Available online at www.sciencedirect.com

ScienceDirect

Proceedings of the Combustion Institute 000 (2022) 1–35

Proceedings of the Combustion Institute

www.elsevier.com/locate/proci

HPC-enabling technologies for high-fidelity combustion simulations

Daniel Mira*, Eduardo J. Pérez-Sánchez, Ricard Borrell, Guillaume Houzeaux

Barcelona Supercomputing Center (BSC), Plaéa Eusebi Güell, Barcelona, 1–3 08034, Spain

Received 22 February 2022; accepted 25 July 2022
Available online xxx

Abstract

With the increase in computational power in the last decade and the forthcoming Exascale supercomputers, a new horizon in computational modelling and simulation is envisioned in combustion science. Considering the multiscale and multiphysics characteristics of turbulent reacting flows, combustion simulations are considered as one of the most computationally demanding applications running on cutting-edge supercomputers. Exascale computing opens new frontiers for the simulation of combustion systems as more realistic conditions can be achieved with high-fidelity methods. However, an efficient use of these computing architectures requires methodologies that can exploit all levels of parallelism. The efficient utilization of the next generation of supercomputers needs to be considered from a global perspective, that is, involving physical modelling and numerical methods with methodologies based on High-Performance Computing (HPC) and hardware architectures. This review introduces recent developments in numerical methods for large-eddy simulations (LES) and direct-numerical simulations (DNS) to simulate combustion systems, with focus on the computational performance and algorithmic capabilities. Due to the broad scope, a first section is devoted to describe the fundamentals of turbulent combustion, which is followed by a general description of state-of-the-art computational strategies for solving these problems. These applications require advanced HPC approaches to exploit modern supercomputers, which is addressed in the third section. The increasing complexity of new computing architectures, with tightly coupled CPUs and GPUs, as well as high levels of parallelism, requires new parallel models and algorithms exposing the required level of concurrency. Advances in terms of dynamic load balancing, vectorization, GPU acceleration and mesh adaptation have permitted to achieve highly-efficient combustion simulations with data-driven methods in HPC environments. Therefore, dedicated sections covering the use of high-order methods for reacting flows, integration of detailed chemistry and two-phase flows are addressed. Final remarks and directions of future work are given at the end.

Keywords: High-Performance Computing (HPC); Exascale; Combustion; Multiphase flow; Chemistry; high-order methods; Graphics Processing Unit (GPU)

* Corresponding author.
E-mail address: daniel.mira@bsc.es (D. Mira).

https://doi.org/10.1016/j.proci.2022.07.222

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

1. Introduction

Combustion is a relevant process in many industrial and energy-related sectors with high impact for the society. In particular, it is key in energy conversion systems and it is expected to play a fundamental role in the energy transition. However, its intrinsically complex nature and the close interaction between chemistry and the turbulent flow makes many of the involved phenomena far from being fully understood. Example of this is how the fuel oxidation, pollutant formation or flame stability are critically affected by the turbulent nature of the velocity and scalar fields. On top of that, in liquid fuel systems, atomization and evaporation phenomena are added to the list of challenging simulation tasks. In spray combustion, after fuel injection, the liquid core is broken into ligaments and eventually droplets, which are subsequently transported, exchanging heat and mass with the surroundings. The size of the droplets usually change between applications, for instance, larger droplet sizes are found in atmospheric spray burners compared to high-pressure injectors. In realistic conditions, the turbulent flow interacts with the chemical reactions that usually take place in thin layers, and lead to a complex interplay of these two processes at the smallest scales of the flow. Other phenomena such as flame/acoustics interactions, localized extinction or droplet/flame interactions also affect the dynamics of the flame and require complex modelling closures.

Combustion involves a wide range of scales. Only from the chemistry side, it goes from the fastest radicals to the slowest species, usually associated with pollutant formation such as NOx or soot. Therefore, advanced multiscale and multiphysics models are needed to obtain detailed descriptions of these physical processes. The development of high-fidelity methods to simulate unsteady combustion phenomena has been a topic of intensive research in the last decades. Numerical simulations have made significant contributions to the understanding of fundamental processes in reacting flows and its interaction with the turbulent multiphase flow environment. These simulations have historically been one of the most demanding applications running on cutting-edge supercomputers. The use of High-Performance Computing (HPC) has extended the capabilities of numerical solvers, expanding the scope of Direct Numerical Simulations (DNS) and Large-Eddy Simulations (LES) towards realistic engine conditions. Examples from DNS include the pioneering work from Yoo et al. [1] using 30,000 CPU-cores; DNS of soot formation in jet flames using up to 32,768 CPU-cores [2]; and more recently Luca et al. [3] executing on up to 131,072 processors. In the context of LES, recent large-scale simulations of an integrated fan, compressor, and combustion chamber system using up to 16,384 CPU-cores was conducted by Arroyo at al. [4]; while Che at al. [5] run a scramjet engine simulation on 998,400 CPU-cores. Such level of computational requirements arises as a consequence of the multiphysics and multiscale nature of the combustion process what evidences the needs of this community to have efficient computational strategies to address problems of practical interest. In fact, most high-fidelity combustion simulations have been conducted for laboratory flames and atmospheric conditions [3,4,6], since the computational cost to tackle larger systems at higher operating pressure, and with more detailed description of the oxidation and pollutant formation can be prohibitive even with modern supercomputers.

The need for higher (beyond Petascale) computing resources to increase simulations reliability makes combustion simulations an Exascale application. Hence, the efficient use of upcoming Exascale architectures is critical, requiring new algorithms and computational strategies based on combining physical models, numerical methods, and parallel algorithms optimized for specific hardware architectures.

HPC drives research and enables industrial innovation in many fields, but the techniques need to be adapted to the requirements of the applications. Some workflows require the aggregation of large numbers of loosely coupled runs, while others need tightly coupled large-scale executions. Combustion simulation workflows can fall into both categories and those aspects will be discussed in the next sections. Undoubtedly, when we consider high-fidelity combustion simulations, the resources necessary to capture the complexity of the physics exceed the capabilities of current supercomputers. That is why we can consider computational combustion as a candidate for the effective exploitation of future supercomputers. The next milestone in supercomputing performance is the ExaFlop - 10^{18} floating-point operations per second (FLOPs). Unlocking this level of performance for combustion simulation, provided in increasingly complex hardware architectures, requires prioritizing high-performance computing aspects in the development of simulation frameworks, along with modeling and numerical strategies. Thus, the development of specific algorithms with HPC is key to develop efficient and reliable methodologies to study turbulent reacting flows as those found in reciprocating combustion engines, gas turbines, rockets or furnaces and industrial combustors [4,5,7,8].

An important aspect when running on Exascale systems is the computational performance of the application, since inefficiencies of the code or suboptimal implementations can represent a massive waste of computing resources. Therefore, HPC-related aspects such as exploiting heterogeneous systems with GPU accelerators, load-balancing methods, data locality optimization, or chip-level

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

vectorization should not be overlooked when developing simulation platforms. These HPC techniques can boost the performance of the different building blocks involved in the simulation of combustion systems. In this paper, a review of these strategies with focus on three key areas is conducted: applying high-order methods, including detailed chemistry, and extending combustion solvers to multiphase flow problems.

This paper provides an outlook of the state-of-the-art of these three topics in terms of numerical simulation methodologies, but with focus on the HPC-related aspects and their impact on high-fidelity simulations. Most examples of the present review will focus on general combustion problems, with a dedicated section to combustion of liquid fuels. The structure of this review paper is given as follows. First, a general description of the multiphysics and multiscale characteristics of turbulent reacting flows is given. The analysis is focused on the description of different methodologies and numerical methods to solve these problems. It is followed by an introduction of the possibilities that Exascale computing brings to combustion science by extrapolation of a reference DNS with a Reynolds and pressure scaling. Second, a general discussion of different HPC strategies used to accelerate and increase the parallel efficiency in combustion simulations is presented. Finally, three additional sections describing the advances in high-order methods for reacting flows, chemistry descriptions and multiphase flow methods for spray flame simulations are given with focus on their algorithmic and HPC aspects. Final remarks and directions for future work are given at the end.

2. High-fidelity combustion simulations

The accurate prediction of multiphase reacting flows in practical combustion systems arises as one of the most complex applications in computational science due to the coexistence of strongly coupled physical and chemical phenomena. This complexity, coming from the broad range of spatial and temporal scales of the turbulence, chemistry, and liquid-gas interactions, in the case of liquid fuels, has led to intensive research in physical modelling and numerical methods [9]. As more challenging combustion problems are tackled, solutions require more comprehensive approaches, including physical models, numerical methods, algorithms, and software implementations. In this new paradigm, computational combustion science evolves simultaneously with new HPC architectures.

This section starts with a brief introduction to the fundamentals of combustion, from gaseous to liquid fuels, in order to describe the context of application and highlight the relevant phenomena to be captured with computational models. It is then followed by a summary of the state-of-the-art on modelling and numerical strategies to simulate these flows and finally, a summary of the computational requirements of these methods in HPC architectures is given.

2.1. Combustion physics

Combustion introduces a set of complex phenomena, which is strongly coupled to the turbulent flow and needs to be correctly addressed to capture the nature of the flames that are present in practical combustion systems. For instance, as the gas mixtures undergo exothermic reactions, an expansion produced by thermal dilatation is known to have an impact on the turbulent flow. Another essential aspect is the mixing of fuel and oxidizer, since the rate of mixing can severely affect the reacting layer, influencing the evolution of the chemical reactions and can lead to complex phenomena like localized extinction, re-ignition or quenching [10].

From the general description of turbulent reacting flows, different combustion regimes can be identified depending on the process of fuel/air mixing, which are usually referred as premixed and non-premixed combustion [11]. In the former, the reacting zone propagates in a homogeneous mixture of reactants, while in the latter the mixing and reaction takes place simultaneously. In realistic applications, the reacting front often experiences a blend of these regimes leading to the so-called, partially premixed flames [12].

Irrespective of the combustion regime, the flame propagation is governed by the same fundamental process: i) the mixture of reactants is heated to a temperature where reactions happen at a significant rate, ii) intermediate radicals and final products are obtained, and heat is released due to the exothermic nature of the process, iii) heat and chemically active radicals diffuse towards the fresh reactant mixture, thus sustaining the process. The rate of production and consumption of the involved chemical species is described by the chemical mechanism. These mechanisms can become extraordinarily complex, characterizing many oxidation pathways and involving a large number of different intermediate species. The large computational cost of high-fidelity turbulent reacting flow simulations is partly associated to the complexity of the reaction process. Despite reduced mechanisms may also capture key aspects of the flame propagation [13], specific reaction pathways are crucial for the correct prediction of complex chemical phenomena like the negative-temperature coefficient behaviour [10], or the burning rates during flame stratification [14,15]. Therefore, there is a need to include detailed reaction mechanisms that can describe the chemical evolution of the flow at a wide range of conditions [10]. Larger chemical mechanisms are also needed for the accurate prediction of pollutant formation, such as nitrogen oxides (NO_x) or soot. Soot formation is particularly sen-

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

sitive to the reaction mechanism, since the formation of these carbon agglomerates is initiated from large aromatic molecules. The accurate prediction of these soot-precursors entails the usage of detailed chemical mechanisms [16–18].

Once an appropriate chemical mechanism is selected, the turbulent reacting flow can be described by a set of partial differential equations (PDEs). This includes the Navier-Stokes equations describing mass and momentum conservation, conservation of energy, and the set of equations describing the chemical state of the flow. This system of PDEs is composed by 5 equations (continuity, energy and the three velocity components), along with a given set of equations describing the chemical state, which depend on the selected reaction mechanism. For instance, when doing a direct integration of the chemical problem like in DNS, an equation is solved for each chemical species. The PDE describing the evolution of the kth species is given by:

$$\frac{\partial(\rho Y_k)}{\partial t} + \nabla \cdot (\rho \mathbf{u} Y_k) = -\nabla \cdot (\rho \mathbf{V_k} Y_k) + \dot{\omega}_k \qquad (1)$$

where ρ is the density, Y_k is the mass fraction of the kth species, and $\mathbf{u}$ is the velocity. The first term on the right hand side describes the diffusion of species given by the diffusion velocity $\mathbf{V_k}$, often described by Fick's law. It is worth mentioning that detailed computation of multi-component diffusion is computationally expensive and dedicated studies to reduce this cost can be found in the literature [19]. Finally, $\dot{\omega}_k$ is the net chemical source term of the kth species, which is determined by the elementary reactions of the mechanism involving the kth species. In general, the solution of the species equations, usually dominates the computational cost in detailed chemistry simulations [10], not only because of the large number of species to be transported with the flow, but also because of the cost of the chemical integration of the highly non-linear source terms $\dot{\omega}_k$. Therefore, advanced methods are required to accelerate the chemistry calculations, specially for conditions involving stiff chemistry [20].

The simulation of spray flames introduces an additional level of complexity, as it requires a detailed description of the two phases, liquid and gas, and their interactions. In realistic systems, the different physical phenomena are distributed unevenly throughout the simulation domain. For instance, the liquid phase and the reaction front, both associated with high local computational cost, are only present in specific regions of the domain. A representation of the physical processes involved in the combustion of liquid fuels is shown in Fig. 1.

After the fuel is injected in the combustion chamber, the inertial forces along with the surrounding gas phase turbulence introduce instabilities at the liquid-gas interface, which result in the formation of elongated ligaments and large

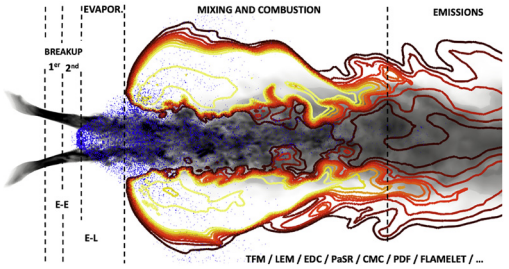

Fig. 1. The representation of a spray flame from a pressure-atomizer illustrates the different processes involved in combustion systems. Such physical processes are given on the top side, while the different methodologies applied to solve these problems are listed below..

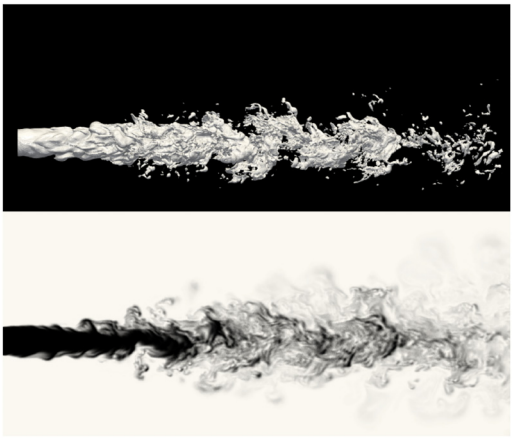

Fig. 2. Primary atomization visualization for a round jet exiting at Re=4659 and We=7239 computed with level set approach in the context of LES [29]. Top: snapshot of an iso-surface of liquid volume fraction at $\phi = 0.5$, bottom: snapshot of a cross-plane of the liquid volume fraction.

droplets. This process is referred to as primary breakup, and it is mainly influenced by the slip velocity between the two phases and the surface tension [21]. Further downstream, the aerodynamic forces acting on the droplets break them into smaller ones in the so-called secondary breakup mechanism [22]. Primary atomization is fundamentally influenced by the internal flow of the atomizer apparatus [23], and by the turbulent interaction of the liquid with the gas phase. The most common strategies to achieve effective atomization include high pressure single and multi-hole injectors in internal combustion engines [24,25], and air-blast and pressure-swirl atomizers for gas turbines [6,26,27]. New concepts like multipoint injectors [28] have also been derived for aeronautical applications. While the process is governed by the same principles in all systems, the formation of the spray is fundamentally different. In case of hole-

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

type injectors, a high speed liquid jet interacts with the gas phase, while in pressure-swirl atomizers, a high speed liquid sheet is formed and destabilized. A more complex atomization process occurs in air-blast atomizers, since the liquid is deposited on a pre-filmer surface and ligaments are carried away by the fast air flow.

A visualization of the liquid core and primary breakup process of a single-hole injection system is shown in Fig. 2 for a round jet exited at Reynolds Re=4659 and Weber We=7239 computed with a conservative level set method in the context of LES [29]. The figure shows the formation of ligaments and droplets during a primary breakup event. The arising ligaments are further atomized in the secondary breakup process.

In general, specific physical processes, like the correct representation of surface tension, are only relevant close to the atomizer apparatus. Consequently, the primary breakup process is often treated separately in high-fidelity simulations of the system, either with dedicated simulations or by phenomenological models. However, a holistic treatment of the system may be advocated in cases where the turbulent reacting flow of the surroundings may have a significant effect on primary atomization. For instance, in case of air-blast atomizers, the high speed gas flow has to be correctly characterised in the air inlet manifold and heat and mass transfer may be significant in the liquid film. Also, in liquid jets interacting with a reacting cross-flow, the atomization is highly affected by the gas phase fluctuations [30].

The majority of heat and mass transfer takes place after the liquid jets or sheets are atomized, because atomization increases the specific surface of the gas-liquid interface drastically by the creation of small droplets. The heat transferred from the gas phase to the droplets causes the liquid fuel to evaporate, acting as local sources of fuel vapor that mixes with the surrounding air. Evaporation may be concluded before the flame front is reached, as in the case of compression ignition engines in the lift-off region [31], but droplet flame interactions can also occur including single droplets sustaining a reaction zone around them [32] or locally extinguishing the flame [6]. In such conditions, the flame can be considered partially premixed, and models based on a diffusion flame structure are also performing considerably well for predicting the flame behaviour [6,33,34]. Note that the two phase representation of the fuel-oxidizer mixing becomes less relevant as the pressure increases and the evaporation is replaced by transcritical heat and mass transfer models [35,36].

2.2. Methodologies for turbulent multiphase reacting flows

Three methodologies prevail in the description of turbulent flow fields, the Reynolds Averaged Navier-Stokes (RANS) approach, LES, and DNS. RANS is popular at the industrial level due to the fact that solutions are obtained at a fraction of the CPU cost of the other methods. This can be attributed to the advances in RANS modelling, parallel computing strategies and multi-grid methods [37]. However, RANS methods are not suitable for the description of intermittent and highly transient phenomena, which can be only reliably predicted by unsteady methods like LES and DNS.

The use of LES and DNS approaches have been widely extended in the last two decades thanks, in part, to the development of hardware and software for parallel computing. In LES, quantities are spatially filtered and only scales above the cut-off filter length are resolved, while the rest are modelled by closure rules. In contrast, DNS solves all the spatial and temporal scales of the flow and requires high spatial and temporal resolution. The requirements in computational cost of a DNS simulation, measured in floating point operations, scales cubically with the integral Reynolds number Re_L (Re_L^3). [38]. Notwithstanding, in the case of turbulent flames, this situation is even more extreme, since the chemical scales are smaller than those of the flow. For instance, in the corrugated flamelet regime, the entire flame is embedded in the Kolmogorov eddies while in the thin reaction zone regime only the reaction layer is thinner than the Kolmogorov eddies. Therefore, accurate predictions of the chemical evolution for all major species and radicals in DNS make the simulations to be restricted to small domains and modest Reynolds numbers, despite important efforts have been done to break such barriers [1,3,39]. In the frame of LES, more realistic conditions can be achieved, as the governing equations are filtered and closure models are used to handle turbulence and turbulence-chemistry interactions [9]. The use of LES has been rapidly extended to a wide variety of flows, as it allows to resolve complex flow conditions in realistic environments at a reduced computational cost [4,40,41].

In the case of spray flames, a detailed description of the liquid phase is of high relevance, as the dynamics of the spray strongly influence the combustion process [42]. However, due to the large range of scales involved and the thin liquid gas interface, the exact solution of the problem with DNS at relevant engine conditions still remains unfeasible. Several approaches have been proposed in the literature to describe two-phase flows in combustion simulations, but two approaches are mostly common: interface capturing [43] methods and Lagrangian Particle Tracking (LPT) methods [44]. The description of the liquid phase with interface capturing methods usually requires higher resolution in order to resolve the interface with several cells and to avoid excessive numerical diffusion. Methods based on Volume of Fluid (VOF), level set or phase fields [45–47] can be computationally expensive, though they can largely benefit from the

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

use of Adaptive Mesh Refinement (AMR) techniques [48]. The modelling of large population of droplets by interface capturing methods is challenging in terms of cost, as it has to deal with the description of locally poly-disperse droplet populations that need to be resolved in fine meshes to account for secondary breakup and eventually, evaporation. Therefore, methods based on LPT are usually preferred for spray flames due to its simplicity and lower CPU cost [32].

In summary, the complete characterization of practical combustion systems is complex and brings challenges associated to various disciplines, from engineering, chemistry, physics, computer science and mathematics. Particularly challenging is the modelling of reacting sprays under realistic conditions, as it includes the interaction of multiphase phenomena with combustion chemistry and turbulence, namely, high pressure liquid fuel injection, atomization, evaporation, of single and multicomponent fuels fuel/air mixing and combustion. This interaction is, however, not yet well understood and additional research is required to properly describe these complex phenomena with today's modelling strategies and computational resources. The use of HPC can contribute to the progress of this field by exploiting the multilevel parallelism from modern supercomputers combined with the use of the most advanced numerical techniques and physical models. These aspects and their interactions will be described in the next sections.

2.3. Numerical methods

As mentioned previously, combustion involves a great diversity of physical phenomena and scales that can be tackled using a wide variety of numerical strategies that deal with one or several aspects of the problem. For instance, the overall methodology should include scalar transport equations with stiff chemical integration for the gas phase, interface capturing or particle tracking methods for the liquid phase, and turbulent/chemistry interaction models for the burning rates and flame dynamics. The strategies used to solve these coupled problems require the use of efficient algorithms and computational strategies that can be combined to ensure efficient and reliable calculations.

