**Name:** ______________________________

1. (25 points) In class, we saw that loss-of-significance errors can be avoided by rearranging the function evaluated. Consider the following functions $f(x)$ and do something similar assuming that $x \approx 0$ (unless stated otherwise):

   (a) (5 points) $f(x) = \dfrac{1-\cos x}{x^2}$

   (b) (5 points) $f(x) = \ln(x+1) - \ln x$, for $x \gg 1$

   (c) (5 points) $f(x) = \sin(a+x) - \sin a$

   (d) (5 points) $f(x) = \sqrt[3]{1+x} - 1$

   (e) (5 points) $f(x) = \dfrac{\sqrt{4+x}-2}{x}$

**Solution:**

(a) From the function given, we have

$$\begin{aligned}
\frac{1-\cos x}{x^2} &= \frac{1-\left[\cos^2\left(\frac{x}{2}\right) - \sin^2\left(\frac{x}{2}\right)\right]}{x^2} \\
&= \frac{1-\left[1-\sin^2\left(\frac{x}{2}\right) - \sin^2\left(\frac{x}{2}\right)\right]}{x^2} \\
&= \frac{1-\left[1-2\sin^2\left(\frac{x}{2}\right)\right]}{x^2} \Rightarrow \\
\boxed{\frac{1-\cos x}{x^2}} &\boxed{= \frac{2\sin^2\left(\frac{x}{2}\right)}{x^2}}.
\end{aligned}$$

(b) Here, we use the basic property of logarithms: $\ln A - \ln B = \ln(A/B)$, thus yielding

$$\ln(x+1) - \ln x = \ln\left(\frac{x+1}{x}\right) \Rightarrow \boxed{\ln(x+1) - \ln x = \ln\left(1+\frac{1}{x}\right)}.$$

(c) Similarly, we use the trigonometric identity:

$$\sin(C) - \sin(D) = 2\cos\left(\frac{C+D}{2}\right)\sin\left(\frac{C-D}{2}\right),$$

together with $C = x + a$ and $D = a$. This way, we obtain

$$\sin(a+x) - \sin a = 2\cos\left(\frac{a+x+a}{2}\right)\sin\left(\frac{a+x-a}{2}\right) \Rightarrow$$

$$\boxed{\sin(a+x) - \sin a = 2\cos\left(\frac{2a+x}{2}\right)\sin\left(\frac{x}{2}\right)}.$$

(d) For this function, we use the cubic identity:

$$a^3 - b^3 = (a-b)\left(a^2 + ab + b^2\right),$$

where we let $a = \sqrt[3]{1+x}$ and $b = 1$. This way, we multiply and divide $f(x)$ by $\left[\left(\sqrt[3]{1+x}\right)^2 + \sqrt[3]{1+x} + 1\right]$ such that

$$\begin{aligned}
\sqrt[3]{1+x} - 1 &= \left(\sqrt[3]{1+x} - 1\right) \cdot \frac{\left[\left(\sqrt[3]{1+x}\right)^2 + \sqrt[3]{1+x} + 1\right]}{\left[\left(\sqrt[3]{1+x}\right)^2 + \sqrt[3]{1+x} + 1\right]} \\
&= \frac{\left(\sqrt[3]{1+x}\right)^3 - 1^3}{\left[\left(\sqrt[3]{1+x}\right)^2 + \sqrt[3]{1+x} + 1\right]} \\
&= \frac{1+x-1}{\left[\left(\sqrt[3]{1+x}\right)^2 + \sqrt[3]{1+x} + 1\right]} \Rightarrow
\end{aligned}$$

$$\boxed{\sqrt[3]{1+x} - 1 = \frac{x}{\left[\left(\sqrt[3]{1+x}\right)^2 + \sqrt[3]{1+x} + 1\right]}}.$$

(e) Finally, and based on the function given in this case, we multiply the numerator and denominator by $\sqrt{4+x} + 2$ yielding

$$\begin{aligned}
\frac{\sqrt{4+x} - 2}{x} &= \frac{\sqrt{4+x} - 2}{x} \cdot \frac{\sqrt{4+x} + 2}{\sqrt{4+x} + 2} \\
&= \frac{\left(\sqrt{4+x}\right)^2 - 2^2}{x\left[\sqrt{4+x} + 2\right]} \\
&= \frac{4+x-4}{x\left[\sqrt{4+x} + 2\right]} \Rightarrow
\end{aligned}$$

$$\boxed{\frac{\sqrt{4+x} - 2}{x} = \frac{1}{\sqrt{4+x} + 2}}.$$

2. (15 points) In one of the examples we discussed in class about, concerned with a strategy for avoiding the loss-of-significance errors by using Taylor polynomials. Thus, use Taylor polynomial approximations to avoid such errors in the following functions when $x \approx 0$:

   (a) (5 points) $f(x) = \dfrac{e^x - e^{-x}}{2x}$

   (b) (5 points) $f(x) = \dfrac{\ln(1-x) + xe^{x/2}}{x^3}$

   (c) (5 points) $f(x) = \dfrac{x - \sin x}{x^3}$

**Solution:**

(a) Recall the Taylor polynomial approximation of $e^x$

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!},$$

such that

$$\begin{aligned}
\frac{e^x - e^{-x}}{2x} &\approx \frac{\left[1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!} - \left(1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \ldots + \frac{(-1)^n x^n}{n!}\right)\right]}{2x} \\
&\approx \frac{2\left[x + \frac{x^3}{3!} + \frac{x^5}{5!} \ldots + \frac{x^n}{n!} - \frac{(-1)^n x^n}{n!}\right]}{2x} \\
&\approx 1 + \frac{x^2}{3!} + \frac{x^4}{5!} \ldots + \frac{\frac{x^{n-1}}{n!} - \frac{(-1)^n x^{n-1}}{n!}}{2}. \qquad (1)
\end{aligned}$$

Two cases can be discerned:

- If $n$ is **odd**, then Eq. (1) suggests that

$$\begin{aligned}
\frac{e^x - e^{-x}}{2x} &\approx 1 + \frac{x^2}{3!} + \frac{x^4}{5!} \ldots + \frac{x^{n-1}}{n!} \Rightarrow \quad (n' = n - 1 \Rightarrow n = n' + 1) \\
\frac{e^x - e^{-x}}{2x} &\approx 1 + \frac{x^2}{3!} + \frac{x^4}{5!} \ldots + \frac{x^{n'}}{(n'+1)!} \Rightarrow \quad (n \equiv n') \\
\frac{e^x - e^{-x}}{2x} &\approx 1 + \frac{x^2}{3!} + \frac{x^4}{5!} \ldots + \frac{x^n}{(n+1)!}.
\end{aligned}$$

- If $n$ is **even**, then Eq. (1) (note that the $n$th terms cancel out and the surviving ones are the $(n-1)$th ones!) gives

$$\frac{e^x - e^{-x}}{2x} \approx 1 + \frac{x^2}{3!} + \frac{x^4}{5!} \ldots + \frac{x^{n-2}}{(n-1)!} \Rightarrow \quad (n' = n-2 \Rightarrow n = n'+2)$$
$$\frac{e^x - e^{-x}}{2x} \approx 1 + \frac{x^2}{3!} + \frac{x^4}{5!} \ldots + \frac{x^{n'}}{(n'+1)!} \Rightarrow \quad (n \equiv n')$$
$$\frac{e^x - e^{-x}}{2x} \approx 1 + \frac{x^2}{3!} + \frac{x^4}{5!} \ldots + \frac{x^{n}}{(n+1)!}.$$

Thus, in both cases the Taylor polynomial approximations to the respective function are the same:

$$\boxed{\frac{e^x - e^{-x}}{2x} \approx 1 + \frac{x^2}{3!} + \frac{x^4}{5!} \ldots + \frac{x^{n}}{(n+1)!}}.$$

(b) On equally footing, the Taylor polynomial approximation of the $\ln(1-x)$ is

$$\ln(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \ldots - \frac{x^n}{n} - \frac{x^{n+1}}{n+1}.$$

This way, we have

$$\begin{aligned}
\frac{\ln(1-x) + xe^{x/2}}{x^3} &\approx \frac{-x - \frac{x^2}{2} - \frac{x^3}{3} - \ldots - \frac{x^n}{n} - \frac{x^{n+1}}{n+1} + x\left[1 + \frac{x}{2} + \frac{x^2}{4\cdot 2!} + \ldots + \frac{x^n}{2^n n!}\right]}{x^3} \\
&\approx \frac{1}{x^3}\Bigg[x^3\left(\frac{1}{2^2\cdot 2} - \frac{1}{3}\right) + x^4\left(\frac{1}{2^3\cdot 3!} - \frac{1}{4}\right) + \\
&\quad \ldots + x^{n+1}\left(\frac{1}{2^n\cdot n!} - \frac{1}{n+1}\right)\Bigg] \Rightarrow
\end{aligned}$$

$$\boxed{\frac{\ln(1-x) + xe^{x/2}}{x^3} \approx \left(\frac{1}{2^2\cdot 2} - \frac{1}{3}\right) + x\left(\frac{1}{2^3\cdot 3!} - \frac{1}{4}\right) + \ldots + x^{n-2}\left(\frac{1}{2^n\cdot n!} - \frac{1}{n+1}\right)}.$$

