LA-7687-MS
Informal Report

UC-20g
Issued: March 1979

A Galerkin Algorithm for Multidimensional Plasma Simulation Codes

Brendan B. Godfrey

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

324

A GALERKIN ALGORITHM FOR MULTIDIMENSIONAL PLASMA SIMULATION CODES

by

Brendan B. Godfrey

ABSTRACT

A Galerkin finite element differencing scheme has been developed for a computer simulation of plasmas. The new difference equations identically satisfy an equation of continuity. Thus, the usual current correction procedure, involving inversion of Poisson's equation, is unnecessary. The algorithm is free of many numerical Cherenkov instabilities. This differencing scheme has been implemented in CCUBE, an already existing relativistic, electromagnetic, two-dimensional PIC code in arbitrary separable, orthogonal coordinates. The separability constraint is eliminated by the new algorithm. The new version of CCUBE exhibits good stability and accuracy with reduced computer memory and time requirements. Details of the algorithm and its implementation are presented.

I. INTRODUCTION

Realistic computer simulation of intense relativistic beam phenomena typically is very demanding. Boundary conditions are complex and often have a dominant effect on the physics. Significant space and time scales usually are of disparate magnitudes. There is occasional need for exotic coordinate systems. In general, relativistic plasma PIC simulations are more prone to numerical instability than nonrelativistic plasma simulations. To cope better with these considerations, we have implemented a Galerkin finite element difference algorithm into an already existing two-dimensional plasma simulation code, CCUBE.

The Galerkin algorithm represents the electric and magnetic fields as sums of finite elements, in this case splines. Equations interrelating the

coefficients of the finite elements, effectively the field values at mesh points, are derived by inserting the expansion for the fields into the usual Lagrange variational integral and minimizing the integral with respect to the coefficients. This procedure not only gives a discretized set of Maxwell's equations but also specifies a consistent interpolation procedure between the fields and the simulation particles. Galerkin's method is employed successfully in many branches of computational physics and engineering. References 1 and 2 provide good overviews. Lewis has discussed the application of Galerkin's method in the spatial domain to plasma simulation,^{3,4} while Godfrey has outlined the extension to space and time.^{5,6}

In plasma simulation Galerkin algorithms conserve charge on the mesh, conserve momentum along cyclic coordinates, minimize energy errors, and ameliorate certain numerical instabilities.⁷ It is the first of these features that we wish to emphasize. If the charge and current densities on the mesh satisfy a discretized equation of continuity, then Maxwell's equations may be advanced explicitly in time without inversion of Poisson's equation to obtain the electrostatic potential or to correct the longitudinal current. Dispensing with Poisson's equation greatly facilitates use of nonseparable coordinates and irregular boundaries.

Features of CCUBE, the two-dimensional plasma simulation code in which the algorithm has been implemented, can be summarized as follows. CCUBE was developed specifically for charged particle beams and, hence, is relativistic and fully electromagnetic. It runs in any orthogonal coordinate system, accommodates irregularly shaped boundaries and internal structures, and injects and absorbs particles at any of the boundaries. The code supports extensive diagnostics, generates movies, and is optimized for the CDC-7600 computer. More details are given in Ref. 8. CCUBE has been employed in a variety of applications, including electron beam heating of plasma, ion beam propagation, collective ion acceleration,^{9,10} space-charge limited flow,⁸ electron beam diodes, slow-wave structures,¹¹ and free electron lasers.

II. DESCRIPTION OF THE ALGORITHM

The heart of the Galerkin algorithm described here is the interpolation between particles and fields. The spatial interpolation procedure in two dimensions is:

$E^{(3)}, J^{(3)}, \rho$	Linear in X_1 and X_2
$E^{(1)}, J^{(1)}, B^{(2)}$	NGP in X_1 , linear in X_2
$E^{(2)}, J^{(2)}, B^{(1)}$	Linear in X_1 , NGP in X_2
$B^{(3)}$	NGP in X_1 and X_2

NGP designates nearest grid point interpolation. The fields and currents are staggered on the spatial mesh in the groups just listed. See Fig. 1.

