A Modular Software for Particle Simulations in Space Physics

Mats Holmström[1]

Swedish Institute of Space Physics (IRF), PO Box 812, SE-981 28 Kiruna, Sweden
matsh@irf.se
http://www.irf.se/~matsh/

Abstract. We present extensions of the FLASH solver for handling two types of particle simulations that often are used in space physics simulations — hybrid simulations of space plasmas and direct simulation Monte Carlo (DSMC) methods for particle collisions. FLASH is an open source parallel solver for general hydrodynamic problems. We present the design of the software modules and show the results of performance measurements. It is found that the particle extensions scales as well with the number of processors as the rest of the FLASH code.

1 Overview

In many regions of space collisions between particles are not frequent enough to justify a fluid approximation such as magnetohydrodynamics (MHD). That collisions are rare imply that the ions or neutrals are not thermalized (they have non-Maxwellian velocity distributions) and that kinetic effects are important (such as the finite gyro radius of ions in a magnetic field). One example of such a problem is the neutral particles in a planetary exosphere (outermost part of the atmosphere). There the neutrals travel along ballistic orbits and are affected by radiation pressure, photo-ionization, charge exchange with ions, and infrequent collisions. Another example is the interaction between the solar wind and Mars. There the ions move from the Lorentz force of the magnetic and electric fields, and the gyro radius of solar wind protons is of the same order as the planet's radius. Also, the low ion densities makes the solar wind essentially collisionless. In both these examples, particle methods can be used to self consistently solve for the motion of particles affected by fields and collisions. For charged particles we can use hybrid methods, where the electrons are represented as a massless fluid, and ions are represented as particles. One also needs to store the magnetic field on a grid, and it is self consistently updated from the motion of the ions. Collisions between particles can be handled by the direct simulation Monte Carlo (DSMC) method, where the computational domain is divided into cells, and collision pairs are randomly drawn in each cell with probability proportional to their crossections and relative velocities. Note that the DSMC method also easily handles collisions that change the participating particles, e.g., charge exchange or chemical reactions.

The drawback of particle methods is that they are computationally expensive. To minimize the statistical fluctuations inherent in averaging discrete particle populations over grid cells we need to have a sufficient number of particles in each cell. The spatial resolution of the simulation is determined by the size of the grid cells, and this size in turn limits the size of the time step. But, since all computations are local, these types of particle simulations are suitable for parallelization. The requirements for hybrid and DSMC simulations are similar in that we need to be able to represent and move particles, and that we need to have a grid. By using an adaptive, non-uniform, grid we can gain computational efficiency by having small cells only where it is needed (where we have many particles). It is also possible to introduce adaptivity by joining and splitting particles.

However, the requirement of the software to be parallel *and* handle an adaptive grid makes it non-trivial. Therefore we have chosen to extend an existing software (FLASH)[2] with the necessary modules to be able to do hybrid and DSMC simulations.

FLASH is a general parallel solver for compressible flow problems. It is written in Fortran 90, well structured into modules, and open source. The parallelization is to a large extent handled by the PARAMESH[3] library that implements a block-structured adaptive cartesian grid with the Message-Passing Interface (MPI) library as the underlying communication layer. Although FLASH was originally written to solve fluid flow problems, there is now support for particles also. However, most of the support for particle simulations is for gravitational interaction, i.e. N-body problems with a gravitational potential on the grid. The particle simulations considered here (hybrid and DSMC) have local instead of global couplings between the particles and require support for slightly different operations, e.g., accurate update of the magnetic field on the grid from computed currents, and iteration over the particles in each cell for the selection of collision pairs. The different operations performed during one time step of a DSMC simulation are shown in Figure 1.

