Lecture 9: Nonlinear Partial Differential Equations (PDEs)

COMP5930M Scientific Computation

Today

Differential equations

A differential equation is a type of equation where the unknown being sought is a function $u(t)$ of a free variable t, and the equation involves one of the derivatives of u with respect to t.

Example 1: The differential equation

$$\frac{du}{dt} = \cos(t)$$

has general solution $u(t) = \sin(t) + C$, where C is any constant.

To determine C, we need to fix the value of $u(t)$ in at least one point, say $u(0) = u_0$. If we fix the value of $u(t)$ at time $t = 0$ and look for a solution for $t > 0$, we say it is an initial-value problem.

Partial differential equations

A partial differential equation is a type of equation where the unknown being sought is a function $u(x, t)$ of two (or more) free variables x and t, and the equation involves the partial derivatives of u with respect to both x and t.

Example 1: The partial differential equation

$$\frac{\partial u}{\partial t} = \cos(t)\frac{\partial u}{\partial x}$$

includes partial derivatives w.r.t to both x and t.

Typically, we will consider the variable t as time and treat the problem as in initial value problem in time i.e. the value of the function at time t_0, $u(x, t_0) = u_0(x)$, is a known function.

Partial differential equations

A partial differential equation is a type of equation where the unknown being sought is a function $u(x, t)$ of two (or more) free variables x and t, and the equation involves the partial derivatives of u with respect to both x and t.

Example 1: The partial differential equation

$$\frac{\partial u}{\partial t} = \cos(t)\frac{\partial u}{\partial x}$$

includes partial derivatives w.r.t to both x and t.

We will typically treat x as a variable representing a spatial coordinate. For such variables, a natural condition is a boundary condition. If $x \in [a, b]$ then we fix the values of u at the interval end-points: $u(a, t) = \alpha$, $u(b, t) = \beta$, for all $t > 0$.

Example: 1d viscous Burger's Equation

Find $u(x,t)$ satisfying

$$\frac{\partial u}{\partial t} + R\, u\, \frac{\partial u}{\partial x} \;=\; \frac{\partial^2 u}{\partial x^2}$$

on $x \in [X_1, X_2]$, and $t > 0$,
with boundary conditions $u(X_1, t) = u_1(t)$, $u(X_2, t) = u_2(t)$
and initial conditions $u(x, 0) = u_0(x)$.
$R > 0$ is a known constant

- Prototype model for fluid dynamics
- Time dependent
- Nonlinear convection

Behaviour of solutions to Burger's equation

Presence of shock fronts in 1-D (for $R \gg 1$):

`https://www.youtube.com/watch?v=FAY_Nla-LYQ`

Example: Cellular action potential models

Find $u(x, y, t)$ and $v(x, y, t)$ satisfying

$$\frac{\partial u}{\partial t} + g(u, v) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(t)$$
$$\frac{\partial v}{\partial t} = h(u, v)$$

on $(x, y) \in \Omega$, with $u(t) = u_b(x, y, t)$ on the boundary $\partial\Omega$.
Initial conditions $u(x, y, 0)$ and $v(x, y, 0)$ also need to be specified.

- The function $u(x, y, t)$ describes the transmembrane potential across the cell membranes (e.g. nervous system or the heart)
- The function $v(x, y, t)$ describes the opening/closing of ionic channels that regulate the cell membranes
- The functions g, h are nonlinear in u, v.

Behaviour of solutions to action potential models

Nonlinear dynamics exhibiting chaotic behaviour:

`https://www.youtube.com/watch?v=S9lHkUCImcs`

Example: Navier-Stokes Equations

Find $\mathbf{u}(\mathbf{x}, t)$ and $\rho(\mathbf{x}, t)$ satisfying

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p &= \frac{1}{Re} \nabla \cdot \nabla \mathbf{u} \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0 \end{aligned}$$

on $\mathbf{x} \in \Omega$ with *appropriate* boundary and initial conditions.

