Flow Turbulence Combust (2018) 100:19–38
DOI 10.1007/s10494-017-9827-9

An Adaptive Cartesian Mesh Based Method to Simulate Turbulent Flows of Multiple Rotating Surfaces

Alexej Pogorelov[1] · **Lennart Schneiders**[1] · **Matthias Meinke**[1,2] · **Wolfgang Schröder**[1,2]

Received: 8 December 2016 / Accepted: 13 June 2017 / Published online: 29 June 2017

Abstract An efficient Cartesian cut-cell/level-set method based on a multiple grid approach to simulate turbulent turbomachinery flows is presented. The finite-volume approach in an unstructured hierarchical Cartesian setup with a sharp representation of the complex moving boundaries embedded into the computational domain, which are described by multiple level-sets, ensures a strict conservation of mass, momentum, and energy. Furthermore, an efficient kinematic motion level-set interface method for the rotation of embedded boundaries described by multiple level-set fields on a computational domain distributed over several processors is introduced. This method allows the simulation of multiple boundaries rotating relatively to each other in a fixed frame of reference. To demonstrate the efficiency of the numerical method and the quality of the computed findings the generic test problem of a rotating cylinder surrounded by a stationary hull and the flow over a ducted rotating axial fan with a stationary turbulence generating grid at the inflow are simulated. The computational results of the axial fan show a good agreement with the experimental data.

Keywords Turbulent flow · Large-eddy simulation · Cartesian method · Level set · Rotating boundary · Axial fan

1 Introduction

Flows over multiple relatively rotating geometries are relevant in a wide range of turbomachinery applications. For boundary-fitted meshes, the computation of a rotating and a stationary object in a single fixed grid system is difficult due to the large boundary

✉ Alexej Pogorelov
a.pogorelov@aia.rwth-aachen.de

[1] Institute of Aerodynamics, RWTH Aachen University, Wüllnerstr. 5a, 52062 Aachen, Germany

[2] Forschungszentrum Jülich JARA-High-Performance Computing, Jülich 52425, Germany

displacements. Different approaches using boundary-fitted meshes exist to simulate rotating boundaries [1], i.e, the single rotating frame (SRF) approach, where the stator is constrained to be rotationally symmetric, the multiple reference frame (MRF) approach, which is limited to steady-state simulations, and the sliding mesh (SM) approach, which is the only approach to obtain a time-accurate solution of the Navier-Stokes equations for complex geometries. However, the sliding mesh technique suffers from non-conservativity at the interface between rotor and stator when arbitrary mesh movement in considered. Conservativity is only directly obtained by posing restrictions on the grid at the interface, i.e., the azimuthal spacial step is related to multiples of the relative rotational speed between rotor and stator [2].

To simulate multiple relatively moving boundaries within one single grid system, boundary-fitted methods either require a mesh deformation, which can only be used for small boundary displacements and reduces the quality of the mesh, or a remeshing of the near-boundary grid, which is very difficult and costly for structured meshes. Although it can be efficiently done for unstructured meshes using tetrahedral cells, which, however, possess a higher numerical diffusion and therefore have a lower numerical accuracy compared to Cartesian cells [3]. The embedded-boundary approach using unstructured Cartesian meshes allows one to overcome all of the aforementioned problems and to simulate multiple relatively moving or rotating boundaries within one fixed grid system. There is no need to remesh the near boundary mesh or to use interfaces which either suffer from non-conservativity or stipulate restrictions on the geometry or the computational mesh. This alternative approach is very efficient and maintains the high mesh quality.

Approaches based on non-boundary-fitted meshes, also known as immersed boundary methods are subdivided by Mittal and Iaccarino [4] into two categories based on the formulation of the boundary conditions on the walls that are not aligned with the mesh. The first category contains continuous approaches such as the immersed boundary method introduced by Peskin [5, 6] or Goldstein et al. [7]. In those methods, the no-slip wall boundary condition is applied by adding forcing terms to the governing equations. The disadvantage of the continuous approaches is that the forcing term which is used to satisfy the boundary condition smears out the geometry. Therefore, the surface of the configuration is not represented by a sharp interface [8], which makes those methods non-conservative. The methods of the second category are considered discrete approaches which can be subdivided into indirect and direct approaches. The indirect methods are similar to the continuous category in the sense that a forcing term is used. However, unlike in the first category the forcing is applied to the discrete system of equations. The methods of LeVeque and Li [9] and of Verzicco et al. [10] and Mohd-Yusof [11] belong to this category. In the direct approaches of the second category, which are often referred to as Cartesian cut-cell methods [12–15] no forcing terms are added. The cells that define the surface of the geometry are cut and only the fluid fraction of the original cell volume is retained, which ensures a strict local and global conservation of mass, momentum, and energy, which is an essential property for the suppression of unphysical disturbances in moving boundary problems [16] and to obtain accurate and stable results for large displacements of the embedded boundaries [17]. This makes the Cartesian cut-cell method a very promising approach.

Different ways to describe and track moving embedded boundaries exist. A popular approach is the use of a triangular representation of the geometry in STereoLithography (STL) format which can be easily exported from CAD data [18]. A more efficient and more robust method, however, is based on the level-set formulation where an implicit surface representation by signed-distance functions is used [19]. For simple geometries analytical level-set functions can be used which do not require any propagation of the level-set function. For complex geometries the level-set function can be initialized from the STL geometry which, however, is very expensive [19]. Therefore, for moving boundaries, this should only be done to initialize the signed-distance function at the initial time-step. To evolve a level-set function in space, a time-accurate solution of a transport equation using reinitialization to improve the shape conservation can be considered [20, 21]. However, even when reinitialization is used, the geometry is never fully conserved such that the error is accumulated and the difference to the initial geometry increases over time. Therefore, a semi-Lagrangian level-set solver for translational boundary displacements was introduced in [19], where the error, which is defined by the interpolation error, is reduced and remains constant over time. Since this method fails for rotational displacements, it is extended to rotating embedded boundaries in this study. That is, an efficient approach for the rotation of the level-set function, which is initialized from CAD data, in a highly parallelized environment is introduced.

The Cartesian cut-cell method for stationary geometries has already been successfully used by Pogorelov et al. [22–24] to analyze the flow field in a ducted axial fan configuration using the single rotating frame approach, which was possible due to the rotational symmetry of the outer casing wall. The current axial fan configuration, however, contains a stationary turbulence generating grid upstream of the rotating fan. To be able to accurately capture the interaction of the turbulence generating grid with the fan blades, the moving boundary formulation of the Cartesian cut-cell method from [25] is combined with the multiple level-set approach from [19] and extended to rotating embedded boundaries.

The purpose of this study is to develop an efficient parallelized numerical methodology based on the cut-cell/level-set approach on Cartesian hierarchical meshes which has the capability to predict the unsteady turbulent flow field in three-dimensional turbomachinery applications with complex relative rotating boundaries using high-fidelity turbulence modeling such as large-eddy simulation (LES).

As of today only a few authors performed LES using the immersed boundary approach, to simulate three-dimensional turbulent turbomachinery flows. You et al. [26–29] successfully applied the immersed boundary method with discrete forcing at the wall boundaries to perform LES for the investigation of the tip-leakage flow of a linear cascade. They obtained a good agreement of their numerical results with experimental data in terms of velocity, Reynolds stresses, and energy spectra. However, their geometry was simple and stationary. Tyagi and Acharya [30] used a continuous immersed boundary approach to solve an unsteady stator-rotor interaction problem for an incompressible fluid on a rather small computational mesh with approx. 6×10^5 cells and a pure translational relative motion between rotor and stator. Furthermore, they used a simple two-dimensional geometry and due to the small spanwise extension of the mesh of 10 percent blade chord and periodic boundary conditions in the spanwise direction no three-dimensional flow phenomena were resolved. Posa et al. [31] performed large-eddy simulations of the flow field in mixed-flow pumps with moving geometries on a structured cylindrical mesh with approx. 28×10^6 grid points. They used a discrete forcing approach proposed by Fadlun et al. [32], which is based on interpolation near the boundaries and also suffers from non-conservativity.

The present paper introduces an accurate and efficient approach based on the conservative cut-cell method for large scale simulations of rotor-stator interactions in turbulent compressible turbomachinery flows with realistic rotating geometries. The application of the method to the rotating ducted axial fan configuration and the good agreement of the numerical results with the experimental data proves the quality of the numerical approach.

Note, however, the main focus of this study is on the numerical methodology. The axial fan test case only represents an application that demonstrates the applicability of this approach to besides generic also industrially relevant problems.

The paper is structured as follows. First, the governing equations and the numerical method are described. Second, the level-set approach for rotating surfaces is introduced and applied to a generic rotor-stator test case. Then, it is shown that the method can be used to determine the flow over a rotating ducted axial fan before conclusions are drawn.

