Seventh International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, Hawaii, July 9-13, 2012

ICCFD7-2704

Towards Petascale Large Eddy Simulation of Reacting Flow

S.L. Yilmaz*, P.H. Pisciuneri** and P. Givi**
Corresponding author: slyilmaz@pitt.edu

* Center for Simulation and Modeling, University of Pittsburgh, USA.
** Mechanical Engineering and Materials Science, University of Pittsburgh, USA.

Abstract: A novel computational methodology, termed "Irregularly Portioned Lagrangian Monte Carlo-Finite Difference" (IPLMCFD) is developed for large eddy simulation (LES) of turbulent flows. This methodology is intended for use in the filtered density function (FDF) formulation and is particularly suitable for simulation of chemically reacting flows on massively parallel platforms. The IPLMCFD facilitates efficient simulations, and thus allows reliable prediction of complex turbulent flames. It allows for tremendous improvements in scalability, and is the key enabler of petascale computations.

Keywords: Turbulence, Combustion, Large Eddy Simulation, Filtered Density Function, Petascale Parallelization, Chemical Kinetics.

1 Introduction

Within the past decade, there has been significant progress in large scale simulation (LES) of turbulent combustion [1, 2]. The primary challenge in large eddy simulation (LES) is accurate modeling of the subgrid scale (SGS) quantities [1–8]. The filtered density function (FDF) methodology; including its mass weighted form, the filtered mass density function (FMDF), has proven particularly effective for this purpose [9, 10]. The FDF is essentially the counterpart of the probability density function (PDF) methods in Reynolds averaged simulations, commonly referred to as the Reynolds-averaged Navier-Stokes (RANS) [5, 8, 11].

The most sophisticated FDF closure available to-date is our frequency-velocity-scalar FMDF (FVS-FMDF) [12], and a simpler version (VS-FMDF) which does not include the frequency [13,14]. Hydrodynamic closure in incompressible, non-reacting flows has been successfully achieved via the velocity-FDF (V-FDF) [15], and the one which has been utilized by most other investigators only considers the scalar field (S-FDF and S-FMDF). This is the most elementary form of FDF when we first introduced it [16,17]. The first LES of a hydrocarbon flame, namely the Sandia-Darmstadt piloted diffusion flame [18, 19] were conducted via both S-FMDF [20] and VS-FMDF [21, 22]. The FDF has also been successful in predicting the more complex field of several other turbulent flames [23–25].

The original work of Colucci *et al.* [16] provides the first demonstration of a *transported* FDF. Since then, this methodology has experienced widespread usage, and is now regarded as one of the most effective and popular means of LES worldwide. Some of the important contributions in FDF by others are in its basic implementation [26–41], fine-tuning of its sub-closures [42–44] and its

validation via laboratory experiments [30, 45–49]. The FDF is finding its way into industry, into commercial codes, and has received broad coverage in several text- and hand-books [1, 5–7, 50–52]. For a recent review, please see Ref. [10].

A major challenge associated with FDF is the amount of computational resources it requires. Petascale computing has been a reality for research for the past couple of years, and exascale platforms are the current technological trend which are expected to be available by the end of this decade [53]. Being able to take advantage of the enormous opportunity of such extreme scale computing platforms is vital to the success of LES and its application in industrial practice.

Design and implementation of scalable parallel algorithms is the key enabler in the petascale arena. However, this is not a trivial task, especially in LES of reacting flows where complex chemistry calculations typically dominate the computation. The issue is further exacerbated by the dynamic and inhomogeneous nature of the flow. The variation in composition affects the level of stiffness of the chemistry equations, which in turn causes the computational load to vary significantly throughout the domain and duration of a simulation. This is a common scenario in simulating all but the most trivial of configurations, and straightforward parallelization techniques are ineffective with limited scalability.

An effective parallelization must be adaptive and be driven by the dynamics of the flow. For this purpose, here we develop and implement such a methodology termed "Irregularly Portioned Lagrangian Monte Carlo Finite Difference" (IPLMCFD). This is a dynamic load balancing technique that is very effective for structured mesh configurations, but is also applicable to unstructured meshes. It allows for efficient LES of reacting flow on thousands of computing units, and in the context of FDF, has the potential to scale to hundreds of thousands of computing units by readily being able to take advantage of the heterogeneous multicore nature of today's petascale systems.

2 LES via FDF

The primary transport variables in exothermic, chemically reacting, variable density flows, are the fluid density $\rho(\mathbf{x},t)$, the velocity vector $u_i(\mathbf{x},t), i=1,2,3$ along the x_i direction, the specific enthalpy $h(\mathbf{x},t)$, the pressure $p(\mathbf{x},t)$, and the mass fractions of N_s species, $Y_\alpha(\mathbf{x},t)\,(\alpha=1,2,\ldots,N_s)$, where $\mathbf{x}\equiv x_i (i=1,2,3)$ and t denote space and time, respectively. Implementation of LES involves the use of the spatial filtering operation [54, 55]

$$\langle Q(\mathbf{x},t)\rangle_\ell = \int_{-\infty}^{+\infty} Q(\mathbf{x}',t)\,\mathcal{H}(\mathbf{x}',\mathbf{x})\,\mathrm{d}\mathbf{x}' \tag{1}$$

where $\mathcal{H}$ denotes the filter function of width $\Delta_{\mathcal{H}}$, and $\langle Q(\mathbf{x},t)\rangle_\ell$ represents the filtered value of the transport variable $Q(\mathbf{x},t)$. In reacting flows, it is convenient to consider the Favré filtered quantity, $\langle Q(\mathbf{x},t)\rangle_L = \langle\rho Q\rangle/\langle\rho\rangle$. The transport variables satisfy the conservation equations of mass, momentum, energy and species mass fractions [56]. The filtered form of these equations are:

$$\frac{\partial\langle\rho\rangle}{\partial t} + \frac{\partial\langle\rho\rangle\langle u_i\rangle_L}{\partial x_i} = 0 \tag{2}$$

