CHAPTER 4

4.1 The function can be developed as

```
function [fx,ea,iter] = SquareRoot(a,es,maxit)
% Divide and average method for evaluating square roots
%   [fx,ea,iter] = SquareRoot(a,es,maxit)
% input:
%   a = value for which square root is to be computed
%   es = stopping criterion (default = 0.0001)
%   maxit = maximum iterations (default = 50)
% output:
%   fx = estimated value
%   ea = approximate relative error (%)
%   iter = number of iterations

% defaults:
if nargin<2|isempty(es),es=0.0001;end
if nargin<3|isempty(maxit),maxit=50;end
if a<= 0,error('value must be positive'),end
% initialization
iter = 1; sol = a/2; ea = 100;
% iterative calculation
while (1)
  solold = sol;
  sol = (sol + a/sol)/2;
  iter = iter + 1;
  if sol~=0
    ea=abs((sol - solold)/sol)*100;
  end
  if ea<=es | iter>=maxit,break,end
end
fx = sol;
end
```

It can be tested for the following cases:

```
>> [fx,ea,iter] = SquareRoot(25)

fx =
     5
ea =
    3.355538069627073e-010
iter =
     7

>> [fx,ea,iter] = SquareRoot(25,0.001)
fx =
   5.000000000016778
ea =
    2.590606381316551e-004
iter =
     6

>> [fx,ea,iter] = SquareRoot(0)
```

```
??? Error using ==> SquareRoot at 16
value must be positive
```

4.2 (a)

$$(1011001)_2 = (1\times 2^6)+(0\times 2^5)+(1\times 2^4)+(1\times 2^3)+(0\times 2^2)+(0\times 2^1)+(1\times 2^0)$$
$$= 1(64)+0(32)+1(16)+1(8)+0(4)+0(2)+1(1) = 89$$

(b)

$$(0.01011)_2 = (0\times 2^{-1})+(1\times 2^{-2})+(0\times 2^{-3})+(1\times 2^{-4})+(1\times 2^{-5})$$
$$= 0(0.5)+1(0.25)+0(0.125)+1(0.0625)+1(0.03125)$$
$$= 0+0.25+0+0.0625+0.03125 = 0.34375$$

(c)

$$(110.01001)_2 = (1\times 2^2)+(1\times 2^1)+(0\times 2^0)+(0\times 2^{-1})+(1\times 2^{-2})+(0\times 2^{-3})+(0\times 2^{-4})+(1\times 2^{-5})$$
$$= 1(4)+1(2)+0(1)+0(0.5)+1(0.25)+0(0.125)+0(0.0625)+1(0.03125)$$
$$= 4+2+0+0+0.25+0+0+0.03125 = 6.28125$$

4.3

$$(61565)_8 = (6\times 8^4)+(1\times 8^3)+(5\times 8^2)+(6\times 8^1)+(5\times 8^0)$$
$$= 6(4096)+1(512)+5(64)+6(8)+5(1)$$
$$= 24576+512+320+48+5 = 25,461$$
$$(2.71)_8 = (2\times 8^0)+(7\times 8^{-1})+(1\times 8^{-2}) = 2(1)+7(0.125)+1(0.015625) = 2.890625$$

4.4

```
function ep = macheps
% determines the machine epsilon
e = 1;
while (1)
  if e+1<=1, break, end
  e = e/2;
end
ep = 2*e;

>> macheps
ans =
  2.2204e-016

>> eps
ans =
  2.2204e-016
```

4.5

```
function s = small
% determines the smallest number
sm = 1;
while (1)
  s=sm/2;
  if s==0,break,end
  sm = s;
```

```
end
s = sm;
```

This function can be run to give

```
>> s=small
s =
  4.9407e-324
```

This result differs from the one obtained with the built-in `realmin` function,

```
>> s=realmin
s =
  2.2251e-308
```

Challenge question: We can take the base-2 logarithm of both results,

```
>> log2(small)
ans =
      -1074
>> log2(realmin)
ans =
      -1022
```

Thus, the result of our function is $2^{-1074} (\cong 4.9407 \times 10^{-324})$, whereas `realmin` gives $2^{-1022} (\cong 2.2251 \times 10^{-308})$. Therefore, the function actually gives a smaller value that is equivalent to

$$\texttt{small} = 2^{-52} \times \texttt{realmin}$$

Recall that machine epsilon is equal to 2^{-52}. Therefore,

$$\texttt{small} = \texttt{eps} \times \texttt{realmin}$$

Such numbers, which are called *denormal* or *subnormal*, arise because the math coprocessor employs a different strategy for representing the significand and the exponent.

4.6 Because the exponent ranges from –126 to 127, the smallest positive number can be represented in binary as

$$\text{smallest value} = +1.0000\ldots0000 \cdot 2^{-126}$$

where the 23 bits in the mantissa are all 0. This value can be translated into a base-10 value of $2^{-126} = 1.1755 \times 10^{-38}$. The largest positive number can be represented in binary as

$$\text{largest value} = +1.1111\ldots1111 \cdot 2^{+127}$$

where the 23 bits in the mantissa are all 1. Since the significand is approximately 2 (it is actually $2-2^{-23}$), the largest value is, therefore, $2^{+128} = 3.4028\times10^{38}$. The machine epsilon in single precision would be $2^{-23} = 1.1921\times10^{-7}$.

These results can be verified using built-in MATLAB functions,

```
>> realmax('single')
ans =
   3.4028e+038

>> realmin('single')
ans =
   1.1755e-038

>> eps('single')
ans =
   1.1921e-007
```

4.7 For computers that use truncation, the machine epsilon is the positive distance from $|x|$ to the next larger (in magnitude) floating point number of the same precision as *x*.

$$1.1\times10^0 - 1.0\times10^0 = 1.0\times10^{-1}$$

For computers that hold intermediate results in a larger-sized word before returning a rounded result, the machine epsilon would be 0.05.

