FEEDBACK ON THE PICUP3D EXPERIENCE AND THE OPEN SOURCE STRATEGY APPLIED TO A SPACECRAFT-PLASMA INTERACTION SIMULATION CODE

Julien Forest
Swedish Institute of Space Physics (IRF-K), Kiruna, Sweden
Planetary and Terrestrial Environment Study Centre (CNRS-UVSQ/CETP)
E-mail: julien@irf.se

Alain Hilgers
Space Environments and Effects Section, (ESA-ESTEC/TOS-EES)

Abstract

PicUp3D is an electrostatic multi-species 3D Particle-In-Cell (PIC) code dedicated to the modeling of the electrostatic sheath of spacecraft. Fully written in JAVA and able to model 3D realistic geometries, PicUp3D is designed to perform fine modeling for various purposes, especially scientific instruments calibration, and active device environment analysis such as ion thrusters. The development of PicUp3D was initiated in the context of the IPICSS project, for Investigation of Plasma Induced Charging of Satellite Systems, in partnership between IRF-K, ESA, CNRS/CETP and CNES, in the framework of the SPINE network and an ESA Academic Research Programme. The development time was followed by a long validation phase during which PicUp3D was intensively tested and used for mission supports at ESA. These application cases have confirmed the ability of PicUp3D models to simulate correctly modern spacecraft-plasma interactions problems. To induce a strong synergy with the scientific and the industrial communities, PicUp3D was developed with an open source software approach, such that the source code is freely accessible and the models can be tailored, validated or deeply modified by advanced users. PicUp3D has been released (http://www.spis.org) for the first time in December 2002 under the GPL license and is now freely available. Since then, feedback have been collected and contributed to improve the code by integration of new models, such as a multi-grid field description, or technologies, such as a script command layer. In parallel, this interaction with the user community have helped to identify key issues that should be taken into account to support a community based development of the next generation spacecraft-plasma interaction simulation library (SPIS). The current status of PicUp3D is presented and analyzed with respect to the initial objectives of the project.

Introduction

Spacecrafts interact with the surrounding plasma via many processes. By virtue of the thermal motion of plasma particles and via electrostatic interactions, exposed surfaces collect charged particles (electrons and ions), leading to the accumulation of a net electrostatic charge and a corresponding potential distribution. The potential is also modified by the ionization of surfaces by the UV and soft X-rays radiations and high energetic particle radiation. Charging may have many consequences for spacecraft systems including the scientific instruments. For example, strong charging events have been observed on geostationary orbit and in auroral regions and may generate sparks. Even moderate potential difference between the spacecraft and the plasma is a concern since it may prevent ambient particles to reach plasma sensors. Other spacecraft-plasma interaction phenomena may create disturbances of a few volts on spacecraft surfaces or in the space surrounding the spacecraft.

For instance ion wakes are observed on low Earth orbit where spacecraft speed is meso-thermal (i.e., higher than ion averaged thermal velocity but lower than electron averaged thermal velocity). The corresponding highly non-isotropic perturbations are difficult to model via analytical models. The need for in-flight plasma instrument calibration is now better perceived and detailed numerical simulations may clearly help to this [1,5].

