Eulerian Algorithms for the Discretization of Plasma Kinetic Equations

James Juno
IREAP
University of Maryland
College Park, Maryland 20742
jjuno@terpmail.umd.edu

Ammar Hakim (advisor)
Princeton Plasma Physics Laboratory
Princeton, New Jersey, 08543-0451
ahakim@pppl.gov

Jason TenBarge (advisor)
Princeton University
Princeton, New Jersey, 08544
tenbarge@princeton.edu

William Dorland (advisor)
IREAP
University of Maryland
College Park, Maryland 20742
bdorland@umd.edu

ABSTRACT

Plasmas are ubiquitous in nature, and the study of plasmas has application to a wide variety of problems, from the development of nuclear fusion to understanding the dynamic interaction between the solar wind and the Earth's magnetosphere. A challenge to progress in plasma physics is that many plasmas of interest are only weakly collisional and far from equilibrium, so the system is best described by kinetic theory, in this case, the Vlasov equation. Kinetic models like the Vlasov equation involve solving for the particle distribution function of the plasma in position and velocity space, making them costly because of their increased dimensionality when compared to traditional fluid models. Nonetheless, the microphysics of the plasma is extremely important to understand in order to answer many outstanding questions regarding a plasma's dynamics and thermodynamics. For example, the dissipation of turbulence via collisionless processes is likely related to the solar corona and solar wind heating problems, wherein the ions are significantly hotter than existing theory can predict.

We have developed a fully kinetic Vlasov-Maxwell solver using the `Gkeyll` framework [5]. In this framework, the Vlasov-Maxwell system is discretized in a combined position-velocity phase space with a discontinuous Galerkin (DG) algorithm [2, 3, 4]. The discontinuous Galerkin method combines the power of finite element methods, including high order accuracy and the ability to handle complicated geometries, with the advantages of finite volume methods, including the introduction of limiters to enforce stability and physicality of the solution, and locality of data, for efficient parallelization. The high order accuracy and locality of data in particular make our approach especially advantageous; by using higher order polynomials we can maximize the arithmetic intensity of the algorithm, effectively increasing the resolution of a simulation at a fraction of the cost, and locality of data is critical to performing five and six dimensional simulations, where communication overhead must be minimized to maintain scalability. The solution is advanced in time with a third order strong-stability preserving Runge-Kutta method.

While [5] includes a number of key algorithmic advances to increase the feasibility of solving a six dimensional partial differential equation in time, such as a reduced polynomial basis set for the finite element discretization, commonly referred to as the Serendipity Basis set in the literature [1], we focus in this poster on two additional algorithmic advances which further improve the performance and scaling of our DG discretization of the Vlasov-Maxwell system. The first improvement is the use of a computer algebra system, in this case Maxima, which pre-generates the required computational kernels to perform a forward Euler time step of the system of equations. Such a re-factoring is possible due to the underlying structure of our algorithm, which dominantly consists of tensor-tensor multiplications to produce the update of the Vlasov-Maxwell system in a forward Euler time step. When the required tensors are sparse, as they are when the finite element polynomial expansion is composed of orthonormal polynomials, then the tensor-tensor multiplications can be neatly unrolled via Maxima, increasing the performance dramatically over a strict numerical quadrature routine. Such a performance improvement is not just a simple reduction in the number of operations required to take a time step, but also due to how favorably the compiler interprets the hand-unrolled tensor-tensor convolutions produced by Maxima.

The second improvement is the use of MPI-3's shared memory directives. While many codes use a hybrid MPI-OpenMP implementation to achieve greater performance on more modern architectures such as Intel's Knight's Landing and Skylake chips, architectures which sacrifice some amount of core to memory ratio for increased overall speed, we find an MPI-only implementation using the full functionality of MPI-3 to be more favorable from a design perspective. This favorability is because the back-end of the

Supercomputing '18 Dallas, TX, USA

DOI: 10.1145/1235

`Gkeyll` framework is written in Lua with a Just-In-Time compiler, or LuaJIT, and thus it is more straightforward to wrap MPI commands using the foreign function interface than it is to wrap pre-processor directives. Further, an MPI-only implementation eliminates issues known to plague hybrid parallel algorithms, such as thread latency and overhead. Since the shared MPI processes behave just like regular MPI processes, there is no need to spawn threads every time the algorithm enters a multi-threaded loop. The result is a lightweight framework which can scale very well on the aforementioned modern architectures.

The results of these improvements are shown via scaling studies and performance benchmarks of the solver. In addition, physics benchmarks are presented to demonstrate the functionality of the tool. Further details of the problems of interest in the plasma physics community, as well as the benefits of a Vlasov-Maxwell solver, are expounded upon, and information on how to download and build the code, obtain sample input files, and run simulations is given.

Keywords

Discontinuous Galerkin; Vlasov/Boltzmann-Maxwell, MPI-3, Computer Generated Kernels

1. ACKNOWLEDGMENTS

This work is supported by a NASA Earth and Space Science Fellowship, grant no. 80NSSC17K0428, and NSF Grant No. AGS-1338944.

2. REFERENCES

[1] D. N. Arnold and G. Awanou. The serendipity family of finite elements. *Found. Comp. Math.*, 11:337–344, 2011.

[2] B. Cockburn and C.-W. Shu. The Runge–Kutta Discontinuous Galerkin Method for Conservation Laws V. *Journal of Computational Physics*, 141:199–224, Apr. 1998.

[3] B. Cockburn and C. W. Shu. Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16(3):173–261, 2001.

[4] J. S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis and Applications*. Springer, 2008.

[5] J. Juno, A. Hakim, J. TenBarge, E. Shi, and W. Dorland. Discontinuous galerkin algorithms for fully kinetic plasmas. *Journal of Computational Physics*, 353:110 – 147, 2018.