

High Performance Computing with a Conservative Spectral Boltzmann Solver

Jeffrey R. Haack* and Irene M. Gamba*,[†]

**Department of Mathematics, The University of Texas at Austin, 2515 Speedway, Stop C1200 Austin, Texas 78712*

[†]*ICES, The University of Texas at Austin, 201 E. 24th St., Stop C0200, Austin, TX 78712*

Abstract. We present new results building on the conservative deterministic spectral method for the space inhomogeneous Boltzmann equation developed by Gamba and Tharkabhushaman. This approach is a two-step process that acts on the weak form of the Boltzmann equation, and uses the machinery of the Fourier transform to reformulate the collisional integral into a weighted convolution in Fourier space. A constrained optimization problem is solved to preserve the mass, momentum, and energy of the resulting distribution. We extend this method to second order accuracy in space and time, and explore how to leverage the structure of the collisional formulation for high performance computing environments. The locality in space of the collisional term provides a straightforward memory decomposition, and we perform some initial scaling tests on high performance computing resources. We also use the improved computational power of this method to investigate a boundary-layer generated shock problem that cannot be described by classical hydrodynamics.

Keywords: Boltzmann equation, spectral methods, parallel computation

PACS: 51.10.+y, 47.11.Df, 47.11.Kb

INTRODUCTION

There are many difficulties associated with numerically solving the Boltzmann equation, most notably the dimensionality of the problem and the conservation of the collision invariants. For physically relevant three dimensional applications the distribution function is seven dimensional and the velocity domain is unbounded. In addition, the collision operator is nonlinear and requires evaluation of a five dimensional integral at each point in phase space. The collision operator also locally conserves mass, momentum, and energy, and any approximation must maintain this property to ensure that macroscopic quantities evolve correctly.

Spectral methods are a deterministic approach that compute the collision operator to high accuracy by exploiting its Fourier structure. These methods grew from the analytical works of Bobylev [1] developed for the Boltzmann equation for Maxwell type potential interactions and integrable angular cross section, where the corresponding Fourier transformed equation takes a closed form. Spectral approximations for this type of models were first proposed by Pareschi and Perthame [2]. Later Pareschi and Russo [3] applied this work to variable hard potentials by periodizing the problem and its solution and implementing spectral collocation methods.

These methods require $O(N^{2d})$ operations per evaluation of the collision operator, where N is the total number of velocity grid points in each dimension. While convolutions can generally be computed in $O(N^d \log N)$ operations, the presence of the convolution weights requires the full $O(N^{2d})$ computation of the convolution, except for a few special cases, e.g., the Fokker-Planck-Landau collision operator [4, 5]. Spectral methods provide many advantages over Direct Simulation Monte Carlo Methods (DSMC) because they are more suited to time dependent problems, low Mach number flows, high mean velocity flows, and flows that are away from equilibrium. In addition, deterministic methods avoid the statistical fluctuations that are typical of particle based methods.

Inspired by the work of Ibragimov and Rjasanow [6], Gamba and Tharkabhushanam [7, 8] observed that the Fourier transformed collision operator takes a simple form of a weighted convolution and developed a spectral method based on the weak form of the Boltzmann equation that provides a general framework for computing both elastic and inelastic collisions. Macroscopic conservation is enforced by solving a numerical constrained optimization problem that finds the closest distribution function in L_2 to the output of the

collision term that conserves the macroscopic quantities. These methods do not rely on periodization but rather on the use of the FFT tool in the computational domain, where convergence to the solution of the continuous problem is obtained by the use of the extension operator in Sobolev spaces.

This paper presents extensions to this method and its implementation on high performance computing resources. Without loss of generalization, we restrict this presentation to elastic collisions. We present a second order in time and space extension of the Gamba and Tharkabhushanam method allowing for nonuniform grids. This method has been implemented on the Lonestar supercomputer at the Texas Advanced Computing Center (TACC) and timing studies are provided to explore the scaling of the method to more and more processors for large problems. Finally we present some 1D in physical space results for a problem proposed by Aoki et al.[9] where a sudden change in wall temperature results in a shock that cannot be explained by classical hydrodynamics.