$$\frac{\partial\langle\rho\rangle\langle u_j\rangle_L}{\partial t} + \frac{\partial\langle\rho\rangle\langle u_i\rangle_L\langle u_j\rangle_L}{\partial x_i} = -\frac{\partial\langle p\rangle}{\partial x_j} + \frac{\partial\langle\tau_{ij}\rangle}{\partial x_i} - \frac{\partial T_{ij}}{\partial x_i} \tag{3}$$

$$\frac{\partial\langle\rho\rangle\langle\phi_\alpha\rangle_L}{\partial t} + \frac{\partial\langle\rho\rangle\langle u_i\rangle_L\langle\phi_\alpha\rangle_L}{\partial x_i} = -\frac{\partial\langle J_i^\alpha\rangle}{\partial x_i} - \frac{\partial M_i^\alpha}{\partial x_i} + \langle\rho S_\alpha\rangle \tag{4}$$

where τ_{ij} and J_i^α denote the viscous stress tensor and the scalar fluxes, respectively. In Eq. (4), S_α denotes the source term and this equation represents transport of the species mass fractions and enthalpy in a common form with $\phi_\alpha \equiv Y_\alpha,\ \alpha = 1, 2, \ldots, N_s,\ \phi_\sigma \equiv h,\ \sigma = N_s + 1$. The SGS closure problem is associated with $T_{ij} = \langle\rho\rangle(\langle u_i u_j\rangle_L - \langle u_i\rangle_L\langle u_j\rangle_L)$, $M_i^\alpha = \langle\rho\rangle(\langle u_i\phi_\alpha\rangle_L - \langle u_i\rangle_L\langle\phi_\alpha\rangle_L)$, and $\langle\rho S_\alpha\rangle$. The FDF provides an effective means for this closure. For the scalars' array $\boldsymbol{\phi}(\mathbf{x}, t)$ and the velocity field, $\boldsymbol{u}(\mathbf{x}, t)$, the SGS statistical information is included in the joint velocity-scalar filtered mass density function (VS-FMDF), denoted by $\mathcal{F}(\boldsymbol{v}, \boldsymbol{\psi}, \mathbf{x}, t)$, where $(\boldsymbol{v}, \boldsymbol{\psi})$ denote the probability-space for the $(\boldsymbol{u}, \boldsymbol{\phi})$ fields. The transport equation for this FDF is [13]:

$$\begin{aligned}\frac{\partial\mathcal{F}}{\partial t} + \frac{\partial\left(v_k\mathcal{F}\right)}{\partial x_k} &= \frac{\partial}{\partial v_k}\left[\left\langle\frac{1}{\rho}\frac{\partial p}{\partial x_k}\,\middle|\,\boldsymbol{v},\boldsymbol{\psi},\theta\right\rangle\mathcal{F}\right] + \frac{\partial}{\partial\psi_\alpha}\left[\left\langle\frac{1}{\rho}\frac{\partial J_j^\alpha}{\partial x_j}\,\middle|\,\boldsymbol{v},\boldsymbol{\psi},\theta\right\rangle\mathcal{F}\right] \\ &- \frac{\partial}{\partial v_k}\left[\left\langle\frac{1}{\rho}\frac{\partial\tau_{kj}}{\partial x_j}\,\middle|\,\boldsymbol{v},\boldsymbol{\psi},\theta\right\rangle\mathcal{F}\right] - \frac{\partial}{\partial\psi_\alpha}\left[S_\alpha(\boldsymbol{\psi})\mathcal{F}\right] \end{aligned} \tag{5}$$

where $\langle\,|\,\rangle$ denotes the conditional filtered values. As Eq. (5) shows, the effects of SGS convection and combustion are in *closed* forms. However, all of the terms involving conditional filtered values require closures. The scalar FMDF $\mathcal{F}_{\boldsymbol{\phi}}(\boldsymbol{\psi}, \mathbf{x}, t)$ (the marginal FMDF of the scalar field) is obtained by integration of the VS-FMDF over the velocity domain:

$$\frac{\partial\mathcal{F}_{\boldsymbol{\phi}}}{\partial t} + \frac{\partial\left[\langle u_i(\mathbf{x},t)|\boldsymbol{\psi}\rangle\,\mathcal{F}_{\boldsymbol{\phi}}\right]}{\partial x_i} = \frac{\partial}{\partial\psi_\alpha}\left[\left\langle\frac{1}{\rho}\frac{\partial J_j^\alpha}{\partial x_j}|\boldsymbol{\psi}\right\rangle\mathcal{F}_{\boldsymbol{\phi}}\right] - \frac{\partial}{\partial\psi_\alpha}\left[S_\alpha(\boldsymbol{\psi})\mathcal{F}_{\boldsymbol{\phi}}\right] \tag{6}$$

Again, the effects of chemical reaction appear in a closed form. However, in this case, the SGS convection (second term on the LHS) requires a closure, and a conventional SGS model is typically employed [17, 24, 26–40, 42, 43].

Currently, the most successful FDF closure is based on the generalized Langevin model (GLM) and the linear mean square estimation (LMSE) model as used in RANS [5]. Assuming a constant value for $\mu = \gamma$; *i.e.* unity Schmidt (Sc) and Lewis (Le) numbers, the model is of the form of coupled stochastic differential equations [13]:

$$dX_i^+ = U_i^+\,dt + \sqrt{\frac{2\mu}{\langle\rho\rangle_\ell}}\,dW_i, \tag{7a}$$

$$\begin{aligned}dU_i^+ &= \left[-\frac{1}{\langle\rho\rangle_\ell}\frac{\partial\langle p\rangle_\ell}{\partial x_i} + \frac{2\mu}{\langle\rho\rangle_\ell}\frac{\partial^2\langle u_i\rangle_L}{\partial x_j\partial x_j} + \frac{\mu}{\langle\rho\rangle_\ell}\frac{\partial^2\langle u_j\rangle_L}{\partial x_j\partial x_i} - \frac{2}{3}\frac{\mu}{\langle\rho\rangle_\ell}\frac{\partial^2\langle u_j\rangle_L}{\partial x_i\partial x_j}\right]dt \\ &+ G_{ij}\left(U_j^+ - \langle u_j\rangle_L\right)dt + \sqrt{C_0\epsilon}\,dW_i' + \sqrt{\frac{2\mu}{\langle\rho\rangle_\ell}}\frac{\partial\langle u_i\rangle_L}{\partial x_j}\,dW_j,\end{aligned} \tag{7b}$$