Further setting apart the algorithm is its requirement for temporal interpolation. As in conventional approaches, electric fields are applied to particles at discrete times. However, the magnetic field is applied throughout each time step, NGP interpolation in time. Moreover, during each time step the average current during the time step is computed, again NGP interpolation. The particle charge density is not needed except at initialization and for diagnostics. It is evaluated at discrete times. Actually performing these NGP temporal interpolations is, unfortunately, impractical. Therefore, we settle for an N point quadrature as an approximation. The magnetic field is applied to the particles at N uniformly spaced points in time during each time step. The instantaneous current is evaluated at each time point and the N values averaged to provide input to the field solver. The particle equations are explicitly

$$dx/dt = \gamma^{-1} p$$

$$dP/dt = E^m \delta(t - m\Delta t) + N \sum_{i=1}^N \gamma^{-1} P \times B^{m+\frac{1}{2}} \delta(t - m\Delta t - \alpha_i \Delta t) \quad (1)$$

$$J^{m+\frac{1}{2}} = N \sum_{i=1}^N \gamma^{-1} P \delta(t - m\Delta t - \alpha_i \Delta t)$$

$$\alpha_i = (i - \frac{1}{2})/N \quad i = 1, N.$$

For simplicity the equations are given for rectangular coordinates only. Generalization to curvilinear systems is straightforward. Note that we have considered in some detail the locations of the temporal quadrature points. The uniform spacing indicated generally is optimal. Although the quadrature approaches true NGP interpolation only as the number of points becomes infinite, using just a few points typically gives good results. Figure 1 illustrates the temporal evolution of particles and fields for $N = 2$.

The field equations naturally consistent with the interpolation procedures described above involve nine-point differencing. However, we have found that five-point differencing gives nearly as good accuracy with reduced computation per time step and a slightly relaxed Courant condition. Only when greater accuracy for high-frequency light-wave phase velocities is required is the nine-point scheme noticeably superior. With this choice the electric fields are initialized from Gauss' law, Eq. (2).

$$\left[E_{n_1+\frac{1}{2}, n_2}^{(1),0} - E_{n_1-\frac{1}{2}, n_2}^{(1),0} \right] / \Delta x_1 + \left[E_{n_1, n_2+\frac{1}{2}}^{(2),0} - E_{n_1, n_2-\frac{1}{2}}^{(2),0} \right] / \Delta x_2 = \rho_{n_1, n_2}^0 \quad (2)$$

A similar expression exists for the magnetic fields,

$$\left[B_{n_1+1, n_2+\frac{1}{2}}^{(1),\frac{1}{2}} - B_{n_1, n_2+\frac{1}{2}}^{(1),\frac{1}{2}} \right] / \Delta x_1 + \left[B_{n_1+\frac{1}{2}, n_2+1}^{(2),\frac{1}{2}} - B_{n_1+\frac{1}{2}, n_2}^{(2),\frac{1}{2}} \right] / \Delta x_2 = 0 \quad (3)$$

How these equations are best solved depends on the details of the physical configuration being initialized. Sometimes it is necessary to solve Eqs. (2) and (3) simultaneously with force balance equations for the particle distribution. In any case it is critical to satisfy the field initialization equations very accurately, as errors introduced here persist throughout the simulation. The electric and magnetic fields, once initialized, are stepped forward in time according to Eqs. (4) and (5), using currents defined above.

