Direct numerical simulations of turbulent reacting flows with shock waves and stiff chemistry using many-core/GPU acceleration

S. Desai, Y.J. Kim, W. Song, M.B. Luong, F.E. Hernández Pérez, R. Sankaran, H.G. Im, Direct numerical simulations of turbulent reacting flows with shock waves and stiff chemistry using many-core/GPU acceleration, Computers & Fluids. 215 (2021) 104787. https://doi.org/10.1016/j.compfluid.2020.104787

Direct numerical simulations of turbulent reacting flows with shock waves and stiff chemistry using many-core/GPU acceleration

Swapnil Desai[a,], Yu Jeong Kim[c], Wonsik Song[c], Minh Bau Luong[c], Francisco E. Hernández Pérez[c], Ramanan Sankaran[a,b], Hong G. Im[c]

[a]*Bredesen Center for Interdisciplinary Research and Graduate Education, University of Tennessee, Knoxville, TN 37996-3394, USA*
[b]*Oak Ridge National Laboratory, Oak Ridge, TN 37831-6008, USA*
[c]*Clean Combustion Research Center, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia*

Abstract

Compressible reacting flows may display sharp spatial variation related to shocks, contact discontinuities or reactive zones embedded within relatively smooth regions. The presence of such phenomena emphasizes the relevance of shock-capturing schemes such as the weighted essentially non-oscillatory (WENO) scheme as an essential ingredient of the numerical solver. However, these schemes are complex and have more computational cost than the simple high-order compact or non-compact schemes. In this paper, we present the implementation of a seventh-order, minimally-dissipative mapped WENO (WENO7M) scheme in a newly developed direct numerical simulation (DNS) code called KAUST Adaptive Reactive Flows Solver (KARFS). In order to make efficient use of the computer resources and reduce the solution time, without compromising the resolution requirement, the WENO routines are accelerated via graphics processing unit (GPU) computation. The performance characteristics and scalability of the code are studied using different grid sizes and block decomposition. The performance portability of KARFS is demonstrated on a variety of architectures including NVIDIA Tesla P100 GPUs and NVIDIA Kepler K20X GPUs. In addition, the capability and potential of the newly implemented WENO7M scheme in KARFS to perform DNS of compressible flows is also demonstrated with model problems involving shocks, isotropic turbulence, detonations and flame propagation into a stratified mixture with complex chemical kinetics.

Keywords: Direct numerical simulation (DNS), WENO, GPU computing, High performance computing (HPC), Compressible turbulence

1. Introduction

The pursuit of improved efficiency and lower emissions in internal combustion engines has been an ongoing quest for several decades and continues to attract significant attention due to its economic and environmental impact. Recent emphasis on reducing the harmful nitrogen oxide and greenhouse gases emissions has renewed the focus on developing better internal combustion (IC) technologies that can derive the full economic potential of fossil fuels for transportation applications while lowering emissions. Several advanced combustion concepts have emerged which hold the promise of simultaneously improving the efficiency and lowering the emissions of internal combustion engines, while also allowing the utilization of renewable fuels [1–3]. The combustion concepts that are being pursued are quite different from the traditional spark ignition (SI) or compression ignition (CI) based engines. Ignition and combustion are expected to occur at conditions that are in the fringes of the traditional operating regimes of IC engines

Email address: swapnildesai1989@gmail.com (Swapnil Desai)

with limited prior understanding and experience. Furthermore, mixed modes of combustion occurs in which propagation of combustion fronts and uniform volumetric combustion exist simultaneously ([4–10] and references therein). Modeling and simulation play a very important role in improving the fundamental understanding of the combustion characteristics at such conditions, and in advancing the combustion technologies for practical applications.

There is currently only a limited fundamental understanding of the nature of combustion in an auto-ignitive mixture that is conducive to both spontaneous ignition and diffusion driven flame propagation. Several previous experimental and numerical studies have focused on characterizing the nature of combustion and delineating it into multiple regimes based on the propagation speed of the autoignition front and also based on the impact of turbulence and stratification on such a front [11–21]. Fundamental numerical simulations that can capture the nature and characteristics of autoignition in stratified reactive mixtures are key to improving and applying the appropriate combustion models in device-scale simulations.

Direct numerical simulation (DNS) of turbulent reacting flows is a first principle approach for turbulent combustion simulations that does not require closure models for turbulence physics and therefore free of underlying statistical and numerical assumptions [22]. DNS of reacting flows in canonical flow configurations such as triply periodic cube domains, quasi-one dimensional planar stabilized and fully three-dimensional spatially developing jet configurations have been used to study turbulence-combustion interaction in reactive mixtures [4, 5, 23]. Since DNS resolves all of the relevant flow scales up to the smallest dissipative scales, it becomes prohibitively expensive to simulate large scale flow motion up to device scales. However, smaller canonical geometries have been within the reach of DNS for several years now. Newer developments in high performance computing (HPC) have allowed an increase in the range of length scales simulated, providing more realism in the turbulence physics, along with simultaneously increasing the chemical complexity, allowing the simulation of fuels and intermediates that are better surrogates of transportation fuels.

One of the primary objectives of DNS is to capture the turbulent flow physics as accurately as possible without introducing numerical distortion that would cause the turbulence energy cascade to be incorrect. Low dissipation by employing higher order accuracy have become essential for the fidelity of simulations. Higher order centered finite difference schemes [24] along with higher order accurate Runge–Kutta time integration schemes [25] have been very successful in enabling DNS with higher solution accuracy while also allowing efficient and affordable computations of large problems. Since the central difference scheme lacks the numerical dissipation needed to ensure stability, it is also customary to use a filter operator, such as the tenth order accurate explicit filter by Kennedy and Carpenter [24]. Such a combination of higher order accurate central difference schemes along with explicit filtering has proven to be well suited and sufficient for DNS of low speed turbulent combustion when the combustion fronts are sufficiently resolved on the numerical grid.

The spatial profile of a propagating reaction front is determined by the transport rate of heat and radicals between the reaction zone and the upstream mixture. The thickness of a typical transport-driven flame front is determined by the molecular or turbulent diffusivity, in the case of laminar and turbulent flames, respectively. However, a spontaneous propagation front is not limited by the transport rates. The thickness of a spontaneous propagation front is determined primarily by the gradients in reactivity of the upstream mixture and the effects of transport rate are secondary. When a spontaneously propagating front encounters an adverse gradient in reactivity, it can turn into a sharp contact discontinuity between the unreacted and reacted mixtures, with steep gradients in density, temperature and composition. Such a contact discontinuity is distinct from a detonation, since there is no large pressure jump between the unburned and burned mixtures. The contact discontinuity cannot be stably resolved by the central difference schemes and requires the use of a numerical scheme capable of capturing discontinuities.

Weighted essentially non-oscillatory (WENO) schemes [26] are a class of numerical methods that have been shown to be capable of capturing discontinuities in compressible flows, while also being suitable for DNS. WENO schemes obtain a numerical approximation to the advective fluxes in the governing equation using a weighted sum of fluxes computed using multiple candidate stencils. The weights are determined by adapting to the presence of discontinuities such that the solution tends to approach high accuracy and low dissipation in smooth regions of the flow. WENO schemes are considerably more expensive than regular central difference schemes due to the need to evaluate the same flux functions from multiple candidate stencils [27]. However, the numerical method is suitable for implementation in finite difference codes using stencil operations very similar to the central difference schemes. Also, the usual advantages of DNS using finite difference schemes such as parallelism and scalability carries over to the WENO schemes as well. Apart from DNS, these advantages of WENO schemes have also been demonstrated in

implicit large eddy simulations (iLES) [28–30].

Success in the general application of computational fluid dynamics (CFD) and affiliated techniques such as DNS is largely dependent on achieving an optimal balance between multi-scale fidelity, multi-physics detail and the overall computational cost. In this regard, recent advances in computational architectures offer both potential and challenges. Dramatic speed-ups can be potentially achieved across platforms using state-of-the-art multi-core central processing units (CPUs) and graphics processing units (GPUs). The challenge, however, is actually attaining this potential through development of new parallel programming models that can be used to efficiently port various algorithms and the related code kernels to these new heterogeneous architectures. A major challenge in utilizing heterogeneous resources is the diversity of devices on different machines, which provide widely varying performance characteristics. A program or algorithm optimized for one architecture may not run as well on the next generation of processors or on a device from a different vendor. A program or algorithm optimized for GPU execution is often very different from one optimized for CPU execution. The relative performance of CPUs and GPUs also varies between machines. On one machine, a specific portion of a computation may run best on the CPU, while on another machine it may run best on the GPU. In some cases, it is best to balance the workload between the CPU and the GPU; in other cases, it may be best to execute an algorithm on a device where it runs more slowly but closer to where its output is needed, in order to avoid expensive data transfer operations. It is not uncommon for large application codes to have several different implementations, with each one optimized for a different architecture. This software development approach leads to a code maintainability issue: every new change to the code needs to be implemented in all versions of the code. Performance portability of a single code base, therefore, has become a critical issue: the parallel code needs to execute correctly and remain performant on machines with different architectures, operating systems, and software libraries.

Several groups have published their efforts for accelerating the flow solvers used for multi-component reacting flow calculations using GPUs. The most common approach is to select the most time consuming and compute-intensive kernels and offload them to the GPUs for accelerated solution [31]. In the case of combustion solvers, the chemical reaction kinetics was the predominant kernel that was offloaded to the GPUs. Spafford and co-workers [32] were the first to use this approach to offload the chemical kinetics evaluation to a GPU using the CUDA programming model and thereby accelerate a DNS solver for turbulent combustion. Their approach was to use grid-level parallelism for acceleration by computing the reaction kinetics across the large number of grid points in parallel on the GPU. In contrast, Shi et al. [33], utilized the parallelism available in the reaction network itself to simultaneously calculate all the reaction rates for a single kinetic system. They observed a considerable speedup for large chemical reaction mechanisms (>1000 species), but could not obtain any acceleration for smaller mechanisms (<100 species). The performance of implicit integration schemes for stiff chemical kinetics against the speedups obtained for explicit integration has also been measured by Stone et al. [34]. More recently, Niemeyer et al. [35] used fully explicit and stabilized explicit methods for accelerating chemical kinetics with low to moderate levels of stiffness. In these studies, it was found that the potential for acceleration of chemical kinetics through GPUs is large when explicit integration is feasible. But in the presence of stiffness, the performance on GPUs was highly susceptible to thread divergence due to varying levels of stiffness across the multiple states.

We introduced a new DNS solver - KARFS (KAUST Adaptive Reacting Flows Solver) for combustion calculations in a previous research article [36], showcasing its application in multidimensional turbulent reacting flow simulations. KARFS is a modern DNS code developed in C++ using modern parallel programming patterns, distributed memory parallelism through message passing interface (MPI) and portable on-node parallelism with the Kokkos C++ programming model [37, 38]. KARFS has performance-portable capabilities for multi- and many-core heterogeneous platforms, which is instrumental to meet emerging demands of exascale computing machines. The employed Kokkos framework allows KARFS to straighforwardly utilize the same source code on CPUs and GPUs while maintaining performance comparable to the code that is optimally tuned to either processing unit. We have taken the approach to use standard C++ programming practices for both on- and off-node parallelism. This has been the main motivation for choosing the Kokkos programming model, in which all kernel launches on a target device and all memory copies between memory spaces are asynchronous by default. With further development of Kokkos to allow multiple stream/task parallelism, it will be possible to schedule multiple streams of tasks to hide latencies. We believe that this will be the most portable programming model for future HPC application. Current choices for asynchronous execution or data management requires vendor specific extensions to the standard programming languages. Furthermore, they require compiler extensions through Pragmas that are not widely supported by all open source compilers. Other

available runtime systems cannot currently inter-operate with Kokkos-type programming abstractions. We do not find any of these approaches attractive and hence will track the task parallelism model being developed in Kokkos.

In this paper, the numerical methods available in KARFS were extended to include a seventh-order mapped WENO (WENO7M) scheme [39] for capturing discontinuities and a stiff ordinary differential equation (ODE) solver for dealing with stiff and complex chemistry. The computational loops and kernels for the WENO scheme are written using Kokkos parallel patterns so as to provide performance portability. Similarly, the linear algebra computations related to the stiff-ODE integrator are accelerated by making use of the MAGMA library [40], which was designed for hybrid computing architectures. In a recent study [41], it was demonstrated that the standard seventh-order WENO scheme developed by Jiang and Shu [42] is less dissipative compared to the corresponding standard fifth-order WENO scheme. Moreover, the standard seventh-order WENO scheme cannot exactly guarantee seventh-order accuracy since the nonlinear weights do not become optimal values near the critical points. Henrick et al. [39] suggested a mapping technique such that the formal order of accuracy is efficiently recovered near critical points. Subsequent simulations of pulsating one-dimensional detonations demonstrated true fifth order accuracy [43]. These are the main reasons for specifically choosing the WENO7M scheme. Note, however, that it is not the only scheme suitable for reacting flow simulations. Alternatives could also be pure WENO schemes with enhanced smoothness measures, such as the WENO-Z schemes [44] or schemes with symmetric candidate stencils and bandwidth optimized weights, such as the WENO-SYMBO schemes [45].

The main scope of the paper is threefold: 1) to present the implementation and validation of the newly implemented WENO7M scheme and the stiff ODE-solver in KARFS, 2) to investigate the performance characteristics of KARFS by adopting various types of parallelism strategies and 3) to demonstrate execution of KARFS on a variety of architectures including NVIDIA Tesla P100 GPUs and NVIDIA Kepler K20X GPUs.

2. Governing equations

The DNS code KARFS [36] solves the fully compressible continuity, momentum, total energy, and species equations in conservative form with detailed chemistry for a mixture of ideal gases on structured, Cartesian grids

$$\frac{\partial \rho}{\partial t} = -\frac{\partial (\rho u_i)}{\partial x_i}\,, \tag{1a}$$

$$\frac{\partial (\rho u_i)}{\partial t} = -\frac{\partial \left(\rho u_i u_j\right)}{\partial x_j} - \frac{\partial P}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}\,, \tag{1b}$$

$$\frac{\partial (\rho e_\mathrm{t})}{\partial t} = -\frac{\partial \left(\rho e_\mathrm{t} u_j\right)}{\partial x_j} - \frac{\partial \left(P u_j\right)}{\partial x_j} + \frac{\partial \left(\tau_{ij} u_i\right)}{\partial x_j} - \frac{\partial q_j}{\partial x_j}\,, \tag{1c}$$

$$\frac{\partial (\rho Y_k)}{\partial t} = -\frac{\partial \left(\rho Y_k u_j\right)}{\partial x_j} - \frac{\partial J_{k,j}}{\partial x_j} + \dot{\omega}_k\,, \tag{1d}$$

where the Einstein summation convention is implied, ρ is the density, u_i is the Cartesian velocity component in the i^{th} coordinate direction ($i = 1, 2, 3$), P is the pressure, Y_k is the mass fraction of the k^{th} species and $\dot{\omega}_k$ is its reaction rate, τ_{ij} is the viscous stress tensor given by $\tau_{ij} = \mu(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_l}{\partial x_l})$, where μ is the molecular viscosity. The total specific energy is given by $e_\mathrm{t} = u_i u_i/2 + h - P/\rho$ and the heat flux vector is given by $q_i = -\alpha\frac{\partial T}{\partial x_i} + \sum_k h_k J_{k,i}$, with h, α, and T being the enthalpy, thermal conductivity and temperature, respectively. The species diffusive flux is computed through a mixture-averaged formulation based on Fick's law, $J_{k,i} = -\rho\mathcal{D}_k\frac{\partial Y_k}{\partial x_i} - \rho\mathcal{D}_k\frac{Y_k}{M_k}\frac{\partial M}{\partial x_i}$ where M_k is the species molecular weight, M is the mixture-averaged molecular weight and $\mathcal{D}_k$ is a mixture-averaged diffusion coefficient computed using Cantera [46].

3. Numerical method

In its original formulation, KARFS [36] solves the compressible Navier–Stokes, species and energy equations fully explicitly employing a fourth-order, six-stage explicit Runge–Kutta operator for time integration [25]. To add

robustness and performance gains when dealing with stiff and complex chemistry, an efficient stiff-ODE solver and a second-order operator splitting algorithm have also been implemented. Depending upon the nature of the problem, spatial discretization can be done using either an eighth-order, non-dissipative, central difference operator [24] or a seventh-order, minimally dissipative, mapped weighted essentially non-oscillatory (WENO) operator. A tenth-order de-aliasing filter is also included to remove spurious high wave-number noise. Moreover, the Navier–Stokes characteristic boundary conditions (NSCBC) [47, 48] for reacting flows have been implemented.

