Appendix C

ASH CODE TIME STEPPING

We discuss here in more detail the time stepping scheme employed by the ASH code. The basic problem is how one numerically solves an initial value problem of the form

$$\frac{\partial y}{\partial t} = f(y(t), t). \tag{C.1}$$

Since any higher-dimensional differential equation can be broken down into a system of one-dimensional equations, solving the one-dimensional problem is sufficient. The ASH code uses both the Crank-Nicholson and Adams-Bashforth methods to advance the solution in time, with each described in turn.

Subscripts are used here to denote the index of the time step. For example, y_i denotes the value of y at the most recently computed time step (occurring at time t_i) and y_{i+1} denotes the yet-to-be-computed value of y at the next time step. The time separation between subsequent time steps is $\Delta t_i \equiv t_{i+1} - t_i$.

C.1 THE SECOND-ORDER CRANK-NICHOLSON METHOD

As a first step in solving equation (C.1), one notices that the function f is simply the slope of y, suggesting the formula

$$\frac{y_{i+1} - y_i}{\Delta t_i} = f_i \qquad \text{or} \qquad y_{i+1} = y_i + \Delta t_i \, f_i, \tag{C.2}$$

commonly referred to as the *forward Euler* method. This method is explicit, meaning that y_{i+1} is the only unknown quantity in the formula and thus can be readily computed.

The accuracy of this method is determined by computing the local truncation error (LTE), which is simply the residual when the true values of y and f are substituted into equation (C.2):

$$\mathrm{LTE} = y(t_i + \Delta t_i) - y(t_i) - \Delta t_i\, f(y(t_i), t_i). \tag{C.3}$$

By noting that the Taylor expansion of $y(t_i + \Delta t_i)$ is

$$y(t_i + \Delta t_i) = y(t_i) + \Delta t_i \left.\frac{dy}{dt}\right|_{t_i} + \frac{\Delta t_i^2}{2} \left.\frac{d^2y}{dt^2}\right|_{t_i} + \cdots , \tag{C.4}$$

and by using the fact that $f = \dfrac{dy}{dt}$, substituting equation (C.4) into equation (C.3) gives

$$\mathrm{LTE} = \frac{\Delta t_i^2}{2} \left.\frac{d^2y}{dt^2}\right|_{t_i} + \cdots . \tag{C.5}$$

Since the local truncation error only contains terms of second and higher order in Δt_i, the forward Euler method is accurate to first order.

We now consider the *backward Euler* method:

$$\frac{y_{i+1} - y_i}{\Delta t_i} = f_{i+1}. \tag{C.6}$$

The only difference between the backward and forward Euler methods is the use of the unknown quantity f_{i+1} rather than the known quantity f_i on the right-hand side. Because we have to simultaneously solve for both y_{i+1} and f_{i+1}, this method is now implicit, and consequently requires a matrix inversion if f is linear in y. If f is nonlinear in y, implicit schemes such as the backward Euler become even more problematic to implement, and as a result they are generally only used to solve linear equations. As with the forward Euler method, the backward Euler method is accurate to first order.

The Crank-Nicholson method is a weighted average of the forward and backward Euler methods presented above:

$$\frac{y_{i+1} - y_i}{\Delta t_i} = \Theta f_{i+1} + (1 - \Theta) f_i, \tag{C.7}$$

where Θ is an adjustable parameter ranging from 0 (fully explicit) to 1 (fully implicit). The Crank-Nicholson method can be shown to be accurate to second order. In addition, this method is numerically stable, meaning that there is no limit on the time step size one can use (though the accuracy of the solution might decrease if too large a time step is chosen). In contrast, the forward Euler method is conditionally stable: choosing too large of a time step will cause numerical errors to grow, eventually overwhelming the actual solution.

C.2 THE SECOND-ORDER ADAMS-BASHFORTH METHOD

We now examine explicit multistep methods, where the results from more than one previous time step are used to solve equations of the form (C.1). In this case, the values of y_i, f_i and f_{i-1} (where the last quantity is either computed from a previous time step or stated as initial conditions) are already available. We now assume the unknown quantity y_{i+1} is given by the general form

$$\frac{y_{i+1} - y_i}{\Delta t_i} = \alpha f_i + \beta f_{i-1} \qquad \text{or} \qquad y_{i+1} = y_i + \Delta t_i(\alpha f_i + \beta f_{i-1}), \tag{C.8}$$

where the constant coefficients α and β will be determined to minimize the local truncation error.

