1-D Tests of the NIMROD 2-fluid Algorithm

LANL 6/18/96 (CRS)

I. Introduction

In the early stages of the NIMROD project, we decided to solve the set of two-fluid equations using a temporal algorithm similar to those in the finite-difference codes PIC3D and TPCN. We then decided to use finite elements to provide flexibility for modeling realistic, complex geometries and to simplify the block decomposition of the computational domain. Though the time-advance and the spatial representation are each well-tested, they have not previously been implemented together. Thus, we are pioneering two-fluid, finite element computations.

While the objective is to understand experimental results with full 3-D computations, it is prudent to exercise and validate the new aspects of the algorithm with a 1-D code, which is quick to run and relatively simple to modify. For this, the DMOM code is ideal. Dan Barnes wrote DMOM to test the Quiet Implicit Particle scheme, and the fluid parts of the code are very similar to PIC3D and TPCN. The primary differences—besides fewer independent spatial directions—are that direct solves are used for inverting matrices and that scalar variables (electrostatic potential, ϕ, number densities, n_e and n_i, and pressures, P_e and P_i) are spatially staggered from the vector components on the grid. The code is typically run with a periodic domain to permit free propagation of an assortment of waves, which are initialized from analytic dispersion relations. The code is very accurate in these cases. Even with coarse resolution (10 grid cells per wave length) and large time steps ($v_{phase}\Delta t/\Delta x \sim 1/4$), the decay rate is several orders of magnitude smaller than the real frequency. The algorithm can also model nonlinear phenomena, though this has not been done recently.

With Dan's guidance, I have converted DMOM to use a finite element representation with linear splines for all quantities. I have run into some difficulties, found a few solutions, and have learned a lot about finite elements in the process (I think). I feel that it is worth recording all of this for a couple of reasons (besides Dalton's prompt). First, it is much easier to learn about the algorithm without all the complications of multiple dimensions, covariant vector components, nonorthogonal coordinates, etc. Thus, others may find this information useful for stepping into NIMROD. Second, though there are bound to be many problems in 3-D that are not encountered here, those encountered in 1-D will be present in 3-D unless we can solve them at this point. The following will therefore help explain why parts of the algorithm are done in a

particular way. Finally, I think that much of the NIMROD numerical analysis can be based on 1-D analysis, so this can serve as a beginning for that aspect of the project.

This note is organized in the following manner: Section II will cover some of the basics of 1-D finite elements including integrals that lead to matrix elements and the effect of the mass matrix on truncation error. Section III is a discussion of the electrostatic aspects of the algorithm and provides support for the change in the electric field representation as discussed in Dan's equations document dated 5/22/96. Section IV deals with the electromagnetic difficulty I have encountered recently. We now understand the problem and have an idea how to fix it, but the new approach has not yet been tested. I shall briefly describe it in Section V and conclude at that point. As we learn more I will keep updating this note.

II. 1-D Finite Element Equations and Basic Properties

As mentioned above, the equations solved in DMOM are nearly identical to those in NIMROD. The exceptions are the lack of resistivity (though it could be implemented), displacement current and $n_e \neq n_i$ is an option, and at this point I have not yet converted the terms representing the advection of current density. To keep things focused, I shall also ignore the displacement current for now. Thus, the equations are,

$$\frac{\partial \mathbf{J}_e}{\partial t} = \frac{e^2 n}{m_e}\mathbf{E} - \frac{e}{m_e c}\mathbf{J}_e \times \mathbf{B} + \hat{\mathbf{x}}\frac{e}{m_e}\frac{\partial P_e}{\partial x} \tag{1}$$

$$\frac{\partial \mathbf{J}_i}{\partial t} = \frac{e^2 n}{m_i}\mathbf{E} + \frac{e}{m_i c}\mathbf{J}_i \times \mathbf{B} - \hat{\mathbf{x}}\frac{e}{m_i}\frac{\partial P_i}{\partial x} \tag{2}$$

$$4\pi \mathbf{J} = c\hat{\mathbf{x}} \times \frac{\partial \mathbf{B}}{\partial x} \tag{3}$$

