Assignment 2 of CISC 2002

ZHANG Huakang/DB92760

February 16, 2021

1

```
function y = Assignment_2_1_f(x)
    y=816*x^3-3835*x^2+6000*x-3125;
end
```

Listing 1: Function

```
function y = Assignment_2_1_derivative(x)
    y=816*3*x^2-3835*2*x+6000;
end
```

Listing 2: Derivative

1.1

```
clear
a=1;
b=2;
m=(a+b)/2;
n=1;
fprintf('%2dth, a=%9.8f, b=%9.8f, error<%9.8f, m=%9.8f\n',n,a,b,a-b,m)
while (b-a)>10^(-6)
    if Assignment_2_1_f(m)==0
        break
    end
    if sign(Assignment_2_1_f(m))==sign(Assignment_2_1_f(a))
        a=m;
    elseif sign(Assignment_2_1_f(m))==sign(Assignment_2_1_f(b))
        b=m;
    end
    m=(a+b)/2;
    n=n+1;
    fprintf('%2dth, a=%9.8f, b=%9.8f, error<%9.8f, m=%9.8f\n',n,a,b,a-b,m)
end
```

Listing 3: Bisection

```
1th, a=1.00000000, b=2.00000000, error<-1.00000000, m=1.50000000
2th, a=1.00000000, b=1.50000000, error<-0.50000000, m=1.25000000
3th, a=1.25000000, b=1.50000000, error<-0.25000000, m=1.37500000
4th, a=1.37500000, b=1.50000000, error<-0.12500000, m=1.43750000
5th, a=1.43750000, b=1.50000000, error<-0.06250000, m=1.46875000
6th, a=1.46875000, b=1.50000000, error<-0.03125000, m=1.48437500
7th, a=1.46875000, b=1.48437500, error<-0.01562500, m=1.47656250
8th, a=1.46875000, b=1.47656250, error<-0.00781250, m=1.47265625
9th, a=1.46875000, b=1.47265625, error<-0.00390625, m=1.47070313
10th, a=1.46875000, b=1.47070313, error<-0.00195313, m=1.46972656
11th, a=1.46972656, b=1.47070313, error<-0.00097656, m=1.47021484
12th, a=1.47021484, b=1.47070313, error<-0.00048828, m=1.47045898
13th, a=1.47045898, b=1.47070313, error<-0.00024414, m=1.47058105
14th, a=1.47058105, b=1.47070313, error<-0.00012207, m=1.47064209
15th, a=1.47058105, b=1.47064209, error<-0.00006104, m=1.47061157
16th, a=1.47058105, b=1.47061157, error<-0.00003052, m=1.47059631
17th, a=1.47058105, b=1.47059631, error<-0.00001526, m=1.47058868
18th, a=1.47058105, b=1.47058868, error<-0.00000763, m=1.47058487
19th, a=1.47058487, b=1.47058868, error<-0.00000381, m=1.47058678
20th, a=1.47058678, b=1.47058868, error<-0.00000191, m=1.47058773
21th, a=1.47058773, b=1.47058868, error<-0.00000095, m=1.47058821
```

Listing 4: Bisection Output

1.2

```
clear
x0=1.6;
x1=x0-(Assignment_2_1_f(x0)/Assignment_2_1_derivative(x0));
n=0;
fprintf('x(%1d)=%9.8f\n',n,x0)
while abs(x1-x0)>10^(-8)
    x0=x1;
    x1=x0-(Assignment_2_1_f(x0)/Assignment_2_1_derivative(x0));
    n=n+1;
    fprintf('x(%1d)=%9.8f\n',n,x0)
end
```

Listing 5: Newton's Method

```
x(0)=1.60000000
x(1)=1.54843750
x(2)=1.56342463
x(3)=1.56250093
x(4)=1.56250000
```

Listing 6: Newton's Method Output

2

```
function y = Assignment_2_2_f(x)
     y=x^2-2;
end
```

Listing 7: Function

```
x_0=1;
x_1=2;
e_0=sqrt(2)-x_0;
e_1=sqrt(2)-x_1;
p=(1+sqrt(5))/2;
n=0;
while abs(e_1)>10^(-10)
    t=x_1;
    x_1= x_1-((x_1-x_0)*Assignment_2_2_f(x_1))/(Assignment_2_2_f(
    x_1)-Assignment_2_2_f(x_0));
    x_0=t;
    e_0=e_1;
    e_1=sqrt(2)-x_1;
    answer=abs(e_1/(e_0^p));
    n=n+1;
    fprintf('%2d-th loop,x=%9.8f, %9.8f\n',n,x_1,answer)
end
```

Listing 8: Matlab Code

```
 1-th loop,x=1.33333333, 0.19215339
 2-th loop,x=1.40000000, 0.83147865
 3-th loop,x=1.41463415, 0.41005503
 4-th loop,x=1.41421144, 0.61638883
 5-th loop,x=1.41421356, 0.47672494
 6-th loop,x=1.41421356, 0.52332385
```

