**RESEARCH**

# Assessment of Numerical Accuracy and Parallel Performance of OpenFOAM and its Reacting Flow Extension EBIdnsFoam

Thorsten Zirwes[1,2,3] · Marvin Sontheimer[3] · Feichi Zhang[2,4] ·
Abouelmagd Abdelsamie[5] · Francisco E. Hernández Pérez[6] · Oliver T. Stein[2,3] ·
Hong G. Im[6] · Andreas Kronenburg[3] · Henning Bockhorn[2]

**Abstract**

OpenFOAM is one of the most widely used open-source computational fluid dynamics tools and often employed for chemical engineering applications. However, there is no systematic assessment of OpenFOAM's numerical accuracy and parallel performance for chemically reacting flows. For the first time, this work provides a direct comparison between OpenFOAM's built-in flow solvers as well as its reacting flow extension EBIdnsFoam with four other, well established high-fidelity combustion codes. Quantification of OpenFOAM's numerical accuracy is achieved with a benchmark suite that has recently been established by Abdelsamie et al. (Comput Fluids 223:104935, 2021. https://doi.org/10.1016/j.compfluid.2021.104935) for combustion codes. Fourth-order convergence can be achieved with OpenFOAM's own cubic interpolation scheme and excellent agreement with other high-fidelity codes is presented for incompressible flows as well as more complex cases including heat conduction and molecular diffusion in multi-component mixtures. In terms of computational performance, the simulation of incompressible non-reacting flows with OpenFOAM is slower than the other codes, but similar performance is achieved for reacting flows with excellent parallel scalability. For the benchmark case of hydrogen flames interacting with a Taylor–Green vortex, differences between low-Mach and compressible solvers are identified which highlight the need for more investigations into reliable benchmarks for reacting flow solvers. The results from this work provide the first contribution of a fully implicit compressible combustion solver to the benchmark suite and are thus valuable to the combustion community. The OpenFOAM cases are publicly available and serve as guide for achieving the highest numerical accuracy as well as a basis for future developments.

**Keywords** OpenFOAM · Chemically reacting flows · Combustion code benchmark · Taylor–Green vortex · High performance computing

---

Extended author information available on the last page of the article

# 1 Introduction

Numerical simulations have proven to be a helpful and reliable tool in science and engineering. One of the most widely used open-source computational fluid dynamics (CFD) libraries is OpenFOAM (Weller et al. (1998), The open surce CFD toolbox www.openfoam.com). However, there is no systematic assessment of OpenFOAM's numerical accuracy and computational performance in the context of chemically reacting flows.

Due to its use of unstructured computational meshes with arbitrarily shaped cells, OpenFOAM is most commonly used for Reynolds-averaged Navier–Stokes (RANS) and large eddy simulations (LES) involving complex geometries of real engineering applications, where the numerical accuracy is typically limited to second order at best. However, if well-conditioned or structured meshes are used, OpenFOAM offers built-in higher-than-second-order discretization schemes. Because of this, OpenFOAM has been used as a tool for direct numerical simulations (DNS) across various fields of science (Komen et al. 2014; Habchi and Antar 2015; Addad et al. 2015; Lecrivain et al. 2016; Chu and Laurien 2016a, b; Chu et al. 2016; Tufano et al. 2016; Vo et al. 2016, 2017; Wang et al. 2017, 2018; Komen et al. 2017; Bricteux et al. 2017; Zhong et al. 2017; Zheng et al. 2018; Tufano et al. 2019; Zheng et al. 2019; Zirwes et al. 2020). In this context, the term quasi-DNS is sometimes used to highlight that OpenFOAM's spatial discretization schemes are essentially limited to fourth-order accurate interpolation schemes and temporal discretization is limited to second order accuracy. Most publications assessing OpenFOAM's accuracy focus on newly developed numerical tools based on the OpenFOAM library, e.g. Hassanaly et al. (2018); Komen et al. (2020); Gärtner et al. (2020), but rarely on its built-in capabilities (Lee (2017); Noriega et al. (2018)). It is therefore important to not only assess OpenFOAM's numerical accuracy, but also to provide a direct comparison with other, well established high-fidelity codes for chemically reacting flows.

Even though OpenFOAM is often used for the simulation of combustion phenomena, it has some important shortcomings. Firstly, OpenFOAM does not possess the ability to compute detailed molecular transport coefficients from kinetic gas theory for multi-component mixtures. The accurate modeling of diffusion coefficients and heat conductivity is however important to capture the correct internal structure of flames (Poinsot and Veynante 2001). Secondly, detailed simulations of three-dimensional flames are computationally very expensive and therefore demand an efficient numerical implementation. The reasons being the large range of time and length scales governing combustion processes (Peters 1999), as well as the the large number of chemical reactions and intermediate chemical species that have to be accounted for. Even the oxidation of simple fuels like methane is described by hundreds of chemical reactions (Smith et al. 1999) forming a numerically stiff system of equations (Law 2010). The tools offered by OpenFOAM for these types of problems lead to high simulation times (Zirwes et al. 2017). To overcome these shortcomings, the extension EBIdnsFoam has been developed for OpenFOAM, which implements detailed transport coefficients, provides several performance optimizations that can reduce total simulation times by up to 70 % for common combustion cases (Zirwes et al. 2018) and achieves excellent parallel scalability for large-scale simulations.

Because there are no standardized numerical benchmarks for reacting flow solvers providing a full verification and validation chain, Abdelsamie et al. have recently proposed a benchmark suite (Abdelsamie et al. 2021) based on the well-known Taylor–Green vortex (TGV). This numerical test case has been used for the validation of numerical schemes of non-reacting flow solvers in the past decades (Orszag 1974; Brachet 1991; Shu et al.

2005), including a predecessor of OpenFOAM (Drikakis et al. 2007). The new combustion benchmark suite consists of four cases (here referred to as steps), from two-dimensional incompressible flow with an analytic solution to a hydrogen flame interacting with the three-dimensional Taylor–Green vortex. The benchmark is originally designed for low-Mach number codes and three high-fidelity codes were presented in the original publication. The benchmark suite has also been used to validate a hybrid Lattice–Boltzmann finite-difference solver (Hosseini et al. 2020).

The aim of this work is to systematically assess OpenFOAM's numerical accuracy and computational performance by providing extensive direct comparison with four well-established, high-fidelity solvers for reacting flows. The main questions addressed in this work are:

- How do OpenFOAM's built-in higher-than-second order schemes perform compared to classical linear interpolation schemes? And how do they compare to dedicated high-fidelity codes?
- What is the parallel efficiency of OpenFOAM for both simple flow cases as well as reacting flows when using state-of-the-art chemistry acceleration techniques implemented into `EBIdnsFoam`?
- What is the current state of code validation for combustion codes? Do different simulation codes yield the same results for flames developing within the TGV flow field?

The basis for this is the benchmark suite by Abdelsamie et al. the new results by OpenFOAM from this work provide an important contribution to the benchmark from the first compressible and fully implicit code. Throughout this work, three different flow types are considered:

- *Incompressible* denotes the simulation of flows assuming constant density;
- *Low-Mach* number treatment of flows refers to the use of a spatially homogeneous thermodynamic pressure computed from the domain-averaged temperature (Abdelsamie et al. (2021)), which is used in the computation of density, thermo-physical properties and chemical reaction rates;
- *Compressible* treatment means that at each location of the computational domain, density and pressure are fully coupled so that the (thermodynamic) pressure field is inhomogeneous in space and allows for the generation of pressure waves.

Although all presented cases fall into the regime of low-Mach flows according to their Mach number, the numerical treatment of either low-Mach flows with spatially constant thermodynamic pressure or fully compressible flows has an impact on the reacting flow cases, later discussed in Sect. 4.4.

The paper is structured as follows: Sect. 2 details the numerical methods available in OpenFOAM relevant for this work and explains the implementation details of the EBIdnsFoam extension. A brief description of the high-fidelity reacting flow codes used for comparison is given as well. Simple validation cases are presented in Sect. 3 that assess the convergence order of OpenFOAM and validate the velocity-pressure coupling with a pulsatile pipe flow setup. Two simulations of canonical combustion setups, i.e. zero-dimensional (0D) auto-ignition and one-dimensional (1D) freely propagating flame, are compared to the open-source library Cantera (Goodwin et al. 2021). The four steps of the new benchmark suite are conducted in Sect. 4. The first step is an incompressible simulation of the two-dimensional Taylor–Green vortex with analytic solution and the second step is

an incompressible simulation of the three-dimensional TGV where a pseudo-spectral DNS reference solution is available. The third step superimposes an inhomogeneous temperature field onto the TGV velocity field as well as a spatially varying mixture composition of the gas, thus simulating the mixing of an inhomogeneous gas mixture including diffusion of mass and energy. The fourth step simulates a flame interacting with the TGV and thus additionally considers chemical reactions. Section 5 summarizes the findings from this work.

## 2 Numerical Codes

The simulations presented throughout this work have been conducted with OpenFOAM. Additionally, results published in Abdelsamie et al. (2021) from three high-fidelity codes, DINO, Nek5000 and YALES2 are presented together with the OpenFOAM results. Lastly, new results from the compressible high-order code KARFS are analyzed in Sect. 4.4. These codes are briefly introduced in this section.

### 2.1 OpenFOAM

OpenFOAM provides a large number of different solvers for many types of applications and employs the finite volume method (FVM) on collocated meshes to discretize the governing equations. It can use unstructured, non-conforming meshes with arbitrary polyhedral cell shapes. The solution of the governing equations is treated implicitly, which makes OpenFOAM the only code with implicit time discretization discussed in this work. Two second-order time discretization schemes are available, the Crank-Nicolson method and the second-order backward (BDF2) scheme.

Because OpenFOAM applies the finite volume method, the spatial discretization consists of an interpolation from cell center values to the cell faces and a subsequent integration of the flux over the cell faces. For the cell face integration, a single integration point is used, which in general is second order accurate. For the interpolation to the cell faces, OpenFOAM offers a number of interpolation methods including the first-order upwind scheme and the second-order linear interpolation scheme. A fourth-order scheme is available as well, which is called "cubic" in OpenFOAM. It uses a cubic interpolation by taking into account the cell centroid values of the quantity $\varphi$ as well as its explicitly computed gradients $\partial_{n_f}$ normal to the cell faces, according to

$$
\varphi_f = \underbrace{\lambda \varphi_C + (1 - \lambda)\varphi_N}_{\text{implicit}}
$$
$$
+ \underbrace{(2\lambda^3 - 3\lambda^2 + \lambda)(\varphi_N - \varphi_C) + (-\lambda^3 + \lambda^2)\partial_{n_f}\varphi_C + (-\lambda^3 + 2\lambda^2 - \lambda)\partial_{n_f}\varphi_N}_{\text{explicit correction}} . \tag{1}
$$

The variable $\varphi_f$ is the value of $\varphi$ interpolated to the cell face $f$, and $\varphi_C$ and $\varphi_N$ are the values in the cell centers of the current ($C$) and neighbor ($N$) cell. The parameter $\lambda = \frac{x_f - x_N}{x_C - x_N}$ is the normalized distance between the cell face and centers. The polynomial of degree three in Eq. (1) is rearranged so that the first part can be evaluated implicitly from the standard central difference scheme and the second part constitutes an explicit correction. Because of the explicit contribution, the scheme tends to be less stable and is only viable on well-conditioned meshes and with low Courant–Friedrichs–Lewy (*CFL*) numbers.

Another higher-than-second-order discretization method is given by the recently developed third-party weighted essentially non-oscillating (WENO) scheme library for Open-FOAM (Gärtner et al. 2020) based on the work of Martin and Shevchuk (2018). It implements the WENO concept, which introduces a polynomial representation of the cell values. This library is used for step 1 and step 2 of the benchmark suite for the convective term of the momentum balance equation with a polynomial of degree four (WENO4).

The time step of the simulation is adjusted according to the Courant–Friedrichs–Lewy (*CFL*) and Fourier (*Fo*) number, where the *CFL* number is defined as

$$CFL = \frac{u\Delta t}{\Delta x},$$ (2)

with *u* being is the local convective velocity, $\Delta t$ the time step width and $\Delta x$ the length scale of the computational cell. The Fourier number is evaluated from

$$Fo = \frac{a\Delta t}{\Delta x^2},$$ (3)

where *a* is the thermal diffusivity. It should be noted that even for the compressible solvers, the *CFL* number is the convective *CFL* number and thus evaluated with the local flow velocity, not the speed of sound. For additional discussion on this, see Sect. 4.4.
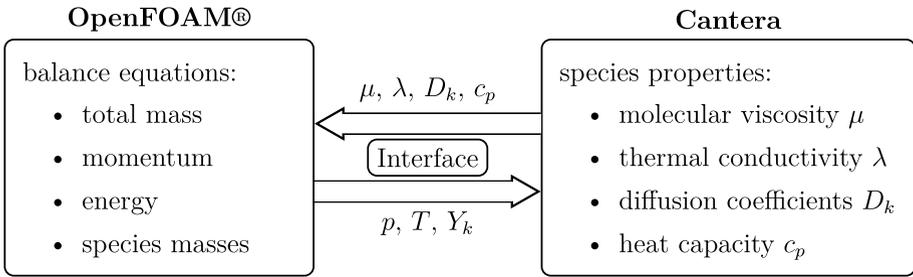
All OpenFOAM simulations presented in this work have been performed with the versions v1712 and v2006. More specifically, all simulations with EBIdnsFOAM are performed with OpenFOAM v1712 as well as the incompressible simulations in step 1. The simulation applying the WENO scheme in step 1 as well as all simulations in step 2 are run with OpenFOAM v2006. The simulations from step 1 and 2 have been performed additionally with the latest versions v2106 and 9, but no significant differences in terms of the numerical accuracy have been observed.

### 2.1.1 pimpleFoam

For the simulation of incompressible flows in the following sections, OpenFOAM's standard solver pimpleFoam is used. It solves the Navier–Stokes equations assuming constant density and applies the PIMPLE algorithm, which combines the PISO (Issa (1986)) and SIMPLE (Caretto et al. 1973) algorithms, for pressure–velocity coupling. An extension of pimpleFoam to compressible flows is given by OpenFOAM's rhoPimpleFoam solver, which solves an additional energy equation and computes the density from an equation of state.

