On FFT–based convolutions and correlations, with application to solving Poisson's equation in an open rectangular pipe*

Robert D. Ryne

Lawrence Berkeley National Laboratory,
1 Cyclotron Road, Berkeley, CA 94720, United States

Abstract

A new method is presented for solving Poisson's equation inside an open-ended rectangular pipe. The method uses Fast Fourier Transforms (FFTs) to perform mixed convolutions and correlations of the charge density with the Green function. Descriptions are provided for algorithms based on the ordinary Green function and for an integrated Green function (IGF). Due to its similarity to the widely used Hockney algorithm for solving Poisson's equation in free space, this capability can be easily implemented in many existing particle-in-cell beam dynamics codes.

Keywords: convolution, correlation, FFT, Poisson equation, Green's function, Hockney method

1. Introduction

The solution of Poisson's equation is an essential component of any self-consistent beam dynamics code that models the transport of intense charged particle beams in accelerators, as well as other plasma particle-in-cell (PIC) codes. If the bunch size is small compared to the transverse size of the beam pipe, the conducting walls are usually neglected. In such cases the method described by Hockney may be employed [1, 2, 3]. In that method, rather than computing N_p^2 point-to-point interactions (where N_p is the number of macroparticles), the potential is instead calculated on a grid of size $(2N)^d$, where N is the number of grid points in each dimension of the physical mesh containing the charge density, and where d is the dimension of the problem

*Work supported by DOE under Contract No. DE-AC02-05CH11231.

(1,2, or 3). Using the Hockney algorithm, the calculation is performed using Fast Fourier Transform (FFT) techniques, with the computational effort scaling as $(2N)^d(log_2 2N)^d$.

When the beam bunch fills a substantial portion of the beam pipe transversely, or when the bunch length is long compared with the pipe transverse size, the conducting boundaries cannot be ignored. Poisson solvers have been developed previously to treat a bunch of charge in an open-ended pipe with various geometries [4, 5]. Another approach is to use a Poisson solver with periodic, Dirichlet, or Neumann boundary conditions on the pipe ends, and to extend the pipe in the simulation to be long enough so that the field is essentially zero there. Here a new algorithm is presented for the open-ended rectangular pipe. The new algorithm is useful for a number of reasons. First, since its structure is essentially identical to the FFT-based free space method, it is straightforward to add this capability to any beam dynamics code that already contains the free space solver. Second, since it is Green-function based, the method does not require modeling the entire transverse pipe cross section, *i.e.*, if the beam was of small transverse extent one could instead model only a small transverse region around the axis. Third, since it is based on convolutions and correlations involving Green functions, the method can use integrated Green function (IGF) techniques which have the potential for higher efficiency and/or accuracy than non-IGF methods [6].

The solution of the Poisson equation, $\nabla^2\phi = -\rho/\epsilon_0$, for the scalar potential, ϕ, due to a charge density, ρ, can be expressed as,

$$\phi(x,y,z) = \int\int\int dx'dy'dz'\rho(x',y',z')G(x,x',y,y',z,z'), \qquad (1)$$

where $G(x,x',y,y',z,z')$ is the Green function, subject to the appropriate boundary conditions, describing the contribution of a source charge at location (x',y',z') to the potential at an observation location (x,y,z). For an isolated distribution of charge this reduces to

$$\phi(x,y,z) = \int\int\int dx'dy'dz'\rho(x',y',z')G(x-x',y-y',z-z'), \qquad (2)$$

where

$$G(u,v,w) = \frac{1}{\sqrt{u^2+v^2+w^2}}. \qquad (3)$$

A simple discretization of Eq. (2) on a Cartesian grid with cell size (h_x, h_y, h_z) leads to,

$$\phi_{i,j,k} = h_x h_y h_z \sum_{i'=1}^{i'_{max}} \sum_{j'=1}^{j'_{max}} \sum_{k'=1}^{k'_{max}} \rho_{i',j',k'} G_{i-i',j-j',k-k'}, \tag{4}$$

where $\rho_{i,j,k}$ and $G_{i-i',j-j',k-k'}$ denote the values of the charge density and the Green function, respectively, defined on the grid.

