Numerical methods
Bachelor in Physics

Practicum 9:
Partial differential equations III: Basics and simple explicit methods.

Contact: jeroen.mulkers@uantwerpen.be U.305

After solving parabolic and hyperbolic PDEs, we will finally study elleptical PDEs. These PDEs are boundary value problems. The solution of these PDEs are static fields, e.g. an electric field which is the solution of the Laplace equation.

1 Boundary value problems

Elliptical PDEs, such as the Poisson and the Laplace equation, are not initial value problems, but are boundary value problems. E.g. the determination of a potential in a rectangle for which the potential at the boundary is given by

$$\Phi(x=0,y)=\Phi_1;\quad \Phi(x=L_x,y)=\Phi_2;$$

$$\Phi(x,y=0)=\Phi_3;\quad \Phi(x,y=L_y)=\Phi_4;$$

The potential in a single point is influenced by the the potential at the borders. The algorithms designed to solve boundary value problems are often called *jury* methods. We will discretize space as follows: $x_i=(i-1)h_x, y_j=(j-1)h_y$ with h_x and h_y the step size in x and y-direction.

2 Relaxation methods

The archetype of the elliptical PDE is the Laplace equation. In two dimensions, this equation can be written down as

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

with electrostatic potential $\Phi(x, y)$. We will solve this PDE in a rectangular domain with borders $x = 0, x = L_x, y = 0$ and $y = L_y$. We will use the following boundary conditions

$$\Phi(x = 0, y) = \Phi(x = L_x, y) = \Phi(x, y = 0) = 0$$
$$\Phi(x, y = L_y) = \Phi_0$$

where Φ_0 is constant.

Using the method of separation of variables, we can find the solution analytically:

$$\Phi(x, y) = \Phi_0 \sum_{n=1,3,5,\ldots}^{\infty} \frac{4}{\pi n} \sin\left(\frac{n\pi x}{L_x}\right) \frac{\sinh(n\pi y/L_x)}{\sinh(n\pi L_y/L_x)}$$

In order to solve numerically the Laplace equation, it is useful to have a look at the diffusion equation

$$\frac{\partial T(x, y, t)}{\partial t} = \kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)$$

with diffusion coefficient κ and temperature $T(x, y, t)$. We know that, for a given initial temperature profile and boundary conditions, the solution will relax to a stationary state $T_s(x, y) = \lim_{t\to\infty} T(x, y, t)$. For the stationary state, it holds true that

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

which corresponds with the Laplace equation.

We can conclude that solution of the Laplace equation is equivalent with the solution of the diffusion equation in the limit $t \to \infty$. Algorithms, which are based upon this idea, are called *relaxation methods*. The diffusion equation can be solved numerically using the following FTCS:

$$\Phi_{i,j}^{n+1} = \Phi_{i,j}^n + \frac{\kappa\tau}{h_x^2}\{\Phi_{i+1,j}^n + \Phi_{i-1,j}^n - 2\Phi_{i,j}^n\} + \frac{\kappa\tau}{h_y^2}\{\Phi_{i,j+1}^n + \Phi_{i,j-1}^n - 2\Phi_{i,j}^n\}$$

with $\Phi_{i,j}^n \equiv \Phi(x_i, y_j, t_n), x_i \equiv (i-1)h_x, y_j \equiv (j-1)h_y$ and $t_n \equiv (n-1)\tau$. Of course, the potential does not really depend on the time. We only introduce the time dependency in order to converge the solution of the diffusion equation towards the solution of the Laplace equation. One can prove that the FTCS scheme is stable if

$$\frac{\kappa\tau}{h_x^2} + \frac{\kappa\tau}{h_y^2} \leq \frac{1}{2}$$

For $h_x = h_y = h$ this reduces to $\kappa\tau/h^2 \leq 1/4$.

Since we are only interested in the stationary state ($n \to \infty$), we want to use a time step as large as possible. For $\kappa\tau/h^2 = 1/4$ we get the following scheme

$$\Phi_{i,j}^{n+1} = \frac{1}{4}\{\Phi_{i+1,j}^{n} + \Phi_{i-1,j}^{n} + \Phi_{i,j+1}^{n} + \Phi_{i,j-1}^{n}\}$$

These method is called the *Jacobi methode*. Note that this method replaces the value in a certain point with the average of the values of its four neighbours.

2.1 The Gauss-Seidel method and over-relaxation

Analogously to the relaxation methods, which we have used to solve systems of equations, we can use the updated values Φ^{n+1} once calculated. E.g.:

$$\Phi_{i,j}^{n+1} = \frac{1}{4}\{\Phi_{i+1,j}^{n} + \Phi_{i-1,j}^{n+1} + \Phi_{i,j+1}^{n} + \Phi_{i,j-1}^{n+1}\}$$

This method is called the *Gauss-Seidel method*. Note that it is no longer necessary to store both Φ^n and Φ^{n+1}.

The previous methods converge relatively slow. We can accelerate the relaxation with the following scheme

$$\Phi_{i,j}^{n+1} = (1-\omega)\Phi_{i,j}^{n} + \frac{\omega}{4}\{\Phi_{i+1,j}^{n} + \Phi_{i-1,j}^{n+1} + \Phi_{i,j+1}^{n} + \Phi_{i,j-1}^{n+1}\}$$

with over-relaxation parameter ω. This method is called *over-relaxation*.

