5. ROOT FINDING

This feature of the program allows the user to solve non-linear equation like:

@COS(X)-X=0

The program offers 3 types of solving methods

1. Plot and search - plot the function and find the points where the function actually cross the zero axis.
2. Under relaxation - A self converging method which converges to the root of the function, the result depends on the initial value which can be determined using the Plot and search first.
3. Newton's - Like Under relaxation but needs defining the derivative too, which is calculated analytically by the user from the known function.

In this example we will show how we can solve the equation:

@cos(X)-X=0

for X.

5.1 <R>oot_find

Press <r> to get the next menu:

```
Plot_search  Under_relaxation  Newton's  View  Repeat  Edit
```

5.2 <P>lot_search

This feature allows the user to plot any analytic function and view where the function is equal to zero. By "zooming" the plot the user can get closer and closer to the root of any equation. The equation should be arranged as F(x)=0 then by plotting F(x) as a function of x we can view where the roots are located.

Let's try to solve the equation   COS(X)-X=0

Press <p> to get the next screen:

```
            A           B           C           D           E
1 ====================================================================
2          Units         |      Output/Input       |      Memory      |
3 ====================================================================
4   None                 |                    0.00 |              0.00|
5 ====================================================================
         EV
1   @COS(VAR)-VAR
2
3
```

```
4
5
6
7
8   Input or edit the function as a FORMULA and press ENTER ....
9
10
11
12
13
14
```

The program is now in the Lotus EDIT mode so you can type the function as a Lotus formula (no prefix) and press ENTER (press the ENTER key only to quit to the previous menu). Notice that we have used the variable "VAR" as the function variable, you can also use "R" as a variable (@COS(R)-R) but we recommend to use the variable "VAR" unless the function is too long to fit in 240 characters.

**** WARNING ****

Since the program is now in the EDIT mode the Lotus keys are operative, however the user is strongly advised to use ONLY the UP, PGUP, DOWN and PGDN keys to scroll between the lines, the program doesn't allow scrolling to the sides by hiding all the columns except the current column. Trying to unhide the columns MAY destroy the program code. For example: if by mistake you have pressed the "/" key press ESC to get back to the READY mode and then press F2 to enter EDIT mode or just continue to type or press ENTER key to exit.

Type the formula [@cos(var)-var] and press ENTER to get the next three prompts:

Input the lower limit - 0

Input the upper limit - 7

Input the number of steps - 100

In each prompt type the number and press ENTER, if all three inputs are null (the ENTER key or the ESC were pressed three times, the program will quit to the previous menu) after the third prompt the next screen will shows up:

```
Plot_search  Under_relaxation  Newton's  View  Repeat  Edit
Search method (the function should be in the form of f(X) = 0 )
            A           B           C           D           E
1 ===================================================================
2          Units        |      Output/Input     |        Memory      |
3 ===================================================================
4   None                |                  0.00 |                0.00|
5 ===================================================================
        EV            EW            EX            EY            EZ
1               1             0             1             1
2    0.9275510003        0.07             1             1
```

3	0.8502159962	0.14	1	1
4	0.7680309147	0.21	1	1
5	0.6810554383	0.28	1	1
6	0.5893727128	0.35	1	1
7	0.4930889403	0.42	1	1
8	0.3923328586	0.49	1	1
9	0.287255111	0.56	1	1
10	0.1780275083	0.63	1	1
11	0.0648421873	0.7	1	1
12	-0.0520893304	0.77	1	1
13	-0.1725371742	0.84	1	1
14	-0.2962542505	0.91	1	1

As you can see the function (in column EV) changes its sign from plus to minus between VAR=0.7 and VAR=0.77 which means that the solution to the equation is somewhere between 0.7 and 0.77. To see if there are more roots to the equation use the <V>iew menu item to view the graph and to scroll through the results.

Press <V> to get the next menu:

Down Up Next Previous Graph Save_graph

Using the Down, Up, Next and Previous menu items you can scroll through the data table to look for sign changes in the function (column EV). To view the graph press <g>. To get closer to the root let's plot the function between 0.7 and 0.77.

Press <R>epeat and answer the prompts (type the numbers and press ENTER) to get the next screen.