THE SPACE INHOMOGENEOUS BOLTZMANN EQUATION

The space inhomogeneous initial-boundary value problem for the Boltzmann equation is given by

$$\frac{d}{dt}f(\mathbf{x}, \mathbf{v}, t) + \mathbf{v} \cdot \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{v}, t) = \frac{1}{\varepsilon} Q(f, f), \quad (1)$$

with

$$\begin{aligned} \mathbf{x} &\in \Omega \subset \mathbb{R}^d, & \mathbf{v} &\in \mathbb{R}^d \\ f(\mathbf{x}, \mathbf{v}, 0) &= f_0(\mathbf{x}, \mathbf{v}) \\ f(\mathbf{x}, \mathbf{v}, t) &= f_B(\mathbf{x}, \mathbf{v}, t), & \mathbf{x} &\in \partial\Omega. \end{aligned}$$

where $f(\mathbf{v}, t)$ is a probability density distribution in \mathbf{v} -space and f_0 is assumed to be locally integrable with respect to \mathbf{v} and the spatial boundary condition f_B will be specified in below. The dimensionless parameter $\varepsilon > 0$ is the scaled Knudsen number, which is defined as the ratio between the mean free path between collisions and a reference macroscopic length scale.

The collision operator $Q(f, f)(\mathbf{x}, \mathbf{v}, t)$ is a bilinear integral form local in t and \mathbf{x} and is given by

$$Q(f, f)(\cdot, \mathbf{v}, \cdot) = \int_{\mathbf{v}_* \in \mathbb{R}^d} \int_{\sigma \in S^{d-1}} B(|\mathbf{v} - \mathbf{v}_*|, \cos\theta) (f(\mathbf{v}'_*)f(\mathbf{v}') - f(\mathbf{v}_*)f(\mathbf{v})) d\sigma d\mathbf{v}_*, \quad (2)$$

where the velocities $\mathbf{v}', \mathbf{v}'_*$ are determined through a collision rule (3), depending on \mathbf{v}, \mathbf{v}_* , and the positive term of the integral in (2) evaluates f in the pre-collisional velocities that will take the direction \mathbf{v} after an interaction. The collision kernel $B(|\mathbf{v} - \mathbf{v}_*|, \cos\theta)$ is a given non-negative function depending on the size of the relative velocity $\mathbf{u} := \mathbf{v} - \mathbf{v}_*$ and $\cos\theta = \frac{\mathbf{u} \cdot \sigma}{|\mathbf{u}|}$, where σ in the $n - 1$ dimensional sphere S^{n-1} is referred as the scattering direction of the post-collisional elastic relative velocity.

For the following we will use the velocity elastic (or reversible) interaction law in center of mass-relative velocity coordinates

$$\begin{aligned} \mathbf{v}' &= \mathbf{v} + \frac{1}{2}(|\mathbf{u}|\sigma - \mathbf{u}), & \mathbf{v}'_* &= \mathbf{v}_* - \frac{1}{2}(|\mathbf{u}|\sigma - \mathbf{u}) \\ B(|\mathbf{u}|, \cos\theta) &= |\mathbf{u}|^\lambda b(\cos\theta). \end{aligned} \quad (3)$$

We assume that the differential cross section function $b(\cos\theta)$ is integrable with respect to σ on S^{d-1} , referred to as the Grad cut-off assumption, and that it is renormalized such that

$$\int_{S^{d-1}} b(\cos\theta) d\sigma = 1. \quad (4)$$

The parameter λ regulates the collision frequency as a function of the relative speed $|\mathbf{u}|$. This corresponds to the interparticle potentials used in the derivation of the collisional kernel and are referred to as variable hard potentials (VHP) for $0 < \lambda < 1$, hard spheres (HS) for $\lambda = 1$, Maxwell molecules (MM) for $\lambda = 0$, and variable soft potentials (VSP) for $-3 < \lambda < 0$. The $\lambda = -3$ case corresponds to a Coulombic interaction potential between particles. If $b(\cos\theta)$ is independent of σ we call the interactions isotropic (like the case of hard spheres in three dimensions).

