Lecture 1

Introduction

The great majority of differential equations which describe real systems cannot be solved analytically, the equations can only be solved approximately using numerical algorithms. It is also the case that a great many problems are set in an evolutionary framework: given the state at one time, determine the state at some future time. In this course we try to set out the methodology whereby the behaviour of numerical algorithms to approximate such differential systems can be studied systematically and the advantages, accuracy and pitfalls of such algorithms can be understood. Many systems, such as used in aircraft control, power station control, guidance systems are operated without human intervention according to computed solutions of differential systems, so we have great interest in knowing that solutions are computed accurately and efficiently.

Some examples of real systems are easy to find.

1. Motion under a central force field

There are many examples in molecular or stellar dynamics where motion is determined by a central force field. The general title for this is an N-body problem. The example you are most likely to have already seen is that of just two bodies moving in a plane with a gravitational field. This simplifies to a one-body problem if one of the bodies has mass (M) sufficiently massive that its position can be taken as fixed at an origin and the position of the second body, of mass m be given by $(\hat{x}(\hat{t}), \hat{y}(\hat{t}))$ with velocities $(\hat{u}(\hat{t}), \hat{v}(\hat{t}))$ as unknowns. Then, in a gravitational problem, applying simple mechanics shows that the motion will be described by

$$m\frac{\mathrm{d}^2\hat{x}}{\mathrm{d}\hat{t}^2} = -\frac{GMm}{\hat{r}^3}\hat{x},$$

$$m\frac{\mathrm{d}^2\hat{y}}{\mathrm{d}\hat{t}^2} = -\frac{GMm}{\hat{r}^3}\hat{y},$$

where $\hat{r}^2 = \hat{x}^2 + \hat{y}^2$ and G is a universal constant, $G = 6.67 \times 10^{-11}$ m^{3}kg^{-1}s^{-2}.

If these equations are rescaled with time scale T_s and length scale $L_s = (GMT_s^2)^{1/3}$, so that $(\hat{x}, \hat{y}) = L_s(x, y)$, $\hat{t} = T_s t$ and $(\hat{u}, \hat{v}) = (L_s/T_s)(u, v)$, then the equations can be rewritten

$$\frac{\mathrm{d}x}{\mathrm{d}t} = u, \tag{1.1}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = v, \tag{1.2}$$

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -\, x(x^2+y^2)^{-3/2}, \tag{1.3}$$

$$\frac{\mathrm{d}v}{\mathrm{d}t} = -\, y(x^2+y^2)^{-3/2}. \tag{1.4}$$

Using the vector $\mathbf{u}^{\mathrm{T}} = [x\ y\ u\ v]$, then the system is in the form

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{f}(\mathbf{u}), \tag{1.5}$$

and solutions can be found starting from some initial data $\mathbf{u}(0) = \mathbf{u}_0 \in \mathbb{R}^4$.

In a general N-body problem with motion in three dimensions we will have $\mathbf{u}, \mathbf{u}_0 \in \mathbb{R}^{6N}$ so that the position and velocity of N-bodies can be determined. There is a broad literature on the N-body problem, for example when the bodies are of equal mass, there are surprisingly many closed loop periodic solutions with the general title of N-body choreographies. As an example, it was known to Lagrange that equal mass bodies could rotate on a circular path, with, in the case of say 3 bodies, the particles located at the vertices of an equilateral triangle. However, for three equal mass bodies, there is also a closed loop periodic solution where the bodies move in a figure of eight pattern, and there are many periodic solutions when the number of particles is increased to four or more. These choreographies are not part of this course but they are an illustration of the importance of robust, accurate and stable numerical computations in generating solutions for complex N-body dynamics.

2. Reaction kinetics - enzyme catalysed reaction

There are many situations in chemistry and biological systems that lead to a set of differential equations to describe evolution from some starting state. An example from reaction kinetics comes from supposing a substrate S (concentration s) combines with an enzyme E (concentration e) at rate α to form a complex C (concentration c) and that the complex forms a product P (concentration p) at rate β (and in doing so releases enzyme E at the same rate) but also breaks back down into S and E at rate

γ. Then simple reaction kinetics give

$$\frac{\mathrm{d}s}{\mathrm{d}t} = -\alpha s e + \gamma c, \tag{1.6}$$

$$\frac{\mathrm{d}e}{\mathrm{d}t} = -\alpha s e + (\gamma + \beta) c, \tag{1.7}$$

$$\frac{\mathrm{d}c}{\mathrm{d}t} = \alpha s e - (\gamma + \beta) c, \tag{1.8}$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \beta c, \tag{1.9}$$

and again defining a vector $\mathbf{u}^{\mathrm{T}} = [s \ e \ c \ p]$ then this system can also be put in the form

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{f}(\mathbf{u}). \tag{1.10}$$

and solved starting from some initial data $\mathbf{u}(0) = \mathbf{u}_0 \in \mathbb{R}^4$. The dimension of $\mathbf{u}$, $\mathbf{u}_0$ will match the number of concentrations being tracked, in this case four.

3. Lorenz model

Lorenz studied motion in a layer of fluid where there is a temperature difference across the layer and he represented the flow and temperature using Fourier series. He then made a rather severe truncation for the fluid stream function to one term and the temperature to two terms so the model had three unknown functions, which might be given by notation $(u(t), v(t), w(t))$ and then conservation of momentum and energy, when non-dimensionalised give

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -\sigma u + \sigma v, \tag{1.11}$$

$$\frac{\mathrm{d}v}{\mathrm{d}t} = -uw + \rho u - v, \tag{1.12}$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = uv - \beta w, \tag{1.13}$$

where σ, ρ and β are non-dimensional parameters. The system too can be put in a standard framework by $\mathbf{u}^{\mathrm{T}} = [u \ v \ w]$ with

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{f}(\mathbf{u}), \tag{1.14}$$

and solved starting from some initial data $\mathbf{u}(0) = \mathbf{u}_0 \in \mathbb{R}^3$.

4. General Formulation

These example systems have forcing function f which is not explicitly dependent on time but

it is possible to have examples where time is an explicit argument in f so we can formulate the very general problem:

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{f}(t, \mathbf{u}). \tag{1.15}$$

where $\mathbf{u}, \mathbf{f} \in \mathbb{R}^k$ for some k and we seek solutions starting from some initial data $\mathbf{u}(0) = \mathbf{u}_0 \in \mathbb{R}^k$.

In setting out the framework for numerical approximation of differential systems, we will mostly look at the case $k = 1$ but do keep in mind that most real examples involve application to larger systems which involve solution of many simultaneous differential equations.

It is easy to think that the numerical solution of differential equations should be a straight forward process of just approximating the derivatives in the system in some systematic way. What can possibly go wrong?

Using Matlab

The numerical analysis we study in this course centres on analytic methods to study discrete approximations to differential equations. As we shall see for ODEs and particularly for PDEs, methods to approximate differential equations may appear to have certain required properties, such as known error estimates, but nevertheless, when applied with finite accuracy computers, they fail. Hence an important adjunct to analysis is implementation of a method. Included in the lectures will be many matlab examples and I have included some matlab exercises in the tutorial sheets. Oxford has a site wide licence for matlab and you are encouraged to be familiar with writing small matlab procedures. The examination will be based only on analysis but using matlab to study how numerical algorithms behave gives insight into analysis and ultimately, will help in mastering this course.

I suggest you create a directory or folder for this course and add that to the default matlab search path and then keep routines you have written, or course ones I provide, in that folder. I will discuss finite precision arithmetic and using matlab in the first lecture and there are many guides for using matlab that you can consult.

5. Example of a numerical problem

To illustrate that possibly unexpected difficulties can occur when using numerical algorithms, consider this simple model problem:

$$\begin{aligned} u'(t) &= 0, \quad t > 0 && (1.16)\\ u(0) &= 1 && (1.17)\end{aligned}$$

with obvious exact solution $u(t) = 1$ for $t > 0$. Suppose the function $u(t)$ is approximated at time values $t_n = n\Delta t$ by U_n, so that $U_n \approx u(t_n)$. Then setting $U_0 = 1$ we would expect $U_n = 1$ for $n = 1, 2, 3, \ldots$ to be the numerical solution. Note the notation which will be used consistently throughout the course: u is the *continuous* function that is the solution of a differential equation, U_n will usually be a set of *discrete* values that are intended to approximate u at certain times t_n and $u_n = u(t_n)$ are discrete values of the correct solution at the discrete times. This means we have two discrete sets, $\{t_n\}$, and $\{U_n\}$, although if the time points are uniform, the first set is trivial.

When dealing with sets of discrete elements we can a process that can be repeatedly applied to elements in turn as an operator, for example, a forward difference operator might be written

$$\Delta_+[U_n] = U_{n+1} - U_n, \quad n = 0, 1, 2, \cdots,$$

and Δ_+ is an operator on the diescrete set (often the brackets [] are left out). It is possible to develop an algebra for operators, see particularly Iserles book if interested.

While it may not be immediately obvious why, consider a discrete operator N which defines the combination of values:

$$N[u_n] = \frac{-u_{n+1} + 3u_n - 2u_{n-1}}{\Delta t}. \tag{1.18}$$

If we suppose that Δt is small and use Taylor expansions

$$\begin{aligned} u_{n+1} = u(t_{n+1}) = u(t_n + \Delta t) &\approx u(t_n) + \Delta t u'(t_n) + \frac{\Delta t^2}{2} u''(t_n) + \ldots && (1.19)\\ u_{n-1} = u(t_{n-1}) = u(t_n - \Delta t) &\approx u(t_n) - \Delta t u'(t_n) + \frac{\Delta t^2}{2} u''(t_n) + \ldots && (1.20)\end{aligned}$$

then

$$N[u_n] = \frac{-u_n + 3u_n - 2u_n + \Delta t u_n' - \frac{3}{2}\Delta t^2 u_n'' + \ldots}{\Delta t}, \tag{1.21}$$

and so

$$N[u_n] = u_n' - \frac{3}{2}\Delta t u_n'' + \ldots, \tag{1.22}$$

where we have simplified the notation beyond just $u_n = u(t_n)$ by having $u'_n = u'(t_n)$, again, notation that we will use consistently throughout the course.

However, what (1.22) shows is that we can regard N as an approximation to the derivative u', since clearly

$$u'_n \quad = \quad N[u_n] + \frac{3}{2}\Delta t u''_n \dots \tag{1.23}$$

and so we might expect for small Δt that

$$u'_n \quad \approx \quad N[u_n], \tag{1.24}$$

or that if we wished to approximate the solution of

$$u' = f,$$

then we could use

$$\frac{U_{n+1} - 3U_n + 2U_{n-1}}{\Delta t} = f(t_n),$$

to generate approximate solutions that would become more accurate as Δt became smaller.

However, when the equation we are trying to integrate is just $u' = 0$, consider values U_n satisfying $N[U_n] = 0$, or equivalently,

$$U_{n+1} = 3U_n - 2U_{n-1}. \tag{1.25}$$

This should be an iteration that can be used to successively calculate an approximation for the values $u_{n+1},\ n = 1, 2, 3, \dots$ that all satisfy $U_n = u_0$.

We will see later that this is an example of a multistep iteration where we need two starting values, U_0 and U_1 in order to apply the first iteration to calculate U_2, and then we can calculate $U_3, \dots$ successively.

In the matlab demo, I have taken $U_0 = 1$ and $U_1 = 1+e^{-15}$, a little more than machine precision might have. The result is a sequence of vlaues that grow exponentially, see figure 1.

We know that $U_0 = 1$ is the correct initial value. Suppose the initial value $U_1 = 1 + \epsilon$ so we have some small error of ϵ only in U_1 and exact value for U_0. Then applying this recursion manually:

$$U_2 \quad = \quad 1 + 3\epsilon, \tag{1.26}$$
$$U_3 \quad = \quad 1 + 7\epsilon, \tag{1.27}$$
$$U_4 \quad = \quad 1 + 15\epsilon. \tag{1.28}$$

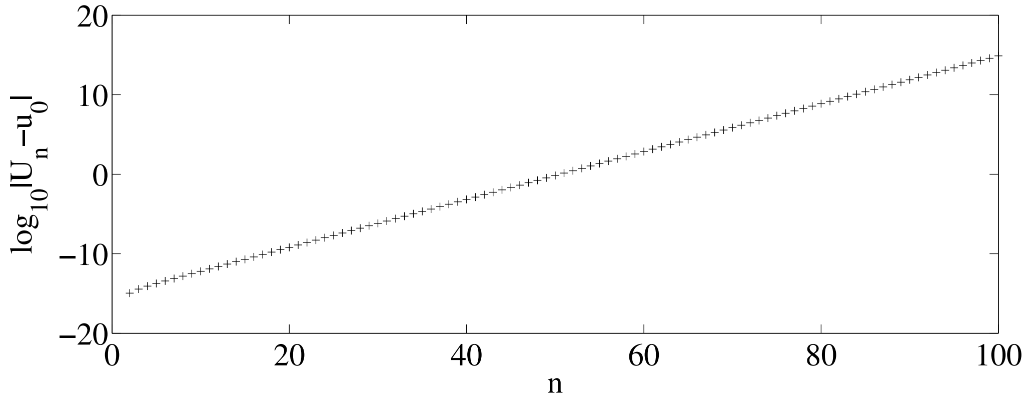


Figure 1: Result of applying iteration $U_{n+2} = 3U_{n+1} - 2U_n$ with $U_0 = 1$ and $U_1 = 1.000000000000001$.

The general solution of the *difference scheme* (1.25) for the starting values $U_0 = 1$, $U_1 = 1+\epsilon$, is $U_n = 1 + (2^n - 1)\epsilon$ and the values we will calculate diverge exponentially with the number of steps, since $2^n = \mathrm{e}^{n \log 2}$. So this very trivial model problem shows that numerical algorithms can go wrong and if they do, it is likely that divergence will be exponential in the number of steps, and so have an observable effect after relatively few calculations. In the matlab example, the computation has an error $\epsilon = 10^{-15}$ in U_1 and the iteration should be noticably diverging from the correct value $u_n = 1$ by an order one value after say m iterations, where $2^m \times 10^{-15} \sim 1$, or $m \sim 21$ iterations.

Lecture 2

6. Picard's Theorem

Returning to the general problem but with a single dependent variable, we will suppose that the dependent variable $u(t)$ is the exact solution of a differential system:

$$\mathcal{N}(t,u) = u' - f(t,u) = 0,\ t > 0, \tag{2.1}$$

together with appropriate initial condition $u(0) = u_0$. We can assume without loss of generality that the initial value is at $t = 0$. Later when we deal with partial differential equations both initial and boundary values will be needed but we will still assume the time dependent variable starts at $t = 0$.

The existence of, or uniqueness for, solutions of differential systems is a major field in itself and outside the material of this course. In the main, we will assume that a solution exists in the region of computation but just be aware that this can be a dangerous assumption. There is one result though which is important enough that the statement of the theorem should be known and understood.

Theorem: Picard. *Suppose that $f(t,u)$ is a continuous function of t and u in a region $\Omega = [0,T) \times [u_0 - \alpha, u_0 + \alpha]$ of the (t,u) plane and that there exists $L > 0$ such that*

$$|f(t,u) - f(t,v)| \quad \le \quad L|u - v| \quad \forall t,u,v \in \Omega. \tag{2.2}$$

In this expression L is called a Lipschitz constant and f is said to satisfy a Lipschitz condition with respect to the second argument. Suppose also that

$$MT \quad \le \alpha, \tag{2.3}$$

where $M = \max_\Omega |f|$.

Then there exists a unique continuously differentiable function $u(t)$ defined on $[0,T)$ satisfying

$$\begin{aligned} \frac{\mathrm{d}u}{\mathrm{d}t} &= f(t,u), \quad 0 < t < T & (2.4)\\ u(0) &= u_0. & (2.5)\end{aligned}$$

7. Lipschitz conditions

As well as understanding the importance of Picard's theorem in giving conditions for a unique solution to exist, you should understand what this theorem does not guarantee.

Consider this example:

$$u' = u^2, \quad u(0) = u_0. \tag{2.6}$$

Provided $|u| < \alpha$ then the function $f(u) = u^2$ will satisfy a Lipschitz condition

$$|f(u) - f(v)| \leq 2\alpha |u - v|, \tag{2.7}$$

so it might be tempting to say, that Picard's theorem will guarantee a solution. We need to be careful. If we look at the theorem, then for this case $M = \alpha^2$ so the condition $MT < \alpha$ is really $T < 1/\alpha$. If we look at the analytic solution for $u_0 > 0$,

$$u(t) = \frac{u_0}{1 - u_0 t},$$

this exhibits finite time blow up a $t = 1/u_0$ and Picard's theorem tells us only that for any starting value, there will be an interval where a solution will exist but even though we can find a Lipschitz condition for the function f, we are not guaranteed a solution u for all times. Solutions to equation (2.6) are shown in figure 2.

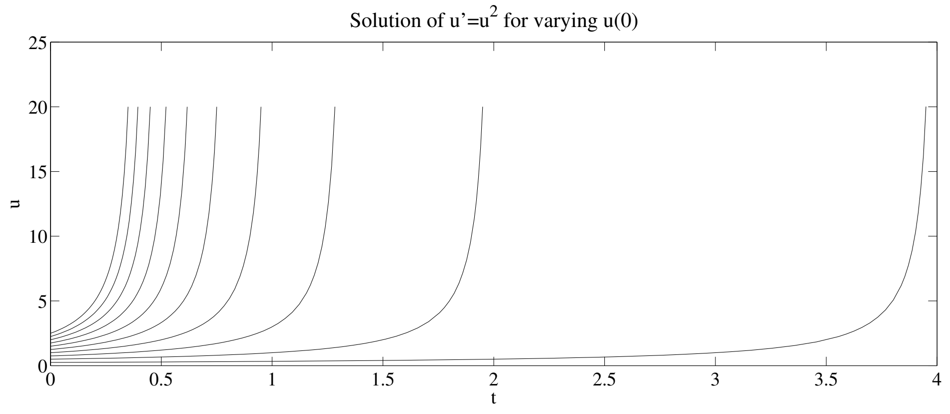


Figure 2: Solutions of the equation $u' = u^2$ for various initial values showing approach to finite time blow up.

Quite simple functions can fail to satisfy a Lipschitz condition. Let

$$f(u) \quad = \quad \sqrt{u} \tag{2.8}$$

on $[a, 1]$ for $a \geq 0$ where in Figure 3 is shown both f and the tangent line through $(a, \sqrt{a})$

$$y(u) = \sqrt{a} + \frac{1}{2\sqrt{a}}(u - a).$$

We have

$$\begin{aligned} f(u) - f(a) &= \sqrt{u} - \sqrt{a} && (2.9) \\ &\leq \sqrt{a} + \frac{1}{2\sqrt{a}}(u - a) - \sqrt{a} && (2.10) \\ &\leq \frac{1}{2\sqrt{a}}(u - a). && (2.11) \end{aligned}$$

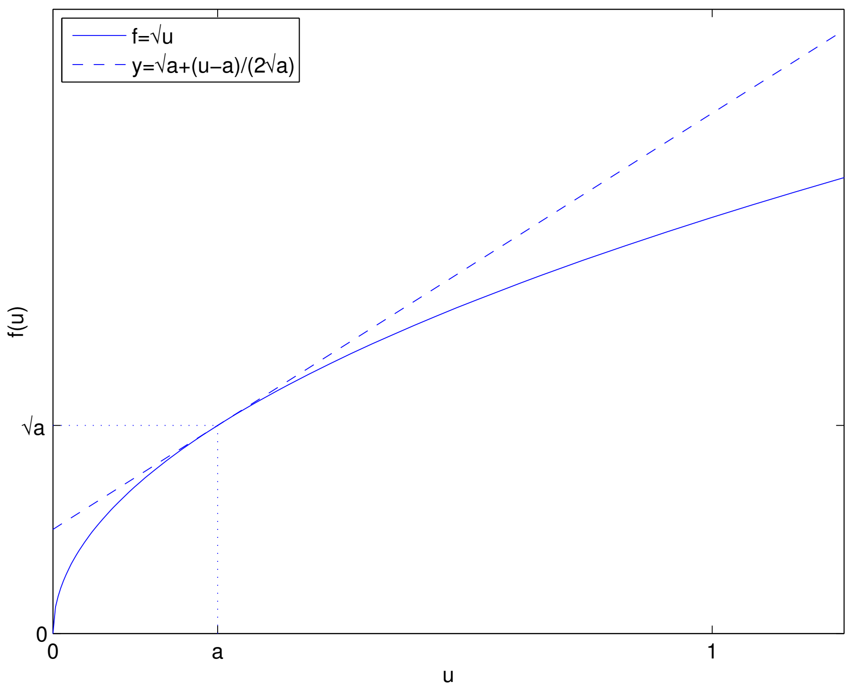


Figure 3: The function $f(u) = \sqrt{u}$ which does not satisfy a Lipschitz condition on $[0, 1]$.

Taking absolute values gives

$$|f(u) - f(a)| \;\leq\; \frac{1}{2\sqrt{a}}|u - a| \tag{2.12}$$

and $\frac{1}{2\sqrt{a}}$ diverges as $a \to 0$ so f is not Lipschitz on $[0, 1]$.

However, it is clear that for $u_0 \geq 0$, $u' = \sqrt{u}$ will have a solution $u(t) = (\sqrt{u_0} + \frac{1}{2}t)^2$ which satisfies $u(0) = u_0$ for all $t > 0$.

Thus we can find systems where a Lipschitz condition does not guarantee a solution and systems where there may not be a Lipschitz condition but a solution still exists, but nevertheless, applying a Lipschitz condition in analysis is hugely important and you need to be able to recognise when such conditions can help with analysis, here often when looking into errors of numerical schemes.

One step methods

A major family of solvers for ODEs are classed as one-step methods where, given data for one time $t = t_n$, the objective is to determine values at a subsequent time

$t = t_{n+1}$. This may result in an algorithm which involves a number of sub-steps within the interval $[t_n, t_{n+1}]$ but there is no supposition that we know or use any details of the solution prior to $t = t_n$ and in principle, no assumption that the values $t_0, t_1, \ldots, t_n, t_{n+1}$ are evenly spaced, and as we shall see later, part of the power of one step methods is that the step length can be varied depending on whether the solution is varying rapidly (and so needs to have short step lengths) or slowly (and so can have longer step lengths) in order to maintain a relatively constant error bound.

We will develop the theory of one-step methods in two parts, first by looking at the method of Euler and some of the associated ideas, particularly for notation, and second, looking at a general framework for one-step methods.

Explicit Euler method for an ODE

We use a standard notation that the continuous system is

$$\mathcal{N}(t, u,) \;=\; \frac{\mathrm{d}u}{\mathrm{d}t} - f(t, u) \;=\; 0. \tag{2.13}$$

As already pointed out, while we will take $u(t)$ to be a single real function, we also note that in principle we could have $u \in \mathbb{R}^k$ for some integer $k > 1$. Since

$$\left.\frac{\mathrm{d}u}{\mathrm{d}t}\right|_t \;=\; \lim_{\Delta t \to 0} \frac{u(t + \Delta t) - u(t)}{\Delta t}, \tag{2.14}$$

and with $t = t_n = n\Delta t$,

$$\left.\frac{\mathrm{d}u}{\mathrm{d}t}\right|_{t_n} \;\approx\; \frac{u(t_n + \Delta t) - u(t_n)}{\Delta t}. \tag{2.15}$$

(i) Explicit Euler: algorithm

Define $u_n = u(t_n)$ and let U_n be our approximation to u_n. Then using the simple definition of a derivative, we can approximate the continuous system $\mathcal{N}$ by a discrete system N where

$$N(t_n, U_n, U_{n+1}) \;=\; \frac{U_{n+1} - U_n}{\Delta t} - f(t_n, U_n) \;=\; 0, \tag{2.16}$$

and the initial condition $u(0) = u_0$ gives

$$U_0 \;=\; u_0. \tag{2.17}$$

Then (2.16) provides an *algorithm*, called Explicit Euler:

$$U_{n+1} \;=\; U_n + \Delta t f(t_n, U_n), \quad n = 0, 1, 2, 3, \ldots. \tag{2.18}$$

Note that the indexing we will normally use is that we have an approximate solution at $t = t_n$ and the discrete scheme or the algorithm provides an approximation at $t = t_{n+1}$, however, when we come to analyse the scheme for accuracy, stability and so on, we will generally apply that at index n.

(ii) Explicit Euler: Matlab implementation

The one step explicit Euler algorithm is very easy to implement, you need to remember that matlab vectors start with index 1, so that if we let $U(1) = u_0$, the fragment of code to carry out N steps, each of size dt will be

```
U(1)=U0;
t(1)=0;
for n=1:N,
    t(n+1)=t(n)+dt;
    U(n+1)=U(n)+dt*f(n*dt,U(n));
end
```

and the values at $t(1), \cdots, t(N+1)$ will be in $U(1), \cdots, U(N+1)$.

This data can now be graphed. Formally, we have a discrete set of data values, $(t_n, U_n),\ n = 1 \dots N+1$, however, conventionally this is graphed as a series of striaght line segments between the discrete data:

```
plot(t,U)
set(gca,'FONTSIZE',18)
xlabel('Time','FONTSIZE',18)
ylable('U','FONTSIZE',18)
```

In this set of statements line 1 plots the data, line 2 sets the current axes (gca means get-current-axis) fontsize to be 18 point, and then lines 3 & 4 label the axes. Please do keep in mind that fonts may need to be larger than default size if they are to be easily readable, particularly if the graph is exported and will then be reduced in size in a document

(iii) Explicit Euler: truncation error

Having set the discrete approximation, (2.16), it is natural to ask how well the discrete system approximates the continuous system. Throughout this course the **truncation error** is defined as the residual when the **true** values for $u(t)$ are inserted into the discrete operator N, that is

$$T_n = N(t_n, u_n, u_{n+1}). \tag{2.19}$$

So in this case

$$T_n = \frac{u_{n+1} - u_n}{\Delta t} - f(t_n, u_n), \tag{2.20}$$

and using a Taylor expansion for u_{n+1},

$$u_{n+1} = u_n + \Delta t u_n' + \frac{1}{2}\Delta t^2 u''(\xi),\ \xi \in (t_n, t_{n+1}), \tag{2.21}$$

we calculate

$$T_n = \frac{1}{2}\Delta t u''(\xi), \tag{2.22}$$

so that provided u'' is bounded, $T_n = \mathcal{O}(\Delta t)$ as $\Delta t \to 0$, that is the scheme N is a first order approximation to $\mathcal{N}$ for small time steps.

In some places you may find a *local truncation error* defined for the *algorithm* assuming that the *true* value u_n is used to determine the approximation over one step, denoted $\tilde{U}_{n+1}$ here. For the Euler scheme this gives

$$\tilde{U}_{n+1} = u_n + \Delta t f(t_n, u_n), \tag{2.23}$$

and a *local truncation error*, denoted $\tilde{T}_n$ here,

$$\tilde{T}_n = u_{n+1} - \tilde{U}_{n+1} = u_{n+1} - u_n - \Delta t f(t_n, u_n) = \frac{1}{2}\Delta t^2 u''(\xi). \tag{2.24}$$

As a generalisation for our continuous system, the local truncation error (which is an error in the approximation of u and not an error in approximation of the differential equation) will be one order higher than the discrete system truncation error (which is an error in approximating $\mathcal{N}$ by N). If there is a local truncation error $O(\Delta t^{p+1})$ and this error occurs for n steps, the solution error after n steps will be $O(n\Delta t^{p+1})$ and letting $n\Delta t \to t$ as $\Delta t \to 0$ then the error in $u(t)$ will be $O(\Delta t^p)$. So the explicit Euler system has a second order local truncation error but as we shall see later when we relate the error,

$$e_n = u_n - U_n, \tag{2.25}$$

to the truncation error T_n, explicit Euler is a first order scheme for calculating $u(t)$.

To repeat: it is important to distinguish the discrete system (2.16) from the algorithm (2.18). To calculate the truncation error in the sense used here, you should use the discrete system, (2.16), and not the algorithm, (2.18), and you should distinguish this system truncation error from the local truncation error in the approximation.

An alternative derivation of the explicit Euler algorithm comes from integrating the equation over $[t_n, t_{n+1}]$,

$$u(t_{n+1}) \quad = \quad u(t_n) + \int_{t_n}^{t_{n+1}} f(s, u(s))\,\mathrm{d}s. \tag{2.26}$$

The integral in this expression could be approximated by

$$\int_{t_n}^{t_{n+1}} f(s, u(s))\,\mathrm{d}s \quad \approx \quad \Delta t f(t_n, u_n). \tag{2.27}$$

This does leave a problem in that we do not know u_n, only U_n, and replacing both u_n and u_{n+1} with approximations U_n, U_{n+1} we get

$$U_{n+1} \quad = \quad U_n + \Delta t f(t_n, U_n), \quad n = 1, 2, 3, \ldots, \tag{2.28}$$

which in this case is the same as that obtained by discretising the derivative directly.

It is important to appreciate that there are two approximations being made: one is in discretising the derivative or integral, depending on how the continuous system is set out, but there is a second approximation in calculating f with an approximate value U_n and not the correct value u_n. As a general notational point, we will later use f_n for $f(t_n, u_n)$ and F_n for $f(t_n, U_n)$ although for the moment we will for clarity continue with the longer form.

It is also reasonable to ask why we focus here on the differential form of the continuous system when the integral form is equally correct, and indeed it is possible to write the differential system as an integral system by redefining $\mathcal{N}$ by

$$\mathcal{N}(t, u) = u(t) - u_0 - \int_0^t f(s, u(s))\mathrm{d}s = 0. \tag{2.29}$$

However, a discrete version of this integral form $\mathcal{N}$ at t_{n+1} will likely involve all the previous calculated values $U_n, U_{n-1}, \ldots, U_1$, and the complexity of analysing this leads to focus on the one step integral form

$$u_{n+1} = u_n + \int_{t_n}^{t_{n+1}} f(s, u(s))\mathrm{d}s, \tag{2.30}$$

with associated local truncation error analysis and global truncation error analysis of this equation.

(iv) Explicit Euler: systems

When we have a higher order system, for example

$$\begin{aligned} u'' &= f(t, u, u') && (2.31)\\ u(0) &= u_0 && (2.32)\\ u'(0) &= u'_0, && (2.33) \end{aligned}$$

we re-write it as a set of first order equations

$$\begin{aligned} u' &= v \qquad\qquad v(0) = u_0' && (2.34)\\ v' &= f(t,u,v) \quad u(0) = u_0 && (2.35)\end{aligned}$$

so that

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\begin{array}{c} u \\ v \end{array}\right] = \left[\begin{array}{c} v \\ f(t,u,v) \end{array}\right] \tag{2.36}$$

and defining vectors

$$\mathbf{u} = \left[\begin{array}{c} u \\ v \end{array}\right], \quad \mathbf{f} = \left[\begin{array}{c} v \\ f(t,u,v) \end{array}\right] \tag{2.37}$$

we have

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{f}(t,\mathbf{u}), \quad \mathbf{u}(0) = \left[\begin{array}{c} u_0 \\ u_0' \end{array}\right], \tag{2.38}$$

and the Explicit Euler method would be

$$\mathbf{U}_{n+1} = \mathbf{U}_n + \Delta t \mathbf{f}(t_n, \mathbf{U}_n). \tag{2.39}$$

In terms of the scalar components, for (2.31)-(2.33),

$$\begin{aligned} U_{n+1} &= U_n + \Delta t V_n, && (2.40)\\ V_{n+1} &= V_n + \Delta t f(t_n, U_n, V_n). && (2.41)\end{aligned}$$

In summary. the general notation which will be used in this course is:

$$\text{Continuous system} \quad \left\{\begin{array}{l} \mathcal{N}(t,u) = \dfrac{\mathrm{d}u}{\mathrm{d}t} - f(t,u) = 0, \quad 0 < t < T \\ u(0) = u_0 \end{array}\right. \tag{2.42}$$

$$\text{Discrete system} \quad \left\{\begin{array}{l} N(t_n, U_n, U_{n+1}) = \dfrac{U_{n+1} - U_n}{\Delta t} - f(t_n, U_n) = 0, \\ \quad \text{with} \quad t_n = n\Delta t, \; n = 0, 1, 2, \ldots, \frac{T}{\Delta t} - 1 \\ U_0 = u_0 \end{array}\right. \tag{2.43}$$

Lecture 3

(v) Explicit Euler: Error

Define two *errors*:

- **solution error** $e_n = u_n - U_n$
- **truncation error** $T_n = N(t_n, u_n, u_{n+1})$

You should appreciate that as U_n and U_{n+1} satisfy $N(t_n, U_n, U_{n+1}) = 0$, and are approximations for u_n and u_{n+1}, the truncation error $T_n = N(t_n, u_n, u_{n+1})$ will not be zero. However, the dependence of T_n on Δt will show how well the *discrete system* approximates the *continuous differential system.* An important goal is to relate e_n to T_n since we can usually estimate T_n using Taylor series and so a bound on T_n can lead to a bound on e_n.

As just set out, for the explicit Euler method

$$T_n = N(t_n, u_n, u_{n+1}) = \frac{u_{n+1} - u_n}{\Delta t} - f(t_n, u_n) \tag{3.1}$$

but

$$u_{n+1} = u_n + \Delta t u'(t_n) + \frac{1}{2}\Delta t^2 u''(\xi_n), \tag{3.2}$$

for some $\xi_n \in (t_n, t_{n+1})$ and so

$$T_n = \frac{u_n + \Delta t u'(t_n) + \frac{1}{2}\Delta t^2 u''(\xi_n) - u_n}{\Delta t} - f(t_n, u_n) \tag{3.3}$$

and using $u'(t_n) = f(t_n, u_n)$

$$T_n = \frac{1}{2}\Delta t u''(\xi_n). \tag{3.4}$$

Provided u'' can be bounded, say $\|u''\|_\infty \le M$ for some M, we have

$$|T_n| \le \frac{1}{2}\Delta t M. \tag{3.5}$$

When $|T_n| = \mathcal{O}(\Delta t^p)$ as $\Delta t \to 0$ we call the method order p.

