Massively-parallelized reciprocal monte-carlo ray tracing for radiative transfer coupled with turbulent LES combustion simulations

Isaac Hunsaker$^1$  Jeremy Thornock$^2$  Todd Harman$^3$  Philip J. Smith$^4$

$^{1,2,3,4}$Institute for Clean & Secure Energy, University of Utah, 155 South 1452 East, Room 380 Salt Lake City, UT 84112

Radiation is the dominant mode of heat transfer in high temperature combustion environments. Radiative heat transfer affects the gas and particle phases, including all the associated combustion chemistry. The radiative properties are in turn affected by the turbulent flow field. This bi-directional coupling of radiation turbulence interactions poses a major challenge in creating parallel-capable, high-fidelity combustion simulations. In this work, a new model is developed in which a reciprocal monte-carlo radiation model is coupled with a turbulent large eddy simulation combustion model. An asynchronous, multi-level mesh is implemented. The combustion model runs in parallel on the fine level of a decomposed domain. The radiation model runs asynchronously in parallel on the coarse level of a recomposed domain. The recomposed domain is stored on each processor after information sharing of the decomposed domain is handled via the message-passing interface. A strong scaling analysis was performed on the Titan supercomputer cluster. The model demonstrates strong scaling to over 16,000 processing cores.

1 Introduction

Radiation is the dominant mode of heat transfer in high temperature combustion environments [1]. Radiative heat transfer in turbulent flames affects the gas and particle phases, including all the associated combustion chemistry. The turbulence-radiation interactions (TRI) have been shown to be of great importance in turbulent flames [2–6]. Modeling TRI is difficult due to the nonlinear coupling between temperature, species concentrations and radiative intensities [4, 7]. Further, coupling parallel simulations of combustion and radiation poses several numerical challenges. The fluid mechanics of combustion are local phenomena, making them amenable to domain decomposition. Conversely, radiation is a long-distance, and potentially all-to-all phenomenon, creating difficulties for domain decomposition. Further, accurate calculation of radiative transfer requires spatially resolved information regarding the temperature and species composition fields. Traditional modeling of turbulent systems has included Reynolds-averaged Navier Stokes (RANS) simulations. The RANS model provides, at a relatively low computational cost, spatially averaged values of the gas temperature and species fields. However, for highly non-linear physics such as radiation, spatial averaging in this manner may introduce large errors [8]. Alternatively, direct numerical simulation (DNS) fully resolves the power spectrum of eddies, giving access to the full spatial distribution of the pertinant fields. Wu et al. [9] and Deshmukh [10] have coupled a
monte-carlo ray tracing method to solve the radiative transfer equation in a turbulent reacting flow modeled by DNS. Unfortunately, due to its high computational demand, DNS remains impractical for use in large-scale combustion simulations. In contrast, large eddy simulations (LES) resolve the largest fluid motions, down to the Nyquist limit for a given turbulent field and mesh resolution. Beyond this limit, the less-important smaller eddies are approximated via simpler models. Because combustion turbulence is generally dominated by large eddies [11], LES gives a better description of the fluid mechanics than RANS, and does so without the computational cost of DNS.

The various levels of accuracy in which thermal radiation has been modeled in combustion simulations has been reviewed by Snegirev [12] and Sacadura [13]. The radiation models cited include the optically-thin approximation [14], the discrete ordinates method [15, 16] the discrete transfer method [17], and the finite volume method [18]. The optically thin model neglects the participation of media (absorption, emission, and scattering), and has been shown to introduce error even in small flames [19]. The remaining methods model radiative emission as energy emanating along a set of pre-defined directions. Such angular discretization suffers from the ray effect [20]. Conversely, monte-carlo techniques that select randomly-distributed rays at each time step have low sensitivity to angular discretization and are applicable regardless of media optical thickness [12]. In his earlier work, Snegirev presents a RANS model of buoyant turbulent diffusion flames coupled with statistical modeling of thermal radiation transfer. Although Snegirev’s earlier model used a robust formulation of thermal radiation via the monte-carlo method, his turbulence model suffered from the lack of resolution of the sharply varying fluctuations of temperature and species concentrations that are lost in RANS approximations. More recently, Snegirev coupled monte-carlo radiation with large eddy simulations [21], [22]. These simulations operated on modest meshes of approximately 498,000 control volumes. Other examples of coupled LES monte-carlo radiation models are rare, but include the work of Zhang et al., in which a larger mesh of 4.7 million cells were used [23]. In this emerging field remain several unresolved issues. One such issue is how to deal with increasing mesh sizes that are run on increasingly parallelized super computers.

Modern super computers are comprised of hundreds of thousands of computing cores, and are used to run simulations with meshes comprised of billions of computational cells [24–26]. Strong scaling in massively-parallel computing is difficult to obtain due to load imbalancing and inter-processor communication demands. The strong scalability limit of a code is reached when an increase in the number of parallel processors used on a fixed problem size does not result in a decrease in computational wall time [27]. Numerous examples of parallelized monte-carlo radiation models were investigated, most of which cease to scale beyond 100 processors [8, 23, 28–42]. An example of a coupled combustion and monte-carlo radiation model that has a scalability limit above 200 processors was not found in the literature.

