ME2 Computing- Session 6: Numerical solution of differential equations: boundary value problems

Learning outcomes:

- Being familiar with the Finite Difference scheme
- Being familiar with the types of boundary conditions
- Experience the Shooting method in conjunction with marching direct methods

Announcements: 1) Deadline for Quiz 5 is 15th January. 2) Deadline for Quiz 6 is doubled (as long tutorial sheet), and it will be two weeks after the session.

ODE with boundary values: Finite Difference Method

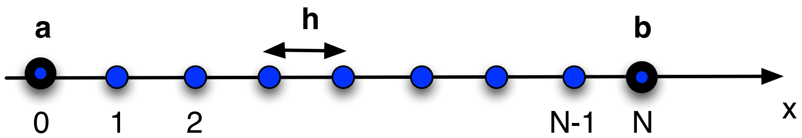
A second order boundary value problem ODE is specified by the ODE itself:

$$\frac{d^2y}{dx^2} + f(x)\frac{dy}{dx} + g(x)y = p(x)$$

by the domain of the solution, i.e. $a < x < b$, and by the boundary conditions.

The domain of the solution, $[a, b]$ is subdivided into N intervals and defined with $N + 1$ points (grid points). The derivatives of the ODEs are approximated at each grid point with the central difference finite difference scheme:

$\left.\frac{dy}{dx}\right\|_i = \frac{y_{i+1} - y_{i-1}}{2h}$	$\left.\frac{d^2y}{dx^2}\right\|_i = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$



The ODE is firstly approximated numerically at the *interior points*, $i = 1$ to $N - 1$, i.e.

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + f(x_i)\frac{y_{i+1} - y_{i-1}}{2h} + g(x_i)y_i = p(x_i)$$

Common terms are rearranged together as:

$$\left[\frac{1}{h^2} - \frac{f(x_i)}{2h}\right]y_{i-1} + \left[g(x_i) - \frac{2}{h^2}\right]y_i + \left[\frac{1}{h^2} + \frac{f(x_i)}{2h}\right]y_{i+1} = p(x_i) \quad i = 1 \dots N - 1$$

and the equation rewritten in compact form:

$$a_i y_{i-1} + b_i y_i + c_i y_{i+1} = p_i \quad i = 1 \dots N-1$$

This procedure generates a set of $N - 1$ (linear) algebraic equations: every equation, on the left-hand side, is formed by three terms only.

The boundary conditions at the endpoints are then added to assign the constrains on the solution:

$$y_0 = y_a$$
$$y_N = y_b$$

At the end, a set of $N + 1$ (linear) algebraic equations is formed altogether, with unknown variables $y_0, y_1, y_1, \dots . y_{N-2}, y_{N-1}, y_N$, i.e.:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & .. & .. & 0 \\ a_1 & b_1 & c_1 & 0 & 0 & .. & .. & 0 \\ 0 & a_2 & b_2 & c_2 & 0 & .. & .. & 0 \\ 0 & 0 & a_3 & b_3 & c_3 & .. & .. & 0 \\ .. & .. & .. & .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. & .. & .. & .. \\ 0 & 0 & 0 & 0 & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \\ \\ y_{N-1} \\ y_N \end{bmatrix} = \begin{bmatrix} y_A \\ p_1 \\ p_2 \\ p_3 \\ \\ \\ p_{N-1} \\ y_b \end{bmatrix}$$

These can then be computed either with a direct or an iterative method.
For a *direct* method, to invert the matrix, you can use you're the algorithm about Gauss elimination from last year Computing.
For an *iterative* method you can use either the Jacobi or the Gauss-Siedel method.

Task A: Direct methods

1. Write a function, *myodebc*, that receives the boudaries of the domain a and b, the value of the solutions at these points, $y(a) = y_a$ and $y(b) = y_b$, and the number N of desired intervals. *myodebc* returns the grid points x_i and the solution $y_i(x_i)$ at the grid points. The ODE is defined through an external function, *myfunc*, that receives the value of x and returns the values of $f(x)$, $g(x)$ and $p(x)$.

2. Solve the ODE $\frac{d^2y}{dx^2} + 2x\frac{dy}{dx} + 2y - \cos(3x) = 0$, in the domain $[0, \pi]$, with boundary conditions $y(0) = 1.5$ and $y(\pi) = 0$. Discretise the domain with 10 intervals. Plot $y(x)$.

