ME2 Maths - Term 2 Schedule (2024/2025)

- Chapter 5: Numerical methods for solving PDEs (~ 7 lectures) – Slide 366
- Chapter 6: Analytical solutions for solving PDEs (~ 6 lectures) – Slide 490
- Chapter 7: Root finding and optimisation methods (~ 4 lectures) – Slide 551

- Recommended reading:

Numerical methods for engineers, by Chapra & Canale, McGraw Hill Education

Advanced engineering mathematics, by Kreyszig, John Wiley and Sons (e-book from library)

So many resources online… find one that works for you for the particular topic!

Chapter 5: Numerical methods for solving PDEs

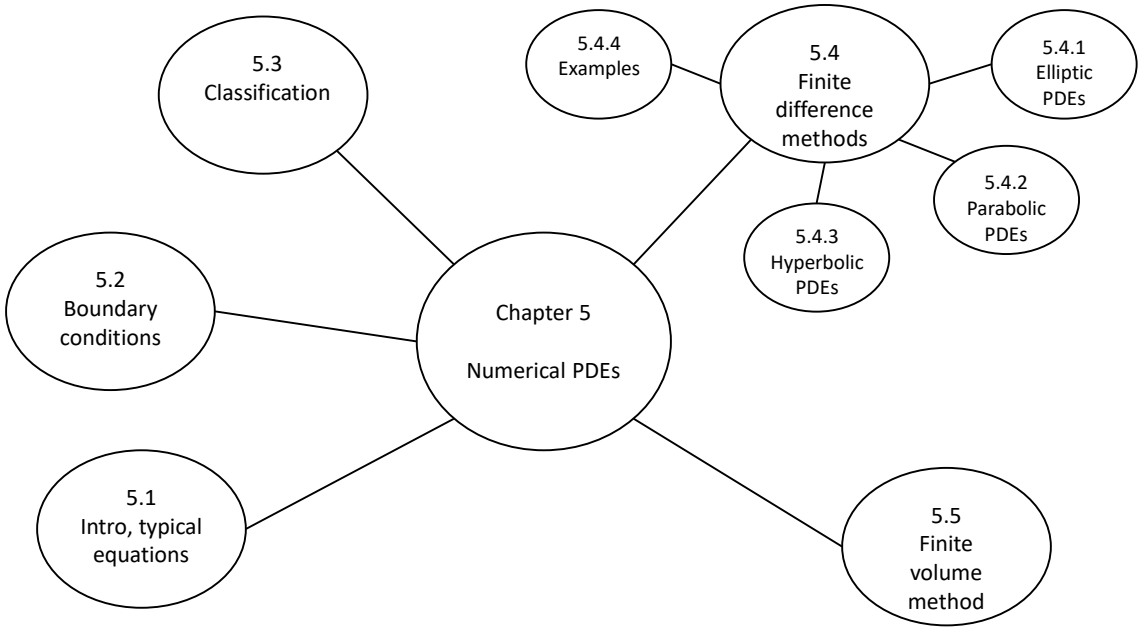
Dr Monica Marinescu

Learning outcomes

You will be able to:

- Identify, classify and derive PDEs and their boundary conditions
- Choose and apply finite difference and finite volume methods to
  - Express PDEs in discretised form
  - Solve PDEs with various types of boundary conditions, via the explicit and implicit methods described
- Explain and use the conditions for stability and convergence

367



368

Contents

369

5.1 Partial Differential Equations

- What are 2nd order PDEs
- PDEs with relevance in science and engineering:
  - Laplace's equation
  - Poisson equation
  - Diffusion (or heat) equation
  - Wave equation
- General classification of PDEs
- Boundary/initial conditions

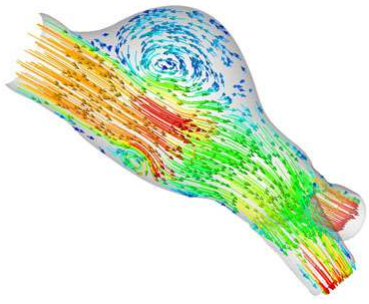
370

What are 2nd order PDEs

- So far you have worked with ODEs – equations involving ordinary derivatives (i.e. of functions of one variable)
- A PDE is an equation that involves an unknown function and its *partial* derivatives (i.e. function of 2 or more variables)
- A PDE is solved for the unknown function of two or more independent variables, over a region in space, subject to boundary and initial conditions
- In Chapter 6, you will learn how to solve some PDEs analytically, using separation of variables – this is only possible in very simple scenarios
  - Even minor increases in complexity of a PDE make this approach impossible
- For most relevant problems, we are forced to use numerical methods

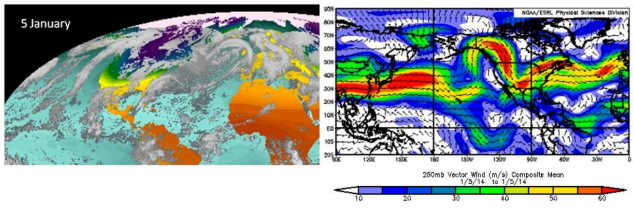
PDE applications

- Turbulence: Navier-Stokes equ



http://mriquestions.com/laminar-v-turbulent.html

- Weather forecasting

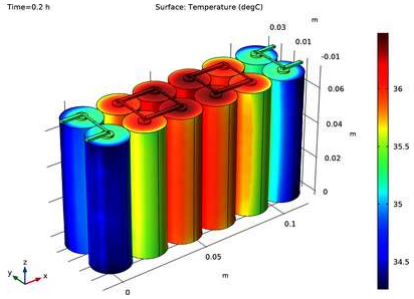


- Opinion forming and behaviour of people – Boltzmann-like equ

https://www.carbonbrief.org/met-office-report-spells-out-climate-change-link-to-uk-storms-and-flooding

PDE applications

- Electric powertrains: electrical – chemical- thermal coupling in batteries



https://www.comsol.com/model/thermal-distribution-in-a-pack-of-cylindrical-batteries-76291

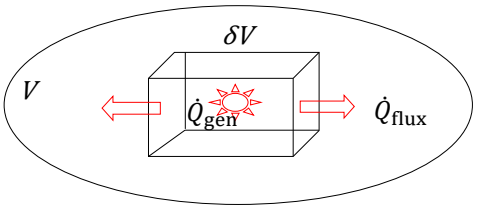
- Traffic flow: reduce congestion and emission

https://www.ptvgroup.com/en/solutions/products/ptv-vissim/areas-of-application/traffic-flow-simulation/

2nd Order PDEs

- Many important engineering problems can be expressed in terms of 2nd order partial differential equations (PDEs)
- In the next sections we will derive some of the most common types of PDEs
  - Diffusion equation
  - Laplace and Poisson equations
  - Wave equation

Diffusion (or heat) equation



GOAL: Obtain an equation for the temperature $T(\mathbf{r}, t)$

- Consider heat conduction in a region V
- Now consider any small region δV of V
- We can now write down the following laws:

$$-\dot{Q}_{\text{flux}}(t) + \dot{Q}_{\text{gen}}(t) = \frac{\mathrm{d}}{\mathrm{d}t} Q_{\text{heat}}(t)$$

Conservation of Energy

+

$$\mathbf{q}(\mathbf{r}, t) = -\kappa \nabla T$$

Fourier's law of heat conduction

- $\mathbf{q}$ is the local heat flux density
- κ is the material's conductivity

375

Diffusion (or heat) equation

$$-\dot{Q}_{\text{flux}}(t) + \dot{Q}_{\text{gen}}(t) = \frac{\mathrm{d}}{\mathrm{d}t} Q_{\text{heat}}(t)$$

Where:

$\dot{Q}_{\text{flux}}(t) = \oint_S \mathbf{q}(\mathbf{r}, t) \cdot \mathbf{n} dS$ Rate of Heat flow out of the region; S = contour of δV

$\dot{Q}_{\text{gen}}(t) = \int_{\delta V} \rho s dV$ Rate of production of heat; s = specific heat generation rate

$Q_{\text{heat}}(t) = \int_{\delta V} \rho c_V T(\mathbf{r}, t) dV$ Heat contained in region; c_V = specific heat capacity

376

Diffusion (or heat) equation

- Substituting these in gives: $-\oint_S \mathbf{q}(\mathbf{r},t)\cdot\mathbf{n}dS + \int_{\delta V}\rho s dV = \int_{\delta V}\rho c_V\frac{\partial T}{\partial t}(\mathbf{r},t)dV$
- Applying the divergence theorem to the 1st term:

$$-\int_{\delta V}\nabla\cdot\mathbf{q}(\mathbf{r},t)dV + \int_{\delta V}\rho s dV = \int_{\delta V}\rho c_V\frac{\partial T}{\partial t}(\mathbf{r},t)dV$$

- Substituting Fourier's law:

$$\int_{\delta V}\nabla\cdot\big(\kappa\nabla T(\mathbf{r},t)\big)dV + \int_{\delta V}\rho s dV = \int_{\delta V}\rho c_V\frac{\partial T}{\partial t}(\mathbf{r},t)dV$$

- Collecting all terms together under one integral:

$$\int_{\delta V}\left\{\nabla\cdot(\kappa\nabla T) + \rho s - \rho c_V\frac{\partial T}{\partial t}(\mathbf{r},t)\right\}dV = 0$$

377

Diffusion (or heat) equation

- So, for this 'control volume' we have:

$$\int_{\delta V}\left\{\nabla\cdot(\kappa\nabla T) + \rho s - \rho c_V\frac{\partial T}{\partial t}\right\}dV = 0$$

- Remember, this control volume could be any part of V we choose (including all of it)
- Since this integral is zero for any δV, then it must be true that the **integrand** (the thing being integrated) is also zero, so we get

$$\boxed{\nabla\cdot(\kappa\nabla T) - \rho c_V\frac{\partial T}{\partial t} = -\rho s}$$ For every point in *V*

- This is the **diffusion equation** (in the context of heat conduction - as we have considered here, it is also called the **heat equation**)

378

Diffusion (or heat) equation

$$\nabla \cdot (\kappa \nabla T) - \rho c_V \frac{\partial T}{\partial t} = -\rho s$$

- This is (possibly) its most general form
- It is frequently seen in slightly simpler forms depending on the physics of the problem:
- If the material is '**linear**' then k is independent of position, and we have:

$$\kappa \nabla \cdot (\nabla T) - \rho c_V \frac{\partial T}{\partial t} = -\rho s$$

Or: $$\kappa \nabla^2 T - \rho c_V \frac{\partial T}{\partial t} = -\rho s$$

- Recall that ∇^2 is the 'Laplacian' and in three dimensions we can write :

$$\kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) - \rho c_V \frac{\partial T}{\partial t} = -\rho s$$

379

Diffusion (or heat) equation

$$\nabla \cdot (\kappa \nabla T) - \rho c_V \frac{\partial T}{\partial t} = -\rho s$$

- This is (possibly) its most general form
- It is frequently seen in slightly simpler forms depending on the physics of the problem:
- If (as well as being linear), there are no **volumetric heat sources** then

$$s = 0 \Rightarrow \kappa \nabla^2 T - \rho c_V \frac{\partial T}{\partial t} = 0$$

- Common examples of volumetric heat sources are:
  - Ohmic heating
  - Microwave heating
  - Nuclear fission
- Note - these are **internal** heat sources, not supplied from external sources through the boundary of the domain (e.g. heating an object with a flame, etc…)

380

Diffusion (or heat) equation

- This discussion has been about heat, but the diffusion equation applies to many more physical situations, all satisfying an equation of the form

$$\nabla^2\varphi - k\frac{\partial\varphi}{\partial t} = f$$

- Diffusion of chemical species
- Population dynamics
- Neutron flux in nuclear reactors
- Pricing/derivatives in financial markets

381

Poisson equation

- Consider, again, the heat equation, but now in the steady state, hence

Diffusion (heat) eqn: $$\kappa\nabla^2 T - \rho c_V\frac{\partial T}{\partial t} = -\rho s$$

But: $$\frac{\partial T}{\partial t} = 0 \Rightarrow \kappa\nabla^2 T = -\rho s$$

- This is an example of a **Poisson equation**, which has the general form

$$\nabla^2\varphi = f$$

- As well as steady state heat conduction, it arises in
  - Electrostatics
  - Astronomy
  - Stress analysis

382

Laplace equation

https://steamdaily.com/revealing-the-physics-behind-the-soap-bubble-burst/

- In the case of steady state heat transfer with no volumetric sources:
  - Steady state: $\frac{\partial T}{\partial t} = 0$
  - No sources: $s = 0$

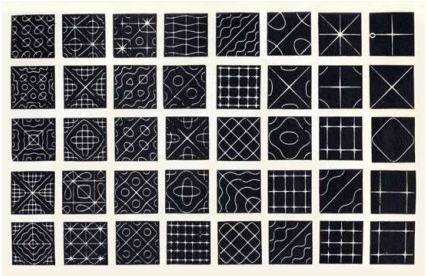
  We then have: $\nabla^2 T = 0$
- This is an example of a **Laplace equation**, which has the general form

$$\nabla^2 \varphi = 0$$

- As well as steady state source-free heat conduction, it arises in
  - Electrostatics
  - Inviscid fluid flow

The Wave Equation

- Thus far, we have studied the Laplace, Poisson & diffusion equations
- The final equation we study occurs in many areas of engineering: the **wave equation**
- This equation arises in:
  - Acoustics
  - Electromagnetics
  - Small amplitude free surface flows
  - Vibrations
- It has the general form: $\nabla^2 \varphi - \frac{1}{c^2}\frac{\partial^2 \varphi}{\partial t^2} = 0$ c is the wave speed

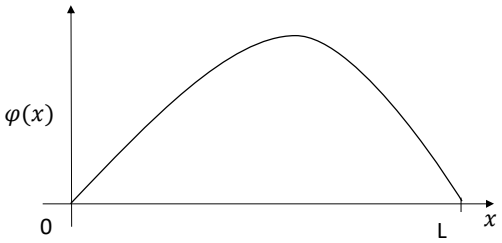


https://touringinstability.wordpress.com/2013/08/08/ernst-chladni-visualizing-the-natural-geometric-patterns-of-sound/
Check out the video

The Wave Equation (1D) - derivation

GOAL: obtain an equation for the vertical displacement $\varphi(x,t)$ along a string

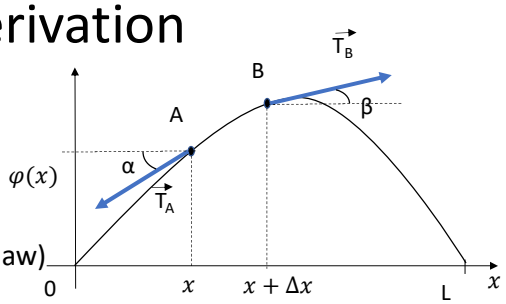
- Describe the 1D motion of the string



385

The Wave Equation (1D) - derivation

- Write down the balance of forces on AB, a small piece of string
- Assume: no resistance to bending (longitudinal tension only)



1. No motion in the horizontal direction (Newton's 1st law)

$$T_A \cos\alpha = T_B \cos\beta = T = \text{const} \quad (1)$$

2. The resultant force in the vertical direction obeys Newton's second law

$$T_B \sin\beta - T_A \sin\alpha = \rho\Delta x \frac{\partial^2\varphi}{\partial t^2}$$

$\rho\Delta x$ is the mass of the section AB, ρ is the mass per unit length

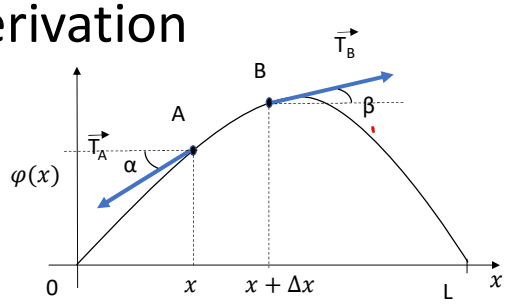
- Divide 1st term by $T_B \cos\beta$, second term by $T_A \cos\alpha$, and third by T, and use Eq. 1:

$$\tan\beta - \tan\alpha = \frac{\rho\Delta x}{T}\frac{\partial^2\varphi}{\partial t^2}$$

386

The Wave Equation (1D) - derivation

$$\tan\beta - \tan\alpha = \frac{\rho\Delta x}{T}\frac{\partial^2\varphi}{\partial t^2}$$



- But $\tan\alpha = \left(\frac{\partial\varphi}{\partial x}\right)\Big|_x$ and $\tan\beta = \left(\frac{\partial\varphi}{\partial x}\right)\Big|_{x+\Delta x}$

$$\left[\left(\frac{\partial\varphi}{\partial x}\right)\Big|_{x+\Delta x} - \left(\frac{\partial\varphi}{\partial x}\right)\Big|_x\right]\frac{1}{\Delta x} = \frac{\rho}{T}\frac{\partial^2\varphi}{\partial t^2} \qquad (2)$$

- For very small AB pieces: $\Delta x \to 0$, LHS = definition of derivative $\partial f/\partial x$, with $f(x) = \partial\varphi/\partial x$
- 1D wave equation: $\frac{\partial^2\varphi}{\partial x^2} - \frac{1}{c^2}\frac{\partial^2\varphi}{\partial t^2} = 0$ where $\frac{1}{c^2} = \frac{\rho}{T}$
- Note: while the diagram shows the ends of the cord at zero, this derivation does not use their position, so it is valid for any string position

387

Summary: Partial Differential Equations

Linear 2nd order PDEs are important sets of equations that are used to model many engineering problems. You have encountered:

- Laplace equation: $\nabla^2\varphi = 0$
  - 2D Laplace equation: $\frac{\partial^2\varphi(x,y)}{\partial x^2} + \frac{\partial^2\varphi(x,y)}{\partial y^2} = 0$
- Poisson equation: $\nabla^2\varphi = f(x,y)$

Steady state problems

- Diffusion equation (source-free): $\nabla^2\varphi - k\frac{\partial\varphi}{\partial t} = 0$
  - 1D diffusion equation: $\frac{\partial^2\varphi(x,t)}{\partial x^2} - k\frac{\partial\varphi(x,t)}{\partial t} = 0$
- Wave equation: $\nabla^2\varphi - \frac{1}{c^2}\frac{\partial^2\varphi}{\partial t^2} = 0$
  - 1D wave equation: $\frac{\partial^2\varphi(x,t)}{\partial x^2} - \frac{1}{c^2}\frac{\partial^2\varphi(x,t)}{\partial t^2} = 0$

Transient (time propagation) problems

388

5.2 Boundary conditions

- We have derived four types of PDEs:
  - Laplace equation $\nabla^2\varphi = 0$
  - Poisson equation $\nabla^2\varphi = f$
  - Diffusion equation $\nabla^2\varphi - k\frac{\partial\varphi}{\partial t} = f$
  - Wave equation $\nabla^2\varphi - \frac{1}{c^2}\frac{\partial^2\varphi}{\partial t^2} = 0$
- Depending on the situation, then, we must solve these equations in order to obtain our 'unknown' φ
- In fact this is not possible (yet) - these problems are not **properly posed**
- What we need are **boundary/initial conditions**

389

Boundary conditions

- In order to properly pose a problem, so it can (in principle) be solved, as well as determining regions and their governing equations, we need to specify appropriate boundary conditions
- Only when boundary conditions are specified can we solve these PDEs
  - Even then it is not easy - especially for realistic geometries
- We often call a PDE together with appropriate boundary conditions a **boundary value problem** (BVP)

- The skill of the engineer is in determining what these boundary conditions should be
- There are a number of types of boundary condition, depending on the type of equation

390

$\nabla^2\varphi = 0$

Boundary conditions: Laplace equation

- For the Laplace equation, there are two basic types of boundary conditions:

'Dirichlet' condition: $\varphi = \varphi_D$ **or** 'Neumann' condition: $\frac{\partial\varphi}{\partial n} \equiv \mathbf{n}\cdot\nabla\varphi = \varphi'_N$ for $\mathbf{r} \in \partial V$

- One **or** other (but **never** both) must be specified at **every** point of the boundary
- For a steady state source-free heat conduction problem, these conditions correspond to specifying
  - Temperature (Dirichlet)
  - Heat flux (Neumann)
- There is a third type of condition, which is a combination of these two, called the Robin condition. It is a linear combination of Dirichlet and Neumann conditions:

$$a\varphi + b\frac{\partial\varphi}{\partial n} = c$$

An example is convective heat transfer: $-\kappa\frac{\partial T}{\partial n} = h(T - T_{\text{amb}})$

391

$\nabla^2\varphi = f$

Boundary conditions: Poisson equation

- The boundary conditions required for the Poisson equation are identical to those used in the Laplace case.
  - Either Dirichlet or Neumann at each point of the boundary, (or more generally a Robin condition)

392

Boundary/initial conditions: Diffusion equation

$$\nabla^2\varphi - k\frac{\partial\varphi}{\partial t} = f$$

- The boundary conditions are the same as those required for the Laplace and Poisson equation:

'Dirichlet' condition: 'Neumann' condition:

$$\varphi = \varphi_D \quad \textbf{or} \quad \frac{\partial\varphi}{\partial n} \equiv \mathbf{n}\cdot\nabla\varphi = \varphi'_N \quad \text{for } \mathbf{r} \in \partial V$$

- Unlike the Laplace & Poisson equations, the diffusion equation is time-dependent
=> For the diffusion equation, we also need an initial condition, which is the value of φ at $t = 0$ throughout the domain:

$$\varphi(\mathbf{r}, t = 0) = \varphi_0 \text{ for } \mathbf{r} \in V$$

393

Boundary/initial conditions: Wave equation

$$\nabla^2\varphi - \frac{1}{c^2}\frac{\partial^2\varphi}{\partial t^2} = 0$$

- The boundary conditions are the same as those required for the Laplace and Poisson equation:

'Dirichlet' condition: 'Neumann' condition:

$$\varphi = \varphi_D \quad \textbf{or} \quad \frac{\partial\varphi}{\partial n} \equiv \mathbf{n}\cdot\nabla\varphi = \varphi'_N \quad \text{for } \mathbf{r} \in S, S = \text{boundary of } V$$

- Like the diffusion equation, the wave equation is time-dependent; BUT 2nd order derivative with time

$\Rightarrow$ For the wave equation, we need TWO initial conditions:

- the value of φ at $t = 0$ throughout the domain: $\varphi(\mathbf{r}, t = 0) = \varphi_0$ for $\mathbf{r} \in V$
- the value of $\frac{\partial\varphi}{\partial t}$ at $t = 0$ throughout the domain: $\frac{\partial\varphi}{\partial t}(\mathbf{r}, t = 0) = \varphi'_0$ for $\mathbf{r} \in V$

394

5.3 Classification of PDEs

$$u_{xx} \equiv \frac{\partial^2 u}{\partial x^2}, u_x \equiv \frac{\partial u}{\partial x}, u_{xy} \equiv \frac{\partial^2 u}{\partial x \partial y}$$

x, y = any 2 independent variables

$$Au_{xx} + 2Bu_{xy} + Cu_{yy} = f(x, y, u, u_x, u_y)$$

- Semilinear 2nd order PDE (linear in the highest derivative) can be classified in three categories

TYPE	Condition	Example (*fill in)
Elliptic	$B^2 - 4AC < 0$	
Parabolic	$B^2 - 4AC = 0$	
Hyperbolic	$B^2 - 4AC > 0$	

$$\nabla^2 \varphi = 0$$

$$\nabla^2 \varphi = f(x, y)$$

$$\nabla^2 \varphi - k\frac{\partial \varphi}{\partial t} = 0$$

$$\nabla^2 \varphi - \frac{1}{c^2}\frac{\partial^2 \varphi}{\partial t^2} = 0$$

- This classification remains the same for PDEs of more than two independent variables (e.g. two-dimensional wave equation)

395

Classification of PDEs - workspace

- $\nabla^2 \varphi = 0$ (dimensions x, y):
- $\nabla^2 \varphi = f(x, y)$ (dimensions x, y):
- $\nabla^2 \varphi - k\frac{\partial \varphi}{\partial t} = 0$ (dimensions x, t):
- $\nabla^2 \varphi - \frac{1}{c^2}\frac{\partial^2 \varphi}{\partial t^2} = 0$ (dimensions x, t):

396

Classification of PDEs

- Classification is important because:
  - Each PDE type has a different mathematical and physical behaviour
  - Different methods are used to solve these different categories
- Elliptic equations are BVPs for steady state problems
- Parabolic and hyperbolic equations describe time-dependent, transient problems; they are 'IVP' in time, BVP in space
- Hyperbolic equations allow non-smooth solutions

Slide content not on the exam

Classification of PDEs

$$Au_{xx} + Bu_{xy} + Cu_{yy} = f(x, y, u, u_x, u_y)$$

TYPE	Condition	Example
Elliptic	$B^2 - 4AC < 0$	Laplace & Poisson equation
Parabolic	$B^2 - 4AC = 0$	Heat equation
Hyperbolic	$B^2 - 4AC > 0$	Wave equation

- The terminology (elliptic, parabolic and hyperbolic) reflects the analogy between the form of the discriminant, $B^2 - 4AC$, for PDEs and the form of the discriminant which classifies conic sections.
- Conic sections are described by the general second-order algebraic equation.

