



Towards direct numerical simulations of low-Mach number turbulent reacting and two-phase flows using immersed boundaries



Abouelmagd Abdelsamie^{a,*}, Gordon Fru^b, Timo Oster^{a,c}, Felix Dietzsch^d, Gábor Janiga^a, Dominique Thévenin^a

^a Lab. of Fluid Dynamics & Technical Flows, University of Magdeburg, Otto von Guericke, Germany

^b GexCon, Norway

^c Visual Computing Group at University of Magdeburg, Germany

^d Lab. of Numerical Thermo-fluid dynamics, TU Bergakademie Freiberg, Germany

ARTICLE INFO

Article history:

Received 15 October 2015

Revised 2 February 2016

Accepted 15 March 2016

Available online 16 March 2016

Keywords:

High order DNS

Immersed boundary methods

On-the-fly analysis

Low Mach number reactive flow

Spray

ABSTRACT

Most reacting and two-phase flows of practical interest are turbulent but take place at low Mach numbers or under incompressible conditions. In order to investigate the properties of such complex flows with high accuracy but acceptable computing times, a suitable tool for Direct Numerical Simulations (DNS), called DINOSOARS, has been developed. The present article describes the numerical components and methods implemented in this code, together with a detailed verification and validation phase, and finishes with two examples of full-scale simulations. We hope it might be useful as a “verification and validation guideline” for other researchers working on DNS of reacting flows. Since applications of growing complexity are considered by DNS, a Direct Boundary Immersed Boundary Method (DB-IBM) has been implemented, allowing a description of arbitrary geometries on a fixed, but possibly refined, Cartesian mesh. A direct force IBM is implemented as well in DINOSOARS in order to resolve large moving spherical particles (much larger than the Kolmogorov scale) on the grid. Particles below the Kolmogorov scale are treated as point particles, taking into account additionally heat and mass transfer with the continuous flow. The efficient parallelization of the code relies on the open-source library 2DECOMP&FFT. The underlying Poisson equation is solved in a fast and accurate manner by FFT, even for non-periodic boundary conditions. The flexibility of DINOSOARS makes it a very promising tool for analyzing a variety of problems and applications involving turbulent reacting and/or two-phase flows.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

The current paper describes the third-generation DNS code developed at the University of Magdeburg, building on top of the experience of our group, first with the PARCOMB family [1,2], then with the code called π^3 [3,4]. Originally, the name of DINOSOARS stood for “Direct Numerical, high-Order Simulation and On-the-fly Analysis of Reacting flows and Sprays,” though solid particles interacting with turbulence are now considered as well.

DNS of gaseous reactive flows has a long history of success over the last three decades, as documented for instance in [5–12]. Nevertheless, it remains a hot research topic leading steadily to new publications, e.g., [13–16]], since many issues remain unsolved for this challenging problem involving extremely high

computational requirements. Besides considering pure gaseous flames, reacting sprays have also been considered extensively by DNS, see for instance [17,18].

Our research group has always specialized in DNS studies taking into account detailed models to describe kinetic processes. As discussed later in this paper, considering complex molecules such as *n*-heptane or ethylene, further increases the computational burden in comparison to non-reacting flows or to approximations relying on single-step chemistry. To add to this challenge, such studies only make sense when describing with a similar level of accuracy all relevant thermodynamic and molecular transport properties, such as diffusion coefficients and viscosity. In the end, the resulting systems can only be solved on parallel supercomputers.

Finally, there is a growing need for applying DNS to configurations involving semi-complex geometries, while keeping a very low numerical dissipation. This has been answered in DINOSOARS by implementing Immersed Boundaries.

* Corresponding author. Tel.: +493916712427.

E-mail address: abouelmagd.abdelsamie@ovgu.de (A. Abdelsamie).

Considering the challenges that must be met and the process conditions relevant for practical purposes, three strategies have been combined in DINOSOARS to keep acceptable computational times for DNS of reacting and/or two-phase flows

- Considering that most processes involve incompressible flows or low Mach numbers, only these two configurations have been considered. A large speed-up can be obtained by using a low-Mach number solver compared to a fully compressible one [19].
- The previous approach removes timestep constraints associated with acoustic waves. For reacting flows, the timestep then becomes classically controlled by fast kinetic processes. In order to solve this issue and obtain a stable, high-order integration in time, a semi-implicit time integration has been successfully implemented.
- The third, classical strategy is to rely heavily on parallelization. For this purpose, coupling DINOSOARS with a recent open-source solution, 2DECOMP&FFT, was found to be very efficient. However, the major issue for incompressible and low-Mach solvers is to implement a fast parallel solver for the Poisson equation coupling pressure and velocity. For this purpose, it was finally possible to develop and implement an innovative FFT-based approach, even for non-periodic boundary conditions, so that even this bottleneck could be released.

While the direct simulation is notably very difficult in itself, the analysis of the obtained results is also very challenging, in particular when considering that it is impossible to store all variable values at all grid points for all timesteps. Therefore, an on-the-fly analysis is implemented as well in DINOSOARS, as discussed later.

This paper is organized as follows. After introducing the governing equations and the physicochemical parameters appearing in these equations, all models and algorithms implemented in DINOSOARS for solving the resulting equation system are presented, including parallelization issues. Then, code performance is quantified and a systematic verification is presented. Finally, a successful validation of the code is proposed, opening the door for two exemplary, full-scale simulations.

2. Governing equations

DINOSOARS has been conceived from the start as a flexible tool, allowing for a highly accurate investigation of turbulent flows at low and intermediate velocities. Thus, it involves a variety of algorithms, models and of equations. Due to space limitations, it is impossible to discuss everything in great detail. Only the most important aspects will be described in what follows, separating in particular between reacting and non-reacting flows. The low-Mach number approach is used in DINOSOARS for reacting flows, since most applications involving chemical reactions and combustion indeed take place at low Mach numbers, $Ma \ll 1$ [20]. For non-reacting flows, incompressible transport equations are considered instead. For both conditions (reacting and non-reacting), either single-phase or two-phase turbulent flows can be described with DINOSOARS, as described later. Since DINOSOARS will rely on external libraries written in dimensional form, it has been written from the start as a dimensional code as well, using SI units.

3. Low-Mach number formulation for reacting flows

In this case, the pressure in the flow is nearly uniform and the coupling between the fluctuating pressure and the density can be neglected. The present formulation is based on an incompressible but dilatable approach, first described for reacting flows by [21]. Pressure is then split into a spatially homogeneous thermodynamic pressure $p(t)$ and a dynamic fluctuating pressure $\tilde{p}(x, t)$, where $\tilde{p}(x, t) \ll p(t)$. The resulting equations still contain vorticity and

entropy waves, but acoustic waves have disappeared. Therefore, the timestep is no longer limited by acoustic times, but only by characteristic convection, diffusion or reaction time scales, leading to a considerable speed-up [19].

The conservations equations underlying the low-Mach number approximation for an ideal gas involving N_s chemical components can be summarized in compact form as follows:

$$\partial_t(\rho) = -\partial_i(\rho u_i), \quad (1)$$

$$\partial_t(\rho u_i) = -\partial_i \tilde{p} + R_{m,i}, \quad (2)$$

$$\partial_t(T) = \vartheta + R_T, \quad (3)$$

$$\partial_t(Y_k) = \mathfrak{S}_k + R_{Y_k}, \quad (4)$$

$$\rho = \frac{pW}{RT}, \quad (5)$$

subject to the additional condition for global mass conservation

$$\sum_{k=1}^{N_s} Y_k = 1, \quad (6)$$

where ρ , u_i , \tilde{p} , p , T , Y_k , N_s , R and W are the mixture density, i th component of flow velocity, fluctuation pressure, thermodynamic pressure, temperature, k th species mass fraction, number of species, ideal gas constant, and mixture mean molecular weight, respectively. The right-hand side of the momentum equation, Eq. (2), reads using the summation convention of Einstein,

$$R_{m,i} = -\frac{\partial(\rho u_j u_i)}{\partial x_j} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] + \frac{\partial}{\partial x_j} \left[\left(\kappa - \frac{2}{3} \mu \right) \frac{\partial u_i}{\partial x_j} \right] \delta_{ij} + \rho \sum_{k=1}^{N_s} Y_k f_{k,i}, \quad (7)$$

where δ_{ij} , $f_{k,i}$, μ and κ are the Kronecker delta, components of the specific volume force acting on species k , dynamic and volume viscosity, respectively. The right-hand side of the temperature equation, Eq. (3), has been split into a stiff term, ϑ , and a non-stiff term, R_T , respectively:

$$\vartheta = -\frac{1}{\rho C_p} \sum_{k=1}^{N_s} h_k \dot{\omega}_k, \quad (8)$$

$$R_T = -u_j \frac{\partial T}{\partial x_j} + \frac{1}{\rho C_p} \left[\frac{\partial}{\partial x_j} \left(\lambda \frac{\partial T}{\partial x_j} \right) - \frac{\partial T}{\partial x_j} \sum_{k=1}^{N_s} \rho C_{p,k} Y_k V_{k,j} + \sum_{k=1}^{N_s} \rho Y_k V_{k,j} f_{k,j} \right]. \quad (9)$$

In Eqs. (8) and (9), C_p , h_k , $\dot{\omega}_k$, λ , $V_{k,j}$ represent the specific heat capacity at constant pressure, specific enthalpy, mass reaction rate, heat diffusion coefficient and j th component of the species molecular diffusion velocity, respectively. The stiff term ϑ involves the reaction rates induced by chemical kinetics and will be integrated in time separately from R_T , leading to a much faster and more robust integration.

In a similar manner, the right-hand side of the conservation equation for the mass fraction of species k , Y_k (Eq. 4), has been split into a stiff term, \mathfrak{S}_k , and a non-stiff term, R_{Y_k} , respectively:

$$\mathfrak{S}_k = -\frac{\dot{\omega}_k}{\rho}, \quad (10)$$

$$R_{Y_k} = -u_j \frac{\partial Y_k}{\partial x_j} - \frac{1}{\rho} \frac{\partial(\rho Y_k V_{k,j})}{\partial x_j}. \quad (11)$$

3.1. Diffusion velocity

To solve the system involving Eqs. (9) and (11), the molecular diffusion velocity of species k , \mathbf{V}_k , must be modeled appropriately. In order to increase flexibility, three different approaches are implemented in DINOSOARS, leading to increasing accuracy but also computational requirements [22],

1. Unity Lewis numbers;
2. Mixture-averaged diffusion velocities;
3. Multicomponent diffusion velocities.

Additionally, thermal diffusion (Soret term) may be considered for light species, but without the inverse (Dufour) effect. Diffusion due to pressure gradients and to external forces is not relevant for the considered applications and is thus not accounted for.

3.1.1. Unity Lewis numbers

In that case, it is assumed that the Lewis number of species k , $Le_k = \lambda / (\rho C_p D_k)$ are all identical and equal to unity, meaning that the molecular diffusion velocity of all species is equal to the heat diffusion velocity. Then, the diffusion coefficient of species k , D_k , can be immediately estimated as $D_k = \lambda / (\rho C_p)$ and is identical for all species, $D_k = D$. The diffusion velocity finally reads:

$$\mathbf{V}_k = -D \frac{\nabla X_k}{X_k}, \quad (12)$$

with X_k the mole fraction of species k .

3.1.2. Mixture-averaged diffusion coefficient

Here, following the classical approximation proposed by Hirschfelder and Curtiss, a mixture-averaged diffusion coefficient D_k is first computed for each species k using,

$$D_k = \frac{1 - Y_k}{\sum_{j \neq k} X_j / D_{jk}}, \quad (13)$$

where the D_{jk} are the binary diffusion coefficients. Then, the diffusion velocity is determined by

$$\mathbf{V}_k = -D_k \frac{\nabla X_k}{X_k} + \sum_{k=1}^{N_s} D_k \nabla X_k \frac{W_k}{W}, \quad (14)$$

The last term in this equation is the correction velocity needed to fulfill mass conservation by molecular diffusion, i.e., $\sum_{k=1}^{N_s} Y_k \mathbf{V}_k = 0$, with W_k the atomic weight of species k .

3.1.3. Multicomponent diffusion velocity

In many applications the combustion process can not be reduced to a representation of a simple binary mixture. Therefore, the formulation of a species transport law has to account for different transport properties of each species. In general, the multicomponent diffusion process is computed by inverting the species linear transport system [23],

$$\mathbf{F} = \mathbf{L}^{00,00^{-1}}, \quad (15)$$

with

$$L_{ij}^{00,00} = \frac{16T}{25p} \sum_{k=1}^{N_s} \frac{X_k}{W_i D_{ik}} \{W_j X_j (1 - \delta_{ik}) - W_i X_j (\delta_{ij} - \delta_{jk})\}. \quad (16)$$

The multicomponent diffusion coefficients are then given by

$$D_{ij} = X_i \frac{16T}{25p} \frac{W}{W_j} (F_{ij} - F_{ii}). \quad (17)$$

Finally, the species diffusion velocities read

$$\mathbf{V}_k = -\frac{1}{X_k W} \sum_{j=1}^{N_s} W_j D_{kj} \nabla X_j. \quad (18)$$

In order to evaluate efficiently all multicomponent transport properties, the EGLib library [23,24] in its version 3.4 has been coupled to DINOSOARS.

3.1.4. Thermal diffusion

Thermal diffusion (Soret effect) is implemented as an additional term that might be switched on by decision of the user. The corresponding diffusion velocity reads

$$\mathbf{V}_{kT} = -D_{kT} \frac{\nabla T}{T}. \quad (19)$$

Considering that the corresponding computation of D_{kT} is relatively expensive and that this term is only important for light molecular species, the Soret effect is usually only activated for species H and H₂.

3.2. Further transport coefficients

Additional transport coefficients appear in the equations listed above. The computation of the heat diffusion coefficient λ and of the dynamic viscosity μ relies by default on Cantera 1.8, or alternatively on EGLib 3.4. The EGLib 3.4 library is always employed to compute the volume viscosity κ appearing in Eq. (7), whenever needed [25].

3.3. Thermodynamic parameters

All thermodynamic parameters appearing in the previous conservation equations, such as $C_{p,k}$ and h_k are computed by coupling DINOSOARS with Cantera 1.8.

3.4. Chemical kinetics

The chemical source terms $\dot{\omega}_k$ appearing in the conservation equations for species are computed by Cantera 1.8 based on detailed reaction schemes from the literature, solving one transport equation for each chemical species appearing in the system. For instance, for one application presented at the end of this paper (a turbulent ethylene flame), 51 transport equations are taken into account; 46 equations for the 46 species of the reaction mechanism, 3 for momentum, 1 for temperature and 1 Poisson equation (using the equation of state to close the system).

4. Incompressible formulation for non-reacting flows

For non-reacting flows at low Mach numbers, it is better to use fully incompressible flow equations, reducing to mass and momentum conservation, since energy conservation need not be considered separately. This leads to a faster integration and to a reduction of the memory requirements. Mass conservation reads simply

$$\frac{\partial u_i}{\partial x_i} = 0, \quad (20)$$

while the right-hand side of the momentum equation, Eq. (7), reduces to,

$$R_{m,i} = -\frac{\partial(\rho u_j u_i)}{\partial x_j} + \mu \frac{\partial^2 u_i}{\partial x_j^2} + \rho f_i, \quad (21)$$

for a constant dynamic viscosity μ , where f is the specific volume force acting on the fluid.