At the computational level, Eulerian methods rely on the numerical approximation of PDEs involving discretization over cells, faces or nodes of the mesh, depending on the numerical method considered: Finite Element (FE), Finite Volume (FV) or Finite Difference (FD). This operation is called *assembly* and results in matrix vector products in the case of explicit schemes, and the solution of a full system when using implicit or semi-implicit schemes. The second operation consists of obtaining the solution at the subsequent time step, which can be non-trivial for implicit methods, since the system can be large and stiff. In particular, the solution of the ordinary differential equations (ODEs) requires looping over the nodes of the mesh and the use of efficient techniques and pre-conditioners to accelerate the convergence of their highly nonlinear couplings. A special case is the particle transport used in Lagrangian methods, where temporal ODEs are used to characterize the particles (position, velocity, temperature, and composition in case of evaporating sprays) at each time step. In addition, Lagrangian setups introduce certain implementation complexity when solved using parallel computing, as parallel particle tracking methods are required to describe the motion of the particles across the domain. This aspect is addressed in Section 6.2.

In a complex setup, the efficient coupling of all these sub-processes is essential, even-though the physical coupling of the processes are well established. The complexity lies in the fact that most of combustion phenomena are local, leading to zones with higher computation loads (e.g. high particle concentration near the nozzle, high chemical reactivity regions leading to tightly coupled source terms in specific locations or thin liquid/gas interfaces). In terms of algorithms, this introduces difficulties in ensuring a balanced load among all the processors and requires the application of computational strategies to mitigate the possible imbalance. Strategies to achieve a balanced load at different levels of parallelism will be described in the next subsection.

2.4. Towards Exascale combustion simulations

Considering the current achievements in combustion simulations in the Petascale era, a thousand-fold increase in computing power by the onset of Exascale computing is bound to open new frontiers in combustion science. From current state-of-the-art in high-fidelity combustion simulations using DNS, it would be possible to consider additional phenomena like detailed descriptions of the liquid phase and thermal radiation. This will allow the capability to simulate applications at a significantly higher pressure and Reynolds number, closer to the typical values encountered in engines.

It would be possible to roughly estimate the requirements for the proposed simulations via extrapolation of the resources used for example in the soot DNS study of Attili et al. [2], using heptane fuel at atmospheric pressure and a Reynolds number of 15,000. That simulation employed 0.5 Billion grid points with 55 variables and 1 Billion Lagrangian tracers containing 10 variables each, altogether summing up a total of 40 Billion degrees of freedom. It was computed on the entire IBM Blue GeneP (65,536 cores, 222 Tera-flops of theoretical performance). An estimation of the computing costs could be evaluated for a simulation of the same flame geometry, with a pressure of

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

about 10 – 20 bar, and a jet Reynolds number of 100,000, considering a realistic liquid fuel (an aviation fuel surrogate) injected in the form of Lagrangian droplets [49]. The five-fold increase in Reynolds number and the ten-fold increase in pressure requires an increase of the number of grid points in each direction of x10, i.e., a factor of x1000 in three dimensions. Using a realistic liquid fuel chemistry description requires a reaction mechanism with about 100 species, which is twice the number of the species in the previous simulation, causing an additional increase by a factor of x2 in the number of degrees of freedom. Taking into account factors relevant to the use of a Lagrangian point-particle approach for the liquid fuel droplets and the longer integration time required, it is possible to estimate the increase in computational cost to be x10000 for the envisioned simulation as compared to the actual Petascale simulation. Assuming the same code efficiency with respect to the theoretical machine peak performance, a supercomputer with a peak performance of 222 Tera-flops $\times$ 10000 $\approx$ 2 Exa-flops would be needed to perform an accurate simulation at engine conditions in a reasonable time.

Exploiting the capabilities of the Exascale systems and benefiting from their increased peak performance is far from trivial, as it requires adapting the codes to the most recent programming standards and supercomputer architectures. Therefore, further efforts in computational methods and strategies are devised considering simultaneously the physical models, numerical methods and HPC strategies. The main concepts and methods to be considered are described in the next section.

3. High-Performance Computing for combustion simulations

Exascale computing enables the scaling in computing power up to 10^{18} FLOPs, which means an important jump for current simulation capabilities. For decades, the evolution of the computing power has been driven by the miniaturization of computing components, following an exponential trend known as the Moore's law [50]. Gordon Moore, founder of Intel, predicted in 1975 that the number of transistors per chip area would double roughly every two years and this trend has been maintained until few years back.

This miniaturization process, based on a continuous shrinking of transistors came along with an increase of the clock frequencies without increasing the power density. This favorable balance, known as Dennard scaling [51], was broken around 2005 due to the rise of the power static losses resulting in heating of chips. Consequently, the rise in the clock-speed halted, and further performance increases were mainly based on parallelism and concurrency. For example, by implementing multi-core processors and wider Single Instruction Multiple Data (SIMD) extensions at the CPU level. The increase in parallelism and concurrency in supercomputers seems to remain. With streams of operands functioning at 2.5GHz, $\mathcal{O}(10^8)$ concurrent streams are required to achieve Exascale performance.

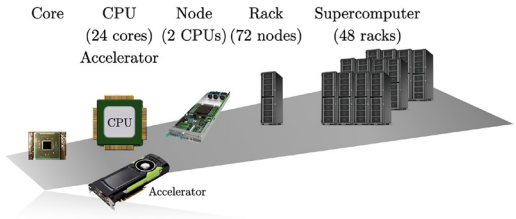

Fig. 3. Classical hierarchy of a supercomputer. The numbers in parenthesis refer to the configuration of Marenostrum 4 supercomputer, hosted at Barcelona Supercomputing Center[52].

The aforementioned concurrency is organized within the supercomputer's architecture through different levels. On the one hand, the parallelization within the node is referred to intra-node level. This includes the utilization of multi-core CPUs and GPUs. On the other hand, the inter-node level, or system level, refers to parallelizations using multiple nodes connected by a network. Various parallel models have to be combined on the software side to expose parallelism at all these levels. Figure 3 illustrates the implementation of this hierarchy in modern supercomputers. The supercomputer is made of racks and those are composed by nodes; then, each node includes several CPUs, while the final computing units are the cores. Accelerators like GPUs can also be connected to the nodes. Exascale systems will definitely involve such accelerators, which provide high throughput at lower energetic cost compared to actual multi-core CPUs.

At the level of the entire system, the capability of supercomputers has been historically evaluated using the LINPACK benchmark [53], which measures the performance of a supercomputer by solving a dense linear systems with random coefficients using a direct method. The Top500 list [54] ranks the systems measured using the LINPACK benchmark twice a year since 1993. A deceleration in performance is observed in the exponential trend, which can be related to the limitations of the hardware miniaturization. Note that LINPACK is a compute intensive benchmark which can represent compute bound applications. However many applications are bounded by the data transfers from the main memory to the cache memory, which is organized in various levels: they are referred to as memory bound. Those applications can be better represented by other benchmarks such as the HPCG [55], which focuses on sparse linear algebra computations. In that case, in terms of FLOPs the percentage of the system's peak performance

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

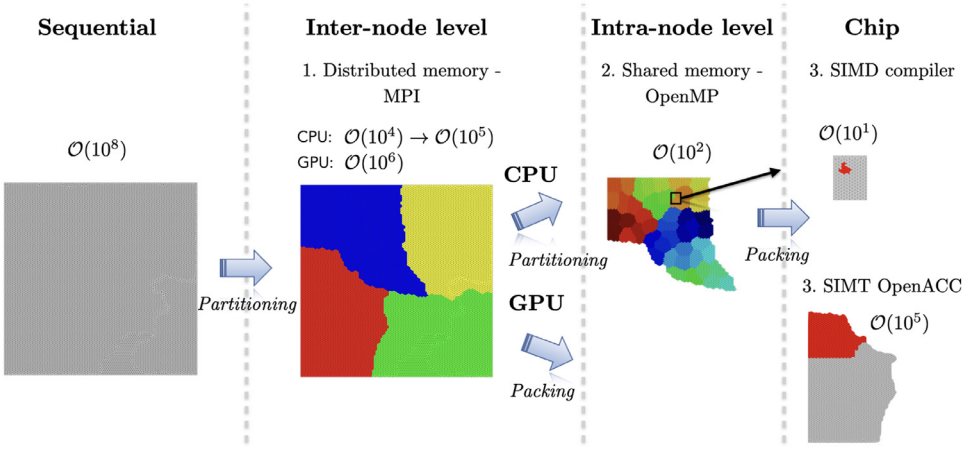

Fig. 4. Hierarchy of mesh partitioning and parallelization models to exploit modern supercomputers. Includes orders of magnitude of the subsets at each partition level..

achieved is generally less than 5%. The computational combustion workflows, mostly rely on sparse linear algebra, and in most of its phases the execution is memory bounded.

Another clear trend observed in supercomputers is the heterogeneity of the architectures implemented at the node level. Nowadays, eight of the first ten systems ranked by the Top500 list include Graphics Processing Units (GPUs) for general-purpose computations. GPUs further increase the level of concurrency since they are throughput-oriented devices exploiting data parallelism. Moreover, GPUs generally deliver higher FLOPs per watt ratios than CPUs, optimizing power consumption, which is a critical concern for the operation of supercomputers.

In the post-Moore era, heterogeneity of systems is expected to keep growing [56], and domain-specific devices (e.g., deep learning chips) shall populate supercomputer nodes. For example, modern NVIDIA GPUs contain tensor cores that are particularly efficient for AI inferences. Combustion simulation workflows will have to evolve to take advantage of special-purpose devices to speed up runtime calculations or generate new knowledge from simulation outcomes when extremely large datasets are generated [57].

3.1. Advanced software for High-Performance Computing

The solution of the problems already described for both DNS and LES requires specific algorithms to take advantage of the different levels of parallelism within the supercomputers. Eulerian methods are solved on a computational grid using domain decomposition methods [58]. The partitions are obtained from the underlying mesh, and the subdomains perform independent operations and exchange the values at the boundaries. This means that operations are performed on groups of mesh entities simultaneously, providing concurrency and thus, parallelism along the different calculation phases of the code. Exposing parallelism at all these levels requires combinations of various programming languages or APIs. Some of them are community standards like MPI [59], OpenMP [60] or OpenCL. In contrast, others are closely related to specific vendors such as CUDA [61] or OpenACC for NVIDIA.

To relieve programming and porting efforts (and let application domain experts to concentrate on models for physics), several abstract programming models have been introduced to hide parallelism complexities and provide porting across HPC platforms. This is generally achieved by adding intermediate software layers that interface with underlying parallelization models. Kokkos is a successful example of a performance portable programming model [62]. A significant speedup was demonstrated by using GPUs through Kokkos on Lagrangian-Eulerian simulations of liquid sprays [63]. Legion is another parallel programming system to implement portable HPC programs for distributed heterogeneous architectures [64]. It is based on a task-based programming model that allows efficient data movement across complex memory hierarchies using data-level parallelism. Similarly, the portable Open Concurrent Compute Abstraction (OCCA) framework, is a vendor-independent abstraction focused on node-level heterogeneous computing that has been successfully implemented on Computational Fluid Dynamics (CFD) codes such as NekRS [65]. The code Pele [66] is another succesful computational platform for Exascale-type simulations of combustion with different modules and libraries for various physics and algorithms. The

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

Portable, Extensible Toolkit for Scientific Computation (PETSc) [67] has also been used as a platform for large-scale subsurface reacting flow calculations [68]. Fully adopting these frameworks gives high flexibility to adapt to different architectures, but creates a strong dependency on its evolution and capacity to adapt to new architectures. For this reason, other strategies for code development are based on general algorithms that can be implemented in legacy codes to rapidly adapt to the changes in hardware. These aspects are discussed in the next subsections.

3.2. Inter-node level

Inter-node parallelism is based on the partitioning of the mesh into disjoint or overlapping sets of elements (depending on the numerical method considered), referred as subdomains or partitions. Computation is then carried out independently by individual processes in charge of the different subdomains, which run simultaneously on the different cores of the supercomputer. To provide coherency of the solution, synchronization of these local solutions requires data from neighboring subdomains at their interfaces. However, the different nodes of a supercomputer do not share a common memory to exchange these interface data. Parallelization thus relies on specific distributed memory techniques: data are transferred through the communication network using message passing library, like MPI [59].

An ideal partitioning should provide well-balanced distributed data, while minimizing the size of the interfaces, where communications take place. The partitioners can be divided into two main categories: topological and geometrical. On the one hand, topological partitioners like ParMETIS [69] are based on the mesh connectivity, and permit reaching balanced subdomains imposing the minimization of the interfaces as a constraint. However, their sequential version is usually the best option in terms of quality, which degrades with the number of processes. On the other hand, geometrical partitioners (like the ones based on space-filling curves (SFC) [70]) are exclusively based on the element coordinates. They are fully parallel and enable to better reach a desired load balance. However, there exists no explicit mechanism to control the sizes of the interfaces, that are indirectly optimized thought the locality of the SFC-projection. Figure 5 illustrates the results of the partitioning of a 11.8M element hybrid mesh into 256 subdomains, both using an SFC and METIS library. The spheres represent the different subdomains and are located at their centers of gravity, while the lines indicate a neighboring relation (in a finite element implementation, a neighbor is a subdomain sharing at least one node). We observe that the SFC provides an almost perfect theoretical load balance while METIS reaches only a value of 0.91. As explained earlier, the good load balance is achieved at the expense of a worse communication. The figures on the bottom compare the maximum number of neighbors for each partitioner. Using the SFC, we obtain up to 32 neighbors compared to a maximum of 20 for METIS. A similar conclusion can be drawn for the number of interface nodes, which is twice higher in the case of the SFC in average (not shown in the figure).

As already introduced in Section 2.3, the algebraic system *assembly* consists of a loop over faces, elements or nodes to assemble matrices in the case of implicit or semi-implicit schemes, and vectors in the general cases including explicit schemes. These operations do not involve communication and thus, the efficiency of this phase relies exclusively on the load balance, that is, how well the computation is distributed among the different processes. In fact, the assembly time is eventually driven by the slowest process arriving at the posterior subsequent synchronization point. Communications mainly take place in the solution of the resulting algebraic system by iterative solvers. In such solvers, the operations are in general *axpy* (combination of scalar multiplication and vector addition), dot products, which require a reduction communication and *SpMV* (sparse matrix-vector products). This last operation ensures continuity at the interfaces of the subdomains and involves the bulk of point-to-point MPI communications. We will comment on these aspects in Section 3.5.

In combustion and CFD simulations in general, typical number of elements per subdomain are given in Fig. 4, ranging from 10^4 to 10^5. In terms of number of subdomains and thus computational resources, we can make a distinction between small and large scale simulations. For small scale simulations, hundreds to thousands of cores can be easily reached, while for large scale simulations, the number of cores can go as much as the full machine. For instance, 1M cores were used for the simulation of a scramjet engine [5]. In [4], the simulation of the full engine DGEN-380 demonstrator was executed on 14,400 cores, corresponding to an average of 90k elements per MPI subdomain. In this work, intra-node parallelization makes use of the shared memory capabilities of MPI3, allowing for faster intra-node communication.

It should be noted that such distributed memory techniques also work in a shared memory context. That is, various MPI processes can coexist in a node, each process working in its own memory space in isolation from the others. However, in this last case, specific techniques can offer advantages over message passing based parallelism, as explained in the next subsection.

3.3. Intra-node level

Intra-node parallelism takes advantage of the fact that inside a computing node, memory is

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

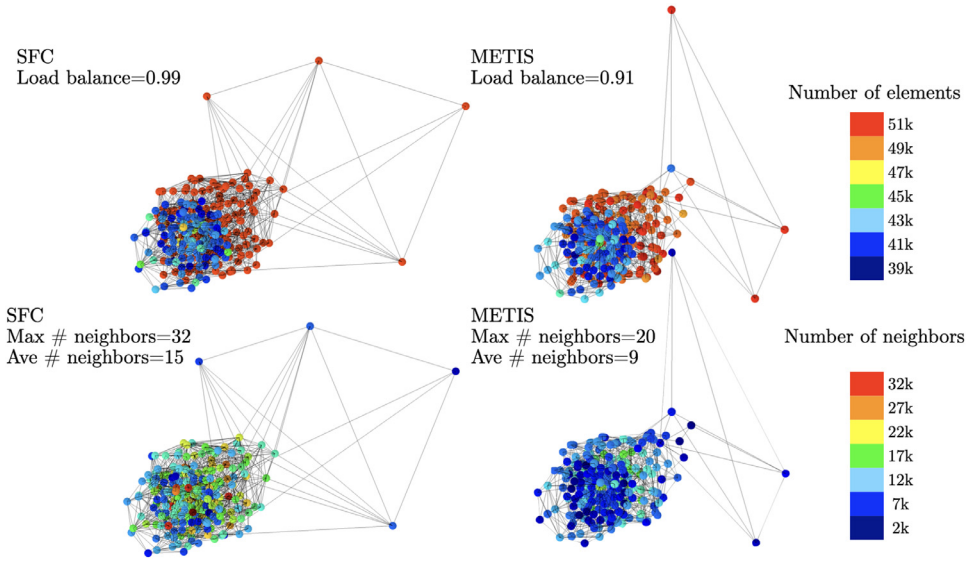

Fig. 5. Partitioning of the 11.8M element hybrid mesh into 256 subdomains, using an SFC (left) and METIS (right). The top figures compare the number of elements per subdomain as well as the theoretical load balance. The SFC provides an almost perfect load balance. The bottom figures compare the number of neighbors, indicating a worst communication obtained by the SFC.

shared between processes. Thus, techniques based on message passing are no longer necessary, although they can be still used. The favorite paradigm for shared memory parallelism is loop and task parallelisms based on the API OpenMP [60]. In the first case, the main loops of the code are divided into chunks, which are executed concurrently on the available cores. By contrast, task parallelism is a more general concept. For example, in the case of the assembly, the mesh can be divided into subdomains on which the assembly (task) will be executed concurrently on the different available cores.

OpenMP parallelism is relatively easy to implement as it consists of simple pragma directives to be placed before each parallelized loop. The main drawback of this technique is the requirement of explicitly annotate all the loops of the code, which makes it a poor approach in terms of modularity. As expressed by the Amdahl's law, the scalability remains limited by the part of the code not being exposed to parallelization directives (sequential part). It is easy to imagine that if only half of the code has been parallelized, the total time will eventually be dominated by the sequential part when increasing the number of cores. In fact, the maximum achievable speedup would be 2. Moreover, another non-negligible drawback is the difficulty of maintaining a living code, where loops can be coded daily.

The main difficulty of implementing this intra-node parallelism is treating the so-called race condition. This condition occurs when parallelized executions of the code (threads) update the same memory position. In the assembly phase, this happens during the gather operation, which assembles local contributions (element/face/node-wise calculations) into the global vectors or matrices. Several strategies have been proposed, as illustrated in Fig. 6:

1. The first straightforward possibility consists of adding to the code some ATOMIC pragmas to prevent the different threads from writing on the same memory position at the same time. Although simple to implement, the ATOMIC adds a non-negligible overhead due to sequentialization.
2. To avoid the addition of these ATOMIC pragmas, coloring techniques can be used [71], where elements are associated with colors in such a way that elements with the same color do not share a node. Safe parallel loops are therefore carried out for each color, consecutively. However, with this strategy, data locality can be compromised.
3. Loop parallelism can also be applied with explicit mesh partitioning instead of coloring. In this case, the mesh is partitioned and separated by layers of elements so that elements from different subdomains do not share a node. Safe parallel loops are therefore carried out for each color, consecutively. Then, loop parallelism can be applied over the subdomains avoiding ATOMIC pragmas. The main drawback is that separators should be treated apart from the main loop [72,73].

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

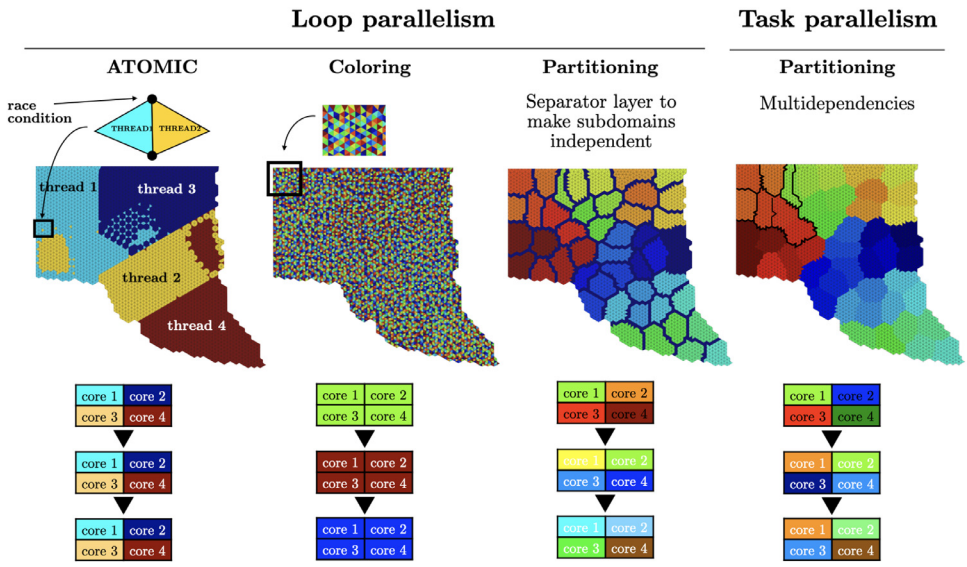

Fig. 6. Different loop and task parallelism techniques to deal with the race condition with OpenMP, at the intra-node level. On the bottom of the figure, it is shown how elements or subdomains are mapped to the different available cores (4 in this illustrative example) and assembled in parallel for the different techniques.