### 2.1.2 EBIdnsFoam

As described in the introduction, OpenFOAM's built-in reacting flow solvers do not support the computation of detailed diffusion coefficients. Because of this, the extension EBIdnsFoam (Solver for detailed and efficient simulation of reacting flows https://vbt.ebi. kit.edu/594.php, Zhang et al. 2015, 2016) has been developed at the Engler-Bunte-Institute. It couples OpenFOAM's finite volume capabilities for the solution of the fully compressible Navier–Stokes equations and the balance equations for the species masses and mixture energy together with Cantera's routines for computing detailed thermo-physical and transport properties. Cantera (Goodwin et al. 2021) is an open-source thermo-chemical library commonly used in the combustion community. Figure 1 illustrates this coupling. In

**OpenFOAM®**                           **Cantera**

balance equations:
- total mass
- momentum
- energy
- species masses

$\mu, \lambda, D_k, c_p$

[Interface]

$p, T, Y_k$

species properties:
- molecular viscosity $\mu$
- thermal conductivity $\lambda$
- diffusion coefficients $D_k$
- heat capacity $c_p$

**Fig. 1** Coupling between OpenFOAM and Cantera for the EBIdnsFoam code

this way, `EBIdnsFoam` behaves numerically in the same manner as the standard `react-ingFoam` solver but uses Cantera to provide the transport coefficients.

The governing equations implemented into EBIdnsFoam are (Poinsot and Veynante (2001)):

- conservation of total mass

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \tag{4}$$

- conservation of momentum

$$\frac{\partial (\rho \vec{u})}{\partial t} + \nabla \cdot (\rho \vec{u} \vec{u}) = -\nabla p + \nabla \cdot \left[ \mu \left( \nabla \vec{u} + (\nabla \vec{u})^{\mathsf{T}} - \frac{2}{3} I \nabla \cdot \vec{u} \right) \right] \tag{5}$$

- balance of species masses

$$\frac{\partial (\rho Y_k)}{\partial t} + \nabla \cdot \left( \rho (\vec{u} + \vec{u}_c) Y_k \right) = \dot{\omega}_k - \nabla \cdot \vec{j}_k, \ \ k = 1 \ldots N-1 \tag{6}$$

- balance of energy

$$\frac{\partial \left( \rho (h_s + \frac{1}{2} \vec{u} \cdot \vec{u}) \right)}{\partial t} + \nabla \cdot \left( \rho \vec{u} (h_s + \frac{1}{2} \vec{u} \cdot \vec{u}) \right) = -\nabla \cdot \vec{q} + \frac{\partial p}{\partial t} - \sum_k h_k^\circ \dot{\omega}_k. \tag{7}$$

Here, $\rho$ is the gas mixture density, $t$ is time, $\vec{u}$ is the bulk fluid velocity, $p$ is the pressure, $Y_k$ is the mass fraction of species $k$, $I$ is the identity tensor, $\mu$ is the dynamic viscosity of the gas mixture, $\dot{\omega}_k$ is the reaction rate of species $k$ and $N$ the total number of species. The correction velocity $\vec{u}_c$ assures that the sum of all diffusive fluxes $\vec{j}_k$ is zero. In the balance equation for energy, $c_p$ is the isobaric heat capacity of the gas mixture and $T$ the temperature. The enthalpy of formation $h_k^0$ of species $k$ and the sensible enthalpy of the mixture $h_s$ for ideal gases are computed from NASA polynomials. The diffusive heat flux is computed from

$$\vec{q} = -\frac{\lambda}{c_p} \nabla h_s + \sum_k \left( \frac{\lambda}{c_p} h_{s,k} \nabla Y_k \right) + \sum_k \left( h_{s,k} \vec{j}_k \right), \tag{8}$$

where $\lambda$ is the heat conductivity of the gas mixture and $h_{s,k}$ is the sensible enthalpy of the $k$-th species.

EBIdnsFoam has been used in previous studies, including the simulation of partially premixed flames (Zirwes et al. 2020, 2021; Hansinger et al. 2020; Soysal et al. 2019), Bunsen and slot burner flames (Zhang et al. 2017; Zirwes et al. 2021; Chen et al. 2021), flame-wall interaction (Steinhausen et al. 2020; Zirwes et al. 2021; Zhang et al. 2021; Steinhausen et al. 2022; Zhang et al. 2022; Driss et al. 2022; Zhu et al. 2023), turbulent flames with instrinsic instabilities (Zirwes et al. 2021, 2022), combustion noise (Zhang et al. 2017, 2019), spherically expanding flames (Zhang et al. 2017; Eckart et al. 2022), as well as ignition phenomena (Zirwes et al. 2019; Häber et al. 2017; Wang et al. 2021, 2022), soot formation (Wang et al. 2023) and combustion under rocket engine conditions (Martinez-Sanchis et al. 2022; Martínez-Sanchis et al. 2022). For more information about the implemented governing equations and code details, we refer the reader to Zirwes et al. (2020); Zhang et al. (2015); Zirwes et al. (2021).

Another focus in the development of EBIdnsFoam lies on computational performance, especially in the context of high-performance computing. One key optimization is the use of automatically generated routines containing highly optimized source code for the computation of chemical reaction rates (Zirwes et al. (2017, 2018, 2019)), which is able to reduce total simulation times by up to 70 % without affecting the accuracy of the simulation. Due to the high numerical stiffness generally introduced by the chemical reaction rates, an operator splitting approach is used in EBIdnsFoam. The open-source integrator CVODE by Sundials (Hindmarsh et al. (2005)) is used to integrate the chemical reaction rates over the simulation time step and provides time-step averaged source terms for the balance equations.

In the next sections, results of OpenFOAM's standard solvers and EBIdnsFoam are compared with other high-fidelity codes. These are described briefly in the following subsections. For a more detailed description, see the original benchmark suite publication (Abdelsamie et al. 2021) and the references given below. A brief overview of the discussed codes and their most important features is given in Table 1.

## 2.2 DINO

DINO (Abdelsamie et al. 2016; Chi et al. 2017) is a three-dimensional, finite differences (FD) based computational fluid dynamics code that supports incompressible and

**Table 1** Overview of numerical codes and their most important features

| Code | EBIdnsFoam | DINO | Nek5000 | YALES2 | KARFS |
| --- | --- | --- | --- | --- | --- |
| Mesh type | Unstructured | Structured | Unstructured | Unstructured | Structured |
| Cell type | Any polyhedra | Hexahedra | Hexahedra | Hexahedra | Hexahedra |
| Discr. type | FVM | FD | SEM | FVM | FD |
| Spatial order | Face interp. 4th flux integr. 2nd | 6th | 7th–15th | 4th | 8th |
| Temporal order | impl. 2nd | expl. RK4 | Semi-impl. BDF3 | expl. RK4 | expl. RK4 |
| Flow type | Compressible | Low-Mach | Low-Mach | Low-Mach | Compressible |
| Thermo-library | Cantera | Cantera | Chemkin | Cantera | Cantera |
| Rate integr | CVODE | Explicit RK4 | CVODE | CVODE | CVODE |

low-Mach number flows. In the context of the Taylor–Green vortex benchmark suite, spatial derivatives are evaluated from a sixth-order central difference scheme and time integration is performed using an explicit fourth-order Runge–Kutta (RK) method. Thermophysical and transport properties are computed with Cantera 2.4.0.

### 2.3 Nek5000

The solver based on Nek5000 (2017), which is used in the Taylor–Green benchmark suite, has been developed at ETH Zürich and applies the spectral element method (SEM) to solve incompressible flows and low-Mach number chemically reacting flows. Its spatial discretization order is 7th–15th order, with a semi-implicit BDF3 scheme for temporal discretization.

### 2.4 YALES2

YALES2 (Moureau et al. 2011) simulates low-Mach number flows with the finite volume method, supporting unstructured meshes. It applies a fourth-order accurate node-based centered scheme for spatial discretization and uses a fourth-order explicit Runge–Kutta method for time integration.

### 2.5 KARFS

The KAUST Adaptive Reacting Flow Solver (KARFS) (Hernández Pérez et al. 2018) simulates fully compressible reacting flows. It uses the finite difference method with an 8th-order spatial discretization and explicit RK4 method for time integration. Its programming model allows work to be offloaded to GPUs and other hardware accelerators.

## 3 Validation Cases for pimpleFoam and EBIdnsFoam

Before the benchmark suite by Abdelsamie et al. (2021) is discussed, a few additional validation cases for both OpenFOAM's standard solvers and EBIdnsFoam are presented.

### 3.1 Convection-Diffusion Equation

A simple one-dimensional steady-state convection-diffusion equation,

$$\nabla \cdot \left( \rho \vec{u} \varphi \right) = \nabla \cdot (\Gamma \nabla \varphi), \ Pe = \frac{\rho |u| L}{\Gamma}, \ \varphi_{\text{exact}}(x) = \frac{\exp(xPe/L) - 1}{\exp(Pe) - 1}, \tag{9}$$

is used to quantify OpenFOAM's convergence order. $Pe$ is the Péclet number and $\varphi$ is an arbitrary scalar quantity. The boundary conditions are $\varphi(x = 0) = 0$ and $\varphi(x = L) = 1$, and the analytic solution is then given by $\varphi_{\text{exact}}(x)$ in Eq. (9). The numerical mesh consists of equidistantly spaced cells with size $\Delta x$. The case has been run with OpenFOAM's scalarTransportFoam solver.

Figure 2 shows the $L_1$ error for different mesh resolutions, which is defined as

Fig. 2 $L_1$ error of the numerical solution of the convection-diffusion equation as a function of the mesh resolution
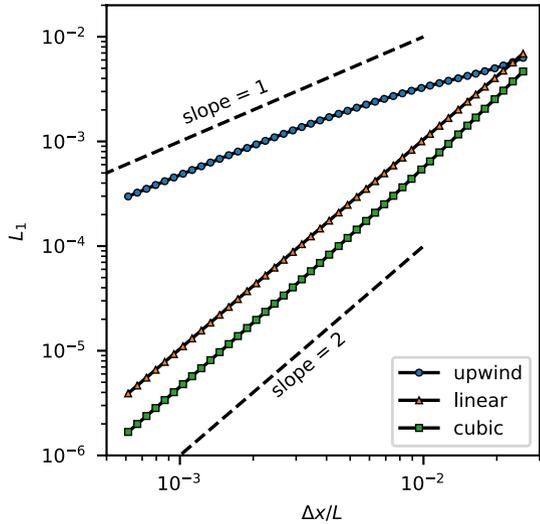


$$L_1 := \frac{1}{N} \sum_{i=1}^{N} \left| \varphi(x_i) - \varphi_{\text{exact}}(x_i) \right|, \tag{10}$$

where $N$ is the number of grid points. Running the simulation with the first-order upwind scheme yields the expected convergence order of one. By using the linear and the cubic scheme (see Sect. 2.1), a second-order convergence is achieved. Even though the cubic scheme employs a fourth-order accurate cell face interpolation, the overall scheme is limited by the second-order face flux integration. However, the cubic scheme is less dissipative than the linear interpolation scheme, resulting in a lower $L_1$ error.

## 3.2 Pulsatile Pipe Flow

In the next setup, the pressure–velocity coupling from OpenFOAM's standard pimpleFoam and rhoPimpleFoam solvers are validated. The numerical setup is built to resemble an experimental setup for pulsatile laminar pipe flows (Ray et al. 2005; Ünsal 2008). Air enters a pipe with a length of $L/D = 714$. The velocity profile at the inlet is set to a block profile. The mean mass flow rate is set according to a Reynolds number of $Re_{\bar{u}} = 2947$. The mass flow rate at the inlet oscillates harmonically with a prescribed amplitude $m_A^\star$ and frequency $f$, as shown in Fig. 3. The pipe is considered as a two-dimensional axisymmetric case, with a radial resolution of $D/\Delta r = 100$ and an axial resolution of $L/\Delta x = 1000$. The numerical settings are summarized in Table 2.

As the air enters the domain and flows along the pipe, the velocity profile changes from the block profile given at the inlet to the parabolic profile for fully developed pipe flow. To validate the pressure–velocity coupling, the amplitude ratio of the mass flow rate $\dot{m}$ and pressure gradient $P = \partial p / \partial x$, as well as the phase shift between mass flow rate and pressure gradient $\Delta\theta$ are evaluated. Figure 4 shows an example of the time signal of the mass flow rate $\dot{m}$ normalized by the mean mass flow rate $\bar{\dot{m}}$ and the time signal of the pressure gradient $P$ normalized by the mean pressure gradient $P_M$. The

**Fig. 3** Numerical domain and inlet condition for the pulsatile pipe flow

inlet: $\frac{u_x}{\bar{u}} = (1 + m_A^\star \sin(2\pi f t))$



$u_x$

$L/D = 714$
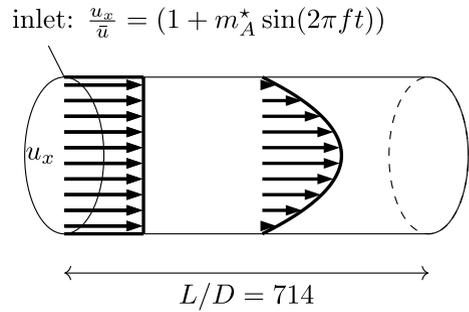
**Table 2** Physical and numerical settings for the pulsatile pipe flow

| Fluid property | Value |
| --- | --- |
| $p_0$ | 5 bar |
| $T_0$ | 300 K |
| $\mu_0$ | $1.8 \times 10^{-5}$ Pa s |
| $M_{\text{air}}$ | 28.9 g/mol |
| $c_p$ | 1007 J/kg/K |
| $Pr$ | 0.7 |
| $\kappa$ | 1.4 |
| $\rho_0$ | 5.79 kg/m$^3$ |
| Mesh/flow | Value |
| $L$ | 10 m |
| $D$ | 14 mm |
| $\Delta r$ | 0.14 mm |
| $\Delta x$ | 10 mm |
| $\bar{m}$ | 35 g/min |
| $\bar{u}$ | 0.654 m/s |
| $Re_{\bar{u}}$ | 2947 |
| $m_A^\star$ | 0.82 |

time signals are recorded at a position near the end of the pipe, where the flow is fully developed. With increasing frequency, the normalized pressure gradient amplitude $P_A^*$ has to increase in order to support the same normalized mass flow rate amplitude $\dot{m}_A^*$. At the same time, the phase shift between the two signals shifts from 0° to 90° with increasing frequency.

