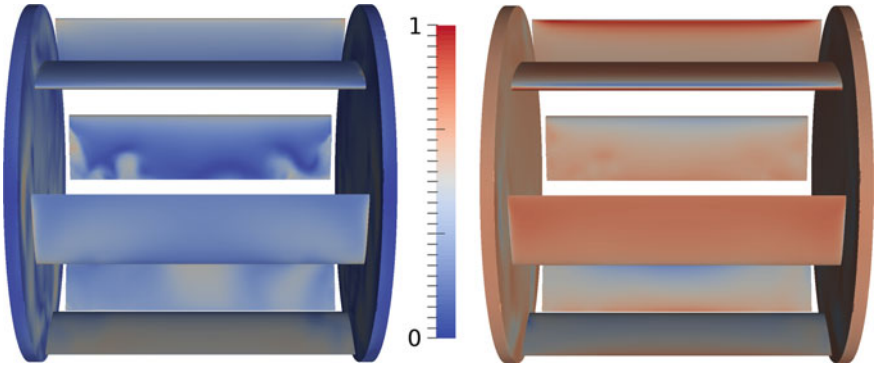


Fig. 3 Visualization of the surface pressure and velocity on the preliminary rotor simulation

which reach roughly 7 times the maximum angular velocity of the previous model. The presence of endplates considerably increases the model's complexity. This is due to the very small space between the oscillating blades, the rotating endplates, and, eventually, the airframe. The spacing between the blades and the endplates is only 3% of the blade span. Thus, the mesher is forced to move faces from cells lying in a very narrow zone and make them adhere to the airframe, to the rotor and blade interfaces, and to the blades. This zone, therefore, requires a carefully refined mesh. The mesh is completely parametrized to allow automatic generation for different geometries and to have a stable and repeatable meshing procedure.

The thrusts obtained for the rotor alone match up in order of magnitude with the experimental data from the same rotor installed on a fixed apparatus in calm air. This confirms that the model is properly set up and the preliminary model development is deemed complete. Figure 3 shows qualitative results from that preliminary single rotor model. The simulation is set up with a rotational velocity of 3970 RPM, a null incoming wind velocity, and a mesh size of 1.3 million cells.

2.4 Entire Aircraft Mesh

The rotor model is combined with a second rotor and the half D-Dalus L1 airframe to create the full aircraft model, using a symmetry about the central plane. The mesh separating the various AMIs, the endplates, and the airframe is very delicate, and thus several iterations of the parametrized mesh generation is undertaken. The important parameters for generating a cyclogyro mesh are described in Table 2.

A proof of concept model is then created with a preliminary mesh. Its purpose is to develop into a working model that converges for the most unstable flow condition before refining the mesh until satisfactory validation results are reached. It represents the most unstable case that can be expected to be encountered during the simulations and its purpose is to verify the robustness of the model. The airframe angle of attack

Table 2 Mesh parameters for *snappyHexMesh* and brief explanations

Parameter	Value	Explanation
addLayers	False	Layers are not needed in a nonviscous simulation
maxGlobalCells	8×10^6	This number limits the mesh size when the level refinement required would otherwise surpass maxGlobalCells
faceType	Baffle	This creates duplicate patches at the AMIs
implicitFeatureSnap	True	Uses the implicit method for finding refinement surfaces
detectNearSurfacesSnap	True	Prevents cell faces from adhering to a nearby surface by mistake

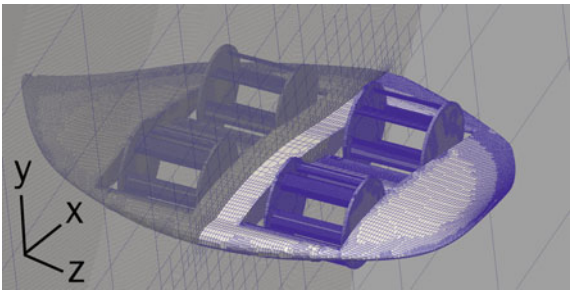


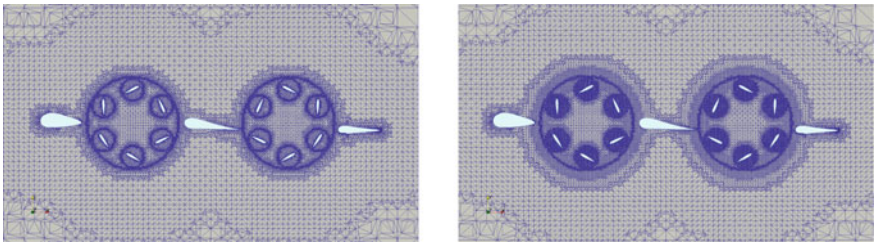
Fig. 4 Mesh of the proof of concept simulation shown along with its symmetry plane

is 15° and the horizontal incoming wind velocity is 30 m/s. The mesh has 2.7 million cells and is shown in Fig. 4. The boundary conditions are null normal gradient for pressure and fixed velocity at the inlet. At the outlet, they are ambient pressure and null normal velocity gradient, which becomes a null velocity when the flow attempts to reenter the domain. The latter velocity condition is referred to as *inletOutlet* in OpenFOAM® jargon. On the outside walls and on the aircraft body, a slip velocity and null normal pressure gradient are used. Finally, on the rotor blades and endplates, the conditions are null normal pressure gradient and the developed moving slip velocity condition. The modeled flow domain is 5.3, 20.5, and 4.3 times the half-aircraft's corresponding lengths in the longitudinal, vertical, and span directions, respectively.

2.5 Final Mesh Tuning

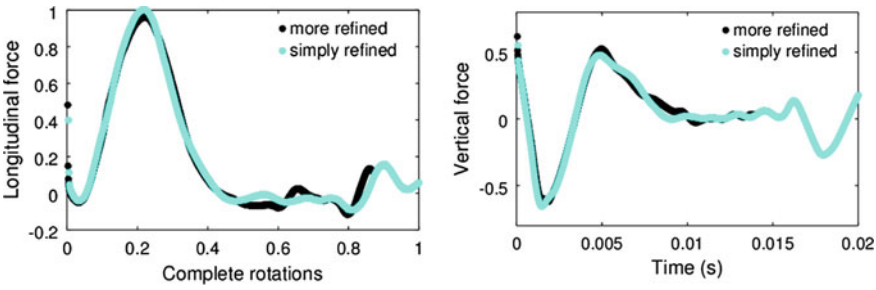
Once the full aircraft model is ready, a final mesh refinement is performed. The *simply refined* and the *more refined* meshes are created with 3.7 million and 5.7 million cells, respectively.

The *more refined* mesh has a greater refinement zone around the rotors and a finer grid within each inner AMI cylinder, as shown in Fig. 5. It also has a wider wake zone



(a) Simply refined. (b) More refined.

Fig. 5 Comparison between the simply and more refined meshes



(a) Drag. (b) Lift.

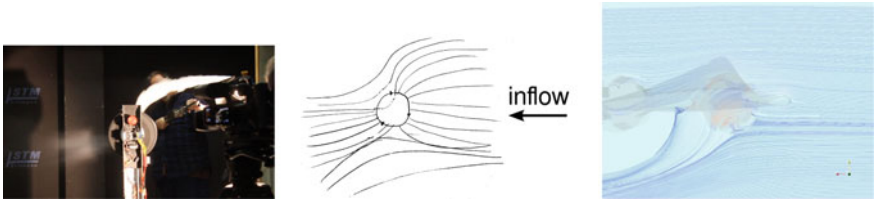
Fig. 6 Comparison of drag and lift on one foil of the rear rotor for both refinement levels over the first simulation cycle

that extends up to the front of the aircraft to fit with both hover and forward flight conditions. Nonetheless, both meshes yield very similar force results right from the start of the simulation. That match between both cases is shown for a foil of the rear rotor, being the most perturbed rotor, in Fig. 6. There are still small differences in values which indicates that a completely mesh-independent solution has not been fully reached. Nevertheless, the smaller, less refined mesh is kept, because both solutions are very close. This avoids doubling the solution time, as required by the *more refined* case, and allows us to run the number of required analyses within the fixed project timeframe. This constraint is further reinforced by the limitations of the sliding interface domain decomposition, which is covered in Sect. 3. However, widening the wake zone has very little time cost and only adds a small number of cells. Thus, a final mesh relying on refinement far from the wall and consisting of a slight improvement of the *simply refined* case is used for the definitive model. It has 4.5 million cells, and the solution is periodically stable after six rotations, because the rotors have a dominant effect on the flow and cause stability to be reached faster than for the airframe alone.

2.6 Validation

The main challenge of the validation is that no wind tunnel data had been gathered while both the front and rear rotors were in use. The experimental operation having been completed and resigned to the past, no more data can be obtained. It follows that the experimental data available is for the quadrotor cyclogyro propelled by the two front rotors alone. No velocity information is available for the rear rotor.

The values of drag and lift obtained by CFD are nevertheless compared to the experimental data from the wind tunnel. The highest wind tunnel velocity is chosen as a basis for validation and to approximate the angular velocity for the unpowered rear rotors during the wind tunnel tests. Attempts with various rear rotor angular velocities lead to the conclusion that the fixed rear rotor most adequately reproduces the experiment. Rotor flow visualization from past experiments matches the CFD streamlines of the powered rotor, as shown in Fig. 7. The trend of thrust matches that of the wind tunnel at 10, 15, 20, and 25 m/s, as shown in Fig. 8. The agreement in vertical forces is the main objective of the project, and this justifies neglecting the contribution of the viscous forces on the airframe. The remainder of the validation process is reported in the article that focuses on the aerodynamics of the aircraft [13].



(a) Snapshot from wind tunnel. (b) Wind tunnel streamlines. (c) Powered rotor.

Fig. 7 Streamlines of the rotor in wind tunnel compared to the powered aircraft rotor

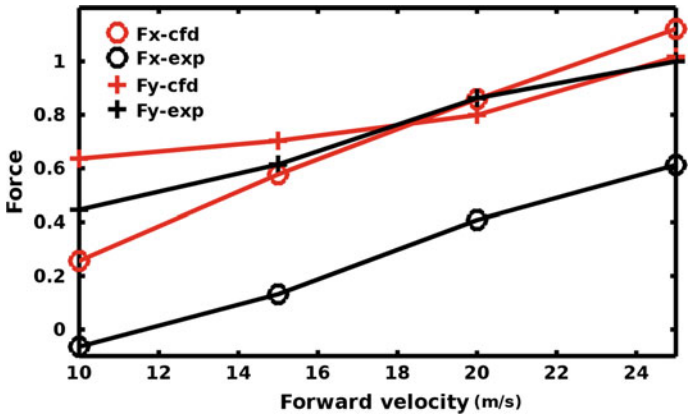


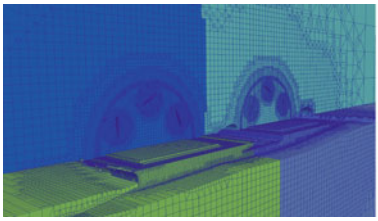
Fig. 8 Trend match between CFD (red) and wind tunnel (black)

3 Domain Decomposition Parallelization

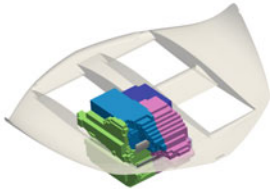
This section presents the method developed in order to fully solve the aircraft in parallel. This starts with parallel meshing, which is followed by a parallel CFD solution, and finally by a parallel visualization. This last one does not require any tuning and is done using a recent version of ParaView. The whole process is achieved locally on a 12-core machine. For the solution phase, the most efficient parallelization strategy is to divide the domain along the vertical and longitudinal axes, leaving an equal number of cells on each side. This decomposition method is called the *simple* method in *snappyHexMesh* jargon. That method reduces the communication across processors for sliding AMI interface pairs to a minimum. Figure 9a shows the four processor submeshes obtained with the *simple* decomposition algorithm. Solving this case in parallel takes roughly 1 day, instead of 4 days, to reach a periodically stable solution.

The part of the process that benefits the most from default parallelization is the meshing process of *snappyHexMesh*, for which the method can be found in the OpenFOAM® tutorials. Diversely, using the default options for AMI interface decomposition with the Scotch [6] algorithm, the solving phase of the simulation has an increase in speed that ranges between 45 and 95% on 2 processors and 100% on 10 processors. An equivalent simulation without the AMIs yields a 350% speedup on 10 processors. The cause is that the AMIs are distributed over different processors, and thus communication is slowed down. This decomposition of the AMIs can be seen for a 10 processor mesh in Fig. 9b, where each color represents 1 processor domain. Coincident sliding interface boundaries, referred to as the master and slave AMI patches in OpenFOAM®, maintain their matching cell faces on the same processor, but the patches themselves are split into two or more portions. This is visible in Fig. 10a and in the close-up in Fig. 10b. The interface irregularities force the AMI cells of one processor to communicate with those of another processor as soon as they start rotating. The extra communication step between processors at the AMI slows down the simulation.

Thus, the goal is to maintain the whole AMI, with its master and slave patches, on a single processor. A summary of the available methods and their observed behaviors

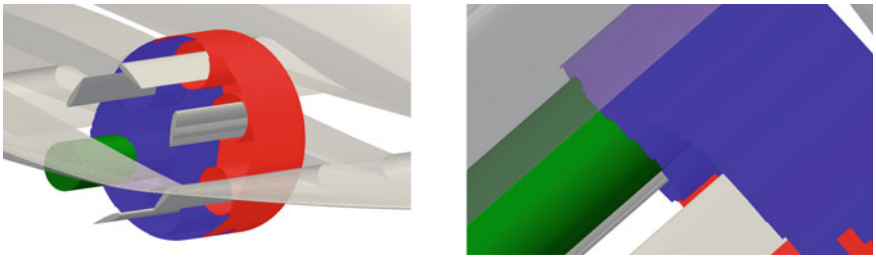


(a) 4 processors *simple* decomposition.



(b) Part of a 10 processor mesh.

Fig. 9 Decomposition methods with distinct processor colors



(a) Rotor view. (b) Close-up on AMI.

Fig. 10 AMI cells distribution over different processors

Table 3 Available options to preserve mesh zones on a specific processor

Method	Description	Result
preserveFaceZones	Preserves face zones on a single processor	Is not effective in doing so
preservePatches	Ensures the patches[a] given are meshed on a single processor	The coincident sliding interface boundaries are, however, not always meshed on the same processor
singleProcessorFaceSets	Ensures that the given face set is meshed on a single processor	Using a trick, it is possible to define a whole volume as a face set, and thus obtain that the meshing algorithm maintains that volume on a single processor

[a]A patch in OpenFOAM® consists of a wall, an interface, or any continuous set of cell faces that represents a surface

is given in Table 3. The Scotch method that uses these options does not automatically yield an efficient AMI decomposition. Also, if not carefully controlled, the decomposition creates more than one cell block for a single processor in an attempt to respect the given interface constraints. The resulting mesh thus has an increased computation time, due to each processor zone being split over parts of the domain that are not physically connected. An example of such a decomposition is shown in Fig. 11, where the blue surface is the AMI and the pale gray zones represent the submesh of a single processor. In that case, although the AMIs are preserved on a single processor, the number of different blocks for one processor cause latency. The *simple* decomposition method, with a domain division in two boxes, remains the most effective one in preserving the whole AMIs of the front and rear rotors on two different processors. Using the *simple* method with four processors, 12 of the 14 sliding interfaces are preserved on a single processor, the yields of which are a nearly linear speedup.

When the *simple* algorithm is no longer efficient due to a large number of processors, the manual creation of sets of cell faces that are designated to be held in

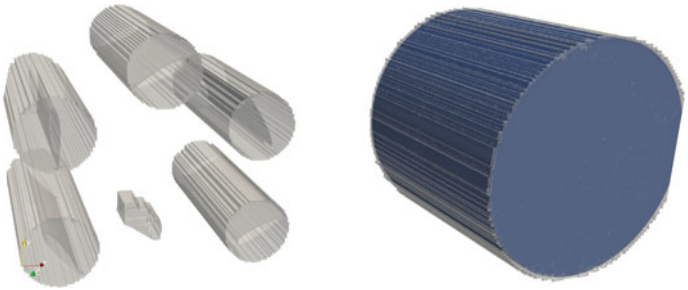


Fig. 11 Decomposed mesh showing a processor divided into multiple zones; the processor zone is in pale gray and the outer AMI interface in blue

a single processor domain can be done. This is called the *singleProcessorFaceSets* method in *snappyHexMesh* jargon. It can be explicitly defined for each processor and for each set. This Scotch method option allows for preserving all the faces of a rotor's AMIs on a single processor. The *singleProcessorFaceSets* method may be applied using the *topoSet* tool to create a set of cell faces, *faceSet*, from a set of cells, *cellSet*, using the *cellToFace* option. After running *topoSet* with these indications, decomposition must be run with the *singleProcessorFaceSets* option. Including the AMIs in the *faceSets* to be assigned to a single processor can, however, confuse the algorithm and create an unbalanced mesh. By considering zones as small as one inner AMI cylinder as single processor zones, meshes with more than 13 processors and a reasonable speedup can be obtained. In that case, an annulus processor zone may be created to ensure that the outer rotor AMIs are also meshed on a single processor. However, the effects of divided outer AMIs on the parallel speedup are less harmful than those of a multitude of divided inner embedded AMIs.

A considerable amount of time is required to implement the *singleProcessorFaceSets* method, thus the favored method remains the use the *simple* division with two or four processors. The retained procedure for running the case is to first run the mesher on any desired number of processors, then reconstruct the case as a single processor mesh, and finally redivide it into four processors using the *simple* method.

4 Closing Remarks

This chapter presented a methodology for modeling rotating and strongly oscillating components of a rotor using open-source CFD software. These rotor components can be embedded one inside another, and parallelization is fairly straightforward up to four processors through the simple division of the domain along the vertical and longitudinal axes. This is called the *simple* method. A more efficient and refined parallelization could be obtained by manually creating sets of cell faces that are designated to be held in a single processor. This is called the *singleProcessorFaceSets*

decomposition method and can eventually be parametrized to allow for large-scale parallel solutions. However, such a process requires a significant set up time that may be rewarding only if a large number of analyses is foreseen. Special care is necessary when generating the mesh near the rotating interfaces and when choosing the decomposition methods. A study of the impact of refinement on the airframe was conducted to grasp the impacts of surface- and region-based refinement levels. The final mesh is small enough to allow the CFD to be solved in one day on four processors, yet refined enough to grasp the important flow features and forces. The best meshes were generated by allowing large cells on nearly flat surfaces and refining near the sliding interfaces. The CFD was done using the laminar Euler equations of the *pimpleDyMFoam* solver. A brief validation section, based on prior experiments both on the rotor alone and on the full aircraft inside a wind tunnel, was presented. The methods from this article, in combination with the available OpenFOAM® tutorials, can be used as a starting point for modeling similar rotating machines.

Acknowledgements The research presented in this paper was supported by the Austrian Research Promotion Agency (FFG) *Basis programe* research grant #849514: *Entwicklung des Fluggertes D-Dalus L2 als eigenstabil flugfähigen Prototypen*.

References

1. Barrass C (2004) Chapter 22 - Improvements in propeller performance. In: Barrass C (ed) Ship Design and Performance for Masters and Mates, Butterworth-Heinemann, Oxford, pp 218–227, 10.1016/B978-075066000-6/50024-6, URL http://www.sciencedirect.com/science/article/pii/B9780750660006500246
2. Benedict M, Ramasamy M, Chopra I, Leishman JG (2009) Experiments on the Optimization of MAV-Scale Cycloidal Rotor Characteristics Towards Improving Their Aerodynamic Performance. In: American Helicopter SocietyInternational Specialist Meeting on Unmanned Rotorcraft, Phoenix, Arizona
3. Benedict M, Ramasamy M, Chopra I, Leishman JG (2010) Performance of a Cycloidal Rotor Concept for Micro Air Vehicle Applications. Journal of the American Helicopter Society 55(2):022,002–1–14, doi:https://doi.org/10.4050/JAHS.55.022002
4. Benedict M, Mattaboni M, Chopra I, Masarati P (2011) Aeroelastic Analysis of a Micro-Air-Vehicle-Scale Cycloidal Rotor. AIAA Journal 49(11):2430–2443, https://doi.org/10.2514/1.J050756
5. Calderon DE, Cleaver D, Wang Z, Gursul I (2013) Wake Structure of Plunging Finite Wings. In: 43rd AIAA Fluid Dynamics Conference
6. Chevalier C, Pellegrini F (2008) Pt-scotch: A tool for efficient parallel graph ordering. Parallel Computing 34(6–8):318–331, https://doi.org/10.1016/j.parco.2007.12.001, URL http://www.sciencedirect.com/science/article/pii/S0167819107001342, parallel Matrix Algorithms and Applications
7. Darrieus GJM (1931) Turbine having its rotating shaft transverse to the flow of the current. URL https://encrypted.google.com/patents/US1835018, US Patent 1,835,018
8. El-Samanoudy M, Ghorab AAE, Youssef SZ (2010) Effect of some design parameters on the performance of a Giromill vertical axis wind turbine. Ain Shams Engineering Journal 1(1):85–95

9. Farrell P, Maddison J (2011) Conservative interpolation between volume meshes by local Galerkin projection. Computer Methods in Applied Mechanics and Engineering 200(1–4):89–100, https://doi.org/10.1016/j.cma.2010.07.015, URL http://www.sciencedirect.com/science/article/pii/S0045782510002276
10. Gagnon L, Morandini M, Quaranta G, Muscarello V, Bindolino G, Masarati P (2014a) Cyclogyro Thrust Vectoring for Anti-Torque and Control of Helicopters. In: AHS 70th Annual Forum, Montréal, Canada
11. Gagnon L, Quaranta G, Morandini M, Masarati P, Lanz M, Xisto CM, Páscoa JC (2014b) Aerodynamic and Aeroelastic Analysis of a Cycloidal Rotor. In: AIAA Modeling and Simulation Conference, Atlanta, Georgia
12. Gagnon L, Morandini M, Quaranta G, Muscarello V, Masarati P (2016) Aerodynamic models for cycloidal rotor analysis. J of Aircraft Engineering and Aerospace Technology 88(2):215–231
13. Gagnon L, Quaranta G, Schwaiger M, Wills D (2017) Aerodynamic analysis of an unmanned cyclogiro aircraft. submitted to SAE Tech Papers
14. Gagnon, L (2014a) Interface within an interface c++ code. http://www.cfd-online.com/Forums/openfoam-solving/124586-dynamic-mesh-within-dynamic-mesh.html#post476869, last accessed Feb. 2016
15. Gagnon, L (2014b) Slip moving wall boundary condition c++ code. http://www.cfd-online.com/Forums/openfoam-solving/105274-free-slip-moving-wall-bc.html#post509989, last accessed Feb. 2016
16. Gibbens R (2003) Improvements in Airship Control Using Vertical Axis Propellers. In: Proceedings of AIAA's 3rd Annual Aviation Technology, Integration, and Operations (ATIO) Forum, https://doi.org/10.2514/6.2003-6853
17. Gibbens R, Boschma J, Sullivan C (1999) Construction and testing of a new aircraft cycloidal propeller. In: Proceedings of 13th Lighter-Than-Air Systems Technology Conference., https://doi.org/10.2514/6.1999-3906
18. Greenshields, C (2016) OpenFOAM® User Guide: 5.4 Mesh generation, snappyHexMesh. http://cfd.direct/openfoam/user-guide/snappyhexmesh/, last accessed Sept. 2016
19. Hwang IS, Lee HY, Kim SJ (2009) Optimization of cycloidal water turbine and the performance improvement by individual blade control. Applied Energy 86(9):1532–1540
20. Ilieva G, Páscoa JC, Dumas A, Trancossi M (2012) A critical review of propulsion concepts for modern airships. Central European Journal of Engineering 2(2):189–200, doi:https://doi.org/10.2478/s13531-011-0070-1
21. Kim JW, Park SH, Yu YH (2009) Euler and Navier-Stokes Simulations of Helicopter Rotor Blade in Forward Flight Using an Overlapped Grid Solver. In: 19th AIAA Computational Fluid Dynamics Conference Proceedings, paper AIAA 2009-4268
22. Koschorrek P, Siebert C, Haghani A, Jeinsch T (2015) Dynamic Positioning with Active Roll Reduction using Voith Schneider Propeller. IFAC-PapersOnLine 48(16):178–183, https://doi.org/10.1016/j.ifacol.2015.10.277, URL http://www.sciencedirect.com/science/article/pii/S2405896315021680, 10th IFAC Conference on Manoeuvring and Control of Marine Craft MCMC 2015Copenhagen, 24–26 August 2015
23. Leger JA, Páscoa JC, Xisto CM (2016) Aerodynamic Optimization of Cyclorotors. Accepted by J of Aircraft Engineering and Aerospace Technology
24. Lind A, Jarugumilli T, Benedict M, Lakshminarayan V, Jones A, Chopra I (2014) Flow field studies on a micro-air-vehicle-scale cycloidal rotor in forward flight. Experiments in Fluids 55(12):1826, https://doi.org/10.1007/s00348-014-1826-1
25. Maître T, Amet E, Pellone C (2013) Modeling of the flow in a Darrieus water turbine: Wall grid refinement analysis and comparison with experiments. Renewable Energy 51:497–512
26. McNabb ML (2001) Development of a Cycloidal Propulsion Computer Model and Comparison with Experiment. Master's thesis, Mississippi State University
27. Schwaiger M (2010) Aircraft: US Patent 7735773 B2. http://www.freepatentsonline.com/7735773.html, last accessed Feb. 2016

28. Schwaiger M (2014) Aeroplane: US Patent USD709430 S1. http://patents.google.com/patent/USD709430S1, last accessed Feb. 2016
29. Xisto CM, Páscoa JC, Leger JA, Masarati P, Quaranta G, Morandini M, Gagnon L, Wills D, Schwaiger M (2014) Numerical modelling of geometrical effects in the performance of a cycloidal rotor. In: 6th European Conference on Computational Fluid Dynamics, Barcelona, Spain
30. Yun CY, Park IK, Lee HY, Jung JS, H IS (2007) Design of a New Unmanned Aerial Vehicle Cyclocopter. Journal of the American Helicopter Society 52(1)

A Review of Shape Distortion Methods Available in the OpenFOAM® Framework for Automated Design Optimisation

Steven Daniels, Alma Rahat, Gavin Tabor, Jonathan Fieldsend and Richard Everson

Abstract Parametrisation of the geometry is one of the essential requirements in shape optimisation, and is a challenging subject when carrying out a automated procedure. It is critically important to maintain the consistency of the shape and grid quality between each evaluation, while providing flexibility for a wide range of shapes using the same parameterisation of the geometry. The sensitivity of the grid to the changes to the geometry must be at a minimum during this process. This contribution presents a review of the grid distortion and regeneration methods available within the OpenFOAM® framework which can be utilised for shape optimisation. The objective of this contribution is to compare the effectiveness of these methods in the automated procedure and to provide suggestions for improvements. Special attention is given to three major factors involving shape optimisation: automation of model abstraction, automation of grid deformation or regeneration and robustness.

1 Introduction

Optimisation of designs using Computational Fluid Dynamics (CFD) is frequently performed across many fields of research, such as the optimisation of an aircraft wing to reduce drag or an increase in the efficiency of a heat exchanger. General optimisation strategies involve modification of design variables with a view to improving the appropriate objective function(s). Often, the objective function(s) is (are) nonlinear and multi-modal, and hence polynomial time algorithms for solving such problems may not be available. In such cases, applying Machine Learning methods such as Evolutionary Algorithms (EAs—a class of stochastic global optimisation technique inspired by natural evolution) may bring to light good solutions within a practical time frame. The traditional CFD design optimisation process is often based on a 'trial-and-error' type of approach. Starting from an initial geometry, Computationally Aided Design changes are introduced manually based on results from a limited number of design evaluations and CFD analyses. The process is usually complex,

S. Daniels (✉) · A. Rahat · G. Tabor · J. Fieldsend · R. Everson
University of Exeter, Harrison Building, North Park Road, Exeter EX4 4QF, UK
e-mail: S.Daniels@exeter.ac.uk

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_28

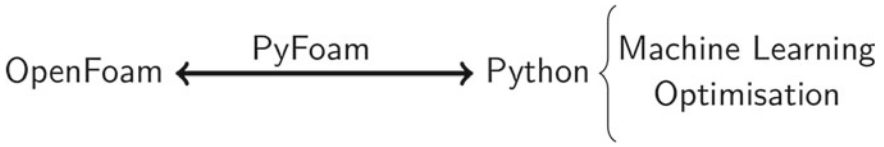


Fig. 1 Framework developed in the present work for combining OpenFOAM® with various Python-based Machine Learning libraries [1]

time-consuming and relies heavily on engineering experience, hence the overall design procedure is often inconsistent, i.e. different 'best' solutions are obtained from different designers.

Based on the limitations of optimisation by hand, using EAs and CFD simulations may be an attractive alternative. There have been other attempts to combine EAs and CFD simulation to optimise design (see for example, [2]). Although EAs do not guarantee the determination of the optimal solution, they may achieve good solutions consistently. In this present work, an automated framework combining Python's EA library DEAP with OpenFOAM® 2.3.1 was developed. The communication of the Python libraries with OpenFOAM® was achieved using PyFoam 0.6.5. PyFoam was used as an interface to control the OpenFOAM® case set-ups for each proposed solution from the EA code and to post-process the data generated after each CFD calculation. A summary of this framework can be seen in Fig. 1.

While the aim of this project is to focus on complex (potentially multi-objective) cases, for the purpose of this contribution, a single-objective optimisation of the PitzDaily tutorial in OpenFOAM® was used to demonstrate the performance of the proposed procedure. The chosen cost function for this case was to minimise the static pressure drop between the inflow and outflow boundary conditions, i.e.

$$f_1 = min(|p_i - p_o|), \tag{1}$$

where both pressures were obtained by averaging over the boundary condition faces. Figure 2 (top-to-bottom) shows the procedure in which the shape is altered after each evaluation. To change the geometry, subdivision curves [3] were generated to form a new lower wall. The grey squares indicate the coordinates for the search space (or bounding box), with the black lines showing the resulting boundary. The fixed points of the spline (indicated by orange squares) were placed across the bottom boundary condition ('lowerWall'). The squares in pink indicate the control points that may be altered by the EA, and thus change the curvature of the spline. After the new positions of the pink squares were proposed by the EA, a Stereolithography (.stl) file was generated and passed to OpenFOAM®. Using this file, the meshing utility snappyHexMesh was used to cut the domain away (and, if required, re-mesh the altered region). Subsequently, the case was run using the steady-state solver simpleFoam. After this, the cost function was obtained and the EA determined a new position for the coordinates of the pink squares.

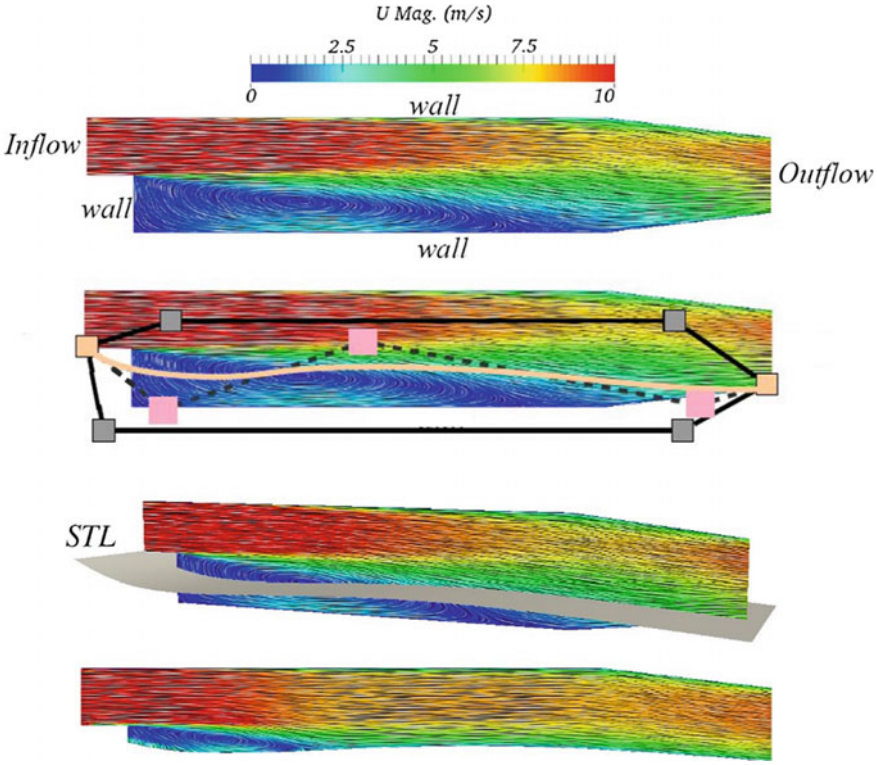


Fig. 2 Shape optimisation procedure of the PitzDaily tutorial case

A single-objective optimisation run of the PitzDaily tutorial was performed with a basic real-valued Genetic Algorithm [4]. For the CFD calculation, the residual tolerances for the SIMPLE algorithm for velocity and pressure were set to 10^{-5} and 10^{-6}, respectively. The pressure drop calculated from the original (base) case was measured as $|\Delta p| = 5.22\,\text{Pa}$. Figure 3 (left) shows the optimised shape of the PitzDaily geometry. The resulting pressure drop was calculated as $|\Delta p| = 4.7 \times 10^{-6}\,\text{Pa}$. The schematic diagrams above the contour diagram of velocity magnitude indicate the position of the control points, and the outer boundary region (in red) that they must not exceed. The GA maintains a population of solutions: Fig. 3 shows the maximum (worst), the average and the minimum (best) pressure drop across the population with each generation. The grey-dashed line indicates the pressure drops for the original PitzDaily geometry. It can be seen that the reduction of the best pressure difference between the inflow and outflow was reached after 15 generations, and an improvement of the pressure difference from the original case was achieved sooner than this.

For shape optimisation problems, the design variables usually involve parameters that modify the shape of a given initial geometry. For a certain objective function,

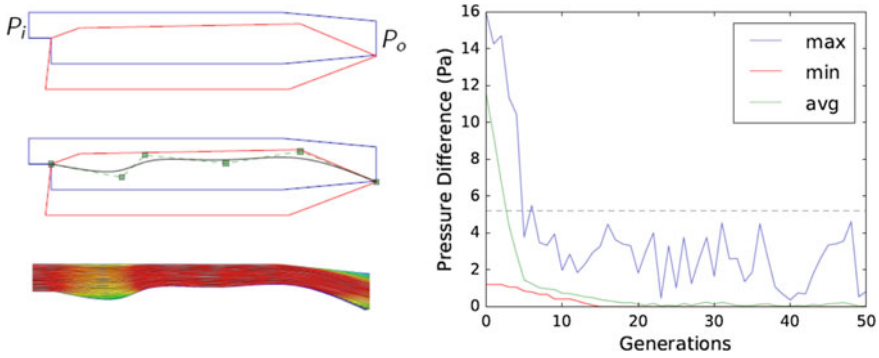


Fig. 3 Single-objective optimisation of the PitzDaily tutorial case in reducing the pressure drop

the design variables are used to define a deformed geometry at each optimisation cycle, which is used to compute the new flow field that is required to evaluate the objective function. In the traditional body-fitted approach, on structured or unstructured meshes, a change in the shape of the surface mesh requires a smooth transition in its deformation and the avoidance of large distortions and interpenetration of neighbouring elements in the CFD mesh. Various mesh deformation methods (e.g. spring stiffness, elastic analogy [5]) have been proposed to tackle this problem, but as the geometric complexity of optimisation problems increases, the robustness of this approach reduces; the resulting computational cost of the mesh deformation is not negligible. In addition, the linearisation of the mesh distortion scheme (i.e. mesh sensitivity) must be computed to take into account the effect of shape perturbations on the flow equations, with additional cost and complexity.

In general, mesh generation is commonly recognised as one of the main challenges in CFD. Mesh quality issues can significantly impact the accuracy of the eventual solution, even to the point at which the solver diverges and no solution is generated; they can also significantly affect the level of computational work (e.g. number of evaluations) necessary to reach the solution. Modern Finite Volume (FV) CFD codes tend to use arbitrary unstructured or polyhedral meshes, allowing for a wide variety of cell shapes to accommodate complex geometries. This also allows for a wide variety of mesh problems; non-orthogonality, face skewness etc., and whilst modern solution algorithms can typically correct for mild levels of mesh problems, this is at the cost of increased numerical error. Pathological levels of mesh problems can lead to algorithm divergence. The acceptable level of mesh quality also varies according to the details of the modelling being used; for example, the turbulence modelling in Large Eddy Simulation (LES) ties in very closely with aspects of the mesh, such as cell size, thus requiring much higher levels of mesh quality than for cases involving Reynolds-Averaged Navier–Stokes (RANS) [6]. Note that, our discussion revolves around issues relating to mesh generation for FV CFD, which is in our area of familiarity. Similar issues undoubtedly arise for Finite Element (FE) methods and

other applications of these techniques. The remainder of this contribution will review the basic methods available in OpenFOAM®, and their effect on the reliability of the solution with respect to the quality of the mesh.

2 Grid Deformation and Regeneration Techniques

During the design optimisation process, the design surfaces are perturbed after each evaluation. These perturbations must be transferred from the surface points to the grid in the surrounding flow field. To achieve this, the methods available can be placed into two categories: grid regeneration and grid deformation; the latter has previously been demonstrated in OpenFOAM® by [7]. The techniques reviewed below are applicable to both structured and unstructured grids.

2.1 snappyHexMesh

A typical example of an automated grid regenerator for complex geometries is snappyHexMesh. To use it, the user provides a Stereolithography (STL) file of the geometry and a base mesh (typically a simple hexahedral block mesh). This utility then operates a three-stage meshing process of castellation, snapping, and boundary layer refinement. In the first step (castellation), cells are identified that are intersected by edges of the surface geometry; these cells are then refined by repeated cell splitting, with maximum and minimum levels of refinement being a definable parameter, and further surface refinement also being controllable. After this refinement process, all cells that lie 'outside' the desired geometric domain are deleted from the mesh. In the second, snapping step, vertices on the edge of the domain are 'snapped' to the STL surface, using an iterative process of mesh movement, cell refinement and face merging, again controlled by user-defined parameters such as number of iterations and specific mesh quality constraints. In the final and optional step, cell layers can be added to the surface to move the mesh away from the boundary so as to specifically refine a boundary layer. The whole process is robust and automated, but is controlled by a large number of user-specified parameters provided in advance as an input file. As with any meshing process, the user typically has to experiment with different settings to optimise the mesh. Mesh quality may ultimately be judged by the success of the resulting CFD run, but as a proxy, various mesh quality indicators, such as skewness and non-orthogonality, can more easily be evaluated.

Figure 4 shows the application of the snappyHexMesh utility to regenerate the geometry and mesh of the PitzDaily case (described in Sect. 1) during one of the design evaluations. Applying the 'boundary-layer refinement' stage of this utility requires a substantial number of input parameters. Previous experimentation as to the most influential parameters on the mesh quality (see [8]) has reduced this to a set of 7, whose definitions are provided below:

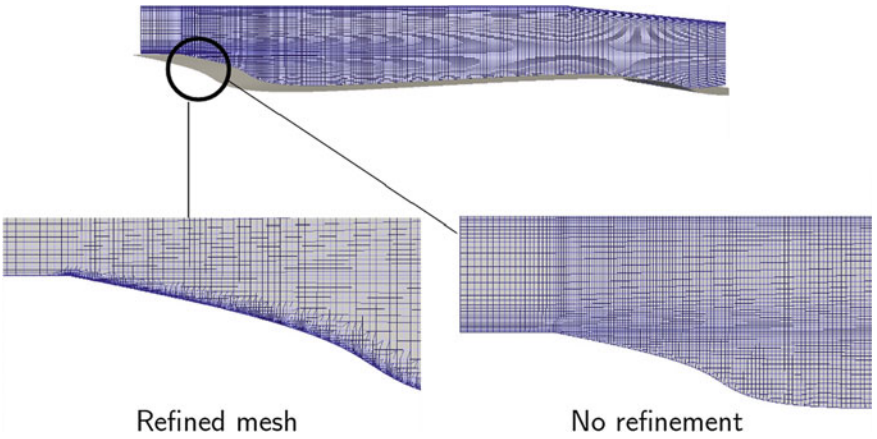


Fig. 4 The figure on the top shows the snappyHexMesh utility applied to the generated STL geometry after the castelling and snapping operation. The diagrams on the bottom show the grid distribution with and without the additional layering operation

- **resolveFeatureAngle**: Maximum level of refinement applied to cells that intersect with edges at angles exceeding this value.
- **nSmoothPatch**: Number of patch smoothing operations before a corresponding point is searched on the target surface. Smooth patches are more likely to be parallel to the target surface, making it more probable to find a matching point.
- **nRelaxIter**: Number of iterations to relax the mesh after moving points. When points are snapped to the target, the displacement propagates through the underlying layers of points that are not on the surface. By relaxing this propagation, a smoother displacement can be achieved.
- **nFeatureSnapIter**: The total number of iterations tried to snap points to the target. If insufficient quality is reached after nFeatureSnapIter iterations, the snapping is cancelled and the last state is recovered.
- **maxNonOrtho**: Non-orthogonality measures the angle between two faces of the same cell. In a grid with only rectangular cells, the value would be zero. Any deviation from this counts as non-orthogonal. High values mean there are very low angles that usually occur in a prism layer.
- **maxSkewness**: Skewness is the ratio between the largest and the smallest face angles in a cell. A value of 0 is the perfect cell and 1 is the worst. Within the OpenFOAM® dictionary, different quality constraints can be assigned to boundary cells and internal cells. In a simple geometry, the cells on the boundaries are more likely to be affected by skewness problems.
- **minVolRatio**: The ratio in cell volume between adjacent cells should not be too large—a large ratio leads to unacceptable interpolation errors.

Of course, these input parameters could also be subjected to optimisation with target measurements for the base geometry (e.g. [8]). It should be noted, however, that in

using the methodology of this contribution, snappyHexMesh can only remove sections from the base geometry—substantially reducing the exploration of the design space.

2.2 Grid Distortion Methods

The available techniques for modifying the grid can be separated into two groups: fixed group methods (such as the immersed boundary) and moving grid methods, with the Arbitrary Lagrangian–Eulerian (ALE) approach as a representative. In [9], the advantages of the ALE approach over the fixed-mesh alternatives is described. This analysis is based on the method's ability to maintain a high-quality grid near the moving body, resulting in a better representation of the boundary interface in this region. In the ALE approach, the grid is moved to allow for the distortion of the boundary's shape. This can be achieved through squeezing and stretching the surrounding cells and their associated vertices. For the FV method, the conservation equation of property, ϕ, over an arbitrary moving control volume, V_C, in integral form is:

$$\frac{d}{dt}\int_{V_C}\phi dV_C + \int_A dA\cdot(\vec{u}-\vec{u_b})\phi = \int_{V_C}\nabla\cdot(\Gamma\nabla\phi)dV_C, \tag{2}$$

where $\vec{u}$ is the velocity vector, A is the cell surface normal vector and $\vec{u_b}$ is the boundary velocity vector of the cell face. To govern the vertex motion, OpenFOAM® adopts a Laplacian smoothing scheme, described by

$$\nabla\cdot(\gamma\nabla u_p) = 0, \tag{3}$$

where u_p is the point velocity, which is imposed at each vertex of the control volume. The boundary velocity u_b is interpolated from u_p. The boundary conditions for Eq. 3 are enforced from the known boundary motion, e.g. a moving wall. The vertex position at the time level $n+1$ is calculated by using u_p

$$x^{n+1} = x^n + u_p\Delta t. \tag{4}$$

The variable γ prescribes the distribution of deforming cells around the moving body. Ideally, for the Laplacian approach, the cell distortion near the moving wall should be less perturbed by the motion of the body, while with increasing distance away from the wall, the cells should have greater freedom to deform. Under this concept, the quadratic diffusion model ($\gamma = 1/l^2$) has been shown to present a suitable distribution of cells around the body, with l being the distance from the moving wall. A comparison between the uniform ($\gamma = uniform$) and quadratic diffusion models on the grid skewness is demonstrated in [10, 11]. As the grid motion in the whole domain is governed by Eq. 3, an interface between the static and dynamic mesh regions is not

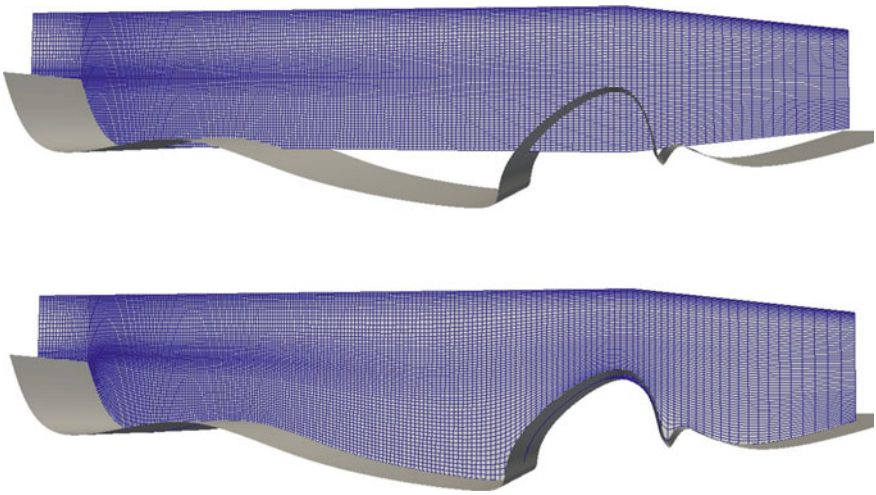


Fig. 5 The Laplace diffusion model ($\gamma = uniform$) applied to the generated STL surface. The 'lowerwall' boundary was chosen to morph into the STL surface

required. Figure 5 shows the application of the uniform diffusion model applied to the PitzDaily tutorial case. It can be seen in this figure that by stretching and squeezing the cells to fit the moving boundary, the skewness and orthogonality qualities of the mesh are compromised. To overcome this, further modifications to the solver can be achieved by including the mesh refinement utilities available in snappyHexMesh; this has shown promising results in previous related studies (e.g. [12, 13]).

