Lecture 1:

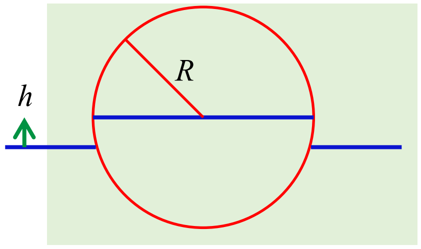
De-dimensionalisation

(simplifying your model)

Dealing with physical units

- Each physical quantity has units (dimensions). Your starting model is likely to have variables and parameters measured in physical units.

Example (ODE): A model of a buoy



$$\frac{2}{3}\pi R^3\rho\frac{d^2h}{dt^2} + \alpha\frac{dh}{dt} + \rho g\pi R^2 h - \frac{1}{3}\rho g\pi h^3 = 0$$

mass*acceleration friction gravity and buoyant forces

Dealing with physical units

$$\frac{2}{3}\pi R^3 \rho \frac{d^2h}{dt^2} + \alpha \frac{dh}{dt} + \rho g \pi R^2 h - \frac{1}{3}\rho g \pi h^3 = 0$$

- **Each term in this equation must have the same units** of $\mathrm{kg{\cdot}m{\cdot}s}^{-2} = \mathrm{N}$
- Have a look at each variable and parameter separately:

$\boldsymbol{h}$ - Position in m

$\boldsymbol{t}$ - Time in s

$\boldsymbol{\rho}$ - Density in kg/m^3

$\boldsymbol{R}$ - Radius in m

$\boldsymbol{g}$ - Free fall acceleration in m/s^2

$\boldsymbol{\frac{dh}{dt}}$ - First derivative (velocity) in …

$\boldsymbol{\frac{d^2h}{dt^2}}$ - Second derivative (acceleration) in …

$\boldsymbol{\alpha}$ - friction coefficient in …

From a theorist's perspective…

$$\frac{2}{3}\pi R^3 \rho \frac{d^2h}{dt^2} + \alpha \frac{dh}{dt} + \rho g \pi R^2 h - \frac{1}{3}\rho g \pi h^3 = 0$$

- **This is an equation for $\boldsymbol{h(t)}$**

VARIABLES:

$\boldsymbol{h}$ – dependent variable

$\boldsymbol{t}$ – independent variable

PARAMETERS:

$R, \alpha, \rho, (g)$ – defined by the properties of the system

- This is a 2nd order ODE problem => **requires 2 auxiliary conditions**

EITHER:

Initial Value Problem (IVP):

specify $h(t=0)$ and $\frac{dh}{dt}(t=0)$

OR:

Boundary Value Problem (BVP):

specify h or $\frac{dh}{dt}$ at each boundary of the domain of the independent variable. E.g.

$\frac{dh}{dt}(t=0) = A\,, \quad h(t=T) = B$

For PDE problems can have combinations of IVP/BVP for each independent variable

- This is a damped oscillator problem with a nonlinear term

Use your knowledge of physics and math to establish useful connections: crucial for choosing a proper numerical method and testing and debugging your codes!

De-dimensionalisation

- Systematically reduce number of parameters by making each term in the equation dimensionless
- Also makes generic behaviour easier to spot.

$$\frac{2}{3}\pi R^3 \rho \frac{d^2h}{dt^2} + \alpha \frac{dh}{dt} + \rho g \pi R^2 h - \frac{1}{3}\rho g \pi h^3 = 0$$

$\boldsymbol{h}$ – dependent variable

$\boldsymbol{t}$ – independent variable

1) Define dimensionless versions of the variables

$$h = Hy, \quad t = Tx$$

Here y is dimensionless position, and H is a scale that we will define later

Similarly x is dimensionless time, and T is a timescale that we will define later

2) Substitute in the original equation

A de-tour: dealing with derivatives

$$h = Hy, \quad t = Tx$$

Hence what is $\frac{dh}{dt}, \frac{d^2h}{dt^2}, \; \frac{d^nh}{dt^n}$?

$$\frac{2}{3}\pi R^3\rho\frac{d^2h}{dt^2}+\alpha\frac{dh}{dt}+\rho g\pi R^2h-\frac{1}{3}\rho g\pi h^3=0 \qquad h = Hy, \quad t = Tx$$

2) Substitute in the original equation

$$\frac{2}{3}\pi R^3\rho\frac{H}{T^2}\frac{d^2y}{dx^2}+\alpha\frac{H}{T}\frac{dy}{dx}+\rho g\pi R^2Hy-\frac{1}{3}\rho g\pi H^3y^3=0$$

3) Make the equation dimensionless. There are choices, e.g.

$$\div\frac{2}{3}\pi R^3\rho\frac{H}{T^2} \quad \rightarrow \quad \frac{d^2y}{dx^2}+\boxed{\frac{3}{2}\frac{\alpha T}{\pi R^3\rho}}\frac{dy}{dx}+\boxed{\frac{3gT^2}{2R}}y-\boxed{\frac{gH^2T^2}{2R^3}}y^3=0$$

