Journal of Computational Physics 229 (2010) 6688–6695

Contents lists available at ScienceDirect

Journal of Computational Physics

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

The use of tricubic interpolation with spectral derivatives to integrate particle trajectories in complicated electromagnetic fields

Q.R. Marksteiner *

Los Alamos National Laboratory, ISR-6, MS H851 ISR-6, Los Alamos, NM 87545, United States

ARTICLE INFO

Article history:
Received 28 September 2009
Received in revised form 3 April 2010
Accepted 17 May 2010
Available online 24 May 2010

Keywords:
Electromagnetic
Hermite
Interpolation
Particle follower
Spectral

ABSTRACT

Codes that calculate the trajectories of particles in complicated electromagnetic fields often include spectral methods, which take advantage of the speed of FFTs to rapidly solve for the fields. In this case, Hermite tricubic interpolation can be used to calculate the fields between grid points, with spectral derivates used to determine the interpolation coefficients. This method is extremely accurate, and in the case of electrostatic fields produces an energy conserving force when incorporated into a particle follower code.

Published by Elsevier Inc.

1. Introduction

When the electric or magnetic fields are known only on a discrete set of points, instead of as an analytical function, interpolation is essential for codes which solve for the trajectory of charged particles. There are many different techniques for interpolation, with different advantages and disadvantages. For example, linear interpolation is fast and easy to implement, but has discontinuous first derivatives; while the natural spline can be difficult to calculate, especially on large multidimensional grids, but provides an interpolation which has continuous first and second derivatives (C_2). The B-spline can be calculated rapidly and provides a solution with continuous first and second derivatives, but necessitates smoothing so that the B-spline is not a true interpolation (i.e., the interpolated function $\phi(x)$ is not equal to the given knot values ϕ_i on q_i, $\phi(q_i) \neq \phi_i$).

The cubic Hermite spline provides a true interpolation that is fast, easy to use, and has continuous first derivatives (C_1) [9,12]. This scheme requires knowledge of both the function value, ϕ_i, and its derivative, $\partial\phi_i/\partial q$. Numerical schemes exist to define these derivatives when they are not known. In multidimensional applications the combinations of first derivatives (such as $\partial\phi_{ijk}/\partial x\partial y$ and $\partial\phi_{ijk}/\partial x\partial y\partial z$ in 3 Cartesian dimensions) are also needed [12].

In many particle follower codes, the electric or magnetic fields are calculated using spectral or pseudo-spectral methods, so that there is *a priori* knowledge that the data which needs to be interpreted is band-limited and effectively periodic. Then the ideal interpolated function $\Psi(q)$ exists and is described by Fourier interpolation. Fourier interpolation can be calculated either through an infinite sum of the sinc function multiplied by the sampling points, or by a finite sum of all of the correctly phased trigonometric components, calculated from the FFT.

There are many numerical applications where neither of these Fourier interpretation methods are practical. In most cases it is prohibitively slow to sum over all the trigonometric components, especially in large multidimensional arrays. For sinc

* Tel.: +1 505 665 3780.
E-mail address: qrm@lanl.gov

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2010.05.011

interpolation, the infinite sum must be cut off at some point, so there is a tradeoff between accuracy and numerical bulkiness [13].

Despite the difficulty of using Fourier interpolation to find the value of a function at an arbitrary point, the derivatives of the function, $\partial\Psi(q_i)/\partial q$ can easily and quickly be calculated at the grid values, through Fourier differentiation, giving the necessary set of derivatives $\partial\phi_i/\partial q$ for a cubic Hermite spline. Then the cubic Hermite spline is a true interpolation ($\phi(q_i) = \phi_i$) with continuous first derivatives that accurately approximates the true Fourier interpolation $\Psi(q)$. Furthermore, the algorithm is fast and easy to implement: the partial derivates are calculated rapidly through the FFT, and interpolation of an arbitrary point only involves summing over a small number of Hermite basis functions.

In Section 2, the algorithm is described in more detail for 3 Cartesian dimensions. Section 3 shows an example where the method is used to follow the trajectory of a weakly magnetized ion in the complicated geometry of the Columbia Non-neutral Torus (CNT) [21,19].

The algorithm presented here is in 3 Cartesian dimensions because of the application it was used for. The algorithm can also be used in other numbers of dimensions and coordinate systems. For example, an algorithm has been reported in the literature which uses spectral derivatives on the surface of a sphere to determine bicubic interpolation coefficients, with applications to atmospheric models [3].