2.3 Immersed Boundary Method (IBM)

Immersed boundary methods provide a promising alternative to the classical body-fitted discretisations. In the Immersed Boundary approach, the CFD grid does not conform to the geometry of the object, eliminating the problem of the mesh deformation. When combined with an efficient flow solver, this approach is particularly well-suited for the automated analysis of complex geometry problems. An additional advantage of non-boundary-conforming numerical methods is that there is no need to compute the internal mesh sensitivity with a substantial reduction in user coding and computation effort. In summary, the advantages of IBM over Body-Fitted Mesh methods include:

- a substantially simplified grid distribution around the complex geometry;
- flexibility with the inclusion of a body motion due to the use of stationary, non-deforming background grids;

while the limitations of the IBM include:

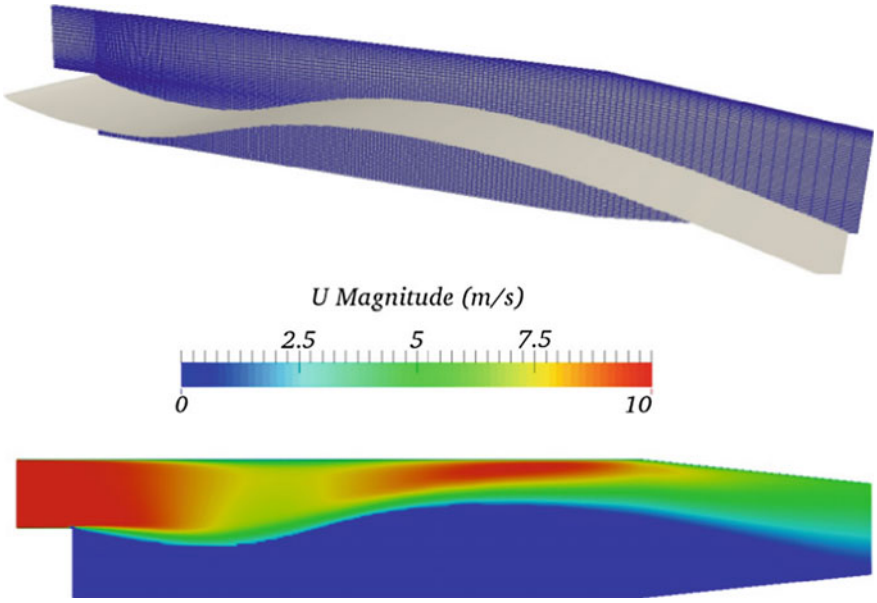


Fig. 6 The Immersed Boundary method applied to the PitzDaily tutorial with the additional STL surface as the immersed boundary

- the requirement of special techniques to simulate the boundary conditions for the immersed boundary;
- problems with grid resolution in the boundary layer region of the geometry;

with topics currently under investigation in the literature:

- IBM mimicking the equivalent body-fitted mesh solutions;
- automated mesh refinement around the Immersed Boundary surface.

Figure 6 demonstrates the application of IBM in the OpenFOAM® framework [14].[1] It can be seen in this figure that the Immersed Boundary has obstructed the cells below the spline from the inflow and outflow, in a similar manner as if that part of the domain had been removed.

[1]The Immersed Boundary Method is not available in OpenFOAM® 2.3.1, but is available in Foam-extend-3.2 and 4.

3 Conclusions

The grid distortion and generation techniques available in the OpenFOAM® framework appropriate for shape optimisation have been reviewed in this contribution. OpenFOAM® provides a convenient range of libraries for distorting the grid (using the Laplacian method) or grid regeneration (snappyHexMesh). The grid distortion technique is somewhat limited by the mesh quality after each design evaluation, but this can be overcome by including the mesh refinement libraries in snappyHexMesh. The grid regeneration technique is somewhat limited in regard to altering the geometry. A possible addition to both these techniques would be to apply grid sensitivity analysis (see [15])—defined as the partial derivative of the grid-point coordinates with respect to the design variable—within the OpenFOAM® framework. Furthermore, the recent development of the Immersed Boundary Method in OpenFOAM® provides a rather efficient approach to automated shape optimisation. However, like the remaining approaches, the generated solution is somewhat sensitive to the grid's resolution.

Acknowledgements This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant (reference number: EP/M017915/1) for the University of Exeter's College of Engineering, Mathematics, and Physical Sciences.

References

1. Daniels, S.J. Rahat, A.A.M. Everson, R.M. Tabor, G.R. and J.E. Fieldsend. A suite of computationally expensive shape optimisation problems using computational fluid dynamics. In 15th International Conference on Parallel Problem Solving from Nature, PPSN2018, 15th International Conference on Parallel Problem Solving from Nature, PPSN2018, (2018). https://doi.org/10.1007/978-3-319-99259-4_24
2. Caputo, A. Pelagagge, P. and Salini, P.: Heat exchanger design based on economic optimisation. Appl. Therm. Eng. **28**, 1151–1159 (2008)
3. Catmull, E. Clark, J.: Recursively generated B-spline surfaces on arbitrary topological meshes. Comput. Aided Des. **10**(6), 350–355 (1978)
4. Back, T. Fogel, D.B. and Michalewicz, Z.: Evolutionary computation 1: Basic algorithms and operators. Taylor and Francis, Milton Park, Abingdon-on-Thames, Oxfordshire United Kingdom (2000)
5. Schmidt, J. Stoevesandt, B.: Dynamic mesh optimization based on the spring analogy. First Symposium on OpenFOAM® in Wind Energy. **2**, (2014) https://doi.org/10.1051/itmconf/20140203001
6. McRae, D. and Laflin, K.: Dynamic grid adaption and grid quality In: Thompson, J. Soni, B. Weatherhill, N. (eds) Handbook of grid generation, chap 34. CRC Press, London. United Kingdom (1999)
7. Petropoulou, S.: Industrial optimisation solutions based on OpenFOAM® technology. V European Conference on Computational Fluid Dynamics (ECCOMAS CFD 2010), Lisbon, Portugal. 14-17 June (2010).
8. Fabritius, B. Tabor, G.: Improving the quality of finite volume meshes through genetic optimisation. Eng. Comput. **32**, 425–440 (2016). https://doi.org/10.1007/s00366-015-0423-0

9. Tezduyar, T. Sathe, S. Schwaab, M. Conklin, B.: Arterial fluid mechanics modeling with the stabilized space time fluid structure interaction technique. Int. J. Numer. Meth. Fluids. **57**, 601–629 (2008)
10. Jasak, H. Tukovic, Z.: Automatic mesh motion for the unstructured Finite Volume Method. Trans. of FAMENA **30**(2), 1–20 (2006)
11. Jasak, H.: Dynamic mesh handling in OpenFOAM®. 47th AIAA Aerospace Sciences Meeting. Orlando, Florida. January (2009)
12. Jarpner, C.: Projection of a mesh on a .stl surface. Chalmers University of Technology. (2011)
13. Rypi, D. Bittnar, Z.: Generation of computational surface meshes of STL models. J. Comp. Appl. Math. **192**, 148–151 (2006). https://doi.org/10.1016/j.cam.2005.04.054
14. Jasak, H. Rigler, D. Tukovic, Z. Design and implementation of Immersed Boundary Method with discrete forcing approach for boundary conditions. In: 11th. World Congress on Computational Mechanics - WCCM XI 5th. European Congress on Computational Mechanics - ECCM V 6th European Congress on Computational Fluid Dynamics - ECFD VI, Barcelona; Spain. (2014)
15. Samareh, J.A.: Geometry and grid/mesh generation issues for CFD and CSM shape optimisation. Opt. Eng. **6**, 21–32 (2005)

Simulating Polyurethane Foams Using the MoDeNa Multi-scale Simulation Framework

Henrik Rusche, Mohsen Karimi, Pavel Ferkl and Sigve Karolius

Abstract The MoDeNa project [20] aims at developing, demonstrating, and assessing an easy-to-use multi-scale software framework application under an open-source licensing scheme that delivers models with feasible computational loads for process and product design of complex materials. The concept of MoDeNa is an interconnected multi-scale software framework. As an application case, we consider polyurethane (PU) foams, which are excellent examples of a large turnover product produced in a variety of qualities of which the properties are the result of designing and controlling the material structure on different scales, from the molecule to the final product. Hence, various models working at individual scales will be linked together by this framework such as meso- and macro-scale models. OpenFOAM® is deployed on the macro-scale level. A new solver (`MODENAFoam`) is formulated and validated to demonstrate the interconnectivity of the scales using the MoDeNa framework. The efficiency of the multi-scale model is evaluated by comparing the numerical predictions of foam density and temperature evolutions with experimental measurements. Validation results showed the capability of the framework when it is assessed for simulation of a complex system such as polyurethane foam.

H. Rusche (✉)
Wikki Ltd., London, UK
e-mail: h.rusche@wikki.co.uk

M. Karimi
DISAT, Politecnico di Torino, Torino, Italy
e-mail: mohsen.karimi@polito.it

P. Ferkl
Department of Chemical Engineering, University of Chemistry and Technology, Prague, Czech Republic
e-mail: ferklp@vscht.cz

S. Karolius
Department of Chemical Engineering, Faculty of Natural Sciences and Technology, Norwegian University of Science and Technology, Trondheim, Norway
e-mail: sigve.karolius@ntnu.no

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_29

1 Introduction

Polyurethane foam is used to demonstrate and evaluate the predictive capability of a multi-scale framework, namely MoDeNa. MoDeNa enables coupling of modeling tools that works on different scales. Modeling and simulation of PU provide apriori information about the final product, and how it can be modified by chemical recipes and operational conditions [17, 23, 30]. Thus, it is necessary to develop a comprehensive, yet feasible modeling platform in which different physical phenomena described on multiple scales, from molecular to macro-scale, can be integrated.

Reviewing the literature reveals that currently there is no multi-scale platform for modeling PU foams. The available models concentrate on one aspect of PU simulation. For example, Baser and Khakhar focused on the macro-scale behavior of PU foams, and solved a set of ordinary differential equations (ODEs) to describe foam density and temperature [1, 2]. Computational fluid dynamics (CFD) has also been an attractive alternative because it allows the spatial and temporal variations of different foam properties to be locally investigated. The typical method is to add extra partial differential equations (PDEs) for macroscopic phenomena, e.g., the progress of the polymerization is modeled by adding two PDEs [3, 10, 25–27, 29]. Considering the lower scale tools, the growth of a single bubble has been studied by different groups. Harikrishnan et al. [11] used a simple mass transfer model for the bubble growth in PU foam. A similar approach (bubble-shell model) was also adopted in which the mass and momentum balance were solved for a spherical bubble in a liquid foam to evaluate the radius of the bubble [6, 16].

Recently, Karimi et al. [14] presented a macro-scale model based on coupling VOF with a population balance equation (PBE). This enables the modeling approach to go one step further and predict the cell or bubble size distribution (BSD). However, they applied simplified models for the bubble growth based on diffusion. This has been addressed by Ferkl et al. [7] using a multi-scale modeling prototype coupling the macro- and bubble-scale models. The objective of this work is to follow up the previous attempt and develop a three-dimensional multi-scale platform for simulation of PU foams. This includes a macro-scale OpenFOAM® solver in which the interconnectivity of scales is realized through MoDeNa. The solver is coupled with a detailed bubble-scale model providing the growth rate due to the presence of different gases. As the CFD code is supplemented by the solution of a PBE, the growth rate is needed to simulate the growth of bubbles via PBE. The growth rate itself depends on the dynamic characteristics of the foam during the foaming process and is too expensive to calculate on a cell-by-cell basis. Therefore, the MoDeNa is used to encapsulate the growth rate in a surrogate model with parameters that are dynamically fitted to detailed simulations. The information is then applied in the `MODENAFoam` solver to realistically simulate the state of gas bubbles within the foam at a much lower computational cost.

2 Governing Equations

2.1 Reaction Kinetics

PU foams are produced using a process called reaction foaming, during which polymerization occurs simultaneously with the expansion. The complex polymerization scheme can be simplified as two global reactions [1, 2]. The reaction source terms can be written in terms of polyol and water conversions as shown below.

$$S_{\mathrm{OH}} = A_{\mathrm{OH}} \exp\left(-\frac{E_{\mathrm{OH}}}{R_{\mathrm{g}} T}\right)(1 - X_{\mathrm{OH}})(c_{\mathrm{NCO},0} - 2c_{\mathrm{W},0} X_{\mathrm{W}} - c_{\mathrm{OH},0} X_{\mathrm{OH}}), \quad (1)$$

$$S_{\mathrm{W}} = A_{\mathrm{W}} \exp\left(-\frac{E_{\mathrm{W}}}{R_{\mathrm{g}} T}\right)(1 - X_{\mathrm{W}}), \quad (2)$$

where A_{OH} and A_{W} are the pre-exponential factors, E_{OH} and E_{W} are the activation energies, R_{g} is the gas constant, T is the temperature and $c_{\mathrm{NCO},0}$, $c_{\mathrm{OH},0}$, and $c_{\mathrm{W},0}$ are the initial concentrations of isocyanate, polyol, and water, respectively.

The temperature source terms associated with these reactions can be written as

$$S_T = \frac{(-\Delta H_{\mathrm{OH}})\, c_{\mathrm{OH},0}}{\rho_{\mathrm{PU}} c_{p,\mathrm{f}}} \frac{\mathrm{D}X_{\mathrm{OH}}}{\mathrm{D}t} + \frac{(-\Delta H_{\mathrm{W}})\, c_{\mathrm{W},0}}{\rho_{\mathrm{PU}} c_{p,\mathrm{f}}} \frac{\mathrm{D}X_{\mathrm{W}}}{\mathrm{D}t} + \sum_{i}^{N} \frac{\left(-\Delta H_{\mathrm{v},i}\right)}{c_{p,\mathrm{f}}} \frac{\mathrm{D}w_i}{\mathrm{D}t}, \quad (3)$$

where t is the time, ΔH_{OH} and ΔH_{W} are the reaction enthalpies of the gelling and blowing reactions, ρ_{PU} is the density of the liquid mixture undergoing polymerization, $c_{\mathrm{p,f}}$ is the thermal capacity of the foam, N is the number of blowing agents, $\Delta H_{\mathrm{v},i}$ is the heat of evaporation for the ith blowing agent, and w_i is the mass fraction for the i-th blowing agent in the gas phase with respect to the foam.

2.2 Bubble-Scale Model

In PU foaming, a large number of air bubbles is entrained into the reaction mixture during the mixing of reactants. Thus, the system never reaches sufficient supersaturation, which would lead to nucleation. Instead, when the blowing agent is supersaturated in the reaction mixture, it diffuses into the bubbles.

The mathematical description of this process is based on idealized geometry. The bubbles are assumed to be spherical and surrounded by an effective shell of the reaction mixture, which accounts for the fact that there is a limited amount of blowing agent available for each bubble. In this case, the growth of a bubble will result in

a purely radial velocity field, and thus the momentum balance can be simplified as in [6]:

$$\sum_{i=1}^{N} p_i + p_{\text{air}} - p_{\text{PU}} = \rho_{\text{PU}} \left[R \frac{\mathrm{d}^2 R}{\mathrm{d}t^2} + \frac{3}{2} \left(\frac{\mathrm{d}R}{\mathrm{d}t} \right)^2 \right] + \frac{2\gamma}{R} + \frac{4\mu_{\text{PU}}}{R} \frac{\mathrm{d}R}{\mathrm{d}t}, \quad (4)$$

where p_i is the partial pressure of the ith blowing agent in the bubble, p_{PU} is the pressure in the reaction mixture, R is the actual bubble radius, γ is the surface tension and μ_{PU} is the viscosity of the reaction mixture. The terms on the right-hand side represent inertial, surface tension and viscous forces, respectively.

The mass balance for the ith blowing agent in the bubble can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{p_i R^3}{R_{\text{g}} T} \right) = 3 D_i R^2 \left. \frac{\partial c_i}{\partial r} \right|_{r=R}, \quad (5)$$

where D_i is the diffusion coefficient of the blowing agent in the reaction mixture, c_i is the molar concentration of the blowing agent in the reaction mixture and r is the spatial coordinate. It is assumed that the resistance to mass transfer is entirely on the side of the reaction mixture.

Finally, the mass balance for the ith blowing agent in the reaction mixture can be written as

$$\frac{\partial c_i}{\partial t} + \frac{R^2}{r^2} \frac{\mathrm{d}R}{\mathrm{d}t} \frac{\partial c_i}{\partial r} = \frac{D_i}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial c_i}{\partial r} \right) + r_i, \quad (6)$$

where r_i is the reaction source term, which can be expressed as

$$r_i = \begin{cases} c_{\text{W},0} S_W & \text{if } i = \text{CO}_2 \\ 0 & \text{if } i \neq \text{CO}_2, \end{cases} \quad (7)$$

We assume that the concentration at the bubble–shell interface is given by the Henry's law and that the blowing agent is not transported across the outer shell boundary. Thus, the boundary conditions for Eq. (6) are set according to

$$c_i|_{r=R} = H_i p_i, \quad (8)$$

$$\left. \frac{\partial c_i}{\partial r} \right|_{r=S} = 0, \quad (9)$$

where H_i is the Henry constant and S is the outer radius of the shell. The size of the shell S is a function of time, but it can be directly calculated from the bubble radius and initial bubble and shell sizes assuming that the density of the reaction mixture is constant [7].

The system of differential Eqs. (4–6) is solved together with the reaction kinetics (see Sect. 2.1) under the assumption that the system is adiabatic. We are most inter-

ested in the continuous bubble growth rate as the quantity, which can be used in the macro-scale model. In this work, we quantify the contributions to the bubble growth rate due to each blowing agent as the molar flow rate of the blowing agent into the bubble:

$$\dot{n}_i = 4\pi D_i R^2 \left.\frac{\partial c_i}{\partial r}\right|_{r=R}. \tag{10}$$

2.3 Modeling the Macroscopic Scale

The macro-scale `MODENAFoam` is based on a VOF solver for two immiscible fluids, which is modified to address modeling concerns for polyurethane foam. One of the main features of the code is the implementation of a PBE. The general form of this equation considers that bubbles can grow and coalesce during foam expansion:

$$\frac{\partial n(\mathrm{v})}{\partial t} + \nabla \cdot (\mathbf{U}_f n(v)) + \frac{\partial}{\partial \mathrm{v}}[G(\mathrm{v})n(\mathrm{v})] = \frac{1}{2}\int_0^{\mathrm{v}} \beta(\mathrm{v}', \mathrm{v}\text{-}\mathrm{v}')n(\mathrm{v}')n(\mathrm{v}\text{-}\mathrm{v}')\,\mathrm{d}\mathrm{v}' - \int_0^{\infty} \beta(\mathrm{v}, \mathrm{v}')n(\mathrm{v})n(\mathrm{v}')\,\mathrm{d}\mathrm{v}'. \tag{11}$$

Where $\mathbf{U}_f$ is the foam velocity and the internal coordinate is the volume of bubble, v. The term $n(\mathrm{v})$ is the bubble size distribution indicating the number of bubbles per unit volume of the liquid mixture. Furthermore, the frequency of coalescence between two bubbles of volume v and v′ is defined with the coalescence kernel $\beta(\mathrm{v}, \mathrm{v}')$ and $G(\mathrm{v}) = \mathrm{d}\mathrm{v}/\mathrm{d}t$ is the overall growth rate. The population balance equation [i.e., Eq. (11)] is solved by transforming it into a set of partial differential equations for the moments of BSD using the generic definition of moments:

$$m_k(t) = \int_0^{\infty} n(\mathrm{v})\mathrm{v}^k\,\mathrm{d}\mathrm{v}\,. \tag{12}$$

This definition assigns physical meaning to each moment. For example, $m_1(t)$ is the total volume of bubbles per unit volume of the liquid mixture. Another benefit of using the definition of generic moments is the efficiency of the computation, as it allows to follow the moments of the BSD using only 4–6 moments [19]. The evolution of the mean bubble diameter can be monitored knowing the first two moments as follows:

$$d_\mathrm{b}(t) = \left(\frac{m_1(t)}{m_0(t)}\frac{6}{\pi}\right)^{1/3}. \tag{13}$$

The transport of moments within the PU foam phase is evaluated as follows:

$$\frac{\partial m_k}{\partial t} + \left(\mathbf{U}_f - \alpha_\mathrm{a}\mathbf{U}_r\right) \cdot \nabla m_k = k \sum_{\mathrm{i}=1}^{N} \overline{G}_k^i + \overline{S}_k. \tag{14}$$

In this work, $k \in [0, 3]$, and N represents the number of blowing agents. The volume fraction of surrounding air is represented as α_a. The source term due to different blowing agents is indicated by $\overline{G}_k^i$, whereas $\overline{S}_k$ is the source term for the coalescence of bubbles. More details on how to treat the source terms are reported elsewhere [14, 18].

As the solution of the PBE provides the total bubble volume per unit volume of the liquid of mixture, i.e., $m_1(t)$, the evolution of the foam density can be expressed as

$$\rho_\mathrm{f} = \rho_b \frac{m_1(t)}{1 + m_1(t)} + \rho_{PU} \frac{1}{1 + m_1(t)}. \tag{15}$$

In Eq. (15), ρ_b and ρ_PU are the densities of the gas within the bubbles and of the liquid mixture, respectively.

The evolution of the temperature of the foam phase is calculated as follows:

$$\frac{\partial T}{\partial \mathrm{t}} + \nabla \cdot (\mathbf{U}T) + (\mathbf{U} - \mathbf{U}_r) \cdot \nabla T - \nabla^2 (\bar{\alpha} T) = \alpha_\mathrm{f} S_T, \tag{16}$$

where $\bar{\alpha}$ is the thermal diffusivity of the PU foam, and S_T is defined in Eq. (3). The conversions of water (X_W) and polyol (X_OH) are accounted for by adding two extra PDEs:

$$\frac{\partial X_\mathrm{W}}{\partial t} + (\mathbf{U} - \alpha_\mathrm{a}\mathbf{U}_\mathrm{r}) \cdot \nabla X_\mathrm{W} = S_\mathrm{W}, \tag{17}$$

$$\frac{\partial X_\mathrm{OH}}{\partial t} + (\mathbf{U} - \alpha_\mathrm{a}\mathbf{U}_\mathrm{r}) \cdot \nabla X_\mathrm{OH} = S_\mathrm{OH}, \tag{18}$$

Additionally, accounting for the blowing agents within the liquid mixture the mass balance for the ith blowing agent in the mixture is written as

$$\frac{\partial w_\mathrm{i}}{\partial \mathrm{t}} + (\mathbf{U} - \alpha_\mathrm{a}\mathbf{U}_r) \cdot \nabla w_\mathrm{i} = r_\mathrm{i} \frac{M_\mathrm{i}}{\rho_\mathrm{PU}} + \overline{G}_1^i \frac{P}{RT} \frac{M_\mathrm{i}}{\rho_\mathrm{PU}}, \tag{19}$$

where w_i is the mass fraction of the ith blowing agent and r_i is defined as

$$r_\mathrm{i} = \begin{cases} C_W^0 \frac{\mathrm{DX_W}}{\mathrm{Dt}}, & \text{if i} = \mathrm{CO_2}, \\ 0, & \text{if i} \neq \mathrm{CO_2}. \end{cases} \tag{20}$$

The molecular mass of the ith blowing agent is M_i, and the symbol $\overline{G}_1^i$ represents the moment of order one of the growth rate due to the ith blowing agent.

3 The MoDeNa Software Framework

Several strategies have been developed for building bridges across the scale-separation, thus coupling the scale-specific models, such as the heterogeneous multi-scale method (HMM) described in [28] and the equation-free approach by Kevrekidis et al. [15]. These approaches are opposite techniques for scale-bridging because HMM is a top-down approach and the equation-free method is bottom-up. The design of the MoDeNa software framework is a top-down and it aims at eliminating approximations, typically represented as constants or constitutive relationships, with surrogate models approximating the detailed models.

3.1 Design Philosophy

The philosophy underpinning the MoDeNa software framework is to ensure loose coupling between applications representing detailed models. The coupling and communication across scales is handled through recipes and adapters. Recipes perform simulations by executing applications (in-house codes or external software packages such as OpenFOAM®, Materials Studio, PC-Saft or in-house software) for a given set of inputs. Adapters handle the communication with the MoDeNa software framework. Both recipes and adapters are application-specific. Adapters exist as outgoing and incoming adapters. Outgoing adapters are relatively straightforward in that they perform a mapping operation (such as averaging) and communicate the results. The averaging process may have to be started and performed within the application (e.g., for time averaging). However, the results can usually be submitted in a separate process after the simulation is finished. Incoming adapters usually require that the surrogate model us embedded within the application and is, therefore, more complex.

3.2 Scale Coupling

When considering multi-scale modeling from the top-down perspective the scale coupling happens in the set of parameters ($\mathbb{D}$) of the individual scale-specific models ($\mathcal{M}$). Traditionally, it is assumed that the parameters are valid for the entire range of model inputs, $\mathbf{u} \in \mathbf{U}_{max}$, resulting in the generic representation in Eq. (21).

$$\mathbf{y}_i = \mathcal{M}\left(\mathbf{u}_i \; ; \; \mathbb{D}\right) \qquad ; \; \mathbf{u}_i \in \mathbf{U}_{max} \; . \tag{21}$$

The accuracy of models such as the Navier–Stokes equations has proven to be sufficient for an astonishing diversity of engineering applications. However, when the assumption no longer holds it may be necessary to replace the parameters with a set of models, $\left\{\mathcal{M}_j^{i-1}\right\}$, which provides the quantity using a model that describes the

physics from first principles, as represented in Eq. (22).

$$\mathcal{M} := \mathcal{M}\left(\mathbf{u}_i\ ;\ \left\{\mathcal{M}_j^{i-1}\right\}\right) \qquad ;\mathbf{u}_i \in \mathbf{U}_{max} \tag{22}$$

When coupling a large number of models using this strategy the practical issue of computational cost becomes important. The approach taken in the MoDeNa project was to employ surrogate models, a simplified model whose purpose is to approximate the input–output behavior of a more detailed model, in-place of scale-specific first-principle models. The parameters ($\boldsymbol{\theta}$) of the surrogate model are validated in a domain ($\mathbf{U}$) using simulation results of the detailed model it approximates.

$$\hat{\mathbf{y}}_i = \hat{\mathcal{M}}\left(\mathbf{u}_i\ ;\ \boldsymbol{\theta}\right) \approx \mathbf{y}_i \qquad ;\mathbf{u}_i \subset \mathbf{U} \in \mathbf{U}_{max}. \tag{23}$$

Using the surrogate model in Eq. (22) the multi-scale detailed model becomes

$$\mathcal{M} := \mathcal{M}\left(\mathbf{u}_i\ ;\ \left\{\hat{\mathcal{M}}_j^{i-1}\right\}\right) \qquad ;\mathbf{u}_i \subset \mathbf{U} \in \mathbf{U}_{max}. \tag{24}$$

The parameter fitting and validation of the individual surrogate models used in a scale-specific model in the form shown in Eq. (24) is done automatically by MoDeNa using the model-based design of experiments framework illustrated in Fig. 1 and outlined in [8]. The execution time of the surrogate is usually negligible compared to that of the detailed model.

3.3 Software Components

The role of the software framework in the multi-scale application is to orchestrate the overall simulation and facilitate scale coupling. It consists of an orchestrator, a database and interface library. The orchestrator is based on FireWorks [12, 13] and constitutes the backbone of the software in that it schedules simulations as well as design of experiments as well as parameter estimation operations which make up the workflow of the overall simulation. It is very much like a dynamic workflow engine, in which the different applications are "orchestrated" to obtain information, analyse and pass it to the other operations. The NoSQL database MongoDB [21] is used to store the state of the workflow as well as the surrogate models together with associated data such as model parameters, data used for parameter estimation, and meta-data.

The interface library consists of two parts. A high-level Python module providing database access and capabilities for performing design of experiments and regression analysis by building on MongoEngine [22] and R [9, 24], respectively. The second part is a low-level library providing unified access to the surrogate models. This component is written in C to ensure interoperability across platforms and target applications while providing the computationally efficient model execution required

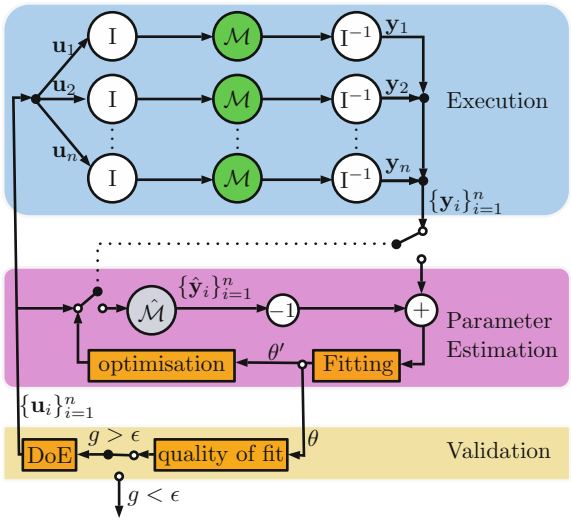


Fig. 1 Workflow of the model-based design of experiments procedure. The design of experiments (DoE) procedure generates a set of inputs $\{\mathbf{u}_i\}_{i=1}^n$, and a detailed simulation is performed using the detailed model $\mathcal{M}$, producing a set of outputs $\{\mathbf{y}_i\}_{i=1}^n$. Because the inputs are generated based on the input space of the surrogate model, the framework includes scale interface models for homogenization I and its inverse, lifting I^{-1}, respectively, transforming between the inputs and outputs of the detailed model and the surrogate model. The parameters of the surrogate model $\boldsymbol{\theta}$ are obtained and validated using a metric (g) against a criteria (ϵ)

by the applications. The library is loaded as a shared library by the macroscopic-scale applications or as a native Python extension by the high-level Python module ensuring that all components instantiate identical model implementations. Complex operations such as database access are referred back to the high-level Python module using call-back mechanisms.

3.4 Coupling of Macro- and Bubble-Scale Models

The MoDeNa framework handles the communication between the `MODENAFoam` solver and embedded models. As an example, the design, implementation and embedding of the bubble growth model through the use of a surrogate model is explained in detail. Other models follow the same principles. A suitable surrogate model for bubble growth is based on the concentration difference of blowing agent between liquid and gas phase:

$$\dot{n}_i = 4\pi R^2 \alpha R^\beta \left(\frac{w_i \rho_{\mathrm{PU}}}{M_i} - H_i(T) p_i, \right), \tag{25}$$

where α and β are the fitting parameters. The surrogate model provides bubble growth rate in terms of state variables R, w_i, p_i and T (solubility is generally a function of temperature).

Since the bubble growth model itself is a transient simulation, it cannot be directly used to determine bubble growth rate for desired state conditions. Instead, the detailed model is used to simulate the foaming from the same initial conditions as the macro-scale tool and both the state variables and the growth rate are saved at fixed intervals. These values are then used to determine the fitting parameters α and β. Afterward, the surrogate model can be called from the macro-scale tool.

4 MoDeNa as a Functional Piece in Applications

In order to understand the interaction between the MoDeNa software framework and the OpenFOAM® application it is necessary to consider the definition of surrogate models and how they are embedded into applications, as well as the role of the MoDeNa framework in the overall simulation.

4.1 Defining Surrogate Models

A core design-principle in the MoDeNa software framework is that the surrogate models are self-contained. Consequently, the definition must provide sufficient information to represent the generic function representation shown in Eq. (23), as well as identifying the model it approximates. However, the MoDeNa framework also demands that the author of the surrogate model provides recipes for how to read/write input files for the detailed model application as well as specifying the strategies that MoDeNa should invoke in order to resolve run-time exceptions related to the model. The figure below illustrates the information that is provided in order to define the model interface and properties of the surrogate model. The definition of the models are implemented in the Python programming language using predefined templates and software infrastructure provided by the high-level application program interface. Consequently, every scale-specific detailed model in the multi-scale application becomes a Python module; thus aiding in providing structure to the overall project which may involve a large number of models.

The purpose of the model-definition is to facilitate the development of models in relative isolation, such that authors of scale-specific models only need to be concerned with the information that should be passed between the models when MoDeNa couples them together. The benefit of this approach becomes apparent when considering how to embed the MoDeNa models inside an OpenFOAM® application.

4.2 Embedding Surrogate Models into OpenFOAM®

The footprint of the MoDeNa framework inside the OpenFOAM® applications is low, and the self-sufficiency of the surrogate models eliminates the need to link the OpenFOAM® application with libraries other than MoDeNa. However, in order to use surrogate models defined as modules in the MoDeNa software framework it is necessary to implement adaptors, i.e., code fragments calling the MoDeNa low-level library, inside the OpenFOAM® application.

The major difference between the MoDeNa coupling and a traditional "exhaustive" approach, where OpenFOAM® writes input files, executes the external application and reads the output is illustrates in Fig. 2. The low-level MoDeNa library is developed in C, but the surrogate models are still implemented as abstract data types that hide unnecessary implementation details from the user. However, since Fig. 2 clearly illustrates that the workflow for using the surrogate models is an excellent candidate for object-orientation the MoDeNa framework also provides a C++ class for easy integration into OpenFOAM® (Fig. 3).

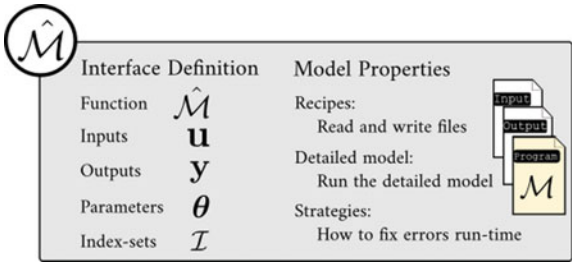


Fig. 2 Illustration of the data that is provided in the definition of a MoDeNa surrogate model. All information is provided as Python dictionaries and the model properties are implemented using templates provided by the framework. This means that objects can be de-serialized directly from a database of a flat file

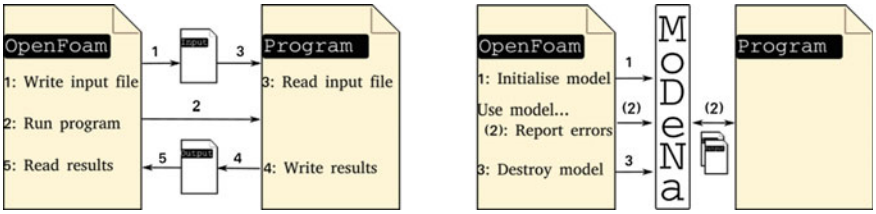


Fig. 3 Illustration of a "exhaustive" (left) and MoDeNa-style (right) coupling between two applications, or detailed models. The MoDeNa approach can be thought of as a combination between a library API and the "exhaustive" input/output file approach. The benefit is that the API will look the same for all models, and there is no intrusion of code specific to other applications

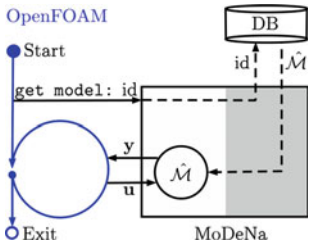


Fig. 4 Illustration of the workflow of an OpenFOAM® application (blue) in which calls to the MoDeNa software framework have been embedded (solid black lines). The workflow is ideal in the sense that error checks and exceptions related to the MoDeNa surrogate model have been omitted. The low-level C-layer of the MoDeNa framework (white) encapsulates the surrogate model, which is instantiated from the database by the high-level Python interface (gray)

4.3 Overall Simulation Workflow

The role of the MoDeNa software framework in the OpenFOAM® application is to ensure that the surrogate models are ready to be used. This is largely handled by the Python layer of the software, which is where the computational workflow and database management take place.

The software framework does not communicate with the database during the OpenFOAM® simulation assuming the execution of the surrogate model produces no exceptions. In this case, the database will only be used to instantiate the models at the beginning of the simulation. The ideal illustration is not accurate when the workflow is changed dynamically to accommodate the parameter-estimation procedure from Fig. 1. However, the software framework updates the state of the model in the database after each parameter-estimation run, which means that subsequent OpenFOAM® simulations will not be interrupted (Fig. 4).

5 Physical Properties and Operating Conditions

The viscosity of the liquid reaction mixture is modeled by Castro–Macosko model:

$$\mu_{\mathrm{PU}} = A \exp\left(\frac{E}{R_{\mathrm{g}} T}\right)\left(\frac{X_{\mathrm{OH,g}}}{X_{\mathrm{OH,g}} - X_{\mathrm{OH}}}\right)^{B+C X_{\mathrm{OH}}}, \tag{26}$$

where $A = 4.1 \times 10^{-8}$ Pas, $E = 38.3 \times 10^{3}$ Jmol^{-1}, $B = 4.0$, $C = -2.0$ are constants determined from experiments [5]. $X_{\mathrm{OH,g}} = 0.5$ is the conversion of polyols at the gel point.

Table 1 Kinetic parameters, operating conditions and material properties

Property	Value
p_{amb} (Pa)	1.0×10^5
ρ_{rm} (kgm^{-3})	1100
$c_{p,pol}$ (J kg^{-1}K^{-1})	1800
c_{p,CO_2} (J kg^{-1}K^{-1})	870
M_{CO_2} (kg mol^{-1})	0.044
D_{CO_2} (m^{2}s^{-1})	4.4×10^{-10}
H_{CO_2}(mol m^{-3}Pa^{-1})	1.1×10^{-4}
γ (Nm^{-1})	0.025

Property	Value
A_{OH} (m^3 mol^{-1}s^{-1})	1.735
E_{OH} (J mol^{-1})	4.04×10^4
$-\Delta H_{OH}$ (J mol^{-1})	7.07×10^4
A_W (m^3 mol^{-1} s^{-1})	1.39×10^3
E_W (J mol^{-1})	3.27×10^4
$-\Delta H_W$ (J mol^{-1})	8.60×10^4
$c_{OH,0}$ (mol m^{-3})	3765
$c_{NCO,0}$ (mol m^{-3})	3765
$c_{W,0}$ (mol m^{-3})	0
T_0 (K)	300
$n_{b,0}$ (m^{-3})	1×10^{12}
R_0 (m)	1×10^{-5}

A non-Newtonian model is instead applied for the calculation of foam apparent viscocity based on the Bird–Carreau theory [4]:

$$\mu_f\left(T, X_{OH}, \dot{\gamma}\right) = A_{OH} \exp\left(\frac{E_{OH}}{R_g T}\right) \times \left(\mu_\infty + \left(\mu_0 - \mu_\infty\right)\left(1 + \left(\dot{\gamma}\bar{\Lambda}\right)^\zeta\right)^{\frac{n-1}{\zeta}}\right) \tag{27}$$

In Eq. 27, μ_0 and μ_∞ are the values of foam viscosity under the minimum and maximum shear rates. The constants utilized in this work are: $\bar{\Lambda} = 11.35$, $\zeta = 2.0$, and $n = 0.2$. The rest of the physical properties and operating conditions are summarized in Table 1.

6 Results and Discussion

In this section, two cases are used to illustrate the efficiency of the coupling strategy in the MoDeNa framework. We first present the comparison between the detailed and surrogate model for the bubble growth (see Fig. 5). This is important in the coupling, as the purpose of the surrogate model is to provide a fast and viable mean value for estimating the growth rate of the bubbles. It can be seen that the growth rates computed by the surrogate model are reasonably accurate when compared to the detailed model. Second, the numerical predictions of the foam properties are validated with experimental measurements. This is to confirm how close the framework outputs are to the reality. Additional validation studies were reported in [7]. The foam properties predicted by `MODENAFoam` are shown in Fig. 6. The recipe,

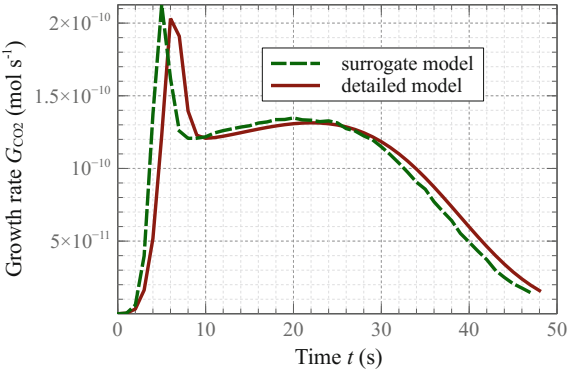


Fig. 5 The bubble growth rate as calculated by the detailed model (see Sect. 2.2) and the surrogate model Eq. 25

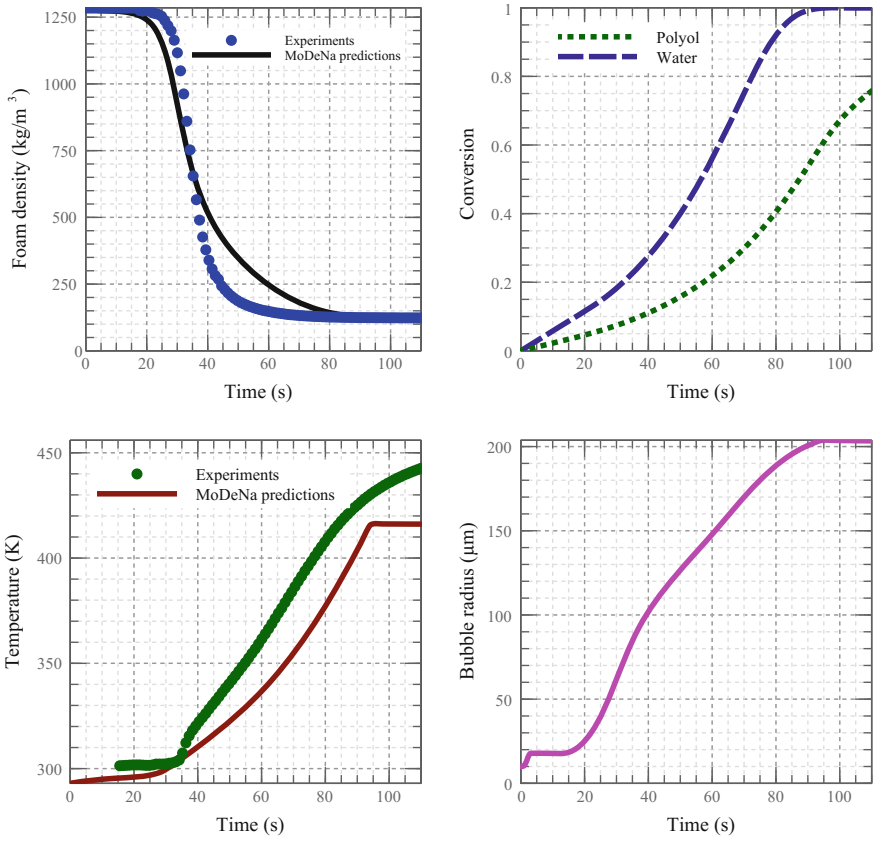


Fig. 6 `MODENAFoam` predictions of foam density (top-left), conversion of components (top-right), temperature profile (bottom-left), and bubble radius (bottom-right) as a function of time

and by extension the simulation case, uses a chemical blowing agent, water, without any physical blowing agent. The measurements are done for a classical "beaker test", which was simulated as a two-dimensional geometry with 10% liquid mixture filling at the beginning. Generally, the agreement between the experimental data and the numerical predictions of the foam density and temperature is reasonably good. However, the density predictions show better agreement compared to the temperature. The inconsistency between the predicted and measured temperatures can mainly be attributed to two factors: (1) that the heat capacity for the liquid mixture was assumed to be constant and (2) the lack of data for the enthalpies of the chemical reactions.

The conversion of reactants as well as the evolution of the bubble radius is also shown in order to demonstrate that detailed information can be extracted from the results. It is interesting to notice that the concentration of blowing agent within the liquid mixture requires a few seconds at the commence of foaming to reach to the equilibrium value. This physical phenomenon can be seen as the flat lines on the plots. This implies that during the first 20 s of the foaming process the concentration of CO_2 in the liquid mixture increases to the equilibrium value before diffusing into the bubbles; thereby growing the bubbles and decreasing the density of the foam. The evolution of bubble size also confirms this phenomenon.

7 Conclusions

The MoDeNa software framework is introduced in this work as an open-source library for multi-scale modeling. The benefit of utilizing MoDeNa is demonstrated when it is applied for the simulation of PU foams. As an example, a bubble-scale model for the growth of gas bubbles is linked to a macro-scale CFD tool. The lower scale model provides growth rates due to the presence of different blowing agents for the macro-scale tool. The CFD code is augmented with a population balance equation that uses the results from the bubble growth model. Furthermore, the multi-scale model is validated for the prediction of PU foam properties and the numerical results are compared with experimental data. The observed agreement assures that the framework handles the transfer of information between different models and that each scale-specific model is accurate in their predictions of meso- and macro-scale properties of the PU foam. This work will continue to incorporate more modeling tools (e.g., a detailed kinetic model) for the simulation of PU foams.