This term is dimensionless, so are all

Moreover, since $\frac{dy}{dx}, y, y^3$ are all dimensionless (just numbers), so are all the

dimensionless products: $\frac{3}{2}\frac{\alpha T}{\pi R^3\rho}, \frac{3gT^2}{2R}, \frac{gH^2T^2}{2R^3}$

4) Fix scales to simplify the equation

$$\frac{d^2y}{dx^2} + \frac{3}{2}\frac{\alpha T}{\pi R^3 \rho}\frac{dy}{dx} + \frac{3gT^2}{2R}y - \frac{gH^2T^2}{2R^3}y^3 = 0$$

E.g. can set $\frac{3gT^2}{2R} = 1$ by selecting the timescale $T = \sqrt{\frac{2R}{3g}}$

Then set $\frac{gH^2T^2}{2R^3} = 1$ by selecting the scale $H = \sqrt{\frac{2R^3}{gT^2}} = \sqrt{\frac{2R^3}{g}\frac{3g}{2R}} = \sqrt{3}R$

=> The dimensionless equation becomes:

$$\frac{d^2y}{dx^2} + a\frac{dy}{dx} + y - y^3 = 0$$

with the single dimensionless parameter

$$a = \frac{3}{2}\frac{\alpha T}{\pi R^3 \rho} = \frac{3}{2}\frac{\alpha}{\pi R^3 \rho}\sqrt{\frac{2R}{3g}} = \sqrt{\frac{3}{2}}\frac{\alpha}{\pi R^{5/2}\rho\sqrt{g}}$$

Benefits of using the de-dimensionalisation procedure

$$\frac{2}{3}\pi R^3 \rho \frac{d^2 h}{dt^2} + \alpha \frac{dh}{dt} + \rho g \pi R^2 h - \frac{1}{3}\rho g \pi h^3 = 0$$

with 3(4) parameters: $R, \alpha, \rho, (g)$

VS

$$\frac{d^2 y}{dx^2} + a \frac{dy}{dx} + y - y^3 = 0$$

with one parameter $a = \sqrt{\frac{3}{2}} \frac{\alpha}{\pi R^{5/2} \rho \sqrt{g}}$

- Reduced number of parameters: easier to explore possible solutions.
  Crucial for tasks like parameter optimization.
- Helps to reveal the important dependencies and identify the generic behaviour

Eg the limit of weak damping is achieved when $\frac{\alpha}{R^{5/2}\rho\sqrt{g}} \ll 1$

Also, systems with the same ratio $\frac{\alpha}{R^{5/2}\rho\sqrt{g}}$ *will display the same behaviour (up to the rescaling factors), as they correspond to the same dimensionless system*

- Easier to work with (certainly) analytically and (arguably) numerically

Some useful tips

1) Don’t forget to re-write the auxiliary conditions in terms of the dimensionless variables!

E.g. the Initial Value Problem:

$$\frac{2}{3}\pi R^3 \rho \frac{d^2h}{dt^2} + \alpha \frac{dh}{dt} + \rho g \pi R^2 h - \frac{1}{3}\rho g \pi h^3 = 0$$

$$h(t=0) = h_0, \quad \frac{dh}{dt}(t=0) = v_0$$

Transforms into:

$$h = Hy, \quad t = Tx$$

$$\frac{d^2y}{dx^2} + a\frac{dy}{dx} + y - y^3 = 0 \qquad a = \sqrt{\frac{3}{2}\frac{\alpha}{\pi R^{5/2}\rho\sqrt{g}}}$$

$$y(x=0) = h_0/H, \quad \frac{dy}{dx}(x=0) = v_0\frac{T}{H}$$

Some useful tips

2) Know how to transform your solution back to the physical variables!

$$h = Hy, \quad t = Tx$$

$$H = \sqrt{3}R\,, \mathrm{T} = \sqrt{\frac{2R}{3g}}$$

$$y = f(x) \quad \Rightarrow \quad h = \sqrt{3}\,Rf\left(\sqrt{\frac{2R}{3g}}\,x\right)$$

Note: *Systems with the same ratio* $\frac{\alpha}{R^{5/2}\rho\sqrt{g}}$ *will have the same solutions* $y(x)$*, but they will correspond to different solutions* $h(t)$ *due to the different scalings*

For example, if two buoys have $\alpha_2 = 2\alpha_1$ *and* $R_2 = 2^{2/5}R_1$*, their oscillations will be described by the same dimensionless function* $y(x)$*. But the physical amplitude of oscillation of the second buoy will be* $2^{2/5}$ *times larger than of the first buoy, while the period of the second buoy will be* $2^{1/5}$ *shorter.*

Some useful tips

3) In <u>linear homogeneous</u> equations, scaling of the dependent variable may be not helpful

E.g. wave equation

$$\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} + \frac{\partial^2 u}{\partial x^2} = 0$$

$$u = A\psi \quad \Rightarrow \quad \frac{1}{c^2}\frac{\partial^2 \psi}{\partial t^2} + \frac{\partial^2 \psi}{\partial x^2} = 0$$

same equation!

