Elliptic Boundary Value Problems

M.Sc. in Mathematical Modelling & Scientific Computing,
Practical Numerical Analysis

Michaelmas Term 2018, Lecture 6

Elliptic Boundary Value Problems

Let Ω be a bounded open set in $\mathbb{R}^n$. We consider the linear second-order partial differential equation

$$-\sum_{i,j=1}^{n} \frac{\partial}{\partial x_j}\left(a_{i,j}(x)\frac{\partial u}{\partial x_i}\right) + \sum_{i=1}^{n} b_i(x)\frac{\partial u}{\partial x_i} + c(x)u \quad = \quad f(x), \quad (1)$$

for $x \in \Omega$, where the coefficients $a_{i,j}$, b_i, c and f satisfy the following conditions:

$$\begin{aligned} a_{i,j} &\in C^1(\bar{\Omega}), & i,j &= 1,\ldots,n; \\ b_i &\in C(\bar{\Omega}), & i &= 1,\ldots,n; \\ c &\in C(\bar{\Omega}), & f &\in C(\bar{\Omega}), \end{aligned}$$

and

$$\sum_{i,j=1}^{n} a_{i,j}(x)\xi_i\xi_j \quad \geq \quad \tilde{c}\sum_{i=1}^{n}\xi_i^2, \ \forall \xi = (\xi_1,\ldots,\xi_n) \in \mathbb{R}^n, \ x \in \bar{\Omega} \quad (2)$$

where $\tilde{c}$ is a positive constant independent of x and ξ.

Elliptic Boundary Value Problems

If all these conditions are satisfied Equation (2) is known as the uniform ellipticity condition and Equation (1) as an elliptic partial differential equation.

In order to solve elliptic PDEs we must also supply boundary conditions on the boundary of Ω (usually denoted by $\partial\Omega$). Boundary conditions typically take one of the following forms:

- $u = g$ on $\partial\Omega$ (Dirichlet boundary condition);
- $\nabla u \cdot \mathbf{n} = g$ on $\partial\Omega$ (Neumann boundary condition);
- $\nabla u \cdot \mathbf{n} + \sigma u = g$ on $\partial\Omega$, where $\sigma(x) \geq 0$ on $\partial\Omega$ (Robin boundary condition).

Here, g is a given function on $\partial\Omega$ and $\mathbf{n}$ is the outward unit normal to $\partial\Omega$.

Elliptic Boundary Value Problems

A final form of boundary condition is a combination of the Neumann and Robin boundary conditions and is given by

$$\sum_{i,j=1}^{n} a_{i,j} \frac{\partial u}{\partial x_i} \cos \alpha_j + \sigma(x) u \quad = \quad g$$

on $\partial\Omega$, where α_j is the angle between the unit outward normal vector $\mathbf{n}$ to $\partial\Omega$ and the x_j axis. Such a boundary condition is known as an oblique derivative boundary condition.

Examples of Elliptic PDEs

Common examples of elliptic PDEs are:

- Laplace's equation: $-\nabla^2 u = 0$;
- Poisson's equation: $-\nabla^2 u = f$;
- Advection diffusion equation: $-\nabla \cdot (A\nabla u) + \mathbf{b} \cdot \nabla u = f$;
- Steady state reaction convection diffusion equation: $-\nabla \cdot (A\nabla u) + \mathbf{b} \cdot \nabla u + cu = f$.

Example for this Lecture

We will consider Poisson's equation with homogeneous Dirichlet boundary conditions in the domain $\Omega = (0,1)^2 \subset \mathbb{R}^2$. Thus we solve

$$\begin{aligned} -\nabla^2 u &= f(x,y) && \text{in } \Omega \\ u &= 0 && \text{on } \partial\Omega. \end{aligned}$$

(Note also that dealing with advection and reaction terms as well as more general boundary conditions will be discussed in NSDEI for parabolic PDEs. We will also come back to this later in term.)

Analytical Solution

One way to solve such problems analytically is via separation of variables. This leads to a solution which is an infinite series (sum of eigenfunctions of the Laplacian). This still raises a question of how to evaluate the solution. At what point can we truncate the infinite series?

