

# A New Spacecraft Plasma Interactions Simulation Software, PicUp3D/SPIS

J. Forest<sup>1,2</sup>, L. Eliasson<sup>2</sup>, A. Hilgers<sup>3</sup>

<sup>1</sup> IRF-K, Sweden, <sup>2</sup> CNRS-UVSQ/CETP, France, <sup>3</sup> ESA/TOS-EMA, The Netherlands

## Abstract

The PicUp3D/Spis code is a prototype of a new simulation software dedicated to accurate quantitative simulation of spacecraft plasma interactions and developed in the context of a European scientific and industrial network. It is based on a 3D Particle-In-Cell (PIC) approach. The use of unstructured schemes in the spacecraft geometrical description and of the Capacitance-Matrix method in the field derivation has been successfully experimented in order to model 3D realistic geometries. The code is organized as an open and versatile object-oriented library and is fully written in JAVA. It is believed that this code will meet the requirements of the scientific community in term of geometric and physics accuracy especially for instrument calibration and observation analysis.

## 1 Introduction

Spacecraft platforms and systems interact sometimes strongly with the space plasma environment. Due to their mobility, charged particles (electrons, ions) hit external surfaces and, in synergy with the ionisation effect by solar UV or high energy particles, lead to the accumulation of a net electrical charge on spacecraft surfaces. This accumulation of charge builds up an electrostatic potential difference between the satellite and the plasma, able to reach many kV negative or a few tens volts positive. Modifications of the spacecraft electrostatic environment and spacecraft-charging may affect the performances of technological systems and must be carefully considered especially for high accuracy measurement sensors. Induced electrostatic field and other perturbations of the plasma by the spacecraft (wake, shadowing) may severely affect low energy particles and plasma measurements. Potential difference greater than or of the order of the ambient plasma temperature observed

on scientific spacecraft as diverse as Freja, Interbal, Polar, FAST, Cluster, offer very good examples of such problems.

The IPICSS project, for Investigation of Plasma Induced Charging of Satellite Systems, was initiated at IRF-K to analyse the effects of plasma interaction on modern satellite systems, including scientific instruments. The project is a collaboration between IRF-K, ESA/TOS-EMA, CNRS/CETP and CNES, in the framework of the SPINE network [1]. The objectives were an in-depth analysis of current knowledge and of the methods used in spacecraft-plasma simulation and to propose new technical solutions whenever appropriate[2] [3] [4]. The main output is a first version of the kernel of an open library of simulation tools, called SPIS for Spacecraft Plasma Interaction System, which takes advantage of modern computing technology, and is based on an Object-Oriented Approach (OOA). It is based exclusively on existing open source software in order to facilitate a development in community. A first example is PicUp3D/SPIS, which is an experimental electrostatic multi-species 3D Particle-In-Cell (PIC) code written in JAVA. The results of the first validation tests of this code are presented and discussed in this paper.

## 2 Models

### 2.1 PicUp3D/SPIS code

In the PIC method, described in details in e.g [5] and [6], the Vlasov-Poisson system is approximated by the integration of the motion of a large set of macro-particles (typically between one and ten million) interacting with the electric and magnetic fields, self-consistently computed as function of the charge densities and currents. In the electrostatic approach, the electric field is obtained via the electric potential derived from Poisson equation. The system of equations to solve is:

$$m \frac{dv}{dt} = \frac{q}{m} (\mathbf{E} + v \wedge \mathbf{B}) \quad (1)$$

$$\frac{dx}{dt} = v \quad (2)$$

$$\mathbf{E} = -\nabla\phi \quad (3)$$

$$\nabla^2\phi = -\frac{\rho}{\epsilon_0} \quad (4)$$

The Poisson equation is solved on a regular 3D Cartesian grid using finite differences and the Gauss-Seidel/Chebyshev method. The motion of particles is integrated with a classic leap-frog scheme and an additional constant magnetic field can be added. The PIC method is based on a kinetic description of the plasma and it is in principle possible to model the complete structure of the sheath, including wakes and magnetic shadowing effects. However, there are also many specific difficulties, e.g. the treatment of particles at the boundaries of the simulation box and the presence of a collecting object inside the computational domain.

