Fall 2008 A. Chertock

MA 584 - Homework #2

due on October 2, 2008

1. Derive the finite difference method for

$$\begin{aligned} u''(x) - q(x)u(x) &= f(x), \quad a < x < b, \\ u(a) &= u(b), \quad \text{periodic BC}, \end{aligned}$$

   using the central difference scheme and a uniform grid. Write down the system of equations $L_hU = F$. How many unknowns are there without redundant? Is the coefficient matrix L_h tri-diagonal? **Hint:** Note that $U_{-1} = U_{N-1}$.

2. (a) Modify the m-file `bvp2.m` (available from the website) so that it implements a Dirichlet boundary condition at $x = a$ and a Neumann condition at $x = b$ and test the modified program.

   (b) Make the same modification to the m-file `bvp4.m` (available from the website), which implements a fourth order accurate method. Again test the modified program.

3. Write down the coefficient matrix of the finite difference method using the standard central five point stencil with the Red-Black and the Natural row ordering for the Poisson equation defined on the rectangle $[a, b] \times [c, d]$. Take $m = n = 3$ and assume a Dirichlet boundary condition at $x = a$, $y = c$ and$y = d$, and a Neumann boundary condition $\dfrac{\partial u}{\partial n} = g(y)$ at $x = b$. Use the ghost point method to deal with the Neumann boundary condition.

4. The MATLAB script `poisson.m` (available from the website) solves the Poisson problem on a square $m \times m$ grid with $\Delta x = \Delta y = h$, using the 5-point Laplacian. It is set up to solve a test problem for which the exact solution is $u(x, y) = \exp(x + y/2)$, using Dirichlet boundary conditions and the right hand side $f(x, y) = 1.25 \exp(x + y/2)$.

   (a) Test this script by performing a grid refinement study to verify that it is second order accurate.

   (b) Modify the script so that it works on a rectangular domain $[a_x, b_x] \times [a_y, b_y]$, but still with $\Delta x = \Delta y = h$. Test your modified script on a non-square domain.

   (c) Further modify the code to allow $\Delta x \neq \Delta y$ and test the modified script.

5. (a) Show that the 9-point Laplacian (3.17) has the truncation error derived in Section 3.5. **Hint:** To simplify the computation, note that the 9-point Laplacian can be written as the 5-point Laplacian (with known truncation error) plus a finite difference approximation that models $\frac{1}{6}h^2 u_{xxyy} + O(h^4)$.

(b) Modify the MATLAB script `poisson.m` to use the 9-point Laplacian (3.17) instead of the 5-point Laplacian, and to solve the linear system (3.18) where f_{ij} is given by (3.19). Perform a grid refinement study to verify that fourth order accuracy is achieved.

6. Implement and compare the Jacobi, Gauss-Seidel, and the SOR (trying to find the best ω by testing) methods for the following elliptic problem:

$$u_{xx} + p(x,y)u_{yy} + r(x,y)u(x,y) = f(x,y), \qquad 0 < x < 1, \quad 0 < y < 1, \tag{1}$$

subject to the boundary conditions:

$$u(0,y) = u(x,0) = u(x,1) = 0, \quad \frac{\partial u}{\partial x}(1,y) = \phi(y). \tag{2}$$

Test and debug your code for the case

$$p(x,y) = (1 + x^2 + y^2), \qquad r(x,y) = -xy, \qquad \phi(y) = -\pi \sin(\pi y). \tag{3}$$

The source term $f(x,y)$ is determined from the exact solution

$$u(x,y) = \sin(\pi x)\sin(\pi y). \tag{4}$$

Do the grid refinement analysis for $n = 16$, $n = 32$, and $n = 64$ (if possible) in the infinity norm. Take the tolerance as 10^{-3}. Does the method behave like a second order method? Compare also the number of iterations and test the optimal relaxation factor ω. Plot the solution and the error for $n = 32$.

Having made sure that you code is working correctly, try your code with a point source $f(x,y) = \delta(x-0.5)\delta(y-0.5)$ and $u_x = -1$ at $x = 1$, with $p(x,y) = 1$ and $r(x,y) = 0$. Note that the $u(x,y)$ can be interpreted as the steady state temperature distribution of a room with insulated wall on three sides, a constant heat flow in from one side, and a point source (a heater for example) in the room. Note that the heat source can be expressed as $f(n/2, n/2) = 1/h^2$, and $f(i,j) = 0$ for other grid points. Use the mesh and contour plots to visualize the solution for $n = 36$ ($\texttt{mesh}(x,y,u)$, $\texttt{contour}(x,y,u,30)$).