PLEASE HAND IN

PLEASE HAND IN

UNIVERSITY OF TORONTO
Faculty of Arts and Science

DECEMBER 2112 EXAMINATIONS

CSC 336 H1F
Instructor: Fairgrieve

Duration — 3 hours

Examination Aids: **One $8\frac{1}{2}'' \times 11''$ sheet of paper, handwritten on both sides.**
No photocopies.
A calculator.

You must earn at least 30% on this final examination in order to pass the course. Otherwise, your final course grade will be no higher than 47%.

Student Number: |_|_|_|_|_|_|_|_|_|_|

Family Name(s): ____________________

Given Name(s): ____________________

This is a collection of questions of the style that I like to ask. There are more questions here than are reasonable, so that you have more to practice with. Please do not panic at the length. Also, please do not over analyze the distribution of the questions - there are too many e.g Ch1 questions relative to the others. That does not indicate anything about your exam. Read this Mock Exam for style not distribution.

Answer each question directly on this paper, in the space provided. Blank pages can be found at the end of the exam and may be used for rough work or in case you need more space for solutions.

THIS EXAM IS DOUBLE-SIDED.

Question 1. [8 MARKS]

For each of the following questions, **circle** the response TRUE if you think that the given statement is true and FALSE if you think that it is false. One mark will be awarded for each correct response. To discourage random guessing, one-half mark may be **deducted** for each incorrect response.

Part (a) [1 MARK]

The length of the mantissa (or significand) of a number in the IEEE double-precision floating-point number system is double the length of the mantissa of a number in the IEEE single-precision floating-point number system.

TRUE FALSE

Part (b) [1 MARK]

Using higher-precision arithmetic will make an ill-conditioned problem better conditioned.

TRUE FALSE

Part (c) [1 MARK]

A stable algorithm applied to a well-conditioned problem necessarily produces an accurate solution.

TRUE FALSE

Part (d) [1 MARK]

For a symmetric matrix $\mathbf{A}$, it is always the case that $\|\mathbf{A}\|_1 = \|\mathbf{A}\|_\infty$. TRUE FALSE

Part (e) [1 MARK]

If $\mathbf{A}$ is any $n \times n$ matrix and $\mathbf{P}$ is any $n \times n$ permutation matrix, then $\mathbf{PA} = \mathbf{AP}$.

TRUE FALSE

Part (f) [1 MARK]

If a system of linear equations is well-conditioned, then pivoting is unnecessary when performing Gaussian elimination.

TRUE FALSE

Part (g) [1 MARK]

When a superlinearly convergent iterative method and a linearly convergent iterative method both compute a solution to a root-finding problem to within a given fixed error tolerance `TOL`, the superlinearly convergent method will always take fewer iterations.

TRUE FALSE

Part (h) [1 MARK]

The bisection method is an example of a fixed-point iteration scheme.

TRUE FALSE

Question 2. [5 MARKS]

Part (a) [2 MARKS]

Explain in words the difference between the machine epsilon, ϵ_{mach}, and the underflow threshold, **TINY**, in a floating-point number system. (Your textbook used the name **UFL** for the quantity we called **TINY**.)

The unit roundoff εmach is determined by the number of digits in the mantissa field of a floating-point system, whereas the underflow level UFL is determined by the number of digits in the exponent field.

Part (b) [1 MARK]

Which of ϵ_{mach} and **TINY** depends only on the number of significant digits in the elements in a floating-point number system? Explain.

ε mach

Part (c) [1 MARK]

Which of ϵ_{mach} and **TINY** depends only on the number of digits in the exponent part of the elements in a floating-point number system? Explain.

TINY

Part (d) [1 MARK]

Which of ϵ_{mach} and **TINY** does *not* depend on the rounding rule used? Explain.

ε = β^(1−t) chopping

ε = 1/2(β^(1−t))
rounding to nearest

TINY does not depend on rounding

Question 3. [16 MARKS]

In this question you will consider the problem of computing all solutions of the quadratic equation

$$ax^2 + bx + c = 0,$$

where a, b and c are real-valued coefficients. Recall that the solutions are given by the mathematical formula

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

a formula which is sometimes called the *quadratic formula*.