$$\frac{\partial J_x}{\partial x} = 0 \tag{4}$$

$$\mathbf{B} = \hat{\mathbf{x}} \times \frac{\partial \mathbf{A}}{\partial x} \tag{5}$$

$$\frac{1}{c}\frac{\partial \mathbf{A}}{\partial t} = -\left(\hat{\mathbf{y}}E_y + \hat{\mathbf{z}}E_z\right) \tag{6}$$

$$\frac{\partial \phi}{\partial x} = -E_x \tag{7}$$

$$\frac{\partial n}{\partial t} = -\frac{1}{e}\frac{\partial J_{i_x}}{\partial x} \tag{8}$$

$$\frac{\partial P_e}{\partial t} = \frac{J_{e_x}}{ne}\frac{\partial P_e}{\partial x} + \gamma P_e \frac{\partial}{\partial x}\left(\frac{J_{e_x}}{ne}\right) \tag{9}$$

$$\frac{\partial P_i}{\partial t} = -\frac{J_{i_x}}{ne}\frac{\partial P_i}{\partial x} - \gamma P_i \frac{\partial}{\partial x}\left(\frac{J_{i_x}}{ne}\right). \tag{10}$$

The closure moments from the particles have not yet been converted and have also been omitted from the above set. Equation (4) is implied by (3), but it is listed separately because it is used in the electric field solve in some of the formulations.

The finite element form uses linear splines (tent functions) as the basis functions, and all quantities have nodal values on the grid points. Thus, there is no staggering to be consistent with NIMROD. A differential equation may be converted to a matrix equation by substituting the spatial approximation for the continuous x-dependence, e.g., $\phi(x) \rightarrow \sum_j \phi_j w_j(x)$, where $w_j(x)$ is the tent function centered at node j, multiplying by $w_i(x)$, and integrating over x to produce each ij element of the matrix. This is the Galerkin method, of course, and produces the same representation as a variational method for symmetric terms. It is more general, however, because nonsymmetric matrices are allowed, and solving them is not a problem in 1-D. In NIMROD we will put the asymmetric part of the susceptibility tensor on the right side of the electric field equation to facilitate indirect solves. Testing this in DMOM is certainly possible, but it is not essential.

The heart of the algorithm is the electric field equations which come from combining Eqs. (3), (5) and (6) with parts of (1) and (2) and from combining Eqs. (4) and (7), also with parts of (1) and (2). Without precisely defining the temporal algorithm at this point, the equations are

$$4\pi\Delta t\mathbf{J} + \mathbf{S}\cdot\left(\mathbf{E} - \hat{\mathbf{x}}\frac{\partial \phi}{\partial x}\right) = -c\Delta t\frac{\partial^2}{\partial x^2}\left(\mathbf{A} - c\Delta t\mathbf{E}\right) \qquad \text{and} \tag{11}$$

$$4\pi\Delta t\frac{\partial J_x}{\partial x}+\frac{\partial}{\partial x}\mathbf{S}\cdot\left(\mathbf{E}-\hat{\mathbf{x}}\frac{\partial\phi}{\partial x}\right)=0\ , \tag{12}$$

where **E** and ϕ are the unknowns, and **S** is the susceptibility tensor derived from (1) and (2) neglecting pressure. Equation (11) is used for the y and z directions—though in one version of the code, it is used for all three directions in which case (12) is not used (see Section IV). Applying the Galerkin method to (11) and moving the unknown terms to the left side and known terms to the right produces

$$\begin{aligned}\mathbf{S}_k\cdot\mathbf{E}_j\int dx w_i w_k w_j-\mathbf{S}_{\eta x_k}\phi_j\int dx w_i w_k w_j'+(c\Delta t)^2\mathbf{E}_j\int dx w_i' w_j' = c\Delta t\mathbf{A}_j\int dx w_i' w_j' \\ -4\pi\Delta t\mathbf{J}_j\int dx w_i w_j\end{aligned}\ , \tag{13}$$

where $\eta\in\{y,z\}$, {i,j,k} are mesh points, and primes on the weight functions indicate derivatives with respect to x. Integration by parts has been applied so that only first derivatives of weight functions appear. Using the same procedure, Eq. (12) becomes

$$-\mathbf{S}_{xx_k}\phi_j\int dx w_i' w_k w_j'-\mathbf{S}_{x\eta_k}\cdot\mathbf{E}_j\int dx w_i' w_k w_j=4\pi\Delta t J_{x_j}\int dx w_i w_j'\ . \tag{14}$$

