University of Toronto
Faculty of Arts and Science

December 2018 Examinations

CSC 336 H1F — Numerical Methods

Duration — 3 hours

No Aids Allowed

Answer ALL Questions

Do **not** turn this page until you have received the signal to start.
In the meantime, please read carefully every reminder on this page.

- Write your answers in the exam booklets provided.
- Fill in your name and student number on each exam booklet that you use.
- Do not begin writing the actual exam until the announcements have ended and the Exam Facilitator has started the exam.
- As a student, you help create a fair and inclusive writing environment. If you possess an unauthorized aid during an exam, you may be charged with an academic offence.
- Turn off and place all cell phones, smart watches, electronic devices, and unauthorized study materials in your bag under your desk. If it is left in your pocket, it may be an academic offence.
- When you are done your exam, raise your hand for someone to come and collect your exam. Do not collect your bag and jacket before your exam is handed in.
- If you are feeling ill and unable to finish your exam, please bring it to the attention of an Exam Facilitator so it can be recorded before you leave the exam hall.
- In the event of a fire alarm, do not check your cell phone when escorted outside.
- This final examination consists of **5 questions** on **10 pages** (including this one), printed on both sides of the paper. When you receive the signal to start, please make sure that your copy of the examination is complete.
- The mark for each question is listed at the start of the question. Do the questions that you feel are easiest first.
- Remember that, in order to pass the course, you must achieve a grade of at least 35% on this final examination.
- The exam was written with the intention that you would have ample time to complete it. You will be rewarded for concise well-thought-out answers, rather than long rambling ones.
  We seek quality rather than quantity.
- Moreover, an answer that contains relevant and correct information as well as irrelevant or incorrect information will be awarded fewer marks than one that contains the same relevant and correct information only.

Write legibly. Unreadable answers are worthless.
Students must hand in all examination materials at the end of the exam.

1. [10 marks; 2 marks for each part]

For each of the five statements below, say whether the statement is **true** or **false** and briefly justify your answer.

(a) A good algorithm will produce an accurate solution to a problem regardless of the conditioning of the problem being solved.

(b) In the IEEE double-precision floating-point number system, *machine epsilon*, often referred to as ϵ_{mach} in your textbook, is the smallest positive floating-point number. That is, there are no double-precision floating-point numbers between ϵ_{mach} and zero.

(c) A well-conditioned matrix can have a very small determinant. That is, an $n \times n$ matrix A can have $\text{cond}(A)$ not too large (for example, $1 \leq \text{cond}(A) \leq 10$), but $\det(A)$ very close to 0 (i.e., $0 < \det(A) \ll 1$).

(d) If an iterative method for solving a nonlinear equation gains more than one bit of accuracy per iteration, then it is said to have a superlinear rate of convergence.

(e) Suppose you are given N data points, $(t_1, y_1), (t_2, y_2), \ldots, (t_N, y_N)$, where

- N is a positive integer,
- each $t_n \in \mathbb{R}$ and each $y_n \in \mathbb{R}$, for $n = 1, 2, \ldots, N$, and
- $t_1 < t_2 < \cdots < t_N$.

Then there are infinitely many polynomials of degree N that interpolate the data points $(t_1, y_1), (t_2, y_2), \ldots (t_N, y_N)$.

2. [10 marks: 5 marks for each part]

Walter was having trouble debugging his program. He traced the problem to a certain section of his code, but what he was computing there was fairly complex. So, he decided to try a simpler example of what he thought might be wrong with his code to see if that might help him determine the problem.

Walter knew that, if

$$f(x) = \frac{\mathrm{e}^x - 1}{x}$$

then

$$\lim_{x \to 0} f(x) = 1 \tag{1}$$

(For this question, just accept (1) as being true: you don't have to prove it.)

So, he expected that, if he computed $f(x)$ for smaller and smaller positive values of x, the computed values of $f(x)$ would get closer and closer to 1. He decided to test this conjecture, since he knew that odd things often happen in floating-point computation. So, he wrote a little MatLab program that computes $f(x)$ in IEEE double-precision floating-point arithmetic for $x = 10^{-k}$ and $k = 1, 2, \ldots, 15$. To his surprise, he got the results shown in the third column of the table on page 4.