References

1. Baser, S. A. and Khakhar, D. V. (1994a). Modeling of the dynamics of r-11 blown polyurethane foam formation. *Polymer Engineering & Science*, 34(8):632–641.
2. Baser, S. A. and Khakhar, D. V. (1994b). Modeling of the dynamics of r-11 blown polyurethane foam formation. *Polymer Engineering & Science*, 34(8):642–649.

3. Bikard, J., Bruchon, J., Coupez, T., and Vergnes, B. (2005). Numerical prediction of the foam structure of polymeric materials by direct 3D simulation of their expansion by chemical reaction based on a multidomain method. *Journal of Materials Science*, 40(22):5875–5881.
4. Byron Bird, R. and Carreau, P. J. (1968). A nonlinear viscoelastic model for polymer solutions and melts—I. *Chemical Engineering Science*, 23(5):427–434.
5. Castro, J. M. and Macosko, C. W. (1982). Studies of mold filling and curing in the reaction injection molding process. *AIChE Journal*, 28(2):250–260.
6. Feng, J. J. and Bertelo, C. A. (2004). Prediction of bubble growth and size distribution in polymer foaming based on a new heterogeneous nucleation model. *Journal of Rheology*, 48(2):439.
7. Ferkl, P., Karimi, M., Marchisio, D., and Kosek, J. (2016). Multi-scale modelling of expanding polyurethane foams: Coupling macro- and bubble-scales. *Chemical Engineering Science*, 148:55–64.
8. Franceschini, G. and Macchietto, S. (2008). Model-based design of experiments for parameter precision: State of the art. *Chemical Engineering Science*, 63(19):4846 – 4872. Model-Based Experimental Analysis.
9. Gautier, L. (2016). Rpy2, an interface to r running embedded in a python process. http://rpy2.bitbucket.org [Online; accessed 29-November-2016].
10. Geier, S., Winkler, C., and Piesche, M. (2009). Numerical Simulation of Mold Filling Processes with Polyurethane Foams. *Chemical Engineering & Technology*, 32(9):1438–1447.
11. Harikrishnan, G., Patro, T. U., and Khakhar, D. V. (2006). Polyurethane FoamClay Nanocomposites: Nanoclays as Cell Openers. *Industrial & Engineering Chemistry Research*, 45(21):7126–7134.
12. Jain, A., Ong, S. P., Chen, W., Medasani, B., Qu, X., Kocher, M., Brafman, M., Petretto, G., Rignanese, G.-M., Hautier, G., Gunter, D., and Persson, K. A. (2015). Fireworks: a dynamic workflow system designed for high-throughput applications. *Concurrency and Computation: Practice and Experience*.
13. Jain, Anubhav (2016). Fireworks: An evirnoment for defining, managing, and executing workflows.
14. Karimi, M., Droghetti, H., and Marchisio, D. L. (2016). Multiscale Modeling of Expanding Polyurethane Foams via Computational Fluid Dynamics and Population Balance Equation. *Macromolecular Symposia*, 360(1):108–122.
15. Kevrekidis, I. G., Gear, C. W., Hyman, J. M., Kevrekidid, P. G., Runborg, O., and Theodoropoulos, C. (2003). Equation-free, coarse-grained multiscale computation: Enabling mocroscopic simulators to perform system-level analysis. *Commun. Math. Sci.*, 1(4):715–762.
16. Kim, C. and Youn, J. R. (2000). Environmentally Friendly Processing of Polyurethane Foam for Thermal Insulation. *Polymer-Plastics Technology and Engineering*, 39(1):163–185.
17. Klempner, D. and Frisch, K. (1991). *Handbook of Polymeric Foams and Foam Technology*. Hanser Gardner Publications.
18. Marchisio, D. and Fox, R. (2013). *Computational Models for Polydisperse Particulate and Multiphase Systems*. Cambridge Series in Chemical Engineering. Cambridge University Press.
19. Marchisio, D. L. and Fox, R. O. (2005). Solution of population balance equations using the direct quadrature method of moments. *Journal of Aerosol Science*, 36(1):43–73.
20. MoDeNa-EUProject (2016). Modelling of morphology development of micro- and nanostructures. https://github.com/MoDeNa-EUProject/MoDeNa [Online; accessed 29-November-2016].
21. MongoDB Inc. (2016). A document-oriented database program. https://mongodb.com [Online; accessed 29-November-2016].
22. MongoEngine (2016). A document-object mapper for working with mongodb from python. http://mongoengine.org [Online; accessed 29-November-2016].
23. Oertel, G. and Abele, L. (1994). *Polyurethane Handbook: Chemistry, Raw Materials, Processing, Application, Properties*. Hanser.
24. R Core Team (2016). R: A language and environment for statistical computing. https://r-project.org [Online; accessed 29-November-2016].

25. Samkhaniani, N., Gharehbaghi, A., and Ahmadi, Z. (2013). Numerical simulation of reaction injection molding with polyurethane foam. *Journal of Cellular Plastics*, 49(5):405–421.
26. Seo, D., Ryoun Youn, J., and Tucker, C. L. (2003). Numerical simulation of mold filling in foam reaction injection molding. *International Journal for Numerical Methods in Fluids*, 42(10):1105–1134.
27. Seo, D. and Youn, J. R. (2005). Numerical analysis on reaction injection molding of polyurethane foam by using a finite volume method. *Polymer*, 46(17):6482–6493.
28. Weinan, E., Engquist, B., and Huang, Z. (2003). Heterogeneous multiscale method: A general methodology for multiscale modeling. *Phys. Rev. B*, 67:092101.
29. Winkler, C.-A. (2009). *Numerische und experimentelle Untersuchungen zu Formfüllvorgängen mit Polyurethanschäsumen unter komplexen Randbedingungen*. PhD thesis, University of Stuttgart.
30. Woods, G. (1990). *Polyurethane Handbook: Chemistry, Raw Materials, Processing, Application, Properties*. ICI Polyurethanes and Wiley in Chichester, New York.

Simulation of a Moving-Bed Reactor and a Fluidized-Bed Reactor by DPM and MPPIC in OpenFOAM®

Kwonwoo Jang, Woojoo Han and Kang Y. Huh

Abstract Simulations are performed for a moving-bed reactor in a rotary kiln and a fluidized-bed reactor in a FINEX plant. The DEM (Discrete Element Method) and the MPPIC (Multiphase Particle-In-Cell) methods are combined with a compressible reacting flow in OpenFOAM® 2.3.x. The computational load is reduced by the DPM (Discrete Particle Method), in which a computational parcel represents a fixed number of identical particles in the DEM. The slumping and rolling modes are reproduced by adjusting particle–particle and particle–wall friction coefficients to match the regime map in Henein et al. [1]. Validation is performed in a pilot-scale rotary kiln for reduction of iron ore with heat input from LPG (Liquefied Petroleum Gas). Simulation results in a lab-scale reactor are validated against those by commercial software and experimental data for the fluidized-bed reactor. Simulation results show good agreement with actual operating data for an industrial-scale fluidized-bed reactor in the FINEX process. Reasonable trends are reproduced for the bed burners and the collective motion of particles of different diameters in the FINEX plant.

1 Introduction

Moving-bed reactors and fluidized-bed reactors are widely used in the industries of petroleum processing, coal gasification, nuclear plants, steel manufacturing and water and waste treatment [2]. Both bed systems usually involve high particle concentrations, intense chemical reaction, and strong convective and radiative heat transfer. Many experiments have been carried out in laboratories and pilot-scale reactors to propose and validate the analytical models for gas flow and particle behavior in the reactor. Computational fluid dynamics (CFD) is a promising tool for understanding complicated multiphase phenomena with rapidly increasing low-cost computational power. There are two modeling approaches for multiphase flow, Eulerian–Eulerian

K. Jang · W. Han · K. Y. Huh (✉)
Department of Mechanical Engineering, Pohang University
of Science and Technology, Pohang, Gyungbuk 790-784, Republic of Korea
e-mail: huh@postech.ac.kr

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®* ,
https://doi.org/10.1007/978-3-319-60846-4_30

and Eulerian–Lagrangian, for continuous and discrete particle phases [3–5]. Two Eulerian–Lagrangian approaches of interest are the Discrete Element Method (DEM) and the Multiphase Particle-in-Cell (MPPIC) method. The DEM solves the transport and collision of all particles going through transitional and rotational motion. The computational load for DEM tends to increase exponentially with the total number of particles for a large-scale industrial system [6]. There have been two approaches to reducing the computation time of DEM; one is the DPM employing computational parcels with each representing a fixed number of particles sharing the same characteristics and the other is the MPPIC model [7], which considers the stress gradient of the solid phase through an interaction force in the Eulerian field. The MPPIC is more efficient than the DEM for a large-scale problem, although a sufficient number of parcels are required to guarantee stability in the corresponding Eulerian grid and flow field [8]. In this work, we applied the open source code, OpenFOAM®, to simulate particles and gas flow in a moving-bed and a fluidized-bed, coupled with 3D turbulent flow, combustion, and heat transfer.

2 Physical Models

2.1 Discrete Particle Method (DPM)

The solid phase is represented by individual Lagrangian parcels, whereas the gas phase is described by the following Eulerian mass, momentum, and energy conservation equations:

Mass:

$$\frac{\partial\left(\theta_f \rho_f\right)}{\partial t} + \nabla \cdot \left(\theta_f \rho_f \mathbf{u}_f\right) = \delta \dot{m}_{\mathrm{pf}}; \tag{2.1}$$

Momentum:

$$\frac{\partial\left(\theta_f \rho_f \mathbf{u}_f\right)}{\partial t} + \nabla \cdot \left(\theta_f \rho_f \mathbf{u}_f \mathbf{u}_f\right) = -\theta_f \nabla p + \theta_f \rho_f \mathbf{g} + \nabla \cdot \left(\theta_f \boldsymbol{\tau}_f\right) + \mathbf{F}; \tag{2.2}$$

Species:

$$\frac{\partial\left(\theta_f \rho_f Y_{f,i}\right)}{\partial t} + \nabla \cdot \left(\theta_f \rho_f Y_f \mathbf{u}_f\right) = \nabla \cdot \left(\rho_f D \theta_f \nabla Y_{f,i}\right) + \delta \dot{m}_{i,\mathrm{chem}}; \tag{2.3}$$

Energy:

$$\frac{\partial}{\partial t}\left(\theta_f \rho_f h_f\right) + \nabla \cdot \left(\theta_f \rho_f h_f \mathbf{u}_f\right) = \theta_f \frac{Dp}{Dt} - \nabla \mathbf{q} + \dot{Q} + S_h. \tag{2.4}$$

Each particle is tracked for its translational and rotational motion by the following equations of motion [9]:

$$m_i \frac{\mathrm{d}^2 r_i}{\mathrm{d}t^2} = f_i + m_i g, \tag{2.5}$$

$$I_i \frac{\mathrm{d}^2 \omega_i}{\mathrm{d}t^2} = t_i, \tag{2.6}$$

for the mass of the ith particle, m_i, its position, r_i, and the total force, f_i, due to drag and contact with wall and other particles, except the gravitational force. I_i and t_i are the moment of inertia and the total torque of the ith particle. f_i is given as a sum of the drag force, f_D, the contact force, f_C, and the net pressure force by the surrounding gas flow. f_D is given by the Ergun-Wen and Yu model [10] to consider the effect of gas volume fraction in a dense particle bed. f_C is the sum of the normal and tangential contact forces, f_{norm} and f_{tang}, estimated as

$$f_{\mathrm{norm}} = k\delta + \lambda_0 v_n \tag{2.7}$$

$$f_{\mathrm{tang}} = -k_t \xi - \lambda_t v_t \tag{2.8}$$

where f_{norm} is given by the linear contact model, whereas f_{tang} is given by the sliding/sticking friction model. k, λ_0, δ and v_n are, respectively, the spring stiffness, damping coefficient, overlap in the soft sphere model and relative velocity in the normal direction. k_t and λ_t are the tangential spring stiffness and tangential dissipation parameter. ξ and v_t are the tangential overlap and tangential relative velocity. The friction force is modeled as $f_{\mathrm{friction}} = \mu f_{\mathrm{norm}}$ for the friction coefficient, μ. The tangential friction force, f_{tang}, is limited as

$$f_{\mathrm{tang}} = \min\left(f_C, \left|f_{\mathrm{tan}\,g}\right|\right). \tag{2.9}$$

In the DPM, each computational parcel represents n identical particles to reduce excessive computational load of the DEM to a manageable level. The force, f_{in}, excluding the contact force on a parcel, is modeled as [11]

$$f_{in} = n f_{i1}, \tag{2.10}$$

where f_{i1} is the force on each single particle according to an experimental correlation. Heat and mass transfer for a parcel are similarly modeled as

$$de_n = n\, de_1, \tag{2.11}$$

$$dm_n = n\, dm_1, \tag{2.12}$$

where de_1 and dm_1 are the estimations for each single particle by experimental correlations. The normal and tangential contact forces are represented by the com-

putational parcels contacting each other in the same way as the individual particles in Eqs. (2.7) and (2.8).

2.2 Multiphase Particle-In-Cell (MPPIC)

The gas phase equations in the MPPIC are the same as those in the DEM. The probability distribution function, $\phi\left(\mathbf{x}, \mathbf{u}_p, m_p, t\right)$, is defined to describe particle dynamics in terms of particle position, $\mathbf{x}$, particle velocity, $\mathbf{u}_p$, and particle mass, m_p. The particle phase is governed by the Liouville equation [12],

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \mathbf{u}_p) + \nabla_{u_p} \cdot (\phi \mathbf{A}) = 0, \tag{2.13}$$

where the particle acceleration is given by

$$\mathbf{A} = D_p(u_f - u_p) - \frac{1}{\rho_p}\nabla p + g - \frac{1}{\theta_s \rho_p}\nabla \tau. \tag{2.14}$$

The four terms on the right represent, respectively, drag force, pressure gradient, gravity, and gradient in the interparticle stress, τ [7]. The drag coefficient, D_p, is given by the Ergun-Wen and Yu model [10]. u_f and u_p are the fluid and particle velocities. ρ_p is the density of particles and θ_s is the particle volume fraction. Equation (2.14) corresponds to Eq. (2.5) with a different expression for the particle contact force, f_C, given in terms of the interparticle stress by the Harris and Crighton model [13] as

$$\nabla \tau = \frac{P_s \theta_s^{\beta}}{\max[(\theta_{\mathrm{CP}} - \theta_s), \varepsilon(1 - \theta_s)]}, \tag{2.15}$$

where θ_{CP} is the maximum volume fraction for the packed particle bed and P_s is the corresponding particle pressure. Aureais et al. [14] recommended a constant value between 2 and 5 for β. ε is a small number on the order of 10^{-7} [15].

3 Implementation Strategy in OpenFOAM®

The basic libraries, CollidingCloud and MPPICCloud, for DEM and MPPIC were included in OpenFOAM® version 2.3.x. The multiphase flow solver with these libraries was provided for a single species in an incompressible form. We developed a new solver and the associated libraries in a compressible form in three sequential stages for gas–solid reaction and heat transfer in a moving-bed reactor and a fluidized-bed reactor. First, the basic CollidingCloud and MPPICCloud are combined together with the physical models for heat transfer and gas–solid reaction. The

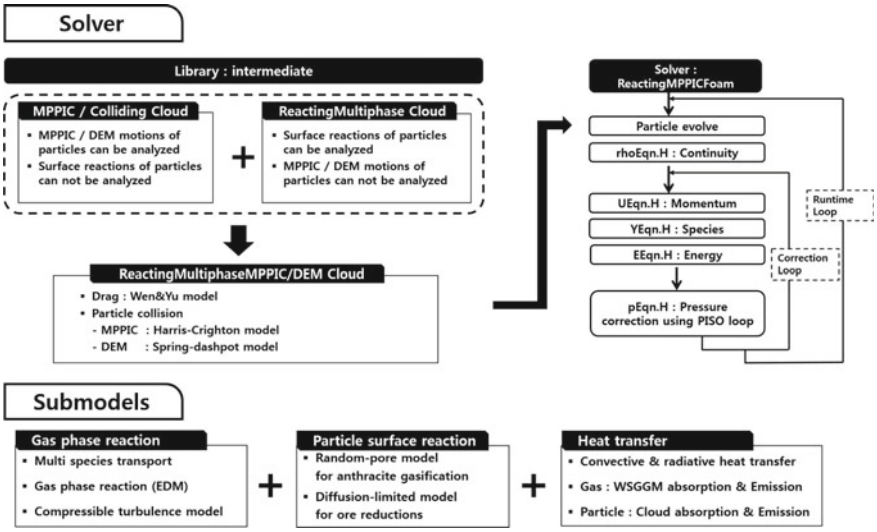


Fig. 1 Implementation strategy in OpenFOAM®

developed cloud libraries are respectively called ReactingMultiphaseCollidingCloud and ReactingMultiphaseMPPICCloud. Second, we modified the basic incompressible solver as a new compressible solver. The developed libraries are implemented into the new solver for heat transfer, reaction, and particle motion, as depicted in Fig. 1. In the final stage, we added the sub-models for chemical reaction and heat transfer. The eddy dissipation model was implemented for diffusion-controlled combustion in a compressible turbulent flow with multiple species transport in the gas flow solver [16]. The random pore model is implemented to consider the surface reaction on DPM particles [17]. The WSGGM (Weighted Sum of Gray Gases Model) was implemented for radiation in the gas phase [18].

4 Results for the Moving-Bed Reactor

4.1 Case Setup

Henein et al. [1] developed a regime map for particle motion in a rotary kiln in terms of bed depth versus rotational speed or percent fill versus Froude number. It includes particle motion in the slipping, slumping, rolling, cascading, and cataracting modes. The experimental conditions are listed in Table 1 [19]. The kiln is a cylinder with a diameter of 0.4 m and an axial length of 0.46 m. Particles are composed of limestone with a mean diameter of 4.3 mm. The friction coefficients are specified arbitrarily

Table 1 Experimental conditions in Henein et al. [19]

Property	Range
Material	Limestone
Kiln axial length (m)	0.46
Kiln diameter (m)	0.4
RPM range (rpm)	0.5–10
Bed depth range (m)	0.03–0.07
Particle diameter (mm)	4.3

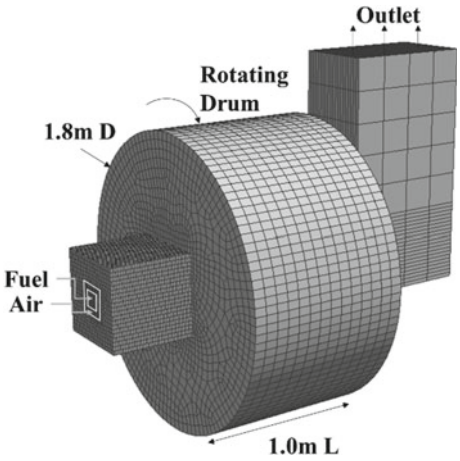


Fig. 2 Computational mesh for the pilot-scale rotary kiln in Tsuji [20]

as 0.1 between particles and 0.2 between particles and wall, with no reliable models available in the given conditions.

Tsuji [20] investigated experimentally the reduction behavior of iron ore in a pilot-scale rotary kiln. Simulation is performed for particles and gas flow in the 3D domain in Fig. 2 in this work. Table 2 lists the operation condition and ore compositions. The kiln rotates at 0.33 rpm with a diameter of 1.8 m and an axial length of 1.0 m. The bed is composed of briguettes of a homogeneous mixture at a uniform size of 30 mm $\times$ 25 mm $\times$ 15 mm. They are represented as spherical particles of the mean diameter of 27.8 mm of the homogeneous mixture treated as a single component in simulation. The initial particle temperature is 1273 K [20]. The reduction ratio is set initially at 5%, as in experiment. The friction coefficients are the same as those employed for the regime map in Henein et al. The adiabatic condition is specified on all wall boundaries.

Table 2 Experimental conditions and ore chemical compositions in Tsuji [20]

Property	Value	Chemical	Mass%
RPM	0.33	SiO_2	44.36
Ore (kg)	560	Fe	9.05
Coal (Anthracite) (kg)	101	Al_2O_3	0.41
Limestone (kg)	44.0	Ni	2.45
Particle size (mm)	30 × 25 × 15	CaO	0.08
Fuel	LPG	MgO	26.68
Fuel mass flow rate (m^3/h)	30		
Air mass flow rate (m^3/h)	709.5		

Table 3 Simulation cases and modes of particle motion

Case #	rpm	h (m)	θ (°)	Mode
1	10	0.035	23.2	Rolling
2	10	0.045	23.0	Rolling
3	10	0.053	23.3	Rolling
4	10	0.07	23.1	Rolling
5	0.5	0.035	21.3–27.2	Slumping
6	0.5	0.045	19.4–24.4	Slumping
7	0.5	0.053	21.3–25.8	Slumping
8	0.5	0.07	30.3	Rolling

4.2 Results and Discussion

4.2.1 Transverse Particle Motion

The cases in Table 3 involve bed depths between 0.035 and 0.07 m at 10 and 0.5 rpm. Simulation was performed with 4000 DPM parcels to represent about 10^5 particles for 14.8 kg of limestone [19]. Figure 3 shows simulation results for transverse particle motion in Henein's regime map. The particles were in the rolling mode for all bed depths at 10 rpm with an inclination angle, θ, of about 23°. The inclination angle did not show any explicit dependence on the bed depth. The particles showed the slumping mode at 0.5 rpm with particle movement of θ between about 20° and 25°, except for Case 8. The particle bed repeated the sequence of moving up with the wall and then sliding down against the wall periodically in the slumping mode. In the rolling mode, the external bed shape remains stationary as the particles slide down, forming an active layer on the top surface and an inactive region in the interior region [19].

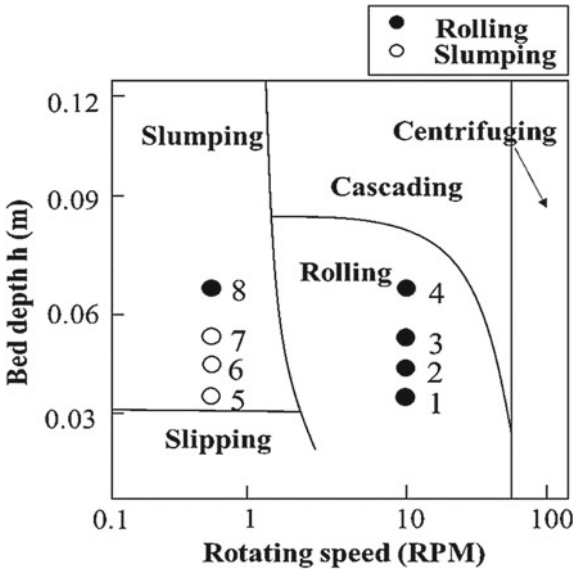


Fig. 3 Transverse particle motion in the regime map

4.2.2 Reduction of Iron Ore

Figure 4a shows the mean temperature distribution on the cross-sectional plane in the quasi-equilibrium state. Note the lower temperature in the bed due to the heat sink as a result of endothermic coal gasification reaction. There is a significant temperature gradient in the axial direction in the bed, as well as in the freeboard, as reduction proceeds with combustion gas flowing toward the outlet. The peak temperature is about 1800 K at the entrance of the kiln, while the minimum temperature of about 1000 K occurs at the lower right corner in the bed in Fig. 4a.

Figure 4b, c shows the temperature and the degree of reduction of the particles at 57 min since the initial state. Particles show higher temperatures and higher reduction ratios near the burner as a result of enhanced heat transfer from the freeboard gas at its maximum temperature. The reduction ratio is relatively uniform due to efficient mixing in the radial direction in Fig. 4c, while particles show higher temperatures on the bed surface as a result of heat transfer from the freeboard gas. This results from a heat transfer faster than particle mixing and reduction chemistry to maintain a non-negligible temperature gradient in the bed. Results show inefficient mixing of particles in the axial direction as compared with mixing in the radial direction. All computations were performed on a cluster with 60 cores of the 2nd Intel Xeon Processor E5-2650 at 2.3 GHz. It took about seven days for the simulation of one hour of real time with one thousand DPM parcels and about 55,000 fluid meshes. This could be evidence of the current DPM implementation not being fully optimized for parallel processing, since CFDEM [21] showed much faster performance for comparable simple problems in an incompressible flow.

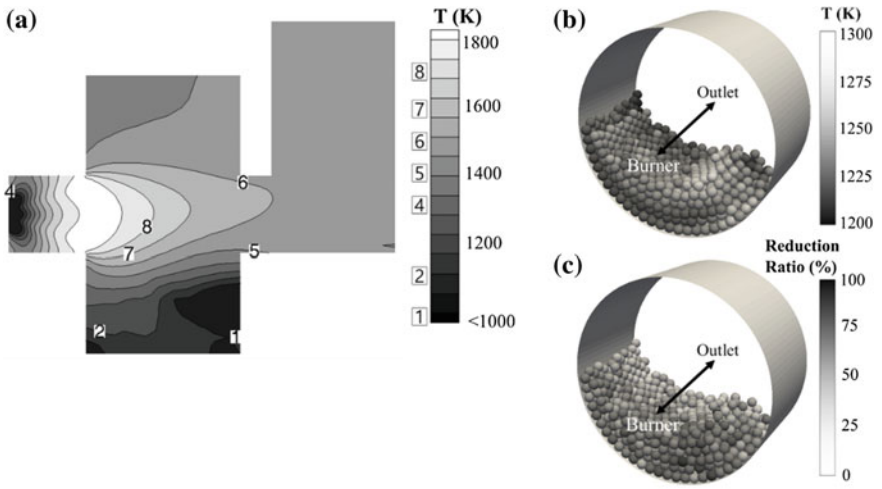


Fig. 4 Temperature distribution in the pilot-scale rotary kiln (**a**) and temperature (**b**) and reduction ratio (**c**) of the particles at 57 min in the pilot-scale rotary kiln

5 Results for the Fluidized-Bed Reactor

5.1 Lab-Scale Fluidized-Bed Reactor

5.1.1 Comparison with the Commercial Software

We compared the results with those by the commercial software, ANSYS-FLUENT [22], for validation of the newly developed solver and libraries. We selected the case of a simple uniformly fluidized-bed reactor in Goldschmidt et al. [23]. The computational domain is rectangular with the width of 0.15 m, the depth of 0.015 m and the height of 0.45 m as shown in Fig. 5. Glass beads of a uniform diameter of 2.5 mm are filled at the initial bed height of 0.15 m. Details of the operation conditions and the numerical models are summarized in Table 4. Figure 5 shows instantaneous contours of the gas volume fraction by ANSYS-FLUENT and OpenFOAM®. A large bubble at the bed surface shows similar shapes with no significant difference in the other aspects of the results by the two programs.

We also compared the results by OpenFOAM® with those by Barracuda [15] which is another commercial software widely used in fluidized-bed reactors. We performed simulation of Lin et al. [24] involving a cylinder filled with glass beads with the diameters ranging between 0.42 and 0.6 mm. The initial bed height is 11 cm and the mean fluidization velocity is 0.648 m/s. A 2D axisymmetric computational domain is shown in Fig. 6. Details of the operation conditions and the numerical models are summarized in Table 5. Figure 6 shows comparison of the measured particle volume fractions with the calculation results by OpenFOAM® and Barracuda. There is no

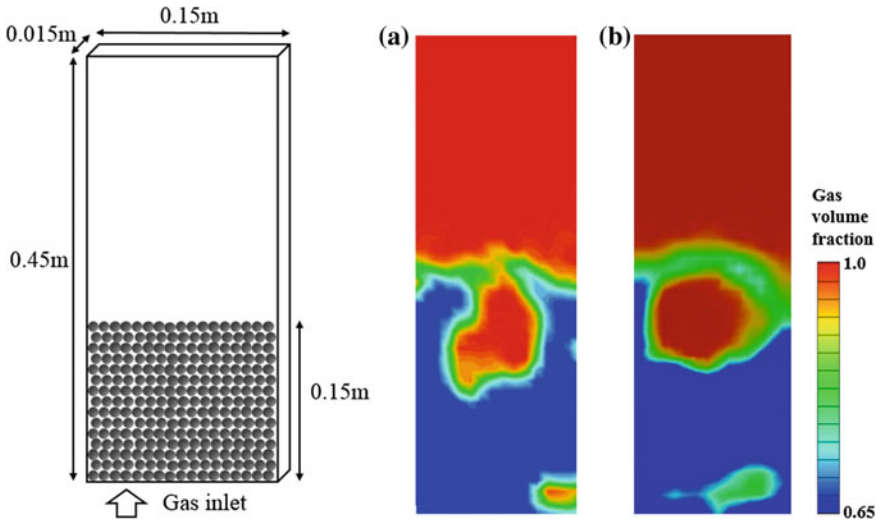


Fig. 5 Schematic configuration and instantaneous contours of the gas volume fraction **a** by ANSYS-FLUENT and **b** by OpenFOAM®

Table 4 Simulation conditions for the reactor in Goldschmidt et al. [23]

Operating conditions	Value	Numerical model	Value
Inlet gas	Air	Number of grid	675/2D
Fluidization velocity (m/s)	1.875	Time step size (s)	2×10^{-4}
Gas density (kg/m^3)	1.2	Drag model	Ergun-Wen and Yu
Particle density (kg/m^3)	2526		
Number of particle	24,750	Particle–particle interaction model	Harris-Crighton
Gas temperature (K)	293		

significant difference between the two programs with the predicted results showing a similar trend with experimental data but with significant deviation at some locations. Simulations show the particle volume fractions larger than measurements at the bottom, as the voidage due to high gas jet velocities could not be reproduced by the MPPIC model. Deviations in the peak volume fraction are caused by the particle behaviors with strong collision and circulation not properly taken into account in the MPPIC model and might be improved by modification of the model constants, P_s and β, in Table 5.

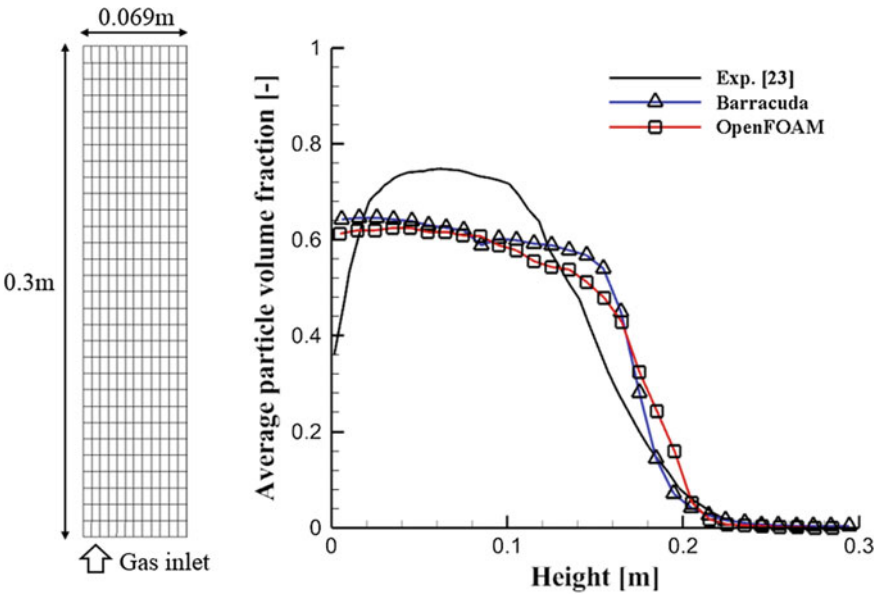


Fig. 6 Schematic configuration and comparison of average particle volume fractions

Table 5 Simulation conditions for the reactor in Lin et al. [24]

Operating conditions	Value	Numerical model	Value
Fluid density (kg/m^3)	1.093	Number of grid	360/2D axisymmetric
Fluid viscosity (Pa s)	1.95×10^{-5}		
Particle density (kg/m^3)	2500	Time step size (s)	1×10^{-4}
Number of particle	20,736	P_s (Pa)/β	75/5

5.1.2 Comparison with DEM Results

We compared the results by OpenFOAM® with experimental data and the results by DEM for another lab-scale reactor in Muller et al. [25]. It has a rectangular shape of the width of 44 mm, the depth of 10 mm and the height of 1200 mm, similar with the shape in Fig. 5. Poppy seeds were used as fluidizing particles at the initial bed height of 30 mm. Details of the operation conditions and the numerical models are summarized in Table 6.

Figure 7 shows comparison of the measured gas volume fractions with the simulation results at two different heights. Note good agreement of the gas volume fractions calculated by DEM and MP-PIC approaches with experimental data in Fig. 7. The MP-PIC considers only linear collision of particles, whereas the DEM takes into account both linear and rotational collisions. Better agreement is observed in the

Table 6 Simulation conditions for the reactor in Muller et al. [25]

Operating conditions	Value	Numerical model	Value
Air temperature (K)	298.15	Number of grid	3000/3D
Fluid viscosity (Pa s)	1.8×10^{-5}	Time step size (s)	1×10^{-4}
Particle density (kg/m^3)	1000	Operating time (s)	23
Number of particle	9240	P_s (Pa)/β	10/2

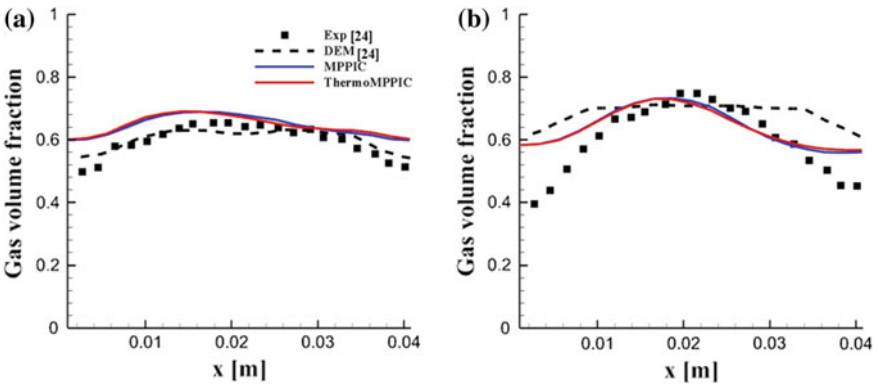


Fig. 7 Comparison of the mean gas volume fractions by experiment, DEM and MP-PIC at the heights of **a** 16.4 mm and **b** 31.2 mm

central region at the mid-axial location of $y = 31.2$ mm, whereas there is larger deviation near the walls on both sides. No significant difference was confirmed in the characteristics of the fluidized bed with constant gas density between the original MPPICFoam and the newly developed ThermoMPPICFoam.

5.2 Industrial-Scale Fluidized-Bed Reactor

5.2.1 Case Setup

Figure 8 shows the FINEX process, a new iron-making technology under development by POSCO. It produces less pollutants and is more economical than the conventional blast furnace since it uses lower quality ore and coal without any pretreatment processes such as cokes and sinter plants [26]. The major role of fluidized-bed reactors is in reducing the ore fine and increasing the particle temperature to produce hot compacted iron (HCI). The R2 reactor is about 17 m high and the radius of the distributor is about 4.5 m. A mixture of carbon monoxide, methane, and hydrogen gas is injected through 960 holes in the distributor on the bottom. The iron ore has

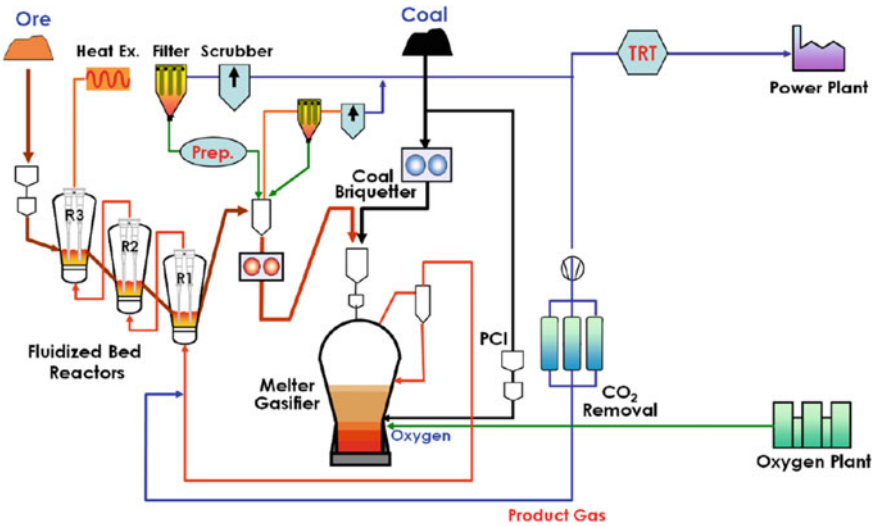


Fig. 8 A schematic diagram of the FINEX process [26]

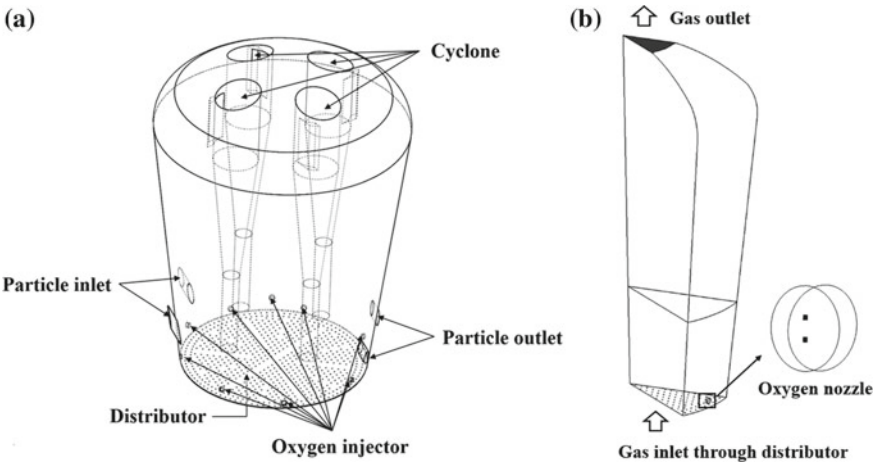


Fig. 9 Simplified geometry of the fluidized-bed R2 reactor; **a** full reactor, **b** 1/10 sector

diameters between 0.063 and 8 mm and ten burners are installed as heat sources around the bottom of the bed in the R2 reactor. The oxygen burner has two small nozzle holes, with the two injections meeting at an angle of 45°. A 1/10 sector of the reactor is modeled to reduce the computational cost in this work. The 1/10 sector consists of approximately 0.8 million hexahedral cells and 2 million computational parcels in MPPIC. The simplified reactor geometry is shown in Fig. 9.

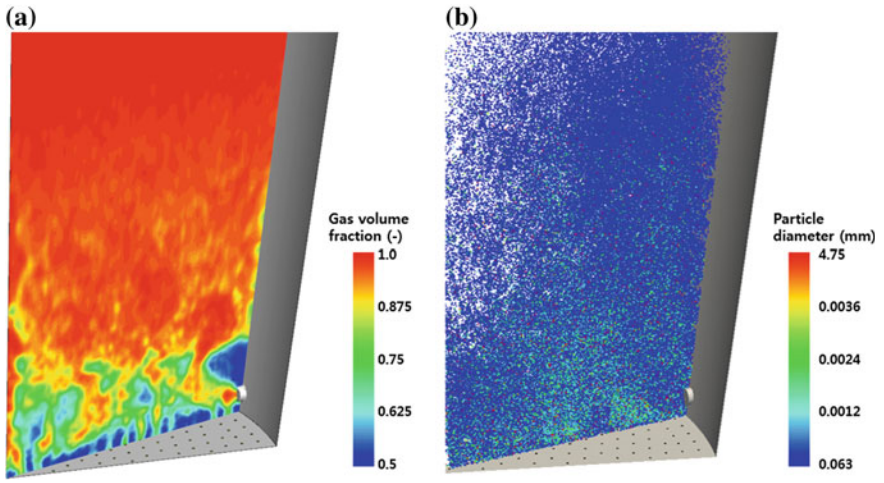


Fig. 10 Instantaneous images of **a** gas volume fraction and **b** particles motion

5.2.2 Simulation Under Actual Operating Conditions

The standard $k-\varepsilon$ model is employed with the EDM (Eddy Dissipation Model) as the turbulent combustion model [16]. The WSGGM [18] is employed to estimate the gas phase absorption and emission coefficients for radiative heat transfer. Figure 10 shows instantaneous images of the gas volume fraction and the particle motion of different particle diameters. It shows a reasonable trend of the internal field, including the gas volume fractions. It is confirmed that bubbles form near the bottom by the jet flow through the distributor. Most small particles tend to move and accumulate along the reactor wall in the upper reactor region.

The axial pressure difference is important for confirming the validity of the predictions of particle motion. It largely depends on the total mass of particles and remains approximately constant beyond the minimum fluidization velocity [27]. Figure 11 shows good agreement with the measured pressure differences, with minor deviation due to inaccurate locations of the pressure transducers away from the distributor. Particles are recirculated continuously by the gas flow through the distributor, while the gas temperature remains relatively uniform due to violent gas phase mixing and heat transfer with particles by turbulence.

6 Conclusion

(1) Simulations are performed for a coupled solution of particle bed, turbulent flow, combustion and heat transfer in a pilot-scale rotary kiln and a full-scale fluidized-bed reactor in the FINEX plant. New libraries are developed for DPM

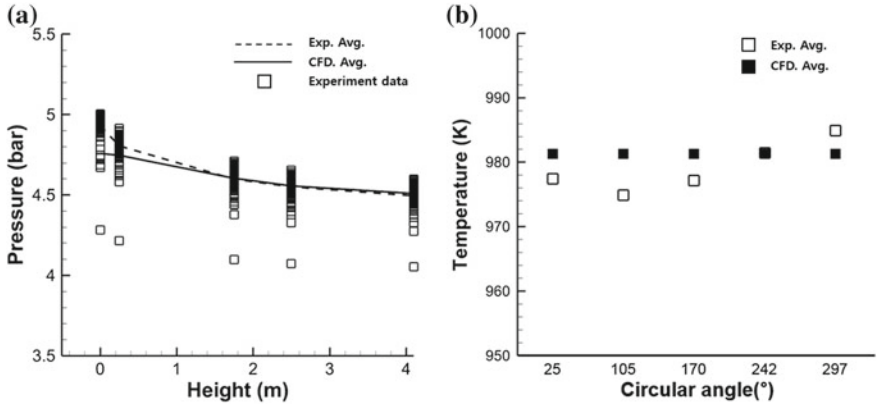


Fig. 11 Measurements and predictions of **a** pressure and **b** mean temperature

and MPPIC to reduce the computational load and to be combined with the compressible reacting flow solver in OpenFOAM® version 2.3.x.

(2) Particle motions are reproduced for the rolling and slumping modes of a moving bed in Henein's regime map. Particles flow continuously forming an active layer on the top surface of the bed, while particles move at a slower rate in the inactive core region in the bed in the rolling mode. More work is required for proper specification of the friction coefficients in different regimes.

(3) Validation is performed in a pilot-scale rotary kiln for reduction of iron ore with heat input from LPG fuel. Results show non-negligible variation of particle temperature and reduction ratio in the radial direction as well as in the axial direction in the bed. This is due to heat transfer with freeboard gas occurring faster than the coal gasification reaction and radial mixing of particles under the given experimental conditions.

(4) The simulation results were validated against those by ANSYS-FLUENT, Barracuda and experimental data in the lab-scale fluidized-bed reactors in literature. There was no significant difference between the results by OpenFOAM® and the commercial software, with good agreement with available experimental data.

(5) Particle motions including recirculation, bubble formation and destruction are successfully reproduced in the fluidized-bed R2 reactor of the FINEX plant. The gas temperature is relatively uniform due to efficient mixing and heat transfer with particles in the bed. Good agreement is shown for the axial pressure difference, although with minor deviation due to inaccurate pressure measurement locations.

