OLD DOMINION UNIVERSITY

Partial Differential Equations

A. Godunov

1. Basics
2. Elliptic PDEs
3. Finite-difference method for elliptic PDEs

updated 10 April 2022

1

Part 1:

Basics

2

Partial Differential Equations in physics

Very many equations in physics, including fundamental equations are Partial Differential Equations (PDEs).

Even for simple problems analytical solutions to PDEs are either limited to very special cases or represented as infinite series solutions.
In practice the convergence of the series is so painfully slow that many terms are needed for good accuracy, and so the round-off error may become a problem.

In majority of problems in science and engineering solutions of PDEs can only be obtained by numerical methods.

3

3

Classification of PDEs

The general second-order nonhomogeneous partial differential equation in two independent variables

$$Af_{xx} + Bf_{xy} + Cf_{yy} + Df_x + Ef_y + Ff = G$$

where A, B, C and G are arbitrary functions of the variables x and y.

The classification depends on the sign of the discriminant $B^2 - 4AC$ as follows:

$B^2 - 4AC < 0$	Elliptic PDE	$f_{xx} + f_{yy} = 0$	Laplace equation
$B^2 - 4AC = 0$	Parabolic PDE	$f_t = af_{xx}$	Heat equation
$B^2 - 4AC > 0$	Hyperbolic PDE	$f_{tt} = c^2 f_{xx}$	Wave equation

4

4

PDEs and ODEs

Solving PDEs numerically differs from solving ODEs in a number of ways.

1. ODEs have only one independent variable. Standard methods, such that RK4, RKF45 cannot be applied. Because PDEs have several independent variables, we would have to apply, e.g. RKF45 simultaneously and independently to each variable, which would be very complicated.
2. Because there are more equations to solve with PDEs than with ODEs, we need more information than just the two initial conditions or boundary conditions
3. In addition, because each PDE often has its own particular set of boundary conditions, we have to develop a special algorithm for each particular problem.

5

5

Boundary and initial conditions

Solutions to PDEs satisfy given initial and boundary conditions.

Although having an adequate boundary condition is necessary for a unique solution, having too many boundary conditions, may be an overspecification for which no solution exists.

- If the boundary condition is the value of the solution on a surrounding closed surface, we have a *Dirichlet boundary condition*.
- If the boundary condition is the value of the normal derivative on the surrounding surface, we have a *Neumann boundary condition*.
- If the value of both the solution and its derivative are specified on a closed boundary, we have a mixed or a *Cauchy boundary condition*

6

6

Boundary conditions and unique solutions

A physical problem is well posed if its solution exists, is unique, and depends continuously on the boundary and/or initial data.

For an elliptic PDE, the solution domain $D(x, y)$ must be closed, and continuous boundary conditions must be specified along the entire physical boundary B.

For a parabolic PDE, the solution domain $D(x, t)$ must be open in the time (or time-like) direction, initial data must be specified along the time (or time-like) boundary, and continuous boundary conditions must be specified along the physical boundaries of the solution domain.

For a hyperbolic PDE, the solution domain $D(x, t)$ must be open in the time (or time-like) direction, initial data must be specified along the time (or time-like) boundary, and continuous boundary conditions must be specified *along the physical boundaries of the solution domain*.

7

7

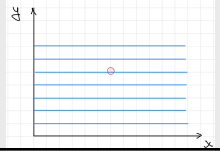
Domain of dependence

It is instructive to consider PDEs using a concept of domain of dependence and domain of influence.

Consider a point P in the solution domain $D(x, y)$

The domain of dependence of point P is defined as the region of the solution domain upon which the solution at point P, $f(x_p, y_p)$, depends. In other words, $f(x_p, y_p)$, depends on everything that has happened in the domain of dependence.

We denote the domain of dependence by blue horizontal lines.



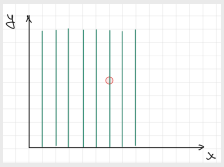
8

8

Range of influence

The range of influence of point P is defined as the region of the solution domain in which the solution $f(x, y)$ is influenced by the solution at point P. In other words, $f(x_p, y_p)$, influences the solution at all points in the range of influence.

We denote the range of influence by green horizontal lines.



