Vlasiator: Hybrid-Vlasov Simulation Code for the Earth's Magnetosphere

Sebastian von Alfthan
Finnish Meteorological Institute
P.O. BOX 503, FI-00101
Helsinki, Finland
sebastian.von.alfthan@fmi.fi

Yann Kempf
Finnish Meteorological Institute
P.O. BOX 503, FI-00101
Helsinki, Finland
yann.kempf@fmi.fi

Dimitry Pokhotelov
Department of Physics,
University of Helsinki
P.O. Box 44, FI-0014
Helsinki, Finland
dimitry.pokhotelov@fmi.fi

ABSTRACT

Here we present Vlasiator, the first global magnetospheric simulation code based on a hybrid-Vlasov description of plasma. Ions are represented by a six-dimensional distribution function, while electrons are modeled as a charge neutralizing fluid. To propagate the distribution function we use a finite volume method, coupled to an upwind constrained transport method for propagating electromagnetic fields. We have parallelized the code with a two-level MPI and OpenMP scheme, and it scales well to tens of thousands of cores. The hybrid-Vlasov approach is enabled through a novel sparse representation of the distribution function that reduces the problem size by more than two orders of magnitude. The capabilities of the code are demonstrated by reproducing key features of the collisionless bowshock in a global five-dimensional magnetospheric simulation. Vlasiator produces the first uniformly discretized ion velocity distribution functions with quality comparable to spacecraft measurements, opening up new opportunities for studying the Earth's magnetosphere.

1. INTRODUCTION

Vlasiator [4, 5] [*http://vlasiator.fmi.fi*] is the first hybrid-Vlasov simulation code that is designed to simulate Earth's magnetosphere on a global scale. The most popular global models have been based on the magnetohydrodynamics (MHD) equations, where plasma is modeled as a fluid. Increasing computational resources enable hybrid approaches to be used which significantly improves the physical description. In the hybrid approach electrons are modeled as a fluid and ions are described by a kinetic model. Typically hybrid approaches are hybrid particle-in-cell (PIC) simulations, where ions are macroparticles for which plasma kinetic equations are solved. This approach may yield solutions with an undesirable level of numerical noise, especially when looking at ion velocity distribution functions. Another approach is to use Vlasov's equation and model ions as a six-dimensional distribution function (three in ordinary space, and three in velocity space) yielding a hybrid-Vlasov approach. This approach does not have numerical noise, but does require massive amounts of memory and computation. Good performance and scalability is thus required, and the main techniques for achieving this target is presented here.

2. NUMERICAL MODEL

Ions are modeled as a six-dimensional distribution function f that obeys Vlasov's equation

$$\frac{\partial}{\partial t} f + \mathbf{v} \cdot \nabla_r f + \frac{q}{m}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_v f = 0, \tag{1}$$

where $\mathbf{r}$ and $\mathbf{v}$ are the spatial and velocity coordinates, $f = f(\mathbf{r}, \mathbf{v}, t)$ is the six-dimensional phase-space density of a particle species with mass m and charge q, $\mathbf{E}$ is the electric field and $\mathbf{B}$ is the magnetic field. The bulk parameters of the plasma, such as the ion charge density ρ_q and bulk velocity $\mathbf{V}_\mathrm{i}$, are obtained as velocity moments of the ion velocity distribution function. Vlasov's equation is coupled with Maxwell's equations in which the displacement current in Ampère-Maxwell's law is neglected. The system is closed by Ohm's law that models the effect of electrons. Vlasiator is currently using an ideal Ohm's law for updating fields,

$$\mathbf{E} = -\mathbf{V}_\mathrm{i} \times \mathbf{B}. \tag{2}$$

In the Lorentz force in Eq. (1) $\mathbf{E}$ also includes the hall term $\mathbf{j} \times \mathbf{B}/\rho_q$, where $\mathbf{j} = (\nabla \times \mathbf{B})/\mu_0$ is the total current.

In Vlasiator we use in ordinary space a three-dimensional Cartesian mesh to describe the system. Each spatial cell contains field variables and a three-dimensional Cartesian velocity mesh where each velocity cell contains the volume average of the distribution function. The velocity mesh is structured into blocks, with $4{\times}4{\times}4$ velocity cells each. This decreases overhead, and improves cache locality. Vlasiator uses a second-order accurate finite volume method (FVM) to solve Vlasov's equation. The method uses Strang splitting to separate propagation in ordinary and velocity spaces, and is based on a three-dimensional wave-propagation algorithm [2]. The field solver is a divergence-conserving second-order accurate upwind constrained transport method [3].

3. PARALLELIZATION

We have implemented a hybrid MPI-OpenMP parallelization scheme. This enables the code to scale to less than one spatial cell per core. It also improves the performance of the MPI-based domain decomposition due to decreased communication, and improved load balance.

