

Due in the locker in the basement of the Burke Science Building by 15:00 on 20 March 2000.

## Stiffness in ODE's

The three properties that determine the usefulness of a numerical method are its domain of *stability* (the parameter range for which the approximation errors are not amplified) its *order* (how the magnitude of the truncation error scales with the stepsize) and its *convergence* (the numerical approximation converges to the true solution as the stepsize goes to zero). In this project we investigate *stiff* ordinary differential equations: equations with two or more widely differing scales. For example, the solution could have a component that quickly becomes insignificant, and another component that changes much more slowly. An example is coupled chemical reactions where reaction constants can differ by many orders of magnitude.

These equations have the property that the usual numerical methods are only stable at very small stepsizes. This means that in order to achieve stability one must use stepsizes much smaller than required to achieve a reasonable numerical accuracy. In addition, such small stepsizes may lead to significant roundoff errors. By using an inappropriate method for a stiff problem one ends up doing thousands of times too much work, and sometimes the problem becomes impossible.

The goal of this project is to explore the concept of stiffness: what it is, how it arises and how to treat it appropriately. We consider three examples of increasing complexity which illustrate these questions.

1. Consider the simple system of linear ordinary differential equations:

$$\begin{aligned} y_1' &= -y_1 + 0.999 y_2 \\ y_2' &= \quad \quad - 0.001 y_2 \end{aligned} \quad (1)$$

The analytic solution for this equation with  $y_1(0) = c_1 + c_2, y_2(0) = c_2$  is

$$y_1 = c_1 e^{-t} + c_2 e^{-0.001t} \quad y_2 = c_2 e^{-0.001t}.$$

- [10] (a) If  $c_1 = 0$  and  $c_2 = 1$  then the exact solution of (1) is  $y_1 = y_2 = \exp(-0.001t)$  which satisfies the simpler system of equations:

$$y_1' = -0.001y_1 \quad y_2' = -0.001y_2 \quad (2)$$

This system of equations can be solved accurately by almost any method provided the step size is less than about 100 (why?).

Now solve the original system (1) numerically with  $c_1 = 0$  and  $c_2 = 1$  using an explicit Euler method and small step size  $h = 5$  until  $t = 150$  and plot the relative error. What happens?

Explain why the solution is now unstable at  $h = 5$  for large  $t \approx 80$  even though the equivalent linear system (2) can be solved accurately for time steps up to 100. How might roundoff error play a role? How small does the time step have to be for the errors to be reasonable?

- [10] (b) Use the implicit (backwards) Euler method to solve (1) with  $c_1 = 0$  and  $c_2 = 1$  and step size  $h = 5$  over the same time interval and plot the relative error. What happens now? Increase the step size until the relative error becomes too large (say 10%), what is this maximum step size? Try to explain what is happening.

- [10] (c) In the previous example with  $c_1 = 0$   $c_2 = 1$  the fast time scale (associated with the term  $\exp(-t)$ ) does not appear in the solution (even though it is present in the linear system 1). What happens if  $c_1 = c_2 = 1$  so the fast time scale term plays a role? How small a time step must be used?
- One way to get around the problem of an initially small time step is to use a variable time step: small while the fast time time scale is important and then larger when it becomes insignificant. Solve the equations with  $h = 0.1$  up to  $t = 5$  and then increase the time step to  $h = 5$  (or larger) for the remainder. How do the implicit and explicit Euler methods compare?
- [10] (d) Plot the stability diagrams of the explicit and implicit Euler methods and use to explain the results you have found in this section.
- [10] (e) The linear system (1) is a specific example of a more general problem: the solution of a homogeneous system of linear equations with constant coefficients

$$y'_i = \sum_{j=1}^m a_{ij} y_j$$

which can be written in matrix notation as

$$y' = Ay. \tag{3}$$

We assume that the numbers of equations and unknowns are equal, so  $A$  is a square  $m \times m$  matrix. The eigenvalues of  $A$  determine the stability of the numerical methods used to solve (3). In most cases of interest  $A$  has  $m$  independent eigenvalues defined by the relation

$$A\phi_k = \lambda_k \phi_k, \quad k = 1, 2, \dots, m.$$

What are the eigenvalues of (1)? How do they relate to the slow and fast time scales? Can you suggest a general criterion for stability of the explicit Euler method for solving a general linear system of equations based on the eigenvalues of  $A$ ? Can the eigenvalues be used to determine when the problem is likely to be stiff?