As is well known [7], FFT's can be used to compute convolutions by appropriate zero padding of the sequences. The proof of this is shown in the Appendix, along with an explanation of the requirements for the zero padding. As a result, the solution of Eq. (4) is then given by

$$\phi_{i,j,k} = h_x h_y h_z \mathcal{F}^{bbb}\{(\mathcal{F}^{fff}\rho_{i,j,k})(\mathcal{F}^{fff}G_{i,j,k})\} \tag{5}$$

where the notation has been introduced that $\mathcal{F}^{fff}$ denotes a forward FFT in all 3 dimensions, and $\mathcal{F}^{bbb}$ denotes a backward FFT in all 3 dimensions. The treatment of the open rectangular pipe relies on the fact that the FFT-based approach works for correlations as well as for convolutions, the only difference being the direction of the FFTs. As an example, consider a case for which the y variable involves a correlation instead of a convolution. Then,

$$\sum_{i'=1}^{\infty} \sum_{j'=1}^{\infty} \sum_{k'=1}^{\infty} \rho_{i',j',k'} G_{i-i',j+j',k-k'} = \mathcal{F}^{bfb}\{(\mathcal{F}^{fff}\rho_{i,j,k})(\mathcal{F}^{fbf}G_{i,j,k})\}. \tag{6}$$

Because of this, and the fact the the Green function for a point charge in an open rectangular pipe is a function of $(x \pm x', y \pm y', z \pm z')$, an FFT-based algorithm follows immediately.

2. Poisson's equation in an open rectangular pipe

The Green function for a point charge in an open rectangular pipe with transverse size $(0, a) \times (0, b)$ is given by,

$$G(x, x', y, y', z, z') = \frac{1}{2\pi ab} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{1}{\kappa_{mn}} \sin\frac{m\pi x}{a} \sin\frac{m\pi x'}{a} \sin\frac{n\pi y}{b} \sin\frac{n\pi y'}{b} e^{-\kappa_{mn}|z-z'|}, \tag{7}$$

where $\kappa_{mn}^2 = (\frac{m\pi}{a})^2 + (\frac{n\pi}{b})^2$. Equivalently, it can be expressed as a function of a single rectangular pipe Green function,

$$\begin{aligned} G &= R(x-x', y-y', z-z') - R(x-x', y+y', z-z') \\ &\quad -R(x+x', y-y', z-z') + R(x+x', y+y', z-z'), \end{aligned} \tag{8}$$

where

$$R(u,v,w) = \frac{1}{2\pi ab} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{1}{\kappa_{mn}} \cos\frac{m\pi u}{a} \cos\frac{n\pi v}{b} e^{-\kappa_{mn}|w|}. \tag{9}$$

It follows that an algorithm for solving the Poisson equation in an open rectangular pipe is given by

$$\begin{aligned} &\phi_{i,j,k}/(h_x h_y h_z) = \\ &\mathcal{F}^{bbb}\{(\mathcal{F}^{fff}\rho_{i,j,k})(\mathcal{F}^{fff}R_{i,j,k})\} - \mathcal{F}^{bfb}\{(\mathcal{F}^{fff}\rho_{i,j,k})(\mathcal{F}^{fbf}R_{i,j,k})\} - \\ &\mathcal{F}^{fbb}\{(\mathcal{F}^{fff}\rho_{i,j,k})(\mathcal{F}^{bff}R_{i,j,k})\} + \mathcal{F}^{ffb}\{(\mathcal{F}^{fff}\rho_{i,j,k})(\mathcal{F}^{bbf}R_{i,j,k})\} \end{aligned} \tag{10}$$

The implementation of this procedure is nearly identical to that of the case of open boundary conditions. At each step of a simulation, the charge density is deposited on a doubled grid, and its forward forward FFT, $(\mathcal{F}^{fff}\rho_{i,j,k})$, is computed. The function R is tabulated within the doubled domain, and four mixed Fourier transforms are computed, namely $(\mathcal{F}^{fff}R_{i,j,k})$, $(\mathcal{F}^{fbf}R_{i,j,k})$, $(\mathcal{F}^{bff}R_{i,j,k})$, and $(\mathcal{F}^{bbf}R_{i,j,k})$. As described by the previous equation, the transformed charged density is multiplied by each of the 4 transformed Green functions, 4 final FFTs are performed, and the results are added to obtain the potential.

Note that, although Eq. (8) contains 4 terms, only a *single* Green function needs to be tabulated on a grid, and it is this tabulated Green function that is transformed in 4 different ways and stored.

3. Integrated Green Function

The discretization, Eq. (4), is a special case of a more general treatment of the problem. To see this, suppose that Eq. (2) is replaced by a sum of elemental integrals,

$$\phi(x,y,z) = \sum_l \phi_l(x,y,z), \tag{11}$$

where the contribution from the l^{th} cell is given, in Cartesian coordinates, by

$$\phi_l(x,y,z) = \int_{x_l}^{x_{l+1}} \int_{y_l}^{y_{l+1}} \int_{z_l}^{z_{l+1}} dx'dy'dz'\rho(x',y',z')G(x-x',y-y',z-z'), \tag{12}$$

where the volume of l^{th} cell corresponds to $\{x_l, x_l+h_x\} \times \{y_l, y_l+h_y\} \times \{z_l, z_l+h_z\}$. How can we accurately approximate the elemental integrals, Eq. (12), that occur in Eq. (11)? Note that Eq. (4) follows from Eq. (11) and Eq. (12) by assuming that

$$\rho(x,y,z) = \delta(x-x_l)\delta(y-y_l)\delta(z-z_l), \tag{13}$$

within a cell. This is the same result that would have been obtained (except for issues at the grid boundary that are usually irrelevant since the charge is usually zero there), by using the trapezoidal rule to approximate Eq. (2). Qiang described a method for computing Eqs. (11) and (12) to arbitrary accuracy using the Newton-Cotes formula [8]. Integrated Green functions (IGF's) provide another means to compute this accurately when certain integrals involving the Green function can be computed analytically [6, 9, 10, 11].