.. but can still make sense for systems of coupled equations

E.g.

$$\frac{du}{dt} = \beta_1 u + C_1 w$$

$$\frac{dw}{dt} = \beta_2 w + C_2 u$$

$$u = A\psi$$
$$w = B\phi$$

$$\Rightarrow$$

$$\frac{d\psi}{dt} = \beta_1 \psi + C_1 \frac{B}{A}\phi$$

$$\frac{d\phi}{dt} = \beta_2 \phi + C_2 \frac{A}{B}\psi$$

By selecting e.g. $\frac{A}{B} = \sqrt{C_1/C_2}$ can reduce it to the three-parameter (instead of the initial four-parameter) problem with the same coupling coefficient $q = \sqrt{C_1 C_2}$

Some useful tips

4) In equations containing functions of a dimensionless variable *[eg $sin\,\theta$, $ln(u)$, etc] a scaling of this variable rarely works*

E.g. a pendulum equation

$$\ddot{\theta} + \sqrt{g/l}\,\sin\theta = 0$$

Note: θ is already dimensionless here (angle).
The argument of a function must always be dimensionless!

Try $\theta = A\alpha$

$$\ddot{\alpha} + \frac{1}{A}\sqrt{g/l}\ \sin(A\alpha) = 0$$

Hmmm, that 's hardly an improvement!

Alternative way of de-dimensionalisation

- Choose scales suitable for the problem at the start

E.g. for the buoy problem you may wish to select the typical amplitude and oscillation period scales:

$H = 1cm\ ,\ T = 250\ ms$

You will not change the number of parameters, but you can work with more comfortable ranges of variables, e.g. by ensuring that all the values are $O(1)$

This option is often used when you work with extremely large (astro) or extremely small (nanoscience, optics, quantum) scales.

How this topic will be assessed:

- Coursework

- Typical questions on exam:

1) For a given equation(s) written in physical units:

- *Identify variables and parameters;*
- *Use de-dimensionalisation to either (i) reduce to the desired form, or (ii) minimize the number of parameters;*
- *Formulate the auxiliary conditions in terms of the dimensionless variables.*

2) Show that a given equation can be reduced to the desired form by scaling transformations

See further examples in worksheet 1

Lecture 2:

Grid-point discretisation

Discretisation

When using digital computers variables, functions, operators etc. must be represented DISCRETELY.

→ an extra source of approximation in our models

Must therefore consider extra issues:

- Discretisation error
- Stability of numerical method

We will consider 2 broad classes of discretisation;

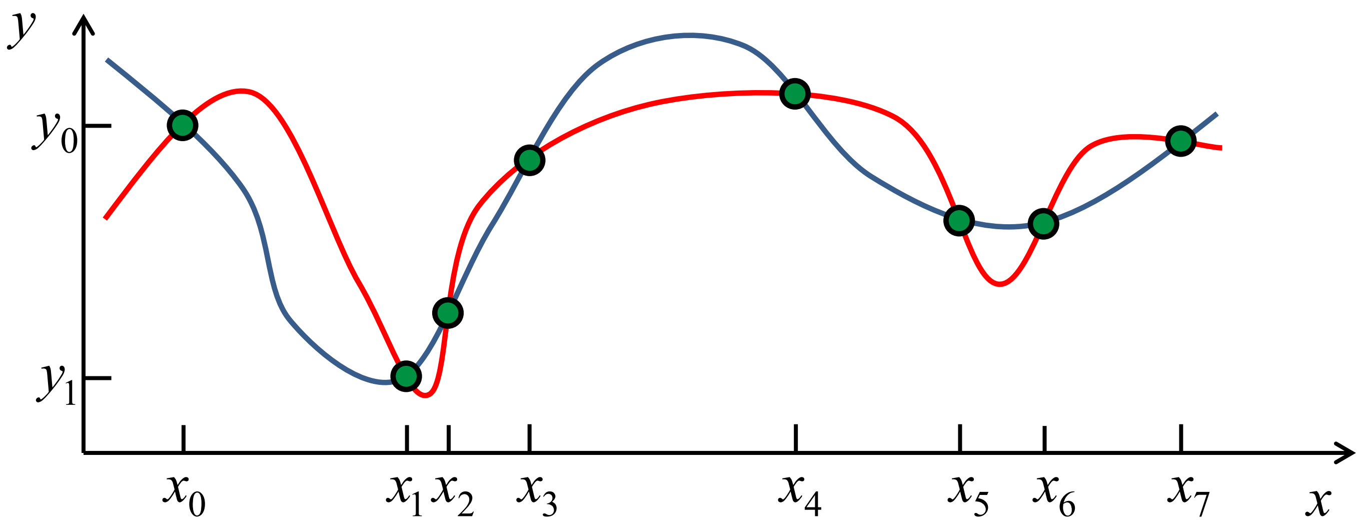
via grids and via basis sets.