The convective term in Equation (1b), $\frac{\partial(\rho u_i u_j)}{\partial x_j}$, makes the system of governing Equations (1a-1d) highly non-linear and is responsible for complex features such as shock waves, contact discontinuities and turbulence. Hence, the use of minimally-dissipative, high-order shock capturing schemes is an essential element when computing complex compressible reacting flows in order to avoid excessive numerical damping of the flow features over a wide range of length scales as well as to prevent spurious numerical oscillations near shock waves and discontinuities. Mapped WENO schemes [39, 43] provide an excellent degree of accuracy and robustness without being prohibitively expensive computationally and, therefore, are considered here. In particular, a seventh-order, minimally dissipative mapped WENO (WENO7M) scheme is implemented and described below. For conciseness, the presentation is restricted to the x-direction only, noting that the same procedure can be easily applied in the y- and z-directions, respectively.

3.1. Implementation of WENO7M

The main concept of the WENO scheme is to use a superposition of several sub-stencils with adaptive coefficients to construct a higher-order approximation of the solution, avoiding the interpolation across discontinuities and preserving a uniformly high-order of accuracy at all points where the solution is smooth. To illustrate this idea, we consider the one-dimensional (x-direction) version of the system of governing Equations (1a-1d):

$$\frac{\partial Q_t}{\partial t} + \frac{\partial C_x}{\partial x} + \frac{\partial D_x}{\partial x} = S \ , \ \text{where} \tag{2}$$

$$Q_t = \begin{pmatrix} \rho \\ \rho u \\ \rho e_t \\ \rho Y_k \end{pmatrix}, \ C_x = \begin{pmatrix} \rho u_x \\ \rho {u_x}^2 + P \\ (\rho e_t + P) u_x \\ \rho Y_k u_x \end{pmatrix}, \ S = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \dot{\omega}_k \end{pmatrix}, \ D_x = \begin{pmatrix} 0 \\ -\tau_{xx} \\ -(\tau_{xx} u_x + \tau_{xy} u_y + \tau_{xz} u_z) + q \\ J_{k,x} \end{pmatrix}$$

In Equation 2, Q_t is the solution vector, C_x is the vector consisting of the convective flux functions, D_x is the vector consisting of the viscous and molecular diffusion flux functions, and S is the vector of source terms. Problems involving shocks or contact discontinuities require separate treatment of the convective and the diffusive flux components. Henceforth, the spatial derivatives of the convective flux functions are determined using the WENO7M scheme whereas the spatial derivatives of the viscous and the molecular diffusion flux functions are determined using an eighth-order centered difference (CD8) scheme. The WENO reconstruction procedure could be directly applied to any given set of quantities: primitive variables, conservative variables or their flux components. For ensuring stability, it is preferred to split the convective flux components in the physical space rather than the characteristic space, as in the present framework. Accordingly, each of the convective flux functions are split using the local Lax-Friedrichs flux splitting methodology [49]:

$$C_x = C_x^+ + C_x^- \ , \ \text{where} \tag{3}$$

$$C_x^+ = \frac{1}{2}(C_x + \lambda_{\max} Q_t) \, , \, C_x^- = \frac{1}{2}(C_x - \lambda_{\max} Q_t) \, .$$

Here, $\lambda_{\max}$ is the maximum local wave propagation speed in x-direction, which can be determined as shown below:

$$(\lambda_{\max})_i = \max((|u_x|)_{i-1}, (|u_x| + c)_{i-1}, (|u_x| - c)_{i-1}, \\ (|u_x|)_i, (|u_x| + c)_i, (|u_x| - c)_i)\,, \tag{4}$$

with u_x being the flow velocity in x-direction and c being the sound speed. For finding the maximum local wave propagation speed in multiple dimensions, the respective flow velocities in y (v_y) and z (w_z) directions are used. The spatial derivative of the convective flux function can be obtained by differentiating Equation 3 with respect to x,

$$\frac{\partial C}{\partial x} = \frac{\partial C^+}{\partial x} + \frac{\partial C^-}{\partial x}\,. \tag{5}$$

In the WENO framework, for each grid point, reconstructed left and right states are determined and used to calculate fluxes at edges as

$$\left.\frac{\partial C^+}{\partial x}\right|_i = \left.\frac{\partial \hat{C}^+}{\partial x}\right|_{i+1/2} - \left.\frac{\partial \hat{C}^+}{\partial x}\right|_{i-1/2}\,,$$
$$\left.\frac{\partial C^-}{\partial x}\right|_i = \left.\frac{\partial \hat{C}^-}{\partial x}\right|_{i+1/2} - \left.\frac{\partial \hat{C}^-}{\partial x}\right|_{i-1/2}\,.$$

This is accomplished in KARFS by using the WENO7M interpolator as follows:

$$\left.\frac{\partial \hat{C}^+}{\partial x}\right|_{i+1/2} = \frac{\text{WENO7M}(C^+_{i-3}, C^+_{i-2}, C^+_{i-1}, C^+_i, C^+_{i+1}, C^+_{i+2}, C^+_{i+3})}{\Delta x}\,, \tag{7a}$$

$$\left.\frac{\partial \hat{C}^+}{\partial x}\right|_{i-1/2} = \frac{\text{WENO7M}(C^+_{i-4}, C^+_{i-3}, C^+_{i-2}, C^+_{i-1}, C^+_i, C^+_{i+1}, C^+_{i+2})}{\Delta x}\,, \tag{7b}$$

$$\left.\frac{\partial \hat{C}^-}{\partial x}\right|_{i+1/2} = \frac{\text{WENO7M}(C^-_{i+4}, C^-_{i+3}, C^-_{i+2}, C^-_{i+1}, C^-_i, C^-_{i-1}, C^-_{i-2})}{\Delta x}\,, \tag{7c}$$

$$\left.\frac{\partial \hat{C}^-}{\partial x}\right|_{i-1/2} = \frac{\text{WENO7M}(C^-_{i+3}, C^-_{i+2}, C^-_{i+1}, C^-_i, C^-_{i-1}, C^-_{i-2}, C^-_{i-3})}{\Delta x}\,. \tag{7d}$$

The equations (7a-7d) give an approximation of the spatial derivative of the numerical flux function [39, 42] at $i\pm1/2$. The functional form of the WENO7M interpolator in equation (7a) is given by

$$\text{WENO7M}(C^+_{i-3}, C^+_{i-2}, C^+_{i-1}, C^+_i, C^+_{i+1}, C^+_{i+2}, C^+_{i+3}) = \sum_{k=0}^{3} w_k a_k\,, \tag{8}$$

where w_k are the WENO7M weights and the component stencils a_k are

$$a_0 = \frac{1}{24}(-6C^+_{i-3} + 26C^+_{i-2} - 46C^+_{i-1} + 50C^+_i)\,, \tag{9a}$$
$$a_1 = \frac{1}{24}(2C^+_{i-2} - 10C^+_{i-1} + 26C^+_i + 6C^+_{i+1})\,, \tag{9b}$$
$$a_2 = \frac{1}{24}(-2C^+_{i-1} + 14C^+_i + 14C^+_{i+1} - 2C^+_{i+2})\,, \tag{9c}$$
$$a_3 = \frac{1}{24}(6C^+_i + 26C^+_{i+1} - 10C^+_{i+2} + 2C^+_{i+3})\,. \tag{9d}$$

The weights w_k are formulated in two steps as outlined in Ref. [43]. The final weights are first approximated following the procedure described in [42] as

$$w_k^* = \frac{\gamma_k}{\sum_{i=0}^{3}\gamma_i}, \text{ where } \gamma_k = \frac{\bar{w}_k}{(\epsilon+\beta_k)^p}. \tag{10}$$

The ideal weights, $\bar{w}_k$, are constants given by

$$\bar{w}_0 = \frac{1}{35}, \ \bar{w}_1 = \frac{12}{35}, \ \bar{w}_2 = \frac{18}{35}, \ \bar{w}_3 = \frac{4}{35}, \tag{11}$$

and the smoothness indicators β_k are defined as

$$\begin{aligned}
\beta_0 =& \frac{1}{36}(-2C_{i-3}^+ + 9C_{i-2}^+ - 18C_{i-1}^+ + 11C_i^+)^2 \\
&+ \frac{13}{12}(-1C_{i-3}^+ + 4C_{i-2}^+ - 5C_{i-1}^+ + 2C_i^+)^2 \\
&+ \frac{781}{720}(-C_{i-3}^+ + 3C_{i-2}^+ - 3C_{i-1}^+ + C_i^+)^2, && \text{(12a)} \\
\beta_1 =& \frac{1}{36}(C_{i-2}^+ - 6C_{i-1}^+ + 3C_i^+ + 2C_{i+1}^+)^2 \\
&+ \frac{13}{12}(C_{i-1}^+ - 2C_i^+ + C_{i+1}^+)^2 \\
&+ \frac{781}{720}(-C_{i-2}^+ + 3C_{i-1}^+ - 3C_i^+ + C_{i+1}^+)^2, && \text{(12b)} \\
\beta_2 =& \frac{1}{36}(-2C_{i-1}^+ - 3C_i^+ + 6C_{i+1}^+ - C_{i+2}^+)^2 \\
&+ \frac{13}{12}(C_{i-1}^+ - 2C_i^+ + C_{i+1}^+)^2 \\
&+ \frac{781}{720}(-C_{i-1}^+ + 3C_i^+ - 3C_{i+1}^+ + C_{i+2}^+)^2, && \text{(12c)} \\
\beta_3 =& \frac{1}{36}(-11C_i^+ + 18C_{i+1}^+ - 9C_{i+2}^+ + 2C_{i+3}^+)^2 \\
&+ \frac{13}{12}(2C_i^+ - 5C_{i+1}^+ + 4C_{i+2}^+ - C_{i+3}^+)^2 \\
&+ \frac{781}{720}(-C_i^+ + 3C_{i+1}^+ - 3C_{i+2}^+ + C_{i+3}^+)^2. && \text{(12d)}
\end{aligned}$$

In Equation 10, $\epsilon = 10^{-40}$ is a small number that prevents division by zero errors [39] and p may be varied to increase or decrease the WENO7M adaptation sensitivity. Unless otherwise stated, $p = 2$ in the present study. The approximated w_k^* are now mapped to the corrected w_k such that the accuracy of the method is seventh order in general. This is achieved through the mapping procedure outlined in Ref. [43]

$$g_k(w) = \frac{w(\bar{w}_k + \bar{w}_k^2 - 3\bar{w}_k w + w^2)}{\bar{w}_k^2 + (1-2\bar{w}_k)w}. \tag{13}$$

The final corrected weights are then given by

$$w_k = \frac{g_k(w_k^*)}{\sum_{i=0}^{3} g_i(w_i^*)}. \tag{14}$$

From a close inspection of the set of Equations (7a-7d), it can be noticed that the spatial derivatives of positive and negative convective flux components at the left edge, i.e. $\frac{\partial \hat{C}^+}{\partial x}\Big|_{i-1/2}$ and $\frac{\partial \hat{C}^-}{\partial x}\Big|_{i-1/2}$, can be obtained by merely shifting the index of the corresponding spatial derivatives evaluated at the right edge, i.e. $\frac{\partial \hat{C}^+}{\partial x}\Big|_{i+1/2}$ and $\frac{\partial \hat{C}^-}{\partial x}\Big|_{i+1/2}$, by one to the left:

$$\begin{aligned}\frac{\partial \hat{C}^{\pm}}{\partial x}\Big|_{i-1/2} &= \frac{\text{WENO7M}(\text{C}^+_{\text{i}-4}, \text{C}^+_{\text{i}-3}, \text{C}^+_{\text{i}-2}, \text{C}^+_{\text{i}-1}, \text{C}^+_{\text{i}}, \text{C}^+_{\text{i}+1}, \text{C}^+_{\text{i}+2})}{\Delta x} \\ &= \frac{\text{WENO7M}(\text{C}^+_{\text{i}-3-1}, \text{C}^+_{\text{i}-2-1}, \text{C}^+_{\text{i}-1-1}, \text{C}^+_{\text{i}-0-1}, \text{C}^+_{\text{i}+1-1}, \text{C}^+_{\text{i}+2-1}, \text{C}^+_{\text{i}+3-1})}{\Delta x} \\ &= \frac{\partial \hat{C}^{\pm}}{\partial x}\Big|_{i+1/2-1}. \end{aligned} \tag{15}$$

Hence, the WENO7M operator needs to be used only twice instead of four times for computing the numerical fluxes. While the number of convective flux evaluations does reduce to half as per Equation 15, the number of MPI communications is increased when performing the aforementioned shift. Also note that for non-periodic domains, special handling is needed when evaluating the spatial derivatives of the convective flux functions at the boundaries. Using Equation 5, the spatial derivative of the convective flux function can finally be evaluated. Subsequently, the spatial derivatives of the viscous and molecular diffusion flux functions are determined using the CD8 scheme and the solution is advanced in time. This implementation procedure can be easily extended to implement other WENO schemes such as WENO-Z [44] or WENO-SYMBO [45] as described in the Supplementary Material.

3.2. Operator splitting scheme

The system of equations (1a–1d) describe physical processes that exhibit different temporal and spatial characteristics. In combustion applications, numerical stiffness mainly originates from the computation of the chemical reaction rates due to a large spectrum of chemical timescales, and it becomes more significant as the size of the chemical kinetics mechanism increases. To achieve numerically efficient and stable computations, an implicit time integration scheme is preferred, particularly when dealing with a large number of species and reactions.

Operator splitting methods allow to treat different terms of the equations using different temporal discretization schemes. By applying operator splitting to the convection-diffusion-reaction system of equations, the temporal discretization can be divided into several sub-steps at each time step, separating the reaction part from the transport one.

The splitting method adopted herein is second order accurate and is often referred to as Strang splitting [50]. To illustrate the method, consider the vector of conserved variables (Q), the transport operator (T), which may contain either convection (C) or diffusion (D) or both, and the chemical source term operator (S). First, the transport part is integrated over half of the time step, $\Delta t/2$. Next, the chemical source term is integrated over the original time step, Δt. Last, the transport part is again integrated over the remaining $\Delta t/2$. The procedure is mathematically expressed as follows:

$$\frac{\partial Q}{\partial t} = C(Q) + D(Q) + S(Q) = T(Q) + S(Q), \tag{16}$$

$$\frac{\partial Q^*}{\partial t} = T(Q^*), \quad Q^*(t_n) = Q(t_n), \qquad \text{for } t = [t_n, t_n + \Delta t/2] \tag{17}$$

$$\frac{\partial Q^{**}}{\partial t} = S(Q^{**}), \quad Q^{**}(t_n) = Q^*(t_{n+1/2}), \qquad \text{for } t = [t_n, t_n + \Delta t] \tag{18}$$

$$\frac{\partial Q^{***}}{\partial t} = T(Q^{***}), \quad Q^{***}(t_{n+1/2}) = Q^{**}(t_{n+1/2}), \qquad \text{for } t = [t_n + \Delta t/2, t_n + \Delta t] \tag{19}$$

$$Q(t_{n+1}) = Q^{***}(t_{n+1}). \tag{20}$$

For the time integration of the transport sub-steps (Eqs. (17) and (19)), the explicit Runge–Kutta method is employed, whereas for the chemistry sub-step (Eq. (18)), the well known implicit stiff-ODE CVODE solver is utilized,

which is part of the SUNDIALS library [51]. Furthermore, the linear algebra computations related to the stiff-ODE integrator are accelerated by making use of the MAGMA library [40], which was designed for hybrid computing architectures. The CVODE-MAGMA framework has been previously coupled with Cantera and applied to the calculation of ignition delays, achieving speedups of two orders of magnitude [36].

4. Results and discusion

4.1. Sod's shock tube problem

The popular Sod shock tube problem [52] constitutes probably one of the most standard numerical benchmarks designed for compressible flow solvers. This test consists of a fluid, initially at rest, in which a virtual membrane located at the center of the domain separates two distinct sections: the one on the left at a higher density and pressure, and the one on the right at a lower density and pressure. The membrane is removed at $t = 0$ and a shock wave develops propagating toward the right, followed by a contact discontinuity and a rarefaction wave propagating to the left. The application of KARFS to this case serves to validate the implementation of the WENO7M scheme and evaluate its performance. In particular, we consider a one-dimensional (1D) setup, solving the 1D Euler equations. The tube is 1 m long and the initial conditions for pressure P, density ρ, and velocity u are as follows:

$$(P, \rho, u) = \begin{cases} (4.0 \text{ MPa}, 4.62659 \text{ x } 10^{-4} \text{ kg/m}^3, 0 \text{ m/s}), & x \leq 0.5 \text{ m}, \\ (1.0 \text{ MPa}, 1.15665 \text{ x } 10^{-4} \text{ kg/m}^3, 0 \text{ m/s}), & x > 0.5 \text{ m}. \end{cases} \tag{21}$$

The domain is uniformly discretized with 1200 grid points and the equations are integrated in time with a constant time step size of 1 ns. The fluid considered here is air and is assumed to be a perfect gas. The numerical solutions for pressure, density, velocity and specific internal energy, e, at time $t = 2.5$ μs are shown in Figure 1. The numerical solution provided by the WENO7M scheme in KARFS closely follows the exact solution. It is also seen that there is negligible spreading of the numerical solution in the vicinity of the contact discontinuity, which has been previously documented for the standard WENO family of schemes [42]. These results suggest that the WENO7M scheme, when properly applied to compressible DNS codes, can yield an excellent shock capturing capability.

