JOURNAL OF COMPUTATIONAL PHYSICS 13, 114–129 (1973)

The Dipole Expansion Method for Plasma Simulation*

W. L. KRUER[†] AND J. M. DAWSON

Plasma Physics Laboratory, Princeton University, Princeton, New Jersey 08540

AND

B. ROSEN

Stevens Institute of Technology, Hoboken, New Jersey 07030

Received September 29, 1972

The dipole expansion technique for determining the grid charge density and the electrostatic forces due to a system of extended charges is presented. In this scheme a charge (monopole) and a dipole moment, both extended, are assigned to the nearest grid point. The corresponding calculations of the force require the determination of the electric field and its gradient. The approach aims to minimize the number of operations per particle at the expense of more operations per grid point. The technique is appealing on physical grounds, since it helps one to relate numerical approximations to physical concepts.

We also present a modified version, the subtracted dipole scheme (SUDS). In this version the field calculations do not take significantly longer than in nearest-grid-point calculations. This is particularly important in two- and three-dimensional simulations. We also discuss an optimization of the coding, particularly for the operations that must be done for each particle (computing its contribution to the charge density and updating its position and velocity). For example, this optimization has given a one-dimensional, nearest-grid-point code (lowest order in the multipole expansion scheme) that takes ~1.8 μsec/particle on the IBM 360/91. We assess the value of the dipole correction by comparing dipole and nearest-grid-point simulations. Finally the relation between the dipole expansion (or SUDS) and charge sharing is discussed.

* This work was supported jointly by U. S. Atomic Energy Commission Contract AT(11-1)-3073 and by the Office of Naval Research Laboratory Contract N00014-67-A-0151-0021. Use was made of computer facilities supported by National Science Foundation Grant NSF-GP579. One of the authors (B. R.) was partially sponsored by the U. S. Atomic Energy Commission Contract AT(30-1)-3785.

[†] Present address: University of California Lawrence Livermore Laboratory, Livermore, California 94550.

I. Introduction

Computer simulations of plasma using particles presently employ extended charges rather than sheets, lines, or points of charge. The advantages of extended particles [1, 2] incude the reduction of grid noise, smoothing of short wavelength interactions, and reduction of collisions. Current simulation methods in wide use are nearest-grid-point (NGP) [3], particle-in-cell (PIC) [4], and cloud-in-cell (CIC) [5]. Here we will present the dipole expansion technique [6, 7], a scheme which we have found very satisfactory in practice and which helps us to relate numerical approximations to physical concepts.

In computing the forces between extended particles it is convenient to use a grid and then to:

(1) assign the charge density of the particles to the grid points according to a rule such as:

(a) assign the entire charge to the nearest grid point (NGP),

(b) share the charge between two or more grid points according to an agreed algorithm;

(2) calculate the electric field by solving Poisson's equation;

(3) determine the force on a particle according to a rule consistent with that used in (1).

A particle of arbitrary shape should, to be exact, have its charge assigned to grid points over the entire support of the function describing the shape. This would lead to a rather inefficient algorithm for particle pushing. Instead of pursuing this path we recognize that the Fourier transform of the exact charge density possesses a systematic expansion in terms of transforms of multipole moments of the charge density. The moments for a given charge are calculated about its nearest grid point. In practice, as the title indicates, we have found the first two terms of this expansion sufficiently accurate.

The technique has proven very satisfactory and has given detailed agreement with a Vlasov code [8]. Computationally the scheme favors more operations on each grid point over fewer operations on each particle. Since there are many more particles than grid points, this can be a useful trade-off. The analytic details are presented in Section II.

A modified version of the dipole expansion technique, the subtracted dipole scheme or SUDS [9], has also been devised and is discussed in Section III. Here the field calculations do not take significantly longer than in nearest-grid-point calculations. We discuss in Section IV the implementation and optimization of the codes, in particular, the coding for those operations that must be done for each particle (computing its contribution to the charge density and updating its position and velocity). The value of the dipole correction is assessed by comparing dipole and nearest-grid-point simulations. Finally in Section V we comment on the relation of the dipole expansion technique (including SUDS) to charge-sharing schemes.

II. The Dipole Approximation

As described in the Introduction each of the simulation particles has a finite extension to its charge distribution. The actual force law between two such particles has a rather complicated spatial dependence but its calculation can be carried out in three straightforward steps. These are the following:

(1) Determination of the charge density;

(2) Fourier analysis of the charge density followed by algebraic operations that convert it to the Fourier transform of the force field;

(3) Fourier synthesis of the force field and determination of the force on any given particle.

The Fourier transform (FT) method is particularly useful here inasmuch as the charge density and the force are given by convolution integrals in ordinary space; in Fourier space the equivalent operations are merely multiplications. The existence of the Cooley–Tukey Fast Fourier Transform algorithm is a further reason for using transform methods.