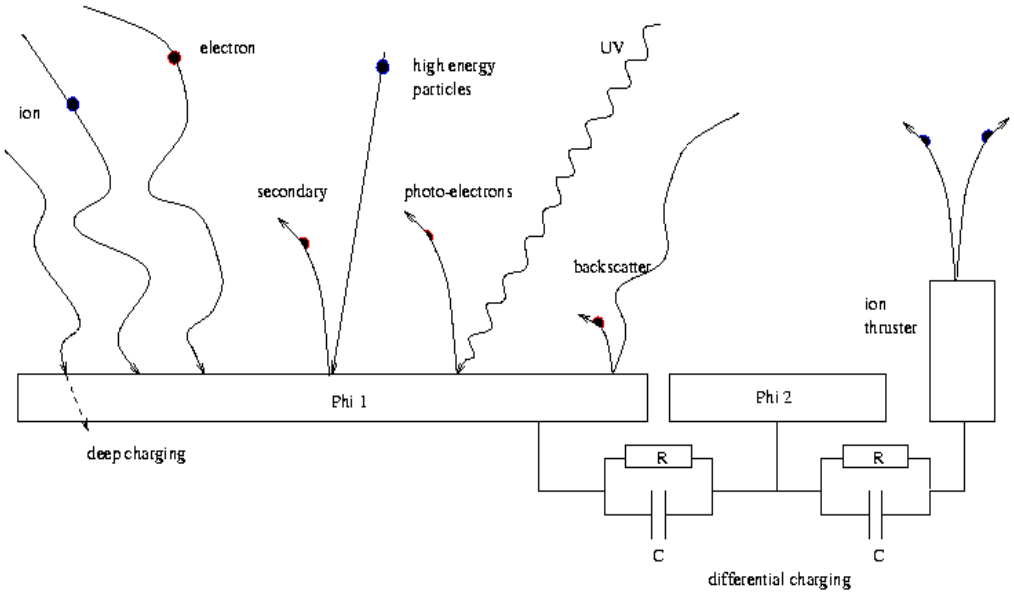

Figure 1. Sketch of surface processes which contribute to spacecraft charging.

On the technological side, new needs appear in Europe with the more common use of electrical propulsion (e.g. ESA missions GOCE and SMART-1). The interaction of the ion plume with the external environment (including the vehicle itself) constitutes a large and open domain of investigation. The new generation of high power solar arrays working with higher bus voltage presents new issues with risks of arcing and loss of power due to current leakage. These phenomena were not well modeled by the previous generation of charging codes and are still very challenging from the numerical point of view (e.g., complex processes involving various fields of physics, large range of scales) [cf 4, 8].

Historical Context and Approach

These new challenging requirements have emphasized the need for new and advanced simulation codes, based on more detailed models and using modern computing technologies. Because new problems are in many cases multi-disciplinary, a very good visibility of the implemented models and the access to the code constitute a critical issue. One solution is the Open Source Software (OSS) approach, according to which software are released with their source codes. This approach may appear as a clear breaking with respect to the pre-existing culture in the domain of simulation codes, especially in an industrial context. However, this approach has existed in the academic world for more than thirty years and may also be an answer to some constraints in the industrial environment. This is somehow the approach that was adopted with the NASA-USAF charging code, NASCAP, except for its geographical restriction [7]. Because simulation software are complex and costly to develop and validate, an in-house team of developers remains generally too small in terms of manpower and knowledge and cannot develop a complete software by its own excepted in very big

companies. By sharing the development costs on one hand and gathering the complementary knowledge on the other one, the open-source approach should reduce the development, validation and maintenance cost. Last, the open source approach should also decrease the dependence on only one specialist and break the effect “one-code one-specialist” and by this extends the frequency of use and the lifetime of the code.

The IPICSS project, for Investigation of Plasma Induced Charging of Satellite Systems, was initiated at IRF-K in the framework a PhD scholarship (ESA Academic Research Programme) in summer 1999, in partnership between IRF-K, ESA, CNRS/CETP and CNES (academic grant). It was performed in the context of the SPINE network (see http://www.spis.org/spine/). The objectives of this project were to analyze in depth the new requirements for spacecraft-plasma interaction modeling especially from the scientific community (e.g. for instrument calibration, interpretation of observations) and the industry (e.g., for analyzing active device and electric thrusters) and to develop a simulation tools to meet such requirements.

One of the main outputs of IPICSS was the development of a new spacecraft-plasma interaction simulation software, called PicUp3D, able to model 3D realistic geometry for a broad range of time varying space plasma environment [2,3]. This new code includes advanced numerical kinetic models and takes advantage of new computing technology such as the Object Oriented Approach and modern hardware (PC clusters). PicUp3D was designed since the beginning in the perspective of an open source strategy and was officially released under the GPL license in December 2002. Since it has been made publicly available its uses has significantly progressed within both communities, scientific and industrial.