9

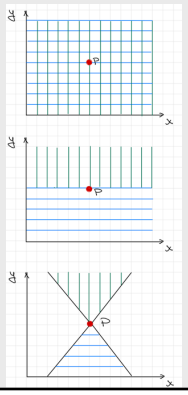
9

Domain of dependence and range of influence

Elliptic PDEs
equilibrium problem

Parabolic PDEs
propagation problem

Hyperbolic PDEs
propagation problem



10

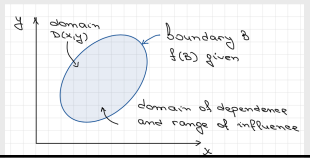
10

Physics I: Equilibrium problems

Equilibrium problems are steady-state problems in closed domains $D(x,y)$ in which the solution $f(x,y)$ is governed by an elliptic PDE subject to boundary conditions specified at each point on the boundary B of the domain.

Equilibrium problems are jury problems in which the entire solution is passed on by a jury requiring satisfaction of all internal requirements and all the boundary conditions simultaneously.

Equilibrium problems are solved numerically by relaxation methods.



11

11

Physics II: Propagation problems

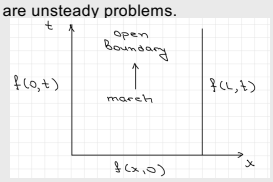
Propagation problems are initial-value problems in open domains (open with respect to one of the independent variables) in which the solution $f(x,t)$ in the domain of interest $D(x,t)$ is marched forward from the initial state, guided and modified by boundary conditions.

Propagation problems are governed by parabolic or hyperbolic PDEs.

Propagation problems in PDEs are analogous to initial-value problems in ODEs

The majority of propagation problems are unsteady problems.

example:
diffusion equation $f_t = af_{xx}$
$f(x,0)$ initial condition
$f(0,t), f(L,t)$ boundary conditions



12

Part 2:
Elliptic PDEs

13

Two most common elliptical PDEs

Laplace's equation in 2D rectangular coordinates

$$\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} = 0$$

The Laplace equation applies to problems in steady state heat conduction, ideal fluid flow, electrostatics, etc.

Poisson's equation in 2D rectangular coordinates

$$\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} = -4\pi\rho(x,y)$$

The Poisson equation is simply the nonhomogeneous Laplace equation. The presence of the nonhomogeneous term $\rho(x,y)$ can greatly complicate the analytic solution of the Poisson equation. However, the presence of this term does not complicate the numerical solution of the Poisson equation.

14

14

About analytic solutions

For the *simple* geometry (rectangular 2D), an analytic solution of Laplace's equation exists in the form of an infinite series (by using the separation of variables).

1. In practice the convergence of the series is so painfully slow that many terms are needed for good accuracy, and so the round-off error may become a problem if we try to evaluate the series numerically.
2. A second problem with the "analytic" solution is that a Fourier series converges only in the *mean square*. *A* phenomenon known as the *Gibbs overshoot* that occurs when a Fourier series with a finite number of terms is used to represent a discontinuous function. To obtain a smooth solution, we may need to sum 40,000 terms, where, in contrast, the numerical solution requires only several hundred steps.

15

15

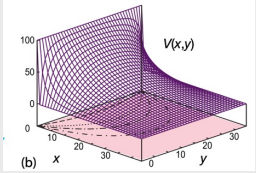
Example

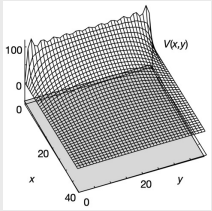
From R. Landau et al 2015

Electric potential in a 2D area

by solving the PDE

21 terms in Fourier analytic solution





Gibbs overshoot leads to the oscillations near $x = 0$, and persist even if a large number of terms is summed.

16

16

Part 3:

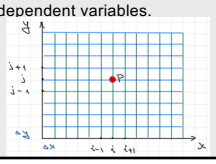
Finite-difference method for elliptical PDEs

17

Finite-difference method

The finite difference method is a numerical procedure which solves a partial differential equation (PDE) by

1. discretizing the continuous physical domain into a discrete finite difference grid,
2. approximating the individual exact partial derivatives in the PDE by algebraic finite difference approximations (FDAs),
3. substituting the FDAs into the PDE to obtain an algebraic finite difference equation (FDE),
4. and solving the resulting algebraic finite difference equations (FDEs) for the dependent variables.



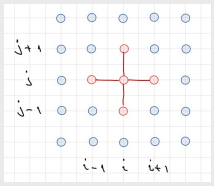
18

18

Taylor series!

