

PDEs and Waves III

Prof Tony Roberts
School of Mathematical Sciences
<mailto:anthony.roberts@adelaide.edu.au>

July 17, 2017

Preface

Why can we hear sound well downwind, but not upwind? Why does sound carry long distances at night? Why do weather systems generally travel from west to east? These and many other questions we can answer in this course by understanding wave systems. Along the way we learn lots about partial differential equations that also illuminates and solves many non-wave problems.

Four themes arise throughout this course:

- Where do partial differential equations come from? (modelling) we focus mainly on wave problems.
- How do we solve partial differential equations? (algebraic only some classes, otherwise numeric)
- What general patterns are seen in solutions? (theoretical guidance)
- What do the results mean in application? (interpretation) mainly to wave problems.

Among all the details, look for the general results about general partial differential equations. Such results empower us to understand the partial differential equations wherever we meet them.

Plan 35+ scheduled classes, six written assignments and frequent homework (totalling 30%), one examination (70%).

Contents

1 Conservation of mass and momentum	6
1.1 Car traffic has waves	6
1.2 Conservation of fluid	8
1.3 Momentum PDE for ideal gases	9
1.4 The wave equation	11
1.5 The dispersion relation of waves	12
2 Separation of variables	14
2.1 Linearity empowers analysis	14
2.2 Separation of variables generates boundary value problems	15
3 Wonderful Sturm–Liouville boundary value problems	20
3.1 Self-adjoint operators form Sturm–Liouville problems	23
3.2 Eigenfunctions expand inhomogeneous solutions	27
4 Discretise 1D space	30
4.1 Lagrange’s theorem underpins the method of lines	30
4.2 Find equilibria	40
4.3 Numerical linearisation characterises solution dynamics	42
4.4 PDE-free patch dynamics	50
4.5 May use Octave or Scilab	52
5 Model shallow water waves	55
5.1 Conservation derives the PDEs	55
5.2 Small amplitude waves	56
5.3 Compute seiches in 1D	57
6 PDEs with at least three independent variables	62
6.1 Vibration of a rectangular membrane	63
6.2 The self-adjoint Sturm–Liouville nature of Helmholtz-like PDEs	66
7 Computational integration	72
7.1 1D heat/diffusion PDE raises fundamental issues	72
7.2 Crank–Nicolson schemes are reasonably stable and accurate	75
7.3 Invoke sparse matrices for implicit schemes	77
7.4 Crank–Nicolson discretises wave systems	81
7.5 Second order PDEs in 2D	84
8 General wave dynamics	89
8.1 Water waves in finite depth	89

8.2	Energy travels at the group velocity	92
8.3	Wave propagation in multi-dimensions	94
9	Shocking classification of PDEs	97
9.1	Change of variables transforms the PDE	97
9.2	Reduction to the hyperbolic canonical form	99
9.3	Elliptic and parabolic canonical form	100
9.4	Traffic flow and the method of characteristics	101
9.5	Loud uni-directional sound	106

Bibliography

- Agarwal, R. P. & O'Regan, D. (2009), *Ordinary and Partial Differential Equations With Special Functions, Fourier Series, and Boundary Value Problems*, Springer.
- Crank, J. & Nicolson, P. (1947), 'A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type', *Mathematical Proceedings of the Cambridge Philosophical Society* **43**, 50–67.
http://journals.cambridge.org/article_S0305004100023197
- Haberman, R. (1987), *Elementary applied partial differential equations: with Fourier series and boundary value problems*, 2nd edn, Prentice-Hall.
- Kevrekidis, I. G. & Samaey, G. (2009), 'Equation-free multiscale computation: Algorithms and applications', *Annu. Rev. Phys. Chem.* **60**, 321–44.
- Kreyszig, E. (2011), *Advanced engineering mathematics*, 10th edn, Wiley.
- Lighthill, J. (1979), *Waves in Fluids*, Cambridge University Press.
- MacKenzie, T. & Roberts, A. J. (2006), 'Accurately model the Kuramoto–Sivashinsky dynamics with holistic discretisation', *SIAM J. Applied Dynamical Systems* **5**(3), 365–402.
http://pubs.siam.org/SIADS/volume-05/art_62773.html
- Quarteroni, A. & Saleri, F. (2006), *Scientific Computing with MATLAB and Octave*, Vol. 2 of *Texts in Computational Science and Engineering*, 2nd edition edn, Springer. (Available in pdf via the library of the University of Adelaide.).
- Roberts, A. J. (1994), *A one-dimensional introduction to continuum mechanics*, World Sci.
- Roberts, A. J. & Kevrekidis, I. G. (2007), 'General tooth boundary conditions for equation free modelling', *SIAM J. Scientific Computing* **29**(4), 1495–1510.

1 Conservation of mass and momentum

Chapter Contents

1.1	Car traffic has waves	6
1.2	Conservation of fluid	8
1.3	Momentum PDE for ideal gases	9
1.4	The wave equation	11
1.5	The dispersion relation of waves	12

1.1 Car traffic has waves

Figure 1.1 shows car traffic on a highway: PDEs model the movement of cars (Roberts 1994, §2.2).

1. *Choose:* model the number of cars per km, not each individual car.

2. *Define:*

- x measures position along the highway (km);
- t is time (min);
- $\rho(x, t)$ is the density of cars on a highway (cars/km); and
- $q(x, t)$ is the rate of cars passing the position x at time t (cars/min), called the **flux** of cars.



Figure 1.1: highway traffic [photo by Daniel Reiter].

3. *Conservation:* cars do not appear or disappear (Roberts 1994, §2.1).

Theorem 1.1. *If a function $f(x)$ is continuous on some interval $[c, d]$ and satisfies $\int_a^b f dx = 0$ for all $c < a < b < d$, then $f(x) = 0$ for all $c < x < d$.*

Consequently, conservation derives the 1st order nonlinear PDE, called the **continuity equation**,

$$\boxed{\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0 \quad \text{for all } x.} \quad (1.1)$$

Observation closes the problem: *on a one lane highway* the flux at any point is $q = \rho u = (\rho - \rho^2/140)$ cars/min. The continuity PDE, via the chain rule, then determines

$$\frac{\partial \rho}{\partial t} + (1 - \rho/70) \frac{\partial \rho}{\partial x} = 0 \quad \text{for all } x, \quad \text{provided } \rho \leq 140. \quad (1.2)$$

Exercise 1.1 Consider car traffic on a one lane highway and their density ρ and flux q .

- i. Suppose there is a uniform stream of cars, each car is 5 m long and they are each separated by 20 m from the car in front and behind, what is the car density?
- ii. Suppose the traffic has 120 cars spread evenly along a two kilometre stretch of road, and the cars are all travelling at 30 km/hr, what is the flux of cars past any point?
- iii. Experiments in the Lincoln Tunnel, New York, found that a good fit to the data (except at low density) was that the car velocity $u = 28 \log(142/\rho)$ km/hr for density in cars/km. What is the corresponding continuity PDE?



4. *Scaling:* omit here.

5. *Linearise* Then constant density, $\rho(x, t) = \rho_*$, is an exact solution where all cars travel at the same speed.

Generally, suppose density is $\rho(x, t) = \rho_* + \hat{\rho}(x, t)$ for small $\hat{\rho}$, then we derive the 1D, elementary, wave PDE

$$\frac{\partial \rho}{\partial t} + c_* \frac{\partial \rho}{\partial x} \approx 0 \quad \text{for } c_* = (1 - \rho_*/70) \text{ km/min.} \quad (1.3)$$

D'Alembert's solution of the wave PDE (1.3) is $\rho = f(x - c_* t)$ for any function f . Substitute and check. Initial conditions determine the function $f(x) = \rho_0(x)$.

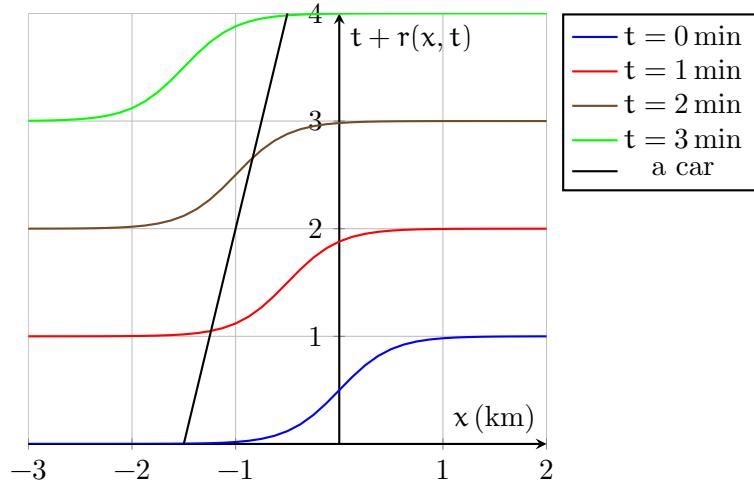


Figure 1.2: density wave of cars, and a car path. Density plots displaced vertically for clarity.

6. *Interpret: car waves propagate.* The predicted density is then $\rho = \rho_0(x - c_*t)$: the c_*t shifts the shape so that the shape travels at the **wave speed** c_* .

Example 1.1 Suppose car density is near 105 cars/km, then $c_* = 1 - 105/70 = -\frac{1}{2}$ km/min. In dense traffic the shape is a backwards wave, see Figure 1.2, even though the cars travel forwards at speed $u = 1 - 105/140 = \frac{1}{4}$ km/min.



Exercise 1.2 Consider car traffic on a one lane highway and the predictions of the continuity PDE.

- Given the car velocity $u \approx (1 - \rho/140)$ km/min, what is the car and wave velocity in traffic of density 35 cars/km?
- Suppose the car velocity is instead $u \approx (1 - \rho/\rho_j)(1 - \rho/\rho_j/2)$ km/min for some value ρ_j : by considering the x -derivative of the corresponding flux, derive the corresponding wave speed c_* at a density ρ_* .



1.2 Conservation of fluid

Consider flow of a fluid, either gas or liquid such as blood, along a tube.

- Choose* to model as a ‘continuum’, not as individual molecules.
- Define:* Suppose the tube aligned along the x -axis has constant cross-sectional area A , in m^2 (will have variable A later).

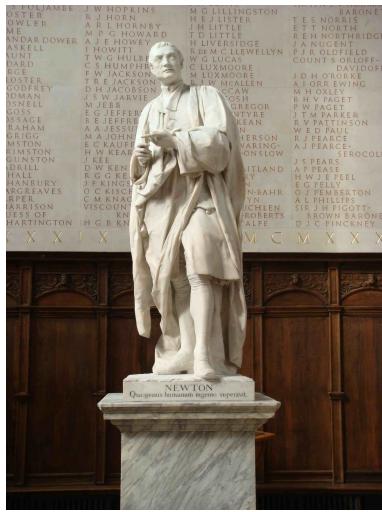


Figure 1.3: Newton's statue in the chapel at Trinity College Cambridge: Wordsworth wrote

*Newton with his prism and silent face,
The marble index of a mind for ever
Voyaging through strange seas of thought, alone.*

The fluid at each position x and time t has some density ρ in kg/m^3 as space is 3D. But our conservation argument applies to density per length: so the density discussed earlier is $\rho_1(x, t) := A\rho(x, t)$ in kg/m . At each position x and time t we choose to suppose the fluid moves at effective velocity $u(x, t)$ along the tube; that is, the flux across any cross-section is $q(x, t) = \rho_1 u = A\rho u$.

3. *Conservation:* If fluid does not leak out of the tube, then the fluid is conserved and the 1D continuity equation (1.1) holds for this fluid:

$$\frac{\partial \rho_1}{\partial t} + \frac{\partial q}{\partial x} = 0 \implies \dots \implies \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = 0. \quad (1.4)$$

Both density ρ and velocity u are unknown: we need a second PDE.

1.3 Momentum PDE for ideal gases

approximation methods derived from physical intuition are frequently more reliable than rigorous mathematical methods, because in the case of the latter it is easier for errors to creep into the fundamental assumptions.

Heisenberg, 1969

For mechanical systems Newton's second law gives a PDE for the velocity u (Roberts 1994, §3.1). Recall that the second law says momentum only changes because of applied forces, $dP/dt = F$ for momentum P being mass \times velocity. We consider the 1D **momentum density** $A\rho u$ and the effect of **pressure** $p(x, t)$ —the force per area exerted by the fluid across some surface (units of kg/m/s^2). But momentum moves into or out of the interval with some flux

across the ends. Consequently we derive the momentum equation

$$\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} \right) = - \frac{\partial p}{\partial x}. \quad (1.5)$$

Now we have two equations in three unknowns: ρ, u, p .

Sanity check: do the units make sense?

Observation closes the problem. Here this comes from school science in the form of the equation of state for ideal gases:

$$p = \rho RT.$$

What about the temperature T ? To a good approximation $T = K\rho^{\gamma-1}$ where K and γ are constants which depend upon the gas. Thus,

$$p = \left(\frac{k^2}{\gamma} \right) \rho^\gamma, \quad (1.6)$$

where γ and $k = \sqrt{\gamma R K}$ are constants of the gas, which are determined experimentally: $\gamma \approx 5/3$ for air (dimensionless).

This gives us the third equation we need for the three unknowns: ρ, u and p . However, it is convenient to eliminate the pressure p from the equations to give the two PDEs

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} (\rho u) = 0 \quad (1.7)$$

$$\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} \right) = -k^2 \rho^{\gamma-1} \frac{\partial \rho}{\partial x}, \quad (1.8)$$

for the two unknowns ρ and u .

Exercise 1.3 Let's explore the overall vertical structure of the atmosphere when at rest. Let x measure distance upwards from the Earth's surface of an imagined 'tube' of air. The continuity PDE is satisfied by zero velocity and by a density $\rho(x)$ constant in time. The momentum PDE determines the density structure.

- (a) Gravity acts downwards on the air with strength g [m/s/s]. Incorporate such forcing in the momentum PDE. Recall Newton's 2nd law for the air in the tube between heights $x = a$ and $x = b$ is here $\frac{dp}{dt} = \text{forces} + \text{influx}$. How should the additional gravitational forcing be included in this force term?
- (b) Given the momentum PDE for an ideal gas ($\gamma = 5/3$) eventually then reduces to

$$\rho(u_t + uu_x) = -g\rho - k^2 \rho^{\gamma-1} \rho_x$$

(using subscripts for partial derivatives), find the ODE satisfied by the density.

- (c) Solve the ODE, as a separable ODE, given the density of air at the Earth's surface is approximately one [kg per cubic metre].



1.4 The wave equation

The most basic process in an ideal gas is the propagation of sound—waves in the density-velocity (Roberts 1994, §3.2).

Start with an exact solution. Here identify a state of rest, or equilibrium (sometimes called a fixed point),

$$u = 0 \quad \text{and} \quad \rho = \rho_*,$$

where ρ_* is any constant, say the density of the atmosphere near the earth's surface.

Linearisation Find the behaviour of disturbances to this basic state by linearisation: we assume

$$u = \hat{u}(x, t) \quad \text{and} \quad \rho = \rho_* + \hat{\rho}(x, t),$$

where the perturbations \hat{u} and $\hat{\rho}$ are ‘small’ in the sense that we neglect products of ‘small’ terms. The result of substitution, approximation, and rearrangement gives the **wave equation**

$$\frac{\partial^2 u}{\partial t^2} = c_*^2 \frac{\partial^2 u}{\partial x^2}, \quad \text{for } c_* = k\rho_*^{(\gamma-1)/2}, \quad (1.9)$$

describing waves of velocity fluctuations (that is, sound).

Waves travel to the left or to the right with wave speed c_* . This is most easily seen by **D'Alembert's solution**:

$$u = f(x - c_* t) + g(x + c_* t) \quad (1.10)$$

where f and g are any twice differentiable functions (Kreyszig 2011, p.553–4). Functions f and g , in any specific situation, are determined by the initial and boundary conditions.

Any sound wave also involves density fluctuations which travel with the wave.

Example 1.2 What if the initial conditions are $u(x, 0) = F(x)$ and $u_t(x, 0) = 0$?



Exercise 1.4 Recall D'Alembert's solution $u(x, t) = f(x - ct) + g(x + ct)$ of the wave PDE $u_{tt} = c^2 u_{xx}$. Determine the form of D'Alembert's solution for the particular case of initial conditions $u(x, 0) = 0$ and $u_t(x, 0) = G(x)$.

- (a) The initial condition $u(x, 0) = 0$ determines what relation between f and g ?
- (b) The initial condition $u_t(x, 0) = G(x)$ determines, for any constant a , what expressions for f and g ?
- (c) Hence write down D'Alembert's solution.



1.5 The dispersion relation of waves

A situation with waves is characterised by mathematical solutions depending upon $x \pm ct$ as these represent some pattern travelling at speed $\pm c$. But unlike *the* wave PDE (1.9), in general, patterns of different wavelength travel at different speeds. The question: how does a pattern of wavelength L , say $\cos(2\pi x/L)$, travel?

So let's look for travelling wave solutions of PDES: solutions that are trigonometric functions of $x - ct$ for wave speed c (negative speed c caters for $x + c_* t$ case). In particular, generally seek solutions

$$u = \cos[k(x - ct)] = \cos(kx - \omega t) \quad (\text{or } \sin() \text{ or } \exp[i()])$$

for **wavenumber** k , **frequency** ω , and **wave speed** $c = \omega/k$.

Then the wavelength is $L = 2\pi/k$ and wave period is $T = 2\pi/\omega$.

Example 1.3 Substitute $u = \cos(kx - \omega t)$ into the wave PDE $u_{tt} = c_*^2 u_{xx}$ to determine such wave solutions for frequency $\omega = \pm c_* k$; that is, for wave speed $c = \omega/k = \pm c_*$. 

Example 1.4 Substitute $u = \cos(kx - \omega t)$ into the fourth order beam PDE $u_{tt} + Eu_{xxxx} = 0$ to determine a wave solution for frequency $\omega = \pm \sqrt{E}k^2$; that is, for wave speed $c = \omega/k = \pm \sqrt{E}k$. 

Example 1.5 Substitute $u = \cos(kx - \omega t)$ into the heat PDE $u_t = Ku_{xx}$ and find $\omega \sin(kx - \omega t) = -Kk^2 \cos(kx - \omega t)$ which has no solution for all x, t , and hence has no travelling waves. Alternatively, try $u = e^{i(kx - \omega t)}$ to find $\omega = -iKk^2$ is unallowably complex. 

Definition 1.2 (dispersion relation). *A linear PDE for $u(x, t)$ supports travelling waves if $u = \cos(kx - \omega t)$, or $u = e^{i(kx - \omega t)}$, is a solution for frequency determined by a (real) dispersion relation $\omega = \omega(k)$. The wave speed is then $c = \omega/k$.*

Example 1.6 (water waves). Later we derive that the dispersion relation for waves on water of depth d is $\omega^2 = gk \tanh(kd)$ where g is gravity.

- So for long waves (tsunamis) or shallow water (the beach) kd is small and the relation is near linear $\omega \approx \pm\sqrt{gdk}$ giving wave speed $c \approx \sqrt{gd}$, constant.
- For shorter waves in relatively deep water, kd large, $\omega \approx \pm\sqrt{gk}$ so wave speed $c = \sqrt{g/k}$ varies and is fastest for longer waves, smaller k , and slower for short waves, larger k .



Exercise 1.5 Recall the dispersion relation of a wave PDE is determined by seeking solutions of the form $\cos(kx - \omega t)$ or $e^{i(kx - \omega t)}$. Determine the dispersion relations for the given PDEs, if one exists.

- What is the dispersion relation for the Klein–Gordon PDE $u_{tt} + 2u - 3u_{xx} = 0$?
- What is the dispersion relation for the Korteweg–de Vries PDE $u_t + 6uu_x + u_{xxx} = 0$ for the wave dynamics linearised about the state $u_* = 1/3$?
- What is the dispersion relation for the Schrodinger PDE $-i\psi_t = \hbar\psi_{xx}$?



2 Separation of variables

Chapter Contents

2.1	Linearity empowers analysis	14
2.2	Separation of variables generates boundary value problems	15
2.2.1	The wave PDE separates	15
2.2.2	Model physical boundary conditions	17
2.2.3	Summary of separation of variables	17
2.2.4	Linearised fish harvesting	18
2.2.5	Circularly symmetric problems	18

2.1 Linearity empowers analysis

Linearity helps enormously in understanding solutions $u(x, t)$ of (Kreyszig 2011, pp.541–2)

Recall we could not (yet) solve the car traffic equation $\rho_t + (1 - \rho/70)\rho_x = 0$ because it is nonlinear, but D'Alembert solves $\rho_t + c_*\rho_x = 0$ because it is linear. Equivalently: introducing the operator $\mathcal{L} := \partial_t + c_*\partial_x$, we can solve $\mathcal{L}\rho = 0$ because the operator \mathcal{L} is linear.

Definition 2.1. An operator \mathcal{L} is called **linear** if $\mathcal{L}(c_1u_1 + c_2u_2) = c_1\mathcal{L}u_1 + c_2\mathcal{L}u_2$ for all constants c_j and all functions $u_j(x, t)$ in the domain of \mathcal{L} (usually the domain is smooth functions). A PDE for $u(x, t)$ is called

- **linear** if it can be written in the form $\mathcal{L}u = f(x, t)$;
- iff $f = 0$ everywhere, then the PDE is **homogeneous**, otherwise **inhomogeneous**.

Example 2.1 • The heat equation is $\mathcal{L}u = 0$ (homogeneous) for linear operator $\mathcal{L} = \partial/\partial t - K\partial^2/\partial x^2$.

- Equations of ideal gas dynamics are nonlinear as there are product terms such as ρu in the continuity PDE, and uu_x and $\rho^{\gamma-1}\rho_x$.
- The wave equation is linear as the operator $\mathcal{L} = \partial^2/\partial t^2 - c^2\partial^2/\partial x^2$ is linear.



Lemma 2.2 (superposition). *If $u_j(x, t)$ are solutions of a homogeneous linear PDE, $\mathcal{L}u_j = 0$, then so is the linear combination $u(x, t) = \sum_j c_j u_j(x, t)$ for all constants c_j .*

Exercise 2.1 Prove Lemma 2.2 by induction on the number of solutions.



2.2 Separation of variables generates boundary value problems

Linear PDEs naturally give rise to ordinary differential, eigenvalue problems over some domain in space: these are eigenvalue boundary value problems. (Haberman 1987, Ch. 2)

Well dressed PDEs have boundary conditions Here BCS appear, later we see that they are an essential component of a PDE.

2.2.1 The wave PDE separates

(Kreyszig 2011, §12.5–6) Consider the waves in a 1D domain, such as gas in a tube (Haberman 1987, §2.2)

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \quad \text{such that} \quad u(0, t) = u(L, t) = 0$$

and with initial condition $u(x, 0) = u_0(x)$ and $u_t(x, 0) = 0$ say.

Now introduce dots for time derivatives and dashes for ‘space’ derivatives. **Separation of variables** Seek solutions in the form $u(x, t) = X(x)\mathcal{T}(t)$. Substitute into the PDE and rearrange to

$$\frac{\ddot{\mathcal{T}}}{c^2 \mathcal{T}} = \frac{X''}{X}$$

The two sides are functions of independent variables, hence the two sides must be constant, say $-\lambda$. Thus $u(x, t) = X(x)\mathcal{T}(t)$ will be a PDE solution if

$$\ddot{\mathcal{T}} = -c^2 \lambda \mathcal{T} \quad \text{and} \quad X'' = -\lambda X.$$

Space dependent equation This is another old friend from level II, the constant coefficient $d^2X/dx^2 + \lambda X = 0$. But we need two extra conditions for a second order ODE: $X(0) = X(L) = 0$.

This ODE arises with an as yet unknown parameter λ . It is an eigenproblem (i.e., first year linear algebra).

Four cases arise:

$\lambda < 0$: Say $\lambda = -k^2$ (for $k > 0$). only trivial solutions. Of no interest.

$\lambda = 0$: Of no interest.

$\lambda > 0$: Say $\lambda = k^2$ (for $k > 0$). $X_n(x) = \sin k_n x$ is a nontrivial solution, called *eigenfunction*, corresponding to the *eigenvalue* $\lambda_n = k_n^2 = n^2\pi^2/L^2$.

λ complex : no nontrivial solutions. Of no interest.

Time dependent equation For $\lambda_n = k_n^2 = n^2\pi^2/L^2 > 0$ the ODE $\ddot{T} + c^2\lambda T = 0$ is our old friend with trigonometric solutions: which determines that the corresponding $T_n(t) = \cos(ck_n t)$.

Assemble a general solution Some solutions to the homogeneous heat PDE and homogeneous BCs are in the product form

$$u_n(x, t) = \sin(k_n x) \cos(ck_n t) = \sin\left(\frac{n\pi x}{L}\right) \cos\left(\frac{cn\pi t}{L}\right).$$

These are called the **normal modes** of the system.

A general solution of the linear PDE and BCs is the general linear combination, with as yet arbitrary coefficients b_n ,

$$u(x, t) = \sum_{n=1}^{\infty} b_n u_n(x, t) = \sum_{n=1}^{\infty} b_n \sin(k_n x) \cos(ck_n t). \quad (2.1)$$

Two difficulties

- This general solution is an infinite sum with all the tricky delicateness of infinite sums.
- How can we be sure that these are the only eigenvalues and eigenfunctions.

Finally consider (remaining) initial condition Given the general solution in the form (2.1) let's satisfy the initial condition $u(x, 0) = u_0(x)$. That is, we need to choose coefficients b_n so that

$$u_0(x) = \sum_{n=1}^{\infty} b_n \sin(k_n x) \cos(ck_n 0) = \sum_{n=1}^{\infty} b_n \sin k_n x.$$

Example 2.2 If the initial distribution $u_0(x) = u(x, 0) = 4 \sin(3\pi x/L) + 7 \sin(8\pi x/L)$, then satisfy the initial condition by choosing $b_3 = 4$, $b_8 = 7$ and all other $b_n = 0$. Hence the solution is $u(x, t) = 4 \sin(3\pi x/L) \cos(c3\pi t/L) + 7 \sin(8\pi x/L) \cos(c8\pi t/L)$.

♣

Careful: distinguish $u_0(x)$, the initial distribution, from $u_n(x, t)$, the n th homogeneous solution.

(Kreyszig 2011, §11.2)

More generally we need to express $u_0(x)$ as an infinite sum of sines. Quite general functions are representable as a Fourier sine series (unique). For a sine-series, the coefficients

$$b_n = \frac{2}{L} \int_0^L u_0(x) \sin \frac{n\pi x}{L} dx.$$

Example 2.3 What is the solution if $u(x, 0) = 1$, $0 < x < L$ (see Figure 2.1).

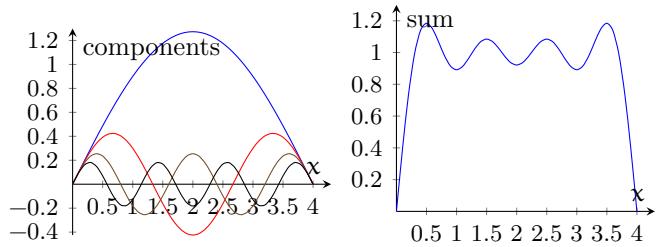


Figure 2.1: approximation of a constant (one) via a Fourier sine series.

Interpret: all modes oscillate perpetually in time at each different frequency; in practice, small damping/radiation will eventually dissipate all modes. ♣

Exercise 2.2 Just occasionally separation of variables finds special solutions of nonlinear partial differential equations. Use separation of variables to find a class of solutions to

$$\frac{\partial u}{\partial t} = u \frac{\partial^2 u}{\partial x^2} \quad \text{such that } u(0, t) = u(1, t) = 0.$$

- (a) Separation of variables, $u = X(x)T(t)$, leads to what pair of ordinary differential equations?
- (b) Hence find solutions of these ordinary differential equations, accounting for the boundary conditions.
- (c) Hence write down solutions of the original partial differential equation.

♣

2.2.2 Model physical boundary conditions

The previous arbitrarily used BCs $u(0, t) = u(L, t) = 0$. For u denoting velocity these BCs are appropriate to (solid) **closed ends** of gas in a tube.

What if the tube has **open ends**? $u_x(0, t) = u_x(L, t) = 0$.

Many musical instruments—such as flutes, clarinets and trumpets—have one end closed and the other end open: $u(0, t) = u_x(L, t) = 0$.

2.2.3 Summary of separation of variables

1. Start with a linear homogeneous PDE with homogeneous and linear BCs.
2. Delay considering inhomogeneous initial (boundary) conditions until the last step.

3. Separate variables in terms of a separation constant.
4. Determine separation constant as eigenvalues of a boundary value problem (BVP).
5. Solve the companion ODE(s).
6. Generally superpose all the product form solutions obtained.
7. Satisfy the inhomogeneous initial conditions, typically as some generalised sort of Fourier series.

This framework *adapts* to many other linear PDE problems.

2.2.4 Linearised fish harvesting

Fish breeding, harvesting and spreading in one space dimension, with density of fish $u(x, t)$, may be modelled by the PDE

$$\frac{\partial u}{\partial t} = A(x)u + K \frac{\partial^2 u}{\partial x^2}$$

where $A(x)$ is the net growth rate of breeding and harvesting, and u_{xx} represents spreading due to random swimming and eddies. Pose ‘insulating’ BCs of “zero flux” that $u_x(0, t) = u_x(L, t) = 0$.

