



Parallel 3D Poisson solver for a charged beam in a conducting pipe

Ji Qiang*, Robert D. Ryne

Ms H817, LANSCE-1, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Received 27 February 2001; accepted 5 March 2001

Abstract

In this paper, we present a parallel three-dimensional Poisson solver for the electrostatic potential of a charged beam in a round or rectangular conducting pipe with open-end boundary conditions. This solver uses an eigenfunction expansion in the transverse direction and a finite difference method in the longitudinal direction. The computational domain in the longitudinal direction contains only the beam since only the potential inside the beam will be calculated. The potential on both ends of the beam is matched into the source-free region for each eigenmode. This method avoids the use of a large computational domain outside the beam to implement the open boundary condition. This saves unnecessary computational time and memory storage that would be required if a large computational domain was used to simulate the open boundary. Parallel implementation using a two-dimensional domain decomposition approach and a message passing paradigm shows good scalability on both distributed memory machines and distributed shared-memory machines. This solver has important applications in accelerator physics studies that involve modeling high-intensity beam dynamics. As a specific example, we present results from large-scale simulations of beam halo formation in a linear accelerator. © 2001 Elsevier Science B.V. All rights reserved.

PACS: 52.65.Rr; 52.75.Di

Keywords: Parallel Poisson solver; Open boundary condition; Eigenfunction expansion; Domain decomposition

1. Introduction

Solving the three-dimensional Poisson equation in a conducting pipe has important applications in beam dynamics studies in accelerator physics. Recently, there has been increasing interest in utilizing high intensity beams for future accelerator applications, e.g., accelerator-driven spallation neutron production for basic and applied research, the accelerator production of tritium and the accelerator transmutation of radioactive waste. These projects place extremely stringent requirements on particle loss, since even very small losses can lead to unacceptably high levels of radioactivity that can hinder or prevent hands-on maintenance. Such losses are known to be associated with a low density region of particles (called beam halo) radially far from the beam core. Understanding and predicting beam halo is a major issue for future accelerator applications involving high

* Corresponding author.

E-mail addresses: jiqiang@lanl.gov (J. Qiang), ryne@lanl.gov (R.D. Ryne).

intensity beams. To model the details of such a system, which involves strong self-fields as well as highly complex externally applied fields, leads us to a full Poisson–Vlasov description. The most widely used method in accelerator physics for solving the Poisson–Vlasov equations is the particle-in-cell (PIC) approach [1–6]. A key issue in these simulations is to solve the Poisson equation efficiently, at each time step, subject to the appropriate boundary conditions. Of particular importance is the computation of the electrostatic potential of a charged beam inside a conducting pipe with open ends.

A number of methods for solving Poisson’s equation have been studied [7–15]. However, all of these methods deal with the situation with a closed computational domain. For a system with open boundary conditions, a brute force method is to solve Poisson’s equation on a large domain so that the potential vanishes on all edges of domain. Furthermore, the choice of computational domain is not straightforward and usually requires solving Poisson’s equation twice to ensure the computational domain is large enough. This is inefficient for beam dynamics studies, since only the potential inside the beam is needed. Hockney et al. suggested an efficient method using fast Fourier transforms (FFTs) and zero padding to calculate the potential inside the beam with the open boundary condition in all three directions [14,15]. In an accelerator, the aperture of the beam pipe may be small in order to save cost, hence the effects of the conducting pipe on the beam are not negligible. The Poisson equation for the beam potential will have a finite boundary condition in the transverse plane and an open boundary condition in the longitudinal direction. A capacity matrix method has been used to find the image charge on the conducting wall and applied to the open boundary calculation including both the image charge and the beam charge as source terms to obtain the potential inside the beam [3]. However, this method involves calculating the capacity matrix and solving Poisson’s equation twice, and is not computationally efficient. An efficient algorithm in a two-dimensional plasma slab has been studied by Buneman and Langdon et al. [12,13]. This method uses a Fourier transform for the periodic boundary condition and an analytic solution of a finite difference equation in the free space region to set the boundary conditions on the open ends of the slab. In this paper, we generalize this algorithm to a three-dimensional conducting pipe by expanding the potential and source term in the transverse plane by the eigenfunction corresponding to the pipe and solving for the longitudinal dependence using a finite difference approximation inside the beam. The longitudinal dependence outside the beam can then be solved analytically for each transverse eigenmode. This solution is used to match the solution on both ends of the beam and to provide the boundary conditions needed for the Gaussian elimination of the potential inside the beam. This algorithm is also parallelized using a two-dimensional domain decomposition method and implemented using the message passing interface (MPI). All the communication operations are contained in a parallel matrix transpose.

The organization of this paper is the following: The physical model and numerical methods are described in Section 2. The parallel algorithm using MPI on distributed parallel machines is discussed in Section 3. An application of the solver to a simulation of beam halo formation in a linear accelerator transport system is presented in Section 4. The conclusions are drawn in Section 5.

