JOURNAL OF COMPUTATIONAL PHYSICS 115, 352–365 (1994)

Numerical Instability in a 2D Gyrokinetic Code Caused by Divergent $E \times B$ Flow

J. A. BYERS, A. M. DIMITS, Y. MATSUDA,* AND A. B. LANGDON

Lawrence Livermore National Laboratory, Livermore, California 94550

Received April 9, 1992

In this paper, a numerical instability first observed in a 2D electrostatic gyrokinetic code is described. The instability should also be present in some form in many versons of particle-in-cell simulation codes that employ guiding center drifts. A perturbation analysis of the instability is given and its results agree quantitatively with the observations from the gyrokinetic code in all respects. The basic mechanism is a false divergence of the $\mathbf{E} \times \mathbf{B}$ flow caused by the interpolation between the grid and the particles as coupled with the specific numerical method for calculating $\mathbf{E} = -\nabla\phi$. Stability or instability depends in detail on the specific choice of particle interpolation method and field method. One common interpolation method, subtracted dipole, is stable. Other commonly used interpolation methods, linear and quadratic, are unstable when combined with a finite difference for the electric field. Linear and quadratic interpolation can be rendered stable if combined with another method for the electric field, the analytic differential of the interpolated potential. © 1994 Academic Press, Inc.

1. INTRODUCTION; RELEVANCE TO OTHER WORK

For many years particle simulation has been an essential tool for the investigation of nonlinear kinetic phenomena throughout many areas of plasma physics [1–3]. Standard particle simulation codes generally use some form of a particle-in-cell [2, 3] method, whereby the particle source terms are accumulated on a discrete grid through some specific method of choice for interpolation from the particles to the grid and where the self-consistent fields are then numerically solved on the grid and then interpolated back to the particles to solve for the particles trajectories by a finite difference in time. The specific choices of the interpolation methods and the field algorithm are the critical points for the issue dealt with in this paper.

The notation used in this paper is standard, employing SI units where dimensional quantities are distinguished. When discussing linear waves, Fourier analysis in space and time is assumed, $\exp(i\mathbf{k}_\perp \cdot \mathbf{x} - i\omega t)$, with frequency ω, and perpendicular wavenumber $\mathbf{k}_\perp$. The ion gyro frequency is Ω_{ci} and the ion gyro radius is ρ_i. The electric field is given by $\mathbf{E}$ and the magnetic field by $\mathbf{B}$.

Here we report the discovery and analysis of, and the remedy to, a numerical instability in a 2D electrostatic gyrokinetic code. The gyrokinetic formalism for low frequency $\omega < \Omega_{ci}$ particle simulation, allowing short perpendicular wavelengths, comparable to or smaller than the ion gyro radius, $k_\perp \rho \gtrsim 1$, was developed in a series of papers by Lee and co-workers [4–8]. Subsequently, others have extended the models and reported results using gyrokinetic particle codes applied to the study of many aspects of nonlinear drift wave phenomena [9–13]. We believe that all of this cited work employed codes using the subtracted dipole [2] interpolation method which is *not* subject to the numerical instability discussed in this paper. We have discovered one published gyrokinetic paper, discussed below, that used an algorithm that we believe to be unstable. The instability may well have been present in some form in other gyrokinetic codes used by others for related work not cited here. The references cited here are not exhaustive; the intention is to indicate related work that *may* be affected by the considerations discussed in this paper. The numerical instability dealt with in this paper depends only on the most basic feature of the gyrokinetic formalism, the $\mathbf{E} \times \mathbf{B}$ drift motion. Numerical instability or stability depends in a detailed way on the specific choice of the methods used to connect the particles and the grid.

The gyrokinetic formalism has recently taken on new importance as it has become recognized that it offers the possibility for successful modeling of instabilities and their effect on transport in tokamaks. It clearly is important to reveal any possibility of numerical effects impacting results purported to be primarily representative of physics.

The principal attribute of the gyrokinetic formalism that concerns us here is the guiding center $\mathbf{E} \times \mathbf{B}$ motion of the particles. Our discussion will be limited to the zero gyro radius limit. The numerical instability results from a false divergence of this motion. For an electrostatic $\mathbf{E}$ and a uniform $\mathbf{B}$, analytically $\nabla \cdot (\mathbf{E} \times \hat{\mathbf{b}}/B) \equiv 0$, but the numerical algorithms in the gyrokinetic codes often produce a nonzero unstable response instead. This instability may be present in some form in some of the other codes in the gyrokinetic community. The only ones we are aware of is Sydora's 2D code [14] which was the origin

* Dr. Yoshi Matsuda died on October 6, 1991 as a result of a fall. He was much loved by his friends and colleagues and will be missed greatly. He is survived by his wife and two young children.

0021-9991/94 $6.00

of the LLNL version, and one verison of Sydora's 3D code used for simulation of trapped electron instability [15]. This latter code used an algorithm that our analysis shows to be unstable (for our cold ion, adiabatic electron model), albeit with a lower growth rate than the method in our unstable 2D code. In addition, Sydora's drift kinetic treatment of the electrons, as opposed to our adiabatic electron model, would tend to further reduce the growth rate. Our own observations indicate that the instability is insidious because it is disguised by smoothing and other effects and so is not apparent for short runs dominated by the growth and saturation of a rapidly growing physical instability; yet the numerical instability can cause trouble in long runs. Based both on these observations and on the lower growth rate for the method used for Sydora's trapped electron paper [15], we expect that his results are not seriously contaminated by the numerical instability. But, we do not recommend the method used in that paper for long runs, since the method is at least weakly unstable. As mentioned above, the work by Lee and his coworkers cited in the references used a code with a stable algorithm. The instability may or may not be operative in other gyrokinetic codes—this depends in detail on the interpolation and electric field algorithms, as will be explained in this paper. In addition, the instability may exist in various other simulation codes that follow particles using guiding center drifts.

The organization of the paper is as follows. In Section 2, we report the characteristics of the numerical instability as observed in our 2D electrostatic gyrokinetic code. In Section 3 a detailed perturbation analysis of the code algorithm is presented. The analysis is extended to many popularly used combinations of interpolation methods and field algorithms. In Section 4 we analytically examine the continuous limit. In Section 5 we give a detailed discussion of the results specifying in detail which methods are stable and which unstable. In Section 6 we give a summary and conclusions and recommendations.

2. OBSERVATIONS FROM THE 2D GYROKINETIC CODE

2.1. *Conditions for Instability vs Stability*

The instability exists with either CIC (linear) [2] or quadratic [2] interpolation when combined with a 2-point finite difference for the electric field.

The subtracted dipole [2] interpolation is perfectly stable, exactly zero error response, when it is combined with a 2-point finite difference for E. In this case, unlike the linear and quadratic interpolation methods, the analytic error response for the subtracted dipole cancels term for term. Despite this favorable stability, our own preference is to avoid a subtracted dipole because of its enhanced noise problems. A subtracted dipole is used in Sydora's 3D electrostatic code, and, we believe, in all of the work cited in Refs. [4–13].

Yet another method for calculating the electric field is to use the analytic derivative of an interpolated potential, as results from "energy conserving" schemes [2]. This produces a finite oscillatory response (stable) for any of the three interpolation methods, CIC, quadratic, or subtracted dipole. This E field method is our present method of choice for avoiding the instability.

The instability is most clearly observed in the limit of cold ions and adiabatic electrons. For particle electrons and cold ions it should cancel out. For finite gyro radius ions and particle electrons, something should remain. For a general thermal ion distribution, we still have to sort out what comes from this instability response and what comes from the particle undersampling noise; changing to a stable algorithm makes a major change in a specific long scaling test run [16].

The instability prevents sensible code operation in perhaps the simplest of all test cases: cold ions and adiabatic electrons, obliquely propagating sound waves, with $\nabla n_0, \nabla T_{e0} = 0$. Clean ion sound waves, with energy well conserved, are observed only with $k_x = k_y$ (plus or minus or both). All other modes exhibit either growth or decay. The instability occurs regardless of the excitation method. Without an explicit initial excitation it starts from machine roundoff; with an explicit excitation it starts at some higher initial level. In either case the instability exhibits clean exponential growth (or decay). It is clearly a linear phenomenon. In general, an oscillation can accompany the growth; i.e., the mode frequency is complex, but this feature is dependent on the location of the particle relative to the grid. Growth rate for the $(k_x, k_y) = (2, -1)$ mode for 16×16 cells is $\gamma/\Omega_{ci} = 0.016$. Saturation nonlinearly occurs, but not at particularly low levels. The instability persists when $k_\parallel = 0$, where the sound wave disappears. So the effect is isolated to the $\mathbf{E} \times \mathbf{B}$ flow. The instability persists when using many particles per cell and it can be shown that it exists in the continuous limit.

The instability is most potent at short wavelength and can be suppressed by a sufficiently strong smoothing factor. But it persists for cold ion tests in a damaging way for 16×16 cells with "typical" smoothing factor $\exp[-(1.7k\Delta)^2]$.

When the instability is present, energy is not conserved at all. This characteristic also separates it from particle undersampling noise for which energy is expected to be conserved. With smoothing, the instability may survive but at a weaker level—raising the possibility of gradual contamination over long times of results that appear to be physically plausible.

2.2. *Summary of Analysis Compared to Observations*

Stability analysis and the code results quoted above agree in all respects. In addition, we have analytically examined other unstable schemes: Combining the CIC or quadratic interpolation with a 6-point difference for E recommended by Birdsall and Langdon [2] reduces the growth rate but is still unstable with either CIC or quadratic interpolation, in the single particle-per-cell limit (the continuous limit is discussed in Section (4) and Appendix B). We have also shown analytically that instability occurs if CIC or quadratic interpolation or subtracted dipole is combined with the electric field obtained from $\mathbf{E}(k) =$

$-i\mathbf{k}\phi(\mathbf{k})$. None of these latter combinations were attempted in the gyrokinetic code.

