Efficient Radiative Transfer Equation Solver Using the Discontinuous Galerkin Method and the GPU

Xing He¹, Rohit Jain², and Ramakanth Munipalli³
HyPerComp Inc
2629 Townsgate Road; Suite 105
Westlake Village, CA 91361

Laurent Pilon⁴ and Ann Karagozian⁵
Department of Mechanical and Aerospace Engineering
University of California, Los Angeles
Los Angeles, CA 90095

The radiative transfer equation is an integro-differential equation whose efficient numerical solution requires significant computational resources. In the present study, we seek to unite two powerful techniques in contemporary scientific computing: the discontinuous Galerkin (DG) methods and computation on graphics processing units (GPU) for solving the radiative transfer equation. Significant improvements on numerical accuracy and speedup have been achieved in the present study.

I. Introduction

THERMAL radiation is a dominant mode of heat transfer in many combustion systems (rocket engines, scramjets, industrial furnaces) [Modest, 2003; Viskanta and Menguc, 1987; Viskanta, 2007] and must be properly accounted for in preliminary- and detailed-design phases of the systems development stages to result in a robust and fail-safe design. Besides heating, accurate modeling of the thermal radiation emitted by the exhaust plume of a rocket or missile is of utmost importance for military applications for the design of low-observable vehicles and remote-sensing [Alexeenko et al., 2002].

Empirical correlations used in conjunction with experimental data have been used in the past. However, increasing complexity and cost has prompted the development of modeling and numerical simulation tools to better understand and design modern combustion systems. Numerical simulation of combustion systems is computationally complex since it involves multi-physics interactions, such as two-phase flow, turbulent mixing, fuel vaporization and atomization, radiative and convective heat transfer, and chemical kinetics. Furthermore, coupling radiation transfer to fluid flow in such systems entails the computational burden of addressing multi-scale physical phenomena.

Currently the discontinuous Galerkin (DG) [Hesthaven and Warburton, 2007] method is being widely adopted for the improvement of modeling efficiency at high order of accuracy. In addition, graphics processing units (GPUs) are being increasingly employed for accelerating computations. In the present study, we seek to unite these two powerful techniques for rapidly, accurately, and efficiently solving the radiative transfer equation.

In the next section of this article, an overview of radiative transfer equation (RTE) is given; also numerical implementation details of RTE with DG method and general purpose computation on GPU are discussed. Validations of our RTE solver and the performance improvement with DG and GPU are presented in Sec. III. Finally concluding remarks are presented in Sec IV.

¹ Member Technical Staff, AIAA Member; corresponding author (hex@hypercomp.net)
² Member Technical Staff, AIAA Senior Member
³ Member Technical Staff, AIAA Senior Member
⁴ Associate Professor, ASME Member
⁵ Professor, AIAA Fellow

American Institute of Aeronautics and Astronautics
II. Methodology

A. Radiative Transfer Equation

The Radiative Transfer Equation (RTE) in an absorbing, emitting, and scattering medium takes the following form [Modest, 2003]:

\[
\tilde{s} \cdot \nabla I_s (\vec{r}, \tilde{s}) = -\kappa_s I_s (\vec{r}, \tilde{s}) - \sigma_s I_s (\vec{r}, \tilde{s}) + \kappa_a I_a (\vec{r}) + \frac{\sigma_{sr}}{4\pi} \int I_s (\vec{r}, \tilde{s}') \Phi_a (\tilde{s}' \rightarrow \tilde{s}) d\Omega \tag{1}
\]

where \(I_s (\vec{r}, \tilde{s})\) is the radiant intensity at spatial location, \(\vec{r}\), and wavelength, \(\lambda\), in direction, \(\tilde{s}\), \(\kappa_s\) is the absorption coefficient and \(\sigma_s\) is the scattering coefficient of medium, both of which greatly depend on the local species concentrations and vary dramatically with wavelength, \(I_a\) is Planck’s blackbody spectral intensity, and \(\Phi_a (\tilde{s}' \rightarrow \tilde{s})\) is the scattering phase function. The radiation typically has no contribution to the conservation laws of mass and momentum of fluid flows, and only appears in the energy equation as a radiative heat source/sink term (gradient of radiative heat flux), \(\vec{\nabla} \cdot \vec{q}_R\):