In NIMROD, the weight function integrals will be complicated, so Alan Glasser recommends that we use Gaussian quadrature. Here the integrals are trivial and are done analytically. Let's examine a few of the resulting matrix elements in detail. The last term in Eq. (13) is

$$-4\pi\Delta t\mathbf{J}_j\int dx w_i w_j\rightarrow -4\pi\Delta t\begin{pmatrix}\bullet & \frac{\Delta x_{i-1}}{6} & \\ \frac{\Delta x_{i-1}}{6} & \frac{\Delta x_{i-1}}{3}+\frac{\Delta x_i}{3} & \frac{\Delta x_i}{6}\\ & \frac{\Delta x_i}{6} & \bullet\end{pmatrix}\times\begin{pmatrix}\bullet\\ \mathbf{J}_{i-1}\\ \mathbf{J}_i\\ \mathbf{J}_{i+1}\\ \bullet\end{pmatrix}, \tag{15}$$

where Δx_i is the mesh spacing between nodes i and i+1, and the bullets represent a continuation of the pattern. This term is linear in **J**, so the matrix is the 1-D mass matrix. The last term on the left side of (13) is $-(c\Delta t)^2$ times the Laplacian operator acting on the electric field:

$$(c\Delta t)^2 \mathbf{E}_j \int dx w_i' w_j' \rightarrow -(c\Delta t)^2 \begin{pmatrix} \bullet & \frac{1}{\Delta x_{i-1}} & \\ \frac{1}{\Delta x_{i-1}} & -\frac{1}{\Delta x_{i-1}} - \frac{1}{\Delta x_i} & \frac{1}{\Delta x_i} \\ & \frac{1}{\Delta x_i} & \bullet \end{pmatrix} \times \begin{pmatrix} \bullet \\ \mathbf{E}_{i-1} \\ \mathbf{E}_i \\ \mathbf{E}_{i+1} \\ \bullet \end{pmatrix}. \tag{16}$$

If the mesh spacing were uniform, the matrix would be equivalent to a compact finite-difference Laplacian operator multiplied by Δx. The term on the right side of (14) represents a divergence (or a gradient):

$$4\pi\Delta t J_{x_j} \int dx w_i w_j' \rightarrow 4\pi\Delta t \begin{pmatrix} \bullet & \frac{1}{2} & \\ -\frac{1}{2} & 0 & \frac{1}{2} \\ & -\frac{1}{2} & \bullet \end{pmatrix} \times \begin{pmatrix} \bullet \\ J_{x_{i-1}} \\ J_{x_i} \\ J_{x_{i+1}} \\ \bullet \end{pmatrix},$$

where the form is again comparable to a centered finite difference, except that it extends over two cells (3 point stencil). Nonlinear terms are a little more complicated. The first term on the left side of (14) is important in the analysis in the next two sections. It is

$$-\mathbf{S}_{xx_k} \phi_j \int dx w_i' w_k w_j' \rightarrow$$

$$\rightarrow \frac{1}{2} \begin{pmatrix} \bullet & \frac{\mathbf{S}_{xx_{i-1}} + \mathbf{S}_{xx_i}}{\Delta x_{i-1}} & \\ \frac{\mathbf{S}_{xx_{i-1}} + \mathbf{S}_{xx_i}}{\Delta x_{i-1}} & -\frac{\mathbf{S}_{xx_{i-1}}}{\Delta x_{i-1}} - \left(\frac{1}{\Delta x_{i-1}} + \frac{1}{\Delta x_i}\right)\mathbf{S}_{xx_i} - \frac{\mathbf{S}_{xx_{i+1}}}{\Delta x_i} & \frac{\mathbf{S}_{xx_i} + \mathbf{S}_{xx_{i+1}}}{\Delta x_i} \\ & \frac{\mathbf{S}_{xx_i} + \mathbf{S}_{xx_{i+1}}}{\Delta x_i} & \bullet \end{pmatrix} \times \begin{pmatrix} \bullet \\ \phi_{i-1} \\ \phi_i \\ \phi_{i+1} \\ \bullet \end{pmatrix}.$$

Note that if **S** is uniform in x, the matrix becomes the Laplacian multiplied by **S**. The other nonlinear terms behave similarly, and this will be used to simplify the linear analysis below.