2.3. *What Is the Cause of the Instability?*

The root cause is a false divergence of $V_{E\times B}$ which leads to a nonzero dn/dt even when $\nabla n_0 = 0$. There is a coupling between the interpolation to and from the particles and the grid that results in a finite change in n, even with an initially uniform distribution of particles. This occurs despite the fact that the finite difference $\mathbf{E} = -\nabla\phi$ does satisfy $\nabla \cdot \mathbf{V}_{E\times B} = 0$, *if* it is evaluated on the grid. But interpolating E to the particle locations and interpolating the new particle locations back to the grid for accumulation of the change in density results in a net finite response even for an initially uniform distribution of particles (a subtracted dipole gives zero response if combined with a centered finite difference for E).

This false finite response is *not* due to undersampling, i.e., not due to using too few particles—it persists when using many particles per cell and it can be shown that it would still exist in the continuous limit. Undersampling is generally responsible for noise problems, but it is not a cause of this instability.

3. PERTURBATION ANALYSIS

3.1. *Outline*

A perturbation analysis of the code algorithm was formulated to check the observations from the gyrokinetic code. The analysis starts with a model of a uniform distribution of particles, with one particle per cell. We develop an expression for the differential change in density caused by the $\mathbf{E} \times \mathbf{B}$ flow from an arbitrary potential. Ideally this change in density should be identically zero, so any finite response represents an error caused by the numerical algorithm. The formulation is general but the detail depends on the choice of the specific interpolation method and of the specific method for calculating the E field.

The immediate result of this analysis is an error stencil, a sum of terms of the potential ϕ_{IJ} over a local region of grid cells. For arbitrary particle location relative to the grid, this error stencil is spread over many grid cells, and each separate ϕ_{IJ} has several factors of products of interpolation weights and their derivatives. The formulation of this error expression is straightforward, but the number of terms can be somewhat daunting: for example, the case of quadratic interpolation with finite difference for E has 324 terms spread over a 7×7 local stencil of ϕ_{IJ}. In order to avoid mistakes, the general analysis was done with Mathematica. Its use here is not very sophisticated. Even a novice user can quickly get up to speed for the way it is used here. It is mainly used to automatically take derivatives of various interpolation formulas and to automatically sum the many terms that can arise.

The next step in the analysis is to relate the change in density to an equivalent change in potential from Poisson's equation as used in the gyrokinetic code; we then have a linear relationship with the only dependent variable being the potential, but involving the inverse of the Poisson operator, and a difference stencil that brings in the value of the potential at many neighboring grid cells. The final step in the analysis is to express the potential as a sum of Fourier modes. Then the many terms can be summed and an expression for the growth rate results for each Fourier mode.

3.2. *Implementation*

The density at the grid points $\mathbf{x}_g$ from a single particle at $\mathbf{x}_p$ is a function W_g of the difference between the particle position and the grid position. The values of the weights, W_g, at the grid points in the neighborhood of the particle are given by the specific interpolation scheme,

$$n_g = W_g(\mathbf{x}_p - \mathbf{x}_g). \tag{1}$$

A single particle will contribute to the density at several neighboring grid points: 4 (CIC), 5 (subtracted dipole), or 9 (quadratic). Thus, for one particle per cell, the density at a given grid point will get contributions from several particles, depending on the interpolation scheme. The differential change in density from a differential change in these several particle positions, $\boldsymbol{\xi}_p$ is

$$\delta n_g = \sum_p + \boldsymbol{\xi}_p \cdot \frac{\partial}{\partial \mathbf{x}_p} W_{g'}(\mathbf{x}_p - \mathbf{x}_{g'}). \tag{2}$$

The electric field that determines the particle motion is then given by the same interpolation formula applied to the fields at neighboring grid points:

$$\frac{d}{dt}\xi_p = \mathbf{E} \times \hat{\mathbf{b}}/B = \sum_g \mathbf{E}_g W_g(x_p - x_g) \times \hat{\mathbf{b}}/B. \tag{3}$$

So we have for the time derivative of the net change in density,

$$\frac{d\delta n_g}{dt} = \sum_p \left[\sum_g \mathbf{E}_g \times \hat{\mathbf{b}}/B W(\mathbf{x}_p - \mathbf{x}_g) \right]_p \frac{\partial}{\partial \mathbf{x}_p} W_{g'}(\mathbf{x}_p - \mathbf{x}_{g'}), \tag{4}$$

where the subscript $_p$ on the square bracket term emphasizes that the specific grid points referenced by the Σ_g depend on the cell location of the particular particle p. And, since we are here dealing with the model case of one particle per cell, the Σ_p can be replaced with a sum over grid points Σ'_g:

$$\frac{d\delta n_g}{dt} = \sum_{g'} \left[\sum_g \mathbf{E}_g \times \hat{\mathbf{b}}/B W(\mathbf{x}_p - \mathbf{x}_g) \right]_{g'} \frac{\partial}{\partial \mathbf{x}_p} W_{g'}(\mathbf{x}_p - \mathbf{x}_{g'}). \tag{5}$$

To be specific, we here give the detailed subscripted form for each particle contributing to our reference grid cell 0,0 for the case of CIC interpolation,

$$\xi_{0,0}\colon [W_{0,0}E_{0,0} + W_{0,1}E_{0,1} + W_{1,0}E_{1,0} + W_{1,1}E_{1,1}]\frac{\partial}{\partial x_p}W_{0,0}$$

$$\xi_{-1,0}\colon [W_{0,0}E_{-1,0} + W_{0,1}E_{-1,1} + W_{1,0}E_{0,0} + W_{1,1}E_{0,1}]\frac{\partial}{\partial x_p}W_{-1,0}$$

$$\xi_{0,-1}\colon [W_{0,0}E_{0,-1} + W_{0,1}E_{0,0} + W_{1,0}E_{1,-1} + W_{1,1}E_{1,0}]\frac{\partial}{\partial x_p}W_{0,-1}$$

$$\xi_{-1,-1}\colon [W_{0,0}E_{-1,-1} + W_{0,1}E_{-1,0} + W_{1,0}E_{0,-1} + W_{1,1}E_{0,0}]\frac{\partial}{\partial x_p}W_{-1,-1}$$

and where the negative subscripted weights have the identification:

$$W_{-1,0} = W_{1,0}$$
$$W_{0,-1} = W_{0,1}$$
$$W_{-1,-1} = W_{1,1}.$$

The precise form of the CIC weights W are not given here. Other interpolation methods will differ in the detail of the form of the weights and in the numbers of grid points used, but otherwise the description is the same.

3.3. *Number of ϕ_{ij} Terms in the Error Expression*

In general a particle can contribute to an arbitrary number of grid points, but in practice it is usually a small number, defined by the interpolation scheme. Thus, for one particle/cell, a single grid point will have density contributions from varying numbers of particles, dependent on the interpolation scheme:

4 (CIC),
5 (subtracted dipole), or
9 (quadratic).

Similarly, the displacements of a single particle will have contributions from the electric field at several neighboring grid points; the number is dependent in the same way on the interpolation scheme. The displacements in the X and Y directions bring in another factor of 2. The finite difference for $\mathbf{E} = -\nabla\phi$ brings in yet another factor of 2:

4	×	4	×	2	×	2	= 64 total CIC
particles		grid points		X, Y		finite diff	
9	×	9	×	2	×	2	= 324 total quadratic.

3.4. *Generation of the Error Stencil*

We now have developed the analysis far enough so that the error stencil can be generated; after substituting for the particular algorithm for E, it is simply the r.h.s. double sum over a local region of ϕ_{ij}. For arbitrary particle location relative to the grid, this sum will involve many terms. The Appendix gives the full error stencil for CIC with finite difference for E. Before expanding the weight factors there a total of 64 terms are spread over a stencil of 5 × 5 grid points. Expanding the weights cancels some terms but adds others to give a total of 88 terms. In this form the expression is a function of the particle location relative to the grid, dx, dy. In this form it is not very illuminating, except that various symmetries can be observed. For particular particle locations, this error stencil simplifies greatly; e.g., for the particle located at the exact midpoint between grid points, $dx, dy = 0.5, 0.5$ we have a sum of only eight terms and all have a coefficient of ± 0.0625:

$$\frac{d\delta n_g}{dt} = 0.0625\begin{pmatrix} 0 & -\phi[-1,2] & 0 & +\phi[1,2] & 0 \\ +\phi[-2,1] & 0 & 0 & 0 & -\phi[2,1] \\ 0 & 0 & 0 & 0 & 0 \\ -\phi[-2,-1] & 0 & 0 & 0 & +\phi[2,-1] \\ 0 & +\phi[-1,-2] & 0 & -\phi[1,-2] & 0 \end{pmatrix}. \tag{6}$$

3.5. *Calculation of the Growth Rate*

For the midpoint particle location we get exactly the same answer for quadratic interpolation. This stencil is much simpler than the general case (64 terms for CIC, 324 terms for quadratic), but even here, to go further we have to relate the density on the l.h.s. to the potential, which of course is given by the Poisson equation here, as used in the gyrokinetic code,

$$\frac{\omega_{pi}^2}{\Omega_{ci}^2}(-\nabla_\perp^2 + 1/\rho_s^2)\phi = \delta ne/\varepsilon_0. \tag{7}$$

Substituting from this for $\delta n = \delta n_g$ into our error expression gives a relationship involving only the potential:

$$\frac{d\phi}{d t \Omega_{ci}} = \sum_{g'} \left[\sum_{g} - \frac{\nabla \phi_g \times \hat{\mathbf{b}}}{(-\nabla_\perp^2 + 1/\rho_s^2)} W(\mathbf{x}_p - \mathbf{x}_g) \right]_{g'} \frac{\partial}{\partial \mathbf{x}_p} W_{g'}(\mathbf{x}_p - \mathbf{x}_{g'}). \tag{8}$$