\[
\frac{\partial \rho c_e}{\partial t} + \vec{\nabla} \cdot (\rho \vec{u} c_e) = -\vec{\nabla} \cdot \vec{p} u + \vec{\nabla} \cdot (k \nabla T) - \vec{\nabla} \cdot \vec{q}_R + S \tag{2}
\]

with

\[
\vec{\nabla} \cdot \vec{q}_R = \int_0^{4\pi} \kappa_{sr} (I_{sr} - I_s) d\Omega \ d\lambda \tag{3}
\]

where \(\rho\) is the fluid density, \(c_e\) is the total energy per unit mass, \(\vec{u}\) is the fluid velocity, \(p\) is the static pressure, \(k\) is the fluid conductivity coefficient, \(T\) is the temperature, and \(S\) is the energy dissipation or contribution due to other mechanisms, e.g., chemical reactions. It is clear that the radiative intensity is directional (\(\tilde{s}\)), spectral (\(\lambda\)), as well as spatial (\(\vec{r}\)) dependent. The RTE is an integro-differential equation with the radiative properties of the participating medium often exhibiting strong and irregular dependence on wavelength. These characteristics are the main origin of complexity and difficulty in practically solving the RTE, either analytically or numerically.

B. Angular Discretization and Discrete Ordinate Method

The directional (angular) dependency of radiative intensity is a special aspect of the RTE. A wide variety of approaches have been proposed to account for this directional nature, including Monte Carlo method, ray tracing, zonal method, spherical harmonic method, discrete transfer method, discrete ordinate method, and finite volume method, etc. [Modest, 2003]. Each method of solution has both advantages and disadvantages. Within the framework of computational fluid dynamics, discrete ordinate method and finite volume method seem to be the most natural choices to solve the RTE with regard to meshing and coupling issues [Coelho et al., 1998]. Comparison studies [Coelho et al., 1998] on the performance of different methods have been carried out on solving the radiative heat transfer in an enclosure with obstacles using the discrete transfer method, finite volume method, and discrete ordinate method, in which the finite volume and discrete ordinate method are found to be more efficient. We therefore choose the discrete ordinate method and finite volume method for angular discretization of the RTE. These methods are simple, numerically efficient and extend easily to complex geometries.

Discrete ordinate method discretizes the solid angle with a set of ordinate directions. The integration over solid angle that appears in the RTE is evaluated by means of a weighted summation over the ordinate directions, where the specified weights are determined through algebraic and geometrical relations [Modest, 2003]. Finite volume method is very similar to the discrete ordinate method by discretizing the sphere of directions into a number of piece-wise non-overlapping angular elements, within which the radiation intensity is assumed to be constant. In both methods the RTE is solved direction by direction. These discrete components interact through the scattering term. The advantages of the discrete ordinate method and finite volume method are that conventional computational fluid dynamics (CFD) spatial grid could be readily adopted, and a generally converged solution could be satisfactorily obtained by simply increasing the \(N_s\) order (for discrete ordinate method) or angular discretization resolution (for finite volume method).

C. Spatial Discretization and Discontinuous Galerkin Method

RTE’s spatial discretization is typically carried out with finite volume method in which the cell-wise averaged radiative variables are saved [Modest, 2003]. The radiative flux through cell surface is usually evaluated with the so-called “step” scheme using the centroid values of its upwind cell and is actually a first-order flux construction. Step scheme introduces relatively large numerical dissipation also called “false scattering” [Chai et al., 1993]. During the
past decades 2nd order approximation of the radiation flux through the cell faces have also been proposed, including diamond scheme, exponential scheme and modified exponential scheme [Chai et al., 1994]. However, these schemes may also introduce physically unrealistic radiative intensity, and some intensity fix-up procedure is required.

