

MHD Modeling of the Interaction Between the Solar Wind and Solar System Objects

Andreas Ekenbäck and Mats Holmström

Swedish Institute of Space Physics (IRF)

P.O. Box 812

98134 Kiruna, Sweden

{andreas.ekenback,mats.holmstrom}@irf.se

Abstract. Magnetohydrodynamic (MHD) models of plasma can be used to model many phenomena in the solar system. In this work we investigate the use of a general MHD solver - the Flash code - for the simulation of the interaction between the solar wind and solar system objects. As a test case we simulate the three-dimensional solar wind interaction with a simplified model of a comet and describe the modifications of the code. The simulation results are found to be consistent with previously published ones. We investigate the performance of the code by varying the number of processors and the number of grid cells. The code is found to scale well. Finally we outline how to simulate the solar wind interaction with other objects using the Flash code.

1 Introduction

The solar wind is a plasma consisting mostly of protons and electrons that are ejected from the Sun's corona, and streams radially outward through the solar system at speeds of several hundred kilometers per second. Embedded into the solar wind is also the interplanetary magnetic field. When this highly supersonic solar wind meets with solar system objects, such as planets, comets and asteroids, an interaction region is formed. The type of the interaction depends on the object. Around planets with a strong internal magnetic field, such as Earth, a cavity is formed, a magnetosphere, shielding the upper atmosphere from direct interaction with the solar wind. This shielding is missing at planets without a strong intrinsic field, e.g., at Mars. Then currents in the planet's ionosphere form an obstacle that diverts the solar wind flow. Objects without any significant atmosphere, such as the Moon, are basically just physical obstacles to the flow, and a wake is formed behind the object. Comets have a very small nucleus compared to the size of the surrounding cloud of ions and neutrals and their interaction with the solar wind is governed by photoionization of neutrals and charge-exchange between ions and neutrals.

In space physics, the numerical modeling of the interaction between the solar wind and solar system objects is an important tool in understanding the physics of the interaction region. The results of simulations is also important as inputs to further modeling, e.g., to predict loss of planetary atmospheres due to the interaction.

Although self consistent particle and kinetic models best capture the actual physics, the computational cost at present limits the number of particles and the refinement of

grids. Since the computational cost of using a fluid model is less, more refined grids can be used. Thus, the modeling of the interaction between the solar wind and solar system objects is often done by using magnetohydrodynamic (MHD) fluid models.

In this work we investigate the possibility of using an existing open source application - the Flash code [1] developed at University of Chicago - to make MHD simulations of the interaction between the solar wind and a non-magnetized object. We model a solar wind and a comet - in the form of a photoionization source - and outline how the simulation model could be generalized to simulate other objects in the solar system.

2 Magnetohydrodynamics

Over small spatial volumes, the average properties of a plasma can be described using the basic conservation laws for a fluid [2]. The plasma is however conducting, so the effects of electric and magnetic fields and currents are added to ordinary hydrodynamics. In the presence of a magnetic field, the added conductivity of a fluid will cause a different behavior due to the interaction between the magnetic field and the motion; electric currents induced in the fluid (as a result of its motion) modify the field, and at the same time the flow of the currents in the field causes mechanical forces which modify the motion.

2.1 MHD Equations

The MHD model used for most simulations is valid under a number of assumptions, listed in full in [3], the fundamental ones being that the time scale of interest must be long compared to microscopic particle motions and that the spatial scale of interest must be large compared to the Debye length and the thermal gyroradius [2]. MHD models have nevertheless proved adequate to describe several physical processes in the solar system [3–5], including the solar wind-comet interaction [6, 7].

