



Computer simulation of the potential distribution inside the plasma chamber of negative ion source

Juliusz Sielanko^{a*}, Marcin Turek^a, A. Tanga^b

^a*Institute of Physics, Maria Curie Skłodowska University,
pl. M. Curie-Skłodowskiej 1, 20-031 Lublin, Poland*

^b*Max Planck Institute of Plasma Physics, Garching, Germany*

Abstract

In the paper the computational method and results of simulation of potential distribution inside the plasma region of the negative ion source are presented. The code uses the well known PIC method [1] for the local charge density determination and finite differences method for the Poisson equation [2]. For simplification, the plasma model does not take into account the interaction between the plasma particles. The results of calculations are compared with the experimental data obtained for the RF negative ion source [3].

1. Introduction

The negative ion is considered a valuable particle, which may be used successfully in various fields as a substitute particle for a positive ion. The difference between negative and positive ions appears in charge polarity, energy of ionisation and its energy polarity related to neutralization. The energy related to the ionization (electron affinity) is low (about 1eV) and the polarity energy is negative (absorbable when neutralized). Such features reduce surface charge up voltage during ion implantation of insulated materials and increase a secondary ion production during sputtering the target. The negative ions beam is useful also in the neutral beam injection (NBI) systems in fusion research because of their high neutralisation efficiency. This article describes the computer code, which simulates the phenomena inside the plasma chamber of negative ion source, and the results of calculations of potential distribution as well as concentration distribution of positive and negative ions. For the simulation purpose we use the geometry of calculation area, which is related to only part of the area of real RF negative ion source, and which may be extrapolated over the whole design due to periodic boundary conditions used in the modeling method.

* Corresponding author: *e-mail address*: jzs@tytan.umcs.lublin.pl

2. Computational method

2.1. Particle In Cell (PIC) model

In the PIC method a very large number of real particles is represented by a much smaller, numerically tractable number of “macro particles” [1]. A typical computer simulation follows the trajectories of 10^5 - 10^6 computational particles, i.e. “macro particles” which represent a region that contains 10^3 - 10^9 real particles. At each time step, the “macro particles” are advanced by the single particle equation of motion. The charge density and electromagnetic fields are calculated with the assumption that all particles in a region move with the computational particle. In other words, the “macro particle” advances as a single particle, but is assigned multiple charges when the charge density and fields are calculated.

In the PIC method the space is divided into small regions creating a spatial mesh. The method weighs particles to a spatial grid using a particle shape factor to obtain charge on the grid. The charge assignment process is the heart of a PIC simulation. It may be the simplest *nearest-grid-points (NGP)* method based on counting the number of particles between grid points and assigning the corresponding density to the cell centre. In other models, “macro particles” are given a *mathematical finite size* with a diameter roughly that of the cell to generate density functions with smoother variations. Depending on its position, a fraction of a particle may be assigned to a cell. This procedure is called the *cloud in cell* CiC method. In the presented code, the PIC model may be optionally chosen as the NGP or CiC [4] method. The calculations are performed in the 3D Cartesian coordinate system.

2.2. The plasma particles trajectory calculations

As mentioned above the PIC model uses the “macro particles” to simulate the motion of a much higher number of real particles. The motion of “macro particle” is treated as the single particle in a given electromagnetic field. To obtain the E field value in every grid point we use the charge density calculated by the PIC method and solve the Poisson equation:

$$\nabla^2 V = -\frac{\rho}{\varepsilon} \quad (1)$$

where ρ is the charge density and ε is the dielectric constant.

In the program the successive *overrelaxation* [2], finite-differences method to solve the equation (1) was used. Once the potentials V are known, the electric field on the grid can be obtained from

$$E = -\nabla V \quad (2)$$

The force on the particles (ion beam particles) is obtained by interpolating E at each position using some weighting. In our program the simple, linear

interpolation was used. The particle velocity is updated using the Lorentz force equation

$$\frac{d\vec{v}_i}{dt} = \frac{q_i}{m_i} [\vec{E} + \vec{v}_i \times \vec{B}] \quad (3)$$

In the calculations presented here no external magnetic field was used, we also treated the nonrelativistic beam, so we may assume that the B field created due to the beam current is small and can be neglected. The differential equation (3) was solved numerically using the 4-th order Runge-Kutta procedure. As a result we obtained the next position \mathbf{x}_i , \mathbf{y}_i , \mathbf{z}_i and velocity \mathbf{v}_x , \mathbf{v}_y , \mathbf{v}_z of each particle (“macro particle”) of the ion beam.

