Practical 2 : Newton's Method

1

Figure 1:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Pseudo code for Newtons Method(from Cheney)

Figure 2:

procedure *Newton*$(f,\ f',\ x,\ nmax,\ \varepsilon,\ \delta)$
integer $n,\ nmax$; **real** $x, fx, fp, \varepsilon, \delta$
external function $f,\ f'$
$fx \leftarrow f(x)$
output $0, x, fx$
for $n = 1$ **to** $nmax$ **do**
  $fp \leftarrow f'(x)$
  if $|fp| < \delta$ **then**
    output "small derivative"
    return
  end if
  $d \leftarrow fx/fp$
  $x \leftarrow x - d$
  $fx \leftarrow f(x)$
  output n, x, fx
  if $|d| < \varepsilon$ **then**
    output "convergence"
    return
  end if
end for
end procedure *Newton*

2

```
kill(all);
```

(%o0) *done*

```
→   newton(g, x0, e, d, Nmax):=block
    (
        [p, x1, i],
        define(f(x), g),
        define(df(x), diff(f(x), x)),
        for i:1 thru Nmax do
        (
            if(abs(df(x0)) < d) then (return("small derivative")),
            p:f(x0)/df(x0),
            x1:x0−p,
            print(i, "      ", x1, "          ", f(x1)),
            if(abs(p)<e) then (return("convergence")),
            x0:x1
        )
    );
```

(%o1) newton$(g,x0,e,d,Nmax):=$block$([p,x1,i],$ define$(f(x),g),$define$\left(df(x),\frac{d}{d\,x}f(x)\right),$for i thru $Nmax$ do $($if $|df(x0)|<d$ then return$($*small derivative*$)$ $,p:\frac{f(x0)}{df(x0)},x1:x0-p,$ print$(i,\quad ,x1,\quad ,f(x1)),$if $|p|<e$ then return$($*convergence*$)$ $,x0:x1))$

