NENG 685
HW 1

Fall 2017
Due Oct. 11, 2017

Name: Robert Torzilli

On homework:

- If you work with anyone else, document what you worked on together.
- Show your work.
- Always clearly label plots (axis labels, a title, and a legend if applicable).
- Homework should be done "by hand" (i.e. not with a numerical program such as MATLAB, Python, or Wolfram Alpha) unless otherwise specified. You may use a numerical program to check your work.
- If you use a numerical program to solve a problem, submit the associated code, input, and output (email submission is fine).

Do not write in the table to the right.

Problem	Points	Score
1	20	17
2	30	24
3	15	15
4	30	20
Total:	95	81

+5

Worked with Kevin Choe
Amy Haybook

Monday, October 2, 2017 11:04 AM

1. (20 points) Using four points on the interval $[x_0, x_3]$, do the following:
   (a) (4 points) Construct all of the Lagrange polynomials $L_j(x)$ that correspond to the points x_0, x_1, x_2, and x_3 by hand.
   (b) (4 points) Use these Lagrange polynomials to construct the interpolating polynomial, $P_3(x)$, that interpolates the function $f(x)$ at the points x_0, x_1, x_2, and x_3 by hand.
   (c) (8 points) Using the $P_3(x)$ you derived, create an interpolant for

$$f(x) = \sin(\frac{\pi}{2}x) + \frac{x^2}{4}$$

over $[x_0, x_3]$ with $x_0 = 0, x_1 = 2, x_2 = 3$, and $x_3 = 4$. You may do this using something like Python or MATLAB, but *write your own functions rather than using the built in ones.*

Plot the actual function and your interpolant using 100 equally spaced points for x between -0.5 and 4.5.

   (d) (4 points) Repeat what you did in part c but instead use $x_0 = 0, x_1 = 1, x_2 = 2.5$, and $x_3 = 4$.

Discuss the differences in how well the function is interpolated using the different point sets.

1] $[x_0, x_1, x_2, x_3]$

a] $f(x) = \sum_{i=1}^{n} y_i L_i(x) = \sum_{i=1}^{n} y_i \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)}$

$$L_0(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}$$

$$L_1(x) = \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}$$

$$L_2(x) = \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}$$

$$L_3(x) = \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}$$

b] $P_3(x) = \sum_{i=0}^{n} f(x_i) L_i(x)$

$$= f(x_0) L_0(x) + f(x_1) L_0(x) + f(x_2) L_2(x) + f(x_3) L_3(x)$$

$$P_3(x) = f(x_0)\left[\frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}\right] + f(x_1)\left[\frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}\right]$$

$$+ f(x_2)\left[\frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}\right] + f(x_3)\left[\frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}\right]$$

c] attached

d] attached

same code

-2pts: Function not plotted at all 100 points for comp; only plotted at 4 original, which is why you don't see the differences you show numerically in part d

-1pts: Function not plotted at all 100 points

The largest % difference ended up at the ends at 8.8%

This was later verified with the built in function

While we expected little changes to have nearly identical graphs indicates the new set was not different enough

Additionally this match could be due to how zoomed the graph is or the fact I am running 3.6.3

Tuesday, October 10, 2017 7:26 AM

NENG 685 HW 1 - Page 3 of 4 Due Oct. 11, 2017

2. (15 points) Using the interpolant $P_3(x)$ derived in question 1:
   (a) (3 points) Write the general expression for the error term, $err(x) = |f(x) - P_3(x)|$.
   (b) (4 points) Given
   $$f(x) = \sin(\tfrac{\pi}{2}x) + \tfrac{x^2}{4},$$
   use information about the function to bound the error expression.
   (c) (8 points) Use the values $x_0 = 0, x_1 = 2, x_2 = 3$, and $x_3 = 4$ to get the upper bound of $err(x)$ over this interval. That is, insert the points into the expression from part b, find an expression for x that maximizes error, and then find the x that gives the maximum. Present one final number. You may use a mathematical package to assist you in solving for x.

a) $err(x) = |f(x) - P_3(x)|$ derivative From given Thm the form takes

$$err(x) = e = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

b) ① solve the $err(x)$ using the given $f(x) = \sin\left(\frac{\pi x}{2}\right) + \frac{x^2}{4}$ then bind the error

We know $n = 3$ for this problem From $P_3(x)$

$$e = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \Rightarrow \frac{f^{IV}(\xi)}{4!} (x - x_0)(x - x_1)(x - x_2)(x - x_3)$$ This is part a