2. In the first part of the project we considered the stiffness problem for systems of linear ordinary differential equations. We now look at the same problem for nonlinear ordinary differential equations. The van der Pol equation describes an oscillator with a nonlinear friction term (friction constant  $\mu$ ):

$$\begin{aligned} y'_1 &= y_2 \\ y'_2 &= \mu(1 - y_1^2)y_2 - y_1 \end{aligned} \tag{4}$$

with initial conditions  $y_1(0) = 2$ ,  $y_2(0) = 0$ . This nonlinear equation can have a variety of complicated quasi-periodic solutions depending on the value of the constant  $\mu$ . Note that the period of the solution is proportional to  $\mu$ .

The van der Pol equation (4) is a particular example of the general case of a system of first-order nonlinear differential equations which can be written in the form

$$y'_i(x) = f_i(x, y_1, y_2, \dots, y_n), \quad i = 1, \dots, n. \tag{5}$$

In the case of linear equations we saw that the eigenvalues of the coefficient matrix  $A$  control the stability of the problem. Here, the role of the matrix  $A$  is played by the Jacobian matrix  $\partial f_i / \partial x_j$ . The eigenvalues of the Jacobian matrix at a particular value of  $y_i$  determines the stability characteristics of the problem for those  $y_i$  (just like the eigenvalues of  $A$ ). This

can be seen by linearizing  $f_i$  in the neighbourhood of a smooth solution  $y_i(x) = \phi_i(x)$  of (5):

$$y'_i(t) = f_i(x, \phi_i(x)) + \sum_{j=1}^n \frac{\partial f_i}{\partial y_j}(x, \phi_i(x))(y_j(x) - \phi_j(x)) + \dots \quad (6)$$

and introducing  $y_i(x) - \phi_i(x) = \bar{y}_i(x)$  to obtain

$$\bar{y}'_i(x) = \sum_{j=1}^n \frac{\partial f_i}{\partial y_j}(x, \phi_i(x))\bar{y}_j(x) + \dots = \sum_{i=0}^n J_{ij}(x)\bar{y}_j(x) + \dots \quad (7)$$

As a first approximation we consider the Jacobian  $J_{ij}(x) = \partial f_i / \partial x_j$  as constant and neglect the error terms. Omitting the bars and writing the resulting expression in matrix notation we have

$$\mathbf{y}' = J\mathbf{y}. \quad (8)$$

Equation (8) is an approximation to the exact equation (5) in the case of small changes in  $y_i$ , which is sufficient for analyzing the stability of the numerical method since  $y_i$  should change only slightly from over one step size  $h$ . Note that (8) is identical in form to the equation for the linear case (3). The stability of a numerical method for a particular problem is found by determining whether  $\lambda_i h$  is within the stability domain of the method for all eigenvalues  $\lambda_i$  of  $J$ .

- [10] (a) Solve van der Pol's equation (4) with  $\mu = 1$  using the fifth-order Dormand & Prince algorithm (i.e. `Matlab` function `ode45`) with a tolerance of  $1.0E - 03$  (default) and plot the result until  $t = 3$ . [Hint: you should use the `Matlab` function `vdpode`.] How many steps are used?
- [10] (b) Now solve (4) with increasing  $\mu = 10, 20, 50, 100$  until  $t = 3\mu$  using the same method as above and plot the results. What happens? How many steps are used? Does the adaptive stepsize solve the problem? Explain why this is an example of stiffness (note that shape of solution does not change much).
- [10] (c) The stability domain for `ode45` is given by  $|R(z)| \leq 1$  where

$$R(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} + \frac{z^6}{600}. \quad (9)$$

Plot the stability domain defined by (9). Re-run the above simulations, and estimate the time step required for the calculation to be stable according to (9) (you will use `vdpode` to generate the Jacobian and `eig` to calculate eigenvalues). How do these values compare with the actual time steps used? Do these results help explain the stiffness? Try to determine the value of  $\mu$  for which the equations become stiff (e.g. when the simulation requires 10 times as many steps as for  $\mu = 1$ ).

- [10] (d) Use `Matlab`'s stiff method `ode15s` to solve (4) for  $\mu = 1, 10, 20, 50, 100, 1000, 10000$  with default tolerance ( $1.0E - 03$ ) and plot the results. How many steps were used? How does this compare with `ode45`? Is stiffness a problem? What can you deduce about the stability domain of `ode15s`?
- [10] 3. Describe (in your own words) what stiffness is and why it is a problem for numerical methods.

[Total: 100]