A HYBRID FILTERED DENSITY FUNCTION SPECTRAL-ELEMENT LARGE EDDY SIMULATOR

by

Krisda Tapracharoen

B.E. in Mehcanical Engineering, King Mongkut's University of
Technology North Bangkok, Thailand, 2011
M.S. in Computational Design and Manufacturing, Carnegie Mellon
University, Pittsburgh, 2014

Submitted to the Graduate Faculty of
the Swanson School of Engineering in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

University of Pittsburgh
2018

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Krisda Tapracharoen

It was defended on

July 16, 2018

and approved by

Peyman Givi, Ph.D., Distinguished Professor, and James T. MacLeod Professor

Hessam Babaee, Ph.D., Assistant Professor

Sangyeop Lee, Ph.D., Assistant Professor

Satbir Singh, Ph.D., Associate Professor, Department of Mechanical Engineering, Carnegie Mellon University

Dissertation Advisors: Peyman Givi, Ph.D., Distinguished Professor, and James T. MacLeod Professor,

Shervin Sammak, Ph.D., Research Assistant Professor, Center for Research Computing

A HYBRID FILTERED DENSITY FUNCTION SPECTRAL-ELEMENT LARGE EDDY SIMULATOR

Krisda Tapracharoen, PhD

University of Pittsburgh, 2018

This dissertation is focused on the development of a hybrid scheme, combining the spectral-element method (SEM) solver with a Lagrangian Monte Carlo (MC) filtered density function (FDF) simulator for large eddy simulation (LES) of turbulent flows. The methodology is shown to be suitable for LES, as a larger portion of the resolved energy is captured as the order of spectral approximation increases. The consistency and the overall performance of the SEM-MC solver, and the realizability of the simulated results are demonstrated via LES of a temporally developing mixing layer under both non-reacting and reacting conditions. The effects of grid resolution and polynomial order ($h-p$ refinement), and ensemble average size (Δ_E) are studied. The computational scheme is fully parallelized via the message passing interface (MPI) methodology.

Keywords: large eddy simulation; filtered density function; Monte Carlo methods; spectral-element method.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I would like to express my sincere appreciation to my advisor, Professor Peyman Givi for his support and guidance through the course of my graduate studies; and also to my co-advisor, Dr. Shervin Sammak. I would like to thank the members of my doctoral committee, Professors Hessam Babaee, Sangyeop Lee and Satbir Singh.

I am indebted to my colleagues and friends whom I had the pleasure to work with during my time at the University of Pittsburgh: Mr. Cajon Gonzales, Mr. Medet Inkarbekov, Mr. Aidyn Aitzhan, Ms. Ling Miao, and Dr. Arash Nouri Gheimassi. My deepest gratitude goes to my parents; for their love and support.

This work is sponsored primarily by Royal Thai government. Additional support is provided by the U.S. National Science Foundation (NSF) under Grants CBET-1609120 and CBET-1603131. Computational resources are provided by the University of Pittsburgh Center for Research Computing.

KRISDA TAPRACHAROEN
UNIVERSITY OF PITTSBURGH, 2018

1.0 INTRODUCTION

Modeling and simulation of turbulent flow have been a challenging task for over a century [2]. Direct numerical simulation (DNS) is one of the methods in which the Navier-Stokes equations are solved directly without the use of turbulence models [3]. However, turbulence has spatial scales that range from the very large, energy containing to the small, dissipating, Kolmogorov scale. The ratio of these scales is directly proportional to the Reynolds number. As the Reynolds number is of the order of thousands for practical problems, the range of scales is enormous. For an accurate solution, the smallest grid spacing needs to be approximately the same as the Kolmogorov scale so that the dissipation is captured accurately [4]. The accuracy by which the scales are represented also depends on the numerical method. A low order scheme requires a more significant number of grid points than a higher order scheme to approximate the solution with the same accuracy.

Large eddy simulation (LES) is one of the ways to alleviate the computational cost by filtering the transport equations [5]. Effectively, LES captures the dominant large scale behavior of the flow and models the effects of the small scale eddies. In LES, the large scales which are, anisotropic, are sensitive to boundary conditions and are computed directly. The small scales are more isotropic, universal, and are modeled. Modeling of the small scales reduces the computational cost, while the computation of the large scales provides detailed flow-field information. In LES of reactive flows, modeling of chemical reaction imposes an extra burden. There have been many attempts to model the effects of turbulence-chemistry interactions [6–10].

Probability density function (PDF) method for solving the Reynolds-averaged Navier-Stokes (RANS) equations has been very effective in turbulent reactive flows [11, 12]. In the early stage of the development, the shape of PDF is parametrized using a few parameters

[13]. The most popular distribution is the so-called β-function [14]. When more than one parameter is to be considered, constructing a multi-dimensional PDF becomes more difficult [15]. Pope [16] developed the PDF transport from its evolution equation. In this method, the flow is represented by the joint PDF of velocities and scalars. The mean velocity and Reynolds stresses can be calculated from first and second moments respectively. Givi [17] and Pope [18] introduced the filtered density function (FDF) to account for the effects of subgrid scale scalars. The primary advantage of FDF is that it provides a closure for the effect of the chemical reaction. To deal with variable density flow, the filtered mass density function (FMDF) [19] is developed. The most sophisticated FDF closure available to-date is pressure-energy-velocity-composition filtered mass density function [20]. FDF has proven to be very effective for LES reactive turbulent flow [21–33]. For its numerical simulation, the FDF is represented by an ensemble of Monte Carlo (MC) particles.

Numerical schemes determine the fidelity of LES. Numerical errors in space and time can smear the solutions as well as dissipate turbulence which leads to inaccurate results. Ghosal [34] shows that for low-order finite difference schemes, the truncation error may be more significant than modeled sub-grid stresses unless the filter width is significantly larger than the grid size. Kravchenko and Moin [35] show that for low-order finite difference schemes (2^{nd} and 3^{rd} order), the energy spectrum is heavily distorted by truncation errors, and the contribution from the sub-grid model becomes negligible. They also show that low-order schemes have large dispersion errors, and reduce the temporal accuracy.

Finite element methods (FEM) have been effective for turbulent flow simulations [36]. The essential components of this method are the trial and the test functions. The trial functions are used as the basis functions for a truncated series expansion of the solution. The test functions are used to ensure that the solutions are satisfied as closely as possible by the truncated series expansion. The error in the solution produced by using the truncated expansion instead of the exact solution is called the residual. The coefficients in the trial function are computed by minimizing the residual. The choice of the trial function is one of the features that distinguish various numerical schemes. The trial functions for FEM are low-order local functions [37]. In FEM, the domain is divided into small elements, and a low-order trial function is specified within each element. Thus, it is well suited for dealing with complex geometries. In Galerkin-FEM, the test functions are the same as the trial ones [38], while in collocation methods, the test functions are the Dirac delta function centered at the collocation points. The main difference between FEM and spectral method is that the latter uses trial functions that are non-zero over the whole domain, while FEM uses trial functions that are non-zero only on the sub-domains. For the spectral accuracy, the residual decays faster than any inverse power of the highest retained mode provided that both the function and all its derivatives are smooth [39]. The major drawback for the spectral method is that the approximation of the flow variables with one polynomial dictates a simple geometry. The spectral element method (SEM) is introduced by Patera [40]. It combines the accuracy of single domain spectral schemes with the flexibility of FEM. In SEM, the spatial resolution can be altered either by increasing number of an element (h-refinement) or by increasing the polynomial order within the elements (p-refinement) [41]. It combines many features including high-order approximation, robustness, and easiness to implement in parallel.

The FDF needs to be combined with a base flow solver. There are many attempts to combine FDF with different numerical schemes, such as finite difference and finite volume [42–50]. Despite its popularity, a major challenge associated with FDF is its implementation on high-order numerical schemes. Sammak *et al.* [51, 52] combine high-order Eulerian discontinuous Galerkin (DG) with an MC solution of the FDF. Another challenge associated with combining high-order numerical scheme and MC solver is the amount of its computational requirements [53]. Message passing interface (MPI) is the standard of exchanging messages between multiple processors running a parallel program across distributed memory [54]. For SEM, the domain is split up into smaller sub-domains, and are allocated to different processors.

The objective of this dissertation is to assess the capability of the SEM-MC solver for LES of reactive flow at low Mach numbers. The consistency of the solver is then evaluated by comparing the moment of FDF with those obtained by SEM. The effects of polynomial degree and grid size are studied. In the reacting case, the consistency of the FDF is investigated by considering the compositional structure in the mixture fraction domain. This dissertation is organized as follows. First, all of the governing equations are presented in Chapter 2. Then, the numerical schemes for solving the base flow and the FDF are given in Chapter 3. All the simulation results are presented in Chapter 4. Finally, some concluding remarks are finished in Chapter 5.

2.0 FORMULATION

2.1 BASIC EQUATIONS

The governing equations for variable density, viscous, reactive flow are the conservation of mass, momentum, energy and species. The transport variables are the density $\rho(\mathbf{x}, t)$, velocity vector $u_i(\mathbf{x}, t)$, pressure $p(\mathbf{x}, t)$, total energy $E(\mathbf{x}, t)$ and species mass fraction $\phi_\alpha(\mathbf{x}, t)$ $(\alpha = 1, 2, ..., N_S)$. The conservation equations are:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} = 0, \tag{2.1}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \left(\rho u_i u_j + p\delta_{ij}\right)}{\partial x_j} = \frac{\partial \sigma_{ij}}{\partial x_j}, \tag{2.2}$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial \left(\rho E u_j + p u_j\right)}{\partial x_j} = -\frac{\partial q_j}{\partial x_i} + \frac{\partial \sigma_{ij} u_i}{\partial x_j}, \tag{2.3}$$

$$\frac{\partial \rho \phi_\alpha}{\partial t} + \frac{\partial \rho \phi_\alpha u_j}{\partial x_j} = -\frac{\partial J_j^\alpha}{\partial x_j} + \rho S_\alpha, \quad \alpha = 1, 2, ..., N_S. \tag{2.4}$$

The chemical reaction source terms $S_\alpha(\phi(\mathbf{x}, t))$ are function of all of a scalars. The total energy, the viscous stress tensor, the heat flux and the scalar fluxes are:

$$E = e + \frac{1}{2} u_k u_k, \tag{2.5}$$

$$\sigma_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \tag{2.6}$$

$$q_j = -\lambda \frac{\partial T}{\partial x_j}, \tag{2.7}$$

$$J_j^\alpha = -\rho \Gamma_\alpha \frac{\partial \phi_\alpha}{\partial x_i}. \tag{2.8}$$

where μ is the dynamic viscosity, λ is the thermal conductivity, e is the internal energy, γ is the specific heat ratio and Γ_α is the mass diffusion coefficient of species α. the Schmidt number ($Sc = \frac{\mu}{\rho \Gamma_\alpha}$) is assumed to be unity $\rho \Gamma_\alpha = \mu$. Equation of state is assumed to obey the ideal gas law:

$$p = (\gamma - 1)\rho e. \tag{2.9}$$

The conservation equations can be written in the matrix form:

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}_i^a}{\partial x_i} - \frac{\partial \mathbf{F}_i^v}{\partial x_i} = \mathbf{S}, \tag{2.10}$$

where

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \\ \rho \phi_\alpha \end{pmatrix}, \mathbf{F}_i^a = \begin{pmatrix} \rho u_i \\ \rho u_1 u_i + p\delta_{i1} \\ \rho u_2 u_i + p\delta_{i2} \\ \rho u_3 u_i + p\delta_{i3} \\ \rho \left(\rho E + p\right) u_i \\ \rho \phi_\alpha u_i \end{pmatrix}, \mathbf{F}_i^v = \begin{pmatrix} 0 \\ \sigma_{i1} \\ \sigma_{i2} \\ \sigma_{i3} \\ -q_i + u_k \sigma_{ik} \\ \mu \frac{\partial \phi_\alpha}{\partial x_i} \end{pmatrix}, \mathbf{S} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \rho S_\alpha \end{pmatrix}. \tag{2.11}$$

$\mathbf{Q}$ is the vector of the conserved variables, $\mathbf{F}_i^a$ is the advective flux vector, $\mathbf{F}_i^v$ is the viscous flux vector, and $\mathbf{S}$ is the source vector. These fluxes have to be separated to advective and viscous fluxes because advective fluxes contained just first order derivative but viscous fluxes contain higher-order.

2.2 FILTERED TRANSPORT EQUATIONS

In LES, the transport equations are filltered using spatial low-pass [55]:

$$\langle Q(\mathbf{x},t)\rangle_\ell = \int_{-\infty}^{+\infty} Q(\mathbf{x}',t)G(\mathbf{x}',\mathbf{x})d\mathbf{x}', \tag{2.12}$$

where $G(\mathbf{x}',\mathbf{x}) \equiv G\left(\mathbf{x}'-\mathbf{x}\right)$ denotes a filter function, and $\langle Q(\mathbf{x},t)\rangle_\ell$ is the filtered value of the transport variable $Q(\mathbf{x},t)$. In variable-density flows, it is convenient to use the Favré-filtered quantity:

$$\langle Q(\mathbf{x},t)\rangle_L = \frac{\langle \rho Q\rangle_\ell}{\langle \rho\rangle_\ell}. \tag{2.13}$$

Application of this filter yields:

$$\frac{\partial \langle \rho\rangle_\ell}{\partial t} + \frac{\partial \langle \rho\rangle_\ell \langle u_j\rangle_L}{\partial x_j} = 0, \tag{2.14}$$

$$\frac{\partial \langle \rho\rangle_\ell \langle u_i\rangle_L}{\partial t} + \frac{\partial \left(\langle \rho\rangle_\ell \langle u_j\rangle_L \langle u_i\rangle_L + \langle p\rangle_\ell \delta_{ij}\right)}{\partial x_j} = \frac{\partial \langle \sigma_{ij}\rangle_L}{\partial x_j} - \frac{\partial \tau_{ij}^{sgs}}{\partial x_j} + \frac{\partial \left(\langle \sigma_{ij}\rangle_\ell - \langle \sigma_{ij}\rangle_L\right)}{\partial x_j}, \tag{2.15}$$

$$\begin{aligned}\frac{\partial \langle \rho E\rangle_\ell}{\partial t} + \frac{\partial \left(\langle \rho E\rangle_\ell + \langle p\rangle_\ell\right)\langle u_j\rangle_L}{\partial x_i} &= -\frac{\partial \langle q_j\rangle_L}{\partial x_j} + \frac{\partial \langle \sigma_{ij}\rangle_L \langle u_i\rangle_L}{\partial x_j} - \frac{\partial q_j^{sgs}}{\partial x_j} + \frac{\partial \left(\langle q_j\rangle_\ell - \langle q_j\rangle_L\right)}{\partial x_j} \\ &+ \frac{\partial \left(\langle \sigma_{jk}\rangle_\ell - \langle \sigma_{jk}\rangle_L\right)\langle u_j\rangle_L}{\partial x_k} + \frac{1}{2}\frac{\partial \langle \rho\rangle_\ell \left(\langle u_k u_k u_j\rangle_L - \langle u_k\rangle_L \langle u_k\rangle_L \langle u_j\rangle_L - \tau_{kk}^{sgs}\langle u_j\rangle_L\right)}{\partial x_j},\end{aligned} \tag{2.16}$$

$$\frac{\partial \langle \rho\rangle_\ell \langle \phi_\alpha\rangle_L}{\partial t} + \frac{\partial \langle \rho\rangle_\ell \langle u_j\rangle_L \langle \phi_\alpha\rangle_L}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\left\langle \mu \frac{\partial \phi_\alpha}{\partial x_j}\right\rangle_\ell\right) - \frac{\partial M_j^\alpha}{\partial x_j} + \langle \rho S_\alpha\rangle_\ell, \tag{2.17}$$

where $M_j^\alpha = \langle \rho\rangle_\ell \left(\langle u_j \phi_\alpha\rangle_L - \langle u_j\rangle_L \langle \phi_\alpha\rangle_L\right)$. The filtering process leads to several terms that require closure. τ_{ij}^{sgs} is the sub-grid scale stress tensor and q_j^{sgs} is the sub-grid turbulent heat flux. These sub-grid terms represent the effect of the unresolved scales. The other unclosed terms are: the filtered diffusive heat flux, $\frac{\partial\left(\langle q_j\rangle_\ell - \langle q_j\rangle_L\right)}{\partial x_j}$; SGS viscous dissipation, $\frac{\partial\left(\langle \sigma_{jk}\rangle_\ell - \langle \sigma_{jk}\rangle_L\right)\langle u_j\rangle_L}{\partial x_k}$; viscous stress, $\left(\langle \sigma_{ij}\rangle_\ell - \langle \sigma_{ij}\rangle_L\right)$; divergence of turbulent diffusion, $\frac{1}{2}\frac{\partial \langle \rho\rangle_\ell \left(\langle u_k u_k u_j\rangle_L - \langle u_k\rangle_L \langle u_k\rangle_L \langle u_j\rangle_L - \tau_{kk}^{sgs}\langle u_j\rangle_L\right)}{\partial x_j}$. The last four terms in the right-hand side of