An analytic solution for fully developed, pulsatile pipe flows is available as function of the dimensionless frequency $F$,

$$F = \frac{R^2 f}{\nu}, \tag{11}$$

where $R$ is the pipe diameter and $\nu$ the viscosity. The analytic solution reads (Lambossy (1952), Grace (1928), Ünsal (2008), Haddad et al. (2010), Ünsal et al. (2005))

**Fig. 4** Time signal of normalized mass flow rate and normalized pressure gradient over normalized time $t \cdot f$ recorded at the point $x/L = 0.75$ near the end of the pipe for a normalized frequency of $F = 1$
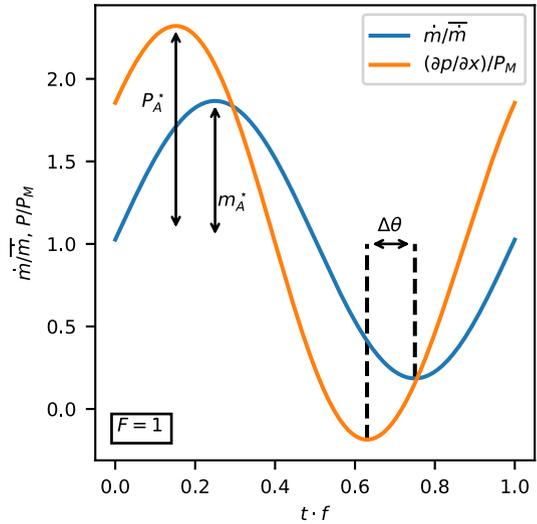


$$\Psi = -\frac{4}{\pi F}\left(1 + \frac{2i^{1/2}J_1\left[(2\pi F)^{1/2}i^{3/2}\right]}{(2\pi F)^{1/2}J_0\left[(2\pi F)^{1/2}i^{3/2}\right]}\right), \tag{12}$$

where $i$ is the imaginary unit and $J_n$ is the $n$-th Bessel function of the first kind. The amplitude ratio can then be obtained from

$$\frac{\dot{m}_A^*}{P_A^*} = |\Psi|, \tag{13}$$

and the phase shift from

$$\Delta\theta = \tan^{-1}\left(\frac{\Re(\Psi)}{\Im(\Psi)}\right), \tag{14}$$

where $\Re(\Psi)$ is the real part and $\Im(\Psi)$ the imaginary part of $\Psi$.

The pulsatile pipe flow case has been run with OpenFOAM's incompressible flow solver pimpleFoam and its compressible equivalent rhoPimpleFoam for different $F$. The evaluated amplitude ratio of the mass flow rate and pressure gradient as well as the phase shift between mass flow rate signal and pressure gradient signal show excellent agreement with the analytic solution, as shown in Fig. 5. In this way, OpenFOAM's pressure–velocity coupling implementation is validated.

## 3.3 Canonical Combustion Cases

In order to validate the implementation of chemical reaction rates and detailed molecular diffusive fluxes in the EBIdnsFoam solver, two canonical cases for combustion solvers are presented.
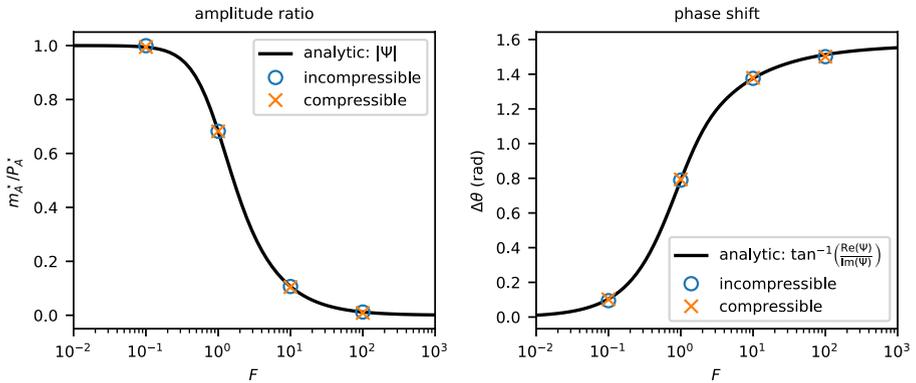
**Fig. 5** Ratio of the normalized mass flow rate amplitude and normalized pressure gradient amplitude (left) as well as time shift between mass flow rate and pressure gradient (right) for pulsatile pipe flow

### 3.3.1 0D Auto-Ignition

The first canonical case for reacting flows is the zero-dimensional (0D) auto-ignition of methane in air. A homogeneous, constant-volume batch reactor is filled with the unburnt gas mixture at a temperature above the auto-ignition temperature. In this case, the initial temperature is set to 1600 K and the initial pressure to 1 bar. Hydrogen is added to the gas mixture to speed up the ignition process. The initial conditions are shown in the table in Fig. 6. The reaction mechanism is GRI 3.0 (Smith et al. 1999). This case evaluates the accuracy of the temporal numerical integration of the chemical source terms.

As the simulation starts, the mixture begins to ignite and transitions to the burnt state. The temporal evolution of species mass fractions obtained from EBIdnsFoam is depicted in Fig. 7 together with reference results from Cantera. The maximum numerical difference between the EBIdnsFoam and Cantera solutions are below 0.1 %.



Initial conditions:

| Species | Value |
|---|---|
| $Y_{CH_4,0}$ | 0.0514 |
| $Y_{O_2,0}$ | 0.2206 |
| $Y_{N_2,0}$ | 0.7261 |
| $Y_{H_2,0}$ | 0.0017 |
| $Y_{H,0}$ | 0.0002 |

$T_0 = 1600\,\text{K}$
$p_0 = 1\,\text{bar}$
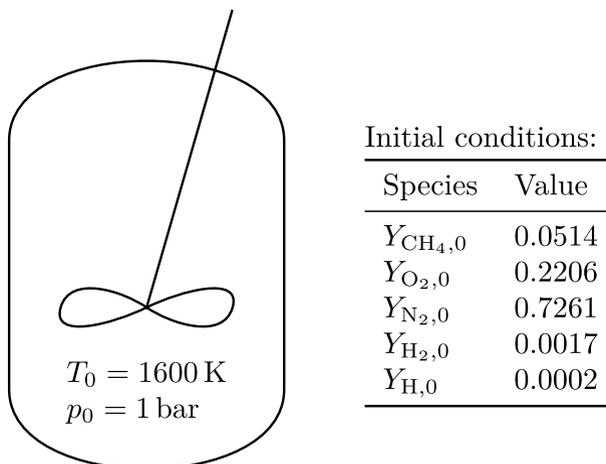
**Fig. 6** Initial conditions for temperature $T$, pressure $p$ and species mass fractions $Y$ for the auto-ignition case
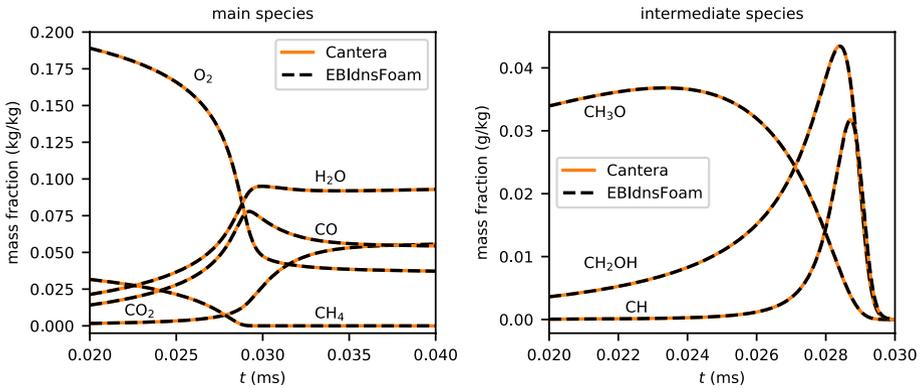
**Fig. 7** Temporal evolution of species mass fractions of main (left) and intermediate (right) species for the auto-ignition case. The reference solution is provided by Cantera

### 3.3.2 1D Freely Propagating Flame

The second case is a one-dimensional freely propagating flame.

A stoichiometric mixture of the unburnt methane and air is introduced at the inflow plane on the left of the computational domain. The inlet velocity is set to the laminar flame speed $s_{L,0}$ to obtain a steady-state solution. The left half of the domain is initially filled with the unburnt gas mixture and the right half is filled with the combustion products at chemical equilibrium. The reaction mechanism is again GRI 3.0, and the diffusion model is the mixture-averaged model (Curtiss-Hirschfelder approximation) (Kee et al. 2005). The computational setup and boundary conditions are summarized in Fig. 8.

Figure 9 shows the spatial profiles of species mass fractions at the steady-state solution. Again, the agreement with Cantera is very good, with largest differences below 0.5 %.

## 4 Taylor–Green Vortex Benchmark Suite

In this section, the four steps of the new benchmark suite for reacting flow solvers by Abdelsamie et al. (2021) are conducted with OpenFOAM's standard solver pimpleFoam (steps 1 and 2) and the custom solver EBIdnsFoam (steps 3 and 4). For steps 1–3, results are directly compared to the low-Mach codes DINO, Nek5000 and YALES2 to give a perspective on the accuracy of OpenFOAM. For step 4, the compressible code KARFS is additionally used to highlight the difference between the low-Mach and compressible
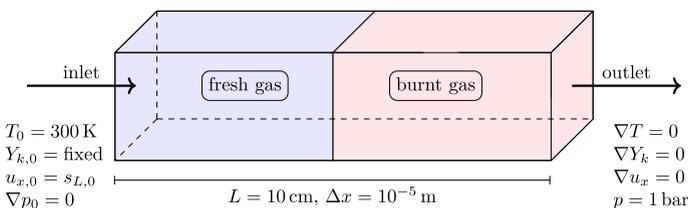


**Fig. 8** Computational domain, boundary and initial conditions for the freely propagating flame case
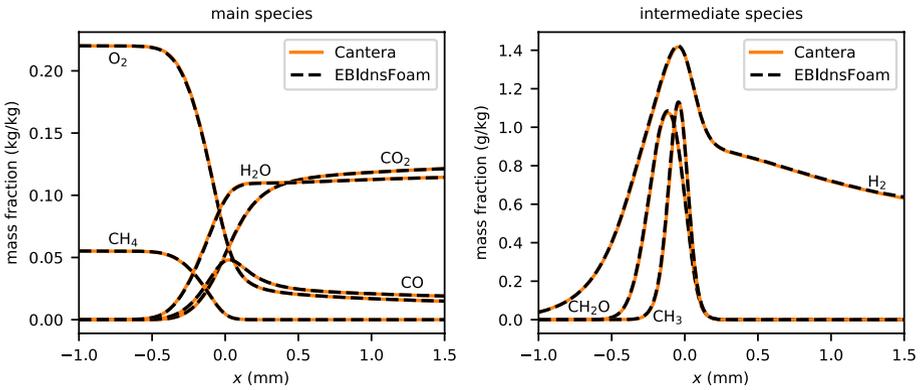
**Fig. 9** Spatial profiles of species mass fractions of main (left) and intermediate (right) species for the freely propagating flame case. The reference solution is provided by Cantera

codes. Each step of the benchmark suite is briefly described in the following subsections. For an in-depth description of the respective numerical cases, see Abdelsamie et al. (2021). The numerical case setups for OpenFOAM for each step are available at (OpenFOAM Data for the TGV benchmark suite for reacting flows 2023).

## 4.1 Step 1: 2D Incompressible Single-Component TGV

The first case is an incompressible, two-dimensional Taylor–Green vortex flow. The domain is a square with side lengths $L$ and periodic boundary conditions. The velocity field is given by the analytic solution

$$u_x(x, y, t) = u_0 \sin\left(\frac{2\pi x}{L}\right) \cos\left(\frac{2\pi y}{L}\right) \exp\left(-\frac{8\pi^2 \nu t}{L^2}\right), \tag{15}$$

$$u_y(x, y, t) = -u_0 \cos\left(\frac{2\pi x}{L}\right) \sin\left(\frac{2\pi y}{L}\right) \exp\left(-\frac{8\pi^2 \nu t}{L^2}\right), \tag{16}$$

**Fig. 10** Flow field of the two-dimensional incompressible Taylor–Green vortex (TGV) from step 1
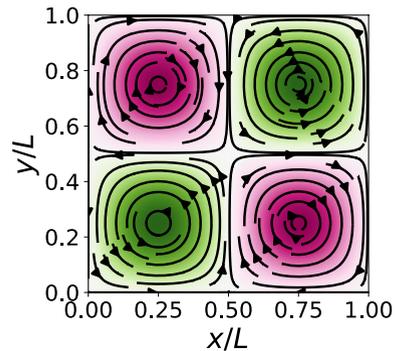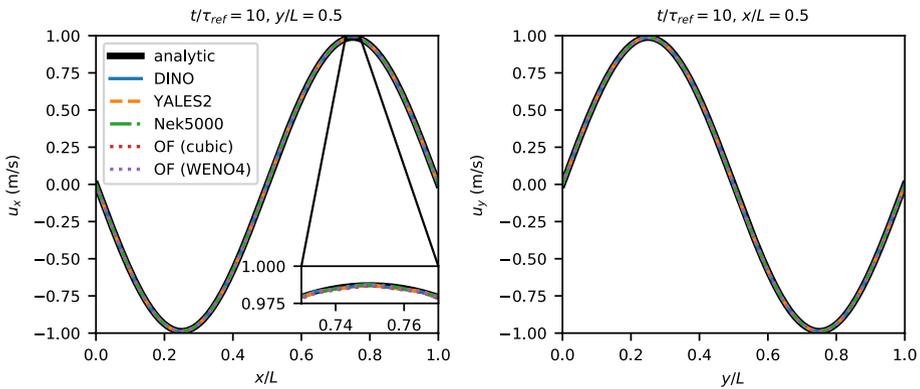
**Fig. 11** Velocity profiles along the centerlines of the computational domain at $t = 10\tau_{ref}$ from the analytic solution, OpenFOAM (OF) and the three high-fidelity codes for the 2D TGV case from step 1

**Table 3** Maximum velocity magnitude of the analytic solution and the different simulation codes as well as the relative error $\varepsilon_{rel}$ to the analytic solution at $t = 10\tau_{ref}$

|  | Analytic | OF (cubic) | OF (WENO4) | DINO | YALES2 | Nek5000 |
|---|---|---|---|---|---|---|
| max($|u|$) (m/s) | 0.9875778 | 0.9873981 | 0.9875671 | 0.987565 | 0.987583 | 0.9875785 |
| $\varepsilon_{rel}$ (%) | — | $1.8 \times 10^{-2}$ | $1.1 \times 10^{-3}$ | $1.3 \times 10^{-3}$ | $5.3 \times 10^{-4}$ | $7.1 \times 10^{-5}$ |

with $u_0 = 1$ m/s, $L = 2\pi L_0$, $L_0 = 1$ m, and $v = \frac{1}{1600}$ m²/s. Figure 10 shows a picture of the initial velocity field colored by the vorticity to show the four counter-rotating vortices.

The numerical simulations are run with a constant time step of $\Delta t = 5 \times 10^{-4}$ s until $t = 10\tau_{ref}$ with a suggested grid resolution of $64^2$ cells. The reference time scale is $\tau_{ref} = 1$ s.