2. The algorithm in 3 Cartesian dimensions

2.1. Fourier differentiation

Assume that an equilibrium solver produces an equally spaced Cartesian grid of potential values ϕ_{ijk}. In order to implement the Hermite tricubic interpolation, it is necessary to know the set η^{ijk}_{abc} of eight nodes at each point, given by $\eta^{ijk}_{000} = \phi_{ijk}, \eta^{ijk}_{100} = \partial\phi_{ijk}/\partial x, \eta^{ijk}_{110} = \partial^2\phi_{ijk}/\partial x\partial y$, and $\eta^{ijk}_{111} = \partial^3\phi_{ijk}/\partial x\partial y\partial z$, etc. These derivatives can be found easily using Fourier differentiation.

For a code which uses a spectral method to calculate ϕ_{ijk} the "true" values of $\phi(x,y,z)$ are given by Fourier interpolation. The Fourier interpolation of ϕ (shown in one dimension for simplicity) is defined as

$$\phi_f(q) = \frac{1}{N}\sum_{k=-N/2}^{N/2-1}\phi_k \exp\left(\frac{2\pi i k q}{N\Delta q}\right). \tag{1}$$

Fourier interpolation itself is prohibitively slow to use in a particle follower code, but can be used to rapidly calculate the nodes η^{ijk}_{abc} at each point. The derivative of Eq. (1) is:

$$\frac{\partial\phi(q)}{\partial q} = \frac{2\pi i}{N^2\Delta q}\left[\sum_{k=-N/2}^{N/2-1} k\phi_k \exp\left(\frac{2\pi i k q}{N\Delta q}\right)\right], \tag{2}$$

$$= \frac{2\pi i}{N^2\Delta q}\left\{\sum_{k=0}^{N/2-1} k\phi_k \exp\left(\frac{2\pi i k q}{N\Delta q}\right) + \sum_{k=N/2}^{N-1}(k-N)\phi_k \exp\left[\frac{2\pi i q}{N\Delta q}(k-N)\right]\right\}, \tag{3}$$

where Δq refers to the inter-grid spacing of the discrete points. At the discrete points the derivative is easily described in terms of the IDFT:

$$\frac{\partial\phi_n}{\partial q} = \frac{2\pi i}{N^2\Delta q}\left[\sum_{k=0}^{N/2-1} k\phi_k \exp\left(\frac{2\pi i k n}{N}\right) + \sum_{k=N/2}^{N-1}(k-N)\phi_k \exp\left(\frac{2\pi i k n}{N}\right)\right]. \tag{4}$$

This method of Fourier differentiation can be used to find derivatives in three dimensions, simply by applying Eq. (4) along each chord in the three dimensional grid. The method can be then be repeated, to find higher order derivatives such as $\partial^2\phi_{ijk}/\partial x\partial y$.

2.2. Hermite cubic interpolation

One dimensional Hermite interpolation methods are described in the literature [9,12], but will be briefly reviewed here. Given two knots x_i and x_{i+1}, where the set of four nodes $\eta^i_0 = f(x_i), \eta^{i+1}_0 = f(x_{i+1}), \eta^i_1 = f'(x_i)$ and $\eta^{i+1}_1 = f'(x_{i+1})$ are known, a unique cubic polynomial interpolation exists [9]. This cubic polynomial can be found easily by use of the four Hermite cardinal functions, defined here as $h^p_a(x)$, where $a = 0, 1$ and $p = 0, 1$. These functions are equal to 1 at one of the four nodes, and are equal to 0 at the other three nodes:

$$\begin{aligned}
&h^0_0(x_0) = 1 \quad dh^0_0(x_0)/dx = 0 \quad h^0_0(x_1) = 0 \quad dh^0_0(x_1)/dx = 0,\\
&h^0_1(x_0) = 0 \quad dh^0_1(x_0)/dx = 1 \quad h^0_1(x_1) = 0 \quad dh^0_1(x_1)/dx = 0,\\
&h^1_0(x_0) = 0 \quad dh^1_0(x_0)/dx = 0 \quad h^1_0(x_1) = 1 \quad dh^1_0(x_1)/dx = 0,\\
&h^1_1(x_0) = 0 \quad dh^1_1(x_0)/dx = 0 \quad h^1_1(x_1) = 0 \quad dh^1_1(x_1)/dx = 1.
\end{aligned} \tag{5}$$

For a pair of knots spaced a distance Δ apart, the cardinal functions are:

$$h_0^0 = \frac{2}{\Delta^3}(\Delta/2 + q)(q - \Delta)^2, \tag{6}$$

$$h_0^1 = \frac{-2q^2}{\Delta^3}(q - 3\Delta/2), \tag{7}$$

$$h_1^0 = \frac{q}{\Delta^2}(q - \Delta)^2, \tag{8}$$

$$h_1^1 = \frac{q^2}{\Delta^2}(q - \Delta). \tag{9}$$

These are plotted in Fig. 1. Then the Hermite interpolation of this function is found to be

$$f_i(x) = \sum_{a=0}^{1}\sum_{p=0}^{1} \eta_a^{i+p} h_a^p(x). \tag{10}$$

If a set of function values and first derivatives η_a^i are known, then it is possible to perform Hermite interpolation over each interval, and thus interpolate over the entire function. The interpolated function will be continuous and differentiable everywhere.

