

The fast multipole method in the differential algebra framework

He Zhang*, Martin Berz

Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824-2320, USA

ARTICLE INFO

Available online 21 January 2011

Keywords:

Space charge effect
Fast multipole method
Differential algebra

ABSTRACT

A method is presented that allows the computation of space charge effects of arbitrary and large distributions of particles in an efficient and accurate way based on a variant of the Fast Multipole Method (FMM). It relies on an automatic multigrid-based decomposition of charges in near and far regions and the use of high-order differential algebra methods to obtain decompositions of far fields that lead to an error that scales with a high power of the order. Given an ensemble of N particles, the method allows the computation of the self-fields of all particles on each other with a computational expense that scales as $O(N)$. Using remainder-enhanced DA methods, it is also possible to obtain rigorous estimates of the errors of the methods. Furthermore, the method allows the computation of all high-order multipoles of the space charge fields that are necessary for the computation of high-order transfer maps and all resulting aberrations.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

The Coulomb interaction between the particles inside a bunch is one of the most important collective effects in the study of beam dynamics. This effect becomes more significant when the particle beam has lower energy such as in contemplated time resolved electric microscopes, or has higher density such as in modern high brilliance particle accelerators or free electron laser devices [1,2]. Numerical simulation is usually necessary to study this effect. If the Coulomb interaction is calculated directly via pairwise interaction, the computational time scales with the square of the particle number N , which makes it practically difficult to simulate a large number of particles. More efficient methods, such as the particle–particle interaction (PPI) method, the particle in cell (PIC) method, the tree code and the fast multipole method (FMM), have been put forward and widely used in many simulation codes. The PPI method uses macroparticles, each of which represents a group of real particles. The field on each macroparticle is calculated according to the charge distribution of the whole bunch. The macroparticle representation entails an inevitable modeling error. Furthermore, a specific distribution is assumed, for example an elliptical Gaussian distribution [3], since otherwise one has to use some complicated algorithm to fit the real charge distribution; see for example [4–6]. On the other hand, the PIC method reassigns the charge density onto a previously chosen mesh, solves the Poisson equation to calculate the field on the mesh points, from which the field on

each particle is calculated by the interpolation; see for example [7–9]. The PIC method naturally suffers from computational error due to statistics, and it is very difficult to calculate the derivatives of the field by this method, which are required in the computation of high-order maps by differential algebraic methods.

However, significant breakthroughs have happened outside the beam physics regarding the efficiency of the direct field computation via Coulomb's law. The year 1986 brought the publication of the tree code algorithm, in which the potentials of particles far away from the observer are represented by multipole expansions in powers of $1/r$, covering larger and larger boxes further and further away from the point of interest. It could be shown that the cost of the method scales with $O(N \log N)$ [10,11]. In 1987, the FMM was published, in which the multipole expansions are converted into the local expansions in the near region of the observer, and the efficiency is further increased to $O(N)$ [12,13]. In the original FMM, the three dimensional potential of a point charge is usually expanded in terms of spherical harmonic functions or exponential functions in spherical coordinates around suitable nearby expansion points [14]. In 2006, an accelerated Cartesian expansions FMM algorithm for any potential of the form $r^{-\nu}$ with positive ν were presented [15].

In the following we will present a formulation of the FMM algorithm in the differential algebra (DA) framework. We also use the Cartesian expansions, and we will see that the DA formalism considerably simplifies the necessary algorithms, also for potentials differing from $1/r$, and in particular those associated with common macroparticle approaches for the avoidance of collisions. Furthermore, the method allows the computation of all derivatives of the potential in a natural way up to a predetermined

* Corresponding author.

E-mail address: zhanghe@msu.edu (H. Zhang).

order, and thus allows the computation of high-order transfer maps and aberrations.

2. The algorithm of the FMM

The key idea of the FMM is to represent the potential of suitable groups of source particles that are far away from the observer in terms of expansions involving powers of $1/r$, which we refer to as a far multipole expansion, and utilize that far enough away, higher powers of $1/r$ are less and less significant. This makes it possible to compute the action of source particles on observer particles in a more efficient manner. Furthermore, far multipole expansions corresponding to certain groups can be translated and combined. And finally, far multipoles can also be locally expanded involving powers of r , which we call a local expansion. This local expansion thus allows the treatment of groups of nearby observer particles in a combined manner.

In practice, one first encloses all the particles in a group of cube boxes, the zero level boxes. Then each of these cube boxes is cut into eight equal small cube boxes, which we call the first level boxes. Then each first level box is cut in the same way into eight smaller boxes, leading to the second level boxes. This process is continued until a pre-specified level is reached. In practice, the number of levels is determined such that the average number of particles in the finest boxes is near a pre-specified value, the size of which will affect the efficiency of the method.