Eq. (2.16) can be neglected [56]. For other sub-grid scale terms, we employ the standard Smagorinsky model [57, 58]:

$$\begin{aligned} \tau_{ij}^{sgs} - \frac{2}{3} \langle \rho \rangle_\ell C_{\nu_2} \Delta^2 S^2 \delta_{ij} &= -2\mu_t \left(\langle S_{ij} \rangle_L - \frac{1}{3} \langle S_{kk} \rangle_L \delta_{ij} \right), \\ M_j^\alpha &= -\mu_t \frac{\partial \langle \phi_\alpha \rangle_L}{\partial x_j}, \\ q_j^{sgs} &= c_v \mu_t \frac{\partial \langle T \rangle_L}{\partial x_j}. \end{aligned} \tag{2.18}$$

The filtered strain rate tensor is $\langle S_{ij} \rangle_L = \frac{1}{2} \left[\frac{\partial \langle u_i \rangle_L}{\partial x_j} + \frac{\partial \langle u_j \rangle_L}{\partial x_i} \right]$. With that, the SGS viscosity is modeled by $\mu_t = \langle \rho \rangle_\ell \left[C_{\nu_1} \Delta \right]^2 S$, where $C_{\nu_1} = 0.2$, $Pr_t = 1$, $C_{\nu_2} = 0.18$, $\gamma_t = \mu_t / Sc_t$, $Sc_t = 1$, $S = \sqrt{2 \langle S_{ij} \rangle_L \langle S_{ij} \rangle_L}$. The parameter Δ denotes the filter size.

2.3 FILTERED MASS DENSITY FUNCTION

The filtered mass density function (FMDF) is the mass-weight spatially filtered value of the fine-grained density. It is defined as [19]:

$$F_L(\boldsymbol{\psi}; \mathbf{x}, t) = \int_{-\infty}^{+\infty} \rho(\mathbf{x}', t) \zeta(\boldsymbol{\psi}, \boldsymbol{\phi}(\mathbf{x}', t)) G(\mathbf{x}' - \mathbf{x}) d\mathbf{x}'. \tag{2.19}$$

The term $\zeta(\boldsymbol{\psi}, \boldsymbol{\phi}(\mathbf{x}', t))$ is the fine-grained density [16, 59]:

$$\zeta(\boldsymbol{\psi}, \boldsymbol{\phi}(\mathbf{x}, t)) = \prod_{\alpha=1}^{\sigma} \delta(\psi_\alpha - \phi_\alpha(\mathbf{x}, t)). \tag{2.20}$$

δ denotes the Dirac delta function, and $\boldsymbol{\psi}$ represents the scalar array in the sample space. With the condition of a positive filter kernel [60], F_L has all the properties of a mass density function [16]. The conditional filtered value of $Q(\mathbf{x}, t)$ is defined as:

$$\left\langle Q \middle| \boldsymbol{\psi} \right\rangle_\ell \equiv \frac{\int_{-\infty}^{+\infty} Q(\mathbf{x}', t) \rho(\mathbf{x}', t) \zeta(\boldsymbol{\psi}, \boldsymbol{\phi}(\mathbf{x}', t)) G(\mathbf{x}' - \mathbf{x}) d\mathbf{x}'}{F_L(\boldsymbol{\psi}, \mathbf{x}, t)}. \tag{2.21}$$

The FMDF is governed by the exact transport equation [17]:

$$\begin{aligned}\frac{\partial F_L}{\partial t}+\frac{\partial[\langle u_j(\mathbf{x},t)|\boldsymbol{\psi}\rangle_\ell F_L]}{\partial x_j} &= -\frac{\partial}{\partial\psi_\alpha}[S_\alpha(\boldsymbol{\psi})F_L]\\ &+ \frac{\partial}{\partial\psi_\alpha}\left[\left\langle\frac{1}{\rho(\boldsymbol{\phi})}\frac{\partial J_j^\alpha}{\partial x_j}\middle|\boldsymbol{\psi}\right\rangle_\ell F_L\right].\end{aligned} \tag{2.22}$$

This equation indicates that the effect of chemical reaction (the first term on right-hand side of Eq. (2.22)) appears in a closed form. The unclosed nature of SGS convection and mixing is shown via the conditional filtered values in the second term on the left-hand side of Eq. (2.22) and the second term on the right-hand side of Eq. (2.22). For closure of these terms, the gradient diffusion model and the linear mean square estimation (LMSE) model are adopted for convection and the molecular mixing terms, respectively [6, 19]. These are given in terms of the stochastic differential equations (SDE) [59, 61]:

$$dX_i^+(t)=\left[\langle u_i\rangle_L+\frac{1}{\langle\rho\rangle_\ell}\frac{\partial(\mu+\mu_t)}{\partial x_i}\right]dt+\sqrt{2(\mu+\mu_t)/\langle\rho\rangle_\ell}\,dW_i(t), \tag{2.23}$$

$$d\phi_\alpha^+=-\Omega_m\left(\phi_\alpha^+-\langle\phi_\alpha\rangle_L\right)dt+S_\alpha(\boldsymbol{\phi}^+)\,dt. \tag{2.24}$$

where dW_i is the Wiener-Levy process [62] and, X_i^+ and ϕ_α^+ are probabilistic representations of the position and the scalar variables. In the model, $\Omega_m = C_\phi\left(\mu+\mu_t\right)/\left(\langle\rho\rangle_\ell\,\Delta^2\right)$ is the SGS mixing frequency and $C_\phi = 4$ is a model constant. The Fokker-Planck equation corresponding to this model is [63]:

$$\frac{\partial F_L}{\partial t}+\frac{\partial[\langle u_j\rangle_L F_L]}{\partial x_j}=\frac{\partial}{\partial x_j}\left[(\mu+\mu_t)\frac{\partial(F_L/\langle\rho\rangle_\ell)}{\partial x_j}\right]+\frac{\partial}{\partial\psi_\alpha}\left[\Omega_m(\psi_\alpha-\langle\phi_\alpha\rangle_L)F_L\right]-\frac{\partial[S_\alpha F_L]}{\partial\psi_\alpha}. \tag{2.25}$$

Equation (2.25) represents the modeled FMDF transport equation. This equation may be integrated to obtain transport equations for the SGS moments. Since the FMDF is involved only for the scalar variable, all of the hydrodynamic SGS terms need to be modeled by other means. To establish the consistency and convergence of the MC solver, the generalized first

SGS moment $\langle\phi_\alpha\rangle_L$, and the SGS variance $\tau_\alpha \equiv \tau(\phi_\alpha,\phi_\alpha)$ are considered. These moments are obtained via integration of the Eq. (2.25), as given by:

$$\frac{\partial(\langle\rho\rangle_\ell \langle\phi_\alpha\rangle_L)}{\partial t} + \frac{\partial[\langle\rho\rangle_\ell \langle u_j\rangle_L \langle\phi_\alpha\rangle_L]}{\partial x_j} = \frac{\partial}{\partial x_j}\left[(\mu+\mu_t)\frac{\partial(\langle\phi_\alpha\rangle_L)}{\partial x_j}\right] + \langle\rho\rangle_\ell \langle S_\alpha\rangle_L, \tag{2.26}$$

$$\begin{aligned}\frac{\partial(\langle\rho\rangle_\ell \tau_\alpha)}{\partial t} + \frac{\partial[\langle\rho\rangle_\ell \langle u_j\rangle_L \tau_\alpha]}{\partial x_j} = \frac{\partial}{\partial x_j}\left[(\gamma+\gamma_t)\frac{\partial\tau_\alpha}{\partial x_j}\right] + 2(\gamma+\gamma_t)\left[\frac{\partial(\langle\phi_\alpha\rangle_L)}{\partial x_j}\frac{\partial(\langle\phi_\alpha\rangle_L)}{\partial x_j}\right] \\ -2\Omega_m \langle\rho\rangle_\ell \tau_\alpha + 2\langle\rho\rangle_\ell (\langle\phi_\alpha S_\alpha\rangle_L - \langle\phi_\alpha\rangle_L \langle S_\alpha\rangle_L).\end{aligned} \tag{2.27}$$

2.4 SPECTRAL ELEMENT

For the spatial discretization, the Chebyshev multidomain spectral method (CMSM) is implemented [64, 65]. In CMSM, the computational domain, Ω, is devided into K non-overlapping hexahedral sub-domains, D_k,

$$\Omega = \sum_{k=1}^{K} D_k. \tag{2.28}$$

Each sub-domain is mapped onto a unit hexahedron ($[0,1]\times[0,1]\times[0,1]$) in 3-D using iso-parametric mapping to ensure that the spectral accuracy is not affected by the domain boundary approximation. Applying the mapping, Eq. (2.10) without source terms becomes [66]:

$$\frac{\partial\widetilde{\mathbf{Q}}}{\partial t} + \frac{\partial\widetilde{\mathbf{F}_i^a}}{\partial X_i} - \frac{\partial\widetilde{\mathbf{F}_i^v}}{\partial X_i} = 0. \tag{2.29}$$

where the tilde indicates a mapped vector, X_i denotes mapped space, x_i denotes physical space. The transformation from physical space to mapped space is via the Jacobian:

$$\widetilde{\mathbf{Q}} = J\mathbf{Q}, \tag{2.30}$$

$$\begin{aligned}J(X,Y,Z) = \frac{\partial x}{\partial X}\left(\frac{\partial y}{\partial Y}\frac{\partial z}{\partial Z} - \frac{\partial y}{\partial Z}\frac{\partial z}{\partial Y}\right) + \frac{\partial x}{\partial Y}\left(\frac{\partial y}{\partial X}\frac{\partial z}{\partial Z} - \frac{\partial y}{\partial Z}\frac{\partial z}{\partial X}\right) + \\ \frac{\partial x}{\partial Z}\left(\frac{\partial y}{\partial X}\frac{\partial z}{\partial Y} - \frac{\partial y}{\partial Y}\frac{\partial z}{\partial X}\right).\end{aligned} \tag{2.31}$$

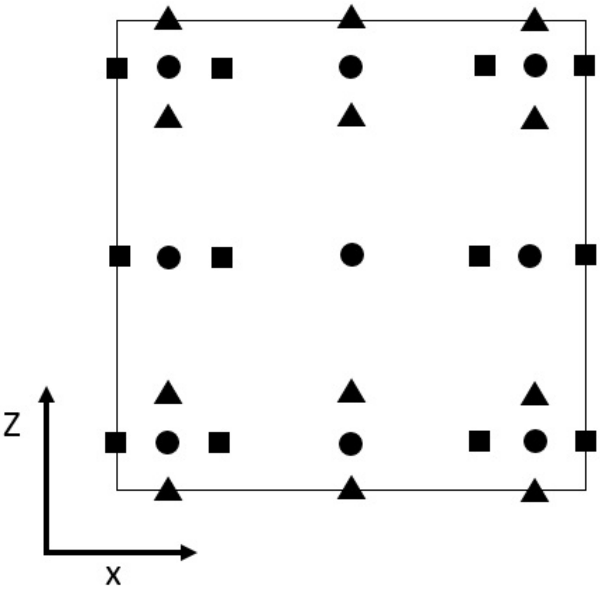

Figure 1: Diagram of a Gauss and Gauss-Lobatto grid system in two dimensions with polynomial degree 3. The filled circle denotes a Gauss-Gauss-Gauss grid point (ggg), the filled square denotes a Lobatto-Gauss-Gauss grid point (lgg), and the filled triangle denotes a Gauss-Gauss-Lobatto grid point (ggl).

Since the domain is divided into sub-domains, the equation needs to be solved for each sub-domain. Let $\widetilde{\mathbf{Q}}^k$, $\widetilde{\mathbf{F}_i^a}^k$, and $\widetilde{\mathbf{F}_i^v}^k$ denote the approximated solution, the advective fluxes, and the viscous fluxes respectively. Substitution of these approximation into Eq. (2.29) gives:

$$\frac{\partial \widetilde{\mathbf{Q}}^k}{\partial t} + \frac{\partial \widetilde{\mathbf{F}_i^a}^k}{\partial X_i} - \frac{\partial \widetilde{\mathbf{F}_i^v}^k}{\partial X_i} = R, \tag{2.32}$$

where R is the residual. In collocation methods, R must be zero at the Gauss points:

$$\left[\frac{\partial \widetilde{\mathbf{Q}}}{\partial t}\right]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} + \left[\frac{\partial \widetilde{\mathbf{F}_i^a}}{\partial X_i}\right]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - \left[\frac{\partial \widetilde{\mathbf{F}_i^v}}{\partial X_i}\right]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = 0, \tag{2.33}$$

The quadrature points for Gauss and Gauss-Lobatto grids which provide zero for the residual are defined by Chevbyshev quadrature points:

$$X_{j+\frac{1}{2}} = \frac{1}{2}\left\{1 - \cos\left[\frac{(2j+1)\,\pi}{2N}\right]\right\}, j = 0, ..., N-1, \tag{2.34}$$

and

$$X_j = \frac{1}{2}\left\{1 - \cos\left[\frac{\pi j}{N}\right]\right\}, j = 0, ..., N. \tag{2.35}$$

on the unit interval $[0, 1]$. $X_{j+\frac{1}{2}}$ does not mean that the point is in between X_j and X_{j+1}. Based on two grids system, two types of Lagrange interpolating polynomials are defined,

$$h_{i+\frac{1}{2}}(\xi) = \prod_{j=0, i\neq j}^{N-1} \frac{\xi - X_{j+\frac{1}{2}}}{X_{i+\frac{1}{2}} - X_{j+\frac{1}{2}}}, \tag{2.36}$$

$$l_i(\xi) = \prod_{j=0, i\neq j}^{N} \frac{\xi - X_j}{X_i - X_j}. \tag{2.37}$$

The solutions are approximate on Guass-Gauss-Gauss grid (ggg) at $(X_{j+\frac{1}{2}}, Y_{j+\frac{1}{2}}, Z_{j+\frac{1}{2}})$. The approximated solutions in mapped space are approximated using Largrange interpolants as:

$$\widetilde{Q}^{ggg}(X, Y, Z) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} Q^{g}_{i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2}} h_{i+\frac{1}{2}}(X)\, h_{j+\frac{1}{2}}(Y)\, h_{k+\frac{1}{2}}(Z). \tag{2.38}$$

