Foundations of Computational Math I Exam 2
Take-home Exam
Open Notes, Textbook, Homework Solutions Only
Calculators/Computers Allowed
No collaborations with anyone
Due beginning of Class Monday, December 5, 2011

Question	Points Possible	Points Awarded
1. Linear Systems	30	
2. Linear Systems	30	
3. Nonlinear Equations	40	
4. Nonlinear Equations	30	
Total Points	130	

Name:________________________________

Alias: ________________________________

to be used when posting anonymous grade list.

Problem 1

(30 points)

Consider the system of equations

$$Ax = b$$

where A is a nonsingular lower triangular matrix, i.e.,

$$A = D - L$$

where D is diagonal and nonsingular, and L is strictly lower triangular.

(1.a) Show that Forward Gauss-Seidel will converge to $x = A^{-1}b$ in a finite number of steps (in exact arithmetic) for any in initial guess x_0 and give a tight upper bound on the number of steps required.

(1.b) What is the complexity of solving the system using Forward Gauss-Seidel: $O(n)$ or $O(n^2)$ or $O(n^3)$ or more?

Solution:

Let G_F be the iteration matrix for forward Gauss-Seidel applied to the problem. We have

$$A = D - L - U \quad \text{where} \quad U = 0$$
$$x_{k+1} = G_F x_k + f$$
$$G_F = M^{-1}N = (D - L)^{-1}U = 0$$
$$f = M^{-1}b = (D - L)^{-1}b = A^{-1}b = x$$

$$\therefore x_1 = A^{-1}b = x$$

Forward Gauss-Seidel converges in one step. The complexity is a single triangular solve or $O(n^2)$.

(1.c) Show that Backward Gauss-Seidel will converge to $x = A^{-1}b$ in a finite number of steps (in exact arithmetic) for any in initial guess x_0 and give a tight upper bound on the number of steps required or show that it does not converge in a finite number of steps for all x_0.

(1.d) What is the complexity of solving the system using Backward Gauss-Seidel: $O(n)$ or $O(n^2)$ or $O(n^3)$ or more?

(1.e) What is the relationship between Backward Gauss-Seidel and Jacobi iterations for solving this system?

Solution:

Let G_B be the iteration matrix for forward Gauss-Seidel applied to the problem. We have

$$
\begin{aligned}
A = D - L - U \quad &\text{where} \quad U = 0 \\
x_{k+1} &= G_B x_k + f \\
G_B &= M^{-1}N = (D)^{-1}L \\
f &= M^{-1}b = (D)^{-1}b
\end{aligned}
$$

Now note that $G_B = (D)^{-1}L$ is a strictly lower triangular matrix. This implies that all of the eigenvalues of G_B are 0 and depending on the structure of nonzero elements there may be as few as 1 eigenvector. G_B is a nilpotent matrix, i.e., it becomes 0 after a finite number of multiplications by itself. This is easily seen by considering a case of $n = 3$ where $*$ mark positions that may be nonzero:

$$
G_B = \begin{pmatrix} 0 & 0 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{pmatrix}, \quad G_B^2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ * & 0 & 0 \end{pmatrix}, \quad G_B^3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}
$$

In general, if a matrix, Z, is strictly lower triangular then $Z^k = 0$ for some $k \leq n$. Another way to see this is that repeated multiplication of a vector by G_B introduces a new leading 0 in the vector, i.e., again for $n = 3$

$$
\begin{aligned}
G_B v &= \begin{pmatrix} 0 & 0 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{pmatrix} \begin{pmatrix} * \\ * \\ * \end{pmatrix} = \begin{pmatrix} 0 \\ * \\ * \end{pmatrix} = u \\
G_B u &= \begin{pmatrix} 0 & 0 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{pmatrix} \begin{pmatrix} 0 \\ * \\ * \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ * \end{pmatrix} = w \\
G_B w &= \begin{pmatrix} 0 & 0 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ * \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = 0
\end{aligned}
$$

Since the error satisfies $e^{(n)} = G_B^n e^{(0)}$ for any $e^{(0)}$ we know that Backward Gauss-Seidel converges in at most n steps for a lower triangular A. A triangular matrix vector product of decreasing dimension is required on each step so we have n steps with $O(n^2)$ computations each for a total computation of $O(n^3)$.

