Lab Session 1

MA581 : Numerical Computations Lab R. Alam August 05, 2025

1. The purpose of this exercise is to illustrate anomaly in automatic computation. On my computer, MATLAB produces

$$\begin{aligned} (\frac{4}{3} - 1) * 3 - 1 &= -2.2204 \times 10^{-16} \\ 5 \times \frac{(1 + \exp(-50)) - 1}{(1 + \exp(-50)) - 1} &= \textbf{NaN} \\ \frac{\log(\exp(750))}{100} &= \textbf{Inf} \end{aligned}$$

   Try on your machine. Can you explain the reason behind these anomalies?

2. Consider $(\beta, t, e_{\min}, e_{\max}) = (10, 8, -99, 99)$ and the normalized floating-point numbers

$$\begin{aligned} x &= 0.23371258 \times 10^{-4} \\ y &= 0.33678429 \times 10^{2} \\ z &= -0.33677811 \times 10^{2} \end{aligned}$$

   Use MATLAB command `round` with $t = 8$ to calculate $\mathrm{fl}(x + \mathrm{fl}(y + z))$ and $\mathrm{fl}(\mathrm{fl}(x + y) + z)$. Is $\mathrm{fl}(x + \mathrm{fl}(y + z)) = \mathrm{fl}(\mathrm{fl}(x + y) + z)$?

   Next, calculate $x + y + z$ in MATLAB and determine the relative errors in calculating $\mathrm{fl}(x + \mathrm{fl}(y + z))$ and $\mathrm{fl}(\mathrm{fl}(x + y) + z)$.

3. This example illustrates finite difference approximation of derivative of a function. Let $f : \mathbb{R} \to \mathbb{R}$ be continuously differentiable. Then

$$f'(c) = \lim_{h \to 0} \frac{f(c + h) - f(c)}{h} \approx \frac{f(c + h) - f(c)}{h} =: D_h f(c).$$

   Consider the error $e_c(h) := |f'(c) - D_h f(c)|$. The following code plots $e_c(h)$ for $f(x) = e^x$. Check that the error $e_c(h)$ increases when h decreases in the interval $[0, \sqrt{eps}]$.

```
>> h=10.^(-15:0);
>> f=@(x) exp(x);
>> c=1;
>> fp=(f(c+h)-f(c))./h;
>> loglog(h,abs(fp-exp(c)));
```

4. Write a code for computing $r := \sqrt{x^2 + y^2}$ that avoids overflow. First check that if $\max(|x|, |y|) > \sqrt{\texttt{realmax}}$ then $r = \texttt{inf}$ if r is computed as $\sqrt{x^2 + y^2}$. Check that the following code avoids over flow. Does it avoid underflow?

```
m=max(abs(x),abs(y));
if m==0,
r=0
else
r=m*sqrt((x/m)^2+(y/m)^2)
end
```

5. Consider the power series for $\sin x$ given by

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

Here is a Matlab function that uses this series to compute $\sin x$.

```
function s = powersin(x)
%
% POWERSIN(x) tries to comptue sin(x) from a power series
%
  s = 0;
  t = x;
  n = 1;
  while s + t ~= s;
   s = s + t;
   t = -x.^2/((n+1)*(n+2)).*t;
   n = n + 2;
  end
```

When does the while loop terminate? Answer each of the following questions for $x = \pi/2, 11\pi/2, 21\pi/2$ and $31\pi/2$.

- How accurate is the computed result?
- How many terms are required?

What is your conclusion about the use of floating point arithmetic and power series to evaluate functions?

6. This example illustrates how to avoid cancellation. Suppose we wish to compute e^x from the series

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots \approx \sum_{j=0}^{n} \frac{x^n}{n!} =: S_n.$$

Computing S_n as it is leads to cancellation when x is negative. Here is a naive algorithm.

```
function s=naivexp(x,tol);
%  computation of the exponential function
% s=naivexp(x,tol); computes an approximation s of exp(x)
% up to a given tolerance tol.
% WARNING: cancellation for large negative x.
s=1; term=1; k=1;
while abs(term)>tol*abs(s)
so=s; term=term*x/k;
s=so+term; k=k+1;
end
```

Now compute `naivexp(20,1e-8)` and `naivexp(1,1e-8)` and compare the values with those of MATLAB function `exp(20)` and `exp(1)`, respectively.

Next, compute `naivexp(-20,1e-8)` and `naivexp(-1,1e-8)` and compare the values with those of `exp(-20)` and `exp(-1)`. What is your observation?

The better (stable) algorithm is obtained by noting the fact that $e^x = \frac{1}{e^{|x|}}$ when $x < 0$.

```
function s=myexp(x);
% myexp stable computation of the exponential function
% s=myexp(x); computes an approximation s of exp(x) up to machine
% precision.
if x<0, v=-1; x=abs(x); else v=1; end
so=0; s=1; term=1; k=1;
while s~=so
so=s; term=term*x/k;
s=so+term; k=k+1;
end
if v<0, s=1/s;
end;
```

Now compare `myexp(-20)` and `myexp(-50)` with those of `exp(-20)` and `exp(-50)`. What is your observation?

7. Consider the function $f(x) = (e^x - 1)/x$, which arises in various applications. By L'Hopital's rule, we know that $\lim_{x\to 0} f(x) = 1$.

   (a) Compute the values of $f(x)$ for $x = 10^{-n}$ for $n = 1, 2, \ldots, 16$. Do your results agree with theoretical expectations?

   (b) Now perform the computation in part (a) again, this time using the mathematically equivalent formulation $f(x) = (e^x - 1)/\log(e^x)$ (evaluate as indicated without simplification). Does this work any better?

8. Consider the recurrence $x_{k+1} = 111 - (1130 - 3000/x_{k-1})/x_k,\ x_0 = 11/2,\ x_1 = 61/11$. In exact arithmetic the x_k form a monotonically increasing sequence that converges to 6. Implement the recurrence in MATLAB and compare the computed x_{34} with the true value 5.998 (correct to four digits). Try to explain what you see.

*** End ***