Here, the ∇ operator depends on the choice of method for calculating $\mathbf{E} = -\nabla\phi$, either (1) a finite difference operator on the ϕ_g (as was used for the midpoint particle limit shown above), or (2) an analytic derivative operating on the $W(x_p - x_g)$, or (3) $-i\mathbf{k}$, operating on the value of $\phi(x, y)$ at the grid points relative to the local value of x, y, as are specified by the interpolation formula,

$$\phi_g = \phi(x + m\Delta, y + n\Delta) = \phi(x + m, y + n), \tag{9}$$

where we have normalized the grid size $\Delta = 1$. Now replace $\phi(x, y)$ with its Fourier representation,

$$\begin{aligned} \phi(x, y) &= s_k \sin(k_x x + k_y y) + c_k \cos(k_x x + k_y y) \\ &= s_k S + c_k C \end{aligned} \tag{10}$$

and for the potential displaced at various grid points,

$$\begin{aligned} \phi_g = {} & s_k \sin(k_x x + k_y y + k_x m + k_y n) \\ & + c_k \cos(k_x x + k_y y + k_x m + k_y n) \end{aligned} \tag{11}$$

or

$$\begin{aligned} \phi_g = {} & s_k S \cos(k_x m + k_y n) + s_k C \sin(k_x m + k_y n) \\ & + c_k C \cos(k_x m + k_y n) - c_k S \sin(k_x m + k_y n). \end{aligned} \tag{12}$$

Substituting into our expression for the time rate of change of the potential and separately collecting terms for s_k and c_k, we have

$$\frac{d}{dt} S = S \sum_{m,n} \cos(k_x m + k_y n) + C \sum_{m,n} \sin(k_x m + k_y n) \tag{13}$$

$$\frac{d}{dt} C = C \sum_{m,n} \cos(k_x m + k_y n) - S \sum_{m,n} \sin(k_x m + k_y n), \tag{14}$$

where the m,n indices result from the $\Sigma_g \phi_g$ over neighboring grid points as dictated by the interpolation formula and the finite difference (if used). Now by evaluating these sums for specific modes we can evaluate the mode complex frequency $p = i\omega$,

$$\left(p - \sum_{m,n} \cos \right) S - \left(\sum_{m,n} \sin \right) C = 0 \tag{15}$$

$$\left(\sum_{m,n} \sin \right) S + \left(p - \sum_{m,n} \cos \right) C = 0,$$

so the mode growth rate is given by solution of

$$\left(p - \sum_{m,n} \cos \right)^2 + \left(\sum_{m,n} \sin \right)^2 = 0. \tag{16}$$

Example for the midpoint particle location for CIC interpolation and 2-point finite difference E: In this case the $\Sigma_{m,n} \sin = 0$, which means the oscillatory part is zero, and the $\Sigma_{m,n} \cos$ is given by the above eight-term stencil, which can be rewritten as

$$\sum_{m,n} \cos = 0.25 (\sin 2k_x \sin k_y - \sin k_x \sin 2k_y). \tag{17}$$

Including the Poisson operator coefficient we have

$$\gamma/\Omega_{ci} = 0.25 \frac{\sin 2k_x \sin k_y - \sin k_x \sin 2k_y}{k_x^2 + k_y^2 + \rho_s^{-2}}. \tag{18}$$

4. CONTINUOUS LIMIT

4.1. *Integration of Single Particle per Cell Results*

The results for the limit of a continuous distribution of particles can be obtained simply by integrating over a single grid cell $\int dxdy$ the Mathematica general result for the single particle-per-cell. This can be done easily within Mathematica; the integrations are trivial, the only complication being the many terms. The result for CIC interpolation with two-point finite difference E is the same error stencil as for the 0.5, 0.5 single particle case but modified by a factor $\frac{2}{3}$. The result for quadratic interpolation is more complex but close to the same result. A variety of combinations of interpolation methods and field methods have been examined and detailed results obtained for the growth rate dependence. The results obtained by integration of the single particle per cell results have been shown to be identical to those obtained by a formal derivation starting with a continuous distribution, which will now be given.

4.2. *Formal Derivation of a Continuous Limit*

The continuous limit can also be approached analytically without having to keep track of numerous terms that arise from the single particle-per-cell limit. The result of this section gives a formal derivation of the continuous limit. Results obtained thereby agree in all details from those obtained from the integral forms of the Mathematica results.

We here derive the dispersion relation for the numerical instability for a uniform distribution of particles. As for the single particle per cell case, the polarization density [8] is kept, but the parallel motion and FLR effects are neglected in the following calculation.

The density of simulation particles, interpolated onto grid point $\mathbf{x}_g$, is given formally by

$$n_g = \sum_i W_g(\mathbf{x}_g - \mathbf{x}_i), \tag{19}$$

where W_g is the weighting function and $\mathbf{x}_i$ is the position of the ith particle. The first-order perturbation in this density due to a flow $\mathbf{V}_i$ of the particle positions is given by

$$\frac{\partial}{\partial t}\delta n_g(t) = \sum_i \mathbf{V}(\mathbf{x}_{i0}, t)\cdot\nabla_\perp W_g(\mathbf{x}_g - \mathbf{x})|\mathbf{x} = \mathbf{x}_{i0}, \tag{20}$$

where $\mathbf{x}_{i0}$ is the zero-order position of the ith particle. Now suppose that the $\mathbf{x}_{i0}$'s are uniformly distributed. Then the particle sum can be replaced by an integral.

$$\frac{\partial}{\partial t}\delta n_g(t) \to \int d\mathbf{x}\mathbf{V}(\mathbf{x}, t)\cdot\nabla_\perp W_g(\mathbf{x}_g - \mathbf{x}). \tag{21}$$

Upon integration by parts, we obtain

$$\frac{\partial}{\partial t}\delta n_g(t) \to -\int d\mathbf{x}W_g(\mathbf{x}_g - \mathbf{x})\nabla_\perp\cdot\mathbf{V}(\mathbf{x}), \tag{22}$$

where the time dependence of $\mathbf{V}(\mathbf{x}, t)$ has been suppressed. From Eq. (22), it is seen that if the perpendicular flow $\mathbf{V}$ is incompressible, as is the case when $\mathbf{V}$ is calculated by analytically differentiating an interpolated potential [2], then a uniform particle distribution gives no density response.

In what follows, we consider the case where $\mathbf{V}$ represents an $\mathbf{E}\times\mathbf{B}$ flow and is calculated at the particle positions by interpolating between the values of the $\mathbf{E}\times\mathbf{B}$ flow at nearby grid points. An instability can result when the perpendicular flow is compressible, as a result of the interpolation scheme used. Consider, therefore, the flow

$$\mathbf{V}(\mathbf{x}) \equiv \sum_g W(\mathbf{x} - \mathbf{x}_g)\mathbf{V}(\mathbf{x}_g),$$

where

$$\mathbf{V}(\mathbf{x}_g) \equiv -\hat{\mathbf{b}}\times\mathbf{E}(\mathbf{x}_g)/B,$$

$\mathbf{E}$ is the electric field, the sum is over grid points g, and W is the field interpolation function. The field is calculated by a grid operation (finite-differencing, or differentiation of a discrete-Fourier representation) of the potential, denoted $\Delta\phi$. Thus we have

$$\frac{\partial}{\partial t}\delta n_g = -\hat{\mathbf{b}}\times\sum_{g'}\Delta\phi_{g'}\cdot\int d\mathbf{x}W_g(\mathbf{x}_g - \mathbf{x})\nabla_\perp W(\mathbf{x} - \mathbf{x}_{g'}). \tag{23}$$

Upon taking the discrete Fourier transform (i.e., applying the operator $(1/N_g)\sum_g \exp(-i\mathbf{k}\cdot\mathbf{x}_g)$, where N_g is the total number of grid points in the system, and the boundary conditions have been taken to be periodic both in the x and y directions) of Eq. (23), and anticipating that the Fourier modes are eigenmodes, we obtain

$$\gamma_k\delta n_k = -\hat{\mathbf{b}}\times\mathbf{D}\cdot\mathbf{U}\phi_k/B, \tag{24a}$$

where

$$\mathbf{D} \equiv [\Delta\exp(i\mathbf{k}\cdot\mathbf{x})]|\mathbf{x} = \mathbf{o} \tag{24b}$$

and

$$\mathbf{U} \equiv \sum_g \exp(-i\mathbf{k}\cdot\mathbf{x}_g)\int d\mathbf{x}W(\mathbf{x}_g - \mathbf{x})\nabla_\perp W(\mathbf{x}). \tag{24c}$$

We now specialize to the case of a rectangular grid and separable "momentum-conserving" choices of the weighting functions $W_g(\mathbf{x}) = W(\mathbf{x}) = W_x(x/\Delta_x)W_y(y/\Delta y)$. This gives

$$\mathbf{U} = [\Delta_y U_p(k_x\Delta_x)U(k_y\Delta_y),\ \Delta_x U(k_x\Delta_x)U_p(k_y\Delta_y)],$$

where

$$U(\xi) \equiv \sum_{j=-\infty}^{\infty}\exp(-ij\xi)\int_{-\infty}^{\infty} dxW(j - x)W(x),$$

$$U_p(\xi) \equiv \sum_{j=-\infty}^{\infty}\exp(-ij\xi)\int_{-\infty}^{\infty} dxW(j - x)W'(x).$$

All of the discretizations of the derivative operator considered give a result of the form

$$\mathbf{D} = [D(k_x\Delta_x)D_1(k_y\Delta_y)/\Delta_x,\ D_1(k_x\Delta_x)D(k_y\Delta_y)/\Delta_y]$$

so that we obtain

$$\gamma_k\delta n_k = [D_1(k_x\Delta_x)U_p(k_x\Delta_x)D(k_y\Delta_y)U(k_y\Delta_y) - D(k_x\Delta_x)U(k_x\Delta_x)D_1(k_y\Delta_y)U_p(k_y\Delta_y)]\phi_k/B.$$

Inserting this into the gyrokinetic Poisson equation, Eq. (7) gives, finally, the dispersion relation for the perturbations associated with the grid errors

$$\frac{\gamma_k}{\Omega_{ci}} = [D_1(k_x\Delta_x)U_p(k_x\Delta_x)D(k_y\Delta_y)U(k_y\Delta_y) - D(k_x\Delta_x)U(k_x\Delta_x)D_1(k_y\Delta_y)U_p(k_y\Delta_y)]/[1 + k_\perp^2\rho_s^2].$$

Thus we need to evaluate only the four single-variable functions $D_1(\xi)$, $D(\xi)$, $U_p(\xi)$, and $U(\xi)$. For most common discretizations of the derivative operator, we have $D_1 = 1$. The functions U_p and U are sums of a small number of exponentials, the

coefficients of which are determined by analytically evaluating simple one-dimensional integrals.

