Lab 12

Ch En 263 – Numerical Tools

Due: 29 Feb. 2024

Instructions

- Complete the exercise(s) below, and submit the following files to Learning Suite:
  - Handwritten portion: scan each page (or take a picture) and combine them into a single pdf named: `LastName_FirstName_Lab12.pdf`
  - Excel portion: submit a workbook named `LastName_FirstName_Lab12.xlsx` where each worksheet tab is named "Problem_1", "Problem_2", etc.
  - Python portion: submit a separate file for each problem named `LastName_FirstName_Lab12_ProblemXX.py` where XX is the problem number.
- Warning: the LS assignment will close promptly at 11:59 pm and late assignments will only receive 50% credit.

Lab Exercises

1. In this problem, we are going to compare the computational cost of using Numpy's methods to solve linear systems.

   (a) Download the file `Lab12Data.zip` from the course website. Inside the zipped folder there are 3 pairs of files: `A_XXXX.dat` and `b_XXXX.dat` where `XXXX` is 1000, 2000, 5000.

   (b) Using the `time` module (see example below), write a code that finds the average time it takes to solve $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}$ for all three pairs of $\boldsymbol{A}$ and $\boldsymbol{b}$ using Numpy's linear algebra solver. For example,

```
import time
t_start = time.time()
for i in range(5):
  # solve for x here
t_end = time.time()
t_avg = (t_end - t_start)/5
```

   Save the times in an array called `t_Numpy`.

   (c) In addition, write a piece of code that finds the average time it takes to solve $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}$ for all three pairs of $\boldsymbol{A}$ and $\boldsymbol{b}$ using Numpy's inverse function. Save the times in an array called `t_Inv`.

   (d) Use your arrays from parts (b) and (c) to make a log-log plot using the matplotlib `plt.loglog(X, Y)` function that plots the average solution time versus the matrix size ($n = 1000, 2000$, or 5000). Also plot the curve $T(n) = 10^{-11} \times n^3$ and add a legend labeling the different curves. What does your plot show about the asymptotic behavior of the method used by the Numpy solvers? How does that compare to the Gauss elimination method we learned? Type your answers in the comments in your code.