Homework 2

Matrices, Interpolation, and Curve Fitting

MATH 210-010 ⋄ Fall 2024

September 28, 2024

Due: Sunday, September 29, 2024

Instructions: To complete a problem set, you must submit a zip file labeled `Yourlastname_HW#` to Dropbox no later than 11:59 PM on the due date above. For example, if I were to complete this assignment, my folder would be named `Emerick_HW2`. In this folder, a `py` file is to be submitted for each problem such that when the `py` file is executed, the output (as presented in Python) is the solution to the problem. Each `py` file must be saved as `Yourlastname_HW#_No#.py`. For example, if I were submitting the answer to Question Number 1 on Homework 1, the `py` file for that problem would be saved as `Emerick_HW2_No1.py`. Each `py` file should be well commented and be free of extraneous lines and commands. Also, each `py` file must output only what the problem asked to be outputted. Failure to abide by these simple homework submission guidelines may result in a deduction of points at my discretion.

Name:

Score:

For each problem below submit a separate `py` file with an initial comment that describes the objective of the `py` file. Always remember to begin your `py` file with `import numpy as np` and `import matplotlib.pyplot as plt`. For some of the problems below, use `import scipy.linalg as LA`. Also, for any problems that require a plot, the title size, label size, etc should follow the default figure settings discussed in class and on the handout.
import numpy as np import matplotlib.pyplot as plt

1.] Solve the following system of equations by setting up the proper matrix equation and solving it in two ways: using `LA.inv` and `LA.solve`:

$$\begin{aligned} -3x_1 + x_2 + x_3 &= 5 \\ 2x_1 - x_2 + 4x_3 &= 1 \\ x_1 + 3x_2 - 11x_3 &= -2 \end{aligned}$$

Your final output should print the solution vector $\boldsymbol{x}$.

2.] Create an `py` file that defines the following matrices. I want you to explore the built-in functions that are used to create matrices such as these in one or two lines.

a.) Define A as a 30×20 matrix of threes using `np.ones`.

b.) Using `np.reshape` and `np.arange`, define B as the 10×10 matrix:

$$\begin{bmatrix} 1 & 11 & \cdots & 91 \\ 2 & 12 & \cdots & 92 \\ \vdots & \vdots & \ddots & \vdots \\ 10 & 20 & \cdots & 100 \end{bmatrix}.$$

c.) Using `np.diag` and `np.ones` more than once, define C as the 16×16 matrix:

$$\begin{bmatrix} -2 & 1 & 0 & 0 & \cdots & 0 & 1 \\ 1 & -2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & -2 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 & -2 & 1 \\ 1 & 0 & \cdots & 0 & 0 & 1 & -2 \end{bmatrix}.$$

(This is a *differentiation matrix*.) Use the command `plt.spy(C)` to plot the sparsity of this matrix in Figure 1.

d.) Read about `LA.toeplitz` and use it to make the matrix from part *c.*). Call this matrix D.

e.) Use `LA.toeplitz`, `np.triu`, and whatever else to make the following 8×8 matrix:

$$E = \begin{bmatrix} 1 & 2 & 3 & \cdots & 8 \\ 0 & 1 & 2 & \cdots & 7 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 2 \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}.$$

Use the command `plt.spy(E)` to plot the sparsity of this matrix in Figure 2.

f.) Use `LA.toeplitz` and whatever else to create the following 8×8 matrix:

$$F = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{8} \\ \frac{1}{2} & 1 & \frac{1}{2} & \cdots & \frac{1}{7} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \frac{1}{7} & \cdots & \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{8} & \cdots & \frac{1}{3} & \frac{1}{2} & 1 \end{bmatrix}.$$

3.] In this problem you are trying to find an approximation to the periodic function $f(x) = e^{\sin(x-1)}$ over one period, $0 \leq x \leq 2\pi$. In Python, let `x = np.linspace(0, 2*np.pi, int(200))` and `y` be the evaluation of f at those points.

a.) Fit a degree 7 polynomial to these data by using the normal equations and the appropriate truncated Vandermonde matrix A.

b.) Fit the following sinusoidal function

$$f(x) = c_0 + c_1 \cos(x) + c_2 \sin(x) + c_3 \cos(2x) + c_4 \sin(2x)$$

using `scipy.optimize.curve_fit`.

c.) Plot the original function, and the two approximations together on a well-labeled graph.

d.) Compute the R^2 value for each function and print the value for each model.

4.] Suppose we want to fit the function $f(x) = x/(ax + b)$ to a set of data by optimizing the two parameters a and b. Show that for any set of data, fitting the parameters a and b can be transformed into a (non-square) system like $A\boldsymbol{x} = \boldsymbol{b}$, for some matrix A and vector $\boldsymbol{b}$. Given the "toy" data:

```
x_data = np.linspace(1, 10, 50)
y_data = x_data/(2.5*x_data + 1.3) + .01*np.random.normal(0,1,size=len(x_data)),
```

solve the least squares fitting problem in two ways: solving the appropriate normal equations using `LA.solve` and using the built-in function `scipy.optimize.curve_fit`. After getting each solution, compare them by computing the R^2 value for each model.