In Appendix B, an alternative derivation of the dispersion relation is given. The results given there can be obtained by inserting Fourier integral representations for W_g and W into Eq. (24c) and performing the integration with respect to **x**.

Equation (24a) also gives a prescription for obtaining a differencing scheme for the field caculation that will be stable for any given interpolation scheme. If **D** *and* **U** *are parallel, then Eq. (24a) implies that* $\gamma_k \delta n_k = 0$. *The choice of interpolation scheme determines* **U**. *One then constructs a differencing operator such that* **D** *is parallel to* **U**. *For the separable cases, for example, the scheme will be stable if one chooses* $D_l(\xi) = U(\xi)$ *and* $D(\xi) = U_p(\xi)$.

4.3. *Results from the Continuous Limit Derivation*

We will consider three discretizations of the derivative operator:

(1) Differentiation of the Fourier representation,

$$D(\xi) = i\xi,$$
$$D_l(\xi) = 1;$$

(2) Two-point centered difference,

$$D(\xi) = i \sin \xi,$$
$$D_l(\xi) = 1;$$

(3) Six-point scheme recommended by Birdsall and Langdon [2]

$$D(\xi) = i \sin \xi,$$
$$D_l(\xi) = (2 + \cos \xi)/3.$$

The following choices are considered for the weighting function W:

(1) CIC:

$$U(\xi) = (2 + \cos \xi)/3,$$
$$U_p(\xi) = i \sin \xi.$$

In combination with the centered-difference discretization of the derivative, CIC weighting gives

$$\frac{\gamma_k}{\Omega_{ci}} = \frac{1}{3} \frac{1}{1 + k_\perp^2 \rho_s^2} \sin(k_x \Delta_x) \sin(k_y \Delta_y)[\cos(k_y \Delta_y) - \cos((k_x \Delta_x)].$$

Expanded to fourth order in $k\Delta$, this becomes

$$\frac{\gamma_k}{\Omega_{ci}} = \frac{1}{6} \frac{1}{1 + k_\perp^2 \rho_s^2} (k_x \Delta_x)(k_y \Delta_y)[(k_y \Delta_y)^2 - (k_x \Delta_x)^2].$$

In combination with the derivative calculated by differentiating the discrete Fourier representation, CIC weighting gives

$$\frac{\gamma_k}{\Omega_{ci}} = \frac{1}{3} \frac{1}{1 + k_\perp^2 \rho_s^2} \{(k_x \Delta_x) \sin(k_y \Delta_y)[2 + \cos(k_x \Delta_x)] - (k_y \Delta_y) \sin(k_x \Delta_x)[2 + \cos(k_y \Delta_y)]\}.$$

This gives zero growth to fourth order in $k\Delta$. It is of interest to evaluate this result for $k\Delta$ of order 1. For $k_x \Delta_x = \pi/2$, $k_y \Delta_y = \pi/4$, we have

$$\gamma/\Omega_{ci} = \frac{\pi}{24} \frac{3\sqrt{2} - 4}{(1 + 5\pi^2/16)} \approx 0.00778. \tag{25}$$

In combination with the Birdsall–Langdon six-point difference operator, the CIC weighting gives an identically zero growth rate.

(2) Quadratic spline:

$$U(\xi) = (132 + 104 \cos \xi + 4 \cos 2\xi)/240 \simeq 1 - \xi^2/4,$$
$$U_p(\xi) = i \sin \xi(5 + \cos \xi)/6 \simeq i\xi(1 - \xi^2/12).$$

In combination with the centered-difference discretization of the derivative, quadratic-spline weighting gives

$$\frac{\gamma_k}{\Omega_{ci}} = \frac{1}{1440(1 + k_\perp^2 \rho_s^2)} \sin(k_x \Delta_x) \sin(k_y \Delta_y)\{[5 + \cos(k_y \Delta_y)][132 + 104 \cos(k_x \Delta_x) + 4 \cos(2k_x \Delta_x)] - [y \leftrightarrow x]\}.$$

This expression can be manipulated to become identical to the form derived from the integral of the single particle results. Expanded to fourth order in $k\Delta$, this gives the same result as CIC weighting, given above. In combination with the derivative calculated by differentiating the discrete Fourier representation, *quadratic-spline weighting gives zero growth rate to fourth* order in $k\Delta$. In combination with the Birdsall–Langdon six-point difference operator, quadratic-spline weighting gives zero growth rate through fourth order in $k\Delta$.

All of the continuous limit results obtained from the formal derivation agree in detail with those obtained from the integration of the Mathematica single particle per cell results. All analytical results are collected together in the tables at the end of Section 5.

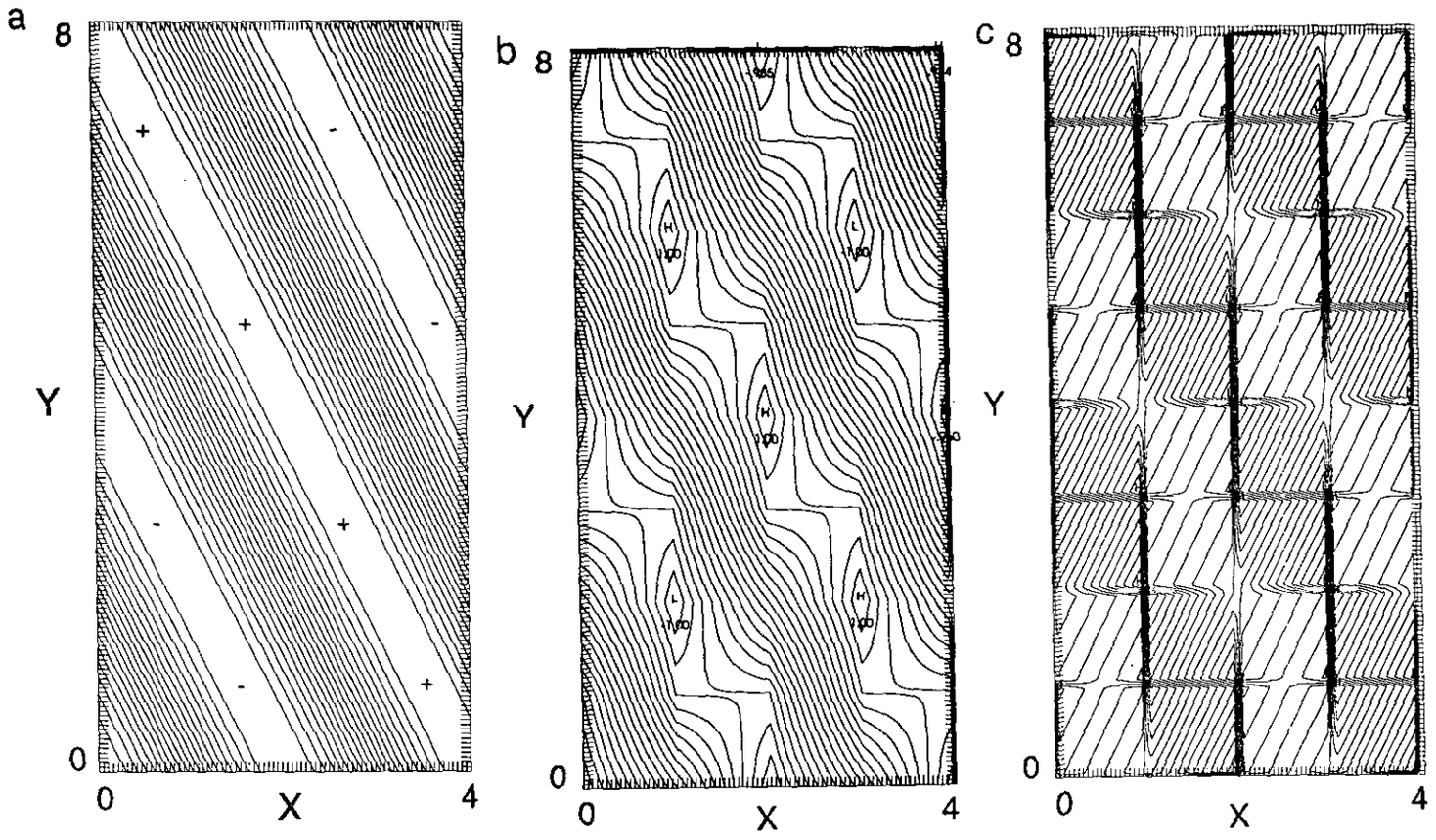

FIG. 1. Contours of (a) $\mathbf{v}(\mathbf{x})$, (b) $v_{\text{int}}(\mathbf{x})$, (c) $\nabla_\perp \cdot (v_{\text{int}}(\mathbf{x})\hat{\mathbf{v}})$ for $k_x = \pi/2$, $k_y = \pi/4$.

