Boundary value problems

January 5, 2015

We want to implement the central differences discretization of the following boundary value problem :

$$u''(x) = f(x), \qquad 0 < x < 1,$$

$$u(0) = \alpha, \qquad u(1) = \beta.$$

Considering the grid of equidistant points

$$x_j = j \cdot h, \qquad j = 0, 1, \ldots, M+1, \qquad h = \frac{1}{M+1}.$$

On each node x_j we replace the second derivative in the differential equation with its approximation by central differences and get

$$\frac{1}{h^2}(U_{i-1} - 2U_i + U_{i+1}) = f_i, \qquad i = 1, \ldots, M.$$

Using the boundary conditions $U_0 = \alpha$, $U_{M+1} = \beta$ we get a system of M equations in the M unknowns $U_1, \ldots, U_M$ that is

$$A_h \vec{U} = \vec{F}$$

where $\vec{U} = [U_1, \ldots, U_M]^T$, $\vec{F} = [f_1 - \frac{\alpha}{h^2}, f_2, \ldots, f_{M-1}, f_M - \frac{\beta}{h^2}]^T$ and

$$A_h := \frac{1}{h^2} \begin{bmatrix} -2 & 1 & 0 & & \\ 1 & -2 & 1 & \ddots & \\ 0 & \ddots & \ddots & \ddots & 0 \\ & \ddots & \ddots & \ddots & 1 \\ & & 0 & 1 & -2 \end{bmatrix}.$$

Task 1 Choose $f = \sin(\pi x)$, $\alpha = \beta = 0$ and $M = 10$ construct the linear system $A_h \vec{U} = \vec{F}$ and solve it with the *backslash* command of Matlab (or in other ways if you are using another programming language) to find the numerical solution $\vec{U}$. Plot the numerical solution. Find the solution[1] of the boundary value problem by integrating twice. Plot the values of the solution on the grid and compare them to the corresponding numerical approximation values.

[1] We will sometimes call the solution *exact solution* to distinguish it from the *numerical solution* which is the approximation produced by a numerical method.

Run the program with different values of M and observe the behaviour of the numerical method. Choose then another f leading to non trivial boundary values and repeat the exercise.

Task 2 In this task we want to see with a numerical experiment how the function-norm of the error decreased as a function of h. To this end use the values of the exact solution on the grid points to compute the the error vector $\vec{e}_h := [U_1 - u_1, \ldots, U_M - u_M]^T$, where $u_j = u(x_j)$. Consider then the piecewise constant error function defined by

$$e_h(x) := e_j, \qquad x \in [x_j, x_{j+1}), \qquad j = 1, \ldots, M.$$

We know from Taylor theorem that the exact solution can be expanded as

$$\frac{1}{h^2}(u_{j+1} - 2u_j + u_{j-1}) = u''(x_j) + \frac{h^2}{12}u^{(4)}(x_j) + \mathcal{O}(h^4).$$

It can also be proved that $e_h(x)$ is going to zero in the 2-norm (for functions) as $\mathcal{O}(h^2)$[2].

We design our numerical experiment as follows: consider increasing values of M, for example $M = 2^k$, $k = 1, 2, \ldots, 8$ and decreasing values of h accordingly. Solve the linear system from taks 1 for each value of M and compute the corresponding norm of the error, (use max-norm, 1-norm and 2-norm), store the obtained values. Plot in logarithmic scale the different values of h versus the corresponding values of the error norm (for the three different choices of norm), you should observe a straight line with slope 2 (testifying second order convergence).

Task 3 You should now modify your programme and implement Neumann boundary conditions (follow the description of chapter 3.1.2 in the note). There are several strategies: CASE 1 is a first order method, CASE 2 is a second order method using fictitious nodes, CASE 3 is a second order method leading to a matrix which is not tridiagonal but without using fictitious nodes. Implement each of these and verify the order of each technique numerically.

Task 4 Implement the method described in section 3.2.1 of the note, where a general self-adjoint linear boundary value problem is considered and discretized so to preserve symmetry under discretization. Verify the order.

[2] We will see the proof in one of the first lectures of the course. To do this we will use the fact that A_h is **invertible** with **inverse bounded in 2-norm** independently on h. This is called (order 2) convergence.