The PicUp3D Code, Models and Technologies

Picup3D is based on a 3D electrostatic full multi-species Particles-In-Cell (PIC) model. The electrostatic potential is solved on a 3D Cartesian structured grid, using an optimized Gauss-Seidel iterative solver. An additional non-uniform static magnetic field can be taken into account. The motion of macro-particle is performed using a classic leap-frog scheme. The spacecraft is modeled using an unstructured meshing, allowing the possibility to optimize the surface meshing and facilitates the link with existing 3D modelers and Delaunay surface meshers. A very large range of plasma conditions may be taken into account, including non-Maxwellian sources (drifting or non-isotropic plasma). Currents of particles may be derived for each surface element.

PicUp3D is fully written in JAVA language and according to an Object Oriented Approach (OOA). The real reasons of this choice are directly related to the perspective of a public release and a future community based development. JAVA is a very popular language today, especially among the young generation of computer engineers. It benefits from the strong dynamics of the Web community and a large number of development tools and advanced libraries are freely available on the Net. The OOA allows a high modularity of the code and future extension by simple integration of new computing components (classes, methods). Actually, the OOA seemed especially appropriate for later insertion of new piece of codes by several developers. The rigorous and simple syntax of JAVA leads to more homogeneous, and then easier to maintain, codes than the C++ language.

Tests have been performed which demonstrate that JAVA programs are able to perform large numerical simulations with acceptable.

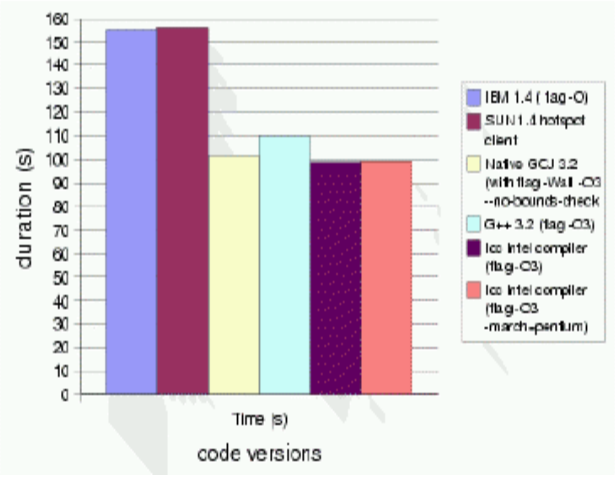

Figure 2. Comparative benchmarks between JAVA and C++ versions of the Poisson's solver implemented in PicUp3D.

Comparative benchmarks (see Figure 2) shows that for critical modules as the Poisson's equation solver, the JAVA byte-coded version of the code is about twice slower than its native equivalent written in C++. The corresponding run duration increase is nevertheless smaller than the time usually spent by the user to modified or adjust code parameters before a run for standard problems (typically a few days). The performance of the latest version of the GNU compiler, GCJ, able to compile JAVA source codes into native executable codes, has also been explored. In this case, the JAVA code performances become almost equivalent to the C++ ones and is even surprisingly sometimes faster than C++. This seems very promising and opens new possibilities for very large simulations with JAVA codes.

PicUp3D and its main library are completed by a set of pre and post-processing tools. A script layer, based on Jython, is under implementation in order to facilitate the access to the low level numerical methods via modules that can be called and manipulated directly by the user in a console mode.

Picup3D has been validated with several Langmuir's probe test case simulations (potential map, current-tension characteristics). The results were compared against Langmuir probe theory and outputs from other validated codes. A specific effort was dedicated to evaluate the accuracy of the code. Tests have confirmed that an accuracy better than 10% on potential and current on Langmuir's probe could be achieved [2].

Problems requiring runs on grids larger than 150^3 mesh nodes and with more than 6 millions of particles have been successfully modeled in support to some ESA missions [4,9,10].