In this work, a new numerical technique has been developed to perform large eddy simulations of large-scale combustion flows coupled with a three-dimensional reciprocal monte-carlo ray tracing radiation model. This model has been optimized for use on high-performance super computers and achieves nearly-ideal strong scaling to over 16,000 processors.

As mentioned above, fluid mechanics and most other phenomena in combustion physics are localized phenomena and are readily solved on domain-decomposed meshes. In this work, to represent the long-range effects of radiation, the computational domain is recomposed at the time of each radiation solve. This is accomplished over a message passage interface, through which
each processor shares the temperature and radiative-properties fields (absorption coefficient, wall emissivity, and cell type) with all other processors. This reconstructed domain combined with the mutually exclusive nature of reciprocal monte-carlo rays is amenable to massive parallelism. Radiative properties are calculated via the full-spectrum k-distribution (FSK) method. For efficiency, these calculations are pre-computed and tabulated in narrow increments of temperature and species mixture fraction values.

2 Methodology

2.1 Turbulent flow and combustion

The ARCHES component solves the conservative, finite volume, compressible, low-mach formulation of the Navier-Stokes equation with a pressure projection that includes the effect of variable density, reaction, and heat transfer modes in the gas phase including radiation. Given the wide range of length and time scales that are present in many combustion problems of interest, ARCHES utilizes models for bridging the molecular (micro) scales to the full, large (macro) scales. This bridging occurs through the use of subgrid and resolved scale mixing, reaction, and turbulence models. For example, momentum turbulence closure is accomplished by using various large eddy simulation (LES) type closure models. Fast chemistry scales are modeled by preprocessing the full chemical mechanism, modeled in various idealized configurations (e.g., equilibrium or flamelets), then tracking a set of reduced parameters on the LES mesh that map the full thermochemical state-space. In general, the chemistry is tabulated and stored in a reaction table. Subgrid turbulence species mixing processes and included by using presumed PDF methods and using models for certain moments of the distribution (e.g., scalar variance and mixture fraction). The turbulence-species subgrid interaction model description are termed mixing models. The chemistry and mixing models are usually completely preprocessed together into one tabular format to give a mixing table.

The following gives a brief introduction to a few of the key concepts of the ARCHES formulation of the transport equations and physical models as well as the CFD algorithm.

2.1.1 Governing Equations

The essential governing equations for the ARCHES component, written in finite volume form, include the mass balance, momentum balance, mixture fraction balance, and energy balance equations. Using a bold-face symbol to represent a vector quantity, the equations are:

1. The mass balance,

   \[
   \int_V \frac{\partial \rho}{\partial t} dV + \oint_S \rho \mathbf{u} \cdot d\mathbf{S} = 0 ,
   \]  

   where \( \rho \) is density and \( \mathbf{u} \) is the velocity vector.

2. The momentum balance,

   \[
   \int_V \frac{\partial \rho \mathbf{u}}{\partial t} dV + \oint_S \rho \mathbf{u} \cdot d\mathbf{S} = \oint_S \tau \cdot d\mathbf{S} - \int_V \nabla p dV + \int_V \rho g dV ,
   \]
where $\tau$ is the deviatoric stress tensor defined as $\tau_{ij} = 2\mu S_{ij} - \frac{2}{3}\mu \delta_{ij}$, the second isotropic term in $\tau_{ij}$ is absorbed into the pressure projection for the current low-Mach scheme, and $S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$. Also in Equation 2, $g$ is the gravitational body force and $p$ is pressure.

3. The mixture fraction balance,

$$\int_V \frac{\partial \rho f}{\partial t} dV + \oint_S \rho u f \cdot dS = \oint_S D \nabla f \cdot dS,$$

where $f$ is the mixture fraction and a Fick’s law form of the diffusion term assuming equal diffusivities results in a single diffusion coefficient, $D$.

4. The thermal energy balance,

$$\int_V \frac{\partial \rho h}{\partial t} dV + \oint_S \rho u h \cdot dS = \oint_S k \nabla h \cdot dS - \oint_S q \cdot dS,$$

where $h$ is the sum of the chemical plus sensible enthalpy, $q$ is the radiative flux, a Fourier’s law form of the conduction term is used with a diffusion coefficient, $k$, and the pressure term is neglected.

These equations are solved in an LES context, meaning filters are applied to the equations. Here, Favre filtering is used, defined as

$$\overline{\phi} = \frac{\rho \phi}{\overline{\rho}},$$

to isolate the density in the filtered equations. The filtering operations result in the classic turbulence closure problem and thus models are required.