He showed his results to his colleague, Irene, who suggested that he try computing instead

$$g(x) = \frac{\mathrm{e}^x - 1}{\ln(\mathrm{e}^x)}$$

where ln is the natural logarithm (also referred to as the logarithm to the base e (i.e., $\log_{\mathrm{e}}$)).

Walter thought that this was a ridiculous suggestion, since $\ln(\mathrm{e}^x) = x$, whence $f(x) = g(x)$ for all $x \in \mathbb{R}$ (assuming you define $f(0) = g(0) = 1$). Nevertheless, he tried Irene's suggestion and, to his surprise, he got the results shown in the fourth column of the table on page 4.

k	x	$f(x)$	$g(x)$
1	10^{-1}	1.051709180756477	1.051709180756476
2	10^{-2}	1.005016708416795	1.005016708416806
3	10^{-3}	1.000500166708385	1.000500166708342
4	10^{-4}	1.000050001667141	1.000050001666708
5	10^{-5}	1.000005000006965	1.000005000016667
6	10^{-6}	1.000000499962184	1.000000500000167
7	10^{-7}	1.000000049433680	1.000000050000002
8	10^{-8}	0.999999993922529	1.000000005000000
9	10^{-9}	1.000000082740371	1.000000000500000
10	10^{-10}	1.000000082740371	1.000000000050000
11	10^{-11}	1.000000082740371	1.000000000005000
12	10^{-12}	1.000088900582341	1.000000000000500
13	10^{-13}	0.999200722162641	1.000000000000050
14	10^{-14}	0.999200722162641	1.000000000000005
15	10^{-15}	1.110223024625157	1.000000000000000

(a) Explain why, when $f(x)$ is computed in IEEE double-precision floating-point arithmetic, the computed values first appear to be converging to 1 for $k = 1, 2, \ldots, 8$, but then diverge from 1 for $k = 11, 12, \ldots, 15$.

(b) Explain why, when $g(x)$ is computed in IEEE double-precision floating-point arithmetic, the computed values appear to be converging to 1 for $k = 1, 2, \ldots, 15$. In particular, explain why the computed values for $g(x)$ are so much more accurate than the computed values for $f(x)$ for $k = 11, 12, \ldots, 15$.

In answering the questions above, you can assume that the MatLab function $\exp(x)$ computes an accurate approximation to e^x. In particular, you can assume that

$$\exp(x) = \mathrm{e}^x(1 + \delta_x)$$

where δ_x changes with x, but its magnitude is at most a few multiples of ϵ_{mach}. (I.e., $|\delta_x| \leq c\,\epsilon_{\text{mach}}$ for some c that is at most 2 or 3.)

You can make some other reasonable assumption about the accuracy of the MatLab ln function, but remember that $\ln(u)$ is ill-conditioned for u near 1. If you make an assumption about the accuracy of the MatLab ln function, be sure to state what your assumption is.

3. [15 marks: 2 marks for each of parts (a) and (c); 3 marks for each of parts (b) and (e); 5 marks for part (d)]

We didn't have time this year to discuss in class the Cholesky factorization, or the closely related LDL factorization, of a *real symmetric positive-definite* matrix. We consider these two factorizations briefly in this question.

To begin, recall that A is an $n \times n$ real matrix if $A_{i,j} \in \mathbb{R}$ for all $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$, where $A_{i,j}$ is the (i, j)-element of the matrix A (i.e., $A_{i,j}$ is the value in row i and column j of the matrix A).

An $n \times n$ real matrix A is *symmetric positive-definite* if

- A is symmetric (i.e., $A^T = A$), and
- $x^T A x > 0$ for all $x \in \mathbb{R}^n$ such that $x \neq \vec{0}$, where $\vec{0}$ is the vector in $\mathbb{R}^n$ with $\vec{0}_i = 0$ for all $i = 1, 2, \ldots, n$ (i.e., all components for the vector $\vec{0}$ are zero).