Grid-point discretisation

Represent function $y(x)$ and its derivatives $\frac{dy}{dx}, \quad \frac{d^2y}{dx^2} \dots$ using

only values of y at a discrete set of points $x_0, x_1, x_2 \dots \equiv \{x_i\}$

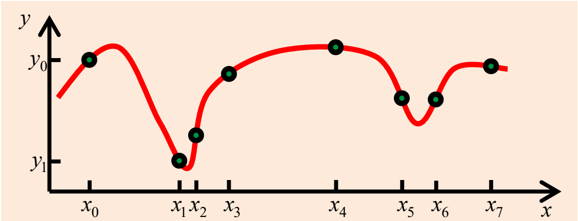
(a) Functions



Continuous $y(x)$ represented by $y(x_0), y(x_1), y(x_2) \dots$;
write $y(x_i) = y_i$, and the set of discrete values as $\{y_i\}$.

Many functions have same $\{y_i\}$, so issue with accuracy.

(b) Derivatives



Use Taylor series

$$y(x+a) = y(x) + ay'(x) + \frac{a^2}{2} y''(x) + \ldots + \frac{a^n}{n!} y^{(n)}(x) + \ldots \qquad ①$$

to approximate derivatives at grid points, using differences between the y_i

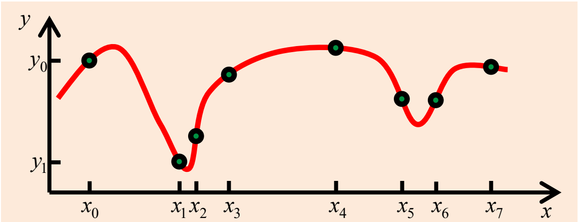
eg $y'(x)$: At grid point x_i, with $x_{i+1} = x_i + a$, ① gives

$$y(x_{i+1}) = y(x_i) + ay'(x_i) + \frac{a^2}{2} y''(x_i) + \ldots$$

$$\Rightarrow y'(x_i) = \frac{1}{a}\left\{ y(x_{i+1}) - y(x_i) - \frac{a^2}{2} y''(x_i) - \ldots \right\}$$

or, in shorthand: $y_i' = \frac{1}{a}\left\{ y_{i+1} - y_i - \frac{a^2}{2} y_i'' - \ldots \right\}$

$$y_i' = \frac{1}{a}\left\{ y_{i+1} - y_i - \frac{a^2}{2} y_i'' - \dots \right\}$$



Terms from here are unknown.
OMIT them all, to yield the
"forward difference approximation" (FDA) to $y'(x_i)$:

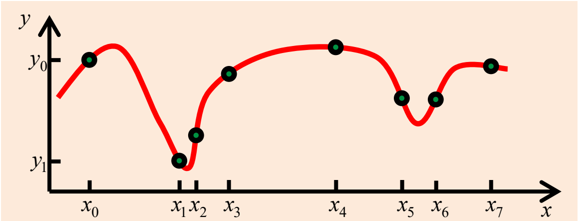
$$y_i' \approx \frac{1}{a}\left\{ y_{i+1} - y_i \right\}.$$

The discretisation error ε contains all the terms we omitted.

Here $$\varepsilon = \frac{1}{a}\left\{ \frac{a^2}{2} y_i'' + \frac{a^3}{6} y_i''' + \dots \right\}$$

terms get smaller (if a is small)

So for small a, the first omitted term ($\propto a$) dominates ε;
we say that the FDA has discretisation error ε which is $O(a)$.



Similarly, use Taylor

$$y(x-a) = y(x) - ay'(x) + \frac{a^2}{2}y''(x) + \cdots + (-\mathbf{1})^{\boldsymbol{n}}\frac{a^n}{n!}y^{(n)}(x) + \ldots \quad (2)$$

to obtain another approximation for the derivative at grid point x_i:

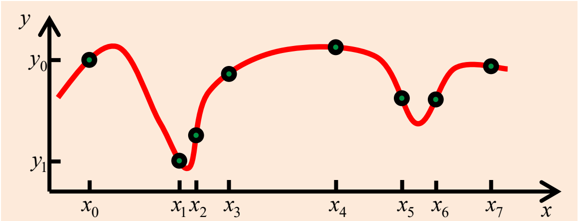
At grid point x_i , with $x_{i-1} = x_i - a$, (2) gives

$$y(x_{i-1}) = y(x_i) - ay'(x_i) + \frac{a^2}{2}y''(x_i) + \ldots$$

$$\Rightarrow y'(x_i) = \frac{1}{a}\left\{y(x_i) - y(x_{i-1}) + \frac{a^2}{2}y''(x_i) - \ldots\right\}$$

or, in shorthand: $y_i' = \frac{1}{a}\left\{y_i - y_{i-1} + \frac{a^2}{2}y_i'' - \ldots\right\}$

$$y_i' = \frac{1}{a}\left\{y_i - y_{i-1} + \frac{a^2}{2}y_i'' - \ldots\right\}$$



Again,

OMIT all terms from here, to yield the "backward difference approximation" (BDA) to $y'(x_i)$:

$$y_i' \approx \frac{1}{a}\{y_i - y_{i-1}\}.$$

It is easy to see that BDA also has discretisation error $O(a)$.