4.8 The true value can be computed as

$$f'(0.577) = \frac{6(0.577)}{(1-3\times0.577^2)^2} = 2{,}352{,}911$$

Using 3-digits with chopping

$$6x = 6(0.577) = 3.462 \xrightarrow{\text{chopping}} 3.46$$
$$x = 0.577$$
$$x^2 = 0.332929 \xrightarrow{\text{chopping}} 0.332$$
$$3x^2 = 0.996$$
$$1-3x^2 = 0.004$$
$$f'(0.577) = \frac{3.46}{(1-0.996)^2} = \frac{3.46}{0.004^2} = 216{,}250$$

This represents a percent relative error of

$$\varepsilon_t = \left|\frac{2{,}352{,}911 - 216{,}250}{2{,}352{,}911}\right| \times 100\% = 90.8\%$$

Using 4-digits with chopping

$$6x = 6(0.577) = 3.462 \xrightarrow{\text{chopping}} 3.462$$
$$x = 0.577$$
$$x^2 = 0.332929 \xrightarrow{\text{chopping}} 0.3329$$
$$3x^2 = 0.9987$$
$$1 - 3x^2 = 0.0013$$
$$f'(0.577) = \frac{3.462}{(1 - 0.9987)^2} = \frac{3.462}{0.0013^2} = 2{,}048{,}521$$

This represents a percent relative error of

$$\varepsilon_t = \left|\frac{2{,}352{,}911 - 2{,}048{,}521}{2{,}352{,}911}\right| \times 100\% = 12.9\%$$

Although using more significant digits improves the estimate, the error is still considerable. The problem stems primarily from the fact that we are subtracting two nearly equal numbers in the denominator. Such subtractive cancellation is worsened by the fact that the denominator is squared.

4.9 First, the correct result can be calculated as

$$y = 1.37^3 - 7(1.37)^2 + 8(1.37) - 0.35 = 0.043053$$

(a) Using 3-digits with chopping

1.37^3	$\rightarrow$	2.571353	$\rightarrow$	2.57
$-7(1.37)^2$	$\rightarrow$	$-7(1.87)$	$\rightarrow$	-13.0
$8(1.37)$	$\rightarrow$	10.96	$\rightarrow$	10.9
				$-\underline{0.35}$
				0.12

This represents an error of

$$\varepsilon_t = \left|\frac{0.043053 - 0.12}{0.043053}\right| \times 100\% = 178.7\%$$

(b) Using 3-digits with chopping

$$y = ((1.37 - 7)1.37 + 8)1.37 - 0.35$$
$$y = (-5.63 \times 1.37 + 8)1.37 - 0.35$$
$$y = (-7.71 + 8)1.37 - 0.35$$
$$y = 0.29 \times 1.37 - 0.35$$
$$y = 0.397 - 0.35$$
$$y = 0.047$$

This represents an error of

$$\varepsilon_t = \left| \frac{0.043053 - 0.047}{0.043053} \right| \times 100\% = 9.2\%$$

Hence, the second form is superior because it tends to minimize round-off error.

4.10 ***(a)*** For this case, $x_i = 0$ and $h = x$. Thus, the Taylor series is

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f^{(3)}(0)}{3!}x^3 + \cdots$$

For the exponential function,

$$f(0) = f'(0) = f''(0) = f^{(3)}(0) = 1$$

Substituting these values yields,

$$f(x) = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \cdots$$

which is the Maclaurin series expansion.

(b) The true value is $e^{-1} = 0.367879$ and the step size is $h = x_{i+1} - x_i = 1 - 0.25 = 0.75$. The complete Taylor series to the third-order term is

$$f(x_{i+1}) = e^{-x_i} - e^{-x_i}h + e^{-x_i}\frac{h^2}{2} - e^{-x_i}\frac{h^3}{3!}$$

Zero-order approximation:

$$f(1) = e^{-0.25} = 0.778801$$
$$\varepsilon_t = \left|\frac{0.367879 - 0.778801}{0.367879}\right| \times 100\% = 111.7\%$$

First-order approximation:

$$f(1) = 0.778801 - 0.778801(0.75) = 0.1947$$
$$\varepsilon_t = \left|\frac{0.367879 - 0.1947}{0.367879}\right| \times 100\% = 47.1\%$$

Second-order approximation:

$$f(1) = 0.778801 - 0.778801(0.75) + 0.778801\frac{0.75^2}{2} = 0.413738$$
$$\varepsilon_t = \left|\frac{0.367879 - 0.413738}{0.367879}\right| \times 100\% = 12.5\%$$

Third-order approximation:

$$f(1) = 0.778801 - 0.778801(0.75) + 0.778801\frac{0.75^2}{2} - 0.778801\frac{0.75^3}{6} = 0.358978$$
$$\varepsilon_t = \left|\frac{0.367879 - 0.358978}{0.367879}\right| \times 100\% = 2.42\%$$

4.11 Use the stopping criterion: $\varepsilon_s = 0.5 \times 10^{2-2}\% \; = 0.5\%$

True value: $\cos(\pi/3) = 0.5$

<u>zero order:</u>

$$\cos\left(\frac{\pi}{3}\right) = 1$$
$$\varepsilon_t = \left|\frac{0.5 - 1}{0.5}\right| \times 100\% = 100\%$$