- Fundamental model for fluid dynamics at the continuum level
- Time dependent
- Diffusive terms
- Nonlinear convection

Behaviour of solutions to Navier-Stokes equations

Complex instabilities can arise even in simple cases:

`https://www.youtube.com/watch?v=Nh1dX7MrukA`

Numerical modelling approach

The method of lines:

1. Approximation in space only
   - Semi-discrete system of ordinary differential equations (ODEs)

$$\mathbf{U}'(t) = \mathbf{F}(\mathbf{U}(t))$$

2. Approximation in time
   - Fully discrete (nonlinear) system, e.g.

$$\mathbf{U}_{k+1} = \mathbf{U}_k + \Delta t \mathbf{F}(\mathbf{U}_{k+1})$$

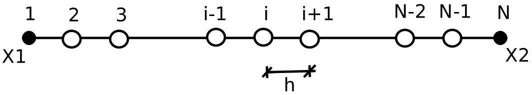
3. Discrete solution algorithm for $\mathbf{U}_{k+1}$ at each step
   - Numerical solution

Approximations in 1d: Finite Difference Methods

Define a grid (mesh) for the spatial domain x

For FDM approximations define a uniform spacing h

$$x_i \;=\; X_1 + (i-1)h, \qquad i = 1, 2, ..., n$$



At point (node) i approximate all spatial terms of the PDE

Notation: $u(x_i, t) \equiv u_i(t), \quad u(x_i, t^n) \equiv u_i^n, \quad \frac{\partial u}{\partial t} \equiv \dot{u}$

Spatial approximation: FDM

Some standard difference approximations

$$\frac{\partial u}{\partial x}(x_i) \approx \frac{u_i - u_{i-1}}{h} \qquad \text{(upwind difference, 1st order)}$$

$$\frac{\partial u}{\partial x}(x_i) \approx \frac{u_{i+1} - u_{i-1}}{2h} \qquad \text{(central difference, 1st order)}$$

$$\frac{\partial^2 u}{\partial x^2}(x_i) \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \qquad \text{(central difference, 2nd order)}$$

We say a FDM is accurate of order p if the error of the approximation is $o(h^p)$. Thus the higher the order, the better.

Spatial approximation: FDM

Some standard difference approximations

$$\frac{\partial u}{\partial x}(x_i) \approx \frac{u_{i+1} - u_i}{h} \qquad \text{(downwind difference, 1st order)}$$

$$\frac{\partial u}{\partial x}(x_i) \approx \frac{u_{i+1} - u_{i-1}}{2h} \qquad \text{(central difference, 1st order)}$$

$$\frac{\partial^2 u}{\partial x^2}(x_i) \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \qquad \text{(central difference, 2nd order)}$$

Many other variants are possible, also of higher orders.

FDM for 1d Burger's Equation

At node i of the grid, the equation is:

$$\frac{\partial u}{\partial t}(x_i) + R\, u(x_i)\, \frac{\partial u}{\partial x}(x_i) \;\; = \;\; \frac{\partial^2 u}{\partial x^2}(x_i)$$

Semi-discrete form:

$$\dot{u}_i + R\, u_i \left(\frac{u_i - u_{i-1}}{h} \right) \;\; = \;\; \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

- Semi-discrete ODE system defined for $i = 2, 3, ..., n-1$
- Upwind/downwind approximation for $\frac{\partial u}{\partial x}$ chosen depending on the direction in which the shock wave travels

Options for nonlinear terms

There are usually alternatives, eg. $u\frac{\partial u}{\partial x} \equiv \frac{1}{2}\frac{\partial(u^2)}{\partial x}$

$$
\begin{aligned}
u\frac{\partial u}{\partial x}(x_i) &\approx u_i\left(\frac{u_i - u_{i-1}}{h}\right) \\
\frac{1}{2}\frac{\partial u^2}{\partial x}(x_i) &\approx \frac{1}{2}\left(\frac{u_i^2 - u_{i-1}^2}{h}\right) \\
&= \frac{1}{2}\left(u_i + u_{i-1}\right)\left(\frac{u_i - u_{i-1}}{h}\right)
\end{aligned}
$$