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

Ellipse	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$	$B^2 - 4AC < 0$
Parabola	$y^2 = 4ax$ with $a > 0$	$B^2 - 4AC = 0$
Hyperbola	$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$	$B^2 - 4AC > 0$

5.4 Finite difference methods for PDEs

- You can approximate derivative terms in PDEs as difference quotients (as you did for ODEs)
  - Discretise the simulation domain into chunks; the properties of the system are stored at discrete nodes
  - Use a Taylor series expansion to describe the value of the function at each node in relation to that at the other nodes
  - Use a truncated series to reduce the number of operations, e.g. keep only nearest neighbour node
  - These lead to the forward Euler, backward Euler and central difference expressions (1st order derivatives), and resulting expressions for higher order derivatives

399

Finite difference method for PDEs (e.g. 2 variables)

$$Au_{xx} + Bu_{xy} + Cu_{yy} = f(x, y, u, u_x, u_y)$$

1. Discretise the simulation domain

$$x_i = (i-1)h, \qquad i = 1, \dots N$$
$$y_j = (j-1)k, \qquad j = 1, \dots M$$

2. Replace partial derivatives by difference quotients (central difference)

$$u_x(x,y) \approx \frac{1}{2h}[u(x+h,y) - u(x-h,y)]$$
$$u_y(x,y) \approx \frac{1}{2k}[u(x,y+k) - u(x,y-k)]$$
$$u_{xx}(x,y) \approx \frac{1}{h^2}[u(x+h,y) - 2u(x,y) + u(x-h,y)]$$
$$u_{yy}(x,y) \approx \frac{1}{k^2}[u(x,y+k) - 2u(x,y) + u(x,y-k)]$$

(you will not need u_{xy})

400

Finite difference method for PDEs

$$Au_{xx} + Bu_{xy} + Cu_{yy} = f(x, y, u, u_x, u_y)$$

3. Substitute derivatives in the PDE with difference quotients

$\Rightarrow$ The PDE is approximated by a difference equation

$\Rightarrow$ At mesh point (x_i, y_j), $u(x_i, y_j)$ is described as a function of u at the neighbouring points, some of them related to the boundary conditions

4. Solve the resulting system of algebraic equations to find $u(x_i, y_j)$ at all mesh points in the domain

- For a small number of internal points (< 100), can use a direct solution method (Gauss elimination, matrix inversion by LU factorisation)
- For a large number of internal points, use an iteration method to avoid storage problems (Gauss-Seidel method aka Liebmann's method) – more about these next

401

5.4.1 Elliptic PDEs

Example: Laplace equation, Poisson equation

402

Finite difference method for Laplace equation

Solve for $u(x,y)$ where

$$\nabla^2 u = u_{xx} + u_{yy} = 0$$

1. Difference quotients corresponding to relevant partial derivatives

$$u_{xx}(x,y) \approx \frac{1}{h^2}[u(x+h,y) - 2u(x,y) + u(x-h,y)]$$

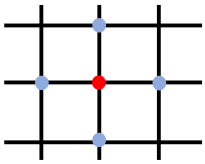
$$u_{yy}(x,y) \approx \frac{1}{k^2}[u(x,y+k) - 2u(x,y) + u(x,y-k)]$$

2. You usually can assume a square grid, so *h=k*, *h* = mesh size (expressions become simpler). Substituting in the PDE:

$$u(x+h,y) + u(x,y+h) + u(x-h,y) + u(x,y-h) - 4u(x,y) = 0$$

403

Finite difference Laplace



$$u(x+h,y) + u(x,y+h) + u(x-h,y) + u(x,y-h) - 4u(x,y) = 0$$
$$u(x_{i+1},y_j) + u(x_i,y_{j+1}) + u(x_{i-1},y_j) + u(x_i,y_{j-1}) - 4u(x_i,y_j) = 0$$

- This expression holds for all interior points
- Interpretation: $u(x,y)$ at a given grid point is the mean of the values of u at the four neighbouring points

$$\frac{1}{4}[u(x_{i+1},y_j) + u(x_i,y_{j+1}) + u(x_{i-1},y_j) + u(x_i,y_{j-1})] = u(x_i,y_j)$$

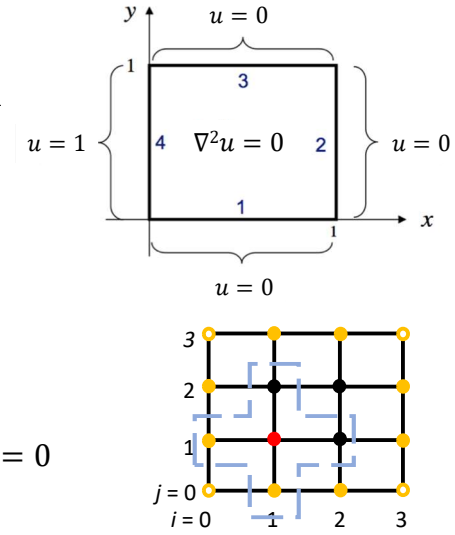
- This is a 5-point approximation with a coefficient scheme of the form:

$$\begin{Bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{Bmatrix} u = 0$$

- A unique solution can be found for given boundary conditions

404

WE42: Finite difference Laplace with Dirichlet boundary conditions



- $u_{xx} + u_{yy} = 0, u = 0$ on all sides except $u(0, y) = 1$

Solution:

Apply the stencil around the four internal points:

- For $u_{1,1}$: $u_{2,1} + u_{1,2} + u_{0,1} + u_{1,0} - 4u_{1,1} = 0$
- For $u_{2,1}$: $u_{3,1} + u_{2,2} + u_{1,1} + u_{2,0} - 4u_{2,1} = 0$
- For $u_{2,2}$: $u_{3,2} + u_{2,3} + u_{1,2} + u_{2,1} - 4u_{2,2} = 0$
- For $u_{1,2}$: $u_{2,2} + u_{1,3} + u_{0,2} + u_{1,1} - 4u_{1,2} = 0$

Evaluate known values from boundary conditions:

$u_{1,0} = u_{2,0} = 0 \qquad u_{3,1} = u_{3,2} = 0 \qquad u_{1,3} = u_{2,3} = 0$

$u_{0,1} = u_{0,2} = 1$

405

WE42: Finite difference Laplace with Dirichlet boundary conditions

$$\begin{pmatrix} -4 & 1 & 0 & 1 \\ 1 & -4 & 1 & 0 \\ 0 & 1 & -4 & 1 \\ 1 & 0 & 1 & -4 \end{pmatrix} \begin{pmatrix} u_{1,1} \\ u_{2,1} \\ u_{2,2} \\ u_{1,2} \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ -1 \end{pmatrix}$$

- Solve by direct method (e.g. Gauss elimination), to obtain $\{u_{1,1}, u_{2,1}, u_{2,2}, u_{1,2}\} = \{3/8, 1/8, 1/8, 3/8\}$
- Whoops – we should have noticed from the set boundary conditions that there is a symmetry around $y = 0.5$, and so $u_{1,1} = u_{1,2}$, and $u_{2,1} = u_{2,2}$. This means we only needed to solve a 2x2 system, rather than a 4x4 system of equations. E.g. it would be sufficient to solve this system:

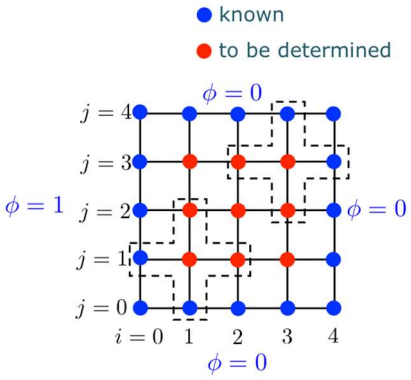
$$\begin{pmatrix} -3 & 1 \\ 1 & -3 \end{pmatrix} \begin{pmatrix} u_{1,1} \\ u_{2,1} \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

406

WE42: Finite difference Laplace

For smaller *h,* increasing number of mesh points => increasing number of equations – sparse matrix forms, with *diagonal dominance*:

$$\underbrace{\begin{pmatrix} 4 & -1 & 0 & -1 & & & & & \\ -1 & 4 & -1 & 0 & -1 & & & & \\ 0 & -1 & 4 & 0 & 0 & -1 & & & \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & & \\ & -1 & 0 & -1 & 4 & -1 & 0 & -1 & \\ & & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ & & & -1 & 0 & 0 & 4 & -1 & 0 \\ & & & & -1 & 0 & -1 & 4 & -1 \\ & & & & & -1 & 0 & -1 & 4 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} \phi_{1,1} \\ \phi_{2,1} \\ \phi_{3,1} \\ \phi_{1,2} \\ \phi_{2,2} \\ \phi_{3,2} \\ \phi_{1,3} \\ \phi_{2,3} \\ \phi_{3,3} \end{pmatrix}}_{\mathbf{y}} = \underbrace{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}}_{\mathbf{b}}$$



Iterative methods for solving systems of linear equations

Convergence – as mesh size approaches zero, the finite difference solution approaches the exact solution.
Stability – the errors are attenuated as the computation progresses.

- As we want more accurate results and decrease the mesh size, it becomes computationally too costly to invert the matrix
- Idea: approximate the solution and obtain improved approximations with each further iteration, until a target accuracy is obtained, i.e. until all perceived relative errors $(\varepsilon_a)_{i,j}$ fall below a pre-specified criterion ε_s:

$$\left|(\varepsilon_a)_{i,j}\right| = \left|\frac{u_{i,j}^{\text{new}} - u_{i,j}^{\text{old}}}{u_{i,j}^{\text{new}}}\right| 100\% < \varepsilon_s$$

Iterative methods for solving systems of linear equations

- Unlike for direct methods, the amount of computation for iterative methods cannot be pre-specified, and depends on the target accuracy
- BUT! They only work if convergent
- Iterative methods are preferred when:
  - The system exhibits rapid convergence (diagonal entries larger than the rest)
  - If the system (matrix) is large and sparse (e.g. p=100, 10,000 elements in matrix, only 500 non-zero), iterative methods can be chosen that are light on storage requirements: can overwrite any solution component *u(x,y)* as soon as a new value is available (e.g. Gauss-Seidel method)

409

Iterative methods for PDEs

- Valid for any square system, diagonal elements must be non-zero!
  - Not a big constraint, as you can re-order lines & columns

$$[A]\{U\} = \{B\}$$

- E.g. for a 3 x 3 matrix (valid for any square matrix)

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \Leftrightarrow \begin{cases} u_1 = (b_1 - a_{12}u_2 - a_{13}u_3)/a_{11} \\ u_2 = (b_2 - a_{21}u_1 - a_{23}u_3)/a_{22} \\ u_3 = (b_3 - a_{31}u_1 - a_{32}u_2)/a_{33} \end{cases}$$

- Start with a guess $(u_1, u_2, u_3)^0$ and improve it iteratively

410

Jacobi method	**Gauss-Seidel (Liebmann's) method**

Starting guess: $(u_1, u_2, u_3)^0$

First iteration

$$\begin{cases} u_1^1 = \left(b_1 - a_{12}u_2^0 - a_{13}u_3^0\right)/a_{11} \\ u_2^1 = \left(b_2 - a_{21}u_1^0 - a_{23}u_3^0\right)/a_{22} \\ u_3^1 = \left(b_3 - a_{31}u_1^0 - a_{32}u_2^0\right)/a_{33} \end{cases}$$

$$\begin{aligned} u_1^1 &= \left(b_1 - a_{12}u_2^0 - a_{13}u_3^0\right)/a_{11} \\ u_2^1 &= \left(b_2 - a_{21}u_1^1 - a_{23}u_3^0\right)/a_{22} \\ u_3^1 &= \left(b_3 - a_{31}u_1^1 - a_{32}u_2^1\right)/a_{33} \end{aligned}$$

In effect: $(u_1, u_2, u_3)^1 = f(u_1, u_2, u_3)^0$

Second iteration

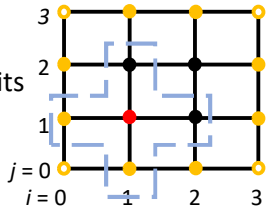
$$(u_1, u_2, u_3)^2 = f(u_1, u_2, u_3)^1$$

$$\begin{aligned} u_1^2 &= \left(b_1 - a_{12}u_2^1 - a_{13}u_3^1\right)/a_{11} \\ u_2^2 &= \left(b_2 - a_{21}u_1^2 - a_{23}u_3^1\right)/a_{22} \\ u_3^2 &= \left(b_3 - a_{31}u_1^2 - a_{32}u_2^2\right)/a_{33} \end{aligned}$$

Generated values are kept for the next iteration.	Generated values are used immediately.

411

WE43: 2D Laplace with Gauss-Seidel method



- $u_{xx} + u_{yy} = 0, u = 0$ on all sides except $u(0, y) = 1$; square of 1 x 1 units

Solution:

- All work from the 1st slide of WE42 is still valid
- Re-arrange equations to have the current mesh point on the left hand side of the equation:

$$u_{1,1} = \left(u_{2,1} + u_{1,2} + 1\right)/4 \qquad u_{2,2} = \left(u_{1,2} + u_{2,1}\right)/4$$
$$u_{2,1} = \left(u_{2,2} + u_{1,1}\right)/4 \qquad u_{1,2} = \left(u_{2,2} + u_{1,1} + 1\right)/4$$

- Implement these equations in Python, Matlab, Excel or by hand, making sure to follow the Gauss-Seidel scheme on the previous slide.
- In fact, we should have used the symmetry of boundary conditions, giving $u_{1,1} = u_{1,2}$, and $u_{2,1} = u_{2,2}$, and only solved for two points, e.g. for $u_{1,1}, u_{2,1}$ the system would be:

$$u_{1,1} = \left(u_{2,1} + 1\right)/3 \qquad u_{2,1} = u_{1,1}/3$$

412

Gauss-Seidel method – convergence criterion

- The Gauss-Seidel method is the most commonly used iterative method
- As with all iterative methods, good starting values can significantly reduce the number of iterations required (e.g. try taking the average of the boundary values)
- As with any method, use the symmetry of the problem, if any, to reduce the number of equations
- A sufficient but not necessary criterion for convergence: diagonally dominant matrix

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|$$

- The diagonal coefficient in each equation must be larger than the sum of the absolute values of the other coefficients in that equation (diagonally dominant system)
- Method might work even when convergence is not guaranteed by this criterion

413

Gauss-Seidel iteration – convergence criterion proof

- It can be shown that a sufficient condition for convergence for such an iteration method (fixed-point iteration) is

$$\left|\frac{\partial f}{\partial x_1}\right| + \left|\frac{\partial f}{\partial x_2}\right| + \left|\frac{\partial f}{\partial x_3}\right| < 1$$

- For the Gauss-Seidel method, these functions $f(x_1, x_2, x_3)$ are given by:

$$f_1 = (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11} \Rightarrow \left|\frac{\partial f_1}{\partial x_1}\right| + \left|\frac{\partial f_1}{\partial x_2}\right| + \left|\frac{\partial f_1}{\partial x_3}\right| = 0 + \left|\frac{a_{12}}{a_{11}}\right| + \left|\frac{a_{13}}{a_{11}}\right| < 1 \Rightarrow |a_{12}| + |a_{13}| < |a_{11}|$$

$$f_2 = (b_2 - a_{21}x_1 - a_{23}x_3)/a_{22} \Rightarrow \left|\frac{\partial f_2}{\partial x_1}\right| + \left|\frac{\partial f_2}{\partial x_2}\right| + \left|\frac{\partial f_2}{\partial x_3}\right| = \left|\frac{a_{21}}{a_{22}}\right| + 0 + \left|\frac{a_{23}}{a_{22}}\right| < 1 \Rightarrow |a_{21}| + |a_{23}| < |a_{22}|$$

$$f_3 = (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33} \Rightarrow \left|\frac{\partial f_3}{\partial x_1}\right| + \left|\frac{\partial f_3}{\partial x_2}\right| + \left|\frac{\partial f_3}{\partial x_3}\right| = \left|\frac{a_{31}}{a_{33}}\right| + \left|\frac{a_{32}}{a_{33}}\right| + 0 < 1 \Rightarrow |a_{31}| + |a_{32}| < |a_{33}|$$

- The diagonal element must be larger than the sum of the non-diagonal elements, for each row in the matrix
- We will re-visit the meaning of this criterion in the chapter on numerical solutions for root finding

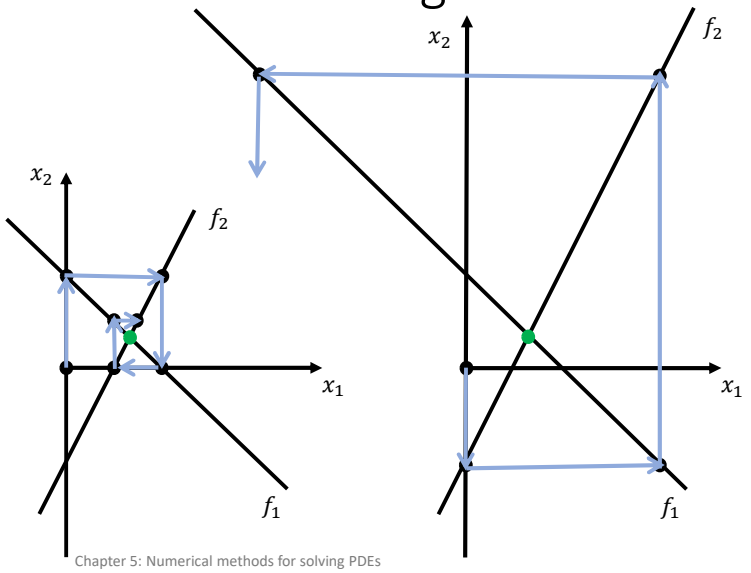
414

Gauss-Seidel iteration – convergence criterion

So what does this mean?

Be careful in which order you use the system equations – the diagonal should be dominant.

Example: with two equations, i.e. 2x2 system, this is what happens if the equations are ordered correctly, or incorrectly.



Relaxation

- *Relaxation* is a modification to the Gauss-Seidel method, applied at every iteration, to improve convergence:

$$u_{i,j}^{\text{new}\prime} = \lambda u_{i,j}^{\text{new}} + (1-\lambda)u_{i,j}^{\text{old}}, 0 < \lambda < 2$$

  - $\lambda = 1 \Rightarrow$ standard method
  - $0 < \lambda < 1 \Rightarrow$ **underrelaxation**, used to dampening oscillations; weighted average of the last two results; used also on nonconvergent systems
  - $1 < \lambda < 2 \Rightarrow$ **overrelaxation**, used to hasten convergence when slow; trust current value and accelerate in the 'same direction'; used only on already convergent systems
- For example, during the second iteration:

$$u_1^{2\prime} = \left(b_1 - a_{12}u_2^1 - a_{13}u_3^1\right)/a_{11}, u_1^2 = \lambda u_1^{2\prime} + (1-\lambda)u_1^1$$
$$u_2^{2\prime} = \left(b_2 - a_{21}u_1^2 - a_{23}u_3^1\right)/a_{22}, u_2^2 = \lambda u_2^{2\prime} + (1-\lambda)u_2^1$$
$$u_3^{2\prime} = \left(b_3 - a_{31}u_1^2 - a_{32}u_2^2\right)/a_{33}, u_3^2 = \lambda u_3^{2\prime} + (1-\lambda)u_3^1$$

Finite difference Poisson equation (elliptic)

$$\nabla^2 u = u_{xx} + u_{yy} = f(x,y)$$

1. Difference quotients corresponding to relevant partial derivatives

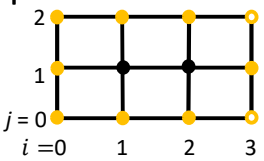
$$u_{xx}(x,y) \approx \frac{1}{h^2}[u(x+h,y) - 2u(x,y) + u(x-h,y)]$$
$$u_{yy}(x,y) \approx \frac{1}{k^2}[u(x,y+k) - 2u(x,y) + u(x,y-k)]$$

2. For $h = k$ (h = mesh size), replace in initial PDE:

$$u(x+h,y) + u(x,y+h) + u(x-h,y) + u(x,y-h) - 4u(x,y) = h^2 f(x,y)$$

417

WE44: Poisson equation with Neumann boundary conditions



What if at (some) boundary points, we know the normal derivative $u_n = \partial u/\partial n$, rather than the value of the function u?

- Solve the Poisson equation $\nabla^2 u = u_{xx} + u_{yy} = xy$ on the domain with length 1.5 and height 1, with $h = 0.5$, and boundary conditions

$$u(x,0) = u(1.5,y) = 0, u(x,1) = x^2, u_n(0,y) = 3y$$

Solution:

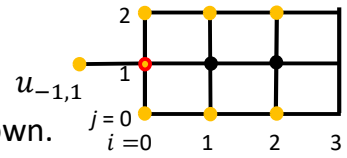
Apply the stencil around the two internal points:

$$u(x+h,y) + u(x,y+h) + u(x-h,y) + u(x,y-h) - 4u(x,y) = 0.5^2 f(x,y)$$

- For $u_{1,1}$: $u_{2,1} + u_{1,2} + u_{0,1} + u_{1,0} - 4u_{1,1} = 0.25^2$ Eq. (1)
- For $u_{2,1}$: $u_{3,1} + u_{2,2} + u_{1,1} + u_{2,0} - 4u_{2,1} = 0.25 \cdot 0.5$ Eq. (2)

418

WE44: Poisson equation with Neumann boundary conditions



We currently have two equations for 3 unknowns.

Despite being on the boundary, the value of $u_{0,1}$ is not known.

Evaluate known values from boundary conditions:

- For $y = 0$: $u_{1,0} = u_{2,0} = 0$; For $x = 1.5$: $u_{3,1} = 0$
- For $y = 1$: $u(x, y = 1) = x^2 \Rightarrow u_{1,2} = 0.25, u_{2,2} = 1$
- For $x = 0$: $u_n(x = 0, y) = 3y \Rightarrow \partial u_{0,1}/\partial n = -\partial u_{0,1}/\partial x = 3 \cdot 0.5$

Extend the domain outside of the boundary with the Neumann condition, and assume the equation continues to hold there:

- For $u_{0,1}$: $u_{1,1} + u_{0,2} + u_{-1,1} + u_{0,0} - 4u_{0,1} = 0.25 \cdot 0 \cdot 0.5$ Eq. (3)
- With $u_{0,0} = 0,\ u_{0,2} = 0^2 = 0$

WE44: Poisson equation with Neumann boundary conditions

- Use the Neumann boundary condition, evaluated in central difference form, to remove the unknown $u_{-1,1}$ from the system of equations:

$$\frac{\partial u_{0,1}}{\partial x} \approx \frac{u_{1,1} - u_{-1,1}}{2h} = \frac{u_{1,1} - u_{-1,1}}{1} \Rightarrow u_{-1,1} = u_{1,1} - \frac{\partial u_{0,1}}{\partial x} = u_{1,1} + 1.5$$

- With this relation for $u_{-1,1}$, and known boundary values plugged in, Eq. (3) becomes:

$$2u_{1,1} - 4u_{0,1} = -1.5$$

- With known boundary values plugged in, Eq. (1) and Eq. (2) are

$$u_{2,1} + u_{0,1} - 4u_{1,1} = 0.25^2 - 0.25$$
$$u_{1,1} - 4u_{2,1} = 0.25 \cdot 0.5 - 1$$

WE44: Poisson equation with Neumann boundary conditions

The system of equations to solve is:

$$\begin{pmatrix} -4 & 2 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & -4 \end{pmatrix} \begin{pmatrix} u_{0,1} \\ u_{1,1} \\ u_{2,1} \end{pmatrix} = \begin{pmatrix} -1.5 \\ -0.1875 \\ -0.875 \end{pmatrix}$$

Which can be solved by Gauss elimination, matrix inversion or Gauss Seidel iteration.

Exact solution: $\{u_{0,1}, u_{1,1}, u_{2,1}\} = \{0.495, 0.240, 0.279\}$

421

Procedure for solving elliptic PDEs with Neumann or mixed boundary conditions

1. Compute information (values or derivatives) on boundary mesh points
2. Extend solution region beyond the Neumann boundaries and assume the same PDE holds there
3. Apply central difference formula for $\partial u/\partial n$ for any mesh points on Neumann boundaries; by this, you incorporate the value of the derivative in the system of equations
4. Solve the system of equations (# internal mesh points to solve for + # boundary mesh points on Neumann boundaries = # equations)

422

Secondary variables

- Secondary variables are related to the function *u(x,y)*, the solution of the PDE
- For example, the Laplace equation is solved for the temperature distribution *T(x,y)*, but the physical quantity of interest might be the heat flux across the 2D plate surface. The heat flux is then a secondary variable.
- In this case, we make use of Fourier's law of heat conduction to relate heat flux and temperature: $\boldsymbol{q} = -\kappa \boldsymbol{\nabla T}$**,** where *q* [W/m^2], thermal conductivity κ [W/(m K)] and temperature T [K].

