

This item is the archived peer-reviewed author-version of:

Experimental methods in chemical engineering : computational fluid dynamics/finite volume method–CFD/FVM

Reference:

Van Hoecke Laurens, Boeye Dieter, Gonzalez-Quiroga Arturo, Patience Gregory S., Perreault Patrice.- Experimental methods in chemical engineering : computational fluid dynamics/finite volume method–CFD/FVM
The Canadian journal of chemical engineering - ISSN 1939-019X - 101:2(2023), p. 545-561
Full text (Publisher's DOI): <https://doi.org/10.1002/CJCE.24571>
To cite this reference: <https://hdl.handle.net/10067/1892840151162165141>

Experimental methods in chemical engineering: Computational fluid dynamics/finite volume method—CFD/FVM

Laurens Van Hoecke^a, Dieter Boeye^a, Arturo Gonzalez-Quiroga^b, Gregory S. Patience^c, Patrice Perreault^{d,e,*}

^a*Sustainable Energy, Air and Water Technology, Department of Bioscience Engineering, University of Antwerp, Groenenborgerlaan 171, 2020 Antwerp, Belgium*

^b*UREMA Research Unit, Department of Mechanical Engineering, Universidad Del Norte, Barranquilla 25138, Atlántico, Colombia*

^c*Chemical Engineering, Polytechnique Montréal, C.P. 6079, Succ. “CV”, Montréal, H3C 3A7, Québec, Canada*

^d*Blue App, University of Antwerp, Belgium, Middelheimlaan 1, 2020 Antwerp, Belgium*

^e*Institute of Environment and Sustainable Development (IMDO), University of Antwerp, Groenenborgerlaan 171, 2020 Antwerp, Belgium*

Abstract

Computational fluid dynamics (CFD) applies numerical methods to solve transport phenomena problems. These include, for example, problems related to fluid flow comprising the Navier–Stokes transport equations for either compressible or incompressible fluids together with turbulence models and continuity equations for single and multi-component (reacting and inert) systems. The design space is first segmented into discrete volume elements (meshing). The finite volume method, the subject of this article, discretizes the equations in time and space to produce a set of non-linear algebraic expressions that are assigned to each volume element—cell. The system of equations is solved iteratively with algorithms like the semi-implicit method for pressure-linked equations (SIMPLE) and the pressure implicit splitting of operators (PISO). CFD is especially useful for testing multiple design elements because it is often faster and cheaper than experiments. The downside is that this numerical method is based on models that require validation to

*Corresponding author

Email address: `Patrice.Perreault@uantwerpen.be` (Patrice Perreault)

check their accuracy. According to a bibliometric analysis, the broad research domains in chemical engineering include: (1) dynamics and CFD-DEM (2) fluid flow, heat transfer and turbulence, (3) mass transfer and combustion, (4) ventilation and environment, and (5) design and optimization. Here, we review the basic theoretical concepts of CFD and illustrate how to set up a problem in the open-source software OpenFOAM to isomerize *n*-butane to *i*-butane in a notched reactor under turbulent conditions. We simulated the problem with 1000, 4000, and 16 000 cells. According to the Richardson extrapolation, the simulation underestimates the adiabatic temperature rise by 7 % with 16 000 cells.

Keywords: computational fluid dynamics, finite volume, Richardson extrapolation, grid convergence index

1. INTRODUCTION

Computational fluid dynamics (CFD) has evolved over the last 40 years to become an independent field of study within chemical engineering and an indispensable tool that applies across many other engineering disciplines and science categories. In 2020 alone, Web of Science Core Collection(WoS) indexed over 7500 articles with CFD in the *Topic* search criteria. Of these, 1242 articles were in journals WoS classified as chemical engineering.

The increase in available computing power has expanded the breadth and scope of the problems researchers consider. In the 1970s, researchers simulated transonic flows based on non-linear potential equations. Examples of the problems they now address include: micro-kinetic models in particle-resolved packed bed reactors;[1] vortex reactors, including catalytic reactions at the interface of moving catalyst particles;[2, 3] detailed kinetic modelling with tens of thousands of reactions and thousands of chemical species (from GRI-MECH 3.0 with its 325 reactions and 53 species for natural gas combustion;[4] and, solid fuel conversion, including secondary gas-phase reactions[5]), which has advanced with the development of automated reaction mechanism algorithms like the Reaction Mechanism Generator[6] or Genesys.[7]

Kinetic model reduction techniques minimize otherwise prohibitively large computational costs, which reaches 90 % of the total in the case of heterogeneous systems.[8] The OpenSMOKE++[9] numerical applies computation cost minimization (CCM) techniques.[10] Kinetic model reduction techniques

include range from lumping procedures,[11] to on-the-fly tabulation techniques and machine learning.[12] These methods automatically reduce the complexity of the kinetic models by selecting the most important chemical species to simulate. In addition, these techniques relax the requirement of a fine mesh or the necessity for stiff chemistry solvers to simulate fast evolving intermediates like gas-phase radicals or fast reversible reactions. The reduction of stiffness is also key to avoid prohibitive computational costs.[9]

Multiphase flow is another active research field. The discrete element method (DEM) characterizes particle motion[13], while CFD-DEM couples the continuous phase with the particulate phase.[14] Population balance modelling simulates the dynamic evolution of polydispersed particles in applications ranging from crystal growth,[15] to soot formation and material synthesis,[16] biomass pyrolysis,[17], comminution[18] crystallization, and liquid–liquid extraction. Population balance equations (PBE) allows the disperse phase to have distinct properties from the continuous phase (diameter, velocity, and mass) to simulate multiphase flow.[19] PBE in bubbly flows account for bubble breakup and coalescence and is a powerful tool to estimate the various closures required in the Eulerian–Eulerian approach (drag, lift, and wall lubrication forces, in addition to turbulent dispersion and virtual mass). [20]

This tutorial review is part of a series on experimental methods in chemical engineering that highlights key challenges and succinctly reviews recent CFD research areas.[21] We showcase CFD as a viable alternative to tedious and expensive experiments by means of virtual prototyping. CFD, however, is difficult since it relies heavily on numerical schemes and models for physics, which have to be validated.[22, 23] Often, it is possible to reach a solution, but to achieve the correct solution, it requires an understanding of the workflow of a CFD project.[24] This article provides an entry-level assessment of the theory behind finite volume method (FVM)-CFD simulations and how to correctly set up a CFD study to achieve reliable and accurate results. An example case performed with the open-source software package OpenFOAM illustrates how to set up the problem.[25] Other numerical techniques to solve the transport equations include finite difference, finite element methods, and population balance modelling coupled with CFD,[19] which will be addressed in future tutorial reviews. The scope of this review is restricted to FVM as the other techniques require a distinct treatment due to their uniqueness/specificity and complexity.

2. THEORY

2.1. Mathematics in CFD

2.1.1. Governing Principles

CFD codes characterize fluid flow, including granular phases, which are treated as pseudo-fluids. The four main constitutive relations (conservation laws) to fully describe reacting fluid flow of Newtonian fluids are:

- The continuity equation (law of mass conservation)

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{v}) \quad (1)$$

- The momentum or Navier–Stokes equations, which is an extension of Newton’s second law for fluid motion (Equation (2)) to calculate pressure and velocity fields. It looks more complex than $\mathcal{F} = m \cdot a$ for solid body motion. The source term, \mathcal{Q}_v , represents external forces, including gravity acting on the fluid per unit volume (i.e. $\rho \mathbf{g}$).

$$\begin{aligned} \frac{\partial}{\partial t}[\rho \mathbf{v}] + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) - \mu \nabla \cdot \left([\nabla \otimes \mathbf{v} + (\nabla \otimes \mathbf{v})^T] - \frac{2}{3} \nabla \mathbf{v} \cdot I \right) \\ = -\nabla \mathcal{P} + \mathcal{Q}_v \end{aligned} \quad (2)$$

- The species balance equation is required for flows involving multicomponent mixing and/or reactions (Equation (3)). This generalized expression of the convection-dispersion equation includes source terms for chemical reactions, \mathcal{R}_k . OpenFOAM includes an additional source term \mathcal{Q}_Y to account for porous zones and moving reference frames without having to modify and recompile the solver.

$$\frac{\partial}{\partial t}[\rho Y_k] + \nabla \cdot (\rho Y_k \mathbf{v}) = \nabla \cdot (\rho D_{\text{eff},k} \nabla Y_k) + \mathcal{R}_k \quad (3)$$

- The conservation of energy equation (Equation (2.1.1)) calculates the temperature over the simulation domain at every time step, based on enthalpy and reaction rates. Again, OpenFOAM allows for source terms to be soft-coded.

$$\begin{aligned} \frac{\partial}{\partial t}[\rho H] + \nabla \cdot (\rho H \mathbf{v}) + \frac{\partial}{\partial t}[\rho \mathcal{K}] + \nabla \cdot (\rho \mathcal{K} \mathbf{v}) - \frac{\partial}{\partial t}[\mathcal{P}] \\ = \nabla \cdot (\rho \alpha_{\text{eff}} \nabla H) + \mathcal{Q}_{\mathcal{R}} \end{aligned} \quad (4)$$

The six physico-chemical properties expressed in the above relationships include pressure (\mathcal{P}), velocity (\mathbf{v}), density (ρ), mass (mole) fraction (Y), enthalpy (H), and temperature (T), which are linked by an equation of state when at thermodynamic equilibrium. The ideal gas law is one example of an equation of state ($\mathcal{P} = \rho RT$). The challenge in resolving this set of equations resides in the fact that both pressure and velocity fields must be calculated via the momentum equation. For turbulent reacting flows, additional computational challenges arise in modelling turbulence and simulating the chemistry-turbulence interactions.[26]

2.1.2. Finite Volume Method

In the FVM, the entire design space is divided into many small, non-overlapping control volumes called cells, and this process is known as meshing. We evaluate the primary variables of the Navier–Stokes equations in each cell, which determine how each is transported through the mesh:

$$\int_V \left[\frac{\partial \phi}{\partial t} (\rho \phi) \right] dV + \int_V [\nabla \cdot (\rho \mathbf{v} \phi)] dV = \int_V \nabla \cdot (\Gamma^\phi \nabla \phi) dV + \int_V Q_\phi dV \quad (5)$$

where, ϕ is a fluid property, T , and \mathbf{v} , is the turbulent kinetics or energy. Equation (5) is solved for each transport variable ϕ in the system and the transport equation represents the conservation of a variable in a FVM cell. This equation thus allows to solve all the conservative equations. In that sense, CFD-FVM can be truly understood as a computational transport phenomenon.[27] The following fluid properties are to be resolved:[28, 29]

- The first term is the *unsteady term* or *accumulation term* and represents the accumulation of a certain variable within a control volume over time.
- The second term represents the *convection term* as it includes the velocity vector.
- The third term is the *diffusive term* and accounts for changes in fluid properties due to gradients in the flow.
- The *source term* (last term) comprises phenomena related to the local production and destruction of a fluid property within the control volume (e.g., external forces in the momentum balance equation).

The FVM model assumes that the value of a variable in a cell is stored at the cell centre and all variables vary linearly across the cell, both in time and space. The overall solution involves computing the transport equation for each cell independently. In order to reach a solution, boundary conditions are imposed for each transport variable. The requirements for these cells are that they are non-overlapping, their faces are planar, and do not protrude into each other, so theoretically they could be of any shape. In practice, regular shapes are best. Although the solution is obtained by solving the transport equation for each cell, the value in each cell also depends on the values of the neighbouring cells, which are transmitted through the diffusion and convection terms. The divergence operator, associated with the diffusion and convection terms, highlights the dependency on the vector field surrounding the cell. Assessing this vector field would be impractical with volume integrals. For this reason we convert them to surface integrals using the Divergence or Gauss–Ostrogradsky theorem, which states that the volume integral can be replaced by a surface integral of the flux over the surface faces:[30]

$$\int_V \nabla \cdot F \, dV = \int_S F \cdot \hat{n} \, dS \quad (6)$$

An additional benefit of the Gauss–Ostrogradsky theorem is that the equations are now conservative since they are based on the flux at the surface. This means that when a flux enters a cell, the same flux leaves the cell in the opposite direction. However, there is still a surface integral to be solved, which—being a mathematical operation—requires a numerical approximation. To estimate this integral, we assume (1) the surface integral over the entire cell is equal to the sum of the integrals for every face and (2) the value of a surface integral of a face can be approximated by the value at the centre of the face. Smaller cells increase the accuracy.