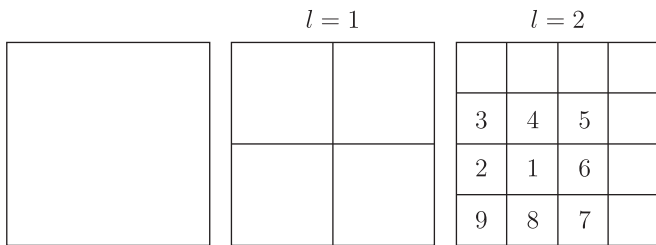


Fig. 1. Sequential cutting of an original box.

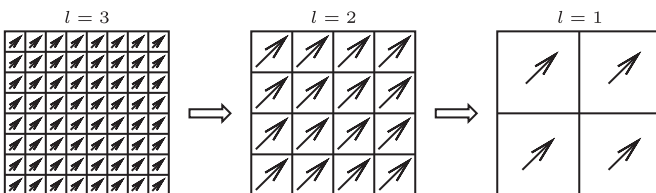


Fig. 2. The propagation of the far multipole expansions.

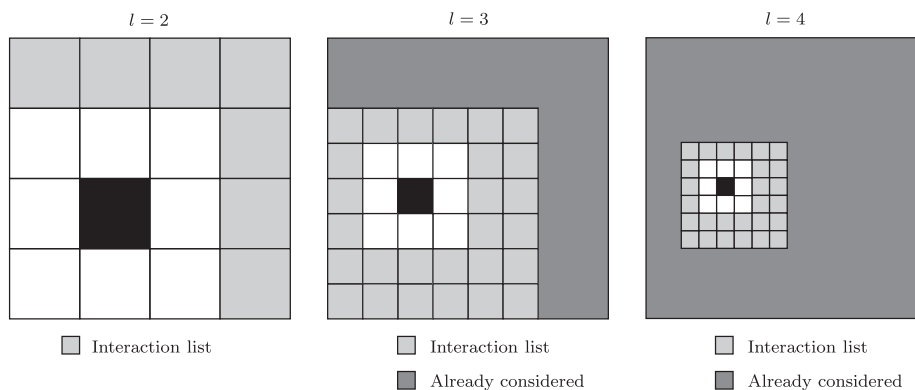


Fig. 3. The behavior of the interaction list of boxes at three levels.

We call boxes of the same level neighbors if they touch. For a given box A, we denote the region made of all same level neighbors and A itself as the near region of A, and everything else as the far region A. For a box A, a next higher level box B containing A is called a parent box of A, and A is called one of the child boxes of B. Apparently each child box has only one parent box, and each parent box has eight child boxes.

A 2D example about how to cut the boxes is shown in Fig. 1. In the first level, the square box is cut into four square boxes, and in the second level, 16 boxes are cut out, and so on. Furthermore, boxes 2 through 9 are all neighbors of box 1, and together they describe the near region of 1. All the unnumbered boxes are in the far region of the box 1. There is no difference between the 2D case and the 3D case in principle, except that each box is cut into four child boxes in the 2D case rather than eight in the 3D case.

The FMM algorithm consists of two parts, the first of which we now describe:

1. Cut the box of interest into the desired levels.
2. In each box of the finest level, calculate the far multipole expansion of the particles inside that box around the center of the box.
3. From the finest level to the second finest level, translate the multipole expansions from the center of the children boxes into the center of their parent box, and then add them up to get the far multipole expansion of the parent box, as Fig. 2 shows.
4. Repeat this at all levels.

After this part is completed, we now have a far multipole expansion for each box in each level.

We now proceed to the second part of the FMM. First, for any box A having a parent B, we define the interaction list to be the collection of those boxes of the same level that belong to the far region of A, but to the near region of its parent B. These boxes of “medium distance” to A will be important in the subsequent algorithm because they require special treatment. To illustrate the concept of the interaction list, consider Fig. 3, which shows a box of interest in black, its near region in white, and its interaction list in light gray, for three different levels.

The second part of the FMM algorithm now begins at the first level at which boxes have interaction lists.

1. For each box with an interaction list, compute a local expansion around its center point from the far expansions of the boxes making up the interaction list.
2. At the next finer level, again compute a local expansion around the center of each box from the far expansion of the boxes making up the interaction list. To this, add the local expansion

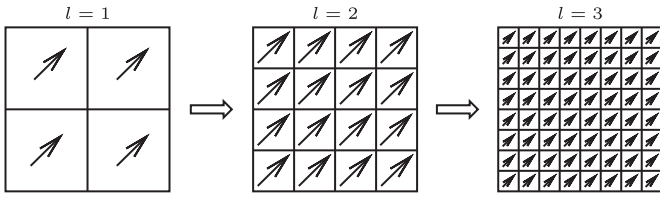


Fig. 4. The translation of the local expansions.

of the parent's interaction list by re-expanding the parent's local expansion to its own center, as shown in Fig. 4.