$$d\phi_\alpha^+ = -C_\phi\,\omega\left(\phi_\alpha^+ - \langle\phi_\alpha\rangle_L\right)dt + S_\alpha(\boldsymbol{\phi}^+)\,dt, \tag{7c}$$

$$G_{ij} = -\omega\left(\frac{1}{2} + \frac{3}{4}C_0\right)\delta_{ij},\quad \omega = \frac{\epsilon}{k},\quad \epsilon = C_\epsilon\frac{k^{3/2}}{\Delta_L},\quad k = \frac{1}{2}\left[\langle u_i u_i\rangle_L - \langle u_i\rangle_L\langle u_i\rangle_L\right]. \tag{7d}$$

where dW_i is the Wiener-Levy process [57], $X_i^+, U_i^+, \phi_\alpha^+$ are probabilistic representations of position, velocity vector, and scalar variables, respectively. In the model, ω is the SGS mixing frequency, ϵ is the dissipation rate, k is the SGS kinetic energy, and Δ_L is the LES filter size. The parameters C_0, C_ϕ and C_ϵ are model constants and need to be specified.

For the marginal SFMDF, the model for the scalar transport is the same. But the velocity field

must be obtained by other (non-FDF) means. In this case, the physical transport is modeled via

$$dX_i(t) = \left[\langle u_i \rangle_L + \frac{1}{\langle \rho \rangle} \frac{\partial(\gamma + \gamma_t)}{\partial x_i} \right] dt + \sqrt{2(\gamma + \gamma_t)/\langle \rho \rangle}\, dW_i(t), \tag{8}$$

In this equation, γ_t is the SGS diffusivity which is modeled with a gradient diffusion model, equivalent to the one used for the SGS flux of the scalars.

3 Simulations

An effective way to solve the FDF numerically is via Lagrangian Monte Carlo (MC) methods [58, 59]. These methods have been the primary means of solving the PDF in RANS [5, 60] and, thus far, the most effective choice for solving the FDF in LES. In the Lagrangian setting, the FDF is represented by an ensemble of *particles*. Each of these particles carry information pertaining to the physical field (*i.e.* each of the scalar and velocity variables) and also the position vector. These properties are updated via temporal integration of the modeled SDEs [61].

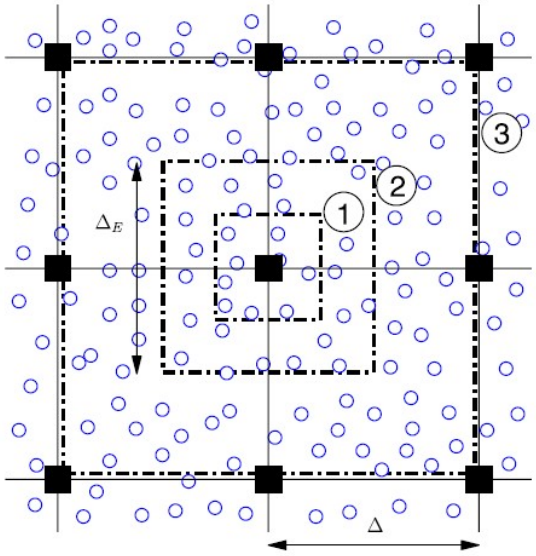

Figure 1: Two-dimensional schematic of the hybrid solver.

The numerical solution procedure is based on a hybrid scheme in which the Lagrangian MC solution of FDF provides the unclosed terms in the filtered Eulerian field Eqs. (2-4). These equations are in turn solved by a conventional CFD technique, such as the second-order in time, fourth-order in space finite difference (FD) scheme of Gottlieb and Turkel [62] as employed previously [24]. This is conducted on a structured three-dimensional mesh, superimposed by the grid-free MC domain. A typical two-dimensional slice of the domain is presented in Fig. 1. The two solvers must communicate with each other, and this communication is achieved through interpolation and ensemble averaging. This averaging is performed by considering an ensemble of particles, N_E, about an FD point with an ensemble domain side length of Δ_E. As N_E increases, and Δ_E decreases the ensemble statistics will approach the filtered values. Generally speaking, enlarged ensemble domains have the advantage of increased N_E, whereas small ensemble domains have the advantage of decreased Δ_E. Communication from the Eulerian solver to the Lagrangian solver is achieved through interpolation. For instance, the filtered hydrodynamic values appearing in Eq. (8) must be interpolated to each particle location. This intimate coupling of the two solvers makes development of an efficient parallel algorithm a challenge.

4 Scalable Parallelization

LES via FDF is computationally more expensive than the conventional LES. This overhead is expected considering all of the SGS statistical information that LES/FDF provides in comparison to other schemes. Furthermore, chemistry needs to be evaluated for each of the MC particles, and their count is typically about an order of magnitude more than that of the FD points, or more in cases where higher order statistics are sought. When finite-rate kinetics are considered this involves solving a stiff set of coupled nonlinear ordinary differential equations. The result is that