These two assumptions transform the integral to a sum of the values at the face centre, which is an essential step to discretize the integrals to solve numerically. This shifts the problem to finding the value at the *face* centres while only the values of the *cell centres* are known. Therefore, we must interpolate to determine the value from the cell centre to the face centre. The three main linearization or discretization schemes are (Figure 1):

- Linear differencing
- Upwind differencing

- Linear upwind differencing

In the linear differencing scheme, the value at the face centre, ϕ_F , is computed as the sum of the cell centre, ϕ_P , and it's neighbour cell centre, ϕ_N , multiplied by a factor f_x to account for the distance between the cell centre and the face. This scheme depends on two points and thus is second-order accurate.

$$\phi_F = f_x \phi_P + (1 - f_x) \phi_N \quad (7)$$

Upwind differencing is simpler and more stable, but it is only first-order accurate, which is convenient to start with. Higher order methods are preferred as they minimize numerical diffusion issues observed with first-order schemes like smearing sharp interfaces. Numerical diffusion comes from truncation errors of the second- and higher order terms, leading to an additional component associated with the diffusion term, proportional to grid size, density, and velocity in the upwind scheme.[27, 31] It states that the value at the face is extrapolated directly from the value at the cell centre. When the flux is positive, that is, pointing towards the outside of the cell, the value at the face equals the value at the owner cell ϕ_P ; when the flux is negative, it takes the value of the neighbouring cell ϕ_N .

$$\phi_f = \begin{cases} \phi_f = \phi_P & \text{for } F \geq 0 \\ \phi_f = \phi_N & \text{for } F \leq 0 \end{cases} \quad (8)$$

Linear upwind differencing lies between the linear differencing and the upwind differencing scheme. It uses a gradient to compute the cell values at the face but this gradient is not computed between the owner cell and the neighbour at that face. In this case, the gradient is computed between the owner cell and the neighbour of the *opposite* cell of the target face. This gradient is then extrapolated to the target face. This scheme is also second-order accurate, and, more importantly, it can be limited for flows with rapid changes, making it more stable than the linear scheme with a slight decrease in accuracy.[27, 33]

The numerical schemes thus provide a solution to solve the convection term. The diffusion term requires additional corrector terms[34] and other approaches deal with the source and unsteady terms.[24] Each term of the transport equation is discretized and placed in a matrix of coefficients, either in the matrix \mathbf{M} , for implicit terms (i.e., their value depends on other variables in the current time step) or they are added to the matrix \mathbf{B} for explicit terms (those that are independent of other variables in the current time

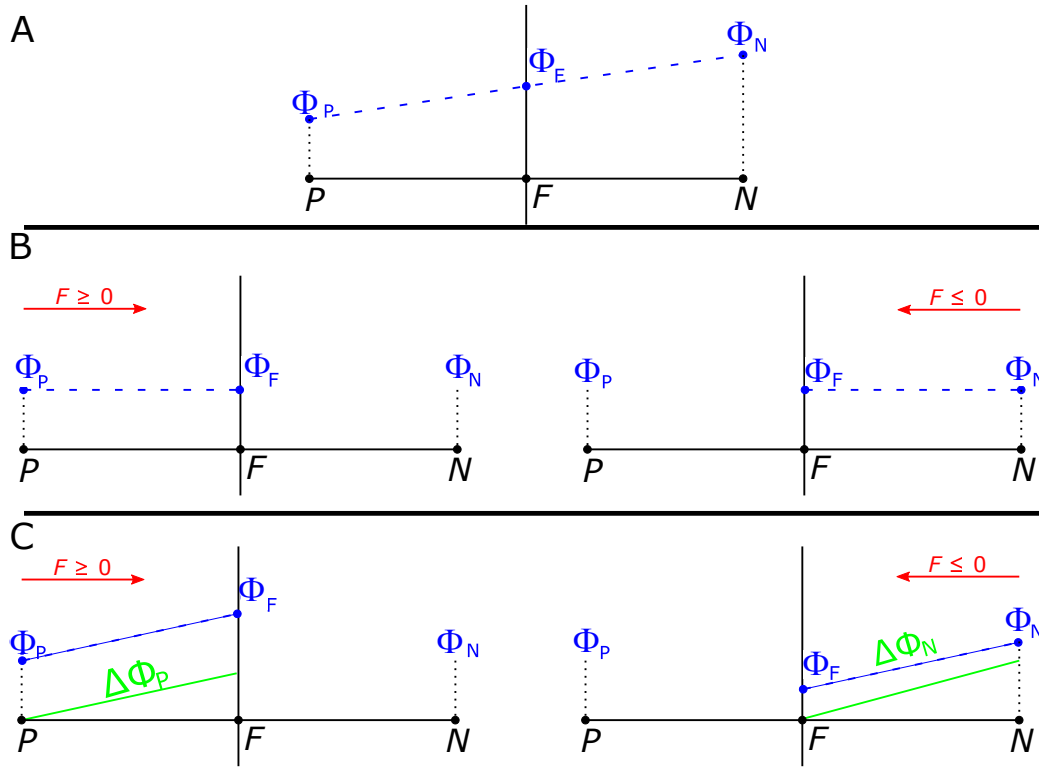


Figure 1: 2D representation of spatial discretization schemes, with owner cell centre P and neighbouring cell centre N . F denotes the value on the face, between the two. The value of F is calculated by: a linear approximation between P and N ((A) linear approximation); extrapolation of the value of the cell centre to the face centre, depending on the sign of the flux, either the value of P or N is selected ((B) upwind scheme); and, the value of the face centre is calculated by the extrapolation of the gradient of the flux within the cell to the face, again this extrapolation depends on the sign of the flux ((C) linear upwind) [32]

step). The entire process comprises solving Equation (9) with the matrix \mathbf{v} , the matrix for the velocity terms:[35, 33]

$$\mathbf{M} \cdot \mathbf{v} = \mathbf{B} \quad (9)$$

This equation is solved iteratively starting from an initial condition we input. The solver evaluates the equation, calculates the errors on the estimate, makes a new estimate, and repeats until the errors fall below a set threshold. Examples of FVM-CFD software packages are OpenFOAM[25], Ansys Fluent,[36] and Star CCM [37]; the latter two are commercial FVM-CFD software packages.

An example of an iterative process is the pressure-velocity coupling, a solution method used mostly for incompressible flows. In this method a system of four equations is iteratively solved until the pressure and velocity both satisfy a user-defined threshold:

- $\mathbf{M} \cdot \mathbf{v} = -\nabla \mathcal{P}$
- $\mathbf{H} = \mathbf{A}\mathbf{v} - \mathbf{M}\mathbf{v}$
- $\nabla(\mathbf{A}^{-1}\nabla \mathcal{P}) = \nabla(\mathbf{A}^{-1}\mathbf{H})$
- $\mathbf{v} = \mathbf{A}^{-1}\mathbf{H} - \mathbf{A}^{-1}\nabla \mathcal{P}$

(The derivation of this system of equations can be found in the supporting information.)

This system can be solved in two methods, either via the black arrow, which denotes the semi-implicit method for pressure-linked equations algorithm (SIMPLE).[38] This system is mostly used for steady-state simulations. The red arrow shows the pressure implicit splitting of operators (PISO) loop [39]; this algorithm is used for transient cases.[35] The momentum predictor step is only solved once, which speeds up the simulation but for accurate solutions, a limiter is needed on the size of the time step. This limiter is known as the Courant, Friedrichs, and Lewy condition or the CFL-number, N_{CFL} .[40]

$$N_{\text{CFL}} = \frac{\Delta t \cdot v}{\Delta x} \quad (10)$$

This condition ensures that the fluid travels less than the distance of one cell during each time step (leak-forward diffusion).[41]

In the field of CFD, we refer to the “user-defined threshold” as the residual and it is one of the key parameters to monitor to confirm that the simulation is working smoothly. Residual values are calculated for each of the transport variables in the simulation, and they are based on the principle of conservation from Equation (6), which makes the residual a scalar field within the simulation domain. Ideally, the deviation between the flux in a cell equals the flux out, so the residual equals zero. Most CFD codes use a representative residual to assess convergence and provide plots of residual versus the number of iterations. The residuals in the plot should decrease until they reach a steady level. When the value of the residual drops below the threshold set by the user this indicates the program has converged to a solution. We stress that converged residuals are only one metric to assess the coherency of the solution. Another criterion to consider is the conservation of the transport properties over the mesh, referred to as the global imbalance. If the global imbalance is small, this means that the quantity is preserved within the complete computational domain. Other qualitative approaches to assess the simulation convergence include the minimum and maximum values of transport properties and point or line monitors within the mesh. These qualitative approaches require a certain knowledge of the system beforehand, to check whether or not the maximum value is realistic.

3. OpenFOAM Case Structure

To set up a simulation case in OpenFOAM requires various folders (blue text Figure 2 and files (green text)). The three main folders are *0*, *constant*, and *system*. The *0* folder contains all the information regarding the initial and boundary conditions for temperature, velocity, pressure, turbulence parameters, and chemical species.[42] We modify an existing (tutorial) simulation case to suit our needs rather than preparing this file from scratch. As a simulation runs, new time folders are written with the same structure as the *0* folder (i.e., containing files for every variable calculated) but named with the selected *write time* in the simulation settings (e.g., folder 10 for the result after 10 seconds in the case of transient simulations).

The *constant* folder contains (1) the chemical species thermo-physical properties, (2) chemical reactions along with their reaction rates (as well as how to model the interaction between chemistry and the fluid flow), (3) the fluid properties (the turbulence model if required, located in the *momentum-Transport* file), and (4) the mesh either imported or defined by the user (in

the *polyMesh* folder).

In the *system* folder, the user specifies the parameters needed for the solution, including simulation settings, discretization schemes, and solvers. The *controlDict* file has the time step, the write step, start and end times, and run-time post-processing requirements, using custom-built or built-in functions (e.g., for residuals, forces, or streamlines). In addition, the *system* folder contains the *blockMeshDict* in which the parameters required for the mesh generation are specified. The meshing tool provided with OpenFOAM is *blockMesh*, which generates the *polyMesh* file in the *constant* folder).

4. TYPICAL WORKFLOW

4.1. Problem Statement

The first step in a CFD study is to establish the goal and select the required information to be extracted bearing in mind that CFD simulations are both time consuming and require massive amounts of storage. We demonstrate the workflow with the liquid phase adiabatic isomerization of *n*-butane in 90% *n*-pentane at 2 MPa.[43] Since the reaction only requires a trace amount of catalyst, we assume the liquid phase is homogeneous. The reaction is first-order and the rate constant is 31.1 h^{-1} at 360 K with an activation energy of $65.7 \text{ kJ} \cdot \text{mol}^{-1}$. The equilibrium constant, K_C , equals 3.03 at 60°C

$$r = k(C_{nC_4} - C_{iC_4}/K_C) \quad (11)$$

We first solved the conventional 1D plug flow model with MATLAB R2019a with the Runge-Kutta method and a variable time step (*ode45*), which provided a baseline to compare results from the CFD simulations (Figure 4). We also tested an irregular-shaped tube to test the simulation. Indeed, the ideal plug flow design equation is independent of the reactor shape (c.f. Fogler’s “Picasso” reactor).[43] The irregular reactor consists of a tube with a sudden expansion, immediately followed by a contraction—a notched reactor (Figure 7). This geometry represents an exaggerated case of coupling between two tubes with a tight joint to illustrate the expansion–contraction effect on *n*-butane conversion.