$O(a)$ errors aren't very good. If $a \to \frac{a}{2}, \quad \varepsilon \to \frac{\varepsilon}{2}$.

Can usually do (a bit) better for not much effort.

Let us write the two versions of Taylor series side by side:

$$y(x+a) = y(x) + ay'(x) + \frac{a^2}{2}y''(x) + \cdots + \frac{a^n}{n!}y^{(n)}(x) + \ldots \quad ①$$

$$y(x-a) = y(x) - ay'(x) + \frac{a^2}{2}y''(x) + \cdots + (-\mathbf{1})^{n}\frac{a^n}{n!}y^{(n)}(x) + \ldots \quad ②$$

①-② yields a "centred-difference approximation" (CDA)

$$y_i' \approx \frac{1}{2a}\left\{y_{i+1} - y_{i-1}\right\}.$$

Since the $\frac{a^2}{2}y''(x)$ terms cancel, ε is now $O(a^2)$.

Note that a **regularly-spaced grid** is a simple way to produce an $O(a^2)$ approximation to this derivative.

Let us summarize the three formulas we obtained for $y'(x_i)$:

Type	Formula	Discretisation error
y', FDA	$y_i' \approx \frac{1}{a}\{y_{i+1} - y_i\}$	$O(a)$
y', BDA	$y_i' \approx \frac{1}{a}\{y_i - y_{i-1}\}$	$O(a)$
y', CDA	$y_i' \approx \frac{1}{2a}\{y_{i+1} - y_{i-1}\}$	$O(a^2)$

Notes: 1) regular grid is required for CDA

2) FDA and BDA are useful for initial value problems (more details later!)

What about $y''(\mathrm{x})$?

$$y(x+a) = y(x) + ay'(x) + \frac{a^2}{2}y''(x) + \cdots + \frac{a^n}{n!}y^{(n)}(x) + \ldots \quad (1)$$

$$y(x-a) = y(x) - ay'(x) + \frac{a^2}{2}y''(x) + \cdots + (-1)^n \frac{a^n}{n!}y^{(n)}(x) + \ldots \quad (2)$$

(1)+(2) gives

$$y_{i+1} + y_{i-1} = 2y_i + a^2 y''(x_i) + \frac{a^4}{12}y''''(x_i) + \cdots.$$

And so we obtain:

$$y''(x_i) = \frac{1}{a^2}\left[y_{i+1} + y_{i-1} - 2y_i - \frac{a^4}{12}y''''(x_i) - \cdots\right]$$

Omit terms from here.

Easy to see that the error is $O(a^2)$

A useful set of formulas for discrete derivatives:

Type	Formula	Error
y', FDA	$y_i' \approx \frac{1}{a}\{y_{i+1} - y_i\}$	$O(a)$
y', BDA	$y_i' \approx \frac{1}{a}\{y_i - y_{i-1}\}$	$O(a)$
y', CDA	$y_i' \approx \frac{1}{2a}\{y_{i+1} - y_{i-1}\}$	$O(a^2)$
y'', CDA	$y''(x_i) \approx \frac{1}{a^2}[y_{i+1} + y_{i-1} - 2y_i]$	$O(a^2)$

Note: regular grid is required for CDA formulas!

- *A side note:* One could attempt to obtain $y''(x_i)$ by applying recursively the earlier derived formulas for y':

$$y''(x) = \frac{d}{dx}[y'(x)] \quad \Rightarrow \quad y''(x_i) \approx \frac{1}{2a}[y'(x_{i+1}) - y'(x_{i-1})]$$

with $\quad y'(x_i) \approx \dfrac{1}{2a}[y(x_{i+1}) - y(x_{i-1})]$

It is easy to see [check it yourself!] that this is equivalent to

$$y''(x_i) \approx \frac{1}{4a^2}[y_{i+2} + y_{i-2} - 2y_i]$$

I.e. we obtained exactly the same formula as we derived earlier, only with a twice larger discretization step $a \to 2a$

Hence the error is 4 times larger - do not do this!

Computational cost is the reason to chase $\varepsilon \sim O(a^2)$

Express in terms of required memory M and time T

Will see later that to solve a discretised problem with N grid points

$$M \sim N^2 \qquad T \sim N^3$$

So for an ODE with fixed domain and regular grids,
halving the grid spacing $a \to a/2$ requires $N \to 2N$ points
leading to $\boldsymbol{M \to 4M}$ and $\boldsymbol{T \to 8T}$

- an expensive change if it only halves ϵ !