3. Proceed until the lowest level is reached.

After the completion of the second step, each box at the finest level now possesses a local expansion of all contributions to the potential from outside its near region. The gradient of this local expansion consequently represents the field contributions from everything in the far region.

The third step of the method now consists of determining the field on each particle of interest.

1. For each particle, determine the finest level box to which the particle belongs, and evaluate the local expansion of the field of that box at the particle's coordinates. This results in an approximation of the field at the particle due to all other particles outside the near region of the box in which the particle lies.
2. To the far field, explicitly add the Coulomb field of all other particles in the near region.

We note that the accuracy of this approximation depends on the orders used for the far expansions and the local expansions that occur, and in principle can be made as high as desired.

In the remainder of the paper we will discuss how to perform all the relevant far and local expansions using the differential algebraic methods.

3. A brief review of DA and COSY

Before we continue to present our work on the FMM in the DA framework, we want to give a brief introduction of the DA method. The basic concepts and the deductions will be presented directly without proof, please refer to Ref. [16] for details.

Consider the vector space of the infinitely differentiable functions $C^\infty(R^v)$, in which we can define an equivalence relation “ $=_n$ ” between two functions $a, b \in C^\infty(R^v)$ via $a =_n b$ if $a(0) = b(0)$ and if all the partial derivatives of a and b at 0 agree up to the order n . Note that the point 0 is selected for convenience, and any other point could be chosen as well. The set of all b that satisfies $b =_n a$ is called the equivalence class of a , which is denoted by $[a]_n$. We denote all the equivalence classes with respect to $=_n$ on $C^\infty(R^v)$ as ${}_nD_v$. The addition, scalar multiplication and multiplication on ${}_nD_v$ can be defined as Eq. (1).

$$[a]_n + [b]_n := [a + b]_n$$

$$c \cdot [a]_n := [c \cdot a]_n$$

$$[a]_n \cdot [b]_n := [a \cdot b]_n \quad (1)$$

where $a, b \in {}_nD_v$ and c is a scalar, so that ${}_nD_v$ is an algebra. We can also define the derivation operator ∂_v as Eq. (2)

$$\partial_v [a]_n := \left[\frac{\partial}{\partial x_v} a \right]_{n-1} \quad (2)$$

where x_v is the v th variable of the function a . The operator ∂_v satisfies

$$\partial_v([a] \cdot [b]) = [a] \cdot (\partial_v[b]) + (\partial_v[a]) \cdot [b]. \quad (3)$$

An algebra with a derivation is called a differential algebra. There are v special classes $d_v = [x_v]$, whose elements are all infinitely small. According to the fixed point theorem [16], the inverse and the roots of any element that is not infinitely small in ${}_nD_v$ exist and can be calculated easily. Further more, all real power series can be extended to the DA within their radius of convergence. If a function a in ${}_nD_v$ has all the derivatives $c_{j_1, \dots, j_v} = \partial^{j_1 + \dots + j_v} a / \partial x_1^{j_1} \dots \partial x_v^{j_v}$, then $[a]$ can be written as

$$[a] = \sum c_{j_1, \dots, j_v} \cdot d_1^{j_1} \cdot \dots \cdot d_v^{j_v}. \quad (4)$$

Thus $d_1^{j_1} \cdot \dots \cdot d_v^{j_v}$ is a basis of the vector space of ${}_nD_v$. The Eq. (4) reminds us of the Taylor expansion of a function. Actually if we have a function f in $C^\infty(R^v)$ and f_T is its Taylor expansion up to order n , obviously we have $f =_n f_T$ in ${}_nD_v$. In practice this means we can express f by its Taylor expansion up to an arbitrary order n as an element in ${}_nD_v$, and we can calculate the derivative classes of f and any other function that can be derived by applying the elemental operations, divisions, roots, and power series on f [17].

A beam optical system can be described in terms of the transfer map method, which relates the final positions and velocities of the particles to the initial conditions, so that it makes the tracking more efficient than solving the dynamic equations element by element. The power of the DA make it possible to calculate the map up to any arbitrary order [18].

COSY Infinity's a program developed for high performance modern scientific computing, which is used in beam optical system design. COSY supports various advanced data types, such as DA, TM and VE [19]. In our following work on the FMM, we will use the DA data type. By evaluating a function f in DA data type in COSY, one can get its Taylor expansion f_T up to an arbitrary predetermined order n automatically, because a DA vector carries all its coefficients c_{j_1, \dots, j_v} , which are exactly the coefficients of f_T . If we consider two functions $f, g \in {}_nD_v$ which can also be viewed as two maps M_f and M_g , the function $g(f)$ or the composition of the two maps $M_f \circ M_g$ can be easily calculated by the command POLVAL in COSY. If f is a linear function or a linear map, a more efficient command DATRN can be used [19]. Furthermore, the TM data type can be used for the rigorous calculation. In the future we will use the TM in our algorithm to get rigorous error bounds [20]. COSY also supports parallel calculation. Our current code can be easily revised for the parallel calculation and run in a cluster machine.