Inasmuch as we can evaluate such quantities as charge density, force field, etc., only at a finite number of points such as the grid points, we must be able to properly assign the charge density of a particle to these points. We do this by expressing the charge density of each simulation particle as a multipole expansion about the grid point nearest to it. If we include only the lowest (i.e., monopole) terms we have a nearest grid point (NGP) estimate. Increased accuracy can be obtained by the inclusion of dipole terms which can be readily calculated. The force on each particle can be computed in a similar fashion with as many multipole terms kept as were required to calculate the charge density.

Let us present some of the analytic details of the dipole approximation. We will illustrate the procedure in one dimension, since its extension to more dimensions is obvious. The charge density ρ, is given exactly by

$$\rho(x) = \sum_i F(x - x_i) \tag{2.1}$$

where the summation is over all particles. The function F describes the charge shape (form factor) of the extended particles. If $\tilde{F}$ and $\tilde{\rho}$ are the FT's of F and ρ respectively, then we can write $\tilde{\rho}(k) = \tilde{F}(k)$ in the form

$$\tilde{\rho}(k) = F(k)\left\{\sum_{n=0}^{\infty}\sum_{g}\left[\sum_{i\in g}\frac{(-ik\,\Delta x_i)^n}{n!}\exp(-ikX_g)\right]\right\}. \tag{2.2}$$

In this last form X_g represents the position of the g-th grid point and $\Delta x_i = x_i - X_g$ is the distance between the i-th particle and the grid point nearest to it. The innermost sum is over all particles in the g-th cell and is proportional to the n-th multipole moment of those charges with respect to the center of the cell. The intermediate sum is over the N cells and is the finite Fourier transform of the corresponding multipole moments. Finally the outermost sum is over the multipole moments.

In the *dipole* approximation then we rewrite (2.1) as:

$$\rho(x) = \sum_{g}[F(X_g - x)\,\rho_{\mathrm{NGP}}(g) + F'(X_g - x)\,\rho_D(g)]; \tag{2.3a}$$

$$\rho_{\mathrm{NGP}}(g) = \sum_{i\in g} 1; \tag{2.3b}$$

$$\rho_D(g) = \sum_{i\in g}(x_i - X_g). \tag{2.3c}$$

We have assumed that F is an even function.

The force on an extended particle is determined in the same manner. Starting with

$$\begin{aligned} F(x_i) &= e\int E(x)\,F(x_i - x)\,dx,\\ &= e\int E(x_i + x)\,F(x)\,dx \end{aligned}$$

as the force on the i-th particle we expand the electric field about the grid point nearest the i-th particle to obtain the dipole approximation to the force:

$$F_D(x_i) = \int F(x)[E(x + X_g) + (x_i - X_g)(\partial E(x + X_g)/\partial X_g)]\,dx. \tag{2.4}$$

The force calculation in the dipole scheme is based directly on Eqs. (2.3) and (2.4) and on Poisson's equation

$$\nabla\cdot E = -4\pi\rho \tag{2.5}$$

and is performed in the following steps:

(1) The monopole (ρ_{NGP}) and dipole (ρ_D) densities are calculated at the grid points using (2.3).

(2) The Fourier components of these densities are calculated using the discrete Fourier transform.

(3) The Fourier components of the monopole ($F1$) and dipole ($F2$) contributions to the electrostatic force field are computed:

$$\begin{aligned} F1(k) &= (ik/k^2)[\rho_{\mathrm{NGP}}(k) - ik\rho_D(k)] \cdot \tilde{F}^2, \\ F2(k) &= ikF1(k). \end{aligned}$$

Note that $\tilde{F}^2$ occurs in the force whereas only one factor of $\tilde{F}$ is needed to compute the Fourier component of the electric field.

(4) After inversion of the transforms the force on a given particle is proportional to

$$F1(X_g) + (x_i - X_g)\, F2(X_g),$$

where once more X_g is the grid point nearest the particle.

In a two-dimensional simulation one would need to compute at least eight FT's. These include the Fourier transforms of ρ_{NGP} and ρ_D and then the inverse transform of the two components of the monopole forces and finally at least three of the four components of the dipole terms in the force. One might think that two rather than one contribution to the dipole terms in the force could be eliminated by use of the Maxwell equations:

$$\begin{aligned} (\partial E_x/\partial x) + (\partial E_Y/\partial Y) &= -4\pi\rho, \\ \partial E_x/\partial Y &= \partial E_Y/\partial x. \end{aligned}$$

However, the calculation of $\rho(x)$, including both dipole and monopole terms, requires a separate FT. The calculation of these gradients in two or three dimensional simulations is time consuming and we present, in the next section, an alternative algorithm for use with the dipole approximation.