5. Algorithms

5.1. Parallelization

The parallelization of the code relies on a 2D pencil decomposition using the open-source library 2DECOMP&FFT [26], which has been designed to perform optimized three-dimensional distributed FFTs. This library is called on top of standard MPI and MPI-I/O libraries and includes user-friendly programming interfaces. 2DECOMP&FFT supports large-scale parallel applications on distributed

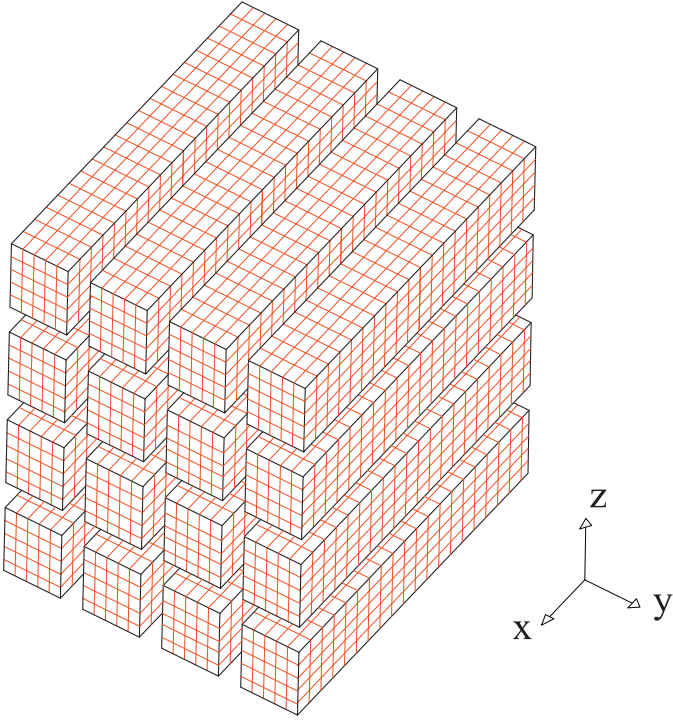


Fig. 1. 2D domain decomposition with x-pencil orientation.

memory systems and shows excellent performance on a variety of existing supercomputers [27,28]. The library comprises two different possible pencil orientations, x-pencils and z-pencils. DINOSOARS has been coded with the x-pencil orientation (Fig. 1) in order to minimize the computational time needed for transposing the data as required for non-periodic boundary conditions. Using 2DECOMP&FFT an excellent parallel scaling has been obtained with DINOSOARS, as discussed later in Section 6.1.

5.2. Time integration

In order to increase flexibility, three different time integration algorithms have been coded in DINOSOARS. The final choice of the user should be based on the considered application, on the required accuracy, on the available computational resources and, most important, on the stiffness of the problem, in particular for reacting flows.

The first algorithm is a fully explicit, fourth-order Runge-Kutta method rewritten as a low-storage algorithm [29]. It is the simplest and computationally most efficient algorithm. It is activated by default for non-reacting cases. However, it suffers from a limited stability for stiff applications. The solution procedure reads

$$T^{n+1} = T^n + \Delta t (\vartheta^n + R_T^n), \quad (22)$$

$$Y_k^{n+1} = Y_k^n + \Delta t (\vartheta_k^n + R_{Y_k}^n), \quad (23)$$

$$\rho^{n+1} = \frac{pW^{n+1}}{RT^{n+1}}, \quad (24)$$

$$(\rho u_i)^* = (\rho u_i)^n + \Delta t R_{m,i}^n, \quad (25)$$

$$\nabla^2 \tilde{p}^{n+1/2} = \frac{1}{\Delta t} [\partial_t \rho^{n+1} + \partial_j (\rho u_j)^*], \quad (26)$$

$$(\rho u_i)^{n+1} = (\rho u_i)^* - \Delta t \partial_i \tilde{p}^{n+1/2}. \quad (27)$$

As documented in Eqs. (25)–(27), the coupling between continuity and momentum equation is handled by using the pressure-free projection method [30–32]. In this approach the intermediate quantity $(\rho u)^*$ is obtained by solving the momentum equation without pressure gradient (Eq. 25). Then, applying the divergence operator to Eq. (27) leads to

$$\nabla^2 \tilde{p}^{n+1/2} = \frac{1}{\Delta t} [\partial_i (\rho u_i)^* - \partial_i (\rho u_i)^{n+1}]. \quad (28)$$

The Poisson equation is closed by substituting the continuity equation, Eq. (1), into Eq. (28), leading to the continuity constraint. In Eq. (26), a third-order backward finite-difference approximation is used to compute $\partial_t \rho^{n+1}$.

The second algorithm for time integration is based on the split semi-implicit fourth-order Runge–Kutta scheme. In this algorithm the right-hand side of the governing equations for temperature and species are split into non-stiff parts (R_T , R_k) and stiff parts (ϑ , ϑ_k), following [33,34]. The non-stiff parts of T and Y_k are solved using the explicit fourth-order Runge–Kutta solver

$$T^* = T^n + \Delta t (R_T^n), \quad (29)$$

$$Y_k^* = Y_k^n + \Delta t (R_{Y_k}^n), \quad (30)$$

$$\rho^* = \frac{pW^*}{RT^*}. \quad (31)$$

The stiff parts are integrated in time by using the implicit solver RADAU-5 [35,36]. Then, both solutions are added

$$T^{n+1} = T^* + \Delta t (\vartheta^{n+1}), \quad (32)$$

$$Y_k^{n+1} = Y_k^* + \Delta t (\vartheta_k^{n+1}), \quad (33)$$

$$\rho^{n+1} = \frac{pW^{n+1}}{RT^{n+1}}. \quad (34)$$

Finally, momentum conservation and Poisson equation are again integrated using the explicit fourth-order Runge–Kutta method, as explained above (Eqs. 25–27).

The third possibility for time integration is to use the additive third-order Runge–Kutta method. This non-split semi-implicit Runge–Kutta algorithm is an extension of the Rosenbrock Runge–Kutta approach. In this algorithm the system of equations is solved as follows:

$$d_t \mathbf{Z} = \mathbf{f}(\mathbf{Z}) + \mathbf{g}(\mathbf{Z}) \quad (35)$$

$$\partial_t (\rho \mathbf{u}) = \mathbf{f}(\rho \mathbf{u}) - \nabla \tilde{p} \quad (36)$$

where \mathbf{f} and \mathbf{g} are the non-stiff and stiff right-hand side vectors of the equation system, respectively, while \mathbf{Z} is the vector containing the scalar variables $(T, Y_1, Y_2, \dots, Y_{N_s})'$. Then, the solution is obtained through following steps:

$$\mathbf{Z}^{n+1} = \mathbf{Z}^n + \sum_{j=1}^r \alpha_j \mathbf{K}_j, \quad (37)$$

$$\rho^{n+1} = \frac{pW^{n+1}}{RT^{n+1}}, \quad (38)$$

$$(\rho \mathbf{u})^* = (\rho \mathbf{u})^n + \sum_{j=1}^r \alpha_j \mathbf{K}_{u_j}, \quad (39)$$

$$\nabla^2 \tilde{p}^{n+1/2} = \frac{1}{\Delta t} [\partial_t \rho^{n+1} + \nabla(\rho \mathbf{u})^*], \tag{40}$$

$$(\rho \mathbf{u})^{n+1} = (\rho \mathbf{u})^* - \Delta t \nabla \tilde{p}. \tag{41}$$

Where,

$$\left[\mathbf{I} - \Delta t a_i \mathbf{J} \left(\mathbf{Z}^n + \sum_{j=1}^{i-1} (d_{ij} \mathbf{K}_{z_j}) \right) \right] \mathbf{K}_{z_i} = \Delta t \left[\mathbf{f} \left(\mathbf{Z}^n + \sum_{j=1}^{i-1} (b_{ij} \mathbf{K}_{z_j}) \right) + \mathbf{g} \left(\mathbf{Z}^n + \sum_{j=1}^{i-1} (c_{ij} \mathbf{K}_{z_j}) \right) \right], \tag{42}$$

and

$$\mathbf{K}_{u_i} = \Delta t \left[\mathbf{f} \left((\rho \mathbf{u})^n + \sum_{j=1}^{i-1} (c_{ij} \mathbf{K}_{u_j}) \right) \right], \forall i = 1, 2, \dots, r. \tag{43}$$

In Eqs. (37)-(43), r is the number of stages (in DINOSOARS, $r = 3$ by default, leading to third order in time) and $\mathbf{J} = \partial \mathbf{g} / \partial \mathbf{Z}$ is the Jacobian matrix of the stiff term, \mathbf{g} . The symbols α_j , a_i , b_{ij} , c_{ij} and d_{ij} designate constant coefficients. The values of these coefficients depend on the retained order. The values used in DINOSOARS are listed in Table 1 and correspond to the third-order case. More details about the method and its accuracy can be found in [37].

Independently from the retained integration procedure, the timestep of the computation can be optionally controlled by three different limiters, separately or in combination: 1) the Courant-Friedrichs-Lewy (CFL) stability criterion, 2) the Fourier stability criterion for diffusive terms, and 3) a dynamic accuracy control obtained through timestep-doubling within the Runge-Kutta procedure, as already used in past DNS codes [1,29].

5.3. Spatial discretization

All partial derivatives in space appearing in the conservation equations are discretized using centered, sixth-order finite differences (seven-point stencil), stepwise reduced down to third-order near domain boundaries. The communication across processor boundaries is handled by using the halo-cell communication function included in the 2DECOMP&FFT library.

5.4. Solving the Poisson equation in parallel

One of the most difficult issues associated with any low-Mach or incompressible flow solver is to find an efficient way for solving the Poisson equation. Usually, it is solved by explicit iterative methods (Conjugate Gradient CG, Multi-grid, etc.) or more rarely by implicit methods (matrix inversion, spectral methods, or combinations of both). In DINOSOARS, the Poisson equation is solved by FFT, even when the boundary conditions of the domain are not periodic.

The developed approach is an extension of that described in [27,28,38,39], where pre- and post-processing were applied both in physical and in wave space. The current algorithm needs only pre- and post-processing in the physical space and for a different purpose. Additional differences result from the fact that the pressure is solved in DINOSOARS in a collocated manner and not on a staggered grid.

Suitable pre- and and post-processing steps are applied to the corresponding array before and after calling the parallel FFT subroutine included in 2DECOMP&FFT. All tests have demonstrated that this method delivers a very high order (equivalent spectral accuracy) and is also computationally very efficient compared to existing alternatives. In order to explain the current algorithm it is

better to start with classical discrete Fourier transform (\hat{F}_k) for a real sequence F_j , $j = 0, 1, 2, \dots, N - 1$, which is defined by,

$$\hat{F}_k = \frac{1}{N} \sum_{j=0}^{N-1} F_j e^{-2\pi j k \sqrt{-1}/N}, \quad k = 0, 1, \dots, N - 1. \tag{44}$$

Thanks to Hermitian symmetry, $\hat{F}_k = \hat{F}_{n-k}^*$, with $\hat{F}_N = \hat{F}_0$. The inverse of this transform reads

$$F_j = \sum_{k=0}^{N-1} F_k e^{2\pi j k \sqrt{-1}/N}. \tag{45}$$

This operation is directly applied only for periodic sequences (periodic boundary conditions). These two transforms can then be obtained immediately with 2DECOMP&FFT and FFTW libraries, using FFT and IFFT algorithms, respectively.

Concerning now the implementation in DINOSOARS, F_j is first transformed in case of Dirichlet-Dirichlet (DD) boundary conditions using a discrete sine transform (DST),

$$\hat{F}_k = \frac{2}{N} \sum_{j=1}^{N-1} F_j \sin(\pi j k / N). \tag{46}$$

In order to obtain the Fourier transform with the standard FFT parallel routines included in 2DECOMP&FFT, the F_j array is extended in a pre-processing step to a temporary, odd symmetry sequence with length of $(2N)$, in the form $(0, F_1, F_2, \dots, F_{N-1}, 0, -F_{N-1}, \dots, -F_2, -F_1)$, where $F_j = -F_{2N-j}$ for $j = 1, N - 1$.

In the same manner, in case of Neumann-Neumann (NN) boundary conditions, a discrete cosine transform (DCT) is used instead

$$\hat{F}_k = \frac{2}{N} \left[\frac{F_0}{2} + \sum_{j=1}^{N-1} F_j \cos(\pi j k / N) + \frac{(-1)^k F_N}{2} \right], \tag{47}$$

The standard FFT routine is applied after extending the array in a pre-processing step into a temporary, even symmetry sequence of length $(2N)$ with the form of $(F_0, F_1, F_2, \dots, F_{N-1}, F_N, F_{N-1}, \dots, F_2, F_1)$, where $F_j = F_{2N-j}$ for $j = 1, N - 1$.

A combination between both boundary conditions is also possible. In the case of a Dirichlet-Neumann (DN) combination, a quarter-wave discrete sine transform (QW-DST) is suitable

$$\hat{F}_k = \frac{1}{N} \left[\frac{F_0}{2} + \sum_{j=1}^{N-1} F_j \sin(\pi j (2k - 1) / (2N)) + \frac{(-1)^{k-1} F_N}{2} \right], \tag{48}$$

Then, a classical FFT routine is again possible, after extending the original sequence to a temporary, odd symmetry sequence with length of $(4N)$ where $(0, F_1, \dots, F_N, F_{N-1}, \dots, F_1, 0, -F_1, \dots, -F_N, -F_{N-1}, \dots, -F_1)$. Similarly, a case with Neumann-Dirichlet (ND) boundary conditions is now transformed with a quarter-wave discrete cosine transform (QW-DCT),

$$\hat{F}_k = \frac{1}{N} \left[\frac{F_0}{2} + \sum_{j=1}^{N-1} F_j \cos(\pi j (2k - 1) / (2N)) \right], \tag{49}$$

with a standard FFT routine after extending the original sequence to a temporary, even symmetry sequence with length of $(4N)$, in the form $(F_0, F_1, \dots, F_{N-1}, 0, -F_{N-1}, \dots, -F_0, -F_1, \dots, -F_{N-1}, 0, F_{N-1}, \dots, F_1)$.

Finally, the algorithm implemented in DINOSOARS for solving the Poisson equation $\nabla^2 p = F$ can be summarized as follows:

1. Pre-processing for sequence F (input to the algorithm), which is a real array of length N , extending its length depending on the boundary conditions;
 - DD: $M = 2N, (0, F_1, F_2, \dots, F_{N-1}, 0, -F_{N-1}, \dots, -F_2, -F_1)$;
 - NN: $M = 2N, (F_0, F_1, F_2, \dots, F_{N-1}, F_N, F_{N-1}, \dots, F_2, F_1)$;

Table 1
Coefficients for third-order additive semi-implicit Runge–Kutta integration.

(a_i)			$(b_{ij} = d_{ij})$			(c_{ij})			(α_j)	
a_1	a_2	a_3	b_{21}	b_{31}	b_{32}	c_{21}	c_{31}	c_{32}	$\alpha_1 = \alpha_2$	α_3
0.797097	0.591381	0.134705	$\frac{8}{7}$	$\frac{71}{252}$	$\frac{7}{36}$	1.05893	0.5	-0.375939	$\frac{1}{8}$	$\frac{3}{4}$

- DN: $M = 4N$, $(0, F_1, \dots, F_N, F_{N-1}, \dots, F_1, 0, -F_1, \dots, -F_N, -F_{N-1}, \dots, -F_1)$;
 - ND: $M = 4N$, $(F_0, F_1, \dots, F_{N-1}, 0, -F_{N-1}, \dots, -F_0, -F_1, \dots, -F_{N-1}, 0, F_{N-1}, \dots, F_1)$.
2. Apply standard FFT routine (Eq. 44) over M discrete points to obtain \hat{f}_k ;
 3. Solve the Poisson equation in wave space, $\hat{p} = -\hat{f}/\kappa^2$;
 4. Apply standard IFFT routine (Eq. 45) to Fourier transform of the pressure (\hat{p}), obtaining the pressure in the physical space saved in temporary array F (overwritten to save memory);
 5. Post-processing for array F by saving the correct part into an array P of length N .

