

MATH 6644

Homework 1

Stephan Boettcher

February 16, 2015

Question 1

Construct a dense 3×3 matrix G with $\rho(G) > 1$. Define an iteration and find an example of \vec{x}_0 and \vec{c} such that the iteration does not converge.

Beginning with the dense A matrix of: $A = \begin{bmatrix} 1 & 2 & 1 \\ 4 & 9 & 2 \\ 7 & 20 & 2 \end{bmatrix}$, we generate the G matrix by first constructing the

N and M matrices. The M matrix is defined as the diagonal elements of A and the N matrix is defined as: $N = M - A$, or the off diagonal elements multiplied by -1. The G matrix is then constructed by $G = M^{-1} * N$,

which creates the following matrix: $G = \begin{bmatrix} 0 & -2 & -1 \\ -\frac{4}{9} & 0 & -\frac{2}{9} \\ \frac{7}{2} & -10 & 0 \end{bmatrix}$. The spectral radius of the G matrix is 2.9413.

This spectral radius makes for an unstable iteration matrix that will most likely result in the iterative method going to infinity, rather than converging on a value. An iteration for this matrix is defined as:

$$\vec{x}_{k+1} = G\vec{x}_k + \vec{c}$$

Where \vec{c} is defined as: $\vec{c} = M^{-1}\vec{b}$. The \vec{b} is the final vector we are iterating to. To find an \vec{x}_0, \vec{c}_0 , the vector \vec{b} must be defined. For this assignment, the following values were chosen for these vectors:

$$\vec{b} = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}, \vec{x}_0 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \vec{c}_0 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

With these values, combined with the unstable G matrix, the iterative method quickly spirals out of control and goes to infinity within 100 iterations.

Question 2

Give the matrix expression for the symmetric Gauss-Seidel iterations.

The Gauss-Seidel (G-S) method can be expressed as both an element-wise expression and in matrix expressions. To express the G-S method in matrix expression form, the notation must first be developed. Given an $n \times n$ nonsingular matrix A, $\vec{x} \in \mathbb{R}^n$, and $\vec{b} \in \mathbb{R}^n$, we can split the A matrix into:

$$A = D - E - F$$

where D is the diagonal elements of A , E is the lower triangular matrix of A , and F is the upper triangular matrix of A . With these matrices defined, the symmetric G-S iteration can be expressed in matrix form.

The symmetric G-S requires two sweeps: a forward sweep followed by a backward sweep. This is done to increase the rate of convergence of the G-S method. The forward sweep portion of the iteration is defined as:

$$\vec{x}^{k+\frac{1}{2}} = (D - E)^{-1} F \vec{x}^{(k)} + (D - F)^{-1} \vec{b}$$

The forward sweep is then followed by the backward sweep which is defined as:

$$\vec{x}^{k+1} = (D - F)^{-1} E \vec{x}^{(k+\frac{1}{2})} + (D - F)^{-1} \vec{b}$$

Thus, the matrix expression for the symmetric G-S iterations is:

$$\begin{aligned}\vec{x}^{k+\frac{1}{2}} &= (D - E)^{-1} F \vec{x}^{(k)} + (D - F)^{-1} \vec{b} \\ \vec{x}^{k+1} &= (D - F)^{-1} E \vec{x}^{(k+\frac{1}{2})} + (D - F)^{-1} \vec{b}\end{aligned}$$

Question 3

Assume A is a $n \times n$ symmetric positive definite matrix with all eigenvalues in the intervals $(1; 1.1) \cup (10; 10.2)$. Assume that the cost of a matrix vector multiplication is about $4n$ floating point multiplications. Estimate the number of floating point operations reduce the A -norm of the error by a factor of 10^{-3} using CG iterations.

Because the $n \times n$ A matrix is symmetric positive definite, convergence can be guaranteed for the Conjugate Gradient (CG) method. This allows an upper limit on the number of floating point operations (FLOPS) to be calculated. This upper bound will be dependent on the theoretical convergence rate of the CG method. The analysis of the CG method assumes that variable assignment and vector addition/subtraction can occur in $\mathcal{O}(1)$ or constant time. This is theoretically possible on vector machines. The analysis also assumes that 'vector division' (i.e. the computation of the α and β values) can also be completed in $\mathcal{O}(4n)$. The breakdown of the FLOPS cost per step of the CG method is below:

$$\begin{aligned}\vec{r}^{(0)} &= \vec{b} - A\vec{x} && 4n \text{ FLOPS} \\ \vec{p}^{(0)} &= \vec{r}^{(0)} && 1 \text{ FLOPS} \\ \text{for } i &= 1 : k && k \text{ iterations} \\ \{ &&& \\ \vec{q}^{(k-1)} &= A\vec{p}^{(k-1)} && 4n \text{ FLOPS} \\ \alpha_{k-1} &= \frac{\langle \vec{r}^{(k-1)}, \vec{r}^{(k-1)} \rangle}{\langle \vec{p}^{(k-1)}, \vec{q} \rangle} && 8n \text{ FLOPS total} \\ \vec{x}^{(k)} &= \vec{x}^{(k-1)} + \alpha_{k-1} \vec{p}^{(k-1)} && 4n \text{ FLOPS} \\ \vec{r}^{(k)} &= \vec{r}^{(k-1)} - \alpha_{k-1} \vec{q} && 4n \text{ FLOPS} \\ \beta_{k-1} &= \frac{\langle \vec{r}^{(k)}, \vec{r}^{(k)} \rangle}{\langle \vec{r}^{(k-1)}, \vec{r}^{(k-1)} \rangle} && 8n \text{ FLOPS total} \\ \vec{p}^{(k)} &= \vec{r}^{(k)} + \beta_{k-1} \vec{p}^{(k-1)} && 4n \text{ FLOPS} \\ \} &&& \end{aligned}$$