The derivatives can be calculated using Fourier differentiation (Eq. (4)). With both the function values and derivatives known, Hermite interpolation gives a rapid interpolation which approximates the Fourier interpolation.

Extending this method to three dimensions is fairly simple. Instead of interpolating between two knots where the first derivatives are known, you interpolate within a cube of eight knots. On each of the eight knots of the cube the set of eight nodes η_{abc}^{ijk} must be known, where $\eta_{000}^{ijk} = \phi_{ijk}, \eta_{100}^{ijk} = \partial\phi_{ijk}/\partial x, \eta_{110}^{ijk} = \partial^2\phi_{ijk}/\partial x\partial y$, and $\eta_{111}^{ijk} = \partial^3\phi_{ijk}/\partial x\partial y\partial z$, etc., for a total of 64 nodes. Only combinations of first derivatives are needed; no second derivatives in the same dimension, such as $\partial^2\phi_{ijk}/\partial x^2$, are needed for this interpolation scheme.

The Hermite cardinal functions in this cube are multiples of three one dimensional Hermite cardinal functions. For example, the function $G_{101}^{011}(x,y,z) = h_1^0(x)h_0^1(y)h_1^1(z)$ defines a cube in the interval $[x_0,x_1]$, $[y_0,y_1]$, $[z_0,z_1]$, which satisfies $\partial^2 G_{101}^{011}(x_0,y_1,z_1)/\partial x\partial z = 1$. The other seven partial derivative combinations, represented by the nodes η_{abc}^{011}, are equal to zero at (x_0,y_1,z_1) for the function $G_{101}^{011}(x,y,z)$. At every other corner point of the cube $[x_0,x_1]$, $[y_0,y_1]$, $[z_0,z_1]$, all eight derivative combinations of $G_{101}^{011}(xyz)$, represented by the nodes η_{abc}^{plf}, equal zero. It can be quickly shown that Hermite interpolation extends to three dimensions in this manner by use of separation of variables.

Then, in any cube $[x_i,x_{i+1}]$, $[y_j,y_{j+1}]$, $[z_k,z_{k+1}]$, where the set of 64 nodes is known, the Hermite tricubic interpolation is given by:

$$f_{ijk}(x,y,z) = \sum_{a=0}^{1}\sum_{b=0}^{1}\sum_{c=0}^{1}\sum_{p=0}^{1}\sum_{l=0}^{1}\sum_{f=0}^{1} \eta_{abc}^{(i+p)(j+l)(k+f)} h_a^p(x)h_b^l(y)h_c^f(z). \tag{11}$$

If the set of eight nodes are known on a grid, η_{abc}^{ijk}, then this method may be used to define a function $f(x,y,z)$ over the entire grid, just as in the one dimensional case.

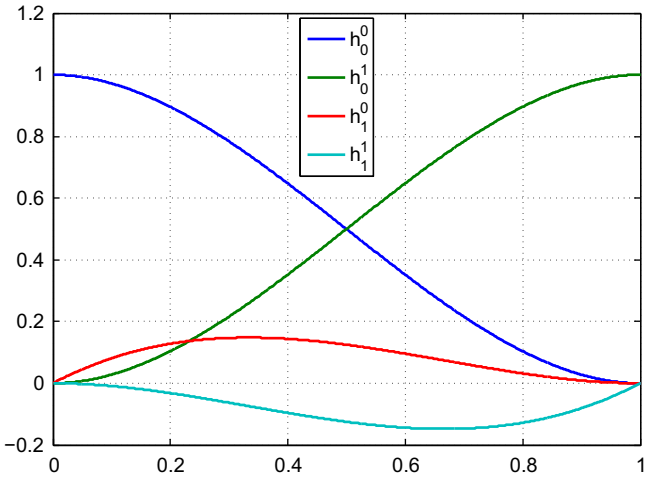

Fig. 1. The set of four Hermite cubic polynomials [9].