2.1

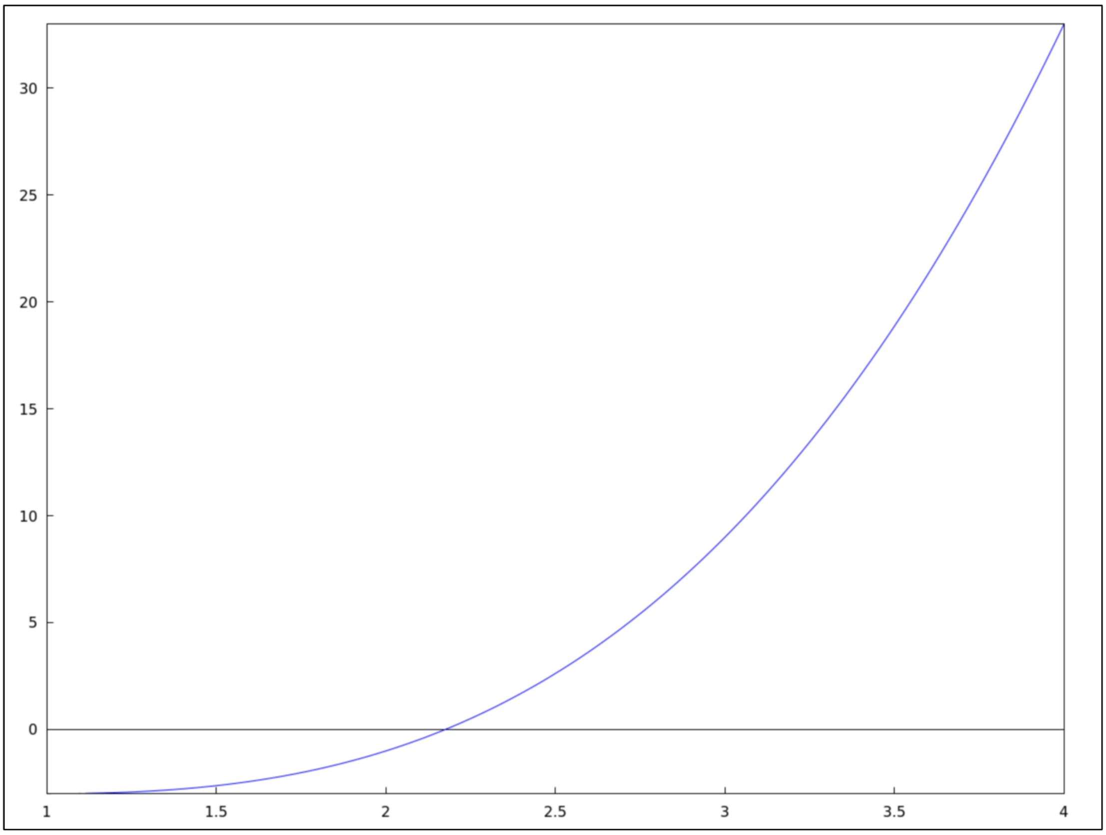
2.1.1

plot

```
(%i3) wxdraw2d(
        xaxis = true, xaxis_type = solid,
        yaxis = true, yaxis_type = solid,

        explicit(x^3−2·x^2+x−3,   x,  1,  4)
      );
```

(%t3)



(%o3)

2.1.2

```
→ newton(x^3−2·x^2+x−3,  3.0,  0.0001,  0.00001,  8);
1       2.4375          2.036865234375
2       2.213032716315109         0.256363850614177
3       2.175554938721488         0.006463361488812325

4       2.174560100666446         4.479068049789703
10^−6
5       2.174559410293312         2.156497203031904
10^−12
(%o2) convergence
```

2.2

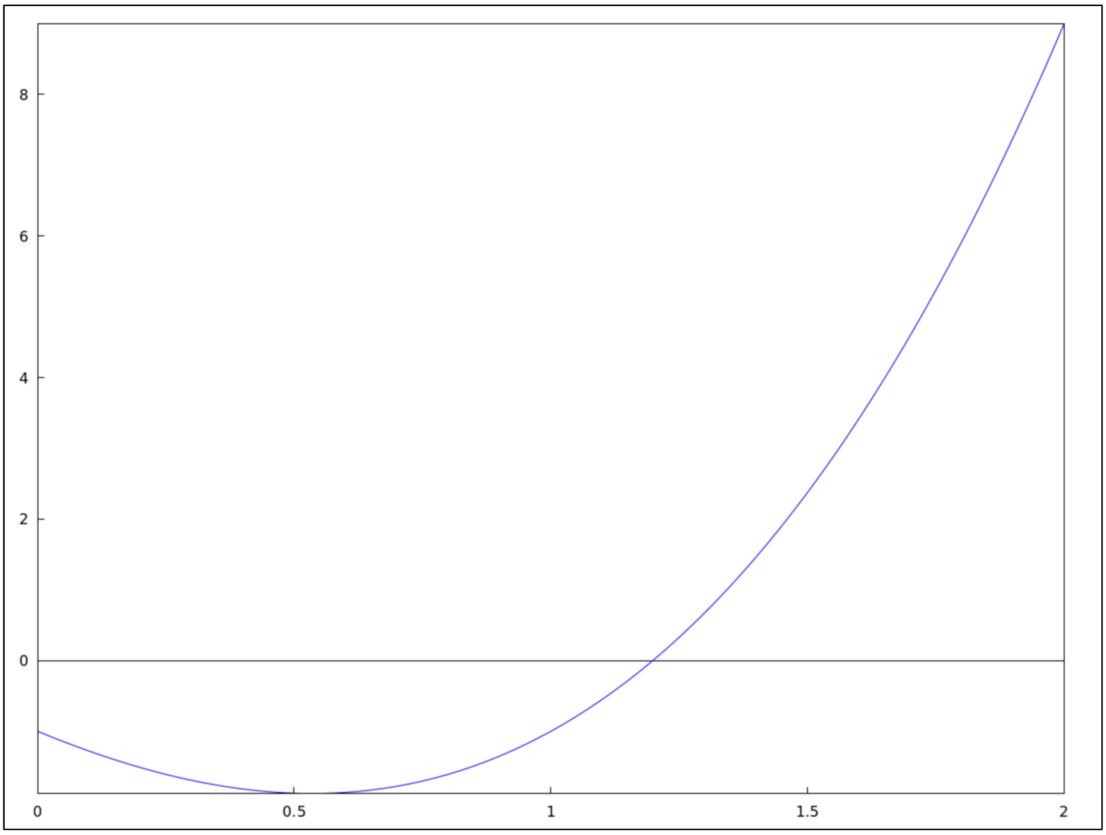
2.2.1

plot

(%i4)
```
wxdraw2d(
    xaxis = true, xaxis_type = solid,
    yaxis = true, yaxis_type = solid,

    explicit(x^3+2·x^2−3·x−1,    x,   0,   2)
);
```