<u>first order:</u>

$$\cos\left(\frac{\pi}{3}\right) = 1 - \frac{(\pi/3)^2}{2} = 0.451689$$

$\varepsilon_t = 9.66\%$ $\qquad\qquad$ $\varepsilon_a = \left|\frac{0.451689 - 1}{0.451689}\right| \times 100\% = 121.4\%$

<u>second order:</u>

$$\cos\left(\frac{\pi}{3}\right) = 0.451689 + \frac{(\pi/3)^4}{24} = 0.501796$$

$\varepsilon_t = 0.359\%$ $\qquad \varepsilon_a = \left|\dfrac{0.501796 - 0.451689}{0.501796}\right| \times 100\% = 9.986\%$

third order:

$$\cos\left(\frac{\pi}{3}\right) = 0.501796 - \frac{(\pi/3)^6}{720} = 0.499965$$

$\varepsilon_t = 0.00709\%$ $\qquad \varepsilon_a = \left|\dfrac{0.499965 - 0.501796}{0.499965}\right| \times 100\% = 0.366\%$

Since the approximate error is below 0.5%, the computation can be terminated.

4.12 Use the stopping criterion: $\varepsilon_s = 0.5 \times 10^{2-2}\% = 0.5\%$

True value: $\sin(\pi/3) = 0.866025\ldots$

zero order:

$$\sin\left(\frac{\pi}{3}\right) = \frac{\pi}{3} = 1.047198$$

$$\varepsilon_t = \left|\frac{0.866025 - 1.047198}{0.866025}\right| \times 100\% = 20.92\%$$

first order:

$$\sin\left(\frac{\pi}{3}\right) = 1.047198 - \frac{(\pi/3)^3}{6} = 0.855801$$

$\varepsilon_t = 1.18\%$ $\qquad \varepsilon_a = \left|\dfrac{0.855801 - 1.047198}{0.855801}\right| \times 100\% = 22.36\%$

second order:

$$\sin\left(\frac{\pi}{3}\right) = 0.855801 + \frac{(\pi/3)^5}{120} = 0.866295$$

$\varepsilon_t = 0.031\%$ $\qquad \varepsilon_a = \left|\dfrac{0.866295 - 0.855801}{0.866295}\right| \times 100\% = 1.211\%$

third order:

$$\sin\left(\frac{\pi}{3}\right) = 0.866295 - \frac{(\pi/3)^7}{5040} = 0.866021$$

$\varepsilon_t = 0.000477\%$ $\qquad \varepsilon_a = \left|\dfrac{0.866021 - 0.866295}{0.866021}\right| \times 100\% = 0.0316\%$

Since the approximate error is below 0.5%, the computation can be terminated.

4.13 The true value is $f(3) = 554$.

zero order:

$$f(3) \cong f(1) = -62 \qquad \varepsilon_t = \left|\frac{554-(-62)}{554}\right| \times 100\% = 111.19\%$$

first order:

$$f'(1) = 75(1)^2 - 12(1) + 7 = 70$$

$$f(3) \cong -62 + 70(2) = 78 \qquad \varepsilon_t = \left|\frac{554-78}{554}\right| \times 100\% = 85.92\%$$

second order:

$$f''(1) = 150(1) - 12 = 138$$

$$f(3) \cong 78 + \frac{138}{2}(2)^2 = 354 \qquad \varepsilon_t = \left|\frac{554-354}{554}\right| \times 100\% = 36.10\%$$

third order:

$$f^{(3)}(1) = 150$$

$$f(3) = 354 + \frac{150}{6}(2)^3 = 554 \qquad \varepsilon_t = \left|\frac{554-554}{554}\right| \times 100\% = 0.0\%$$

Because we are working with a third-order polynomial, the error is zero. This is due to the fact that cubics have zero, fourth, and higher derivatives.

4.14

$$f(x) = ax^2 + bx + c$$
$$f'(x) = 2ax + b$$
$$f''(x) = 2a$$

Substitute these relationships into Eq. (4.11),

$$ax_{i+1}^2 + bx_{i+1} + c = ax_i^2 + bx_i + c + (2ax_i + b)(x_{i+1} - x_i) + \frac{2a}{2!}(x_{i+1}^2 - 2x_{i+1}x_i + x_i^2)$$

Collect terms

$$ax_{i+1}^2 + bx_{i+1} + c = ax_i^2 + 2ax_i(x_{i+1} - x_i) + a(x_{i+1}^2 - 2x_{i+1}x_i + x_i^2) + bx_i + b(x_{i+1} - x_i) + c$$
$$ax_{i+1}^2 + bx_{i+1} + c = ax_i^2 + 2ax_ix_{i+1} - 2ax_i^2 + ax_{i+1}^2 - 2ax_{i+1}x_i + ax_i^2 + bx_i + bx_{i+1} - bx_i + c$$
$$ax_{i+1}^2 + bx_{i+1} + c = (ax_i^2 - 2ax_i^2 + ax_i^2) + ax_{i+1}^2 + (2ax_ix_{i+1} - 2ax_{i+1}x_i) + (bx_i - bx_i) + bx_{i+1} + c$$
$$ax_{i+1}^2 + bx_{i+1} + c = ax_{i+1}^2 + bx_{i+1} + c$$