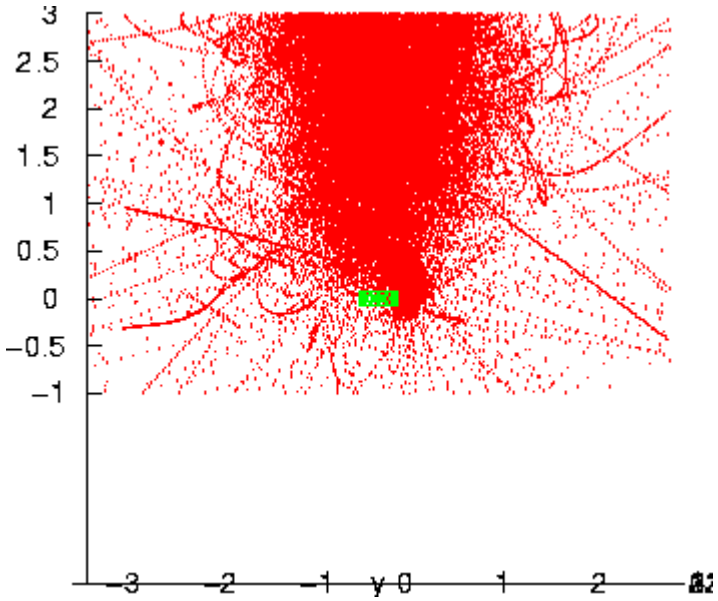

Figure 3. Photo-electron cloud attracted by the ion emitter (ASPOC) plume on the ESA CLUSTER-II mission [10].

Open Source Strategy

It has appeared from previous experiences, that a public release of an open source software should respect a set of rules to be successful. First of all, the released code should satisfy well-defined need. Generally, OSS are initiated by a reduced team of initial developers (kernel-team hereafter) until the development reaches a "critical mass" when the code is able to perform a minimum set of the required functionalities. The date of release in the development time-line is very variable depending on the objective of the software and its related communities. A project may have interest to be released the earliest possible to benefit from the community dynamics and inputs. Conversely, in some area it is better not to release the software before enough validation has been performed to create confidence on the applicability of the code. Another criteria of maturity is the characteristic time to install and learn how-to-use the code for a new user. If the learning curve is prohibitively long, the software will be rejected by the community. Presumably, the need to stimulate co-development from community members not belonging to the kernel team set even more constrained on the required maturity and structure of the code before release. These aspects are critical for the dynamics of a community-based development.

A Gant's diagram corresponding to the evolution of the PicUp3D project is shown on Figure 4. There are two well identified phases in the project. The first one, before the first release, corresponds to the design and the development of the simulation kernel. This was done by a very small team, typically one to three developers at a given time with the contribution of a few experts for validation which were performed in short iterative sequences most of the time on the complete chain of the simulation but lead sometimes to drastic revisions of some modules.

