FEM, 2011 Fall

Programming Assignment #1

Name: ____________________

Student ID: ____________________

Problem Specifications

Write a program for solving the system $Ax = b$ by using (1) two-term and three-term Chebyshev acceleration iterative method and (2) Conjugate Gradient Method, which are also iterative methods for solving linear equations, just like Gauss-Seidel, SOR and SSOR.

Computational examples - solve the following problems:

Problem 1 Consider the following linear system:

$$x_0 = 0, \quad k_{i-1}(x_i - x_{i-1}) + k_i(x_i - x_{i+1}) + c_i x_i = b_i, \quad i = 1, ..., n, \quad x_{n+1} = 1,$$

which after the elimination of x_0 and x_{n+1} is written in the form $Ax = b$ with $x \in R^n$. Solve for $n =$ 20, 40.

Introduce the following notations: $h = 1/(n+1)$, $k_i = k((i+0.5)h)$, where $k(x)$ is given below in two cases:

(1) (constant k(x)): $k(x) = 1$, $c_i = 0$, $b_i = 0$, $i = 1, \ldots, n$.

(2) (jump in k(x)): $K = 2,\ 10,\ 100$ and

$$k(x) = 1,\ for\ 0 < x < 0.5,\ and\ k(x) = K,\ for\ 0.5 \leq x < 1,\ c_i = 0,\ b_i = 0.$$

Problem 2 Now the unknowns are given as a two dimensional array $x_{ij}, i, j = 0, \ldots, n+1$ that satisfy the system

$$(4 + h^2)x_{i,j} - x_{i-1,j} - x_{i+1,j} - x_{i,j-1} - x_{i,j+1} = h^2, x_{0,j} = x_{n+1,j} = x_{i,0} = x_{i,n+1} = 0.$$

Here $h = 1/(n+1)$ so that the system represents a finite difference approximation of the boundary value problem $-\Delta u + u = 1$ in $\Omega = (0,1) \times (0,1)$ and $u = 0$ on the boundary of Ω. Solve for $n = 20,\ 40$.

Problem 1 (Tri-diagonal system)

The linear system is ($n = 25$, 50, 100):

$$x_0 = 0, k_{i-1}(x_i - x_{i-1}) + k_i(x_i - x_{i+1}) + c_i x_i = b_i, i = 1, ..., n, x_{n+1} = 1$$

$h = 1/(n+1)$, $b = h^2(1, 1, ..., 1)^t$, $k_i = k((i+1.5)h)$ where $k(x)$ is given below
(1) (constant $k(x)$): $k(x) = 1$, $c_i = 0$, $b_i = 0$, $i = 1, ..., n$;
(2) (jump in $k(x)$): $K = 2$, 10, 100 and

$$k(x) = 1, for 0 < x < 0.5, and k(x) = K, for 0.5 < x < 1, c_i = 0$$

In this one dimensional boundary value problem on the interval $(0, 1)$, we use two-term and three-term Chebyshev acceleration iterative and Conjugate Gradient Methods for different $n's$. The results are listed in Table 1(using $||r^{(m)}||_2/||r^{(0)}||_2 <= TOL$ as stopping condition and set $TOL = 10^{-6}$, and the initial iterate is $x = (0, 0, ..., 0)^T$). The comparitions of Jacobi and SOR are also there. From the table, we can tell that,CG method is the best among first three, since it costs less iteration steps in all situations. Three-term Chebyshev is also acceptable. However, two-term Chebyshev seems to be influenced a lot by the choosing of recurrence parameter. The results in Table 1 of 2-term Chebyshev are using $K = 32$. Then we try $K = 48$ and $K = 64$ (results listed in Table 2.). One can easily find the huge difference among them. Also, the numbers of iteration steps proportion to number of dimension for all methods.

The iteration formulas of these three methods are

Two-term Chebyshev

$$x^{(k+1)} = (I - \omega_{k+1}A)x^{(k)} + \omega_{k+1}b$$

where $\omega_k = \frac{\omega_0}{1+\rho_0 u_k}, u_k = \cos(\frac{2k-1}{2K}\pi), \omega_0 = \frac{2}{\lambda+\Lambda}, \rho_0 = \frac{1-\frac{\lambda}{\Lambda}}{1+\frac{\lambda}{\Lambda}}, k = 1, 2, ..., K$