Consider a control volume, $V$, with surface area $S$. Because the equations will be solved on a computational grid, it can be assumed that the the control volume has $N$ faces, where unique faces are identified with their index, $k$. The discussion is further simplified by only considering cubic volumes with length $h$. Given the cubic control volume, a surface-filtered field for a variable $\phi$ is defined as $\overline{\phi}^{(j)}(x)$, where the variable is filtered on a plane in the $x_j$ orthogonal direction. Then, for any surface, $k$, the field is sampled at the face centered location. For example, if $j = 1$, the surface-filtered quantity is

$$\overline{\phi}^{2d,(1)}(x) = \frac{1}{h^2} \int_{x_2-h/2}^{x_2+h/2} \int_{x_3-h/2}^{x_3+h/2} \phi(x') dx'_2 dx'_3 .$$

The volume average follows as

$$\overline{\phi}^{3d}(x) = \frac{1}{h^3} \int_{x_1-h/2}^{x_1+h/2} \int_{x_2-h/2}^{x_2+h/2} \int_{x_3-h/2}^{x_3+h/2} \phi(x') dx'_1 dx'_2 dx'_3 .$$

The bars over the variable, $\phi$, are labeled with ‘2d’ and ‘3d’ superscripts to distinguish between the two filters. Pope [43] identifies the proceeding definitions as using the “anisotropic box” filter kernel where the resultant variables are simply averages over the intervals $x_j - \frac{1}{2}h < x'_j < x_j + \frac{1}{2}h$. 


For convenience in isolating density in the filtered equations, a Favre-filtered quantity is defined for an arbitrary variable, \( \varphi \), as

\[
\overline{\varphi}^{2d} \equiv \frac{\rho \varphi}{\rho^{2d}}, \tag{7}
\]
and

\[
\overline{\varphi}^{3d} \equiv \frac{\rho \varphi}{\rho^{3d}}. \tag{8}
\]

Because the 2D and 3D filters are explicitly defined, this convention is slightly different than what is normally observed in the literature. Most literature, however, derives the filtered equations from the finite difference equations rather than the finite volume equations. Thus, using \( \rho^{2d} \) and \( \rho^{3d} \) in Equations 7 and 8 to stress surface and volume filtered densities are appropriate for the present discussion.

The previous definitions are applied to the integral forms of the governing equations to obtain the Favre-filtered LES equations. Nevertheless, there are terms in the Favre-filtered equations that cannot be solved. These include the surface filtered convection of momentum, \( \overline{u_i u_j^{2d}} \), the surface filtered convection of mixture fraction, \( \overline{u_j f^{2d}} \), and the surface filtered convection of enthalpy, \( \overline{u_j h^{2d}} \).

For the filtered velocity product, \( \rho^{2d} \overline{u_i u_j^{2d}} \), a subgrid stress tensor is defined as,

\[
\tau_{ij}^{s gs} = \overline{u_i u_j^{2d}} - \overline{u_i^{2d} u_j^{2d}}. \tag{9}
\]

Similarly, subgrid diffusion terms are defined for mixture fraction and enthalpy,

\[
J^f = \overline{u_j f^{2d}} - \overline{u_j^{2d} f^{2d}}, \tag{10}
\]
\[
J^h = \overline{u_j h^{2d}} - \overline{u_j^{2d} h^{2d}}. \tag{11}
\]

Using these definitions, the final form of the Favre-filtered equations is

1. The filtered mass balance,

\[
\frac{d}{dt} (\rho^{3d}) + \frac{S_k}{V} n_{kj} \left( \rho^{2d} \overline{u_j^{2d}} \right) = 0. \tag{13}
\]

2. The filtered momentum balance,

\[
\frac{d}{dt} (\rho^{3d} \overline{u_i^{3d}}) = \frac{S_k}{V} n_{kj} \left( -\rho^{2d} \overline{u_i^{2d} u_j^{2d}} + \tau_{ij}^{2d} + \tau_{ij}^{s gs} - \rho^{2d} \delta_{ij} \right) + \rho^{3d} g_i. \tag{14}
\]

3. The filtered mixture fraction balance,

\[
\frac{d}{dt} (\rho^{3d} \overline{f^{3d}}) = \frac{S_k}{V} n_{kj} \left( -\rho^{2d} \overline{u_j^{2d} f^{2d}} + D \nabla \overline{f^{2d}} + J^f \right). \tag{15}
\]
4. The filtered thermal energy balance,

\[
\frac{d}{dt} \left( \frac{p^3}{\rho} \tilde{h}^3 \right) = \frac{S_k}{V} n_{kj} \left( -p^2 \tilde{u}_j \tilde{h}^2 + k \nabla \tilde{h}^2 - \tilde{q}^2 + f^h \right). \tag{16}
\]

The subgrid momentum stress, \( \tau_{ij}^{sgs} \), the subgrid mixture fraction dissipation, \( f^f \), and the subgrid heat dissipation, \( f^h \), contain the unresolved or subgrid action of the turbulence on the transported quantities. Since these terms arise from definitions, models are introduced to include the subgrid effects that they represent. These models are discussed next.

### 2.1.2 Subgrid Turbulence Models

The construction of both \( f^f \) and \( f^h \) is relatively straightforward. Invoking an “eddy-viscosity” modeling concept, the subgrid transport due to turbulent advection is treated as an enhanced diffusion term for the unclosed terms listed above. That is, the subgrid mixture fraction dissipation and subgrid enthalpy dissipation are respectively written as,

\[
f^f = D_t \frac{\partial f^2}{\partial x_j}, \tag{17}
\]

and

\[
f^h = k_t \frac{\partial h^2}{\partial x_j}. \tag{18}
\]

To model \( D_t \) and \( k_t \), constant turbulent Schmidt \( (Sc_t) \),

\[
Sc_t = \frac{1}{\rho \mu_t} D_t, \tag{19}
\]

and Prandlt \( (Pr_t) \),

\[
Pr_t = \frac{1}{\rho \ k_t}, \tag{20}
\]

numbers are assumed with where \( \mu_t \) is a turbulent viscosity. Following Pitsch and Steiner [44], the values of the turbulent Schmidt and Prandlt number are taken as \( Sc_t = Pr_t = 0.4 \), which is consistent with a unity Lewis number assumption.