$$\begin{aligned}
& \left[E_{n_1+\frac{1}{2}, n_2}^{(1), m+1} - E_{n_1+\frac{1}{2}, n_2}^{(1), m} \right] / \Delta t = - \left[B_{n_1+\frac{1}{2}, n_2+\frac{1}{2}}^{(3), m+\frac{1}{2}} - B_{n_1+\frac{1}{2}, n_2-\frac{1}{2}}^{(3), m+\frac{1}{2}} \right] / \Delta x_2 \\
& \quad - J_{n_1+\frac{1}{2}, n_2}^{(1), m+\frac{1}{2}} \\
& \left[E_{n_1, n_2+\frac{1}{2}}^{(2), m+1} - E_{n_1, n_2+\frac{1}{2}}^{(2), m} \right] / \Delta t = - \left[B_{n_1+\frac{1}{2}, n_2+\frac{1}{2}}^{(3), m+\frac{1}{2}} - B_{n_1-\frac{1}{2}, n_2+\frac{1}{2}}^{(3), m+\frac{1}{2}} \right] / \Delta x_1 \\
& \quad - J_{n_1, n_2+\frac{1}{2}}^{(2), m+\frac{1}{2}} \\
& \left[E_{n_1, n_2}^{(3), m+1} - E_{n_1, n_2}^{(3), m} \right] / \Delta t = \left[B_{n_1+\frac{1}{2}, n_2}^{(2), m+\frac{1}{2}} - B_{n_1-\frac{1}{2}, n_2}^{(2), m+\frac{1}{2}} \right] / \Delta x_1 \\
& \quad - \left[B_{n_1, n_2+\frac{1}{2}}^{(1), m+\frac{1}{2}} - B_{n_1, n_2-\frac{1}{2}}^{(1), m+\frac{1}{2}} \right] / \Delta x_2 - J_{n_1, n_2}^{(3), m+\frac{1}{2}} \\
& \left[B_{n_1, n_2+\frac{1}{2}}^{(1), m+\frac{1}{2}} - B_{n_1, n_2+\frac{1}{2}}^{(1), m-\frac{1}{2}} \right] / \Delta t = - \left[E_{n_1, n_2+1}^{(3), m} - E_{n_1, n_2}^{(3), m} \right] / \Delta x_2 \\
& \left[B_{n_1+\frac{1}{2}, n_2}^{(2), m+\frac{1}{2}} - B_{n_1+\frac{1}{2}, n_2}^{(2), m-\frac{1}{2}} \right] / \Delta t = \left[E_{n_1+1, n_2}^{(3), m} - E_{n_1, n_2}^{(3), m} \right] / \Delta x_1 \\
& \left[B_{n_1+\frac{1}{2}, n_2+\frac{1}{2}}^{(3), m+\frac{1}{2}} - B_{n_1+\frac{1}{2}, n_2+\frac{1}{2}}^{(3), m-\frac{1}{2}} \right] / \Delta t = - \left[E_{n_1+1, n_2+\frac{1}{2}}^{(2), m} - E_{n_1, n_2+\frac{1}{2}}^{(2), m} \right] / \Delta x_1 \\
& \quad + \left[E_{n+\frac{1}{2}, n+1}^{(1), m} - E_{n+\frac{1}{2}, n}^{(1), m} \right] / \Delta x_2 .
\end{aligned} \tag{4}$$

It is interesting that within the context of the Galerkin algorithm the magnetic field equations, Eqs. (3) and (5), are free of truncation error. Substitution of the finite element expansions of the electric and magnetic fields into Maxwell's equations give Eqs. (3) and (5) identically without approximation. That Eq. (5) is exact means that finite element differencing schemes

based on scalar and vector potentials can be constructed such that they yield numerical results identical to the present algorithm based directly on the fields. Moreover, those potential algorithms are related by a limited algebraic gauge invariance.⁵ The field equations given in this report are to be preferred only because they are more convenient to implement.

Although Eqs. (3) and (5) are truncation error free, Eqs. (2) and (4) are not. Nonetheless, an algebraic relation exists among them. Substituting Eq. (2) into Eq. (4) yields

$$\begin{aligned} & \left[J_{n_1+\frac{1}{2}, n_2}^{(1), m+\frac{1}{2}} - J_{n_1-\frac{1}{2}, n_2}^{(1), m+\frac{1}{2}} \right] / \Delta x_1 + \left[J_{n_1, n_2+\frac{1}{2}}^{(2), m+\frac{1}{2}} - J_{n_1, n_2-\frac{1}{2}}^{(2), m+\frac{1}{2}} \right] / \Delta x_2 \\ & + \left[\rho_{n_1, n_2}^{m+1} - \rho_{n_1, n_2}^m \right] / \Delta t = 0 \end{aligned} \quad (6)$$

This relation is recognizable as an equation of continuity for the interpolated charge and currents. Note, however, that it follows naturally from the discretized electric field equations and is not introduced in some ad hoc fashion. Alternatively, Eq. (6) can be viewed as a consistency relationship. If it is true, then Eq. (2) satisfied at one time step is necessarily satisfied at all time steps. Unfortunately, Eq. (6) is exact only for a true Galerkin algorithm. The quadrature approximation introduced into the particle equations, Eq. (1), gives rise to small errors in Eq. (6). Determining their magnitude is one goal of the simulations outlined in the following section.