Plot_search Under_relaxation Newton's View Repeat Edit
Search method (the function should be in the form of f(X) = 0)

	A	B	C	D	E
1	=====	=====	=====	=====	=====
2	Units		Output/Input		Memory
3	=====	=====	=====	=====	=====
4	None		0.00		0.00
5	=====	=====	=====	=====	=====

	EV	EW	EX	EY	EZ
1	0.0648421873	0.7	1	1	
2	0.0636910476	0.7007	1	1	
3	0.0625395333	0.7014	1	1	
4	0.0613876447	0.7021	1	1	
5	0.0602353819	0.7028	1	1	
6	0.0590827453	0.7035	1	1	
7	0.0579297351	0.7042	1	1	
8	0.0567763513	0.7049	1	1	
9	0.0556225944	0.7056	1	1	
10	0.0544684645	0.7063	1	1	
11	0.0533139617	0.707	1	1	
12	0.0521590865	0.7077	1	1	
13	0.0510038389	0.7084	1	1	
14	0.0498482192	0.7091	1	1	

Press <V>iew to see the next menu and the data table:

```
Down  Up  Next  Previous  Graph  Save_graph
Move one cell down                                   [ESC] - Previous menu
            A           B           C           D           E
1 ======================================================================
2         Units         |     Output/Input      |        Memory        |
3 ======================================================================
4   None                |                  0.00 |                  0.00|
5 ======================================================================
         EV            EW            EX            EY            EZ
49   0.0091688481       0.7336
50   0.0079999826       0.7343
51   0.0068307534        0.735
52   0.0056611607       0.7357
53   0.0044912047       0.7364
54   0.0033208857       0.7371
55   0.0021502038       0.7378
56   0.0009791594       0.7385
57  -0.0001922473       0.7392
58  -0.0013640162       0.7399
59  -0.0025361469       0.7406
60  -0.0037086393       0.7413
61  -0.0048814931        0.742
62  -0.0060547081       0.7427
```

Using the Down, Up, Next and Previous menu items you can scroll through the data table to look for the sign changes in the function (column EV). It is clear that the root is between 0.7385 to 0.7392. to find more accurate results we can plot the function again between these two values.

To view the graph press <g>.

<U>nder relaxation

This feature uses the Under Relaxation method to solve non-linear equations. The equation should be arranged in the form of G(x)=x . The program uses the iteration equation:

X2 = C*G(X1) + (1-C)*X2

where C is the relaxation factor which affects the convergence rate of the solution. By "playing" with different values of C (0<C<1) the user can find the best C to converge as fast as possible.

Let's try to solve the equation   COS(X)-X=0

Press <U> to get the next screen:

```
Plot_search  Under_relaxation  Newton's  View  Repeat  Edit
```

```
---------------------------------------------------------------------
           A           B           C           D           E
1 ==================================================================
2        Units         |     Output/Input      |       Memory      |
3 ==================================================================
4   None               |                  0.00 |               0.00|
5 ==================================================================
        EV            EW            EX
1
2
3  Function -                    @cos(var)
4
5
6
7
8   Input or edit the function as a FORMULA and press ENTER ....
9
10
11
12
13
14
```

The program is now in the Lotus EDIT mode so you can type the function as a Lotus formula (no prefix) and press ENTER (press the ENTER key only to quit to the previous menu). Notice that we have used the variable "VAR" as the function variable, you can also use "R" as a variable (@COS(R)) but we recommend to use the variable "VAR" unless the function is too long to fit in 240 characters.

**** WARNING ****

Since the program is now in the EDIT mode the Lotus keys are operative, however the user is strongly advised to use ONLY the UP, PGUP, DOWN and PGDN keys to scroll between the lines, the program doesn't allow scrolling to the sides by hiding all the columns except the current column. Trying to unhide the columns MAY destroy the program code. For example: if by mistake you have pressed the "/" key press ESC to get back to the READY mode and then press F2 to enter EDIT mode or just continue to type or press ENTER key to exit.

Type the formula [@cos(var)] and press ENTER to get the next screen:

Accuracy -

```
           A           B           C           D           E
1 ==================================================================
2        Units         |     Output/Input      |       Memory      |
3 ==================================================================
4   None               |                  1.00 |               0.00|
5 ==================================================================
        EV            EW            EX            EY            EZ
1              1
```

```
 2
 3   Function -                  @COS(EW1)
 4
 5
 6   Accuracy -                         0.0001
 7
 8   Initial value -                         2
 9
 10  Max. iterations -                     100
 11
 12  Number of iterations -                  0
 13
 14  Relaxation factor (0<C<1) -            .5
```