The Cholesky factorization of an $n \times n$ real symmetric positive-definite matrix A is

$$A = \hat{L}\hat{L}^T \tag{2}$$

where $\hat{L}$ is an $n \times n$ lower triangular matrix. The LDL factorization of an $n \times n$ real symmetric positive-definite matrix A is

$$A = LDL^T \tag{3}$$

where L is an $n \times n$ lower triangular matrix with 1's on its diagonal (like the L in the LU factorization) and D is an $n \times n$ diagonal matrix (i.e., $D_{i,j} = 0$ for $i \neq j$) with $D_{i,i} > 0$ for $i = 1, 2, \ldots, n$. Note that $\hat{L}$ does not usually have 1's on its diagonal.

You don't have to show this here, but, if you have a Cholesky factorization (2) of A you can easily compute that LDL factorization (3) of A from (2) and vice-versa. So, (2) and (3) are almost equivalent factorizations of A.

The advantages of the Cholesky factorization (2) of A over the usual LU factorization of A are

- the Cholesky factorization (2) of A requires you to store about half as many values as the LU factorization of A,
- the Cholesky factorization (2) of A requires about half as many arithmetic operations to compute as the LU factorization of A, and
- you don't have to pivot for stability when computing the Cholesky factorization (2).

The LDL factorization (3) of A shares the same advantages as the Cholesky factorization (2) of A.

In this question, we focus on computing the LDL factorization (3) of A.

(a) Show that, if A is an $n \times n$ real symmetric positive-definite matrix, then $A_{i,i} > 0$ for all $i = 1, 2, \ldots, n$.

Hint: for each $i = 1, 2, \ldots, n$, choose a particular $\hat{x} \in \mathbb{R}^n$ for which $\hat{x} \neq \vec{0}$ and $A_{i,i} = \hat{x}^T A \hat{x}$. Then note that $\hat{x}^T A \hat{x} > 0$, since $\hat{x} \neq \vec{0}$ and A is an $n \times n$ real symmetric positive-definite matrix.

What is the required vector $\hat{x}$?

Even if you were not able to prove $A_{i,i} > 0$ for all $i = 1, 2, \ldots, n$, assume that this is true for the remainder of this question.

From part (a) above, if follows that $A_{1,1} > 0$. Therefore, you can compute that multipliers

$$m_{i,1} = A_{i,1}/A_{1,1} \qquad \text{for } i = 2, \ldots, n$$

and form the vectors

$$m_1 = \begin{pmatrix} 0 \\ m_{2,1} \\ m_{3,1,} \\ \vdots \\ m_{n,1} \end{pmatrix} \qquad e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

and the matrix

$$M_1 = I - m_1 \, e_1^T$$

where I is the $n \times n$ identity matrix.

(b) Show that

$$A_1 = M_1 A M_1^T = \begin{pmatrix} A_{1,1} & 0 & 0 & \cdots & 0 \\ 0 & \hat{A}_{2,2} & \hat{A}_{2,3} & \cdots & \hat{A}_{2,n} \\ 0 & \hat{A}_{3,2} & \hat{A}_{3,3} & \cdots & \hat{A}_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \hat{A}_{n,2} & \hat{A}_{n,3} & \cdots & \hat{A}_{n,n} \end{pmatrix} \tag{4}$$

where $A_{1,1}$ is the (1,1)-element of the original matrix A and the $\hat{A}_{i,j}$, for $i = 2, \ldots, n$ and $j = 2, \ldots, n$, are modified elements of A computed by multiplying A by M_1 on the left and by M_1^T on the right.

Even if you were not able to prove that A_1 has the structure shown in (4), assume that A_1 has this structure for the remainder of this question.

(c) Show that the matrix A_1 shown in (4) is an $n \times n$ real symmetric positive-definite matrix.

Even if you were not able to prove that A_1 is an $n \times n$ real symmetric positive-definite matrix, assume that this is the case for the remainder of this question.

It might at first appear that $2(n-1)^2$ adds and multiplications are required to compute A_1, since

- the operation $M_1 A$ essentially multiplies row 1 of A by $m_{i,1}$ and adds it to row i of A, for $i = 2, \ldots, n$, and
- $(M_1 A) M_1^T$ multiplies column 1 of $(M_1 A)$ by $m_{i,1}$ and adds it to column i of $(M_1 A)$, for $i = 2, \ldots, n$.

However, because of the symmetry of A and A_1, you can compute A_1 much faster than this.

(d) Show that you can compute A_1 with $\frac{1}{2}n(n-1)$ adds and multiplications and $n-1$ divisions.