4. Finally, one can take advantage of the multidependencies capabilities of OpenMP [74]. Here, the dependency graph is explicitly given in the OpenMP pragmas to indicate the dependence between the subdomains. Subdomains that do not share nodes are then executed in parallel as OpenMP tasks.

Inter-node and intra-node parallelisms can be combined to enhance the performance of the code. This parallelism is referred to as hybrid. For example, to achieve distributed parallelism between computing nodes, one MPI process per node can be selected; then, OpenMP is in charge of the parallelization of the MPI subdomains inside each node. In general, the right combination MPI/OpenMP and the best configuration depends on the specific implementation of each code. Hybrid strategies have been used to ensure high parallel performance in combustion simulations [75,76].

OpenMP can also be used as a mechanism to reduce load imbalance at node level, as specific scheduling techniques enable automatic load balance of loop/tasks. Also, runtime libraries executed on the top of OpenMP can help to further enhance the load balance. For example, in [74], the DLB library enables adding idle CPU-cores to an MPI process by spawning additional OpenMP threads at node level, thus reducing the load imbalance. A typical situation is when a subdomain has already finished its assembly, and the bounded cores can be lent to other MPI tasks. Figure 4 illustrates a mesh partitioning carried out on one MPI subdomain. In general, the granularity of the second level partition (formed of loop chunks or tasks) should be sufficiently large to hide parallelization overheads while being sufficiently small to enable load balance by dynamic distribution. Typical chunk or task sizes are of the order of 10^2. An alternative to the manual implementation of pragma directives is discussed in [77], where a library that provides abstractions to automatically generate shared or distributed parallel parallel code is presented.

3.4. Chip level

Parallelism within a single processor includes various mechanisms such as instruction-level parallelism, pipelining, or vectorization. All of them are optimizations that compilers can perform. However, programmers can maximize the outcomes by creating compiler- and architecture- friendly codes [78]. A clear example is vectorization, which performs operations on chunks of data in parallel using the single instruction multiple data model. The programmer can help the compiler to generate vector instructions. For instance, reorganizing loops to expose parallelism and improve locality, or implementing gather-scatter operations to ensure regular access to data. The main purpose of vectorization is to increase the arithmetic intensity. The arithmetic intensity is defined as the number of operations FLOPs per total bytes transferred to perform the operation. The algorithmic intensity will be formally defined in next section.

On modern CPUs, vector or SIMD extensions are becoming more and more relevant. Besides the AVX-512 SIMD extension by Intel, there are other

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

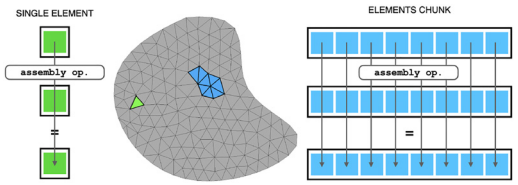

Fig. 7. Illustration of vectorized assembly implemented in Alya code. In green, a classical assembly where elements are assembled one by one. In blue, chunks of elements assembled at the same time.

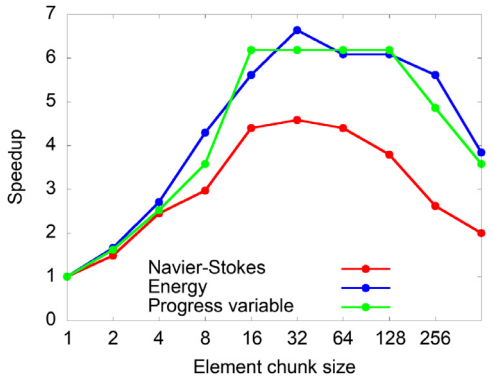

Fig. 8. Speedup obtained as a function of the element chunk size, for the different equations solved using the PRECCINSTA burner [81].

players like the im15 CPUs implementing the Arm SVE extension, installed in the Fugaku supercomputer (ranked second in the Top500 list [54]) and the NEC SX-Aurora vector engine, which is a discrete accelerator leveraging vector CPUs able to operate with registers of up to 256 double-precision elements. As stated before, the efficient use of vector units within CPUs relies on auto-vectorization by the compiler. However, in order to expose parallelism at this lowest level, it is common for CFD code writers to develop specific strategies to exploit vectorization efficiently.

For instance, in the FV code OpenFoam, vectorization was improved by helping the compiler with the extensions *__restrict* and *__builtin_assume_aligned*, which gave further hints to the compiler about how to access and align the data. Introducing these two extensions to the source code reduced the time for computing chemical reaction rates by up to 50% and the total simulation time by 25% in the cases investigated [79].

In the FE code Alya [80], for an element-based assembly, chunks of elements are created to ensure regular access to data and thus leverage the performance boost delivered by vectorization. This strategy is illustrated in Fig. 7. The chunk size is a parameter that can be determined during compilation time to maximize the information provided to the compiler. Detailed analysis of the computational impact of this methodology on the PRECCINSTA burner using tabulated chemistry shows a reduction of x4.67 in the overall simulation time using the MareNostrum IV supercomputer [81]. Figure 8 shows the speedup obtained in this case when solving the Navier-Stokes equations along with transport equations for the energy and the chemical progress variable. We observe a speedup factor of x6 for the solution of scalar transport, using a chunk size of 32.

The same vectorization technique applied to a spectral finite element assembly is presented in [82], showing a speedup of x2 over the reference version. In [83], the authors discuss a vectorization strategy together with data reordering for thermochemical source term evaluation, achieving speedups between x3 and x10 times with respect to their reference solutions.

Another common way of enhancing arithmetic intensity is to improve the spatial and temporal locality of the data. When performing loops, new data is accessed continuously. The purpose of ensuring data locality is that, as the loop progresses through the mesh entities, the access occurs on data close to each other from the main memory. This can be achieved through renumbering strategies applied to the different mesh entities (element, faces, nodes). The well-known Cuthill-McKee technique [84] was presented early in 1969 with the objective of reducing the bandwidth of an sparse matrix and it is still used today. In [85], a reordering strategy for an edge-based CFD solver is developed allowing vectorization, while reducing jumps in memory access.

In addition, the use of GPU chips is an efficient way to speedup high arithmetic-intensity operations, while focusing on throughput via parallelism. Indeed, GPUs leverage massive thread-level parallelism based on the single instruction multiple threads (SIMT) model to achieve high computation throughput. Currently, directives-based programming models, such as OpenACC or OpenMP, can express parallelism within the GPU. With this approach, the compiler generates code, which eventually is executed by the threads. There are also lower-level options such as CUDA or OpenCL. In those cases, the programmer explicitly generates the code that is executed on each thread and controls all the parallelization-related aspects. Studies of trade-offs between those options can be found in [86,87].

Adaptations performed on loops and memory structures to optimize the SIMD model on the CPU and the SIMT model on GPU are quite consistent. In both cases, the goal is to expose additional concurrence in the calculations and regularity (coalescence and alignment) on the data accesses. For instance, in the code Alya, the same

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

implementation, based on packing elements into chunks, is used for both devices, the optimal chunk sizes being on the $\mathcal{O}(10)$ for the CPU and $\mathcal{O}(10^5)$ for the GPU [80]. Other authors have developed automatic porting strategies to define the chunk size [88].

In the case of reacting flow simulations, several authors have taken advantage of the power density of GPUs. As combustion simulations often present high numerical stiffness due to the chemical integration, this part can be conducted directly in the GPU, while the assembly of the transport terms is done in the CPU [89–92]. High-order methods are also well suited for GPUs due to its higher arithmetic intensity compared to linear discretizations, which make these strategies highly efficient for high-fidelity simulations [93]. All these aspects will be discussed in more detail in Section 4.

3.5. Computational performance

Combustion involves different physics and interactions between physical laws, exhibiting a large variety of computational patterns. Algorithms for combustion simulations need to include not only the corresponding physics, but also the correct management of memory and instructions to ensure the efficient use of resources. In this subsection, the concept of performance is introduced with focus on parallel efficiency.

The evaluation of performance in a given simulation is determined by the algorithmic and computational performances respectively. On the one hand, computational efficiency accounts for the performance of the algorithm's execution. This can be envisaged with a time-to-solution criterion or an energy-to-solution criterion. On the other hand, the algorithmic performance measures the efficiency in terms of convergence speed and accuracy. Take, for example, an algorithm for solving a proposed problem with given physical and numerical components: the physical modeling (e.g., compressible Navier-Stokes, specific combustion model, boundary conditions, etc.); the discretization method (FV, FE, FD, Lattice-Boltzmann, etc.); the iterative solver; the coupling strategy for the multiphysics problem, etc.

The selection of the optimal methodology for solving a given problem, comes not only from the selection of the physical models, but also from the numerical methods used to solve these equations and more importantly, from the series of operations associated to the use of the adopted methodologies. Algorithmic and computational performances are eventually inter-linked, and both need to be analyzed together to develop an efficient HPC simulation framework. The algorithmic and parallelization setups depend on various factors such as the desired outcome, level of accuracy, hardware architecture, and all the tools involved during the execution (compilers, runtime libraries, system, etc.). The selected strategy is tested following certain recommendations that are used to evaluate the performance of the algorithms and the codes. There are different metrics that can help to assess the parallel computation efficiency. A simple model [94] involving the main metrics defining the efficiency of a parallel simulation is shown in Fig. 9.

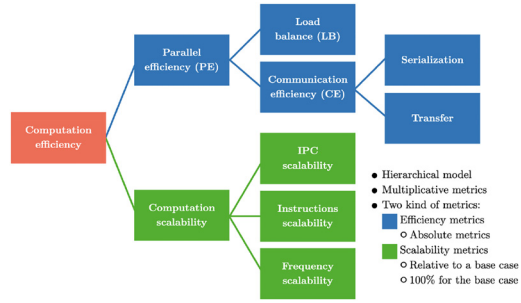

Fig. 9. Performance model.

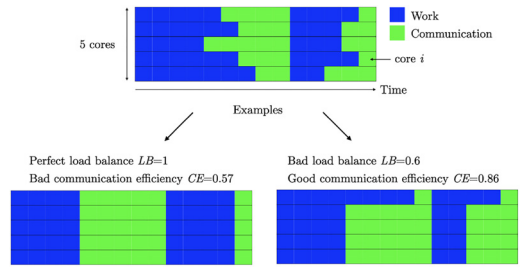

Fig. 10. A schematic trace of an execution. Examples of bad and good LB and CE.

The blue metrics in Fig. 9 give absolute values, while the green ones are relative to a reference case (usually based on the smallest core count). The model is multiplicative, for example, the parallel efficiency PE is given by $PE = LB \times CE$, where LB and CE are the load balance and communication efficiency, respectively. Subsequently after the metric of interest is computed, appropriate runtime checks are applied to the problematic parts of the code to obtain reliable diagnosis. Several libraries provide some of these metrics and different tools enable to obtain more intricate details [95,96]. Tracing tools collect and store information on hardware counters, MPI, and OpenMP calls that occur during the execution together with the timing information. These traces can then be visualized and analyzed with trace visualization tools [97,98]. Figure 10 shows a typical trace of an execution. It represents the time as function of the number of processes involved in the simulation. The colors indicate the activities of the processor, where blue is used for working, and green for communication.

Load balance is a measure of how the computation is distributed among the different pro-

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

Table 1
Definitions of important performance metrics. i refers to core number out of P cores. Colors correspond to the colors of Fig. 10.

Symbol	Value	Definition	Color reference
t_c^i	Measured	Communication time of i	■
t_w^i	Measured	Working time of i	■
$\overline{t_w}$	$\overline{t_w} = \sum_i t_w^i / P$	Average working time	ave ■
t_e	$t_e = \max_i (t_c^i + t_w^i)$	Elapsed time	max (■+■)
LB	$LB = \overline{t_w} / \max_i t_w^i$	Load balance	ave ■/ max ■
CE	$CE = \max_i t_w^i / t_e$	Communication efficiency	max ■/ max (■+■)
PE	$PE = LB \times CE = \overline{t_w} / t_e$	Parallel efficiency	ave ■/ max (■+■)

cesses. It is equal to the average working time of the processes divided by the maximum time. Thus, the speed of simulation is determined by the slowest process. Communication efficiency measures useful computation with respect to communication. Mathematically, it is defined as the maximum computation time over the MPI tasks divided by the elapsed time. Computation scalability refers either to Instructions Per Cycle (IPC), instruction scaling or frequency scalabilities in terms of threads/processes. IPC is related to the rate of computation, for example, when increasing the number of subdomains, the cache hit rate may increase due to the smaller subdomains sizes, resulting in an increase in IPC. Instruction scaling is related to the number of instructions. Finally, frequency scaling refers to the processor frequency variation during useful computation. Usually this frequency decreases with the number of cores used on the node.

We will now concentrate mainly on parallel efficiency, as it consists of the most penalizing factor in large scale simulations. Formally, the are efficiencies shown in Table 1, where i refers to the core number out of total P cores.

From the formulae, it is understood that a perfect parallel efficiency means that the average work time is equal to the elapsed time. This parallel efficiency is, in turn, influenced by LB and CE in a multiplicative way. Figure 10 shows two examples of how these two factors affect the global efficiency. Left side of Fig. 10 shows a perfectly load balance scenario. However, the relative weight of communications is high, leading to a parallel efficiency of $PE = 0.57$. This means that almost half of the time is lost at the expense of useful computation and thus, half of the computing resources are wasted. By contrast, the scenario on the right side shows that the load balance is bad, but the communications are at a reasonable level. Here, the parallel efficiency is $PE = 0.52$ meaning that also in this case, half of the resources are wasted, but this time because of a high load imbalance.

Such a detailed analysis is rarely carried out in practice. In general, to asses the performance of the code, the code users rely on the relative speedup and relative efficiency by performing a *strong scalability* analysis. A strong scalability test consists in executing the code on an increasing number of cores and measuring the acceleration based on simple timings. Let P^b be the minimum number of CPU-cores used during the campaign, and t_e^b its associated execution time. Then, if the execution time on P cores is t_e, the speedup s is evaluated by

$$s = \frac{t_e^b}{t_e}, \tag{2}$$

while the perfect speedup is P/P^b. From these figures, the efficiency EF_{rel} is thus defined by

$$EF_{rel} = \frac{t_e^b}{t_e} \frac{P}{P^b}. \tag{3}$$

Note that this efficiency is not the parallel efficiency PE defined earlier (as often named in the literature), but just a relative efficiency evaluation normalized by the base run. Usually, the minimum affordable number of cores used is already large for memory and time limitations. Therefore, it should be considered an indicative metric and not an absolute one. Figure 11 shows an example of speedup obtained with Alya code. The normalization time was computed with $P^b = 16384$ cores. As the number of cores increases, the load per core decreases while the relative weight of the communications increases, eliciting a reduction of the speedup and the parallel efficiency.

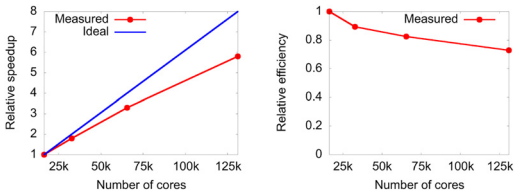

Fig. 11. Relative speedup and efficiency obtained with Alya code for an Eulerian combustion simulation. Base number of cores is 16384 and maximum number of cores is 131072.

While strong scalability indicates how faster the simulation goes when increasing the number of cores, *weak scalability* measures the change in execution time when the number of CPU-cores and the number of degrees-of-freedom are multiplied

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

by the same factor. The result of a weak scalability analysis enables one to answer the following question: what is the penalty in terms of execution time if one refines the mesh but increases the number of cores with the same proportion? Regarding the assembly phase, weak scalability is not a real issue, however, the case is different for linear solvers. The condition number of the system matrices increases with the mesh size (and thus, in general, the iterations of the iterative solver), so robust solvers must be used to remove this strong dependence on the mesh size. In addition, this convergence should be independent of the number of cores. Eventually, the time-to-solution up to a given accuracy should only depend on the load per CPU.

Domain decomposition solvers [99] are iterative solvers that leverage this double dependence. While some of them perform well in terms of iterations to reach convergence, the independence of time-to-solution strongly relies on the problem to be solved, the specific solver, and its implementation [100]. It is important to stress that weak scalability should always be measured at reaching the same solution. For example, in the case of a transient simulation, one should measure the execution time to reach the same physical time of the problem. Therefore, explicit solvers are by nature not weak scalable as the Courant-Friedrichs-Lewy (CFL) condition makes the time step reduce as the mesh size increases (if convection dominates, the time step is proportional to the mesh size).

As mentioned previously, the computation efficiency measures how well the computational load of an application scales with the number of threads or processes, but does not give information on the absolute performance at the chip level. Complementary information can be provided by the so-called roofline performance model. A given computational kernel is characterized by its algorithmic intensity. If we define F as the number of operations (in FLOPs) performed by the kernel, and B the number of bytes transferred to carry out the operations, then the algorithmic intensity AI in FLOPs/byte is given by

$$AI = F/B. \tag{4}$$

A high AI indicates that data are being well reutilized to perform computations, while a low AI indicates that memory traffic is dominating over these computations. AI is thus dependent on the characteristics of the algorithm but also on how well data locality has been addressed. However, the attainable AI is also limited by the characteristics of the platform on which the kernel is executed, that is the peak performance and the memory bandwidth. To assess the quality of algorithmic intensity of a specific kernel, one can rely on the so-called roofline performance model [101]. The roofline model indicates how far the AI of the kernel is from the attainable performance of the platform where it is executed.

All these metrics are useful to obtain a diagnosis of the code performance based on the definition of specific metrics. Then, the performance of the code can be enhanced, by applying specific optimization techniques, which depend on the selected kernel. We will now present some of the techniques applied to the main three kernels discussed in Section 2.3.

3.6. Enhancing performance

As discussed in the previous sections, the parallel performance of a code depends on many factors at all the levels of the supercomputer hierarchy. Some of them can drastically affect the performance of the code, like the load balancing, which is a general problem in fluid mechanics and specially relevant in combustion. In reacting flow simulations, the major contributor to the load imbalance comes from the description of chemistry, mainly during the chemical integration, but also from the evaluation of transport properties [10]. The load imbalance is a consequence of the nature of the problem, since turbulent flames can be considered as thin interfaces with high rates of heat and mass exchange, and introduce strong variations in computational load in both space and time.

The cost of the assembly of these PDEs is exclusively driven by the load balance as it does not involve communication. A typical criterion to achieve is to obtain the same number of elements in each subdomain by relying on the partitioner. At the inter-node level, a measured load imbalance can be corrected by redistributing the mesh between the different MPI processes; at the intra-node level, when using OpenMP, runtime techniques like the ones described in Section 3.3 can be used.

Another element to be evaluated is the iterative solvers, where communication among processes dominate the cost of calculation, so a common strategy to increase communication efficiency is to overlap them with floating point operations. In the case of the SpMV, matrix-vector product operations are computed on the degrees of freedom at the subdomain interfaces, exchanging the results between the subdomains using non-locking communications and simultaneously carrying out the matrix-vector product on internal degrees of freedom [102].

The solution of the chemical integration can be considered as a local phenomenon and thus, it can introduce high load imbalance. In [103], an MPI-based redistribution is proposed to balance the computation of the chemical kinetics, resulting in a gain of factor x3 on 4096 cores. In [104], the load balance of chemistry is achieved at the intra-node level, using OpenMP and the DLB library to exploit the resources of the MPI subdomains with lower chemical load achieving a factor of x5 in detailed chemistry simulations. These strategies will be discussed more extensively in Section 5.

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

Particle transport suffers from similar load balancing issues as chemistry due to the localised nature of Lagrangian sprays (evident during injection). Load imbalances in the particles can seriously reduce the performance of the simulation. At the inter-node level, migration is necessary to transfer particles from one subdomain to another whenever the particles reach the subdomain interface. To leverage this load imbalance, particle transport can be easily parallelized with OpenMP. In [105], hybrid parallelism is used to enhance the performance of particle transport. The DLB library is used on the top of OpenMP in order to continuously exploit all the resources available at the node level, thus increasing the load balance and the parallel efficiency. In this study, authors compare a classical single-code coupling to a multi-code approach, where one instance is dedicated to the flow equations, while the other is on the particle transport introducing certain level of asynchronism. An original approach is proposed by [106,107], based on the shared memory capabilities of the new version of MPI (MPI3) to redistribute particles at a reasonable cost.

As pointed out, previous techniques are essential to achieve high efficiency and proper use of computational resources when running combustion simulations in HPC architectures. In the following sections, the state-of-the-art of these techniques are presented with focus on the use of high-order methods, reacting flow simulations with detailed chemistry and multiphase flows.

4. High-order methods in combustion simulations

The use of high-order methods for solving chemically reactive flows has gained popularity in the last years due to the limitations inherently present in low-order methods in terms of dissipation and computational cost [93]. While high-order methods have been widely extended in the context of FD and spectral methods for DNS [2,7,14,35,57,108], their application to complex geometries with LES has been more scarce. The main reason for the scarcity can be attributed to the complexity in extension of high-order operators to FV and FE methods, which are the most used approaches for LES on unstructured meshes. Recent developments in HPC hardware architectures, i.e. moving from large-scale shared and distributed memory clusters of CPUs to hybrid CPU/GPU architectures with extreme levels of parallelism, has opened new possibilities for the development of numerical methods that can exploit more effectively these architectures. This trend motivates an increased interest in developing high-order schemes that can be used to conduct scale-resolving simulations on unstructured meshes and complex geometries [93,109].