Secondary variables

$$\boldsymbol{q} = -\kappa \boldsymbol{\nabla T}$$

- In a domain with a square mesh, $\boldsymbol{q}$ is given by:

$$q_n = \sqrt{q_x^2 + q_y^2},\ \theta = \begin{cases} \tan^{-1}(q_y/q_x), q_x > 0 \\ \tan^{-1}(q_y/q_x) + \pi, q_x < 0 \\ \pi/2 \text{ or } 3\pi/2\,, q_x = 0 \end{cases}$$

with components obtained from $T_{i,j}$:

$$q_x = -\kappa \frac{T_{i+1,j} - T_{i-1,j}}{2h},\ q_y = -\kappa \frac{T_{i,j+1} - T_{i,j-1}}{2h}$$

- Another example: a particle diffusion PDE has as solution the concentration of particles *c(x,t)*, while the secondary variable is the flux of particles

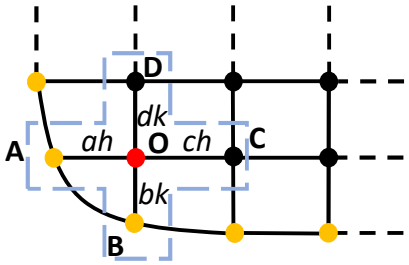
Laplace/Poisson with irregular boundaries

- Irregular = non-rectangular; some points are at an irregular distance away from boundaries.
- We re-derive the expression for $u(x,y)$ in a generalised mesh with unequal mesh size:

$$u_C = u_O + ch\frac{\partial u_O}{\partial x} + \frac{1}{2}(ch)^2\frac{\partial^2 u_O}{\partial x^2} + \cdots \quad (I)$$

$$u_A = u_O - ah\frac{\partial u_O}{\partial x} + \frac{1}{2}(ah)^2\frac{\partial^2 u_O}{\partial x^2} + \cdots \quad (II)$$

- Aim: Find out the central difference approximation for u_{xx}, u_{yy} from *(I)* and *(II)*. We can eliminate $\partial u_O/\partial x$ by calculating $cu_A + au_C$ from the above.



$$u_O = u(x,y)$$
$$u_A = u(x-ah,y)$$
$$u_B = u(x,y-bk)$$
$$u_C = u(x+ch,y)$$
$$u_D = u(x,y+dk)$$

425

Laplace/Poisson with irregular boundaries

$$cu_A + au_C \approx cu_O - cah\frac{\partial u_O}{\partial x} + \cancel{\frac{1}{2}c(ah)^2\frac{\partial^2 u_O}{\partial x^2}} + au_O + ach\frac{\partial u_O}{\partial x} + \cancel{\frac{1}{2}a(ch)^2\frac{\partial^2 u_O}{\partial x^2}}$$

$$cu_A + au_C \approx u_O(a+c) + \frac{1}{2}ach^2(a+c)\frac{\partial^2 u_O}{\partial x^2}$$

$$\frac{\partial^2 u_O}{\partial x^2} \approx \frac{2}{h^2}\left(\frac{u_A}{a(a+c)} + \frac{u_C}{(a+c)c} - \frac{u_O}{ac}\right)$$

- This is $u_{xx}(x,y)$. For a regular mesh $a = c = 1$, and we retrieve the known expression:

$$\frac{\partial^2 u_O}{\partial x^2} \approx \frac{1}{h^2}(u_A + u_C - 2u_O)$$

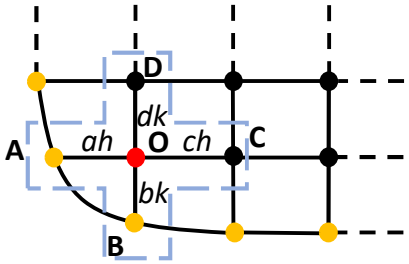
which is identical to expression for rectangular boundaries:

$$u_{xx}(x,y) \approx \frac{1}{h^2}[u(x+h,y) - 2u(x,y) + u(x-h,y)]$$

426

Laplace/Poisson with irregular boundaries

$u_O = u(x,y)$
$u_A = u(x-ah, y)$
$u_B = u(x, y-bk)$
$u_C = u(x+ch, y)$
$u_D = u(x, y+dk)$



- $\frac{\partial^2 u_O}{\partial x^2} \approx \frac{2}{h^2}\left(\frac{u_A}{a(a+c)} + \frac{u_C}{(a+c)c} - \frac{u_O}{ac}\right)$
- It can be shown similarly for points B, O, D:

$$\frac{\partial^2 u_O}{\partial y^2} \approx \frac{2}{k^2}\left(\frac{u_B}{b(b+d)} + \frac{u_D}{(b+d)d} - \frac{u_O}{bd}\right)$$

So that for a general Poisson equation on a general rectangular mesh, $\nabla^2 u = f(x,y)$ as a difference equation is:

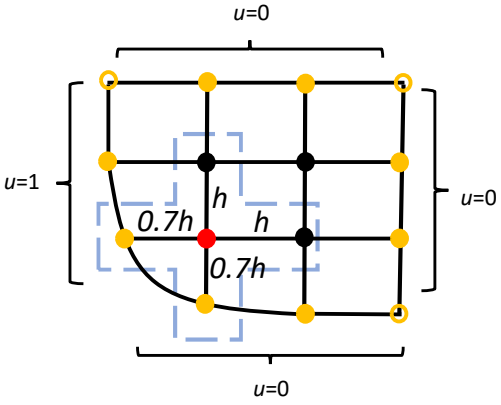
$$\frac{2}{h^2}\left(\frac{u_A}{a(a+c)} + \frac{u_C}{(a+c)c} - \frac{u_O}{ac}\right) + \frac{2}{k^2}\left(\frac{u_B}{b(b+d)} + \frac{u_D}{(b+d)d} - \frac{u_O}{bd}\right) = f(x,y)$$

427

WE45: Laplace with irregular boundaries

Solve the Laplace equation with the boundary conditions as in WE41, but with one curved corner on the boundary

- Given: $a = b = 0.7$
- On a square mesh, $h = k = 0.33$



- From boundary geometry, we know: $c = d = 1$

For the three regular inner points, use the standard expression. For the red point, use the expression derived on the previous slide for $f(x,y) = 0$:

- $\frac{u_A}{a(a+c)} + \frac{u_C}{(a+c)c} - \frac{u_O}{ac} + \frac{u_B}{b(b+d)} + \frac{u_D}{(b+d)d} - \frac{u_O}{bd} = 0$

428

WE45: Laplace with irregular boundaries

Plugging in the values of a, b, c, d:

$$\frac{u_A}{0.7 \cdot 1.7} + \frac{u_C}{1.7 \cdot 1} - \frac{u_O}{0.7} + \frac{u_B}{0.7 \cdot 1.7} + \frac{u_D}{1.7 \cdot 1} - \frac{u_O}{0.7} = 0$$

Substituting the notation in terms of $u_{i,j}$:

$$\frac{u_{0,1}}{0.7 \cdot 1.7} + 2 \cdot \frac{u_{2,1}}{1.7} - \frac{u_{1,1}}{0.7} + \frac{u_{1,0}}{0.7 \cdot 1.7} + \frac{u_{1,2}}{1.7} = 0$$

WE45: Laplace with irregular boundaries

Take into account the boundary conditions:

$$u_{0,1} = 1, \qquad u_{1,0} = 0$$

The system to be solved for the 4 discretisation points:

$$\begin{pmatrix} -2/0.7 & 1/1.7 & 0 & 1/1.7 \\ 1 & -4 & 1 & 0 \\ 0 & 1 & -4 & 1 \\ 1 & 0 & 1 & -4 \end{pmatrix} \begin{pmatrix} u_{1,1} \\ u_{2,1} \\ u_{2,2} \\ u_{1,2} \end{pmatrix} = \begin{pmatrix} -1/(0.7 \cdot 1.7) \\ 0 \\ 0 \\ -1 \end{pmatrix}$$

5.4.2 Parabolic PDEs

Example: The heat (or diffusion) equation

Finite difference method for parabolic PDEs

- Usually characterising time-dependent problems; e.g. heat conduction or diffusion equation

$$\kappa \frac{\partial^2 T}{\partial x^2} - \frac{\partial T}{\partial t} = 0$$

- Or through non-dimensionalisation

$$u_{xx} - u_t = 0$$

- Parabolic PDEs are bounded in space (as elliptic PDEs are), but open-ended in time
- Time-variability introduces the problem of *stability*
- Unlike for elliptic, *convergence* (and *stability*) are not guaranteed for $h \to 0$

Finite difference - diffusion equation

Solve for $u(x,t)$ where

$$u_{xx} - u_t = 0$$

1. Difference quotients corresponding to relevant partial derivatives

$$u_{xx}(x,t) \approx \frac{1}{h^2}[u(x+h,t) - 2u(x,t) + u(x-h,t)] \qquad O(h^2)$$

$$u_t(x,t) \approx \frac{1}{k}[u(x,t+k) - u(x,t)] \qquad O(k)$$

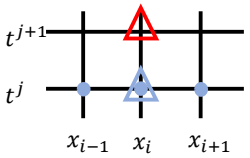
with h = mesh size in x, and k = time step, $h \neq k$. Forward difference in time due to initial conditions; cannot use central difference, as we have no information about negative times.

2. Replace in initial PDE:

$$\frac{1}{k}(u(x,t+k) - u(x,t)) - \frac{1}{h^2}\big(u(x+h,t) - 2u(x,t) + u(x-h,t)\big) = 0$$

Finite difference - diffusion equation by explicit method

$$\frac{1}{k}(u(x,t+k)-u(x,t))-\frac{1}{h^2}\big(u(x+h,t)-2u(x,t)+u(x-h,t)\big)=0$$



- Interpretation: $u(x,t)$ depends on the values of u at the three points at the previous time step

$$u(x,t+k)=r\big(u(x+h,t)+u(x-h,t)\big)+(1-2r)u(x,t),\quad r\equiv\frac{k}{h^2}$$

△ Time discretisation

● Space discretisation

- The larger the time step *k*, the larger the contribution from the previous values of neighbouring points, compared to the contribution of the previous value of the same point
- Given initial and boundary conditions, this explicit method can be used to compute the solution,
- This method is only convergent (and stable) if

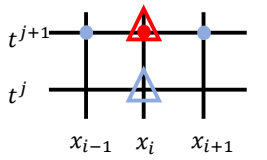
$$1-2r\geq 0\Rightarrow r\leq 1/2\Rightarrow 2k\leq h^2$$

i.e. we cannot move too fast in time.

The problem with the explicit method

- Explicit methods have stability problems, and exclude 'lateral' information that has a bearing on the solution.
- We usually choose small *h* for accuracy (e.g. $h=0.1$) which constraints *k* to extremely small values (e.g. $\mathrm{k}\leq 0.005$); computationally, this is very inefficient.
- For example, if the space mesh size is halved, the number of operations is 4 times larger (for fixed *r* value).
- Implicit methods overcome both these limitations, but their algorithms are more complex.
- There are multiple implicit methods

Finite difference diffusion equation by a simple implicit method



△ Time discretisation

● Space discretisation

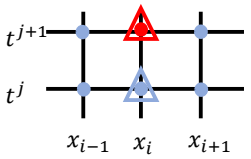
- For example, a simple implicit method is given by:

$$u_{xx}(x,t) \approx \frac{1}{h^2}[u(x+h,t+k) - 2u(x,t+k) + u(x-h,t+k)] \quad O(h^2)$$

$$u_t(x,t) \approx \frac{1}{k}[u(x,t+k) - u(x,t)] \quad O(k)$$

- However, this method is still only first order accurate in time, so it is rarely used
  - It is also not accurate for too large time steps, so not much more efficient than the explicit method

The Crank-Nicolson (implicit) method



△ Time discretisation

● Space discretisation

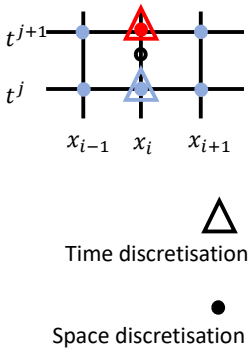
- Widely used, 2nd order accurate in both space and time
- Replace the 'space' quotient in the explicit method by the average of space quotients at two time steps

$$\frac{1}{k}(u(x,t+k) - u(x,t)) = \frac{1}{h^2}\big(u(x+h,t) - 2u(x,t) + u(x-h,t)\big)$$

$$\frac{1}{k}(u(x,t+k) - u(x,t)) = \frac{1}{2h^2}\big(u(x+h,t) - 2u(x,t) + u(x-h,t)\big) + \frac{1}{2h^2}\big(u(x+h,t+k) - 2u(x,t+k) + u(x-h,t+k)\big)$$

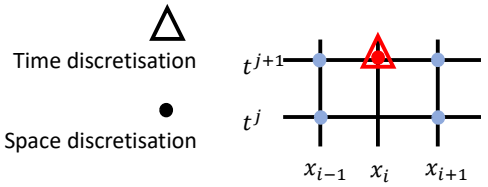
Crank-Nicolson method

$$\frac{1}{k}(u(x,t+k)-u(x,t))$$
$$=\frac{1}{2h^2}\big(u(x+h,t) \quad -2u(x,t) \quad +u(x-h,t)\big)$$
$$+\frac{1}{2h^2}\big(u(x+h,t+k)-2u(x,t+k)+u(x-h,t+k)\big)$$



- Effectively, we are calculating the value of the second derivative at a mid-point
- For $r \equiv \frac{k}{h^2}$, 2k 'times' the above gives:
$$(2+2r)u(x,t+k)-r\big(u(x+h,t+k)+u(x-h,t+k)\big)$$
$$=(2-2r)u(x,t) \quad +r(u(x+h,t) \quad +u(x-h,t))$$
- For each time, we solve the linear system of (n-1) equations

Crank-Nicolson method



- $r=\frac{k}{h^2}$ is no longer restricted to $\leq 1/2$, but smaller *r* values give better results
- Often, the chosen value is $r=1$, for which the method becomes 5-points:

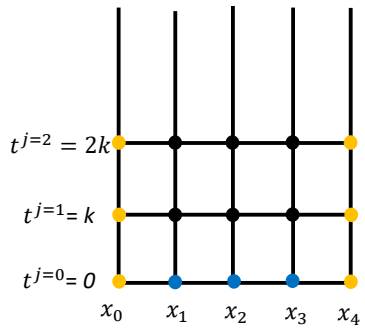
$$4u(x,t+k)-u(x+h,t+k)-u(x-h,t+k)=u(x+h,t)+u(x-h,t)$$

- The Crank-Nicolson method is most often used, particularly for non-uniform meshes
- Derivative boundary conditions can be incorporated by the addition of external points, leading to the addition of extra equations in the system, as for the elliptic equations

WE46: Crank-Nicolson method for the temperature evolution in a bar

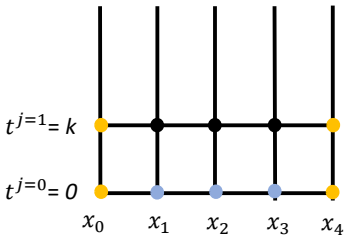
$$4u(x,t+k) - u(x+h,t+k) - u(x-h,t+k) = u(x+h,t) + u(x-h,t)$$

- Diffusion equation for heat propagation across the width of a slab kept between fixed temperatures, 10°C and 50°C, with initial condition *T* = 0°C throughout.
- Width =1, $\kappa = 1$
- Solve the equation for r = 1 and h = 0.25
$$\Rightarrow k = h^2 = 0.0625$$
- Initial conditions: $u_{1,0} = 0 = u_{2,0} = u_{3,0}$;
- Boundary cond's: $u_{0,j} = 10;\ u_{4,j} = 50$
- At each time step beyond *t* = 0, solve a system of 3 equ's and 3 unknowns



441

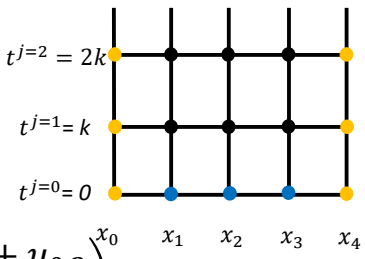
WE45: Crank-Nicolson method for T evolution in a bar



- $@t = k = 0.0625$: $4u_{1,1} - u_{2,1} - u_{0,1} = u_{2,0} + u_{0,0}$
$$4u_{2,1} - u_{3,1} - u_{1,1} = u_{3,0} + u_{1,0}$$
$$4u_{3,1} - u_{4,1} - u_{2,1} = u_{4,0} + u_{2,0}$$
- For this time step:
$$\begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{pmatrix} = \begin{pmatrix} u_{2,0} + u_{0,0} + u_{0,1} \\ u_{3,0} + u_{1,0} \\ u_{4,0} + u_{2,0} + u_{4,1} \end{pmatrix} = \begin{pmatrix} 20 \\ 0 \\ 100 \end{pmatrix}$$
- Solve for $\{u_{1,1}, u_{2,1}, u_{3,1}\}$

442

WE45: Crank-Nicolson method for T evolution in a bar



- @$t = 2k = 0.125$, Solve:

$$\begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} u_{1,2} \\ u_{2,2} \\ u_{3,2} \end{pmatrix} = \begin{pmatrix} u_{2,1} + u_{0,1} + u_{0,2} \\ u_{3,1} + u_{1,1} \\ u_{4,1} + u_{2,1} + u_{4,2} \end{pmatrix}$$

- Where $\{u_{1,1}, u_{2,1}, u_{3,1}\}$ are the solutions from the previous step.

- @$t = 3k$ …

5.4.3 Hyperbolic PDEs

Example: The wave equation

445

Finite difference method for hyperbolic PDEs

- Propagation (time-dependent) problems, with 2nd order time derivative => oscillatory solution.
- E.g. solution of the wave equation is the number of characteristic states with which the string oscillates.
- The wave equation

$$\frac{\partial^2 \varphi}{\partial t^2} - c^2 \frac{\partial^2 \varphi}{\partial x^2} = 0 \quad \text{or} \quad \varphi_{tt} = c^2 \varphi_{xx}$$

- Or by non-dimensionalisation:

$$u_{tt} = u_{xx}$$

- As for parabolic PDEs, the time-variability in hyperbolic PDEs introduces the problem of *stability* and *convergence*

446

Finite difference – The wave equation

Solve for $u(x,t)$ where $u_{tt} = u_{xx}$

1. Difference quotients corresponding to relevant partial derivatives

$$u_{xx}(x,t) \approx \frac{1}{h^2}[u(x+h,t) - 2u(x,t) + u(x-h,t)] \qquad O(h^2)$$

$$u_{tt}(x,t) \approx \frac{1}{k^2}[u(x,t+k) - 2u(x,t) + u(x,t-k)] \qquad O(k^2)$$

with h = mesh size in *x*, and *k* = time step. Central difference in both coordinates.

2. Replace in initial PDE:

$$\frac{1}{k^2}(u(x,t+k) - 2u(x,t) + u(x,t-k)) = \frac{1}{h^2}\big(u(x+h,t) - 2u(x,t) + u(x-h,t)\big)$$

to obtain a 5-point equation.

447

Finite difference – The wave equation

$$\frac{1}{k^2}(u(x,t+k) - 2u(x,t) + u(x,t-k))$$
$$= \frac{1}{h^2}\big(u(x+h,t) - 2u(x,t) + u(x-h,t)\big)$$

with $r = k^2/h^2$

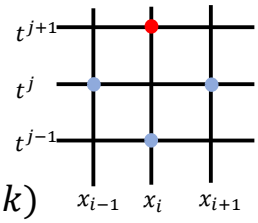
- Stability condition: $r \leq 1$, obtained as for parabolic equations, i.e. contribution of $u(x,t)$ to $u(x,t+k)$ must be positive
- For $r = 1$

$$u(x,t+k) = u(x-h,t) + u(x+h,t) - u(x,t-k)$$

- Note this relation uses the last two time steps
- Unconditionally stable implicit methods also exist for hyperbolic equations; we look at the explicit method only

448

The wave equation by explicit method



$$u(x,t+k) = u(x-h,t) + u(x+h,t) - u(x,t-k)$$

In general, solve for $u(x,t)$ for $0 \le x \le 1$ (normalised dimension) with

- Boundary conditions: $u(0,t), u(1,t)$ known for all $t \ge 0$
- Initial conditions known for all $0 \le x \le 1$
  - Initial displacement: $u(x, t=0)$
  - Initial velocity $u_t(x, t=0)$

Known initial velocity $u_t(x, t=0) = g(x)$ compensates for unknow $u(x, 0-k)$:

$$u_t(x,t=0) = \frac{1}{2k}\big(u(x,0+k) - u(x,0-k)\big) = g(x)$$

449

The wave equation – explicit method

$$g(x) = \frac{1}{2k}\big(u(x,0+k) - u(x,0-k)\big)$$

Extract value at time $(0-k)$

$$u(x,-k) = u(x,k) - 2kg(x)$$

At $t = 0$:

$$u(x,0+k) = u(x+h,0) + u(x-h,0) - u(x,0+k) + 2kg(x)$$

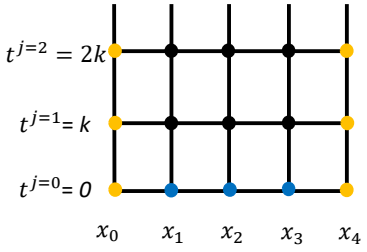
$$u(x,0+k) = \frac{1}{2}\big(u(x+h,0) + u(x-h,0)\big) + kg(x)$$

At $t > 0$, use expression:

$$u(x,t+k) = u(x-h,t) + u(x+h,t) - u(x,t-k)$$

450

WE47: Wave equation by explicit method



Solve for $u(x,t)$ for $0 \leq x \leq 1$ with

- $u(0,t) = u(1,t) = 0$, for all $t \geq 0$
- $u(x,t=0) = f(x)$ for all $0 \leq x \leq 1$
- $u_t(x,t=0) = g(x)$ for all $0 \leq x \leq 1$

For $k = h$:

- At $j = 1$, i.e. $t = k$

$$u(x,0+k) = \frac{1}{2}\big(u(x+h,0) + u(x-h,0)\big) + kg(x)$$

- At $j > 1$, i.e. $t = jk$

$$u(x,t+k) = u(x-h,t) + u(x+h,t) - u(x,t-k)$$

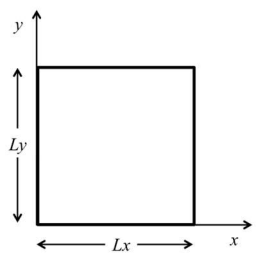
5.4.4 Coding numerical solutions for PDEs

Example of Laplace 2D and wave equation

Laplace 2D code

- Code in Matlab on Blackboard: Laplace_eqn_2D.m

```
%=========================================================================
% Solves Laplace equation phi_xx + phi_yy = 0 using iterative Jacobi method
% and central-difference discretisation.
%
% laura.nicolaou@imperial.ac.uk
%=========================================================================
clear all;
close all;

Lx = 1.0;       % domain length in x
Ly = 1.0;       % domain length in y

nx=49;          % number of gridpoints in x
ny=49;          % number of gridpoints in y
dx=Lx/(nx-1);   % grid spacing in x
dy=Ly/(ny-1);   % grid spacing in y

niter=10000;    % number of iterations
```

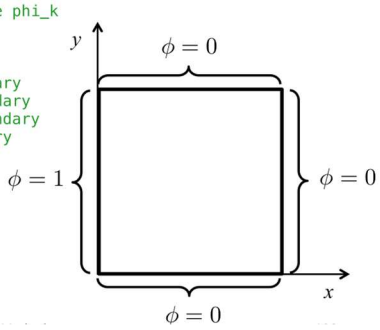


Laplace 2D code

```
% Discretize x and y
x = linspace(0,Lx,nx);
y = linspace(0,Ly,ny);

% Initial Conditions
phi   = zeros(nx,ny);
phi_k = zeros(nx,ny);

% Boundary conditions
phi(1 ,: ) = 1.0;
phi(nx,: ) = 0.0;
phi(: ,1 ) = 0.0;
phi(: ,ny) = 0.0;
```

$$x_i = (i-1)\Delta x, \qquad i = 1, \cdots, nx$$
$$y_j = (j-1)\Delta y, \qquad j = 1, \cdots, ny$$

```
% preallocate phi_k

% left boundary
% right boundary
% bottom boundary
% top boundary
```



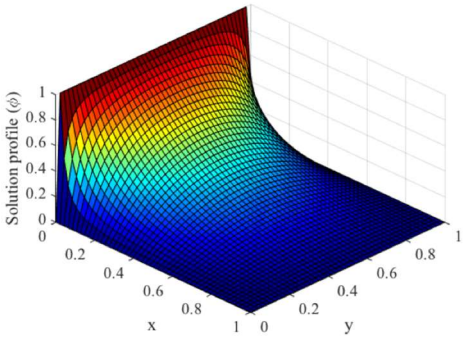
Laplace 2D code

```
%=========================================================================
% Iterative scheme (Jacobi method)
%=========================================================================
for it=1:niter
    phi_k = phi;
    for j=2:ny-1
    for i=2:nx-1
        phi(i,j)=((dy^2*(phi_k(i+1,j)+phi_k(i-1,j))) ...
                 +(dx^2*(phi_k(i,j+1)+phi_k(i,j-1))))/(2*(dx^2+dy^2));
    end
    end
end

% Plot solution
surf(x,y,phi');
colormap jet
xlabel('x')
ylabel('y')
zlabel('Solution profile (\phi)')
```

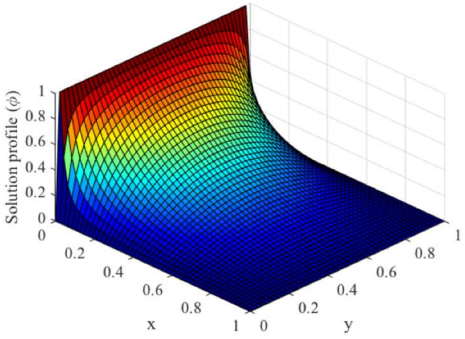
$$\phi_{i,j}^{k+1} = \frac{\Delta y^2(\phi_{i-1,j}^k + \phi_{i+1,j}^k) + \Delta x^2(\phi_{i,j-1}^k + \phi_{i,j+1}^k)}{2(\Delta x^2 + \Delta y^2)}$$

