

Mathematical methods for continuum modelling

Stephen Millmore

Laboratory for Scientific Computing, University of Cambridge

Mathematical methods for continuum modelling

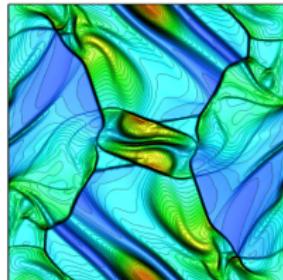
- The continuum stream courses start with systems of partial differential equations, e.g.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + \mathbf{I} p) = 0$$

$$\frac{\partial E}{\partial t} + \nabla \cdot [(E + p) \mathbf{v}] = 0$$

- And finally, we get to simulation results



- From these, numerical methods for their solution are developed

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{\Delta t}{\Delta x} \left(\mathbf{f}_{i+1/2}^{\text{FORCE}} - \mathbf{f}_{i-1/2}^{\text{FORCE}} \right)$$

- In order to derive, and understand the properties of these methods, it is worth reviewing some of the frequently used mathematical techniques

Mathematical methods for continuum modelling

- Throughout the continuum stream courses, you will frequently need to use vector calculus, discrete approximations to continuous variables and simple linear algebra
- Much, if not all, of this is probably familiar, this course is primarily to refresh your memory of these topics
- It may also be useful to set these techniques in the context of numerical simulation applications
- This course is not examined, but all aspects of the material covered could come up in the continuum stream examinations

Outline

- 1 Vector calculus and index notation
- 2 Multivariable calculus and thermodynamics
- 3 Differential and integral equations
- 4 Discretisation and error
- 5 Iterative solution techniques

Outline

- 1 Vector calculus and index notation
- 2 Multivariable calculus and thermodynamics
- 3 Differential and integral equations
- 4 Discretisation and error
- 5 Iterative solution techniques

Vector calculus and evolution equations

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + \mathbf{I} p) = 0$$

$$\frac{\partial E}{\partial t} + \nabla \cdot [(E + p) \mathbf{v}] = 0$$

- The continuum courses consider how a material evolves over time, and it does this by considering the material's properties, as well as the forces on the material
- These equations (the inviscid, compressible Euler equations) model a material's density, momentum and total energy
- With a bit of thermodynamics, this is then enough to completely describe the material (at a continuum scale)

Vector calculus and evolution equations

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

- Whilst these equations will be covered in detail in the continuum courses, we want to motivate why vector calculus keeps coming up
- This is an expression of **conservation of mass**
- The density (mass per unit volume) of a region in space changes according to the **gradient** of the momentum
- As we shall see, this is a limiting case of the mass within a volume being updated by the amount of mass entering a volume minus the amount which leaves
- Perhaps unsurprisingly, because gradients exist in three dimensions, the grad, div and curl operators are frequently encountered in the continuum evolution equations

Vector calculus

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + \mathbf{I} p) = 0$$

$$\frac{\partial E}{\partial t} + \nabla \cdot [(E + p) \mathbf{v}] = 0$$

- Real-world simulations typically cannot be described by a single variable; we need to consider the interaction of multiple variables in a **coupled system**
- The density depends on the momentum, but the momentum itself also depends on density (and pressure)
- It is therefore common to find these systems described in “vector form”

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ E \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + \mathbf{I} p \\ (E + p) \mathbf{v} \end{pmatrix} = 0$$

Vector inside a vector?

- When dealing with coupled continuum systems, the concept of a vector term within a vector is common

$$\begin{pmatrix} \rho \\ \rho\mathbf{v} \\ E \end{pmatrix}$$

- This allows for a **coordinate independent**, if mathematically dodgy, description of quantities such as velocity, which are vectors themselves
- The equation on the previous slide holds, regardless of whether

$$\mathbf{v} = v_x \mathbf{e}_i + v_y \mathbf{e}_j + v_z \mathbf{e}_k \quad \text{or} \quad \mathbf{v} = v_r \mathbf{e}_r + v_\theta \mathbf{e}_\theta + v_\phi \mathbf{e}_\phi$$

- Once you know your coordinate system, you can simply use e.g.

$$\begin{pmatrix} \rho \\ \rho\mathbf{v} \\ E \end{pmatrix} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ E \end{pmatrix}$$

Vector operations

- From now, unless otherwise stated, we shall assume we are working in Cartesian coordinates (which will be the case throughout the continuum courses)
- In these coordinates, grad, div and curl are familiar

$$\nabla \alpha = \begin{pmatrix} \frac{\partial \alpha}{\partial x} \\ \frac{\partial \alpha}{\partial y} \\ \frac{\partial \alpha}{\partial z} \end{pmatrix}$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) + \nabla p = 0$$

- Momentum is affected by pressure **normal** in the normal direction
- For example, a pressure gradient in the x -direction will affect the x -momentum (and x -velocity) of a material

Vector operations

- From now, unless otherwise stated, we shall assume we are working in Cartesian coordinates (which will be the case throughout the continuum courses)
- In these coordinates, grad, div and curl are familiar

$$\nabla \cdot \mathbf{a} = \frac{\partial a_x}{\partial x} + \frac{\partial a_y}{\partial y} + \frac{\partial a_z}{\partial z}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

- To calculate the change in mass in a volume, we need to know how much enters or leaves in all directions
- In other words we need to sum up the gradients at all edges of the volume

Vector operations

- From now, unless otherwise stated, we shall assume we are working in Cartesian coordinates (which will be the case throughout the continuum courses)
- In these coordinates, grad, div and curl are familiar

$$\nabla \times \mathbf{a} = \begin{vmatrix} \mathbf{e}_i & \mathbf{e}_j & \mathbf{e}_k \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ a_x & a_y & a_z \end{vmatrix} = \left(\begin{array}{c} \frac{\partial a_z}{\partial y} - \frac{\partial a_y}{\partial z} \\ \frac{\partial a_x}{\partial z} - \frac{\partial a_z}{\partial x} \\ \frac{\partial a_y}{\partial x} - \frac{\partial a_x}{\partial y} \end{array} \right)$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = 0$$

- Electric and magnetic fields act perpendicularly to each other
- So a gradient in the x -direction of an electric field will cause evolution of the y - and z -components of the magnetic field

More vector operations

- Sometimes less familiar operations may appear
- For example, the outer product, \otimes , and the divergence of a matrix (tensor)

$$\mathbf{a} \otimes \mathbf{b} = \begin{pmatrix} a_x b_x & a_x b_y & a_x b_z \\ a_y b_x & a_y b_y & a_y b_z \\ a_z b_x & a_z b_y & a_z b_z \end{pmatrix}$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + \mathbf{I} p) = \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{f}$$

- A quantity moving in one direction can still be affected by quantities in other directions
- The components of the quantity $\rho \mathbf{v} \otimes \mathbf{v}$ are, for example, the speed with which the x -momentum is moving in the x -direction, and in the y -direction, z -direction and so on
- Some papers are lazy, and write this as $\rho \mathbf{v} \mathbf{v}$ (because it is technically the multiplication of two vectors)

More vector operations

- Sometimes less familiar operations may appear
- For example, the outer product, \otimes , and the divergence of a matrix (tensor)

$$\nabla \cdot \mathbf{A} = \left(\begin{array}{l} \frac{\partial A_{xx}}{\partial x} + \frac{\partial A_{yx}}{\partial y} + \frac{\partial A_{zx}}{\partial z} \\ \frac{\partial A_{xy}}{\partial x} + \frac{\partial A_{yy}}{\partial y} + \frac{\partial A_{zy}}{\partial z} \\ \frac{\partial A_{xz}}{\partial x} + \frac{\partial A_{yz}}{\partial y} + \frac{\partial A_{zz}}{\partial z} \end{array} \right)$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + \mathbf{I} p) = \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{f}$$

- Some quantities, like stress, have different behaviour dependent on direction (e.g. shearing vs stretching)
- The directions this can occur in get complicated (diagonal movement causes both shearing and stretching in all directions) - matrices (or tensors) are good descriptions here
- And the rate of change of these variables is still important

More vector operations

- Sometimes things get even worse; some papers include quantities such as \mathbf{AB} where \mathbf{A} and \mathbf{B} are both matrices, as is \mathbf{AB}

$$\mathbf{AB} = \begin{pmatrix} A_{xx}B_{xx} + A_{xy}B_{yx} + A_{xz}B_{zx} & A_{xx}B_{xy} + A_{xy}B_{yy} + A_{xz}B_{zy} & A_{xx}B_{xz} + A_{xy}B_{yz} + A_{xz}B_{zz} \\ A_{yx}B_{xx} + A_{yy}B_{yx} + A_{yz}B_{zx} & A_{yx}B_{xy} + A_{yy}B_{yy} + A_{yz}B_{zy} & A_{yx}B_{xz} + A_{yy}B_{yz} + A_{yz}B_{zz} \\ A_{zx}B_{xx} + A_{zy}B_{yx} + A_{zz}B_{zx} & A_{zx}B_{xy} + A_{zy}B_{yy} + A_{zz}B_{zy} & A_{zx}B_{xz} + A_{zy}B_{yz} + A_{zz}B_{zz} \end{pmatrix}$$

- It is not always clear whether \mathbf{A} and \mathbf{B} are matrices, let alone which entries are expected to 'match up'
- Fortunately, there is a better way to express these quantities

Index notation

- Some of these quantities are not natural to work with, for example, we may want to write $\nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v})$ as partial derivatives of just \mathbf{v}
- Additionally, for some cases, such as elasticity, we end up with rank-4 tensors (like a matrix, but in four dimensions...)
- In such cases, **index notation** makes life much more straightforward
- For example,

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + \mathbf{I} p) = \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{f}$$

becomes

$$\frac{\partial \rho v_i}{\partial t} + \nabla_j (\rho v_i v_j + \delta_{ij} p) = \nabla_j \tau_{ij} + \rho f_i$$

Reading index notation

$$\frac{\partial \rho v_i}{\partial t} + \nabla_j (\rho v_i v_j + \delta_{ij} p) = \nabla_j \tau_{ij} + \rho f_i$$

- The indices tell you what you are dealing with
- Scalars have no indices, e.g. ρ and p
- Vectors have one index, e.g.

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = v_i,$$

- In Cartesian coordinates, the $(1, 2, 3)$ and (x, y, z) subscripts are often used to mean the same thing $(v_1, v_2, v_3)^T = (v_x, v_y, v_z)^T$

Reading index notation

$$\frac{\partial \rho v_i}{\partial t} + \nabla_j (\rho v_i v_j + \delta_{ij} p) = \nabla_j \tau_{ij} + \rho f_i$$

- Matrices have two indices, the first gives the row, and the second gives the column

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{pmatrix} = \begin{pmatrix} \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \tau_{33} \end{pmatrix} = \tau_{ij}$$

$$\boldsymbol{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \delta_{11} & \delta_{12} & \delta_{13} \\ \delta_{21} & \delta_{22} & \delta_{23} \\ \delta_{31} & \delta_{32} & \delta_{33} \end{pmatrix} = \delta_{ij}$$

- When written in index notation, the identity matrix is typically called the **Kronecker delta**

Combined quantities

$$\frac{\partial \rho v_i}{\partial t} + \nabla_j (\rho v_i v_j + \delta_{ij} p) = \nabla_j \tau_{ij} + \rho f_i$$

- As might be expected, everything separated by addition, subtraction or an equals sign must be the same **rank**, i.e. the same type of thing
- Probably an obvious statement (you can't add a vector to a matrix) but helpful when checking your calculations
- Similarly, everything within a set of parentheses must be the same thing
- So here, both $\rho v_i v_j$ and $\delta_{ij} p$ have two indices and both are matrices
- $\delta_{ij} p$ is a scalar multiplied by a matrix; we clearly expect this to be a matrix
- As is the outer product quantity, $\rho v_i v_j$, it is now very clear to see what the entries of this matrix should be

$$\frac{\partial \rho v_i}{\partial t} + \nabla_j (\rho v_i v_j + \delta_{ij} p) = \nabla_j \tau_{ij} + \rho f_i$$

- We have so far avoided talking about what the $\nabla_j \tau_{ij}$ term is
- It should be a vector, since $\frac{\partial \rho v_i}{\partial t}$ and ρf_i are vectors
- It also has one index repeated (j) and one which only appears once (i)
- A free index is one that appears **only once** per term
- This is what determines what type of quantity you have
- Repeated indices represent a summation

Repeated indices

- Repeated indices represent a sum over that index - **Einstein summation convention**

$$v_i v_i = \sum_{i=1}^3 v_i v_i = v_1 v_1 + v_2 v_2 + v_3 v_3 = \mathbf{v} \cdot \mathbf{v} = v^2$$

- Used for simplifying notation
- This index can be repeated in two different quantities, such as the dot product above
- Or a repeated index within a single quantity, as in Hooke's law for stress and strain

$$\sigma_{ij} = \lambda \delta_{ij} e_{kk} + 2\mu e_{ij}$$

- Here,

$$e_{kk} = e_{11} + e_{22} + e_{33} = \text{Tr}(\mathbf{e})$$

Back to grad, div and curl

- The grad operator, ∇ , is a vector operator, so can be written with index notation

$$\nabla = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} \rightarrow \nabla_i = \frac{\partial}{\partial x_i}$$

- Grad and div are now straightforward,

$$\nabla \alpha \rightarrow \nabla_i \alpha, \quad \nabla \cdot \mathbf{a} \rightarrow \nabla_i a_i$$

- Curl requires the addition of a new quantity, the Levi-Civita symbol

$$\nabla \times \mathbf{a} \rightarrow \epsilon_{ijk} \nabla_j a_k, \quad \epsilon_{ijk} = \begin{cases} +1 & \text{if } (i, j, k) \text{ is } (1, 2, 3), (2, 3, 1), \text{ or } (3, 1, 2), \\ -1 & \text{if } (i, j, k) \text{ is } (3, 2, 1), (1, 3, 2), \text{ or } (2, 1, 3), \\ 0 & \text{if } i = j, \text{ or } j = k, \text{ or } k = i \end{cases}$$

Levi-Civita symbol

$$\nabla \times \mathbf{a} = \epsilon_{ijk} \nabla_j a_k$$

- There is one free index in $\epsilon_{ijk} \nabla_j a_k$, so as expected, this is a vector term
- The term $\nabla_j a_k$ is a matrix and the Levi-Civita symbol a '3-D matrix'

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \quad \nabla_j a_k = \begin{pmatrix} \frac{\partial a_x}{\partial x} & \frac{\partial a_x}{\partial y} & \frac{\partial a_x}{\partial z} \\ \frac{\partial a_y}{\partial x} & \frac{\partial a_y}{\partial y} & \frac{\partial a_y}{\partial z} \\ \frac{\partial a_z}{\partial x} & \frac{\partial a_z}{\partial y} & \frac{\partial a_z}{\partial z} \end{pmatrix}$$

- The quantity $\epsilon_{ijk} \nabla_j a_k$ then multiplies each 'slice' of ϵ_{ijk} with $\nabla_j a_k$
- Also note that any cross product is written in index notation with the Levi-Civita symbol,

$$\mathbf{a} \times \mathbf{b} = \epsilon_{ijk} a_j b_k$$

Vector calculus in index notation

- Working in index notation has a few advantages
- Firstly, notation for matrices, and 'higher dimensional matrices' is straightforward, since there is no universal convention for these quantities
- Simplification of expressions is also more straightforward - there is no need to memorise any relationships
- Well... maybe one relationship

$$\epsilon_{ijk}\epsilon_{ilm} = \delta_{jl}\delta_{km} - \delta_{jm}\delta_{kl}$$

- Additionally, dealing with derivatives is a lot easier, as in index notation, you are treating things at the level of scalar quantities

Changing indices with the Kronecker delta

- One of the key features of vector calculus operations in index notation is the use of the Kronecker delta to change indices
- Sometimes this is obvious

$$\delta_{ij}a_j = a_i$$

- This is really just saying that multiplying a vector by the identity matrix does not change the vector - which is the definition of the identity matrix!
- Sometimes it takes a bit more thought to work out what is happening

$$\delta_{ij}A_{ij} = A_{ii}, \quad \delta_{im}\delta_{jn}B_{mnkl} = B_{ijkl}$$

- And occasionally, we work backwards

$$\nabla_j \rho v_i v_j + \nabla_i p = \nabla_j (\rho v_i v_j + \delta_{ij} p)$$

Some examples

- All ‘classic’ vector calculus identities can be demonstrated using index notation

$$\nabla \cdot (\mathbf{a} \times \mathbf{b}) = (\nabla \times \mathbf{a}) \cdot \mathbf{b} - \mathbf{a} \cdot (\nabla \times \mathbf{b})$$

- Throughout the continuum courses, we will make use of index notation for expressing equations in ‘desired’ forms
- If we want to solve the Maxwell-Faraday equation (with a few other coupled equations, and making use of Ohm’s law),

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = 0, \quad \eta \mathbf{J} = \mathbf{E} + \mathbf{v} \times \mathbf{B}, \quad \eta = 0$$

- Then it is often desirable to express all spatial derivatives through a divergence operator

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{B} \otimes \mathbf{v} - \mathbf{v} \otimes \mathbf{B}) = 0$$

A disclaimer

- Technically, index notation does not just use subscripts, but also superscripts

$$a^i = \begin{pmatrix} a^x \\ a^y \\ a^z \end{pmatrix}, \quad a_i = (a_x, a_y, a_z)$$

- Because we tend to focus on Cartesian coordinates, we ignore this description (we have $a^x = a_x$ and so on)
- But, in more general coordinate systems, **this is not true**, and there is a transformation required to move between the two forms
- We rarely encounter this in this MPhil; we even write non-Cartesian coordinates in a 'Cartesian-like' manner
- Fusion plasma equilibrium literature is a notable exception - non-orthogonal coordinate systems are common

Tensors

- Using index notation makes it easy to write down things like a 'three-dimensional matrix' in a straightforward way
- And when dealing with these sort of quantities, it is common to refer to them as tensors
- Though, in reality, tensors have a slightly stricter definition
- This starts with coordinate systems, which are defined by **basis vectors**

$$\mathbf{a} = a_1 \mathbf{e}_1 + a_2 \mathbf{e}_2 + a_3 \mathbf{e}_3 \quad (1.1)$$

$$\mathbf{a} = a'_1 \mathbf{e}'_1 + a'_2 \mathbf{e}'_2 + a'_3 \mathbf{e}'_3 \quad (1.2)$$

- For example, we may have Cartesian basis vectors, \mathbf{e}_x , \mathbf{e}_y , \mathbf{e}_z (often written \mathbf{i} , \mathbf{j} , \mathbf{k}), or spherical polar basis vectors, \mathbf{e}_r , \mathbf{e}_ϕ , \mathbf{e}_θ
- Coordinates transform through

$$\mathbf{e} = \frac{\partial \mathbf{e}}{\partial \mathbf{e}'} \mathbf{e}' \quad \text{or} \quad e_i = R_{ij} e'_j$$

where R_{ij} , or $\partial \mathbf{e} / \partial \mathbf{e}'$, is the Jacobian

Why do we care about tensors?