For the subgrid scale stress tensor, \( \tau_{ij}^{sgs} \), two common LES turbulence closure models are the constant coefficient Smagorinsky model [45] and the dynamic coefficient Smagorinsky model [46]. As with the scalar subgrid modeling terms, the eddy viscosity model is again invoked for \( \tau_{ij}^{sgs} \). Defining the deviatoric subgrid stress tensor as,

\[
\tau_{ij}^{d, sgs} = \tau_{ij}^{sgs} - \frac{1}{3} \tau_{kk}^{sgs} \delta_{ij}, \tag{21}
\]

the subgrid stress is taken as,

\[
\tau_{ij}^{d, sgs} \approx -2 \nu_t \nabla^2 \nabla = -2 (C_s \Delta)^2 |\nabla| \nabla_{ij}, \tag{22}
\]
where $\Delta$ is the filter width, $\nu_t$ is the eddy viscosity and $|S| \equiv (2S_{ij}S_{ij})^{1/2}$. For the Smagorinsky model, $C_s \approx 2$ depending on the filter type, numerical method, and flow configuration [43].

For the dynamic Smagorinsky model, $C_s$ is computed by taking a least squares approach to determine the length scale [47],

$$ (C_s \Delta)^2 = \frac{\langle L_{ij}M_{ij} \rangle}{\langle M_{ij} \rangle} , $$

where

$$ L_{ij} = 2(C_s \Delta)^2 |\mathbf{S}|S_{ij} - 2(C_s \Delta)^2 |\mathbf{S}|\mathbf{S}_{ij} , $$

and

$$ M_{ij} \equiv 2 \left( |\mathbf{S}|S_{ij} - \alpha^2 |\mathbf{S}|S_{ij} \right) . $$

The hat defines an explicit test filter and the angled brackets in Equation 33 conceptually represent an averaging over a homogeneous region of space that, experience has shown, is necessary for stability. Experience has also shown that averaging over the test filter width is adequate and the filter width ratio, $\alpha = \hat{\Delta}/\Delta$, is usually taken to be 2.

2.1.3 Subgrid Momentum Dissipation

Addressing the momentum closure involves finding a suitable model for the subgrid scale stress tensor, $\tau_{ij}^{sgs}$. Two common LES turbulence closure models are examined: the constant coefficient Smagorinsky model and the dynamic coefficient Smagorinsky model. In LES modeling, field variables are decomposed into a spatially filtered field and a residual component, $u = \bar{u} + u'$. This decomposition is known as a Leonard decomposition. While seemingly similar to a Reynolds decomposition used in Reynolds Averaged Navier-Stokes (RANS) models, the Leonard decomposition has the property that the filtered residual component is generally not equal to zero, $u' \neq 0$.

As a result, the subgrid stress term contains several terms,

$$ \tau_{ij}^{sgs} = \left( \bar{u}_i + u'_i \right) \left( \bar{u}_j + u'_j \right) - \bar{u}_i \bar{u}_j , $$

$$ = \left( \bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j \right) + \left( \bar{u}_i u'_j + u'_i \bar{u}_j + u'_i + u'_j \right) , $$

referred to as the Leonard stress, the cross stresses, and the Reynolds stress respectively.

It is useful to consider the physical interpretation of the various components of the stress. The Leonard term is responsible for filtering and projecting the nonlinear interactions of the resolved components back to the finite LES space. This is a correction to the resolved advective term in accordance with the stated explicit filter used to derive the LES equations. It does not account for aliasing errors. The first cross term represents advection of the resolved field by turbulent fluctuations. The second cross term represents the advection of subgrid scales by the resolved field. The Reynolds stress is familiar from RANS and represents the advection of subgrid scales by turbulent fluctuations.

As with the scalar subgrid modeling terms, the eddy viscosity model is again invoked for $\tau_{ij}^{sgs}$. The most common eddy viscosity model in LES is the Smagorinsky model [45]. Defining the
deviatoric subgrid stress tensor as,
\[
\tau_{ij}^{d, sgs} = \tau_{ij}^{sgs} - \frac{1}{3} \tau_{kk}^{sgs} \delta_{ij}, \quad (27)
\]
the subgrid stress is approximated by,
\[
\tau_{ij}^{d, sgs} \approx -2 \nu_t \bar{S}_{ij} = -2(\nu_t \Delta)^2 |\bar{S}| \bar{S}_{ij}, \quad (28)
\]
where, \(\Delta\) is the filter width, \(\nu_t\) is the eddy viscosity, \(|\bar{S}| \equiv (2\bar{S}_{ij} \bar{S}_{ij})^{1/2}\), and typically \(C_s \approx 2\) depending on the filter type, numerical method, and flow configuration [43]. This model is basically identical to Prandtl’s mixing length model with \(l = C_s \Delta\).