Numerical Solution via Finite Differences

Let $u(x) \in C^4(\mathbb{R})$ then, using Taylor series expansions, we may write

$$\begin{aligned} u(x+h) &= u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u'''(x) + \frac{h^4}{24}u''''(\xi_+) \\ u(x-h) &= u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u'''(x) + \frac{h^4}{24}u''''(\xi_-) \end{aligned}$$

for some $\xi_+ \in (x, x+h)$ and $\xi_i \in (x-h, x)$.

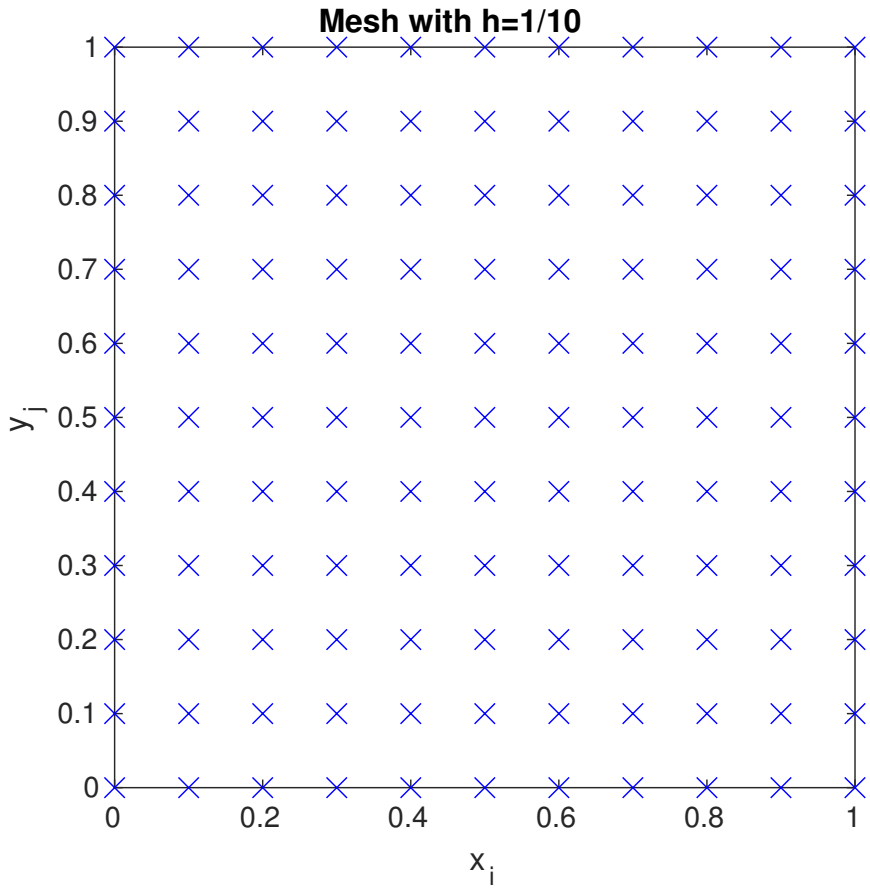
Thus we can combine these to see

$$\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} = u''(x) + \frac{h^2}{12}u''''(\xi) \qquad (3)$$

for some $\xi \in (x-h, x+h)$.

Numerical Solution via Finite Differences

This motivates our numerical scheme. Consider the mesh on $[0,1]^2$ defined by nodes $x_i = ih$, $y_j = jh$, $i,j = 0,\ldots,N$ and $h = 1/N$. We use the abbreviation $u_{i,j} = u(x_i, y_j)$ and define our numerical approximation by $U_{i,j} \approx u_{i,j}$.



