5 Higher order ODE's and 1st order systems

Discussion. A higher order ODE can be written as a 1st order system of K equations,

$$\frac{d\mathbf{Y}}{dt} = \mathbf{G}(t, \mathbf{Y}), \tag{5.1}$$

where $\mathbf{Y}$ is a $K-$vector of the unknowns, $\mathbf{Y} = (y_1(t), ..., y_K(t))$, and $\mathbf{G} = (g_1(\mathbf{Y}, t), ..., g_K(\mathbf{Y}, t))$ is the vector of right-hand sides in the equation. By applying the techniques developed above we can generate a finite-difference scheme which approximates the system, for example,

$$\mathbf{Y}_{n+1} = \mathbf{Y}_n + h\mathbf{G}(t_n, \mathbf{Y}_n), \tag{5.2}$$

for a forward Euler approximation.

Written as an iterative algorithm

$$\mathbf{Y}_{n+1} = \mathbf{T}(t_n, \mathbf{Y}_n), \tag{5.3}$$

we have close similarity with 1D case. In particular, if $\boldsymbol{v}_n$ denotes the disturbance at t_n then, after linearisation,

$$\boldsymbol{v}_{n+1} = J\boldsymbol{v}_n \tag{5.4}$$

where $J = \partial\mathbf{T}/\partial\mathbf{Y}_n$ is the Jacobian matrix. If J is diagonalizable then we can write in the usual way

$$J = M^{-1}DM \tag{5.5}$$

and solve the vector difference equation (5.4) as

$$\boldsymbol{v}_n = M^{-1}D^nM\boldsymbol{v}_n. \tag{5.6}$$

Since D is a diagonal matrix of eigenvalues the stability of the algorithm is decided by eigenvalue with the largest absolute value or the spectral radius of J leading to the stability condition

$$\rho(J) < 1. \tag{5.7}$$

Discussion. Convergence. Consistency and stability. Lax theorem.

RK4 for two dependent variables.

For reference, here we reproduce the RK4 algorithm for the case of two first-order equations of the form,

$$\frac{dy}{dt} = f(y, z, t) \tag{5.8}$$

$$\frac{dz}{dt} = g(y, z, t). \tag{5.9}$$

In the usual notation,

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{5.10}$$

$$z_{n+1} = z_n + \frac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4) \tag{5.11}$$

where

$$k_1 = hf(y_n, z_n, t_n) \tag{5.12}$$

$$l_1 = hg(y_n, z_n, t_n) \tag{5.13}$$

$$k_2 = hf(y_n + k_1/2, z_n + l_1/2, t_n + h/2) \tag{5.14}$$

$$l_2 = hg(y_n + k_1/2, z_n + l_1/2, t_n + h/2) \tag{5.15}$$

$$k_3 = hf(y_n + k_2/2, z_n + l_2/2, t_n + h/2) \tag{5.16}$$

$$l_3 = hg(y_n + k_2/2, z_n + l_2/2, t_n + h/2) \tag{5.17}$$

$$k_4 = hf(y_n + k_3, z_n + l_3, t_n + h) \tag{5.18}$$

$$l_4 = hg(y_n + k_3, z_n + l_3, t_n + h) \tag{5.19}$$

Discussion. Stiff ODEs. The task is to solve the following two equations as a system rather than two separate ODE's:

$$\frac{du}{dt} = -au, \tag{5.20}$$

$$\frac{dv}{dt} = -bv, \tag{5.21}$$

with constant $a > 0$, $b > 0$. Show that the Forward Euler method is stable if the time step h satisfies simultaneously $h < 2/a$ and $h < 2/b$. Suppose $a = 10^3$, $b = 10^{-3}$. We need the time step $h < 0.002$ for stability as dictated by the u-equation which will be a complete waste of time as far at the v-equation is concerned.

Discussion. Boundary-value problems for ODEs. Eigenvalue problems. Shooting methods.