4.4. *Simple Picture of Unstable Eigenmode*

In order to give the reader some intuitive understanding of the instability mechanism, we give a picture of the flows involved. We treat the simplest case: CIC interpolation with $\mathbf{E}_\mathbf{k} = -i\mathbf{k}\phi_\mathbf{k}$ used for the field calculation at the grid points. This case is not the most unstable method, but it is perhaps the easiest to understand because it preserves the direction of the analytic flow. We choose the unstable mode $\mathbf{k} = (2\pi/4, 2\pi/8)$ (in units in which $\Delta_x = \Delta_y = 1$). We specify the phase by the choice $\phi = \sin[\pi(x/2 + y/4)]$. Then the $\mathbf{E} \times \mathbf{B}$ velocity calculated at the grid points is given by $\mathbf{v}_{EB} \equiv \hat{\mathbf{b}} \times \nabla_\perp \phi/B = v(\mathbf{x}_g)\hat{\mathbf{v}}$, where $v(\mathbf{x}) = \cos[\pi(x/2 + y/4)]$ and $\hat{\mathbf{v}} = (-1/\sqrt{5}, 2/\sqrt{5})$, and the subscript g denotes the coordinates of a grid point. The contours of $\mathbf{v}(\mathbf{x})$ are shown in Fig. 1(a). Now a simulation particle at $\mathbf{x}$ not on a grid point does *not* move with velocity $v(\mathbf{x})\hat{\mathbf{v}}$, but rather with velocity $v_{\text{int}}(\mathbf{x})\hat{\mathbf{v}}$, where

$$v_{\text{int}}(\mathbf{x}) \equiv \sum_g W(\mathbf{x} - \mathbf{x}_g)v(x_g).$$

The contours of v_{int}, with W given by the CIC interpolation function are shown in Fig. 1(b). It can be seen that there are now variations parallel as well as orthogonal to $\hat{\mathbf{v}}$, so a nonzero divergent flow is evident. The contours of the $\nabla_\perp \cdot (v_{\text{int}}\hat{\mathbf{v}})$ are shown in Fig. 1(c), and are seen to be discontinuous across the cell boundaries, a characteristic expected from the CIC interpolation method. These contours can be understood from Fig. 1(b) by noting that $\nabla_\perp \cdot (v_{\text{int}}\hat{\mathbf{v}}) = \hat{\mathbf{v}} \cdot \nabla_\perp v_{\text{int}}$, since $\hat{\mathbf{v}}$ is independent of position. If v_{int} is increasing in the direction of $\hat{\mathbf{v}}$, then $\nabla_\perp \cdot (v_{\text{int}}\hat{\mathbf{v}})$ is positive. We see from Fig. 1(c) that $\nabla_\perp \cdot (v_{\text{int}}\hat{\mathbf{v}})$, deposited onto the grid points, is 180° out of phase with ϕ. Using the linearized continuity equation $\partial n_g/\partial t = -n_0 \int d\mathbf{x} W(\mathbf{x}_g - \mathbf{x})\hat{\mathbf{v}} \cdot \nabla_\perp v_{\text{int}}(\mathbf{x})$ and $n = \alpha\phi$, where α is real and positive, we see that the change in the potential produced by the seed potential is in phase with the seed potential, resulting in instability.

5. SUMMARY OF STABILITY DEPENDENCE ON THE NUMERICAL ALGORITHMS

The perturbation analysis of the code algorithm predicts the instability–stability characteristics and agrees with the observations from the gyrokinetic code in *all* respects. The predictions of the analysis agrees within measurement accuracy from the code results for the magnitude of growth rate and real frequency. Similarly both agree in detail on the presence of unstable growth

(decay) or a stable oscillation or an identically zero error response. Another feature clearly evident in both the code and the analysis is the dependence of the stability characteristics on the particle position relative to grid in the limit of one particle per cell. Perhaps the most critical item with respect to the particle position is the behavior in the limit of a continuous distribution of particles (for the analysis) and the limit of many particles per cell (for the code). Both the analysis and the code show persistence of instability in these limits; and they both show reduction to zero error of the oscillatory (stable) methods in these limits. Some specialized methods (examined analytically only) have a finite growth rate for a single particle per cell but reduce to zero or very small growth rate in the continuous limit. The analysis and code also show detailed agreement of the stability dependence on mode number, k_x, k_y. The method originally used in the code, I2—E1A, has growth rate scaling as

$$\gamma \approx [k_x k_y^3 - k_y k_x^3] + .../(k_x^2 + k_y^2 + 1/\rho_s^2).$$

Thus the mode $(k_x, k_y) = (2, 1)$ has negative growth rate (damping) while the mode $(k_x, k_y) = (2, -1)$ has positive growth rate. Asymmetry with respect to the sign of the k_y mode number also has been observed by Lodestro [16] in long noise runs with the LLNL code using the unstable algorithm. Not all methods analyzed were also put into the gyrokinetic code. We indicate below which methods were examined analytically only.

The stability characteristics depend in detail on the particle to grid interpolation methods and on the method for calculating the field at the particles. We examined three basic interpolation methods analytically and with the code stability variation:

(I1) CIC (bilinear) [2]

(I2) quadratic [2]. This has been the standard interpolation in the LLNL 2D gyrokinetic code.

(I3A) subtracted dipole [2]. This method was used in all of the work cited by Lee and co-workers.

(I3B) Kruer-modified subtracted dipole [2]. This method has a smoother interpolation than I3A, the standard subtracted dipole, and is expected to be less noisy. Its stability characteristics will be seen to be nearly identical to I3A. Examined analytically only.

(I3C) Nevins–Langdon improved subtracted dipole [2]. This is yet another version of subtracted dipole with even smoother interpolation; its noise characteristics should improve over I3A and probably over I3B. But its stability characteristics will be seen to be similar to those of I1 or I2, unlike the other subtracted dipole methods. Examined analytically only.

We also examined stability variation with three methods of calculating E at the particles:

(E1A) Two-point finite difference of ϕ (standard in the LLNL 2D gyrokinetic code) calculated at the grid points; this E is then interpolated to the particles. Here, any of (I1) CIC or (I2) quadratic or (I3C) NL interpolation produces instability. Subtracted dipole I3A or I3B is stable—each term cancels identically to give zero false response. Instability persists for many particles/cell for any of (I1) CIC or (I2) quadratic or (I3C) NL interpolation. The growth rate observed in the gyrokinetic code asymptotes quickly as a function of the numbers of particles/cell.

TABLE I

Error Response from Divergence of $E \times B$

	E field		
Interpolation	Finite diff.	$\partial/\partial_x\phi_p$	$ik\phi_k$
I1 CIC	Grows	Osc[a]	Grows
I2 Quadratic	Grows	Osc[a]	Grows
I3A Subdipole	Zero	Zero	Grows
I3B SubdK[b]	Zero	Zero	Grows

[a] Osc = finite oscillatory (stable) response; approaches zero response in continuous limit.

[b] SubdK = Kruer improved subtracted dipole.

(E1B) Six-point difference, recommended by Birdsall and Langdon. Examined analytically only. Interpolation methods I1 and I2 in the single particle per cell limit remain unstable at a rate $\frac{1}{3}$ of the peak single particle rate. But the continuous limit is much different; method I1 now has identically zero growth rate, and I2 has a finite but strongly reduced growth rate. On the other hand, this 6-point difference scheme destroys the perfect zero error response obtained by the subtracted dipole methods I3A and I3B when combined with the 2-point difference. We have derived a modified 6-point difference for E, method E1Bm, that recovers the perfect zero response when combined with interpolation methods I3A and I3B:

$$\text{E1B:}\quad \partial_x\phi = \frac{\phi(I,J)}{12\Delta x}\begin{pmatrix} -1 & 0 & +1 \\ -4 & 0 & +4 \\ -1 & 0 & +1 \end{pmatrix} \tag{26}$$

$$\text{E1Bm:}\quad \partial_x\phi = \frac{\xi(I,J)}{24\Delta x}\begin{pmatrix} -1 & 0 & 0 & 0 & +1 \\ 0 & -8 & 0 & +8 & 0 \\ -1 & 0 & 0 & 0 & +1 \end{pmatrix}. \tag{27}$$

(E2) analytic derivative of the interpolated ϕ at the particle. Each interpolation weight is analytically differentiated:

$$\mathbf{E}_p = -\sum_g \phi_g \frac{\partial}{\partial \mathbf{x}} W(\mathbf{x} - \mathbf{x}_g)|_{x=x_p}. \tag{28}$$

Any of (I1) CIC, (I2) quadratic, or (I3C) NL subtracted dipole

TABLE II

Growth Rate as Function of k_x, k_y for the Midcell Particle Position

$\frac{\gamma}{\Omega_{ci}} =$

k_y	k_x = 1	2	3	4	5	6	7	8
8	0	0	0	0	0	0	0	0
7	+.015	+.024	+.023	+.016	+.008	+.002	0	0
6	+.032	+.049	+.044	+.027	+.010	0	−.002	0
5	+.045	+.064	+.052	+.024	0	−.010	−.008	0
4	+.048	+.061	+.036	0	−.024	−.027	−.016	0
3	+.037	+.035	0	−.036	−.052	−.044	−.023	0
2	+.016	0	−.035	−.061	−.064	−.049	−.024	0
1	0	−.016	−.037	−.048	−.045	−.032	−.015	0

Note. For either CIC or quadratic interpolation and 2-point finite difference *E*.

interpolation methods produces pure oscillation. Subtracted dipole (I3A) or (I3B) has identically zero response. The oscillation response for (I1), (I2), or (I3C) approaches zero in the limit of a continuous uniform distribution of particles. Analytic proof of this statement results from the formalism in the previous section where it was shown that any $\mathbf{V}$ satisfying $\nabla \cdot \mathbf{V} = \mathbf{0}$ gives an identically zero response in the continuous limit. In addition, the gyrokinetic code and the analysis show zero response for more than one particle per cell for certain symmetric distributions of particles, which are not uniform for any one set, but which when added to other symmetric sets would be uniform, fully consistent with the analytic proof of a zero response in the continuous limit.