2 Mathematical Model

2.1 Governing equations

We consider a viscous and compressible fluid flow in the three-dimensional domain Ω. The non-dimensional conservation equations in a moving control volume $V(t) \in \Omega$ bounded by the surface $A(t) = \partial V(t)$ are given in arbitrary Lagrangian-Eulerian (ALE) formulation [33]

$$\frac{d}{dt}\int_{V(t)} \mathbf{Q} \mathrm{d}V + \oint_{\partial V(t)} \overline{\mathbf{H}} \cdot \mathbf{n} \mathrm{d}A = \mathbf{0}, \tag{1}$$

where $\mathbf{Q} = [\rho, \rho\mathbf{u}, \rho E]^T$ is the vector of the conservative variables, ρ the density, $\mathbf{u}$ the velocity vector , and $E = e + \mathbf{u}^2/2$ the total specific energy containing the specific internal energy e. Furthermore, let $\mathbf{P}$ denote the vector of primitive variables $\mathbf{P} = [\rho, \mathbf{u}, p]^T$ and the quantity $\mathbf{n}$ the outward unit normal vector on dA.

The flux tensor $\overline{\mathbf{H}}$ can be decomposed into an inviscid $\overline{\mathbf{H}}^{inv}$ and a viscous part $\overline{\mathbf{H}}^{vis}$ defined by

$$\overline{\mathbf{H}}^{inv} + \overline{\mathbf{H}}^{vis} = \begin{pmatrix} \rho(\mathbf{u} - \mathbf{u}_{\partial \mathbf{V}}) \\ \rho\mathbf{u}(\mathbf{u} - \mathbf{u}_{\partial \mathbf{V}}) + p\overline{\mathbf{I}} \\ \rho E(\mathbf{u} - \mathbf{u}_{\partial \mathbf{V}}) + p\mathbf{u} \end{pmatrix} + \frac{1}{\mathrm{Re}_0} \begin{pmatrix} \mathbf{0} \\ \overline{\boldsymbol{\tau}} \\ \overline{\boldsymbol{\tau}}\mathbf{u} + \mathbf{q} \end{pmatrix}, \tag{2}$$

where $\mathbf{u}_{\partial \mathbf{V}}$ denotes the velocity of the control volume surface.

All variables are non-dimensionalized by fluid properties of the state at rest denoted by the subscript “0”. The Reynolds number is obtained by $\mathrm{Re}_0 = \rho_0 a_0 l_{ref}/\eta_0$ using the speed of sound $a_0 = \sqrt{\gamma p_0/\rho_0}$ with the ratio of specific heats $\gamma = 1.4$ and the characteristic length l_{ref}. The stress tensor $\overline{\boldsymbol{\tau}}$ for a Newtonian fluid with zero bulk viscosity is expressed by

$$\overline{\boldsymbol{\tau}} = \frac{2}{3}\eta\,(\nabla \cdot \mathbf{u})\,\overline{\mathbf{I}} - \eta\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right). \tag{3}$$

According to Fourier’s law, the conductive heat flux reads

$$\mathbf{q} = -\frac{\lambda}{\mathrm{Pr}\,(\gamma - 1)}\nabla T, \tag{4}$$

where ∇T is the gradient of the static temperature, $Pr = \eta_0 c_p/\lambda_0 = 0.72$ is the Prandtl number, and c_p is the specific heat capacity at constant pressure. For a constant Prandtl number, the thermal conductivity is $\lambda(T) = \eta(T)$ and the dynamic viscosity η is calculated using Sutherland’s Law [34]

$$\eta(T) = T^{3/2}\frac{1 + S}{T + S}, \tag{5}$$

with $S = 111K/293K$ for air at moderate temperatures. The fluid is assumed an ideal gas such that the equation of state reads

$$e = \frac{p}{\rho\,(\gamma - 1)}. \tag{6}$$

2.2 Boundary conditions

At fixed or moving walls $\Gamma(t)$, the no-slip boundary condition is imposed

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_\Gamma(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma(t), \tag{7}$$

and the walls are considered adiabatic

$$\frac{\partial T}{\partial n} = 0, \quad \mathbf{x} \in \Gamma(t). \tag{8}$$

According to [35], a Robin-type boundary condition for the pressure is derived from the momentum equation by projecting the pressure gradient normal to the surface and neglecting the viscous terms

$$\frac{\partial p}{\partial n} = -\left(\rho \frac{d\mathbf{u}}{dt} \cdot \mathbf{n}\right), \quad \mathbf{x} \in \Gamma(t), \tag{9}$$

where d/dt denotes the material derivative.

3 Numerical Method

The numerical method is based on a multiple grid approach and uses two hierarchical Cartesian grids, where one grid is used to solve the conservation equation (1) and a second grid is used to track the moving geometries which are described by multiple level-set fields [19]. The computational meshes are generated by an efficient parallel mesh generator [36] and are adapted during the solution procedure according to the location of the moving boundary or local flow phenomena. The mesh for the fluid domain and the mesh for the level-set field can differ due to different requirements for the resolution of the geometry and the local flow field. However, they share the coarsest refinement level to ensure identical subdomain boundaries after the decomposition across different processors such that the hierarchical nature of the data structure allows a quick link and an efficient data exchange between the meshes.

In the following, the numerical method for the fluid domain is discussed. The conservation equation (1) is approximated on a hierarchical Cartesian grid with local refinements and cut cells at the boundaries using a cell-centered finite-volume discretization. The resolution of the Cartesian grid acts as an implicit filter and decomposes the turbulent structures into a resolved part containing the dominant large eddies and the non-resolved sub-grid scale part. That is, in the LES formulation, the sub-grid scales are implicitly modeled based on the monotone integrated LES approach (MILES) [37], where appropriate discretization schemes are used to numerically dissipate the energy at the smallest scales. This approach has been successfully validated by computing various complex turbulent flow problems [38–41].

The cut-cell concept is used for a sharp representation of wall boundaries. That is, in each time step all cells intersecting the moving boundary Γ, which is described by the zero contour of the level-set function, are cut by computing their intersection points. Then, the cell is divided into a fluid part and a solid part using a piecewise linear approximation of the geometry in between the cut points, where the solid part is discarded. Based on the cutting points, the new volume V, surface areas A, and boundary-surface normals $\mathbf{n}$ of the fluid part of the cell can be computed. Furthermore, the new cell center is obtained by $\mathbf{x}_c = \frac{1}{V}\int_V \mathbf{x} d\tilde{V}$. For further details on the Cartesian cut-cell procedure for forced boundary motion, the reader is referred to [25].

The surface fluxes are approximated by an upwind-biased scheme, where the primitive variables at the cell surfaces are obtained by a second-order accurate monotonic upstream-centered scheme for conservation laws (MUSCL). The left and right values at each surface centroid $\mathbf{x}_s = \frac{1}{A}\int_A \mathbf{x} d\tilde{A}$, marked by the superscripts L and R, are extrapolated from the cell centers of the two facing cells

$$\mathbf{P}_s^{L/R} = \mathbf{P}_c^{L/R} + \nabla\mathbf{P} \mid_c^{L/R} \cdot(\mathbf{x}_s - \mathbf{x}_c^{L/R}), \tag{10}$$

where the cell-centered gradients $\nabla\mathbf{P} \mid_c$ of each primitive variable $\xi \in \mathbf{P}$ in cell c are computed by a weighted least-squares reconstruction [42]

$$\nabla\xi \mid_c = \sum_{l\in N_c} \mathbf{C}_l(\xi_l - \xi_c) \tag{11}$$

using all direct neighbors of the cell denoted by N_c. The reconstruction constants are obtained by solving a linear equation system

$$\overline{\mathbf{A}}\mathbf{C}_l = W_l{}^2\mathbf{dx}_l, \tag{12}$$

with the distance vector $\mathbf{dx}_l = \mathbf{x}_l - \mathbf{x}_c$ and the tensor $\overline{\mathbf{A}} = \sum_{l\in N_c} W_l{}^2\mathbf{dx}_l\mathbf{dx}_l$, where $W_l{}^2$ is the squared reconstruction weight of the neighboring cell l. Details on the weighting and the solution of the linear equation system can be found in [25].

