UNIVERSITY of TORONTO
FACULTY of APPLIED SCIENCE and ENGINEERING

FINAL EXAMINATION, APRIL 2001

SECOND YEAR — PROGRAM 9

CSC 282S — COMPUTATION & NUMERICAL METHODS

EXAM TYPE: **X**

EXAMINER — K. R. JACKSON

DURATION OF THE EXAM: 2.5 HOURS

Do NOT turn this page over until you are TOLD to start.

Aids Allowed:

1. All programmable and non-programmable electronic calculators.
2. Any books, notes or other printed or written material, without restriction.

Please fill-in **ALL** the information requested on the front cover of **EACH** exam booklet that you use.

The exam consists of 5 pages, including this one. Make sure you have all 5.

The exam consists of 8 questions. **Answer all 8**. The mark for each question is listed at the start of the question. Do the questions that you feel are easiest first.

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.

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) If two real numbers are each exactly representable as floating-point numbers, then their sum will also be exactly representable as a floating-point number.

(b) If you compute the sum

$$S = \sum_{i=0}^{n} \frac{x^i}{i!}$$

in MatLab for a large enough n, then S will always be a good approximation to e^x, provided that neither S nor e^x overflows nor underflows.

(c) Let $[a, b]$ be a given interval. Suppose that, for each $n = 1, 2, 3, \ldots$, you interpolate a continuous function $f(x)$ at n equally spaced points on the interval $[a, b]$ by a polynomial $p_n(x)$ of degree $n - 1$. Then

$$\max_{x \in [a,b]} |f(x) - p_n(x)| \to 0 \quad \text{as } n \to \infty.$$

(d) Let $[a, b]$ be a given interval. For each $n = 2, 3, 4, \ldots$,

$$\{x_{n,i} = a + i(b - a)/n : i = 0, 1, 2, \ldots, n\}$$

is a set of $n + 1$ equally spaced points on $[a, b]$. Let $L_n(x)$ be the piecewise linear spline with knots $\{x_{n,i} : i = 0, 1, 2, \ldots, n\}$ that satisfies $L_n(x_{n,i}) = f(x_{n,i})$ for $i = 0, 1, 2, \ldots, n$. If $f''(x)$ exists, is continuous and is bounded on $[a, b]$, then

$$\max_{x \in [a,b]} |f(x) - L_n(x)| \to 0 \quad \text{as } n \to \infty.$$

(e) Simpson's Quadrature Rule

$$Q_S(f; a, b) = \frac{b - a}{2} [f(a) + 4f(m) + f(b)] \quad \text{where } m = \frac{a + b}{2}$$

satisfies

$$Q_S(f; a, b) = \int_a^b f(x)\, dx$$

if $f(x)$ is a polynomial of degree 3 or less.

2. [10 marks]

Let $\omega_n = e^{-2\pi i/n}$, where $i = \sqrt{-1}$. Thus $\omega_n^n = 1$.

The $n \times n$ matrix F_n associated with the Fast Fourier Transform is given by

$$F_n = \left[f_{p,q}^{(n)} \right]$$

where $f_{p,q}^{(n)} = \omega_n^{(p-1)(q-1)}$ for $p = 1, \ldots, n$ and $q = 1, \ldots, n$. That is, for $p = 1, \ldots, n$ and $q = 1, \ldots, n$, element (p, q) of the matrix F_n is $\omega_n^{(p-1)(q-1)}$.

For example, for $n = 4$,

$$F_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & w_4 & w_4^2 & w_4^3 \\ 1 & w_4^2 & w_4^4 & w_4^6 \\ 1 & w_4^3 & w_4^6 & w_4^9 \end{bmatrix}$$

Write a MatLab script that

(a) asks the user to input a positive integer n,

(b) reads the integer n that the user inputs,

(c) prints an error message if n is not a positive integer,

(d) computes the matrix F_n if n is a positive integer.

Some marks for this question will be for efficiency and good programming practices. In particular, your script should use the vector facilities of MatLab as much as possible in computing F_n. That is, try to avoid loops as much you can.