References

1. H. Henein, J.K. Brimacombe, A.P. Watkinson (1983) The modeling of transverse solids motion in rotary kilns. Metall Trans B 14 (2):207–220.
2. P. Trambouze, J.-P. Euzen (2004) Chemical Reactors: From Design to Operation. Technip, Paris.
3. P.A. Cundall, O.D. Strack (1979) A discrete numerical model for granular assemblies. Geotechnique 29 (1):47–65.
4. Y. Tsuji, T. Kawaguchi, T. Tanaka (1993) Discrete particle simulation of two-dimensional fluidized bed. Powder Technol 77 (1):79–87.
5. Y. Tsuji (2007) Multi-scale modeling of dense phase gas–particle flow. Chem Eng Sci 62(13):3410–3418.
6. Dalibor. Jajcevic, Eva Siegmann, Charles Radeke et al (2013) Large-scale CFD-DEM simulations of fluidized granular systems. Chem Eng Sci 98(19):298–310.
7. M. J. Anderews, P.J. O'Rourke (1996) The multiphase particle-in-cell (MP-PIC) method for dense particulate flows. Int J Multiphase Flow 22(2):379–402.
8. Sreekanth Pannala, Madhava Syamlal, Thomas J. O'Brien (2010) Computational gas-solids flows and reacting system: Theory, Methods and Practice. IGI Global, New York.
9. Stefan Luding (2008) Introduction to discrete element methods: basic of contact force models and how to perform the micro-macro transition to continuum theory. European Journal of Environmental and Civil Engineering 12:785–826.
10. C. Y. Wen, Y. H. Yu (1966) A generalized method for predicting the minimum fluidization velocity, Chem. Eng. AIChE J 12(3):610–612.
11. Mikio Sakai, Yoshinori Yamada, Yusuke Shigeto et al (2010) Large-scale discrete element modeling in a fluidized bed. Int J Numer Methods Fluids 64:1319–1335.
12. F.A. Williams (1985) Combustion Theory, 2nd edition. The Benjamin/Cumming Publishing, California.
13. S. E. Harris, D. G. Crighton (1994) Solitons, solitary waves, and voidage disturbances in gas-fulidized beds. J Fluid Mech 266:243–276.
14. F. M. Aureais, R. Jackson, W. B. Russel (1988) The solution of shocks and the effects of compressible sediments in transient settling. J Fluid Mech 195:437–462.
15. D. M. Snider (2001) An incompressible three-dimensional multiphase particle-in-cell method for dense particle flows. J Comput Phys 170(2):523–549.
16. N. Peters (2000) Turbulent Combustion. Cambridge University Press, Cambridge.
17. H. Watanabe, M. Otaka (2006) Numerical simulation of coal gasification in entrained flow coal gasifier. Fuel 85:1935–1943.
18. John R. Howell, Pinar Menguc, Robert Siegel (2015) Thermal radiation heat transfer, 6th edition. CRC press, Florida.
19. H. Henein, J.K. Brimacombe, A.P. Watkinson (1983) Experimental study of transverse bed motion in rotary kilns. Metall Trans B 14 (2):191–205.
20. H. Tsuji (2012) Behavior of reduction and growth of metal in Smelting of Saprolite Ni-ore in a Rotary Kiln for Production of Ferro-nickel Alloy. ISIJ Int 52 (6):1000–1009.
21. CFDEM (2016) CFEDM Benchmarks. http://www.cfdem.com/cfdem-benchmarks.
22. ANSYS. ANSYS FLUENT User's Guide. ANSYS Inc., Cannonsburg, PA, 2012.
23. M.J.V. Goldschmidt, R. Beetstra, J.A.M. Kuipers (2004) Hydrodynamic modelling of dense gas-fluidised beds: comparison and validation of 3D discrete particle and continuum models. Powder Technol 142:23–47.
24. J. S. Lin, M. M. Chen, B. T. Chao (1985) A novel radioactive particle tracking facility for measurement of solids motion in gas fluidized beds. AIChE J 31(3):465–473.

25. C. R. Muller, S. A. Scott, D. J. Holland et al. (2009) Validation of a discrete element model using magnetic resonance measurements. Particuology 7(4):297–306.
26. POSCO (2014) Technological Advance of the FINEX® Ironmaking Process. Paper presented at Asian Pellets and DRI Conference, Zurich Marriott Hotel, Singapore, 8–9 July 2014.
27. D. Kunii, O. Levenspiel (1991) Fluidization Engineering, 2nd edition. Butterworth-Heinemann, Oxford.

Simulation of Particulate Fouling and its Influence on Friction Loss and Heat Transfer on Structured Surfaces using Phase-Changing Mechanism

Robert Kasper, Johann Turnow and Nikolai Kornev

Abstract Numerical simulations of particulate fouling using highly resolved Large-Eddy Simulations (LES) are carried out for a turbulent flow through a smooth channel with a single spherical dimple or square cavity (dimple depth/cavity depth to dimple diameter/cavity side length ratio of $t/D = 0.261$) at $\mathrm{Re}_D = 42{,}000$. Therefore, a new multiphase method for the prediction of particulate fouling on structured heat transfer surfaces is introduced into OpenFOAM® and further described. The proposed method is based on a combination of the Lagrangian Particle Tracking (LPT) and Eulerian approaches. Suspended particles are simulated according to their natural behavior by means of LPT as solid particles, whereas the carrier phase is simulated using the Eulerian approach. The first numerical results obtained from LES approve the capabilities of the proposed method and reveal a superior fouling performance of the spherical dimple due to asymmetric vortex structures, compared to the square cavity.

1 Introduction

The common way to evaluate the performance of heat transfer enhancement methods like ribs, fins, or dimples is the determination of the thermo-hydraulic efficiency, which is the relationship of the increased heat exchange to pressure loss [1]. Despite the fact that particulate fouling, the accumulation of particles on the heat transfer surfaces, reduces thermo-hydraulic performance significantly, an universal method for the prediction of particulate fouling still does not exist. With respect to the large time instants and variety of fouling, its influence is mainly determined using experimental investigations. Due to improved numerical algorithms and access to high computational resources, numerical simulations of fouling have become more and more advantageous. However, commonly used fouling models are derived for a

R. Kasper · J. Turnow (✉) · N. Kornev
Chair of Modeling and Simulation, University of Rostock,
Albert-Einstein-Str. 2, 18059 Rostock, Germany
e-mail: johann.turnow@uni-rostock.de

R. Kasper
e-mail: robert.kasper@uni-rostock.de

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_31

definite set of boundary conditions and are calibrated for specific cases [2]. Hence, the existing fouling approaches are unsuitable for a general prediction and analysis of particulate fouling for industrial applications. Within the present work, a new approach has been introduced to determine the local fouling layer growth and its influence on heat transfer and pressure loss. Numerical simulations using different existing fouling algorithms have shown that the efficiency of the numerical simulations depend enormously on the fouling model used and its empirical parameters, in combination with the complexity of the heat transfer surface. To reduce computational time and increase the universality of the model, the proposed method was developed based on Lagrangian Particle Tracking (LPT) and the Euler approach, in which the deposited particles are transferred into an extra fouling layer phase with predefined material properties, causing additional friction losses and heat transfer resistance. The vortex formations and their direct impact on fouling probability, and thus on the heat transfer for a single spherical dimple and rectangular cavity, are analyzed and compared to results obtained for a clean surface.

2 Multiphase Approach for the Simulation of Particulate Fouling

The numerical simulation of particulate fouling on heat transfer surfaces is complex, consisting mainly of the deposition of small particles due to adhesion and sedimentation of larger particles resulting from gravitational forces. Therefore, the proposed multiphase method for the simulation of particulate fouling is composed of two different branches that are closely related to each other. The first one is the Lagrangian branch and describes the physics of the suspended particles or, respectively, the dispersed phase using the LPT. This branch is mainly responsible for the transport of the particles to the heat transfer surfaces, the deposition of the particles due to adhesion and sedimentation, and also the resuspension of deposited particles due to local shear stresses. The second one is the Eulerian branch and determines the flow fields of the carrier flow (i.e., the continuous phase) with respect to the settled fouling layer, which are then again required within the Lagrangian branch.

2.1 Lagrangian Branch

The description of particle motions within a fluid using the Lagrangian Particle Tracking (LPT) requires the solution of the following set of ordinary differential equations along the particle trajectory, which enables the calculation of the particle location and the linear, as well as the angular, particle velocity at anytime:

$$\frac{\mathrm{d}\mathbf{x}_p}{\mathrm{d}t} = \mathbf{u}_p, \tag{1}$$

$$m_p \frac{\mathrm{d}\mathbf{u}_p}{\mathrm{d}t} = \sum \mathbf{F}_i, \tag{2}$$

$$I_p \frac{\mathrm{d}\boldsymbol{\omega}_p}{\mathrm{d}t} = \sum \mathbf{T}, \tag{3}$$

where $m_p = \pi/6\rho_p D_p^3$ is the particle mass, $I_p = 0.1 m_p D_p^2$ is the moment of inertia (for a sphere), $\mathbf{F}_i$ includes all the relevant forces acting on the particle, and $\mathbf{T}$ is the torque acting on a rotating particle due to viscous interaction with the carrier fluid [3]. Equation (2) represents Newton's second law of motion and presupposes the consideration of all relevant forces acting (drag, gravity and pressure forces) on the particle

$$m_p \frac{\mathrm{d}\mathbf{u}_p}{\mathrm{d}t} = \sum \mathbf{F}_i = \mathbf{F_D} + \mathbf{F_G} + \mathbf{F_P} + \cdots \tag{4}$$

However, analytical solutions for different forces exists only for small particle Reynolds numbers, respectively, for the Stokes regime [4]. Due to the fact that the consideration of intermediate and high particle Reynolds numbers is also desirable, the LPT used in this work is extended to a wide range of particle Reynolds numbers. The implemented drag model is based on the particle Reynolds number, which is defined as

$$\mathrm{Re}_p = \frac{\rho_f D_p \left|\mathbf{u}_f - \mathbf{u}_p\right|}{\mu_f}, \tag{5}$$

with the density ρ_f and the dynamic viscosity μ_f of the fluid or continuous phase, the particle diameter D_p and the difference between flow and particle velocity $\left|\mathbf{u}_f - \mathbf{u}_p\right|$. The drag coefficient is now determined, based on the particle Reynolds number, through the following empirical relation proposed by Putnam [5]:

$$C_D \mathrm{Re}_p = \begin{cases} 24\left(1 + \frac{1}{6}\mathrm{Re}_p^{2/3}\right) & \text{if } \mathrm{Re}_p \leq 1000 \\ 0.424\mathrm{Re}_p & \text{if } \mathrm{Re}_p > 1000. \end{cases} \tag{6}$$

After determination of the drag coefficient, the basic equation of motion for a spherical particle is used to evaluate the drag force

$$\mathbf{F_D} = C_D \frac{\pi D_p^2}{8} \rho_f \left(\mathbf{u}_f - \mathbf{u}_p\right) \left|\mathbf{u}_f - \mathbf{u}_p\right|. \tag{7}$$

In addition to the drag force, the gravitational and buoyancy force and the pressure gradient force have to be taken into account as well. Within the LPT used, gravitation and buoyancy are computed as one total force as follows:

$$\mathbf{F_G} = m_p \mathbf{g} \left(1 - \frac{\rho_f}{\rho_p}\right), \tag{8}$$

where **g** is the gravitational acceleration vector. The resultant force due to a local fluid pressure gradient acting on a spherical particle can be found from Eq. (9) using the differential form of the momentum equation to express the pressure gradient:

$$\mathbf{F_P} = \rho_f \frac{\pi D_p^3}{6} \left(\frac{\mathrm{D}\mathbf{u}_f}{\mathrm{D}t} - \nabla \cdot \nu_f \left(\nabla \mathbf{u}_f + \nabla \mathbf{u}_f^T \right) \right). \tag{9}$$

From Newton's third law of motion, it follows that if a particle is either accelerated or decelerated in a fluid, an accelerating or decelerating of a certain amount of the fluid surrounding the particle is required. This additional force is known as added mass force, sometimes referred to as virtual mass force, and can be expressed as

$$\mathbf{F_A} = C_A \rho_f \frac{\pi D_p^3}{6} \left(\frac{\mathrm{D}\mathbf{u}_f}{\mathrm{D}t} - \frac{\mathrm{d}\mathbf{u}_p}{\mathrm{d}t} \right), \tag{10}$$

where C_A is the so-called added mass coefficient. This coefficient can be exactly derived for spherical particles from potential theory and is $C_A = 1/2$.

The last considered force arises due to local shear flows, and therefore from a nonuniform velocity distribution over the particle surface. This lift force is called the Saffman force and is modeled using the Saffman-Mei model, derived by Saffman [6, 7] and advanced by Mei [8]. In order to determine the lift force due to local shear flows, the shear Reynolds number has to be calculated

$$\mathrm{Re}_s = \frac{\rho_f D_p^2 \left| \nabla \times \mathbf{u_f} \right|}{\mu_f}, \tag{11}$$

which is used to evaluate the coefficients of the Saffman-Mei model

$$\beta = \frac{1}{2} \frac{\mathrm{Re}_s}{\mathrm{Re}_p}, \qquad \alpha = 0.3314\sqrt{\beta}, \qquad f = (1-\alpha)\exp(-0.1\mathrm{Re}_p) + \alpha. \tag{12}$$

Afterward, the lift coefficient C_{LS} is calculated using the following approximation:

$$C_{LS} = \begin{cases} 6.46 f & \text{if } \mathrm{Re}_p < 40 \\ 6.46 \cdot 0.0524\sqrt{\beta \mathrm{Re}_p} & \text{if } \mathrm{Re}_p \geq 40. \end{cases} \tag{13}$$

The lift coefficient C_{LS} is now expressed in terms of a nondimensional lift coefficient

$$C_L = \frac{3}{2\pi\sqrt{\mathrm{Re}_s}} C_{LS}. \tag{14}$$

This conversion allows for a more universal way of determining the lift force using any conceivable force model. Finally, the lift force is calculated as

$$\mathbf{F_L} = C_L \rho_f \frac{\pi D_p^3}{6} \left(\mathbf{u}_f - \mathbf{u}_p\right) \times \left(\nabla \times \mathbf{u}_f\right). \tag{15}$$

In summary, it can be stated that the proposed LPT is capable of considering the most important forces acting on a particle. Because numerical simulations of Sommerfeld have shown that the consideration of the Basset force increases the computational time by a factor of about 10 [3], this force is neglected. This strategy is valid for small density ratios $\rho_f/\rho_p << 1$ [4], which is probably not the case for liquid-solid flows, as investigated below. Thus, the influence of the Basset force has to be analyzed in the future.

However, another important concept in the analysis of dispersed multiphase flows is phase coupling. One-way coupling exists if the carrier flow effects the particles while there is no reverse effect. If there is a mutual effect between carrier flow and particles, then the flow is two-way coupled [4]. In the case of dense flows, there will be an additional interaction among the particles themselves, which is what is meant by four-way coupling. The proposed method is capable of considering all different types of coupling.

2.1.1 Deposition of Particles

As already mentioned, the proposed method has to include an algorithm for the physical modeling of the deposition of suspended particles on solid walls or, more precisely, on heat transfer surfaces. Due to the fact that particle deposition is mainly caused by particle–wall adhesion within this work, the implemented model is based on the suggestions of Löffler and Muhr [9] and, furthermore, Heinl and Bohnet [10]. This model consists of an energy balance around the particle–wall and particle-fouling collision. Thus, a critical particle velocity can be derived from a local energy balance, which contains the kinetic energy before and after the collision, the energy ratio describing the adhesion due to van der Waals forces and a specific amount considering the energy loss of a particle resulting from particle–wall and particle-fouling collision. From the condition of adhesion (i.e., a particle is not able release itself from the wall after the collision), the critical particle velocity yields

$$u_{p,\mathrm{crit}} = \sqrt{\left(\frac{\hbar\varpi}{e D_p 4\pi^2 z_0^2}\right)^2 \frac{3}{4 H \rho_p}}, \tag{16}$$

where $\hbar\varpi$ is the Liffschitz–van der Waals energy, z_0 is the distance at contact, H is the strength of the contact wall, and e is the coefficient of restitution. It should be mentioned at this point that the determination of the critical particle velocity (16) can be easily extended for the consideration of electrostatic forces [9, 10]. However, the condition of adhesion is achieved, if the particle velocity before the wall collision (impact velocity) is smaller then the critical particle velocity

$$\left|\mathbf{u}_p\right| \leq u_{p,\mathrm{crit}}. \tag{17}$$

To increase the computational efficiency of our proposed method, particles that fulfill the adhesion condition, Eq. (17), are converted into an additional continuous/solid phase (fouling layer) and will be deactivated within the LPT. Thus, the amount of particles is kept nearly constant during the calculations, which reduces the computational time enormously. The initiated volume or phase fraction α is evaluated by the particle volume with respect to the cell volume

$$\alpha_{new,i} = \alpha_{old,i} + \frac{V_p}{V_i}, \tag{18}$$

where $\alpha_{old,i}$ is the phase fraction from the previous time step and V_p and V_i are the particle and cell volume, respectively. Figure 1 shows the basic concept of the implemented phase conversion algorithm. According to this, it can be distinguished between two different cases if a deposited particle has to be converted into the fouling phase. If the residual local cell volume is greater than the particle volume, the new phase fraction α can be simply determined using Eq. (18). If the remaining local cell volume is smaller than the particle volume, the phase fraction is allocated to the neighbor cell with the maximum cell-based phase fraction gradient $\max(\nabla\alpha)$. Hence, the neighbor cell with the lowest phase fraction is filled with the fouling phase. To consider the influence of the fouling phase, an additional porosity source term (based on Darcy's law)

$$\mathbf{S}_p = \alpha \frac{\mu_f}{K} \mathbf{u}_f, \tag{19}$$

has been introduced into the momentum balance equation, where K is the permeability of the fouling phase. Thus, the blocking effect or flow section contraction due to deposited particles is not explicitly considered within the calculations, but rather is modeled implicitly in terms of a porous fouling layer. Furthermore, any physical property x_i (e.g., density, dynamic/kinematic viscosity, or thermal diffusivity) for partially filled cells is interpolated as follows:

$$x_i = \alpha \cdot x_{fouling} + (1 - \alpha) \cdot x_{fluid}, \tag{20}$$

whereas, the physical properties of the carrier fluid are fully applied at cells without the fouling phase ($\alpha = 0$) and cells that are completely occupied by the fouling phase ($\alpha = 1$) take the physical properties of the fouling material. This procedure likewise

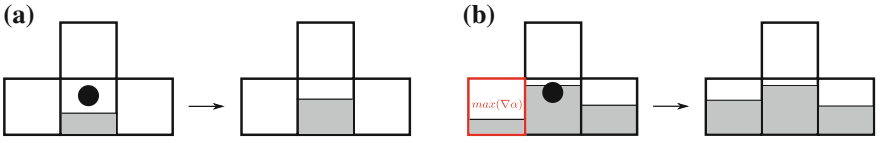


Fig. 1 Basic mechanism of the phase conversion algorithm: **a** $V_c > V_p$ and **b** $V_c < V_p$

allows for evaluation of the heat transfer under consideration of particulate fouling and prevents the solving of an additional advection/transport equation for the fouling phase, as well as the application of costly remeshing procedures.

2.1.2 Resuspension of Deposited Particles

The resuspension of deposited particles (i.e., the release of particles from the fouling layer and re-entrainment into the carrier fluid due to high local shear stresses) is an important mechanism, which has to be considered in the proposed approach to simulate the particulate fouling as accurately as possible. Therefore, a simple resuspension model is derived based on the Kern and Seaton model [2]

$$\alpha_{removed} = \min\left(\frac{V_p}{\tau_{rel}}\frac{|\tau_c|}{V_c}, \frac{V_p}{V_c}\right), \tag{21}$$

where τ_{rel} is a relative shear stress and τ_c is the cell-based local shear stress. The relative shear stress has to be measured in experiments and can be interpreted as a threshold value for the release of fouling volume due to high local shear stresses. The number of resuspended (spherical) particles can be determined using the definition of the sphere volume

$$n = \frac{\alpha_{removed} V_c}{\pi D_p^3/6}. \tag{22}$$

The resuspended particles will be re-activated and become part of the LPT again, whereby the initial momentum and forces are calculated according to the force models described above.

2.2 Eulerian Branch

The governing equations are the incompressible Navier–Stokes equations (extended by the porosity source term $\mathbf{S}_p$, which takes the influence of the fouling layer into account), the continuity equation and a passive scalar transport equation for the temperature. This system of partial differential equations is solved numerically using OpenFOAM®. Although the turbulence modeling is generic, Large-Eddy Simulations (LES) are carried out to investigate the interaction between local vortex structures and particulate fouling. LES is a widely used technique for simulating turbulent flows and allows one to explicitly solve for the large eddies and implicitly account for the small eddies by using a Subgrid-Scale model (SGS model).

2.2.1 Large-Eddy Simulation

The LES equations are derived by filtering the continuity equation, the Navier–Stokes equations and the passive scalar transport equation for the temperature using an implicit box filter with a filter width $\overline{\Delta}$ (depending on the computational grid):

$$\nabla \cdot \overline{\mathbf{u}} = 0, \tag{23}$$

$$\frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}}\,\overline{\mathbf{u}}) = -\frac{1}{\rho}\nabla p + \nabla \cdot \nu \left(\nabla \overline{\mathbf{u}} + \nabla \overline{\mathbf{u}}^T\right) + \nabla \cdot \boldsymbol{\tau}_{SGS} - \overline{\mathbf{S}}_p, \tag{24}$$

$$\frac{\partial \overline{T}}{\partial t} + \nabla \cdot \left(\overline{\mathbf{u}}\,\overline{T}\right) = \nabla \cdot \left(\frac{\nu(\alpha)}{\mathrm{Pr}(\alpha)} \nabla \overline{T}\right) + \nabla \cdot \mathbf{J}_{SGS}. \tag{25}$$

The unclosed subgrid-scale stress tensor $\boldsymbol{\tau}_{SGS} = \overline{\mathbf{u}\mathbf{u}} - \overline{\mathbf{u}}\,\overline{\mathbf{u}}$ is modeled using a dynamic one-equation eddy viscosity model proposed by Yoshizawa and Horiuti [11] and Kim and Menon [12]. This SGS model uses a modeled balance equation to simulate the behavior of the subgrid-scale kinetic energy k_{SGS} in which the dynamic procedure of Germano et al. [13] is applied to evaluate all required coefficients dynamically in space and time. The subgrid-scale scalar flux $\mathbf{J}_{SGS}$ [see Eq. (25)] can be considered using a gradient diffusion approach.

2.3 Computational Grid and Boundary Conditions

For the simulation of particulate fouling on structured surfaces, two academical testcases, a smooth channel with a single square cavity and one with a spherical dimple, have been investigated. The computational domain for both testcases is shown in Fig. 2. The origin of the coordinate system is located in the center of the dimple, respectively, the cavity, and is projected onto the lower wall plane, therefore the

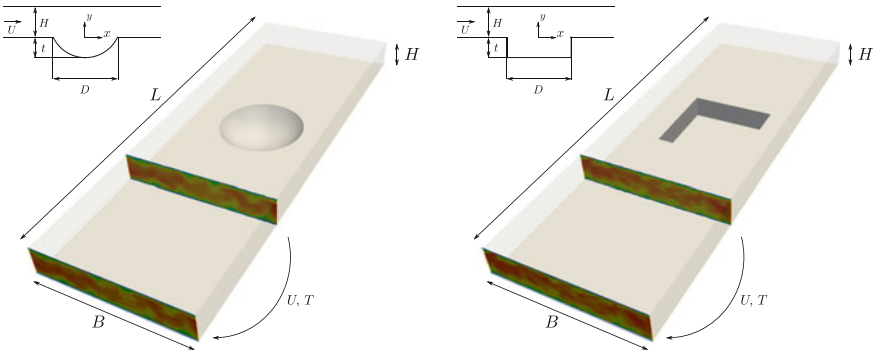


Fig. 2 Computational domain for a smooth channel with a single spherical dimple (left) and a square cavity (right)

lower wall is located at $y/H = 0.0$. The length of the channel is $L = 230$ mm, while channel height H and channel width B are set to $H = 15$ mm and $B = 80$ mm. For the spherical dimple with a sharp edge, a diameter of $D = 46$ mm and a dimple depth $t = 12$ mm are chosen, while the side length of the square cavity equals the dimple diameter D and the cavity depth is likewise set to $t = 12$ mm. Periodic boundary conditions were applied in the spanwise direction, whereas no slip boundary conditions were set at the lower and upper channel walls. Turbulent inlet conditions were produced using a precursor method, which copies the turbulent velocity and temperature field from a plane downstream the channel entrance back onto the inlet. The nondimensionalized form of the temperature

$$T^+ = \frac{T - T_\infty}{T_w - T_\infty} \tag{26}$$

is used, where a constant $T^+ = 1$ is assumed at the lower wall. The molecular Prandtl number was set to be $\mathrm{Pr} = 0.71$, whereas the turbulent Prandtl number Pr_t was 0.9 in all simulations. The Reynolds number based on the averaged bulk velocity U_b and the dimple diameter, respectively the cavity side length D, was equal to $\mathrm{Re}_D = 42{,}000$. To assure grid independence of the obtained results, a series of calculations on different grid resolutions was carried out. Therefore, block-structured curvilinear grids consisting of around 7.8×10^5, 1.6×10^6 and 3.3×10^6 cells were used. In the spanwise and streamwise direction, an equidistant grid spacing is applied, whereas in the wall-normal direction, a homogeneous grid stretching is used to place the first grid node inside of the laminar sublayer at $y^+ \approx 1$. Spherical monodisperse quartz particles (SiO_2) with a diameter of $D_p = 20\,\mu$m are randomly injected within the flow inlet for the fouling simulations. Based on an earlier experimental fouling investigation [14], a total particle mass up to $m_p \approx 5.5$ g/s is chosen to ensure an asymptotic fouling layer growth against a limit value within a few minutes of physical realtime. The estimated volume fraction of the dispersed phase is $\alpha_d < 0.001$, which corresponds to a dilute flow and allows for the negligence of particle collisions [4]. Hence, only two-way coupling is considered during the simulations.

3 Results

3.1 Validation

The functionality and accuracy of our proposed method, as well the Lagrangian Particle Tracking, as the simulation of a fully turbulent flow in a smooth channel with a single spherical dimple at $\mathrm{Re}_D = 42{,}000$ is validated within this section.

3.1.1 Lagrangian Particle Tracking

The Taylor–Green vortex flow is chosen to investigate the performance of the implemented Lagrangian Particle Tracking (LPT). This flow is selected due to the existence of its exact solution of the corresponding instantaneous velocity field and streamfunction as a special case of a two-dimensional time-dependent solution to the Navier–Stokes equations. The two-dimensional Taylor–Green vortex is assumed to be in the x-y-plane, and the flow is uniform in the z-direction. Therefore, the instantaneous local streamfunction can be written as [15]

$$\boldsymbol{\Psi}(x,\ y,\ t) = \frac{\omega_0}{k^2} \cos\left(k_x x\right) \cos\left(k_y y\right) \exp\left(-\mathrm{Re}_0^{-1} k^2 t\right), \tag{27}$$

where the corresponding instantaneous fluid velocity components can be directly derived from Eq. (27)

$$u_x = \frac{\partial \boldsymbol{\Psi}}{\partial y} = -\omega_0 \frac{k_y}{k^2} \cos\left(k_x x\right) \sin\left(k_y y\right) \exp\left(-\mathrm{Re}_0^{-1} k^2 t\right), \tag{28}$$

$$u_y = -\frac{\partial \boldsymbol{\Psi}}{\partial x} = \omega_0 \frac{k_x}{k^2} \sin\left(k_x x\right) \cos\left(k_y y\right) \exp\left(-\mathrm{Re}_0^{-1} k^2 t\right), \tag{29}$$

where ω_0 is the initial vorticity maximum, k_x and k_y are the wave numbers in the x- and y-directions and $k^2 = k_x^2 + k_y^2$. It is assumed that the gravitational acceleration is normal to the flow plane $(x - y)$, so that it does not affect the particle dynamics. The simulations are performed with the following parameters: $\mathrm{Re}_0^{-1} = 0.004$, $k_x = k_y = 1$, $\omega_0 = 2$, x and y are in the range of 0 and 2π. Spherical particles with a constant diameter of $D_p = 0.001$ m were randomly seeded within the flow field.
The primary objective of this testcase is to examine the effects of the so-called momentum (velocity) response time

$$\tau_V = \frac{\rho_p D_p^2}{18 \mu_f} \tag{30}$$

with respect to the particle trajectories, which exemplifies the functionality of the implemented Lagrangian Particle Tracking in a qualitative manner. The momentum response time relates to the time required for a particle to respond to a change in velocity [4]. For the case of nearly massless particles ($\tau_V \approx 0$), the particle trajectories should follow the flow streamlines exactly. For the case of heavier particles ($\tau_V > 0$), their trajectories are expected to deviate from the flow streamlines due to inertial effects. Therefore, only the drag force is considered within the simulations. Fig. 3 compares the particle trajectories after simulating 3.4 s of physical real time for different momentum response times to the streamlines of the two-dimensional Taylor–Green vortex computed from the stream function Eq. (27).

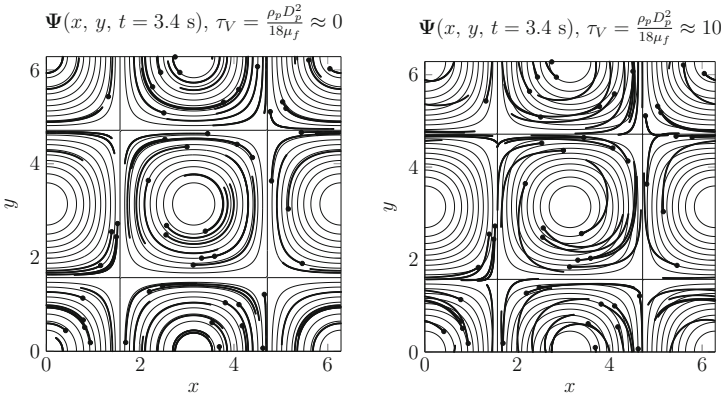


Fig. 3 Two-dimensional Taylor–Green vortex: streamlines (contour lines) and particle trajectories (thick dotted lines) for $\tau_V \approx 0$ (left) and $\tau_V \approx 10$ (right); thick dots represents the initial positions of the randomly seeded particles

It is obvious that the motion of the particles seems to be correctly described by the LPT, because the fluid particles in the case of τ_V follows the streamlines very well. With increasing particle response time τ_V, the particles get their own inertia and they lose the ability to follow the carrier flow. As expected, other numerical results shows that a step-wise increase of the particle response times τ_V leads to a step-wise increased deviation of the particle trajectories from the flow streamlines. The obtained results confirms that the implemented LPT routines are capable of describing the motion of the particles correctly.

3.1.2 Flow in a Smooth Channel with a Single Spherical Dimple

Numerical results for a smooth channel with a single spherical dimple are validated using experimental data published by Terekhov et al. [16] and Turnow et al. [17]. Figure 4 shows the profile of the normalized mean velocity $\langle U \rangle / U_0$ and Reynolds stress $\langle U_{rms} \rangle / U_0$ in the flow direction received from URANS (k-ω-SST model with a fine grid; for comparative purposes only) and LES for three different grid resolutions and from LDA measurements along the y-axis at $x/D = 0.0$ and $z/H = 0.0$ (center of the dimple). The numerical results are obtained for Reynolds number $\mathrm{Re}_D = 42{,}000$ and are normalized by the maximum velocity U_0 in the center of the channel at $y = H/2$. A satisfactory overall agreement of calculated and measured mean velocity profiles has been obtained for all three grid resolutions. The mean velocity profiles from LES and URANS matches well with the measurements in the center of the channel where the maximum flow velocity occurs, and even in the upper near wall region. However, slight deviations from the measured profiles can be registered for both methods and all grid resolutions inside of the spherical dimple within the distinct recirculation zone. Nevertheless, since URANS and LES results are in good

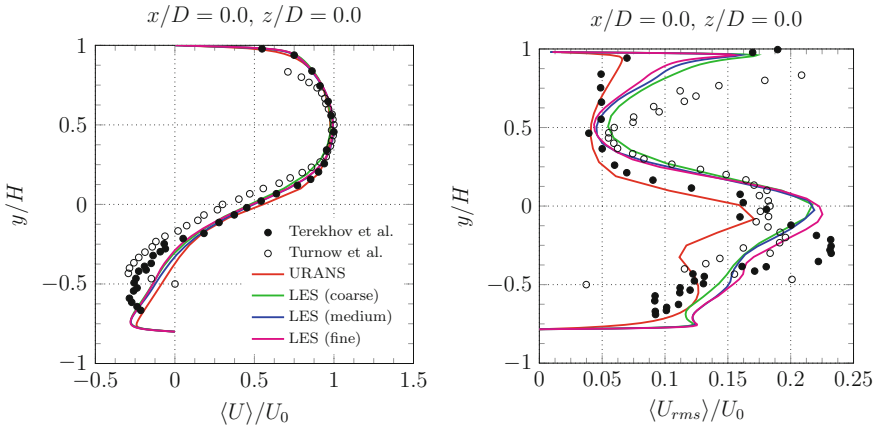


Fig. 4 Mean velocity and Reynolds stress profiles obtained from URANS and LES in comparison with experiments by Terekhov et al. and Turnow et al.

agreement in this region, the likeliest reason for the discrepancy between measurements and calculations might be LDA measurement problems in close proximity to the wall [17]. From the mean velocity profiles, one can observe, that the strongest velocity gradients arises at the level of the lower channel wall $y/H = 0.0$. The instabilities of the shear layer within this region result in strong vortices, and therefore in high Reynolds stresses, which can be observed in all Reynolds stress profiles. Unlike the mean velocity profiles, the level of the Reynolds stresses and, furthermore, the location of the maximum turbulent fluctuations measured in experiments can only be gained using LES. A deviation between the measured and calculated locations of the maximum Reynolds stresses are notable within the center of the dimple at position $x/D = 0.0$. The weakness of URANS is clearly visible in the near wall region and the level of the lower wall, where the magnitude of the turbulent fluctuations can not be captured. Due to the significant importance of the resolved Reynolds stresses for calculating the resuspension rate of deposited particles (see Sect. 2.1.2), URANS seems to be inappropriate for further investigations.

Probably the most important feature of the investigated spherical dimple with a dimple depth to dimple diameter ratio of $t/D = 0.261$ can be observed from the phase averaged streamline pattern given in Fig. 5. The streamlines shows unsteady asymmetrical monocore vortex structures inside the dimple, which switches their orientation arbitrarily from $\alpha = -45°$ to $\alpha = +45°$ with respect to the main flow direction. The existence of long period self-sustained oscillations [17] within the dimple flow could be investigated and approved experimentally (see, for example, [16]), as numerically using highly resolved LES [17] for dimple depth to dimple diameter ratios of $t/D = 0.261$ and larger. In contrast to experimental observations and LES results, the asymmetric vortex structures obtained from URANS are steady and predict only one of the two extreme vortex positions ($\alpha = \pm 45$) in the time-

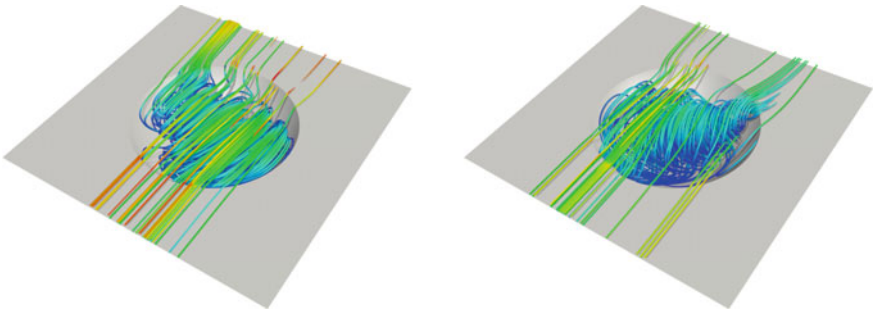


Fig. 5 Different orientations ($\alpha = \pm 45°$) of the oscillating vortex structures inside the spherical dimple for $\mathrm{Re}_D = 42{,}000$

averaged flow pattern, which switch in reality almost periodically in reality. It is assumed that the self-sustained oscillations and periodic outbursts due to unsteady asymmetric vortex structures could endorse a possible self-cleaning process inside the spherical dimple and at the lower channel wall downstream of the trailing edge. Thus, LES is chosen to simulate the particulate fouling and to investigate its influence on heat transfer and friction loss for a smooth channel with a spherical dimple or square cavity.

3.2 Particulate Fouling on Structured Heat Transfer Surfaces

To investigate the influence of particulate fouling on the friction/pressure loss and heat transfer, a series of LES using injected particle masses up to $m_p \approx 5.5$ g/s are carried out for a smooth channel with a single spherical dimple or square cavity ($t/D = 0.261$, $\mathrm{Re}_D = 42{,}000$). Due to the results given in Fig. 4, a medium grid resolutions with around 1.6×10^6 cells seems to be sufficient to capture the mean velocity profiles, as well as the Reynolds stresses accurate enough and is chosen for the simulation of particulate fouling. The pressure loss due to friction is expressed in terms of the skin friction factor

$$C_f = \frac{\tau_w}{\frac{1}{2}\rho_f U_b^2}, \tag{31}$$

with the shear stress τ_w, the density of the fluid ρ_f and the bulk velocity U_b of the fluid. It should be noted that Eq. (31) has to be extended for further investigations to allow for consideration of the form drag due to structured surfaces. The heat transfer is evaluated using the Nusselt number

$$\mathrm{Nu} = \frac{hL}{\lambda}, \tag{32}$$

where h is the convective heat transfer coefficient of the flow, L is the characteristic length (L is set to $2H$ due to periodic boundary conditions in the streamwise direction) and λ is the thermal conductivity of the fluid. The Nusselt number relates the total heat transfer to the conductive heat transfer. The friction coefficient and Nusselt number received from the fouling simulations are divided by the ones obtained from a turbulent channel flow without particulate fouling, which directly expresses the increase or decrease of pressure loss and heat transfer. Therefore, the friction coefficient C_{f0} and the Nusselt number Nu_0 of the smooth channel are determined using the correlations of Petukhov and Gnielinski for turbulent channel flows $\left(1500 < \mathrm{Re}_H < 2.5 \times 10^6\right)$:

$$C_{f0} = \left(1.58 \ln\left(\mathrm{Re}_H\right) - 2.185\right)^{-2}, \tag{33}$$

$$\mathrm{Nu}_0 = \frac{\left(C_{f0}/2\right)\left(\mathrm{Re}_H - 500\right)\mathrm{Pr}}{1 + 12.7\left(C_{f0}/2\right)^{1/2}\left(\mathrm{Pr}^{2/3} - 1\right)}. \tag{34}$$

The original correlations (see [18]) are in terms of the Reynolds number Re_{D_h} based on the hydraulic diameter, but they are used in a rewritten form assuming $D_h = 2H$ for a smooth, infinitely wide channel. Figure 6 shows the space- and time-averaged friction coefficient $\overline{\overline{C_f}}/C_{f0}$ and Nusselt number $\overline{\overline{\mathrm{Nu}}}/\mathrm{Nu}_0$ for a spherical dimple and square cavity after 30 s of physical real time, whereby no particles are considered (I) or a particle mass of $m_p \approx 2.2$ g/s (II) or $m_p \approx 5.5$ g/s (III) is injected. It can be seen, that the pressure loss is increased by around 29% in the case of the clean spherical dimple and approximately 61% for the clean square cavity compared to the turbulent channel flow, whereas the heat transfer is enhanced by circa 32% for the spherical dimple and 27% in the case of the square cavity. The thermo-

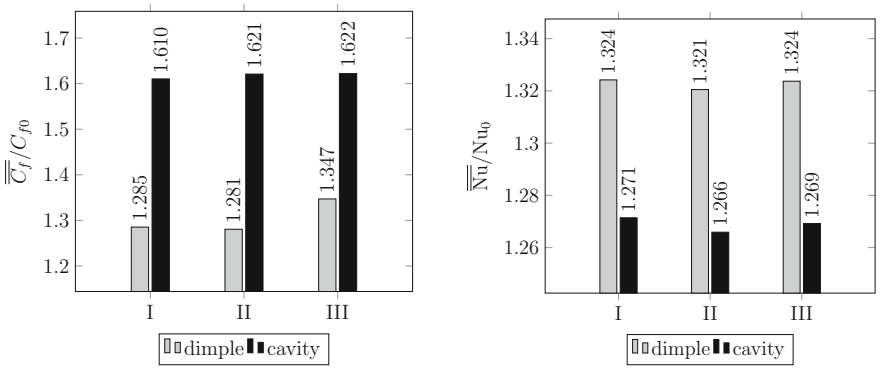


Fig. 6 Comparison of the space- and time-averaged friction coefficient $\overline{\overline{C_f}}/C_{f0}$ and Nusselt number $\overline{\overline{\mathrm{Nu}}}/\mathrm{Nu}_0$ for a spherical dimple and square cavity after 30 s of physical real time: (I) clean surface (no particle injection), (II) particle mass injection: $m_p \approx 2.2$ g/s, and (III) particle mass injection: $m_p \approx 5.5$ g/s

hydraulic efficiency $\overline{\overline{\mathrm{Nu}}}/\mathrm{Nu}_0/(\overline{\overline{C_f}}/C_{f0})^{1/3}$ achieved for the dimple is 1.22, and 1.08 for the cavity. Thus, the clean spherical dimple shows great advantages compared to the clean square cavity. In contrast, the change of pressure loss and heat transfer due to particulate fouling after 30 s of physical real time is less unambiguous. The increase of the pressure loss $\overline{\overline{C_f}}/C_{f0}$ ranges between 0.4 % (II) and 6% (III) for the spherical dimple and between 1.1% (II) and 0.1% (III) in the case of the square cavity. The decrease in the heat transfer is even smaller, varying between 0.3 and 0.6%, whereas the spherical dimple shows a slightly better fouling performance. Finally, Fig. 7 presents the total fouling layer height h_f and the total height h_r of the removed fouling layer for the spherical dimple and square cavity after 30 s of simulated physical real time for an injected particle mass of $m_p \approx 2.2$ g/s. A relatively high particle deposition can be observed within the recirculation zone of the spherical dimple, whereas no particulate fouling is detected in the lee side of the dimple where the reattachment point lies. Due to the switching of asymmetric vortex structures and vortex outbursts (see Sect. 3.1.2), a high fouling removal rate is obtained within the lee side, and additionally downstream of the dimple's trailing edge. Moreover, due to the flow acceleration in front of the dimple, no fouling can be observed in the area

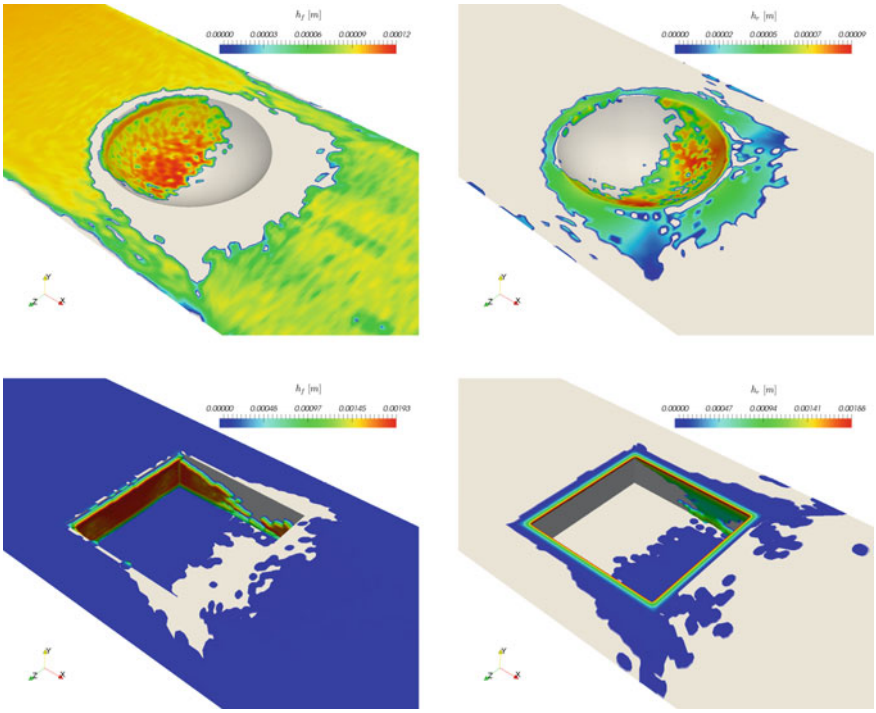


Fig. 7 Height of the settled h_f and removed h_r fouling layer after 30 s of physical real time for an injected particle mass $m_p \approx 2.2$ g/s: spherical dimple (left), square cavity (right)

of the dimple's front edge. The highest removal rates are obtained at both extreme positions of the asymmetric vortex structures ($\alpha = \pm 45$). In the case of the square cavity, the highest particle deposition rates are gained at the vertical front side and side faces of the cavity due to adhesion, whereas the clean area downstream of the cavity's trailing edge is significantly smaller compared to the spherical dimple. More detailed investigations of the fouling mechanisms (deposition and removal) under consideration of local vortex structures using vortex identification methods will be a primary part of the next work.

4 Conclusion

A new multiphase method for the numerical simulation of particulate fouling processes on structured heat transfer surfaces is introduced and described in detail. A verification and validation of the LPT is carried out using the two-dimensional Taylor–Green vortex. The investigation of the flow inside a smooth channel with a spherical dimple ($t/D = 0.261$, $\mathrm{Re}_D = 42{,}000$) confirms unsteady asymmetric vortex structures inside the dimple in the case of LES, which are primarily responsible for a self-cleaning process, respectively for the better fouling performance, in comparison to the square cavity. One of the main advantages of our new multiphase approach is the fairly low computational effort due to phase conversion and particle deletion. In addition to that, solution of an advection/transport equation for the fouling phase and costly remeshing procedures are not required, which allows for the usage of the proposed method for a general prediction of particulate fouling on structured heat transfer surfaces.

Acknowledgements The authors would like to thank the German Research Foundation (DFG, grant KO 3394/10-1 and INST 264/113-1 FUGG) and the North-German Supercomputing Alliance (HLRN) for supporting this work.