Reed and Hill [Reed and Hill, 1973] first introduced the discontinuous Galerkin (DG) method to solve the neutron transport equation. Substantial improvement on the stability of solution was achieved. DG scheme could be easily incorporated with conventional radiation solution techniques, such as discrete ordinate method and finite volume method, and is very flexible for p-type local adaptation [Hesthaven and Warburton, 2007], which would be extremely useful for multi-scale radiation problems. The conventional step scheme is actually a p-0 DG scheme. The application of discontinuous Galerkin method for solving the RTE has been attracting more and more attention [see, e.g., Li, 2006].

For the sake of clarity, here we present implementation of the DG scheme to the RTE for a two-dimensional, absorbing and emitting but non-scattering medium combined with discrete ordinate method for angular discretization. In considering a particular angular direction, m, the RTE is written as:

$$\left(\hat{\mathbf{s}}_m \cdot \nabla \right) I_m = -\kappa I_m + \kappa I_{b,m}$$

(4)

Beginning with a two-dimensional p-1 discontinuous-Galerkin scheme, we use the monomial test functions; \(\{v_i\}_{i=1,2,3}\} = \{1, x - x_c, y - y_c\}, \) where \((x_c, y_c)\) are the Cartesian coordinates of the element’s centroid. The radiative intensity within the element considered could be prescribed as:

$$I_m = \sum_{i=1}^{3} a_i v_i = a_1 + a_2 (x - x_c) + a_3 (y - y_c)$$

(5)

where \(\{a_1, a_2, a_3\}\) are the unknown coefficient associated to the basis functions. We now define a weak form of Equation (4) by multiplying test function by the test function, \(v_j\), integrating over the element volume, \(V\), and applying the Gauss theorem:

$$\int_A I_m v_j \hat{s}_m \cdot dA - \int_V I_m \left( \hat{s}_m \cdot \nabla v_j \right) dV = -\int_V \kappa v_j I_m dV + \int_V \kappa v_j I_{b,m} dV$$

(6)

where \(\int_A\) is the cell surface integration and \(\int_V\) is the cell volume integration of the element under consideration. In the surface integration, the radiative flux along the surface, \(I_m^s\), has to be appropriately chosen. Substituting Eq. (5) into Eq. (6), yields:

$$\int_A \sum_{i=1}^{3} a_i v_i^* v_j v_i \hat{s}_m \cdot dA - \int_V \sum_{i=1}^{3} a_i v_i \left( \hat{s}_m \cdot \nabla v_j \right) dV = -\int_V \kappa \sum_{i=1}^{3} a_i v_i dV + \int_V \kappa v_j I_{b,m} dV$$

(7)

Where \(a_i^*\) and \(v_i^*\) are variables associated with the numerical flux along the cell surface. We therefore obtain 3 equations for each element by expanding the above equations for each \(v_j\) as follows:

For \(j = 1, v_j = v_1 = 1:\)

$$\int_A \sum_{i=1}^{3} a_i^* v_i^* v_1 v_i \hat{s}_m \cdot dA - \int_V \sum_{i=1}^{3} a_i v_i dV + \int_V \kappa I_{b,m} dV$$

(8)

or:

$$\int_A \left( a_1^* + a_2^* (x - x_c) + a_3^* (y - y_c) \right) \hat{s}_m \cdot dA = -\int_V \kappa (a_1 + a_2 (x - x_c) + a_3 (y - y_c)) dV + \int_V \kappa I_{b,m} dV$$

(9)

For \(j = 2, v_j = v_2 = (x - x_c):\)

$$\int_A \sum_{i=1}^{3} a_i^* v_i^* (x - x_c) \hat{s}_m \cdot dA - \int_V \sum_{i=1}^{3} a_i v_i \left( \hat{s}_m \cdot \nabla (x - x_c) \right) dV = -\int_V \kappa (x - x_c) \sum_{i=1}^{3} a_i v_i dV + \int_V \kappa (x - x_c) I_{b,m} dV$$

(10)

or:

American Institute of Aeronautics and Astronautics
\begin{align}
\int_A \left( a_i^* + a_z(x-x_e^*) + a_y^*(y-y_e^*)(x-x_c) \hat{s}_m \cdot dA - \int_V \left( a_1 + a_2(x-x_e) + a_3(y-y_e) \right) \hat{s}_m \cdot (1,0) \right) dV \\
= -\int_V \kappa(x-x_c) \left( a_1 + a_2(x-x_e) + a_3(y-y_e) \right) dV + \int_V \kappa(x-x_c) I_{b,m} dV 
\end{align} (11)

for \( j = 3 \), \( v_i = v_3 = (y-y_e) \):
\begin{align}
\int_A \sum_{i=1}^3 a_i^* v_i (y-y_e) \hat{s}_m \cdot dA - \int_V \sum_{i=1}^3 a_i v_i (\hat{s}_m \cdot \nabla y) dV = -\int_V \kappa(y-y_e) \sum_{i=1}^3 a_i v_i dV + \int_V \kappa(y-y_e) I_{b,m} dV
\end{align} (12)

or:
\begin{align}
\int_A \left( a_i^* + a_z(x-x_e^*) + a_y^*(y-y_e^*)(y-y_c) \hat{s}_m \cdot dA - \int_V \left( a_1 + a_2(x-x_e) + a_3(y-y_e) \right) \hat{s}_m \cdot (0,1) \right) dV \\
= -\int_V \kappa(y-y_c) \left( a_1 + a_2(x-x_e) + a_3(y-y_e) \right) dV + \int_V \kappa(y-y_c) I_{b,m} dV
\end{align} (13)

Missing pieces in Equations (9), (11), and (13) are the evaluation of the radiative flux through the element boundary. Fig. 1 shows the illustration of a grid setup in formulating the grid element, \( P \), in considering the radiation in direction \( \hat{s}_m \). Upwind scheme is a natural choice for the surface flux, e.g., considering the flux along face \( s \), the intensity is evaluated from element, \( S \), while surface \( n \)’s flux is evaluated through the information of element \( P \) itself. Similar procedures are carried out for the surfaces \( w \) and \( e \). For each cell we therefore have three equations [Eqs. (9), (11), and (13)], which cannot be solved directly since they are coupled with neighboring cell’s unknown as depicted in the numerical flux construction procedure.

**D. Computations on Graphics Processing Unit**

GPU is a highly parallel, multithreaded, and many-core processor (100s of cores) of enormous computing power. Fig. 2 shows the comparison of floating point operation capability per second of NVIDIA GPUs vs. Intel CPUs over the past 5 years [CUDA 2009]. It also demonstrates that GPU grew at a much more aggressive pace of computational power than CPU does.

Its characteristics of low cost, high throughput floating-point operation and high bandwidth memory access are...
attracting more and more researchers in the field of high performance computing (HPC) to employ the GPUs as acceleration devices. Meanwhile, the binding languages/interfaces for the GPU devices have also evolved from fully graphics oriented to general purpose language thus facilitating access to programmers with little or no graphics background. For HPC applications, GPU cards work as co-processors to CPUs. The CPU is responsible for I/O, submission and overseeing the computation tasks executed on GPU. In November 2006, NVIDIA introduced Compute Unified Device Architecture (CUDA) --- a general-purpose parallel computing architecture. CUDA is a software environment that allows developers to use C as high-level programming language with a small set of extensions for codes executed on GPUs [CUDA, 2009]. CUDA is also compatible with the GPU open industry standard --- OpenCL (http://en.wikipedia.org/wiki/OpenCL). In addition, the CUDA model has been mapped onto multicore CPUs with good success [Stratton et al., 2008]. In the present study we also employed CUDA for GPU computation.

In the CFD community, different groups have implemented structured grid Euler solvers for compressible flows on GPUs [Brandvik and Pullan 2008, Elsen et al. 2008]. Typically one order of magnitude speedup by using a single GPU card was achieved in comparing with a single-node CPU implementation. Philips et al. [Philips et al. 2009] implemented a parallel two-dimensional structured grid Euler solver and achieved a speedup of 160 by using a GPU cluster consisting of 8 GPU cards compared with a single CPU implementation. It is worth to mention a recent work by Klockner et al. [Klockner et al. 2009], in which they implemented a Maxwell’s equation solver with GPU using the discontinuous-Galerkin scheme and general unstructured three-dimensional grids, and a 40-time speedup was achieved.

