Math 228B - Feb/10/2009
Class Notes for week 6, lecture 11

Non-rectangular Domains

We want to solve the equation

$$u_t = b\triangle u$$

in a non-rectangular domain with Dirichlet BCs.
There are few options:

1. use a Cartesian grid – modify discrete operators near boundaries
2. use body fitted coordinates (see Polar Coordinates example below)
3. use unstructured mesh (note that this is better suited for finite volume and finite element methods)

Cartesian grid

For example, Ω is a square with a circle removed, then there are two boundaries, namely the outer square and the inner circle. Grid points that are not on the boundary are divided into two types: (1) regular points - all neighbors are inside the domain (or on the boundary), and (2) irregular points - at least one neighbor point is located outside the domain.

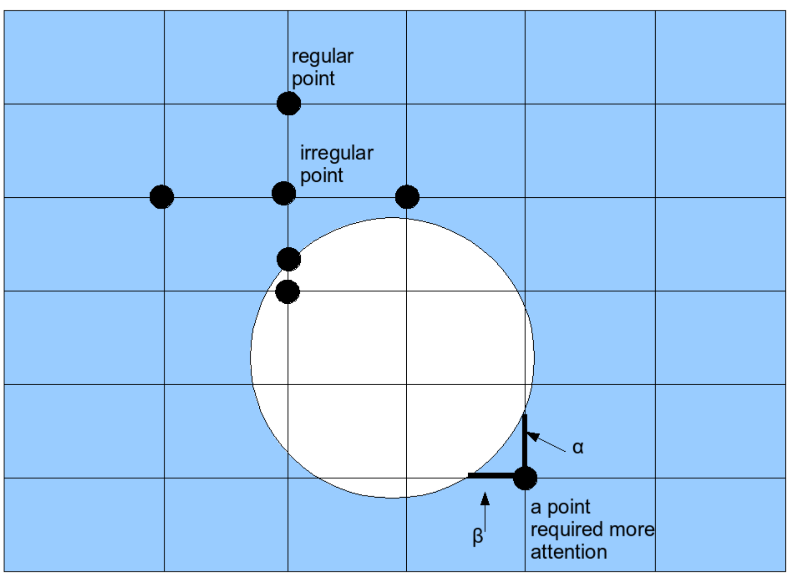
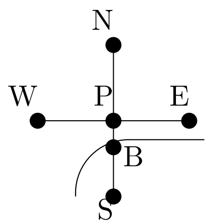


Figure 1: Ω is square with circle removed.



Suppose that point B is on the boundary and point S is outside the domain. How to discretize at the irregular point P?

Let L denote the Laplacian operator $L = L_x + L_y$, where L_x is the standard 3-point, one-dimensional Laplacian (W,P,E are inside the domain). What to do for L_y? Could use N, P, B to extrapolate a value to S, and apply the standard stencil. However, you can run into trouble applying this method. Some points on the exterior of the domain would require two values (one for each direction). We need to modify the discrete operator at P. Let $y_P - y_B = \alpha h$ for $\alpha < 1$, so that e α is the distance between point P and point B. The stencil of L_y is

$$u_{yy} \approx \frac{2\alpha u_N - 2(\alpha+1)u_p + 2u_B}{\alpha(1+\alpha)h^2}.$$

One can show that applying this modified operator is equivalent to extrapolation followed by the application of the standard operator.

Implementation

How to invert the operator? and, How to handle storage?
Need to store the coefficients of the discrete Laplacian at each point.

What about points outside the domain? Since these points are not coupled to the points in the physical domain, we can do whatever is convenient. For example, replace the discrete Laplacian with the identity operator at these points.

How to solve?
1. Use iterative methods
   –SOR, just need to change the stencil accordingly.
   –MG, requires a nontrivial change to existing code.

2. For time-dependent equations (e.g., the heat equation)
   –ADI, need to be careful with forming the tridiagonal matrices in the solver.

What to program?
decide if a point is in or out, and then check to see if the neighbors are in or out. If the point is irregular, then you need to calculate the distances (α and β) to the boundary to compute the modified stencil.

Stability

Want to avoid explicit schemes, even if $h^2/(4b)$ is not small. For stability of forward Euler, we need

$$\triangle t \leq \frac{h^2}{2b(\frac{1}{\alpha} + \frac{1}{\beta})}.$$

This can be very restrictive if α and β are small.

Accuracy

At the irregular points, the LTE is of $O(h)$. In one dimension (see 228A HW), on $O(h)$ LTE adjacent to the boundary contributes an $O(h^3)$ error to the solution. Therefore we will still get a second order accurate solution. This holds in higher dimensions, but it's much more difficult to show.

Boundary fitted meshes

Idea: map the physical domain to rectangular domain and solve on the transformed domain.

An easy example: Polar Coordinates. Suppose the physical domain is an annulus. A natural way to discretize the domain is to use polar coordinates, (r, θ), and use regularly spaced points in r and θ. To generalize this idea, we take a slightly different view. How is this a mapping of the annulus to a rectangle? The map

$$x = r\cos\theta$$
$$y = r\sin\theta$$

transforms the rectangle $[r_0, r_1] \times [0, 2\pi]$ to the annulus $r_0^2 \leq x^2 + y^2 \leq r_1^2$. The mesh lines in the rectangular domain map to the standard polar grid in the original domain. When changing coordinates, we also need to transform the PDE. In polar coordinates the Laplacian is

$$\triangle u = \frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial u}{\partial r}) + \frac{1}{r}\frac{\partial^2 u}{\partial \theta^2}.$$

After changing coordinates, we now have a variable coefficient operator.

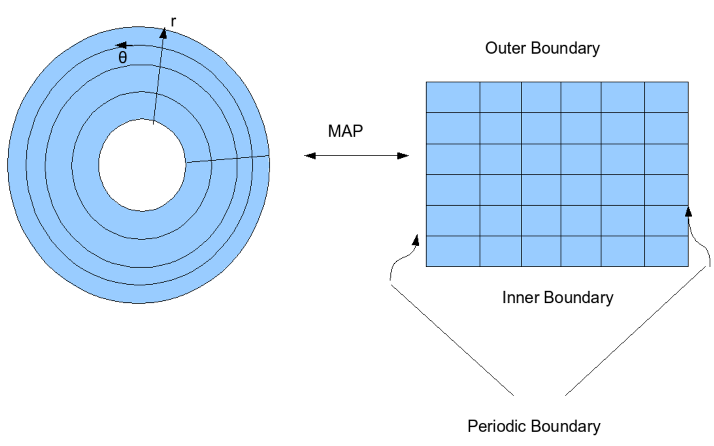


Figure 2: Mapping of Polar and standard coordinates. Given boundary conditions on top and bottom, periodic boundary conditions on left and right.