The scheme of the calculation of the trajectories of ion beam, in the E field produced by the electrodes and by the charge in the beam itself, at each time step looks as follows:

1. At the time t the charge density $\rho(i,j,k)$ (primary ions and secondary electrons and ions) is evaluated at the mesh points by assigning the charge of the “macro particles” according to their position.
2. Next, the Poisson equation (1) is solved using *overrelaxation* method to find the electric field.
3. The E field is used to advance the vectors \mathbf{x} and \mathbf{v} for the particles to $(t+\Delta t)$ using the Runge-Kutta difference scheme to solve the equation (3).
4. Procedure continues until the beam advances to the desired final state.

2.3. CPU Time consumption optimization – omega calculations

As has been said before, electric field in the plasma chamber is found by solving the Poisson equation using the *successive overrelaxation* method (SOR) [2,5,6]. This is the most time-consuming part of the code, therefore the optimal choice of ω (overrelaxation parameter) is crucial for the code efficiency. We performed some tests for the empty plasma chamber to find how fast the SOR method converges for different ω values. The $100 \times 100 \times 100$ Cartesian grid was used. It is worth mentioning that the optimal ω choice rule given e.g. in [2] does not work for large grids. The rule claims that optimal ω_o in the case of Cartesian grid is the smaller of the two roots of quadratic equation

$$a^2 \omega^2 - 4\omega + 1 = 0 \quad (4)$$

where

$$a = \cos\left(\frac{\pi}{N_x}\right) + \cos\left(\frac{\pi}{N_y}\right) + \cos\left(\frac{\pi}{N_z}\right) \quad (5)$$

and N_x , N_y and N_z are the numbers of the grid points in x, y and z direction, respectively. It is easy to see that for large grids a is close to 3, and there are no real solutions of eq. (5).

Tests were done for two kinds of boundary conditions: a) potential is established on walls of the chamber (boundary conditions I), b) gradient of potential vanishes on edges of the grid, and computational area may be treated as a part of a bigger plasma chamber (boundary conditions II).

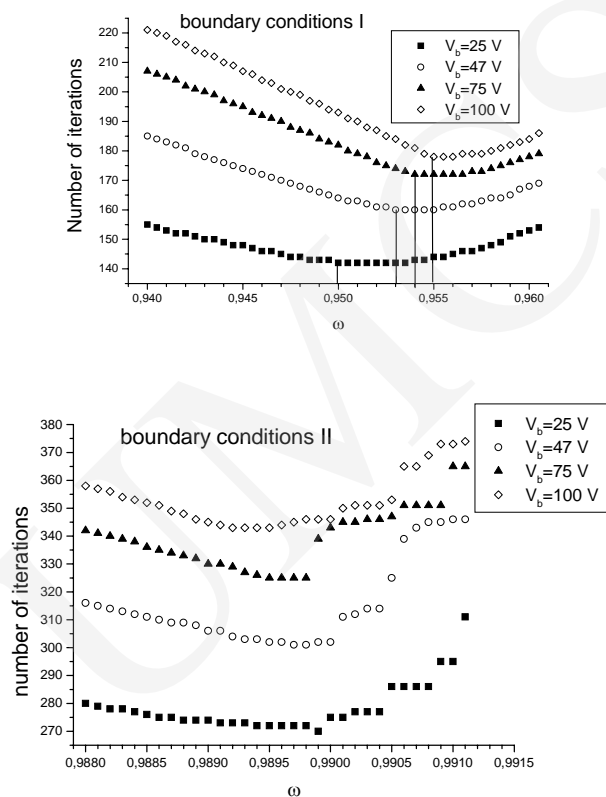


Fig. 1. Dependence of rate of SOR method convergence on overrelaxation parameter ω for different values of V_b voltage and two kinds of boundary conditions

The results are presented in Figure 1. In the case of boundary conditions I optimal ω increases with the increasing V_b voltage, whilst in the case of boundary conditions II it decreases. Note, that the number of iterations after which the SOR method converges is nearly two times greater in the case of boundary conditions II. Especially in this case the choice of ω value that differs from optimal value e.g. by 0.001 may lead to more than 10% increase of iterations number, which means hours of CPU time in non-test runs.