(E3) $\mathbf{E}(k) = i\mathbf{k}\phi(\mathbf{k})$; $\mathbf{E}_g$ at the grid is obtained from an inverse FFT and this $\mathbf{E}_g$ is then interpolated to the particles. Examined analytically only. Instability again occurs, now for all five interpolation methods, (I1) CIC, (I2) quadratic, and (I3), (I3B), (I3C) subtracted dipole. In the continuous limit, growth rates are strongly reduced but none of them are perfectly stable.

Summary of instability characteristics: Interpolation methods I1, I2, and I3C are unstable when coupled with E field methods E1 or E3. All three of I1, I2, and I3C are oscillatory (stable) with E2. This oscillatory response reduces to zero response in the continuous limit. Interpolation methods I3A or I3B give identically zero error response (stable) with E1A or E2; but I3A or I3B are unstable with the 6-point difference E1B; these methods require a modified form of the 6-point difference, E1Bm, to retain the identically zero response. Also I3A or I3B is unstable with E3.

Our recommendation is I1—E2 or I2—E2. The subtracted dipole combinations I3A, I3B—E1A or I3A, I3B—E2 or I3A, I3B—E1Bm also are free of this instability, but they can be expected to be noisier.

Certain combinations are stable or very small growth rate in the continuous limit but have finite growth rate otherwise; it is possible that such combinations will give practical stability, but we cannot recommend them as being completely free of the instability. See Table I for a partial summary chart.

TABLE IIIa

Growth Rate $\gamma(k_x, k_y)$ Dependence on Method CIC (I1), Quadratic (I2)—Single Particle per Cell

	$\hat{\gamma} = \max\left(\frac{\gamma}{\Omega_{ci}}\right)(k_x^2 + k_y^2 + \rho_s^{-2})$
I1, I2—E1A:	$\hat{\gamma} = (\sin 2k_x \sin k_y - \sin k_x \sin 2k_y)/4$ $\approx (k_x k_y^3 - k_x^3 k_y)/4$
I1, I2—E1B:	$\hat{\gamma} = (\sin 2k_x \sin k_y - \sin k_x \sin 2k_y)/12$ $\approx (k_x k_y^3 - k_x^3 k_y)/12$
I1—E2:	$\hat{\gamma} \equiv 0$; $\hat{\omega} = 2[\sin(k_x - k_y) - \sin k_x + \sin k_y]$ $\approx k_x^2 k_y - k_x k_y^2$
I2—E2:	$\hat{\gamma} \equiv 0$; $\hat{\omega} = \{312 \sin(k_x - k_y) - 234[\sin k_x - \sin k_y]$ $- 27[\sin 2k_x - \sin 2k_y] - 21[\sin(k_x - 2k_y)$ $+ \sin(2k_x - k_y)] - 57[\sin(k_x + 2k_y) - \sin(2k_x + k_y)]$ $- 9 \sin(2k_x - 2k_y)\}/1024$ $\approx (k_x^4 k_y - k_x k_y^4)3/64$
I1, I2—E3:	$\hat{\gamma} = 1/2[k_x \sin k_y - k_y \sin k_x]$ $+ 1/4[(k_x - k_y)\sin(k_x + k_y) - (k_x + k_y)\sin(k_x - k_y)]$ $\approx (k_x k_y^3 - k_x^3 k_y)/12$

TABLE IIIb

Growth Rate $\gamma(k_x, k_y)$ Dependence on Method Subtracted Dipole(I3)—Single Particle per Cell

	$\hat{\gamma} = \max\left(\frac{\gamma}{\Omega_{ci}}\right)(k_x^2 + k_y^2 + \rho_s^{-2})$
I3A, I3B—E1A:	$\hat{\gamma} \equiv 0$; $\hat{\omega} \equiv 0$
I3C—E1A:	$\hat{\gamma} = (\sin 2k_x \sin k_y - \sin k_x \sin 2k_y)/4$ $\approx (k_x k_y^3 - k_x^3 k_y)/4$
I3A, I3B—E1B:	$\hat{\gamma} = (\sin 2k_x \sin k_y - \sin k_x \sin 2k_y)/12$ $\approx (k_x k_y^3 - k_x^3 k_y)/12$
I3A, I3B—E1Bm:	$\hat{\gamma} \equiv 0$; $\hat{\omega} \equiv 0$
I3A, I3B—E2:	$\hat{\gamma} \equiv 0$; $\hat{\omega} \equiv 0$
I3A, I3B—E3:	$\hat{\gamma} = k_x \sin k_y - k_y \sin k_x$ $\approx (k_x^3 k_y - k_x k_y^3)/6$

TABLE IIIc

Growth Rate $\gamma(k_x, k_y)$ Dependence on Method CIC (I1), Quadratic (I2)—Continuous Limit

	$\tilde{\gamma} = \frac{\gamma}{\Omega_{ci}}(k_x^2 + k_y^2 + \rho_x^{-2})$
I1—E1A:	$\tilde{\gamma} = (\sin 2k_x \sin k_y - \sin k_x \sin 2k_y)/6$ $\approx (k_x k_y^3 - k_x^3 k_y)/6$
I2—E1A:	$\tilde{\gamma} = [195(\sin 2k_x \sin k_y - \sin k_x \sin 2k_y)$ $+ 10(\sin 3k_x \sin k_y - \sin k_x \sin 3k_y)$ $+ 1(\sin 3k_x \sin 2k_y - \sin 2k_x \sin 3k_y)]/1440$ $\approx (k_x k_y^3 - k_x^3 k_y)/6$
I1—E1B:	$\tilde{\gamma} = 0$ (only in continuous limit)
I2—E1B:	$\tilde{\gamma} = [26(\sin 2k_x \sin k_y - \sin k_x \sin 2k_y)$ $- 4(\sin 3k_x \sin k_y - \sin k_x \sin 3k_y)$ $- 2(\sin 3k_x \sin 2k_y - \sin 2k_x \sin 3k_y)]/1440$ $\approx (k_x k_y^5 - k_x^5 k_y)/180$
I1—E2:	$\tilde{\gamma} \equiv 0$; $\tilde{\omega} = 0$ (only in continuous limit)
I2—E2:	$\tilde{\gamma} \equiv 0$; $\tilde{\omega} = 0$ (only in continuous limit)
I1—E3:	$\tilde{\gamma} = 2/3[k_x \sin k_y - k_y \sin k_x]$ $+ 1/6[(k_x - k_y)\sin(k_x + k_y) - (k_x + k_y)\sin(k_x - k_y)]$ $\approx (kx^5 ky - kx ky^5)/180$
I2—E3:	$\tilde{\gamma} = \{660[k_x \sin(k_y) - k_y \sin(k_x)]$ $+ 66[k_x \sin(2k_y) - k_y \sin(2k_x)]$ $+ 260[-(k_x + k_y)\sin(k_x - k_y) + (k_x - k_y)\sin(k_x + k_y)]$ $+ 26[-k_x \sin(k_x - 2k_y) - k_y \sin(2k_x - k_y)$ $- k_y \sin(2k_x + k_y) + k_x \sin(k_x + 2k_y)]$ $+ 10[-k_y \sin(k_x - 2k_y) - k_y \sin(2k_x - k_y)$ $+ k_x \sin(2k_x + k_y) - k_y \sin(k_x + 2k_y)]$ $+ 1[-(k_x + k_y)\sin(2k_x - 2k_y) + (k_x - k_y)\sin(2k_x$ $+ 2k_y)]\}/1440 \approx 0.000198(kx^3 ky - kx ky^3)$

Table II shows the numerical value of the growth rate as a function of mode numbers k_x, k_y, for method I1—E1A or I2—E1A with one particle per cell for particle location midway between the grid points.

Tables IIIa–d give the growth rate's functional dependence on wavenumber, $\gamma(k_x, k_y)$ as it varies with the interpolation and field methods. Table IV shows the numerical value of the growth rate for $k_x = \pi/8$, $k_y = \pi/4$ as it varies with the interpolation and field methods.

TABLE IIId

Growth Rate $\gamma(k_x, k_y)$ Dependence on Method Subtracted Dipole (I3)—Continuous Limit

	$\tilde{\gamma} = \frac{\gamma}{\Omega_{ci}}(k_x^2 + k_y^2 + \rho_x^{-2})$
I3A, I3B—E1A:	$\tilde{\gamma} \equiv 0$; $\tilde{\omega} \equiv 0$
I3C—E1A:	$\tilde{\gamma} = (\sin 2k_x \sin k_y - \sin k_x \sin 2k_y)/4$ $\approx (k_x k_y^3 - k_x^3 k_y)/4$
I3A, I3B—E1B:	$\tilde{\gamma} = (\sin 2k_x \sin k_y - \sin k_x \sin 2k_y)/12$ $\approx (k_x k_y^3 - k_x^3 k_y)/12$
I3A, I3B—E1Bm:	$\tilde{\gamma} \equiv 0$; $\tilde{\omega} \equiv 0$
I3A, I3B—E2:	$\tilde{\gamma} \equiv 0$; $\tilde{\omega} \equiv 0$
I3A, I3B—E3:	$\tilde{\gamma} = k_x \sin k_y - k_y \sin k_x$ $\approx (k_x^3 k_y - k_x k_y^3)/6$