## 2.2 Spacecraft geometry

A strong effort was devoted to provide the most versatile and detailed description of the spacecraft geometry. The issue of the geometrical description of the spacecraft is directly related to the particle/surface intersection problem and the resolution of Poisson equation. Forseen applications of the code to instrument calibration requires the possibility of a very detailed description of the geometry of the detector as well as the whole spacecraft structure. A description of the spacecraft geometry using triangular unstructured meshes was introduced in PicUp3D. This allows the possibility to adapt the mesh size and to increase the resolution for the most relevant elements only. A large majority of existing 3D modellers use similar approaches and may be easily linked to the SPIS system via an input format similar to the VRML format, used e.g. in virtual reality.

The interception of particles is done using a ray-tracing technique. Because the trajectories are not straight lines, the intersection test should be done for each time step and the corresponding subroutine should be optimised carefully. The interception algorithm is constituted of two steps. First, it is determined whether the particle will cross the plane defined by the triangle or not. Second, it is determined whether the intersection point (if any) is inside or outside the triangle. To

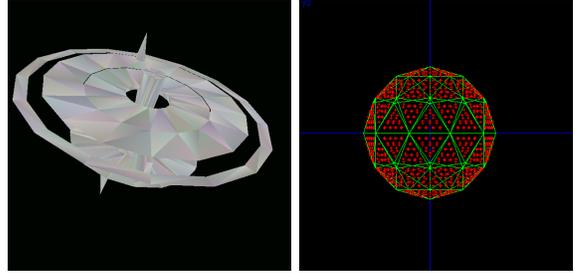


Figure 1: Examples of objects modelled with PicUp3D, viewed using the Java/Java3D based 3D visualisation features of SPIS. Left side shows a first model of the Swedish/German Freja satellite, modelled with 859 elements of surface. The right side shows standard spherical target used for the Langmuir probes simulations. Red crosses correspond to the virtual electrodes used in the charge equivalent method.

this end, barycentric coordinates  $(\alpha, \beta, \gamma)$  of the intersection point  $I$  are introduced. This is equivalent to express the coordinates of  $I$  in the coordinates system  $R_{tri}$  defined by two sides and the normal of the triangle. With this formalism, the conditions of interception are reduced only to the passage from the main coordinates system of the  $R_{tri}$  system and three tests on scalars. The current collection is then determined on each element of surface counting reaching particles by unit of time. It is more meaning full to average the total collected charge over a few plasma periods.

## 2.3 Potential derivation

The difficulty is to take into account the inner boundary constituted by the spacecraft surface. Two approaches are currently investigated. In the first one, the spacecraft structure is projected to the Nearest Grid Point (NGP). Corresponding grid cells are switched inactive for the Poisson equation solving subroutine and become a boundary condition. In an alternative approach, the boundary condition in potential is converted into an equivalent condition in charge, with virtual electrodes (see red crosses on Fig.1) uniformly distributed on the spacecraft surface and charged. The spacecraft potential is let floating and Poisson equation is computed everywhere on the grid, taking into account these charges. The spacecraft potential is then just read as an output. The main advantage of this approach is a better final resolution in the geometrical description of the spacecraft. Electrodes can be positioned exactly

on the spacecraft surface as illustrated in Fig.1 and independently on the grid structure. In case of complex objects, the difficulty is to define the right value of each charge. This problem can be solved by the Capacitance Matrix method [7] [8] or introducing an internal equivalent circuit à la NASCAP [3], which will define dynamically the distribution of charges assuming a set of coupling factors. However, these modules are not fully implemented yet in PicUp3D/SPIS and the charge equivalence method is currently used only for objects with a symmetry of revolution.

## 2.4 Boundary conditions and particle injection

The injection of particles at the boundary of the grid is critical for the consistency of the simulation. The difficulty is to define an open boundary corresponding to the unperturbed plasma. In PicUp3D, Singh’s tank method [9] is used.

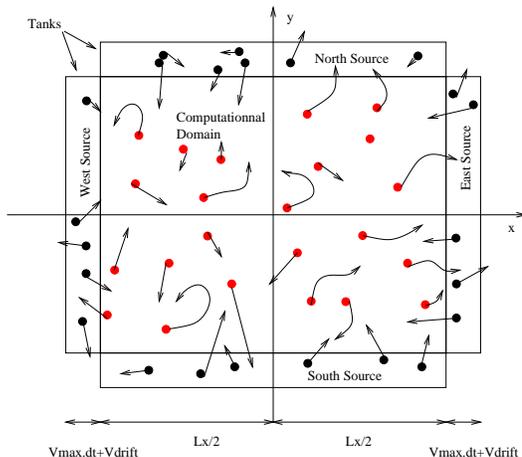


Figure 2: Geometry and principle of the sources of particles. “Tanks” are located just outside the computational grid and uniformly filled with particles. Particles scatter freely toward the grid.