Three-term Chebyshev

$$x^{(1)} = \gamma(Gx^{(0)} + C) + (1-\gamma)x^{(0)}$$

$$x^{(k)} = \rho_k(\gamma(Gx^{(k-1)} + C) + (1-\gamma)x^{(k-1)}) + (1-\rho_{(k)})x^{(k-2)}, k \geq 2$$

where $G = I - \frac{1}{\Lambda}A, C = \frac{1}{\Lambda}b, 0 = a \leq \lambda(G) \leq b = 1 - \frac{\lambda}{\Lambda} < 1, \gamma = \frac{2}{2-a-b}, \rho_1 = 2, \rho_k = \frac{1}{1-\alpha\rho_{k-1}}(k \geqslant 2), \alpha = (\frac{b-a}{2(2-a-b)})^2$

Conjugate Gradient Method

$$r^{(0)} = b - Ax^{(0)}, p^{(1)} = r^{(0)}, \alpha^{(m)} = \frac{(r^{(m)}, r^{(m)})}{(Ap^{(m)}, p^{(m)})}$$

$$x^{(m+1)} = x^{(m)} + \alpha^{(m)}p^{(m)}$$

$$r^{(m+1)} = r^{(m)} - \alpha^{(m)}Ap^{(m)}, \beta^{(m)} = \frac{(r^{(m+1)}, r^{(m+1)})}{(r^{(m)}, r^{(m)})}, p^{(m+1)} = r^{(m+1)} + \beta^{(m+1)}p^{(m)}$$

Dimension	Jump	2-term Chebyshev	3-term Chebyshev	CG	Jacobi	SOR
n=25	1	133	119	25	1485	69
	2	153	136	25	1484	67
	10	245	211	32	1504	66
	100	1017	588	38	1512	67
n=50	1	277	231	50	5192	130
	2	345	268	56	5257	128
	10	604	396	71	5431	129
	100	3484	1133	106	5497	130
n=100	1	696	457	100	18253	254
	2	956	533	118	18370	253
	10	1948	811	168	18860	253
	100	13340	2213	281	19058	252

Table 1: Number of iteration steps for different methods of problem 2

Dimension	Jump	K=32	K=48	K=64
n=25	1	133	136	243
	2	153	159	3251
	10	245	212	100000
	100	1017	852	100000
n=50	1	277	258	100000
	2	345	306	100000
	10	604	472	100000
	100	3484	2580	100000
n=100	1	696	554	100000
	2	956	712	100000
	10	1948	1336	100000
	100	13340	7864	100000

Table 2: Comparation of 2-term Chebyshev when K is different. (100,000 is the largest iteration steps we can accept. This means it diverges when K=64)

We also plot the solutions of this system for different jump gotten by CG when $n = 25$ in Figure 1. Other results using different methods and for different dimensions are basically similar.

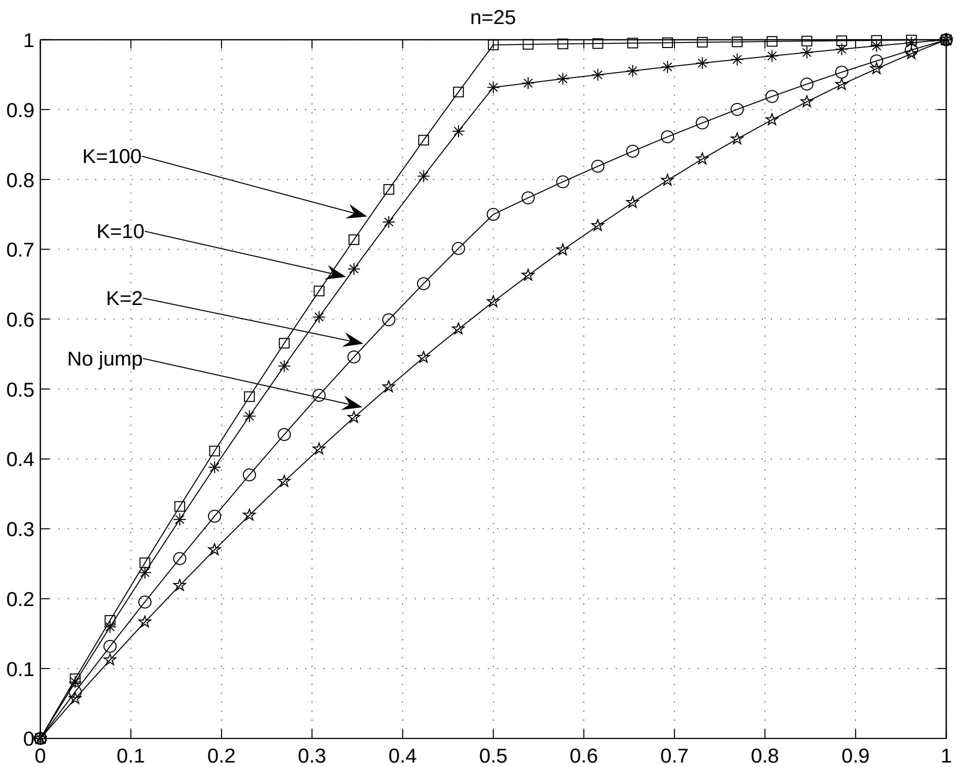

Figure 1: Solutions of different jumps when n=25 using CG

Problem 2 (Two dimensional Array)