Part (a) [5 MARKS]

Using 4-digit, rounded, decimal arithmetic, compute accurate approximations to the roots of the quadratic equation $ax^2 + bx + c = 0$ for $a = 0.05010$, $b = -98.78$ and $c = 5.015$.

b^2 =

4ac =

b^2 - 4ac =

sqrt(b^2 - 4ac) =

Part (b) [3 MARKS]

Show that your solution in part (a) is the true solution of a quadratic equation that is close to the given quadratic equation.

x1, x2 => (x- x1)(x-x2) = 0 => x^2 –(x1+x2) +x1x2 = 0

Part (c) [8 MARKS]

Searching the web uncovered a **matlab** function with prologue:

```
function [roots] = quadraticSolver( a, b, c )
%
% quadraticSolver: Computes all roots of the quadratic equation a*x*x + b*x + c = 0.
%
% Input:   a,b,c = the coefficients of the quadratic equation.
% Output:  roots = a two-component vector that contains approximations to
%                  the two roots of the equation.
```

You remember that the straightforward evaluation of the quadratic formula is not a stable algorithm for computing roots and therefore you decided to test the `quadraticSolver` function before using it your programs. You want to be sure that `quadraticSolver` does not simply use the quadratic formula to compute the roots.

Describe test cases that you would apply to judge the accuracy of the roots computed by the `quadraticSolver` function. You do not need to give precise values for the coefficients a, b and c, but you should describe their values and also the potential numerical instability in the quadratic formula that your test case is designed to test.

	Description of a, b, c values	The potential numerical instability this case tests.
test case 1	*b^2 -4ac really small*	*C.C.*
test case 2		
test case 3		
test case 4		

Question 4. [5 MARKS]

Consider the problem of evaluating the function $f(x) = \sqrt{x+1} - \sqrt{x}$, for $x > 0$.

Part (a) [2 MARKS]

c.c

Is the algorithm `f = sqrt(x+1) - sqrt(x)` a numerically stable algorithm for evaluating $f(x)$? Answer YES or NO and justify your response.

Stable algorithm

[sqrt(x+1) – sqrt(x)] • [sqrt(x+1) + sqrt(x)]/ [sqrt(x+1) + sqrt(x)]

=

Part (b) [3 MARKS]

Is evaluating $f(x)$ a well-conditioned problem? Answer YES or NO and justify your response.

Question 5. [10 MARKS]

Consider the IEEE double precision floating-point number system $F(2, 53, -1022, 1023)$.

Part (a) [2 MARKS]

2^(52) 2^(51)

How many floating-point numbers x satisfy $1 \leq x < 2$? How many of these satisfy $1 \leq x < 3/2$ and how many satisfy $3/2 \leq x < 2$?

Part (b) [2 MARKS]

2^52

How many floating-point numbers y satisfy $1/2 \leq y < 1$? Approximately how many of these satisfy $1/2 \leq y < 2/3$ and how many satisfy $2/3 \leq y < 1$?

Part (c) [2 MARKS]

Does it follow that there must exist two different floating-point numbers x_1 and x_2 between $3/2$ and 2 for which the computed reciprocals $1/x_1$ and $1/x_2$ are the same after rounding to the floating-point number system? Explain.

Part (d) [2 MARKS]

Does it also follow that there exist floating-point numbers x for which $1/(1/x)$ is not exactly x? Explain.

Part (e) [2 MARKS]

For what fraction of the floating point numbers x satisfying $3/2 \leq x < 2$ might you expect that the floating point computation $1/(1/x)$ is exactly x? (Be approximate)

Question 6. [12 MARKS]

Consider the equation $Ax = b$ with $A = \begin{bmatrix} 4 & 8 & 1 \\ 2 & 5 & 1 \\ 1 & 4 & 2 \end{bmatrix}$, $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ and $b = \begin{bmatrix} 6 \\ 3 \\ 3 \end{bmatrix}$.

Part (a) [5 MARKS]

Compute the LU factorization (**without** pivoting) of the matrix A. Please show all steps in your solution and place your final results in the appropriate boxes below.

$L = \begin{pmatrix} \\ \\ \\ \end{pmatrix}$

$U = \begin{pmatrix} \\ \\ \\ \end{pmatrix}$

Part (b) [3 MARKS]

Use your LU factorization to solve the problem $Ax = b$ for x. Please show all steps in your solution and place your answer in the box.

$LUx = b$
$Lp = b$
$Ux = p$

$x = \begin{pmatrix} \\ \\ \\ \end{pmatrix}$

Part (c) [3 MARKS]

What elements of L and U would change if row pivoting were used during the computation of the LU factorization of A?

Part (d) [1 MARK]

What elements of x would change if row pivoting were used during the computation of the LU factorization of A?

Question 7. [6 MARKS]

Given an $n \times n$ nonsingular matrix $\mathbf{A}$, a $1 \times n$ row-vector $\mathbf{c}$ and a $n \times 1$ column-vector $\mathbf{b}$, explain how you would efficiently compute the quantity

$$\alpha = \mathbf{c}\mathbf{A}^{-1}\mathbf{b}.$$

What is the operation count for your algorithm? Express your answer in terms of the number of *flops* required.