- $\varepsilon \sim O(a^2)$ is not great, but MUCH better than $\varepsilon \sim O(a)$

An example: discretize the following Boundary Value Problem

$$\frac{dy}{dt} + t^2\, y = \cos(2\pi t)\ , \qquad y(0) = 0, \qquad \frac{dy}{dt}(1) = 3$$

Natural vs Essential auxiliary conditions

- Auxiliary conditions are an important part of the problem. They are as important as the equation you are trying to solve!
- One and the same equation with different auxiliary conditions corresponds to different physical setups and may require different numerical methods
- Some auxiliary conditions you will need to implement explicitly in your numerical model – such conditions are called ESSENTIAL conditions
- Some auxiliary conditions will be implemented automatically by using a particular numerical technique – NATURAL conditions

In point-grid discretisation:

The zero Dirichlet boundary condition y(x=a)=0 is the NATURAL boundary condition

All other (including non-zero Dirichlet) boundary conditions are essential.

Finite Difference Method

- Solving ODEs and PDEs with grid-point discretization is generally known as the **Finite Difference Method (FDM)**
- FDM reduces differential equations to a set of algebraic equations for the values of the unknown function at grid points. This can then be solved numerically using well-established methodology: either linear algebra methods (for linear problems), or various iteration methods (for nonlinear problems).
- There are many approximations for derivatives via finite differences. Most common are the central-difference based involving nearest neighbours, but other options are also available.
- Always need to consider the cost (in terms of memory and computation time) required to reduce the computation error.
- **FDM** is the easiest method to implement (hence it is the most popular method) but it has its own drawbacks, especially in higher-dimensional problems. We will discuss this in more details in Part 2 of the course.

How this topic will be assessed:

- Coursework

- Typical questions on exam:

Discretise a given Boundary Value Problem ODE:

- *Specify the grid step size, what coordinates the first and the last grid point correspond to, justify your choices;*
- *Write down a generic discretised equation for a grid point away from the boundaries;*
- *Write down the equations for the two boundary points;*
- *Quantify the discretisation error.*

See further examples in worksheet 1

Lecture 3:

Basis-set discretisation

Basis-set discretisation

Expand functions as a sum of BASIS FUNCTIONS.

$$f(t) = \sum_n c_n F_n(t) = c_1 F_1(t) + c_2 F_2(t) + c_3 F_3(t) + \cdots$$

- The sum is formally infinite, but will have to be truncated in numerics

- The (discrete) set of coefficients c_n describes the function

- BASIS functions are some known analytical functions.

Some popular choices are:

- Fourier series expansion *(periodic functions)*

- Hermite polynomials *(quantum harmonic oscillator type equations)*

- Hankel functions *(dispersion/diffraction in radially symmetric geometries)*

Basis-set discretisation

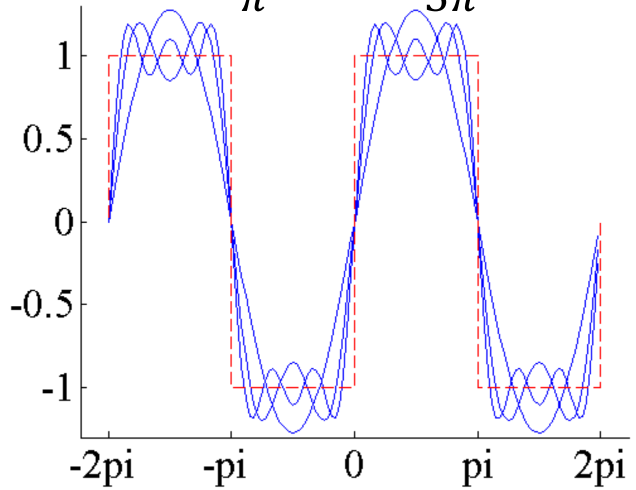
Expand functions as a sum of BASIS FUNCTIONS.

An example: Fourier series

$$f(t)=\sum_{n=-\infty}^{\infty} c_n e^{in\omega t}; \quad g(t)=\frac{a_0}{2}+\sum_{n=1}^{\infty}\left[a_n\cos(n\omega t)+b_n\sin(n\omega t)\right]$$

$f(t)$ must be periodic! $f(t+T)=f(t)\ \forall\ t,\ T=2\pi/\omega$

For a square wave $g(t)=\frac{4}{\pi}\sin t+\frac{4}{3\pi}\sin 3t+\frac{4}{5\pi}\sin 5t+\ldots$



$$T=2\pi; \quad \omega=1$$

Note that $g(t)$ is defined for ALL t.

Discretisation error (usually) arises once the sum is truncated.

