DRAFT DRAFT: Notes on implicit method used in gyrokinetic continuum codes

G. W. Hammett, W. Dorland, et.al.

March 29, 2001

(First draft was Feb. 26, 2001.)

1 General stability of implicit methods

These short notes describe some of the properties of the implicit method used in the GS2 code. The details of the method are described in Kotschenreuther, Rewoldt, Tang, Comp. Phys. Comm. **88**, 128 (1995), hereafter denoted **KRT**.

The linearized Vlasov or gyrokinetic equation can be written in the generic form

$$\frac{\partial f}{\partial t} = -iLf$$

where L is some complicated integro-differential operator, and $f(z, v, t)$ is the distribution function vs space, velocity, and time. The operator L involves integral operator terms because the potential Φ depends on a velocity integral of f. In general we can just think of L as some huge linear matrix, but we can at first focus on just one eigenvector at at time so that we can just think of L as some (complex) number representing a frequency.

Using some of the same notation as the KRT paper, time advancement is done with the general method

$$\frac{f^{n+1} - f^n}{\Delta t} = -iL(\delta f^{n+1} + (1 - \delta) f^n)$$

where the superscript n is the time index, such that $t = n\Delta t$. Setting the time-centering parameter $\delta = 1/2$ gives a standard time-centered half-implicit/half-explicit method (like Crank-Nicholson), $\delta = 0$ gives a simple fully explicit method, and $\delta = 1$ gives a fully (non-time-centered) implicit method. $\delta = 1/2$ gives second order accuracy. Solving this equation for f^{n+1} gives

$$f^{n+1} = \frac{1 - i(1 - \delta)L\Delta t}{1 + i\delta L\Delta t} f^n$$

For real L we can multiply this by its complex conjugate to find

$$|f^{n+1}|^2 = \frac{1 + (1 - \delta)^2 (L\Delta t)^2}{1 + \delta^2 (L\Delta t)^2} |f^n|^2$$

This demonstrates the well known result that this method is numerically stable for a centered-implicit method with $\delta = 1/2$, but has some numerical damping if $\delta > 1/2$ and some numerical amplification if $\delta < 1/2$. This damping or amplification is small if the frequency of interest is well resolved, $L\Delta t \ll 1$. Sometimes the code is run slightly biased towards future time ($\delta \approx 0.6$ in the KRT paper) to provide some damping of unresolved high frequency waves that are being ignored anyway.

?? Might put the general formula here for how the finite-time-step numerical frequency is related to the exact frequency of a mode.

2 The problem of inverting large matrices

Note that because L depends on the fields Φ, which is found from a velocity integral of f, in general the L operator couples together the values of f at all grid points and all velocities. I.e., in general the inverse of $(1+iL\delta\Delta t)$ is a huge $(N_z N_v)\times(N_z N_v)$ matrix. [A typical case might involve $N_z \sim 40$ and $N_v \sim N_{\mathrm{Energy}} N_{\mathrm{pitch\ angles}} = 10 \times 32$ and thus about 10^8 matrix elements.] In ballooning coordinates, the various k_θ and k_r modes are independent, so N_z is only the number of grid points along the field line, but the number of $(v_\perp, v_\parallel)$ velocity grid points N_v might be a few hundred, so this appears to be a time-consuming problem. However, Kotschenreuther's implicit method finds a way to exactly factor this problem in such a way that the fields are solved implictly by inverting just an $N_z \times N_z$ matrix, and f is solved implicitly by an $\mathcal{O}(N_z)$ sweeping solution of a lower-diagonal matrix (since it is a simple convection equation) for each velocity grid point. See the KRT paper for how this neat inversion trick is done. I'm not sure, but similar tricks for the fields might have been done in the past in implicit-particle codes.

3 Discrete analogue of the plasma dispersion function

Consider the linearized Vlasov equation in a 1-D limit:

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial z} = -\frac{q}{m}\frac{\partial \Phi}{\partial z} v F_m \tag{1}$$

Here we will do a straightforward calculation of the discretized Vlasov/continuum dispersion relation, similar to Langdon79's (A.B. Langdon, JCP, 30, 202 (1979)) calculation for a particle simulation (i.e., we will derive the equivalent of Eq.(11b) of Langdon79). As in Langdon, we will calculate the response to an imposed electric field of the form $\Phi(z,t) = \Phi \exp(ikz - i\omega t)$, with $\mathrm{Im}\,\omega > 0$ to give a growing mode. This solution of this linear equation in f has a homogeneous part (dependent on the initial condition for f) and an inhomogenous part (proportional to Φ). Assuming the homogenous part is stable, or has a growth rate smaller than $\mathrm{Im}\,\omega$, then in the long time limit the homogenous solution is negligible compared to the inhomogeneous solution, and we can just focus on the latter. Since we are using an implicit solution, the homogeneous part of the solution is stable (and thus ignorable) as long as $\delta \geq 0.5$.

[A digression: in the particle simulation approach, the homogeneous part of the solution is automatically stable. I.e., the homogeneous solution to Eq.(3) of Langdon79 must satisfy:

$$\mathbf{x}_{n+1}^{(1)} - 2\mathbf{x}_n^{(1)} + \mathbf{x}_{n-1}^{(1)} = 0$$

Looking for eigenvalue solutions with $\mathbf{x}_{n+1}^{(1)} = \lambda \mathbf{x}_n^{(1)}$, we get the quadratic equation $\lambda - 2 + 1/\lambda = 0$, which has the two roots $\lambda = 1 \pm 0$, i.e., it is automatically stable.]