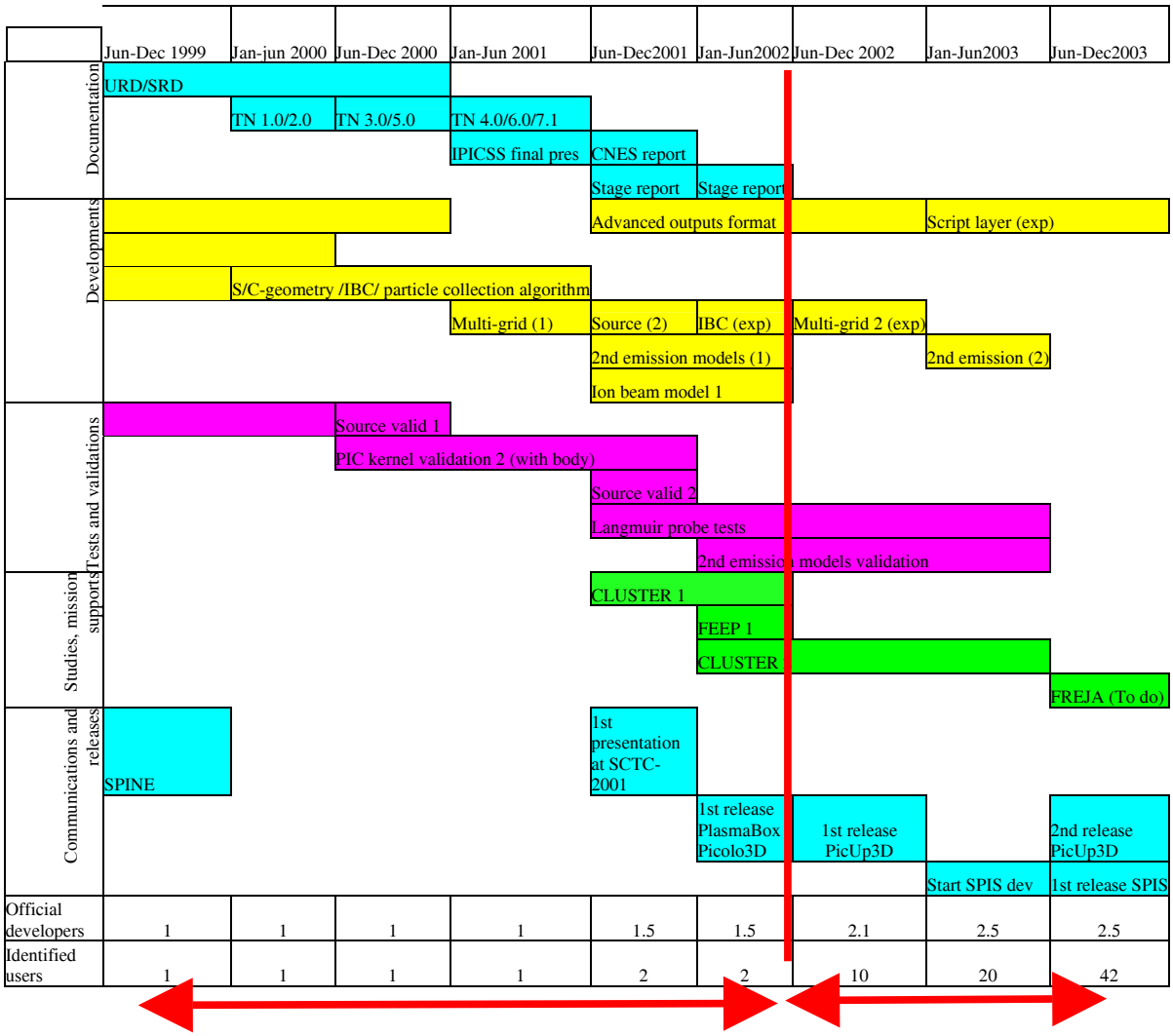

Figure 4. Gant's diagram corresponding to the evolution of the PicUp3D project.

At the date of the first release, the simulation kernel of PicUp3D was fully operational and had been already applied for problem solving in an operational context. The second phase, is based on a broader community contribution and has already seen improvement in terms of new models (new photo-electrons emission, multi-grid modules) or specialization of the code. During this phase, the initial effort of the kernel team has to be progressively re-oriented from the pure development toward the management of the community, the integration and the validation of the external contributions.

The last critical point is the choice of the license and the ownership of the intellectual rights. At the opposite of what is generally believed, the open source software approach does not imply that authors loose their intellectual property rights which should be defined carefully in relation with an appropriate license. A community-based development is based on an agreement and the choice of the license is essential. For PicUp3D distribution, it has been decided to use the well known GPL license which forces any software derived from PicUp3D to be distributed under the same license. The advantage of this license is that it tends to increase the resources available to the community of users and developers without discrimination.

PicUp3D Community Description and Feedback

PicUp3D and its derived versions (PlasmaBox and PicOlo3D) have been directly downloaded from the official web site by about 65 different users in less than 6 months. This is more than the community of users identified within the Spacecraft Plasma Interaction Network in Europe (SPINE) which consist of a few tens of members (www.spis.org/spine).

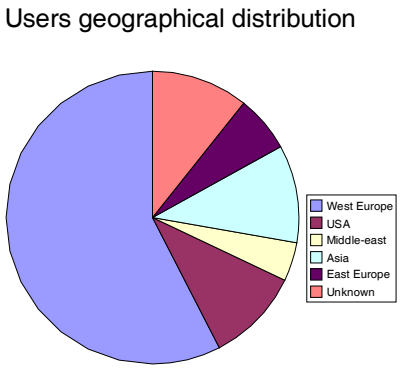

Figure 5. Geographical distribution of downloads of PicUp3D

The geographical distribution of locations from where code downloads have been performed is shown on Figure 5. It shows that PicUp3D was mainly downloaded from western European countries (mainly EU) with (about 57 % of the downloads), followed by USA (~11%). Eastern European and Asian countries together represent 21 % of the downloads. The high representation of western European countries is probably due in part to the active promotion made within the SPINE network. About 11 % of the downloads were performed without that any official identification has been provided. A majority of registered users expressed the wish to be kept informed of the future evolutions of the project.