Basis-set discretisation

How to choose a basis set?

$$f(t) = \sum_n c_n F_n(t) = c_1 F_1(t) + c_2 F_2(t) + c_3 F_3(t) + \cdots$$

Main criteria when selecting the set $F_n(t)$:

- (PREFERABLY) a COMPLETE and ORTHONORMAL set

(means you can expand ANY function)

(this is useful for deriving equations for c_n - see further examples)

- Consistency with the auxiliary conditions

[e.g. complex exponents $F_n(t) = \exp(in\omega t)$ satisfy periodic boundary conditions: $F_n\left(t + \frac{2\pi}{\omega}\right) = F_n(t)$]

- Speed of numerical conversion from/to the basis set

(i.e. for any function $f(t)$ you should be able to obtain the corresponding set of coefficients c_n, and the same in the opposite direction)

Differential operators act on the individual basis functions.

eg $$f(t) = \sum_n c_n \mathrm{e}^{in\omega t}\,; \quad \frac{df}{dt} = \sum_n in\omega c_n \mathrm{e}^{in\omega t}$$

So, to discretise an ODE via basis sets:

- Choose a set of basis functions
- Substitute expansion into ODE and re-write as an algebraic equation in the expansion coefficients (c_n)
- Truncate summation if computing…
- Make sure the auxiliary conditions are satisfied (can lead to additional conditions on c_n)

Complex Fourier series expansion

$$f(t) = \sum_{n=-\infty}^{\infty} c_n \mathrm{e}^{in\omega_0 t}$$

- Eigen-functions of differential operators

$$\frac{d^n}{dt^n}\exp(in\omega_0 t) = (in\omega_0)^n \exp(in\omega_0 t)$$

- Requires periodic boundary condition!

$$f(t+T) = f(t) \qquad \omega_0 = 2\pi/T$$

- The set is complete, and orthonormal:

$$e_l(t) = e^{il\omega_0 t}$$
$$e_m(t) = e^{im\omega_0 t}$$

$$\Rightarrow \frac{1}{T}\int_0^T e_l^*(t)e_m(t)dt = \frac{1}{T}\int_0^T e^{i(m-l)\omega_0 t}dt = \boxed{\delta_{l,m}}$$

Kronecker delta:

$$\delta_{l,m} = \begin{cases} 0, if\ l \neq m \\ 1, if\ l = m \end{cases}$$

(revise Fourier series course!)

An example (Boundary Value Problem):

$$\frac{dy}{dt} + t^2\, y = \cos(2\pi t)\ , \qquad\qquad y(0) = y(1)$$

- <u>NOTE:</u> This problem has **periodic BC** => Convenient to use complex Fourier series as the Basis set:

$$\mathrm{y}(t) = \sum_{n=-\infty}^{\infty} c_n \mathrm{e}^{in\omega_0 t} \qquad \text{select} \qquad \omega_0 = \frac{2\pi}{1} = 2\pi$$

Substitute into the equation:

$$\sum_n (in2\pi)\, c_n e^{in2\pi t} + t^2 \sum_n c_n e^{in2\pi t} = \cos(2\pi t)$$

$$\frac{dy}{dt} + t^2\, y = \cos(2\pi t)\ , y(0) = y(1)$$

$$\mathrm{y}(t) = \sum_{n=-\infty}^{\infty} c_n \mathrm{e}^{in2\pi t}$$

$$\sum_n (in2\pi)\, c_n e^{in2\pi t} + t^2 \sum_n c_n e^{in2\pi t} = \cos(2\pi t)$$

Apply "closure": multiply both sides by $\exp(-im2\pi t)$ and integrate over the period:

$$\sum_n (in2\pi) c_n \boxed{\int_0^1 \exp[i(n-m)2\pi t]dt} + \sum_n c_n \int_0^1 t^2 \exp[i(n-m)2\pi t]dt$$

$$= \int_0^1 \cos(2\pi t) \exp[-im2\pi t]dt$$

$= \delta_{n,m}$ **(orthogonality)**

$$\sum_n (in2\pi) c_n \int_0^1 \exp[i(n-m)2\pi t]dt = \sum_n (in2\pi) c_n \delta_{n,m} = im2\pi c_m$$

$$\frac{dy}{dt} + t^2\, y = \cos(2\pi t)\ , y(0) = y(1)$$

$$y(t) = \sum_{n=-\infty}^{\infty} c_n e^{in2\pi t}$$

$$\sum_n (in2\pi) c_n \int_0^1 \exp[i(n-m)2\pi t]dt + \sum_n c_n \int_0^1 t^2 \exp[i(n-m)2\pi t]dt$$
$$= \int_0^1 \cos(2\pi t) \exp[-im2\pi t]dt$$

$$\boxed{im2\pi c_m + \sum_n c_n K_{m-n} = F_m}$$

- this is a set of algebraic equations for the complex Fourier Series coefficients c_m of the solution $y(t)$