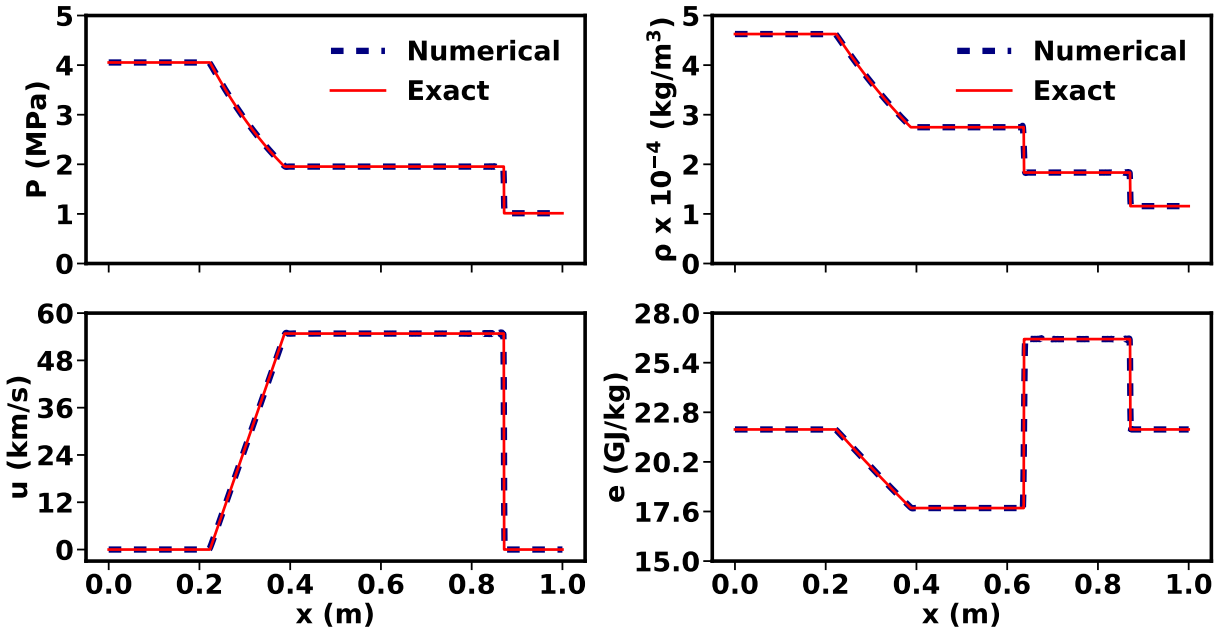

Figure 1: Comparison of exact and numerical solutions for pressure, density, velocity and specific internal energy at $t = 2.5$ μs.

4.2. Decay of isotropic turbulence

Turbulence is an irregular motion that can be described by statistical values, depending on both position and time, of the quantities that characterize the flow field. When all possible states of turbulence are equally probable at any given point of the flow field, turbulence is said to be homogeneous. When turbulence is isotropic, the statistical averages are independent of the coordinate system. Isotropic, homogeneous turbulence is an idealization and its study is the simplest possible study of turbulence. In this section, we consider decaying isotropic turbulence to test the efficiency of the WENO7M method in resolving all the relevant turbulent length scales and time scales. The governing equations are the equations (1a-1d). The simulation domain is a cube with 128^3 grid points and the boundary conditions are periodic in all three Cartesian directions for all the flow variables. We start with a uniform density, pressure, temperature and a Passot-Pouquet spectrum [53] for velocity which is divergence-free initially and given by

$$E(k) = 16\sqrt{\frac{2}{\pi}}\left(\frac{u'^2}{k_e}\right)\left(\frac{k}{k_e}\right)^4 \exp\left(-2\left(\frac{k}{k_e}\right)^2\right), \tag{22}$$

where k is the wave number, k_e is the wave number at which the spectrum peaks, and $u' = \langle(u_x^2+v_y^2+w_z^2)/3\rangle^{1/2}$ is the turbulent fluctuating velocity. Here, $\langle....\rangle$ is a volume average over the computational domain at a fixed time instant. We consider the flow conditions where shock waves are not present and the comparison against the CD8 scheme is meaningful. Specifically, the turbulent Mach number, $M_t = 0.17$ and the Reynolds number based on the Taylor micro-scale (λ), $Re_\lambda = 35$, where

$$M_t = \frac{\langle u_x{}^2 + v_y{}^2 + w_z{}^2\rangle^{1/2}}{\langle c\rangle}, \text{ and} \tag{23}$$

$$Re_\lambda = \frac{u'\lambda}{\langle\nu\rangle}. \tag{24}$$

k_e is determined from the most energetic length scale, l_e, which has been set as 446 μm. As before, the fluid considered here is air with an average density, $\langle\rho\rangle = 1.17$ kg/m^3 and an average temperature $\langle T\rangle = 300$ K. Under the chosen conditions, the Kolmogorov length scale, $l_k = 17.7$ μm. Accordingly, a uniform grid resolution of 8.8 μm is used such that there are at least 2 grid points across the Kolmogorov length scale and numerically converged results are ensured. Based on the chosen conditions, the highest frequency of turbulence captured in the simulation is 4 MHz. We simulate the decay of isotropic turbulence for 20 non-dimensional time periods based on the initial large-eddy turn over time, $\tau_{eddy} = \lambda/u'$.

Figure 2(a) shows the temporal evolution of normalized turbulent kinetic energy, K_e for the CD8 and WENO7M schemes. As stated earlier in section 3.1, the power parameter, p, can be varied to increase or decrease its adaptation sensitivity. Irrespective of the scheme employed, the normalized kinetic energy decays at almost an identical rate until $t \leq 7\tau_{eddy}$. It is also seen that with $p = 2$, the WENO7M scheme has significantly higher dissipation than the CD8 scheme for $t \geq 10\tau_{eddy}$. However, the dissipation of the WENO7M scheme is found to be significantly reduced by setting $p = 1$. Figure 2(b) shows the energy spectra at $t = 20\ \tau_{eddy}$ for the same simulations. Minor aliasing errors are found for the WENO7M scheme in the tail of the spectrum, which do not affect the overall results. These results indicate that the WENO7M scheme, with an appropriate value of power parameter p, can be successfully used to simulate turbulent flows, especially in the absence of shocks. It is important to note that the WENO family of schemes tend to have higher numerical dissipation because of insufficient distinction between shock-containing and smooth regions on typical DNS grids. Decreasing the dissipation that is inherent to the WENO adaptation mechanism is challenging because potential deficiencies relevant to the damping of turbulent features may become apparent only in realistic simulations and must then be examined locally.

4.3. Detonation in a thermally stratified constant volume reactor

Detonation development from a flame kernel initiated by a pre-ignition event is demonstrated here for further assessing the robustness of the newly implemented WENO7M scheme. The specific test case has been adopted from a recent study by Sow et al [54]. The WENO7M scheme is implemented for the convective terms whereas the eighth-order central difference scheme is employed for the diffusive terms. The solution is advanced in time with a fourth-order, six-stage explicit Runge-Kutta scheme. The implicit stiff ODE CVODE solver with Strang splitting

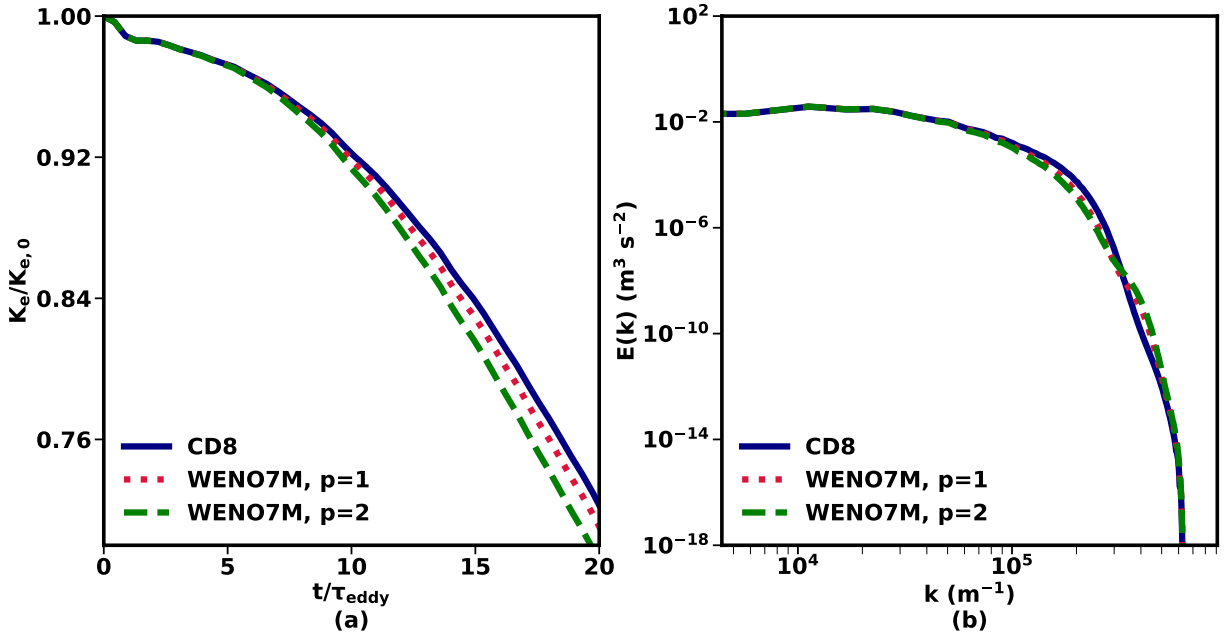

Figure 2: DNS data of decaying isotropic turbulence at $M_t = 0.17$ and $Re_\lambda = 35$, comparing the WENO7M and the CD8 schemes. (a) Temporal evolution of turbulent kinetic energy and (b) energy spectra at $t = 20\ \tau_{eddy}$.

is also employed to deal with chemical stiffness. The numerical configuration (see Figure 3) is a constant volume reactor with impermeable and adiabatic walls at both boundaries. For the initial condition, the Cantera [46] solution for a freely-propagating stoichiometric hydrogen-air flame with detailed chemistry [55] is mapped near the left wall to represent the initial flame front developed by the pre-ignition event.

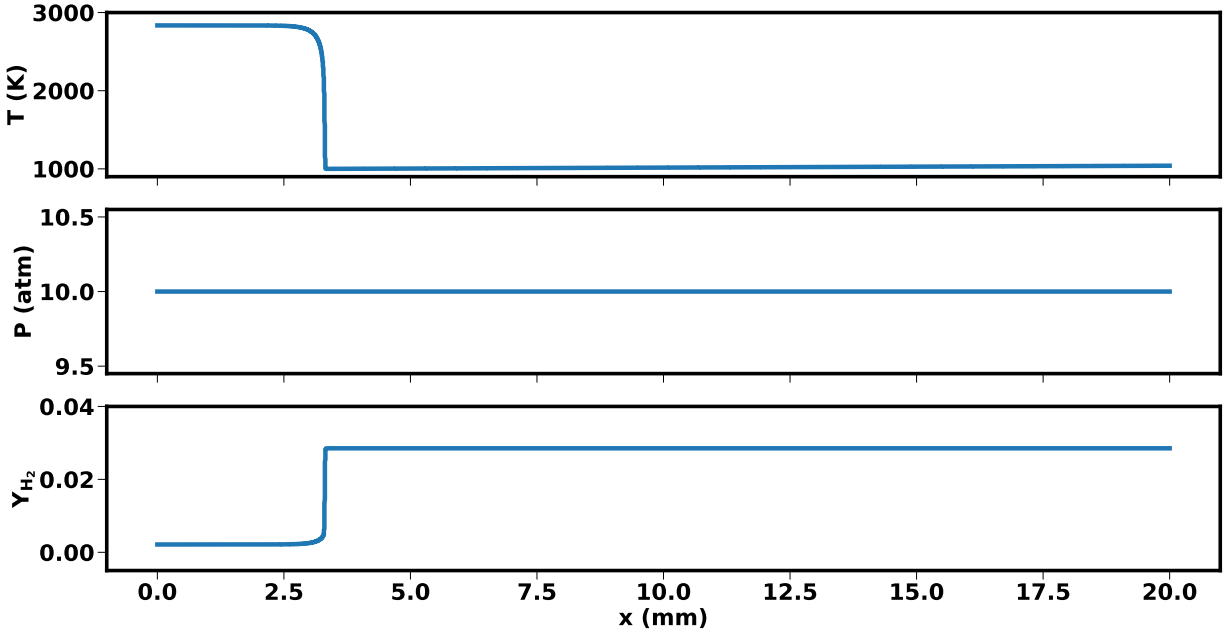

Figure 3: Numerical configuration for simulating detonation in a thermally stratified constant-volume reactor, showing the initial temperature, pressure and fuel (H_2) mass fraction profiles.

The flame travels from the left to the right where the initial pressure and velocity are constant and set to 10 atm and 0 m/s, respectively. The thermal stratification is modeled as a linear function and the magnitude of the temperature gradient, dT/dx, is set to 2.4 K/mm, which corresponds to the temperature difference of 40 K over the length of the temperature variation. The total domain length, x_L, is 20 mm and the initial position, x_0, where the temperature variation starts is 3.35 mm. A uniform grid spacing of 1 μm and a constant time step of 0.2 ns are used in the present study. The laminar flame thickness based on temperature and a progress variable in terms of the hydrogen (H_2) mass fraction is computed to be 34.8 μm and 20.2 μm, respectively.

Figure 4 depicts some results obtained at six different times. Due to a positive temperature gradient, the end gas ignites. Propagation towards the flame amplifies the pressure wave. A subsequent increase in the chemical heat release is also noticeable during this period. When the auto-ignition front transitions to a developing detonation wave (256.8 μs), there is enough fresh mixture ahead of it. The reactive shock wave develops afterwards and collides with the pre-ignition front.

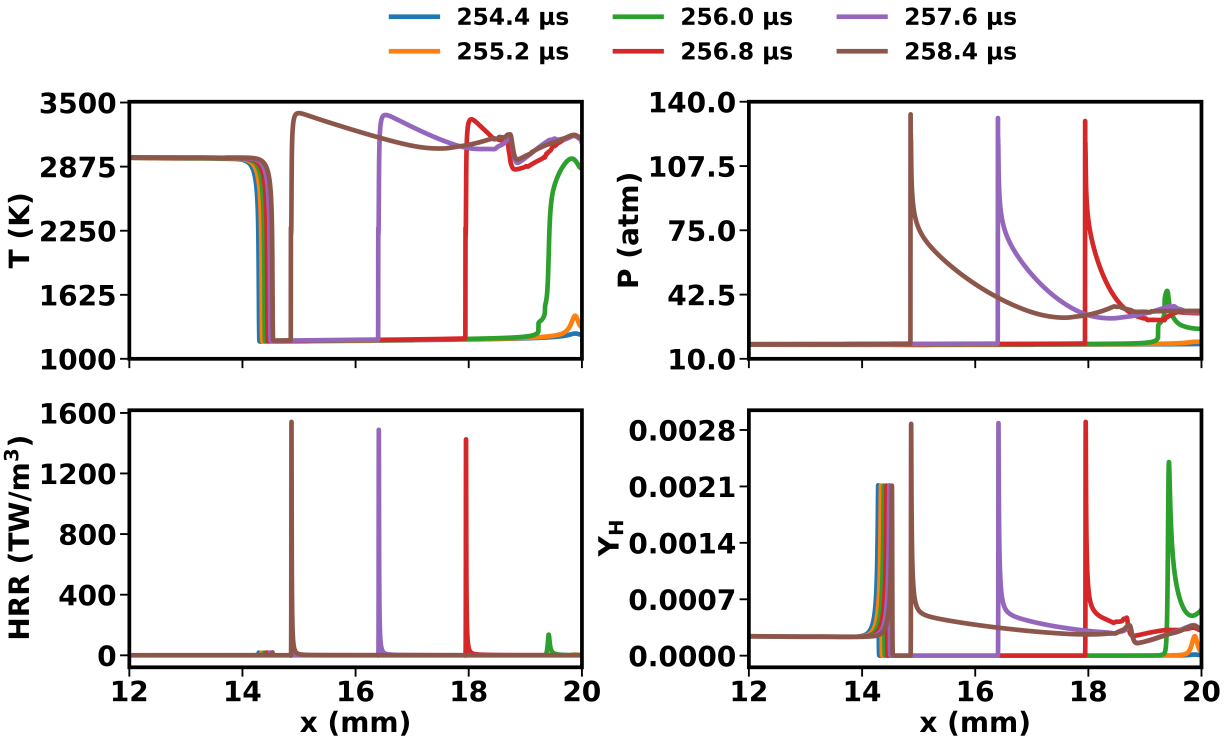

Figure 4: Temporal evolution of temperature, pressure, heat release rate, and H radical mass fraction.