We can look at errors by subtracting the two equations:

$$\frac{u_{n+1} - u_n}{\Delta t} - f(t_n, u_n) = T_n, \tag{3.6}$$

$$\frac{U_{n+1} - U_n}{\Delta t} - f(t_n, U_n) = 0, \tag{3.7}$$

so that

$$\frac{e_{n+1}-e_n}{\Delta t} - [f(t_n,u_n) - f(t_n,U_n)] \;=\; T_n. \tag{3.8}$$

If we rearrange this

$$e_{n+1} \;=\; e_n + \Delta t\,[f(t_n,u_n) - f(t_n,U_n)] + \Delta t T_n \tag{3.9}$$

and now assume that f satisfies a Lipschitz condition with constant L then

$$|e_{n+1}| \;\leq\; |e_n| + \Delta t L|u_n - U_n| + \Delta t|T_n| \tag{3.10}$$

or

$$|e_{n+1}| \;\leq\; (1+\Delta t L)|e_n| + \Delta t|T_n|. \tag{3.11}$$

Let $T = \max_n |T_n|$ and $\Delta t L = \mu$, then

$$|e_{n+1}| \;\leq\; (1+\mu)|e_n| + \mu\frac{T}{L}. \tag{3.12}$$

It is then be shown by induction (exercise) that

$$|e_n| \;\leq\; (1+\mu)^n|e_0| + [(1+\mu)^n - 1]\,\frac{T}{L} \tag{3.13}$$

and using $1+\mu < \mathrm{e}^{\mu}$, for $\mu > 0$

$$|e_n| \;\leq\; \mathrm{e}^{n\mu}|e_0| + [\mathrm{e}^{n\mu} - 1]\,\frac{T}{L} \tag{3.14}$$

$$\;\leq\; \mathrm{e}^{Lt_n}|e_0| + \left[\mathrm{e}^{Lt_n} - 1\right]\frac{1}{2}\frac{M}{L}\Delta t. \tag{3.15}$$

Hence, provided $e_0 = 0$ then as $t_n \to t$, $\Delta t \to 0$, $n \to \infty$ we should have $|e_n| \to 0$ linearly as $\Delta t \to 0$ and so in terms of the time step used, explicit Euler has first order truncation error and gives a first order accurate solution.

Summary

- Continuous system $u(t) : \mathcal{N}(t,u) = 0$ for $t > 0$ with $u(0) = u_0$,
- Discrete system $\{U_n\} : N(t_n, U_n, U_{n+1}) = 0$ for $n = 0, \ldots$ with initial data U_0 given,
- Error in solution $e_n = u_n - U_n$ (where $u_n = u(t_n)$)
- Truncation error $T_n = N(t_n, u_n, u_{n+1})$
- The scheme is order p provided $T_n = O(\Delta t^p)$ as $\Delta t \to 0$,
- If we can bound e_n using T_n then provided $T_n \to 0$ we will have $e_n \to 0$ as $\Delta t \to 0$,
- The discrete system is a *consistent* approximation to the continuous system provided $T_n \to 0$ as $\Delta t \to 0$.

More on One step methods

In trying to improve on the simple explicit Euler method we can start by considering the integral form of the solution

$$u_{n+1} \quad = \quad u_n + \int_{t_n}^{t_{n+1}} f(s, u(s))\,\mathrm{d}s. \tag{3.16}$$

Denote $g(s) = f(s, u(s))$. As a general problem, if we want to estimate $\int_{t_n}^{t_{n+1}} g(s)\,\mathrm{d}s$, there are many quadrature methods which might be used.

In explicit Euler we used

$$\int_{t_n}^{t_{n+1}} g(s)\,\mathrm{d}s \quad \approx \quad \Delta t g(t_n), \tag{3.17}$$

and approximated $g(t_n) = f(t_n, u_n)$ by $g(t_n) \approx f(t_n, U_n)$. We know from the mean value theorem that there is one value $t^\star \in [t_n, t_{n+1}]$ such that

$$\int_{t_n}^{t_{n+1}} g(s)\,\mathrm{d}s \quad = \quad \Delta t g(t^\star) \tag{3.18}$$

but we don't know $t^\star$, and even if we did, we could not evaluate $g(t^\star) = f(t^\star, u(t^\star))$ without also knowing $u(t^\star)$. However, this does suggest that we might be able to 'sample' g at one or more points in the interval $[t_n, t_{n+1}]$ in order to better approximate an integral over the interval. However, since we start with only the value U_n, if we are to sample the function g at other time points, we need to set out how we will approximte the function u at these points in order to evaluate $g = f(t, u)$

(i) Theta-method: algorithm

One method, called sometimes a theta-method uses a weighted average of two values evaluated at either end of the interval to give a quadrature formula

$$\int_{t_n}^{t_{n+1}} g(s)\,\mathrm{d}s \quad \approx \quad \Delta t\left[(1-\theta) g(t_n) + \theta g(t_{n+1})\right], \tag{3.19}$$

and inserting approximate values for u, gives a discrete system:

$$N(t_n, t_{n+1}, U_n, U_{n+1}) \quad = \quad \frac{U_{n+1} - U_n}{\Delta t} - \left[(1-\theta) f(t_n, U_n) + \theta f(t_{n+1}, U_{n+1})\right] = 0 \tag{3.20}$$

The case $\theta = 1/2$ is of course the trapezoidal rule. Also as soon as $\theta > 0$ we no longer have a simple explicit formula for U_{n+1} as the value also appears as an argmuent in the funtion f. Such schemes have the general label of being **implicit**.

For the ODE this gives a discrete **implicit** *algorithm*:

$$U_{n+1} \quad = \quad U_n + \Delta t \left[(1-\theta) f(t_n, U_n) + \theta f(t_{n+1}, U_{n+1})\right] \quad n = 0, 1, 2, \ldots. \tag{3.21}$$

To repeat, the algorithm is implicit because the value U_{n+1} appears inside the function f on the RHS of this equation. As already noted, we distinguish the discrete system (3.20) from the algorithm (3.21) that would be programmed in order to compute an approximate solution. In this implicit theta-method, because U_{n+1} appears on the right-hand-side inside the function f, U_{n+1} is determined by

$$U_{n+1} - \theta \Delta t f(t_{n+1}, U_{n+1}) \quad = \quad U_n + (1-\theta) \Delta t f(t_n, U_n) \quad n = 0, 1, 2, \ldots. \tag{3.22}$$

This nonlinear equation for U_{n+1} would need to be solved by some iterative method such as Newton iteration, fixed point iteration or interval halving iteration at each time step.

The implicit scheme with $\theta = \frac{1}{2}$ is

$$U_{n+1} \quad = \quad U_n + \frac{1}{2} \Delta t \left[f(t_n, U_n) + f(t_{n+1}, U_{n+1})\right], \tag{3.23}$$

and is the same as using the trapezoidal rule for approximating the integral over (t_n, t_{n+1}).

(ii) Theta-method: error

A generalisation of the error analysis used for the explicit Euler method to this implicit form gives

$$|e_n| \quad \leq \quad \mathrm{e}^{\frac{Lt_n}{1-\theta L \Delta t}} |e_0| + \frac{\Delta t}{L} \left\{ \left| \frac{1}{2} - \theta \right| M + \frac{1}{3} \Delta t \|u'''\|_\infty \right\} \left[\mathrm{e}^{\frac{Lt_n}{1-\theta L \Delta t}} - 1 \right]. \tag{3.24}$$

The value $\theta = \frac{1}{2}$ is a special case where the second term is $\mathcal{O}(\Delta t^2)$.

Semi-implicit framework: modified & improved Euler

In order to avoid having to solve a nonlinear iteration in an implicit scheme every time step there are *semi-implicit* methods that rely on using an estimate for the unknown value U_{n+1} to approximate $f(t_{n+1}, U_{n+1})$. A simple example is to first make an estimate for U_{n+1} with

$$U^\star_{n+1} \quad = \quad U_n + \Delta t f(t_n, U_n) \tag{3.25}$$

and then use this estimate, $U^\star_{n+1}$, in the function f:

$$U_{n+1} \quad = \quad U_n + \frac{1}{2} \Delta t \left[f(t_n, U_n) + f(t_{n+1}, U^\star_{n+1})\right]. \tag{3.26}$$

(See example on problem sheet.) This is called improved Euler. There is also a second form commonly used

$$U^\star_{n+1/2} = U_n + \frac{1}{2}\Delta t f(t_n, U_n) \tag{3.27}$$

$$U_{n+1} = U_n + \Delta t f(t_{n+1/2}, U^\star_{n+1/2}). \tag{3.28}$$

This is sometimes called modified Euler.

General one step methods

We can look at one step methods in a general framework by observing that both the improved Euler algorithm

$$U_{n+1} = U_n + \frac{1}{2}\Delta t \left[f(t_n, U_n) + f(t_{n+1}, U_n + \Delta t f(t_n, U_n)) \right]. \tag{3.29}$$

and the modified Euler algorithm

$$U_{n+1} = U_n + \Delta t f(t_n + \frac{1}{2}\Delta t, U_n + \frac{1}{2}\Delta t f(t_n, U_n)), \tag{3.30}$$

can be written in the form

$$U_{n+1} = U_n + \Delta t \Phi(t_n, U_n; \Delta t), \tag{3.31}$$

where Φ is continuous in its variables. A little algebra will show that provided Φ satisfies a Lipschitz condition with constant L with respect to its second argument

$$|\Phi(t, u; \Delta t) - \Phi(t, v; \Delta t)| \leq L|u - v|, \tag{3.32}$$

then our previous analysis still holds. We first note that the algorithmic form of discretisation, (3.31), corresponds to the discrete scheme:

$$N(t_n, U_n, U_{n+1}) = \frac{U_{n+1} - U_n}{\Delta t} - \Phi(t_n, U_n; \Delta t) = 0, \tag{3.33}$$

so that the truncation error is

$$T_n = \frac{u_{n+1} - u_n}{\Delta t} - \Phi(t_n, u_n; \Delta t). \tag{3.34}$$

Let $T = \max_n |T_n|$, then as before

$$|e_n| \leq \mathrm{e}^{Lt_n}|e_0| + \left(\mathrm{e}^{Lt_n} - 1\right)\frac{T}{L}. \tag{3.35}$$

We also see from (3.34) that in the limit $\Delta t \to 0$ and $t_n \to t$ the right-hand-side becomes

$$\lim_{\substack{\Delta t \to 0 \\ n \to \infty \\ n\Delta t \to t}} T_n = u' - \Phi(t, u; 0) \tag{3.36}$$

so that consistency ($T_n \to 0$ as $\Delta t \to 0$) requires that

$$\Phi(t, u; 0) = f(t, u). \tag{3.37}$$

Theorem: Convergence. *Under the conditions assumed for f and Φ we have*

$$|e_n| \;=\; |u_n - U_n| \to 0 \text{ as } \Delta t \to 0,\; n\Delta t \to t \tag{3.38}$$

and $U_n \to u(t)$.

Proof. Write

$$\begin{aligned} T_n &= \frac{u_{n+1}-u_n}{\Delta t} - \Phi(t_n, u_n; \Delta t) &(3.39)\\ &= \frac{u_{n+1}-u_n}{\Delta t} - f(t_n, u_n) + [\Phi(t_n, u_n; 0) - \Phi(t_n, u_n; \Delta t)] &(3.40)\end{aligned}$$

but

$$\frac{u_{n+1}-u_n}{\Delta t} \;=\; u'(\xi) \tag{3.41}$$

for some $\xi \in (t_n, t_{n+1})$ and hence

$$\frac{u_{n+1}-u_n}{\Delta t} - f(t_n, u_n) \;=\; u'(\xi) - u'(t_n) \tag{3.42}$$

and as u' is uniformly continuous with respect to t, for any $\epsilon > 0$, $\exists \Delta t_1$, such that $\Delta t < \Delta t_1 \implies |u'(\xi) - u'(t_n)| \leq \epsilon$.

As Φ is continuous with respect to its arguments, $\exists \Delta t_2$ such that

$$\Delta t < \Delta t_2 \implies |\Phi(t_n, u_n; 0) - \Phi(t_n, u_n; \Delta t)| \leq \epsilon \tag{3.43}$$

and hence

$$\Delta t \leq \min(\Delta t_1, \Delta t_2) \implies |T_n| \leq 2\epsilon \tag{3.44}$$

so that

$$\Delta t \leq \min(\Delta t_1, \Delta t_2) \implies |e_n| \leq (\mathrm{e}^{Lt_n} - 1)\frac{2\epsilon}{L} \tag{3.45}$$

provided $|e_0| = 0$.

Write

$$|u(t) - U_n| \;\leq\; |u(t) - u(t_n)| + \underbrace{|u(t_n) - U_n|}_{e_n} \tag{3.46}$$

so as $t_n \to t$ and as u is continuous, $\exists N, \Delta t_3$ such that $n > N, \Delta t < \Delta t_3 \implies |u(t) - u(t_n)| \leq \epsilon$ and hence for $\Delta t < \min(\Delta t_1, \Delta t_2, \Delta t_3), n > N$

$$|u(t) - U_n| \;\leq\; \epsilon + (\mathrm{e}^{LT} - 1)\frac{2\epsilon}{L} \tag{3.47}$$

where $T = \max\{t\}$. Hence $|u(t) - U_n| \to 0$ as $\Delta t \to 0$, $n\Delta t \to t$. □

Lecture 4

Explicit Runge–Kutta Methods

We can view the one step function Φ as 'sampling' the derivative function f at a number of points in the interval $[t_n, t_{n+1}]$ although of course such 'samples' are always taken at values of u which are only approximations to the correct values. However, if we do this systematically, and use Taylor expansions we can derive families of methods with higher order of accuracy.

Denote a *Runge–Kutta R stage method* where $R = 2, 3, \dots$ as one where we 'sample' the derivative $f(t, u)$ at R points in $[t, u]$ space with t in the interval $[t_n, t_{n+1}]$ with the intention of making the truncation error is as high order in Δt a possible. Here we will focus on explicit methods where the function $f(t, u)$ is always evaluated at known values of both arguments. There are implicit Ruge-Kutta methods where the value of the second argument is not known and some form of iterative solution is necessary within each time step, see for example the more general scheme described in Süli & Mayers.

Assume that the function f is 'sampled' at R points $t_n, t_n + a_2\Delta t, \dots, t_n + a_R\Delta t$ and let

$$\Phi(t_n, U_n; \Delta t) \;=\; c_1k_1 + c_2k_2 + \dots c_Rk_R \;=\; \sum_{r=1}^{R} c_rk_r, \tag{4.1}$$

where

$$\begin{aligned} k_1 &= f(t_n, U_n), && (4.2)\\ k_2 &= f(t_n + a_2\Delta t, U_n + b_{2,1}\Delta t k_1), && (4.3)\\ k_3 &= f(t_n + a_3\Delta t, U_n + b_{3,1}\Delta t k_1 + b_{3,2}\Delta t k_2), && (4.4) \end{aligned}$$

with general term:

$$k_r \;=\; f(t_n + a_r\Delta t, U_n + \Delta t\sum_{s=1}^{r-1} b_{r,s}k_s), \quad r = 2, \dots, R. \tag{4.5}$$

Let also

$$\begin{aligned} a_2 &= b_{2,1}, && (4.6)\\ a_3 &= b_{3,1} + b_{3,2}, && (4.7)\\ a_4 &= b_{4,1} + b_{4,2} + b_{4,3}, && (4.8) \end{aligned}$$

with general term:

$$a_r \;=\; \sum_{s=1}^{r-1} b_{r,s} \quad r = 2, \dots, R. \tag{4.9}$$

We want to determine the coefficients $b_{r,s}$ for $r = 2, \ldots, R$ and $s = 1, \ldots, r-1$ and the constants a_r so that the scheme is consistent and has truncation error of order $O(\Delta t^R)$.

One compact way of displaying the various coefficents is called a Butcher table, which for the way we have formulated the system, is, for say $R = 4$:

$$
\begin{array}{l|llll}
0 & 1 & & & \\
a_2 & b_{2,1} & & & \\
a_3 & b_{3,1} & b_{3,2} & & \\
a_4 & b_{4,1} & b_{4,2} & b_{4,3} & \\
\hline
 & c_1 & c_2 & c_3 & c_4
\end{array}
$$

(i) Consistency

As we saw previously in (3.33), the truncation error here is

$$T_n = \frac{U_{n+1} - U_n}{\Delta t} - \Phi(t_n, U_n; \Delta t) = \frac{u_{n+1} - u_n}{\Delta t} - \sum_{r=1}^{R} c_r \hat{k}_r \tag{4.10}$$

where the $\hat{k}_r$ are evaluated with the correct function values of u. To simplify notation, for $r = 2, 3, \ldots$, denote $\beta_r = \sum_{s=1}^{r-1} b_{r,s} \hat{k}_s$ and $f_n = f(t_n, u_n)$ so that

$$\hat{k}_1 = f_n, \tag{4.11}$$
$$\hat{k}_r = f(t_n + a_r \Delta t, u_n + \beta_r \Delta t), \; r = 2, \ldots, R. \tag{4.12}$$

Expand in time to get

$$\hat{k}_r = f(t_n, u_n + \beta_r \Delta t) + a_r \Delta t \frac{\partial f}{\partial t}(t_n, u_n + \beta_r \Delta t) + \frac{a_r^2 \Delta t^2}{2!} \frac{\partial^2 f}{\partial t^2}(t_n, u_n + \beta_r \Delta t) + \tag{4.13}$$

Now expand in u to get

$$\hat{k}_r = f_n + \beta_r \Delta t \left. \frac{\partial f}{\partial u} \right|_n + \frac{\beta_r^2 \Delta t^2}{2!} \left. \frac{\partial^2 f}{\partial u^2} \right|_n + \frac{\beta_r^3 \Delta t^3}{3!} \left. \frac{\partial^3 f}{\partial u^3} \right|_n + \ldots \tag{4.14}$$
$$+ a_r \Delta t \left. \frac{\partial f}{\partial t} \right|_n + a_r \beta_r \Delta t^2 \left. \frac{\partial^2 f}{\partial t \partial u} \right|_n + \frac{a_r \beta_r^2 \Delta t^3}{2!} \left. \frac{\partial^3 f}{\partial t \partial u^2} \right|_n + \ldots \tag{4.15}$$
$$+ \frac{a_r^2 \Delta t^2}{2} \left. \frac{\partial^2 f}{\partial t^2} \right|_n + \frac{a_r^2 \beta_r \Delta t^3}{2} \left. \frac{\partial^3 f}{\partial t^2 \partial u} \right|_n + \ldots, \tag{4.16}$$

so that

$$\hat{k}_r = f_n + \Delta t \left\{ \beta_r \frac{\partial f}{\partial u} + a_r \frac{\partial f}{\partial t} \right\} + \Delta t^2 \{\ldots\} + \Delta t^3 \{\ldots\}, \; r = 2, \ldots, R. \tag{4.17}$$

Next add up the $\hat{k}_r$ and collect powers of Δt. We need to use

$$u_n' = f_n = f(t,u)|_n \tag{4.18}$$

$$u_n'' = \left.\left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial u}\frac{\mathrm{d}u}{\mathrm{d}t}\right)\right|_n = \left.\left(\frac{\partial f}{\partial t} + f\frac{\partial f}{\partial u}\right)\right|_n \tag{4.19}$$

and so on, with algebra becoming increasingly intricate as R increases. One general observation is that, as $\hat{k}_r = f_n + \mathcal{O}(\Delta t),\ r = 2, 3, \ldots,$ when substituted into the truncation error

$$T_n = \frac{u_{n+1} - u_n}{\Delta t} - \sum_{r=1}^{R} c_r k_r \tag{4.20}$$

$$= u_n' + \mathcal{O}(\Delta t) - \left(\sum_{r=1}^{R} c_r\right) f_n + \mathcal{O}(\Delta t), \tag{4.21}$$

so that

$$T_n = [1 - \left(\sum_{r=1}^{R} c_r\right)]u_n' + \mathcal{O}(\Delta t), \tag{4.22}$$

and we must always require that

$$\sum_{r=1}^{R} c_r = 1,$$

for consistency.

(ii) R = 2 two stage Runge Kutta

It is a class exercise to show that when $R = 2$, the truncation error $T \sim \mathcal{O}(\Delta t^2)$ provided

$$a_2 = b_{2,1}, \tag{4.23}$$

$$c_1 + c_2 = 1, \tag{4.24}$$

$$b_{2,1}c_2 = \frac{1}{2}. \tag{4.25}$$

This creates a one parameter family of methods, all of which are formally second order in Δt. We shall see later that this does not mean that all are usable. If we want the 'sampling' to be within the interval $[t_n, t_{n+1}]$ we need to choose a_2 to be in $(0, 1]$. To simplify notation, let $a_2 = \alpha$ for $0 < \alpha \leq 1$, then

$$a_2 = \alpha, \tag{4.26}$$

$$c_2 = \frac{1}{2\alpha} \tag{4.27}$$

$$c_1 = \frac{2\alpha - 1}{2\alpha}. \tag{4.28}$$

This gives a one parameter family of methods, for $0 < \alpha \leq 1$:

$$\begin{aligned}
k_1 &= f(t_n, U_n) && (4.29)\\
k_2 &= f(t_n + \alpha\Delta t, U_n + \alpha\Delta t k_1) && (4.30)\\
U_{n+1} &= U_n + \frac{\Delta t}{2\alpha}\left\{(2\alpha - 1)k_1 + k_2\right\}. && (4.31)
\end{aligned}$$

Special Cases When $\alpha = 1$ we have improved Euler:

$$\begin{aligned}
k_1 &= f(t_n, U_n) && (4.32)\\
k_2 &= f(t_n + \Delta t, U_n + \Delta t k_1) && (4.33)\\
U_{n+1} &= U_n + \frac{\Delta t}{2}(k_1 + k_2). && (4.34)
\end{aligned}$$

The Butcher table for this scheme is

$$\begin{array}{c|cc}
0 & 1 & \\
1 & 1 & \\
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array}$$

When $\alpha = \frac{1}{2}$ we have modified Euler:

$$\begin{aligned}
k_1 &= f(t_n, U_n) && (4.35)\\
k_2 &= f\left(t_n + \frac{1}{2}\Delta t, U_n + \frac{1}{2}\Delta t k_1\right) && (4.36)\\
U_{n+1} &= U_n + \Delta t k_2. && (4.37)
\end{aligned}$$

The Butcher table for this scheme is

$$\begin{array}{c|cc}
0 & 1 & \\
\frac{1}{2} & \frac{1}{2} & \\
\hline
 & 0 & 1
\end{array}$$

(iii) R = 3 three stage Runge Kutta

$$U_{n+1} = U_n + \Delta t(c_1 k_1 + c_2 k_2 + c_3 k_3). \qquad (4.38)$$

The consistency condition is

$$c_1 + c_2 + c_3 = 1 \qquad (4.39)$$

and $T_n \sim \mathcal{O}(\Delta t^3)$ forces

$$
\begin{aligned}
c_2 b_{2,1} + c_3(b_{3,1} + b_{3,2}) &= \frac{1}{2} && (4.40)\\
c_2 b_{2,1}^2 + c_3(b_{3,1} + b_{3,2})^2 &= \frac{1}{3} && (4.41)\\
c_3 b_{2,1} b_{3,2} &= \frac{1}{6}. && (4.42)
\end{aligned}
$$

The gives a two parameter family of solutions although the popularity of fourth order and the simplicity of second order Runge-Kutta means that third order schemes are not often used. One standard third order Runge Kutta scheme is:

$$
\begin{aligned}
U_{n+1} &= U_n + \frac{1}{6}\Delta t(k_1 + 4k_2 + k_3) && (4.43)\\
k_1 &= f(t_n, U_n) && (4.44)\\
k_2 &= f(t_n + \frac{1}{2}\Delta t, U_n + \frac{1}{2}\Delta t k_1) && (4.45)\\
k_3 &= f(t_n + \Delta t, U_n - \Delta t k_1 + 2\Delta t k_2) && (4.46)
\end{aligned}
$$

The Butcher table for this scheme is

$$
\begin{array}{c|ccc}
0 & 1 & & \\
\frac{1}{2} & \frac{1}{2} & & \\
1 & -1 & 2 & \\
\hline
 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6}
\end{array}
$$

In the case where f is independent of u, $f = f(t)$, this is of course just a variant of Simpson's Rule.

(iv) $\mathbf{R = 4}$ fourth order Runge Kutta

The algebra becomes increasingly tedious as R increases, for $R = 4$ details can be found in many text books and a widely used and powerful case of a fourth order method is

$$
\begin{aligned}
U_{n+1} &= U_n + \frac{1}{6}\Delta t\left\{k_1 + 2k_2 + 2k_3 + k_4\right\} && (4.47)\\
k_1 &= f(t_n, U_n) && (4.48)\\
k_2 &= f(t_n + \frac{1}{2}\Delta t, U_n + \frac{1}{2}\Delta t k_1) && (4.49)\\
k_3 &= f(t_n + \frac{1}{2}\Delta t, U_n + \frac{1}{2}\Delta t k_2) && (4.50)\\
k_4 &= f(t_n + \Delta t, U_n + \Delta t k_3). && (4.51)
\end{aligned}
$$

The Butcher table for this scheme is

$$\begin{array}{c|cccc}
0 & 1 & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & \\
1 & 0 & 0 & 1 & \\
\hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}
\end{array}$$

Runge-Kutta methods are very straight forward to implement and as we shall see next, very easy to adapt to the behaviour of the solution.

Lecture 5

(v) Adaptive step length for Runge Kutta

One step methods can easily be modified to vary the step length Δt as the function u varies in order to account for how rapidly or slowly the solution u is varying with time. There are a number of commonly used adaptive schemes, the underlying principle is to use knowledge of how the error depends on Δt to help estimate whether the error may be increasing beyond some pre-set tolerance (and so smaller time steps should be used), or whether errors are sufficiently below a tolerance that the step size can be increased.

As an example of one adaptive method, consider the Runge-Kutta fourth order method given above, (4.47)-(4.51), denoted RK4. The background theory is that when using RK4 there will be some constants K_1, K_2, K_3, such that

$$\begin{aligned} \text{Trunction error} \quad T_n &\sim K_1(\Delta t)^4 u^{(v)} && (5.1)\\ \text{Local error} \quad |\bar{e}_n| &\sim K_2 \Delta t |T_n| && (5.2)\\ \text{so} \quad |\bar{e}_n| &\sim K_3(\Delta t)^5 u^{(v)}. && (5.3) \end{aligned}$$

Suppose we are at t_n and want to use a step Δt_n

1. apply RK4 over step Δt_n to get value U_a where

$$U_a = u_{n+1} + \frac{(\Delta t_n)^5 u^{(v)}}{5!} + \mathcal{O}(\Delta t_n^6) \tag{5.4}$$

2. apply RK4 *twice* over step $\Delta t_n/2$ to get value U_b

$$U_b = u_{n+1} + \frac{2(\frac{\Delta t_n}{2})^5 u^{(v)}}{5!} + \mathcal{O}(\Delta t_n^6) \tag{5.5}$$

Then

$$U_a - U_b \sim \frac{15}{16}\frac{u^{(v)}}{5!}(\Delta t_n)^5. \tag{5.6}$$

Suppose we had used an alternative step $\overline{\Delta t}_n$ for which this difference would exactly be a pre-set tolerance, that is

$$\textit{tolerance} = \frac{15}{16}\frac{u^{(v)}}{5!}(\overline{\Delta t}_n)^5. \tag{5.7}$$

Thus we can remove the unknown fifth-derivative of u by dividing these expressions

$$\left(\frac{\overline{\Delta t}_n}{\Delta t_n}\right)^5 = \frac{\textit{tolerance}}{U_a - U_b}, \tag{5.8}$$

or

$$\overline{\Delta t}_n = \left(\frac{tolerance}{U_a - U_b}\right)^{1/5} \Delta t_n, \tag{5.9}$$

and this gives an algorithm for adapting the step length: if (5.9) gives a step length which is less than Δt_n, then we repeat the step from t_n with the reduced step length, if (5.9) gives an increased step length, we use that step as Δt_{n+1} and the value $U_{n+1} = U_b$ at t_{n+1}. Of course we have to do more work each step since we effectively apply RK4 *three* times.

Algorithm: User provides start time, end time, tolerance.

```
1. Set TOL=tolerance
2. Set $t_0$ as start time, $\Delta t_0$ as first step length and $n = 0$
3. while $t_n \leq$ end time
    (a) apply RK4 with step $\Delta t_n$ to determine $U_a$ and twice with step $\Delta t_n/2$ to
        calculate $U_b$
    (b) if $|U_a - U_b| >$TOL, step fails, set
            $\Delta t_n = \left(\frac{\texttt{TOL}}{|U_a - U_b|}\right)^{1/5} \Delta t_n$        (5.10)
        and go back to (a). (This reduces the step and repeats.)
        else $|U_a - U_b| \leq$TOL, set
            $\Delta t_{n+1} = \left(\frac{\texttt{TOL}}{|U_a - U_b|}\right)^{1/5} \Delta t_n$      (5.11)
            $U_{n+1} = U_b$
            $t_{n+1} = t_n + \Delta t_n$
            $n = n + 1$
        (this increases the step length for *next* step).
   end while
```

As we have been dealing with local error the lengthening of step can be misleading, the global error has order $(\Delta t)^4$ so in (5.11) above we can use as an alternative

$$\Delta t_{n+1} = \left(\frac{\texttt{TOL}}{|U_a - U_b|}\right)^{1/4} \Delta t_n. \tag{5.12}$$

This is more robust in practice. It is also good practice to set some minimum step length and to alert a user if the predicted step length becomes too small, usually stopping the integration from continuing. We will look at implementing this adaptive algorithm in the routine `rk4a.m`.

The adaptation method above is only one of a number that are widely used. Another strategy is to use the flexibility inherent in Runge-Kutta families of solutions to produce increments with different order of magnitude but using the same time points within the step from t_n to t_{n+1} and then to use the difference between the estimates to control step length.

In the matlab `ode23` family, a second and a third order method is used. The Butcher table is modified slightly so as to have two rows at the bottom showing coefficents for the two different order estimates of the increment. The Butcher table is:

$$\begin{array}{l|rrrr}
0 & 1 & & & \\
1/2 & 1/2 & & & \\
3/4 & 0 & 3/4 & & \\
1 & 2/9 & 3/9 & 4/9 & \\
\hline
 & 2/9 & 3/9 & 4/9 & \\
 & 11/72 & 30/72 & 40/72 & -9/72
\end{array}$$

For this routine, a second order estimate for the update is

$$U_a = U(n) + \frac{\Delta t}{9}(2k_1 + 3k_2 + 4k_3),$$

and this value is used to evaluate k_4 at time $t_n + \Delta t$, and then a third order estimate for the update is

$$U_b = U_n + \frac{\Delta t}{72}(11k_1 + 30k_2 + 40k_3 - 9k_4).$$

With these two estimates, one might crudely say $U_a = u_n + O(\Delta t^2)$ and $U_b \approx u_n$ so that the difference $U_a - U_b$ can be used as a measure of accuracy and by introducing a tolerance, the step length reduced if the difference is too large or increased if the difference is smaller than the tolerance. This is illustrated in the course matlab routine *rk23a.m* and is the basis of the *ode23* family of routines in matlab.

The same idea can be applied using a fourth and fifth order comparison, this is used for *ode45* in matlab. It is based on a method by Dormand and Prince. In this case there are seven calculations to estimate the tangent, f, and the Butcher table is:

0	1						
1/5	1/5						
3/10	3/40	9/40					
4/5	44/45	−56/15	32/9				
8/9	19372/6561	−25360/2187	64448/6561	−212/729			
1	9017/3168	−355/33	46732/5247	49/176	−5103/18656		
1	35/384	0	500/1113	125/192	−2187/6784	11/84	
c_r	35/384	0	500/1113	125/192	−2187/6784	11/84	0
d_r	5179/57600	0	7571/16695	393/640	−92097/339200	187/2100	1/40

As with the 2-3 method, we calculate a fourth order estimate,

$$U_a = U_n + \Delta t(c_1 * k_1 + c_2 * k_2 + c_3 * k_3 + c_4 * k_4 + c_5 * k_5 + c_6 * k_6),$$

and a fifth order estimate

$$U_b = U_n + \Delta t(d_1 * k_1 + d_2 * k_2 + d_3 * k_3 + d_4 * k_4 + d_5 * k_5 + d_6 * k_6 + d_7 * k_7),$$

and use the difference $U_a - U_b$ as an indicator of the error in the step, again repeating the step with a decreased step size if this difference is greater than some pre-set tolerance, and increasing the next time step length if the difference is less that the tolerance. One way of doing this is shown in the course routine `rk45a.m` but it is also the basis for `ode45` in matlab.

Semi-implicit methods: Predictor-Corrector

All implicit methods will in principle require solution of a non-linear equation (or system) at each time step. A version of Newton iteration can be used so that within the time stepping sequence, there is another iteration that provides convergence of the Newton iteration. A drawback with Newton iteration is that the derivative (or for a system, the Jacobian) of the function f is required, this may be known analytically or it may be approximated numerically but this can be a significant problem. Alternative methods for implicit schemes, which are a variant on fixed point methods, and which do not require evaluation of a derviative or Jacobian matrix are often labelled predictor-corrector or semi-implicit schemes These usually combine an explicit 'predictor' step with a semi-implicit iteration even if the latter is only one step and not taken to convergence.

As an example of this idea, consider a fully implicit Euler method

$$U_{n+1} = U_n + \Delta t f(t_{n+1}, U_{n+1}), \tag{5.13}$$

where U_n is the computed value at t_n. Then an explicit predictor for the value U_{n+1} is

$$U_{n+1}^{(0)} = U_n + \Delta t f(t_n, U_n), \tag{5.14}$$

and this allows a corrector step which uses this value in the implicit formula,

$$U_{n+1} = U_n + \Delta t f(t_{n+1}, U_{n+1}^{(0)}). \tag{5.15}$$

Alternately, the predictor step can be used to give an initial value for a fixed point iteration where iterates are successively calculated by

$$U_{n+1}^{(k+1)} = U_n + \Delta t f(t_{n+1}, U_{n+1}^{(k)}) \quad k = 0, 1, 2, \ldots, \tag{5.16}$$

until $|U_{n+1}^{(k+1)} - U_{n+1}^{(k)}|$ is less than a predefined tolerance.

Another fixed point iteration might be based on Improved Euler (or the trapezoidal rule), so that

$$U_{n+1}^{(k+1)} = U_n + \frac{1}{2}\Delta t \left[f(t_n, U_n) + f(t_{n+1}, U_{n+1}^{(k)}) \right] \quad k = 0, 1, 2, \ldots, \tag{5.17}$$

again, until $|U_{n+1}^{(k+1)} - U_{n+1}^{(k)}|$ is less than a predefined tolerance.