This algorithm has been coded for parallel simulations (parallel FFT) to speed-up the process. Considering for instance a small 3D DNS involving $65 \times 65 \times 32$ grid points parallelized using 16 cores with Dirichlet–Neumann, Neumann–Neumann, and periodic-periodic boundary conditions in x , y , and z direction, respectively, the implemented algorithm is already 6.75 faster than the CG solver implemented in the well-known HYPRE library [40].

5.5. Boundary conditions in time and space

For DNS relying on the low-Mach number or on the incompressible formulation, boundary conditions in space are straightforward [3], at the difference of compressible DNS tools [41]. Standard Dirichlet and Neumann boundary conditions perform well and are stable.

Concerning initial conditions in time, a suitable approximation of real turbulence must usually be generated for DNS. It is then either used directly for time-decaying simulations in the same domain, or it is employed for turbulence injection through an inflow boundary, leading to spatially-evolving turbulence. Both solutions are available in DINOSOARS.

To generate the initially needed approximation of the turbulent flow field, four different algorithms are available in the code, so that initial solutions can be obtained for a broad range of integral length scales, turbulence intensities and geometrical configurations. By default, an Inverse Fast Fourier Transform (IFFT) relying either on the Passot–Pouquet spectrum or on the Von Kármán spectrum with Pao correction [1] (see later Eq. 68) is implemented in parallel using 2DECOMP&FFT. As alternatives, the Kraichnan technique [42], digital filtering [43], or random noise diffusion [44] are available as well.

5.6. Immersed boundaries

Using two different formulations, the Immersed Boundary Method (IBM) is implemented within DINOSOARS. The first one pertains to static immersed boundaries, that will not change in time; this is particularly useful to carry out DNS in complex geometries. Considering now moving particles (solid particles, droplets) resolved on the grid, the second usage of IBM is to describe two-phase flows. Both approaches are described in more detail in the next two sections.

5.6.1. DNS in complex geometries

By default, a regular tensor-product grid is generated by DINOSOARS. If desirable for the considered application, a static re-

finement can be applied based on stretching functions close to the boundaries (for a better resolution of the boundary layer) or at any place within the domain (e.g., for a better resolution of a mixing region). When applying a static grid refinement, a numerical transformation is applied based on the chain rule. Hence, the derivatives are still computed on a regularly-spaced computational grid, and then projected onto the real, adapted grid [45–47]. The transformation function must only be computed once, since it does not change. In that case, it is much easier to keep the equidistant formulation of the derivative computation, and multiply afterwards with the (fixed) transformation function. Tests have shown that it is also slightly faster in terms of computing time.

When considering now cases with a complex (but currently time-independent) geometry, DINOSOARS relies on the Direct Boundary Immersed Boundary Method (DB-IBM). The details of the algorithm can be found in [48,49]. For this purpose, DINOSOARS reads the discretized input geometry in 3D as binary values transformed into a boolean variable describing if the corresponding point is within the wall (value: 0) or within the flow (value: 1). In this manner, even very complex and irregular geometries can be considered in the DNS simulation. Note that the spectral Poisson solver described previously in Section 5.4 is compatible with all other features implemented in the code, like those described here.

As an example, Fig. 2 shows a patient-specific cerebral aneurysm [50] represented in DINOSOARS by using DB-IBM, and instantaneous streamlines obtained in this geometry at peak systole, colored by velocity magnitude.

5.6.2. Multiphase flows

Currently, three different multiphase flow scenarios of increasing complexity can be handled by DINOSOARS. Until now, the disperse phase involves only spherical particles though working solutions for resolving spheroids and ellipsoids are available as well [51]

1. Non-resolved solid particles (also called point particles), suitable to describe particles smaller than the Kolmogorov length scale, which cannot be resolved on the employed DNS grid, as depicted in Fig. 3 (left).
2. Non-resolved droplets, possibly involving evaporation and chemical reactions. Here, mass and heat exchange between both phases are considered additionally, similar to the approach retained in [52]. Applied to describe the momentum exchange between flow and particles, the Lagrangian point-force approach considers Stokes model for the drag force [53,54]. The droplet momentum equation reads

$$\frac{d\mathbf{V}_k}{dt} = \frac{\mathbf{U}_\infty - \mathbf{V}_k}{\tau_{v,k}}, \quad (50)$$

$$\frac{d\mathbf{X}_k}{dt} = \mathbf{V}_k, \quad (51)$$

$$\tau_{v,k} = \frac{\rho_L a_k^2}{18\mu_f \left(1 + \frac{1}{6} \text{Re}_k^{2/3}\right)}. \quad (52)$$

In Eqs. (50) – (52), \mathbf{V}_k and \mathbf{U}_∞ are the velocity of the k th droplet and of the surrounding gas at droplet location. Also, \mathbf{X}_k ,

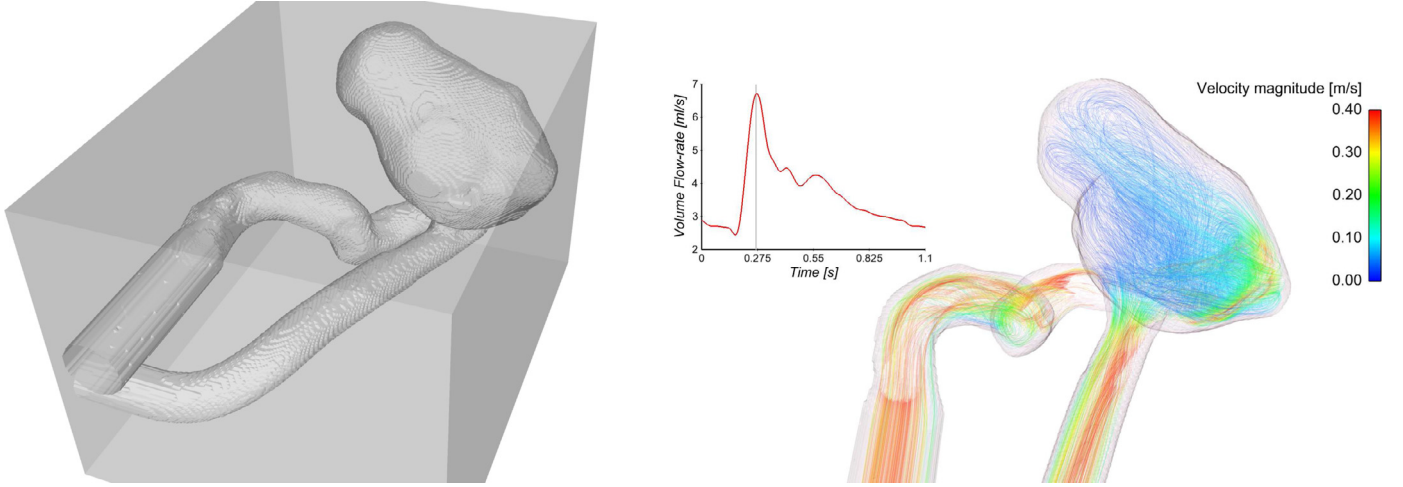


Fig. 2. Representation of domains with a complex geometry in DINOSOARS using DB-IBM. Left: patient-specific cerebral aneurysm [50]. Right: Streamlines colored by velocity magnitude at peak systole.

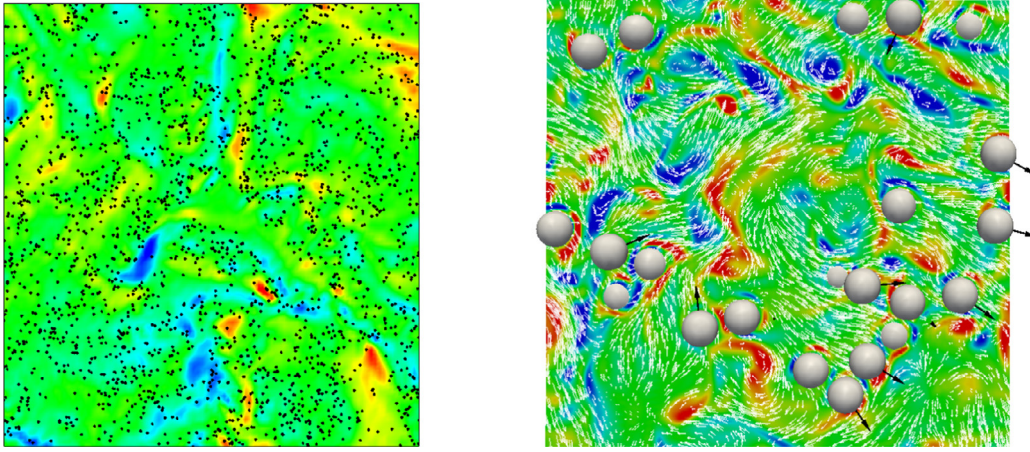


Fig. 3. Particles in initially homogeneous isotropic turbulence (HIT). The color field shows vorticity. Left: non-resolved spherical particles (ratio diameter/Kolmogorov length scale of 0.1). Right: fully resolved solid spherical particles (ratio diameter/Kolmogorov length scale of 20.6).

ρ_L , Re_k , a_k and μ_f are droplet position vector, density, Reynolds number, diameter and gas viscosity at the droplet position, respectively. Subscripts ∞ , f are standing for properties of gaseous mixture in the far-field and film regions, respectively.

The evaporation process is computed by using an infinite conduction model inside the droplet and a variable Spalding mass transfer number, B_k and heat transfer number, $B_{T,k}$ (see Eqs. 53–60)

$$\frac{da_k^2}{dt} = -\frac{a_k^2}{\tau_{a,k}}, \quad (53)$$

$$\frac{dT_k}{dt} = \frac{1}{\tau_{t,k}} \left[T_\infty - T_k - \frac{B_{T,k} L_v}{W_f C_{p,F}} \left(\frac{T_c - T_k}{T_c - T_{ref}} \right)^{0.38} \right], \quad (54)$$

$$\tau_{a,k} = \frac{Sc_k}{4Sh_k} \frac{\rho_L a_k^2}{\mu_f} \frac{1}{\ln(1 + B_{m,k})}, \quad (55)$$

$$\tau_{t,k} = \frac{Pr_k}{6Nu_k} \frac{\rho_L a_k^2}{\mu_f} \frac{B_{T,k}}{\ln(1 + B_{T,k})} \frac{C_{p,L}}{C_{p,f}}, \quad (56)$$

$$B_{m,k} = \frac{Y_{k,s} - Y_{\infty,F}}{1 - Y_{k,s}}, \quad (57)$$

$$B_{T,k} = (1 + B_{m,k}) \frac{C_{p,F} Sh_k Pr_k}{C_{p,L} Nu_k Sc_k} - 1, \quad (58)$$

$$Y_{k,s} = \frac{W_F}{W_F + W_O (P_\infty / P_{k,s} - 1)}, \quad (59)$$

$$P_{k,s} = P_{ref} \exp \left[-\frac{L_v \left(\frac{T_c - T_k}{T_c - T_{ref}} \right)^{0.38}}{R_f} \left(\frac{1}{T_k^s} - \frac{1}{T_{ref}} \right) \right]. \quad (60)$$

In this set of equations, T_k , $Y_{k,s}$, $P_{k,s}$, P_{ref} , T_{ref} , T_c , $Y_{\infty,F}$, P_∞ and T_∞ , are the temperature of droplet k , gaseous fuel mass fraction flux at the droplet surface, partial pressure of fuel at the droplet surface, reference pressure, reference temperature, critical temperature, fuel mass fraction in the surrounding gas mixture, surrounding gas pressure and gas temperature at droplet location, respectively. Also, Pr , $C_{p,L}$, $C_{p,F}$, $C_{p,f}$, Nu , Sc , Sh_c , L_v , W_F and W_O are Prandtl number, specific heat of the liquid droplet, specific heat of the fuel vapor, specific heat of the gas mixture, Nusselt number, Schmidt number, Sherwood number, latent heat of evaporation, molecular weight of fuel and oxidizer, respectively.

3. Fully resolved, moving solid particles. At the difference of the previous approaches, when considering spherical particles

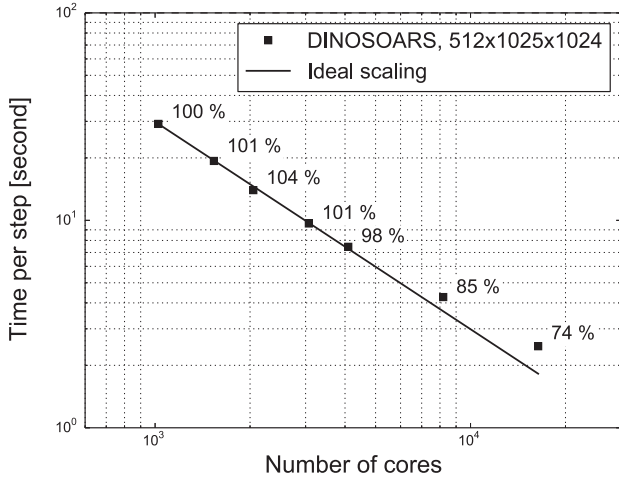


Fig. 4. Parallel efficiency (strong scaling) of DINOSOARS on SuperMUC.

noticeably larger than the Kolmogorov length scale, the external surface of the particles can and should be resolved (Fig. 3, right). For this purpose, the DF-IBM described in [32,55] has been implemented in DINOSOARS [56,57]. Three different collision models are included to describe particle-particle and particle-wall interactions: hard sphere model, lubrication force model, and repulsive force model.

6. Code performance and verification

6.1. Parallel efficiency

After profiling and optimizing single-processor performance, parallel efficiency has been tested on a variety of machines in order to check portability. In the interest of space, only results obtained with SuperMUC at Leibniz supercomputing center in Munich will be discussed. Fig. 4 shows the strong scalability performance obtained with DINOSOARS. This test corresponds to the DNS of a turbulent hydrogen/air flame using a detailed reaction scheme. For this test, the number of processors was varied from 1024 to 16384. The obtained parallel efficiency still reaches 85% with 8192 cores, which is considered as an excellent result for a low-Mach solver. The considerably lower performance with 16384 cores (74%) is due to the fact that, for this number of cores, two separate islands have to be combined on SuperMUC, leading to large communication overheads.

DINOSOARS thus shows a very good parallel efficiency, ensuring efficient computations for up to $O(10^4)$ cores, which is a very satisfactory result for a low-Mach solver, since global operations associated to the Poisson equation severely constrain the parallel performance.

6.2. Accuracy and order

Many benchmarks have been considered during the development of DINOSOARS. In the interest of space, it is impossible to document all the corresponding results. In the present section, the most important benchmarks will be discussed. In order to quantify the accuracy of the solution, the L_2 and L_∞ error norms will be computed:

$$L_2 = \| u^d - U^{\text{ex}} \|_2 = \sqrt{\frac{1}{N} \sum_{n=1}^N |u_n^d - U_n^{\text{ex}}|^2}, \quad (61)$$

$$L_\infty = \| u^d - U^{\text{ex}} \|_\infty = \max_{1 \leq n \leq N} |u_n^d - U_n^{\text{ex}}|. \quad (62)$$

Where u^d and U^{ex} are the numerical and exact analytical solutions, respectively.