4.15 The true value is $\ln(2) = 0.693147...$

zero order:

$$f(2) \cong f(1) = 0 \qquad \varepsilon_t = \left|\frac{0.693147 - 0}{0.693147}\right| 100\% = 100\%$$

first order:

$$f'(x) = \frac{1}{x} \qquad f'(1) = 1$$

$$f(2) \cong 0 + 1(1) = 1 \qquad \varepsilon_t = \left|\frac{0.693147 - 1}{0.693147}\right| 100\% = 44.27\%$$

second order:

$$f''(x) = -\frac{1}{x^2} \qquad f''(1) = -1$$

$$f(2) = 1 - 1\frac{1^2}{2} = 0.5 \qquad \varepsilon_t = \left|\frac{0.693147 - 0.5}{0.693147}\right| 100\% = 27.87\%$$

third order:

$$f^{(3)}(x) = \frac{2}{x^3} \qquad f''(1) = 2$$

$$f(2) \cong 0.5 + 2\frac{1^3}{6} = 0.833333 \qquad \varepsilon_t = \left|\frac{0.693147 - 0.833333}{0.693147}\right| 100\% = 20.22\%$$

fourth order:

$$f^{(4)}(x) = -\frac{6}{x^4} \qquad f^{(4)}(1) = -6$$

$$f(2) \cong 0.833333 - 6\frac{1^4}{24} = 0.583333 \qquad \varepsilon_t = \left|\frac{0.693147 - 0.583333}{0.693147}\right| 100\% = 15.84\%$$

The series is converging, but at a slow rate. A smaller step size is required to obtain more rapid convergence.

4.16 The first derivative of the function at $x = 2$ can be evaluated as

$$f'(2) = 75(2)^2 - 12(2) + 7 = 283$$

The points needed to form the finite divided differences can be computed as

$$x_{i-1} = 1.75 \qquad f(x_{i-1}) = 39.85938$$
$$x_i = 2.0 \qquad f(x_i) = 102$$
$$x_{i+1} = 2.25 \qquad f(x_{i+1}) = 182.1406$$

forward:

$$f'(2)=\frac{182.1406-102}{0.25}=320.5625 \qquad |E_t|=|283-320.5625|=37.5625$$

backward:

$$f'(2)=\frac{102-39.85938}{0.25}=248.5625 \qquad |E_t|=|283-248.5625|=34.4375$$

centered:

$$f'(2)=\frac{182.1406-39.85938}{0.5}=284.5625 \quad E_t=283-284.5625=-1.5625$$

Both the forward and backward differences should have errors approximately equal to

$$|E_t|\approx\frac{f''(x_i)}{2}h$$

The second derivative can be evaluated as

$$f''(2)=150(2)-12=288$$

Therefore,

$$|E_t|\approx\frac{288}{2}0.25=36$$

which is similar in magnitude to the computed errors. For the central difference,

$$E_t\approx-\frac{f^{(3)}(x_i)}{6}h^2$$

The third derivative of the function is 150 and

$$E_t\approx-\frac{150}{6}(0.25)^2=-1.5625$$

which is exact. This occurs because the underlying function is a cubic equation that has zero fourth and higher derivatives.

4.17 The second derivative of the function at $x = 2$ can be evaluated as

$$f'(2)=150(2)-12=288$$

For $h = 0.2$,

$$f''(2) = \frac{164.56 - 2(102) + 50.96}{(0.2)^2} = 288$$

For $h = 0.1$,

$$f''(2) = \frac{131.765 - 2(102) + 75.115}{(0.1)^2} = 288$$

Both are exact because the errors are a function of fourth and higher derivatives which are zero for a 3rd-order polynomial.

4.18 Use $\varepsilon_s = 0.5 \times 10^{2-2} = 0.5\%$. The true value = $1/(1 - 0.1) = 1.11111\ldots$

zero-order:

$$\frac{1}{1-0.1} \cong 1$$

$$\varepsilon_t = \left|\frac{1.11111-1}{1.11111}\right| 100\% = 10\%$$

first-order:

$$\frac{1}{1-0.1} \cong 1 + 0.1 = 1.1$$

$$\varepsilon_t = \left|\frac{1.11111-1.1}{1.11111}\right| 100\% = 1\%$$

$$\varepsilon_a = \left|\frac{1.1-1}{1.1}\right| 100\% = 9.0909\%$$

second-order:

$$\frac{1}{1-0.1} \cong 1 + 0.1 + 0.01 = 1.11$$

$$\varepsilon_t = \left|\frac{1.11111-1.11}{1.11111}\right| 100\% = 0.1\%$$

$$\varepsilon_a = \left|\frac{1.11-1.1}{1.11}\right| 100\% = 0.9009\%$$

third-order:

$$\frac{1}{1-0.1} \cong 1 + 0.1 + 0.01 + 0.001 = 1.111$$

$$\varepsilon_t = \left|\frac{1.11111-1.111}{1.11111}\right| 100\% = 0.01\%$$

$$\varepsilon_a = \left|\frac{1.111-1.11}{1.111}\right| 100\% = 0.090009\%$$

The approximate error has fallen below 0.5%, so the computation can be terminated.

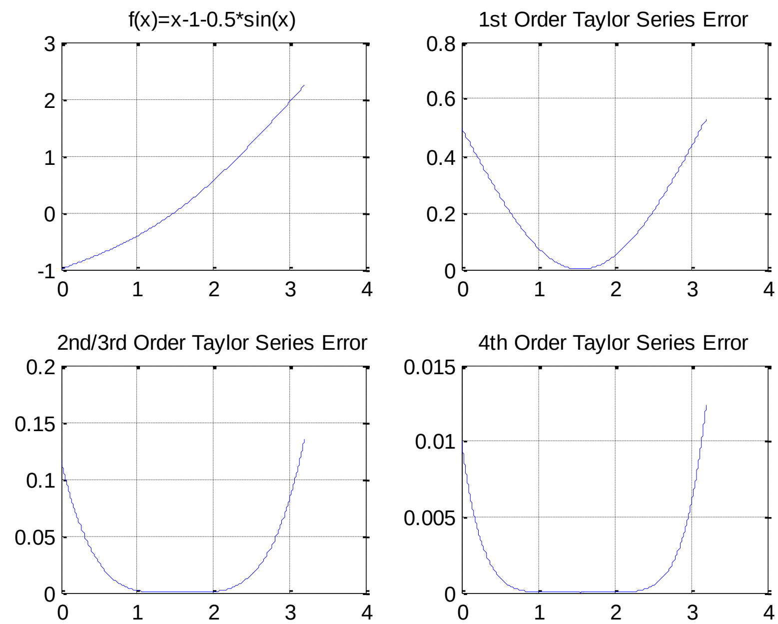
4.19 Here is the function and its derivatives

