International Journal of Mass Spectrometry 303 (2011) 42–50

Contents lists available at ScienceDirect

International Journal of Mass Spectrometry

journal homepage: www.elsevier.com/locate/ijms

3DCylPIC—A 3D particle-in-cell code in cylindrical coordinates for space charge simulations of ion trap and ion transport devices

R. Ringle

National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI, USA

ARTICLE INFO

Article history:
Received 28 October 2010
Received in revised form
15 December 2010
Accepted 21 December 2010
Available online 31 December 2010

Keywords:
Space charge
Ion transport
Particle in cell
RF carpet

ABSTRACT

A 3D particle-in-cell code in cylindrical coordinates, that can be used to characterize space charge effects in various ion traps and ion transport devices where cylindrical boundary conditions are more appropriate, is presented. By solving Poisson's equation on a cylindrical grid at every time step, the presence of an arbitrary amount of charged particles is handled in a completely natural way. Hard sphere scattering and statistical diffusion models can be used to include effects of interaction with a neutral buffer gas. An overview of the code is discussed, and its first application to study RF carpet devices is presented.

1. Introduction

One of the least intuitive phenomena in ion trap or ion transport devices is space charge. Large concentrations of charged particles can distort applied DC and RF potentials, leading to poor transport efficiencies and increased spatial distributions. Robust simulation methods must be employed in order to mitigate these effects and to gain a better understanding of the device in the presence of space charge. However, standard ion optics software, such as SIMION [1], have limited, or no, ability to handle space charge. Therefore, other, more specialized, techniques must be used.

The particle-in-cell (PIC) method has been used to study plasmas and gravitational systems for decades, typically employing 2D cartesian or cylindrical or 3D cartesian coordinate systems. Thorough treatments of the subject can be found in [2,3]. Recently, a 3D PIC code in cartesian coordinates was developed to study the time evolution of large numbers of ions in a FT-ICR cell [4], demonstrating that simulations of up to 10^6 particles and 3×10^6 time steps are now feasible with modern computers.

However, there are many situations where a 3D cylindrical model space is desirable. As the PIC algorithm requires the electric potential to be solved at every time step, this potential-solving routine must be fast. Fast available solvers, based on Fast Fourier Transform (FFT) routines, require the boundary conditions to be specified. If the model space and the application of interest possess the same symmetry, then the boundary conditions can be specified in a very natural way, e.g., a cylindrical Penning trap where RF fields can be applied by periodically changing the potentials assigned to the boundaries.

In this paper a 3D PIC code in cylindrical coordinates, 3DCylPIC, is presented. It was developed to study the effects of space charge in a variety of applications where cylindrical boundary conditions are the natural choice. The code also allows for the inclusion of magnetic fields and buffer gases. The first section will cover the structure of the code and describe the algorithms used, while the second section will be devoted to the application of 3DCylPIC to the study of a cylindrical RF carpet design.

2. Structure of 3DCylPIC

3DCylPIC is a fully 3D PIC code in cylindrical coordinates written in the C++ programming language. It consists of a handful of classes which can be used to construct a simulation. By using a modern object-oriented programming language, the code is very modular, which minimizes the amount of recoding necessary to construct an entirely new simulation. As performance is a key concern, the Intel C++ compiler is used along with the Intel Math Kernel Library for FFT's and random number generation.

A simplified block diagram of the execution flow structure of 3DCylPIC is shown in Fig. 1. The boxes with the grey fill indicate a topic which will be further elaborated upon in the section listed to the left of the box. The topics will be covered in an order which makes comprehension of the program as a whole easier, which is not necessarily the chronological order of execution. The program begins by initializing the ions' properties such as charge,

E-mail address: ringle@nscl.msu.edu

1387-3806/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.ijms.2010.12.015

mass, initial position and velocity, etc. It is not required that all of the ions be identical. The total number of ions is limited only by memory and practical computation time. In practice, several hundreds of thousands of ions are routinely used. In the next phase any optional classes are initialized. These include scattering processes if a buffer gas is used, or generating and loading a capacity matrix if electrodes within the cylindrical model space are required. Details of these optional classes will be discussed in a later section. Potential values are then assigned to the boundaries. Currently only Dirichlet-type boundary conditions have been implemented. The individual charges from the group of ions are assigned to nearby grid points, and a charge density at each grid point is calculated. Now Poisson's equation is solved on the grid, after which the electric field can be calculated. The newly calculated electric field is used in updating each ions' position and velocity. Finally, the boundary conditions are changed, if necessary, and the cycle loops back to the distribute charge phase. The cycle continues, incrementing in time steps of length dt, until the total simulation time has been reached, or other end-of-simulation criteria are met.

2.1. Solving Poisson's equation in a cylindrical model space

At the heart of 3DCylPIC is the algorithm used to calculate the electric potential in a cylindrical model space with length $L_\circ$ and radius $R_\circ$. Poisson's equation in cylindrical coordinates can be written as

$$\frac{\partial^2\phi}{\partial r^2} + \frac{1}{r}\frac{\partial\phi}{\partial r} + \frac{1}{r^2}\frac{\partial^2\phi}{\partial\theta^2} + \frac{\partial^2\phi}{\partial z^2} = -\frac{\rho}{\epsilon_\circ}, \tag{1}$$

where $\phi(r, z, \theta)$ is the electric potential, r is the radial coordinate, θ is the azimuthal angle, z is the axial coordinate, ρ is the charge density, and $\epsilon_\circ$ is the permittivity of free space. This can be solved quickly using the algorithm given in [5], and will be summarized here.

2.1.1. Summary of potential solver algorithm

It is given that the boundary conditions on a cylinder are Dirichlet such that $\phi(r, L_\circ, \theta) = \phi_T(r, \theta)$, $\phi(r, 0, \theta) = \phi_B(r, \theta)$ and $\phi(R_\circ, z, \theta) = \phi_S(z, \theta)$. As the solution for ϕ must be periodic in θ it can be written as a truncated Fourier series such that

$$\phi(r, z, \theta) = \sum_{n=-N_\theta/2}^{N_\theta/2-1} \phi_n(r, z)e^{in\theta}, \tag{2}$$

where ϕ_n is the complex Fourier coefficient given by

$$\phi_n(r, z) = \frac{1}{N_\theta}\sum_{k=0}^{N_\theta-1} \phi(r, z, \theta)e^{-in\theta_k}, \tag{3}$$

and $\theta_k = 2k\pi/N_\theta$, where N_θ is the number of grid points along a circle. By inserting Eq. (2) into Eq. (1) and equating the Fourier coefficients we arrive at

$$\frac{\partial^2\phi_n}{\partial r^2} + \frac{1}{r}\frac{\partial\phi_n}{\partial r} + \frac{\partial^2\phi_n}{\partial z^2} - \frac{n^2}{r^2}\phi_n = -\frac{\rho_n}{\epsilon_\circ} = -\frac{1}{\epsilon_\circ \cdot N_\theta}\sum_{k=0}^{N_\theta-1} \rho(r, z, \theta_k)e^{-in\theta_k}. \tag{4}$$

Notice that in Eq. (4) the dependence on θ has been removed and we are left with N_θ two-dimensional equations, in terms of the Fourier coefficients, to solve. This requires that the boundary values, ϕ_T, ϕ_B, and ϕ_S also be transformed in the same manner as in Eq. 3 to ϕ_T^n, ϕ_B^n, and ϕ_S^n.