Listing 9: Output

3

When $\epsilon_n = \frac{1}{10^n}$

$$
\begin{aligned}
\epsilon_n =& \frac{1}{10^n} \\
=& \frac{1}{10}\epsilon_{n-1} \\
r - \epsilon_n =& r - \frac{1}{10}\epsilon_{n-1} \\
=& r - \epsilon_{n-1} + \frac{9}{10}\epsilon_{n-1} \\
\frac{r - \epsilon_n}{r - \epsilon_{n-1}} =& 1 + \frac{\frac{9}{10}\epsilon_{n-1}}{r - \epsilon_{n-1}} \\
=& 1 + \frac{\frac{9}{10}\epsilon_{n-1}}{0 - \epsilon_{n-1}} \\
=& 1 - \frac{9}{10} \\
=& \frac{1}{10} \\
\lim_{n\to\infty} |\frac{r - \epsilon_n}{r - \epsilon_{n-1}}| =& \lim_{n\to\infty} |\frac{1}{10}| \\
=& \frac{1}{10}
\end{aligned}
$$

Therefore, the rate of convergence $\mu = \frac{1}{10}$**, and the order of convergence** $p = 1$

When $\epsilon_n = \frac{1}{100^n}$

$$\begin{aligned}
\epsilon_n =& \frac{1}{100^n} \\
=& \frac{1}{100}\epsilon_{n-1} \\
r - \epsilon_n =& r - \epsilon_{n-1} + \frac{99}{100}\epsilon_{n-1} \\
\frac{r - \epsilon_n}{r - \epsilon_{n-1}} =& 1 + \frac{\frac{99}{100}\epsilon_{n-1}}{r - \epsilon_{n-1}} \\
=& 1 - \frac{99}{100} \\
=& \frac{1}{100} \\
\lim_{n\to\infty} |\frac{r - \epsilon_n}{r - \epsilon_{n-1}}| =& \lim_{n\to\infty} |\frac{1}{100}| \\
=& \frac{1}{100}
\end{aligned}$$

Therefore, the rate of convergence $\mu = \frac{1}{100}$**, and the order of convergence** $p = 1$

When $\epsilon_n = \frac{1}{2^{w^n}}$

$$\begin{aligned}
\epsilon_n =& \frac{1}{2^{2^n}} \\
=& (\frac{1}{2^{2^{n-1}}})^2 \\
=& \epsilon_{n-1}^2 \\
r - \epsilon_n =& r - \epsilon_{n-1}{}^2 \\
\frac{r - \epsilon_n}{r - \epsilon_{n-1}} =& \frac{r - \epsilon_{n-1}^2}{r - \epsilon_{n-1}} \\
=& \frac{-\epsilon_{n-1}^2}{-\epsilon_{n-1}} \\
=& \epsilon_{n-1} \\
\lim_{n\to\infty} |\frac{r - \epsilon_n}{r - \epsilon_{n-1}}| =& \lim_{n\to\infty} |\epsilon_{n-1}| \\
=& 0
\end{aligned}$$

Therefore, the rate of convergence $\mu = 0$**, and the order of convergence** $p = 1$

```
clear
n=1:1:9;
y1=1./(10.^n);
y2=1./(100.^n);
y3=1./(2.^(2.^n));
semilogy(n,y1,n,y2,n,y3)
legend('$\frac{1}{10^n}$','$\frac{1}{100^n}$','$\frac{1}{2^{2^n}}$'
    ,'Location','southwest','Interpreter','latex','FontSize',19)
```

Listing 10: Matlab Code

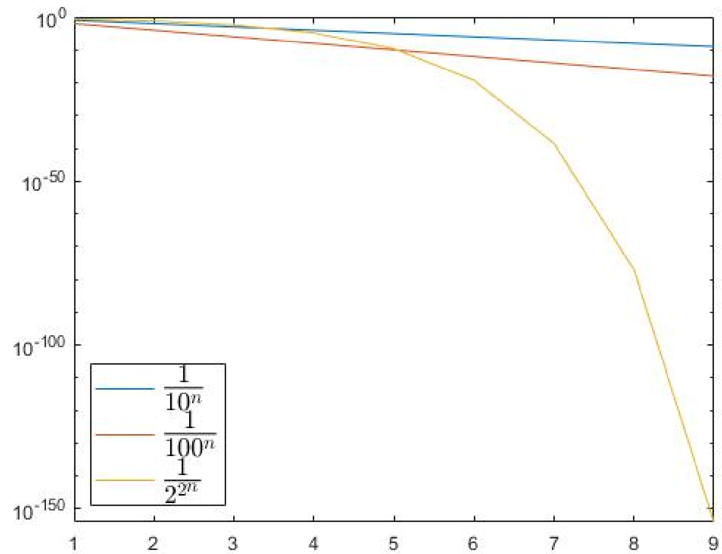

Figure 1: Output