1 Preliminaries

Numbers are represented in binaries, thus creating errors.
Numerical procedures also introduce errors.
Numerical analysis is the study of the behavior of errors in computation.

- Suppose that $\widehat{p}$ is an approximation to p. The (absolute) error is $E_p = |p - \widehat{p}|$, and the relative error is $R_p = \frac{E_p}{|p|}$, provided that $p \neq 0$.
  - Let $x = 3.141592$ (approx. π?) and $\widehat{x} = 3.14$; then the error is

    $$E_x = |\widehat{x} - x| = |3.14 - 3.141592| = 0.001592$$

    and the relative error is

    $$R_x = \frac{|\widehat{x} - x|}{|x|} = \frac{0.001592}{3.141592} = 0.00507$$

  - Let $y = 1,000,000$ and $\widehat{y} = 999,996$; then the error is (large?)

    $$E_y = |\widehat{y} - y| = |999996 - 1000000| = 4$$

    and the relative error is (small?)

    $$R_y = \frac{|\widehat{y} - y|}{|y|} = \frac{4}{1000000} = 0.000004$$

  - Let $z = 0.000012$ and $\widehat{z} = 0.000009$; then the error is (small?)

    $$E_z = |\widehat{z} - z| = |0.000009 - 0.000012| = 0.000003$$

    and the relative error is (large?)

    $$Rz = \frac{E_z}{|z|} = \frac{0.000003}{0.000012} = 0.25 = 25\%$$

    The relative error R_p is a better indicator of accuracy and is preferred for floating-point representations since it deals directly with the mantissa.

- The number $\widehat{p}$ is said to approximate p to d significant digits if d is the largest positive integer for which

  $$\frac{|\widehat{p} - p|}{|p|} < 0.5 \times 10^{-d}$$

- If $x = 3.141592$ and $\widehat{x} = 3.14$, then $\frac{|\widehat{x}-x|}{|x|} = 0.000507 < 0.5 \times 10^{-2}$. Therefore, $\widehat{x}$ approximates x to 2 significant digits.
- If $y = 1000000$ and $\widehat{y} = 999996$, then $\frac{|\widehat{y}-y|}{|y|} = 0.000004 < 0.5 \times 10^{-2}$. Therefore, $\widehat{y}$ approximates y to 5 significant digits.
- If $z = 0.000012$ and $\widehat{z} = 0.000009$, then $\frac{|\widehat{z}-z|}{|z|} = 0.25 < 0.5 \times 10^{-2}$. Therefore, $\widehat{z}$ approximates z to no significant digits.

- Given that

$$p = \int_0^{1/2} e^{x^2} dx = 0.544987104184$$

and is approximated by using Taylor series as

$$\widehat{p} = \int_0^{1/2} P_8(x)dx = \left[x + \frac{x^3}{3} + \frac{x^5}{5(2)!} + \frac{x^7}{7(3)!} + \frac{x^9}{9(4)!} \right]_0^{1/2} = 0.544986720817$$

Since $0.5 * 10^{-5} > R_p = 7.03442 \times 10^{-7} > 10^{-6}/2$, the approximation $\widehat{p}$ agrees with the true answer p to 5 significant figures.

- Calculate $f(500)$ and $g(500)$ using 6 digits and rounding, with

$$f(x) = x(\sqrt{x+1} - \sqrt{x}), \ \ g(x) = \frac{x}{\sqrt{x+1} + \sqrt{x}}$$

We have

$$f(500) = 500(\sqrt{501} - \sqrt{500} = 500(22.3830 - 22.3607) = 500(0.0223) = 11.1500$$

$$g(500) = \frac{500}{\sqrt{501} + \sqrt{500}} = \frac{500}{22.3830 + 22.3607} = \frac{500}{44.7437} = 11.1748$$

Note that $g(x)$ is algebraically equivalent to $f(x)$, but $g(500) = 11.1748$ is more accurate than $f(500)$ to the true answer $11.174755300747198\ldots$ to six digits.

