Solving Nonlinear Equation(s) in MATLAB

1 Introduction

This tutorial helps you use MATLAB to solve nonlinear algebraic equations of single or multiple variables.

2 Writing MATLAB functions

In order to use the MATLAB solvers, you must first be able to write MATLAB functions. There are two different methods to create a function - (a) inline command, and (b) Matlab editor

2.1 The '*inline*' command

The inline command can be used for simple, one-line functions. For example, to create $f(x) = x^3 - 5x^2 - x + 2$:

```
>> f = inline('x^3 -5*x^2 - x+2')

f =

    Inline function:
    f(x) = x^3-5*x^2-x+2
```

You can now evaluate the function value at any given x. For example, to evaluate the function value at $x = 4$, simply type '*f*(4)' at Matlab command line.

```
EDU>> f(4)

ans =

   -18
```

2.2 The MATLAB editor

The editor allows the user to write functions of any length and/or complexity.

1. Set the current working directory to your diskspace

   e.g. "c:\CHEE222\Matlab\Iamhappy\Temp\"

2. (a) type \"edit fun" at the command prompt - enter yes to create file

ALTERNATIVELY

(b) go to "File", select "New"; select "M-File".

type the following:

```
function y = fun(x)
y  = x^3 - 5x^2 -x +2 ;
```

NOTE: The filename and the function name should be the same. In the previous example, we have chosen '***fun***' as the filename and the function name.

3. Save the file as "**fun.m**" in the working directory

3. MATLAB function FZERO

fzero can be used to solve a single variable nonlinear equation of the form $f(x) = 0$. The equation must first be programmed as a function (either inline or m-file).

3.1 Using FZERO for a function defined by *inline* command

The following command solves the equation $y = f(x) = x^3 - 5x^2 -x +2$;, starting from an initial guess of $x = 4$.

EDU>> fzero(f,4)

MATLAB returns the answer:

ans =

5.1190

Changing the initial guess to $x = 2$

EDU>> fzero(f,2)

gives

ans =

0.5684

Clearly, which solution the solver arrives at depends on the initial guess. You can restrict the search to an interval by replacing the initial guess with an interval $x \in [3 \;\; 6]$:
z = *fzero*(*f*; [3 6])

3.2 Using FZERO for a function defined in script file 'fun'

Now, try solving the function from section 2 defined in the script file fun.

```
EDU>> x=fzero('fun', 4)

x =

   5.1190
```

fzero uses a bisection approach to locating roots. Can you forsee any limitations to this? Try repeating the above with different initial conditions - how many roots can you locate?

4. MATLAB function ROOTS

If the nonlinear algebraic system is a polynomial equation, we could use the MATLAB routine roots to find the zeros of the polynomial. Consider the same function
$f(x) = x^3 - 5x^2 - x + 2$ that we discussed earlier.

The user must create a vector of the coefficients of the polynomial, in **descending** order,
p = [1 5 -1 2]:

Then the user can type the following command
roots(p)

and MATLAB returns the roots

```
EDU>> roots(p)

ans =

   5.1190
  -0.6874
   0.5684
```

Confirm that $x = 0.5684$ is a root by typing *f*(0.5864).

```
EDU>> f(.5684)

ans =

 -1.5495e-004
```

NOTE: In utilizing ROOTS function, all coefficients of the polynomial must be specified.

e.g. $f(x) = x^4 - 3x^2 + 2$.

The function in the full polynomial form must be expressed as:

$f(x) = 1 \cdot x^4 - 0 \cdot x^3 + 3 \cdot x^2 - 0 \cdot x + 2$.

Accordingly, the polynomial must be defined in MATLAB as follows:

p = [1 0 -3 0 2]:

5 FSOLVE

The MATLAB routine fsolve is used to solve sets of nonlinear algebraic equations using a quasi-Newton method. The user must supply a routine to evaluate the function vector. Consider the following system of nonlinear equations, and solve for x_1 and x_2:

$$f_1(x_1, x_2) = x_1 - 4x_1^2 - x_1 x_2$$

$$f_2(x_1, x_2) = 2x_2 - x_2^2 - 3x_1 x_2$$

The m-file used to solve the above problem using fsolve is:

```
function  f = nle(x)
f(1) = x(1) - 4 * x(1)^2 - x(1) * x(2);
f(2) = 2 * x(2) - x(2)^2 + 3 * x(1) * x(2);
```

which is placed in a m-file called *nle.m*.

Enter the initial guess

$$x_o = [11]'$$

Note: x_o is the TRANSPOSE of a row vector

Now, solve with

$x = fsolve('nle'; x_0)$

which gives us the results $x = [0.25\ 0.00]'$.