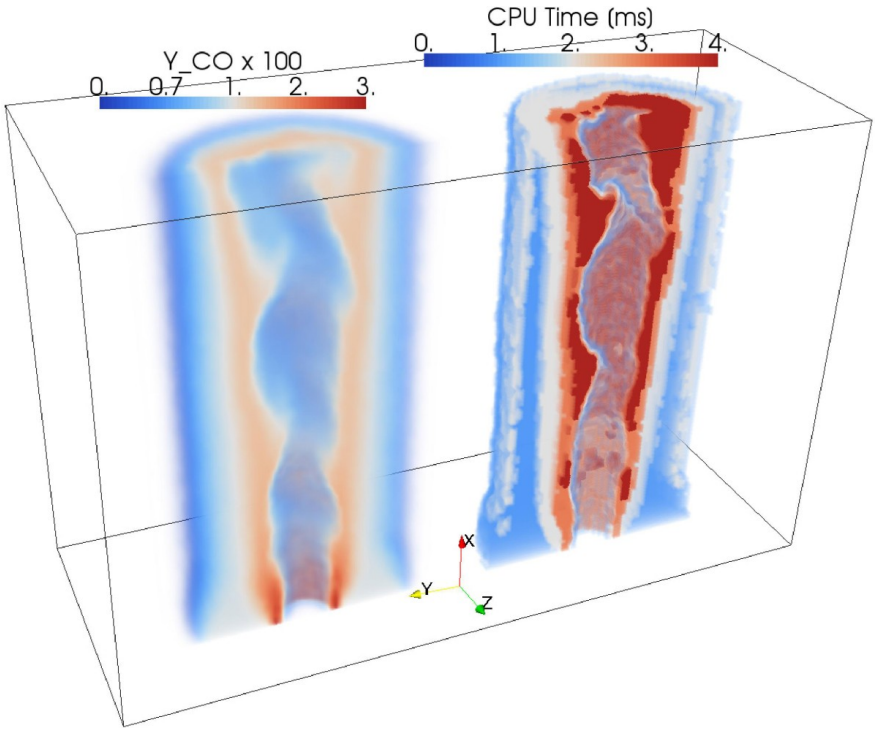

Figure 2: Instantaneous distribution of CPU requirements in the LES/FDF solver. (Left) filtered CO mass fraction field, (right) CPU time in milliseconds spent during particle computations. Transparent regions (cold center, and co-centric air) indicate negligible computation.

LES/FDF simulations for reacting flows quickly become very expensive. As outlined by Yilmaz *et al.* [63], serial algorithms would take years to execute for even moderate Reynolds numbers and grid resolution. Therefore, development of a highly scalable algorithm is necessary.

It might be mistakenly perceived that the Lagrangian MC methodology as described lends itself to embarrassing parallelism, as is the case with some particle based simulations. However, the matter is complicated by the hybrid particle/mesh nature of methodology (due to intimate coupling) and the stiffness of chemical reactions (which causes spatial load imbalance). At any instant of the simulation, different regions of the flame undergo different stages of combustion. Due to stiffness, for some states the integration of chemistry sub-step can be done very quickly (*e.g.* in cold regions), but for some others implicit integration is required (extinction/reignition regions), which is particularly expensive due to calculations of the Jacobian matrix corresponding to Eq. (7c) and dense matrix inversion for each sub-iteration. This is made evident in Fig. 2 which shows the variation in the computational load at some instant during the simulation. It is observed that the computational requirement is highly non-uniform throughout; virtually no time is spent for calculations near the cold jet or the cold surrounding air, and most of the computational load is concentrated around the hot pilot (identified by the CO levels).

A commonly employed parallelization strategy in typical structured finite-difference or finite-volume methods is the uniform and fixed-in-time block domain decomposition. Similarly, and even more commonly, in unstructured solvers a static decomposition is employed whereby the mesh is partitioned *a priori* during the preprocessing stage. With this technique, termed "uniform decomposition" here, the number of degrees-of-freedom is commonly adjusted to be equal for each partition, and work per partition is assigned to different processors at the onset of the simulation.

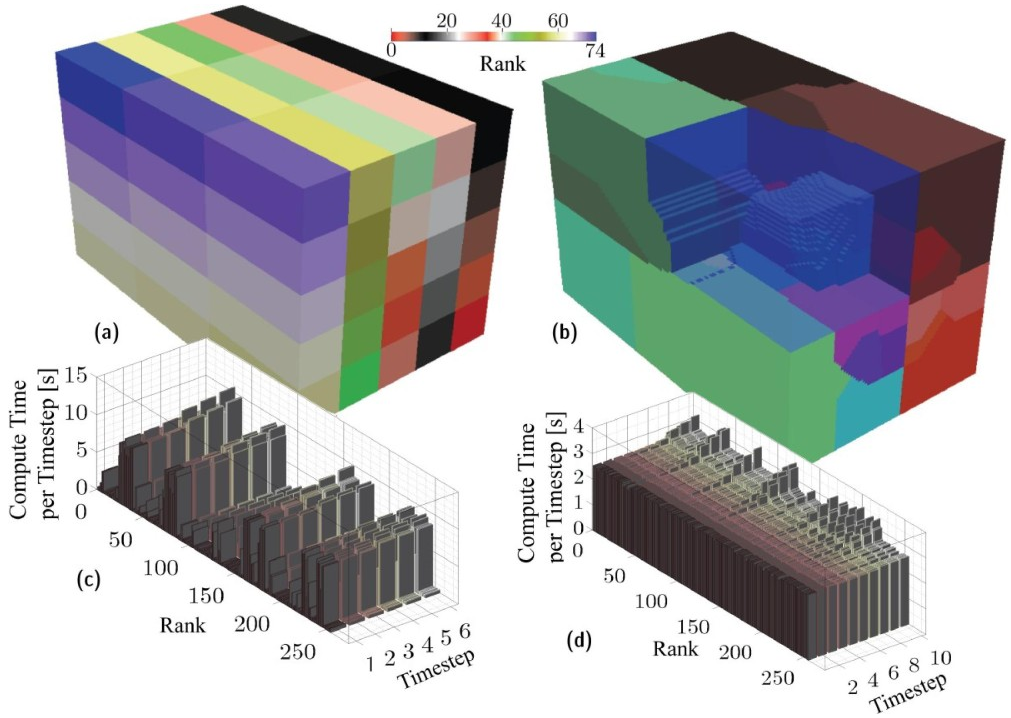

Figure 3: Domain topology with (a) the uniform decomposition, and (b) adaptive irregular decomposition. Non-idle CPU times per time-step for each rank for subsequent time-steps with (c) uniform decomposition, and (d) adaptive irregular decomposition.