The equations for the fluid moments can be arranged to use the same matrix element integrals. To avoid terms which are the product of more than two variables,

we take a trick from PIC3D in the current density equations. The time-advance of $\mathbf{J}_i$ and $\mathbf{J}_e$ have the form [see Eq. (1.7) of the equations document dated 5/22/96],

$$\left(\mathbf{I}+\frac{f_\Omega q_s \Delta t}{m_s c}\mathbf{B}\times\mathbf{I}\right)\cdot\mathbf{J}_s^{n+1}=\left(\mathbf{I}+\frac{(f_\Omega-1) q_s \Delta t}{m_s c}\mathbf{B}\times\mathbf{I}\right)\cdot\mathbf{J}_s^{n}+\Delta t\mathbf{F}_s\ ,$$

which is equivalent to

$$\mathbf{J}_s^{n+1}=\mathbf{R}_s\cdot\left(\frac{1}{f_\Omega}\mathbf{J}_s^{n}+\Delta t\mathbf{F}_s\right)-\left(1-\frac{1}{f_\Omega}\right)\mathbf{J}_s^{n}\ , \tag{17}$$

where $\mathbf{R}_s$ is the spatially local operator, $\frac{\mathbf{I}+\mathbf{r}_s\mathbf{r}_s-\mathbf{r}_s\times\mathbf{I}}{1+r_s^2}$, $\mathbf{r}_s=\frac{f_\Omega q_s \Delta t}{m_s c}\mathbf{B}$, $\mathbf{F}_s$ represents forces other than $\mathbf{J}\times\mathbf{B}$, and superscripts indicate the time-step index. The only terms that may be the product of more than two variable are those representing the flux of current density. For now we shall assume that the tensor can be constructed at the nodes and splined as a single quantity in the same fashion as the susceptibility tensor.

The one finite element matrix that does not have a finite difference analog is the mass matrix. While it may be essential to use a distributed mass matrix in some equations, it may be possible to use a lumped mass matrix to simplify others. This is another issue that will be tested with DMOM. For now, let's consider the truncation error in a couple of equations to develop some intuition about this subject. From this point, I will assume a uniform spatial grid [DMOM is capable of using a nonuniform grid, but this has not been test recently—another item to examine]. Note that each row of the distributed mass matrix from (15) is then the same as $1+(\Delta x)^2 L/6$, where L is the corresponding row of the Laplacian operator from (16). For the simple relation, $E_x=-\partial\phi/\partial x$, an application of the Taylor series leads to the truncation error,

$$E_{x_i}+\frac{(\Delta x)^2}{6}\left(E''_{x_i}+\frac{(\Delta x)^2}{12}E^{IV}_{x_i}+\ldots\right)=-\left(\phi'_i+\frac{(\Delta x)^2}{6}\phi'''_i+\frac{(\Delta x)^4}{120}\phi^V_i+\ldots\right),$$

so the relation is fourth-order accurate in Δx. If a lumped mass matrix were used, the left side would be just E_{x_i}, and the relation would be second-order accurate like a

centered finite difference. For a component of Ampere's law, $4\pi J_y = -c\,\partial^2 A_y / \partial x^2$, we have

$$4\pi J_{y_i} + \frac{4\pi(\Delta x)^2}{6}\left(J''_{y_i} + \frac{(\Delta x)^2}{12} J^{IV}_{y_i} + \ldots \right) = -c\left(A''_{y_i i} + \frac{(\Delta x)^2}{12} A^{IV}_{y_i i} + \ldots \right).$$

Here the distributed mass matrix changes the sign of the truncation error, but the relation is second-order accurate. Our entire system of equations will therefore only be second-order accurate in space, so lumping mass matrices where possible may save computer time without a significant loss of accuracy.