Key idea: Use Taylor series for the dependent variable at several neighboring grid points using grid point (i, j) as the base point, and combining these Taylor series to solve for the desired partial derivatives.

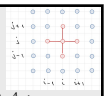
We need Taylor series for points: $f_{i+1,j}$, $f_{i-1,j}$, $f_{i,j+1}$, $f_{i,j-1}$



19

19

Taylor series



$$f_{i+1,j} = f_{i,j} + f_x\Big|_{i,j}\Delta x + \frac{1}{2}f_{xx}\Big|_{i,j}\Delta x^2 + \frac{1}{6}f_{xxx}\Big|_{i,j}\Delta x^3 + \frac{1}{24}f_{xxxx}\Big|_{i,j}\Delta x^4 + \cdots$$

$$f_{i-1,j} = f_{i,j} - f_x\Big|_{i,j}\Delta x + \frac{1}{2}f_{xx}\Big|_{i,j}\Delta x^2 - \frac{1}{6}f_{xxx}\Big|_{i,j}\Delta x^3 + \frac{1}{24}f_{xxxx}\Big|_{i,j}\Delta x^4 + \cdots$$

Then, the second-order central-difference derivative

$$f_{xx}\Big|_{i,j} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta x^2} + O(\Delta x^2)$$

and the same for f_{yy}

$$f_{i,j+1} = f_{i,j} + f_y\Big|_{i,j}\Delta y + \frac{1}{2}f_{yy}\Big|_{i,j}\Delta y^2 + \frac{1}{6}f_{yyy}\Big|_{i,j}\Delta y^3 + \frac{1}{24}f_{yyyy}\Big|_{i,j}\Delta y^4 + \cdots$$

$$f_{i,j-1} = f_{i,j} - f_y\Big|_{i,j}\Delta y + \frac{1}{2}f_{yy}\Big|_{i,j}\Delta y^2 - \frac{1}{6}f_{yyy}\Big|_{i,j}\Delta y^3 + \frac{1}{24}f_{yyyy}\Big|_{i,j}\Delta y^4 + \cdots$$

$$f_{yy}\Big|_{i,j} = \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\Delta y^2} + O(\Delta y^2)$$

20

20

Finite difference method for the Laplace equation

Consider the two-dimensional Laplace equation:

$$f_{xx} + f_{yy} = 0$$

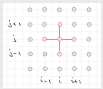
Replacing f_{xx} and f_{yy} by the second-order centered-difference approximations at grid point (i, j)

$$\frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta x^2} + \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\Delta y^2} = 0$$

Equation above can be written as

$$f_{i+1,j} + \beta^2 f_{i,j+1} + f_{i-1,j} + \beta^2 f_{i,j-1} - 2(1+\beta^2)f_{i,j} = 0$$

where $\beta = \Delta x/\Delta y$. Solving for $f_{i,j}$ yields



$$f_{i,j} = \frac{f_{i+1,j} + \beta^2 f_{i,j+1} + f_{i-1,j} + \beta^2 f_{i,j-1}}{2(1+\beta^2)}$$

The solution at every grid point depends on the solutions at the four neighboring grid points

21

Finite difference method for the Poisson equation

Consider the two-dimensional Poisson equation:

$$f_{xx} + f_{yy} = -4\pi\rho(x,y)$$

$$\frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta x^2} + \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\Delta y^2} = -4\pi\rho(x_i, y_j)$$

$$f_{i+1,j} + \beta^2 f_{i,j+1} + f_{i-1,j} + \beta^2 f_{i,j-1} - 2(1+\beta^2) f_{i,j} + \Delta x^2 4\pi\rho(x_i, y_j) = 0$$

or

$$f_{i,j} = \frac{f_{i+1,j} + \beta^2 f_{i,j+1} + f_{i-1,j} + \beta^2 f_{i,j-1} + \Delta x^2 4\pi\rho(x_i, y_j)}{2(1+\beta^2)}$$

All the general features of the numerical solution of the Laplace equation presented apply directly to the numerical solution of the Poisson equation.

22

22

Special case $\Delta x = \Delta y$

In this case we have

$$f_{i+1,j} + f_{i,j+1} + f_{i-1,j} + f_{i,j-1} - 4f_{i,j} = 0$$

Although there is no formal mathematical advantage when $\beta = 1$, values of β greater than unity tend to produce less accurate solutions than values of $\beta \sim 1$. Equation

$$f_{i,j} = \frac{1}{4}(f_{i+1,j} + f_{i,j+1} + f_{i-1,j} + f_{i,j-1})$$

has a very simple physical interpretation. It shows that, for a grid aspect ratio of unity, the solution at every point is the arithmetic average of the solutions at the four neighboring points.