Even if you were not able to prove that you can compute A_1 with $\frac{1}{2}n(n-1)$ adds and multiplication and $n-1$ divisions, assume that this is the case for the remainder of this question.

We have from part (c) that A_1 shown in (4) is an $n \times n$ real symmetric positive-definite matrix. Hence, by applying the result of part (a) to A_1, rather than A, we know that $\hat{A}_{i,i} > 0$ for all $i = 2, \ldots, n$.

Therefore, since $\hat{A}_{2,2} > 0$, you can compute that multipliers

$$m_{i,2} = \hat{A}_{i,2} / \hat{A}_{2,2} \qquad \text{for } i = 3, \ldots, n$$

and form the vectors

$$m_2 = \begin{pmatrix} 0 \\ 0 \\ m_{3,2} \\ \vdots \\ m_{n,2} \end{pmatrix} \qquad e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

and the matrix

$$M_2 = I - m_2 \, e_2^T$$

where I is the $n \times n$ identity matrix. Then, from an argument similar to the one used to prove part (b) above, it follows that

$$A_2 = M_2 A_1 M_2^T = \begin{pmatrix} A_{1,1} & 0 & 0 & 0 & \cdots & 0 \\ 0 & \hat{A}_{2,2} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \tilde{A}_{3,3} & \tilde{A}_{3,4} & \cdots & \tilde{A}_{3,n} \\ 0 & 0 & \tilde{A}_{4,3} & \tilde{A}_{4,4} & \cdots & \tilde{A}_{4,n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \tilde{A}_{n,3} & \tilde{A}_{n,4} & \cdots & \tilde{A}_{n,n} \end{pmatrix} \tag{5}$$

where $A_{1,1}$ is the (1,1)-element of the original matrix A, $\hat{A}_{2,2}$ is the (2,2)-element of the matrix A_1 in equation (4) and $\tilde{A}_{i,j}$, for $i = 3, \ldots, n$ and $j = 3, \ldots, n$, are modified elements of A_1 computed by multiplying A_1 by M_2 on the left and by M_2^T on the right.

Then, from an argument similar to the one used to prove part (d) above, it follows that you can compute A_2 with $\frac{1}{2}(n-1)(n-2)$ adds and multiplications and $n-2$ divisions.

Continuing in this way, we get that

$$M_{n-1}M_{n-2}\cdots M_2M_1AM_1^TM_2^T\cdots M_{n-2}^TM_{n-1}^T = D \tag{6}$$

where D is a diagonal matrix (i.e., $D_{i,j} = 0$ for $i \neq j$) with $D_{1,1} = A_{1,1} > 0$, $D_{2,2} = \hat{A}_{2,2} > 0$ and $D_{i,i} > 0$ for $i = 3, \ldots, n$. Also

$$M_k = I - m_k e_k^T$$

for $k = 1, 2, \ldots, n-1$, where m_k is the vector of multipliers used in the k^{th}-stage of the LDL factorization and e_k is the k^{th} column of the $n \times n$ identity matrix. To be more specific, the top k elements of m_k are zero and the bottom $n-k$ are the actual multipliers used in the k^{th}-stage of the LDL factorization.

Also, as noted above, the first stage of the LDL factorization requires $\frac{1}{2}n(n-1)$ adds and multiplications and $n-1$ divisions and the second stage of the LDL factorization requires $\frac{1}{2}(n-1)(n-2)$ adds and multiplications and $n-2$ divisions. By a similar argument, it follows that the k^{th}-stage of the LDL factorization requires $\frac{1}{2}(n-k+1)(n-k)$ adds and multiplications and $n-k$ divisions. Hence, the total computation work required to compute the LDL factorization is

$$\begin{aligned}\sum_{k=1}^{n-1}\frac{1}{2}(n-k+1)(n-k) &= \frac{1}{2}\left(\sum_{k=1}^{n-1}(n-k)^2 + \sum_{k=1}^{n-1}(n-k)\right)\\ &= \frac{1}{2}\left(\frac{n(2n-1)(n-1)}{6} + \frac{n(n-1)}{2}\right)\\ &= \frac{n^3}{6} + \mathcal{O}(n^2)\end{aligned}$$

adds and multiplications plus

$$\sum_{k=1}^{n-1}(n-k) = \frac{n(n-1)}{2} = \frac{n^2}{2} + \mathcal{O}(n)$$

divisions. Note that this is about half the computational work required by the LU factorization that we discussed in class.