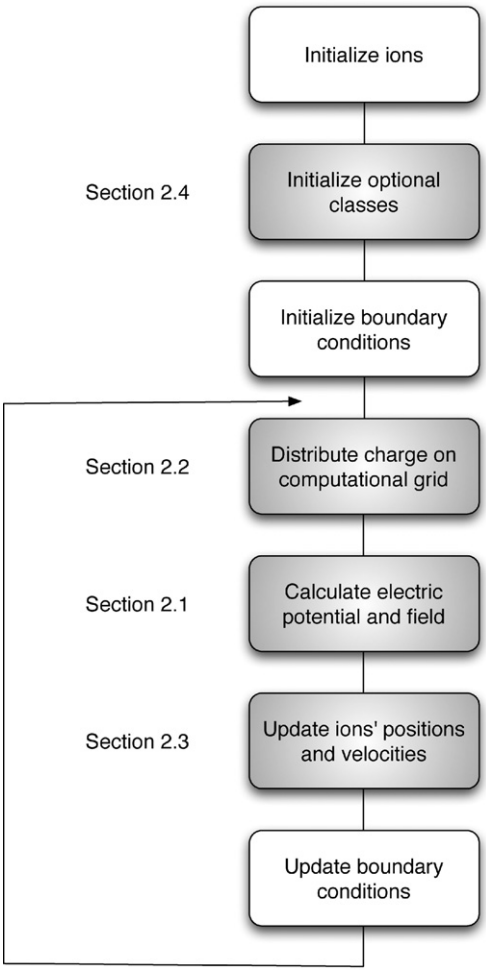

Fig. 1. Simplified execution flow structure of 3DCylPIC.

Next we choose a grid in the r–z plane such that for $N_{r,z}$ being the number of points in the radial and axial directions, respectively

$$r_i = (i - 1/2)\Delta r \quad \text{and} \quad z_j = j\Delta z, \tag{5}$$

for $1 \le i \le N_r + 1$ and $0 \le j \le N_z + 1$, with $\Delta r = 2/(2N_r + 1)$ and $\Delta z = 1/(N_z + 1)$ as the spacing between grid points in the radial and axial directions. Using these conventions yield $r_1 = \Delta r/2$, $r_{N_r+1} = R_\circ$, $z_0 = 0$, and $z_{N_z+1} = L_\circ$. As Eq. (4) will be solved on a grid, the derivatives must be discretized, and the centered difference approximations are used

$$\left.\frac{\partial\phi}{\partial x}\right|_{x_p} \simeq \frac{\phi_{p+1} - \phi_{p-1}}{2\Delta_p}$$
$$\left.\frac{\partial^2\phi}{\partial x^2}\right|_{x_p} \simeq \frac{\phi_{p+1} - 2\phi_p + \phi_{p-1}}{\Delta_p^2} \tag{6}$$

where Δ_p is the spacing between the grid points in the x direction. The functions must also be discretized such that $\phi_{i,j}^n \approx \phi_n(r_i, z_j)$ and $\rho_{i,j}^n \approx \rho_n(r_i, z_j)$. Applying Eq. (6) to Eq. (4) yields

$$\frac{\phi_{i+1,j}^n - 2\phi_{i,j}^n + \phi_{i-1,j}^n}{\Delta r^2} + \frac{1}{r_i}\frac{\phi_{i+1,j}^n - \phi_{i-1,j}^n}{2\Delta r} + \frac{\phi_{i,j+1}^n - 2\phi_{i,j}^n + \phi_{i,j-1}^n}{\Delta z^2} - \frac{n^2}{r_i^2}\phi_{i,j}^n = -\frac{\rho_{i,j}^n}{\epsilon_\circ}. \tag{7}$$

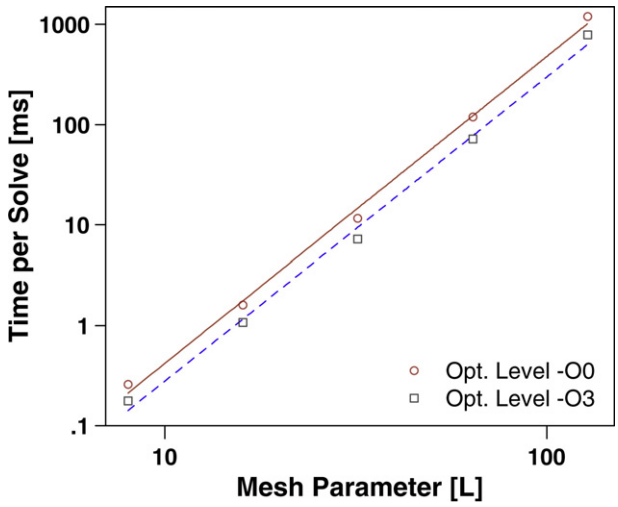

Fig. 2. Time to solve Poisson's equation on a grid with parameter size L for code not optimized by the compiler (−O0) and highly optimized code (−O3). Lines are fits to the data.

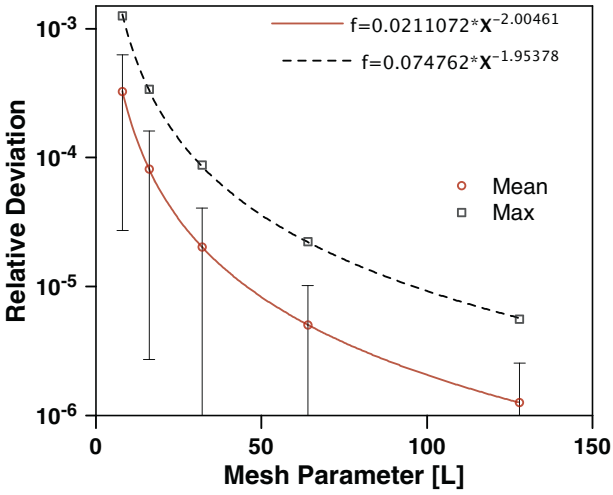

Fig. 3. Relative mean and maximum deviation of the calculated value from the true analytical value for the test function $f(r, z, \theta) = \exp(r \cdot \cos\theta + r \cdot \sin\theta + z)$ as a function of the grid parameter. The solid and dashed lines represent fits to the data.