Laplace 2D code

- Numerical solution:



- Analytical solution (separation of variables):



$$\phi = \frac{4}{\pi} \sum_{\substack{n=1 \\ n\ odd}}^{\infty} \frac{\sinh\{n\pi(x-1)\}}{n\sinh(-n\pi)} \sin n\pi y$$

Wave 1D code

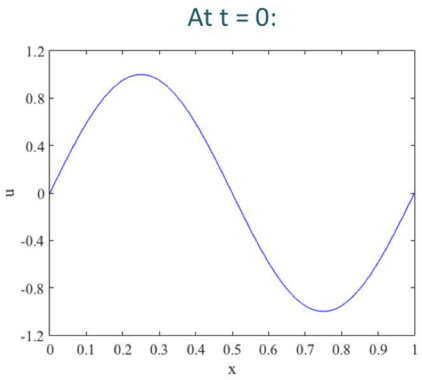
- Code in Matlab on Blackboard: wave_eqn_1D.m

```
%=========================================================================
% Solves the wave equation u_tt = c^2 u_xx using central-differencing.
% Example for vibrating string. Boundary conditions: fixed at both ends.
%
% laura.nicolaou@imperial.ac.uk
%=========================================================================
clear all;
close all;

L = 1.0;                                % space interval
T = 3.0;                                % final time
c = 1.0;                                % propagation speed
n = 2  ;                                % mode of vibration

nx = 101;                               % number of gridpoints in x
dt = 0.005;                             % time step

nt = T/dt +1;                           % number of timesteps
dx = L/(nx-1);                          % grid spacing
CFL = c*dt/dx;                          % CFL number
disp(sprintf('dx= %0.4f, dt= %0.4f, CFL number= %0.3f', dx, dt, CFL))

dt_plot = 0.01;                         % time step to output solution
```

Wave 1D code

```
% Discretize x
x = linspace(0,L,nx);

% Create u array
u = zeros(nx, nt);

% Initial conditions
u0(1:nx,1) = sin(n*pi*x/L);
dudt_0(1:nx,1) = 0.0;

% Boundary conditions
u(1,: ) = 0.0;
u(nx,:) = 0.0;
```

$$x_i = (i-1)\Delta x, \qquad i = 1, \cdots, nx$$



459

Wave 1D code

```
% Solve for interior grid points
u(2:nx-1,1) = u0(2:nx-1,1);
u(2:nx-1,2) = u0(2:nx-1,1) + dt*dudt_0(2:nx-1,1);
for n = 3:nt
   u(2:nx-1,n) = 2*u(2:nx-1,n-1) - u(2:nx-1,n-2) ...
                 + CFL^2*(u(1:nx-2,n-1) - 2*u(2:nx-1,n-1) + u(3:nx,n-1));

   t = (n-1)*dt;

   % Plot solution
   if (mod(t,dt_plot) == 0)
    clf
    plot(x,u0,'b:',x,u(:,n),'r.-')
    axis([0 1 -1.2 1.2])
    xlabel('x')
    ylabel('u')
    title(sprintf('time t=%0.2f',t))
    drawnow
   end
end
```

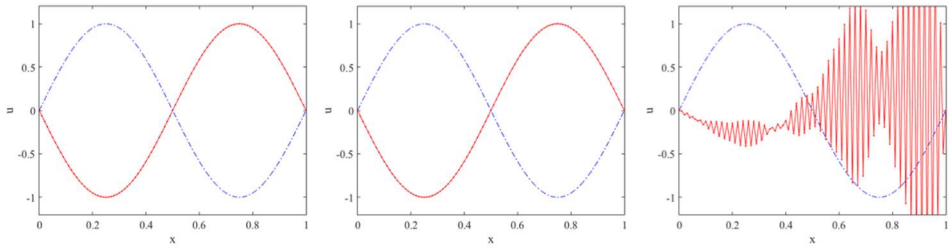
$$u_i^{n+1} = 2u_i^n - u_i^{n-1} + \left(\frac{c\Delta t}{\Delta x}\right)^2 \left(u_{i-1}^n - 2u_i^n + u_{i+1}^n\right)$$

460

Wave 1D code

What happens when we increase the time step?

$\Delta t = 0.005, \quad CFL = 0.5$ $\Delta t = 0.01, \quad CFL = 1.0$ $\Delta t = 0.015, \quad CFL = 1.5$



- CFL condition for stability:

$$CFL = \frac{c\Delta t}{\Delta x} \leq 1$$

461

PDEs beyond engineering
Example: Biological morphogenesis

462

Biological morphogenesis

- Animal pattern formation – Alan Turing* proposed a mechanism based on two interacting chemicals, an activator and an inhibitor
- Imagine the activator promotes the production of a pigment
- Mathematically, this translated to a system of diffusion-reaction equations
- The pattern results from the steady state solution of the chemicals concentration

https://commons.wikimedia.org/wiki/File:Biot%C3%B3pica,_breeding_discus_fish_Turquoise_Carnation.jpg

https://www.chemistryworld.com/features/turing-patterns/4991.article

Turing's Theory of Morphogenesis: Where We Started, Where We Are and Where We Want to Go, Woolley et al

Morphogenesis PDEs

Reaction-diffusion equation for the concentration of two chemicals, $u(x, y, t)$ and $v(x, y, t)$:

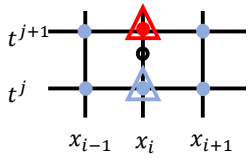
$$\begin{cases} u_t = \gamma(a - u + u^2 v) + (u_{xx} + u_{yy}) \\ v_t = \gamma(b - u^2 v) + d(v_{xx} + v_{yy}) \end{cases}$$

- Initial conditions (so called *fixed point*):

$$u_0 = u(x, y, t = 0) = a + b$$
$$v_0 = v(x, y, t = 0) = \frac{b}{(a + b)^2}$$

- Boundary conditions: homogeneous Neumann on all boundaries (i.e. we don't know the value of those chemicals, but there should be zero flux across the boundaries)

Finite difference system of equations



$$u_{i,l}^{j+1}\left[1+2\theta\Delta t\, d\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}\right)-\theta\Delta t\left(\frac{\partial Q}{\partial u}\right)_{i.l}^{j}\right]=$$

$$=u_{i,l}^{j}+\Delta t\left[Q_{i,l}^{j}-\theta\left(\frac{\partial Q}{\partial u}\right)_{i.l}^{j}u_{i,l}^{j}\right]+\Delta t(1-\theta)d\left[\frac{u_{i-1,l}^{j}-2u_{i,l}^{j}+u_{i+1,l}^{j}}{\Delta x^2}+\frac{u_{i,l-1}^{j}-2u_{i,l}^{j}+u_{i,l+1}^{j}}{\Delta y^2}\right]$$

$$+\theta\Delta t\, d\left[\frac{u_{i-1,l}^{j+1}+u_{i+1,l}^{j+1}}{\Delta x^2}+\frac{u_{i,l-1}^{j+1}+u_{i,l+1}^{j+1}}{\Delta y^2}\right]$$

For Crank Nicholson: $\theta = 0.5$

Neumann boundary conditions, e.g. $\frac{\partial u}{\partial x}(x=0,y)=0 \Rightarrow \frac{u_{-1,l}-u_{1,l}}{2\Delta x}=0$

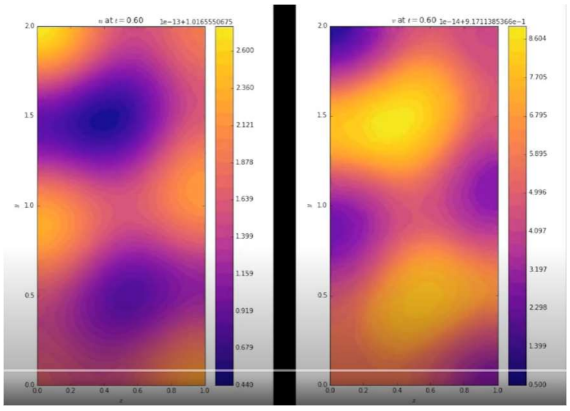
- Parameter values: $\gamma = 100, a = 0.07, b = 0.95, d = 10$

https://pixabay.com/photos/cheetah-and-cubs-cheetahs-wildlife-4405001/

Results

On a 2x1 rectangular domain

Same type of spots form on domains up to 5x1

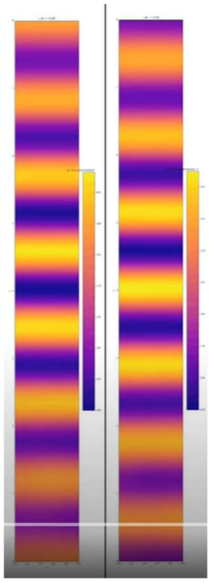


https://youtu.be/AY3D_EC3ODo

Results

On a 8x1 rectangular domain

Same type of stripes form on domains of ratio from 6:1 up.

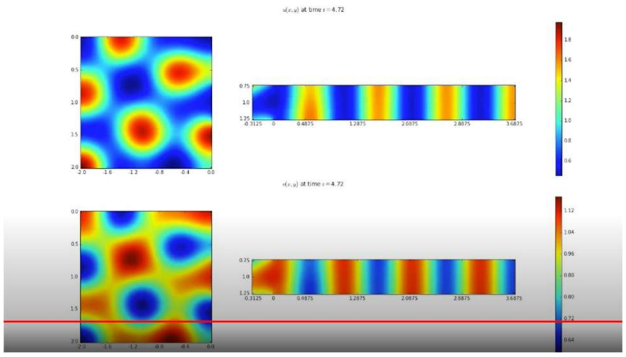


https://youtu.be/6vZIBEIwZ2w

467

Results

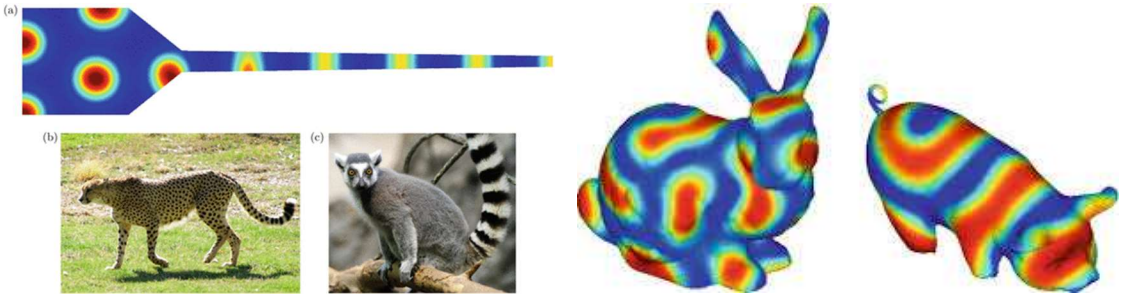
On two inter-connected rectangular domains, with condition of continuity.



https://youtu.be/i2ZyX25NB2E

468

Beyond simple geometries



https://people.maths.ox.ac.uk/maini/PKM%20publications/428.pdf

469

5.5 Finite volume methods (FVM)

Recommended reading:

An introduction to computational fluid dynamics the finite volume method, Versteeg, Malalasekera, 2nd ed. (e-book from ICL library)

470

Learning outcomes

- Describe the difference between finite difference and finite volume methods, and when each would be advantageous

On simplified geometries:

- Solve PDEs describing steady state problems in two dimensions (Laplace and Poisson)
- Solve PDEs describing time-dependent problems in one spatial dimension

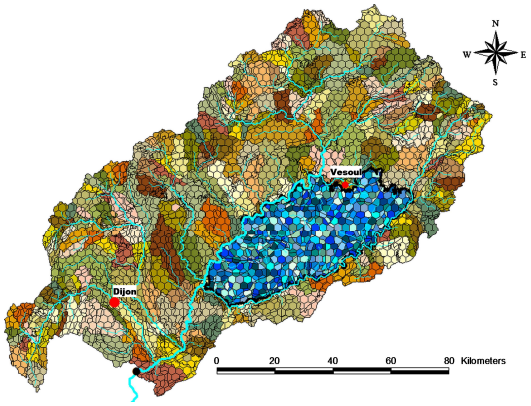
471

Why FVM?

Finite difference is very efficient for regular grids (think block-shaped, no material discontinuities)

Finite volume is preferred when:

- Irregular geometries, unstructured meshes
- Can be more stable
- Ensure local conservation of fluxes (due to how they are derived)
  - Attractive for problems where flux is of importance
- Amenable to adaptive mesh refinement

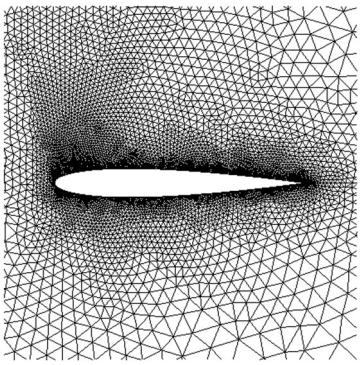


https://ascelibrary.org/doi/10.1061/%28ASCE%29HE.1943-5584.0000296

https://www.machinedesign.com/3d-printing-cad/fea-and-simulation/article/21832072/whats-the-difference-between-fem-fdm-and-fvm

472

How FVM?



http://ta.twi.tudelft.nl/users/wesselin/projects/unstructured.html

1. Discretise domain into cells – *control volume* of simple shape
2. Write local balance for each control volume (i.e. conservation laws)
3. Use divergence theorem to obtain an integral formulation of the fluxes over the boundary of the control volume

   => Flux conservation equations defined in an averaged sense over a control volume
4. Discretise the fluxes over boundaries

Local balance at a control volume

In the control volume (CV):

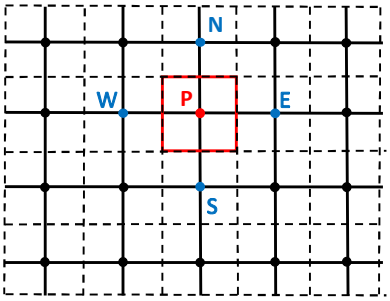
- Rate of change of *u* in CV = increase of *u* due to convection into CV
  + increase of *u* due to diffusion into CV
  + increase of *u* due to creation in CV
- *u* may represent: mass, momentum, energy

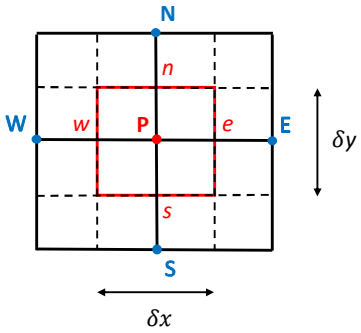
Remember we obtained the heat equation from:

$$\int_{\delta V} \left\{ \nabla \cdot (\kappa \nabla T) + \rho s - \rho c_V \frac{\partial T}{\partial t} \right\} dV = 0$$

Discretise the domain into CVs in 2D

Generate grid: place nodal points and surround them by a CV each





Steady state diffusion (Poisson) in 2D

$$\nabla \cdot (\kappa \nabla T) - \rho c_V \frac{\partial T}{\partial t} = -\rho s$$
$$\nabla(k \nabla u) = -S(x, y)$$

Steady state diffusion equation, most generally written as:

$$\frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial u}{\partial y}\right) + S = 0$$

With material property $k = f_1(x, y)$ and a source/sink of u, $S = f_2(x, y)$.

For example: if $u = T$, then $k = \kappa$ and S is any source of heat within the domain.

Solve for $u(x, y)$ over the domain.

Steady state diffusion (Poisson) in 2D

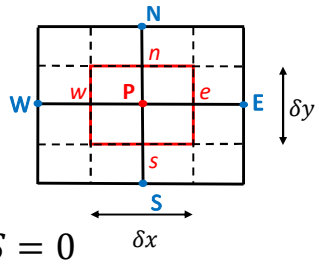
$$\frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial u}{\partial y}\right)+S=0$$

- Integrate over the 'control volume':

$$\int_{\Delta A}\mathrm{d}x\mathrm{d}y\,\frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right)+\int_{\Delta A}\mathrm{d}x\mathrm{d}y\,\frac{\partial}{\partial y}\left(k\frac{\partial u}{\partial y}\right)+\int_{\Delta A}\mathrm{d}x\mathrm{d}y\,S=0$$

477

Local balance at a 'control volume' ΔA



$$\int_{\Delta A}\mathrm{d}x\mathrm{d}y\,\frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right)+\int_{\Delta A}\mathrm{d}x\mathrm{d}y\,\frac{\partial}{\partial y}\left(k\frac{\partial u}{\partial y}\right)+\int_{\Delta A}\mathrm{d}x\mathrm{d}y\,S=0$$

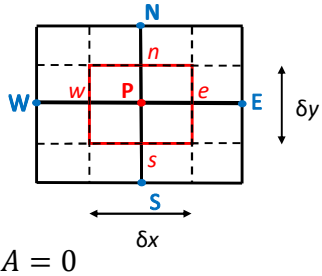
$$\left[k_eA_e\left(\frac{\partial u}{\partial x}\right)_e-k_wA_w\left(\frac{\partial u}{\partial x}\right)_w\right]+\left[k_nA_n\left(\frac{\partial u}{\partial y}\right)_n-k_sA_s\left(\frac{\partial u}{\partial y}\right)_s\right]+\bar{S}\,\Delta A=0$$

(using $A_e=A_w=\delta y$ and $A_n=A_s=\delta x$)

- The generation of *u* must equal the sum of fluxes through the four faces

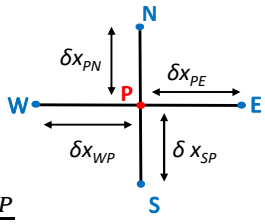
478

Discretise fluxes over boundaries



$$\left[k_e A_e \left(\frac{\partial u}{\partial x}\right)_e - k_w A_w \left(\frac{\partial u}{\partial x}\right)_w\right] + \left[k_n A_n \left(\frac{\partial u}{\partial y}\right)_n - k_s A_s \left(\frac{\partial u}{\partial y}\right)_s\right] + \bar{S}\Delta A = 0$$

Express the fluxes across the 4 faces via central difference:



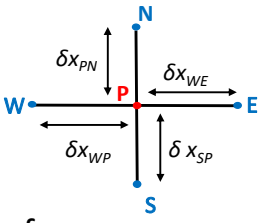
$$k_e A_e \left(\frac{\partial u}{\partial x}\right)_e = k_e A_e \frac{u_E - u_P}{\delta x_{PE}} \qquad k_s A_s \left(\frac{\partial u}{\partial y}\right)_s = k_s A_s \frac{u_P - u_S}{\delta y_{SP}}$$

$$k_w A_w \left(\frac{\partial u}{\partial x}\right)_w = k_w A_w \frac{u_P - u_W}{\delta x_{WP}} \qquad k_n A_n \left(\frac{\partial u}{\partial y}\right)_n = k_n A_n \frac{u_N - u_P}{\delta y_{PN}}$$

Linearise the source term: $\bar{S}\Delta A = S_u + S_p u_P$

479

The nodal point equation



We are looking to cast any problem in the form of a general equation. This will allow us to set up the problem as system of simultaneous equations.

The value at each node must be given by the discretised equation:

$$a_P u_P = a_E u_E + a_W u_W + a_N u_N + a_S u_S + S_u$$

For the expression derived, these become:

$$a_W = \frac{k_w A_w}{\delta x_{WP}} \qquad a_E = \frac{k_e A_e}{\delta x_{PE}}$$

$$a_S = \frac{k_s A_s}{\delta y_{SP}} \qquad a_N = \frac{k_n A_n}{\delta y_{PN}}$$

$$a_P = a_W + a_E + a_S + a_N - S_p$$

480

The nodal point equation - cases

- isotropic material

$$k_w = k_e = k_s = k_n = k$$

- rectangular regular mesh

$$A_w = A_e, A_n = A_s$$

$$\delta x_{WP} = \delta x_{PE}, \delta y_{SP} = \delta y_{PN}$$

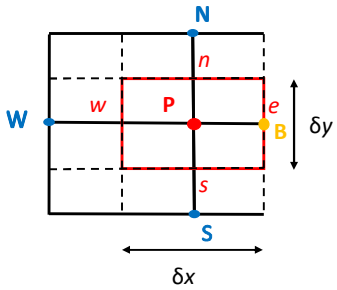
- uniform heat generation density

$$S_u = q\ \delta x\ \delta y\ , S_p = 0$$

- convective heat loss $\propto h(u - u_\infty)$

$$S_u, S_p \neq 0$$

Boundaries – example *East*



Modify the discretised equations for the nodal points close to the boundary:

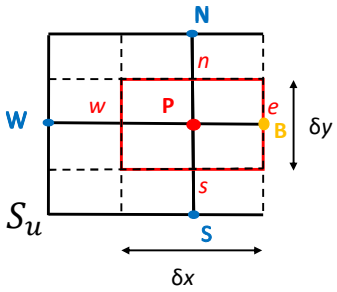
- Example: point P near the *East* boundary

$$\left[k_e A_e \left(\frac{\partial u}{\partial x}\right)_e - k_w A_w \left(\frac{\partial u}{\partial x}\right)_w\right] + \left[k_n A_n \left(\frac{\partial u}{\partial y}\right)_n - k_s A_s \left(\frac{\partial u}{\partial y}\right)_s\right] + \bar{S}\Delta A = 0$$

$$k_e A_e \left(\frac{\partial u}{\partial x}\right)_e = k_e A_e \frac{u_B - u_P}{\delta x_{PB}}$$

- Dirichlet condition B: u_B
- Or directly $\left(\frac{\partial u}{\partial x}\right)_e$ from Neumann condition at B
- For regular rectangular mesh: $\delta x_{PB} = \delta x_{PE}/2$

Boundaries – example *East*



$$a_P u_P = a_E u_E + a_W u_W + a_N u_N + a_S u_S + S_u$$

With:

$$a_W = \frac{k_w A_w}{\delta x_{WP}} \qquad a_E = 0 \qquad a_S = \frac{k_s A_s}{\delta y_{SP}} \qquad a_N = \frac{k_n A_n}{\delta y_{PN}}$$

$$S_p \to S_p - \frac{k_e A_e}{\delta x_{PB}}, S_u \to S_u + \frac{k_e A_e}{\delta x_{PB}} u_B$$

$$a_P = a_W + a_E + a_S + a_N - S_p$$

Time dependence - diffusion equation (not on exam)

$$\nabla \cdot (\kappa \nabla T) - \rho c_V \frac{\partial T}{\partial t} = -\rho s$$

For example, for 1D diffusion we start from:

$$\int_{\Delta V} \mathrm{d}x \int_{t}^{t+\Delta t} \mathrm{d}t \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) + \int_{\Delta V} \mathrm{d}x \int_{t}^{t+\Delta t} S \, \mathrm{d}t = \int_{\Delta x} \mathrm{d}x \int_{t}^{t+\Delta t} \mathrm{d}t \, \frac{\partial T}{\partial t}$$

We obtain the nodal point equation, explicit in time:

$$a_P T_P^j = a_E T_E^{j-1} + a_W T_W^{j-1} + \left[a_P - \left(a_w + a_E - S_p\right)\right] T_P^{j-1} + S_u$$

485

Time dependence - diffusion equation (not on exam)

$$a_P T_P^j = a_E T_E^{j-1} + a_W T_W^{j-1} + \left[a_P - \left(a_w + a_E - S_p\right)\right] T_P^{j-1} + S_u$$

This forms a system of ODEs. The time stepping can be solved via:

- forward Euler (explicit with limitations for stability) – 'method of lines' – only this on the exam! – expression as above
- Crank-Nicolson (implicit, problematic for parabolic equations with fast decaying transients) – see Versteeg if interested
- backward Euler (implicit, used sometimes in combination with Crank-Nicolson)

486

487

Numerical methods for PDEs - Summary

- Classification of PDEs into elliptic, parabolic and hyperbolic indicates which methods might be most suitable; time dependency introduces the issue of stability (and convergence)
- For time-dependent problems, we have practised explicit and implicit methods
  - explicit methods are guaranteed to be stable only when the time step is a given fraction of the mesh size
  - Implicit methods might take more time per step, but have guaranteed stability and can cope with significantly larger time steps
- The most common method for stepping in time is the Crank-Nicolson method. This is an implicit method in the same category as the forward and backward Euler methods used in ODEs, but second order accurate in the mesh size

488

Numerical methods for PDEs - Summary

- In finite volume methods we approximate the average of the solution over a 'control volume'
  - This ensures conservation, as derived from balance of fluxes and sources/sinks
  - This allows easy implementation for irregular meshes
- Solving PDEs implicitly results in solving a system of simultaneous equations
- For matrix inversion
  - use co-factor matrix or Gaussian elimination, if matrix size is small (almost never true in problems of interest in applications)
  - use an iterative method, such as Jacobi or Gauss-Seidel (Liebmann) method for large matrices - equations must be arranged to give a diagonally dominant matrix

489

Chapter 6
Analytical methods for solving PDEs
(by separation of variables)

490

Contents

491

Solution by Separation of Variables

- Partial differential equations are generally very difficult to solve
- For general problems engineers use numerical methods
  - Finite elements, finite differences, etc
- Some problems can be solve (semi-) analytically using a technique called **separation of variables**
- We will study a few particular cases:
  - Laplace equation on a square
  - Laplace equation on a disc
  - (Laplace equation on a cylinder)
  - 1D diffusion equation
  - 1D wave equation