Consistency and LTE.

When a finite-difference approximation is derived for a differential equation using standard formulae for the derivatives, the local truncation error is usually easy to estimate. But suppose you are given a numerical algorithm and you need to check whether it is consistent with the original differential equation (roughly speaking, consistency means that the finite-difference equation turns into the original equation when the step sizes tend to zero).

In general, let $L(y(t)) = 0$ be a differential equation (we consider an ODE for simplicity) and $N(y_n)$=0 its finite-difference approximation at the node y_n. To check for consistency, we replace y_n with the exact solution $y(t)$ at the corresponding grid points and see how accurately the equation $N(y(t)) = 0$ is satisfied. It will all become clearer from the following two examples.

Example 1. We are given the forward Euler algorithm,

$$\frac{y_{n+1} - y_n}{h} = f(y_n, t_n), \tag{5.22}$$

on the grid $y_n = y_0 + nh$. Is it consistent with the ODE

$$\frac{dy}{dt} = f(y, t)? \tag{5.23}$$

To answer that, we replace y_n with $y(t)$ in the finite-difference scheme (5.22). Then $y(t_{n+1}) = y(t_n + h) = y(t_n) + hy'(t_n) + O(h^2)$, where the dash is the derivative, and (5.22) becomes

$$\frac{y(t_n) + hy'(t_n) + O(h^2) - y(t_n)}{h} = f(y(t_n), t_n), \tag{5.24}$$

or

$$y'(t_n) + O(h) = f(y(t_n), t_n). \tag{5.25}$$

This is the same as the ODE (5.23) with a local truncation error $O(h)$. Hence the forward Euler method is consistent and it is first-order accurate.

Example 2. Is the following scheme,

$$\frac{2y_{n+1} - 3y_n}{h} = f(y_n, t_n), \tag{5.26}$$

consistent with the ODE (5.23)?

Replace y_n with $y(t_n)$ as before. The equation (5.26) becomes,

$$\frac{2y(t_n) + 2hy'(t_n) + O(h^2) - 3y(t_n)}{h} = f(y(t_n), t_n). \tag{5.27}$$

On the left, we can simplify,

$$-\frac{y(t_n)}{h} + O(1) = f(y(t_n), t_n). \tag{5.28}$$

As $h \to 0$, the left-hand side in (5.28) looks nothing like the left-hand side in the ODE (5.23). Hence the scheme (5.26) is not consistent with (5.23).

The procedure for partial differential equations is similar.

6 Partial differential equations (PDE).

PDEs arise in physics, engineering, biology, computer sciences, finance, in every problem where the unknown quantity varies in several spatial dimensions or in space and time. The general form of a PDE for a function $u(x_1, x_2, ...x_n)$ is

$$F(x_1, ..., x_n, u, u_{x_1}, u_{x_2}, ..., u_{x_1x_1}, u_{x_2x_2}, u_{x_1x_2}, ...) = 0, \tag{6.1}$$

where

$$u_{x_1} = \frac{\partial u}{\partial x_1}, \qquad u_{x_1x_1} = \frac{\partial^2 u}{\partial x_1^2}, \tag{6.2}$$

and so on, are partial derivatives. The equation (6.1) requires initial and/or boundary conditions.

Often the boundary conditions are clear from the physical formulation behind the PDE. If, for instance, you boil an egg and (6.1) models the temperature distribution inside the egg ('the egg equation') then you need to know the temperature of the water surrounding the egg (boundary condition) and also the initial distribution of temperature inside the egg (is the egg straight from the fridge and therefore uniformly chilled or it spent some time in the kitchen and is cooler on the inside and warmer near the shell). In this example both the initial and boundary conditions are needed to have a unique solution.

Well-posedness.

