Poisson Differential Equation

Problem Description

Find a function $u : [0,1]^2 \to \mathbb{R}$ that satisfies the Poisson differential equation

$$-\frac{\partial^2 u(x,y)}{\partial x^2} - \frac{\partial^2 u(x,y)}{\partial y^2} = f(x,y),$$

where $f : [0,1]^2 \to \mathbb{R}$ is a given continuous function, and u is zero on the boundary of the square $[0,1]^2$.

Discretization Method

To solve this numerically, we discretize the equation by assuming u_h has a specific form dependent on finitely many variables and setting up a system of equations to solve for these variables.

We introduce a regular grid of points in the square $[0,1]$. Let $N > 1$ be a natural number and define the points (x_i, y_j) where $x_i = i/N$ and $y_j = j/N$ for $i, j = 0, \ldots, N$. If i or j equals 0 or N, the point (x_i, y_j) lies on the boundary.

Define $h = 1/N$ so that $x_i = ih$ and $y_j = jh$. The function u_h is defined only at the points (x_i, y_j) with values u_{ij}. For points inside the square $[0,1]$, the sum of the second-order partial derivatives is replaced by

$$\frac{1}{h^2}(-u_{i-1,j} - u_{i,j-1} + 4u_{ij} - u_{i+1,j} - u_{i,j+1}).$$

Thus, we obtain a system of $(N-1)^2$ linear equations

$$\frac{1}{h^2}(-u_{i-1,j} - u_{i,j-1} + 4u_{ij} - u_{i+1,j} - u_{i,j+1}) = f_{ij},$$

where $f_{ij} = f(x_i, y_j)$.

Solving the System

The matrix of this system consists of $(N-1)^2$ blocks of size $(N-1) \times (N-1)$, with a block tridiagonal structure:

$$A = \frac{1}{h^2} \begin{bmatrix} T & -I & 0 & \cdots & 0 \\ -I & T & -I & \cdots & 0 \\ 0 & -I & T & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & -I \\ 0 & 0 & 0 & -I & T \end{bmatrix},$$

where each I is the $(N-1) \times (N-1)$ identity matrix and T is the tridiagonal matrix:

$$T = \begin{bmatrix} 4 & -1 & 0 & \cdots & 0 \\ -1 & 4 & -1 & \cdots & 0 \\ 0 & -1 & 4 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & -1 \\ 0 & 0 & 0 & -1 & 4 \end{bmatrix}.$$

Implementation

Implement an Octave script to find the approximate solution u_h using an iterative method like the conjugate gradient method. Let N be a parameter to choose. Assume $f(x, y) = x$ or another low-degree polynomial. Print the norm of the residual vector $b - Ax_k$ in each iteration. Optionally, plot the solution using mesh or surf.