Boundary conditions

On the spatial boundary $\partial\Omega$ we use a diffusive Maxwell boundary condition which is given by, for $\mathbf{x} \in \partial\Omega$,

$$f(\mathbf{x}, \mathbf{v}, t) = \frac{\sigma_w}{(2\pi RT_w)^{(d/2)}} \exp\left(-\frac{|\mathbf{v} - \mathbf{V}_w|^2}{2RT_w}\right), \quad (\mathbf{v} - \mathbf{V}_w) \cdot \mathbf{n} > 0 \quad (5)$$

$$\sigma_w = -\left(\frac{2\pi}{RT_w}\right)^{1/2} \int_{(\mathbf{v} - \mathbf{V}_w) \cdot \mathbf{n} < 0} (\mathbf{v} - \mathbf{V}_w) \cdot \mathbf{n} f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v},$$

where \mathbf{V}_w and T_w are the wall velocity and temperature, respectively, and \mathbf{n} is the unit normal vector to the boundary, directed into Ω . The term σ_w accounts for the amount of particles leaving the domain and ensures mass conservation in Ω .

Spectral formulation

The key step our formulation of the spectral numerical method is the use of the weak form of the Boltzmann collision operator. For a suitably smooth test function $\phi(\mathbf{v})$ the weak form of the collision integral is given by

$$\int_{\mathbb{R}^d} Q(f, f) \phi(\mathbf{v}) d\mathbf{v} = \int_{\mathbb{R}^d \times \mathbb{R}^d \times S^{d-1}} f(\mathbf{v}) f(\mathbf{v}_*) B(|\mathbf{u}|, \cos\theta) (\phi(\mathbf{v}') - \phi(\mathbf{v})) d\sigma d\mathbf{v}_* d\mathbf{v} \quad (6)$$

If one chooses

$$\phi(\mathbf{v}) = e^{-i\zeta \cdot \mathbf{v}} / (\sqrt{2\pi})^d,$$

then (6) is the Fourier transform of the collision integral with Fourier variable ζ :

$$\begin{aligned} \widehat{Q}(\zeta) &= \frac{1}{(\sqrt{2\pi})^d} \int_{\mathbb{R}^d} Q(f, f) e^{-i\zeta \cdot \mathbf{v}} d\mathbf{v} \\ &= \int_{\mathbb{R}^d \times \mathbb{R}^d \times S^{d-1}} f(\mathbf{v}) f(\mathbf{v}_*) \frac{B(|\mathbf{u}|, \cos\theta)}{(\sqrt{2\pi})^d} (e^{-i\zeta \cdot \mathbf{v}'} - e^{-i\zeta \cdot \mathbf{v}}) d\sigma d\mathbf{v}_* d\mathbf{v} \end{aligned} \quad (7)$$

$$= \int_{\mathbb{R}^d} G(\mathbf{u}, \zeta) \mathcal{F}[f(\mathbf{v}) f(\mathbf{v} - \mathbf{u})](\zeta) d\mathbf{u}, \quad (8)$$

where $\widehat{[\cdot]} = \mathcal{F}(\cdot)$ denotes the Fourier transform and

$$G(\mathbf{u}, \zeta) = |\mathbf{u}|^\lambda \int_{S^{d-1}} b(\cos\theta) \left(e^{-i\frac{\zeta}{2} \cdot |\mathbf{u}| \sigma} e^{i\frac{\zeta}{2} \cdot \mathbf{u}} - 1 \right) d\sigma \quad (9)$$