Example: Van der Pol oscillator

A Van der Pol oscillator is an example of a second order system, essentially an oscillator with non-linear damping, which with no forcing is described by

$$\begin{aligned} u'' + \alpha(u^2 - 1)u' + u &= 0 && (5.18)\\ u(0) &= u_0, && (5.19)\\ u'(0) &= v_0, && (5.20) \end{aligned}$$

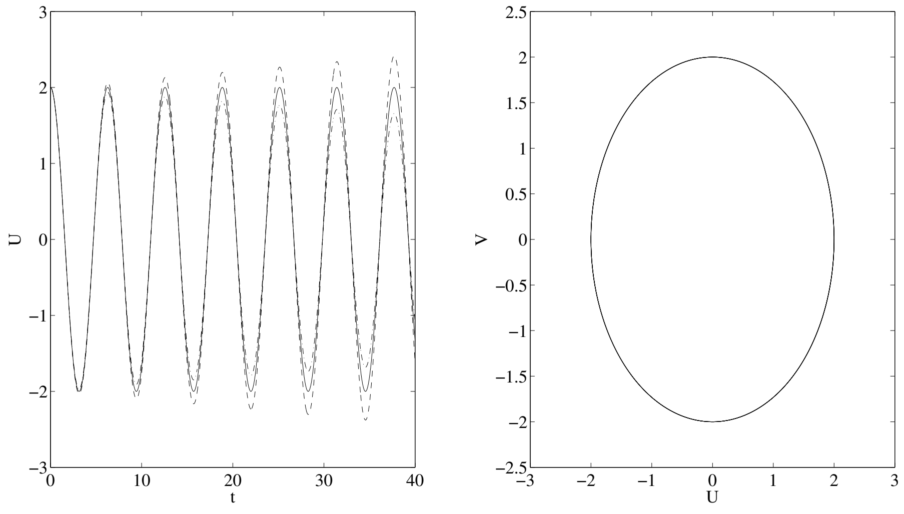
or as a second order system

$$\begin{aligned} u' &= v, & u(0) &= u_0, && (5.21)\\ v' &= -u - \alpha(u^2 - 1)v, & v(0) &= v_0. && (5.22) \end{aligned}$$

This is a modification of the simple spring and mass we considered at the start of this lecture, reducing to the same equations when $\alpha = 0$ and like that system, has a limit cycle behaviour in (u, v) space. In the matlab code *vanderpol.m* we use three algorithms, explicit Euler, improved Euler and implicit Euler using a predictor step to give an initial value for a fixed point iteration. In each case we use the initial conditions $u_0 = 2$, $v_0 = 0$.

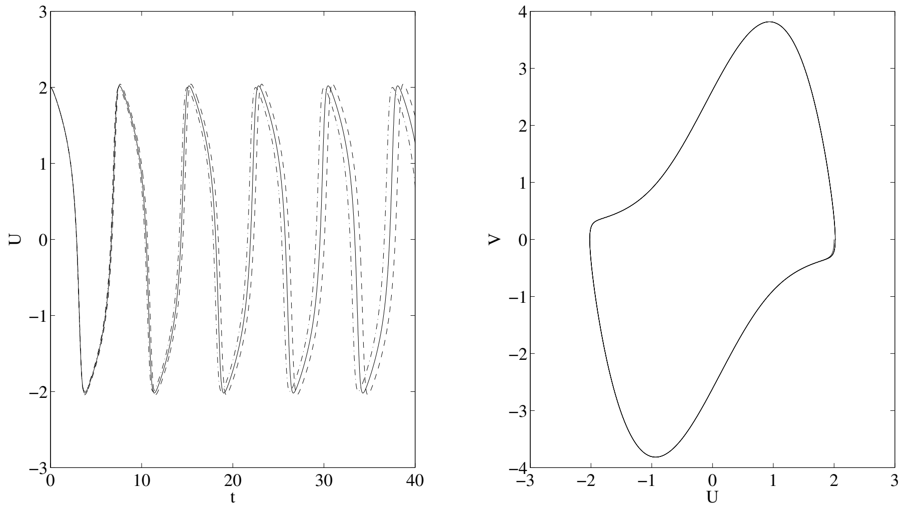
If we set $\alpha = 0$, then a computed solution is illustrated in Figure 4 where as should be expected now, the explicit scheme results in a growing solution, the implicit scheme decays and while Improved Euler appears to conserve the solution correctly, it also has a discrete solution where $u^2 + v^2$ is increasing very slowly.

Setting $\alpha > 0$ brings in the non-linear term, example calculations for $\alpha = 2$ with the same time step, $\Delta t = 0.01$ are shown in Figure 5 where both first order explicit



(a) (b)

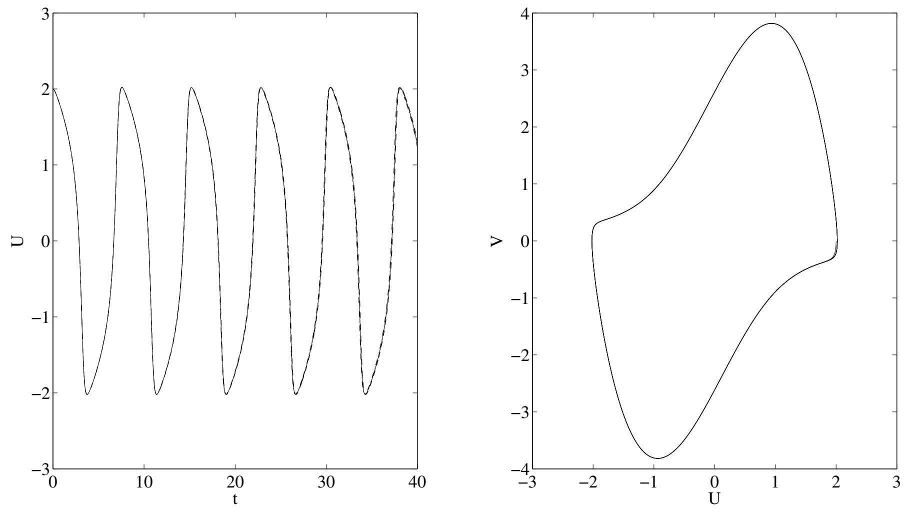
Figure 4: Numerical solution of Van der Pol system when $\alpha = 0$, $\Delta t = 0.01$. (a) Plot of U versus t: (- - -) Explicit Euler, (—) Improved Euler, (- . -) Implicit Euler. (b) Plot of V versus U for Improved Euler



(a) (b)

Figure 5: Numerical solution of Van der Pol system when $\alpha = 2$, $\Delta t = 0.01$. (a) Plot of U versus t: (- - -) Explicit Euler, (—) Improved Euler, (- . -) Implicit Euler. (b) Plot of V versus U for Improved Euler

and implicit Euler are poor, one showing a cycle growing in amplitude and with an increasing period (explicit Euler) and one decreasing in amplitude and with decreasing



(a) (b)

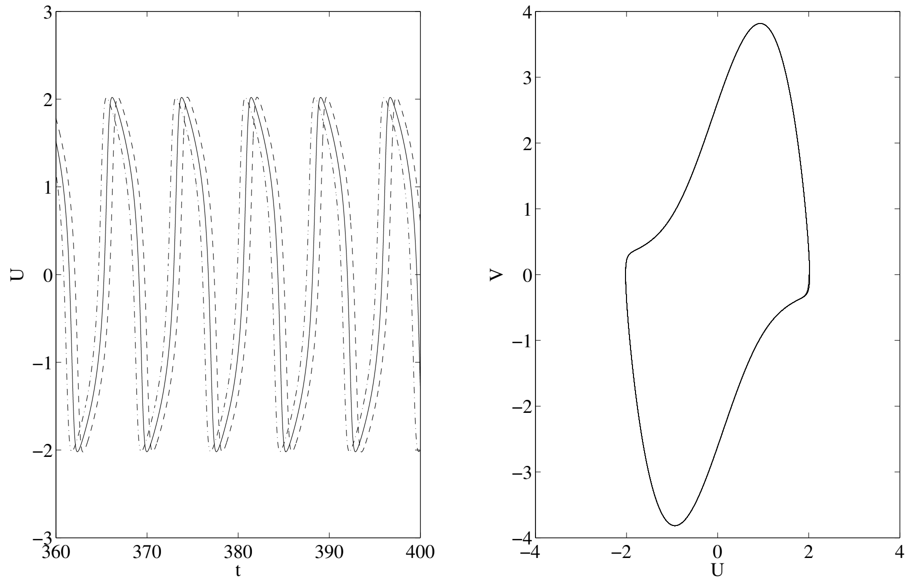
Figure 6: Numerical solution of Van der Pol system when $\alpha = 0$, $\Delta t = 0.001$. (a) Plot of U versus t: (- - -) Explicit Euler, (—) Improved Euler, (- . -) Implicit Euler. (b) Plot of V versus U for Improved Euler

period (implicit Euler). In part (b) of the figure the phase plot of Improved Euler shows that periodicity is being preserved by the method.

Of course, one can improve the performance of a first order scheme by decreasing the time step, and in Figure 6, the solution appears much better.

However, if this integration is carried on to longer itmes, the solution for the two first order methods remains problematic, see Figure 7

It is also worth observing that for the linear system with the two first order discretisations, both theory and the computations show that growth (explicit) or decay (implicit) are exponential, yet in the non-linear system, while the oscillation period shows very slow change, the amplitude does not appear to grow exponentially.



(a) (b)

Figure 7: Numerical solution of Van der Pol system when $\alpha = 0$, $\Delta t = 0.001$ at longer time. (a) Plot of U versus t: (- - -) Explicit Euler, (—) Improved Euler, (- . -) Implicit Euler. (b) Plot of V versus U for Improved Euler

Lecture 6

Symplectic methods

The numerical solution of evolutionary dynamical systems is a major application area for the material we are studying. We can only touch briefly on this large field of research and understanding. Consider a simple mass and spring system, scaled so that the displacement, x, and the velocity, u, are given by

$$x' = u, \; u' = -x. \tag{6.1}$$

As should be well known to you, this can also be set in a Hamiltonian framework, where the Hamiltonian, representing here the sum of potential and kinetic energy,

$$H = \frac{1}{2}(x^2 + u^2),$$

is conserved. There is a geometric view of the solutions, namely that for this case, solutions where the Hamiltonian, H, is constant, are given by circles in (x, u) space and in terms of mapping in the (x, u) space, trajectories are area preserving. So an important question is whether a numerical method preserves these properties.

For this example, let the system be defined by the vector $\mathbf{u}^{\mathrm{T}} = (x, u)$, then the energy of the system can be characterised by $H = \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{u}$. A sensible question is to ask how this quantity evolves using the numerical scheme. If we look at say, explicit Euler, where in matrix form,

$$\begin{pmatrix} X_{n+1} \\ U_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ -\Delta t & 1 \end{pmatrix} \begin{pmatrix} X_n \\ U_n \end{pmatrix},$$

then

$$\begin{pmatrix} X_{n+1} & U_{n+1} \end{pmatrix} \begin{pmatrix} X_{n+1} \\ U_{n+1} \end{pmatrix} = \begin{pmatrix} X_n & U_n \end{pmatrix} \begin{pmatrix} 1 & -\Delta t \\ \Delta t & 1 \end{pmatrix} \begin{pmatrix} 1 & \Delta t \\ -\Delta t & 1 \end{pmatrix} \begin{pmatrix} X_n \\ U_n \end{pmatrix}$$

or

$$H_{n+1} = (1 + \Delta t^2) H_n,$$

so that the energy will grow in time for all practical time steps.

Consider the hybrid Euler scheme

$$\begin{aligned} U_{n+1} &= U_n - \Delta t X_n, & (6.2) \\ X_{n+1} &= X_n + \Delta t U_{n+1}. & (6.3) \end{aligned}$$

In matrix terms

$$\begin{pmatrix} 1 & -\Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X_{n+1} \\ U_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -\Delta t & 1 \end{pmatrix} \begin{pmatrix} X_n \\ U_n \end{pmatrix}.$$

This scheme also does not leave the original Hamiltonian, $H(x, u)$, unaltered, but it does preserve a modified Hamiltonian

$$\hat{H}(x, u) = \frac{1}{2}(x^2 + u^2) - \frac{1}{2}\Delta t x u = H(x, u) - \frac{1}{2}\Delta t x u.$$

This is called a symplectic scheme and it is a general property of symplectic schemes that while not preserving the original Hamiltonian, they do exactly preserve an approximate Hamiltonian, which, in the limit of $\Delta t \to 0$, restores the original Hamiltonian.

A commonly referred to symplectic scheme that is second order is the **Stömer-Verlet** scheme. Suppose the system to be integrated is

$$\begin{aligned} x' &= u && (6.4) \\ u' &= f(x), && (6.5) \end{aligned}$$

for example, derived from a dynamical system $x'' = f(x)$. In the Stömer-Verlet scheme a half step is taken to estimate the velocity at the mid point of the interval, a full step to estimate the displacement at the new time step and then a second half step based on the new displacement updates the velocity, so that

$$\begin{aligned} \hat{U}_{n+1/2} &= U_n + \frac{1}{2}\Delta t f(X_n), && (6.6) \\ X_{n+1} &= X_n + \Delta t \hat{U}_{n+1/2}, && (6.7) \\ U_{n+1} &= \hat{U}_{n+1/2} + \frac{1}{2}\Delta t f(X_{n+1}). && (6.8) \end{aligned}$$

If the intermediate value is eliminated, the scheme is

$$\begin{aligned} X_{n+1} &= X_n + \Delta t U_n + \frac{1}{2}\Delta t^2 f(X_n), && (6.9) \\ U_{n+1} &= U_n + \frac{1}{2}\Delta t[f(X_n) + f(X_{n+1})]. && (6.10) \end{aligned}$$

A further property we can investigate is how area is or is not preserved in dynamical systems. If we start with the example:

$$\begin{aligned} x' &= y, \quad x(0) = x_0, && (6.11) \\ y' &= -x, \quad y(0) = y_0, && (6.12) \end{aligned}$$

with Hamiltonian $H = (x^2 + y^2)/2 = (x_0^2 + y_0^2)/2$ being constant since

$$\frac{\mathrm{d}H}{\mathrm{d}t} = xx' + yy' = 0.$$

Now take a small area in the (x, y) plane, a rectangle with three corners given by (x, y), $(x + dx, y)$, $(x, y + dy)$, and area

$$\mathrm{d}A = \begin{vmatrix} x & y & 1 \\ x+dx & y & 1 \\ x & y+dy & 1 \end{vmatrix} = \begin{vmatrix} x & y & 1 \\ dx & 0 & 0 \\ 0 & dy & 0 \end{vmatrix} = dx * dy,$$

as expected.

Now consider a small parallelagram with three corners (x_1, y_1), (x_2, y_2) and (x_3, y_3), and area

$$A = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

so that

$$\frac{\mathrm{d}A}{\mathrm{d}t} = \begin{vmatrix} x_1' & y_1' & 0 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} + \begin{vmatrix} x_1 & y_1 & 1 \\ x_2' & y_2' & 0 \\ x_3 & y_3 & 1 \end{vmatrix} + \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3' & y_3' & 0 \end{vmatrix}.$$

Hence

$$\frac{\mathrm{d}A}{\mathrm{d}t} = \begin{vmatrix} y_1 & -x_1 & 0 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} + \begin{vmatrix} x_1 & y_1 & 1 \\ y_2 & -x_2 & 0 \\ x_3 & y_3 & 1 \end{vmatrix} + \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ y_3 & -x_3 & 0 \end{vmatrix},$$

and evaluating the determinants and summing,

$$\frac{\mathrm{d}A}{\mathrm{d}t} = 0,$$

and the mapping is area preserving.

Now look at a numerical method, here we consider Euler's method, where in one time step of size Δt, if we take three points (X_1, Y_1), (X_2, Y_2), (X_3, Y_3),

$$X_r \to X_r + \Delta t Y_r, \quad Y_r \to Y_r - \Delta t X_r, \quad r = 1, 2, 3,$$

with area

$$A = \begin{vmatrix} X_1 & Y_1 & 1 \\ X_2 & Y_2 & 1 \\ X_3 & Y_3 & 1 \end{vmatrix}.$$

Algebra can be simplified by choosing points (X_1, Y_1), $(X_1 + h, Y_1)$, $(X_1, Y_1 + k)$ and denote the initial element of area as $A_0 = hk$. In one timestep the vertices change accordiny to Euler's method to give area

$$A_1 = A(\Delta t) = \begin{vmatrix} X_1 + \Delta t Y_1 & Y_1 - \Delta t X_1 & 1 \\ X_1 + h + \Delta t Y_1 & Y_1 - \Delta t(X_1 + h) & 1 \\ X_1 + \Delta t(Y_1 + k) & Y_1 + k - \Delta t X_1 & 1 \end{vmatrix}$$

or

$$A_1 = \begin{vmatrix} X_1 + \Delta t Y_1 & Y_1 - \Delta t X_1 & 1 \\ h & -\Delta t h & 0 \\ \Delta t k & k & 0 \end{vmatrix},$$

and thus

$$A_1 = (1 + \Delta t^2) hk,$$

so that explicit Euler's method will not preserve area and elements of area will increase exponentially.

Showing that a simple simplectic method does preserve area is in problem sheet 2, question 4.

Further reading: see *Simulating Hamiltonian Dynamics* by Benedikt Leimkuhler & Sebastian Reich

Lecture 7

Linear Multistep Methods

(i) Formulation

The integral form (3.16) where we used the interval $[t_n, t_{n+1}]$ as the interval of integration is not the only form possible, for example, instead of integrating over this interval, the integral could be over a larger interval, such as $[t_{n-1}, t_{n+1}]$ with starting value u_{n-1}:

$$u_{n+1} = u_{n-1} + \int_{t_{n-1}}^{t_{n+1}} f(s, u(s))\, \mathrm{d}s, \tag{7.1}$$

and we could use a trapezoidal rule

$$u_{n+1} \approx u_{n-1} + \Delta t \left[f(t_{n-1}, u_{n-1}) + f(t_{n+1}, u_{n+1}) \right], \tag{7.2}$$

leading to an implicit method:

$$U_{n+1} = U_{n-1} + \Delta t \left[f(t_{n-1}, U_{n-1}) + f(t_{n+1}, U_{n+1}) \right]. \tag{7.3}$$

Alternately, Simpson's rule applied to the integral in (7.1) gives,

$$u_{n+1} \approx u_{n-1} + \frac{\Delta t}{3} \left[f(t_{n-1}, u_{n-1}) + 4f(t_n, u_n) + f(t_{n+1}, u_{n+1}) \right] \tag{7.4}$$

giving an implicit method:

$$U_{n+1} = U_{n-1} + \frac{\Delta t}{3} \left[f(t_{n-1}, U_{n-1}) + 4f(t_n, U_n) + f(t_{n+1}, U_{n+1}) \right]. \tag{7.5}$$

Both these algorithms relate the approximation U_{n+1} at t_{n+1} to values at more than one previous time step. This idea leads to a range of methods which are classed as 'multi-step'. We will later see that there are good reasons other than accuracy why some such methods may not be useful and particularly for multi-step methods, we will become concerned with numerical stability in addition to accuracy.

Definition. *Denote $f_n = f(t_n, u_n)$ and $F_n = f(t_n, U_n)$, then a linear k-step method can be written*

$$\sum_{j=0}^{k} \alpha_j U_{n+j} = \Delta t \sum_{j=0}^{k} \beta_j F_{n+j} \quad n = 0, 1, 2, \ldots \tag{7.6}$$

where the coefficients $\alpha_0, \ldots, \alpha_k$ and $\beta_0, \ldots, \beta_k$ are specified real constants with $\alpha_k \neq 0$ (that is the constant values are not dependent on the specific function f being integrated). If $\beta_k = 0$ the method is explicit, if $\beta_k \neq 0$ the method is implicit. For the scheme to be k step we also need one of α_0 and β_0 to be non-zero.

One method to systematically develop multi-step algorithms uses the following approach. Suppose that in addition to being given $U_0 = u_0$, we have by some method calculated U_1. This means we know values $F_0 = f(t_0, U_0)$ and $F_1 = f(t_1, U_1)$. We can then fit a linear polynomial to the values (t_0, F_0) and (t_1, F_1) as shown in Figure 8.

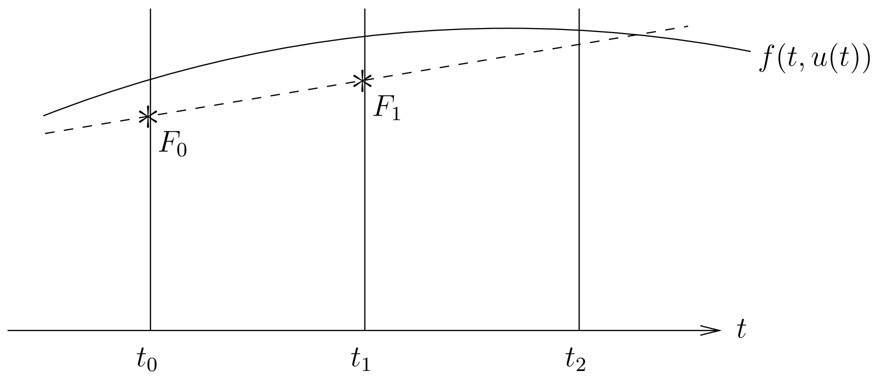


Figure 8: Fitting a linear polynomial to the values (t_0, F_0) and (t_1, F_1).

The linear polynomial

$$F(t) = F_0 + \frac{t - t_0}{t_1 - t_0}(F_1 - F_0), \tag{7.7}$$

can be used to estimate integrals of the function $f(t, u(t))$. The algebra is considerably simplified if the variable t is rescaled by the step length (which we are assuming is constant), let $\eta = (t - t_0)/\Delta t$ so that

$$F(\eta) = (1 - \eta)F_0 + \eta F_1. \tag{7.8}$$

Now we know that

$$u_2 = u_0 + \int_{t_0}^{t_2} f(s, u(s))\,\mathrm{d}s \tag{7.9}$$

so we can approximate

$$\begin{aligned} U_2 &= U_0 + \int_{t_0}^{t_2} F(s)\,\mathrm{d}s & (7.10)\\ &= U_0 + \Delta t \int_0^2 F(\eta)\,\mathrm{d}\eta & (7.11)\\ &= U_0 + \Delta t \int_0^2 \left((1 - \eta)F_0 + \eta F_1\right)\,\mathrm{d}\eta. & (7.12)\end{aligned}$$

This gives

$$U_2 = U_0 + 2\Delta t F_1. \tag{7.13}$$

While we have not specified how the value U_1 might be determined, if we used

$$U_1 = U_0 + \Delta t f(t_0, U_0),$$

then the estimate U_2 will be the same as for modified Euler using a step length $2\Delta t$ but subsequent interates are not the same as for modified Euler.

There are other integral variants we could use together with the linear approximation for $f(s, u(s))$, for example, we could write

$$u_2 = u_1 + \int_{t_1}^{t_2} f(s, u(s))\,\mathrm{d}s \tag{7.14}$$

giving a discrete scheme

$$\begin{aligned} U_2 &= U_1 + \Delta t \int_1^2 F(\eta)\,\mathrm{d}\eta & (7.15)\\ &= U_1 + \Delta t \int_1^2 \left((1-\eta)F_0 + \eta F_1\right)\,\mathrm{d}\eta, & (7.16)\end{aligned}$$

which is

$$U_2 = U_1 + \Delta t \left(\frac{3}{2}F_1 - \frac{1}{2}F_0\right). \tag{7.17}$$

The process of fitting a curve to previous data can be extended using more terms, for example, if we use estimates for U_0, U_1 and U_2 we can fit a quadratic to F; in terms of $\eta = (t - t_0)/\Delta t$,

$$\begin{aligned} F(\eta) &= F_0 + \eta(F_1 - F_0) + \frac{1}{2}\eta(\eta-1)(F_2 - 2F_1 + F_0) & (7.18)\\ &= \left(1 - \frac{3}{2}\eta + \frac{1}{2}\eta^2\right)F_0 + (2\eta - \eta^2)F_1 + \frac{1}{2}\eta(\eta-1)F_2, & (7.19)\end{aligned}$$

and

$$U_3 = U_2 + \Delta t \int_2^3 F(\eta)\,\mathrm{d}\eta, \tag{7.20}$$

gives

$$U_3 = U_2 + \frac{\Delta t}{12}(23F_2 - 16F_1 + 5F_0). \tag{7.21}$$

For all linear multi-step methods, when you determine coefficients β_r, we will see that there are consistency checks that can be made: if the function f is $f = 1$ then the solution should be $U_n = U_0 + n\Delta t$, in this case the left-hand-side of (7.21) would be equal to $3\Delta t$, so putting $F_0 = F_1 = F_2 = 1$ and $U_2 = 2\Delta t$, then the sum of right-hand-side terms should also be equal to $3\Delta t$. We will look at consistency conditions in more detail later.

These are examples of what are generally called Adams methods. Two important fourth order cases are

1. **Adams-Bashforth** (explicit)

$$U_{n+4} = U_{n+3} + \frac{1}{24}\Delta t\left[55F_{n+3} - 59F_{n+2} + 37F_{n+1} - 9F_n\right]. \tag{7.22}$$

2. **Adams-Moulton** (implicit)

$$U_{n+4} = U_{n+3} + \frac{1}{24}\Delta t\left[9F_{n+4} + 19F_{n+3} - 5F_{n+2} + F_{n+1}\right]. \tag{7.23}$$

(ii) Zero Stability

Suppose initial data $U_0, \dots U_{k-1}$ is used successively to generate $U_k, \dots U_n$ and different initial data $\hat{U}_0, \dots \hat{U}_{k-1}$ generates $\hat{U}_k, \dots \hat{U}_n$ using the **same** method for discretising $u' = f(t,u)$. Then the method is zero-stable provided for all functions f, there $\exists K > 0$ such that

$$|U_n - \hat{U}_n| \le K \max\left\{|U_0 - \hat{U}_0|, \dots, |U_{k-1} - \hat{U}_{k-1}|\right\} \tag{7.24}$$

as $\Delta t \to 0$.

Example: Consider the multistep method

$$2U_{n+1} + 3U_n - 6U_{n-1} + U_{n-2} = 6\Delta t f(t_n, U_n). \tag{7.25}$$

This is third order accurate, which can be seen by setting out Taylor expansion in a tableau:

Terms in					
Taylor expansion	1	$\Delta t u'$	$\Delta t^2 u''$	$\Delta t^3 u'''$	$\Delta t^4 u^{\iota v}$
$2U_{n+1}$	2	2	1	$\frac{1}{3}$	$\frac{1}{12}$
$3U_n$	3				
$-6U_{n-1}$	-6	6	-3	1	$-\frac{1}{4}$
U_{n-2}	1	-2	2	$-\frac{4}{3}$	$\frac{2}{3}$
	0	6	0	0	$\frac{1}{2}$

so we see that $T_n = \frac{1}{12}\Delta t^3 u^{\iota v}(\xi_n)$.

However, take data $U_0 = 1 + \epsilon$, $U_1 = 1 - \epsilon$, $U_2 = 1 + \epsilon$, and $f \equiv 0$. Then using the formula

$$U_{n+1} = \frac{1}{2}\left\{-3U_n + 6U_{n-1} - U_{n-2}\right\} \tag{7.26}$$

gives

$$U_3 = 1 - 5\epsilon \tag{7.27}$$

$$U_4 = 1 + 11\epsilon \tag{7.28}$$

$$U_5 = 1 - 32\epsilon \dots \tag{7.29}$$

whereas data $U_0 = 1$, $U_1 = 1$, $U_2 = 1$ gives $U_n = 1$ for $n = 3, 4, \ldots$. This is an example of an instability in the discrete approximation which becomes important as soon as we use finite precision computation.

To understand what has happened and how we can analyse multistep methods we go back to the discrete method with $f = 0$

$$2U_{n+1} + 3U_n - 6U_{n-1} + U_{n-2} \;=\; 0. \tag{7.30}$$

This is a difference equation, we try solutions $U_n \sim k\lambda^n$ to get

$$k(2\lambda^3 + 3\lambda^2 - 6\lambda + 1) \;=\; 0. \tag{7.31}$$

We assume $k \neq 0$ so that λ should be a root of

$$2\lambda^3 + 3\lambda^2 - 6\lambda + 1 \;=\; 0. \tag{7.32}$$

Hence

$$(\lambda - 1)(2\lambda^2 + 5\lambda - 1) \;=\; 0 \tag{7.33}$$

and

$$\lambda = 1, \quad \lambda = -\frac{1}{4}(5 \pm \sqrt{33}) = -2.69,\ 0.19. \tag{7.34}$$

This means that the general solution of the difference equation is

$$U_n \;=\; k_1 1^n + k_2(0.19)^n + k_3(-2.69)^n, \tag{7.35}$$

where k_1, k_2, k_3 will come from the initial data. When $k_3 \neq 0$, the values computed for U_n will grow exponentially ($2.69^n = \mathrm{e}^{n \log 2.69} = \mathrm{e}^{0.99n}$). Finite precision arithmetic means there will always be a component in the initial data that has $k_3 \neq 0$, even if only at the level of machine precison. For most double precision arithmetic using 64bit arithmetic, machine precision is around 10^{-15} so in that case the growing last term will be $\mathcal{O}(1)$ after $(2.69)^n \times 10^{-15} \sim 1$, around only 16 iterations.

Lecture 8

Definition. *For a linear multistep method*

$$\sum_{j=0}^{k} \alpha_j U_{n+j} \quad = \quad \Delta t \sum_{j=0}^{k} \beta_j F_{n+j} \quad n = 0, 1, 2, \ldots \tag{8.1}$$

define two characteristic polynomials

$$\rho(z) \quad = \quad \sum_{j=0}^{k} \alpha_j z^j \quad \textit{First characteristic polynomial} \tag{8.2}$$

$$\sigma(z) \quad = \quad \sum_{j=0}^{k} \beta_j z^j \quad \textit{Second characteristic polynomial.} \tag{8.3}$$

Theorem. *A linear multistep method is zero stable if and only if the roots of ρ lie in the closed unit disc with any on the unit circle being simple.*

The condition on the roots of ρ in this theorem is called the *root condition* .

Proof. The necessary part follows the same line that we saw in the example above, if a root has magnitude greater than one then there will be an exponentially growing part of any solution. The need for roots on circle to be simple is best understood by recalling that for a linear ODE with constant coefficients, for example $u'' - 2u' + u = 0$ which has a double root in characteristic equation, the solutions are e^t and $t\mathrm{e}^t$, similarly for a difference equation $U_{n+1} - 2U_n + U_{n-1} = 0$ in addition to $U_n = 1$ being a solution, $U_n = n$ is also a solution which will grow errors in the original data (multiplicity r implies $1, n, n^2, \ldots, n^{r-1}$ as solutions).

Sufficiency is a long and complicated proof and so is regarded as outside of the course, nevertheless we can look briefly at the essence of this part of the proof of the theorem.

Let $\mathbf{U}^{(n)} = [U_n, \ldots, U_{n+k-1}]^T$.

For $f = 0$, define matrix A by $\mathbf{U}^{(n+1)} = A\mathbf{U}^{(n)}$. A is the $k \times k$ matrix

$$A \quad = \quad \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & & & & \\ 0 & \ldots & \ldots & 0 & 1 \\ -\frac{\alpha_0}{\alpha_k} & -\frac{\alpha_1}{\alpha_k} & \ldots & \ldots & -\frac{\alpha_{k-1}}{\alpha_k} \end{pmatrix}, \tag{8.4}$$

so that in this particular case where $f = 0$, $\mathbf{U}^{(n)} = A^n \mathbf{U}^{(0)}$. Note also that because of the unit values on the first $k-1$ rows, we must have $\|A\|_\infty \geq 1$.

Lemma 1. *Root condition and* $f = 0 \Rightarrow \|\mathbf{U}^{(n)}\|_\infty \leq K\|\mathbf{U}^{(0)}\|_\infty$ *for some* K *so that*

$$\|A^n\|_\infty \leq K \quad \forall n. \tag{8.5}$$

Note here that because of $\|A\|_\infty \geq 1$, we must have $K \geq 1$. This is critical in deriving the next lemma.

Lemma 2. *For* A *as defined and* $k \times k$ *matrices* D_m, $m = 0, \ldots, n-1$ *all with* $\|D_m\|_\infty \leq K_1$

$$\|\prod_{m=0}^{n-1}(A + \Delta t D_m)\|_\infty \quad \leq \quad K\mathrm{e}^{KK_1 n\Delta t}. \tag{8.6}$$

This lemma is shown by expanding the product, using triangle inequalites and then inequalities such as $\|A^{n-m}D_mA^{m-1}\|_\infty \leq \|A^{n-m}\|_\infty\|D_m\|_\infty\|A^{m-1}\|_\infty \leq KK_1K$ and $K \geq 1$ to show that

$$\|\prod_{m=0}^{n-1}(A + \Delta t D_m)\|_\infty \quad \leq \quad K\prod_{m=0}^{n-1}(1 + \Delta t KK_1) = K(1 + \Delta t KK_1)^n, \tag{8.7}$$

from which the result follows by the comparison $1 + x \leq \exp x$.

Having established the inequality in the second lemma, the outline proof follows as:

Let $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ be two sequences from different starting values, use the Lipschitz condition on f to write

$$\mathbf{U}^{(n+1)} - \mathbf{V}^{(n+1)} \quad = \quad (A + \Delta t D_n)\left(\mathbf{U}^{(n)} - \mathbf{V}^{(n)}\right) \tag{8.8}$$

$$= \quad \left[\prod_{m=0}^{n}(A + \Delta t D_m)\right]\left(\mathbf{U}^{(0)} - \mathbf{V}^{(0)}\right). \tag{8.9}$$

So using Lemma 2 and 1

$$\|\mathbf{U}^{(n+1)} - \mathbf{V}^{(n+1)}\|_\infty \quad \leq \quad K\mathrm{e}^{KK_1\Delta t(n+1)}\|\mathbf{U}^{(0)} - \mathbf{V}^{(0)}\|_\infty \tag{8.10}$$

and because $n\Delta t$ is bounded by the interval of integration, the constant in this equation must also be bounded and thus the root condition implies zero stability. $\square$

Examples

1. Explicit Euler and implicit Euler are zero stable as $\rho(z) = z - 1$.

2. Adams Bashforth written in the form
$$U_{n+4} - U_{n+3} = \frac{\Delta t}{24}\left[55F_{n+3} - 59F_{n+2} + 37F_{n+1} - 9F_n\right] \tag{8.11}$$
has $\rho(z) = z^3(z-1)$ so satisfies the root condition and the method is zero stable.
3. Third order example
$$2U_{n+3} + 3U_{n+2} - 6U_{n+1} + U_n = 6\Delta t F_{n+2} \tag{8.12}$$
has
$$\begin{aligned}\rho(z) &= 2z^3 + 3z^2 - 6z + 1 && (8.13)\\ &= (z-1)(z+2.69)(z-0.19) && (8.14)\end{aligned}$$
which has a root outside the unit disc and so the method is not zero stable.

To reiterate, we refer to conditions on ρ that roots lie in closed unit disc and with simple roots on unit circle as the *root condition*.