Fig. 2 shows, that ω should be also optimized with respect to accuracy parameter denoted δV that determines whether iteration stops or continues. In

other words, iteration stops when the condition $|V_{i,j,k}^{old} - V_{i,j,k}^{new}| \leq \delta V$ is fulfilled in every point of the grid. Optimal ω moves towards larger values with the increasing accuracy (i.e. with decreasing δV), and minimum becomes more pronounced. Due to this fact the choice of ω must be careful in order to avoid unnecessary CPU time consumption.

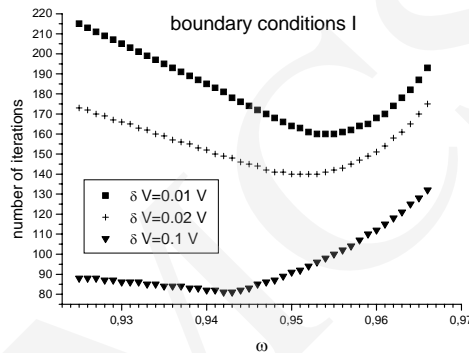


Fig. 2. Dependence of the convergence rate of the SOR method on the accuracy parameter δV . Results for $V_b=47$ V

2.4. Influence of the number of pseudoparticle and their charge representation

In the PIC method one computational “macro-particle” represents a large number (usually 10^3 - 10^9) of real particles of the same kind. We wanted to test the influence of this number, and what follows, the charge of “macro-particle” on the results of charge density and potential computations. A large number (denoted N_p) of charged macro-particles is put into a rectangular box of sizes 6.8 cm/2cm/2cm. These particles, corresponding to protons and electrons and carrying the charge $+Q_p$ and $-Q_p$ are randomly distributed inside the box. Their movement is neglected. It is easy to see that in the case of perfect random distribution of the infinite number of particles, charge density and potential should be equal to zero in the whole volume. Unfortunately, in computer simulation N_p is restricted by the size of accessible memory, so it seems advisable to estimate limitations of that method.

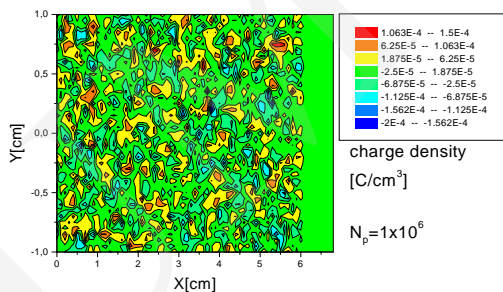
Tests were performed for several values of N_p . Every time Q_p was chosen to keep the total charge of particles of one kind constant. In other words, $N_p Q_p$ is invariant, as seen in Table 1.

Table 1. The values of N_p and Q_p used in test simulations

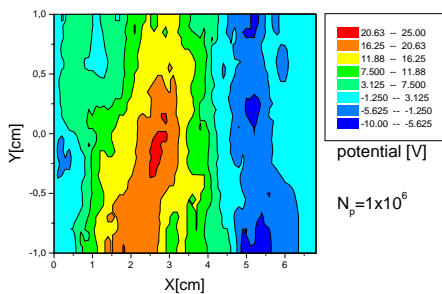
N_p	Q_p [C]
100000	$6.95 \cdot 10^{-15}$
500000	$1.39 \cdot 10^{-15}$
1000000	$6.95 \cdot 10^{-16}$
2000000	$3.47 \cdot 10^{-16}$

In Figs. 4 and 5 the results of calculations of charge density and equivalent potential distribution for the randomly distributed protons and electrons (with their number ratio 50/50%) are presented. In this case the data are related to the central area of the grid system.

