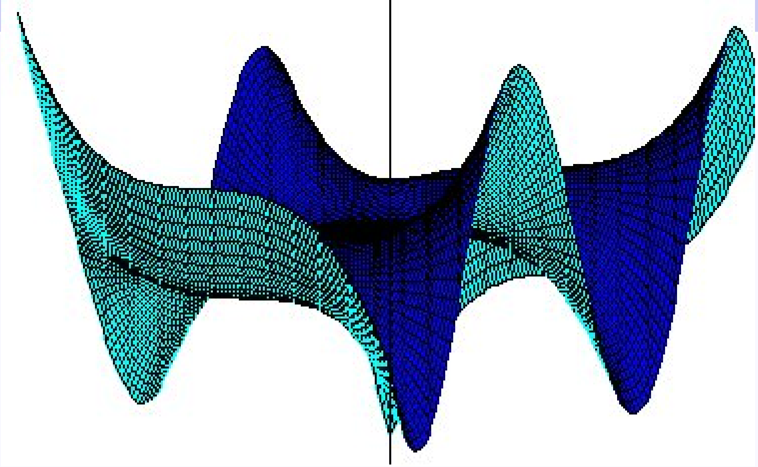

Scilab

Prof. S. A. Katre
Dept. of Mathematics, University of Pune
sakatre@math.unipune.ernet.in
sakatre@gmail.com
sakatre@bprim.org

July 25, 2009

1. n-th roots

```
->%i
 %i  =

    i

 -->sqrt(%i)
 ans  =

    0.7071068 + 0.7071068i

 -->(%i)^(1/3)
 ans  =

    0.8660254 + 0.5i

-->(1)^(1/5)
 ans  =

    1.
```

2. Newton-Raphson Method

$f(x)$ and initial guess x_0 is given, find out 'zero' for $f(x)$.

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Let $f(x) = \cos(x)$, and $x_0 = 10$

```
-->deff('[y]=f(x)','y=cos(x)');


-->deff('[y]=f1(x)','y=-sin(x)');



-->deff('[y]=g(x)','y=x-f(x)/f1(x)');
```

```
-->g(10)
 ans  =

    11.542351

-->g(ans)
 ans  =

    10.933672

-->g(ans)
 ans  =

    10.995653
```

```
-->g(ans)
 ans  =

    10.995574

-->g(ans)
 ans  =

    10.995574
```

3. Newton-Raphson: More zeros

```
-->x=[0:.2:10]
 x  =


         column  1 to 10

    0.   0.2    0.4    0.6    0.8    1.    1.2    1.4    1.6

         column 11 to 20

    2.   2.2    2.4    2.6    2.8    3.    3.2    3.4    3.6

         column 21 to 30

    4.   4.2    4.4    4.6    4.8    5.    5.2    5.4    5.6
```

```
        column 31 to 40

6.   6.2    6.4    6.6    6.8    7.    7.2    7.4    7.6

        column 41 to 50

8.   8.2     8.4    8.6    8.8    9.    9.2    9.4    9.6

        column 51

10.
```

```
-->cos(x)
 ans  =

          column 1 to 5

  1. 0.9800666  0.9210610  0.8253356   0.6967067

         column  6 to 10

  0.5403023  0.3623578  0.1699671  -0.0291995 -0.227202

         column 11 to 15

 -0.4161468  -0.5885011  -0.7373937 -0.8568888 -0.9422
```

```
column 16 to 20

-0.9899925 -0.9982948 -0.9667982 -0.8967584 -0.7909677

column 21 to 25

-0.6536436 -0.4902608 -0.3073329 -0.1121525 0.0874990

column 26 to 30
```

Introduction
Special Symbols in . . .
User defined functions
Rows and Columns
Graphics
Newton-Raphson . . .
Book on Scilab

Home Page
Title Page
Page 9 of 27
Go Back
Full Screen
Close
Quit

Introduction
Special Symbols in . . .
User defined functions
Rows and Columns
Graphics
Newton-Raphson . . .
Book on Scilab

```
0.2836622 0.4685167  0.6346929  0.7755659  0.8855195

        column 31 to 35

0.9601703 0.9965421  0.9931849  0.9502326  0.8693975

        column 36 to 40

0.7539023 0.6083513  0.4385473 0.2512598  0.0539554
```