In what follows we use the notation:

\mathbf{r}	is position
ρ	is mass density
\mathbf{v}	is the velocity of the fluid
\mathbf{B}	is the magnetic field
μ_0	is permeability in vacuum = $4\pi \times 10^{-7}$ Vs/Am
p	is the plasma pressure
ε	is the internal energy
\mathbf{g}	is external forces such as gravity
T	is the temperature of the fluid
t	is elapsed time

It is common to use the MHD equations in their dimensionless, or normalized, form. We choose to use a reference velocity v_0 , a reference density ρ_0 and a reference length R_0 . These three factors are then used to scale all quantities in the MHD equations. Letting an overbar indicate the variables in SI units we set

$$\mathbf{r} = \frac{\bar{\mathbf{r}}}{R_0} \quad \rho = \frac{\bar{\rho}}{\rho_0} \quad \mathbf{v} = \frac{\bar{\mathbf{v}}}{v_0}$$

which for the other interesting parameters leads to

$$t = \frac{v_0}{R_0} \bar{t} \quad \mathbf{B} = \frac{\bar{\mathbf{B}}}{\sqrt{\mu_0 \rho_0 v_0^2}} \quad T = \frac{\bar{T}}{v_0^2} \quad p = \frac{\bar{p}}{\rho_0 v_0^2}$$

Approximating the plasma as collisionless and without resistivity we can use the equations of ideal MHD [2, 8]. Using the ideal limit of the one-fluid MHD equations in conservative form we will solve in Flash

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (2.1)$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v} - \mathbf{B} \mathbf{B}) = -\nabla \left(p + \frac{B^2}{2} \right) + \rho \mathbf{g} \quad (2.2)$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot \left[\mathbf{v} \left(\rho E + p + \frac{B^2}{2} \right) - \mathbf{B} (\mathbf{v} \cdot \mathbf{B}) \right] = \rho \mathbf{g} \cdot \mathbf{v} \quad (2.3)$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{v} \mathbf{B} - \mathbf{B} \mathbf{v}) = 0 \quad (2.4)$$

where

$$E = \frac{1}{2} v^2 + \varepsilon + \frac{1}{2} \frac{B^2}{\rho}$$

The MHD system is in the Flash code complemented by the equation of state, relating the pressure to internal energy or temperature:

$$\bar{p} = (\gamma - 1) \bar{\rho} \bar{\varepsilon} \quad \bar{p} = \frac{N_a k_B}{M} \bar{\rho} \bar{T}$$

where

- γ is the adiabatic index
- N_a is the Avogadro number = 6.022×10^{23}
- M is the mean atomic mass [kg]
- k_B is the Boltzmann constant = 1.38×10^{-23} J/K

3 Use of the Open Source Flash Code

The Flash code is a modular, adaptive, parallel simulation code capable of handling general compressible flow problems in astrophysical environments. The Flash code is written mainly in Fortran90 and uses the Message-Passing Interface (MPI) library [9] for inter-processor communication and portability. It further uses a customized version of the PARAMESH library [10, 11] to implement a block-structured adaptive cartesian grid, increasing resolution where it is needed.

By using the MHD module of the Flash code, we can benefit from all the testing, development and performance tuning that has been done. Flash is also distributed with an IDL-based visualization tool that we found very useful. What we add is the specific initial conditions, boundary conditions and source terms for the problem at hand. The

various examples provided with Flash facilitates the task of setting up a simulation model.

We here explain how to add a new problem to Flash by using the solar wind-comet interaction as an example. Where needed we also indicate how to model other possible situations.

3.1 Creating a Problem Directory

The Flash code was written with the intention of users being able to easily add their specific problems. The code is hence well prepared for adding a new problem; in the source code there is a directory `$FLASHHOME$/setups` where we create a directory `comet_mhd` for our problem. There are two files that must be included in a problem setup directory:

`Config` where it is specified which of the modules in Flash that should be used. It also lists (and sets default values to) required runtime parameters. In case we specify a module in our `Config` file with several submodules, there are default ones which are used unless we also specify the submodule.