In this problem, we are asked to solve a two dimensional array $x_{ij}, i, j = 0, \ldots, n+1$ that satisfies the system

$$(4 + h^2)x_{i,j} - x_{i-1,j} - x_{i+1,j} - x_{i,j-1} - x_{i,j+1} = h^2, x_{0,j} = x_{n+1,j} = x_{i,0} = x_{i,n+1} = 0.$$

$h = 1/(n+1)$ for $n = 8, 16, 32$. The system represents a finite difference approximation of the boundary value problem $-\Delta u + u = 1$ in $\Omega = (0,1) \times (0,1)$ and $u = 0$ on the boundary of Ω. Once we solve the linear system, we get the solution for this problem.

Here we still use Chebyshev and CG method for different n and the same stopping condition as last problem. The results are list in Table 3. It is clear from the table that we still get the same answer in comparing the behavior of these three methods, namely, CG is better than three-term Chebyshev, which is better than two-term one. But this time, we found that the differences among the numbers of iterative steps are not as large as last problem. This shows that the behavior of different methods also has relation with the property of coefficient matrix.

Again since the solution $x_{i,j}$ of this system represents an approximation of the solution $u(ih, jh)$ of a boundary value problem in Ω, it makes sense to plot $x_{i,j}$. The following graphs are plotted for $n = 8, 16, 32$.

Dimension	2-term Chebyshev	3-term Chebyshev	CG	Jacobi	SOR
n=8	60	45	10	210	44
n=16	101	80	25	758	47
n=32	200	139	51	1756	189