The computation of Eq. (12) can be challenging when the scale over which ρ and G change is disparate. This is because the accuracy of Eq. (4) depends on the product ρG being nearly linear within a cell. Consider a long bunch whose longitudinal density profile, $\rho_z(z)$, changes little over a distance of a few pipe radii. Suppose h_z is chosen to be a few pipe radii, which is sufficient to resolve $\rho_z(z)$ reasonably well. But G falls of exponentially with distance over a few pipe radii. As a result, approximating Eq. (12) using values of ρ and G only at the grid points is likely to be highly inaccurate. But we normally know G everywhere within a cell, not just at the grid points, and, if we assume a simple form for ρ within a cell, then it may be possible to compute Eq. (12) analytically. In this way, IGF's make it so that the requirements on the grid spacing depend only on choosing the grid fine enough to resolve the charge density, *i.e.,* fine enough that the density change is approximately linear inside a cell; there is no requirement on resolving the Green function, since that is handled analytically.

Now suppose that, within a cell bounded by $\{x_l, x_l+h_x\} \times \{y_l, y_l+h_y\} \times \{z_l, z_l+h_z\}$, ρ is approximated by a linear function of z, so that, inside a cell,

$$\rho(x,y,z) \;=\; \frac{1}{h_z^2}\delta(x-x_l)\delta(y-y_l)\left[\rho_{i,j,k}(h_z-z) + \rho_{i,j,k+1}\; z\right], \tag{14}$$

where $\rho_{i,j,k} = \rho(x_l, y_l, z_l)$ and $\rho_{i,j,k+1} = \rho(x_l, y_l, z_l + h_z)$. Now substitute this into Eq. (12), along with the Green function Eq. (9), and combine terms as in Eq. (11). The IGF is found by collecting terms that are the coefficient of $\rho_{i,j,k}$. In this case two adjacent terms in the summation contribute (except at the end points), the result being that the longitudinal dependence of the IGF is given by,

$$\begin{aligned} g_z(z) &= \int_{z_k}^{z_{k+1}} dz' e^{-\kappa_{mn}|z-z'|}(z_{k+1} - z') + \int_{z_{k-1}}^{z_k} dz' e^{-\kappa_{mn}|z-z'|}(z' - z_{k-1}) \\ &= \int_0^{h_z} dz' e^{-\kappa_{mn}|z-z_k-z'|}(h_z - z') + \int_0^{h_z} dz' e^{-\kappa_{mn}|z-z_{k-1}-z'|} z'. \end{aligned} \quad (15)$$

Performing the integrals leads to,

$$g_z(z) = \frac{1}{\kappa_{mn}^2}\Big[2h_z\kappa_{mn}\delta_{z,0} + \left(e^{-\kappa_{mn}|z+h_z|} - 2e^{-\kappa_{mn}|z|} + e^{-\kappa_{mn}|z-h_z|}\right)\Big]. \quad (16)$$

It follows that, in analogy to Eq. (9), the integrated Green function, R_{int}, integrated in just the longitudinal coordinate, for a distribution of charge in an open-ended rectangular pipe is given by,

$$R_{int}(u, v, w) = \frac{1}{2\pi a b h_z^2} \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \frac{1}{\kappa_{mn}} \cos\frac{m\pi u}{a} \cos\frac{n\pi v}{b} g_z(w). \quad (17)$$

4. Numerical Example

As an example consider a rectangular waveguide of full width and full height $a = 4$ cm and $b = 4$ cm. Consider a 3D Gaussian charge distribution with transverse rms sizes $\sigma_x = 0.15a = 6$mm, $\sigma_y = 0.15b = 6$mm, and longitudinal rms size σ_z. Three cases with different rms bunch length will be considered, $\sigma_z = 1.2$ cm, $\sigma_z = 12$ cm, and $\sigma_z = 1.2$ m. The distribution is set to zero at $x^2/\sigma_x^2 + y^2/\sigma_y^2 + z^2/\sigma_z^2 > 3^2$. Fig. 1, left, shows the charge density and Green function as a function of z, down the center of the pipe, for the case $\sigma_z = 1.2$ cm. As shown in the Appendix, the convolution can be performed using FFTs by zero padding the charge density over the domain of the Green function, and by treating the Green function as a periodic function. Fig. 1, right, shows how the charge density and Green function are actually stored in memory, using 256 longitudinal grid: ρ is left in place and zero-padded, and G is circular-shifted so that $G(x = 0, y = 0, z = 0)$ is at the lower bound

of the Green function array. For the case $\sigma_z = 12$ cm, the figure would look similar except that the Green function would be confined to narrow regions at the left and right edges of the plot. For the case $\sigma_z = 1.2$ m, it would be confined to extremely narrow regions at the left and right edges. Figures 2, 3, and 4 show convolution results for the three different bunch