Further simplification can be made by writing the Fourier transform inside the integral as a convolution of Fourier transforms:

$$\widehat{Q}(\zeta) = \int_{\mathbb{R}^d} \widehat{G}(\xi, \zeta) \widehat{f}(\zeta - \xi) \widehat{f}(\xi) d\xi, \quad (10)$$

where the convolution weights $\widehat{G}(\xi, \zeta)$ are given by

$$\widehat{G}(\xi, \zeta) = \frac{1}{(\sqrt{2\pi})^d} \int_{\mathbb{R}^d} G(\mathbf{u}, \zeta) e^{-i\xi \cdot \mathbf{u}} d\mathbf{u} \quad (11)$$

These convolution weights can be precomputed once to high accuracy and stored for future use. For many collision types the complexity of the integrals in the weight functions can be reduced dramatically through analytical techniques. In this paper we will only consider isotropic scattering in dimension 3 ($b(\cos\theta) = 1/4\pi$). In this case we have that

$$\widehat{G}(\xi, \zeta) = \frac{4\pi}{(\sqrt{2\pi})^3} \int_{\mathbb{R}^+} r^{\lambda+2} \left(\operatorname{sinc}\left(\frac{\beta r |\zeta|}{2}\right) \operatorname{sinc}(r|\xi - \frac{\beta}{2}\zeta|) - \operatorname{sinc}(r|\xi|) \right) dr. \quad (12)$$

This integral will be cut off at a point $r = r_0$, which will be determined below. Given this cutoff point, we can explicitly compute \widehat{G} for integer values of λ . For other values of λ , this is simply a one-dimensional integral that can be precomputed to high accuracy using numerical quadrature. The entirety of the collisional model being used is encoded in the weights, which gives the algorithm a large degree of flexibility in implementing different models.

THE CONSERVATIVE NUMERICAL METHOD

Temporal and velocity space discretization

We use an operator splitting method to separate the mechanisms of collisions and advection. The system is split into the subproblems

$$\frac{\partial}{\partial t} f + v \cdot \nabla_{\mathbf{x}} f = 0 \quad (13)$$

$$\frac{\partial}{\partial t} f = Q(f, f), \quad (14)$$

which are solved separately.

Each system is evolved in time using a second-order Runge-Kutta method, and the systems are combined using Strang splitting.

We assume that the distribution function is negligible outside of a ball

$$B_{R_x}(\mathbf{V}(\mathbf{x})) = \{\mathbf{v} \in \mathbb{R}^d : |\mathbf{v} - \mathbf{V}(\mathbf{x})| \leq R_x\}, \quad (15)$$

where $\mathbf{V}(\mathbf{x})$ is the local flow velocity which depends in the spatial variable \mathbf{x} . For ease of notation in the following we will work with a ball centered at 0 and choose a length R large enough that $B_{R_x}(\mathbf{V}(\mathbf{x})) \subset B_R(0)$ for all \mathbf{x} .

With this assumed support for the distribution f , the integrals in (10) will only be nonzero for $\mathbf{u} \in B_{2R}(0)$. Therefore, we set $L = 2R$ and define the cube

$$C_L = \{\mathbf{v} \in \mathbb{R}^d : |v_j| \leq L, j = 1, \dots, d\} \quad (16)$$

to be the domain of computation. For such domain, the computation of the weight function integral (12) is cut off at $r_0 = L$.

Let $N \in \mathbb{N}$ be the number of points in velocity space in each dimension. Then the uniform velocity mesh size is $\Delta v = \frac{2L}{N}$ and due to the formulation of the discrete Fourier transform the corresponding Fourier space mesh size is given by $\Delta \zeta = \frac{\pi}{L}$.

The mesh points are defined by

$$\begin{aligned} v_{\mathbf{k}} &= \Delta v (\mathbf{k} - N/2) \\ \zeta_{\mathbf{k}} &= \Delta \zeta (\mathbf{k} - N/2) \end{aligned} \quad (17)$$

$$\mathbf{k} = (k_1, \dots, k_d) \in \mathbb{Z}^d, \quad 0 \leq k_j \leq N-1, j = 1, \dots, d \quad (18)$$

Collision step discretization

Returning to the spectral formulation (10), the weighted convolution integral then becomes an integral over $-\frac{\pi}{\Delta v} \leq \xi_j \leq \frac{\pi}{\Delta v}, j = 1, \dots, d$.