Seek solutions by separation of variables: $u = X(x)T(t)$. The PDE becomes

$$\frac{\dot{T}}{T} = \frac{A(x)X + KX''}{X} = -\lambda.$$

That is, $\dot{T} = -\lambda T$ with solution $T \propto e^{-\lambda t}$, and

$$KX'' + [A(x) + \lambda]X = 0.$$

This ODE to be solved with BCs $X'(0) = X'(L) = 0$.

Challenge: as the ODE has a variable coefficient, we can no longer use exponential or trigonometric functions as solutions for the spatial structure. We need to understand the general behaviour of solutions of the ODE in order to deal with the PDE in general.

2.2.5 Circularly symmetric problems

(Haberman 1987, §5.2.2) Consider the cooling of a disc of material

$$\frac{\partial u}{\partial t} = K\nabla^2 u \quad \text{in } x^2 + y^2 < r_o^2;$$

or the vibrations of a circular drum ($K = c^2$)

$$\frac{\partial^2 u}{\partial t^2} = K\nabla^2 u \quad \text{in } x^2 + y^2 < r_o^2;$$

both with $u = 0$ on the circular boundary. Change to polar coordinates (r, θ) . For brevity assume no dependence upon θ . From

multivariable calculus (Kreyszig 2011, §12.10, e.g.), the PDE becomes

$$\frac{\partial^2 u}{\partial t^2} \text{ or } \frac{\partial u}{\partial t} = \frac{K}{r} \frac{\partial}{\partial r} \left(r \frac{\partial u}{\partial r} \right).$$

Try separation of variables again: seek $u(r, t) = R(r)T(t)$, then

$$\frac{\ddot{T}}{KT} \text{ or } \frac{\dot{T}}{KT} = \frac{1}{rR} (rR')' = -\lambda.$$

That is, $T(t)$ is trigonometric or exponential, and

$$(rR')' + \lambda rR = rR'' + R' + \lambda rR = 0, \quad R(r_0) = 0,$$

(Kreyszig 2011, §12.10)

and, *implicitly*, $R(r)$ well behaved at $r = 0$. Instead in Level II you were introduced to using Bessel functions and Fourier–Bessel series for such a problem.

Fourier Series and Fourier–Bessel series are two special cases of a wide class of series that are fundamental in representing complicated spatial structures. The next module explores this general class.

3 Wonderful Sturm–Liouville boundary value problems

Chapter Contents

3.1	Self-adjoint operators form Sturm–Liouville problems	23
3.1.1	Eigenfunctions are orthogonal	25
3.1.2	Eigenvalues are real	25
3.1.3	Eigenfunctions are often unique	25
3.1.4	The Rayleigh quotient approximates: optional	26
3.2	Eigenfunctions expand inhomogeneous solutions . .	27

The second module showed that to use separation of variables to solve and understand solutions to PDEs we need to solve BVPs with non-constant coefficients in the ODE. They all are in the form of a two point, boundary value problem, with an eigenvalue. We start (Haberman 1987, §5.3), by exploring the regular *Sturm–Liouville eigenproblem*.
(Kreyszig 2011, §11.5)

Almost all sections of this module are first year linear algebra, in a calculus disguise.

Definition 3.1 (Regular Sturm–Liouville eigenvalue problem). *The regular Sturm–Liouville eigenvalue problem for $u(x)$ consists of the ODE*

$$\frac{d}{dx} \left(p(x) \frac{du}{dx} \right) + [q(x) + \lambda r(x)] u = 0, \quad a < x < b, \quad (3.1)$$

subject to BCS of the form

$$\alpha_1 u(a) + \alpha_2 u'(a) = 0 \quad \text{and} \quad \beta_1 u(b) + \beta_2 u'(b) = 0.$$

The coefficient functions are all real and $p, r > 0$ for $a \leq x \leq b$.

Exercise 3.1 Consider the differential equation in non-Sturm–Liouville form

$$\frac{d^2u}{dx^2} + \alpha(x) \frac{du}{dx} + [\beta(x) + \lambda \gamma(x)] u = 0.$$

Multiply this differential equation by $H(x)$. Determine $H(x)$ such that the differential equation transforms to Sturm–Liouville form

$$\frac{d}{dx} \left[p(x) \frac{du}{dx} \right] + [q(x) + \lambda r(x)] u = 0.$$

- (a) Determine the necessary multiplying function $H(x)$.

-
- (b) In terms of $\alpha(x)$, $\beta(x)$ and $\gamma(x)$, what is $q(x)$?
(c) Similarly deduce the equivalent Sturm–Liouville form of the differential equation

$$\frac{d^2(xu)}{dx^2} + \lambda u = 0.$$



- Theorem 3.2** (great properties).
1. All eigenvalues λ are real.
 2. There are a countably infinite number of eigenvalues $\lambda_1 < \lambda_2 < \lambda_3 < \dots$ (some could be negative):
 - (a) there is a smallest eigenvalue (say λ_1 , sometimes λ_0);
 - (b) there is not a largest eigenvalue and $\lambda_n \rightarrow \infty$ as $n \rightarrow \infty$.
 3. Corresponding to each eigenvalue λ_n , there is an eigenfunction $v_n(x)$ (unique up to scaling), and which has exactly $n - 1$ zeros for $a < x < b$.
 4. The set of eigenfunctions $\{v_n(x)\}$ is complete: any piecewise smooth function $f(x)$ can be written as a generalised Fourier series, the convergent sum

$$f(x) = \sum_{n=1}^{\infty} a_n v_n(x) \quad \left(= \frac{1}{2}[f(x-) + f(x+)] \right).$$

5. Eigenfunctions corresponding to different eigenvalues are orthogonal relative to the weight function $r(x)$:

$$\int_a^b v_n(x) v_m(x) r(x) dx = 0 \quad \text{if } m \neq n.$$

Example 3.1 Return to the spatial structure of heat diffusion or wave PDE. Separation of variables led to $X'' + \lambda X = 0$ such that $X(0) = X(L) = 0$. This is a regular Sturm–Liouville problem. Recall:

- (a) eigenvalues are $\lambda_n = n^2\pi^2/L^2$ (real);
- (b) countably infinite, with smallest π^2/L^2 , but no largest;
- (c) eigenfunctions are $v_n = \sin(n\pi x/L)$ which has $n - 1$ zeros in $0 < x < L$;
- (d) eigenfunctions lead to the Fourier sine series which we know is complete;
- (e) orthogonal with respect to the weight function 1.



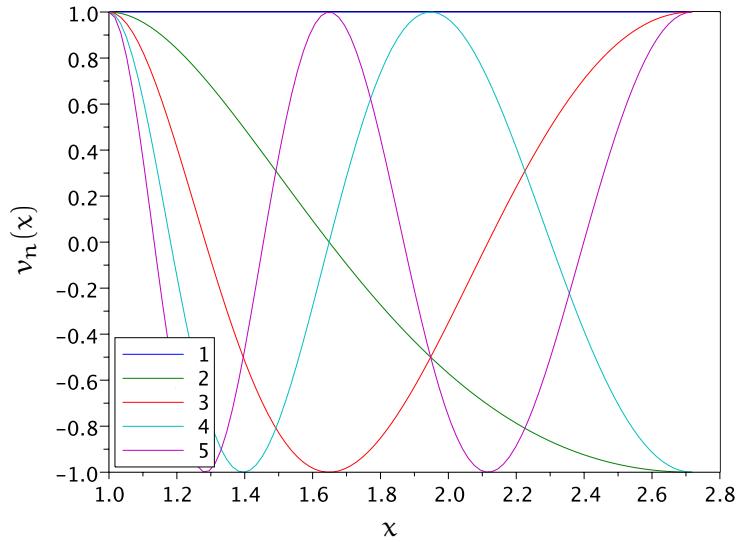


Figure 3.1: eigenfunctions of Example 3.2.

Example 3.2 (Euler–Cauchy). Consider the Sturm–Liouville system

$$(xu')' + (\lambda/x)u = 0 \quad \text{such that} \quad u'(1) = u'(e) = 0.$$

- (a) Seek solutions $u = x^r$
- (b) Eigenfunctions $v_n = \cos(k_n \log x)$.
- (c) Eigenfunctions are orthogonal relative to the weight function $1/x$



These great properties often hold for non-regular Sturm–Liouville problems.

(Kreyszig 2011, §5.2)

Example 3.3 (Legendre polynomials). Recall the Legendre ODE $(1-x^2)u'' - 2xu' + k(k+1)u = 0$. This ODE may be written as

$$\frac{d}{dx} \left[(1-x^2) \frac{du}{dx} \right] + \lambda u = 0 \quad \text{for} \quad -1 < x < 1.$$

The ODE is in Sturm–Liouville form, but it is not a regular problem. Nonetheless recall:

- (a) there exist nontrivial solutions for eigenvalues $\lambda_n = n(n-1)$ for $n = 1, 2, 3, \dots$;
- (b) countably infinite, smallest but no largest;

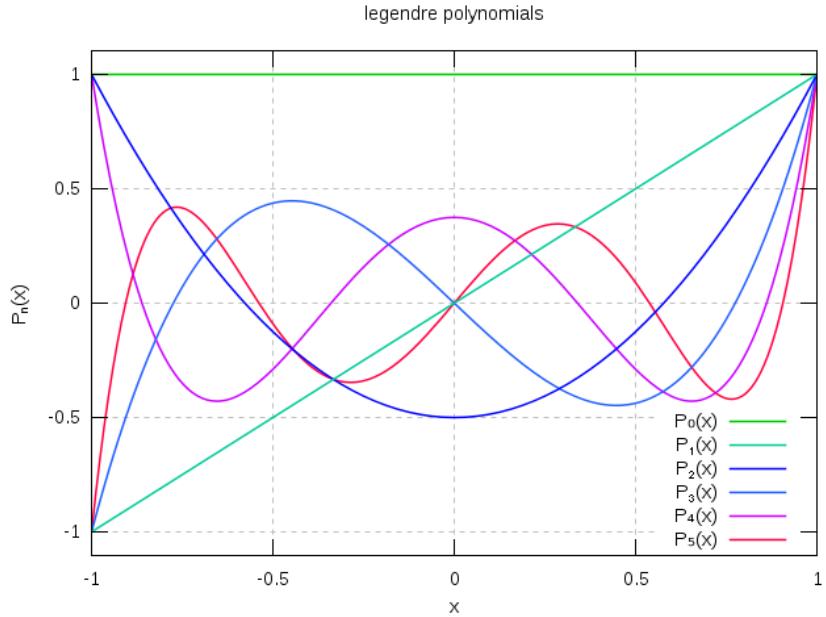


Figure 3.2: the first six Legendre polynomials [Wikipedia, 2011].

(c) eigenfunctions are Legendre polynomials

$$\begin{aligned} v_1 &= P_0(x) = 1, & v_2 &= P_1(x) = x, \\ v_3 &= P_2(x) = \frac{1}{2}(3x^2 - 1), & v_4 &= P_3(x) = \frac{1}{2}(5x^3 - 3x), \\ &\dots; \end{aligned}$$

(d) they are orthogonal (constant weight) over the domain

$$\int_{-1}^1 v_n v_m dx = \frac{2}{2n-1} \text{ if } m = n, \text{ otherwise } 0;$$



3.1 Self-adjoint operators form Sturm–Liouville problems

We prove some of the results for regular Sturm–Liouville eigenproblems. In this section define the linear operator
(Haberman 1987, §5.5)

$$\mathcal{L} = \frac{d}{dx} \left(p(x) \frac{du}{dx} \right) + q(x), \quad \text{so that} \quad \mathcal{L}u + \lambda r(x)u = 0$$

is the Sturm–Liouville ODE. To make the definition of \mathcal{L} complete, at first assume the operator \mathcal{L} has the BCs of the regular Sturm–Liouville eigenproblem:

$$\alpha_1 u(a) + \alpha_2 u'(a) = 0 \quad \text{and} \quad \beta_1 u(b) + \beta_2 u'(b) = 0.$$

The coefficient functions are all real and $p, r > 0$ for $a \leq x \leq b$. I also comment on variations on these BCs.



Figure 3.3: street name in Paris.

Lagrange's identity is crucial namely that

$$u\mathcal{L}v - v\mathcal{L}u = \frac{d}{dx} [p(uv' - vu')] .$$

Now integrate to obtain the integral form of *Lagrange's identity*, also known as *Green's formula*,

$$\int_a^b u\mathcal{L}v - v\mathcal{L}u \, dx = [p(uv' - vu')]_a^b . \quad (3.2)$$

Definition 3.3. A linear operator \mathcal{L} (with its BCs) is termed self-adjoint on a domain D if $\int_D u\mathcal{L}v \, dx = \int_D v\mathcal{L}u \, dx$ for all smooth functions u and v that satisfy the BCs of the operator \mathcal{L} .

The differential operator \mathcal{L} in a Sturm–Liouville ODE becomes self-adjoint when the BCs are such that $[p(uv' - vu')]_a^b = 0$. The BCs are crucial to the definition of a differential operator.

The following are three examples.

- In a slightly restricted regular Sturm–Liouville problem the BCs are $u'(a) = \alpha u(a)$ and $u'(b) = \beta u(b)$.
- Consider the ‘periodic’ BCs when $u(b) = u(a)$ and $u'(b) = u'(a)$, and $p(b) = p(a)$:
- For Legendre polynomials $p = 1 - x^2 \rightarrow 0$ as $x \rightarrow \pm 1$:

Exercise 3.2 The concept of a self-adjoint operator applies much more widely than just second order differential operators. Here consider the operator

$$\mathcal{L} = \frac{d^2}{dx^2} \left(p(x) \frac{d^2}{dx^2} \right)$$

on the domain $a < x < b$ for some given $p(x)$. Use integration by parts on $\int_a^b u\mathcal{L}v - v\mathcal{L}u \, dx$ to derive a new version of the integral Lagrange's identity suitable for this operator. Then determine some boundary conditions which together make the operator self-adjoint.



3.1.1 Eigenfunctions are orthogonal

Now prove one part of Theorem 3.2. Let there be two eigenfunctions v_n and v_m corresponding to distinct eigenvalues λ_n and λ_m . They satisfy

$$\mathcal{L}v_n + \lambda_n r(x)v_n = 0 \quad \text{and} \quad \mathcal{L}v_m + \lambda_m r(x)v_m = 0.$$

Since $(\lambda_n - \lambda_m) \int_a^b v_n r v_m dx = 0$ and the eigenvalues are distinct, $\lambda_n - \lambda_m \neq 0$, consequently $\int_a^b v_n v_m r dx = 0$. That is, the eigenfunctions are orthogonal with respect to the weight function.

3.1.2 Eigenvalues are real

Prove by contradiction. Suppose there exists an eigenvalue λ with non-zero imaginary part so that $\bar{\lambda} \neq \lambda$. By orthogonality $\int_a^b v \bar{v} r dx = \int_a^b |v|^2 r dx = 0$. Since weight function $r > 0$, this integral cannot be zero for non-trivial v . Contradiction. Hence all eigenvalues are real.

Exercise 3.3 Use Lagrange's identity applied to self-adjoint Sturm–Liouville problems to prove that there cannot be any generalised eigenfunctions: such generalised eigenfunctions w are defined as satisfying $\mathcal{L}w + \lambda r w = rv$ and the appropriate BCS.

- (a) The eigenvalue problem satisfied by the eigenfunction v , subject to boundary conditions, is what ODE.
- (b) For a self-adjoint Sturm–Liouville problem, Lagrange's identity applied to v and w then leads to what identity?
- (c) This is a contradiction because, for all $a < x < b$, what?



3.1.3 Eigenfunctions are often unique

Suppose there are two eigenfunctions v and w corresponding to the same eigenvalue λ : $\mathcal{L}v + \lambda r v = 0$ and $\mathcal{L}w + \lambda r w = 0$. Consider the following using the differential form of Lagrange's identity for a Sturm–Liouville ODE: Deduce $p(vw' - wv') = \text{constant}$ over the domain $a \leq x \leq b$. When any one BC is such that this constant is zero, each eigenfunction must be a trivial scalar multiple of the other.

In what circumstances do BCS assure $p(vw' - wv') = 0$? When either BC is regular, or when $p \rightarrow 0$ at either boundary. In such cases the eigenfunctions are unique up to a scalar multiple.

This completes Figure 3.4.

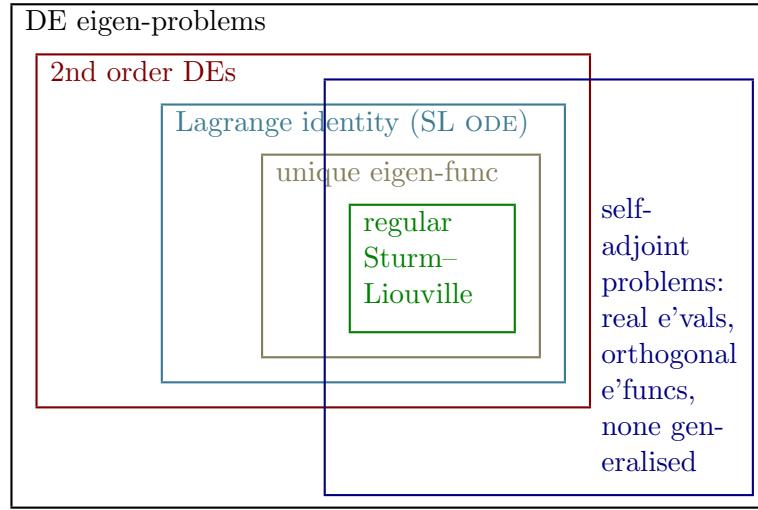


Figure 3.4: Venn diagram of the scenario of this section, and the results proved. More is true than proved: for example, unique eigenfunctions holds for more self-adjoint problems than just 2nd order.

3.1.4 The Rayleigh quotient approximates: optional

The Rayleigh quotient is typically used to approximate well the eigenvalue given only a rough approximation to the eigenfunction v_n .

Theorem 3.4 (Rayleigh quotient). *Consider the Sturm–Liouville ODE (3.1).*

- For any given eigenfunction $v_n(x)$, the Rayleigh quotient determines the corresponding eigenvalue

$$\lambda_n = \frac{-[pv_nv'_n]_a^b + \int_a^b [pv'_n]^2 - qv_n^2 dx}{\int_a^b v_n^2 r dx}.$$

- If $q(x) \leq 0$ for $a \leq x \leq b$, and the BCS are such that $[pv_nv'_n]_{x=a}^b \leq 0$, then all eigenvalues $\lambda_n \geq 0$.

Example 3.4 Recall the Euler–Cauchy Sturm–Liouville eigenproblem of Example 3.2

$$(xu')' + (\lambda/x)u = 0 \quad \text{such that} \quad u'(1) = u'(e) = 0.$$

Roughly estimate the leading three eigenvalues from the Rayleigh quotient. ♣

Example 3.5 Consider the Sturm–Liouville problem $X'' + \lambda X = 0$ with BCS $X(a) + \alpha X'(a) = 0$ and $X(b) + \beta X'(b) = 0$. For what values of α and β are we assured that all eigenvalues are non-negative? ♣

Normal modes of vibration Recall that we envisage that the Sturm–Liouville ODE arises by separation of variables. For example, it might arise from the wave PDE

$$r(x) \frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x} \left(p(x) \frac{\partial u}{\partial x} \right) + q(x)u.$$

Then separation of variables seeks solutions in the form $u = X(x)T(t)$ to find

$$(pX')' + [q + \lambda r]X = 0 \quad \text{and} \quad \ddot{T} + \lambda T = 0.$$

Then the Sturm–Liouville ODE determines certain spatial structures correspond to real eigenvalues λ_n , $n = 1, 2, 3, \dots$. The time ODE then determines the frequency of the corresponding oscillation is $\omega_n = \pm\sqrt{\lambda_n}$. Hence in wave systems on bounded domains we typically expect a discrete set of infinite frequencies $\omega_n = \sqrt{\lambda_n}$ for $n = 1, 2, 3, \dots$. These are the **normal modes** of vibrations on the domain.

Importantly, the Rayleigh quotient helps us to know when all eigenvalues $\lambda_n \geq 0$ as then the frequencies are all real. In this class of problems, if $\lambda_n < 0$ then a mode grow exponentially in time predicting ‘physical explosion’!

3.2 Eigenfunctions expand inhomogeneous solutions

(related to Haberman
SS7.2–7.3)

Recall that in separation of variables we used the Fourier expansion of the eigenfunctions to transform initial conditions into a form we could use. Similarly, eigenfunctions cater for inhomogeneous BVPs. Suppose we wish to solve, for some *given* number α , the inhomogeneous BVP $\mathcal{L}u + \alpha r(x)u = f(x)$ subject to BCS that make the operator \mathcal{L} self-adjoint.

Consider the Sturm–Liouville eigenproblem $\mathcal{L}v + \lambda rv = 0$. It has eigenvalues λ_n and corresponding eigenfunctions $v_n(x)$. The eigenfunctions are complete (not proved) which means we can expand the solution as

$$u(x) = \sum_{n=1}^{\infty} c_n v_n(x).$$

In essence this is a linear algebra change of basis: instead of viewing the solution as a ‘sum of things’ at each x , we view it as a sum over the eigenfunctions.

Suppose we have somehow written the right-hand side of the BVP in terms of eigenfunctions:

$$\mathcal{L}u + \alpha r(x)u = f(x) = r(x) \sum_{n=1}^{\infty} f_n v_n(x).$$

Substitute the eigenfunction expansion for solution $u(x)$ into this inhomogeneous ODE, with α appearing as the given parameter. That is, given appropriate Sturm–Liouville BCS, the *eigenfunction solution* to the inhomogeneous BVP

$$\mathcal{L}u + \alpha ru = f(x) \quad \text{is} \quad u(x) = \sum_{n=1}^{\infty} \frac{f_n}{\alpha - \lambda_n} v_n(x).$$

This is the eigenfunction solution of the ODE.

Fredholm alternative

Unique solution : Generally.

No solution : Except for cases where there exists a mode m for which eigenvalue $\lambda_m = \alpha$.

Infinite number of solutions : Unless $f_m = 0$ also.

The above is classic linear algebra for a system of linear equations.

Example 3.6 Solve $(xu')' + (\pi^2/x)u = \frac{1}{x} \sin^2[\pi \log x]$ such that $u'(1) = u'(e) = 0$.

Recall that the Sturm–Liouville problem

$$(xv')' + (\lambda/x)v = 0 \quad \text{such that} \quad v'(1) = v'(e) = 0,$$

has eigenvalues $\lambda_n = (n-1)^2\pi^2$ and corresponding eigenfunctions $v_n(x) = \cos[(n-1)\pi \log x]$, $n = 1, 2, 3, \dots$. ♣

Exercise 3.4 You are given that the Sturm–Liouville problem $X'' + \lambda X = 0$ such that $X(0) = X(\pi) = 0$ has eigenvalues $\lambda_n = n^2$ and eigenfunctions $v_n(x) = \sin nx$ for integer $n = 1, 2, 3, \dots$.

- (a) How many solutions does the inhomogeneous differential equation $X'' + 3X = \sin(2x)$ such that $X(0) = X(\pi) = 0$ have?
- (b) How many solutions does the inhomogeneous differential equation $X'' + 4X = \sin(x) + 2 \sin(3x)$ such that $X(0) = X(\pi) = 0$ have?
- (c) How many solutions does the inhomogeneous differential equation $X'' + 9X = -\sin(x) + 3 \sin(2x) - \sin(3x)$ such that $X(0) = X(\pi) = 0$ have?



The general eigenfunction expansion of the right-hand side
 We seek coefficients f_n such that

$$f(x) = \sum_{n=1}^{\infty} r(x)f_n v_n(x).$$

Now integrate over the domain: for brevity use inner product notation, with weight function $r(x)$,

$$\langle u, v \rangle = \int_a^b u v r \, dx.$$

By orthogonality $\langle v_m, v_n \rangle = 0$ when $m \neq n$. Take $\int_a^b v_m \cdot \, dx$ of both sides of the equation:

$$\langle v_m, f/r \rangle = f_m \langle v_m, v_m \rangle$$

Hence the coefficients in the expansion of the right-hand side are

$$f_m = \frac{\langle v_m, f/r \rangle}{\langle v_m, v_m \rangle}.$$

4 Discretise 1D space

Chapter Contents

4.1	Lagrange's theorem underpins the method of lines	30
4.1.1	Ensure a discretisation is consistent with the PDE	38
4.1.2	Operator algebra generates discretisations	39
4.2	Find equilibria	40
4.3	Numerical linearisation characterises solution dynamics	42
4.4	PDE-free patch dynamics	50
4.5	May use Octave or Scilab	52

We aim to be able to computationally solve most PDES knowing something about the typical structures found in separation of variables and Sturm–Liouville theory. We start with the heat PDE, $u_t = u_{xx}$, and then more complicated PDES.

4.1 Lagrange's theorem underpins the method of lines

Theorem 4.1 (Lagrange's remainder theorem, (Kreyszig 2011, p.691)). *For functions $f(x)$ differentiable $(P + 1)$ times in some domain,*

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2f''(x) + \cdots + \frac{1}{P!}h^P f^{(P)}(x) + R_P(x, h),$$

where the **remainder** $R_P(x, h) := \frac{1}{(P+1)!}h^{P+1}f^{(P+1)}(\xi)$ for some $\xi(x, h)$ such that $x \leq \xi \leq x + h$.

Example 4.1 $f''(x) \approx [f(x-h) - 2f(x) + f(x+h)]/h^2$ as seen in other courses. ♣

Example 4.2 Approximate solutions $u(x, t)$ to the heat PDE, $u_t = u_{xx}$, on $0 < x < 1$ with BCS $u(0, t) = a$ and $u(1, t) = b$. ♣

Example 4.3 Use finite differences to compute approximate solutions $u(x, t)$ to the heat PDE, $u_t = u_{xx}$, on $0 < x < 1$ with BCS $u(0, t) = 0$ and $u(1, t) = \sin 8t$. ♣



Figure 4.1: early computer tools: Napier's bones led to slide rules.

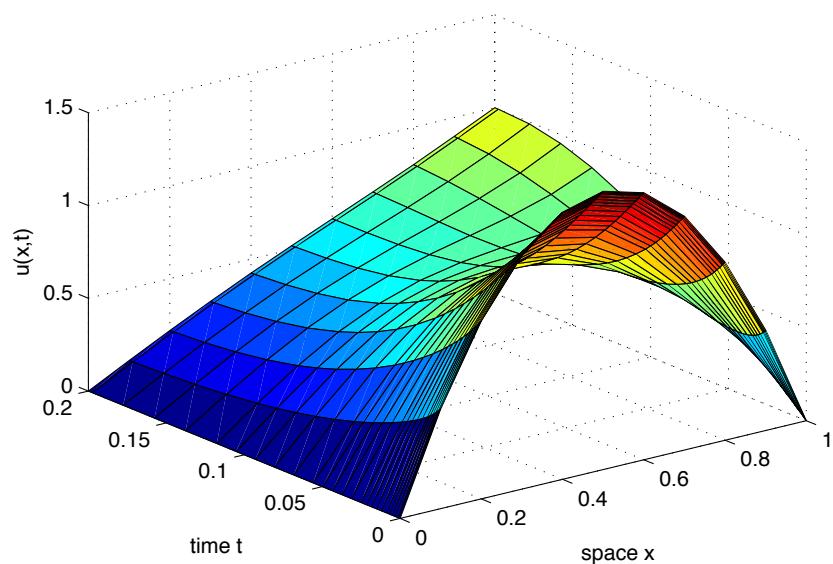


Figure 4.2: Algorithm 4.1 computes this solution of the heat PDE $u_t = u_{xx}$.

For this module we assume integrating in time is a solved task (*Modelling with ODEs*). So Algorithm 4.1 invokes `ode15s()` in Matlab (or in Octave invoke `lsode()` but interchange `u` and `t`, or in Scilab invoke `ode()`).

Algorithm 4.1: method of lines to solve the heat PDE. This framework adapts to many similar PDE problems. Note: every document must include title, author and date.

```

1 function heat
2 % Compute solutions of heat PDE with finite differences
3 % AJR, 13/8/2014
4 global j x
5 nPoints=11
6 x=linspace(0,1,nPoints)'
7 j=2:nPoints-1
8 u0=6*x.* (1-x)
9
10 [t,u]=ode15s(@dudt,[0 0.2],u0(j))
11
12 u=[uleft(t) u uright(t)]
13 surf(x,t,u)
14 xlabel('space\backslash x'), ylabel('time\backslash t'), zlabel('u(x,t)')
15 %-----
16 function ut=dudt(t,u)
17 global j x
18 dx=x(2)-x(1);
19 u=[uleft(t);u;uright(t)];
20 ut=(u(j-1)-2*u(j)+u(j+1))/dx^2;
21 %-----
22 function ua=uleft(t)
23 ua=zeros(size(t));
24 %-----
25 function ub=uright(t)
26 ub=zeros(size(t));

```

Algorithm 4.1 exhibits an important principle: code once, use several times.

Software This course specifically addresses Matlab, but Octave and Scilab are similar and may be used (see Section 4.5 which lists example code corresponding to Algorithm 4.1). Matlab is available

- in most computer laboratories,
- or purchase a Student Version,
- or use the university's Citrix server¹ to access via any personal e-device, anywhere and anytime.