4.4. Flame propagation into a compositionally stratified mixture under compression heating

As discussed earlier in the introduction, when a propagating reaction front encounters an adverse gradient in reactivity, it can turn into a contact discontinuity with steep gradients in density, temperature and composition. In internal combustion (IC) engines, compression of the fresh gas mixture due to piston motion and the propagating reaction front further aggravates this process. This phenomenon has also been observed in experimental investigations of IC engines [13, 14, 56]. An accurate prediction of front propagation speed under such conditions still remains a challenge. Moreover, the presence of contact discontinuities may render the central difference schemes cost prohibitive due to very fine grid resolution requirements or ineffective in capturing the underlying physics altogether. As such, the importance of the WENO7M scheme to multidimensional reacting flows is further emphasized. To compare the performance of the WENO7M scheme and the CD8 scheme in accurately resolving the physics of reacting flows with contact discontinuities, the computation of an initially planar flame propagating into a compositionally stratified mixture under compression heating is conducted in this part of the study.

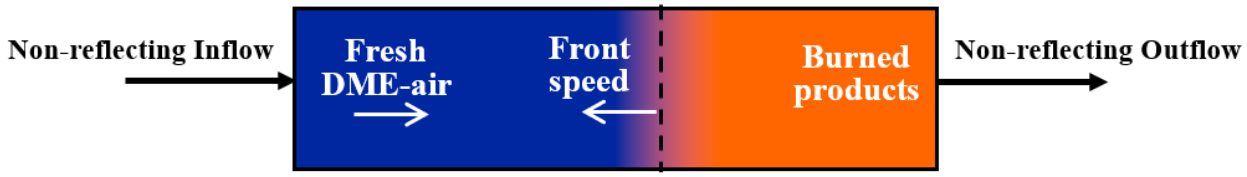

Figure 5: Numerical configuration for simulating flame propagation into a compositionally stratified mixture under compression heating.

A two-dimensional (2D) simulation is performed for a rectangular domain with an inlet and outlet along the axis of propagation. The simulation is carried out using both, the CD8 and the WENO7M schemes. A uniform mixture of dimethyl-ether (DME) and air, at an equivalence ratio of 0.4, initial temperature of 800 K and an elevated pressure of 20 atm, is considered with the description of chemical kinetics by the mechanism of Bhagatwala et al. [4]. The rectangular domain is 9.75 mm (L_x) long and 1.44 mm (L_y) wide. It is discretized with 249,600 points (1300 $\times$ 192) with a uniform grid spacing of 7.5 μm (corresponding to at least 12 grid points across the thinnest species reaction rate layer which in the present study has been identified to be CH_3O) when using the WENO7M scheme. The CD8 scheme with an identical grid spacing was noticed to cause spurious oscillations due to its inability to capture the contact discontinuity. As such, a uniform grid spacing of 3.75 μm (corresponding to at least 24 grid points across the thinnest CH_3O reaction rate layer) is required when using the CD8 scheme. Therefore, the total number of grid points required for the CD8 scheme is 4 times higher. Moreover, unlike the WENO7M scheme for which the de-aliasing filter is applied at every 10 time-steps, the filter is applied at every time step while using the CD8 scheme so as to further prevent high wave-number oscillations.

To mimic pressure rise due to piston motion and front propagation, an inert mass source term is added to the governing equations (1a–1d), following the previous approaches [10, 20, 57–59]. A schematic of the numerical configuration used for this case is shown in Figure 5. Pockets of stratified temperature and composition fields are routinely observed to occur in IC engines due to wall heat transfer and imperfect mixing with the residuals. Hence, apart from the addition of inert mass source term to the governing equations, the velocity and temperature at the inlet are held constant whereas the mass fraction of DME at the inlet is varied using a monochromatic 2D sinusoidal wave as

$$Y_{\mathrm{DME}}(t) = Y_{\mathrm{DME}}(0) + A \sin\left(\frac{2\pi t}{\tau_0}\right) \sin\left(\frac{2\pi y}{\lambda}\right), \tag{25}$$

where t is time, A is the wave amplitude, τ_0 is the time period, y is the coordinate along the width of the domain, and λ is the length-scale of the stratification eddies. At all times, the deficit or excess of DME concentration is also compensated by adjusting the air concentration. This configuration corresponds to an IC engine condition wherein a flame propagates into a pocket of stratified concentration field of a certain length scale. Non-reflecting inflow and outflow boundary conditions [47] are also imposed to avoid large pressure waves within the domain. The value of $A = 0.2$ has been chosen such that the root-mean-square of the equivalence ratio at the inlet is of the same order of magnitude as the one that been used in previous studies [15, 59]. In a recent study [5], it was found that stratification eddies, with time-scales comparable to the auto-ignition delay times, have maximum interaction with the ignition chemistry and significantly affect the front propagation speed. Hence, for the present study, $\tau_0 = 240$ μs and $\lambda = 0.4775$ mm such that the domain has three stratification eddies along the width.

The simulation is initially run till $t = 1.92$ ms with constant inflow velocity and temperature of 2 m/s and 800 K, respectively, such that the flame remains statistically stationary. This is carried out to generate pockets of stratified composition field ahead of the flame without causing any change its propagation speed, S_c (defined as per Equation 26). It should be noted that chemistry is active throughout the entire simulation time. At time $t = 1.92$ ms, the fresh gas mixture is isentropically compressed such that the pressure in the domain increases from 20 atm to 40 atm, indicating the pressure rise due to piston motion and flame propagation. The pressure rise rate is controlled so as to match the combustion duration time in a typical homogeneous charge compression ignition (HCCI) engine [3].

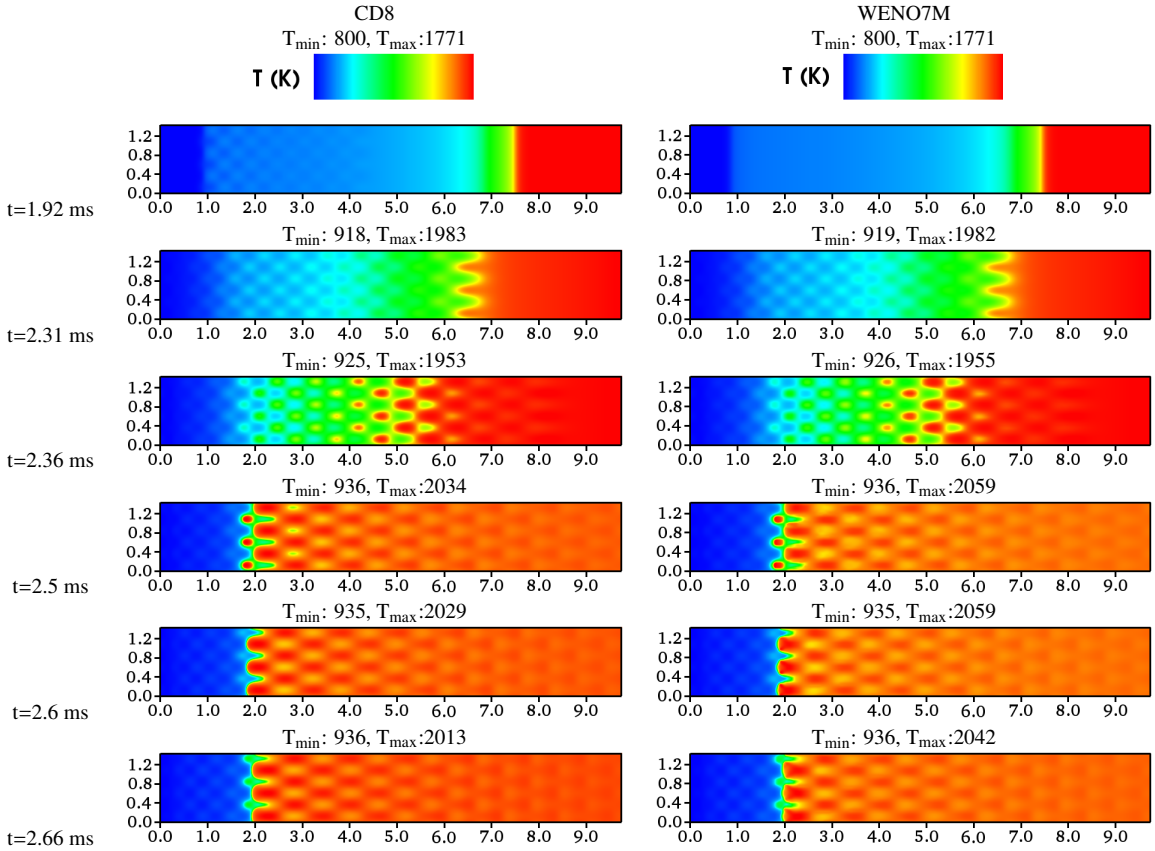

Figure 6: Instantaneous temperature fields obtained using the eighth-order centered difference (CD8) scheme and the seventh-order mapped WENO (WENO7M) scheme. The domain dimensions are in mm.

The pressure and temperature at the boundaries are also varied isentropically at the same rate. A comparison of the instantaneous temperature fields obtained using both the CD8 and the WENO7M schemes is shown in Figure 6. At $t = 1.92$ ms, the temperature fields obtained using either of the two methods are almost indistinguishable.

Once the compression heating is started at $t = 1.92$ ms, the flame begins to propagate towards the inlet by consuming the upstream fresh gas. At the same time, pre-ignition is observed to occur in the upstream fresh gas due to the rise in temperature and pressure caused by compression heating, as can be seen between $t = 2.31$ ms and $t = 2.5$ ms. This is also evident by examining the mass fractions of major (O_2) and minor (OH) species as depicted in Figures 7 and 8. The minimum and maximum values of temperature, Y_{O_2} and Y_{OH} obtained using either of the two methods are nearly identical till $t = 2.36$ ms. However, for $t \geq 2.5$ ms, a difference in peak values of the respective quantities obtained using either of the two methods is observed. A lower peak value of temperature, Y_{O_2}, Y_{OH} observed when using the CD8 scheme is mainly due to the application of the de-aliasing filter to the solution vector at every time step.

Compression heating due to piston motion and flame propagation causes the fresh gas to spontaneously auto-ignite, resulting in an abrupt change in temperature and hence density. This leads to the formation of a contact discontinuity which, by definition, is a surface that separates zones of different density and temperature while being in pressure equilibrium. As mentioned earlier, the CD8 scheme created spurious oscillations due to its inability to capture the contact discontinuity when a uniform grid resolution of 7.5 μm was used, leading to an unstable solution. Subsequently, the CD8 scheme provided a stable solution with a finer grid resolution of 3.75 μm. It also necessitated the use of the de-aliasing filter at every time step which is observed to damp out the flow features as well. Unlike the CD8 scheme, the WENO7M scheme remains stable through the application of the de-aliasing filter at every 10 time-steps on a coarser 7.5 μm grid and provides insight into the dominant physics at play when a flame propagates

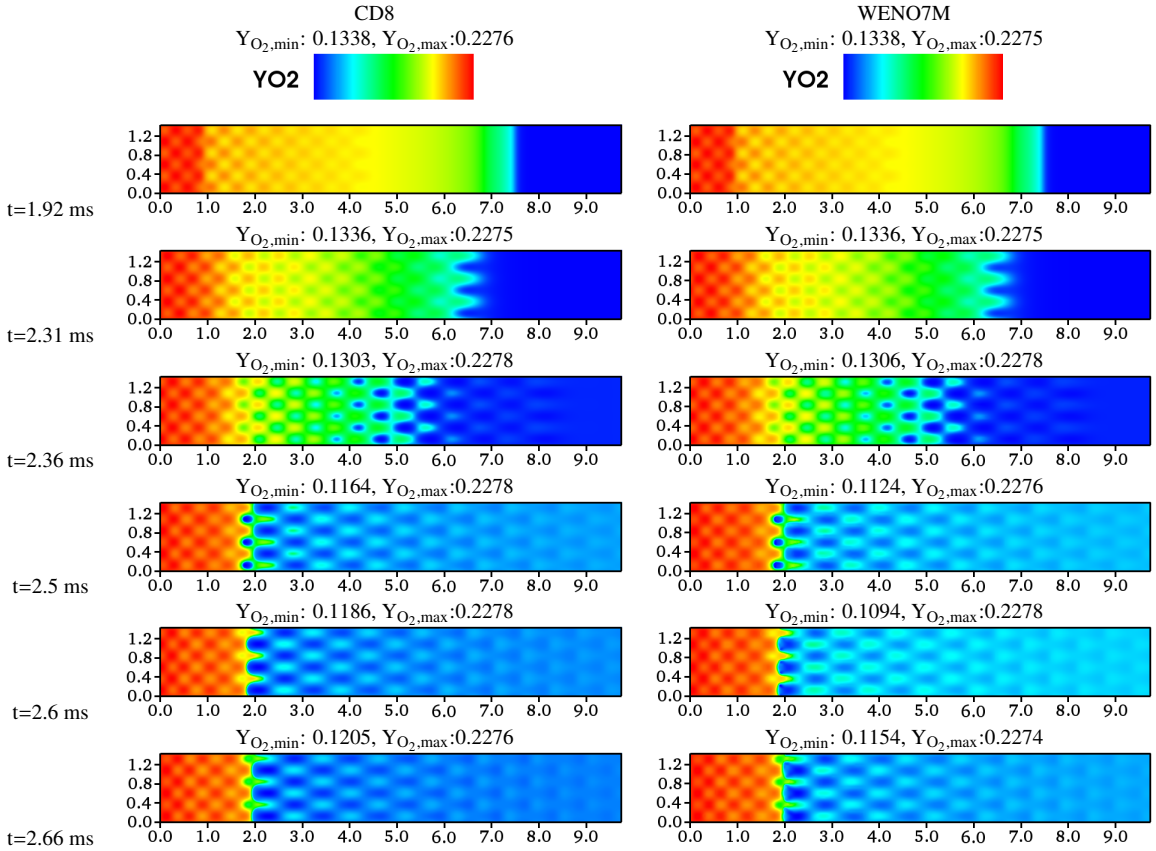

Figure 7: Instantaneous mass fraction of oxygen (O_2) obtained using the eighth-order centered difference (CD8) scheme and the seventh-order mapped WENO (WENO7M) scheme. The domain dimensions are in mm.

into a compositionally stratified mixture under compression heating. The front propagation speed, S_c, using either of the two discretization schemes, is computed based on oxygen consumption as

$$S_c = -\frac{1}{\rho_u (Y^u_{O_2} - Y^b_{O_2})} \int_0^{L_x} \int_0^{L_y} \dot{\omega}_{O_2} \, dy \, dx \,, \tag{26}$$

where ρ_u is the unburnt gas density, $Y^u_{O_2}$ and $Y^b_{O_2}$ is the mass fraction of oxygen in the unburnt and burnt gas respectively and $\dot{\omega}_{O_2}$ is the oxygen reaction rate. The values of S_c obtained with the two discretization schemes at different times is reported in Table 1. It is noticed that S_c quickly rises from 2 m/s at 1.92 ms to $\approx$ 24 m/s at 2.36 ms due to multiple pre-ignition events occurring ahead of the flame. As the simulation progresses further in time, S_c drops significantly (as observed at 2.66 ms) and eventually stabilizes at 2 m/s which corresponds to the velocity imposed

Time (ms)	S_c (m/s) (CD8)	S_c (m/s) (WENO7M)
1.92	2.00	2.00
2.31	22.44	22.42
2.36	23.56	23.86
2.50	2.91	2.84
2.60	1.24	1.23
2.66	1.50	1.39

Table 1: Instantaneous front propagation speed, S_c, obtained using the two discretization schemes.