The boundary conditions are given for $\phi^n_{N_r+1,j} = \phi^n_S(z_j)$, $\phi^n_{i,0} = \phi^n_B(r_i)$, and $\phi^n_{i,N_z+1} = \phi^n_T(r_i)$. From Eq. (7) it can be seen that the $\phi^n_{0,j}$ term cancels out, which is a consequence of using a staggered grid and eliminates the need for any pole conditions at $r = 0$.

Eq. (7) represents N_θ systems of linear equations, each of which can be solved using the Fourier method presented in [6]. Each system must be solved and then the potential at the interior grid points is obtained by transforming the $\phi^n_{i,j}$'s to $\phi_{i,j}$'s using Inverse Fourier Transforms.

2.1.2. Speed and accuracy of the Poisson solver

Two of the primary concerns regarding using a Poisson solver in a PIC code are speed and accuracy. The number of computations required to solve for the electric potential, and the resulting accuracy at which it does so, is an inherent property of the algorithm. The actual time required to perform the calculation depends on hardware, complier efficiency, etc.

Fig. 2 shows the time required to solve Poisson's equation on a grid with size parameter L, where L is defined as $(N_r, N_z, N_\theta) = (L, L, 2L)$. In these tests the Intel C++ compiler, version 11.1.088, was used on a Macbook Pro with a 2.66 GHz Core i7 processor running OSX 10.6.4. Using the full optimization flag, −O3, results in a significant speed increase, a factor of about 1.5, over no compiler optimzations, −O0. Although the Fourier Transform and Linear system solving routines implemented in the MKL support multicore operation, enabling it does not result in a reduced execution time. In fact, the overhead associated with multicore operation actually increases the execution time. This is most likely due to the fact that none of the individual operations are of sufficient size to see a reduced execution time due to the overhead, as each time Poisson's equation is solved the problem is broken down into N_θ independent linear systems. However, as these linear systems are independent, they could be solved in parallel which would likely reduce the execution time, but has not yet been implemented.

The accuracy of the Poisson solver was tested by comparing calculated values against the true analytical solution. The function $f(r, z, \theta) = \exp(r \cdot \cos\theta + r \cdot \sin\theta + z)$ was chosen for simplicity as $\nabla^2 f = 3f$. Fig. 3 shows the relative deviation of the calculated value from the analytical value on a grid with grid parameter L and characteristic dimensions of $L_\circ = R_\circ = 1$. The square data points represent the maximum relative deviation while the circles represent the mean relative deviation over all points on the grid. The lines represent fits to the data. This agrees with the results presented in [5] and reassures us that the algorithm has been implemented correctly.

2.2. Charge and force distribution on the cylindrical grid

The simplest method of assigning the charges to the grid is to iterate over each particle and add its charge to the nearest grid point (NGP). Similarly, the easiest method to obtain the force on a particle is to calculate it from the electric field at the NGP. These methods are also computationally inexpensive. However, simplicity comes at the cost of crude charge assignment and discontinuities in the force a particle experiences as it moves through a cell, resulting in a departure from the physical reality one is trying to achieve with the simulation [3].

The method employed in 3DCylPIC is the cloud-in-cell (CIC) scheme [7], which results in much less noise as compared to the NGP method. Here the charge from a particle is distributed over the grid points in the cell which the particle is contained. A volume-weighting method is used to determine what fraction of a particle's charge is assigned to a given grid point, which is illustrated in Fig. 4. For ease of visualization, the figure shows a 2D area weighting, but is easily extended to 3D by extruding the cartoon in or out of the plane defined by points. In the top cartoon of Fig. 4 a particle with $r > r_1$ is located within an area in a polar coordinate space defined by the points A–D. In the CIC scheme, the first step is to determine the position of the points A–D. Next, the areas a, b, c, and d are calculated. For charge distribution the charge assigned to a point is $Q_X = Q_T \cdot x/(a + b + c + d)$, where $X \in (A, B, C, D)$, $x \in (a, b, c, d)$, and Q_T is the particle's total charge. Similarly this method is used to calculate the weighted-mean of the electric field at the particle's position. The bottom cartoon of Fig. 4 illustrates the weighting scheme used for a particle with $r < r_1$.

2.3. Integration of the equations of motion

After the charges have been distributed on the grid, and the electric potential has been calculated, the next phase consists of updating the ions' positions and velocities. To do so, it is necessary to integrate the equations of motion for every particle. Therefore, we want a method which is as fast as possible and which remains suitably accurate. One of the most commonly used integration schemes which meets these requirements is the leap-frog method, illustrated in Fig. 5. This method integrates the velocities and posi-

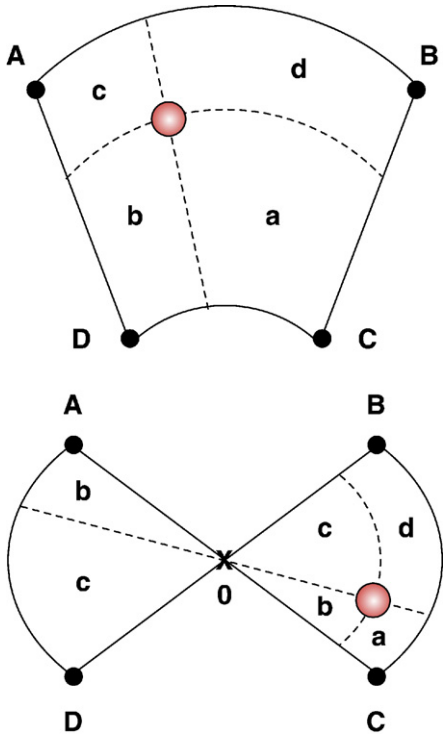

Fig. 4. Illustration of the CIC weighting scheme shown in 2D for simplicity. (Top) Scheme for a particle with $r > r_1$. Weight is assigned to the grid points A–D according to the areas a–d. (bottom) Same as above, except the particle has $r < r_1$.

tions in increments of time, dt, but the time at which the position and velocity are known are offset by $dt/2$. This time centering makes the leap frog method accurate to second order.