(iii) Consistency

For a linear k step method we must rearrange the scheme so it is an analogue for the continuous system $u' = f$; doing this defines the truncation error as
$$T_n = \frac{\sum_{j=0}^{k} \alpha_j u_{n+j} - \Delta t \sum_{j=0}^{k} \beta_j f_{n+j}}{\Delta t \sum_{j=0}^{k} \beta_j} \tag{8.15}$$
where we need to have
$$\sum_{j=0}^{k} \beta_j = \sigma(1) \neq 0. \tag{8.16}$$
A method will be consistent provided $T_n \to 0$ as $\Delta t \to 0$. To show this, expand the right-hand-side of (8.15) as $\Delta t \to 0$ using Taylor series.
$$\begin{aligned}T_n &= \frac{\sum_{j=0}^{k} \alpha_j u_{n+j} - \Delta t \sum_{j=0}^{k} \beta_j u'_{n+j}}{\Delta t \sum_{j=0}^{k} \beta_j} && (8.17)\\ &= \frac{1}{\Delta t \sigma(1)}\left[C_0 u_n + C_1 \Delta t u'_n + C_2 \Delta t^2 u''_n + \ldots + C_p \Delta t^p u_n^{(p)} + \ldots\right]. && (8.18)\end{aligned}$$
For the scheme to be consistent we need $T_n \to 0$ as $\Delta t \to 0$, so that we will require
$$\begin{aligned}C_0 &= 0 && (8.19)\\ C_1 &= 0. && (8.20)\end{aligned}$$

To determine the coefficients C_p in (8.18) we have to apply Taylor expansions to the various terms:

$$u_{n+j} = u_n + j\Delta t u_n' + \frac{(j\Delta t)^2}{2!}u_n'' + \quad \dots \quad + \frac{(j\Delta t)^p}{p!}u_n^{(p)} + \dots \tag{8.21}$$

$$u_{n+j}' = u_n' + \frac{j\Delta t}{1!}u_n'' + \frac{(j\Delta t)^2}{2!}u_n''' + \dots + \frac{(j\Delta t)^{p-1}}{(p-1)!}u_n^{(p)} + \dots \tag{8.22}$$

Hence

$$C_0 = \sum_{j=0}^{k} \alpha_j = 0 \tag{8.23}$$

$$C_1 = \sum_{j=0}^{k} j\alpha_j - \sum_{j=0}^{k} \beta_j = \rho'(1) - \sigma(1) = 0, \tag{8.24}$$

and consistency requires

$$\rho(1) = 0, \quad \rho'(1) = \sigma(1) \neq 0 \tag{8.25}$$

(and as $\rho'(1) \neq 0$ the root at $z = 1$ must be simple).

The general form for $p \geq 2$ is

$$C_p = \sum_{j=0}^{k} \frac{j^p}{p!}\alpha_j - \sum_{j=0}^{k} \frac{j^{p-1}}{(p-1)!}\beta_j \tag{8.26}$$

$$= \sum_{j=1}^{k} \frac{j^p}{p!}\alpha_j - \sum_{j=1}^{k} \frac{j^{p-1}}{(p-1)!}\beta_j. \tag{8.27}$$

We suppose $C_0 = C_1 = \dots = C_p = 0$, with $C_{p+1} \neq 0$ so that

$$T_n = \frac{C_{p+1}}{\sigma(1)}\Delta t^p u_n^{(p+1)} + \mathcal{O}(\Delta t^{p+1}) \tag{8.28}$$

and the method is order p. The value $C_{p+1}/\sigma(1)$ is called the error constant of the method.

Example: wlog set $\alpha_2 = 1$ and consider the method

$$U_{n+2} + \alpha_1 U_{n+1} + \alpha_0 U_n = \Delta t(\beta_2 F_{n+2} + \beta_1 F_{n+1} + \beta_0 F_n) \tag{8.29}$$

with first and second characteristic polynomials

$$\rho(z) = z^2 + \alpha_1 z + \alpha_0 \tag{8.30}$$

$$\sigma(z) = \beta_2 z^2 + \beta_1 z + \beta_0. \tag{8.31}$$

There are five unknowns α_1, α_0, β_2, β_1 and β_0 so if we set $C_0 = C_1 = C_2 = C_3 = 0$ there will be a one parameter family of third order methods. The equations are

- $p = 0, \qquad 1 + \alpha_1 + \alpha_0 = 0$
- $p = 1, \qquad 2 + \alpha_1 = \beta_2 + \beta_1 + \beta_0$
- $p = 2, \qquad \frac{2^2}{2!} + \frac{1^2}{2!}\alpha_1 = 2\beta_2 + \beta_1$
- $p = 3, \qquad \frac{2^3}{3!} + \frac{1^3}{3!}\alpha_1 = \frac{2^2}{2!}\beta_2 + \frac{1^2}{2!}\beta_1$

with solution

$$\alpha_1 = -(1+\alpha_0) \quad \beta_2 = \frac{5+\alpha_0}{12} \quad \beta_1 = \frac{2-2\alpha_0}{3} \quad \beta_2 = -\frac{1+5\alpha_0}{12}. \tag{8.32}$$

The first characteristic polynomial is $z^2 - (1+\alpha_0)z + \alpha_0 = 0$ so

$$\rho(z) \ = \ (z-\alpha_0)(z-1) \ = \ 0 \tag{8.33}$$

and so scheme is zero stable provided $-1 \le \alpha_0 < 1$. If we look at the term where $p = 4$ then

$$C_4 \ = \ -\frac{1+\alpha_0}{24} \ \neq \ 0 \text{ if } \alpha_0 \neq -1. \tag{8.34}$$

In the specific case of $\alpha_0 = -1$ this will give a fourth order scheme:

$$U_{n+2} - U_n \ = \ \frac{\Delta t}{3}\left(F_{n+2} + 4F_{n+1} + F_n\right) \tag{8.35}$$

which is Simpson's rule.

Lecture 8

Definition. *For a linear multistep method*

$$\sum_{j=0}^{k} \alpha_j U_{n+j} = \Delta t \sum_{j=0}^{k} \beta_j F_{n+j} \quad n = 0, 1, 2, \ldots \tag{8.1}$$

define two characteristic polynomials

$$\rho(z) = \sum_{j=0}^{k} \alpha_j z^j \quad \textit{First characteristic polynomial} \tag{8.2}$$

$$\sigma(z) = \sum_{j=0}^{k} \beta_j z^j \quad \textit{Second characteristic polynomial.} \tag{8.3}$$

Theorem. *A linear multistep method is zero stable if and only if the roots of ρ lie in the closed unit disc with any on the unit circle being simple.*

The condition on the roots of ρ in this theorem is called the *root condition* .

Proof. The necessary part follows the same line that we saw in the example above, if a root has magnitude greater than one then there will be an exponentially growing part of any solution. The need for roots on circle to be simple is best understood by recalling that for a linear ODE with constant coefficients, for example $u'' - 2u' + u = 0$ which has a double root in characteristic equation, the solutions are e^t and $t\mathrm{e}^t$, similarly for a difference equation $U_{n+1} - 2U_n + U_{n-1} = 0$ in addition to $U_n = 1$ being a solution, $U_n = n$ is also a solution which will grow errors in the original data (multiplicity r implies $1, n, n^2, \ldots, n^{r-1}$ as solutions).

Sufficiency is a long and complicated proof and so is regarded as outside of the course, nevertheless we can look briefly at the essence of this part of the proof of the theorem.

Let $\mathbf{U}^{(n)} = [U_n, \ldots, U_{n+k-1}]^T$.

For $f = 0$, define matrix A by $\mathbf{U}^{(n+1)} = A\mathbf{U}^{(n)}$. A is the $k \times k$ matrix

$$A = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & & & & \\ 0 & \ldots & \ldots & 0 & 1 \\ -\frac{\alpha_0}{\alpha_k} & -\frac{\alpha_1}{\alpha_k} & \ldots & \ldots & -\frac{\alpha_{k-1}}{\alpha_k} \end{pmatrix}, \tag{8.4}$$

so that in this particular case where $f = 0$, $\mathbf{U}^{(n)} = A^n\mathbf{U}^{(0)}$. Note also that because of the unit values on the first $k-1$ rows, we must have $\|A\|_\infty \geq 1$.

Lemma 1. *Root condition and $f = 0 \Rightarrow \|\mathbf{U}^{(n)}\|_\infty \leq K\|\mathbf{U}^{(0)}\|_\infty$ for some K so that*

$$\|A^n\|_\infty \leq K \quad \forall n. \tag{8.5}$$

Note here that because of $\|A\|_\infty \geq 1$, we must have $K \geq 1$. This is critical in deriving the next lemma.

Lemma 2. *For A as defined and $k \times k$ matrices D_m, $m = 0, \ldots, n-1$ all with $\|D_m\|_\infty \leq K_1$*

$$\| \prod_{m=0}^{n-1} (A + \Delta t D_m)\|_\infty \;\; \leq \;\; K\mathrm{e}^{KK_1 n\Delta t}. \tag{8.6}$$

This lemma is shown by expanding the product, using triangle inequalites and then inequalities such as $\|A^{n-m}D_mA^{m-1}\|_\infty \leq \|A^{n-m}\|_\infty\|D_m\|_\infty\|A^{m-1}\|_\infty \leq KK_1K$ and $K \geq 1$ to show that

$$\| \prod_{m=0}^{n-1} (A + \Delta t D_m)\|_\infty \;\; \leq \;\; K \prod_{m=0}^{n-1} (1 + \Delta t K K_1) = K(1 + \Delta t K K_1)^n, \tag{8.7}$$

from which the result follows by the comparison $1 + x \leq \exp x$.

Having established the inequality in the second lemma, the outline proof follows as:

Let $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ be two sequences from different starting values, use the Lipschitz condition on f to write

$$\mathbf{U}^{(n+1)} - \mathbf{V}^{(n+1)} \;\; = \;\; (A + \Delta t D_n)\left(\mathbf{U}^{(n)} - \mathbf{V}^{(n)}\right) \tag{8.8}$$

$$= \;\; \left[\prod_{m=0}^{n} (A + \Delta t D_m)\right]\left(\mathbf{U}^{(0)} - \mathbf{V}^{(0)}\right). \tag{8.9}$$

So using Lemma 2 and 1

$$\|\mathbf{U}^{(n+1)} - \mathbf{V}^{(n+1)}\|_\infty \;\; \leq \;\; K\mathrm{e}^{KK_1\Delta t(n+1)}\|\mathbf{U}^{(0)} - \mathbf{V}^{(0)}\|_\infty \tag{8.10}$$

and because $n\Delta t$ is bounded by the interval of integration, the constant in this equation must also be bounded and thus the root condition implies zero stability. □

Examples

1. Explicit Euler and implicit Euler are zero stable as $\rho(z) = z - 1$.

2. Adams Bashforth written in the form

$$U_{n+4} - U_{n+3} = \frac{\Delta t}{24}\left[55F_{n+3} - 59F_{n+2} + 37F_{n+1} - 9F_n\right] \tag{8.11}$$

has $\rho(z) = z^3(z-1)$ so satisfies the root condition and the method is zero stable.

3. Third order example

$$2U_{n+3} + 3U_{n+2} - 6U_{n+1} + U_n = 6\Delta t F_{n+2} \tag{8.12}$$

has

$$\begin{aligned}\rho(z) &= 2z^3 + 3z^2 - 6z + 1 &(8.13)\\ &= (z-1)(z+2.69)(z-0.19) &(8.14)\end{aligned}$$

which has a root outside the unit disc and so the method is not zero stable.

To reiterate, we refer to conditions on ρ that roots lie in closed unit disc and with simple roots on unit circle as the *root condition*.

(iii) Consistency

For a linear k step method we must rearrange the scheme so it is an analogue for the continuous system $u' = f$; doing this defines the truncation error as

$$T_n = \frac{\sum_{j=0}^{k} \alpha_j u_{n+j} - \Delta t \sum_{j=0}^{k} \beta_j f_{n+j}}{\Delta t \sum_{j=0}^{k} \beta_j} \tag{8.15}$$

where we need to have

$$\sum_{j=0}^{k} \beta_j = \sigma(1) \neq 0. \tag{8.16}$$

A method will be consistent provided $T_n \to 0$ as $\Delta t \to 0$. To show this, expand the right-hand-side of (8.15) as $\Delta t \to 0$ using Taylor series.

$$\begin{aligned}T_n &= \frac{\sum_{j=0}^{k} \alpha_j u_{n+j} - \Delta t \sum_{j=0}^{k} \beta_j u'_{n+j}}{\Delta t \sum_{j=0}^{k} \beta_j} &(8.17)\\ &= \frac{1}{\Delta t \sigma(1)}\left[C_0 u_n + C_1 \Delta t u'_n + C_2 \Delta t^2 u''_n + \ldots + C_p \Delta t^p u_n^{(p)} + \ldots\right]. &(8.18)\end{aligned}$$

For the scheme to be consistent we need $T_n \to 0$ as $\Delta t \to 0$, so that we will require

$$\begin{aligned}C_0 &= 0 &(8.19)\\ C_1 &= 0. &(8.20)\end{aligned}$$

To determine the coefficients C_p in (8.18) we have to apply Taylor expansions to the various terms:

$$
\begin{aligned}
u_{n+j} &= u_n + j\Delta t u'_n + \frac{(j\Delta t)^2}{2!}u''_n + \quad \dots \quad + \frac{(j\Delta t)^p}{p!}u_n^{(p)} + \dots && (8.21)\\
u'_{n+j} &= \qquad u'_n + \frac{j\Delta t}{1!}u''_n + \frac{(j\Delta t)^2}{2!}u'''_n + \dots + \frac{(j\Delta t)^{p-1}}{(p-1)!}u_n^{(p)} + \dots && (8.22)
\end{aligned}
$$

Hence

$$
\begin{aligned}
C_0 &= \sum_{j=0}^{k} \alpha_j = 0 && (8.23)\\
C_1 &= \sum_{j=0}^{k} j\alpha_j - \sum_{j=0}^{k} \beta_j = \rho'(1) - \sigma(1) = 0, && (8.24)
\end{aligned}
$$

and consistency requires

$$\rho(1) = 0, \quad \rho'(1) = \sigma(1) \neq 0 \qquad (8.25)$$

(and as $\rho'(1) \neq 0$ the root at $z = 1$ must be simple).

The general form for $p \geq 2$ is

$$
\begin{aligned}
C_p &= \sum_{j=0}^{k} \frac{j^p}{p!}\alpha_j - \sum_{j=0}^{k} \frac{j^{p-1}}{(p-1)!}\beta_j && (8.26)\\
&= \sum_{j=1}^{k} \frac{j^p}{p!}\alpha_j - \sum_{j=1}^{k} \frac{j^{p-1}}{(p-1)!}\beta_j. && (8.27)
\end{aligned}
$$

We suppose $C_0 = C_1 = \dots = C_p = 0$, with $C_{p+1} \neq 0$ so that

$$T_n = \frac{C_{p+1}}{\sigma(1)}\Delta t^p u_n^{(p+1)} + \mathcal{O}(\Delta t^{p+1}) \qquad (8.28)$$

and the method is order p. The value $C_{p+1}/\sigma(1)$ is called the error constant of the method.

Example: wlog set $\alpha_2 = 1$ and consider the method

$$U_{n+2} + \alpha_1 U_{n+1} + \alpha_0 U_n = \Delta t(\beta_2 F_{n+2} + \beta_1 F_{n+1} + \beta_0 F_n) \qquad (8.29)$$

with first and second characteristic polynomials

$$
\begin{aligned}
\rho(z) &= z^2 + \alpha_1 z + \alpha_0 && (8.30)\\
\sigma(z) &= \beta_2 z^2 + \beta_1 z + \beta_0. && (8.31)
\end{aligned}
$$

There are five unknowns α_1, α_0, β_2, β_1 and β_0 so if we set $C_0 = C_1 = C_2 = C_3 = 0$ there will be a one parameter family of third order methods. The equations are

- $p = 0, \qquad 1 + \alpha_1 + \alpha_0 = 0$
- $p = 1, \qquad 2 + \alpha_1 = \beta_2 + \beta_1 + \beta_0$
- $p = 2, \qquad \frac{2^2}{2!} + \frac{1^2}{2!}\alpha_1 = 2\beta_2 + \beta_1$
- $p = 3, \qquad \frac{2^3}{3!} + \frac{1^3}{3!}\alpha_1 = \frac{2^2}{2!}\beta_2 + \frac{1^2}{2!}\beta_1$

with solution

$$\alpha_1 = -(1+\alpha_0) \quad \beta_2 = \frac{5+\alpha_0}{12} \quad \beta_1 = \frac{2-2\alpha_0}{3} \quad \beta_2 = -\frac{1+5\alpha_0}{12}. \tag{8.32}$$

The first characteristic polynomial is $z^2 - (1+\alpha_0)z + \alpha_0 = 0$ so

$$\rho(z) \;=\; (z-\alpha_0)(z-1) \;=\; 0 \tag{8.33}$$

and so scheme is zero stable provided $-1 \leq \alpha_0 < 1$. If we look at the term where $p = 4$ then

$$C_4 \;=\; -\frac{1+\alpha_0}{24} \;\neq\; 0 \text{ if } \alpha_0 \neq -1. \tag{8.34}$$

In the specific case of $\alpha_0 = -1$ this will give a fourth order scheme:

$$U_{n+2} - U_n \;=\; \frac{\Delta t}{3}\left(F_{n+2} + 4F_{n+1} + F_n\right) \tag{8.35}$$

which is Simpson's rule.

Lecture 9

(iv) Convergence

Convergence is a very important property that allows us to be sure that, provided we take sufficiently small time steps, we will obtain better and better approximations to the true solutions of the differential system. This does not refer to how 'good' our approximation is to the true value, only that the more effort we apply, the better will be our approximation (but not how much better).

For a scheme to converge we need

$$U_n \quad \to \quad u(t) \quad \text{as } \Delta t \to 0 \text{ and } n\Delta t \to t, \tag{9.1}$$

and in particular we can define consistent starting conditions by the requirement that $U_j \to u_0$ as $\Delta t \to 0$ for $j = 0, \ldots, k-1$.

Theorem. *Zero stability is a necessary condition for convergence.*

Proof. Suppose a linear multistep method is convergent for all functions f.

Since there has to be convergence for all f, we can in particular choose $f \equiv 0$ and the method must converge.

Consider this case, $u' = 0$, $u(0) = 0$ with solution $u = 0$.

The discrete scheme reduces to:

$$\alpha_k U_{n+k} + \ldots + \alpha_0 U_n \quad = \quad 0. \tag{9.2}$$

As the method is convergent, $U_n \to 0$ as $\Delta t \to 0$, $n\Delta t \to t$, for consistent starting values $U_0, \ldots, U_{k-1}$.

Now suppose $z = re^{i\theta}$ is a root of $\rho(z) = 0$ and choose $U_n = \Delta t r^n \cos(n\theta)$. This set is consistent with the initial value $u(0) = 0$, since $U_0, \ldots U_{k-1} \to 0 = u(0)$ as $\Delta t \to 0$. The set also satisfies (9.2) since by writing

$$U_n = \text{Re}(\Delta t r^n \text{e}^{in\theta}) = \Delta t \text{Re}(z^n)$$

then

$$\alpha_k U_{n+k} + \ldots + \alpha_0 U_n = \Delta t \text{Re}(\alpha_k z^{n+k} + \ldots + \alpha_0 z^n) \tag{9.3}$$

and so

$$\alpha_k U_{n+k} + \ldots + \alpha_0 U_n = \Delta t \text{Re}(z^n \rho(z)) \; = \; 0, \tag{9.4}$$

as z is a root of ρ. Hence we know that the values $U_n = \Delta t r^n \cos(n\theta)$ are solutions of the discrete method starting from a consistent set of intial values.

1. Suppose $\theta \neq 0, \pi$, then

$$U_n^2 - U_{n+1}U_{n-1} = \Delta t^2 r^{2n} \left[\cos^2(n\theta) - \cos((n+1)\theta)\cos((n-1)\theta)\right] \quad (9.5)$$

   so that

$$U_n^2 - U_{n+1}U_{n-1} = \Delta t^2 r^{2n} \sin^2\theta. \quad (9.6)$$

   The LHS of this equation must tend to zero as $n \to \infty$, $n\Delta t \to t > 0$ as all the values must converge to the solutions of the continuous system. Thus the RHS of this equation must also tend to zero. However, since $n\Delta t \to t > 0$ and $\sin\theta \neq 0$, that will only be possible provided $r \leq 1$; if $r > 1$ then the RHS would become unbounded.

2. If $\theta = 0, \pi$ let

$$U_n = \begin{cases} \Delta t r^n & \theta = 0 \\ \Delta t r^n (-1)^n & \theta = \pi \end{cases} \quad (9.7)$$

   or equivalently

$$U_n = \begin{cases} (n\Delta t)\frac{r^n}{n} & \theta = 0 \\ (n\Delta t)\frac{r^n(-1)^n}{n} & \theta = \pi \end{cases} \quad (9.8)$$

   and again we will have that since $U_n \to 0$, we must also have $r \leq 1$.

If $z = re^{i\theta}$ is a double root of $\rho(z)$ then $U_n = \Delta t n r^n \cos(n\theta)$ also satisfies (9.2) and now we deduce that we need $r < 1$ for $U_n \to 0$, so any root on unit circle must be simple (higher order roots follow same argument, if root of order s consider $U_n = \Delta t n^m r^n \cos(n\theta)$ for $m = 0, \ldots, s-1$).

Hence, for a discrete scheme that is convergent for all functions f, the roots of the first characteristic polynomial ρ must satisfy root condition, and so the scheme must be zero stable. □

Theorem. *Consistency is a necessary condition for convergence.*

Proof. We use the same methods here as in the last proof, we assume a linear multistep method is convergent for all functions f and then consider two special cases.

1. Use the function $f \equiv 0$ with inital value $u(0) = 1$, so that $u' = 0$, with solution $u(t) = 1$.

   Let $U_0 = U_1 = \ldots = U_{k-1} = 1$ be a consistent set of initial data. Convergence gives $U_n \to 1$ as $n\Delta t \to t$, $\Delta t \to 0$ but method is

$$\alpha_k U_{n+k} + \alpha_{k-1}U_{n+k-1} + \ldots + \alpha_0 U_n = 0 \quad (9.9)$$

so in limit $U_n, U_{n+1}, \ldots, U_{n+k} \to 1$ we must have

$$\alpha_k + \alpha_{k-1} + \ldots + \alpha_0 = 0$$

and hence $\rho(1) = 0$.

2. Next use the function $f \equiv 1$ with initial value $u(0) = 0$ so that $u' = 1$, with solution $u(t) = t$ and convergence will guarantee that $U_n \to t$ as $n\Delta t \to t$, $\Delta t \to 0$.

   When $f \equiv 1$ the method is

   $$\alpha_k U_{n+k} + \alpha_{k-1} U_{n+k-1} + \ldots + \alpha_0 U_n \quad = \quad \Delta t(\beta_k + \ldots + \beta_0). \qquad (9.10)$$

   Convergence to the solution $u(t) = t$ means that when we consider the limit $n \to \infty$, and $n\Delta t \to t$, we can replace the values U_{n+r}, $r = 0, 1, \ldots, k$ in this scheme with $U_{n+r} = (n+r)\Delta t$ so that it must be the case that

   $$\alpha_k(n+k)\Delta t + \ldots + \alpha_0 n\Delta t \quad = \quad \Delta t(\beta_k + \ldots + \beta_0) \qquad (9.11)$$

   so using $\alpha_k + \alpha_{k-1} + \ldots + \alpha_0 = 0$ we are left with

   $$\underbrace{k\alpha_k + \ldots + \alpha_1}_{\rho'(1)} \quad = \quad \underbrace{\beta_k + \ldots + \beta_0}_{\sigma(1)} \qquad (9.12)$$

   It is also true that $U_0, \ldots, U_{k-1}$ are consistent initial data. Hence convergence implies

   $$\rho(1) \quad = \quad 0, \quad \rho'(1) \quad = \quad \sigma(1), \qquad (9.13)$$

   that is, the scheme will be consistent. (We have already shown that convergence implies zero stability so that $\rho'(1) \neq 0$ as the root at $z = 1$ must be simple.)

□

Theorem (Dahlquist). *Suppose the linear multistep method*

$$\sum_{j=0}^{k} \alpha_j U_{n+j} = \Delta t \sum_{j=0}^{k} \beta_j F_{n+j} \qquad (9.14)$$

is consistent with the ODE $u' = f(t,u)$ and that f satisfies a Lipschitz condition. Then provided initial data $U_0, \ldots, U_{k-1}$ is consistent as $\Delta t \to 0$, zero stability is necessary and sufficient for convergence. If $u(t) \in C^{p+1}$ and truncation error is order $(\Delta t)^p$ then global error is order $(\Delta t)^p$ provided initial data error also at least as small.

Proof. (Outline for the explicit method)

We have already shown that convergence ⇒ zero stability and consistency. From the definition of the scheme and the truncation error we have

$$\begin{aligned}\alpha_k U_{n+k} + \ldots + \alpha_0 U_n &= \Delta t(\beta_{k-1}F_{n+k-1} + \ldots + \beta_0 F_n) && (9.15)\\ \alpha_k u_{n+k} + \ldots + \alpha_0 u_n &= \Delta t(\beta_{k-1}f_{n+k-1} + \ldots + \beta_0 f_n) + \sigma(1)\Delta t T_n. && (9.16)\end{aligned}$$

Subtract and re-arrange to that

$$e_{n+k} = -\sum_{j=0}^{k}\frac{\alpha_j}{\alpha_k}e_{n+j} - \Delta t\sum_{j=0}^{k-1}\frac{\beta_j}{\alpha_k}(f_{n+j} - F_{n+j}) + \frac{\sigma(1)}{\alpha_k}\Delta t T_n. \quad (9.17)$$

Let

$$\mathbf{e}^{(n)} = [e_n, e_{n+1}, \ldots, e_{n+k-1}]^T \quad (9.18)$$

and apply Lipschitz condition to

$$\begin{aligned}|f_{n+j} - F_{n+j}| &= |f(t_{n+j}, u_{n+j}) - f(t_{n+j}, U_{n+j})| && (9.19)\\ &\le L|u_{n+j} - U_{n+j}| && (9.20)\end{aligned}$$

or using constants $L_{n,j}$,

$$f_{n+j} - F_{n+j} = L_{n,j}(u_{n+j} - U_{n+j}) \quad (9.21)$$

for some $L_{n,j}$ with $|L_{n,j}| \le L$ so that

$$\mathbf{e}^{(n+1)} = (A + \Delta t D_n)\mathbf{e}^{(n)} + \frac{\sigma(1)}{\alpha_k}\Delta t\begin{bmatrix}0\\ \vdots\\ 0\\ T_n\end{bmatrix} \quad (9.22)$$

with

$$A = \begin{pmatrix}0 & 1 & 0 & \ldots & 0\\ 0 & 0 & 1 & \ldots & 0\\ \vdots & & & & \\ 0 & \ldots & \ldots & 0 & 1\\ -\frac{\alpha_0}{\alpha_k} & -\frac{\alpha_1}{\alpha_k} & \ldots & \ldots & -\frac{\alpha_{k-1}}{\alpha_k}\end{pmatrix}, D_n = \begin{pmatrix}0 & \ldots & \ldots & 0\\ & \ddots & & \\ & & \ddots & \\ 0 & & & 0\\ L_{n,0} & \ldots & \ldots & L_{n,k-1}\end{pmatrix} \quad (9.23)$$

Hence

$$\mathbf{e}^{(n)} = \prod_{m=0}^{n-1}(A + \Delta t D_m)\mathbf{e}^{(0)} + \frac{\sigma(1)}{\alpha_k}\Delta t\sum_{j=0}^{n-1}\prod_{j+1}^{n-1}(A + \Delta t D_m)T_j \quad (9.24)$$

then using zero stability we can bound matrices so that

$$\|\mathbf{e}^{(n)}\|_\infty \le K\|\mathbf{e}^{(0)}\|_\infty + \left|\frac{\sigma(1)}{\alpha_k}\right| n\Delta t K \max_j |T_j|. \quad (9.25)$$

1. As $\Delta t \to 0$, starting data is consistent, so that $\|\mathbf{e}^{(0)}\|_\infty \to 0$ as $\Delta t \to 0$.
2. As $n\Delta t \to t$ and $\Delta t \to 0$, $\max_j |T_j| \to 0$ so we must also have $\|\mathbf{e}^{(n)}\|_\infty \to 0$, that is the method converges.

When $\max |T_j| = \mathcal{O}(\Delta t^p)$ then as $\Delta t \to 0$, so also is $\|\mathbf{e}^{(n)}\|_\infty$ provided $\|\mathbf{e}^{(0)}\|_\infty = \mathcal{O}(\Delta t^p)$ as $\Delta t \to 0$. $\square$

Lecture 10

(v) Absolute Stability

Zero stability tells us that as $\Delta t \to 0$, a method will converge. However, we compute with fixed $\Delta t > 0$ so we need to know that for fixed Δt, $|U_n|$ will approximate closely the correct solution, usually a pre-requisite for this to happen is that we need to ensure at least that the approximation remains bounded as n becomes large. It is too difficult to proceed much further for arbitrary functions $f(t,u)$ so theory is developed for a simplified system, our general experience is that the results we obtain for this simple case are excellent indicators of conditions that are needed for general situations.

In particular, we study the application to the problem where $f(t,u) = \lambda u$,

$$u' = \lambda u, \quad u(0) = u_0 \neq 0,\ \mathrm{Re}(\lambda) < 0 \tag{10.1}$$

with exact solution

$$u(t) = u_0 \mathrm{e}^{\lambda t} \tag{10.2}$$

where

$$\left|\frac{u(t)}{u_0}\right| = \mathrm{e}^{-|\mathrm{Re}(\lambda)|t} \tag{10.3}$$

so that

$$\left|\frac{u(t)}{u_0}\right| \to 0 \quad \text{as } t \to \infty. \tag{10.4}$$

Applying a linear multistep method with $f = \lambda u$ give the scheme:

$$\sum_{r=0}^{k} \left(\alpha_r - \lambda \Delta t \beta_r\right) U_{n+r} = 0. \tag{10.5}$$

This is a difference equation and as we have used before, we try a solution $U_n = az^n$ with $|z| \neq 0$ for some constant a, so that

$$\sum_{r=0}^{k} \left(\alpha_r - \lambda \Delta t \beta_r\right) z^{n+r} = 0 \tag{10.6}$$

$$z^n \left[\left(\sum_{r=0}^{k} \alpha_r z^r\right) - \lambda \Delta t \sum_{r=0}^{n} \beta_r z^r\right] = 0 \tag{10.7}$$

$$z^n \left[\rho(z) - \lambda \Delta t \sigma(z)\right] = 0. \tag{10.8}$$

Define a polynomial

$$\Pi(z;\overline{\Delta t}) = \rho(z) - \overline{\Delta t}\sigma(z) \tag{10.9}$$

where $z^n\Pi(z;\overline{\Delta t}) = 0$ when U_n is a solution of the recurrence relation. However, since

$$|az^n| = |a\mathrm{e}^{n\log z}| \tag{10.10}$$

for $U_n \to 0$ as $n \to \infty$ we will need $|z| < 1$ for *each* root of $\Pi(z;\overline{\Delta t}) = 0$.

This is an important but subtle point. The continuous system has but one decaying exponential solution. When we use a k-term multi-step method, we introduce additional solutions to the k^{th} order difference equation, and we need as a minimum requirement for stability that *each* solution of the difference equation will decay when f is proportional to u *for the value of* Δt *we use.* If there is one solution of the discrete system that, for our fixed $\Delta t > 0$ that does not have the property of decay to zero when $f \sim u$, then when we use floating point arithmetic, there will always be a component of this 'unstable' solution in the discrete approximation and it will grow exponentially in value with the number of time steps.

Definition. *A linear multistep method is absolutely stable for* $\overline{\Delta t}$ *if and only if all roots of* $\Pi(z;\overline{\Delta t})$ *lie* inside *the unit disc. An interval* $a < \overline{\Delta t} < b$ *where the method is absolutely stable is called an* **interval of absolute stability**.

Example 1: Simpson's Rule

$$\begin{aligned} U_{n+2} - U_n &= \frac{\Delta t}{3}\left(F_{n+2} + 4F_{n+1} + F_n\right) & (10.11)\\ \rho(z) &= z^2 - 1 & (10.12)\\ \sigma(z) &= \frac{1}{3}\left(z^2 + 4z + 1\right) & (10.13)\\ \Pi(z;\overline{\Delta t}) &= \left(1 - \frac{\overline{\Delta t}}{3}\right)z^2 - \frac{4}{3}\overline{\Delta t}z - \left(1 + \frac{\overline{\Delta t}}{3}\right). & (10.14)\end{aligned}$$

Roots are

$$\begin{aligned} z_1(\overline{\Delta t}) &= \frac{\sqrt{1 + \frac{1}{3}\overline{\Delta t}^2} + \frac{2}{3}\overline{\Delta t}}{1 - \frac{\overline{\Delta t}}{3}} & (10.15)\\ z_2(\overline{\Delta t}) &= \frac{-\sqrt{1 + \frac{1}{3}\overline{\Delta t}^2} + \frac{2}{3}\overline{\Delta t}}{1 - \frac{\overline{\Delta t}}{3}} & (10.16)\end{aligned}$$

and in class exercise, show that one root always has modulus greater than one. Hence method zero stable but never absolutely stable.

Example 2: Euler's method (class exercise)

$$\begin{aligned} U_{n+1} - U_n &= \Delta t F_n & (10.17)\\ \Pi(z;\overline{\Delta t}) &= z - 1 - \overline{\Delta t}. & (10.18)\end{aligned}$$

Root is $z = 1 + \overline{\Delta t}$ and $|z| < 1$ implies $-1 < 1 + \overline{\Delta t} < 1$ or $-2 < \overline{\Delta t} < 0$, so interval of absolute stability is $(-2, 0)$.

Definition. *A linear k step method is absolutely stable in an open set R_A of the complex plane if for all $\overline{\Delta t} \in R_A$, the roots z_j of the stability polynomial $\Pi(z; \overline{\Delta t})$ satisfy $|z_j| < 1$. The set R_A is the* **region of absolute stability**.

Definition. *The method is A-stable if the region of absolute stability R_A contains the whole left hand complex half plane.*

Example 1: Explicit Euler

$$\Pi(z; \overline{\Delta t}) \quad = \quad z - (1 + \overline{\Delta t}). \tag{10.19}$$

This has root $z = 1 + \overline{\Delta t}$ and so the region of absolute stability is a circle with centre -1 and radius 1 as shown in Figure 9. This method is not A-stable.

Example 2: Implicit Euler

$$\Pi(z; \overline{\Delta t}) \quad = \quad (1 - \overline{\Delta t})z - 1. \tag{10.20}$$

This has root $z = (1 - \overline{\Delta t})^{-1}$ and so the region of absolute stability is the whole complex plane except for a circle with centre 1 and radius 1 as shown in Figure 10. This method is A-stable.

Some further results (proofs beyond the scope of the course) are

Theorem: Dahlquist (1963). *Dahlquist's theorem states:*

1. *No explicit linear multistep method is A stable.*
2. *The order of an A stable implicit linear multistep method cannot exceed 2.*
3. *The second order A stable linear multistep method with smallest error constant in the trapezoidal rule.*

Definitions. *A linear multistep method is $A(\alpha)$-stable for $\alpha \in \left(0, \frac{\pi}{2}\right)$ if the region of absolute stability contains the infinite wedge*

$$-\pi - \alpha \quad < \quad \arg(\overline{\Delta t}) \quad < \quad -\pi + \alpha \tag{10.21}$$

as shown in Figure 11.