Home Page
Title Page
Page 10 of 27
Go Back
Full Screen
Close
Quit

```
        column 41 to 45

-0.1455000 -0.3391549 -0.5192887 -0.6787200 -0.8110930

        column 46 to 50

-0.9111303 -0.9748436 -0.9996930 -0.9846879 -0.9304263

        column 51

-0.8390715
```

We find from the above list of values of $\cos x$ from 0 to 10, that cos changes sign 3 times in [0, 10]. We thus expect to find 3 zeros of $\cos x$, one between 1.4 and 1.6, one between 4.6 and 4.8, and one between 7.8 and 8.
We use the Newton-Raphson formula
$f(x_{n+1}) = x_n - f(x_n)/f'(x_n)$.
Let $f(x) = \cos(x)$, and consider the values of x for which $f(x)$ is nearer to zero.

Thus let $x_0 = [1.6, 4.8, 7.8]$.

```
-->deff('[y]=f(x)','y=cos(x)');
-->deff('[y]=f1(x)','y=-sin(x)');
-->deff('[y]=g(x)','[y]= x-f(x)./f1(x)');

-->x0=[1.6,4.8,7.8]    (vector of approximations to root
 x0  =

    1.6    4.8    7.8
```

```
-->g(x0)
 ans  =
     1.570788    4.7121641    7.8540341

-->g(ans)
 ans  =
     1.5707963    4.712389    7.8539816

-->g(ans)
 ans  =
     1.5707963    4.712389    7.8539816
```

(We have got the 3 zeros of cos x between 0 and 10.)

```
-->%pi/2

 ans  =
     1.5707963
```

This checks that the zeros are $\pi/2, 3\pi/2$ and $5\pi/2$ as expected.

```
-->x=poly(0,'x')
 x  =

    x

-->roots(x^3-x+1)
 ans  =

    0.6623590 + 0.5622795i
    0.6623590 - 0.5622795i
  - 1.324718
```

The polynomial has 1 real root and 2 nonreal roots which are complex conjugates. Recall that complex roots of a polynomial with real coefficients occur in conjugate pairs.

Let us find a complex root of $f(x) = x^3 - x + 1$ by Newton Raphson method. To get it we have to start with a complex (nonreal) initial approximation, e.g. $1 + i$.

```
 -->f(1+%i)
 ans  =
   - 2. + i



-->deff('[y]=f(x)','y=x^3-x+1')

-->deff('[y]=f1(x)','y=3*x^2-1')

-->deff('[y]=g(x)','y=x-f(x)/f1(x)')

-->g(1+%i)
 ans  =
     0.7837838 + 0.7027027i
```

```
-->g(ans)
ans  =
      0.6860940 + 0.5813564i

-->g(ans)
 ans  =
      0.6633325 + 0.5625179i

-->g(ans)
 ans  =
      0.6623599 + 0.5622788i

-->g(ans)
 ans  =
      0.6623590 + 0.5622795i

-->g(ans)
 ans  =
      0.6623590 + 0.5622795i
```

4. Bisection Method

The bisection method is very slow as compared to Newton Raphson method.
To find a root of $f(x)$ we first find some CLOSE values of a and b for which $f(a)$ and $f(b)$ have opposite signs.
Then find $c = (a+b)/2$.
If $f(a)$ and $f(c)$ have same signs (so that $f(b)$ and $f(c)$ have opposite signs, consider c as better than a and replace a by c (i.e. put $a = c$) and keep b as it is.
If $f(b)$ and $f(c)$ have same signs, replace b by c, i.e. put $b = c$.
Then find $c = (a+b)/2$ for the new a and b and proceed till you get sufficiently close values of a and b. Then declare $c = (a+b)/2$ as an (approximate) root of $f(x)$.
We shall now find a real root of $f(x) = x^3 - x + 1$ by bisection method.

```
-->x=poly(0,'x')
 x  =

      x

-->roots(x^3-x+1)
 ans  =

      0.6623590 + 0.5622795i
     0.6623590 - 0.5622795i
  - 1.324718

-->deff('[y]=f(x)','y=x^3-x+1')

-->f(3)
 ans  =

      25.

-->f(2)
 ans  =

      7.
```

```
-->f(0)
 ans  =
      1.
 -->f(-2)
 ans  =
    - 5.

-->a=0,b=-2
 a  =
      0.
 b  =
    - 2.

-->c=(a+b)/2
 ans  =
    - 1.
 -->d=[a,b,c]
 d  =
      0.  - 2.  - 1.
 -->f(d)
 ans  =
      1.  - 5.    1.
```

```
-->a=c
 a  =
   - 1.
 -->c
 c  =
   - 1.
 (This c is not relevant.)

-->c=(a+b)/2
 c  =
   - 1.5

-->d=[a,b,c],f(d)
 d  =
   - 1.  - 2.  - 1.5
 ans  =
     1.  - 5.  - 0.875
```

```
-->b=c
 b  =
   - 1.5
 -->c=(a+b)/2
 c  =
   - 1.25
 -->d=[a,b,c],f(d)
 d  =
   - 1.  - 1.5  - 1.25
 ans  =
     1.  - 0.875    0.296875

-->b=c,c=(a+b)/2,d=[a,b,c],f(d)
 b  =
   - 1.25
 c  =
   - 1.125
 d  =
   - 1.  - 1.25  - 1.125
 ans  =
     1.    0.296875    0.7011719
```