This provides an effective parallelization as the communication boundaries are minimized and the messaging is relatively easy to implement. However, in general, especially in the case of reacting flows with finite-rate kinetics, uniform decomposition exhibits very poor load balancing. This feature is made evident in Fig. 3a. The result is that calculations on a few processors continue while the rest of the processors remain idle, which is quite inefficient.

The approach developed in this work is termed "Irregularly Portioned Lagrangian Monte Carlo Finite Difference" (IPLMCFD). The name captures the essence of the methodology, which is that the computational domain is decomposed into irregularly shaped and sized partitions. Each partition constitutes an entirely self-contained hybrid Eulerian/Lagrangian flow solver. The advantage to this approach is that communication between the Lagrangian and Eulerian solvers is purely local, and data exchange is limited to the communication between neighboring partition. This is typical of finite difference and particle methods, in which the communication is due to the so called *halo exchange*, where the solution on each partition boundary is communicated to the overlapping parts of the neighboring partition. The halo size is determined by the order of the spatial discretization, and is a property of the simulation which is *independent* of the number of computing units. An earlier version of the methodology, IPLMC [63] considered the load-balanced parallelization of only the MC solver in which the partitions communicated with the Eulerian solver in a collective *all-to-one* pattern. This method, while an improvement over the uniform counterpart, is not scalable beyond a few hundred computing units. Implementation of IPLMCFD is somewhat challenging but presents a significant step forward, as shown in Fig. 5.

As mentioned above, a consequence of the dynamic and unsteady nature of LES is that as the simulation proceeds, spatial distribution of the computational load varies in a transient manner following the changes in the chemical composition. Therefore, any *a priori* and static decomposi-

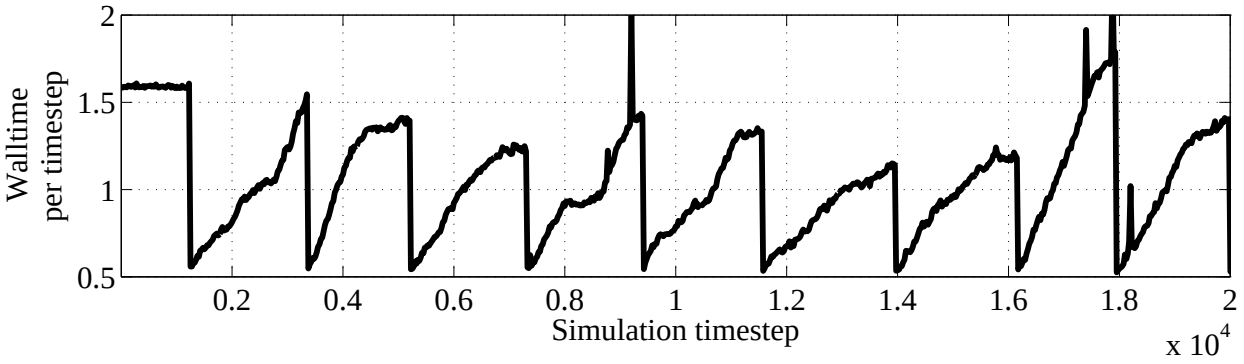

Figure 4: Evolution of the walltime-per-iteration using 4000 cpu cores.

tion, no matter how balanced, would not facilitate optimum scalability throughout the simulation. In IPLMCFD, this is accounted for in a straightforward manner by repeated application of the algorithm throughout a give simulation, each time generating a new set of balanced partitions.

An important consideration is the frequency of such repartitioning. Ideally, each iteration begins with a perfectly balanced set of partitions. However, this is not practical as there is a significant communication cost associated with rebalancing of the load, so much so that in extreme cases all of the simulation data may need to be communicated between all computational units involved. Such communication significantly hinders scalability. Fortunately, the temporal change in the spatial distribution is relatively smooth, allowing a decomposition to remain fairly balanced for several timesteps. Figure 4 shows walltime per timestep of a simulation with 4000 computing units. Initially, the simulation is started with an unbalanced partition. Then, at around 8000 iterations, balancing is applied which provides an immediate boost in performance of more than 300%. The performance degrades as the simulation evolves for the next 2000 or so iterations, then balancing is reapplied. In its current implementation, load balancing is not optimized, and relies on repeated checkpointing (and is responsible for the sporadic spikes in the plot). A much more efficient implementation is possible by allowing for more frequent rebalancing and improved overall performance. The issue of local data migration required for such a dynamic load balancing without sacrificing scalability has been investigated by many; *e.g.* see Ref. [64]. Softwares like Zoltan [65] can easily deal with this issue.

5 Scalability of IPLMCFD

To present the scalability characteristics of the IPLMCFD methodology, the FDF simulation of a Bunsen burner [66] is conducted. There are two ways by which the scalability is assessed. (1) *Strong* scalability: the total number of degrees-of-freedom (dof) N (her, the size of the simulation) is kept constant while the number of computing units p are varied. (2) *Weak* scalability: Here, N and p are varied proportionately, keeping dof per computing unit N_p fixed. These are called *strong* and *weak* scalability. Here, only strong scaling results are presented.

The total number of FD points is $2.5M$ and the number of MC particles is approximately $25M$. The speed-up, defined as the ratio of sequential walltime (t_1) to the walltime using p processors (t_p) is considered for a varying number of processors. Timings are obtained by running the simulation for 10 full time steps and taking the average. No rebalancing of load is done for this interval, and the input/output operations (for checkpointing, and postprocessing) are omitted. The runs are conducted on Kraken, the supercomputer at the National Institute for Computational Sciences,

