

Newton's method to find a root of equations system.

This example tries to find any one of the roots of below nonlinear system:

$$\begin{aligned}x_1^2 + x_2^2 - 1 &= 0 \\ 5x_1^2 - x_2 - 2 &= 0\end{aligned}$$

This is the code for Newton-Raphson algorithm.

```
1 function [xsolution, Xk, Fk, Jk, IFLAG, IterationsUsed] = newton(FunctionName, x0, epsilon,
2 IterationMax)
3
4 Xk = x0; % list to store x_k.
5 Fk = []; % list to store f_k.
6 Jk = []; % list to store J_k.
7 IFLAG = 0; % Flag to indicate whether x_k converge.
8
9 for i = 1:IterationMax
10     [f, J] = FunctionName(x0); % generate f and J matrix at x0.
11     s = J\(-f); % solve for step value s.
12     x1 = x0 + s; % find next x_k.
13
14     Xk(:,i+1) = x1; % store new x_k.
15     Fk(:,i) = f; % store new f_k.
16     Jk(:,i) = J; % store new J_k.
17
18     if norm(x1-x0) <= epsilon % stop if the step is small enough.
19         IterationsUsed = i; % return a number of iterations.
20         xsolution = x1; % return the root of function.
21         [f, J] = FunctionName(x1); % finding new f, J final value.
22         Fk(:,i+1) = f; % store last f_k.
23         Jk(:,i+1) = J; % store last J_k.
24         break
25     end
26
27     x0 = x1; % set next x_k to be x_(k+1).
28 end
29
30 if IterationsUsed == IterationMax % to indicate x_k is not converge.
31     IFLAG = 1;
32 end
33 end
```

This is the code for the particular function handling.

```
1 function [f,J] = FunctionName(xin)
2
3 % Declare the functions.
4 f_fun = @(x)[x(1)^2+x(2)^2-1;5*x(1)^2-x(2)-2];
5 J_fun = @(x)[2*x(1),2*x(2);10*x(1),-1];
6
7 % Evaluate numerical values.
8 f = f_fun(xin);
9 J = J_fun(xin);
10
11 end
```

This is the main script file, together with the code used to generate the table result.

```
1 [xsolution, Xk, Fk, Jk, IFLAG, IterationsUsed] = newton(@FunctionName, [8;5], 0.0001, 100);
2
3 % report the values as a table.
4 fprintf('% 5s % 10s % 10s % 10s % 10s \n', 'Iter', 'x1_k', 'x2_k', 'f1_k', 'f2_k');
5 for i = 1:IterationsUsed+1
6     fprintf('% 5.2d % 10.3f % 10.3f % 10.3f % 10.3f \n', i, Xk(1,i), Xk(2,i), Fk(1,i), Fk(2,
7 i));
8 end
```

The result is reported as in this table.

1	Iter	x1_k	x2_k	f1_k	f2_k
2	01	8.000	5.000	88.000	313.000
3	02	4.056	2.510	21.753	77.761
4	03	2.110	1.322	5.199	18.940
5	04	1.189	0.825	1.095	4.242
6	05	0.821	0.692	0.153	0.677
7	06	0.737	0.681	0.007	0.035
8	07	0.732	0.681	0.000	0.000
9	08	0.732	0.681	0.000	0.000

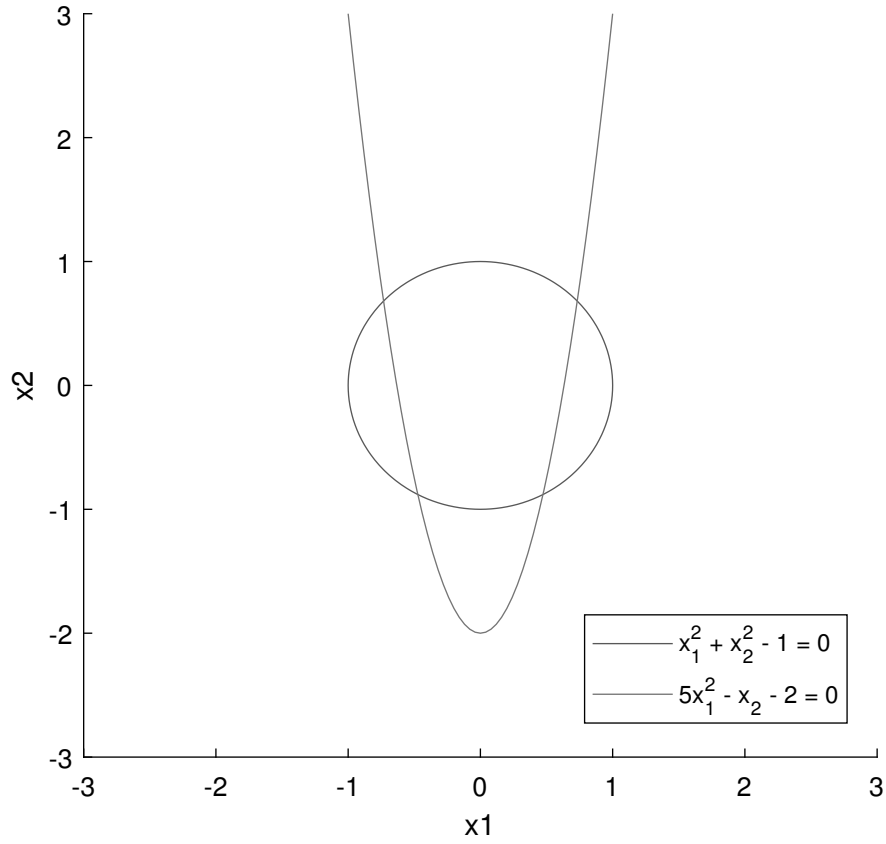


Figure 1: The graphs of $x_1^2 + x_2^2 - 1 = 0$ and $5x_1^2 - x_2 - 2 = 0$.

From the graphic, there are 4 solutions in this nonlinear system. The result from the iteration, $(x_1, x_2) = (0.732, 0.681)$ is represented by the upper-right point of the 4 intersection points (which are the solutions of nonlinear system.)