

In [1]: 1 autosave

Autosave disabled

Chapter 31-4: Elliptic PDEs Using Finite Difference Method.

The finite difference method (FDM) is an approximate method for solving partial differential equations. It has been used to solve a wide range of problems. These include linear and non-linear, time independent and dependent problems. This method can be applied to problems with different boundary shapes, different kinds of boundary conditions, and for a region containing a number of different materials. Even though the method was known by such workers as Gauss and Boltzmann, it was not widely used to solve engineering problems until the 1940s. The mathematical basis of the method was already known to Richardson in 1910 and many mathematical books were published which discussed the finite difference method. Specific reference concerning the treatment of electric and magnetic field problems is made. The application of FDM is not difficult as it involves only simple arithmetic in the derivation of the discretization equations and in writing the corresponding programs. During 1950–1970 FDM was the most important numerical method used to solve practical problems. With the development of high speed computers having large scale storage capability many numerical solution techniques appeared for solving partial differential equations. However, due to the ease of application of the finite difference method it is still a valuable means of solving these problems.

Comparison between FDM and FEM. Broadly, the finite-difference method starts with a differential statement of the problem and proceeds to replace the derivatives with their discrete analogs. The finite-element method starts with a variational statement of the problem and introduces piecewise definitions of the functions defined by a set of mesh point values.

Although the FDM is used to solve both elliptic differential equations as well as hyperbolic differential equations, the former are considered more difficult to solve than the latter. This is because elliptic equations have boundary conditions at all boundaries of the interval. So any finite difference scheme has to combine multiple boundary conditions at different points.

Poisson's equation is an elliptic partial differential equation of broad utility in theoretical physics. For example, the solution to Poisson's equation is the potential field caused by a given electric charge or mass density distribution; with the potential field known, one can then calculate electrostatic or gravitational (force) field. It is a generalization of Laplace's equation, which is also frequently seen in physics. The equation is named after French mathematician and physicist Siméon Denis Poisson.

Defining Elliptic PDE's

•The general form for a second order linear PDE with two independent variables (x, y) and one dependent variable (u) is

$$A \frac{\partial^2 u}{\partial x^2} + B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + D = 0$$

•Recall the criteria for an equation of this type to be considered elliptic

$$B^2 - 4AC < 0$$

•For example, examine the Laplace equation given by

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

where $A = 1, B = 0, C = 1$ and $D = 0$
then

$$B^2 - 4AC = 0 - 4(1)(1) = -4 < 0$$

thus allowing the classification of this equation as elliptic.

1. Solve the following PDE:

$$u_t = \alpha^2 u_{xx}, \quad 0 < x < 1, 0 < t < \infty,$$

with boundary conditions

$$u(0, t) = u(1, t) = 0, \quad 0 \leq t \leq \infty$$

and initial conditions

$$u(x, 0) = \phi(x), \quad 0 \leq x \leq 1.$$

(The problem and developed answer is taken from the repository of Nicolás Guarín.)

```

In [ ]: 1 function w=poisson(xl,xr,yb,yt,M,N)
2         % run from command window as poisson(0,1,0,2,8,8)
3         f=@(x,y) 4; % define input function data
4         g1=@(y) (y^2-2*y+1); % right
5         g2=@(x) (x^2-4*x+4); % top
6         g3=@(x) x^2; %bottom
7         g4=@(y) y^2; % left
8         m=M+1;n=N+1; mn=m*n;
9         h=(xr-xl)/M;h2=h^2;k=(yt-yb)/N;k2=k^2;
10        x=xl+(0:M)*h; % set mesh values
11        y=yb+(0:N)*k;
12        A=zeros(mn,mn);b=zeros(mn,1);
13        for i=2:m-1 % interior points
14            for j=2:n-1
15                A(i+(j-1)*m,i-1+(j-1)*m)=1/h2;A(i+(j-1)*m,i+1+(j-1)*m)=1/h2;
16                A(i+(j-1)*m,i+(j-1)*m)=-2/h2-2/k2;
17                A(i+(j-1)*m,i+(j-2)*m)=1/k2;A(i+(j-1)*m,i+j*m)=1/k2;
18                b(i+(j-1)*m)=f(x(i),y(j));
19            end
20        end
21        for i=1:m % bottom and top boundary points
22            j=1;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g3(x(i));
23            j=n;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g2(x(i));
24        end
25        for j=2:n-1 % left and right boundary points
26            i=1;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g4(y(j));
27            i=m;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g1(y(j));
28        end
29        v=A\b; % solve for solution in v labeling
30        w=reshape(v(1:mn),m,n); %translate from v to w
31        mesh(x,y,w')
32
33

```

