Partial Differential Equations

Three types of PDEs: elliptic, parabolic, and hyperbolic

Examples:

Poisson equation $\nabla^2 \phi(\mathbf{r}) = -\rho(\mathbf{r})/\epsilon_0$ elliptic

Diffusion equation $\frac{\partial n(\mathbf{r}, t)}{\partial t} - \boldsymbol{\nabla} \cdot D(\mathbf{r}) \boldsymbol{\nabla} n(\mathbf{r}, t) = S(\mathbf{r}, t)$ parabolic

Time-dependent Schrodinger equation

$$-\frac{\hbar}{i} \frac{\partial \Psi(\mathbf{r}, t)}{\partial t} = \mathcal{H} \Psi(\mathbf{r}, t)$$

Wave equation $\frac{1}{c^2} \frac{\partial^2 u(\mathbf{r}, t)}{\partial t^2} - \nabla^2 u(\mathbf{r}, t) = R(\mathbf{r}, t)$ hyperbolic

Solution of the PDE
Separation of Variables

Basic idea: transform the PDE to a set of equations with fewer variables

Consider a simple wave equation

$$\frac{\partial^2 u(x,t)}{\partial t^2} - c^2 \frac{\partial^2 u(x,t)}{\partial x^2} = 0$$

Assume the solution is of the form $u(x,t) = X(x)\Theta(t)$

The wave equation becomes $\dfrac{\Theta''(t)}{\Theta(t)} = c^2 \dfrac{X''(x)}{X(x)} = -\omega^2$

Thus $X''(x) = -\dfrac{\omega^2}{c^2} X(x) = -k^2 X(x)$

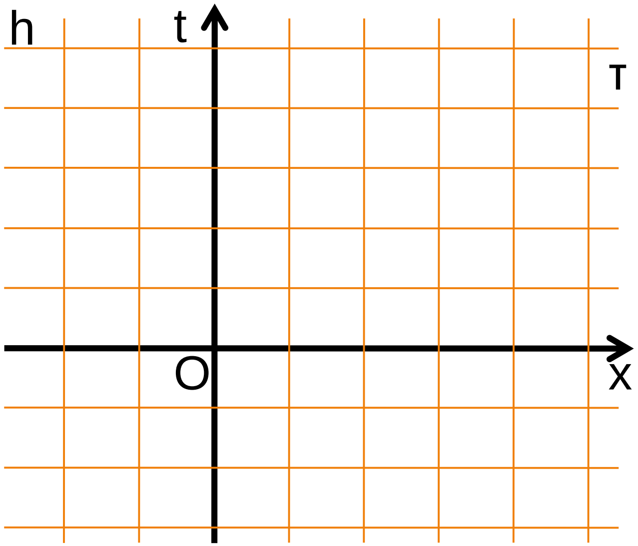
and the solution is $X(x) = A \sin kx + B \cos kx$

Similarly, one can obtain the solution for Θ(t)

Solution of the PDE
Discretization of the PDE

Basic scheme:

- Construct the lattice
- Use finite difference to replace derivatives
- Find solution at lattice points



$$f'(x_i) = \lim_{\Delta x \to 0} \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x}$$

$$f_i' = \frac{f_{i+1} - f_i}{h} + O(h)$$

$$f_i' = \frac{f_{i+1} - f_{i-1}}{2h} + O(h^2)$$

$$f_i'' = \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + O(h^2)$$

Solution of the PDE
The matrix method i.e, $u_0=u_{n+1}=0$

Example: Consider a person sitting at the middle of a bench supported at both ends

Newton's equation: $$YI\frac{d^2u(x)}{dx^2}=f(x)$$

Discretize the equation with evenly spaced intervals, $x_0=0, x_1=h, \ldots, x_{n+1}=L$

and use the three-point formula for the second-order derivative,

$$\Delta_2=\frac{u_{i+1}-2u_i+u_{i-1}}{h^2}=u_i''+\frac{h^2u_i^{(4)}}{12}+O(h^4).$$