(c) Finally, we consider the Taylor polynomial approximation of $\sin x$:

$$\sin x \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{(-1)^n x^{2n+1}}{(2n+1)!}.$$

Thus, the function given becomes:

$$
\begin{aligned}
\frac{x-\sin x}{x^3} &\approx \frac{x-\left(x-\frac{x^3}{3!}+\frac{x^5}{5!}+\ldots+\frac{(-1)^n x^{2n+1}}{(2n+1)!}\right)}{x^3} \\
&\approx \frac{\frac{x^3}{3!}-\frac{x^5}{5!}+\frac{x^7}{7!}-\ldots-\frac{(-1)^n x^{2n+1}}{(2n+1)!}}{x^3} \\
&\approx \frac{1}{3!}-\frac{x^2}{5!}+\frac{x^4}{7!}-\ldots+\frac{(-1)^{n+1}x^{2n-2}}{(2n+1)!} \quad (n'=n-1) \\
&\approx \frac{1}{3!}-\frac{x^2}{5!}+\frac{x^4}{7!}-\ldots+\frac{(-1)^{n'+2}x^{2n'}}{(2n'+3)!} \quad (n\equiv n') \Rightarrow \\
&\boxed{\frac{x-\sin x}{x^3} \approx \frac{1}{3!}-\frac{x^2}{5!}+\frac{x^4}{7!}-\ldots+\frac{(-1)^{n}x^{2n}}{(2n+3)!}}.
\end{aligned}
$$

3. (20 points) Let $f(x)$ be:

$$f(x)=\cosh x+\cos x-\gamma,$$

where $\gamma$ is a parameter and takes the values of $\gamma = 0, 1, 2, 3$. Make a graph of the function $f(x)$ for each value of $\gamma$ on the interval $[-3, 3]$ and determine whether $f(x)$ has a **root**. To do so, you have to check the criteria required by the **Intermediate Value Theorem**. Then, using the m-file "bisect.m", **approximate** the root with absolute tolerance $10^{-10}$ for the value of $\gamma$ that $f(x)$ does have a root. Include a copy of the graph of $f(x)$ for the respective cases and MATLAB output.

**Solution:** The graph of the function $f(x)$ is depicted in Fig. 1 for the respective values of $\gamma$. It can be discerned from this plot that $f(x)$ **does not** have any roots for $\gamma = 0, 1$, although for $\gamma = 2, 3$ does. However, and for $\gamma = 2$ at hand, it turns out that the only root here is $x^* = 0$, which corresponds to a **multiple root**. In particular, note that this is the case since $f(x^*) = 0$ and $f^{(k)}(x^*) = 0$, $\forall k \geq 2$, as well as the function itself for $\gamma = 2$ **does not** change sign near $x^*$. Thus, the intermediate value theorem **does not** apply here and we exclude this case.

On the other hand, the function $f(x)$ for $\gamma = 3$ has two simple (and isolated) roots that appear symmetrically in the figure. Furthermore, $f(x)$ is symmetric due to the fact that the function itself $f(x)$ is an *even* function, i.e., $f(x) = f(-x)$. This way, it suffices to find one root, say, $x^* = \alpha (> 0)$ while the other one corresponds to $x^* = -\alpha$. Thus, we employ the bisection method to find/approximate the positive root by considering $[a, b] = [1, 3]$.