4.2. One Dimensional Model

In the case of the conventional 1D model, the resolution boils down to the coupled mass (Equation (12)) and energy balances (Equation (13)). The

momentum balance is excluded since the reaction rate is independent of pressure for a constant density liquid-phase reaction.

$$\frac{\partial C_i}{\partial t} = -\frac{\partial(u_z C_i)}{\partial z} + \mathcal{R}_i \quad (12)$$

The energy balance reduces to a linear relationship between the conversion of *n*-butane, and the reactor temperature, neglecting the temperature dependency on the heat capacity, as well as the temperature correction for the enthalpy of reaction.

$$T = T_0 - \frac{X\Delta H}{(C_{p,nC_4} + \theta_{nC_4}C_{p,iC_4} + \theta_{C_5}C_{p,C_5})} \quad (13)$$

4.3. Model and Solver Selection

CFD packages, and OpenFOAM especially, come with various codes to solve different types of problems. A package of CFD codes that solves a specific task is called a solver: simple solvers treat single-phase, isothermal, and incompressible flows while advanced solvers handle problems related to heat transfer, multi-phase flow, and chemical reactions.[3]

Here, we treat a constant density chemical reaction in the liquid phase, so we selected the *rhoReactingFoam* solver, which solves the equations of continuity, momentum, species conservation, and energy sequentially—Equations (1–2.1.1) (Figure 5—statements referring to the equations in the source code are shown in colour). The solution algorithm starts with the continuity equation, solved prior to entering in the outer loop. The outer loop then starts with the momentum equation (Equation (2)), followed by the species transport equation (Equation (3)) and the energy equation (Equation (2.1.1).) The energy equation is solved considering enthalpy or internal energy (the former here), and the temperature is derived from:

$$C_p = \left[\frac{dH}{dT} \right]_p \quad (14)$$

The solution algorithm then enters in the inner loop, which consists of solving the pressure–velocity coupling and the continuity equation. After the solver has completed a user-defined number of inner iterations, the turbulence parameters are calculated and the numerical values of D_{eff} and α_{eff} are updated to account for the turbulence: The effective diffusivity of each

species is the sum of the molecular and turbulence diffusivity (ratio of the molecular viscosity and density and that of the turbulent kinematic viscosity and the N_{Sc} , respectively). As a final step, ρ , C_p and k are updated. The solution algorithm finally returns to the momentum equation and iterates over this outer loop a user-defined number of times. Once the number of programmed iterations is reached (or the accuracy is reached), the time step is updated and the solution algorithm repeats until the final time step.

5. TYPICAL WORKFLOW

5.1. Problem Statement

A good practice for chemical reactions is to first check the kinetic expressions in a simplified simulation case. This minimizes the risks of dimensional errors. OpenFOAM requires mass-based reaction rates and thermo-physical properties, where conventional chemical engineering approaches favour mole-based values. Table 1 lists the boundary conditions for the OpenFOAM model. The physical interpretation of the boundary conditions are here self-explanatory (e.g., “fixedValue”, or “zeroGradient”), except for the pressure, where we used a “fixedFluxPressure” boundary conditions, allowing to calculate the pressure gradient based on velocity. The results from both simulations (Figure 3) agree well with only a 0.14% difference in *i*-butane mass fraction at the outlet and 0.007 K for temperature.

With this confirmation, we demonstrate that we correctly implemented the kinetic model, then we built the conventional and irregular-shaped geometries with 5° wedges along the length of the reactor. This wedge shape is a simplification of the tubular shape of the reactors that accelerates the simulation by making use of azimuthal symmetry about the longitudinal axis of the tube.

When setting up the CFD simulation, we must select between laminar and turbulent flow regimes. Erroneously selecting a laminar regime for high Reynolds number flows or clearly non-laminar flows produces artefacts and oscillations in the residuals. Turbulence modelling is a field on its own[44] and a plethora of models exist, each with its strengths and weaknesses but not all solvers are compatible. For the example case, we selected the Reynolds averaged Navier–Stokes (RANS) approach.[45] RANS models comprise an equation for turbulence intensity and another for specific dissipation rate (ω), or the dissipation rate (ϵ). Here, we selected the $k - \omega$ shear stress

transport (SST) model, which is a blended turbulence model with higher accuracy than the $k - \epsilon$ model or the $k - \omega$ model from which it is derived.[46] It is especially suited for flows that detach from walls, which is the case in the notched reactor. The simplified forms of the transport equation are given in Equation (15). The terms $C_{\phi_{\text{turb}}}$, $\epsilon_{\phi_{\text{turb}}}$, and $D_{\phi_{\text{turb}}}$ represent the creation, dissipation, and diffusion of the turbulence parameters, while k is the turbulent kinetic energy and the ω is the turbulence specific dissipation rate.[47] From these two values, a turbulent viscosity ν_t is calculated. The formula to evaluate the turbulent viscosity itself depends on a blending function and on the distance to the wall and the vorticity, Ω (Equation (16)), where its value is given by k/ω , except in an adverse pressure gradient where the dissipation of turbulent kinetic energy is greater than its dissipation rate. For more information, readers are referred to Menter’s original paper on the $k - \omega$ SST turbulence model.[46]

$$\frac{\partial \rho \phi_{\text{turb}}}{\partial t} + \nabla \cdot (\rho \cdot \mathbf{v} \phi_{\text{turb}}) = C_{\phi_{\text{turb}}} - \rho \epsilon_{\phi_{\text{turb}}} + D_{\phi_{\text{turb}}} \quad (15)$$

$$\nu_t = \frac{a_1 k}{\max(a_1 \omega; \Omega F_2)} \quad (16)$$

5.2. Geometry and Meshing

Once the problem has been defined, we establish the geometry and create the mesh. Meshing is one of the most time-consuming steps in a project. A high-quality mesh improves stability and accuracy. Meshing discretizes the design space into cells connected via cell faces to neighbouring cells. The FVM is constructed in such a way that each cell should have 1 neighbouring cell per cell face (except for the cells near the boundaries).[24]

Structured meshes have a logical indexing structure and each cell has an index (i , j , and z) and its next neighbour will have ($i+1$, $y+1$, and $z+1$). This makes numerical interpretation of the cell indices easy. For unstructured meshes, two adjacent cells do not necessarily have the same index, and this numerical interpretation has to be taken into account in the solver. In practice, the structured meshes tend to consist of the same cell type (in most cases hexahedral cells), unstructured meshes can consist of a wide collection of cell types with different polyhedral shapes. OpenFOAM has integrated automated meshing codes for both structured (blockMesh)[48] and unstructured meshes (snappyHexMesh).[48] SALOME,[49] Pointwise, and ANSYS - Fluent have open-source and commercial software packages to generate meshes.

5.2.1. Mesh Quality

The mesh is a collection of the cells in the design space. However, not all meshes are equally suited for CFD, and before the start of the simulation, the quality of the mesh, which depends on the quality of the individual cells, should be checked. Much like the strength of a chain depends on the weakest link, the quality of a mesh is defined by the cell with the lowest quality. The four metrics for mesh quality are: aspect ratio, skewness, orthogonality, and smoothness.[50]

5.2.2. Example Meshes

Three distinct meshes were created for the example cases (Figure 7):

- 1 A coarse 1D structured mesh with an inlet and outlet boundary condition to confirm the chemical rate equations was correctly implemented. All other faces are defined using the OpenFOAM-specific type boundary *empty*, representing dimensions of the flow where no solution is required (Figure 7A).[48]
- 2 A tubular reactor mesh is constructed as a 5° wedge of a tubular reactor with a 0.1 m and a 0.5 m diameter (Figure 7B). The simplification is due to the azimuthal symmetry of the reactor.[51]

This mesh has three additional sets of boundary conditions compared to the 1D case: a *wall* type boundary on the outer wall (the tube's inner wall), a *symmetry* type (axis of the reactor), as well as a *wedge* boundary (planes along the axis). The *wedge* boundary conditions result from the rotational symmetry of the domain (but should not be used for rotating flows, where periodic boundary conditions are used).

- 3 The mesh represents the notched reactor and is designed to ensure that the total volume of the reactor was the same as the conventional tubular reactor (Figure 7C). This shape illustrates that CFD simulations apply to variable geometries for which a PFR model is inapplicable.[52]

All of these meshes are structured meshes created with the blockMesh utility.

5.3. Tubular Reactor in OpenFOAM

The boundary conditions to simulate the tubular reactor with FVM-CFD consists of *fixed-value* and *no slip* conditions *turbulent intensity kinetic energy inlet* and *turbulent mixing length frequency inlet* (Table 2). The latter

Table 1: Boundary conditions for the 1D model provided in the context of their implementation in OpenFOAM.

	U	p	T	species
inlet	Fixed-value	Fixed-flux- P	Fixed-value	Fixed-value
outlet	Zero-gradient	Fixed-value	Zero-gradient	Zero-gradient
defaultFaces	empty	empty	empty	empty

two relate to turbulence model-specific boundary conditions and only require as entries the intensity of turbulence (typically between 2 and 5%) and the mixing length (0.07 times the hydraulic diameter), respectively. The turbulence intensity (equal to the squared root of the sum of squared of the velocity fluctuations) is calculated from:

$$k = 1.5(T_i U_0)^2 \quad (17)$$

The numerical value of the specific rate of kinetic turbulent energy dissipation from the calculated turbulent kinetic energy using:

$$\omega = \frac{k^{1/2}}{L} \quad (18)$$

Other turbulence model-specific boundary conditions are given in the case files available on request.

- Zero-gradient: The normal gradient of the selected variables is fixed at zero so the values at the boundaries are calculated.
- Inlet–outlet: A mixed boundary condition to prevent fluid re-entering through the face by using the *zero gradient* when fluid flows out of the domain and a *fixed value* (here, set to 0), when the fluid flows into the domain.
- Wall functions (applied here to both k and ω): Empirically derived functions to simulate the flow near the wall. No input from the user is required.

Table 2: Boundary conditions for the wedge geometry simulation in the context of their implementation in OpenFOAM

	U m/s	p Pa	T K	species $\text{g} \cdot \text{g}^{-1}$	k $\text{m}^2 \cdot \text{s}^{-2}$	ω s^{-1}
inlet	fixed (0, 0, 26)	fixed-flux- P	fixed: 420	fixed: 0.88 (n - butane)	turbulent-intensity- kinetic-energy-inlet	turbulent-mixing- length-frequency-inlet
outlet	zero- gradient	fixed: 2×10^6	zero- gradient	zero- gradient	inlet-outlet	inlet-outlet
outerwall	no slip	zero- gradient	zero- gradient	zero- gradient	kqR wallfunction	omega wall function

We ran the simulation with second-order accuracy in space from the beginning since the mesh was simple as is the example problem. The PISO algorithm, with three inner correctors (for the pressure correction) and zero non-orthogonal correctors (due to the highly orthogonal mesh), was used to reach the solution of the simulation (Figure 8). The simulation results agree with the PFR model (MATLAB solution), which was expected due to the tubular geometry and high Reynolds number, that for increasing Reynolds number, turbulent flow velocity profiles approach a plug flow profile more. The main differences can be attributed to the no-slip conditions at the wall. This example shows that the CFD simulation accurately predicts n -butane conversion to i -butane in a conventional reactor case, and so we assume that the thermophysical and chemical properties of the simulation case are correctly implemented. With this assumption, we can solve the cases for the notched reactor with a reasonable certainty.