III. PRELIMINARY TEST RESULTS

As a test of the algorithm, we have performed a series of simulations of a non-neutral relativistic electron beam propagating along a guide magnetic field in a metallic waveguide. Simulations are doubly aperiodic in cylindrical coordinates with $\Delta z = 0.391$, $\Delta r = 0.185$, and $\Delta t = 0.165$. (Units are chosen such that $c = \omega_p = 1$.) There are approximately four electrons per occupied cell. Standard smoothing is applied to the shortest wavelengths of the interpolated current. Beam parameters are $v = 1.75$, $\gamma = 7.0$, and $\omega_c = 2.0$. The beam enters through a ground plane and, therefore, bounces radially as it propagates. Such configurations are characteristic of some collective ion acceleration studies.⁹ Figure 2 is a movie frame taken from one simulation. The beam enters from the

left and exits at the right, although some electrons strike the waveguide at $r = 3.8$ and are absorbed.

These accuracy tests were intended to be as realistic as possible. Thus, the results are presented even though the simulations were subsequently found to be weakly numerically unstable. The instability, which typically is absolute, occurs at large perpendicular wave numbers and arises from the interaction of high-frequency light waves and aliases of the Doppler-shifted beam modes. In other tests in which this instability is suppressed, accuracy is much improved. The present results, shown in Figs. 3 and 4, are nonetheless satisfactory.

In each figure we compare results for $N = 1$ -, 2 -, and 3 -point quadratures. These results are contrasted with those of a "control" case, the canonical momentum algorithm discussed in Ref. 12 and implemented in an earlier version of CCUBE.

CCUBE is almost entirely vectorized, and for optimized running on the CDC-7600 employs the vector arithmetic package STACKLIB¹³ and a few specially written vector routines.¹⁴ Perhaps, a 25% increase in speed could be achieved by hand-coding key subroutines in assembly language but at the cost of considerable effort and a loss of flexibility. Figure 3a gives total central processor running time per particle per time step for our tests. Production runs for the same physical parameters would be nearly 10% faster due to a reduced need for diagnostics. The new algorithm actually is faster for one- and two-point quadratures and is equal in speed to the control algorithm for three points. The savings result principally from eliminating the Poisson equation solver, which is quite slow in curvilinear coordinates.

Continuity errors, a significant factor in evaluating the quadrature approximation to our Galerkin algorithm, are determined by evaluating the errors in Eq. (2) at late times. Figure 3b shows the relative RMS accumulated error at $t = 200$. For the control case, the usual current correction routine is bypassed. We see systematic improvement in the error as the number of quadrature points is increased. Recall that all these test simulations were mildly numerically unstable. Further simulations with that instability suppressed show a marked decrease in the continuity errors. We shall report on those results at a later date.

Both algorithms are of the "energy conserving" type^{3,4} and exhibit very good total energy conservation over long periods of time as illustrated in

Fig. 4a. The accuracy is particularly impressive in view of the large energy fluxes through the boundaries. The linear decrease in energy error with increasing number of quadrature points results principally from the improved integration of cyclotron orbits about the magnetic guide field.

Relative noise levels given in Fig. 4b were obtained by comparing the peak amplitudes of high frequency electric field fluctuations with the average amplitudes of the low frequency electric fields. Relative energy densities were, of course, much lower and rarely exceeded one percent. Although some of this noise is physical, most of it is caused by particle discreteness effects near the boundaries and by the weak numerical instability already mentioned. Increasing the number of quadrature points decreases the former effect. It is straightforward to reduce further the noise levels by judicious smoothing, if desired.