For this method to be second-order accurate, we require the local truncation error to vanish through terms of order Δt_i^2. As with the forward Euler method in the previous section, the local truncation error is calculated by substituting into equation (C.8) the true values of y and f at the various time steps and calculating the residual. That is,

$$\text{LTE} = y(t_i + \Delta t_i) - y(t_i) - \Delta t_i \left[\alpha f(y(t_i), t_i) + \beta f(y(t_i - \Delta t_{i-1}), t_i - \Delta t_{i-1})\right]. \tag{C.9}$$

We first must evaluate the Taylor series expansions of y_{i+1} and f_{i-1} around t_i. These expansions are given by

$$y(t_i + \Delta t_i) = y_i + \Delta t_i \left.\frac{dy}{dt}\right|_{t_i} + \frac{\Delta t_i^2}{2} \left.\frac{d^2y}{dt^2}\right|_{t_i} + \frac{\Delta t_i^3}{6} \left.\frac{d^3y}{dt^3}\right|_{t_i} + \cdots \tag{C.10}$$

and

$$f(y(t_i - \Delta t_{i-1}), t_i - \Delta t_{i-1}) = f_i - \Delta t_{i-1} \left.\frac{df}{dt}\right|_{t_i} + \frac{\Delta t_{i-1}^2}{2} \left.\frac{d^2 f}{dt^2}\right|_{t_i} - \frac{\Delta t_{i-1}^3}{6} \left.\frac{d^3 f}{dt^3}\right|_{t_i} + \cdots . \tag{C.11}$$

Using these Taylor expansions, and the fact that $\frac{df}{dt} = y$, $\frac{d^2 f}{dt^2} = \frac{dy}{dt}$, etc., we can now write equation (C.9) in terms of the coefficients α and β, plus the dependent variable y and its derivatives evaluated at the current time step t_i. Grouping by power of Δt_i, we now have

$$\mathrm{LTE} = \Delta t_i \left.\frac{dy}{dt}\right|_{t_i} (1 - \alpha - \beta) + \Delta t_i^2 \left.\frac{d^2 y}{dt^2}\right|_{t_i} \left(\frac{1}{2} + \beta \frac{\Delta t_{i-1}}{\Delta t_i}\right) + \cdots , \tag{C.12}$$

where only terms of first and second order in Δt_i are listed explicitly. Since we require the parenthesized expressions to vanish for the method to be second-order accurate, we must have

$$\alpha = 1 + \frac{1}{2}\frac{\Delta t_i}{\Delta t_{i-1}} \qquad \text{and} \qquad \beta = -\frac{1}{2}\frac{\Delta t_i}{\Delta t_{i-1}}. \tag{C.13}$$

Substituting these values of α and β into equation (C.8), we obtain the second-order Adams-Bashforth formula:

$$\frac{y_{i+1} - y_i}{\Delta t_i} = \left(1 + \frac{1}{2}\frac{\Delta t_i}{\Delta t_{i-1}}\right) f_i - \left(\frac{1}{2}\frac{\Delta t_i}{\Delta t_{i-1}}\right) f_{i-1}. \tag{C.14}$$

Note that higher-order Adams-Bashforth formulae can be derived in a similar manner.

C.3 COMBINING THE METHODS

The ASH code evolution equations are of the form

$$\frac{\partial y}{\partial t} = \mathcal{L} + \mathcal{N}, \tag{C.15}$$

where y can represent any dependent variable in the system of equations (W, Z, p, s), and $\mathcal{L}$ and $\mathcal{N}$ respectively designate all linear and nonlinear source terms. Generally, these source terms can be a function of any of the dependent variables as well as a function of the independent variables $\vec{\boldsymbol{r}}$ and t.

Combining the two methods, equation (C.15) becomes

$$\frac{y_{i+1} - y_i}{\Delta t_i} = \left(1 + \frac{1}{2}\frac{\Delta t_i}{\Delta t_{i-1}}\right)\mathcal{N}_i - \left(\frac{1}{2}\frac{\Delta t_i}{\Delta t_{i-1}}\right)\mathcal{N}_{i-1} + \Theta\mathcal{L}_{i+1} + (1-\Theta)\mathcal{L}_i, \tag{C.16}$$

which is equation (4.61) in Chapter 4.