Figure 11 shows the $x$-component of velocity along the centerline at $y/L = 0.5$ (left) and the $y$-component of velocity along the centerline at $x/L = 0.5$ (right) at $t = 10\tau_{ref}$. The black line in the background denotes the analytic solution. The results from DINO and YALES2 are obtained using a mesh resolution of $64^2$ cells or grid points, and the results of Nek5000 are reported for $8^2$ spectral elements which is equivalent to $65^2$ grid points. For the OpenFOAM (OF) solutions, a mesh with $65^2$ cells was used so that the domain centerlines cross through the cell centers of the mesh. Two OpenFOAM simulations are conducted: one where all spatial derivatives are discretized with the cubic scheme (OF cubic), and one where the WENO4 scheme is used for the convective term of the momentum equation (OF WENO4). Visually, all simulation codes are in perfect agreement. Even the zoomed view on the left of Fig. 12 shows no visual differences. Because of this, Table 3 compares the maximum velocity at $t = 10\tau_{ref}$ from all codes with the analytic solution. Using the same mesh and the same time step, the peak velocity values of OpenFOAM lie within $1.8 \times 10^{-2}$ % and $1.1 \times 10^{-3}$ % of the analytic solution. In this way, the achieved accuracy is of the same order of magnitude as the one from DINO and YALES2. Appendix 29 provides additional velocity profiles at coarser mesh resolutions.

Before moving to the next step of the benchmark suite, the influence of various discretization schemes from OpenFOAM as well as the achieved convergence order are discussed. Figure 12 shows the same velocity profile as in Fig. 11 on the left, but all results shown in the Figure are now obtained from OpenFOAM using different spatial discretization schemes. It is

**Fig. 12** Velocity profile of $u_x$ along the centerline at $t = 10\tau_{\mathrm{ref}}$ from OpenFOAM simulations with different spatial discretization schemes for the 2D TGV case from step 1
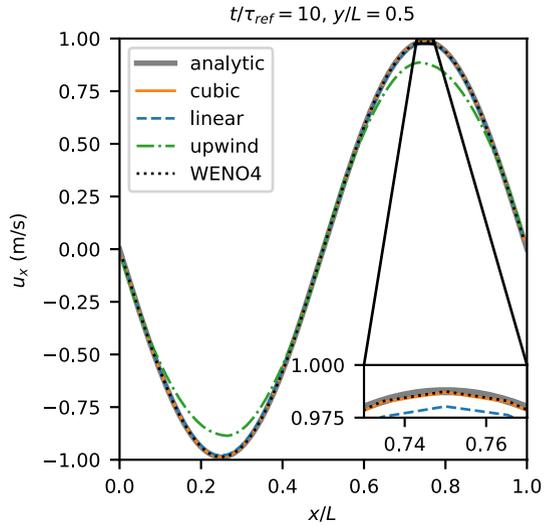


clearly visible that the first-order upwind scheme is too dissipative. By zooming into the peak velocity region, the solution obtained with the linear scheme also shows visible deviation from the analytic solution, whereas the cubic and the WENO4 schemes show excellent agreement with the analytic solution.

Due to the existence of an analytic solution, the spatial convergence order of OpenFOAM can be determined. Figure 13 shows the $L_1$ (see Eq. (10)) and $L_2$ error for the $u_x$ velocity profile at $t = 10\tau_{\mathrm{ref}}$ for different mesh resolutions obtained with the cubic discretization scheme. The $L_2$ error is evaluated from

**Fig. 13** $L_1$ and $L_2$ errors obtained with OpenFOAM's cubic discretization for the 2D TGV case from step 1
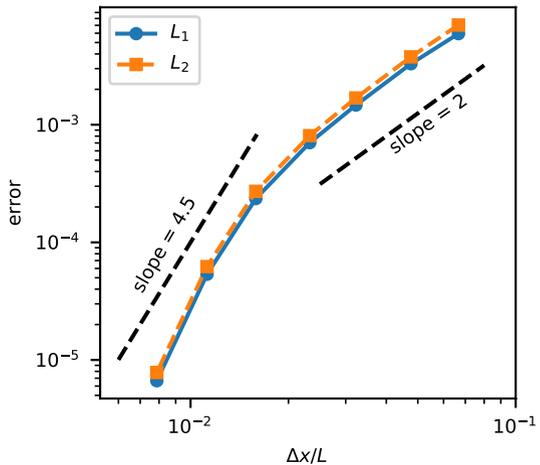
$$L_2 := \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left( u_x(x_i) - u_{x,\text{exact}}(x_i) \right)^2}. \tag{17}$$

For coarse meshes, the convergence order is approximately 2 (see also Sect. 3.1). However, at finer meshes, the convergence order approaches 4th order. This explains the overall good agreement with the high-fidelity codes shown above.

## 4.2 Step 2: 3D Incompressible Single-Component TGV

Step 2 of the benchmark suite is the incompressible simulation of the three-dimensional Taylor–Green vortex. The computational domain is a cube with side length $L = 2\pi L_0$ and $L_0 = 1$ m. All faces of the cube are defined as periodic boundary conditions, and the initial velocity field is given by

$$u_x = u_0 \sin\left(\frac{2\pi x}{L}\right) \cos\left(\frac{2\pi y}{L}\right) \cos\left(\frac{2\pi z}{L}\right), \tag{18}$$

$$u_y = -u_0 \cos\left(\frac{2\pi x}{L}\right) \sin\left(\frac{2\pi y}{L}\right) \cos\left(\frac{2\pi z}{L}\right), \tag{19}$$

$$u_z = 0, \tag{20}$$

with $u_0 = 1$ m/s. The viscosity is again set to $\nu = \frac{1}{1600}$ m²/s, leading to a Reynolds number of $Re = u_0 L_0 / \nu = 1600$. The characteristic time scale is $\tau_{\text{ref}} = 1$ s. All OpenFOAM simulations performed for the 3D TGV cases are run with outer PIMPLE iterations until initial residuals of all quantities are below $5 \times 10^{-10}$ or up to a maximum of 10 outer iterations. For this case, there is no analytic solution, but a reference pseudo-spectral DNS on a mesh with $512^3$ grid points is available, which is denoted here as RLPK (Van Rees et al. 2011). This and all subsequent 3D TGV cases are run without sub-grid turbulence models. While the resolution of $512^3$ grid points can be seen as DNS resolution (Van Rees et al. 2011), simulations at lower resolution would correspond to an underresolved DNS or LES without explicit turbulence model. However, since the decaying flow field of the TGV stays symmetrical throughout the whole simulation time considered here (pseudo-turbulence), the terms underresolved LES or implicit sub-grid model can be misleading and are omitted from the description of all cases in this paper.

Figure 14 shows iso-surfaces of the Q-criterion at the beginning of the simulation (left). After about 10 s, the flow decays into pseudo-turbulence, as shown on the right of Fig. 14.

The time step of the simulations is dynamically adjusted to ensure $CFL \leq 0.3$. The computational mesh for the OpenFOAM simulations consists of $513^3$ equidistant cells to ensure that the centerlines pass through the cell centroids, while DINO and YALES2 use $512^3$ cells or grid points. Figure 15 shows the velocity profiles of the $x$ (left) and $y$ (right) components of velocity along the centerlines at $t = 12.11\tau_{\text{ref}}$ from the different codes. The reference solution from RLPK is not available for this specific data set. The simulation with OpenFOAM is again conducted with purely cubic discretization schemes (OF cubic) and a separate simulation is performed using the WENO4 scheme for the convective term of the momentum equation. Visually, the velocity profiles from OpenFOAM agree very well with the three high-fidelity codes. The zoomed region in Fig. 15 on the left shows very good
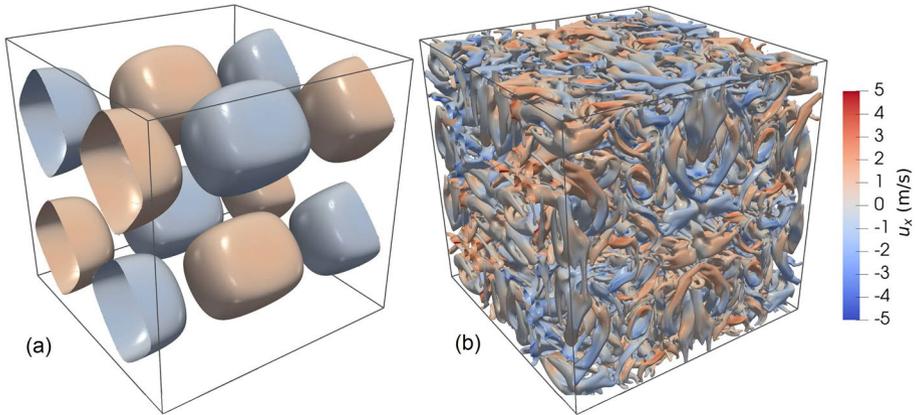
**Fig. 14** Visualization of the incompressible 3D Taylor-Green vortex from step 2 simulated with Open-FOAM showing iso-surfaces of the Q-criterion, colored by the $x$-component of velocity. At the beginning of the simulation, the flow is regular (**a**) but after some time decays into a pseudo-turbulence (**b**)
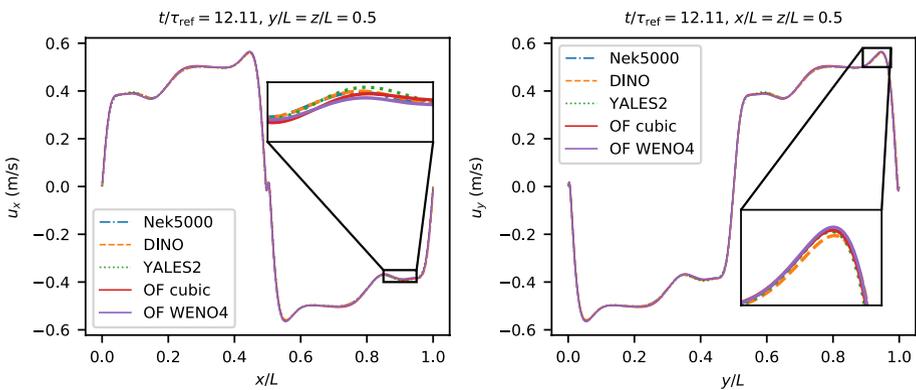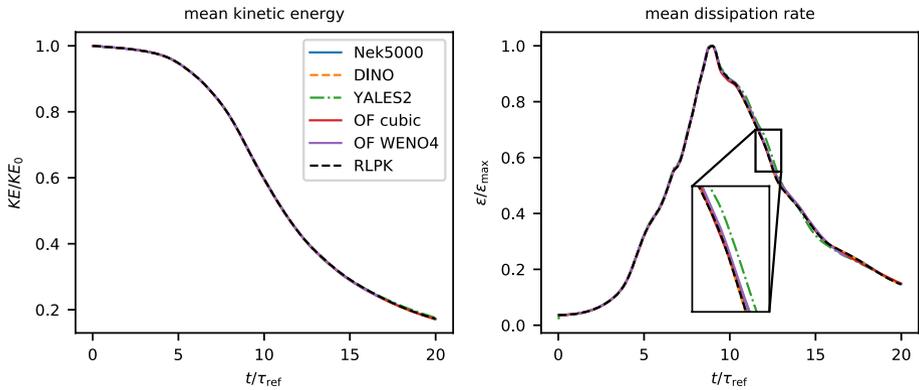


**Fig. 15** Velocity profiles along the centerlines at $t = 12.11\tau_{\text{ref}}$ for the 3D incompressible TGV from step 2

agreement between DINO and Nek5000, while YALES2 and the OpenFOAM results show small deviations. The zoomed region to the peak velocity on the right of Fig. 15, shows a good agreement between all codes, with small deviations by DINO. Again, the results by OpenFOAM compare very well to the high-fidelity codes.

Another important validation is the temporal evolution of the volume-averaged kinetic energy and dissipation rate within the computational domain, defined as

$$KE = \frac{1}{V} \int \frac{1}{2} \vec{u} \cdot \vec{u} \, \mathrm{d}V \tag{21}$$

$$\varepsilon = -\frac{\partial KE}{\partial t} = \frac{2\mu}{V} \int S : S \, \mathrm{d}V, \ S := \frac{1}{2}\left(\nabla \vec{u} + (\nabla \vec{u})^{\mathsf{T}}\right). \tag{22}$$

**Fig. 16** Normalized volume-averaged kinetic energy (left) and dissipation rate (right) over time for the 3D incompressible TGV from step 2

These quantities are shown in Fig. 16 together with the reference solution from the pseudo-spectral DNS (RLPK). Both the temporal evolution of the kinetic energy and of the dissipation rate show very good agreement between all codes (see also appendix B for additional data). The only deviation is observed by YALES2, as shown in the zoomed region on the right of Fig. 16.

The influence of the choice of discretization scheme in OpenFOAM is shown in Fig. 17. All simulations are run with $513^3$ cells, but with different discretization schemes. When looking at the volume-averaged dissipation rate, it is immediately clear that the first-order upwind scheme is not sufficient to describe the dynamics of the TGV. The second-order central difference scheme (CD) is less dissipative, but also shows significant deviations from the reference solution (zoomed region). With the cubic and WENO4 schemes, the reference solution is recovered. Since the dissipation rate in Fig. 17 is computed from the stress tensor, which in turn is computed from the gradient of the velocity field (Eq. (22)),

**Fig. 17** Volume-averaged dissipation rate over time on a mesh with $513^3$ grid cells and different OpenFOAM discretization schemes for the 3D incompressible TGV from step 2
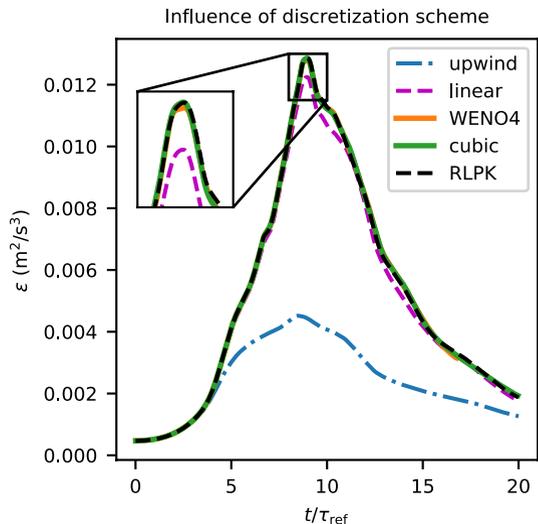
**Fig. 18** Volume-averaged dissipation rate over time for different mesh resolutions from OpenFOAM and DINO for the 3D incompressible TGV from step 2
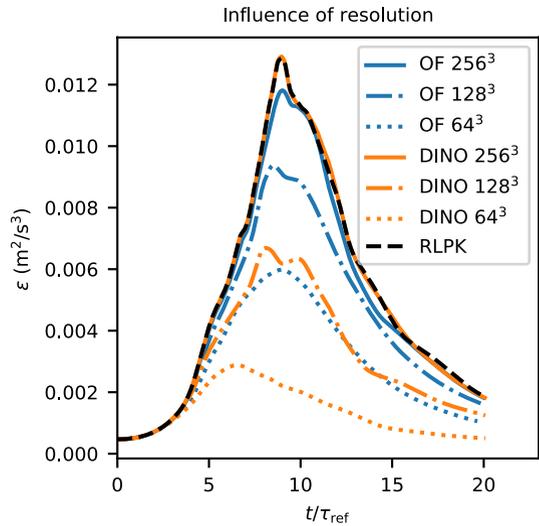


simulations run with schemes having larger numerical dissipation lead to smoother velocity gradients and therefore to lower predicted physical dissipation rates.