IV. CONCLUSIONS

A proper evaluation of the Galerkin space-time-symmetric algorithm clearly requires additional practical experience. We offer as a tentative judgment that the Galerkin algorithm in CCUBE generally is competitive with more conventional finite difference algorithms and yields significant advantages in cases when solution of Poisson's equation is burdensome. Clearly, it is at its best in multidimensional curvilinear systems. In closing, we make some more specific observations.

The Galerkin algorithm facilitates simulations in complex geometries. Particle motion in large magnetic fields is treated well. Energy is very well conserved. Noise levels are adequately low, although current smoothing is sometimes required. Residual continuity errors accumulate slowly and, if desired, can be eliminated by occasional reinitialization of the electrostatic fields. Numerical Cherenkov instabilities^{7,12} are to some extent suppressed. Computer requirements are not excessive.

ACKNOWLEDGMENTS

This project has profited from the contributions of far more people than can be acknowledged individually here. Their assistance is no less appreciated. This research was performed under the auspices of the U. S. Department of Energy, with additional support from the U. S. Army Ballistic Missile Defense Advanced Technology Center and the U. S. Air Force Weapons Laboratory.

REFERENCES

1. S. G. Mikhlin, The Numerical Performance of Variational Methods (Wolters-Noordhoff, Groningen, 1971).
2. B. A. Finlayson, The Method of Weighted Residuals and Variational Principles (Academic Press, New York, 1972).
3. H. R. Lewis, "Application of Hamilton's Principle to the Numerical Analysis of Vlasov Plasmas," Meth. Comp. Phys. 9, 307 (1970).
4. H. R. Lewis, "Variational Algorithms for Numerical Simulation of Collisionless Plasma with Point Particles Including Electromagnetic Interactions," J. Comp. Phys. 10, 400 (1972).
5. B. B. Godfrey and L. E. Thode, "Galerkin Difference Schemes for Plasma Simulation Codes," Proc. Seventh Conf. Num. Sim. Plas., New York, 1975, p. 87.
6. B. B. Godfrey, "Application of Galerkin's Method to Particle-in-Cell Plasma Simulation Codes," Proc. Eighth Conf. Num. Sim. Plas., Monterey, California, 1978, PE-3.
7. B. B. Godfrey, "Numerical Cherenkov Instabilities in Electromagnetic Particle Codes," J. Comp. Phys. 15, 504 (1974).
8. L. E. Thode, B. B. Godfrey, and W. R. Shanahan, "Vacuum Propagation of Solid Relativistic Electron Beams," Phys. Fluids, to be published.
9. B. B. Godfrey, "Numerical Simulation of Autoresonant Ion Acceleration," IEEE Plas. Sci. 5, 223 (1977).
10. B. B. Godfrey, "The Localized Pinch Model as a High Energy Ion Collective Acceleration Mechanism," IEEE Plas. Sci. 6, 256 (1978).
11. R. J. Faehl, B. S. Newberger, and B. B. Godfrey, "Simulation of Cyclotron Wave Growth in a Helical Slow Wave Structure," Phys. Fluids, to be published.
12. B. B. Godfrey, "Canonical Momenta and Numerical Instabilities in Particle Codes," J. Comp. Phys. 19, 58 (1975).
13. F. H. McMahon, L. J. Sloan, and G. A. Long, Stacklib (Lawrence Livermore Laboratory, 1977).
14. Vector scatter-gather routines were written by K. Fong; a vector reciprocal square root routine was written by L. Rudinski.

Fig. 1.

CCUBE employs a field mesh staggered in both space and time. Corresponding components of J and E are collocated in space. The temporal evolution illustrates an $N = 2$ quadrature.

Fig. 2.

Typical movie frame from test simulation of beam propagation in cylindrical drift tube, showing electron positions, contours of the axial electric field, and electron axial momenta.

Fig. 3.

Compared to conventional algorithms, the Galerkin method is reasonably fast and accumulates continuity errors slowly.

Fig. 4.

Energy conservation and noise suppression are good with the Galerkin algorithm.