To simplify notation we will use one index to denote multidimensional sums with respect to an index vector \mathbf{m}

$$\sum_{\mathbf{m}=0}^{N-1} = \sum_{m_1, \dots, m_d=0}^{N-1}.$$

To compute $\widehat{Q}(\zeta_{\mathbf{k}})$, we first compute the Fourier transform integral giving $\widehat{f}(\zeta_{\mathbf{k}})$ via the FFT. The weighted convolution integral is approximated using the trapezoidal rule

$$\widehat{Q}(\zeta_{\mathbf{k}}) = \sum_{\mathbf{m}=0}^{N-1} \widehat{G}(\xi_{\mathbf{m}}, \zeta_{\mathbf{k}}) \widehat{f}(\xi_{\mathbf{m}}) \widehat{f}(\zeta_{\mathbf{k}} - \xi_{\mathbf{m}}) \omega_{\mathbf{m}}, \quad (19)$$

where $\omega_{\mathbf{m}}$ is the quadrature weight and we set $\widehat{f}(\zeta_{\mathbf{k}} - \xi_{\mathbf{m}}) = 0$ if $(\zeta_{\mathbf{k}} - \xi_{\mathbf{m}})$ is outside of the domain of integration. We then use the inverse FFT on \widehat{Q} to calculate the integral returning the result to velocity space.

Note that in this formulation the distribution function is not periodized, as is done in the collocation approach of Pareschi and Russo [3]. This is reflected in the omission of Fourier terms outside of the Fourier domain. All integrals are computed directly only using the FFT as a tool for faster computation. The convolution integral is accurate to at least the order of the quadrature. The calculations below use the trapezoid rule, but in principle Simpson's rule or some other uniform grid quadrature can be used. However, it is known that the trapezoid rule is spectrally accurate for periodic functions on periodic domains (which is the basis of spectral accuracy for the FFT), and the same arguments can apply to functions with sufficient decay at the integration boundaries [10]. These accuracy considerations will be investigated in future work. The overall cost of this step is $O(N^{2d})$.

Discrete conservation enforcement

This implementation of the collision mechanism does not conserve all of the quantities of the collision operator. To correct this fact, we formulate these conservation properties as a constrained optimization problem as proposed in [7, 8]. Depending on the type of collisions we can change this constraint set (for example, inelastic collisions do not preserve energy). We focus here just on the case of elastic collisions, which preserve mass, momentum, and energy.

Let $M = N^d$ be the total number of grid points, let $\tilde{\mathbf{Q}} = (\tilde{Q}_1, \dots, \tilde{Q}_M)^T$ be the result of the spectral formulation from the previous section, written in vector form, and let ω_j be the quadrature weights over the domain in this ordering. Define the integration matrix

$$\mathbf{C}_{5 \times M} = \begin{pmatrix} \omega_j \\ v_j^i \omega_j \\ |\mathbf{v}_j|^2 \omega_j \end{pmatrix},$$

where $v^i, i = 1, 2, 3$ refers to the i th component of the velocity vector. Using this notation, the conservation method can be written as a constrained optimization problem.

$$\text{Find } \mathbf{Q} = (Q_1, \dots, Q_M)^T \text{ that minimizes } \frac{1}{2} \|\tilde{\mathbf{Q}} - \mathbf{Q}\|_2^2 \text{ such that } \mathbf{C}\mathbf{Q} = \mathbf{0} \quad (20)$$

The solution is given by

$$\begin{aligned} \mathbf{Q} &= \tilde{\mathbf{Q}} + \mathbf{C}(\mathbf{C}\mathbf{C}^T)^{-1} \mathbf{C}\tilde{\mathbf{Q}} \\ &:= \mathbf{P}_N \tilde{\mathbf{Q}} \end{aligned} \quad (21)$$

The overall cost of the conservation portion of the algorithm is a $O(N^d)$ matrix-vector multiply, significantly less than the computation of the weighted convolution.