- Let $P(x) = x^3 - 3x^2 + 3x - 1$, $Q(x) = ((x-3)x+3)x-1$. Use 3-digit rounding arithmetic to compute $P(2.19) = Q(2.19) = 1.685159$:

$$P(2.19) \approx 2.19^3 - 3(2.19)^2 + 3(2.19) - 1 = 10.5 - 14.4 + 6.57 - 1 = 1.67$$

$$Q(2.19) \approx ((2.19-3)2.19+3)2.19 - 1 = 1.69$$

The errors are 0.015159 and -0.004841, respectively. Thus the approximation $Q(2.19) \approx 1.69$ has less error.

- Consider the Taylor polynomial expansions

$$e^h = 1 + h + \frac{h^2}{2!} + \frac{h^3}{3!} + O(h^4)$$

$$cosh = 1 - \frac{h^2}{2!} + \frac{h^4}{4!} + O(h^6)$$

With $O(h^4) + O(h^6) = O(h^4) = O(h^4) + \frac{h^4}{4!}$, we have the sum

$$e^h + cosh = 2 + h + \frac{h^3}{3!} + \frac{h^4}{4!} + O(h^4) + O(h^6) = 2 + h + \frac{h^3}{3!} + O(h^4)$$

The difference behaves similarly.
The product

$$e^h * cosh = \left(1 + h + \frac{h^2}{2!} + \frac{h^3}{3!} + O(h^4)\right) * \left(1 - \frac{h^2}{2!} + \frac{h^4}{4!} + O(h^6)\right)$$

$$= \left(1 + h + \frac{h^2}{2!} + \frac{h^3}{3!}\right) * \left(1 - \frac{h^2}{2!} + \frac{h^4}{4!}\right) + \left(1 + h + \frac{h^2}{2!} + \frac{h^3}{3!}\right) * O(h^6) +$$

$$\left(1 - \frac{h^2}{2!} + \frac{h^4}{4!}\right) * O(h^4) + O(h^4) * O(h^6)$$

$$= 1 + h - \frac{h^3}{3} - \frac{5h^4}{24} - \frac{h^5}{24} + \frac{h^6}{48} + \frac{h^7}{144} + O(h^6) + O(h^4) + O(h^4)O(h^6)$$

$$= 1 + h - \frac{h^3}{3} + O(h^4)$$

and the order of approximation is $O(h^4)$.

- $x_n = \frac{1}{3^n}$, approximated by (for n = 1, 2, $\cdots$)

$$r_0 = 1, r_n = \frac{1}{3} r_{n-1} \left(= \frac{A}{3^n}\right)$$

$$p_0 = 1, p_1 = \frac{1}{3}, p_n = \frac{4}{3} p_{n-1} - \frac{1}{3} p_{n-2} \left(A \frac{1}{3^n} + B\right)$$

$$q_0 = 1, q_1 = \frac{1}{3}, q_n = \frac{10}{3} q_{n-1} - q_{n-2} \left(A \frac{1}{3^n} + B 3^n\right)$$

Generate a table for $x_n - r_n, x_n - p_n, x_n - q_n$, with errors introduced in the starting values:

$$r_0 = 0.99996, p_0 = q_0 = 1, p_1 = q_1 = 0.33332$$

The error for r_n is stable and decreases exponentially.
The error for p_n is stable, but eventually dominates as $p_n \to 0$.
The error for q_n is unstable and grows exponentially.

- Write the following code and study the response.

```
%      Determines effective machine precision for MATLAB
  a = 1.0 ;
  while  ( (1. + a) ~= 1)
     a     = a/2.  ;
  end
  delta = 2.0*a ;
  sprintf(' Machine Precision of MATLAB  is  %9.2e', delta )
```