The inviscid flux vector $\mathbf{F}_i^{inv} = \overline{\mathbf{H}}^{inv} \cdot \mathbf{e}_i$ is computed by a modified version of Liou and Steffen's advection upstream splitting method (AUSM) [43] in a low dissipation version suitable for LES proposed by Meinke et al. [39]

$$\mathbf{F}_i^{inv} = \frac{1}{2}[M_i^{LR}(\mathbf{f}^L + \mathbf{f}^R) + |M_i^{LR}|(\mathbf{f}^L - \mathbf{f}^R)] + \mathbf{p}_i, \tag{13}$$

where $i = 1, 2, 3$, $M_i^{LR} = \frac{1}{2}[(u_i/a)^L + (u_i/a)^R]$, and

$$\mathbf{f} = [\rho a, \rho a\mathbf{u}, \rho a(E + p/\rho)]^T. \tag{14}$$

The term for the pressure is obtained by

$$\mathbf{p}_i = \left[p^L\left(\frac{1}{2} + \chi(u_i/a)^L\right) + p^R\left(\frac{1}{2} - \chi(u_i/a)^R\right)\right]\begin{pmatrix}0\\ \mathbf{e}_i\\ 0\end{pmatrix}, \tag{15}$$

where the parameter χ determines the numerical dissipation of the scheme. It is set to 0.5 at surfaces between cells of different refinement levels and 1/96 everywhere else. For the viscous flux vector $\mathbf{F}_i^{vis} = \overline{\mathbf{H}}^{vis} \cdot \mathbf{e}_i$, a central-difference scheme is used, where the gradients at the cell-surface centroids are computed using the recentering approach proposed by Berger [44]. As shown in [16], the overall spatial accuracy of this approach is second order. Further details on the basic numerical method can be found in [16, 25, 45].

The temporal integration of the Navier-Stokes equations (1) is done by a new more efficient explicit second-order accurate 5-stage Runge-Kutta scheme proposed by Schneiders et al. [25], which is a modification of the scheme from Jameson and Mavriplis [46]

$$\begin{aligned}
(\mathbf{Q}V)^{(0)} &= (\mathbf{Q}V)^{(n)},\\
(\mathbf{Q}V)^{(1)} &= (\mathbf{Q}V)^{(0)} - \Delta t\,\mathbf{R}(t^n;\mathbf{Q}^{(0)}),\\
(\mathbf{Q}V)^{(k)} &= (\mathbf{Q}V)^{(0)} - \Delta t\left[(1-\alpha_{k-1})\,\mathbf{R}(t^n;\mathbf{Q}^{(0)}) + \alpha_{k-1}\,\mathbf{R}(t^{n+1};\mathbf{Q}^{(k-1)})\right], \quad k = 2,...,5,\\
(\mathbf{Q}V)^{(n+1)} &= (\mathbf{Q}V)^{(5)}.
\end{aligned} \tag{16}$$

The superscript n denotes the time level $t = n\Delta t$ and $\mathbf{R}(t;\mathbf{Q})$ is the finite-volume aproximation of the cell-surface flux integral at time level t, i.e,

$$\mathbf{R}(t;\mathbf{Q}) = \oint_{\partial V(t)} [\overline{\overline{\mathbf{H}}}^{inv}(\mathbf{Q}) + \overline{\overline{\mathbf{H}}}^{vis}(\mathbf{Q})] \cdot \mathbf{n} dA + O(h^2). \tag{17}$$

Note that unlike in the Jameson and Mavriplis [46] method the residual operator $\mathbf{R}$ has not to be recomputed in the intermediate Runge-Kutta stages. This makes the new method extremely attractive when problems with moving boundaries are considered, where in the standard formulation [46] the residual operator for each intermediate stage has to be reconstructed.

In this study, the coefficients $\boldsymbol{\alpha} = (\frac{1}{4}, \frac{1}{6}, \frac{3}{8}, \frac{1}{2}, 1)^T$ are used, which are optimized for stability. The time step Δt is computed via the Courant-Friedrichs-Lewy (CFL) stability constraint

$$\Delta t = \mathrm{CFL} \min_{i=1,2,3} \frac{h}{|u_i| + a}, \tag{18}$$

where h is the length of the smallest non-cut cell of the computational domain.

Using the aforementioned cut-cell approach arbitrarily small cells can be generated, which lead to numerical instabilities. Therefore, a flux redistribution method developed by Schneiders et al. [25], which ensures stability without reducing the time step to extremely small values, is applied.

4 Kinematic Motion Level-set Interface Method

As mentioned in Section 3, a second Cartesian mesh is used for the level-set field to track the moving boundaries. A multiple level-set approach with a semi-Lagrangian level-set solver for multiple translationally moving boundaries has been introduced in [19]. In the following, this approach is extended to rotating boundaries such that arbitrary turbomachinery flows can be computed. An efficient approach for the rotation of the embedded boundaries described by a level-set field on a computational domain distributed across different processors is introduced. The approach is explained for an example with a simple geometry, i.e., a rotating cylinder surrounded by an outer wall and subsequently, applied to the flow over a ducted axial fan with an upstream installed turbulence generating grid.

To represent an arbitrary moving embedded boundary $\Gamma(t)$ a signed-distance function $\phi(\mathbf{x}, t)$ defined as

$$\begin{aligned}
\Omega^f(t) &= \{\mathbf{x}|\phi(\mathbf{x},t) > 0\},\\
\Gamma(t) &= \{\mathbf{x}|\phi(\mathbf{x},t) = 0\},\\
\Omega^s(t) &= \{\mathbf{x}|\phi(\mathbf{x},t) \leq 0\}
\end{aligned} \tag{19}$$

with the property $|\nabla\phi|$=1 is used. The sign of $\phi(\mathbf{x}, t)$ defines wether a point $\mathbf{x} \in \Omega$ is located inside the solid $\Omega^s(t)$ or the fluid domain $\Omega^f(t)$ at a time t. The interface location is given by the zero contour of the level-set function $\Gamma(t) = \{\mathbf{x}|\phi(\mathbf{x}, t) = 0\}$ denoted by ϕ_0. The distance to each point $\mathbf{x} \in \Omega$ can be evaluated by

$$dist(\mathbf{x}, \Gamma(t)) = |\phi(\mathbf{x}, \mathbf{t})|. \tag{20}$$

The use of a signed-distance function allows an easy evaluation of the normal vector $\mathbf{n}(\mathbf{x}, t)$ pointing into the solid

$$\mathbf{n}(\mathbf{x}, t) = -\frac{\nabla\phi(\mathbf{x}, t)}{|\nabla\phi(\mathbf{x}, t)|}. \tag{21}$$

Further details on the computation of intersection information required for the computation of the cut cell geometry are discussed in [19].

There are different approaches for the evolution of a level-set function. For simple geometries analytical level-set functions can be used which allows a straightforward evaluation of the embedded boundaries at any time t for arbitrary translational and rotational motions. If the level-set function is initialized from CAD data, a time accurate solution of the level-set transport equation

$$\frac{\partial\phi}{\partial t} + \mathbf{f} \cdot \nabla\phi = 0, \tag{22}$$

with the velocity vector $\mathbf{f}(\mathbf{x}, t)$ can be determined using the reinitialization concept to improve the shape conservation [20, 21]. However, despite high-order discretization methods and constraint reinitialization approaches numerical errors are accumulated when (22) is solved such that the geometric conservation is not completely fulfilled, i.e., the geometry slightly changes in time [19]. Therefore, a semi-Lagrangian level-set solver was introduced in [19]. Using this approach the geometric error is just defined by the interpolation error and remains constant over time. However, this semi-Lagrangian method was developed for translational displacements only. Therefore, it is extended to rotational displacements in this study.

The overall methodology is discussed for a simple rotor-stator test case, i.e., a ducted rotating cylinder illustrated in Fig. 1, where the embedded boundary Γ_1 denotes the outer wall and Γ_2 the inner rotating cylinder. The length and the diameter of the inner cylinder are $L/D_o = 2/3$ and $D_i/D_o = 2/15$, where D_o denotes the diameter of the outer wall. The origin of the coordinate system coincides with the centerlines of the outer and inner cylinder which also define the x and y directions. Figure 2 shows the separate level-set fields of the stationary outer cylinder ϕ^1 (top left), the rotating inner cylinder ϕ^2 (top right) initialized

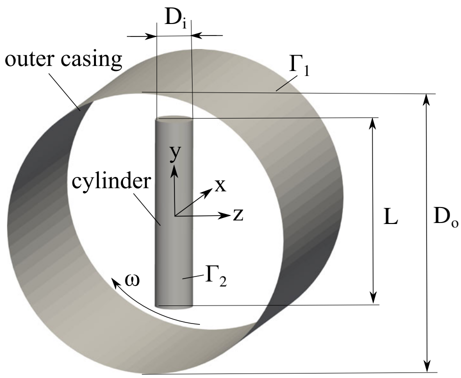

Fig. 1 Inner cylinder and outer cylinder wall given in STereoLithography (STL) format; Γ_1 and Γ_2 denote the boundaries of the outer wall and the inner cylinder