Spatial and Transport discretization

For simplicity this will be presented in 1D in space, though the ideas apply to higher dimensions. In this case the transport equation reduces to

$$\frac{\partial f}{\partial t}(x, \mathbf{v}, t) + v_1 \frac{\partial}{\partial x} f(x, \mathbf{v}, t) = 0.$$

We partition the domain into cells of size Δx_j (not necessarily uniform) with cell centers x_j . Using a finite volume approach, we integrate the transport equation over a single cell to obtain

$$\frac{f_j^{n+1}(v_i) - f_j^n(v_i)}{\Delta t} + \frac{F_{j+1/2}^n - F_{j-1/2}^n}{\Delta x_j} = 0,$$

where $t^n = n\Delta t$ and $F_{j\pm 1/2}^n$ is an approximation of the edge fluxes $v_1 f$ of the cell between time t^n and t^{n+1} . We use a second order upwind scheme defined by

$$F_{j+1/2}^n = \begin{cases} v_1(f_j^n + \frac{\Delta x}{2}\sigma_j^n), & v_1 \geq 0 \\ v_1(f_{j+1}^n - \frac{\Delta x}{2}\sigma_{j+1}^n), & \text{otherwise,} \end{cases} \quad (22)$$

where σ_j is a cell slope term used in the reconstruction defined by the minmod limiter.

On wall boundaries the incoming flux is determined using ghost cells and the diffusive reflection formula (5). For problems without meaningful boundary interactions (e.g. shocks), a no-flux boundary condition is applied for the incoming characteristics.

PARALLELIZATION

The major bottleneck in parallelizing a program is memory access times related to communication between processes. However, the relative locality of the dynamics of the Boltzmann equation allow for a straightforward decomposition of phase space. In each time step, a single grid point only “sees” the particles at the same spatial grid point, through the collision term, and particles with the same velocity at neighboring spatial grid points, through the transport term. As most of the computational time is spent on the collision term we choose to keep all of the information needed for this step on the same computational node, and thus partition the spatial grid points across the computational nodes.

Further parallelization can be realized on each node by noting that the computation of $\widehat{Q}(\zeta)$ is simply a sum involving the precomputed weights \widehat{G} , the distribution function \hat{f} , and ζ , and is completely independent of computation on another velocity grid point. We use OpenMP [11] to distribute the weighted convolutions to each computational core on the node. Each core computes the convolution for a single grid point in ζ , marching through until all of the convolutions on the node are complete. Because there is no memory transfer required between cores, this should speed up the computation of the collisional terms by a factor of p , where p is the total number of cores used in the computation.

We use the Message Passing Interface (MPI) [12] to communicate distribution function data between nodes using an interleaved ghost cells technique [13]. Each node maintains two spatial ghost cells on the left and right sides of its local spatial domain, which are filled by the distribution functions sent from the neighboring node. Once this information is received, the node has enough information to update the “regular” spatial grid points through the finite volume transport scheme described above.

To make a rough estimate of the total speedup let n be the number of nodes used, and np be the number of cores used (assume a fixed number of cores per node). Then the speedup can be described by, to leading order,

$$\frac{T_{\text{SERIAL}}}{T_{\text{PARALLEL}}} = \frac{CN_x N^6 T_{\text{FLOP}}}{4nN^3 T_{\text{MEM}} + CN_x N^6 T_{\text{FLOP}}/np}, \quad (23)$$

where T_{MEM} is the time to transfer a single floating point piece of data, and T_{FLOP} is the time for a single floating point operation. As n becomes large with N and N_x fixed, more and more data transfer is required, however one would need $4n^2 p T_{\text{MEM}}/T_{\text{FLOP}} \approx CN_x N^3$ before the memory access time would begin to dominate the collisional computations.

NUMERICAL RESULTS

We test this method with the sudden change in wall temperature example suggested by Aoki et al. [9]. In this problem the gas is initially at equilibrium and the temperature of the wall at the boundary of the domain is instantaneously changed at $t = 0$. This gives rise to a discontinuous distribution function at the wall, which propagates into the domain and eventually forms a shock on the interior of the domain. This problem is especially well suited to a deterministic method because the discontinuity near the wall results in a distribution function that is far from equilibrium, which is difficult to simulate with Monte Carlo based solvers.