III. Substracted Dipole Scheme (SUDS)

The modification we present relates to the grid calculation and reduces the number of Fourier transforms to a minimum [9]. The basic new feature is that we approximate derivatives appearing in higher-order expansion terms by differences over the cells. The error incurred represents a correction to a correction, and hence is higher order in the expansion parameter. We again illustrate the procedure in one dimension. As in Section II, we consider a collection of finite-size particles.

Recall that the charge density in the dipole approximation was given by

$$\rho(x) = \sum_{g} \sum_{i \in g} \left[F(x - X_g) - \Delta x_i F'(x - X_g) + O\left(\frac{1}{10} \frac{\delta^2}{a^2}\right)\right]$$

Truncating the expansion at the dipole term incurs an error or order (1/10) δ^2/a^2, where δ is the cell size and a the effective size of the particle.

The charge density can now be Fourier-transformed, but this means that the correction (ie., dipole) term involves computation of additional transforms (N more in an N-dimensional simulation). Let us instead approximate the derivative of the form factor by using a centered difference over the adjacent cells:

$$F'(x - X_g) = ((F(x - X_{g+1}) - F(x - X_{g-1}))/2\delta) + O(\delta^2/a^2).$$

then,

$$\rho(x) = \Sigma F(x - X_g)\{\rho_{\text{NGP}} + (1/2\delta)[\rho_D(g + 1) - \rho_D(g - 1)]\}.$$

The charge density can now be Fourier-transformed simply by making one transform of the modified array shown in braces. In this way we save one, two, or three transforms, depending on the dimensionality of the problem.

Figure 1 shows the two functions

$$f'(x) \qquad \text{and} \qquad (f(x + 1) - f(x - 1))/2,$$

where $f(x) = \exp - (x^2/2)$ and indicates the nature of the approximation for what would be the *worst* case in practice; namely, the particle width equal to a cell size. In physical terms we are replacing the dipole $d \cdot f'(x)$ by a dipole which consists of two extended charges $f(x)$, of equal but opposite charge $(d/2\delta)$, a distance of 2δ apart. The dipole moments of these two charge distributions agree, but the higher moments differ.

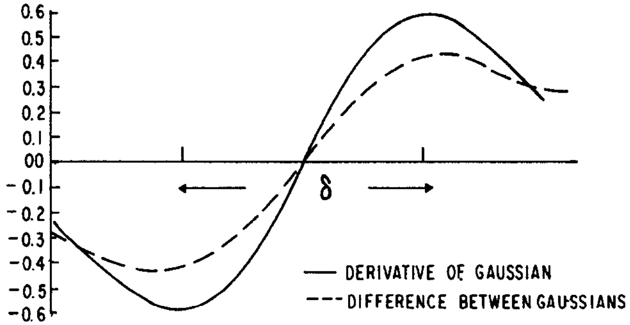

FIG. 1. Comparison of the derivative of a Gaussian curve with the difference of two Gaussian curves whose centers are two cells apart. The Gaussian half-width is equal to a cell size.

The same technique is used when we expand the electric field to determine the force on the particle. The net force on the particle is given by the monopole moment times the electric field plus the dot product of the dipole moment and the gradient of the electric field. By approximating the derivatives of the field using central differences, we again avoid performing Fourier transforms to compute those derivatives. This saves two additional transforms in two-dimensional, and five in three-dimensional simulations. Of course, we must perform some additional operations to difference the various quantities, but the number of Fourier transforms now required is the same as in nearest-grid-point calculations.

IV. Implementation and Optimization

A. *The Basic Dipole Approximation*

The basic dipole scheme has been implemented and has proven very satisfactory. For example, we have tested the code by running out and reversing a violent two-stream instability. We use 10^4 particles in a one-dimensional system 256 Debye lengths long. The result is shown in Fig. 2. The top of the figure shows phase space at $\omega_{pe}t = 5$. We see the two Maxwellian beams, separated in velocity by 6 thermal velocities. The instability exponentiates many times, making the field energy $\sim 5\,\%$ of the total energy. Even so, the code conserved energy to within $\sim 0.03\,\%$.The center of the figure shows phase space at $\omega_{pe}t = 35$. The instability has saturated; vortices have formed and are beginning to coalesce.

At this time we reverse the velocities of the particles. Finally, the bottom of the figure shows phase space after the system has traced itself back in time to $\omega_{pe}t = 5$. The top and bottom pictures are identical in detail (reversing the velocities interchanges the top and bottom beams). This test gives a good check on the numerical stability of the code as well as checking the accuracy of the programs and errors due to round off.