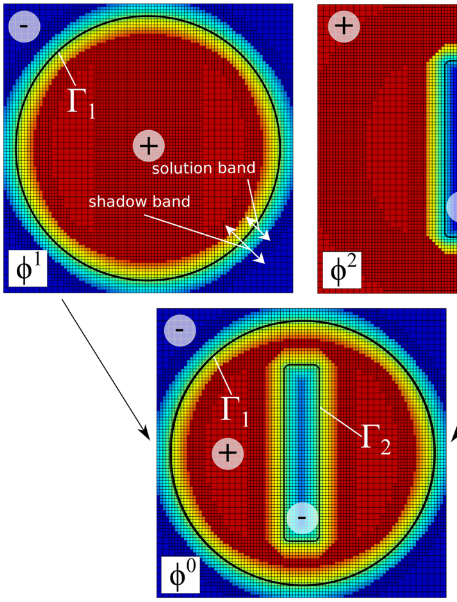

Fig. 2 Illustration of the $x = 0$, y-z plane, i.e., the cross section through the center of the cylinder; the level-set function ϕ^1 (*top left*) describes the embedded boundary Γ_1 initialized from the STL geometry of the outer wall; the level-set function ϕ^2 describes the embedded boundary Γ_2 (*top right*) initialized from the STL geometry of the inner cylinder; the level-set function ϕ^0 (*bottom*) is a combination of ϕ^1 and ϕ^2 and is used for the generation of cut cells; the *colors* correspond to the value of the respective level-set function; *black solid lines* denote the zero level-set contour ϕ_0

from STL data, and a combined level-set field ϕ^0 (bottom) according to [19]. The latter is used for the computation of the cut cells. For a sharper representation of complex geometries or tip gaps, the multi-cut cell technique introduced by Schneiders et al. [47] can be used, where ϕ^1 and ϕ^2 are used for the computation of the cut cells, instead of ϕ^0. This method allows complex intersections of a single Cartesian cell with multiple surfaces. The positive values of the level-set function ϕ^0 denote the fluid part and the negative values the solid part of the domain and black solid lines denote the zero level-set contour ϕ_0. Furthermore, Fig. 2 illustrates the Cartesian mesh used for the level-set field. The mesh is adaptively refined along the embedded boundaries Γ_1 and Γ_2, where the maximum refinement level, which defines the geometric accuracy, and the width of the refined band denoted by shadow band in Fig. 2 (top left), can be specified. Since the level-set information is only required close to the boundaries and to reduce the costs, the level-set field is defined in a thin band denoted as solution band Fig. 2 (top left) located within the shadow band. The rest of the domain is initialized by the maximum positive or negative level-set value. Both bands are defined based on the zero level-set contour ϕ_0. When one embedded boundary is propagated in time, the solution band moves within the shadow band. Once the distance between the outer boundaries of the shadow and solution band becomes smaller than $3h$, with h being the cell length inside the shadow band, the mesh is adapted to retain a fixed band of refined cells along the embedded boundaries.

In the present formulation, the level-set field ϕ^2 is to be propagated in time to rotate the embedded boundary Γ_2 about the x-axis. To prevent the accumulation of the geometric error in time, the level-set function is propagated by interpolating from the reference level-set field initialized from the STL geometry such that the geometric error is just defined by the interpolation error and remains constant over time. For any arbitrary prescribed motion of the embedded boundary Γ_2, the level-set function in the cell center $\mathbf{x}$ can be evaluated by

$$\widetilde{\phi}^2(\mathbf{x}, t_0 + t) = \phi^2(\mathbf{x}_{int}) \tag{23}$$

at every time step using trilinear interpolation according to [19]. For linear translational motion without acceleration $\mathbf{x}_{int} = \mathbf{x} - \mathbf{v}(t - t_0)$, where $\mathbf{v}$ is the translational velocity of the embedded boundary.

For small displacements $||\mathbf{x} - \mathbf{x}_{int}|| \leq 3h/2$ only the direct neighbors of a cell have to be searched to find the corresponding cell which contains the point $\mathbf{x}_{int}$. Then, $\widetilde{\phi}^2$ is evaluated by trilinear interpolation using the level-set values at the corner vertices ϕ_v^2, which are computed using the cell center values of the eight surrounding cells $\phi_v^2 = \sum_{k=1}^{8} \phi_k^2$. For parallel computations, one layer of halo and window cells is used to exchange data at the sub-domain interfaces, i.e., the cell center value and the eight values at the corner vertices are exchanged between the window and the halo cells, where window cells are internal cells at the sub-domain boundaries and halo cells are a copy of the according window cells on the neighboring domain.

Once the displacement becomes larger than $3h/2$, the corresponding cell containing $\mathbf{x}_{int}$ has to be searched in the entire domain. This is very expensive, especially for parallel computations where the domain is distributed on an arbitrary number of processors. To avoid this, the entire level-set field within the solution band is shifted by exactly one cell in the direction of motion

$$\phi^2_{shift}(\mathbf{x}) = \phi^2(\mathbf{x} - \mathbf{x}_{shift}) \tag{24}$$

each time the displacement becomes larger than half the cell length such that $||\mathbf{x} - \mathbf{x}_{int}|| \leq h/2$ at all times. Thus, the interpolation can be performed locally within the cell in which the level-set field has to be evaluated without the need to search the corresponding cell in the entire domain. Since the level-set field is shifted by exactly one cell, the level-set values can be directly copied from the neighboring cells and no interpolation is required for this step such that the accuracy of the reference level-set field ϕ^2 is kept. Therefore, the level-set function $\widetilde{\phi}^2$ is evaluated by

$$\widetilde{\phi}^2(\mathbf{x}, t_0 + t) = \phi^2_{shifted}(\mathbf{x}_{int}). \tag{25}$$

This approach works fine for translational motion. However, due to the Cartesian mesh, for rotational motion the reference level-set field ϕ^2 cannot be shifted in the direction of motion without a loss of accuracy to avoid the search of the corresponding cell containing $\mathbf{x}_{int}$, which for rotational motion without acceleration is defined as $\mathbf{x}_{int} = \overline{\mathbf{Q}}(\mathbf{x} - \mathbf{x}_{ref}) + \mathbf{x}_{ref}$. The quantity $\mathbf{x}_{ref}$ denotes the center of rotation and $\overline{\mathbf{Q}}$ denotes the rigid body rotation, which for a rotation at a constant angular velocity ω about the x-axis reads

$$\overline{\mathbf{Q}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\omega(t - t_0)) & sin(\omega(t - t_0)) \\ 0 & -sin(\omega(t - t_0)) & cos(\omega(t - t_0)) \end{pmatrix}. \tag{26}$$

Therefore, the reference level-set field initialized from the STL data remains unchanged at all times. To keep the accuracy of the reference level-set field, the level-set mesh is not adapted within the initial solution band. To rotate the embedded boundary the level-set function is evaluated according to Eq. 23 and the global cell ID of the cell containing $\mathbf{x}_{int}$ and its sub-domain ID are stored for each cell inside the solution band and are updated at each time step by checking whether the stored cell contains the point $\mathbf{x}_{int}$. This allows restarting the solution procedure at any instant in time using an arbitrary number of processors. At a time when the point $\mathbf{x}_{int}$ leaves the corresponding cell, all direct neighbors are searched to find the new corresponding cell and the stored global cell ID is updated. If $\mathbf{x}$ and $\mathbf{x}_{int}$ are located in the same sub-domain, no communication is required. However, if the new corresponding cell is a halo cell, the global cell ID of its corresponding window cell located on the neighbor sub-domain and the ID of the sub-domain are stored. In the next time step, the coordinates $\mathbf{x}_{int}$ and the global cell ID of the stored window cell are sent directly to the stored sub-domain. Then, it is checked whether the window cell still contains the point $\mathbf{x}_{int}$, otherwise its neighbors are searched. Subsequently, the level-set function ϕ^2 is interpolated based on values at the vertices of the identified corresponding cell and sent back to the sub-domain which requested this information. This approach is applied to each cell located inside the solution band and avoids a global search in all sub-domains which is very expensive. After the interpolation, the level-set fields ϕ^1 and ϕ^2 are combined to the level-set field ϕ^0, which is then exchanged with the computational mesh used to solve the conservation equations to compute the cut cells. Using this approach the inner cylinder can be rotated. It is exemplarily shown in Fig. 3 for three angles of rotation, i.e., 0°, 45°, and 90°.