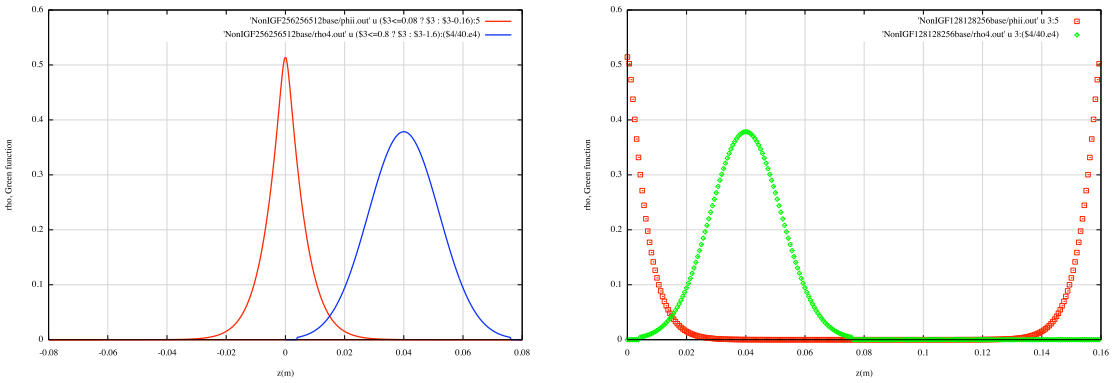

Figure 1: Left: On-axis charge density and Green function vs. z, as they appear in physical space, for the case $\sigma_z = 1.2$ cm. Right: On-axis charge density and Green function as stored in memory.

lengths, $\sigma_z = 1.2$ cm, $\sigma_z = 12$ cm, and $\sigma_z = 1.2$ m, respectively, for grid sizes $64 \times 64 \times 128$ up to $512 \times 512 \times 1024$. The left hand side of each figure shows plots of the potential as a function of z on-axis for various grid sizes, comparing results based on the ordinary Green function and the integrated Green function. The right hand side shows the relative error of the calculated potential. In Fig. 2, σ_z is less than the pipe transverse size (1.2 cm vs. 4 cm); both the ordinary Green function and the IGF are accurate to better than 1% for all the grid sizes shown. In Fig. 3, σ_z is somewhat larger than the pipe transverse size (12 cm vs. 4 cm); when the grid is coarse, the ordinary Green function has signficant errors. In Fig. 4, σ_z is much larger than the pipe transverse size (1.2 m vs. 4 cm); in this case when the grid is coarse the ordinary Green function results exhibit huge errors. As mentioned above, the accuracy of the IGF results is controlled by how well the grid resolves *just* the charge density. For the non-IGF results, the accuracy depends on resolving *both* the charge density and Green function, and, due to the exponential fall-off of the Green function, a coarse grid gives unusable results. The relative error in the potential is plotted on the right hand side of the figures. These were obtained be plotting $(\phi - \phi_{highres})/\phi_{highres})$, where $\phi_{highres}$ is the highest

resolution result, obtained using the IGF with a $512 \times 512 \times 1024$.