5.4. Notched Reactor

The simulation case was copied from the conventional tubular reactor geometry, except for the mesh and a decrease in time step to account for the higher velocities around the notch. The decrease in time step was necessary to respect the N_{CFL} condition (Equation (10)).

The notched reactor simulation (Figure 9) shows that the flow pattern changes drastically near the volume expansion–contraction region, where pressure builds up near the narrow part, velocity increases, so less n -butane

is converted to *i*-butane due to a reduced residence time. As a result, the mass fraction of *n*-butane and *i*-butane clearly deviates from the PFR configuration (Figure 3).

As mentioned in the theory section, every CFD simulation should be assessed for its convergence behaviour. The convergence performance figures are in the supporting information. We assessed the convergence based on the residuals, mass conservation, and steady state behaviour. For the notched reactor, the convergence behaviour was acceptable: the residuals decreased and the deviation in the mass conservation was small (0.2%), and after 0.06 s, the simulation reaches a steady state solution, which is well below the final time step of 0.2 s.

However, a simulation that is converged does not necessarily mean it is accurate. In order to check the accuracy of a solution, the errors induced from the mesh quality have to be evaluated. The method to quantify the uncertainty induced by a mesh was standardized by Roache in 1994.[53] This step in the CFD simulation workflow is called grid refinement or mesh independence study.

5.4.1. Grid Refinement Study

The grid refinement study applies a grid convergence index (GCI) to quantify the grid independence of the solution. The GCI applies the Richardson extrapolation to estimate the refinement error when comparing simulation solutions with distinct grid spacing. It was originally designed to estimate the results from any mesh refinement when a grid is doubled ($r = 2$). To quantify the GCI, we simulate at least two but preferably three meshes, with an increasing number of cells. Then, a given quantity of interest is extracted from those simulations (temperature, pressure, etc.), and the GCI is computed. To illustrate the procedure, we extracted the temperature in the region after the expansion–contraction, where the flow is developed. The overall grid convergence process is as follows:

- Create three meshes with a set refinement ratio, that is, the number of points in a direction in the mesh. In the example, the refinement ratio was 2, which means that for the 2D mesh, there was a coarse mesh with 1000 cells, an intermediate mesh with 4000 cells and a finer mesh with 16 000 cells (Figure 11).
- Simulate the system with the three meshes. To reduce simulation time

for the finer meshes, we apply the results of a coarser mesh as initial conditions, since these inputs are closer to the expected value than best guesses. In OpenFOAM, the *mapFields* function completes this operation.[48]

- We extract temperature from the same location from each of the three meshes to compute the GCI: 0.35 m in the axial direction and 0.0075 m in the radial direction (Table 3). From these data, the real order of convergence can be computed, which is different from the theoretical order of convergence ($r_f = 2$ in this case) due to stretching and non-linearity in the solution. The calculated order of convergence is obtained by using Equation (19). The simulated temperature profiles along the axis of the notched reactor for the three resulting meshes are presented in Figure 12.

$$p = \frac{\ln\left(\frac{\Delta T_{\text{coarse}} - \Delta T_{\text{med}}}{\Delta T_{\text{med}} - \Delta T_{\text{fine}}}\right)}{\ln(r_f)} = \frac{\ln\left(\frac{45.9 - 55.4}{55.4 - 59.5}\right)}{\ln(2)} = 1.19 \quad (19)$$

Table 3: Grid independence study on temperature

Grid	N_{cells}	Normalized Grid Spacing	ΔT (K)
Coarse	1000	4	45.9
Medium	4000	2	55.4
Fine	16 000	1	59.5

- The next step is to calculate the temperature at a grid with zero spacing, that is, a theoretically infinitely fine grid, based on a Richardson extrapolation.

$$\begin{aligned} T_{\infty} &= \Delta T_{\text{fine}} + \frac{\Delta T_{\text{fine}} - \Delta T_{\text{med}}}{r_f^p - 1} \\ &= 59.5 + \frac{59.5 - 55.4}{2^{1.19} - 1} = 62.7 \end{aligned} \quad (20)$$

- After the extrapolation step, the GCI is calculated between two grids: GCI_{12} considers the change from grid 1 to 2 and GCI_{23} from grid 2 to

3. We apply a safety factor, $F_s = 1.25$, for a 3 grid independence study and a higher value for a 2 grid study.[54]

$$GCI_{12} = \frac{F_s \cdot |\delta_{12}|}{r_f^p - 1} = \frac{1.25 \cdot \left| \frac{59.5 - 55.4}{59.5} \right|}{2^{1.19} - 1} = 0.07 \quad (21)$$

$$GCI_{23} = \frac{F_s \cdot |\delta_{23}|}{r_f^p - 1} = \frac{1.25 \cdot \left| \frac{55.4 - 45.9}{55.4} \right|}{2^{1.19} - 1} = 0.16 \quad (22)$$

- In this step, we verify if Equation (23) is approximately equal to one to confirm the solutions are in the asymptotic range:

$$r_f^p \frac{GCI_{12}}{GCI_{23}} = 2^{1.19} \frac{0.07}{0.16} = 1.07 \cong 1 \quad (23)$$

- Finally, we can state that the temperature at the probe point in the mesh equals $420.0 \text{ K} + 62.7 \text{ K} = 482.7 \text{ K}$ and the error with respect to the temperature rise (63°C) ($\Delta T_\infty - \Delta T_{in}$) is 7%. [55] The fine mesh and intermediate mesh temperature profiles are similar while the coarse mesh is noticeably cooler (Figure 13). We conclude that the fine mesh is sufficient to extract quantitative data from the simulation, as the simulated temperature is within the 7% error of the Richardson extrapolation. We also want to highlight again that eventhough the simulation on the coarse mesh converged nicely, the errors induced from the mesh were quite high.

As a final note, the GCI method for grid convergence identifies numerical errors on a grid. We only illustrated the procedure using a simple example at one temperature. However, complete datasets can be used to check the GCI, for instance, using the data extracted over a line in the mesh.[2] After performing a grid refinement study, the CFD engineer should make a choice on accuracy versus time. The finer the grid, the more accurate the solution tends to be, but, on the other hand, the computational time for the simulation will increase.[2] In this example, we used a single Intel Core i7 (8th gen) processor, and the wall time to obtain the results with the coarse mesh took 392s, 1296s for the intermediate mesh, and 6772s for the fine mesh, for only 0.2s of simulated time. The reported clock times are very short (simple homogeneous reaction) and highlights the effect of simulation using 5° wedge geometries. For more complex cases (e.g., multiphase turbulent

reacting flows), the clock time can exceed weeks, even considering parallel computing. It is also worth checking which areas of the mesh need additional refinement to reach grid convergence, as not all areas of a geometry tend towards convergence at the same pace.[56] Sometimes, the GCI is not reported, and authors use a less rigorous approach. For example, grids with increasing refinement are used until there is no longer a change in the value of interest.[57, 58] This is a valid, qualitative approach to reduce grid-related numerical errors, and offers an alternative to the quantitative GCI procedure.

6. VALIDATION

The mesh independence study gave the most accurate result. However, to take this from a simulation to reality, a prototype should be built, and the results from the simulations should be experimentally validated, in this case, for the temperature and species mass fractions. This is a validation step required in a complete CFD study,[59, 60] which falls out of the scope of this work. CFD simulations can provide a reasonable first estimate of what can be expected for chemical engineering applications. However, it is not the holy grail that allows for simulation without experimental work. Experimental validation is mandatory for simulations heavily relying on closure or multiphase models (e.g., multiphase turbulent simulations elaborated from the kinetic theory of granular flows).

7. APPLICATIONS

The acronym CFD was seldom mentioned in the 1970s and even into the 1980s articles refer to computational fluid dynamics most rather than CFD. The early applications were for internal engine combustion,[61] propulsion in aerospace,[62] grid generation,[63] and 3D turbulent flow.[64] From the modest number of contributions in the 1980s, the field has grown and produced over 7500 articles in 2020. The open-access program VOSViewer generates maps of bibliometric data indexed by Scopus and WoS. The program lumped the keywords of the articles in WoS into five categories, and for each, it assigns a colour to identify the related topics. The category in red deals mostly with modelling (multi-phase flow, CFD-DEM, dynamics, and validation). The green cluster has the second most number of keywords centred on mass transfer, combustion, and emissions). Design, optimization, and turbulence dominate the blue cluster with the third most number of

keywords. Heat transfer is the prominent keyword of the yellow cluster and finally LES, ventilation, and the environment are the main keywords for the magenta cluster.

Since the early 1980s, mechanical engineers have contributed 25 % of the scientific articles followed by energy & fuels (15 %) and chemical engineering (15 %). In 2020, these three categories remained the top but energy & fuels had the most articles (1477), while the mechanical engineering category only had 100 more articles than chemical engineering (1342 vs. 1242). The other top science categories were thermodynamics (1032), mechanics (934), and civil engineering (589).

Since 2020, *Safety Science* published the most cited article entitled “Modelling aerosol transport and virus exposure with numerical simulations in relation to SARS-CoV-2 transmission by inhalation indoors” (123 citations).[65] The second most cited article (99 citations) was in *Thermal Science* entitled “Investigation of Air Conditioning Temperature Variation by Modifying The Structure of Passenger Car Using Computational Fluid Dynamics”.[66] The subject of the third most cited article (88 cites) was about vortex cyclones and submicron particles entitled “Charging effect on separation performance of outer vortex type cyclone for submicron particles at different operating parameters” in *Particulate Science And Technology*.[67]

The *Canadian Journal of Chemical Engineering* published 17 articles on CFD in 2020, and the three top cited were: “Effect of substrate geometry and flow condition on the turbulence generation after a monolith,”[68] “Numerical simulation of the mass transfer process of CO₂ absorption by different solutions in a microchannel,”[69] and “A CFD-DEM approach to study the breakup of fractal agglomerates in an internal mixer.”[70]

7.1. Limitations

CFD studies are only as accurate as the models upon which they are built—turbulence models unless direct numerical simulations are performed, multiphase simulations relying either on the kinetic theory of granular flow or on the DEM approach. Complex domains coupled to complex physics require excessive CPU time but this has become less of a concern with the availability of high performance computing systems. Even so, the example case with 16 000 cells and 0.2 s simulation time took close to 2 h on the single Intel Core i7 (8th gen) processor. An important question is how long before you decide that the system has reached steady state—when the velocity, temperature, and species concentrations change less than a set threshold with a successive

iteration. For example, a change in average species mole fraction in the gas-phase and on a catalyst surface of less than $0.01\% \cdot \text{s}^{-1}$. [71] For multiphase systems that have natural fluctuations, like fluidized beds, it is based on a time-average steady state. [72]

Apart from CFD's reliance on external programs and models, CFD also depends on the quality of the mesh. Poor quality cells introduce numerical problems during the simulation, but making a mesh for complex geometries with only perfect hexahedral cells, as were used in the example case, is rarely possible.

FVM-CFD has not advanced sufficiently to replace experiments; rather, it provides initial estimates to help design prototypes instead. In many cases, an accuracy of on the order of 10% is considered good. Other limitations include the software: not all FVM - CFD programs have suitable solvers. Users of open-software packages can add their own code, but this is often very time consuming. Another advantage of open-source codes is that they are free on high-performance computer clusters, whereas commercial CFD codes cost, and this becomes prohibitively expensive for problems requiring many licenses. The commercial software models and algorithms are robust, and most of the errors come from user-added inputs. Integrating custom code into process simulators is cumbersome and takes longer than writing a program from scratch.

