AERO-222: Introduction to Aerospace Computation – Fall 2021
Homework #3 – Due Date: October 20, 2021

Show all work and justify your answers!

Instructions

- *This homework contains both handwritten and coding problems and shall be submitted according to the following guidelines.*
- *Hardcopy:*
  - *Due on CANVAS at 11:59 PM on the day of the deadline.*
  - *Shall include screenshots of any hand-written work.*
  - *For coding problems, the hardcopy shall include any relevant derivations and emphasize the final results (i.e. boxed, highlighted, etc.). INCLUDE ALL CODING RESULTS (including plots, final values) IN THE HARDCOPY.*
  - *Shall be submitted as a single file according to the provided template with the following naming scheme:* "LastnameHW#.pdf"
- *Coding Submission:*
  - *Due on CANVAS at 11:59 PM on the day of the deadline.*
  - *Shall be submitted as a single file according to the provided template with the following naming scheme:* "LastnameHW#.py"
  - *The script shall print out all outputs asked for in the problem.*
- *Late submissions will be accepted with a 10 point deduction per day late.*

---

1. **Least-squares (20 pts) Code.** Create a set of $N = 100$ points, x_k, uniformly distributed in the $[-1, +1]$ range. Compute the function $f(x) = \sin(3x)$ at the x_k data points. Corrupt the $f(x_k)$ values by adding a Gaussian noise with standard deviation $\sigma = 0.08$. Plot $f(x_k)$ and $f_c(x_k)$. Perform the least-squares of $f_c(x_k)$ using the fitting functions,

$$f_1(x) = c_1 + c_2\, x + c_3\,(2x^2 - 1) + c_4\,(4x^3 - 3x)$$
$$f_2(x) = c_1\, \sin x^2 + c_2\,(1 - \cos x) + c_3\,(\cos x\, \sin x) + c_4\, \frac{2 - x}{2 + x}$$

by computing the c_k coefficients for each function. Plot the estimated functions, $f_1(x)$ and $f_2(x)$, and compute the 1-norm, the 2-norm, and the ∞-norm of the two residuals vectors.

Solution: Here are the results I got. The numbers may vary a little depending on how they corrupted the data and determined the derivatives:

Norm	1	2	∞
f_1	4.86	0.6	0.2
f_2	22.7	2.64	0.6

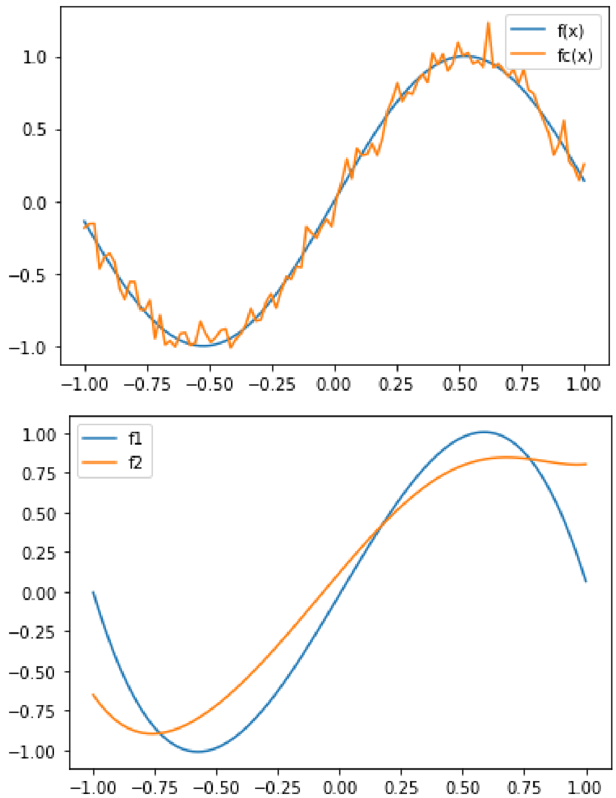

Figure 1: Problem 1 Results

2. Interpolation (20 pts) by hand. This problem compare Lagrange interpolation with interpolation using linear combination of different functions. Consider the points,

x_k	1	2	3	5
y_k	-1	1	0	1

Perform the interpolation at $x = 4$ using Lagrange polynomials and using the function,

$$f(x) = c_1 \sin x + c_2 (1 - x^2) + 5\, c_3 \frac{\cos x}{x+1} + 3\, c_4$$

Solution: Use the 4th Legrange polynomial for 4 values, n = 3:

$$L_n(x) = [\sum_{k=1}^{n+1} (\Pi_{k!=i} \frac{x - x_i}{x_k - x_i})] y_k$$

Lagrange Solution: -1.0
Least squares solution: 1.47
These seemed weirdly different, so if theres a trend of different answers just let me know.