In Figure 1 we show shock formation due to a sudden change in wall temperature. Unlike the computations in [9], only two grid points are used per mean free path in the interior of the domain. Near the wall, the grid is refined to eight points per mean free path to better capture the finer dynamics where the distribution function is discontinuous. In both examples, the scaled Knudsen number $\varepsilon = 1$. Note that despite the discontinuity we do not observe any Gibbs phenomena in the solution. We hypothesize that this is due to the mixing effects of the convolution weights; this will be explored in future work.

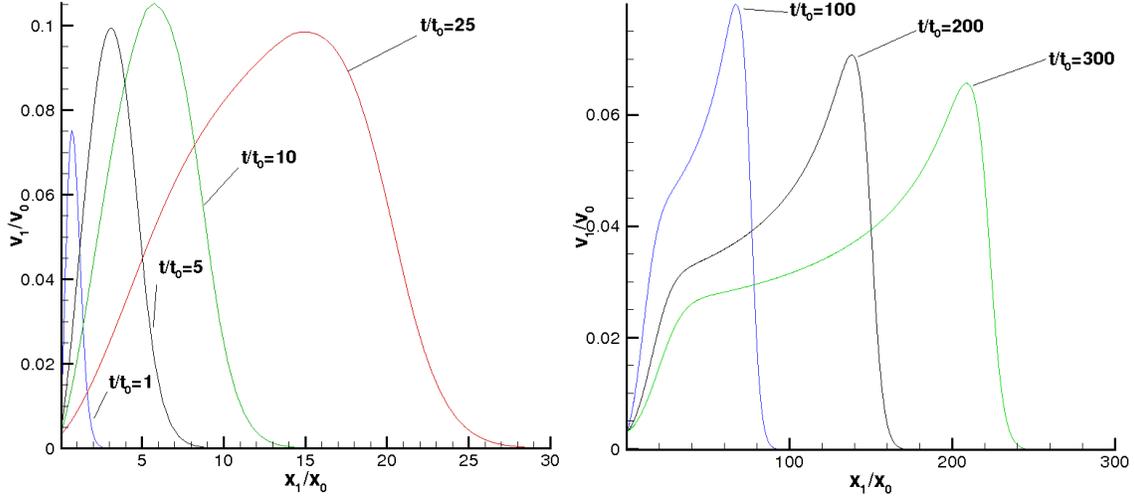


FIGURE 1. Formation of shock from sudden change in wall temperature. Evolution of bulk velocity.

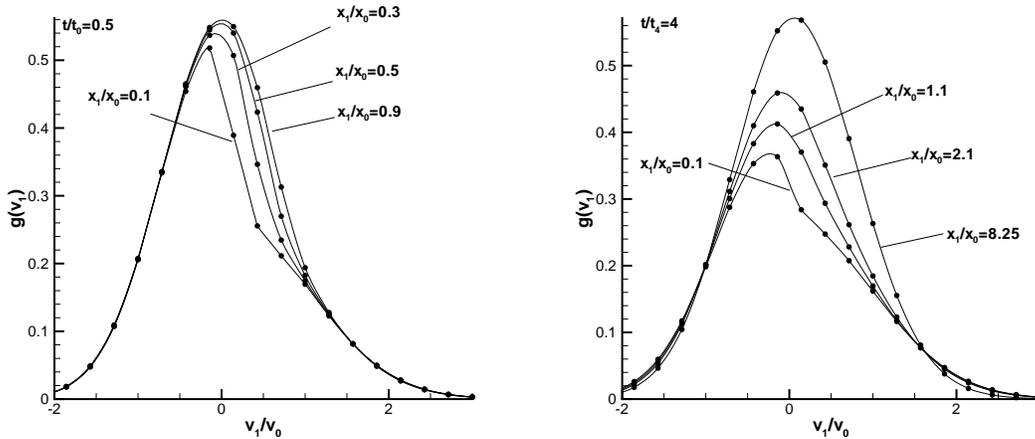


FIGURE 2. Sudden heating: evolution of discontinuous marginal distribution near the wall.