Quarteroni & Saleri (2006) is a useful reading reference for reminding yourself of how to use Matlab/Octave. For example, chapter 1,

¹ <http://www.adelaide.edu.au/its/adapt/>

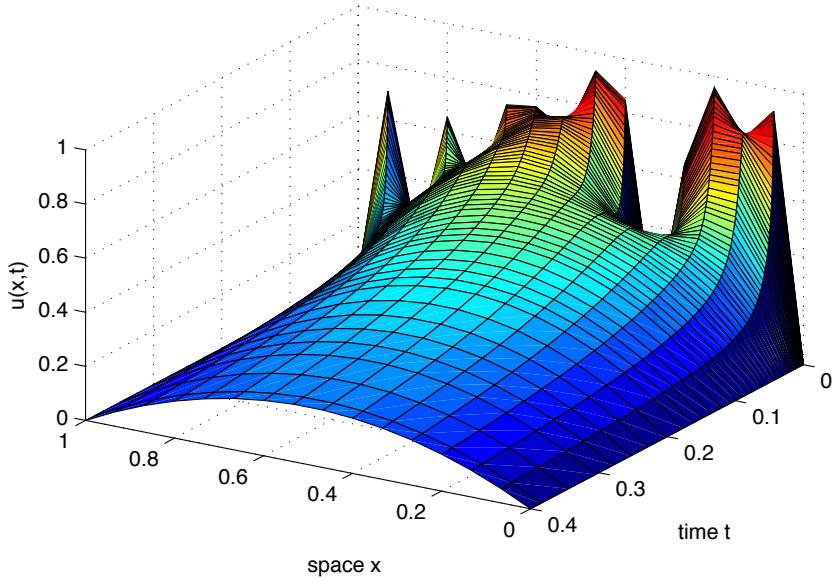


Figure 4.3: Algorithm 4.2 computes this solution of the nonlinear PDE $u_t = uu_{xx}$ showing the emergence of a slowly decaying parabola.

especially §1.7, introduces the Matlab language and functions.

Exercise 4.1 Recall Lagrange's theorem that

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2f''(x) + \cdots + \frac{1}{P!}h^P f^{(P)}(x) + R_P(x, h),$$

where the remainder $R_P(x, h) := \frac{1}{(P+1)!}h^{P+1}f^{(P+1)}(\xi)$ for some $\xi(x, h)$ such that $x \leq \xi \leq x + h$.

- (a) Apply Lagrange's theorem to e^x about $x = 0$ to deduce $e = 2\frac{2}{3}$ plus what remainder?
- (b) Consider the simplest ($N = 3$) spatial discretisation of the PDE $u_t = uu_{xx}$ for $0 < x < 1$ with boundary conditions $u(0, t) = u(1, t) = 0$ and initial condition $u(x, 0) = 4x^2(1 - x)$. What solution at $x = 1/2$ is predicted by this simplest spatial discretisation?
- (c) Consider the following Matlab/Octave code for a time derivative:

```
ut=sin(u(j)).*(u(j-1)-(2+dx^2)*u(j)+u(j+1))/dx^2+sin(x(j)).
```

For what PDES is it a reasonable spatial discretisation?



Example 4.4 Numerical methods are most valuable for nonlinear problems such as solving $u_t = uu_{xx}$ for $0 < x < 1$ and

boundary values $u(0, t) = u(1, t) = 0$. Recall from ‘homework’ that separation of variables finds the special solution $u(x, t) = x(1 - x)/(c + 2t)$: Algorithm 4.2 suggests that this is an emergent solution from all initial conditions.

Ongoing research in the School explores such emergence: how does such relatively simple structures emerge and evolve from complexity?



Algorithm 4.2: method of lines to solve the nonlinear PDE $u_t = uu_{xx}$.
Observe emergence of simple parabola.

```

1 function heatnon
2 % Compute solutions of nonlinear PDE via finite differences
3 % AJR, 13/8/2014
4 global j x
5 nPoints=16
6 x=linspace(0,1,nPoints)'
7 j=2:nPoints-1
8 u0=rand(size(x))
9
10 [t,u]=ode15s(@dudt,[0 0.4],u0(j))
11
12 u=[uleft(t) u uright(t)]
13 surf(x,t,u)
14 xlabel('space\backslash x'), ylabel('time\backslash t'), zlabel('u(x,t)')
15 %
16 function ut=dudt(t,u)
17 global j x
18 dx=x(2)-x(1);
19 u=[uleft(t);u;uright(t)];
20 ut=u(j).*(u(j-1)-2*u(j)+u(j+1))/dx^2;
21 %
22 function ua=uleft(t)
23 ua=zeros(size(t));
24 %
25 function ub=uright(t)
26 ub=zeros(size(t));

```

Example 4.5 (Korteweg–de Vries PDE). The nonlinear Korteweg–de Vries PDE $u_t + 6uu_x + u_{xxx} = 0$ partially models waves in shallow water and nonlinear optics. Let’s solve the PDE on $0 < x < 40$ with boundary conditions that the solution is 40-periodic. Start with an initial ‘lump’ of water for $0 < x < 4$ on depth otherwise $u(x, 0) = 2$. Figure 4.4 shows an emergent travelling solitary wave solution.

But how do we justify the discretisation of the Korteweg–de Vries PDE coded in Algorithm 4.3? (relevant to Lax’s theorem)
Answer: see next section.



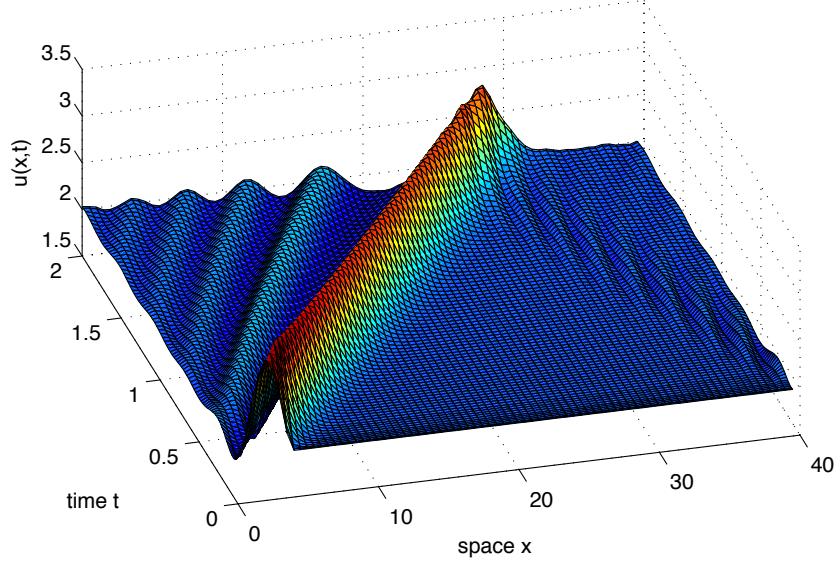


Figure 4.4: Algorithm 4.3 computes this solution of the nonlinear Korteweg–de Vries PDE $u_t + 6uu_x + u_{xxx} = 0$. A travelling solitary wave emerges. Very strange are the small waves in $20 < x < 40$ that each seem to travel forwards, but as a group appear to be travelling backwards—the next module explains.

Algorithm 4.3: method of lines to solve the nonlinear Korteweg–de Vries PDE $u_t + 6uu_x + u_{xxx} = 0$.

```

1 function kdv
2 % Compute solutions of Korteweg-de Vries PDE
3 % with periodic boundary conditions.
4 % AJR, 14/8/2014
5 global x
6 nPoints=80
7 x=linspace(0,40,nPoints+1); x=x(1:nPoints)';
8 u0=2+x.^2.*((1-x/4).^2.*((x<4));
9
10 [t,u]=ode15s(@dudt,[0 2],u0);
11
12 surf(x,t,u)
13 xlabel('space\backslash x'), ylabel('time\backslash t'), zlabel('u(x,t)')
14 %-----
15 function ut=dudt(t,u)
16 global x
17 dx=x(2)-x(1);
18 u=[u;u;u];
19 j=(1:length(x))+length(x);
20 ut=-3*(-u(j-1).^2+u(j+1).^2)/(2*dx) ...
21     -(-u(j-2)+2*u(j-1)-2*u(j+1)+u(j+2))/(2*dx.^3) ;

```

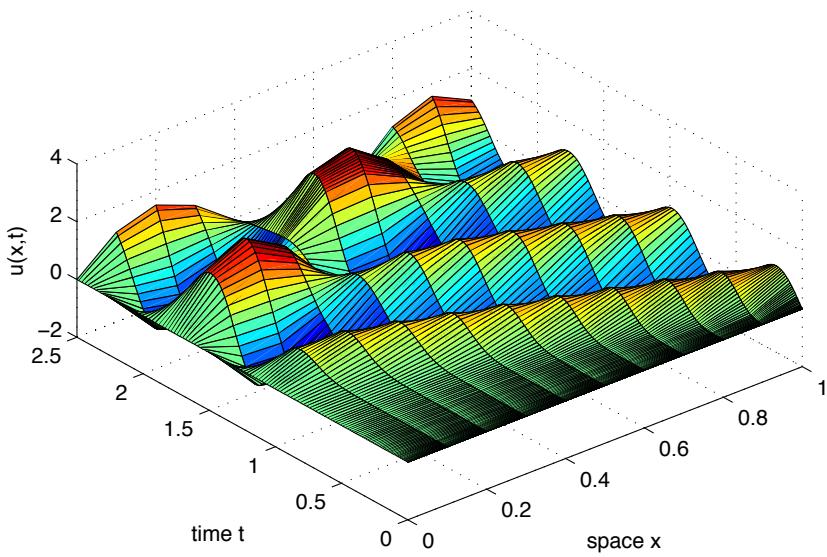


Figure 4.5: Algorithm 4.4 computes this solution of the Klein–Gordon wave PDE $u_{tt} = -u + u_{xx}$. Observe travelling waves reflecting and superposing.

Avoid loops There are no loops in any code of this module. Loops often ruin the execution of Matlab (Octave or Scilab), and harm the readability, flexibility and maintenance of code. Because BLAS and LAPACK exist, loops are often inefficient in large problems. Instead code on the high-level data structures of vectors and matrices. *Unnecessary loops will lose marks.*

Example 4.6 (optional—2nd order in time wave PDE). Let's solve the Klein–Gordon wave PDE $u_{tt} = -u + u_{xx}$ on $0 < x < 1$ with boundary conditions $u(0, t) = 0$ and $u(1, t) = \sin 8t$. Start with the equilibrium $u(x, 0) = u_t(x, 0) = 0$. Figure 4.5 shows travelling waves reflecting and superposing.

The trick here is to convert the single PDE that is second order in time derivatives, to a pair of PDEs that are first order in time derivatives. Define $v(x, t) := u_t(x, t)$, then the Klein–Gordon PDE may be written as

$$u_t = v, \quad v_t = -u + u_{xx}.$$

Discretise and then apply numerical integration routines like `ode15s`. ♣

Exercise 4.2 Confirm by hand or computer algebra some spatial discretisations and determine the leading errors through invoking various Taylor expansions.

- (a) By hand or computer algebra confirm that $(u(x + h) - u(x - h))/(2h)$ approximates the first derivative $u'(x)$ with what error?

Algorithm 4.4: method of lines to solve the Klein–Gordon wave PDE $u_{tt} = -u + u_{xx}$. Code the PDE as the pair $u_t = v$ and $v_t = -u + u_{xx}$ and pack into odd and even positions of the vector of unknowns.

```

1 function kgwave
2 % Compute solutions of Klein-Gordon wave PDE
3 % discretised in space. Store u in odd and v
4 % in even numbered entries in uv vectors.
5 % AJR, 17/8/2014
6 global j x
7 nPoints=11
8 x=linspace(0,1,nPoints)';
9 j=2:nPoints-1;
10 uv0=nan(2*length(j),1);
11 uv0(1:2:end)=0*x(j).*(1-x(j));
12 uv0(2:2:end)=0*x(j);
13
14 [t,uv]=ode15s(@duvdt,[0 2.5],uv0);
15
16 u=[uleft(t) uv(:,1:2:end) uright(t)];
17 surf(x,t,u)
18 xlabel('space\backslash x'), ylabel('time\backslash t'), zlabel('u(x,t)')
19 %-----
20 function uvt=duvdt(t,uv)
21 global j x
22 dx=x(2)-x(1);
23 u=[uleft(t);uv(1:2:end);uright(t)];
24 uvt=nan(2*length(j),1);
25 uvt(1:2:end)=uv(2:2:end);
26 uvt(2:2:end)=-u(j)+(u(j-1)-2*u(j)+u(j+1))/dx^2;
27 %-----
28 function ua=uleft(t)
29 ua=zeros(size(t));
30 %-----
31 function ub=uright(t)
32 ub=sin(8*t);

```

(b) By hand or computer algebra confirm that $(u(x+2h) - 3u(x+h) + 3u(x) - u(x-h))/h^3$ approximates the third derivative $u'''(x)$ with what error?

(c) By hand or computer algebra confirm that

$$\frac{u(x+h)u(x+2h) - [u(x+h) + u(x-h)]u(x) + u(x-h)u(x-2h)}{4h^2}$$

approximates what nonlinear second derivative term with what order of error?



4.1.1 Ensure a discretisation is consistent with the PDE

Computer algebra checks discretisations are consistent via some of the following steps in Matlab.

- Find the Taylor series of any given function such as the arcsinh , namely $\text{arcsinh } x = x - \frac{1}{6}x^3 + \frac{3}{40}x^5 + \mathcal{O}(x^6)$,

```
1 syms x
2 taylor(asinh(x),x)
```

- But we want to consider an arbitrary function $u(x)$ and its nearby values $u(x \pm h), u(x \pm 2h), \dots$: for example, compute the Taylor series $u(x + h) = u(x) + hu'(x) + \frac{1}{2}h^2u''(x) + \dots + \mathcal{O}(h^6)$ (albeit printed in reverse order) as $D(u)$ denotes a derivative:

```
1 syms u(x) h
2 taylor(u(x+h),h)
```

- Now check that $[u(x - h) - 2u(x) + u(x + h)]/h^2$ estimates the second derivative to error approximately $h^2u'''/12$ by

```
1 taylor((u(x-h)-2*u(x)+u(x+h))/h^2,h)
```

- Similarly, as coded in Algorithm 4.3, we find that $3(u(x + h)^2 - u(x - h)^2)/(2h) = 6uu' + h^2(uu''' + 3u'u'') + \mathcal{O}(h^4)$ after asking for simplified results:

```
1 taylor(3*(-u(x-h)^2+u(x+h)^2)/(2*h),h)
2 simplify(ans)
```

- Lastly, as coded in Algorithm 4.3, we find that $(u(x + 2h) - 2u(x + h) + 2u(x - h) - u(x - 2h))/(2h^3) = u''' + \frac{1}{4}h^2u'''' + \mathcal{O}(h^4)$

```
1 taylor((u(x+2*h)-2*u(x+h)+2*u(x-h)-u(x-2*h))/(2*h^3),h)
```

Putting the last two results together, the time derivative function in Algorithm 4.3 codes the PDE

$$u_t + 6uu_x + u_{xxx} = -h^2(uu_{xxx} + 3u_xu_{xx} + \frac{1}{4}u_{xxxx}) + \mathcal{O}(h^4),$$

This is called the **equivalent pde** of the discretisation. A discretisation is called **consistent** with a given PDE if the equivalent PDE \rightarrow given PDE as the grid spacing $h \rightarrow 0$ for all x in the domain (as it does here).

Rule of thumb: prefer $\mathcal{O}(h^2)$ consistency (or sometimes $\mathcal{O}(h^4)$), avoid $\mathcal{O}(h)$ consistency as being too inaccurate, and avoid $\mathcal{O}(h^6)$ as being too likely to be unstable.

Theorem 4.2 (Lax's equivalence theorem: consistency + stability = convergence). *If a discretisation is consistent, and the solutions of the discretisation are stable as $h \rightarrow 0$, then the solutions of the discretisation approach the true solutions as $h \rightarrow 0$.*

Example 4.7 Consider the previous three examples. ♣

4.1.2 Operator algebra generates discretisations

Challenge: how do we find a consistent discretisation for a given PDE? Answer: some operator algebra helps.

Definition 4.3. Define the following operators:

- differentiation ∂ so that $\partial u(x) := u'(x)$;
- shift \mathcal{E} by one grid spacing h so that $\mathcal{E}u(x) := u(x + h)$;
- centred difference $\delta := \mathcal{E}^{1/2} - \mathcal{E}^{-1/2}$; and
- centred mean $\mu := (\mathcal{E}^{1/2} + \mathcal{E}^{-1/2})/2$.

For example, $\partial^2 u = u''(x)$, $\partial(u^2) = 2uu'$, $\mathcal{E}^2 u(x) = \mathcal{E}u(x + h) = u(x + 2h)$, by induction $\mathcal{E}^p u(x) = u(x + ph)$ so that naturally $\mathcal{E}^{-1} u(x) = u(x - h)$ and $\mathcal{E}^{\pm 1/2} u(x) = u(x \pm h/2)$.

Theorem 4.4. These operators are related by this table:

	\mathcal{E}	δ	$h\partial$
$\mathcal{E} =$	\mathcal{E}	$1 + \frac{1}{2}\delta^2 + \delta\sqrt{1 + \frac{1}{4}\delta^2}$	$e^{h\partial}$
$\delta =$	$\mathcal{E}^{\frac{1}{2}} - \mathcal{E}^{-\frac{1}{2}}$	δ	$2 \sinh \frac{1}{2}h\partial$
$h\partial =$	$\log \mathcal{E}$	$2 \operatorname{arcsinh} \frac{1}{2}\delta$	$h\partial$
$\mu =$	$\frac{1}{2}(\mathcal{E}^{\frac{1}{2}} + \mathcal{E}^{-\frac{1}{2}})$	$\sqrt{1 + \frac{1}{4}\delta^2}$	$\cosh \frac{1}{2}h\partial$

Example 4.8 Discretise the second derivative using

```
1 syms delta
2 taylor(4*asinh(delta/2)^2,delta)
```



Example 4.9 Discretise the first derivative using

```
1 syms delta
2 taylor(2*asinh(delta/2),delta)
3 taylor(2*asinh(delta/2)/sqrt(1+delta^2/4),delta)
```

and the mean operator. ♣

Similarly for other derivatives in a PDE.

Rule of thumb: where possible, prefer to discretise in ‘conservative’ form.

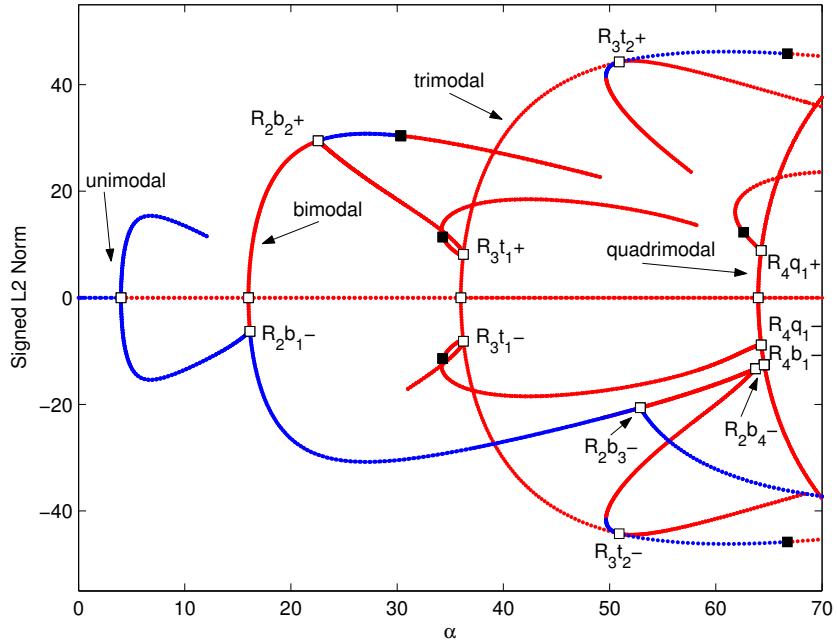


Figure 4.6: the rich structure of equilibria of the nonlinear Kuramoto–Sivashinsky PDE as a function of a parameter α (MacKenzie & Roberts 2006): blue, stable equilibria; red, unstable equilibria.

Exercise 4.3 Recall the operators: differentiation $\partial u(x) := u'(x)$; shift $\mathcal{E}u(x) := u(x + h)$; centred difference $\delta := \mathcal{E}^{1/2} - \mathcal{E}^{-1/2}$, and centred mean $\mu := (\mathcal{E}^{1/2} + \mathcal{E}^{-1/2})/2$.

- Define the new operator $\Delta := \mathcal{E} - 1$, what is $\Delta u(x)$?
- Use the expression of operators μ and δ in terms of \mathcal{E} to deduce \mathcal{E}^{-1} in terms of μ and δ .
- Use the definition of operators μ and δ to deduce a simplify $(\delta^2 u)(\mu \delta u) + 2u(\mu \delta u)$.



4.2 Find equilibria

In applications we often want to find ‘steady states’, ‘fixed points’ or ‘equilibria’. A not so quick and far too dirty approach is simply to computationally integrate for long times and hope an equilibrium emerges. Resist this temptation.

The reason for resisting is that integration can never find the rich ‘skeleton’ of unstable equilibria shown by Figure 4.8 in red. Such a ‘skeleton’ underpins understanding of the range of solutions of the system.

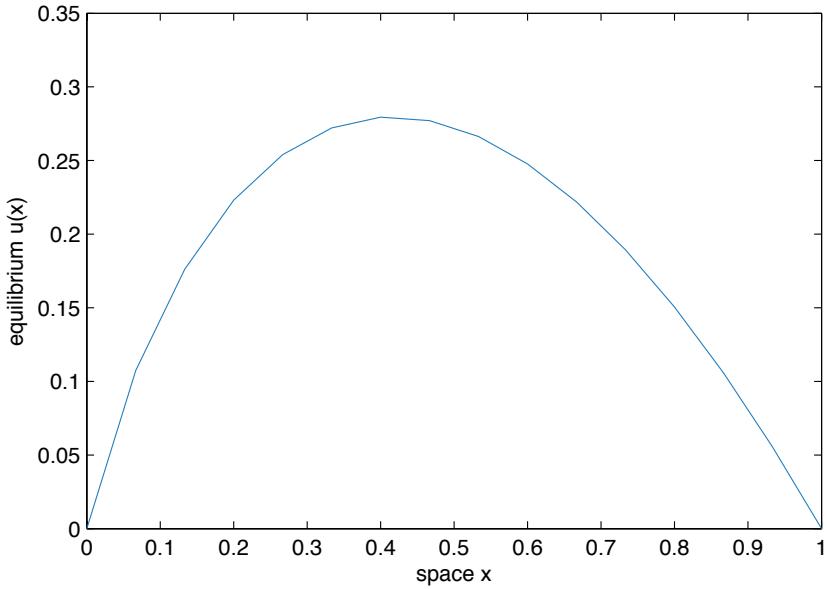


Figure 4.7: Algorithm 4.5 computes this equilibrium of the forced nonlinear PDE $u_t = uu_{xx} + 1 - x$.

Instead encode a wrapper around your time derivative function to find a u such that $\partial u / \partial t = 0$.

The function `u=fsolve(fun,u0)` For any coded function `fun(v)` of a vector $\vec{v} \in \mathbb{R}^n$, returning vector $\vec{f}(\vec{v}) \in \mathbb{R}^n$, this `fsolve()` computes a vector \vec{u} such that $\vec{f}(\vec{u}) = \vec{0}$. For example,

```
>> fun=@(v) [v(1)/4-v(1)*v(2);-v(2)+v(1)^2];
>> u=fsolve(fun,[1;0])
```

Equation solved.

`fsolve` completed because the vector of function values is near zero as measured by the default value of the function tolerance, and the problem appears regular as measured by the gradient.

`<stopping criteria details>`

```
u =
0.5000
0.2500
```

Always check that such a successful message is reported.

Example 4.10 Use the Matlab function `fsolve` to find the equilibrium in Figure 4.7 of the forced nonlinear PDE $u_t = uu_{xx} + 1 - x$ for $0 < x < 1$ and boundary values $u(0, t) = u(1, t) = 0$. Algorithm 4.5 reuses and modifies the code of Algorithm 4.2.



Algorithm 4.5: use `fsolve` to find equilibria of the nonlinear PDE $u_t = uu_{xx} + 1 - x$.

```

1 function equilheatnon
2 % Compute equilibrium of nonlinear PDE
3 % via finite differences. AJR, 15/8/2014
4 global j x
5 nPoints=16
6 x=linspace(0,1,nPoints)';
7 j=2:nPoints-1;
8 u0=ones(size(x));
9
10 fun=@(u) dudt(0,u);
11 u=fsolve(fun,u0(j))
12
13 u=[uleft(0);u;uright(0)];
14 plot(x,u)
15 xlabel('space\lx'), ylabel('equilibrium\lu(x)')
16 print -depsc2 equilheatnon
17 %-----
18 function ut=dudt(t,u)
19 global j x
20 dx=x(2)-x(1);
21 u=[uleft(t);u;uright(t)];
22 ut=u(j).*(u(j-1)-2*u(j)+u(j+1))/dx^2+1-x(j);
23 %-----
24 function ua=uleft(t)
25 ua=zeros(size(t));
26 %-----
27 function ub=uright(t)
28 ub=zeros(size(t));

```

4.3 Numerical linearisation characterises solution dynamics

Recall that we often algebraically linearise equations to be able to find complete but approximate solutions analytically. Analogously, numerical linearisation completely characterises the numerical code in the neighbourhood of an equilibrium.

- Hartman–Grossman theorem asserts that generally the non-linear dynamics near the equilibrium are just a ‘bent’ version of the linear dynamics.
- If the numerical characteristics do not match analytic expectations, then suspect there are errors or discretisation issues to resolve.

Form an implicit eigenproblem Given a PDE for $u(x, t)$, say. Suppose we have coded a discretisation $d\vec{u}/dt = \tilde{f}(\vec{u})$, and found an

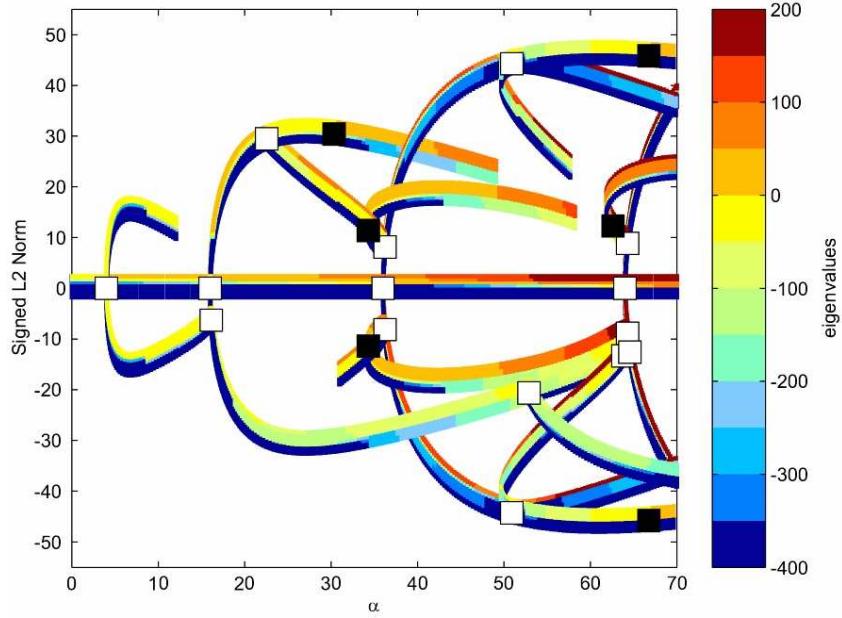


Figure 4.8: the eigenvalues of linearised dynamics about all equilibria of the nonlinear Kuramoto–Sivashinsky PDE as a function of a parameter α (MacKenzie & Roberts 2006): at each point for each branch of equilibria the eight leading eigenvalues are plotted in colour; any red–orange indicate unstable modes whereas yellow–blue are stable.

equilibrium \vec{u}^* , which satisfies $\vec{f}(\vec{u}^*) = \vec{0}$. We seek the dynamics of small perturbations away from equilibrium: seek $\vec{u}(t) = \vec{u}^* + \epsilon \vec{v} e^{\lambda t}$ for small ϵ . The discretisation then reduces to an eigenproblem $\lambda \vec{v} = A \vec{v}$: Instead of finding the matrix A , we invoke the procedure `eigs()` that uses just the product $A \vec{v}$: this product we get direct from the function \vec{f} by evaluating $\vec{f}(\vec{u}^* + \epsilon \vec{v})/\epsilon$ for some small ϵ , say 10^{-6} .