- A tensor is a quantity that also transforms according to these transformation rules, e.g.

$$t_{ij} = R_{ik} R_{jl} t'_{kl}$$

or

$$t_{ijkl} = R_{im} R_{jn} R_{kp} R_{lq} t'_{mnpq}$$

- The number of indices is also known as its **rank**
- One of the key aspects of a tensor is that the quantity represents the same thing, **no matter what coordinate system we are working in** (though individual entries will, of course, be different)
- This is important if you want consistent definitions of quantities such as stress, or elastic deformation
- We see this importance in **invariant quantities**

Invariant quantities

- Invariant quantities are those which are the same, no matter the coordinate system
- For a rank-2 tensor, the invariant quantities come from finding its eigenvalues

$$\begin{vmatrix} a_{11} - \lambda & a_{12} & a_{13} \\ a_{21} & a_{22} - \lambda & a_{23} \\ a_{31} & a_{32} & a_{33} - \lambda \end{vmatrix} = 0$$

- This determinant can be written

$$\lambda^3 - I_1\lambda^2 + I_2\lambda - I_3 = 0$$

$$I_1 = a_{ii} = a_{11} + a_{22} + a_{33} = \text{tr}(\mathbf{a})$$

$$I_2 = \frac{1}{2} (a_{ii}a_{jj} - a_{ij}a_{ji}) = \frac{1}{2} [(\text{tr}(\mathbf{a}))^2 - \text{tr}(\mathbf{a}^2)]$$

$$I_3 = \det |a_{ij}|$$

- I_1 , I_2 and I_3 are the **invariants** of a rank-2 tensor

Invariants, pressure and stress

- When modelling any continuum material, stress (σ_{ij}) is a rank-2 tensor which describes how the material is being pushed (or pulled) in any direction
- Pressure (p) is a scalar quantity, which gives some information about how much stress the material is under
- Perhaps unsurprisingly, the pressure experienced at any point within a material does not depend on the coordinate system used to calculate it - it is an invariant quantity
- In fact,

$$p = \frac{1}{3}I_1 = \frac{1}{3}\text{tr}(\sigma)$$

Matrices which aren't tensors

- There are two commonly encountered quantities which are matrices, but not tensors
- The first is the coordinate transformation, R_{ij} ,

$$\mathbf{e} = \frac{\partial \mathbf{e}}{\partial \mathbf{e}'} \mathbf{e}' \quad \text{or} \quad e_i = R_{ij} e'_j$$

- The second is the **deformation gradient**, F_{ij}

$$F_{ij} = \frac{dx_i}{dX_j}$$

- This transforms an infinitesimal particle from its reference configuration $d\mathbf{X}$ to its current position dx , by deformation alone (no rotation or translation)

What are these matrices

- Both these examples take one vector (e' or X), and transform them to a new vector (e or x)
- These quantities **transform as vectors**, i.e. would need one coordinate transform, whilst a regular rank-2 tensor would require two
- On inspection, perhaps not surprising - one of the indices is already in the coordinate system we are transforming to
- Such quantities are known as **two-point tensors** or **double vectors**
- This name does not always make it to the literature though...

Outline

- 1 Vector calculus and index notation
- 2 Multivariable calculus and thermodynamics
- 3 Differential and integral equations
- 4 Discretisation and error
- 5 Iterative solution techniques

Coupled systems of equations

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + \mathbf{I} p) = 0$$

$$\frac{\partial E}{\partial t} + \nabla \cdot [(E + p) \mathbf{v}] = 0$$

- The description of the evolution of continuum materials is frequently achieved through **coupled equations**
- The equations for evolving density, ρ , momentum, $\rho \mathbf{v}$, and total energy, E cannot be separated into three independent equations (in general)
- For a closed system of equations, we also need to define velocity, \mathbf{v} , and pressure, p , in terms of these evolved variables

Coupled systems of equations

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + \mathbf{I} p) = 0$$

$$\frac{\partial E}{\partial t} + \nabla \cdot [(E + p) \mathbf{v}] = 0$$

$$\mathbf{v} = \frac{\rho \mathbf{v}}{\rho}, \quad p = p(\rho, \varepsilon), \quad E = \rho \varepsilon + \frac{1}{2} \rho v_i v_i$$

- Sometimes this is obvious (velocity), sometimes not (pressure - more on this later)
- Note that here, ρ , $\rho \mathbf{v}$ and E are all being treated as **independent variables**
- We could use other variables (e.g. ρ , \mathbf{v} and p), such choices are often dependent on our application

Closure and derivatives

$$\mathbf{v} = \frac{\rho \mathbf{v}}{\rho}, \quad p = p(\rho, \varepsilon), \quad E = \rho \varepsilon + \frac{1}{2} \rho v_i v_i$$

- When we need to manipulate these equations, we frequently find ourselves taking derivatives with respect to other variables
- Before doing this, we need to be sure which independent variables we have
- For example, in the situation with ρ , $\rho \mathbf{v}$ and E being the independent variables,

$$\frac{\partial E}{\partial(\rho v_x)} = \left. \frac{\partial E}{\partial(\rho v_x)} \right|_{\rho, \rho v_x, \rho v_y = \text{const}} = 0$$

- Another parameterisation might use ρ , ε and $\rho \mathbf{v}$ as independent variables

$$\frac{\partial E}{\partial(\rho v_x)} = \left. \frac{\partial E}{\partial(\rho v_x)} \right|_{\rho, \varepsilon, \rho v_y, \rho v_z = \text{const}} = v_x$$

Thermodynamics and closure

- Systems of equations such as the compressible Euler equations are derived by considering conservation of mass, momentum and energy
- Closure of these equations (i.e. how $p = p(\rho, \varepsilon)$ is actually defined) is typically provided by the laws of thermodynamics
- We have the **first law of thermodynamics**

$$dQ = dU + pdV$$

where Q is heat, U is internal energy and V is volume

- And the **second law of thermodynamics**

$$dU = -pdV + TdS$$

where T is temperature and S is entropy

- We shall see these laws being used to derive closure conditions (equations of state) for specific materials in “Simulation of Matter under Extreme Conditions”

Thermodynamic quantities

$$dQ = dU + pdV, \quad dU = -pdV + TdS$$

- The laws of thermodynamics may seem to introduce many new variables, rather than close the system
- In particular, we have introduced two very unnatural quantities to work with, heat and entropy
- We shall revisit entropy, in context, at various points throughout the continuum courses
- For now, it is a quantity that can be computed using the second law of thermodynamics, and not needed to close the system
- Heat, however, is something we need to take care of - typically through known physical quantities - the **specific heats**

Specific heat capacity

- Specific heat capacity is a measure of the amount of heat required to raise the temperature of a material (measured per unit mass)
- This is dependent on how the heat is added, two typical specific heats are considered
- **Specific heat at constant volume**, c_V , considers the amount of heat required if the volume of material does not change whilst temperature rises (but the pressure, or stress, within the material does)
- **Specific heat at constant pressure**, c_p , considers the amount of heat required if the pressure (or stress) does not change, but instead temperature rise causes volume to change

Specific heat capacity

$$c_V = \frac{1}{M} \left. \frac{dQ}{dT} \right|_V, \quad c_p = \frac{1}{M} \left. \frac{dQ}{dT} \right|_p$$

- Typically, actual specific heats are calculated either from first-principles atomistic calculation, or experimentally
- Either way, in a continuum course, we have to assume we can look them up for our materials of interest
- But if we know c_V , we can show

$$U = Mc_V T$$

- We can relate energy to temperature without worrying about heat (or, technically, hiding it in c_V)

Volumetric quantities

$$U = Mc_V T$$

- Many of the laws of thermodynamics we have considered work with mass, M , and volume, V
- When dealing with continuum systems, typically we work with quantities per unit volume, e.g. density measured in kg/m^3 , or quantities per unit mass, e.g. specific internal energy measured in J/kg
- For example, we have total internal energy, U , internal energy, $e = U/V$, or specific internal energy, $\varepsilon = U/(\rho V)$
- It is common to then see the equivalent relationship

$$\varepsilon = c_V T$$

Dimensional analysis

- Different quantities can get confusing, especially with things like total entropy, S , or entropy per unit mass, s
- The literature cannot be guaranteed to be consistent
- When in doubt, dimensional analysis can help a lot
- The units of total internal energy are Joules,

$$[U] = \text{J} = \text{kg m}^2 \text{s}^{-2}$$

- We can confirm that this is the quantity given by $U = M c_V T$, which either have obvious units (M , T) or are easy to look up (c_V)

Outline

- 1 Vector calculus and index notation
- 2 Multivariable calculus and thermodynamics
- 3 Differential and integral equations
- 4 Discretisation and error
- 5 Iterative solution techniques

Integral form of a differential equation

- The equations for evolving continuum descriptions of matter are typically written in differential equation form, e.g.

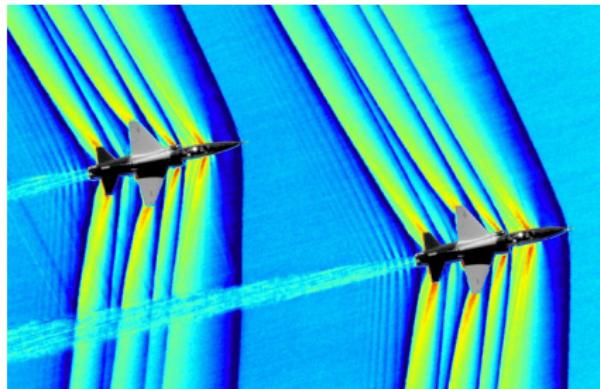
$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f}(u) = 0$$

- For this equation to be valid, we require that u is differentiable
- Mathematically, there is no reason that this is true - we shall see cases where even data which starts differentiable generates a **discontinuity**
- If this happens, we cannot solve this form of the equations, but must instead solve the **integral form**

Discontinuities

- Many discontinuities will be encountered through the continuum courses, including shock waves and material interfaces

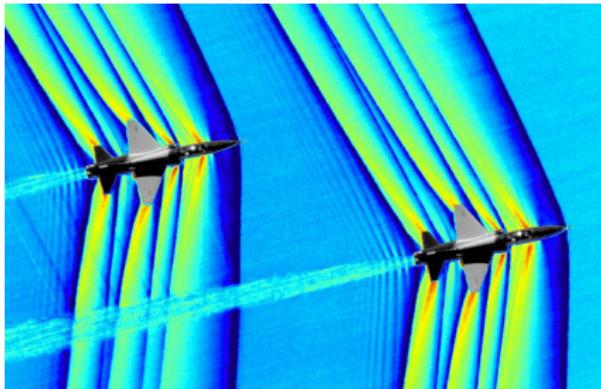
Source: nasa.gov



- In reality, these aren't true discontinuities; there is always some form of transition

Discontinuities

Source: nasa.gov



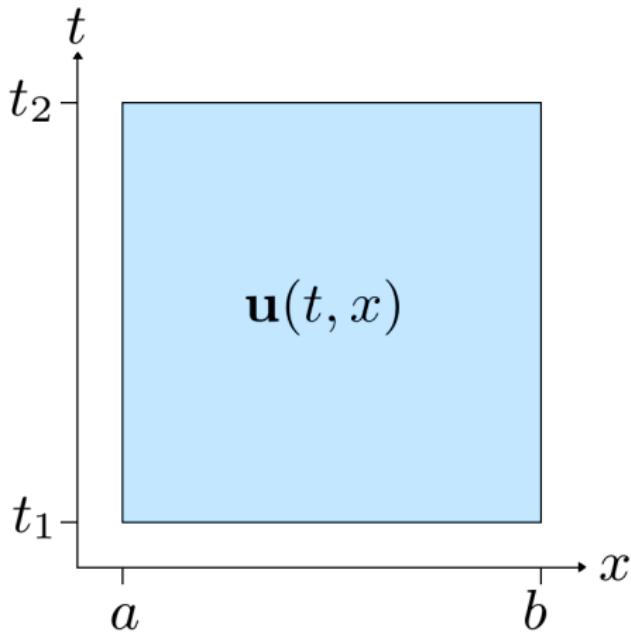
- Thermal diffusion, no matter how small, exists over shock waves
- Materials undergo a transition region (which may be atomic in scale)
- But often these are too small for a computational, or a continuum, approach to capture - equations are formulated which permit discontinuities mathematically

Integral domain

- Consider a differential equation

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0$$

- In order to convert this to an integral equation, we first define some area we are trying to solve this equation over
- Typically, we would consider the case in which we know the solution $u(t_1, x)$, $x \in [a, b]$
- And we want to solve it for some (or all) times up until t_2 , still for $x \in [a, b]$
- We can integrate our differential equation over this entire region



Integral form and conservation laws

- Begin with the assumption that u is continuous, and integrate

$$\int_a^b \int_{t_1}^{t_2} \frac{\partial u(t, x)}{\partial t} dt dx + \int_{t_1}^{t_2} \int_a^b \frac{\partial f(u(t, x))}{\partial x} dx dt = 0$$

- We can then use the fundamental theorem of calculus

$$\int_a^b u(t_2, x) - u(t_1, x) dx + \int_{t_1}^{t_2} f(u(t, b)) - f(u(t, a)) dt = 0$$

- In this form, it would not matter if u was discontinuous, a solution could still be obtained
- For actually obtaining the solution, see “Numerical Methods for Compressible Fluid Dynamics” and “Numerical Methods for Incompressible Fluid Dynamics”

Weak solutions

- A weak solution is defined as a solution of

$$\int_a^b u(t_2, x) - u(t_1, x) dx + \int_{t_1}^{t_2} f(u(t, b)) - f(u(t, a)) dt = 0$$

which may, or may not, be discontinuous

- If it is discontinuous, it cannot satisfy

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$$

- However, this discontinuous solution is indeed a (weak) solution to the underlying physical problem, just not one that can be found by considering this form of the governing equation
- It is important to note that when solving for u on $x \in [a, b]$, $t \in [t_1, t_2]$, weak solutions **may not be unique** - we shall encounter this later

Different forms of PDEs

- Throughout the continuum courses, you will be looking at solutions to various partial differential equations
- In general, these can all be written in a form which looks like

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{f}(\mathbf{u}) = \mathbf{s}(\mathbf{u})$$

- They will not always look like this, though
- Especially since the term $\mathbf{s}(\mathbf{u})$ can hide anything that is not a time derivative or divergence of a quantity
- However, the physical systems we consider, whether you start from the Navier-Stokes equations, Maxwell's equations or even Einstein's field equations, can be written in this form (in some cases, with considerable effort)

Commonly encountered forms

- For simplicity, we shall consider 1D, and $\mathbf{s} = \mathbf{0}$

$$\frac{\partial u_i}{\partial t} + \frac{\partial f_i(\mathbf{u})}{\partial x} = 0$$

- Which might look like

$$\frac{\partial u_i}{\partial t} + A_{ij} \frac{\partial u_j}{\partial x} = 0$$

- or

$$\frac{\partial w_i(\mathbf{u})}{\partial t} + B_{ij} \frac{\partial w_j(\mathbf{u})}{\partial x} = 0$$

- or

$$\frac{\partial \mathcal{V}_i}{\partial t} + \Lambda_{ij} \frac{\partial \mathcal{V}_j}{\partial x} = 0$$

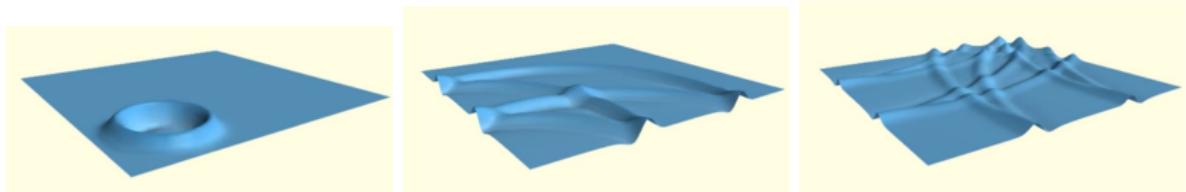
where Λ is a diagonal matrix

- These forms may allow for accurate numerical solution, analysis of the equations or even identification of analytic solutions