For a more interesting application of the code we have investigated the effective collision frequency in a plasma of finite-size charged rods (two-dimensional charges). To do this we have measured the collisional damping of plama oscillations. Working with a thermal equilibrium plasma we compute the time correlations of the E field for the Fourier modes; that is, we average $E_k(t)\, E_k(t + \tau)$ over t. The correlation function is a slowly decaying oscillation: the frequency of the oscillation yields the frequency of the plasma mode and its decay gives the damping (for long wavelength modes this is collisional damping).

For these runs we used a two-species plasma in a system $32\lambda_D$ by $32\lambda_D$ (where λ_D is the Debye length). We varied both the particle size and the total number of particles used. The results for a number of runs are shown in Fig. 3, where we have plotted collision frequency versus the ratio of the particle size to the Debye length.

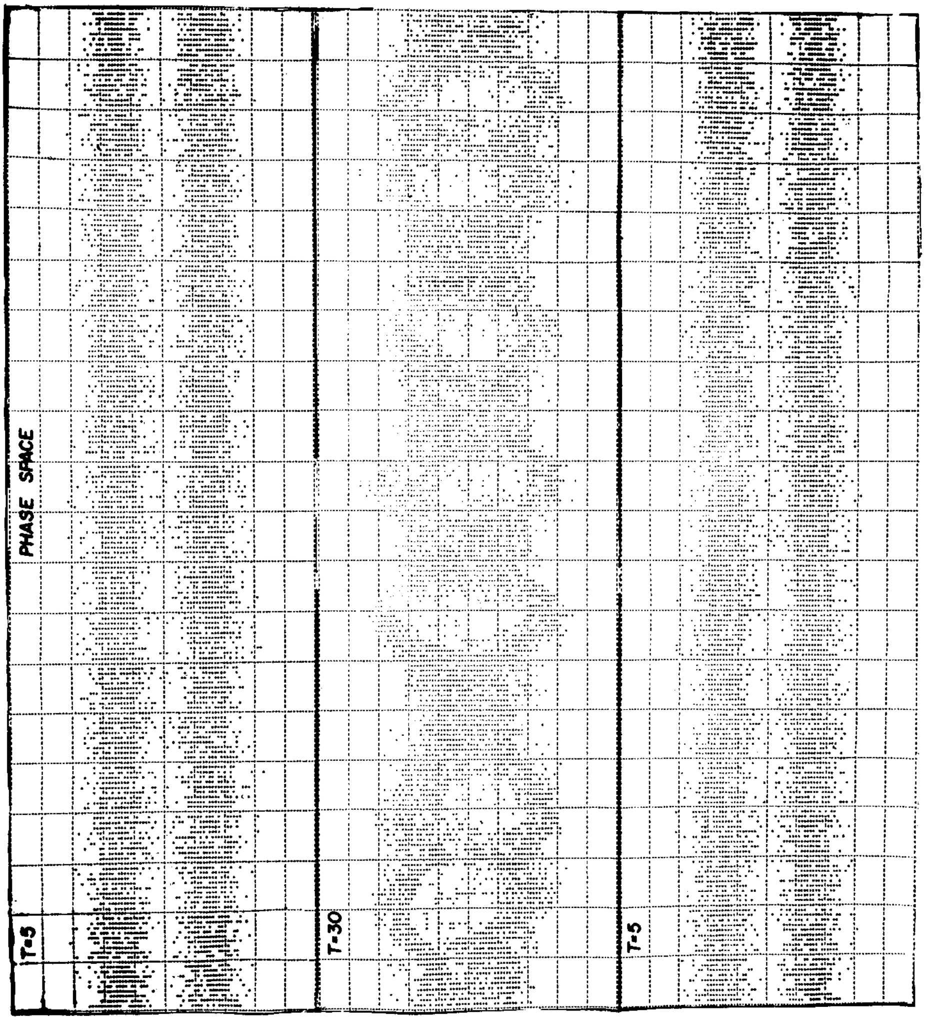

FIG. 2. Phase space plot of a strong two-stream instability calculated by the dipole method. The lowest portion arises from the middle part by time reversal.

The upper curve is for the number of particles per Debye circle equal to five, and the lower curve for that number equal to 15. The $R = 0$ points are taken from an empirical formula deduced by Hockney [10] from a large number of runs with essentially zero-size particles. The reduction of collision frequency with increasing particle size is evident. A particle size of λ_D leads to a reduction of about a factor of 10. The magnitude of this reduction agrees with the calculations of Okuda [2].