The number of downloads according to academic organisations versus commercial companies is shown on Figure 6. It shows an almost even distribution with a slight advantage for the academic world (47% against 40%). This even distribution constitutes a real success regarding the initial goals of the project. It is difficult, however, to know how many downloads correspond to real applications and whether the applications have been extended beyond the spacecraft-plasma interaction domain.

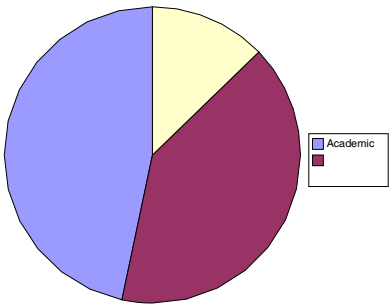

Figure 6. Academic versus commercial companies distribution of downloads

The feedback from the community is mainly organised through the SPINE meetings where discussion on the relevant issues can take place. The feedback takes several aspects ranging from provision of piece of new code to bug reports. Significant contributions have been done on multi-grid algorithm, secondary emission improvement. Ongoing analyses are being performed to take into account wire structures, wakes, magnetic field effects and to shorten computation time to reach equilibrium. It must be noted, however, that most of these activities are mainly funded via similar sources as the kernel team. A substantial feedback is also provided by the analysis and critics of the code by the users, regarding the models, the computing implementation and the user interface. PicUp3D, like many research codes, is still a prototype and is sometimes difficult to install and use. This is also partly due to the lack of official documentation (not released yet). Critics regarding the computing aspect may take various forms, from the simple bug report to a complete proposition of alternative design. The research of the inter-compatibility with another project is a very efficient way of contribution. It requires re-engineering of both codes, allows the quick introductions of new functionalities. The interaction users-developers on the basis of the first released version constitutes a very good source of new user requirements and may lead to define functional specifications of a new version or a new code. This is especially true in case of iterative development, or "development in spiral", typical of community based software development.

Future of PicUp3D

The observations made in the previous section show that the public release of a scientific code may actually help to increase the dynamics of a scientific community and that PicUp3D has played and may continue to play an efficient role in the development of the SPINE community. Initially designed as a simple technological demonstrator, PicUp3D's life lasted already longer than expected. It is now adopted by a large community which may even extend outside the boundaries of the SPINE network. Its development phase as an exclusively in-house code is now finished. A new phase is opening, based on the participation of the user and developers community. It is expected that the main effort in the SPINE network will be progressively dedicated to the future SPIS library of codes [8]. As a result PicUp3D will keep a simple structure and its distribution and installation will continue to be very simple. It is foreseen that it will be used as an educational tool by academia. In the very near future, it will assure a progressive transition of the community toward SPIS. In this perspective, PicUp3D has been imbedded in the first release of the SPIS-UI framework of the SPIS project [8]. The pre and post-processing tools will provide extended CAD capabilities and better surface meshing to PicUp3D (see Figure 7). Eventually, based on the already proven capability of PicUp3D it will also certainly be used as a benchmark for new codes.

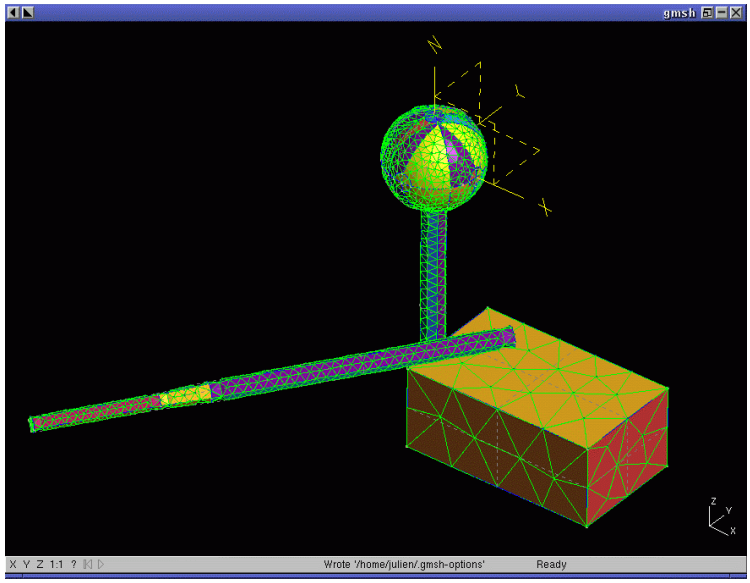

Figure 7. Example of set of Langmuir's probes modeled with the SPIS-UI framework and imported into PicUp3D's format.