The dynamic procedure [48] eliminates the need to specify the model constant, \(C_s\), a priori, with the basic assumption that the constant is the same for two different filter scales. The smaller scale is historically referred to as the “grid scale” (though the filter width need not equal the grid spacing, \(\Delta \geq h\)), and the larger scale is referred to as the “test scale”. Implicit in this assumption is the requirement that both scales lie within the inertial subrange.

Defining the deviatoric residual stress tensor as,
\[
T_{ij}^d = T_{ij} - \frac{1}{3} T_{kk} \delta_{ij}, \quad (29)
\]
the residual stress at the test scale is given by,
\[
T_{ij}^d \equiv \hat{u}_i \hat{u}_j - \bar{u}_i \bar{u}_j \approx -2(\nu_t \Delta)^2 |\bar{\hat{S}}| \hat{S}_{ij}, \quad (30)
\]
where \(\hat{\Delta}\) is the test filter width and the hat defines an explicit test filter. By test filtering Equation 9 and combining this with 30, the Leonard term, \(L_{ij}\) can be calculated. This is also known as the “Germano identity”,
\[
L_{ij} = T_{ij} - \tau_{ij}^{sgs} = \hat{u}_i \hat{u}_j - \bar{u}_i \bar{u}_j. \quad (31)
\]
Notice that the Leonard term is directly computable from resolved LES quantities. Restating the Smagorinsky model in terms of the Germano identity results in an over-determined system of equations for the unknown, \(C_s\),
\[
L_{ij} = 2(C_s \Delta)^2 |\bar{\hat{S}}| \hat{S}_{ij} - 2(C_s \Delta)^2 |\bar{\hat{S}}| \hat{S}_{ij}. \quad (32)
\]
Although \(C_s\) has been removed from the test filtering operation of the subgrid stress, this approximation yields acceptable results. In practice, the least squares approach is taken to determine the length scale [47],
\[
(C_s \Delta)^2 = \frac{\langle L_{ij} M_{ij} \rangle}{\langle M_{ij} M_{ij} \rangle}, \quad (33)
\]
where
\[
M_{ij} \equiv 2 \left( |\bar{\hat{S}}| \hat{S}_{ij} - \alpha^2 |\bar{\hat{S}}| \hat{S}_{ij} \right). \quad (34)
\]
The only model parameter, then, is the filter width ratio, \(\alpha = \hat{\Delta}/\Delta\), usually taken to be 2.

The angled brackets in Equation 33 conceptually represent averaging over a homogeneous region of space which, experience has shown, is necessary for stability. With these implementation practices, the dynamic model is generally robust. The implementation can be made more efficient by computing the constant roughly every 10 time steps (based on the advective CFL), and only for the first Runge-Kutta step.
2.1.4 LES Algorithm

The set of filtered equations (Equations 13-16) are discretized in space and time and solved on a staggered, finite volume mesh. The staggering scheme consists of four offset grids. One grid stores the scalar quantities and the remaining three grids store each component of the velocity vector. The velocity components are situated so that the center of their control volume is located on the face centers of the scalar grid in their respective direction.

The staggering arrangement is advantageous for computing low-Mach LES reacting flows. First, since a pressure projection algorithm is used, the velocities are exactly projected without interpolation error because the location of the pressure gradient coincides directly with the location of the velocity storage location. Second, Morinishi et al. [49] showed that kinetic energy is exactly conserved when using a central differencing scheme on the convection and diffusion terms without a subgrid model and in combination with a staggered grid. Having a spatial scheme that conserves kinetic energy is advantageous because it limits artificial dissipation that arises from the differencing scheme. These conservation properties make the staggered grid a prime choice for LES reacting flow simulation.

For the spatial discretization of the LES scalar equations, flux limiting and upwind schemes for the convection operator are used. These schemes are advantageous for ensuring that scalar values remain bounded. For the momentum equation, a central differencing scheme for the convection operator is used. All diffusion terms are computed with a second order approximation of the gradient.

When computing the 2D surface filtered field on the faces of the control volume, an interpolation from the 3D volume filtered field becomes necessary. This approximation is tolerated because computing the 2D surface field is simply not possible with the given grid scheme.

An explicit time stepping scheme is chosen. A general, multistep explicit update for a variable, \( \phi \), may be written as,

\[
\phi^{0} = \phi^{n},
\]

\[
\phi^{i} = V \sum_{k=0}^{m-1} \left( \alpha_{i,k} \phi^{(k)} + \Delta t \beta_{i,k} L(\phi^{(k)}) \right), \quad i = 1, ..., m
\]

\[
\phi^{(m)} = \phi^{n+1},
\]

where \( n \) is the time level, \( m \) is the substep between \( n \) and \( n+1 \), \( \alpha \) and \( \beta \) are integration coefficients, and \( L \) is a linearization operator on the the convective flux and source terms. Letting \( m = 1 \) and \( \alpha = \beta = 1 \) the forward-Euler time integration scheme is determined,

\[
(\phi)^{n+1} = (\phi)^{n} + \Delta t(L(\phi)^{n}).
\]

A higher order, multistep method is derived by letting \( m > 1 \) and choosing appropriate constants for \( \alpha \) and \( \beta \). For this study, two step and three step, strong stability preserving (SSP) coefficients were chosen from Gottlieb et al. [50].