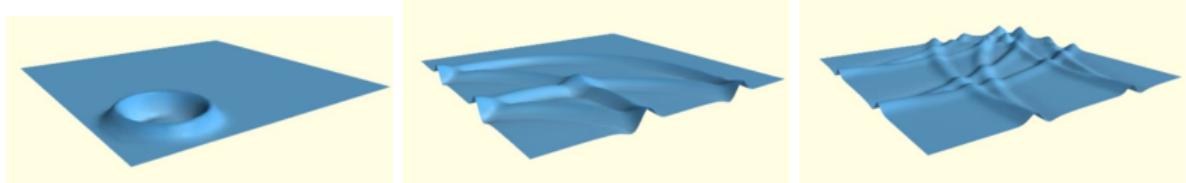
An example - the shallow water equations

- As a simple, but non-trivial example to demonstrate these different forms, and some of the mathematical techniques used, we consider the shallow water equations
- They are a reduction of the Navier-Stokes equations and are a suitable approximation for some systems where there is a vertical length scale much smaller than other length scales (such as shallow water)

$$\frac{\partial}{\partial t} \begin{pmatrix} \phi \\ \phi u \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \phi u \\ \phi u^2 + \frac{1}{2}\phi^2 \end{pmatrix} = \begin{pmatrix} 0 \\ g\phi \frac{\partial h}{\partial x} \end{pmatrix}.$$



An example - the shallow water equations



$$\frac{\partial}{\partial t} \begin{pmatrix} \phi \\ \phi u \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \phi u \\ \phi u^2 + \frac{1}{2}\phi^2 \end{pmatrix} = \begin{pmatrix} 0 \\ g\phi \frac{\partial h}{\partial x} \end{pmatrix}.$$

- In this one-dimensional form, u is the velocity, h the height (or depth) of the lower surface (e.g. the sea bed) and g is gravitational acceleration
- ϕ is the geopotential, related to the height through $\phi = gH = g(\eta + h)$ where η is the height above an initial surface
- If a uniform depth is considered (which, for laziness, it will be in this lecture), the right hand side vanishes

Conservation form

$$\frac{\partial u_i}{\partial t} + \frac{\partial f_i(\mathbf{u})}{\partial x} = 0$$

- This form of equations is referred to as conservation form (for reasons that will become clear)

$$\frac{\partial}{\partial t} \begin{pmatrix} \phi \\ \phi u \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \phi u \\ \phi u^2 + \frac{1}{2}\phi^2 \end{pmatrix} = \mathbf{0}$$

- It is clear that the shallow water equations, as presented so far, are in this form,

$$\mathbf{u} = \begin{pmatrix} \phi \\ \phi u \end{pmatrix} \quad \mathbf{f} = \begin{pmatrix} \phi u \\ \frac{(\phi u)^2}{\phi} + \frac{1}{2}\phi^2 \end{pmatrix}$$

- Note that ϕ and ϕu are **independent variables** in this case

Non-conservation form (and the Jacobian)

$$\frac{\partial u_i}{\partial t} + A_{ij} \frac{\partial u_j}{\partial x} = 0$$

- It can be good to have all derivatives being taken of the same vector, (\mathbf{u}), this is a non-conservative form
- The matrix quantity A is the **Jacobian** of the vector \mathbf{f}

$$A_{ij} = \frac{\partial f_i}{\partial u_j} \quad \text{and by the chain rule} \quad \frac{\partial f_i}{\partial x} = \frac{\partial f_i}{\partial u_j} \frac{\partial u_j}{\partial x}$$

- For the shallow water equations, we get

$$\frac{\partial}{\partial t} \begin{pmatrix} \phi \\ \phi u \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ -u^2 + \phi & 2u \end{pmatrix} \frac{\partial}{\partial x} \begin{pmatrix} \phi \\ \phi u \end{pmatrix} = \mathbf{0}$$

- Because ϕ and ϕu are independent variables,

$$\frac{\partial(\phi u)}{\partial \phi} = 0$$

Primitive-variable form

$$\frac{\partial w_i(\mathbf{u})}{\partial t} + B_{ij} \frac{\partial w_j(\mathbf{u})}{\partial x} = 0$$

- As we shall see, sometimes conservation form has advantages, but sometimes the variables aren't the most intuitive
- Why use geopotential if we could use height?
- Alternative variable choices are **primitive variables**, and may allow for intuitive or analytic understanding of the equations
- In the shallow water equations, a suitable choice of primitive variables might be total height, H , and velocity, u

Primitive-variable form

$$\frac{\partial w_i(\mathbf{u})}{\partial t} + B_{ij} \frac{\partial w_j(\mathbf{u})}{\partial x} = 0$$

- Obtaining primitive-variable form is now straightforward

$$\frac{\partial u_i}{\partial w_j} \frac{\partial w_j}{\partial t} + \frac{\partial f_j(\mathbf{w})}{\partial w_j} \frac{\partial w_j}{\partial x} = 0$$

- Therefore

$$B_{ij} = \left(\frac{\partial u_i}{\partial w_k} \right)^{-1} \frac{\partial f_k(\mathbf{w})}{\partial w_j} = \left(\frac{\partial u_i}{\partial w_k} \right)^{-1} \frac{\partial f_k}{\partial u_l} \frac{\partial u_l}{\partial w_j}$$

- For the shallow water equations,

$$\frac{\partial}{\partial t} \begin{pmatrix} H \\ u \end{pmatrix} + \begin{pmatrix} u & H \\ g & u \end{pmatrix} \frac{\partial}{\partial x} \begin{pmatrix} H \\ u \end{pmatrix} = \mathbf{0}$$

- Often, it is not worth going through the matrix calculations, and it is easy to do this by simple manipulation of the equations

Characteristic form

$$\frac{\partial \mathcal{V}_i}{\partial t} + \Lambda_{ij} \frac{\partial \mathcal{V}_j}{\partial x} = 0$$

- Characteristic form allows for a wide range of analysis of partial differential equations
- We shall see it being used for finding exact solutions in some cases, approximate solutions in others, or understanding the behaviour of the system
- It is important to note that not all equations can be written in characteristic form; since Λ is a diagonal matrix, at some point we will need to diagonalise a matrix
- We could consider characteristic form to be a very particular case of primitive variable form

$$\frac{\partial w_i(\mathbf{u})}{\partial t} + B_{ij} \frac{\partial w_j(\mathbf{u})}{\partial x} = \frac{\partial w_i(\mathbf{u})}{\partial t} + (C_{ik})^{-1} A_{kl} C_{lj} \frac{\partial w_j(\mathbf{u})}{\partial x} = 0$$

Characteristic form

$$\frac{\partial \mathcal{V}_i}{\partial t} + \Lambda_{ij} \frac{\partial \mathcal{V}_j}{\partial x} = 0$$

- We need to find a matrix, C , such that $(C_{ik})^{-1} A_{kl} C_{lj}$ is diagonal

$$\frac{\partial w_i(\mathbf{u})}{\partial t} + (C_{ik})^{-1} A_{kl} C_{lj} \frac{\partial w_j(\mathbf{u})}{\partial x} = 0$$

- In doing so, we will effectively choose the 'primitive' variables for this case, since

$$C_{ij} = \frac{\partial u_i}{\partial w_j} = \frac{\partial u_i}{\partial \mathcal{V}_j}$$

- This may well be familiar, we are **diagonalising the Jacobian** to achieve this
- Therefore the entries of Λ are the eigenvalues of A , and C_{ij} and $(C_{ij})^{-1}$ are matrices of the right and left eigenvectors respectively

Eigenvalues and eigenvectors

- Eigenvalues of a matrix, M , are solutions, λ , to

$$|M - \lambda I| = 0$$

- Each eigenvalue, λ_α , has a corresponding right eigenvector, \mathbf{r}_α , which is the solution to

$$M\mathbf{r}_\alpha = \lambda_\alpha \mathbf{r}_\alpha$$

- These are the most commonly encountered eigenvectors, if neither right nor left is specified, assume this
- Left eigenvectors, \mathbf{l}_α are solutions to

$$\mathbf{l}_\alpha M = \lambda_\alpha \mathbf{l}_\alpha$$

- Note that \mathbf{r}_α is a column vector, and \mathbf{l}_α is a row vector

Similar matrices

- In order to compute the characteristic form, we need to know the eigenvalues of the Jacobian, A
- Sometimes this is not easy to calculate
- However, when introducing primitive variable form, we wrote the equations as

$$\frac{\partial w_i(\mathbf{u})}{\partial t} + B_{ij} \frac{\partial w_j(\mathbf{u})}{\partial x} = \frac{\partial w_i(\mathbf{u})}{\partial t} + (C_{ik})^{-1} A_{kl} C_{lj} \frac{\partial w_j(\mathbf{u})}{\partial x} = 0$$

- If the relationship $B_{ij} = (C_{ik})^{-1} A_{kl} C_{lj}$ exists, then A and B are **similar matrices**
- Such matrices have many interesting algebraic properties, including one which we are interested in; they have the same eigenvalues
- For the shallow water equations, we can compute the eigenvalues of either A or B , both are equally easy

Shallow water equations' eigenvalues and eigenvectors

- Two eigenvalues,

$$\lambda_1 = u + \sqrt{\phi} = u + \sqrt{gH} \quad \lambda_2 = u - \sqrt{\phi} = u - \sqrt{gH}$$

- These eigenvalues are both **real and distinct** - we know we can write the equation in characteristic form as a result
- The right eigenvectors of B are:

$$\mathbf{r}_1 = \begin{pmatrix} \frac{1}{g}\sqrt{gH} \\ 1 \end{pmatrix} \quad \mathbf{r}_2 = \begin{pmatrix} -\frac{1}{g}\sqrt{gH} \\ 1 \end{pmatrix}$$

- The matrix of right eigenvectors is then

$$\frac{\partial w_i}{\partial \mathcal{V}_j} = \begin{pmatrix} \frac{1}{g}\sqrt{gH} & -\frac{1}{g}\sqrt{gH} \\ 1 & 1 \end{pmatrix}$$

Shallow water equations' eigenvalues and eigenvectors

$$\frac{\partial w_i}{\partial \mathcal{V}_j} = \begin{pmatrix} \frac{1}{g}\sqrt{gH} & -\frac{1}{g}\sqrt{gH} \\ 1 & 1 \end{pmatrix}$$

- Left eigenvalues can then be computed either directly, or by inverting this matrix

$$\mathbf{l}_1 = \frac{1}{2} \left(\frac{g}{\sqrt{gH}}, 1 \right), \quad \mathbf{l}_2 = \frac{1}{2} \left(-\frac{g}{\sqrt{gH}}, 1 \right)$$

- We now have, as expected,

$$\Lambda = \begin{pmatrix} u + \sqrt{gH} & 0 \\ 0 & u - \sqrt{gH} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \frac{g}{\sqrt{gH}} & 1 \\ -\frac{g}{\sqrt{gH}} & 1 \end{pmatrix} \begin{pmatrix} u & H \\ g & u \end{pmatrix} \begin{pmatrix} \frac{1}{g}\sqrt{gH} & -\frac{1}{g}\sqrt{gH} \\ 1 & 1 \end{pmatrix}$$

- Now all we need is \mathcal{V}

Characteristic form of the shallow water equations

- Note that (for well behaved w and \mathcal{V})

$$(C_{ij})^{-1} = \left(\frac{\partial w_i}{\partial \mathcal{V}_j} \right)^{-1} = \frac{\partial \mathcal{V}_j}{\partial w_i}$$

- This means we can integrate both sides with respect to w to get \mathcal{V}

$$\mathcal{V} = \begin{pmatrix} \sqrt{gH} + \frac{1}{2}u \\ \sqrt{gH} - \frac{1}{2}u \end{pmatrix}$$

- This gives

$$\frac{\partial}{\partial t} \begin{pmatrix} \sqrt{gH} + \frac{1}{2}u \\ \sqrt{gH} - \frac{1}{2}u \end{pmatrix} + \begin{pmatrix} u + \sqrt{gH} & 0 \\ 0 & u - \sqrt{gH} \end{pmatrix} \frac{\partial}{\partial x} \begin{pmatrix} \sqrt{gH} + \frac{1}{2}u \\ \sqrt{gH} - \frac{1}{2}u \end{pmatrix} = \mathbf{0}$$

Summary

$$\frac{\partial}{\partial t} \begin{pmatrix} \phi \\ \phi u \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \phi u \\ \phi u^2 + \frac{1}{2}\phi^2 \end{pmatrix} = \mathbf{0}$$

$$\frac{\partial}{\partial t} \begin{pmatrix} \phi \\ \phi u \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ -u^2 + \phi & 2u \end{pmatrix} \frac{\partial}{\partial x} \begin{pmatrix} \phi \\ \phi u \end{pmatrix} = \mathbf{0}$$

$$\frac{\partial}{\partial t} \begin{pmatrix} H \\ u \end{pmatrix} + \begin{pmatrix} u & H \\ g & u \end{pmatrix} \frac{\partial}{\partial x} \begin{pmatrix} H \\ u \end{pmatrix} = \mathbf{0}$$

$$\frac{\partial}{\partial t} \begin{pmatrix} \sqrt{gH} + \frac{1}{2}u \\ \sqrt{gH} - \frac{1}{2}u \end{pmatrix} + \begin{pmatrix} u + \sqrt{gH} & 0 \\ 0 & u - \sqrt{gH} \end{pmatrix} \frac{\partial}{\partial x} \begin{pmatrix} \sqrt{gH} + \frac{1}{2}u \\ \sqrt{gH} - \frac{1}{2}u \end{pmatrix} = \mathbf{0}$$

- Four different ways to express the same equation
- All will be encountered through the continuum courses, and the reasons for these forms will become clearer at this point!

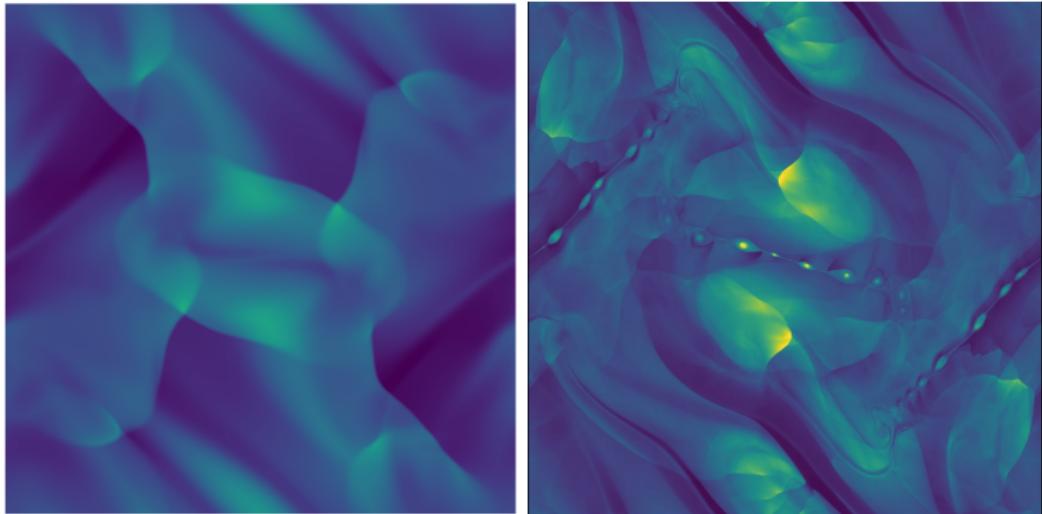
Outline

- 1 Vector calculus and index notation
- 2 Multivariable calculus and thermodynamics
- 3 Differential and integral equations
- 4 Discretisation and error
- 5 Iterative solution techniques

Discretisation

- All computational simulations require some means of discretisation of a continuum domain into a discrete one
- Any such discretisation will introduce an error
- Understanding this error, and how it will affect your simulations is key when employing a numerical method
- The various continuum courses will cover the error in specific methods in more detail, here we look at a few techniques for analysing error

Effects of error



- Two different numerical methods attempting to simulate the same problem

Order of accuracy

- All numerical methods have an **order of accuracy**; this is a measure of how quickly a numerical simulation approaches the true mathematical solution to the problem
- Taylor series are a good way to identify order of accuracy
- The Taylor series of a function $f(x)$ is computed about a point, $x = a$, as

$$f(a) + \frac{\partial f}{\partial x} \Big|_{x=a} (x - a) + \frac{1}{2} \frac{\partial^2 f}{\partial x^2} \Big|_{x=a} (x - a)^2 + \frac{1}{3!} \frac{\partial^3 f}{\partial x^3} \Big|_{x=a} (x - a)^3 + \dots$$

- In some cases, increasing the number of terms in the Taylor series gives a function that gets increasingly close to $f(x)$ (**it converges**)
- In many cases, this does not hold in general, but might be expected to give a reasonable approximation for x close to a
- And we can use this to define approximations to derivatives

Derivative approximations

- We can consider a Taylor expansion at a point $x + \Delta x$ about x ,

$$f(x + \Delta x) = f(x) + \frac{\partial f}{\partial x} \Big|_x \Delta x + \frac{\partial^2 f}{\partial x^2} \Big|_x \frac{(\Delta x)^2}{2} + \frac{\partial^3 f}{\partial x^3} \Big|_x \frac{(\Delta x)^3}{3!} + \dots$$

- This approximation assumes that Δx is sufficiently small
- In terms of numerically solving a PDE, we may well know a value of f at both x and $x + \Delta x$, and it is the derivatives we need to calculate
- From this, we get

$$\frac{\partial f}{\partial x} \Big|_x = \frac{f(x + \Delta x) - f(x)}{\Delta x} + \frac{\partial^2 f}{\partial x^2} \Big|_x \frac{(\Delta x)}{2} + \frac{\partial^3 f}{\partial x^3} \Big|_x \frac{(\Delta x)^2}{3!} + \dots$$