On the algorithmic side, high order methods have shown a higher rate of mesh convergence than their low-order counterparts [93]. On the computational side, it has been demonstrated that such methods offer a better float per byte ratio due to higher arithmetic intensity produced by an increased data reuse, and better data locality given by regular memory accesses. However, what is eventually relevant is the improved rates of time-to-solution and energy-to-solution of high-order methods compared to classical second and fourth order central schemes. This increased performance comes from both algorithmic and computational benefits, bringing the interest in these methods for the efficient exploitation of future HPC architectures with GPUs. Notwithstanding, due to the complex memory hierarchy, involving several levels of cache, together with the large variety of available methods, the expected benefits need to be investigated carefully. For instance, despite the benefits from the repeated application of high-order first derivative operators to compute the diffusion term, which can be efficient in terms of floating point operations, the performance can be shadowed by the communication between processors with a penalty in computational efficiency for multi-core architectures [110].

Classical methods for LES include second-order accurate spatial discretizations based on central-schemes in the case of FV [111,112] and stabilized linear elements for FE [113,114] methods. On the one hand, the FV method has been widely extended to solve reacting flow simulations for both gaseous and liquid fuels [79,111,112,115–117] and despite that higher-order approaches have also been derived [118], their use has been more limited. On the other hand, considerable effort has been made in deriving high-order numerical schemes for the FE method at reduced computational cost. A third-order Taylor-Galerkin scheme was originally proposed by Donea et. al. [119] and then, extended to a third and fourth order with wider stability limits by Selmin and Quartapelle [120], also referred as two-step Taylor-Galerkin schemes (TTG). Notwithstanding, due to its dissipation levels at intermediate and high frequencies, an improved TTG (TTGC), was proposed to minimize the dissipation at high frequencies, while maintaining third or fourth order at lower computational cost [121]. This algorithm has been successfully applied to different flame configurations, from sprays to gaseous flames [4,122].

An extension from linear FE to a higher-order discretization with quadratic and cubic elements has been used to address classical fluid mechanics problems [123–125]. In general, it is found that high-order methods reduce the degrees of freedom of a given problem, and the time-to-solution, but it creates spurious oscillations near shocks and steep gradients. Discontinuous finite element methods have shown arbitrary high-order of accuracy on

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

unstructured or curved grids, showing good potential for LES [126]. The compact stencils are very well suited for the use in GPUs, and GPU-based spectral elements methods have good potential for turbulent reacting flows [127].

A new generation of high-order techniques, also referred to as Discontinuous Galerkin (DG) methods, have also shown high potential for high-fidelity simulations [93,109]. In particular, the flux reconstruction method (FR), first introduced by Huynh [128], has been applied to high-order LES and DNS simulations [129]. However, despite the important progress made for general fluid mechanics applications, their use in multiphase reacting flow simulations has been rather low mainly due to difficulties for this method to ensure conservation of properties across the interfaces [130]. Lv and Ihme [131] developed a DG framework for subsonic and supersonic combustion with detailed and stiff chemistry using a hybrid-flux formulation combining a conservative Riemann-solver with an extended double-flux scheme. It was shown that the lower dissipation of the higher-order polynomial approximation, captures short-wavelength perturbations providing better descriptions of the flame front and flame structure. This work was recently extended by Bando et al. [130] to include a fully conservative formulation. All these methods are still at the exploratory level, but show high potential at increasing the accuracy and robustness of high fidelity simulations, and are well suited to be used with accelerators due to the high arithmetic intensity.

Notwithstanding, developing high-order numerical schemes for reacting flow simulations brings additional challenges beyond the use of high-order operators. One of the main problems of solving the transport equations for reacting systems deals with the integration of convection, diffusion and chemical source terms in the case of stiff chemistry. While explicit methods are very efficient and highly scalable, they are limited by small time step sizes to ensure stability [132,133]. In such situations, implicit methods are more robust, but also more costly and difficult to parallelize. In particular, considering the conservation equation for the chemical species Eq. (1), it can be seen that the different terms of the equation are associated to different time scales, which makes the system stiff and complicates the integration procedure.

High-order operator splitting methods were derived to mitigate this problem by taking advantage of the different scales of the reacting flow [134]. Operator splitting has been combined with *In Situ* Adaptive Tabulation (ISAT) and speed-ups of several orders of magnitude have been reported [135]. In [136], a new splitting scheme that used staggered time steps was developed with a single evaluation of chemistry reducing the overall cost of chemical integration in multistep calculations. Moreover, other works confirmed second-order splitting schemes with a single chemistry evaluation without loss of accuracy [137]. Such schemes have also been applied in combination with dynamic adaptive chemistry methods [132], which will be introduced in Section 5.2. As previous splitting schemes are limited to second order in time, alternative methods have been also proposed in order to further reduce the splitting errors. A multirate time stepping strategy based on spectral deferred corrections (SDC) was developed based on the use of different time scales during the time marching [110,138]. For a given time step, the method constructs high-order solutions by approximating a series of correction equations at collocation of nodes using low-order sub-stepping methods. Based on the SDC method, the multi-implicit spectral deferred correction (MISDC) [139] was developed and applied to combustion simulations [140,141]. In MISDC, all the processes are iteratively coupled implicitly, so the splitting error can be reduced substantially. Alternatively, convection can be explicitly integrated while diffusion and reaction are implicitly solved giving rise to the SISDC (semi-implicit SDC) methods [139]. An extension of these methods based on the multirate SDC (MRSDC) [110] was used to solve a dimethyl ether jet flame. MRSDC allowed to advance the transport with larger time steps than the chemistry and hence, reduce the coupling frequency and the resulting computational cost. Alternatively to operator splitting, Implicit-Explicit or IMEX algorithms have been proposed to deal with the multiple scales of the different processes [142,143]. In this algorithm, the non-stiff regions of the domain are integrated according to an explicit scheme, while the stiff regions are integrated with an implicit method.

In conclusion, the separation of the chemistry permits the evaluation of the different processes independently. The transport can be evaluated by advective and diffusive operators, and chemistry can be computed separately. This brings the possibility of exploiting the asynchronous by using GPUs to accelerate the chemical integration. This aspect is discussed in Section 5.

5. Managing detailed chemistry in combustion simulations

One of the fundamental problems in combustion simulations is how to account for chemical effects, while controlling the error and the computational cost. Combustion chemistry is responsible for complex phenomena in the flow like ignition, extinction or pollutant formation and therefore, the selection of the reaction mechanism is an important element of the simulation [10]. Chemical kinetics is a highly non-linear process and it is usually described by elementary reactions of Arrhenius form. Solving transport equations for the all species involved in large reaction mechanisms can

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

be prohibitive due to the stiffness associated to the chemical source and the coupling with turbulence. This cost forces a trade-off between computational cost and accuracy when solving high fidelity reacting flow simulations. The cost comes from two main sources: the reactive scalar transport and the chemical integration. On the one hand, the scalar transport involves the assembly and solver, already described in Section 2.3. Explicit and implicit methods are developed to integrate these equations, and both methods have been widely used in the context of LES and DNS. The exact number of equations and terms can change substantially from case to case due to the different mechanisms, physics (liquid, gas, NOx, soot,...) and variety of turbulent combustion models. On the other side, the chemical integration requires the solution of a system of ODEs that can be stiff in some situations. The next sub-sections focus on these three aspects.

5.1. Integration of chemical kinetics in high-fidelity simulations.

Before describing the implications of models for turbulent/chemistry interaction, let us first introduce the possibilities of including chemistry in turbulent reacting flow simulations with LES.

A widespread approach to simulate the reacting flow at reasonable CPU costs relies on the use of reduced chemistry [4]. These methods accurately predict flame propagation, heat release rates and flame acoustics. Reduced schemes based on quasi-steady sate analysis (QSS) [144], analytically reduced chemistry (ARC) [13], or single-step chemistry [145] are often applied to predict unsteady effects and flame dynamics in practical applications. Reduced schemes can also be extended to describe gas phase compositions and pollutant formation at reasonable costs, ensuring that the chemical source term is generally non-stiff. A low-dimensionality of the chemistry can also be retained by the use of virtual chemistry, where only few chemical steps are required to obtain the gas phase compositions [146].

More detailed chemistry models include those based on dynamic adaptive chemistry (DAC), which emerge as an efficient technique to reduce the computational cost in reacting flow calculations combined with on-the-fly chemistry reduction and operator splitting methods [132,147–149]. Figure 12 shows a methane triple flame computed with DAC using the GRI3.0 mechanism (53 species). The triple flame is characterized by the formation of two partially premixed flames on the leading edge and a trailing diffusion flame with the hot gases. It is observed how the relevant number of species to be considered for chemical integration changes depending on the spatial location. The speed-ups attained with this method can be highly remarkable ranging from x3, in most parts of the flame, to x6 in the regions of high temperature. This is obtained using an implicit solver for chemical integration CVODE [150] with analytical jacobian [151].

The separation of transport and chemical scales with operator splitting schemes facilitates the development of efficient and scaling algorithms using HPC-based techniques. However, it is convenient to devise methods that provide accuracy, while reducing the number of chemistry evaluations, since this is the most computationally demanding step [136]. Locally reduced chemistry can be used to obtain speedups during chemical integration, as only the most relevant reactions are integrated to compute the chemical source terms. In general, even with the removal of a large number of chemical reactions, global burning characteristics can still be predicted accurately as the relative importance of these reactions is sensitive to the local thermo-chemical conditions [15]. This idea is pursued by on-the-fly chemistry reduction strategies, which are commonly based on chemical pathways, namely, Directed Relation Graph (DRG) [152], DRG with error propagation (DRGEP) [153] or Path Flux Analysis (PFA) [154]. The reduction process can be computationally expensive as it usually requires the evaluation of matrices that scales quadratically with the size of the reaction mechanism. The reduction is also sensitive to local changes in the thermochemical states and hence must be obtained in runtime at regular frequencies. The use of chemistry agglomeration, initially proposed by Jangi et al. [155] with spatial and temporal correlations (CO-DAC) [156] can be used to reduce the cost of the chemistry reduction. On the-fly-chemistry reduction is a data-intensive process and can strongly benefit from the use of heterogeneous architectures and modern supercomputers. Moreover, the matrix-type calculations of the reduction algorithms are well suited to be off loaded to GPU's with significant speedups.

The computational strategy needs to manage the available resources among the two main subprocesses i.e. the chemical integration substep, which is generally expensive and localised and the transport substep that is faster when using explicit methods. To deal with such computational imbalances in parallel calculations, load balancing techniques are used to redistribute the load among the different processors and increase the efficiency of the calculations. A factor of x5 in speedup was achieved in the integration of stiff chemistry by the use of dynamic load balancing [104], while other authors have reported factors around x10 [157,158]. Additional speedups can be obtained by combining HPC strategies based on load balancing with hybrid MPI/OpenMP with vectorization [104]. While DAC approaches based on *in situ* adaptive tabulation (ISAT) [159] reduce the frequency of both on-the-fly reductions and chemistry integration, are known to predict accu-

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

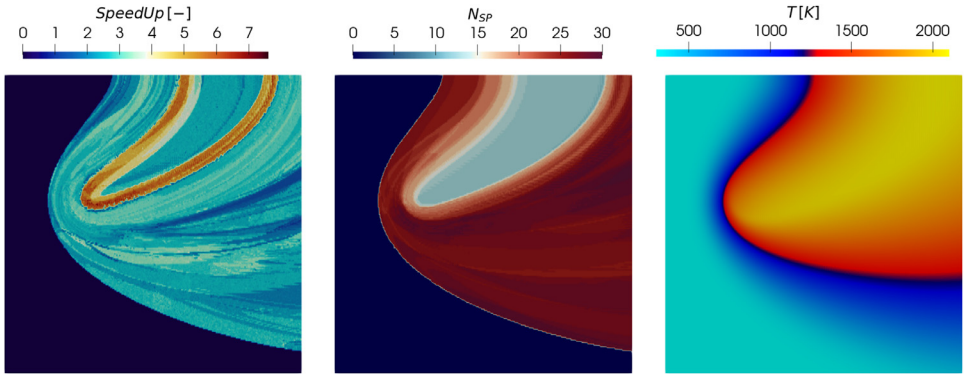

Fig. 12. Triple flame computed with DAC method starting from GRI3.0 chemical mechanism. From left to right: speed-up compared to the integration with the original mechanism, number of species retained when applying DAC method and field of temperature.

rately the flame structure, but are limited to data storage and retrieval strategies.

Another widespread methodology to include detailed chemistry in high-fidelity simulations is the use of tabulated chemistry, for which a family of flamelet-based tabulated models has been derived [160–162]. In flamelet methods, the chemical time scales are assumed to be faster than those of the turbulent flow, so the flame structure is retained and not affected by turbulence. For tabulated flamelet methods, the thermochemical states of the flame can be obtained in a pre-processing step and stored in a lookup table. The structure of the flame is recovered by the use of controlling variables that represent the multidimensional manifold space.

The reduction in computational cost is substantial as the transport of all the reacting species is reduced to a pre-defined set of variables and the chemical source terms are tabulated, avoiding the integration of the chemical source terms during runtime. Therefore, the computational load reduces to the transport of the controlling variables and the data retrieval from the database, which is mainly required to recover transport properties, source terms and certain species of interest [163].

Tabulated chemistry methods require the use of multiple controlling variables when dealing with unsteady non-premixed flames, and the tabulation of a set of variables of practical interest. Therefore, these methods are usually limited by the size of the database and strategies for storage and retrieval. In general, the transport of additional variables does not introduce relevant overcosts in massively parallel combustion codes, as the scalar transport is usually computed in vector form. Besides, the computation of these additional PDEs directly benefit from the existing MPI/openMP implementations and the load balancing strategies. However, extending the dimensions of the flamelet database to arbitrary sizes and dimensions is restricted by

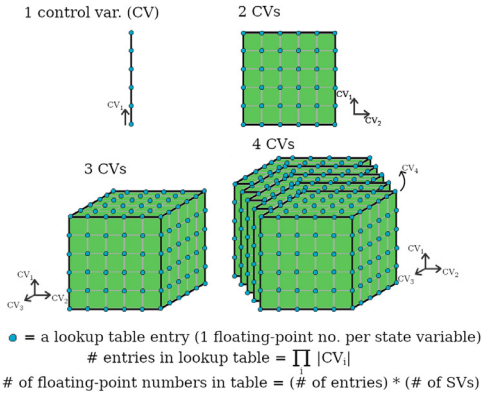

Fig. 13. The discretization of the flamelet lookup table can be visualized on the p-dimensional unit hypercube, where p is the number of controlling variables.

the memory of the computing nodes. A representation of a flamelet database is shown in Fig. 13. As more physical phenomena are introduced into the multidimensional flamelet manifold, the memory requirements increase according to N^p, where N is the number of tabulated values per dimension and p is the dimension of the manifold. Reducing the memory footprint and increasing the efficiency of these approaches is required to extend these methods to more general conditions [164–167]. A novel on-the-fly flamelet generation strategy was proposed by Kundu et al. [168] to combine the advantages of an online flamelet solver along with the computational efficiency of tabulated methods. The approach is based on the solution of the unsteady flamelet equations during runtime, so history effects and unsteady chemical kinetic effects can be included. Some recent strategies to reduce the memory footprint are based on the use of machine learning (ML) models.

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

Ranade et al. [169] proposed the use of an adaptive training algorithm that relies on multi-layer perception (MLP) neural networks for regression and self-organizing maps (SOMs) for clustering data to tabulate using different networks. The resulting strategy shows improvements in both the multi-dimensionality of the tables as well as the computational efficiency of the algorithm. Ding et al. [170] developed a ML-based algorithm to accelerate the thermochemistry computations and reduce the prediction error, especially for states yielding small composition changes based on artificial neural networks (ANN). ANN has also been used to completely replace the lookup tables and increase the computational speed of tabulated flamelets and related approaches [171–174].

5.2. Advanced methodologies for turbulent combustion.

Describing turbulent combustion involves the solution of the set of governing equations described in Section 2 and numerical methods given in Section 4. While for DNS no model is required, a variety of models for premixed and non-premixed combustion were derived for LES. The reader is referred to some review papers for further details on the wide spectrum of methods and approaches for turbulent combustion modelling [9,11,115].

When solving LES of turbulent reacting flows, models for turbulence-chemistry interactions are required in order to predict the correct evolution of the chemical species and burning rates [20]. The different possibilities to account for these effects led to a wide variety of specialised methods, which are now discussed in brief. Flame-thickening approaches have been used to predict complex flow conditions from lab-scale burners to flames in realistic engine environments in premixed [175] and non-premixed conditions [176]. Other methods describing turbulent chemistry interactions include the Eddy Dissipation Concept (EDC) [177], the partially-stirred reactor approach (PaSR-LES) [178], the Eulerian Stochastic field method (ESF) [179] or the Linear Eddy Model (LEM) [180]. Tabulated chemistry methods have also demonstrated to be robust and efficient to predict premixed [163] and non-premixed combustion [181], including complex flow conditions such as dilution, gas recirculation, or dual fuel by simply adding additional dimensions into the manifold space [182]. For most of these conditions, the use of a mixture fraction Y_ξ and a progress variable Y_C was found to reproduce well the combustion process, even in partially premixed conditions [183], multi-mode combustion [166,184] and soot formation [185,186]. In order to account for turbulence-chemistry interactions, the use of presumed-shape Probability Density Functions (PDFs) has been a common approach to describe the influence of turbulence on the reaction rates [40,41,187], though more sophisticated closures have also been proposed such as stochastic fields [188,189] or flame thickening [175]. While these two approaches do not require additional dimensions into the flamelet database, their use in practical applications has been less extended due to the more complex coupling with the governing equations.

Despite the differences in the modelling descriptions of the methods, from a computational point of view, they share the same workloads: solving the scalar transport and evaluating the reaction rates. Considering that solving the scalar transport only involves the *assembly* of the right-hand side for explicit schemes, including evaluation of transport properties and source terms, efficient algorithms based on dynamic load balancing, vectorization and GPU-acceleration can be used to increase the computational performance during the assembly. Examples were presented in Section 3.

5.3. Stiff chemical integration

Despite the possibility to reduce the size of the reaction mechanism, the last step in the solution requires the integration of the chemical source term to obtain the concentrations for the subsequent time-step. The implementation of these iterative methods using HPC algorithms is critical to achieve high computational performance in combustion simulations, as the chemical integration is usually the limiting factor [190]. As described by Lu and Law [15], the increase in the number of species and reactions can exponentially increase the stiffness of the chemical mechanisms.

Different strategies based on implicit and explicit methods have been derived to integrate the chemical source term. These methods can be embedded in parallel algorithms to exploit the heterogeneous platforms based on CPUs+GPUs [90,191] or with data-driven methods [192,193]. A brief overview of some achievements in this field is presented below. While explicit methods are well suited when running simulations with small timesteps [150], like those involve in thermoacoustics, high speed flows or non-stiff chemistry, implicit methods are preferred when having stiff systems with large time steps. Methods based on backward difference formulation (BDF) are commonly used for implicit chemical integration and require a Jacobian matrix to solve the nonlinear algebraic system. Computing this matrix can be computationally expensive and scales quadratically with the number of species in the mechanism. Analytical Jacobians [194] can be used to reduce the cost of the ODE integration by reducing the scaling from quadratic to linear. Other methods based on adaptive preconditioners [195] or Krylov subspace sparse iterative solvers [196] can also be used to speed-up the integration. Schur-Krylov approximation algorithms (EISKA) based on the exponential of

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

the Jacobian matrix with the added dimensional reduction through Krylov subspace approximation were shown to have substantial speedup when compared to fully implicit methods [197]. BDF methods involve MPI communications and require high arithmetic-intensity operations, so they can benefit from the use of accelerators.

The advantages of both implicit and explicit methods can be combined into the so-called hybrid or semi-implicit methods. A dynamic adaptive integration based on time-scale separation is used by Muela et al. [157] to accelerate the chemistry calculation. The *G*-scheme [198,199] is a multi-scale adaptive model reduction algorithm based on scale separation using Computational Singular Perturbation (CSP). It uses a Jacobian-free time integration methods based on the extended robustness-enhanced numerical algorithm (ERENA) [200].

Another way to increase the performance of implicit methods is the use of accelerators to perform the high arithmetic-intensity operations. Several methodologies have been proposed for GPUs. Explicit methods for non-stiff chemistry were first ported to GPUs by Niemeyer and Sung [191], while Sewerin and Rigopoulos [90] developed a 3-stage/5th order implicit Runge-Kutta method called Radau5 for GPUs and achieved a speedup of x5 compared to the MPI/CPU counterpart in a fractional step method. Stone and Davis [201] ported a 5th order accurate variable coefficient BDF-solver DVODE to the CUDA framework and showed a speedup above x7 for a relatively small reaction mechanism with 19 species. Hybrid explicit/implicit methods like the CHEMEQ2 solver for dealing with stiff chemistry have also been tested in GPUs showing speedup factors of x14 for a heptane reaction mechanism [202]. Stone and co-workers [91] showcased load balancing techniques for stiff and non-stiff ODE integrations using operator-splitting schemes for multi-thread and instruction-level parallelism architectures to achieve speedups from x2.5 up to almost x5.0 for both CPU and GPU implementations. To efficiently exploit the computational power in exascale systems, the popular SUNDIALS library [203] has extended its support for application on current hybrid CPU+GPU systems. The matrix-based computations involved in computing the chemical sources terms favour GPU implementations, nonetheless, it was found by reaction-type classifications that the complexity of the individual reactions can influence the speedup obtained through GPUs [92].

Finally, a third block of methods, derived from data driven approaches and neural networks, have been applied to accelerate chemistry calculations. An adaptive time-integrator scheme based on CSP with a local, projection-based, reduced-order model (ROM) [192] has shown good potential for practical applications. The degrees of freedom with faster scales are removed by a projection of the chemical source terms onto a basis function, which is obtained by solving the eigen-system of its Jacobian matrix. The neural-network model replaces the most computationally expensive part of the algorithm, i.e. the local basis calculation, while maintaining a high level of robustness. Neural networks have also been used to aid the selection of an optimal ODE solver with a spatial and temporal basis [193]. The models are trained using a wide set of thermochemical states generated through partially-stirred reactors and flame simulations. All these methods show good potential to be used in high-fidelity simulations and should be tested on the new architectures to ensure good efficiency and performance during the chemical integration.