Using the coefficients given by Gottlieb et al., the SSP-RK 2 stepping scheme is

\[
(\phi)^{(1)} = (\phi)^{n} + \Delta t(L(\phi)^{n})
\]
\((\phi)^{n+1} = \frac{1}{2}(\phi)^n + \frac{1}{2}(\phi)^1 + \frac{1}{2}\Delta t(L(\phi)^{(1)})\).

SSP-RK 3 time stepping scheme is,
\[
(\phi)^{(1)} = (\phi)^n + \Delta t(L(\phi)^n)
\]
\[
(\phi)^{(2)} = \frac{3}{4}(\phi)^n + \frac{1}{4}(\phi)^{(1)} + \frac{1}{4}\Delta t(L(\phi)^{(1)})
\]
\[
(\phi)^{(n+1)} = \frac{1}{3}(\phi)^n + \frac{2}{3}(\phi)^{(2)} + \frac{1}{4}\Delta t(L(\phi)^{(2)})
\]

The time step is limited by
\[
\Delta t \leq c\Delta t_{F,E}
\]
where \(\Delta t_{F,E}\) is the forward-Euler time step limited by the Courant-Friedrichs-Levy condition and \(c\) is a constant less than or equal to one.

A higher order, multistep method is derived by letting \(m > 1\) and choosing appropriate constants for \(\alpha\) and \(\beta\). For this study, two step and three step, strong stability preserving (SSP) coefficients were chosen from Gottlieb et al. [50]. The coefficients for SSP-RK 2 and SSP-RK 3 are optimal in the sense that the scheme is stable when \(c = 1\) if the forward-Euler time step is stable for hyperbolic problems. In practice, for the Navier-Stokes equations, the value of \(c\) is taken less than one.

Choosing an explicit time stepping scheme, rather than an implicit one, creates a challenge for solving the set of equations. The density at the \(n + 1\) timestep, which is required to determine the cardinal variables, requires an estimation. Taking the estimated density for \(\rho^{n+1}\) to be \(\rho^*\), the estimation can be as simple as \(\rho^* = \rho^n\). Note that the 2D and 3D filter distinction is dropped for the remainder of this discussion for the sake of simplicity. A slightly more complicated procedure involves a forward-Euler step of the continuity equation to obtain \(\rho^*\). This is written as,
\[
\rho^* = \rho^n - \Delta t \frac{S_k}{V} n_{kj} (\rho u_j^n)
\]

In an ideal situation, \(\rho^{n+1}\) would be a known quantity rather than an approximation. \(\rho\) is a function of the same variables that are being updated in time, namely, the mixture fraction, \(f\), and enthalpy, \(h\). This quandary is a result of the explicit time stepping method and will not be resolved for variable density flows without using a fully implicit method. Explicit methods, however, do have advantages, especially for large scale parallel computations. Specifically, explicit methods are easier to load balance because the amount of work required for each processor is readily determined a priori, which makes for an efficient parallel computation. Explicit methods are also easier to code into a computer and to debug. For these reasons, the current algorithm discussion is limited to explicit methods only.

2.1.5 Direct Quadrature Method of Moments

The direct quadrature method of moments (DQMOM) is a recently-developed moment method for tracking distributions. It has been applied to distributions of evaporating droplets, soot particle distributions, fluidized beds, and subgrid chemistry PDFs. The method is similar to the
quadrature method of moments (QMOM), in that it uses quadrature to provide closure for the moment transport equation; however, it differs in that the moment transport equations are not actually solved, unlike the quadrature method of moments (QMOM). Rather, the DQ MOM tracks the quadrature weights and weighted abscissas representing the NDF directly, rather than using the product-difference algorithm to transform between a set of moments and quadrature weights and abscissas that would best represent that distribution.

The basic outline of the method, given below, defines and covers these fundamental steps and concepts:

1. Number Density Function (NDF)
2. Moments
3. Moment Methods for NDF Transport
4. Quadrature
5. Direct Quadrature Method of Moments

The number density function is the starting point, as it is the function of interest that is being tracked. Moments are defined, and moment methods are explained and applied to the NDF transport equation to yield the moment transport equation. Furthermore, quadrature is defined and applied to approximate the moments, which leads to a quadrature-approximated moment transport equation. This equation leads to the fundamental equations governing DQ MOM.

2.1.6 Number Density Function

Using the direct quadrature method of moments (DQ MOM) in ARCHES, the dispersed phase is represented as a number density function (NDF), which is tracked in a stationary Eulerian reference frame. The NDF is denoted as \( f \) and represents the number of particles at a particular point in space and time. Using DQ MOM, this NDF is parameterized on several different variables - independent variables for the particles. These are called “internal coordinates,” and are denoted by \( \mathbf{x} \) (where boldface denotes a vector quantity). In this case the NDF is written as \( f (\xi; \mathbf{x}, t) \).