References

1. P. Ligrani, International Journal of Rotating Machinery **2013**, 32 (2013)
2. H. Mueller-Steinhagen, Heat Transfer Eng. **32**, 1 (2013)
3. M. Sommerfeld, *Particle Motion in Fluids - VDI Heat Atlas* (Springer, 2010)
4. T.C. Crow, J.D. Schwarzkopf, M. Sommerfeld, Y. Tsuji, *Multiphase flows with droplets and particles*, 2nd edn. (CRC Press, Taylor & Francis, 2011)
5. A. Putnam, ARS Journal **31**, 1467 (1961)
6. P.G. Saffman, J. Fluid Mech. **22**, 385 (1965)
7. P.G. Saffman, J. Fluid Mech. **31**, 624 (1968)
8. R. Mei, Int. J. Multiphase Flow **18**, 145 (1992)
9. F. Loeffler, W. Muhr, Chem. Ing. Tech. **44**, 510 (1972)
10. E. Heinl, M. Bohnet, Powder Technol. **159**, 95 (2005)
11. A. Yoshizawa, K. Horiuti, J. Phys. Soc. Jpn. **54**, 2834 (1985)
12. W. Kim, S. Menon, in *AIAA, Aerospace Sciences Meeting and Exhibit, 33 rd, Reno, NV* (1995)

13. M. Germano, U. Piomelli, P. Moin, W.H. Cabot, Phys. Fluids **3**, 1760 (1991)
14. R. Bloechl, H. Mueller-Steinhagen, The Canadian Journal of Chemical Engineering **68**, 585 (1990)
15. S. Wetchagarun, A numerical study of turbulent two-phase flows. Ph.D. thesis, University of Washington (2008)
16. V.I. Terekhov, S.V. Kalinina, Y.M. Mshvidobadse, Journal of Enhanced Heat Transfer **4**, 131 (1997)
17. J. Turnow, N. Kornev, S. Isaev, E. Hassel, Heat Mass Transfer. **47**, 301 (2011)
18. M.A. Elyyan, A. Rozati, D.K. Tafti, Int. J. Heat Mass Transfer **51**, 2950 (2007)

solidificationMeltingSource: A Built-in fvOption in OpenFOAM® for Simulating Isothermal Solidification

Mahdi Torabi Rad

Abstract In this chapter, I introduce, document, and verify solidificationMeltingSource: a built-in fvOption in OpenFOAM® for simulating isothermal solidification. The main challenge in simulating isothermal solidification is the incorporation of movements of the solidification front. To overcome this challenge, solidificationMeltingSource adds source terms to the momentum and energy equations. First, I rigorously derive the equations for these source terms and outline their implementation in the source code. Then, I verify solidificationMeltingSource by simulating a well-known numerical benchmark for isothermal solidification. Finally, I end the chapter by suggesting possible future extensions for solidificationMeltingSource.

Keywords Phase change · fvOption · Isothermal solidification solidificationMeltingSource

1 Introduction

1.1 fvOptions

fvOptions is a flexible framework in OpenFOAM® that allows users to add source terms to equations at run time [1]. fvOptions is easy to use since users do not have to modify source code. Some of the implemented fvOptions are: (1) explicit porosity source; to simulate flow in a domain with porous subzones; (2) MRF source: to simulate flow on both stationary and rotating frames; (3) fixed temperature constraint: to fix temperature at given locations to constant or time-varying values. The reader is referred to reference [1] for a complete list of available fvOptions and example syntaxes on how to use them.

solidificationMeltingSource [2] is a fvOption that can be used to simulate isothermal phase-change (solidification and melting) problems such as windshield defrost-

M. Torabi Rad (✉)
Seamans Center, 3131, Iowa City, IA 52242, USA
e-mail: mahdi-torabirad@uiowa.edu

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®* ,
https://doi.org/10.1007/978-3-319-60846-4_32

ing or solidification of phase-change materials (PCM). solidificationMeltingSource adds source terms to: (1) the momentum equation, to account for the drag force due to the presence of a solid in a liquid, and (2) the energy equation to account for latent heat release during the phase change.

1.2 Background on Isothermal Solidification

Pure materials solidify at a single temperature, whereas alloys solidify within a range of temperatures. This chapter focuses only on solidification of pure materials: isothermal solidification. Figure 1 shows a schematic of a system under isothermal solidification. Cooling is from the left. The domain consists of a solid region on the left, with $T < T_f$, where T_f is the melting temperature, and a liquid region, with $T > T_f$, on the right. The two regions are separated by the sharp and moving solidification front located at $x = x^*$. For solidification to continue, latent heat released at the solidification front has to be dissipated by net conduction away from the solidification front. Heat balance at the solidification front implies

$$-\rho L_f \frac{\mathrm{d}x^*}{\mathrm{d}t} = \left(-k_s \frac{\partial T}{\partial x}\right)^* - \left(-k_l \frac{\partial T}{\partial x}\right)^*. \tag{1}$$

In solidification literature, Eq. (1) is typically referred to as the classical Stefan condition [3]. The term on the left-hand side represents the latent heat released at $x = x^*$ (due to solidification). On the right-hand side, the first and second terms are the heat fluxes through the solid and liquid, respectively. The superscript * indicates that the fluxes are evaluated at $x = x^*$.

The main challenge in simulating isothermal solidification problems is to incorporate movements of the solidification front. Numerical methods for overcoming this challenge can be categorized as: (1) moving boundary methods, and (2) enthalpy methods. In the moving boundary methods, the location of the solidification front is tracked *explicitly*, and two different energy equations are solved, the first one for the solid side and the second one for the liquid side. In the enthalpy methods, there is no need for explicit tracking of the solidification front. Instead, the energy equation is written in a form that is valid on both (solid and liquid) sides of the solidification front. The latter methods are numerically easier to implement. However, one needs to modify the local single-phase equations so that they are valid on both sides of the solidification front. solidificationMeltingSource uses an enthalpy method, which is discussed in this chapter.

The objective of the present chapter is to introduce, document, and verify solidificationMeltingSource. Equations implemented in solidificationMeltingSource are rigorously derived and their implementation in the source code is outlined. solidificationMeltingSource is verified through the simulation of a numerical benchmark for isothermal solidification and the results are compared with data available in the literature.

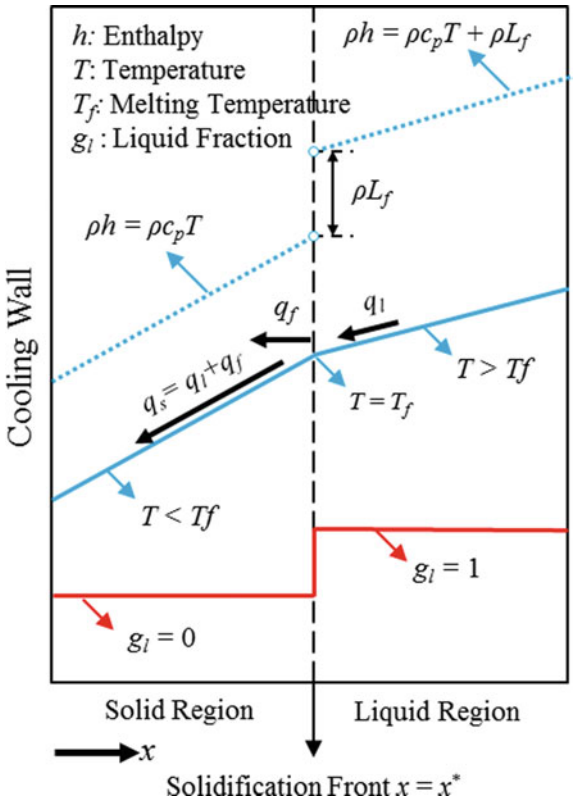


Fig. 1 Schematic of a system under isothermal solidification. The red, solid blue, and dashed blue lines show solid fraction, temperature, and enthalpy, respectively. The solid region on the left, with $T < T_f$ and $g_l = 0$, is separated from the liquid region on the right, with $T > T_f$ and $g_l = 1$, by the solidification front (dashed line). Heat fluxes are represented by arrows

2 Governing Equations

Equations governing isothermal solidification are listed below. These equations are taken from Voller and Prakesh [4].

2.1 Conservation Equations

2.1.1 Continuity

$$\nabla \cdot \boldsymbol{v} = 0, \tag{2}$$

where $\boldsymbol{v}$ is the flow velocity.

2.1.2 Momentum

$$\rho_0 \frac{\partial \boldsymbol{v}}{\partial t} + \rho_0 \nabla \cdot (\boldsymbol{v}\boldsymbol{v}) = -\nabla p + \mu_0 \nabla^2 \boldsymbol{v} + \boldsymbol{S}_b + \boldsymbol{S}_d, \tag{3}$$

where ρ_0 and μ_0 are the density and the dynamic viscosity (both taken to be constant here), $\boldsymbol{S}_b$ is the buoyancy source term, and $\boldsymbol{S}_d$ is a source term added to force the velocities in the solid region to zero. These source terms will be derived in the next subsection.

2.1.3 Energy

$$\rho c_p \frac{\partial T}{\partial t} + \rho c_p \nabla \cdot (\boldsymbol{v} T) = \nabla \cdot (k \nabla T) - S_h, \tag{4}$$

where c_p is the specific heat and k is the thermal conductivity, and S_h is a source term that accounts for the latent heat release during solidification. The equation for this source term will be also derived in the next section.

2.2 Derivation of the Equations for Source Terms

The momentum and energy equations, listed in the previous subsection, have source terms $\boldsymbol{S}_b$, $\boldsymbol{S}_d$, and S_h. The equations for these source terms are derived below.

2.2.1 Momentum Source Terms: $\boldsymbol{S_b}$ and $\boldsymbol{S_d}$

Equation (3) has two source terms: $\boldsymbol{S}_b$ and $\boldsymbol{S}_d$. The first one is the buoyancy source term, which is given by the Boussinesq approximation:

$$\boldsymbol{S}_b = \rho_0 \mathbf{g} \beta_T (T - T_{\text{ref}}), \tag{5}$$

where $\mathbf{g} = -9.8\hat{\mathbf{j}}$ is the acceleration due to gravity, β_T is the thermal expansion coefficient, and T_{ref} is the reference temperature.

The second source term in Eq. (3) is added to make sure that the velocities in the solid region will become zero. An acceptable relation for $\boldsymbol{S}_d$ should satisfy two conditions: (1) in the liquid region, $\boldsymbol{S}_d$ should be zero so that Eq. (3) reduces to the normal single-phase Navier–Stokes equations, and (2) in the solid region, $\boldsymbol{S}_d$ should dominate all other terms in the momentum equation and should result in $\boldsymbol{v} = 0$. Motivated by the Darcy law for flow through porous media, Voller and Prakesh [4] have suggested

$$\boldsymbol{S}_d = A\boldsymbol{v}, \tag{6}$$

where A is inversely proportional to permeability K:

$$A \sim \frac{1}{K}. \tag{7}$$

Permeability K is related to liquid fraction g_l through the well-known Carman–Koseny equation [5]:

$$K \sim \frac{g_l^3}{(1-g_l)^2}. \tag{8}$$

The reader should note that, for isothermal solidification, permeability has no physical significance; it is introduced here only as a numerical technique to force the velocities in the solid region to zero. In fact, Voller and Prakesh [4] stated that $A \sim 1/K$, in Eq. (7), is not the only correct relation. Any relation that gives $A = 0$ in the liquid region and a relatively high value for A in the solid region will be equally correct.

Now, after substituting Eqs. (7) and (8) into Eq. (5), we get the final equation for $\boldsymbol{S}_d$ as

$$\boldsymbol{S}_d = -C\frac{(1-g_l)^2}{g_l^3}\boldsymbol{v}. \tag{9}$$

It is apparent from this equation that in the liquid region (where $g_l = 1$), $\boldsymbol{S}_d$ will be zero; in the solid region, where g_l is a sufficiently small number (on the order of 10^{-6}), $\boldsymbol{S}_d$ will be high enough to dominate all other terms in the momentum equation. Therefore, the momentum equation will reduce to $\boldsymbol{S}_d = 0$, and, consequently, $\boldsymbol{v} = 0$.

2.2.2 Energy Source Term: S_h

The source term in the energy equation, S_h, represents the latent heat released during solidification. The equation for S_h is derived below.

For a system with phase change, the energy equation reads as

$$\frac{\partial}{\partial t}(\rho h_{\text{tot}}) + \nabla \cdot (\rho \boldsymbol{v} h_{\text{tot}}) = \nabla \cdot (k \nabla T), \tag{10}$$

where h_{tot} is the total enthalpy. In the solid region, $h_{\text{tot}} = c_p T$; in the liquid region, $h_{\text{tot}} = c_p T + L_f$. To have a single relation valid in both solid and liquid sides, we write

$$h_{\text{tot}} = c_p T + g_l L_f, \tag{11}$$

where g_l is the liquid fraction equal to zero in the solid region and equal to one in the liquid region.

Now, if we substitute Eq. (11) into (10) and subtract Eq. (4), we get

$$S_h = \rho_0 L_f \left[\frac{\partial g_l}{\partial t} + \nabla \cdot (\boldsymbol{v} g_l) \right], \tag{12}$$

which is the final equation for the energy source term S_h.

Finally, we need a relation that will transform $g_l = 1$ into $g_l = 0$ as temperatures become lower than the melting temperature, i.e., $T < T_f$. Discretizing the energy equation explicitly, and taking the temperature derivative of both sides (i.e.,$\partial/\partial T$), will give us

$$\frac{\partial g_l}{\partial T} = -\frac{c_p}{L_f}. \tag{13}$$

From Eq. (13), we can write

$$\begin{aligned} g_l^{n+1} &= g_l^n + \frac{\gamma c_p}{L_f}\left(T - T_f\right) \\ g_l^{n+1} &= \max\left[0, \min\left(1, g_l^{n+1}\right)\right], \end{aligned} \tag{14}$$

where the superscripts $n + 1$ and n refer to the present and last iteration levels, respectively, and γ is an under-relaxation factor (of order 1). Equation (14) updates the liquid fractions as follows: initially, we have $g_l^n = 1$ and $T > T_f$; therefore, the first line in Eq. (14) will give us $g_l^{n+1} > 1$, which, in the second line, will be reset back to $g_l^{n+1} = 1$. For temperatures slightly lower than T_f, $\gamma c_p\left(T - T_f\right)/L_f$ will be negative, and therefore Eq. (14) will result in $g_l^{n+1} < 1$: the liquid fraction will start to decrease. The liquid fraction will keep decreasing until it reaches zero, and after that, it will no longer change.

3 Implementation in solidificationMeltingSource

Listings (1, 2, and 3) are code pieces from the solidificationMeltingSource source code, in which implementation of the source terms outlined in the previous section is shown. In listing (1), lines 337 and 338 are the implementations of Eq. (9) and Eq. (5), respectively. In listing (2), line 54 is the implementation of Eq. (12). In listing (3), line 216 is the implementation of Eq. (14).

```
337         scalar S = -Cu_*sqr(1.0 - alpha1c)/(pow3(alpha1c) + q_);
338         vector Sb = rhoRef_*g*beta_*deltaT_[i];
339
340         Sp[cellI] += Vc*S;
341         Su[cellI] += Vc*Sb;
```

Listing (1): Code piece showing how momentum source terms are added.

```
50      if (eqn.psi().dimensions() == dimTemperature)
51      {
52          // isothermal phase change - only include time derivative
53          // eqn -= L/Cp*(fvc::ddt(rho, alpha1_) + fvc::div(phi, alpha1_));
54          eqn -= L/Cp*(fvc::ddt(rho, alpha1_));
55      }
```

Listing (2): Code piece showing how an energy source term is added.

```
214           scalar Tc = T[cellI];
215           scalar Cpc = Cp[cellI];
216           scalar alpha1New = alpha1_[cellI] + relax_*Cpc*(Tc - Tmelt_)/L_;
```

Listing (3): Code piece showing how a liquid fraction, represented by alpha in the code, is updated.

4 Problem Statement and Simulation Setup

The isothermal solidification problem studied here is the benchmark introduced in Voller and Prakesh [4] and is sketched in Fig. (2). It consists of a square cavity initially filled with liquid above its freezing temperature. The cavity cools from the left wall, while the top and bottom walls are thermally isolated. Solidification starts from the left, with the solidification front moving to the right as time proceeds.

The OpenFOAM® solver that is used is buoyantBoussinesqPimpleFoam. The mesh size and time step are 2.5 cm and 1 s, respectively. In the fvOptions file, we have $T_{\text{melt}} = 0$, $L = 5$, thermoMode = lookup, beta = 0, and rhoRef = 1. Note that the value of beta in the fvOptions file is set to zero since buoyant Boussinesq Pimple Foam already accounts for the buoyancy source term. The properties and boundary conditions are taken from Voller and Prakesh [4] and, due to space limitations, are not listed here.

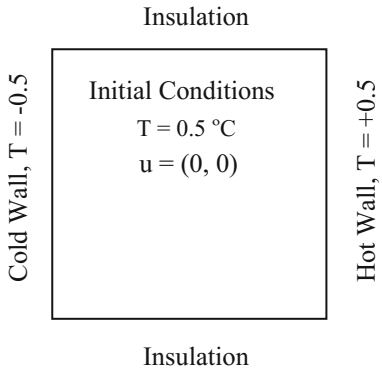


Fig. 2 Schematic of the benchmark problem introduced in Voller and Prakesh [4] and simulated here

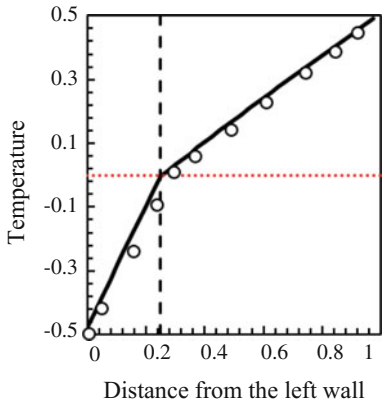


Fig. 3 Temperature distributions at $t = 500$ s calculated in the present study (solid line) and reported in Voller et al. [6]. The dashed line is the solidification front and the red dotted line shows the melting temperature ($T_f = 0$)

5 Results

In the absence of flow, the results are compared with data in Voller et al. [6]. Figure (3) shows a comparison of temperature profiles reported in Voller et al. [6] (markers), with the temperatures calculated in the present study (solid line at $t = 500$ s). The dashed line is the solidification front and the dotted line shows the melting temperature ($T_f = 0$). The two temperatures are in good agreement, which verifies solidificationMeltingSource.

In the presence of flow, results are shown in Fig. (4). This figure shows the evolution of the solidification over time. Columns represent different times. In the top row, color represents the solid fraction $g_s (= 1 - g_l)$ and the vectors represent the liquid velocity. In the bottom row, color represents temperature, and the white line, which is the solidification front (isoline $g_s = 0.5$), is superimposed from the solid fraction contours at the top. One can easily notice that this line lies on the freezing temperature, i.e., $T = 0$. Furthermore, regions with $T > 0$ in the bottom row have $g_s = 0$ in the top row; these regions are fully liquid. Similarly, regions with $T < 0$ in the bottom row have $g_s = 1$ in the top row; these regions are fully solid. These agreements verify solidificationMeltingSource in the presence of flow.

6 Conclusions

In this chapter, I introduced, documented and verified solidificationMeltingSource, which is a built-in fvOption in OpenFOAM® for simulating isothermal solidification problems. The main challenge in simulating these problems is the incorporation of movements of the solidification front. To account for these movements, solidificationMeltingSource adds source terms to the momentum and energy equations. I rigorously derived the equations for these source terms and outlined their implemen-

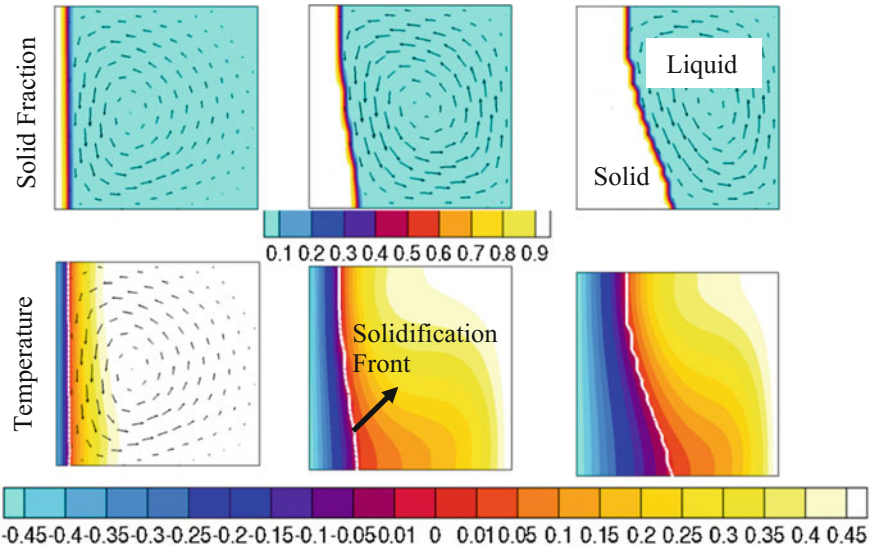


Fig. 4 Time evolution of the solid fraction (top row) and temperature (bottom row) during solidification. Left, middle, and right columns show different times. The vectors in the top row and the white line in the bottom row represent liquid velocity and the solidification front (isoline $g_s = 0.5$), respectively

tation in the source code. I simulated a benchmark for isothermal solidification and compared the results with the data in the literature. The agreement was found to be good, and therefore solidificationMeltingSource is verified.

Clearly, the solidification problem investigated here was a highly simplified one. This problem was chosen because realistic and more interesting solidification problems, such as the solidification of alloys, can only be represented by multiphase/-scale mathematical models [7], which need to be handled through complex numerical techniques [7–10] and cannot be numerically solved solely by adding source terms to the equations. The material presented in this chapter should be viewed as a starting guide for FOAMers interested in solidification problems and, more importantly, as an attempt to help with future efforts to extend OpenFOAM®'s built-in capabilities in simulating solidification problems. As an example of these extensions, implementation of algebraic lever and Scheil solidification models [3] in solidificationMeltingSource is suggested.

References

1. http://openfoam.org/release/2-2-0/fv-options/, OpenFOAM®, 2016. Accessed 15 Oct 2016
2. https://github.com/OpenFOAM/OpenFOAM-dev/blob/master/src/fvOptions/sources/derived/solidificationMeltingSource/solidificationMeltingSource Accessed 15 Oct 2016

3. Dantzig J A, Rappaz M (2009) Solidification. CRC Press, Lausanne
4. Voller VR, Prakash C (1987) A fixed grid numerical modeling methodology for convection-diffusion mushy region phase-change problems. International Journal of Heat and Mass Transfer 30:1709–1719
5. Torabi Rad M, Kotas P, Beckermann C (2013) Rayleigh Number Criterion for Formation of A-Segregates in Steel Castings and Ingots. Metall. Mater. Trans. A 44:4266–4281
6. Voller VR, Cross M, Markatos NC (1987) An enthalpy method for convection/diffusion phase change. International Journal of Numerical Methods in Engineering 24:271–284
7. Torabi Rad M, Beckermann C (2016) Validation of a Model for the Columnar to Equiaxed Transition with Melt Convection. In: Nastac L et al. (eds) CFD Modeling and Simulation in Materials Processing 2016, TMS, John Wiley & Sons Inc., Hoboken, NJ, pp. 85–92
8. Mohaghegh F, Udaykumar H (2016) Comparison of sharp and smoothed interface methods for simulation of particulate flows I: Fluid structure interaction for moderate Reynolds numbers. Computers and Fluids, 140, 39–58
9. Mohaghegh F, Udaykumar H (2016) Comparison of sharp and smoothed interface methods for simulation of particulate flows II: Fluid structure interaction for moderate Reynolds numbers. Computers and Fluids, 143, 103–119
10. Mohaghegh F, Mousel J, Udaykumar H (2014) Comparison of sharp interface and smoothed profile methods for laminar flow analysis over stationary and moving boundaries. Paper presented at the American Society of Mechanical Engineers, Fluids Engineering Division (Publication) FEDSM, 1A https://doi.org/10.1115/fedsm2014-21171

Study of OpenFOAM® Efficiency for Solving Fluid–Structure Interaction Problems

Matvey Kraposhin, Ksenia Kuzmina, Ilia Marchevsky and Valeria Puzikova

Abstract In the present research, the well-known test FSI problem of wind resonance phenomenon simulation for a circular cylinder is considered. It is well-investigated, both experimentally and numerically (Chen et al. in Phys Fluids 2011, [3]), for a wide range of parameters: Reynolds number, airfoil surface roughness, incident flow turbulence, etc. In this research, the simplest case is considered, in which the roughness influence is neglected and the incident flow is assumed to be laminar. Several numerical codes, both commercial and open source, can be used for simulating airfoil oscillations in the flow. Four numerical methods and the corresponding open-source codes are considered: the finite volume method with deformable mesh in `OpenFOAM`®; the particle finite element method with deformable mesh in the `Kratos` software; the meshfree Lagrangian vortex element method; and the LS-STAG immersed boundary method. The last two methods are implemented as in-house numerical codes. A comparison is carried out for the efficiency analysis of these methods and their implementations. It is shown that using `OpenFOAM`® is preferable for numerical simulations with FSI problems similar to the ones presented here, in which the investigation of system behavior within a wide range of parameters is required.

M. Kraposhin
Institute for System Programming of the Russian Academy of Sciences,
Alexander Solzhenitsyn st., 25., 109004 Moscow, Russia
e-mail: m.kraposhin@ispras.ru

K. Kuzmina · I. Marchevsky (✉) · V. Puzikova
Bauman Moscow State Technical University,
2-nd Baumanskaya st., 5., 105005 Moscow, Russia
e-mail: iliamarchevsky@mail.ru

K. Kuzmina
e-mail: kuz-ksen-serg@yandex.ru

V. Puzikova
e-mail: valeria.puzikova@gmail.com

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM*®,
https://doi.org/10.1007/978-3-319-60846-4_33

1 Introduction

In a number of engineering applications, bodies are immersed in a gas or fluid flow and exposed to aerohydrodynamic loads. Fully coupled 3D fluid–structure interaction (FSI) problems are extremely complicated, both from the mathematical and computational points of view. In many practical cases, the average density of the immersed body is higher than the density of the flow, thus it is possible to apply a well-known "splitting" approach, in which a single time step is divided into at least two substeps. During the first substep, a semi-implicit scheme, for which body motion parameters are assumed to be known, is used while simulating the flow around the body. During the second substep, an explicit scheme is used for body motion simulation under known hydrodynamic loads.

In practice, we often deal with extruded structures (with high elongation), so as a rough approximation, we consider this to be a 2D problem of interaction between the flow and the corresponding airfoil. This approach is particularly applicable when the airfoil cross section has angle points (points, at which the camber line of the airfoil loses its smoothness) and sharp edges. This approach is also accurate enough for FSI problems with bluff bodies that have smooth cross-sectional shapes, for example, flows around cylindrical rods, or pipes. Such bodies oscillate under von-Karman vortex shedding, and frequency-lock phenomenon takes place, which leads to similar flows around different cross sections.

In order to compare different numerical methods and computational codes for analyzing FSI problems, a simple test case considering the resonance of a circular cylinder inside the flow is proposed (Fig. 1).

The most interesting case of this phenomenon corresponds to the situation in which the eigenfrequency of the system is close to the von-Karman vortex shedding frequency. This phenomenon has been well-investigated, both experimentally and numerically, for a wide range of parameters: Reynolds number, airfoil surface roughness, incident flow turbulence, etc.

We consider the simplest case, in which we neglect the influence of surface roughness and the incident flow is assumed to be laminar. Four numerical methods and the corresponding open-source codes have been used: the finite volume method

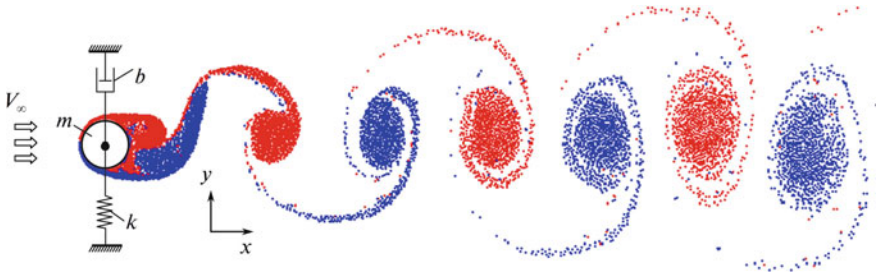


Fig. 1 Circular airfoil in a flow under viscoelastic constraints with a vortex wake behind it

(FVM) [9] with deformable mesh in `OpenFOAM®` [26, 30]; the particle finite element method (PFEM) [10, 11] with deformable mesh in the `Kratos` software [16]; the meshfree Lagrangian vortex element method (VEM) [5, 22]; and the level-set staggered mesh immersed boundary method (LS-STAG) [4]. The VEM and LS-STAG methods are implemented as in-house numerical codes.

2 Governing Equations

As mentioned before, the problem is considered to be a 2D unsteady case in which the flow around a cylindrical airfoil is viscous and incompressible. The continuity and momentum equations are as follows:

$$\nabla \cdot V = 0, \tag{1}$$

$$\frac{\partial V}{\partial t} + (V \cdot \nabla)V = -\nabla p + \frac{1}{\mathrm{Re}}\Delta V. \tag{2}$$

Here, $V = V(x, y, t) = u \cdot e_x + v \cdot e_y$ is the dimensionless velocity and $p = p(x, y, t)$ is the dimensionless pressure. The boundary conditions for the velocity field are defined as

$$V\big|_{\text{inlet}} = V_\infty, \qquad V\big|_{\text{airfoil}} = V^{\text{ib}}, \qquad \frac{\partial V}{\partial n}\Big|_{\text{outlet}} = 0. \tag{3}$$

Here, V^{ib} is the immersed boundary velocity. The airfoil is assumed to be rigid.

In order to simulate the wind resonance phenomenon, we consider one degree of freedom motion of the circular airfoil of diameter D across the stream. Constraint of the airfoil motion is assumed to be of the Kelvin–Voigt type (linear viscoelastic, Fig. 1) described by the following ordinary differential equation:

$$m\ddot{y} + b\dot{y} + ky = F_y. \tag{4}$$

Here, m is the airfoil mass, b is a small damping factor, k is the elasticity coefficient of the constraint, F_y is the lift force, and y is the deviation from the equilibrium state. The natural frequency of the system $\omega \approx \sqrt{k/m}$ can be changed by varying the elasticity coefficient k. The deviation from the equilibrium state on the n-th step of computation is $y^n = Y_C^n - Y_C^0$, where Y_C^0 is the coordinate of the airfoil center at the initial time and Y_C^n is its coordinate at the n-th step of computation. Numerical integration of the motion equation (4) was performed using the explicit 2nd order Runge–Kutta method

$$y^* = y^n + v_y^n \frac{\Delta t}{2}, \quad v_y^* = v_y^n + \frac{F_y - b v_y^n - k y^n}{m} \frac{\Delta t}{2}.$$
$$y^{n+1} = y^n + v_y^* \Delta t, \quad v_y^{n+1} = v_y^n + \frac{F_y - b v_y^* - k y^*}{m} \Delta t.$$

Here, v_y is the airfoil vertical velocity and "$*$" denotes values at the half of the time step.

3 Numerical Methods

Here, we briefly describe the numerical methods implemented and used for numerical simulations of the previously mentioned FSI problems. For more details regarding the described methods, please see [4, 5, 9–11, 16, 22–24, 26, 30].

3.1 OpenFOAM®: A Fluid–Structure Interaction Analysis Using the Finite Volume Method

A flow simulation with moving mesh is performed using the `pimpleDyMFoam` application implemented in `OpenFOAM®` [26]. The application allows for simulations of laminar and turbulent incompressible flows with prescribed boundary motion. In order to solve FSI problems, a special `function object` was implemented. It provides a weak coupling strategy between the fluid and the structure (Fig. 2), which includes the following steps:

1. Calculation of hydrodynamic forces exerted on the airfoil from the fluid.
2. Numerical integration of the motion equation of the structure under hydrodynamic forces.
3. Application of the airfoil camber line motion law to the boundary of the fluid domain.

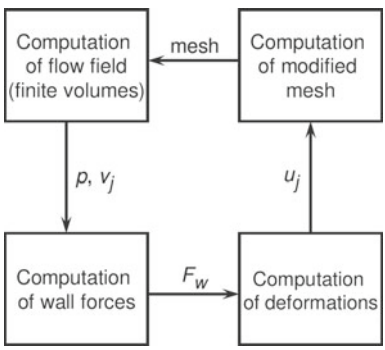


Fig. 2 Scheme of momentum exchange during the fluid–structure interaction process

The current implementation [15] of this `function object` allows for the simulation of airfoil motion with one degree of freedom (oscillations across the stream).

3.2 Kratos: Particle Finite Element Method with Fixed Mesh

This method belongs to the hybrid Lagrangian–Eulerian methods, and its recent modification PFEM2 allows for the use of a fixed mesh and large time steps. The convection of the fluid is taken into account via the motion of the Lagrangian particles, which are viewed as material points of the flow that can freely move in the flow region through the cells of the fixed mesh. In practice, this means that instead of solving the Navier–Stokes equation (2), we solve the equation with the material derivative

$$\frac{dV}{dt} = -\nabla p + \frac{1}{\mathrm{Re}}\Delta V.$$

Using this method, it is possible to simulate multidisciplinary and multiphase problems, in particular, flows with a free surface. In this case, particles can even separate from the main flow domain, representing, for instance, water drops.

When the convective substep is finished, the particle data are projected to the background fixed mesh, and the standard Finite Element-based approach is used to take into account the other terms of the Navier–Stokes equation (2). In order to satisfy the continuity equation, fractional step solvers [10] or monolithic solvers [1] can be used.

3.3 Vortex Element Method

Navier–Stokes equations (2) can be written down in Helmholtz form with respect to the vorticity field $\Omega(r,\ t) = \nabla \times V(r,\ t)$

$$\frac{\partial \Omega}{\partial t} + \nabla \times (\Omega \times U) = 0. \tag{5}$$

Here, $U(r, t) = V(r, t) + W(r, t)$, $W(r, t)$ is the so-called "diffusive velocity" [7], which is proportional to the viscosity coefficient

$$W(r, t) = \nu \frac{(\nabla \times \Omega) \times \Omega}{|\Omega|^2}. \tag{6}$$

If vorticity distribution is known, the flow velocity can be reconstructed by using the Biot–Savart law

$$V(r) = V_\infty + \frac{1}{2\pi}\int_S \frac{\Omega(\xi,t)\times(r-\xi)}{|r-\xi|^2}dS. \tag{7}$$

In order to compute the pressure distribution and hydrodynamic forces exerted on the airfoil, the analog of Bernoulli and Cauchy–Lagrange integrals is used [6].

Equation (5) describes vorticity transport in the flow with velocity U. "New" vorticity is generated only as a vortex sheet on the surface line of the airfoil, and its intensity $\gamma(\xi)$ can be found from the boundary condition on the airfoil's surface.

The vortex element method is a meshless particle-type method, so the vorticity field in the flow is discretized into separate vortex elements. Each vortex element is described by its position r_i and circulation Γ_i, $i = 1, \ldots, N$, where N is the number of vortex elements in the flow. So, the discretized Biot–Savart law has the following form:

$$V(r) = V_\infty + \sum_{i=1}^{N}\frac{\Gamma_i}{2\pi}\frac{k\times(r-r_i)}{|r-r_i|^2} + \oint_K \frac{k\times(r-\xi)}{2\pi|r-\xi|^2}\gamma(\xi)\,dl_\xi. \tag{8}$$

Here, k is the unit vector of the axis, which is orthogonal to the plane of the flow.

The movement of the vortex elements according to (5) is simulated via solution of the following ordinary differential equations system:

$$\frac{dr_i}{dt} = V(r_i) + W(r_i), \quad i = 1, \ldots, N. \tag{9}$$

The number of vortex elements in the flow N changes at every time step due to the vorticity flux from the surface line of the airfoil, which, in turn, is simulated by the vortex element generation near the airfoil. Their circulation is calculated from the vortex sheet intensity $\gamma(\xi)$ on the airfoil surface line. The circulation of all vortex elements in the flow remains constant and it can change only through a special numerical procedure of vortex wake restructuring that allows for the merging of closely spaced vortex elements and lowers their number in the flow.

Vortex sheet intensity $\gamma(\xi)$ can be found from the following boundary condition:

$$V(r) = V^{\text{ib}}(r), \quad r \in K. \tag{10}$$

This boundary condition can be reduced either to a singular integral equation of the first kind (in "classical" numerical schemes of a vortex method, see [5, 7, 21]) or to a Fredholm-type integral equation of the second kind with bounded (for smooth airfoils) kernel [12]

$$\oint_K \frac{[k\times(r-\xi)]\cdot\tau(r)}{2\pi|r-\xi|^2}\gamma(\xi)dl_\xi - \frac{\gamma(r)}{2} = -\tau(r)\cdot\left(V_\infty - V^{\text{ib}}(r) + \sum_{i=1}^{N}\frac{\Gamma_i}{2\pi}\frac{k\times(r-r_i)}{|r-r_i|^2}\right). \tag{11}$$

The solution to Eq. (11) is nonunique, so we need an additional condition for total vortex sheet circulation

$$\oint_K \gamma(r) dl_r = G. \tag{12}$$

There are high-accuracy numerical schemes developed for the numerical solution of Eqs. (11–12), which allow for the reduction of these equations to a linear algebraic equations system with well-conditioned matrix [17, 18]. By using these schemes, accuracy increases significantly [22]: in some cases, the error becomes one or even two orders of magnitude smaller in comparison with the classical schemes. As a result, this makes it possible to simulate FSI problems with high resolution.

It should be noted that the computational cost of simulating fixed and movable rigid airfoils when using the vortex element method remains nearly the same [19], so they are suitable for coupled FSI problems. However, vortex element movement simulation is an "N-body"-type problem [8], so special acceleration algorithms should be implemented. The well-known Barnes–Hut fast algorithm analog [8] can be very effective, especially when using an accurate analytical estimate of its computational cost [20], which allows for the choice of optimal parameters. Parallel computation algorithms are also used in order to reduce the computational time [19, 20].

3.4 LS-STAG Method

The LS-STAG immersed boundary method is a Eulerian method based on the finite volume approach. The fixed Cartesian mesh with cells $\Omega_{i,j} = (x_{i-1}, x_i) \times (y_{j-1}, y_j)$ and faces $\Gamma_{i,j}$ is introduced in the rectangular computational domain. Pressure $p_{i,j}$ and normal stresses are computed at centers $x^c_{i,j} = (x^c_i, y^c_j)$ of these cells. Unknown components $u_{i,j}$ and $v_{i,j}$ of velocity vector v are computed at the face center of the fluid mesh cell. These points are the centers of finite volumes of the staggered x-mesh and y-mesh: $\Omega^u_{i,j} = (x^c_i, x^c_{i+1}) \times (y_{j-1}, y_j)$ and $\Omega^v_{i,j} = (x_{i-1}, x_i) \times (y^c_j, y^c_{j+1})$, with faces $\Gamma^u_{i,j}$ and $\Gamma^v_{i,j}$, respectively. Shear stresses are computed at the corners of the base mesh (Fig. 3).

The fourth xy-mesh with cell centers in the corners of the base mesh is introduced for the simulation of turbulent flows. It is possible to simulate high Reynolds turbulent flows by using RANS, LES and DES approaches with Spalart–Allmaras, $k-\varepsilon$, $k-\omega$, and $k-\omega$ SST models [29].

The level-set function $\varphi = \varphi(x, y)$ [27] is used for capturing the airfoil immersed boundary Γ^{ib} [4]. The cells that the immersed boundary intersects are the so-called "cut-cells". These cells contain both the solid and liquid parts. The boundary Γ^{ib} is represented by a line segment on the cut-cell $\Omega_{i,j}$. In 2D cases, the cut-cells can be classified into trapezoidal, triangular and pentagonal cells. Examples of each type of cut-cell are presented in Fig. 4. The hydrodynamic force exerted on the airfoil can be computed by integrating the pressure distribution and shear stresses along the camber line of the airfoil.

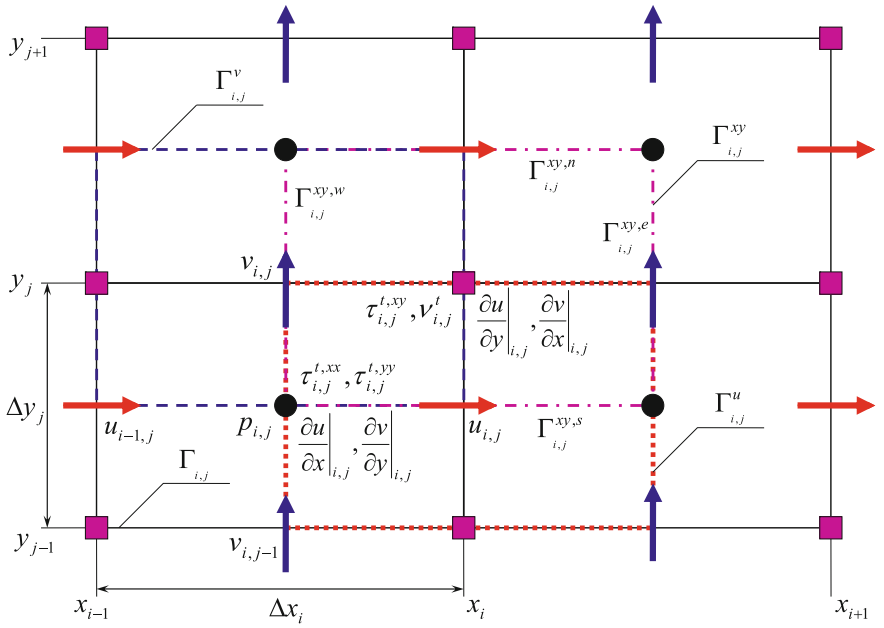


Fig. 3 Staggered arrangement of the variables on the LS-STAG mesh

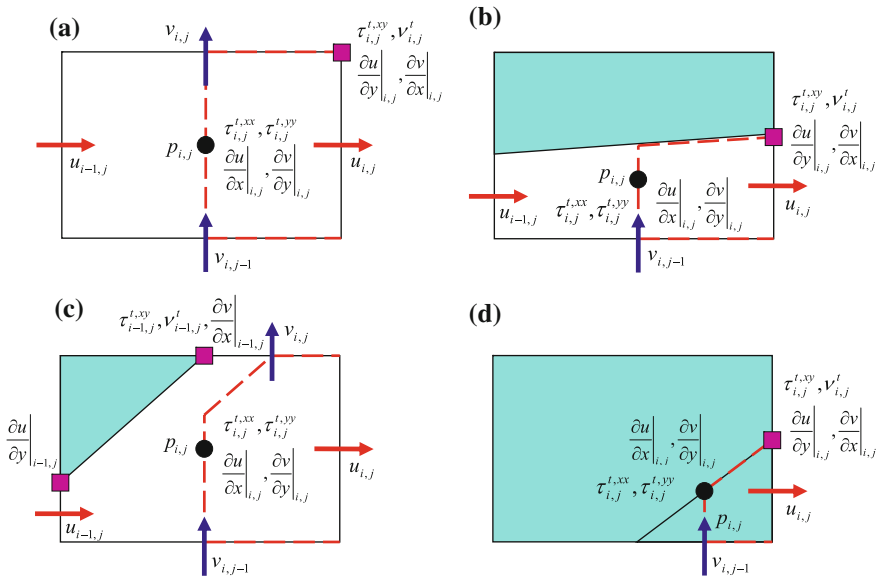


Fig. 4 Location of the variables' discretization points on the LS-STAG mesh: **a**—Cartesian Fluid Cell; **b**—North Trapezoidal Cell; **c**—Northwest Pentagonal Cell; **d**—Northwest Triangle Cell

According to the concept of the LS-STAG method [4], the governing equations (1–2) should be written in integral form for cells of base mesh, cells of x-mesh, and cells of y-mesh, respectively.

$$\begin{aligned}
&\int_{\Gamma_{i,j}} v \cdot n \, dS = 0, \\
&\frac{d}{dt} \int_{\Omega^u_{i,j}} u \, dV + \int_{\Gamma^u_{i,j}} (v \cdot n) u \, dS + \int_{\Gamma^u_{i,j}} p e_x \cdot n \, dS - \int_{\Gamma^u_{i,j}} \nu \nabla u \cdot n \, dS = 0, \\
&\frac{d}{dt} \int_{\Omega^v_{i,j}} v \, dV + \int_{\Gamma^v_{i,j}} (v \cdot n) v \, dS + \int_{\Gamma^v_{i,j}} p e_y \cdot n \, dS - \int_{\Gamma^v_{i,j}} \nu \nabla v \cdot n \, dS = 0. \quad (13)
\end{aligned}$$

Transport equations that correspond to turbulence model equations are being integrated over cells of the xy-mesh.

The time integration of the differential algebraic system that corresponds to a semi-discrete analog of the governing equations (1–2) is performed with a semi-implicit Euler scheme. The predictor step leads to discrete analogs of the Helmholtz equation for velocity prediction, while the corrector step leads to a discrete analog of the Poisson equation for pressure correction. The resulting linear systems are solved by using the FGMRES method with the ILU- and multigrid [31] preconditioning. The optimal parameters of the multigrid preconditioner were chosen by using the original algorithm for estimation of the solver cost-coefficient [25].

4 Numerical Simulation

We consider a viscous incompressible flow with low Reynolds number $\mathrm{Re} = 150$, i.e., the flow is laminar and a turbulence model is not needed. In dimensionless variables, the cylinder has unit diameter, the media has unit density, and the velocity of the incident flow is $V_\infty = 3$. The mass of the cylinder is $m = 39.15$. The fixed cylinder generates periodical vortex shedding in a quasi-steady regime with frequency f, which corresponds to the Strouhal number $\mathrm{Sh} = f d / V_\infty \approx 0.185$. Therefore, we choose the rigidity of the elastic constraint, so that the dimensionless construction frequency

$$\mathrm{Sh}_\omega = \frac{1}{2\pi} \sqrt{\frac{k}{m}} \frac{d}{V_\infty}$$

varies within the range $0.160 \ldots 0.220$. The viscosity of the constraint has a small value, $b = 0.731$, and it practically does not influence the eigenvalue of the system at all. The rest of the system parameters were chosen in the same way as in [13].

