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.

$$S_{\mathrm{OH}} = A_{\mathrm{OH}} \exp\left(-\frac{E_{\mathrm{OH}}}{R_{\mathrm{g}}T}\right)(1 - X_{\mathrm{OH}})(c_{\mathrm{NCO},0} - 2c_{\mathrm{W},0}X_{\mathrm{W}} - c_{\mathrm{OH},0}X_{\mathrm{OH}}), \tag{1}$$

$$S_{\mathrm{W}} = A_{\mathrm{W}} \exp\left(-\frac{E_{\mathrm{W}}}{R_{\mathrm{g}}T}\right)(1 - X_{\mathrm{W}}), \tag{2}$$

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

$$S_T = \frac{(-\Delta H_{\mathrm{OH}})\, c_{\mathrm{OH},0}}{\rho_{\mathrm{PU}} c_{p,\mathrm{f}}} \frac{\mathrm{D}X_{\mathrm{OH}}}{\mathrm{D}t} + \frac{(-\Delta H_{\mathrm{W}})\, c_{\mathrm{W},0}}{\rho_{\mathrm{PU}} c_{p,\mathrm{f}}} \frac{\mathrm{D}X_{\mathrm{W}}}{\mathrm{D}t} + \sum_{i}^{N} \frac{\left(-\Delta H_{\mathrm{v},i}\right)}{c_{p,\mathrm{f}}} \frac{\mathrm{D}w_i}{\mathrm{D}t}, \tag{3}$$

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]:

$$\sum_{i=1}^{N} p_i + p_{\mathrm{air}} - p_{\mathrm{PU}} = \rho_{\mathrm{PU}} \left[R \frac{\mathrm{d}^2 R}{\mathrm{d}t^2} + \frac{3}{2} \left(\frac{\mathrm{d}R}{\mathrm{d}t} \right)^2 \right] + \frac{2\gamma}{R} + \frac{4\mu_{\mathrm{PU}}}{R} \frac{\mathrm{d}R}{\mathrm{d}t}, \tag{4}$$

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

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{p_i R^3}{R_{\mathrm{g}} T} \right) = 3 D_i R^2 \left. \frac{\partial c_i}{\partial r} \right|_{r=R}, \tag{5}$$

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

$$\frac{\partial c_i}{\partial t} + \frac{R^2}{r^2} \frac{\mathrm{d}R}{\mathrm{d}t} \frac{\partial c_i}{\partial r} = \frac{D_i}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial c_i}{\partial r} \right) + r_i, \tag{6}$$

where r_i is the reaction source term, which can be expressed as

$$r_i = \begin{cases} c_{\mathrm{W},0} S_W & \text{if } i = \mathrm{CO}_2 \\ 0 & \text{if } i \neq \mathrm{CO}_2, \end{cases} \tag{7}$$

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

$$c_i|_{r=R} = H_i p_i, \tag{8}$$

$$\left. \frac{\partial c_i}{\partial r} \right|_{r=S} = 0, \tag{9}$$

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 interested 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:

$$\frac{\partial n(\mathrm{v})}{\partial t} + \nabla \cdot (\mathbf{U}_f n(v)) + \frac{\partial}{\partial \mathrm{v}}[G(\mathrm{v})n(\mathrm{v})] = \frac{1}{2}\int_0^{\mathrm{v}} \beta(\mathrm{v}', \mathrm{v}\text{-}\mathrm{v}')n(\mathrm{v}')n(\mathrm{v}\text{-}\mathrm{v}')\,\mathrm{dv}' - \int_0^{\infty} \beta(\mathrm{v}, \mathrm{v}')n(\mathrm{v})n(\mathrm{v}')\,\mathrm{dv}'. \tag{11}$$

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 $G(\mathrm{v}) = \mathrm{dv}/\mathrm{d}t$ 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:

$$m_k(t) = \int_0^{\infty} n(\mathrm{v})\mathrm{v}^k\,\mathrm{dv} . \tag{12}$$

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).

$$\mathbf{y}_i = \mathcal{M}\left(\mathbf{u}_i\ ;\ \mathbb{D}\right) \qquad\qquad ;\ \mathbf{u}_i \in \mathbf{U}_{max}\ . \tag{21}$$