To quantify the geometric error, three mesh resolutions inside the solution band are used, i.e., $h/D_o = 0.0114$, $h/D_o = 0.0228$, and $h/D_o = 0.0456$. For all cases, the zero level-set contours ϕ_0 are exported as triangular surface grids Γ_h^{LVS} at each time step. The zero level-set contour ϕ_0 of the finest mesh, i.e., $h/D_o = 0.0114$, at time step t_0, which is reconstructed from the level-set field initialized by the STL data, is denoted Γ_h^{ref} and used as reference. To compute the error, we define the distance $dist(\mathbf{x}_v, \Gamma_h^{ref})$ between a point $\mathbf{x}_v$ belonging to a surface Γ_h^{LVS}, $\mathbf{x}_v \in \Gamma_h^{LVS}$ with the superscript LVS meaning level set, and the reference surface Γ_h^{ref}

$$dist(\mathbf{x}_v, \Gamma_h^{ref}) = \min_{\mathbf{x}_q \in \Gamma_h^{ref}} ||\mathbf{x}_v - \mathbf{x}_q||_2, \tag{27}$$

where $||\cdot||_2$ denotes the Euclidian norm. Furthermore, we require $\mathbf{x}_v \in \Gamma_h^{LVS}$ to be a grid vertex, while $\mathbf{x}_q \in \Gamma_h^{ref}$ may be located on a triangle edge or surface. The mean distance

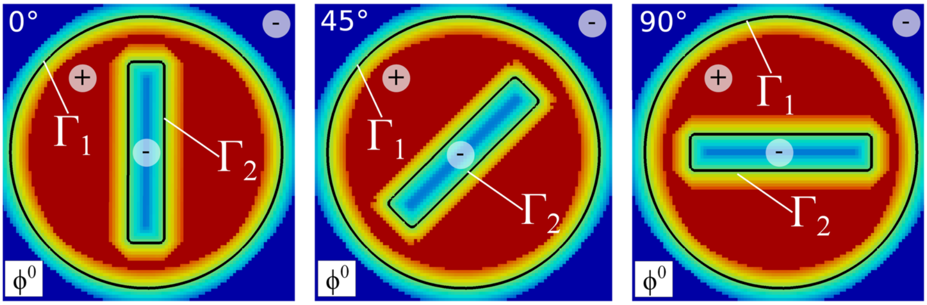

Fig. 3 Illustration of the $x = 0$, y-z plane, i.e., the cross section through at the center of the cylinder; the level-set field ϕ^0, which is a combination of the level-set field ϕ^1 and the propagated level-set field ϕ^2 for three angles of rotation, i.e., 0°, 45°, and 90°; the *colors* correspond to the value of the respective level-set function; *black solid lines* denote the zero level-set contour ϕ_0

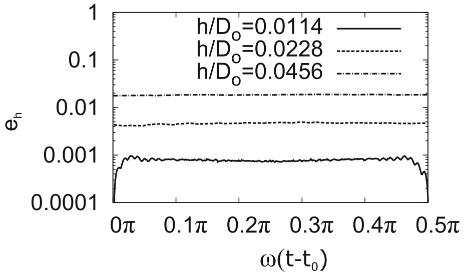

Fig. 4 L1-norm of the geometric error e_h over time for a rotation of the inner cylinder by 90° about the x-axis for three grid resolutions, i.e., $h/D_o = 0.0114$, $h/D_o = 0.0228$, and $h/D_o = 0.0456$

based on the L1-norm of the distance field from Γ_h^{LVS} to Γ_h^{ref} can be computed by

$$dist(\Gamma_h^{LVS}, \Gamma_h^{ref}) = \frac{1}{N_{\Gamma_h^{LVS}}} \sum_{\mathbf{x}_v \in \Gamma_h^{LVS}} dist(\mathbf{x}_v, \Gamma_h^{ref}), \tag{28}$$

where $N_{\Gamma_h^{LVS}}$ is the number of grid vertices $\mathbf{x}_v$ on Γ_h^{LVS}. Therefore, we define the geometric error e_h at each time step as

$$e_h = \frac{dist(\Gamma_h^{LVS}, \Gamma_h^{ref})}{D_o}, \tag{29}$$

which is depicted in Fig. 4 over a quarter rotation of the inner cylinder about the x-axis for three grid resolutions. The geometric error which is just defined by the interpolation error remains almost constant during the rotation of the inner cylinder. Since Γ_h^{LVS} of the finest mesh at t_0 is used as the reference surface grid Γ_h^{ref} and due to the Cartesian structure of the level-set mesh, the geometric error is zero at 0° and 90° for $h/D_o = 0.0114$. The interpolation error decreases, as the mesh is refined. To quantify the convergence rate of the geometric error, it is averaged over a quarter rotation

$$e_{avg} = \frac{2}{\pi} \int_0^{\pi/2} e_h d\theta. \tag{30}$$

Figure 5 evidences the second-order convergence of the mean geometric error.

Note that the method is not restricted to rotational motions at a constant angular velocity ω. It can be easily applied to arbitrary translational and rotational motions using $\mathbf{x}_{int} = \overline{\mathbf{Q}}(\mathbf{x} - \mathbf{x}_{ref}) + \mathbf{x}_{ref} - \mathbf{v}(t - t_0)$, with $\overline{\mathbf{Q}}$ from Eq. 26, where the angular velocity $\omega(T)$ and the translational velocity $\mathbf{v}(t)$ can vary in time.

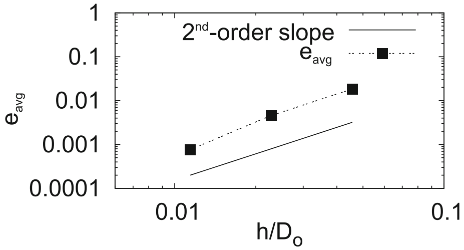

Fig. 5 L1-norm of the geometric error averaged over a quarter rotation e_{avg} of the inner cylinder about the x-axis for three grid resolutions, i.e., $h/D_o = 0.0114$, $h/D_o = 0.0228$, and $h/D_o = 0.0456$; the *solid line* indicates a second-order slope

5 Results

Having described the method to numerically simulate the rotating embedded boundaries for a generic case, we now turn to the discussion of an industrially relevant flow problem using the multiple grid Cartesian cut-cell/level-set method. The flow over a rotating ducted axial fan with a stationary turbulence generating grid at the inflow is computed. The flow configuration is a standardized fan test rig, which has been used for acoustic measurements by, e.g., Sturm and Carolus [48]. The axial fan is a low pressure rotor-only fan and has five twisted blades made up from standard NACA 0010-63 airfoil sections with a maximum camber of 4% at midchord. The chord length of the blades varies between 67.7 mm and 90.5 mm. The trailing edge thickness is 0.6 mm and the diameter of the outer casing 300 mm. The Reynolds number based on the relative velocity $\pi D_o n$ and the diameter of the outer wall is $Re = \frac{\rho \pi D_o^2 n}{\eta} = 9.36 \times 10^5$ and the Mach number is $M = \frac{\pi D_o n}{a} = 0.136$, where n=3000 rpm.

The geometry of the configuration visualized by the zero level-set contour ϕ_0 and two axial cross sections through the computational mesh are shown in Fig. 6. The mesh has approx. 275 Mio. cells and is adapted according to the relative location of the turbulence generating grid and the fan blades. The analysis is performed at a fixed operating point defined by the flow rate coefficient $\Phi = \frac{4\dot{V}}{\pi^2 D_o^3 n} = 0.165$ and the tip-gap width $s/D_o = 0.01$. The ratio between the diameter of the outer casing wall and the inner diameter of the hub is D_o/D_i = 20/9. The turbulence generating grid consists of perpendicular rods with a quadratic cross section. The geometric quantities defining the turbulence grid $a/D_o = 0.05$ and $b/D_o = 0.2$ are shown in Fig. 6.

Due to a 90° periodicity of the turbulence generating grid geometry and 72° periodicity of the fan blades geometry in the azimuthal direction, a periodic boundary condition, which has been imposed in previous investigations of the axial fan flow [22, 23], where only one out of five blades has been resolved, cannot be used for the current configuration. Therefore, a 360° full-fan computation is conducted. The grid convergence study in [22] has shown that approx. 1 billion cells are required to accurately resolve all flow phenomena around one fan

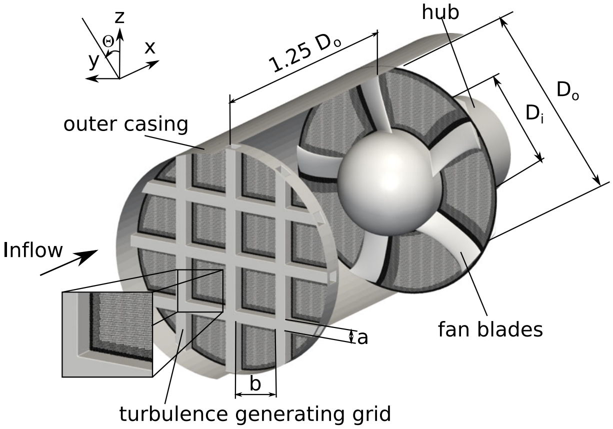

Fig. 6 Geometry of the axial fan configuration visualized by the zero level-set contour ϕ_0 and axial cross sections through the computational mesh for the fluid flow; simulation of the relative rotation between the turbulence generating grid and the fan blades

blade. When considering all five blades and a much larger axial extent of the computational domain due to the turbulence generating grid located $1.25D_o$ upstream of the fan blades, which not only increases the number of cells, but also the number of time steps to obtain a fully developed flow field, the computational costs for an accurate resolution of the flow field are enormous. Therefore, this analysis is not aimed at resolving all the detailed flow physics such as the vortical structures inside the tip-gap or the transition phenomena on the suction side. The reason for this investigation is twofold. On the one hand, the applicability of the kinematic motion level-set interface method to real turbomachinery flows is to be demonstrated and on the other hand, the impact of the interaction of the turbulence grid and the rotating fan on the flow field upstream of the rotating blades, i.e., the definition of the inflow data, is to be shown.