The Jacobi and backward Gauss-Seidel iterations are identical for this matrix.

Problem 2

(30 points)

2.a

(10 points)

(2.a.i) Suppose you must solve a linear system $Ax = b$ and all you know about A is that it is nonsingular and well-conditioned. Suppose further you only have a iterative method that is applicable to linear systems where the coefficient matrix is symmetric positive definite. How would you solve $Ax = b$?

Solution:

The matrix $A^T A$ is symmetric positive definite if and only if A has full rank. Therefore solve

$$A^T A x = A^T b$$

using the available routine. In practice, when this is used for sparse A, the routine is modified to replace all matrix vector products with $A^T A$ with two sparse matrix vector products involving A and A^T in turn. This has data structure and/or coding implications.

(2.a.ii) When attempting to solve $Ax = b$ where A is known to be nonsingular via an iterative method, we have seen various theorems that give sufficient conditions on A to guarantee the convergence of various iterative methods. It is not always easy to verify these conditions for a given matrix A. Let P and Q be two permutation matrices. Rather than solving $Ax = b$ we could solve $(PAQ)(Q^T x) = Pb$ using an iterative method. Sometimes it is possible to examine A and choose P and/or Q so that it is easy to apply one of our sufficient condition theorems.

Can you choose P and/or Q so that the permuted system converges for one or both of Gauss-Seidel and Jacobi with

$$A = \begin{pmatrix} 3 & 7 & -1 \\ 7 & 4 & 1 \\ -1 & 1 & 2 \end{pmatrix}?$$

Solution: We have

$$P = I \quad Q = \begin{pmatrix} e_2 & e_1 & e_3 \end{pmatrix}$$
$$AQ = \begin{pmatrix} 7 & 3 & -1 \\ 4 & 7 & 1 \\ 1 & -1 & 2 \end{pmatrix}.$$

PAQ is irreducibly diagonally dominant therefore both methods converge. The system with A does not converge for Jacobi and does not coverge for Gauss-Seidel.

2.b

(10 points)
Let

$$T = \begin{pmatrix} 4 & -1 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 \\ 0 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & -1 & 4 \end{pmatrix}$$

and consider solving a linear system $Tx = b$ with accelerated stationary Richardson's method

$$x_{k+1} = x_k + \alpha r_k$$

(2.b.i) Can α be set so that the iteration will converge for all x_0? Justify your answer.

(2.b.ii) If values of α exist for which the iteration conveges for all x_0 how would you set its value so the convergence rate was optimal or nearly so? Give specific values in your response. If no such α exists indicate what method you would use to solve the system.

Solution:

Stationary accelerated Richardson's method requires a definite matrix, i.e., the real parts of the eigenvalues must be all positive or all negative. T is symmetric so it has a set of orthonormal eigenvectors and real eigenvalues. applying Gershgorin's theorems yields disks centered at 4 with radii 2 for four disks and 1 for two disks. These are all in the positive half-plane so T is symmetric positive definite. We also have

$$2 < \lambda_i < 6$$

The strict inequalities are due to Gershgorin's third theorem.

For this particular class of matrix, tridiagonal and Toeplitz, we have the eigenvalues and eigenvectors analytically (recall Homework 6). The eigenvalues are

$$\lambda_j = \gamma - 2\cos(\frac{j\pi}{n+1})$$

for the matrix with subdiagonal, main diagonal and superdiagonal with values -1, γ and -1 respectively. So we have

$$\lambda_{min} = 4 - 2\cos(\frac{\pi}{7}) \approx 2.1981$$
$$\lambda_{max} = 4 - 2\cos(\frac{6\pi}{7}) \approx 5.8019$$

and

$$\alpha < \frac{2}{\lambda_{max}} \approx 0.3447$$

gives convergence.