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

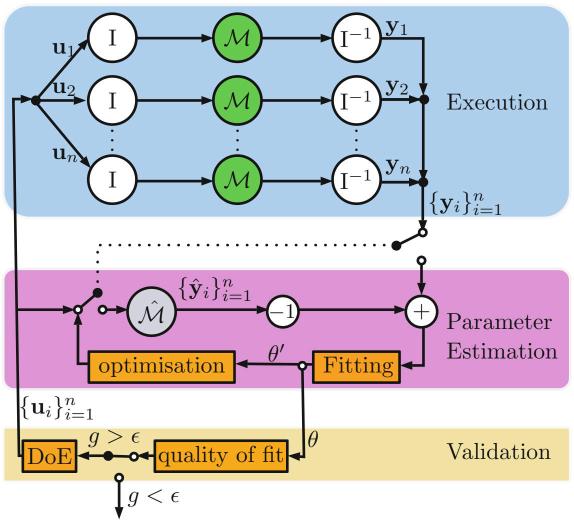

Fig. 1 Workflow of the model-based design of experiments procedure. The design of experiments (DoE) procedure generates a set of inputs $\{\mathbf{u}_i\}_{i=1}^n$, and a detailed simulation is performed using the detailed model $\mathcal{M}$, producing a set of outputs $\{\mathbf{y}_i\}_{i=1}^n$. Because 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).

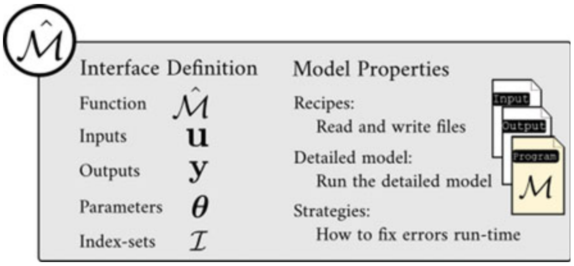

Fig. 2 Illustration of the data that is provided in the definition of a MoDeNa surrogate model. All information is provided as Python dictionaries and the model properties are implemented using templates provided by the framework. This means that objects can be de-serialized directly from a database of a flat file

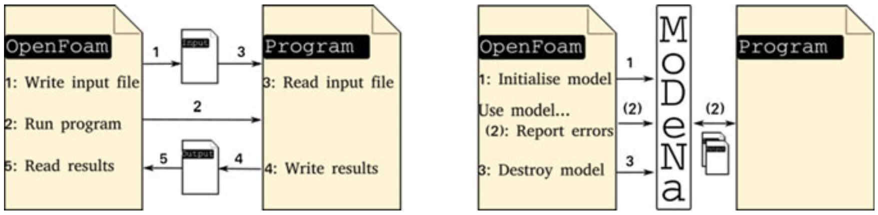

Fig. 3 Illustration of a "exhaustive" (left) and MoDeNa-style (right) coupling between two applications, or detailed models. The MoDeNa approach can be thought of as a combination between a library API and the "exhaustive" input/output file approach. The benefit is that the API will look the same for all models, and there is no intrusion of code specific to other applications

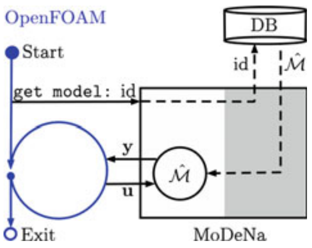

Fig. 4 Illustration of the workflow of an OpenFOAM® application (blue) in which calls to the MoDeNa software framework have been embedded (solid black lines). The workflow is ideal in the sense that error checks and exceptions related to the MoDeNa surrogate model have been omitted. The low-level C-layer 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:

$$\mu_{\mathrm{PU}} = A \exp\left(\frac{E}{R_{\mathrm{g}} T}\right) \left(\frac{X_{\mathrm{OH,g}}}{X_{\mathrm{OH,g}} - X_{\mathrm{OH}}}\right)^{B + C X_{\mathrm{OH}}}, \qquad (26)$$