It may be desirable to include magnetic fields in 3DCylPIC simulations, so the equations of motion which must be integrated are

$$m\frac{d\mathbf{v}}{dt} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \tag{8}$$

$$\frac{d\mathbf{r}}{dt} = \mathbf{v}, \tag{9}$$

where m is the mass of the particle, q is the particle's charge, $\mathbf{E}$ is the electric field, $\mathbf{B}$ is the magnetic field, $\mathbf{v}$ is the particle's velocity, and $\mathbf{r}$ is the particle's position. Making use of a centered-difference form of the Lorentz equation, Eq. (8) becomes

$$\frac{\mathbf{v}_{t+dt/2} - \mathbf{v}_{t-dt/2}}{dt} = \frac{q}{m}\left(\mathbf{E} + \frac{\mathbf{v}_{t+dt/2} + \mathbf{v}_{t-dt/2}}{2} \times \mathbf{B}\right). \tag{10}$$

Using the method presented in [8], we are able to decouple the electric and magnetic forces by making the following transformation,

$$\mathbf{v}_{t-dt/2} = \mathbf{v}^- - \frac{q\mathbf{E}}{m}\frac{dt}{2} \tag{11}$$

$$\mathbf{v}_{t+dt/2} = \mathbf{v}^+ + \frac{q\mathbf{E}}{m}\frac{dt}{2}. \tag{12}$$

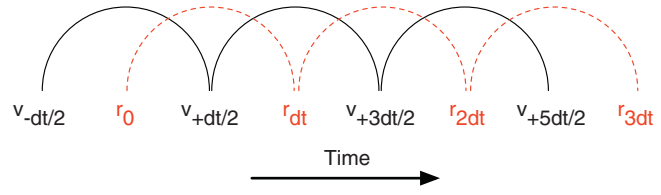

Fig. 5. Cartoon illustrating the leap-frog integration method. Both velocity and position are incremented in steps of dt, but are offset from one another by $dt/2$.

and inserting them into Eq. (10) to obtain

$$\frac{\mathbf{v}^+ - \mathbf{v}^-}{dt} = \lambda \cdot \frac{q}{2m}(\mathbf{v}^+ + \mathbf{v}^-) \times \mathbf{B}. \tag{13}$$

A correction term, $\lambda = \tan(\omega_c \cdot dt/2)/(\omega_c \cdot dt/2)$, is introduced in Eq. (13) to correct for small errors in the true cyclotron frequency, ω_c, introduced due to a fixed time step [2]. Letting $\mathbf{B} = B\hat{\mathbf{z}}$, and solving for $\mathbf{v}^+$ in the radial plane, the solution can be written in terms of a rotation

$$\begin{pmatrix} v_r^+ \\ v_\theta^+ \end{pmatrix} = \begin{pmatrix} \frac{1-\alpha^2}{1+\alpha^2} & \frac{2\alpha}{1+\alpha^2} \\ -\frac{2\alpha}{1+\alpha^2} & \frac{1-\alpha^2}{1+\alpha^2} \end{pmatrix} \begin{pmatrix} v_r^- \\ v_\theta^- \end{pmatrix}, \tag{14}$$

where $\alpha = \Omega_c \cdot dt/2$, with $\Omega_c = \lambda\omega_c$. Since v_z is parallel to $\mathbf{B} = B\hat{\mathbf{z}}$, the velocity integration in the axial direction is simply

$$v_{t+dt/2}^z = \frac{qE_z}{m}dt + v_{t-dt/2}^z. \tag{15}$$

The steps which must be followed in order to integrate the velocity are:

1. Calculate $v_{r,\theta}^-$ using Eq. (11).
2. Calculate $v_{r,\theta}^+$ using the rotation matrix in Eq. (14).
3. Transform $\mathbf{v}^+$ into $\mathbf{v}_{t+dt/2}$ using Eq. (11).
4. Calculate $v_{t+dt/2}^z$ using Eq. (15).

The next step is integrating the position of the particle. The differenced form of Eq. (9) is simply

$$\frac{\mathbf{r}_{t+dt} - \mathbf{r}_t}{dt} = \mathbf{v}_{t+dt/2}. \tag{16}$$

One issue that arises in the radial plane during the position integration is that as the particles get close to $r = 0$, the change in the θ coordinate becomes very large. One way to avoid this is to use a method described in [8], where the radial position advance is first calculated in cartesian coordinates, and then translated back into polar coordinates. No additional steps are required when advancing the particle in the axial direction. The steps which are followed to integrate the particle's position are

1. Calculate $x_{t+dt} = r_t + v_{r,t+dt/2} \cdot dt$ and $y_{t+dt} = v_{\theta,t+dt/2} \cdot dt$.
2. Calculate $r_{t+dt} = \sqrt{x_{t+dt}^2 + y_{t+dt}^2}$.
3. Calculate $\theta_{t+dt} = \theta_t + \beta$, where $\sin\beta = y_{t+dt}/r_{t+dt}$.
4. Rotate $\mathbf{v}_{t+dt/2}$ to $\mathbf{v}'_{t+dt/2}$ to account for the new angle, θ_{t+dt}.
5. Calculate $z_{t+dt} = z_t + v_{z,t} \cdot dt$.

Step 4 refers to a rotation of the radial velocity vector, as given before the position advance, that is necessary to preserve $\mathbf{v}$, and is given by the rotation matrix

$$\begin{pmatrix} v_r' \\ v_\theta' \end{pmatrix} = \begin{pmatrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{pmatrix} \begin{pmatrix} v_r \\ v_\theta \end{pmatrix}, \tag{17}$$

where $\sin\beta = y_{t+dt}/r_{t+dt}$ and $\cos\beta = x_{t+dt}/r_{t+dt}$, and the primed values are components of the rotated velocity vector. If the particle's position is on the axis, $r_{t+dt} = 0$, then set $\cos\beta = 1$ and $\sin\beta = 0$ to make the momentum purely radial.

2.4. Optional classes

2.4.1. interaction with a buffer gas

In many cases it may be desirable to simulate interactions with a buffer gas, e.g., transport of ions through some form of gas stopping cell along an RF carpet. Two different models for buffer gas interactions have been included, a hard sphere interaction, similar

to the implementation described in [4], and a statistical diffusion simulation (SDS) model [9]. The hard sphere interaction can be used when the probability of multiple scatters per time step is insignificant. However, it becomes impractical at higher pressures when the rate of collisions begins to dictate shorter time steps to avoid missing multiple scatter events. The SDS model, on the other hand, incorporates mobility and random ion jumping to simulate many collisions per time step. As implemented in 3DCylPIC, neither model includes bulk gas flow. For more details on both models, the reader is referred to the works referenced previously.

2.4.2. internal boundary points

There are times when a simple cylindrical boundary will not suffice, and one would like to define points within the cylindrical model space to have a certain potential, e.g., an internal electrode. This is possible by precalculating a capacitance matrix, C, which relates the potential and charge on a given number of grid points within the model space [3]. The capacitance matrix is used to calculate the amount of surface charge induced on the internal boundary points by every other internal boundary point. The capacitance matrix is constructed in the following manner:

1. In turn, place a unit charge on each internal boundary point and solve Poisson's equation (with zero everywhere else).
2. Record the potential induced on every other internal boundary point.
3. Form a matrix A, whose columns consist of the induced charge calculated above.
4. Invert A to obtain C ($C=A^{-1}$).