The starting point for the DQ MOM equations is the NDF transport equation; this is derived in several places and will not be derived here. The NDF transport equation is:

\[
\frac{\partial f (\xi; \mathbf{x}, t)}{\partial t} + \frac{\partial}{\partial x_i} \left( \langle u_i | \xi; \mathbf{x}, t \rangle f (\xi; \mathbf{x}, t) \right) + \frac{\partial}{\partial \xi_j} \left( \langle G_j | \xi; \mathbf{x}, t \rangle f (\xi; \mathbf{x}, t) \right) = h (\xi; \mathbf{x}, t),
\]

where \( G_j \) is the velocity of the NDF in phase-space (that is, internal coordinate-space), and is defined by:

\[
G_j = \frac{d \xi_j}{dt}
\]

(note this is analogous to spatial velocity \( v \), defined by:

\[
v_i = \frac{dx_i}{dt},
\]

\( \xi_j \) and \( v_i \) are the internal coordinate and spatial velocity, respectively, and \( \langle \cdot | \cdot \rangle \) denotes the expected value of \( \cdot \) conditional on \( \cdot \).
and that \( G_j \) takes the same form as Lagrangian particle models); also, \( h \) is a birth and death term representing the appearance or disappearance of particles within the domain. Note also that the velocities \( v_i \) and \( G_j \) are conditioned on the value of the internal coordinates, as well as on space and time. This implies that the spatial and phase-space velocities are full distributions in \((\xi, x)\) space, just as the NDF is.

The number of internal coordinates is denoted by \( N_\xi \). If \( N_\xi = 1 \), the NDF is called “univariate”; if \( N_\xi > 1 \), then the NDF is called “multivariate”.

Verification and validation of the ARCHES component have been performed previously [51, 52]

2.2 Radiation properties

Radiation is assumed to be controlled by motions on the resolved scale, and no subgrid radiation model is taken into account. A similar assumption is made by Goncalves dos Santos, et al. [8]. A full spectrum k-distribution (FSK) property model has been implemented into the ARCHES component [53]. Radiation properties oscillate sharply as a function of wavelength. The FSK model reorders the property values into a smoothly-varying g-space, where the cumulative k-distribution, \( g \), is a nondimensional, Planck-function-weighted, reordered wavenumber. The general derivation of the FSK method is given by Modest [54], and the specific implementation in the ARCHES component is explained in [53].

2.3 Ray tracing code

The robustness of using monte-carlo ray tracing to predict radiative fluxes has previously been established by Snegirev and Modest among others [12, 54]. In a massively parallelized framework, where the computational domain is heavily decomposed, traditional forward monte-carlo methods (FM) suffer due to the large number of traced rays that never reach the subdomain of interest handled by a particular processor. Therefore, an emission-based reciprocity method (ERM) similar to that developed by Tesse et al. [55, 56] has been implemented. Optical paths (i.e. rays) propagate away from cells whose radiative-source terms are currently being solved, and the emission from the cells along the paths are attenuated in a reciprocal manner back to the target cells. In this manner, rays are generated only from cells where results are expected [57].

The governing equation for reciprocal monte-carlo ray tracing in nonhomogeneous, participating media was developed by Walters and Buckius [58]. Specifically,

\[
I_{i,k} = \int_0^{l_k} I_{0,cv} \kappa(l') e^{\exp[-\int_{l'}^{l_k} \kappa(l) dl'] dl'} + I_{o,sur}(T_\text{w}) e^{\exp[-\int_{l_w}^{l_k} \kappa(l') dl']},
\]

where \( I_{i,k} \) represents the incident intensity at location \( k \), \( \kappa \) represents the absorption coefficient, and \( l' \) represents the locations of the segment lengths along a ray.

In a discretized domain, piecewise homogeneity is assumed and Eqn. (44) is posed in the following form,
where $M$ represents the total number of discretized segment lengths of the ray. Thus, at a given location a distance $l'$ away from the starting point of the ray, $\int_{l_k}^{l_k} \kappa(l')dl$ represents the optical thickness of the path from $l_k$ to $l'$ and $\int_{l_{m-1}}^{l_k} \kappa(l')dl$ is the optical thickness of the path from $l_k$ to the previous $l'$. The intensities from each of the rays of a cell can then be weighted according to the solid angle that each ray subtends [59]. Assuming uniform distribution of the rays, each ray will subtend $\frac{\Omega}{N}$ steradians, where $\Omega$ is $4\pi$ Sr. for flow cells, $2\pi$ for boundary cells, and $N$ is the number of rays per cell used in the simulation. The radiative flux can then be calculated from the intensities of the rays, weighted by the discretized solid angle,

$$q_i = \frac{\Omega}{N} \sum_{r=1}^{N} I_i(\theta(\theta)),$$

(46)

where $I_i(\theta)$ and $\theta(\theta)$ represent for a particular ray, the incoming intensity and angle from the cell boundary normal, respectively. The radiative flux divergence of flow cells can be calculated as

$$\nabla \cdot q = \kappa(4\pi I_b - \int_{4\pi} I_{in} d\Omega),$$

(47)

where $\int_{4\pi} I_{in} d\Omega$ is represented by

$$\sum_{r=1}^{N} \frac{4\pi}{N}.$$

The target locations of the rays are distributed randomly throughout the cell. In this model, the Mersenne Twister random number generator is used to select the target locations and ray orientations [60]. In Cartesian meshes, randomly distributed ray location generation is trivial, and is accomplished by scaling three random numbers with the length, width, and height of the cell, respectively. Randomly-distributed ray orientation requires more treatment. Rays propagating from boundary surfaces are distributed over a hemisphere as follows.