4. FMM in the DA framework

4.1. The far multipole expansion

Given a unit point charge located at position \vec{r}_i , its potential at the location \vec{r} of an observer is given by

$$\phi = \frac{1}{|\vec{r} - \vec{r}_i|}. \quad (5)$$

The key idea of the FFM method lies in grouping together particles that are sufficiently near and treat them with a combined expansion. As it turns out, it is not immediately obvious what the best choice of variables is to achieve this end. It is known that using spherical coordinates, ϕ can be expressed as series of the spherical harmonic functions, which are composed of products of $\cos \theta$, $e^{i\psi}$ and $1/r$ [14]. The desire to operate directly in Cartesian coordinates ultimately leads to the inspiration of expanding the function with respect to $1/r$, x/r^2 , y/r^2 , and z/r^2 . As it turns out this

is one variable more than necessary and contains redundancy because (x,y,z) already determine r . But the use of four variables leads to a particularly transparent algorithm.

Back to the FMM, we consider a cube box centered at the origin, whose side length is a , and there are n particles inside the box whose coordinates are (x_i,y_i,z_i) . At any point (x,y,z) , whose distance to the origin $r \gg a$, the potential can be expressed as

$$\begin{aligned} \phi &= \sum_i^n \frac{1}{\sqrt{(x_i-x)^2 + (y_i-y)^2 + (z_i-z)^2}} \\ &= \sum_i^n \frac{1/\sqrt{x^2+y^2+z^2}}{\sqrt{1 + \frac{x_i^2+y_i^2+z_i^2}{x^2+y^2+z^2} - \frac{2x_ix}{x^2+y^2+z^2} - \frac{2y_iy}{x^2+y^2+z^2} - \frac{2z_iz}{x^2+y^2+z^2}}} \\ &= \sum_i^n \frac{d_1}{\sqrt{1 + (x_i^2+y_i^2+z_i^2)d_1^2 - 2x_id_2 - 2y_id_3 - 2z_id_4}} \end{aligned} \quad (6)$$

with

$$\begin{aligned} d_1 &= \frac{1}{\sqrt{x^2+y^2+z^2}} = \frac{1}{r}, \quad d_2 = \frac{x}{x^2+y^2+z^2} = \frac{x}{r^2} \\ d_3 &= \frac{y}{x^2+y^2+z^2} = \frac{y}{r^2}, \quad d_4 = \frac{z}{x^2+y^2+z^2} = \frac{z}{r^2}. \end{aligned}$$

If we consider points (x,y,z) whose distance to the origin $r \gg a$, the expression above can apparently be expanded in powers of the quantities d_1, d_2, d_3 and d_4 . Since these quantities go to zero as r becomes larger and larger, the resulting expansion mathematically represents a Laurent series, and in common physical terminology a multipole expansion. To avoid confusion with the common definition of multipole expansions in particle optics, we call this expansion the far multipole expansion if it is not clear from the context.

Using the variables d_1, d_2, d_3 and d_4 as the expansion variables in the DA algorithm, COSY can now readily compute the far multipole expansions to any order automatically. Furthermore, quite importantly, the same method can be applied to other types of the potential, for example the Coulomb potential in other dimensions, the common van der Waals potentials, or the potentials of certain macroparticle charge distributions, and in the DA formalism, without much additional difficulty far multipole expansions can be calculated by mere evaluation in DA arithmetic.

We now use the formula to calculate the multipole expansions in the finest level boxes. To assess the performance of the method, we performed some test calculations of the far multipole expansion. One group of results are presented in Table 1. We put 50 electrons into each of the two cube boxes, whose centers are $(0,0,0)$ and $(4,0,0)$ and whose side length are both 2. In terms of the terminology introduced above, this means the second box lies in the far region of the first and is hence subject to the far multipole expansion; but it represents the closest such box, where thus the convergence of the method is particularly difficult to achieve.

Table 1
The relative errors of the potential.

DA order	Error
1	2.088×10^{-2}
2	3.816×10^{-3}
3	1.524×10^{-3}
4	9.681×10^{-4}
5	1.381×10^{-4}
6	5.758×10^{-5}

The potential on the electrons in the second box is calculated and repeated for 1000 times with different random numbers, so that we have 50,000 data points together. The relative error is calculated by $\sqrt{\sum_{i=1}^n (\Delta\phi/\phi)^2}$ with $n=50,000$. We can see that as the DA order increases, the relative error decreases, which shows the convergence of this multipole expansion.