6.2.1. Spatial order and Poisson equation: 2D Taylor–Green vortex (2D-TGV)

The first benchmark considers the two-dimensional Taylor–Green vortex (2D-TGV). The 3D version of this problem will be discussed later for validation. The initial velocity and pressure fields are prescribed at $t = 0$ as follows:

$$u(x, y, 0) = \sin(2\pi x/L) \cos(2\pi y/L),$$

$$v(x, y, 0) = -\cos(2\pi x/L) \sin(2\pi y/L),$$

$$p(x, y, 0) = \frac{\rho}{24} [\cos(4\pi x/L) + \cos(4\pi y/L)]. \quad (63)$$

The analytical solution reads:

$$u(x, y, t) = u(x, y, 0) \exp^{-8\pi^2 \nu t/L^2},$$

$$v(x, y, t) = v(x, y, 0) \exp^{-8\pi^2 \nu t/L^2},$$

$$p(x, y, t) = p(x, y, 0) \exp^{-16\pi^2 \nu t/L^2}. \quad (64)$$

In the present case, the 2D-TGV is simulated in a periodic square domain with a side length of $L = 1.0$ m, with varying number of grid points (N) and kinematic viscosity (ν) but with a fixed time step ($\Delta t = 0.1$ ms). The error norm L_2 of the velocity field shown in Fig. 5 (left) demonstrates that DINOSOARS reaches 6th order in space for the solution of the Navier–Stokes equation. Concerning now the Poisson solver, the error norms of the pressure field have been computed based on the analytical solution. Fig. 5 (right) shows indirectly that the order of convergence for the Poisson solver is much higher than sixth-order. In fact, exponential convergence is achieved, as demonstrated in further tests (not shown in the interest of space). Due to the interaction between finite-difference and spectral solver in this particular test case (Eq. 28), the overall order shown in Fig. 5 (right) cannot be better than the sixth-order of the finite-difference solution.

6.2.2. Temporal order: Wave equation

To check the temporal order of the time integration algorithm relying on the explicit Runge–Kutta scheme, a hyperbolic, one-dimensional wave equation was considered in a periodic DNS domain

$$\frac{\partial u}{\partial t} = -c \frac{\partial u}{\partial x}, \quad (65)$$

$$u(x, t = 0) = \sin(2\pi x), \quad (66)$$

For this system, the analytical solution is known

$$U(x, t) = \sin[2\pi(x - ct)]. \quad (67)$$

Fig. 6 (left) shows the comparison between the solution obtained with DINOSOARS, u , and the analytical solution, U , at different time instants. The simulation employs 128 grid points, a fixed timestep ($\Delta t = 0.5$ ms), and a fixed wave speed of $c = 0.5$ m/s. In order to obtain the temporal order, both norms are again computed, this time with a fixed number of grid points but changing the timestep. Fig. 6 (right) shows that the Runge–Kutta procedure delivers as expected an overall fourth-order approximation in time.

7. Validation

7.1. Homogeneous isotropic turbulence (HIT)

For a DNS solver, describing properly turbulence properties is obviously essential. In order to check this feature, one of most fundamental test cases is the decay of homogeneous isotropic turbulence. Validation is done here by comparing the turbulence statistics obtained by DINOSOARS with those obtained experimentally

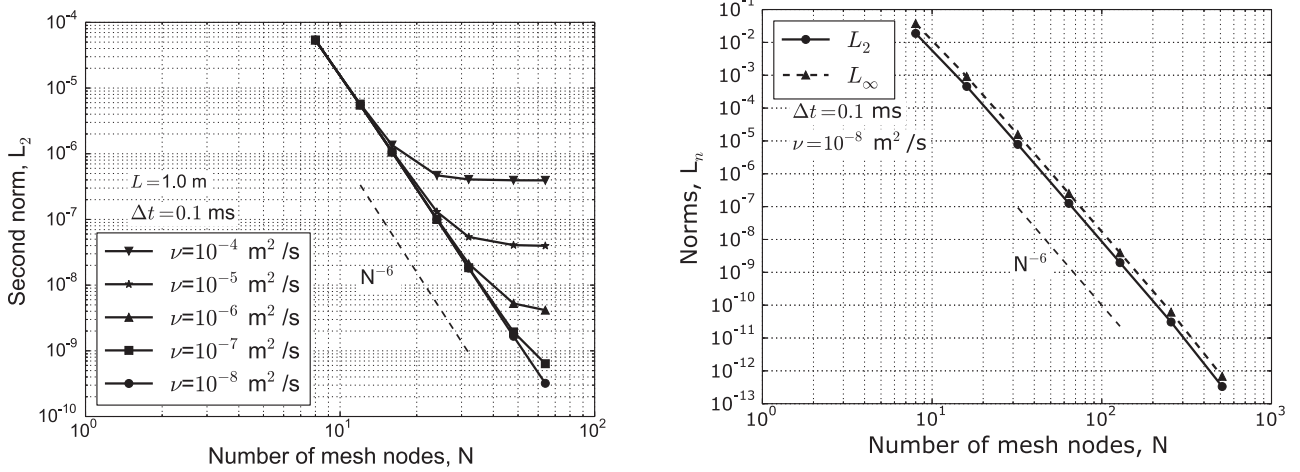


Fig. 5. Verification and spatial order obtained by solving the 2D Taylor-Green vortex problem. Left: L_2 of velocity field. Right: L_∞ and L_2 of pressure field.

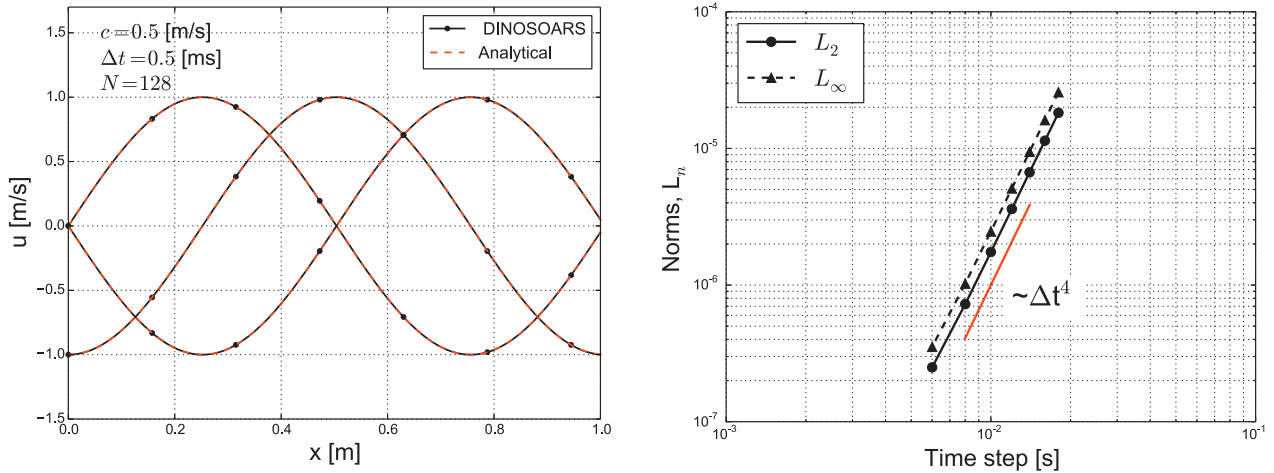


Fig. 6. Verification and temporal order obtained by solving the one-dimensional wave equation. Left: time-dependent solution. Right: error norms and resulting temporal order of DINOSOARS.

behind grid turbulence [58]. The normalized spectrum obtained with DINOSOARS for decaying HIT is compared directly with the spectrum from the experiment of Comte-Bellot and Corrsin (1971) at a Taylor Reynolds number of $Re_\lambda = 71.6$ and $Re_\lambda = 65.1$. To initiate the simulation, the DNS is started with an analytically prescribed turbulence spectrum $E(\kappa)$ following von Kármán with Pao correction (VKP spectrum):

$$E(\kappa) = \frac{3u_e^5}{2\epsilon_e} \frac{[\kappa/\kappa_e]^4}{[1 + (\kappa/\kappa_e)^2]^{17/6}} \exp\left(-\frac{9}{4} \left[\frac{\kappa}{\kappa_d}\right]^{4/3}\right), \quad (68)$$

where u_e , ϵ_e , κ_d and κ_e are the spectrum coefficients and κ is the wave number vector. Here, values of 0.21, 0.01, 0.08 and 0.018 have been chosen for the spectrum coefficients, respectively, in order to generate initial turbulence in a domain of size $10 \times 10 \times 10$ cm³. After a time $t = 11.2\tau_\eta$, a Taylor-based Reynolds number $Re_\lambda = 71.1$ is obtained and the turbulence properties are compared with the experimental data. Table 2 also contains Re_λ , ϵ , u' , λ , Λ , τ_Λ , η , and τ_η , which correspond to Reynolds number based on integral length scale, dissipation rate of the kinetic energy, fluctuation velocity root mean square, Taylor length scale, integral length scale, large eddy turn-over time scale, Kolmogorov length scale, and Kolmogorov time scale, respectively.

Fig. 7 shows the results for the Q-criterion isosurface (Fig. 7, left), and the normalized energy spectrum of the simulation at time $t = 11.2\tau_\eta$ and of the experiments (Fig. 7, right). The compar-

isons show that the DINOSOARS simulation agrees perfectly with the experimental results. It is important to keep in mind that, in this simulation, the ratio of $\eta/\Delta x$, where Δx denotes the grid spacing, is larger than unity and the product $\kappa_{max}\eta$ is larger than 2 during the whole simulation, ensuring that all small flow structures are correctly captured.

7.2. 3D Taylor-Green vortex (TGV)

The Taylor-Green vortex (TGV) is a canonical problem in fluid dynamics to study vortex dynamics, laminar to turbulent transition, turbulent decay and energy dissipation. Furthermore, it was retained as one of the central benchmarks in the International Workshops on High-Order CFD Methods [59]. The TGV problem involves different key physical processes found in turbulence and is therefore an excellent testcase for the evaluation of DNS codes. The problem consists of a cubic volume of fluid that contains a smooth initial distribution of vorticity, as shown in Fig. 8.

As time advances the vortices roll-up, stretch and interact before breaking down. Eventually, viscosity will dissipate all the energy in the fluid and it will come to rest along a well-defined trajectory [60]. Here, results of DINOSOARS are compared with simulation results obtained with a pseudo-spectral code using 512^3 grid points [61]. These results are later denoted RLPK. The geometry is a periodic box of dimension of $0 \leq x, y, z \leq 2\pi$ [m], a value

Table 2Turbulence properties at time $t = 11.2\tau_\eta$

N^3	Re_λ	Re_Λ	ϵ [m^2/s^3]	u' [m/s]	λ [mm]	Λ [mm]	τ_Λ [ms]	η [mm]	τ_η [ms]
512^3	71.1	336.8	42.52	0.92	3.86	18.3	81.4	0.23	1.1

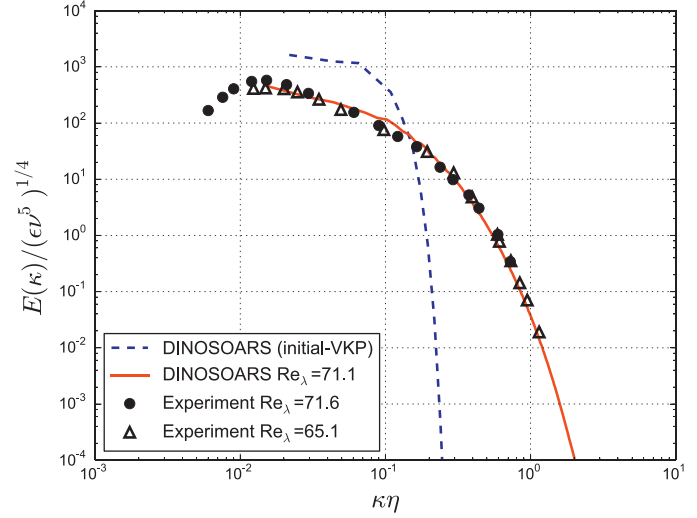
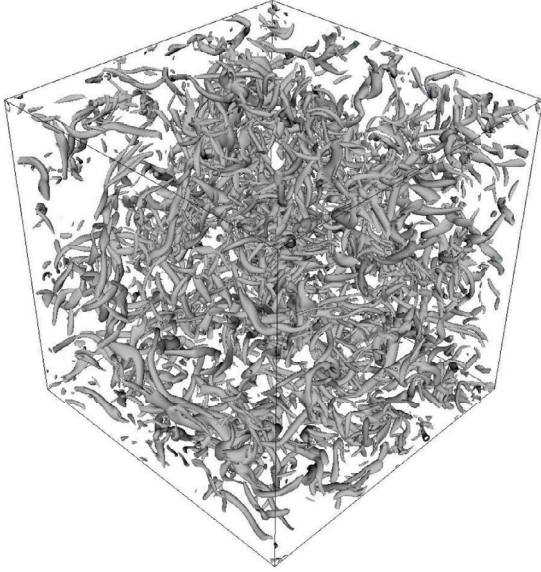


Fig. 7. Homogeneous isotropic turbulence decaying in time. Left: Q-criterion isosurface (value of 6×10^5 $1/s^2$) at time $t = 11.2\tau_\eta$. Right: energy spectrum obtained by DINOSOARS compared with experimental results of [58].

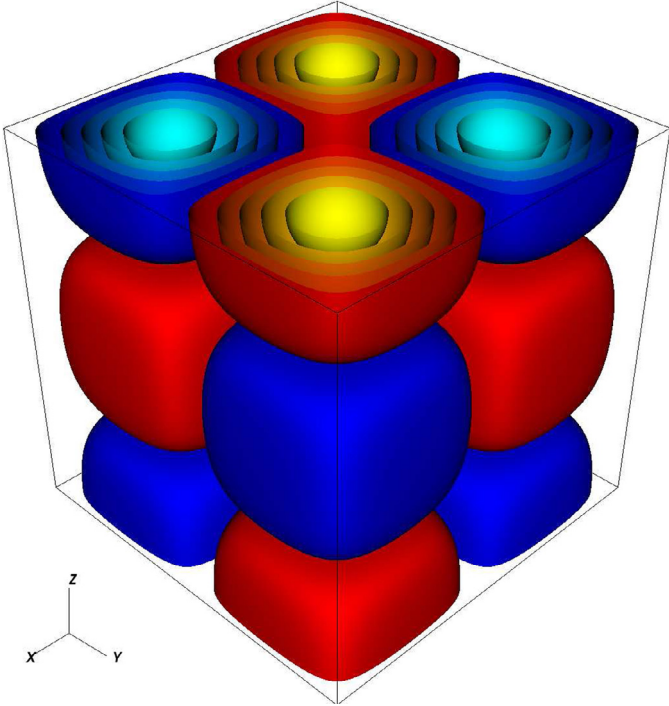


Fig. 8. Isosurface of z-vorticity of 3D-TGV at initial conditions.

kept to facilitate post processing and comparisons with results from the literature. The initial conditions for TGV are given by the following:

$$u(x, y, z, 0) = u_0 \sin(x/L) \cos(y/L) \cos(z/L), \quad (69)$$

$$v(x, y, z, 0) = -u_0 \cos(x/L) \sin(y/L) \cos(z/L), \quad (70)$$

$$w(x, y, z, 0) = 0, \quad (71)$$

where $u_0 = 1$ m/s and $L = 1$ m are reference velocity and length, respectively. The flow is computed at a Reynolds number ($Re = u_0 L/\nu$) of 1600. Using 384^3 or 512^3 grid points, the simulation is performed for a duration of $20T_c$, where $T_c = L/u_0$ is the characteristic convective time scale. By solving the Poisson equation, the initial pressure field is obtained. The time evolution of isosurfaces of z-vorticity at different times when using 512^3 grid points is illustrated in Fig. 9. Starting from Fig. 9 (top-left), the vortices start to roll-up. Then, a breakdown of the coherent structures occurs (Fig. 9, top-right). Fig. 9 (bottom-left) shows the onset of turbulence before decay due to dissipation (Fig. 9, bottom-right).