3. Repeat the analysis with 100 intervals and compare it with the previous results.

Task B: Types of boundary conditions

The boundary conditions, specified at the two boundaries a and b, can be of different types:

- **Dirichlet**: two values for the solution, $y(x)$, are specified at a and b:
  $y(a) = BC_a$ and $y(b) = BC_b$ (this is the case for the example in Task A)

- **Neumann**: two values for the derivative of the solution, $\frac{dy}{dx}$, are specified a and b:
  $\frac{dy}{dx}(a) = BC_a$ and $\frac{dy}{dx}(b) = BC_b$

- **Mixed (or Robin)**: two values, for a combination of the solution, $y(x)$, and its derivative, $\frac{dy}{dx}$, are specified at a and b:
  $r_0 \frac{dy}{dx}(a) + r_1 y(a) = BC_a$ and $r_2 \frac{dy}{dx}(b) + r_3 y(b) = BC_b$

Note that to constrain the derivative at endpoint a it is necessary to implement the finite difference forward scheme, whilst at endpoint b the finite difference backward scheme is needed, instead.
Note also that by setting $r_0 = r_2 = 0$ the mixed boundary conditions become of Dirichlet type, and that by setting $r_1 = r_3 = 0$ the mixed boundary conditions become of Neumann type.

1. Modify the function, *myodebc*, to accommodate all the various types of boundary conditions. *myodebc* still receives the boundaries of the domain a and b, the boundary conditions at these points BC_a and BC_b, and the number N of desired intervals. In addition, it receives an array R of length 4, with the values of r_0, r_1, r_2, r_3, specifying the type of boundary conditions.

2. Solve the ODE $\frac{d^2y}{dx^2} + x\frac{dy}{dx} + y = 5x$, in the domain $[0,2]$, with boundary conditions $\left.\frac{dy}{dx}\right|_{x=0} = 0$ and $y(x = 2) = 5$. Discretise the domain with 50 intervals. Plot $y(x)$.

3. Repeat the calculation with boundary conditions $y(x = 0) = 5$ and $\left.\frac{dy}{dx}\right|_{x=2} = 0$.

Task C: Heat transfer in a nuclear fuel rod

The fuel rod of a nuclear reactor is a cylindrical structure with the fuel contained within a metal cladding. The heat is generated by the nuclear reaction in the fuel region and conducted, through the thickness of the cladding, to the outer surface of the cladding. Outside the cladding, cooling occurs with flowing water at temperature $T_w = 473K$ through convective heat transfer (heat transfer coefficient $h = 6 \cdot 10^4 \frac{W}{m^2K}$).

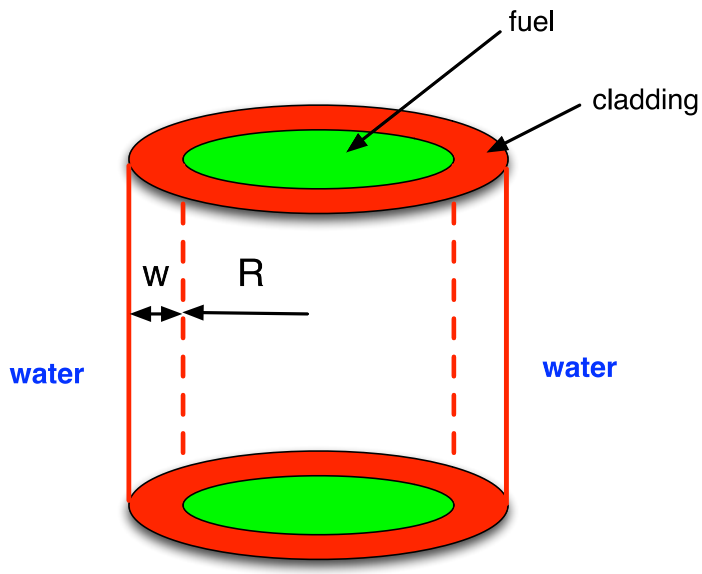
The temperature distribution within the cladding is determined by the ODE:

$$\frac{1}{r}\frac{d}{dr}\left(rk\frac{dT}{dr}\right) = -10^8\frac{e^{-r/R}}{r}$$

in the region of the cladding $R < r < R + w$, with boundary conditions:

$\left.\frac{dT}{dr}\right|_{r=R} = -\frac{6.32 \cdot 10^5}{k}$ and $\left.\frac{dT}{dr}\right|_{r=R+w} = -\frac{h}{k}(T_{r=R+w} - T_w)$.

The thermal conductivity of the metal is $k = 16.75 \frac{W}{mK}$. The dimensions of the rod are: $R = 15mm$ and $w = 3mm$.



1. Compute the temperature distribution within the metal cladding, with $N = 50$.

Task D: The shooting method: Blasius's boundary layer equation

The Blasius' equation, as given in ME2 Fluid Dynamics (boundary layer on a flat plate parallel to a uniform flow) is:

$$\frac{d^3 f(\eta)}{d\eta^3} + \frac{1}{2} f(\eta) \frac{d^2 f(\eta)}{d\eta^2} = 0$$

with boundary conditions: $\begin{cases} f(0) = 0 \\ \frac{df(0)}{d\eta} = 0 \\ \frac{df(\infty)}{d\eta} = 1 \end{cases}$

Solve numerically the ODE, plot the results for $f(\eta)$ and $u(\eta) = \frac{df}{d\eta}$.