6. Computational strategies for multiphase flows

The use of DNS and LES for multiphase flows still remains today an extremely challenging problem due to the extensive range of spatial and temporal scales observed in turbulent flows of practical interest. During the fuel atomization, liquid ligaments are formed from the jet core that eventually break into droplets forming a populated droplet cloud at certain distance from the injector. Predicting such droplet population is of paramount importance, since it determines the evolution of the spray, but requires complex techniques to capture and track the liquid-gas interfaces. In the framework of RANS and LES, this aspect was partially alleviated by the use of blobs [204,205] or, alternatively, directly injecting a population of droplets (typically according to a Rosin-Rammler distribution) in the frame of Lagrangian models. However, in the path to develop self-closed models, HPC offers new possibilities for the modelling of the dense region of the spray by the application of high-fidelity methods to predict the breakup process [206–209]. In contrast to the dense region, a large population of droplets usually interacts with the surroundings in the dilute part of the spray. In such conditions, Lagrangian particles can be used to represent the droplets by point-sources. It is worth emphasizing the importance of accurate predictions of the spray, since it has a direct impact on combustion and pollutant formation, especially on pollutants like soot due to its dependency on mixture fraction. The air entrained controls the droplets sizes and the evaporation rate, which in turn influences the fuel/air mixing and eventually combustion. All these processes may show a strong dependence on the gas density, temperature, and nozzle geometry, and the models for atomization and evaporation usually consider these variables to give accurate predictions. While a wide variety of methods can be used to describe two-phase flows, see the classification proposed by Mirjalili et al. [47], this review focuses on the methods more commonly used in combustion of sprays, which in-

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

clude interface capturing methods and Lagrangian Particle Tracking. Dedicated subsections are presented next that aim to introduce recent advances in these two methodologies with focus on the algorithmic and HPC aspects of the strategies.

6.1. Interface capturing methods

The liquid phase is described as a continuous medium using an Eulerian formulation and interface capturing methods are conventionally used to track the evolution of the liquid-gas interface. These methods are usually applied in the dense region of the spray, where primary atomization occurs, since predicting the large population of droplets in the dilute region can be prohibitive with interface capturing methods [210]. Several strategies have been used in the literature to describe the liquid-gas interface with Eulerian methods, with the Volume of Fluid, level set and phase field methods being the most common [45,207,211,212]. VOF methods are sometimes preferred over other type of methods due to the intrinsic mass conservation properties of the scheme. However, it can introduce certain complexities due to the reconstruction of the interface and the calculation of interface geometrical properties [213]. The reconstruction step requires the capturing of sharp interfaces with their deformation and breakups, which may result in high computational costs. Another popular method is the conservative level set method (CLS), which was derived in [46,214] to circumvent the mass conservation properties for conventional level set methods based on distance functions. CLS methods require a re-initialization step to compress the interface to a target thickness based on the mesh resolution after the advection of the phase function [46,215]. Several proposals have been developed for the re-initialization equations, and a review on these approaches can be found in [47]. A dedicated review on the progress of level set methods to predict atomization and evaporation is given by Luo et al. [43]. Phase fields methods derived from the Cahn-Hilliard or the Allen-Cahn equations have also been extended to solve multiphase problems [47,216], but have not yet been applied to combustion problems to the authors knowledge.

With the aim of exploiting the potential of these methods, Zandian et al. [208] applied the level set and VOF in the frame of DNS simulations to cases with liquid Reynolds and gas Weber numbers up to 5000 and 36000, respectively, and contributed to explain the breakup mechanisms based on the surface and vortex dynamics. In [209], a DNS with a correction algorithm for high-density ratios coupled to the immersed boundary method is used to overcome the problem of complex geometries for FD codes. Excellent parallel performance was reported thanks to the use of parallel polydiagonal solvers in the implicit formulation [217]. While in general these methods have been applied to incompressible flows, recent developments also include the coupling with heat transfer and evaporation [218,219]. Other remarkable advancements consist of solving the interface in an Eulerian framework, but coupling to a Lagrangian framework once small droplets are formed [220]. Hermann [221] applied this methodology by using an efficient parallel algorithm up to 2000 processors, while Guillamon et al [222] also address Lagrangian droplet formation from resolved primary breakup simulations using a jet in crossflow configuration. Finally, more recent developments of two-phase flow models also include the moment of fluid method [223].

While of all the methods mentioned above are used in DNS-type approaches, diffuse interface methods have also been derived for LES [224–226]. In this family of models, the $\Sigma - Y$ model [227] has been extended significantly to describe inert and reacting sprays. This method was expanded to account for high density ratios by adding new terms that account for turbulence production caused by density fluctuations [228]. Other notable advancements for primary atomization modelling are the joint sub-grid probability density function of liquid volume and surface using stochastic methods [229] and the hybrid Eulerian-Lagrangian approach with self-closed subgrid turbulent atomization model [224]. A recent review by Jofre and Urzay [36] also discusses about the modelling of the diffuse-interfaces for transcritical flow conditions, usually encountered in high-pressure combustion systems. A new formulation is proposed, which is derived from constitutive laws in the Navier-Stokes equations for multicomponent flows and new forms of the stress tensor and transport fluxes of heat and species. While most of these methodologies have been applied to liquid fuel injection and atomization, their application to combustion simulations has been more limited [224,226].

All these methods have in common the need to capture thin interfaces and usually require the use of high-order methods and dynamic load balancing (DLB) techniques to ensure efficient parallel calculations [43]. Agbaglah et al. [230] developed a parallelization strategy based on domain decomposition with an octree-type adaptive grid together with a piecewise linear volume of fluid interface tracking method. Jofre et al. [70] developed a load balancing strategy for VOF for Cartesian and unstructured meshes with a speedup up to 12x with respect to the classical domain decomposition approach. Recent works have also shown speedups in the order of 10x by the use of GPU acceleration in a conservative level set method with a two-mesh approach [231]. The nature of the interface requires the use of highly refined meshes only in the vicinity of the interface, which strongly benefits from the use of Adaptive Mesh Refinement techniques [232].

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

6.2. Lagrangian Particle Tracking methods

Another family of methods used to describe the liquid phase is based on the use of a Lagrangian description for the droplets, while keeping an Eulerian formulation for the gas phase. A dedicated review on this topic is given by Subramaniam [44] and more specifically for spray flames by Jenny et al. [32]. Droplet methods are usually employed in spray flame simulations due to their simplicity of including heat and mass transfer models. A Lagrangian description of the droplets drastically reduces the numerical diffusion, however, due to the overpopulated cloud of droplets, only a statistical description of the stochastic processes is achievable if a representative cloud of points is considered. In this way, sets of droplets with similar properties can be clustered as parcels [233], giving rise to LPT methods. The droplets are subjected to atomization, heating, collisions and evaporation, which are computed according to sub-models that introduce additional source terms in the transport equations for the carrier phase (two-way and four-way coupling, with the later including interactions between droplets like collisions). In general, due to the underlying physical models devised for the description of droplet dynamics, the maximum liquid volume fraction within the cells is limited to 10% to neglect interactions between droplets [44]. This condition restricts the application of this kind of models to dilute sprays. These models have enjoyed a great popularity and have been applied to the modelling of a variety of sprays [44].

The first aspect to deal with LPT models for multiphase flows is the addition of droplet breakup models. Significant strides in this direction were made in the past by Senecal et al. [234] and Beale and Reitz [204], in which 'blobs' of a diameter equal to the nozzle diameter were injected. However, it has been reported that, as the interface is not resolved in LPT methods [235], a direct injection of a population of droplets is preferred in some cases [117,222] Downstream the break-up length, which delimits the primary atomization from secondary breakup, atomization has been modelled according to the Taylor analogy breakup (TAB) model, that has been widely extended along with the KH-RT hybrid model in which Kelvin-Helmholtz and Rayleigh-Taylor instabilities compete for the droplet breakup [204,236]. More advanced methods have also been developed for high speed sprays based on Lagrangian stochastic models [237]. However, based on previous findings, primary atomization is omitted and droplets are injected according to a Rosin-Rammler distribution many times in practical LES applications [6,24,111]. In such applications of spray flames, the thermodynamic conditions in the combustion chamber force a rapid evaporation of the droplets and the selection of appropriate evaporation models can have a strong influence on the spray pattern and eventually in the combustion process [238]. Particle tracking equations can be solved by using Newton-Raphson algorithms for the non-linear dynamic equations together with implicit schemes for the heat and mass transfer [44].

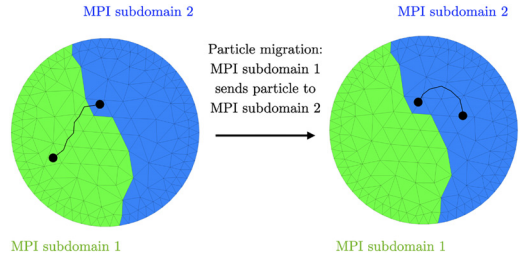

Fig. 14. Particle transport on a simplified mesh involving two MPI subdomains. Particle travels from element to element when solving its transport equation. When it reaches the interface of the subdomain, particle is migrated from subdomain 1 to subdomain 2 by using an MPI message containing its dynamical properties. Then particle is transported through subdomain 2.

LPT formulations have been successfully applied to reacting flows in the context of reciprocating combustion engines and gas turbines using finite rate and tabulated chemistry methods [6,24,116,239]. Spray flames with finite rate chemistry coupled to the Eulerian Stochastic fields [240], flamelet methods with presumed shape PDF [6,241] or flame thickening [242], finite rate with flame thickening [243], CMC [111] or Linear Eddy Model [180] have shown accurate integration of LPT sprays with most turbulent combustion models for LES.

From the computational point of view, one of the main challenges of LPT methods for spray flames is the intense computational requirements when tracking the large number of Lagrangian particles needed for high fidelity simulations. Particle tracking methods require many different operations, apart from solving the ODE's itself: cell/element location, particle-boundary interactions (deposition, bouncing), collisions and, in the context of HPC, migration. However, for the sake of brevity, and in order to maintain the focus on the computational aspects, the reader is referred to the articles and reviews to deepen in the techniques to treat particle-boundary interactions and collisions [44]. Concerning the cell/element identification, several approaches have been proposed to solve these mathematical problems [244], although the neighbor searching method is the most widely used in spray flame simulations. This method assumes that the particles only advance small distances at each time-step, so the location can be fast and adapted to hybrid meshes on complex geometries. Figure 14 shows an intentionally simplified

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

example of particle transport algorithm. The particle starts being transported in subdomain 1, going from one element to its neighboring elements at each time step. This strategy is generally chosen for practical reasons, as it is relatively easy and inexpensive to find the location in neighboring elements, and for accuracy reasons, to accurately account for the fluid variations of velocity and thermal properties.

Figure 14 also illustrates a classic inter-node parallelization. When the particle reaches the interface of a subdomain, this is migrated to the corresponding neighbor using an MPI message. Note that all the particle properties must be exchanged, meaning that if a simulation involves millions of particles, the weight of such communications can be absolutely penalizing.

In addition, for most reacting flow applications, the particles concentrate near the injector and most droplets evaporate just few millimeters downstream. Therefore, high concentration of particles tend to accumulate in specific regions of the domain resulting in high load imbalance. As the physics of the particles do not demand expensive computations for heat and mass transfer, the tracking algorithm can be expensive when a lot of particles are located in the same subdomain. Regarding the performance of the model according to the metrics discussed in Section 3.5, both *LB* and *CE* issues occur in LPT methods. Different load balancing strategies have been proposed in the literature to address this problem. A local time-step based algorithm for individual particles with a global time-step strategy for the gas phase was used to reduce the cost of the tracking algorithm [245]. Strategies based on spatial grid decomposition were compared with particle sharing algorithms, where particles are shared equally among all MPI ranks irrespective of their spatial location [246]. The particle sharing algorithm was found to be more efficient due to data locality and load balancing of particles in runs up to 32000 MPI ranks. A parallel algorithm including particle collisions based on a mirror domain technique showed a maximum speed-up up to x20 using a small number of CPU-cores [247]. A highly scalable parallel algorithm using a partitioned global address space (PGAS) programming model was demonstrated in a DNS of turbulent flow with around 300 million particles and 0.55 trillion grid points on 262,144 Cray XE6 cores [248]. Cubic splines were employed to minimize the need for communication using Co-Array Fortran features with one-sided communication. Thari et al. [106] proposed a parallel load balanced strategy for Eulerian-Lagrangian spray flame simulations using an asynchronous task-based approach obtaining more than 60% reduction in computational cost using a single computing node and also for multiple nodes [107]. A new algorithm for reacting flow simulations involving solid particles for massively parallel simulations was proposed by Dufresne et al. [249]. A non-blocking MPI algorithm for extra-processor communications with a packing/unpacking of halo data based on the Voronoi regions decomposition was developed to account for particle/wall interactions using a dynamic load balancing algorithm based on coloring. Houzeaux et al. [250] proposed a hybrid approach based on a multi-code execution where each solver (Lagrangian and Eulerian) access separate computer nodes, so that OpenMP can be used for the main loops in the LPT solver within each MPI subdomain. A load balance at node level is enforced by the use of the DLB library. This approach was tested for large scale HPC applications including up to 1024 MPI ranks using one-way coupling.

The performance of the load balancing strategy is a key element to enable certain methodologies to be run efficiently on supercomputing architectures for specific algorithms. This has motivated the use of hybrid architectures CPU/GPU and accelerators. A computational strategy based on hardware ray tracing cores and GPU parallel computing was used to accelerate a particle tracking method for unstructured meshes [251]. A GPU implementation using four NVIDIA GPUs with one-way coupling and a number of particles in the order of $O(N_p) \approx 10^7$, was demonstrated for the solution of a particle-laden turbulent flow. A speed-up of x14 was shown compared to the CPU version [252]. The library Grit was developed to track the Lagrangian particles across different HPC architectures with CPU/GPU ensuring performance-portability [63]. The parallel strategy is based on the Kokkos library and focuses on shared and distributed memory parallelism. Grit was tested on the pre-exascale machine Summit in a direct numerical simulation of multiphase flow.

7. Conclusions and directions for future work

The accurate simulation of combustion systems arises as one of the most complex applications in computational science, as it involves strongly coupled physical and chemical phenomena. This complexity is originated from the broad spatial and temporal scales of the turbulence, chemistry, and liquid-gas interactions, in the case of spray flames. Considering the increase in computational power in the last decade and the forthcoming Exascale supercomputers, new horizons in computational modelling and simulation can be envisioned. The necessity of higher computing resources to increase simulations reliability makes combustion simulations an Exascale application. Hence, the efficient use of upcoming architectures demands new algorithms and computational strategies based on combining physical models, numerical methods, and parallel algorithms adapted to the hardware architectures.

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

This review introduces recent developments in numerical methods for large-eddy simulations and direct-numerical simulations focusing on the computational performance and algorithmic capabilities. The increasing complexity of new computing architectures, with tightly coupled CPUs and GPUs and high levels of parallelism, requires new parallel models and algorithms to reach the required level of concurrency. Advances in dynamic load balancing, vectorization, and GPU computing have allowed a significant acceleration and increase of efficiency for combustion simulations in HPC environments. The higher levels of parallelism and concurrency are organized through different levels: inter-node, intra-node, and chip-level, respectively. Detail descriptions on how to exploit efficiently these hierarchies for combustion simulations with recommendations for defining the correct metrics and techniques for parallel performance evaluations have been presented.

This review paper provides an outlook of the state-of-the-art on three fundamental areas of relevance for high-fidelity combustion simulation: extension of high-order methods for reacting flows, advanced methodologies to include detailed chemistry and descriptions of the multiphase flow in spray flame simulations. It introduces recent developments in these fields with focus on the HPC-related aspects and their impact on high-fidelity simulations. While this review has focused mainly on methodologies based on the use of the continuous equations for fluid mechanics, the authors would like to highlight the tremendous progress made on meshless methods like Lattice-Boltzmann or Smoothed Particle hydrodynamics (SPH) for multiphase and reacting flows. These methodologies have shown excellent predictive capabilities to describe multiphase flows, turbulent flames and thermoacoustics, and are very well suited for HPC systems and heterogeneous architectures. However, a proper review of these methods would extend substantially the length of this study, and it is left out of the scope of the present work.

Declaration of Competing Interest

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

Acknowledgments

The research leading to these results has received funding from the European Union's Horizon 2020 Programme under the CoEC project, grant agreement No. 952181 and the CoE RAISE project grant agreement no. 951733.

References

[1] C.S. Yoo, E.S. Richardson, R. Sankaran, J.H. Chen, A DNS study on the stabilization mechanism of a turbulent lifted ethylene jet flame in highly-heated coflow, Proc. Combust. Inst. 33 (1) (2011) 1619–1627, doi:10.1016/j.proci.2010.06.147. https://www.sciencedirect.com/science/article/pii/S154074891000310X

[2] A. Attili, F. Bisetti, M.E. Mueller, H. Pitsch, Formation, growth, and transport of soot in a three-dimensional turbulent non-premixed jet flame, Combust. Flame 161 (7) (2014) 1849–1865, doi:10.1016/j.combustflame.2014.01.008. https://www.sciencedirect.com/science/article/pii/S0010218014000133

[3] S. Luca, A. Attili, E. Lo Schiavo, F. Creta, F. Bisetti, On the statistics of flame stretch in turbulent premixed jet flames in the thin reaction zone regime at varying Reynolds number, Proc. Combust. Inst. 37 (2) (2019) 2451–2459, doi:10.1016/j.proci.2018.06.194. https://www.sciencedirect.com/science/article/pii/S1540748918303808

[4] C.P. Arroyo, J. Dombard, F. Duchaine, L. Gicquel, B. Martin, N. Odier, G. Staffelbach, Towards the Large-Eddy simulation of a full engine: integration of a 360 azimuthal degrees fan, compressor and combustion chamber. Part I: methodology and initialisation, J. Global Power Propul. Soc. (May) (2021) 1–16, doi:10.33737/jgpps/133115.

[5] Y. Che, M. Yang, C. Xu, Y. Lu, Petascale scramjet combustion simulation on the Tianhe-2 heterogeneous supercomputer, Parallel Comput. 77 (2018) 101–117, doi:10.1016/j.parco.2018.06.004. https://www.sciencedirect.com/science/article/pii/S016781911830190X

[6] J. Benajes, J.M. García-Oliver, J.M. Pastor, I. Olmeda, A. Both, D. Mira, Analysis of local extinction of a n-heptane spray flame using Large-Eddy simulation with tabulated chemistry, Combust. Flame 235 (2022) 111730.

[7] M. Schmitt, C.E. Frouzakis, A.G. Tomboulides, Y.M. Wright, K. Boulouchos, Direct numerical simulation of the effect of compression on the flow, temperature and composition under engine-like conditions, Proc. Combust. Inst. 35 (3) (2015) 3069–3077, doi:10.1016/j.proci.2014.06.097. https://www.sciencedirect.com/science/article/pii/S1540748914002557

[8] S. Wu, S.S. Patel, M.M. Ameen, Investigating the origins of cyclic variability in internal combustion engines using wall-resolved large eddy simulations, in: ASME 2021 Internal Combustion Engine Division Fall Technical Conference, 2021, doi:10.1115/ICEF2021-67671. V001T06A003, https://asmedigitalcollection.asme.org/ICEF/proceedings-pdf/ICEF2021/85512/V001T06A003/6802928/v001t06a003-icef2021-67671.pdf

[9] H. Pitsch, Large-Eddy simulation of turbulent combustion, Annu. Rev. Fluid Mech. 38 (1) (2006) 453–482, doi:10.1146/annurev.fluid.38.050304.092133.

[10] T. Lu, C.K. Law, Toward accommodating realistic fuel chemistry in large-scale computations, Prog. Energy Combust. Sci. 35 (2) (2009) 192–215.

[11] D. Veynante, L. Vervisch, Turbulent combustion modeling, Prog. Energy Combust. Sci. 28 (3) (2002) 193–266, doi:10.1016/S0360-1285(01)00017-X.