The function `[v,d]=eigs(fun,n,k)` For any coded function `fun(v)` of a vector $\vec{v} \in \mathbb{R}^n$, returning vector $(A\vec{v}) \in \mathbb{R}^n$, this `eigs()` computes k eigenvalues and eigenvectors of implicit matrix A . For example,

```
>> fun=@(v) [1 2;3 4]*v;
>> [v,d]=eigs(fun,2,2)
v =
-0.4160   -0.8246
-0.9094    0.5658
d =
5.3723         0
0    -0.3723
```

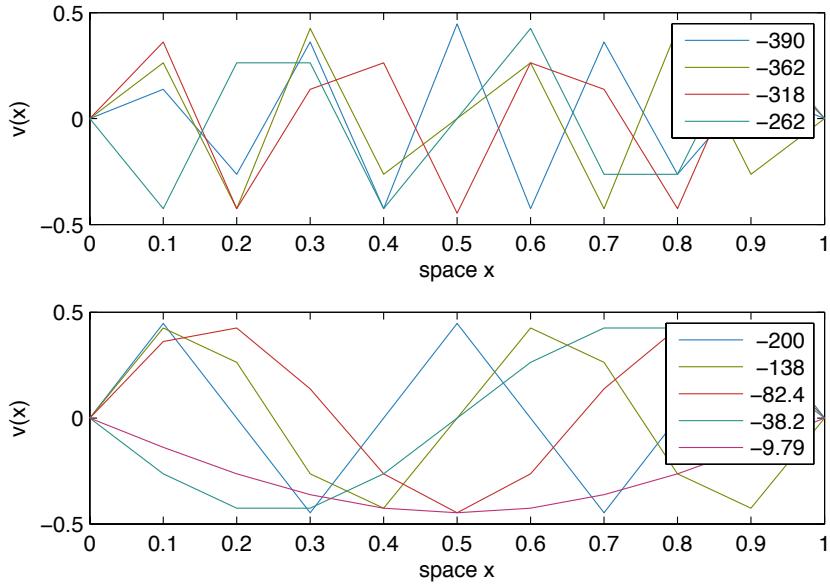


Figure 4.9: Algorithm 4.6 computes these eigenvalues and eigenvectors of the coded discretisation of the heat PDE $u_t = u_{xx}$: the bottom ones are reasonably accurate; the top are not.

We use `eigs()` with $k = n$ to find all eigenvalues/vectors, and usually Matlab complains about this—ignore the complaint.

(However, be wary that the iteration of `eigs()` can return rubbish for problems with non-symmetric matrices having repeated eigenvalues.)

Example 4.11 Compute the eigenvalues and eigenvectors of the coded discretisation of the heat PDE:

$$\frac{du_j}{dt} = u_{xx}(x_j, t) \approx \frac{u_{j-1} - 2u_j + u_{j+1}}{h^2},$$

for $0 < x < 1$ with homogeneous boundary conditions $u(0, t) = u(1, t) = 0$. Here the equilibrium is the zero solution $u(x, t) = 0$. Algorithm 4.6 invokes `eigs()` in Matlab. ♣

General solution The function `eigs()` gives many eigenvalues and eigenvectors. But we do not just get a list of separate possible solutions—we get much more. For *small* solutions near the equilibria, the dynamics are approximately linear, so we obtain a general solution by linear superposition: that is, a general solution near \vec{u}^* is

$$\vec{u}(t) = \vec{u}^* + \epsilon \sum_j c_j \vec{v}_j e^{\lambda_j t}$$

for any coefficients c_j .

- Imaginary parts to λ_j correspond to components oscillatory in time.

Algorithm 4.6: use numerical linearisation to characterise the coded discretisation of the heat PDE. This framework adapts to many similar PDE problems.

```

1 function heateig
2 % Compute spectrum of heat PDE with finite differences
3 % knowing zero is an equilibrium. AJR, 26/8/2015
4 global j x
5 nPoints=11
6 x=linspace(0,1,nPoints)';
7 j=2:nPoints-1;
8 u0=zeros(size(x));
9
10 small=1e-6
11 fun=@(v) dudt(0,u0(j)+small*v)/small;
12 [v,d]=eigs(fun,nPoints-2,nPoints-2)
13
14 d=diag(d)'
15 analytic=-(nPoints-2:-1:1).^2*pi.^2
16 v=[uleft(d);v;uright(d)];
17 subplot(2,1,1), plot(x,v(:,1:4))
18 legend(num2str(d(1:4)',3))
19 xlabel('space\backslash x'), ylabel('v(x)')
20 subplot(2,1,2), plot(x,v(:,end-4:end))
21 legend(num2str(d(end-4:end)',3))
22 xlabel('space\backslash x'), ylabel('v(x)')
23 print -depsc2 heateig
24 %-----
25 function ut=dudt(t,u)
26 global j x
27 dx=x(2)-x(1);
28 u=[uleft(t);u;uright(t)];
29 ut=(u(j-1)-2*u(j)+u(j+1))/dx.^2;
30 %
31 function ua=uleft(t)
32 ua=zeros(size(t));
33 %
34 function ub=uright(t)
35 ub=zeros(size(t));

```

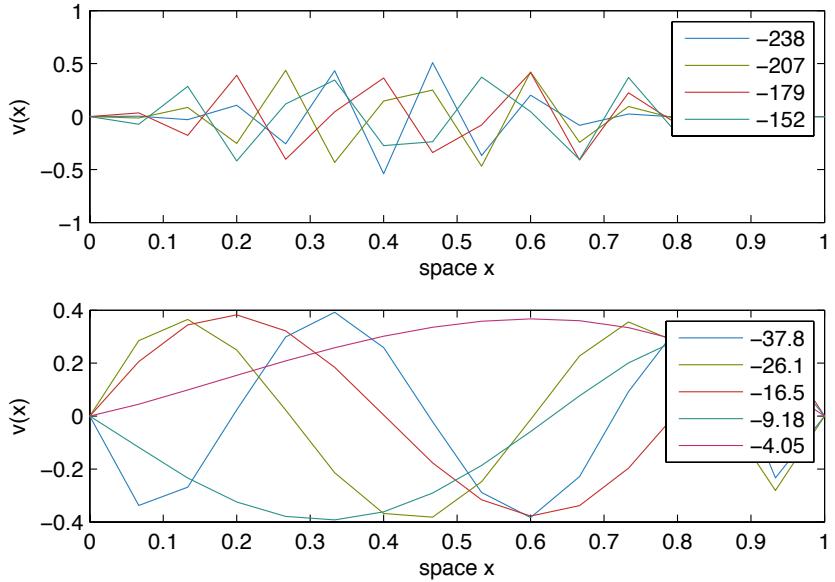


Figure 4.10: Algorithm 4.7 computes these eigenvalues and eigenvectors of the coded discretisation of the nonlinear PDE $u_t = uu_{xx} + 1 - x$ when linearised about its equilibrium. The deformed oscillatory structure matches what is expected from Sturm–Liouville theory.

- If any one or more eigenvalue has real-part > 0 , then the equilibrium \vec{u}^* is unstable.
- If all eigenvalues have real-part < 0 , then the equilibrium \vec{u}^* is stable.

Definition 4.5 (stiffness). *When there is a large ratio between smallest magnitude and largest magnitude eigenvalues, say ratio $\gg 10$, as shown in Figure 4.9, then a system of ODEs is termed **stiff**.*

For a stiff system, one should use an integrator specially designed for stiffness, like `ode15s`. Almost always, discretisations of PDEs are stiff.

Example 4.12 Consider the discretisation of the nonlinear PDE $u_t = uu_{xx} + 1 - x$ for $0 < x < 1$ with homogeneous boundary conditions $u(0, t) = u(1, t) = 0$. Compute the eigenvalues and eigenvectors of the dynamics linearised about the computed equilibrium. Algorithm 4.7 invokes `eigs()` in Matlab.



Example 4.13 Consider the discretisation of the Korteweg–de Vries PDE $u_t + 6uu_{xx} + u_{xxx} = 0$ for $0 < x < 10$ and 10-periodic. Compute the eigenvalues and eigenvectors of the dynamics linearised about $u^* = 1$. Algorithm 4.8 invokes `eigs()` in Matlab.



Algorithm 4.7: use numerical linearisation to characterise the coded discretisation of the nonlinear PDE $u_t = uu_{xx} + 1 - x$.

```

1 function heatnoneig
2 % Characterise discretisation of nonlinear PDE
3 % by numerical linearisation. AJR, 15/8/2014
4 global j x
5 nPoints=16
6 x=linspace(0,1,nPoints)'
7 j=2:nPoints-1
8
9 u0=ones(size(x))
10 fun=@(u) dudt(0,u);
11 u0(j)=fsolve(fun,u0(j))
12
13 small=1e-6
14 fun=@(v) dudt(0,u0(j)+small*v)/small;
15 [v,d]=eigs(fun,nPoints-2,nPoints-2)
16
17 d=diag(d)'
18 v=[uleft(d);v;uright(d)];
19 subplot(2,1,1), plot(x,v(:,1:4))
20 legend(num2str(d(1:4)',3))
21 xlabel('space\|x'), ylabel('v(x)')
22 subplot(2,1,2), plot(x,v(:,end-4:end))
23 legend(num2str(d(end-4:end)',3))
24 xlabel('space\|x'), ylabel('v(x)')
25 print -depsc2 heatnoneig
26 %-----
27 function ut=dudt(t,u)
28 global j x
29 dx=x(2)-x(1);
30 u=[uleft(t);u;uright(t)];
31 ut=u(j).*(u(j-1)-2*u(j)+u(j+1))/dx^2+1-x(j);
32 %-----
33 function ua=uleft(t)
34 ua=zeros(size(t));
35 %-----
36 function ub=uright(t)
37 ub=zeros(size(t));

```

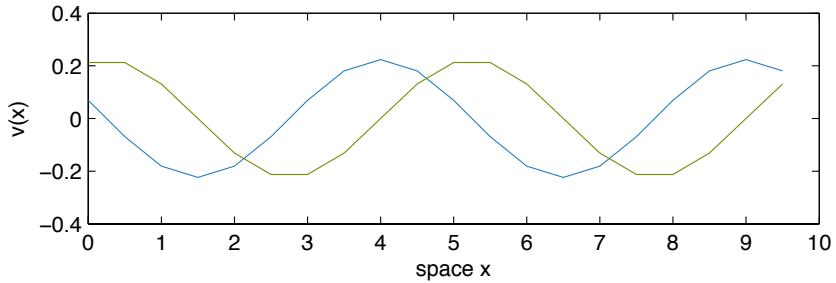


Figure 4.11: an example eigenvector computed by Algorithm 4.8 for the Korteweg–de Vries PDE: linearised about equilibrium $u^* = 1$; these real and imaginary parts show the beautiful shape of a e^{ikx} for wavenumber $k = 2\pi/5 = 1.26$.

Important This sort of check on the eigenvalues of a wave system is vital: it is very easy for a discretisation of a wave system to generate unphysical growth, from small real-parts in the eigenvalues, that destroys the usefulness of a simulation.

Exercise 4.4 Spatial discretisations of wave PDEs also have a dispersion relation. Such a relation informs us of the properties of the discretisation. Sometimes we can find the relation analytically. Recall that a linear (nondimensional) Schrodinger equation is $-i\psi_t = \psi_{xx}$ with dispersion relation $\omega = k^2$. Find the dispersion relation of its spatial discretisation $-i\dot{\psi}_j = (\psi_{j-1} - 2\psi_j + \psi_{j+1})/h^2$ on a grid $x_j = jh$.

- Start by seeking solutions with wavenumber k and frequency ω , namely $\psi_j(t) \propto \exp[i(kx_j - \omega t)]$: substitute into the given Schrodinger PDE, use $x_{j\pm 1} = x_j \pm h$, and deduce the frequency in terms of complex exponentials.
- Recalling a relation between complex exponentials and trigonometric functions, and also some trigonometric identities, what is the dispersion relation?
- What is the (numerical) ratio of the frequencies of the PDE and the discretisation, $\omega_{\text{pde}}/\omega_{\text{discrete}}$, for a wave of wavenumber $k = \pi/h$?



Incidentally computational procedures such as `fsove()` and `eigs()` need not just be applied to computed functions. They could be applied to real experiments to find equilibria and linearisation of actual experiments!

Algorithm 4.8: use numerical linearisation to characterise waves in the coded discretisation of the Korteweg–de Vries PDE $u_t + 6uu_x + u_{xxx} = 0$.

```

1 function kdveig
2 % Characterise discretisation of Korteweg-de
3 % Vries PDE with periodic boundary conditions by
4 % numerical linearisation. AJR, 15/8/2014
5 global x
6 nPoints=20
7 x=linspace(0,10,nPoints+1); x=x(1:nPoints)';
8 u0=ones(size(x));
9
10 small=1e-6
11 fun=@(v) dudt(0,u0+small*v)/small;
12 [v,d]=eigs(fun,nPoints,nPoints)
13
14 d=diag(d)'
15 subplot(2,1,1)
16 plot(x,[real(v(:,9)) imag(v(:,9))])
17 xlabel('space\_\_x'),ylabel('v(x)')
18 print -depsc2 kdveig
19 %
20 function ut=dudt(t,u)
21 global x
22 dx=x(2)-x(1);
23 u=[u(end-1:end);u;u(1:2)];
24 j=(1:length(x))+2;
25 ut=-3*(-u(j-1).^2+u(j+1).^2)/(2*dx) ...
26 -(-u(j-2)+2*u(j-1)-2*u(j+1)+u(j+2))/(2*dx.^3) ;

```

- One provides such procedures (`fsolve/eigs`) with the name of a computational function.
- The computational function is a ‘wrapper’ for an experiment in the following sense.
 - the function sets an experiment going with the prescribed parameters;
 - the experiment executes and returns values to the function;
 - the function packs the output values together to return to the procedure (`fsolve/eigs`).
- The procedure continually interacts with the experiment via the function wrapper, systematically exploring the parameter space, in order to answer your request.

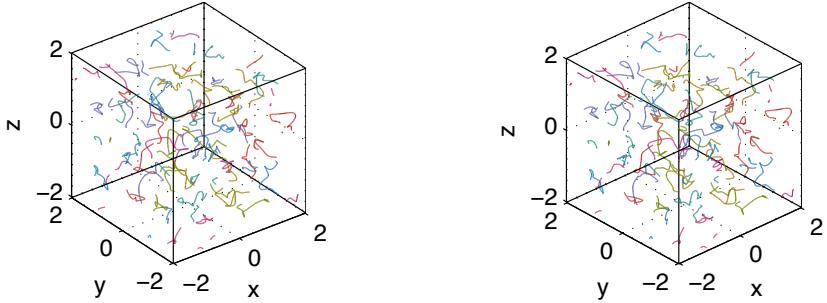


Figure 4.12: trajectories of 64 atoms, over a time $0 \leq t \leq 3$, in a triply-periodic, cubic, spatial domain, showing the beginnings of the complicated inter-atomic interactions. This stereo pair when viewed cross-eyed gives a 3D effect. The code `demoAtomSim.m` shows stereo real-time simulation.

4.4 PDE-free patch dynamics

Recall that many engineers and scientists run microscale simulations of large systems. But typically the microscale is so detailed that there is no foreseeable hope of simulating real life macroscale situations. So-called Projective Integration is a method to skip over much of the time simulation, such as the atomistic simulations of Figure 4.12. This section introduces the complementary method of omitting most of the spatial simulation to empower macroscale simulation—the so-called *patch dynamics*.

(Kevrekidis & Samaey 2009, §5)

The scenario We want to simulate the space-time dynamics of some quantity $u(x, t)$, in 1D for simplicity. We are given a ‘black box’ simulator \mathcal{U} that computes for any (small) region of space, $a \leq x \leq b$, the solution

$$u(x, t) = \mathcal{U}(x, t; a, u_a(t), b, u_b(t), t_0, u_0(x)),$$

where to be specified are

- the left boundary condition is $u(a, t) = u_a(t)$,
- the right boundary condition is $u(b, t) = u_b(t)$, and
- the initial condition $u(x, t_0) = u_0(x)$.

But suppose that this computer simulation is only practicable for *small* domains in space.

The macroscale is too expensive The desired simulation of the system is over some macroscale domain, $A \leq x \leq B$. $\mathcal{U}(x, t; A, 0, B, 0, 0, u_0(x))$ is not available in realistic time.

Couple small patches Create on the macroscale domain a macroscale grid, $A = X_1 < X_2 < X_3 < \dots < X_{N-1} < X_N = B$, usually equi-spaced with $\Delta X_j = H$. The main macroscale quantities of

interest are the field evaluated at the grid points, $U_j(t) = u(X_j, t)$. Create small patches of space, of size $2h$, centred around each interior grid point: patch/element $\{x \mid |x - X_j| < h\} = \{x \mid a_j = X_j - h \leq x \leq X_j + h = b_j\}$. We simulate only in these small patches.

Such patchy simulation is a great saving in computation. For example, if the ratio $2h/H = 0.1$ then we compute solutions ten times quicker. Achieve higher efficiencies for smaller h/H or in higher dimensions.

But the simulator for each patch needs boundary conditions. Recent research (Roberts & Kevrekidis 2007, e.g.) shows that a good scheme is to provide these boundary conditions by classic standard Lagrange interpolation from the surrounding *macroscale* values $U_j(t)$. Such patch boundary conditions couple the patches together over the macroscale domain.

That is the basic scheme. Does it work?

Example 4.14 (Simplest). Suppose—although we do not know it—that the microsimulator \mathcal{U} is simply the diffusion equation $u_t = u_{xx}$ (it might arise as the results of thousands of random ‘walkers’). For simplicity we put one patch in the middle of a domain of size $2H$, $-H \leq x \leq H$; the patch being $|x| \leq h$. Assume macroscale boundary conditions $u(\pm H, t) = 0$, that is $U_1 = U_3 = 0$. Then the classic macroscale interpolant is the parabola²

$$u = U_2(t)(1 - x^2/H^2).$$

The microscale boundary values given to the simulator are then that, introducing $r = h/H$,

$$u(\pm h, t) = U_2(t)(1 - h^2/H^2) = U_2(t)(1 - r^2).$$

Because the middle of the patch is the macroscale grid, the grid value $U_2(t) = u_2(0, t)$, the above microscale boundary values give the funny non-local boundary conditions

$$u_2(\pm h, t) = (1 - r^2)u_2(0, t).$$

What is the general solution of the heat PDE $u_t = u_{xx}$ with such patch boundary conditions? Does the solution behave properly? Do solutions even exist?

Use separation of variables to predict behaviour. ♣

Conclude: we can not only discretise PDEs, but also processes. With more patches in the domain, we analogously obtain more physically reasonable macroscale modes, and more accurate dynamics of these

² In applications one would often use patches adjacent to the physical boundaries in order to resolve so-called boundary layers. Then $U_1(t)$ would be the mid-value of the left boundary patch.

modes. Such patch dynamics can empower large scale simulation of microscopically detailed systems. Research is needed.

Exercise 4.5 Repeat the above example but for the case where the ‘black box’ microsimulator of diffusion uses specified boundary values of the gradient u_x instead of specified boundary values of the field u . Keep the boundary conditions on the physical domain as $u(\pm H, t) = 0$. Remember to differentiate the interpolant to find its gradient on the boundaries of the patch.

- (a) Argue that interpolation through $u(0, t)$ provides the patch simulator with what non-local boundary conditions, for ratio $r = h/H$?
- (b) You are given that separation of variables, $u = X(x)T(t)$, finds that the dynamics of the patch diffusion satisfy the ODEs

$$\dot{T} + K\lambda T = 0, \quad X'' + \lambda X = 0,$$

together with non-local boundary conditions on $X(x)$ from the first part. Assume eigenvalues $\lambda > 0$. In terms of k such that $\lambda = k^2$, find the algebraic equations determining the eigenvalues of the spatial ODE for X .

- (c) From the smallest solution, $k > 0$, of these equations approximate the smallest eigenvalue for the case of a small patch size h .



4.5 May use Octave or Scilab

Octave and Scilab are very similar to Matlab but are free to download and use. Indeed all three are built upon the superb LAPACK and BLAS3 routines that implement linear algebra operations highly efficiently on the complicated architectures of modern computers.

- Octave: <http://www.gnu.org/software/octave/>; or with a web browser via <http://octave-online.net>; or via some mobile device ‘apps’ such as *Sage* or *iAnoc Pro*; its core is a ‘clone’ of Matlab, but various functions have different names.
- Scilab: <http://www.scilab.org>; many things are the same as Matlab, but with differences to ‘trip over’ (such as `end` is `$`).

As examples of their use, Algorithms 4.9 and 4.10 list code corresponding to the basic template of Algorithm 4.1.

Algorithm 4.9: method of lines to solve the heat PDE using Octave.

```

1 function heatOct
2 % Compute solutions of heat PDE with finite differences
3 % Octave version. AJR, 22/8/2014
4 global j x
5 nPoints=11
6 x=linspace(0,1,nPoints)'
7 j=2:nPoints-1
8 u0=6*x.* (1-x)
9
10 t=linspace(0,0.2,15)'
11 u=lsode(@dudt,u0(j),t)
12
13 u=[uleft(t) u uright(t)]
14 surf(x,t,u)
15 xlabel('space\backslash x'), ylabel('time\backslash t'), zlabel('u(x,t)')
16 %-----
17 function ut=dudt(u,t)
18 global j x
19 dx=x(2)-x(1);
20 u=[uleft(t);u;uright(t)];
21 ut=(u(j-1)-2*u(j)+u(j+1))/dx^2;
22 %-----
23 function ua=uleft(t)
24 ua=zeros(size(t));
25 %-----
26 function ub=uright(t)
27 ub=sin(8*t);

```

Algorithm 4.10: method of lines to solve the heat PDE using Scilab.
Execute with `exec("heatSci.sce")`

```

1 // Compute solutions of heat PDE with finite differences
2 // Scilab version: exec("heatSci.sce"); AJR, 25/8/2014
3 function ut=dudt(t,u)
4 global j x
5 dx=x(2)-x(1);
6 u=[uleft(t);u;uright(t)];
7 ut=(u(j-1)-2*u(j)+u(j+1))/dx^2;
8 endfunction
9 //-----
10 function ua=uleft(t)
11 ua=zeros(t);
12 endfunction
13 //-----
14 function ub=uright(t)
15 ub=sin(8*t);
16 endfunction
17 //-----
18 global j x
19 nPoints=11
20 x=linspace(0,1,nPoints)'
21 j=2:nPoints-1
22 u0=6*x.*((1-x)
23
24 t=linspace(0,0.4,20)
25 u=ode(u0(j),t(1),t,dudt)
26
27 u=[uleft(t);u;uright(t)]
28 clf(); plot3d1(x',t,u)
29 xlabel('space\_{\}x'), ylabel('time\_{\}t'), zlabel('u(x,t)')
30 set(gcf(),'color_map',hotcolormap(32))

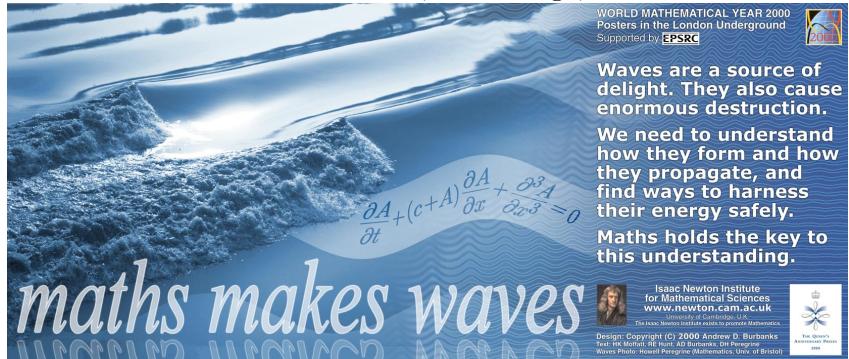
```

5 Model shallow water waves

Chapter Contents

5.1 Conservation derives the PDEs	55
5.2 Small amplitude waves	56
5.3 Compute seiches in 1D	57

From the Issac Newton Institute, Cambridge,



5.1 Conservation derives the PDEs

As in Figure 5.1, consider the flow of water with surface located at $z = \eta(x, t)$ above a varying bottom at $z = -d(x)$. Take $z = 0$ to be the water surface at rest, so that at rest the water depth is $d(x)$. Assume the vertical movement of water is negligible and consider only the horizontal velocity $u(x, t)$. We aim to derive the (approximate) nonlinear shallow water PDEs

$$\frac{\partial \eta}{\partial t} + \frac{\partial[(\eta + d)u]}{\partial x} = 0, \quad \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = -g \frac{\partial \eta}{\partial x}. \quad (5.1)$$

These represent conservation of

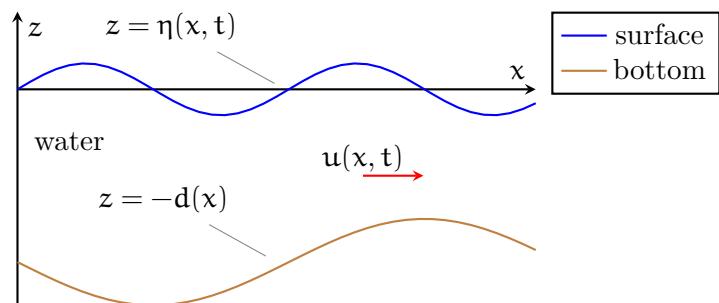


Figure 5.1: water waves in shallow water of varying depth.

Water from the continuity PDE (1.1),

Momentum similar to the momentum PDE (1.5).

A complicating difference is that the pressure forces vary with depth, they are no longer constant in the cross-section.

But that is not all: pressures act at right angles to surfaces, so a sloping bed applies a pressure to the fluid that is equal and opposite to the fluid pressure and has a component in the horizontal.

Vary the earlier conservation argument:

Exercise 5.1 Recall the conservation of mass and momentum arguments for the dynamics of shallow water. We assumed the width of the water channel was fixed at W . Now suppose that the width varies with x ; that is, the width is $W(x)$.

- (a) With varying width $W(x)$ the continuity PDE would then be either the same or modified to what?
- (b) Recall that with constant width W the total force across a cross-section from pressure changes is $W(\eta + d)\rho g\eta$. With varying width $W(x)$ this total force would remain the same or change to what?
- (c) Recall that with constant width W the change in pressure acts on variations in depth and water height to generate a body force over an interval $[a, b]$ of $\int_a^b W\rho g(\eta + d)_x \eta \, dx$. With varying width $W(x)$ an additional contribution to this pressure induced body force is what?



5.2 Small amplitude waves

Dispersion relation Suppose the bottom is flat, d constant. Further seek solutions with *small* variations from the equilibrium $u = \eta = 0$, that is, linearise in small u and η . Then the governing PDES (5.1) lead to dispersion relation $\omega^2 = gdk^2$ of the classic wave PDE with wave speed \sqrt{gd} .

Sturm–Liouville form in varying depth Consider varying depth $d(x)$, with surface η and velocity u small. Leads to the Sturm–Liouville ODE

$$(gdX')' + \lambda X = 0, \quad \text{with } \ddot{T} + \lambda T = 0,$$

which is well understood provided BCs are appropriate. Consequently, we predict oscillations with frequency $\sqrt{\lambda}$ for the set of eigenvalues.

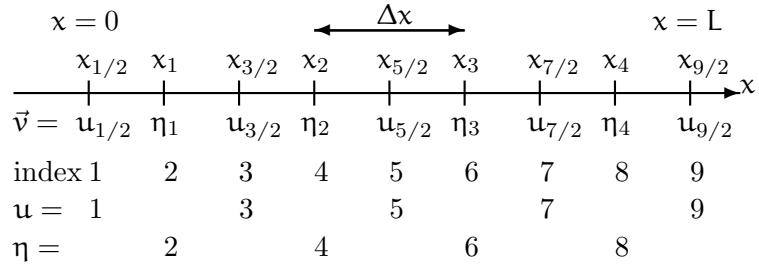


Figure 5.2: staggered grid for 1D wave computations of Section 5.3. From top to bottom: first, the domain is say from $x = 0$ to $x = L$ with a grid of spacing $\Delta x/2$; we compute $u_{j/2}(t)$ for odd j and $\eta_{j/2}(t)$ for even j , $j = 1, \dots, N$.

Exercise 5.2 (seiches in a swimming pool). Consider a swimming pool (or bath) of constant depth d and of length L , say over the domain $0 \leq x \leq L$. Appropriate boundary conditions are that $u = 0$ at $x = 0, L$ to reflect the water cannot go through the walls of the pool. Deduce corresponding boundary conditions for η_x from the PDEs (5.1). Find a general solution for small amplitude waves in the pool (in terms of trigonometric functions).

- (a) From $u = 0$ at $x = 0, L$, determine the boundary conditions for the water surface at $x = 0, L$.
- (b) Separation of variables applied to the shallow water wave PDE $\eta_{tt} = g d \eta_{xx}$ then leads to what spatial structures?
- (c) Hence for arbitrary constants a_n and b_n a general solution for small amplitude waves in the pool is what?



Example 5.1 (seiches in a parabolic lake). What are the small amplitude oscillations of a lake of depth $d(x) = D(1 - x^2/L^2)$ for $|x| < L$?



5.3 Compute seiches in 1D

Rule Use a staggered grid such as Figure 5.2 to spatially discretise wave PDEs which are ‘dominantly’ in the form

$$\frac{\partial \eta}{\partial t} = ? \frac{\partial u}{\partial x} + \dots, \quad \frac{\partial u}{\partial t} = ? \frac{\partial \eta}{\partial x} + \dots.$$

Called staggered because we interleave in space the two complementary unknowns u and η .