The interpolation scheme described here is similar to the one described by Lekien and Marsden [12]. The key differences from [12] are that spectral derivatives are used to calculate the derivates, and the interpolation is calculated directly in terms of the basis functions, which are each multiplied by the eight combinations of derivates at the eight adjacent nodes.

2.3. Proof of C^1 continuity

The function $f(x,y,z)$ is continuous and has continuous first derivatives. To see that this is true, consider the two adjacent cubes V_1 and V_2, joined by the surface S in Fig. 2. Let $f_{ijk}(x,y,z)$ be the Hermite interpolation in V_1, and $f_{(i+1)jk}(x,y,z)$ be the interpolation in V_2, such that $f_{ijk}(x,y,z)$ and $f_{(i+1)jk}(x,y,z)$ are each calculated according to the cardinal functions on their respective eight knots, according to Eq. (11). Let the grid spacing be Δ in the x direction. All of the cardinal functions $G^{plf}_{abc}(x,y,z)$ of $f_{ijk}(x,y,z)$ that describe nodes that are not on S must contain either the term $h^0_0(x)$ or $h^0_1(x)$; these functions go to zero on S. Thus, on S the function $f_{ijk}(x,y,z)$ is described only in terms of the knots that are on S, which all contain either $h^1_0(x)$ or $h^1_1(x)$. Considering only the value $f_{ijk}(x = x_{1+1},y,z)$, and not the derivative $\partial/\partial x$, all values but $h^1_0(x)$ disappear:

$$f_{ijk}(x = x_{1+1}, y, z) = \sum_{b=0}^{1}\sum_{c=0}^{1}\sum_{l=0}^{1}\sum_{f=0}^{1}\eta_{0bc}^{(i+1)(j+l)(k+f)} h^1_0(\Delta) h^l_b(y) h^f_c(z), \tag{12}$$

The same logic that led to Eq. (12) can be applied to describe the interpolation of $f_{(i+1)jk}(x,y,z)$ on S. All of the terms here involve knots on S, and contain $h^0_0(x)$:

$$f_{(i+1)jk}(x = x_{i+1}, y, z) = \sum_{b=0}^{1}\sum_{c=0}^{1}\sum_{l=0}^{1}\sum_{f=0}^{1}\eta_{0bc}^{(i+1)(j+l)(k+f)} h^0_0(0) h^l_b(y) h^f_c(z), \tag{13}$$

Then $f_{ijk}(x = x_{1+1},y,z) = f_{(i+1)jk}(x = x_{1+1},y,z)$, because $h^1_0(\Delta) = h^0_0(0) = 1$, and all other terms in Eqs. (12) and (13) are the same. Thus $f(x,y,z)$, and any combination of derivatives in the y and z directions, are continuous in the entire volume of V_1 and V_2.

On the boundary S, the derivatives $\frac{\partial}{\partial x}[f_{ijk}(x,y,z)]$ will contain only $h^1_1(x)$:

$$\frac{d}{dx}[f_{ijk}(x = x_{1+1}, y, z)] = \sum_{b=0}^{1}\sum_{c=0}^{1}\sum_{l=0}^{1}\sum_{f=0}^{1}\eta_{1bc}^{(i+1)(j+l)(k+f)} h^1_1(\Delta) h^l_b(y) h^f_c(z). \tag{14}$$

Similarly, the derivatives $\frac{d}{dx}[f_{(i+1)jk}(x,y,z)]$ are:

$$\frac{d}{dx}[f_{(i+1)jk}(x = x_{i+1}, y, z)] = \sum_{b=0}^{1}\sum_{c=0}^{1}\sum_{l=0}^{1}\sum_{f=0}^{1}\eta_{1bc}^{(i+1)(j+l)(k+f)} h^0_1(0) h^l_b(y) h^f_c(z). \tag{15}$$

So that $\partial f/\partial x$, and all possible y and z derivatives of $\partial f/\partial x$ are also continuous in the entire volume of V_1 and V_2. Thus the Hermite interpolation, using $f_{ijk}(x,y,z)$ in V_1 and $f_{(i+1)jk}(x,y,z)$ in V_2 has continuous values, first derivatives, and combinations of first derivatives.

This can be extended to Hermite interpolation on an entire grid, where the set of nodes η^{ijk}_{abc} is known. The Hermite interpolation method, as describes above, will have continuous values and first derivatives everywhere. C_1 continuity has also been proven for three dimensional Hermite tricubic interpolation, where the higher derivatives are calculated to ensure maximum smoothness, instead of being spectrally accurate [12].