Question 8. [5 MARKS]

Under what circumstances does a small residual vector $\underline{\hat{r}} = \underline{b} - \mathbf{A}\underline{\hat{x}}$ imply that $\underline{\hat{x}}$ is an accurate solution to the linear system $\mathbf{A}\underline{x} = \underline{b}$? Explain.

A small relative residual is not necessarily a good indicator that a computed solution is close to the "true" solution unless the system is well-conditioned.

Question 9. [7 MARKS]

Newton's method is sometimes used to implement the built-in square root function on a computer. The initial guess can be taken from a table of approximate solutions.

Part (a) [4 MARKS]

What is the Newton's method iteration for computing the square root of a positive number y (i.e. for solving the equation $f(x) = x^2 - y = 0$ for x given y)? Simplify your expression.

Part (b) [3 MARKS]

If we assume that the starting guess has an accuracy of at least 4 bits, how many iterations would you expect to be necessary to obtain a result with:

1. 24 bits of accuracy? Explain.

2. 53 bits of accuracy? Explain.

Question 10. [8 MARKS]

On a computer with no functional unit for floating-point division, one might use multiplication by the reciprocal of the divisor. That is, to compute a/b, multiply a by the reciprocal of b, namely $1/b$. The difficulty then is finding an approximation to the reciprocal of b.

Part (a) [5 MARKS]

Use Newton's method to produce an iterative scheme for approximating the reciprocal of a number $y > 0$. Considering the intended application, your final formula should contain no divisions!

f(x) : 1/x -y = 0, where y the number whose reciprocal need to be calculated
f(x) : x –xy = 0
Newton iteration ...

Part (b) [3 MARKS]

For $y > 1$, with what initial guess would you recommend starting the iteration? Are there any initial guesses that you would recommend not using?

initial guess i, ,0 <i <1.
guesses that are bigger than 1 or smaller than 0 are not recommended

Question 11. [8 MARKS]

Part (a) [4 MARKS]

If the errors at successive iterations of an iterative method are as follows, how would you characterize the convergence rates? Explain your answer.

Errors: $10^{-2}, 10^{-4}, 10^{-8}, 10^{-16}, \ldots$

quadratic

Errors: $10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}, \ldots$

linear with C = 10^(-2)

Part (b) [4 MARKS]

The values of `x(i), i=0,1,2,3,...,` in the following table of computational results appear to be converging to `2.0`. Estimate the rate of convergence. Explain your answer.

```
 i   x(i)                     | x(i)-x(i-1) |
==   ====================     ===============
 0   2.00523654181316e+00
 1   2.00119088895221e+00     4.05e-03
 2   2.00021688122653e+00     9.74e-04
 3   2.00003059322488e+00     1.86e-04
 4   2.00000321691105e+00     2.74e-05
 5   2.00000024128199e+00     2.98e-06
 6   2.00000001227082e+00     2.29e-07
 7   2.00000000039919e+00     1.19e-08
 8   2.00000000000777e+00     3.91e-10
 9   2.00000000000008e+00     7.68e-12
```

Question 12. [4 marks]

Suppose that you have forgotten your calculator at home and you need a rough approximation to $\sin(\pi x)$ for several values of $x \in [0, 1/2]$. You do remember, however, that $\sin(0) = 0$, $\sin(\pi/6) = 1/2$, $\sin(\pi/4) = 1/\sqrt{2}$, $\sin(\pi/3) = \sqrt{3}/2$ and $\sin(\pi/2) = 1$. With this information, you can easily compute the polynomial $p_4(x)$ of degree 4 or less that interpolates $\sin(\pi x)$ at $x = 0$, $1/6$, $1/4$, $1/3$ and $1/2$. Of course, you might be concerned that replacing $\sin(\pi x)$ with $p_4(x)$ may introduce large errors in your later calculations.

Show that

$$\max_{0 \le x \le 1/2} |\sin(\pi x) - p_4(x)| \le 3.83 \times 10^{-4}.$$

Hint: you may use without proof that

$$\max_{0 \le x \le 1/2} |x(x-1/6)(x-1/4)(x-1/3)(x-1/2)| \le 1.5 \times 10^{-4}.$$

Question 13. [6 MARKS]

Determine the coefficients a, b, c and d so that the function

$$S(x) = \begin{cases} x^3 - 1, & -9 \le x \le 0, \\ ax^3 + bx^2 + cx + d, & 0 \le x \le 5 \end{cases}$$

is a cubic spline that takes the value 2 when $x = 1$.