For Poisson equation we have (with $\Delta x = \Delta y = \Delta$)

$$f_{i,j} = \frac{1}{4}(f_{i+1,j} + f_{i,j+1} + f_{i-1,j} + f_{i,j-1}) + \pi\rho(x_i, y_j)\Delta^2$$

23

23

Consistency, Order and Convergence

A finite difference equation is consistent with a partial differential equation if the difference between the FDE and the PDE (i.e., the truncation error) vanishes as the sizes of the grid spacings go to zero independently.

The order of a finite difference approximation of a partial differential equation is the rate at which the error of the finite difference solution approaches zero as the sizes of the grid spacings approach zero.

A finite difference method is convergent if the solution of the finite difference equation approaches the exact solution of the partial differential equation as the sizes of the grid spacings go to zero.

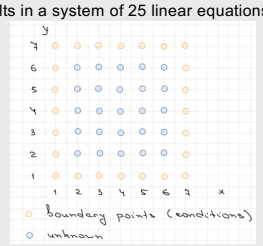
24

24

Example: 7×7 grid

$$f_{i,j} = \frac{1}{4}(f_{i+1,j} + f_{i,j+1} + f_{i-1,j} + f_{i,j-1})$$

The problem results in a system of 25 linear equations (one equation for every blue point).



The system equation can be solved by either a direct method (e.g., Gauss elimination) or an iterative method (e.g., successive-over-relaxation)

25

25

Computational efforts with direct methods

For Gauss elimination the number of multiplicative operations is $N \sim n^3$.

For a two-dimensional problem, the number of grid points increases as the square of the reciprocal of the grid size (assuming that the grid aspect ratio β remains constant as the grid size is reduced).

Thus, the amount of computational effort increases as the sixth power of the reciprocal of the grid size.

Clearly the amount of computational effort increases at an alarming rate.

However, the coefficient matrices arising in the numerical solution of partial differential equations are banded matrices.

When such systems are solved by Gauss elimination, all of the zero coefficients outside of the outer bands remain zero and do not need to be computed.

Therefore, iterative methods, should be employed.

26

26

Using iterative methods

Some popular iterative methods: Jacobi method, Gauss-Seidel method, Successive-over-relaxation method.

The Jacobi method converges slowly comparing to the Gauss-Seidel method.

The Gauss-Seidel method is a special case of the successive-over-relaxation method.

27

27

Successive-over-relaxation method

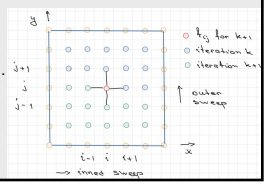
Applying to the Laplace equation the iterative procedure written as

$$f_{i,j}^{k+1} = f_{i,j}^{k} + \omega \Delta f_{i,j}^{k+1}, \qquad \Delta f_{i,j}^{k+1} = \frac{f_{i+1,j}^{k} + \beta^2 f_{i,j+1}^{k} + f_{i-1,j}^{k+1} + \beta^2 f_{i,j-1}^{k+1}}{2(1+\beta^2)}$$

where the superscript k ($k = 0, 1, 2, \ldots$) denotes the iteration number, and ω is the over-relaxation factor (normally the maximum rate of convergence is achieved for ω between 1.0 and 2.0.

Equation above is based on the sweep directions as illustrated.

The order of the sweeps is irrelevant, but once chosen, it should be maintained.



28

Initial approximation (guess) for $f_{i,j}$

We need initial approximation to start the iterative procedure. Several choices are available.

1. Let $f_{i,j} = 0$ at all interior points
2. Approximate $f_{i,j}$ by some weighted average of the boundary values
3. Construct a solution on a coarser grid, then interpolate for starting values on a finer grid. This procedure can be repeated on finer and finer grids.

29

29

Convergence criteria

Iterative methods do not yield the exact solution of the system equation directly. They approach the exact solution asymptotically as the number of iterations increase.