It should be noted that the matrix A is lower triangular as, in this case, the induced potential depends only on the vector distance between the two internal boundary points. This is used to reduce calculation time and storage space for the matrix. The space required to store the matrix is then $l(l+1)/2$, where l is the number of internal boundary points.

In order to use the capacitance matrix to set the potential of the internal boundary points, Poisson's equation is solved and the difference in the potential from the desired value is recorded. This difference, when negated and multiplied by the capacitance matrix, yields the amount of surface charge necessary to achieve the desired potential at each internal boundary point. That charge is then assigned to each internal boundary point and Poisson's equation is solved again, resulting in the correct potential in the entire model space.

The obvious downsides to this method is the time it takes to pre-calculate the capacitance matrix, and having to solve Poisson's equation twice per time step. One mitigating factor is that once the capacitance matrix is calculated, it can be saved and loaded on demand as it only changes if the internal boundary points are changed. However, there is no way to avoid the two solves of Poisson's equation.

3. Application – ion transport along cylindrical RF carpet

The conversion of high-energy rare-isotope beams to high-quality low-energy beams has gained a lot of attention in the past decade as it is able to match the reach and chemical-independent nature of beams created via projectile fragmentation to experiments formerly restricted to ISOL-type facilities. Currently the LEBIT facility [10] at the NSCL, and the SLOWRI facility [11] at RIKEN have successfully thermalized such high-energy projectile-fragmentation beams and delivered them to experiments. Thermalization of the rare isotopes was achieved with the use of gas cells [12,13], gas-filled chambers where the ions thermalize by collisions with the buffer gas after an initial energy degradation stage.

After the ions have thermalized, it is necessary to transport them out of the cell in the fastest, most efficient way possible. Traditional methods include employing RF transport devices, DC gradients, or both. One such RF transport device, the RF carpet, is currently in use at SLOWRI [11,13], and is being actively researched [14,15] for the next-generation gas stopping systems at the NSCL.

Space charge plays a large role in limiting transport efficiencies in gas cells used to thermalize high-energy beams. Thermalization of an ion with ≈100 MeV of energy when it enters the gas cell can create 10^6 or more He^+-e^- pairs per stopped ion. The amount of space charge created during the stopping process becomes more pronounced for increasingly intense beams, distorting applied RF and DC potentials which leads to a reduced transport efficiency of the ions of interest. The thermalized ions of interest, in sufficient quantity, can also create significant space charge potentials as they are being transported across RF devices.

3.1. Model of an RF carpet device

The system we are interested in modeling consists of a cylindrical RF carpet with a surface ion source positioned 4.5 cm above the carpet surface as this corresponds to a real prototype setup currently being tested at NSCL. Fig. 6 shows a cartoon of the RF carpet model. It consists of a series of ring electrodes with widths of 0.2 mm, separated by a space of 0.2 mm. Two RF signals, with a 180° relative phase shift, are applied to alternating ring segments. Groups of four ring segments are coupled to an independent DC signal. This allows a DC gradient, or drag field, to be applied on top of the RF, drawing the ions toward the center of the carpet. At the center of the carpet is a circle with a radius of 1 cm which is held at a potential of 0 V, and serves as a collection pad. The model is created by setting the boundary conditions on one of the axial boundaries. In order to generate a DC gradient in the axial direction, called a push field, the same boundary conditions are applied on the opposite axial boundary, minus the RF, with the DC values raised by a constant value. The boundary values on the side wall of the model space also are decreased linearly to form the push field. The total DC potential, or the sum of the push and drag potentials, is illustrated in Fig. 6a as contour lines.

The combination of applied RF and push fields, modified by the damping due to the buffer gas, creates a time-averaged effective potential that, for the right choice of operating parameters, has a minimum some distance above the carpet. The analytical solution for the effective potential of an RF carpet is given in [15]. To illustrate, Fig. 7 shows a plot of the effective potential, above the center of an electrode, as a function of the magnitude of the applied push field using the parameters given in Table 1. A minimum in the effective potential occurs around 0.55 mm above the carpet surface using these parameters.

The most efficient way to simulate the entire system, from ion source to successful transport, is to break it up into two separate simulations. As the ions are not affected by the RF carpet until they are sufficiently close (<1 mm), the drift of the ions down to 2.5 mm above the carpet can be simulated relatively fast as the grid resolution and time step can be relaxed, with respect to simulating transport across the carpet. Once the ions are close to the carpet, and begin to feel its effect, the resolution must be increased, particularly in the r and z dimensions. Additionally, the time step must be small enough to adequately sample the RF at the applied frequency.

3.2. Radial expansion of an ion beam due to diffusion and space charge effects

A series of simulations was run to determine the expansion of the beam in the radial plane as a function of ion current and magni-

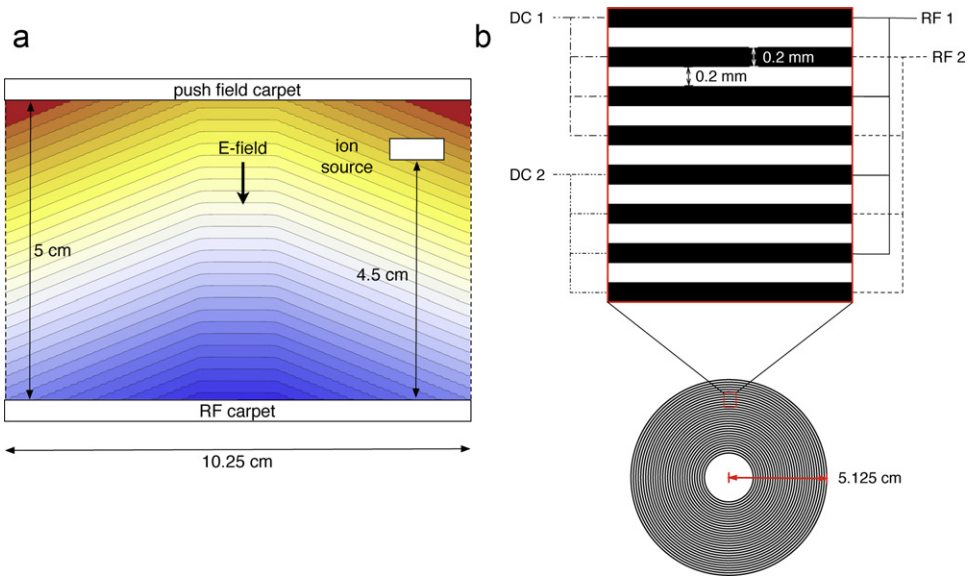

Fig. 6. (a) Side-view schematic illustration of the RF carpet system with DC potential contour lines. (b) Top-view schematic illustration of the RF carpet with applied RF and DC signal routing and electrode size and spacing illustrated in the zoomed area.

tude of the push field. The goal was to determine the beam radius slightly above the surface of the RF carpet for use as initial conditions for later transport simulaions. In the limit of no space charge, the results from 3DCylPIC were checked against results from a SIMION simulation using the hard sphere collision model HS1.