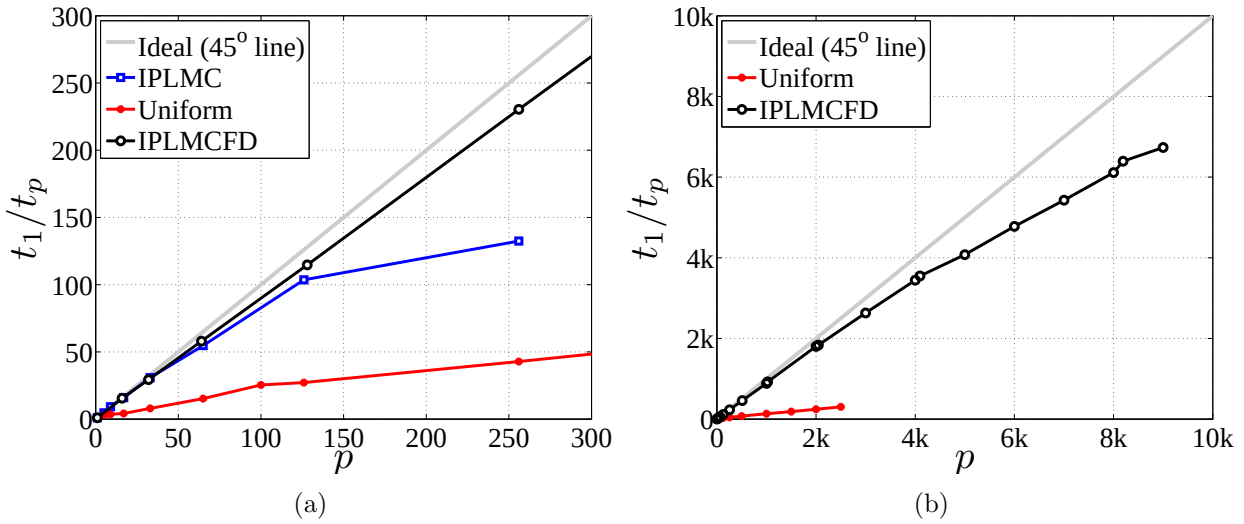

Figure 5: Strong scaling results for IPLMCFD shown in comparison with that of uniform decomposition, and earlier IPLMC [63] methodology.

Tennessee, the largest resource on National Science Foundation's (NSF) Extreme Science and Engineering Discovery Environment (XSEDE). Kraken is comprised of 9,408 compute nodes, each of which has 12 processing cores. Except for one of the cases, which is to be described shortly, all runs are performed with one processor core per partition, using pure message passing in between all cores.

From Fig. 5a it is evident that the IPLMCFD methodology outperforms uniform decomposition, as well as the IPLMC method. It is important to emphasize that the size of the simulation is kept fixed for all three methods and for all processors. Not shown are the actual walltimes. In particular, the sequential run for this configuration takes about 3,000 seconds *per iteration*[1]; with 256 processors, uniform partitioning has $45/250 = 18\%$ scalability with 66.7 seconds per time step, whereas IPLMCFD soars at 97% scalability bringing down the walltime to 11.4 seconds. In other words, a mere load balancing with the same resources provides almost a 6 fold increase in computational throughput!

Figure 5b shows that by moving into the thousands of processors range an unbalanced decomposition is simply not feasible. IPLMCFD scales almost ideally up to 4000 processors, and starts to tail off slightly beyond this limit as the effects of communication become more and more pronounced. Nevertheless, for such a moderately sized simulation, the IPLMCFD performs nicely. We project this tailing-off to surpass 10,000 processors for larger simulations with which the data-to-communication ratio is much higher.

Being able to utilize almost ten thousand processors effectively in order to conduct LES of complex hydrocarbon flames within hours, is encouraging. However, ten thousand processors do not qualify as "petascale," and will be rather insufficient given that our objective is to be able to conduct much larger, realistic, and industry scale simulations. It is important to emphasize once again that the analysis shown thus far has been for purely message passing, single processor-core per partition runs. The next phase of improvement will substantiate from recognizing the trend in today's supercomputing: *petascale computers are becoming predominantly heterogeneous* [67][2].

[1]The mesh size for this analysis is moderate enough to allow for sequential runs. In a more realistic setting memory and time constraints will prohibit this.

[2]See `http://top500.org/list/2012/06/100` for June 2012 listing

The evaluation of chemistry ODEs Eq. 7c is computationally the most significant portion of per partition workload, and for each MC particle this computation can be carried out independently. With this observation, a further local refinement can be made in a straightforward manner whereby the work for ensembles of MC particles in a given partition can be split out to individual CPU cores and/or graphical processing units (GPUs), using shared memory parallelism (for example, via OpenMP [68]), and general purpose GPU programming (for example, via NVIDIA/CUDA [69]), respectively. Given thread-safe and GPU based implementations of chemistry and stiff-solver routines, such fine grained parallelism can be implemented scalably in a straightforward fashion. This is an active area of research, and some example implementations are available [70, 71]. These efforts, combined with IPLMCFD, will provide a hybrid implementation that can scale up and beyond the petascale limit, enabling LES of reacting flows with unprecedented fidelity.

6 Summary and Conclusions

The dynamic and unsteady effects of combustion chemistry and the consequent computational load imbalance issue are some of the primary challenges in the design of scalable parallel domain decomposition algorithms. Here, we introduce the "irregularly portioned Lagrangian Monte Carlo" (IPLMCFD) method which overcomes some of these challenges in the context of a solver for the filtered density function (FDF) on structured as well as unstructured grid based solvers. We demonstrate the scalability of the algorithm via large eddy simulation of a laboratory scale flame with realistic chemistry.

The LES/FDF methodology is aptly suited to exploit the multi-core and accelerator based hybrid architecture of modern supercomputers. Our on-going work is focused on shared memory multi-threaded and general purpose GPU enabled implementations of chemistry and stiff ODE solver routines. This effort will provide the fine-grained parallelism within a partition. Then, in combination with the virtually unbounded distributed parallel scalability enabled by IPLMCFD, we will be able to leverage petascale platforms fully for high-fidelity, highly reliable LES of industry scale applications.

Acknowledgements

This work is part of a research sponsored by the National Center for Hypersonic Combined Cycle Propulsion sponsored by AFOSR and NASA under Grant FA-9550-09-1-0611; by the Department of Energy under Contract RES10000027/96 and by the National Science Foundation through Extreme Science and Engineering Discovery Environment (XSEDE) resources provided under Grant TG-CTS070055N and TG-CTS120015. Additional computational resources are provided by the Center for Simulation and Modeling at the University of Pittsburgh.