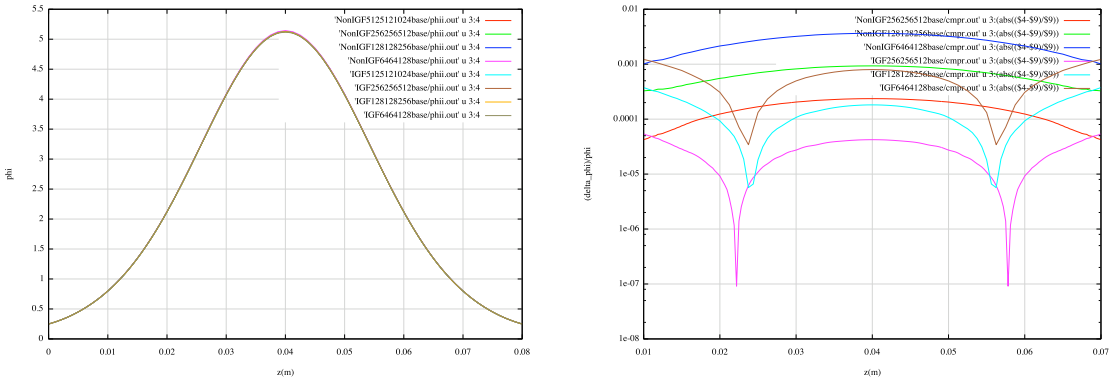

Figure 2: Left: On-axis potential vs. z showing the ordinary Green function result and the Integrated Green function (IGF) result for various grid sizes. The bunch is a Gaussian distribution with $\sigma_x = \sigma_y = 6$mm, $\sigma_z = 1.2$ cm. Right: Relative error of the on-axis potential vs. z for grid sizes $64 \times 64 \times 128$, $128 \times 128 \times 256$, and $256 \times 256 \times 512$.

Fig. 5 shows the ordinary Green function and the IGF $(x = 0, y = 0, z)$ near $z = 0$ for the three cases $\sigma_z = 1.2$ cm, $\sigma_z = 12$ cm, and $\sigma_z = 1.2$ m, all computed using 256 longitudinal grid points. The left hand side shows the ordinary Green function. The exponential fall-off of the Green function is obvious. Note that the ordinary Green function is the *same* for all 3 cases, it is just sampled differently. Note also that for case $\sigma_z = 1.2$m, the grid is so coarse that only the central point is significant, all the others are so far into the region of exponential fall-off that they are effectively zero. If ρ were constant inside of a cell (which is approximately true in the $\sigma_z = 1.2$m case), use of Eq. (4) would correspond to approximating the area under the true Green function with the area under the triangular shaped region of Fig. 5, left, so it is no surprise that it gives wrong results. The right hand side shows the IGF. As before, the exponential fall-off of the Green function is obvious. However, in this case it does not matter that the coarse data poorly resolves the IGF, because the accuracy does not depend on that, it only depends on having a fine enough grid that the charge density is well approximated by a linear function function within a cell. For the $\sigma_z = 1.2$m case, this a approximately true even with as few as 128 grid points, hence the IGF result in Fig. 4 has good accuracy.

Lastly note that, for $\sigma_z = 1.2$m with 256 grid points, the IGF is approximately equal to zero for all but the central point. Hence, Eq. (4) could be

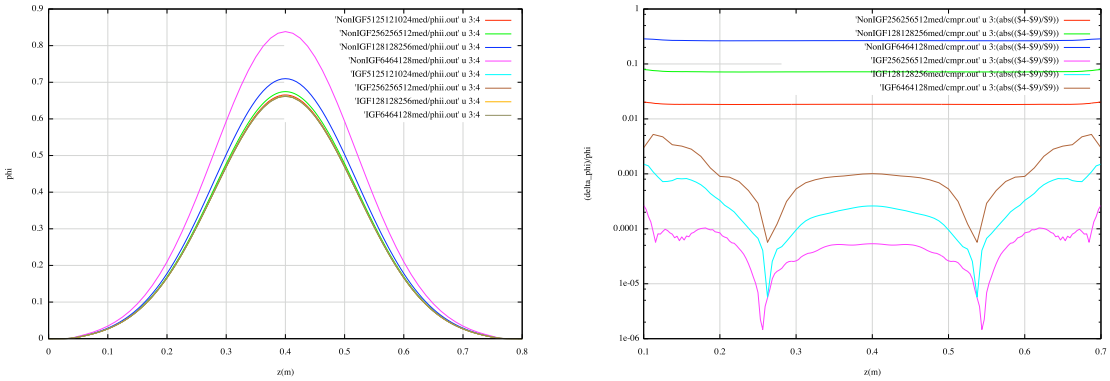

Figure 3: Left: On-axis potential vs. z showing the ordinary Green function result and the IGF result for various grid sizes. The bunch is a Gaussian distribution with $\sigma_x = \sigma_y =$ 6mm, $\sigma_z = 12$ cm. Right: Relative error of the on-axis potential vs. z.

approximated by,

$$\phi_{i,j,k} = h_x h_y h_z \sum_{i'=1}^{i'_{max}} \sum_{j'=1}^{j'_{max}} \rho_{i',j',k} G_{i-i',j-j',0}. \tag{18}$$

In other words, the problem has been converted to one involving multiple 2D convolutions instead of a 3D convolution. The would reduce execution time by a factor equal to $\log_2$ of the number of longitudinal grid points. More generally, depending on the grid size and the exponential fall-off, it would be sufficient compute,

$$\phi_{i,j,k} = h_x h_y h_z \sum_{i'=1}^{i'_{max}} \sum_{j'=1}^{j'_{max}} \sum_{k'=k}^{k \pm N_{neighbors}} \rho_{i',j',k'} G_{i-i',j-j',k-k'}, \tag{19}$$

where $N_{neighbors} = 0$ if the IGF, $G_{i,j,k}$, is appreciable only when $k = 0$, or $N_{neighbors} = 1$ if the IGF is appreciable at $k = 0$ and $k = \pm 1$, or $N_{neighbors} = 2$ if the IGF is appreciable at $k = 0$, $k = \pm 1$ and $k = \pm 2$, *etc.*