4.2. Translation of a multipole expansion

If we have a multipole expansion in the cube box centered at the origin $(0,0,0)$, we can translate it into another frame whose origin is at (x'_o, y'_o, z'_o) . The distance from the observer to the origin of the new frame should be much larger than the side length of the box centered at (x'_o, y'_o, z'_o) , which assures that the new DA variables $d'_1 \dots d'_4$ in Eq. (7) are small enough to still have convergence. Specifically, we have

$$\begin{aligned} d'_1 &= \frac{1}{\sqrt{(x-x'_o)^2 + (y-y'_o)^2 + (z-z'_o)^2}} = \frac{1}{\sqrt{x^2+y^2+z^2}} = \frac{1}{r'} \\ d'_2 &= \frac{x-x'_o}{r'^2} = \frac{x'}{r'^2} \\ d'_3 &= \frac{y-y'_o}{r'^2} = \frac{y'}{r'^2} \\ d'_4 &= \frac{z-z'_o}{r'^2} = \frac{z'}{r'^2} \end{aligned} \quad (7)$$

In Eq. (7), (x',y',z') is the position of the observer in the new frame of (x'_o, y'_o, z'_o) , and r' is the distance from the observer to the new origin (x'_o, y'_o, z'_o) . The new DA variables and the old DA variables in Eq. (6) are related via

$$\begin{aligned} d_1 &= \frac{1}{r} = \frac{1}{\sqrt{(x-x'_o+x'_o)^2 + (y-y'_o+y'_o)^2 + (z-z'_o+z'_o)^2}} \\ &= \frac{1}{\sqrt{(x-x'_o)^2 + (y-y'_o)^2 + (z-z'_o)^2 + x_o'^2 + y_o'^2 + z_o'^2 + 2(x-x'_o)x'_o + 2(y-y'_o)y'_o + 2(z-z'_o)z'_o}} \\ &= \frac{1}{\sqrt{1/d_1'^2 + x_o'^2 + y_o'^2 + z_o'^2 + 2x'_o d'_2/d_1'^2 + 2y'_o d'_3/d_1'^2 + 2z'_o d'_4/d_1'^2}} \\ &= \frac{d'_1}{\sqrt{1 + (x_o'^2 + y_o'^2 + z_o'^2)d_1'^2 + 2x'_o d'_2 + 2y'_o d'_3 + 2z'_o d'_4}} \\ &= d'_1 \cdot \sqrt{R}, \end{aligned} \quad (8)$$

$$d_2 = x d_1^2 = (x-x'_o+x'_o)d_1^2 = (d'_2 + x'_o d'_1) \cdot R$$

$$d_3 = y d_1^2 = (y-y'_o+y'_o)d_1^2 = (d'_3 + y'_o d'_1) \cdot R$$

$$d_4 = z d_1^2 = (z-z'_o+z'_o)d_1^2 = (d'_4 + z'_o d'_1) \cdot R$$

with

$$R = \frac{1}{1 + (x_o'^2 + y_o'^2 + z_o'^2)d_1'^2 + 2x'_o d'_2 + 2y'_o d'_3 + 2z'_o d'_4}.$$

If we substitute Eq. (8) into Eq. (6), we obtain the multipole expansion in the new frame. If one considers Eq. (6) as a map $M_{c2m}(d_1, d_2, d_3, d_4)$ and Eq. (8) as a map $M_1(d'_1, d'_2, d'_3, d'_4)$, the substitution is actually merely the composition of the two maps:

$$M_{m2m} = M_{c2m} \circ M_1 \quad (9)$$

which is a common operation in DA methods, and which here yields the expression of ϕ in the new frame with the new DA variables. With Eqs. (8) and (9), to calculate the multipole expansion in a box not in the finest level, we can translate all

the multipole expansions of all its children boxes to its center, then take the summation.

4.3. Conversion of a far multipole expansion into a local expansion

The potential of a far multipole expansion at the frame at (0,0,0) on an observer (x,y,z) can be converted into a local expansion in the near region of the observer. Since the new frame has its origin (x'_o, y'_o, z'_o) close to the observer, it is natural to choose the coordinates of the observer (x', y', z') in the new frame as the new DA variables, as Eq. (12) shows:

$$\begin{aligned} d'_1 &= x' \\ d'_2 &= y' \\ d'_3 &= z'. \end{aligned} \quad (10)$$

So, in contrast to the far multipole expansion, which corresponds to a Laurent expansion, the local expansion corresponds to a conventional Taylor expansion. The old DA variables in Eq. (6) and the new DA variables in Eq. (12) have the relation as shown in Eq. (11):