(e) Show that you can rewrite (6) as (3).
Can you determine the L needed in (3) without any additional arithmetic work? Justify your answer.

4. [15 marks: 5 marks for each part]

It is well-known that, in many cases, Newton's method converges quadratically if you start close enough to a root, but that it may not converge at all, if you start too far from a root. However, there are some cases for which Newton's method always converges. We consider one such case below.

Assume $f''(x)$ exists and is continuous for all $x \in \mathbb{R}$ and that $f''(x) > 0$ for all $x \in \mathbb{R}$. Hence, $f(x)$ is a convex function. Assume also that there is a point $\hat{x} \in \mathbb{R}$ for which $f'(\hat{x}) = 0$ and $f(\hat{x}) < 0$.

$f(x) = x^2 - 1$ is an example of such a function; $\hat{x} = 0$ in this case.

(a) Show that there is a unique point $x^* > \hat{x}$ for which $f(x^*) = 0$.
Hint: to show that there is at least one point $x^* > \hat{x}$ for which $f(x^*) = 0$, you might find it helpful to first show that $f(x) \to \infty$ as $x \to \infty$.

(b) Show that, if $x_0 > \hat{x}$ and x_n, for $n = 1, 2, \ldots$, is generated by Newton's method

$$x_n = x_{n-1} - f(x_{n-1})/f'(x_{n-1}), \qquad \text{for } n = 1, 2, \ldots \tag{7}$$

then

- $x^* \leq x_n$ for $n = 1, 2, \ldots$, and
- $x_{n+1} \leq x_n$ for $n = 1, 2, \ldots$

That is, the x_n form a decreasing sequence that is bounded below by x^*.

(c) You can use without proof that a decreasing sequence that is bounded below must converge. That is, you can conclude from part (b) without proof that $x_n \to y^*$ as $n \to \infty$ and that $x^* \leq y^*$.
Show that $x^* = y^*$.

The results above show that, if you start Newton's method with an initial guess $x_0 > \hat{x}$, then the Newton iterates $x_n \to x^*$ as $n \to \infty$, where x^* is the unique root of $f(x)$ that is greater than $\hat{x}$.

Similarly, you can show that, if you start Newton's method with an initial guess $x_0 < \hat{x}$, then the Newton iterates $x_n \to x_*$ as $n \to \infty$, where x_* is the unique root of $f(x)$ that is less than $\hat{x}$. (You don't have to proof this result; I just stated it for completeness.)

The only small problem is that, if you start Newton's method with an initial guess $x_0 = \hat{x}$, then there is a divide-by-zero in Newton's method (7) for $n = 1$. However, if this divide-by-zero problem occurs, just choose another x_0 and the divide-by-zero problem cannot occur with this new initial guess x_0. Hence, Newton's method will converge to either x_*, if $x_0 > \hat{x}$, or to x_*, if $x_0 < \hat{x}$.

5. [10 marks: 5 marks for each part]

Suppose you are given the data

$$\begin{array}{lll} t_1 = -1 & t_2 = 0 & t_3 = 1 \\ y_1 = 1 & y_2 = 0 & y_3 = 1 \end{array}$$

and you want to find a polynomial $p(t)$ of degree 2 or less that satisfies

$$p(t_i) = y_i \qquad \text{for } i = 1, 2, 3.$$

(a) Use the monomial basis approach to find the polynomial $p(t)$ in the form

$$p(t) = c_1 + c_2 t + c_3 t^2 \tag{8}$$

What are the values of the coefficients c_1, c_2, c_3?

(b) Use the Lagrange basis approach to find the polynomial $p(t)$ in the form

$$p(t) = l_1(t)y_1 + l_2(t)y_2 + l_3(t)y_3 \tag{9}$$

where the $l_i(t)$, for $i = 1, 2, 3$, are the Lagrange basis functions.

Show that the polynomial $p(t)$ written in the monomial basis form (8) is the same as the polynomial $p(t)$ written in the Lagrange basis form (9).

Total Marks = 60

Total Pages = 10

Have a Happy Holiday