Astrophysics Graduate Course
Solving Ordinary Differential Equations Numerically
(Ph. Podsiadlowski, Oxford, MT05)

1 Textbooks

- Numerical Recipes by Press et al., chapters 16 and 17: good standard reference with powerful, up-to-date, ready-to-use algorithms.
- Applied Numerical Analysis by C. F. Gerald, P. O. Wheatley (Addison-Wesley), chapter 5: good basic introduction to the problem.

2 General Principles

A set of coupled ordinary differential equations (of arbitrary order) can be written as a set of coupled, first-order differential equations of the form

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = f_i(x, y_1, y_2, \ldots, y_N) \quad (i = 1, \ldots, N),$$

where x is the independent variable, the y_i's N dependent variables, and f_i are functions of these $N + 1$ variables (example: *one* second-order equation can be reduced to *two* first-order equations by defining a new dependent variable $y_2 = \mathrm{d}y_1/\mathrm{d}x$).

General considerations in designing a numerical scheme:

- accuracy: controlling the local and global errors;
- efficiency: using a higher-order scheme may require many fewer integration steps;
- stability: iterative schemes may not converge, treatment of singularities;
- hardware issues: e.g. memory constraints, speed of I/O operations involving harddisk access.

3 Simple Integration Schemes

Consider a simple first-order equation (generalizations are trivial)

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x, y).$$

3.1 Euler method

Start integration at some point x_0 and advance integration in steps of length h. If x_n and y_n are the values of x and y after n steps, one can find the value of y_{n+1} from

$$y_{n+1} = y_n + h\, f(x_n, y_n) + O(h^2).$$

This simple scheme is not very efficient; it produces a local error of order h^2, implying a global error of order h.

3.2 Runge-Kutta methods

To obtain a better scheme, one can Taylor expand $y_{n+1}(x+h)$ to second order

$$y_{n+1} = y_n + hf(x_n, y_n) + (h^2/2)\, f'(x_n, y_n),$$

and use the ansatz

$$\begin{aligned} y_{n+1} &= y_n + ak_1 + bk_2, \\ k_1 &= hf(x_n, y_n), \\ k_2 &= hf(x_n + \alpha h, y_n + \beta k_1). \end{aligned}$$

By comparing these two expressions, these can be made identical (up to second order), e.g., if $a = b = 1/2$, $\alpha = \beta = 1$. This scheme gives a local error of order h^3 (global error: h^2).

Fourth-order Runge-Kutta method

A powerful commonly used scheme uses a Taylor expansion to fourth order leading to

$$\begin{aligned} y_{n+1} &= y_n + \frac{1}{6}\,(k_1 + 2k_2 + 2k_3 + k_4), \\ k_1 &= hf(x_n, y_n), \\ k_2 &= hf(x_n + 1/2h, y_n + 1/2k_1), \\ k_3 &= hf(x_n + 1/2h, y_n + 1/2k_2), \\ k_4 &= hf(x_n + h, y_n + k_3). \end{aligned}$$

This scheme gives local errors of order h^5 and global errors of order h^4. In order to obtain an estimate of the numerical error, one can use schemes that use five estimates of Δy (Runge-Kutta-Merson) or six estimates (Runge-Kutta-Fehlberg). This allows to vary the step size subject to chosen accuracy criteria.

3.3 Bulirsch-Stoer method

For differential equations that only involve smooth functions and have no singularities, more powerful methods have been developed in recent years, e.g. the Bulirsch-Stoer method, that uses rational function extrapolations and a sequence of smaller and smaller substeps which are then extrapolated to zero stepsize; this allows very large main steps.

4 Complications

4.1 Singularities and discontinuities

Singularities and discontinuities generally need special attention. For example, singular boundary conditions ($\mathrm{d}y/\mathrm{d}x = x$, where $x_0 = 0$) can often be replaced by non-singular boundary conditions by Taylor-expanding the equation(s) about the singular point and shifting the boundary point by a small Δx (in the example use $y(\Delta x) = 0.5(\Delta x)^2$). For unresolved shocks, use an artificial viscosity term to ensure the Rankine-Hugoniot relations across shocks.

4.2 Stiff equations

Stiff differential equations are equations that involve more than one scale and where the scales are of very different orders.