`init_block.F90` to initialize data in each of the blocks of the mesh. We give the initial values in terms of ordinary geometrical coordinates and Flash takes care of the mapping to blocks. The code first initializes the blocks at the lowest level of refinement and then where needed refines the mesh and initializes the newly created blocks. If necessary this refinement procedure is repeated until we reach a highest level of refinement.

3.2 Boundary Conditions

To specify the boundary conditions it is possible to use some of the standard ones (“outflow”, “periodic” or “reflecting”) by a line for each boundary in the runtime parameter file. By stating a boundary as “user” Flash will use a file `user_bnd.F90` in the problem directory to set specified variable values there. Our solar wind inflow is modeled by setting velocity, density, magnetic field and temperature at the left x -boundary and setting the boundary condition at the right x -boundary as “outflow”. The simulation is periodic in the y - and z -directions.

3.3 Alternation to Standard Algorithms

In order to add the comet source terms for the MHD equations as specified in Section 4 it was necessary to make alterations in the main file for MHD calculations. This main file is `mhd_sweep.F90` and it was copied to our problem directory. Alternations are then made in our `mhd_sweep.F90` and this one will be used instead of the standard one. Calls to a created file `comet_source_mhd.F90`, that do additions to the right hand side of the equations (2.1) and (2.2), were inserted in `mhd_sweep.F90`. To see that our non-standard file `comet_source_mhd.F90` is compiled into the final executable it is necessary to have a `Makefile` in our problem directory stating the target `comet_source_mhd.o` and its dependencies.

3.4 Runtime Parameter File

The final file needed to run our comet simulation is the runtime parameter file `flash.par`. It should be placed in the directory from where Flash is run. Besides assigning values to parameters and specifying boundary conditions, it controls the output. Checkfiles are dumped at intervals specified in `flash.par` and can be used to restart simulations from. We can also control when smaller files, to be used with the enclosed visualization tool, containing only a desired selection of the variables are dumped.

4 A Test Case: The Solar Wind-Comet Interaction

To investigate the use of the Flash code for simulating the interaction between solar system objects and the solar wind we choose a comet as a test problem. We model the comet using a single fluid MHD. The setup follows Ogino et al. [6] where the comet is modeled as a spherically symmetric source of ions. Although much more refined comet models have been used later, e.g. by Gombosi et al. [7], we use this simpler model since we in this work primarily are interested in the performance and correctness of the code and our modifications.

Because of the small mass (and the resulting small gravitational forces) of the cometary nucleus, a large amount of gas evaporates from the nucleus. The gas extends into the solar wind and is ionized by charge-exchange processes and by photoionization caused by solar ultraviolet radiation. In the MHD equations the interaction process is, following Ogino et al. [6] (originally by Schmidt et al. [12]), modeled as a plasma production centered at the cometary nucleus. We set the cometary plasma production rate

$$\bar{A}(\bar{r}) = \frac{\bar{Q}_p e^{-\bar{r}/\bar{\lambda}_c}}{4\pi\bar{\lambda}_c\bar{r}^2} \quad (4.5)$$

where

\bar{Q}_p is the mass production rate of ions = $2.656 \times 10^4 \text{ s}^{-1} \text{ kg}$
 \bar{r} is the distance to the center of the comet [m]
 $\bar{\lambda}_c$ is the ionization distance = $3.03 \times 10^5 \text{ m}$,
 that controls the width of the source

for which we make the following normalizations (in order to make \bar{A} dimensionless):

$$Q_p = \frac{\bar{Q}_p}{\rho_0 v_0 R_0^2} \quad r = \frac{\bar{r}}{R_0} \quad \lambda_c = \frac{\bar{\lambda}_c}{R_0}$$

We modify the continuity equation (2.1) to

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = A(r) \quad (4.6)$$

The cometary ions will also cause a modification to the momentum equation (2.1) so that we have

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v} - \mathbf{B} \mathbf{B}) = -\nabla \left(p + \frac{B^2}{2} \right) + \rho \mathbf{g} - A(r) \mathbf{v} \quad (4.7)$$

in the same region.