Errors compound as the simulations proceeds from one step of the process to the next. Simulators have no error assessment capability. Finally, the scale at which the simulation is performed is crucial in any FVM-CFD study but is especially important for chemical engineering applications, for example, the mass and heat transfer between a catalyst particle and the surrounding reacting gas. [73] These effects are local and thus need small cells to simulate them accurately. For small-scale reactors, this can be achieved, but for larger reactors, the mean cell size would have to stay the same so the absolute number of cells would be a lot higher. This would result in much longer computational times. The scalability of a CFD code is of primary importance: an increase in simulation cores should be accompanied by a linear increase in computational time. This is a main drawback of current CFD-DEM codes where the CFD and DEM parts are treated sequentially. The scalability of a CFD code strongly depends on the complexity of the simulations (large Eddy simulation vs. Reynolds-averaged, multiphase turbulence reacting flows, etc.).

8. CONCLUSIONS

From 1997 to 2017, the number of CFD articles indexed by WoS grew from 400 to 7000. However, the productivity has stagnated somewhat at below 8000 in the last 5 years. Here, we provided new CFD users with a framework to get started and illustrated the typical workflow with the adiabatic isomerization reaction of *n*-butane to *i*-butane. The CFD implementation in OpenFOAM was first validated with a PFR model resolved using MATLAB to check the accuracy of the CFD code. We then simulated a tubular reactor using CFD and pushed the concept of asymmetric reactors. The mesh independence study confirmed that 16 000 meshes were sufficient to reach within 7% of the solution estimated with the Richardson extrapolation. The cost of commercial packages can be prohibitive while working on high-performance computer clusters. For this reason, researchers will continue to program with open-access programs like OpenFOAM. However, the downside of open-access is technical support and new releases that improve performance but have issues: the latest OpenFOAM release includes 750 commits and 200+ resolved issues.[74]

Nomenclature

a	acceleration ($\text{m} \cdot \text{s}^{-2}$)
a_1	$k - \omega$ SST model parameter
\mathbf{A}	diagonal matrix of implicit coefficients
\mathbf{B}	matrix of explicit coefficients
C_i	concentration of species i ($\text{mol} \cdot \text{m}^{-3}$)
$C_{p,i}$	specific heat capacity for species i ($\text{J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$)
$C_{\phi_{\text{turb}}}$	creation of turbulence transport variable
D_{eff}	effective diffusivity ($\text{m}^2 \cdot \text{s}^{-1}$)
$D_{\phi_{\text{turb}}}$	diffusion of turbulence transport variable
\mathcal{F}	force (N)
F	flux
F_s	safety factor
F_1	blending function
g	gravitational acceleration ($\text{m} \cdot \text{s}^{-2}$)
GCI	grid convergence index
\mathbf{H}	matrix of the off-diagonal coefficients

H	specific enthalpy ($\text{J} \cdot \text{mol}^{-1}$)
\mathbf{I}	matrix of the unit vectors
k	turbulence kinetic energy ($\text{m}^2 \cdot \text{s}^{-2}$)
k	first-order kinetic constant—Equation (11) (s^{-1})
K_C	equilibrium constant defined in the liquid phase
\mathcal{K}	kinetic energy
m	mass (kg)
\mathbf{M}	matrix of implicit coefficients
N_{CFL}	Courant-Friedrichs-Lewy number
N_{Sc}	Schmidt number
\hat{n}	unit vector normal to a surface
p	apparent order of accuracy
\mathcal{P}	pressure (Pa)
\mathcal{Q}_ϕ	general source term for variable ϕ
$\mathcal{Q}_{\mathcal{R}}$	heat of reaction
r	reaction rate ($\text{mol} \cdot \text{m}^{-3} \cdot \text{s}^{-1}$)
r_f	refinement factor
R	universal gas constant ($\text{J} \cdot \text{K}^{-1} \cdot \text{mol}^{-1}$)
\mathcal{R}	net production rate ($\text{mol} \cdot \text{m}^{-3} \cdot \text{s}^{-1}$)
S	surface (m^2)
t	time (s)
T	temperature (K)
u_z	superficial velocity for the 1D model ($\text{m} \cdot \text{s}^{-1}$)
\mathbf{v}	velocity ($\text{m} \cdot \text{s}^{-1}$)
V	volume (m^3)
x	cell size (m)
X	chemical conversion
Y	mass fraction
z	axial coordinate (m)
Greek Letters	
α_{eff}	effective thermal diffusion coefficient ($\text{m}^2 \cdot \text{s}^{-1}$)
Γ	diffusion coefficient ($\text{m}^2 \cdot \text{s}^{-1}$)
$\epsilon_{\text{phi}_{\text{turb}}}$	dissipation of turbulence transport variable
ρ	density ($\text{kg} \cdot \text{m}^{-3}$)
μ	viscosity ($\text{kg} \cdot \text{m}^{-1} \cdot \text{s}^{-1}$)
ν	kinematic viscosity ($\text{m}^2 \cdot \text{s}^{-1}$)

ν_t	turbulent kinematic viscosity ($\text{m}^2 \cdot \text{s}^{-1}$)
ϕ	general transport property
ϕ_{turb}	general transport property for turbulence
θ_i	ratio of flowrates with respect to limiting reactant
ω	specific dissipation rate (s^{-1})
Ω	vorticity magnitude

References

- [1] N. Jurtz, M. Kraume, G. D. Wehinger, Advances in fixed-bed reactor modeling using particle-resolved computational fluid dynamics (cfd), *Reviews in Chemical Engineering* 35 (2) (2019) 139–190.
- [2] L. A. Vandewalle, A. Gonzalez-Quiroga, P. Perreault, K. M. Van Geem, G. B. Marin, Process intensification in a gas–solid vortex unit: Computational fluid dynamics model based analysis and design, *Industrial & Engineering Chemistry Research* 58 (28) (2019) 12751–12765.
- [3] L. A. Vandewalle, G. B. Marin, K. M. Van Geem, catchyfoam: Euler–euler cfd simulations of fluidized bed reactors with microkinetic modeling of gas-phase and catalytic surface chemistry, *Energy & Fuels* 35 (3) (2020) 2545–2561.
- [4] G. P. Smith, D. M. Golden, M. Frenklach, N. W. Moriarty, B. Eiteneer, M. Goldenberg, C. T. Bowman, R. K. Hanson, S. Song, W. C. Gardiner Jr., V. V. Lissianski, Z. Qin, Gri-mech 3.0, <http://combustion.berkeley.edu> (accessed: December 2021) (2021).
- [5] C. Choi, W. Zhang, K. Fukumoto, H. Machida, K. Norinaga, A review on detailed kinetic modeling and computational fluid dynamics of thermochemical processes of solid fuels, *Energy & Fuels* 35 (7) (2021) 5479–5494.
- [6] C. W. Gao, J. W. Allen, W. H. Green, R. H. West, Reaction mechanism generator: Automatic construction of chemical kinetic mechanisms, *Computer Physics Communications* 203 (2016) 212–225.
- [7] N. M. Vandewiele, K. M. Van Geem, M.-F. Reyniers, G. B. Marin, Genesys: Kinetic model construction using chemo-informatics, *Chemical Engineering Journal* 207–208 (2012) 526–538.

- [8] M. Bracconi, M. Maestri, A. Cuoci, In situ adaptive tabulation for the cfd simulation of heterogeneous reactors based on operator-splitting algorithm, *AIChE Journal* 63 (1) (2017) 95–104.
- [9] A. Cuoci, A. Frassoldati, T. Faravelli, E. Ranzi, Opensmoke++: An object-oriented framework for the numerical modeling of reactive systems with detailed kinetic mechanisms, *Computer Physics Communications* 192 (2015) 237–264.
- [10] T. Lu, C. K. Law, Toward accommodating realistic fuel chemistry in large-scale computations, *Progress in Energy and Combustion Science* 35 (2) (2009) 192–215.
- [11] E. Ranzi, A. Frassoldati, A. Stagni, M. Pelucchi, A. Cuoci, T. Faravelli, Reduced kinetic schemes of complex reaction systems: Fossil and biomass-derived transportation fuels, *International Journal of Chemical Kinetics* 46 (9) (2014) 512–542.
- [12] M. Bracconi, M. Maestri, Training set design for machine learning techniques applied to the approximation of computationally intensive first-principles kinetic models, *Chemical Engineering Journal* 400 (2020) 125469.
- [13] B. Blais, D. Vidal, F. Bertrand, G. S. Patience, J. Chaouki, Experimental methods in chemical engineering: Discrete element method—dem, *Canadian Journal of Chemical Engineering* 97 (7) (2019) 1964–1973.
- [14] A. Bérard, G. S. Patience, B. Blais, Experimental methods in chemical engineering: Unresolved cfd-dem, *Canadian Journal of Chemical Engineering* 98 (2) (2020) 424–440.
- [15] Q. Y. Li, X. Z. Wang, Population balance and cfd simulation of particle aggregation and growth in a continuous confined jet mixer for hydrothermal synthesis of nanocrystals, *Crystals* 11 (2) (2021) 144.
- [16] S. Rigopoulos, Population balance modelling of polydispersed particles in reactive flows, *Progress in Energy and Combustion Science* 36 (4) (2010) 412–443.

- [17] B. Liu, K. Papadikis, S. Gu, B. Fidalgo, P. Longhurst, Z. Li, A. Kolios, Cfd modelling of particle shrinkage in a fluidized bed for biomass fast pyrolysis with quadrature method of moment, *Fuel Processing Technology* 164 (2017) 51–68.
- [18] H. Li, A. Ndjamo, P. Sauriol, G. S. Patience, Optimization of lifepo4 wet media milling and regressive population balance modeling, *Applied Powder Technology* 28 (3) (2017) 1000–1007.
- [19] M. Shiea, A. Buffo, M. Vanni, D. Marchisio, Population balance equations coupled with computational fluid dynamics, *Annual Review of Chemical and Biomolecular Engineering* 11 (2020) 339–366. doi:10.1146/annurev-chembioeng-092319-075814.
- [20] A. Mühlbauer, M. W. Hlawitschka, H.-J. Bart, Models for the numerical simulation of bubble columns: A review, *Chemie Ingenieur Technik* 91 (12) (2019) 1747–1765.
- [21] G. S. Patience, Experimental methods in chemical engineering: Preface, *Canadian Journal of Chemical Engineering* 96 (2018) 2312–2316.
- [22] E. Robertson, V. Choudhury, S. Bhushan, D. K. Walters, Validation of openfoam numerical methods and turbulence models for incompressible bluff body flows, *Computers & Fluids* 123 (2015) 122–145.
- [23] C. Sadino-Riquelme, R. E. Hayes, D. Jeison, A. Donoso-Bravo, Computational fluid dynamic (cfd) modelling in anaerobic digestion: General application and recent advances, *Critical Reviews in Environmental Science and Technology* 48 (1) (2018) 39–76. doi:10.1080/10643389.2018.1440853.
- [24] H. Versteeg, W. Malalasekera, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, Pearson Education Limited, London, 2007.
- [25] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, *Computers in Physics* 12 (6) (1998) 620–631.

- [26] R. O. Fox, Computational Models for Turbulent Reacting Flows, Cambridge Series in Chemical Engineering, Cambridge University Press, Cambridge, 2003.
- [27] M. Moukalled, L. Mangani, M. Darwish, The Finite Volume Method in Computational Fluid Dynamics An Advanced Introduction with OpenFOAM and Matlab, 1st Edition, Fluid Mechanics and Its Applications, Springer International Publishing, Cham, Switserland, 2016.
- [28] M. G. Edwards, C. F. Rogers, Finite volume discretization with imposed flux continuity for the general tensor pressure equation, Computational Geosciences 2 (4) (1998) 259–290.
- [29] R. Eymard, T. Gallouët, R. Herbin, Finite volume methods, in: P. G. Ciarlet, J. L. Lions (Eds.), Handbook of Numerical Analysis, Vol. 7, Elsevier, Amsterdam, 2000, pp. 713–1018.
- [30] T. Holzmann, Mathematics, Numerics, Derivations and OpenFOAM, Vol. 7, Holmzann CFD, Augsburg, 2019.
- [31] J. Y. Murthy, S. R. Mathur, Numerical methods in heat, mass, and momentum transfer, https://engineering.purdue.edu/scalo/menu/teaching/me608/ME608.Notes_Murthy.pdf (accessed: December 2021) (2002).
- [32] J. H. Ferziger, M. Peric, Computational Methods for Fluid Dynamics, 3rd Edition, Springer, Berlin, 2001. doi:<https://doi.org/10.1007/978-3-642-56026-2>.
- [33] M. K. Saqr, Practical introduction to cfd, harvard dataverse, <https://doi.org/10.7910/DVN/D84IZM> (accessed: September 2021) (2017).
- [34] I. Demirdžić, On the discretization of the diffusion term in finite-volume continuum mechanics, Numerical Heat Transfer Part B: Fundamentals 68 (1) (2015) 1–10.
- [35] A. Wimshurst, Fluid mechanics 101 courses, <https://www.fluidmechanics101.com/> (accessed: December 2021) (2021).