This type of interpolation is useful for describing conservative forces, such as electrostatic fields. By performing Hermite interpolation on ϕ, and then calculating the electric field by analytically differentiating the interpolation, it is possible to define an electric field that is conservative, continuous, and spectrally accurate.

2.4. Comparison of methods

A simple numerical experiment, identical to the one described by Boyd [13], was done to compare the interpolation method described in this paper with other common interpolation methods. The different interpolation methods were used to

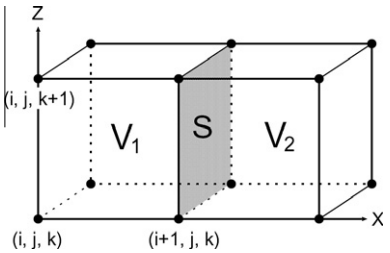

Fig. 2. Two adjacent Hermite interpolation cubes. Function values and derivatives are continuous at the boundary between V_1 and V_2.

interpolate the function cos (kx), and the maximum error was found from varying both the location of the knots (always using an evenly spaced grid), and the interpolated point between the knots. For simplicity, this comparison was done in one dimension. Like in Boyd [13], the cosine function is interpolated for different values of k, and the dimensionless wavenumber $\kappa = k/k_{\text{limit}}$ is used, where k_{limit} is the Nyquist wavenumber. The results are shown in the chart below.

	$\kappa = 1/16$	$\kappa = 1/8$	$\kappa = 1/4$	$\kappa = 1/2$
Hermite FFT	3.87E−6	6.18E−5	9.81E−4	0.0152
Hermite FD	1.23E−4	1.01E−3	9.25E−3	0.116
Polynomial, $M = 1$	4.72E−4	3.75E−3	0.0293	0.212
Polynomial, $M = 2$	3.40E−6	1.07E−4	3.25E−3	0.0829
Polynomial, $M = 3$	2.73E−8	3.41E−6	3.99E−4	0.0354
Polynomial, $M = 4$	2.29E−10	1.14E−7	5.13E−5	0.0158
Euler, $M = 4$	–	0.0135	0.0127	0.0205
Euler, $M = 8$	–	6.63E−4	6.25E−4	4.81E−3
Euler, $M = 12$	–	3.53E−5	3.33E−5	7.32E−4

The Polynomial and Euler methods are the ones described by Boyd [13]. The Hermite FFT and the Hermite finite difference (Hermite FD) methods both use Hermite polynomials to interpolate the cosine function. For the Hermite FFT method (which is the one dimensional version of the method described in this paper), the exact spectral derivative is used for the Hermite interpolating polynomial. The Hermite FD method instead uses a three point finite difference estimation of the derivative. The Hermite FD method is a one dimensional scheme similar to the three dimensional scheme described by Lekien and Marsden [12].

The Hermite FFT method is about 10 times more spectrally accurate than the Hermite FD method. This is not surprising, because the Hermite FFT method uses the exact spectral derivative while the Hermite FD method uses an approximation.

The Hermite FFT method is comparable in accuracy to the Polynomial $M = 2$ and the Euler $M = 8$ methods. However, if the spectral derivatives are already known, the Hermite interpolation is much faster than either of these two methods, because it only involves the addition of four polynomials of fourth degree, with known weights. In addition, the Hermite method has C^1 continuity.

Thus the Hermite FFT interpolation scheme is useful when spectral accuracy is desired, and many interpolations will be done on a single grid, so that the added cost of calculating the spectral derivatives is acceptable.

3. Implementation in particle follower code

3.1. Description of problem

The Columbia Non-neutral Torus is a magnetic stellarator trap used to confine pure electron [6] and electron-ion [15] plasmas. The magnetic coils of CNT are a simple set of four circular, planar coils; but the magnetic geometry is very complicated [20]. For ion-electron plasmas, the ratio of ions to electrons is typically $N_i/N_e \leqslant 0.1$, so that there are always complicated electric fields present in the machine. The electron density and electrostatic potential of the plasma equilibrium are measured along a line using particle flux probes [7], and then the equilibrium and electrostatic potential is numerically reconstructed in the remaining volume of plasma [11], using a Poisson–Boltzmann solver (PBS).

The problem here is to integrate the motion of a test ion particle in the reconstructed equilibria of CNT. The ions are weakly magnetized, so that the drift approximations are inappropriate [4]. Instead we integrate the ion motion directly using the Lorentz force law [5]:

$$\frac{d\mathbf{p}}{dt} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}). \tag{16}$$

Because of the simple coil geometry in CNT, an analytical solution to the magnetic field exists [21]. On the other hand, the numerically reconstructed equilibria of CNT solves for the electrostatic potential on a Cartesian grid, ϕ_{ijk}. After this, an interpolation scheme is needed to calculate the electric field at an arbitrary point (x,y,z) that is not on the grid. The ion must be followed accurately for a very large number of steps, therefore it is necessary to develop an accurate interpolation scheme for the electric field.