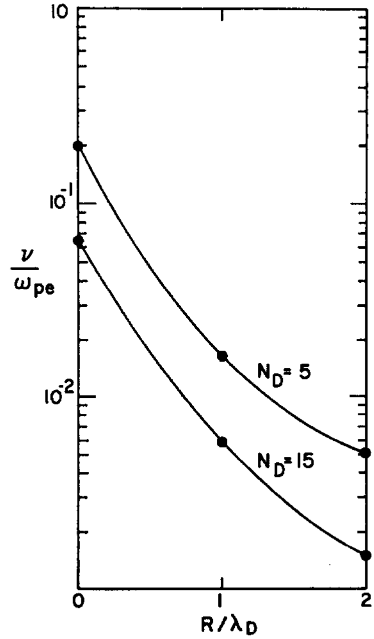

FIG. 3. Collision frequency versus particle size.

B. *SUDS*

The modified version of the dipole scheme has also been implemented and has given close agreement with the usual dipole technique. As a simple example, we consider a two-dimensional simulation of a thermal equilibrium plasma. The system consists of 5000 particles in a doubly periodic system ($8\lambda_D$ by $8\lambda_D$). Figure 4 shows a comparison of energy conservation for all three techniques: NGP, dipole, and modified dipole. Energy conservation is quite good for both dipole techniques, and is an order of magnitude better than that for NGP.

The grid calculation is observed to be nearly twice as fast in the modified dipole version for this two-dimensional simulation. Furthermore, the grid calculation has been speeded up by an additional factor of ~4 through the use of an optimized Fourier transform package. For example, in ~0.15 sec on the IBM 360/91 we can now Fourier-transform the charge density on a 64 by 64 grid, convert to the Fourier components of E_x and E_Y, and invert the transform to find the fields.

We have concentrated on discussing the field calculations, since the technique for doing these has been modified. For completeness, let us now briefly discuss the time required for the operations that must be performed for each particle (computing its contribution to the charge density and updating its position and

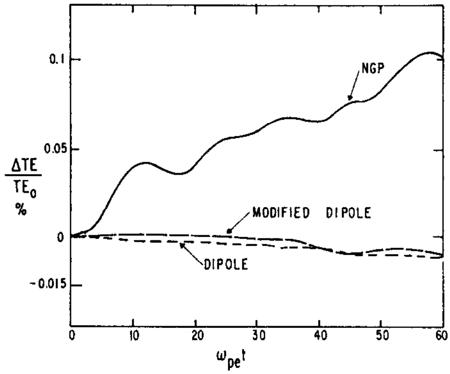

FIG. 4. A comparison of energy conservation in NGP, dipole, and SUDS simulations of a plasma in thermal equilibrium.

velocity). At Princeton a two-dimensional NGP code (lowest order in the multipole expansion) has been developed which requires for one time step ~4 μsec/particle on the IBM 360/91 [11]. In this code the two particle coordinates are packed into one double word, and similarly for the two components of the velocity [12]. In a similar code, one of the authors (B. R.) has developed a one-dimensional NGP code that takes ~1.8 μsec/particle. In this latter code each particles's velocity and position are packed into one double word. Both of these codes take advantage of the architecture of the IBM 360/91. In particular, such features as the double-word fetch and the parallel execution of fixed and floating point instructions are exploited. A dipole version of this optimized one-dimensional code requires ~4 μsec/particle.

C. *Comparison of Dipole and NGP Simulations*

Finally, let us indicate the value of the dipole correction (or any charge-sharing type of correction) by comparison with simulations using the nearest-grid-point approximation. It is particularly convenient to do this in the dipole expansion scheme, since NGP is simply zero order in this expansion. In the following examples we use the modified dipole technique.

First we consider a strong two-beam instability—two Maxwellian beams separated by $4(2)^{1/2}$ thermal velocities [13]. We use 5000 particles in a doubly periodic system ($32\lambda_D$ by $32\lambda_D$). This is a convenient problem for numerical tests, but of course we would use both a larger grid and more particles to investigate the physics with confidence. Figure 5 shows the total electrostatic wave energy for both the dipole and NGP simulations. This energy exponentiates to ~2.6 % of the total energy. The NGP results agree reasonably well with the dipole results. In the course of the simulation, the energy for the NGP solution varies from the initial

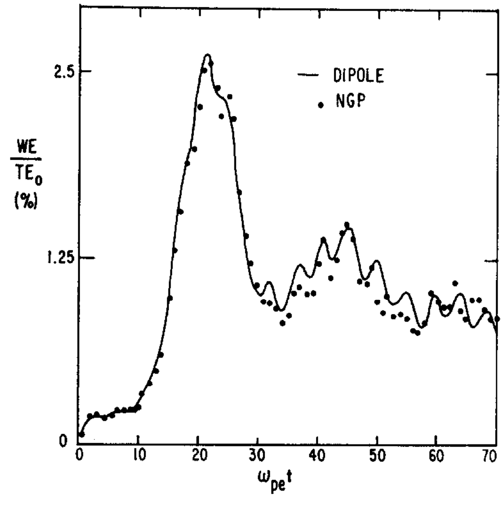

FIG. 5. Wave energy versus time compared for dipole and NGP simulations of a strong two-beam instability.

total energy of the system by ~0.2 %, but this is adequate to quantitatively follow the gross behavior of this rapidly growing instability.