Thus it takes $\mathcal{O}(36n)$ FLOPS to complete one iteration. The error equation is given by:

$$\|\vec{e}^{(k+1)}\|_A \leq \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right) \|\vec{e}^{(k)}\|_A$$

For this problem, $\kappa(A) = \frac{10.2}{1.1} = 9.\overline{2727}$. By the error equation above, for each iteration, the error will be reduced by $\left(\frac{\sqrt{9.\overline{2727}} - 1}{\sqrt{9.\overline{2727}} + 1} \right) \approx 0.5056$. As a result, after 10 iterations, the error will be reduced by $\approx 10^{-3}$. Thus, the number of FLOPS to reduce the A-norm of the error by a factor of 10^{-3} is $\mathcal{O}(360n)$.

Question 4

Assume that A is a $n \times n$ symmetric positive definite matrix. If $\kappa(A) = \mathcal{O}(n)$, give a rough estimate of the number of CG iterations required to reduce the relative residual to $\mathcal{O}(\frac{1}{n})$.

The residual matrix is defined as $\vec{r}^{(k)} = \vec{b} - A\vec{x}^{(k)}$. The error vector can be written as $A\vec{e}^{(k)} = A(\vec{x}^{(*)} - \vec{x}^{(k)}) = \vec{b} - A\vec{x}^{(k)}$. Thus, $\vec{r}^{(k)} = A\vec{e}^{(k)}$. Since $\kappa(A) = \mathcal{O}(n)$, the error equation can be written as:

$$\|A^{-1}\vec{r}^{(k+1)}\|_A \leq \left(\frac{\sqrt{\mathcal{O}(n)} - 1}{\sqrt{\mathcal{O}(n)} + 1} \right) \|A^{-1}\vec{r}^{(k)}\|_A$$

To determine the number of iterations necessary to reduce the error to a given value, the following equation must be solved:

$$\left(\frac{\sqrt{\mathcal{O}(n)} - 1}{\sqrt{\mathcal{O}(n)} + 1} \right)^x = \mathcal{O}\left(\frac{1}{n}\right)$$

where x is the number of iterations. Solving for x , we get:

$$x = \frac{-\log(\mathcal{O}(n))}{\log(\sqrt{\mathcal{O}(n)} - 1) - \log(\sqrt{\mathcal{O}(n)} + 1)}$$

To provide a quicker, rougher estimate, the above equation follows the equation $x' = \frac{3 \cdot 2^{\log(n)}}{4}$ fairly closely for multiple values of n . While the values provided by x' do not exactly match that generated by x , the x' values can be computed quicker and will give the correct order of magnitude.

Question 5

Discretize the following differential equation:

$$\begin{cases} -u'' + 4u = & x \in [0, 1] \\ u(0) = -1, & u(1) = 2 \end{cases}$$

by the central difference scheme. Write your linear system of equations (you must give the matrix A and \vec{b}). Solve the system by using classical iteration such as Jacobi, Gauss-Seidel and SOR with $n = 1000$. Test you relaxation parameter in SOR for several values and decide which one is better. You need to discuss your results.

The above differential equation can be solved analytically to generate a truth curve, against which the numerical methods can be compared. The solution for this particular ODE is:

$$u(x) = \frac{e^{-2x}(e^{4x} + 2e^{4x+2} - 2e^2 - e^4)}{e^4 - 1}$$

To solve this equation numerically, the equation must first be discretized via the central difference theorem. The first step is to discretize the term $-u''$:

$$u'' = \frac{d^2u}{dx^2} = \frac{1}{h} \left(\frac{u(x+h) - u(x)}{h} - \frac{u(x) - u(x-h)}{h} \right) = \frac{1}{h^2} \left(u(x+h) - 2u(x) + u(x-h) \right)$$

This gives the final form of the differential equation, with the same boundary conditions, of:

$$0 = -\frac{1}{h^2} \left(u(x+h) - 2u(x) + u(x-h) \right) + 4u(x)$$

Realizing that $h = x_i - x_{i-1}$, this can be simplified and rewritten as:

$$0 = u_{i+1} - 2u_i + u_{i-1} - 4h^2u_i$$

This form of the equation can be used to generate the A and \vec{b} variables, where $A\vec{u} = \vec{b}$:

$$A = \begin{bmatrix} -2 - 4h^2 & 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & -2 - 4h^2 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & -2 - 4h^2 & 1 & 0 & \dots & 0 \\ & & \ddots & \ddots & \ddots & & \\ 0 & \dots & \dots & \dots & \dots & 1 & -2 - 4h^2 \end{bmatrix}, \vec{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-2} \\ u_{n-1} \end{bmatrix}, \vec{b} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -2 \end{bmatrix}$$

where $h = \frac{1}{n}$. This discretized ODE was then solved using the Jacobi, Gauss-Seidel, and SOR with $n = 1000$.