Conclusion

Although it is still too early to draw a definitive conclusion about the new community based development phase of the PicUp3D project we can however notice a number of successful achievements made with this project. Tests and applications performed in an operational environment have confirmed the relevance of first principle based numerical models like 3D PIC simulations to address contemporary modelling requirements in spacecraft-plasma interactions. Tests and benchmarks have confirmed the capability of JAVA to perform large simulations with a reasonable computing cost. The JAVA language associated to a simple object oriented design seems also very well adapted to a community based development and future extensions of the code. The open source approach allows a very large range of possible extensions and adaptations of the code around the core numerical kernel. The most successful test is presumably the popularity of the code gained in less than one year within the SPINE community and world-wide as evidenced by the number of references made to it in this conference.

Acknowledgments

This study has been performed in the frame of the Spacecraft Plasma Interaction Network in Europe (SPINE) activities (cf. www.spis.org). We acknowledge useful discussions at the 5th SPINE workshop organised by J.-F. Roussel. The work of J. Forest has been performed in the frame of a PhD study co-supervised by IRF-K and CETP partly funded under ESA contract No 13590/99/NL/MV) and a CNES grant.

References

1. Eriksson, A.I., Wedin, L., Wahlund, J-E, and Holback, B.: "Analysis of Freja Charging Events: Modelling of Freja Observations by Spacecraft Charging Codes", IRF Scientific report 252, January 1999.

2. Forest J., Eliasson, L., Hilgers, A.: "A new spacecraft plasma simulation software, PicUp3D/SPIS", 7th Spacecraft Charging Technology Conference, Proceedings pp.515-520, ESA/SP-476, ISBN No 92-9092-745-3, ESA-ESTEC, Noordwijk, The Netherlands, 23-27 April 2001.

3. Forest J., A. Hilgers, B. Thiebault, L. Eliasson, J.-J. Berthelier, and H. de Feraudy, "A new open source spacecraft plasma interaction simulation software library, PicUp3D", to be submitted.

4. Forest, J., A. Hilgers, B. Thiebault and A. Popescu, "Modelling of the plasma environment of micro FEEP thrusters using PicUp3D PIC code", in Proceedings of the 8th Spacecraft Charging Technology Conf., Huntsville, USA, Oct. 20-24, 2003.

5. Hilgers, A. and B. Thiebault, "A review of spacecraft effects on plasma measurements", in Proceedings of the 8th Spacecraft Charging Technology Conference, Huntsville, AL, USA, October 20-24, 2003.

6. Mandell M. J., et al., "NASCAP programmer's reference manual", NASA contract No NAS3-22826, SSS-R-84-6638, La Jolla USA, 1984.

7. Roussel, J.-F., F. Rogier, M. Lemoine, D. Volpert., G. Rousseau, G. Sookahet, P. Séng, and A. Hilgers, Design of a New Modular Spacecraft Plasma Interaction Modeling Software (SPIS)," in Proceedings of the 8th Spacecraft Charging Technology Conference, Huntsville, AL, USA, October 20-24, 2003.

8. Thiébault, B., Hilgers, A., Forest, J., Escoubet, P., Fehringer, M. Laakso, H.: Modelling of the photo-electron sheath around an active mgnetospheric spacecraft, in Proceedings of the 8th Spacecraft Charging Technology Conference, Huntsville, AL, USA, October 20-24, 2003.

9. Thiébault, B., Hilgers, A., Sasot, E., Forest, J., Génot, V., Escoubet, P.: "Investigation of electrostatic potential barrier near an electron-emitting body", in Proceedings of the 8th Spacecraft Charging Technology Conference, Huntsville, AL, USA, October 20-24, 2003.