Figure 5.2 illustrates a staggered grid on $0 \leq x \leq L$. This grid is appropriate for boundary conditions that $u(0, t)$ and $u(L, t)$

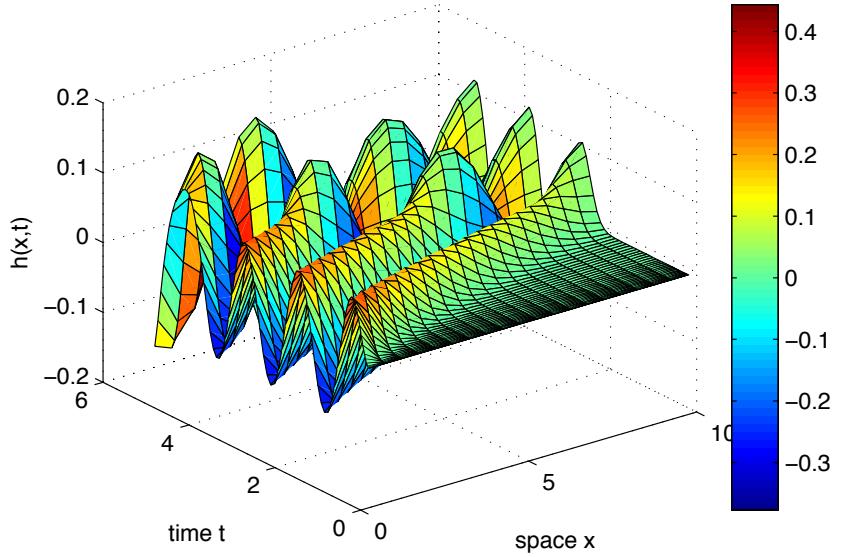


Figure 5.3: Algorithm 5.1 computes such solutions of the linearised shallow water PDEs in water depth $d(x) = 1 + 3x/10$ in a pool of length 10 m. Surface colour is velocity $u(x, t)$.

are prescribed. For example, Figure 5.2 shows we resolve, for $j = 1, \dots, N$, the water surface $\eta(x, t)$ at even indexed points, and the water velocity $u(x, t)$ at the odd indexed points (or vice versa). Then the vector of unknowns to integrate over time is the interleaved vector

$$\vec{v}(t) = (u_{1/2}, \eta_1, u_{3/2}, \eta_2, u_{5/2}, \dots, \eta_{(N-1)/2}, u_{N/2})$$

with BCS determining the left/rightmost values. Then approximate the PDEs by (schematically):

- integer j , $\dot{\eta}_j \approx ?(u_{j+1/2} - u_{j-1/2})/\Delta x + \dots$;
- half-integer j , $\dot{u}_j \approx ?(\eta_{j+1/2} - \eta_{j-1/2})/\Delta x + \dots$.

Figure 5.3 shows one simulation of the linearised shallow water PDEs where a wave-maker at $x = 0$ creates waves by pumping the left boundary with velocity $u(0, t) = 0.3 \sin 5t$. The surface height is the water surface, while the surface colour is determined by the water velocity at that position and time. Algorithm 5.1 computes this figure.

The next challenge is to modify Algorithm 5.1 to simulate the nonlinear dynamics by incorporating the effects of the terms uu_x and $(hu)_x$. The difficulty in detail is that whereas derivatives $\partial/\partial x$ in the linearised PDEs only need the immediate neighbouring grid points, the derivatives involved in these nonlinear terms need information from the next-to-nearest grid points.

Algorithm 5.1: method of lines computes linear shallow water PDEs.
Herein h denotes the water surface η .

```

1 function shallowlin
2 % Compute solutions of linear shallow water PDEs with
3 % finite differences AJR, 16/9/2016
4 % Use staggered grid v=(u1,h2,u3,h4,...,uN)
5 global x g d u h
6 g=10
7 nPoints=41 % should be odd for the BCs
8 x=linspace(0,10,nPoints)'
9 d=1+3*x/x(end);% some varying depth
10 u=3:2:nPoints-1% index of u points
11 h=2:2:nPoints-1% index of h points
12 v0=zeros(nPoints,1);
13
14 [t,v]=ode15s(@dvdt,[0 5],v0(2:end-1));
15
16 v=[uleft(t) v uright(t)];
17 i=1:2:length(t);% only plot selected times
18 surf(x(h),t(i),v(i,h),v(i,h+1))
19 xlabel('space\llcorner x'), ylabel('time\llcorner t'), zlabel('h(x,t)')
20 colorbar
21 print -depsc2 shallowlin
22 %-----
23 function vt=dvdt(t,v)
24 global x g d u h
25 dx=x(3)-x(1);
26 v=[uleft(t);v;uright(t)];
27 vt=nan(size(x));
28 vt(h)=-(d(h+1).*v(h+1)-d(h-1).*v(h-1))/dx;
29 vt(u)=-g*(v(u+1)-v(u-1))/dx;
30 vt=vt(2:end-1);
31 %-----
32 function u=uleft(t)
33 u=0.3*sin(5*t);
34 function u=uright(t)
35 u=zeros(size(t));

```

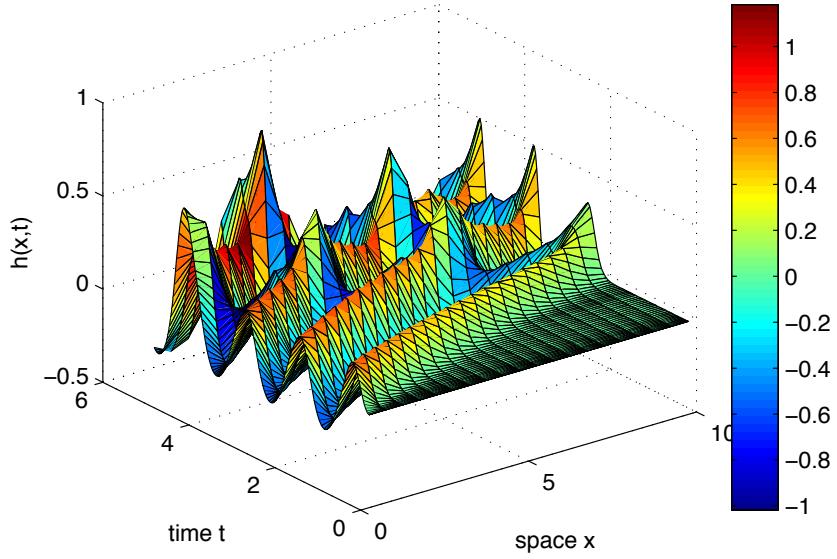


Figure 5.4: Algorithm 5.2 computes such solutions of the nonlinear shallow water PDEs (5.1) in water depth $d(x) = 1 + 3x/10$ in a pool of length 10 m. Surface colour is velocity $u(x, t)$. Observe nonlinear steepening and interactions.

Figure 5.4 shows one simulation of the nonlinear shallow water PDEs where a wave-maker at $x = 0$ creates waves by pumping the left boundary with larger velocity $u(0, t) = 0.7 \sin 5t$. The surface height is the water surface, while the surface colour is determined by the water velocity at that position and time. Algorithm 5.2 computes this figure.

This code and the considerations mentioned provide a flexible framework for effectively simulating a wide range of wave PDE systems in the form

$$\frac{\partial \eta}{\partial t} = ? \frac{\partial u}{\partial x} + \dots, \quad \frac{\partial u}{\partial t} = ? \frac{\partial \eta}{\partial x} + \dots.$$

Algorithm 5.2: compute nonlinear shallow water PDEs by the following changes to Algorithm 5.1. Herein h denotes the water surface η . The Unix command `diff`, of the first line, reports such change information.

```

1 1,2c1,2
2 < function shallowlin
3 < % Compute solutions of linear shallow water PDEs with
4 ---
5 > function shallownon
6 > % Compute solutions of nonlinear shallow water PDEs with
7 21c21
8 < print -depsc2 shallowlin
9 ---
10 > print -depsc2 shallownon
11 26a27,30
12 > hxu=nan(size(x));
13 > hxu(1:2:end)=[v(1)*v(2)
14 > v(u).*(v(u+1)+v(u-1))/2
15 > v(end)*v(end-1)];
16 28,29c32,35
17 < vt(h)=-(d(h+1).*v(h+1)-d(h-1).*v(h-1))/dx;
18 < vt(u)=-g*(v(u+1)-v(u-1))/dx;
19 ---
20 > vt(h)=-(d(h+1).*v(h+1)-d(h-1).*v(h-1))/dx ...
21 > -(hxu(h+1)-hxu(h-1))/dx;
22 > vt(u)=-g*(v(u+1)-v(u-1))/dx ...
23 > -v(u).*(v(u+2)-v(u-2))/(2*dx);
24 33c39
25 < u=0.3*sin(5*t);
26 ---
27 > u=0.7*sin(5*t);

```

6 PDEs with at least three independent variables

Chapter Contents

6.1	Vibration of a rectangular membrane	63
6.1.1	Separate time	63
6.1.2	Boundary is a rectangular membrane	63
6.2	The self-adjoint Sturm–Liouville nature of Helmholtz-like PDEs	66

So far have dealt with one time variable and one space variable. We now introduce similar patterns for three or more space-time variables.
(Haberman 1987, §6.1–4)

Lakes, pools and oceans have two lateral dimensions in space. Instead of the 1D nonlinear shallow water PDEs (5.1), we should analyse and solve the 2D nonlinear shallow water PDEs

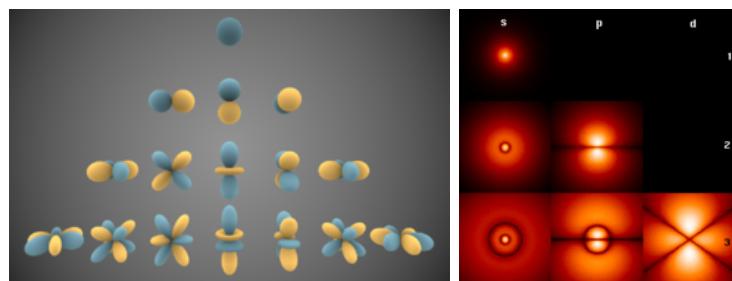
$$\begin{aligned}\frac{\partial \eta}{\partial t} + \frac{\partial[(\eta + d)u]}{\partial x} + \frac{\partial[(\eta + d)v]}{\partial y} &= 0, \\ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= -g \frac{\partial \eta}{\partial x}, \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} &= -g \frac{\partial \eta}{\partial y},\end{aligned}$$

for wave height $\eta(x, y, t)$ and mean velocity $\vec{u} = u(x, y, t)\vec{i} + v(x, y, t)\vec{j}$ over a bottom of (rest) depth $d(x, y)$.

As a more straightforward working example, start with the wave PDE (Kreyszig 2011, §12.8)

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad \text{in domain } D. \quad (6.1)$$

The subsequent theoretical structures of section 6.2 are much more general than this basic wave PDE. For example, the theory would illuminate the spherical harmonics illustrated below-left, or the electron orbitals shown below-right (both from Wikipedia).



6.1 Vibration of a rectangular membrane

(Haberman 1987, §6.2.1) For example, let $u(x, y, t)$ be the up/down displacement of a membrane stretched over a domain D with the membrane's edge specified on the boundary of D , namely $u = 0$. The displacement satisfies the PDE (6.1).

Because of linearity we seek to find as many particular solutions as possible and, if complete, a general solution is an infinite linear combination of the particular solutions.

6.1.1 Separate time

Try separation of variables, here first $u = v(x, y)T(t)$. Substitute and rearrange to

$$\frac{\ddot{T}}{c^2 T} = \frac{1}{v} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) = -\lambda$$

for some constant λ . The time evolution is well known constant coefficient ODE, $\ddot{T} + \lambda c^2 T = 0$, with solution $T = A \cos \omega t + B \sin \omega t$ for frequency $\omega = c\sqrt{\lambda}$. The boundary conditions also separates. Thus the spatial structure becomes the 2D eigenproblem

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \lambda v = 0 \quad \text{in } D, \quad v = 0 \text{ on } \partial D, \quad (6.2)$$

sometimes called the **Helmholtz equation**. The eigenvalues λ of this problem determine the natural frequencies of the membrane.

6.1.2 Boundary is a rectangular membrane

(Kreyszig 2011, §12.9) Suppose the membrane is held on a rectangular boundary. Align (Haberman 1987, §6.3) (x, y) coordinate system so the membrane is $0 \leq x \leq a$ and $0 \leq y \leq b$. The boundary conditions are then

$$u(0, y, t) = u(a, y, t) = u(x, 0, t) = u(x, b, t) = 0.$$

We need to solve the Helmholtz eigenproblem (6.2).

Use separation of variables again. Seek solutions $v(x, y) = X(x)Y(y)$. First, the boundary conditions on the membrane separate to $X(0) = X(a) = 0$ and $Y(0) = Y(b) = 0$. Second, the PDE (6.2) becomes

$$Y \frac{d^2 X}{dx^2} + X \frac{d^2 Y}{dy^2} + \lambda X Y = 0,$$

which separates to, say,

$$\frac{1}{X} \frac{d^2 X}{dx^2} = -\lambda - \frac{1}{Y} \frac{d^2 Y}{dy^2} = -\mu,,$$

for some new constant μ that could be anything. There are two separation constants.

Spatial structure Consider the problem for X :

$$\frac{d^2X}{dx^2} + \mu X = 0, \quad X(0) = X(a) = 0.$$

This is a Sturm–Liouville problem with all its nice properties. In addition we already know its eigen-solutions are

$$\mu_n = n^2\pi^2/a^2, \quad X_n = \sin(n\pi x/a).$$

Similarly for the problem for Y :

$$\frac{d^2Y}{dy^2} + (\lambda - \mu)Y = 0, \quad Y(0) = Y(b) = 0.$$

This is a Sturm–Liouville problem with all its nice properties. In addition we already know its eigen-solutions are

$$\lambda_{nm} - \mu_n = m^2\pi^2/b^2, \quad Y_m = \sin(m\pi y/b).$$

Putting these together, we find solutions for all $m, n = 1, 2, 3, \dots$, namely the eigenfunctions and eigenvalues

$$v_{nm} = \sin(n\pi x/a) \sin(m\pi y/b) \quad \text{for } \lambda_{nm} = \frac{n^2\pi^2}{a^2} + \frac{m^2\pi^2}{b^2}.$$

Figure 6.1 plots contours of some of these eigenfunctions for an example rectangle.

Question: are there other eigenfunctions which are not in product form? Answer: not here, but it is by no means obvious.

Combine with time evolution Assuming these eigenfunctions are complete, the general solution to the original wave equation is then the general linear combination of the spatial structures times their time evolution

$$u = \sum_{n,m=1}^{\infty} \sin \frac{n\pi x}{a} \sin \frac{m\pi y}{b} \\ \times [A_{nm} \cos c\sqrt{\lambda_{nm}}t + B_{nm} \sin c\sqrt{\lambda_{nm}}t]$$

This write the vibrations of the membrane in terms of physical patterns that oscillate at a spectrum of frequencies $c\sqrt{\lambda_{nm}}$. The frequencies depend upon the size and aspect ratio of the membrane.

Determine coefficients with the initial conditions What if the initial conditions are of prescribed deflection, $u(x, y, 0) = u_0(x, y)$, and zero initial speed, $u_t(x, y, 0) = 0$. Then the component in $\sin c\sqrt{\lambda_{nm}}t$ must vanish so all $B_{nm} = 0$. Leaving

$$u_0(x, y) = \sum_{n,m=1}^{\infty} A_{nm} \sin \frac{n\pi x}{a} \sin \frac{m\pi y}{b}$$

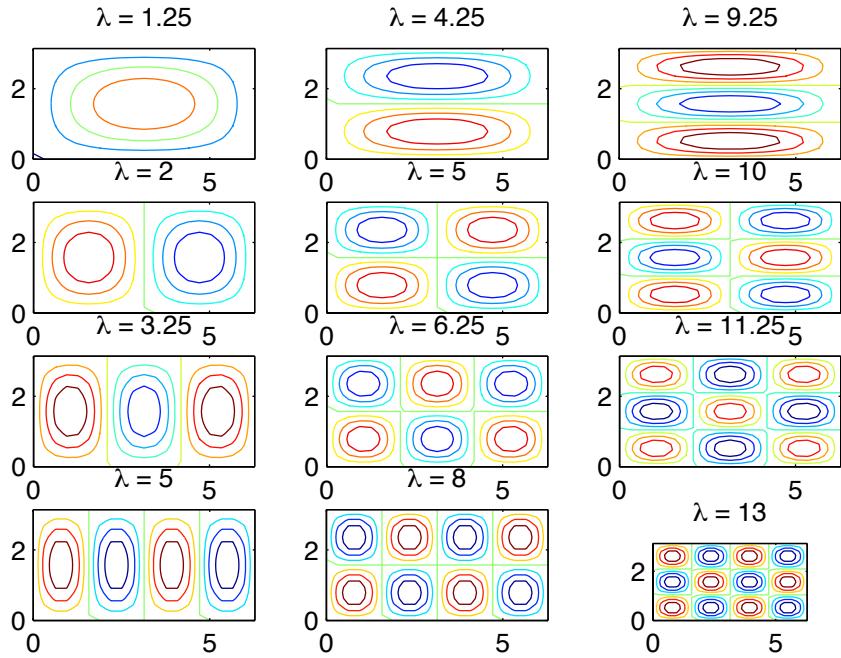


Figure 6.1: contour plots of some eigenfunctions of the wave PDE on a rectangle of size $2\pi \times \pi$. Each subplot is labelled by its corresponding eigenvalue.

to determine the coefficients A_{nm} , for $m, n = 1, 2, 3, \dots$

$$A_{nm} = \frac{4}{ab} \int_0^b \int_0^a u_0(x, y) v_{nm}(x, y) dx dy .$$

Just as in 1D Sturm–Liouville/self-adjoint problems, the spatial eigenfunction itself determines coefficients in the expansion of initial conditions.

Exercise 6.1 Consider the heat conduction PDE on a square

$$\frac{\partial u}{\partial t} = \nabla^2 u, \quad 0 \leq x, y \leq \pi$$

with boundary conditions that $u = 0$ on the left, top and right sides of the square, but that $\partial u / \partial y = 0$ on the bottom side $y = 0$. Use separation of variables to find all eigenvalues and eigenfunctions of the spatial structure.

- (a) Separate time and space variables to find the Helmholtz PDE and time dependence.
- (b) Separate spatial variables, $v = X(x)Y(y)$, and find the x structure of eigenfunctions.
- (c) Also find that the y structure of eigenfunctions.
- (d) Hence write down the set of eigenvalues λ .





Figure 6.2: chinese bronze bell, 3000 years old.

6.2 The self-adjoint Sturm–Liouville nature of Helmholtz-like PDEs

Question: how much of the structure of the previous section also holds for more general PDEs in $\mathbb{N}D$?

Consider heat, wave or other PDEs for $u(\vec{x}, t)$ in 2D or 3D—such as for the bell of Figure 6.2—in the form

$$r(\vec{x}) \frac{\partial^\ell u}{\partial t^\ell} = \nabla \cdot [p(\vec{x}) \nabla u] + q(\vec{x}) u, \quad \ell = 1, 2. \quad (6.3)$$

Separate variables, $u = v(\vec{x})T(t)$, to arrive at an eigenproblem of the form

$$\nabla \cdot (p \nabla v) + qv + \lambda rv = 0 \quad \text{in } D, \quad (6.4)$$

with conditions on the boundary of D are of the form

$$\alpha_1(\vec{x})v + \alpha_2(\vec{x}) \frac{\partial v}{\partial n} = 0, \quad \text{where } n \text{ is the normal direction.} \quad (6.5)$$

In these problems, p, q, r, α_j may all depend upon position \vec{x} —because of the shape, knobs and bumps of the bell of Figure 6.2. The eigenvalue λ is constant.

Example 6.1 Consider the Helmholtz PDE on the $2\pi \times \pi$ rectangular membrane of section 6.1.2.

- (a) All eigenvalues we found were real, $\lambda_{n,m} = \frac{1}{4}n^2 + m^2$.
- (b) The countably infinite eigenvalues are $1\frac{1}{4} < 2 < 3\frac{1}{4} < 4\frac{1}{4} < 5 = 5 < 6\frac{1}{4} < 7\frac{1}{4} < 8 \leq \dots$
- (c) Eigenvalues have multiplicity two for $(n, m) = (2, 2)$ and $(4, 1)$ corresponding to spatial structure $\sin(x)\sin(2y)$ and $\sin(2x)\sin(y)$. Also duplication of $16\frac{1}{4} = \lambda_{7,2} = \lambda_{1,4}$ for spatial modes $\sin(7x/2)\sin(2y)$ and $\sin(x/2)\sin(4y)$.
- (d) Here the eigenfunctions form a double Fourier series and is thus complete.
- (e) Similarly, the orthogonality reduces to orthogonality of Fourier series.
- (f) For any eigenfunction $v(\vec{x})$, the so-called Rayleigh quotient gives the corresponding eigenvalue

$$\lambda = \frac{\int_D v_x^2 + v_y^2 d\vec{x}}{\int_D v^2 d\vec{x}}.$$

The eigenfunction corresponding to the smallest eigenvalue has the one sign over the whole domain and is to be zero at the boundaries so roughly $v_1 \approx x(2\pi - x)y(\pi - y)$. The following Matlab code uses this Rayleigh quotient to estimate the eigenvalue as $\lambda_1 \approx \frac{25}{2\pi^2} = 1.2665$ which has remarkably small error of about 1%.

```
syms x y
v=x*(2*pi-x)*y*(pi-y)
vx=diff(v,x)
vy=diff(v,y)
lamb=int(int(vx^2+vy^2,x,0,2*pi),y,0,pi)
lamb=lamb/int(int(v^2,x,0,2*pi),y,0,pi)
```



Theorem 6.1 (self-adjoint eigenproblems in nD). *Consider PDE (6.4), usually with BC (6.5).*

1. All eigenvalues λ are real.
2. There exists a countably infinite number of eigenvalues: there is a smallest, but no largest.

3. Repeated according to the number of linearly independent eigenfunctions, the eigenvalues are $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \dots$ with corresponding linearly independent eigenfunctions $v_1(\vec{x}), v_2(\vec{x}), v_3(\vec{x}), \dots$
4. The eigenfunctions $v_k(\vec{x})$ are complete in that any piecewise smooth function

$$f(\vec{x}) = \sum_{k=1}^{\infty} a_k v_k(\vec{x})$$

for appropriate choice of coefficients a_k .

5. Eigenfunctions corresponding to different eigenvalues are orthogonal with respect to the weight function $r(\vec{x})$:

$$\int_D v_k(\vec{x}) v_l(\vec{x}) r(\vec{x}) d\vec{x} = 0 \quad \text{when } \lambda_k \neq \lambda_l.$$

Further, different eigenfunctions of the same eigenvalue can be made orthogonal by Gram–Schmidt to ensure orthogonality for all $k \neq l$.

6. The **Rayleigh quotient** gives the eigenvalue

$$\lambda = \frac{-\int_{\partial D} p v (\partial v / \partial n) dA + \int_D [p |\nabla v|^2 - q v^2] d\vec{x}}{\int_D v^2 r d\vec{x}},$$

and is typically used to approximate well the eigenvalue given only a rough approximation v to the eigenfunction.¹

To prove properties in Theorem 6.1 proceed as for 1D. Define the linear operator \mathcal{L} such that

$$\mathcal{L}u = \nabla \cdot (p(\vec{x}) \nabla u) + q(\vec{x}) u$$

so the Sturm–Liouville eigenproblem (6.4) becomes $\mathcal{L}u + \lambda ru = 0$. Implicitly with \mathcal{L} are BCs that here are as yet unspecified as we want to be flexible.

Green's formula is crucial First note the following differential form of Lagrange's identity for any smooth functions u and v ,

$$u\mathcal{L}v - v\mathcal{L}u = \nabla \cdot [p(u\nabla v - v\nabla u)].$$

Now integrate both sides over the spatial domain D , provided the integrals exist, and use Gauss' divergence theorem to obtain **Green's formula**

$$\begin{aligned} \int_D u\mathcal{L}v - v\mathcal{L}u d\vec{x} &= \int_D \nabla \cdot [p(u\nabla v - v\nabla u)] d\vec{x} \\ &= \int_{\partial D} p \left(u \frac{\partial v}{\partial n} - v \frac{\partial u}{\partial n} \right) dA, \end{aligned} \quad (6.6)$$

where ∂D denotes the boundary of the domain, and dA an infinitesimal element of the boundary.

¹ In 2D the infinitesimal $d\vec{x}$ denotes an area element such as $dx dy$ or $r d\theta dr$, in 3D $d\vec{x}$ denotes a volume element such as $dx dy dz$, and so on.

Self adjointness When the bcs associated with \mathcal{L} are such that the boundary integral in Green's formula (6.6) vanishes, then the operator \mathcal{L} is self-adjoint. Among other cases, the boundary integral vanishes for the boundary conditions (6.5).

Eigenfunctions are orthogonal Now prove one part of Theorem 6.1. Let there be two eigenfunctions v_k and v_l corresponding to distinct eigenvalues λ_k and λ_l . They satisfy orthogonality with respect to the weight function: $\int_D v_k v_l r d\vec{x} = 0$.

Other proofs are also close to those for the 1D self-adjoint Sturm–Liouville problem.

Exercise 6.2 Use Green's formula for self-adjoint Sturm–Liouville problems in multiple space dimensions, $\nabla \cdot (p(\vec{x}) \nabla v) + q(\vec{x})v + \lambda r(\vec{x})v = 0$ in D subject to suitable boundary conditions on ∂D , to prove that all eigenvalues λ are real when p, q, r are real and $r > 0$. Use contradiction.

- (a) Assuming $\nabla \cdot (p \nabla v) + qv + \lambda rv = 0$ for possibly complex $v(\vec{x})$ and λ , and letting overbars denote complex conjugation, deduce an ODE for \bar{v} .
- (b) For a self-adjoint Sturm–Liouville problem, Green's formula then leads to what identity?
- (c) This identity implies the eigenvalue is real because, for all $\vec{x} \in D$, what?



The new item in Theorem 6.1 is the Rayleigh quotient. Before proving it, let's use it to establish that drums do not explode.

(Haberman 1987, §6.7)

Example 6.2 (drums vibrate). The membrane of a circular drum, radius R say, satisfies the wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u \quad \text{in } D$$

subject to the boundary condition that the membrane is fixed at the rim,

$$u = 0 \text{ on } r = R.$$

Separate variables $u = v(\vec{x})T(t)$ to find

$$\ddot{T} + c^2 \lambda T = 0 \quad \text{and} \quad \nabla^2 v + \lambda v = 0 \text{ in } D.$$

The solution in time is that of oscillations, $T = A \cos \omega t + B \sin \omega t$ for frequency $\omega = c\sqrt{\lambda}$. But this holds only provided eigenvalues $\lambda > 0$: if any eigenvalue $\lambda < 0$, then the temporal solution has exponentially growing modes—an ‘explosion’.

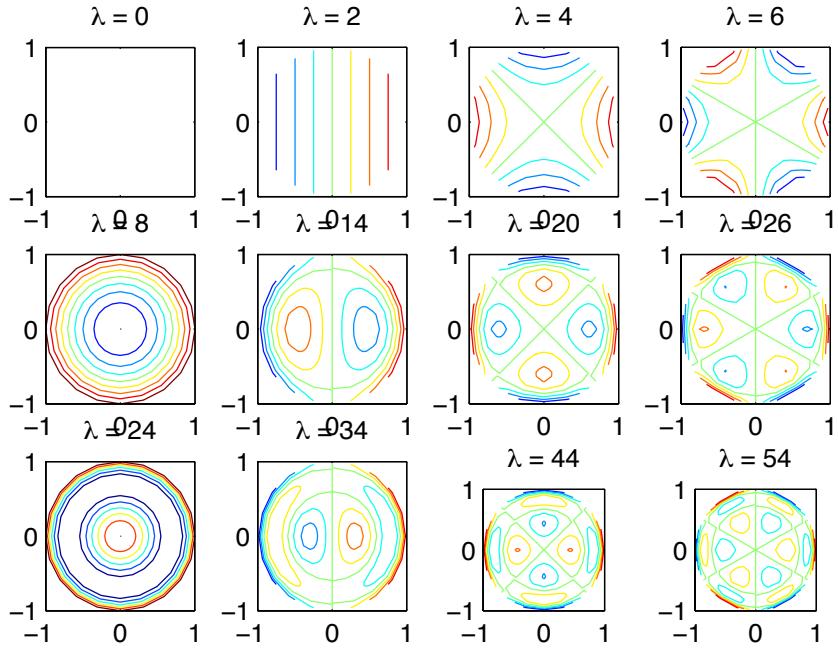


Figure 6.3: contour plots of some eigenfunctions of the shallow water wave PDE in a circular parabolic lake. Each subplot is labelled by its corresponding eigenvalue.

The Rayleigh quotient establishes $\lambda > 0$. Since $\nabla^2 v = \nabla \cdot (\nabla v)$, the spatial structure is a self-adjoint eigenproblem which must thus

- have real (positive) eigenvalues, and hence real frequencies;
- there are an infinite number of modes of oscillation of indefinitely high frequency;
- the spatial patterns of oscillation of the drum are orthogonal.



Proof of the Rayleigh quotient Establish via multiplying the PDE (6.4) by the eigenfunction $v(\vec{x})$ and integrating over the domain.

Example 6.3 (parabolic circular lake). Find the waves, normal modes, in a circular parabolic lake of mid-depth D and radius L ; that is, depth $d(\vec{x}) = D[1 - (x^2 + y^2)/L^2]$. Figure 6.3 plots the modes eigenfunctions and corresponding eigenvalue. Polar coordinates are most convenient so change to (scaled) radius r and angle θ such that $r^2 = (x^2 + y^2)/L^2$ and hence depth $d(\vec{x}) = d(r) = D(1 - r^2)$. An exercise shows we need to solve $\partial^2 \eta / \partial t^2 = \nabla \cdot (g d \nabla \eta)$ which, we are given, becomes

in polar coordinates

$$\frac{\partial^2 \eta}{\partial t^2} = \frac{gD}{L^2} \left[\frac{1}{r} \frac{\partial}{\partial r} \left(r(1-r^2) \frac{\partial \eta}{\partial r} \right) + \frac{1-r^2}{r^2} \frac{\partial^2 \eta}{\partial \theta^2} \right].$$

Since the depth $d(r) = 1 - r^2 \rightarrow 0$ at the ‘beach’ $r = 1$, the BCS are that $\eta(r, \theta, t)$ is bounded as $r \rightarrow 1$.