Fig. 10 presents the comparison of the DINOSOARS results with that of the pseudo-spectral code (RLPK [61]). On the left, the time evolution of kinetic energy

$$KE(t) = \frac{1}{2} \langle u_i u_i \rangle, \quad (72)$$

normalized by its initial value is presented, while on the right the evolution of the dissipation rate ϵ (normalized by its maximum value) is shown. For this, ϵ is computed directly from the strain rate tensor, S_{ij}

$$\epsilon(t) = 2\nu \langle S_{ij} S_{ij} \rangle, \quad (73)$$

where $\langle \cdot \rangle$ in Eqs. (72) and (73) denotes a spatial average. The comparison between DINOSOARS and RLPK results shows an excellent agreement. This benchmark proves again the ability of DINOSOARS not only to simulate turbulent flows but also transition and dissipation processes.

7.3. Turbulent channel flow

Since resulting features are non-homogeneous and non-isotropic, DNS of a wall-bounded turbulent channel is far more

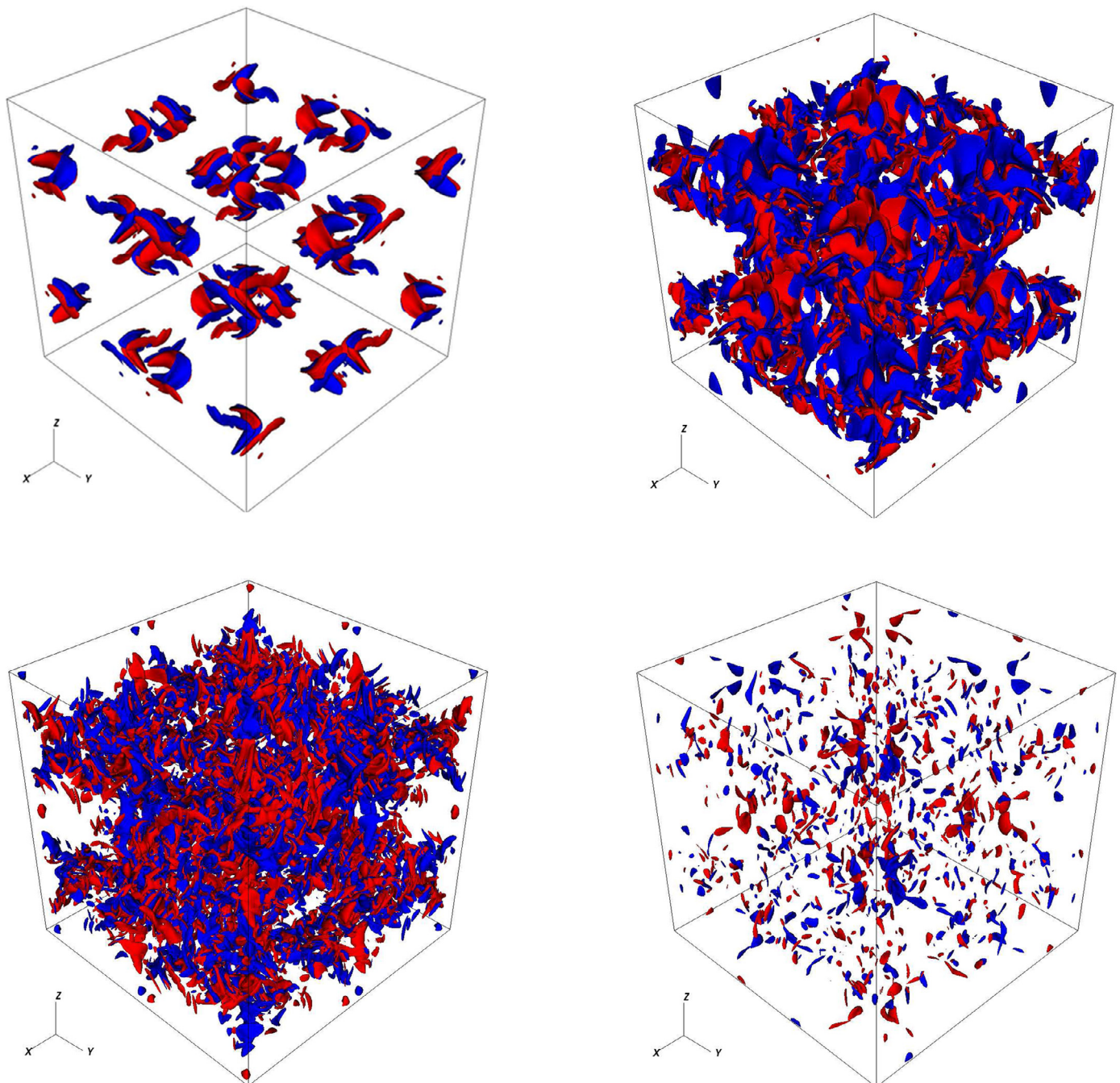


Fig. 9. Time evolution of isosurfaces of z -vorticity for 3D-TGV using 512^3 grid points at times: $t/T_c = 5.46$ (vortex roll-up, top left), $t/T_c = 8.0$ (coherent structure breakdown, top right), $t/T_c = 12.11$ (turbulence, bottom left), $t/T_c = 18.55$ (decay, bottom right), respectively.

complicated than HIT, but also more interesting for practical purposes. Considering that DINOSOARS will ultimately be used to investigate multiphase reacting flows, and keeping in mind that any injection channel and most combustion systems involve walls, it is absolutely necessary to obtain a correct description of such configurations. Near-wall turbulent structures are typically much smaller than in the center of the channel. Hence, using a refined grid near the wall is helpful. As already mentioned in Section 5.6.1, this feature is implemented in DINOSOARS, and has been activated for the present test case. In order to check the accuracy of DINOSOARS when employing the developed spectral solver described in Section 5.4 on a non-uniform grid, the results obtained for a turbulent channel flow with a friction Reynolds number

$Re_\tau \approx 180$ are compared with published DNS databases [62,63]. These databases were generated with two different spectral codes. Simulation dimensions and parameters considered for DINOSOARS and for the databases are summarized in Table 3. The current simulation is initiated by using a semi-empirical profile for the turbulent channel flow and adding divergence-free turbulent fluctuations on top of that profile. These initial fluctuations are again generated by IFFT using a VKP spectrum as described in Section 5.5. Compared to initialization with a random noise, preliminary tests have confirmed that this leads to a much faster convergence toward statistically-steady results. The boundary conditions are chosen to be periodic in streamwise and spanwise flow directions, using a uniform grid. Along the third direction, wall boundary

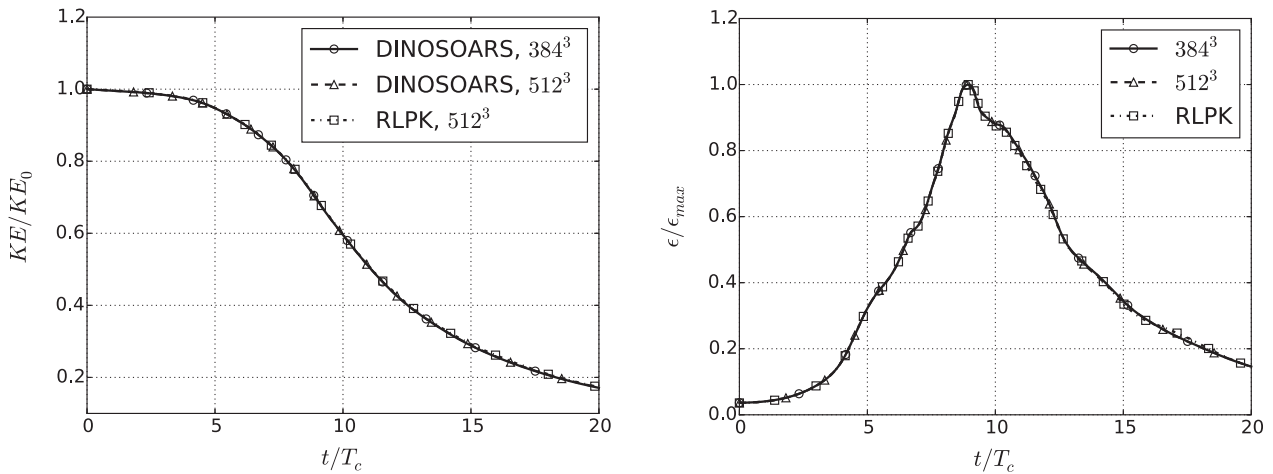


Fig. 10. Evolution with time of normalized kinetic energy (left) and of its dissipation rate (right).

Table 3
Parameters of the turbulent channel flow.

Data	Re_τ	δ	L_x	L_z	N_x	N_y	N_z	Δx^+	Δy^+	Δz^+
DINOSOARS	175	0.125 [m]	8δ	4δ	256	193	128	5.4	2.0	5.4
MKM	178	1.00 [-]	$4\pi\delta$	$4\pi\delta$	128	129	128	17.7	4.4	5.9
VK	180	1.00 [-]	$4\pi\delta$	$4\pi\delta$	384	193	192	5.9	2.9	3.9

Note that the mesh sizes Δx^+ , Δy^+ , Δz^+ are in wall units, and that Δy^+ represents the grid size at the center, the grid being refined close to the walls.

conditions are used together with near-wall grid refinement. The smallest grid spacing near the wall is $\Delta y_{min}^+ = 1.4$. Turbulence statistics start being collected after reaching statistically steady-state. The mean streamwise velocity in the (dimensional) DINOSOARS simulation is 0.83 m/s. Fig. 11 shows the comparison between DINOSOARS, Moser et al. (MKM) and Vreman and Kuerten (VK) concerning mean velocity profile (u^+ , Fig. 11, left), as well as velocity fluctuations and correlations (Fig. 11, right). An excellent agreement is observed, showing that DINOSOARS is able to correctly describe turbulence with the spectral Poisson solver, even for wall-bounded flows and using non-uniform grids.

7.4. IBM validation: Flow past a circular cylinder

In order to validate the implemented IBM approaches, the classical benchmark involving the flow past a circular cylinder has been revisited [64]. The circular cylinder has a diameter of $D_c = 25$ mm. In order to avoid blockage effects, the cylinder is located at $(16D_c, 20D_c)$ within a square domain with side length of $40D_c$. Two different Reynolds numbers ($Re = U_\infty D_c / \nu = 20$ and 40) have been considered by changing the free stream velocity U_∞ while keeping $\nu = 1 \times 10^{-4} \text{ m}^2/\text{s}$. A fine regular grid with 1025×1025 points has been used to discretize the domain. Fig. 12 depicts the streamlines of the flow around the cylinder with Re equal to 20 (left) and 40 (right), using DB-IBM (note that both methods, DB-IBM and DF-IBM, deliver streamlines that cannot visually be distinguished). The drag coefficients obtained by these simulations have been compared with values from the literature in Table 4, showing an excellent agreement.

7.5. Chemistry and transport

After successfully completing the validation of DINOSOARS for turbulent non-reacting incompressible flows, and keeping in mind that DINOSOARS shall be also used to investigate reacting configurations, it is necessary to check that all models employed to describe the properties needed for such cases (chemical kinetics,

Table 4
Comparison of the drag coefficient C_d obtained by IBM in DINOSOARS with literature values.

Reference	$Re = 20$	$Re = 40$
Body-fitted grid [65]	2.045	1.522
Body-fitted grid [66]	2.111	1.574
IBM [64]	2.144	1.589
DINOSOARS, DF-IBM	2.140	1.597
DINOSOARS, DB-IBM	2.104	1.581

thermodynamic parameters and transport properties) are indeed working accurately. For this purpose, comparisons with experimental measurements will be carried out for configurations of growing complexity.

7.5.1. Ignition delay

The first comparison allows one to check the correct implantation of the kinetic terms, showing that the coupling between DINOSOARS and Cantera 1.8 is working properly. Additionally, the efficiency of the time integration for stiff processes can be assessed. For this purpose, the ignition delay of a reacting mixture is computed in a zero-dimensional simulation, comparable to an experiment involving a homogeneous mixture. In such a case, all gradients appearing in the equations are zero, and the integration proceeds only in time, leading to very fast computations.

Since many different fuels are important for research and practical applications, two different combustibles have been considered here, both burning in air: (1) C_2H_4 (ethylene); (2) C_7H_{16} (*n*-heptane). Different reaction mechanisms are available as a database in the literature to describe oxidation of these fuels. Established mechanisms have been retained for this validation, as summarized in Table 5.

Fig. 13 (left) shows the results obtained with DINOSOARS concerning ignition delay for ethylene combustion, compared to a variety of published experimental data [71–73]. Please note that experimental results have been obtained under different

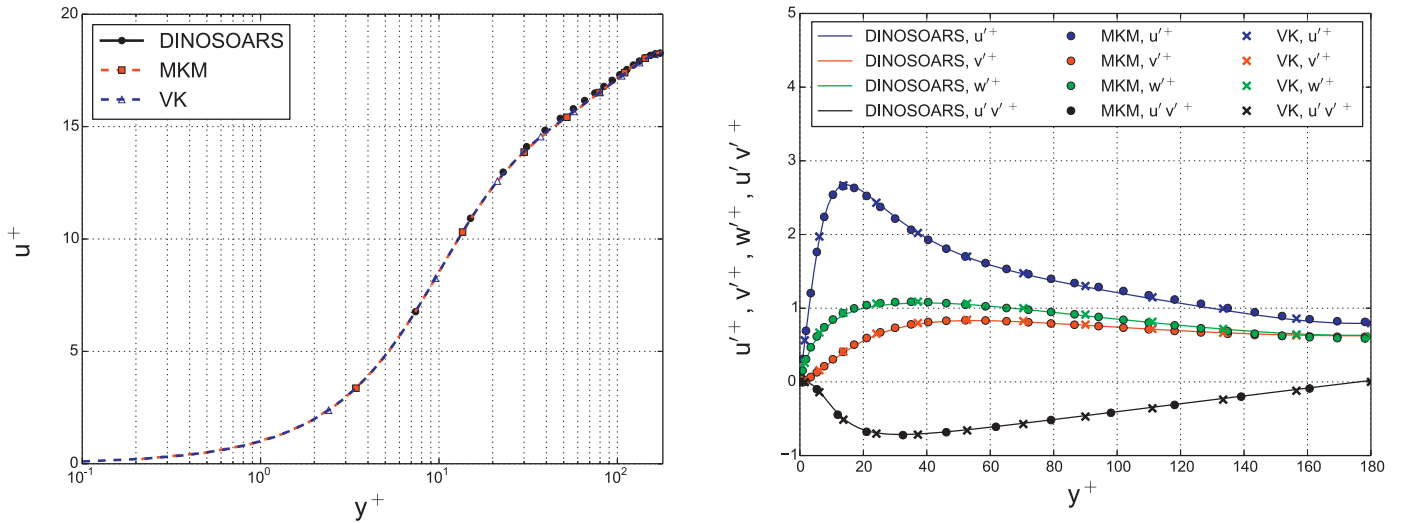


Fig. 11. Comparison between DINOSOARS, Moser et al. (MKM) and Vreman and Kuerten (VK) concerning the turbulent channel flow at $Re_\tau \approx 180$. Left: mean velocity. Right: velocity fluctuations and correlations.