(We made a mistake in the choice of a,b.)

```
-->a=-1,b=-1.5,c=-1.25
a  =
   - 1.
b  =
   - 1.5
c  =
   - 1.25
-->a=c
a  =
   - 1.25

-->c=(a+b)/2,d=[a,b,c],f(d)
c  =
   - 1.375
d  =
   - 1.25  - 1.5  - 1.375
ans  =
     0.296875  - 0.875  - 0.2246094
```

```
-->b=c
 b  =
    - 1.375

-->c=(a+b)/2,d=[a,b,c],f(d)
 c  =
    - 1.3125
 d  =
    - 1.25   - 1.375   - 1.3125
 ans  =
      0.296875  - 0.2246094    0.0515137

 --> a=c;

-->c=(a+b)/2,d=[a,b,c],f(d)
 c  =
    - 1.34375
 d  =
    - 1.3125  - 1.375  - 1.34375
 ans  =
      0.0515137  - 0.2246094  - 0.0826111
```

```
-->b=c
 b  =
   - 1.34375

-->c=(a+b)/2,d=[a,b,c],f(d)
 c  =
   - 1.328125
 d  =
   - 1.3125  - 1.34375  - 1.328125
 ans  =
     0.0515137  - 0.0826111  - 0.0145760
-->b=c
 b  =
   - 1.328125

-->c=(a+b)/2,d=[a,b,c],f(d)
 c  =
   - 1.3203125
 d  =
   - 1.3125  - 1.328125  - 1.3203125
 ans  =
     0.0515137  - 0.0145760    0.0187106
```

```
-->a=c
 a  =
   - 1.3203125
-->c=(a+b)/2,d=[a,b,c],f(d)
 c  =
   - 1.3242188
 d  =
   - 1.3203125  - 1.328125  - 1.3242188
 ans  =
     0.0187106  - 0.0145760    0.0021279

-->a=c
 a  =
   - 1.3242188

-->c=(a+b)/2,d=[a,b,c],f(d)
 c  =
   - 1.3261719
 d  =
   - 1.3242188  - 1.328125  - 1.3261719
 ans  =
     0.0021279  - 0.0145760  - 0.0062088
```

```
-->b=c
 b  =
   - 1.3261719
-->c=(a+b)/2,d=[a,b,c],f(d)
 c  =
   - 1.3251953
 d  =
   - 1.3242188  - 1.3261719  - 1.3251953
 ans  =
     0.0021279  - 0.0062088  - 0.0020367

-->b=c
 b  =
   - 1.3251953

-->c=(a+b)/2,d=[a,b,c],f(d)
 c  =
   - 1.324707
 d  =
   - 1.3242188  - 1.3251953  - 1.324707
 ans  =
     0.0021279  - 0.0020367    0.0000466
```

```
-->a=c
 a  =
    - 1.324707

-->c=(a+b)/2,d=[a,b,c],f(d)
 c  =
    - 1.3249512
 d  =
    - 1.324707  - 1.3251953  - 1.3249512
 ans  =
      0.0000466  - 0.0020367  - 0.0009948
```

The answer by Newton Raphson method comes out to be -1.324718 in a few steps only and $f(-1.324718)$ is $-.0000002$. This example illustrates that Newton Raphson method which has quadratic convergence is far superior to bisection method