In this study, the size of the finest cells at the wall is approx. $\Delta x_i = 8.0 \times 10^{-4} D_o$, the size of the coarsest cells off the wall is approx. $\Delta x_i = 32.0 \times 10^{-4} D_o$, and the time step is $\Delta t \pi n = 1.064 \times 10^{-4}$. In total, six full rotations of the rotor are simulated. Data of the last two rotations are used for the statistical analysis of the flow field. The computation has been performed on 4608 CPUs and the total CPU time to simulate 6 full rotations is approx. 340 h.

In Fig. 7 the instantaneous vortical structures of the flow field are visualized by the contours of the Q-criterion and colored by the relative Mach number in the frame of reference of the fan blades. The maximum relative Mach number is located near the tip-gap region due to the rotation of the blades. The grid generated vortical structures and the turbulent fluctuations, which are generated by the flow separating from the rectangular rods, decay in the axial direction due to turbulent mixing such that a more homogeneous turbulent flow field develops. Several flow phenomena, i.e., passage vortices at the hub of the blades, which lead to a turbulent transition of the boundary layer, and tip-gap vortices at the blade tips, which are generated by the tip-leakage flow inside the tip gap and produce highly unsteady turbulent wakes, are observed. Note that the details of those phenomena have been discussed in [22, 23]. Turbulent transition on the suction side of the blades has been also observed in [22] under the same operating conditions. This mechanism is not captured in the current solution due to the underresolution of the near-wall flow.

Figure 8 shows the impact of the turbulence generating grid on the flow field over the axial fan. The contours of the vorticity magnitude $|\vec{\omega}|/\pi n$ of the instantaneous flow field at

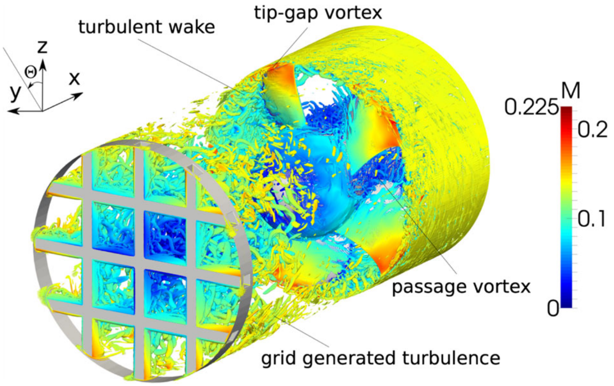

Fig. 7 Contours of the Q-criterion of the instantaneous flow field colored by the relative Mach number in the frame of reference of the fan blades

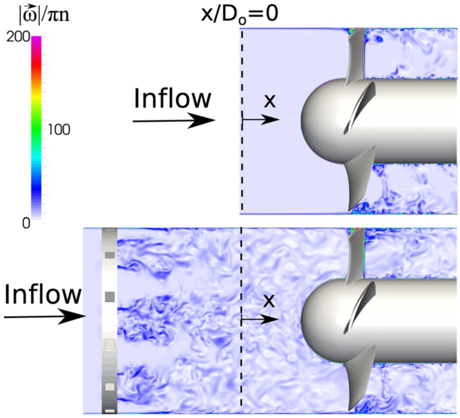

Fig. 8 Contours of the vorticity magnitude $|\vec{\omega}|/\pi n$ of the instantaneous flow field at $\Theta = -45°$ for the 72° computation using periodic boundary conditions in the circumferential direction without a turbulence generating grid from [22] (*top*) and the current 360° computation with a turbulence generating grid at the inflow (*bottom*); the *dashed line* denotes the axial location $x/D_o = 0.0$

$\Theta = -45°$ for the 72° computation using periodic boundary conditions in the circumferential direction without a turbulence generating grid on a mesh with approx. 50 Mio. cells from [22] (top) are compared with the current 360° computation including a turbulence generating grid at the inflow (bottom). Note that the off-wall mesh resolution is identical in both cases. A major impact on the overall flow field upstream of the rotating blades is evident, which also affects the flow field near the tip-gap region and downstream of the fan blades. Without the turbulence generating grid, the inflow is free of vorticity fluctuations. The main vortical structures are generated by the fan blades in the tip-gap regions by the

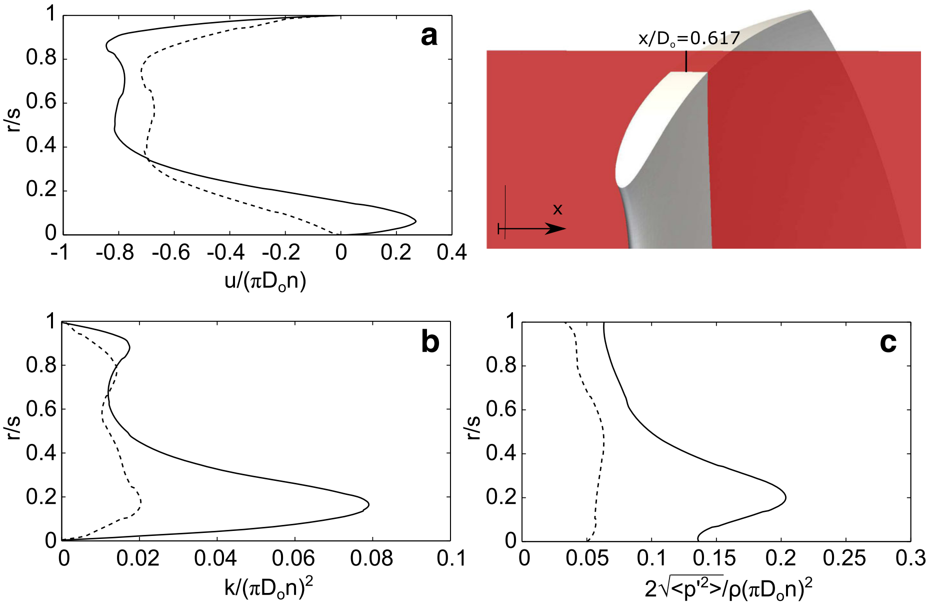

Fig. 9 Radial distribution of the time-averaged axial velocity (**a**), turbulent kinetic energy (**b**) and pressure fluctuations (**c**) inside the tip-gap; without a turbulence generating grid (—); with a turbulence generating grid at the inflow (- - -)

tip-gap vortices and their highly unsteady turbulent wakes, downstream of the blades by the turbulent wakes generated at their trailing edges, and at the hub by the turbulent boundary layer. With the turbulence generating grid, high vorticity is generated upstream of the fan, which affects the flow field around the blades and increases the vorticity especially in the tip-gap regions and the blade wake regions.

Figure 9a–c clearly show the strong impact of the turbulence generating grid on the flow field inside the tip gap. Compared to the case without inflow turbulence, which shows a positive axial velocity close to the blades tip surface Fig. 9a, due to the separation bubble inside the tip gap, for the case with the grid generated turbulence the separation bubble reattaches closer to the pressure side such that no positive axial velocity is observed over the entire gap width. This effect is also reflected in the radial distributions of the pressure fluctuations Fig. 9b and the turbulent kinetic energy Fig. 9c, where the peaks close to the blade-tip surface disappears for the case with inflow turbulence. However, as mentioned before, for a reliable quantitative analysis of the impact of the turbulence generating grid on the tip-leakage flow dynamics and the unsteady vortical flow features in the tip-gap region, a higher grid resolution is required [22, 23].