(%t4)



(%o4)

2.2.2

```
newton(x^3+2·x^2−3·x−1,   1.0,  0.0001,  0.00001,   8);
```

1   1.25   0.328125
2   1.200934579439252   0.0137245182414003
3   1.198695841064738   2.806974900448899 10^{-5}
4   1.198691243535371   1.182869358018479 10^{-10}

(%o3) *convergence*

2.2.3

```
→ newton(x^3+2·x^2−3·x−1, 3.0, 0.0001, 0.00001, 8);
1    2.027777777777777    9.4784164951989
2    1.484501152339814    2.225443905080841
3    1.251451723790631    0.3378455240456555
4    1.201058617038117    0.01448501321182349
5    1.19869636259783    3.125391789771115 10^−5
6    1.198691243540016    1.466440302522187 10^−10
(%o4) convergence
```

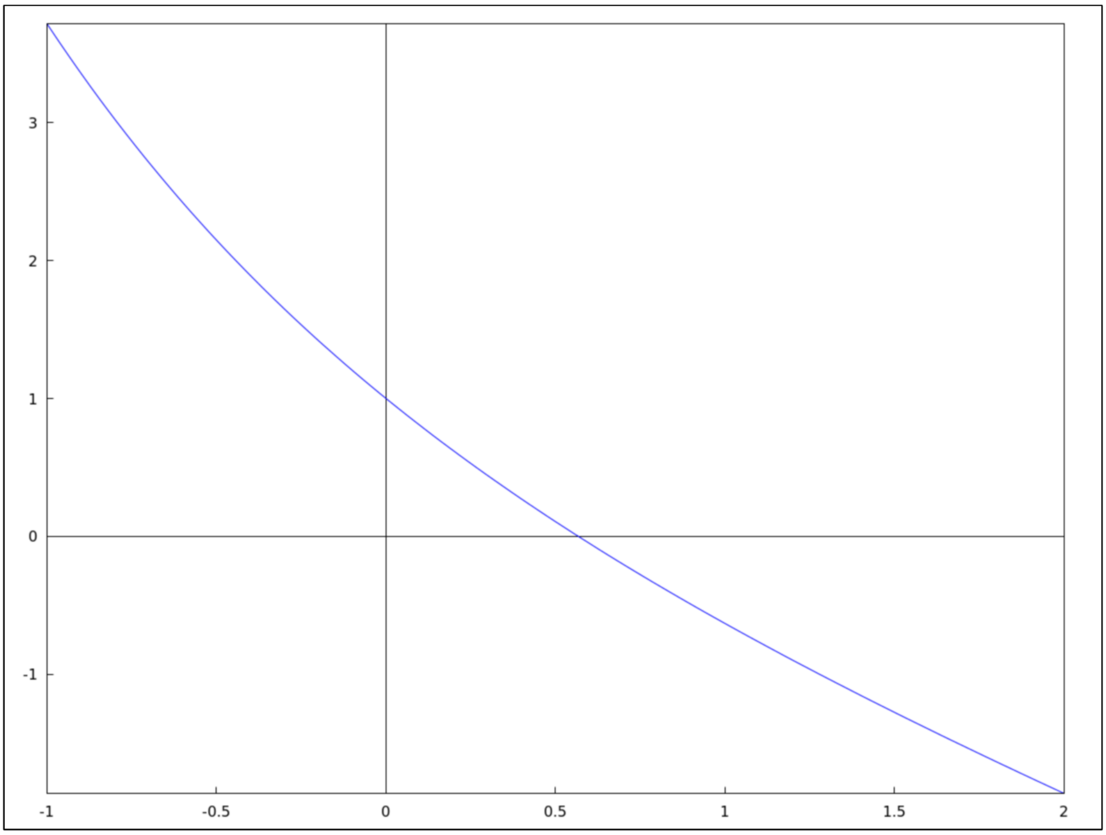
2.3

2.3.1

plot

```
(%i5) wxdraw2d(
    xaxis = true, xaxis_type = solid,
    yaxis = true, yaxis_type = solid,

    explicit(exp(−x)−x,   x,  −1,  2)
);
(%t5)
```



(%o5)

2.3.2

```
→ newton(exp(−x)−x,  0.0,  0.0001,  0.00001,  8);
1    0.5    0.1065306597126334
2    0.5663110031972182
0.001304509806020037
3    0.5671431650348622    1.964804717813351
 −7
10
4    0.5671432904097811    4.440892098500626
 −15
10
(%o5) convergence
```