Type the accuracy, Initial value, Max. iterations and Relaxation factor (0<C<1) and press ENTER as the program prompts, the next screen is:

```
Plot_search  Under_relaxation  Newton's  View  Repeat
Search method (the function should be in the form of f(X) = 0 )
             A           B           C           D           E
 1 ===================================================================
 2          Units        |      Output/Input       |        Memory      |
 3 ===================================================================
 4   None                |            0.7390616306 |                0.00|
 5 ===================================================================
          EV            EW            EX            EY            EZ
 1    0.7390616306   0.739120023
 2
 3   Function -                  @COS(EW1)
 4
 5
 6   Accuracy -                         0.0001
 7
 8   Initial value -                         2
 9
 10  Max. iterations                       100
 11
 12  Number of iterations -                  6
 13
 14  Relaxation factor (0<C<1) -           0.5
```

The program used only six iterations to calculate the root to .0001 accuracy 0.0001 = last value/previous value) the last two iterations appear in the cells EW1 and EV1. Using other values of C will result in different number of iterations.

<N>ewton's

This feature uses the Newton's method to solve non-linear equations. The equation should be arranged in the form of G(x)=0. The user also needs to calculate the derivative analytically. The program uses the iteration equation:

X2 = C*G(X1) + (1-C)*X2 (see reference 1 in the README file)

where C is the relaxation factor which affects the convergence rate of the solution. By "playing" with different values of C (0<C<1) the user can find the best C to converge as fast as possible.

Let's try to solve the equation   COS(X)-X=0

Press <N> to get the next screen:

```
            A           B           C           D           E
1 ====================================================================
2         Units         |      Output/Input      |       Memory       |
3 ====================================================================
4   None                |                   0.74 |                0.00|
5 ====================================================================
       EV            EW            EX
1
2
3   Function -          @cos(var)-var
4
5
6
7
8   Input or edit the function as a FORMULA and press ENTER ....
9
10
11
12
13
14
```

The program is now in the Lotus EDIT mode so you can type the function as a Lotus formula (no prefix) and press ENTER (press the ENTER key only to quit to the previous menu). Notice that we have used the variable "VAR" as the function variable, you can also use "R" as a variable (@COS(R)) but we recommend to use the variable "VAR" unless the function is too long to fit in 240 characters.

**** WARNING ****

Since the program is in the EDIT mode the Lotus keys are operative, however the user is strongly advised to use ONLY the UP, PGUP, DOWN and PGDN keys to scroll between the lines, the program will not allow scrolling to the sides by hiding all the columns except the current columns. Trying to unhide the columns MIGHT destroy the program code. For example: if you by mistake pressed the "/" key press ESC to get back to the READY mode and then press F2 to enter EDIT mode or just continue to type or press ENTER to get out.

Type the formula [@cos(var)] and press ENTER to get the next screen:

```
           A           B           C           D           E
1 ====================================================================
2         Units         |     Output/Input       |       Memory       |
3 ====================================================================
4   None                |                   0.74 |                0.00|
5 ====================================================================
       EV            EW            EX            EY
1            1
2
3   Function -                  @COS(EW1)-EW1
4   Derivative -                -@SIN(EW1)-1
5
6
7
8   Input or edit the Derivative as a FORMULA and press ENTER ....
9
10
11
12
13
14
```

Again you are in the EDIT mode; type the derivative and press ENTER to get the next screen:

```
           A           B           C           D           E
1 ====================================================================
2         Units         |     Output/Input       |       Memory       |
3 ====================================================================
4   None                |           0.7345361689 |                0.00|
5 ====================================================================
       EV            EW            EX            EY            EZ
1    0.7345361689            2
2
3   Function -                  @COS(EW1)-EW1
4   Function derivative -       -@SIN(EW1)-1
5
6   Accuracy -                        0.0001
7
8   Initial value -                        2
9
10  Max. No. of iterations               100
11
12  Number of iterations -                 0
13
14  Relaxation factor (0<C<1) -
```

To get the next screen type the accuracy, Initial value and Max. iterations and press ENTER after each prompt.

```
Plot_search  Under_relaxation  Newton's  View  Repeat  Edit
```

```
Search method (the function should be in the form of f(X) = 0 )
               A           B           C           D           E           F
1   ============================================================================
2           Units          |     Output/Input      |         Memory          |
3   ============================================================================
4   None                   |          0.7390851332 |                       0 |
5   ============================================================================
          EV            EW            EX            EY            EZ
1    0.7390851332  0.7390897242
2
3   Function -                  @COS(EW1)-EW1
4   Derivative -                -@SIN(EW1)-1
5
6   Accuracy -                         0.0001
7
8   Initial value -                         2
9
10  Max. iterations                       100
11
12  Number of iterations -                  3
13
14
```

The program used only two iterations to calculate the root to .0001 (0.0001 = last value/previous value) the last two iterations appear in the cells EW1 and EV1.