- Write the following code and study the response.

```
% uses the MATLAB   chop.m   function to find simulated  machine
% precision for a NDIGITS decimal ( base 10 ) machine.
data = [] ;
for NDIGITS = 2: 20 ;
   a = 1.0 ;
   while  ( chop( (1.+a), NDIGITS ) ~= chop( (1.+a/2.), NDIGITS) )
       a  = chop( a/2. , NDIGITS) ;
   end
   theoret = 0.5*10^(1-NDIGITS) ;
   data = [ data ; NDIGITS (1.5)*a theoret ] ;
end
%  Note the use of (semi)logarithmic plots is usually preferable
%  for displaying error behavior.
semilogy( data(:,1) , data(:,2) , '*',  ...
            data(:,1) , data(:,3)  ) ;
xlabel('NDIGITS');
ylabel('Machine Precision')
legend('Observed','Theoretical');
title('Dependence of Machine Precision on Machine "Size"');
```

- Write the following code and study the response.

```
% Determines the accuracy of a computed expression which is potentially
% subject to cancellation errors, using the MATLAB chop.m  function.
  clear ;
  data = [] ;
  NDIGITS    = 8 ;
  mu_NDIGITS = 0.5*10^(1-NDIGITS) ;
  mu_calc    = 50*mu_NDIGITS      ;
```

```
  for n = 1: 30 ;
     x = 2^n ;
     xsing       = chop( x , NDIGITS ) ;
     xm1_sing    = chop( xsing - 1 , NDIGITS ) ;
     xsq_sing    = chop( xsing*xsing , NDIGITS) ;
     xsqp4_sing = chop( xsq_sing + 4 , NDIGITS ) ;
     sroot_sing = chop( sqrt( xsqp4_sing ) , NDIGITS ) ;
     fval_sing   = chop( sroot_sing - xm1_sing , NDIGITS ) ;
     f_double    = sqrt( x^2 + 4 ) - ( x - 1 ) ;
     rel_err     = abs( f_double - fval_sing )/abs(f_double + eps ) + eps ;
     data        = [ data ; x rel_err f_double fval_sing]  ;
  end
  xmin =  min(data(:,1)) ;   xmax =  max(data(:,1)) ;
  loglog( data(:,1) , data(:,2) , '-.' , ...
               [ xmin xmax ] , [ mu_calc  mu_calc ] , ':' ) ;
  axis( [ xmin  10*xmax   10^(-10) 10^3 ] );
  xlabel( 'x' ) ; ylabel( 'Relative Difference') ;
  legend('Observed','"Acceptable"');
  title('Variation of the Accuracy of a Computed Function with x');
  figure(2);
  semilogx( data(:,1), data(:,3), data(:,1), data(:,4),':');
  xlabel('x') ; ylabel('Computed Value of f(x)')
  axis([min(data(:,1)), 10*max(data(:,1)),-.25, 2.25])
  legend('Double Precision','Single Precision');
  title('Effect of Machine Precision on the Accuracy of a Computed Function
```

- Write the code and analysis output

```
a=123*2*pi*/360
L=inline('9/sin(pi-2.1468-c)+7/sin(c)')
fplot(L,[0.4,0.5]); grid on
fminbnd(L,0.4,0.5)
L(0.4677)
fminbnd(L,0.4,0.5,optimset('Display','iter'))
```

- Write a code that adds 0.0001 one thousand times. The result should equal 1.0 exactly but this is not true for single precision.

```
k=0.0;
j=0.0001;
for l=1:10000
```

```
 k=k+j;
end
k
k=0.0;
j=0.0001;
a=single(k);
b=single(j);
for l=1:10000
 a=a+b;
end
```

- Write a code that computes values of this expression

$$z = \frac{(x+y)^2 - 2xy - y^2}{x^2}$$

with different values of x and y. (Hint: use $y = 10000$ and change the x-value as $0.01, 0.001, 0.0001, , \ldots$)

```
x=0.01;  y=10000;z=((x+y)^2-2*x*y-y^2)/x^2;
x=0.001;  y=10000;z=((x+y)^2-2*x*y-y^2)/x^2
x=0.0001;  y=10000;z=((x+y)^2-2*x*y-y^2)/x^2
x=0.00001;  y=10000;z=((x+y)^2-2*x*y-y^2)/x^2
x=0.000001;  y=10000;z=((x+y)^2-2*x*y-y^2)/x^2
x=0.0000001;  y=10000;z=((x+y)^2-2*x*y-y^2)/x^2
```