2. Physical model and numerical methods

The physical system under consideration is a bunched charged-particle beam inside a conducting pipe. The length of the pipe is assumed to be infinitely long compared with the bunch length. Also, if we are considering a train of bunches, the effect of neighboring bunches is assumed to be negligible. For example, in a high-power radio frequency proton linear accelerator (linac), the linac is hundreds of meters long while the bunch length is typically of order millimetres to a centimetre. Also, the separation between bunches is of order 10 cm. The potential outside of the bunch will fall to zero after some distance in the longitudinal direction. In the transverse direction, the radius of wall is typically of order centimetres, and the potential is set to zero by grounding the conducting wall of the pipe. In regard to accelerator design, the wall radius is chosen to balance cost and risk. The larger the radius, the more expensive the machine will be. However, if the radius is too small, this increases the risk of particle loss and resulting radioactivity.

The electrostatic potential generated by the charged particles follows from the Poisson equation. For the round pipe, we write this equation in cylindrical coordinates as:

$$\frac{\partial^2 \phi}{\partial r^2} + \frac{1}{r} \frac{\partial \phi}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \phi}{\partial \theta^2} + \frac{\partial^2 \phi}{\partial z^2} = -\frac{\rho}{\epsilon_0}, \quad (1)$$

where ϕ is the electrostatic potential, ρ is the charge density, and ϵ_0 is the permittivity of vacuum. The boundary conditions for the above Poisson's equation are:

$$\phi(r = a, \theta, z) = 0, \quad (2)$$

$$\phi(r, \theta + 2\pi, z) = \phi(r, \theta, z), \quad (3)$$

$$\phi(r, \theta, z = \pm\infty) = 0, \quad (4)$$

where a is the radius of the conducting pipe.

The periodic boundary condition for the potential along the θ direction suggests the use of a complex exponential eigenfunction in that direction. A Bessel function is the appropriate eigenfunction in the radial direction. Hence, we can expand the potential ϕ and source term ρ as follows,

$$\rho(r, \theta, z) = \sum_{m=-N_m/2}^{N_m/2-1} \sum_{l=1}^{N_l} \rho^{lm}(z) J_m(\gamma_{lm} r) \exp(-im\theta), \quad (5)$$

$$\phi(r, \theta, z) = \sum_{m=-N_m/2}^{N_m/2-1} \sum_{l=1}^{N_l} \phi^{lm}(z) J_m(\gamma_{lm} r) \exp(-im\theta), \quad (6)$$

where γ_{lm} is a solution of

$$J_m(\gamma_{lm} a) = 0. \quad (7)$$

The $\rho^{lm}(z)$ and $\phi^{lm}(z)$ are determined from

$$\rho^{lm}(z) = \frac{1}{\pi a^2 J_m^2(\gamma_{lm} a)} \int_0^{2\pi} \int_0^a \rho(r, \theta, z) \exp(im\theta) r J_m(\gamma_{lm} r) dr d\theta \quad (8)$$

$$\phi^{lm}(z) = \frac{1}{\pi a^2 J_m^2(\gamma_{lm} a)} \int_0^{2\pi} \int_0^a \phi(r, \theta, z) \exp(im\theta) r J_m(\gamma_{lm} r) dr d\theta. \quad (9)$$

Multiplying Eq. (1) by $\exp(im\theta) r J_m(\gamma_{lm} r)$ and integrating from 0 to 2π and 0 to a , we obtain

$$\frac{\partial^2 \phi^{lm}(z)}{\partial z^2} - \gamma_{lm}^2 \phi^{lm}(z) = -\frac{\rho^{lm}(z)}{\epsilon_0}. \quad (10)$$

Assuming that the length of the bunch is z_N , for the region outside the beam we have

$$\frac{\partial^2 \phi^{lm}(z)}{\partial z^2} - \gamma_{lm}^2 \phi^{lm}(z) = 0. \quad (11)$$

Using the boundary conditions along z , the above equation has a solution of the form,

$$\phi^{lm}(z) = \exp(-\gamma_{lm} z), \quad z > z_N, \quad (12)$$

$$\phi^{lm}(z) = \exp(\gamma_{lm} z), \quad z < 0. \quad (13)$$

Inside the bunch, using a central finite difference scheme yields

$$\frac{\phi_{n+1}^{lm} - 2\phi_n^{lm} + \phi_{n-1}^{lm}}{h_z^2} - \gamma_{lm}^2 \phi_n^{lm} = -\frac{\rho_n^{lm}}{\epsilon_0}, \tag{14}$$

where $n = 0, 1, 2, 3, \dots, N$ and h_z is the mesh size in the z direction. This is a group of $N + 1$ linear algebraic equations with $N + 3$ unknown variables for each mode l and m . However, on both ends of the bunch, the potentials have to be continuous. Using Eqs. (12), (13), we have

$$\phi_{-1}^{lm} = \exp(-\gamma_{lm}h_z)\phi_0^{lm}, \quad n = 0, \tag{15}$$

$$\phi_{N+1}^{lm} = \exp(-\gamma_{lm}h_z)\phi_N^{lm}, \quad n = N. \tag{16}$$

After substituting the above equations into Eq. (14), we obtain a group of tridiagonal linear algebraic equations for $n = 0, 1, \dots, N$, which can be solved using a Gaussian elimination method in a time $O(N)$.