Finally, the influence of the mesh resolution is discussed. Figure 18 shows the volume-averaged dissipation rate obtained by OpenFOAM with its cubic discretization scheme as well as DINO with its 6th-order spatial discretization for $64^3$, $128^3$ and $256^3$ cells or grid points. For $256^3$ points, DINO's solution agrees very well with the reference solution and is less dissipative compared to that from OpenFOAM. In order to test the solution with less grid points, an 8th-order explicit filter was activated in DINO, which is not needed in OpenFOAM. Looking at the results in Fig. 18, it can be observed that OpenFOAM with $64^3$, and $128^3$ cells shows less dissipative solutions compared to the ones from DINO with filter. In this way, the comparison with the high-order code DINO again shows the good numerical performance of OpenFOAM's built-in cubic scheme.

### 4.3 Step 3: 3D Multi-component Non-reacting TGV

In steps 1 and 2, the incompressible flow solver pimpleFoam from OpenFOAM has been used. It achieved excellent agreement with both the analytic solution in step 1 as well as with the three high-fidelity codes in steps 1 and 2. In step 3, the EBIdnsFoam solver is used, as the three-dimensional TGV velocity field is superimposed by a temperature field, so that an incompressible or constant-density approach is no longer valid. The concentration profiles for $N_2$, $O_2$ and $H_2$ are set according to
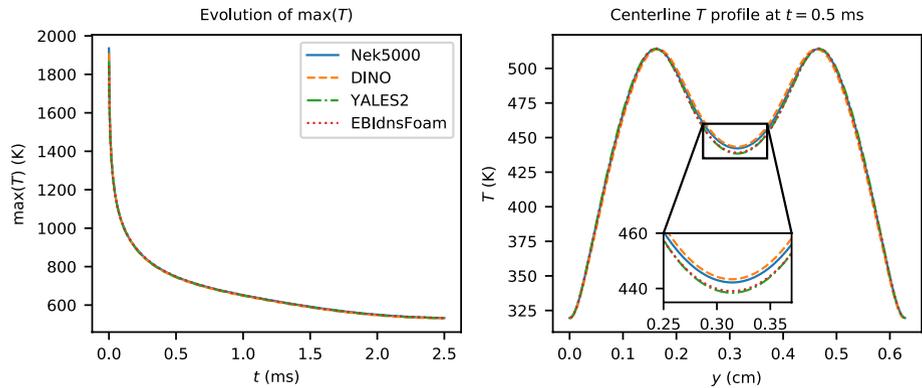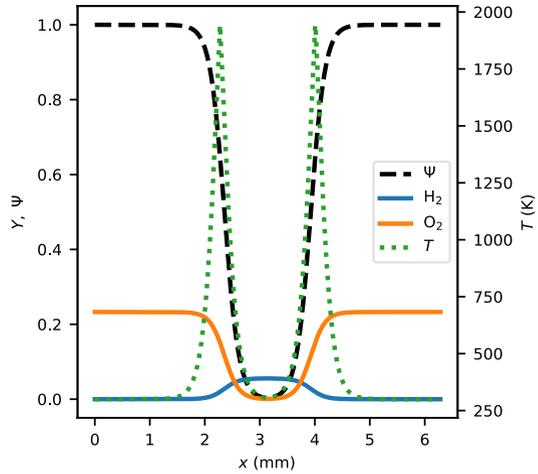
$$Y_{H_2} = 0.0556(1 - \Psi(x)), \tag{23}$$

$$Y_{O_2} = 0.233(1 - \Psi(x)), \tag{24}$$

$$Y_{N_2} = 1 - Y_{O_2} - Y_{H_2}, \tag{25}$$

where $Y_k$ are the mass fractions of species $k$. The shape function $\Psi$ is defined as

**Fig. 19** Initial conditions for mass fraction profiles $Y$ of $H_2$ and $O_2$ (solid lines), shape function $\Psi$ (black dashed line) as well as initial temperature profile (green dotted line) as a function of $x$ for the multi-component non-reacting 3D TGV case





**Fig. 20** Left: temporal evolution of the maximum temperature within the computational domain. Right: Spatial temperature profile along the $y$-centerline at $t = 0.5$ ms for the non-reacting mixing case in step 3

$$\Psi(x) = \frac{1}{2}\left(1 + \tanh\left(\frac{3\left(\left|x - \frac{1}{2}L\right| - R\right)}{R}\right)\right),\ R = \pi/4\,\text{mm}. \tag{26}$$

The characteristic length scale is set to $L = 2\pi$ mm and the velocity to $u_0 = 4$ m/s. The temperature at each position is set to the temperature of the chemical equilibrium according to the local species mixture at 1 atm and 300 K. The thermo-physical properties of the chemical species are given by the thermodynamic quantities associated with the reaction mechanism by Boivin et al. (2011). Figure 19 shows the initial profiles of $H_2$, $O_2$ and $T$ along the centerline. The simulations are performed on a mesh with $256^3$ cells. The time step is dynamically adjusted to ensure $CFL \leq 0.25$ and $Fo \leq 0.15$. Chemical reactions are turned off in this step and the mass diffusion of the species is treated using constant but non-equal Lewis numbers. For more information, see Abdelsamie et al. (2021).

Figure 20 (left) shows the temporal evolution of the maximum temperature inside the simulation domain. Since chemical reactions are disabled for this step, the mixing leads to

a decrease of the peak temperatures. The OpenFOAM-based EBIdnsFoam solver as well as the three high-fidelity codes display again a good agreement. On the right of Fig. 20, the spatial profile of the temperature along the $y$ centerline at $t = 0.5$ ms is depicted. By zooming into the central region, minor differences can be observed between the codes, where the solutions from YALES2 and EBIdnsFoam overlap but are slightly lower than the results from the high-order codes DINO and Nek5000. The peak temperatures on the other hand agree within 1 % between all codes, as shown in Table 4.

Figure 21 shows the $u_x$ and $u_y$ velocity profiles along centerlines at $t = 0.5$ ms. The $u_x$ velocity profile shows again remarkable agreement while smaller deviations by about 3 % of the $u_y$ profile of EBIdnsFoam are visible near the peak values.

**Table 4** Peak temperature along the $y$ centerline at $t = 0.5$ ms

|             | EBIdnsFoam | DINO  | YALES2 | Nek5000 |
| ----------- | ---------- | ----- | ------ | ------- |
| max($T$) (K) | 514.5      | 513.9 | 514.06 | 514.0   |
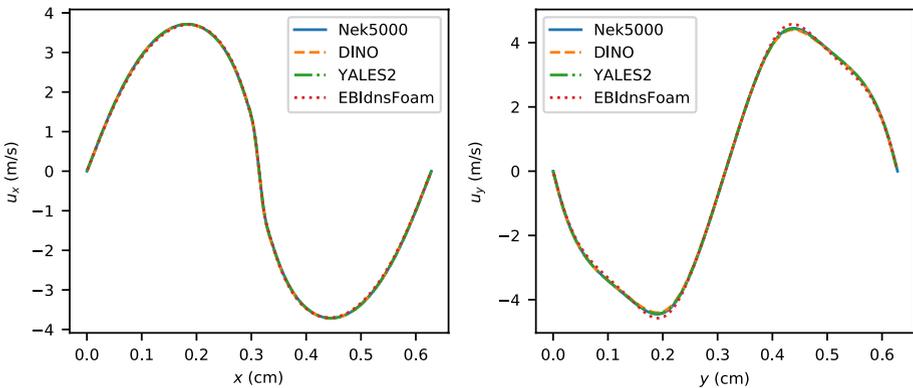


**Fig. 21** Velocity profiles along the $x$ and $y$ centerlines at $t = 0.5$ ms for step 3
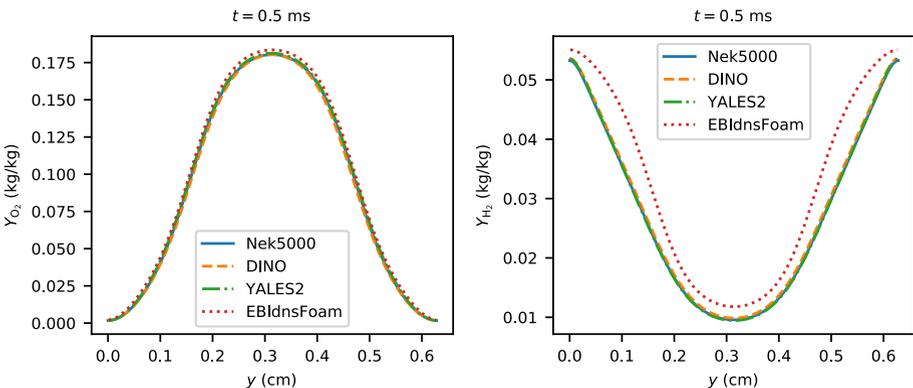


**Fig. 22** Profiles of $O_2$ and $H_2$ mass fractions along the $y$ centerline at $t = 0.5$ ms for step 3. All simulations depicted here use the constant Lewis number diffusion approach

A comparison between the spatial profiles of species mass fractions of $O_2$ and $H_2$ is given in Fig. 22. While the oxygen mass fraction profile yields close agreement with the three high-fidelity codes, there are considerable deviations between EBIdnsFoam and the other three codes for the profile of the hydrogen mass fraction. One major difference between EBIdnsFoam and the three other codes considered so far is that EBIdnsFoam is a compressible code and the other three codes solve for low-Mach number flows. Because of this, the simulation of the non-reactive mixing case has been repeated with the mixture-averaged diffusion model instead of the constant Lewis number approach in order to compare with results from the high-order KARFS code, which is also fully compressible and employs the mixture-averaged diffusion approach.

Figure 23 shows again the mass fraction profile of hydrogen with the same values from DINO as before for reference, but this time with results from EBIdnsFoam using the mixture-averaged diffusion model (mix-avg) and the KARFS code using the mixture-averaged model as well. This time, the agreement for the hydrogen profile between KARFS and EBIdnsFoam is very good. This is further discussed in the next subsection.

### 4.4 Step 4: 3D Multi-component Reacting TGV

The most challenging numerical setup of the benchmark suite is step 4. Compared to step 3, chemical reactions are enabled and 9 species as well as 12 chemical reactions from the Boivin reaction mechanism for hydrogen combustion are considered. The initial conditions for the species are changed, as now not only the temperature is set according to the chemical equilibrium at each position, but also the species concentrations (see Abdelsamie et al. (2021) for reference). The reacting TGV case corresponds to two, initially planar, diffusion flame fronts. Due to the TGV flow field, the flame fronts are distorted. Initially, the local mixture composition is set to the chemical equilibrium at each grid point. Because of this, heat release rates, even within the hot temperature region, are initially zero. As the fuel and oxygen species start to diffuse into the high temperature region and the profiles are convected by the TGV flow field, an ignition event occurs and a net heat release rate establishes that forms the diffusion flame fronts.

**Fig. 23** Profile of the $H_2$ mass fractions along the $y$ centerline at $t = 0.5$ ms using the mixture-averaged diffusion model for EBIdnsFoam and KARFS and constant Lewis number approach for DINO as reference for step 3
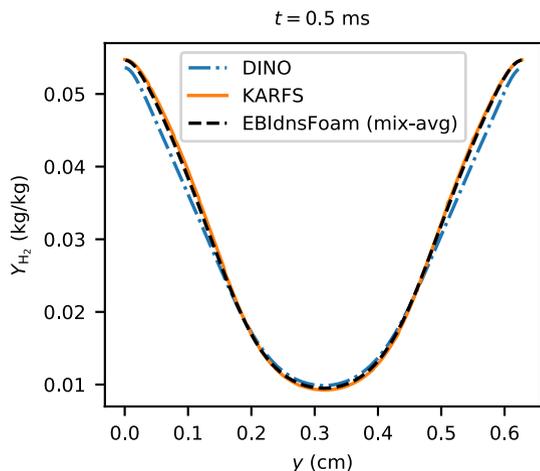
**Fig. 24** Temporal evolution of the maximum temperature inside the domain with the diffusion models mixture-averaged (mix-avg) and constant Lewis number (const Le) from the two compressible codes (comp) and DINO's low-Mach number solution for step 4
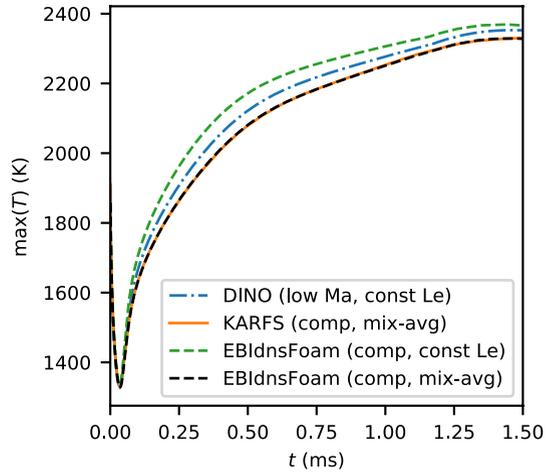


**Fig. 25** Spatial temperature profile along the $y$-centerline at $t = 0.5$ ms for step 4
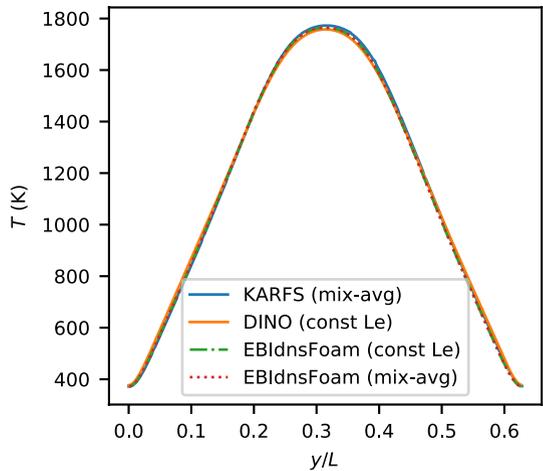


Again, the mesh consists of $256^3$ cells with $CFL \leq 0.25$ and $Fo \leq 0.15$. Because a flame propagates during the simulation, the gas inside the simulation domain expands, which leads to an increase of the pressure, as the periodic boundary conditions act as a closed domain for the global pressure. As a consequence, the low-Mach number codes DINO, YALES2 and Nek5000, which treat the thermodynamic pressure as homogeneous throughout the domain (but variable in time), show a different behavior as explored below.