In particular, it is necessary for the interpolated electric field $\vec{E}(x,y,z)$ to be a conservative force. This is best accomplished by introducing the electrostatic potential ϕ, with $\vec{E} = -\nabla\phi$. By interpolating ϕ rather than $\vec{E}$, then calculating $\vec{E}$ by analytically calculating $\vec{E} = -\nabla\phi$ the electric field will be continuous and conservative, as long as the interpolated potential ϕ (x,y,z) is smooth (C_1). The continuity of the electric field will increase the accuracy of numerical integration methods such as the adaptive Runge–Kutta or Adams–Moulton method.

The PBS code which calculates the potential uses a pseudo-spectral method [18,10], so that the solutions to the electrostatic potential can be interpolated using the algorithm described in this paper. The Poisson–Boltzmann solver solves the

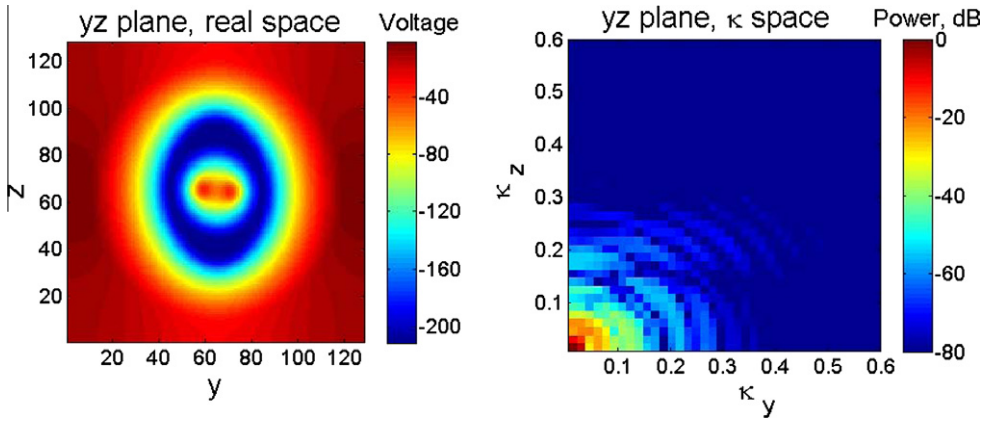

Fig. 3. Two dimensional plot of the electrostatic voltage of CNT, in midplane of the x direction. (Left): plot in real space (Right): plot in κ space.

electrostatic potential of CNT on a three dimensional grid that is $128 \times 128 \times 128$ in size. The grid points are evenly spaced, with each cube of grid points corresponding to a cube that is 1.28 cm on each side in the real CNT experiment. The values are given in double precision.

Fig. 3 shows a two dimensional plot of the electrostatic voltage used in CNT, in the midplane of the device. On the left this is shown in real space, with the y and z-axes labels in terms of grid points. On the right, this is shown in κ space, with $\kappa = k/k_{\text{limit}}$. Most of the spectral content of the three dimensional grid is contained in $|\kappa| < .1$, with the spectral content below -80 dB when $|\kappa| > 0.5$. Because most of the spectral content is contained in low κ values, adequate spectral accuracy is achieved without Fourier augmentation of the grid. The interpolation scheme described here is very accurate, with a relative error of $\approx 10^{-5}$ at producing results which are in agreement with the exact Fourier interpolation of the solutions of the PBS code.

3.2. Computational cost

For this application, a cubic domain is used, where there are $N = 128$ elements on each side. A single FFT requires $N \log N$ operations, and N^2 of these are needed to take a single derivative over the entire domain. Because there are seven combinations of first derivatives that need to be calculated, the total cost of calculating the derivatives is $7N^3 \log N$. These values are then stored in a $128 \times 128 \times 128 \times 2 \times 2 \times 2$ grid for use in the particle follower code.

For each individual interpolation, the computational cost is the cost of adding up 64 third or fourth degree polynomials.

3.3. Particle trajectories

The trajectory is calculated by integrating Eq. (16), using the analytical solution of the magnetic field, and the electric field interpreted using the methods described in this paper. Eq. (16) is integrated using the adaptive Adams–Moulton method. The code was benchmarked by solving the ion motion in an infinitely long Penning trap, where an analytical solution for the electric field [1] and the ion motion [14] is known. Excellent agreement was found with the analytical trajectories.