Question 14. [5 MARKS]

Define

$$S(x) = \begin{cases} -5 + 8x - 6x^2 + 2x^3, & 1 \le x \le 2, \\ 27 - 40x + 18x^2 - 2x^3, & 2 \le x \le 3. \end{cases}$$

Part (a) [4 MARKS]

Verify that $S(x)$ is a cubic spline function on $[1, 3]$.

Part (b) [1 MARK]

Is $S(x)$ a **natural** cubic spline function on $[1, 3]$? Explain.

Question 15. [10 MARKS]

Consider the Lagrange polynomial interpolation of the n data points $\{(x_i, y_i)\}_{i=1}^{n}$.

Part (a) [2 MARKS]

What is the degree of each polynomial function $l_i(x)$ in the Lagrange basis? Explain.

n–1

Part (b) [2 MARKS]

List one advantage and one disadvantage of using the Lagrange basis instead of using the monomial basis for polynomial interpolation.

pros: easy to determine the interpolating
cons: more expensive to calculate for a given input.

Part (c) [6 MARKS]

What is the value of $\sum_{i=1}^{n} 1 \cdot l_i(x)$, the sum of the n Lagrange basis functions $l_i(x)$?

Question 16. [8 marks]

Consider the data set $\{(x_i, y_i)\}_{i=1}^{3} = \{(0,3),(1,1),(3,5)\}$ and the problem of determining a polynomial $p_2(x)$ of degree at most 2 that interpolates the data set.

Part (a) [4 marks]

Draw graphs of each of the Lagrange basis functions to show their essential properties.

Part (b) [4 marks]

Draw graphs of each of the Newton basis functions to show their essential properties.

Question 17. [12 MARKS]

Part (a) [2 MARKS]

What is the computational cost (number of additions and multiplications) of evaluating a polynomial of degree n using Horner's method?

Part (b) [4 MARKS]

When interpolating a continuous function $f(t)$ by a polynomial, what key factors determine the error in the approximation of the function by the interpolant?

interpolation points and f(x)

Under what circumstances can the error be large even though the number of interpolation points is large?

1. max length of interval could still be large
2. n-th derivative of f(x) could be large

Part (c) [2 MARKS]

How does a cubic spline interpolant differ from a piecewise cubic Hermite interpolant?

hermite has n degree of freedom
cubic has 2

Part (d) [4 MARKS]

The continuity and smoothness requirements on a cubic spline interpolant still leave two free parameters. Give two examples of additional constraints that might be imposed to determine the cubic spline interpolant to a set of data points. Give the usual name for each of the cubic splines that would be determined.

natural
not a knot
clamped

Question 18. [10 MARKS]

Part (a) [5 MARKS]

Given the three data points

t	-2	0	2
y	1	0	1

use a **Newton** basis to construct the polynomial $p_2(t)$ of degree at most 2 that interpolates the data. Show your calculations.

Part (b) [5 MARKS]

The given data was generated by evaluating the function $f(t) = \sin^2\left(\frac{\pi t}{4}\right)$ at the points $t = -2$, 0 and 2. Use the error bound formula for polynomial interpolation to derive an accurate (tight) upper bound for

$$\max_{t\in[-2,2]} |f(t) - p_2(t)|.$$

Give a numerical (decimal) value for your bound. Show your calculations. You may want to use the identity $\sin(2\theta) = 2\sin(\theta)\cos(\theta)$. And don't forget that $\frac{d\sin(u)}{du} = \cos(u)$ and $\frac{d\cos(u)}{du} = -\sin(u)$.

Question 19. [10 MARKS]

Part (a) [4 MARKS]

Consider the data points $\{(0, 1), (1, 1), (2, 5)\}$. Find the piecewise linear interpolant for the data.

Part (b) [6 MARKS]

The `matlab` program

```
x = linspace( 0, pi, n );
y = sin( x );
S = spline( x, y );
```

may be used to determine a not-a-knot cubic spline interpolant of the function $f(x) = \sin(x)$ over the interval $[0, \pi]$. For what values of `n` is the not-a-knot cubic spline guaranteed to have an absolute error that is less than 5.0×10^{-5} over the interval $[0, \pi]$? The variable `n` represents the number of knots (or breakpoints) in the spline.

HINT: The constant is $\dfrac{5}{384}$.

*[Use the space below for rough work. This page will **not** be marked, unless you clearly indicate the part of your work that you want us to mark.]*

Total Marks = 155