The solution procedures require the interpolation from Gauss grid to Gauss-Lobatto grid. The interpolation uses the same interpolating polynomial as for approximated solutions. The following equation is for the lgg point. In case of another Gauss-Lobatto point, the arguments in interpolating polynomial must be changed accordingly.

$$Q^{lgg}\left(X_i, Y_{j+\frac{1}{2}}, Z_{k+\frac{1}{2}}\right) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1}\sum_{p=0}^{N-1} Q^{ggg}_{m+\frac{1}{2}, n+\frac{1}{2}, p+\frac{1}{2}} h_{m+\frac{1}{2}}(X_i)\, h_{n+\frac{1}{2}}\left(Y_{j+\frac{1}{2}}\right) h_{p+\frac{1}{2}}\left(Z_{k+\frac{1}{2}}\right). \tag{2.39}$$

Once the values are interpolated to the Gauss-Lobatto grid, the advective fluxes are computed. Some of the flux is stored in the interface between two sub-domains, and the problem occurs when the fluxes calculated from each sub-domain are different. To overcome this problem, patching fluxes algorithm is considered. The viscous fluxes require the derivative of the solutions. To ensure the continuity of the derivative at the interface, the patching process of the advective flux is implemented. After patching the fluxes, their derivatives are computed at the Gauss grid. These gradients are then interpolated back to Gauss-Lobatto points. The viscous fluxes are computed using Eq. (2.11), and then be patched. Finally, the total fluxes are obtained by adding inviscid and viscous fluxes. Once the total fluxes are computed at the Gauss-Lobatto grid, the flux interpolants are constructed as:

$$\widetilde{\mathbf{F}}_1\left(X,Y,Z\right)=\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}\sum_{p=0}^{N-1}\widetilde{\mathbf{F}}_{1m,n+\frac{1}{2},p+\frac{1}{2}}l_m\left(X\right)h_{n+\frac{1}{2}}\left(Y\right)h_{p+\frac{1}{2}}\left(Z\right), \tag{2.40}$$

$$\widetilde{\mathbf{F}}_2\left(X,Y,Z\right)=\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}\sum_{p=0}^{N-1}\widetilde{\mathbf{F}}_{2m+\frac{1}{2},n,p+\frac{1}{2}}h_{m+\frac{1}{2}}\left(X\right)l_n\left(Y\right)h_{p+\frac{1}{2}}\left(Z\right), \tag{2.41}$$

$$\widetilde{\mathbf{F}}_3\left(X,Y,Z\right)=\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}\sum_{p=0}^{N-1}\widetilde{\mathbf{F}}_{3m+\frac{1}{2},n+\frac{1}{2},p}h_{m+\frac{1}{2}}\left(X\right)h_{n+\frac{1}{2}}\left(Y\right)l_p\left(Z\right), \tag{2.42}$$

These fluxes are differentiated and evaluated at the Gauss grid,

$$\frac{\partial\widetilde{\mathbf{F}}_1\left(X_{i+\frac{1}{2}},Y_{j+\frac{1}{2}},Z_{k+\frac{1}{2}}\right)}{\partial X}=\sum_{m=0}^{N}\widetilde{\mathbf{F}}_1\left(X_m,Y_{j+\frac{1}{2}},Z_{k+\frac{1}{2}}\right)\frac{\partial l_m\left(X_{i+\frac{1}{2}}\right)}{\partial X}, \tag{2.43}$$

$$\frac{\partial\widetilde{\mathbf{F}}_2\left(X_{i+\frac{1}{2}},Y_{j+\frac{1}{2}},Z_{k+\frac{1}{2}}\right)}{\partial Y}=\sum_{n=0}^{N}\widetilde{\mathbf{F}}_2\left(X_{i+\frac{1}{2}},Y_n,Z_{k+\frac{1}{2}}\right)\frac{\partial l_n\left(Y_{j+\frac{1}{2}}\right)}{\partial Y}, \tag{2.44}$$

$$\frac{\partial\widetilde{\mathbf{F}}_3\left(X_{i+\frac{1}{2}},Y_{j+\frac{1}{2}},Z_{k+\frac{1}{2}}\right)}{\partial Z}=\sum_{p=0}^{N}\widetilde{\mathbf{F}}_3\left(X_{i+\frac{1}{2}},Y_{j+\frac{1}{2}},Z_p\right)\frac{\partial l_p\left(Z_{k+\frac{1}{2}}\right)}{\partial Z}. \tag{2.45}$$

2.5 PATCHING DOMAINS: RIEMANN SOLVER

To patch the values at the interface, the Mortar method is implemented [67]. In this method, the fluxes at Lobatto grid from the two adjacent sub-domains are projected onto the interface as shown in Fig. 2. This problem is treated as a Riemann problem. To describe this method in one-dimension, we consider the following initial value problem,

$$\begin{gathered}\frac{\partial\mathbf{Q}}{\partial t}+\frac{\partial\mathbf{F}\left(\mathbf{Q}\right)}{\partial x}=0,\\ \mathbf{Q}\left(x,t=0\right)=\begin{cases}\mathbf{Q}_L, & x<0\\ \mathbf{Q}_R, & x>0\end{cases},\end{gathered} \tag{2.46}$$

where,

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho E \\ \rho \phi \end{pmatrix}, \tag{2.47}$$

$$\mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho \left(\rho E + p \right) u \\ \rho \phi u \end{pmatrix}. \tag{2.48}$$

The solution between these states can be determined from left ($\mathbf{Q}_L$) and right ($\mathbf{Q}_R$) states. The solution is approximated using Roe's scheme [68]. The key idea for this scheme is to transform $\frac{\partial \mathbf{F}(\mathbf{Q})}{\partial x}$ to $A\left(\mathbf{Q}_L, \mathbf{Q}_R\right)\frac{\partial \mathbf{Q}}{\partial x}$ as shown by Eq. (2.49):

$$\frac{\partial \mathbf{Q}}{\partial t} + A\left(\mathbf{Q}_L, \mathbf{Q}_R\right)\frac{\partial \mathbf{Q}}{\partial x} = 0. \tag{2.49}$$

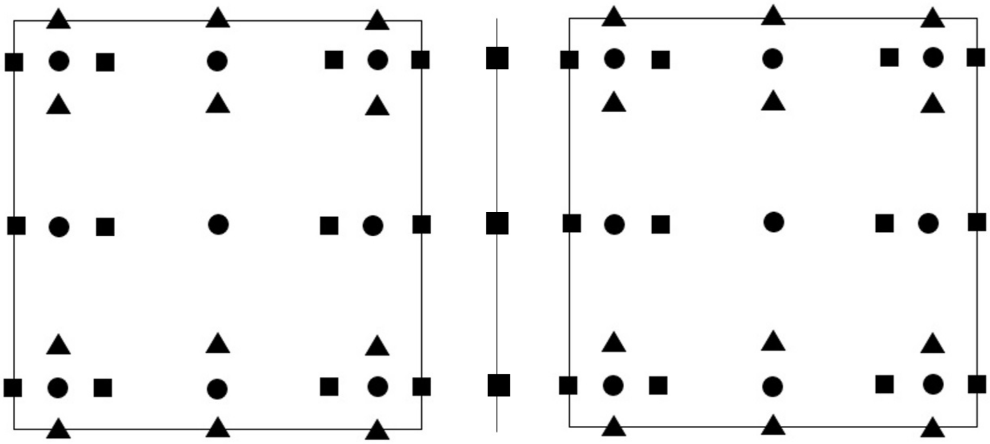

Figure 2: Mortar method is used to patch flux values between two sub-domains

Here, A is the Jacobian matrix.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ (\gamma-1)\,H - u^2 - a^2 & (3-\gamma)\,u & \gamma-1 & 0 \\ \frac{1}{2}u\left[(\gamma-3)\,H - a^2\right] & H - (\gamma-1)\,u^2 & \gamma u & 0 \\ -\phi u & \phi & 0 & u \end{bmatrix}, \tag{2.50}$$

where $a = \sqrt{\frac{\gamma p}{\rho}}$ and $H = e + \frac{p}{\rho}$. After replacing the non-linear term, $\frac{\partial \mathbf{F}(\mathbf{Q})}{\partial x}$, Eq. (2.46) is treated as a simplest PDE of hyperbolic type or can be called linear advection equation. The solution of this equation can be obtained from Eigen values and vectors of matrix A:

$$\lambda_1 = u - a, \lambda_2 = \lambda_3 = u, \lambda_4 = u + a, \tag{2.51}$$

and

$$\mathbf{K}^1 = \begin{bmatrix} 1 \\ u-a \\ H-ua \\ \phi \end{bmatrix}; \mathbf{K}^2 = \begin{bmatrix} 1 \\ u \\ \frac{1}{2}u^2 \\ \phi \end{bmatrix} \mathbf{K}^3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{K}^4 = \begin{bmatrix} 1 \\ u+a \\ H+ua \\ \phi \end{bmatrix}; \tag{2.52}$$

where

$$\begin{gathered} \alpha_1 = \frac{1}{2\widetilde{a}^2}\left[\Delta p - \widetilde{\rho}\widetilde{a}\Delta u\right], \alpha_4 = \frac{1}{2\widetilde{a}^2}\left[\Delta p + \widetilde{\rho}\widetilde{a}\Delta u\right], \widetilde{a} = \sqrt{(\gamma-1)\left(\widetilde{H} - \frac{1}{2}\widetilde{u}^2\right)}, \\ \widetilde{\rho} = \sqrt{\rho_L \rho_R}, \widetilde{u} = \frac{\sqrt{\rho_L}u_L + \sqrt{\rho_R}u_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \widetilde{H} = \frac{\sqrt{\rho_L}H_L + \sqrt{\rho_R}H_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}. \end{gathered} \tag{2.53}$$

According to Harten and Hyman [69], the flux between two sub-domains can be calculated by quantities from both side of sub-domain:

$$\mathbf{F}_{i+\frac{1}{2}} = \frac{1}{2}\left(\mathbf{F}_L + \mathbf{F}_R\right) + \frac{1}{2}\left(\lambda_1 \alpha_1 \mathbf{K}^1 + \lambda_4 \alpha_4 \mathbf{K}^4\right). \tag{2.54}$$

Here, the interface values are corrected by taking the average of the two interface values. It involves just the domain face values.

2.6 TIME INTEGRATION

The Runge-Kutta (RK) scheme of Williamson [70] is employed for discretization in time. This schemes is explicit with large stability limits and ease of programming. There is no information of previous time-steps used for updating in each iteration. After spatial discretization, Eq. (2.55) is obtained,

$$\frac{\partial \mathbf{Q}}{\partial t} = \mathbf{D}\left(t, \mathbf{Q}\left(t\right)\right), \mathbf{Q}\left(t_0\right) = \mathbf{Q}_0, \tag{2.55}$$

In each time step, the solutions are updated s times:

$$\mathbf{Q}^n = \mathbf{Q}^{n-1} + \Delta t \sum_{t=0}^{s} b_i K_i, \tag{2.56}$$

$$K_i = \mathbf{D}\left(t^{n-1} + c_i \Delta t, \mathbf{q}^{n-1} + \Delta t \sum_{j=1}^{i-1} a_{ij} K_j\right), \tag{2.57}$$

Williamson's scheme avoids calculation of K_j matrix for every iteration by storing the information:

$$w_i = \alpha_i w_{i-1} + \Delta t \mathbf{D}\left(_{i-1}, q_{i-1}\right) q_i = q_{i-1} + \beta w_i, \tag{2.58}$$

and

$$q_i = q_{i-1} + \beta w_i. \tag{2.59}$$

where the coefficients are given in Table 1.

Table 1: Coefficients for 4^{th} order Runge-Kutta with five stages [1].

Stage	α	β	c
1	0	0.14965	0
2	-0.41789	0.37921	0.14965
3	-1.19215	0.82295	0.3704
4	-1.69778	0.69945	0.62225
5	-1.51418	0.15305	0.95828

2.7 MONTE CARLO SIMULATION

Monte Carlo (MC) methods have been very effective for simulation of the PDF and FDF transport [71, 72]. Here, Eq. (2.25) is solved by using a MC method. The FDF is represented by an ensemble of MC particles, each with set of scalars $\phi_\alpha^n(t) = \phi_\alpha^n(X^{(n)}(t), t)$ and the Lagrangian position vector $X^{(n)}$. The information on each MC particles is updated according to convection, diffusion, mixing and chemical reaction. For demonstration, Fig. 3 shows the position and concentration values for each MC particle, and are updated in every time-step.

Numerical solution of the Eq. (2.24) requires the values of the filtered velocity, the diffusion coefficient, and gradients of the scalar field at the particle location for every iteration. These are obtained from the SEM solution. The required information is evaluated by considering every particle within the ensemble element centered at the point of interest. This ensemble domain is characterized by the size of Δ_E which is the diameter of the circle at the point of interest (Fig. 4). For accurate statistics, it is important to maximize the number of particles within the ensemble domain, as well as minimize Δ_E. The basis function method [73] is used to calculate the ensemble averaged statistics. A basis function for a vertex i ($i = 1, 2, ..., N_v^E$) of the ensemble domain E with respect to the particle j is given by:

$$b_{ij}^E = \frac{1}{1 + r_{ij} \sum_{k=1, k\neq i}^{N_v^E} \frac{1}{r_{kj}}}, \tag{2.60}$$

where r_{ij} is the distance between the vertex i and the particle j. An estimation of the mean scalar at the vertex i can be obtained by summing over all the particles in the cells which share the vertex and taking into account the vertex basis function:

$$\langle Q_i \rangle_L = \frac{\sum_{j=1}^{N_p^E} b_{ij}^E Q_j}{\sum_{j=1}^{N_p^E} b_{ij}^E}. \tag{2.61}$$

Here, N_p^E denotes the number of particles in cell E. $\langle Q_i \rangle_L$ is the ensemble average of Q.

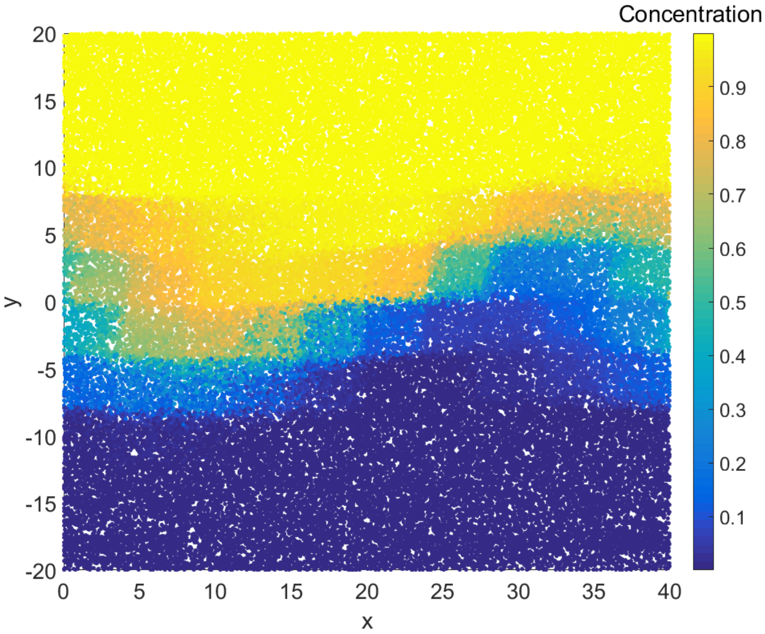

Figure 3: Monte Carlo particles distribution in 2-D.