Even in seemingly natural situations the solution of a PDE may not be well behaved - it may show extreme sensitivity to variations in boundary conditions or changes in control parameters, may not be unique or may not even exist. As an example, return to the boiling egg problem and pose the following question. Suppose boiling begins at $t = 0$min and stops at $t = 10$min when we have a very hard-boiled egg. Suppose at $t = 10$min we know the temperature everywhere inside the egg. Can we go back in time and reconstruct the temperature for earlier times, that is for $t < 10$ min? During boiling, the surface of the egg is subjected to constant fluctuations in temperature (due to rising vapour bubbles in boiling water, flickering flame of the gas oven etc.) In forward time these fluctuations diffuse very quickly and have no noticeable effect on the final product - the boiled egg. But then in backward time any small perturbations will grow - the problem in reverse time proves to be unstable.

A PDE formulation (inducing the equation, initial and boundary conditions) is said to be well-posed if the solution exists, it is unique and depends continuously on the problem parameters (the parameters may be a part of boundary conditions, for instance). Sometimes the continuous dependence on the parameters is replaced by the condition of stability although many physical problems involving instability phenomena are well-posed in the sense that their solutions can be studied successfully within the PDE framework.

Ill-posed problems are also important and are studied in the theory of inverse problems (re-constructing the cause given an observed effect). Examples include computerized tomography (reconstructing the absorption pattern inside a human body from intensity of X-rays passing through it), inverse heat conduction (to determine the boundary heat flux required for a desired

temperature field in continuous steel casting), inverse wave scattering (determine the shape of an obstacle by observing the waves scattered by it in oil exploration). In this course we are concerned with well-posed problems.

Example of an ill-posed problem. Solve

$$u_t = -u_{xx} \tag{6.3}$$

on $t > 0$, $-\infty < x < \infty$ with $u = f(x)$ at $t = 0$ and decay conditions at infinity. Using Fourier transforms, the solution is

$$u = \frac{1}{2\pi} \int_{-\infty}^{\infty} \bar{f}(k) e^{ikx+k^2t} dk, \tag{6.4}$$

where $\bar{f}(k)$ is the transform of the initial condition. The integral does not converge unless $\bar{f}(k)$ decays at infinity faster than exponential. This is very restrictive. The problem is in the term

$$u \sim e^{ikx+k^2t}. \tag{6.5}$$

It shows that the solution can be made arbitrarily large at any chosen time by taking k large enough. A similar test will be used for stability calculations later on.

Classification of PDEs.

The **order** of a PDE is determined by the order of the highest derivative.

A PDE is linear if F in (6.1) is a linear function of u and all its derivatives. For example,

$$e^{x+y} \cdot u + x^{100} \cdot u_x + \sin y \cdot u_y = \tan(xy), \tag{6.6}$$

is a linear first-order PDE. On the other hand,

$$u_{xx}^2 + u_y^2 = 4, \tag{6.7}$$

is a non-linear second-order PDE.

Depending on the severity of non-linearity, subclasses of semi-linear and quasi-linear equations are considered. A second-order semi-linear equation in two independent variables is linear with respect to the second derivatives,

$$a(x, y)u_{xx} + 2b(x, y)u_{xy} + c(x, y)u_{yy} = F(x, y, u, u_x, u_y), \tag{6.8}$$

where the right-hand side could be anything. A quasi-linear equation may have the function and lower-order derivatives in the coefficients to the highest derivatives, for example a first-order quasi-linear PDE in two variables will have the following general form,

$$a(x, y, u)u_x + b(x, y, u)u_y = c(x, y, u). \tag{6.9}$$

As we can see, the highest derivatives in the quasi-linear PDE still appear in a linear combination.

A further classification can be given to linear PDEs, with reference to a forcing term, into homogeneous and non-homogeneous (this is similar to linear ODEs). A homogeneous linear PDE has solutions which satisfy the superposition principle (again in similarity with linear ODEs).

7 First-order PDE.

Hyperbolicity. Characteristics.