With the coefficients defined as:

$$K_{m-n} = \int_0^1 t^2 \exp[-i(m-n)2\pi t]dt$$

- this is $(\boldsymbol{m}-\boldsymbol{n})$th complex **Fourier Series coefficient** of the *(periodic version of)* function t^2

$$F_m = \int_0^1 \cos(2\pi t) \exp[-im2\pi t]dt$$

- this is $\boldsymbol{m}$th complex **Fourier Series coefficient** of the *(periodic version of)* right-hand side function $\cos(2\pi t)$

Note: *in this example the integrals are easy to tackle analytically and obtain all the coefficients. In real-life situation, you will need to compute such integrals numerically using Fast Fourier Transforms. (Revise year 2 Fourier series course?)*

$$\boxed{im2\pi c_m + \sum_n c_n K_{m-n} = F_m}$$

$$\frac{dy}{dt} + t^2\, y = \cos(2\pi t)\ , y(0) = y(1)$$

$$y(t) \approx \sum_{n=-N}^{N} c_n e^{in2\pi t}$$

Can write it in the matrix form:

$$\boxed{\widehat{M}\vec{c} = \vec{F},}$$

Note: *for numerical solution, we will have to truncate the infinite Fourier series at some (large enough) N.*

(2N+1)-component vector

$$\vec{c} = [c_{-N}, c_{-N+1}, \dots, c_0, c_1, \dots, c_N]^T$$ - **Unknown Fourier coeffs**

$$\vec{F} = [F_{-N}, F_{-N+1}, \dots, F_0, F_1, \dots, F_N]^T$$ - Fourier coeffs of the right-hand side func

- The truncation of the Fourier series **defines the discretisation error** of the Basis-set method

$$\widehat{M} = \begin{bmatrix} D_{-N} & K_{-1} & K_{-2} & \cdots & \cdots & \cdots & K_{-2N} \\ K_1 & D_{-N+1} & K_{-1} & K_{-2} & \cdots & \cdots & K_{-2N+1} \\ K_2 & K_1 & \ddots & \ddots & \ddots & \cdots & \cdots \\ \cdots & \cdots & K_1 & D_0 & K_{-1} & \cdots & \cdots \\ \cdots & \cdots & \dots & \ddots & D_1 & \ddots & \dots \\ \cdots & \cdots & \cdots & \dots & \ddots & \ddots & K_{-1} \\ K_{2N} & \cdots & \cdots & \cdots & K_2 & K_1 & D_N \end{bmatrix}$$ - $(2N+1) \times (2N+1)$ ***full matrix***

$$D_m = im2\pi + K_0 \qquad K_m = \int_0^1 t^2 e^{-ik2\pi t}\, dt$$ - Fourier coeffs of the variable coeff func t^2

Point-grid VS Basis-set discretisation

$$\frac{dy}{dt} + t^2\, y = \cos(2\pi t)\ , y(0) = y(1)$$

1. Discretize

N-point grid: $y_j = \mathrm{y}(t_j)$

Replace derivatives with finite differences

Implement essential BCs

Derive the set of equations for y_j

$$\boxed{\hat{M}\vec{y} = \vec{f}}$$

Basis-set: $\mathrm{y(t)} = \sum_n c_n \exp(in2\pi t)$

Make sure the chosen basis is compatible with BCs!

Truncate infinite sum e.g. for $-N \leq n \leq N$

Substitute and derive the set of equations for c_n

$$\boxed{\hat{M}\vec{c} = \vec{F}}$$

2. Solve (on a computer) **the set of linear algebraic equations**

3. Obtain values of the function at the grid points y_j

3. Obtain values of the expansion coefficients c_n

Grid

- Easy to setup,
including various auxiliary conditions

- Reduces the problem to the set of N coupled algebraic equations

(The resulting matrices are typically sparce => good for numerics!)

But in higher-dimensional problems the number of equations grows exponentially!

- **Solution is obtained at discrete points => may be problematic for post-processing**

Basis-set

- **Requires some effort to setup, take special care of:**
  - *auxiliary conditions;*
  - *choice of the basis set*

- Reduces the problem to the set of N coupled algebraic equations

(The resulting matrices are typically full => harder for numerics!)

(with a proper basis set choice) **easy to solve and/or have good convergence** *(i.e. require only few equations to solve)*

- **Solution is obtained in analytic form: convenient for post-processing BUT can have poor convergence**

- **Basis-set** discretisation is a powerful tool for **higher-dimensional** problems. In particular, <u>spectral and pseudo-spectral methods</u> for solving IVP PDE problems are based on basis-set discretisation.

How this topic will be assessed:

- Coursework

- Typical questions on exam:

Discuss advantages/disadvantages of Basis-set vs grid-point methods

Discretise a given Boundary Value Problem ODE using complex Fourier:

- *Write down the basis-set expansion;*
- *Substitute in the original equation, apply the "closure" procedure, derive the equations for the Fourier coefficients;*
- *Write down the equations in the matrix form;*
- *Explain the nature of the discretisation error.*

See further examples in worksheet 1