Since for this class of matrix,

$$\lambda_{min} = \gamma - 2\cos(\frac{\pi}{n+1})$$
$$\lambda_{max} = \gamma - 2\cos(\frac{n\pi}{n+1})$$

and $\cos(\frac{\pi}{n+1}) = -\cos(\frac{n\pi}{n+1})$ we have

$$\alpha_{opt} = \frac{2}{\lambda_{max} + \lambda_{min}} = \frac{1}{\gamma}$$

independently of n and therefore we have $\alpha_{opt} = 0.25$.

Of course, the lower and upper bounds due to Gershgorin can be used to set the bound on α and an estimate of α_{opt}. We have

$$2 < \lambda_{min} < \lambda_{max} < 6$$
$$\alpha < \frac{1}{3}$$
$$\alpha_{opt} \approx \frac{2}{2+6} = 0.25$$

Note that it is only by chance that the estimate and the actual value of α_{opt} are identical.

2.c

(10 points)
Let

$$T = \begin{pmatrix} 4 & -1 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 \\ 0 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & -1 & -4 & -1 \\ 0 & 0 & 0 & 0 & -1 & -4 \end{pmatrix}$$

and consider solving a linear system $Tx = b$ with accelerated stationary Richardson's method

$$x_{k+1} = x_k + \alpha r_k$$

(2.c.i) Can α be set so that the iteration will converge for all x_0? Justify your answer.

(2.c.ii) If values of α exist for which the iteration conveges for all x_0 how would you set its value so the convergence rate was optimal or nearly so? Give specific values in your response. If no such α exists indicate what method you would use to solve the system.

Solution:
The modification to the diagonal elements in positions $5,5$ and $6,6$ make a crucial difference in the applicability of the iterative method. Gershgorin yields two disks in the left half-plane disjoint from the four disks in the right half-plane. So while the matrix has all real eigenvalues due to symmetry, it is indefinite and therefore stationary accelerated Richardson's method does not converge for any $\alpha > 0$. The matrix is still strictly diagonally dominant however so Jacobi and Gauss-Seidel (and others) converge.

Problem 3

(40 points)

3.a

(30 points)

Let $f(x) = x + \ln x$ on $x > 0$ and define

$$\phi_1(x) = -\ln x$$

$$\phi_2(x) = e^{-x}$$

$$\phi_3(x) = \frac{\alpha x + e^{-x}}{\alpha + 1} \qquad \alpha \neq -1$$

(3.a.i) Which of iteration functions $\phi_1(x)$ and $\phi_2(x)$ would you recommend to solve $f(x) = 0$ on $x > 0$? Justify your answer.

Solution:

It is easy to bound the root, x^*, by noting that $f(x)$ is monotonically increasing on $x > 0$ and

$$f(0.5) = -0.1931\ldots \qquad f(1) = 1$$

We have

$$\phi_1(x) = -\ln x \rightarrow \phi_1'(x) = -\frac{1}{x}$$

$$\phi_2(x) = e^{-x} \rightarrow \phi_2'(x) = -e^{-x}$$

$$\phi_3(x) = \frac{\alpha x + e^{-x}}{\alpha + 1} \rightarrow \phi_3'(x) = \frac{\alpha - e^{-x}}{\alpha + 1}$$

Clearly, on $0.5 < x < 1$ we have $\phi_1'(x) \geq 1$ and $|\phi_2'(x)| < 1$. Since x^* is a fixed point of ϕ_1 and ϕ_2, $x_{k+1} = \phi_2(x_k)$ will converge and $x_{k+1} = \phi_1(x_k)$ will not.

(3.a.ii) Suppose you have, β, an approximation to the root x_*, i.e., $\beta \approx x_*$. How would you set the parameter α in $\phi_3(x)$ so that the resulting iteration was better than your choice between $\phi_1(x)$ and $\phi_2(x)$?