For a rectangular conducting pipe, the Poisson equation is written in Cartesian coordinates as

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = -\frac{\rho}{\epsilon_0} \tag{17}$$

with boundary conditions

$$\phi(x = 0, y, z) = 0, \tag{18}$$

$$\phi(x = a, y, z) = 0, \tag{19}$$

$$\phi(x, y = 0, z) = 0, \tag{20}$$

$$\phi(x, y = b, z) = 0, \tag{21}$$

$$\phi(x, y, z = \pm\infty) = 0, \tag{22}$$

where a is the horizontal width of the pipe and b is the vertical width of the pipe. Expanding the source term ρ and potential ϕ using two sine functions yields

$$\rho(x, y, z) = \sum_{l=1}^{N_l} \sum_{m=1}^{N_m} \rho^{lm}(z) \sin(\alpha_l x) \sin(\beta_m y), \tag{23}$$

$$\phi(x, y, z) = \sum_{l=1}^{N_l} \sum_{m=1}^{N_m} \phi^{lm}(z) \sin(\alpha_l x) \sin(\beta_m y), \tag{24}$$

where

$$\rho^{lm}(z) = \frac{4}{ab} \int_0^a \int_0^b \rho(x, y, z) \sin(\alpha_l x) \sin(\beta_m y), \tag{25}$$

$$\phi^{lm}(z) = \frac{4}{ab} \int_0^a \int_0^b \phi(x, y, z) \sin(\alpha_l x) \sin(\beta_m y), \tag{26}$$

where $\alpha_l = l\pi/a$ and $\beta_m = m\pi/b$. After substituting these equations into Poisson's equation, we obtain

$$\frac{\partial^2 \phi^{lm}(z)}{\partial z^2} - \gamma_{lm}^2 \phi^{lm}(z) = -\frac{\rho^{lm}(z)}{\epsilon_0}, \tag{27}$$

where $\gamma_{lm}^2 = \alpha_l^2 + \beta_m^2$. The potential inside the bunch can be found following the same procedure as for the round pipe by matching the solution of the potential inside and outside the bunch. The resulting finite difference equations inside the bunch are then solved using Gaussian elimination.

3. Parallel implementation

A two-dimensional domain-decomposition approach is employed to implement the above algorithm on high performance computers. A schematic plot of the two-dimensional decomposition on a θ - z plane for the round pipe and y - z plane for the rectangular pipe is shown in Fig. 1. The solid grid lines define the computational domain grids. The dashed lines define the local computational domain on each processor. The physical computational domain is defined as a 3-dimensional box with range $0 \leq r \leq a$, $0 \leq \theta \leq 2\pi$, and $0 \leq z \leq z_N$. Here, a is the radius of the round pipe and z_N is the length of the bunch. The domain is decomposed on the θ - z plane into a number of small rectangular blocks. These blocks are mapped to a logical two-dimensional Cartesian processor grid. Each processor contains one rectangular block domain. Since all the grids used here are uniform in all three dimensions, each processor will contain an equal volume block to achieve a good load balance. The local range of a block on a single processor is defined as $0 \leq r \leq a$, $\text{idpy}\Delta\theta \leq \theta \leq (\text{idpy} + 1)\Delta\theta$, and $\text{idpz}\Delta z \leq z \leq (\text{idpz} + 1)\Delta z$. Here, idpy and idpz are the processor coordinates in the Cartesian processor grid. The quantities $\Delta\theta$ and Δz are given by $\Delta\theta = 2\pi/N\text{proc}_y$, $\Delta z = z_N/N\text{proc}_z$, where $N\text{proc}_y$ and $N\text{proc}_z$ are the number of processors along the y - and z -directions, respectively, in the two-dimensional processor grid. The local number of computational grid points along each dimension on a single processor is defined as:

$$Nr_{\text{local}} = \text{int}[a/h_r] + 1, \quad (28)$$

$$N\theta_{\text{local}} = \text{int}[(\text{idpy} + 1)\Delta\theta/h_\theta] - \text{int}[\text{idpy}\Delta\theta/h_\theta] + 1, \quad (29)$$

$$Nz_{\text{local}} = \text{int}[(\text{idpz} + 1)\Delta z/h_z] - \text{int}[\text{idpz}\Delta z/h_z] + 1, \quad (30)$$

where h_r , h_θ , and h_z are the mesh sizes along the r -, θ - and z -directions, respectively. For the processor containing the starting grid in the global mesh, there is one more grid point along the θ - and z -directions.