TABLE IV

Growth Rate ($k_x = \pi/8$, $k_y = \pi/4$) versus Method

Method	Single particle per cell γ/Ω_{ci}	Continuous limit γ/Ω_{ci}
I1—E1A	.0165	.0110
I2—E1A	.0165	.0106
I1—E1B	.00549	0
I2—E1B	.00549	.000262
I1—E2	$\gamma = 0$, $\omega = .0654$	$\gamma = 0$, $\omega = 0$
I2—E2	$\gamma = 0$, $\omega = .00300$	$\gamma = 0$, $\omega = 0$
I1—E3	.00593	−.000326
I2—E3	.00593	-7.5610^{-6}
I3A, I3B—E1A	$\gamma = 0$, $\omega = 0$	$\gamma = 0$, $\omega = 0$
I3C—E1A	.0165	.0165
I3A, I3B—E1B	.00549	.00549
I3A, I3B—E1Bm	$\gamma = 0$, $\omega = 0$	$\gamma = 0$, $\omega = 0$
I3A, I3B—E2	$\gamma = 0$, $\omega = 0$	$\gamma = 0$, $\omega = 0$
I3A, I3B—E3	−.0128	−.0128

6. SUMMARY AND CONCLUSIONS

We have demonstrated the existence of a numerical instability caused by a false divergence of $\mathbf{V}_{E\times B}$. The agreement between analysis and observations from our gyrokinetic code is essentially perfect. This instability may be operating in some existing gyrokinetic codes and possibly in a variety of other particle simulation codes that employ guiding center drifts. Its presence depends in detail on the particle interpolation methods used and on the field algorithm. The instability can be suppressed by smoothing, but even with smoothing there is the danger that instability can be present at low growth rates and so it can be unnoticed and then grow over long times to damaging levels.

Much of the analysis was done with Mathematica. Its use here greatly speeded up the calculations. It is mainly used to automatically take derivatives of various interpolation formulas and to automatically sum the many terms that can arise. An independent analysis of the continuous limit agrees in detail with results obtained from the integral of the single particle per cell Mathematica results.

At the risk of stating the obvious, we emphasize that the instability, or the oscillation, is not physics. Either form of a finite response represents an outright error. The lack of any physical interpretation for the instability is emphasized by the fact that the energy is not conserved at all during the development of this instability. This is in contrast to undersampling noise, common to all particle-in-cell simulation methods, that *is* expected to conserve energy to some degree of approximation. Undersampling noise, while undesirable, still is usually thought of as representing model physical effects, such as absorption and emission of waves, albeit at enhanced levels. No such interpretation can be assigned to this numerical instability. This suggests that this phenomenon could be responsible for the intermittent "jumps" in the energy that we observe in many of our thermal ion runs with the gyrokinetic code.

We are presently gearing up to test the gyrokinetic code with the stable I2—E2 analytic E method for standard Lee–Tang cases. Initial results show clear differences, but trends, gross effects, etc., are still being analyzed. Major differences have been seen by Lodestro [16] in a specific long scaling run.

APPENDIX A: GENERAL ERROR STENCIL FOR CIC INTERPOLATION AND 2-POINT FINITE DIFFERENCE E

The following is the error stencil as evaluated by the Mathematica program. It has 64 terms spread over a local 5×5 grid points.

```
drhotot:
-(dy*(0.5*(1. - dx)*(1. - dy)*(-pote[-1, -2] + pote[-1, 0]) +
        0.5*(1. - dx)*dy*(-pote[-1, -1] + pote[-1, 1]) +
        0.5*dx*(1. - dy)*(-pote[0, -2] + pote[0, 0]) +
        0.5*dx*dy*(-pote[0, -1] + pote[0, 1]))) -
  (1. - dy)*(0.5*(1. - dx)*(1. - dy)*(-pote[-1, -1] + pote[-1, 1]) +
     0.5*(1. - dx)*dy*(-pote[-1, 0] + pote[-1, 2]) +
     0.5*dx*(1. - dy)*(-pote[0, -1] + pote[0, 1]) +
     0.5*dx*dy*(-pote[0, 0] + pote[0, 2])) +
  dx*(0.5*(1. - dx)*(1. - dy)*(-pote[-2, -1] + pote[0, -1]) +
     0.5*(1. - dx)*dy*(-pote[-2, 0] + pote[0, 0]) +
     0.5*dx*(1. - dy)*(-pote[-1, -1] + pote[1, -1]) +
     0.5*dx*dy*(-pote[-1, 0] + pote[1, 0])) -
  dx*(0.5*(1. - dx)*(1. - dy)*(-pote[-2, 0] + pote[0, 0]) +
     0.5*(1. - dx)*dy*(-pote[-2, 1] + pote[0, 1]) +
     0.5*dx*(1. - dy)*(-pote[-1, 0] + pote[1, 0]) +
     0.5*dx*dy*(-pote[-1, 1] + pote[1, 1])) +
  dy*(0.5*(1. - dx)*(1. - dy)*(-pote[0, -2] + pote[0, 0]) +
     0.5*(1. - dx)*dy*(-pote[0, -1] + pote[0, 1]) +
     0.5*dx*(1. - dy)*(-pote[1, -2] + pote[1, 0]) +
     0.5*dx*dy*(-pote[1, -1] + pote[1, 1])) +
  (1. - dy)*(0.5*(1. - dx)*(1. - dy)*(-pote[0, -1] + pote[0, 1]) +
     0.5*(1. - dx)*dy*(-pote[0, 0] + pote[0, 2]) +
     0.5*dx*(1. - dy)*(-pote[1, -1] + pote[1, 1]) +
     0.5*dx*dy*(-pote[1, 0] + pote[1, 2])) +
  (1. -dx)*(0.5*(1. - dx)*(1. - dy)*(-pote[-1, -1] + pote[1, -1]) +
     0.5*(1. - dx)*dy*(-pote[-1, 0] + pote[1, 0]) +
     0.5*dx*(1. - dy)*(-pote[0, -1] + pote[2, -1]) +
     0.5*dx*dy*(-pote[0, 0] + pote[2, 0])) -
  (1. - dx)*(0.5*(1. - dx)*(1. - dy)*(-pote[-1, 0] + pote[1, 0]) +
     0.5*(1. - dx)*dy*(-pote[-1, 1] + pote[1, 1]) +
     0.5*dx*(1. - dy)*(-pote[0, 0] + pote[2, 0]) +
     0.5*dx*dy*(-pote[0, 1] + pote[2, 1]))
```

Now expand this

```
0. - 0.5*dx*pote[-2, -1] + 0.5*dx^2*pote[-2, -1] + 0.5*dx*dy*pote[-2, -1] -
  0.5*dx^2*dy*pote[-2, -1] + 0.5*dx*pote[-2, 0] - 0.5*dx^2*pote[-2, 0] -
  1.*dx*dy*pote[-2, 0] + 1.*dx^2*dy*pote[-2, 0] + 0.5*dx*dy*pote[-2, 1] -
  0.5*dx^2*dy*pote[-2, 1] + 0.5*dy*pote[-1, -2] - 0.5*dx*dy*pote[-1, -2] -
  0.5*dy^2*pote[-1, -2] + 0.5*dx*dy^2*pote[-1, -2] + 0.5*dx*pote[-1, -1] -
  1.*dx^2*pote[-1, -1] - 0.5*dy*pote[-1, -1] + 1.*dx^2*dy*pote[-1, -1] +
  1.*dy^2*pote[-1, -1] - 1.*dx*dy^2*pote[-1, -1] + 0.5*pote[-1, 0] -
  1.*dx*pote[-1, 0] + 1.*dx^2*pote[-1, 0] - 1.*dy*pote[-1, 0] +
  2.*dx*dy*pote[-1, 0] - 2.*dx^2*dy*pote[-1, 0] - 0.5*pote[-1, 1] +
  0.5*dx*pote[-1, 1] + 1.5*dy*pote[-1, 1] - 2.*dx*dy*pote[-1, 1] +
```

```
1.*dx^2*dy*pote[-1, 1] - 1.*dy^2*pote[-1, 1] + 1.*dx*dy^2*pote[-1, 1] -
0.5*dy*pote[-1, 2] + 0.5*dx*dy*pote[-1, 2] + 0.5*dy^2*pote[-1, 2] -
0.5*dy*dy^2*pote[-1, 2] - 0.5*dy*pote[0, -2] + 1.*dx*dy*pote[0, -2] +
0.5*dy^2*pote[0, -2] - 1.*dx*dy^2*pote[0, -2] - 0.5*pote[0, -1] +
1.*dx*pote[0, -1] + 1.*dy*pote[0, -1] - 2.*dx*dy*pote[0, -1] -
1.*dy^2*pote[0, -1] + 2.*dx*dy^2*pote[0, -1] + 0.5*pote[0, 1] -
1.*dx*pote[0, 1] - 1.*dy*pote[0, 1] + 2.*dx*dy*pote[0, 1] +
1.*dy^2*pote[0, 1] - 2.*dx*dy^2*pote[0, 1] + 0.5*dy*pote[0, 2] -
1.*dx*dy*pote[0, 2] - 0.5*dy^2*pote[0, 2] + 1.*dx*dy^2*pote[0, 2] -
0.5*dx*dy*pote[1, -2] + 0.5*dx*dy^2*pote[1, -2] + 0.5*pote[1, -1] -
1.5*dx*pote[1, -1] + 1.*dx^2*pote[1, -1] - 0.5*dy*pote[1, -1] +
2.*dx*dy*pote[1, -1] - 1.*dx^2*dy*pote[1, -1] - 1.*dx*dy^2*pote[1, -1] -
0.5*pote[1, 0] + 1.*dx*pote[1, 0] - 1.*dx^2*pote[1, 0] + 1.*dy*pote[1, 0] -
2.*dx*dy*pote[1, 0] + 2.*dx^2*dy*pote[1, 0] + 0.5*dx*pote[1, 1] -
0.5*dy*pote[1, 1] - 1.*dx^2*dy*pote[1, 1] + 1.*dx*dy^2*pote[1, 1] +
0.5*dx*dy*pote[1, 2] - 0.5*dx*dy^2*pote[1, 2] + 0.5*dx*pote[2, -1] -
0.5*dx^2*pote[2, -1] - 0.5*dx*dy*pote[2, -1] + 0.5*dx^2*dy*pote[2, -1] -
0.5*dx*pote[2, 0] + 0.5*dx^2*pote[2, 0] + 1.*dx*dy*pote[2, 0] -
1.*dx^2*dy*pote[2, 0] - 0.5*dx*dy*pote[2, 1] + 0.5*dx^2*dy*pote[2, 1]
```

This expanded form has 88 terms.