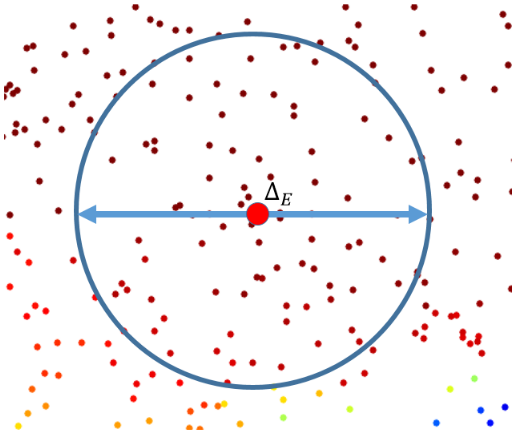

Figure 4: Ensemble averaging.

3.0 PARALLEL COMPUTATIONS

3.1 SPECTRAL ELEMENT SIMULATION

The spectral-element method has a feature that makes its parallel implementation somewhat easy. It does not employ global assembly to establish continuity across elements like FEM; only the information between element is exchanged. There is no need to construct the huge matrix with all constrained variables. The information is passed along the element faces. Spatial and time discretization can be done within one element. The computational domain can be divided into smaller domains, and then can be allocated to a different processor [74]. To decompose the mesh, the geometric partitioning software METIS is used which employs recursive spectral bisection algorithm and partitions the grid per processor [75]. In METIS, the sub-domains are decomposed in such a way that the number of neighboring domains is minimized. This way the communication between each neighbor processors is optimal. For a simple geometry, this process is demonstrated in Fig. 5. Partitioning the domain on two processors leads to the allocation of five and four domains on each processor. As shown in Fig. 6, the domains are globally numbered from 1 to 9, whereas the local numbering per processor is from 1 to 5, and 1 to 4. To keep track of how the local numbering is connected to the global numbering, a domain matrix with the row equals to the maximum global domain number (9 in this case), and column equals to the maximum local domain number (5 in this case). In each column, the local grid number and the processor number are stored.

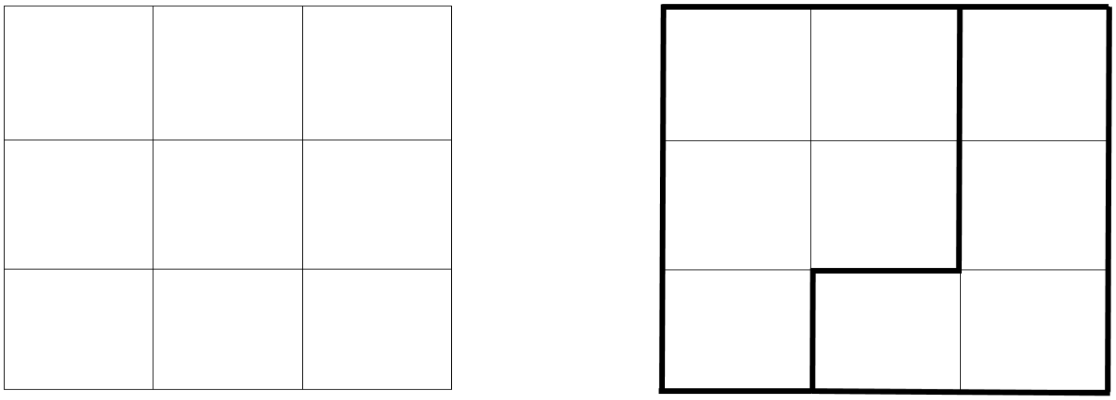

Figure 5: Domain partitions

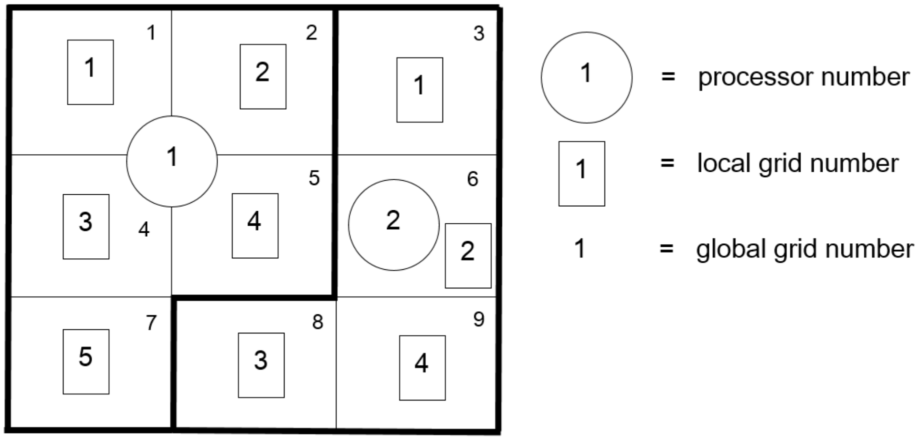

Figure 6: Grid partitioning on two processors.

Dealing with mortar in parallel implementation is a difficult task. As shown in Fig. 7, the mortars which will be used to calculate the average values can be in different processors. Keeping track of the mortars requires storage of the local mortar number in mortar matrix. Also, calculating average flux between each element requires information from both sides of the element. The domain of interest is called the master domain, and the adjacent domains are called slaves. The master and slave domain numbers are stored in the mortar matrix. To send the information from the domain to the mortar, there is one loop that counts on the global mortar numbering. For each mortar, it is determined whether the mortar is an interprocessor mortar or not. In the parallel implementation, different information can be reached at the same time. Interprocessor mortar is defined such that the information of interest is reached with some specific processor. For the mortars, the information needs to be exchanged between the processors. More specifically, an array of variables has to be sent from the domain to the mortar. The domain number and processor number from where the information needs to be sent can be found by looking up the slave domain in the mortar matrix, and consequently looking up the processor number in the domain matrix. The specific domain face from where information needs sending is looked up in the mortar matrix. The counter on that loop is used as the tag in the send operation. The receiving processor number is found by looking up the master domain in the mortar matrix and the processor number in the domain matrix. The receiving mortar is found by looking up the local mortar number in the mortar matrix. The receiving tag is again available in the form of the counter on the loop. The process of receiving information in the domain from the mortar uses the same procedure.

The main advantage of looping through the global mortar numbering is that the tag number is readily available, and every send operation is matched by its receive operation which will prevent the code from stalling. A disadvantage is that the code would do excessive execution, since it loops through the global mortar number, while the number of mortars per processors is significantly less. The execution, however, is limited to an IF statement, which checks whether the mortar is an edge mortar, and what processor it belongs. To decrease the computational cost and increase the scalability, non-blocking MPI task is implemented [76]. This implementation copies the interprocessor mortar to its slave domain. With this modification, the amount of message passing does not change. However, this passing can be implemented at a more convenient location. Another modification is the addition of storage of a parameter that determines whether a domain has a face that needs message passing. The purpose of this parameter is that if a domain does not need message passing, then computations within that domain can be performed without needing the information resulting from message passing. This process is demonstrated in Algorithm 1.

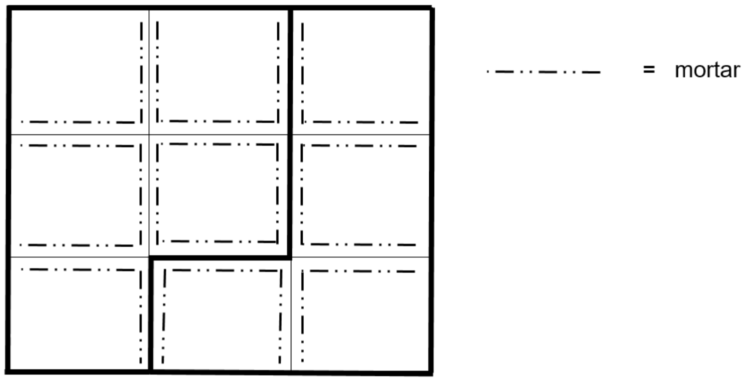

Figure 7: Mortars on grid partitioning

3.2 MONTE CARLO SIMULATION

The challenging task in the parallelization of the Monte Carlo simulation is associated with particle tracking. The MC particles can move from one element to another, causes the solver not to have a local character (Fig. 8). Another difficulty is that the MC simulation must couple with SEM. The communication between these two algorithms is through interpolation and ensemble averaging [53]. The MC simulation requires interpolation of SEM values to the particle position, and SEM requires ensemble average information from the MC solver. This strong coupling makes the parallel implementation very challenging. One way to overcome the problem is to preserve the local character of the solver. To track a particle within a grid partition, the only information required is the local domain number in which the particles reside. With this domain number, the fluid properties of the particle are determined, and the particle position is updated. The information required for the exchange of particles between processors is stored in a particle-matrix local to the processor and a particle-matrix global to all processors. The entries in the local matrix are represented by all the particles that leave the respective processor in an iteration. The local matrix is mostly sized by a percentage of the estimated maximum number of particles per processor. The size of this matrix represents the maximum number of particles being exchanged between processors per iteration.

For exchange of particles between processors, firstly the entries in the local matrix are determined. The procedure is as follows: If the mapped space coordinate of a particle exceeds a domain boundary, it is determined which face it crosses. With the known domain face and number, the global mortar number, which is stored in each domain array, is determined. With the mortar matrix from SEM section, this mortar can be justified if this is an interprocessor mortar. If this is the case, the particle is crossing processor. If only one coordinate direction exceeds the domain boundaries, then the processor to which the particle is moving into is determined through the mortar matrix. The local particle matrix can now be updated with one entry. If more than one coordinate direction exceeds the domain boundary, the particle may be crossing a corner of the grid partitioning. In this case, a search algorithm over the processors is required to find the processor to which the particle is moving. If the grids are structured, the processor corresponding to a specific particle coordinate is easily found.

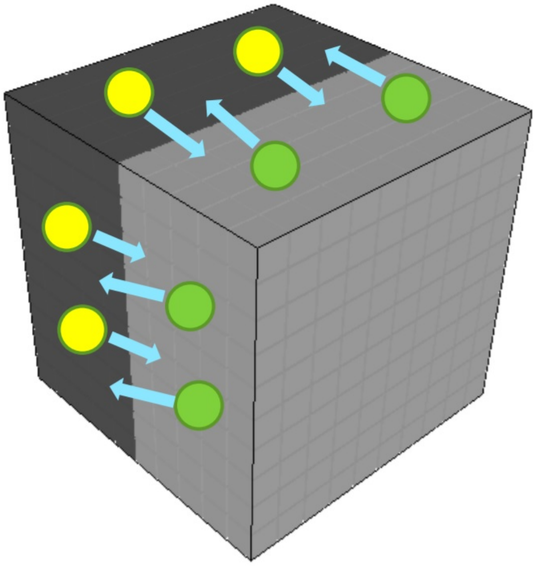

Figure 8: Particle exchange

All local particle matrices are reduced to the global particle matrix with the gather operation. This operation is performed on an equally sized local particle matrix for all processors, but it is noted that a variable sized gather operation will most likely be more effective. After the operation, the global particle matrix is sorted such that all particles that move between two processors are following entries in the global particle matrix. With the sorted global particle matrix, all particles that move between processors are exchanged with non-blocking receive operations. Non-blocking send and receive operations are employed since during the operations, all of the particles that are not crossing a partition boundary, can be updated. This effectively reduces the cost of the MPI operations. Once the particles are exchanged, their information is updated as well. The parallel algorithm for one-time step per processor is presented in Algorithm 2.

In previous studies, it is suggested that using 40-60 particles per element provides sufficient accuracy [21, 77]. In a domain with 60^3 elements, the number of particles would be about 8,640,000. It means that in every time step, the number of interpolation loop has to be at least 8,640,000. Spectral interpolation is optimum for this purpose.

3.3 PARALLEL SCALABILITY

There are two basic ways to measure the parallel performance: strong and weak scaling. Strong scaling demonstrates the capability in parallelization. It is used as justification for a program that takes a long time to run. In this test, the problem size stays fixed, but the number of processor increases and the number of processors varies from 1 to 64. The weak scaling efficiency is $\frac{100t_1}{Nt_N}$ where t_1 is the time to complete the work with one processor, and t_N is the time to complete the work with N processor. The weak scaling is shown in Fig. 9.

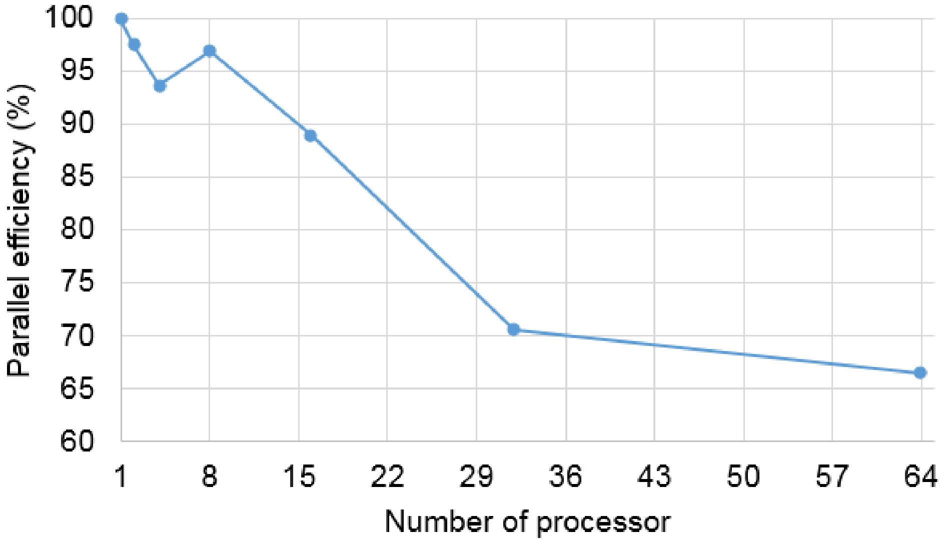

Figure 9: Weak scaling.

Algorithm 1: The parallel algorithm of SEM for one time step per processor

```
Interpolate Q from the Gauss to Lobatto grid and determine the fluxes;
for i=1,number of mortar do
    if interprocessor mortar then
        Send Q between processors to mortar ;
    else
        Project Q onto mortar;
    end
end
for i=1,number of mortar do
    Determine the advective flux and determine the average of Q on the mortar;
end
for i=1,number of mortar do
    if interprocessor mortar then
        Send the average of Q between processors to the domain ;
    else
        Project the average of Q onto the domain;
    end
end
Determine the gradient of Q and viscous flux, then interpolate it to Lobatto grid;
for i=1,number of mortar do
    if interprocessor mortar then
        Send the viscous flux between processors to the mortar;
        Determine the viscous flux, then send them between processors to the domain;
    else
        Project the viscous flux onto mortar;
        Determine the viscous flux, Project viscous flux onto the domain;
    end
end
Determine the gradient at the Gauss grid, and update;
```

Algorithm 2: The parallel algorithm of MC simulation for one-time step per processor

```
Gather the local particle matrix into the global particle matrix;
Sort the global particle matrix;
Exchange particles between processors with non-blocking sends and receives;
for i=1,number of particle do
    Update particles;
    if interprocessor paricle then
        Update the local mortar matrix;
    end
end
Update the exchanged particles;
for i=1,number of particle do
    if interprocessor paricle then
        Update the local mortar matrix;
    end
end
```

4.0 RESULTS

To demonstrate its effectiveness, the hybrid SEM-MC solver is employed for LES of a two-dimensional, temporally developing mixing layer, as considered in previous DNS [30, 60]. Unsteady turbulent mixing of two adjacent streams of fluid with different speeds are considered. In the flow configuration, x and y denote the stream-wise and the cross-stream directions, respectively. The velocity components in these directions are denoted by u and v, respectively. The flow is periodic in the stream-wise direction. Both the filtered stream-wise velocity and passive scalar fields are initialised with hyperbolic tangent profiles, where $\langle u \rangle_L = 1$, $\langle \phi \rangle_L = 1$ on the top stream, and $\langle u \rangle_L = -1$, $\langle \phi \rangle_L = 0$ on the bottom stream.