A method is $A(0)$-stable if it is $A(\alpha)$-stable for some $\alpha \in \left(0, \frac{\pi}{2}\right)$.

A method is A_0-stable if R_A includes the negative real axis.

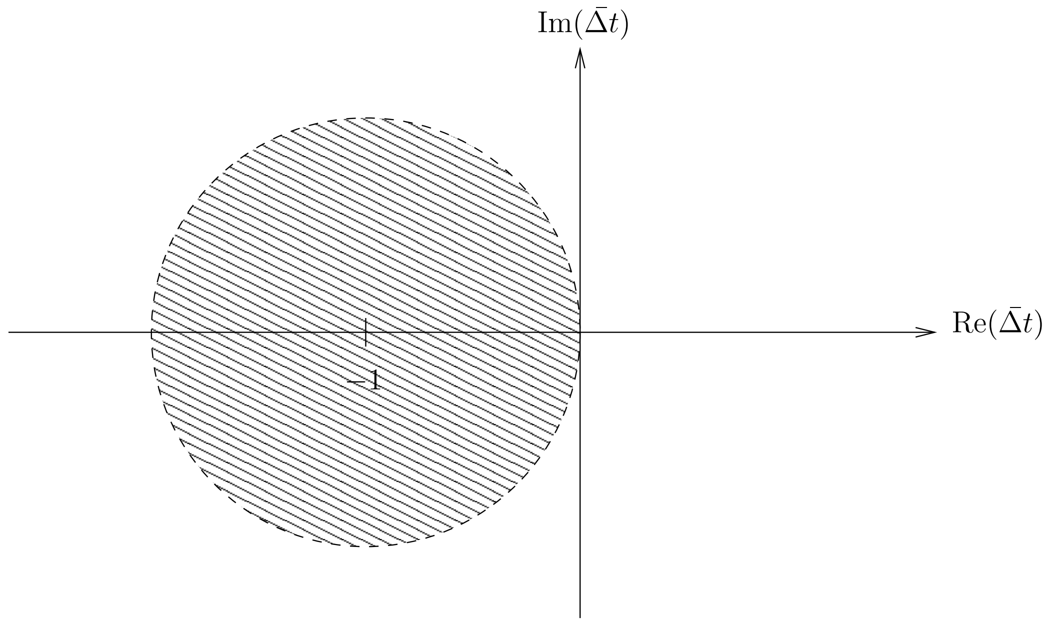


Figure 9: Showing shaded region of absolute stability for the explicit Euler method.

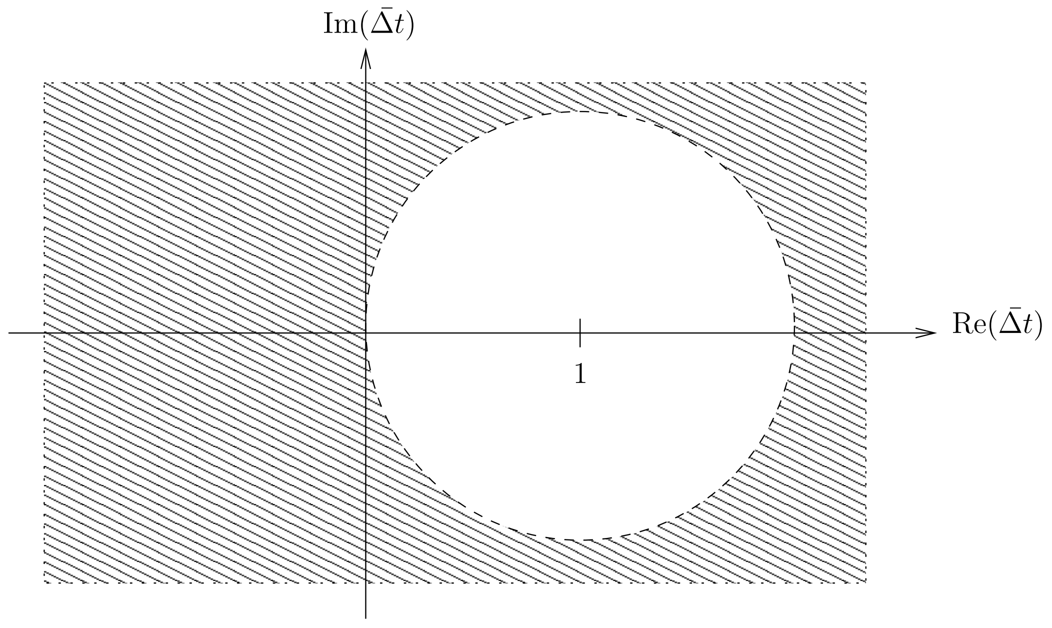


Figure 10: Showing shaded region of absolute stability for the implicit Euler method.

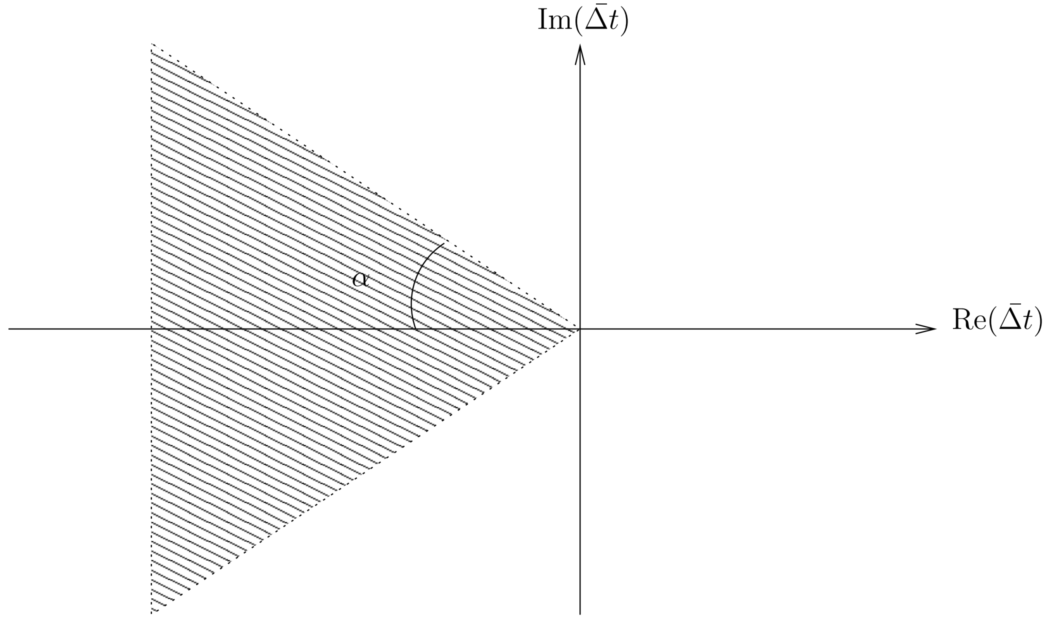


Figure 11: Showing shaded region of absolute stability for an A(α) stable method.

Since $\Pi(z;\overline{\Delta t}) = \rho(z) - \overline{\Delta t}\sigma(z)$, if ξ is a root then

$$\frac{\rho(\xi)}{\overline{\Delta t}} \quad = \quad \sigma(\xi) \tag{10.22}$$

and as $\overline{\Delta t} \to -\infty$ will need $\sigma(\xi(\overline{\Delta t})) \to 0$, so roots ξ will also be roots of just $\sigma(z) = 0$. If choose $\sigma(z) = \beta_k z^k$ then will know that $\xi(\overline{\Delta t}) \to 0$ as $\overline{\Delta t} \to -\infty$. This leads to a family of implicit backward differentiation (BDF) methods

$$\sum_{j=0}^{k} \alpha_j U_{n+j} \quad = \quad \beta_k \Delta t F_{n+k} \tag{10.23}$$

(with $\alpha_k = 1$).

k	stability	convergence p
1	A stable	$p = 1$
2	A stable	$p = 2$
3	A($88\pi/180$) stable	$p = 3$
4	A($73\pi/180$) stable	$p = 4$

For $k > 2$, stability is restricted to a wedge as in Figure 11.

with solutions of implicit equations found by and explicit predictor step and then by

Newton iteration to solve

$$U_{n+k} - \Delta t \beta_k f(t_{n+k}, U_{n+k}) \quad = \quad -\sum_{j=0}^{k-1} \alpha_j U_{n+j} \qquad (10.24)$$

using the explicit predictor value as starting value for Newton iterations.

(vi) Stiff systems

In dealing with systems of differential equations it is relatively common to find that the one step and multistep methods we have studied may not work well with some systems where different parts of the overall solution evolve or decay on very different time scales.

As a model for this type of behaviour, consider

$$u'' + (1+a)u' + au \quad = \quad 0 \qquad (10.25)$$

re-written as

$$\begin{aligned} u' &= v && (10.26) \\ v' &= -(1+a)v - au && (10.27) \end{aligned}$$

or

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} u \\ v \end{bmatrix} \quad = \quad \underbrace{\begin{bmatrix} 0 & 1 \\ -a & -(1+a) \end{bmatrix}}_{A} \begin{bmatrix} u \\ v \end{bmatrix} \qquad (10.28)$$

with solutions $u = c_1 \mathrm{e}^{-t} + c_2 \mathrm{e}^{-at}$.

If $a \gg 1$ then there are two very different time scales, one term $\mathcal{O}(1)$ and one $\mathcal{O}(a^{-1})$.

If we calculate eigenvalues ($\lambda_1 = -1$ and $\lambda_2 = -a$) and eigenvectors

$$\mathbf{z}_1 \quad = \quad \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}, \quad \mathbf{z}_2 \quad = \quad \begin{pmatrix} -\frac{1}{\sqrt{1+a^2}} \\ \frac{a}{\sqrt{1+a^2}} \end{pmatrix} \qquad (10.29)$$

then A can be decomposed

$$A \quad = \quad X \Lambda X^{-1} \quad \text{where } X \quad = \quad [\mathbf{z}_1, \mathbf{z}_2] \;\text{ and } \Lambda \quad = \quad \begin{bmatrix} -1 & 0 \\ 0 & -a \end{bmatrix} \qquad (10.30)$$

so if

$$\begin{bmatrix} p \\ q \end{bmatrix} \quad = \quad X^{-1} \begin{bmatrix} u \\ v \end{bmatrix} \quad = \quad \begin{bmatrix} \sqrt{2}(au+v) \\ \sqrt{1+a^2}(u+v) \end{bmatrix} \qquad (10.31)$$

where

$$X = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{1+a^2}} \\ -\frac{1}{\sqrt{2}} & \frac{a}{\sqrt{1+a^2}} \end{bmatrix} \tag{10.32}$$

then

$$\frac{\mathrm{d}p}{\mathrm{d}t} = -p \tag{10.33}$$

$$\frac{\mathrm{d}q}{\mathrm{d}t} = -aq \tag{10.34}$$

however, integration of both equations together would proceed using the time step needed for stability of (10.34). Such a system is called *stiff*.

If we consider a more general linear system with some matrix A on the RHS,

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = A\mathbf{u} \tag{10.35}$$

explicit Euler would give

$$\mathbf{U}_{n+1} = (I + \Delta t A)\mathbf{U}_n = (I + \Delta t A)^{n+1}\mathbf{U}_0 \tag{10.36}$$

and re-writing

$$I + \Delta t A = X\Lambda X^{-1} \tag{10.37}$$

where

$$\Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{pmatrix} \tag{10.38}$$

is diagonal matrix of eigenvalues (Λ may be more complicated if there are multiple eigenvalues, if interested, look up Jordan normal form) then

$$\mathbf{U}_{n+1} = X\Lambda^{n+1}X^{-1}\mathbf{U}_0 \tag{10.39}$$

so we need $\max_j |\lambda_j| < 1$ where λ_j are eigenvalues of $I + \Delta t A$. However, if eigenvalues of A are μ_j, then the eigenvalues of $I + \Delta t A$ are $1 + \Delta t \mu_j$. Clearly we would need $\mathrm{Re}(\mu_j) < 0$ for each j in order that the system has only decaying solutions. Hence for absolute stability we will need

$$\Delta t < \frac{2}{\max |\mu_j|} \tag{10.40}$$

if the eigenvalues were all real and negative. In a stiff system normally have groups of eigenvalues near the origin (but with negative real part) and a few with large negative real part and they will force Δt to be very small for an explicit method.

If, however, we used an implicit method

$$\mathbf{U}_{n+1} = \mathbf{U}_n + \Delta t A \mathbf{U}_{n+1} \tag{10.41}$$

which can be written as

$$(I - \Delta t A)\mathbf{U}_{n+1} = \mathbf{U}_n \tag{10.42}$$

or

$$\mathbf{U}_{n+1} = (I - \Delta t A)^{-(n+1)}\mathbf{U}_0. \tag{10.43}$$

So now

$$\left| \frac{1}{1 - \Delta t \mu_j} \right| < 1 \tag{10.44}$$

for all j provided $\mathrm{Re}(\mu_j) < 0$ hence this method is stable for all step sizes if A is positive definite.

ODEs — Summary

- Continuous system $\mathcal{N}(t, u) = u' - f(t, u) = 0$, $u(0) = u_0$
- Discrete system $N(t_n, U_{n+1}, U_n) = 0$, $U_0 = u_0$
- Truncation error $T_n = N(t_n, u_{n+1}, u_n)$, explicit/implicit distinction
- One step methods $U_{n+1} = U_n + \Delta t \Phi(t_n, U_n)$
- Runge-Kutta methods, analysis of T_n, adaptive time steps
- Multistep methods $\sum_{j=0}^{k} \alpha_j U_{n+j} = \Delta t \sum_{j=0}^{k} \beta_j F_{n+j}$. Derivation of simple cases, analysis of truncation error
- Zero stability — definition, root condition, proof that root condition is necessary
- Consistency — derivation of schemes of order p
- Convergence — definition and theorems
  1. Zero stability a necessary condition with proof
  2. Consistency a necessary condition with proof
  3. Dahlquist: convergence $\Leftrightarrow$ zero stability and consistency (without proof)
- Absolute stability, definition of stability polynomials, application to simple examples
- Definition of A stability and variants, understanding of background ideas about location of eigenvalues in complex plane
- Basic understanding of why some systems may be stiff

Lecture 11

II. Numerical Solution of Parabolic PDEs

While a great many interesting physical systems can be set in the framework of evolutionary ordinary differential equations, there are many more situations where both spatial and temporal variability needs to be determined. In the main, we will look at problems where there is one space dimension, and most of the methods we derive can be generalised to multiple space dimensions using orthogonal constructions of the one space dimension discretisation.

As a generalisation, whereas for systems of ODEs, stability is important, it is usually accuracy and implementation that dominate choice of numerical parameters such as time step or numerical method and the computation is algorithmically relatively straight forward to implement (although large systems with say, implicit backward difference (BDF) methods, can have implementation problems); when dealing with both time and space independent variables, stability can become an extremely problematic requirement and the consequent size of the computational problem can lead to algorithmic complexity that is separate from the discretisation of the original problem and is related to how solution of the discrete system can be computed in a practical way.

As a model problem in one space dimension, let $u = u(x,t)$ satisfy for $t > 0$

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(D(u)\frac{\partial u}{\partial x}\right), \quad D(u) > 0, \tag{11.1}$$

with $u(x,0) = u_0(x)$ and one of

(a) $u(x_L,t) = u_L(t)$, $u(x_R,t) = u_R(t)$ (bounded domain)

(b) $u(x,t) \to 0$ as $x \to \pm\infty$ (infinite domain).

Generally we will take $D = 1$.

We can have more complicated boundary conditions involving derivatives, you should have already encountered the notation: Dirichlet conditions for fixed boundary values, Neumann conditions for derivative boundary conditions and Robin conditions for mixed value and derivative conditions.

In order to understand some of the problems that come about in discretising PDEs we need to recall a little about the behaviour of continuous systems with diffusion.

In particular, consider the specific model problem

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \tag{11.2}$$
$$u(x,0) = u_0(x) \tag{11.3}$$
$$u \to 0 \quad \text{as } x \to \pm\infty. \tag{11.4}$$

Methods for analytic solution of this equation are not part of this course but the general form of the analytic solution is of interest since that defines many properties we require a numerical scheme to replicate if it is to be accurate.

Define a Fourier transform for $v(x)$ by

$$\hat{v}(k) = \int_{-\infty}^{\infty} v(x)\mathrm{e}^{-ikx}\,\mathrm{d}x = F(v). \tag{11.5}$$

Apply the transform to the PDE giving

$$\int_{-\infty}^{\infty} \frac{\partial u}{\partial t}\mathrm{e}^{-ikx}\,\mathrm{d}x = \int_{-\infty}^{\infty} \frac{\partial^u}{\partial x^2}\mathrm{e}^{-ikx}\,\mathrm{d}x. \tag{11.6}$$

On the left-hand-side we assume we can change the order of integration and differentiation and on the right-hand-side we integrate by parts and use the fact that $u \to 0$ and $\frac{\partial u}{\partial x} \to 0$ as $x \to \pm\infty$ to get

$$\frac{\mathrm{d}}{\mathrm{d}t}\underbrace{\int_{-\infty}^{\infty} u(x,t)\mathrm{e}^{-ikx}\,\mathrm{d}x}_{\hat{u}(k,t)} = -k^2\underbrace{\int_{-\infty}^{\infty} u(x,t)\mathrm{e}^{-ikx}\,\mathrm{d}x}_{\hat{u}(k,t)}, \tag{11.7}$$

or

$$\frac{\mathrm{d}\hat{u}}{\mathrm{d}t} = -k^2\hat{u}, \tag{11.8}$$

with solutions $\hat{u}(k,t) = A(k)\mathrm{e}^{-k^2 t}$ for some function $A(k)$. Applying the initial condition allows us to determine $A(k)$ giveing

$$\hat{u}(k,t) = \hat{u}_0(k)\mathrm{e}^{-k^2 t}.$$

Next we apply the inverse Fourier transform

$$v(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{v}(k)\mathrm{e}^{ikx}\,\mathrm{d}k = F^{-1}[\hat{v}] \tag{11.9}$$

so

$$u(x,t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{u}_0(k)\mathrm{e}^{-k^2 t}\mathrm{e}^{ikx}\,\mathrm{d}k. \tag{11.10}$$

The reason for deriving this form of the continuous solution is that it shows there will be a sum (integral over k) of modes of the form

$$e^{-k^2t}e^{ikx}. \tag{11.11}$$

When we look in more detail at a numerical approximation on a discrete mesh, with $t = n\Delta t$ and $x = rh$, as we shall see shortly, we define a numerical parameter $\mu = \Delta t/h^2$ and then the continuous solution should obey

$$\hat{u}(k, t + \Delta t) = e^{-\mu(kh)^2}\hat{u}(k, t), \tag{11.12}$$

or that each continuous mode should have amplification factor over a time step Δt of

$$\Lambda(k) = e^{-\mu(kh)^2}. \tag{11.13}$$

If the continuous solution is then restricted to the discrete mesh, each mode of the continuous solution will have the form

$$u_r^n \sim [e^{-\mu(kh)^2}]^n e^{ikrh} \equiv \Lambda(k)^n e^{ikrh}. \tag{11.14}$$

A consequence of this structure is that in a numerical scheme we look for discrete modes of the form

$$U_r^n \sim \lambda(k)^n e^{ikrh}, \tag{11.15}$$

and we can view the discrete system as an attempt to have $\lambda(k)$ replicate $\Lambda(k)$.

This is a key concept: provided we are dealing with a linear PDE, the analytic solution will be a composition (i.e. integral over k) of modes of the form (11.11) each with an amplification factor $\Lambda(k)$ and assuming our discrete approximation is also a linear system, then the numerical solution should likewise be a composition (i.e. sum over r) of modes of the form (11.15) and these modes are oscillatory in the space dimension together with some amplification (decay or growth) factor λ for each time step.

The inverse Fourier transform can be manipulated further by substituting the ingtegral form for $\hat{u}_0(k)$,

$$\begin{aligned} u(x,t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u_0(s)e^{isk}\,\mathrm{d}s\ e^{-k^2t}e^{\imath kx}\,\mathrm{d}k & (11.16)\\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} u_0(s)\int_{-\infty}^{\infty} e^{-k^2t+\imath k(x-s)}\,\mathrm{d}k\,\mathrm{d}s. & (11.17)\end{aligned}$$

Let $k^2t = p^2$ so that $\mathrm{d}k = \mathrm{d}p/\sqrt{t}$ and $\xi = \frac{x-s}{2\sqrt{t}}$. Then

$$\begin{aligned} u(x,t) &= \frac{1}{2\pi\sqrt{t}}\int_{-\infty}^{\infty} u_0(s)\int_{-\infty}^{\infty} e^{-\left(p-\frac{\imath(x-s)}{2\sqrt{t}}\right)^2} e^{\frac{(x-s)^2}{4t}}\,\mathrm{d}p\,\mathrm{d}s & (11.18)\\ &= \frac{1}{2\pi\sqrt{t}}\int_{-\infty}^{\infty} u_0(s)e^{\frac{(x-s)^2}{4t}}\int_{-\infty}^{\infty} e^{-(p-\imath\xi)^2}\,\mathrm{d}p\,\mathrm{d}s. & (11.19)\end{aligned}$$

The integral in p has no poles so we can change variables to integrate along a new path from $-\infty - \imath\xi$ to $\infty - \imath\xi$ and then we can move the path to $-\infty$ to ∞ so that the value of the integral is $\int_{-\infty}^{\infty} \mathrm{e}^{-p^2}\,\mathrm{d}p = 2\int_0^{\infty} \mathrm{e}^{-p^2}\,\mathrm{d}p = \sqrt{\pi}$. Hence

$$u(x,t) \quad = \quad \frac{1}{2\sqrt{\pi t}} \int_{-\infty}^{\infty} u_0(s) \mathrm{e}^{\frac{(x-s)^2}{4t}}\,\mathrm{d}s. \tag{11.20}$$

Or, using $\frac{x-s}{2\sqrt{t}} = \xi$ so that $s = x - 2\sqrt{t}\xi$, $\mathrm{d}s = -2\sqrt{t}\,\mathrm{d}\xi$, as $s \to -\infty$, $\xi \to +\infty$ and as $s \to +\infty$, $\xi \to -\infty$, we get

$$u(x,t) \quad = \quad \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} u_0(x - 2\sqrt{t}\xi) \mathrm{e}^{-\xi^2}\,\mathrm{d}\xi \tag{11.21}$$

and if we let $u_0(x) = \delta(x)$ then

$$u(x,t) \quad = \quad \frac{1}{2\sqrt{\pi t}} \mathrm{e}^{-x^2/4t}. \tag{11.22}$$

The solution decays to zero as $t \to \infty$ as shown in Figure 12.

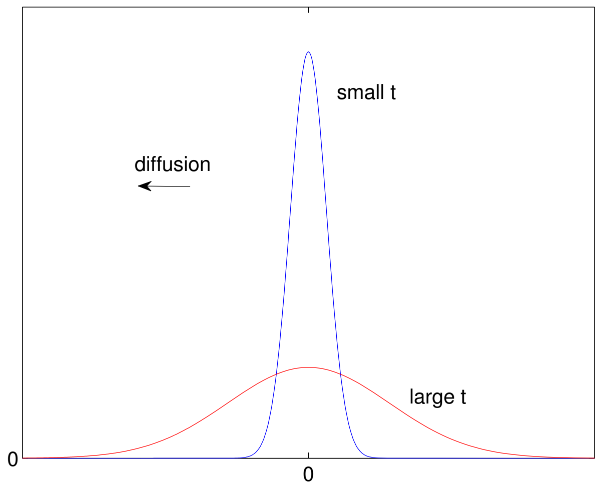


Figure 12: Solution in time and space of a diffusion equation with an initial delta function distribution in space.

For continuous functions $v(x)$ with transform $\hat{v}(k)$ we can define an L_2 norm

$$\|v\|_{L_2} \quad \equiv \quad \|v\|_2 \quad = \quad \left| \int_{-\infty}^{\infty} |v(x)|^2\,\mathrm{d}x \right|^{1/2} \tag{11.23}$$

(note the term $|v(x)|^2$ rather than $v(x)^2$ since v may be complex and a similar integral over k for $\hat{v}$).

Parseval's Identity. *An important identity relating the norm of a function to the norm of its Fourier transform is:*

$$\|v\|_2 \;=\; \frac{1}{\sqrt{2\pi}}\|\hat{v}\|_2 \qquad (11.24)$$

or equivalently

$$\int_{-\infty}^{\infty} |v(x)|^2 \,\mathrm{d}x \;=\; \frac{1}{2\pi}\int_{-\infty}^{\infty} |\hat{v}(k)|^2 \,\mathrm{d}k. \qquad (11.25)$$

Proof. **Method A** (the course does not assume knowledge of δ functions but this is a standard proof of Parseval's identity)

$$\begin{aligned}
\|v\|_2^2 &= \int_{-\infty}^{\infty} v(x)\bar{v}(x)\,\mathrm{d}x && (11.26)\\
&= \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{v}(k)\mathrm{e}^{ikx}\,\mathrm{d}k \int_{-\infty}^{\infty} \bar{\hat{v}}(s)\mathrm{e}^{-isx}\,\mathrm{d}s\,\mathrm{d}x && (11.27)\\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{v}(k)\bar{\hat{v}}(s)\underbrace{\left[\frac{1}{2\pi}\int_{-\infty}^{\infty} \mathrm{e}^{ix(k-s)}\,\mathrm{d}x\right]}_{\delta(k-s)}\mathrm{d}k\,\mathrm{d}s && (11.28)\\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{v}(k)\int_{-\infty}^{\infty} \bar{\hat{v}}(s)\delta(k-s)\,\mathrm{d}s\,\mathrm{d}k. && (11.29)
\end{aligned}$$

Hence

$$\|v\|_2^2 \;=\; \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{v}(k)\bar{\hat{v}}(k)\,\mathrm{d}k \;=\; \frac{1}{2\pi}\|\hat{v}\|_2^2 \qquad (11.30)$$

Method B

Let $v(x)$ and $w(x)$ be two functions, then

$$\begin{aligned}
\int_{-\infty}^{\infty} \hat{w}(x)v(x)\,\mathrm{d}x &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w(s)\mathrm{e}^{-isx}\,\mathrm{d}s\; v(x)\,\mathrm{d}x && (11.31)\\
&= \int_{-\infty}^{\infty} w(s)\int_{-\infty}^{\infty} \mathrm{e}^{-isx}v(x)\,\mathrm{d}x\,\mathrm{d}s. && (11.32)
\end{aligned}$$

Hence

$$\int_{-\infty}^{\infty} \hat{w}(x)v(x)\,\mathrm{d}x \;=\; \int_{-\infty}^{\infty} w(s)\hat{v}(s)\,\mathrm{d}s. \qquad (11.33)$$

Let $w(s) = \bar{\hat{v}}(s)$, then

$$\hat{w}(s) \;=\; \int_{-\infty}^{\infty} \bar{\hat{v}}(x)\mathrm{e}^{-isx}\,\mathrm{d}x \;=\; \overline{\int_{-\infty}^{\infty} \hat{v}(x)\mathrm{e}^{isx}\,\mathrm{d}x} \;=\; \overline{2\pi v}. \qquad (11.34)$$

This gives

$$2\pi \int_{-\infty}^{\infty} \bar{v}(x) v(x) \, \mathrm{d}x \quad = \quad \int_{-\infty}^{\infty} \bar{\hat{v}}(s) \hat{v}(s) \, \mathrm{d}s, \tag{11.35}$$

or equivalently

$$\|v\|_2^2 \quad = \quad \frac{1}{2\pi} \|\hat{v}\|_2^2. \tag{11.36}$$

$\square$

This leads to an important feature of this model problem since we have shown

$$\hat{u}(k,t) \quad = \quad \mathrm{e}^{-k^2 t} \hat{u}_0(k) \tag{11.37}$$

so that

$$\|u\|_2 \quad = \quad \frac{1}{\sqrt{2\pi}} \|\hat{u}\|_2 \quad = \quad \frac{1}{\sqrt{2\pi}} \|\mathrm{e}^{-k^2 t} \hat{u}_0(k)\|_2 \tag{11.38}$$

and

$$\|u\|_2 \leq \frac{1}{\sqrt{2\pi}} \max_k |\mathrm{e}^{-k^2 t/2}| \cdot \|\hat{u}_0\|_2 \quad \leq \quad \frac{1}{\sqrt{2\pi}} \|\hat{u}_0\|_2 = \|u_0\|_2, \tag{11.39}$$

or

$$\|u(t)\|_2 \leq \|u(0)\|_2, \quad \forall \, t > 0, \tag{11.40}$$

and so we will want any numerical solution to behave this way too.

If we consider the case where x is confined to a finite region, for example

$$\begin{aligned} u_t &= u_{xx}, \quad 0 < x < 1 && (11.41) \\ u(0,t) &= 0 && (11.42) \\ u(1,t) &= 0 && (11.43) \\ u(x,0) &= u_0(x), \quad 0 \leq x \leq 1 && (11.44) \end{aligned}$$

(with initial and boundary conditions shown in Figure 13), then using separation of variables we try

$$u(x,t) \quad = \quad X(x) T(t) \tag{11.45}$$

and we see that

$$\frac{X''}{X}(x) \quad = \quad \frac{\dot{T}}{T}(t) \tag{11.46}$$

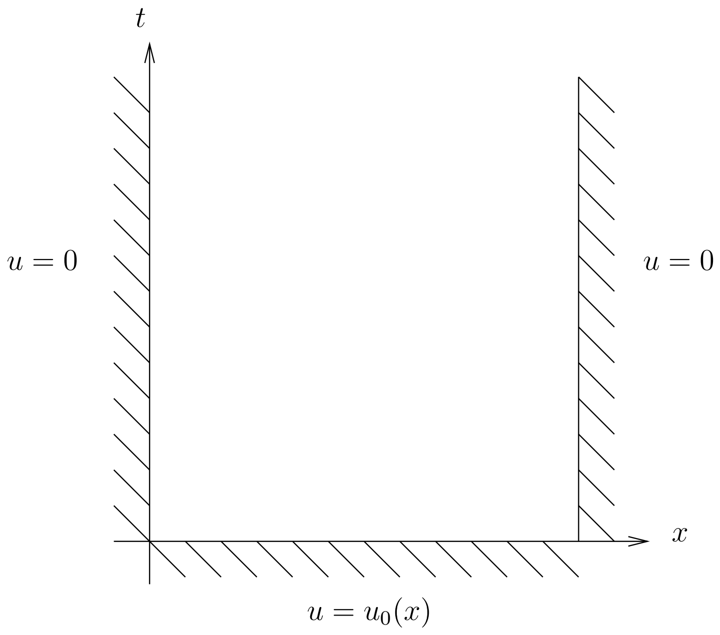


Figure 13: The initial and boundary conditions for the finite x case.

and so each side of this equation of these must be some constant. To satisfy the homogeneous boundary conditions we need the constant to be negative and to satisfy the boundary conditions at $x = 0$ and $x = 1$ we can choose

$$X_m = \sin(m\pi x), \quad m = 1, 2, \ldots \tag{11.47}$$

with

$$\dot{T}_m = -m^2\pi^2 T_m \tag{11.48}$$

or

$$T_m(t) = A_m \mathrm{e}^{-m^2\pi^2 t} \tag{11.49}$$

and so the solution is a weighted sum of modes

$$u(x,t) = \sum_{m=1}^{\infty} A_m \mathrm{e}^{-m^2\pi^2 t} \sin(m\pi x) \tag{11.50}$$

where the constants or weights A_m will come from the Fourier series of the initial condition:

$$u_0(x) = \sum_{m=1}^{\infty} A_m \sin(m\pi x). \tag{11.51}$$

Again, solutions will decay as $t \to \infty$.

Formulating a numerical approximation

In the first half of this course we looked into how approcimate solutions for ODE's are obtained at a set of discrete time points. For PDEs, we just extend the idea of approximating the solution at discrete time points to cater for both time and space discrete points. Indexing becomes more cumbersome as the number of space dimensions increases, we will adopt the notation that a superscript index relates to time stepping and subscript indices relate to space variables.

In the case where there is only one space dimension, define a mesh $x_r = rh$, $t_n = n\Delta t$, $h = 1/M$ and suppose $U_r^n \approx u(rh, n\Delta t)$. The mesh structure is illustrated in Figure 14.

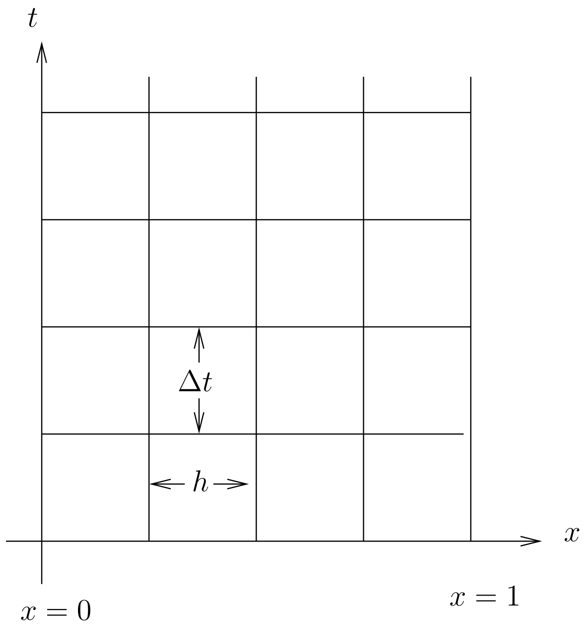


Figure 14: A mesh for the numerical solution.

The mesh can be viewed as a collection of nodes, with indicies r, n where $r = 0, 1, \ldots, M$, and $n = 0, 1, 2, \ldots$, each node corresponding to a physical point $(rh, n\Delta t)$.

The simplest form of approximation gives an explicit formula for the function values at a new time step, given all the values at the previous time step. We approximate derivatives with

$$\left(\frac{\partial u}{\partial t}\right)_r^n \approx \frac{U_r^{n+1} - U_r^n}{\Delta t} \tag{11.52}$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_r^n \approx \frac{U_{r+1}^n - 2U_r^n + U_{r-1}^n}{h^2}. \tag{11.53}$$

This can also be ilustrated by the finite difference stencil shown in Figure 15. The stencil is just a convenient way of showing which nodes are involved in the discretisation, in some books you will also see weights associated with the stencil nodes, the weights corresponding to coefficients in the finite difference approximations.