So we will focus on (inhomogenous or normal mode) solutions with $f \propto \Phi \propto \exp(ikz - i\omega t)$. Differencing the spatial derivative in Eq.(1) as was done in Eq.(13) of KRT,

$$\frac{\partial f}{\partial z} \to \delta \frac{f_{j+1}^{n+1} - f_j^{n+1}}{\Delta z} + (1-\delta) \frac{f_{j+1}^n - f_j^n}{\Delta z}$$

(where j is used for the spatial index instead of the i that was used in KRT, to avoid confusion with my $i = \sqrt{-1}$, n is the time index and δ is the implicitness parameter). Using the Fourier representation $f = \exp(ikj\Delta z - i\omega n \Delta t)$ and carrying out a few lines of algebra, yields

$$\frac{\partial f}{\partial z} \to f_j^n e^{i(k\Delta z - \omega \Delta t)/2} \left[\cos(\omega \Delta t/2) + (1-2\delta) i \sin(\omega \Delta t/2)\right] \frac{i \sin(k\Delta z/2)}{\Delta z/2}.$$

Differencing the time derivative term in as was done in Eq.(12) of KRT, using the Fourier representation, and carrying out a few lines of algebra, the time derivative become

$$\frac{\partial f}{\partial t} \to -f_j^n e^{i(k\Delta z - \omega \Delta t)/2} \; \frac{i \sin(\omega \Delta t/2)}{\Delta t/2} cos(k\Delta z/2)$$

Define

$$\hat{k} = \frac{\sin(k\Delta z/2)}{\Delta z/2}$$

$$\hat{\omega} = \frac{\sin(\omega \Delta t/2)\cos(k\Delta z/2)}{(\Delta t/2)[\cos(\omega \Delta t/2) + (1-2\delta) i \sin(\omega \Delta t/2)]}$$

In the well resolved limits ($k\Delta z/2 \ll 1$) and $\omega \Delta t/2 \ll 1$), these formulas simplify to $\hat{k} = k$ and $\hat{\omega} = \omega$ so that one can think of $\hat{k}$ and $\hat{\omega}$ as discrete versions of k and ω. Note that in the centered implicit limit ($\delta = 0.5$), the expression for $\hat{\omega}$ will give a real result for real ω. The discretized linearized Vlasov equation can then be written as

$$f_j^n = -\frac{q\Phi_j^n}{T} \frac{i\hat{k}v}{-i\hat{\omega} + i\hat{k}v} f_M$$

and the linearized density response is

$$\tilde{n}_j^n = -\frac{q\Phi_j^n}{T} \int dv \frac{i\hat{k}v}{-i\hat{\omega} + i\hat{k}v} f_M = -\frac{q\Phi_j^n}{T} n_0 (1 + \hat{\zeta}\hat{Z}(\hat{\zeta})) \tag{2}$$

where $\hat{\zeta} = \hat{\omega}/(\hat{k} v_t \sqrt{2})$, and $\hat{Z}$ is a discrete approximation to the integration that defines the plasma dispersion function Z. (GS2 uses high order Gaussian integration for such integrals.) For comparison, Langdon79's Eq.(11b) for the dielectric function can be written in the $\Delta t \rightarrow 0$ limit as

$$0 = \epsilon(k, \omega) = 1 + \frac{\omega_{pe}^2}{k^2 v_{te}^2}(1 + \zeta Z(\zeta))$$

Thus it would seem that Eq.(2) would be a good approximation to the real plasma response in the limit $\omega \Delta t/2 \ll 1$ (and $k \Delta z/2 \ll 1$). Note that there is no requirement that fast transit time scales must be resolved. I.e., the Courant condition $\Delta t v/\Delta z < 1$ can be violated. For certain problems, such as electron Landau damping of ion acoustic waves in 1-D or electron resonant destabilization of a drift wave in an unsheared slab, to accurately get the imaginary part of the Z function corresponding to resonant damping will require a fine velocity space grid to be able to resolve the slow electrons with $v \approx \omega/k \approx c_s \ll v_{te}$ that are resonant with the wave. However, for many problems of fusion interest, the dominant source of non-adiabatic electrons is the trapped electrons. The trapped electrons give a resonant response when ω is comparable to their toroidal precession frequency ω_d, which doesn't require such a fine velocity grid to resolve. The KRT paper describes how the implicit method is applied in the presence of trapped particles (there is a symmetry boundary condition imposed at the mirror points.)

Subtleties that I could elaborate on: recurrence phenomena in continuum codes, Landau damping transfers information to small scales in velocity space (see the pictures of the water-bag simulation of a two-stream instability in Chen's textbook). As is well known, a small but finite level of collision will wipe out such small scale effects.

4 Misc.

The algorithm described in the KRT paper uses centered spatial differencing. A parameter was introduced into the code after the KRT paper was written to allow for an adjustable amount of upwind differencing. Sometimes the code is run with a slight upwind bias, which provides some damping for high $k_{\parallel}$ modes. A future upgrade might be to implement higher-order upwind differencing to reduce the artificial damping, and perhaps to add sub-grid scale models to handle the damping at the smallest scales in a controlled way. (In a kinetic phase-space code the "Reimann" problem is easy because the direction of the particle velocity is known—you don't have to decompose into multiple waves travelling in various directions—so I think higher-order upwind methods will be easier to implement.)