After mapping the physical domain onto processors, computation can be done on each processor. Since all the mesh points which store the coefficients of expansion polynomials along r in the round pipe and along x in rectangular pipe are assigned to a single processor, the Bessel function expansion and the sine function expansion in these directions can be done simultaneously for all processors. Communication is required to move the data among processors for the Fourier transformation along the θ - or y -direction. In the forward Gaussian elimination, the processor below has to wait until the above processor finishes its calculation and sends the data to it. To avoid this situation, we define a global transpose operation. Whenever the Fourier transform

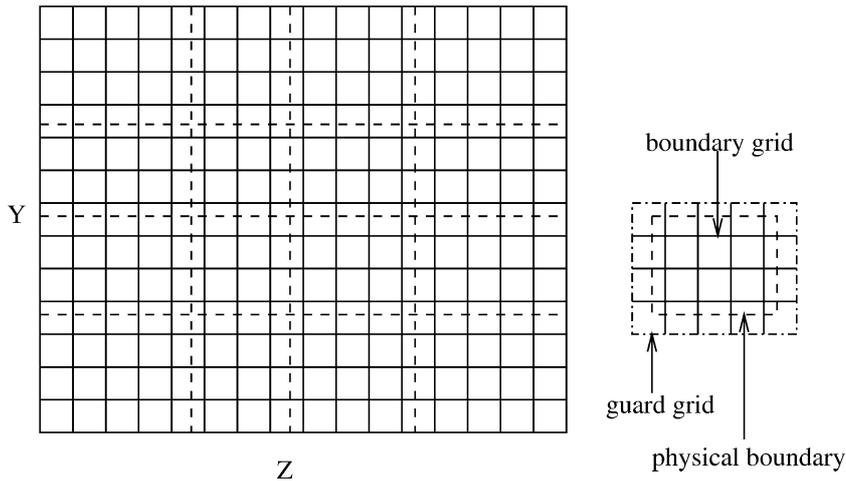


Fig. 1. A schematic plot of two-dimensional decomposition on θ - z or y - z domain.

or Gaussian elimination are required in one dimension, that dimension is transposed with the serial r or x dimension so that each processor contains all the mesh points along that dimension. Then, computation can be done on all processor simultaneously. The transpose will move the data among processors and requires an all-to-all communication. The global all-to-all communication used here to solve the tridiagonal system can be avoided by using a parallel tridiagonal solver [16–19]. In the parallel tridiagonal solver, the original tridiagonal system is partitioned into a number of subsystems and distributed among processors. Then the subsystems on each processor are reduced in parallel into N shaped matrices. The reduced subsystems are solved for the unknowns on the boundary of each processor. The number of equations for the unknowns is proportional to the number of processors. Interprocessor communication is required to solve the reduced system. However, it can be significantly reduced using a folded skip-decoupling scheme described by Lambert et al. [19]. Using the substructured techniques, the bottleneck associated with the serial Gaussian elimination can be minimized in the parallel domain-decomposed implementation. We expect this will improve the performance of our 3D Poisson solver proposed in this paper.

The performance of the code was tested on the Cray T3E-900, the SGI Origin 2000 and the IBM SP RS/6000. Fig. 2 shows the execution time as a function of processor number on the three machines. The fixed size of the problem is $129 \times 129 \times 129$. Good scalability is seen on all three machines. The performance of the code on the Origin is better than that on the T3E and SP machines. This may be due to the faster clock speed of the Origin (250 MHz) than the SP (200 MHz) and to the larger cache size of the Origin (4 MB) than the T3E machine (~ 100 KB). Another reason for slower performance on T3E machine can be due to the default double precision running on this machine. To understand the details of the computational cost in the Poisson solver, we also measured the time spent on each dimensional operation, i.e. Bessel expansion, Fourier transformation, and Gaussian elimination, as a function of processor number for the fixed problem size on the Origin. The results are given in Fig. 3. The Bessel function expansion takes more than 80% of the total computational time due to the $O(N^2)$ cost of the matrix–vector multiplication. However, it has very good scalability with increasing number of processors since all the computation is done locally. The Fourier transform and Gaussian elimination are much faster than the Bessel expansion but have poorer scalability. This is due to the transpose used in these operations,

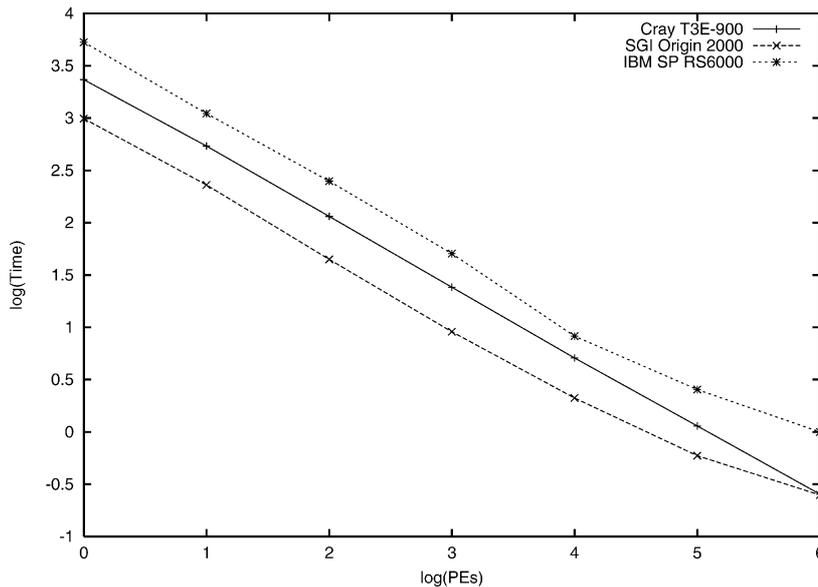


Fig. 2. The log base 2 time cost as function of log base 2 number of processors on Cray T3E and SGI Origin and IBM SP for cylindrical case.