III. Analysis of Electrostatic Waves

To advance the current densities, the original algorithm uses an existing electric field in Eqs. (1) and (2) and updates the vector potential before the electric field solve. Thus, the electric field and scalar potential appearing in (13) and (14) are the difference between the new values and the last best guess; they are therefore labeled with 'δ.' In successive predictor/corrector loops the solution vector for $\delta\mathbf{E}$ and $\delta\phi$ should approach zero if nonlinear terms converge. When this algorithm is used in DMOM, electrostatic ion acoustic waves are numerically unstable unless Δt is so small that an *electron* plasma wave moves less than a grid cell per time step. This is unacceptable for simulating low-frequency phenomena and is not the case in the finite difference code. A little analysis led to the formulation where (1) and (2) are advanced without electric field and (13) and (14) determine $\mathbf{E}$ and ϕ (not just the changes). The new algorithm produces results that are comparable to the finite difference code. In this section, I will provide a basic numerical stability analysis that explains the observed behavior and supports the new algorithm.

To determine numerical stability, I use a linear form of the equations. The waves are purely electrostatic ($\mathbf{B}_0$ is in the x-direction) in this section, and I set P_i=0. I will also reduce the problem by considering a leapfrog algorithm which is close to the actual predictor/corrector. The two algorithms have similar behavior in that they both have similar CFL conditions when the semi-implicit coefficient on pressure is set to 0. Both are stable at larger time steps when the semi implicit coefficient is nonzero. Assuming an e^{ikx} dependence for all perturbed quantities—the usual approach in finite difference analysis—simplifies the spatial operators. The Laplacian reduces to $2(cos(k\Delta x)-1)/(\Delta x)^2$ and gradients become $i\,sin(k\Delta x)/(\Delta x)$. The relation between the

distributed mass matrix and the Laplacian implies that the former reduces to 1+(*cos*(kΔx)-1)/3.

The leapfrog version of the original algorithm advances quantities in the following manner (with fully advanced centering for the electrostatic potential, f_e=1):

$$\rho J_e^* = \rho J_e^n - i\alpha\sigma\phi^n + i\alpha\sigma P_e^n \tag{18a}$$

$$\rho J_i^* = \rho J_i^n - i\alpha\sigma\nu\phi^n \tag{18b}$$

$$2(1+\nu)\alpha\chi\ \delta\phi = i\sigma\left(J_e^* + J_i^*\right) \tag{18c}$$

$$\rho J_e^{n+1} = \rho J_e^* - i\alpha\sigma\ \delta\phi \tag{18d}$$

$$\rho J_i^{n+1} = \rho J_i^* - i\alpha\sigma\nu\ \delta\phi \tag{18e}$$

$$(\rho - 2\beta\chi)P_e^{n+1} = (\rho - 2\beta\chi)P_e^n + i\alpha\sigma J_e^{n+1}, \tag{18f}$$

where only the x-component of current density is used, $\chi \equiv cos(k\Delta x) - 1$, $\sigma \equiv sin(k\Delta x)$, and ρ is the mass matrix. The pressure and current densities have been normalized for convenience, P->P/γP_0 and J->J/enc$_e$, where P_0 is the background electron pressure and $c_e = \sqrt{\gamma P_0 / n m_e}$. In addition, $\alpha \equiv c_e \Delta t/\Delta x$, $\nu \equiv m_e/m_i$, and β is the coefficient for the semi-implicit operator, $(\Delta x)^2 \partial^2/\partial x^2$, which is applied to $\partial P/\partial t$. Substituting $\delta\phi = \phi^{n+1} - \phi^n$, and rewriting Eqs. (18a-f) in matrix form produces

$$\begin{pmatrix} \rho & 0 & i\alpha\sigma & 0 \\ 0 & \rho & i\alpha\sigma\nu & 0 \\ 0 & 0 & 2(1+\nu)\alpha\chi & 0 \\ -i\alpha\sigma & 0 & 0 & \rho - 2\beta\chi \end{pmatrix} \times \begin{pmatrix} J_e \\ J_i \\ \phi \\ P_e \end{pmatrix}^{n+1}$$

$$= \begin{pmatrix} \rho & 0 & 0 & i\alpha\sigma \\ 0 & \rho & 0 & 0 \\ i\sigma & i\sigma & (1+\nu)\alpha\left(\dfrac{\sigma^2}{\rho} + 2\chi\right) & -\dfrac{\alpha\sigma^2}{\rho} \\ 0 & 0 & 0 & \rho - 2\beta\chi \end{pmatrix} \times \begin{pmatrix} J_e \\ J_i \\ \phi \\ P_e \end{pmatrix}^n \tag{19}$$