$$\begin{aligned} d_1 &= \frac{1}{\sqrt{x^2+y^2+z^2}} = \frac{1}{\sqrt{(x'_o+d'_1)^2+(y'_o+d'_2)^2+(z'_o+d'_3)^2}} = \sqrt{R} \\ d_2 &= \frac{x}{x^2+y^2+z^2} = \frac{x'_o+d'_1}{(x'_o+d'_1)^2+(y'_o+d'_2)^2+(z'_o+d'_3)^2} = (x'_o+d'_1) \cdot R \\ d_3 &= \frac{y}{x^2+y^2+z^2} = \frac{y'_o+d'_2}{(x'_o+d'_1)^2+(y'_o+d'_2)^2+(z'_o+d'_3)^2} = (y'_o+d'_2) \cdot R \\ d_4 &= \frac{z}{x^2+y^2+z^2} = \frac{z'_o+d'_3}{(x'_o+d'_1)^2+(y'_o+d'_2)^2+(z'_o+d'_3)^2} = (z'_o+d'_3) \cdot R. \end{aligned} \quad (11)$$

with

$$R = \frac{1}{(x'_o+d'_1)^2+(y'_o+d'_2)^2+(z'_o+d'_3)^2}.$$

If we call Eq. (11) the map $M_2(d'_1, d'_2, d'_3)$, the local expansion can again be written as a composition of M_{c2m} and M_2 as

$$M_{m2l} = M_{c2m} \circ M_2. \quad (12)$$

Having Eqs. (11) and (12), we can convert the multipoles in the interaction list of each box into the local expansion inside the box itself. The result of a test calculation to show the convergence of the local expansion is presented in Table 2. The set up of the boxes and electrons, the number of the data points, and the definition the relative error are the same with those of Table 1. We can see that the relative error decreases as we increase the DA order.

4.4. Translation of a local expansion

Consider two local frames whose origins are both close to the observer. If we know the local expansion of the frame at (0,0,0) is

Table 2
The relative errors of the potential.

DA order	Error
1	3.136×10^{-2}
2	8.348×10^{-3}
3	3.024×10^{-3}
4	1.165×10^{-3}
5	2.086×10^{-4}
6	8.036×10^{-5}

as Eq. (12) shows, we can translate it into the other local frame at (x'_o, y'_o, z'_o). Since the new frame's origin is also close to the observer, we can still choose the coordinates of the observer in the new frame as the new DA variables d'_1, d'_2 , and d'_3 . Obviously the old DA variables d_1, d_2 , and d_3 , which are the coordinates of the observer in the local frame at (0,0,0), have the simple relation shown in Eq. (13) with the new DA variables:

$$\begin{aligned} d_1 &= x'_o + d'_1 \\ d_2 &= y'_o + d'_2 \\ d_3 &= z'_o + d'_3. \end{aligned} \quad (13)$$

We call Eq. (13) the map M_3 , then the new local expansion can be calculated by composing M_{m2l} with M_3 as

$$M_{l2l} = M_{m2l} \circ M_3. \quad (14)$$

We can translate the local expansion of a box into its children boxes using Eqs. (13) and (14).

4.5. Representation of potential and the field as polynomials

In order to calculate the high-order transfer map of a beam optical system, one needs not only the field on the reference orbit but also the derivatives of the field [18]. If we have the field of the Coulomb interaction and its derivatives, we can include the Coulomb interaction into the map. Eq. (14) provides the expansion of the potential ϕ with respect to the position of the observer up to the order n . This expansion converges inside the corresponding finest box. The constant part of the expansion is the value of the potential at the center of the box, and the coefficient of each monomial is the value of the corresponding partial derivative of the potential.

Table 3 shows an example of the local expansion. In a large cube box, we put 10,000 electrons. The cube box is cut into the third level. And the third level box size is 2, which means if we put a frame at the center of one finest box, we have $x,y,z \in [-1,1]$. The local expansion of each third level box is calculated up to the 10th order, and the coefficients till the third order of one third level box are presented in Table 3. The fourth column and the eighth column show the exponents of each DA variables in each monomial. For example, the index triple 2 1 0 denotes the monomial $x^2y^1z^0 = x^2y$. The second and sixth columns show the coefficient of each monomial. From the table, we observe the decreasing trend of the coefficients according to the increase of the order. To obtain sufficiently meaningful statistics, we repeat the above process for 40 times with different groups of electrons. Each time we find the maximum absolute value of the coefficients for each order and calculate the average absolute value for the coefficients for each order. Then we take the average over the 40 processes and present the result in Fig. 5. The green dots show the maximum value of the coefficients, and the red dots show the average value of the coefficients. It is clear that the coefficients decrease as the order increase, which suggests the convergence of the local expansion.

If we want to calculate the expansion of the Coulomb interaction field on a reference point, we only need to determine which finest box the point belongs to, then by Eqs. (13) and (14) we can translate the local expansion of the potential from the center of the finest box to the reference point. Taking the derivative of the position (x,y,z), we get the expansion of the field (E_x, E_y, E_z) up to the order $n-1$. The expansion of the field only includes the contribution of the electrons in the far region. As to the electrons in the near region, we obtain the extension of their field by representing each of them by Gaussian distribution, which will be discussed in the following.