As was mentioned before in all tests presented below the particles after generating their position in the chamber (using Monte-Carlo method) are not movable.



a)



b)

Fig. 4. Charge density and potential distribution obtained for 10^6 macro-particles distributed homogeneously in a rectangular box

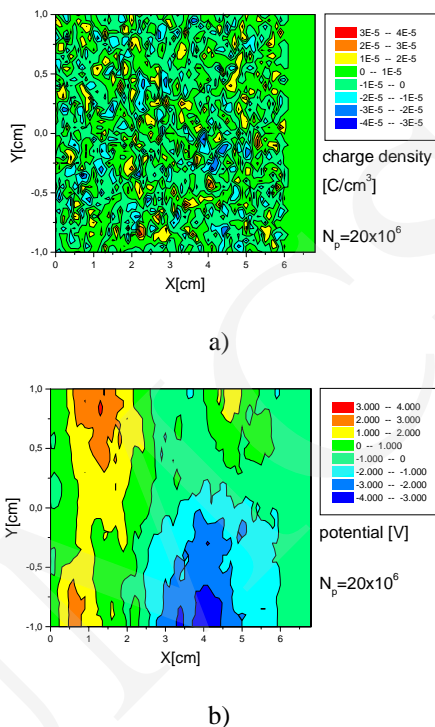


Fig. 5. The same as on the previous picture but for $2 \cdot 10^7$ macro-particles

Figs. 4 and 5 show that an increasing number of macro-particles makes the calculated charge density distribution smoother and more homogeneous. The range of charge density oscillations change from $4.5 \cdot 10^{-4} \text{ C/cm}^3$ for $N_p = 10^6$ to $8.0 \cdot 10^{-5} \text{ C/cm}^3$ for $N_p = 2 \cdot 10^7$.

Charge density becomes more balanced with respect to zero value, which can be seen even better in the next picture.

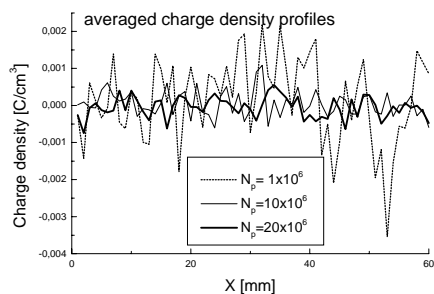


Fig. 6. Charge density profiles along the chamber axis. The charge density is averaged on thin (6 mm) slices

Charge density averaged over the 6mm wide slices of the grid (perpendicular to the X axis) is presented for three N_p values. One can see that amplitude of spatial oscillations rapidly decreases for larger N_p . The more inhomogeneous charge density distribution is the more bumpy and oscillating the potential is calculated via the above mentioned SOR method. For a relatively small number of macro-particles $N_p=10^6$ the potential has “ridge” of 25V height, and “valley” of 10V depth next to each other. Better statistics, i.e. larger N_p smoothes these undesired oscillation, e.g. for $N_p=2 \cdot 10^7$ potential becomes more flat and centered around zero value. Hence, using the large number of macro-particles and smaller charge per one macro-particle gives better, more physical results, and the cost of it is longer computation time and greater memory requirements.

2.5. Comparison of the results of calculations for NGP and CiC models

It was mentioned in section b) that there is a variety of PiC models depending on how macro-particle shape function is chosen. Putting it in other words: PiC models differ from each other mainly by the algorithm describing assignment of macro-particle charge to grid points. In this section we compare the results of charge density and potential distributions in the plasma chamber obtained using three different PiC schemes. The first and the simplest one is the *nearest grid point* scheme (NGP) when whole macro-particle charge is assigned to only one grid point. This PiC algorithm is very fast, and easy to implement. A little bit more complicated is the first order CiC method (denoted here by CiC1). According to this method, particle charge is distributed among eight grid points that are vortices of the grid cell particle. Description of the weighting coefficient can be found elsewhere. We also tested the second order CiC method (CiC2) very similar to that used in Numerical Tokamak Project and described in [7]. This method distributes with sophisticated weighting coefficients charge of macro-particle among the nearest 27 grid points.

All these methods were tested in a way described in the previous section, which means N_p macro-particles were put into a cylindrical plasma chamber with homogeneous distribution, and their movement was frozen. Differences in the results given by these PiC models are most visible in the case of small N_p .

Looking at the averaged charge density profiles (Figs. 8 and 9) one can see that both CiC methods give smoother distributions than this obtained by NGP. Even simple “smearing” of particle reduces oscillations in charge density profiles. Potential distributions given by CiC methods are also smoother. However, despite the fact that CiC2 method is far more advanced than CiC1, it gives roughly the same distributions in the plasma chamber we considered, so using this method does not seem to be purposeful in this particular case.