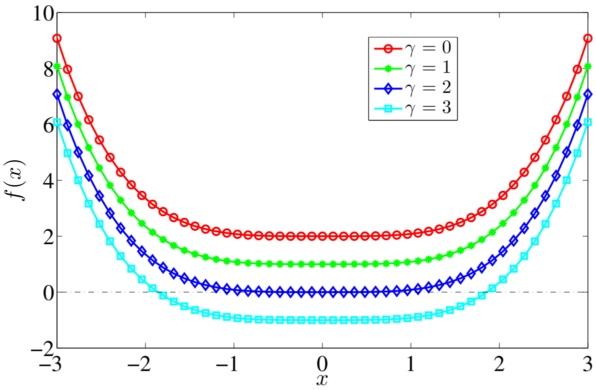


Figure 1: The graph of the function $f(x)$ of Question 3 for various values of $\gamma$ (see, the legend as well).

The MATLAB output follows:

```
        x_{k}                  f(x_{k})
---------------------------------------------------------
  2.000000000000000   3.460488545364888e-01
  1.500000000000000  -5.768531830890500e-01
  1.750000000000000  -2.140577459214046e-01
  1.875000000000000   3.755353739794653e-02
  1.812500000000000  -9.486303597334933e-02
  1.843750000000000  -3.036814157462109e-02
  1.859375000000000   3.156614216319742e-03
  1.851562500000000  -1.371381071232225e-02
  1.855468750000000  -5.305731306345596e-03
  1.857421875000000  -1.081357010183304e-03
  1.858398437500000   1.035927084317656e-03
  1.857910156250000  -2.314010470660932e-05
  1.858154296875000   5.062871746299713e-04
  1.858032226562500   2.415469598848752e-04
  1.857971191406250   1.091967842845598e-04
  1.857940673828125   4.302667902145174e-05
  1.857925415039062   9.942871972867806e-06
  1.857917785644531  -6.598720161843374e-06
  1.857921600341797   1.672049956269461e-06
  1.857919692993164  -2.463341590264179e-06
  1.857920646667480  -3.956474392552423e-07
  1.857921123504639   6.382008530536609e-07
  1.857920885086060   1.212766056468695e-07
```

```
  1.857920765876770  -1.371854416731821e-07
  1.857920825481415  -7.954424674494476e-09
  1.857920855283737   5.666108915391987e-08
  1.857920840382576   2.435333223971270e-08
  1.857920832931995   8.199453560564507e-09
  1.857920829206705   1.225148871242254e-10
  1.857920827344060  -3.915955115729730e-09
  1.857920828275383  -1.896720114302752e-09
  1.857920828741044  -8.871028356338684e-10
  1.857920828973874  -3.822937522102166e-10
  1.857920829090290  -1.298894325429956e-10
  1.857920829148497  -3.687716798594920e-12
x^{\ast  } =

   1.857920829148497
```

It should be noted in passing that the values of the $f(x_k)$-column *oscillate*, that is, they do not go to 0 monotonically. Such an observation suggests that for the bisection method the successive estimates $x_k$ may get close to $x^*$ and then move away *slightly* before continuing towards $x^*$ again.

4. (10 points) Consider **Newton's method** for finding $+\sqrt{\alpha}$ with $\alpha > 0$ by finding the **positive root** of $f(x) = x^2 - \alpha = 0$. Assuming that $x_0 > 0$, show the following:

   (a) (5 points)
   $$x_{k+1} = \frac{1}{2}\left(x_k + \frac{\alpha}{x_k}\right),$$

   (b) (5 points)
   $$x_{k+1}^2 - \alpha = \left(\frac{x_k^2 - \alpha}{2x_k}\right)^2,$$
   for $k \geq 0$ and therefore $x_k > \sqrt{\alpha}$ for $k \geq 1$. Note that the formula in part (a) refers to as the "Babylonian method" for computing square roots!

**Solution:**

(a) Let us employ Newton's method to $f(x)$. In particular, we have that

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^2 - \alpha}{2x_k} = \frac{x_k^2 + \alpha}{2x_x} \Rightarrow \boxed{x_{k+1} = \frac{1}{2}\left(x_k + \frac{\alpha}{x_k}\right)}.$$

(b) Starting from the left-hand-side (lhs) of part (b) and utilizing the result from part (a), we obtain

$$\begin{aligned} x_{k+1}^2 - \alpha &= \frac{1}{4}\left(x_k + \frac{\alpha}{x_k}\right)^2 - \alpha = \frac{1}{4}\left(x_k^2 + \frac{\alpha^2}{x_k^2} - 2\alpha\right) = \frac{1}{4}\left(\frac{x_k^2-\alpha}{x_k}\right)^2 \\ &= \left(\frac{x_k^2-\alpha}{2x_k}\right)^2. \end{aligned}$$

Therefore, if $x_k^2 \neq \alpha$, then $\left(\frac{x_k^2-\alpha}{2x_k}\right)^2 > 0$ which implies $x_{k+1}^2 > \alpha$, or $x_{k+1} > \sqrt{\alpha}$ for $k \geq 0$. It then follows that $x_k > \sqrt{\alpha}$ for $k \geq 1$.