$$f(x) = x - 1 - \frac{1}{2}\sin x$$
$$f'(x) = 1 - \frac{1}{2}\cos x$$
$$f''(x) = \frac{1}{2}\sin x$$
$$f^{(3)}(x) = \frac{1}{2}\cos x$$
$$f^{(4)}(x) = -\frac{1}{2}\sin x$$

Using the Taylor Series expansion, we obtain the following 1st, 2nd, 3rd, and 4th order Taylor Series functions shown below in the MATLAB program–`f1`, `f2`, and `f4`. Note the 2nd and 3rd order Taylor Series functions are the same.

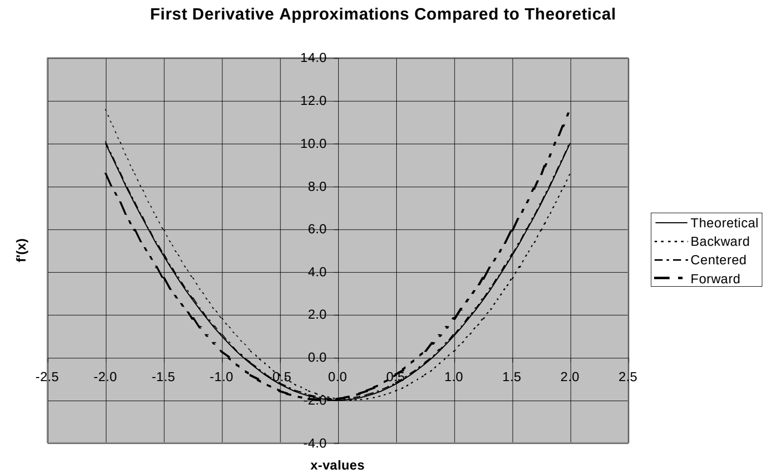
From the plots below, we see that the answer is the 4th Order Taylor Series expansion.

```
x=0:0.001:3.2;
f=x-1-0.5*sin(x);
subplot(2,2,1);
plot(x,f);grid;title('f(x)=x-1-0.5*sin(x)');hold on

f1=x-1.5;
e1=abs(f-f1);   %Calculates the absolute value of the
difference/error
subplot(2,2,2);
plot(x,e1);grid;title('1st Order Taylor Series Error');

f2=x-1.5+0.25.*((x-0.5*pi).^2);
e2=abs(f-f2);
subplot(2,2,3);
plot(x,e2);grid;title('2nd/3rd Order Taylor Series Error');

f4=x-1.5+0.25.*((x-0.5*pi).^2)-(1/48)*((x-0.5*pi).^4);
e4=abs(f4-f);
subplot(2,2,4);
plot(x,e4);grid;title('4th Order Taylor Series Error');hold
off
```

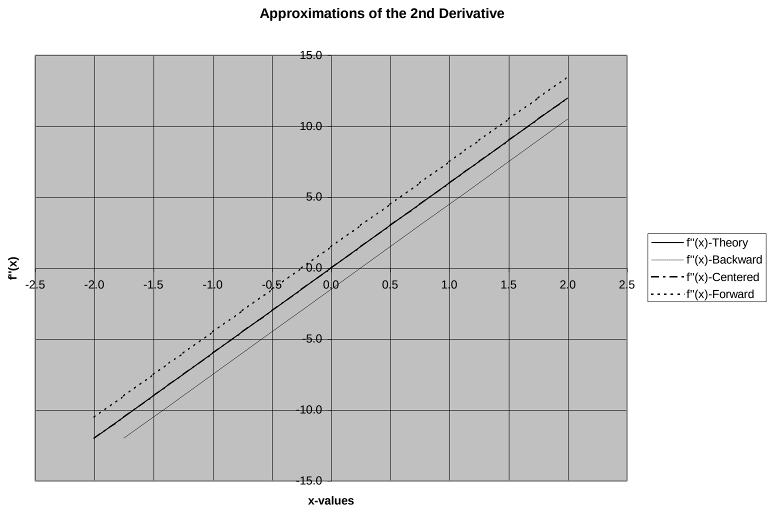


4.20

x	f(x)	f(x−1)	f(x+1)	f′(x)-Theory	f′(x)-Back	f′(x)-Cent	f′(x)-Forw
−2.000	0.000	−2.891	2.141	10.000	11.563	10.063	8.563
−1.750	2.141	0.000	3.625	7.188	8.563	7.250	5.938
−1.500	3.625	2.141	4.547	4.750	5.938	4.813	3.688
−1.250	4.547	3.625	5.000	2.688	3.688	2.750	1.813
−1.000	5.000	4.547	5.078	1.000	1.813	1.063	0.313
−0.750	5.078	5.000	4.875	−0.313	0.313	−0.250	−0.813
−0.500	4.875	5.078	4.484	−1.250	−0.813	−1.188	−1.563
−0.250	4.484	4.875	4.000	−1.813	−1.563	−1.750	−1.938
0.000	4.000	4.484	3.516	−2.000	−1.938	−1.938	−1.938
0.250	3.516	4.000	3.125	−1.813	−1.938	−1.750	−1.563
0.500	3.125	3.516	2.922	−1.250	−1.563	−1.188	−0.813
0.750	2.922	3.125	3.000	−0.313	−0.813	−0.250	0.313
1.000	3.000	2.922	3.453	1.000	0.313	1.063	1.813
1.250	3.453	3.000	4.375	2.688	1.813	2.750	3.688
1.500	4.375	3.453	5.859	4.750	3.688	4.813	5.938
1.750	5.859	4.375	8.000	7.188	5.938	7.250	8.563
2.000	8.000	5.859	10.891	10.000	8.563	10.063	11.563



