CH2013
Computational Programming and Simulations Lab
Aug-Dec 2021

Problem Sheet #3

September 8, 2021, Wednesday

In this problem sheet, we work with fsolve and also with polynomials, extensively.

1. a) Represent the following polynomials in MATLAB. Label them p1, p2, and p3, respectively
   (i) $p1 = 4x^4 + 3.2x^2 - 4$
   (ii) $p2 = 4.6x^3 - 12x^2 + 10$
   (iii) $p3 = (x-1)(x-3)^2$

   b) Find the values of p1, p2 and p3 at x=4.5; and label the results as value1, value2, and value3, respectively

   c) Find the roots of the polynomials and label the roots as r1, r2, and r3, respectively

2. The following data is obtained from a laboratory experiment

```
x3=[-3.00    -2.33 -1.67 -1.00 -0.33 0.33  1.00  1.67  2.33  3.00]
y3=[0.82     -0.77 -1.98 -1.26 0.46  1.11  0.43  0.01  0.68  1.10]
```

   a) Use the "**polyfit**" function to fit a line, a quadratic & a sixth-degree polynomial to the above data. Save the coefficients of the polynomials in poly1, poly2 and poly6 respectively.
   b) For each of the fitted polynomials, find the "predicted" values of y at each of the given x. Put these y vectors in yv1, yv2, and yv6, respectively.
   c) Plot the experimental data as symbols, and the yv1, yv2 and yv6 as lines, in the same graph.

3. Although fsolve is meant for systems of equations, it can also be used for a single f(x)=0 root finding problem as well. In this problem, we will examine the polynomial you worked with last week, again, but in three different ways

$$f(x) = x^3 - 10x^2 + 33x - 36$$

   a) Express *f(x)* as a polynomial **p** in your code. Find the roots and put them in a variable labelled '**rootsofp**'
   b) Now use **fzero** to solve the problem, use function tolerance of 1.e-2; x-tolerance of 1.e-2 and; and initial guess = 0. Save the values as xfinal1, ffinal1, iter1 (same as last week).
   c) Next use **fsolve** to solve the problem with the same tolerances & initial guess. Save the final values as xfinal2, ffinal2, and iter2. Note that you can use the same "myfun" code for this as in (b)
   d) If you use the MATLAB default tolerances for fzero and fsolve (instead of what is given here), do the roots change? Do this and put the answers in xfinal3, ffinal3 and iter3 (for fzero) and xfinal4, ffinal4 and iter4 (for fsolve).
   e) Overall what do you observe for this problem which has multiple roots, though it is a simple enough problem otherwise?
   (no submission for (e))

4. Solve the following system of equations using the MATLAB function '**fsolve**'.

$$2e^x + y = 0$$
$$3x^2 + 4y^2 = 8$$

Set the following options –
Maximum number of iterations: 10
Tolerance value (on function): 1.e-6
Display: all the iterations

Use an initial guess of (-1,-2) and put the final solution you obtain in a structure array "**Solution1**" with fields as below.

x1 -> converged solution for x
x2 -> converged solution for y
f1 -> value of the first function at convergence
f2 -> value of the second function at convergence
jacobian -> value of the jacobian at convergence
exitflag -> the exitflag
iterations-> number of iterations

Now use an initial guess of (-2,100) and put the final solution you obtain in a structure array **Solution2**, with the same fields x1, x2, etc. Figure out why this initial guess fails, and try to play around to see how to fix the problem.

--END--