Figure 24 shows the temporal evolution of the maximum temperature inside the simulation domain. After about 0.1 ms, there are significant discrepancies depending on the employed simulation model: the blue dash-dotted line is the result from DINO, which represents the low-Mach number solvers using the constant Lewis number diffusion approach (const Le); the green dashed line is the compressible EBIdnsFoam solver, using constant Lewis number diffusion as well. However, these two results are not comparable due to the low-Mach vs. compressible approach. On the other hand, the results from KARFS and EBIdnsFoam are almost identical, as in this case both codes use the
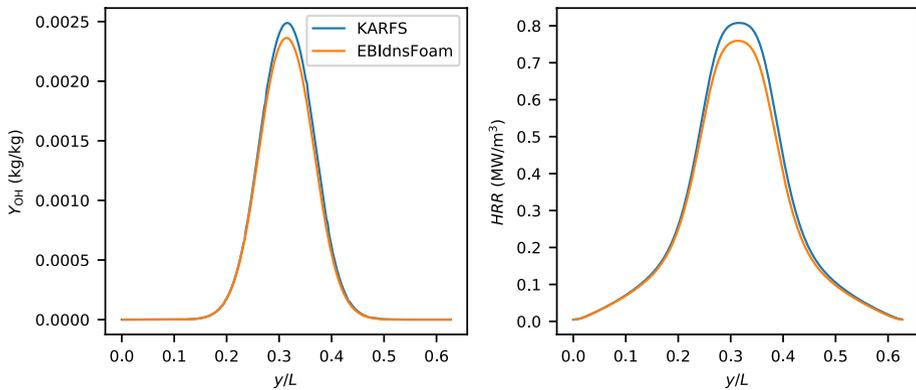
mixture-averaged diffusion model and utilize a compressible flow formulation. The spatial temperature profile along the $y$ centerline at $t = 0.5$ ms is less sensitive to the choice of compressible or low-Mach solver and also to the choice of diffusion model, as shown in Fig. 25. Note that the maximum temperature within the domain, that is shown in Fig. 24 is not located on the line from Fig. 25 but at a different location inside the simulation domain.

Although Fig. 24 suggests that the two compressible codes EBIdnsFoam and KARFS should compare well when considering other variables for the reacting Taylor–Green vortex case, an evaluation of the velocity profile reveals noticeable differences. While the profile of $u_y$ along the centerline at $t = 0.5$ ms shows a good agreement in all cases on the right of Fig. 26 and the two EBIdnsFoam cases agree very well with DINO for $u_x$ (Fig. 26 on the left), the velocity profile of KARFS shows a considerable deviation. One difference is the treatment of the compressibility or velocity-density-pressure coupling. Since KARFS is an explicit, fully compressible code, its time step is based on the acoustic *CFL* number and thus resolves the evolution of pressure waves in the simulation. On the other hand, due to OpenFOAM's implicit treatment of the governing equations and the PIMPLE method, OpenFOAM's stability criterion depends on the convective *CFL* number, so that time steps are usually much larger than what would be needed to resolve pressure waves. The difference in time step size also affects the operator splitting for the coupling of the flow and chemical source terms. However, an additional simulation with OpenFOAM applying the acoustic *CFL* number as time step limiter yields the same results as with the convective *CFL* number. The difference in the velocity profile is also visible when comparing other quantities, like the mass fraction of OH and the heat release rate (HRR), as shown in Fig. 27. Although the results between EBIdnsFoam and KARFS are similar, a significant deviation remains. Since the velocity profiles of EBIdnsFoam and DINO match remarkably well, but the maximum temperature evolution between KARFS and EBIdnsFoam also matches remarkably well, it is not immediately clear which result, if any, is more accurate.

While the uncertainty reported in Abdelsamie et al. (2021) for the reacting TGV case between the three low-Mach number codes is about 1 %, it is much higher for compressible reacting flow solvers. A similar discussion has arisen during the preliminary presentation of benchmark results at the 17th International Conference on Numerical Combustion (May
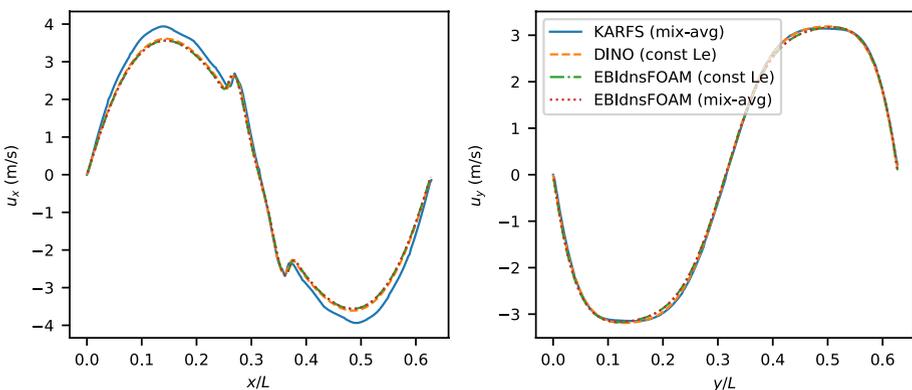


**Fig. 26** Velocity profiles along the centerlines at $t = 0.5$ ms for step 4

**Fig. 27** Spatial profiles of OH mass fraction (left) and heat release rate (right) along the $y$-centerline at $t = 0.5$ ms for step 4, computed with the mixture-averaged diffusion model

6–8, 2019) in Aachen, Germany, without any final conclusion. It is therefore important, to include more compressible reacting flow solver results for this benchmark suite to aid the improvement of existing reacting flow solvers and to provide a reliable numerical benchmark for future codes. All results and setups from the OpenFOAM/EBIdnsFoam simulations are publicly available (OpenFOAM Data for the TGV benchmark suite for reacting flows https://github.com/g3bk47/TGV-OpenFOAM-benchmark).

## 4.5 Parallel Performance

The computational and parallel performance of the different codes can be evaluated by tracking the time required for performing one simulation time step for the different steps of the benchmark suite in dependence on the number of parallel processes. These performance measurements are presented in Fig. 28 for the simulations from step 2–4. The numbers near the data points inside the figures indicate the parallel scaling efficiency $E_n$

$$E_n = \frac{t_{n_0}}{t_n} \frac{n_0}{n},$$ (27)

where $n_0$ is the number of parallel processes or MPI ranks of the first data point, and $t$ the wall-clock time per time step. The measurements have been performed on the following supercomputers:

- YALES2: Irene Joliot-Curie, TGCC, France (Intel Skylake Xeon Platinum 8168 2×24 cores/node)
- Nek5000: Pitz Daint, CSCS, Switzerland (Intel Broadwell Xeon E5-2695 v4 2×18 cores/node)
- DINO: SuperMUC-GN, LRZ, Germany (Intel Skylake Xeon Platinum 8174 2×24 cores/node)
- EBIdnsFoam: HoreKa, KIT, Germany (Intel Xeon Platinum 8368, 2×38 cores/node)

It should be noted that the absolute values given in Fig. 28 serve only as an approximation of the codes' performance. Due to the different CPU types and compute node interconnects,

**Fig. 28** Parallel performance metrics of the different codes from steps 2–4 of the benchmark suite. The numbers near the data points show the scaling efficiency $E_n$ in percent
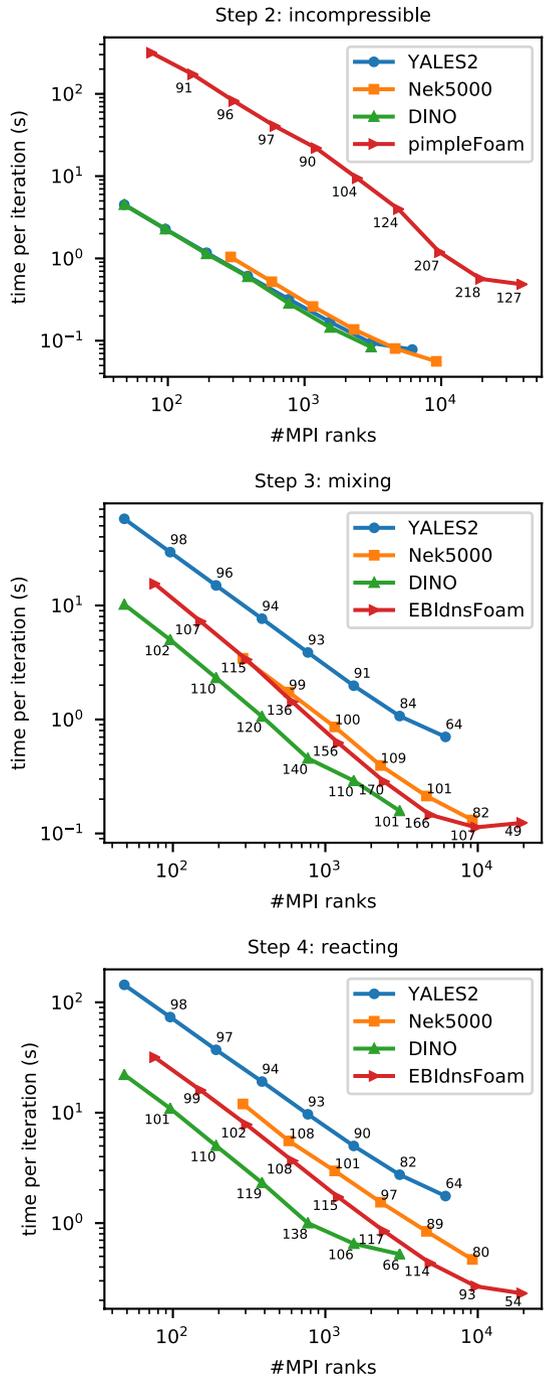
a direct performance comparison is difficult. A more in-depth performance analysis will be conducted in a follow-up work and only basic assessments are given here.

For OpenFOAM's standard pimpleFoam solver (top figure of Fig. 28), the time per iteration is almost an order of magnitude higher than the other codes included in this comparison. The reason for this is that OpenFOAM is a general purpose code with focus on unstructured meshes with arbitrary cell shapes and thus cannot be as efficient as e.g. structured mesh solvers. In the depicted range, the parallel scaling however is very good, being in a super-linear regime.

As briefly described in Sect. 2.1, EBIdnsFoam has been designed with a focus on computational performance, specifically targeting the computation of thermo-physical properties and chemical reaction rates. Therefore, the time required per time step for steps 3 and 4 lies well within the range of the tested high-fidelity codes. The second to last data point for OpenFOAM on the bottom figure in Fig. 28 for step 4 is the performance measurement with 9728 MPI ranks. With that many parallel processes, OpenFOAM is the only code that retains near-perfect parallel scaling ($E_{9728} = 0.93$). This confirms that the EBIdnsFoam extension is well suited for large-scale parallel simulations.

## 5 Summary and Conclusions

In this work we provide a thorough comparison and validation of one of the most used open-source CFD codes OpenFOAM. Although OpenFOAM is often used in the context of complex geometries that do not allow for high-order schemes, it supports fourth-order interpolation schemes. Specifically in the context of reacting flows, there are not many publications assessing the numerical accuracy of OpenFOAM. It is therefore important to provide well-designed benchmark cases and to perform direct comparisons with other well-established codes. Because of this, an extensive validation of OpenFOAM's built-in solvers and numerical schemes, as well as of external developments (a high-order WENO discretization library and the EBIdnsFoam extension for reacting flows) has been conducted in this work. For the first time, a full benchmark suite recently developed for reacting flow solvers has been performed with OpenFOAM in conjunction with direct comparison including four high-fidelity codes to systematically assess the numerical accuracy of OpenFOAM.

The main findings are summarized as follows:

- If the simulation setup allows for hexahedral cells and low *CFL* numbers, it is strongly recommended to use OpenFOAM's cubic discretization scheme due to its higher accuracy, instead of defaulting to its "linear" scheme, which is commonly done in the OpenFOAM community.
- The pressure–velocity coupling approach with OpenFOAM's PIMPLE algorithm has been validated by means of pulsatile pipe flows. Likewise, the implementation of diffusive fluxes and chemical reaction rates in EBIdnsFoam has been validated by comparison of canonical combustion cases with the open-source thermo-chemical library Cantera.
- The numerical accuracy of OpenFOAM in the incompressible flow setups of steps 1 and 2 of the Taylor–Green vortex benchmark is excellent. Depending on the mesh reso-

lution, a convergence order of up to four can be achieved. For the 2D setup, the deviation from the analytic solution is about 0.001 % and thus of the same order of magnitude as DINO and YALES2. Similar results are found for the 3D TGV case.

- For the non-reactive mixing case in step 3, agreement is again very good. However, considerable deviations exist in the spatial profiles of the hydrogen mass fraction. This difference vanishes, when the compressible high-order code KARFS is used for comparison together with the mixture-averaged diffusion model. Therefore, these deviations might be attributable to the low-Mach vs. compressible approach.

- The reactive TGV case in step 4 is sensitive to the flow treatment, i.e. low-Mach number or compressible flow. This, however, leads to additional uncertainties. While the temporal evolution of the maximum temperature agrees best between the compressible KARFS and EBIdnsFoam solvers, the spatial velocity profiles contained in the benchmark match best between the low-Mach DINO code and EBIdnsFoam. In general, the reacting 3D TGV case reveals different results between DINO, KARFS, OpenFOAM and YALES2, depending on which quantity is compared, i.e. velocity fields or mass fraction profiles. This highlights the need for more reliable benchmark cases, especially for compressible reacting flow solvers in the future.

- The results presented in this paper constitute a valuable contribution to the benchmark repository, not only from one of the most widely-used open-source CFD codes, but also from the first fully implicit and compressible reacting flow code. All results and numerical setups for OpenFOAM used in this work are publicly available (OpenFOAM Data for the TGV benchmark suite for reacting flows 2023).
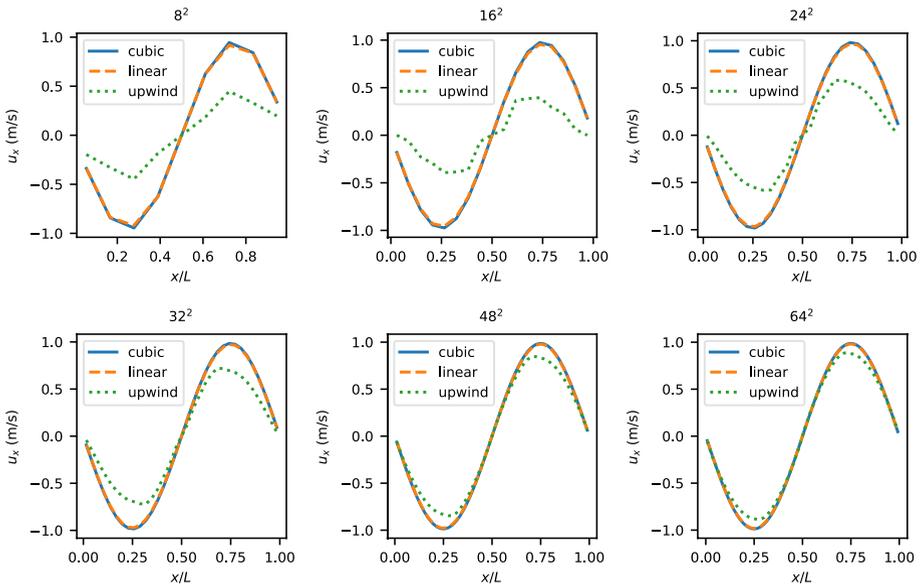


**Fig. 29** Profiles of the $x$-component of velocity along the $x$-centerline at $t/\tau_{\mathrm{ref}} = 18$ for different mesh resolutions and discretization schemes

- EBIdnsFoam is suited for large scale parallel simulations, showing comparable simulation times per time step as the high-fidelity codes and excellent parallel scaling behavior, especially for reacting flow simulations.