Fig. 4 shows the trajectory of an ion particle in CNT. A bouncing motion, similar to the motion of ions in a Penning trap, can be seen. In each bounce, the ion moves through the complicated electric and magnetic fields and picks up 20–80 eV of energy. The total energy of an ion as it goes through its trajectory can be calculated by summing the kinetic energy with the electrostatic potential, where the electrostatic potential is calculated using the interpolation scheme describe in this paper. It is found that even after 1099 bounces, which corresponds to 15 ms of ion motion in real time, the total ion energy is conserved to an accuracy 6×10^{-4} eV, or a relative accuracy of about 10^{-5}. There is no need to impose conservation of energy on the ion as its motion in integrated, it is valid because the interpolated electric field is calculated as the exact (to double precision accuracy) derivative of the potential, which itself has C^1 continuity.

The code was used to calculate ion trajectories, densities, and bounce times for a variety of experimental parameters in the Columbia Non-neutral Torus [16]. A typical ion trajectory, which lasts for ≈ 2 ms, or over 115 ion bounces, takes about 4 s to calculate on a standard PC.

4. Other applications of this algorithm

4.1. Converting to Boozer coordinates

For highly magnetized particles, such as electrons in CNT, magnetic coordinates are the more natural coordinate system for particle trajectory calculation in stellarators [8,22,2]. Converting to magnetic coordinates involves application of the FFT

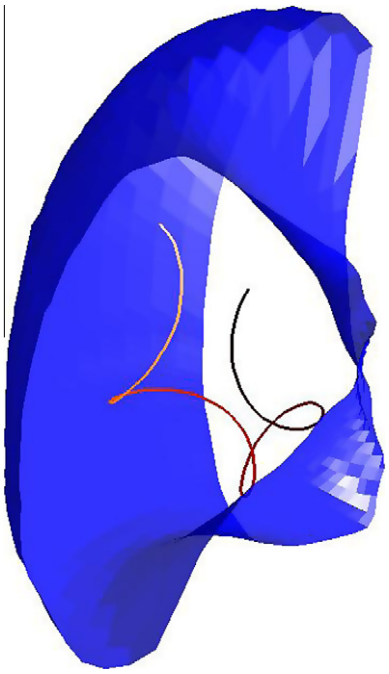

Fig. 4. Numerically integrated motion of N_2^+ in the reconstructed equilibria of a −200 V, 0.1 T plasma in the CNT geometry, along with a three dimensional rendition of the last closed flux surface of the magnetic field. The ion is moving from the yellow to the black section of the curve.

after integrating along a magnetic field line [8]. The interpolation technique described in this paper can be used to convert scalar parameters that are on a Cartesian grid, such as the equilibrium results of the PBS code in CNT [11] to magnetic coordinates, by providing an interpolation for values integrated along a magnetic field.

4.2. Self consistent beam trajectories

A class of codes exist which iterate between solving Poisson's equation and calculating the trajectories of a number of particles, in order to achieve self-consistency between the electrostatic boundaries and the electric field from the beam [17]. These codes often use a gridded mesh, with a finite difference method of calculating the electric field. In some situations the electric field can be calculated rapidly with a spectral method. Then, using the algorithm described in this paper, the particles can be followed rapidly and accurately.

5. Conclusion

A three dimensional interpolation scheme is proposed that is spectrally accurate, i.e., it accurately approximates the ideal Fourier interpolation, and has continuous first derivatives. The interpolation is also fast: a set of first derivates can be calculated rapidly using the FFT, and these can be stored in memory. Then, for each interpolation that is needed, the 64 Hermite basis functions are added together.

The interpolation scheme was tested by using it to calculate the electric field in a code that integrates the Lorentz force equation in the complicated electric and magnetic fields of the Columbia Non-neutral Torus. It was found that the Hermite interpolated voltages were almost identical to the Fourier interpolated voltages. It was also found that energy was very accurately conserved in the calculated particle trajectories.

Acknowledgments

The authors thank Paul Ennever and Dr. Thomas Sunn Pedersen. This work was supported by the US DOE Grant No. DE-FG02–02ER54690, the NSF-DOE Partnership in Basic Plasma Science, Grant No. NSF-PHY-03–17359, the NSF CAREER program, Grant Nos. NSF-PHY-04–49813 and NSF-PHY-06–13662.