Solution:

Given that

$$\phi_3'(x) = \frac{\alpha - e^{-x}}{\alpha + 1}$$

depends on α, the parameter should be chosen to make the value of the derivative as small as possible in the neighborhood of the root. If β is assumed to be near x^* then choosing α such that $\phi_3'(\beta) = 0$ means $\phi_3'(x^*)$ should be small.

(3.a.iii) Show that the closer β is to the root x_* the faster the resulting convergence of $\phi_3(x)$ using the associated α.

Solution:

Choosing $\alpha = e^{-\beta}$ yields a derivative such that

$$\phi_3'(x) = \frac{e^{-\beta} - e^{-x}}{e^{-\beta} + 1}$$
$$\phi_3'(\beta) = 0$$

So as $\beta \to x^*$ we have $\phi_3'(x^*) \to 0$ and the iteration improves its asymptotic convergence rate toward being quadratic, i.e., as fast as Newton's method.

(3.a.iv) Determine the iteration defined by Newton's method to solve $f(x) = 0$ and comment on its convergence compared to $\phi_3(x)$ using β and the associated α.

Solution:

Newton's method for this function is

$$f(x) = x + \ln x$$

$$f'(x) = 1 + \frac{1}{x}$$

$$\phi_N(x) = x - \frac{x + \ln x}{1 + x^{-1}} = \frac{x}{x+1}(1 - \ln x)$$

(3.a.v) Use Newton's method and one of the others to determine the root x^*. You need not turn in any code simply report the computed value of x^* and indicate something about the observed rate of convergence.

Solution:

Newton's method with $x_0 = 0.9$ and $x_0 = 0.5$ yields

k	x_k	$f(x_k)$
1	0.9000000000000	0.7946394843422
2	0.5235918232063	-0.1234510383510
3	0.5660165417316	-0.0031154338002
4	0.5671425752176	-0.0000019762362
5	0.5671432904095	-0.0000000000008

k	x_k	$f(x_k)$
1	0.5000000000000	-0.1931471805599
2	0.5643823935200	-0.0076408610091
3	0.5671389877151	-0.0000118893331
4	0.5671432903994	-0.0000000000288

Running ϕ_2 with $x_0 = 1.0$ converges much more slowly than Newton. Note it only reaches $f \approx 10^{-6}$ after 21 steps.

k	x_k	$f(x_k)$
1	1.0000000000000	1.0000000000000
2	0.3678794411714	-0.6321205588286
3	0.6922006275553	0.3243211863839
4	0.5004735005636	-0.1917271269917
5	0.6062435350856	0.1057700345220
6	0.5453957859750	-0.0608477491106
7	0.5796123355034	0.0342165495284
8	0.5601154613611	-0.0194968741423
9	0.5711431150802	0.0110276537191
10	0.5648793473910	-0.0062637676891
11	0.5684287250291	0.0035493776380
12	0.5664147331469	-0.0020139918822
13	0.5675566373283	0.0011419041814
14	0.5669089119215	-0.0006477254068
15	0.5672762321756	0.0003673202541
16	0.5670678983908	-0.0002083337848
17	0.5671860500994	0.0001181517086
18	0.5671190400572	-0.0000670100421
19	0.5671570440013	0.0000380039441
20	0.5671354902063	-0.0000215537950
21	0.5671477142601	0.0000122240538

Taking $\alpha = 0.5$ in ϕ_3 yields the Newton-like behavior predicted by the analyis.

k	x_k	$f(x_k)$
1	1.0000000000000	1.0000000000000
2	0.5785862941143	0.0314187193782
3	0.5666557366660	-0.0013475893625
4	0.5671651593305	0.0000604279575
5	0.5671423115993	-0.0000027046729
6	0.5671433342237	0.0000001210675

3.b

(10 points)

Discuss the main differences in convergence and complexity when solving a nonlinear equation using

(3.b.i) Newton's method

(3.b.ii) Secant method

(3.b.iii) Regula Falsi method

Solution:

- Newton's method:
  - local convergence, i.e., x_0 must be close enough to the root.
  - quadratic convergence
  - requires the availability and evaluation of $f'(x)$ on every step.
  - behavior can be erratic, i.e., very large steps moving far away from a root before eventually converging quickly.
- Secant method:
  - local convergence, i.e., x_0 must be close enough to the root.
  - superlinear convergence, $p \approx 1.6$.
  - does not require $f'(x)$
  - uses two values of x_k, i.e., is not a first-order recurrence.
  - like Newton, the behavior can be erratic, i.e., very large steps moving far away from a root before eventually converging quickly.
- Regula Falsi:
  - global convergence on an interval $[a, b]$ containing the root.
  - much more well-behaved than Newton or Secant.
  - linear convergence
  - does not require $f'(x)$
  - uses two values of x_k, i.e., is not a first-order recurrence.

Problem 4

(30 points)

Suppose you are asked to design an algortihm that could be used as an intrinsic library function to compute $\sqrt[3]{\alpha}$ for any $\alpha \in \mathbb{R}$ and $\alpha > 0$.

(4.a) Give an efficient algorithm in the form of $x = \phi(x)$ that solves the problem.
Solution:

Newton's method is often used for this purpose. We have

$$f(x) = x^3 - \alpha$$
$$\phi_N(x) = x = \frac{x^3 - \alpha}{3x^2} = \frac{2}{3}x + \frac{1}{3}\alpha x^{-2}$$

$$\phi_N'(x) = \frac{2}{3}(1 - \alpha x^{-3})$$

(4.b) Determine the restrictions, if any, on your choice of x_0. Also indicate how you would generate x_0 given α.

Solution:

Since $\alpha > 0$ we have $\sqrt[3]{\alpha} > 0$ and we concentrate on $x > 0$ for most of the analysis. The first part of the question requires determining sufficient conditions for convergence. The second part requires turning those conditions into a practical method of choosing x_0. So the simpler the answer to the first part the easier the second part becomes.

Note that we only know α and we are assuming $\alpha > 0$. An initial condition like $x_0 = \sqrt[3]{\alpha}$ does yield a trivially convergent iteration but it requires knowledge we do not have. Similar conditions such as $x_0 > \gamma\sqrt[3]{\alpha}$ for some constant γ is also unusable in that form.

Also note that the ordering of α and $\sqrt[3]{\alpha}$ depends on $\alpha > 1$ or $\alpha < 1$. While this is known we would like the initial condition choice to be simple, i.e., as few conditional statements as possible.

For $x > 0$ we can check when $|\phi_N'(x)| < 1$. We require

$$-1 < \frac{2}{3}(1 - \alpha x^{-3}) < 1$$
$$x > 0 \rightarrow \frac{2}{3}(1 - \alpha x^{-3}) < 1$$

$$-1 < \frac{2}{3}(1 - \alpha x^{-3}) \rightarrow x^3 > \frac{2}{5}\alpha$$
$$\therefore x > \sqrt[3]{\frac{2}{5}}\sqrt[3]{\alpha} \rightarrow |\phi_N'(x)| < 1.$$

Since clearly $\sqrt[3]{\alpha}$ is a fixed point we have convergence due to the contraction mapping property when

$$x > \sqrt[3]{\frac{2}{5}}\sqrt[3]{\alpha}$$

We can turn this into a selection of x_0 under the assumption that $\alpha > 1$ since we have

$$\alpha > 1 \rightarrow \alpha^3 > \alpha > \frac{2}{5}\alpha$$

$$\therefore \quad \alpha > \sqrt[3]{\frac{2}{5}} \;\; \sqrt[3]{\alpha}$$

So if $\alpha > 1$ then $x_0 > \alpha$ gives convergence.

However, we can extend the interval of convergence to avoid the conditional.