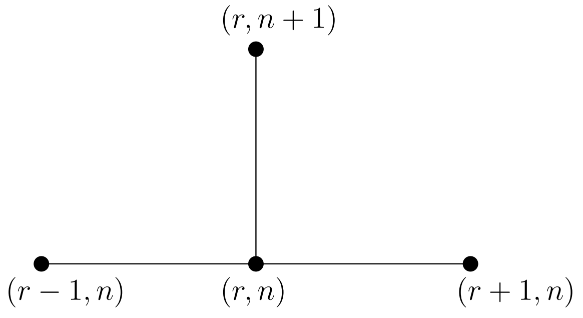


Figure 15: The finite difference stencil for the heat equation.

We can then proceed with analysis similarly to that for ODEs, the continuous system

$$\mathcal{N}(t,u) \;=\; \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} \;=\; 0, \tag{11.54}$$

can be written as a discrete system by

$$N(t,U) \;=\; \frac{U_r^{n+1} - U_r^n}{\Delta t} - \frac{U_{r+1}^n - 2U_r^n + U_{r-1}^n}{h^2} = 0. \tag{11.55}$$

Note that for ease of notation, the second argument of N is just written U without sub- or super-scripts. If we define $\mu = \Delta t/h^2$ then the algorithm to calculate approximate values is

$$U_r^{n+1} \;=\; (1-2\mu)U_r^n + \mu(U_{r-1}^n + U_{r+1}^n), \quad r = 1,\ldots,M-1, \quad n = 0,1,\ldots \tag{11.56}$$

Having defined the discrete operator N we can define a truncation error as before, to be the residual when the true continuous solution is substituted into the discrete operator,thgat is, set the truncation error to be

$$T_r^n = N(t,u) \;=\; \frac{u_r^{n+1} - u_r^n}{\Delta t} - \frac{u_{r+1}^n - 2u_r^n + u_{r-1}^n}{h^2}, \quad r = 1,\ldots,M-1, \; n = 1,2,\ldots.$$

The boundary conditions here are $U_0^n = 0$ and $U_M^n = 0$; there are inital conditions $U_r^0 = u_0(rh)$, $r = 0,1,\ldots,M$. With the boundary and initial conditions specified, then we can apply (11.56) for $r = 1,\ldots,M-1$ and $n = 1,2,\ldots$ giving a very simple explicit algorithm:

Algorithm

set $U_r^0 = u_0(rh)$ for $r = 0,\ldots,M$

```
while $t_n = n\Delta t < T_{\max}$
    $U_0^{n+1} = U_M^{n+1} = 0$
    apply (11.56) to calculate $U_r^{n+1}$ for $r = 1, \dots, M-1$
    increment $n$
end while
```

Lecture 12

Analysis of an explicit scheme

As we have noted, just as in the case of ODEs, in addition to stability, an important property of an approximation scheme is how well the scheme approximates the original continuous system and this means determining the truncation error of the scheme. Now we need to consider Taylor expansions in two or more variables, the time variable and however many space dimensions are involved. Having determined the truncation error, we next consider convergence, where we derive a criterion on the numerical parameters to guarantee convergence in the limit of time and space step vanishing, and, here we will complete the anlysis for conditions for which the scheme will be stable.

(i) Tuncation error: Explicit Scheme

$$\frac{U_r^{n+1} - U_r^n}{\Delta t} - \frac{U_{r+1}^n - 2U_r^n + U_{r-1}^n}{h^2} = 0 \tag{12.1}$$

$$T_r^n = \frac{u_r^{n+1} - u_r^n}{\Delta t} - \frac{u_{r+1}^n - 2u_r^n + u_{r-1}^n}{h^2}. \tag{12.2}$$

We have

$$u_r^{n+1} = u(rh,(n+1)\Delta t) = u_r^n + \Delta t \left(\frac{\partial u}{\partial t}\right)_r^n + \frac{1}{2}\Delta t^2 \left(\frac{\partial^2 u}{\partial t^2}\right)_r^n + \dots \tag{12.3}$$

$$u_{r+1}^n = u((r+1)h, n\Delta t) = u_r^n + h \left(\frac{\partial u}{\partial x}\right)_r^n + \frac{1}{2}h^2 \left(\frac{\partial^2 u}{\partial x^2}\right)_r^n + \dots \tag{12.4}$$

so that using $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$ either

$$T_r^n = \frac{1}{2}\Delta t \left.\frac{\partial^2 u}{\partial t^2}\right|_r^n + \mathcal{O}(\Delta t^2) - \frac{1}{12}h^2 \left.\frac{\partial^4 u}{\partial x^4}\right|_r^n + \mathcal{O}(h^4) \tag{12.5}$$

or if we terminate the Taylor series,

$$T_r^n = \frac{1}{2}\Delta t \frac{\partial^2 u}{\partial t^2}(rh, \eta_n) - \frac{1}{12}h^2 \frac{\partial^4 u}{\partial x^4}(\xi_r, t_n) \tag{12.6}$$

for $\eta_n \in (t_n, t_{n+1})$ and $\xi_r \in (x_{r-1}, x_{r+1})$. So the scheme is first order in time and second order in space.

Since $T_r^n \to 0$ as $\Delta t \to 0, h \to 0$ we call the scheme *unconditionally consistent.*

We can use the PDE to write $u_{tt} = u_{xxxx}$, so that we can write $T_r^n = \frac{1}{2}\Delta t u_{xxxx}(1 - 1/(6\mu)) + \mathcal{O}(\Delta t^2)$ with $\mu = \Delta t/h^2$, although this ordering only works provided h and Δt to go to zero on a refinement path with μ =constant (i.e. $\mu \nrightarrow 0$), otherwise the two leading terms will have different asymptotic order (but T will still go to zero as both Δt and h go to zero).

(ii) Convergence: Explicit Scheme

Lemma. *The explicit scheme for $u_t = u_{xx}$ converges if $\mu = \Delta t/h^2 \leq 1/2$.*

Proof. Let $e_j^n = u_j^n - U_j^n$, subtracting (12.1) from (12.2) gives

$$\frac{e_r^{n+1} - e_r^n}{\Delta t} - \frac{e_{r+1}^n - 2e_r^n + e_{r-1}^n}{h^2} = T_r^n \tag{12.7}$$

and so

$$e_r^{n+1} = (1-2\mu)e_r^n + \mu e_{r+1}^n + \mu e_{r-1}^n - \Delta t T_r^n. \tag{12.8}$$

Let $E^n = \max_r |e_r^n|$, then provided $1 - 2\mu \geq 0$

$$|e_r^{n+1}| \leq (1-2\mu)E^n + \mu E^n + \mu E^n + \Delta t|T_r^n| \tag{12.9}$$

or

$$|e_r^{n+1}| \leq E^n + \Delta t|T_r^n| \tag{12.10}$$

and provided $E^0 = 0$, then with $T = \max_{r,n}|T_r^n| \leq C\Delta t$ we have

$$|e_r^{n+1}| \leq (n+1)\Delta t T = t_{n+1}C\Delta t \tag{12.11}$$

and if $t_n \to t$ as $\Delta t \to 0$, $n \to \infty$

$$|e_r^{n+1}| \leq tC\Delta t \to 0. \tag{12.12}$$

Hence $|e_r^n| \to 0$ as $\Delta t \to 0$, $n \to \infty$, $n\Delta t \to t$.

We now need one further step to prove convergence, if we write

$$\begin{aligned} |u(x_r,t) - U_r^n| &= |u(x_r,t) - u(x_r,t_n) + u(x_r,t_n) - U_r^n|, && (12.13)\\ &\leq |u(x_r,t) - u_r^n| + |u_r^n - U_r^n|, && (12.14)\end{aligned}$$

since u is continuous in time the first term will vanish in the limit $t_n \to t$ and we have shown that the second term likewise vanishes in this limit so the scheme converges. $\square$

(iii) Stability: Explicit Scheme

We have noted two ways to approach stability, one is by using semi-discrete Fourier transforms, the other is by a modal analysis, both assume a whole space domains, or a finite domain extended to the whole space. Analysis using a semi-descrete transform proceeds as in (12.32)-(12.36) to obtain

$$\hat{U}^{n+1}(k) = [1-4\mu\sin^2(kh/2)] = \lambda(k)\hat{U}^n(k). \tag{12.15}$$

Alternately, in a modal analysis, substitute into the discrete scheme

$$U_r^{n+1} = U_r^n + \mu(U_{r+1}^n - 2U_r^n + U_{r-1}^n). \tag{12.16}$$

the mode

$$U_r^n \sim \lambda^n \mathrm{e}^{ikrh} \tag{12.17}$$

and cancel common factors to obtain

$$\lambda = 1+\mu(\mathrm{e}^{ikh}-2+\mathrm{e}^{-ikh}) \tag{12.18}$$
$$= 1+\mu(2\cos(kh)-2) \tag{12.19}$$

and so

$$\lambda = 1-4\mu\sin^2\frac{kh}{2}. \tag{12.20}$$

Hence both methods give exactly the same amplitude factor λ for the numerical scheme.

If we let $p=\sin^2\frac{kh}{2}$, with $0\le p\le 1$

$$\lambda(k) = 1-4\mu p, \tag{12.21}$$

for stability we will need $-1\le 1-4\mu$ or $\mu\le 1/2$ so that $|\lambda|\le 1$ for all k as shown in Figure 16.

We can also observe the extent to which $\lambda(k)=1-4\mu\sin^2\frac{kh}{2}$ matches $\Lambda(k)=\mathrm{e}^{-\mu(kh)^2}$. The Taylor series for each are

$$\lambda(k) \sim 1-4\mu[\frac{kh}{2}-\frac{1}{6}(\frac{kh}{2})^3+\cdots]^2 \tag{12.22}$$
$$= 1-\mu(kh)^2+4\cdot\mu\cdot\frac{2}{6}(\frac{kh}{2})^4+\cdots \tag{12.23}$$
$$= 1-\mu(kh)^2+\frac{1}{12}\mu(kh)^4+\cdots, \tag{12.24}$$

and

$$\Lambda(k) \sim 1-\mu(kh)^2+\frac{1}{2}\mu^2(kh)^4+\cdots., \tag{12.25}$$

so the match is only for the first two terms. This is also illustrated in figure 17.

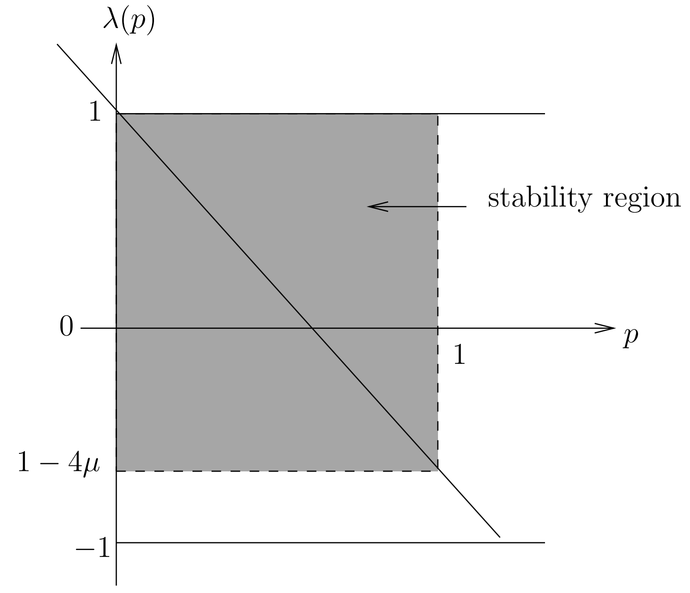


Figure 16: Stability region for the explicit scheme.

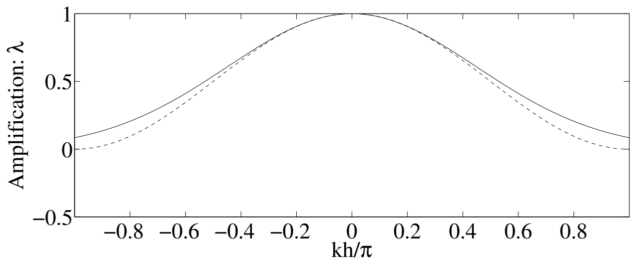


Figure 17: Comparison of amplification factors Λ (continuous diffusion equation) and λ (explicit Euler method) as the wave number kh varies in $[-\pi, \pi]$ for the case $\mu = 0.25$. (—-): Λ, (- - -) λ.

Further analysis of discrete schemes for PDEs

We have used that the solution for a linear system will be composed of modes of the form (11.15), that is $U_r^n \sim \lambda^n e^{ikrh}$ but we need to develop a more formal methodology to handle differences between solutions when we have space variation. For ODEs we considered stability solely by the behaviour of λ as $\Delta t \to 0$. Now we have to allow that $\lambda = \lambda(k, h, \Delta t)$ [although we will normally just write $\lambda = \lambda(k)$]. One way of considering differences between solutions *when the space domain is the whole real line* (or plane etc in higher space dimensions) is to use a use a *semi-discrete Fourier transform* for analysis. Suppose U_r, $r = 0, \pm 1, \pm 2, \ldots$ is a set of data (in our context, the space discretisation of a function at some point in time). First note that we cannot distinguish modes of the form e^{ikrh} when $|k| > \pi/h$ from those of lower k value (this is the phenomenon of aliasing) so for our numerical mesh we have to restrict $k \in [-\pi/h, \pi/h]$ and we can define a *semi-discrete Fourier transform*

$$\hat{U}(k) \;=\; h \sum_{r=-\infty}^{\infty} U_r e^{-ikrh} \quad k \in \left[-\frac{\pi}{h}, \frac{\pi}{h}\right], \tag{12.26}$$

with inverse

$$U_r \;=\; \frac{1}{2\pi} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{U}(k) e^{ikrh} \, dk. \tag{12.27}$$

There is an associated Parseval's Identity (class exercise)

$$\|U\|_{\ell_2} \;=\; \frac{1}{\sqrt{2\pi}} \|\hat{U}\|_{L_2}, \tag{12.28}$$

where we distinguish the norm of discrete data and a continuous function by

$$\|U^n\|_{\ell_2} \;=\; \left[h \sum_{r=-\infty}^{\infty} |U_r^n|^2 \right]^{1/2}. \tag{12.29}$$

and

$$\|\hat{U}\|_{L_2} \;=\; \left(\int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} |\hat{U}(k)|^2 \, dk \right)^{1/2}. \tag{12.30}$$

Having set up these norms, we can use them to measure differences between functions and between discrete data sets.

Definition. *A finite difference scheme is* practically stable *in the ℓ_2 norm if*

$$\|U^n\|_{\ell_2} \;\le\; \|U^0\|_{\ell_2} \tag{12.31}$$

If we take the explicit scheme for our model problem, written as

$$U_r^{n+1} \;=\; U_r^n + \mu \left(U_{r+1}^n - 2U_r^n + U_{r-1}^n \right) \tag{12.32}$$

multiply each term by $h e^{\imath krk}$ and sum we obtain the semi-discrete Fourier transform, so

$$\begin{aligned} h \sum_{r=-\infty}^{\infty} U_r^{n+1} e^{ikrh} \;&=\; h \sum_{r=-\infty}^{\infty} U_r^n e^{ikrh} \\ &+\; \mu \left(h \sum_{r=-\infty}^{\infty} e^{ikrh} \left(U_{r+1}^n - 2U_r^n + U_{r-1}^n \right) \right) \end{aligned} \tag{12.33}$$

gives when we re-index two terms on the RHS (and use the sum being over an infinite range, clearly this would not work if the range of the index r were bounded)

$$\hat{U}^{n+1}(k) \;=\; \hat{U}^n(k) + \mu\left(\mathrm{e}^{-ikh} - 2 + \mathrm{e}^{ikh}\right)\hat{U}^n(k) \tag{12.34}$$

so that

$$\hat{U}^{n+1}(k) \;=\; \hat{U}^n(k) - 4\mu\sin^2\frac{kh}{2}\hat{U}^n(k) \tag{12.35}$$

or with $\lambda(k) = 1 - 4\mu\sin^2(kh/2)$, we have

$$\hat{U}^{n+1}(k) \;=\; \lambda(k)\hat{U}^n(k). \tag{12.36}$$

Hence using Parseval's Identity

$$\|U^{n+1}\|_{\ell_2} \;=\; \frac{1}{\sqrt{2\pi}}\|\hat{U}^{n+1}\|_{L_2} = \frac{1}{\sqrt{2\pi}}\|\lambda\hat{U}^n\|_{L_2} \tag{12.37}$$

or

$$\|U^{n+1}\|_{\ell_2} \;\le\; \max_k|\lambda|\;\|U^n\|_{\ell_2}. \tag{12.38}$$

Hence $\max_k|\lambda(k)| \le 1$ (with $k \in [-\pi/h, \pi/h]$) is sufficient for practical stability, so that,

$$-1 \;\le\; 1 - 4\mu\sin^2\frac{kh}{2} \;\le\; 1 \tag{12.39}$$

gives practical stability (and as we shall show shortly, this is satisfied when $\mu \le 1/2$).

For most linear discrete schemes on the whole real line, analysis for stability proceeds by deriving

$$\hat{U}^{n+1}(k) \;=\; \lambda(k)\hat{U}^n(k) \tag{12.40}$$

and then determining conditions for which

$$\max_{k\in(-\pi/h,\pi/h)}|\lambda(k)| \le 1, \tag{12.41}$$

and the general experience is that even if the problem is not defined on the whole real line, ignoring the boundaries and extending the domain to the whole line and using a mode

$$U_r^n \sim \lambda^n\mathrm{e}^{ikrh} \tag{12.42}$$

and then applying (12.41) [or using a smi-discrete transform] can still give useful practical stability conditions.

Definition. *A finite difference scheme is* von Neumann stable *for* $0 \le t \le T_{\max}$ *if there exists* $C > 0$ *such that*

$$\|U^n\|_{\ell_2} \;\le\; C\|U^0\|_{\ell_2}. \tag{12.43}$$

This is a little less restrictive than practical stability (practical stability $\Rightarrow$ von Neumann stability.)

Lemma. *If $\hat{U}^{n+1}(k) = \lambda(k)\hat{U}^n(k)$ and $|\lambda(k)| \leq 1 + K\Delta t$ $\forall k \in [-\pi/h, \pi/h]$ then the scheme is von Neumann stable.*

Proof. Using Parseval's Identity

$$\|U^{n+1}\|_{\ell_2} \leq (1 + K\Delta t)\|U^n\|_{\ell_2} \tag{12.44}$$

so

$$\begin{aligned} \|U^n\|_{\ell_2} &\leq (1 + K\Delta t)^n \|U^0\|_{\ell_2} && (12.45) \\ &\leq \mathrm{e}^{nK\Delta t}\|U^0\|_{\ell_2} && (12.46) \end{aligned}$$

but for $0 \leq n\Delta t < T_{\max}$, the exponential is bounded and so

$$\|U^n\|_{\ell_2} \leq \mathrm{e}^{KT_{\max}}\|U^0\|_{\ell_2} \tag{12.47}$$

and the scheme is von Neumann stable. $\square$

Theorem: Lax Equivalence Theorem. *(without proof) For a consistent difference approximation to a well posed linear evolutionary problem, stability as $\Delta t \to 0$ is necessary and sufficient for convergence.*

This is the equivalent statement for PDEs to Dahlquist's theorem for linear multistep methods.

Lecture 13

Analysis of a Fully Implicit Scheme

The scheme just considered is an explicit scheme, we can derive an implicit scheme from

$$\left(\frac{\partial u}{\partial t}\right)_r^{n+1} \approx \frac{U_r^{n+1} - U_r^n}{\Delta t} \tag{13.1}$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_r^{n+1} \approx \frac{U_{r+1}^{n+1} - 2U_r^{n+1} + U_{r-1}^{n+1}}{h^2} \tag{13.2}$$

with a finite difference stencil as shown in Figure 18.

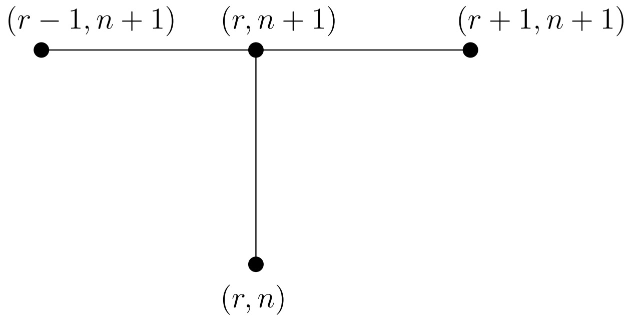


Figure 18: A finite difference stencil for the heat equation using an implicit scheme.

So now, letting $\mu = \Delta/h^2$

$$-\mu U_{r-1}^{n+1} + (1+2\mu)U_r^{n+1} - \mu U_{r+1}^{n+1} = U_r^n. \tag{13.3}$$

If we write this out for, say, $h = 1/4$

$$\begin{array}{rcrcrcrcrcl}
U_0^{n+1} & & & & & & & & & = & 0 \\
-\mu U_0^{n+1} & + & (1+2\mu)U_1^{n+1} & - & \mu U_2^{n+1} & & & & & = & U_1^n \\
 & - & \mu U_1^{n+1} & + & (1+2\mu)U_2^{n+1} & - & \mu U_3^{n+1} & & & = & U_2^n \\
 & & & - & \mu U_2^{n+1} & + & (1+2\mu)U_3^{n+1} & - & \mu U_4^{n+1} & = & U_3^n \\
 & & & & & & & & U_4^{n+1} & = & 0.
\end{array} \tag{13.4}$$

In matrix terms

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -\mu & 1+2\mu & -\mu & 0 & 0 \\ 0 & -\mu & 1+2\mu & -\mu & 0 \\ 0 & 0 & -\mu & 1+2\mu & -\mu \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U_0^{n+1} \\ U_1^{n+1} \\ U_2^{n+1} \\ U_3^{n+1} \\ U_4^{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ U_1^n \\ U_2^n \\ U_3^n \\ 0 \end{bmatrix} \tag{13.5}$$

This is a tridiagonal coefficient matrix, and for general M, gives a system that can be efficiently solved by the Thomas Algorithm.

We can in this model problem trivially eliminate boundary values and define

$$K = \begin{pmatrix} +2 & -1 & 0 & \dots & \dots & 0 \\ -1 & +2 & -1 & 0 & \dots & \dots \\ 0 & -1 & +2 & -1 & 0 & \dots \\ & & \ddots & \ddots & \ddots & \\ \dots & \dots & 0 & -1 & +2 & -1 \\ 0 & \dots & \dots & 0 & -1 & +2 \end{pmatrix}_{(M-1)\times(M-1)} \tag{13.6}$$

then we have:

Explicit Scheme

$$\mathbf{U}^{n+1} = \begin{bmatrix} U_1^{n+1} \\ \vdots \\ U_{M-1}^{n+1} \end{bmatrix} = (I - \mu K) \begin{bmatrix} U_1^{n} \\ \vdots \\ U_{M-1}^{n} \end{bmatrix} = (I - \mu K)\mathbf{U}^n. \tag{13.7}$$

Implicit Scheme

$$(I + \mu K)\mathbf{U}^{n+1} = \mathbf{U}^n. \tag{13.8}$$

The implicit scheme could be modified by using a combination of space derivatives at level n and $n+1$. Let $0 \le \theta \le 1$ and approximate the space deivative by a weighted sum of the values at time level n and level $n+1$:

$$\frac{\partial^2 u}{\partial x^2} \approx \theta \left(\frac{\partial^2 u}{\partial x^2}\right)^{n+1} + (1-\theta)\left(\frac{\partial^2 u}{\partial x^2}\right)^{n} \tag{13.9}$$

or

$$\frac{U_r^{n+1} - U_r^n}{\Delta t} = \theta \frac{U_{r+1}^{n+1} - 2U_r^{n+1} + U_{r-1}^{n+1}}{h^2} + (1-\theta)\frac{U_{r+1}^{n} - 2U_r^{n} + U_{r-1}^{n}}{h^2} \tag{13.10}$$

or in matrix terms

$$(I + \mu\theta K)\mathbf{U}^{n+1} = (I - \mu(1-\theta)K)\,\mathbf{U}^n, \tag{13.11}$$

and is a form of **theta method**.

Thomas Algorithm

Many numerical schemes which use three point stencils give tridiagonal matrix problems, for example, we have just seen that the implicit scheme

$$\frac{U_r^{n+1} - U_r^n}{\Delta t} = \frac{U_{r+1}^{n+1} - 2U_r^{n+1} + U_{r-1}^{n+1}}{h^2}, \tag{13.12}$$

results in the matrix problem:

$$(I+\mu K)\mathbf{U}^{n+1} \;=\; \mathbf{U}^n. \tag{13.13}$$

This has the matrix (with boundary values eliminated) of the general form

$$\begin{array}{ccccccccc} b_1U_1^{n+1} & - & c_1U_2^{n+1} & & & & & = & U_1^n \\ -a_2U_1^{n+1} & + & b_2U_2^{n+1} & - & c_2U_3^{n+1} & & & = & U_2^n \\ & & \ddots & & \ddots & & & & \\ & & & - & a_{M-1}U_{M-2}^{n+1} & + & b_{M-1}U_{M-1}^{n+1} & = & U_{M-1}^n \end{array} \tag{13.14}$$

So we can state a fairly common matrix representation

$$\begin{array}{ccccccccccc} b_1U_1 & - & c_1U_2 & & & & & & & = & d_1 \\ -a_2U_1 & + & b_2U_2 & - & c_2U_3 & & & & & = & d_2 \\ & - & a_3U_2 & + & b_3U_3 & - & c_3U_4 & & & = & d_3 \\ & & & & \ddots & & & & & & \\ & & & & & - & a_{M-1}U_{M-2} & + & b_{M-1}U_{M-1} & = & d_{M-1} \end{array} \tag{13.15}$$

If we normalise the first row and eliminate the a_2 term in the second row

$$\begin{array}{ccccccccc} U_1 & - & \frac{c_1}{b_1}U_2 & & & & & = & \frac{d_1}{b_1} \\ 0 & - & \left(b_2-\frac{a_2}{b_1}c_1\right)U_2 & - & c_3U_3 & & & = & d_2+\frac{a_2}{b_1}d_1 \\ 0 & - & a_3U_2 & + & b_3U_3 & - & c_3U_4 & = & d_3 \end{array} \tag{13.16}$$

etc. It is a class exercise to show that

$$\begin{array}{ccccccc} U_1 & - & e_1U_2 & & & = & f_1 \\ & & U_2 & - & e_2U_3 & = & f_2 \\ & & & \ddots & & & \\ & & & U_{M-2} & - \; e_{M-2}U_M & = & f_{M-2} \\ & & & & U_{M-1} & = & f_{M-1} \end{array} \tag{13.17}$$

where $e_0 = 0$ and $f_0 = 0$ and

$$e_r \;=\; \frac{c_r}{b_r - a_re_{r-1}}, \quad r = 1,\ldots,M-2 \tag{13.18}$$

$$f_r \;=\; \frac{d_r + a_rf_{r-1}}{b_r - a_re_{r-1}}, \quad r = 1,\ldots,M-1. \tag{13.19}$$

Then with $U_{M-1} = f_{M-1}$

$$U_{M-2} \;=\; f_{M-2} + e_{M-2}U_{M-1} \tag{13.20}$$

or

$$U_r \;=\; f_r + e_rU_{r+1} \qquad \text{for } r = M-2, M-3, \ldots, 1. \tag{13.21}$$

From an implementaiton point of view, note that the Thomas algorithm does not require the full matrix to be stored, only four vectors **a**, **b**, **c**, **d**, so it is perfectly feasible to handle problems where M is very large and where the full $M-1\times M-1$ coefficient matrix could not possibly be stored.

(i) Truncation error: Implicit Scheme

If we turn to the fully implicit scheme

$$-\mu U_{r-1}^{n+1} + (1+2\mu)U_r^{n+1} - \mu U_{r+1}^{n+1} \;=\; U_r^n, \tag{13.22}$$

and evaluate the truncation error by expanding about $(rh, (n+1)\Delta t)$

$$T_r^{n+1} \;=\; \frac{1}{2}\Delta t \frac{\partial^2 u}{\partial t^2}(rh, \hat{\eta}_{n+1}) + \frac{1}{12}h^2 \frac{\partial^4 u}{\partial x^4}(\hat{\xi}_{r+1}, t_{n+1}), \tag{13.23}$$

where $\hat{\eta}_{n+1} \in (t_n, t_{n+1})$ and $\hat{\xi}_{r+1} \in (x_{r-1}, x_{r+1})$. Hence the order of the scheme is the same as for the explicit scheme, that is first order in time and second order in space.

(ii) Convergence: Implicit Scheme

As before, let $e_j^n = u_j^n - U_j^n$ to obtain

$$\frac{e_r^{n+1} - e_r^n}{\Delta t} - \frac{e_{r+1}^{n+1} - 2e_r^{n+1} + e_{r-1}^{n+1}}{h^2} \;=\; T_r^{n+1} \tag{13.24}$$

so that

$$(1+2\mu)e_r^{n+1} \;=\; e_r^n + \mu e_{r+1}^{n+1} + \mu e_{r-1}^{n+1} - \Delta t T_r^n. \tag{13.25}$$

Let $E^n = \max_r |e_r^n|$, then

$$(1+2\mu)|e_r^{n+1}| \;\leq\; E^n + \mu E^{n+1} + \mu E^{n+1} + \Delta t |T_r^{n+1}| \tag{13.26}$$

for all r, so that we must also have

$$(1+2\mu)E^{n+1} \;\leq\; E^n + \mu E^{n+1} + \mu E^{n+1} + \Delta t |T^{n+1}| \tag{13.27}$$

where $T^{n+1} = \max_r T_r^{n+1}$, and hence

$$E^{n+1} \;\leq\; E^n + \Delta t |T^{n+1}|. \tag{13.28}$$

Provided $E^0 = 0$, then with $T = \max_{r,n} |T_r^n| \leq C\Delta t$ we have

$$|E^{n+1}| \;\leq\; (n+1)\Delta t T \;=\; t_{n+1} C \Delta t \tag{13.29}$$

and if $t_n \to t$ as $\Delta t \to 0$, $n \to \infty$

$$|E^{n+1}| \;\leq\; tC\Delta t \;\to\; 0. \tag{13.30}$$

However, by definition $|e_r^{n+1}| \leq E^{n+1}$ so we must have $|e_r^{n+1}| \to 0$ as $\Delta t \to 0$, $n \to \infty$, $(n+1)\Delta t \to t$, and so as for the explicit scheme, we deduce convergence, only now the scheme converges unconditionally.

(iii) Stability: Implicit Scheme

As for the explicit scheme, we can either ise a semi-discrete transform, multiply the discrete equation by $h\mathrm{e}^{\imath krk}$ and sum, reindex, to obtain

$$[1+\mu(2-\mathrm{e}^{\imath kh}-\mathrm{e}^{-\imath kh})]\hat{U}^{n+1}(k) \;=\; \hat{U}^{n}(k), \tag{13.31}$$

or

$$(1+4\sin^2\frac{1}{2}kh)\hat{U}^{n+1}(k) \;=\; \hat{U}^{n}(k), \tag{13.32}$$

so that

$$\lambda \;=\; \frac{1}{1+4\mu\sin^2\frac{kh}{2}}, \tag{13.33}$$

or we can take a mode, $U_r^n \sim \lambda^n \mathrm{e}^{ikrh}$, substitute into the discretisation as before, cancel common factors, to obtain

$$\lambda\left(1+2\mu-\mu\mathrm{e}^{ikh}-\mu\mathrm{e}^{-ikh}\right) \;=\; 1 \tag{13.34}$$

so giving exactly the same result.

The structure of $\lambda = 1/(1+\mathrm{non-negative})$ means that $|\lambda| \leq 1$ for all k and the scheme is unconditionally stable.

As for the explicit method, we can see that the numerical amplification factor $\lambda(k)$ approximates the continuous amplification factor $\Lambda(k)$very well for small values of k but much less well for large values (but of course, the mode aplictude of the solutons decreases rapidly as k increases so the failure of λ to match Λ for large k may have very little effect on the computed solution. The comparison between the tow amplification factors is shown in figure 19.

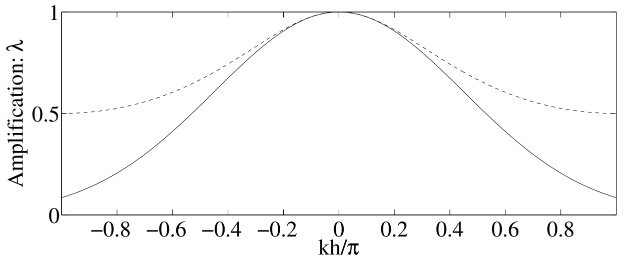


Figure 19: Comparison of amplification factors Λ (continuous diffusion equation) and λ (implicit Euler method) as the wave number kh varies in $[-\pi, \pi]$ for the case $\mu = 0.25$. (—-): Λ, (- - -) λ.

Lecture 14

Analysis of a Theta-method

As before, let $\mu = \Delta t/h^2$ and assume $0 \le \theta \le 1$, then a *theta method* will relate values at level n and $n+1$ according to

$$\frac{U_r^{n+1} - U_r^n}{\Delta t} = (1-\theta)\frac{U_{r+1}^n - 2U_r^n + U_{r-1}^n}{h^2} + \theta\frac{U_{r+1}^{n+1} - 2U_r^{n+1} + U_{r-1}^{n+1}}{h^2} \tag{14.1}$$

so that

$$\begin{aligned} -\theta\mu U_{r-1}^{n+1} + (1+2\theta\mu)U_r^{n+1} &- \theta\mu U_{r+1}^{n+1} = \\ (1-\theta)\mu U_{r-1}^n &+ (1-2(1-\theta)\mu)U_r^n + (1-\theta)\mu U_{r+1}^n, \end{aligned} \tag{14.2}$$

.

(i)Truncation Error of θ-method

While Taylor series expansion can be about a number of different points, the simplest expansion for a theta method is about the mid point $(x_r, t_{n+1/2})$ since symmetry helps with cancelation of some terms.

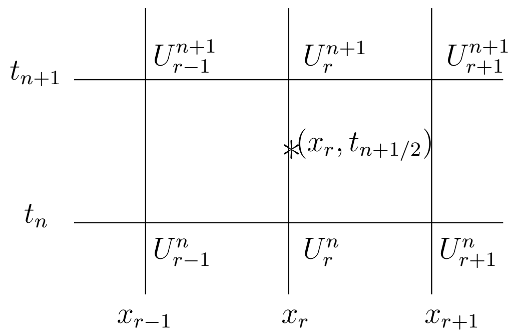


Figure 20: Grid points for the theta method.

