CH2013
Computational Programming and Simulations Lab
Aug-Dec 2021
Problem Sheet #6

Sep 29 2021 (Wednesday – batch 1)

This problem sheet is about some plotting, interpolation, curve fitting & optimization.

1.
```
x=[0    0.4   0.8    1.2    1.6    2.0     2.4]
y=[0    0.1   0.5    1.0    1.8    2.7    14.0]
```

   Plot the above data, using markers.
   In the menu bar of the figure (once the code has run), under tools, pick "Basic Fitting"
   Try out the spline interpolation first. Note that it passes through all the points.
   Now the various curve fits like linear, quadratic etc.
   Note the differences.
   (no submission)

2.
```
x=  [0  0.5      1.0      1.5     2.0     2.5];
y=  [0  0.2      0.4      0.6     0.8     1.0];
z=  [0   0         0        1     0.6       0.4
   0   0.0047  0.0374  0.1263  0.2994   0.5848
   0   0.0059  0.0472  0.1592  0.3772   0.7368
  0   0.0067  0.0540  0.1822  0.4318   0.8434
  0   0.0074  0.0594  0.2005  0.4753   0.9283
  0   0.0080  0.0640  0.2160  0.5120   1.0000]
```

   a) Make surface plots using surf(z) and surf(x,y,z) commands and examine the difference between the two plots (if any).
   b) Set up a meshgrid `[x1,y1] = meshgrid(0:0.25:2.5,0:0.1:1)`. Use the **interp2** command to perform an interpolation for z over this meshgrid, using the above data. Call the interpolated values z1, use the default method.
   c) Using the same meshgrid perform another interpolation, this time use the spline method. Call the interpolated values z2.
   d) Make a surface plot of z vs. x,y ; z1 vs. x1,y1 and z2 vs. x1,y1 on different figures and examine the results. Also look at contour plots.

3. In this example, we just play around a little bit with spline interpolation.
```
x=[0    0.4 0.8 1.2 1.6 2.0 2.4]
y=[0    0.  0.0 1.0 1.8 2.7 2.7]
```

   a) Create a new vector x1 that goes from 0 to 2.4 in steps of 0.2
   b) Create y1 with spline interpolated values for the data (x,y) with query vector x1. Use the ***interp1*** command.
   c) Create y2 with nearest-neighbor interpolated values, use the ***interp1*** command.
   d) Create y3 with spline interpolated values for the data (x,y) at x1. Use the ***spline*** command directly (not within interp1). Is there any difference between y1 and y3?
   e) Create y4 with a 4th degree polynomial fit of the original data, evaluate the fitted values at x1, call it y4.
   f) Plot the original data and the interpolated data together, with the original data being symbols and interpolated ones being lines, and examine the results.
   g) Note that the "spline" function creates piecewise polynomials. Figure out how to get the coefficients of the piecewise polynomials for the spline interpolation. Report the coefficients in a matrix labelled "coefs". What is the size of this matrix?

4. Optimisation involves finding x which minimizes or maximises f(x) under given constraints. The f(x) may be an objective function. This is useful in many applications for us. For functions of single variables, use fminbnd and use fminsearch for functions of multiple variables. Visualisation is important (& easy to do in MATLAB).

   a) Plot the function $f(x) = 2\sin(x) - \frac{x^2}{10}$ in the range $x \in [0,8]$ and find its MAXIMUM value using ***fminbnd*** in the range [4,8]. Report the maximum function value as ffinal1, and the x at this function value as xfinal1, iterations in iter1 and exitflag as flag1. Make sure you check the answer. Use an X Tolerance of 10^{-2}, function tolerance of 10^{-12} and maximum number of iterations as 20. Note: You go back to using optimset for this function to set the non-default options.
   b) What's the exitflag if the maximum number of iterations is limited to 5? What is the algorithm used by fminbnd? (No submission for (b)).

c) If I wanted to use fzero instead of fminbnd, what would I have to do? (no submission but please think)

5. I am choosing a very simple 2-variable problem here. The textbook "Numerical methods for engineers" by *Chapra and Canale* provides a nice set of steps to solve for such situations.

$$f(x, y) = -8x + x^2 + 12y + 4y^2 - 2xy$$

a) Make contour and surface plots of the function to check where the minimum is likely to lie. To start with, choose $x \in [-5,5]$ and $y \in [-5,5]$ and fine tune this further if you wish to focus. Also figure out an approximate value for the function minimum from the plots.

b) Now invoke the fminsearch function with an initial guess of [0,0] for the values of x and y. Use function tolerance and x tolerance values of 10^{-6} and don't impose any limits from your end on the maximum iterations and function evaluations. Report the final function value as ffinal2, the values of x and y as xfinal2 and yfinal2, number of iterations as iter2 and of course exitflag as exitflag2.

c) What is the algorithm used by MATLAB for this problem?