- Will lead to differences in Jacobian terms and numerics
- Both are consistent and formally of equal accuracy

Approximating in time

The continuous problem is reduced to a semi-discrete system of ODEs for $\mathbf{u}(t)$: $\dot{\mathbf{u}} = f(\mathbf{u})$

Initial-value problem: We start from a given $\mathbf{u}^0$ and find a formula that computes $\mathbf{u}^{k+1}$ given $\mathbf{u}^k$, for $k = 1, 2, 3, \ldots$. Here k is the time index, i.e. the solution at time t_k is $\mathbf{u}^k$.

This procedure is called time-stepping or time-integration. Unlike the spatial variable, we do not need to solve all the unknowns at once but proceed one time-step at a time.

Approximating in time

The continuous problem is reduced to a semi-discrete system of ODEs for $\mathbf{u}(t)$: $\dot{\mathbf{u}} = \mathbf{f}(\mathbf{u})$

- Explicit Euler

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^{k}}{\Delta t} = \mathbf{f}(\mathbf{u}^{k})$$

No actual equation to solve (very efficient), but typically result is unstable unless $\Delta t \ll 1$...

- Implicit Euler

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^{k}}{\Delta t} = \mathbf{f}(\mathbf{u}^{k+1})$$

Fully nonlinear equation for $\mathbf{u}^{k+1}$ to solve, but usually can take larger time steps Δt.

Approximating in time

The continuous problem is reduced to a semi-discrete system of ODEs for $\mathbf{u}(t)$: $\quad \dot{\mathbf{u}} = \mathbf{f}(\mathbf{u})$

- Explicit Euler

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\Delta t} = \mathbf{f}(\mathbf{u}^k)$$

No actual equation to solve (very efficient), but typically result is unstable unless $\Delta t \ll 1$...

- Implicit Euler

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\Delta t} = \mathbf{f}(\mathbf{u}^{k+1})$$

Fully nonlinear equation for $\mathbf{u}^{k+1}$ to solve, but usually can take larger time steps Δt.

1d Burger's Equation

Use Implicit Euler. At node i of the grid:

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} + R\, u_i^{k+1} \frac{u_i^{k+1} - u_{i-1}^{k+1}}{h} = \frac{u_{i+1}^{k+1} - 2u_i^{k+1} + u_{i-1}^{k+1}}{h^2}$$

- Fully discrete nonlinear system for the unknown u_i^{k+1}, $i = 2, ..., n-1$
- The equations are coupled to the neighbouring nodes through the FDM approximation

Nonlinear system structure

Equations are formed at each internal node $i = 2, ..., n-1$

- At node $i-1$

$$F_{i-1}(u_{i-2}^{k+1}, u_{i-1}^{k+1}, u_i^{k+1}) = 0$$

- At node i

$$F_i(u_{i-1}^{k+1}, u_i^{k+1}, u_{i+1}^{k+1}) = 0$$

- At node $i+1$

$$F_{i+1}(u_i^{k+1}, u_{i+1}^{k+1}, u_{i+2}^{k+1}) = 0$$

Each equation depends only on local, neighbouring information

Time-stepping approach

- Initial conditions are specified at $t = t_0$ as $u = U(x, t_0)$
  - Sets the discrete solution $u_i^0 \; = \; U(x_i, t_0)$
- For each time step $k = 0, 1, 2, ...$
  we solve a nonlinear system to find $\mathbf{u}_{k+1}$:
  - $\mathbf{F}(\mathbf{u}^{k+1}, \mathbf{u}^k) \; = \; \mathbf{0}$
  - Advances u_i^k to u_i^{k+1} , $i = 2, 3, ..., n-1$

Summary

- Nonlinear PDEs can exhibit complex spatiotemporal dynamics
- The method of lines for numerically approximating PDEs
  - Separate approximations in space and time
- Approximation in space: Finite difference method
- Approximation in time: Implicit Euler's method
- The final discrete system can be solved with Newton's method at each timestep