The error stencil evaluated at 0.5, 0.5 is obtained by:

```
In[5] := Expand[drhotot] /. {dx->.5,dy->.5}
Out[5]= 0. - 0.0625 pote[-2, -1] + 0.0625 pote[-2, 1] + 0.0625 pote[-1, -2] -
>     0.0625 pote[-1, 2] - 0.0625 pote[1, -2] + 0.0625 pote[1, 2] +
>     0.0625 pote[2, -1] - 0.0625 pote[2, 1]
```

APPENDIX B: ALTERNATIVE ANALYSIS FOR THE CONTINUOUS LIMIT

The analysis here is an alternative to that of Section 4.2 and, like that analysis, it shows ways to stabilize the instability that are applicable in the limit of a large number of particles per cell. One way, as noted in Section 4.2, is to use the analytical derivative "energy-conserving" force calculation (in the terminology of [2]). Also, the special case of the result stated at the end of Section 4.2 for the case of bilinear (area) weighting of charge and "momentum-conserving" force is treated here. This leads directly to scheme E1B of Section 5 (which is the modification of the differencing of φ to get mesh **E**, that is the same as Boris proposed in 1970 to make the *direction* of the differenced field more accurate—closer to the direction of **k**). This is consistent with the origin of the instability, since in this case the direction error leads to error in the divergence of particle flow.

In Chapter 8 of [2], find Eqs. (8-9(7)), (8-9(8)), and (8-9(15)) relating particle force **F** and density n to grid potential φ anad charge density ρ:

$$\rho_1(\mathbf{k}) = q \sum_{\mathbf{p}} S(\mathbf{k}_\mathbf{p}) n_1(\mathbf{k}_\mathbf{p})$$

$$\mathbf{F}(k) = -iqS(-\mathbf{k})\boldsymbol{\kappa}(\mathbf{k})\varphi(\mathbf{k}),$$

so the particle flux is $-in_0 S(-\mathbf{k})\boldsymbol{\kappa}(\mathbf{k})\varphi(\mathbf{k}) \times \hat{\mathbf{z}}/B_0$ and

$$q\frac{\partial}{\partial t} n_1(\mathbf{k}) = -\mathbf{k} \cdot \frac{n_0 q}{B_0} S(-\mathbf{k}) S(-\mathbf{k})\boldsymbol{\kappa}(\mathbf{k})\varphi(\mathbf{k}) \times \hat{\mathbf{z}},$$

where $n_0 q/B_0 = \varepsilon_0\, \omega_{\mathrm{pi}}^2/\Omega_{\mathrm{ci}}$,

$$\frac{\partial}{\partial t}\rho_1(\mathbf{k}) = q \sum_{\mathbf{p}} S(\mathbf{k}_\mathbf{p}) \frac{\partial}{\partial t} n_1(\mathbf{k}_\mathbf{p})$$

$$= -\varepsilon_0 \frac{\omega_{\mathrm{pi}}^2}{\Omega_{\mathrm{ci}}} \varphi(\mathbf{k}) \sum_{\mathbf{p}} S^2(\mathbf{k}_\mathbf{p})(\mathbf{k}_\mathbf{p} \times \boldsymbol{\kappa}(\mathbf{k}_\mathbf{p})) \cdot \hat{\mathbf{z}}.$$

Now use the gyrokinetic field equation

$$\frac{\omega_{\mathrm{pi}}^2}{\Omega_{\mathrm{ci}}^2}\left(K^2(\mathbf{k}) + \frac{1}{\rho_s^2}\right)\varphi = \frac{\rho_1}{\varepsilon_0},$$

where K^2 is the mesh Fourier transform of the Poisson operator, to get the dispersion relation

$$\left(K^2(\mathbf{k}) + \frac{1}{\rho_s^2}\right)\frac{1}{\Omega_{ci}}\frac{\partial\varphi}{\partial t} = \varphi\hat{\mathbf{z}}\cdot\sum_{\mathbf{p}}\boldsymbol{\kappa}(\mathbf{k}_\mathbf{p})\times\mathbf{k}_\mathbf{p}S^2(\mathbf{k}_\mathbf{p}).$$

The equivalence of this result and those of Section 4.2 can be shown by inserting the Fourier representations of the weights in Eq. (24c) and carrying out the integration with respect to $\mathbf{x}$.

For the "momentum-conserving" schemes covered by Chapter 8 of [2], $\boldsymbol{\kappa}(\mathbf{k}_\mathbf{p}) = \mathbf{k}(\mathbf{k})$ can be removed from the sum, but this expression is more general. Depending on only the expression $\mathbf{F}(\mathbf{k}) = -iq\boldsymbol{\kappa}(\mathbf{k})S(-\mathbf{k})\varphi(\mathbf{k})$, it applies also to the "energy-conserving" schemes (see Section 10-6) and the "multipole" schemes (see Section 11-5).

For the "energy-conserving" schemes, meaning that the force is given by Eq. (10-2(6)),

$$\mathbf{F}(\mathbf{x}) = -q\nabla|\Delta x|\sum_{\mathbf{j}}\varphi_\mathbf{j}S(\mathbf{X}_\mathbf{j} - \mathbf{x})$$

(E2 of the main text), where $|\Delta X|$ denotes the cell area, we have $\boldsymbol{\kappa}(\mathbf{k}) = \mathbf{k}$, so there is no instability. (I1—E2 and I2—E2 of the main text are stable methods; these are also stable in the single particle-per-cell limit.) Time integration errors can spoil this nice result.

For area-weighting (CIC),

$$S(\mathbf{k}) = \frac{\sin^2(k_x\Delta x/2)}{(k_x\Delta x/2)^2}\frac{\sin^2(k_y\Delta y/2)}{(k_y\Delta y/2)^2}.$$

For "momentum conserving" schemes, the summation can be factored into x and y parts and evaluated as per page 173 of [2] to find

$$\kappa_y\sum_{p_x,p_y}k_{x,p_x}S^2 = \kappa_y\frac{\sin k_x\Delta x}{\Delta x}\left(1 - \frac{2}{3}\sin^2\frac{1}{2}k_y\Delta y\right)$$
$$= \kappa_y\frac{\sin k_x\Delta x}{\Delta x}\left(\frac{2}{3} + \frac{1}{3}\cos k_y\Delta y\right)$$

and similarly for $\kappa_x\sum k_{y,p_y}S^2$. With this result the dispersion relation easily shows that the rate is zero if we choose

$$\kappa_x = \frac{\sin k_x\Delta x}{\Delta x}\left(\frac{2}{3} + \frac{1}{3}\cos k_y\Delta y\right);$$

i.e., E_x is the common centered difference in x, plus a smoothing in y with weights $(\frac{1}{6}, \frac{2}{3}, \frac{1}{6})$. (CIC with this E is method I1—E1B of the main text; it is stable in the continuous limit but unstable in the single particle-per-cell limit.) Compare to Eqs. (14-9(8)) and (14-9(11)) of [2] to see that this is the same as Boris' 6-point scheme. While the present result is correct for all $\mathbf{k}$, Boris sought only to correct the direction to a relative error of order $(k_x\Delta x)^4$ (Eq. (14-9(12))).

Using the result in problem 11-5e for $S(\mathbf{k})$ for the subtracted-dipole (SUDS) scheme one could similarly derive the SUDS dispersion relation.

ACKNOWLEDGMENT

This work was performed by LLNL under DoE Contract No. W-7405-ENG-48.

REFERENCES

1. J. M. Dawson, *Rev. Mod. Phys.* **55,** 403 (1983).
2. C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation* (McGraw–Hill, New York, 1985).
3. R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (McGraw–Hill, New York, 1981).
4. W. W. Lee, *Phys. Fluids* **26,** 556 (1983).
5. D. H. E. Dubin, J. A. Krommes, C. Oberman, and W. W. Lee, *Phys. Fluids* **26,** 3524 (1983).
6. W. W. Lee, J. A. Krommes, C. Oberman, and R. A. Smith, *Phys. Fluids* **27,** 2652 (1984).
7. J. F. Federici, W. W. Lee, and W. M. Tang, *Phys. Fluids* **30,** 425 (1986).
8. W. W. Lee, *J. Comput. Phys.* **72,** 243 (1987).
9. W. W. Lee and W. M. Tang, *Phys. Fluids* **31,** 612 (1988).
10. R. D. Sydora, T. S. Hahm, W. W. Lee, and J. M. Dawson, *Phys. Rev. Lett.* **64,** 2015 (1990).
11. W. W. Lee, T. S. Hahm, and J. V. W. Reynders, *Considerations of Electron Dynamics in a Gyrokinetic Plasma, Sherwood Theory Conference, 1989,* Paper-3D2 (unpublished).
12. A. M. Dimits, J. F. Drake, and W. W. Lee, *Kinetic Effects on Ion Temperature Gradient Driven Turbulence, Sherwood Theory Conference, 1990,* Paper-3D27 (unpublished).
13. A. M. Dimits and W. W. Lee, *Phys. Fluids B* **3,** 1558 (1991).
14. R. D. Sydora, private communication.
15. R. D. Sydora, *Phys. Fluids B* **2,** 1455 (1990).
16. L. L. Lodestro, private communication.