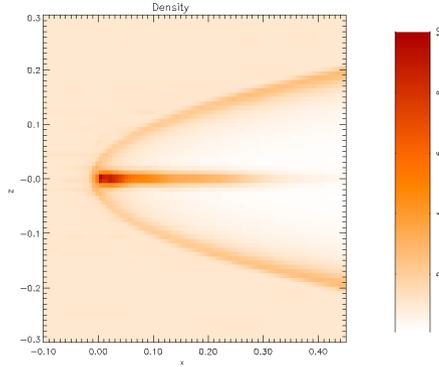


Fig. 1. Density at steadystate for our 3D simulation of a comet with parameter settings $n_{sw} = 15 \text{ cm}^{-3}$, $\mathbf{v}_{sw} = v_x = 500 \text{ km/s}$, $\mathbf{B}_{sw} = B_z = 6 \text{ nT}$ and source terms according to (4.5)-(4.7). This steady state occurred ≈ 1 second after we add the comet as a source term, and the figure shows the density distribution 1.4 seconds after the encounter. We here show the x - z plane and thus have the solar wind coming in from the left and its magnetic field in the upward direction. The minimal refinement level was set to 2 and the maximal to 5, resulting in a peak of 1,024,000 in the number of cells

We use the same normalization as in [6], using the earth radius $R_E = 6.37 \times 10^6 \text{ m}$ as R_0 . Other quantities are scaled with typical solar wind values: $\rho_0 = 2.51 \times 10^{-20} \text{ m}^{-3}\text{kg}$ (corresponding to a number density of the solar wind n_{sw} of $15 \times 10^6 \text{ m}^{-3}$) and $v_0 = 500 \text{ km/s}$. The magnetic field of the solar wind is set so that $\bar{\mathbf{B}} = \bar{B}_z = 6 \text{ nT}$ and we use a solar wind temperature $\bar{T}_{sw} = 2 \times 10^5 \text{ K}$. We use an adiabatic index $\gamma = 5/3$ and include no external forces ($\mathbf{g} = 0$).

5 Results for the Test Case

We here present results for the test case of the solar wind interaction with a comet, as described in Section 4. We first present the simulation results in the physical context and compare them with expected results and previously done simulations. Thereafter we present how the code scales with problem size and when we increase the number of used processors.

From a physical point of view the simulations give result in agreement with Ogino et al. [6]. Figure 1 shows the result of a typical simulation in 3D. The characteristic weak bow shock is seen in all simulations. As seen in Figure 2, the solar wind is decelerated by the cometary ions and the magnetic field is draped and obtains peak values near the comet nucleus.

The code seems to scale well for our test problem. The execution time grows almost linearly with the problem size as seen in Figure 3. There is also profit to be made by using more processors for a larger problem as seen to the right in Figure 3 - the deviation from the linear speedup appears later when we increase the problem size. As expected the speedup is sub-linear as we increase the number of processors due to increased communication time.

