Finite Difference Solver of a Poisson Equation in One Dimension

The objective of this assignment is to guide the student to the development of a finite difference method (FDM) solver of a Poisson Equation in one dimension from scratch. This assignment consists of both pen-and-paper and implementation exercises.

We request the students to prepare a report on these assignments. We appreciate receiving a clearly structured report with an introduction, body and conclusions.

1 Finite Difference Method in 1D

In the first part of this assignment we aim at solving the Poisson equation on the open interval \( \Omega = (0, 1) \).

Our objective is to numerically approximate the function \( u(x) \) that is the solution of the following problem:

given the source function \( f(x) \) and the number \( \alpha \), find the function \( u(x) \) such that \( u(x) \) is the solution of the differential equation

\[
-\frac{d^2 u}{dx^2} = f(x)
\]

supplied with the following homogeneous Dirichlet boundary condition in \( x = 0 \) and the non-homogeneous Neumann boundary conditions in \( x = 1 \).

\[
u(x) = 0 \quad \text{and} \quad \frac{du}{dx}(x = 1) = \alpha.
\]

Compulsory Analytical Part

Assignment 1  Choose \( f(x) = x \) and \( \alpha = 0.5 \) and show that the function \( u(x) = \frac{1}{3}x^3 + x \) is the exact solution of the problem. Do so by verifying using pen and paper that \( u(x) \) satisfies the differential equation as well as both boundary condition. There is no need to use symbolic differentiation, not to use integration to construct \( u(x) \). We will use the short hand notation \( u'(x) = \frac{du}{dx} \) and \( u''(x) = \frac{d^2 u}{dx^2} \).

Assignment 2  Plot the function \( u(x) \) as a function of \( x \) for \( 0 \leq x \leq 1 \) by avoiding for-loops in Matlab.

Compulsory Numerical Part

Assume that the interval \( \Omega \) is discretized by an uniform mesh consisting of \( N \) elements with mesh width \( h = 1/N \) and vertices \( x_i = (i - 1) h \), where \( i \) runs from 1 to \( N + 1 \). This enumeration includes the end points of \( \Omega \), that is, \( x_1 = 0 \) and \( x_{N+1} = 1 \). The grid nodes can then be denotes by

\[
G_h = \{ x_i \mid x_i = (i - 1) h; h = \frac{1}{N}, 0 \leq i \leq N + 1 \}.
\]

Assignment 3  (Discretization in the interior nodes) The differential equation holds in particular for all of the internal nodes on \( \Omega \), that is, we have that

\[
-\frac{d^2 u}{dx^2} \bigg|_{x=x_i} = f(x_i) \quad \text{for all } 2 \leq x_i \leq N.
\]

Use a finite difference formula twice to show that the second derivative \( \frac{d^2 u}{dx^2} \bigg|_{x=x_i} = u''(x_i) \) can be discretized as follows

\[
u''(x_i) \approx \frac{u'(x_i + h/2) - u'(x_i - h/2)}{h} = \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2}.
\]
(Using Taylor polynomials is can be shown that the local truncation error is of second order in \( h \)). The finite difference discretization thus leads to the following stencil for the approximation of \( \frac{d^2 u}{dx^2} \) (beware of the minus-sign)

\[
\frac{1}{h^2} \begin{bmatrix}
-1 & 2 & -1 \\
1 & -2 & 1 \\
\end{bmatrix}
\]

This stencil implies that each node \( x_i \) is coupled to its left \( (x_{i-1}) \) and right neighbour \( (x_{i+1}) \) with a weight of \(-\frac{1}{h^2}\).

**Assignment 4 (Discretization in the left end point)** Verify that the Dirichlet boundary condition in the left end point can be enforced by requiring that

\[
u_1 = 0.
\]

The finite difference stencil in the left end point thus reduces to

\[
\begin{bmatrix}
1 & 0 & 0 \\
1 & 0 & 0 \\
\end{bmatrix}
\]

**Assignment 5 (Discretization in the right end point)** Verify that the Neumann boundary condition in the right end point can be enforced by requiring that

\[
\frac{du}{dx}(x = 1) = \frac{u_{N+1} - u_N}{h} = \alpha.
\]

The finite difference stencil in the right end point thus reduces to

\[
\frac{1}{h} \begin{bmatrix}
0 & 1 & -1 \\
0 & 1 & -1 \\
\end{bmatrix}
\]

**Assignment 6** Assume \( h = 1/3 \) (and thus N=3). Determine the size of the global matrix \( A^h \) and the global right-hand vector \( f \). Give all the elements of this matrix and vector with pen (or pencil) on paper.

**Assignment 7** Assume \( h = 1/4, 1/8, 1/16, \ldots \) and assemble for all these values the global matrix \( A^h \) and the global right-hand vector \( f^h \). Solve the linear system

\[
A u^h = f^h
\]

using the Matlab backslash (\) command. Plot the various solution for \( u^h \) found and compare this plot with the plot of the exact solution in the first assignment.

**Elective Assignments**

**Assignment 8** Redo Assignment 7 for other choices for \( f(x) \) and/or \( \alpha \).

**Assignment 9** Verify that the numerical scheme is indeed second order accurate by investigating how the max-norm of the discretization error

\[
E = \|u(x) - u^h(x)\|_\infty = \max_{1 \leq i \leq N+1} |u(x_i) - u^h(x_i)|
\]

scales with the meshwidth \( h \) as expected.

**Assignment 10** Assemble the matrix \( A^h \) and the vector \( f^h \) avoiding for-loops in Matlab.

**Assignment 11** Extend your implementation is such a way to be able to treat a variable diffusion coefficient \( c(x) \), i.e, the differential equation

\[
- \frac{d}{dx} \left[ c(x) \frac{du}{dx} \right] = f(x).
\]