References

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

[2] Poinsot, T. and Veynante, D., *Theoretical and Numerical Combustion*, R. T. Edwards, Inc., Philadelphia, PA, 3rd edition, 2011.

[3] Pope, S. B., HOTTEL Lecture: Small Scales, Many Species and the Manifold Changes of Turbulent Combustion, *Proc. Combust. Inst.*, **34** (2012), in press.

[4] Janicka, J. and Sadiki, A., Large Eddy Simulation of Turbulent Combustion Systems, *Proc. Combust. Inst.*, **30**(1):537–547 (2005).
[5] Pope, S. B., *Turbulent Flows*, Cambridge University Press, Cambridge, U.K., 2000.
[6] Bilger, R. W., Future Progress in Turbulent Combustion Research, *Prog. Energ. Combust.*, **26**(4–6):367–380 (2000).
[7] Peters, N., *Turbulent Combustion*, Cambridge University Press, Cambridge, UK, 2000.
[8] Givi, P., Model-Free Simulations of Turbulent Reactive Flows, *Prog. Energ. Combust.*, **15**(1):1–107 (1989).
[9] Givi, P., Filtered Density Function for Subgrid Scale Modeling of Turbulent Combustion, *AIAA J.*, **44**(1):16–23 (2006).
[10] 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*, pp. 1–22, International Energy and Environment Foundation, 2011.
[11] Pope, S. B., Computations of Turbulent Combustion: Progress and Challenges, *Proc. Combust. Inst.*, **23**(1):591–612 (1990).
[12] 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).
[13] 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).
[14] 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).
[15] Gicquel, L. Y. M., Givi, P., Jaberi, F. A., and Pope, S. B., Velocity Filtered Density Function for Large Eddy Simulation of Turbulent Flows, *Phys. Fluids*, **14**(3):1196–1213 (2002).
[16] 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).
[17] 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).
[18] Barlow, R. S. and Frank, J. H., Effects of Turbulence on Species Mass Fractions in Methane/Air Jet Flames, *Proc. Combust. Inst.*, **27**(1):1087–1095 (1998).
[19] Sandia National Laboratories, TNF Workshop Website, `http://www.sandia.gov/TNF/`, 2012.
[20] 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).
[21] 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).
[22] Nik, M. B., Yilmaz, S. L., Sheikhi, M. R., and Givi, P., Grid Resolution Effects on VSFMDF/LES, *Flow Turbul. Combust.*, **85**(3–4):677–688 (2010).
[23] 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).
[24] 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).
[25] Ansari, N., Pisciuneri, P. H., Strakey, P. A., and Givi, P., SFMDF Simulation of Swirling Reacting Flows on Unstructured Grids, *AIAA J.*, (2012), in press.
[26] Zhou, X. Y. and Pereira, J. C. F., Large Eddy Simulation (2D) of a Reacting Plane Mixing Layer using Filtered Density Function Closure, *Flow Turbul. Combust.*, **64**(4):279–300 (2000).
[27] Heinz, S., On Fokker-Planck Equations for Turbulent Reacting Flows. Part 2. Filter Density Function for Large Eddy Simulation, *Flow Turbul. Combust.*, **70**(1–4):153–181 (2003).

[28] 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).

[29] Raman, V. and Pitsch, H., Large-Eddy Simulation of a Bluff-Body-Stabilized Non-Premixed Flame using a Recursive Filter-Refinement Procedure, *Combust. Flame*, **142**(4):329–347 (2005).

[30] van Vliet, E., Derksen, J. J., and van den Akker, H. E. A., Turbulent Mixing in a Tubular Reactor: Assessment of an FDF/LES Approach, *AIChE J.*, **51**(3):725–739 (2005).

[31] Carrara, M. D. and DesJardin, P. E., A Filtered Mass Density Function Approach for Modeling Separated Two-Phase Flows for LES I: Mathematical Formulation, *Int. J. Multiphas. Flow*, **32**(3):365–384 (2006).

[32] Mustata, R., Valio, L., Jimnez, C., Jones, W., and Bondi, S., A Probability Density Function Eulerian Monte Carlo Field Method for Large Eddy Simulations: Application to a Turbulent Piloted Methane/Air Diffusion Flame (Sandia D), *Combust. Flame*, **145**(1–2):88–104 (2006).

[33] Jones, W. P., Navarro-Martinez, S., and Rohl, O., Large Eddy Simulation of Hydrogen Auto-Ignition with a Probability Density Function Method, *Proc. Combust. Inst.*, **31**(2):1765–1771 (2007).

[34] Jones, W. P. and Navarro-Martinez, S., Large Eddy Simulation of Autoignition with a Subgrid Probability Density Function Method, *Combust. Flame*, **150**(3):170–187 (2007).

[35] James, S., Zhu, J., and Anand, M. S., Large Eddy Simulation of Turbulent Flames Using the Filtered Density Function Model, *Proc. Combust. Inst.*, **31**(2):1737–1745 (2007).

[36] Chen, J. Y., A Eulerian PDF Scheme for LES of Nonpremixed Turbulent Combustion with Second-Order Accurate Mixture Fraction, *Combust. Theor. Model.*, **11**(5):675–695 (2007).

[37] McDermott, R. and Pope, S. B., A Particle Formulation for Treating Differential Diffusion in Filtered Density Function Methods, *J. Comput. Phys.*, **226**(1):947–993 (2007).

[38] Raman, V. and Pitsch, H., A Consistent LES/Filtered-Density Function Formulation for the Simulation of Turbulent Flames with Detailed Chemistry, *Proc. Combust. Inst.*, **31**(2):1711–1719 (2007).

[39] 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).

[40] Drozda, T. G., Wang, G., Sankaran, V., Mayo, J. R., Oefelein, J. C., and Barlow, R. S., Scalar Filtered Mass Density Functions in Nonpremixed Turbulent Jet Flames, *Combust. Flame*, **155**(1–2):54–69 (2008).