First-order derivative

- As Δx is supposed to be small, it is reasonable to assume that any term multiplied by this is negligible
- For this reason, the derivative approximation is often written

$$\left. \frac{\partial f}{\partial x} \right|_x = \frac{f(x + \Delta x) - f(x)}{\Delta x} + \mathcal{O}(\Delta x)$$

- The $\mathcal{O}(\Delta x)$ term is the **error** within this approximation, and since $\Delta x > (\Delta x)^2 > (\Delta x)^3 > \dots$ (again, for small Δx), this is the dominant term in the error
- Since $\Delta x = (\Delta x)^1$, this is referred to as a **first order accurate approximation**
- If we halve Δx , we halve the error from that dominant term

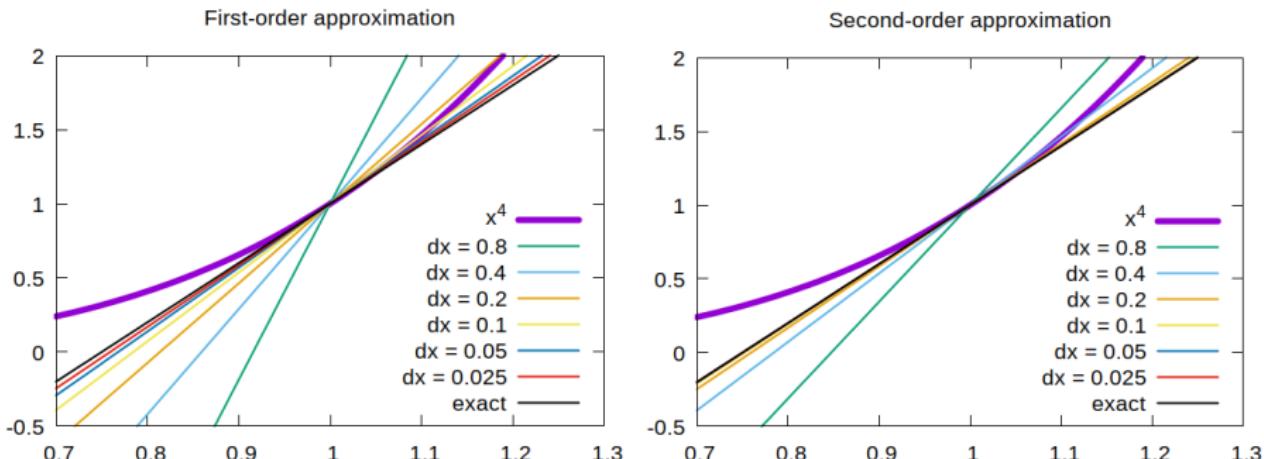
Second-order derivative

- The first order approximation used two values, $f(x)$ and $f(x + \Delta x)$, to compute the derivative
- It is, perhaps, unsurprising then that this is a relatively low-accuracy approximation
- We could do better by considering more points to compute the derivative
- Taylor expansions of $f(x + \Delta x)$ and $f(x - \Delta x)$ combine to give

$$\frac{\partial f}{\partial x} \Big|_x = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} + \mathcal{O}(\Delta x^2)$$

- This is a **second order accurate approximation**
- If we half Δx , the error from the dominant term decreases by a factor 4

Approximating derivatives



Quantifying order of accuracy

- When solving partial differential equations, you do not have a single derivative to consider, but multiple terms in a coupled system
- And numerical methods for solving these equations do not always give you the answer you expect, this might be because:
 - The errors from one derivative approximation dominate over others
 - The method does not **converge** to a solution
 - There is a bug in the code you are running
- In these cases, you want to use your numerical solution to obtain an order of accuracy, and check that you are seeing the expected behaviour
- This is commonly done using **p-norms** (or **l_p -norms**)

- Norms are a way of measuring the size of a vector, and are a generalisation of absolute value
- One type of vector norm is very familiar (the Euclidean norm):

$$|\mathbf{a}| = (\mathbf{a} \cdot \mathbf{a})^{1/2} = (a_1^2 + a_2^2 + a_3^2)^{1/2}$$

- However, similar expressions exist for any length of vector, and the squaring and then taking the root can also be generalised
- If \mathbf{b} is a vector with N components, its p -norms are

$$\|\mathbf{b}\|_p = (|b_1|^p + |b_2|^p + |b_3|^p + \dots + |b_N|^p)^{1/p}$$

- The idea for assessing order of accuracy of a numerical method is for \mathbf{b} to be some measure of the error in a numerical solution compared to an exact solution

Error norms

- Suppose that the N components of \mathbf{b} are the approximate solution to the function $B(x)$ computed at N positions, $x = (x_1, x_2, \dots, x_N)$
- For a given position, x_i , the error is given by

$$E_i = b_i - B(x_i)$$

- Effectively we want to sum up the error, but it needs to be appropriately weighted
- If each of the positions is equally distributed in space, the **error norms** are defined by

$$\|\mathbf{E}\|_p = \left(\frac{x_N - x_1}{N} \sum_{i=1}^N |E_i|^p \right)^{1/p}$$

Common error norms

- In theory, all error norms provide a useful measure of error, but in practice, it is rare to find anyone using a 4-norm, or a 23-norm
- It is most common to see the 1-norm (particularly for nonlinear problems), 2-norm (particularly for linear problems) and ∞ -norm (particularly for smooth problems)
- The ∞ -norm is the limit of the error norms as $p \rightarrow \infty$
- In this limit:

$$\lim_{p \rightarrow \infty} \sum_{i=1}^N |E_i|^p = \max_i |E_i|^p \quad \text{and} \quad \lim_{p \rightarrow \infty} \left(\frac{x_N - x_1}{N} \right)^p = 1$$

- As a result,

$$\|\mathbf{E}\|_\infty = \max_i |E_i|$$

Computing order of accuracy from error norms

- If error norms are computed for two different approximations, the ratio of these errors can be used to compute the order of accuracy
- This is easiest when one approximation (\mathbf{b}_2) uses double the number of points in its computation as the other (\mathbf{b}_1) (or half the Δx)
- Then the order of accuracy for these results is obtained simply by the ratio

$$\frac{\|\mathbf{E}(\mathbf{b}_1)\|_p}{\|\mathbf{E}(\mathbf{b}_2)\|_p}$$

- In practice, multiple error norms are computed for a given method (e.g. repeatedly doubling the number of points) for confidence in the value
- So far, we have assumed that an exact solution exists too - if it doesn't, comparisons of e.g. \mathbf{b}_1 and \mathbf{b}_2 can be made with a third solution, \mathbf{b}_3 , however, additional work may be required to show that there is **convergence to the correct solution**

Fourier series

- Error norms are a practical technique to show that a numerical solution is converging
- Fourier series can be used to provide a theoretical technique for finding when a numerical solution will converge
- Full details will be in “Numerical Methods for Compressible Fluid Dynamics”, for now we’ll just recap Fourier series
- Taylor series represent a function as a sum of polynomials

$$f(x + \Delta x) \approx f(x) + \frac{\partial f}{\partial x} \Big|_x \Delta x + \frac{\partial^2 f}{\partial x^2} \Big|_x \frac{(\Delta x)^2}{2} + \frac{\partial^3 f}{\partial x^3} \Big|_x \frac{(\Delta x)^3}{3!} + \dots$$

- Fourier series represent a **periodic function** as a sum of trigonometric functions

$$g(x) = A_0 + \sum_{n=1}^{\infty} \left(A_n \cos \left(\frac{2\pi n x}{P} \right) + B_n \sin \left(\frac{2\pi n x}{P} \right) \right)$$

Computing Fourier series

- For a given function, $g(x)$, with periodicity, P , the constants of the Fourier series are given by

$$A_0 = \frac{1}{P} \int_{-P/2}^{P/2} g(x) dx$$

$$A_n = \frac{2}{P} \int_{-P/2}^{P/2} g(x) \cos\left(\frac{2\pi nx}{P}\right) dx$$

$$B_n = \frac{2}{P} \int_{-P/2}^{P/2} g(x) \sin\left(\frac{2\pi nx}{P}\right) dx$$

- The idea behind using Fourier series in computing convergence, however, is not to compute these quantities
- If a solution is found on a finite domain, then by replicating this domain, it is straightforward to make it periodic
- We can express the solution, **or its error**, as a Fourier series

Converging solutions

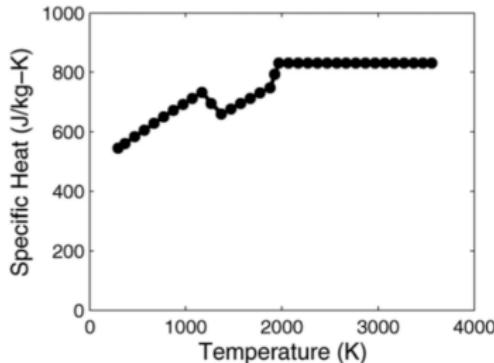
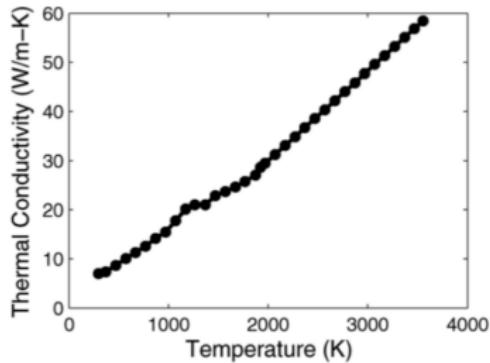
- Assume we have a Fourier series expansion for the error of a solution
- If we can show, for every n , that this error decreases as the simulation progresses, we know we will **converge to a solution**
- This is often possible to do simply, and analytically
- This requires an equivalent form of the Fourier series,

$$g(x) \approx \sum_{n=-\infty}^{\infty} \left(c_n \exp \left(\frac{2\pi i n x}{P} \right) \right)$$

- The advantage of this form is that we only need to worry about a single c_n , which makes simplification easier

Discretisation and interpolation

- When we have a solution on discrete space, we may need information about points **in between** these known values
- This could be for identifying material interfaces, identifying contours or determining material properties



- To achieve this, we may find we need to interpolate data

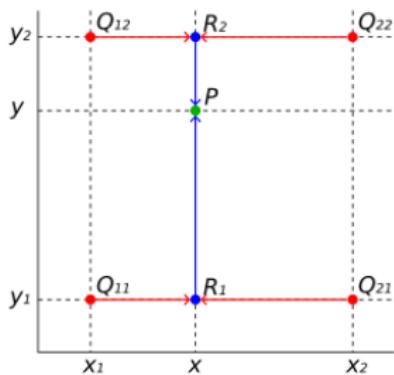
Discretisation and interpolation

- Like with other numerical methods, different types of discretisation have different orders of accuracy (and different computational cost)
- In many cases, a simple linear interpolation is sufficient because:
 - There is a low computational cost for such interpolation
 - There is no issue with overshooting near the edges of the data range
 - Often, e.g. at material interfaces, accuracy is already low (for reasons we shall cover)
 - The tabulated data is sufficiently accurate that even first-order accurate interpolation has a low error
- For a function $y = y(x)$, if we know (x_1, y_1) and (x_2, y_2) , and that $x_1 \leq x \leq x_2$, then linear interpolation gives

$$y(x) \approx y_1 + (y_2 - y_1) \frac{x - x_1}{x_2 - x_1}$$

Bilinear and trilinear interpolation

- The use of simple interpolation functionality is more important if you are in multiple dimensions
- Bilinear and trilinear interpolation is the 2D and 3D equivalent to linear interpolation
- And they also work by considering one-dimension at a time
- In 2D, if we have $z = z(x, y)$



$$z(x, y_1) \approx \frac{x_2 - x}{x_2 - x_1} z(x_1, y_1) + \frac{x - x_1}{x_2 - x_1} z(x_2, y_1)$$

$$z(x, y_2) \approx \frac{x_2 - x}{x_2 - x_1} z(x_1, y_2) + \frac{x - x_1}{x_2 - x_1} z(x_2, y_2)$$

$$z(x, y) \approx \frac{y_2 - y}{y_2 - y_1} z(x, y_1) + \frac{y - y_1}{y_2 - y_1} z(x, y_2)$$

Other interpolation

- Bilinear and trilinear interpolation work in very similar ways, taking one direction at a time
- The order of the direction is also unimportant
- If more accurate interpolation is needed, spline interpolation exists in 1D, bicubic in 2D and tricubic in 3D
- In order to get greater accuracy, however, more points are needed, 16 in the case of bicubic interpolation, and 64 in the case of tricubic

Outline

- 1 Vector calculus and index notation
- 2 Multivariable calculus and thermodynamics
- 3 Differential and integral equations
- 4 Discretisation and error
- 5 Iterative solution techniques

Iterative solutions

- Solving partial differential equations is typically done by a series of iterations - these will be the focus of Numerical Methods for Compressible Fluid Dynamics and Numerical Methods for Incompressible Fluid Dynamics
- At various other times, however, we may need other iterative techniques
- For example, many materials are governed by complex relationships between physical quantities such as internal energy (ε), density (related to I_1 , I_2 and I_3) and entropy (s)

$$\varepsilon = \frac{K_0}{2\alpha_R^2} \left(I_3^{\alpha_R/2} - 1 \right)^2 + c_V T_0 I_3^{\gamma_R/2} \left(e^{s/c_v} - 1 \right) + \frac{b_0^2}{2} I_3^{\beta_R/2} \left(\frac{I_1^2}{3} - I_2 \right)$$

- If we need to invert this equation for any reason (e.g. to obtain I_3), we may struggle to do it analytically
- Iterative techniques can help us

Analytic, exact and approximate solutions

- Numerical methods that will be studied in the continuum courses are approximate solutions - these have a finite error and order of accuracy
- Analytic solutions are those that can be written down as an equation on paper - there is no inaccuracy in calculating a solution (in theory)

$$p = (\gamma - 1) \rho \varepsilon$$

- Exact solutions are those to which an equation exists defining the solution, but this could be implicit, or non-invertible (analytic solutions are a subset of this case)

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{t^2/2} dt$$

- Iterative solutions (or root-finding techniques) are computational (or algorithmic) methods which can compute exact solutions to an arbitrary degree of accuracy

Bisection method

- The bisection method is a reliable, but relatively inefficient, algorithm for finding the roots to an equation
- Whatever we are trying to solve needs to be written as

$$f(x) = 0$$

- For example,

$$f(\Phi) = \Phi - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{t^2/2} dt = 0$$

where we know x , and want to find the value of Φ

- To start a bisection root finding algorithm, we need two values; one where $f(x) > 0$, and one where $f(x) < 0$
- The bisection method is then guaranteed to find **one root** between these two values - this may not be the only root

Bisection method

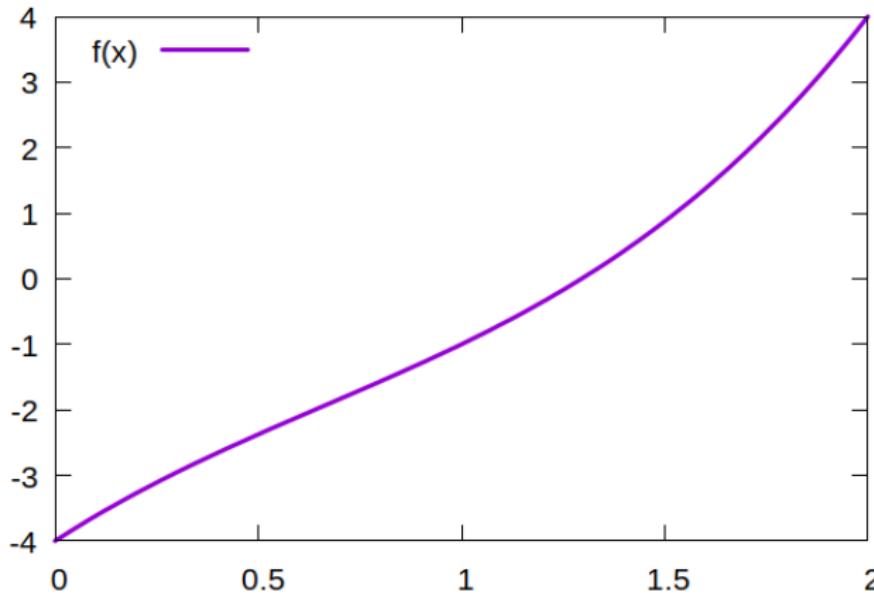
- Assume we have a and b such that $f(a) < 0$ and $f(b) > 0$
- ① Set $c = \frac{1}{2}(a + b)$
- ② Evaluate $f(c)$
- ③ If $f(c) < \epsilon$ where ϵ is some sufficiently small number, then c is the root of the equation
- ④ Otherwise, c will replace either a or b

$$\begin{cases} f(c) < 0 : & \text{set } a = c \\ f(c) > 0 : & \text{set } b = c \end{cases}$$