Determine the displacement thickness $\int_0^\infty [1 - u(\eta)] d\eta$ and the momentum thickness $\int_0^\infty u(\eta)[1 - u(\eta)] d\eta$ (these two quantities will make physical sense after you are taught boundary layers in FMX2).

Guidance to the solution:

- Rewrite Blasius' equation as three first order ODEs, with new variables $f(\eta)$, $g(\eta)$ and $h(\eta)$:

$$\begin{cases} \frac{df(\eta)}{d\eta} = g(\eta) \\ \frac{d^2 f(\eta)}{d\eta^2} = \frac{dg(\eta)}{d\eta} = h(\eta) \\ \frac{d^3 f(\eta)}{d\eta^3} = \frac{dh(\eta)}{d\eta} = -\frac{1}{2} f(\eta) h(\eta) \end{cases}$$

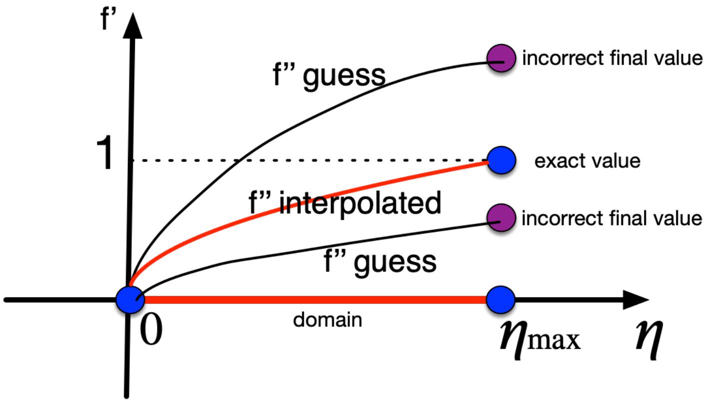
- Define the domain of analysis:
  The physical domain is between $\eta = 0$ and $\eta = \infty$.
  Since we cannot solve for an infinite domain, we need to limit the analysis to a finite domain, i.e., $\eta = [0 \cdots \eta_{max}]$.

- Establish the numerical boundary conditions:

$$f(0) = 0$$
$$\frac{df(0)}{d\eta} = g(0) = 0$$
$$\frac{df(\eta_{max})}{d\eta} = g(\eta_{max}) = 1$$

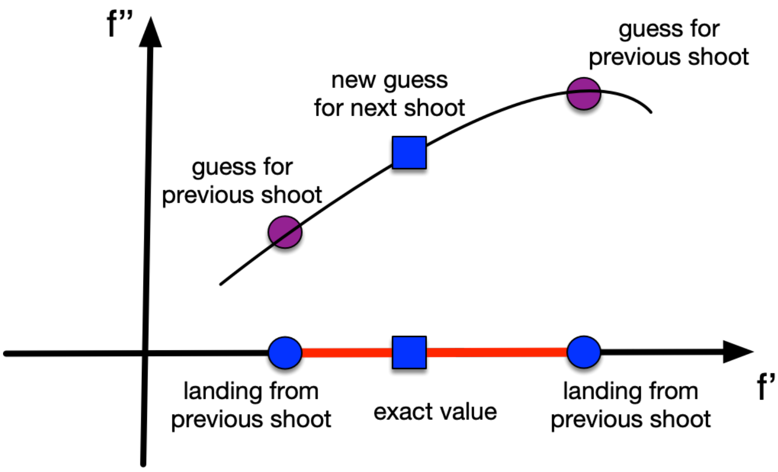
- Establish the numerical method.
  Since the ODE is non-linear it might prove difficult to apply the Finite Difference method. We could instead apply the Shooting method (slide 283).

- When shooting, we guess the value of $\frac{d^2f(0)}{d\eta^2}$, i.e., of $h(0)$, at the beginning of the boundary, and we aim at targeting the given value $\frac{df(\infty)}{d\eta} = 1$ at the end of the boundary.



Note that we know the exact values, given at the beginning of the boundary for $f(0)$ and its first derivative $\frac{df(0)}{d\eta}$, and these remain so throughout the computation.
We shoot by using a direct marching method, i.e., the Forward Euler or the Runge-Kutta.

- After shooting:
  a) We update the guessed value of $\frac{d^2f(0)}{d\eta^2}$. To do so, we interpolate against the values of $\frac{df(\eta_{max})}{d\eta}$ obtained numerically, after shooting:



  b) We assess the error, i.e., how far our target is from the exact value $\frac{df(\infty)}{d\eta} = 1$, and eventually continue shooting, until the desired tolerance is achieved.