To keep things simple, we consider a 1st-order PDE,

$$a(x, y)u_x + b(x, y)u_y = c(x, y), \tag{7.1}$$

with given $a(x, y), b(x, y), c(x, y)$ and unknown $u(x, y)$. The solution is to be found in a certain region in the $(x, y)-$plane. Since this is a first order equation we would expect that knowing the function $u(x, y)$ on some curve Γ in the plane should be enough to find the solution at least in the vicinity of this curve.

[diagram] This, however, is not always the case. Let $x = x(s), y = y(s)$ be

a parametric equation of the curve Γ. Using the chain rule, we can compute

the derivative of u along Γ,

$$\frac{du(x(s),y(s))}{ds} = u_x\frac{dx}{ds} + u_y\frac{dy}{ds}. \tag{7.2}$$

But since u_x and u_y are related by (7.1) we have

$$u_x = -\frac{b}{a}u_y + \frac{c}{a}, \tag{7.3}$$

assuming $a \neq 0$ for definiteness. Then for the derivative along Γ we have

$$\frac{du(x(s),y(s))}{ds} = u_y\left(\frac{dy}{ds} - \frac{b}{a}\frac{dx}{ds}\right) + \frac{c}{a}. \tag{7.4}$$

For a general curve Γ, the last formula just establishes a relationship between the derivative along Γ and one of the partial derivatives of u, u_y in this case. However on a curve given by

$$\frac{dy}{ds} = \frac{b}{a}\frac{dx}{ds} \text{ or } \tfrac{dy}{b(x,y)} = \tfrac{dx}{a(x,y)}, \tag{7.5}$$

equation (7.4) reduces to the ODE,

$$\frac{du}{ds} = \frac{c}{a}, \tag{7.6}$$

which we can solve and find u along this curve. Such curves, called characteristic curves or characteristics, have a special significance for the equation (7.1). Firstly, the function u cannot be given an arbitrary distribution along the curve, instead it is sufficient to specify u at one point on the curve and then the equation (7.6) determines u everywhere along the characteristic (in a sense, information propagates along characteristics). Secondly, even if we attempt to specify u along the characteristic, the equation (7.1) does not allow us to compute the solution in the vicinity of the characteristic (the equation degenerates into (7.6)). And the third property used often in analysis of hyperbolic PDEs is that, when written in the characteristic variables, a system of PDEs can be reduced to ODEs and therefore admits explicit solutions.

The conclusion for us from here is that, when setting up initial conditions for the equation (7.1) on a certain curve Γ we need to make sure that $\Gamma$is not a characteristic curve.

A PDE whose solution can be completely described in terms of characteristics are called hyperbolic.

[You may have noticed that the preceding calculation was just a long way of saying that the equation (7.1) rewritten in vector form as,

$$(a, b) \cdot \nabla u = c, \tag{7.7}$$

gives a directional derivative of u along the vector (a, b) in the (x, y)- plane. If the vector (a, b) is everywhere tangent to the curve Γ then the equation (7.1) (or (7.7)) simply tells us how u evolves along Γ and does not say anything about the function u in the vicinity of Γ.]

Finite difference methods.

We shall consider several methods for the simplest PDE with constant coefficients for a function, $u = u(x, t)$, given an initial condition of the form,

$$u_t + au_x = 0, \qquad u(0, x) = u_0(x), \tag{7.8}$$

to be solved for $t > 0$. We assume $a = const > 0$.

There is an exact solution, $u = u_0(x - at)$ which shows that the characteristics of the equation are lines $x - at = const$ in the $(t, x)-$plane.

Numerical solution is considered on the grid $x_n = n\Delta x, t_m = m\Delta t$, with the step sizes Δt and Δx so that $u(x_n, t_m)$ is approximated by $u_{n,m}$.

Explicit method with forward $x-$derivative.