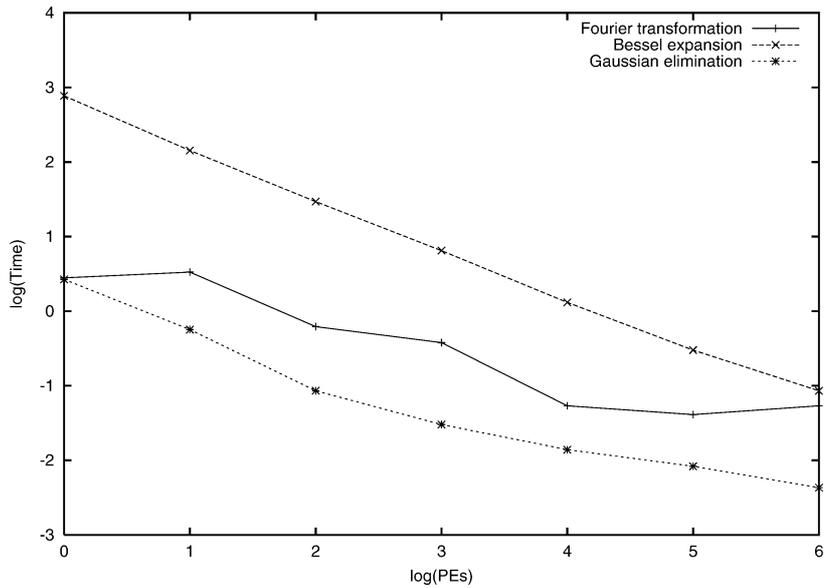


Fig. 3. The log base 2 time cost of three major components, Fourier transformation, Bessel expansion, and Gaussian elimination, as a function of log base 2 number of processors on SGI Origin for cylindrical case.

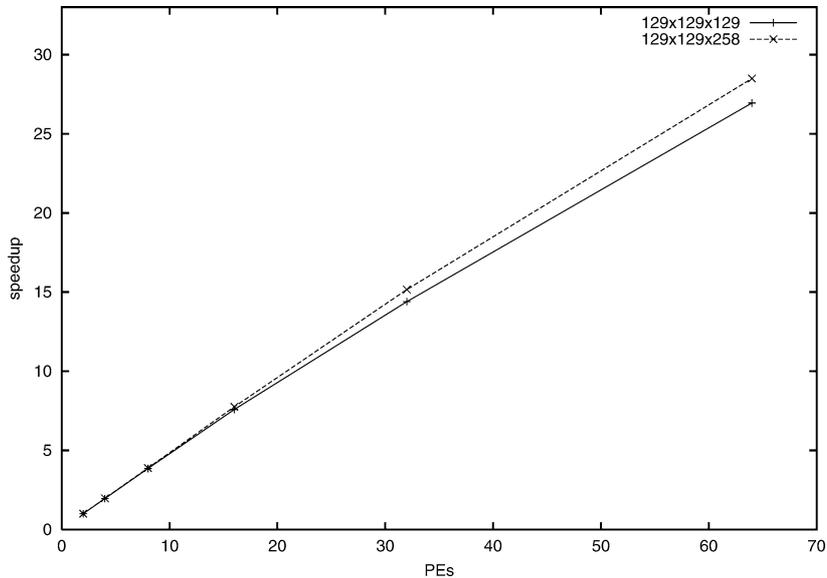


Fig. 4. The speedup as a function of processor number on Cray T3E with different problem size for cylindrical case.

which requires an all-to-all global communication. Fig. 4 shows a comparison of speedup as function of the number of processors for two different problem sizes, $129 \times 129 \times 129$ and $129 \times 129 \times 258$. It is seen that with increasing problem size, the speedup also increases. This shows that the code is more efficient with a larger problem size.

4. Application

An example has been used to test this solver with an analytical solution of Poisson’s equation for an axisymmetric beam in a round pipe. The charge density function is assumed to be given by

$$\rho(r, z) = \begin{cases} \exp(-10(r + z)^2), & r \leq 0.5a, 0.0 \leq z \leq z_N, \\ 0.0, & r > 0.5a. \end{cases} \tag{31}$$

The analytical solution for this problem is given by

$$\phi(r, z) = \frac{1}{a\epsilon_0} \sum_{n=1}^{n=\infty} \frac{J_0(\beta_n r)}{\alpha_n J_1^2(\alpha_n)} \int_{-\infty}^{+\infty} \int_0^{\infty} J_0(\beta_n r_s) \exp(-\beta_n |z - z_s|) \rho(r_s, z_s) r_s dr_s dz_s, \tag{32}$$

where $J_0(\alpha_n) = 0$ and $\beta_n a = \alpha_n$ [20]. Fig. 5(a) shows the potential as a function of r at three different longitudinal locations, $z = 0.0, z = 0.3z_N, z = 0.76z_N$, from the numerical calculation and from the analytical solutions. Fig. 5(b) shows the comparison of the potential as a function of z at three different radial locations, $r = 0.016a, r = 0.3a, r = 0.61a$. Here, the radius $a = 0.1027$ m and the beam bunch length $z_N = 0.1025$ m. The computation grid number used here is $65 \times 7 \times 65$. It is seen that the numerical solution from the above algorithm and the analytical solution agree with each other quite well. The slight differences in Fig. 5(b) are due to the finite numerical resolution of the grid. By doubling the radial grid number from 65 to 129, the maximum error in this figure is reduced by a factor of 7, which is within 0.2% of agreement.