The iterative process is usually terminated when some form of convergence criterion has been achieved. Various convergence criteria are possible. For example:

$|\Delta f_{i,j}^{k+1}| < \varepsilon$ for all i, j, $\qquad |\Delta f_{i,j}^{k+1} / f_{i,j}^{k}| < \varepsilon$, for all i, j

$$\sum_{i,j}^{n} |\Delta f_{i,j}^{k+1}| < \varepsilon, \qquad \sum_{i,j}^{n} |\Delta f_{i,j}^{k+1} / f_{i,j}^{k}| < \varepsilon$$

where ε is the convergence tolerance.

Caution in the use of relative criteria is necessary if any of the $f_{i,j}$ are close to zero in magnitude.

30

30

Example: MatLab code

```
%{
  Solving the Poisson equation with Dirichlet BCs
  INPUT:
  f(i,j)    boundary conditions + initial guess for internal points
  F(i,j)    nonhomogeneous term
  dx, dy    grid increments
  Nx        number of grid points in x direction
  Ny        number of grid points in y direction
  iter      maximum number of iterations
  tol       convergence tolerance
  omega     over-relaxation factor
  OUTPUT
  f2(x,y)   the solution
AG: April 2022
%}
function[f2] = pde01(f,F,dx,dy,Nx,Ny,iter,tol,omega)
% preparation
f2 = zeros(Nx,Ny);
beta2 = (dx/dy)^2;
d = 2.0*(1.0+beta2);
```

31

31

Example: MatLab code (cont.)

```
% iterations
for it=1:iter
  dfmax= 0.0;
  for j=2:Ny-1
    for i=2:Nx-1
      df=(f(i+1,j)+beta2*f(i,j+1)+f(i-1,j)+beta2*f(i,j-1)-d*f(i,j))/d;
        if (abs(df) >= dfmax)
          dfmax = df;
        end
      f(i,j)=f(i,j)+omega*df;
    end
  end
  if(dfmax <= tol)
    fprintf(' \n The solution has converged, it = %3i',it)
    break
  end
end
% Prepare OUTPUT
for j=1:Ny
  for i=1:Nx
    f2(i,j)= f(i,j);
  end
end
if it==iter
  fprintf('  \n The solution failed to converge, iter = %3i',iter)
end
end
```

32

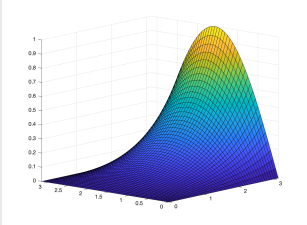
32

Example: Laplace equation

$$f_{xx} + f_{yy} = 0$$

Boundary conditions:

$$f(x,0) = 0, \quad f(x,\pi) = \sin(x), \quad f(0,y) = 0, \quad f(\pi,y) = 0$$



Time – less than 1 second for 51*51 grid

33

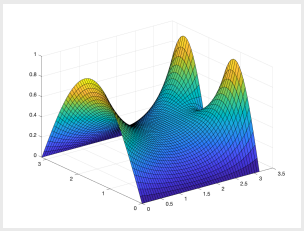
33

Example: Laplace equation

$$f_{xx} + f_{yy} = 0$$

Boundary conditions:

$$f(x,0) = \sin(x), \quad f(x,\pi) = \sin(2x), \quad f(0,y) = 0, \quad f(\pi,y) = 0$$



Time – less than 1 second for 51*51 grid

34

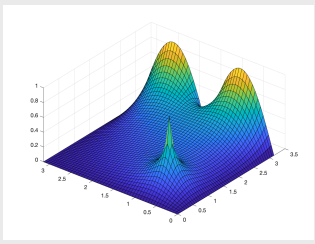
34

Example: Poisson equation (with a point source)

$$f_{xx} + f_{yy} = F(x,y)$$

Boundary conditions:

$$f(x,0) = 0, \quad f(x,\pi) = \sin(2x), \quad f(0,y) = 0, \quad f(\pi,y) = 0$$

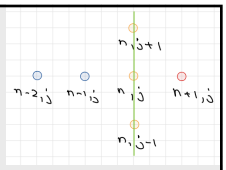


Time – less than 1 second for 51*51 grid

35

35

Derivative boundary conditions



Assume that the derivative boundary condition is specified for the right side.