- ⑤ Repeat until we have found the root of the equation

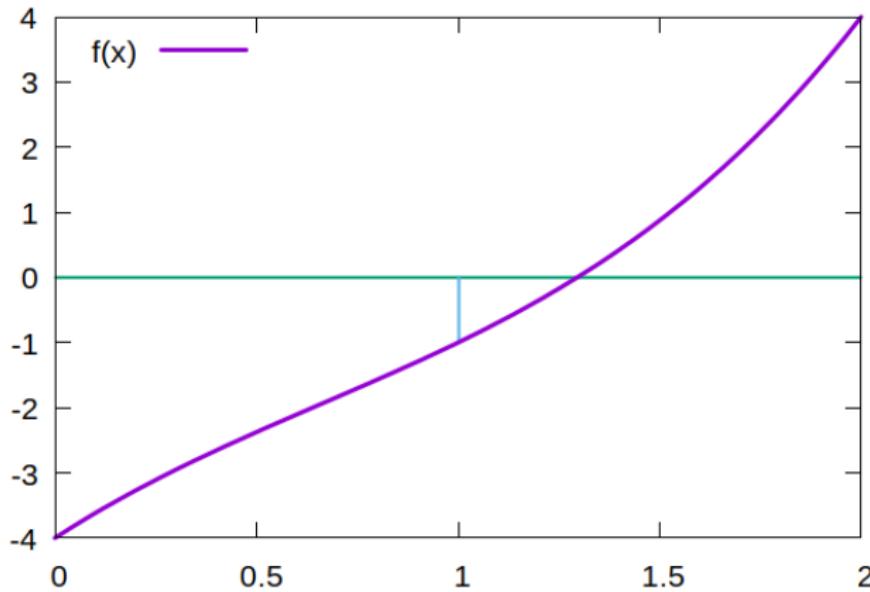
Bisection method

$$f(x) = x^3 - 2x^2 + 4x - 4$$



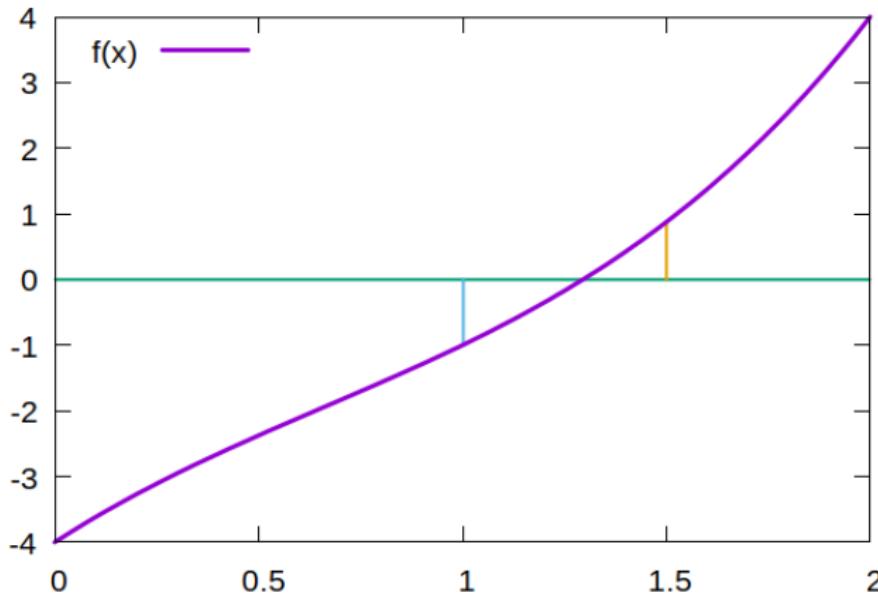
Bisection method

$$f(x) = x^3 - 2x^2 + 4x - 4$$



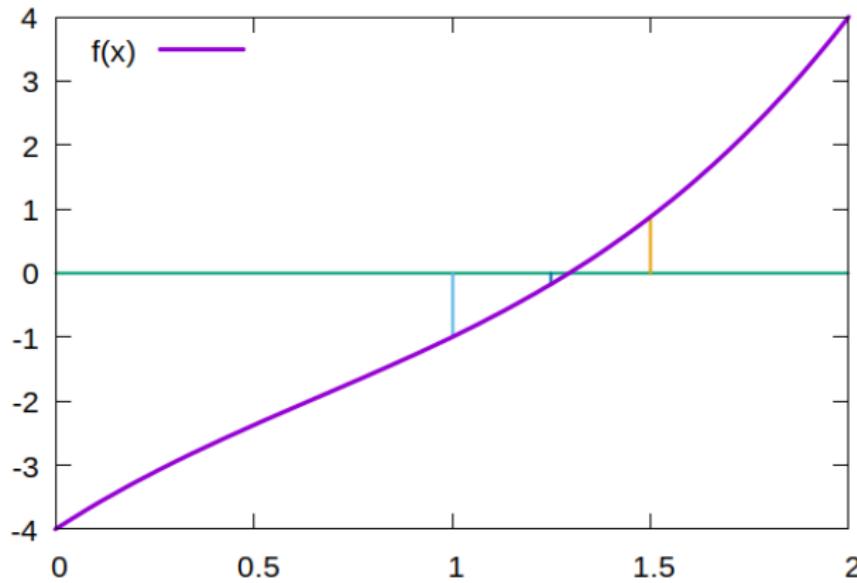
Bisection method

$$f(x) = x^3 - 2x^2 + 4x - 4$$



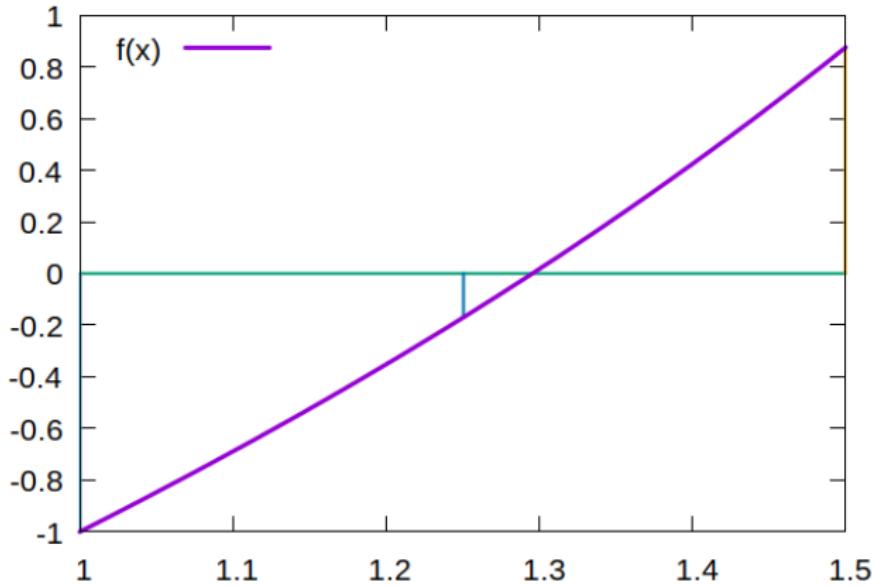
Bisection method

$$f(x) = x^3 - 2x^2 + 4x - 4$$



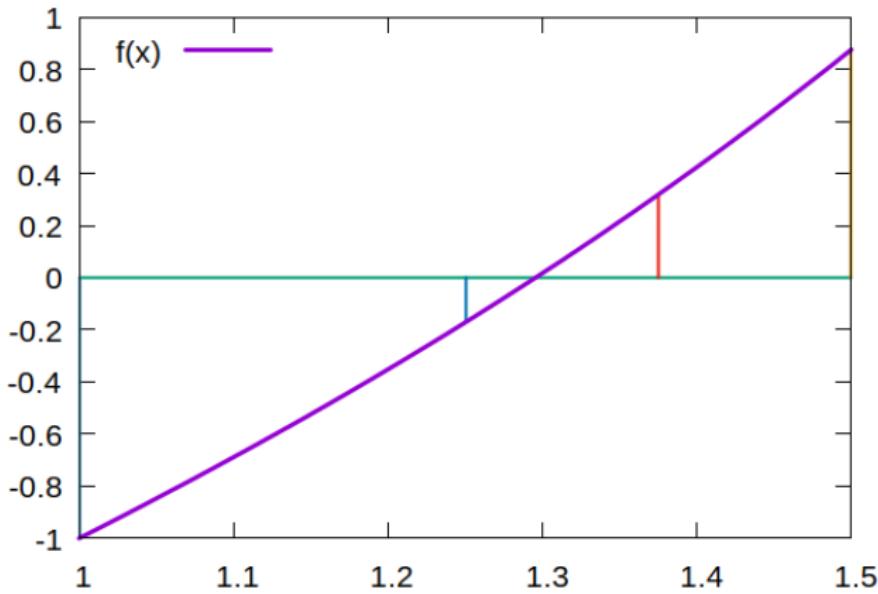
Bisection method

$$f(x) = x^3 - 2x^2 + 4x - 4$$



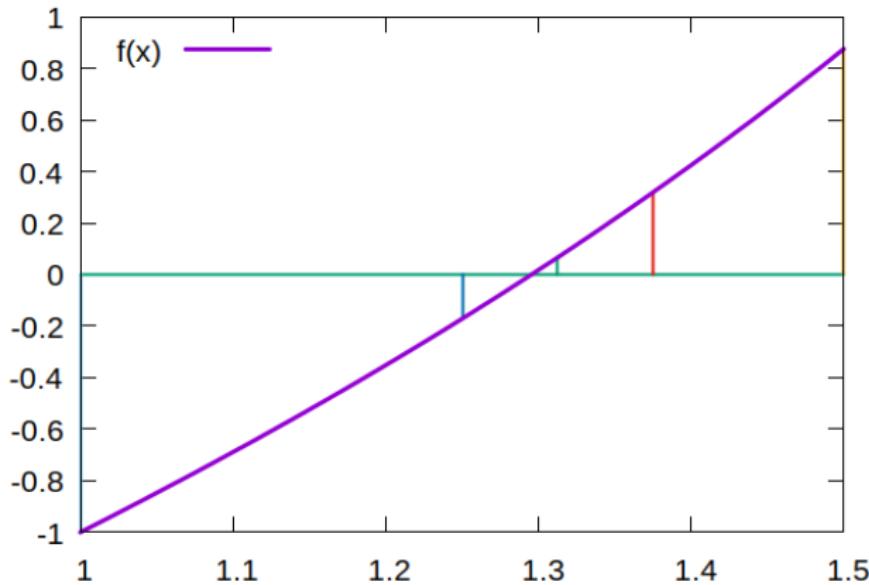
Bisection method

$$f(x) = x^3 - 2x^2 + 4x - 4$$



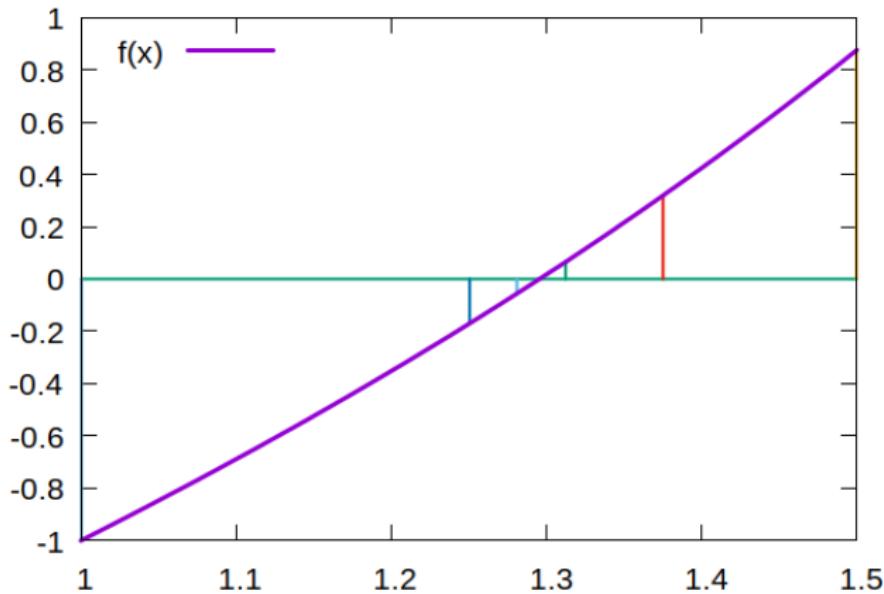
Bisection method

$$f(x) = x^3 - 2x^2 + 4x - 4$$



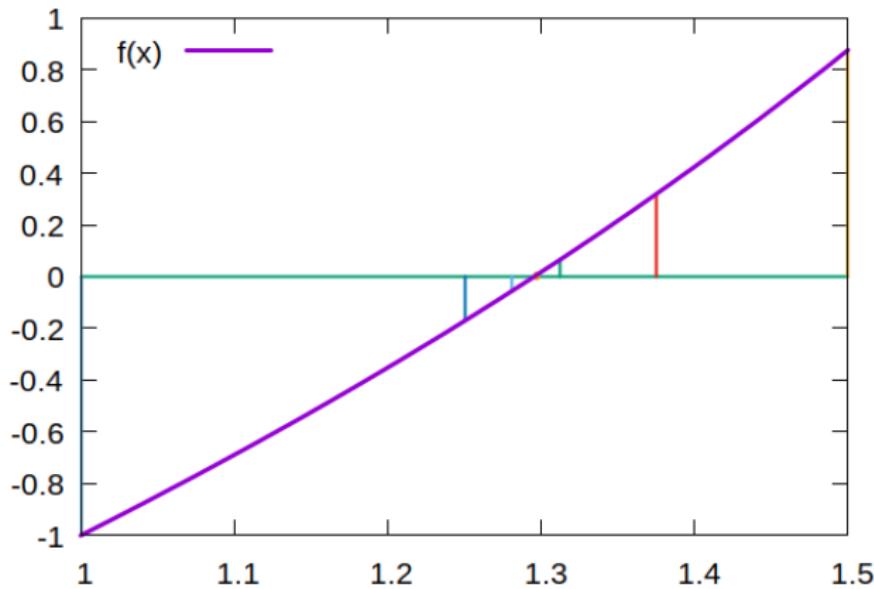
Bisection method

$$f(x) = x^3 - 2x^2 + 4x - 4$$



Bisection method

$$f(x) = x^3 - 2x^2 + 4x - 4$$



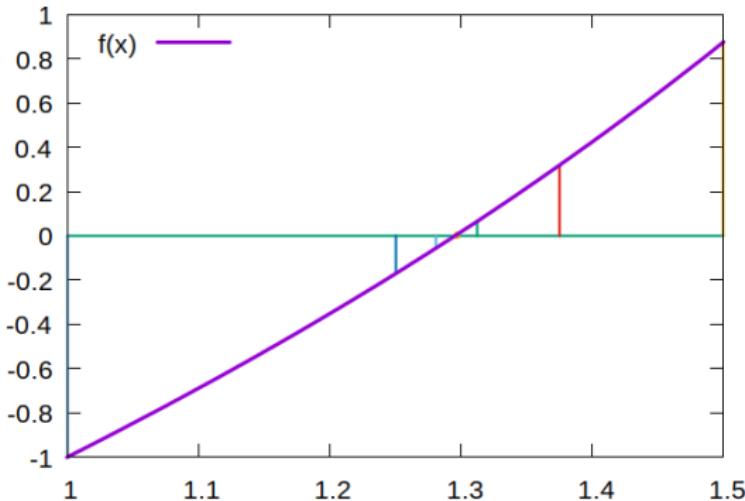
Bisection method

- After 6 iterations:

$$c = 1.296875$$

$$f(c) = 0.004925$$

$$f(x) = x^3 - 2x^2 + 4x - 4$$



Newton's method

- The bisection method is reliable, simple, but inefficient
- Under certain measures, it is the best method which is **guaranteed** to find a root
- In a numerical simulation, however, root finding may need to be done a lot - inefficient is not good
- Newton's method, which is efficient, simple, and usually reliable, is a popular choice to improve things
- The idea behind the method is that if, for any single guess, we follow the slope of the derivative down to the axis, we get closer to the solution

Newton's method

- For Newton's method, we make an initial guess, x_n , for which we know $f(x_n)$
- For the method to work, we need to know the derivative of $f(x)$, and be able to calculate it at x_n
- We then find the point x_{n+1} such that $f(x_{n+1}) = 0$
- Fortunately, this is straightforward, since, for a straight line

$$\frac{\partial f}{\partial x} \Big|_{x=x_n} = \frac{0 - f(x_n)}{x_{n+1} - x_n}$$

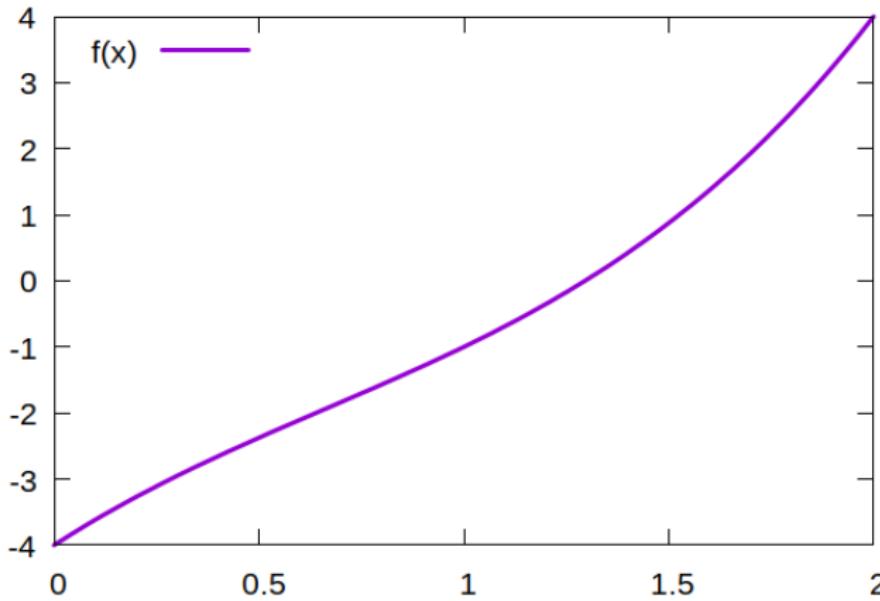
- Our new guess for x is then

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

- If $f(x_{n+1}) < \epsilon$ then we have converged
- Otherwise, repeat the above formula to get x_{n+2} and continue until convergence