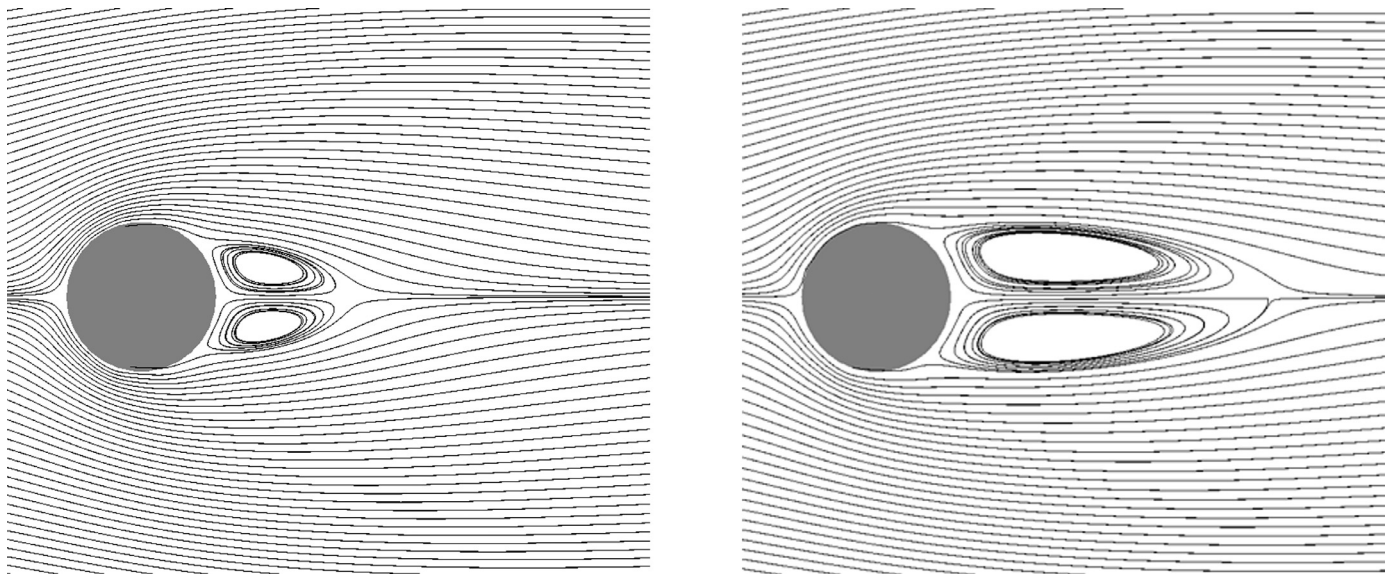


Fig. 12. Streamlines of flow past a circular cylinder using DB-IBM at $Re = 20$ (left) and $Re = 40$ (right).

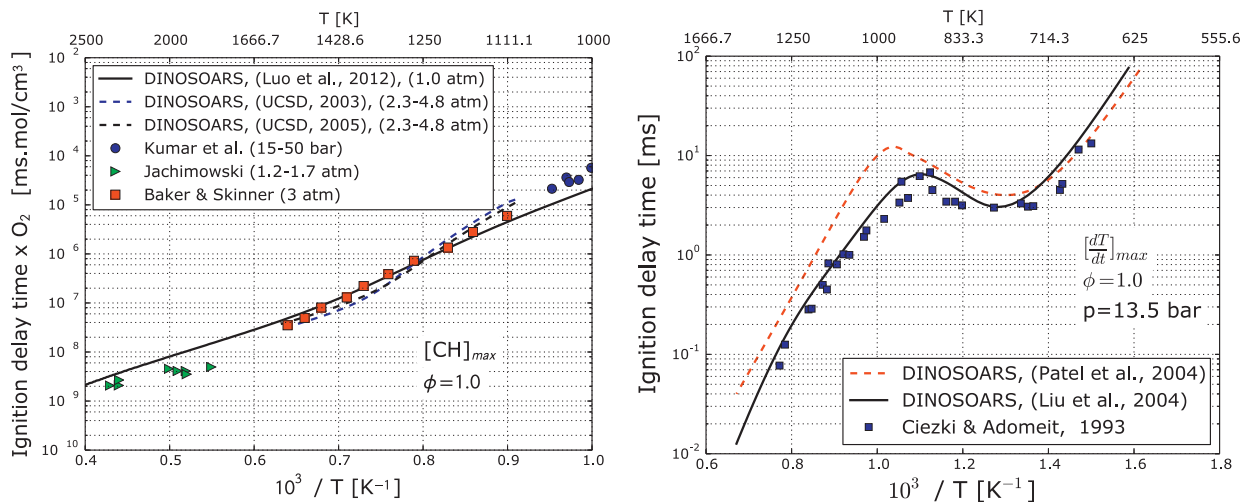


Fig. 13. Ignition delay obtained with DINOSOARS compared with experimental results. Left: C_2H_4/Air . Right: C_7H_{16}/Air . Note that the data have been scaled by the oxygen concentration in the left figure.

Table 5

Reaction mechanisms employed to compute ignition delays.

Mixture	Mechanism	Number of species	Number of reactions
C ₂ H ₄ /Air	UCSD-2003 [67]	39	173
	UCSD-2005[67]	46	235
	Luo et al.[68]	32	206
C ₇ H ₁₆ /Air	Patel et al. [69]	29	52
	Liu et al. [70]	44	114

conditions. Therefore, following a standard procedure in this case, the results of Fig. 13 (left) have been scaled by the oxygen concentration in the mixture. A very good agreement is obtained, in particular with the most complex mechanism [67]. However, the computational time associated to the mechanism of [68] is considerably shorter, while an acceptable agreement is still observed.

Fig. 13 (right) shows the ignition delays obtained for *n*-heptane in air, compared to published experimental data [74]. Again, the most complex mechanism [70] leads to an excellent agreement with the experiments, while the noticeably smaller (and thus computationally faster) mechanism of [69] still leads to an acceptable prediction, at least qualitatively.

All results in this section have been obtained by using the split, semi-implicit Runge–Kutta time integration relying on RADAU-5 (see Section 5.2). A fast but stable time integration has been obtained for all conditions.

7.5.2. Laminar flame speed

Together with the ignition delay previously discussed, the laminar flame speed S_L observed when burning a fully premixed system (also called in what follows fresh gas), leading to so-called burnt gas conditions, is considered as the key quantity to properly describe gaseous reacting flows. This property can be easily computed using dedicated routines like those implemented in Chemkin or Cantera. However, when using an unsteady solver like DINOSOARS, a proper computation of S_L must first be implemented.

Assuming that a quasi steady-state has been obtained with DINOSOARS, all time derivatives are zero ($\partial_t(\cdot) = 0$). Since the position of the flame is fixed within the computational domain under such conditions, and following [20], it can also be assumed that the momentum in the fresh gas mixture (subscript 1) and in the burnt mixture (subscript 2) are constant and equal to $(\rho_1 S_L)$, where S_L is the (at first unknown) laminar flame speed. Consequently, by integration of the species equation (Eq. 4), the following equation is obtained to compute the laminar flame speed out of corresponding DNS simulations:

$$S_L = \frac{\int_1^2 \dot{\omega}_k dx - \rho_1 V_{k1} Y_{k1} + \rho_2 V_{k2} Y_{k2}}{\rho_1 (Y_{k2} - Y_{k1})}. \quad (74)$$

This equation delivers one value for each species (index k). Following values are the average obtained when considering the obtained values of S_L for all reactants and products. In order to achieve steady-state conditions with DINOSOARS, an initially planar premixed flame configuration is initiated in a one-dimensional domain, with a fresh gas mixture on one side and the corresponding burned gas composition (separately computed beforehand with Cantera 1.8) on the other side of the domain, first without any underlying velocity. Realistic flame profiles progressively develop, after starting the simulation. After a prescribed number of iterations, a first estimation of the flame speed is computed using Eq. (74). In order to stabilize the reaction front within the computational domain, the computed value is then prescribed in the simulation as fixed velocity inlet. This procedure is repeated iteratively until the updated flame speed computed using Eq. (74) remains constant

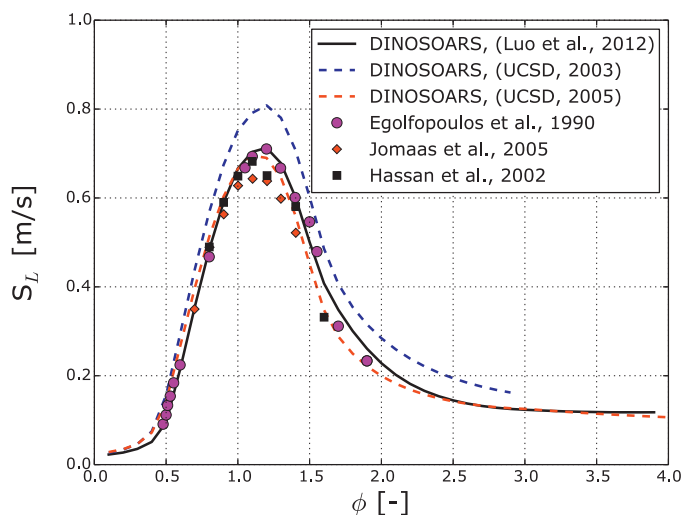


Fig. 14. Comparison of numerically obtained laminar flame speeds with experimental measurements for ethylene/air flames.

within $\pm 5\%$. Using this approach, a quasi steady-state can be obtained, with a stable flame established at a fixed position within the one-dimensional domain.

Fig. 14 shows the comparison of the numerically obtained flame speeds with experimental measurements for ethylene. The integration in time relies again on the additive semi-implicit Runge–Kutta method (see Section 5.2). Different independent experimental results have been involved in the comparison [75–77]. Fig. 14 shows that DINOSOARS reproduces with an excellent accuracy the laminar flame speed of ethylene flames when using as reaction mechanisms either UCSD-2005 or Luo *et al.* The third scheme (UCSD-2003) shows noticeable deviations and will not be considered further.

Concluding Section 7.5, DINOSOARS coupled with Cantera 1.8 and Eglib 3.4 is able to represent accurately the key properties controlling combustion, even for complex fuels, provided corresponding reaction schemes are available in the scientific literature. Thanks to the implicit time-integration solvers, stiff systems can be treated in a stable but accurate manner. Until now, up to 46 species and up to 235 reactions (ethylene flames) have been successfully considered in DINOSOARS.

8. Preprocessing and postprocessing

The parameters for a simulation run are basically defined in a text file. In order to get a user-friendly DNS code, a graphical user interface (GUI) has been developed for editing this settings file. This GUI helps the user understand the large number of available parameters and prevents them from choosing unreasonable or incompatible values. The interface presents all available parameters in a hierarchical structure. Parameters that are not relevant given the current settings are hidden. Each parameter is displayed with a human-readable name, and a detailed explanation is available via tooltip.

Two levels of checks help to ensure that the user chooses sensible parameter values: Input fields will not allow one to set values outside a sensible range for each parameter, and a series of sanity checks is performed to detect problems with parameters that are interdependent. For example, when simulating a reactive flow with combustion, the low Mach number solver must be used. Sanity checks also warn the user about possible mistakes, such as restarting the simulation from a previous snapshot instead of from the beginning. Errors and warnings about parameter values are displayed in the interface, along with hints on how to fix them, as

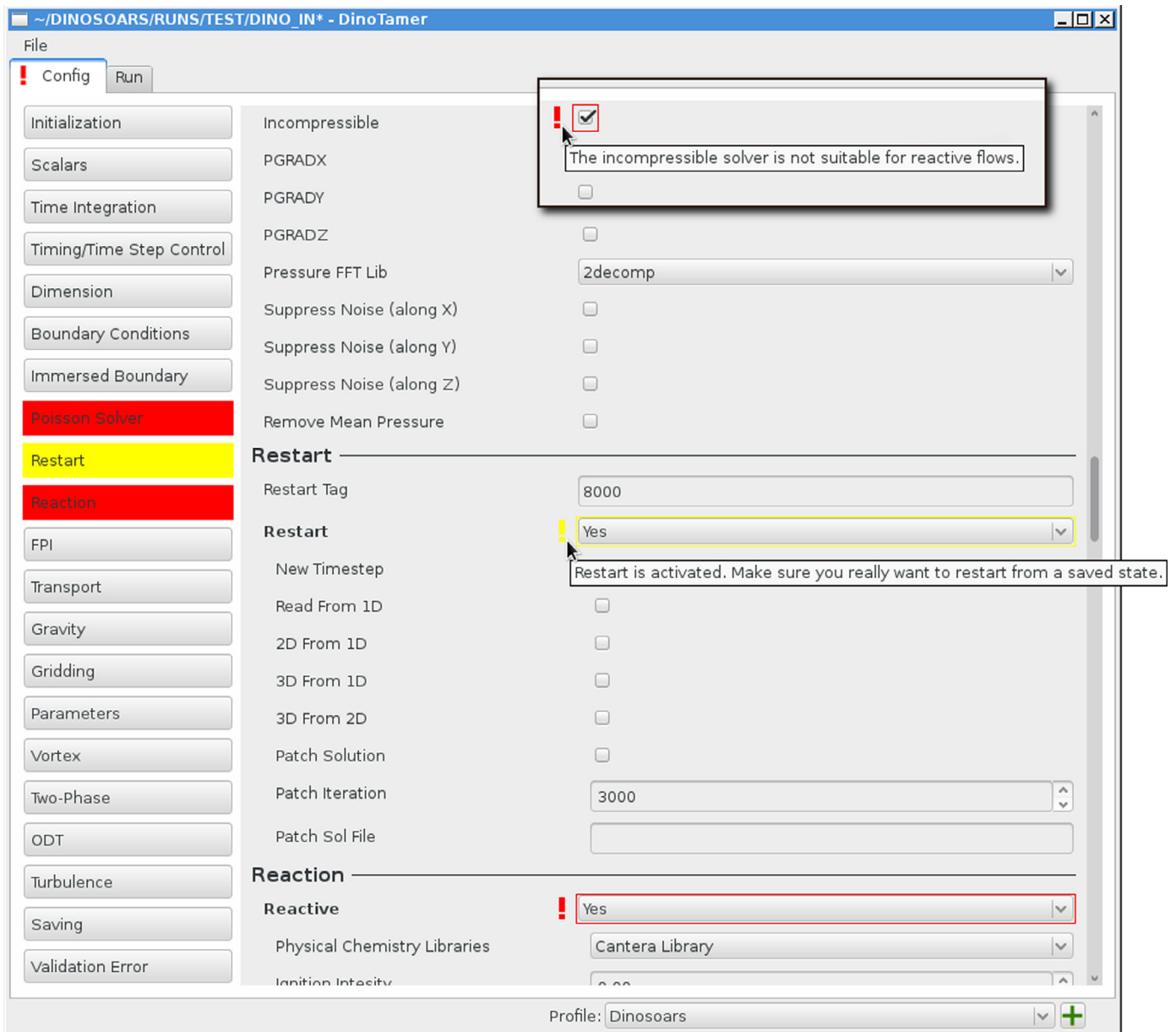


Fig. 15. User interface DinoTamer for configuring a simulation run. Here, some errors and warnings were detected by the integrated sanity checks. The reasons for the errors and warnings are displayed in tooltips. The buttons on the left side allow the user to quickly jump to a group of settings and show which groups have errors (red) or warnings (yellow). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

seen in Fig. 15. This makes it easier for new users to get acquainted with the software and its capabilities.