Consider $0 < x_0 \leq \sqrt[3]{\alpha}$, i.e., $x_0 = \gamma\sqrt[3]{\alpha}$ with $0 < \gamma \leq 1$. We have

$$x_1 = \phi_N(x_0) = \sqrt[3]{\alpha}\left[\frac{2}{3}\gamma + \frac{1}{3}\gamma^{-2}\right]$$

$$\gamma = 1 \rightarrow x_1 = \sqrt[3]{\alpha} \quad \text{as desired}$$

$$0 \leq \gamma < 1 \rightarrow 2\gamma^3 - 3\gamma^2 + 1 > 0 \rightarrow \left[\frac{2}{3}\gamma + \frac{1}{3}\gamma^{-2}\right] > 1$$

$$\therefore x_1 \geq \sqrt[3]{\alpha}$$

It follows that the iteration converges for any positive x_0, i.e., $x_0 > 0$. No conditional involving α is needed.

If $x_0 = 0$ then x_1 is undefined since the line defining Newton does not intersect the x axis.

We can look into $x_0 < 0$ to see if it is worthwhile practically. If $x_0 < 0$ and $|x_0| < \sqrt[3]{\alpha}/2$ then $x_1 > 0$ and convergence follows. This then makes our selection condition on x_0:

$$x_0 \neq 0 \quad \text{and} \quad x_0 > -\sqrt[3]{\alpha}/2$$

This adds a conditional and involves an unknown quantity. Also given that $x_1 > 0$ and we, in general, want to stay away from the vicinity of 0 to avoid a very large positive x_1, the previous $x_0 > 0$ still is the more practical condition.

If $x_0 = \sqrt[3]{\alpha}/2$ then $x_1 = 0$ and x_2 is undefined. One can clearly can continue this "backward" unrolling of the iteration to create a series of points, $x_i < 0$ that eventually lead to $x_i = 0$ and x_{i+1} undefined. We leave it as an exercies to determine the fate of other sequences with $x_0 < -\sqrt[3]{\alpha}/2 < 0$. Again we see $x_0 > 0$ still is the more practical condition.

Note that even if we put a practical bound away from 0, i.e.,

$$x_0 > \delta > 0$$

for some user selected δ, our analysis shows that we do not care if $\delta > \sqrt[3]{\alpha}$ or $\delta < \sqrt[3]{\alpha}$ convergence follows.

(4.c) Determine the order of convergence for your method and the number of computations needed to get single precision accuracy given the initial accuracy for x_0.

Solution:

Newton's method is quadratically convergent which in this case is easily seen since

$$\phi'_N(\sqrt[3]{\alpha}) = 0$$

We also know that the coefficient in the asymptotic error expression is

$$|e^{(k+1)}| \approx \frac{|\phi'_N(\sqrt[3]{\alpha})|}{3!}|e^{(k)}|^2$$

We have

$$\phi''_N(x) = 2\alpha x^{-4} \quad \text{and} \quad \therefore \quad \phi''_N(\sqrt[3]{\alpha}) = \frac{2}{\sqrt[3]{\alpha}}$$

Quadratic convergence implies asymptotically we double the number of digits in the solution on each step. So having an initial guess that has 1/4 or 1/8 of the digits correct yields a very small number of iterations and computations to get to the desired accuracy. Since only the mantissa matters (the exponent is easily determined), the initial guesses could for example be stored in a small table with a relatively small number of bits since for single precision we only need 4 bits accurate in x_0 to have 1/8 of the bits correct.

(4.d) Use your algorithm to determine $\sqrt[3]{69}$. Give the value of x_0 used and describe the observed convergence.

Solution:

Using $x_0 = 12$, i.e., no digits accurate, and Newton we have the following

k	x_k	$f(x_k)$
1	12.0000000000000	1659.0000000000000
2	8.1597222222222	474.2830098888674
3	5.7852583639321	124.6280521799437
4	4.5440375918776	24.8265497873118
5	4.1432527544448	2.1253281733716
6	4.1019839498592	0.0210990127092
7	4.1015659723000	0.0000021498412
8	4.1015659297023	0.0000000000000