492

Learning outcomes

- Given a system, be able to specify boundary conditions, and to assess whether a problem is well posed (no. of initial & boundary conditions)
- Solve
  - the 2D Laplace in carthesian and polar coordinates
  - the 1D diffusion equation
  - the 1D wave equation
- Solve = given a well posed problem, create the general solution for the equation and apply the constraints imposed by ICs and BCs to find the specific solution
- Use the superposition property and the symmetry of the system to obtain solutions to similar problems
- Solve problems with multiple Neumann boundary conditions, and initial or boundary conditions that are not constant *f(x)*, *f(t)*
- Employ Fourier series to manipulate the inhomogeneous boundary condition

493

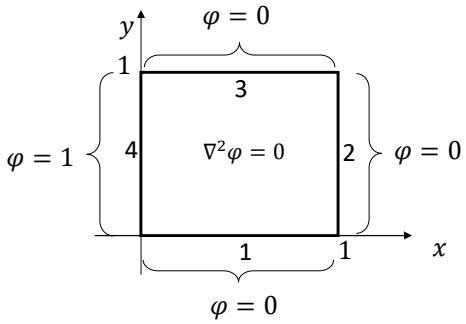
6.1 The Laplace equation

2D in Cartesian, polar (and cylindrical)

494

Separation of Variables: Laplace (2D)

- Consider Laplace equation in 2D where the domain is the unit square shown below
- Let us also assume we have the Dirichlet boundary conditions indicated:



Separation of Variables: Laplace (2D)

- First we must construct the **general solution**, that is, we will ignore for the moment, boundary conditions
- In this case then, we have $\nabla^2\varphi = 0 \Rightarrow \dfrac{\partial^2\varphi}{\partial x^2} + \dfrac{\partial^2\varphi}{\partial y^2} = 0$
- We proceed by 'separating the variables', that is, we write φ as the product of two functions, one wholly in x and one wholly in y:

$$\varphi(x, y) = X(x)Y(y)$$

- Substituting this into the equation above gives: $Y\dfrac{\mathrm{d}^2X}{\mathrm{d}x^2} + X\dfrac{\mathrm{d}^2Y}{\mathrm{d}y^2} = 0$
- And dividing both sides by $\varphi(x, y) = X(x)Y(y)$ gives

$$\frac{1}{X}\frac{\mathrm{d}^2X}{\mathrm{d}x^2} + \frac{1}{Y}\frac{\mathrm{d}^2Y}{\mathrm{d}y^2} = 0$$

Separation of Variables: Laplace (2D)

- Note then, by this separation we have converted $\nabla^2\varphi = 0$ into

$$\frac{1}{X}\frac{\mathrm{d}^2X}{\mathrm{d}x^2} + \frac{1}{Y}\frac{\mathrm{d}^2Y}{\mathrm{d}y^2} = 0$$

- Note that the first term is only a function of *x* and the second term is only a function of *y*. At every point of the domain, however, their sum must vanish.
- For this to be generally true, each term must be a fixed constant k^2, and one possibility (there is another, see later) is that

$$\frac{1}{X}\frac{\mathrm{d}^2X}{\mathrm{d}x^2} = k^2 \qquad \frac{1}{Y}\frac{\mathrm{d}^2Y}{\mathrm{d}y^2} = -k^2$$

- Rearranging gives two separate equations:

$$\frac{\mathrm{d}^2X}{\mathrm{d}x^2} - k^2X = 0 \qquad \frac{\mathrm{d}^2Y}{\mathrm{d}y^2} + k^2Y = 0$$

497

Separation of Variables: Laplace (2D)

- From the last slide, we have the following equations:

$$\frac{\mathrm{d}^2X}{\mathrm{d}x^2} - k^2X = 0 \qquad \frac{\mathrm{d}^2Y}{\mathrm{d}y^2} + k^2Y = 0$$

- These are just ordinary differential equations, whose solutions are well known:

$$e^{-kx}, e^{kx} \qquad \cos ky\,, \sin ky$$

- Note also that we could have chosen to solve (the choice used was entirely arbitrary):

$$\frac{\mathrm{d}^2X}{\mathrm{d}x^2} + k^2X = 0 \qquad \frac{\mathrm{d}^2Y}{\mathrm{d}y^2} - k^2Y = 0$$

- Solutions to these equations are then

$$\cos kx\,, \sin kx \qquad e^{-ky}, e^{ky}$$

- The general solution then will be some linear combination of products of these solutions

498

Separation of Variables: Laplace (2D)

- The general solution then will be some linear combination of the following terms for all possible k:

$$e^{-kx}\cos ky\,,\ e^{-k}\ \sin ky\,,\ e^{kx}\cos ky\,,\ e^{kx}\sin ky\,,$$
$$e^{-k}\ \cos kx\,,\ e^{-ky}\sin kx\,,\ e^{ky}\cos kx\,,\ e^{ky}\sin kx$$

i.e.

$$\varphi(x,y)=\sum_k\begin{pmatrix}A_k e^{-k}\ \cos ky+B_k e^{-k}\ \sin ky+C_k e^{kx}\cos ky+D_k e^{kx}\sin ky\\ +E_k e^{-k}\ \cos kx+F_k e^{-ky}\sin kx+G_k e^{ky}\cos kx+H_k e^{ky}\sin kx\end{pmatrix}$$

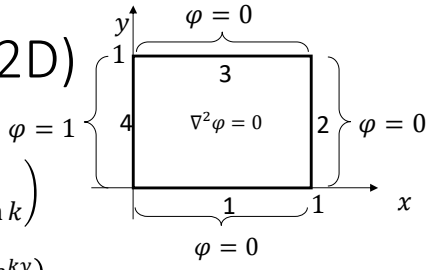
- We now need to determine the constants A-H. To do this we use the boundary conditions
- We begin with the 'zero' conditions on three of the edges:
- Edge 1 (y = 0, φ = 0): By substitution we have:

$$0=\sum_k\begin{pmatrix}A_k e^{-k}\ +C_k e^{kx}+\\ +E_k\cos kx+F_k\sin kx+G_k\cos kx+H_k\sin kx\end{pmatrix}\Rightarrow A_k=0, C_{\mathrm{k}}=0, E_{\mathrm{k}}=-G_{\mathrm{k}}, F_{\mathrm{k}}=-H_k$$

$$\varphi(x,y)=\sum_k\begin{pmatrix}B_k e^{-kx}\sin ky+D_k e^{kx}\sin ky+\\ +E_k e^{-k}\ \cos kx+F_k e^{-ky}\sin kx-E_k e^{ky}\cos kx-F_k e^{ky}\sin kx\end{pmatrix}$$

499

Separation of Variables: Laplace (2D)



- Edge 2 (x = 1, φ = 0): We must now have

$$0=\sum_k\begin{pmatrix}B_k e^{-k}\sin ky+D_k e^{k}\sin ky+\\ +E_k e^{-ky}\cos k+F_k e^{-k}\ \sin k-E_k e^{ky}\cos k-F_k e^{ky}\sin k\end{pmatrix}$$

$$0=\sum_k\left(B_k e^{-k}+D_k e^{k}\right)\sin ky+\left(E_k\cos k+F_k\sin k\right)\left(e^{-ky}-e^{ky}\right)$$

$$\Rightarrow\begin{cases}B_k e^{-k}+D_k e^{k}=0\Rightarrow -B_k e^{-k}=D_k e^{k}\Rightarrow D_k=-B_k e^{-2k}\\ E_k\cos k+F_k\sin k=0\Rightarrow E_k=-F_k\tan k\end{cases}$$

$$\Rightarrow\varphi(x,y)=\sum_k\left(B_k e^{-kx}-B_k e^{kx}e^{-2}\ \right)\sin ky+\left(-F_k\tan k\cos kx+F_k\sin kx\right)\left(e^{-ky}-e^{ky}\right)$$

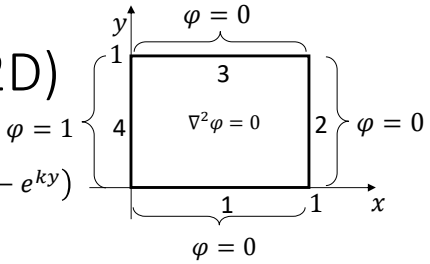
$$B_k\left(e^{-k}\ -e^{k(x-2)}\right)\sin ky=2B_k\frac{\left(e^{-k(x-1)}-e^{k(x-1)}\right)}{2}e^{-k}\sin ky=a_k\sinh\{k(x-1)\}\sin ky\qquad a_k=-2B_k e^{-k}$$

$$\Rightarrow\varphi(x,y)=\sum_k a_k\sinh\{k(x-1)\}\sin ky+F_k(\sin kx-\tan k\cos kx)\left(e^{-k}\ -e^{ky}\right)$$

500

501

Separation of Variables: Laplace (2D)



$$\varphi(x,y) = \sum_k a_k \sinh\{k(x-1)\}\sin ky + F_k(\sin kx - \tan k \cos kx)\left(e^{-ky} - e^{ky}\right)$$

- Then we need to consider Edge 3 (y = 1, φ = 0):

$$0 = \sum_k a_k \sinh\{k(x-1)\}\sin k + F_k(\sin kx - \tan k \cos kx)\left(e^{-k} - e^{k}\right) \quad \Rightarrow \begin{cases} k = n\pi, n = 0,1,2,3,\cdots \\ F_k = 0 \end{cases}$$

- If $\sin kx - \tan k \cos kx = 0 \Rightarrow \tan k = \tan kx \Rightarrow k = kx + m\pi \Rightarrow k = \dfrac{m\pi}{1-x}$, k function of x

Impossible!

$$\therefore \varphi(x,y) = \sum_{n=1}^{\infty} a_n \sinh\{n\pi(x-1)\}\sin n\pi y$$

502

Separation of Variables: Laplace (2D)

- We must now determine the constants a_n in $\varphi(x,y) = \sum_{n=1}^{\infty} a_n \sinh\{n\pi(x-1)\} \sin n\pi y$
- Edge 4 (x = 0, φ = 1): We must now have:

$$1 = \sum_{n=1}^{\infty} a_n \sinh\{-n\pi\} \sin n\pi y$$

- Multiply both sides by $\sin m\pi y$

$$\sin m\pi y = \sum_{n=1}^{\infty} a_n \sinh\{-n\pi\} \sin n\pi y \sin m\pi y$$

- Integrate along the edge (from y = 0 to y = 1)

$$\int_0^1 \sin m\pi y \, dy = \sum_{n=1}^{\infty} a_n \sinh\{-n\pi\} \int_0^1 (\sin n\pi y \sin m\pi y) \, dy$$

- It can be shown that $\int_0^1 (\sin n\pi y \sin m\pi y) \, dy = \begin{cases} \frac{1}{2} & \text{if } m = n \\ 0 & \text{if } m \neq n \end{cases}$

503

Separation of Variables:

$$\therefore \varphi(x,y) = \sum_{n=1}^{\infty} a_n \sinh\{n\pi(x-1)\} \sin n\pi y$$

- Therefore:

$$\int_0^1 \sin m\pi y \, dy = \frac{1}{2} a_m \sinh(-m\pi) \Rightarrow \frac{1-(-1)^m}{m\pi} = \frac{1}{2} a_m \sinh(-m\pi)$$

- Hence:

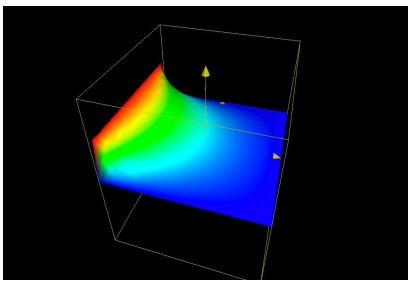
$$a_m = \frac{2(1-(-1)^m)}{m\pi \sinh(-m\pi)}$$

- So, we only have non-zero a_m when m is odd, i.e.

$$a_m = \frac{4}{m\pi \sinh(-m\pi)}, \qquad m \text{ odd}$$

- Finally then

$$\varphi = \frac{4}{\pi} \sum_{\substack{n=1 \\ n \text{ odd}}}^{\infty} \frac{\sinh\{n\pi(x-1)\}}{n \sinh(-n\pi)} \sin n\pi y$$

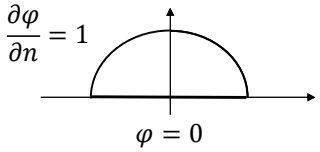


504

505

Separation of Variables: Laplace (2D polar)

- We can also solve the Laplace equation in problems with circular symmetry
- For example the disc in 2D: Consider the semi-circular region below of radius 1 with associated boundary conditions:



$$\nabla^2 \varphi = 0$$

- We wish to solve this in **polar** coordinates, recall that (in 2D):

$$\nabla \varphi \equiv \frac{\partial \varphi}{\partial r}\mathbf{u}_r + \frac{1}{r}\frac{\partial \varphi}{\partial \theta}\mathbf{u}_\theta \quad \text{and} \quad \nabla \cdot \mathbf{v} = \frac{\partial v_r}{\partial r} + \frac{v_r}{r} + \frac{1}{r}\frac{\partial v_\theta}{\partial \theta}$$

- If $\mathbf{v} = \nabla \varphi$ then,

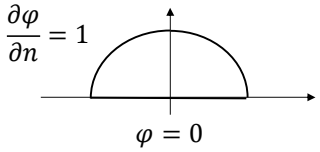
$$\nabla^2 \varphi \equiv \nabla \cdot (\nabla \varphi) = \frac{\partial}{\partial r}\left(\frac{\partial \varphi}{\partial r}\right) + \frac{1}{r}\frac{\partial \varphi}{\partial r} + \frac{1}{r}\frac{\partial}{\partial \theta}\left(\frac{1}{r}\frac{\partial \varphi}{\partial \theta}\right) = \frac{\partial^2 \varphi}{\partial r^2} + \frac{1}{r}\frac{\partial \varphi}{\partial r} + \frac{1}{r^2}\frac{\partial^2 \varphi}{\partial \theta^2}$$

506

507

Separation of Variables: Laplace (2D polar)

- We can also solve the Laplace equation in problems with circular symmetry
- For example the disc in 2D: Consider the semi-circular region below of radius 1 with associated boundary conditions:



$$\nabla^2 \varphi = 0$$

- So we have:

$$\nabla^2 \varphi = 0 \Rightarrow \frac{\partial^2 \varphi}{\partial r^2} + \frac{1}{r}\frac{\partial \varphi}{\partial r} + \frac{1}{r^2}\frac{\partial^2 \varphi}{\partial \theta^2} = 0$$

- Using the method of separation of variables, we assume

$$\varphi = R(r)\Theta(\theta)$$

508

$\varphi = R(r)\Theta(\theta)$

Separation of Variables: Laplace (2D polar)

- Substitution then gives: $\Theta\frac{\partial^2 R}{\partial r^2} + \Theta\frac{1}{r}\frac{\partial R}{\partial r} + \frac{R}{r^2}\frac{\partial^2 \Theta}{\partial \theta^2} = 0$
- Dividing by $R(r)\Theta(\theta)/r^2$, then gives: $\frac{r^2}{R}\frac{\partial^2 R}{\partial r^2} + \frac{r}{R}\frac{\partial R}{\partial r} + \frac{1}{\Theta}\frac{\partial^2 \Theta}{\partial \theta^2} = 0$
- The first two terms are only functions of r and the final term is only a function of θ, hence, as before, we can assume that

$$\frac{r^2}{R}\frac{\partial^2 R}{\partial r^2} + \frac{r}{R}\frac{\partial R}{\partial r} = k^2 \quad \text{and} \quad \frac{1}{\Theta}\frac{\partial^2 \Theta}{\partial \theta^2} = -k^2 \quad \text{for any real number } k$$

- or

$$r^2\frac{\partial^2 R}{\partial r^2} + r\frac{\partial R}{\partial r} = k^2 R \quad \text{and} \quad \frac{\partial^2 \Theta}{\partial \theta^2} + k^2\Theta = 0 \quad \text{for any real number } k$$

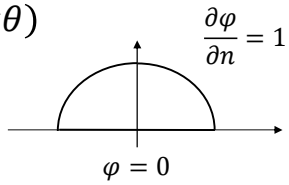
- Solutions are:

$$R(r) = A_k r^k + B_k r^{-k} \qquad \Theta(\theta) = C_k \cos k\theta + D_k \sin k\theta$$

509

Separation of Variables: Laplace (2D polar)

$$\varphi(r,\theta) = \sum_k \left(A_k r^k + B_k r^{-k}\right)(C_k \cos k\theta + D_k \sin k\theta)$$



Firstly, we know that the solution must be finite at r = 0:

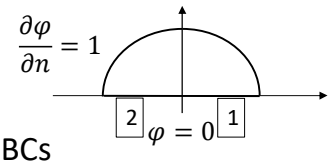
$$\varphi(0,\theta) = 0 \Rightarrow B_k = 0$$

$$\Rightarrow \varphi(r,\theta) = \sum_k A_k r^k (C_k \cos k\theta + D_k \sin k\theta)$$

Now we can consider the boundary conditions.

510

Separation of Variables: Laplace (2D polar)



Firstly, consider the flat base: ($\theta = 0$ and $\theta = \pi$)

• $\theta = 0, \varphi = 0 \Rightarrow 0 = \sum_k A_k r^k (C_k) \Rightarrow C_k = 0 \; \forall k$

Note: if A_k = 0, the whole solution = 0, unsuitable for given BCs

$$\Rightarrow \varphi(r,\theta) = \sum_k A_k D_k r^k \sin k\theta$$

Let $A_k D_k \coloneqq a_k \Rightarrow \varphi(r,\theta) = \sum_k a_k r^k \sin k\theta$

• $\theta = \pi, \varphi = 0 \Rightarrow 0 = \sum_k a_k r^k \sin k\pi \Rightarrow k = m, m = 1,2,3, \ldots$

$$\Rightarrow \; \varphi(r,\theta) = \sum_{m=1}^{\infty} a_m r^m \sin m\theta$$

511

S of Var: Laplace (2D polar)

$$\varphi(r,\theta) = \sum_{m=1}^{\infty} a_m r^m \sin m\theta$$

The other boundary condition is: $r = 1, \frac{\partial\varphi}{\partial n} = 1$

Recall: $\nabla\varphi \coloneqq \frac{\partial\varphi}{\partial r}\mathbf{u}_r + \frac{1}{r}\frac{\partial\varphi}{\partial\theta}\mathbf{u}_\theta \Rightarrow \mathbf{u}_r \cdot \nabla\varphi \coloneqq \frac{\partial\varphi}{\partial r}$

$$\frac{\partial\varphi}{\partial n} \coloneqq \mathbf{n}\cdot\nabla\varphi = \mathbf{u}_r \cdot \nabla\left(\sum_{m=1}^{\infty} a_m r^m \sin m\theta\right) = \sum_{m=1}^{\infty} a_m \frac{\partial}{\partial r} r^m \sin m\theta = \sum_{m=1}^{\infty} a_m m r^{m-1} \sin m\theta$$

$$\text{At } r = 1 \Rightarrow 1 = \sum_{m=1}^{\infty} a_m m \sin m\theta \qquad \Rightarrow \qquad \sin n\theta = \sum_{m=1}^{\infty} a_m m \sin m\theta \sin n\theta$$

512

Separation of Variables: Laplace (2D polar)

$$\sin n\theta = \sum_{m=1}^{\infty} a_m m \sin m\theta \sin n\theta$$

Integrate around the boundary on which this condition applies:

$$\int_0^{\pi} \sin n\theta \, d\theta = \sum_{m=1}^{\infty} a_m m \int_0^{\pi} \sin m\theta \sin n\theta \, d\theta$$

$$\Rightarrow a_n n \frac{\pi}{2} = \begin{cases} 0 & \text{if } n \text{ even} \\ 2/n & \text{if } n \text{ odd} \end{cases} \Rightarrow a_n = \begin{cases} 0 \text{ if } n \text{ even} \\ 4/(\pi n^2) & \text{if } n \text{ odd} \end{cases}$$

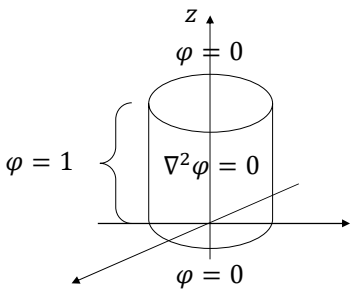
$$\varphi(r,\theta) = \frac{4}{\pi} \sum_{\substack{n=1 \\ n \text{ odd}}}^{\infty} \frac{r^n}{n^2} \sin n\theta$$

513

514

Separation of Variables: Laplace (3D, cylinder) (not on exam)

- Consider Laplace equation in 3D where the domain is the cylinder shown below
- Let us **assume** that the boundary conditions on the curved surface are independent of θ
- Let us also assume we have the Dirichlet boundary conditions indicated



Separation of Variables: Laplace (3D, cylinder) (not on exam)

- Again we must construct the general solution, that is, we will ignore for the moment, boundary conditions
- In this case then, we have $\nabla^2\varphi = 0 \Rightarrow \frac{\partial^2\varphi}{\partial r^2} + \frac{1}{r}\frac{\partial\varphi}{\partial r} + \frac{\partial^2\varphi}{\partial z^2} = 0$
- We proceed by 'separating the variables', that is, we write φ as the product of two functions, one wholly in r and one wholly in z:

$$\varphi(r,z) = R(r)Z(z)$$

- Substituting this into the equation above gives: $Z\frac{\partial^2 R}{\partial r^2} + Z\frac{1}{r}\frac{\partial R}{\partial r} + R\frac{\partial^2 Z}{\partial z^2} = 0$
- And dividing both sides by $\varphi(r,z) = R(r)Z(z)$ gives $\frac{1}{R}\frac{\partial^2 R}{\partial r^2} + \frac{1}{r}\frac{1}{R}\frac{\partial R}{\partial r} + \frac{1}{Z}\frac{\partial^2 Z}{\partial z^2} = 0$

Separation of Variables: Laplace (3D, cylinder) (not on exam)

- Note then, by this separation we have converted $\nabla^2\varphi = 0$ into

$$\frac{1}{R}\frac{\partial^2 R}{\partial r^2} + \frac{1}{r}\frac{1}{R}\frac{\partial R}{\partial r} + \frac{1}{Z}\frac{\partial^2 Z}{\partial z^2} = 0$$

- Note that the first two terms are only a function of r and the third term is only a function of z. At every point of the domain, however, their sum must vanish.
- For this to be generally true, each term must be a fixed constant k^2, and one possibility (there is another, see later) is that

$$\frac{1}{R}\frac{\partial^2 R}{\partial r^2} + \frac{1}{r}\frac{1}{R}\frac{\partial R}{\partial r} = -k^2 \qquad \frac{1}{Z}\frac{\partial^2 Z}{\partial z^2} = k^2$$

- Rearranging gives the two separate equations:

$$r^2\frac{\partial^2 R}{\partial r^2} + r\frac{\partial R}{\partial r} + k^2 r^2 R = 0 \qquad \frac{\partial^2 Z}{\partial z^2} - k^2 Z = 0$$

517

Separation of Variables: Laplace (3D, cylinder) (not on exam)

- Consider the first equation and make the following substitution: $x = kr$

$$r^2\frac{\partial^2 R}{\partial r^2} + r\frac{\partial R}{\partial r} + k^2 r^2 R = \frac{x^2}{k^2}\frac{\partial^2 R}{\partial x^2}k^2 + \frac{x}{k}\frac{\partial R}{\partial x}k + x^2 R$$

$$x^2\frac{\partial^2 R}{\partial x^2} + x\frac{\partial R}{\partial x} + x^2 R = 0 \qquad R(x) = J_0(x) = J_0(kr)$$

Bessel's equation — Zeroth order Bessel function

- Consider the second equation:

$$\frac{\partial^2 Z}{\partial z^2} - k^2 Z = 0 \qquad Z(z) = Ae^{kz} + Be^{-kz}$$

- General solutions arising from this will be of the form: $R(r)Z(z) = J_0(kr)\left(Ae^{kz} + Be^{-kz}\right)$
- At z = 0, φ = 0: $J_0(kr)(A + B) = 0 \Rightarrow A = -B$
- At z = 1, φ = 0: $J_0(kr)\left(Ae^{k} + Be^{-k}\right) = 0 \Rightarrow A = B = 0$

None of these terms can contribute to the solution

518

Separation of Variables: Laplace (3D, cylinder) (not on exam)

- Of course, we can also choose

$$\frac{1}{R}\frac{\partial^2 R}{\partial r^2}+\frac{1}{r}\frac{1}{R}\frac{\partial R}{\partial r}=k^2 \qquad \frac{1}{Z}\frac{\partial^2 Z}{\partial z^2}=-k^2$$

- Rearranging gives, and again making the substitution: $x = kr$

Modified Bessel's equation	$x^2\frac{\partial^2 R}{\partial x^2}+x\frac{\partial R}{\partial x}-x^2R=0$	$\frac{\partial^2 Z}{\partial z^2}+k^2Z=0$
Zeroth order Modified Bessel function	$R(x)=I_0(x)=I_0(kr)$	$Z(z)=C\cos kz+D\sin kz$

- General solutions arising from this will be of the form: $R(r)Z(z)=I_0(kr)(C\cos kz+B\sin kz)$
- At z = 0, φ = 0: $I_0(kr)(C)=0\Rightarrow C=0$
- At z = 1, φ = 0: $I_0(kr)(B\sin k)=0\Rightarrow k=m\pi, m=1,2,3,\ldots$