References

[1] R.C. Davidson, Physics of Nonneutral Plasmas, second ed., Imperial College Press and World Scientific Publishing, London, UK, 2001.
[2] B. Durand de Gevigney, T.S. Pedersen, A.H. Boozer, Numerical studies of transport in the Columbia non-neutral torus, in: Non-neutral Plasma Physics VII, AIP Conference Proceedings, New York, AIP, New York, 2008.

[3] Takeshi Enomoto, Bicubic interpolation with spectral derivatives, Scientific Online Lett. Atmos. 4 (2008) 5.
[4] R.J. Goldston, P.H. Rutherford, Introduction to Plasma Physics, second ed., Institute of Physics Publishing, Philadelphia, USA, 1995.
[5] David J. Griffiths, Introduction to Electrodynamics, third ed., Prentice Hall, Upper Saddle River, NJ, 1999.
[6] J.P. Kremer, T. Sunn Pedersen, R.G. Lefrancois, Q.R. Marksteiner, Experimental confirmation of stable, small-debye-length, pure-electron-plasma equilibria in a stellarator, Phys. Rev. Lett. 97 (9) (2006) 095003.
[7] J.P. Kremer, T. Sunn Pedersen, Q. Marksteiner, R.G. Lefrancois, M. Hahn, Diagnosing pure-electron plasmas with internal particle flux probes, Rev. Sci. Instrum. 78 (2007) 013503.
[8] G. Kuo-Petravic, A.H. Boozer, J.A. Rome, R.H. Fowler, Numerical evaluation of magnetic coordinates for particle transport studies in asymmetric plasmas, J. Comput. Phys. 51 (1983) 261.
[9] P. Lancaster, K. Salkauskas, Curve and Surface Fitting: An Introduction, first ed., Academic Press, London, UK, 2001.
[10] R.G. Lefrancois, The equilibrium of electron plasmas confined on magnetic surfaces, Ph.D. Thesis, 2006.
[11] R.G. Lefrancois, T. Sunn Pedersen, A.H. Boozer, J.P. Kremer, Numerical investigation of three-dimensional single-species plasma equilibria on magnetic surfaces, Phys. Plasmas 12 (7) (2005) 072105.
[12] F. Lekien, J. Marsden, Tricubic interpolation in three dimensions, Int. J. Numer. Methods Eng. 63 (2005) 455–471.
[13] J.P. Boyd, A fast algorithm for Chebyshev and Fourier interpolation on to an irregular grid, J. Comput. Phys. 103 (1992) 243–257.
[14] R.H. Levy, J.D. Daugherty, O. Buneman, Ion resonance instability in grossly nonneutral plasmas, Phys. Fluids 12 (12) (1969) 2616.
[15] Q. Marksteiner, T. Sunn Pedersen, J.W. Berkery, M.S. Hahn, J.M. Mendez, B. Durand de Gevigney, H. Himura, Observations of an ion driven instability in non-neutral plasmas confined on magnetic surfaces, Phys. Rev. Lett. 100 (9) (2008) 065002.
[16] Q.R. Marksteiner, Studies of non-neutral ion electron plasmas confined on magnetic surfaces, Ph.D. Thesis, 2008.
[17] J.Gr. Pagonakis, J.L. Vomvoridis, The self-consistent 3d trajectory electrostatic code ariadne for gyrotron beam tunnel simulation, in: Gyro-Devices and other Vacuum Electronic Devices, IEEE Conference Proceedings, IEEE, 2004.
[18] T.S. Pedersen, Numerical investigation of two-dimensional pure electron plasma equilibria on magnetic surfaces, Phys. Plasmas 10 (02) (2003) 334.
[19] T.S. Pedersen, J.P. Kremer, R.G. Lefrancois, Q. Marksteiner, N. Pomphrey, W. Reiersen, F. Dahlgren, X. Sarasola, Construction and initial operation of the Columbia non-neutral torus, Fusion Sci. Technol. 50 (2006) 372–381.
[20] T.S. Pedersen, J.P. Kremer, R.G. Lefrancois, Q. Marksteiner, X. Sarasola, N. Ahmad, Experimental demonstration of a compact stellarator magnetic trap using four circular coils, Phys. Plasmas 13 (01) (2006) 012502.
[21] T. Sunn Pedersen, A.H. Boozer, J.P. Kremer, R. Lefrancois, F. Dahlgren, N. Pomphrey, W. Reiersen, W. Dorland, The Columbia nonneutral torus: a new experiment to confine nonneutral and positron–electron plasmas in a stellarator, Fusion Sci. Technol. 46 (2004) 200–208.
[22] J.F. Lyon, R.H. Fowler, J.A. Rome, Monte carlo studies of transport in stellarators, Phys. Fluids 28 (1) (1985) 338.