x	f(x)	f(x−1)	f(x+1)	f(x−2)	f(x+2)	f″(x)-Theory	f″(x)-Back	f″(x)-Cent	f″(x)-Forw
−2.000	0.000	−2.891	2.141	−6.625	3.625	−12.000	−13.500	−12.000	−10.500
−1.750	2.141	0.000	3.625	−2.891	4.547	−10.500	−12.000	−10.500	−9.000
−1.500	3.625	2.141	4.547	0.000	5.000	−9.000	−10.500	−9.000	−7.500
−1.250	4.547	3.625	5.000	2.141	5.078	−7.500	−9.000	−7.500	−6.000
−1.000	5.000	4.547	5.078	3.625	4.875	−6.000	−7.500	-6.000	−4.500
−0.750	5.078	5.000	4.875	4.547	4.484	−4.500	−6.000	−4.500	−3.000
−0.500	4.875	5.078	4.484	5.000	4.000	−3.000	−4.500	−3.000	−1.500
−0.250	4.484	4.875	4.000	5.078	3.516	−1.500	−3.000	−1.500	0.000
0.000	4.000	4.484	3.516	4.875	3.125	0.000	−1.500	0.000	1.500
0.250	3.516	4.000	3.125	4.484	2.922	1.500	0.000	1.500	3.000
0.500	3.125	3.516	2.922	4.000	3.000	3.000	1.500	3.000	4.500
0.750	2.922	3.125	3.000	3.516	3.453	4.500	3.000	4.500	6.000
1.000	3.000	2.922	3.453	3.125	4.375	6.000	4.500	6.000	7.500
1.250	3.453	3.000	4.375	2.922	5.859	7.500	6.000	7.500	9.000
1.500	4.375	3.453	5.859	3.000	8.000	9.000	7.500	9.000	10.500
1.750	5.859	4.375	8.000	3.453	10.891	10.500	9.000	10.500	12.000
2.000	8.000	5.859	10.891	4.375	14.625	12.000	10.500	12.000	13.500



4.21 We want to find the value of *h* that results in an optimum of

$$E = \frac{\varepsilon}{h} + \frac{h^2 M}{6}$$

Differentiating gives

$$\frac{dE}{dh} = -\frac{\varepsilon}{h^2} + \frac{M}{3} h$$

This result can be set to zero and solved for

$$h^3 = \frac{3\varepsilon}{M}$$

Taking the cube root gives

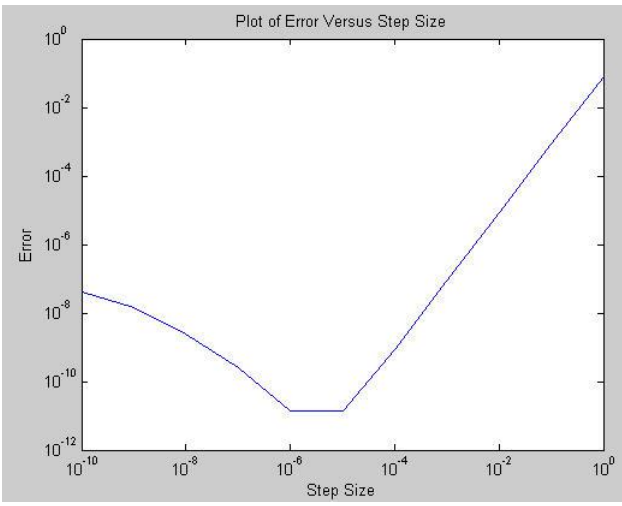
$$h_{opt} = \sqrt[3]{\frac{3\varepsilon}{M}}$$

4.22 Using the same function as in Example 4.5:

```
ff=@(x) cos(x);
df=@(x) -sin(x);
diffex(ff,df,pi/6,11)
```

Results:

```
   step size   finite difference    true error
  1.0000000000 -0.42073549240395  0.0792645075961
  0.1000000000 -0.49916708323414  0.0008329167659
  0.0100000000 -0.49999166670833  0.0000083332917
  0.0010000000 -0.49999991666672  0.0000000833333
  0.0001000000 -0.49999999916672  0.0000000008333
  0.0000100000 -0.49999999998662  0.0000000000134
  0.0000010000 -0.50000000001438  0.0000000000144
  0.0000001000 -0.49999999973682  0.0000000002632
  0.0000000100 -0.50000000251238  0.0000000025124
  0.0000000010 -0.49999998585903  0.0000000141410
  0.0000000001 -0.50000004137019  0.0000000413702
```