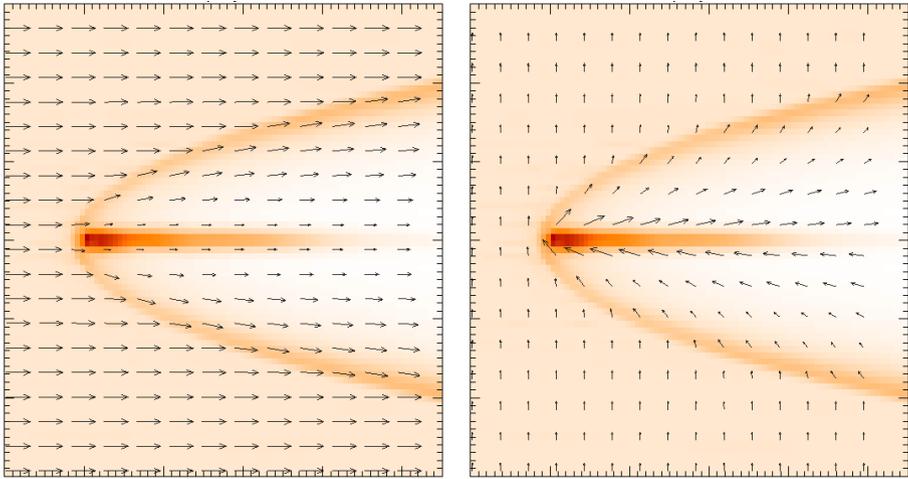


Fig. 2. Velocity (left) and magnetic (right) field vectors for our 3D simulation of a comet for comparison with Ogino et al. [6]. This corresponds to their Figure 3. Parameter values, length and density scale are the same as in Figure 1

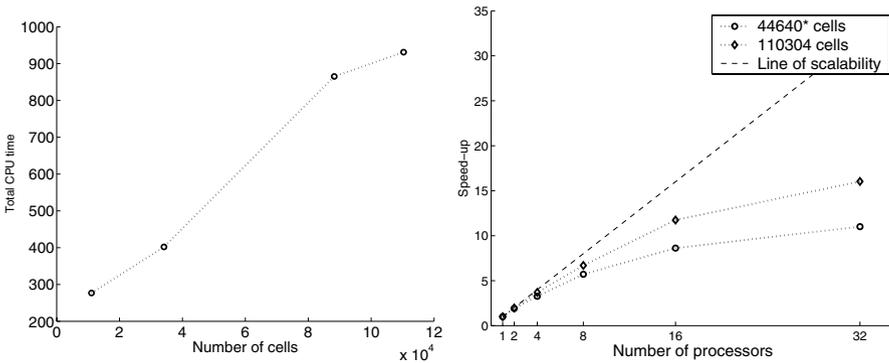


Fig. 3. Left: Execution time as function of the number of cells in the problem. To the right we show the speedup as a function of the number of processors for two problem sizes

The mesh refinement works fine for our test problem; refinements are made in critical regions and only there. The automatic refinement is thus much satisfactory. The code is also capable of handling the discontinuities of our test problem. In our simulations we start with a flowing solar wind and then insert the production of cometary ions in the middle of our simulation box. This does not cause any problem in spite of the much lower velocity of these ions. Both the automatic refinement mechanism and discontinuities are shown in Figure 4 where we show a simulation at $t=0.2, 0.4$ and 0.6 seconds. This simulation was done in 2D so that we could have large refinements differences and still keep simulation time reasonable.

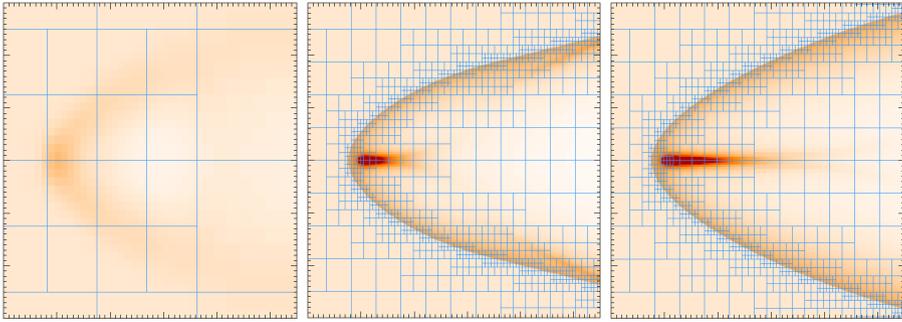


Fig. 4. The start of a simulation in 2D. At $t=0$ we add the source term A to the MHD equations as explained in Section 4. This shows the density at $t=0.2s$ (left), $t=0.4s$ (middle) and $t=0.6s$. Note how the mesh is refined (only) where needed. The minimal refinement level was set to 1 and the maximal to 8, resulting in a peak number of 128,000 cells

The simulations were performed at the Seth Linux Cluster at the High Performance Computing Center North (HPC2N) [13]. The cluster has AMD Athlon MP2000+ processors in 120 dual nodes with each CPU running at 1.667 GHz. The network has 667 Mbytes/s bandwidth and an application level latency of $1.46 \mu s$. The network connects the nodes in a 3-dimensional torus organized as a $4 \times 5 \times 6$ grid. Each node is also equipped with fast Ethernet. For all our simulations the limitation is CPU speed (and communication as we increase the number of nodes), as we do not use the full 1 GB memory of each node.