We also tested the above Poisson solver for the rectangular pipe using the analytical solution from a Green’s function method. The analytical solution of the potential is

$$\phi(x, y, z) = \frac{2}{ab\epsilon_0} \sum_{l=1}^{\infty} \sum_{m=1}^{\infty} \frac{1}{\gamma_{lm}} \sin(\alpha_l x) \sin(\beta_m y) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sin(\alpha_l x_s) \sin(\beta_m y_s) \times \exp(-\gamma_{lm} |z - z_s|) \rho(x_s, y_s, z_s) dx_s dy_s dz_s. \tag{33}$$

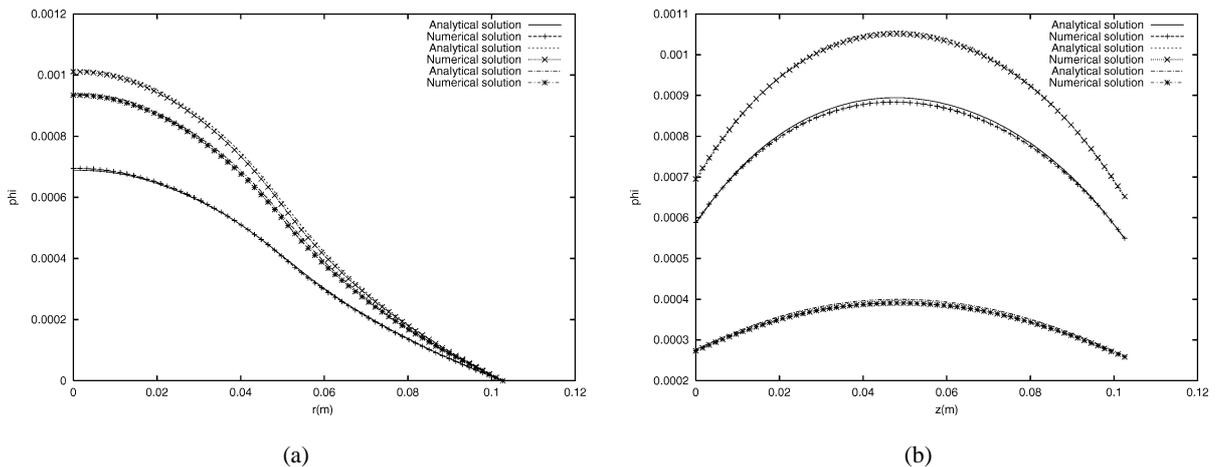


Fig. 5. The potential in the round conducting pipe as a function of (a) r at three locations of $z, z = 0, z = 0.031$ m, $z = 0.078$ m, (b) z at three locations of $r, r = 0.0016$ m, $r = 0.031$ m, $r = 0.063$ m from both numerical and analytical solutions.

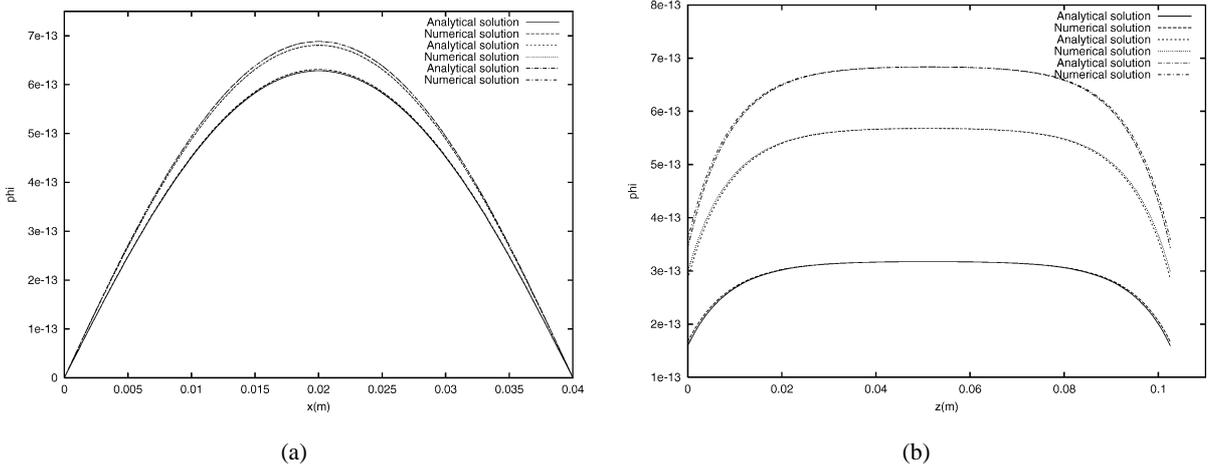


Fig. 6. The potential in the rectangular conducting pipe as a function of (a) x with $y = 0.00062$ m at different locations of z , $z = 0.015$ m, $z = 0.031$ m, $z = 0.047$ m, (b) z with $y = 0.00062$ m at different locations of x , $x = 0.006$ m, $x = 0.012$ m, $x = 0.018$ m from numerical and analytical solutions.

