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

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

$$x_1^2 + x_2^2 - 1 = 0$$
$$5x_1^2 - x_2 - 2 = 0$$

This is the code for the Newton-Raphson algorithm.

```
function [xsolution, Xk, Fk, Jk, IFLAG, IterationsUsed] = newton(FunctionName, x0, epsilon,
    IterationMax)

    Xk = x0; % list to store x_k.
    Fk = []; % list to store f_k.
    Jk = []; % list to store J_k.
    IFLAG = 0; % Flag to indicate whether x_k converge.

    for i = 1:IterationMax
        [f, J] = FunctionName(x0); % generate f and J matrix at x0.
        s = J\(-f); % solve for step value s.
        x1 = x0 + s; % find next x_k.

        Xk(:,i+1) = x1; % store new x_k.
        Fk(:,i) = f; % store new f_k.
        Jk(:,:,i) = J; % store new J_k.

        if norm(x1-x0) <= epsilon % stop if the step is small enough.
            IterationsUsed = i; % return a number of iterations.
            xsolution = x1; % return the root of function.
            [f, J] = FunctionName(x1); % finding new f, J final value.
            Fk(:,i+1) = f; % store last f_k.
            Jk(:,:,i+1) = J; % store last J_k.
            break
        end

        x0 = x1; % set next x_k to be x_(k+1).
    end

    if IterationsUsed == IterationMax % to indicate x_k is not converge.
        IFLAG = 1;
    end

end
```

This is the code for handling the particular function.

```
function [f,J] = FunctionName(xin)

    % Declare the functions.
    f_fun = @(x)[x(1)^2+x(2)^2-1;5*x(1)^2-x(2)-2];
    J_fun = @(x)[2*x(1),2*x(2);10*x(1),-1];

    % Evaluate numerical values.
    f = f_fun(xin);
    J = J_fun(xin);

end
```

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

```
[xsolution, Xk, Fk, Jk, IFLAG, IterationsUsed] = newton(@FunctionName, [8;5], 0.0001, 100);

% report the values as a table.
fprintf('% 5s % 10s % 10s % 10s % 10s \n', 'Iter', 'x1_k', 'x2_k', 'f1_k', 'f2_k');
for i = 1:IterationsUsed+1
    fprintf('% 5.2d % 10.3f % 10.3f % 10.3f % 10.3f \n', i, Xk(1,i), Xk(2,i), Fk(1,i), Fk(2,
    i));
end
```

The result is reported in this table.

```
Iter        x1_k        x2_k        f1_k        f2_k
  01       8.000       5.000      88.000     313.000
  02       4.056       2.510      21.753      77.761
  03       2.110       1.322       5.199      18.940
  04       1.189       0.825       1.095       4.242
  05       0.821       0.692       0.153       0.677
  06       0.737       0.681       0.007       0.035
  07       0.732       0.681       0.000       0.000
  08       0.732       0.681       0.000       0.000
```

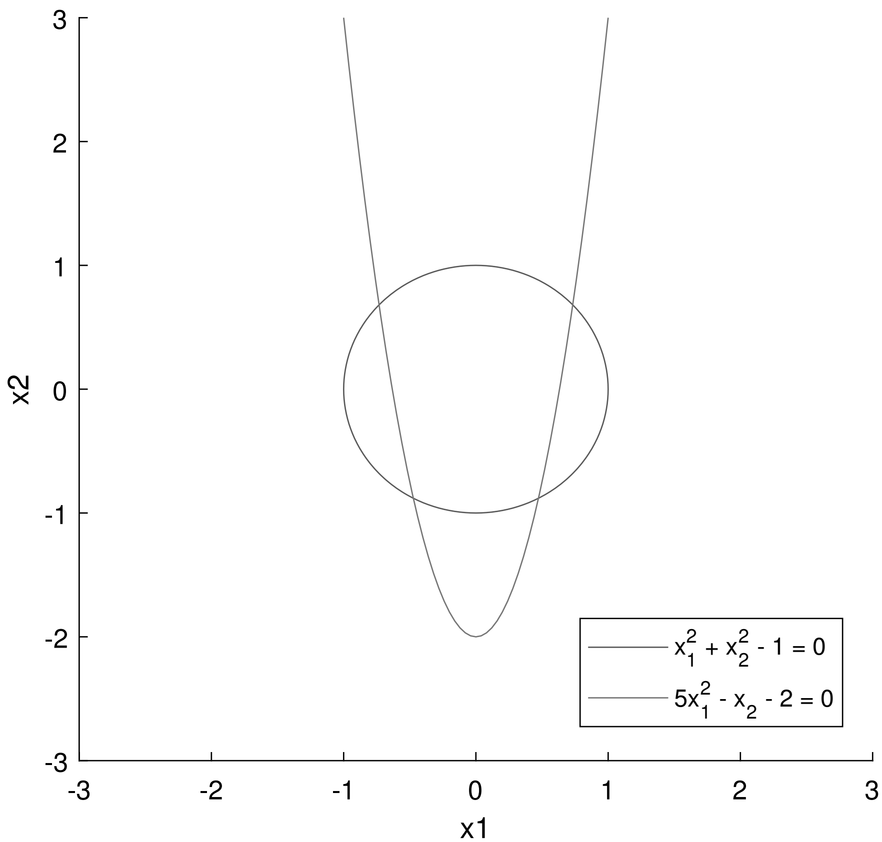

Figure 1: The graphs of $x_1^2 + x_2^2 - 1 = 0$ and $5x_1^2 - x_2 - 2 = 0$.

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 the nonlinear system).