Numerical Solution via Finite Differences

On the grid we may then use Equation (3) to see

$$\frac{\partial^2 u}{\partial x^2}(x_i, y) = \frac{u(x_{i+1}, y) - 2u(x_i, y) + u(x_{i-1}, y)}{h^2} - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}u(\xi_i, y)$$

for some $\xi_i \in (x_{i-1}, x_{i+1})$. Thus we can write the numerical scheme as

$$-\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h^2} - \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{h^2} = f(x_i, y_j)$$

for $i, j = 1, \ldots, N-1$. The boundary conditions are applied as

$$U_{i,0} = U_{i,N} = U_{0,j} = U_{N,j} = 0$$

for $i, j = 0, \ldots, N$.

Numerical Solution via Finite Differences

We may rewrite this system of equations as

$$4U_{i,j} - U_{i+1,j} - U_{i-1,j} - U_{i,j+1} - U_{i,j-1} \quad = \quad h^2 f_{i,j}$$

for $i, j = 1, \ldots, N-1$ and where $f_{i,j} = f(x_i, y_j)$. The values $U_{i,0}$, $U_{i,N}$, $U_{0,j}$ and $U_{N,j}$ are all zero.

You have seen such a system before!! (See NLA lecture 6)

Linear System

The system of equations can be thought of as a linear system of size $(N-1)^2 \times (N-1)^2$ (assuming we do not count the solution on the boundary as unknown we can just eliminate these terms from the equations).

Let $\mathbf{U} = (U_{1,1}, U_{1,2}, \ldots U_{1,N-1}, U_{2,1} \ldots U_{N-1,N-1})^T$. If we order $\mathbf{f}$ in the same way we may write a linear system

$$A\mathbf{U} \quad = \quad \mathbf{f}\,,$$

where the form of the matrix $A \in \mathbb{R}^{(N-1)^2 \times (N-1)^2}$ is

$$A \quad = \quad \begin{pmatrix} B & C & & & \\ C & B & C & & \\ & \ddots & \ddots & \ddots & \\ & & C & B & C \\ & & & C & B \end{pmatrix}$$

Linear System

In this expression, $B \in \mathbb{R}^{(N-1)\times(N-1)}$ is a tridiagonal matrix of the form

$$B = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}$$

and $C = -I_{N-1}$ (where I_{N-1} denotes the identity matrix of size $N-1$).

Solution Methods: Splitting Methods

Recall from the Numerical Linear Algebra course we may solve a linear system $A\mathbf{U} = \mathbf{f}$ using a splitting method by writing $A = M - N$. Then the system becomes

$$M\mathbf{U} = N\mathbf{U} + \mathbf{f}$$

and iterative methods based on this are:

1. Make an initial guess $\mathbf{U}^{(0)}$ of solution
2. Solve the linear system

$$M\mathbf{U}^{(k+1)} = N\mathbf{U}^{(k)} + \mathbf{f}$$

Repeat until convergence achieved.

Success of Splitting Methods

Splitting methods are likely to be successful if it is easy to solve systems with the matrix M.

- Jacobi — $M = D$ where D is the diagonal of the matrix A (requires diagonal solve)
- Gauss-Seidel — $M = L + D$ where D is the diagonal and L is the strict lower triangle of the matrix A (requires solution with lower triangular matrix — forwards substitution)

Of course we also require convergence. It can be shown that

$$\mathbf{U} - \mathbf{U}^{(k)} \quad = \quad (M^{-1}N)^k(\mathbf{U} - \mathbf{U}^{(0)})$$

Thus if $M^{-1}N$ is diagonalisable, $\|\mathbf{U} - \mathbf{U}^{(k)}\| \to 0$ as $k \to \infty$, for any initial guess $\mathbf{U}^{(0)}$, iff $\rho(M^{-1}N) < 1$.