Simulations are conducted on a box, $0 \leq x \leq L$ and $-L/2 \leq y \leq L/2$. The stream-wise length L is specified such that $L = 2^{n_p} \lambda_u$, where n_p is the desired number of successive vortex pairings and λ_u is the wavelength of the most unstable mode corresponding to the mean stream-wise velocity profile imposed at the initial time. The flow variables are normalized with respect to the half initial vorticity thickness, $L_r = \left[\delta_v \left(t = 0\right)/2\right]; \delta_v = \Delta U / |\partial \overline{\langle u \rangle} / \partial y|_{max}$, where $\overline{\langle u \rangle}$ is the Reynolds averaged value of the filtered stream-wise velocity and, ΔU is the velocity difference across the layer. The reference velocity is $U_r = \Delta U/2$. The Reynolds number ($Re = \frac{\rho U_r L_r}{\mu}$) is equal to 50 and the Mach number ($Ma = \frac{U_r}{a_r}$) is 0.2. Here a_r is the reference speed of sound based on the reference temperature. The formation of the large scale vortical structures are expedited by harmonic forcing of the layer. To initiate turbulence, 2D perturbations are added with a random phase shift [78, 79]. The Reynolds averaged values of the filtered values are obtained by ensemble averaging in the homogeneous x-direction. There are denoted by an overbar.

In the reacting case, an irreversible, second-order reaction of type $A + \nu B \rightarrow (\nu + 1)P$ is considered. In this case, the reactants are initialized such that $A = \phi$, and $B = 1 - A$.

The reactant conversion is governed by $S_A = -k_r AB$, where k_r is the reaction rate constant; and A, B, P denote the mass fractions of the three reactants. The non-dimensional number associated with reaction rate is the Damköhler number ($Da = \frac{k_r L_r}{U_r}$). This number plays an important role as it describes the interaction between chemical reaction and hydrodynamics. As the Damköhler number is increased, the chemical reaction becomes faster compared to fluid dynamics times scale.

The vortical structure form at $t = 40$ as shown in Fig. 10. To assess the consistency of FDF solver, the values of the conserved scalar ($\langle \phi \rangle_L$) from SEM and FDF are compared in Fig. 11. It is shown that the two results are highly correlated, with the correlation coefficient equals to 0.999. For the second-order moments, $\tau(a,b) = \langle ab \rangle_L - \langle a \rangle_L \langle b \rangle_L$ denotes the SGS stresses, the resolved stresses are expressed by $R(a,b) = \overline{\langle a \rangle_L \langle b \rangle_L} - \left(\overline{\langle a \rangle_L}\right)\left(\overline{\langle b \rangle_L}\right)$, and the total stresses are $r(a,b) = \overline{(ab)} - \overline{a}\overline{b}$. For a generic filter, $r(a,b) = R(a,b) + \overline{\tau(a,b)}$. The overall consistency of these two methods is best achieved by comparing the second-order moments as obtained from FDF with those via SEM. This is shown in Fig. 12 and provides a quantitative demonstration of the consistency of the FDF simulations.

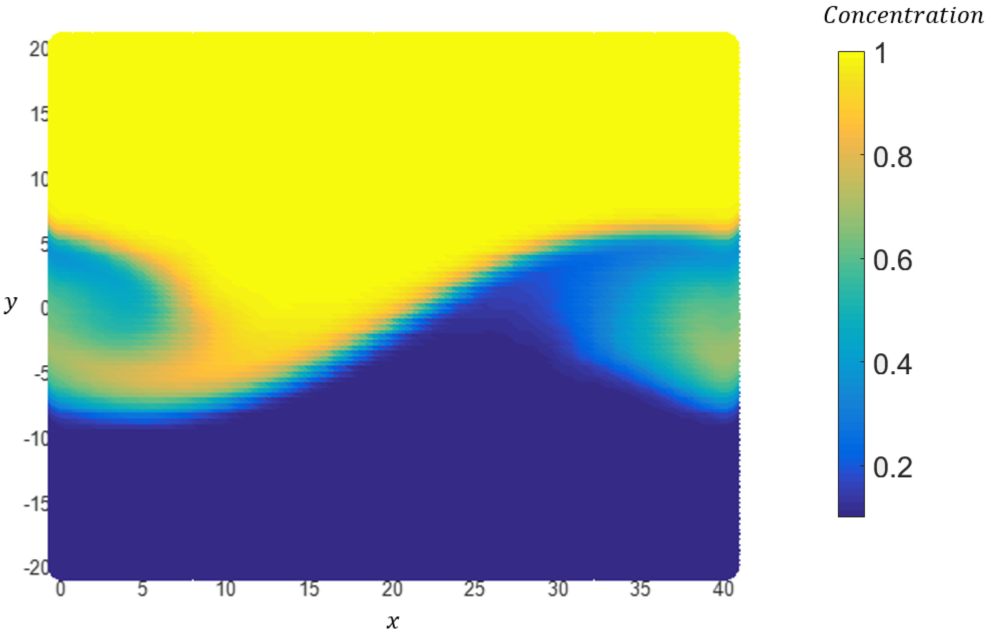

Figure 10: Contour plot of the filtered scalar field at t = 40.

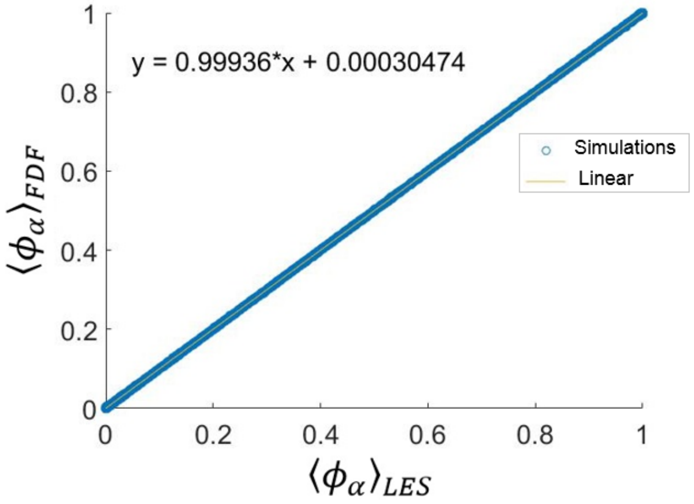

Figure 11: Scatter plot of the filtered scalar SEM vs FDF. The correlation coefficient is 0.999.

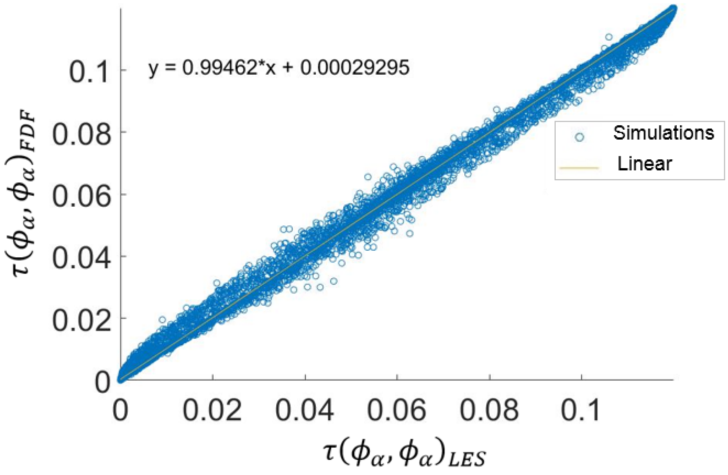

Figure 12: Scatter plot of the SGS variance at $t = 40$ on 64×64 mesh.

For the h-refinement study [80], the polynomial degree is fixed to be $p = 3$ and the grid resolution is varied: 32×32, 64×64 and 128×128. Figures 13– 15 show the averaged SGS variance, the resolved variance, and the total stress, respectively. The averaged resolved SGS variance does not change significantly. In Figs. 16-18, it is shown that as polynomial degree increases, the variance decreases significantly. According to Boyd [81], the error from changing grid resolution is of order $O(h^n)$. The error from changing polynomial degree is proportional to $O((\frac{1}{N})^N)$, where N is the number of collocation points defined by the polynomial order.

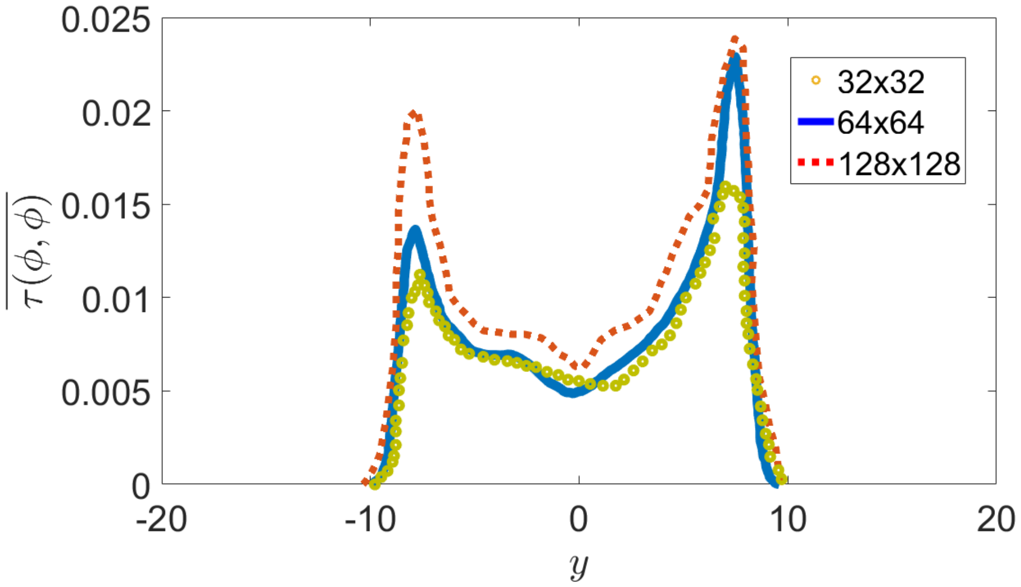

Figure 13: Averaged SGS variance at $t = 60$ with $p = 3$.

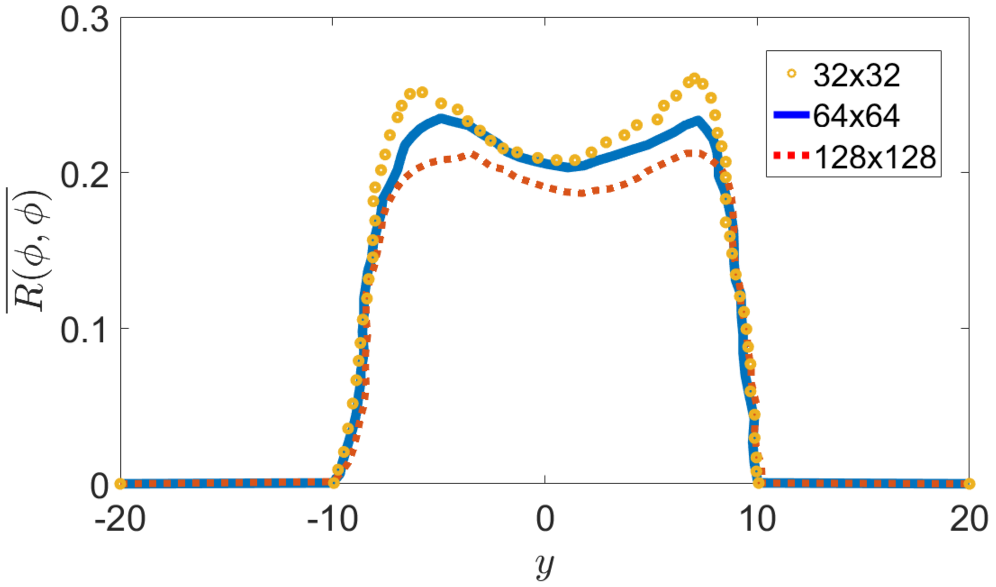

Figure 14: Averaged resolved variance at $t = 40$ with $p = 3$..

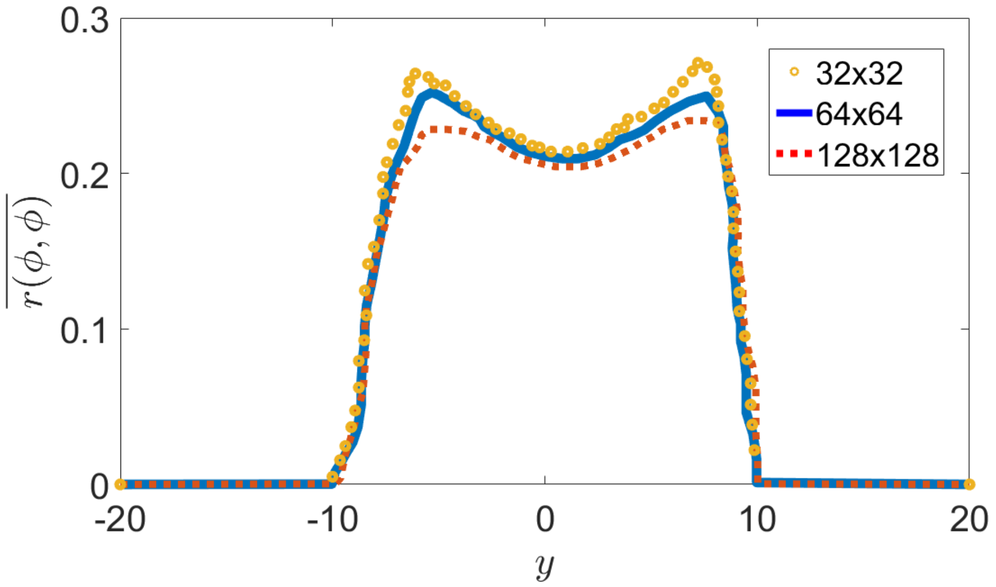

Figure 15: Averaged total stress at $t = 40$ with $p = 3$.

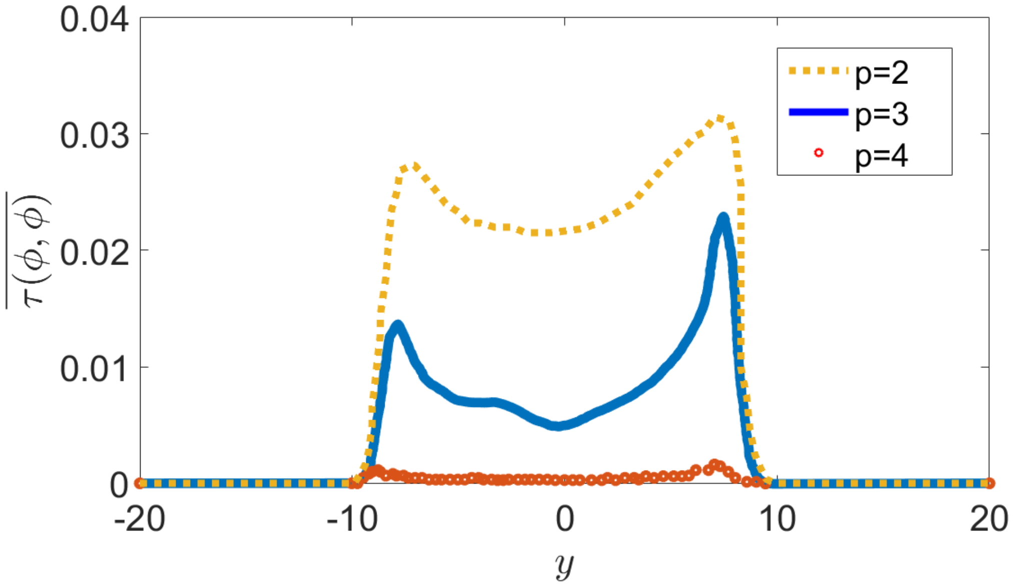

Figure 16: Averaged SGS variance at $t = 40$ with 64×64 resolution.