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

https://www.sciencedirect.com/science/article/pii/S036012850100017X
[12] A.N. Lipatnikov, Stratified turbulent flames: Recent advances in understanding the influence of mixture inhomogeneities on premixed combustion and modeling challenges, Prog. Energy Combust. Sci. 62 (2017) 87–132, doi:10.1016/j.pecs.2017.05.001. https://www.sciencedirect.com/science/article/pii/S0360128517300217
[13] A. Felden, P. Pepiot, L. Esclapez, E. Riber, B. Cuenot, Including analytically reduced chemistry (ARC) in CFD applications, Acta Astronaut. 158 (2019) 444–459, doi:10.1016/j.actaastro.2019.03.035.
[14] E.S. Richardson, J.H. Chen, Analysis of turbulent flame propagation in equivalence ratio-stratified flow, Proc. Combust. Inst. 36 (2) (2017) 1729–1736, doi:10.1016/j.proci.2016.06.140. https://www.sciencedirect.com/science/article/pii/S1540748916301985
[15] T. Lu, C.K. Law, Toward accommodating realistic fuel chemistry in large-scale computations, Prog. Energy Combust. Sci. 35 (2) (2009) 192–215.
[16] Y. Wang, A. Raj, S.H. Chung, A PAH growth mechanism and synergistic effect on PAH formation in counterflow diffusion flames, Combust. Flame 160 (9) (2013) 1667–1676.
[17] A. Kalbhor, J. van Oijen, An assessment of the sectional soot model and FGM tabulated chemistry coupling in laminar flame simulations, Combust. Flame 229 (2021) 111381.
[18] F. Ferraro, C. Russo, R. Schmitz, C. Hasse, M. Sirignano, Experimental and numerical study on the effect of oxymethylene ether-3 (OME3) on soot particle formation, Fuel 286 (2021) 119353, doi:10.1016/j.fuel.2020.119353. https://www.sciencedirect.com/science/article/pii/S0016236120323498
[19] R.J. Kee, G. Dixon-Lewis, J. Warnatz, M.E. Coltrin, J.A. Miller, H.K. Moffat, Transport: A Software Package for the Evaluation of Gas-Phase, Multicomponent Transport Properties, Chemkin Collection, 1999.
[20] S.B. Pope, Small scales, many species and the manifold challenges of turbulent combustion, Proc. Combust. Inst. 34 (1) (2013) 1–31, doi:10.1016/j.proci.2012.09.009. https://www.sciencedirect.com/science/article/pii/S1540748912003963
[21] R. Reitz, F. Bracco, Mechanism of atomization of a liquid jet, Phys. Fluids 25 (10) (1982) 1730–1742.
[22] M. Pilch, C. Erdman, Use of breakup time data and velocity history data to predict the maximum size of stable fragments for acceleration-induced breakup of a liquid drop, Int. J. Multiphase Flow 13 (6) (1987) 741–757, doi:10.1016/0301-9322(87)90063-2. https://www.sciencedirect.com/science/article/pii/0301932287900632
[23] G.M. Magnotti, S. Som, Assessing fuel property effects on cavitation and erosion propensity using a computational fuel screening tool, J. Eng. Gas Turbines Power 142 (11) (2020) 111015, doi:10.1115/1.4048457. https://asmedigitalcollection.asme.org/gasturbinespower/article-pdf/142/11/111015/6584643/gtp_142_11_111015.pdf
[24] J. Desantes, J.M. García-Oliver, R. Novella, E. Pérez-Sánchez, Application of a flamelet-based CFD combustion model to the LES simulation of a diesel-like reacting spray, Comput. Fluids 200 (2020) 104419.
[25] R. Torelli, S. Som, Y. Pei, Y. Zhang, M. Traver, Influence of fuel properties on internal nozzle flow development in a multi-hole diesel injector, Fuel 204 (2017) 171–184.
[26] J.M. Apeloig, F.-X. d'Herbigny, F. Simon, P. Gajan, M. Orain, S. Roux, Liquid-fuel behavior in an aeronautical injector submitted to thermoacoustic instabilities, J. Propul. Power 31 (1) (2015) 309–319, doi:10.2514/1.B35290.
[27] A. Asuri Mukundan, T. Ménard, J.C. Brändle de Motta, A. Berlemont, Detailed numerical simulations of primary atomization of airblasted liquid sheet, Int. J. Multiphase Flow 147 (2022) 103848, doi:10.1016/j.ijmultiphaseflow.2021.103848. https://www.sciencedirect.com/science/article/pii/S0301932221002718
[28] L.C. Mesquita, A. Vié, L. Zimmer, S. Ducruix, Numerical analysis of flame shape bifurcation in a two-stage swirled liquid burner using large eddy simulation, Proc. Combust. Inst. 38 (4) (2021) 5971–5978, doi:10.1016/j.proci.2020.06.044. https://www.sciencedirect.com/science/article/pii/S1540748920300705
[29] O. Lehmkuhl, D. Mira, L. Gasparino, H. Owen, G. Houzeaux, Large-Eddy simulation of primary atomization using an entropy stable conservative level set, in: M. García-Villalba, H. Kuerten, M.V. Salvetti (Eds.), Direct and Large Eddy Simulation XII, Springer International Publishing, Cham, 2020, pp. 207–213.
[30] P.M. de Oliveira, D. Fredrich, G. De Falco, I. El Helou, A. D'Anna, A. Giusti, E. Mastorakos, Soot-free low-NOx aeronautical combustor concept: the lean azimuthal flame for kerosene sprays, Energy Fuels 35 (9) (2021) 7092–7106.
[31] LES Study on spray combustion with renewable fuels under ECN spray-A conditions, ASME 2021 Internal Combustion Engine Division Fall Technical Conference, 2021. V001T06A004, https://asmedigitalcollection.asme.org/ICEF/proceedings-pdf/ICEF2021/85512/V001T06A004/6802926/v001t06a004-icef2021-67745.pdf10.1115/ICEF2021-67745
[32] P. Jenny, D. Roekaerts, N. Beishuizen, Modeling of turbulent dilute spray combustion, Prog. Energy Combust. Sci. 38 (6) (2012) 846–887, doi:10.1016/j.pecs.2012.07.001. https://www.sciencedirect.com/science/article/pii/S0360128512000445
[33] Y. Zhang, H. Wang, A. Both, L. Ma, M. Yao, Effects of turbulence-chemistry interactions on auto-ignition and flame structure for n-dodecane spray combustion, Combust. Theory Model. 23 (5) (2019) 907–934.
[34] L. Ma, D. Roekaerts, Modeling of spray jet flame under MILD condition with non-adiabatic FGM and a new conditional droplet injection model, Combust. Flame 165 (2016) 402–423.
[35] P.E. Lapenna, F. Creta, Mixing under transcritical conditions: an a-priori study using direct numerical simulation, J. Supercrit. Fluids 128 (2017) 263–278, doi:10.1016/j.supflu.2017.05.005. https://www.sciencedirect.com/science/article/pii/S0896844616305782

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

[36] L. Jofre, J. Urzay, Transcritical diffuse-interface hydrodynamics of propellants in high-pressure combustors of chemical propulsion systems, Prog. Energy Combust. Sci. 82 (2021) 100877, doi:10.1016/j.pecs.2020.100877. https://www.sciencedirect.com/science/article/pii/S0360128520300873
[37] D. Drikakis, M. Frank, G. Tabor, Multiscale computational fluid dynamics, Energies 12 (17) (2019), doi:10.3390/en12173272. https://www.mdpi.com/1996-1073/12/17/3272
[38] S.B. Pope, Turbulent Flows, Cambridge University Press, 2000.
[39] A. Attili, S. Luca, D. Denker, F. Bisetti, H. Pitsch, Turbulent flame speed and reaction layer thickening in premixed jet flames at constant Karlovitz and increasing Reynolds numbers, Proc. Combust. Inst. 38 (2) (2021) 2939–2947, doi:10.1016/j.proci.2020.06.210. https://www.sciencedirect.com/science/article/pii/S1540748920303023
[40] D. Mira, O. Lehmkuhl, A. Both, P. Stathopoulos, T. Tanneberger, T.G. Reichel, C.O. Paschereit, M. Vázquez, G. Houzeaux, Numerical characterization of a premixed hydrogen flame under conditions close to flashback, Flow Turbul. Combust. 104 (2) (2020) 479–507.
[41] S. Gövert, D. Mira, J.B. Kok, M. Vázquez, G. Houzeaux, The effect of partial premixing and heat loss on the reacting flow field prediction of a swirl stabilized gas turbine model combustor, Flow Turbul. Combust. 100 (2) (2018) 503–534.
[42] A.R. Masri, Turbulent combustion of sprays: from dilute to dense, Combust. Sci. Technol. 188 (10) (2016) 1619–1639, doi:10.1080/00102202.2016.1198788.
[43] K. Luo, C. Shao, M. Chai, J. Fan, Level set method for atomization and evaporation simulations, Prog. Energy Combust. Sci. 73 (2019) 65–94, doi:10.1016/j.pecs.2019.03.001. https://www.sciencedirect.com/science/article/pii/S0360128518301710
[44] S. Subramaniam, Lagrangian-Eulerian methods for multiphase flows, Prog. Energy Combust. Sci. 39 (2) (2013) 215–245, doi:10.1016/j.pecs.2012.10.003. https://www.sciencedirect.com/science/article/pii/S0360128512000603
[45] V. Boniou, T. Schmitt, A. Vié, Comparison of interface capturing methods for the simulation of two-phase flow in a unified low-Mach framework, Int. J. Multiphase Flow 149 (2022) 103957, doi:10.1016/j.ijmultiphaseflow.2021.103957. https://www.sciencedirect.com/science/article/pii/S0301932221003530
[46] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, J. Comput. Phys. 210 (1) (2005) 225–246.
[47] B.S. Mirjalili, S.S.L. Jain, M.S. Dodd, Interface–capturing methods for two-phase flows: an overview and recent developments, 2017.
[48] R. Janodet, C. Guillamón, V. Moureau, R. Mercier, G. Lartigue, P. Bénard, T. Ménard, A. Berlemont, A massively parallel accurate conservative level set algorithm for simulating turbulent atomization on adaptive unstructured grids, J. Comput. Phys. (2022) 111075, doi:10.1016/j.jcp.2022.111075. https://www.sciencedirect.com/science/article/pii/S0021999122001371
[49] A. Attili, (Personal communication).
[50] G. Moore, Progress in digital integrated electronics, in: International Electron Devices Meeting Technical Digest, 1975, pp. 11–13.
[51] R. Dennard, F. Gaensslen, H.-N. Yu, V. Rideout, E. Bassous, A. LeBlanc, Design of ion-implanted MOSFET's with very small physical dimensions, IEEE J. Solid-State Circuits 9 (5) (1974) 256–268, doi:10.1109/JSSC.1974.1050511.
[52] Description of marenostrum 4 supercomputer,
[53] J.J. Dongarra, Performance of various computers using standard linear equations software, SIGARCH Comput. Archit. News 20 (3) (1992) 22–44, doi:10.1145/141868.141871.
[54] E. Strohmaier, J. Dongarra, H. Simon, M. Meuer, H. Meuer, Top 500 list,
[55] J. Dongarra, M.A. Heroux, P. Luszczek, High-performance conjugate-gradient benchmark: a new metric for ranking high-performance computing systems, Int. J. High Perform.Comput. Appl. 30 (1) (2016) 3–10, doi:10.1177/1094342015593158.
[56] C.E. Leiserson, N.C. Thompson, J.S. Emer, B.C. Kuszmaul, B.W. Lampson, D. Sanchez, T.B. Schardl, There's plenty of room at the top: what will drive computer performance after moore's law? Science 368 (6495) (2020) eaam9744, doi:10.1126/science.aam9744.
[57] J.H. Chen, A. Choudhary, B. de Supinski, M. DeVries, E.R. Hawkes, S. Klasky, W.K. Liao, K.L. Ma, J. Mellor-Crummey, N. Podhorszki, R. Sankaran, S. Shende, C.S. Yoo, Terascale direct numerical simulations of turbulent combustion using S3D, Comput. Sci. Discov. 2 (1) (2009) 015001, doi:10.1088/1749-4699/2/1/015001.
[58] H. Tang, R. Haynes, G. Houzeaux, A review of domain decomposition methods for simulation of fluid flows: concepts, algorithms, and applications, Arch. Computat. Methods Eng. 28 (2021) 841–873, doi:10.1007/s11831-019-09394-0. https://www.sciencedirect.com/science/article/pii/S0021999116301966
[59] MPI: a message-passing interface standard version 3.0(http://mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf).
[60] OpenMP technical report 6: Version 5.0 preview 2, November 2017, (http://www.openmp.org/wp-content/uploads/openmp-TR6.pdf).
[61] CUDA toolkit documentation v11.6.0, January 2022, (https://docs.nvidia.com/cuda).
[62] H. Carter Edwards, C.R. Trott, D. Sunderland, Kokkos: enabling manycore performance portability through polymorphic memory access patterns, J. Parallel Distrib. Comput. 74 (12) (2014) 3202–3216, doi:10.1016/j.jpdc.2014.07.003. Domain-Specific Languages and High-Level Frameworks for High-Performance Computing, https://www.sciencedirect.com/science/article/pii/S0743731514001257
[63] W. Ge, R. Sankaran, J.H. Chen, Development of a CPU/GPU portable software library for Lagrangian-Eulerian simulations of liquid sprays, Int. J. Multiphase Flow 128 (2020) 103293, doi:10.1016/j.ijmultiphaseflow.2020.103293. https://www.sciencedirect.com/science/article/pii/S030193221930953X
[64] M. Bauer, S. Treichler, E. Slaughter, A. Aiken, Legion: Expressing locality and independence with logical regions, in: SC '12: Proceedings of the international conference on high performance comput-

ing, networking, storage and analysis, 2012, pp. 1–11, doi:10.1109/SC.2012.71.
[65] P. Fischer, S. Kerkemeier, M. Min, Y.-H. Lan, M. Phillips, T. Rathnayake, E. Merzari, A. Tomboulides, A. Karakus, N. Chalmers, T. Warburton, NekRS, a GPU-accelerated spectral element Navier-Stokes solver, 2021, 2104.05829
[66] M. Day, L. Esclapez, M.H. de Frahan, J. Rood, H. Sitaraman, N. Wimer, R. Grout, J. Chen, Pele: an exascale-ready suite of combustion codes, in: 18th Numerical Combustion Conference, 2022. San Diego, CA May 11
[67] S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knep-ley, L.C. McInnes, B.F. Smith, H. Zhang, Extrae: generating Paraver trace-files for a post-mortem analysis,
[68] R.T. Mills, G.E. Hammond, P.C. Lichtner, V. Sripathi, G.K. Mahinthakumar, B.F. Smith, Modeling subsurface reactive flows using leadership-class computing, J. Phys. Conf. Ser. 180 (2009) 012062, doi:10.1088/1742-6596/180/1/012062.
[69] G. Karypis, V. Kumar, Multilevel k-way partitioning scheme for irregular graphs, J. Parallel Distrib. Comput. 48 (1) (1998) 96–129, doi:10.1006/jpdc.1997.1404. https://www.sciencedirect.com/science/article/pii/S0743731597914040
[70] R. Borrell, J.C. Cajas, D. Mira, A. Taha, S. Koric, M. Vázquez, G. Houzeaux, Parallel mesh partitioning based on space filling curves, Comput. Fluids 173 (2018) 264–272, doi:10.1016/j.compfluid.2018.01.040. https://www.sciencedirect.com/science/article/pii/S0045793018300446
[71] C. Farhat, L. Crivelli, A general approach to nonlinear fe computations on shared-memory multiprocessors, Comput. Methods Appl. Mech.Eng. 72 (2) (1989) 153–171, doi:10.1016/0045-7825(89)90157-6. https://www.sciencedirect.com/science/article/pii/0045782589901576
[72] R. Aubry, G. Houzeaux, M. Vázquez, J.M. Cela, Some useful strategies for unstructured edge-based solvers on shared memory machines, Int. J. Numer. MethodsEng. 85 (5) (2011) 537–561, doi:10.1002/nme.2973. https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2973
[73] L. Thébault, E. Petit, M. Tchiboukdjian, Q. Dinh, W. Jalby, Divide and conquer parallelization of finite element method assembly, in: International Conference on Parallel Computing - ParCo2013, in: Advances in Parallel Computing, vol. 25, 2013, pp. 753–762, doi:10.3233/978-1-61499-381-0-753. Munich (Germany)
[74] M. Garcia-Gasulla, G. Houzeaux, R. Ferrer, A. Artigues, V. López, J. Labarta, M. Vázquez, MPI+X: task-based parallelisation and dynamic load balance of finite element assembly, Int. J. Comput. Fluid Dyn. 33 (3) (2019) 115–136, doi:10.1080/10618562.2019.1617856.
[75] L. Környei, Parallel implementation of a combustion chamber simulation with MPI-OpenMP hybrid techniques, in: 2012 Proceedings of the 35th International Convention MIPRO, 2012, pp. 356–361.
[76] Y.Y. Zeng, W.T. Zhao, Z.H. Wang, Hybrid MPI + OpenMP parallelization of scramjet simulation with hypergraph partitioning, in: Advances in Manufacturing Science and Engineering, in: Advanced Materials Research, vol. 712, Trans Tech Publications Ltd, 2013, pp. 1294–1297, doi:10.4028/www.scientific.net/AMR. 712–715.1294.
[77] P. Mohanamuraly, G. Staffelbach, Hardware locality-aware partitioning and dynamic load-balancing of unstructured meshes for large-scale scientific applications, in: PASC '20: Proceedings of the Platform for Advanced Scientific Computing Conference, Association for Computing Machinery, New York, NY, USA, 2020, doi:10.1145/3394277.3401851.
[78] D.F. Bacon, S.L. Graham, O.J. Sharp, Compiler transformations for high-performance computing, ACM Comput. Surv. 26 (4) (1994) 345–420, doi:10.1145/197405.197406.
[79] T. Zirwes, F. Zhang, J. Denev, P. Habisreuther, H. Bockhorn, D. Trimis, Improved vectorization for efficient chemistry computations in OpenFOAM for large scale combustion simulations, High Performance Computing in Science and Engineering ' 18, 2019.
[80] R. Borrell, D. Dosimont, M. Garcia-Gasulla, G. Houzeaux, O. Lehmkuhl, V. Mehta, H. Owen, M. Vázquez, G. Oyarzun, Heterogeneous CPU/GPU co-execution of CFD simulations on the POWER9 architecture: Application to airplane aerodynamics, Future Gener. Comput. Syst. 107 (2020) 31–48, doi:10.1016/j.future.2020.01.045. https://www.sciencedirect.com/science/article/pii/S0167739X1930994X
[81] F. Banchelli, G. Oyarzuna, M. Garcia-Gasulla, F. Mantovani, A. Both, G. Houzeaux, D. Mira, A portable coding strategy to exploit vectorization on combustion simulations, Comput. Fluids (2022) In-press.
[82] S. Jubertie, F. Dupros, F. De Martin, Vectorization of a spectral finite-element numerical kernel, in: Proceedings of the 2018 4th Workshop on Programming Models for SIMD/Vector Processing, in: WPMVP'18, Association for Computing Machinery, New York, NY, USA, 2018, doi:10.1145/3178433.3178441.
[83] N.J. Curtis, K.E. Niemeyer, C.-J. Sung, Using simd and simt vectorization to evaluate sparse chemical kinetic Jacobian matrices and thermochemical source terms, Combust. Flame 198 (2018) 186–204, doi:10.1016/j.combustflame.2018.09.008. https://www.sciencedirect.com/science/article/pii/S0010218018303997
[84] E. Cuthill, J. McKee, Reducing the bandwidth of sparse symmetric matrices, in: Proceedings of the 1969 24th National Conference, in: ACM '69, Association for Computing Machinery, New York, NY, USA, 1969, pp. 157–172, doi:10.1145/800195.805928.
[85] R. Löhner, Cache-efficient renumbering for vectorization, Int. J. Numer. MethodsBiomed. Eng. 26 (5) (2010) 628–636, doi:10.1002/cnm.1160. https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.1160
[86] G. Oyarzun, D. Mira, G. Houzeaux, Performance assessment of CUDA and OpenACC in large scale combustion simulations, 2021, 2107.11541
[87] S. Memeti, L. Li, S. Pllana, J. Kołodziej, C. Kessler, Benchmarking OpenCL, OpenACC, OpenMP, and CUDA: programming productivity, performance, and energy consumption, in: Proceedings of the 2017 Workshop on Adaptive Resource Management and Scheduling for Cloud Computing, in:

ARMS-CC '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 1–6, doi:10.1145/3110355.3110356.
[88] A. Corrigan, F. Camelli, R. Löhner, F. Mut, Semi-automatic porting of a large-scale Fortran CFD code to GPUs, Int. J. Numer. MethodsFluids 69 (2) (2012) 314–331, doi:10.1002/fld.2560. https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.2560
[89] N.J. Curtis, K.E. Niemeyer, C.-J. Sung, Using SIMD and SIMT vectorization to evaluate sparse chemical kinetic Jacobian matrices and thermochemical source terms, Combust. Flame 198 (2018) 186–204, doi:10.1016/j.combustflame.2018.09.008. https://www.sciencedirect.com/science/article/pii/S0010218018303997
[90] F. Sewerin, S. Rigopoulos, A methodology for the integration of stiff chemical kinetics on GPUs, Combust. Flame 162 (4) (2015) 1375–1394, doi:10.1016/j.combustflame.2014.11.003. https://www.sciencedirect.com/science/article/pii/S0010218014003459
[91] C.P. Stone, A.T. Alferman, K.E. Niemeyer, Accelerating finite-rate chemical kinetics with coprocessors: comparing vectorization methods on GPUs, MICs, and CPUs, Comput. Phys. Commun. 226 (2018) 18–29, doi:10.1016/j.cpc.2018.01.015. https://www.sciencedirect.com/science/article/pii/S0010465518300286
[92] S. Barwey, V. Raman, A neural network-inspired matrix formulation of chemical kinetics for acceleration on GPUs, Energies 14 (9) (2021). https://www.mdpi.com/1996-1073/14/9/2710
[93] B. Vermeire, F. Witherden, P. Vincent, On the utility of GPU accelerated high-order methods for unsteady flow simulations: a comparison with industry-standard tools, J. Comput. Phys. 334 (2017) 497–521, doi:10.1016/j.jcp.2016.12.049. https://www.sciencedirect.com/science/article/pii/S0021999116307136
[94] M. Wagner, S. Mohr, J. Gimánez, J. Labarta, A structured approach to performance analysis, in: Tools for High Performance Computing 2017, Springer, Cham, 2017, pp. 1–15, doi:10.1007/978-3-030-11987-4.
[95] Extrae: generating paraver trace-files for a post-mortem analysis, 2021, (https://tools.bsc.es/extrae).
[96] A. Knüpfer, C. Rössel, D.a. Mey, S. Biersdorff, K. Diethelm, D. Eschweiler, M. Geimer, M. Gerndt, D. Lorenz, A. Malony, W.E. Nagel, Y. Oleynik, P. Philippen, P. Saviankou, D. Schmidl, S. Shende, R. Tschüter, M. Wagner, B. Wesarg, F. Wolf, Score-P: a joint performance measurement run-time infrastructure for Periscope, Scalasca, TAU, and Vampir, in: H. Brunst, M.S. Müller, W.E. Nagel, M.M. Resch (Eds.), Tools for High Performance Computing 2011, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 79–91.
[97] V. Pillet, J. Labarta, T. Cortes, S. Girona, PARAVER: a tool to visualize and analyze parallel code, in: Proceedings of WoTUG-18: transputer and occam developments, vol. 44, IOS Press, 1995, pp. 17–31.
[98] M. Geimer, F. Wolf, B.J.N. Wylie, E. Åbraham, D. Becker, B. Mohr, The Scalasca performance toolset architecture, Concurr. Comput. Pract.Exp. 22 (6) (2010) 702–719, doi:10.1002/cpe.1556.
[99] V. Dolean, P. Jolivet, F. Nataf, An Introduction to Domain Decomposition Methods, SIAM, 2015.
[100] S. Badia, A. Martín, J. Principe, Implementation and scalability analysis of balancing domain decomposition methods, Arch. Computat. Methods Eng. 20 (2013) 239–262, doi:10.1007/s11831-013-9086-4.
[101] G. Ofenbeck, R. Steinmann, V. Caparros, D.G. Spampinato, M. Püschel, Applying the roofline model, in: 2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2014, pp. 76–85, doi:10.1109/ISPASS.2014.6844463.
[102] M. Vázquez, G. Houzeaux, S. Koric, A. Artigues, J. Aguado-Sierra, R. Arís, D. Mira, H. Calmet, F. Cucchietti, H. Owen, et al., Alya: multiphysics engineering simulation toward exascale, J. Comput. Sci. 14 (2016) 15–27.
[103] J. Kodavasal, K. Harms, P. Srivastava, S. Som, S. Quan, K. Richards, M. García, Development of a stiffness-based chemistry load balancing scheme, and optimization of input/output and communication, to enable massively parallel high-fidelity internal combustion engine simulations, J. Energy Resources Technol. 138 (5) (2016) 052203, doi:10.1115/1.4032623. https://asmedigitalcollection.asme.org/energyresources/article-pdf/138/5/052203/6148052/jert_138_05_052203.pdf
[104] G. Ramírez-Miranda, D. Mira, E.J. Pérez-Sánchez, A. Surapaneni, R. Borrell, G. Houzeaux, M. Garcia-Gasulla, Dynamic load balance of chemical source term evaluation in high-fidelity combustion simulations, Comput. FluidsIn press
[105] G. Houzeaux, M. Garcia, J.C. Cajas, A. Artigues, E. Olivares, J. Labarta, M. Vázquez, Dynamic load balance applied to particle transport in fluids, Int. J. Comput. FluidDyn. 30 (6) (2016) 408–418, doi:10.1080/10618562.2016.1227070.
[106] A. Thari, N. Treleaven, M. Staufer, G. Page, Parallel load-balancing for combustion with spray for large-scale simulation, J. Comput. Phys. 434 (2021) 110187, doi:10.1016/j.jcp.2021.110187. https://www.sciencedirect.com/science/article/pii/S0021999121000826
[107] A. Thari, M. Staufer, G. Page, Asynchronous task based Eulerian-Lagrangian parallel solver for combustion applications, J. Comput. Phys. 458 (2022) 111103, doi:10.1016/j.jcp.2022.111103. https://www.sciencedirect.com/science/article/pii/S0021999122001656
[108] F.E. Hernández Pérez, N. Mukhadiyev, X. Xu, A. Sow, B.J. Lee, R. Sankaran, H.G. Im, Direct numerical simulations of reacting flows with detailed chemistry using many-core/GPU acceleration, Comput. Fluids 173 (2018) 73–79, doi:10.1016/j.compfluid.2018.03.074. https://www.sciencedirect.com/science/article/pii/S0045793018301786
[109] P. Vincent, A. Farrington, F. Witherden, A. Jameson, An extended range of stable-symmetric-conservative flux reconstruction correction functions, Comput. Methods Appl. Mech.Eng. 296 (2015) 248–272, doi:10.1016/j.cma.2015.07.023. https://www.sciencedirect.com/science/article/pii/S0045782515002418

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

[110] M. Emmett, W. Zhang, J.B. Bell, High-order algorithms for compressible reacting flow with complex chemistry, Combust. Theory Model. 18 (3) (2014) 361–387.
[111] M.P. Sitte, E. Mastorakos, Large eddy simulation of a spray jet flame using doubly conditional moment closure, Combust. Flame 199 (2019) 309–323, doi:10.1016/j.combustflame.2018.08.026. https://www.sciencedirect.com/science/article/pii/S0010218018303973
[112] X. Wen, L. Dressler, A. Dreizler, A. Sadiki, J. Janicka, C. Hasse, Flamelet LES of turbulent premixed/stratified flames with H2 addition, Combust. Flame 230 (2021) 111428, doi:10.1016/j.combustflame.2021.111428. https://www.sciencedirect.com/science/article/pii/S001021802100167X
[113] A. Both, O. Lehmkuhl, D. Mira, M. Ortega, Low-dissipation finite element strategy for low Mach number reacting flows, Comput. Fluids 200 (2020) 104436, doi:10.1016/j.compfluid.2020.104436. https://www.sciencedirect.com/science/article/pii/S0045793020300128
[114] A. Corsini, C. Iossa, F. Rispoli, T. Tezduyar, A DRD finite element formulation for computing turbulent reacting flows in gas turbine combustors, Comput. Mech. 46 (2010) 159–167.
[115] A. Giusti, E. Mastorakos, Turbulent combustion modelling and experiments: Recent trends and developments, Flow Turbul. Combust. 103 (4) (2019) 847–869, doi:10.1007/s10494-019-00072-6.
[116] A. Wehrfritz, O. Kaario, V. Vuorinen, B. Somers, Large eddy simulation of n-dodecane spray flames using flamelet generated manifolds, Combust. Flame 167 (2016) 113–131.
[117] A. Broatch, M. Carreres, J. García-Tíscar, M. Belmar-Gil, Spectral analysis and modelling of the spray liquid injection in a lean direct injection (LDI) gas turbine combustor through Eulerian-Lagrangian large eddy simulations, Aerosp. Sci. Technol. 118 (2021) 106992.
[118] R. Mercier, V. Moureau, D. Veynante, B. Fiorina, LES of turbulent combustion: On the consistency between flame and flow filter scales, Proc. Combust. Inst. 35 (2) (2015) 1359–1366, doi:10.1016/j.proci.2014.05.149. https://www.sciencedirect.com/science/article/pii/S1540748914001527
[119] J. Donea, A Taylor–Galerkin method for convective transport problems, Int. J. Numer.MethodsEng. 20 (1) (1984) 101–119.
[120] L. Quartapelle, V. Selmin, High-order Taylor–Galerkin methods for nonlinear multidimensional problems, Finite Ele. Fluids 76 (90) (1993) 46.
[121] O. Colin, M. Rudgyard, Development of high-order Taylor–Galerkin schemes for LES, J. Comput. Phys. 162 (2) (2000) 338–371.
[122] B. Rochette, F. Collin-Bastiani, L. Gicquel, O. Vermorel, D. Veynante, T. Poinsot, Influence of chemical schemes, numerical method and dynamic turbulent combustion modeling on LES of premixed turbulent flames, Combust. Flame 191 (2018) 417–430.
[123] R. Sevilla, O. Hassan, K. Morgan, An analysis of the performance of a high-order stabilised finite element method for simulating compressible flows, Comput. Methods Appl. Mech.Eng. 253 (2013) 15–27, doi:10.1016/j.cma.2012.09.001. https://www.sciencedirect.com/science/article/pii/S0045782512002757
[124] J. Carpio, J. Prieto, An anisotropic, fully adaptive algorithm for the solution of convection-dominated equations with semi-Lagrangian schemes, Comput. Methods Appl. Mech.Eng. 273 (2014) 77–99, doi:10.1016/j.cma.2014.01.025. https://www.sciencedirect.com/science/article/pii/S0045782514000462
[125] A. Jaeschke, M. Möller, High-Order Isogeometric Methods for Compressible Flows, Springer International Publishing, Cham, 2020, pp. 21–29.
[126] J.-B. Chapelier, G. Lodato, A spectral-element dynamic model for the Large-Eddy simulation of turbulent flows, J. Comput. Phys. 321 (2016) 279–302, doi:10.1016/j.jcp.2016.05.051. https://www.sciencedirect.com/science/article/pii/S0021999116301966
[127] P. Fischer, S. Kerkemeier, M. Min, Y.-H. Lan, M. Phillips, T. Rathnayake, E. Merzari, A. Tomboulides, A. Karakus, N. Chalmers, T. Warburton, NekRS, a GPU-accelerated spectral element Navier-Stokes solver, 2021, 10.48550/ARXIV.2104.05829
[128] H.T. Huynh, A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods, 10.2514/6.2007–4079
[129] J. Bull, A. Jameson, High-order flux reconstruction schemes for LES on tetrahedral meshes, Progress in Hybrid RANS-LES Modelling: Volume 130 of the series Notes on Numerical Fluid Mechanics and Multidisciplinary Design, 2014, doi:10.1007/978-3-319-15141-0_5. http://link.springer.com/chapter/10.1007/978-3-319-15141-0_5
[130] K. Bando, M. Sekachev, M. Ihme, Comparison of algorithms for simulating multi-component reacting flows using high-order discontinuous Galerkin methods, https://arc.aiaa.org/doi/pdf/10.2514/6.2020-175110.2514/6.2020-1751
[131] Y. Lv, M. Ihme, Discontinuous Galerkin method for multicomponent chemically reacting flows and combustion, J. Comput. Phys. 270 (2014) 105–137, doi:10.1016/j.jcp.2014.03.029. https://www.sciencedirect.com/science/article/pii/S0021999114002101
[132] Z. Ren, C. Xu, T. Lu, M.A. Singer, Dynamic adaptive chemistry with operator splitting schemes for reactive flow simulations, J. Comput. Phys. 263 (2014) 19–36.
[133] Y. Wu, Y. Gao, T. Lu, A second-order dynamic adaptive hybrid scheme for time-integration of stiff chemistry, Combust. Flame 228 (2021) 193–201, doi:10.1016/j.combustflame.2021.01.045. https://www.sciencedirect.com/science/article/pii/S001021802100064X
[134] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (3) (1968) 506–517.
[135] S.B. Pope, Z. Ren, Efficient implementation of chemistry in computational combustion, Flow Turbul. Combust. 82 (4) (2009) 437–453.
[136] Z. Ren, S.B. Pope, Second-order splitting schemes for a class of reactive systems, J. Comput. Phys. 227 (17) (2008) 8165–8176.
[137] E. Motheau, J. Abraham, A high-order numerical algorithm for DNS of low-Mach-number reactive

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

flows with detailed chemistry and quasi-spectral accuracy, J. Comput. Phys. 313 (2016) 430–454.
[138] A. Dutt, L. Greengard, V. Rokhlin, Spectral deferred correction methods for ordinary differential equations, BIT Numer. Math. 40 (2) (2000) 241–266.
[139] M.L. Minion, Semi-implicit spectral deferred correction methods for ordinary differential equations, Commun. Math. Sci. 1 (3) (2003) 471–500.
[140] A. Bourlioux, A.T. Layton, M.L. Minion, High-order multi-implicit spectral deferred correction methods for problems of reactive flow, J. Comput. Phys. 189 (2) (2003) 651–675.
[141] W.E. Pazner, A. Nonaka, J.B. Bell, M.S. Day, M.L. Minion, A high-order spectral deferred correction strategy for low Mach number flow with complex chemistry, Combust. Theory Model. 20 (3) (2016) 521–547.
[142] U.M. Ascher, S.J. Ruuth, B.T. Wetton, Implicit-explicit methods for time-dependent partial differential equations, SIAM J. Numer. Anal. 32 (3) (1995) 797–823.
[143] A. Kanevsky, M.H. Carpenter, D. Gottlieb, J.S. Hesthaven, Application of implicit–explicit high order Runge–Kutta methods to discontinuous-Galerkin schemes, J. Comput. Phys. 225 (2) (2007) 1753–1781.
[144] P. Boivin, A.L. Sánchez, F.A. Williams, Four-step and three-step systematically reduced chemistry for wide-range H2-air combustion problems, Combust. Flame 160 (1) (2013) 76–82, doi:10.1016/j.combustflame.2012.09.014. https://www.sciencedirect.com/science/article/pii/S0010218012002751
[145] D. Fernández-Galisteo, A. Weiss, A.L. Sánchez, F.A. Williams, A one-step reduced mechanism for near-limit hydrogen combustion with general stoichiometry, Combust. Flame 208 (2019) 1–4, doi:10.1016/j.combustflame.2019.06.018. https://www.sciencedirect.com/science/article/pii/S0010218019302779
[146] M. Cailler, N. Darabiha, B. Fiorina, Development of a virtual optimized chemistry method. application to hydrocarbon/air combustion, Combust. Flame 211 (2020) 281–302, doi:10.1016/j.combustflame.2019.09.013. https://www.sciencedirect.com/science/article/pii/S0010218019304274
[147] L. Liang, J.G. Stevens, J.T. Farrell, A dynamic adaptive chemistry scheme for reactive flow computations, Proc. Combust. Inst. 32 (1) (2009) 527–534.
[148] H. Yang, Z. Ren, T. Lu, G.M. Goldin, Dynamic adaptive chemistry for turbulent flame simulations, Combust. Theory Model. 17 (1) (2013) 167–183.
[149] S. Yang, R. Ranjan, V. Yang, S. Menon, W. Sun, Parallel on-the-fly adaptive kinetics in direct numerical simulation of turbulent premixed flame, Proc. Combust. Inst. 36 (2) (2017) 2025–2032.
[150] A. Imren, D. Haworth, On the merits of extrapolation-based stiff ode solvers for combustion CFD, Combust. Flame 174 (2016) 1–15, doi:10.1016/j.combustflame.2016.09.018. https://www.sciencedirect.com/science/article/pii/S001021801630267X
[151] K.E. Niemeyer, N.J. Curtis, C.-J. Sung, pyJac: analytical Jacobian generator for chemical kinetics, Comput. Phys. Commun. 215 (2017) 188–203, doi:10.1016/j.cpc.2017.02.004. https://www.sciencedirect.com/science/article/pii/S0010465517300462
[152] T. Lu, C.K. Law, A directed relation graph method for mechanism reduction, Proc. Combust. Inst. 30 (1) (2005) 1333–1341.
[153] P. Pepiot-Desjardins, H. Pitsch, An efficient error-propagation-based reduction method for large chemical kinetic mechanisms, Combust. Flame 154 (1–2) (2008) 67–81.
[154] W. Sun, Z. Chen, X. Gou, Y. Ju, A path flux analysis method for the reduction of detailed chemical kinetic mechanisms, Combust. Flame 157 (7) (2010) 1298–1307.
[155] M. Jangi, X.-S. Bai, Multidimensional chemistry coordinate mapping approach for combustion modelling with finite-rate chemistry, Combust. Theory Model. 16 (6) (2012) 1109–1132.
[156] W. Sun, X. Gou, H.A. El-Asrag, Z. Chen, Y. Ju, Multi-timescale and correlated dynamic adaptive chemistry modeling of ignition and flame propagation using a real jet fuel surrogate model, Combust. Flame 162 (4) (2015) 1530–1539.
[157] J. Muela, R. Borrell, J. Ventosa-Molina, L. Jofre, O. Lehmkuhl, C. Pérez-Segarra, A dynamic load balancing method for the evaluation of chemical reaction rates in parallel combustion simulations, Comput. Fluids 190 (2019) 308–321, doi:10.1016/j.compfluid.2019.06.018. https://www.sciencedirect.com/science/article/pii/S0045793019301914
[158] B. Tekgül, P. Peltonen, H. Kahila, O. Kaario, V. Vuorinen, DLBFoam: an open-source dynamic load balancing model for fast reacting flow simulations in OpenFOAM, Comput. Phys. Commun. 267 (2021) 108073, doi:10.1016/j.cpc.2021.108073.
[159] J. An, G. He, F. Qin, X. Wei, B. Liu, Dynamic adaptive chemistry with mechanisms tabulation and in situ adaptive tabulation (ISAT) for computationally efficient modeling of turbulent combustion, Combust. Flame 206 (2019) 467–475, doi:10.1016/j.combustflame.2019.05.016. https://www.sciencedirect.com/science/article/pii/S0010218019302299
[160] J. van Oijen, A. Donini, R. Bastiaans, J. ten Thije Boonkkamp, L. de Goey, State-of-the-art in premixed combustion modeling using flamelet generated manifolds, Prog. Energy Combust. Sci. 57 (2016) 30–74.
[161] B. Fiorina, O. Gicquel, L. Vervisch, S. Carpentier, N. Darabiha, Premixed turbulent combustion modeling using tabulated detailed chemistry and PDF, Proc. Combust. Inst. 30 (1) (2005) 867–874, doi:10.1016/j.proci.2004.08.062. https://www.sciencedirect.com/science/article/pii/S0082078404001158
[162] C.D. Pierce, P. Moin, Progress-variable approach for Large-Eddy simulation of non-premixed turbulent combustion, J. Fluid Mech. 504 (2004) 73–97, doi:10.1017/S0022112004008213.
[163] J. van Oijen, A. Donini, R. Bastiaans, J. ten Thije Boonkkamp, L. de Goey, State-of-the-art in premixed combustion modeling using flamelet generated manifolds, Prog. Energy Combust. Sci. 57 (2016) 30–74, doi:10.1016/j.pecs.2016.07.001. https://www.sciencedirect.com/science/article/pii/S0360128515300137
[164] S. Popp, S. Weise, C. Hasse, A novel approach

for efficient storage and retrieval of tabulated chemistry in reactive flow simulations, in: E. Di Napoli, M.-A. Hermanns, H. Iliev, A. Lintermann, A. Peyser (Eds.), High-Performance Scientific Computing, Springer International Publishing, Cham, 2017, pp. 82–95.

[165] T. Honzawa, R. Kai, K. Hori, M. Seino, T. Nishiie, R. Kurose, Experimental and numerical study of water sprayed turbulent combustion: Proposal of a neural network modeling for five-dimensional flamelet approach, Energy AI 5 (2021) 100076, doi:10.1016/j.egyai.2021.100076. https://www.sciencedirect.com/science/article/pii/S2666546821000306

[166] E. Illana, D. Mira, A. Mura, An extended flame index partitioning for partially premixed combustion, Combust. Theory Model. 25 (1) (2021) 121–157.

[167] C.E. Lacey, A.G. Novoselov, M.E. Mueller, In-situ adaptive manifolds: enabling computationally efficient simulations of complex turbulent reacting flows, Proc. Combust. Inst. 38 (2) (2021) 2673–2680, doi:10.1016/j.proci.2020.06.207. https://www.sciencedirect.com/science/article/pii/S1540748920302984

[168] P. Kundu, J. Scroggins, M.M. Ameen, A novel in situ flamelet tabulation methodology for the representative interactive flamelet model, Combust. Sci. Technol. 192 (1) (2020) 1–25, doi:10.1080/00102202.2018.1539715.

[169] R. Ranade, G. Li, S. Li, T. Echekki, An efficient machine-learning approach for PDF tabulation in turbulent combustion closure, Combust. Sci. Technol. 193 (7) (2021) 1258–1277, doi:10.1080/00102202.2019.1686702.

[170] T. Ding, T. Readshaw, S. Rigopoulos, W. Jones, Machine learning tabulation of thermochemistry in turbulent combustion: an approach based on hybrid flamelet/random data and multiple multilayer perceptrons, Combust. Flame 231 (2021) 111493, doi:10.1016/j.combustflame.2021.111493. https://www.sciencedirect.com/science/article/pii/S0010218021002364

[171] O. Owoyele, P. Kundu, M.M. Ameen, T. Echekki, S. Som, Application of deep artificial neural networks to multi-dimensional flamelet libraries and spray flames, Int. J. Engine Res. 21 (1) (2020) 151–168, doi:10.1177/1468087419837770.

[172] M. Ihme, C. Schmitt, H. Pitsch, Optimal artificial neural networks and tabulation methods for chemistry representation in LES of a bluff-body swirl-stabilized flame, Proc. Combust. Inst. 32 (1) (2009) 1527–1535, doi:10.1016/j.proci.2008.06.100. https://www.sciencedirect.com/science/article/pii/S1540748908001132

[173] M. Hansinger, Y. Ge, M. Pfitzner, Deep residual networks for flamelet/progress variable tabulation with application to a piloted flame with inhomogeneous inlet, Combust. Sci. Technol. 0 (0) (2020) 1–27, doi:10.1080/00102202.2020.1822826.

[174] A. Seltz, P. Domingo, L. Vervisch, Z.M. Nikolaou, Direct mapping from LES resolved scales to filtered-flame generated manifolds using convolutional neural networks, Combust. Flame 210 (2019) 71–82, doi:10.1016/j.combustflame.2019.08.014. https://www.sciencedirect.com/science/article/pii/S0010218019303773

[175] B. Fiorina, R. Vicquelin, P. Auzillon, N. Darabiha, O. Gicquel, D. Veynante, A filtered tabulated chemistry model for LES of premixed combustion, Combust. Flame 157 (3) (2010) 465–475, doi:10.1016/j.combustflame.2009.09.015. https://www.sciencedirect.com/science/article/pii/S0010218009002739

[176] B. Cuenot, F. Shum-Kivan, S. Blanchard, The thickened flame approach for non-premixed combustion: principles and implications for turbulent combustion modeling, Combust. Flame (2021) 111702, doi:10.1016/j.combustflame.2021.111702. https://www.sciencedirect.com/science/article/pii/S0010218021004454

[177] B. MAGNUSSEN, On the structure of turbulence and a generalized eddy dissipation concept for chemical reaction in turbulent flow, https://arc.aiaa.org/doi/pdf/10.2514/6.1981-4210.2514/6.1981-42

[178] A. Péquin, S. Iavarone, R. Malpica Galassi, A. Parente, The partially stirred reactor model for combustion closure in large eddy simulations: physical principles, sub-models for the cell reacting fraction, and open challenges, Phys. Fluids 34 (5) (2022) 055122, doi:10.1063/5.0090970.