The elemental process within a GPU computation is a so-called thread, a large number of threads are grouped into a block, and all blocks are organized as a grid. Therefore, a GPU program has only one grid, which consists of multiple blocks, and each block has a large number of threads (several hundreds). Each thread distinguishes itself by its thread id and block id. There are different types (levels) of memories available for GPU program. All threads can access global memory with large fetch latency, and threads of the same block can access faster in-chip block-shared memory, in addition, each thread has its private register-stored memory.

To efficiently harness the enormous computing power of GPUs, some technical aspects, quite different from conventional CPU parallel computing, need to be addressed in the implementation of GPU programs. The most important are memory coalescing and instruction diverging. For the consideration of efficiency and hardware cost, a block is partitioned into warps, consisting of 32 threads, and the GPU process half-warp of threads simultaneously. Therefore, instruction diverging due to branching needs to be avoided within a warp to minimize threads idling. During the computation, all threads within an active warp inevitably need to access the global memory. If the targeting memory addresses for different threads are adjacent, the system will coalesce all threads’ instructions together and fetch the whole chunk of required memory data at once, which amortize the fetch latency to each thread and greatly improve the averaged fetching speed in comparison to the case of discontinuous one. Also a single
access to the global memory has a latency of several hundred-clock cycles. To hide this latency, a multi-processor will schedule other warps for processing if available and ready.

III. Results

We have carried out a set of preliminary studies on the solution of RTE by employing the DG scheme and GPU.

A. Validation of RTE Solver

We first validated our RTE solver with some benchmark problems. Fig. 3(a) shows the geometry and non-dimensional emissive power profile along the symmetric centerline of a two-dimensional rectangular enclosure with absorbing, emitting and an isotropically scattering gray medium. Fig 3(b) compares the results obtained with our RTE solver with those reported in the literature for optical thickness \( \tau_a = (\sigma_t + \kappa) L = 1.0 \) [Razzaque et al., 1983]. A discretization of 50 uniform grid points along the z- direction and \( S_0 \) discrete ordinate method for angular discretization were employed. The mean error in temperature was less than 5%.

For further validation, we have studied a classic benchmark problem of steady-state one-dimensional combined conduction and radiation in a plane-parallel slab of thickness L, by analogy with Katika and Pilon [2004]. The slab was assumed to be a gray absorbing, emitting but non-scattering medium with absorption coefficient \( \kappa \) and thermal conductivity \( k \). The boundary conditions were taken as \( T(z=0) = T_{lower} \) and \( T(z=L) = 0.5T_{lower} \). In this problem, the relative importance of the radiative and conductive heat transfer modes is indicated by the following dimensionless parameter:

\[
N = \frac{k\kappa}{4\sigma T_{lower}^3} \tag{14}
\]

where \( \sigma \) is the Stefan-Boltzmann constant. Also a discretization of 50 uniform grid points along the z- direction and \( S_0 \) discrete ordinate method for angular discretization were employed. A set of cases with different values of \( N \) was carried out. Fig. 4 shows the comparison of simulated dimensionless temperature profiles with the exact solution reported by Viskanta and Grosh [1962]. The error in the values of temperature computed by the present study and the solution reported by Viskanta and Grosh [1962] was less than 0.6% for all values of \( N \) between 10 and 0.