3. [5 marks]

The Fibonacci numbers $\{f_n : n \geq 0\}$ are defined by

$$\begin{aligned} f_0 &= f_1 = 1 \\ f_n &= f_{n-1} + f_{n-2} \quad \text{for } n \geq 2. \end{aligned}$$

The first few Fibonacci numbers are 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Write a **recursive** MatLab function `fib(n)` that computes f_n for any integer $n \geq 0$.

Your function doesn't need to test that n is a non-negative integer.

[Using recursion as described above is not an efficient way to compute the Fibonacci numbers, but it does make for a simple problem to test your understanding of recursion.]

4. [10 marks; 5 marks for each part]

Billy wrote the MatLab function

```
function [r1,r2] = roots(a,b,c)
   r1 = ( -b + sqrt(b^2 - 4*a*c) ) / (2*a) ;
   r2 = ( -b - sqrt(b^2 - 4*a*c) ) / (2*a) ;
```

to compute the two roots r_1 and r_2 of the quadratic $ax^2 + bx + c$.

For $a = c = 1$ and $b = 10^7$, his function returned the values

$$r_1 = -9.9652 \times 10^{-8} \qquad r_2 = -1.0000 \times 10^{+7}$$

However, he knew that something was wrong, because he remembered from a high-school math course that the two roots r_1 and r_2 of the quadratic $ax^2 + bx + c$ satisfy $ar_1r_2 = c$, but his computed roots satisfied $ar_1r_2 = 0.9965$, while $c = 1$. So he knew that at least one of the two roots he calculated must be inaccurate.

Billy checked his function carefully, but he couldn't find anything wrong with it.

(a) Why did Billy's function compute an inaccurate result?

(b) Advise Billy on how to modify his function so that both computed roots are accurate.

5. [5 marks]

Find the polynomial $p(x)$ of degree 4 or less that satisfies

$$\begin{array}{rclcrclcrcl} p(-1) & = & 0 & \qquad & p(0) & = & 0 & \qquad & p(1) & = & 0 \\ p'(-1) & = & 1 & & & & & & p'(1) & = & 1 \end{array}$$

6. [5 marks]

Suppose you need to approximate e^x to 3-digit accuracy for many values of $x \in [0, 1]$ (possibly for a graphics application). For efficiency, instead of computing e^x itself, you could first find a low degree polynomial $q(x)$ that approximates e^x and then evaluate $q(x)$ at the required points $x \in [0, 1]$, instead of evaluation e^x at those points. (This will reduce the time required to run your program because it is typically much faster to evaluate a low degree polynomial than to evaluate `exp(x)`).

Suppose you choose to determine $q(x)$ by interpolating e^x at an appropriate set of points. How would you choose these points to ensure that

$$\max_{x \in [0,1]} |e^x - q(x)| \le 10^{-3} \tag{1}$$

while keeping the degree of $q(x)$ low?

Explain how you know that the associated polynomial $q(x)$ satisfies (1).

7. [5 marks]

Find the *complete cubic spline* $S(x)$ on the interval $[-1,1]$ with knots $\{-1,0,1\}$ that satisfies

$$\begin{array}{rclcrclcrcl} S(-1) & = & 1 & \quad & S(0) & = & 0 & \quad & S(1) & = & 1 \\ S'(-1) & = & 0 & & & & & & S'(1) & = & 0 \end{array}$$

8. [5 marks]

Explain how you can devise a scheme based on Simpson's Quadrature Rule

$$Q_S(f;a,b) = \frac{b-a}{2}\left[f(a) + 4f(m) + f(b)\right] \quad \text{where } m = \frac{a+b}{2}$$

to compute both an approximation $A(f;a,b)$ to

$$\int_a^b f(x)\,dx$$

and an estimate $E(f;a,b)$ of the error

$$\int_a^b f(x)\,dx - A(f;a,b)$$

using at most five evaluations of $f(x)$ and no higher derivatives of $f(x)$.

In answering this question, you may assume that $f^{(4)}(x)$ exists and is continuous. You can use $f^{(4)}(x)$ to justify your error estimate $E(f;a,b)$, but you may not use $f^{(4)}(x)$ in the formula for computing either $A(f;a,b)$ or $E(f;a,b)$.

Have a good summer.