2.4

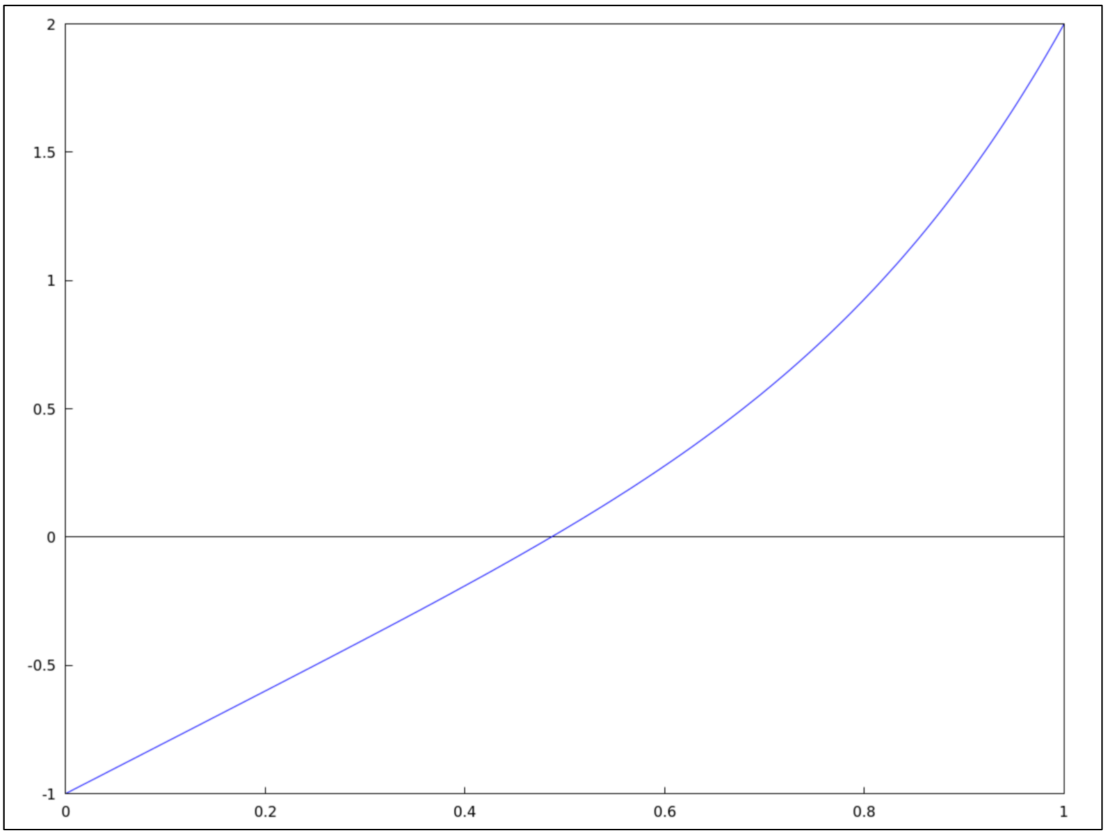
2.4.1

plot

(%i7)
```
wxdraw2d(
    xaxis = true, xaxis_type = solid,
    yaxis = true, yaxis_type = solid,

    explicit(x^5+2·x−1,   x,  0,  1)
);
```

(%t7)



(%o7)