The model space is a cylinder 5 cm in length with a 3 cm radius. The source of ions is a plane 4.5 cm above the lower boundary, on axis, with a radius of 3.175 mm. The initial ion distribution was assumed to be uniform over this area. The boundary conditions on the two ends and sidewall were chosen such that a push field with constant magnitude is applied. No drag field was applied. Simulated ions with a mass-to-charge ratio of $A/Q = 85$ were used as a Rb ion source is installed in the test chamber. A buffer gas of He at a pressure $P = 60$ Torr and temperature $T = 273$ K was used. In the 3DCylPIC simulations the total simulation time ranged from 1.5 ms to 3 ms, depending on the magnitude of the push field. In all cases, the radial positions of more than 175,000 ions were recorded when they reached 2.17 mm above the lower boundary, which will be the starting point for the subsequent transport simulations. The SIMION simulations were limited to 100 ions due to the lengthy simulation time. The characteristic size of the final ion distribution was taken to be a circle containing 95% of the ions.

Fig. 8a shows a snapshot of all ions' positions at the end of the simulation for a 9.6 nA beam with a push field magnitude of 35 V/cm. In Fig. 8a the radius of the bounding circle containing 95% of the ions at a distance of 2.17 mm above the lower boundary is plotted versus the applied push field for several values of the ion beam current. Also shown are the results of a similar SIMION simulation in which there is no space charge. As one can see, there is good agreement between the SIMION and low-current (28.8 pA) case. It should be noted that additional simulations were run with 9.6 nA where the time step was halved and the resolution doubled. These simulations yielded no significant difference from those using the standard time step, $dt = 50$ ns, and grid size, (64, 64, 64).

As expected, larger currents result in a more diffuse beam just above the carpet due to an increased space charge. As the magnitude of the push field is increased, the time required for the ions

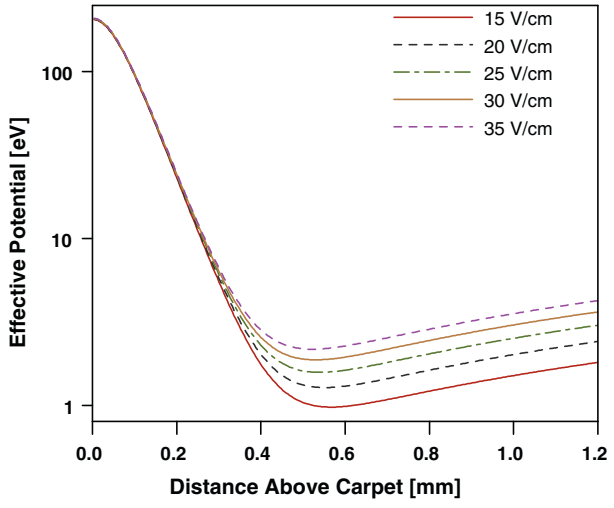

Fig. 7. Effective potential, above the center of an arbitrary ring electrode, of an RF carpet as a function of the distance above the carpet for multiple strengths of the applied push field, calculated using the formalism presented in [15] using input parameters given in Table 1.

Table 1
Simulation parameters for ion transport across the RF carpet

Grid parameters	
Num. radial points	512
Num. axial points	128
Num. azimuthal points	32
$R_\circ$	51.25 mm
$L_\circ$	2.5 mm
Initial ion distribution	
Ion A/Q	85
Mean radial pos.	40 mm
Axial pos.	2.17 mm
Mean azimuthal pos.	0.785 rad
Radial extent	varies
Environmental parameters	
Drag field mag.	3 V/cm
Buffer gas mass	4 amu
Buffer gas temp.	273 K
Buffer gas press.	80 mbar
Reduced mobility (Rb^+ in He)	21.2 $cm^2/(V\,s)$
RF amp.	85 V
RF freq.	2 MHz
Run parameters	
Run time	6 ms
Time step, dt	20 ns
Ion current	Varies

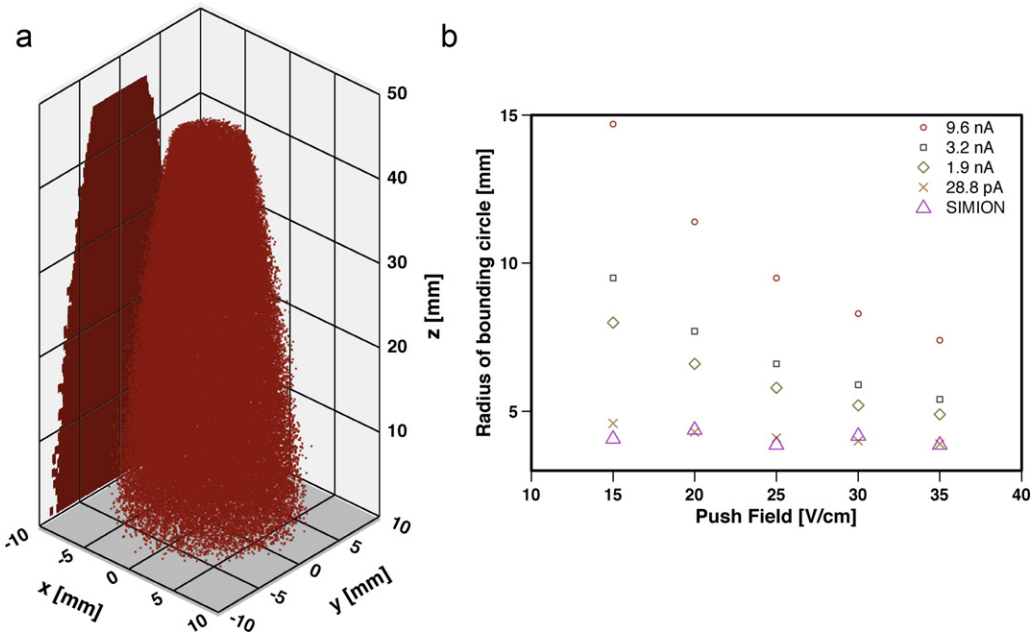

Fig. 8. (a) Snapshot of all ions' positions at the end of an expansion simulation for a 9.6 nA beam with a push field magnitude of 35 V/cm. The positions of all ions as they cross the $z = 2.17$ mm plane are recorded for use in later transport simulations. (b) Radius of the circle enclosing 95% of ions at $z = 2.17$ mm as a function of the applied push field. Results for several ion beam currents are shown, along with the results from a separate SIMION (no space charge) simulation.

to traverse the distance between the ion source and the carpet is reduced, resulting in an increased current density for a given ion current. This information can then be used to specify the initial ion distribution in a subsequent simulation of ion transport across the carpet.