$\uparrow x$

$\bullet(n + 1, m)$

$\bullet(n, m)$ $\qquad$ $\bullet(n, m + 1)$ $\rightarrow t$

Approximating the equation in (7.8) at the grid point t_m, x_n we shall attempt forward derivatives in both t and x,

$$\frac{u_{n,m+1} - u_{n,m}}{\Delta t} + a\frac{u_{n+1,m} - u_{n,m}}{\Delta x} = 0, \tag{7.9}$$

hence

$$u_{n,m+1} = (1 + C)u_{n,m} - Cu_{n+1,m}. \tag{7.10}$$

We use here the so-called Courant number,

$$C = a\frac{\Delta t}{\Delta x}. \tag{7.11}$$

Stability.

Let us first consider stability of solutions of the original PDE (7.8). A standard approach is to assume a travelling-wave solution,

$$u = Ae^{i(kx-\omega t)} + c.c. \tag{7.12}$$

where $c.c.$ denotes the complex conjugate, A is a constant amplitude (which in a linear equation is immaterial), k is the wavenumber and ω is the frequency. The wave is assumed periodic in space therefore k is real. The frequency ω should follow from the equation as a function of the wavenumber.

Substitution of (7.12) into (7.8) yields

$$-i\omega + aik = 0. \tag{7.13}$$

This is the dispersion relation for our equation. It follows that $\omega = ak$ which means that waves of given wavenumber neither decay nor grow - they in fact propagate with constant phase speed $c = \omega/k$.

Now we shall try a similar approach to the finite-difference algorithm (7.10). In the numerical stability theory this is known as the von Neumann stability analysis. We take

$$u_{n,m} = \lambda^m e^{ink} \tag{7.14}$$

with real k. The aim is to compute λ. Using $u_{n,m+1} = \lambda^{m+1}e^{ink}, u_{n+1,m} = \lambda^m e^{i(n+1)k}$, the equation (7.10) gives us

$$\lambda = 1 + C - Ce^{ik}. \tag{7.15}$$

It is almost obvious that $|\lambda| > 1$ therefore the method is unstable.

Note: the equation (7.8) is a first-order advection equation, a is the advection speed (e.g. the speed of wind carrying particles). When $a > 0$ the advection is in the positive $x-$direction. In such interpretation the forward $x-$derivative is called a downwind approximation (downwind scheme, downwind derivative etc). So our approximation can be called a forward in time downwind scheme.

Discussion: where does instability come from? Sketch the location of the approximation nodes and draw a characteristic through the node $(n, m+1)$

to see what happens. Region of dependence and influence for a hyperbolic PDE.

Explicit method with backward $x-$derivative.

$\uparrow x$
$\bullet(n,m)$ $\bullet(n,m+1)$
$\bullet(n-1,m)$
$\rightarrow t$

This is a forward-time backward-space approximation or an upwind scheme,

$$\frac{u_{n,m+1}-u_{n,m}}{\Delta t}+a\frac{u_{n,m}-u_{n-1,m}}{\Delta x}=0, \tag{7.16}$$

giving

$$u_{n,m+1}=(1-C)u_{n,m}+Cu_{n-1,m}. \tag{7.17}$$

The Courant number is the same, $C=a\Delta t/\Delta x$. Now using the von Neumann substitution (7.14) we get

$$\lambda=1-C+Ce^{-ik}. \tag{7.18}$$

By sketching λ in the complex plane, for example, we can see that the scheme is unstable if $C>1$and stable in the case $C<1$.Hence the method is stable if

$$a\Delta t/\Delta x<1 \text{ or } \Delta t<\Delta x/a. \tag{7.19}$$

The stability condition (7.19) is known as the Courant-Friedrichs-Lewy condition.

Discussion: derive the CFL condition graphically considering the characteristic line relative to the nodes of the computational grid.

Lax-Friedrichs method.

Second-order central derivative in x and forward in t, only for the time derivative calculation $u_{n,m}$ is replaced by the average of the neighbouring points.