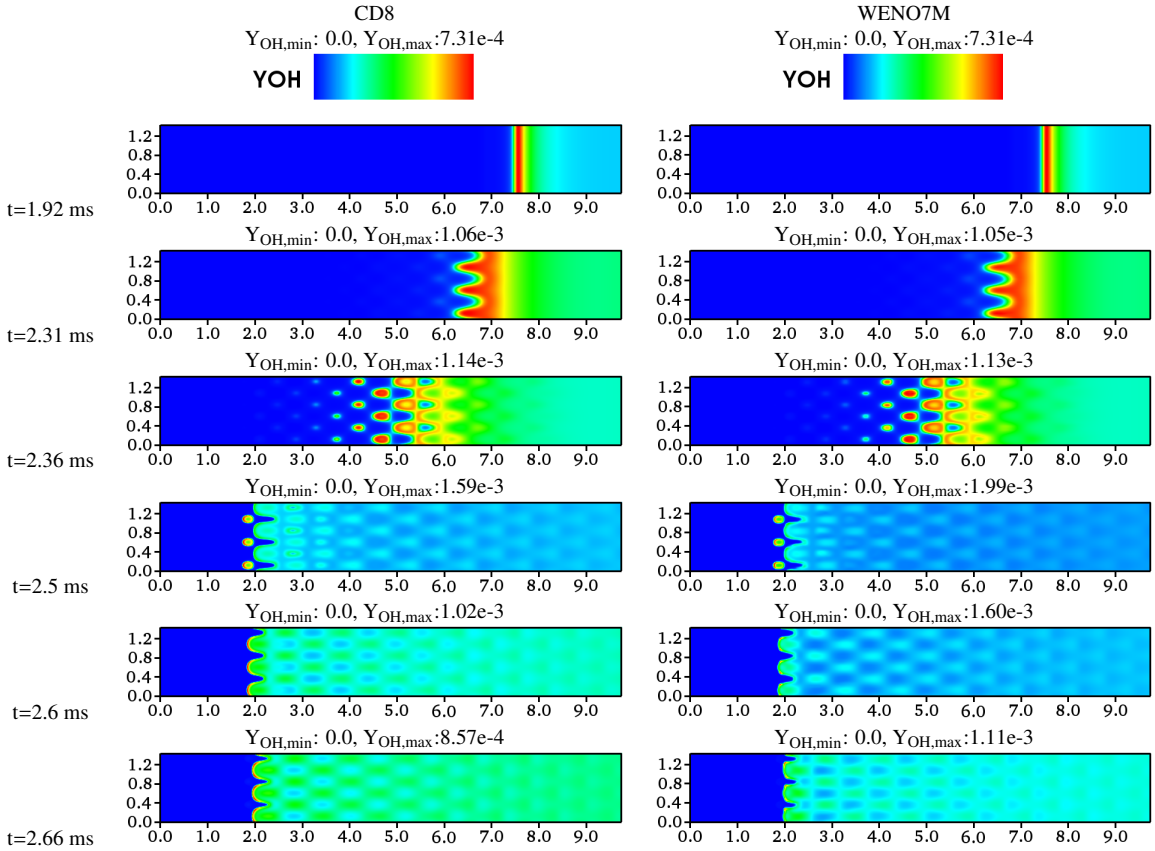

Figure 8: Instantaneous mass fraction of hydroxyl (OH) radical obtained using the eighth-order centered difference (CD8) scheme and the seventh-order mapped WENO (WENO7M) scheme. The domain dimensions are in mm.

at the inlet. Note that the difference in instantaneous S_c obtained using either of the two discretization schemes is negligible, even though the grid spacing used for the CD8 scheme is half that of the WENO7M scheme. Moreover, due to the increased number of grid points needed for the CD8 scheme, the total wall-clock time was found to be about twice that of the WENO7M scheme when the simulation was run on an identical number of compute nodes. It is worth noting here that while the shock tube and the decaying isotropic turbulence have been used by previous studies for validating WENO schemes, the other two cases, namely the detonation of a constant volume combustor and flame propagation in a stratified mixture with compression heating have not been used by any other study to assess the implementation and robustness of the WENO scheme.

4.5. Multi-dimensional simulations of super-knock under realistic internal combustion engine conditions

First-principle direct numerical simulation allows unraveling the complex interplay between turbulence and chemical reactions to provide a better understanding of the mechanism of detonation development encountered in modern combustion devices under extreme high-load operating conditions, and to develop a reliable predictive model for real-world industrial applications. However, large-scale turbulence, combustion, and detonation simulations pose a significant challenge in terms of a wide spectrum of length and time scales, which requires extremely fine spatial and temporal resolutions to capture the highly intermittent localized phenomena, resulting in intensive checkpoint output data and extensive computational resources. Multidimensional simulations of super-knock phenomena in combustion devices face such highly intensive input/output (I/O) and computational resource requirements [60, 61].

Super-knock is typically encountered in a downsized and boosted engine due to its higher power density per volume and being operated at high-load conditions, which makes it more prone to detonation development [62]. Super-knock is characterized by excessive pressure oscillations and extremely high-pressure spikes [60, 61] that may

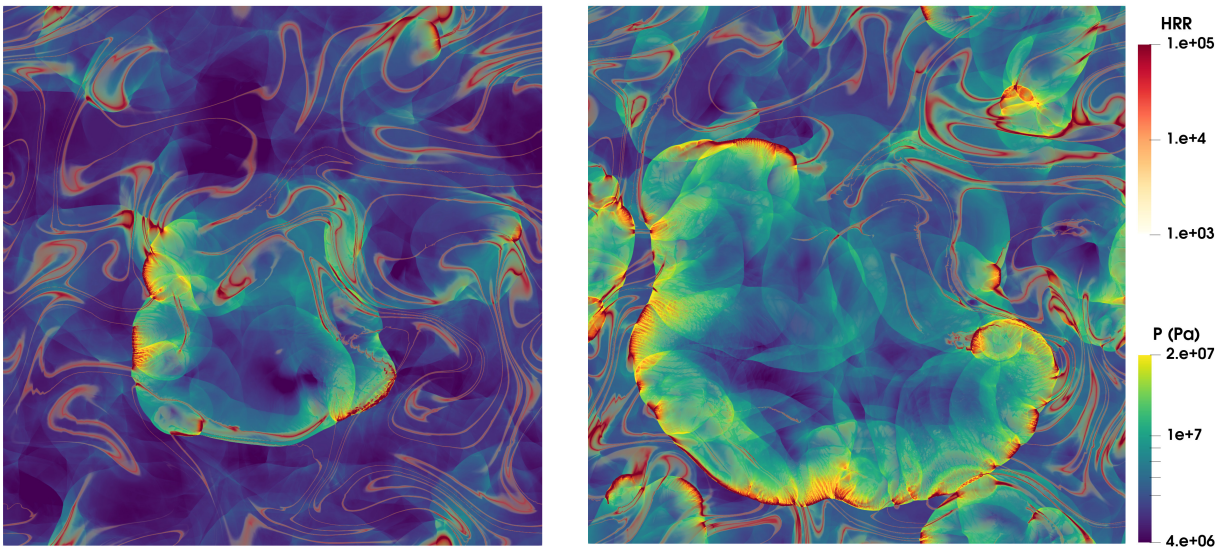

Figure 9: Two-dimensional contours of pressure and heat release rate (HRR) at the times of onset of detonation development (left) and the peak detonation intensity (right) for a 2-D case with the most energetic length scale of temperature and turbulent field, l_T of 5 mm and l_e of 5 mm, respectively, the turbulent velocity fluctuation, u' of 83.3 m/s, and the ratio of the ignition delay time to turbulent time scale τ_{ig}/τ_t of 1.0 [61].

lead to mechanical failure [62]. The fundamental understanding of the developing super-knock mechanism and a reliable criterion to predict super-knock are needed to prevent destructive operation of combustion devices. High fidelity direct numerical simulations with the capability of fully resolving all temporal and spatial scales and the complex interaction of thermochemistry and turbulence will help address the knocking issues [60, 61]. With the aid of ParaView Catalyst, coupled with our DNS solver, an integrated framework has been set up and deployed, which allows us to capture, visualize and analyze in-situ the highly localized detonation events occurring at sub-nanosecond timescales, as representatively shown in Figures 9 and 10 for 2D simulations, and Figure 11 for a 3D simulation.

A systematically parametric set of two-dimensional (2D) and three-dimensional (3D) DNS simulations with and without turbulence fluctuations [60, 61] was conducted at the initial conditions temperature T_0 of 1200 K, pressure P_0 of 35 atm, and stoichiometric ethanol/air mixture (equivalence ratio ϕ_0 of 1.0). The corresponding homogeneous ignition delay time, τ^0_{ig}, and the equilibrium pressure, P_e, under the adiabatic constant-volume conditions are 75 μs, and 100 bar, respectively. Other relevant ideal one-dimensional detonation parameters associated with this initial condition are the Chapman–Jouguet pressure, P_{CJ} of 185 atm, von Neumann pressure, P_{VN} of 315 atm, and Chapman–Jouguet speed, V_{CJ} of 1836 m/s.

The computational domain, $L_{x,y}$ for 2D cases, was chosen to cover more than twenty eddies with the size of the integral length scale and at least four biggest eddy sizes. $L_{y,z}$ of the three-dimensional (3D) case with a most energetic length scale l_T of 5 mm was chosen to cover approximately ten integral length scales to reduce the computational cost. The 2D computational domain of 20.48×20.48 mm^2 was discretized by $10,240 \times 10,240$ grid points with a uniform spacing of 2 μm. This fine grid resolution was needed to resolve the detonation exothermic width. The 3D domain of $20.48 \times 10.24 \times 10.24$ mm^3 was discretized by $2,560 \times 1,280 \times 1,280$ grid points with a coarser resolution of 8 μm, comparable with the Kolmogorov length scale.

These representative 2D and 3D simulations (Figures 9–11) were performed with a high turbulent velocity fluctuation, u' of 83.3 m/s, and root-mean-square temperature fluctuation, T' of 15 K. Other relevant parameters are the most energetic length scale of temperature and turbulent field, l_T of 5 mm and l_e of 1 mm, respectively, and the ratio of the ignition delay time to turbulent time scale τ_{ig}/τ_t of 5.0 for the cases in Figures 10 and 11, and $l_T = l_e = 5$ mm and τ_{ig}/τ_t of 1.0 for the case in Figure 9. For a small $l_t < l_T$, high turbulence intensity together with short mixing time scale allows significant turbulence-chemistry interaction, revealing that high turbulence intensity can significantly attenuate the knock intensity as shown in Figures 10 and 11. The detonation wave speed, the von Neumann spike, and knock intensity were well reproduced by the 2D and 3D simulations, and the results were found to be quantitatively

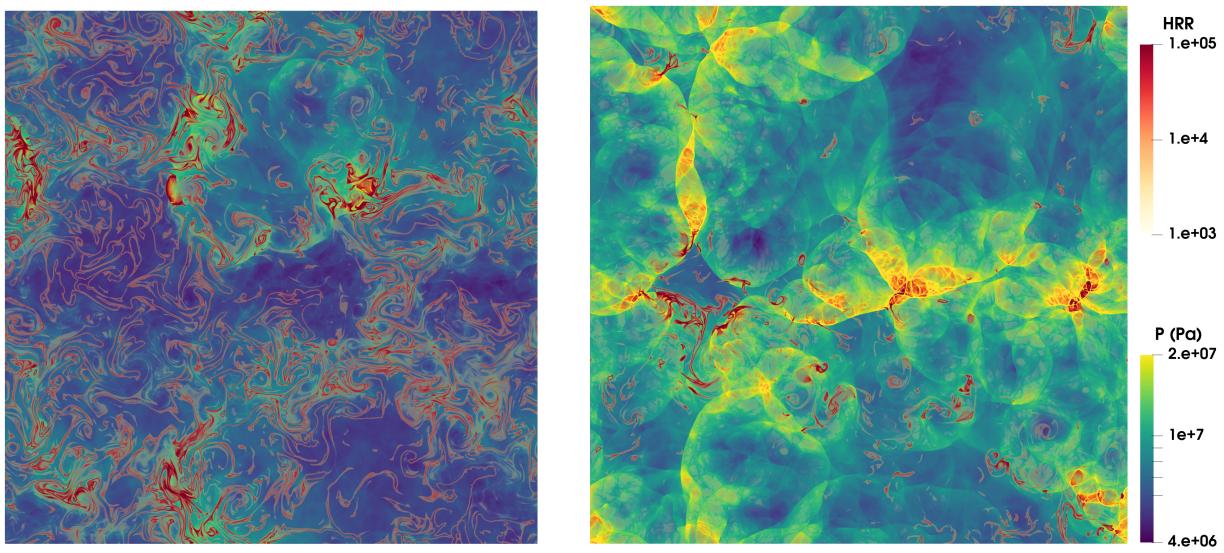

Figure 10: Two-dimensional contours of pressure and heat release rate (HRR) at the times of onset of detonation development (left) and the peak detonation intensity (right) for a 2-D case with the most energetic length scale of temperature and turbulent field, l_T of 5 mm and l_e of 1 mm, respectively, the turbulent velocity fluctuation, u' of 83.3 m/s, and the ratio of the ignition delay time to turbulent time scale τ_{ig}/τ_t of 5.0 [61].

in good agreement with the theoretical analysis (see Refs. [60, 61] for more details).

In the following section, we present the performance characteristics of KARFS, including weak scalability, strong scalability as well as GPU scalability, using both the CD8 and WENO7M schemes. Additionally, the performance of the operator splitting scheme using GPU acceleration is also presented using detailed chemical mechanisms of different sizes.

5. Performance characteristics

In this section, we study the performance characteristics of KARFS and its scalability using different block sizes. The primary system used in this work is the Titan supercomputer at the Oak Ridge Leadership Computing Facility (OLCF). Titan is a hybrid architecture Cray XK7 system, consisting of 18,688 compute nodes that are interconnected by Cray's Gemini network. Each node is composed of a 16-core AMD Opteron processor and an NVIDIA Tesla K20X graphics processing unit (GPU) as an accelerator. Moreover, each node has 32 GB of memory on the host Opteron processor and 6 GB memory on the GPU accelerator.

The particular benchmark test, herein referred to as the "Quiescent Test", involves integrating the governing equations (1a–1d) over 10 time steps. Preliminary scaling tests performed on a single node with 10, 50 and 100 time-steps per run revealed a difference of only 1% in the results. Hence, only the results obtained over 10 time steps have been presented here. The computational domain consists of a periodic cube containing air at a temperature of 800 K and an elevated pressure of 20 atm. Nevertheless, the description of chemical kinetics is specified using the 30 species dimethyl-ether mechanism [4] which consists of 175 reactions as well as 9 species identified as global quasi-steady state species. As such, the performance characteristics obtained herewith are representative of an actual production DNS run. The performance characteristics of the newly implemented WENO7M scheme are compared with those of the existing CD8 scheme. The time taken to advance the solution field over a known number of time steps is measured in the performance tests. It should be noted that the time reported here represents the total time to solution and not just the time taken by the advection operators. Specifically, the reported times consist of the time spent in the Runge–Kutta time loop including each of the six sub-stages, time spent for calculating the right-hand-side terms of the governing equations (1a–1d) including the viscous fluxes, the diffusive fluxes as well as the chemical source terms, plus the time taken to apply the de-aliasing filter. Only the I/O time has been excluded from the benchmark test as this can limit scalability. The chemical source terms are independent of the spatial derivative operators used

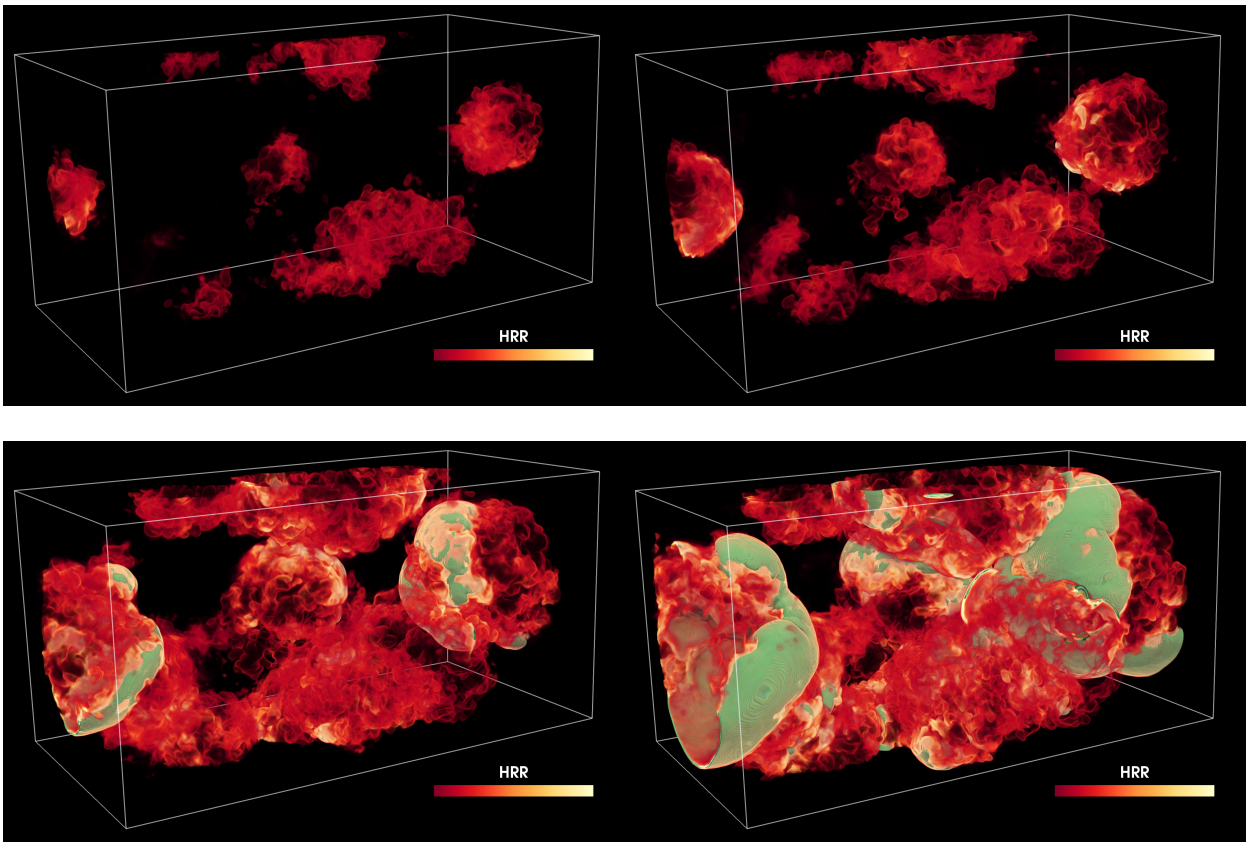

Figure 11: The volume rendering of HRR and the iso-surface of pressure (green) with $P = P_{CJ}$ at successive times prior to the onset of the detonation formation (top row) until the detonation intensity reaches its peak (bottom row) for a 3-D case with the most energetic length scale of temperature and turbulent field, l_T of 5 mm and l_e of 1 mm, respectively, the turbulent velocity fluctuation, u' of 83.3 m/s, and the ratio of the ignition delay time to turbulent time scale τ_{ig}/τ_t of 5.0 [61].

in the transport solvers. As such, GPU acceleration of the chemical source terms has been excluded from this part of the study and is considered separately in a later section. Note that, oftentimes, only the speedup obtained using the different parallelism strategies is reported while the computational cost in terms of node-time is rarely discussed. In the present study, the computational performance is quantified using a metric for the computational time per grid point per time step in terms of node-time as

$$\text{Cost} = \frac{\text{number of nodes} \times \text{wall clock time}}{\text{number of grid points} \times \text{number of time steps}}. \tag{27}$$

5.1. MPI weak scalability

An MPI weak scalability study is performed on up to 8788 nodes (140,608 cores) for two block sizes: 16^3 and 32^3. The two block sizes have been chosen such that they are neither too small for the high order discretization methods being investigated nor too big to mask the underlying deficiencies of the DNS code. To fully utilize all the processing cores on the node, 16 MPI tasks are placed on each node irrespective of the block size. We note that neither GPU nor OpenMP threading is utilized for weak scalability. Figure 12 shows the performance of KARFS while scaling from 4 nodes to 8788 nodes on the Titan system.