Table 1 Results of flow simulation around fixed cylinder

	C_{xa}	Sh	C_{ya}^{ampl}
Experiment	1.15 . . . 1.45	0.175 . . . 0.195	0.50 . . . 0.65
`OpenFOAM®`	1.44	0.190	0.60
`Kratos`	1.20	0.190	0.53
Vortex method	1.31	0.177	0.51
LS-STAG	1.32	0.191	0.63

4.1 Flow Simulation Around the Fixed Airfoil

The described numerical methods were first validated by using a classical test problem of flow around an immovable airfoil. At the beginning of the numerical simulation, the incident flow velocity was set to zero, and the flow was gradually accelerated, until its value reached $V_\infty = 3$. During the simulation, the dimensionless drag coefficient, the amplitude of the lift coefficient and the vortex shedding frequency was measured. The results are shown in Table 1.

Calculations show that all the methods give reasonable results compared to the experimental data. At the same time, it is necessary to notice that the Particle Finite Element Method (PFEM2) implemented in the `Kratos` software overpredicts the pressure values near the critical point compared to the experiment.

Mesh motion in `OpenFOAM®` is calculated by solving the point displacement Laplacian equation. This type of numerical implementation is efficient only for simple airfoil motion trajectory—displacements with a small rotation angle.

The LS-STAG method has potential in flow simulations around arbitrary moving airfoils with complicated shapes. However, its computational complexity is rather high. Unfortunately, its parallel implementation is not available at the moment (due to principal problems with matrix decompositions).

4.2 Wind Resonance Simulation

A number of numerical simulations have been carried out for different values of the dimensionless frequency in which the unsteady process has been simulated from $t = 0$ to $t = 200$. The obtained dependencies of the dimensionless oscillation amplitude A/d (in quasi-steady mode) of the airfoil on the dimensionless natural frequency of the system Sh_ω are shown in Fig. 5.

The obtained dependencies are very close, while they differ only in shift of frequency, which corresponds to the results of the Strouhal number (dimensionless vortex shedding frequency) computation for the fixed cylinder (see Table 1). The maximum value of the amplitude is close to the experimental value [2, 13].

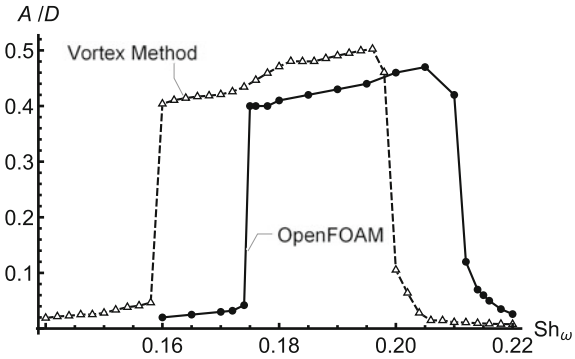


Fig. 5 Circular airfoil oscillations' amplitude dependence on the natural frequency of the system, obtained with the finite volume method (OpenFOAM®) and the vortex method

Table 2 Computational time (in hours) for OpenFOAM® and for the vortex method

	1 CPU	2 CPU	4 CPU	8 CPU	16 CPU
OpenFOAM®					
Far from resonance	58.1	36.0	24.3	15.5	9.8
Close to resonance	74.4	45.8	30.1	19.6	13.7
Vortex method					
Far from resonance	41.3	22.6	12.0	7.1	4.7
Close to resonance	63.4	34.7	17.9	10.1	6.6

The computational times spent for numerical simulations of the flow around freely oscillating airfoils by using OpenFOAM® and the vortex method is given in Table 2. Parallel computational technologies (MPI) reduce the required computational time significantly. Measured computational time is given for two regimes: close to wind resonance and far from it.

When solving the same problem by using the LS-STAG method in sequential mode, the time of computation was approximately 110 and 150 h for non-resonance and resonance cases, respectively; the time for non-resonance simulation in Kratos (PFEM2 method) was approximately 55 h in sequential mode and 16 h in parallel mode with OpenMP technology using 4 CPU cores. However, when simulating airfoil oscillations with rather high amplitude, the amplitude of the lift force coefficient in Kratos is much higher than in the experiment. Therefore, this method should be used only for simulations in which the amplitude of airfoil oscillations is not larger than several percent of the diameter size. The given results for all numerical methods were obtained on the same computational cluster consisting of 8 personal computers with Intel Core i7 3.2 GHz processors connected by Cisco Gigabit Ethernet.

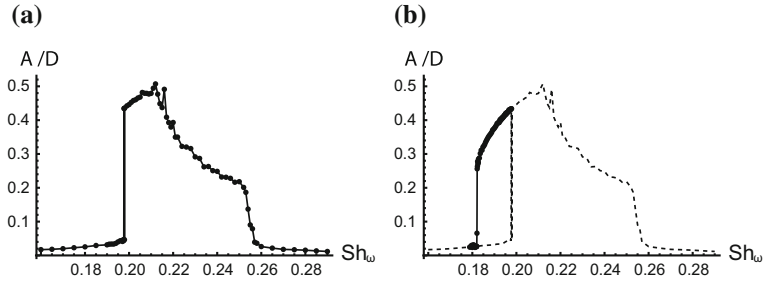


Fig. 6 Maximum amplitude of the circular airfoil oscillations simulated using the vortex element method: **a** airfoil's initial state in the equilibrium position, **b** airfoil's initial state is close to resonance oscillations

4.3 Hysteresis Simulation

In order to capture a well-known hysteresis phenomenon [14] during the flow around an airfoil with elastic constraints, 80 simulations with different dimensionless frequencies were performed. The Reynolds number was set to 1000, while the other parameters remained the same as in the previously stated resonance simulations. The dependency of the oscillations' amplitude on the natural frequency Sh_ω is shown in Fig. 6a, where a sharp amplitude increase is observed at $\mathrm{Sh}_\omega \approx 0.198$.

In the second series of numerical simulations, Sh_ω was equal to 0.21 from $t = 0$ to $t = 100$. During this time interval, the oscillations reached steady state with amplitude $A/D \approx 0.47$. At simulation time $t = 100$, the elasticity coefficient of the constraint was changed abruptly to the values that correspond to oscillation frequencies Sh_ω from 0.178 to 0.198. In each case, after the transitional period, new steady oscillations were reached. Their amplitudes are shown in Fig. 6b (dots connected by a solid line).

The obtained results for maximum oscillation amplitude, the resonance frequency and the hysteresis properties are in good agreement with the results given in [13, 14].

5 Comparison of the Considered Numerical Methods

A comparison of the properties between the considered numerical methods is presented in Table 3. The sign "+" means that this property is inherent in the method and is implemented efficiently; "±" means that the method yields to others in this feature; "−" means the absence of the property or its inefficient implementation; the asterisk "∗" means that this property is not fully implemented or it is in a development stage.

Unfortunately, it is impossible to use vortex methods and the particle finite element method when simulations of flows with high Reynolds numbers are required. In vortex methods in a purely Lagrangian framework, attempts to implement turbulence

Table 3 Comparison of the numerical methods

	OpenFOAM®	Vortex method	Kratos	LS-STAG
Computational cost	±	+	+	−
Accuracy	+	±	±	+
Airfoil motion	±	+	−	+
Parallel implementation	+	+	*	*
Automatic time step choice	+	*	−	±
Turbulent flows	+	−	−	+
3D flows	+	*	+	*

models are not very efficient [28]. The current implementation of the FEM particle model inside the Kratos software package does not have the required functionality for implementing turbulence models.

6 Conclusion

When simulating flows with relatively small Reynolds numbers (from 100 to 10,000), all the abovementioned numerical models give results that are in good agreement with the experimental data. The overall efficiencies of all considered methods and codes are comparable. However, simulating flows in high Reynolds regimes, in which turbulent effects must be taken into account while preserving acceptable computational costs, can be done only with OpenFOAM®. It should be noted, however, that the choice of the turbulence model, as well as that of the RANS/LES approach, is a nontrivial problem. Moreover, OpenFOAM® is well parallelized, thus computational time can be reduced significantly. At the same time, parallel algorithms in vortex methods are efficient only for a small number of computational cores. Current PFEM implementation in the Kratos software works in parallel mode only when using OpenMP technology (on shared-memory systems).

As a result, the following conclusion can be made: OpenFOAM® is a good tool for fast estimation of dynamic processes in fluid–structure interaction problems. It enables analysis of FSI system behavior within a wide range of parameters.

References

1. Becker, P., Idelsohn, S.R., Onate, E.: A unified monolithic approach for multi-fluid flows and fluid-structure interaction using the Particle Finite Element Method with fixed mesh. Computat.

Mech. (2015) https://doi.org/10.1007/s00466-014-1107-0
2. Blevins, R.D., Coughran, C.S.: Experimental investigation of vortex-induced vibration in one and two dimensions with variable mass, damping, and reynolds number. J. of Fluids Eng. (2009) https://doi.org/10.1115/1.3222904
3. Chen, S.-S., Yen, R.-H., Wang, A.-B.: Investigation of the resonant phenomenon of flow around a vibrating cylinder in a subcritical regime. Physics of Fluids (2011) https://doi.org/10.1063/1.3540673
4. Cheny, Y., Botella, O.: The LS-STAG method: A new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties. J. of Comput. Phys. (2010) https://doi.org/10.1016/j.jcp.2009.10.007
5. Cottet, G.-H., Koumoutsakos, P.D.: Vortex methods. Theory and practice. Cambridge University Press (2008)
6. Dynnikova, G.Ya.: Vortex motion in two-dimensional viscous fluid flows. Fluid Dynamics (2003) https://doi.org/10.1023/B:FLUI.0000007829.78673.01
7. Dynnikova, G.Ya.: The Lagrangian approach to solving the time-dependent Navier-Stokes equations. Dokl. Phys. (2004) https://doi.org/10.1134/1.1831530
8. Dynnikova, G.Ya.: Fast technique for solving the N-body problem in flow simulation by vortex methods. Comput. Math. and Math. Phys. (2009) https://doi.org/10.1134/S0965542509080090
9. Ferziger, J., Peric, M.: Computational methods for fluid dynamics. Springer (2002)
10. Idelsohn, S., Nigro, N., Gimenez, J., Rossi, R., Marti, J.: A fast and accurate method to solve the incompressible Navier-Stokes equations. Engineering Computations. (2013) https://doi.org/10.1108/02644401311304854
11. Idelsohn, S., Nigro, N., Limache, A., Onate, E.: Large time-step explicit integration method for solving problems with dominant convection. Comput. Methods. Appl. Mech. Eng. (2012) https://doi.org/10.1016/j.cma.2011.12.008
12. Kempka, S.N., Glass, M.W., Peery, J.S., Strickland, J.H.: Accuracy considerations for implementing velocity boundary conditions in vorticity formulations. SANDIA Report SAND96-0583 (1996)
13. Klamo, J.T., Leonard, A., Roshko, A.: On the maximum amplitude for a freely vibrating cylinder in cross flow. J. Fluids. Struct. (2005) https://doi.org/10.1016/j.jfluidstructs.2005.07.010
14. Klamo, J.T., Leonard, A., Roshko, A.: The effects of damping on the amplitude and frequency response of a freely vibrating cylinder in cross-flow. J. Fluids. Struct. (2006) https://doi.org/10.1016/j.jfluidstructs.2006.04.009
15. Kraposhin, M.V., Marchevsky, I.K.: Implementation of simple FSI model with functionObject (2016) https://doi.org/10.13140/RG.2.1.1526.6809/1 https://github.com/unicfdlab/TrainingTracks/tree/master/OpenFOAM/simpleFsi-OF3.0.0. Cited 1 Oct 2016
16. Kratos multi-physics. http://www.cimne.com/kratos. Cited 1 Oct 2016
17. Kuzmina, K.S., Marchevskii, I.K., Moreva, V.S.: Vortex Sheet Intensity Computation in Incompressible Flow Simulation Around an Airfoil by Using Vortex Methods. Math. Model. and Comput. Simul. (2018) https://doi.org/10.1134/S2070048218030092.
18. Kuz'mina, K.S., Marchevskii, I.K., Moreva, V.S., Ryatina, E.P.: Numerical scheme of the second order of accuracy for vortex methods for incompressible flow simulation around airfoils. Russian Aeronautics (2017) https://doi.org/10.3103/S1068799816030114
19. Kuzmina, K.S., Marchevsky, I.K., Moreva, V.S.: Parallel Implementation of Vortex Element Method on CPUs and GPUs. Procedia Computer Science (2015) https://doi.org/10.1016/j.procs.2015.11.010
20. Kuzmina, K., Marchevsky, I., Ryatina, E.: VM2D: Open source code for 2D incompressible flow simulation by using vortex methods. Commun. Comput. and Inf. Sci. (2018) https://doi.org/10.1007/978-3-319-99673-8_18
21. Lifanov, I.K., Belotserkovskii, S.M.: Methods of Discrete Vortices. CRC Press (1993)
22. Marchevsky, I.K., Moreva, V.S., Puzikova V.V.: The efficiency comparison of the vortex element method and the immersed boundary method for numerical simulation of airfoils hydroelastic oscillations COUPLED PROBLEMS 2015: Proc. of VI Int. Conf. on Computational Methods for Coupled Problems in Sci. and Eng., 800-811 (2015). http://congress.cimne.com/coupled2015/frontal/doc/EbookCOUPLED15.pdf. Cited 1 Oct 2016

23. Marchevskii, I.K., Puzikova V.V.: Numerical simulation of the flow around two fixed circular airfoils positioned in tandem using the LS-STAG method. J. Mach. Manuf. Reliab. (2016) https://doi.org/10.3103/S1052618816020084
24. Marchevskii, I.K., Puzikova V.V.: Numerical simulation of the flow around two circular airfoils positioned across the stream using the LS-STAG method. J. Mach. Manuf. Reliab. (2017) https://doi.org/10.3103/S105261881702011X
25. Marchevsky, I.K., Puzikova, V.V.: OpenFOAM® iterative methods efficiency analysis for linear systems solving. Proceedings of the Institute for System Programming of RAS (2013) https://doi.org/10.15514/ISPRAS-2013-24-4 [in Russian]
26. OpenFOAM® —the open source computational fluid dynamics (cfd) toolbox. http://www.openfoam.com. Cited 1 Oct 2016
27. Osher, S., Fedkiw, R.P.: Level set methods and dynamic implicit surfaces. Springer (2003)
28. Pereira, L.A.A., Hirata, M.H., Silveira Neto A.: Vortex method with turbulence sub-grid scale modelling. J. Braz. Soc. Mech. Sci. & Eng. (2003) https://doi.org/10.1590/S1678-58782003000200005
29. Puzikova, V.V.: Development of the LS-STAG immersed boundary method modification or mathematical modelling in coupled hydroelastic problems. Ph.D. dissertation, Bauman Moscow State Technical University, Russia (2016) [in Russian] http://mechmath.ipmnet.ru/lib/?s=thesis_mech. Cited 1 Oct 2016
30. Weller, H.G., Tabor, G., Jasak, H., Fureby, C.: A tensorial approach to computational continuum mechanics using object-oriented techniques. Computers in Physics (1998) https://doi.org/10.1063/1.168744
31. Wesseling, P.: An introduction to multigrid methods. John Willey & Sons Ltd. (1991)

The Harmonic Balance Method for Temporally Periodic Free Surface Flows

Inno Gatin, Gregor Cvijetić, Vuko Vukčević and Hrvoje Jasak

Abstract The Harmonic Balance Method for temporally periodic, non-linear, turbulent, free surface flows is presented in this work. The method transforms a periodic transient problem into a set of coupled steady-state problems, increasing the efficiency of calculation. The methodology is primarily targeted to efficient simulations related to wave–structure interaction in naval and offshore hydrodynamics. The method is validated on a 2D periodic free surface flow over a ramp test case and a 3D ship wave diffraction test case.

1 Introduction

Transient flows in marine hydrodynamics are often periodic, e.g. due to ocean waves (wave propagation and diffraction, seakeeping of a ship) and rotating propellers. Such flows often have a well-defined base frequency: the wave frequency or rotational frequency of the propeller. In fully non-linear, two-phase state-of-the-art CFD algorithms, such flows are almost exclusively resolved in the time domain [7, 8]. Transient simulations usually require a large number of periods in order to achieve a harmonically steady (purely oscillatory) solution. Due to its spectral decomposition, the Harmonic Balance Method (HBM) allows us to efficiently model flow effects up to a specified order, without performing a fully transient simulation. Hence, a substantial performance improvement is expected, with an almost negligible decrease in accuracy for flows with a well-defined base frequency. Due to the steady-state mathematical formulation of the HBM, the authors believe that the method is highly

I. Gatin (✉) · G. Cvijetić · V. Vukčević · H. Jasak
University of Zagreb, Zagreb, Croatia
e-mail: innogatin@gmail.com

G. Cvijetić
e-mail: gregor.cvijetic@gmail.com

V. Vukčević
e-mail: vuko.vukcevic@fsb.hr

H. Jasak
e-mail: hrvoje.jasak@fsb.hr

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_34

suitable for adjoint optimisation regarding seakeeping of ships in the ship-building industry. This suitability has been recently confirmed by Huang and Ekici [6], who developed an adjoint shape optimisation tool based on the HBM for turbomachinery applications.

The HBM [5] was originally developed to tackle periodic single-phase turbomachinery flows in an efficient way. This paper presents an extension of the single-phase HBM [2, 4] to two-phase free surface flows, comparing the results and computational efficiency with a transient simulation. The implementation is carried out in a second-order accurate, polyhedral Finite Volume framework developed within foam-extend, a community-driven fork of the OpenFOAM® software.

2 Harmonic Balance Method

In the HBM, a transient governing equation set is replaced with a specified number of coupled steady-state problems, each represented by an equation for a unique time instant. The method simulates a periodic flow by evaluating the temporal derivative via spectral decomposition, yielding a flow solution at discrete instants in time simultaneously. Multi-mode transformation from a transient to a set of coupled steady-state problems is achieved through a Fourier transform, assuming temporally periodic flow. The accuracy of the model is controlled by a specified number of harmonics to allow for the efficient capturing of higher order flow effects. Generally, the specified number of harmonics n, yields solutions at $2n + 1$ discrete time instants.

2.1 Mathematical Model

A general field variable ϕ is expanded in a truncated Fourier series with a known base frequency ω:

$$\phi(t) = \Phi_0 + \sum_{i=1}^{N} \left(\Phi_{S_i} \sin(i\omega t) + \Phi_{C_i} \cos(i\omega t) \right), \tag{1}$$

where Φ is the general field variable in the frequency domain, while indices S_i and C_i indicate the sine and cosine coefficients, respectively. Equation 1 can be presented using a Fourier transformation matrix $\underline{\underline{E}}$: $\underline{\phi} = \underline{\underline{E}}\,\underline{\Phi}$, where $\underline{\phi}$ denotes the vector of ϕ at discrete time instants. The general transport equation reads as

$$\frac{\partial \phi}{\partial t} + \mathcal{R} = 0, \tag{2}$$

where $\mathcal{R}$ stands for convection, diffusion and sink/source terms. The field transformed using Eq. 1 is then inserted into Eq. 2, and corresponding terms are equated, yielding $2n + 1$ equations:

$$\omega \underline{\underline{A}}\, \underline{\Phi} + \underline{R} = \underline{0}, \tag{3}$$

where $\underline{\underline{A}}$ represents the $2n + 1$ by $2n + 1$ coupling matrix stemming from the Fourier transformation, while $\underline{\Phi}$ and $\underline{R}$ are the vectors of field variables and convection, diffusion and sink/source terms in the frequency domain, respectively. The time-spectral approach is based on the equations obtained by transforming Eq. 3 into the time domain using Discrete Fourier Transform (DFT) matrix $\underline{\underline{E}}$:

$$\omega \underline{\underline{E}}^{-1}\, \underline{\underline{A}}\, \underline{\underline{E}}\, \underline{\phi} + \underline{\mathcal{R}} = \underline{0}. \tag{4}$$

Equation 4 presents a set of $2n + 1$ equations coupled with the analytically transformed temporal term via spectral decomposition. Each equation represents one discrete time instant in one period of oscillation corresponding to ω.

In the present study, HBM is applied to Navier–Stokes equations and the Level Set interface capturing Eq. [9], yielding a coupled set of two-phase flow equations for discrete instants of time within one period. In addition, SWENSE decomposition [9] is used to facilitate incident wave propagation. The reader is directed to [3, 4] for further details.

2.2 Coupling of Steady-State Equations

The term $\omega \underline{\underline{E}}^{-1}\, \underline{\underline{A}}\, \underline{\underline{E}}\, \underline{\phi}$ presents a source term that couples the steady-state equations. The coupling can either be resolved in an explicit or implicit manner. In this work, the coupling is resolved implicitly by solving the equations simultaneously in one block system. The block system contains the block matrix and block vectors, where each entry presents a vector of size $2n + 1$. The implicit approach enhances the stability of the calculation and enables low mean velocities with respect to the magnitude of oscillation. This is important for the naval hydrodynamic application, since wave-related flows often have low or zero mean velocities. The explicit approach is also used for purposes of comparison.

3 Test Cases

In this section, two test cases are shown: a 2D simulation of a periodic flow over a ramp and a 3D wave diffraction of a DTMB ship simulation. The results are compared with transient simulations to validate the method.

3.1 2D Ramp Test Case

A simple 2D test case is devised to validate the method for periodic free surface flows. A periodically changing inlet velocity is prescribed that enforces a periodic variation of the free surface throughout the domain. The simulation geometry can be seen in Fig. 1. The inlet velocity is determined as $U_{inlet} = [6, 0, 0] + [1, 0, 0] \sin(2\pi t/T)$, where $T = 0.5$ s stands for the prescribed period of oscillation. Figure 2 shows the initial condition with the calm free surface. The free surface elevation is measured 0.5 m from the outlet boundary. 13,000 cells are used in both the transient and HBM simulations, while 200 time steps per period are used in the transient simulation. Simulations using 1–8 harmonics are performed to asses the sensitivity of the solution on spectral resolution. In this case, the oscillation of velocity is small compared to the mean velocity, hence the explicit approach for resolving the source coupling can be used. For comparison, both the explicit and implicit methodologies are used.

Figure 3 shows the dynamic pressure and velocity field in the discrete time instants for the simulation with 2 harmonics. The comparison of the free surface elevation from the HBM simulations using different numbers of harmonics with the transient simulation is shown in Table 1, where η_a stands for the free surface elevation amplitudes, with indices 0 and 1 indicating zero- and first-order harmonic amplitudes, respectively. ε is the relative difference of the transient result and the HBM method, $\varepsilon = \left(\eta_{a,t} - \eta_{a,hb}\right)/\eta_{a,t}$; here indices t and hb present the transient and HBM results, respectively. The difference decreases with the increase in the number of harmonics, reducing to -0.2 % for the mean and -2.1 % for the first order.

Table 2 shows the comparison of required computational time for the explicit and implicit HBM simulation and the transient simulation. The explicit HBM simulation is more than ten times faster than the transient simulation. The implicit simulation is more than three times slower than the explicit HBM simulation, however, it is still 2.7 times faster than the transient simulation. The decrease in performance between the explicit and implicit approaches is caused by the high cost of solving the block system of equations. It should be noted that two harmonics were resolved in these simulations, and that using more than two harmonics would deteriorate the increase in speed. However, the motivation behind applying HBM to the field of naval hydrodynamics is to provide a trade-off between accuracy and performance by choosing the number of harmonics accordingly, rather than increasing the accuracy of the existing methods.

3.2 DTMB Wave Diffraction Test Case

Wave diffraction against a static DTMB ship model [1] is simulated using the implicit HBM and transient approaches. Regular waves are imposed, while induced longitudinal and vertical forces acting on the hull are measured and compared. The model is $L = 3.05$ m long with a velocity of $U = 1.52$ m/s corresponding to the Froude

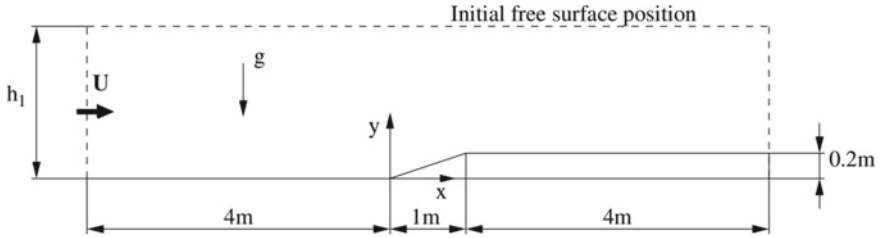


Fig. 1 2D ramp test case geometry

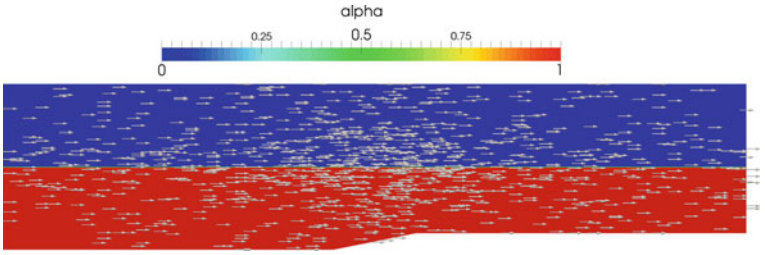
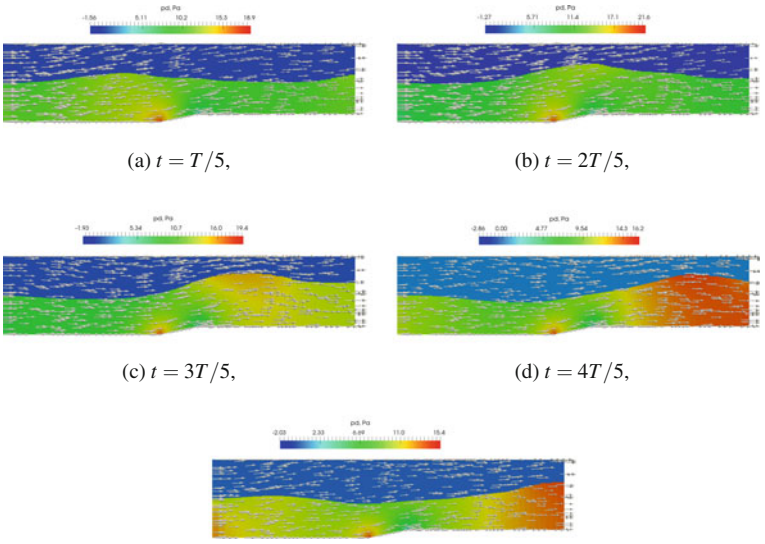


Fig. 2 Initial free surface and velocity in the ramp test case



(a) $t = T/5$, (b) $t = 2T/5$,

(c) $t = 3T/5$, (d) $t = 4T/5$,

(e) $t = T$.

Fig. 3 Dynamic pressure and velocity distribution in the HBM simulation in discrete time instants

Table 1 Comparison of HBM and transient simulation results for the ramp test case

No. Harmonics	$\eta_{a,0}$ (m)	$\eta_{a,1}$ (m)	ε_0 (%)	ε_1 (%)
1	1.29035	0.179472	−1.6	−15.0
2	1.28485	0.186762	−1.2	−19.6
3	1.26487	0.175538	0.4	−12.5
4	1.27065	0.163556	−0.1	−4.8
5	1.27084	0.164377	−0.1	−5.3
6	1.27240	0.161346	−0.2	−3.4
7	1.27210	0.159447	−0.2	−2.1
8	1.27218	0.159308	−0.2	−2.1

Table 2 Comparison of computational time between the HBM and transient simulation for the ramp test case

Simulation type	CPU time (s)	Acceleration
Transient (10 periods)	5067	1
Explicit HB (2 harmonics)	488	10.4
Implicit HB (2 harmonics)	1851	2.7

number $F_r = 0.28$. The waves are $H = 0.036$ m high with a period of $T = 1.09$ s and wavelength $\lambda = 4.57$ m. Two harmonics are used for the HBM simulations, while 200 time steps per period are used in the transient simulation. 521,000 cells mesh is used in both the simulations. The wave-induced velocity is significant with respect to the ship model's velocity, therefore, implicit treatment of the HBM source terms must be used [4] in order to ensure numerical stability.

Figure 4 shows the convergence of the mean (zero) and first-order longitudinal forces and the first order of the vertical force, where N_{Iter} denotes the number of iterations. It can be seen that the forces converge smoothly. The mean of the vertical force is excluded, since it has a very large absolute value.

Figure 5 shows the comparison of the free surface elevation in the transient with the HBM simulation, where good correspondence can be observed. The colour scale represents the elevation of the free surface. The forces calculated on the hull of the model in the HBM and transient simulations are compared in Table 3, where $\varepsilon = (F_t - F_{hb}) / F_t$ is given in percentages. Indices x and z denote the axis of force direction, while 0 and 1 denote zero- and first-order harmonic amplitudes. The difference is smaller than ≈10% for all items, the smallest difference being for the mean of vertical force $F_{z,0}$ (−0.11%), and the largest for the mean of the longitudinal force $F_{x,0}$ (−10.2%).

The comparison of the required computational time in the two simulations is shown in Table 4. The increase in speed is similar to that in the ramp test case with implicit coupling.

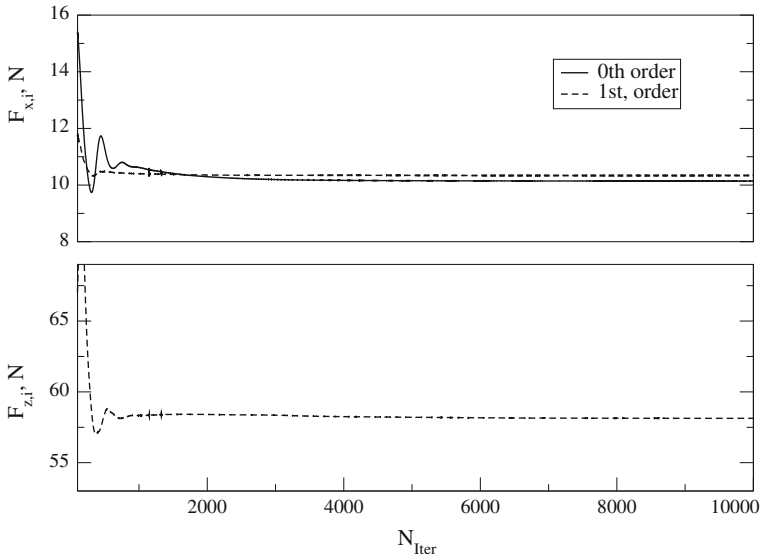
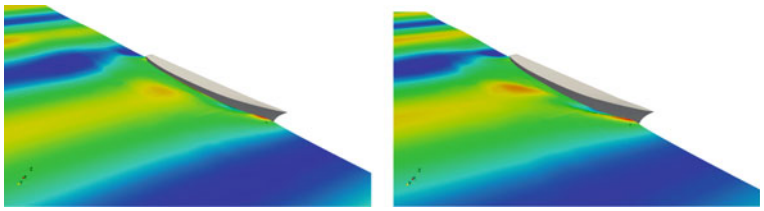


Fig. 4 Convergence of longitudinal and vertical forces acting on the DTMB hull in the HBM simulation



(a) HBM simulation, (b) Transient simulation.

Fig. 5 Free surface elevation in the HBM and transient wave diffraction simulations

Table 3 Comparison of diffraction forces in the HBM and transient simulations

Item	Transient	Harmonic balance	ε (%)
$F_{x,0}$, N	9.20	10.14	−10.2
$F_{x,1}$, N	10.70	10.34	3.36
$F_{z,0}$, N	784.88	785.72	−0.11
$F_{z,1}$, N	62.63	58.14	7.17

Table 4 Comparison of computational time between the HBM and transient simulations for the wave diffraction test case

Simulation type	CPU time (h)	Acceleration
Transient (20 periods)	18.86	1
Implicit HB (2 harmonics)	8.6	2.2

4 Conclusion

A Harmonic Balance Method applied to two-phase flows is presented in this paper with the application in the field of marine hydrodynamics. The method transforms transient periodic flows into a set of coupled steady-state problems, accelerating the calculations.

Two test cases are presented to validate the method: a 2D periodic two-phase flow over a ramp, and a ship model wave diffraction case in 3D. Both cases showed good agreement with the transient simulations with lower necessary computational time. The transient ramp test case simulation took ten times more computational time than the explicit HBM simulation, and 2.7 times more than the implicit HBM. The wave diffraction test case was simulated using an implicit HBM for reasons of numerical stability, with an acceleration by a factor of 2.2.

The implicit HBM is applicable for wave-related problems in naval hydrodynamic, however, larger computational savings were anticipated by the authors. The implicit treatment of coupling source terms exerts higher computational demands, reducing the efficiency of the method. Future efforts will be directed towards enhancing the efficiency of the implicit approach to achieve larger savings in computational resources.

Nonetheless, the method presents an attractive alternative for transient periodic flows with a free surface. Moreover, the steady-state formulation enables automatic optimisation techniques such as adjoint shape optimisation, presenting a new opportunity to optimise ships for added wave resistance in the future.

References

1. Chalmers University of Technology. Gothenburg 2010: A Workshop on CFD in Ship Hydrodynamics. http://www.insean.cnr.it/sites/default/files/gothenburg2010/index.html, 2010 [Online; accessed 20 August 2015].
2. G. Cvijetić, H. Jasak, and V. Vukčević. Finite Volume Implementation of Non-Linear Harmonic Balance Method for Periodic Flows. In *SciTech*, January 2016.
3. G. Cvijetić, H. Jasak, and V. Vukčević. Finite volume implementation of the harmonic balance method for periodic non-linear flows. In *54th AIAA Aerospace Sciences Meeting*, page 0070, 2016.
4. K. C. Hall, K. Ekici, J. P. Thomas, and E. H. Dowell. Harmonic balance methods applied to computational fluid dynamics problems. *Int. J. Comput. Fluid D.*, 27(2):52–67, 2013.
5. K. C. Hall, J. P. Thomas, and W. S. Clark. Computation of unsteady nonlinear flows in cascades using a harmonic balance technique. *AIAA Journal*, 40(5):879–886, 2002.
6. H. Huang and K. Ekici. A discrete adjoint harmonic balance method for turbomachinery shape optimization. *Aerospace Science and Technology*, 39:481–490, 2014.
7. L. Larsson, F. Stern, M. Visonneau, N. Hirata, T. Hino, and J. Kim, editors. *Tokyo 2015: A Workshop on CFD in Ship Hydrodynamics*, volume 2, Tokyo, Japan, 2015. NMRI (National Maritime Research Institute).

8. L. Larsson, F. Stern, M. Visonneau, N. Hirata, T. Hino, and J. Kim, editors. *Tokyo 2015: A Workshop on CFD in Ship Hydrodynamics*, volume 3, Tokyo, Japan, 2015. NMRI (National Maritime Research Institute).
9. V. Vukčević, H. Jasak, and S. Malenica. Decomposition model for naval hydrodynamic applications, Part I: Computational method. *Ocean Eng.*, 121:37–46, 2016.

Two-Way Coupled Eulerian–Eulerian Simulations of a Viscous Snow Phase with Turbulent Drag

Ziad Boutanios and Hrvoje Jasak

Abstract A novel two-way coupled Eulerian–Eulerian CFD formulation was developed to simulate drifting snow based on turbulent drag and a new viscous treatment of the drifting snow phase, derived from first principles. This approach allowed explicit resolution of the saltation layer without resorting to empiricism, unlike other Eulerian–Eulerian models based on mixture formulations and one-way coupling. Initial validations were carried out against detailed snow flux, airflow velocity, and turbulent kinetic energy measurements in a controlled experimental simulation of drifting snow in a wind tunnel using actual snow particles. The two-way coupled approach was found capable of simulating drifting snow fluxes in both saltation and suspension layers with reasonable accuracy. Recommendations were made to improve the accuracy of the method for air velocity and turbulent kinetic energy, and to allow simulating a drifting snow phase with a particle size distribution.

Keywords Drifting snow · Eulerian–Eulerian · Viscosity · Turbulent drag · Snow flux

1 Introduction

Drifting snow results from the aeolian motion of snow particles deposited on the ground. Such motion is possible when the drag force induced by the airflow exceeds the opposing actions of interparticle cohesive bonding, particle weight, and surface friction. This aerodynamic entrainment threshold is called the *fluid threshold*. If a large enough amount of particles is displaced by the airflow it can extract enough momentum from it that the airflow velocity is noticeably reduced; a two-way cou-

Z. Boutanios (✉)
Binkz Inc, Laval, Canada
e-mail: ziad@binkz.ca

Z. Boutanios · H. Jasak
Faculty of Mechanical Engineering and Naval Architecture,
University of Zagreb, Zagreb, Croatia
e-mail: hrvoje.jasak@fsb.hr

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_35

pling phenomenon. Particle collisions also help sustain drifting by putting snowbed particles in motion and making it easier for the slower airflow to carry them at a lower threshold shear stress referred to as the *impact threshold*. Both definitions were first coined by Bagnold [1] in his investigations of desert sand transport by the wind. Bagnold also classified the aeolian motion of particles under three modes: creeping, saltation, and suspension. These modes are shown in Fig. 1 as they pertain to drifting snow. Of particular interest to this research is the two-way coupled saltation mode. Several aeolian snow transport models are available in the literature. Most are based on Reynolds-Averaged Navier–Stokes (RANS) formulations in the Eulerian–Eulerian and Eulerian–Lagrangian frames regarding the air and snow phases, respectively. Both approaches can yield reasonable results for particle-laden flows, but the Eulerian–Eulerian approach requires a lower computational effort since a lot of particles are required for Lagrangian particle tracking to yield statistically meaningful results [2]. Presently, Eulerian–Eulerian modeling of drifting snow is based on two main approaches: the transport of snowdrift density approach and the Volume of Fluid (VOF) approach.

The snowdrift density approach solves a one-way coupled Partial Differential Equation (PDE) for the drifting snow density in the suspension layer (where the snow phase motion does not affect the airflow, a valid assumption in the suspension layer thanks to the very low snow phase concentration) in addition to the airflow continuity and momentum equations with a Prandtl mixing layer model [3]. Another variation of the transport of snow density approach uses the mixture continuity and momentum equations with a standard k–ϵ turbulence model corrected for turbulence damping by particles [4]. Note that the mixture formulation of Naaim et al. [4] is in reality two-way coupled. However, the snow phase momentum equation is not solved. Instead, the snow velocity is set equal to a terminal velocity derived from empirical and experimental considerations. This effectively fixes the two-way coupling effects and the results are indeed quite comparable to other one-way coupled approaches

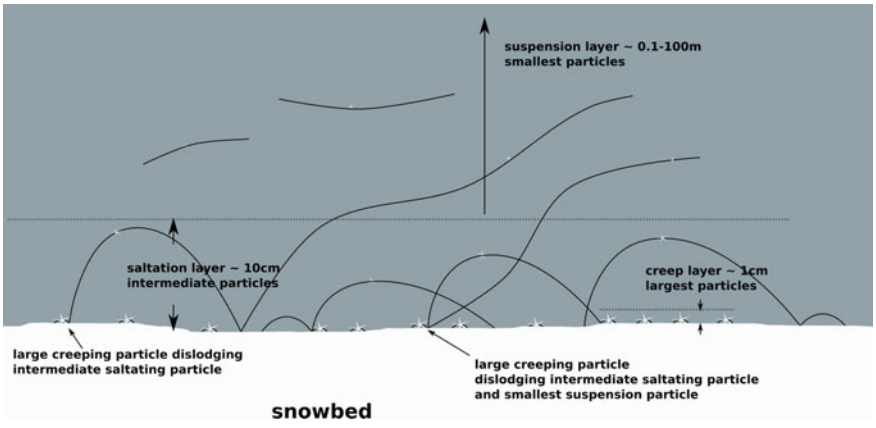


Fig. 1 The different modes of drifting snow

discussed in this section. Another application of the snowdrift density approach uses the airflow continuity and momentum equations with a modified Launder–Kato k–ϵ turbulence model [5]. The drifting snow density is transported by the airflow, the snowfall velocity is set constant, and the saltation layer, which is not resolved, is represented by a steady-state empirical formulation of the transport rate of drifting snow in the saltation layer for equilibrium conditions over natural flat terrain [6]. However, this empirical representation of the saltation layer has been experimentally shown to overestimate the transport rate of drifting snow in accelerating and decelerating flows [7]. This makes such a representation of the empirical layer at best conservative for flows around bluff bodies where important regions of accelerating and decelerating flows are present. The snowdrift density approach with the Launder–Kato turbulence model has been lately modified to account for snow particle damping of turbulence and with saltation computed with the snowdrift density one-way coupled transport equation, without the empirical saltation flux relationships [8]. These modifications resulted in some improvement, but substantial deviations remain from experimental measurements of the snow surface in the lateral vicinity of a cube structure, where accelerating and decelerating effects of bluff body aerodynamics dominate. The computation overestimates the snow accumulation in the stagnation zone ahead of the cube where the flow is reasonably steady state, and is quite good behind it but the simulation authors wonder whether accumulation behind the cube is due to snowfall or snowdrift.

The VOF approach is a one-way coupling interface capturing method that treats the snow phase as a fluid and relies on the assumption that the fluids are not interpenetrating. Capturing of the interface between the phases is done by solving a continuity equation of one or more of the phases, in addition to the mixture continuity and momentum equations. The relative velocity of both phases is based on drift-flux theory, which assumes low drift [9], a reasonable assumption for smaller particles. The VOF approach relies on the same steady-state empirical equilibrium saltation flux treatment as the transport of snowdrift density approach and both methods are equivalent. Different attempts at improving the VOF approach by accounting for particle impingement in saltation as well as modifications of the turbulent wall function roughness parameter based on experiment-specific measurements did not show much improvement compared to experiment, especially close to bluff bodies where accelerating and decelerating flows dominate [10]. No decisive improvement was seen either by using mesh adaptation based on the balance of the convected horizontal snow flux and flow divergence at the ground [11]. Yet another application of the VOF approach uses two different snow phase continuity equations, one with mass diffusion and a suspension particle settling velocity in the suspension layer and another without mass diffusion and a saltation particle settling velocity in the saltation layer [12]. The implementation is based on ad hoc empirical coefficients and parameters, and does not improve on the previously mentioned limitations.

An exception to both approaches above is the physically based one by Gauer [13], which resolves the saltation layer but still uses considerable parameterization and self-similarity assumptions between the airflow profile and the snow concentration profile in the saltation layer, which does not necessarily hold true in the vicinity of

bluff bodies. In particular, the air velocity and snow concentration profiles in the saltation layer are assumed and related to the air velocity on top of the saltation layer obtained from the suspension layer computation. The simulations manage to capture the general trends in comparison to experimental snowdrift rates, wind field velocity, and new snow depth for an Alpine crest with order of magnitude agreement in qualitative comparisons. In all fairness, much of the discrepancies are due to poor terrain accuracy and large uncertainty in choosing the correct numerical boundary conditions, as pointed out by the author of the simulations, but the results are inconclusive nonetheless.

The objective of this paper is to present a viable snow phase viscosity model for high rates of strain and two-way coupled situations such as snow saltation, in Eulerian–Eulerian simulations of drifting snow. The viscosity model is implemented in a modified version of `twoPhaseEulerFoam` [14], the formulation of which is based on the following conditional ensemble-averaged equations of conservation of mass and linear momentum used to represent interpenetrating phases in the Gosman model [15]:

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot \left(\alpha_i \mathbf{u}_i\right) = 0, \tag{1}$$

$$\frac{\partial}{\partial t}\left(\alpha_i \mathbf{u}_i\right) + \nabla \cdot \left(\alpha_i \mathbf{u}_i \mathbf{u}_i\right) + \nabla \cdot \left(\alpha_i \mathbf{R}_i\right) = -\frac{\alpha_i}{\rho_i}\nabla p + \alpha_i \mathbf{g} + \frac{\mathbf{M}_i}{\rho_i}. \tag{2}$$

Here, α_i, ρ_i, $\mathbf{u}_i$, and $\mathbf{R}_i$ are the volume fraction, density, velocity, and stress tensor of phase i, respectively; p is the static pressure field; and $\mathbf{g}$ is the gravitational acceleration vector; $\mathbf{M}_i$ is the momentum exchange term between the phases,

$$\mathbf{M}_i = \mathbf{F}_l + \mathbf{F}_d + \mathbf{F}_t. \tag{3}$$

Here, $\mathbf{F}_l$, $\mathbf{F}_d$, and $\mathbf{F}_t$ are, respectively, the aerodynamic lift, generalized drag, and turbulent drag forces. Using scale analysis, [13] found the aerodynamic lift and drag forces to dominate at the onset of drifting, but did not consider the turbulent drag force. An earlier scale analysis by [15] including the turbulent drag force found the lift to be negligible for gas/solid particle-laden flows, where the ratio of continuous gas density to dispersed solid density is proportional to 10^{-3}. The lift force is given by

$$\mathbf{F}_l = \alpha_2 \alpha_1 (\alpha_2 C_{ls} \rho_2 + \alpha_1 C_{la} \rho_1) \mathbf{U}_{rel} \times \nabla \times \mathbf{U}. \tag{4}$$

Here, the snow phase is represented by index 1 and the air phase by index 2. $\mathbf{U}_{rel} = \mathbf{U}_2 - \mathbf{U}_1$ is the relative velocity between the phases, and $\mathbf{U} = \alpha_2 \mathbf{U}_2 + \alpha_1 \mathbf{U}_1$ is the mixture velocity. Numerical tests with the present model confirmed the lift force to be negligible; therefore, it was not used in the present simulations. The only two forces found relevant for saltation and suspension are then the generalized aerodynamic and turbulent drag forces. The latter force was also reported to be the main mechanism

for transporting smaller particles into suspension [1]. The generalized aerodynamic drag model used is the Gidaspow–Schiller-Naumann model [16], which is expressed as follows for the snow phase:

$$\mathbf{F_d} = K\mathbf{U}_{rel}, \tag{5}$$

$$K = \frac{3}{4d_p}\rho_2 C_D \alpha_2^{-1.65}(1-\alpha_2)|\mathbf{U}_{rel}|. \tag{6}$$

Here, C_D is the drag coefficient on a single sphere given by the following relationship [17]:

$$C_D = \begin{cases} \dfrac{24}{Re_p}(1+0.15Re_p^{0.687}) & \text{if } Re_p < 1000, \\ 0.44 & \text{if } Re_p \geq 1000. \end{cases} \tag{7}$$

Re_p is the particle Reynolds number based on the particle diameter d_p and air kinematic viscosity ν_2,

$$Re_p = \frac{\alpha_2|\mathbf{U}_{rel}|d_p}{\nu_2}. \tag{8}$$

The Gidaspow–Schiller-Naumann drag model is valid for dilute flows with $\alpha_2 > 0.8$ [18], which is the case in the creep, saltation, and suspension layers. Moreover, the Gidaspow–Schiller-Naumann drag model applies to spherical particles and is used here since no practical correlations for irregular particles as depicted in Fig. 4 are available in the literature. However, as the present irregular particles drift they will rotate around three axes within a somewhat spheroidal volume of air. Therefore, their drag function could be similar to that of a spherical particle with differences that cannot be predicted at the moment. It remains that the spherical particle drag correlations are the only present recourse.