Example: pp chain in hydrogen burning

$$\begin{aligned}
\frac{\mathrm{d}n_{\mathrm{H}}}{\mathrm{d}t} &= -2\lambda_{\mathrm{pp}}\,\frac{(n_{\mathrm{H}})^2}{2} - \lambda_{\mathrm{pd}}\,n_{\mathrm{H}}n_{\mathrm{D}} + 2\lambda_{33}\,\frac{(n_{^3\mathrm{He}})^2}{2},\\
\frac{\mathrm{d}n_{\mathrm{D}}}{\mathrm{d}t} &= \lambda_{\mathrm{pp}}\,\frac{(n_{\mathrm{H}})^2}{2} - \lambda_{\mathrm{pd}}\,n_{\mathrm{H}}n_{\mathrm{D}},\\
\frac{\mathrm{d}n_{^3\mathrm{He}}}{\mathrm{d}t} &= \lambda_{\mathrm{pd}}\,n_{\mathrm{H}}n_{\mathrm{D}} - 2\lambda_{33}\,\frac{(n_{^3\mathrm{He}})^2}{2},\\
\frac{\mathrm{d}n_{^4\mathrm{He}}}{\mathrm{d}t} &= \lambda_{33}\,\frac{(n_{^3\mathrm{He}})^2}{2}.
\end{aligned}$$

Here, $\lambda_{\mathrm{pd}} \gg \lambda_{33} \gg \lambda_{\mathrm{pp}}$. To solve this in a simple fashion, one would have to use ridiculously short time steps that prevent large changes in n_{D}, the most variable element, in each step.

Solutions:

- If interested in fairly long time scales only, one can assume that D and ^{3}He are in equilibrium (i.e. $\mathrm{d}n_{\mathrm{D}}/\mathrm{d}t = \mathrm{d}n_{^3\mathrm{He}}/\mathrm{d}t = 0$) and reduce the equations to two simpler *non-stiff* differential equations.
- If interested in all but the shortest time scales, one can assume that D is in equilibrium, solve the equation for ^{3}He *analytically* (assuming that H and ^{4}He do not change much during a time step) and then use the analytical solution $n_{^3\mathrm{He}}(t)$ when solving for n_{H} and $n_{^4\mathrm{He}}$.
- Use powerful modern methods that efficiently deal with stiff equations (see Numerical Recipes).

5 Boundary value problems

Most problems are boundary value problems where values are given at certain boundaries and solutions are sought that satisfy these boundary conditions. For example, the four equations of stellar structure require four boundary conditions, two at the centre and two at the surface; this is a *mixed boundary-value problem.*

5.1 Shooting methods

One way to solve problems with mixed boundary conditions (i.e. at a and b) is to guess the values of the unknown values at each boundary and integrate from both boundaries to some chosen point. By adjusting these guesses, one can then develop an iterative scheme that after a few interations makes the integrations from both boundaries match at the chosen matching point.

5.2 Newton-Raphson schemes

An alternative method is to guess a solution of the problem (with the correct boundary values), check the errors made in the guess and then correct for this using a Taylor expansion.

Example:

Rewrite the N differential equations as

$$\Psi_i(\mathbf{y}) = 0, \quad \mathbf{y} \equiv (y_1, \ldots, y_N), \tag{1}$$

where the y_i's are the dependent variables we want to solve for. If $\mathbf{y}^{(0)}$ denotes an approximate solution, and if the residual errors of this approximate solution are given by

$$b_i^{(0)} \equiv \Psi_i(\mathbf{y}^{(0)}), \quad i = 1, \ldots, N,$$

one can obtain an improved solution $\mathbf{y}^{(1)} = \mathbf{y}^{(0)} + \delta\mathbf{y}^{(0)}$, by Taylor-expanding equation (1) to obtain a matrix equation

$$\left[\partial_j \Psi_i(\mathbf{y}^{(0)})\right] \delta y_j^{(0)} = -b_i^{(0)}, \quad i = 1, \ldots, N.$$

This basically requires an *efficient* way of solving the matrix equation. It provides an efficient *iterative* scheme to obtain the desired solution, *provided that the initial guess is relatively good.*