Table 3
A local expansion.

l	Coefficient	n	Exp.	l	Coefficient	n	Exp.
1	-10173.27230228843	0	0 0 0	11	-.3770785081465575	3	3 0 0
2	77.96202591617063	1	1 0 0	12	0.3065719640561415	3	2 1 0
3	84.83337126367421	1	0 1 0	13	1.430687924061862	3	1 2 0
4	94.59868392563654	1	0 0 1	14	0.1167018056511110	3	0 3 0
5	-3.199969762547595	2	2 0 0	15	-1.761289530718459	3	2 0 1
6	4.505077944915953	2	1 1 0	16	0.7359490919857412	3	1 1 1
7	2.824067476091086	2	0 2 0	17	-1.758649785663027	3	0 2 1
8	1.929372636609234	2	1 0 1	18	-.2994523996221934	3	1 0 2
9	-3.396989424915588	2	0 1 1	19	-.6566773810094656	3	0 1 2
10	0.3759022864564996	2	0 0 2	20	1.173313105460495	3	0 0 3

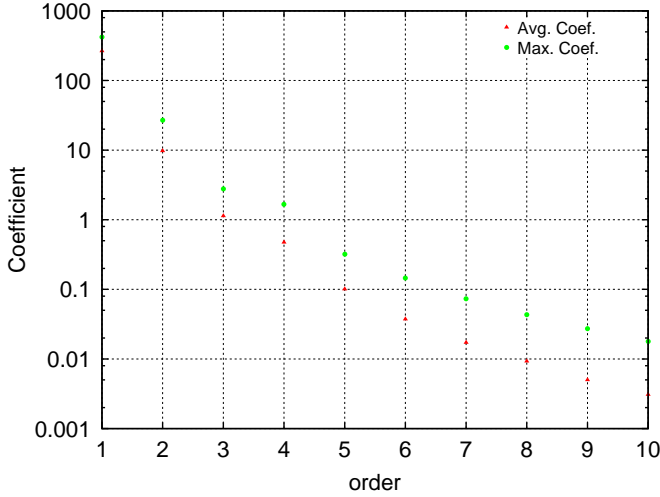


Fig. 5. The coefficients of the different orders of the local expansion. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.6. Gaussian macroparticles instead of point charges

Sometimes it is important to use a finite number of particles to represent a continuous distribution. There are two important cases where this arises; the first is when the number of actual particles is substantially larger than the number than what can practically be handled in a simulation. The second case is the desire to compute a transfer map and aberrations of the system at hand, which requires the knowledge of high-order field expansions near the reference point, which cannot be done reliably if point charges are near. To avoid this, one can use 3D spherical Gaussian distributions instead of point charges as

$$f(x,y,z) = \frac{1}{(\sqrt{2\pi}\sigma)^3} \cdot \exp\left(-\frac{x^2}{\sigma^2} - \frac{y^2}{\sigma^2} - \frac{z^2}{\sigma^2}\right) = \frac{1}{(\sqrt{2\pi}\sigma)^3} \cdot \exp\left(-\frac{r^2}{\sigma^2}\right). \quad (15)$$

It is easy to see from Gauss's law that the field along the radius can be expressed as

$$E_r = \frac{1}{r^2 \sqrt{2\pi}\sigma^3} \left[\sqrt{2\pi} \operatorname{erf}\left(\frac{r}{\sqrt{2}\sigma}\right) \sigma^3 - 2r\sigma^2 \exp\left(-\frac{r^2}{2\sigma^2}\right) \right] \quad (16)$$

where $\operatorname{erf}(r/\sqrt{2}\sigma)$ is the error function. By setting a proper value for σ , we can use the Gaussian distribution in Eq. (15) to represent a point charge. When $r \gg \sigma$, E_r is equal to the field of a point charge. When $r \rightarrow 0$, $E_r \rightarrow 0$. To calculate the expansion of the error function, we need to find all the derivatives of it. Considering the

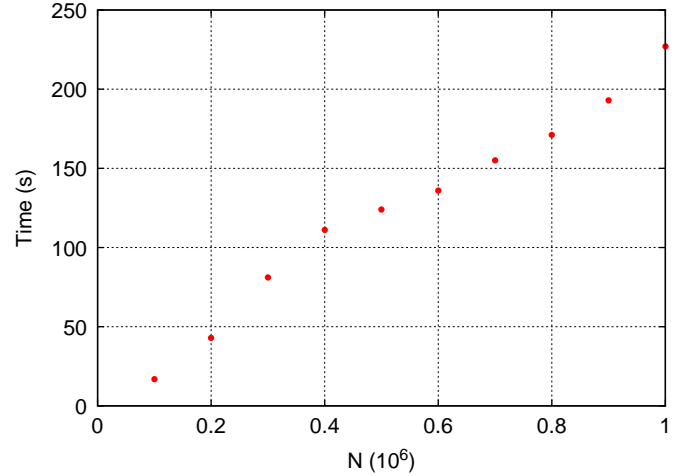


Fig. 6. Computation time for different numbers of electrons.

definition of the error function

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (17)$$

it is easily to determine its derivative

$$\frac{d\operatorname{erf}(x)}{dx} = \frac{2}{\sqrt{\pi}} e^{-x^2}. \quad (18)$$

For any given point x , we calculate $\operatorname{erf}(x)$ by the rational Chebyshev approximation [21] and $d\operatorname{erf}(x)/dx$ by Eq. (18), which is infinitely differentiable and allows the computation of all higher derivatives.