[179] W. Jones, V. Prasad, Large eddy simulation of the Sandia Flame series (D-F) using the Eulerian stochastic field method, Combust. Flame 157 (9) (2010) 1621–1636, doi:10.1016/j.combustflame.2010.05.010. https://www.sciencedirect.com/science/article/pii/S0010218010001525

[180] N. Patel, S. Menon, Simulation of spray-turbulence-flame interactions in a lean direct injection combustor, Combust. Flame 153 (1) (2008) 228–257, doi:10.1016/j.combustflame.2007.09.011. https://www.sciencedirect.com/science/article/pii/S0010218007002647

[181] B. Cuenot, The Flamelet Model for Non-Premixed Combustion, Springer Netherlands, Dordrecht, 2011, pp. 43–61.

[182] B.A. Perry, M.E. Mueller, A.R. Masri, A two mixture fraction flamelet model for large eddy simulation of turbulent flames with inhomogeneous inlets, Proc. Combust. Inst. 36 (2) (2017) 1767–1775, doi:10.1016/j.proci.2016.07.029. https://www.sciencedirect.com/science/article/pii/S1540748916302851

[183] Y. Hu, R. Kurose, Nonpremixed and premixed flamelets LES of partially premixed spray flames using a two-phase transport equation of progress variable, Combust. Flame 188 (2018) 227–242, doi:10.1016/j.combustflame.2017.10.004. https://www.sciencedirect.com/science/article/pii/S0010218017303930

[184] E. Knudsen, H. Pitsch, Capabilities and limitations of multi-regime flamelet combustion models, Combust. Flame 159 (1) (2012) 242–264, doi:10.1016/j.combustflame.2011.05.025. https://www.sciencedirect.com/science/article/pii/S0010218011001982

[185] A. Kalbhor, J. van Oijen, An assessment of the sectional soot model and FGM tabulated chemistry coupling in laminar flame simulations, Combust. Flame 229 (2021) 111381, doi:10.1016/j.combustflame.2021.02.027. https://www.sciencedirect.com/science/article/pii/S0010218021000961

[186] A. Wick, A. Attili, F. Bisetti, H. Pitsch, DNS-driven

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

analysis of the flamelet/progress variable model assumptions on soot inception, growth, and oxidation in turbulent flames, Combust. Flame 214 (2020) 437–449.
[187] S. Gövert, D. Mira, J. Kok, M. Vázquez, G. Houzeaux, Turbulent combustion modelling of a confined premixed jet flame including heat loss effects using tabulated chemistry, Appl. Energy 156 (2015) 804–815, doi:10.1016/j.apenergy.2015.06.031. https://www.sciencedirect.com/science/article/pii/S0306261915007898
[188] P. Breda, U.M. C. Yu, M. Pfitzner, Validation of an Eulerian stochastic fields solver coupled with reaction-diffusion manifolds on LES of methane/air non-premixed flames, Flow Turbul. Combust. 107 (2021) 441–477.
[189] A. Avdić, G. Kuenne, F. di Mare, J. Janicka, LES combustion modeling using the Eulerian stochastic field method coupled with tabulated chemistry, Combust. Flame 175 (2017) 201–219, doi:10.1016/j.combustflame.2016.06.015. Special Issue in Honor of Norbert Peters
[190] J.H. Chen, Petascale direct numerical simulation of turbulent combustion-fundamental insights towards predictive models, Proc. Combust. Inst. 33 (1) (2011) 99–123, doi:10.1016/j.proci.2010.09.012. https://www.sciencedirect.com/science/article/pii/S1540748910003962
[191] K.E. Niemeyer, C.-J. Sung, Accelerating moderately stiff chemical kinetics in reactive-flow simulations using GPUs, J. Comput. Phys. 256 (2014) 854–871, doi:10.1016/j.jcp.2013.09.025. https://www.sciencedirect.com/science/article/pii/S0021999113006396
[192] R. Malpica Galassi, P.P. Ciottoli, M. Valorani, H.G. Im, An adaptive time-integration scheme for stiff chemistry based on computational singular perturbation and artificial neural networks, J. Comput. Phys. 451 (2022) 110875, doi:10.1016/j.jcp.2021.110875. https://www.sciencedirect.com/science/article/pii/S0021999121007701
[193] S. Lapointe, S. Mondal, R.A. Whitesides, Data-driven selection of stiff chemistry ode solver in operator-splitting schemes, Combust. Flame 220 (2020) 133–143, doi:10.1016/j.combustflame.2020.06.033. https://www.sciencedirect.com/science/article/pii/S0010218020302510
[194] K.E. Niemeyer, N.J. Curtis, C.-J. Sung, pyJac: analytical Jacobian generator for chemical kinetics, Comput. Phys. Commun. 215 (2017) 188–203, doi:10.1016/j.cpc.2017.02.004.
[195] M.J. McNenly, R.A. Whitesides, D.L. Flowers, Faster solvers for large kinetic mechanisms using adaptive preconditioners, Proc. Combust. Inst. 35 (1) (2015) 581–587, doi:10.1016/j.proci.2014.05.113. https://www.sciencedirect.com/science/article/pii/S1540748914001163
[196] F. Perini, E. Galligani, R.D. Reitz, A study of direct and Krylov iterative sparse solver techniques to approach linear scaling of the integration of chemical kinetics with detailed combustion mechanisms, Combust. Flame 161 (5) (2014) 1180–1195, doi:10.1016/j.combustflame.2013.11.017. https://www.sciencedirect.com/science/article/pii/S0010218013004306
[197] Z. Liu, J.-L. Consalvi, W. Kong, An exponential integrator with Schur-Krylov approximation to accelerate combustion chemistry computation, Combust. Flame 203 (2019) 180–189, doi:10.1016/j.combustflame.2019.01.031. https://www.sciencedirect.com/science/article/pii/S0010218019300495
[198] M. Valorani, S. Paolucci, The g-scheme: a framework for multi-scale adaptive model reduction, J. Comput. Phys. 228 (13) (2009) 4665–4701, doi:10.1016/j.jcp.2009.03.011. https://www.sciencedirect.com/science/article/pii/S0021999109001302
[199] M. Valorani, P. Ciottoli, R. Galassi, S. Paolucci, T. Grenga, E. Martelli, Enhancements of the g-scheme framework, Flow Turbul. Combust. 101 (2018) 1023–1033, doi:10.1007/s10494-018-9942-2.
[200] Y. Morii, H. Terashima, M. Koshi, T. Shimizu, E. Shima, ERENA: a fast and robust Jacobian-free integration method for ordinary differential equations of chemical kinetics, J. Comput. Phys. 322 (2016) 547–558, doi:10.1016/j.jcp.2016.06.022. https://www.sciencedirect.com/science/article/pii/S0021999116302479
[201] C. Stone, R. Davis, Techniques for solving stiff chemical kinetics on GPUs, https://arc.aiaa.org/doi/pdf/10.2514/6.2013-36910.2514/6.2013-369
[202] Y. Shi, W.H. Green, H.-W. Wong, O.O. Oluwole, Accelerating multi-dimensional combustion simulations using GPU and hybrid explicit/implicit ODE integration, Combust. Flame 159 (7) (2012) 2388–2397, doi:10.1016/j.combustflame.2012.02.016. https://www.sciencedirect.com/science/article/pii/S001021801200065X
[203] C.J. Balos, D.J. Gardner, C.S. Woodward, D.R. Reynolds, Enabling GPU accelerated computing in the sundials time integration library, Parallel Comput. 108 (2021) 102836, doi:10.1016/j.parco.2021.102836. https://www.sciencedirect.com/science/article/pii/S0167819121000831
[204] J.C. Beale, R.D. Reitz, Modeling spray atomization with the Kelvin-Helmholtz/Rayleigh-Taylor hybrid model, Atomization Sprays 9 (6) (1999).
[205] P.J. O'Rourke, A.A. Amsden, The TAB Method for Numerical Calculation of Spray Droplet breakup, Technical Report, Los Alamos National Lab., NM (USA), 1987.
[206] J. Shinjo, A. Umemura, Simulation of liquid jet primary breakup: Dynamics of ligament and droplet formation, Int. J. Multiphase Flow 36 (7) (2010) 513–532.
[207] T. Ménard, S. Tanguy, A. Berlemont, Coupling level set/VOF/ghost fluid methods: validation and application to 3D simulation of the primary break-up of a liquid jet, Int. J. Multiphase Flow 33 (5) (2007) 510–524, doi:10.1016/j.ijmultiphaseflow.2006.11.001. https://www.sciencedirect.com/science/article/pii/S0301932206001832
[208] A. Zandian, W.A. Sirignano, F. Hussain, Understanding liquid-jet atomization cascades via vortex dynamics, J. Fluid Mech. 843 (2018) 293–354.
[209] O. Desjardins, J. McCaslin, M. Owkes, P. Brady, Direct numerical and Large-Eddy simulation of primary atomization in complex geometries, Atomization Sprays 23 (11) (2013).

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

[210] X. Jiang, G. Siamas, K. Jagus, T. Karayiannis, Physical modelling and advanced simulations of gas-liquid two-phase jet flows in atomization and sprays, Prog. Energy Combust. Sci. 36 (2) (2010) 131–167, doi:10.1016/j.pecs.2009.09.002. https://www.sciencedirect.com/science/article/pii/S0360128509000458
[211] C. Shao, K. Luo, Y. Yang, J. Fan, Detailed numerical simulation of swirling primary atomization using a mass conservative level set method, Int. J. Multiphase Flow 89 (2017) 57–68, doi:10.1016/j.ijmultiphaseflow.2016.10.010. https://www.sciencedirect.com/science/article/pii/S0301932216306206
[212] J. Shinjo, A. Umemura, Simulation of liquid jet primary breakup: dynamics of ligament and droplet formation, Int. J. Multiphase Flow 36 (7) (2010) 513–532, doi:10.1016/j.ijmultiphaseflow.2010.03.008. https://www.sciencedirect.com/science/article/pii/S0301932210000637
[213] C. Hirt, B. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1) (1981) 201–225, doi:10.1016/0021-9991(81)90145-5. https://www.sciencedirect.com/science/article/pii/0021999181901455
[214] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase flow ii, J. Comput. Phys. 225 (1) (2007) 785–807.
[215] N. Balcázar, L. Jofre, O. Lehmkuhl, J. Castro, J. Rigola, A finite-volume/level-set method for simulating two-phase flows on unstructured grids, Int. J. Multiphase Flow 64 (2014) 55–72.
[216] S. Mirjalili, A. Mani, Consistent, energy-conserving momentum transport for simulations of two-phase flows using the phase field equations, J. Comput. Phys. 426 (2021) 109918, doi:10.1016/j.jcp.2020.109918. https://www.sciencedirect.com/science/article/pii/S0021999120306926
[217] O. Desjardins, G. Blanquart, G. Balarac, H. Pitsch, High order conservative finite difference scheme for variable density low Mach number turbulent flows, J. Comput. Phys. 227 (15) (2008) 7125–7159.
[218] N. Kumar Singh, B. Premachandran, A coupled level set and volume of fluid method on unstructured grids for the direct numerical simulations of two-phase flows including phase change, Int. J. Heat Mass Transf. 122 (2018) 182–203, doi:10.1016/j.ijheatmasstransfer.2018.01.091. https://www.sciencedirect.com/science/article/pii/S0017931017341194
[219] Y. Atmani, F. Pecquery, M. Cailler, V. Moureau, G. Lartigue, R. Mercier, R. Janodet, G. Sahut, G. Balarac, Consistent scalar transport with front capturing methods: application to two-phase heat transfer, in: ICLASS 2021, 15th Triennial International Conference on Liquid Atomization and Spray Systems, 2021.
[220] Y. Ling, S. Zaleski, R. Scardovelli, Multiscale simulation of atomization with small droplets represented by a Lagrangian point-particle model, Int. J. Multiphase Flow 76 (2015) 122–143.
[221] M. Herrmann, A parallel Eulerian interface tracking/Lagrangian point particle multi-scale coupling procedure, J. Comput. Phys. 229 (3) (2010) 745–759.
[222] C. Guillamon, R. Janodet, L. Voivenel, R. Mercier, V. Moureau, Building Lagrangian injectors from resolved primary atomization simulations. application to jet in crossflow fuel injection, in: ICLASS 2021, 15th Triennial International Conference on Liquid Atomization and Spray Systems, 2021.
[223] A. Asuri Mukundan, T. Ménard, J.C. Brändle de Motta, A. Berlemont, A 3D moment of fluid method for simulating complex turbulent multiphase flows, Comput. Fluids 198 (2020) 104364, doi:10.1016/j.compfluid.2019.104364. https://www.sciencedirect.com/science/article/pii/S0045793019303238
[224] A. Umemura, J. Shinjo, Detailed SGS atomization model and its implementation to two-phase flow LES, Combust. Flame 195 (2018) 232–252, doi:10.1016/j.combustflame.2018.01.026. Special Commemorative Issue: Professor Chung King (Ed) Law 70th Birthday
[225] J. Desantes, J. García-Oliver, J. Pastor, I. Olmeda, A. Pandal, B. Naud, LES Eulerian diffuse-interface modeling of fuel dense sprays near- and far-field, Int. J. Multiphase Flow 127 (2020) 103272, doi:10.1016/j.ijmultiphaseflow.2020.103272. https://www.sciencedirect.com/science/article/pii/S0301932219305245
[226] J. Wen, Y. Hu, T. Nishiie, J. Iino, A. Masri, R. Kurose, A flamelet LES of turbulent dense spray flame using a detailed high-resolution vof simulation of liquid fuel atomization, Combust. Flame (2021) 111742, doi:10.1016/j.combustflame.2021.111742. https://www.sciencedirect.com/science/article/pii/S0010218021004855
[227] A. Vallet, R. Borghi, Modélisation eulerienne de l'atomisation d'un jet liquide, Comptes Rendus de l'Académie des Sciences-Series IIB-Mechanics–Physics-Astronomy 327 (10) (1999) 1015–1020.
[228] F.-X. Demoulin, P.-A. Beau, G. Blokkeel, A. Mura, R. Borghi, A new model for turbulent flows with large density fluctuations: application to liquid atomization, Atomization Sprays 17 (4) (2007).
[229] S. Navarro-Martinez, Large eddy simulation of spray atomization with a probability density function method, Int. J. Multiphase Flow 63 (2014) 11–22.
[230] G. Agbaglah, S. Delaux, D. Fuster, J. Hoepffner, C. Josserand, S. Popinet, P. Ray, R. Scardovelli, S. Zaleski, Parallel simulation of multiphase flows using octree adaptivity and the volume-of-fluid method, Comptes Rendus Mécanique 339 (2) (2011) 194–207, doi:10.1016/j.crme.2010.12.006. High Performance Computing
[231] Y. Li, B. Zhou, X. Hu, A two-grid method for level-set based topology optimization with GPU-acceleration, J. Comput. Appl. Math. 389 (2021) 113336, doi:10.1016/j.cam.2020.113336. https://www.sciencedirect.com/science/article/pii/S0377042720306270
[232] D. Fuster, A. Bagué, T. Boeck, L. Le Moyne, A. Leboissetier, S. Popinet, P. Ray, R. Scardovelli, S. Zaleski, Simulation of primary atomization with an octree adaptive mesh refinement and VOF method, Int. J. Multiphase Flow 35 (6) (2009) 550–565.
[233] J.K. Dukowicz, A particle-fluid numerical model

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222

for liquid sprays, J. Comput. Phys. 35 (2) (1980) 229–253.
[234] P. Senecal, D.P. Schmidt, I. Nouar, C.J. Rutland, R.D. Reitz, M. Corradini, Modeling high-speed viscous liquid sheet atomization, Int. J. Multiphase Flow 25 (6–7) (1999) 1073–1097.
[235] P.N. Nordin, Complex Chemistry Modeling of Diesel Spray Combustion, vol. 18, Chalmers University of Technology Sweden, 2001.
[236] T. Su, M. Patterson, R.D. Reitz, P. Farrell, Experimental and numerical studies of high pressure multiple injection sprays, SAE Trans. (1996) 1281–1292.
[237] M. Gorokhovski, S. Oruganti, Stochastic models for the droplet motion and evaporation in underresolved turbulent flows at a large Reynolds number, J. Fluid Mech. 932 (2022) A18, doi:10.1017/jfm.2021.916.
[238] A. Both, D. Mira, O. Lehmkuhl, Evaporation of volatile droplets subjected to flame-like conditions, Int. J. Heat Mass Transf. 187 (2022) 122521, doi:10.1016/j.ijheatmasstransfer.2022.122521. https://www.sciencedirect.com/science/article/pii/S0017931022000035
[239] Y. Hu, H. Olguin, E. Gutheil, A spray flamelet/progress variable approach combined with a transported joint pdf model for turbulent spray flames, Combust. Theory Model. 21 (3) (2017) 575–602.
[240] D. Noh, S. Gallot-Lavallée, W.P. Jones, S. Navarro-Martinez, Comparison of droplet evaporation models for a turbulent, non-swirling jet flame with a polydisperse droplet distribution, Combust. Flame 194 (2018) 135–151, doi:10.1016/j.combustflame.2018.04.018. https://www.sciencedirect.com/science/article/pii/S0010218018301706
[241] E.J. Pérez-Sánchez, J.M. Garcia-Oliver, R. Novella, J.M. Pastor, Understanding the diesel-like spray characteristics applying a flamelet-based combustion model and detailed large eddy simulations, Int. J. Engine Res. 21 (1) (2020) 134–150.
[242] A. Chatelier, B. Fiorina, V. Moureau, N. Bertier, Large eddy simulation of a turbulent spray jet flame using filtered tabulated chemistry, J. Combust. 2020 (2020).
[243] J. Wirtz, B. Cuenot, E. Riber, Numerical study of a polydisperse spray counterflow diffusion flame, Proc. Combust. Inst. 38 (2) (2021) 3175–3182, doi:10.1016/j.proci.2020.05.042. https://www.sciencedirect.com/science/article/pii/S1540748920300900
[244] R. Löhner, J. Ambrosiano, A vectorized particle tracer for unstructured grids, J. Comput. Phys. 91 (1) (1990) 22–31, doi:10.1016/0021-9991(90)90002-I. https://www.sciencedirect.com/science/article/pii/002199919090002I
[245] R. Pankajakshan, B.J. Mitchell, L.K. Taylor, Simulation of unsteady two-phase flows using a parallel Eulerian-Lagrangian approach, Comput. Fluids 41 (1) (2011) 20–26, doi:10.1016/j.compfluid.2010.09.020. Implicit Solutions of Navier-Stokes Equations. Special Issue Dedicated to Drs. W.R. Briley and H. McDonald
[246] H. Sitaraman, R. Grout, Balancing conflicting requirements for grid and particle decomposition in continuum-Lagrangian solvers, Parallel Comput. 52 (2016) 1–21, doi:10.1016/j.parco.2015.10.010. https://www.sciencedirect.com/science/article/pii/S0167819115001428
[247] D. Darmana, N. Deen, J. Kuipers, Parallelization of an euler-lagrange model using mixed domain decomposition and a mirror domain technique: Application to dispersed gas-liquid two-phase flow, J. Comput. Phys. 220 (1) (2006) 216–248, doi:10.1016/j.jcp.2006.05.011. https://www.sciencedirect.com/science/article/pii/S0021999106002294
[248] D. Buaria, P. Yeung, A highly scalable particle tracking algorithm using partitioned global address space (PGAS) programming for extreme-scale turbulence simulations, Comput. Phys. Commun. 221 (2017) 246–258, doi:10.1016/j.cpc.2017.08.022. https://www.sciencedirect.com/science/article/pii/S0010465517302680
[249] Y. Dufresne, V. Moureau, G. Lartigue, O. Simonin, A massively parallel CFD/DEM approach for reactive gas-solid flows in complex geometries using unstructured meshes, Comput. Fluids 198 (2020) 104402, doi:10.1016/j.compfluid.2019.104402. https://www.sciencedirect.com/science/article/pii/S0045793019303603
[250] G. Houzeaux, M. Garcia, J.C. Cajas, A. Artigues, E. Olivares, J. Labarta, M. Vázquez, Dynamic load balance applied to particle transport in fluids, Int. J. Comput. FluidDyn. 30 (6) (2016) 408–418, doi:10.1080/10618562.2016.1227070.
[251] B. Wang, I. Wald, N. Morrical, W. Usher, L. Mu, K. Thompson, R. Hughes, A GPU-accelerated particle tracking method for Eulerian-Lagrangian simulations using hardware ray tracing cores, Comput. Phys. Commun. 271 (2022) 108221, doi:10.1016/j.cpc.2021.108221. https://www.sciencedirect.com/science/article/pii/S0010465521003337
[252] J. Sweet, D.H. Richter, D. Thain, GPU acceleration of Eulerian-Lagrangian particle-laden turbulent flow simulations, Int. J. Multiphase Flow 99 (2018) 437–445, doi:10.1016/j.ijmultiphaseflow.2017.11.010. https://www.sciencedirect.com/science/article/pii/S0301932217306869

Please cite this article as: D. Mira, E.J. Pérez-Sánchez, R. Borrell et al., HPC-enabling technologies for high-fidelity combustion simulations, Proceedings of the Combustion Institute, https://doi.org/10.1016/j.proci.2022.07.222