Then

$$u_{i+1}-2u_i+u_{i-1}=\frac{h^2f_i}{YI}$$

With the boundary conditions

$$u(0)=u(L)=0$$

i.e, $u_0=u_{n+1}=0$

for $i=0,1,\ldots,n+1$, which is equivalent to

$$\begin{pmatrix} -2 & 1 & \cdots & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 & -2 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}=\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Solution of the PDE
The matrix method

The force distribution on the bench is given by

$$f(x) = \begin{cases} -f_0[e^{-(x-L/2)^2/x_0^2} - e^{-1}] - \rho g & \text{for } |x - L/2| \leq x_0 \\ -\rho g & \text{otherwise,} \end{cases}$$

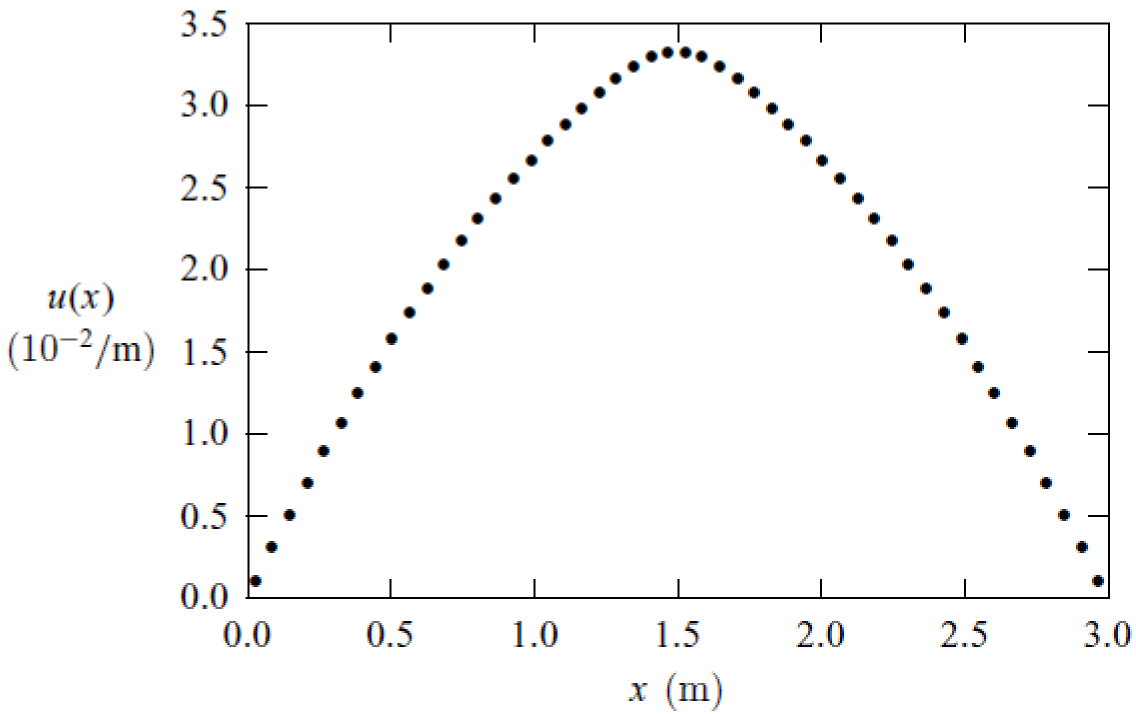


Fig. 7.1 The curvature of the bench, as evaluated with the program given.