However, let us now investigate a weaker instability. We consider a small beam moving at 6.4 thermal velocities in a Maxwellian plasma. The ratio of the beam density to the background plasma is ~0.25 %. The main plasma contains 19,000 simulation particles and the beam contains 1000 particles, each with 0.05 of the charge and mass of a plasma particles. The system is one-dimensional with a periodicity length of $256\lambda_D$.

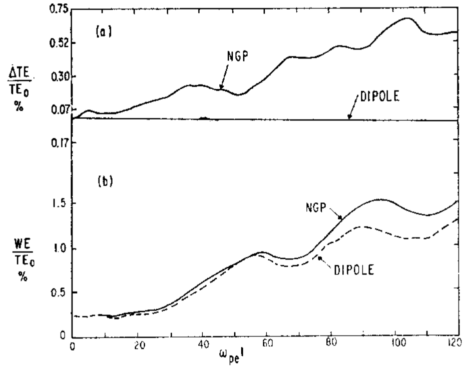

FIG. 6. Dipole and NGP results for (a) the total energy versus time and (b) the electrostatic energy versus time for a weak two-beam instability.

Figure 6 shows both the electrostatic wave energy and the change in the total energy versus time for both of the techniques. The wave energy at $\omega_{pe}t = 120$ has grown to ~1.2 % of the initial total energy of the system. The NGP result agrees qualitatively with the dipole result, but it does not agree quantitatively. Indeed, by $\omega_{pe}t$ ~100 the total *wave* energies in the two solutions differ by ~25 %. This quantitative disagreement is not surprising when we note that the energy conservation for NGP is rather poor in this case. The total energy deviates by ~0.8 % from the initial total energy, while the wave energy becomes ~1.2 %.

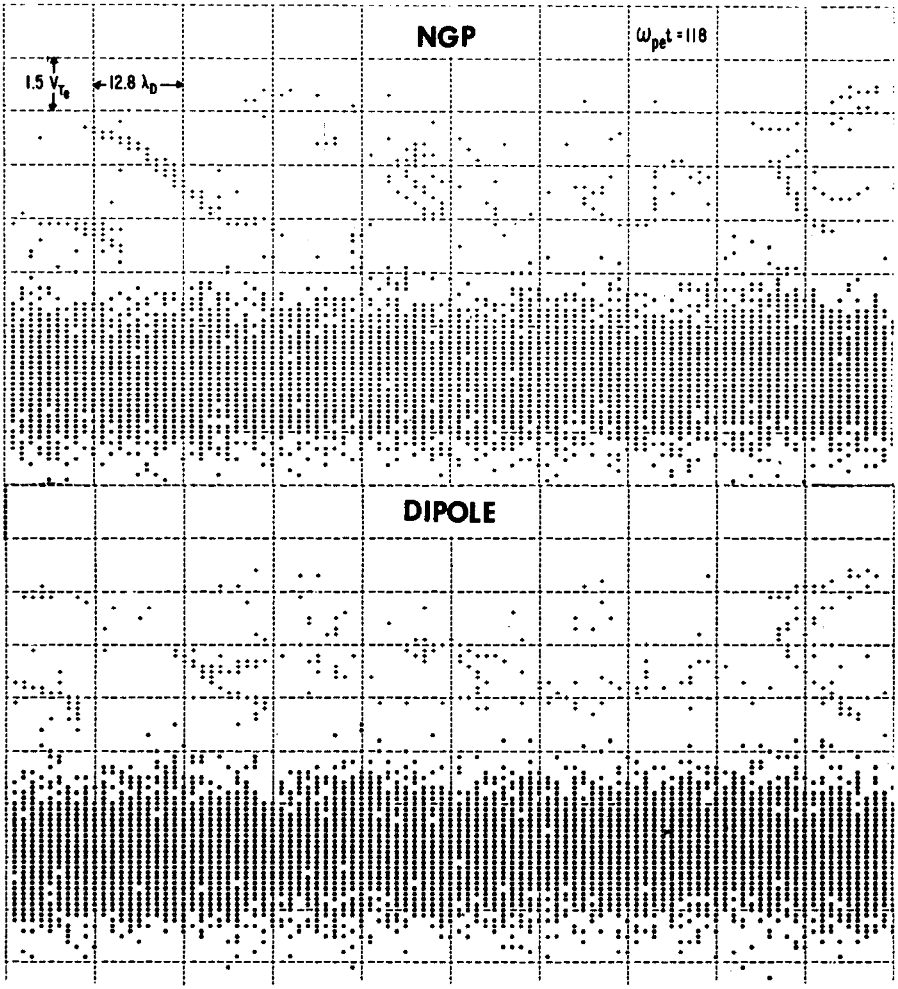

FIG. 7. A comparison of phase space $\omega_{pe}t = 118$ for NGP and dipole simulations of a weak two-beam instability.