Setting the vector of new fields to λ times the old, moving everything to one side, and setting the determinant of the resulting matrix to zero gives the following equation for the eigenvalues,

$$0 = (1+\nu)\alpha\sigma^2\left\{\left[\rho(1-\lambda)^2(\rho-2\beta\chi)+\alpha^2\sigma^2\lambda\right]\left[(1-2\lambda)+\frac{2\chi\rho}{\sigma^2}(1-\lambda)^2\right]+\frac{\alpha^2\sigma^2\lambda^3}{1+\nu}\right\}. \quad (20)$$

To interpret Eq. (20), consider first the related equation for the original finite difference form of the algorithm. In that case, there is no mass matrix, $\rho \to 1$, and a compact operator is used for gradients and derivatives, $\sigma \to 2\,sin(k\Delta x/2)$. Thus, the coefficient $2\chi\rho/\sigma^2$ is -1 for all possible $k\Delta x$, and (20) simplifies to

$$0 = \lambda^2\left[(1-\lambda)^2(1-2\beta\chi)+\frac{\alpha^2\sigma^2\lambda\nu}{1+\nu}\right]. \quad (21)$$

The roots which represent numerical waves are those found by setting the term in brackets to zero. They are

$$\lambda = \frac{2-\xi\pm\sqrt{\xi^2-4\xi}}{2}, \text{ where } \xi = \frac{\alpha^2\sigma^2\nu}{(1-2\beta\chi)(1+\nu)}. \quad (22)$$

The magnitude of these roots is unity for $0\le\xi\le4$, so the algorithm is numerically stable when this condition is met. With β=0, there is a CFL condition, but it is based on the ion acoustic wave, not the electron plasma wave, $\alpha\sqrt{\nu/(1+\nu)} \le 1$ at $k\Delta x=\pi$. As familiar to semi-implicit aficionados, the condition can be relaxed by increasing β. The two roots at 0 are somewhat artificial, as they do not imply that waves will not propagate. Without displacement current, there is no dependence on ϕ, and the equations for J_i and J_e are linearly dependent. Thus, there are only two linearly independent equations, hence two nonzero roots.

For the finite element algorithm, Eq. (20) simplifies in the two limits of $k\Delta x$. The coefficient $2\chi\rho/\sigma^2$=-2(2+*cos*($k\Delta x$))/3(1+*cos*($k\Delta x$)) goes to -1 as $k\Delta x \to 0$, so in the well-resolved limit, the eigenvalues are the same as those from the finite difference algorithm. However, as $k\Delta x \to \pi$, $2\chi\rho/\sigma^2 \to -\infty$, and the terms which include this factor dominate. In this limit, the dispersion relation is

$$0 = (1-\lambda)^2\left[\rho(1-\lambda)^2(\rho - 2\beta\chi) + \alpha^2\sigma^2\lambda\right].$$

The form of this relation is similar to (21), but the second term in the brackets does not have a factor of ν. Therefore, the CFL condition is based on the phase speed of the electron plasma wave not the ion acoustic wave. This is consistent with the results of DMOM. The relation suggests that using β of order ν^{-1}, will allow stable calculations with α of order $\nu^{-1/2}$, but the accuracy would be unnecessarily poor.

The difficulties in the finite element representation stem from an inconsistency between the Laplacian operator and successive applications of two first-order derivatives. The former appears on the left side of Eq. (14), while the latter appears on the right side—J on the right side contains the term $-\mathbf{S}_{xx}\,\partial\phi/\partial x$ from (1) and (2) or (18a), (18b), (18c) and (18d). In PIC3D and TPCN, scalars are on the same grid as the vectors, like NIMROD, but second-order differences are based on successive applications of first-order differences. Thus, the operators are not compact. This does not fit into the usual finite-element framework, and it has problems of its own.