519

Separation of Variables: Laplace (3D, cylinder) (not on exam)

- Hence we now have solutions of the form $\varphi=\sum_{m=1}^{\infty}a_m I_0(m\pi r)\sin m\pi z$
- To determine the a_m, we need to apply the boundary condition on the curved surface ($r=1, \varphi=1$):

$$\therefore 1=\sum_{m=1}^{\infty}a_m I_0(m\pi)\sin m\pi z \Rightarrow \int_0^1 \sin n\pi z\,dz=\int_{z=0}^{1}\sum_{m=1}^{\infty}a_m I_0(m\pi)\sin m\pi z\sin n\pi z\,dz$$

$$\frac{1-(-1)^n}{n\pi}=\frac{1}{2}a_n I_0(n\pi) \quad\Rightarrow\quad a_n=\begin{cases}0 & \text{if } n \text{ even}\\ 4/\left(n\pi I_0(n\pi)\right) & \text{if } n \text{ odd}\end{cases}$$

- Hence:

$$\varphi=\frac{4}{\pi}\sum_{\substack{n=1\\ n\text{ odd}}}^{\infty}\frac{I_0(n\pi r)}{nI_0(n\pi)}\sin n\pi z$$

520

6.2 The diffusion equation

Source-free, 1D

521

Separation of Variables: Diffusion equation

- Apply the separation of variables method to the diffusion equation:

$$\alpha\nabla^2\varphi - \frac{\partial\varphi}{\partial t} = 0 \qquad \alpha = \text{diffusion coefficient}$$

with appropriate boundary and initial conditions

- Consider a 1D case: an infinite slab of material of thickness 1 in the x-direction (extending from x = 0 to x = 1) and of infinite extent in the y & z directions

$$\alpha\nabla^2\varphi - \frac{\partial\varphi}{\partial t} = 0 \Rightarrow \alpha\left(\frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} + \frac{\partial^2\varphi}{\partial z^2}\right) - \frac{\partial\varphi}{\partial t} = 0 \Rightarrow \alpha\frac{\partial^2\varphi}{\partial x^2} - \frac{\partial\varphi}{\partial t} = 0$$

Boundary and initial conditions:

- Assume that initially the temperature throughout the slab is 1°C

$$\varphi(x, 0) = 1$$

- Assume for t > 0 the slab is placed between two fixed temperature walls

Dirichlet conditions of $\varphi = 0$ on the boundaries x = 0 & x = 1

522

Separation of Variables: Diffusion (1D)

- We wish to solve the following diffusion equation:

$$\alpha \frac{\partial^2 \varphi}{\partial x^2} = \frac{\partial \varphi}{\partial t}$$

- As in the Laplace cases, we begin by separation of variables, i.e. $\varphi = X(x)T(t)$
- We then have: $\alpha T \frac{\partial^2 X}{\partial x^2} = X \frac{\partial T}{\partial t}$
- Dividing both sides by φ gives: $\alpha \frac{1}{X} \frac{\partial^2 X}{\partial x^2} = \frac{1}{T} \frac{\partial T}{\partial t}$
- As before, the LHS is only a function of x, and the RHS only a function of t, so we must have, either:

$$\alpha \frac{1}{X} \frac{\partial^2 X}{\partial x^2} = k^2 \text{ and } \frac{1}{T} \frac{\partial T}{\partial t} = k^2 \quad \text{or} \quad \alpha \frac{1}{X} \frac{\partial^2 X}{\partial x^2} = -k^2 \text{ and } \frac{1}{T} \frac{\partial T}{\partial t} = -k^2$$

523

Separation of Variables: Diffusion (1D)

- So: $\frac{\partial^2 X}{\partial x^2} - \frac{k^2}{\alpha} X = 0 \text{ and } \frac{\partial T}{\partial t} - k^2 T = 0$ or $\frac{\partial^2 X}{\partial x^2} + \frac{k^2}{\alpha} X = 0 \text{ and } \frac{\partial T}{\partial t} + k^2 T = 0$
- These have solutions of the form:

$$X = e^{-\frac{k}{\sqrt{\alpha}}x}, e^{\frac{k}{\sqrt{\alpha}}x} \qquad T = e^{k^2 t} \qquad X = \cos\frac{k}{\sqrt{\alpha}}x, \sin\frac{k}{\sqrt{\alpha}}x \qquad T = e^{-k^2 t}$$

- Hence, the general solution will be of the form:

$$\varphi = \sum_k \left(A_k e^{-\frac{k}{\sqrt{\alpha}}x} + B_k e^{\frac{k}{\sqrt{\alpha}}x} \right) e^{k^2 t} + \sum_k \left(C_k \cos\frac{k}{\sqrt{\alpha}}x + D_k \sin\frac{k}{\sqrt{\alpha}}x \right) e^{-k^2 t}$$

- We expect the solution to remain finite for all t, so: $A_k = B_k = 0$
- Thus our solution must now take the form: $\varphi = \sum_k \left(C_k \cos\frac{k}{\sqrt{\alpha}}x + D_k \sin\frac{k}{\sqrt{\alpha}}x \right) e^{-k^2 t}$

524

Separation of Variables: Diffusion (1D)

$$\varphi(x,t) = \sum_k \left(C_k \cos\frac{k}{\sqrt{\alpha}}x + D_k \sin\frac{k}{\sqrt{\alpha}}x \right) e^{-k^2 t}$$

- We have the following boundary/initial conditions:
  $\varphi(0,t) = 0$ $\varphi(1,t) = 0$ $\varphi(x,0) = 1$
- Taking the boundary conditions first: $\varphi(x = 0, t) = 0$
- $$0 = \sum_k \left(C_k \cos\frac{k}{\sqrt{\alpha}}0 + D_k \sin\frac{k}{\sqrt{\alpha}}0 \right) e^{-k^2 t} \quad \Rightarrow 0 = \sum_k C_k e^{-k^2 t} \Rightarrow C_k = 0$$
$$\varphi(x,t) = \sum_k D_k \sin\frac{k}{\sqrt{\alpha}}x\, e^{-k^2 t}$$

525

Separation of Variables: Diffusion (1D)

$$\varphi(x,t) = \sum_k D_k \sin\frac{k}{\sqrt{\alpha}}x\, e^{-k^2 t}$$

- Taking the other boundary condition: $\varphi(x = 1, t) = 0$

$$0 = \sum_k \left(D_k \sin\frac{k}{\sqrt{\alpha}}1 \right) e^{-k^2 t} \quad \Rightarrow 0 = \sum_k D_k \sin\left(\frac{k}{\sqrt{\alpha}}\right) e^{-k^2 t}$$

$$\Rightarrow \frac{k}{\sqrt{\alpha}} = m\pi, m = 1,2,\ldots \quad \Rightarrow k = m\sqrt{\alpha}\pi, m = 1,2,\ldots$$

$$\varphi(x,t) = \sum_{m=1}^{\infty} D_m \sin(m\pi x)\, e^{-\alpha m^2 \pi^2 t}$$

526

Separation of Variables: Diffusion (1D)

- Thus the solution must take the form: $\varphi = \sum_{m=1}^{\infty} D_m \sin(m\pi x)\, e^{-\alpha m^2 \pi^2 t}$

- Take the initial condition: $\varphi(x,0) = 1$

$$1 = \sum_{m=1}^{\infty} D_m \sin m\pi x$$

$$\sin n\pi x = \sum_{m=1}^{\infty} D_m \sin m\pi x \sin n\pi x$$

$$\int_0^1 \sin n\pi x\, dx = \sum_{m=1}^{\infty} D_m \int_0^1 \sin m\pi x \sin n\pi x\, dx$$

$$\left.\begin{matrix} 2/n\pi \text{ if } n \text{ odd} \\ 0 \text{ if } n \text{ even} \end{matrix}\right\} = \frac{1}{2} D_n \Rightarrow D_n = \frac{4}{n\pi}, n \text{ odd}$$

Recall:

$$\int_0^1 \sin n\pi x\, dx = \frac{1-(-1)^n}{n\pi} = \begin{cases} 2/n\pi \text{ if } n \text{ odd} \\ 0 \text{ if } n \text{ even} \end{cases}$$

$$\int_0^1 (\sin n\pi x \sin m\pi x)\, dx = \begin{cases} \frac{1}{2} & \text{if } m = n \\ 0 & \text{if } m \neq n \end{cases}$$

$$\therefore \varphi = \frac{4}{\pi} \sum_{\substack{n=1 \\ n \text{ odd}}}^{\infty} \frac{1}{n} e^{-\alpha n^2 \pi^2 t} \sin n\pi x$$

6.3 The wave equation

1D

529

The Wave Equation

- Thus far, we have studied the Laplace, Poisson & diffusion equations
- The final equation we study occurs in many areas of engineering: the **wave equation**
- This equation arises in:
  - Acoustics
  - Electromagnetics
  - Small amplitude free surface flows
  - Vibrations
- It has the general form: $\nabla^2\varphi - \frac{1}{c^2}\frac{\partial^2\varphi}{\partial t^2} = 0$ c is the wave speed
- In order for this problem to be well posed (i.e. solvable) we need appropriate boundary & initial conditions:

Boundary conditions: $\varphi = \varphi_D$ **or** $\frac{\partial\varphi}{\partial n} \equiv \mathbf{n}\cdot\nabla\varphi = \varphi'_{\mathrm{N}}$ for $\mathbf{r} \in S, S = \text{boundary of } V$

Initial conditions: $\varphi(\mathbf{r}, t=0) = \varphi_0$ **and** $\frac{\partial\varphi}{\partial t}(\mathbf{r}, t=0) = \varphi'_0$ for $\mathbf{r} \in V$

530

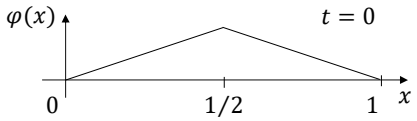
The Wave Equation (1D)

$$\nabla^2\varphi - \frac{1}{c^2}\frac{\partial^2\varphi}{\partial t^2} = 0$$

- Generally, solutions to this equation are complicated - we will consider a simple example in 1 space dimension - a 'plucked' guitar string extending from $x = 0$ to $x = 1$
- Here, $\varphi(x,t)$ represents the displacement of the string from the equilibrium position, so

$$\frac{\partial^2\varphi}{\partial x^2} - \frac{1}{c^2}\frac{\partial^2\varphi}{\partial t^2} = 0$$

- E.g. the guitar string is fixed at each end: $\varphi(x,t) = 0$ at $x = 0, x = 1$ (Boundary conditions)
- Let's assume the guitar string is plucked midway between the end points - before release, the string will have the following shape:



(Initial conditions)

$$\varphi(x,t=0) = \begin{cases} x, & 0 \le x \le \frac{1}{2} \\ 1-x, & \frac{1}{2} \le x \le 1 \end{cases} \quad \textbf{and} \quad \frac{\partial\varphi}{\partial t}(x,t=0) = 0 \text{ for } x \in [0,1]$$

531

The Wave Equation (1D)

- Thus we must solve the following equation $\frac{\partial^2\varphi}{\partial x^2} - \frac{1}{c^2}\frac{\partial^2\varphi}{\partial t^2} = 0$
- We will use separation of variables: assume solutions of the form: $\varphi(x,t) = X(x)T(t)$
- Inserting this into the equation gives (after division by $X(x)T(t)$): $\frac{1}{X}\frac{\partial^2 X}{\partial x^2} = \frac{1}{c^2 T}\frac{\partial^2 T}{\partial t^2}$
- As before, we must then have

$$\frac{\partial^2 X}{\partial x^2} - k^2 X = 0 \quad \text{and} \quad \frac{\partial^2 T}{\partial t^2} - c^2k^2T = 0 \qquad \begin{cases} X = Ae^{-kx} + Be^{kx} \\ T = Ce^{-kct} + De^{kct} \end{cases}$$

or

$$\frac{\partial^2 X}{\partial x^2} + k^2 X = 0 \quad \text{and} \quad \frac{\partial^2 T}{\partial t^2} + c^2k^2T = 0 \qquad \begin{cases} X = E\cos kx + F\sin kx \\ T = G\cos kct + H\sin kct \end{cases}$$

532

The Wave Equation (1D)

- Consider first the solutions: $\begin{cases} X(x) = Ae^{-k} + Be^{kx} \\ T(t) = Ce^{-kct} + De^{kct} \end{cases}$ Called **evanescent** solutions
- These are all exponentials, and never vanish, except maybe at infinity, while we need solutions that vanish $\varphi(x,t) = 0$ at $x = 0$ and $x = 1$

533

The Wave Equation (1D)

- We will thus only consider the remaining solutions: $\begin{cases} X(x) = E\cos kx + F\sin kx \\ T(t) = G\cos kct + H\sin kct \end{cases}$
- Hence the general solution is of the form

$$\varphi(x,t) = \sum_k (E_k \cos kx + F_k \sin kx)(G_k \cos kct + H_k \sin kct)$$

- Impose boundary conditions:

$$x = 0, \varphi = 0 \Rightarrow 0 = \sum_k E_k (G_k \cos kct + H_k \sin kct) \ \forall t \Rightarrow E_k = 0$$

$$\varphi(x,t) = \sum_k F_k \sin kx \, (G_k \cos kct + H_k \sin kct)$$

534

The Wave Equation (1D)

$$\varphi(x,t) = \sum_k F_k \sin kx \,(G_k \cos kct + H_k \sin kct)$$

- Impose 2nd boundary condition:

$$x = 1, \varphi = 0 \Rightarrow 0 = \sum_k F_k \sin k \,(G_k \cos kct + H_k \sin kct) \;\; \forall t \Rightarrow \sin k = 0$$

$$\therefore \sin k = 0 \Rightarrow k = m\pi, m = 1,2,3, \ldots$$

Note: If $F_k = 0$, the whole solution $\varphi(x,t) = 0$ Impossible!

- New solution, obeying boundary conditions:

$$\varphi(x,t) = \sum_{m=1}^{\infty} (g_m \cos m\pi ct + h_m \sin m\pi ct) \sin m\pi x$$

With:
$F_k G_k := g_k$
$F_k H_k := h_k$

535

The Wave Equation (1D)

$$\varphi(x,t) = \sum_{m=1}^{\infty} (g_m \cos m\pi ct + h_m \sin m\pi ct) \sin m\pi x$$

- Impose initial conditions:
  - Note that there are two initial conditions, sufficient to find the 'two' remaining constants
- First initial cond: $\frac{\partial \varphi}{\partial t}(x, t = 0) = 0$ for $x \in [0,1]$
- Hence, from the general solution:

$$\frac{\partial \varphi}{\partial t} = \sum_{m=1}^{\infty} (-g_m m\pi c \sin m\pi ct + h_m m\pi c \cos m\pi ct) \sin m\pi x \;\; \forall x \in [0,1]$$

So at $t = 0$: $0 = \sum_{m=1}^{\infty} (h_m m\pi c) \sin m\pi x \;\; \forall x \in [0,1] \Rightarrow h_m = 0$

$$\varphi(x,t) = \sum_{m=1}^{\infty} g_m \cos m\pi ct \sin m\pi x$$

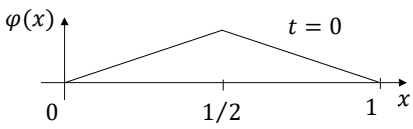
536

The Wave Equation (1D)

$$\varphi(x,t) = \sum_{m=1}^{\infty} g_m \cos m\pi ct \sin m\pi x$$

- The final condition is the initial displacement of the string, i.e.: $\varphi(x, t=0) = \begin{cases} x, & 0 \le x \le 1/2 \\ 1-x, & 1/2 \le x \le 1 \end{cases}$
- Hence, from the general solution:

$$\sum_{m=1}^{\infty} g_m \sin m\pi x = \begin{cases} x, & 0 \le x \le 1/2 \\ 1-x, & 1/2 \le x \le 1 \end{cases}$$



- To determine the g_m, we proceed as before:

Multiply both sides by $\sin n\pi x$: $\sum_{m=1}^{\infty} g_m \sin m\pi x \sin n\pi x = \begin{cases} x \sin n\pi x\,, & 0 \le x \le 1/2 \\ (1-x)\sin n\pi x\,, & 1/2 \le x \le 1 \end{cases}$

Integrate along string: $\sum_{m=1}^{\infty} g_m \int_0^1 \sin m\pi x \sin n\pi x\, dx = \int_0^{1/2} x \sin n\pi x\, dx + \int_{1/2}^1 (1-x)\sin n\pi x\, dx$

537

The Wave Equation (1D)

- We can prove the following results:

Note: Integration by parts

$$\int_0^1 \sin n\pi x \sin m\pi x\, dx = \begin{cases} 1/2 \text{ if } m = n \\ 0 \text{ if } m \ne n \end{cases} \quad \text{and} \quad \int_0^{1/2} x \sin n\pi x\, dx + \int_{1/2}^1 (1-x)\sin n\pi x\, dx = \frac{2}{n^2\pi^2}\sin\frac{n\pi}{2}$$

- Thus:

$$\frac{1}{2} g_n = \frac{2}{n^2\pi^2}\sin\frac{n\pi}{2} \Rightarrow g_n = \frac{4}{n^2\pi^2}\sin\frac{n\pi}{2}$$

- So the final solution is:

$$\varphi(x,t) = \frac{4}{\pi^2}\sum_{n=1}^{\infty} \frac{\sin\left(\frac{n\pi}{2}\right)}{n^2} \sin n\pi x \cos n\pi ct$$

- The first few terms are:

$$\varphi(x,t) = \frac{4}{\pi^2}\left[\cos \pi ct \sin \pi x - \frac{1}{9}\cos 3\pi ct \sin 3\pi x + \frac{1}{25}\cos 5\pi ct \sin 5\pi x + \cdots\right]$$

Harmonics - decreasing in amplitude

538

539

6.4 The use of Fourier series in solving PDEs

For all inhomogeneous boundary and initial conditions

540

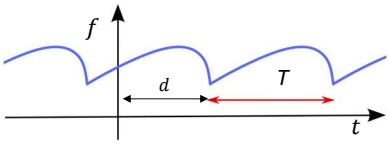
Fourier series applications to PDEs

- You have already used Fourier series without realising in manipulating the inhomogeneous condition
  - When using the orthogonality relation and expressing a constant BC or IC as an infinite sum, you have performed the operation that allows to obtain a_n, b_n terms in the Fourier series (ME1 Maths Lectures and Tutorial Sheet)
- Being aware of the link between PDEs solutions via separation of variables and Fourier series saves time when:
  - An IC is a variable, not a constant
  - A BC is a variable, not a constant
  - You want to verify that the solution obtained fulfils the inhomogeneous condition

541

Fourier theorem - reminder

- Fourier series (and related methods) are most commonly associated with time dependent phenomena, but the theorem and expressions hold for any independent variable
- A function $f(t)$ is said to be **periodic**, with period T if $f(t+T) = f(t)$



- **Fourier theorem** states that such a periodic function $f(t)$ of period T can be expanded as a **Fourier series**, given by

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty}\left(a_n \cos\frac{2n\pi t}{T} + b_n \sin\frac{2n\pi t}{T}\right)$$ where

$$a_n = \frac{2}{T}\int_d^{d+T} f(t)\cos\frac{2n\pi t}{T}\,dt$$

$$b_n = \frac{2}{T}\int_d^{d+T} f(t)\sin\frac{2n\pi t}{T}\,dt$$

542

Fourier theorem - reminder

From your notes from ME1, Autumn Term Part II

- f(x) written as an infinite sum of elementary functions – **basis functions**
- Valid also for functions that are not everywhere continuous (guitar string ICs)
- The cosine and sine basis functions can be identified as necessary for approximating the even and odd parts of a function (usually need both)
  - They are a useful set of basis functions also because of their orthogonality properties
- Although the Fourier Series applies only to periodic functions, it can also be used to approximate non-periodic functions over a finite range of *x* (or *t*), by identifying that range as the period of the function, *L*

543

Fourier Series in solutions to PDEs

- In our study of PDEs we obtained general solutions, to which we subsequently applied boundary and initial conditions
- Consider the diffusion equation in 1D (this is actually where Fourier series were first discovered)
- Using separation of variables, and application of the boundary conditions we arrived at a general solution to the diffusion equation of the form

$$\varphi(x,t) = \sum_{m=1}^{\infty} a_m \sin(m\pi x)\, e^{-\alpha m^2 \pi^2 t}$$

- In the case studied in the lecture, we had an initial condition which stated that $\varphi(x,0) = 1$
- This led us to the constraint that

$$1 = \sum_{m=1}^{\infty} a_m \sin m\pi x$$

- We proceeded by multiplication by sin (or generally cos too) and integrating
- Let's consider this constraint in more detail

544

General *f(x)* or *f(t)* for the inhomogeneous condition

- Fourier was interested in more general initial conditions, say where $\varphi(x,0)$ was a function of *x* (as we assumed for the wave equation)
- That is, suppose that $\varphi(x,0) = f(x)$
- Following the same principles as before, this would lead us to the following constraint:

$$f(x) = \sum_{m=1}^{\infty} a_m \sin m\pi x$$

- Note that this expression says that *f(x)* can be written as an infinite sum of terms of the form $\sin m\pi x$ - we just need to determine what the a_m should be
- As before, we proceed by multiplying both sides by sin or cos (sin in this case) and integrating along the region (from $x = 0$ to 1 in this case), so

This is what we have used for the wave equation example

$$\int_0^1 f(x) \sin n\pi x \, dx = \sum_{m=1}^{\infty} a_m \int_0^1 \sin m\pi x \sin n\pi x \, dx = \frac{1}{2} a_n \Rightarrow a_n = 2\int_0^1 f(x) \sin n\pi x \, dx$$

545

Example: initial condition *f(x)* in diffusion eq.

- Problem sheet – solution obtained for diffusion equation after the information of two Dirichlet homogeneous BCs has been used:

$$T(x,t) = \sum_{m=1}^{\infty} D_m \sin\frac{m\pi x}{L} e^{-\frac{\alpha m^2 \pi^2 t}{L^2}}$$

- Apply BC: $T(x,0) = \sin\frac{\pi x}{L}$ gives $\sin\frac{\pi x}{L} = \sum_{m=1}^{\infty} D_m \sin\frac{m\pi x}{L}$
  - By inspection: $D_1 = 1$ and $D_m = 0$ for all $m \neq 1$
  - Or by 'force':

$$\int_0^L \sin\frac{\pi x}{L}\sin\frac{n\pi x}{L}\mathrm{d}x = \sum_{m=1}^{\infty} D_m \int_0^L \sin\frac{m\pi x}{L}\sin\frac{n\pi x}{L}\mathrm{d}x$$

$$\Rightarrow \begin{cases} n = 1, & L/2 = L/2\, D_1 \\ n \neq 1, & 0 = L/2\, D_n \end{cases} \Rightarrow D_1 = 1, D_n = 0 \text{ for all } n \neq 1$$

- Similarly, for a different initial condition: $T(x,0) = \sin\frac{5\pi x}{L}$

$$\Rightarrow D_5 = 1 \text{ and } D_m = 0 \text{ for all } m \neq 5$$

546

Practised cases – PDEs via separation of variables

Equation	All Dirichlet BCs (&ICs)	One Neumann, the rest Dirichlet	More than 1 inhomogeneous BCs	More than 1 Neumann	Non-constant BCs or ICs f(x)
Laplace cartesian 2D	Lecture		PS #14		
Laplace polar 2D	PS #15	Lecture			
Diffusion 1D (source-free)	Lecture	PS #15		PS #16	PS #15 (IC)
Wave 1D	Lecture	PS #16			Lecture (IC)

547

548

Separation of Variables: Summary & Hints

- Write down the appropriate equation in the appropriate coordinate system
- Write a trial solution in terms of products & insert into the equation

(e.g. $\varphi(x,y) = X(x)Y(y)$)

- Rearrange (usually by dividing both sides appropriately) so that each term depends on **only one** variable
- Separate the terms so that LHS = const. and RHS=const for some constant (often $\pm k^2$)
- Solve each ODE in terms of k - obtain trial solutions
- Write down the general solution as a 'sum' of products of these trial solutions for all possible values of the constant k

549

Separation of Variables: Summary & Hints

- Now apply boundary conditions:
- First consider 'homogeneous' conditions (i.e. where φ = 0):
  - Of these, consider the ones that lie along coordinate axes first
  - Eliminate terms from the general solution
  - At least one condition should provide constraints on k
- Then consider 'inhomogeneous' conditions (e.g. where φ = 1)
  - Substitute boundary conditions into solution and effectively employ Fourier series to obtain the coefficient, e.g.:
  - Multiply both sides of the resulting equation by some sin (or cos) expression so that we can:
  - Evaluate the resulting expressions, applying an 'orthogonality' relations, for example

$$\int_0^1 (\sin n\pi x \sin m\pi x)\, dx = \begin{cases} \frac{1}{2} & \text{if } m = n \\ 0 & \text{if } m \neq n \end{cases}$$

  - Hence obtain an expression for the remaining constant
- Substitute this into the general solution
- **Check your solution fulfills the equation and (at least) the homogeneous conditions**

550

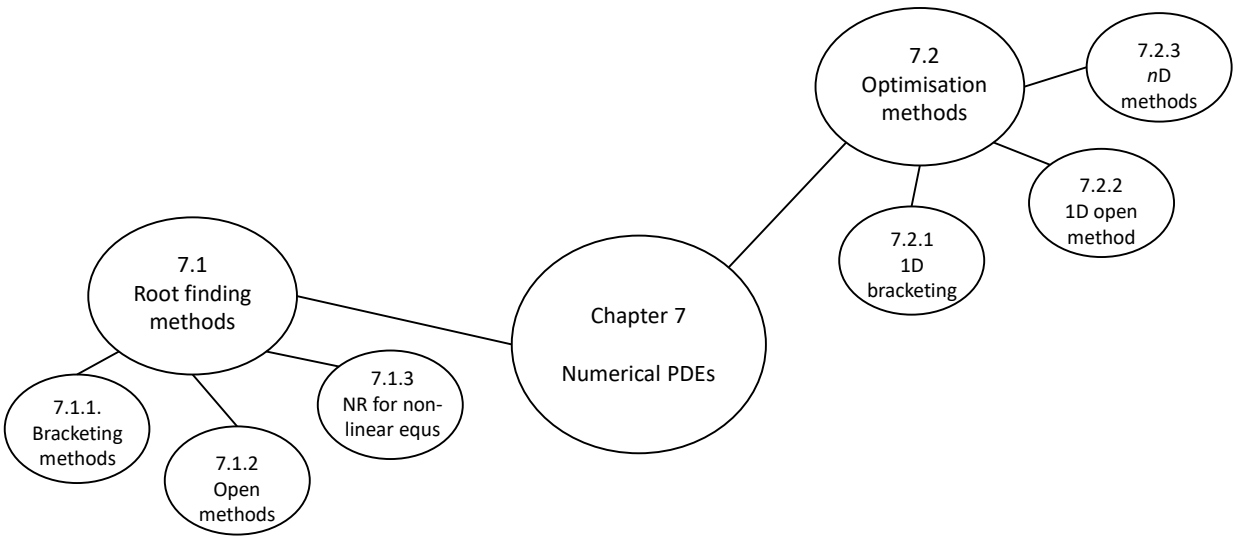
Chapter 7: Numerical methods for root finding and optimisation

Recommended reading:

Numerical methods for engineers, by Chapra & Canale, McGraw Hill Education

Advanced engineering mathematics, by Kreyszig

551



552

Contents

553

7.1 Root finding

Recommended reading:

Numerical methods for engineers, by Chapra & Canale, McGraw Hill Education

Advanced engineering mathematics, by Kreyszig

554

Learning outcomes

- Apply any of the methods presented for calculating iterative estimations of the root of a given function
- Interpret geometrically iterations in any of the methods presented
- Recall cases in which each method fails
- Demonstrate the order of convergence and error bounds of bisection and Newton-Raphson methods
- Given a function, choose the appropriate method in finding its root and justify why it is the optimum method
- Use Newton-Raphson to solve a system of non-linear equations

Methods: bisection, false position
Newton-Raphson, secant

555

Contents

Note: we will not look at methods to find all roots of polynomials, i.e. incl. complex (for this, see Chapra)

https://pixabay.com/photos/roots-tree-nature-227931/

556

Why

The roots x_r of $f(x)$ are the x values for which $f(x_r) = 0$.