## Appendix A Step 1: Velocity Profiles

This appendix presents additional results of the velocity profiles in Fig. 29 for the 2D TGV from Step 1 (Sect. 4.1) for different mesh resolutions.
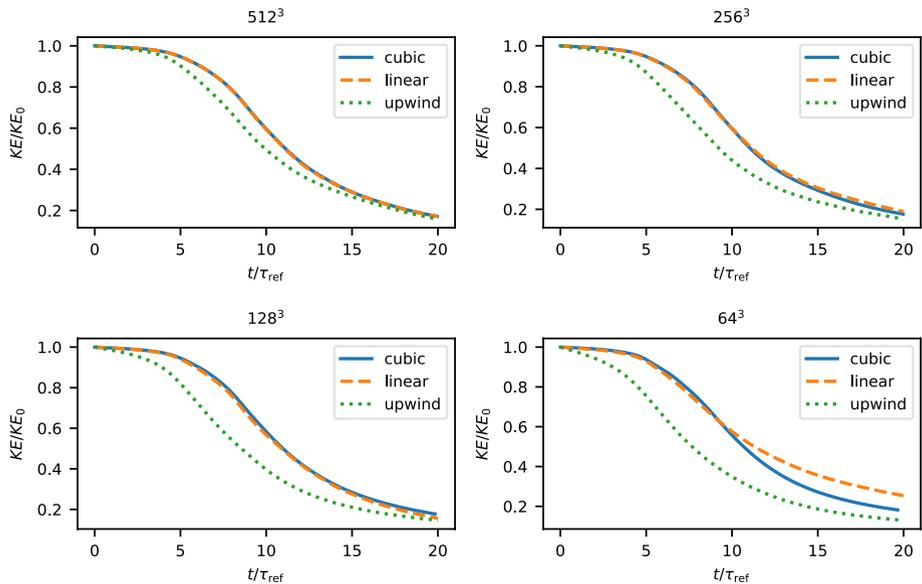


**Fig. 30** Normalized volume-averaged kinetic energy over time for meshes with different mesh sizes, evaluated with the three standard OpenFOAM discretization schemes for the incompressible 3D TGV from Step 2

## Appendix B Step 2: Additional Data for KE

In addition to the comparison of *KE* shown in Fig. 16 of Sect. 4.2, Fig. 30 shows the time evolution of normalized volume-averaged kinetic energy over time for different mesh resolutions and the three standard discretization schemes of OpenFOAM. At the different mesh resolutions, the trend between the three schemes stays roughly the same. The only outlier is the simulation with $64^3$ cells, where the simulation utilizing the linear scheme overpredicts the kinetic energy toward the end of the simulation. This is attributed to numerical oscillations due to under-resolved velocity and pressure gradients.

**Author Contributions** TZ wrote the main text and implemented EBIdnsFoam. TZ and MS ran validation cases and contributed extended analyses on the results. FZ, OTS, AK and HB contributed to the conception of the presented results and structure of the work. AA contributed data for DINO and devised the reference test cases. HGI and FEHP contributed data for KARFS. All authors reviewed the manuscript.

**Availability of data and materials** Described in the manuscript

## Declarations

**Conflict of interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Ethical approval** Not applicable

**Consent to participate** Not applicable

**Consent for publication** Not applicable

**Code availability** Described in the manuscript

## References

Abdelsamie, A., Fru, G., Oster, T., Dietzsch, F., Janiga, G., Thévenin, D.: Towards direct numerical simulations of low-Mach number turbulent reacting and two-phase flows using immersed boundaries. Comput. Fluids **131**, 123–141 (2016)

Abdelsamie, A., Lartigue, G., Frouzakis, C.E., Thévenin, D.: The Taylor-Green vortex as a benchmark for high-fidelity combustion simulations using low-Mach solvers. Comput. Fluids **223**, 104935 (2021). https://doi.org/10.1016/j.compfluid.2021.104935

Addad, Y., Zaidi, I., Laurence, D.: Quasi-DNS of natural convection flow in a cylindrical annuli with an optimal polyhedral mesh refinement. Comput. Fluids **118**, 44–52 (2015). https://doi.org/10.1016/j.compfluid.2015.06.014

ANL: Nek5000 version v17.0 (2017). https://nek5000.mcs.anl.gov, Argonne National Laboratory II

Boivin, P., Jiménez, C., Sánchez, A.L., Williams, F.A.: An explicit reduced mechanism for H2-air combustion. Proc. Combust. Inst. **33**(1), 517–523 (2011)

Brachet, M.: Direct simulation of three-dimensional turbulence in the Taylor-Green vortex. Fluid Dyn. Res. **8**(1–4), 1 (1991)

Bricteux, L., Zeoli, S., Bourgeois, N.: Validation and scalability of an open source parallel flow solver. Concurr. Comput. Pract. Exp. **29**(21), 4330 (2017). https://doi.org/10.1002/cpe.4330

Caretto, L., Gosman, A., Patankar, S., Spalding, D.: Two calculation procedures for steady, three-dimensional flows with recirculation. In: Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics, pp. 60–68 (1973). Springer. https://doi.org/10.1007/BFb0112677

Chen, X., Wang, Y., Zirwes, T., Zhang, F., Bockhorn, H., Chen, Z.: Heat release rate markers for highly-stretched premixed $CH_4$/air and $CH_4$/ $H_2$/air flames. Energy Fuels **35**(16), 13349–13359 (2021). https://doi.org/10.1021/acs.energyfuels.1c02187

Chi, C., Janiga, G., Abdelsamie, A., Zähringer, K., Turányi, T., Thévenin, D.: DNS study of the optimal chemical markers for heat release in syngas flames. Flow Turbul. Combust. **98**(4), 1117–1132 (2017)

Chu, X., Laurien, E.: Direct numerical simulation of heated turbulent pipe flow at supercritical pressure. J. Nuclear Eng. Radiat. Sci. **2**(3), 031019 (2016). https://doi.org/10.1115/1.4032479

Chu, X., Laurien, E.: Flow stratification of supercritical CO2 in a heated horizontal pipe. J. Supercrit. Fluids **116**, 172–189 (2016)

Chu, X., Laurien, E., McEligot, D.M.: Direct numerical simulation of strongly heated air flow in a vertical pipe. Int. J. Heat Mass Transf. **101**, 1163–1176 (2016). https://doi.org/10.1016/j.ijheatmasstransfer.2016.05.038

Drikakis, D., Fureby, C., Grinstein, F.F., Youngs, D.: Simulation of transition and turbulence decay in the Taylor-Green vortex. J. Turbul. **1**(8), 20 (2007)

Driss, K., Steinhausen, M., Zirwes, T., Bockhorn, H., Hasse, C., Ferraro, F.: Combined effects of heat loss and curvature on turbulent flame-wall interaction in a premixed dimethyl ether/air flame. Proc. Combust. Inst. (2022). https://doi.org/10.1016/j.proci.2022.08.060

EBIdnsFoam: solver for detailed and efficient simulation of reacting flows. https://vbt.ebi.kit.edu/594.php

Eckart, S., Pio, G., Zirwes, T., Zhang, F., Salzano, E., Krause, H., Bockhorn, H.: Impact of carbon dioxide and nitrogen addition on the global structure of hydrogen flames. Fuel **335**, 126929 (2022). https://doi.org/10.1016/j.fuel.2022.126929

Gärtner, J.W., Kronenburg, A., Martin, T.: Efficient WENO library for OpenFOAM. SoftwareX **12**, 100611 (2020)

Goodwin, D.G., Speth, R.L., Moffat, H.K., Weber, B.W.: Cantera: An Object-oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes. https://www.cantera.org. Version 2.5.1 (2021). https://doi.org/10.5281/zenodo.4527812

Grace, S.: XCIV Oscillatory motion of a viscous liquid in a long straight tube. Lon. Edinb. Dublin Philos. Mag. J. Sci. **5**(31), 933–939 (1928). https://doi.org/10.1080/14786440508564536

Habchi, C., Antar, G.: Direct numerical simulation of electromagnetically forced flows using OpenFOAM. Comput. Fluids **116**, 1–9 (2015). https://doi.org/10.1016/j.compfluid.2015.04.011

Häber, T., Zirwes, T., Roth, D., Zhang, F., Bockhorn, H., Maas, U.: Numerical simulation of the ignition of fuel/air gas mixtures around small hot particles. Z. Phys. Chem. **231**(10), 1625–1654 (2017). https://doi.org/10.1515/zpch-2016-0933

Haddad, K., Ertunç, Ö., Mishra, M., Delgado, A.: Pulsating laminar fully developed channel and pipe flows. Phys. Rev. E **81**(1), 016303 (2010). https://doi.org/10.1103/PhysRevE.81.016303

Hansinger, M., Zirwes, T., Zips, J., Pfitzner, M., Zhang, F., Habisreuther, P., Bockhorn, H.: The Eulerian stochastic fields method applied to large eddy simulations of a piloted flame with inhomogeneous inlet. Flow Turbul. Combust. **105**, 837–867 (2020). https://doi.org/10.1007/s10494-020-00159-5

Hassanaly, M., Koo, H., Lietz, C.F., Chong, S.T., Raman, V.: A minimally-dissipative low-Mach number solver for complex reacting flows in OpenFOAM. Comput. Fluids **162**, 11–25 (2018)

Hernández Pérez, F.E., Mukhadiyev, N., Xu, X., Sow, A., Lee, B.J., Sankaran, R., Im, H.G.: Direct numerical simulations of reacting flows with detailed chemistry using many-core/GPU acceleration. Comput. Fluids **173**, 73–79 (2018)

Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., Woodward, C.S.: SUNDIALS: suite of nonlinear and differential/algebraic equation solvers. ACM Transactions on Mathematical Software (TOMS) **31**(3), 363–396 (2005). https://doi.org/10.1145/1089014.1089020

Hosseini, S., Abdelsamie, A., Darabiha, N., Thévenin, D.: Low-Mach hybrid lattice Boltzmann-finite difference solver for combustion in complex flows. Phys. Fluids **32**(7), 077105 (2020)

Issa, R.I.: Solution of the implicitly discretised fluid flow equations by operator-splitting. J. Comput. Phys. **62**(1), 40–65 (1986). https://doi.org/10.1016/0021-9991(86)90099-9

Kee, R.J., Coltrin, M.E., Glarborg, P.: Chemically Reacting Flow: Theory and Practice. Wiley, Hoboken (2005). https://doi.org/10.1002/0471461296

Komen, E., Shams, A., Camilo, L., Koren, B.: Quasi-DNS capabilities of OpenFOAM for different mesh types. Comput. Fluids **96**, 87–104 (2014). https://doi.org/10.1016/j.compfluid.2014.02.013

Komen, E., Camilo, L., Shams, A., Geurts, B.J., Koren, B.: A quantification method for numerical dissipation in quasi-DNS and under-resolved DNS, and effects of numerical dissipation in quasi-DNS and under-resolved DNS of turbulent channel flows. J. Comput. Phys. **345**, 565–595 (2017)

Komen, E.M.J., Frederix, E.M.A., Coppen, T.H.J., D'Alessandro, V., Kuerten, J.G.M.: Analysis of the numerical dissipation rate of different Runge-Kutta and velocity interpolation methods in an unstructured collocated finite volume method in OpenFOAM®. Comput. Phys. Commun. **253**, 107145 (2020)

Lambossy, P.: Oscillations forcees d'un liquide incompressibile et visqueux dans un tube rigide et horizontal. Calcul de la force frottement Helv Physica Acta **25**, 371–386 (1952)

Law, C.K.: Combustion Physics. Cambridge University Press, Cambridge (2010)

Lecrivain, G., Rayan, R., Hurtado, A., Hampel, U.: Using quasi-DNS to investigate the deposition of elongated aerosol particles in a wavy channel flow. Comput. Fluids **124**, 78–85 (2016). https://doi.org/10.1016/j.compfluid.2015.10.012

Lee, S.B.: A study on temporal accuracy of OpenFOAM. Int. J. Naval Archit. Ocean Eng. **9**(4), 429–438 (2017)

Martin, T., Shevchuk, I.: Implementation and validation of semi-implicit WENO schemes using Open-FOAM®. Computation **6**(1), 6 (2018)

Martinez-Sanchis, D., Banik, S., Sternin, A., Sternin, D., Haidn, O., Tajmar, M.: Analysis of turbulence generation and dissipation in shear layers of methane-oxygen diffusion flames using direct numerical simulations. Phys. Fluids **34**(4), 045121 (2022)

Martínez-Sanchis, D., Sternin, A., Tagscherer, K., Sternin, D., Haidn, O., Tajmar, M.: Interactions between flame topology and turbulent transport in high-pressure premixed combustion. Flow Turbul. Combust. **109**(3), 813–838 (2022)

Moureau, V., Domingo, P., Vervisch, L.: Design of a massively parallel CFD code for complex geometries. Comptes Rendus Mécanique **339**(2–3), 141–148 (2011)

Noriega, H., Guibault, F., Reggio, M., Magnan, R.: A case-study in open-source CFD code verification, part I: convergence rate loss diagnosis. Math. Comput. Simul. **147**, 152–171 (2018)

OpenFOAM Data for the TGV benchmark suite for reacting flows (2023). https://github.com/g3bk47/TGV-OpenFOAM-benchmark

OpenFOAM: The open surce CFD toolbox. www.openfoam.com

Orszag, S.A.: Numerical simulation of the taylor-green vortex. In: Computing Methods in Applied Sciences and Engineering Part 2: International Symposium, Versailles, December 17–21, 1973, pp. 50–64 (1974). Springer

Peters, N.: The turbulent burning velocity for large-scale and small-scale turbulence. J. Fluid Mech. **384**, 107–132 (1999). https://doi.org/10.1017/S0022112098004212