The charge density is assumed to have the distribution

$$\rho(x, y, z) = \begin{cases} \left[\frac{1}{4}a^2 - \left(x - \frac{a}{2}\right)^2 \right] \left[\frac{1}{4}b^2 - \left(y - \frac{b}{2}\right)^2 \right], & 0.0 \leq z \leq z_N, \\ 0.0, & \text{otherwise,} \end{cases} \quad (34)$$

where the aperture size $a = b = 0.04$ m and longitudinal bunch length $z_N = 0.1025$ m. Fig. 6(a) shows the potential as a function of x with $y = 0.016b$ at $z = 0.15z_N$, $z = 0.30z_N$, $z = 0.46z_N$ from the analytical solution and from the numerical computation. Fig. 6(b) shows the potential as a function of z with $y = 0.016b$ at different $x = 0.15a$, $x = 0.30a$, $x = 0.46a$ from the analytical and numerical solutions. It is seen that both solutions are in good agreement in both the x - and z -directions. The computation grid used in this case is $129 \times 129 \times 129$. This gives maximum 5% discrepancy with the largest errors near the two ends of beam bunch. By increasing the longitudinal grid number to 257, the maximum error is reduced by a factor of 2. Further increasing the longitudinal grid number to 513 gives the agreement within 1%. In this example, the largest errors are near the ends of beam bunch, whereas in the example of round pipe, the largest errors are near the middle of bunch. This is due to the fact that the charge density in this example has a step function distribution along z whereas the density density function in the previous example has a gradually longitudinal variation. The sharp density edges in this example also require more longitudinal numerical grids to be used to resolve the potential distribution around the edges.

As an application, we have incorporated the parallel Poisson solver into a parallel particle-in-cell beam dynamics code to study beam halo formation in an upcoming experiment [21]. In the simulation, the initial distribution of particles is transported through a system of 52 alternating gradient quadrupole magnets that provide strong transverse external focusing. Here, the initial distribution has 1 mm centroid offset in x - and y -directions to account for the possible misalignment of transverse focusing. The radius of the conducting pipe is 1.4 cm. The particles moving along the transport system will be subject to the force generated by the external quadrupole magnets and the self-force from the Coulomb interactions among charged particles inside the beam bunch. The self-force (also called space charge force) can be obtained from the solution of three-dimensional Poisson equation using the method described in the previous section. The computation grid number used here is $129 \times 65 \times 129$. A split-operator method has been used to transport the particles [6]. For each time step, the particles are advanced half step using a transfer map corresponding to external field. Then, the space charge force is calculated by solving the three-dimensional Poisson equation subject to external boundary conditions. This force is used to kick the particles

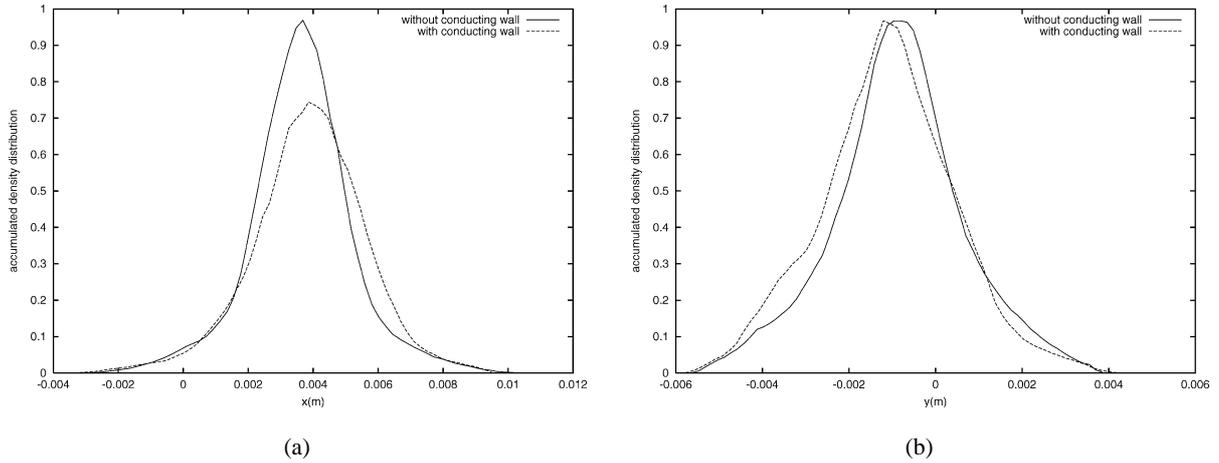


Fig. 7. The accumulated density distribution in the (a) x -, (b) y -directions after quadrupole magnet 49 with and without finite transverse boundary conditions.

for one step in momentum space. The particles then move another half step using the transfer map from external field. Fig. 7 show the accumulated density distribution in both the x - and y -directions after quadrupole magnet number 49 where the measurement will be taken. The particle distributions obtained from the Poisson solver with open transverse boundary conditions (i.e. no finite wall) are more peaked than the ones with a finite conducting wall boundary condition. The broader distribution from the conducting wall is due to the image charge effects on the conducting wall. These results show that it will be important to include the transverse finite boundary condition in order to accurately model the beam dynamics in the upcoming halo experiment.