Summary of Common Splitting Methods

Write $A = D + L + U$ where D is the diagonal, L is the strict lower triangle and U is the strict upper triangle of A. Then common splitting methods are:

- Jacobi: $D\mathbf{U}^{(k+1)} = -(L+U)\mathbf{U}^{(k)} + \mathbf{f}$
- Relaxed Jacobi: $D\mathbf{U}^{(k+1)} = [(1-\omega)D - \omega(L+U)]\mathbf{U}^{(k)} + \omega\mathbf{f}$
- Gauss-Seidel: $(D+L)\mathbf{U}^{(k+1)} = -U\mathbf{U}^{(k)} + \mathbf{f}$
- SOR: $(D+\omega L)\mathbf{U}^{(k+1)} = [(1-\omega)D - \omega U]\mathbf{U}^{(k)} + \omega\mathbf{f}$
  ($\omega < 1$ corresponds to under-relaxation, $\omega = 1$ corresponds to Gauss-Seidel and $\omega > 1$ corresponds to over-relaxation)
- SSOR: $(D+\omega L)\mathbf{U}^{(k+1/2)} = [(1-\omega)D - \omega U]\mathbf{U}^{(k)} + \omega\mathbf{f}$
  $(D+\omega U)\mathbf{U}^{(k+1)} = [(1-\omega)D - \omega L]\mathbf{U}^{(k+1/2)} + \omega\mathbf{f}$

Splitting Methods: Component Form

For our elliptic problems, the matrix A is very sparse and we can take advantage of this. Thus in component form we can write the Jacobi method as

$$U_{i,j}^{(k+1)} = \frac{1}{4}\left(U_{i-1,j}^{(k)} + U_{i+1,j}^{(k)} + U_{i,j-1}^{(k)} + U_{i,j+1}^{(k)} + f_{i,j}\right)$$

for $i, j = 1, \ldots, N-1$ (and where boundary values are 0).

Similar expressions are possible for other splitting methods.

Convergence

It can be shown that the eigenvalues for the Jacobi iteration matrix ($M^{-1}N$) are given by

$$\lambda^{r,s} = \frac{1}{2}(\cos(r\pi h) + \cos(s\pi h))$$

and so for Jacobi's method

$$\rho(M^{-1}N) = \cos(\pi h) \approx 1 - \frac{\pi^2 h^2}{2} + \mathcal{O}(h^4)$$

so that convergence gets worse as h gets smaller (and making h smaller improves the numerical solution to the underlying PDE).

Convergence

If we use the relaxed Jacobi method then the eigenvalues of the iteration matrix are given by

$$\lambda^{r,s} = (1-\omega) + \frac{\omega}{2}(\cos(r\pi h) + \cos(s\pi h)) \ .$$

In particular, if we choose $\omega = 0.5$ we have

$$\lambda^{r,s} = \frac{1}{2} + \frac{1}{4}(\cos(r\pi h) + \cos(s\pi h)) \ ,$$

and so all eigenvalues lie in the interval $(0, 1)$. The corresponding eigenfunctions for $M^{-1}N$ have entries

$$V_{i,j}^{r,s} = \sin(ri\pi h)\sin(sj\pi h) \ .$$

Thus high frequency eigenvectors (r, s large) correspond to small eigenvalues.

Convergence

We can expand the error in the initial guess in terms of these eigenfunctions so that

$$\mathbf{U} - \mathbf{U}^{(0)} = \sum_{r,s=1}^{N-1} \alpha_{r,s} \mathbf{V}^{r,s}$$

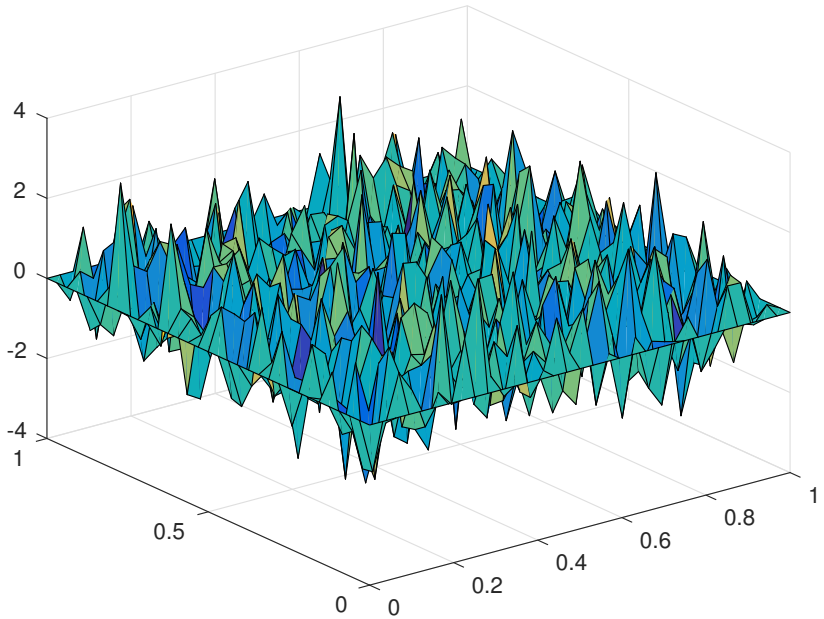
and then the error at later iterations is given by