Irrespective of the block size, the WENO7M scheme is found to be about twice as expensive compared to the CD8 scheme. This is primarily because of the additional computations that are needed when using the WENO7M scheme including flux-splitting, interpolation and separate flux evaluations. It is observed that weak scaling is nearly perfect for the selected block sizes when the CD8 scheme is used for evaluating the spatial derivatives. The WENO7M

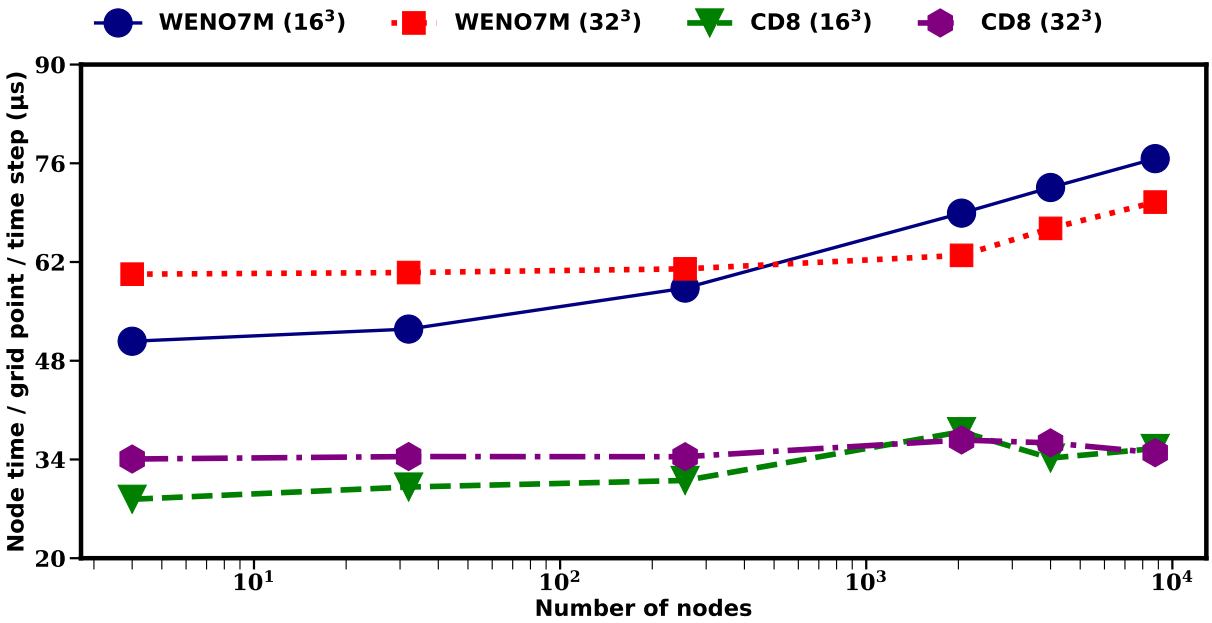

Figure 12: MPI weak scalability of KARFS. The block size per MPI as well as the discretization scheme for each case is indicated.

scheme also shows excellent weak scalability till around 2000 nodes for the larger block size of 32^3, beyond which its computational cost begins to increase linearly. The weak scalability of the WENO7M scheme deteriorates more rapidly for the smaller block size of 16^3. In each case, the reason for the observed rise in computational cost is twofold: 1) increasing the number of nodes increases the amount of time spent in MPI communications and 2) as discussed earlier in Section 3, extra time for MPI communications is added while evaluating the numerical fluxes at the left cell boundary when using the WENO7M scheme.

5.2. OpenMP scalability

In this section, the effects of multi-threading on the performance of KARFS is investigated by using both numerical discretization schemes (CD8 and WENO7M) for two block sizes: 16^3 and 32^3. For this part of the study, 64 nodes are utilized, each with 1 MPI task. Also, on each node, the number of threads per MPI task is varied from 1 to 16. Results are shown in Figure 13.

For the smaller block size of 16^3, multi-threading provides a substantial reduction in the computational cost for the WENO7M scheme when the number of OpenMP threads is increased from 1 to 2. A similar trend is observed for the CD8 scheme until the number of OpenMP threads is increased to 4. Further increase in the number of OpenMP threads degrades the performance of both schemes. This is mainly because of two reasons: 1) there is not enough work for each thread and 2) OpenMP thread creation and synchronization result in an additional time overhead. However, as the block size is increased to 32^3, an improved OpenMP scalability is observed while using either of the numerical schemes. The overall computational cost of each scheme is reduced by more than 67% when the number of OpenMP threads is increased from 1 to 8. A marginal degradation in OpenMP scalability is observed for each scheme when the number of OpenMP threads is increased from 8 to 16 due to the reasons stated above. In addition, the computational cost of the WENO7M scheme remains about twice that of the CD8 scheme, irrespective of the number of OpenMP threads, as has been already observed in the previous section.

It is also interesting to note that regardless of the numerical scheme or the number of OpenMP threads, the computational cost associated with MPI+OpenMP is higher than MPI only. To explain this, it is important to understand the difference between "Speedup" and "Cost." Table 1 shows a comparison of Speedup and Cost for the MPI+OpenMP and MPI-only strategies when using the CD8 scheme for a block size of 32^3. It can be seen that the total time to advance the solution field over a known number of time-steps is smaller for MPI+OpenMP, irrespective of the number

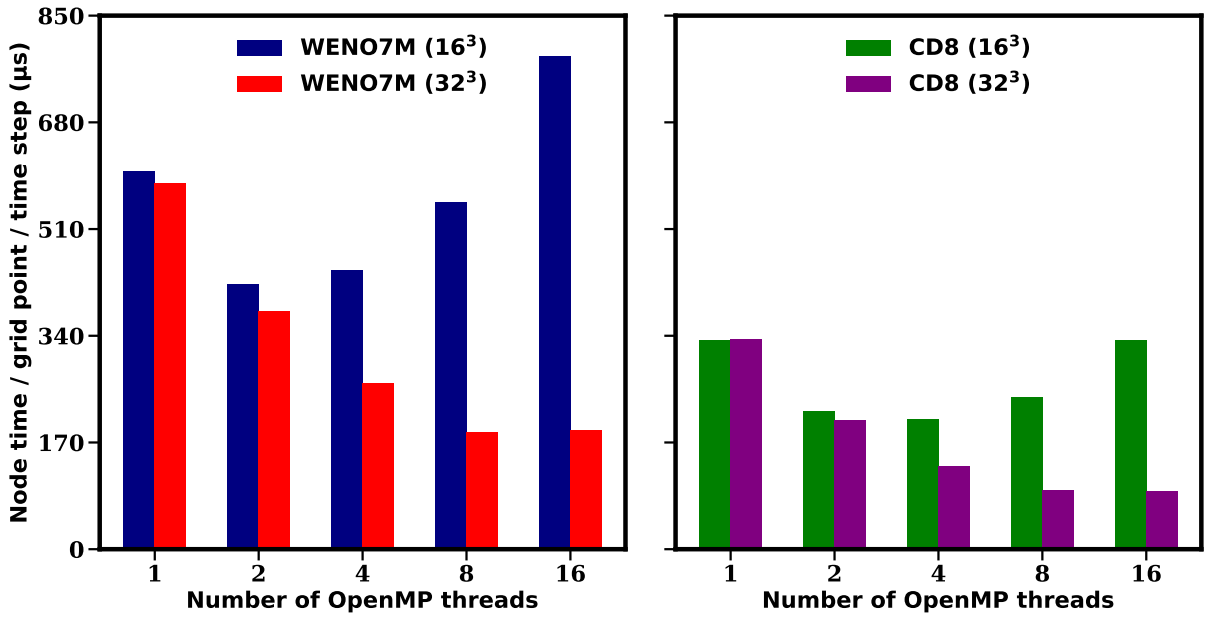

Figure 13: OpenMP scalability of KARFS. The block size per MPI as well as the discretization scheme for each case is indicated.

NX	NY	NZ	px	py	pz	MPI tasks	Nodes	OpenMP threads	Time (s)	Speedup	Cost
32	32	32	4	4	4	64	4	-	169.15	-	32.26
32	32	32	4	4	4	64	64	1	109.65	1.54	334.62
32	32	32	4	4	4	64	64	2	67.54	2.5	206.11
32	32	32	4	4	4	64	64	4	43.21	3.91	131.87
32	32	32	4	4	4	64	64	8	31	5.46	94.6
32	32	32	4	4	4	64	64	16	30.1	5.62	91.85

Table 2: Comparison of "speedup" and "cost" of MPI+OpenMP versus MPI only strategies when using the CD8 scheme for a block size of 32^3. Speedup = Time (MPI only) / Time (MPI+OpenMP).

of OpenMP threads, than the MPI only strategy. As a result, there is a definite "Speedup" in performance. However, since more nodes are utilized in the former, the "Cost" of MPI+OpenMP is significantly higher.

5.3. MPI strong scalability

Here, we investigate the strong scalability of KARFS with a fixed problem size while increasing the number of nodes in successive runs. Specifically, the problem size is kept constant at 256^3 with an initial block size of 128^3 partitioned into 8 MPI tasks on a single node. For subsequent runs, the block size is progressively halved by doubling the number of MPI tasks in each direction until it becomes 16^3. Note that neither GPU acceleration nor OpenMP threading is employed for this part of the study. As shown in Figure 14, the computational cost of both numerical schemes reduces considerably when the number of nodes is increased from 1 to 4. However, the reduction in computational cost for each numerical scheme is rather gradual as the number of nodes is further increased to 32 and subsequently to 256. This is mainly because, as the number of nodes is increased, the MPI communication overhead also increases. Moreover, with an increase in the number of nodes, the amount of work available for each node also decreases.

In addition, results obtained after converting the "Cost" (node time/grid point/time step) metric to "Time" metric are shown in Figure 15. This is done so as to clearly demonstrate excellent strong scaling of KARFS while using either of the two discretization schemes. The log-log plot for the data is very similar to what would be expected for ideal scaling: a straight line for strong scaling. As has been observed before, the computational cost of the WENO7M

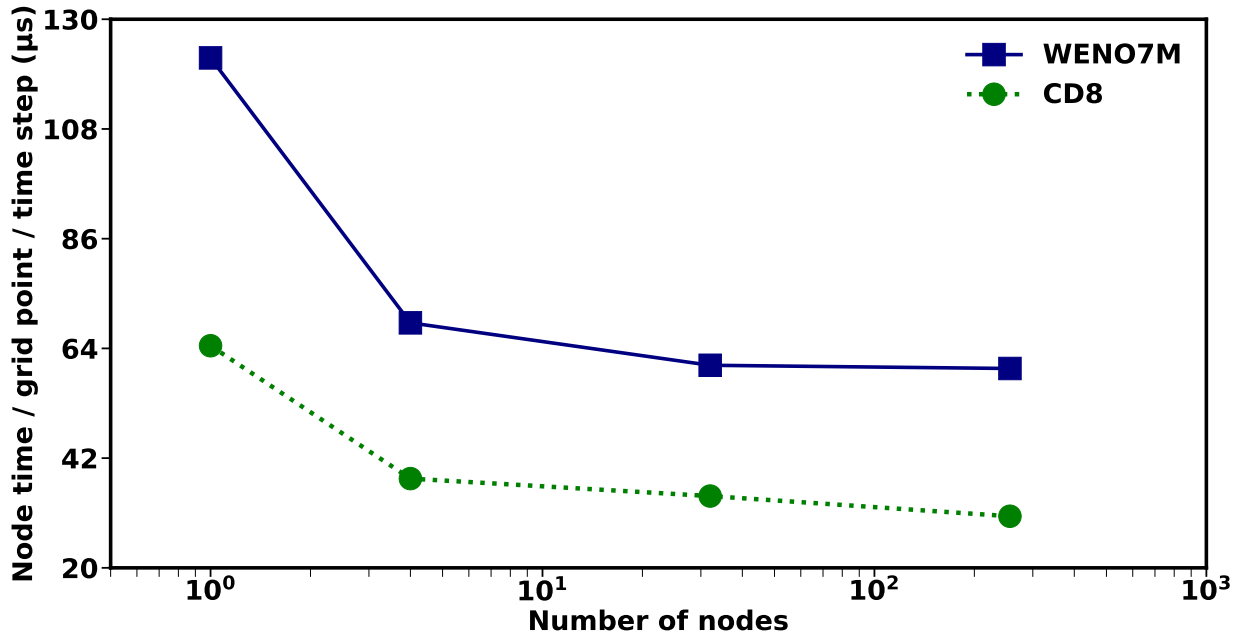

Figure 14: MPI strong scalability of KARFS using "Cost" metric.

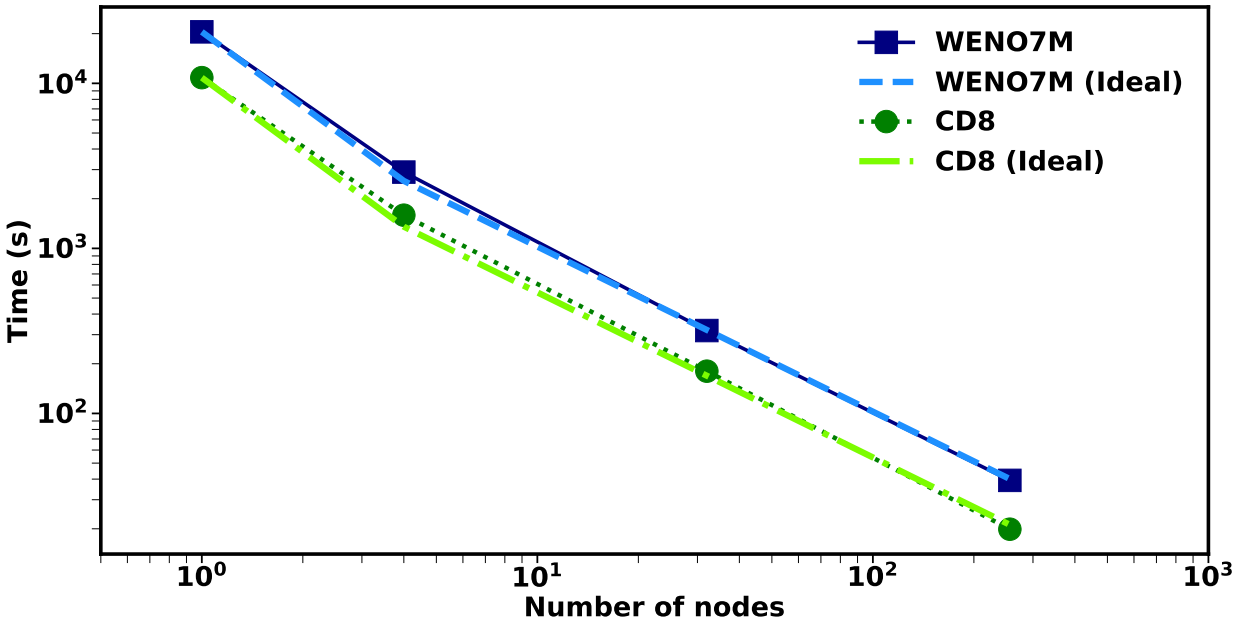

Figure 15: MPI strong scalability of KARFS using "Time" metric.