5. Conclusions

In this paper, we have presented a parallel 3-dimensional Poisson solver for the electrostatic potential of a charged beam in a round or rectangular conducting pipe with open-end boundary conditions. Instead of solving Poisson's equation on a large domain so that the potential vanishes on the boundary, we solve Poisson's equation in a domain containing only the bunch by using an eigenfunction expansion in the transverse direction and a finite difference method in the longitudinal direction with matching conditions on both ends. This method avoids the use of a computational domain outside the beam and hence is more computationally efficient. A parallel implementation using a two-dimensional domain decomposition approach with message passing shows good scalability on the Cray T3E, SGI Origin 2000, and IBM SP3. Tests against analytical solutions for the round pipe and rectangular pipe show excellent agreement. An application to the study of beam halo formation in a planned experiment shows that the inclusion of the conducting pipe in the computer model is important to accurately predict the beam distribution function.

Acknowledgements

We would like to thank Drs. Thomas Wangler and Christopher Allen for the lattice layout of the accelerator transport system used in the application presented above. We would also like to thank Drs. Andreas Adelman, James Billen, Salman Habib, Lloyd Young, and T.F. Wang for helpful discussions. This work was performed on the Cray T3E and IBM SP at the National Energy Research Scientific Computing Center located at Lawrence Berkeley

National Laboratory, and the SGI Origin 2000 at the Advanced Computing Laboratory located at Los Alamos National Laboratory. This work was supported by the U.S. Department of Energy, Office of Science, Division of High Energy and Nuclear Physics, under the project, Advanced Computing for 21st Century Accelerator Science and Technology. This work was also supported by the Division of High Energy Physics through the Los Alamos Accelerator Code Group.

References

- [1] R. Ryne, S. Habib, in: J.J. Bisognano, A.A. Mondelli (Eds.), *Computational Accelerator Physics*, AIP Conference Proceedings, Vol. 391, Woodbury, New York, 1997, p. 377.
- [2] M.E. Jones, B.E. Carlsten, M.J. Schmitt, C.A. Aldrich, E.L. Lindman, *Nucl. Instr. Meth. in Phys. Res. A* 318 (1992) 323.
- [3] A. Friedman, D.P. Grote, I. Haber, *Phys. Fluids B* 4 (1992) 2203.
- [4] R. Ryne (Ed.), *Computational Accelerator Physics*, Los Alamos, NM, AIP Conference Proceedings, Vol. 297, 1993.
- [5] J. Qiang, R.D. Ryne, S. Habib, *Comput. Phys. Comm.* 133 (2000) 18.
- [6] J. Qiang, R.D. Ryne, S. Habib, V. Decyk, *J. Comput. Phys.* 163 (2000) 434.
- [7] D.B. Haidvogel, T. Zang, *J. Comput. Phys.* 30 (1979) 167.
- [8] S. Ohring, *J. Comput. Phys.* 50 (1983) 307.
- [9] H. Dang-Vu, C. Delcarte, *J. Comput. Phys.* 104 (1993) 211.
- [10] E. Braverman, M. Israeli, A. Averbuch, L. Vozovoi, *J. Comput. Phys.* 144 (1998).
- [11] L. Plagne, J. Berthou, *J. Comput. Phys.* 157 (2000) 419.
- [12] O. Buneman, *J. Comput. Phys.* 12 (1973) 124.
- [13] C.K. Birdsall, A.B. Langdon, *Plasma Physics Via Computer Simulation*, McGraw-Hill Book Company, New York, 1985.
- [14] J.W. Eastwood, D.R.K. Brownrigg, *J. Comput. Phys.* 32 (1979) 24.
- [15] R.W. Hockney, J.W. Eastwood, *Computer Simulation Using Particles*, Adam Hilger, New York, 1988.
- [16] H.H. Wang, *ACM Trans. Math. Software* 7 (1981) 170.
- [17] S.P. Kumar, *Int. J. Supercomput. App.* 3 (1989) 75.
- [18] N. Mattor, T.J. Williams, D.H. Hewett, *Parallel Comput.* 21 (1995) 1769.
- [19] M.A. Lambert, G.H. Rodrigue, D.W. Hewett, *Parallel Comput.* 23 (1997) 2041.
- [20] C.K. Allen, N. Brown, M. Reiser, *Particle Accel.* 45 (1994) 149.
- [21] T. Wangler, LEDA beam halo experimental — physics and concept of the experiment, LA-UR-00-3181, 2000.