As illustrated in Fig.2, a “tank” of particles is built around the computational grid and uniformly filled of particles at the nominal density. No field is applied in the tank. Particles can freely scatter into the computational box. Conversely, particles inside the computational grid may scatter into the tank. For optimisation reasons, the total number of macro-particles defined in memory is fixed. Only one part of them are switched active and used for the simulation. This defines a relative loading ratio with respect to the total number

of macro-particles. In order to always have a stock of free particles, a nominal ratio of active particles is set to 80 % of the total number of macro-particles. In Fig.3, the loading ratio is shown as a function of the time for various configurations. The dashed line corresponds to the nominal loading ratio. The dotted line corresponds to the free scattering case, without any field present in the computational box, and the solid line to the case where the electric field is consistently computed inside the grid such as in the PIC model. Especially in this case, one can observe a significant depletion of the density, due to a loss of a part of the electron population. The difference between the two cases proves that this loss is probably due to the discontinuity of the electric field at the grid limit and not only to the higher mobility of electrons as interpreted in [9]. Longer simulation runs have shown that the density is globally constant with fluctuations around 10%. In all cases, no disturbances or density barriers have been observed during various tests, including at very low temperatures ( $T_e = 0.01 eV$ ). The residual random fluctuation in potential remains of the order of the thermal energy (see Fig.4).

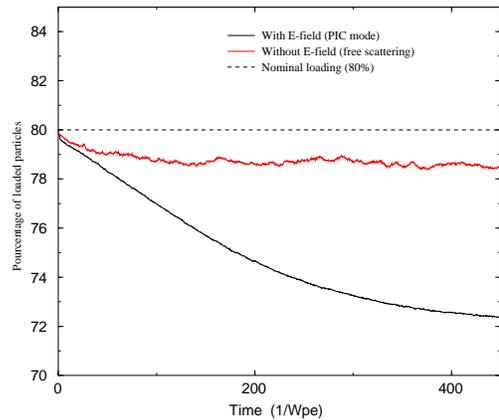


Figure 3: Relative loading ratio: dotted and solid curves correspond respectively to the free scattering (without field) and self-consistent fields (PIC) modes. The dashed line corresponds to the nominal number of active particles.

## 3 Tests and first applications

### 3.1 Langmuir probe

So far, the capabilities of PicUp3D have been mainly tested via the simulation of spherical Lang-

muir probes in various plasma conditions. The sheath structure and potential-current characteristics have been especially studied in detail. Fig.4 shows the iso-levels of the potential map corresponding to a spherical probe immersed in a dense plasma ( $\lambda_D \leq 0.02 R_{probe}$ ) and averaged over  $50 1/\omega_{pe}$ . A “cut” was performed along the y-axis (see Fig.5). The potential plotted in logarithmic scale and compared with [10]. A very good agreement is found between results obtained with PicUp3D and the results of [10], inside the sheath structure ( $R < 3cm$ ). Outside the sheath, the residual thermal noise remains just in reasons of a too short averaging time ( $T_{average} < 1/\omega_{pi}$ ).

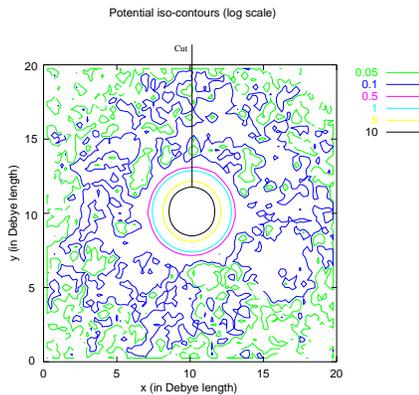


Figure 4: Electric potential iso-contours for a spherical Langmuir probe biased to  $25K_B T_e/e$  in a Maxwellian plasma at rest (averaged over 50 plasma periods).