CD8						WENO7M					
Nodes	Cores	Time (s)	Speedup	S	P	Nodes	Cores	Time (s)	Speedup	S	P
1	8	10826.81	1	100.0	0.0	1	8	20509.78	1	100.0	0.0
4	64	1589.29	6.81	13.33	86.67	4	64	2899.25	7.07	12.78	87.22
32	512	180.32	60.04	1.47	98.53	32	512	317.80	64.54	1.36	98.64
256	4096	19.89	544.39	0.16	99.84	256	4096	39.30	521.84	0.17	99.83

Table 3: Strong scaling analysis based on Amdahl's law (1 node = 16 cores).

scheme is about twice that of the CD8 scheme. Note that domain decomposition with block sizes less than 16^3 was not possible since both CD8 and WENO7M operators used in KARFS require at least an 8-point and a 7-point stencil, respectively.

It is also well-known that strong scaling is limited by Amdahl's law [63] wherein the maximum "Speedup" achievable is limited by the serial part of the code. Amdahl's law can be formulated as

$$\text{Speedup} = \frac{1}{(S + \frac{P}{N})}. \tag{28}$$

In Equation 28, N represents the number of cores used, S represents the percentage of serial portion of KARFS while P represents the percentage of parallel portion of KARFS such that $S + P = 100$. Here, "Speedup" is defined based on the ratio of measured wall-clock time at the lowest core count to the actual measured time. For an obtained value of "Speedup", the values of S and P can be determined using Equation 28. Table 2 shows the speedup achieved in the strong scaling study using both the CD8 and WENO7M schemes. It can be noticed that, at the highest core count, more than 99% of KARFS has been parallelized. As such, almost the entire portion of KARFS is parallelizable and there is no lack of strong scalability. From the above table, it can also be seen that when the WENO7M scheme is used, the obtained speedup as well as the resulting percentage of parallel portion is higher when the number of cores is between 8 and 512. This is expected since the WENO7M scheme involves more operations compared to the CD8 scheme. Increasing the number of cores to 4096 results in marginal degradation of the WENO7M performance. This has also been observed in the weak scaling results discussed in section 5.1. It is mainly due to 1) an increased amount of time spent in MPI communication resulting from the increase in number of nodes/cores, 2) extra time for MPI communication being added while evaluating the numerical fluxes at the left cell boundary when using the WENO7M scheme.

5.4. GPU scalability

In this section, the performance of KARFS with GPU acceleration is compared against its OpenMP scalability by using four different block sizes: 16^3, 32^3, 64^3, and 96^3. Furthermore, to demonstrate KARFS portability, the benchmark runs are also performed on a developmental system at ORNL consisting of 8 compute nodes, each with two NVIDIA Tesla P100 accelerators. Each GPU accelerator has a memory of 16 GB. A number of combinations for the MPI+X hierarchical parallelism of KARFS, where X stands for OpenMP or CUDA backends, are tested and the respective computational cost for each discretization scheme is shown in Figures 16 and Fig. 17. Since the focus of this part of the study is to solely evaluate the performance of CD8 and WENO7M schemes with GPU acceleration, the chemistry has been frozen (i.e., the term $\dot{\omega}_k$ in equation 1d is not evaluated) for these bench-marking runs.

Figure 16 shows the performance of KARFS when using the CD8 numerical scheme. It is seen that for a block size of 16^3, the MPI+OpenMP parallelism strategy results in a lower computational cost than MPI+CUDA. CUDA has a start-up overhead and for a small block size of 16^3, the start-up overhead outweighs any gains from using the GPU. As the block size is increased to 32^3, a significant reduction in the computational cost is observed for the MPI+CUDA strategy compared to MPI+OpenMP. Moreover, the resulting computational cost with the Tesla P100 GPU is about half that of the Tesla K20X GPU. For the two larger block sizes of 64^3 and 96^3, there is no noticeable change in the computational cost with MPI+OpenMP. However, the computational cost decreases even further with MPI+CUDA. Note also that for the two smaller block sizes of 16^3 and 32^3, the computational cost of MPI+CUDA is significantly higher than the MPI-only strategy. These results demonstrate that domains with larger block sizes could be efficiently run on a significantly smaller number of nodes with GPUs.

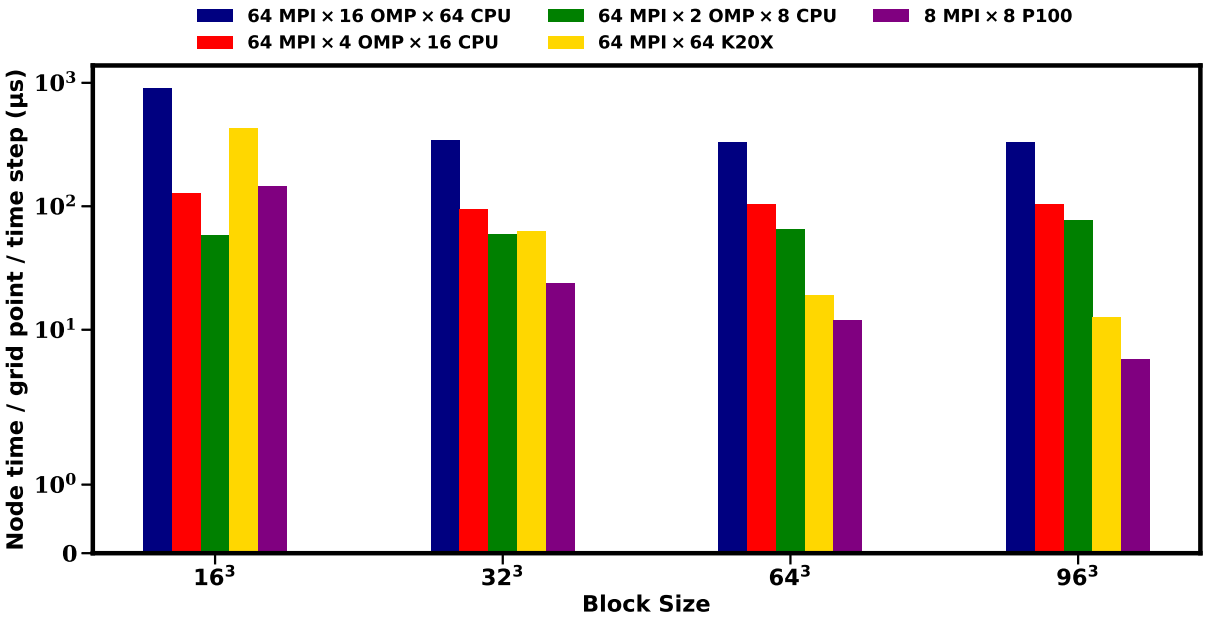

Figure 16: Performance of KARFS when using the CD8 numerical scheme. 1 CPU = 16 cores.

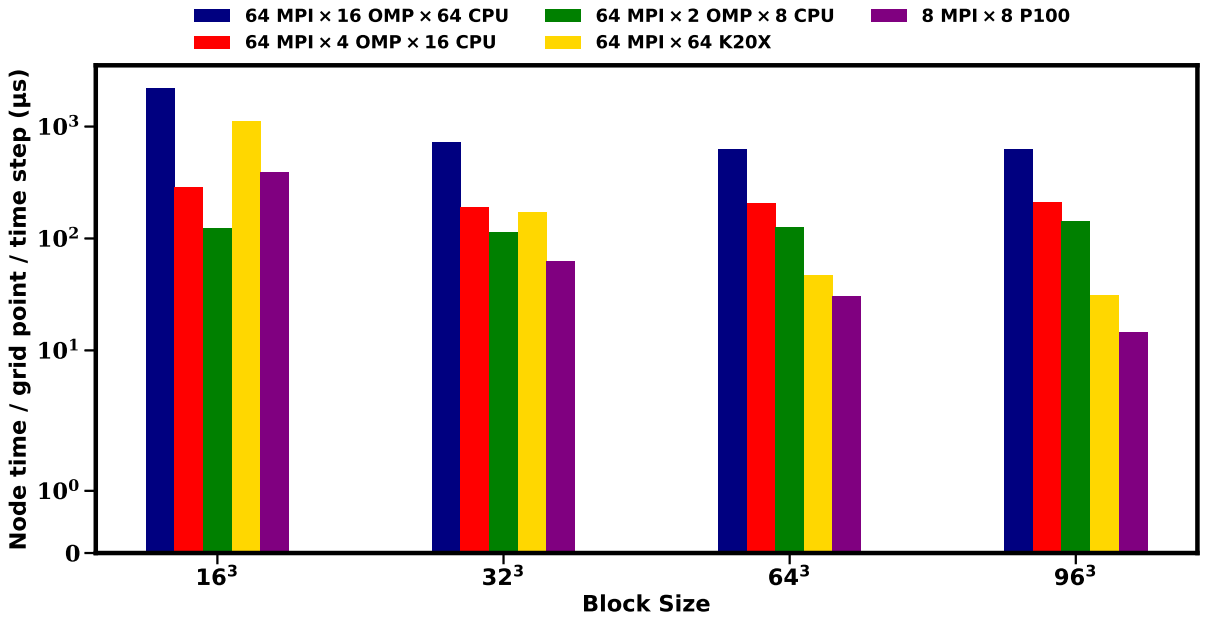

Figure 17: Performance of KARFS when using the WENO7M numerical scheme. 1 CPU = 16 cores.

The performance of KARFS when using the WENO7M is shown in Figure 17. The computational costs obtained for the different block sizes are qualitatively similar to those obtained using the CD8 scheme. Additionally, as has been observed in the previous sections, the respective costs for the WENO7M scheme are higher than those for the CD8 scheme.

5.5. Operator splitting scheme with GPU acceleration

GPU acceleration with the efficient stiff ODE solver and operator-splitting algorithm are tested using a zero-dimensional (0D) homogeneous reactor problem (autoignition) with various detailed chemical reaction mechanisms including one of the largest available in the community. The size of the mechanisms ranges from 55 species and 290 reactions to 7,171 species and 47,793 reactions [6, 64–69]. Computations of ignition delays have been conducted on an heterogeneous cluster at KAUST, Ibex, consisting of three families of processors (Intel, AMD, and NVIDIA). In particular, these runs are carried out on a node consisting of two Intel Xeon CPUs (16 cores each) and four NVIDIA Tesla P100 accelerators. Seven different reaction mechanisms are considered and the performance of KARFS is assessed employing one CPU core and one GPU. The results are summarized in Table 4. As the size of reaction mechanisms increases, a significant speedup can be achieved. Indeed, for the largest chemical mechanism (7,171 species and 47,793 reactions), a speedup of 63.41 is attained. Note, however, that for small reaction mechanisms the GPU does not provide any acceleration. Overall, it is evident that the use of the GPU in the linear-algebraic operations associated with operator splitting is effective to enhance the performance of KARFS when dealing with stiff and complex chemistry.

Mechanism size (Species/Reactions)	1CPU (s/step)	1CPU+1GPU (s/step)	Speedup
55 / 290	0.005	0.098	0.051
171 / 861	0.024	0.181	0.133
654 / 5,258	0.841	0.299	2.813
1,271 / 9,785	7.523	0.854	8.809
2,115 / 15,787	46.619	3.034	15.366
2,855 / 18,049	207.035	6.545	31.633
7,171 / 47,793	3677.000	37.849	63.410

Table 4: Cost of time-step evaluation and speedup for various sizes of chemical reaction mechanisms using one CPU and one GPU. (1 CPU equals 1 MPI and 1 OMP.)

In addition to the 0D ignition cases, multidimensional benchmarking tests are carried out, selecting three reaction mechanisms which are representative of small, medium and large sizes. These runs are conducted on the same node of Ibex. To assess the performance of KARFS, four different CPU/GPU combinations are considered for a one-dimensional (1D) reacting flow problem with three different grid sizes (4,800, 9,216, and 110,592 points). Noting that a rather minor speedup dependence on the grid size is observed and, for brevity, only the results associated with the largest grid size for each CPU/GPU combination and reaction mechanism are listed in Table 5. The speedup is reported with respect to a run on a single CPU using one MPI process. As in the 0D case, no acceleration is obtained for the small reaction mechanism (55 species and 290 reactions). On the other hand, for the medium (654 species and 5,258 reactions) and large (1,271 species and 9,785 reactions) reaction mechanisms, the utilization of the GPU leads to speedup factors of about 3 and 5.62, respectively. For the large reaction mechanism, the speedup obtained when using one MPI process with one OpenMP thread and one GPU is comparable with that of 16 MPI processes with one OpenMP thread.

Species/Reactions	55/290		654/5,258		1,271/9,785	
Case	Wall clock (s/step)	Speedup	Wall clock (s/step)	Speedup	Wall clock (s/step)	Speedup
1 MPI × 1 OMP	164.18	1.00	39,508.20	1.00	252,512.77	1.00
1 MPI × 1 OMP + 1 GPU	1,680.11	0.098	13,115.80	3.01	44,919.69	5.62
16 MPI × 1 OMP	13.41	12.24	3,112.82	12.69	42,714.58	5.91
16 MPI × 1 OMP + 1 GPU	774.17	0.21	2,506.22	15.76	6,829.69	36.97

Table 5: KARFS performance using different CPU/GPU combinations for three different sizes of chemical reaction mechanisms with the same number of grid points (110,592).

As a final performance test, the number of spatial dimensions is increased. A summary of results for 2D and 3D cases with the same reaction mechanisms considered for the 1D case is included in Table 6. Again, acceleration

of the computations is achieved for the medium and large mechanism sizes, which is consistent with both the 0D and 1D benchmark tests. Moreover, the speedup factors are similar to those obtained in the 1D case for each kinetics mechanism. The weak dependence of the level of GPU acceleration on the spatial dimensionality of the problem is due to the use of the GPU exclusively in the linear-algebraic operations associated with operator splitting, which depend on the size of the reaction mechanism. Considering that next-generation HPC platforms such as Summit at ORNL have multiple Volta GPUs (3) per CPU, interconnected using NVLink, the present benchmark results demonstrate that DNS of reacting flows with an unparalleled scale of complex and detailed chemistry will indeed be possible.

Finally, although the numerical schemes employed in the solution algorithm of KARFS are well established, their actual implementations into the reacting flow DNS on multiple GPU architectures along with a thorough performance assessment have not been reported in the past. As for the performance study, note that strong scaling in the MPI-everywhere implementation is limited as the number of grids per core is reduced to the stencil width. As such, the block sizes used in previous scaling studies have been much bigger [70, 71], making it difficult to assess the true HPC performance of the code. In addition, the strong scaling analysis based on Amdahl's law as presented in the current paper has not been demonstrated in previous studies.

	Two-dimensions (96^2)			Three-dimensions (48^3)		
Mechanism Size (Species/Reactions)	1 CPU (s/step)	1 CPU + 1 GPU (s/step)	Speedup	1 CPU (s/step)	1 CPU + 1 GPU (s/step)	Speedup
55 / 290	13.59	152.02	0.09	166.40	1,816.57	0.09
654 / 5,258	3,279.98	1,200.03	2.73	40,267.47	14,207.76	2.83
1,271 / 9,785	20,997.04	3,651.57	5.75	247,682.72	43,962.36	5.63

Table 6: Cost of time-step evaluation and speedup for various chemical reaction mechanisms in two-dimensional and three-dimensional domains. (1 CPU equals 1 MPI and 1 OMP.)

6. Conclusions

Using the MPI+X programming model, the implementation and validation of a seventh-order, minimally dissipative, mapped WENO (WENO7M) scheme and an efficient operator-splitting method in KARFS are demonstrated. The MPI+X programming model relies on Kokkos for "X" for performance portability to multi-core, many-core and GPUs. The capability and potential of the newly implemented WENO7M scheme in KARFS to perform DNS of compressible flows is also demonstrated with model problems involving shocks, isotropic turbulence, detonations and flame propagation into a stratified mixture with complex chemical kinetics. In addition, performance and scalability of KARFS using the two spatial discretization schemes (eighth-order central difference (CD8) and seventh-order, mapped WENO (WENO7M)) is evaluated to provide estimates that can be used for an appropriate load-balanced decomposition of future production simulations. The computational performance is quantified using a metric termed as "cost" for the computational time per grid point per time step so as to provide a better estimate of the computational resources (i.e. node/core hours).