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
$c_{p,\text{pol}}$ (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
$-\Delta H_{\text{OH}}$ (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
$-\Delta H_{\text{W}}$ (J mol^{-1})	8.60×10^4
$c_{\text{OH},0}$ (mol m^{-3})	3765
$c_{\text{NCO},0}$ (mol m^{-3})	3765
$c_{\text{W},0}$ (mol m^{-3})	0
T_0 (K)	300
$n_{\text{b},0}$ (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]:

$$\mu_f\,(T, X_{\text{OH}}, \dot{\gamma}) = A_{\text{OH}} \exp\left(\frac{E_{\text{OH}}}{R_g T}\right) \times \left(\mu_\infty + (\mu_0 - \mu_\infty)\left(1 + (\dot{\gamma}\bar{\Lambda})^{\zeta}\right)^{\frac{n-1}{\zeta}}\right) \tag{27}$$

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,

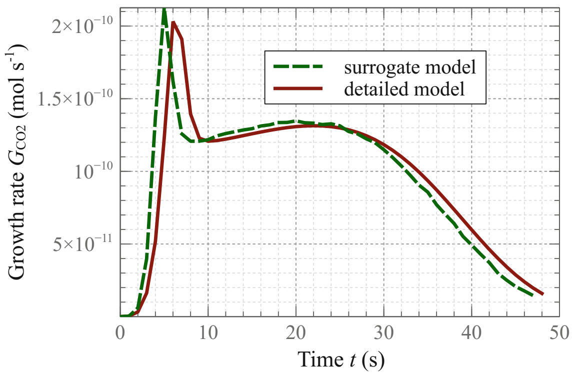

Fig. 5 The bubble growth rate as calculated by the detailed model (see Sect. 2.2) and the surrogate model Eq. 25

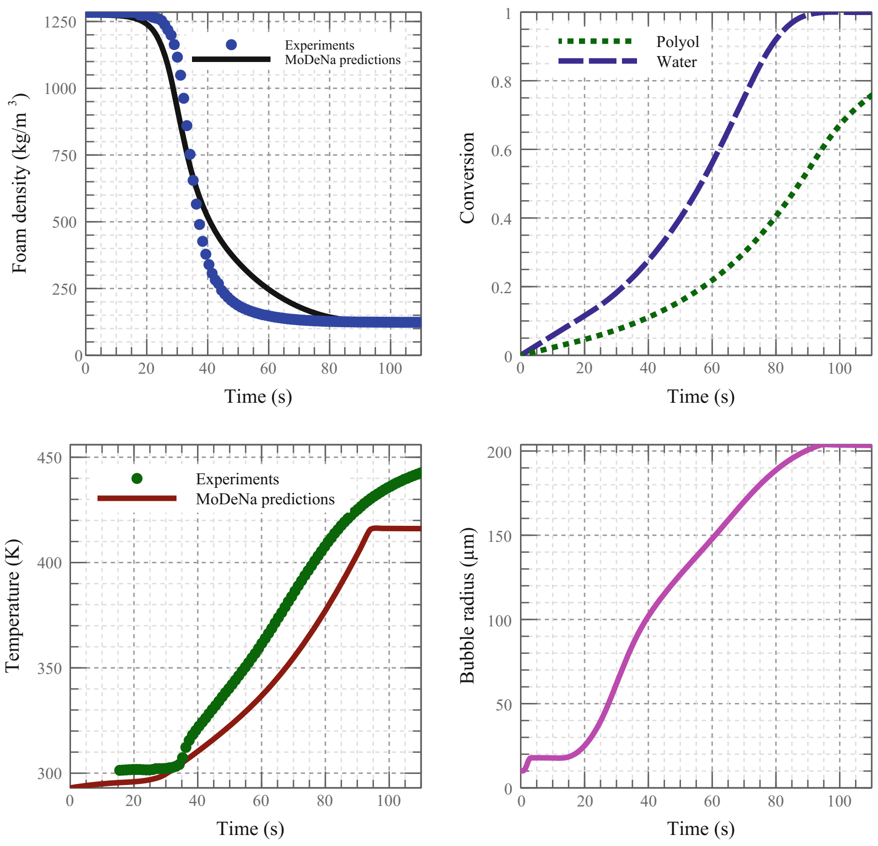

Fig. 6 `MODENAFoam` predictions of foam density (top-left), conversion of components (top-right), temperature profile (bottom-left), and bubble radius (bottom-right) as a function of time

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.