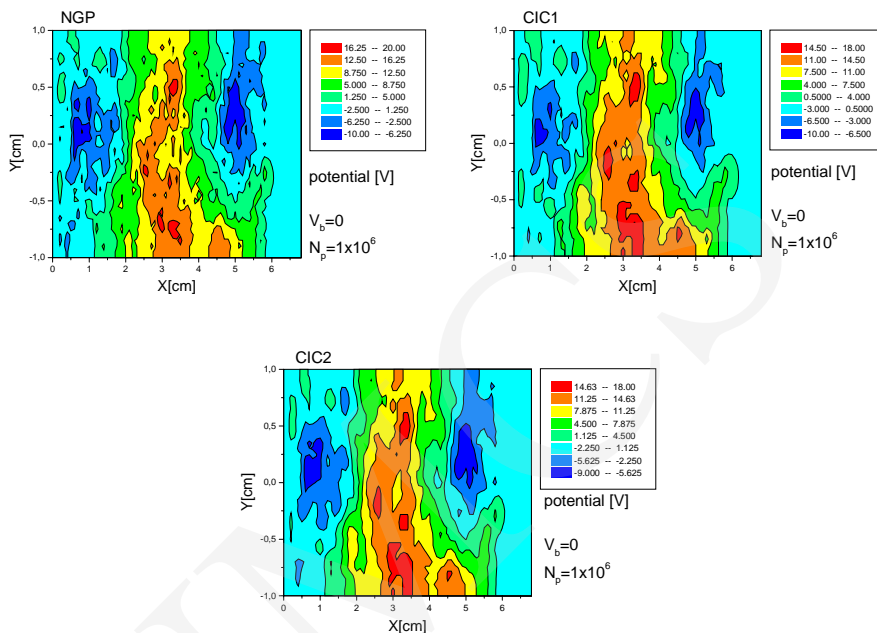


Fig. 7. The comparison of potential distributions obtained from the three PiC models: NGP, CiC1 and CiC2

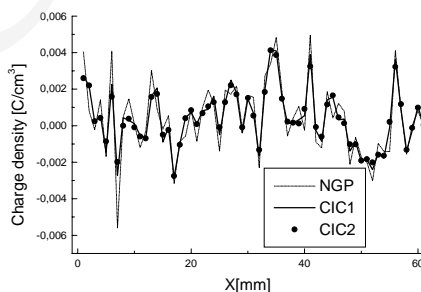


Fig. 8. Charge density profiles in the three PiC models obtained for 10^6 macro-particles

Density profiles obtained for $N_p=10^7$ presented in Fig. 9 behave in a very similar way.

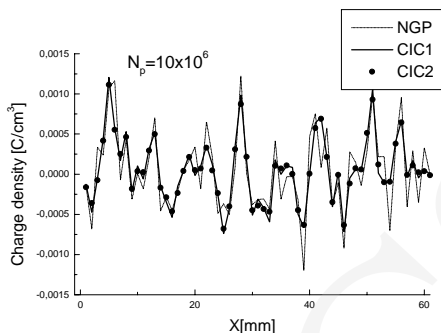


Fig. 9. The same as in the previous picture but for 10^7 macro-particles

3. Results of the calculation of potential distribution and concentration distribution of ions inside the plasma chamber

As for the test purpose, we use the similar geometry of simulation area (x,y cross section for $z=0$ as a middle of the chamber is presented in Fig. 10). The calculations were made for the plasma consisting of 3 kinds of particles – H^+ , H and e^- with the starting uniformly distributed concentration as $5E13 / 1E13 / 4E13 \text{ 1/m}^3$ respectively. The 10^6 pseudoparticles are uniformly distributed inside the vessel at the time $t=0$. There is no interaction between the plasma particles taken into account, it means that *time of life* for each particle is equal to infinity. The particles are moved until they hit the plasma chamber wall, and next they “disappeared” in the calculations. Positive grid bias is equal to 47V and no voltage is applied to the extraction electrode and no magnetic field inside the plasma chamber.