Across the different scaling studies, it is revealed that the cost of WENO7M scheme is about two to three times higher than the CD8 scheme, mainly due to an increase in the amount of computations involved in the former. In the weak scaling studies, the code shows almost perfect scaling when using the CD8 scheme. Meanwhile, when the WENO7M scheme is used, the performance deteriorates with an increase in the number of nodes. An increase in the amount of time spent in MPI communications is found to be majorly responsible for such degradation in performance. In the strong scaling studies, the reduction in cost of both the numerical schemes is found to gradually subside with an increase in the number of nodes, in accordance with Amdahl's law.

A marked speedup is obtained when MPI+OpenMP strategy is used, especially for the larger block size. However, irrespective of the block size, the computational cost of MPI+OpenMP is found to be significantly higher as compared to MPI only strategy. This is due to an increase in the number of nodes being used at a given time with MPI+OpenMP strategy.

From the GPU scaling studies, it is found that block sizes larger than 32^3 are needed to overcome the startup overhead associated with MPI+CUDA strategy and subsequently achieve a reduction in computational cost of each

numerical scheme. Nonetheless, MPI+CUDA strategy with larger block sizes results in the usage of smaller number of nodes and also leads to a substantial reduction ($\approx$ 90%) in the computational cost of each numerical scheme in comparison to MPI+OpenMP strategy. With regards to GPU acceleration of the operator splitting scheme, a significant speedup is achieved, as the size of reaction mechanisms increases. It is also found that the dependence of speedup on the grid size is relatively minor since the GPU is exclusively used in the linear-algebraic operations associated with operator splitting.

Acknowledgments

This work was sponsored by competitive research funding from King Abdullah University of Science and Technology (KAUST). This research used resources of the computer clusters at KAUST Supercomputing Laboratory (KSL), the Oak Ridge Leadership Computing Facility and the Compute Data and Environment for Science (CADES) at ORNL, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

[1] M. Yao, Z. Zheng, H. Liu, Progress and recent trends in homogeneous charge compression ignition (HCCI) engines, Progress in Energy and Combustion Science 35 (5) (2009) 398 – 437.

[2] J. E. Dec, Advanced compression-ignition engines - understanding the in-cylinder processes, Proceedings of the Combustion Institute 32 (2) (2009) 2727 – 2742.

[3] A. K. Agarwal, A. P. Singh, R. K. Maurya, Evolution, challenges and path forward for low temperature combustion engines, Progress in Energy and Combustion Science 61 (2017) 1–56.

[4] A. Bhagatwala, Z. Luo, H. Shen, J. A. Sutton, T. Lu, J. H. Chen, Numerical and experimental investigation of turbulent DME jet flames, Proceedings of the Combustion Institute 35 (2) (2015) 1157 – 1166.

[5] S. Desai, R. Sankaran, H. G. Im, Unsteady deflagration speed of an auto-ignitive dimethyl-ether (DME)/air mixture at stratified conditions, Proceedings of the Combustion Institutedoi:https://doi.org/10.1016/j.proci.2018.09.019.

[6] M. B. Luong, Z. Luo, T. Lu, S. H. Chung, C. S. Yoo, Direct numerical simulations of the ignition of lean primary reference fuel/air mixtures with temperature inhomogeneities, Combust. Flame 160 (2013) 2038–2047.

[7] M. B. Luong, G. H. Yu, T. Lu, S. H. Chung, C. S. Yoo, Direct numerical simulations of ignition of a lean *n*-heptane/air mixture with temperature and composition inhomogeneities relevant to HCCI and SCCI combustion, Combust. Flame 162 (2015) 4566–4585.

[8] M. B. Luong, G. H. Yu, S. H. Chung, C. S. Yoo, Ignition of a lean PRF/air mixture under RCCI/SCCI conditions: A comparative DNS study, Proc. Combust. Inst. 36 (2017) 3623–3631.

[9] M. B. Luong, G. H. Yu, S. H. Chung, C. S. Yoo, Ignition of a lean PRF/air mixture under RCCI/SCCI conditions: Chemical aspects, Proc. Combust. Inst. 36 (2017) 3587–3596.

[10] M. B. Luong, R. Sankaran, G. H. Yu, S. H. Chung, C. S. Yoo, On the effect of injection timing on the ignition of lean PRF/air/EGR mixtures under direct dual fuel stratification conditions, Combust. Flame 183 (2017) 309–321.

[11] D. Assanis, S. W. Wagnon, M. S. Wooldridge, An experimental study of flame and autoignition interactions of iso-octane and air mixtures, Combustion and Flame 162 (4) (2015) 1214 – 1224.

[12] C. Strozzi, A. Mura, J. Sotton, M. Bellenoue, Experimental analysis of propagation regimes during the autoignition of a fully premixed methane−air mixture in the presence of temperature inhomogeneities, Combustion and Flame 159 (11) (2012) 3323 – 3341.

[13] T. Urushihara, K. Yamaguchi, K. Yoshizawa, T. Itoh, A study of a gasoline-fueled compression ignition engine: expansion of HCCI operation range using SI combustion as a trigger of compression ignition , SAE transactions 114 (3) (2005) 419–425.

[14] X. Ma, Z. Wang, C. Jiang, Y. Jiang, H. Xu, J. Wang, An optical study of in-cylinder CH2O and OH chemiluminescence in flame-induced reaction front propagation using high speed imaging, Fuel 134 (Supplement C) (2014) 603 – 610.

[15] G. Bansal, H. G. Im, Autoignition and front propagation in low temperature combustion engine environments, Combustion and Flame 158 (11) (2011) 2105 – 2112.

[16] S. Gupta, H. G. Im, M. Valorani, Classification of ignition regimes in HCCI combustion using computational singular perturbation, Proceedings of the Combustion Institute 33 (2) (2011) 2991 – 2999.

[17] S. Gupta, H. G. Im, M. Valorani, Analysis of n-heptane auto-ignition characteristics using computational singular perturbation, Proceedings of the Combustion Institute 34 (1) (2013) 1125 – 1133.

[18] P. Pal, M. Valorani, P. G. Arias, H. G. Im, M. S. Wooldridge, P. P. Ciottoli, R. M. Galassi, Computational characterization of ignition regimes in a syngas/air mixture with temperature fluctuations, Proc. Combust. Inst. 36 (2017) 3705–3716.

[19] M. B. Luong, F. E. Hernández Pérez, H. G. Im, Prediction of ignition modes of NTC-fuel/air mixtures with temperature and concentration fluctuations, Combust. Flame 213 (2020) 382–393.

[20] G. H. Yu, M. B. Luong, S. H. Chung, C. S. Yoo, Ignition characteristics of a temporally evolving *n*-heptane jet in an *iso*-octane/air stream under RCCI combustion-relevant conditions,, Combust. Flame 208 (2019) 299–312.

[21] S. Desai, R. Sankaran, H. G. Im, Auto-ignitive deflagration speed of methane (ch4) blended dimethyl-ether (dme)/air mixtures at stratified conditions, Combustion and Flame 211 (2020) 377–391.

[22] L. Vervisch, T. Poinsot, Direct numerical simulation of non-premixed turbulent flames, Annual review of fluid mechanics 30 (1) (1998) 655–691.

[23] R. Sankaran, H. G. Im, E. R. Hawkes, J. H. Chen, The effects of non-uniform temperature distribution on the ignition of a lean homogeneous hydrogen air mixture, Proceedings of the Combustion Institute 30 (1) (2005) 875 – 882.
[24] C. A. Kennedy, M. H. Carpenter, Several new numerical methods for compressible shear-layer simulations, Applied Numerical Mathematics 14 (4) (1994) 397 – 433.
[25] C. A. Kennedy, M. H. Carpenter, R. M. Lewis, Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations, Appl. Numer. Math. 35 (3) (2000) 177–219.
[26] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, Journal of computational physics 115 (1) (1994) 200–212.
[27] S. Pirozzoli, Numerical methods for high-speed flows, Annual review of fluid mechanics 43 (2011) 163–194.
[28] A. Mosedale, D. Drikakis, Assessment of Very High Order of Accuracy in Implicit LES models, Journal of Fluids Engineering 129 (12) (2007) 1497–1503. doi:10.1115/1.2801374.
[29] K. Ritos, I. W. Kokkinakis, D. Drikakis, Physical insight into the accuracy of finely-resolved iLES in turbulent boundary layers, Computers & Fluids 169 (2018) 309–316.
[30] K. Ritos, I. W. Kokkinakis, D. Drikakis, Performance of high-order implicit large eddy simulations, Computers & Fluids 173 (2018) 307–312.
[31] K. E. Niemeyer, C.-J. Sung, Recent progress and challenges in exploiting graphics processors in computational fluid dynamics, The Journal of Supercomputing 67 (2) (2014) 528–564.
[32] K. Spafford, J. Meredith, J. Vetter, J. Chen, R. Grout, R. Sankaran, Accelerating S3D: a GPGPU case study, in: European Conference on Parallel Processing, Springer, 2009, pp. 122–131.
[33] Y. Shi, W. H. Green Jr, H.-W. Wong, O. O. Oluwole, Redesigning combustion modeling algorithms for the graphics processing unit (GPU): Chemical kinetic rate evaluation and ordinary differential equation integration, Combustion and Flame 158 (5) (2011) 836–847.
[34] C. Stone, R. Davis, Techniques for solving stiff chemical kinetics on GPUs, in: 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2013, p. 369.
[35] K. E. Niemeyer, C.-J. Sung, Accelerating moderately stiff chemical kinetics in reactive-flow simulations using gpus, Journal of Computational Physics 256 (2014) 854–871.
[36] 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, Computers & Fluids 173 (2018) 73–79.
[37] H. C. Edwards, D. Sunderland, V. Porter, C. Amsler, S. Mish, Manycore performance-portability: Kokkos multidimensional array library, Scientific Programming 20 (2) (2012) 89–114.
[38] H. C. Edwards, C. R. Trott, D. Sunderland, Kokkos: Enabling manycore performance portability through polymorphic memory access patterns, Journal of Parallel and Distributed Computing 74 (12) (2014) 3202–3216.
[39] A. K. Henrick, T. D. Aslam, J. M. Powers, Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points, Journal of Computational Physics 207 (2) (2005) 542–567.
[40] S. Tomov, J. Dongarra, M. Baboulin, Towards dense linear algebra for hybrid GPU accelerated manycore systems, Parallel Computing 36 (5-6) (2010) 232–240.
[41] S. Zhao, N. Lardjane, I. Fedioun, Comparison of improved finite-difference WENO schemes for the implicit large eddy simulation of turbulent non-reacting and reacting high-speed shear flows, Computers & Fluids 95 (2014) 74–87.
[42] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, Journal of computational physics 126 (1) (1996) 202–228.
[43] A. K. Henrick, T. D. Aslam, J. M. Powers, Simulations of pulsating one-dimensional detonations with true fifth order accuracy, Journal of Computational Physics 213 (1) (2006) 311–329.
[44] R. Borges, M. Carmona, B. Costa, W. S. Don, An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws, Journal of Computational Physics 227 (6) (2008) 3191–3211.
[45] M. P. Martín, E. M. Taylor, M. Wu, V. G. Weirs, A bandwidth-optimized WENO scheme for the effective direct numerical simulation of compressible turbulence, Journal of Computational Physics 220 (1) (2006) 270–289.
[46] D. G. Goodwin, H. K. Moffat, R. L. Speth, Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes, Version 2.2.1.
[47] T. Poinsot, S. Lele, Boundary conditions for direct simulations of compressible viscous flows, Journal of Computational Physics 101 (1) (1992) 104–129.
[48] C. S. Yoo, H. G. Im, Characteristic boundary conditions for simulations of compressible reacting flows with multi-dimensional, viscous and reaction effects, Combustion Theory and Modelling 11 (2007) 259–286.
[49] O. San, K. Kara, Evaluation of Riemann flux solvers for WENO reconstruction schemes: Kelvin–Helmholtz instability, Computers & Fluids 117 (2015) 24–41.
[50] G. Strang, On the construction and comparison of difference schemes, SIAM Journal on Numerical Analysis 5 (3) (1968) 506–517.
[51] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, C. S. Woodward, SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers, ACM Transactions on Mathematical Software 31 (3) (2005) 363–396.
[52] G. A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, Journal of computational physics 27 (1) (1978) 1–31.
[53] T. Passot, A. Pouquet, Numerical simulation of compressible homogeneous flows in the turbulent regime, Journal of Fluid Mechanics 181 (1987) 441–466.
[54] A. Sow, B. J. Lee, F. E. Hernández Pérez, H. G. Im, Detonation onset in a thermally stratified constant volume reactor, Proceedings of the Combustion Institute 37 (3) (2019) 3529–3536.
[55] M. P. Burke, M. Chaos, Y. Ju, F. L. Dryer, S. J. Klippenstein, Comprehensive H2/O2 kinetic model for high-pressure combustion, International Journal of Chemical Kinetics 44 (7) (2012) 444–474.
[56] D. L. Reuss, T.-W. Kuo, G. Silvas, V. Natarajan, V. Sick, Experimental metrics for identifying origins of combustion variability during spark-assisted compression ignition, International Journal of Engine Research 9 (5) (2008) 409–434.
[57] P. Domingo, L. Vervisch, Triple flames and partially premixed combustion in autoignition of non-premixed turbulent mixtures, in: Symposium (International) on Combustion, Vol. 26, Elsevier, 1996, pp. 233–240.

[58] A. Bhagatwala, J. H. Chen, T. Lu, Direct numerical simulations of HCCI/SACI with ethanol, Combustion and Flame 161 (7) (2014) 1826–1841.
[59] A. Bhagatwala, R. Sankaran, S. Kokjohn, J. H. Chen, Numerical investigation of spontaneous flame propagation under RCCI conditions, Combustion and Flame 162 (9) (2015) 3412 – 3426.
[60] M. B. Luong, S. Desai, F. E. Hernández Pérez, R. Sankaran, B. Johansson, H. G. Im, A statistical analysis of developing knock intensity in a mixture with temperature inhomogeneities, Proc. Combust. Inst. 37.
[61] M. B. Luong, S. Desai, F. E. Hernández Pérez, R. Sankaran, B. Johansson, H. G. Im, Effects of turbulence and temperature fluctuations on knock development in an ethanol/air mixture, Flow Turbul. Combust.doi:https://doi.org/10.1007/s10494-020-00189-z.
[62] Z. Wang, H. Liu, R. D. Reitz, Knocking combustion in spark-ignition engines, Prog. Energy Combust. Sci. 61 (2017) 78–112.
[63] G. M. Amdahl, Validity of the single processor approach to achieving large scale computing capabilities, in: Proceedings of the April 18-20, 1967, spring joint computer conference, ACM, 1967, pp. 483–485.
[64] Z. Zhao, M. Chaos, A. Kazakov, F. Dryer, Thermal decomposition reaction and a comprehensive kinetic model of dimethyl ether, Int. J. Chem. Kinet 40 (2008) 1–18.
[65] M. Mehl, W. J. Pitz, C. K. Westbrook, H. J. Curran, Kinetic modeling of gasoline surrogate components and mixtures under engine conditions, Proc. Combust. Inst. 33 (2012) 193–200.
[66] H. J. Curran, P. Gaffuri, W. J. Pitz, C. K. Westbrook, A comprehensive modeling study of iso-octane oxidation, Combust. Flame 129 (2002) 253–280.
[67] C. K. Westbrook, W. J. Pitz, O. Herbinet, H. J. Curran, E. J. Silke, A comprehensive detailed chemical kinetic reaction mechanism for combustion of n-alkane hydrocarbons from n-octane to n-hexadecane, Combust. Flame 156 (2008) 181–199,.
[68] Y. Pei, M. Mehl, W. Liu, T. Lu, W. J. Pitz, S. Som, A multi-component blend as a diesel fuel surrogate for compression ignition engine applications, Journal of Engineering for Gas Turbines and Power (2015) GTP–15–1057.
[69] S. M. Sarathy, C. K. Westbrook, M. Mehl, W. J. Pitz, C. Togbe, P. Dagaut, H. Wang, M. A. Oehlschlaeger, U. Niemann, K. Seshadri, P. S. Veloo, C. Ji, F. N. Egolfopoulos, T. Lu, Comprehensive chemical kinetic modeling of the oxidation of 2-methylalkanes from C7 to C20, Combust. Flame 158 (2011) 2338–2357.
[70] J. M. Levesque, R. Sankaran, R. Grout, Hybridizing S3D into an exascale application using OpenACC: an approach for moving to multi-petaflops and beyond, in: Proceedings of the International conference on high performance computing, networking, storage and analysis, IEEE Computer Society Press, 2012, p. 15.
[71] I. Bermejo-Moreno, J. Bodart, J. Larsson, Scaling compressible flow solvers on the IBM Blue Gene/Q platform on up to 1.97 million cores, Annual Research Briefs, Stanford Center for Turbulence Research.