For the snow phase, the turbulent component of the drag force, arising from turbulent fluctuations of the volume fractions and velocities in the Gosman two-fluid model is given by

$$\mathbf{F_t} = -K\frac{\nu^t}{\sigma_\alpha}\nabla\alpha_1. \tag{9}$$

Here, ν^t and σ_α are, respectively, the turbulent kinematic viscosity of the air phase and the Schmidt number. The standard formulation of `twoPhaseEulerFoam` does not include the turbulent drag term. Instead, it uses the continuity equation in the following form:

$$\frac{\partial\alpha_i}{\partial t} + \nabla\cdot(\mathbf{U}\alpha_i) - \nabla\cdot(\mathbf{U}_{rel}\alpha_i(1-\alpha_i)) = 0. \tag{10}$$

Equation 10 provides tighter coupling between the phases since it uses the mixture and relative velocities, as well as both volume fractions [19]. It does not include a turbulent diffusion term, but the third term on the left-hand side can be consid-

ered a volumetric mass flow rate source term, playing the same role as a turbulent diffusion term, and a similar role to the turbulent drag term in the momentum equation. For the present simulations, the turbulent drag term of Eq. 9 was added to `twoPhaseEulerFoam`, while retaining the treatment of Eq. 10. The standard incompressible k–ϵ turbulence model is used unmodified and applied only to the air phase, so the interaction between the snow phase and turbulence is not directly taken into account.

The viscous stresses terms in the momentum equations are modeled according to the Boussinesq formulation, which requires a viscosity parameter readily available for air but not for snow. Many snow compactive viscosity models are available for very low rates of strain typical of settling snow, the latest by Teufelsbauer [20] who also provides a review of the main models in the literature. However, nothing is available at the high rates of strain of drifting snow. The next section discusses the high rate of strain viscosity model and its derivation, while the validation of the formulation is presented in the Validationsection. The relevant details of the controlled drifting snow experiment are presented in the Validation Experiment subsection. The numerical setup is presented in the Simulation Setup subsection, followed by the Results and Discussion subsection. The paper concludes with the Conclusionssection which includes recommendations for future work.

2 The Drifting Snow Viscosity Model

The snow phase viscosity is derived by matching the momentum of a number of ideal spherical drifting particles within a control volume, with the momentum of the same control volume containing an equal amount of the equivalent viscous fluid. Drifting snow particles move in transient hops and bounce over the snowbed surface. However, drifting snow can easily be observed in self-sustained steady-state mode in natural and controlled environments, so the motion of the spherical particles can be considered steady state *in the average sense*. This approximation is only used for the purpose of deriving an expression of the snow phase Newtonian viscosity model. A scale analysis showed that the rolling friction force is negligible compared to the drag force, so the former was not retained in the analysis. Neglecting friction forces is further justified by the fact that their effects and that of snowbed asperities are already implicitly included in the surface threshold shear–stress parameter.

The derivation starts with the momentum equations of the air and snow phase for fully developed steady-state flow in a control volume containing a number of rolling particles on the snowbed and having the same height as a particle. At typical drifting snow particle height $d_p \leqslant 1\,\text{mm}$, typical surface threshold friction velocity $u_* \leqslant 0.5$ [21] and air temperature below freezing, the nondimensional wall distance to the snowbed is $y_+ \leqslant 50$. Under such conditions the airflow profile is weakly nonlinear and the divergence of the stress tensor negligible compared to the pressure gradient. Therefore, we can write the snow and air momentum equations for steady-state fully developed flow as

$$-\alpha_1 \frac{\partial P}{\partial x} + F_d + \alpha_1 \mu_1 \frac{d^2 u_1}{dy^2} = 0, \tag{11}$$

$$-\alpha_2 \frac{\partial P}{\partial x} - F_d = 0. \tag{12}$$

Here, $\partial P/\partial x$ is the downstream pressure gradient, F_d is the drag force on a particle and μ_1 is the snow phase dynamic viscosity. One can eliminate the drag force between the equations above and solve for the snow phase velocity on the snowbed using the following no-slip and threshold shear stress boundary conditions at the snowbed surface:

$$u_1 = 0, \tag{13}$$

$$\tau_t = \alpha_2 \rho_2 u_*^2. \tag{14}$$

The resulting snow phase velocity on the snowbed is

$$u_1(y) = \frac{1}{2\alpha_1 \mu_1} \frac{\partial P}{\partial x} y^2 + \frac{\tau_t}{\alpha_1 \mu_1} y. \tag{15}$$

The expression for the dynamic viscosity is obtained by matching the linear momentum of the Lagrangian snow particle phase with that of the equivalent Eulerian snow fluid phase. The Lagrangian linear momentum per unit volume $P_{L,v}$ is given by Eq. 16, where ρ_i is the ice density and V_p the average particle velocity. The Eulerian linear momentum per unit volume $P_{E,v}$ is given by Eq. 17.

$$P_{L,v} = \alpha_1 \rho_i V_p, \tag{16}$$

$$P_{E,v} = \alpha_1 \rho_i d_p \int_0^{d_p} u_1(y) dy. \tag{17}$$

Integrating and setting $P_{L,v} = P_{E,v}$ provides the following expression of the drifting snow dynamic viscosity:

$$\mu_1 = \frac{\frac{1}{6} \frac{\partial P}{\partial x} d_p + \frac{1}{2} \tau_t}{\dot{\gamma}_1}. \tag{18}$$

Here, $\dot{\gamma}_1 = \alpha_1 \dfrac{V_p}{d_p}$ is the particle phase rate of strain. Equation 18 can also be reformulated in terms of the drag force,

$$\mu_1 = \frac{-\frac{1}{6}\frac{F_d d_p}{\alpha_2} + \frac{1}{2}\tau_t}{\dot{\gamma}_1}. \tag{19}$$

In aeolian transport phenomena particles, the drag force usually points downstream, in the direction of the decreasing downstream pressure gradient. On the other hand, the surface shear stresses usually resist the particle motion, and this competition between the threshold shear stress τ_t and the pressure gradient (or drag force) is highlighted in Eqs. 18 and 19. The pressure gradient and the drag force tend to induce motion, reducing the effective viscosity of the snow phase, whereas surface shear stresses tend to inhibit motion, increasing the effective viscosity of the snow phase. Within the snowbed, the snow Eulerian continuum should still be characterized by the threshold shear stress. Since there is no significant airflow beneath the snow surface, and we are not interested in an accurate simulation of snowbed packing, the drag/pressure gradient term can be eliminated from Eqs. 18 and 19 in that region. The resulting drifting snow viscosity expression implemented and tested here is the following:

$$\mu_1 = \begin{cases} 0.5\frac{\tau_t}{\dot{\gamma}_1} & \text{in snowbed} \\ \left(-\frac{1}{6}\frac{F_d d_p}{\alpha_2} + \frac{1}{2}\tau_t\right)/\dot{\gamma}_1 & \text{in creeping, saltation and suspension.} \end{cases} \tag{20}$$

3 Validation

This section presents the relevant details of the controlled wind tunnel drifting snow experiment and the numerical setup. The discussion proceeds around the results of the numerical simulations as compared to the experimental measurements.

3.1 Validation Experiment

The experimental data used to validate the present viscosity model comes from a controlled wind tunnel experiment of drifting snow using actual snow particles [22]. The experiment was carried out at the Cryospheric Environment Simulator (CES) of the Shinjo Branch of the Snow and Ice Research Center (SIRC), at the National Research Institute for Earth Science and Disaster Prevention (NIED) in Japan, by the snow research group of Tohoku University. This experiment was selected because it included detailed measurements of the snow fluxes and airflow velocity profiles at four measurement stations in the working section of the tunnel, and across the

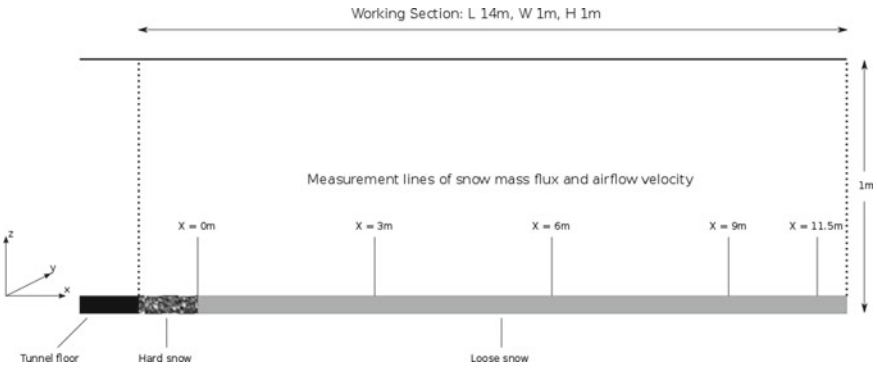


Fig. 2 Side view of the wind tunnel experimental layout (adaptation of Fig. 2 from Okaze et al. [22])

entire saltation layer and lower part of the suspension layer. Turbulent kinetic energy profiles were also measured at the most downstream measurement station.

The experimental layout is shown in Fig. 2 with the locations of the experimental measurement stations. The first measurement station is at $X = 0$ m and is preceded by a 1 m fetch of hardened snow that cannot drift. This induces a nonequilibrium boundary layer before the 14 m working section which includes a 0.02 m deep groove filled with loose snow that can drift. The turbulence kinetic energy and airflow velocity profiles were measured at X = 0 m, and they are shown in Fig. 3. They are nondimensionalized using the reference airflow velocity U_r at a height of 0.2 m over the snowbed. The airflow velocity and snow flux profiles were also measured at the downstream stations located at $X = 3$, 6, 9 and 11.5 m. The turbulent kinetic energy profile was also measured at the last downstream station at $X = 11.5$ m. The experiment was initiated with a start-up phase of 25 s to reach the conditions shown in Fig. 3, followed by a waiting period of 5 s. The airflow velocity, turbulent kinetic energy, and snow flux profiles were then measured at all four downstream stations within a global time window of 30 s, consisting of 5 s for each station and a 2 s transfer period in between stations. The reported results of snow flux, airflow velocity, and turbulent kinetic energy profiles were obtained by averaging the data measured during the individual 7-s windows. The reader is referred to Okaze et al. [22] for the experimental details. Samples of the snow particles used in the experiment are shown in Fig. 4 with a 1 mm scale bar. The experimental snow particles are quite irregular and bulky, exceeding 1 mm in length quite often but rarely smaller than 0.10 mm in either length or width.

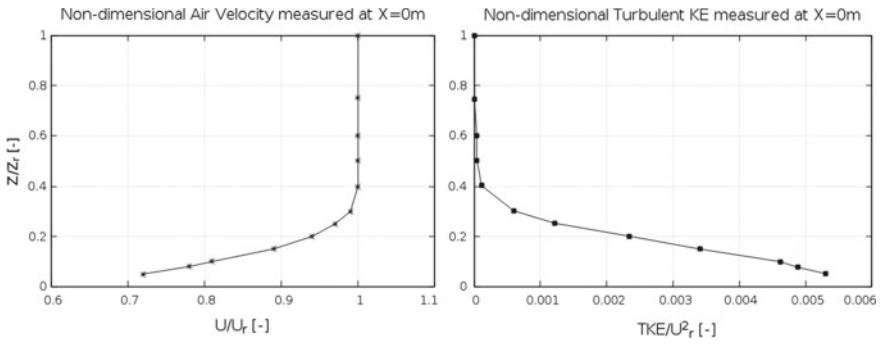


Fig. 3 Experimental airflow velocity and turbulent kinetic energy profiles measured at $X = 0$ (adapted from Fig. 3 of Okaze et al. [22])

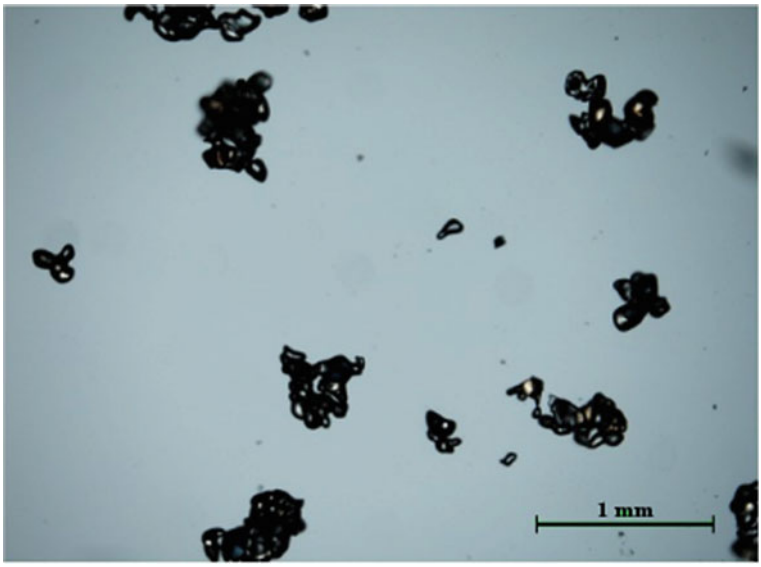


Fig. 4 Samples of the snow particles used in the experiment with a 1 mm scale bar (provided by Dr. Tsubasa Okaze)

3.2 Simulation Setup

The 2D computational mesh used for the simulations is shown in Fig. 5. A close-up of the mesh at the inlet of the computational domain at $X = 0$ m is also shown, with the loose snow layer in the gutter in white. The volume fraction of the snowbed was set to 0.394 in order to match the experimentally measured snowbed density of 361 kg/m^3. The mesh is fully structured, composed of hexahedral elements with a transverse element size of 4 mm in the gutter and at the top of the tunnel. The

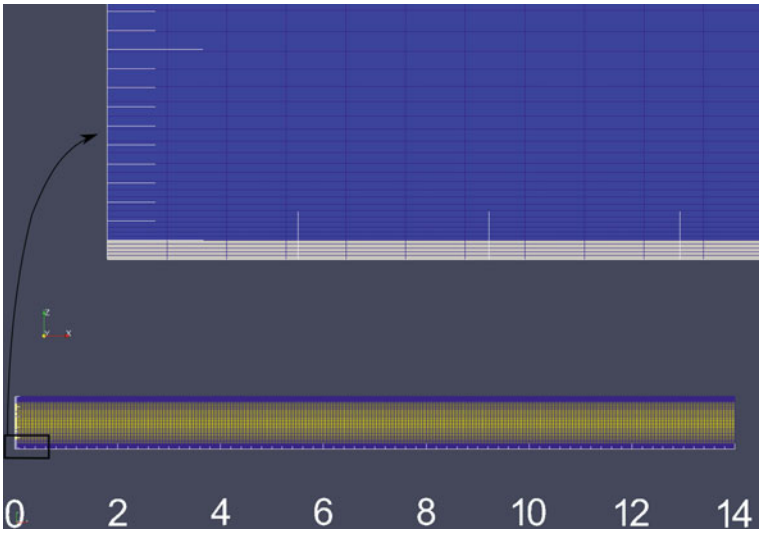


Fig. 5 Simulation mesh with a close-up of the inlet region showing the loose snow layer in the gutter

longitudinal element size in the flow direction along the X-axis is about 6 cm. Tests were carried out with a mesh twice finer in both directions and the results varied by less than 15%, so the results obtained from the present mesh can be reasonably considered mesh-independent.

The `Gauss linear` scheme was used for gradients and divergence of viscous terms, and the `Gauss upwind` scheme for divergence of nonlinear convection terms. The `Gauss linear orthogonal` scheme was used for all Laplacians. The pressure equation was solved with GAMG and DIC smoother, and all other equations were solved with PBiCG and DILU preconditioner. All solvers were converged to ten orders of magnitude. The measured profiles of airflow velocity and turbulent kinetic energy at $X = 0$ m from Fig. 3 were imposed as inlet boundary conditions for the simulations, with a Neumann zero-gradient inlet boundary condition for the pressure. At the inlet, the `turbulentMixingLengthDissipationRateInlet` boundary condition was used for ϵ, which is based on the following equilibrium relationship:

$$\epsilon = \frac{C_\mu^{0.75} k^{1.5}}{l_m}. \tag{21}$$

Here, $C_\mu = 0.09$ is the familiar k–ϵ model constant and l_m is the mixing length set to half the wind tunnel height, or the assumed dimension of the large inertial eddies. Tests were conducted with a mixing length equal to 10% of the wind tunnel height with negligible differences in the results, perhaps due to the low sensitivity of the k–ϵ model to changes in inlet conditions. At the outlet, Neumann zero-gradient boundary

conditions were used for all variables except the pressure, with a Dirichlet zero-value boundary condition. Standard wall functions were used at the walls.

Simulations were carried out at a snow threshold shear stress $\tau_t = 0.044\,\mathrm{Kg/(m\,s^2)}$, which is the minimum experimentally observed value for drifting, and corresponds to a threshold velocity of $u_* = 0.23\,\mathrm{m/s}$. The drifting experiment analyzed here is transient, given the limited supply of drifting snow in the wind tunnel gutter, so the simulations were accordingly carried out in transient mode. Moreover, it is necessary to take into account the particle size distribution when using two-way coupled simulations [23]. The present formulation can only account for one particle size at a time. Therefore, the only way to reproduce the results of a particle size distribution was to combine the results of several single particle size simulations, using the statistical weight of each size class in the distribution. Particle size distributions were not reported in the experimental paper but mechanical breakage phenomena such as drifting snow usually obeyed a two-parameter Gamma distribution be it as aggregate on the ground [24] or drifting above it [25]. The two-parameter Gamma Probability Distribution Function (PDF) $f(x)$ and the Gamma function Γ are expressed as

$$f(x) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}, \tag{22}$$

$$\Gamma(a) = \int_0^\infty x^{a-1} e^{-x} dx, \qquad a \in (0, \infty). \tag{23}$$

Here, a and b are, respectively, the distribution shape and scale parameters, and they define the distribution average size as $d_{avg} = a/b$. The statistical weight w_i of a particle class d_i is calculated as follows:

$$w_i = \int_{\frac{d_i + d_{i-1}}{2}}^{\frac{d_{i+1} + d_i}{2}} f(x) dx. \tag{24}$$

Here, d_{i-1} and d_{i+1} are the lower and upper particle size classes. The percentage contributions of the different particle classes to several distributions with average particle size of 0.7, 0.8, and 0.9 mm are shown in Fig. 6. In this section, the snow flux and airflow velocity profiles of a two-parameter Gamma distribution with average diameters of 0.7, 0.8, and 0.9 mm are reproduced using all seven particle size simulations, namely, 0.1, 0.3, 0.5, 0.7, 0.9, 1.1, and 1.3 mm particle sizes. The particle size distribution results are then compared to the results of the single diameter distributions from the previous section and the experimental measurements.

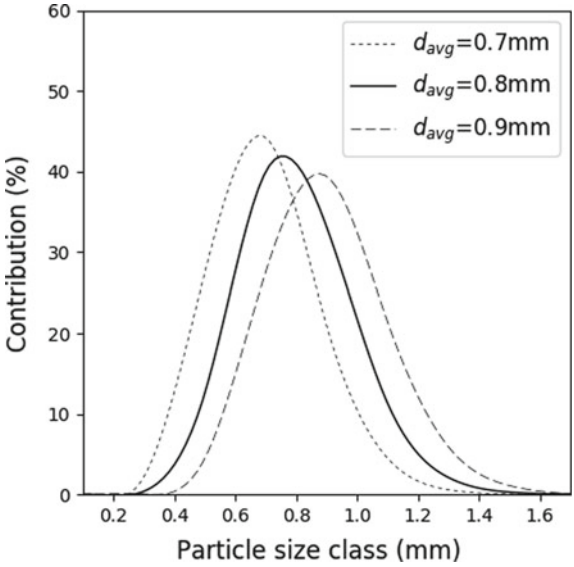


Fig. 6 Percentage contributions of different particle size classes to distributions with d_{avg} = 0.7, 0.8 and 0.9 mm

3.3 Results and Discussion

The average snow flux profiles at X = 11.5 m are shown in Fig. 7, along with the experimental measurements. The numerical profile for the average diameter of 0.7 mm exceeds the experimental measurements by far, especially high in the saltation and suspension layers. This is due to the important contributions of the smallest particles that are most present in that distribution. The smallest and lightest particles are transported in saltation and suspension more easily than the larger particles, which are heavier and tend to drift closer to the snowbed. Accordingly, the distributions with average diameters of 0.8 and 0.9 mm have less contribution from the smallest particles and less saltation/suspension snow flux. The distribution snow flux profiles appear quite sensitive to small changes in the average distribution diameter since the PDF curves in Fig. 6 are pretty narrow and have little spread around the average diameter value. However, all three profiles show a reasonable qualitative agreement with the experimental data since the shapes of the experimental and simulation profiles are similar and the simulation profiles intersect the experimental profile, especially for the 0.9 mm average diameter case.

The profiles of average nondimensional airflow velocity for the same average diameters of 0.7, 0.8, and 0.9 mm are shown in Fig. 8. Again, a reasonable qualitative agreement is found with the experimental measurements since the shapes of all curves are pretty much the same, but with much smaller differences between the three distributions. This implies that the distribution velocity is less sensitive than the distribution snow flux to small changes in the average distribution diameter, for

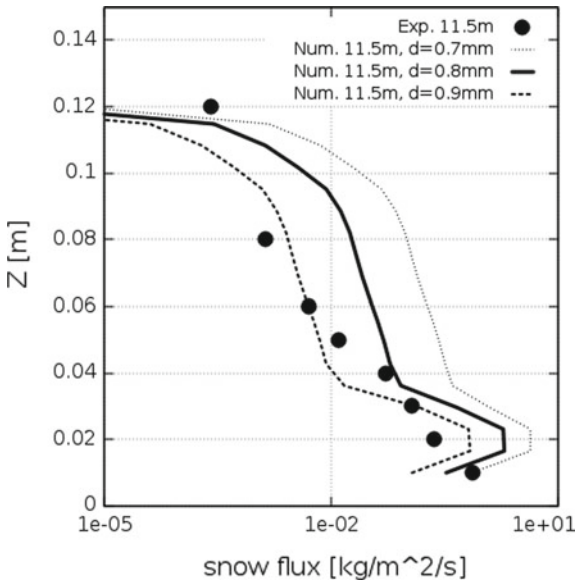


Fig. 7 Average snow flux profiles at $X = 11.5$m, for distribution average diameters of 0.7, 0.8, and 0.9 mm

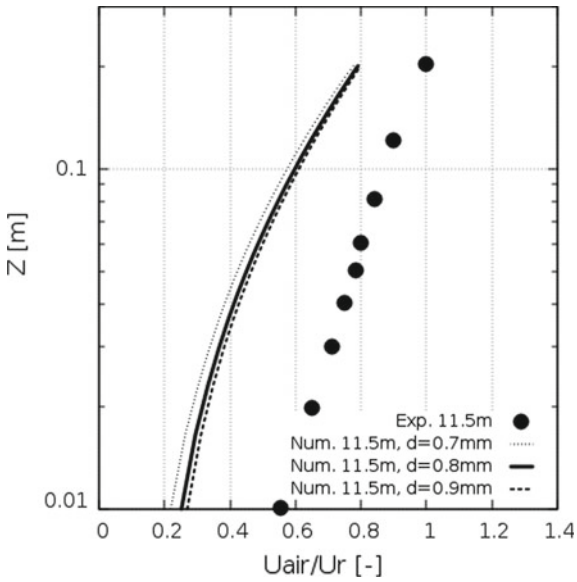


Fig. 8 Average profiles of nondimensional airflow velocity at $X = 11.5$m, for distribution average diameters of 0.7, 0.8, and 0.9 mm

the range of average distribution diameters considered. Therefore, the quantitative differences should not be due to the averaging process.

The average nondimensional turbulent kinetic energy profiles are shown in Fig. 9 and are found to be equally insensitive as the airflow velocity profiles. Moreover, the numerical results exceed the experimental measurements by two orders of magnitude,

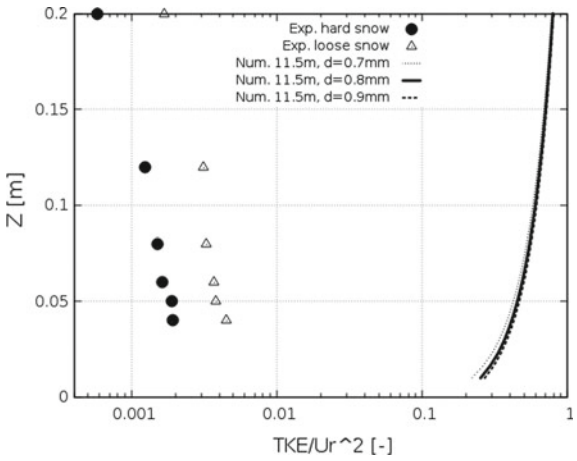


Fig. 9 Average profiles of nondimensional turbulent kinetic energy at X = 11.5m, for distribution average diameters of 0.7, 0.8, and 0.9 mm

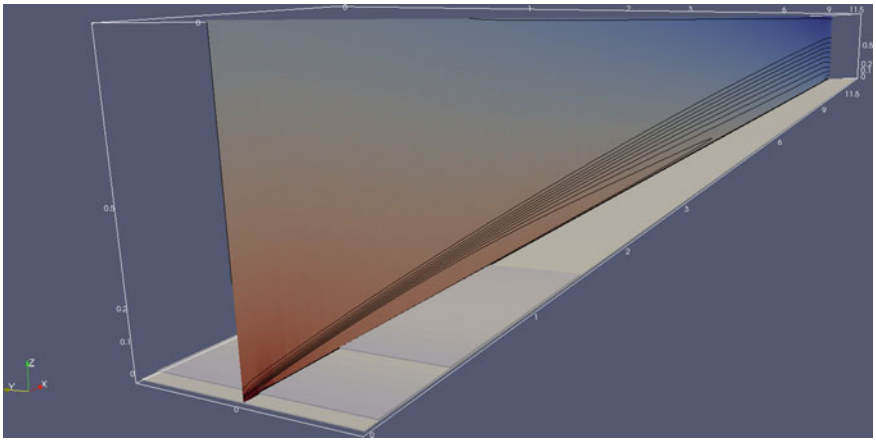


Fig. 10 Pressure stagnation zone forming at the beginning of the eroding snowbed, with turbulent kinetic energy contours

a well-known flaw of the k–ϵ model in stagnation regions [26, 27]. This deficiency is also known to affect downstream parts of the flow [28]. In the present case, the stagnation region is located upstream right after the inlet, where the flow trips into the eroding snowbed as can be seen in Fig. 10. The turbulent kinetic energy contours are also shown in black, and they extend downstream reaching the 11.5 m measurement station. The drifting snow particles constitute another contributing factor since they extract significant momentum from the airflow by drag in the saltation layer. This two-way coupled effect fades away in the suspension layer, where the airflow subsequently accelerates, forming another shear layer where turbulence is also generated [1]. The

standard k–ϵ model used here is not equipped to handle such dispersed two-phase flow situations. However, many fixes are available in the literature in the form of turbulent timescale limiters [28] and particle effect source terms in the k and ϵ transport equations [29]. They will be investigated in future work, as well as the specific k–ϵ implementation in `twoPhaseEulerFoam`.

4 Conclusions and Future Work

A new two-way coupled Eulerian–Eulerian formulation to simulating drifting snow was presented along with validation results against a controlled wind tunnel drifting snow experiment. The new formulation was implemented in OpenFOAM®, based on `twoPhaseEulerFoam`. It includes a novel drifting snow viscosity model developed to allow computing the viscous stress tensor in the snow phase momentum equation, which itself made it possible to simulate drifting snow in the saltation layer without resorting to empirical correlations used by other Eulerian methods.

Comparisons of simulation results with experimental measurements showed that the new two-way coupled formulation behaves physically with respect to particle size. The model showed reasonable qualitative agreement with measured snow flux in the saltation and suspension layers. Comparisons of numerical airflow velocity to experiment showed reasonable qualitative agreement with the experimental profiles, with quantitative deficit in the numerical results. These are believed to be due to well-known deficiencies of the standard k–ϵ model, which can be addressed with fixes available in the literature.

The present formulation allows simulating one particle diameter class, and work is ongoing to allow simulating several diameter classes simultaneously. Corrections to the k–ϵ model excess turbulent kinetic energy production are also being implemented. This viscous model can be extended quite easily to aeolian transport of sand and even riverbed sediment transport, to give only a couple of examples.

Acknowledgements The authors warmly thank Profs. Akashi Mochida, Tsubasa Okaze, and Yoshihide Tominaga for sharing their experimental results. In particular, the patience and dedication of Prof. Okaze to answering our numerous questions are gratefully acknowledged. Many thanks!

References

1. R. Bagnold, *The physics of blown sand and desert dunes*. London, Methuen, 1941.
2. B. Lee, J. Tu, and C. Fletcher, "On numerical modeling of particle-wall impaction in relation to erosion prediction: Eulerian versus Lagrangian method," *Wear*, vol. 252, pp. 179–188, 2002.
3. T. Uematsu, T. Nakata, K. Takeuchi, Y. Arisawa, and Y. Kaneda, "Three-dimensional numerical simulation of snowdrift," *Cold Regions Science and Technology*, vol. 20, pp. 65–73, 1991.
4. M. Naaim, F. Naaim-Bouvet, and H. Martinez, "Numerical simulation of drifting snow: erosion and deposition model," *Annals of Glaciology*, vol. 26, pp. 191–196, 1998.

5. Y. Tominaga and A. Mochida, "CFD prediction of flowfield and snowdrift around a building complex in a snowy region," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 81, no. 13, pp. 273 – 282, 1999.
6. J. Pomeroy and D. Gray, "Saltation of snow," *Water Resources Research*, vol. 26, no. 7, pp. 1583–1594, 1990.
7. T. Okaze, A. Mochida, Y. Tominaga, M. Nemoto, Y. Ito, and T. Shida, "Modeling of drifting snow development in a boundary layer and its effect on windfield," in *The Sixth Snow Engineering Conference*, Whistler, B.C., Canada, June 1–5 2008.
8. Y. Tominaga, T. Okaze, and A. Mochida, "CFD modeling of snowdrift around a building: overview of models and evaluation of a new approach," *Building and Environment*, vol. 46, pp. 899–910, 2011.
9. B. Bang, A. Nielsen, P. Sundsbø, and T. Wiik, "Computer simulation of wind speed, wind pressure and snow accumulation around buildings (SNOW-SIM)," *Energy and Buildings*, vol. 21, no. 3, pp. 235–243, 1994.
10. J. Beyers, "Numerical modeling of the snowdrift characteristics surrounding the SANAE IV research station," Ph.D. Dissertation, Department of Mechanical Engineering, University of Stellenbosch, 2004.
11. J. Beyers and B. Waechter, "Modeling transient snowdrift development around complex three-dimensional structures," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 96, pp. 1603–1615, 2008.
12. P. Sundsbø, "Numerical simulations of wind deflection fins to control snow accumulation in building steps," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 74–76, pp. 543–552, 1998.
13. P. Gauer, "Blowing and drifting snow in alpine terrain: A physically-based numerical model and related field measurements," Ph.D. dissertation, ETH Zurich, 1999.
14. *OpenFOAM Documentation, Extended Code Guide*, 2018.
15. A. Gosman, R. Issa, C. Lekakou, S. Politis, and M. Looney, "Multidimensional modeling of turbulent two-phase flows in stirred vessels," *AIChE Journal*, vol. 38, no. 12, pp. 1946–1956, 1992.
16. D. Gidaspow, "Hydrodynamics of fluidization and heat transfer: supercomputer modelling," *Appl. Mech. Rev.*, vol. 39, pp. 1–22, 1986.
17. L. Schiller and Z. Naumann, "A drag coefficient correlation," *Z. Ver. Deutsch. Ing.*, vol. 77, 1935.
18. H. Enwald, E. Peirano, and A.-E. Almstedt, "Eulerian two-phase flow theory applied to fluidization," *Int. J. of Multiphase Flow*, vol. 22, pp. 21–66, 1996.
19. H. Weller, "Derivation, modelling and solution of the conditionally averaged two-phase flow equations," OpenCFD Ltd, Report TR/HGW/02, 2005.
20. H. Teufelsbauer, "A two-dimensional snow creep model for alpine terrain," *Natural Hazards*, vol. 56, pp. 481–497, 2011.
21. R. Kind, *Handbook of Snow, Principles, Processes, Management and Use*. Pergamon Press, 1981, ch. Snowdrifting, pp. 338–359.
22. T. Okaze, A. Mochida, Y. Tominaga, M. Nemoto, T. Sato, Y. Sasaki, and K. Ichinohe, "Wind tunnel investigation of drifting snow development in a boundary layer," *J. Wind Eng. Ind. Aerodyn.*, vol. 104-106, pp. 532–539, 2012.
23. Z. Boutanios and H. Jasak, "Viscous treatment of the snow phase in Eulerian-Eulerian simulations of drifting snow," in *The 14th International Conference on Wind Engineering*, Porto Alegre, Brazil, June 21-26 2015.
24. W. Budd, "The drifting of nonuniform snow particles," in *Studies in Antarctic meteorology*, M. Rubin, Ed. American Geophysical Union, 1966.
25. R. Schmidt, "Vertical profiles of wind speed, snow concentration and humidity in blowing snow," *Boundary-Layer Meteorology*, vol. 23, no. 2, pp. 223–246, 1982.
26. W. Strahle, "Stagnation point flows with freestream turbulence – the matching condition," *AIAAJ*, vol. 23, pp. 1822–1824, 1985.

27. B. Launder and M. Kato, "Modeling flow-induced oscillations in turbulent flow around square cylinder," in *ASME Fluid Eng. Conference*, 1993, p. 20.
28. P. Durbin, "Separated flow computations with the k–epsilon-v2 model," *AIAA Journal*, vol. 33, pp. 659–664, 1995.
29. T. Okaze, Y. Takano, A. Mochida, and Y. Tominaga, "Development of a new $k - \epsilon$ model to reproduce the aerodynamic effects of snow particles on a flow field," *J. Wind Eng. Ind. Aerodyn.*, vol. 144, pp. 118–124, 2015.

Use of OpenFOAM® for the Investigation of Mixing Time in Agitated Vessels with Immersed Helical Coils

Alexander Stefan and Heyko Juergen Schultz

Abstract This chapter deals with the investigation of potentials in energy efficiency optimization for widespread agitated vessels. A lab-scale model is derived from an industrially used reactor vessel with immersed helical coils, which is utilized for several chemical basis operations. The model is analyzed with particle image velocimetry (PIV) and laser-induced fluorescence (LIF) concerning velocity and concentration fields, which gives a good validation basis for CFD analysis. However, it is challenging to validate simulations of industrial reactors. In this work, the idea is pursued of comparing the flow fields of simulations and measurements in order to validate the computational results. The simulation task implies the generation of complex geometry meshes, solving for steady-state, as well as for transient solutions, and seeking fast and effective methods. An approach to the validation of technical, large-scale simulation results is proposed through comparison of mixing times in simulations and industrial trial runs.

1 Computational Fluid Dynamics in the Chemical Industry

Fluid dynamics is omnipresent in the chemical industry. Processes of arbitrary complexity involve handling material and energy fluxes. Several works serve as a basis for practical planning and the designing of instruments and apparatuses for production plants. In compliance with given requirements, dimensioning and designing are performed according to pertinent specifications given in standards, monographs, or guidelines. Process simulations are very common in the chemical industry, too. Due to high complexity, simplified and semiempirical models were developed for commonly used apparatuses. Those models are the basis for underlying algorithms in

A. Stefan · H. J. Schultz (✉)
Faculty of Chemistry, University of Applied Sciences Niederrhein, Krefeld, Germany
e-mail: heyko_juergen.schultz@hs-niederrhein.de

A. Stefan
e-mail: stalexander@gmx.de

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®* ,
https://doi.org/10.1007/978-3-319-60846-4_36

process simulation software. A multitude of software applications serve as an aid for the design and operation of chemical plants.

Computational fluid dynamics simulation tools, however, still occupy a niche in the chemical industry. The big success of CFD in, e.g., aviation or the racing industry is explainable by the fact that it optimizes the final product. In the chemical industry, on the other hand, the optimized object is a tool itself. Due to high implementation effort, as well as the computational and financial costs of commercial CFD, chemical engineers still mostly prefer well-established process simulation tools. As an open and free-to-use CFD tool, OpenFOAM® [1, 2] poses an economic solution and represents huge, previously hidden potential for the optimization of chemical industry utilities, which is demonstrated in this contribution with the example of a stirred process.

1.1 Agitated Vessels in the Chemical Industry

In process engineering, and especially in the chemical industry, agitated vessels are commonly used for a multitude of applications. They have therefore been investigated in numerous papers and monographs [3–14]. Typical basic operations are heating, homogenizing, suspending, blending and chemical reactions. For an appropriate design and the effective operation of agitated vessels, it is necessary to understand the hydrodynamics and the transfer phenomena under the given circumstances. In previous works, good models for different aspects of stirred processes have been given for vessels with and without baffles. It is, however, not possible to formulate a universal model for different design scales of vessels; foremost, not for different equipment and internals. It is therefore necessary to look at each case individually. Experimental investigation quickly reaches its limits concerning applicability and costs. At this point, numerical simulation appears to be of large value. However, the results of numerical calculations have to be validated and verified.

This work follows the idea of developing valid CFD methods and simulation procedures for lab-scale mock-ups, which can be validated using measurement technology. These procedures can then be transferred to the industrial scale for the final optimization task.

2 Heat Exchange in Stirred Vessels

To achieve or maintain a certain temperature in the reactor, it is often necessary to install a heat exchanger inside or on the wall of the vessel. The standard types are diverse types of jacket heating, meander and register pipe, heating plug and helical coil in single or double variant [3]. Thanks to their low complexity and cost, huge exchange area and possibility for simple fixation (on most current baffles at any rate),

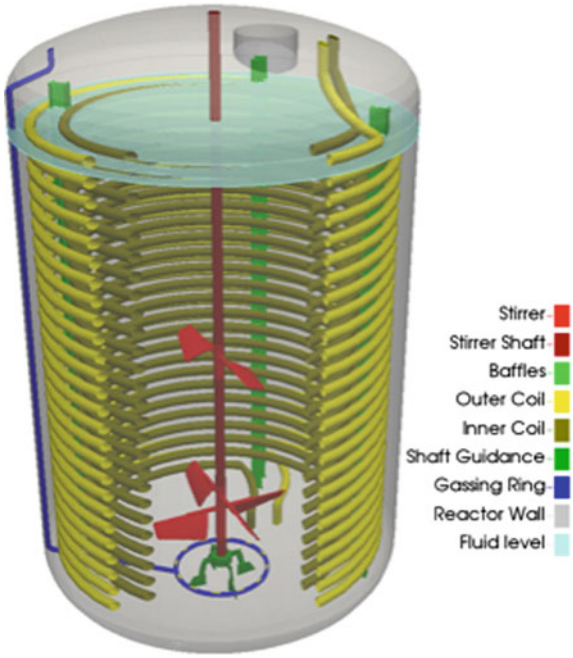


Fig. 1 Typical industrial reactor with two immersed helical coils

the latter option is very popular in the chemical and biotechnical industries. Figure 1 shows a typical industrial-stirred reactor with double helical coils and four baffles.

The disadvantage of immersed helical coils is the obstacle they pose for fluid dynamics. The works of Bliem and Schultz [15, 16] detail the problems that are caused by coils in regard to velocity fields and heat transfer. Anyhow, a CFD study showed that heat transfer could be improved by up to 24% for the investigated object through optimization of the stirrer position [17].

3 Investigated Object

The simulation study is conducted following the design of the experimental model, which serves as basis for validation. Up to now, little research has been published about vessels with internals, due to the very elaborate measurement approach. For noninvasive optical methods, complex geometries pose a challenging obstacle concerning measurement accuracy. Those problems are caused by image distortion and ray diffraction. This can be circumvented by building the model out of a translucent material. The refractive index of the fluid can then be adapted to it, so that geometries inside the observed domain become invisible and do not disturb the measurement. This method is described in detail in [12]. Bliem investigated the velocity fields via particle image velocimetry (PIV) [16]. The mixing time investi-

Table 1 Surface refinement levels for the snappyHexMesh process

Geometry surface	Reactor wall	Interphase	Stirrer	Baffles	Heater coil	Rigid cylinder
Minimal level	1	1	2	1	2	3
Maximum level	1	1	2	1	2	3

For regions where the surface feature angle is less than 30°, the maximum refinement level is used, otherwise the minimum level

gations were performed by Hirtsiefer via laser-induced fluorescence (LIF) and can be reviewed in [18]. The mixing time was determined by using rhodamine B as the tracer substance. A 10 L beaker glass was used as the vessel, the coil was made out of polymethylmethacrylate (PMMA), and ammonium thiocyanate solution was utilized as the refractive index matched (RIM) liquid. In the scope of this work, high Reynolds numbers between 16,000 and 32,000 were used.

On the simulative side, the described geometry is modeled with the free CAD software Blender® and exported in stl-format. A hexahedral mesh with cells of 5 mm edge length is created with the OpenFOAM® utility blockMesh. It is used for refining and geometry-snapping with the snappyHexMesh utility. A rigid cylinder body is implemented around the stirrer geometry to account for the rotational movement. The local surface refinement levels used in the snappyHexMesh process are listed in Table 1.

Three orthogonal cell layers were added around the heater geometry to suppress numerical diffusion. The same mesh quality controls as in the default implementation of openfoam4 were used for this study. The resulting mesh is presented in Fig. 2.

For simplification, the interphase between gas and liquid is assumed to satisfy slip conditions. This allows for monophasic simulation and is justified by relatively low vortex formation due to the baffles and coil, as, in the scope of the rotating frequencies used, can be demonstrated in experiments and two-phase simulations. Depending on the stirrer used, the final mesh has a size of 1.0–1.4 million cells.

Due to the high Reynolds numbers, a RAS-model (k-epsilon) was used for the simulations.

4 Measurement Approach

4.1 Velocity Field via Particle Image Velocimetry (PIV)

PIV is a rather modern noninvasive optical measurement method for velocity fields. In general, a slice of the flow domain is illuminated via a fanned-out laser beam. Two

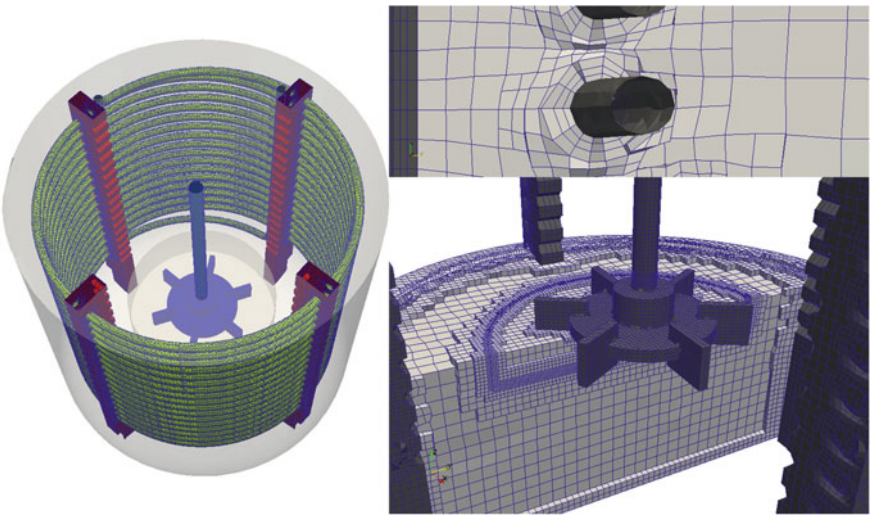


Fig. 2 Details of a snappyHexMesh-generated mesh of a lab-scale reactor with immersed helical coils and a Rushton turbine impeller

images are taken consecutively and the movement of tracer particles is determined by cross-correlation.