Figure 7 shows a comparison of phase space at $\omega_{pe}t = 118$ for both of the techniques. We see quite noticeable differences, showing that significant errors have accumulated in the particle orbits in the NGP solution. (The phase space plots in the dipole and the modified dipole solutions agree closely.)

V. The Dipole Approximation and Charge-Sharing

In this last section let us briefly comment on the relation between the dipole expansion technique (or SUDS) and charge sharing. The multipole expansion scheme, when truncated at the dipole order, is of course similar to the usual charge-sharing techniques. Indeed these latter schems, in effect, apportion the charge on the grid in such a way as to conserve both the monopole and dipole moments of the original charge (plus in general parts of higher order moments). The way to do this is not uniquely defined and corresponds to our freedom to approximate the derivative terms in various ways.

From a computational viewpoint our technique has both advantages and disadvantages. Basically we emphasize a minimization of the number of operations per particle. This can be quite important for simulations in more than one dimension. In two (three) dimensions the specification of the charge and dipole moment of a particle requires the assignment of a monopole moment and two (three) dipole moment components to the nearest grid point. In contrast, the usual charge sharing methods assign a portion of the charge to four (eight) different grid points. There are similar economies in the force calculations.

On the other hand, the dipole scheme yields a less smooth interpolation from the grid to the particles (and vice versa) than does conventional charge sharing. As Langdon has pointed out, in the dipole scheme the force on a particle has a small discontinuity at the cell boundaries. The magnitude of this discontinuity across the right-hand boundary of the I-th cell, for example, is

$$[F(I) + (\delta/2)\,F'(I)] - [F(I+1) - (\delta/2)\,F'(I+1)]$$

where F and F' are the values of the force and its *slope* at the indicated grid points. Note that in using the dipole approximation we have used two data defined at one grid point, viz., $F(x)$ and $F'(x)$, to calculate the force whereas charge sharing used two data, $F(x)$ and $F(x + \Delta x)$ say, each from a different grid point. Figure 8 shows the difference in the two methods graphically. Alternatively one might say that in effect charge sharing uses functional values and two-sided slopes at each

[1] We use the work slope here to mean the actual derivate in the dipole scheme proper or the difference formed in SUDS.

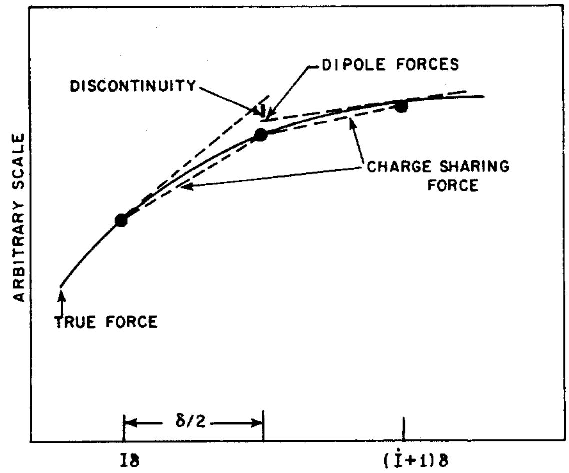

Fig. 8. Comparison of forces given by dipole method and charge sharing.

grid point to represent a function polygonally whereas the dipole approximation uses functional values and a symmetric slope to represent a function piecewise.

In the SUDS form, which is most akin to charge sharing, the weight function $w_s(x)$ which interpolates the particle density onto the grid is shown in Fig. 9. Its Fourier transform is given by

$$\tilde{\omega}_s(k) = -(1/\beta) + (1/\beta^2) \sin \beta \cos 2\beta; \qquad \beta = k\delta/2.$$

Because $w_s(x)$ has a discontinuity in its value, $\tilde{\omega}_s(k)$ falls off asymptotically as $1/k$, i.e. in a manner similar to that of NGP weighting function.