The solution for finite elements is to not separate the electric field acceleration but to put it entirely on the left side of the electric field solve. Thus, J* does not contain terms with **E** and $-\partial\phi/\partial x$. [Actually, J* is no longer created as a nodal quantity. The contributions to the right sides of (13) and (14) are constructed directly from old current densities and the remaining forces.] The vector **J** is then found from Ampere's law with **A** updated from the new electric field, consistent with (13) and (14). In 1-D without displacement current, we automatically have J_x=0. The total momentum density, **M**, is independent of **E** when quasineutrality holds and is advanced with the new **J**. This is like an MHD velocity advance, but we keep all of the terms from (1) and (2), so no aspect of the two-fluid formulation is lost. From the new **M** and **J**, we then determine the new current densities for each species.

For the electrostatic problem, the new algorithm consists of the following steps:

$$2(1+\nu)\alpha\chi\phi = i\sigma\left(J_e^n + J_i^n\right) + 2\alpha\chi P_e^n \tag{23a}$$

$$J_e^{n+1} = -J_i^{n+1} \tag{23b}$$

$$\rho M^{n+1} = \rho\left(J_i^n + \nu J_e^n\right) - i\alpha\sigma\nu P_e^n \tag{23c}$$

$$J_e^{n+1} = \frac{1}{\nu}\left(J_i^{n+1} - M^{n+1}\right) \tag{23d}$$

$$(\rho - 2\beta\chi)P_e^{n+1} = (\rho - 2\beta\chi)P_e^n + i\alpha\sigma J_e^{n+1}, \tag{23e}$$

where the x-component is implied for M as well as J, and M->M/$m_i nc_e$. Note that ϕ only appears in (23a) and has no effect on the wave. Also, (23b) and (23d) may be used in (23c), so that there are only two equations, one for electron pressure and one for electron current. The matrix equation is then

$$\begin{pmatrix} (1+\nu)\rho & 0 \\ -i\alpha\sigma & \rho - 2\beta\chi \end{pmatrix} \times \begin{pmatrix} J_e \\ P_e \end{pmatrix}^{n+1} = \begin{pmatrix} (1+\nu)\rho & i\alpha\sigma\nu \\ 0 & \rho - 2\beta\chi \end{pmatrix} \times \begin{pmatrix} J_e \\ P_e \end{pmatrix}^{n},$$

and the dispersion relation is similar to (22), except that the mass matrix appears in the denominator of ξ, and σ represents the 3-point stencil instead of the compact operator. Thus, the CFL condition is based on the ion acoustic wave.

IV. Electromagnetic Inaccuracies

The first electromagnetic tests of the finite element algorithm were linear Alfven waves perpendicular to the magnetic field, and the results were similar to finite difference results. However, when I put $\mathbf{B}_0$ at an angle in the x-z plane, the x- and z-components of the electric field disappeared with the first electric field solve. I also had problems with the displacement current and made another version of DMOM that uses E_x instead of ϕ for comparison. [Eq. (13) is used for all three components, and (14) is not used.] This finite element version propagates waves as well as the finite-difference code. Narrowing it down, I found that the problem in the ϕ-version was more noticeable in the slow branch (torsional Alfven wave) than in the fast branch, where the electric field is mostly in the y-direction. I also found that this effect set in very rapidly as the angle between $\mathbf{B}_0$ and the x-direction is made greater than zero (B_{0z}~$0.001B_{0x}$). Furthermore, the reduction could be mitigated at these small angles by reducing either Δt or Δx . At first I was concerned with the condition number of the electric field matrix. It is ill-conditioned, but it is no worse than the finite-difference version. Analyzing the matrix in a similar fashion to the previous section, I finally found an electromagnetic analogue to the electrostatic inconsistency problem. I will describe the problem in this section.