The instantaneous contours of the axial Mach number M_x at $\Theta = -45°$ and at three time steps, i.e, t_0, $t_0 + \Delta T$, and $t_0 + 2\Delta T$ with $\Delta T \pi n = \frac{\pi}{15}$ which corresponds to 24° rotation angle of the blades, in Fig. 10 show that the flow is accelerated due to the displacement defined by the rods and deadwater regions are generated downstream of each rod, which results in multiple free-shear layers that become unstable and strongly mix further downstream of the turbulence generating grid. This leads to an almost homogenous turbulent flow that interacts with the blades. In addition, Fig. 11 illustrates instantaneous contours of the axial Mach number M_x at $x/D_o = 0.617$, i.e., this cross section cuts fan blades, and the same three time steps. The dashed line denotes the location of the plane at $\Theta = -45°$ from Fig. 10 and the turbulence generating grid is indicated by the black solid lines. The

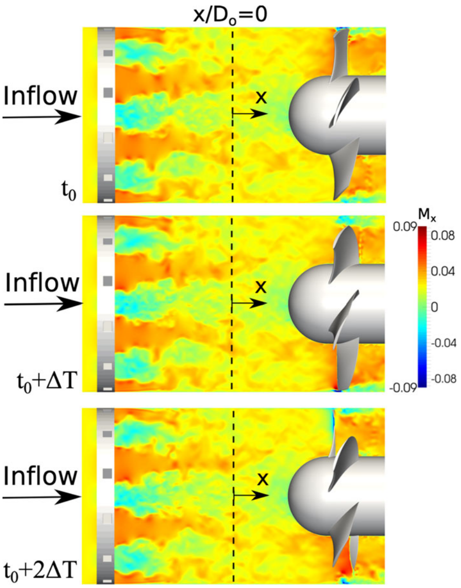

Fig. 10 Instantaneous contours of the axial Mach number M_x at $\Theta = -45°$ and at three time steps, i.e, t_0, $t_0 + \Delta T$, and $t_0 + 2\Delta T$ with $\Delta T \pi n = \frac{\pi}{15}$ which corresponds to 24° rotation angle of the blades; the *dashed line* denotes the axial location $x/D_o = 0.0$

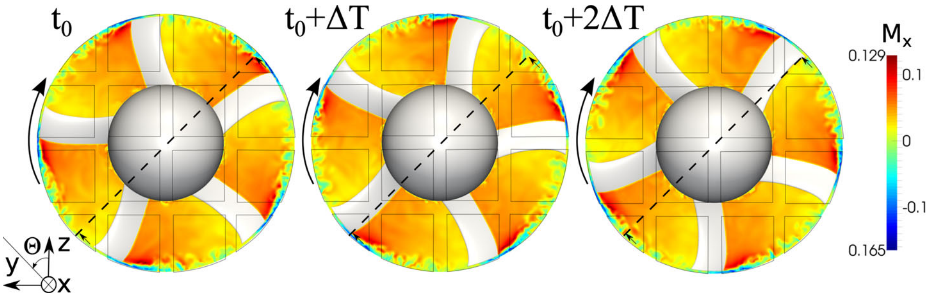

Fig. 11 Instantaneous contours of the axial Mach number M_x at $x/D_o = 0.617$ and at three time steps, i.e, t_0, $t_0 + \Delta T$, and $t_0 + 2\Delta T$ with $\Delta T \pi n = \frac{\pi}{15}$ which corresponds to 24° of rotation; the *black solid lines* indicate the relative position of the turbulence generating grid; the *dashed line* denotes the location of the plane at $\Theta = -45°$ from Fig. 10

visualization of the flow field shows that the large-scale structures hardly change as a function of time and possess an azimuthal periodicity which is related to the five blades of the fan.

The present computational results are compared to experimental data from [49] in Fig. 12 where the pressure coefficient $\Psi = \Delta p/(\frac{\pi^2}{2}\rho D_o^2 n^2)$ is shown as a function of the flow coefficient $\Phi = \frac{4\dot{V}}{\pi^2 D_o^3 n}$ with Δp being the average pressure increase over the fan blades. In addition, previous numerical results from [22, 23] are depicted. It is evident that a good agreement between the LES and the experimental data is achieved. Furthermore, in Fig. 13 experimental and numerical data of the radial distributions of the time-averaged velocity u/u_∞ (left) and the velocity fluctuations u_{rms}/u_∞ at $x/D_o = 0.0$ are juxtaposed for the present case. This comparison also evidences the good experimental-numerical agreement.

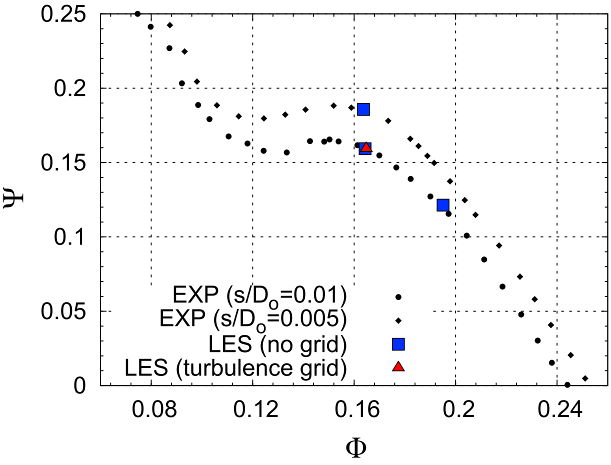

Fig. 12 Pressure coefficient $\Psi = \frac{\Delta p}{(\frac{\pi^2}{2}\rho D_o^2 n^2)}$ versus flow rate coefficient $\Phi = \frac{4\dot{V}}{\pi^2 D_o^3 n}$; experimental data for $s/D_o = 0.01$ and $s/D_o = 0.005$ from [49] compared previous numerical results from [22, 23] without a turbulence generating grid and to present numerical results considering a turbulence generating grid

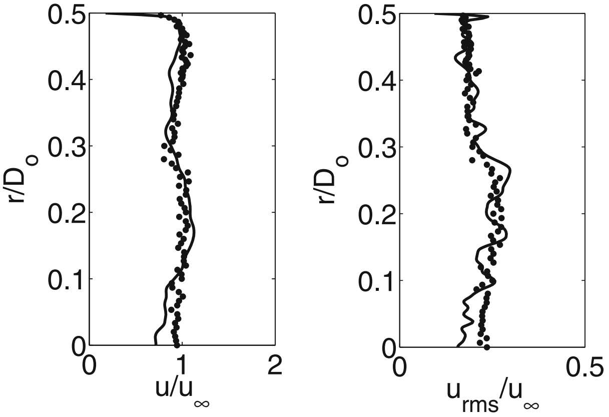

Fig. 13 Radial distributions of the time-averaged velocity u/u_∞ (*left*) and the velocity fluctuations u_{rms}/u_∞ at $x/D_o = 0.0$; experimental data (●) [49] compared to present numerical results (—)

6 Concluding Remarks

A Cartesian cut-cell/level-set method in an unstructured hierarchical Cartesian setup based on a multiple grid approach to simulate rotor-stator interactions in turbulent turbomachinery flows has been presented. The sharp representation of the complex moving boundaries being embedded in the computational domain are described by multiple level-sets. This ensures a strict conservation of mass, momentum, and energy and allows the simulation of multiple relatively rotating boundaries in a fixed frame of reference. An efficient kinematic motion level-set interface method for the rotation of the embedded boundaries, which are described by multiple level-set fields on a computational domain distributed across several processors, has been introduced. Unlike the sliding interface concept the relative rotation of the surfaces to each other can be arbitrary, i.e., the mesh does not have to satisfy constraints being imposed by the predefined relative motion of the boundaries to ensure conservativity.

First, the kinematic motion level-set interface method has been developed by analyzing the generic flow problem of a ducted rotating cylinder. Then, this method has been applied to a large-scale application, i.e, a rotating ducted axial fan with a stationary turbulence generating grid at the inflow to demonstrate its efficiency and applicability to real industrial turbomachinery flows. The results show the good agreement of the numerical and experimental data and a dominant impact of the turbulence generating grid on the flow field upstream of the fan which also affects the flow field near the tip-gap region and downstream of the fan.

Acknowledgements This study was funded under Grant No. 17747N (L238) by the German Federal Ministry of Economics and Technology via the ”Arbeitsgemeinschaft industrieller Forschungsvereinigungen Otto von Guericke e.V. ” (AiF) and the ”Forschungsvereinigung Luft- und Trocknungstechnik e.V.” (FLT). Computing resources were provided by the High Performance Computing Center Stuttgart (HLRS) and by the Jülich Supercomputing Center (JSC) within a Large-Scale Project of the Gauss Center for Supercomputing (GCS).