7 Computational integration

Chapter Contents

7.1	1D heat/diffusion PDE raises fundamental issues	72
7.2	Crank–Nicolson schemes are reasonably stable and accurate	75
7.3	Invoke sparse matrices for implicit schemes	77
7.4	Crank–Nicolson discretises wave systems	81
7.5	Second order PDEs in 2D	84

This module explores fundamental issues of discretise modelling in multiple space-time dimensions. We explore linear PDEs with constant coefficients since we have to simulate these acceptably before addressing nonlinearity and spatially varying coefficients.

7.1 1D heat/diffusion PDE raises fundamental issues

(Haberman 1987, §13.3)

Model problem Consider a rod with temperature field $u(x, t)$ for $0 \leq x \leq 1$ and for times $t \geq 0$. Say the rod does not lose heat from the interior so the governing PDE is $u_t = Ku_{xx}$, but the extremes are kept at fixed temperature $u(0, t) = u(1, t) = 0^\circ\text{C}$, and there is some initial value $u(x, 0) = u^0(x)$.

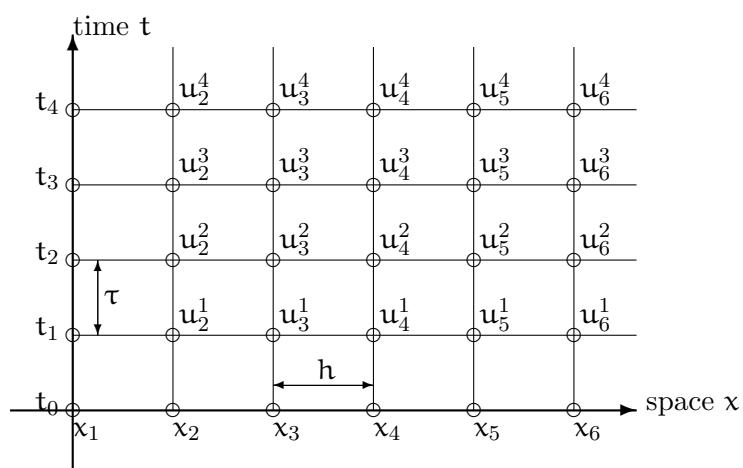


Figure 7.1: example discrete grid in the xt -plane with grid spacing h in x and time-step τ in t .

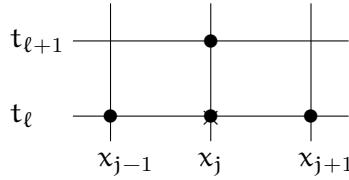


Figure 7.2: computational stencil for the forward time, centred space, discretisation (7.1) of the heat PDE.

Computational discretisation The space-time domain is $0 \leq x \leq 1$ and $t \geq 0$. Figure 7.1 illustrates that we define a space-time grid with space $x_j = (j - 1)h$ for $j = 1, 2, \dots, N$ and times $t_\ell = \ell\tau$ for $\ell = 0, 1, 2, \dots$. Define grid values

$$u_j^\ell := u(x_j, t_\ell), \quad \text{then } u_t(x_j, t_\ell) = u_t|_j^\ell = \frac{\partial u}{\partial t}|_j^\ell \text{ and so on.}$$

Using only values on such a grid,

$$u_j^{\ell+1} \approx u_j^\ell + s(u_{j+1}^\ell - 2u_j^\ell + u_{j-1}^\ell), \quad s := \frac{K\tau}{h^2}. \quad (7.1)$$

Given values of u_j^ℓ at time t_ℓ , this approximation explicitly computes the values at time $t_{\ell+1}$ —see Figure 7.2.

Example 7.1 Set coefficient $K = 1$, the simplest spatial discretisation with $h = 1/2$, and the time-step of $\tau = 1/24$. What is predicted for u_2^ℓ ? ♣

Operator algebra proves consistency Use subscripts t and x to denote operators in the corresponding variable. The computation (7.1) is equivalent to

$$\partial_t = \frac{1}{\tau} \log [1 + s4 \sinh^2(h\partial_x/2)]$$

```

1 syms tau h dx s
2 taylor(1/tau*log(1+s*4*sinh(h*dx/2)^2),h)
3 expand(ans)

```

Consequently, the equivalent PDE for (7.1) is

$$\partial_t u = K\partial_x^2 u + Kh^2(\frac{1}{12} - \frac{1}{2}s)\partial_x^4 u + O(h^4).$$

That is, the numerical scheme exactly solves this slightly different PDE. This PDE approximates the original heat/diffusion PDE apparently with error $O(h^2)$ —although since $sh^2 = \tau K$ appears the error is more informatively $O(h^2, \tau)$.

If we happen to choose a grid so that $s = 1/6$, then for this particular PDE the leading error disappears giving a method of better error $O(h^4)$.

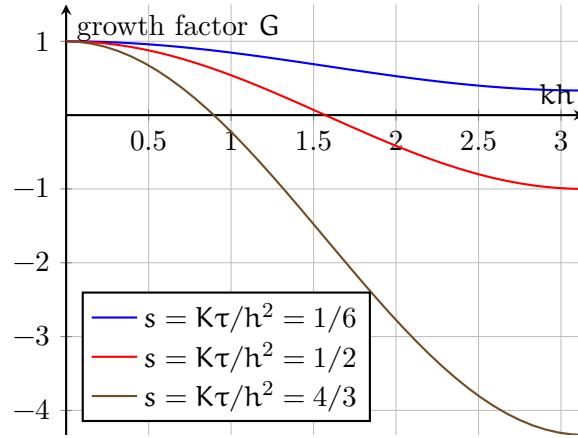


Figure 7.3: growth factor $G(kh)$ for various parameters $s = K\tau/h^2$: illustrates instability for $s > 1/2$.

Recall the Lax Equivalence Theorem 4.2: consistency + stability = convergence. We need the computation to be stable as well as consistent.

Example 7.2 Set coefficient $K = 1$, the simplest spatial discretisation with $h = 1/2$, and the time-step of $\tau = 1/3$. What is predicted for u_2^ℓ ? ♣

von Neumann stability analysis Can small perturbations grow to swamp solutions? Since the computation rule (7.1) is linear, by superposition the evolution of perturbations to a solution satisfies the same rule. Thus just explore the stability of (7.1). Analogous to wave dispersion relations, seek λ such that $u = e^{ikx+\lambda t}$ is a solution, for real wavenumber k : but rather than λ , it is more convenient to seek the *growth factor* $G := e^{\lambda \tau}$ (per time step).

Seek $u^\ell(x) = e^{ikx+\lambda t_\ell} = e^{ikx+\lambda \tau \ell} = e^{ikx} e^{\lambda \tau \ell} = G^\ell e^{ikx}$. The computational rule (7.1) then determines the growth factor (Figure 7.3)

$$G = 1 - s4 \sin^2(kh/2).$$

A perturbation decays only when parameter $s < 1/2$ as only then is $|G| < 1$ for all k . Since $s = \tau K/h^2$ we must have time-step $\tau < h^2/(2K)$.

This is a severe limitation on the time-step: seeking higher accuracy by say doubling the spatial resolution, halving h , means the time-step τ has to be reduced by a quarter. Results in eight times as much computation to simulate over a given time.

Example 7.3 What modes are unstable if the discretisation has parameter $s = 2/3$? ♣

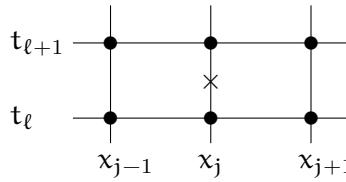


Figure 7.4: computational stencil for the centred time, centred space, Crank–Nicolson, discretisation (7.2) of the heat PDE.

Exercise 7.1 Explore a simple space-time discretisation of the linear 1D wave PDE $u_t + cu_x = 0$. We would like the discretisation to appropriately model the travelling wave solutions $u(x, t) = f(x - ct)$: let's explore. Define time grid points $t_\ell = \ell\tau$ with time step τ , define space grid points $x_j = jh$ with space step h , and consider the grid values $u_j^\ell = u(x_j, t_\ell)$.

- (a) For the wave PDE $u_t + cu_x = 0$, at a general grid point (x_j, t_ℓ) , use a forward difference in time to approximate u_t and a centred difference in space to approximate u_x in terms of grid values. Derive the discretisation.
- (b) Apply a von Neumann stability analysis by seeking solutions to the discretisation $u_j^\ell = e^{ikx_j} G^\ell$ with any wavenumber k and some growth factor G to be determined. Substitute and find the growth factor.
- (c) Using this growth factor, determine for what range of parameter s there is unstable growing solutions: $|G| > 1$, for some wavenumber(s) k .



7.2 Crank–Nicolson schemes are reasonably stable and

(Haberman 1987, pp.502–3)

Section 7.1 implemented a scheme that used a forward difference in time and centred differences in space: abbreviated to FTCS. Other combinations are: BTCS, backward in time and centred in space; CTCS, centred in time and centred in space; and so on. Crank–Nicolson schemes are CTCS (Crank & Nicolson 1947).

John Crank (1916–2006) was a mathematical physicist, best known for his work on the numerical solution of partial differential equations. Phyllis Nicolson (1917–1968) was a British mathematician and research fellow from 1946–49 at Girton College, Cambridge, most known for her work on the Crank–Nicolson scheme.

Computational discretisation As shown in Figure 7.4, consider the heat PDE at the cross in the centre of the stencil. Substitute into the PDE and rearrange to

$$-su_{j-1}^{\ell+1} + 2(1+s)u_j^{\ell+1} - su_{j+1}^{\ell+1} = su_{j-1}^\ell + 2(1-s)u_j^\ell + su_{j+1}^\ell \quad (7.2)$$

where as before $s := \tau K/h^2$. Equation (7.2) is an implicit equation for the quantities at time $t_{\ell+1}$ given the quantities at time t_ℓ on the RHS.

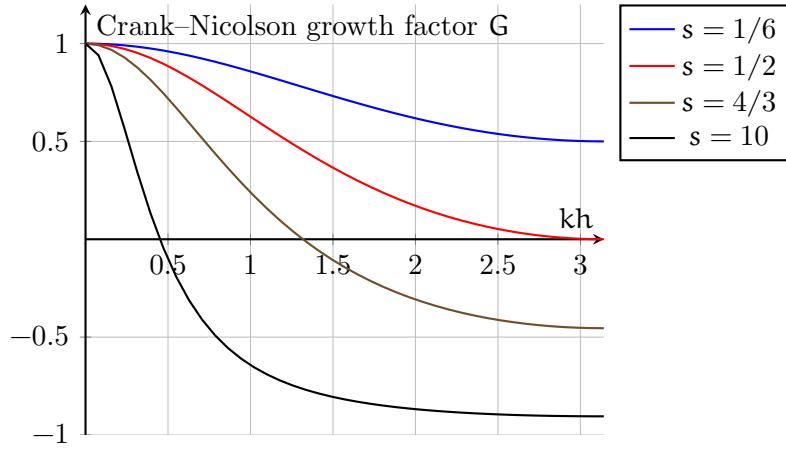


Figure 7.5: growth factor $G(kh)$ for various parameters $s = K\tau/h^2$ illustrating stability for all s .

Example 7.4 Set coefficient $K = 1$, the simplest spatial discretisation with $h = 1/2$, and the time-step of $\tau = 1/3$. What is predicted for u_2^ℓ ? ♣

Operator algebra proves consistency From the stencil of Figure 7.4, the scheme (7.2) is

$$\frac{1}{\tau} \delta_t u_j^{\ell+1/2} = K \mu_t \frac{1}{h^2} \delta_x^2 u_j^{\ell+1/2}.$$

Arranging gives

$$\mathcal{E}_t = \frac{1 + s2 \sinh^2(h\partial_x/2)}{1 - s2 \sinh^2(h\partial_x/2)} \quad (7.3)$$

We use this equation in two ways. Firstly, recalling $\mathcal{E}_t = e^{\tau \partial_x}$, deduce

$$\partial_t = \frac{1}{\tau} \log \left[\frac{1 + s2 \sinh^2(h\partial_x/2)}{1 - s2 \sinh^2(h\partial_x/2)} \right].$$

Using Taylor series then gives the equivalent PDE as

$$u_t = Ku_{xx} + \frac{1}{12}Kh^2u_{xxxx} + \mathcal{O}(h^4),$$

which is consistent to the heat PDE to error $\mathcal{O}(h^2)$.

von Neumann stability analysis The operator algebra short-cuts the stability analysis: seek solutions of (7.2) of the form $u = G^\ell e^{ikx}$ (for factor $G = e^{\lambda\tau}$):

$$G = \frac{1 - s2 \sin^2(kh/2)}{1 + s2 \sin^2(kh/2)}$$

as in Figure 7.5. Letting say $z = s2 \sin^2(kh/2) \in [0, \infty)$, then $G = (1 - z)/(1 + z)$ which satisfies $-1 < G \leq 1$ (sketch a graph).

Hence $|G^\ell|$ decays to zero for all wavenumbers k : the Crank–Nicolson scheme is stable for all discretisation parameters $s = \tau K/h^2$.

By the Lax Equivalence Theorem 4.2, discretisation solutions should converge to a solution of the PDE as $h, \tau \rightarrow 0$.

Aliasing limits relevant wavenumbers On a spatial grid of spacing h , we need only consider wavenumbers $|kh| \leq \pi$ (as in Figures 7.3 and 7.5). For example, the highest wavenumber $k = \pi/h$ is the ‘sawtooth’ mode on the grid: $e^{ikx_j} = e^{i(\pi/h)(jh)} = e^{i\pi j} = (-1)^j$.

Exercise 7.2 Explore a Crank–Nicolson space-time discretisation of the linear 1D wave PDE $u_t + cu_x = 0$. Define time grid points $t_\ell = \ell\tau$ with time step τ , define space grid points $x_j = jh$ with space step h , and consider the grid values $u_j^\ell = u(x_j, t_\ell)$.

- (a) Consider the wave PDE $u_t + cu_x = 0$, at the point $(x_{j+1/2}, t_{\ell+1/2})$, halfway from the ℓ th time to the next and halfway from the j th location and the next. Use a centred difference in time to approximate u_t and a centred difference in space to approximate u_x . But get both in terms of grid values by averaging, respectively, the time derivative over x_j and x_{j+1} , and the space derivative over t_ℓ and $t_{\ell+1}$. Derive the discretisation, where $s := c\tau/h$.
- (b) Apply a von Neumann stability analysis by seeking solutions to the discretisation $u_j^\ell = e^{ikx_j} G^\ell$ with any wavenumber k and some growth factor G to be determined. Substitute and derive the growth factor [recall $1 \pm e^{ikh} = e^{ikh/2}(e^{-ikh/2} \pm e^{ikh/2})$].
- (c) Using this growth factor, determine for what range of parameter s there is unstable growing solutions in this Crank–Nicolson scheme: $|G| > 1$, for some wavenumber(s) k .



7.3 Invoke sparse matrices for implicit schemes

This section codes the Crank–Nicolson scheme for the heat PDE ($K = 1$) $u_t = u_{xx}$ on $0 < x < 1$ with boundary conditions $u(0, t) = 0$ and $u(1, t) = \sin 8t$ with initial condition $u(x, 0) = 6x(1 - x)$. Figure 7.7 shows the computed solution for times $0 \leq t \leq 0.2$. This example introduces a computational framework adaptable to many PDE problems, and introduces the incredibly useful Matlab function `sparse()`.



Figure 7.6: Phyllis Nicolson (top-right) with Douglas Hartree at the Hartree Digital Analyser at Manchester University (c1945).
[\[http://www.cs.man.ac.uk/CCS/res/images/res52b.jpg\]](http://www.cs.man.ac.uk/CCS/res/images/res52b.jpg)

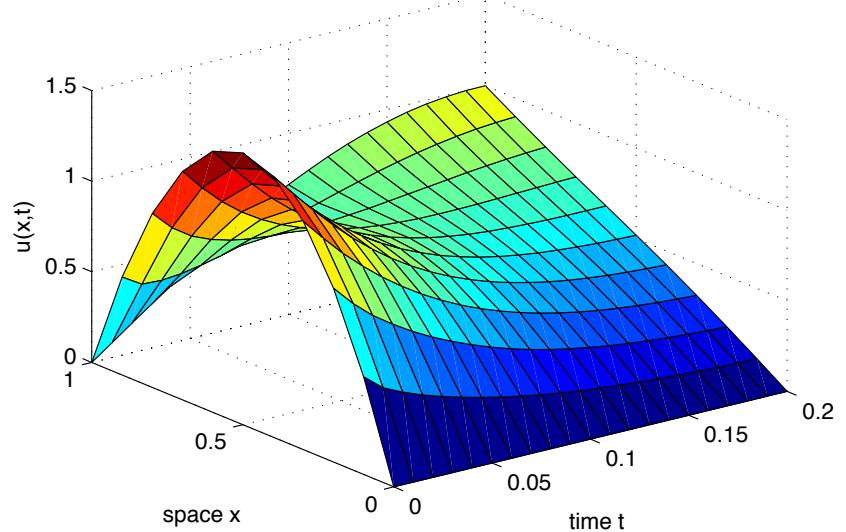


Figure 7.7: Algorithm 7.1 computes this solution of the heat PDE $u_t = u_{xx}$.

The key is to treat the computational rule (7.2) as a set of linear equations. With the BCs, write as

$$\underbrace{\begin{bmatrix} 1 & 0 \\ -s & 2(1+s) & -s \\ & -s & 2(1+s) & -s \\ & & \ddots & \ddots & \ddots \\ & & & -s & 2(1+s) & -s \\ & & & & 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} u_1^{\ell+1} \\ u_2^{\ell+1} \\ \vdots \\ u_N^{\ell+1} \end{bmatrix}}_{\bar{u}^\ell} = \text{RHS},$$

where the RHS comes from the RHS of (7.2) and the boundary values, namely

$$\text{RHS} = \begin{bmatrix} u(x_1, t_{\ell+1}) \\ su_1^\ell + 2(1-s)u_2^\ell + su_3^\ell \\ su_2^\ell + 2(1-s)u_3^\ell + su_4^\ell \\ \vdots \\ su_{N-2}^\ell + 2(1-s)u_{N-1}^\ell + su_N^\ell \\ u(x_N, t_{\ell+1}) \end{bmatrix}.$$

Most of matrix A is zero and it would be silly to store and manipulate such zero entries. Matlab's sparse facility not only provides an economical storage, but also efficient algorithms to manipulate.¹

- `A=sparse(i,j,a,m,n)` creates an $m \times n$ matrix of ‘zeros’ except for the elements $A_{i_k j_k} = a_k$ for $k = 1, \dots, \text{length}(i) = \text{length}(j)$.

Algorithm 7.1 creates the sparse matrix A in lines 17–20: it adds the interior diagonal (the case $j=j$) to the two off-diagonals (cases $j=j+1$ and $j=j-1$), and then adds the corner ones of the BCs. Figure 7.8 shows the non-zero entries.

Algorithm 7.1 adapts the earlier Algorithm 4.1 by removing `ode15s` and replacing it with an explicit loop to step forward in time. Making errors is typical, so `spy()` checks the structure of A , and `condest()` checks the condition number to catch some coding/formulation errors: $1 < \text{good} < 10^2 < \text{poor} < 10^4 < \text{bad} < 10^8 < \text{terrible}$.

Matlab's `ode15s` is also an implicit scheme (solves linear equations at each time step), but is higher order, with variable time step size, and error control.

¹ Remember that Matlab, as does almost all good software, invokes LAPACK and BLAS. The ongoing outcomes of one of the longest lasting public domain software developments, LAPACK and BLAS provide the best and fastest computation of linear algebra tasks, both dense and sparse, for the complicated architecture of modern computers.

Algorithm 7.1: Crank–Nicolson scheme to solve the heat PDE. This framework adapts to many similar PDE problems.

```

1 function heatCN
2 % Compute solutions of heat PDE with Crank-Nicholson
3 % AJR, 9/9/2014
4 nPoints=11 % number of spatial grid points
5 x=linspace(0,1,nPoints)'
6 j=2:nPoints-1
7 dx=x(2)-x(1) % space step of grid
8 nTimes=21 % number of time steps
9 t=linspace(0,0.2,nTimes)
10 dt=t(2)-t(1) % time step
11 % storage and initial condition
12 u=nan(nPoints,nTimes);
13 u(:,1)=6*x.* (1-x);
14
15 % sparse matrix of the CN scheme
16 s=dt/dx^2
17 A=sparse(j,j,2*(1+s),nPoints,nPoints) ...
18 +sparse(j,j-1,-s,nPoints,nPoints) ...
19 +sparse(j,j+1,-s,nPoints,nPoints) ...
20 +sparse([1 nPoints],[1 nPoints],1,nPoints,nPoints);
21 spy(A),print -depsc2 heatCNspy
22 condA=condeest(A) % check condition number not large
23
24 for l=1:nTimes-1 % time step loop
25   rhs=[uleft(t(l+1))
26       s*u(j-1,l)+2*(1-s)*u(j,l)+s*u(j+1,l)
27       uright(t(l+1))];
28   u(:,l+1)=A\rhs; % solve linear eqns A.u=rhs
29 end
30
31 surf(t,x,u)
32 ylabel('space\_{\}x'), xlabel('time\_{\}t'), zlabel('u(x,t)')
33 %-----
34 function ua=uleft(t), ua=zeros(size(t));
35 function ub=uright(t), ub=sin(8*t);
```

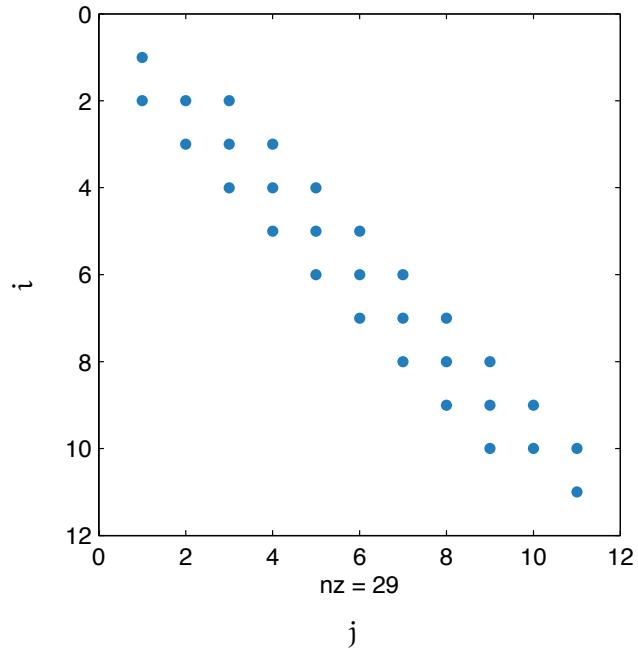


Figure 7.8: the function `spy(A)` displays the non-zero entries in a sparse matrix. Use as a check.

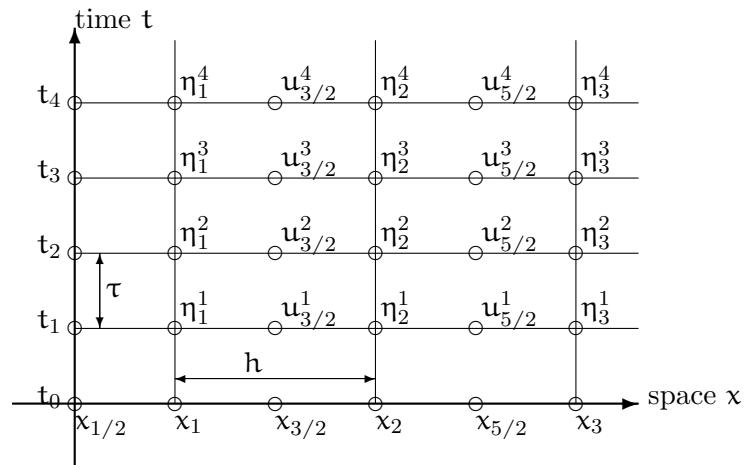


Figure 7.9: example discrete grid in the xt -plane with time-step τ in t , and a staggered grid in x , spacing h .

7.4 Crank–Nicolson discretises wave systems

This section develops a discretisation and simulation of wave systems expressed as two coupled first-order PDEs. Specifically, the linearised shallow water wave PDEs

$$\eta_t = -(du)_x, \quad u_t = -g\eta_x, \quad (7.4)$$

for surface height $\eta(x, t)$ and water velocity $u(x, t)$. The technique adapts to nonlinear versions and other wave problems.

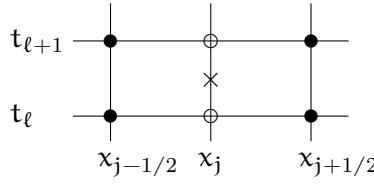


Figure 7.10: computational stencil for the centred time, centred space, Crank–Nicolson, discretisation (7.4) of the wave PDEs (7.4).

Computational discretisation Use a staggered space grid as illustrated in Figure 7.9 (until we get to Matlab implementation, use half-integer indices). Figure 7.10 shows a small part of the staggered grid. Centred at the cross at $(x_j, t_{l+1/2})$:

- for integer j , using subscripts x and t on operators indicate the direction/variable of the operator,
so $\eta_t = -(du)_x$ is discretised $\frac{1}{\tau} \delta_t \eta + \frac{1}{h} \mu_t \delta_x (du) \approx 0$;
- for half-integer j ,
so $u_t = -g\eta_x$ is discretised $\frac{1}{\tau} \delta_t u + \frac{g}{h} \mu_t \delta_x \eta \approx 0$.

The Crank–Nicolson scheme for waves is to solve

$$\delta_t \eta + \frac{\tau}{h} \mu_t \delta_x (du) = 0, \quad \delta_t u + \frac{g\tau}{h} \mu_t \delta_x \eta = 0, \quad (7.5)$$

evaluated at half-integer ℓ , and integer/half-integer j , respectively.

Consistency Because we used centred differences, the discretisation (7.5) should have error $\mathcal{O}(\tau^2, h^2)$. For example, rearrange the first PDE: so that in effect

$$\begin{aligned} \delta_t \eta &= -\frac{2}{\tau} \tanh^{-1} \left[\frac{\tau}{h} \sinh(h\partial_x/2) \right] (du) \\ &= \left\{ -\partial_x - \frac{h^2 + 2\tau^2}{24} \partial_x^3 + \dots \right\} (du) \\ &\rightarrow -(du)_x \text{ as } h, \tau \rightarrow 0, \end{aligned}$$

upon using the following Taylor expansion in which the variable z is a dummy that counts the number of small h and small τ (sometimes `expand()` is better than `simplify()`)

```
1 syms tau h dx z
2 taylor(-2/(tau*z)*atanh(tau/h*sinh(h*z*dx/2)),z)
3 expand(ans)
```

Analogously, the second of (7.5) is also consistent with the corresponding PDE (7.4).

Stability via operator algebra Must consider the pair of equations (7.5) as a coupled system. Further, stability analysis only

addresses constant coefficients, constant depth d . Put into a matrix-vector form, but the two discretisations are shifted by half a grid spacing so should really multiply the second by the half-shift $\mathcal{E}_x^{1/2}$:

$$\begin{bmatrix} \delta_t & \frac{d\tau}{h} \mu_t \delta_x \\ \mathcal{E}_x^{1/2} \frac{g\tau}{h} \mu_t \delta_x & \mathcal{E}_x^{1/2} \delta_t \end{bmatrix} \begin{bmatrix} \eta \\ u \end{bmatrix} = \vec{0}.$$

This implies the two cases, for $s := \sqrt{gd\tau}/h$,

$$\mathcal{E}_t = \frac{1 \pm s \sinh(h\partial_x/2)}{1 \mp s \sinh(h\partial_x/2)}.$$

Look for wave solutions by seeking solutions $e^{i(kx - \omega t)} = e^{i(kx_j - \omega t_\ell) = G^\ell e^{ikx_j}}$ for factor $G = e^{i\omega\tau}$. To correspond with real frequency ω we need the factor G to be complex with unit magnitude, $|G| = 1$. If $|G| > 1$ or $|G| < 1$, then the corresponding ‘frequency’ would be complex and the corresponding solution would *not* be a wave. The above equation applied to $G^\ell e^{ikx}$ implies

$$G = \frac{1 \pm s \sinh(ikh/2)}{1 \mp s \sinh(ikh/2)} = \frac{1 \pm is \sin(kh/2)}{1 \mp is \sin(kh/2)}.$$

Hence the magnitude of the factor G is

$$|G| = \frac{|1 \pm is \sin(kh/2)|}{|1 \mp is \sin(kh/2)|} = \frac{\sqrt{1 + s^2 \sin^2(kh/2)}}{\sqrt{1 + s^2 \sin^2(kh/2)}} = 1,$$

for both cases and for all s, k, h . Marvellous: this Crank–Nicolson scheme has only waves.