3. **System of nonlinear equations (20 pts) Code.** Solve the following nonlinear system,

$$\begin{cases} 3\,z - \sin(xy) - 1/8 = 0 \\ x^2 - 19\,(y + 0.1)^3 + \cos z = 0 \\ e^{-y} + 20\,x\,z^2 + \pi/4 = 0 \end{cases}$$

using $\boldsymbol{v}_0 = \{x_0, y_0, z_0\}^{\mathrm{T}} = \{0.1, 0.1, -0.1\}^{\mathrm{T}}$ as initial guess. Iterates until $\|\boldsymbol{v}_{k+1} - \boldsymbol{v}_k\|_2 < 10^{-9}$. Generate a plot of $\|\boldsymbol{v}_{k+1} - \boldsymbol{v}_k\|_2$ as a function of the number of iterations.

Solution: Function vector and Jacobian matrix are:

$$\boldsymbol{F} = \begin{Bmatrix} 3z - \sin(xy) - 1/8 \\ x^2 - 19(y + 0.1)^3 + \cos z \\ e^{-y} + 20xz^2 + \pi/4 \end{Bmatrix} \quad \text{and} \quad J = \begin{bmatrix} -y\sin(xy) & -x\sin(xy) & 3 \\ 2x & -57(y + 0.1)^2 & -\sin z \\ 20z^2 & -e^{-y} & 40xz \end{bmatrix}$$

Newton iteration is:

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k - J_k^{-1}\,\boldsymbol{F}_k$$

4. **System of nonlinear equations (20 pts) Code.** Solve the following nonlinear system using iterative Newton,

$$\begin{cases} f_1(x, y) = x^2 - y^2 - 1 = 0 \\ f_2(x, y) = -x^2 + 2\,y^3 - e^{-x\,y} + 1 = 0 \end{cases}$$

using $x_0 = 3$ and $y_0 = 4$ as initial guess. 1) How many iterations do you need to obtain the 2-norm of $\begin{Bmatrix} x_k - x_{k-1} \\ y_k - y_{k-1} \end{Bmatrix} < 10^{-6}$. 2) Plot the following table for 10 iterations

iteration	x_k	y_k	$f_1(x_k, y_k)$	$f_2(x_k, y_k)$
1	???	???	???	???
⋮	⋮	⋮	⋮	⋮
10	???	???	???	???

Solution: a) I found a solution took 10 iterations using iterative Newton and finite differences to determine the Jacobian. Answers and the table may be ever so slightly different due to machine precision or calculating the Jacobian via the analytical derivatives for $f_1(x_k, y_k)$ and $f_2(x_k, y_k)$.

iteration	x_k	y_k	$f_1(x_k, y_k)$	$f_2(x_k, y_k)$
0	3	4	-8	120
1	2.63636	2.72727	-1.4876	34.6198
2	2.04363	1.88157	-0.363882	10.1248
3	1.62067	1.32548	-0.130336	2.91422
4	1.3828	0.985477	-0.0590228	0.746017
5	1.29174	0.827751	-0.0165848	0.122447
6	1.27576	0.7928	-0.00096633	0.00533324
7	1.27516	0.791218	-2.1383e-06	1.07499e-05
8	1.27515	0.791214	-8.63398e-12	4.25471e-11
9	1.27515	0.791214	-3.33067e-16	3.33067e-16