6 Conclusions and Discussion

In this work we have shown the feasibility of using a general MHD solver for the simulation of the interaction between the solar wind and solar system objects. The open source Flash code needed small changes to specify the initial conditions, boundary conditions and source terms to solve our selected model - the three-dimensional interaction between the solar wind and a spherical symmetric source region, a simplified model of a comet.

We assured the correctness of our results by comparison with previously published ones. The simulations showed that the code scales well, both as a function of problem size, and as a function of number of computing nodes.

Thus, the Flash code seems to have the possibility of being used as a general tool for the investigation of the interaction between solar system objects and the solar wind.

Some areas of future investigation is the possibility of solving multi fluid MHD problems (e.g., H^+ and O^+), test particle trajectory integration in the MHD solution field, and the way to handle planetary bodies.

Acknowledgments

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, USA.

References

1. Alliances Center for Astrophysical Thermonuclear Flashes (ASCI), <http://flash.uchicago.edu>
2. M.G. Kivelson and C.T. Russell (editors). *Introduction to Space Physics*, University Press, Cambridge, UK, 1995 ISBN 0 521 45104 4
3. E.R. Priest and A.W. Hood (editors). *Advances in Solar System Magnetohydrodynamics*, University Press, Cambridge, UK, 1991 ISBN 0 521 40325 1
4. H. Matsumoto and Y. Omura (editors). *Computer Space Plasma Physics*, Terra Scientific Publishing Company, Tokyo, Japan, 1993 ISBN 4 88704 111 X
5. K. Kabin, K.C. Hansen, T.I. Gombosi, M.R. Combi, T.J. Linde, D.L. DeZeeuw, C.P.T. Groth, K.G. Powell and A.F. Nagy. Global MHD simulations of space plasma environments: heliosphere, comets, magnetospheres of planets and satellites *Astrophysics and Space Science*, 274:407-421, 2000
6. T. Ogino, R.J. Walker and M. Ashour-Abdalla. A Three-Dimensional MHD Simulation of the Interaction of the Solar Wind With Comet Halley *Journal of Geophysical Research*, vol. 93, NO. A9, pages 9568-9576, September 1988
7. T.I. Gombosi, D.L. De Zeeuw, R.M. Häberli and K.G. Powell. Three-dimensional multiscale MHD model of cometary plasma environments *Journal of Geophysical Research*, vol. 101, no. A7, pages 15233-15232, July 1996
8. T. I Gombosi. *Physics of the Space Environment*, University Press, Cambridge, UK, 1998 ISBN 0 521 59264 X
9. The Message Passing Interface (MPI), <http://www-unix.mcs.anl.gov/mpi/>
10. Parallel Adaptive Mesh Refinement (PARAMESH), http://ct.gsfc.nasa.gov/paramesh/Users_manual/amr.html
11. P. MacNeice, K.M. Olson, C. Mobarry, R. deFainchtein and C. Packer. PARAMESH: A parallel adaptive mesh refinement community toolkit *Computer Physics Communications*, vol. 126, pages 330-354, 2000
12. H.U. Schmidt and R. Wegmann. Plasma flow and magnetic fields in comets *Comets: Gases, Ices, Grains and Plasma* edited by L.L. Wilkening. University of Arizona Press, Tuscon, USA
13. High Performance Computing Center North, Umeå, Sweden, <http://www.hpc2n.umu.se>