Solution of the PDE --- The Relaxation Method

Basic scheme:

- Set up difference equations by discretizing the PDE
- Input an initial guess for the solution
- Perform iteration to obtain converged solution

Example: Consider the bench problem again $YI\,\frac{d^2u(x)}{dx^2} = f(x)$

Use the three-point formula to replace the second derivative

$$u_i'' = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

$$\Longrightarrow \quad u_{i+1} - 2u_i + u_{i-1} = \frac{h^2 f_i}{YI} \quad \Longrightarrow \quad u_i = \frac{1}{2}\left(u_{i+1} + u_{i-1} - \frac{h^2 f_i}{YI}\right) \qquad (1)$$

Iteration scheme: $u_i^{k+1} = (1-p)u_i^k + pu_i$

where p is an adjustable parameter in the range [0, 2],

u_i^k is the solution of the kth iteration at the ith lattice point, and u_i is calculated from Eq. (1)

with the terms on the right-hand side using the result of kth iteration

```
// A program to solve the problem of a person sitting
// on a bench with the relaxation scheme.


  // Evaluate the source in the equation
    for (int i=0; i<=n; ++i) {
      s[i] = rho*g;
      x = h*i-l2;
      if (Math.abs(x) < x0)
        s[i] += f0*(Math.exp(-x*x/x2)-e0);
      s[i] *= h2/y;
    }
    for (int i=1; i<n; ++i) {
      x = Math.PI*h*i/l;
      u[i] = u0*Math.sin(x);
      d[i] = 1;
    }
    d[0] = d[n] = 1;
    relax(u, d, s, p, del, nmax);
```

$$u_{i+1} - 2u_i + u_{i-1} = \frac{h^2 f_i}{YI}$$

$$u_i = \frac{1}{2}\left(u_{i+1} + u_{i-1} - \frac{h^2 f_i}{YI}\right)$$

Define the initial guessed solution that satisfies the boundary condition

```java
// Method to complete one step of relaxation.

  public static void relax(double u[], double d[],
    double s[], double p, double del, int nmax) {
    int n = u.length-1;
    double q = 1-p, fi = 0;
    double du = 2*del;
    int k = 0;

    while ((du>del) && (k<nmax)) {
      du = 0;
      for (int i=1; i<n; ++i) {
        fi = u[i];
        u[i] = p*u[i]
              +q*((d[i+1]+d[i])*u[i+1]
              +(d[i]+d[i-1])*u[i-1]+2*s[i])/(4*d[i]);
        fi = u[i]-fi;
        du += fi*fi;
      }
      du = Math.sqrt(du/n);
      k++;
    }
    if (k==nmax) System.out.println("Convergence not" +
      " found after " + nmax + " iterations");
  }
```

$$u_i^{k+1} = (1-p)u_i^k + pu_i$$

$$u_i = \frac{1}{2}\left(u_{i+1} + u_{i-1} - \frac{h^2 f_i}{YI}\right)$$

(Ds are constant
p and q are reversed)

Solution of the PDE --- The Relaxation Method

For a 2D case, consider the Poisson equation $\nabla^2\phi(\mathbf{r}) = -\rho(\mathbf{r})/\epsilon_0 = -s(\mathbf{r})$

Assuming rectangular boundaries, we have

$$\frac{\phi_{i+1j} + \phi_{i-1j} - 2\phi_{ij}}{h_x^2} + \frac{\phi_{ij+1} + \phi_{ij-1} - 2\phi_{ij}}{h_y^2} = -s_{ij}$$

The above equation can be rearranged to

$$\phi_{ij} = \frac{1}{2(1+\alpha)}\left[\phi_{i+1j} + \phi_{i-1j} + \alpha(\phi_{ij+1} + \phi_{ij-1}) + h_x^2 s_{ij}\right], \qquad (7.76) \qquad \text{with } \alpha = (h_x/h_y)^2$$

A converged solution can be obtained by using the iterative scheme

$$\phi_{ij}^{(k+1)} = (1-p)\phi_{ij}^{(k)} + p\phi_{ij}$$

where p is an adjustable parameter close to 1. Here $\phi_{ij}^{(k)}$ is the result of the kth iteration, and ϕ_{ij} is obtained from Eq. (7.76) with $\phi_{ij}^{(k)}$ used on the right-hand side.

An example of flow cart