2.4.2

```
newton(x^5+2·x−1,  0.0,  0.0001,  0.00001,  8);
```

1 0.5 0.03125

2 0.4864864864864865 2.221823020570369 10^{-4}

3 0.4863890407290883 1.093078168604155 10^{-8}

(%o6) *convergence*

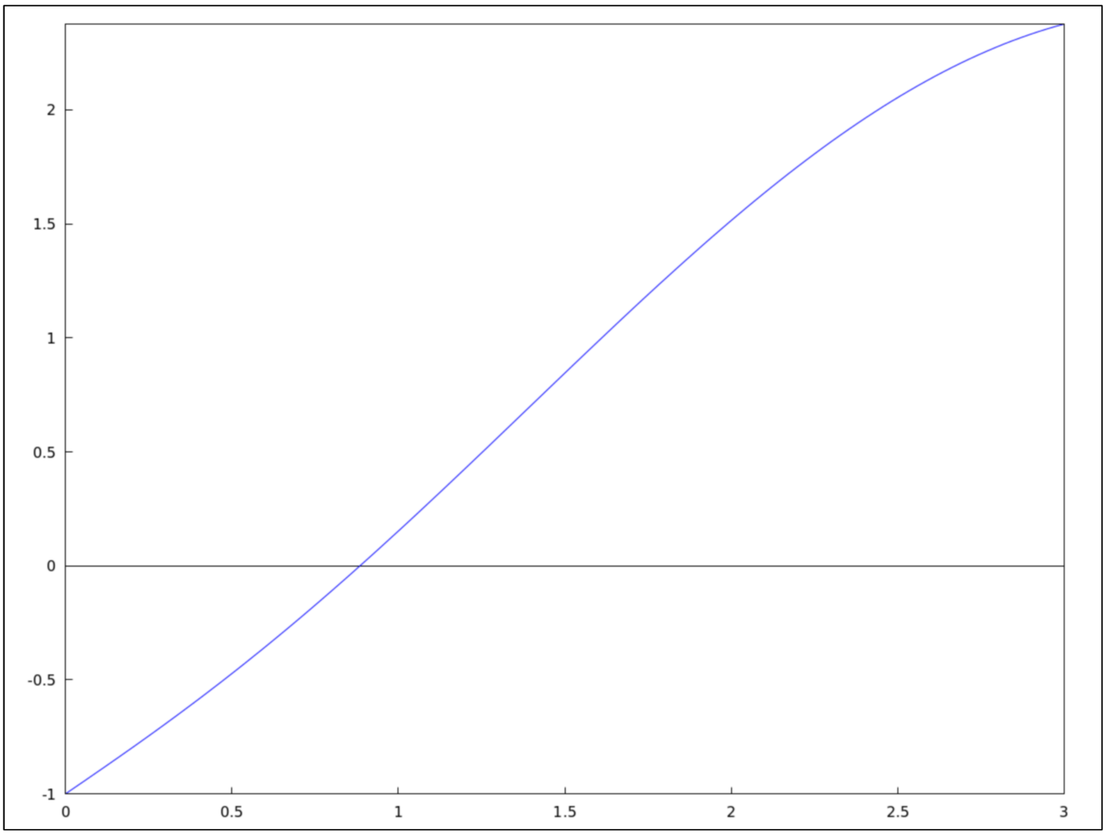
2.5

plot

```
(%i14) wxdraw2d(
    xaxis = true, xaxis_type = solid,
    yaxis = true, yaxis_type = solid,

    explicit(log(1+x)−cos(x),   x,  0,  3)
);
```

(%t14)



(%o14)

2.5.1

```
→   newton(log(1+x)−cos(x),  1.0,  0.0001,  0.00001,  8);
```

1 0.8860617363903003
0.002023466721291056
2 0.8845109403287758 4.22788446918787 10^{-7}
3 0.8845106161658667 1.854072451124011 10^{-14}

(%o7) *convergence*

3

Exercise

3.1

Figure 3:

1. Each of the following equations has a root on the interval $(0,1)$. Perform Newton's method to determine p_4, the fourth approximation to the location of the root.
   (a) $\ln(1+x) - \cos x = 0$
   (b) $x^5 + 2x - 1 = 0$
   (c) $e^{-x} - x = 0$
   (d) $\cos x - x = 0$

3.2

Figure 4:

In Exercises 3–6, an equation, an interval on which the equation has a root, and the exact value of the root are specified.

(1) Perform five (5) iterations of Newton's method.

Figure 5:

3. The equation $x^3 + x^2 - 3x - 3 = 0$ has a root on the interval $(1,2)$, namely $x = \sqrt{3}$.
4. The equation $x^7 = 3$ has a root on the interval $(1,2)$, namely $x = \sqrt[7]{3}$.
5. The equation $x^3 - 13 = 0$ has a root on the interval $(2,3)$, namely $\sqrt[3]{13}$.
6. The equation $1/x - 37 = 0$ has a zero on the interval $(0.01, 0.1)$, namely $x = 1/37$.

3.3

Figure 6:

14. For each of the functions given below, use Newton's method to approximate all real roots. Use an absolute tolerance of 10^{-6} as a stopping condition.
    (a) $f(x) = e^x + x^2 - x - 4$
    (b) $f(x) = x^3 - x^2 - 10x + 7$
    (c) $f(x) = 1.05 - 1.04x + \ln x$