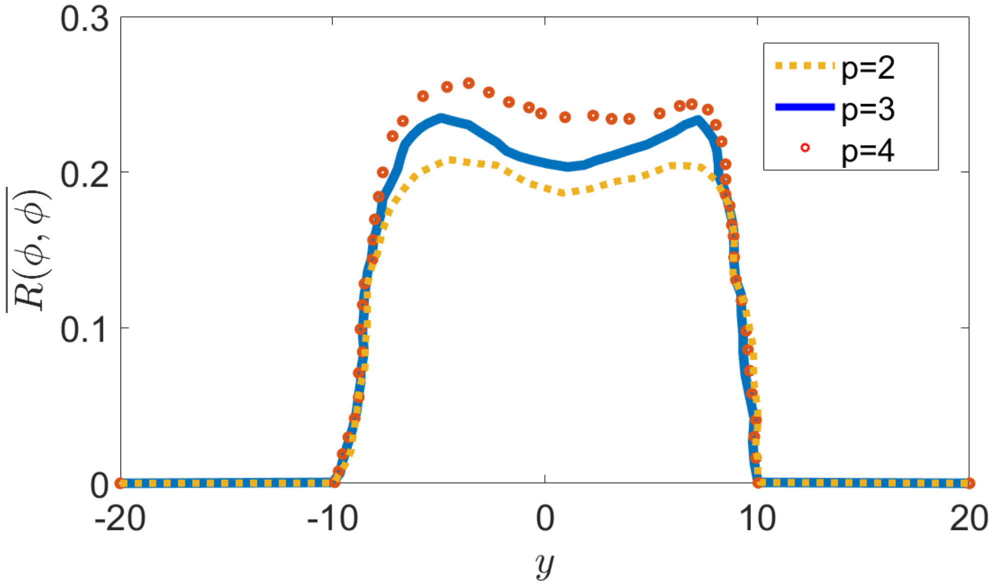

Figure 17: Averaged resolved variance at $t = 40$ with 64×64 resolution.

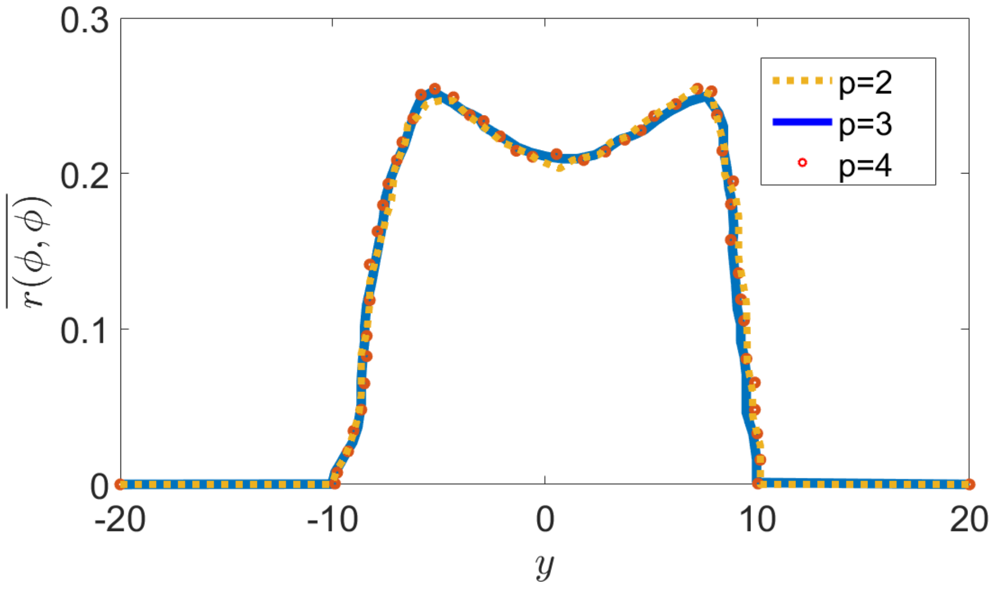

Figure 18: Averaged total stress at $t = 40$ with 64×64 resolution.

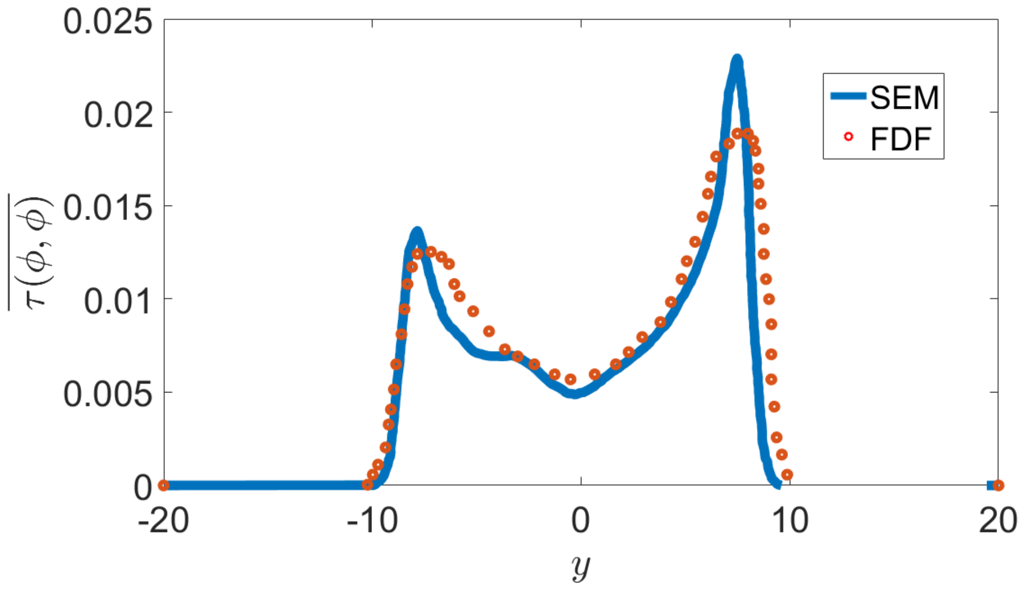

Figure 19: Cross-stream variation of the Reynolds-averaged values of SGS variance at $t = 40$, $p = 3$.

Figures 19−21 provide a demonstration of the consistency of the FDF simulator as the MC results are in agreement with those via SEM. Figure 22 shows the influence of Δ_E. It is shown for the second-order moment that the size of the ensemble domain has a significant influence on the variance. The results from FDF converge to those obtained from SEM as the size of the ensemble domain is reduced.

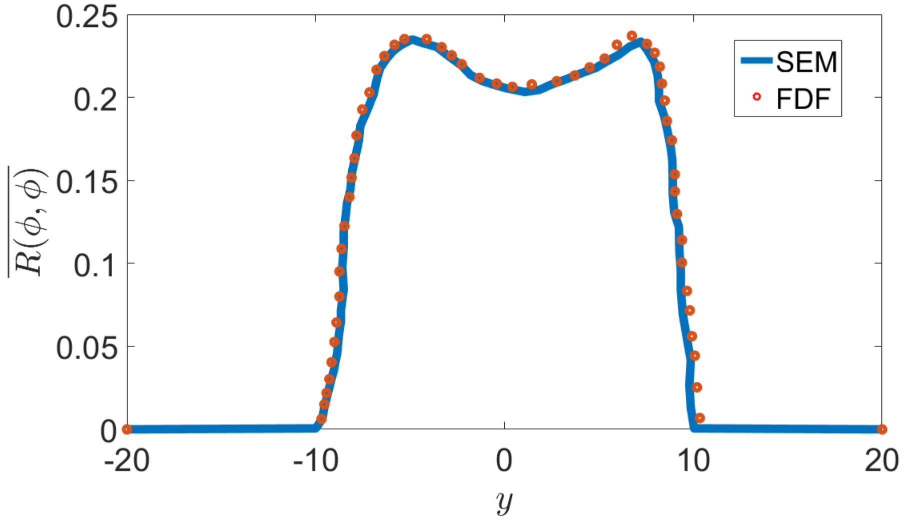

Figure 20: Cross-stream variation of the Reynolds-averaged values of the resolved variance at $t = 40$, $p = 3$.

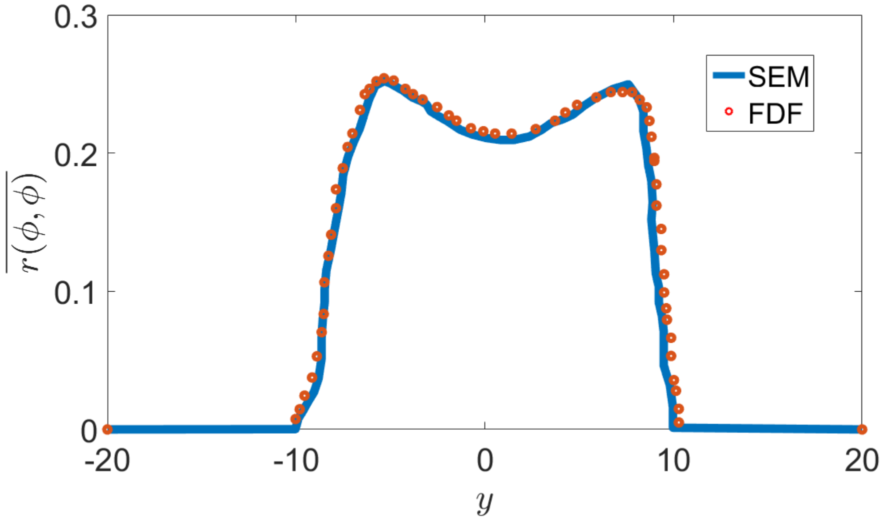

Figure 21: Cross-stream variation of the Reynolds-averaged values of the total stress at $t = 40$, $p = 3$.

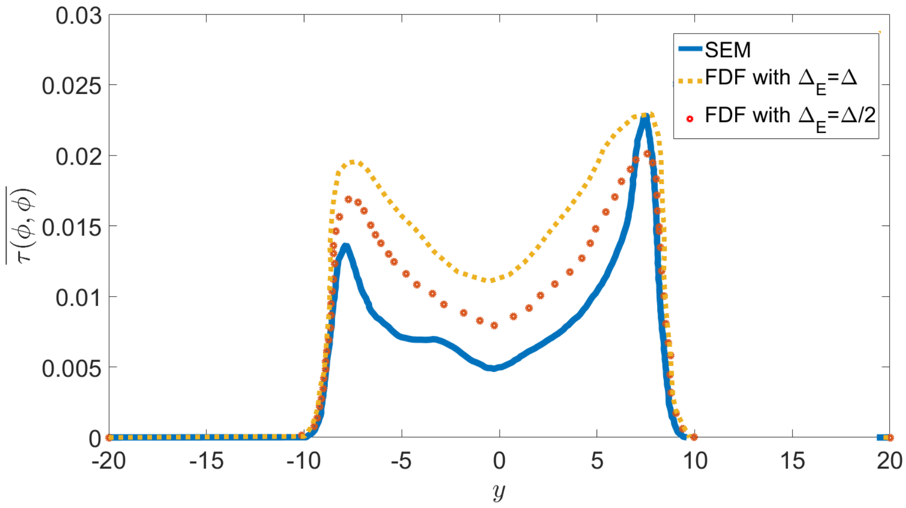

Figure 22: Cross-stream variation of the Reynolds-averaged values of the SGS variance at $t = 40$, $p = 3$.

To assess the realizability of reacting flow simulation, the compositional structure of the flame in the mixture fraction domain (ξ) is considered. As shown in Figs. 23 and 24, when the Da number is high, the composition is close to the infinitely fast reaction, and when the Da number is low, the values are close to those of the mixing only case. This is collaborated by the mass fractions of all of the other species, as shown in the scatter plots (Figs. 25 and 26) and the cross stream variation of the Reynolds averaged values (Figs. 27 and 28). The superiority of FDF is best demonstrated in Fig. 29 where the predicted values of the product mass fractions are compared with DNS data. It is shown that the FDF results are much closer to DNS than the LES results without the inclusion of the subgrid scale fluctuations. This figure demonstrates the power of FDF in accurate predictions of the turbulence-chemistry interactions.

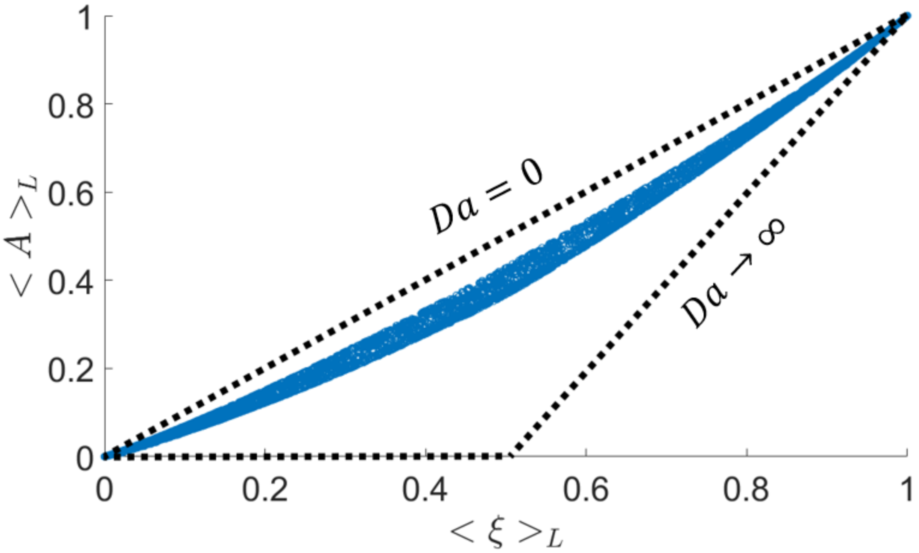

Figure 23: Scatter plots of the filtered composition variables versus the filtered mixture fraction for $Da = 1$.

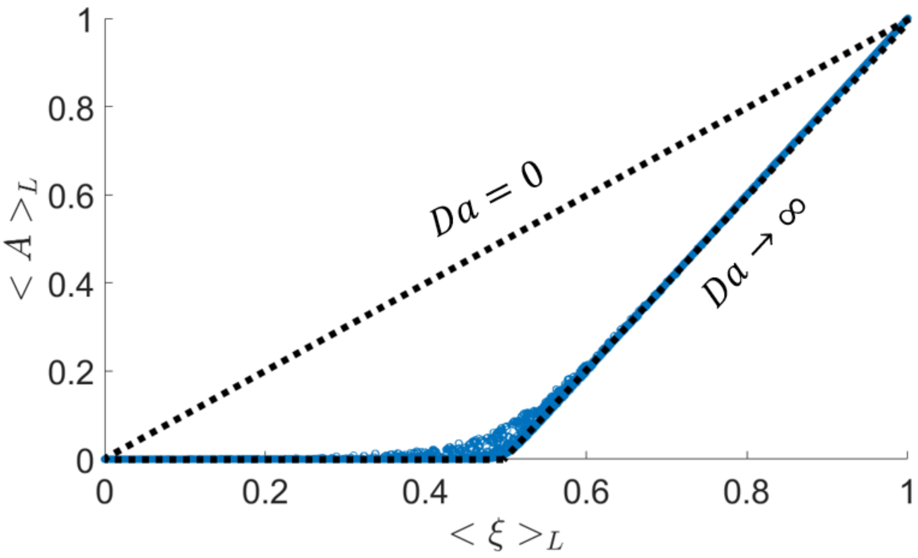

Figure 24: Scatter plots of the filtered composition variables versus the filtered mixture fraction for $Da = 10$.