E.g. $f(x) = ax^2 + bx + c$ has roots x_{r_1,r_2} given by the quadratic formula.

Most functions do not have a formula for finding out their root(s), i.e. cannot be solved analytically:

- Higher order polynomials
- Transcendental functions (non-algebraic, i.e. containing trigonometric, exp, ln terms)
  e.g. $f(x) = \sin x - x$
- Multivariable functions
- Non-linear functions

$\Rightarrow$ Numerical methods (usually iterative) for finding the *approximate* roots of real-valued functions

There are lots of methods, and others are constantly being developed.

557

Examples - find a parameter from an implicit equation

Any time the equation is implicit, e.g. parachutist's velocity:

$$v(t) = \frac{gm}{c}\left(1 - e^{-\left(\frac{c}{m}\right)t}\right)$$

v = velocity
t = time
g = gravitational constant
m = mass
c = drag coefficient

Equation explicit in v, but implicit in c.

Q: What is the drag coefficient so that a given velocity is reached in a given time?

Meaning: c=? So that $v(t^*) = v^*$, with t^*, v^*given constants

A: $f(c) = \frac{gm}{c}\left(1 - e^{-\left(\frac{c}{m}\right)t}\right) - v(t)$, find root c_r of $f(c)$

558

7.1.1 Bracketing methods

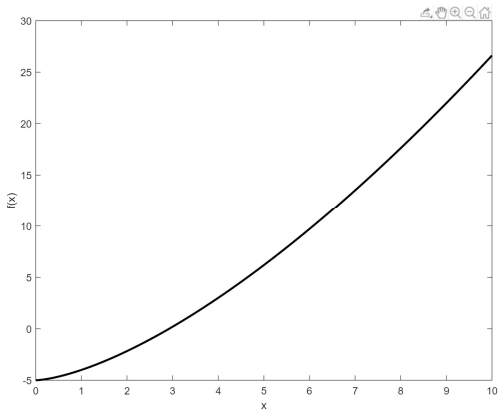
Bisection method

False position method

559

Bisection

Start with an interval in which the root is known to reside (most commonly because the function changes sign), and halve the interval until close enough to the root (i.e. interval becomes small enough).



Example
Q: what is the root of order 1.5 of 5?
Meaning: x = ? Such that $\sqrt[1.5]{5} = x$
⇒ find roots of $f(x) = x^{1.5} - 5$

560

WE 48 – bisection method

$f(x) = x^{1.5} - 5$

- Guess two values of the unknown for which the function changes sign

From plotting, $0 \leq x_r \leq 4$

Can verify that $f(x)$ changes sign:
$f(0) = -5 < 0$
$f(4) = 4^{1.5} - 5 = 3 > 0$

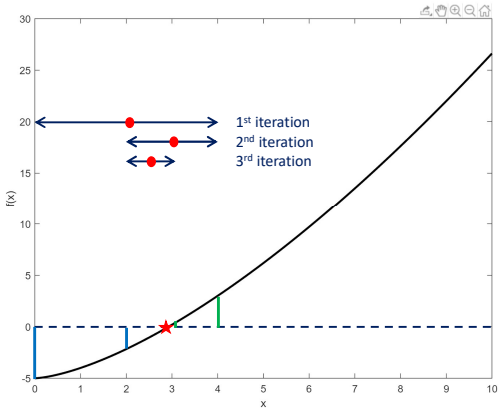
1st estimate: mid-point of $0 \leq x \leq 4$: $x_1 = 2$

Keep the sub-interval in which the function changes sign:
$f(0)f(2) > 0 \Rightarrow$ keep $2 \leq x \leq 4$

2nd estimate: midpoint of $2 \leq x \leq 4$: $x_2 = 3$

$f(2)f(3) < 0 \Rightarrow$ keep $2 \leq x \leq 3$

3rd estimate: midpoint of $2 \leq x \leq 3$: x_3 =2.5



561

Bisection algorithm

Iteration *n*:

- Calculate the midpoint of the current interval

$$x_n = \frac{x_u + x_l}{2}$$

- Assess if convergence is satisfactory
  - If $(x_u - x_n) \leq \varepsilon$ then accept x_n as the root and stop.
- Otherwise: Check the sign of $f(x_n)$ and replace either x_u or x_l by x_n. In code, this might look like:
  - if $\text{sign}[f(x_u)] * \text{sign}[f(x_n)] \leq 0$ then $x_l = x_n$ (computationally safer than testing for the value of the product of functions!)
  - otherwise $x_u = x_n$
- Go to next iteration.

562

Bisection: end condition

- You might be lucky with the selection of the initial interval, such that the bisection point at some iteration *n* coincides with the root itself ($f(x_n) = 0$), but this is rare.
- Usually: iterate the bisection method until the interval containing the root is acceptably small, or the value of the function is acceptably small.

563

Bisection method: convergence and error bounds

- This method is guaranteed to converge to a root (if the function is continuous)

At every iteration $n+1$, $n \geq 2$ we halve the interval:

$$x_{u,n} - x_{l,n} = \frac{1}{2}(x_{u,n-1} - x_{l,n-1})$$

So by iteration

$$x_{u,n} - x_{l,n} = \frac{1}{2^{n-1}}(x_{u,1} - x_{l,1})$$

Where $[x_{l,1}, x_{u,1}]$ is the starting interval.

At iteration n, the root estimate is x_n. The real root x_r lies in either one of the half-intervals $[x_n, x_{u,n}]$ or $[x_{l,n}, x_n]$

$$|x_r - x_n| \leq x_n - x_{l,n} = x_{u,n} - x_n = \frac{1}{2}(x_{u,n} - x_{l,n})$$

564

Bisection method: convergence and error bounds

$$x_{u,n} - x_{l,n} = \frac{1}{2^{n-1}}(x_{u,1} - x_{l,1})$$

$$|x_r - x_n| \leq x_n - x_{l,n} = x_{u,n} - x_n = \frac{1}{2}(x_{u,n} - x_{l,n})$$

⇒ The error bound for x_n:

$$\boxed{\begin{array}{c} |x_r - x_n| \leq 1/2^n \ (x_{u,1} - x_{l,1}) \\ \Rightarrow \ |x_r - x_n| \to 0 \text{ for } n \to \infty \end{array}}$$

The estimate x_n converges to the true root x_r for $n \to \infty$.

Pre-calculate how many iterations are required to reach a given accuracy $|x_r - x_n| \leq \varepsilon$ based on the error bound relation above.(Problem sheet)

565

Bisection method - summary

Advantages: guaranteed convergence, error bound guaranteed to halve with each iteration

Disadvantages: relatively slow convergence (there are faster methods, if $f(x)$ has a continuous derivative), sensitive to the choice of initial interval

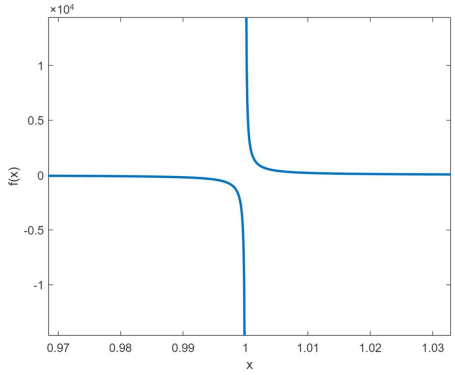
⇒ used in practice only as a way to narrow down onto a good interval, followed by a faster method

Method fails when the function has

- a discontinuity in the interval, NOT a root
  - Bisection would locate the discontinuity and think it is a root
- multiple equal roots
  - Bisection does not detect a root
- multiple roots packed *closely* together
  - Bisection only finds one root

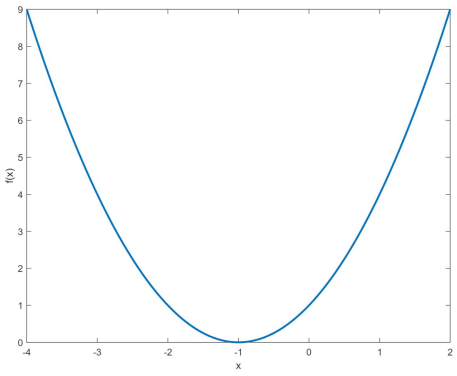
566

Bisection method failing



Function changing sign does not guarantee there is a root in-between

$f(x) = 2(x-1)^{-1}$

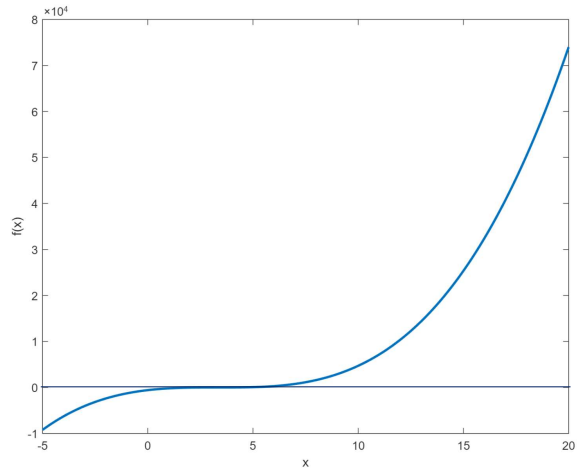


Function has multiple roots without changing sign (even # roots)

$f(x) = x^2 + 2x + 1$

567

Bisection method failing



Multiple roots closely packed together

$f(x) = x(x(16x - 160) + 529) - 578$

568

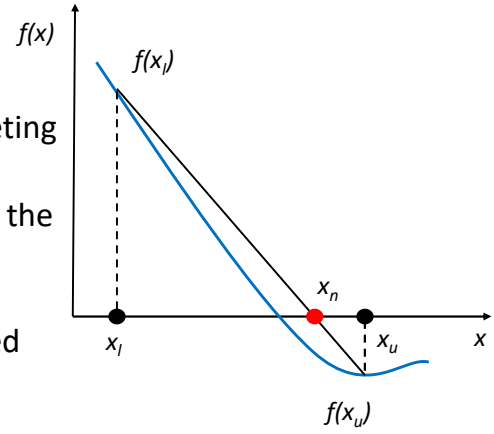
The false position method
(regula falsi, linear interpolation)

Based on graphical insight, an improved bracketing method:

- When dividing the interval, take into account the relative values of $f(x_l)$, $f(x_u)$
- Join the two function values by a line; its intersection with the x-axis forms an improved estimate of the root

$$x_n = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$

- Replace x_l or x_u by x_n to maintain opposite signs



Algorithm: same as bisection, but using the formula on this slide

569

The false position method

Advantages: the interval always brackets the true root, faster convergence than the bisection method

Disadvantages: in particular cases, can converge slower than the bisection method, particularly for functions with a large curvature

⇒ can use a modified false position method (not taught)

570

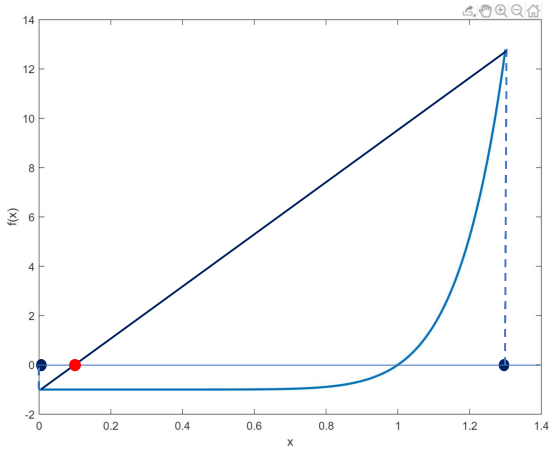
False position method fails

The false position method is not always better than the bisection method

$$f(x) = x^{10} - 1$$

In this case, the real root is NOT closer to the interval end corresponding to the smaller function value.

Always also check: $f(x_n) \leq \varepsilon_f$, where ε_f is a pre-decided value



571

572

7.1.2 Open methods

Newton-Raphson method

Secant method

573

Open methods

- Single starting value of x required (or two, but not required to be on either side of the root)
- As root is not contained within a bracket, method might diverge
- But usually converges faster than bracketing methods

- Most well known: Newton-Raphson
- Other methods in this category: secant

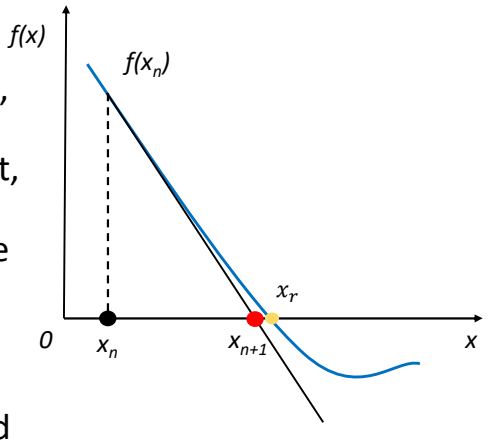
574

The Newton-Raphson method

Starting from an initial guess for the root, x_1, use the value of the function at this point, $f(x_1)$, as well as the function's local gradient, $f'(x_1)$ to construct a tangent line at that point. Gives the x-intercept of this line as the next guess of the root:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- If x_n is near the actual root x_r, this method usually converges



575

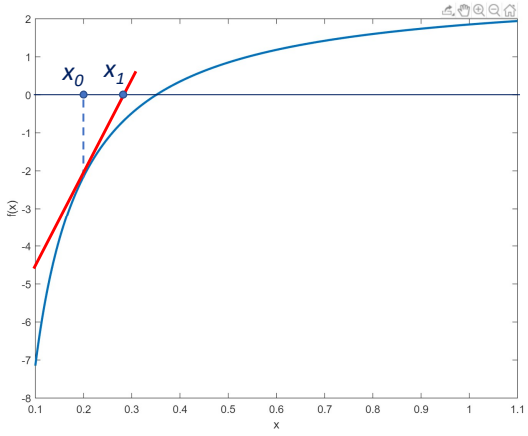
Example: Newton-Raphson method

- Compute $\frac{a}{b}$ as an early computer, i.e. using only addition, subtraction and multiplication: $\frac{a}{b} = a\frac{1}{b}$, where $\frac{1}{b}$ is calculated by Newton-Raphson (assume $b > 0$):

$$\frac{1}{b} = x \Rightarrow f(x) = b - \frac{1}{x}$$

$$f'(x) = \frac{1}{x^2}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \Rightarrow x_{n+1} = x_n(2 - bx_n)$$



576

.

Newton-Raphson: convergence and error bounds

- It can be shown that the relative error of Newton-Raphson is proportional to the square of the previous error
- $f(x_r) = 0, f'(x_r) \neq 0 \Rightarrow f'(x_n) \neq 0$ for x_n in the vicinity of x_r
- Taylor expansion of $f(x_r)$ around $f(x_n)$, the value at iteration n

$$f(x_r) = f(x_n) + (x_r - x_n)f'(x_n) + \frac{1}{2!}(x_r - x_n)^2 f''(\xi)$$

with ξ between x_r and x_n.

$$0 = f(x_n) + (x_r - x_n)f'(x_n) + \frac{1}{2!}(x_r - x_n)^2 f''(\xi) \qquad \left| \times \frac{1}{f'(x_n)} \right.$$

577

Newton-Raphson: convergence and error bounds

$$0 = \underbrace{\frac{f(x_n)}{f'(x_n)} + (x_r - x_n)}_{x_r - x_{n+1}} + \frac{1}{2!}(x_r - x_n)^2 \frac{f''(\xi)}{f'(x_n)}$$

$$\underbrace{x_r - x_{n+1}}_{E_{t,n+1}} = \underbrace{(x_r - x_n)^2}_{E_{t,n}} \frac{-f''(\xi)}{2f'(x_n)} \qquad n \to \infty \Rightarrow \xi, x_n \to x_r$$

$$E_{t,n+1} = E_{t,n}^2 \frac{-f''(\xi)}{2f'(x_n)} \Rightarrow \textit{quadratic convergence}$$

578

Newton-Raphson: convergence and error bounds

$$x_r - x_{n+1} = M(x_r - x_n)^2 \qquad \text{where} \quad M \equiv \frac{-f''(\xi)}{2f'(x_n)}$$

$$M(x_r - x_{n+1}) = [M(x_r - x_0)]^{2^{n+1}}$$

For the method to be convergent, $E_{t,n+1} \to 0$ when $n \to \infty$. The relation above then requires:

$$|M(x_r - x_0)| < 1 \Rightarrow |x_r - x_0| < \frac{1}{|M|}$$

The larger $|M|$, i.e. $f'' \gg f'$ around the root, the closer the initial guess x_0 needs to be to the root, in order for the Newton-Raphson method to converge.

The true relative error is given by: $\varepsilon_{t,n}\,(x_n) = \frac{x_r - x_n}{x_r}$. From the results of the error bound analysis, for $\varepsilon_{t,n}\,(x_n)$ to decrease when $n \to \infty$, then $\left|\varepsilon_{t,0}(x_0)\right| < 1$. Apply this constraint to previous example - what bounds does this impose on x_0?

$$\left|\frac{x_r - x_0}{x_r}\right| < 1 \Rightarrow \left|\frac{1/b - x_0}{1/b}\right| < 1 \Rightarrow -1 < 1 - bx_0 < 1 \Rightarrow 0 < x_0 < \frac{2}{b}$$

579

Newton-Raphson method for multiple roots

- It can be shown that, for functions with *m* multiple roots at location x_r, the Newton-Raphson method is locally convergent only linearly.
- Example: $f(x) = x^m$

$$x_{n+1} = x_n - \frac{x_n^m}{m x_n^{m-1}} = \frac{m-1}{m} x_n$$

So:
$$\lim_{n\to\infty} \frac{\varepsilon_{t,n+1}}{\varepsilon_{t,n}} = \frac{m-1}{m}$$

If you know the multiplicity of the root in advance, you can use a modified Newton-Raphson method with quadratic convergence:

$$x_{n+1} = x_n - m\frac{f(x_n)}{f'(x_n)}$$

580

581

The Newton-Raphson method fails

- Initial (or any encountered) point gives $f'(x_n) = 0$

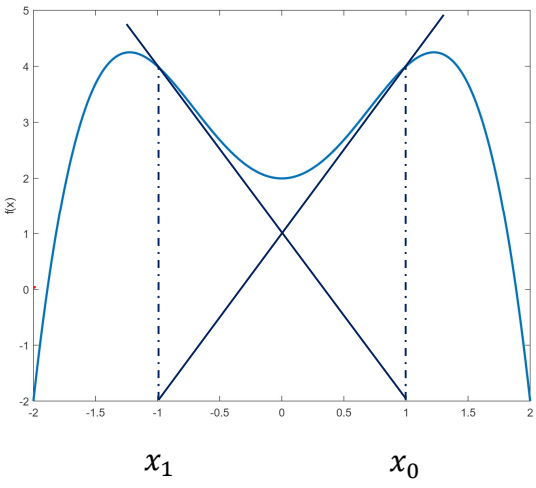
  ⇒ Newton-Raphson method gives $x_{n+1} = \pm\infty$

- Or, more difficult to predict:

  Example: $f(x) = -x^4 + 3x^2 + 2$ with $x_0 = 1$

  $$x_{n+1} = x_n - \frac{-x_n^4 + 3x_n^2 + 2}{-4x_n^3 + 6x_n}$$

  $\Rightarrow x_1 = -1; x_2 = 1; x_3 = -1; \ldots$



582

Secant method

Use the derivative approximated on the spot by 'rise over run', for when the analytical derivative is difficult or costly to calculate.

1. Select two initial guesses x_0 and x_1. They do not need to bracket the true root, i.e. can be on the same side of x_r!
2. Use these two points to construct a *secant,* i.e. draw a line.
3. Take the intersection between the secant and the x-axis as the approximation, x_n.
4. Stop if $f(x_n)$ is close enough to zero.
5. Return to step 2 using the most recent two points, x_n and x_{n-1}.

$$x_{n+1} = x_n - \frac{f(x_n)(x_{n-1} - x_n)}{f(x_{n-1}) - f(x_n)}$$

583

584

7.1.3 Newton-Raphson for systems of non-linear equations

585

Systems of non-linear equations with Newton-Raphson – WE 49

- The problem of solving an $n \times n$ system of non-linear equations can be translated into a root-finding problem for a set of simultaneous equations, which can be solved by the Newton-Raphson method
- For example, we might be interested in finding the pair (x, y) that fulfils the following two equations:

$$\begin{cases} y = x^2 + 1 \\ y = 2\cos x \end{cases}$$

- To transform it into a root problem, the system can be written as:

$$\begin{cases} u(x, y) = x^2 + 1 - y \\ v(x, y) = 2\cos x - y \end{cases}$$

- The pair (x, y) that fulfils the initial system, is also the root of the u and v functions defined in the second system

586

Systems of non-linear equations with Newton-Raphson – WE 49

- Remember the Newton-Raphson method is based on the first order series expansion (Problem sheet #16)

Remember the Taylor series of a function with two independent variables:

$$f(x_{i+1}, y_{i+1}) = f(x_i, y_i) + \left.\frac{\partial f}{\partial x}\right|_{x_i,y_i} (x_{i+1} - x_i) + \left.\frac{\partial f}{\partial y}\right|_{x_i,y_i} (y_{i+1} - y_i) + \frac{1}{2!}\left[\frac{\partial^2 f}{\partial x^2}(x_{i+1} - x_i)^2 + \frac{\partial^2 f}{\partial y^2}(y_{i+1} - y_i)^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x_{i+1} - x_i)(y_{i+1} - y_i)\right] + \cdots$$

- We can use the series to write an approximation of the value of the function at iteration $n+1$ based on its value at iteration n. It is an approximation because we truncate the second and higher order terms.