Here we describe the extensions we have made to the FLASH code to allow hybrid and DSMC simulations. The additional modules were written in a general way to be useful in any type of particle simulations involving local interactions. In fact, there is also a clear separation from the FLASH code itself to make possible the use of another software for handling parallelization and the adaptive grid. This separation and layering of the software is illustrated in Figure 2. Also, we have kept in mind the possibility of future inclusions of fluid flows coupled to the particles, e.g., a more realistic electron fluid. We have previously developed modules for MHD simulations of the interaction between solar-system objects and the solar wind[1], and combining these two models enables a coupling between particle and fluid simulations. An example of such a coupled simulation is the MHD flow around Mars, coupled to a particle model of Mars' exosphere that is a computationally easier problem than a full hybrid simulation.

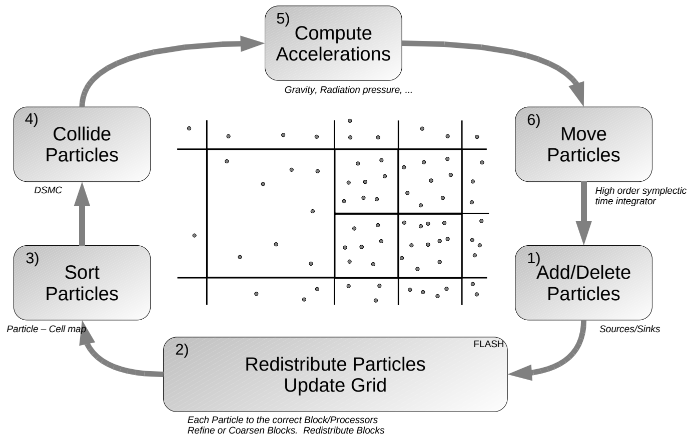

Fig. 1. An illustration of a DSMC particle simulation cycle that is performed for each time step.

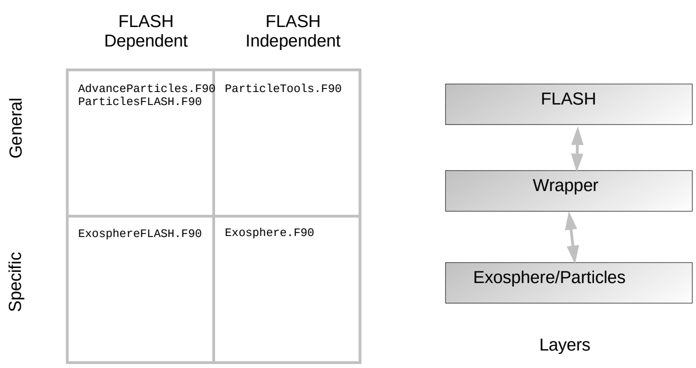

Fig. 2. An illustration of the software design. To the left it is shown that the modules and subroutines can be sorted along two dimensions. Along the y-axis: General for all particle simulations, and problem specific for a specific class of problems, e.g., exosphere simulations. Along the x-axis: The interaction with the library handling the adaptive grid and the parallelization. Either routines are specific for the used library (FLASH in the present work) or they can be used with any library. To the right we show this layering that separates all FLASH-dependent parts of the software.

We present the design of the software modules and show the results of performance measurements. It is found that the particle extensions scales as well with the number of processors as the rest of the FLASH code.

The physical problems that are investigated so far using the developed software is the exosphere of Mars (DSMC) and the interaction between the solar wind and Mars (hybrid) with the goal of coupling the two models.

2 Acknowledgements

The software used in this work was in part developed by the DOE-supported ASC / Alliance Center for Astrophysical Thermonuclear Flashes at the University of Chicago.

References

1. A. Ekenbäck, and M. Holmström, MHD modeling of the interaction between the solar wind and solar system objects. in *PARA 2004*, J. Dongarra, K. Madsen, and J. Waśniewski, Eds., Lecture Notes in Computer Science 3732, Springer-Verlag, 2005 (554–562)
2. B. Fryxell et al., FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes. Astrophysical Journal Suppl. **131** 2000 (273–)
   `http://flash.uchicago.edu/`
3. P. MacNeice, K.M. Olson, C. Mobarry, R. deFainchtein, and C. Packer, PARAMESH : A parallel adaptive mesh refinement community toolkit. Computer Physics Communications **126** 2000 (330–)
   `http://sourceforge.net/projects/paramesh/`