Determine the numerical frequency of the waves from $G = e^{i\omega\tau}$: upon using $s = c\tau/h$ for $c = \sqrt{gd}$, the frequency

$$\begin{aligned} \omega &= \pm \frac{2}{\tau} \tan^{-1} \left[\frac{c\tau}{h} \sin \frac{kh}{2} \right] \\ &= \pm \left[ck - \left(\frac{1}{24} ch^2 + \frac{1}{12} c^3 \tau^2 \right) k^3 + \mathcal{O}(k^5) \right] \end{aligned}$$

This is the dispersion relation for the numerical water waves.

Implement with sparse matrices First, for programming with a staggered grid change from the half-index Figure 7.9 to the integer indices of Figure 7.11. Write the discrete equations (7.5) in matrix form.

$$\begin{bmatrix} 1 & 0 & & & & & & \\ -\frac{\tau d_1}{2h} & 1 & \frac{\tau d_3}{2h} & & & & & \\ & -\frac{g\tau}{2h} & 1 & \frac{g\tau}{2h} & & & & \\ & & -\frac{\tau d_3}{2h} & 1 & \frac{\tau d_5}{2h} & & & \\ & & & -\frac{g\tau}{2h} & 1 & \frac{g\tau}{2h} & & \\ & & & & \ddots & \ddots & \ddots & \end{bmatrix} \begin{bmatrix} u_1^{\ell+1} \\ \eta_2^{\ell+1} \\ u_3^{\ell+1} \\ \eta_4^{\ell+1} \\ \vdots \\ u_N^{\ell+1} \end{bmatrix} = \text{RHS}$$

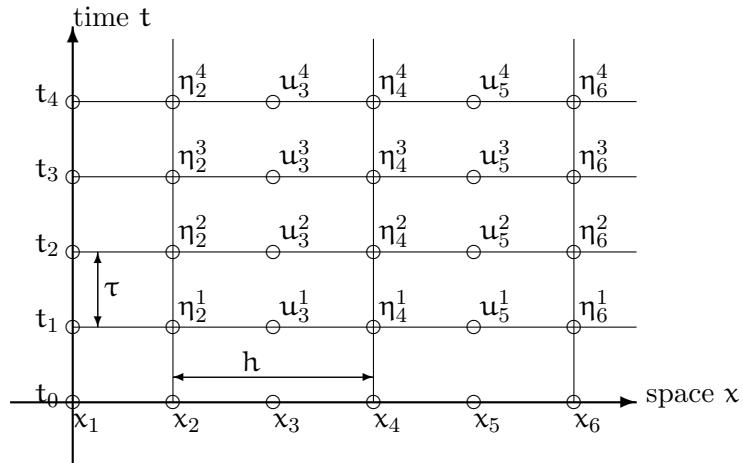


Figure 7.11: example discrete grid in the xt -plane with time-step τ in t , and a staggered grid in x , spacing h : now with spatial grid indexing set to be integers for computation.

where the right-hand side

$$\text{RHS} := \begin{bmatrix} u(x_1, t_{\ell+1}) \\ \eta_2^\ell - \frac{\tau}{2h} [d_3 u_3^\ell - d_1 u_1^\ell] \\ u_3^\ell - \frac{g\tau}{2h} [\eta_4^\ell - \eta_2^\ell] \\ \eta_4^\ell - \frac{\tau}{2h} [d_5 u_5^\ell - d_3 u_3^\ell] \\ u_5^\ell - \frac{g\tau}{2h} [\eta_6^\ell - \eta_4^\ell] \\ \vdots \\ u(x_N, t_{\ell+1}) \end{bmatrix}$$

Solving this matrix-vector equation step forwards in time to simulate shallow water waves: Figure 7.12 shows an example.

Algorithm 7.2 implements this Crank–Nicolson scheme. The algorithm is a synthesis of the Crank–Nicolson scheme for the heat PDE, Algorithm 7.1, and the staggered grid for waves, Algorithm 5.1.

7.5 Second order PDEs in 2D

This section considers the PDE

$$\frac{\partial u}{\partial t} = \nabla^2 u + f(\vec{x})u \quad (7.6)$$

in a rectangular domain D with Dirichlet boundary conditions. More complicated PDEs and BCs are similar to this fundamental case.

For example, Figure 7.13 plots the solution field $u(x, y, t)$ at one time in a 3×2 rectangle for forcing factor $f = \frac{1}{2}(x^2 + y^2)$.

The challenge is to modify Algorithm 4.1 in order to cater for the two spatial dimensions here. Algorithm 7.3 is my result, one that

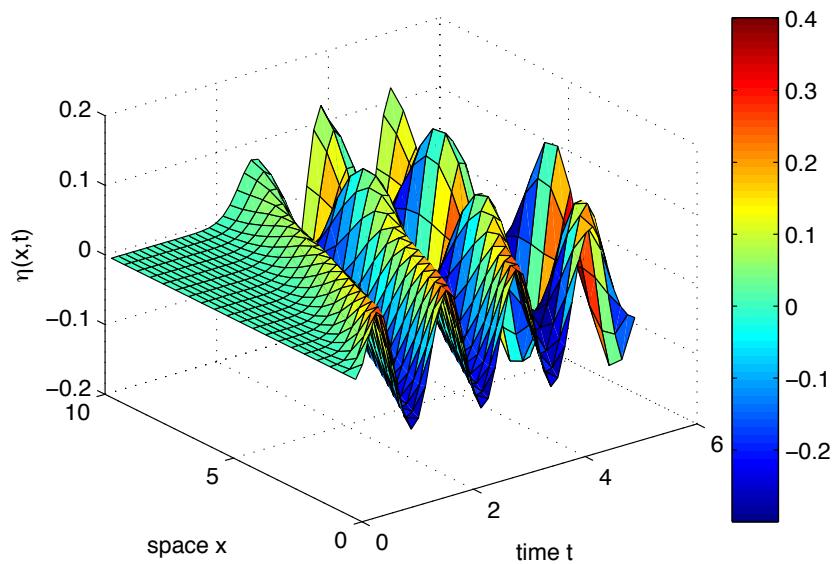


Figure 7.12: Algorithm 7.2 computes this solution of the linearised shallow water wave PDEs (7.4).

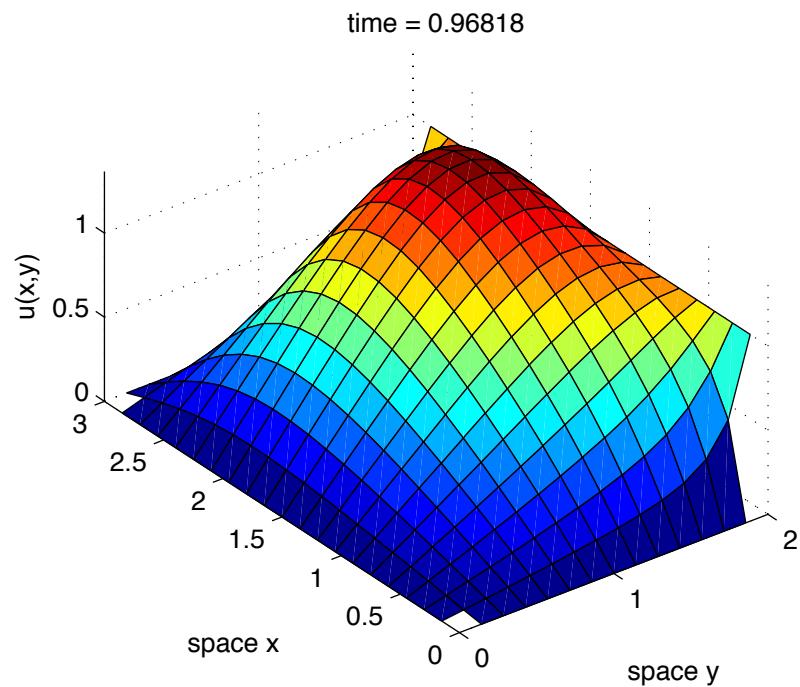


Figure 7.13: The field $u(x,y,t)$ at time $t \approx 1$ in the evolution of the 2D heat PDE (7.6) for forcing factor $f(\vec{x}) = \frac{1}{2}(x^2 + y^2)$.

Algorithm 7.2: Crank–Nicolson scheme to compute the linearised shallow water wave PDEs (7.4) on a staggered grid.

```

1 function shallowCNlin
2 % Compute solutions of linear shallow water PDEs with
3 % Crank–Nicolson. AJR, 11/9/2014
4 % Use staggered grid ( $u_1, h_2, u_3, h_4, \dots, u_N$ )
5 g=10
6 nPoints=41 % should be odd for the BCs
7 x=linspace(0,10,nPoints)'
8 d=1+3*x/x(end);% some varying depth
9 u=3:2:nPoints-1% index of  $u$  points
10 h=2:2:nPoints-1% index of  $h$  points
11 dx=x(3)-x(1); % space grid step
12 nTimes=41 % number of time steps
13 t=linspace(0,5,nTimes);
14 dt=t(2)-t(1) % time step size
15 % storage and initial condition
16 uh=nan(nPoints,nTimes);
17 uh(:,1)=0;
18
19 % sparse matrix of the CN scheme
20 s=dt/dx/2
21 A=spye(nPoints,nPoints) ...
22 +sparse(u,u-1,-g*s,nPoints,nPoints) ...
23 +sparse(u,u+1,+g*s,nPoints,nPoints) ...
24 +sparse(h,h-1,-d(h-1)*s,nPoints,nPoints) ...
25 +sparse(h,h+1,+d(h+1)*s,nPoints,nPoints);
26 spy(A),print -depsc2 shallowCNspy
27 condA=conde(A) % check condition number not large
28
29 rhs=nan(size(x));
30 for l=1:nTimes-1 % time step loop
31   rhs([1 nPoints])=[uleft(t(l+1));uright(t(l+1))];
32   rhs(u)=uh(u,l)-g*s*(uh(u+1,l)-uh(u-1,l));
33   rhs(h)=uh(h,l)-s*(d(h+1).*uh(h+1,l)-d(h-1).*uh(h-1,l));
34   uh(:,l+1)=A\rhs;% solves linear eqns  $A.u=rhs$ 
35 end
36
37 surf(t,x(h),uh(h,:),uh(h+1,:))
38 ylabel('space\_\mathbf{x}'), xlabel('time\_\mathbf{t}'), zlabel('\'\eta(\mathbf{x},\mathbf{t})')
39 colorbar, print -depsc2 shallowCNlin
40 %-----
41 function u=uleft(t), u=0.3*sin(5*t);
42 function u=uright(t), u=zeros(size(t));

```

provides a framework for analogous problems. One of the main challenges is to transform between the naturally 2D arrays of data in xy -domain and the 1D column/row vectors of Matlab functions such as `ode15s`, `fsolve`, and `eigs`. (Typically use the convention that capital letter variables are 2D arrays, and lowercase variables are 1D.) The first decision is to choose the x -direction to go down 2D arrays, and the y -direction to go across. The details then follow. Algorithm 7.3 shows the output $u(x, y, t)$ as a little movie—with non-uniform rate in time.

Algorithm 7.3 is a foundation for many variations in multiple spatial dimensions.

- Straightforwardly change the PDE or BCS for other problems.
- Changing the domain shape is typically more difficult: either change to suitable curvilinear coordinates, or unpack and pack 1D arrays more complicatedly into a subset of a 2D array.
- Use arrays of higher dimension to simulate more spatial dimensions.
- Modify to seek equilibria and the linearised dynamics about any such equilibrium.
- Could synthesise much of this algorithm with the Crank–Nicolson methods in order to explicitly code time integration steps.

Algorithm 7.3: spatial discretisation of the heat PDE with reaction in two space dimensions can be visualised as a ‘movie’.

```

1 function heat2d
2 % Simulates heat diffusion in a rectangle with a reactive
3 % source. In an array, x varies downwards and y varies
4 % across. AJR 17/9/2014
5 global j k X Y
6 nx=21, ny=15 % number of grid points in x,y directions
7 x=linspace(0,3,nx)';
8 y=linspace(0,2,ny);
9 j=2:nx-1; k=2:ny-1; % indices of interior points in x and y
10 [Y,X]=meshgrid(y,x);% 2D grid points in space
11 U0=rand(size(X)); % initial condition, generally fn(X,Y)
12
13 [t,u]=ode15s(@dudt,[0 1],reshape(U0(j,k),[],1));
14
15 U=nan(size(X)); % unpack rows of u into this 2D
16 for l=1:length(t)
17   U(j,k)=reshape(u(l,:),nx-2,ny-2);
18   U([1 end],k)=[uleft(t(1),y(k)); uright(t(1),y(k))];
19   U(j,[1 end])=[ubottom(t(1),x(j)) utop(t(1),x(j))];
20   surf(y,x,U) % draw surface
21   xlabel('space\u2193y'), ylabel('space\u2192x'), zlabel('u(x,y)')
22   title(['time=\u2193' num2str(t(1))]), axis equal
23   pause(0.1) % pause to see
24 end
25 %-----
26 function ut=dudt(t,u)
27 global j k X Y
28 dx=X(2,1)-X(1,1); dy=Y(1,2)-Y(1,1); % grid spacings
29 U=nan(size(X)); % unpack column u into this 2D
30 U(j,k)=reshape(u,length(j),length(k));
31 U([1 end],k)=[uleft(t,Y(1,k)); uright(t,Y(end,k))];
32 U(j,[1 end])=[ubottom(t,X(j,1)) utop(t,X(j,end))];
33 ut=(U(j+1,k)-2*U(j,k)+U(j-1,k))/dx^2 ...
34   +(U(j,k+1)-2*U(j,k)+U(j,k-1))/dy^2 ...
35   +(1/2)*(X(j,k).^2+Y(j,k).^2).*U(j,k);
36 ut=ut(:); % reshapes ut into a column
37 %-----
38 function u=uleft(t,y), u=zeros(size(y));
39 function u=uright(t,y), u=zeros(size(y));
40 function u=ubottom(t,x), u=zeros(size(x));
41 function u=utop(t,x), u=ones(size(x));

```

8 General wave dynamics

Chapter Contents

8.1	Water waves in finite depth	89
8.2	Energy travels at the group velocity	92
8.3	Wave propagation in multi-dimensions	94

8.1 Water waves in finite depth

Many wave systems have cross-sectional structure. For example, a flexing beam is compressed on one side and stretched on the other, and this variation is the crucial ingredient in the beam flexing. Also, light travelling along a fibre optic cable has a cross-sectional profile dictated by the trapping structure of the fibre. Similarly, water waves have structure in depth as the wave propagates across the surface (Figure 8.1). This section develops a mathematical description for water waves.

Other courses and/or texts will derive the mathematical equations stated next. Consider 2D with x horizontal and z vertical with $z = -d$ a flat bed, and $z = 0$ the rest water level but $z = \eta(x, t)$ the water surface when in motion. The fluid moves with velocity field determined by a velocity potential $\phi(x, z, t)$: $u(x, z, t) = \phi_x$ and $w(x, z, t) = \phi_z$. Then *small amplitude* irrotational waves are governed by Laplace's PDE in the fluid with boundary conditions:

$$\begin{aligned}\nabla^2\phi &= \phi_{xx} + \phi_{zz} = 0 \quad -d < z < 0; \\ \phi_z &= 0 \quad \text{on } z = -d; \\ \eta_t &= \phi_z \text{ and } \phi_t + g\eta = 0 \quad \text{on } z = 0.\end{aligned}$$



Figure 8.1: water waves generated by a ferry off Italy.

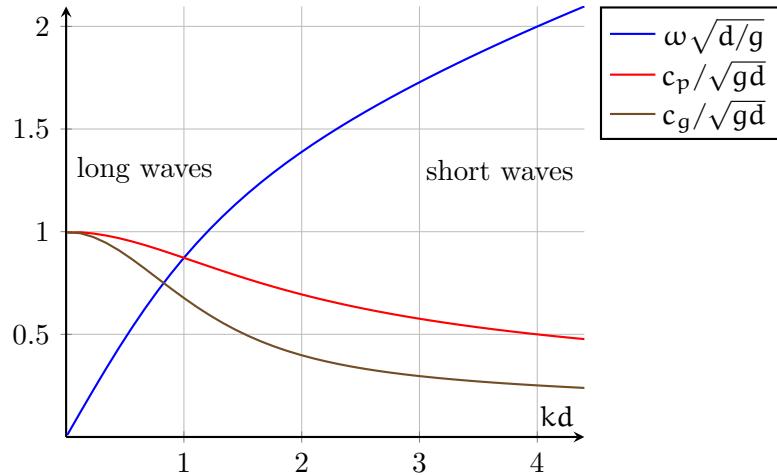


Figure 8.2: frequency, wave/phase speed and group velocity of water waves in various depths and wavelengths from the dispersion relation (8.1).

For any given physical situation there may be given initial conditions and further boundary conditions in the horizontal x .

Recall that for waves we seek solutions $\cos(kx - \omega t)$, $\sin(kx - \omega t)$ or $e^{i(kx - \omega t)}$, but here there is cross-sectional structure in ϕ . Hence the velocity potential $\phi = c \cosh[k(z + d)] e^{i(kx - \omega t)}$.

Dispersion relation The dispersion relation is

$$\omega^2 = gk \tanh(kd). \quad (8.1)$$

Then the wave speed $c = \omega/k = \pm\sqrt{g \tanh(kd)/k}$ as plotted in Figure 8.2 as c_p . Waves of different wavelength, $2\pi/k$, travel at different speeds.

Real wave heights Then

$$\eta = A \cos(kx - \omega t), \quad \phi = \frac{Ag}{\omega} \frac{\cosh[k(z + d)]}{\cosh(kd)} \sin(kx - \omega t). \quad (8.2)$$

Shallow water When the wavelength is much bigger than the depth, then $kd \ll 1$ and so the dispersion relation (8.1) becomes $\omega^2 \approx gdk^2$, the wave speed (**phase velocity**) $c = \pm\sqrt{gd}$ is constant—non-dispersive, and

$$\eta = A \cos(kx - \omega t), \quad \phi \approx \frac{Ag}{\omega} \sin(kx - \omega t).$$

So there is effectively no vertical velocity, $w = \phi_z \approx 0$, as assumed by earlier derivations.

Deep water For depth bigger than the wavelength, then $kd \gg 1$ and so the dispersion relation (8.1) becomes $\omega^2 \approx g|k|$, the wave speed (phase velocity) $c = \pm\sqrt{g/|k|}$ varies with wavenumber/wavelength—dispersive, and

$$\eta = A \cos(kx - \omega t), \quad \phi \approx \frac{Ag}{\omega} e^{kz} \sin(kx - \omega t).$$

Due to e^{kz} , the motion decays exponentially quickly below the water surface. Also longer waves (smaller k) travel faster than shorter waves (larger k).

Exercise 8.1 This homework is quite long so summarise concisely. Consider the PDE and BCs for small water waves in water of finite depth d :

$$\begin{aligned}\nabla^2 \phi &= \phi_{xx} + \phi_{zz} = 0 \quad -d < z < 0; \\ \phi_z &= 0 \quad \text{on } z = -d; \\ \eta_t &= \phi_z \text{ and } \phi_t + g\eta = 0 \quad \text{on } z = 0.\end{aligned}$$

We solve these equations approximately and in a different way. Let's find a solution for depths d that are small, but not shallow. In essence we seek a Taylor series approximation to the velocity potential, a Taylor series centred about the bed at $z = -d$:

$$\begin{aligned}\phi(x, z, t) &= \phi_0(x, t) + \phi_1(x, t)(z + d) + \phi_2(x, t)\frac{1}{2}(z + d)^2 \\ &\quad + \phi_3(x, t)\frac{1}{6}(z + d)^3 + \phi_4(x, t)\frac{1}{24}(z + d)^4 + \dots\end{aligned}$$

We end by deriving a wave PDE (one of the so-called Boussinesq equations) and corresponding dispersion relation. (Justification of this approach in general is still the subject of research in the school.)

- (a) Substitute the Taylor series into the PDE and equate coefficients of $(z + d)$. Also substitute into the BC that $\phi_z = 0$ on $z = -d$. Deduce relations between the coefficients ϕ_n .
- (b) Use the BC $\eta_t = \phi_z$ on $z = 0$ to write the surface derivative η_t as a linear combination of $\phi_n(x, t)$. Differentiate with respect to t the BC $\phi_t + g\eta = 0$ on $z = 0$, substitute the series for ϕ_{tt} and η_t , to deduce a PDE involving the ϕ_n for even n .
- (c) Replace ϕ_n for $n \geq 1$ in the above equation by suitable x -derivatives of ϕ_0 and neglect any term with more than four derivatives (in x and t together in total). Seek a wave solution $\phi_0 = a \cos(kx - \omega t)$ to deduce the dispersion relation.



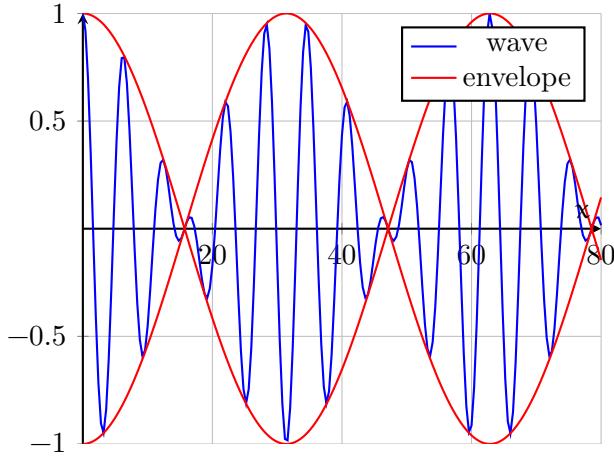


Figure 8.3: a superposition of two waves illustrates how a group of waves propagate.

8.2 Energy travels at the group velocity

Through a stone into a still pond. Fix on any one wave crest generated: you will see it

advance through the group [of waves], gradually dying out as it approaches the front, whilst its former place in the group is occupied in succession by other waves which have come forward from the rear.

H. Lamb, Hydrodynamics

Consider a superposition of two harmonic waves—in any wave system—of equal amplitude as plotted by Figure 8.3:

$$\eta = A \cos(k_1 x - \omega_1 t) + A \cos(k_2 x - \omega_2 t).$$

The figure shows that $\cos(kx - \omega t)$ is a travelling wave shape as before, but it is modulated by the *envelope* $\cos(\delta k x - \delta \omega t)$. The long envelope also appears as a sort of travelling wave. The envelope has wavenumber δk and frequency $\delta \omega$ and so travels with speed, called the **group velocity**,

$$c_g = \frac{\delta \omega}{\delta k} = \frac{\partial \omega}{\partial k}. \quad (8.3)$$

Non-dispersive waves For the basic wave equation $\omega = \pm ck$ so the group velocity $c_g = \partial k / \partial \omega = \pm c$ the wave speed (phase velocity).

Water waves Then

$$c_g = c_p \frac{1}{2} \left[1 + \frac{2kd}{\sinh 2kd} \right]$$

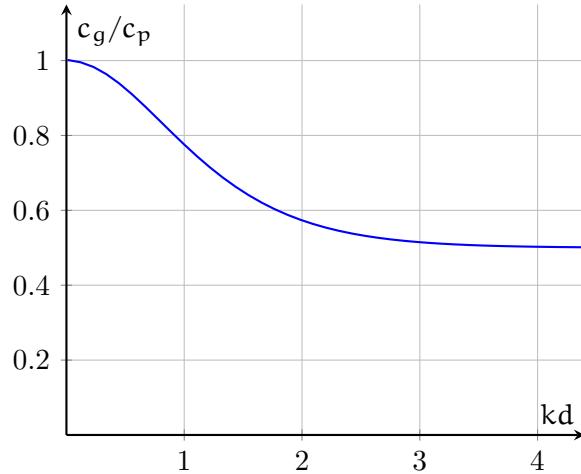


Figure 8.4: ratio of group velocity to phase velocity for water waves at different depths and wavenumbers.

where c_p is the wave/phase speed. Figure 8.4 illustrates the relation that for finite depth waves the group velocity is slower than the wave/phase velocity.

Energy travels at the group velocity The pattern of local energy travels at the same speed as the envelope, $\frac{2\delta\omega}{2\delta k} \approx \partial\omega/\partial k = c_g$.

In general:

- the underlying material that supports waves travels at one velocity (cars at $v(\rho)$, water with zero mean);
- a wave on the material travels at a different velocity (backwards in dense car traffic, and both directions in water waves); and
- the energy in a wave travels at a third velocity (and sometimes not even in the same direction as the waves, see Figure 8.6).

Exercise 8.2 Recall that the energy in a wave propagates at the group velocity, which is usually not the wave speed. From the dispersion relation for a wave system, the group velocity is $c_g = \partial\omega/\partial k$.

- (a) Consider a superposition of two sine waves of equal amplitude $\eta = A \sin(k_1 x - \omega_1 t) + A \sin(k_2 x - \omega_2 t)$. To show the propagation of energy, this superposition is written as what modulated wave?
- (b) What is the group velocity for waves in the Klein–Gordon PDE $u_{tt} + 2u - 3u_{xx} = 0$?
- (c) What is the group velocity for the Korteweg–de Vries PDE $u_t + 6uu_x + u_{xxx} = 0$ for the wave dynamics linearised



Figure 8.5: Why do the waves behind this paddle steamer not spread out? Taken on the River Murray near Echuca.

about the state $u_* = 1/3$?



8.3 Wave propagation in multi-dimensions

Figure 8.3 shows water waves behind a paddle steamer: why do the waves not spread out?

Consider a wave propagating in space of any dimensions with cartesian coordinates $\vec{x} = (x_1, x_2, \dots)$. The basic wave solution is $u = \cos(\vec{k} \cdot \vec{x} - \omega t)$ or $e^{i(\vec{k} \cdot \vec{x} - \omega t)}$ or For example, this form for $u(\vec{x}, t)$ satisfies the 2D wave PDE, $u_{tt} = c^2(u_{x_1 x_1} + u_{x_2 x_2})$, provided $\omega^2 = c^2(k_1^2 + k_2^2)$. In general, $u = \cos(\vec{k} \cdot \vec{x} - \omega t)$ will satisfy the system's PDEs with some **dispersion relation** of the frequency $\omega = \omega(\vec{k})$ for vector wavenumber \vec{k} .

Phase velocity The peak of $\cos(\cdot)$ occurs at $\vec{k} \cdot \vec{x} - \omega t = 0$, and also at $2n\pi$. This identifies the **phase velocity** $\vec{c}_p = \omega \vec{k} / |\vec{k}|^2$.

Group velocity However, the energy travels at the **group velocity** $\vec{c}_g = \nabla_{\vec{k}} \omega = (\omega_{k_1}, \omega_{k_2}, \dots)$ (no derivation).

Classic waves For $\omega^2 = c^2(k_1^2 + k_2^2)$: the frequency $\omega = \pm c \sqrt{k_1^2 + k_2^2} = \pm c |\vec{k}|$; the phase velocity $\vec{c}_p = \omega \vec{k} / |\vec{k}|^2 = \pm c |\vec{k}| \vec{k} / |\vec{k}|^2 = \pm c \vec{k} / |\vec{k}|$ which is in the direction of \vec{k} and of magnitude c ; and the group velocity $\vec{c}_g = \nabla_{\vec{k}} \omega = \pm c (k_1 / \sqrt{k_1^2 + k_2^2}, k_2 / \sqrt{k_1^2 + k_2^2}) = \pm c \vec{k} / |\vec{k}| = \vec{c}_p$.

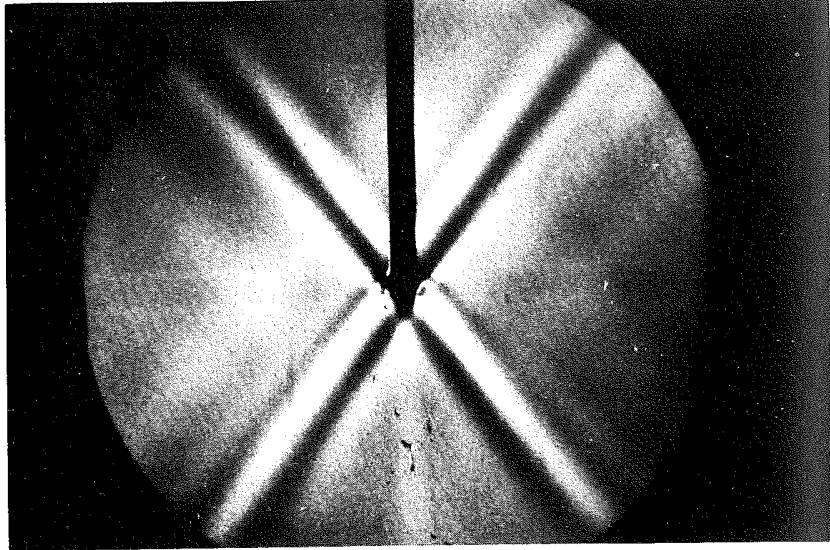


Figure 76. Schlieren picture of waves generated in stratified fluid of uniform Väisälä–Brunt frequency N by oscillation of a horizontal cylinder at frequency $0.7\omega N$. Note that surfaces of constant phase stretch out radially from the source.
[Photograph by D. H. Mowbray.]

Figure 8.6: internal waves in the atmosphere have energy propagating at right angles to the phase propagation (Lighthill 1979, p.314)

Water waves et al. What if the dispersion relation depends only upon the magnitude $|\vec{k}|$ of the wavenumber \vec{k} , and not upon the direction of \vec{k} ? That is, all three directions are the same for such *isotropic waves*.

