Q1. Numerical Linear Algebra: (Dr. Peterson) Summer 2016

Amirhessam Tahmassebi

June 23, 2016

a)

Stationary Iterative Method:

$$x^{k+1} = px^k + c$$

These are methods where the data in the equation remains fixed. Solve a linear system with an operator approximating the original one and based on a measurement of the error in the result. Jacobi, Gauss-Seidel, and Successive Over Relaxation (SOR) are three examples.

Non-Stationary Iterative Method:

$$x^{k+1} = x^k + \sigma_k d_k$$

These are methods where the data changes at each iteration. σ_k and d_k change for each iteration. d_k called search direction and σ_k is the step length. Krylov space method, Conjugate Gradient (CD), Generalized Minimal Residual (GMRES), BICG, and BICGSTAB are some examples of this method.

b)

$$A = T + L + U$$

For each numerical method, we have two ways: Stable and Cheap.
As we recall, for Gauss-Seidel we keep a lower triangular matrix on the left hand side and move the upper triangular to the right hand side, so we could use forward substitution.
But, here we are dealing with a full lower triangular matrix which cost the order of n^2. Solving a tridiagonal matrix is the order of n.

$$Ax = b$$

$$(T + L + U)x = b \Rightarrow Tx = b - (L + U)x$$
$$X^{k+1} = -T^{-1}(L + U)x^k + T^{-1}b$$
$$\Rightarrow p = -T^{-1}(L + U)x^k, c = T^{-1}b$$

c)

Recall SOR method:
Given a square system of n linear equations with unknown x:

$$Ax = b$$

A can be decomposed into a diagonal component D, and strictly lower and upper triangular components L and U:

$$A = D + L + U$$

The system of linear equations may be rewritten as:

$$(D + \omega L)x = \omega b - (\omega U + (\omega - 1)D)x$$

$$x_i^{k+1} = (1-\omega)x_i^k + \frac{\omega}{a_{ii}}(b_i - \sum_{j<i} a_{ij}x_j^{k+1} - \sum_{j>i} a_{ij}x_j^k), with : i = 1, 2, ..., n$$

For finding the similarity with SOR we have:

$$Tx^{k+1} = b - (L+U)x^k \Rightarrow x^{k+1} - x^k = T^{-1}[b - (L+U+T)x^k]$$

We also have from SOR: $x^{k+1} = x^k + \omega(x^{k+1} - x^k)$

$$x^{k+1} = x^k + \omega(T^{-1}[b - (L+U+T)x^k])$$

$$x^{k+1} = T^{-1}[(1-\omega)T - \omega(L+U)]x^k + \omega b T^{-1}$$

It is obvious with $\omega = 1$ we have the same formula like part b.

d)

Pseudo Code:
input: Matrix A, Matrix b for RHS, initial guess x_0
We have to decompose matrix A to Matrices L, U, T
Set the counter = 1
While $Counter < P = Max_{iteration}$
Calculate Right-Hand-Side: $b - (L+U)x_0$
Call TDMA Solver to solve: $Tx = RHS$
Then Substitute x: $x_0 \leftarrow x$
Increase the counter: $counter \leftarrow counter + 1$
End While

We should also know the Thomas Algorithm to solve Tridiagonal Matrix T:

$$Tx = d \Rightarrow \begin{bmatrix} b_1 & c_1 & 0 & \dots & 0 \\ a_2 & b_2 & c_2 & \dots & 0 \\ 0 & a_3 & b_3 & \dots & 0 \\ 0 & \dots & \dots & \dots & c_{n-1} \\ 0 & 0 & \dots & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ . \\ . \\ d_n \end{bmatrix}$$

TDMA Pseudo Code:
Forward Elimination Phase:
for k=2 until n do:

$$m = \frac{a_k}{b_k - 1}$$

$$b_k = b_k - mc_{k-1}$$

$$d_k = d_k - md_{k-1}$$

End for Loop(k)
Backward Substitution Phase:

$$x_n = \frac{d_n}{b_n}$$

for k = n-1 stepdown until 1 do:

$$x_k = \frac{d_k - c_k x_{k+1}}{b_k}$$

End Loop(k)

For the Operation Count for each iteration we have:
1) LU: 2N multiplications, N additions
2) Forward: N multiplications, N additions
3) Backward: 2N multiplications, N additions
4) RHS : N^2 Multiplications , N Subtractions
Total = $9N + N^2$
For P iteration we will have: Total = $9PN + PN^2 \sim O(N^2)$

e)

I have tested out three different examples, Symmetric Positive Definite, Diagonally dominant matrix and a general matrix.
Theorem:
Thomas Algorithm will work if:

$$|b_i| > |a_i| + |c_i| \text{ with } i = 2, 3, ..., N-1$$

$$|a_1| > |c_1| and |a_N| > |c_N|$$