The comparison of simulation and measurement data shows good correlation concerning the main flow pattern, as well as transient structures like moving vortices and the fluctuation of the jet stream that is ejected from the stirrer.

The undeniable disadvantage is the limited image area. In standard PIV, only the velocity components inside the plane, but not the so-called out-of-plane component, are detected. This is to be considered when comparing two-dimensional PIV velocity magnitudes with three-dimensional CFD data. The CFD data therefore has to be projected onto the observed plane first. Better results can be achieved with stereo-PIV, in which images are created simultaneously from different perspectives. The out-of-plane component can then be calculated from the angle of the cameras and the differences between the respective images. For more information, see [12].

4.2 Concentration Field via Laser-Induced Fluorescence (LIF)

LIF uses the same experimental setup as PIV. The principle, however, is quite different. A fluorescent substance is used as a tracer. For mixing time investigations, rhodamine B is a good choice, since it is soluble in water and does not noticeably change the viscosity. Thus, its concentration may be considered to be governed by

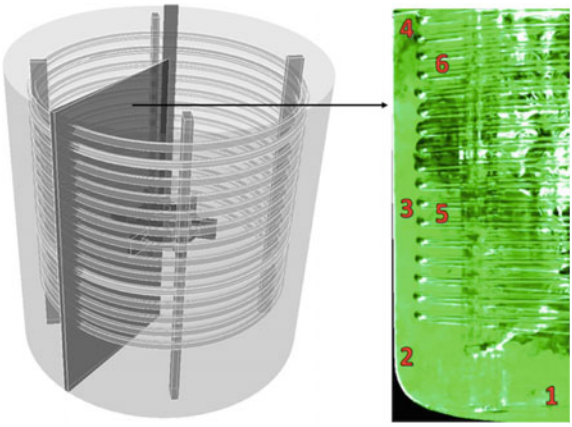


Fig. 3 Probe locations in a vertical slice of a stirred vessel geometry for determination of mixing time via LIF technology

scalar transport. The local intensity of fluorescence can be correlated to the local concentration, which is the basis for determination of mixing time.

Hirtsiefer [18] picked six points in the image for observation of concentration progress. Those points are pictured in Fig. 3.

5 Mixing Time

Mixing time is one of the central feature sizes in a stirred process. It has fundamental influence on both material and thermal transport phenomena. Its value is the basis for process design and the key to determination of the operating parameters and estimation of the energy requirement. However, experimental determination is highly individual for each problem, and even each experimental setup. Nonetheless, it is very important to ensure accurate projection of the experiment to the numerical calculation.

5.1 Definition of Mixing Time

The local concentration $c(x)$ of a tracer substance, where x is the spatial variable, converges to its final concentration c_∞ during the mixing process. The latter can be calculated from the tracer quantity and vessel volume. The current local concentration, however, fluctuates even after reaching a homogenous mixture. It is therefore necessary to specify the relative variance v of the concentration that it is supposed to satisfy the goodness of mixing $M_v = [M_v - v, M_v + v]$. It is common to use a value of $v = 5\%$ [7]. The current local goodness of mixing $M(x, t)$ is defined as the ratio

between $c(x, t)$ and c_∞. The mixing time t_m is defined as the point when $M(x, t)$ is inside the interval M_v for all $t > t_m$ and all x in the domain.

In practice, $c(x, t)$, or any correlated signal, is plotted against time, where $t = 0$ marks the tracer injection time. The final concentration is calculated as a mean value of the signal within a time range in which the mixing time is obviously exceeded. A typical plot is shown in Fig. 5. In this case, numerical data was used. However, the local concentration curves in experiments have similar characteristics. Horizontal margins mark M_v. The signal curve leaves the box for the first time at the mixing time, when viewed from the right side.

5.2 Simulation of Mixing Processes

For investigation of the mixing time, it is necessary to bring the agitated system to a quasi-steady state, in which the main flow field is already developed. In order to augment the efficiency and minimize the calculation cost, a combination of several OpenFOAM® tools is used during the mixing time investigation.

6 Velocity Field

In multiple works, steady-state simulation is used for stirred vessels. It is very fast, and hence a good choice for preliminary studies. A comparison with time-averaged velocity measurement data shows good agreement with this kind of simulation concerning the mean flow pattern [16]. The temporally resolved observation of velocity fields, however, produces evidence of vortex structures that have a comparably long lifetime and move slowly through the flow domain. Those vortices play a crucial role in mixing and must not be ignored. In a steady-state simulation and temporally averaged fields, however, those structures cannot be detected, as they vanish in the mean field. For transient simulations, the OpenFOAM® solver pimpleDyMFoam is used, since it is an adequate solver for large time step, turbulent flows and accounts for the stirrer rotation by dynamic mesh motion.

A steady-state solution is obtained with the simpleFoam solver, including the use of a multi-reference-frame simulation. This is used as an initial condition for transient simulation. After 25 turns of the stirrer, the afore-mentioned vortex structures are developed and the velocity field is assumed to be quasi-stationary. Mixing time investigations can start from these conditions. Snapshots of the resulting vertical velocity fields from CFD and PIV are compared in Fig. 4.

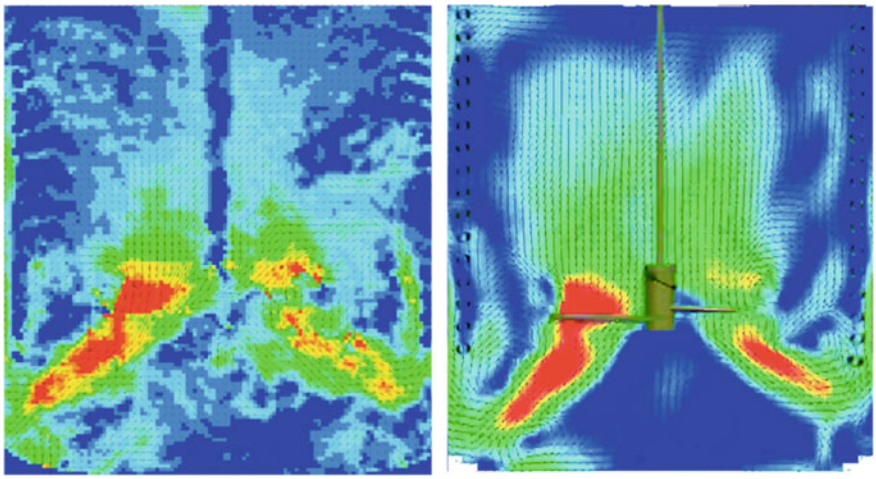


Fig. 4 Comparison of the vertical PIV-measurement (left) and transient CFD (right) results for a pitched blade impeller in a fully turbulent state

7 Tracing via Passive Scalar Transport on Existing Velocity Fields

A passive scalar transport equation is easy to implement on an existing solver in OpenFOAM®, as has been described in numerous tutorials on the web [19] or in [20].

If the mixing time is to be investigated under novel geometry or operating conditions, it is sensible to use the afore-mentioned quasi-stationary field as an initial condition and use the pimpleDyMFoam, extended by the scalar transport equation

$$\frac{\partial}{\partial t}\psi = \nabla \cdot (\Gamma \nabla \psi) - \nabla \cdot (\boldsymbol{u}\psi),$$

where ψ is the scalar of interest, $\boldsymbol{u}$ the velocity, and Γ the scalar diffusion coefficient.

If, however, velocity fields are present from other work, they can be reused for calculation of the mixing time. This method spares one the extremely time-consuming transient calculation of the velocity and pressure fields. The present velocity data indeed have to be adequately temporally resolved and have to reach over a time range that safely exceeds the mixing time. This approach is also especially useful when mixing times with different tracer injection spots, but the same velocity fields, have to be compared. In this case, only the scalar transport has to be calculated twice.

Mimicking the experimental trial, in the simulation, the tracer injection is implemented by setting ψ to the value 1 (setFields utility) on a domain that corresponds to the tracer volume and injection spot, while the residual ψ field remains at 0. Since

the tracer is injected quickly in experiments, the domain has the shape of a column. Two injection spots, centric and eccentric, are selected and compared.

8 Determination of Mixing Time at Probe Locations

The measurements of local concentrations during the mixing process were taken at six different points in the vessel (see Fig. 5). The OpenFOAM® utility "probeLocations" allows for analogous tracking of local tracer concentrations. For each of these locations, the local mixing time is determined in the manner described above. Of course, the local mixing times can differ due to dead zones. The total mixing time is assumed to be reached with the latest local mixing time. The exactness of such a method is questionable since local mixture goodness cannot be ensured throughout the entire domain. Possible dead zones, which extend the mixing time, may not be detected. Due to limitations in measurements, the available data are very sparse, and hence the described method is usually practiced.

9 Determination of Global Mixing Time

In CFD simulations, field data for the total volume of the domain are available. Local mixture goodness can be tracked at each computational cell. Following the 95% criterion, all cells that have a concentration higher than the lower margin of $M_{0.05}$ are extracted. The point that marks the mixing time is when the integral of these extracted cells reaches the total domain volume. This method is comparable to the first

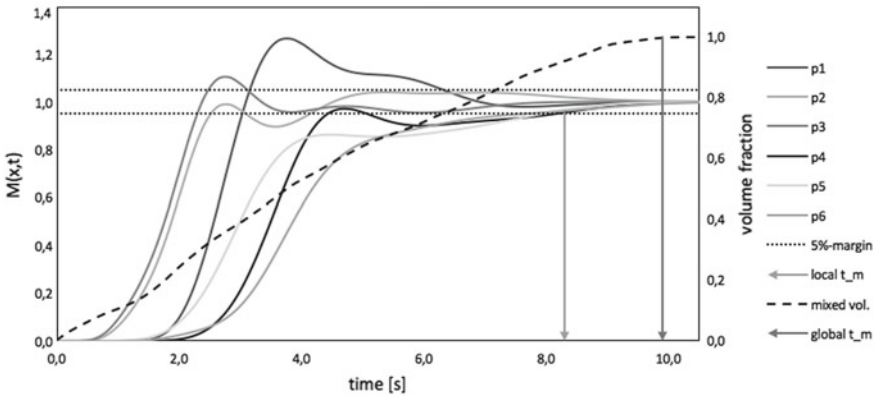


Fig. 5 Mixing time determination in a baffled vessel with a coil and Rushton stirrer at 200 rpm and a central tracing spot. Local and global determination strategies lead to different mixing time values

presented procedure if the probe locations cover the whole domain representatively. With a rising number of probes, the mixing time determined by the first method converges to the global mixing time. Both described methods are compared in Fig. 5. In the present case, the mixing time determined at six probe locations (see Fig. 3) is 8.3 s, but it is 9.8 s when determined globally.

10 Time Resolution for Scalar Transport

For the transient solution of velocity data, the Courant number Co [21] plays an essential role in regarding the time step size. It gives a good control mechanism for adaptive time discretization and is often indispensable for numerical stability. In the case of passive scalar transport, the time step has no influence on the stability, since only the actual velocity field is used in the equation. Rather, it is far more important that the temporal consistency, as it is defined in [21], be ensured, i.e., with decreasing step size, the results must converge to the analytical solution of the equation. An exemplary case would be a time resolution of $T/6$, where T denotes the stirrer revolution time, according to the finest resolution of $Co = 2$ to the stage of 95%. This is an acceptable compromise between accuracy and the additional amount of velocity and mesh data needed for further time resolution.

11 Validation of CFD Results

The CFD results for mixing time are validated with measurement data from [18]. For this study, different stirrer types, rotational frequencies and injection spots were chosen. The simulation and measurement results are plotted in Fig. 6. The results have linear dependency, although the measured mixing times are proportionally higher by a factor of 3. The reason for this discrepancy may be based on the simplicity of the simulation. The tracer concentration is simulated by a scalar transport equation, which does not account for possible interaction with or influence on the transport properties of the fluid. Further reasons can be based on certainty issues of the experimental work. The referenced work of Hirtsiefer [18] points out these problems. However, a correlation between CFD and LIF results is obvious.

12 Conclusions and Outlook

The meaning of CFD is still being underestimated in the chemical industry. The work contributed in this chapter is a demonstration of industrial application possibilities of the example of mixing time determination in an agitated vessel. Several OpenFOAM® and other free software utilities are used in the workflow. Transient quasi-stationary

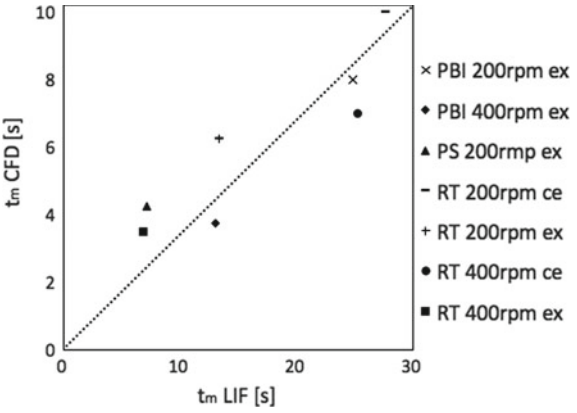


Fig. 6 Comparison of CFD and LIF results. Three different stirrer types are used: PBI—pitched blade impeller; PS—paddle stirrer; RT—Rushton turbine. Two rotational frequencies and two injection spots are chosen: centric and eccentric

velocity fields are used as initial conditions for a concentration-based mixing time determination method. The concentration is mimicked by the implementation of a positive scalar field, which is bounded to 1 at the highest concentration. A memory and computational time-saving method is used for this investigation. The CFD results coincide with LIF data proportionally, although the mixing time is predicted as three times lower than measured. This endorses the principals of the presented procedure and is valid for optimization research, since it reflects the tendencies very well, even under variation of independent parameters.

These results encourage application of the presented methods to industrial scale apparatuses for energy and product optimization, even beyond mixing time investigations. Recently, optimization of energy efficiency has attracted huge interest in industry. In the search for suitable instruments for its realization, CFD has the capacity to play a central role. In this context, OpenFOAM® poses a powerful and economical solution.

Acknowledgements The authors thank all those involved in the organization of OFW11 and all the contributors who have enriched this event. This project upon which this chapter is based is funded by the Federal Ministry of Education and Research under the project number: 03FH020PX4. Responsibility for the contents of this publication rests with the authors.

References

1. OpenCFD Ltd (2004–2016) OpenFOAM®. http://openfoam.com. Accessed 30 Sep 2016
2. Jasak H (2009), OpenFOAM®: Open source CFD in research and industry. Inter J Nav Archit Oc Engng 1:89–94. https://doi.org/10.2478/ijnaoe-2013-0011
3. Zlokarnik M (1999), Ruehrtechnik: Theorie und Praxis. Springer, Berlin Heidelberg
4. Kraume M (2003), Mischen und Ruehren. WILEY-VCH Weinheim
5. Ekato Holding GmbH (2012), Ekato. The Book. Ekato, Freiburg
6. Oldshue JY (1983), Fluid Mixing Technology. Chemical engineering McGraw-Hill Pub. Co., New York

7. Paul EL, Atiemo-Obeng VA, Kresta SM (eds) (2004), Handbook of Industrial Mixing. WILEY-INTERSCIENCE, Hoboken, New Jersey
8. Paschedag AR (2004), CFD in der Verfahrenstechnik. WILEY-VCH, Weinheim
9. Tatterson GB (2003), Fluid mixing and gas dispersion in agitated tanks. Gary Tatterson
10. Lohse R (2008), Simplified Calculation of Mixing and Suspending Processes in Agitated Vessels, Chem. Eng. Technol. 31(12):1799–1805
11. Platzer B (1988), Modeling of the Local Destributions of Velocity Components and Turbulence Parameters in Agitated Vessels – Method and Results. Chem. Eng. Process 23:13–31
12. Jaehrling K, Wolinski S, Ulbricht M, Schultz HJ (2016), Vergleich von Tubebaffles und Rohrschlangen in Bezug auf thermische und mechanische Energieeintragung in Ruehrreaktoren. Chem. Ing. Tech. 88(9):1224. https://doi.org/10.1002/cite.201650211
13. Stefan A, Wuenscher P, Schultz HJ, Ulbricht M (2016), Untersuchung der Gasdichteverteilung in begasten Ruehrkesseln mit eintauchenden Rohrschlangen. Chem. Ing. Tech. 88(9):1385–1386. https://doi.org/10.1002/cite.201650341
14. Schultz HJ, Matzke M, Kessel C, Jaehrling K, Stefan A, Bliem V (2016), Experimente und gekoppelte CFD-Simulationen von Wärmeübergängen an Rohrschlangen in Fermentern und Rührreaktoren. Chem. Ing. Tech. 88(9):1380. https://doi.org/10.1002/cite.201650216
15. Bliem V, Schultz HJ (2014), Investigation of Horizontal Velocity Fields in Stirred Vessels with Helical Coils by PIV. International Journal of Chemical Engineering. https://doi.org/10.1155/2014/763473
16. Bliem V (2016), Untersuchung des Einflusses der Stroemungsverhaeltnisse auf den Waermeuebergang in Ruehrreaktoren mit Rohrschlangeneinbauten mittels PIV/LIF. Dissertation, University of Duisburg-Essen
17. Stefan A, Hindges S, Schultz HJ (2016), Simulation of the Heat Exchange in Vessels with Helical Coils and the Influence of Stirrer Position. Chem. Ing. Tech. 89(4):470-474. https://doi.org/10.1002/cite.201600146
18. Hirtsiefer M (2015), Investigation of Heat Transfer and Mixing Characteristics of Helical Coils in Stirred Tank Reactors by Laser-induced Fluorescence. Master Thesis, University of Applied Sciences Niederrhein
19. 11th OpenFOAM® Workshop (2016) Training Courses. openfoamworkshop.com. Accessed 30 Sep 2016
20. Maric T, Hoepken J, Mooney K (2014), The OpenFOAM® Technology Primer. Sourceflux
21. Ferzinger JH, Peric M (2002), Computational Methods for Fluid Dynamics. Springer, Berlin Heidelberg

Wind Turbine Diffuser Aerodynamic Study with OpenFOAM®

Félix Sorribes-Palmer, Antonio Figueroa-González, Ángel Sanz-Andrés and Santiago Pindado

Abstract The aim of this work is to analyze the influence of the pressure losses of a Diffuser-Augmented Wind Turbine (DAWT) on the extractable power. Multielement diffuser geometries, generated with Salome and meshed with snappyHexMesh, are studied numerically with OpenFOAM® to find a configuration of maximum area expansion (reducing flow detachment), for different pressure losses at the actuator disk. Different geometries are studied with a $k-\varepsilon$ turbulence model. The influence of the vanes inside the diffuser has also been analyzed. The results of the present work show the importance of a careful design of the diffuser entrance.

1 Introduction

The extractable energy of a horizontal axis turbine rotor of fixed size can be increased by installing it at the entrance of a diffuser. The flow around flange diffusers has been studied experimentally and numerically by many researchers [6, 10, 12]. By recovering exhaust kinetic energy, the diffuser produces a greatly reduced pressure behind the turbine compared to the one behind a bare turbine. This effect increases the mass flow rate through the DAWT, with at least as much pressure change as across a conventional turbine [4]. The overall effect is to increase the generated power for a given rotor diameter. The increase of the mass flow rate in the diffuser is influenced by four main factors [11]:

- The diffuser area ratio A_{ex}/A_c.
- The flow separation downstream in the diffuser.
- The base pressure reduction at the diffuser exit caused by the obstruction flow.
- Viscous losses.

According to Blevins' Handboook [9], the flow separation in a diffuser, also called the diffuser stall, depends on diffuser inlet and outlet conditions, Reynolds number,

F. Sorribes-Palmer (✉) · A. Figueroa-González · Á. Sanz-Andrés · S. Pindado
Instituto Universitario de Microgravedad Ignacio Da Riva, Universidad Politécnica de Madrid, Madrid, Spain
e-mail: felix.sorribes@upm.es

J. M. Nóbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4_37

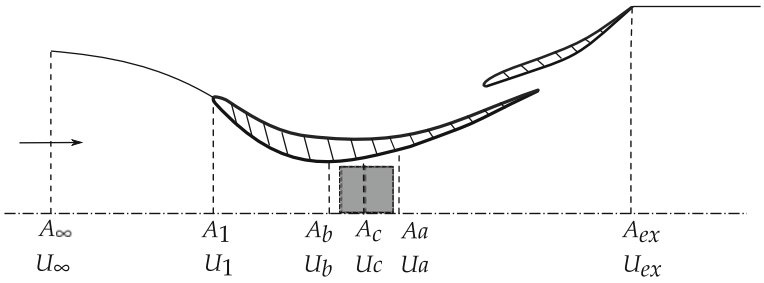


Fig. 1 Sketch of a general DAWT duct. The areas and flow velocities at the characteristic section of the duct are indicated

Mach number, and diffuser geometry. In the aforementioned handbook, diffuser stall is studied for different configurations: two-dimensional, conical, annular, straight-walled, and curved wall diffusers. It also includes an analysis of the influence of different diffuser geometry parameters on different stall regimes: first appreciable, large transitory, fully developed, hysteresis zone, and jet flow.

A general sketch of a DAWT duct is shown in Fig. 1. In this work, a two-slot diffuser configuration is analyzed as an improvement (see Figs. 2 and 3), bearing in mind that the tangential injection of air available from the external wind supplied to the boundary layer helps the main flow overcome the adverse pressure gradient and frictional losses in the wall region. Additionally, the effect of a vane installed in the duct was analyzed too (see Fig. 3). Vanes subdivide the diffuser into a series of diffusing passages, each of which will have divergence angles and area ratios much smaller than those of the vaneless diffuser [9].

The sampled sections are far field upstream A_∞, the inlet A_1, the section immediately upstream of the turbine A_b, the turbine section A_c, the section immediately downstream of the turbine A_a and the diffuser exit A_{ex}.

The present work is organized as follows: in Sect. 2, the main parameters of DAWT performance are presented. In Sect. 3, the CFD (Computational Fluid Dynamics) numerical set up of different configurations for finding the maximum extractable power are summarized, whereas the main results of the simulations are described in Sect. 4. Finally, in Sect. 5, the main conclusions of the numerical study are drawn.

2 Analytical Framework

The different cases are compared by analyzing the variation of pressure through the duct; from the energy conservation equation, for a steady and incompressible flow [2], the relation between the exit and freestream condition is given by

$$p_\infty + \frac{1}{2}\rho U_\infty^2 = p_{ex} + \frac{1}{2}\rho U_{ex}^2 + \Delta p_c + \Delta p_d \,, \tag{1}$$

where ρ is the density of the fluid, p_∞ is the pressure far upstream, p_{ex} is the pressure at the diffuser exit, Δp_c and Δp_d are the total pressure losses in the turbine and in the duct, respectively, and U_∞ and U_{ex} are the flow speed upstream and at the exit of the duct (see Fig. 1). The total pressure losses ($\Delta p = \Delta p_c + \Delta p_d$) can be modeled as

$$\Delta p_c = k_{pc} \frac{1}{2} \rho U_c^2 , \quad \Delta p_d = k_{pd} \frac{1}{2} \rho U_c^2 . \tag{2}$$

Besides, the pressure loss coefficient in the turbine can be referred to the flow speed upstream U_∞, with the cross section area ratio, $\lambda = A_{ex}/A_c = U_c/U_{ex}$:

$$K_{pc} = \frac{\Delta p_c}{\frac{1}{2}\rho U_c^2} \lambda^2 = k_{pc} \lambda^2 . \tag{3}$$

From Betz's limit, it can be deduced that the pressure loss at a turbine in relation to the extractable maximum energy is $K_{pc} \sim 2$ [8].

The pressure coefficient at the diffuser exit is

$$c_{pex} = \frac{p_{ex} - p_\infty}{\frac{1}{2}\rho U_\infty^2} , \tag{4}$$

where p_∞ and p_{ex} are, respectively, the static pressure of the flow upstream and at the exit.

Finally, the extracted power by the turbine can be estimated as

$$c_{Wc} = \frac{\Delta p_c U_c A_c}{\frac{1}{2}\rho A_c U_\infty^3} = K_{pc} \lambda \left(\frac{U_{ex}}{U_\infty} \right)^3 . \tag{5}$$

This parameter is used in the present work to compare the different DAWT configurations.

3 Numerical Setup

The open-source CFD software OpenFOAM® has been used to carry out the simulations. The performance analysis of several DAWT duct configurations has been conducted by employing a 2D model, placing a porous region in the throat to simulate an actuator disk.

The length of the studied DAWT duct configurations is about 6 m, with a cross section area ratio $\lambda = 3.7$. The computational domain dimensions are 60 m length and 24 m high. The geometries were developed with Salome, through use of a

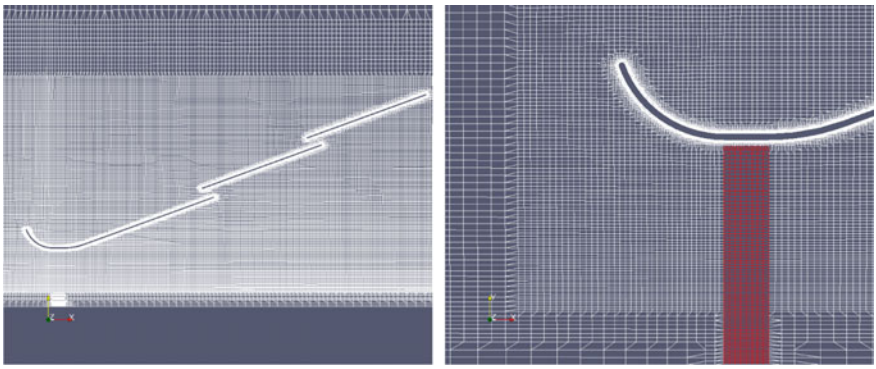


Fig. 2 Mesh details. The porous region is colored in red

python script, which allowed for the creation of parametric geometries, while the meshes were generated with snappyHexMesh. The procedure followed was to generate the .stl files first, with the mesh subsequently being generated employing surfaceFeatureExtract, snappyHexMesh and extrudeMesh to create the final mesh.

In Fig. 2, one diffuser geometry, together with the detail of the mesh in the porouse region, is shown. The final mesh is obtained from the castellation of the .stl geometries, as indicated in the aforementioned figure. Then, the porosity region was assigned with topoSet using the Darcy–Forchheimer formulation. Only inertial terms, F, were considered to take into account the effect of the porosity in this region. This term was included in the momentum equation as follows [5]:

$$\frac{\partial}{\partial t}(\rho u_i) + u_j \frac{\partial}{\partial x_j}(\rho u_i) = -\frac{\partial p}{\partial x_i} + \mu \frac{\partial \tau_{ij}}{\partial x_j} - \left(\mu D + \frac{1}{2}\rho u_{jj} F\right) u_i \,. \tag{6}$$

Each configuration has been simulated for several values of F. The turbine has been modeled applying the widely known actuator disk theory, placing a thin porous region of 0.2 m in length, in the throat of the turbine. Some examples of the use of porous disks for simulating turbines can be found in [1, 7]. The aforementioned porous region simulates the behavior of the turbine blades: the pressure drops in it, with the obvious advantage of speeding up the numerical simulations. In return, the rotational flow component is lost.

The numerical simulation was performed on the steady RANS equations with a $k - \varepsilon$ turbulence model for computational time reasons, as the main goal of the work is to study the different configurations within a reasonable time. Further work has already been done in [3], in which other turbulence models with and without wall functions have been applied to analyze their influence on the overall performance of DAWT. As it was expected, more complete turbulence models for the analysis of detached flows agree better with the empirical and theoretical relations that describe the evolution of the pressure in DAWTs.

The generated meshes provide y^+ close to 12 in all studied cases. Although y^+ was smaller than 30, wall functions were used to model turbulence at the walls. The Reynolds number based on the length of the first diffuser segment reached up to $Re \sim 550{,}000$, taking a freestream velocity of $U_\infty = 4$ m/s. The turbulence intensity introduced at the inlet was $I_u = 0.03$, the turbulent kinetic energy being calculated as $k = \frac{3}{2}(I_u U_\infty)^{3/2}$, and the dissipation rate as $\varepsilon = \rho C_\mu k/\mu_t$, where $C_\mu = 0.09$ and $\mu_t = 1000 I\mu$. The selection of the turbulence intensity was made based on a previous study in which the diffuser was located in the ground in a fully developed flow using equation $I_u = 0.16 \cdot Re_d^{-1/8}$.

The equations are numerically solved by means of the SIMPLE algorithm, the cell-based solver with the least squares being used. A bounded Gauss linear upwind difference scheme was used for spatial discretization. The solver used was porous-SimpleFoam with moderate underrelaxation factors.

A velocity boundary was used at the inlet and a static pressure boundary was used at the outlet. The top wall was defined as a no-slip boundary, and the back and front as empty so as to simulate a 2D domain. In addition, a symmetry plane boundary

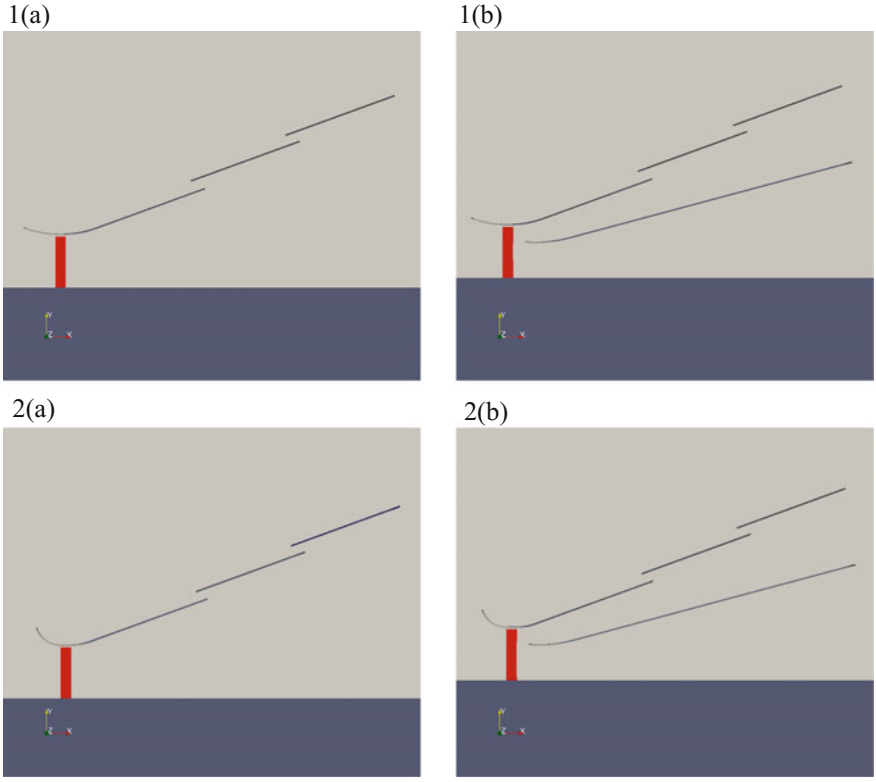


Fig. 3 Diffuser-augmented wind turbine configurations

condition simulating the other half of the turbine is used in the plane $y = 0$. The residuals were monitored using PyFoamPlotRunner, being the convergence criteria to keep the residual values under 10^{-4}. The DAWT duct configurations analyzed are shown in Fig. 3. The red zone represents the porous region simulating the turbine. From configuration (1a) to (1b), an interior vane was introduced to reduce detachment on the diffuser; from (1a) to (2a), the inlet was modified to avoid detachment at the leading edge of the diffuser; finally, between (2a) and (2b), the aforementioned vane has was introduced again.

4 Results

Flow and pressure fields distribution are shown in Figs. 4 and 5 for the different configurations analyzed. In Fig. 5, it can be observed how the vanes help to generate a more uniform and negative pressure profile at the exit, this result being even better observed in Fig. 6. The suction at the exit induces higher speed and more mass flow rate through the turbine section.

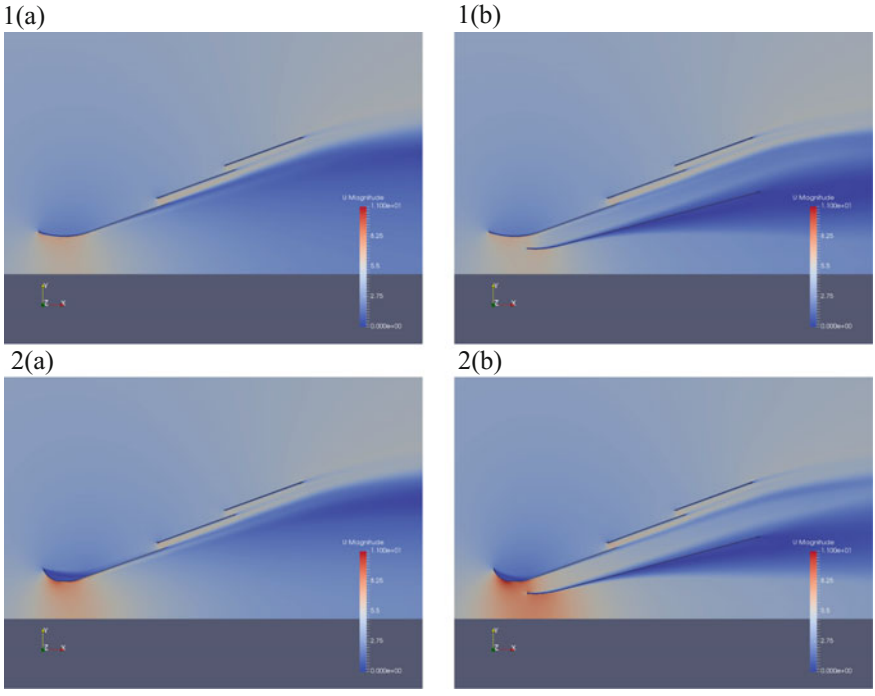


Fig. 4 Velocity fields at the different DAWT configurations for the optimum K_{pc} for each configuration

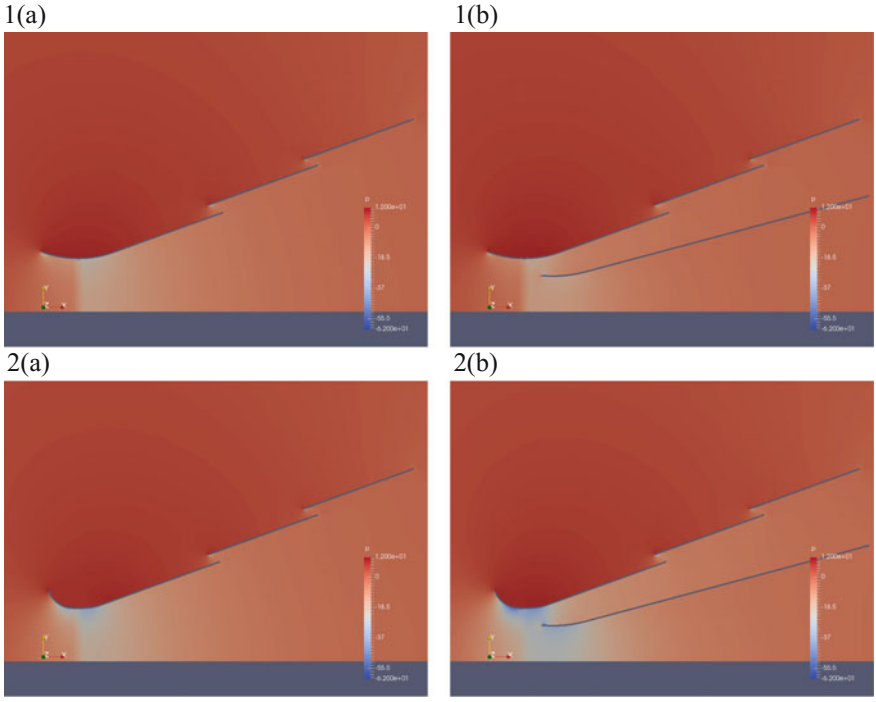


Fig. 5 Pressure fields at the different DAWT configurations for the optimum K_{pc} for each configuration

The profiles of velocity and pressure at the different sections of the DAWT duct configurations are shown in Fig. 6. These graphs indicate that the velocity profile at the first section of the duct is smoother in the configurations with the modified entrance. The flow acceleration in the gap between the porosity region and the diffuser, which helps to maintain the attached boundary layer, can also be observed. At the exit of the ducts, the velocity profile highlights the detached region. While adding a vane barely affects the pressure coefficient at the exit of the DAWT, it is significant that this coefficient decreases when the entrance of the diffuser is modified.

Comparing configurations (1a) and (1b), the effect of the vane implies a reduction of the energy extracted. However, once the entrance was improved [configurations (2a) and (2b)], the vane generated a more negative pressure at the exit, which induces higher speeds. Looking at Eq. 5, for the same pressure drop at the turbine K_{pc}, these higher speeds at the exit increase the energy extracted.

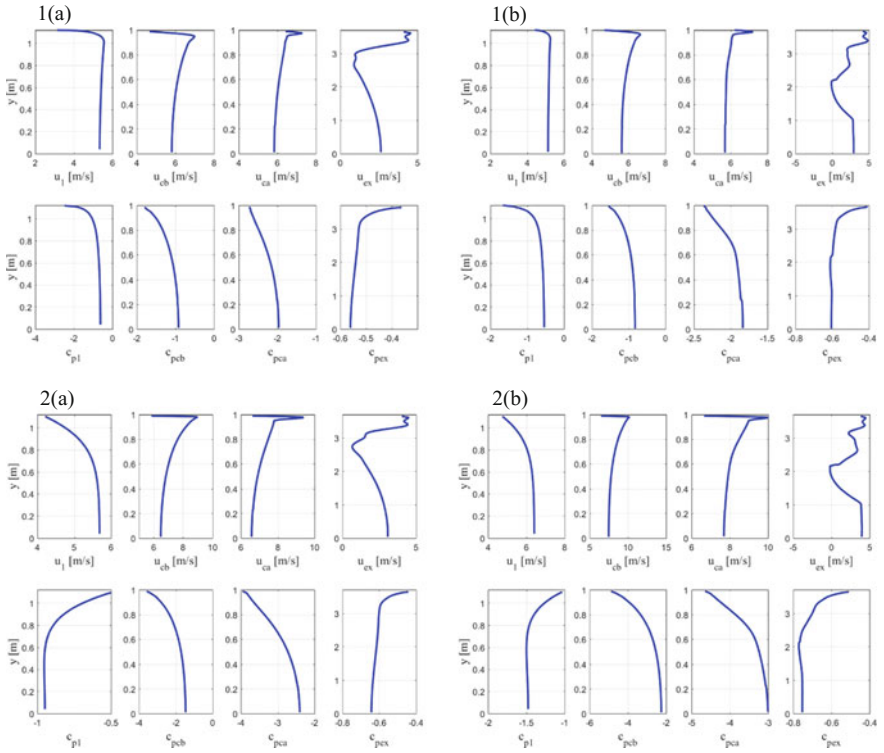


Fig. 6 Velocity and pressure coefficient profiles in the sections A_1, A_{cb}, A_{ca} and A_{ex} for the optimum K_{pc} for each configuration

The curves of the extracted power coefficient and pressure coefficient are shown in Fig. 7. The pressure coefficient has been estimated as an average of the pressure coefficient profile at the exit of the diffuser.

In Table 1, the values of the maximum efficiency in relation to the different DAWT analyzed are summarized. Although the splitter vane helps to reattach the boundary layer at the diffuser, as the flow detaches from the vane, the pressure losses does not compensate, because the configurations with the vane do not improve the performance within the entire simulated range of K_{pc}. A possible solution to this effect could be to split the vane or make it shorter to avoid flow detachment.

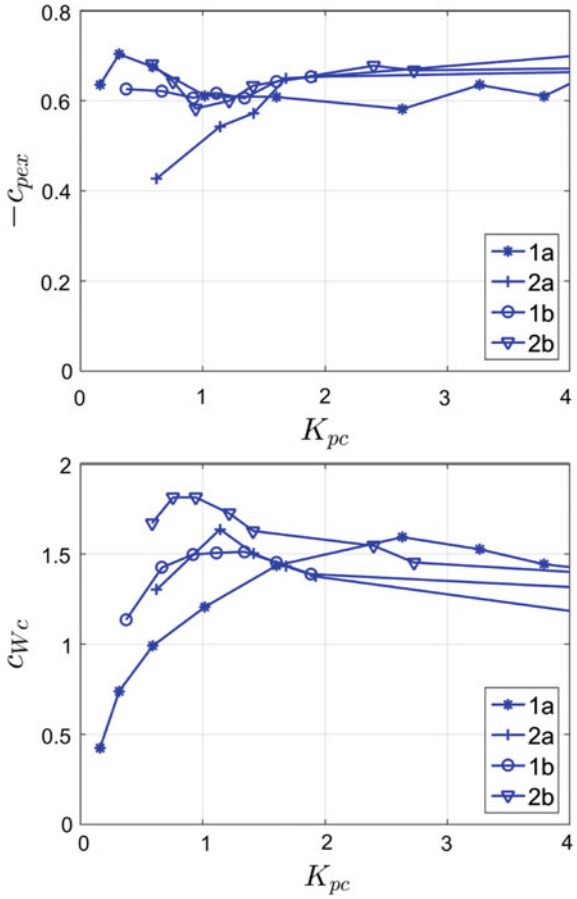


Fig. 7 Pressure coefficient at the turbine exit and specific power extracted

Table 1 Pressure loss coefficient K_{pc} at the for maximum extracted power coefficient C_{Wcmax} for the diffuser configurations

Configuration	N airfoils	K_{pcmax}	C_{Wcmax}
1a	3	2.6	1.6
1b	4	1.3	1.5
2a	3	1.1	1.6
2b	4	0.6	1.8

5 Conclusions

Four different configurations of DAWT ducts have been compared by obtaining the point of maximum extracted energy. The simulations show the importance of a well-designed entrance to the duct, in order to reduce pressure losses due to flow detachment. In addition, when the efficiency entrance increases, the optimum value of the extractable power is obtained for lower values of the coefficient K_{pc}, a result that result agrees with the work of [8]. In comparison with configuration (1a), the flow is detached in (1b) in a part of the vane, causing a reduction of the extractable power. However, when the entrance of the duct is enhanced, installation of a vane increases the extractable energy. Accordingly, the configuration (2b) has the best performance. The obtained value of c_{Wc} exceeds the Betz limit for all configurations.

Acknowledgements The authors thank the OFW11 organizers and to all the contributors who enriched this event. We also want to thank OpenFoamWiki, CFDsupport, Chalmers University, and ICE Stroemungsforschung for their contribution to OpenFoam® documentation.

References

1. G. Crasto and A. R. Gravdahl. CFD wake modeling using a porous disc. *European Wind Energy Conference and Exhibition. vol. 6, pp. 3028–3037*, 2008.
2. Dietrich Küchemann and Johanna Weber. *Aerodynamics of Propulsion*. McGraw-Hill, 1953.
3. Felix Sorribes-Palmer, Angel Sanz Andres, Antonio Figueroa, Lorenzo Donisi, Sebastian Franchini, and Mikel Ogueta. Aerodynamic Design of a Wind Turbine Diffuser with Openfoam®. In *7th European-African Conference on Wind Engineering*, Liege, 2017.
4. B.L. Gilbert, R.a. Oman, and K.M. Foreman. Fluid dynamics of diffuser-augmented wind turbines. *Journal of Energy*, 2(6): 368–374, 1978. ISSN 0146-0412. https://doi.org/10.2514/3.47988.
5. Haukur Elvar Hafsteinsson. Porous Media in OpenFOAM®. *Chalmers University of Technology. Thecnical Report*, 2009.
6. Ken Ichi Abe and Yuji Ohya. An investigation of flow fields around flanged diffusers using CFD. *Journal of Wind Engineering and Industrial Aerodynamics*, 92(3–4): 315–330, 2004. ISSN 01676105. https://doi.org/10.1016/j.jweia.2003.12.003.
7. W. X. M. Koh and E. Y. K. Ng. A CFD study on the performance of a tidal turbine under various flow and blockage conditions. *Renewable Energy. vol. 107, pp. 124–137*, 2017.
8. C J Lawn. Optimization of the power output from ducted. In *Proceedings of the Institution of Mechanical Engineers–Part A– Power & Energy*, volume 217, pages 107–118, 2003.
9. Robert D. Blevins. *Applied Fluid Dynamics Handbook*. New York, 1984. ISBN 9780442212964.
10. Rym Chaker, Mouldi Kardous, Fethi Aloui, and Sassi Ben Nasrallah. Open angle effects on the aerodynamic performances of a flanged Diffuser Augmented Wind Turbine (DAWT). 2014.

11. M. Shives and C. Crawford. Developing an empirical model for ducted tidal turbine performance using numerical simulation results. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 226(1): 112–125, 2012. ISSN 0957-6509. https://doi.org/10.1177/0957650911417958.
12. Yuji Ohya, Takashi Karasudani, Akira Sakurai, Ken Ichi Abe, and Masahiro Inoue. Development of a shrouded wind turbine with a flanged diffuser. *Journal of Wind Engineering and Industrial Aerodynamics*, 96(5): 524–539, 2008. ISSN 01676105. https://doi.org/10.1016/j.jweia.2008.01.006.

Index

J. M. Nbrega and H. Jasak (eds.), *OpenFOAM®*,
https://doi.org/10.1007/978-3-319-60846-4