- [36] Ansys Fluent, Ansys fluent fluid simulation software, www.ansys.com/products/fluids/ansys-fluent (accessed: December 2021) (2016).
- [37] Siemens, Simcenter star-ccm+, www.plm.automation.siemens.com/global/en/products/simcenter/STAR-CCM.html (accessed: December 2021) (2021).
- [38] S. V. Patankar, D. B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, *International Journal of Heat and Mass Transfer* 15 (10) (1972) 1787–1806.
- [39] R. I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, *Journal of Computational Physics* 62 (1) (1986) 40–65.
- [40] R. Courant, K. Friedrichs, H. Lewy, On the partial difference equations of mathematical physics, *IBM Journal of Research and Development* 11 (2) (1967) 215–234.
- [41] G. S. Patience, J. Chaouki, Gas phase hydrodynamics in the riser of a circulating fluidized bed, *Chemical Engineering Science* 48 (18) (2020) 3195–3205.
- [42] J. Tu, G.-H. Yeoh, C. Liu, *Computational Fluid Dynamics Chapter 7. Practical Guidelines for CFD Simulation and Analysis*, 3rd Edition, Elsevier, Amsterdam, 2018, pp. 255–290.
- [43] H. S. Fogler, *Elements of Chemical Reaction Engineering*, 4th Edition, Prentice Hall, Needham Heights, MA, 2006.
- [44] D. C. Wilcox, *Turbulence modelling for CFD*, DCW Industries, La Cañada, 1993.
- [45] S. N. A. Yusuf, Y. Asako, N. A. Che Sidik, S. B. Mohamed, W. M. A. Aziz Japar, A short review on rans turbulence models, *CFD Letters* 12 (11) (2020) 83–96.
- [46] F. R. Menter, Two-equation eddy-viscosity turbulence models for engineering applications, *AIAA Journal* 32 (8) (1994) 1598–1605.

- [47] L. Konozy, The k - ω Shear-Stress Transport (SST) Turbulence Model, Springer International Publishing, Cham, Switzerland, 2019, pp. 57–66.
- [48] The OpenFOAM Foundation, C++ source code guide, <https://cpp.openfoam.org/v8/> (accessed: December 2021) (2020).
- [49] A. Ribes, C. Caremoli, Salome platform component model for numerical simulation, in: 31st Annual International Computer Software and Applications Conference (COMPSAC 2007), Vol. 2, 2007, pp. 553–564. doi:10.1109/COMPSAC.2007.185.
- [50] P. M. Knupp, Remarks on mesh quality, <https://cfwebprod.sandia.gov/cfdocs/CompResearch/docs/reno07paper.pdf> (accessed: June 2021) (2007).
- [51] L. Gutierrez Marcantoni, J. Tamagno, S. Elaskar, Rans simulation of turbulent diffusive combustion using openfoam, *Journal of Applied Fluid Mechanics* 9 (2) (2016) 669–682.
- [52] D. T. Allen, *The engineering of chemical reactions*. by lanny d. schmidt, oxford university press, new york, 1998, *AIChE Journal* 45 (8) (1999) 1836–1836.
- [53] P. J. Roache, Perspective: A method for uniform reporting of grid refinement studies, *Journal of Fluids Engineering* 116 (3) (1994) 405–413.
- [54] T. Xing, F. Stern, Factors of safety for richardson extrapolation, *Journal of Fluids Engineering* 132 (6) (2010).
- [55] J. W. Slater, Examining spatial (grid) convergence, www.grc.nasa.gov/www/wind/valid/tutorial/spatconv.html (accessed: November 2021) (2021).
- [56] K. Inthavong, A. Chetty, Y. Shang, J. Tu, Examining mesh independence for flow dynamics in the human nasal cavity, *Comput Biol Med* 102 (2018) 40–50.
- [57] D. Misiulia, S. Antonyuk, A. G. Andersson, T. S. Lundström, High-efficiency industrial cyclone separator: A cfd study, *Powder Technology* 364 (2020) 943–953.

- [58] S. Wang, K. Luo, C. Hu, L. Sun, J. Fan, Impact of operating parameters on biomass gasification in a fluidized bed reactor: An eulerian-lagrangian approach, *Powder Technology* 333 (2018) 304–316.
- [59] P. E. Rodríguez-Ocampo, M. Ring, J. V. Hernández-Fontes, J. C. Alcérreca-Huerta, E. Mendoza, R. Silva, Cfd simulations of multiphase flows: Interaction of miscible liquids with different temperatures, *Water* 12 (9) (2020).
- [60] A. Laitinen, K. Saari, K. Kukko, P. Peltonen, E. Laurila, J. Partanen, V. Vuorinen, A computational fluid dynamics study by conjugate heat transfer in openfoam: A liquid cooling concept for high power electronics, *International Journal of Heat and Fluid Flow* 85 (2020) 108654.
- [61] M. D. Griffin, E. Jones, J. D. Anderson, Computational fluid dynamic technique valid at the center-line for non-axisymmetric problems in cylindrical coordinates, *Journal of Computational Physics* 30 (3) (1979) 352–360.
- [62] G. C. Paynter, Cfd technology for propulsion installation design—forecast for the 80s, *Journal of Engineering for Power* 104 (4) (1982) 832–837.
- [63] J. F. Thompson, Grid generation techniques in computational fluid-dynamics, *AAIA Journal* 22 (11) (1984) 1505–1523.
- [64] A. A. Busnaina, D. G. Lilley, A basic code for the prediction of transient three-dimensional turbulent flowfields, *Chemical Engineering Communications* 26 (1-3) (1984) 89–104.
- [65] V. Vuorinen, M. Aarnio, M. Alava, V. Alopaeus, N. Atanasova, M. Auvinen, N. Balasubramanian, H. Bordbar, P. Erasto, R. Grande, N. Hayward, A. Hellsten, S. Hostikka, J. Hokkanen, O. Kaario, A. Karvinen, I. Kivisto, M. Korhonen, R. Kosonen, J. Kuusela, S. Lestinen, E. Laurila, H. J. Nieminen, P. Peltonen, J. Pokki, A. Puisto, P. Raback, H. Salmenjoki, T. Sironen, M. Osterberg, Modelling aerosol transport and virus exposure with numerical simulations in relation to sars-cov-2 transmission by inhalation indoors, *Safety Science* 130 (2020). doi:10.1016/j.ssci.2020.104866.

- [66] K. Haribabu, S. Muthukrishnan, S. Thanikodi, G. A. Arockiaraj, V. Venkatrama, Investigation of air conditioning temperature variation by modifying the structure of passenger car using computational fluid dynamics, *Thermal Science* 24 (1) (2020) 495–498. doi:10.2298/TSCI190409397K.
- [67] J. Zhang, Z. Zha, P. Che, H. Ding, W. Pan, Charging effect on separation performance of outer vortex type cyclone for submicron particles at different operating parameters, *Particulate Science And Technology* 38 (5) (2020) 596–604. doi:10.1080/02726351.2019.1571542.
- [68] I. Cornejo, P. Nikrityuk, R. E. Hayes, Effect of substrate geometry and flow condition on the turbulence generation after a monolith, *Canadian Journal of Chemical Engineering* 98 (4) (2020) 947–956. doi:10.1002/cjce.23687.
- [69] R. Dong, D. Chu, Q. Sun, Z. Jin, Numerical simulation of the mass transfer process of co2 absorption by different solutions in a microchannel, *Canadian Journal of Chemical Engineering* 98 (12) (2020) 2648–2664. doi:10.1002/cjce.23781.
- [70] G. Frungieri, G. Boccardo, A. Buffo, D. Marchisio, H. A. Karimi-Varzaneh, M. Vanni, A cfd-dem approach to study the breakup of fractal agglomerates in an internal mixer, *Canadian Journal of Chemical Engineering* 98 (9) (2020) 1880–1892. doi:10.1002/cjce.23773.
- [71] R. W. Ashcraft, J. Kovacevic, G. J. Heynderickx, G. B. Marin, Assessment of a gas–solid vortex reactor for so₂/nox adsorption from flue gas, *Industrial & Engineering Chemistry Research* 52 (2) (2013) 861–875. doi:10.1021/ie300399w.
- [72] R. W. Ashcraft, G. J. Heynderickx, G. B. Marin, Modeling fast biomass pyrolysis in a gas–solid vortex reactor, *Chemical Engineering Journal* 207–208 (2012) 195–208. doi:https://doi.org/10.1016/j.cej.2012.06.048.
- [73] B. Partopour, A. G. Dixon, n-butane partial oxidation in a fixed bed: A resolved particle computational fluid dynamics simulation, *Canadian Journal of Chemical Engineering* 96 (9) (2018) 1946–1956. doi:doi.org/10.1002/cjce.23130.

[74] OpenFOAM, OpenFOAM 9 released, <https://openfoam.org/release/9/> (accessed: December 2021) (2021).