4.7. Example calculations exhibiting the linear scaling property

To show the performance of the method, for reasons of space we limit ourselves here to the most important characteristic, namely the scaling of the computation time to the number of particles. We perform a series of simulations with expansion order 5 and with varying numbers of electrons N . Fig. 6 shows the required computation time for a simulation with varying particle numbers N . We observe the nearly linear behavior, with a slight deviation around $N = 0.4 \times 10^6$, at which point the number of particles crosses a threshold for transitions from a three-level scheme to a four-level scheme, which leads to slightly improved performance after the transition. Overall, the expected linearity is well achieved.

5. Conclusions and future work

We have shown how we apply the FMM in the DA framework to calculate the electrostatic potential. Future work will see the

application to other potentials, the derivation of explicit high-order time stepping schemes, and the direct connection to codes performing map integration. Furthermore, we will develop parallel versions of the algorithms, making use of COSY's support for MPI-type parallelization at the machine level. Finally, we will use the Taylor model data type in COSY to obtain fully rigorous error estimates of the methods.

Acknowledgments

For many fruitful discussions we are indebted to He Huang, Kyoko Makino and Alexander Wittig. Financial support was appreciated from the US Department of Energy under contract number DE-FG02-08ER41546.

References

- [1] G.H. Jansen, Coulomb Interactions in Particle Beams (Advances in Electronics and Electron Physics Supplement 21), Academic Press, 1990.
- [2] P. Kruit, G. Jansen, Space Charge and Statistical Coulomb Effects, second ed., Handbook of Charged Particle Optics, 2009, pp. 341–391.
- [3] M. Martini, M. Prome, Particle Accelerators 2 (1971) 289.
- [4] R.W. Garnett, T.P. Wangler, Space-charge calculation for bunched beams with 3-D ellipsoidal symmetry, in: Proceedings of the 1991 IEEE Particle Accelerator Conference (APS Beam Physics), 1991, p. 330.
- [5] P.M. Lapostolle, A.M. Lombardi, S. Nath, E. Tanke, S. Valero, T.P. Wangler, A new approach to space charge for linac beam dynamics codes, in: Proceedings of the 18th International Linear Accelerator Conference, 1996, p. 375.
- [6] P. Lapostolle, A.M. Lombardi, E. Tanke, S. Valero, R.W. Garnett, T.P. Wangler, Nuclear Instruments and Methods A 379 (1) (1996) 21.
- [7] N. Pichoff, J.M. Lagniel, S. Nath, Simulation results with an alternate 3D space charge routine, PICNIC, in: Proceedings of the XIX International Linac Conference, vol. 141, 1998, pp. 23–28.
- [8] G. Poplauer, U. van Rienen, B. van der Geer, M. de Loos, IEEE Transactions on Magnetics 40 (2 Part 2) (2004) 714.
- [9] Y.K. Batygin, Nuclear Instruments and Methods A 539 (3) (2005) 455.
- [10] J. Barnes, P. Hut, Nature 324 (1986) 4.
- [11] J.E. Barnes, Journal of Computational Physics 87 (1) (1990) 161.
- [12] L. Greengard, V. Rokhlin, Journal of Computational Physics 73 (2) (1987) 325.
- [13] J. Carrier, L. Greengard, V. Rokhlin, SIAM Journal on Scientific and Statistical Computing 9 (1988) 669.
- [14] R.K. Beatson, L. Greengard, Wavelets, Multilevel Methods and Elliptic PDEs (1997) 1.
- [15] B. Shanker, H. Huang, Journal of Computational Physics 226 (1) (2007) 732.
- [16] M. Berz, Modern Map Methods in Particle Beam Physics, Academic Press, San Diego, 1999 Also available at <<http://bt.pa.msu.edu/pub>>.
- [17] M. Berz, Nuclear Instruments and Methods A 298 (1990) 426.
- [18] M. Berz, Nuclear Instruments and Methods 363 (1995) 100.
- [19] M. Berz, K. Makino, COSY INFINITY Version 9.0 programmer's manual, Technical Report MSUHEP-060803, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, see also <<http://cosyinfinity.org>>, 2006.
- [20] M. Berz, AIP Conference Proceedings 405 (1997) 1.
- [21] W.J. Cody, Mathematics of Computation 23 (107) (1969) 631.