Figure 3: (a) Geometry configuration of a two-dimensional rectangular enclosure with gray participating medium, and (b) non-dimensional emission power profile along the centerline with results reported in the literature [Razzaque et al., 1983].
To identify the applicability of GPUs device for the acceleration of computational intensive task in the RTE solver, we have implemented two versions of programs of matrix inversion procedures for the CPU and the GPU, respectively. For demonstration, we solved the two-dimensional radiative equilibrium problem, shown in Fig. 5. The top and sidewalls were black and cold, the temperature of the bottom wall was specified, and the medium was gray, absorbing, emitting and scattering. The width of enclosure, and the absorption coefficient of medium were specified to unity, and the isotropically scattering coefficient of medium were specified to 10. Finite volume method with 48 angular directions was employed for the RTE solution, and all directions of radiation intensity were solved simultaneously. Table 1 lists the time spent for each iteration step using the BiCGStab scheme with different spatial resolution. It is important to point out that exactly the same numerical procedures were employed for the CPU and GPU versions of implementation. As we can see, a ten-fold speedup was achieved by employing the GPU.

Figure 4: Validation of numerical results for non-dimensional temperature profile for a one-dimensional slab with combined conduction /radiation and gray participating medium against exact solution reported in the literature [Viskanta and Grosh, 1962].

B. GPU Acceleration

To identify the applicability of GPUs device for the acceleration of computational intensive task in the RTE solver, we have implemented two versions of programs of matrix inversion procedures for the CPU and the GPU, respectively. For demonstration, we solved the two-dimensional radiative equilibrium problem, shown in Fig. 5. The top and sidewalls were black and cold, the temperature of the bottom wall was specified, and the medium was gray, absorbing, emitting and scattering. The width of enclosure, and the absorption coefficient of medium were specified to unity, and the isotropically scattering coefficient of medium were specified to 10. Finite volume method with 48 angular directions was employed for the RTE solution, and all directions of radiation intensity were solved simultaneously. Table 1 lists the time spent for each iteration step using the BiCGStab scheme with different spatial resolution. It is important to point out that exactly the same numerical procedures were employed for the CPU and GPU versions of implementation. As we can see, a ten-fold speedup was achieved by employing the GPU.

Figure 5: Geometric configuration of a two-dimensional square enclosure with emitting, absorbing, and isotropically scattering gray medium.
<table>
<thead>
<tr>
<th>Spatial resolution</th>
<th>GPU implementation</th>
<th>CPU implementation</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>21x21</td>
<td>0.0036 sec.</td>
<td>0.035 sec.</td>
<td>9.76</td>
</tr>
<tr>
<td>51x51</td>
<td>0.0152 sec.</td>
<td>0.205 sec.</td>
<td>13.46</td>
</tr>
<tr>
<td>101x101</td>
<td>0.0550 sec.</td>
<td>0.800 sec.</td>
<td>14.55</td>
</tr>
</tbody>
</table>

C. Discontinuous Galerkin Method

The DG scheme was used for solving the radiative transfer test problem, illustrated in Fig. 6(a). We considered a two-dimensional gray absorbing and emitting enclosure with black walls. The temperature and absorption coefficients vary within the domain as shown in Fig. 6(a). We simulated the problem using the p-0 and p-1 schemes with different spatial grid resolutions. The simulation results were symmetric about both axes. Fig. 6(b) shows the gradient of radiation heat flux along the x-axis. Since the solutions were symmetric, only half of the solution profiles are plotted for comparison. The simulation results of both p-0 and p-1 DG schemes converged as the grid refines. However, the p-1 scheme were able to provide fairly accurate result with only 21x21 grid cells, while the p-0 scheme requires much finer grid resolution, 81x81 cells, to achieve a solution of comparable accuracy.

![Diagram](image)

**Figure 6**: Solution of a two-dimensional enclosure with gray absorbing and emitting medium, employing the p-0 and p-1 discontinuous Galerkin scheme and different grid resolution: (a) geometry, and (b) the gradient of radiative heat flux profiles along the centerline in x-direction.

IV. Conclusions and Future Work

In this paper, we have carried out a set of preliminary studies on the solution of RTE by employing the discontinuous Galerkin (DG) method and graphics processing units (GPU) and significant performance improvements have been achieved. In the future we will extend our DG based RTE solver’s capability such as local order adaptation, further exploit the computational power of GPU, and incorporate sophisticated non-gray radiative gas model into the RTE solver.
Acknowledgments

This work was supported by the U.S. Air Force STTR Phase-I contract # F08A-T020-0170.

References