3.3. Ion transport across the RF carpet

Using the initial ion distributions above the carpet from the previous section several transport simulations were performed where the magnitude of the push field was varied. The initial positions of the ions were uniformly distributed over a circle in the $z = 2.17$ mm plane, ignoring the effects of the pillar of charge extending up to the position of the ion source. A complete list of parameters used in the transport simulations are given in Table 1. For the current densities given here, a grid size of (512, 128, 32) proved sufficient. Increasing the resolution did not significantly change the results. At an initial current density of ≈0.3 nA/mm^2, increasing the grid size from (512, 128, 32) to (512, 128, 64) increased the relative transport efficiency by about 18%. However, this current density is about an order of magnitude greater than the highest density presented in these results. It should also be noted that all ion losses occur on the carpet, and that none cross the radial boundary of the model space.

At every time step, two ions are created within the spatial distribution given in Table 1. The charge of the ions are set to obtain a desired current, with the mass being adjusted such that $A/Q = 85$ is constant. For example, for $dt = 20$ ns, if an ion current of 9.6 nA is desired then one must set $A = 51,000$ and $Q = 600$.

For interaction with the buffer gas, $A = 85$ and $Q = 1$ are always assumed. If any of the ions leave the bounds of the simulation they are counted as lost. They are counted as being successfully transported if they cross the $z = 0$ plane with $r < 1$ cm, striking the collection pad. For example, Fig. 9 shows a plot of ions transported or lost as a function of unit time for a 1.9 nA beam and a push field

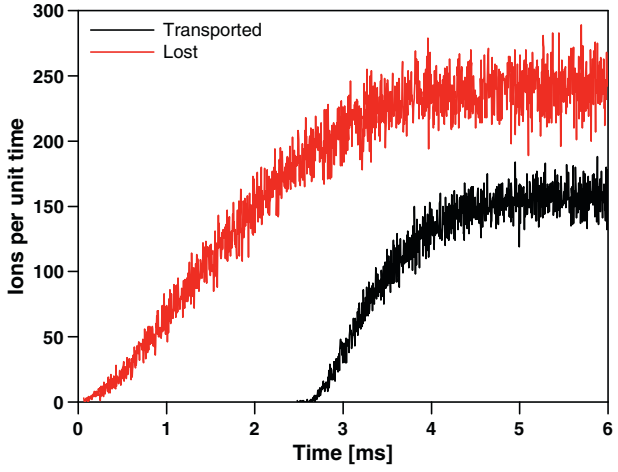

Fig. 9. Number of ions successfully transported or lost per unit time for an ion current of 1.9 nA and push field with a magnitude of 25 V/cm.

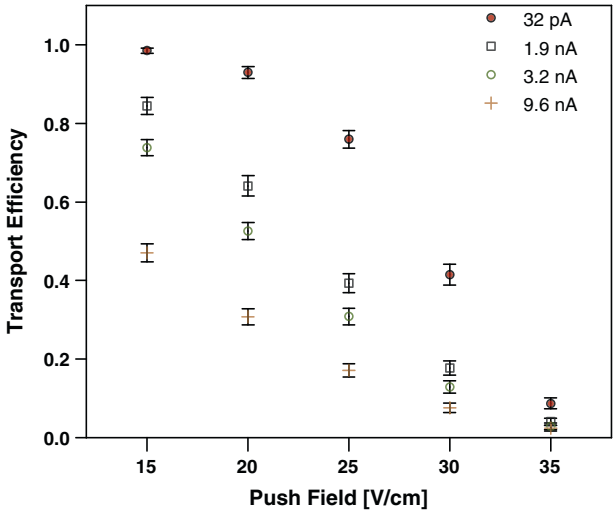

Fig. 10. Ion transport efficiency across the RF carpet as a function of the push field magnitude for different ion beam currents.

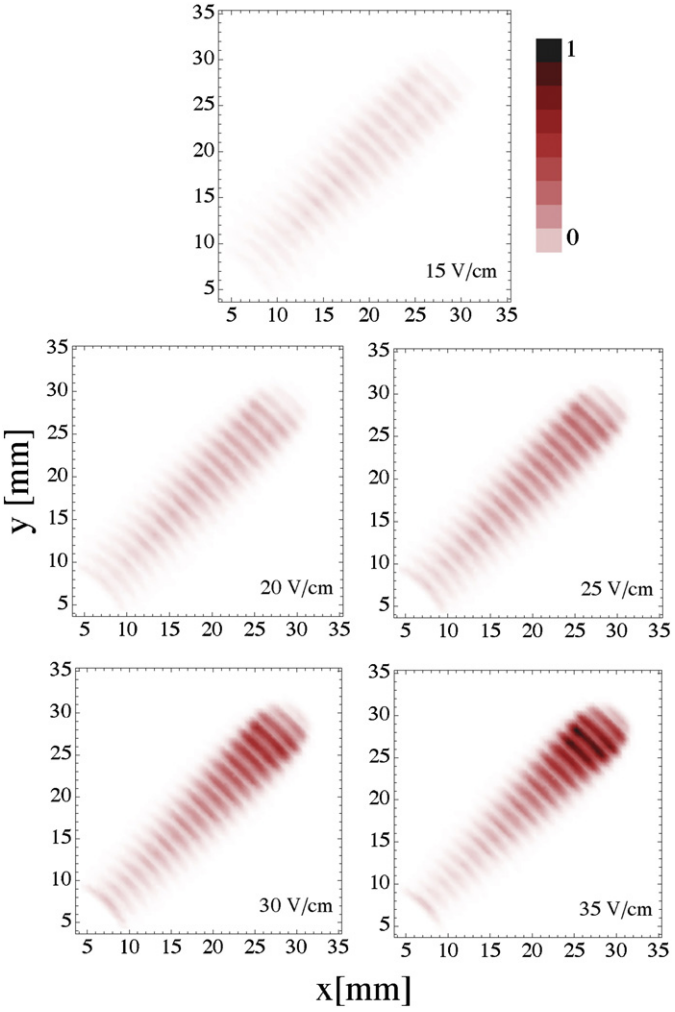

Fig. 11. Normalized density plots of ions lost for different push field magnitudes with a beam current of 1.9 nA.