5. Discussion and Conclusion

This note does not address the effort require to compute the Green function, Eq. (9). In theory, the calculation of the Green function could make the simulation much more time consuming than the case for isolated boundary

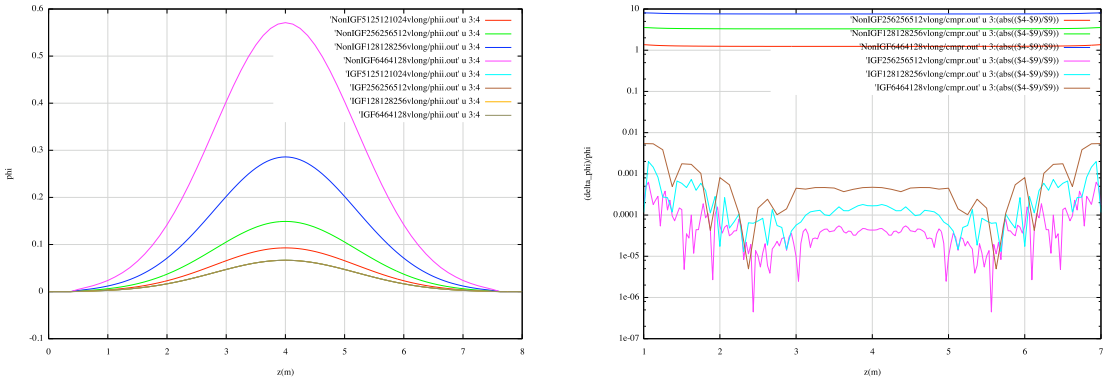

Figure 4: Left: On-axis potential vs. z showing the ordinary Green function result and the IGF result for various grid sizes. The bunch is a Gaussian distribution with $\sigma_x = \sigma_y =$ 6mm, $\sigma_z = 1.2$ m. Right: Relative error of the on-axis potential vs. z.

conditions, but in practice, it might not be since simulations with isolated boundary conditions often re-grid at every time step to take account of the changing transverse beam size. This would not be the case for the rectangular pipe if the beam fills most of the pipe transversely, i.e., the Green function would be computed over the full cross section, transformed, and stored once at the beginning of the simulation. A possible exception would be if the longitudinal extent were changing significantly, but that is not usually the case. In fact, in many applications involving circular accelerators the longitudinal bunch oscillations are very slow. It should also be pointed out that it is not necessary to compute the Green function over the full transverse cross section. The domain on which the Green function is computed need only encompass the full transverse size of the beam. If the beam happened to be very small transversely (but long compared with the transverse pipe dimensions so that the open treatment would be inappropriate), then the transverse physical mesh size could be a small central rectangle centered on the z-axis (symmetrized over the doubled mesh as usual), thus still allowing for high transverse accuracy with a modest number of grid points.

To implement this approach one needs to be able to control the direction of the FFTs in all dimensions. Unfortunately most packages don't offer this capability for built-in multidimensional FFTs. (It was available, for example, in the Connection Machine Scientific Software Library). Such a capability has other uses. For example, when dealing with the Vlasov equation in the

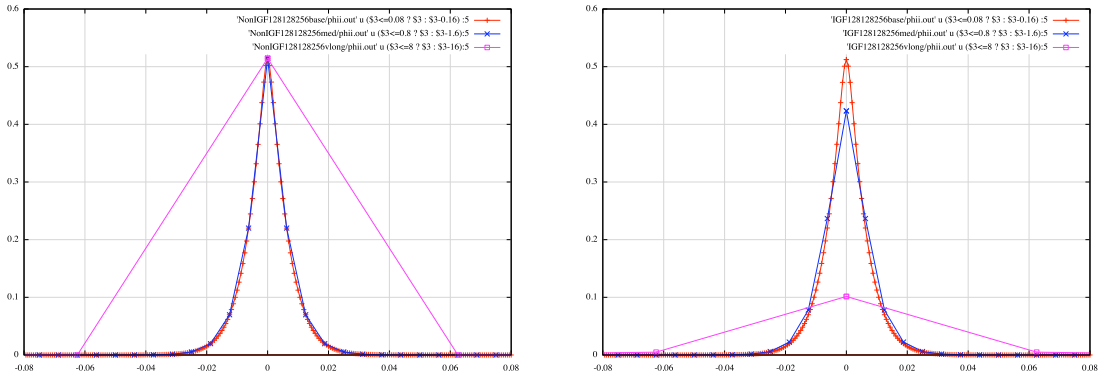

Figure 5: Left: Ordinary Green function vs. z, for three different bunch lengths, using 256 grid points. Right: Integrated Green function vs. z.