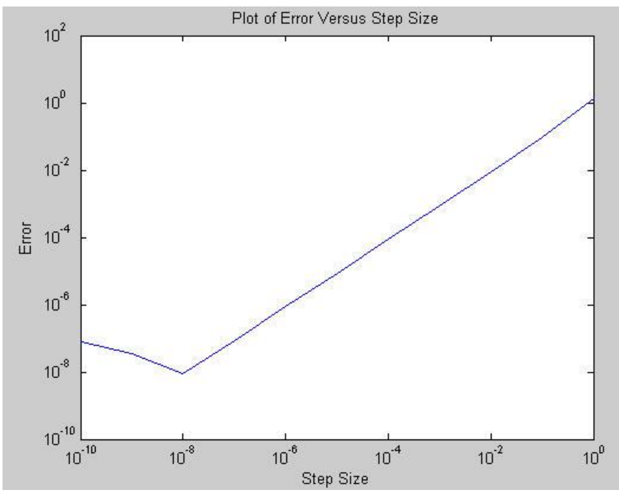


4.23 First, we must develop a function like the one in Example 4.5, but to evaluate a forward difference:

```
function prob0423(func,dfunc,x,n)
format long
dftrue=dfunc(x);
h=1;
H(1)=h;
D(1)=(func(x+h)-func(x))/h;
E(1)=abs(dftrue-D(1));
for i=2:n
  h=h/10;
  H(i)=h;
  D(i)=(func(x+h)-func(x))/h;
  E(i)=abs(dftrue-D(i));
end
L=[H' D' E']';
fprintf('   step size   finite difference    true error\n');
fprintf('%14.10f %16.14f %16.13f\n',L);
loglog(H,E),xlabel('Step Size'),ylabel('Error')
title('Plot of Error Versus Step Size')
format short
```

We can then use it to evaluate the same case as in Example 4.5:

```
>> ff=@(x) -0.1*x^4-0.15*x^3-0.5*x^2-0.25*x+1.2;
>> df=@(x) -0.4*x^3-0.45*x^2-x-0.25;
>> prob0423(ff,df,0.5,11)
   step size    finite difference     true error
  1.0000000000 -2.23750000000000  1.3250000000000
  0.1000000000 -1.00360000000000  0.0911000000000
  0.0100000000 -0.92128509999999  0.0087851000000
  0.0010000000 -0.91337535009994  0.0008753500999
  0.0001000000 -0.91258750349987  0.0000875034999
  0.0000100000 -0.91250875002835  0.0000087500284
  0.0000010000 -0.91250087497219  0.0000008749722
  0.0000001000 -0.91250008660282  0.0000000866028
  0.0000000100 -0.91250000888721  0.0000000088872
  0.0000000010 -0.91249996447829  0.0000000355217
  0.0000000001 -0.91250007550059  0.0000000755006
```



4.24 First, we can evaluate the exact values using the standard formula with double-precision arithmetic as

$$\begin{matrix} x_1 \\ x_2 \end{matrix} = \frac{5{,}000.002 \pm \sqrt{(5{,}000.002)^2 - 4(1)10}}{2(1)} = \begin{matrix} 5{,}000 \\ 0.002 \end{matrix}$$

Secondly, we can then determine the square root term with 5-digit arithmetic and chopping

$$\sqrt{5{,}000.002^2 - 4(1)10} = \sqrt{5000.0^2 - 40} = \sqrt{25000000 - 40} = \sqrt{2.5000 \times 10^8 - 0.0000004 \times 10^8}$$

$$= \sqrt{2.5000 \times 10^8 - 0.00000 \times 10^8} = \sqrt{2.5000 \times 10^8} = 5000 = 5000$$

First equation

$$x_1 = \frac{5000 + 5000}{2} = 5000$$
$$x_2 = \frac{5000 - 5000}{2} = 0$$

Thus, although the first root is exact, the second is incorrect, due primarily to subtractive cancellation.

<u>Second equation:</u>

$$x_1 = \frac{-2(10)}{-5{,}000.0 + 5000} = \infty$$

$$x_2 = \frac{-2(10)}{-5{,}000.0 - 5{,}000} = \frac{-20}{-10{,}000} = 0.002$$

For this case, the second root is exact, whereas, the first is incorrect. Again, the culprit is the subtraction of two nearly equal numbers.

4.25 It is useful to first express the series as a summation:

$$\cos x = \sum_{i=0}^{\infty} \frac{(-1)^i x^{2i}}{(2i)!}$$

A MATLAB function can be then written as

```
function [fx,ea,iter] = cosine(x,es,maxit)
% Maclaurin series of cosine
% [fx,ea,iter] = IterMeth(x,es,maxit)
% input:
% x = value at which series evaluated
% es = stopping criterion (default = 0.0001)
% maxit = maximum iterations (default = 50)
% output:
% fx = estimated value
% ea = approximate relative error (%)
% iter = number of iterations

% defaults:
if nargin<2|isempty(es),es=0.0001;end
if nargin<3|isempty(maxit),maxit=50;end
% initialization
iter = 0; sol = 0; ea = 100;
% iterative calculation
while (1)
  solold = sol;
  sol = sol + (-1)^iter * x^(2*iter) / factorial(2*iter);
  iter = iter + 1;
```

```
  if sol~=0
    ea=abs((sol - solold)/sol)*100;
  end
  if ea<=es | iter>=maxit,break,end
end
fx = sol;
```

The following script can be used to check that the function yields the correct result,

```
clc, format compact
[ser, ea, i] = cosine(pi/3)
[ser, ea, i] = cosine(7*pi/3)

ser =
    0.5000
ea =
   8.7408e-05
i =
     6
ser =
    0.5000
ea =
   6.7791e-05
i =
    16
```

As expected, both cases yield the correct result of $\cos(\pi/3) = \cos(7\pi/3) = 0.5$. However, more iterations (6 versus 16) are required in the second case to obtain the correct result with the desired approximate absolute error (ε_a). This is because the second angle is farther away from zero and, hence, more terms in the series are required to attain the desired accuracy.