For input/output operations the code relies on the MPI-based I/O routines provided by the 2DECOMP&FFT library. Full binary files are only used for restarting the simulations. For file-based post-processing, DINOSOARS saves results using parallel HDF5, directly compatible with many analysis tools (for example VisIt and ParaView).

Even if only relevant variables and parameters are stored in this manner, the resulting files remain very large. For this reason, file-based post-processing is only feasible at greatly reduced temporal or spatial resolution. Therefore, an on-the-fly co-processing using the ParaView Catalyst framework [78,79] has been developed. Implementing a ParaView Catalyst adaptor for DINOSOARS allows one to define ParaView and VTK [80] processing pipelines to be executed during the simulation. This gives us the freedom to perform arbitrarily complex visualization and analysis tasks. The co-

processing, as the name implies, runs in parallel and in the same processes as the simulation itself, such that DINOSOARS memory can be accessed without copying or communication overhead. In addition to this, live visualization can be enabled. The running simulation can then be opened in a ParaView client much like a file on disk during post-processing. ParaView also allows for pausing of the simulation in this mode, so that a time step can be fully explored before the simulation continues.

Processing pipelines can be defined in Python using the ParaView Python API. The ParaView Catalyst includes a plugin that allows to export a pipeline defined in the ParaView client as a Python script, which can then be manually adjusted if desired. Used during the simulation, Python co-processing scripts are specified at launch time. If more control over the co-processing is required, pipelines can also be written in C++ and compiled into the executable. In this case, the complete VTK and ParaView libraries are available and arbitrary analysis and visualization tasks can be

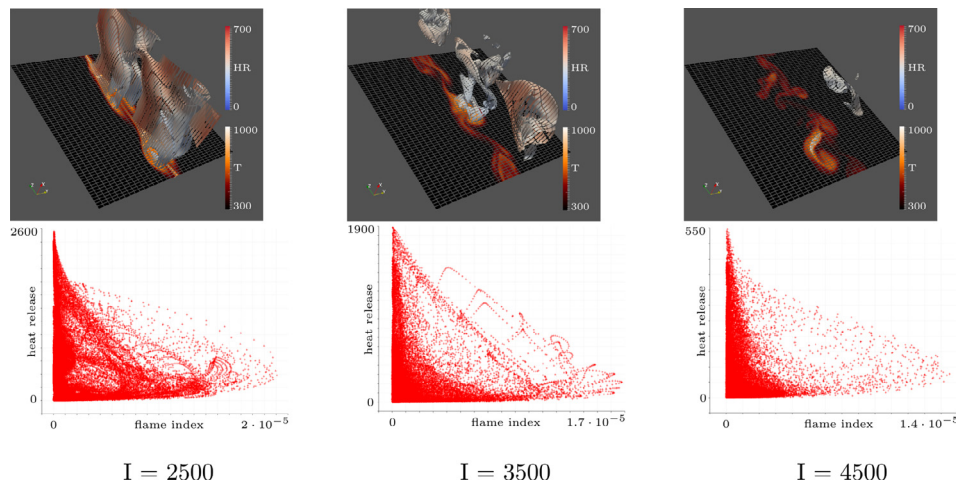


Fig. 16. Renders generated by ParaView Catalyst during tests for different iteration intervals. Top: Pipeline 1 renders a slice through the temperature field and the $T = 700$ K isosurface with the local heat release (HR) color mapped onto the surface. The pencil domain decomposition is visible as gaps in the surface. Bottom: Pipeline 2 computes the Takeno flame index and generates scatterplots of this value vs. heat release. As expected for a premixed combustion regime, the index has positive values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

implemented. During the simulation, the Catalyst adaptor wraps the DINOSOARS arrays into VTK data structures. Due to their simple structure, this is possible without copying or duplicating any memory. These data structures are then passed on to the processing pipelines. Depending on the configuration, a pipeline can be executed after each simulation step, or in larger intervals. Results are written to disk in the form of images or VTK data files, or they can be viewed via the live visualization feature.

To check that co-processing is working properly, two different visualization pipelines have been executed during a simulation run. The first pipeline simply renders a slice of the temperature field as well as an isosurface of the temperature with the local heat release color-mapped onto the surface. The second pipeline computes the Takeno flame index $\xi = \nabla Y_F \cdot \nabla Y_O$ [81] for each grid point and renders a scatterplot of this flame index vs. heat release, where Y_F and Y_O are fuel and oxidizer mass fraction, respectively. Positive values of ξ denote a premixed combustion regime, while negative values correspond to nonpremixed combustion. Example images can be seen in Fig. 16. These tests correspond to the DNS of a premixed ethylene jet flame computed with 8.4 million grid points, using 2048 cores on the SuperMUC. It was found for instance that rendering each iteration generates a 16% overhead in execution time, while a negligible overhead of less than 2% is found when rendering every 5 iterations.

Implementing an adaptor for ParaView Catalyst allows us to perform on-the-fly co-processing in DINOSOARS. Using Python or C++, analysis and visualization pipelines can be defined and executed during the simulation run. Thus, a detailed insight into the simulation can be obtained without writing huge amounts of data to disk.

9. Final application examples

To conclude this work, two different application examples will be presented: 1) a gaseous jet premixed flame, 2) spray combustion in a turbulent environment. These two examples are very important benchmarks for research concerning in particular industrial burners, internal combustion engines, gas turbines, and safety predictions.

9.1. Turbulent jet flame

The first case is particularly relevant for safety considerations. A small, central hot gas jet comes into contact with a combustible

co-flow (stoichiometric ethylene/air mixture). It is then checked if ignition occurs in the whole gas volume, or not. Burned gases are injected in the central jet with a jet velocity (U_j) of 180 m/s at a temperature of 1800 K. A co-flow combustible mixture is injected with a low velocity (U_{co}) of 10 m/s and temperature of 300 K. The central jet has a width of only 0.28 mm, corresponding for instance to a leakage. The kinematic viscosity of the co-flow and of the central jet are $\nu_{co} = 1.53 \times 10^{-5}$ and $\nu_j = 3.14 \times 10^{-4}$ m²/s respectively. This configuration has Reynolds number, $Re = (U_j - U_{co})H/\nu_{co}$, of 3111 with characteristic jet time scale, $\tau_j = H/(U_j - U_{co})$, of 1.65 μ s. The domain of $2 \times 8 \times 6$ mm³ is discretized with $128 \times 513 \times 384$ grid points to obtain a resolution of 16 μ m in all directions. Inlet/outlet boundary conditions are implemented in the streamwise direction. Turbulence is injected in the high-speed central jet through the boundary condition, using a top-hat filter. Fig. 17 depicts the time-evolution of temperature at three different instants: $t = 30, 70$ and $130 \tau_j$, from left to right, respectively. It is observed that the combustible co-flow does not ignite in this particular configuration (the temperature increase stays below 1 K), which can thus be deemed “safe.” This case has been computed on SuperMUC using 46 700 CPU-hours with 2048 cores, including the time required for all I/O operations and on-the-fly analysis. This simulation confirms the ability of DINOSOARS to represent complex turbulent flames in a computationally efficient manner.

9.2. Spray evaporating and burning in a turbulent environment

DNS of the interaction between combustible droplets and turbulence is considered here by randomly distributing 2000 droplets (diameter $a_k = 11.0 \mu$ m, initial temperature, $T_k = 300$ K), in a central slab of the DNS domain (size $H = 0.6$ mm). The entire domain contains a hot oxidizer (air) at initial temperature $T_0 = 1500$ K, pressure $p = 3$ bar and kinematic viscosity $\nu = 8 \times 10^{-5}$ m²/s. While the surrounding gas within the central slab moves initially with a jet velocity $U_j = 100$ m/s, the rest of the domain (surrounding co-flow) moves at a velocity $U_{co} = 10$ m/s. The resulting initial jet Reynolds number is $Re_j = 675$. By initializing with an isotropic turbulent flow field, the turbulence is triggered. The overall arrangement is equivalent to a temporally-evolving jet, the domain being periodic in streamwise and spanwise directions. The numerical setting is shown in Fig. 18 (left). Simulations are conducted over a domain of size $2.4 \times 5.0 \times 5.0$ mm³, in spanwise, transverse and streamwise directions, respectively. The overall mesh contains

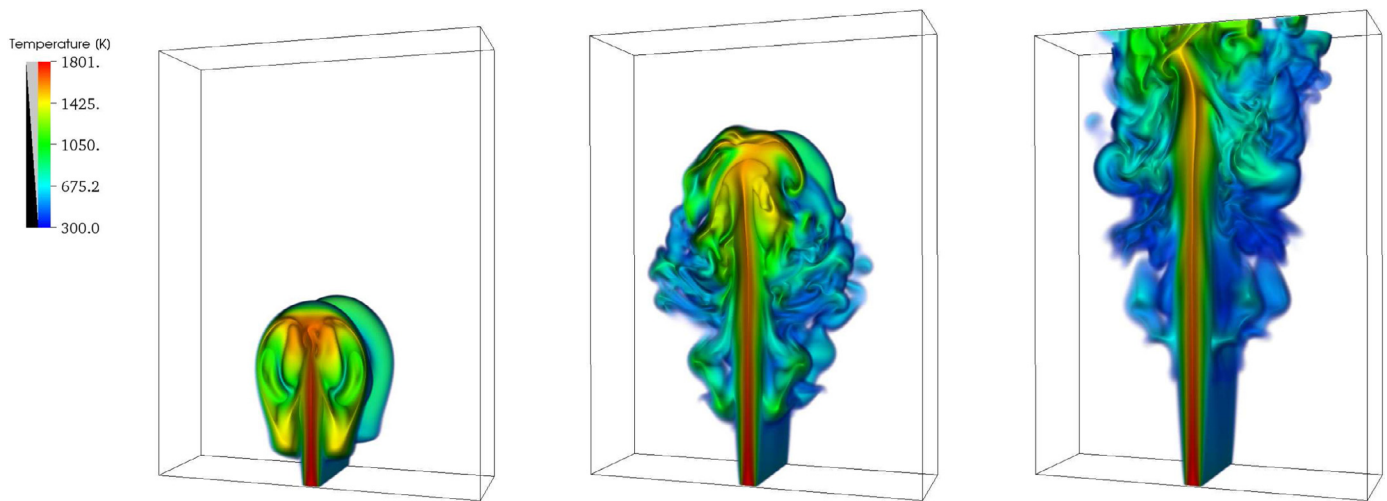


Fig. 17. Volume rendering of temperature at three different instants: $t = 30, 70$ and $130 \tau_{jet}$, respectively.

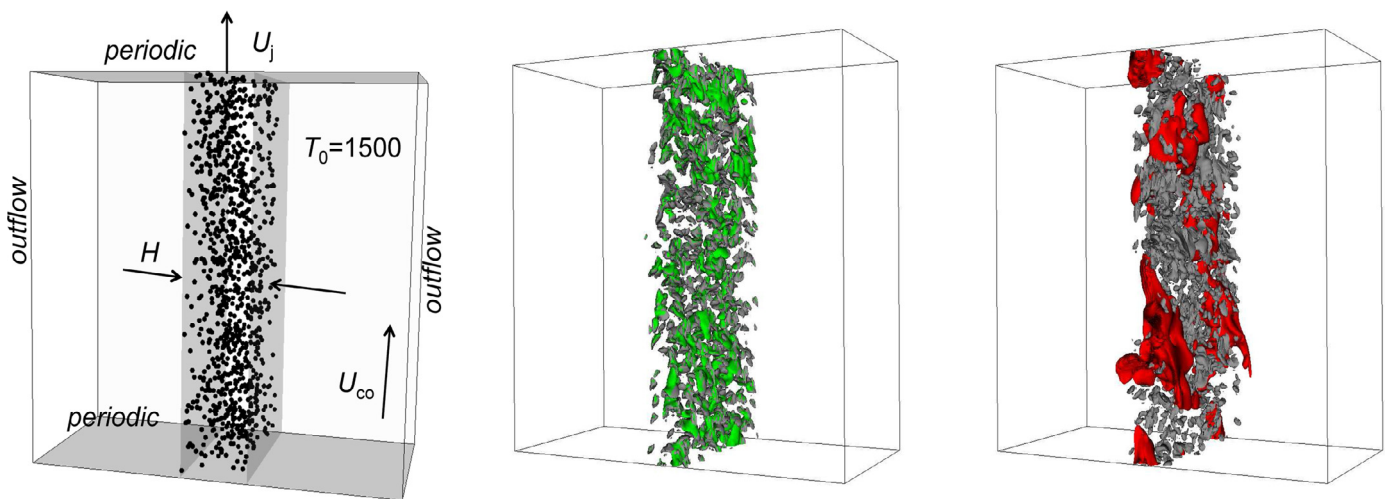


Fig. 18. Spray evaporating and burning in turbulence. Left: schematic diagram of the considered configuration. Center: isosurface of mass fraction (0.015) of *n*-heptane in the gas phase and of temperature (1106 K) at $t = 5 \tau_j$, shown in gray and green, respectively. Right: isosurface of mass fraction (0.0174) of *n*-heptane in the gas phase and of temperature (1800 K) at $t = 11 \tau_j$, shown in gray and red, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

more than 8 millions grid points to ensure a maximum mesh size of $20 \mu\text{m}$ in all directions. In the present work, a *n*-heptane skeletal mechanism accounting for 44 species and 114 reactions [70] is used for kinetics. Fig. 18 (center and right) show the time evolution of temperature and *n*-heptane mass fraction found in the gas phase (thus, after evaporation).

As expected from the physics of burning liquid droplets, the surrounding gas temperature first decreases due to evaporation. As observed in Fig. 18 (center), the liquid droplets start to evaporate into the gas mixture (gray isosurface), hereby consuming energy from the surrounding mixture, leading to a decrease in the surrounding temperature (green isosurface). After a while, ignition starts at different locations in the flow simultaneously, as illustrated by the red isosurface in Fig. 18 (right). In this figure, the red isosurface of temperature correspond to a temperature of 1800 K and thus to ignited pockets, with an increase of 300 K compared to the initial gas temperature ($T_0 = 1500$ K). This application demonstrates that DINOSOARS can be used to consider turbulent reacting two-phase flows. Further applications will be presented in dedicated publications in the interest of space.

10. Conclusions

A new software allowing the Direct Numerical Simulation of turbulent reacting and/or two-phase flows, called DINOSOARS, has been presented in this work. It contains both a low-Mach number formulation for reacting flows and an incompressible formulation for non-reacting configurations. While reaching high order in space and time, it can be efficiently used on large parallel supercomputers, as demonstrated for more than 10, 000 cores on SuperMUC. The parallelization relies on an efficient 2D domain decomposition thanks to the 2DECOMP&FFT library. The kinetics, transport and thermodynamic properties are computed by coupling with Cantera 1.8 and Eglib 3.4. For stiff systems, a fourth-order split semi-implicit Runge–Kutta approach based on Radau-5 and a third-order additive Runge–Kutta method have been implemented. Two immersed boundary approaches are available in DINOSOARS: (1) a direct boundary IBM is activated to describe complex geometries in the DNS, and (2) a direct force IBM is used to represent large particles for two-phase simulations, while small particles are described as point-particles. All these features make DINOSOARS a promising tool to investigate complex turbulent flows involving

reactions and/or particles or droplets, as demonstrated at the end of the article.

The code has been carefully verified and validated by using a variety of analytical solutions and experimental results. An excellent agreement has been obtained for all test-cases. The authors hope that this detailed description might provide useful guidelines concerning verification and validation for other researchers developing high-accuracy simulation tools for turbulent reacting flows.

