MECH3750 PBL Content Summary

Week 7

Content:

- Numerical Solutions for PDEs
  - Explicit Schemes
  - Implicit Schemes

Upcoming assessment:

- Assignment 1 (Due Friday)
- Problem Sheet 7 (due before Week 8 PBL session)

Tutors: Nathan Di Vaira, Alex Muirhead, William Snell, Tristan Samson, Nicholas Maurer, Jakob Ivanhoe, Robert Watt

1 Numerical Solutions for PDEs

Recall the finite difference approximations we covered earlier in the course. Now we are using these to numerically solve PDES.

The first type of PDE we are looking at in lectures are parabolic PDEs, the standard form of which is the diffusion equation (written here in 1D),

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

The diffusion equation can be used to model heat conduction by rewriting the diffusion coefficient in terms of the conducting properties of the material, $D = \frac{k}{\rho C_p}$.

Now, to solve the diffusion equation using finite difference approximations, it is common to use a second order central difference for the spatial derivative,

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2},$$

where j represents the spatial nodes we have discretised our physical system into.

1.1 Forward Difference Scheme (Explicit)

The first way in which we can handle the solution of the time component of the PDE is with a forward difference,

$$\frac{\partial u}{\partial t} \approx \frac{u_j^{m+1} - u_j^m}{(\Delta t)},$$

where m represents the current time step of the discretised solution.

Now, substituting the above finite difference approximations for space and time into the original PDE, we obtain an *explicit* system of equations to be solved. This means that updated values of the solution $(m+1)$ can be written explicitly in terms of past values (m),

$$u_j^{m+1} \approx \sigma u_{j+1}^m + (1-2\sigma)u_j^m + \sigma u_{j-1}^m$$

where $\sigma = D\frac{\Delta t}{(\Delta x)^2}$. These equations can be written concisely in matrix form up to the number of spatial nodes N,

$$\begin{bmatrix} u_0^{m+1} \\ u_1^{m+1} \\ \vdots \\ \vdots \\ \vdots \\ u_N^{m+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \dots & \dots & 0 \\ \sigma & 1-2\sigma & \sigma & \ddots & \ddots & \vdots \\ 0 & \sigma & 1-2\sigma & \sigma & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \sigma & 1-2\sigma & \sigma \\ 0 & \dots & \dots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_0^{m} \\ u_1^{m} \\ \vdots \\ \vdots \\ \vdots \\ u_N^{m} \end{bmatrix}$$

This system of equations represents *Dirichlet* boundary conditions, where the actual value at the first and last spatial nodes ($j = 0$, $j = N$) are specified. We will look at how Neumann boundaries are applied next week.

1.2 Backward Difference Scheme (Implicit)

We can also use a backward finite difference approximation for the time derivative,

$$\frac{\partial u}{\partial t} \approx \frac{u_j^m - u_j^{m-1}}{(\Delta t)},$$

rendering the update equation,

$$-\sigma u_{j-1}^{m+1} + (1+2\sigma)u_j^{m+1} - \sigma u_{j+1}^{m+1} \approx u_j^m.$$

We say that this is *implicit*, because the solution at the current time step m depends on the solution at the next time step $m+1$. We can again write this system of equations in matrix form,

$$\begin{bmatrix} 1 & 0 & 0 & \dots & \dots & 0 \\ -\sigma & 1+2\sigma & -\sigma & \ddots & \ddots & \vdots \\ 0 & -\sigma & 1+2\sigma & -\sigma & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -\sigma & 1+2\sigma & -\sigma \\ 0 & \dots & \dots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_0^{m+1} \\ u_1^{m+1} \\ \vdots \\ \vdots \\ \vdots \\ u_N^{m+1} \end{bmatrix} = \begin{bmatrix} u_0^{m} \\ u_1^{m} \\ \vdots \\ \vdots \\ \vdots \\ u_N^{m} \end{bmatrix}$$

this time inverting the matrix to solve for the updated values $(m+1)$.