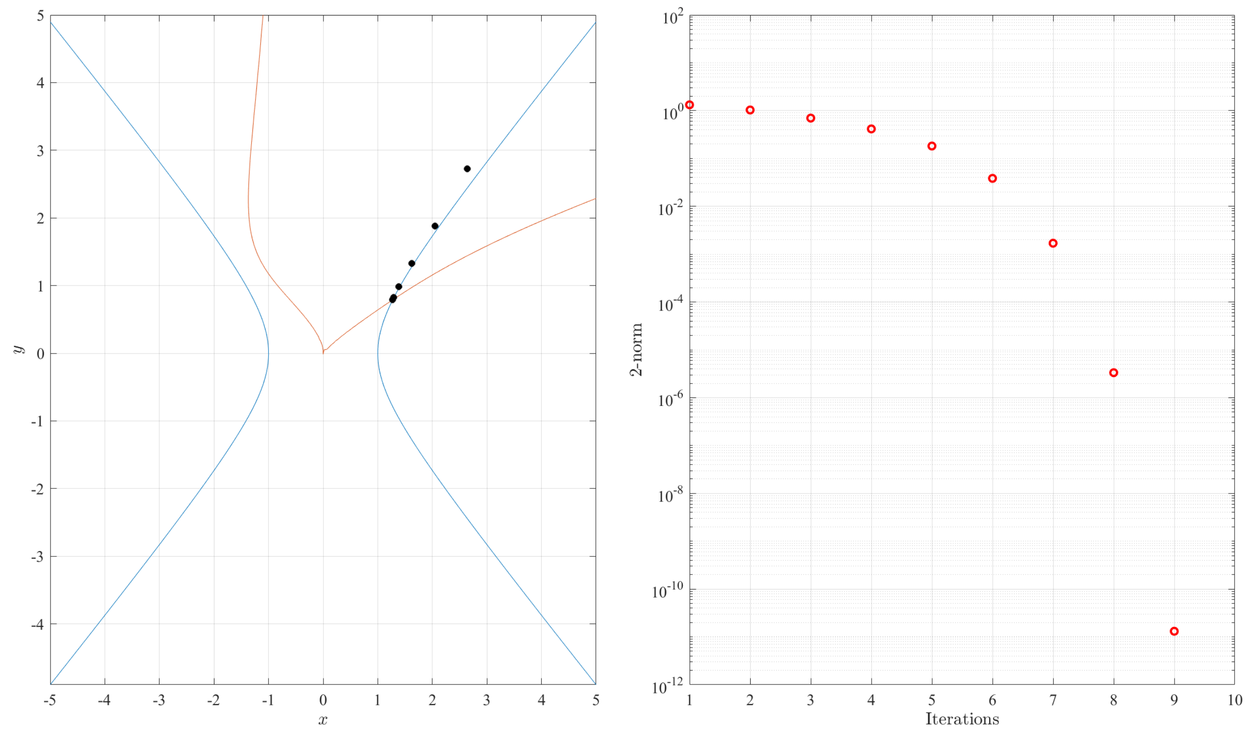

Figure 2: Results

5. Least-squares (20 pts) by hand. Show how to estimate φ, ξ, and λ by linear least-squares using the n set of data points, $[x_k, y_k]$, and the function,

$$y(x) = \sin(3x + \varphi) - x + x^2 - x^3 + \xi\,\cos(2x - \lambda)$$

Solution: Expand the sin function and introduce the new variables, α and β,

$$\sin(3x + \varphi) = \sin(3x)\cos\varphi + \cos(3x)\sin\varphi = \sin(3x)\,\alpha + \cos(3x)\,\beta$$

Then, expand the cos function and introduce the new variables, γ and δ,

$$\cos(2x - \lambda) = \cos(2x)\cos\lambda + \sin(2x)\sin\lambda = \cos(2x)\,\gamma + \sin(2x)\,\delta$$

Then, the least-squares is set in terms of the new variables and performed by introducing the arrays,

$$A = \begin{bmatrix} \sin(3x_1), & \cos(3x_1), & \cos(2x_1), & \sin(2x_1) \\ \sin(3x_2), & \cos(3x_2), & \cos(2x_2), & \sin(2x_2) \\ \vdots & \vdots & \vdots & \vdots \\ \sin(3x_n), & \cos(3x_n), & \cos(2x_n), & \sin(2x_n) \end{bmatrix} \quad \text{and} \quad \boldsymbol{v} = \begin{Bmatrix} x_1 - x_1^2 + x_1^3 \\ x_2 - x_2^2 + x_2^3 \\ \vdots \\ x_n - x_n^2 + x_n^3 \end{Bmatrix}$$

with the classic least-squares solution of α, β, γ, and δ,

$$\begin{Bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{Bmatrix} = \left(A^{\mathrm{T}} A\right)^{-1} A^{\mathrm{T}} \boldsymbol{v}$$

once compute α, β, γ, and δ, then the original variable, φ, is computed as,

$$\begin{cases} \alpha = \cos\varphi \\ \beta = \sin\varphi \end{cases} \qquad \to \qquad \varphi = \mathrm{atan2}(\beta, \alpha)$$

and, similarly, the least-squares value of λ is given by,

$$\begin{cases} \gamma = \cos\lambda \\ \delta = \sin\lambda \end{cases} \qquad \to \qquad \varphi = \mathrm{atan2}(\delta, \gamma)$$