form,

$$\frac{\partial f}{\partial t} + (\vec{p}\cdot\partial_{\vec{q}})f - (\nabla\Phi\cdot\partial_{\vec{p}})f = 0, \tag{20}$$

a split-operater, FFT-based method to for advancing the distribution in phase space one time step can be expressed as [12],

$$f(\vec{q},\vec{p},t) = e^{-\frac{t}{2}(\vec{p}\cdot\partial_{\vec{q}})}e^{t(\nabla\Phi\cdot\partial_{\vec{p}})}e^{-\frac{t}{2}(\vec{p}\cdot\partial_{\vec{q}})}f(\vec{q},\vec{p},0). \tag{21}$$

This can be evaluated easily using an FFT package that allows the user to control the FFT direction in each dimension, namely, by alternately performing forward and backward FFT's separately in $\vec{q}$ and $\vec{p}$. Applications such as these argue that FFT packages with built-in multi-dimensional capabilities should allow the user to control the FFT direction in every dimension separately.

In summary, a new method has been presented for solving Poisson's equation in an open-ended rectangular pipe. The only difference in implementation between this case and the isolated case is that a different Green function is used (namely Eq. (17) instead of Eq. (3)), and a procedure is used involving 4 mixed convolutions and correlations instead of a single convolution (namely Eq. (10) instead of Eq. (5)).

Appendix A. FFT-based Convolutions and Correlations

Discrete convolutions arise in solving the Poisson equation in free space, as well as in signal processing. In regard to the Poisson equation, one is typically interested in the following,

$$p_j = \sum_{k=0}^{K-1} r_k x_{j-k} \quad , \quad \begin{array}{l} j = 0, \ldots, J-1 \\ k = 0, \ldots, K-1 \\ j-k = -(K-1), \ldots, J-1 \end{array} \tag{A.1}$$

where x corresponds to the free space Green function, r corresponds to the charge density, and p corresponds to the scalar potential. The sequence $\{p_j\}$ has J elements, $\{r_k\}$ has K elements, and $\{x_m\}$ has $M = J+K-1$ elements.

One can zero-pad the sequences to a length $N \geq M$ and use FFTs to efficiently obtain the $\{p_j\}$ in the unpadded region. To see this, define a zero-padded charge density, ρ,

$$\rho_k = \begin{cases} r_k & \text{if } k = 0, \ldots, K-1 \\ 0 & \text{if } k = K, \ldots, N-1. \end{cases} \tag{A.2}$$

Define a periodic Green function, ξ_m, as follows,

$$\xi_m = \begin{cases} x_m & \text{if } m = -(K-1), \ldots, J-1 \\ 0 & \text{if } m = J, \ldots, N-K, \\ \xi_{m+iN} = \xi_m & \text{for any integer } i \end{cases} \tag{A.3}$$

Now consider the sum

$$\phi_j = \frac{1}{N} \sum_{k=0}^{N-1} W^{-jk} (\sum_{n=0}^{N-1} \rho_n W^{nk})(\sum_{m=0}^{N-1} \xi_m W^{mk}) \qquad 0 \leq j \leq N-1, \tag{A.4}$$

where $W = e^{-2\pi i/N}$. This is just the FFT-based convolution of $\{\rho_k\}$ with $\{\xi_m\}$. Then,

$$\phi_j = \sum_{n=0}^{K-1} \sum_{m=0}^{N-1} r_n \xi_m \frac{1}{N} \sum_{k=0}^{N-1} W^{(m+n-j)k} \qquad 0 \leq j \leq N-1. \tag{A.5}$$

Now use the relation

$$\sum_{k=0}^{N-1} W^{(m+n-j)k} = N \delta_{m+n-j,iN} \qquad (i \text{ an integer}). \tag{A.6}$$

It follows that

$$\phi_j = \sum_{n=0}^{K-1} r_n \xi_{j-n+iN} \qquad 0 \leq j \leq N-1. \tag{A.7}$$

But ξ is periodic with period N. Hence,

$$\phi_j = \sum_{n=0}^{K-1} r_n \xi_{j-n} \qquad 0 \le j \le N-1. \tag{A.8}$$

In the physical (unpadded) region, $j \in [0, J-1]$, so the quantity $j-n$ in Eq. (A.8) satisfies $-(K-1) \le j-n \le J-1$. In other words the values of ξ_{j-n} are identical to x_{j-n}, hence, the FFT-based convolution, Eq. (A.4), matches the convolution in Eq. (A.1).