The susceptibility tensor is simplest in the low frequency limit. Here, $|r_e|$ and $|r_i|$>>1, so the symmetric terms dominate. Thus,

$$\mathbf{S} = \sum_s (\omega_s \Delta t)^2 \frac{\mathbf{I} + \mathbf{r}_s \mathbf{r}_s - \mathbf{r}_s \times \mathbf{I}}{1 + r_s^2} \rightarrow \hat{\mathbf{b}}\hat{\mathbf{b}} \sum_s (\omega_s \Delta t)^2 \ ,$$

where $\hat{\mathbf{b}}$ is a unit vector in the direction of $\mathbf{B}_0$. Using the same notation for the trigonometric functions as in the previous section, plus $\tau \equiv \omega_p \Delta t$ and $\zeta \equiv c\Delta t/\Delta x$, the dominant terms of the electric field matrix are,

$$\begin{pmatrix} 2\tau^2 \chi \, cos^2(\theta) & -i\tau^2 \sigma \, cos(\theta) sin(\theta) \\ -i\tau^2 \sigma \, cos(\theta) sin(\theta) & \rho\tau^2 \, sin^2(\theta) - 2\chi\zeta^2 \end{pmatrix},$$

where the first row represents the left side of (14) and the second row is the z-component of (13). The angle θ is the angle between the x-axis and $\mathbf{B}_0$. When solving for ϕ and E_z, we effectively divide by the determinant of this matrix, which is

$$2\tau^4 \, cos^2(\theta) sin^2(\theta)\left(2\chi\rho + \sigma^2\right) - 4\tau^2\zeta^2\chi^2 \ .$$

The factor in parentheses in the first term is our old friend (enemy) that represents the inconsistency between the finite element second derivative and successive first derivatives. It is zero in the finite difference code, in which case only the second term survives. With the error of the finite element version, the first term is of order $(\tau/\zeta)^2$ times larger than the second for $\theta \sim \pi/4$. For typical simulations, $\Delta x \sim v_A \Delta t$, so $\tau/\zeta \sim v_A \omega_p \Delta t/c \sim eB_0 \Delta t / c\sqrt{m_e m_i}$, which is quite large. This explains why ϕ and E_z disappear during the first time step. The problem is reduced when B_0 is nearly parallel or perpendicular to **k**, consistent with DMOM results.

The finite element version that uses E_x does not have this problem. The important terms from the x- and z-components on the left side of (13) yield the following matrix,

$$\begin{pmatrix} \rho\tau^2 \, cos^2(\theta) & \rho\tau^2 \, cos(\theta) sin(\theta) \\ \rho\tau^2 \, cos(\theta) sin(\theta) & \rho\tau^2 \, sin^2(\theta) - 2\chi\zeta^2 \end{pmatrix}.$$

Here, the offensive terms in the determinant cancel as they do in the finite difference algorithm. Note that both successful versions have ill-conditioned matrices, because we always subtract large terms to leave a small term.

V. Future Directions and Concluding Comments

While the E_X-version of the finite element code runs as well as the finite difference code, it does not address how the divergence of current density will be controlled in NIMROD, where $\nabla \cdot \nabla \times$ is not automatically zero. Dan has recently proposed a new formulation, where we essentially divide the current density into longitudinal and transverse pieces instead of dividing the electric field, $\mathbf{S} \cdot \mathbf{E} \rightarrow \nabla \times \zeta - \nabla \psi$. The scalar ψ is determined by the divergence of Ampere's law, and the vector ζ is determined indirectly from the curl of Ampere's. My next step is to figure out what this algorithm would look like in 1-D and try to implement it if there is a nontrivial difference with the existing E_X version. I will also pursue some of the other issues mentioned throughout this note: nonuniform grid, nonlinear problems, lumping mass matrices, semi-implicit anisotropic susceptibilities, and converting current advection and particle closures.

So far, I have observed no particularly good reason for using finite elements in 1-D; though, there are good reasons in 3-D. I have not given numbers supporting this claim, but the accuracy of the various waves is comparable to but not better than the finite difference version of DMOM. This supports the prediction from the truncation error considerations discussed in Section II. In addition, we have to deal with operator consistency problems that are avoided in a natural fashion with finite differences; Sections III and IV show that we must avoid the inconsistency between second-order derivatives and consecutive first order derivatives. Nonetheless, the complications from realistic geometries will undoubtedly put finite elements in a more favorable light, so we will keep modifying the algorithm until we arrive at something satisfactory.