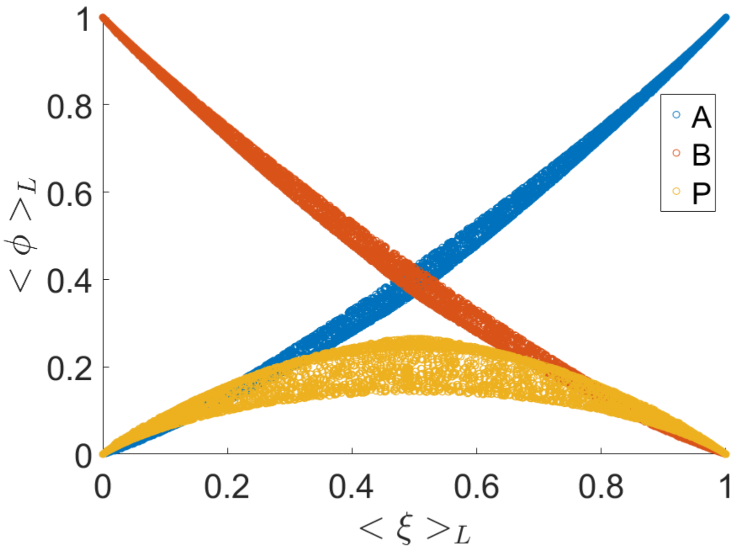

Figure 25: Scatter plots of the filtered composition variables versus the filtered mixture fractions for $Da = 1$.

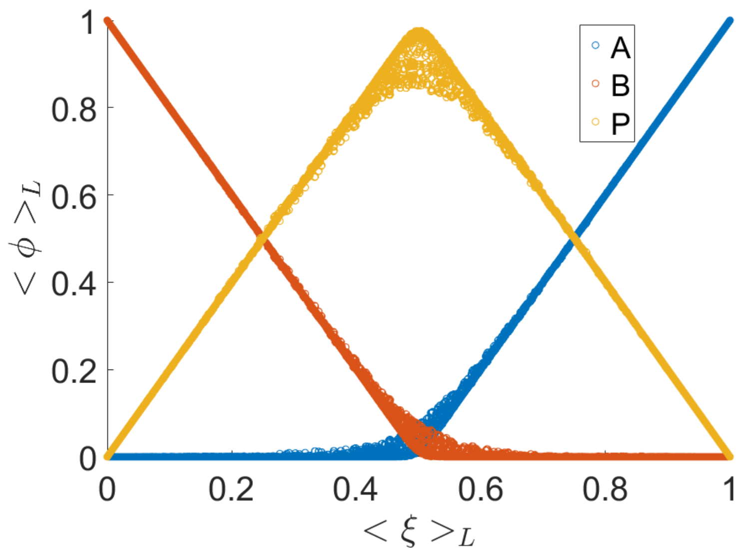

Figure 26: Scatter plots of the filtered composition variables versus the filtered mixture fraction for $Da = 10$.

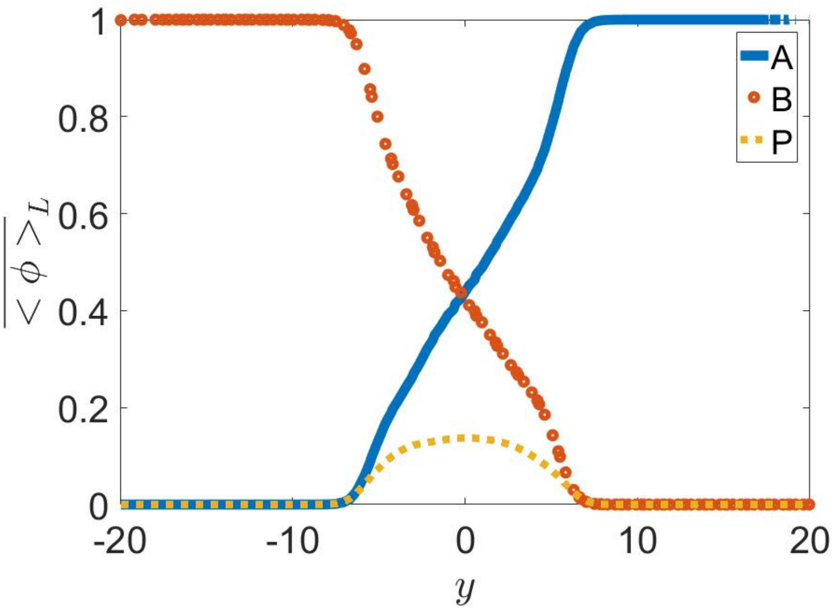

Figure 27: Mean value of the filtered mass fraction of a scalar for $Da = 1$.

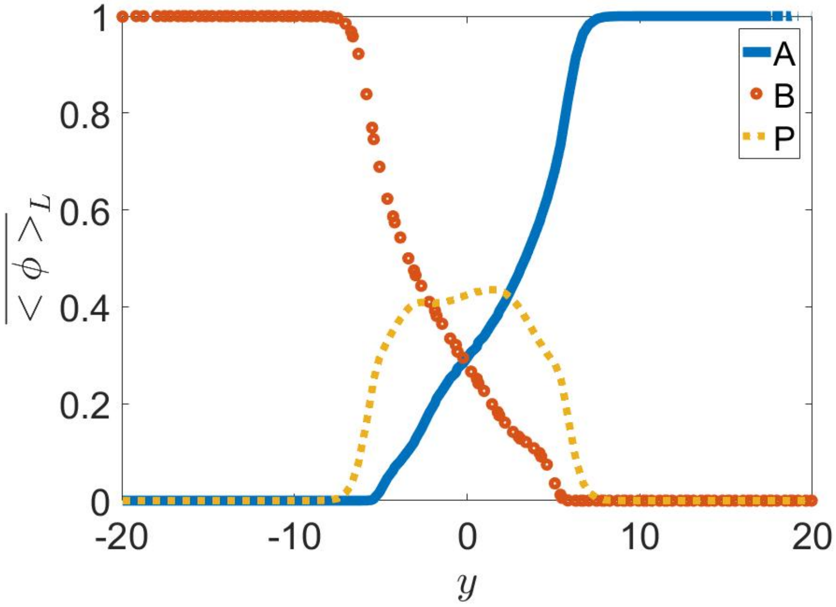

Figure 28: Mean values of the filtered scalar for $Da = 10$.

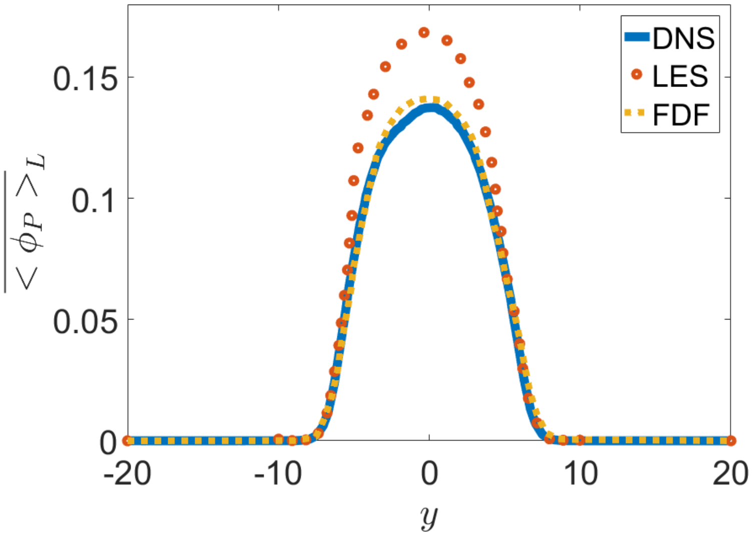

Figure 29: Cross-stream variation of the product distribution for $Da = 1$.

For further assessment of realizability, the behavior of two Shvab-Zeldovich (conserved scalar) variables are considered:

$$
\begin{aligned}
Z_1 &= A = (1 - B) \quad with\ Da = 0 \\
Z_2 &= \frac{A - \frac{B}{\nu} + \frac{B_\infty}{\nu}}{A_\infty + \frac{B_\infty}{\nu}},
\end{aligned}
\tag{4.1}
$$

where the subscript ∞ denotes the values at the free stream. It is shown that the correlation between $\overline{\langle Z_1 \rangle}$ and $\overline{\langle Z_2 \rangle}$ is excellent as shown in Fig. 30.

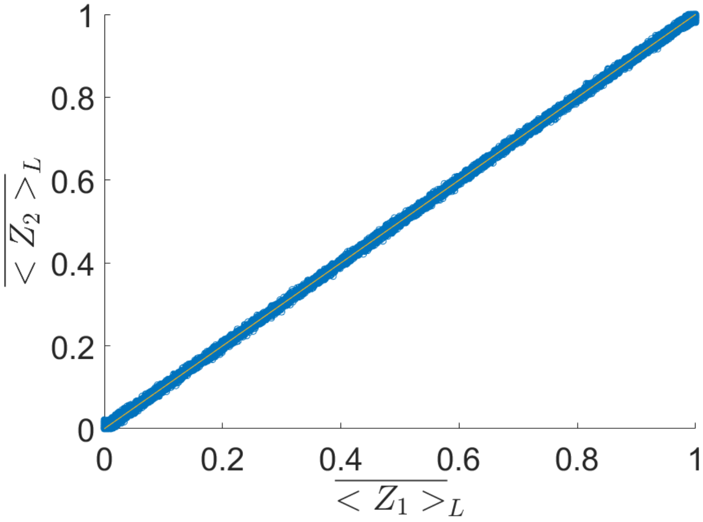

Figure 30: Scatter plot of $\overline{\langle Z_2 \rangle}$ versus $\overline{\langle Z_1 \rangle}$ for $Da = 1$.

5.0 CONCLUDING REMARKS

The subject of this dissertation is the merger of the spectral element method (SEM) with the Lagrangian Monte Carlo (MC) solution of the filtered density function (FDF). This merger provides a novel tool for conducting large eddy simulation (LES) of turbulent reacting flows. A two-dimensional temporally developing mixing layer is considered under both non-reacting and reacting conditions. The consistency of the methodology is assessed by comparing the first two moments of the FDF with those obtained by the SEM solutions of the same moments' transport equations. The properties of the SEM-MC simulator can be summarized as follows:

1. The SEM method combines the versatility of finite element method with the accuracy of spectral approximations and is particularly effective when utilized in conjunction with the Lagrangian MC solver.
2. The SEM solver supports combined $h-p$ refinement which results in an optimal solution accuracy for a given computational cost.
3. Even at low p values, when the resolved energy is significantly reduced, the total energy is captured accurately. This feature is particularly attractive when the prediction of the total energies (stresses) are of primary concern.
4. A significant advantage of the hybrid methodology is that it will allow us to reach to DNS limit via p-refinement. Based on the close to the exponential convergence of this refinement, the procedure is much more efficient than the conventional approach of refining the grid (reducing h) as is the practice in typical Eulerian LES.
5. A particular advantage of the approach is that the physical variables can easily be evaluated at the MC particle locations since these variables are represented by simple poly-

nomials on each element. Hence, there is no loss of accuracy due to the use of a lower order interpolation method as is used in conventional approximations.

6. Due to the local character of SEM, it enables superior scalability on massively parallel computer architectures. The solver is designed to scale to very large cases, and simulations involving several billion degrees of freedom are within reach.
7. Due to the high order polynomial approximation, the SEM mesh elements are typically much larger than the cells in FD or FV discretizations. This implies that the MC particles will remain much longer in one element as compared to that in conventional approaches, and thus the computational effort for the particle tracking algorithm will be reduced significantly.
8. In reactive flows, the result from modeling the source term using FDF is promising. This method can be extended to deal with the complex turbulent combustion problem since the computational expense does not increase exponentially when increasing the number of species.

The success of the SEM-MC FDF simulator as demonstrated here warrants its further extension and applications for LES of complex turbulent combustion problems. For future work, three-dimensional simulations are recommended to provide a more realistic setting for turbulent transport. The preliminary version of the 3-D SEM-MC code has just been completed (Fig. 31) and can be used for extensive future simulations, provided that sufficient computational resources are available.

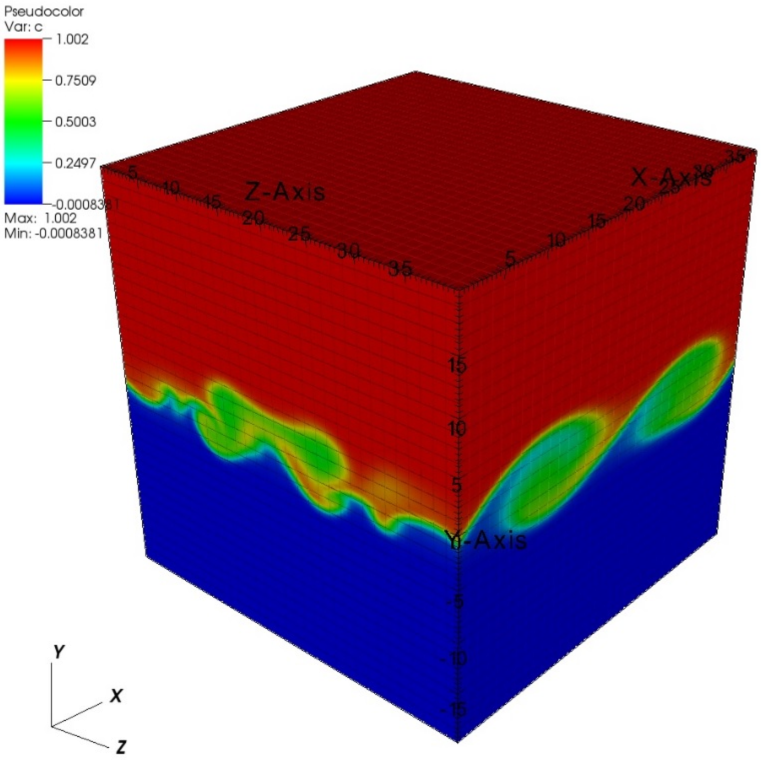

Figure 31: Contour plot of the filtered scalar filed in a three-dimensional mixing layer.

BIBLIOGRAPHY

[1] Carpenter, M. H. and Kennedy, C. A., Fourth-Order 2N-Storage Runge-Kutta Schemes, *NASA Technical Memorandum 109112*, (1994).

[2] Reynolds, O., An Experimental Investigation of the Circumstances which Determine Whether the Motion of Water Shall be Direct and Sinuous, and the Law of Resistance in Parallel Channels, *Phil. Trans. Royal Soc.*, **174**:935–982 (1883).

[3] Givi, P., Model-Free Simulations of Turbulent Reactive Flows, *Prog. Energ. Combust.*, **15**(1):1–107 (1989).

[4] Xu, H. and Bodenschatz, E., Motion of Inertial Particles with Size Larger than Kolmogorov Scale in Turbulent Flows, *Physica D*, **237**(14-17):2095–2100 (2008).

[5] Moeng, C. and Sullivan, P. P., Large-Eddy Simulation, *Encyc. Atmodpheric. Sci.*, **2**:232–240 (2015).