with a magnitude of 25 V/cm. One can see from the red (light) curve that ions are lost almost immediately, while the black (dark) curve shows that it takes about 2.5 ms for the first ions to be transported. This is consistent with a transport time of $t = 3$ ms calculated using a mean velocity of 8 mm/ms, from a mobility of $K \approx 269\,\mathrm{cm}^2/(\mathrm{V\,s})$ and a drag field magnitude of 3 V/cm, and a travel distance of 24 mm (from the edge of the collection pad to the closest edge of the ion distribution). In this simulation the unit time is equal to $200 \cdot dt$, and as 2 ions are released per dt, a total of 400 ions are released per unit time. In equilibrium, the average number of ions lost and transported should be equal to 400. As this is true in the last millisecond of the simulation, the average number of ions lost and transported during this time can be calculated and used to determine the steady-state transport efficiency. Fig. 10 shows the steady-state transport efficiency as a function of push field magnitude for four different ion beam currents. Increasing the space charge has the expected effect of reducing the transport efficiency for a given push field magnitude.

It is also instructive to record the final positions of ions that are lost. Fig. 11 shows normalized density plots of ions which are lost on the carpet for several values of the push field magnitude and an ion beam current of 1.9 nA. The first feature one notices is that there is a band structure in the ion loss plots. The areas of minimal ion loss are centered on the steps in the DC gradient used to draw the ions toward the collection pad. The ions move faster over these steps, spending less time in these areas and reducing the probability of being lost there. In addition, for the smaller push field magnitudes (15 and 20 V/cm) the ion losses are fairly uniform in the radial direction which should lead to a linear relation between transport distance and total transport efficiency in these cases. However for larger values of the push field magnitude the ion losses are concentrated directly under the ion source. This effect could be mitigated by transporting the ions faster across the carpet, reducing the accumulated space charge directly under the source. In this scheme that would mean increasing the drag field magnitude, resulting in larger steps across the individual DC segments shown in Fig. 6. Another solution, described in [14], would be to use a traveling wave superimposed on top of the RF, allowing the ions to "surf" the traveling wave and remove the need for a separate DC gradient.

4. Summary

A fully 3D PIC code for space charge simulations in cylindrical coordinates has been presented. The choice of a cylindrical coordinate system makes it a natural choice for simulations where a cylindrical boundary is more appropriate. Internal electrodes, background gas, and magnetic fields can be included. It was used to study ion transport across an RF carpet device similar to one that has been prototyped at the NSCL. Future applications may include charge-state evolution of ions in an EBIT, electron or proton cooling of highly charged ions in a cylindrical Penning trap, additional RF carpet simulations using traveling waves, and more.

Acknowledgements

I wish to acknowledge the support of Michigan State University, the National Science Foundation under Cooperative Agreement No. PHY-06-06007. I would also like to thank Stefan Schwarz for his advice on the subject.

References

[1] D.A. Dahl, Simion for the personal computer in reflection, Int. J. Mass Spectrom. 200 (1–3) (2000) 3–25.
[2] C.K. Birdsall, A.B. Langdon, Plasma Physics Via Computer Simulation, McGraw-Hill, New York, 1985.
[3] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, A. Hilger, Bristol [England]; Philadelphia, 1988.
[4] E.N. Nikolaev, R.M.A. Heeren, A.M. Popov, A.V. Pozdneev, K.S. Chingin, Realistic modeling of ion cloud motion in a fourier transform ion cyclotron resonance cell by use of a particle-in-cell approach, Rapid Commun. Mass Spectrom. 21 (22) (2007) 3527–3546.
[5] M.C. Lai, W.W. Lin, W. Wang, A fast spectral/difference method without pole conditions for Poisson-type equations in cylindrical and spherical geometries, IMA J. Numer. Anal. 22 (4) (2002) 537–548.
[6] B.L. Buzbee, G.H. Golub, C.W. Nielson, Direct methods for solving Poisson's equation, SIAM J. Numer. Anal. 7 (4) (1970) 627.
[7] C.K. Birdsall, D. Fuss, Clouds-in-clouds, clouds-in-cells physics for many-body plasma simulation, J. Comput. Phys. 3 (4) (1969) 494–511.
[8] J.P. Boris, in: Proceedings of the Fourth Conference on Numerical Simulation of Plasma, 1970, pp. 3–67.
[9] A.D. Appelhans, D.A. Dahl, Simion ion optics simulations at atmospheric pressure, Int. J. Mass Spectrom. 244 (1) (2005) 1–14.
[10] R. Ringle, P. Schury, T. Sun, G. Bollen, D. Davies, J. Huikari, E. Kwan, D.J. Morrisey, A. Prinke, J. Savory, S. Schwarz, C. Sumithrarachchi, Precision mass measurements with lebit at msu, Int. J. Mass Spectrom. 251 (2–3) (2006) 300–306.
[11] M. Wada, Y. Ishida, T. Nakamura, Y. Yamazaki, T. Kambara, H. Ohyama, Y. Kanai, T.M. Kojima, Y. Nakai, N. Ohshima, A. Yoshida, T. Kubo, Y. Matsuo, Y. Fukuyama, K. Okada, T. Sonoda, S. Ohtani, K. Noda, H. Kawakami, I. Katayama, Slow ribeams from projectile fragment separators, Nucl. Instrum. Methods Phys. Res., Sect. B 204 (2003) 570–581.
[12] L. Weissman, D.J. Morrissey, G. Bollen, D.A. Davies, E. Kwan, P.A. Lofy, P. Schury, S. Schwarz, C. Sumithrarachchi, T. Sun, R. Ringle, Conversion of 92 MeV/u Ca-38/K-37 projectile fragments into thermalized ion beams, Nucl. Instrum. Methods Phys. Res., Sect. A 540 (2–3) (2005) 245–258.
[13] A. Takamine, M. Wada, Y. Ishida, T. Nakamura, K. Okada, Y. Yamazaki, T. Kambara, Y. Kanai, T.M. Kojima, Y. Nakai, N. Oshima, A. Yoshida, T. Kubo, S. Ohtani, K. Noda, I. Katayama, P. Hostain, V. Varentsov, H. Wollnik, Space-charge effects in the catcher gas cell of a rf ion guide, Rev. Sci. Instrum. 76 (10) (2005).
[14] G. Bollen, Ion surfing; with radiofrequency carpets, Int. J. Mass Spectrom. 299 (2011) 131–138.
[15] S. Schwarz, Rf ion carpets: the electric field, the effective potential, operational parameters and an analysis of stability, Int. J. Mass Spectrom. 299 (2011) 71–77.