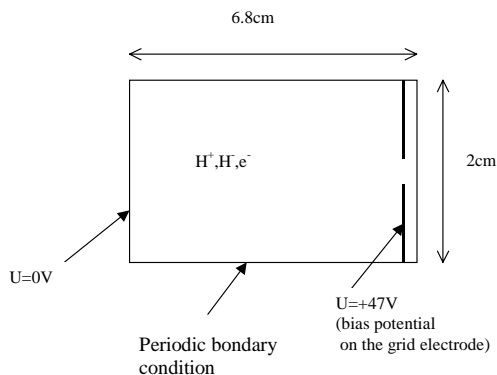


Fig. 10. X,Y cross section for $Z=0$ of ion source chamber

In Fig. 11a,b the space concentration distribution of H^+ and H is presented.

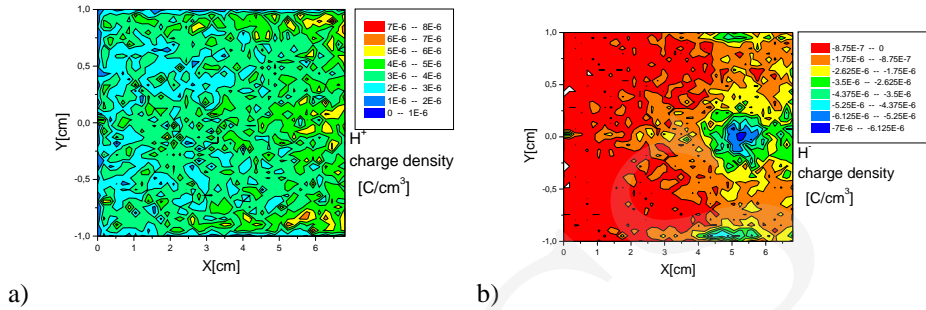


Fig. 11. Concentration distribution of (a) - H^+ and (b) - H^- ions inside the plasma chamber of the ion source

As can be seen the positive grid bias increased the H^- density near the grid about by a factor of 2 to 3. This result is consistent with the experimental results obtained by the M.Bandyopadhyay and A.Tanga [3]. As mentioned before we also calculated the potential distribution inside the plasma chamber of an ion source. The results are presented in Fig 12. In Fig. 12 the plasma potential gradient over the distance of 6cm from the grid edge is shown.

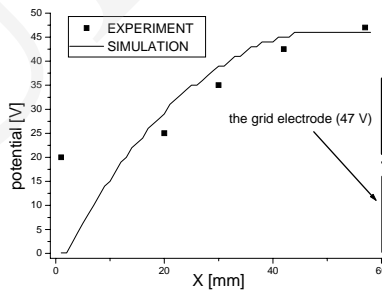


Fig. 12. Plasma potential near the grid – the line represents the calculated values, point – the data from experimental measurements [3]. The grid electrode is biased with +47V

The agreement between the experimental data and the potential gradient calculated by the code is good in the distance up to nearly 5cm from the grid. The discrepancy between the results for the last point (plane at 6cm from a grid) can come from the boundary condition chosen for this plane (constant potential equal 0). In reality there is a plasma flow from the RF chamber.

4. Conclusions

In the paper we present the code which can model the phenomena inside the plasma of the ion source chamber. The test of the code shows that for a given

geometry of grid system the number of 1×10^6 pseudoparticles used for the simulation purpose is enough to obtain quite good results for the concentration distribution of particles inside the chamber. The NGP method in PIC model for the charge density distribution gave practically satisfying results in comparison with the better, but much more time consuming CIC method.

The positive grid bias increased the H^- density near the grid by about a factor of 2 to 3. This result is consistent with the experimental results [3]. The calculated potential gradient of plasma is in good agreement in the near grid edge (up to 5cm from the electrode).

References

- [1] Birdsall C.K., Langdon A.B., 1985, *Plasma Physics Via Computer Simulation*. McGraw-Hill, New York.
- [2] Slavicek E., 1991, *Technika Obliczeniowa dla Chemików*, WNT Warszawa, in Polish.
- [3] Bandyopadhyay M., Tanga A., *not published*.
- [4] Potter D., *Computational Physics*, Wiley, London, (1977).
- [5] Hockney R., Eastwood J., *Computer Simulation Using Particles*. Mir, Moskwa, (1987), in Russian.
- [6] Press W.H., Flannery B.P., Teukolsky S.A., Vetterling W.T., *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, (1986).
- [7] Decyk V.K., Karmesin S.R., de Boer A., Liewer P.C., *Optimization of Particle-in-Cell Codes on RISC Processors*, *Computers in Physics*, 10 (1996) 290.