For point $f_{n,j}$ we have

$$f_{n+1,j} + \beta^2 f_{n,j+1} + f_{n-1,j} + \beta^2 f_{n,j-1} - 2(1+\beta^2) f_{n,j} = 0$$

but the grid point $(n+1, j)$ is is outside of the solution domain, *so* $f_{n+1,j}$ is not defined. However, using the central difference for f_x we have

$$f_x\Big|_{n,j} = \frac{f_{n+1,j} - f_{n-1,j}}{2\Delta x}$$

then $f_{n+1,j} = f_{n-1,j} + 2\Delta x f_x|_{n,j}$

and the FDM with the derivative boundary condition on the right reads

$$\beta^2 f_{n,j+1} + 2f_{n-1,j} + \beta^2 f_{n,j-1} - 2(1+\beta^2) f_{n,j} = -2\Delta x f_x\Big|_{n,j}$$

36

36

Various combinations of boundary conditions

The extension from Dirichlet to Neumann (derivative) boundary conditions is straightforward.

However, we have many possible combinations of types of boundary conditions even for 2D case (4 sides), e.g. DDDN, DDNN, etc. gives 16 combinations of boundary conditions.

How do we handle so many combinations?

37

37

Higher-order methods

The five-point method developed above is a second-order method.

Higher-order method are derived similar to higher-order methods for ODEs boundary value problem.

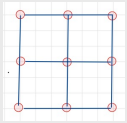
The explicit fourth-order centered difference FDA uses five grid points along both x and y. Such nine-point method when applied at points adjacent to a boundary, requires a point outside of the boundary, and thus cannot be used. An unsymmetrical fourth-order FDA or the second-order five-point method must be used at these points, thus reducing the accuracy somewhat.

38

38

Compact fourth-order method

The compact fourth-order method is a nine-point method but the points are adjacent to (i, j) point as



$$f_{i+1,j+1} + f_{i+1,j-1} + f_{i-1,j+1} + f_{i-1,j-1} + \frac{2(5-\beta^2)}{\beta^2+1}(f_{i+1,j} + f_{i-1,j}) + \frac{2(5\beta^2-1)}{\beta^2+1}(f_{i,j+1} + f_{i,j-1}) - 20f_{i,j} = 0$$

For unity grid aspect ratio (i.e. $\beta = 1$)

$$f_{i+1,j+1} + f_{i+1,j-1} + f_{i-1,j+1} + f_{i-1,j-1} + 4\left(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}\right) - 20f_{i,j} = 0$$

This approach works well for Dirichlet boundary conditions. However, it is difficult to obtain fourth-order accuracy at the boundaries for Neumann (i.e., derivative) boundary conditions.

39

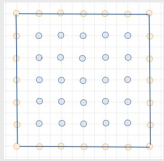
39

Non-rectangular domains

The method above expressed in rectangular coordinates

Several significant simplifications result in this case:

1. Grid points of the finite difference grid fall on the boundary of the physical space, so boundary conditions can be specified.
2. The computational grid is uniform and orthogonal, so accurate finite difference approximations of exact partial derivatives can be derived.
3. The grid spacing adjacent to the boundaries is uniform and orthogonal.

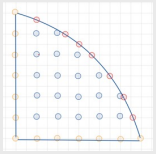


40

40

Non-rectangular domains

When the physical space is not rectangular, however, problems arise.



Consider the quarter-round physical space above, which is discretized by a rectangular finite difference grid. Except for rare points, grid points do not fall on the curved boundary of the physical space, thus making it impossible to specify boundary conditions. The finite difference grid is not uniform at interior points adjacent to the curved boundary. Obviously, some new finite difference approach is required.

41

41

Non-rectangular domains

There are several approaches available for modeling nonrectangular physical spaces:

1. Approximate physical boundary (not recommended)
2. Other coordinate systems (recommended when possible))
3. Nonuniform finite difference approximations (possible)
4. Transformed spaces (recommended)

42

42

Three-dimensional problems

Three-dimensional problems can be solved by the same methods that are used to solve two-dimensional problems by including the finite difference approximations of the exact partial derivatives in the third direction.

The major complication is that the size of the system of FDEs increases dramatically.

43

43

SUMMARY for elliptical PDEs

- Elliptic PDEs describe equilibrium problems.
- The domain of dependence and range of influence of every point is the entire closed solution domain.
- Such problems are solved numerically by relaxation methods.
- Finite difference methods, yield a system of finite difference equations, which must be solved by relaxation methods. The successive-over-relaxation (SOR) method is generally the method of choice.
- Numerous libraries and software packages are available for integrating the Laplace and Poisson equations.
- Due to the wide variety of elliptic PDEs, many elliptic PDE solvers have been developed.

44

44