Slight modifications to SUDS can improve the interpolation of quantities on the grid, while retaining most of the economies. For example, one can adopt the

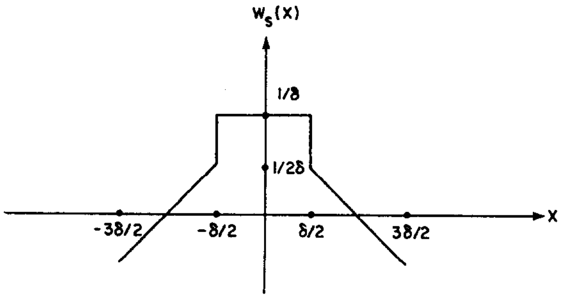

Fig. 9. The weight function used in the SUDS algorithm.

weighting function $w_s(x)$ shown in Fig. 10. In one dimension, of course, this weight function is the average of two PIC weight functions shifted by the grid spacing. Once again the net moments rather than the individual particle moments are shared but in this case both the monopole and dipole moment are distributed.

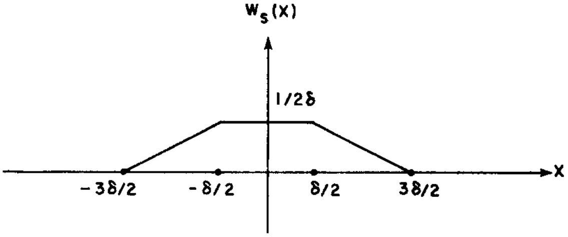

FIG. 10. Proposed modified weight function to be used in SUDS.

Finally we would like to mention a hybrid version of SUDS in which the force calculation is very nearly NGP while the charge calculation is as before. Usually such schemes exhibit a lack of momentum conservation because of the inequality between action and reaction. If, however, we consider the total force on the N_g particles in the g-th cell then SUDS would predict this to be

$$\sum_{i \in g} [F(X_g) + \Delta x_i \cdot F'(X_g)].$$

However, we can write this as

$$N_g[F(X_g) + (\Sigma\, \Delta x_i/N_g) \cdot F'(X_g)].$$

Since the value of the slope is derived in SUDS by a differencing procedure we can interpret this formula as "field-sharing" or applying an average dipole correction to each particle in a cell. The field quantity in brackets is used to accelerate every particle in the cell. We can look upon it also as the field at the center of charge in the cell since the bracketed term can be regarded as approximately

$$F[X_g + (\Sigma\, \Delta x_i/N_g)].$$

SUMMARY

We have presented the dipole expansion technique for particle simulation of plasmas. This technique is a physical way to introduce the spatial grid and helps one to relate numerical approximations to physical concepts. The approach aims to minimize the number of operations per particle at the expense of more operations per grid point. A modified version of the dipole expansion technique is also presented. Here the number of Fourier transforms is minimized, so that the field calculations do not take significantly longer than in NGP calculations.

The implementation and optimization of these codes have been discussed, as well as their relationship to other techniques. NGP and dipole results have been compared for several simple problems. These comparisons show that NGP gives adequate results for strong instabilities and rapid evolution but is poor for treating weak instabilities or more subtle aspects of plasma dynamics. The dipole and SUDS methods are capable of handing fairly subtle behavior.

Acknowledgments

We would like to thank Dr. A. Bruce Langdon for penetrating remarks on the dipole approximation. We are grateful to Mr. H. Fallon for his able assistance and to Drs. J. P. Boris and N. Winsor for useful discussions.

References

1. A. B. Langdon and C. K. Birdsall, *Phys. Fluids* **13** (1970), 2115.
2. H. Okuda and C. K. Birdsall, *Phys. Fluids* **13** (1970), 2123.
3. S. P. Yu, G. P. Kooyers and O. Buneman, *J. Appl. Phys.* **36** (1965), 2550.
4. R. L. Morse and C. W. Nielsen, *Phys. Fluids* **12** (1969), 2418.
5. C. K. Birdsall and Dieter Fuss, *J. Comp. Phys.* **3** (1969), 494.
6. J. M. Dawson, C. Hsi, and R. Shanny, Princeton University Plasma Physics Laboratory Matt-719, 1969.
7. W. L. Kruer and J. M. Dawson, *Bull. Amer. Phys. Soc.* **14** (1969), 1025.
8. J. Denavit and W. L. Kruer, *Phys. Fluids* **14** (1971), 1782.
9. B. Rosen, W. L. Kruer and J. M. Dawson, Proceedings of the Fourth Conference on Numerical Simulation of Plasmas, Office of Naval Research, Va., 1970, p. 561.
10. R. W. Hockney, *Phys. Fluids* **9** (1966), 1826.
11. H. Fallon, N. Winsor, and W. L. Kruer, *Bull. Amer. Phys. Soc.* **14** (1969), 1025.
12. J. P. Boris and K. V. Roberts, *J. Comp. Phys.* **4** (1969), 4.
13. R. L. Morse and C. W. Nielsen, *Phys. Rev. Lett.* **23** (1969), 1087.