① $$\frac{f^{IV}(\xi)}{4!} = \frac{1}{4!}\frac{d^4}{dx^4}\left[\sin\left(\frac{\pi}{2}x\right) + \frac{x^2}{4}\right]$$

$$= \frac{1}{24}\left[\left(\frac{\pi}{2}\right)^4 \sin\left(\frac{\pi}{2}x\right)\right]$$

since $\frac{d^3}{dx^3}\frac{x^2}{4} = 0$

$\frac{d^4}{dx^4}(\sin Ax) = A^4 \sin(Ax)$

① $$= \frac{\pi^4}{384}\sin\left(\frac{\pi}{2}x\right)$$

$$e = \frac{\pi^4}{384}\sin\left(\frac{\pi}{2}x\right)\left[(x - x_0)(x - x_1)(x - x_2)(x - x_3)\right]$$

Note error is bound with respect to $-Ax \le e \le Ax$

Since $\sin\left(\frac{\pi}{2}x\right) = \begin{cases} 0 & x = \text{even, whole} \\ 1 & x = \text{odd, whole} \end{cases}$

② we bind error

$$-\frac{\pi^4}{384}\sin\left(\frac{\pi}{2}x\right)\left[(x - x_0)(x - x_1)(x - x_2)(x - x_3)\right] \le e \le \frac{\pi^4}{384}\sin\left(\frac{\pi}{2}x\right)\left[(x - x_0)(x - x_1)(x - x_2)(x - x_3)\right]$$

where X is bound $-A \leq X \leq A$

where $A = \pm 1, \pm 3, \pm 5, \ldots \pm a$

Thus (2) is reduced to

$$-\frac{\pi}{384}(X-X_0)(X-X_1)(X-X_2)(X-X_3) \leq e \leq \frac{\pi^4}{384}(X-X_0)(X-X_1)(X-X_2)(X-X_3)$$

c) Given $X = [0, 2, 3, 4]$

Find X that gives e_{max}
Find X that gives maximum

From b) we know that e_{max} will occur

For some X in $|3 \text{ to } 3.\overline{9}|$ Since 4 will collapse the function to $\emptyset$

and the other data points are smaller in terms of absolute values

Using the non-reduced max since symetry

$$e \leq \frac{\pi^4}{384}\sin\left(\frac{\pi}{2}x\right)\left[(x)(x-2)(x-3)(x-4)\right]$$

Solution From code (attached)

$e_{max} \approx 0.238839$

when

$X = 3.9999$ → Python expression had to go far enough. When the below X value was used directly we showed a larger number 4 decimals after the written values here

Solution From WolFram Alpha

$e_{max} \approx 0.238839$

at

$X = 3.45321$

-6pts: Note you had the bound in part b. Substituting and taking the derivative gives you the inflection points. Your approach could work, but you missed that the error expression has an abs value

3

Tuesday, October 10, 2017 7:26 AM

3. (15 points) We have the following data:

$$x = [1, 2, 3, 4, 5, 6, 7]\,,$$

$$f(x) = [1, 4, 10, 12, 5, 4, 0]\,.$$

(a) (10 points) Using *built in* Python or MATLAB functions, interpolate this data using

- Piecewise linear interpolation
- Lagrange polynomial interpolation
- Spline interpolation

Attached as .txt
Located in repo as well

Create a subplot for each of your interpolants over $[0.75, 6.25]$ using a fine mesh spacing, e.g. 0.05 (note that to use scipy's piecewise linear polynomial interpolation you will need to restrict the range to the exact endpoints, 1.0 and 6.0). Include the data points on the interpolation plots.

(b) (5 points) Briefly discuss the differences between the resulting interpolations.

b)

The Piecewise Linear Interpolation simply connects straight lines between the given points so its not a very good curve. Doesn't extrapolate

The Lagrange Interpolation took a better curve. Its extrapolation is from a -Y to a +Y

The spline interpolation is a nice curve fits like the lagrange but its extrapolation fits better and is not as exaggerated in the Y-axis we can see that it would hit the last given point if it etrapolated further.

4

Tuesday, October 10, 2017 7:28 AM

4. (30 points) The errors generated by a numerical method on a test problem with various grid resolutions have been recorded in the following table:

Grid Spacing (h)	Error (E)
5.00000e-02	1.036126e-01
2.50000e-02	3.333834e-02
1.25000e-02	1.375409e-02
6.25000e-03	4.177237e-03
3.12500e-03	1.103962e-03
1.56250e-03	2.824698e-04
7.81250e-04	7.185644e-05
3.90625e-04	1.813937e-05

For this numerical method, the error should be of the form

$$E = kh^p$$

(a) (3 points) Write this problem as a linear system $\mathbf{A}\vec{x} = \vec{b}$, where $x = \begin{pmatrix} \ln(k) \\ p \end{pmatrix}$ is the vector of unknowns.

(b) (5 points) Derive the normal equations for this over-determined system: write the matrices in $\mathbf{A}\vec{x} = \vec{b}$ form, where you include formulas / values for each entry.

