Homework # 4

Due: October 7, 2013

---

1. Let A be the symmetric, positive definite, tridiagonal matrix given by

$$(\dagger)\qquad A = \begin{pmatrix} a_1 & b_2 & & & & \\ b_2 & a_2 & b_3 & & & \\ & b_3 & a_3 & b_4 & & \\ & & \ddots & \ddots & \ddots & \\ & & & b_{n-1} & a_{n-1} & b_n \\ & & & & b_n & a_n \end{pmatrix}$$

a. Write the equations for performing a Cholesky factorization LL^T of A. Don't just give the general equations but justify how you got them. Justify that L has the form

$$L = \begin{pmatrix} \alpha_1 & 0 & & & & \\ \beta_2 & \alpha_2 & 0 & & & \\ & \beta_3 & \alpha_3 & 0 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \beta_{n-1} & \alpha_{n-1} & 0 \\ & & & & \beta_n & a_n \end{pmatrix}$$

that is, L is lower triangular such that $L_{ij} = 0$ for $i - j > 1$. Use the notation a_i, α_i, $i = 1, \ldots, n$ and b_i, β_i, $i = 2, \ldots, n$ that was introduced here.

b. Once you have the Cholesky factorization LL^T of A, write the equations for a forward and a backward solve to determine $\vec{x}$ from $LL^T\vec{x} = \vec{f}$.

c. Write a code to implement the solution of a symmetric, positive definite, tridiagonal system of equations using your results from (a) and (b). Test your code on the problem where $a_i = 2$, $i = 1, \ldots, n$ and $b_i = -1$, $i = 2, \ldots, n$ and the right hand side is

$$\vec{f} = \left(3, -4, 4, -4, 4, \cdots, -4, 4, -3\right)^T$$

with $n = 20$. The exact solution is $(1, -1, 1, -1, \cdots, 1, -1)^T$.

---

2. In the first problem you wrote a direct solver for the symmetric positive definite tridiagonal matrix ($\dagger$). In this problem you are asked to implement the iterative method SOR for the same matrix. Recall that SOR takes a weighted average of the previous iteration and the one that would be obtained by using Gauss-Seidel. The weight ω is chosen so that it reduces to Gauss-Seidel when $\omega = 1$.

a. Implement SOR in a component-wise manner rather than forming matrices and taking matrix times vector products; take advantage of the sparsity of the matrix in your loops. As a stopping criteria use

$$\frac{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2}{\|\mathbf{x}^{k+1}\|_2} \leq 10^{-5}$$

where $\mathbf{x}^k$ is the result of the kth iteration and $\|\cdot\|_2$ denotes the standard Euclidean norm.

b. Set $\omega = 1$ so that you are using the Gauss-Seidel method. Test your code on the problem in (1c) and compare results. Use an initial guess of $(1, 1, \ldots, 1)^T$.

c. Now we want to perform a study to determine the optimal value of ω where $0 < \omega < 2$ for the problem where $a_i = 2.5$, $i = 1, \ldots, n$ and $b_i = -1$, $i = 2, \ldots, n$ and the right hand side is $\vec{f} = \big(3, -4, 4, -4, 4, \cdots, -4, 4, -3\big)^T$. Use $n = 1000$. Starting with $\omega = 0.05$, increment ω by 0.05 and calculate the number of iterations required to get the stopping criteria is met or a maximum number of 500 iterations is reached. Plot the number of iterations versus ω. What is the optimal value of ω?

---

3. Suppose we have a BVP in one dimension for $u(x)$ and we want to use a finite difference approximation on a *nonuniform* grid. Assume that the grid spacing h_i is defined by $h_i = x_i - x_{i-1}$, $i = 1, \ldots, N$ for a grid $x_0, x_1, \ldots, x_N$.

a. Perform Taylor series expansions for $u(x_{i-1})$ and $u(x_{i+1})$ and then take an appropriate linear combination of the two expansions to obtain the difference approximation

$$u'(x_i) \approx \frac{1}{2h_i h_{i+1}} \Big[h_i u(x_{i+1}) - h_{i+1} u(x_{i-1}) + (h_i - h_{i+1}) u(x_i) \Big] .$$

b. When we have a uniform grid, i.e., $h_i = h$ for all i, then what does this approximation reduce to?

---

4. (G) In the last homework you saw that the Trapezoidal method has good stability properties. When you compute with a large Δt the solution may not be very accurate but it doesn't grow in an unbounded way that most explicit methods do. However, to implement the Trapezoidal method for an IVP we have to solve a nonlinear equation. Assume that we want to do this using the Newton-Raphson method to compute Y_{i+1}. For the IVP $y'(t) = f(t, y)$ we need an iteration of the form $Y_{i+1}^{k+1} = g(t, Y_i, Y_{i+1}^k, f)$ and we need a starting guess Y_{i+1}^0; typically one takes Y_i as this guess. (i) First write the nonlinear equation $F(Y) = 0$ that must be solved for the Trapezoidal method for $y' = f$. (ii) Write the equation for Newton's method to determine Y_{i+1}^{k+1} in terms of Y_i, Y_{i+1}^k, f and f_y. (iii) Implement the Trapezoidal method using your equations for Newton's method for the IVP $y'(t) = -20y$, $y(0) = 1$ on $(0, 1]$ with $\Delta t = 1/4, 1/8, \ldots, 1/32$. Use the stopping criteria for Newton's method that the normalized difference in successive iterations is less than 10^{-8}.

Points

UG - #1- 20 points #2 - 20 points, # 3 - 10 points

G -#1- 15 points #2 - 15 points, # 3 - 5 points, # 4 - 15 points