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Blais, B., Lassaigne, M., Goniva, C., Fradette, L., Bertrand, F.: A semi-implicit immersed boundary method and its application to viscous mixing. Comput. Chem. Eng. **85**, 136–146 (2016)
2. Blades, E., Marcum, D.: A sliding interface method for unsteady unstructured flow simulations. AIAA, 2005–5233 (2005)
3. Ferziger, J.H., Peric, M.: Computational Methods for Fluid Dynamics, 3rd edn. Springer-Verlag, Berlin (2002)
4. Mittal, R., Iaccarino, G.: Immersed boundary methods. Annu. Rev. Fluid Mech. **37**(1), 239–261 (2005)
5. Peskin, C.S.: Flow patterns around heart valves: A numerical method. J. Comput. Phys. **10**(2), 252–271 (1972)
6. Peskin, C.S.: Numerical analysis of blood flow in the heart. J. Comput. Phys. **25**(3), 220–252 (1977)
7. Goldstein, D., Handler, R., Sirovich, L.: Modeling a no-slip flow boundary with an external force field. J. Comput. Phys. **105**(2), 354–366 (1993)
8. Peskin, C.S.: The immersed boundary method. Acta Numerica **11**, 479–517 (2002)
9. LeVeque, R., Li, Z.: The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM J. Numer. Anal. **31**(4), 1019–1044 (1994)
10. Verzicco, R., Mohd-Yusof, J., Orlandi, P., Haworth, D.: Large eddy simulation in complex geometric configurations using boundary body forces. AIAA J. **38**(3), 427–433 (2000)
11. Mohd-Yusof, J.: Combined immersed-boundary/b-spline methods for simulations of flow in complex geometries. Center Turb Res Ann Res Briefs, 317–327 (1997)
12. Hartmann, D., Meinke, M., Schröder, W.: An adaptive multilevel multigrid formulation for Cartesian hierarchical grid methods. Comput. Fluids **37**(9), 1103–1125 (2008)
13. Johansen, H., Colella, P.: A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. J. Comput. Phys. **147**, 60–85 (1998)
14. McCorquodale, P., Colella, P., Johansen, H.: A Cartesian grid embedded boundary method for the heat equation on irregular domains. J. Comput. Phys. **173**, 620–635 (2001)
15. Udaykumar, H., Mittal, R., Shyy, W.: Computation of solid–liquid phase fronts in the sharp interface limit on fixed grids. J. Comput. Phys. **153**(2), 535–574 (1999)
16. Schneiders, L., Hartmann, D., Meinke, M., Schröder, W.: An accurate moving boundary formulation in cut-cell methods. J. Comput. Phys. **235**, 786–809 (2013)
17. Murman, S.M., Aftosmis, M.J., Berger, M.J.: Implicit approaches for moving boundaries in a 3-d Cartesian method. AIAA Paper, 2003–1119 (2003)
18. Iaccarino, G., Verzicco, R.: Immersed boundary technique for turbulent flow simulations. Appl. Mech. Rev. **56**(3), 331–347 (2003)
19. Günther, C., Meinke, M., Schröder, W.: A flexible level-set approach for tracking multiple interacting interfaces in embedded boundary methods. Comput. Fluids **102**, 182–202 (2014)
20. Hartmann, D., Meinke, M., Schröder, W.: Differential equation based constrained reinitialization for level set methods. J. Comput. Phys. **227**(14), 6821–6845 (2008)
21. Hartmann, D., Meinke, M., Schröder, W.: The constrained reinitialization equation for level set methods. J. Comput. Phys. **229**(5), 1514–1535 (2010)
22. Pogorelov, A., Meinke, M., Schröder, W.: Cut-cell method based large-eddy simulation of tip-leakage flow. Phys. Fluids **27**(7), 075106 (2015)
23. Pogorelov, A., Meinke, M., Schröder, W.: Effects of tip-gap width on the flow field in an axial fan. IntJ Heat Fluid Flow. doi:10.1016/j.ijheatfluidflow.2016.06.009 (2016)
24. Pogorelov, A., Meinke, M., Schröder, W.: Impact of periodic boundary conditions on the flow field in an axial fan. AIAA Paper, 2016–0610 (2016)
25. Schneiders, L., Günther, C., Meinke, M., Schröder, W.: An efficient conservative cut-cell method for rigid bodies interacting with viscous compressible flows. J. Comput. Phys. **311**, 62–86 (2016)
26. You, D., Mittal, R., Wang, M., Moin, P.: Computational methodology for large-eddy simulation of tip-clearance flows. AIAA J. **42**(2), 271–279 (2004)
27. You, D., Wang, M., Moin, P., Mittal, R.: Effects of tip-gap size on the tip-leakage flow in a turbomachinery cascade. Phys. Fluids **18**(10), 105102 (2006)
28. You, D., Wang, M., Moin, P., Mittal, R.: Large-eddy simulation analysis of mechanisms for viscous losses in a turbomachinery tip-clearance flow. J. Fluid Mech. **586**, 177–204 (2007)
29. You, D., Wang, M., Moin, P., Mittal, R.: Vortex dynamics and low-pressure fluctuations in the tip-clearance flow. J. Fluids Eng. **129**(8), 1002–1014 (2007)
30. Tyagi, M., Acharya, S.: Large eddy simulation of turbulent flows in complex and moving rigid geometries using the immersed boundary method. Int. J. Numer. Methods Fluids **48**, 691–722 (2005)

31. Posa, A., Lippolis, A., Verzicco, R., Balaras, E.: Large-eddy simulations in mixed-flow pumps using an immersed-boundary method. Comput. Fluids **47**, 33–43 (2011)
32. Fadlun, E.A., Verzicco, R., Orlandi, P., Mohd-Yusof, J.: Combined immersed-boundary finite-difference methods for three dimensional complex flow simulations. J. Comput. Phys. **161**, 35–60 (2000)
33. Hirt, C.W., Amsden, A.A., Cook, J.L.: An arbitrary Lagrangian-Eulerian computing method for all flow speeds. J. Comput. Phys. **14**(3), 227–253 (1974)
34. Sutherland, W.: The viscosity of gases and molecular force. Philos. Mag. **36**, 507–531 (1893)
35. Udaykumar, H.S., Mittal, R., Rampunggoon, P., Khanna, A.: A sharp interface cartesian grid method for simulating flows with complex moving boundaries. J. Comput. Phys. **174**(1), 345–380 (2001)
36. Lintermann, A., Schlimpert, S., Grimmen, J.H., Günther, C., Meinke, M., Schröder, W.: Massively parallel grid generation on hpc systems. Comput. Methods Appl. Mech. Eng. **277**, 131–153 (2014)
37. Boris, J.P., Grinstein, F.F., Oran, E.S., Kolbe, R.L.: New insights into large eddy simulation. Fluid Dyn. Res. **10**(4-6), 199–228 (1992)
38. Alkishriwi, N., Meinke, M., Schröder, W.: A large-eddy simulation method for low Mach number flows using preconditioning and multigrid. Comput. Fluids **35**(10), 1126–1136 (2006)
39. Meinke, M., Schröder, W., Krause, E., Rister, T.: A comparison of second- and sixth-order methods for large-eddy simulation. Comput. Fluids **31**(4–7), 695–718 (2002)
40. Renze, P., Schröder, W., Meinke, M.: Large-eddy simulation of film cooling flows at density gradients. Int. J. Heat Fluid Flow **29**(1), 18–34 (2008)
41. Rütten, F., Schröder, W., Meinke, M.: Large-eddy simulation of low frequency oscillations of the Dean vortices in turbulent pipe. Phys. Fluids **17**(3), 035107 (2005)
42. Mavriplis, D.J.: Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. AIAA, 2003–3986 (2003)
43. Liou, M.-S., Steffen, C.J. Jr.: A new flux splitting scheme. J. Comput. Phys. **107**(1), 23–39 (1993)
44. Berger, M.J., Aftosmis, M.J.: Progress towards a Cartesian cut-cell method for viscous compressible flow. AIAA, 2012–1301 (2012)
45. Hartmann, D., Meinke, M., Schröder, W.: A strictly conservative Cartesian cut-cell method for compressible viscous flows on adaptive grids. Comput. Methods Appl. Mech. Eng. **200**(9–12), 1038–1052 (2011)
46. Jameson, A., Mavriplis, D.: Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh. AIAA J. **24**(4), 616–618 (1986)
47. Schneiders, L., Günther, C., Grimmen, J., Meinke, M., Schröder, W.: Sharp resolution of complex moving geometries using a multi-cut-cell viscous flow solver. AIAA Paper, 2015–3427 (2015)
48. Sturm, M., Carolus, T.: Tonal fan noise of an isolated axial fan rotor due to inhomogeneous coherent structures at the intake. Noise Control Eng. J. **60**(6), 699–706 (2012)
49. Zhu, T., Carolus, T.H.: Experimental and numerical investigation of the tip clearance noise of an axial fan. In: Proceedings of ASME Turbo Expo 2013 (GT2013-94100). San Antonio (2013)