$$\phi = 2\pi R_1$$

$$\theta = \arccos(R_2)$$

$$\hat{x} = \sin(\theta) \cos(\phi)$$

$$\hat{y} = \sin(\theta) \sin(\phi)$$

$$\hat{z} = \cos(\theta).$$
$R_1$ and $R_2$ are random numbers that vary between zero and one, $\phi$ and $\theta$ are the azimuthal and polar angles, respectively, and $\hat{x}$, $\hat{y}$, and $\hat{z}$ are the resulting components of the direction vector in Cartesian coordinates. The above formulation will generate rays that are randomly distributed over a hemisphere with a normal vector in the positive $z$ direction. The ray marching model adjusts the ray directions into the proper orientation based on the surface normal of the boundary cell at hand. This is accomplished by changing the order and sign of the three direction components.

For flow cells, no re-orientation of the direction vector is necessary, as the rays are randomly distributed over the full $4\pi$ Sr. Direction assignment is as follows,

$$\hat{z} = 2R_1 - 1$$

$$r = \sqrt{1 - z^2}$$

$$\phi = 2\pi R_2$$

$$\hat{x} = r \cos(\phi)$$

$$\hat{y} = r \sin(\phi)$$

Ray marching proceeds in a manner similar to that described by Amanatides and Woo [61]. The location and orientation of a ray are used to calculate the distances to each of the 3 potential exit faces of the cell in which the ray currently resides. The shortest of these three distances is used in determining through which face the ray will pass. This information is then used to calculate the next cell in which the ray will reside. Reflections are allowed to occur on non-black boundary faces. The temperature and emissivity of the boundaries are referenced, and the intensity at the ray-boundary intersection is computed and attenuated to the target location. For non-black surfaces, the ray reflects off the surface, and the subsequently referenced intensities are attenuated both by the total optical thickness and by the absorption of the boundary. Ray marching continues until the optical thickness of a ray exceeds a predetermined threshold value. In general, the threshold is met when $fe^{-\tau} < 0.1$, where $\tau$ is the current optical thickness, and $f$ is unity multiplied by one minus the absorptivity of each intersected boundary, $(1 - \alpha_b)$. In other words when less than 1% of the intensity from a location in the domain will reach the target cell, ray tracing of the current ray ceases. The mesh reconstruction technique described in the following section allows ray generation and propagation to occur on a each processor independently, negating the passing of rays, and minimizing inter-processor communication.

2.4 Data Exchange and Parallelism

The radiative heat transfer was computed once for every 20 LES timesteps. Results in [8] support the validity of this asynchronous approach. At each radiation solve, the decomposed domain used
for parallelism of the combustion model is recomposed and the radiation-specific field variables from each processor are shared with all other processors. Information sharing is accomplished through a message-passing interface.

The radiation model was translated from its original language of C++ into the GPU-specific language, CUDA. This allowed the model to be run on the GPU processors of the super-computing cluster, Titan.

3 Results and Discussion

Verification of this reciprocal monte carlo approach was performed previously [62]. Convergence rates were of half-order, as expected for a statistical model. Validation of the radiation model was performed in [63], where good agreement between the model and radiative flux data from a propellant fire was attained.

A strong scaling analysis of the CUDA implementation of the radiation model was performed on the super-computing cluster, Titan. Strong scaling was attained to 16,384 processing cores. Current input parameters limited the test to this number of cores. It is expected that strong scaling will continue much further, as at 16,384 cores, there was virtually no deviation from ideal scaling Fig. (1).

4 Conclusion

Radiation plays a critical role in combustion modeling due to the bi-directional coupling with the flow field through the energy equation. Radiative heat transfer is highly non-linear and requires spatially resolved information to be accurately represented. A reciprocity monte-carlo radiation
model has successfully been coupled with a large eddy simulation (LES) combustion model. The LES combustion model resolves the large fluid motions of the flow field, and models the less-important smaller eddies. The radiation model has been optimized for performance on massively-parallel GPU architectures. Strong scaling of the radiation model was accomplished to over 16,000 processors. This scaling value is approximately two orders of magnitude greater than that achieved by prior monte-carlo radiation models found in the literature. Continued studies are planned, and it is anticipated that the radiation model will continue to scale much further, as little deviation from ideal scaling was noted at 16,384 processing cores.

Acknowledgments

Thanks go to Steven Parker for his insight in ray tracing, Alan Humphry for his work on the CUDA translation, Allen Ricks for much input on the requirements of the model, and Stefan Domino for his insight into algorithm efficiency. We would also like to thank Patrick Notz and James Sutherland for being tremendous resources for C++. The Center for High Performance Computing (CHPC) at the University of Utah has provided much of the expertise necessary to achieve the parallel performance of this work.

This material is based upon work supported by the Department of Energy under Award Number DE-NT0005015. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References