Poinsot, T.J., Veynante, D.: Theoretical and Numerical Combustion. R.T. Edwards, New Jersey (2001)-10: 1930217056

Ray, S., Ünsal, B., Durst, F., Ertunc, Ö., Bayoumi, O.: Mass flow rate controlled fully developed laminar pulsating pipe flows. J. Fluids Eng (2005). https://doi.org/10.1115/1.1906265

Shu, C.-W., Don, W.-S., Gottlieb, D., Schilling, O., Jameson, L.: Numerical convergence study of nearly incompressible, inviscid Taylor-Green vortex flow. J. Sci. Comput. **24**, 1–27 (2005)

Smith, G.P., Golden, D.M., Frenklach, M., Moriarty, N.W., Eiteneer, B., Goldenberg, M., et al.: GRI 3.0 Reaction Mechanism (1999)

Soysal, M., Berghoff, M., Zirwes, T., Vef, M.-A., Oeste, S., Brinkmann, A., Nagel, W.E., Streit, A.: Using on-demand file systems in HPC environments. In: 2019 International Conference on High Performance Computing and Simulation (HPCS), pp. 390–398 (2019). IEEE. https://doi.org/10.1109/HPCS48598.2019.9188216

Steinhausen, M., Luo, Y., Popp, S., Strassacker, C., Zirwes, T., Kosaka, H., Zentgraf, F., Maas, U., Sadiki, A., Dreizler, A., Hasse, C.: Numerical investigation of local heat-release rates and thermochemical states in sidewall quenching of laminar methane and dimethyl ether flames. Flow Turbul. Combust. **106**, 681–700 (2020). https://doi.org/10.1007/s10494-020-00146-w

Steinhausen, M., Zirwes, T., Ferraro, F., Scholtissek, A., Bockhorn, H., Hasse, C.: Flame-vortex interaction during turbulent side-wall quenching and its implications for flamelet manifolds. Proc. Combust. Instit. (2022). https://doi.org/10.1016/j.proci.2022.09.026

Tufano, G.L., Stein, O.T., Kronenburg, A., Frassoldati, A., Faravelli, T., Deng, L., Kempf, A., Vascellari, M., Hasse, C.: Resolved flow simulation of pulverized coal particle devolatilization and ignition in

air- and O2/CO2 atmospheres. Fuel **186**, 285–292 (2016). https://doi.org/10.1016/j.fuel.2016.08.073

Tufano, G., Stein, O.T., Kronenburg, A., Gentile, G., Stagni, A., Frassoldati, A., Faravelli, T., Kempf, A., Vascellari, M., Hasse, C.: Fully-resolved simulations of coal particle combustion using a detailed multi-step approach for heterogeneous kinetics. Fuel **240**, 75–83 (2019). https://doi.org/10.1016/j.fuel.2018.11.139

Ünsal, B.: Time-dependent laminar, transitional and turbulent pipe flows. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) (2008). ISBN: 978-3-639-08075-9

Ünsal, B., Ray, S., Durst, F., Ertunç, Ö.: Pulsating laminar pipe flows with sinusoidal mass flux variations. Fluid Dyn. Res. **37**(5), 317 (2005). https://doi.org/10.1016/j.fluiddyn.2005.06.002

Van Rees, W.M., Leonard, A., Pullin, D.I., Koumoutsakos, P.: A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers. J. Comput. Phys. **230**(8), 2794–2805 (2011)

Vo, S., Kronenburg, A., Stein, O.T., Hawkes, E.R.: Direct numerical simulation of non-premixed syngas combustion using OpenFOAM. In: Nagel, W.E., Kröner, D.H., Resch, M.M. (eds.) High Performance Computing in Science and Engineering' 16, pp. 245–257. Springer, Berlin (2016). https://doi.org/10.1007/978-3-319-47066-5_17

Vo, S., Kronenburg, A., Stein, O.T., Cleary, M.: Multiple mapping conditioning for silica nanoparticle nucleation in turbulent flows. Proc. Combust. Inst. **36**(1), 1089–1097 (2017). https://doi.org/10.1016/j.proci.2016.08.088

Wang, B., Kronenburg, A., Dietzel, D., Stein, O.T.: Assessment of scaling laws for mixing fields in inter-droplet space. Proc. Combust. Inst. **36**(2), 2451–2458 (2017). https://doi.org/10.1016/j.proci.2016.06.036

Wang, B., Kronenburg, A., Tufano, G.L., Stein, O.T.: Fully resolved DNS of droplet array combustion in turbulent convective flows and modelling for mixing fields in inter-droplet space. Combust. Flame **189**, 347–366 (2018). https://doi.org/10.1016/j.combustflame.2017.11.003

Wang, Y., Zhang, H., Zirwes, T., Zhang, F., Bockhorn, H., Chen, Z.: Ignition of dimethyl ether/air mixtures by hot particles: impact of low temperature chemical reactions. Proc. Combust. Inst. **38**, 2459–2466 (2021). https://doi.org/10.1016/j.proci.2020.06.254

Wang, Y., Han, W., Zirwes, T., Zhang, F., Bockhorn, H., Chen, Z.: Effects of low-temperature chemical reactions on ignition kernel development and flame propagation in a DME-air mixing layer. Proc. Combust. Inst. (2022). https://doi.org/10.1016/j.proci.2022.07.024

Wang, Y., Han, W., Zirwes, T., Attili, A., Cai, L., Bockhorn, H., Yang, L., Chen, Z.: A systematic analysis of chemical mechanisms for ethylene oxidation and pah formation combustion and flame. Combust. Flame **253**, 112784 (2023). https://doi.org/10.1016/j.combustflame.2023.112784

Weller, H.G., Tabor, G., Jasak, H., Fureby, C.: A tensorial approach to computational continuum mechanics using object-oriented techniques. Comput. Phys. **12**(6), 620–631 (1998)

Zhang, F., Bonart, H., Zirwes, T., Habisreuther, P., Bockhorn, H., Zarzalis, N.: Direct numerical simulation of chemically reacting flows with the public domain code OpenFOAM. In: Nagel, W.E., Kröner, D.H., Resch, M.M. (eds.) High Performance Computing in Science and Engineering '14, pp. 221–236. Springer, Berlin (2015). https://doi.org/10.1007/978-3-319-10810-0_16

Zhang, F., Zirwes, T., Habisreuther, P., Bockhorn, H.: A DNS analysis of the correlation of heat release rate with chemiluminescence emissions in turbulent combustion. In: Nagel, W.E., Kröner, D.H., Resch, M.M. (eds.) High Performance Computing in Science and Engineering '16, pp. 229–243. Springer, Berlin (2016). https://doi.org/10.1007/978-3-319-47066-5_16

Zhang, F., Zirwes, T., Habisreuther, P., Bockhorn, H.: Effect of unsteady stretching on the flame local dynamics. Combust. Flame **175**, 170–179 (2017). https://doi.org/10.1016/j.combustflame.2016.05.028

Zhang, F., Zirwes, T., Nawroth, H., Habisreuther, P., Bockhorn, H., Paschereit, C.O.: Combustion-generated noise: an environment-related issue for future combustion systems. Energ. Technol. **5**(7), 1045–1054 (2017). https://doi.org/10.1002/ente.201600526

Zhang, F., Baust, T., Zirwes, T., Denev, J., Habisreuther, P., Zarzalis, N., Bockhorn, H.: Impact of infinite thin flame approach on the evaluation of flame speed using spherically expanding flames. Energ. Technol. **5**(7), 1055–1063 (2017). https://doi.org/10.1002/ente.201600573

Zhang, F., Zirwes, T., Habisreuther, P., Bockhorn, H., Trimis, D., Nawroth, H., Paschereit, C.O.: Impact of combustion modeling on the spectral response of heat release in LES. Combust. Sci. Technol. **191**(9), 1520–1540 (2019). https://doi.org/10.1080/00102202.2018.1558218

Zhang, F., Zirwes, T., Häber, T., Bockhorn, H., Trimis, D., Suntz, R.: Near wall dynamics of premixed flames. Proc. Combust. Inst. **38**(2), 1955–1964 (2021). https://doi.org/10.1016/j.proci.2020.06.058

Zhang, F., Zirwes, T., Häber, T., Bockhorn, H., Trimis, D., Suntz, R., Stapf, D.: Correlation of heat loss with quenching distance during transient flame-wall interaction. Proc. Combust. Inst. (2022). https://doi.org/10.1016/j.proci.2022.10.010

Zheng, E.Z., Rudman, M., Singh, J., Kuang, S.B.: Assessing OpenFOAM for DNS of turbulent non-Newtonian flow in a pipe. In: 21st Australasian Fluid Mechanics Conference, vol. 21, pp. 1–4 (2018). Australasian Fluid Mechanics Society

Zheng, E., Rudman, M., Singh, J., Kuang, S.: Direct numerical simulation of turbulent non-Newtonian flow using OpenFOAM. Appl. Math. Model. **72**, 50–67 (2019). https://doi.org/10.1016/j.apm.2019.03.003

Zhong, S., Peng, Z., Li, Y., Li, H., Zhang, F.: Direct numerical simulation of methane turbulent premixed oxy-fuel combustion. Technical report, SAE Technical Paper (2017). https://doi.org/10.4271/2017-01-2192

Zhu, J., Pan, J., Zhang, F., Zirwes, T., Ojo, A.O., Li, F.: Near-wall dynamics of premixed methane/air flames. Fuel **331**, 125774 (2023). https://doi.org/10.1016/j.fuel.2022.125774

Zirwes, T., Zhang, F., Denev, J.A., Habisreuther, P., Bockhorn, H.: Automated code generation for maximizing performance of detailed chemistry calculations in OpenFOAM. In: Nagel, W.E., Kröner, D.H., Resch, M.M. (eds.) High Performance Computing in Science and Engineering '17, pp. 189–204. Springer, Berlin (2017). https://doi.org/10.1007/978-3-319-68394-2_11

Zirwes, T., Zhang, F., Denev, J.A., Habisreuther, P., Bockhorn, H., Trimis, D.: Improved vectorization for efficient chemistry computations in OpenFOAM for large scale combustion simulations. In: Nagel, W.E., Kröner, D.H., Resch, M.M. (eds.) High Performance Computing in Science and Engineering '18, pp. 209–224. Springer, Berlin (2018). https://doi.org/10.1007/978-3-030-13325-2_13

Zirwes, T., Zhang, F., Häber, T., Bockhorn, H.: Ignition of combustible mixtures by hot particles at varying relative speeds. Combust. Sci. Technol. **191**(1), 178–195 (2019). https://doi.org/10.1080/00102202.2018.1435530

Zirwes, T., Zhang, F., Denev, J.A., Habisreuther, P., Bockhorn, H., Trimis, D.: Enhancing OpenFOAM's Performance on HPC Systems. In: Nagel, W.E., Kröner, D.H., Resch, M.M. (eds.) High Performance Computing in Science and Engineering '19, pp. 225–239. Springer, Berlin (2019). https://doi.org/10.1007/978-3-030-66792-4_16

Zirwes, T., Zhang, F., Habisreuther, P., Hansinger, M., Bockhorn, H., Pfitzner, M., Trimis, D.: Quasi-DNS dataset of a piloted flame with inhomogeneous inlet conditions. Flow Turbul. Combust. **104**, 997–1027 (2020). https://doi.org/10.1007/s10494-021-00244-3

Zirwes, T., Zhang, F., Habisreuther, P., Hansinger, M., Bockhorn, H., Pfitzner, M., Trimis, D.: Identification of flame regimes in partially premixed combustion from a quasi-DNS dataset. Flow Turbul. Combust. **106**(2), 373–404 (2021). https://doi.org/10.1007/s10494-020-00228-9

Zirwes, T., Zhang, F., Wang, Y., Habisreuther, P., Denev, J.A., Chen, Z., Bockhorn, H., Trimis, D.: In-situ flame particle tracking based on barycentric coordinates for studying local flame dynamics in pulsating bunsen flames. Proc. Combust. Inst. **38**, 2057–2066 (2021). https://doi.org/10.1016/j.proci.2020.07.033

Zirwes, T., Häber, T., Zhang, F., Kosaka, H., Dreizler, A., Steinhausen, M., Hasse, C., Stagni, A., Trimis, D., Suntz, R., et al.: Numerical study of quenching distances for side-wall quenching using detailed diffusion and chemistry. Flow Turbul. Combust. **106**(2), 649–679 (2021). https://doi.org/10.1016/S0360-1285(98)00026-4

Zirwes, T., Zhang, F., Denev, J.A., Habisreuther, P., Bockhorn, H., Trimis, D.: Implementation of Lagrangian Surface Tracking for High Performance Computing. In: Nagel, W.E., Kröner, D.H., Resch, M.M. (eds.) High Performance Computing in Science and Engineering '20, pp. 223–236. Springer, Berlin (2021). https://doi.org/10.1007/978-3-030-80602-6_15

Zirwes, T., Zhang, F., Bockhorn, H.: Memory effects of local flame dynamics in turbulent premixed flames. Proc. Combust. Inst. (2022). https://doi.org/10.1016/j.proci.2022.07.187

## Authors and Affiliations

**Thorsten Zirwes[1,2,3]** · **Marvin Sontheimer[3]** · **Feichi Zhang[2,4]** ·
**Abouelmagd Abdelsamie[5]** · **Francisco E. Hernández Pérez[6]** · **Oliver T. Stein[2,3]** ·
**Hong G. Im[6]** · **Andreas Kronenburg[3]** · **Henning Bockhorn[2]**

✉ Thorsten Zirwes
thorsten.zirwes@kit.edu

Marvin Sontheimer
marvin.sontheimer@itv.uni-stuttgart.de

Feichi Zhang
feichi.zhang@kit.edu

Abouelmagd Abdelsamie
abouelmagd.abdelsamie@ovgu.de

Francisco E. Hernández Pérez
francisco.hernandezperez.1@kaust.edu.sa

Oliver T. Stein
oliver.t.stein@kit.edu

Hong G. Im
hong.im@kaust.edu.sa

Andreas Kronenburg
andreas.kronenburg@itv.uni-stuttgart.de

Henning Bockhorn
henning.bockhorn@kit.edu

[1]   Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany

[2]   Engler-Bunte-Institute, Karlsruhe Institute of Technology, Engler-Bunte-Ring 7, 76131 Karlsruhe, Germany

[3]   Institute for Combustion Technology, University of Stuttgart, Pfaffenwaldring 31, 70569 Stuttgart, Germany

[4]   Institute for Technical Chemistry, Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany

[5]   Laboratory of Fluid Dynamics & Technical Flows, University of Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany

[6]   Clean Combustion Research Center (CCRC), King Abdullah University of Science and Technology, 4700 King Abdullah University of Science and Technology, Thuwal 23955-6900, Kingdom of Saudi Arabia