(c) (9 points) Solve, using the program/language of your choice, the normal equations to obtain a least squares estimate to the parameters k and p.

(d) (6 points) Solve for the parameters k and p using SciPy's CurveFit function

(e) (7 points) Make a log-log and a lin-lin plot that displays both the input data and the function $E = kh^p$. Comment on the differences between the two approximations. (Checkout Python's matplotlib.pyplt.loglog command.)

BONUS (5 points): submit your code by providing read/clone access to an online version control repository where your code is stored (e.g. github or bitbucket). If you don't know what that means and want to learn about it, come talk to me or check out resources here: `http://software-carpentry.org/lessons.html` For Windows, you want to setup the Git GUI. NOTE: You will not be able to do this from AFIT's systems.

a) given $E = kh^p$ write this as a linear function

$A\vec{x} = \vec{b}$ wh $\vec{x} = \begin{bmatrix} \ln(k) \\ p \end{bmatrix}$ E is a function of h

same constant

First attempt to match $\vec{x}$

$$\ln(E) = \ln(kh^p)$$

$$\ln(E) = \ln(k) + \ln(h^p)$$

$$= \ln(k) + p\ln(h) = A\vec{x}$$

$$\ln(E) = p\ln(h) + \ln(k)$$

Didn't write matrix form, but you have it below...

b) Find the normal equations -5pts: See pg 7-8 of lesson 2 notes

note $Y = \ln(E)$
$a_1 = \rho$
$X = \ln(h)$
$a_0 = \ln(K) \Rightarrow e^{a_0} = K$

$$\begin{bmatrix} A & \cdots \\ \vdots & \end{bmatrix} \begin{bmatrix} \ln(K) \\ \rho \end{bmatrix} = \begin{bmatrix} 1 & \ln(h) \end{bmatrix} \begin{bmatrix} \ln(K) \\ \rho \end{bmatrix}$$

Thus

$$\begin{bmatrix} 1 & \ln(h_0) \\ 1 & \ln(h_1) \\ \vdots & \vdots \\ 1 & \ln(h_n) \end{bmatrix} \begin{bmatrix} \ln(K) \\ \rho \end{bmatrix} = \begin{bmatrix} \ln(E_0) \\ \ln(E_1) \\ \vdots \\ \ln(E_n) \end{bmatrix}$$

This is part a

Normal equations where h_0 to h_7 and E_0 to E_7 are found in the given table

$$\begin{bmatrix} 1 & \ln(h_0) \\ 1 & \ln(h_1) \\ 1 & \ln(h_2) \\ 1 & \ln(h_3) \\ 1 & \ln(h_4) \\ 1 & \ln(h_5) \\ 1 & \ln(h_6) \\ 1 & \ln(h_7) \end{bmatrix} \begin{bmatrix} \ln(K) \\ \rho \end{bmatrix} = \begin{bmatrix} \ln(E_0) \\ \ln(E_1) \\ \ln(E_2) \\ \ln(E_3) \\ \ln(E_4) \\ \ln(E_5) \\ \ln(E_6) \\ \ln(E_7) \end{bmatrix}$$

c) Solving the normal equations

ref "Numerical Methods" by Gilat and Subramaniam

$Y = a_1 x + a_0$ $\quad a_1 = P \quad a_0 = \ln(K)$

$\ln E = P \ln(h) + \ln(K)$ $\quad x_i = \ln(h_i) \quad Y_i = \ln E_i$

given

$$E = \sum_{i=1}^{n} [Y_i - (a_1 x_i + a_0)]^2$$

we know the solutions will take this form for a given linear function

6.19) $$a_1 = \frac{n S_{xy} - S_x S_y}{n S_{xx} - (S_x)^2} \qquad a_0 = \frac{S_{xx} S_y - S_{xy} S_x}{n S_{xx} - (S_x)^2}$$

where

n = terms or data points

$$S_x = \sum_{i=0}^{n} x_i \qquad S_y = \sum_{i=0}^{n} y_i \qquad S_{xy} = \sum_{i=0}^{n} x_i y_i \qquad S_{xx} = \sum_{i=0}^{n} (x_i)^2$$

Using a program

$a_0 = \ln(K) \Rightarrow e^{a_0} = K$

$K \approx .99416$

$a_1 = P \approx 2.0397$

-1pts: See pg 7-8 of lesson 2 course notes. This error follows from part b, so difficult to evaluate. Answer is really different, but so is scipy curve fit

See program output for exact values

d) Using Curve Fit

$K \approx 11.4786$

$P \approx 1.5722$

e) The log-log plot is essentially straight with a slight crook in its latter half indicating our data is governed by the Power Law

The standard linear-linear plot is like an exponential function drawn with piecewise interpolation

-4 pts: Plots don't show comparison of the input data and the function; not labeled; lack of comparison data on the plot limits ability to comment on the differences.