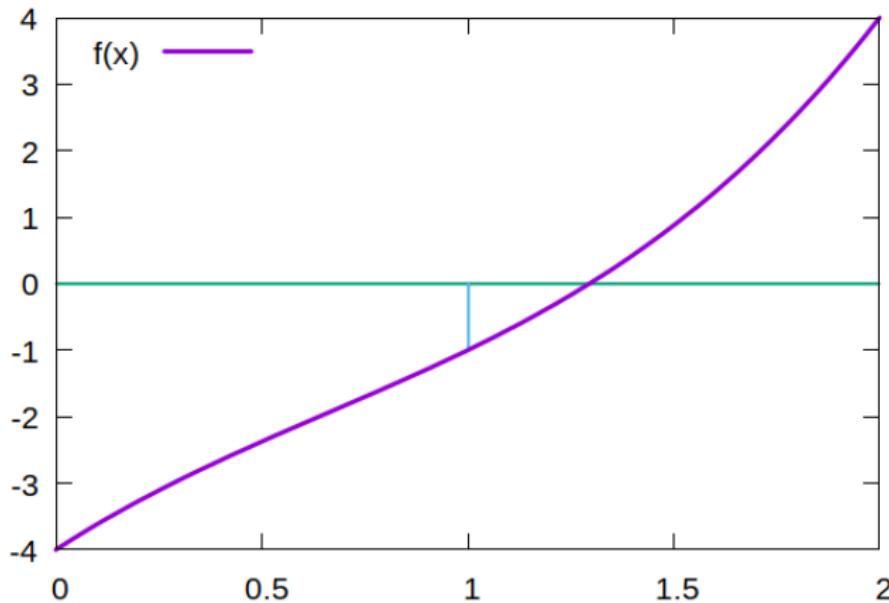
Newton's method

$$f(x) = x^3 - 2x^2 + 4x - 4, f' = 3x^2 - 4x + 4$$



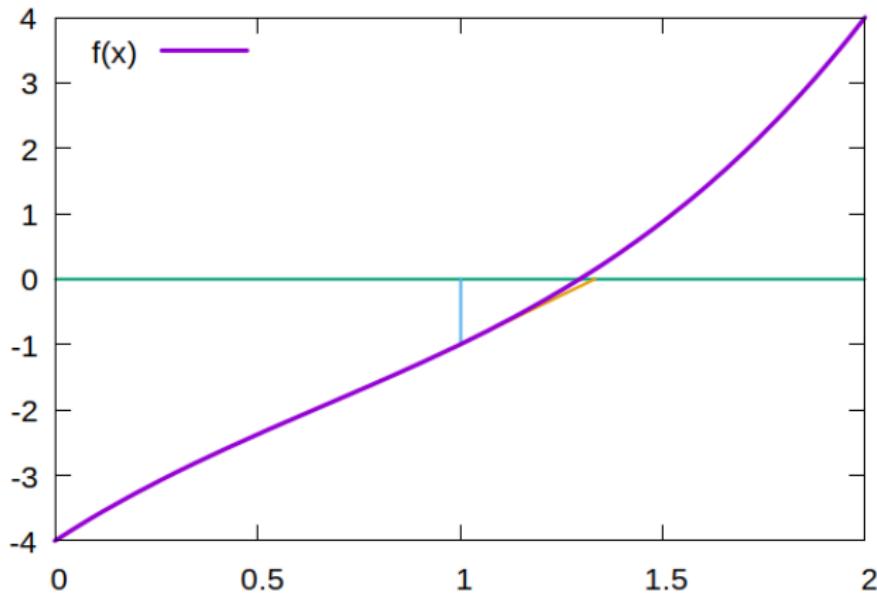
Newton's method

$$f(x) = x^3 - 2x^2 + 4x - 4, f' = 3x^2 - 4x + 4$$



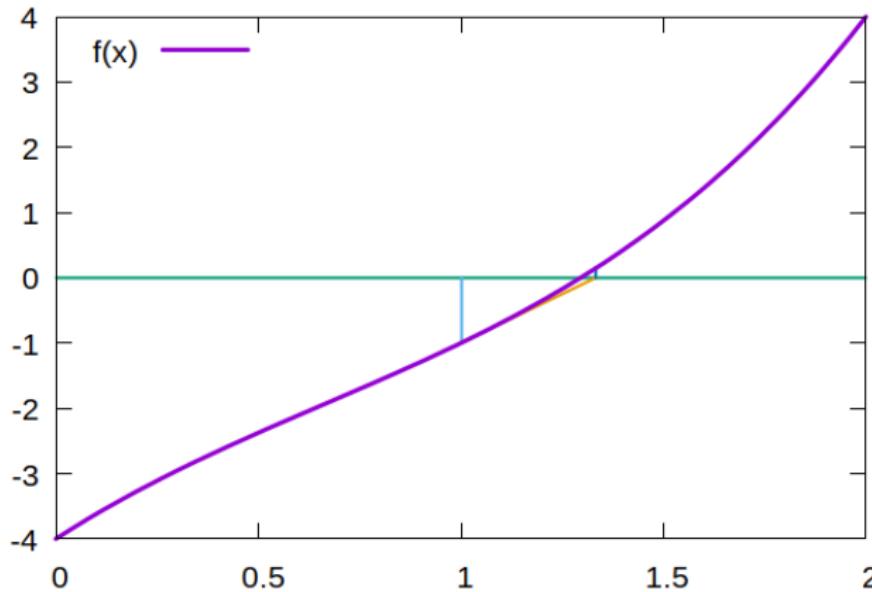
Newton's method

$$f(x) = x^3 - 2x^2 + 4x - 4, f' = 3x^2 - 4x + 4$$



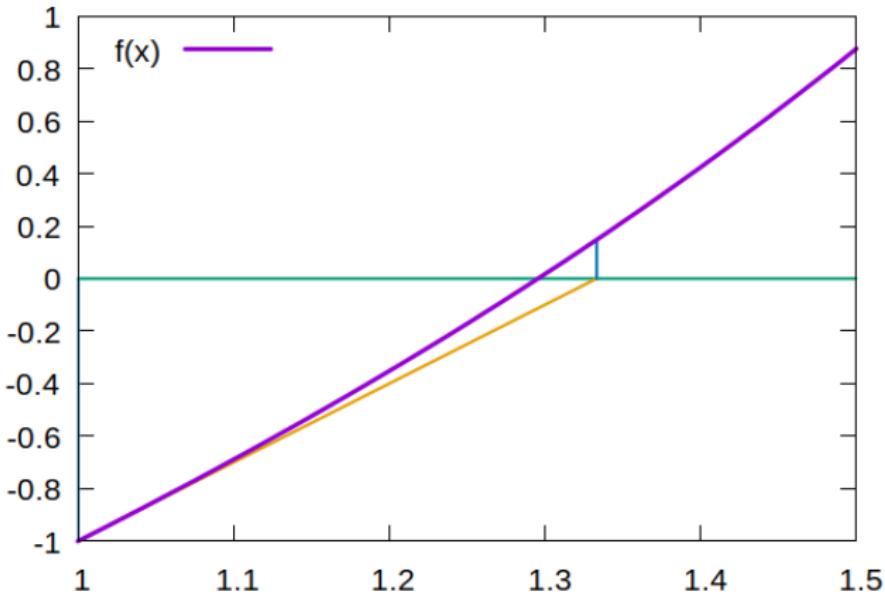
Newton's method

$$f(x) = x^3 - 2x^2 + 4x - 4, f' = 3x^2 - 4x + 4$$



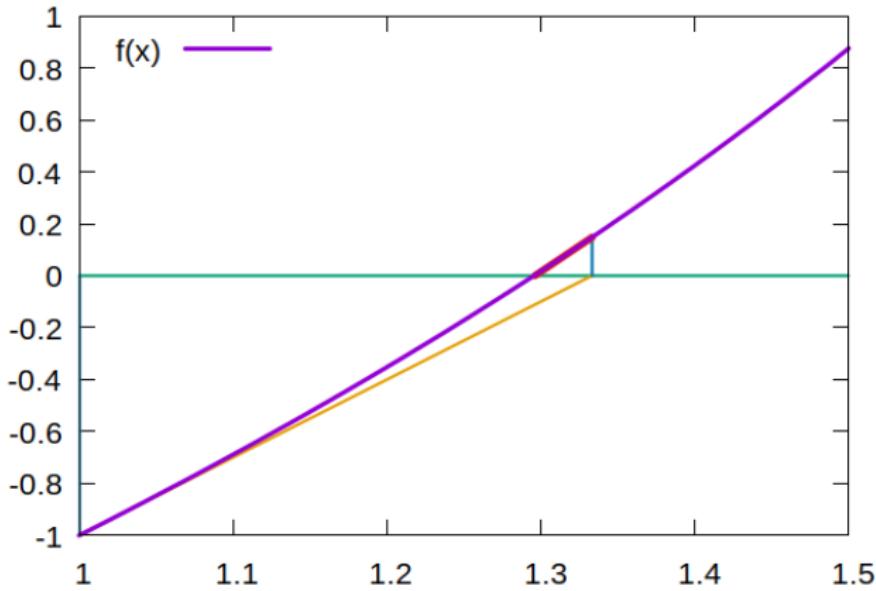
Newton's method

$$f(x) = x^3 - 2x^2 + 4x - 4, f' = 3x^2 - 4x + 4$$



Newton's method

$$f(x) = x^3 - 2x^2 + 4x - 4, f' = 3x^2 - 4x + 4$$



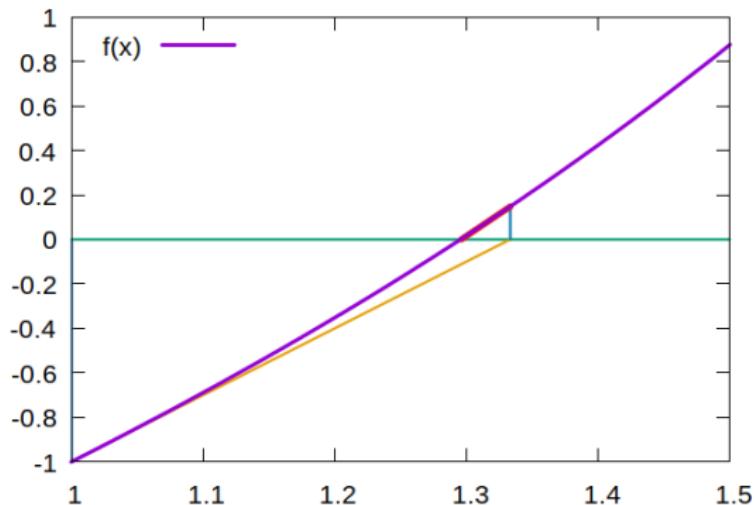
Newton's method

- After 2 iterations:

$$c = 1.2963$$

$$f(c) = 0.002967$$

$$f(x) = x^3 - 2x^2 + 4x - 4, f' = 3x^2 - 4x + 4$$

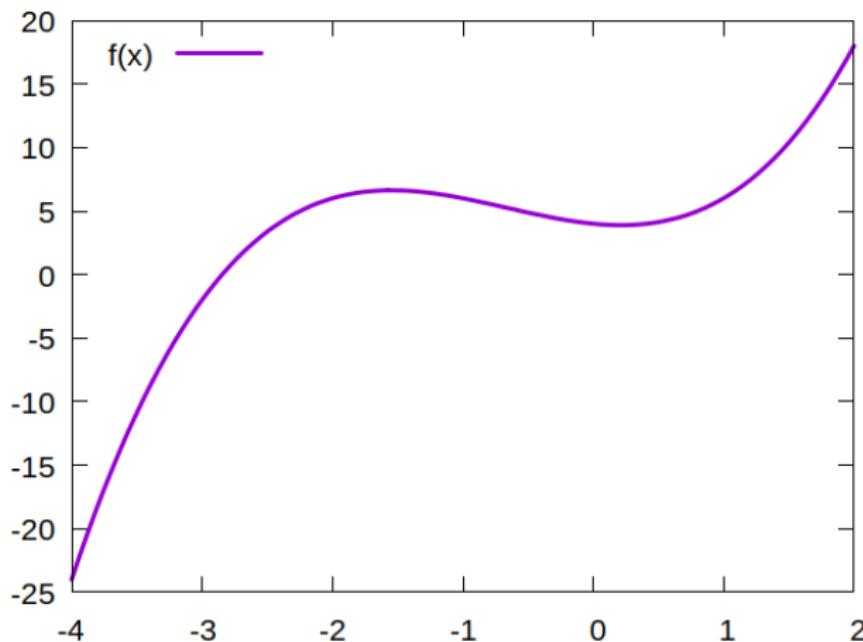


Newton's method

- When Newton's method works, it usually works quickly, but it can have issues
- Clearly, if the derivative of $f(x)$ doesn't exist, or if $f'(x_n) = 0$, the algorithm fails
- However, it is also possible for x_{n+1} to end up further from the root than x_n was, and for this cycle to continue
- This is most likely if the initial guess is too far from the true root
- Sometimes, convergence is just slow (e.g. if the root is a minimum or maximum of $f(x)$)
- Whilst there are sometimes techniques to deal with this behaviour, in practice, it is often easiest to try Newton's method, and if that fails, try something guaranteed to find the root, e.g. the bisection method