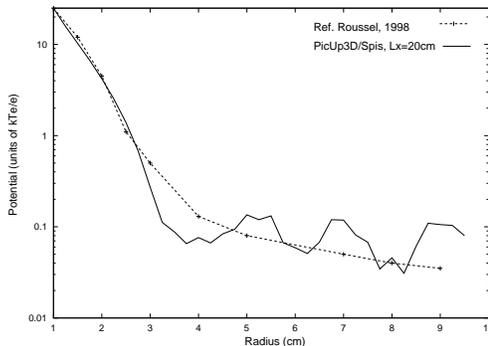


Figure 5: Langmuir probe in a Maxwellian plasma at rest: This figure corresponds to a cut along the line called “cut” in Fig.4.

In Fig.6, the potential-current characteristic of the same sphere is displayed. Two methods to solve the electrostatic potential are compared each other. The value of the floating potential of the

probe can be derived from the characteristics. Assuming an electron-proton Maxwellian plasma at rest, the floating potential  $\phi_f$  of a spherical probe is given by:

$$e \left( \frac{q_e \phi_f}{k_B T_e} \right) = \sqrt{\frac{m_e}{m_i}} \sqrt{\frac{T_e}{T_i}} \left( 1 + \frac{q_e \phi_f}{k_B T_e} \right) \quad (5)$$

For a mass ratio of  $\frac{m_i}{m_e} = 400$  and  $T_i \simeq T_e$ , this leads to:

$$\phi_f \simeq -2.5 \frac{k_B T_e}{e} \quad (6)$$

Let  $\tilde{\phi}_f = \frac{e \phi_f}{k_B T}$  be the dimensionless potential. A good agreement between the numerical and theoretical results, with a floating potential  $\tilde{\phi}_f \simeq 3.0 \pm 1.0$  given by the simulation. Transient dynamics of the sheath structure was observed at each step in potential. The relaxation time depends on the mass ratio electrons to ions and the amplitude of potential variations of the collecting object. For each step, a simulation duration of a few tens of plasma periods seems a minimum to have a stabilised system. In practice, this value remains much less than the typical charging time of the complete satellite, but may be significant in case of microscopic and fast mechanisms (e.g. discharges) or for scientific observations with a high time resolution.

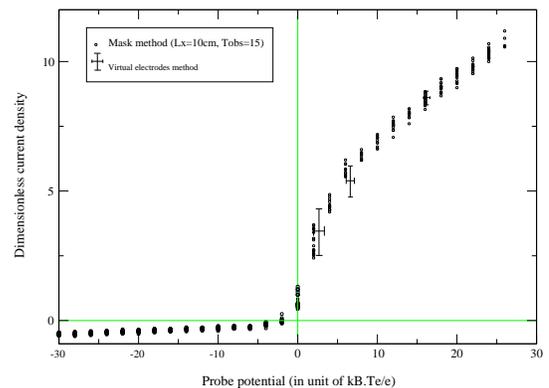


Figure 6: Potential-current characteristic for a spherical Langmuir probe. Dots correspond to the “mask technique” with an observation time of  $15 1/\omega_{pe}$ . Crosses with error bars correspond to potential/current characteristic obtained with the charge equivalent method. One can observe a good agreement between the two techniques.

The equivalent electrodes method was used in a second simulation. The sphere was uniformly charged with a given elementary charge, the potential left floating and read as an output. The corresponding values are designed by the crosses in Fig.8. Error bars are due to the fluctuation in potential and current. We can observe a very good agreement between both techniques which are complementary in practice. In fine, the charge equivalent method presents a better resolution in the geometrical description of the spacecraft and seems more adapted to floating objects, where the potential is dynamically integrated as function of the collected currents.

### 3.2 Computing benchmarks

JAVA is a semi-interpreted language, where source codes are compiled into bytes-codes, which are themselves interpreted using a specific runtime environment called Java-Virtual-Machine (JVM). Due to this interpreted approach, JAVA had the reputation to be prohibitively slow and costly in memory to perform large and realistic numerical simulations such as the PIC method. Tests performed with PicUp3D in this study show however that considerable improvements have been done with the last generation of JVM, especially with the introduction of Just-In-Time compilers. In Fig.7, a comparison of the computing performances of PicUp3D using various JVM under Linux is shown. This corresponds to the technological evolution observed in less than one year. A factor of five between the first JVMs and the most efficient one has been observed.