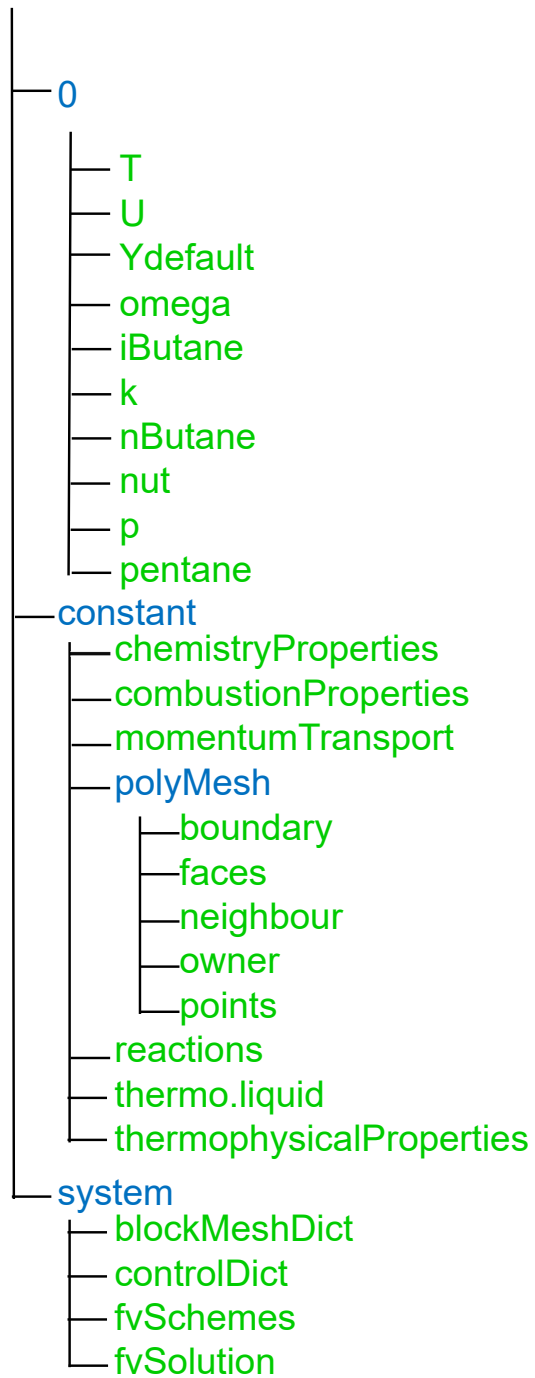


Figure 2: Structure for the OpenFOAM case of the PFR model

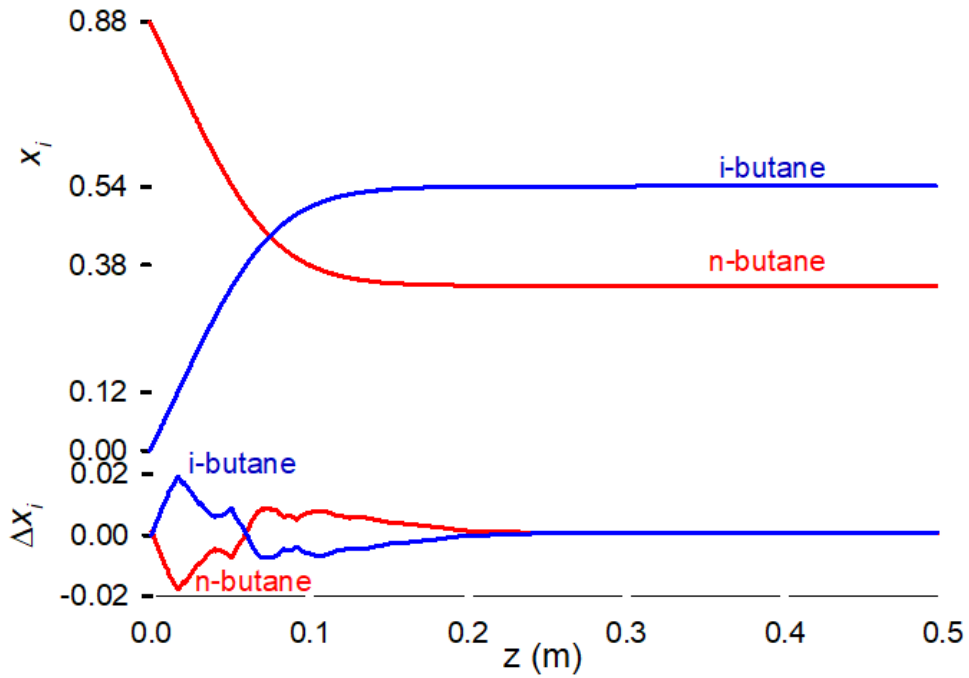


Figure 3: Top plot: Mass fraction of *n*-butane and *i*-butane along the reactor axis. Bottom plot: Difference between MATLAB R2019a and 1D flow in OpenFOAM. The maximum Δx_i reaches a mass fraction of $0.02 \text{ g} \cdot \text{g}^{-1}$ at the entrance where the x_i varies most

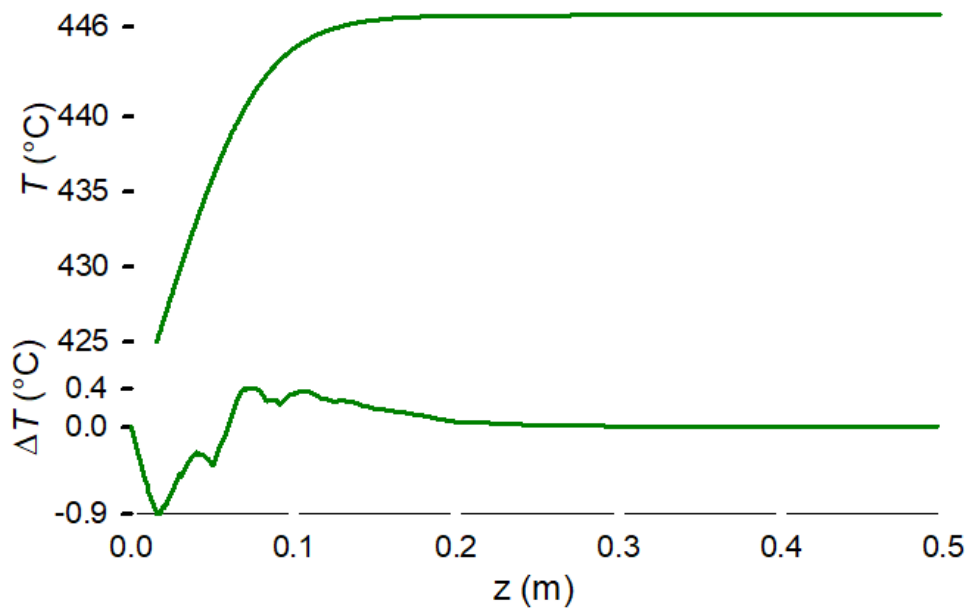
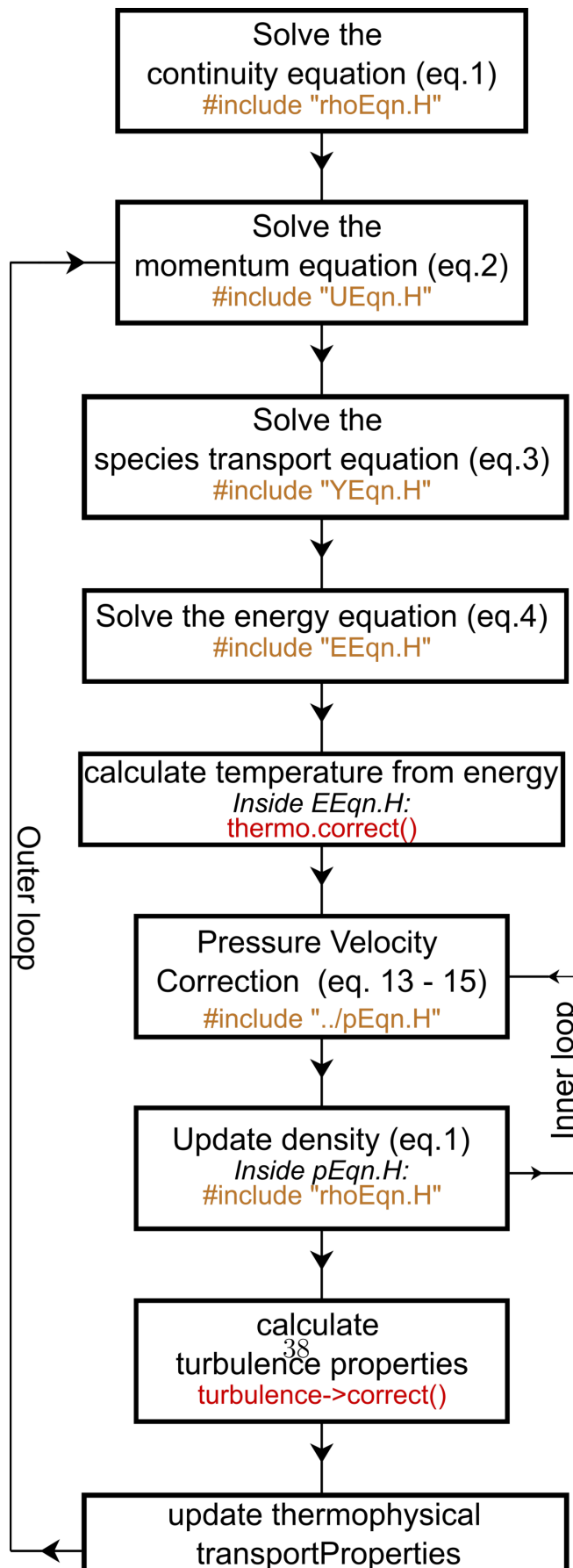


Figure 4: Top plot: temperature profile along the PFR horizontal axis. Bottom plot: Difference between MATLAB R2019a and 1D flow in OpenFOAM. The maximum ΔT reaches 0.9°C at the entrance where T varies most



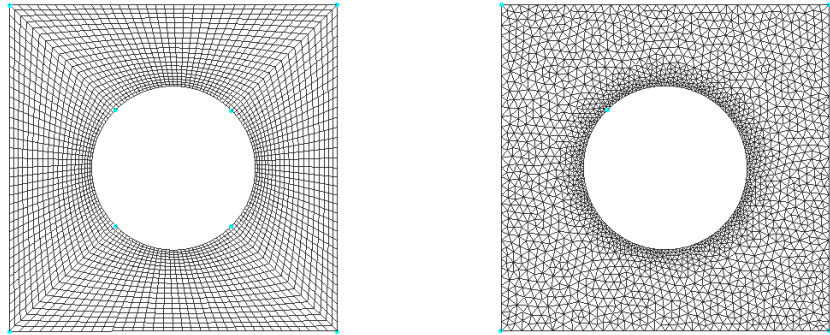


Figure 6: Structured (left) and unstructured (right) meshes. The structured mesh shows a logical ordering of cells where the unstructured mesh does not

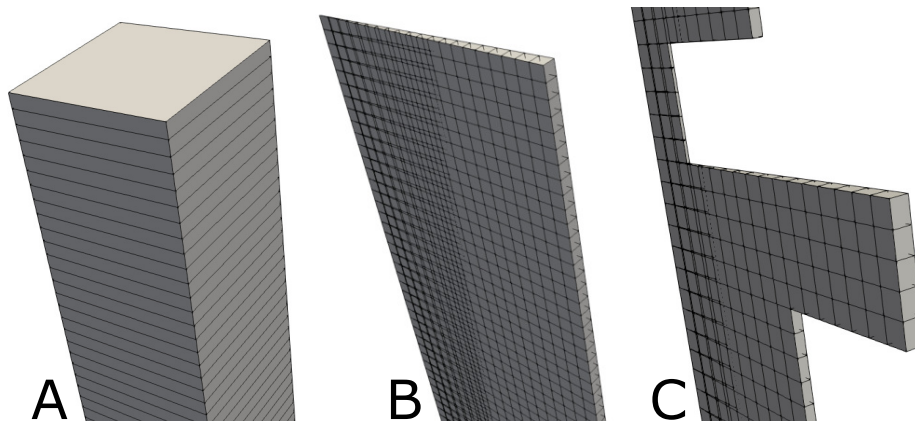


Figure 7: Meshes for the example case: A 1D mesh, B wedge shape mesh of the conventional tubular reactor, and C wedge shape mesh of the unconventional notched reactor

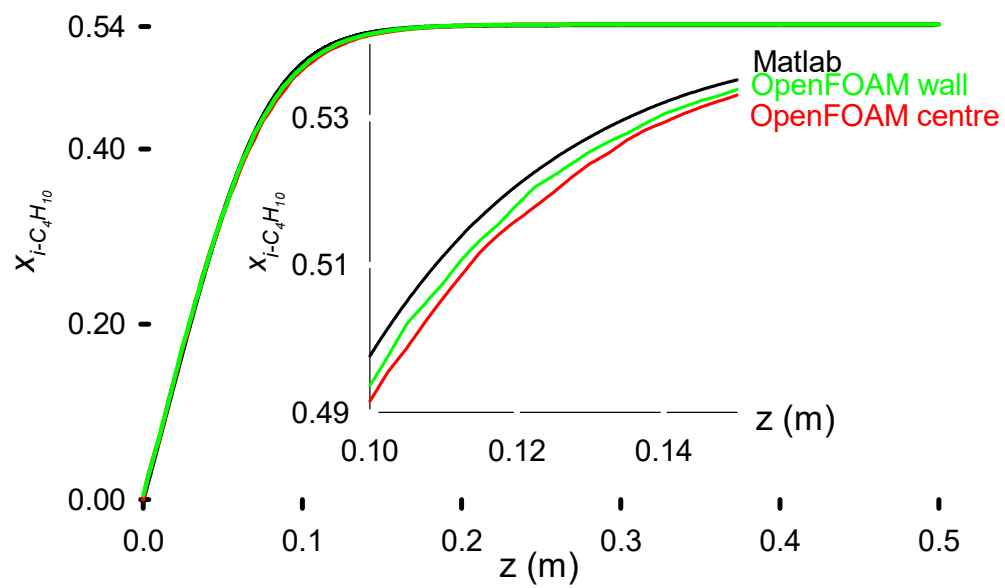


Figure 8: PFR model solved with MATLAB R2019a (black line) versus CFD simulation (concentration profile extracted at the centre and near the wall). Window: zoom of the region with the highest gradient