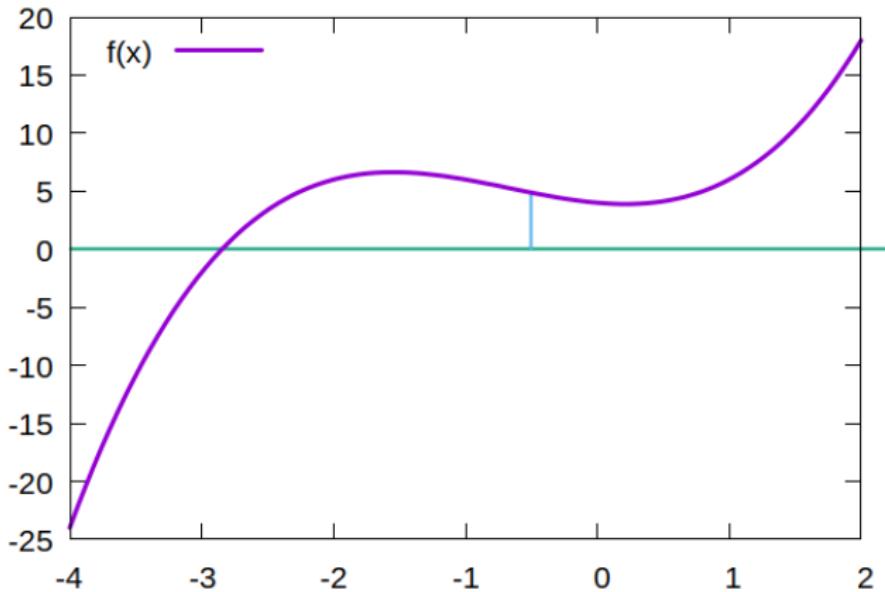
Newton's method - a bad choice

$$f(x) = x^3 + 2x^2 - x + 4, f' = 3x^2 + 4x - 1$$



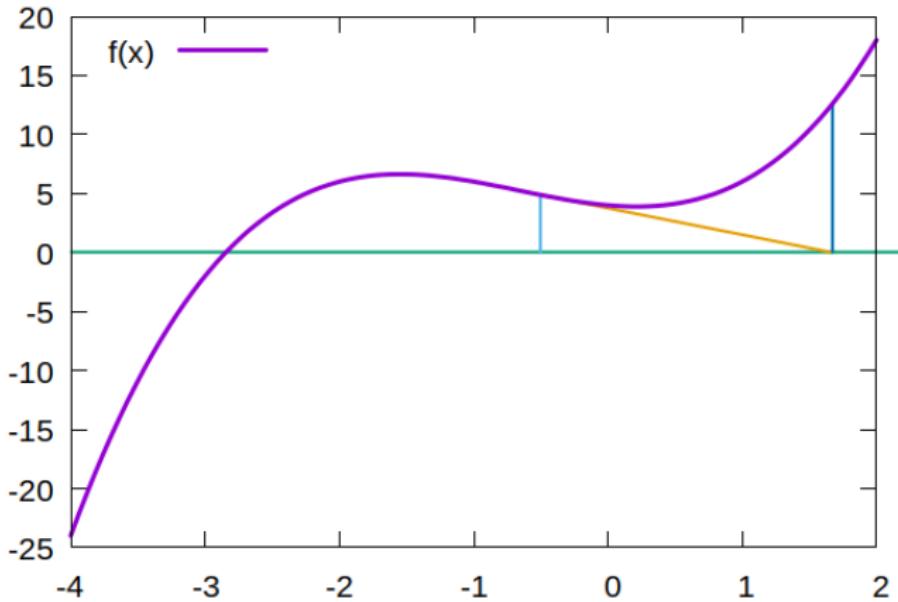
Newton's method - a bad choice

$$f(x) = x^3 + 2x^2 - x + 4, f' = 3x^2 + 4x - 1$$



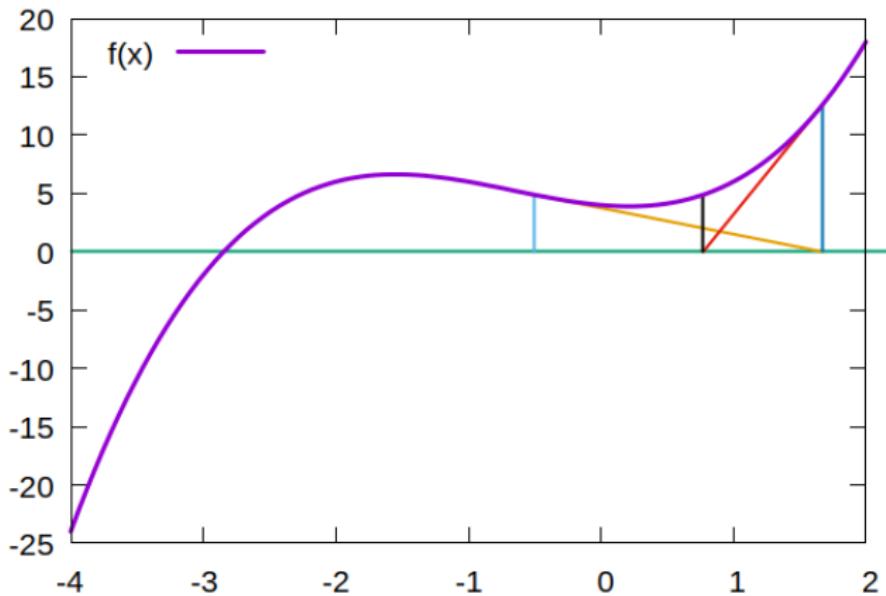
Newton's method - a bad choice

$$f(x) = x^3 + 2x^2 - x + 4, f' = 3x^2 + 4x - 1$$



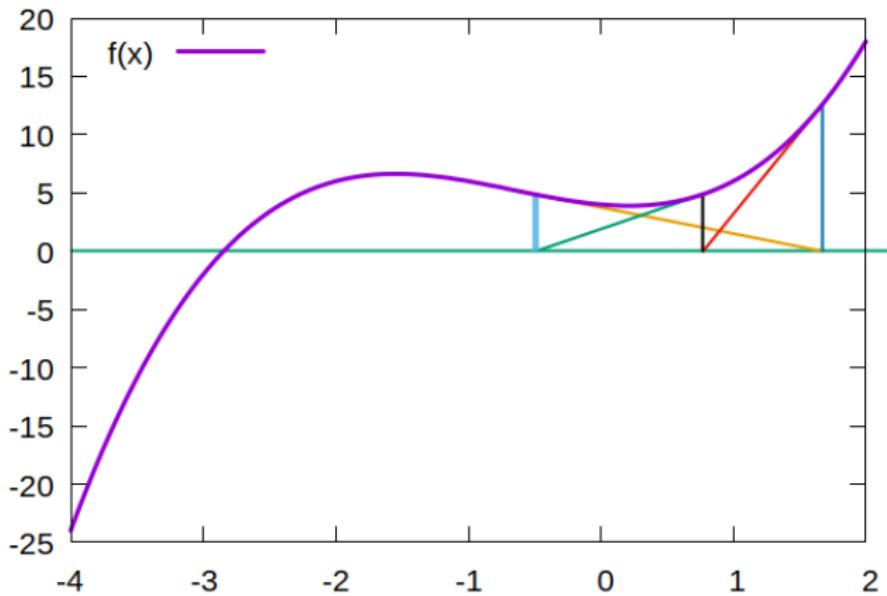
Newton's method - a bad choice

$$f(x) = x^3 + 2x^2 - x + 4, f' = 3x^2 + 4x - 1$$



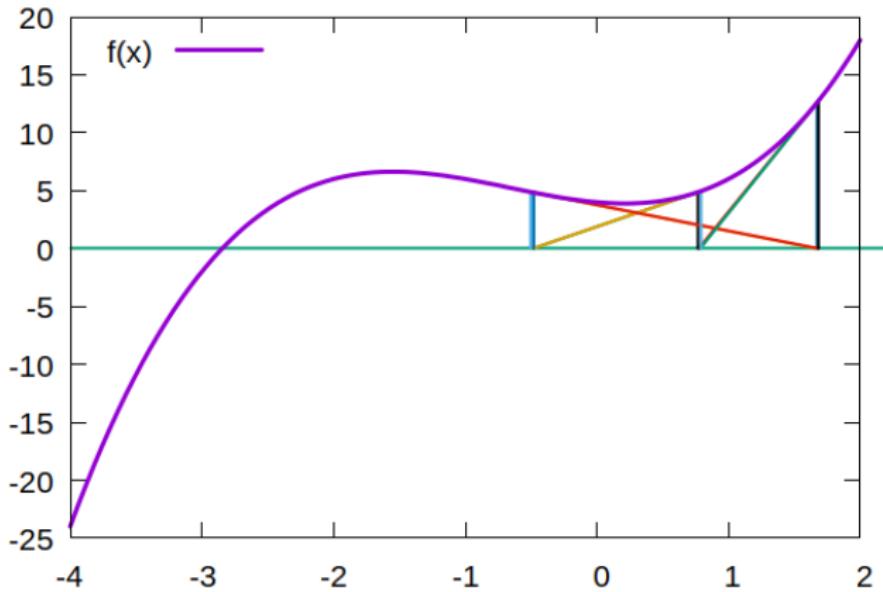
Newton's method - a bad choice

$$f(x) = x^3 + 2x^2 - x + 4, f' = 3x^2 + 4x - 1$$

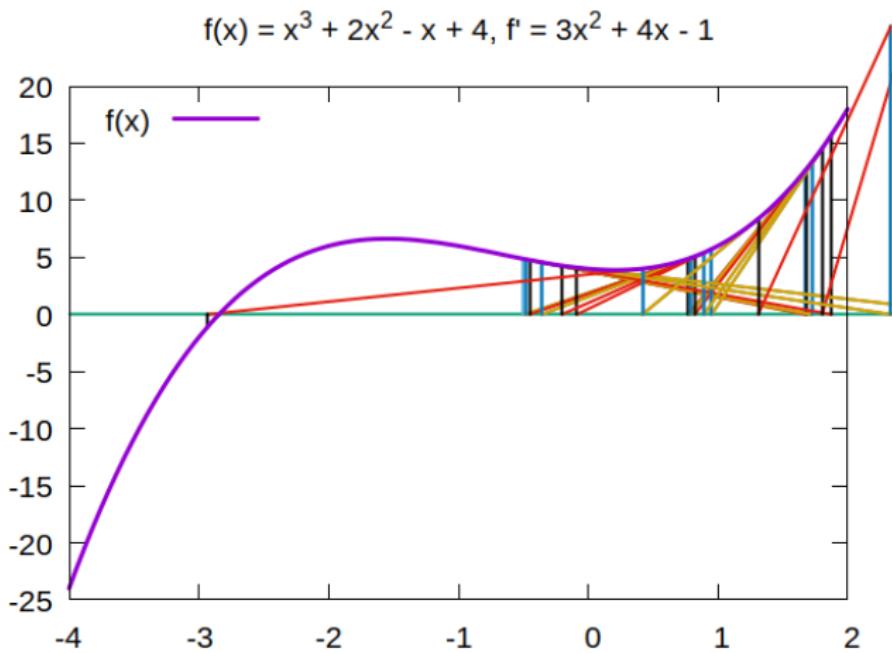


Newton's method - a bad choice

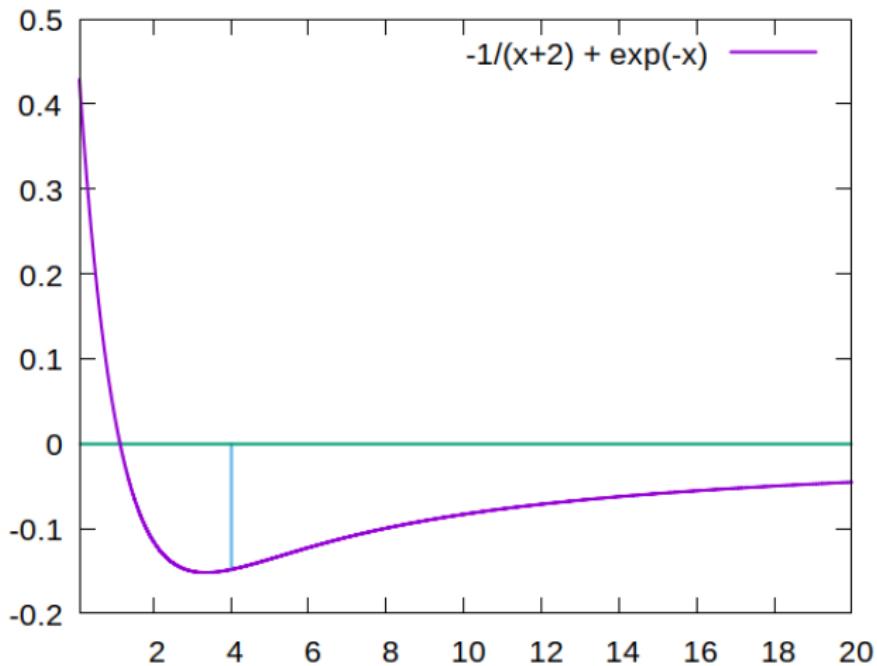
$$f(x) = x^3 + 2x^2 - x + 4, f' = 3x^2 + 4x - 1$$



Newton's method - a bad choice



Newton's method - a worse choice



Runge-Kutta methods

- Much of the continuum courses is going to focus on finding solutions to partial differential equations, e.g.

$$\frac{\partial u_i}{\partial t} + \nabla_j f_{ij}(\mathbf{u}) = 0$$

- These techniques will often follow the **method of lines**, in which an approximation to the space derivative is made first, and this is used to solve the time derivative
- A lot of effort focuses on the space derivative, and in doing so, deals with the really complex aspects of the system
- There are then many fewer methods to required to deal with the time derivative, though they do tend to get glossed over
- Runge-Kutta methods are perhaps the most useful

Runge-Kutta methods

$$\frac{\partial u_i}{\partial t} + \nabla_j f_{ij}(\mathbf{u}) = 0$$

- We shall assume that we know $F_i(\mathbf{u}) = -\nabla_j f_{ij}(\mathbf{u})$
- At which point, each entry in \mathbf{u} is now independent; we shall drop the index
- Runge-Kutta methods are designed to solve

$$\frac{\partial u}{\partial t} = F(u, t)$$

- They are an iterative method,

$$u_{n+1} = u_n + h \sum_{i=1}^s b_i k_i$$

Runge-Kutta methods

$$u_{n+1} = u_n + h \sum_{i=1}^s b_i k_i$$

- We assume that at a give time, t_n , we know the solution, $u_n = u(t_n)$
- We want to know the solution at a future time, $u_{n+1} = u(t_{n+1}) = u(t_n + h)$
- b_i are a series of constant coefficients
- k_i are a series of successive evaluations of $F(u)$, with additional constants c_i and a_{ij}

$$k_1 = F(u_n, t_n)$$

$$k_2 = F(u_n + h(a_{21}k_1), t_n + hc_2)$$

$$k_3 = F(u_n + h(a_{31}k_1 + a_{32}k_2), t_n + hc_3)$$

$$\vdots$$

$$k_s = F(u_n + h(a_{s1}k_1 + a_{s2}k_2 + \dots + a_{s,s-1}k_{s-1}), t_n + hc_s)$$

Euler method

- Identifying constants for a Runge-Kutta method is not always straightforward - there may be accuracy or efficiency issues
- However, some of the simpler methods can be derived fairly straightforwardly
- Recall our first-order derivative approximation

$$\frac{\partial u}{\partial x} \approx \frac{u(x+h) - u(x)}{h}$$

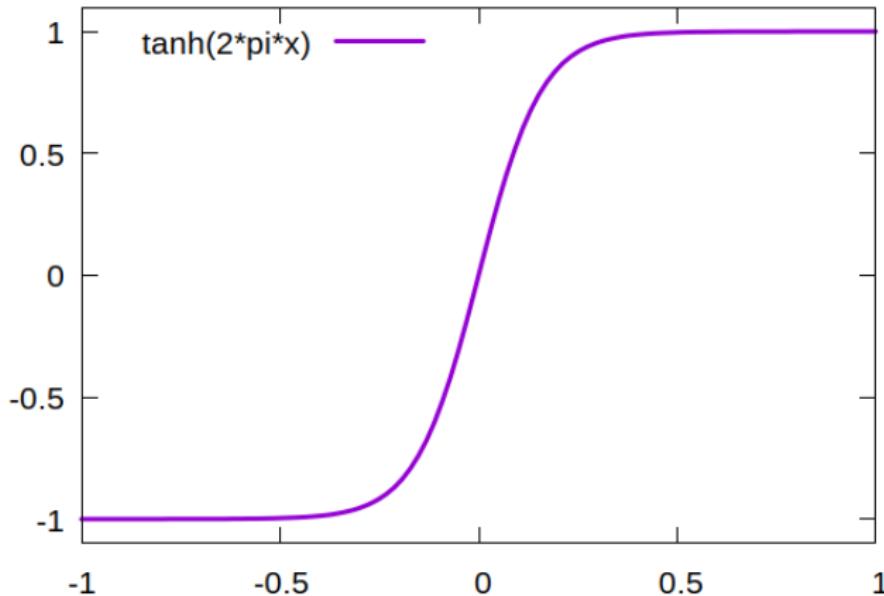
- Simple substitution gives

$$u_{n+1} = u_n + hF(u_n, t_n)$$

- Effectively, we assume the solution at t_{n+1} is a straight line based on the derivative, $F(u, t)$

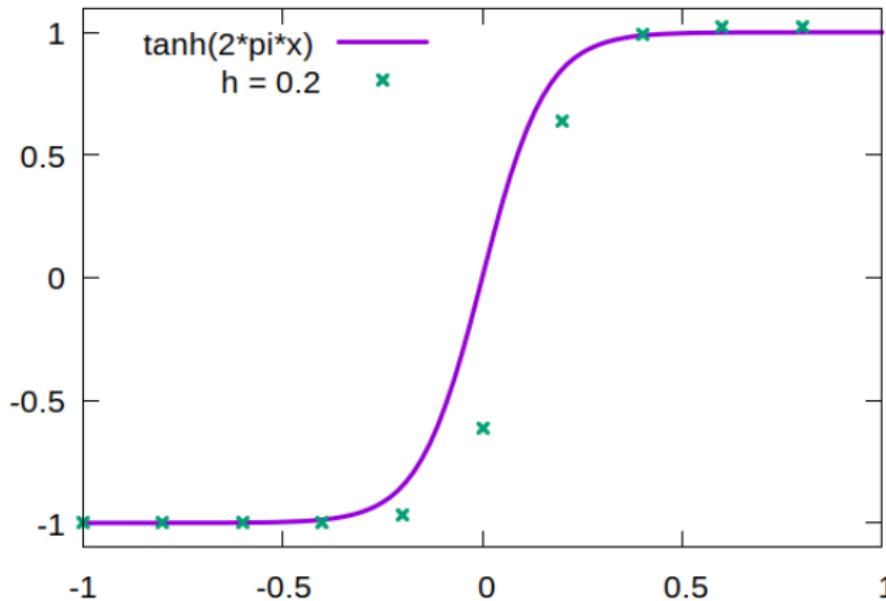
Euler method

$$\frac{du}{dx} = 2 \pi \operatorname{sech}^2(2\pi x)$$



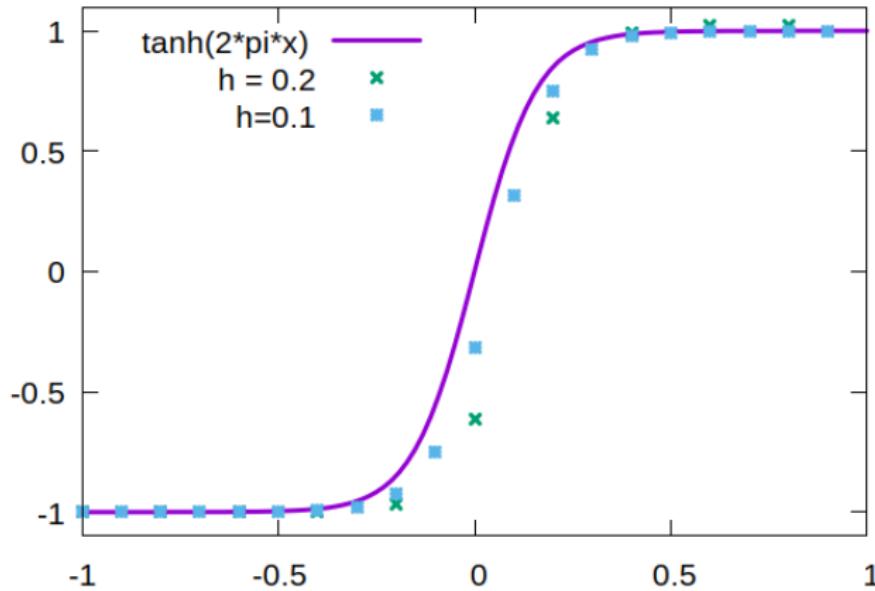
Euler method

$$\frac{du}{dx} = 2 \pi \operatorname{sech}^2(2\pi x)$$



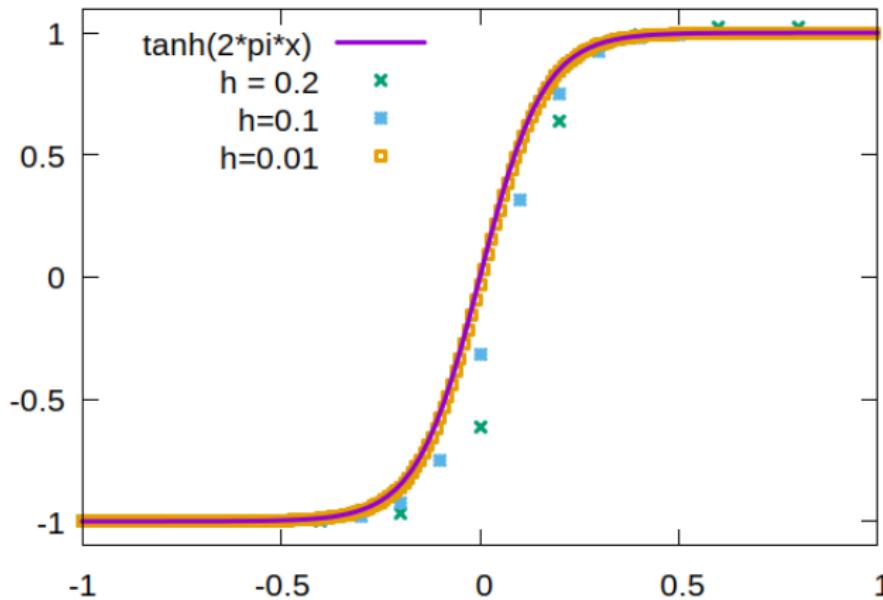
Euler method

$$\frac{du}{dx} = 2 \pi \operatorname{sech}^2(2\pi x)$$



Euler method

$$\frac{du}{dx} = 2 \pi \operatorname{sech}^2(2\pi x)$$



Midpoint method

- By starting with a second-order accurate derivative approximation, the **midpoint method** is reached

$$u_{n+1} = u_n + hF\left(u_n + \frac{h}{2}F(u_n, t_n), t_n + \frac{h}{2}\right)$$