[41] Pope, S., Self-Conditioned Fields for Large-Eddy Simulations of Turbulent Flows, *J. Fluid Mech.*, **652**:139–169 (2010).

[42] Réveillon, J. and Vervisch, L., Subgrid-Scale Turbulent Micromixing: Dynamic Approach, *AIAA J.*, **36**(3):336–341 (1998).

[43] Cha, C. M. and Trouillet, P., A Subgrid-Scale Mixing Model for Large-Eddy Simulation of Turbulent Reacting Flows using the Filtered Density Function, *Phys. Fluids*, **15**(6):1496–1504 (2003).

[44] Heinz, S., Unified Turbulence Models for LES and RANS, FDF and PDF Simulations, *Theor. Comp. Fluid Dyn.*, **21**(2):99–118 (2007).

[45] Tong, C., Measurements of Conserved Scalar Filtered Density Function in a Turbulent Jet, *Phys. Fluids*, **13**(10):2923–2937 (2001).

[46] Wang, D. and Tong, C., Conditionally Filtered Scalar Dissipation, Scalar Diffusion, and Velocity in a Turbulent Jet, *Phys. Fluids*, **14**(7):2170–2185 (2002).

[47] Rajagopalan, A. G. and Tong, C., Experimental Investigation of Scalar-Scalar-Dissipation Filtered Joint Density Function and its Transport Equation, *Phys. Fluids*, **15**(1):227–244

(2003).

[48] Wang, D., Tong, C., and Pope, S. B., Experimental Study of Velocity Filtered Joint Density Function for Large Eddy Simulation, *Phys. Fluids*, **16**(10):3599–3613 (2004).

[49] Wang, D. and Tong, C., Experimental Study of Velocity-Scalar Filtered Joint Density Function for LES of Turbulent Combustion, *Proc. Combust. Inst.*, **30**(1):567–574 (2005).

[50] Minkowycz, W. J., Sparrow, E. M., and Murthy, J. Y., editors, *Handbook of Numerical Heat Transfer*, Wiley, New York, NY, second edition, 2006.

[51] Fox, R. O., *Computational Models for Turbulent Reacting Flows*, Cambridge University Press, Cambridge, UK, 2003.

[52] Heinz, S., On Fokker-Planck Equations for Turbulent Reacting Flows. Part 1. Probability Density Function for Reynolds-Averaged Navier-Stokes Equations, *Flow Turbul. Combust.*, **70**(1–4):115–152 (2003).

[53] Dongarra, J. and *et al.*, The International Exascale Software Project Roadmap, *Int. J. High Perform. C.*, **25**:3–60 (2011).

[54] Sagaut, P., *Large Eddy Simulation for Incompressible Flows*, Springer, New York, NY, third edition, 2010.

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

[56] Williams, F. A., Turbulent Combustion, in Buckmaster, J. D., editor, *The Mathematics of Combustion*, Frontiers in Applied Mathematics, SIAM, Philadelphia, PA, 1985.

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

[58] Grigoriu, M., *Applied Non-Gaussian Processes*, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[59] Kloeden, P. E., Platen, E., and Schurz, H., *Numerical Solution of Stochastic Differential Equations through Computer Experiments*, Springer, Berlin; New York, corr. 2nd print edition, 1997.

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

[61] Madnia, C., Jaberi, F. A., and Givi, P., Large Eddy Simulation of Heat and Mass Transport in Turbulent Flows, In Minkowycz *et al.* [50], Chapter 5, pp. 167–190.

[62] Gottlieb, D. and Turkel, E., Dissipative Two-Four Methods for Time-Dependent Problems, *Math. Comput.*, **30**(136):703–723 (1976).

[63] Yilmaz, S. L., Nik, M. B., Sheikhi, M. R. H., Strakey, P. A., and Givi, P., An Irregularly Portioned Lagrangian Monte Carlo Method for Turbulent Flow Simulation, *J. Sci. Comput.*, **47**(1):109–125 (2011).

[64] Devine, K. D., Boman, E. G., and Karypis, G., Partitioning and Load Balancing for Emerging Parallel Applications and Architectures, in Heroux, M., Raghavan, A., and Simon, H., editors, *Frontiers of Scientific Computing*, SIAM, Philadelphia, 2006.

[65] Boman, E., Devine, K., Fisk, L. A., Heaphy, R., Hendrickson, B., Vaughan, C., Catalyurek, U., Bozdag, D., and Mitchell, W., *Zoltan 3.0: Data Management Services for Parallel Applications; User's Guide*, Sandia National Laboratories, Albuquerque, NM, 2006, Tech. Report SAND2006-2958 `http://www.cs.sandia.gov/Zoltan/ug_html/ug.html`.

[66] Chen, Y.-C., Peters, N., Schneemann, G. A., Wruck, N., Renz, U., and Mansour, M. S., The Detailed Flame Structure of Highly Stretched Turbulent Premixed Methane-Air Flames, *Combust. Flame*, **107**(3):223–226 (1996).

[67] Top 500 Supercomputer Sites, `http://www.top500.org/`, 2012.

[68] The OpenMP API Specification for Parallel Programming, `http://openmp.org/wp/`, 2012.

[69] Nickolls, J., Buck, I., Garland, M., and Skadron, K., Scalable Parallel Programming with

CUDA, *Queue*, **6**(2):40–53 (2008).

[70] Linford, J. C., Michalakes, J., Vachharajani, M., and Sandu, A., Multi-core Acceleration of Chemical Kinetics for Simulation and Prediction, in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, SC '09, pp. 7:1–7:11, New York, NY, USA, 2009, ACM.

[71] Shi, Y., Green Jr., W. H., Wong, H.-W., and Oluwole, O. O., Redesigning Combustion Modeling Algorithms for the Graphics Processing Unit (GPU): Chemical Kinetic Rate Evaluation and Ordinary Differential Equation Integration, *Combust. Flame*, **158**(5):836–847 (2011).