Acknowledgements

The financial support of the DFG (Deutsche Forschungsgemeinschaft) for Abouelmagd Abdelsamie within the project FOR1447 “Physicochemical-based models for the prediction of safety-relevant ignition processes,” and for Timo Oster within the project TH 692/12 is gratefully acknowledged. Felix Dietzsch acknowledges financial support by the Federal Ministry of Education and Research of Germany in the framework of Virtuhcon (project number 03Z2FN11). The computer resources provided by the Gauss Center for Supercomputing/Leibniz Supercomputing Center Munich under grant pro84qo have been essential to obtain the DNS results presented in this work. DinoFamer has been developed with the support of Raphael Badel. Last but not least, very interesting discussions with Christian Hasse and Michael Gauding at TU Freiberg are gratefully acknowledged.

References

- [1] Thévenin D, Behrendt F, Maas U, Przywara B, Warnatz J. Development of a parallel direct simulation code to investigate reactive flows. *Comput Fluids* 1996;25(5):485–96.
- [2] Hilbert R, Thévenin D. Influence of differential diffusion on maximum flame temperature in turbulent non-premixed hydrogen/air flames. *Combust Flame* 2004;138:175–87.
- [3] Thévenin D, Gicquel O, de Charentenay J, Hilbert R, Veynante D. Two- versus three-dimensional direct simulations of turbulent methane flame kernels using realistic chemistry. *Proc Combust Inst* 2003;29:2031–9.
- [4] Thévenin D. Three-dimensional direct simulations and structure of expanding turbulent methane flames. *Proc Combust Inst* 2005;30:629–37.
- [5] Oran ES, Boris JP. Numerical simulation of reactive flows. New York: Elsevier; 1987.
- [6] Reynolds WC. The potential and limitations of direct and large eddy simulation. Lecture notes in physics. Berlin, Heidelberg:Springer; 1990;357:313–43.
- [7] Poinso T, Veynante D, Candel S. Quenching processes and premixed turbulent combustion diagrams. *J Fluid Mech* 1991;228:561–606.
- [8] Baum M, Poinso T, Haworth DC, Darabiha N. Direct numerical simulation of $H_2/O_2/N_2$ flames with complex chemistry in two-dimensional turbulent flows. *J Fluid Mech* 1994;281:1–32.
- [9] Poinso T, Candel S, Trouvé A. Applications of direct numerical simulation to premixed turbulent combustion. *Prog Energy Combust Sci* 1996;21:531–76.
- [10] Vervisch L, Veynante D. Direct numerical simulation of non-premixed turbulent flames. *Annu Rev Fluid Mech* 1998;30:655–91.
- [11] Thévenin D, Behrendt F, Maas U, Warnatz J. Simulation of reacting flows with a portable parallel code using dynamic load-balancing, in: High-performance computing and networking. Lecture notes in computer science. Berlin, Heidelberg: Springer; 1995;919:378–83.
- [12] Thévenin D, Behrendt F, Maas U, Warnatz J. Parallel simulation of reacting flows using detailed chemistry. Lecture notes in computer science. Berlin, Heidelberg:Springer; 1994;796:125–30.
- [13] Hawkes ER, Sankaran R, Sutherland JC, Chen JH. Scalar mixing in direct numerical simulations of temporally evolving plane jet flames with skeletal CO/H_2 kinetics. *Proc Combust Inst* 2007;32:1633–40.
- [14] Coussement A, Gicquel O, Fiorina B, Degrez G, Darabiha N. Multicomponent real gas 3-D-NSCBC for direct numerical simulation of reactive compressible viscous flows. *J Comput Phys* 2013;245:259–80.
- [15] Bansal G, Mascarenhas A, Chen JH. Direct numerical simulations of autoignition in stratified dimethyl-ether (DME)/air turbulent mixtures. *Combust Flame* 2015;162:688–702.
- [16] Bhagatwala A, Luo Z, Shen H, Sutton JA, Lu T, Chen JH. Numerical and experimental investigation of turbulent DME jet flames. *Proc Combust Inst* 2015;35:1157–66.
- [17] Duret B, Reveillon J, Menard M, Demoulin FX. Improving primary atomization modeling through DNS of two-phase flows. *Int J Multiphase Flow* 2013;55:130–7.
- [18] Kitano T, Nishio J, Kurose R, Komori S. Effects of ambient pressure, gas temperature and combustion reaction on droplet evaporation. *Combust Flame* 2014;161:551–64.
- [19] de Charentenay J, Thévenin D, Zamuner B. Comparison of direct numerical simulations of turbulent flames using compressible or low-Mach number formulations. *Int J Numer Meth Fluids* 2002;39:497–515.
- [20] Poinso T, Veynante D. Theoretical and numerical combustion. 3rd. Bordeaux, France: Aquaprint; 2011.
- [21] Majda A, Sethian J. The derivation and numerical solution of the equations for zero Mach number combustion. *Combust Sci Technol* 1985;42:185–205.
- [22] Hilbert R, Tap F, El-Rabii H, Thévenin D. Impact of detailed chemistry and transport models on turbulent combustion simulations. *Prog Energy Combust Sci* 2004;30:61–117.
- [23] Ern A, Giovangigli V. Multicomponent transport algorithms. Lecture notes in physics monographs. Berlin, Heidelberg: Springer; 1994;24.
- [24] Ern A, Giovangigli V. Fast and accurate multicomponent transport property evaluation. *J Comput Phys* 1995;120(1):105–16.
- [25] Fru G, Janiga G, Thévenin D. Impact of volume viscosity on the structure of turbulent premixed flames in the thin reaction zone regime. *Flow Turb Combust* 2012;88:451–78.
- [26] Li N, Laizet S. 2DECOMP&FFT - a highly scalable 2D decomposition library and FFT interface. In: Cray user's group 2010 conference; 2010.
- [27] Laizet S, Lamballais E, Vassilicos JC. A numerical strategy to combine high-order schemes, complex geometry and parallel computing for high resolution DNS of fractal generated turbulence. *Comput Fluids* 2010;39:471–84.
- [28] Laizet S, Li N. Incompact3d: a powerful tool to tackle turbulence problems with up to $O(10^6)$ computational cores. *Int J Numer Meth Fluids* 2011;67:1735–57.
- [29] Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical Recipes in FORTRAN. 2nd. New York: Cambridge University Press; 1992.
- [30] Kim J, Moin P. Application of a fractional-step method to incompressible Navier–Stokes equations. *J Comput Phys* 1985;59:308–23.
- [31] Brown D, Cortez R, Minion M. Accurate projection methods for the incompressible Navier–Stokes equations. *J Comput Phys* 2001;168(2):464–99.
- [32] Lucci F, Ferrante A, Elghobashi S. Modulation of isotropic turbulence by particles of Taylor-length-scale size. *J Fluid Mech* 2010;650:5–55.
- [33] Najm HN, Wyckoff PS, Knio OM. A semi-implicit numerical scheme for reacting flow. I. stiff chemistry. *J Comput Phys* 1998;143:381–402.
- [34] Knio OM, Najm HN, Wyckoff PS. A semi-implicit numerical scheme for reacting flow. II. stiff, operator-split formulation. *J Comput Phys* 1999;154:428–67.
- [35] Hairer E, Wanner G. Solving ordinary differential equations II: stiff and differential-algebraic problems. 2nd revised. Berlin: Springer-Verlag; 1999.
- [36] Caudal J, Fiorina B, Labégorre B, Gicquel O. Modeling interactions between chemistry and turbulence for simulations of partial oxidation processes. *Fuel Process Technol* 2015;134:231–42.
- [37] Zhong X. Additive semi-implicit Runge–Kutta methods for computing high speed non equilibrium reactive flows. *J Comput Phys* 1996;128:19–31.
- [38] Cooley JW, Lewis PAW, Welsh PD. The fast Fourier transform algorithm: programming considerations in the calculation of sine, cosine and Laplace transforms. *J Sound Vib* 1970;12:315–37.
- [39] Swartztrauber PN. Symmetric FFTs. *Math Comput* 1986;47:323–46.
- [40] http://computation.llnl.gov/project/linear_solvers/software.php.
- [41] Baum M, Poinso T, Thévenin D. Accurate boundary conditions for multicomponent reactive flows. *J Comput Phys* 1995;116:247–61.
- [42] Kraichnan RH. Diffusion by a random velocity field. *Phys Fluids* 1970;13:22–31.
- [43] Klein M, Sadiqi A, Janicka J. A digital filter based generation of inflow data for spatially developing direct numerical or large eddy simulation. *J Comput Phys* 2002;186:625–65.
- [44] Kempf A, Klein M, Janicka J. Efficient generation of initial-and inflow-conditions for transient turbulent flows in arbitrary geometries. *Flow Turb Combust* 2005;74:67–84.
- [45] Anderson JD. Computational fluid dynamics: the basics with applications. New York: McGraw-Hill; 1995.
- [46] Tannehill JC, Anderson DA, Pletcher RH. Computational fluid mechanics and heat transfer. 2nd. Washington: Taylor & Francis; 1997.
- [47] Farrashkhalvat M, Miles JP. Basic structured grid generation: with an introduction to unstructured grid generation. Oxford: Butterworth-Heinemann; 2003.
- [48] Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *J Comput Phys* 2000;161:35–60.
- [49] Mittal R, Iaccarino G. Immersed boundary methods. *Annu Rev Fluid Mech* 2005;37:239–61.
- [50] Steinman DA, Hoi Y, Fahy P, Morris L, Walsh MT, Aristokleous N, et al. Variability of computational fluid dynamics solutions for pressure and flow in a giant aneurysm: the ASME 2012 summer bioengineering conference CFD challenge. *J Biomech Eng* 2013;135:021016.
- [51] Eshghinejadfard A, Abdelsamie A, Janiga G, Thévenin D. Direct-forcing immersed boundary lattice Boltzmann simulation of particle/fluid interactions for spherical and non-spherical particles. *Particuology* 2016;25:93–103.
- [52] Abdelsamie A, Thévenin D. Direct numerical simulation of burning *n*-heptane droplets: auto-ignition and turbulence modulation mechanisms. In: ERCOFTAC workshop Direct and Large-Eddy Simulation 10. Springer; 2015. p. pp152.
- [53] Abdelsamie A, Lee C. Decaying versus stationary turbulence in particle-laden isotropic turbulence: Turbulence modulation mechanism. *Phys Fluids* 2012;24:015106.
- [54] Abdelsamie A, Lee C. Decaying versus stationary turbulence in particle-laden isotropic turbulence: heavy particle statistics modifications. *Phys Fluids* 2013;25:033303.

- [55] Uhlmann M. An immersed boundary method with direct forcing for the simulation of particulate flows. *J Comput Phys* 2005;209:448–76.
- [56] Abdelsamie A, Thévenin D. Modulation of isotropic turbulence by resolved and non-resolved spherical particles. In: Fröhlich J, editor. *Direct and Large-Eddy Simulation IX*. Cham, Switzerland: ERCOFTAC Series, Springer; 2015. p. 621–9.
- [57] Abdelsamie A, Eshghinejadfard A, Oster T, Thévenin D. Impact of the collision model for fully resolved particles interacting in a fluid. In: *Proceedings of the 4th ASME 2014 joint US-European fluids engineering division summer meeting* (Chicago, IL). ASME; 2014.
- [58] Comte-Bellot G, Corrsin S. Simple eulerian time correlation of full and narrow band velocity signals in isotropic turbulence. *J Fluid Mech* 1971;48(2):273–337.
- [59] Wang ZJ, Fidkowski K, Abgrall R, Bassi F, Caraen D, Cary A, et al. High-order CFD methods: current status and perspective. *Int J Numer Meth Fluids* 2013;72:811–45.
- [60] DeBonis JR. Solutions of the Taylor–Green vortex problem using high-resolution explicit finite difference methods. *AIAA Pap* 2013-0382 2013.
- [61] van Rees WM, Leonard A, Pullin DI, Koumoutsakos P. A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers. *J Comput Phys* 2011;230:2794–805.
- [62] Moser R, Kim J, Mansour NN. Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$. *Phys Fluids* 1999;11(4):943–5.
- [63] Vreman AW, Kuerten JGM. Comparison of direct numerical simulation databases of turbulent channel flow at $Re_\tau = 180$. *Phys Fluids* 2014;26:015102.
- [64] Niu XD, Shu C, Chew YT, Peng Y. A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows. *Phys Lett A* 2006;354:173–82.
- [65] Dennis SCR, Chang G-Z. Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100. *J Fluid Mech* 1970;42(3):471–89.
- [66] Niu XD, Chew YT, Shu C. Simulation of flows around an impulsively started circular cylinder by Taylor series expansion- and least squares-based lattice Boltzmann method. *J Comput Phys* 2003;188:176–93.
- [67] Chemical kinetic mechanism for combustion applications, Center for energy research (Combustion division), University of California at San Diego, <http://web.eng.ucsd.edu/mae/groups/combustion/mechanism.html>.
- [68] Luo Z, Yoo CS, Richardson ES, Chen JH, Law CK, Lu T. Chemical explosive mode analysis for a turbulent lifted ethylene jet flame in highly-heated co-flow. *Combust Flame* 2012;159:265–74.
- [69] Patel A, Kong S, Reitz R. Development and validation of a reduced reaction mechanism for HCCI engine simulations. *SAE Tech Pap* 2004-01-0558 2004. Doi:10.4271/2004-01-0558
- [70] Liu S, Hewson JC, Chen JH, Pitsch H. Effects of strain rate on high-pressure nonpremixed *n*-heptane autoignition in counterflow. *Combust Flame* 2004;137:320–39.
- [71] Jachimowski CJ. An experimental and analytical study of acetylene and ethylene oxidation behind shock waves. *Combust Flame* 1977;29:55–66.
- [72] Baker JA, Kinner GB. Shock-tube studies on the ignition of ethylene-oxygen-argon mixtures. *Combust Flame* 1972;19:347–50.
- [73] Kumar K, Mittal G, Sung C-J, Law CK. An experimental investigation of ethylene/O₂/diluent mixtures: laminar flame speeds with preheat and ignition delays at high pressures. *Combust Flame* 2008;153:343–54.
- [74] Ciezki HK, Adomeit G. Shock-tube investigation of self-ignition of *n*-heptane-air mixtures under engine relevant condition. *Combust Flame* 1993;93(4):421–33.
- [75] Egolfopoulos FN, Zhu DL, Law CK. Experimental and numerical determination of laminar flame speeds: mixtures of C₂-hydrocarbons with oxygen and nitrogen. *Proc Combust Inst* 1990;23:471–8.
- [76] Jomaas G, Zheng XL, Zhu DL, Law CK. Experimental determination of counterflow ignition temperatures and laminar flame speeds of C₂-C₃ hydrocarbons at atmospheric and elevated pressures. *Proc Combust Inst* 2005;30:193–200.
- [77] Hassan MI, Aung KT, Kwon OC, Faeth GM. Properties of laminar premixed hydrocarbon/air flames at various pressures. *J Propul Power* 1998;14(4):479–88.
- [78] Ayachit U. *The paraview guide: a parallel visualization application*. Clifton Park, New York: Kitware Inc.; 2015.
- [79] Bauer AC, Geveci B, Schroeder W. *The paraview catalyst user's guide*. Clifton Park, New York: Kitware Inc.; 2015.
- [80] Schroeder W, Martin K, Lorensen B. *Visualization toolkit: an object-oriented approach to 3D graphics*. 4th ed. Clifton Park, New York: Kitware Inc.; 2006.
- [81] Mizobuchi Y, Tachibana S, Shinio J, Ogawa S, Takeno T. A numerical analysis of the structure of a turbulent hydrogen jet lifted flame. *Proc Combust Inst* 2002;29(2):2009–15.