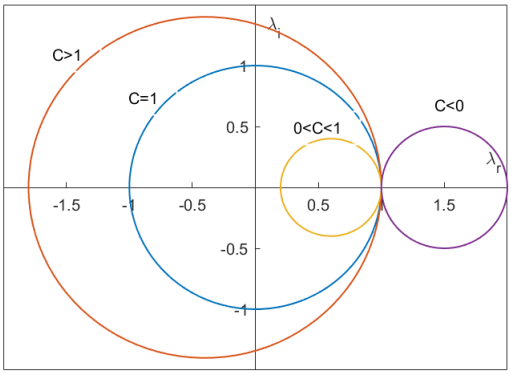

Figure 1: $\lambda(k)$

$\uparrow x$
$\bullet\ (n+1,m)$
$\bullet(n,m) \qquad \bullet(n,m+1)$
$\bullet(n-1,m) \qquad\qquad \rightarrow t$

We have (7.8) discretized as follows,

$$\frac{u_{n,m+1} - \frac{1}{2}\left(u_{n+1,m} + u_{n-1,m}\right)}{\Delta t} + a\frac{u_{n+1,m} - u_{n-1,m}}{2\Delta x} = 0, \tag{7.20}$$

or

$$u_{n,m+1} = \frac{1}{2}\left(u_{n+1,m} + u_{n-1,m}\right) - \frac{C}{2}\left(u_{n+1,m} - u_{n-1,m}\right). \tag{7.21}$$

The Courant number is the same, $C = a\Delta t/\Delta x$. Testing for stability,

$$u_{n,m} = \lambda^m e^{ikn}, \tag{7.22}$$

we obtain

$$\lambda = \frac{1}{2}\left(e^{ik} + e^{-ik}\right) - \frac{C}{2}\left(e^{ik} - e^{-ik}\right), \tag{7.23}$$

$$\lambda = \cos k - iC\sin k. \tag{7.24}$$

The magnitude of λ is given by

$$|\lambda|^2 = \cos^2 k + C^2\sin^2 k = 1 - \sin^2 k + C^2\sin^2 k \tag{7.25}$$

$$= 1 + (C^2 - 1)\sin^2 k. \tag{7.26}$$

For stability, $0 < |\lambda| < 1$, we find $0 < C^2 < 1$, hence the Lax-Friedrichs algorithm is conditionally stable.

The Lax-Friedrichs scheme tends to give smoothed-out solutions - any initial discontinuities in u become leveled out over time. This is due to an interesting and important phenomenon of numerical dissipation which we can illustrate as follows. Re-write (7.20) as

$$\frac{u_{n,m+1} - u_{n,m}}{\Delta t} + a\frac{u_{n+1,m} - u_{n-1,m}}{2\Delta x} = \frac{u_{n+1,m} - 2u_{n,m} + u_{n-1,m}}{2\Delta t}. \qquad (7.27)$$

The last equation we can interpret as a finite-difference approximation of the diffusion-type equation,

$$u_t + au_x = \frac{1}{2}\frac{(\Delta x)^2}{\Delta t}u_{xx}, \qquad (7.28)$$

where the diffusion coefficient (viscosity) in front of u_{xx} denoted often as D depends on the grid size,

$$D = \frac{1}{2}\frac{(\Delta x)^2}{\Delta t}. \qquad (7.29)$$

For any finite values of the step sizes the diffusion coefficient is finite hence giving the effect of dissipation in the numerical solution.

Discussion. What to do if we have

$$a(x, y)u_x + b(x, y)u_y = c(x, y), \qquad (7.30)$$

with variable coefficients? As a first approximation, 'freeze' the coefficients but needs to be done carefully.

Discussion. What if the x-range is finite? Only selected values of k will be allowed however the effect will be small if we are away from the boundaries and the grid is fine enough.

Discussion. Graphical interpretation of CFL. And how do we set the boundary condition if x-range is finite?