Table 3: Number of iteration steps for different methods of problem 3

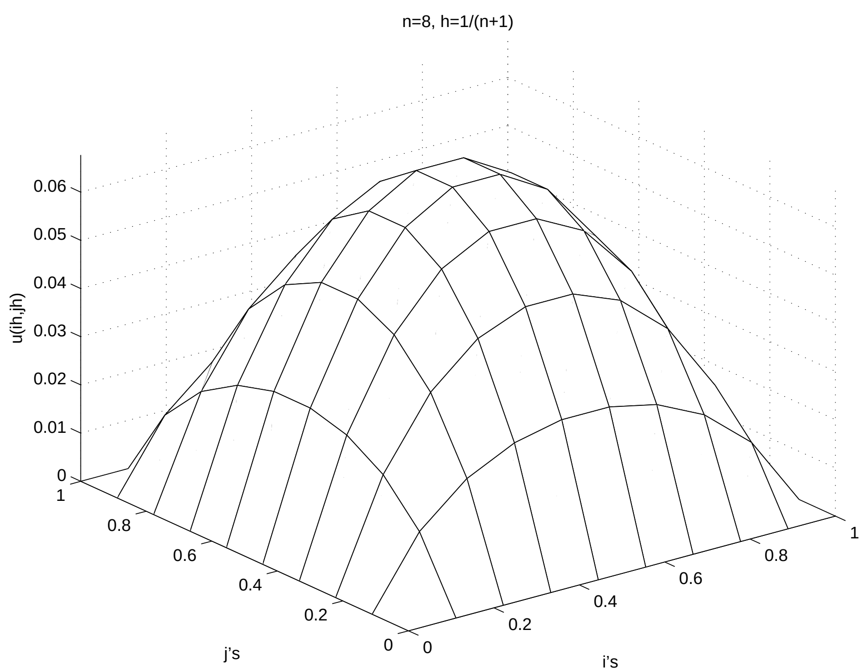

Figure 2: $n = 8$

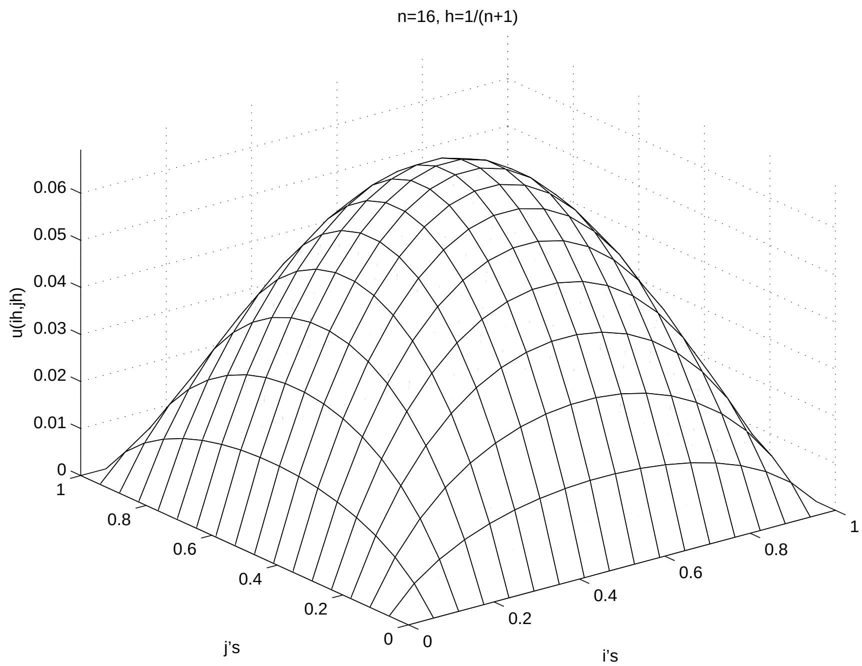

Figure 3: $n = 16$

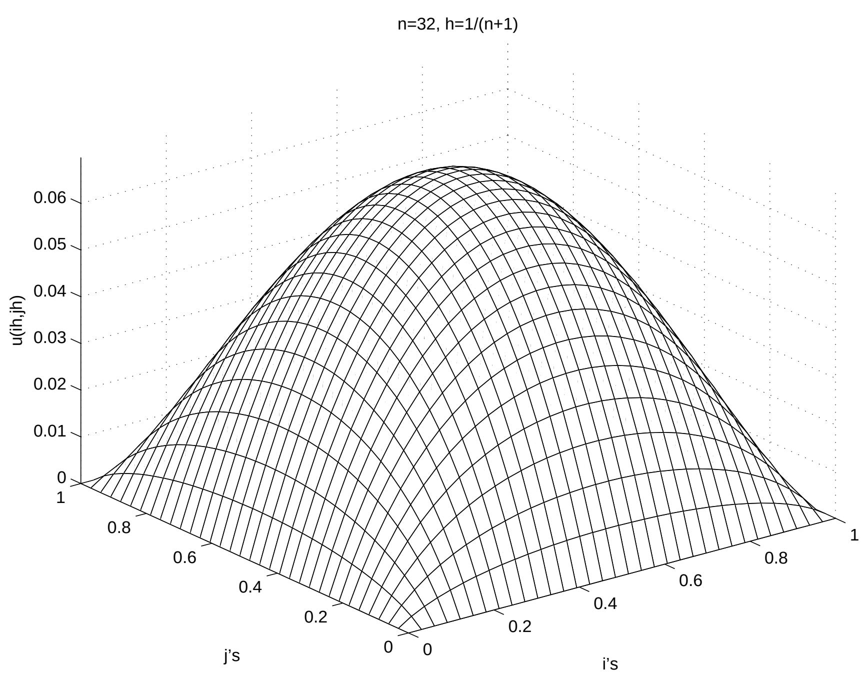

Figure 4: $n = 32$

Summary

In this programming assignment, we want to solve the systems $Ax = b$ by using (1) two-term and three-term Chebyshev acceleration iterative method and (2) Conjugate Gradient Method, which are also iterative methods for solving linear equations, just like Gauss-Seidel, SOR and SSOR.

Both Chebyshev and CG methods are trying to use more information from the solutions gotten from former iterations to improve the result of new step. Here more information means a linear combination of known iterative solutions. The problems are using which solutions and what the weights of these solutions. Two-term Cheybeshev and CG just use solution obtained from last iteration with some special coefficients, while three-term Cheybeshev uses solutions obtained from last two iteration steps.

There is difference between Chebyshev and CG methods. In Chebyshev iteration the iteration parameters are know as soon as we know the field of eigenvalues of iteration matrix (which satisfies some property), but CG method needs compute inner product three times (we can make it to be twice by rearranging). Then Chebyshev iteration is good for solving a large sparse linear systerm of equations in a parallel environments.

We use Richarson-like matrix as the iterative matrix here. From the results we can find that, the number of iteration steps is smallest when using CG method, three-term Chebyshev needs more steps and two-term's behavior dependends a lot to the choosing of iteration parameters. When dimension is large, CG method do take more time to compute. Though the differences are not so obvious, the trend tells that if the matrix becomes larger and sparse, CG do need more time. Then by comparing the results with Jacobi and SOR, we can find CG is even better than SOR in number of iteration steps , and Jacobi always need the largest number of steps.