We expand terms in the truncation error about $(x_r, t_{n+1/2})$ (see Figure 20):

$$u_r^{n+1} = \left[u + \frac{1}{2}\Delta t u_t + \frac{1}{2}\left(\frac{1}{2}\Delta t\right)^2 u_{tt} + \frac{1}{6}\left(\frac{1}{2}\Delta t\right)^3 u_{ttt} + \dots\right]_r^{n+1/2} \tag{14.3}$$

$$u_r^n = \left[u - \frac{1}{2}\Delta t u_t + \frac{1}{2}\left(\frac{1}{2}\Delta t\right)^2 u_{tt} - \frac{1}{6}\left(\frac{1}{2}\Delta t\right)^3 u_{ttt} + \dots\right]_r^{n+1/2} \tag{14.4}$$

so that

$$\frac{u_r^{n+1}-u_r^n}{\Delta t} = \left[u_t + \frac{1}{24}\Delta t^2 u_{ttt} + \dots\right]_r^{n+1/2}. \tag{14.5}$$

We also know that

$$A = \frac{u_{r+1}^{n+1}-2u_r^{n+1}+u_{r-1}^{n+1}}{h^2} = \left[u_{xx} + \frac{1}{12}h^2 u_{xxxx} + \dots\right]_r^{n+1} \tag{14.6}$$

$$B = \frac{u_{r+1}^{n}-2u_r^{n}+u_{r-1}^{n}}{h^2} = \left[u_{xx} + \frac{1}{12}h^2 u_{xxxx} + \dots\right]_r^{n} \tag{14.7}$$

and we expand each of these about $(x_r, t_{n+1/2})$ to get

$$A = \left[u_{xx} + \frac{1}{2}\Delta t u_{xxt} + \frac{1}{2}\left(\frac{1}{2}\Delta t\right)^2 u_{xxtt} + \dots\right]_r^{n+1/2} \tag{14.8}$$

$$+\frac{1}{12}h^2\left[u_{xxxx} + \frac{1}{2}\Delta t u_{xxxxt} + \dots\right]_r^{n+1/2} + \dots \tag{14.9}$$

$$B = \left[u_{xx} - \frac{1}{2}\Delta t u_{xxt} + \frac{1}{2}\left(\frac{1}{2}\Delta t\right)^2 u_{xxtt} + \dots\right]_r^{n+1/2} \tag{14.10}$$

$$+\frac{1}{12}h^2\left[u_{xxxx} - \frac{1}{2}\Delta t u_{xxxxt} + \dots\right]_r^{n+1/2} + \dots \tag{14.11}$$

so that

$$\theta A + (1-\theta)B = u_{xx} + \frac{1}{12}h^2 u_{xxxx} + \frac{2}{6!}h^4 u_{xxxxxx} + \dots \tag{14.12}$$

$$+\frac{1}{2}\left(\theta - (1-\theta)\right)\Delta t\left[u_{xxt} + \frac{1}{12}h^2 u_{xxxxt} + \dots\right] + \frac{1}{8}\Delta t^2 u_{xxtt} + \dots \tag{14.13}$$

so that

$$T_r^{n+1/2} = \frac{1}{2}(1-2\theta)\Delta t u_{xxt} - \frac{1}{12}h^2 u_{xxxx} + \Delta t^2\left[\frac{1}{24}u_{ttt} - \frac{1}{8}u_{xxtt}\right] + \dots \tag{14.14}$$

If we use $u_t = u_{xx}$, $u_{tt} = u_{xxxx}$ etc then

$$T_r^{n+1/2} = \frac{1}{2}(1-2\theta)\Delta t u_{xxxx} - \frac{1}{12}h^2 u_{xxxx} - \frac{1}{12}\Delta t^2 u_{xxxxxx} + \dots. \tag{14.15}$$

The case $\theta = 1/2$ is special in that the first term is zero and so the scheme is second order in space and time. This is called the *Crank-Nicholson* scheme.

(ii) Convergence: Theta-method

For convergence of the θ method, let $1-\theta \geq 0$ *and* $1-2(1-\theta) \geq 0$.

Combining the definition of the scheme and the truncation error then the errors for the scheme much satisfy

$$\begin{aligned}(1 \;&+\; 2\mu\theta)e_r^{n+1} = \mu\theta(e_{r-1}^{n+1} + e_{r+1}^{n+1}) \\ &+\; \mu(1-\theta)(e_{r-1}^n + e_{r+1}^n) + (1-2\mu(1-\theta))e_r^n - \Delta t T_r^{n+1/2}, \end{aligned} \tag{14.16}$$

and letting $E^n = \max|e_r^n|$ and $T^{n+1/2} = \max|T_r^{n+1/2}|$ and since $1-\theta \geq 0$ and $1-2(1-\theta) \geq 0$ we have

$$(1+2\mu\theta)E^{n+1} \leq 2\mu\theta E^{n+1} + 2\mu(1-\theta)E^n + (1-2\mu(1-\theta))E^n + \Delta t T^{n+1/2}. \tag{14.17}$$

and thus

$$E^{n+1} \leq E^n + \Delta t T^{n+1/2}. \tag{14.18}$$

Starting with exact initial data, $E^0 = 0$,

$$E^n \leq \Delta t \sum_{p=0}^{n-1} T^{p+1/2} \leq n\Delta t \max T^{n+1/2} \tag{14.19}$$

and $n\Delta t \leq t_{\max}$ so that $E^n \to 0$ as $T^{n+1/2} \to 0$ with $\Delta t \to 0$ and the scheme is convergent.

(iii) Stability: Theta-method

As in the two previous examples, we obtain exactly the same result using either a semi-discrete transofrom or by substituting a Fourier mode and you should be able to do both if asked in an examination. Here, we apply analysis using a Fourier mode

$$U_r^n \sim \lambda^n e^{\imath krh},$$

which gives after cancellation of common factors

$$\left(1 + 4\theta\mu\sin^2\frac{kh}{2}\right)\lambda = 1 - 4(1-\theta)\mu\sin^2\frac{kh}{2} \tag{14.20}$$

so that

$$\lambda = \frac{1-4(1-\theta)\mu\sin^2\frac{kh}{2}}{1+4\theta\mu\sin^2\frac{kh}{2}} \tag{14.21}$$

or, with $p = \sin^2(kh/2)$

$$\lambda \;=\; \frac{1 - 4(1-\theta)\mu p}{1 + 4\theta\mu p} \tag{14.22}$$

where $\mu > 0$ and $\theta \in [0, 1]$. This is a monotonically decreasing function of p and our interest is for $0 \le p \le 1$. If $p \to \infty$ then $\lambda \to (\theta - 1)/\theta$ so we have two cases as shown in Figure 21 depending on the relation of this limit to minus one.

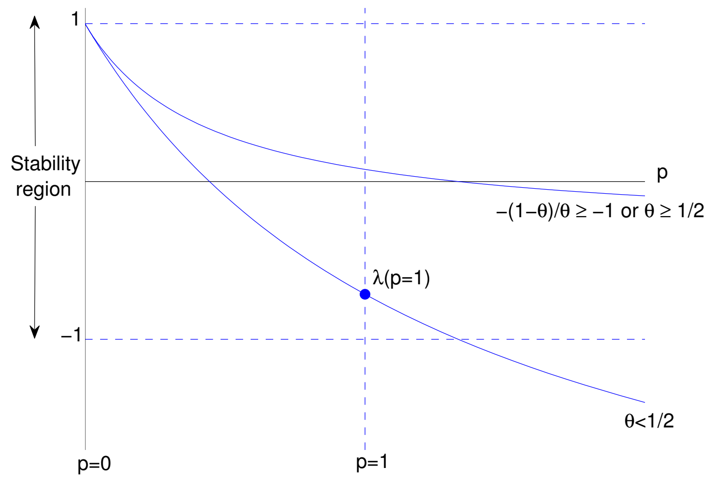


Figure 21: Stability region for the theta method.

Provided the large p limit of λ is greater than or equal to -1, the scheme is guaranteed to be stable since the value of λ must be between ± 1. This case corresponds to $\theta \ge 1/2$.

In the case $\theta < 1/2$ we need to check that $\lambda(p = 1) \ge -1$ for the scheme to be stable. We need

$$\lambda(p = 1) \;=\; \frac{1 - 4(1-\theta)\mu}{1 + 4\theta\mu} \;\ge\; -1 \tag{14.23}$$

or

$$\mu(1 - 2\theta) \;\le\; \frac{1}{2}, \quad \theta < \frac{1}{2}. \tag{14.24}$$

So for stability when $\theta < 1/2$, we must have

$$\mu \;\le\; \frac{1}{2(1 - 2\theta)}. \tag{14.25}$$

As for explict and fuly implicit Euler methods, we can calculate values of the amplification factor *lambda* and compare with Λ; the case $\theta = 1/2$ is known as Crank–Nicholson and values are shown in figure 22 for the same mesh ratio as before,

$\mu = 0.25$. As with the previous methods the agreement for small k is excellent, agreement for larger k is less good but as observed, this has little impact on the calculated solution as the Fourier transform of the original distirbution of u at the start is highly likely to have very small coefficients at large k.

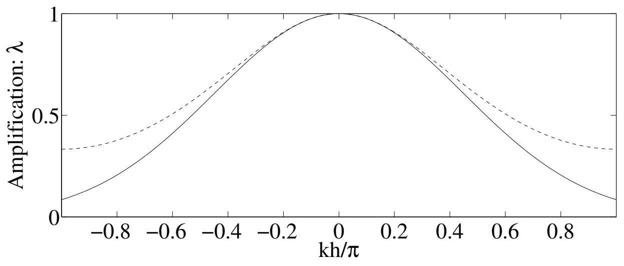


Figure 22: Comparison of amplification factors Λ (continuous diffusion equation) and λ (Crank Nicholson, $\theta = 1/2$) as the wave number kh varies in $[-\pi, \pi]$ for the case $\mu = 0.25$. (—-): Λ, (- - -) λ.

We can see the effect of the amplification values if the solution of diffusion problem is computed. One simple model problem is that already anlysed, that of a delta function distribution in space as initial value with exact solution given by (11.22). It is not practical to start this calculation at $t = 0$, however, the exact solution can be specified at $t = \Delta t$ and the evolution computed to some later time. All three methods give nearly identical computed values at later times, see figure 23 where the initial spike is partially shown and all three methods overlay the exact solution very closely.

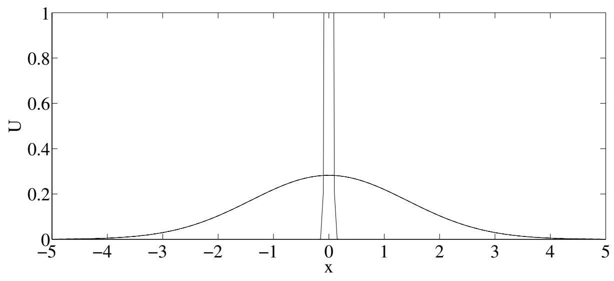


Figure 23: Comparison of computed solutions for a delta function initial distribution, calculated for $\mu = 0.25$, $h = 0.05$ starting from the exact solution at $t = \Delta t$ and computing until $t = 1$ for explicit Euler, implicit Euler and Crank–Nicholson. On this visible resolution all three computed solutions are identical.

The error between the computed solutions and the exact solution are not apparent in figure 23, they are show in a semi-log plot in figure 24, all three methods give errors of similar magnitude (even though Crank-Nicholson has higher order convergence in h the errors are much the same and dominated by all three methods being first order in Δt.

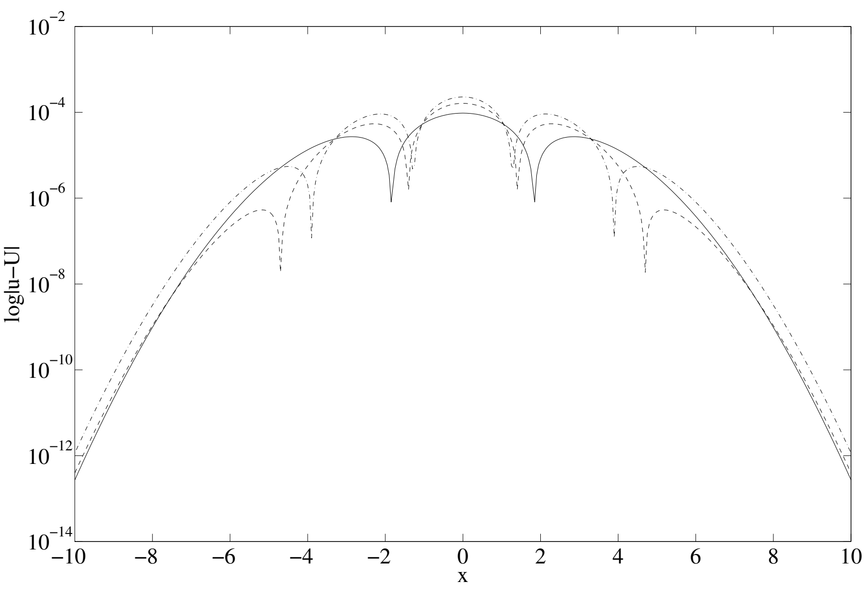


Figure 24: Comparison of errors at $t = 1$ for $\mu = 0.25$, $h = 0.05$ for (i) (—-) Explicit Euler, (ii) (- - -) Fully Implicit Euler, (iii) (- . -) Crank–Nicholson.

You should appreciate that any limit on μ is a severe practical constraint and for the explicit scheme we have in terms of time and space step

$$\Delta t \leq \frac{1}{2}h^2, \tag{14.26}$$

so, for example, if we were to reduce the space step by a factor of two, not only would we have twices as many space computations each time step, we would also need to reduce the time step by a factor of four; halving the mesh size leads to eight time as many computations. If, on the other hand, we used an implicit scheme so as to avoid a stablity constraint, halving the mesh to increase accuracy would lead to a matrix problem where the coefficient matrix has doubled in size, and inversion of this matrix requires a similar increase in number of computations (that is, roughly a factor of eight) although very efficient matrix solvers can have a big effect here.

The Leapfrog Method

The methods we have looked at so far are all first order in time. One possibility to increase temporal accuracy is to use central differences in time as well as in space. This adds a level of complexity since data has to be stored for two previous time steps, giving the three level leapfrog method (see Figure 25 for the finite difference stencil):

$$\left(\frac{\partial u}{\partial t}\right)_r^n \approx \frac{U_r^{n+1} - U_r^{n-1}}{2\Delta t} \tag{14.27}$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_r^n \approx \frac{U_{r+1}^n - 2U_r^n + U_{r-1}^n}{h^2} \tag{14.28}$$

$$U_r^{n+1} = U_r^{n-1} + 2\mu\left(U_{r+1}^n - 2U_r^n + U_{r-1}^n\right). \tag{14.29}$$

It is a simple exercise to show that $T_r^n \sim \mathcal{O}(\Delta t^2, h^2)$.

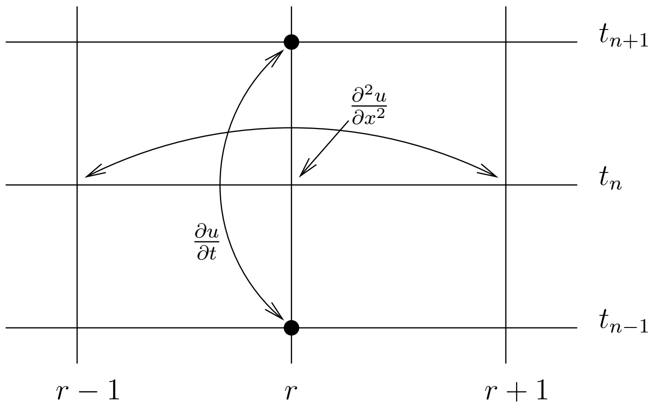

t_{n+1}
$\frac{\partial^2 u}{\partial x^2}$
t_n
$\frac{\partial u}{\partial t}$
t_{n-1}
$r-1$
r
$r+1$

Figure 25: The finite difference stencil for the leapfrog method

If we look at Fourier modes $U_r^n \sim \lambda^n \mathrm{e}^{ikrh}$ then

$$\lambda = \frac{1}{\lambda} + 2\mu(2\cos(kh) - 2) \tag{14.30}$$

$$= \frac{1}{\lambda} - 8\mu \sin^2 \frac{kh}{2}. \tag{14.31}$$

Let $p = \sin^2(kh/2)$ then

$$\lambda^2 + 8\mu p\lambda - 1 = 0 \tag{14.32}$$

with $0 \leq p \leq 1$. The roots are

$$\lambda_{1,2} = -4\mu p \pm \sqrt{16\mu^2 p^2 + 1} \tag{14.33}$$

so while

$$\lambda_1 = \sqrt{16\mu^2 p^2 + 1} - 4\mu p < 1 \tag{14.34}$$

the other root

$$\lambda_2 = -\sqrt{16\mu^2 p^2 + 1} - 4\mu p < -1 \tag{14.35}$$

so the scheme will always be *unstable*.

Lecture 15

Eigenvalue analysis

The use of individual Fourier modes $U_r^n \sim \lambda^n e^{ikrh}$ or the discrete Fourier transform, $\hat{U}^n$ is based on an infinite x domain. If the domain is finite, analysis is more difficult. It may be that the domain can be extended to an infinite domain by making the solution periodic in space. It is also sometimes possible to apply an eigenvalue analysis of the finite domain approximation. We give one example here.

If we take the domain $0 \le x \le 1$ with $u(0,t) = u(1,t) = 0$ and apply an explicit method we saw that the matrix

$$K = \begin{pmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & \dots \\ 0 & -1 & 2 & -1 & 0 & \dots \\ & & \ddots & \ddots & \ddots & \\ \dots & \dots & 0 & -1 & 2 & -1 \\ 0 & \dots & \dots & 0 & -1 & 2 \end{pmatrix}_{(M-1)\times(M-1)} \tag{15.1}$$

was central to the scheme with

$$\mathbf{U}^{n+1} = (I - \mu K)\mathbf{U}^n. \tag{15.2}$$

K is an $(M-1)\times(M-1)$ matrix.

Suppose the eigenvalues of K are denoted $\lambda_1, \dots, \lambda_{M-1}$ and eigenvectors $\mathbf{z}^1, \dots, \mathbf{z}^{M-1}$. Consider

$$\mathbf{z}^p = \begin{pmatrix} \sin(p\pi h) \\ \sin(2p\pi h) \\ \vdots \\ \sin(rp\pi h) \\ \vdots \\ \sin((M-1)p\pi h) \end{pmatrix} \tag{15.3}$$

Note also that as $M = 1/h$ then $\sin(Mp\pi h) = \sin(p\pi) = 0$. Now evaluate $K\mathbf{z}^p$

$$K\mathbf{z}^p = \begin{pmatrix} 0 & + & 2\sin(p\pi h) & - & \sin(2p\pi h) \\ -\sin(p\pi h) & + & 2\sin(2p\pi h) & - & \sin(3p\pi h) \\ & & \vdots & & \\ -\sin((r-1)p\pi h) & + & 2\sin(rp\pi h) & - & \sin((r+1)p\pi h) \\ & & \vdots & & \\ -\sin((M-2)p\pi h) & + & 2\sin((M-1)p\pi h) & - & \sin(Mp\pi h) \end{pmatrix} \tag{15.4}$$

but

$$\begin{aligned}\sin((r-1)p\pi h) &= \sin(rp\pi h)\cos(p\pi h) - \cos(rp\pi h)\sin(p\pi h) && (15.5)\\ \sin((r+1)p\pi h) &= \sin(rp\pi h)\cos(p\pi h) + \cos(rp\pi h)\sin(p\pi h) && (15.6)\end{aligned}$$

and so

$$\begin{aligned}-\sin((r-1)p\pi h) + 2\sin(rp\pi h) &- \sin((r+1)p\pi h)\\ &= \underbrace{(2-2\cos(p\pi h))}_{\text{factor same for each } r}\sin(rp\pi h). && (15.7)\end{aligned}$$

Hence

$$K\mathbf{z}^p = (2-2\cos(p\pi h))\mathbf{z}^p = 4\sin^2\frac{p\pi h}{2}\mathbf{z}^p \qquad (15.8)$$

and the eigenvalues of K are $\lambda_p = 4\sin^2(p\pi h/2)$. Hence the eigenvalue of $I-\mu K$ are $1-4\mu\sin^2(p\pi h/2)$ and if the eigenvalues of $I-\mu K$ are all to have modulus less than or equal to 1, we need $\mu \leq 1/2$ as before.

Maximum principle analysis

In some situations it is not possible to apply any Fourier type analysis, often becasue the domain shape does not allow either Fourier modes or matrix eigenvalue analysis. It may still be possible bound how the the solution can evolve by showing that there is a maximum principle that constrains any calculation in the numerical solution to be between maximum and minimum values on the boundary of the computational region..

We first look at a relatively trivial result from linear interpolation, this is worth doing as the undelying argument is at the heart of proof of maximum principles. The lemma is just that for linear interpolation, the interpolated value will always lay between the smaller and the larger of the two values being interpolated.

Lemma. *Let $U = \alpha V + (1-\alpha)W$ for $0 \leq \alpha \leq 1$ then $U \leq \max\{V, W\}$.*

Proof. Let $M = \max\{V, W\}$ and write $V = M - \epsilon_1$, $W = M - \epsilon_2$ for some $\epsilon_1, \epsilon_2 \geq 0$ and

$$\begin{aligned}U &= \alpha(M-\epsilon_1) + (1-\alpha)(M-\epsilon_2) && (15.9)\\ &= M - \alpha\epsilon_1 - (1-\alpha)\epsilon_2 && (15.10)\\ &\leq M. && (15.11)\end{aligned}$$

□

Lemma. *Extension*

For data $\{V_1, \ldots, V_k\}$, let

$$U = \sum_{r=1}^{k} \alpha_r V_r, \tag{15.12}$$

where $\alpha_1, \ldots, \alpha_k$ satisfy

1. *$\alpha_r \geq 0$ for $r = 1, \ldots, k$,*
2. *$\sum_{r=1}^{k} \alpha_r = 1$,*

then

$$\min_r \{V_r\} \;\; \leq \;\; U = \sum_{r=1}^{k} \alpha_r V_r \;\; \leq \;\; \max_r \{V_r\}. \tag{15.13}$$

In the case of the finite difference schemes we develop, if we can write

$$\beta U_r^{n+1} \quad = \sum_{s \in \text{adjacent nodes}} \alpha_s U_s^{n,n+1} \tag{15.14}$$

where

1. $\alpha_r \geq 0$,
2. $\beta > 0$,
3. $\beta = \sum \alpha_r$,

then

$$\min \{\text{adjacent nodes}\} \leq U_r^{n+1} \leq \max \{\text{adjacent nodes}\}. \tag{15.15}$$

Maximum Principle for θ-method for Diffusion Equation. *Suppose on the boundaries*

$$t = 0, \quad 0 \leq x \leq 1; \quad x = 0, \quad t > 0; \quad x = 1, \quad t > 0,$$

Dirichlet data is given by:

$$U_0^n, \quad U_M^n, \quad n = 1, 2 \ldots N = \frac{T_{\max}}{\Delta t} \tag{15.16}$$

$$U_r^0 \qquad r = 0, \ldots, M, \text{ with } h = \frac{1}{M}. \tag{15.17}$$

Let

$$\begin{aligned}\partial U_{\max} &= \max\left\{U_0^n, U_M^n, n=1,\dots,N; U_r^0, r=0,\dots,M\right\}, &(15.18)\\ \partial U_{\min} &= \min\left\{U_0^n, U_M^n, n=1,\dots,N; U_r^0, r=0,\dots,M\right\}, &(15.19)\end{aligned}$$

then the θ method for $0 \le \theta \le 1$ satisfies

$$\partial U_{\min} \le U_r^n \le \partial U_{\max}, \qquad (15.20)$$

provided

$$\mu(1-\theta) \le 1/2. \qquad (15.21)$$

Proof. Let $M = \max_{r,n} U_r^n$ and suppose M is achieved at the location $(r, n+1)$ in the interior of the computation region.

Suppose $U_{r-1}^{n+1} = M - \epsilon_1$, $U_{r+1}^{n+1} = M - \epsilon_2$, $U_{r-1}^n = M - \epsilon_3$, $U_r^n = M - \epsilon_4$, and $U_{r+1}^n = M - \epsilon_5$, where $\epsilon_1, \dots, \epsilon_5 \ge 0$

The θ scheme is

$$\begin{aligned}(1+2\mu\theta)U_r^{n+1} &=\\ \mu\theta(U_{r-1}^{n+1} &+ U_{r+1}^{n+1}) + \mu(1-\theta)(U_{r-1}^n + U_{r+1}^n) + (1-2\mu(1-\theta))U_r^n \quad (15.22)\end{aligned}$$

and we have $1-\theta \ge 0$ and $1 - 2(1-\theta)\mu \ge 0$. Thus

$$\begin{aligned}(1+2\mu\theta)M = (1 &+ 2\mu\theta)M\\ &+ \mu\theta(\epsilon_1+\epsilon_2) + \mu(1-\theta)(\epsilon_3+\epsilon_5) + (1-2\mu(1-\theta))\epsilon_4 \quad (15.23)\end{aligned}$$

or

$$\mu\theta(\epsilon_1+\epsilon_2) + \mu(1-\theta)(\epsilon_3+\epsilon_5) + (1-2\mu(1-\theta))\epsilon_4 = 0. \qquad (15.24)$$

This is only possible if $\epsilon_1 = \epsilon_2 = \epsilon_3 = \epsilon_4 = \epsilon_5 = 0$ and hence all points at $(n+1, r-1)$, $(n+1, r+1)$, $(n, r-1)$, (n, r), and $(n, r+1)$ must also have value M. Continue this process at neighbouring x-locations, $r-1, r+1$ or at time step n until the boundary (either the boundaries at $x = 0, 1$ or the initial value boundary at $t = 0$) is reached so M must occur on the boundary (and indeed M cannot be achieved in the interior unless $U_r^n = M$ everywhere).

A similar argument for $\bar{M} = \min_{r,n} U_r^n$, assume minimum in the interior, let adjacent points be $\bar{M} + \epsilon_i$ with $\epsilon_i \ge 0$ and deduce $\epsilon_i = 0$. □

Lemma. *Provided $2\mu(1-\theta) \le 1$, then θ-method for our model diffusion equation (defined on the whole real line) satisfies*

$$\|U^n\|_{\ell_\infty} \le \|U^0\|_{\ell_\infty}, \quad n = 1, 2, \dots,$$

where the infinity norm of a discrete set of data $\{U_r : r \in \mathbb{Z}\}$ is

$$\|U\|_{\ell_\infty} = max_{r\in\mathbb{Z}}|U_r|.$$

Proof. This result follows from the previous theorem, alternately this can be proved in two steps, first establish that

$$\|U^{n+1}\|_{\ell_\infty} \leq \|U^n\|_{\ell_\infty}, \quad n = 0, 1, \ldots, \tag{15.25}$$

from which it follows that

$$\|U^n\|_{\ell_\infty} \leq \|U^0\|_{\ell_\infty}, \quad n = 1, 2, \ldots.$$

The proof of (15.25) may be from the same argument as for the maximum principle (that is by contradiction) or by taking a norm and using a triangle inequality to obtain

$$\begin{aligned}(1+2\mu\theta)\|U^{n+1}\|_\infty &\leq \\ \mu\theta(\|U^{n+1}\|_\infty &+ \|U^{n+1}\|_\infty) + \mu(1-\theta)(\|U^n\|_\infty + \|U^n\|_\infty) + (1-2\mu(1-\theta))\|U^n\|_\infty\end{aligned}$$

provided $1-\theta \geq 0$ and $1-2(1-\theta)\mu \geq 0$, so that

$$\|U^{n+1}\|_{\ell_\infty} \leq \|U^n\|_{\ell_\infty}.$$

$\square$

An advantage of using a maximum principle is that the method can be applied both for a finite domain or in the context of our original problem on the whole real line where the solutions has to decay to zero when $|x| \to \infty$, and so we can prove stability in the ℓ_∞ norm (which is a much stronger norm than the ℓ_2 norm).

The condition $\mu(1-\theta) \leq 1/2$ or $\Delta t \leq h^2/2(1-\theta)$ is more restrictive than the stability condition $\mu \leq 1/2(1-2\theta)$. This is illustrated in Figure 26.

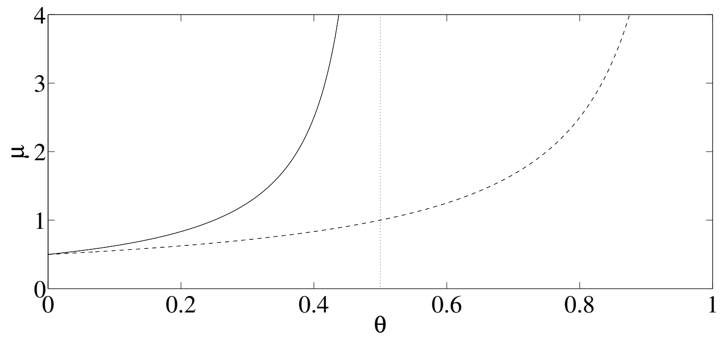


Figure 26: Stability limits on $\mu = \Delta t/h^2$ from ℓ_2 analysis, (—-), and from ℓ_∞ analysis or a maximum principle, (- - -). In each case, the numerical scheme is stable for μ values *up to and below* the relevant bounding curve.

It is worth being clear about how figure 26 should be interpreted. The limits show where we can prove stability of the numerical scheme for our model problem using either, for the whole real line, a semi-discrete or modal analysis and the ℓ_2 norm, or for a finite or infinite space domain, a maximum principle and the ℓ_∞ norm. In the case of semi-discrete transform or modal analysis, if the stability condition on μ is exceeded, then there will be growing modes and the nuerical solution will become unbounded. In the case where a maximum principle is used for a finite interval, we only claim stability in the ℓ_∞ norm for μ values below a critical value but we do not say anything about what should happen for μ values above this stability value.

Boundary conditions

On the boundaries of the interval $0 \le x \le 1$, we have so far considered only zero Dirichlet conditions $u(0,t) = u(1,t) = 0$. Suppose there were Neumann or mixed Robin boundary conditions, for example,

$$\begin{aligned} \text{Neumann} \qquad \frac{\partial u}{\partial x}(0,t) &= f(t) \qquad (15.26) \\ \text{Robin} \qquad \alpha u(0,t) + \beta \frac{\partial u}{\partial x}(0,t) &= g(t) \quad \beta \neq 0. \qquad (15.27) \end{aligned}$$

These can be handled best by applying an implicit discrete boundary condition. Suppose we temporarily assume that a value U_{-1} exists, the second derivative u_{xx} at the boundary has discrete approximation

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_0^n \doteqdot \frac{U_1^n - 2U_0^n + U_{-1}^n}{h^2} \qquad (15.28)$$

and the boundary condition can be discretised by

$$\alpha u + \beta \left. \frac{\partial u}{\partial x} \right|_0^n = \alpha U_0^n + \beta \frac{U_1^n - U_{-1}^n}{2h} = g^n. \qquad (15.29)$$

Rearranging the boundary condition (15.29) we have

$$U_{-1}^n = \frac{2\alpha h U_0^n + \beta U_1^n - 2hg^n}{\beta} \qquad (15.30)$$

and substituting this into (15.28) gives

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_0^n \approx \frac{U_1^n - 2U_0^n + U_1^n + \frac{2\alpha h}{\beta} U_0^n - \frac{2hg^n}{\beta}}{h^2}. \qquad (15.31)$$

Thus the ficticious point U_{-1}^n can be eliminated algebraically and we have a discretisation that implicitly applies the mixed boundary condition and the differential equation at the boundary point

$$\frac{U_0^{n+1} - U_0^n}{\Delta t} = \frac{2U_1^n - 2(1 - \frac{\alpha h}{\beta})U_0^n}{h^2} - \frac{2g^n}{h\beta}. \qquad (15.32)$$

Finite Element Approximation

All the finite difference methods we have considered attempt to approximate a continuous function at a discrete set of points. An alternative way of looking at the problem of approximating the continuous system is to try to represent the solution using a finite set of continuous functions. One method that uses this idea is called finite elements. While the origins of the method were in analysis of elastic deformation of engineering structures, the mathematical theory has been developed and extended and is the subject of whole courses: here we only will get a brief taste for the method.

We can look at a specific problem: the differential equation

$$u_t = u_{xx},\ 0 < x < 1,\ t > 0, \tag{15.33}$$

with initial condition $u(x, 0) = u_0(x)$ and boundary conditions $u_x(0, t) = u_x(1, t) = 0$. We suppose that for the inner product

$$(u, v) = \int_0^1 u(x)v(x)\mathrm{d}x, \tag{15.34}$$

that we require u to be the solution of

$$(u_t, v) + (u_x, v_x) = 0, \tag{15.35}$$

subject to the initial condition, also in inner product form,

$$(u(\cdot, 0), v) = (u_0, v) \tag{15.36}$$

where v ranges over a space of functions. This is called a weak form of the equation.

In order to determine an appropriate space of functions $\{\phi_r\}$,we retain the same mesh we might use for finite difference approximation, $x_r = rh,\ r = 0, 1, \ldots M$, with $h = 1/M$, and consider a finite set of continuous functions, here denoted $\phi_r(x),\ r = 0, 1, \ldots M$.

Thus we define the space of functions

$$V_h = span\{\phi_0, \ldots, \phi_M\}, \tag{15.37}$$

and assume both that the numerical solution will come from the space V_h and that the weak form will use the space V_h as a test space. Hence U^n is given by a linear combination of these functions and at time t_n the weight for each of these functions is denoted U^n_r, so that

$$U^n(x) = \sum_{r=0}^{M} U^n_r \phi_r(x), \tag{15.38}$$

is the approximation to the continuous solution $u^n(x) = u(x, t_n)$. The solution is then determined by choosing the approximating function U to satisfy the weak form equation for each function that generates V_h (and so will satisfy the weak form equation for any linear combination of these functions), that is

$$(U_t, \phi_r) - (U_x, \phi_r'), \quad r = 1, \dots, M-1. \tag{15.39}$$

The time derivative can be discretised using a forward difference,

$$(\frac{U^{n+1}(x) - U^n(x)}{\Delta t}, \phi_r) = -(U_x^n, \phi_r'), \quad r = 1, \dots, M-1. \tag{15.40}$$

Finally using the representation of the solution in terms of the functions ϕ_r,

$$\sum_{s=0}^{M} U_s^{n+1}(\phi_s, \phi_r) = \sum_{s=0}^{M} U_r^n(\phi_s, \phi_r) - \Delta t \sum_{s=0}^{M} U_s^n(\phi_s', \phi_r'), \ r = 1, \dots, M-1. \tag{15.41}$$

Without it being obvious, the weak formulation has made a great simplification. If we retained the orginal differential equation, we would have had to require that the functions ϕ_r were twice differentiable (at least piecewise so), now we only need them to be once differentiable (again, in a piecewise sense). The practical consequence is that we can use piecewise linear functions with compact support so that the sums in this equation range only over a few index values.