[6] Colucci, P. J., Jaberi, F. A., Givi, P., and Pope, S. B., Filtered Density Function for Large-Eddy Simulation of Turbulent Reacting Flows, *Phys. Fluids*, **10**(2):499–515 (1998).

[7] Garrick, S. C., Jaberi, F. A., and Givi, P., Large Eddy Simulation of Scalar Transport in a Turbulent Jet Flow, in Knight, D. and Sakell, L., editors, *Recent Advances in DNS and LES*, *Fluid Mechanics and Its Applications*, Vol. 54, pp. 155–166, Springer Netherlands, 1999.

[8] Poinsot, T. and Veynante, D., *Theoretical and Numerical Combustion*, R. T. Edwards, Inc., Philadelphia, PA, third edition, 2011.

[9] Libby, P. A. and Williams, F. A., editors, *Turbulent Reacting Flows*, *Topics in Applied Physics*, Vol. 44, Springer-Verlag, Heidelberg, 1980.

[10] Libby, P. A. and Williams, F. A., editors, *Turbulent Reacting Flows*, Academic Press, London, UK, 1994.

[11] Hawthorne, W. R., Weddell, D. S., and Hottel, H. C., Mixing and Combustion in Turbulent Gas Jets, *Third Symposium on Combustion and Flame and Explosion Phenomena*, **3**(1):266–288 (1948).

[12] Lockwood, F. and Naguib, A., The Prediction of the Fluctuations in the Properties of Free, Round-Jet, Turbulent, Diffusion Flames, *Combust. Flame*, **24**:109–124 (1975).

[13] Bray, K. and Moss, J. B., A Unified Statistical Model of the Premixed Turbulent Flame, *Acta Astronaut.*, **4**(3-4):291–319 (1977).

[14] Borghi, R., Turbulent Combustion Modelling, *Prog. Energ. Combust.*, **14**(4):245–292 (1988).

[15] Gutheil, E. and Bockhorn, H., The Effect of Multidimensional PDFs on the Turbulent Reaction-Rate in Turbulent Reacting Flows at Moderate Damkohler Numbers, *Physicochem. Hydrodyn.*, **9**(3-4):525–535 (1987).

[16] Pope, S. B., PDF Methods for Turbulent Reactive Flows, *Prog. Energ. Combust.*, **11**(2):119–192 (1985).

[17] Givi, P., Filtered Density Function for Subgrid Scale Modeling of Turbulent Combustion, *AIAA J.*, **44**(1):16–23 (2006).

[18] Pope, S. B., Computationally Efficient Implementation of Combustion Chemistry using *in situ* Adaptive Tabulation, *Combust. Theor. Model.*, **1**(1):41–63 (1997).

[19] Jaberi, F. A., Colucci, P. J., James, S., Givi, P., and Pope, S. B., Filtered Mass Density Function for Large-Eddy Simulation of Turbulent Reacting Flows, *J. Fluid Mech.*, **401**:85–121 (1999).

[20] Nouri, A. G., Nik, M. B., Givi, P., Livescu, D., and Pope, S. B., Self-Contained Filtered Density Function, *Phys. Rev. Fluids*, **2**:094603 (2017).

[21] Ansari, N., Jaberi, F. A., Sheikhi, M. R. H., and Givi, P., Filtered Density Function as a Modern CFD Tool, in Maher, A. R. S., editor, *Engineering Applications of Computational Fluid Dynamics: Volume 1*, Chapter 1, pp. 1–22, International Energy and Environment Foundation, 2011.

[22] Jaberi, F. A. and James, S., A Dynamic Similarity Model for Large Eddy Simulation of Turbulent Combustion, *Phys. Fluids*, **10**(7):1775–1777 (1998).

[23] Sheikhi, M. R. H., Drozda, T. G., Givi, P., Jaberi, F. A., and Pope, S. B., Large Eddy Simulation of a Turbulent Nonpremixed Piloted Methane Jet Flame (Sandia Flame D), *Proc. Combust. Inst.*, **30**(1):549–556 (2005).

[24] Haworth, D. C., Progress in Probability Density Function Methods for Turbulent Reacting Flows, *Prog. Energ. Combust.*, **36**(2):168–259 (2010).

[25] Haworth, D. C. and Pope, S. B., Transported Probability Density Function Methods for Reynolds-Averaged and Large-Eddy Simulations, in Echekki, T. and Mastorakos, E., editors, *Turbulent Combustion Modeling*, *Fluid Mechanics and Its Applications*, Vol. 95, pp. 119–142, Springer Netherlands, 2011.

[26] Kuo, K. K. and Acharya, R., *Fundamentals of Turbulent and Multiphase Combustion*, John Wiley and Sons Inc., Hoboken, NJ, 2012.

[27] Pope, S. B., Small Scales, Many Species and the Manifold Challenges of Turbulent Combustion, *Proc. Combust. Inst.*, **34**(1):1–31 (2013).

[28] Heinz, S., *Statistical Mechanics of Turbulent Flows*, Springer-Verlag, New York, NY, 2003.

[29] Sheikhi, M. R. H., Drozda, T. G., Givi, P., and Pope, S. B., Velocity-Scalar Filtered Density Function for Large Eddy Simulation of Turbulent Flows, *Phys. Fluids*, **15**(8):2321–2337 (2003).

[30] Sheikhi, M. R. H., Givi, P., and Pope, S. B., Velocity-Scalar Filtered Mass Density Function for Large Eddy Simulation of Turbulent Reacting Flows, *Phys. Fluids*, **19**(9):095106 (2007).

[31] Sheikhi, M. R. H., Givi, P., and Pope, S. B., Frequency-Velocity-Scalar Filtered Mass Density Function for Large Eddy Simulation of Turbulent Flows, *Phys. Fluids*, **21**(7):075102 (2009).

[32] Nik, M. B., Yilmaz, S. L., Givi, P., Sheikhi, M. R. H., and Pope, S. B., Simulation of Sandia Flame D using Velocity-Scalar Filtered Density Function, *AIAA J.*, **48**(7):1513–1522 (2010).

[33] Yilmaz, S. L., Ansari, N., Pisciuneri, P. H., Nik, M. B., Otis, C. C., and Givi, P., Advances in FDF Modeling and Simulation, in *47th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, San Diego, CA, 2011, AIAA-2011-5918.

[34] Ghosal, S., An Analysis of Numerical Errors in Large-Eddy Simulations of Turbulence, *J. Comput. Phys.*, **125**(1):187–206 (1996).

[35] Kravchenko, A. and Moin, P., On the Effect of Numerical Errors in Large Eddy Simulations of Turbulent Flows, *J. Comput. Phys.*, **131**(2):310–322 (1997).

[36] Fletcher, C., Comparison of Finite-Difference, Finite-Element, and Spectral Methods, in *Computational Galerkin Methods*, pp. 225–245, Springer, Berlin, Heidelberg, 1984.

[37] Zienkiewicz, O., *The Finite Element Method*, McGraw-Hill, London, 1977.

[38] Thomee, V., *Galerkin Finite Element Methods for Parabolic Problems*, Springer Series in Computational Mathematics, Springer, Berlin ,Heidelberg, 2013.

[39] Hussaini, M. Y., Kopriva, D. A., and Patera, A. T., Spectral Collocation Methods, *Appl. Numer. Math.*, **5**(3):177–208 (1989).

[40] Patera, A. T., A Spectral Element Method for Fluid Dynamics: Laminar Flow in a Channel Expansion, *J. Comput. Phys.*, **54**(3):468–488 (1984).

[41] Karniadakis, G. and Sherwin, S., *Spectral/hp Element Methods for Computational Fluid Dynamics*, Oxford University Press, New York, NY, 1999.

[42] Raman, V., Pitsch, H., and Fox, R. O., Hybrid Large-Eddy Simulation/Lagrangian Filtered-Density-Function Approach for Simulating Turbulent Combustion, *Combust. Flame*, **143**(1–2):56–78 (2005).

[43] Drozda, T. G., Sheikhi, M. R. H., Madnia, C. K., and Givi, P., Developments in Formulation and Application of the Filtered Density Function, *Flow Turbul. Combust.*, **78**(1):35–67 (2007).

[44] Yilmaz, S. L., Nik, M. B., Givi, P., and Strakey, P. A., Scalar Filtered Density Function for Large Eddy Simulation of a Bunsen Burner, *J. Propul. Power*, **26**(1):84–93 (2010).

[45] Drozda, T. G., Quinlan, J. R., Pisciuneri, P. H., and Yilmaz, S. L., Progress Toward Affordable High Fidelity Combustion Simulations for High-Speed Flows in Complex Geometries, in *48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, Atlanta, GA, 2012, AIAA-2012-4264.

[46] Banaeizadeh, A., Li, Z., and Jaberi, F. A., Compressible Scalar Filtered Density Function Model for High-Speed Turbulent Flows, *AIAA J.*, **49**(10):2130–2143 (2011).

[47] Ansari, N., Pisciuneri, P. H., Strakey, P. A., and Givi, P., Scalar-Filtered Mass-Density-Function Simulation of Swirling Reacting Flows on Unstructured Grids, *AIAA J.*, **50**(11):2476–2482 (2012).

[48] Afshari, A., Jaberi, F., and Shih, T., Large-Eddy Simulations of Turbulent Flows in an Axisymmetric Dump Combustor, *AIAA J.*, **46**(7):1576–1592 (2008).

[49] Ansari, N., Strakey, P. A., Goldin, G., and Givi, P., Filtered Density Function Simulation of a Realistic Swirled Combustor, *Proc. Combust. Inst.*, **35**(2):1433–1442 (2015).

[50] Sammak, S., Nouri, A. G., Ansari, N., and Givi, P., Quantum Computing and Its Potential for Turbulence Simulations, in Danaev, N., Shokin, Y., and Akhmed-Zakin, D., editors, *Mathematical Modeling of Technological Processes*, Communications in Computer and Information Science, Chapter 13, pp. 124–132, Springer, 2015.

[51] Sammak, S., Brazell, M. J., Givi, P., and Mavriplis, D. J., A Hybrid DG-Monte Carlo FDF Simulator, *Comput. Fluids*, **140**:158–166 (2016).

[52] Sammak, S., Nouri, A. G., Brazell, M. J., Mavriplis, D. J., and Givi, P., Discontinuous Galerkin-Monte Carlo Solver for Large Eddy Simulation of Compressible Turbulent Flows, in *55th AIAA Aerospace Sciences Meeting*, pp. 1–13, Grapevine, TX, 2017, AIAA, AIAA-2017-0982.

[53] Pisciuneri, P. H., Yilmaz, S. L., Strakey, P., and Givi, P., An Irregularly Portioned FDF Simulator, *SIAM J. Sci. Comput.*, **35**(4):C438–C452 (2013).

[54] Nielsen, F., Introduction to MPI: The Message Passing Interface, in *Introduction to HPC with MPI for Data Science*, pp. 21–62, Springer, Switzerland, 2016.

[55] Geurts, B., *Elements of Direct and Large-Eddy Simulation*, R. T. Edwards, Inc., Philadelphia, PA, 2004.

[56] Vreman, B., Geurts, B., and Kuerten, H., Subgrid-Modelling in LES of Compressible Flow, *Appl. Sci. Res.*, **54**(3):191–203 (1995).

[57] Smagorinsky, J., General Circulation Experiments with the Primitive Equations. I. The Basic Experiment, *Mon. Weather Rev.*, **91**(3):99–164 (1963).

[58] Ghosal, S. and Moin, P., The Basic Equations for the Large Eddy Simulation of Turbulent Flows in Complex Geometry, *J. Comput. Phys.*, **118**(1):24–37 (1995).

[59] O'Brien, E. E., The Probability Density Function (PDF) Approach to Reacting Turbulent Flows, in Libby, P. and Williams, f., editors, *Turbulent Reacting Flows*, *Topics in Applied Physics*, Vol. 44, Chapter 5, pp. 185–218, Springer, Berlin, Germany, 1980.

[60] Vreman, B., Geurts, B., and Kuerten, H., Realizability Conditions for the Turbulent Stress Tensor in Large-Eddy Simulation, *J. Fluid Mech.*, **278**:351–362 (1994).

[61] Pope, S. B., *Turbulent Flows*, Cambridge University Press, Cambridge, U.K., 2000.

[62] Gikhman, I. I. and Skorokhod, A. V., *Stochastic Differential Equations*, Springer-Verlag, New York, NY, 1972.

[63] Risken, H., *The Fokker-Planck Equation, Methods of Solution and Applications*, Springer-Verlag, New York, NY, 1989.

[64] Kopriva, D. A. and Kolias, J. H., A Conservative Staggered-Grid Chebyshev Multidomain Method for Compressible Flows, *J. Comput. Phys.*, **125**(1):244–261 (1996).

[65] Kopriva, D. A., A Staggered-Grid Multidomain Spectral Method for the Compressible Navier-Stokes Equations, *J. Comput. Phys.*, **143**(1):125–158 (1998).

[66] Fletcher, C. A., *Computational Techniques for Fluid Dynamics*, Springer-Verlag New York, Inc., New York, NY, USA, 1988.

[67] Maday, Y., Mavriplis, C., and Patera, A., Nonconforming Mortar Element Methods: Application to Spectral Discretizations, *SIAM*, pp. 392–418 (1988).

[68] Roe, P., Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes, *J. Comput. Phys.*, **43**(2):357–372 (1981).

[69] Harten, A. and Hyman, J. M., Self Adjusting Grid Methods for One-Dimensional Hyperbolic Conservation Laws, *J. Comput. Phys.*, **50**(2):235–269 (1983).

[70] Williamson, J., Low-Storage Runge-Kutta Schemes, *J. Comput. Phys.*, **35**(1):48–56 (1980).

[71] Pope, S., Monte Carlo Calculations of Premixed Turbulent Flames, *Proc. Combust. Inst.*, **18**(1):1001 – 1010 (1981).

[72] Pope, S. B., A Monte Carlo Method for the PDF Equations of Turbulent Reactive Flow, *Combust. Sci. Technol.*, (1981).

[73] Subramaniam, S. and Haworth, D. C., A Probability Density Function Method for Turbulent Mixing and Combustion on Three-Dimensional Unstructured Deforming Meshes, *Int. J. Engine Res.*, **1**(2):171–190 (2000).

[74] Gicquel, L. Y., Gourdain, N., Boussuge, J.-F., Deniau, H., Staffelbach, G., Wolf, P., and Poinsot, T., High Performance Parallel Computing of Flows in Complex Geometries, *Comptes Rendus Mécanique*, **339**(2):104–124 (2011).

[75] Karypis, G. and Kumar, V., *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*, Minneapolis, MN, 1998.

[76] Gropp, W., Lusk, E., and Skjellum, A., *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, MIT Press, Cambridge, MA, 1999.

[77] Givi, P., Sheikhi, M. R. H., Drozda, T. G., and Madnia, C. K., Large Scale Simulation of Turbulent Combustion, *Combust. Plasma Chem.*, **6**(1):1–9 (2008).

[78] Moser, R. D. and Rogers, M. M., The Three-Dimensional Evolution of a Plane Mixing Layer: Pairing and Transition to Turbulence, *J. Fluid Mech.*, **247**:275–320 (1993).

[79] Sandham, N. D. and Reynolds, W. C., Three-Dimensional Simulations of Large Eddies in the Compressible Mixing Layer, *J. Fluid Mech.*, **224**:133–158 (1991).

[80] Ainsworth, M. and Senior, B., An Adaptive Refinement Strategy for hp-Finite Element Computations, *Appl. Numer. Math.*, **26**(1-2):165–178 (1998).

[81] Boyd, J. P., *Chebyshev and Fourier Spectral Methods*, Dover, New York, NY, 2001.