As stated above, the zero-padded sequences need to have a length $N \ge M$, where M is the number of elements in the Green function sequence $\{x_m\}$. In particular, one can choose $N = M$, in which case the Green function sequence is not padded at all, and only the charge density sequence, $\{r_k\}$, is zero-padded, with $k = 0, \ldots, K-1$ corresponding to the physical region and $k = K, \ldots, M-1$ corresponding to the zero-padded region.

As just shown, FFTs can be used to compute the convolution shown in Eq. (A.1), with no zero padding of the Green function, as long as there are J values of the potential, K values of the charge density, and $J+K-1$ values of the Green function. These conditions are usually satisfied in particle-in-cell beam dynamics code, which have a a grid containing the charge density, and which have a grid containing the Green function (*e.g.*, Eq. (3), Eq. (9), or Eq. (17)) for both positive and negative arguments.

The above FFT-based approach – zero-padding the charge density array, and circular-shifting the Green function – will work in general. In addition, if the Green function is a symmetric function of its arguments, the value at the end of the Green function array (at grid point $J-1$) can be dropped, since it will be recovered implicitly through the symmetry of Eq. (A.3). In that case the approach is identical to the Hockney method [1, 2, 3].

In the area of signal processing, the values of charge density, $\{r_k\}$, correspond to the response function, and the values of the Green function, $\{x_{j-k}\}$ or $\{x_m\}$, correspond to the signal. Based on the above discussion it follows that, if two arbitrary sequences $\{r_0, \ldots, r_{K-1}\}$ and $\{x_0, \ldots, x_{M-1}\}$ are to be convolved to produce a sequence $\{p_0, \ldots, p_{J-1}\}$, and if the number of points in $\{x_m\}$ is less than $J+K-1$, then, in order to use FFTs, the sequence $\{p_j\}$ needs to be zero-padded to extend it to have $J+K-1$ points.

Lastly, note that the above proof that the convolution, Eq. (A.4), is identical to Eq. (A.1) in the un-padded region, works even when W^{-jk} and W^{mk} are replaced by W^{jk} and W^{-mk}, respectively, in Eq. (A.4). In other

words, the FFT-based approach can be used to compute

$$p_j = \sum_{k=0}^{K-1} r_k x_{j+k} \quad , \begin{array}{l} j = 0, \ldots, J-1 \\ k = 0, \ldots, K-1 \\ j-k = -(K-1), \ldots, J-1 \end{array} \tag{A.9}$$

simply by changing the direction of the Fourier transform of the Green function and changing the direction of the final Fourier transform.

References

[1] R. W. Hockney, Methods Comput. Phys. 9, 136-210 (1970).

[2] J. W. Eastwood and D. R. K. Brownrigg, J. Comp. Phys, 32, 24-38 (1979).

[3] R. W. Hockney and J. W. Eastwood, "Computer Simulation using Particles," Taylor & Francis Group (1988).

[4] J. Qiang and R. Ryne, "Parallel 3D Poisson solver for a charged beam in a conducting pipe," Comp. Phys. Comm. 138, 18-28 (2001).

[5] J. Qiang, R. L. Gluckstern, "Three-dimensional Poisson solver for a charged beam with large aspect ratio in a conducting pipe," Comp. Phys. Comm. 160 (2) (2004) 120–128.

[6] D. T. Abell, P. J. Mullowney, K. Paul, V. H. Ranjbar, J. Qiang, R. D. Ryne, "Three-Dimensional Integrated Green Functions for the Poisson Equation," THPAS015, Proc. 2007 Particle Accelerator Conference (2007).

[7] William H. Press , Saul A. Teukolsky , William T. Vetterling , Brian P. Flannery, "Numerical Recipes: The Art of Scientific Computing," Cambridge University Press (2007).

[8] J. Qiang, "A high-order fast method for computing convolution integral with smooth kernel," Comp. Phys. Comm. 18, 313316, (2010).

[9] J. Qiang, S. Lidia, R. Ryne, C. Limborg-Deprey, Phys. Rev. ST Accel. Beams 9, 044204 (2006), and Phys. Rev. ST Accel. Beams 10, 129901(E) (2007).

[10] J. Qiang, "Strong-strong beam-beam simulation of crab cavity compensation at LHC," Particle Accelerator Conference 2009, Vancouver, Canada, 04 - 08 May 2009, pp.WE6PFP038 (2009).

[11] R. D. Ryne et al., "Recent progress on the MaryLie/IMPACT Beam Dynamics Code," Proc. 2006 International Computational Accelerator Physics Conference, Chamonix Mont-Blanc, France, 2-6 Oct 2006. (2006).

[12] R. D. Ryne, S. Habib, and T. P. Wangler, "Halos of Intense Proton Beams," Proc. 1995 Particle Accelerator Conference (1995).