If we define

$$\Phi(x) = \begin{cases} 0, & |x| > h, \\ 1 + x/h, & -h \le x \le 0, \\ 1 - x/h, & 0 < x \le h, \end{cases} \tag{15.42}$$

and then let

$$\phi_r(x) = \Phi(x - x_r), \tag{15.43}$$

then we never have more than three successive functions overlap so that

$$(\phi_{r-1}, \phi_r) \equiv \int_{-h}^{0} -\frac{x}{h}(1 + \frac{x}{h})\mathrm{d}x = \frac{1}{6}h,$$

$$(\phi_r, \phi_r) \equiv 2\int_{0}^{h} (1 - \frac{x}{h})^2\mathrm{d}x = \frac{2}{3}h,$$

$$(\phi_{r-1}', \phi_r') \equiv \int_{-h}^{0} -\frac{1}{h} \cdot \frac{1}{h}\mathrm{d}x = -\frac{1}{h},$$

and

$$(\phi_r', \phi_r') \equiv 2\int_{0}^{h} (-\frac{1}{h})^2\mathrm{d}x = \frac{2}{h},$$

with all inner products zero when indices differ by more than one.

Using these in (15.41), for each $r = 1, \ldots, M-1$,

$$\frac{h}{6}(U_{r-1}^{n+1} + 4U_r^{n+1} + U_{r+1}^{n+1}) = \frac{h}{6}(U_{r-1}^n + 4U_r^n + U_{r+1}^n) + \frac{\Delta t}{h}(U_{r-1}^n - 2U_r^n + U_{r+1}^n). \quad (15.44)$$

To these equations have to be added the two boundary conditions at $x = 0$, $x = 1$ and then the coefficients at time step $n + 1$ will be given by solution of a tridiagonal system. Using the basis functions ϕ_r above, the values U_r^n have a dual interpretation, being both the coefficient or weight of the basis function ϕ_r at time t_n, and also the predicted value for u at the mesh point (x_r, t_n).

This brief introduction only begins the theory for finite element approximation which is covered much more fully in other courses. Particularly when working in multiple space dimensions, finite element methods offer great flexibility in covering irregular or odd shaped domains and a range of basis functions (for example, piecewise quadratic or higher order functions ϕ_r) also adds to flexibility. There are finite difference methods that can cover such domains (finite volume methods fall in this category) but often finite element methods are the prime choice for such problems.

Lecture 16

Two dimensional problems

Only a few real problems can be reduced to one space dimension, more usually interest will be in situations involving surfaces (2D) or volumes (3D). We have time only to look briefly at how the ideas from 1D extend to higher space dimensions, as a general rule, for explicit methods, stability restrictions become more severe, for implicit methods, the matrix inversion step becomes much more difficult.

Suppose $u = u(x, y, t)$ on $0 \le x \le 1$, $0 \le y \le 1$, $t \ge 0$ with

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} && (16.1) \\ u &= 0 \quad \text{on } x = 0,\ x = 1,\ y = 0,\ y = 1 && (16.2) \\ u &= u_0(x, y) \quad \text{on } t = 0. && (16.3) \end{aligned}$$

In the unbounded domain, fundamental solutions are of the form

$$u(x, y, t) \sim \mathrm{e}^{-(k^2+\ell^2)t}\mathrm{e}^{ikx}\mathrm{e}^{i\ell y} \tag{16.4}$$

and defining a 2D Fourier Transform for functions $v(x, y)$

$$\hat{v}(k, \ell) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} v(x, y)\mathrm{e}^{-ikx}\mathrm{e}^{-i\ell y}\,\mathrm{d}x\,\mathrm{d}y \tag{16.5}$$

gives transform solutions

$$\hat{u}(k, \ell, t) = \hat{u}_0(k, \ell)\mathrm{e}^{-(k^2+\ell^2)t}. \tag{16.6}$$

The inverse transform for u can be manipulated into a convolution integral as before.

There are many methods to solve the continous 2D problem using discrete meshes. Here we will look only briefly at rectangular meshes, as illustrated in figure 27(b) but many other mesh shapes can be used, for example a mesh if equliateral triangles is shown in figure 27(a) and there are methods which do not rely on any mesh over the domain. This is particularly important for real domains which are rarely rectangular in shape.

Now return to the problem as formulated on the unit square and let $\Delta x = 1/M_x$, $\Delta y = 1/M_y$, $t_n = n\Delta t$, $x_r = r\Delta x$, $y_s = s\Delta y$ and

$$U^n_{r,s} \approx u(r\Delta x, s\Delta y, n\Delta t).$$

Define two space difference operators by

$$\begin{aligned} \delta_x^2 U^n_{r,s} &= U^n_{r+1,s} - 2U^n_{r,s} + U^n_{r-1,s}, && (16.7) \\ \delta_y^2 U^n_{r,s} &= U^n_{r,s+1} - 2U^n_{r,s} + U^n_{r,s-1}. && (16.8) \end{aligned}$$

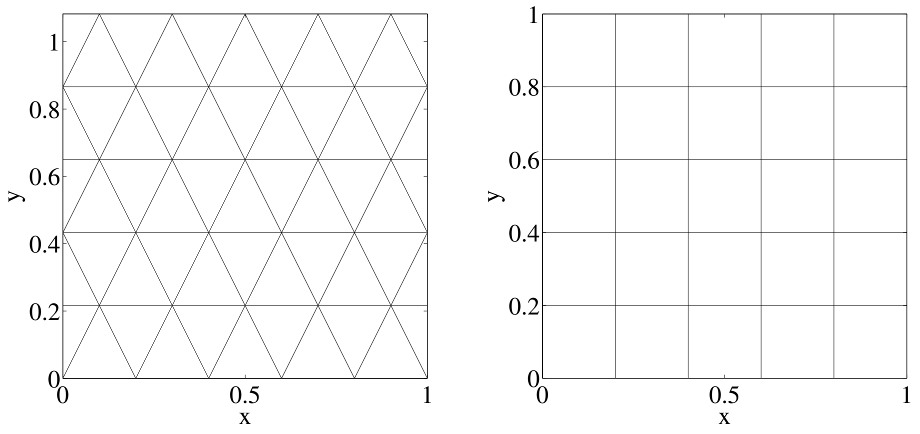


Figure 27: Two example 2D meshes: (a) equilateral triangles, (b) square mesh.

(a) Explicit method

We approximate the equation by an *explicit method*

$$\frac{U_{r,s}^{n+1} - U_{r,s}^{n}}{\Delta t} = \frac{1}{\Delta x^2}\delta_x^2 U_{r,s}^{n} + \frac{1}{\Delta y^2}\delta_y^2 U_{r,s}^{n} \tag{16.9}$$

for $r = 1, \ldots, M_x - 1$, $s = 1, \ldots, M_y - 1$ and $n = 1, \ldots, N = T_{\max}/\Delta t$.

Thus the algorithm is that given all valules $U_{r,s}^n$ at time level n, we explicitly calculate $U_{r,s}^{n+1}$ at the next time level as shown in the stencil in Figure 28.

(b) Fully Implicit scheme

The fully implicit method will evaluate all the space derivatives at the $n + 1$ time level

$$\frac{U_{r,s}^{n+1} - U_{r,s}^{n}}{\Delta t} = \frac{1}{\Delta x^2}\delta_x^2 U_{r,s}^{n+1} + \frac{1}{\Delta y^2}\delta_y^2 U_{r,s}^{n+1}. \tag{16.10}$$

Here, even though $U_{r,s}^{n+1}$, $U_{r,s}^n$ look like arrays, we have to treat all the values as part of a *vector* by

$$(\mathbf{U}^n)^T = \left[U_{1,1}, U_{1,2} \ldots, U_{1,M_y-1}, U_{2,1}, \ldots, U_{2,M_y-1}, , \ldots, U_{M_x-1,1}, \ldots, U_{M_x-1,M_y-1}\right] \tag{16.11}$$

so that $\mathbf{U}^n$ is a vector with $(M_x - 1)(M_y - 1)$ elements.

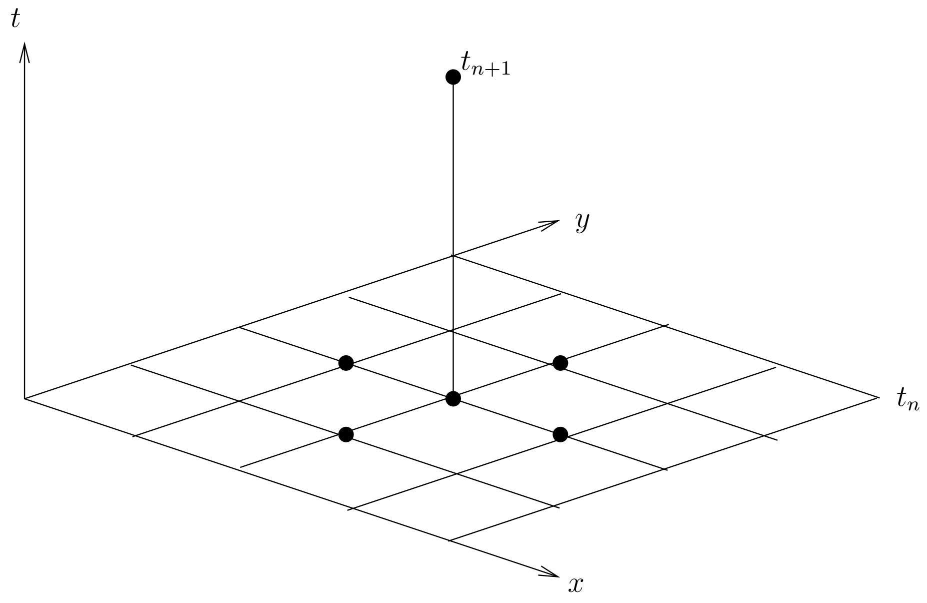


Figure 28: The finite difference stencil for the explicit method in 2D.

For simplicity let $M_x = M_y = M$, $\Delta x = \Delta y = 1/M$, $\mu = \Delta t/\Delta x^2$. Then equation (16.10) is

$$-\mu U_{r,s-1}^{n+1} - \mu U_{r-1,s}^{n+1} + (1+4\mu)U_{r,s}^{n+1} - \mu U_{r+1,s}^{n+1} - \mu U_{r,s+1}^{n+1} \;=\; U_{r,s}^{n}. \qquad (16.12)$$

Define a $(M-1)\times(M-1)$ matrix

$$B \;=\; \begin{pmatrix} 1+4\mu & -\mu & 0 & \dots & \dots & 0 \\ -\mu & 1+4\mu & -\mu & 0 & \dots & \dots \\ 0 & -\mu & 1+4\mu & -\mu & 0 & \dots \\ & & \ddots & \ddots & \ddots & \\ \dots & \dots & 0 & -\mu & 1+4\mu & -\mu \\ 0 & \dots & \dots & 0 & -\mu & 1+4\mu \end{pmatrix}_{(M-1)\times(M-1)} \qquad (16.13)$$

and let

$$A \;=\; \begin{pmatrix} B & -\mu I & 0 & 0 & \dots & 0 \\ -\mu I & B & -\mu I & 0 & & \\ 0 & -\mu I & B & -\mu I & 0 & \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & 0 & -\mu I & B & -\mu I \\ & & & & -\mu I & B \end{pmatrix}_{(M-1)^2\times(M-1)^2} . \qquad (16.14)$$

Then

$$A\mathbf{U}^{n+1} \;=\; \mathbf{U}^{n} \qquad (16.15)$$

or

$$\mathbf{U}^{n+1} = A^{-1}\mathbf{U}^n. \tag{16.16}$$

The matrix A is very large (e.g. with $h = 0.01$, the vector $\mathbf{U}$ has 10^4 elements and A is a $10^4 \times 10^4$ matrix). Special methods are needed to solve the system $A\mathbf{U}^{n+1} = \mathbf{U}^n$. Next term the course looks at methods to solve such systems. For the present we just have to recognise that A is very large (in 3D with $h = 0.01$ on the unit cube, A would be $10^6 \times 10^6$, i.e. 10^{12} entries, nearly all of which are zero).

(c) Theta-method

It is straightforward to set out a θ method:

$$\begin{aligned}\frac{U_{r,s}^{n+1} - U_{r,s}^n}{\Delta t} &= \theta\frac{1}{\Delta x^2}\delta_x^2 U_{r,s}^{n+1} + (1-\theta)\frac{1}{\Delta x^2}\delta_x^2 U_{r,s}^n &(16.17)\\ &+ \theta\frac{1}{\Delta y^2}\delta_y^2 U_{r,s}^{n+1} + (1-\theta)\frac{1}{\Delta y^2}\delta_y^2 U_{r,s}^n. &(16.18)\end{aligned}$$

As with the fully implicit method (which is obviously also the case here when $\theta = 1$), this requires a matrix problem to be solved at each time step when $\theta > 0$ and reduces to an explicit method when $\theta = 0$.

Stability

For a modal analysis of stability, we neglect the boundary and suppose the scheme is applied over a plane and look at modes $U_{r,s}^n \sim \lambda^n \mathrm{e}^{ikr\Delta x}\mathrm{e}^{i\ell s\Delta y}$ so that we have

$$\begin{aligned}\delta_x^2 U_{r,s}^n &= (-2 + 2\cos(k\Delta x))U_{r,s}^n = -4\sin^2\frac{k\Delta x}{2}U_{r,s}^n, &(16.19)\\ \delta_y^2 U_{r,s}^n &= (-2 + 2\cos(\ell\Delta y))U_{r,s}^n = -4\sin^2\frac{\ell\Delta y}{2}U_{r,s}^n. &(16.20)\end{aligned}$$

(a) Stability: Explicit method Substituion of the mode into the discrete scheme gives

$$\lambda = 1 - 4\mu_x\sin^2\frac{k\Delta x}{2} - 4\mu_y\sin^2\frac{\ell\Delta y}{2} \tag{16.21}$$

with $\mu_x = \Delta t/\Delta x^2$, $\mu_y = \Delta t/\Delta y^2$ and stability requires

$$\mu_x + \mu_y \le \frac{1}{2} \tag{16.22}$$

or

$$\Delta t\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}\right) \;\leq\; \frac{1}{2} \tag{16.23}$$

which is more restrictive than the 1D result.

(b) Stability: Implicit Method In this case, the modal analysis gives

$$\lambda \;=\; \frac{1}{1+4\mu_x\sin^2\frac{k\Delta x}{2}+4\mu_y\sin^2\frac{\ell\Delta y}{2}} \tag{16.24}$$

so the method is unconditionally stable.

(c) Stability: Theta-method The algebra is a little more involved than for 1D but essentially the same, we find

$$\lambda \;=\; \frac{1-4(1-\theta)\left(\mu_x\sin^2\frac{k\Delta x}{2}+\mu_y\sin^2\frac{\ell\Delta y}{2}\right)}{1+4\theta\left(\mu_x\sin^2\frac{k\Delta x}{2}+\mu_y\sin^2\frac{\ell\Delta y}{2}\right)} \tag{16.25}$$

and we need $2(1-2\theta)(\mu_x+\mu_y)\leq 1$ for stability when $\theta<1/2$ and there is unconditional stability for $\theta\geq 1/2$.

Truncation error

We now need to use Taylor expansions in three dimensions, x, y and t.

We have $u(x,y,t)$ with $u^n_{r,s}=u(r\Delta x,s\Delta y,n\Delta t)$

$$\begin{aligned}
\delta_x^2 u^n_{r,s} &= u^n_{r+1,s}-2u^n_{r,s}+u^n_{r-1,s} && (16.26)\\
&= \left[u+\Delta x\frac{\partial u}{\partial x}+\frac{\Delta x^2}{2!}\frac{\partial^2 u}{\partial x^2}+\frac{\Delta x^3}{3!}\frac{\partial^3 u}{\partial x^3}+\frac{\Delta x^4}{4!}\frac{\partial^4 u}{\partial x^4}+\ldots\right]^n_{r,s} && (16.27)\\
&\quad -2u^n_{r,s} && (16.28)\\
&\quad +\left[u-\Delta x\frac{\partial u}{\partial x}+\frac{\Delta x^2}{2!}\frac{\partial^2 u}{\partial x^2}-\frac{\Delta x^3}{3!}\frac{\partial^3 u}{\partial x^3}+\frac{\Delta x^4}{4!}\frac{\partial^4 u}{\partial x^4}+\ldots\right]^n_{r,s} && (16.29)\\
&= \Delta x^2\frac{\partial^2 u}{\partial x^2}\bigg|^n_{r,s}+\frac{\Delta x^4}{12}\,\frac{\partial^4 u}{\partial x^4}\bigg|^n_{r,s}. && (16.30)
\end{aligned}$$

Similarly

$$\delta_y^2 u^n_{r,s} \;=\; \Delta y^2\frac{\partial^2 u}{\partial y^2}\bigg|^n_{r,s}+\frac{\Delta y^4}{12}\,\frac{\partial^4 u}{\partial y^4}\bigg|^n_{r,s}, \tag{16.31}$$

with similar expressions at time level $n+1$.

We also have

$$u^{n+1}_{r,s} - u^n_{r,s} = \left[u + \Delta t \frac{\partial u}{\partial t} + \frac{\Delta t^2}{2!}\frac{\partial^2 u}{\partial t^2} + \dots \right]^n_{r,s} \quad (16.32)$$

$$-u^n_{r,s} \quad (16.33)$$

$$= \Delta t \frac{\partial u}{\partial t}\bigg|^n_{r,s} + \frac{\Delta t^2}{2!}\frac{\partial^2 u}{\partial t^2}\bigg|^n_{r,s} + \dots \quad (16.34)$$

As with one dimensional schemes, algebra is easiest if Taylor expansions are carefly placed with respect to time level.

Explicit expand about (x_r, y_s, t_n), see Figure 29.

Implicit expand about (x_r, y_s, t_{n+1}), see Figure 30.

Theta-method expand about $(x_r, y_s, t_{n+1/2})$, see Figure 31.

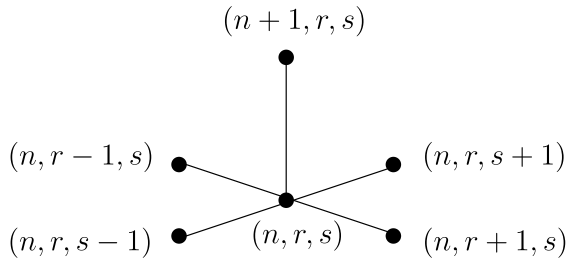


Figure 29: Finite difference stencil for an explicit scheme.

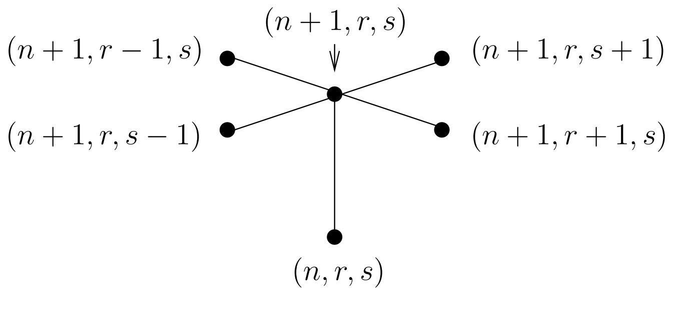


Figure 30: Finite difference stencil for an implicit scheme.

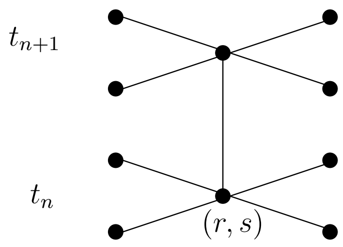


Figure 31: Finite difference stencil for a Theta-method.

(a) Truncation error: explicit method

For the explicit method

$$T^n_{r,s} = \frac{u^{n+1}_{r,s} - u^n_{r,s}}{\Delta t} - \frac{1}{\Delta x^2}\delta_x^2 u^n_{r,s} - \frac{1}{\Delta y^2}\delta_y^2 u^n_{r,s} \tag{16.35}$$

$$= \underbrace{\left(\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2}\right)\Bigg|^n_{r,s}}_{\text{use differential equation}} + \frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2}\Bigg|^n_{r,s} + \frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}\Bigg|^n_{r,s} + \frac{\Delta y^2}{12}\frac{\partial^4 u}{\partial y^4}\Bigg|^n_{r,s} + \tag{16.36}$$

so using $u_t = u_{xx} + u_{yy}$

$$T^n_{r,s} = \frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2}\Bigg|^n_{r,s} + \frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}\Bigg|^n_{r,s} + \frac{\Delta y^2}{12}\frac{\partial^4 u}{\partial y^4}\Bigg|^n_{r,s} + \ldots \tag{16.37}$$

and then provided

$$\left\|\frac{\partial^2 u}{\partial t^2}\right\|_\infty < C_1, \quad \left\|\frac{\partial^4 u}{\partial x^4}\right\|_\infty < C_2, \quad \left\|\frac{\partial^4 u}{\partial y^4}\right\|_\infty < C_3, \tag{16.38}$$

we have

$$|T^n_{r,s}| \le \frac{1}{2}\Delta t C_1 + \frac{1}{12}\Delta x^2 C_2 + \frac{1}{12}\Delta y^2 C_3, \tag{16.39}$$

and the scheme is first order in time and second order in space.

(b) Truncation Error: fully implicit method

Using Taylor expansions centred at (x_r, y_s, t_{n+1}) is is straight forward repetition of the last section to show that

$$|T^{n+1}_{r,s}| \le \frac{1}{2}\Delta t C_1 + \frac{1}{12}\Delta x^2 C_2 + \frac{1}{12}\Delta y^2 C_3, \tag{16.40}$$

and as we have seen before, usning an implicit scheme has no accuracy advantage,

(c) Truncation Error: Theta-method

Calculation of the truncation error for a theta-method is best approached by expansion about $((x_r, y_s, t_{n+1/2})$ and is very tedious, you should be able to show that

$$T_{r,s}^{n+1/2} = \frac{1-2\theta}{2}\Delta t(u_{xxt}+u_{yyt})|_{r,s}^{n+1/2}+\frac{1}{24}\Delta t^2 u_{ttt}|_{r,s}^{n+1/2}-\frac{1}{12}(\Delta x^2 u_{xxxx}+\Delta y^2 u_{yyyy})|_{r,s}^{n+1/2}+\dots, \tag{16.41}$$

and as in 1D, the case $\theta = 1/2$, Crank Nicholson, gives a scheme that is second order in both space and time.

Stability and Convergence

The ideas developed for one space dimension using a semi-discrete Fourier transform carry over to multiple space dimensions.

Define a 2D semi-discrete Fourier Transform for $u_{r,s}^n$ with $u_{r,s}^n = u(r\Delta x, s\Delta y, n\Delta t)$ by

$$\hat{u}^n(k,\ell) \;=\; \Delta x\Delta y\sum_r\sum_s u_{r,s}^n e^{-ikr\Delta x}e^{-i\ell s\Delta y}, \tag{16.42}$$

and define norms

$$\|\hat{u}(k,\ell)\|_{L_2}^2 \;=\; \int_{-\frac{\pi}{\Delta x}}^{\frac{\pi}{\Delta x}}\int_{-\frac{\pi}{\Delta y}}^{\frac{\pi}{\Delta y}} |\hat{u}(k,\ell)|^2 \, d\ell \, dk, \tag{16.43}$$

and

$$\|u^n\|_{\ell_2}^2 \;=\; \Delta x\Delta y\sum_r\sum_s |u_{r,s}^n|^2. \tag{16.44}$$

As in the 1D case with $\Omega_x = \left(-\frac{\pi}{\Delta x}, \frac{\pi}{\Delta x}\right)$, $\Omega_y = \left(-\frac{\pi}{\Delta y}, \frac{\pi}{\Delta y}\right)$

$$\begin{aligned}
\|\hat{u}\|_{L_2}^2 &= \int_{\Omega_x} dk \int_{\Omega_y} d\ell \; |\hat{u}(k,\ell)|^2 \\
&= \Delta x^2\Delta y^2 \int_{\Omega_x} dk \int_{\Omega_y} d\ell \left(\sum_r\sum_s u_{r,s}^n e^{-ikr\Delta x}e^{-i\ell s\Delta y}\right)\left(\sum_p\sum_q \bar{u}_{p,q}^n e^{ikp\Delta x}e^{i\ell q\Delta y}\right) \\
&= \Delta x^2\Delta y^2 \sum_r\sum_s\sum_p\sum_q u_{r,s}^n\bar{u}_{p,q}^n \int_{\Omega_x} e^{ik(p-r)\Delta x}\, dk \int_{\Omega_y} e^{i\ell(q-s)\Delta x}\, d\ell. \qquad (16.45)
\end{aligned}$$

We need $r = p$ and $s = q$ for the integrals to be non zero, hence

$$\|\hat{u}\|_{L_2}^2 \;=\; 4\pi^2\|u^n\|_{\ell_2}^2, \tag{16.46}$$

and so in 2D Parseval's identity becomes

$$\|u^n\|_{\ell_2} = \frac{1}{2\pi}\|\hat{u}\|_{L_2} \tag{16.47}$$

in 2D. Thus for the various discrete schemes on unbounded domains we still obtain

$$\hat{U}^{n+1}(k,\ell) = \lambda(k,\ell)\hat{U}^n(k,\ell) \tag{16.48}$$

where:

1. Explicit:

$$\lambda(k,\ell) = 1 - 4\mu_x \sin^2\frac{k\Delta x}{2} - 4\mu_y \sin^2\frac{\ell\Delta y}{2}, \tag{16.49}$$

2. Implicit:

$$\lambda(k,\ell) = \left(1 + 4\mu_x \sin^2\frac{k\Delta x}{2} + 4\mu_y \sin^2\frac{\ell\Delta y}{2}\right)^{-1}, \tag{16.50}$$

3. θ method:

$$\lambda(k,\ell) = \frac{1 - 4(1-\theta)\left(\mu_x \sin^2\frac{k\Delta x}{2} + \mu_y \sin^2\frac{\ell\Delta y}{2}\right)}{1 + 4\theta\left(\mu_x \sin^2\frac{k\Delta x}{2} + \mu_y \sin^2\frac{\ell\Delta y}{2}\right)}. \tag{16.51}$$

On a bounded domain we can still apply a maximum principle, for example $0 \le x \le 1$, $0 \le y \le 1$ with $\partial U_{\max}$ and $\partial U_{\min}$ defined as maximum and minimum on boundaries and initial data, θ method is

$$\begin{aligned}\left[1 + 2\theta(\mu_x + \mu_y)\right] U_{r,s}^{n+1} &= \theta\left[\mu_x\left(U_{r+1,s}^{n+1} + U_{r-1,s}^{n+1}\right) + \mu_y\left(U_{r,s+1}^{n+1} + U_{r,s-1}^{n+1}\right)\right] \\ &\quad + (1-\theta)\left[\mu_x\left(U_{r+1,s}^{n} + U_{r-1,s}^{n}\right) + \mu_y\left(U_{r,s+1}^{n} + U_{r,s-1}^{n}\right)\right] \\ &\quad + \left[1 - 2(1-\theta)(\mu_x + \mu_y)\right] U_{r,s}^n \end{aligned} \tag{16.52}$$

so making all coefficients on right-hand-side non negative will guarantee a maximum principle, so we need for $0 \le \theta \le 1$

$$1 - 2(1-\theta)(\mu_x + \mu_y) \ge 0 \tag{16.53}$$

or

$$\Delta t \le \frac{\Delta x^2 \Delta y^2}{2(1-\theta)(\Delta x^2 + \Delta y^2)}. \tag{16.54}$$

The results we have derived for truncation error, order of convergence, error analysis are all generalisations of 1D results and derived the same way.

In 2 and 3 dimensions the size of the matrix A grows rapidly as Δx and Δy become small so the major problem is solving the system $A\mathbf{U}^{n+1} = \mathbf{U}^n$ in an implicit method. Essentially

- explicit $\Rightarrow$ very small time steps for stability, many computational steps needed
- implicit $\Rightarrow$ freedom with time steps but matrix inversion hard.

ADI (Alternating Direction Implicit)

A method that has been popular for rectangular domains is called ADI. We start with Crank Nicholson, and as before, define $\mu_x = \Delta t/\Delta x^2$, $\mu_y = \Delta t/\Delta y^2$

$$\left(1 - \frac{1}{2}\mu_x\delta_x^2 - \frac{1}{2}\mu_y\delta_y^2\right)U_{r,s}^{n+1} = \left(1 + \frac{1}{2}\mu_x\delta_x^2 + \frac{1}{2}\mu_y\delta_y^2\right)U_{r,s}^{n} \tag{16.55}$$

and observe that

$$1 - \frac{1}{2}\mu_x\delta_x^2 - \frac{1}{2}\mu_y\delta_y^2 \approx \left(1 - \frac{1}{2}\mu_x\delta_x^2\right)\left(1 - \frac{1}{2}\mu_y\delta_y^2\right) \tag{16.56}$$

$$1 + \frac{1}{2}\mu_x\delta_x^2 + \frac{1}{2}\mu_y\delta_y^2 \approx \left(1 + \frac{1}{2}\mu_x\delta_x^2\right)\left(1 + \frac{1}{2}\mu_y\delta_y^2\right) \tag{16.57}$$

so we examine the scheme

$$\left(1 - \frac{1}{2}\mu_x\delta_x^2\right)\left(1 - \frac{1}{2}\mu_y\delta_y^2\right)U_{r,s}^{n+1} = \left(1 + \frac{1}{2}\mu_x\delta_x^2\right)\left(1 + \frac{1}{2}\mu_y\delta_y^2\right)U_{r,s}^{n}. \tag{16.58}$$

We can show that this is a consistent scheme. The truncation error for Crank Nicholson is

$$T_{r,s}^{n+1/2} = \frac{1}{\Delta t}\left\{u_{r,s}^{n+1} - \left(\frac{1}{2}\mu_x\delta_x^2 + \frac{1}{2}\mu_y\delta_y^2\right)u_{r,s}^{n+1} - u_{r,s}^{n} - \left(\frac{1}{2}\mu_x\delta_x^2 + \frac{1}{2}\mu_y\delta_y^2\right)u_{r,s}^{n}\right\}$$

and so the truncation error of the new scheme is

$$\left(T_{r,s}^{n+1/2}\right)_{\text{new}} = T_{r,s}^{n+1/2} - \frac{1}{2}\mu_x\mu_y\delta_x^2\delta_y^2\left(\frac{u_{r,s}^{n+1} - u_{r,s}^{n}}{\Delta t}\right). \tag{16.59}$$

When the additional term is expanded using Taylor series, the leading term in that expansion will be

$$-\frac{1}{2}\mu_x\mu_y\Delta x^2\Delta y^2 u_{xxyyt} \tag{16.60}$$

so that

$$\left(T_{r,s}^{n+1/2}\right)_{\text{new}} = T_{r,s}^{n+1/2} - \frac{1}{2}\mu_x\mu_y\Delta x^2\Delta y^2 u_{xxyyt} + \ldots, \tag{16.61}$$

and provided $T_{r,s}^{n+1/2} \to 0$ as Δt, Δx, $\Delta y \to 0$, so too will $\left(T_{r,s}^{n+1/2}\right)_{\text{new}}$. As Crank Nicholson is second order in space and time and the additonal term is fourth order in space, the modified scheme will still have good approximation properties and be a least similar to the original Crank Nicholson scheme in accuracy.

However, the huge advantage advantage of the modified scheme is that $1+\frac{1}{2}\delta_x^2$, $1-\frac{1}{2}\delta_x^2$, $1-\frac{1}{2}\delta_y^2$, and $1+\frac{1}{2}\delta_y^2$ are all operators which act in one space direction only so their matrix representation wil be in the form of tri-diagonal matrices and we know both how to store tridiagonal systems compactly, and, how to invert them efficiently.

Let $U_{r,s}^{n+1/2}$ be defined by

$$\left(1-\frac{1}{2}\mu_x\delta_x^2\right)U_{r,s}^{n+1/2} = \left(1+\frac{1}{2}\mu_y\delta_y^2\right)U_{r,s}^{n} \tag{16.62}$$

and then

$$\left(1-\frac{1}{2}\mu_y\delta_y^2\right)U_{r,s}^{n+1} = \left(1+\frac{1}{2}\mu_x\delta_x^2\right)U_{r,s}^{n+1/2}. \tag{16.63}$$

As $1+\frac{1}{2}\mu_x\delta_x^2$ and $1-\frac{1}{2}\mu_x\delta_x^2$ commute, multiply (16.62) by $1+\frac{1}{2}\mu_x\delta_x^2$ and (16.63) by $1-\frac{1}{2}\mu_x\delta_x^2$ to get (16.58).

Now the solution of (16.62) for $U_{r,s}^{n+1/2}$ is a tridiagonal problem for $s = 1, \ldots, M_y - 1$ and similarly the solution of (16.63) is a sequence of tridiagonal problems for $r = 1, \ldots, M_x - 1$.

If we look at stability with mode $U_{r,s}^n = \lambda^n e^{ikr\Delta x} e^{i\ell s\Delta y}$ so that

$$\delta_x^2 U_{r,s}^n = -4\sin^2\frac{k\Delta x}{2}U_{r,s}^n \tag{16.64}$$

$$\delta_y^2 U_{r,s}^n = -4\sin^2\frac{\ell\Delta y}{2}U_{r,s}^n \tag{16.65}$$

then we obtain

$$\left(1+\frac{4}{2}\mu_x\sin^2\frac{k\Delta x}{2}\right)\left(1+\frac{4}{2}\mu_y\sin^2\frac{\ell\Delta y}{2}\right)\lambda = \left(1-2\mu_x\sin^2\frac{k\Delta x}{2}\right)\left(1-2\mu_y\sin^2\frac{\ell\Delta y}{2}\right)$$

or

$$\lambda = \frac{\left(1-2\mu_x\sin^2\frac{k\Delta x}{2}\right)\left(1-2\mu_y\sin^2\frac{\ell\Delta y}{2}\right)}{\left(1+2\mu_x\sin^2\frac{k\Delta x}{2}\right)\left(1+2\mu_y\sin^2\frac{\ell\Delta y}{2}\right)} \tag{16.66}$$

and it will always be the case that $|\lambda| \leq 1$, so the scheme is unconditionally stable.

Evolutionary PDEs: Summary

1. Parabolic system

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}. \tag{16.67}$$

   Fourier modes, semi discrete Fourier Transform, Parseval's Identity

   Numerical Approximations

   continuous scheme $\mathcal{N}(t, u) = 0$

   discrete scheme $N(t_n, U) = 0$

   truncation error $T_n = N(t_n, u)$

   Explicit/Implicit schemes: matrix representation of implicit discrete schemes, Thomas Algorithm

   Convergence: Fourier analysis either by modes or semi discrete transforms, θ method for diffusion equations and truncation error

   Stability: practical stability, von Neumann stability, Lax equivalence theorem (without full proof)

   Finite domain problems: eigenvalues and maximum principle

   Boundary conditions: implicit derivative conditions

   Finite elements: basic ideas

   Two dimensional problems: explicit, implicit, stability, truncation error, ADI — splitting into tridiagonal problems.