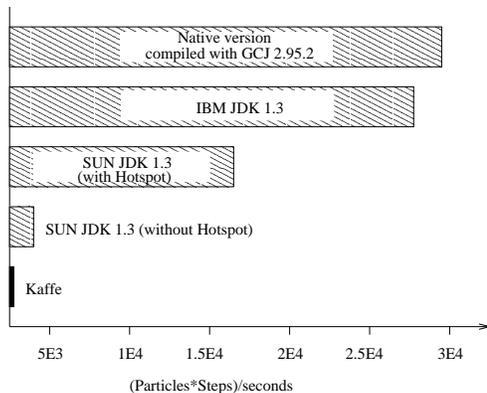


Figure 7: Benchmarks of the PIC loop of PicUp3D with various JVM.

To compare with the performances of native codes, the kernel of PicUp3D has been compiled

with the last version of the GNU compiler (GCJ/GCC 2.95) able to produce native executable codes from JAVA sources. Surprisingly, the speed improvement is not significant, probably due to the fact that the current version of GCJ is still under development and not fully optimised, especially regarding the array access. JVMs have been considerably improved, the level of optimisation of compilers remains however very low and many improvements may be expected here in a near future. We have to underline, that the final performances are considerably dependent on the JVM/system couple. Under Linux, the best performances have been obtained with the IBM JVM 1.3. PicUp3D/Spis is not compared with other codes written in C or FORTRAN yet. It is likely that codes written in JAVA are still slower. However, realistic runs are already accessible on modern personal computers. As in illustration, a run of  $1000 1/\omega_{pe}$  with  $10^6$  particles on a  $64 \times 64 \times 64$  grid can be performed in about 24 hours on a Pentium III at 700 MHz.

## 4 Conclusion

The development of the PicUp3D code started in December 1999. This computer code is still under the testing and validation phase and more detailed and quantitative tests are needed. The preliminary results confirm the capabilities of the PIC method to model complicated sheath structures and S/C-plasma interaction mechanisms on large and realistic systems. The study has shown that an Object Oriented Approach, e.g. based on the JAVA language, is today relevant to large numerical simulation tasks. The advantage of this approach is that it facilitates a community based development. In this perspective, it is planned in the near future to make the PicUp3D code publicly accessible in the framework of an international collaborative network. The new models and tools to be further developed in the frame of this network will focus on high geometrical resolution and will therefore be very suitable to scientific applications such as instrument calibration and data analysis.

## Acknowledgments

Part of this work was performed under European Space Agency contract No 13590/99/NL/MV. The authors are grateful to J.-P. Catani and D. Payan (CNES) and to J.-J. Berthelie and H. de Feraudy (CETP) for useful comments and support for the continuation of this activity.

## References

- [1] ESA/TOS-EMA. *SPINE Network*. <http://www.estec.esa.nl/wmwww/Spine/>, March 2000.
- [2] Eriksson A.I. et al. *Modelling of Freja Observations by Spacecraft Charging Codes*. ESA contract 11974/96/NL/JG(SG), SPEE-WP120-TN, June 1998.
- [3] Mandell M.J. et al. *NASCAP Programmer's Reference Manual*. NASA Contract No. NAS3-22826, SSS-R-84-6638, La Jolla, USA, 1984.
- [4] Cooke D. et al. *POLAR User's Manual*. Geophysics Laboratory Hanscom Air Force Base, MA 01731, Contract F19628-82-C-0081, October 1989.
- [5] Birdsall C.K. Langdon A.B. *Plasma Physics via Computer Simulation*. McGraw-Hill Book Company, NY, 1985.
- [6] Eastwood JW. Hockney RW. *Computer simulation using particles*. Adam Hilger, NY, 1988.
- [7] Hockney R.W. *POT4, A FACR Algorithm for Arbitrary Regions*. Computers, fast elliptic solvers and applications, Proceeding, 1977.
- [8] Martin E. Dale. *A Generalized-Capacity-Matrix Technique for Computing Aerodynamic Flows*. Computers and Fluids, Vol2, pp. 79-97, Pergamon Press, 1974.
- [9] Singh N. et al. *Three-dimensional numerical simulation of current collection by a probe in a magnetized plasma*. Geophysical Research Letters, Vol.21, No.9, p.833-836, May 1994.
- [10] Roussel J-F. *Spacecraft Plasma Environment and Contamination Simulation Code : Description and First Test*. Journal of Spacecraft and Rockets, Vol.35, No.2, p.205-211, March-April 1998.