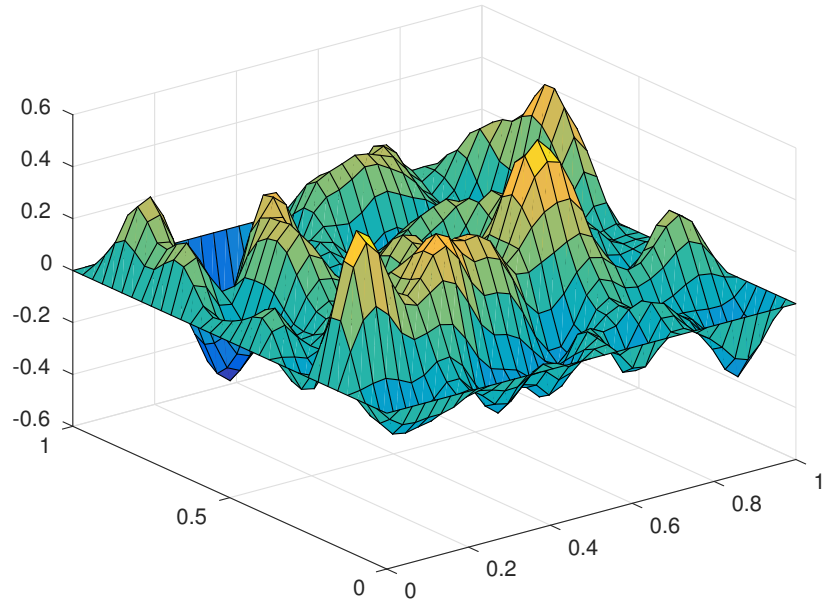
$$\mathbf{U} - \mathbf{U}^{(k)} = \sum_{r,s=1}^{N-1} \alpha_{r,s} (\lambda^{r,s})^k \mathbf{V}^{r,s} \ .$$

This means that the high frequency parts of the error are quickly damped out.

Smoothing

The left hand figure below shows the initial (high frequency) error in a PDE solution while the right hand figure shows the solution after a few steps of relaxed Jacobi. The high frequency components have been damped out. This error could then be approximated on a coarser grid. This is the idea behind multigrid.





Further Remarks

- The derivation of the method we used shows that the truncation error for the finite difference scheme is of size $\mathcal{O}(h^2)$.
- There is no need to use the same mesh size in both the x and y directions. In fact there may be cases where this is not sensible.
- There are other numerical methods for solving elliptic PDEs which are more flexible in terms of geometry and are easily extended to higher order convergence. One example is the Finite Element Method (see next term's special topic).
- In the computations we make there are two sources of error and it is important to distinguish them. The first is the error committed by solving the PDE (which is of size $\mathcal{O}(h^2)$), the second is the error committed by not solving the linear system perfectly. The size of this depends on the tolerance used in the linear solver. With a direct solver (e.g. Matlab's backslash command), it is assumed that this second error is negligible.

Further Remarks

- There are many more methods which can be used to solve linear systems (arising from PDEs) including
  - Conjugate gradient method — for symmetric positive definite systems
  - MINRES — minimum residual method for symmetric indefinite systems
  - GMRES — for non-symmetric systems
  - ...
- Many of these iterative methods can be sped up by the use of preconditioning. (For more details see later Numerical Linear Algebra lectures and the related Scientific Computing Case Study next term.)