4.26 It is useful to first express the series as a summation:

$$\sin x = \sum_{i=0}^{\infty} \frac{(-1)^i x^{2i+1}}{2(i+1)!}$$

A MATLAB function can be then written as

```
function [fx,ea,iter] = sine(x,es,maxit)
% Maclaurin series of cosine
% [fx,ea,iter] = IterMeth(x,es,maxit)
% input:
% x = value at which series evaluated
% es = stopping criterion (default = 0.0001)
% maxit = maximum iterations (default = 50)
% output:
% fx = estimated value
% ea = approximate relative error (%)
% iter = number of iterations
```

```
% defaults:
if nargin<2||isempty(es),es=0.0001;end
if nargin<3||isempty(maxit),maxit=50;end
% initialization
iter = 0; sol = 0; ea = 100;
% iterative calculation
while (1)
  solold = sol;
  sol = sol + (-1)^iter * x^(2*iter+1) / factorial(2*iter+1);
  iter = iter + 1;
  if sol~=0
    ea=abs((sol - solold)/sol)*100;
  end
  if ea<=es || iter>=maxit,break,end
end
fx = sol;
```

The following script can be used to check that the function yields the correct result,

```
clc, format compact
[ser, ea, i] = sine(pi/3)
[ser, ea, i] = sine(7*pi/3)
```

```
ser =
    0.8660
ea =
   4.8043e-06
i =
     6
ser =
    0.8660
ea =
   9.2549e-06
i =
    16
```

As expected, both cases yield the correct result of $\sin(\pi/3) = \sin(7\pi/3) = 0.866$. However, more iterations (6 versus 16) are required in the second case to obtain the correct result with the desired approximate absolute error (ε_a). This is because the second angle is farther away from zero and, hence, more terms in the series are required.

4.27 The Taylor series can be written as

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f^{(3)}(x_i)}{3!}h^3 + \frac{f^{(4)}(x_i)}{4!}h^4 + \frac{f^{(5)}(x_i)}{5!}h^5 + \frac{f^{(6)}(x_i)}{6!}h^6 + \cdots$$

The Maclaurin series corresponds to substituting $x_{i+1} = x$, $x_i = 0$, and $h = x_{i+1} - x_i = x$,

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f^{(3)}(0)}{3!}x^3 + \frac{f^{(4)}(0)}{4!}x^4 + \frac{f^{(5)}(0)}{5!}x^5 + \frac{f^{(6)}(0)}{6!}x^6 + \cdots$$

For $f(x) = \cos x$,

$$\cos(x) = \cos(0) + \cos'(0)x + \frac{\cos''(0)}{2!}x^2 + \frac{\cos^{(3)}(0)}{3!}x^3 + \frac{\cos^{(4)}(0)}{4!}x^4 + \frac{\cos^{(5)}(0)}{5!}x^5 + \frac{\cos^{(6)}(0)}{6!}x^6 + \cdots$$

Recognize that because $\cos'(x) = -\sin(x)$ and $\sin'(x) = \cos(x)$,

$$\begin{aligned}
&\cos(0) = 1\\
&\cos'(0) = -\sin(0) = 0\\
&\cos''(0) = -\cos(0) = -1\\
&\cos^{(3)}\ 0 = \sin(0) = 0\\
&\cos^{(4)}(0) = \cos(0) = 1\\
&\cos^{(5)}(0) = -\sin(0) = 0\\
&\cos^{(6)}(0) = -\cos(0) = -1
\end{aligned}$$

Substituting these values gives the final result

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots$$

4.28 (a)

$$\begin{aligned}
\arctan x &\cong \frac{(-1)^0}{2(0)+1}x^{2(0)+1} + \frac{(-1)^1}{2(1)+1}x^{2(1)+1} + \frac{(-1)^2}{2(2)+1}x^{2(2)+1} + \frac{(-1)^3}{2(3)+1}x^{2(3)+1}\\
&= x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7}
\end{aligned}$$

(b) Use the stopping criterion: $\varepsilon_s = 0.5\times10^{2-2}\% = 0.5\%$

True value: $\arctan(\pi/6) = 0.482347907$

<u>zero order:</u>

$$\arctan\left(\frac{\pi}{6}\right) \cong \frac{\pi}{6} = 0.523599 \qquad \varepsilon_t = \left|\frac{0.482348 - 0.523599}{0.482348}\right| \times 100\% = 8.55\%$$

<u>first order:</u>

$$\arctan\left(\frac{\pi}{6}\right) \cong \frac{\pi}{6} - \frac{(\pi/6)^3}{3} = 0.523599 - 0.04785 = 0.47575$$

$$\varepsilon_t = 1.37\% \qquad \varepsilon_a = \left|\frac{0.47575 - 0.523599}{0.47575}\right| \times 100\% = 10.06\%$$

<u>second order:</u>

$$\arctan\left(\frac{\pi}{6}\right) \cong \frac{\pi}{6} - \frac{(\pi/6)^3}{3} + \frac{(\pi/6)^5}{5} = 0.47575 + 0.007871 = 0.48362$$

$\varepsilon_t = 0.26\%$ $\quad\quad$ $\varepsilon_a = \left|\frac{0.48362 - 0.47575}{0.48362}\right| \times 100\% = 1.63\%$

<u>third order:</u>

$$\arctan\left(\frac{\pi}{6}\right) \cong \frac{\pi}{6} - \frac{(\pi/6)^3}{3} + \frac{(\pi/6)^5}{5} - \frac{(\pi/6)^7}{7} = 0.48362 - 0.00154 = 0.482079$$

$\varepsilon_t = 0.06\%$ $\quad\quad$ $\varepsilon_a = \left|\frac{0.482079 - 0.48362}{0.482079}\right| \times 100\% = 0.32\%$

Since the approximate error is below 0.5%, the computation can be terminated.