In Table 1 we show the wall time scaling results for the sudden heating example in Figure 1. This code was executed on the TACC supercomputer Lonestar, which has twelve cores per node. There is a jump in

computational time when moving from a single node to two nodes, but every doubling of the node amount thereafter results in an almost exact halving of the computational time required with the previous number of nodes scowing near-perfect linear scaling for this range of nodes.

TABLE 1. Computational time for a single timestep in sudden heating example from Figure 1.

nodes	cores	time (s)
1	12	203
2	24	235.3
4	48	120.8
8	96	61.4
16	192	30.9
32	384	15.2
64	768	7.7

CONCLUSIONS

We have extended the spectral method of Gamba and Tharkabhushanam to a second order scheme with a nonuniform grid in physical space, and investigated its scaling to high performance computers. The method showed nearly linear speedup when applied to a large computation problem solved on a supercomputer. However, at some point memory access will still become a problem, not with transfer between nodes but with memory access on a single node. The most expensive object to store is the six dimensional weight array $\widehat{G}(\zeta, \xi)$, and if N becomes too large it may not fit on a single node's memory, significantly slowing down computation. At that point, it may be faster to simply compute the weights on the fly, as flops are much cheaper than memory accesses on the large distributed systems used in high performance computing today. Future advances in hardware such as Intel's new Many Integrated Core (MIC) nodes, which will be used in TACC's new Stampede computer starting in 2013, have many more cores and much more local memory than ever before. Future work will investigate scaling when pushing the bounds of both number of cores p of the system as well as the memory requirements on each node.

ACKNOWLEDGMENTS

This work has been supported by the NSF under grant number DMS-0636586.

REFERENCES

1. A. V. Bobylev, *Mathematical Physics Reviews* **7**, 111–233 (1988), Soviet Sci. Rev. Sect. C Math. Phys.Rev., 7, Harwood Academic Publ., Chur.
2. L. Pareschi, and B. Perthame, *Trans. Theo. Stat. Phys.* **25**, 369–382 (1996).
3. L. Pareschi, and G. Russo, *SIAM J. Num. Anal.* pp. 1217–1245 (2000).
4. L. Pareschi, G. Russo, and G. Toscani, *J. Comput. Phys.* **165**, 216–236 (2000).
5. C. Mouhot, and L. Pareschi, *Math. Comput.* **75**, 1833–1852 (2006).
6. I. Ibragimov, and S. Rjasanow, *Computing* **69**, 163–186 (2002).
7. I. M. Gamba, and S. H. Tharkabhushanam, *J. Comput. Phys* **228**, 2016–2036 (2009).
8. I. M. Gamba, and S. H. Tharkabhushanam, *J. Comput. Math* **28**, 430–460 (2010).
9. K. Aoki, Y. Sone, K. Nishini, and H. Sugimoto, “Numerical analysis of unsteady motion of a rarefied gas caused by sudden changes of wall temperature with special interest in the propogation of a discontinuity in the velocity distribution function,” in *Rarefied Gas Dynamics*, edited by A. E. Beylich, VCH, Weinheim, 1991, pp. 222–231.
10. K. Atkinson, *An Introduction to Numerical Analysis*, John Wiley and Sons, New York, 1984.
11. OpenMP Architecture Review Board, OpenMP application program interface version 3.0 (2008), URL [\url{http://www.openmp.org/mp-documents/spec30.pdf}](http://www.openmp.org/mp-documents/spec30.pdf).
12. E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall, “Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation,” in *Proceedings, 11th European PVM/MPI Users’ Group Meeting, Budapest, Hungary, 2004*.

13. W. Gropp, E. Lusk, and A. Skjellum, *Using MPI : portable parallel programming with the message-passing interface*, MIT Press, 1999.