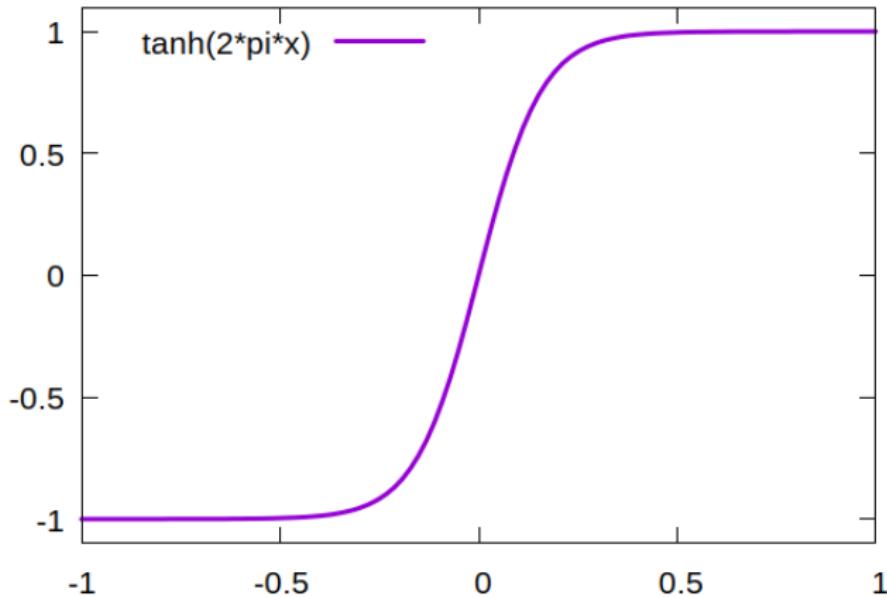
- This can be thought of as a two-step process
- First, we approximate the derivative at $t + h/2$,

$$F\left(u_n + \frac{h}{2}F(u_n, t_n), t_n + \frac{h}{2}\right)$$

- Then we use this approximation to advance to t_{n+1} in the same manner as we did for the Euler method

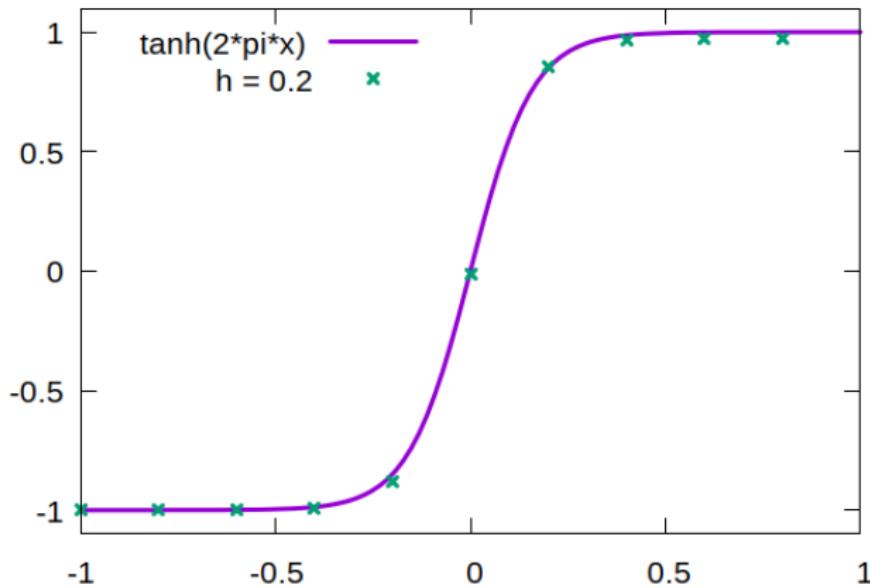
Midpoint method

$$\frac{du}{dx} = 2 \pi \operatorname{sech}^2(2\pi x)$$



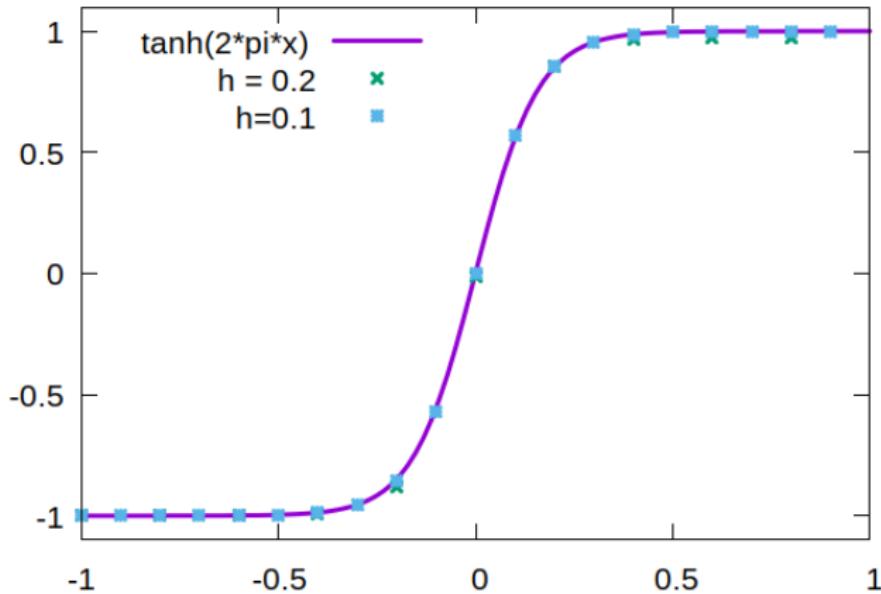
Midpoint method

$$\frac{du}{dx} = 2 \pi \operatorname{sech}^2(2\pi x)$$



Midpoint method

$$\frac{du}{dx} = 2 \pi \operatorname{sech}^2(2\pi x)$$



Butcher tableaux

$$u_{n+1} = u_n + h \sum_{i=1}^s b_i k_i$$

$$k_1 = F(u_n, t_n)$$

$$k_2 = F(u_n + h(a_{21}k_1), t_n + hc_2)$$

$$\vdots$$

$$k_s = F(u_n + h(a_{s1}k_1 + a_{s2}k_2 + \dots + a_{s,s-1}k_{s-1}), t_n + hc_s)$$

- The methods considered so far fit into this form
- For the Euler method, we have $b_1 = 1$
- For the midpoint method, we have $b_1 = 0$, $b_2 = 1$, $a_{21} = \frac{1}{2}$ and $c_2 = \frac{1}{2}$
- There are many other Runge-Kutta methods, in order to make things slightly easier to read, these constants can be written in a **Butcher tableau**

Butcher tableaux

$$u_{n+1} = u_n + h \sum_{i=1}^s b_i k_i$$

$$k_1 = F(u_n, t_n)$$

$$k_2 = F(u_n + h(a_{21}k_1), t_n + hc_2)$$

$$\vdots$$

$$k_s = F(u_n + h(a_{s1}k_1 + a_{s2}k_2 + \dots + a_{s,s-1}k_{s-1}), t_n + hc_s)$$

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
\vdots	\vdots		\ddots		
c_s	a_{s1}	a_{s2}	\dots	$a_{s-1,s}$	a_{ss}
	b_1	b_2	\dots	b_{s-1}	b_s

Some Runge-Kutta examples

- Euler method (first-order accurate):

0		
		1

- The (original) Runge-Kutta method (fourth-order accurate):

0				
$\frac{1}{2}$				
$\frac{1}{2}$				
$\frac{1}{2}$				
1				
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

- Midpoint method (second-order accurate):

0		
$\frac{1}{2}$		
$\frac{1}{2}$		
	0	1

- 3/8-rule (fourth-order accurate):

0				
$\frac{1}{3}$				
$\frac{2}{3}$				
$\frac{2}{3}$				
1				
	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

- Heun's method (second-order accurate)

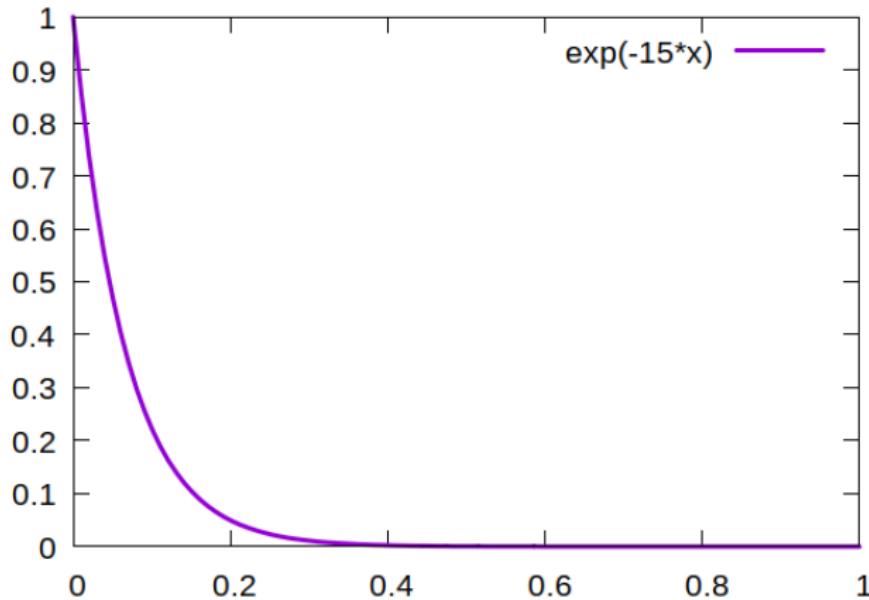
0		
1		
	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{1}{2}$	$\frac{1}{2}$

Why are there so many?

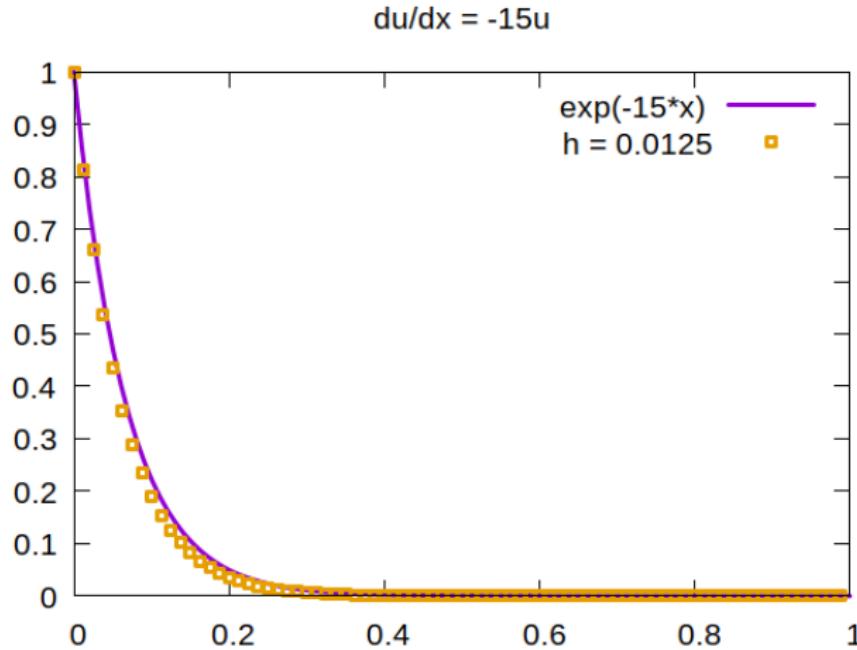
- As with any numerical method, the choice of Runge-Kutta method used is a balance between accuracy, efficiency and stability
- Runge-Kutta methods are often applied as part of a numerical method for PDE's, ideally the accuracy of the numerical method and the RK method should be the same
- The 3/8 rule has lower **absolute error** than the original Runge-Kutta method
- However, the original method can be written in such a way that k_1 , k_2 etc. do not need to be stored
- This can represent a large saving in computational memory (if an array of k values is needed)
- If a Runge-Kutta method does not produce a good solution (for example, it oscillates wildly), this could be because it is **unstable**
- We shall consider stability later, but for these methods, this stability can be dependent on the form of $F(u, t)$

Unstable behaviour

$$\frac{du}{dx} = -15u$$

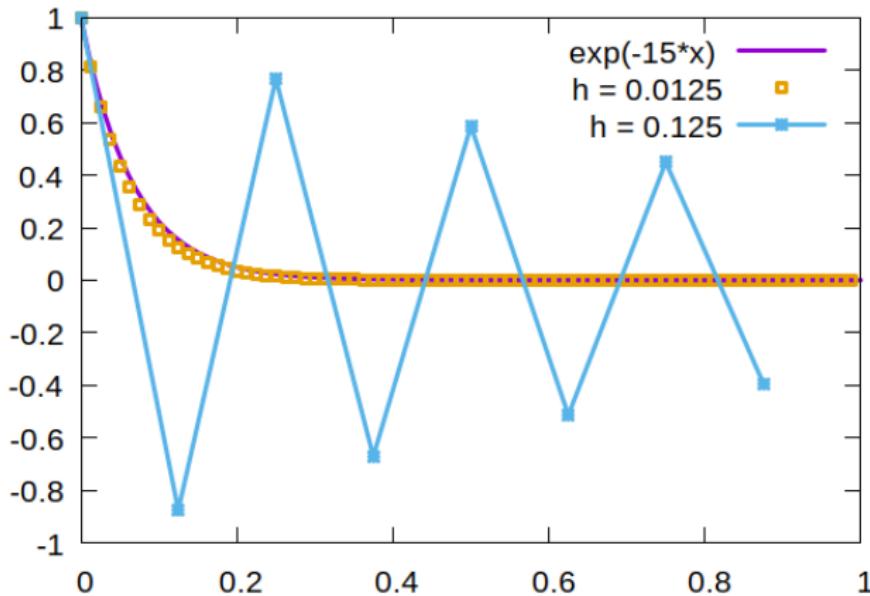


Unstable behaviour



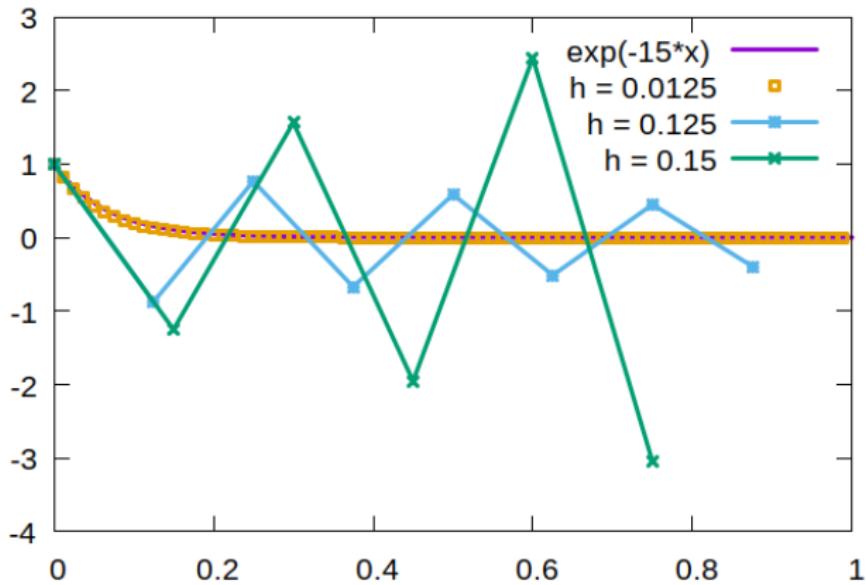
Unstable behaviour

$$\frac{du}{dx} = -15u$$



Unstable behaviour

$$du/dx = -15u$$



Summary

- We have covered some of the underlying mathematics which will appear repeatedly throughout the continuum courses
- Some of these will be built upon in greater detail, e.g. thermodynamics for equations of state, and discretisation techniques
- Anything in this course should be considered examinable - these methods could be required in any of the continuum exams
- However, this will always be in context of the course itself, it will not come as a surprise