End Semester Examination

Numerical Computing,
B. Math., 1st year,
January - April 2022.
Instructor: Prabuddha Chakraborty (pcphysics@gmail.com)

May 27th, 2022, Morning Session.
Duration: 3 hours.
Total points: 60.

Please give arguments where necessary. If it is unclear from your answer why a particular step is being taken, full credit will not be awarded. Grades will be awarded not only based on what final answer you get, but also on the intermediate steps.

1. Imagine that you have a calculator that can represent positive numbers (i.e., positive numbers uniquely representable on the calculator) between $10^{-499} \leq x \leq 9.999\ldots \times 10^{+499}$. The significand/mantissa $9.999\ldots$ has 12 decimal digits (as does any other number on the calculator). Look at the following code snippet (written using a standard **for** loop structure, applicable as syntax on most popular languages):

```
initialize real x
for i = 1 to 60,
x = sqrt(x)
end for loop
for i = 1 to 60,                                          [1]
x = x^2
end for loop
print x.
```

   What will be the output of this code on your calculator for any allowed real positive x ? You must explain your reasoning. Do remember $2^{10} = 1024$. [7 points]

2. You are trying to evaluate the following function $f(x) = \frac{e^x - 1}{x} = \sum_{i=0}^{\infty} \frac{x^i}{(i+1)!}$. Also imagine that your computer hardware/software/package evaluates the functions $f_1(x) = e^x, x \in \mathbb{R}$ and $f_2(x) = ln(x), x \in (0, \infty)$ with the relative error within the floating point accuracy bound ϵ_M of your machine, i.e., if $y = e^x$ is the exact relation, then $\hat{y} = y(1 + \delta), 0 < |\delta| < \epsilon_M$,

and similarly for the natural logarithm, with the same ϵ_M (The precise value of ϵ_M is not important, which in any case will vary from machine to machine). Below are two different algorithms, both evaluating $f(x)$:

Algorithm 1

```
if x = 0 set f=1
else f = (e$^x$-1)/x                                        [2]
```

Algorithm 2

```
y=e$^x$
if y = 1 set f=1                                            [3]
else f = (y-1)/ln(y)
```

Do the error analysis for the two algorithms for $|x| \ll 1$, and hence find which algorithm is more accurate for such values of x. Please note that no credit will be awarded for a prediction of the correct algorithm without the accompanying error analysis. [8 points]

3. Consider the function $f(x) = 1 - cx$, where $c > 0$. Solving $f(x) = 0$ is equivalent to calculating $1/c$, thereby doing a division numerically.

   (a) Find an interval $[a, b]$, or prescribe a method to calculate it, such that it is guaranteed to contain $1/c$. You must not use division while doing either. [6 points]

   (b) Assume $1 < c < 2$. By using some interval enclosing $1/c$, give an estimate of the number of subdivisions, n, needed to calculate $1/c$ to an accuracy of 2^{-25}. [4 points]

4. Let the Lagrange interpolation operator $L_n : \mathbb{R}^{n+1} \to \mathcal{P}_n$ be extended to any continuous function $f(x)$ such that $L_n f(x) = \sum_{i=0}^{n} f(x_i)\phi_i(x)$ where $\phi_i(x)$ are the usual Lagrange interpolation basis functions for *distinct* support point abscissae $x_i, i = 0 \ldots n$. Here $\mathcal{P}_n$ is the space of real polynomials of degree n.

   (a) Show that any L_n, restricted to act on $f(x) \in \mathcal{P}_n$, acts as a projection operator i.e., $L_n q(x) = q(x) \;\forall\; q(x) \in \mathcal{P}_n$. [2 points]

   (b) The error for the Lagrange interpolation, as defined above, of any continuous function $f(x)$ can be written as

$$f(x) - L_n f(x) = \frac{\omega_{n+1}(x)}{(n+1)!} f^{(n+1)}(\xi(x))$$

where $\omega_j(x) = \prod_{k=0}^{j-1}(x-x_k)$ (the result above was proved by repeated use of Rolle's theorem in class). Show, without Rolle's theorem, that the expression above is valid when $f(x)$ is the monomial $p(x) = x^{n+1}$. [3 points]

(c) If we define Newton's divided difference $f[x_0, x_1, ..., x_k]$ through the equation for Lagrange interpolation operator for distinct support abscissae:

$$L_n f(x) = \sum_{k=0}^{n} f[x_0, x_1, ..., x_k]\omega_k(x)$$

show that (for $n = 0$, define $f[x_0] = f(x_0)$)

$$f[x_0, x_1, ..., x_n] = \sum_{k=0}^{n} \frac{f(x_k)}{\left[\prod_{i \neq j}(x_i - x_j)\right]}$$

[7 points]

(d) In any Lagrange interpolation scheme, the commutator $[L_n, x]$ is defined, for any continuous function $f(x)$, as

$$[L_n, x]f(x) = L_n g(x) - g_n(x)$$

where $g(x) = xf(x)$ and $g_n(x) = x(L_n f(x))$. Show that

$$[L_n, x]f(x) = (-1)^{(n+1)} f[x_0, x_1, ..., x_n]\omega_n(x)$$

[8 points]

5. Derive (a) the Newton-Cotes formula for integration for any continuous function $f(x)$ in the interval $[a, b] = [0, 1]$ for $n = 3$ and (b) the corresponding error term following Peano's analysis. Please note that the derivation is necessary, and simply writing down the form is not enough to get any credit. [5 + 10 = 15 points].

6. (a) If $A \in \mathbb{C}^{m\times m}$ (assume A is of full rank), prove that the singular values of A (obtained from a SVD) are square roots of the eigenvalue decomposition $AA^\dagger$. ($A^\dagger$ is the Hermitian conjugate of A i.e., $A^\dagger_{ij} = A^*_{ji}$). [5 points]

(b) For A above, express $|det(A)|$ in terms of the singular values of A. Provide proof for your answer. [5]