Now, it comes down to choose an appropriate value for ω. If we choose $\omega = 1$, we get again the Gauss-Seidel method. The rate of convergence will be slowed down for $\omega < 1$ due to under-relaxation. De method is not stable for $\omega > 2$ due to over-relaxation. There is an optimal value between 1 and 2 which gives the highest rate of convergence. This optimal value is known for some geometries. E.g. for a $N_x \times N_y$ rectangular grid, the optimal value is given by

$$\omega_{\text{opt}} = \frac{2}{1 + \sqrt{1 - r^2}}$$

with

$$r = \frac{1}{2}\left(\cos\frac{\pi}{N_x} + \cos\frac{\pi}{N_y}\right)$$

If $N_x = N_y = N$, then this reduces to

$$\omega_{\text{opt}} = \frac{2}{1 + \sin(\pi/N)}$$

If $N = 100$, then $\omega_{\text{opt}} \approx 1.939$. The choice of ω for other geometries has to be determined with trail and error. Often, the value of ω is automatically updated on the fly.

2.2 The Poisson equation

The method can be easily extended in order to solve the Poisson equation. The Poisson equation in two dimensions is given by

$$\frac{\partial^2\Phi(x,y)}{\partial x^2} + \frac{\partial^2\Phi(x,y)}{\partial y^2} = -\frac{1}{4\pi\epsilon_0}\rho(x,y)$$

with charge density $\rho(x,y)$. The Jacobi relaxation scheme for the Poisson equation is given by

$$\Phi_{i,j}^{n+1} = \frac{1}{4}\left\{\Phi_{i+1,j}^{n} + \Phi_{i-1,j}^{n} + \Phi_{i,j+1}^{n} + \Phi_{i,j-1}^{n} + \frac{1}{4\pi\epsilon_0}h^2\rho_{i,j}\right\}$$

with $h = h_x = h_y$.

2.3 The program

The program relax.m contains an implementation of the Jacobi, the Gauss-Seidel and the over-relaxation method. This program solves the Laplace equation based on the following algorithm

- Initialize the parameters $h, L, \ldots$
- Choose a value for ω (only for over-relaxation methods)
- Choose an initial potential profile.
- Iterate until the desired fractional change per iteration is obtained
  - Update the potential Φ
  - Check if $[\sum_{i,j} |(\Phi_{i,j}^{n+1} - \Phi_{i,j}^{n})/\Phi_{i,j}^{n+1}|]/N^2$ is small enough.
- Plot $\Phi(x,y)$ using contour and mesh.
- Plot $[\sum_{i,j} |(\Phi_{i,j}^{n+1} - \Phi_{i,j}^{n})/\Phi_{i,j}^{n+1}|]/N^2$ for each iteration.

2.4 Tasks

1. Study relax.m and execute the three different methods. The program starts with an appropriate initial condition (first term of the analytical solution). In order to check the importance of an appropriate initial condition, choose $\Phi = 0$ as initial condition of the inner points. Use again the three different methods.

2. Study relax_hall.m. This program solves also the Laplace equation, but the geometry is in this case a Hall-cross with boundary conditions as given in Fig. 1. In order to implement a non-trivial grid, one often uses a geometry-grid. This geometry-grid is a matrix with zeros and ones which indicates if grid points are in or out the geometry. This approach makes it easy to change the geometry. Solve the Laplace equation for the potential on a disc, bounded by two concentric circles with radii $r = 0.4$ and $r = 0.1$. Use the following boundary conditions: $\Phi = 0$ if $r = 0.1$ and $\Phi = 1$ if $r = 0.4$.

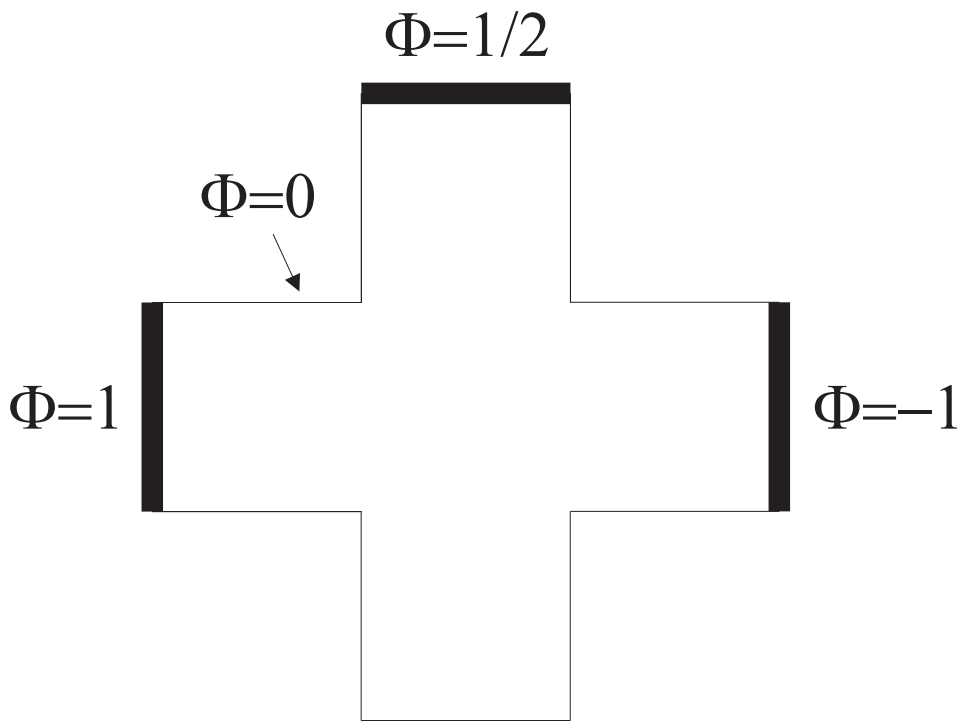

Figure 1: Geometry and boundary conditions for the hall cross