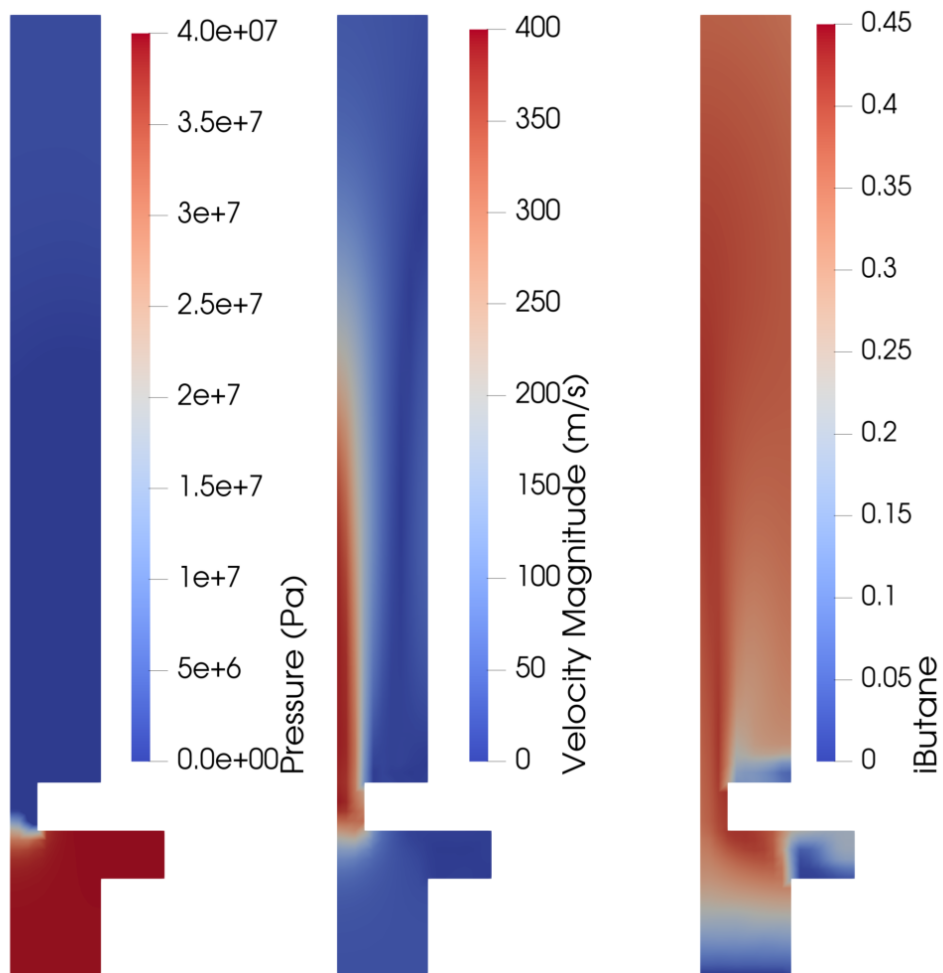


Figure 9: The results from the simulation of the notched reactor, showing (from left to right) pressure (in Pa), velocity (in $\text{m} \cdot \text{s}^{-1}$), and *i*-butane mass fraction. The direction of the flow goes from bottom to top

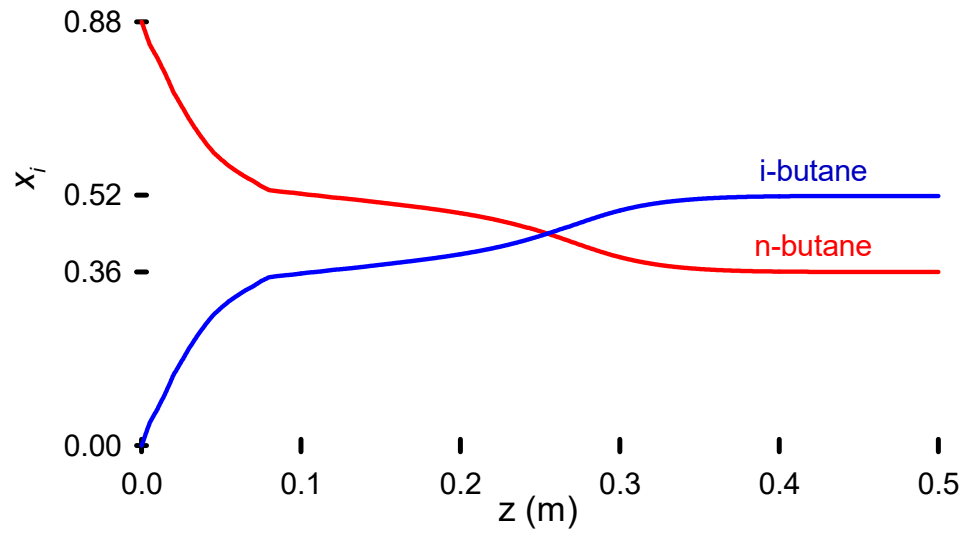


Figure 10: *n*-Butane and *i*-butane mass fraction over the length of the notched reactor at the centre

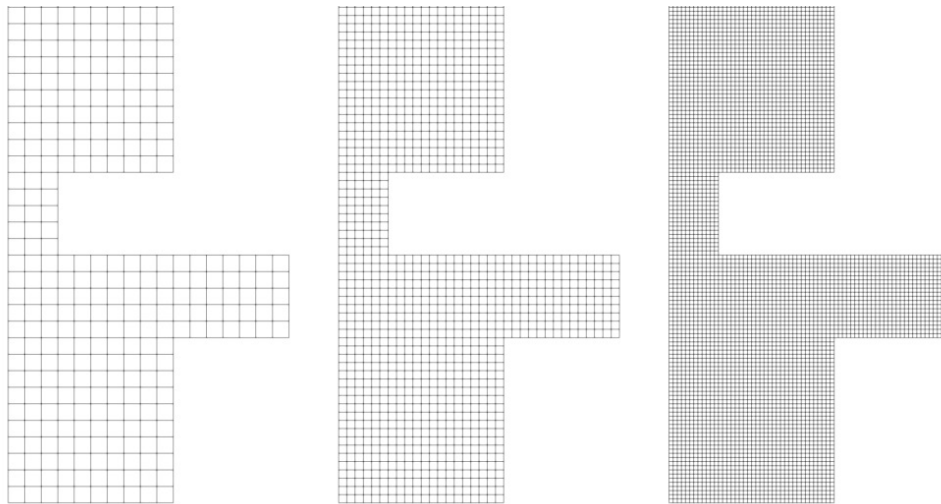


Figure 11: Mesh geometry for the grid refinement study: coarse grid (left), medium grid (middle), and fine grid (right)

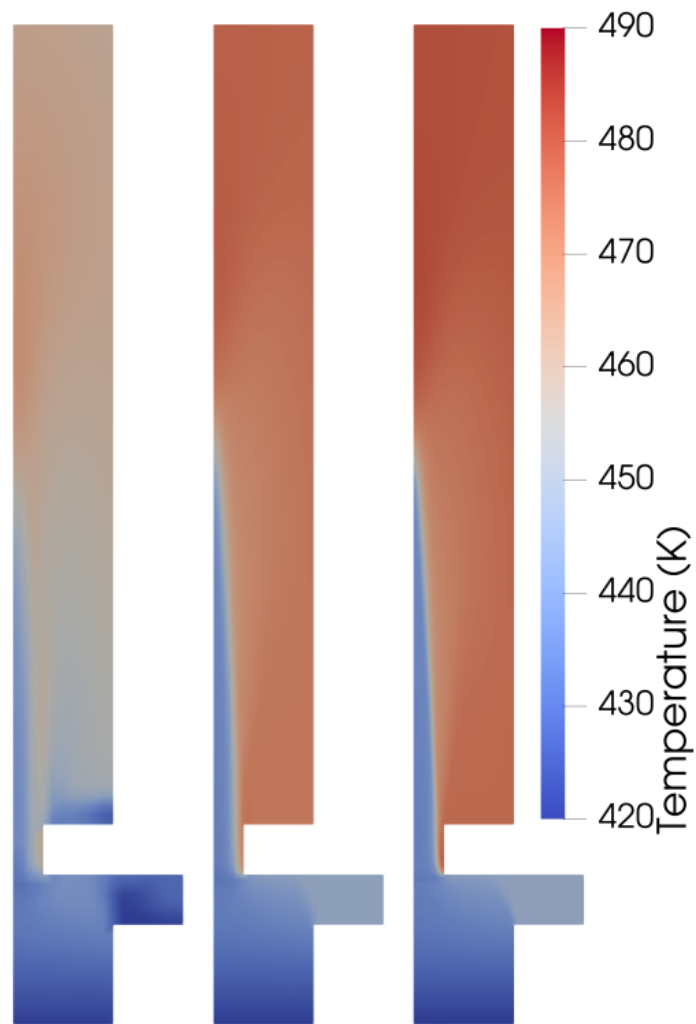


Figure 12: The temperature on the notched reactor for each of the three meshes used for the grid convergence study, that is, on the coarse mesh (left), the intermediate mesh (middle), and the fine mesh (right)

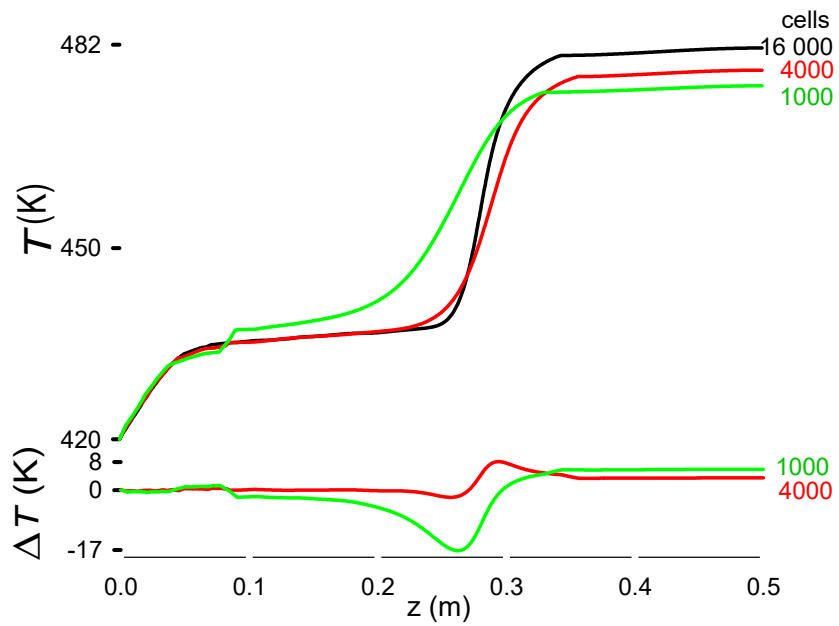


Figure 13: Temperature profile as a function of number of cells. The lower plot is the difference in temperature between the profile with 16 000 cells and 4000 cells (red) and 16 000 cells and 1000 cells (green)