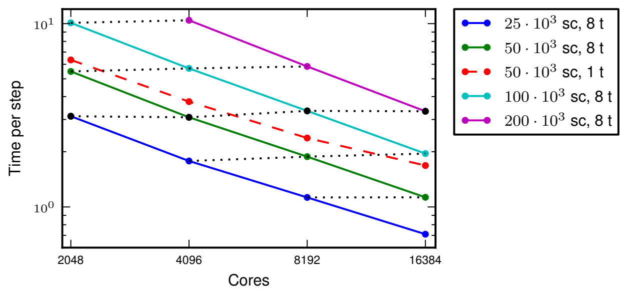

Figure 1: Scalability of a 5D test with 80×80×80 velocity cells per spatial cell (sc), run with 1 or 8 threads (t). Black dotted lines indicate weak scaling.

We use the DCCRG grid library [1] to implement the parallel three-dimensional ordinary space grid. It is parallelized using MPI with a domain decomposition strategy and uses non-blocking send and receive routines to overlap communication and computation. The Zoltan library is used for load balancing. OpenMP threading is implemented using a loop based approach. In the field solver we have threaded all major loops over spatial cells. In the Vlasov solver we have threaded the loops over velocity blocks in each spatial cell in ordinary space propagation. In velocity space propagation we group the velocity blocks in all cells into superblocks of 2×2×2 blocks and compute the blocks in the same octant of all superblocks in one threaded loop to avoid race conditions.

We have tested the scalability on a Cray XE6 supercomputer. In Fig. 1 we have plotted the scalability of Vlasiator for a five-dimensional test case with homogeneous plasma. Close to ideal strong scaling and weak scaling are achieved. Eight threads per process is the optimal choice due to the machine having 8 cores per NUMA node. Fig. 1 also demonstrates reduced scalability when using 1 thread per process.

4. SPARSE VELOCITY GRID

A hybrid-Vlasov simulation is computationally demanding due to the large number of cells in the six-dimensional distribution function. A key technique for enabling large-scale global simulations in Vlasiator is the sparse velocity grid, where we neither propagate nor store the empty cells. In a typical simulation this can reduce the number of cells by up to two orders of magnitude. In the sparse representation a velocity block exists with all of its 64 velocity cells, or it does not exist at all. We define that a block has content if any of its 64 velocity cells has a density above a user-specified threshold value. A velocity block exists if it has content, or if it is a neighbor to a block with content in any of the six dimensions. If a non-content block has accumulated enough material to exceed the threshold it becomes a content-block and all of its neighbors are created. On the other hand, if we have an orphaned non-content block with no neighbors with content it is removed. Analytically Vlasov's equation conserves mass, as does our propagation algorithm in the limit of an infinite velocity space. A sparse velocity grid loses some mass, mainly through the outer faces of the grid. With reasonable threshold values the losses are negligible and all physically relevant populations are retained.

5. CONCLUSIONS

Vlasiator is parallelized with MPI and OpenMP to scale to high-end supercomputers in order to run the simulations on length scales comparable to the Earth's magnetosphere. An important algorithmic advancement has been done to increase performance: the distribution function is described with a sparse grid in velocity space.

Results of global magnetospheric simulations (not presented here) suggests that Vlasiator is able to reproduce the key features of solar wind–magnetosphere interactions such as the collisionless bowshock and the ion foreshock boundary. Characteristics of the backstreaming ion populations and associated electromagnetic waves are in agreement with observations. The most striking difference with respect to hybrid-PIC simulations is that hybrid-Vlasov velocity distribution functions appear as uniformly discretized functions similar to those seen in the experimental data in contrast to spiky distributions derived from hybrid-PIC simulations.

6. ACKNOWLEDGMENTS

The Quantifying Energy circulation in Space plasmas (QuESpace) project has received funding from the European Research Council under the European Community's seventh framework programme (FP-7/2007-2013/ERC) agreement no. 200141-QuESpace. The work has been supported by the Academy of Finland. We acknowledge that the results of this research have been achieved using the PRACE Research Infrastructure resource Hermit based in Germany at the High Performance Computing Center Stuttgart (HLRS).

7. ADDITIONAL AUTHORS

Ilja Honkonen (FMI, `ilja.honkonen@fmi.fi`),
Sanni Hoilijoki (FMI, `sanni.hoilijoki@fmi.fi`),
Arto Sandroos (FMI, `arto.sandroos@fmi.fi`) and
Minna Palmroth (FMI, `minna.palmroth@fmi.fi`)

8. REFERENCES

[1] I. Honkonen et al. Parallel grid library for rapid and flexible simulation development. *Comput. Phys. Commun.*, 184(4):1297–1309, 2013.

[2] J. O. Langseth et. al. A Wave Propagation Method for Three-Dimensional Hyperbolic Conservation Laws. *J. Comput. Phys.*, 165:126–166, Nov. 2000.

[3] P. Londrillo et. al. On the divergence-free condition in Godunov-type schemes for ideal magnetohydrodynamics: the upwind constrained transport method. *J. Comput. Phys.*, 195:17–48, Mar. 2004.

[4] M. Palmroth et. al. Preliminary testing of global hybrid-Vlasov simulation: Magnetosheath and cusps under northward interplanetary magnetic field. *J. Atmos. Sol.-Terr. Phys.*, 99(0):41–46, 2013.

[5] A. Sandroos et. al. Multi-GPU simulations of Vlasov's equation using Vlasiator. *Parallel Computing*, 39:306, 2013.