Numerical schemes are also not isotropic Recall that section 4.1 derives basic spatial discretisation of the heat PDE. I assert that algebra in 2D-space would derive the corresponding dispersion relation for waves on the 2D grid of

$$\omega^2 = \frac{4c^2}{h^2} \left[\sin^2 \frac{k_1 h}{2} + \sin^2 \frac{k_2 h}{2} \right].$$

But what is the numerical group velocity for finite space step h ?

$$\vec{c}_g = \frac{c^2}{\omega h} (\sin(k_1 h), \sin(k_2 h)).$$

This is generally not in the direction of the wavenumber vector \vec{k} .

In numerical schemes, the energy of high wavenumber modes often gets unphysically trapped, and may then accumulate to ruin simulations.

What about the paddle steamer waves? Currents speed or slow waves. Variations in current then bend waves.

Analogously, sound waves travel slower in cold air, and so bent by temperature gradients.

9 Shocking classification of PDEs

Chapter Contents

9.1	Change of variables transforms the PDE	97
9.2	Reduction to the hyperbolic canonical form	99
9.3	Elliptic and parabolic canonical form	100
9.4	Traffic flow and the method of characteristics	101
9.5	Loud uni-directional sound	106

Consider second order partial differential equation in two independent variables for a field $u(x, y)$ (could be two space variables, or (Agarwal & O'Regan 2009, one space and one time, or whatever):
Lecture 28)

$$au_{xx} + bu_{xy} + cu_{yy} = h, \quad (9.1)$$

where the simplest case is when the coefficients are boringly constant, but in general we deal with

- linear PDEs where $h = -du_x - eu_y - fu + g$ and coefficients a, \dots, g are functions of x and y ;
- but also included in scope, the PDE (9.1) could be *quasi-linear* where a, b, c, h are functions of x, y, u, u_x and u_y .

Definition 9.1 (PDE classification). *Depending upon the value of $\Delta = b^2 - 4ac$, the PDE (9.1) is*

- hyperbolic if $\Delta = b^2 - 4ac > 0$ (*wave-like*);
- elliptic if $\Delta = b^2 - 4ac < 0$ (*Laplacian-like*);
- parabolic if $\Delta = b^2 - 4ac = 0$ (*diffusion-like*).

9.1 Change of variables transforms the PDE

Let $u(x, y)$ be a solution of PDE (9.1) and consider a coordinate transformation to new independent variables: $u = U(\xi, \eta)$ where

$$\xi = \xi(x, y), \quad \eta = \eta(x, y). \quad (9.2)$$

Example 9.1 Consider the 1D Wave Equation for $u(x, t)$,

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}.$$

Recall from DEs II that the change of variables $\xi(x, t) = x + ct$ and $\eta(x, t) = x - ct$ and seek $u(x, t) = U(\xi, \eta)$. Show the PDE becomes $\partial^2 U / \partial \xi \partial \eta = 0$ which has a general solution $U(\xi, \eta) = F(\xi) + G(\eta)$. Hence the original wave PDE has d'Alembert's solution $u(x, t) = F(x + ct) + G(x - ct)$. ♣

Now back to the general linear case of the PDE (9.1). Seek solutions $u(x, y) = U(\xi(x, y), \eta(x, y))$. To obtain a PDE for $U(\xi, \eta)$ express partial derivatives u_x, u_y, u_{xx}, u_{xy} and u_{yy} in terms of $U_\xi, U_\eta, U_{\xi\xi}, U_{\xi\eta}$ and $U_{\eta\eta}$. Apply the multivariable chain rule copiously:

$$\begin{aligned} u_x &= U_\xi \xi_x + U_\eta \eta_x; \\ u_y &= U_\xi \xi_y + U_\eta \eta_y; \\ u_{xx} &= (U_\xi \xi_x + U_\eta \eta_x)_x \\ &= U_{\xi x} \xi_x + U_{\eta x} \eta_x + U_{\xi \xi x x} + U_{\eta \eta x x} \\ &= (U_{\xi \xi} \xi_x + U_{\xi \eta} \eta_x) \xi_x + (U_{\eta \xi} \xi_x + U_{\eta \eta} \eta_x) \eta_x \\ &\quad + U_{\xi \xi x x} + U_{\eta \eta x x} \\ &= U_{\xi \xi} \xi_x^2 + 2U_{\xi \eta} \xi_x \eta_x + U_{\eta \eta} \eta_x^2 + U_{\xi \xi x x} + U_{\eta \eta x x}; \\ u_{yy} &= U_{\xi \xi} \xi_y^2 + 2U_{\xi \eta} \xi_y \eta_y + U_{\eta \eta} \eta_y^2 + U_{\xi \xi y y} + U_{\eta \eta y y}; \\ u_{xy} &= \dots \\ &= U_{\xi \xi} \xi_x \xi_y + U_{\xi \eta} (\xi_y \eta_x + \xi_x \eta_y) + U_{\eta \eta} \eta_x \eta_y \\ &\quad + U_{\xi \xi x y} + U_{\eta \eta x y}. \end{aligned}$$

Substitute into PDE (9.1) and regrouping gives

$$AU_{\xi\xi} + BU_{\xi\eta} + CU_{\eta\eta} + DU_\xi + EU_\eta + FU = g, \quad (9.3)$$

where coefficients

$$\begin{aligned} A &= a\xi_x^2 + b\xi_x \xi_y + c\xi_y^2, \\ B &= 2a\xi_x \eta_x + b(\xi_x \eta_y + \xi_y \eta_x) + 2c\xi_y \eta_y, \\ C &= a\eta_x^2 + b\eta_x \eta_y + c\eta_y^2, \\ D &= a\xi_{xx} + b\xi_{xy} + c\xi_{yy} + d\xi_x + e\xi_y, \\ E &= a\eta_{xx} + b\eta_{xy} + c\eta_{yy} + d\eta_x + e\eta_y, \\ F &= f. \end{aligned}$$

The form of PDE (9.3) is also the same for quasi-linear PDEs but the transformational derivatives are much more complicated as they depend upon not only x and y but also u , u_x and u_y .

Exercise 9.1 Define a new coordinate system (ξ, η) by the transform $\xi = \frac{1}{2}(x^2 - y^2)$ and $\eta = xy$ in terms of the cartesian coordinates (x, y) . This exercise aims to transform the PDE $u_{xy} = 0$ into this new coordinate system.

- Consider a function $u(x, y)$ in the new coordinates, say $u = U(\xi, \eta)$. What are the first derivatives?
- What is the mixed derivative u_{xy} ?
- Consider the PDE $u_{xy} = 0$ in the (ξ, η) coordinates system: what domain in (ξ, η) is the PDE hyperbolic?



9.2 Reduction to the hyperbolic canonical form

When the PDE (9.1) is hyperbolic, $\Delta = b^2 - 4ac > 0$, we can find $\xi(x, y)$ and $\eta(x, y)$, such that A and C in the transformed PDE (9.3) become zero and the PDE reduces to the canonical form $U_{\xi\eta} = \dots$. Choosing $\lambda_1(x, y) = (+b + \sqrt{b^2 - 4ac})/(2a)$ and $\lambda_2(x, y) = (+b - \sqrt{b^2 - 4ac})/(2a)$, leads to equations for a useful new coordinate system (note the $+b$ because λ_j term moves to the left-hand side):

$$\xi_x + \lambda_1(x, y)\xi_y = 0, \quad \eta_x + \lambda_2(x, y)\eta_y = 0. \quad (9.4)$$

In coordinates satisfying these equations, the PDE (9.1) takes the canonical form $BU_{\xi\eta} + DU_\xi + EU_\eta + FU = g$ with only one second derivative term.

Coordinate curves—characteristics We need to find $\xi(x, y)$ and $\eta(x, y)$ from (9.4). Consider a *characteristic curve* $y_j(x)$ defined by being a solution of the ODE

$$\frac{dy}{dx} = \lambda_j(x, y).$$

Consider $\xi(x, y)$ on a curve $y = y_1(x)$:

$$\frac{d\xi}{dx} = \frac{\partial \xi}{\partial x} + \frac{\partial \xi}{\partial y} \frac{dy}{dx} = \xi_x + \xi_y \lambda_1 = 0$$

by the PDE (9.4). Thus ξ is constant on a curve $y = y_1(x)$. Similarly, η is constant on a curve $y = y_2(x)$. These are coordinate curves.

The ODES $dy_j/dx = \lambda_j(x, y)$ have general solutions that then fill out the general coordinate curves of the (ξ, η) system. These curves are called *characteristics* of the PDE (9.1).

The nature of the wave equation is fundamental to understanding all hyperbolic PDEs.

Example 9.2 Consider the second order linear PDE $x\partial^2u/\partial x^2 - y\partial^2u/\partial x\partial y + \partial u/\partial x = 0$. Show that this PDE is hyperbolic and find the characteristics. Obtain the transformed PDE using the characteristics as transformed coordinates and hence obtain a general solution.



Exercise 9.2 Consider the PDE $u_{xx} - y^2u_{yy} = 0$. Classify the PDE and find its characteristics.

- Use the discriminant to find the PDE $u_{xx} - y^2u_{yy} = 0$ is hyperbolic in what domain?
- Find two differential equations for the (ξ, η) coordinate system.
- Hence the coordinate curves could be what expressions?



9.3 Elliptic and parabolic canonical form

In the elliptic case $\Delta = b^2 - 4ac < 0$, therefore $\lambda_{1,2} = (b \pm \sqrt{\Delta})/(2a)$ are complex and there are no real characteristics. Without elaborating the details, one can show it is possible to transform an elliptic PDE (9.1) to the canonical form of a generalised Laplace's equation ($B = 0$ and $C = A$):

$$U_{\xi\xi} + U_{\eta\eta} = \frac{H}{A}.$$

Hence the nature of Laplace's equation is fundamental to understanding general elliptic differential equations.

Parabolic PDE If the PDE (9.1) is of a parabolic type, $\Delta = 0$, there is only one family of characteristic curves $dy/dx = \lambda_1(x, y) = -b/(2a)$. In diffusion-like dynamics, these curves are space-like, and a transverse direction is time-like. The canonical form has $B = C = 0$ so the transformed PDE (9.3) is $U_{\xi\xi} = H/A$.

Example 9.3 (mixed type). Determine the type of the PDE, and characteristics if possible,

$$\frac{\partial^2 u}{\partial x^2} - y \frac{\partial^2 u}{\partial y^2} = 0.$$



Conclusion Second order PDES are all deformed versions of the three canonical cases: wave, diffusion, Laplace. The solutions are deformed both by a coordinate transform and by the other terms in the PDE.

9.4 Traffic flow and the method of characteristics

We now turn to the task of finding exact solutions to the traffic flow PDE

$$\frac{\partial \rho}{\partial t} + c(\rho) \frac{\partial \rho}{\partial x} = 0, \quad (9.5)$$

The **method of characteristics** is typically useful for hyperbolic PDES (Roberts 1994, §2.2.4).

Start by imagining that you are considering some definite physical situation for which some definite car density field $\rho(x, t)$ exists. We could then find a curve \mathcal{C} , described by $x = X(t)$ say, for which

$$\frac{dX}{dt} = c[\rho(X, t)].$$

(Imagine solving this as a first-order ODE for the unknown $X(t)$.) The curve \mathcal{C} is called a **characteristic**. Then on \mathcal{C} we can consider ρ to be purely a function of t , namely $\rho = \rho(X(t), t)$. Since $d\rho/dt = 0$ on each such characteristic \mathcal{C} , then the density ρ must be constant.

We make considerably more progress by further deducing the characteristic must here be a straight line. With these observations we now draw the solution of the car traffic problem—equation (9.5) with the initial car density given as $\rho = \rho_0(x)$ at time $t = 0$. The procedure is as follows:

- at any point $x = s$ (and $t = 0$), calculate $c_0(s) = c[\rho_0(s)]$;
- then the straight line through $(x, t) = (s, 0)$ and of slope $c_0(s)$, namely $x = s + c_0(s)t$, is a characteristic \mathcal{C} on which ρ is constant, namely $\rho_0(s)$;
- thus a parametric solution of equation (9.5) is

$$\rho = \rho_0(s) \quad \text{on} \quad x = s + c_0(s)t$$

and so at any time t we vary s and plot $\rho_0(s)$ versus this $x(s)$ to obtain a graph of $\rho(x, t)$.

Example 9.4 Consider a uniform stream of cars except that a group of cars are bunched closer together than the rest. This gives rise to a constant density except at the bunch where there is a region of higher density, as shown in Figure 9.1

Now, a characteristic passes through every point on the x -axis. Drawing a *selection* of the characteristics emanating from the x -axis we would draw a **characteristic diagram** as shown in Figure 9.3.

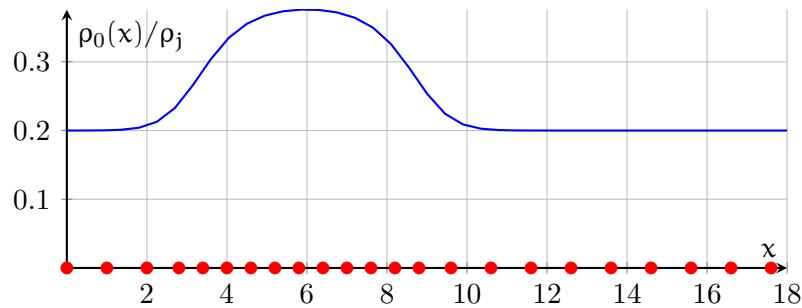


Figure 9.1: A uniform stream of cars (discs) with a localised bunching, and the associated density field $\rho_0(x)$.

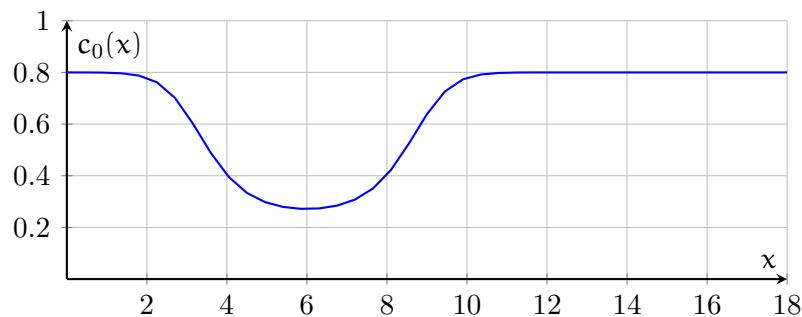


Figure 9.2: A plot of the inverse-slopes of characteristics $c_0(x) = c[\rho_0(x)]$ as a function of x at time $t = 0$ for the initial traffic density shown in Figure 9.1.

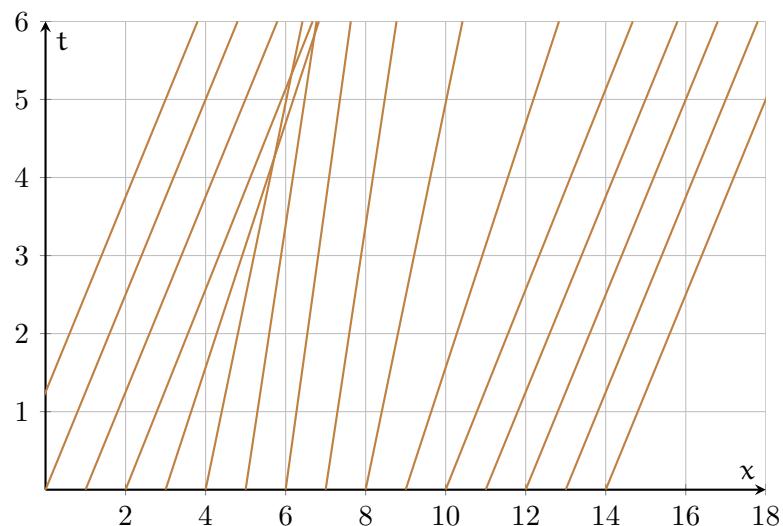


Figure 9.3: A characteristic diagram for a group of cars bunched together in an otherwise uniform stream of cars. Horizontal lines are fixed times at which the density on each characteristic gives the density as a function of x .

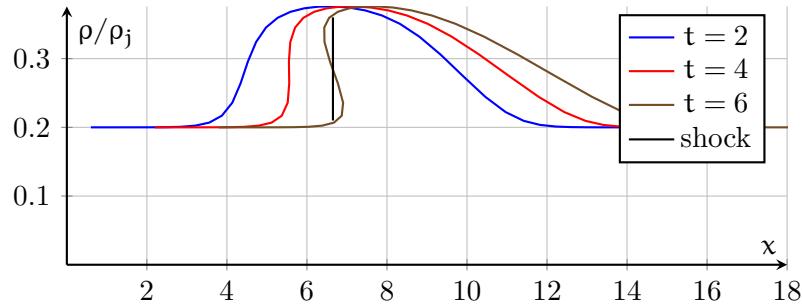


Figure 9.4: The typical evolution of a bump in the density of cars as time progresses: the density steepens at the back, and flattens at the front; until a shock forms at the back.

On each of the characteristics the density is constant. Thus from Figure 9.3 we draw the car density as time progresses; this is shown in Figure 9.4.

Observe that the bump in density, *the density wave*, progresses somewhat as in the linearised solution discussed by Section 1.1. The difference here is that the bump changes shape since the characteristics are not strictly parallel. In particular, see the typical behaviour of car traffic—that the back of the bump steepens and the front of the bump flattens out, see $t = t_1$ in Figure 9.4. Since $V(\rho) > c(\rho)$, individual cars travel quicker than the density bump: a car enters the bunched group from the back and has to brake sharply to slow down to a velocity appropriate to the higher local density; the car can only accelerate slowly through the gently decreasing density at the front of the bump. This is in accord with experience.

As time increases the steepening continues until at some time, $t = t_2$ say, the back of the density bump is just vertical! However, the conservation of cars must still apply, thus we may patch-up the solution by fitting a **shock** in the solution; that is, we allow $\rho(x, t)$ to jump at the correct location in the multi-valued region, as indicated in the $t = t_3$ graph in Figure 9.4.



We shall not pursue shocks any further here although the fitting of shocks can be justifiably done.

Exercise 9.3 Consider car traffic satisfying the PDE $\frac{\partial \rho}{\partial t} + \frac{\partial(u\rho)}{\partial x} = 0$. Suppose the cars travel at speed $u(\rho) = (1 - \rho/150)$ km/min when the density is measured in cars per km. You are given

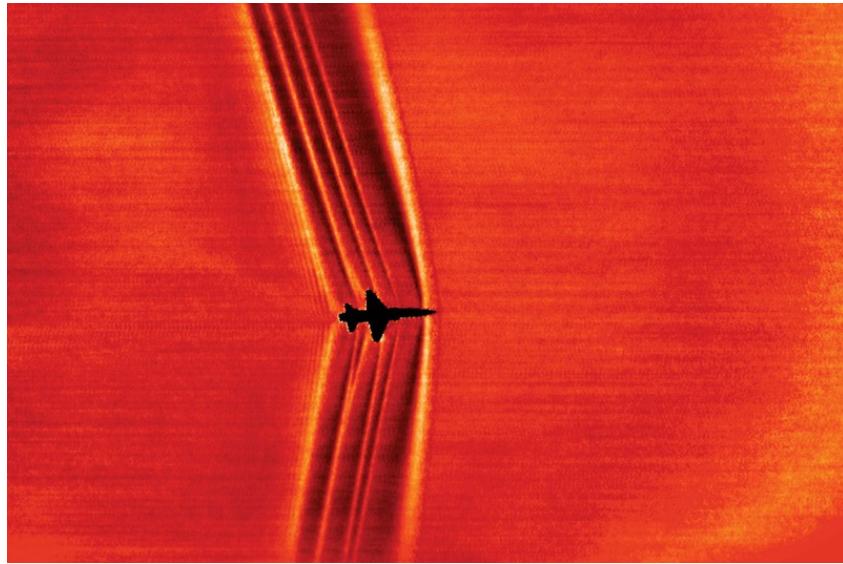


Figure 9.5: a jet's shock waves in air captured against the sun's brightness (courtesy of NASA and the New Scientist, 2015).

the initial densities at various positions are as follows:

x (km)	0	1	2	3	4	5
ρ (cars/km)	100	50	25	50	75	100

From this data, use six characteristic curves to predict car traffic densities at some specified locations and times.

- (a) What is the density in cars per km at $x = 2$ km and $t = 3$ min, or is it indeterminate?
- (b) What is the density in cars per km at $x = 4$ km and $t = 2$ min, or is it indeterminate?
- (c) What is the density in cars per km at $x = 4$ km and $t = 3$ min, or is it indeterminate?



Example 9.5 Traffic lights An interesting situation is when there are a queue of cars waiting at a red traffic light, which then turns green. Figure 9.6 shows the scenario: for $t < 0$ there is a queue of cars waiting to the left of a red traffic light which is situated at $x = 0$; at $t = 0$ the light turns green and the lead car starts to accelerate; for $t > 0$ successive cars in the queue accelerate until they too pass the traffic light. The continuum model of car traffic predicts remarkably well all the details of the evolution of the car traffic in this situation.

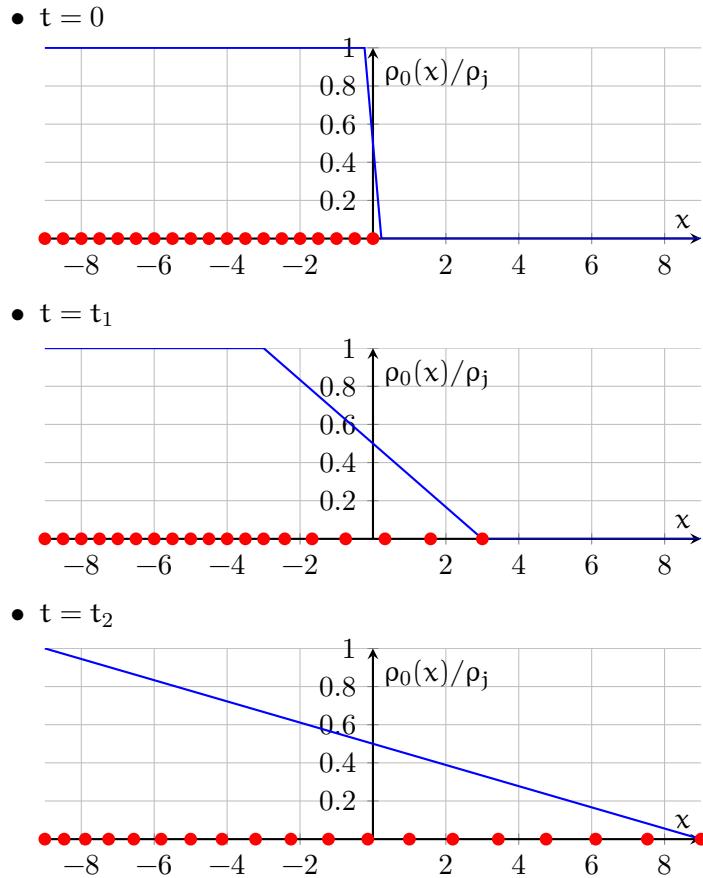


Figure 9.6: Simulation of car traffic at a traffic light which suddenly turns green. Also plotted is the resultant density field.

The appropriate initial condition on the car density is

$$\rho(x, 0) = \rho_0(x) = \begin{cases} 0 & \text{if } x > 0, \\ \rho_j & \text{if } x < 0, \\ \text{all values in between} & \text{if } x = 0. \end{cases}$$

The typical characteristic diagram for this situation is illustrated in Figure 9.7. There are three types of characteristic lines: the lines emanating from the region of no cars in front of the traffic light, described by $x = s + c(0)t$ for $s > 0$, carry a density of $\rho = 0$ with them; the lines emanating from the region of bumper-to-bumper cars behind the traffic light, described by $x = s + c(\rho_j)t$ for $s < 0$, carry a density of $\rho = \rho_j$ with them ($c(\rho_j) < 0$ and so these characteristics slope backwards); and the characteristics emanating from the traffic light are of all slopes in between, described by $x = c(\rho)t$ for $0 < \rho < \rho_j$, and carry their particular value of the density with them (depending upon their slope). These characteristics are said to form an **expansion fan**.

To find the density at any particular time, say $t = t_1$, after the light turns green we just look at the intersections of the char-

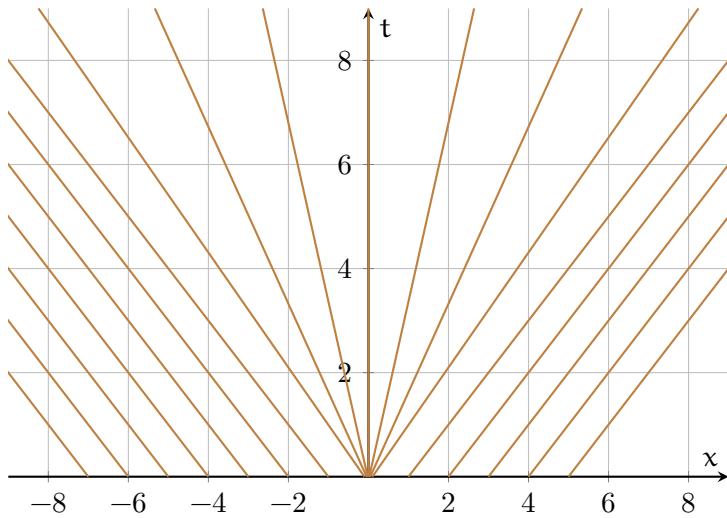


Figure 9.7: A typical characteristic diagram for car traffic which has been waiting at a red traffic light after it turns green.

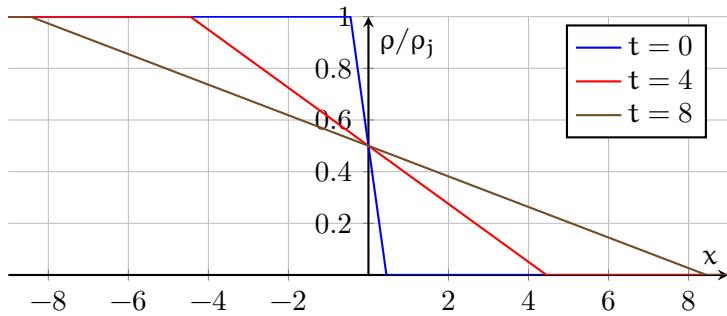


Figure 9.8: The evolution of the density of cars after a red traffic light turns green—as predicted by the method of characteristics.

acteristics and the the line $t = t_1$ and plot the density of each of the characteristics as a function of the x -location which the intersection occurs. Figure 9.8 shows example plots for the density. There is a remarkably good agreement between these predictions and the numerical simulations. Furthermore, these both agree well with our experience of such traffic queues. ♣

9.5 Loud uni-directional sound

We ‘specialise’ the governing equations of gas dynamics so that they describe sound which propagates in only one direction. In such a case the governing equations simplify and they may be solved by the simple method of characteristics developed for car traffic flow (Roberts 1994, §3.2.3).

Assume that body forces (gravity) are negligible: from section 1.3

the governing PDEs are

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = 0, \quad (9.6)$$

$$\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} \right) = -k^2 \rho^{\gamma-1} \frac{\partial \rho}{\partial x}. \quad (9.7)$$

In principle: convert to one 2nd order PDE; find characteristic curves (§9.1); transform the PDE (§9.2); solve in simple example cases. Instead we take a shortcut direct to solving the simple cases.

The shortcut, the simplification, is to consider only those solutions of the PDEs (9.6)–(9.7) for which

$$\rho = R(u),$$

where $R(u)$ is some specific but as yet unknown function. We derive that this restriction is consistent with the original two PDEs only for some specific choice for $R(u)$.

For simplicity take the gas constant $\gamma = 1$. For other values of γ the details to follow will be different, but the qualitative results are the same.

Thus, taking

$$\rho = \rho_* \exp(u/k), \quad (9.8)$$

where $\rho_* = e^A$ is an arbitrary constant (ρ_* happens to be the density whenever the gas is stationary) then both the continuity PDE and the momentum PDE reduce to the same PDE. (The $\exp(-u/k)$ case is similar but with sign changes.)

From the continuity PDE

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = -\frac{\rho_* \exp(u/k)}{\rho_* \frac{1}{k} \exp(u/k)} \frac{\partial u}{\partial x} = -k \frac{\partial u}{\partial x},$$

which may be written

$$\frac{\partial u}{\partial t} + (k + u) \frac{\partial u}{\partial x} = 0. \quad (9.9)$$

This is exactly the same PDE as was solved for car traffic, namely PDE (9.5), but here the coefficient function is the increasing linear function $c(u) = k + u$. Thus we solve the PDE in exactly the same manner, via the method of characteristics, to find a class of exact solutions of the ideal gas PDEs (9.6)–(9.7).

As before imagine $u(x, t)$ is given; consider characteristic curves $dX/dt = k + u(x(t), t)$; on each $d/dtu(X(t), t) = u_t + u_x t X = u_t + (k + u)u_x = 0$ by the PDE; so u is constant on each; so $dX/dt = k + u = \text{constant}$ so each is a straight line. For example, draw a diagram of the solution following a puff of air started along a tube.

Recall the discussion of this sort of equation modelling car traffic. The nonlinear term in the PDE $u\partial u/\partial x$ induces ‘shocks’ to form eventually in most situations; these shocks are more commonly known as **sonic booms**.

This method of characteristics applies to other PDEs. Further, it can be developed to the important case of two sets of characteristics crossing and interacting (Roberts 1994, §3.3, e.g.).