5. (30 points) **Implement Newton's method** in MATLAB (or in **any** programming language). In particular, create a function, which utilizes the method, and store it as an m-file. Then, use your function to estimate $\sqrt{7}$ by finding the **positive root** of $f(x) = x^2 - 7$. Try two different initial guesses: (i) $x_0 = 2$ and (ii) $x_0 = 500$ and consider $|x_{k+1} - x_k| < 10^{-10}$ as a convergence criterion.

Attach your codes and provide MATLAB output for both cases.

**Solution:** An implementation of Newton's method is given in the following script named as `newton.m`:

```
function [ x, k, fout ] = newton( func, dfunc, x0, atol, nmax )
%
% On input:
%           a)  func: is the nonlinear function f(x).
%           b) dfunc: the derivative of f(x).
%           c)    x0 : initial guess.
%           d) atol : absolute tolerance.
%           e) nmax : maximum number of iterations allowed.
%
% On output:
%           a)     x : is the root of f(x).
%           b)     k : number of iterations required to achieve atol.
%           c)  fout: Residuals store at each k.

   xk = rand;                        % Just to start the while loop.
  res = xk - x0;                     % Compute the absolute error.
    k = 0;                           % Iteration index.
    iflag = 0;                       % For stopping purposes.
    while abs(res)>atol              % Do-while loop.
        fxk = feval(func,x0);        % Compute the residual.
```

```
        fout(k+1) = fxk;          % Store the residual.
       dfxk = feval(dfunc,x0);    % Evaluate the derivative of ...
           f(x).
          xk = x0 - fxk / dfxk;   % Newton step.
         res = xk - x0;           % Compute the "new" absolute ...
             error.
         x0 = xk;                 % Store it.
         k = k + 1;               % Increase iteration index.
         if(k == nmax)            % Check whether we reached ...
             the maximum
             iflag = -1;          % number of iterations!
             break;               % If yes, then we stop Newton.
         end
     end
     if(iflag==0)                 % On successful exit, return ...
         the root!
         k = k + 1;
   fout(k) = feval(func,xk);
   xout(k) = xk;
         x = xk;
     else
         disp(' maximum number of iterations reached!');
         x = [];
         k = [];
     end
 end
```

Then, the main driver consists of the following code which calls `newton.m`:

```
clearvars; close all; clc; format long;
 func = @(x) x^2-7;
dfunc = @(x) 2 * x;

x0 = 2; atol = 1e-10; nmax = 100;
[ xstar, k, fout ] = newton( func, dfunc, x0, atol, nmax );
```

and the corresponding MATLAB output for $x_0 = 2$ is given below:

```
        x_{k}                    f(x_{k})
------------------------------------------------------------
   2.000000000000000   -3.000000000000000e+00
   2.750000000000000    0.562500000000000e+00
   2.647727272727273    0.010459710743802e+00
   2.645752048380804    3.901511218984410e-06
   2.645751311064693    5.435651928564766e-13
   2.645751311064591    8.881784197001252e-16
```

Similarly, for $x_0 = 500$:

```
       x_{k}                 f(x_{k})
----------------------------------------------------------
  5.000000000000000e+02 2.499930000000000e+05
  2.500070000000000e+02 6.249650004900000e+04
  1.250174996080110e+02 1.562237520823903e+04
  6.253674588464200e+01 3.903844585840287e+03
  3.132434003735431e+01 9.742142787757983e+02
  1.577390421629556e+01 2.418160542248670e+02
  8.108837568589735e+00 5.875324671377228e+01
  4.486046618789610e+00 1.312461426595369e+01
  3.023220284018386e+00 2.139860885700212e+00
  2.669316055303705e+00 1.252482031021342e-01
  2.645855325943224e+00 5.504058221266206e-04
  2.645751313109127e+00 1.081866862762126e-08
  2.645751311064591e+00 8.881784197001252e-16
```

*Date*: January 23, 2020

Mathematics Department, California Polytechnic State University, San Luis Obispo, CA 93407-0403, USA
*Email address*: echarala@calpoly.edu