587

Systems of non-linear equations with Newton-Raphson – WE 49

- With the new notation: $u_n \equiv u(x_n, y_n)$, the first order Taylor series:

$$u_{n+1} \approx u_n + \frac{\partial u_n}{\partial x}(x_{n+1} - x_n) + \frac{\partial u_n}{\partial y}(y_{n+1} - y_n)$$

$$0 \approx u_n + \frac{\partial u_n}{\partial x}(x_{n+1} - x_n) + \frac{\partial u_n}{\partial y}(y_{n+1} - y_n)$$

$$\frac{\partial u_n}{\partial x} x_{n+1} + \frac{\partial u_n}{\partial y} y_{n+1} \approx -u_n + \frac{\partial u_n}{\partial x} x_n + \frac{\partial u_n}{\partial y} y_n$$

Similarly for $v_n \equiv v(x_n, y_n)$

$$\frac{\partial v_n}{\partial x} x_{n+1} + \frac{\partial v_n}{\partial y} y_{n+1} \approx -v_n + \frac{\partial v_n}{\partial x} x_n + \frac{\partial v_n}{\partial y} y_n$$

588

Systems of non-linear equations with Newton-Raphson – WE 49

$u_n \equiv u(x_n, y_n)$
$v_n \equiv v(x_n, y_n)$

$$\underbrace{\begin{pmatrix} \frac{\partial u_n}{\partial x} & \frac{\partial u_n}{\partial y} \\ \frac{\partial v_n}{\partial x} & \frac{\partial v_n}{\partial y} \end{pmatrix}}_{\boldsymbol{J_n}\ \text{(Jacobian)}} \underbrace{\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix}}_{\boldsymbol{x_{n+1}}} = \underbrace{\begin{pmatrix} \frac{\partial u_n}{\partial x} & \frac{\partial u_n}{\partial y} \\ \frac{\partial v_n}{\partial x} & \frac{\partial v_n}{\partial y} \end{pmatrix}}_{\boldsymbol{J_n}\ \text{(Jacobian)}} \underbrace{\begin{pmatrix} x_n \\ y_n \end{pmatrix}}_{\boldsymbol{x_n}} - \underbrace{\begin{pmatrix} u_n \\ v_n \end{pmatrix}}_{\boldsymbol{f_n}}$$

$$\boldsymbol{J}_n \boldsymbol{x}_{n+1} = \boldsymbol{J}_n \boldsymbol{x}_n - \boldsymbol{f}_n \Rightarrow \boldsymbol{x}_{n+1} = \boldsymbol{J}_n^{-1} \boldsymbol{J}_n \boldsymbol{x}_n - \boldsymbol{J}_n^{-1} \boldsymbol{f}_n$$

$$\boxed{\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \boldsymbol{J}_n^{-1} \boldsymbol{f}_n}$$

(Practice in Problem sheet)

589

590

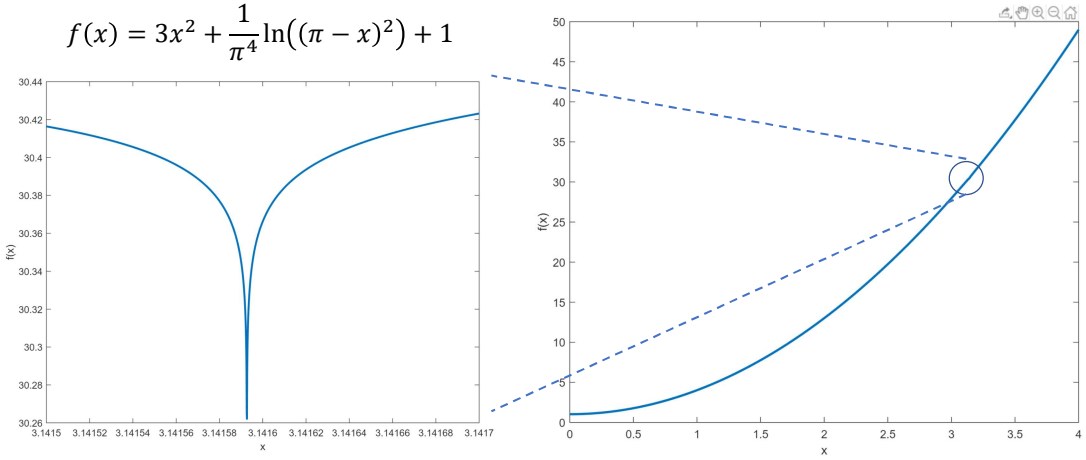
Some ending remarks

- Why is this so hard? – The computer can't see the functions. It only has function values at a few points. You'd find it hard to solve equations with this little information also.
- As with all computational methods – small changes can make your code more efficient
  - If you need a calculated value at a next iteration, store, do not re-calculate it
  - If you have a choice between using an expression with more or fewer operations, choose fewer
- Code in measures to guard against (or at least detect) the cases where the method would fail
  - Verify $f(x_n)$
  - For the false position method check if one boundary is stuck

591

Whatever you do, some things you cannot guard against

$$f(x) = 3x^2 + \frac{1}{\pi^4}\ln\left((\pi - x)^2\right) + 1$$



At $x = \pi$, $f(x)$ dips below 0 in the interval $x = \pi \pm 10^{-667}$. This interval is less than double precision ⇒ You will never find these two roots numerically. There is still need for analytical work 😊

592

Some analytical pre-work

- Try to solve the equation(s) analytically - use Mathematica?
- Graph the equation(s): Matlab, etc., to:
  - Check if continuous, smooth, how differentiable
  - Bracket the ranges where roots are expected
- Linearise the equations and use the matrix methods you learnt to get approximate solutions

7.2 Optimisation methods

Recommended reading:

Numerical methods for engineers, by Chapra & Canale, McGraw Hill Education

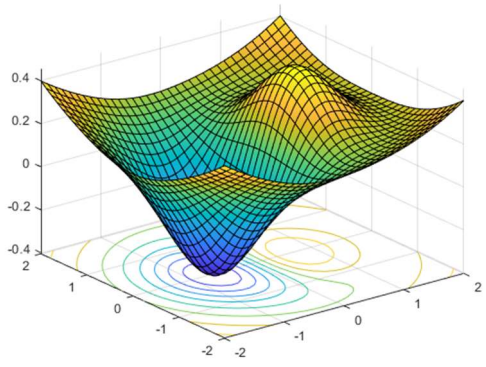
Learning outcomes

- Use the golden-section search and the Newton method to calculate the maximum/minimum of a given function of one variable
- Use the optimal steepest ascent/descent method to calculate the optimum of a function of 2-variables
- Describe the advantages and disadvantages of the taught methods

595

Contents - Unconstrained optimisation (only)

- One-dimensional bracketing methods
  - golden-section search
- One-dimensional open method
  - Newton method
- Multidimensional methods
  - Steepest ascent method



https://uk.mathworks.com/help/optim/ug/optimization-toolbox-tutorial.html

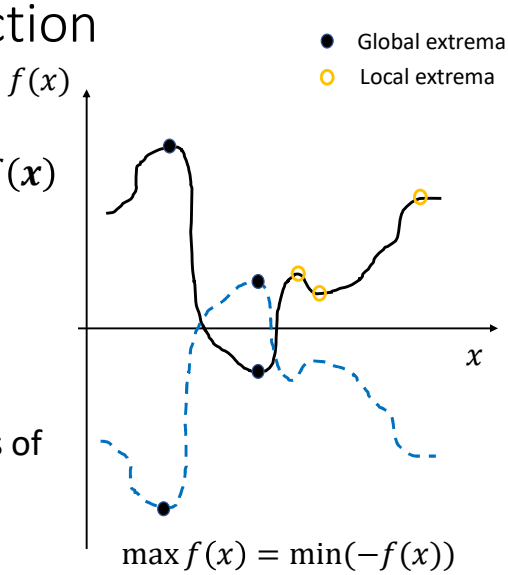
596

Optimisation of a (cost) function

- Find the value of $\boldsymbol{x}$ that yields a global extremum, i.e. maximum or minimum of $f(\boldsymbol{x})$
  - In 1D: $\boldsymbol{x} = x_o$; in 2D: $\boldsymbol{x} = (x_o, y_o)$

$$\Rightarrow x_o \text{ for which } f'(x_o) = 0$$

- $f''(x_o) < 0 \Rightarrow$ max, $f''(x_o) > 0 \Rightarrow$ min
- Mathematically, finding the extrema of $f(\boldsymbol{x})$ can be translated to finding the roots of $f'(\boldsymbol{x})$
- Often, no way to know if local or global



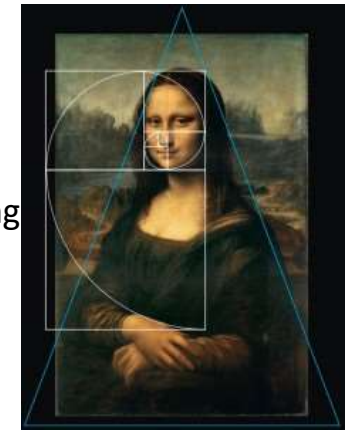
$$\max f(x) = \min(-f(x))$$

597

7.2.1 The golden search method

598

The golden-section search



- The equivalent of the bisection method for root finding
  - Always converging, but possibly slowly
- Technique only valid for single-variable search
- Initial interval chosen to contain a single maximum
- Need 3 values to detect if maximum occurs inside
- Need 4th value to split the interval

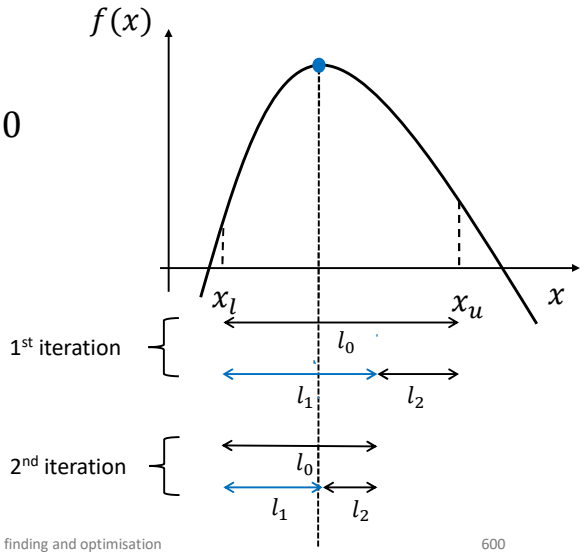
Golden ratio: $\frac{\sqrt{5}-1}{2} = 0.618\ldots$

http://monalisa.org/2012/09/12/leonardo-and-mathematics-in-his-paintings/

The golden section search - concept

$$\begin{cases} l_0 = l_1 + l_2 \\ \dfrac{l_1}{l_0} = \dfrac{l_2}{l_1} \equiv R \end{cases} \Rightarrow R^2 + R - 1 = 0$$

$$\Rightarrow R = \frac{\sqrt{5}-1}{2}$$



The golden section search algorithm to find the maximum of $f(x)$

- Select initial guess for the interval x_l, x_u that brackets a local extremum of $f(x)$
- Choose two interior points, at

$$\begin{cases} x_1 = x_l + R(x_u - x_l) \\ x_2 = x_u - R(x_u - x_l) \end{cases} \qquad R \equiv \frac{\sqrt{5}-1}{2}$$

- Evaluate $f(x_1), f(x_2)$

$$\text{if } f(x_1) > f(x_2) \Rightarrow \text{keep } [x_2, x_u]$$
$$\text{if } f(x_1) < f(x_2) \Rightarrow \text{keep } [x_l, x_1]$$

- Only one function evaluation required @ next step

$$\text{keep } [x_2, x_u] \Rightarrow \begin{cases} \text{new } x_2 = \text{old } x_1 \\ x_1 = x_l + R(x_u - x_l) \end{cases}$$

$$\text{keep } [x_l, x_1] \Rightarrow \begin{cases} \text{new } x_1 = \text{old } x_2 \\ x_2 = x_u - R(x_u - x_l) \end{cases}$$

- To find the minimum of a function $g(x)$, use the property $\min\big(g(x)\big) = \max(-g(x))$, and follow the algorithm above to find $\max(f(x))$, where $f(x) = -g(x)$

601

The golden ratio

- Allows optima to be found efficiently: after n iterations

$$(x_u - x_l)_{\mathrm{n}} = R^{n-1}(x_u - x_l)_1 \cong 0.8\%(x_u - x_l)_0 \text{ for } n = 11$$

- The efficiency comes from performing only a few operations for every iteration
  - This becomes very important in larger systems, where this search is only a part, or
  - for cases where functions are very time consuming to calculate (unlike our simple, toy-functions)

602

7.2.2. Newton method for optimisation

603

Newton's method

- Finding an optimum of $f(x)$ is the same as finding a root of $f'(x)$

$$\underbrace{x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}}_{\text{Root of } f(x)} \quad \Rightarrow \quad \underbrace{x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}}_{\text{Optimum of } f(x)}$$

- Iteration formula can be obtained by
  - defining a function $g(x) = f'(x)$ for which the root is searched, or
  - expanding $f(x_o)$ via the second order Taylor series, and setting $f'(x_o)$ to zero.

604

Newton's method

- Counterpart of Newton-Raphson, with same advantages and disadvantages
- Open method, fast converging, but also possibly diverging, depending on shape of function and initial guess
- Must be able to evaluate the derivative; otherwise can use secant-like alternative
- Can be used after a bracketing method provides a close enough starting point

7.2.3 The steepest ascent method

Multidimensional unconstrained optimisation

- For the purpose of being able to visualise it, we only look at 2D functions $f(x, y)$
- We will look at gradient or descent/ascent methods
  - Use derivative information explicitly to advance to next iteration
- Non-gradient (or direct) approaches also exist,
  - Random search, artificial neural networks, genetic algorithms (used particularly for nonlinear or discontinuous functions)
  - Univariate search (step-wise solve a problem with only one variable, while keeping the others constant) => can use the previous methods

607

Gradient-based algorithms

- Use the local slope to climb down the mountain

Steepest ascent: choose fastest uphill direction, and re-assess after a certain distance

Optimal steepest ascent: optimal distance travelled in a direction is pre-calculated **(only this on the exam)**

Conjugate gradient method: corrected steepest descent

Newton: expand method for one variable to two variables

And many others..

https://www.peakpx.com/455101/rocky-hill-with-thick-clouds

608

The steepest ascent: gradient

We need the equivalent to $f'(x)$ and $f''(x)$ for a multivariate function $f'(\boldsymbol{x})$

- $f'(\boldsymbol{x})$ is given by the gradient, $f''(\boldsymbol{x})$ is given by the Hessian

For $\boldsymbol{x} = (x, y)$ the gradient is:

$$\nabla f = \frac{\partial f}{\partial x}\boldsymbol{i} + \frac{\partial f}{\partial y}\boldsymbol{j}$$

- Gives the path of the steepest ascent, starting from any point $(x, y) = (a, b)$ (not necessarily a path to the summit!)

609

The steepest ascent: gradient

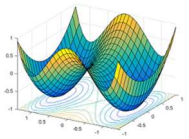
- Graphically, following the gradient means moving orthogonally to contour lines
- If

$$\left.\frac{\partial f}{\partial x}\right|_{x=a,y=b} = 0 \quad \text{and} \quad \left.\frac{\partial f}{\partial y}\right|_{x=a,y=b} = 0$$

then $(x = a, y = b)$ is a two-dimensional optimum

610

The Hessian



$$f(x,y) = x^4 - 2x^2 + y^2$$

- As for the one-dimensional problems, $f''(\boldsymbol{x})$ gives information on the type of optimum:
  - In 1D we had: maximum for $f''(x) < 0$, minimum for $f''(x) > 0$
- In 2D, we take extra care to distinguish between an optimum and a saddle point (partial second derivative with x and y could be both positive or negative, but the point is not an optimum)
- For this, the Hessian of f is defined as:

$$\boldsymbol{H}(x,y) = \begin{pmatrix} \dfrac{\partial^2 f}{\partial x^2} & \dfrac{\partial^2 f}{\partial x \partial y} \\ \dfrac{\partial^2 f}{\partial y \partial x} & \dfrac{\partial^2 f}{\partial y^2} \end{pmatrix} \text{ to distinguish between } \begin{cases} |\boldsymbol{H}| > 0 \;\&\; \partial^2 f/\partial x^2 > 0 \Rightarrow \min \\ |\boldsymbol{H}| > 0 \;\&\; \partial^2 f/\partial x^2 < 0 \Rightarrow \max \\ |\boldsymbol{H}| < 0 \Rightarrow \text{saddle point} \end{cases}$$

Where H is evaluated at the point of interest.

Steepest ascent method

- Primitive concept (to find max):
  - Step 1: Walk a short distance along the steepest ascent (along gradient direction)
  - Step 2: check if at optimum; if not,
  - Step 3: Re-evaluate gradient and go to Step 1
- Problem: how small steps to take? Too computationally inefficient
- Alternate (preferred) method (optimal steepest descent):
  - in Step 1 continue to walk until $f(x,y)$ stops increasing
  - In effect – you are transforming the search to an univariate search of maximum along the direction of the gradient

Optimal steepest ascent method

Essentially, the method searches for the optimum by:

- Find the best direction locally, i.e. move along the gradient
- Find the best distance along that direction, i.e. up to a maximum
- Re-evaluate best direction locally and continue

To find the best distance, we need to transform $f(x, y) \rightarrow g(h)$, where h is the coordinate along the gradient direction

613

Optimal steepest ascent method: $f(x, y) \rightarrow g(h)$

- Choose a starting point (x_0, y_0)
- Calculate the gradient vector ∇f at that point (calculate 1st order partial derivatives)
- Express (x, y) in terms of starting point (x_0, y_0) and partial derivatives along gradient direction:

$$x = x_0 + h\left.\frac{\partial f}{\partial x}\right|_{x_0,y_0} \qquad y = y_0 + h\left.\frac{\partial f}{\partial y}\right|_{x_0,y_0}$$

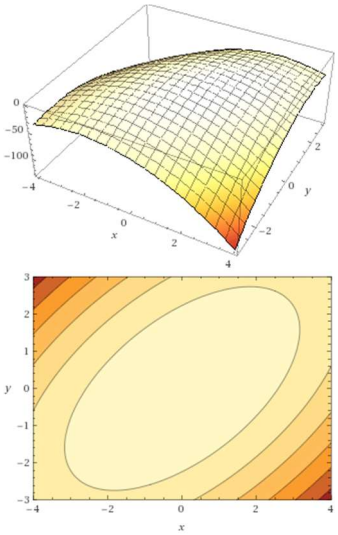
- Substitute the above relations for x, y into $f(x, y)$ to get a new function $g(h)$
- Find the distance to travel along *h* by using univariate methods for maximum finding, or by pre-calculating it (if $g(h)$ is quadratic)

614

WE 50 - optimal steepest ascent

Starting from the initial point $(x_0, y_0) = (3,2)$, find the location of the maximum $f(x_M, y_M)$ of

$$f(x,y) = 4xy - 2x^2 - 4y^2$$



For this simple function, find max analytically:

$$\frac{\partial f}{\partial x} = 4y - 4x \qquad \frac{\partial f}{\partial y} = 4x - 8y$$

$$\begin{cases} 4y - 4x = 0 \\ 4x - 8y = 0 \end{cases} \Rightarrow (x_?, y_?) = (0,0)$$

Beware, we don't know if this location is min, max or saddle

615

To find out which it is, calculate the Hessian determinant:

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{x_?,y_?} = -4; \quad \left.\frac{\partial^2 f}{\partial y^2}\right|_{x_?,y_?} = -8; \quad \left.\frac{\partial^2 f}{\partial x \partial y}\right|_{x_?,y_?} = 4; \quad \left.\frac{\partial^2 f}{\partial y \partial x}\right|_{x_?,y_?} = 4$$

$$H(x_?, y_?) = \begin{pmatrix} \left.\frac{\partial^2 f}{\partial x \partial y}\right|_{x_?,y_?} & \left.\frac{\partial^2 f}{\partial x \partial y}\right|_{x_?,y_?} \\ \left.\frac{\partial^2 f}{\partial y \partial x}\right|_{x_?,y_?} & \left.\frac{\partial^2 f}{\partial y^2}\right|_{x_?,y_?} \end{pmatrix} = \begin{pmatrix} -4 & 4 \\ 4 & -8 \end{pmatrix} \Rightarrow |H(x_?, y_?)| = 16 > 0 \;\&\; \frac{\partial^2 f}{\partial x^2} < 0$$

$\Rightarrow f(0,0) = 0$ is a maximum; could be local or global.

Note: generally $H(x,y)$ will not be a constant when $f(x,y)$ contains higher powers. Its evaluation at $(x_?, y_?)$ gives information about the quality of that location.

616

WE 50: Steepest ascent method – iteration 1

a) Find direction of steepest ascent from $(x_0, y_0) = (3,2) \Rightarrow \nabla f(3,2)$

$$\left.\frac{\partial f}{\partial x}\right|_{x=3,y=2} = 8 - 12 = -4; \qquad \left.\frac{\partial f}{\partial y}\right|_{x=3,y=2} = 12 - 16 = -4$$

So $\nabla f(3,2) = -4\boldsymbol{i} - 4\boldsymbol{j}$. If at this step you obtain $\nabla f(3,2) = 0\boldsymbol{i} - 0\boldsymbol{j}$, then do the Hessian check to determine if at a max. If yes, then done!

b) Convert $f(x, y)$ into $g(h)$, where h points along the direction $\nabla f(3,2)$, with $h = 0$ at (x_0, y_0)

$$g(h) = f\left(x_0 + h\left.\frac{\partial f}{\partial x}\right|_{x_0,y_0}, y_0 + h\left.\frac{\partial f}{\partial y}\right|_{x_0,y_0}\right) = f(3 + h(-4), 2 + h(-4))$$
$$= 4(3 - 4h)(2 - 4h) - 2(3 - 4h)^2 - 4(2 - 4h)^2$$
$$\Rightarrow g(h) = -32h^2 + 32h - 10$$

This function traces the same contour as f, but has only one variable, along $\nabla f(3,2)$

617

c) Find the optimum distance to travel along h, by finding h^* for which

$$g(h^*) = \max(g(h))$$

For this simple problem, we can calculate the maximum analytically. If this were not possible, you would find h^* by one of the single variable optimisation methods (golden search, Newton-Raphson).

$$\frac{\mathrm{d}g}{\mathrm{d}h} = -64h + 32; \left.\frac{\mathrm{d}g}{\mathrm{d}h}\right|_{h^*} = 0 \Rightarrow h^* = \frac{1}{2}$$

Travel along $g(h)$ from $h = 0$ to ½ to reach its maximum.

d) Calculate the coordinates of the new position

$$x_1 = x_0 + h^*\left.\frac{\partial f}{\partial x}\right|_{x_0,y_0} = 3 + \frac{1}{2}(-4) = 1; \qquad y_1 = y_0 + h^*\left.\frac{\partial f}{\partial y}\right|_{x_0,y_0} = 2 + \frac{1}{2}(-4) = 0$$

The elevation **must** have increased as a result of this iteration. Check:

$$f(x_0, y_0) = -10; \qquad f(x_1, y_1) = -2$$

618

WE 50: Steepest ascent method – iteration 2

a) Direction of steepest ascent from new location: $\nabla f(1,0)$

$$\left.\frac{\partial f}{\partial x}\right|_{x=1,y=0} = -4; \qquad \left.\frac{\partial f}{\partial y}\right|_{x=1,y=0} = 4; \qquad \nabla f = -4\boldsymbol{i} + 4\boldsymbol{j}$$

b) Obtain $g(h)$

$$g(h) = f\left(x_1 + h\left.\frac{\partial f}{\partial x}\right|_{x_1,y_1}, y_1 + h\left.\frac{\partial f}{\partial y}\right|_{x_1,y_1}\right) = f\big(1 + h(-4), 0 + h(4)\big)$$

$$= -160h^2 + 32h - 2$$

c) Find h^*, here analytically: $h^* = 1/10$

d) New position: $x_2 = 1 + \frac{1}{10}(-4) = \frac{3}{5}$; $\quad y_2 = 0 + \frac{1}{10}4 = \frac{2}{5}$

Check elevation has increased: $f(x_2, y_2) = -2/5$

WE 50: Steepest ascent method – iteration 3

a) Direction of steepest ascent from new location: $\nabla f\left(\frac{3}{5}, \frac{2}{5}\right)$

$$\nabla f = -\frac{4}{5}\boldsymbol{i} - \frac{4}{5}\boldsymbol{j}$$

b) Obtain $g(h)$

$$g(h) = -\frac{32}{25}h^2 + \frac{32}{25}h - \frac{2}{5}$$

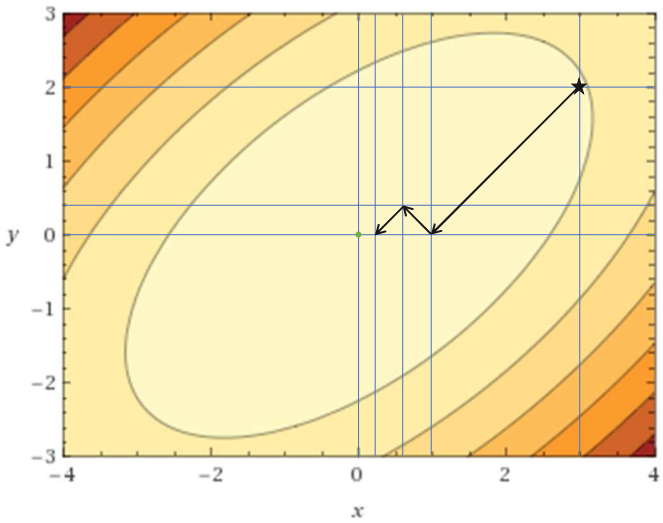
c) Find h^*, here analytically: $h^* = 1/2$

d) New position: $x_3 = 1/5$; $\quad y_3 = 0$

Check elevation has increased: $f(x_3, y_3) = -2/25$ (we are getting closer to the maximum)

WE 50

Trajectory of the path during the first 3 iterations of the steepest ascent method



Function plot via Wolfram alpha: https://www.wolframalpha.com/