MENG 21200 - Principles of Engineering Analysis II
Winter 2023

Molecular Engineering
University of Chicago

Problem Set 2

ASSIGNED: January 13, 2023
DUE: **Friday, January 20, 2023** at the beginning of class.

Problem set guidelines:

1. Each solution must be your own work.
2. All problems that involve python must be completed in Jupyter Notebooks**.** Some problems may not require python and may be better completed with pen and paper.
3. Highlight your final answer when providing numerical results. Provide plots, graphs, and tables of your results when appropriate.
4. **Submission Instructions:** This quarter we will use [Gradescope] to collect your submissions and grade them. We will only accept a single PDF file with your compiled solutions. Please follow the instructions below to obtain a PDF file from a Jupyter Notebook:
    i. Make sure all your code runs without error, and all figures (if any) show up where intended. We will not be running your code, therefore it is essential that your solutions output and highlight your results. Please be mindful of your line length so that it fits into the PDF layout and your results are clearly shown.
    ii. Select `File->Download as->HTML (.html)`. This will download an HTML version `your_homework.html` of your notebook to your computer, typically in your `Download` folder.
    iii. Open `your_homework.html` in your web browser (just double-click on it). ***Use the `File->Print` command of your browser to produce a PDF file.*** *(Do not produce the PDF in other ways, such as "Export to PDF" or other. Alternative methods will usually produce poor page margins, format headers differently, fail to incorporate figures, and so forth.)*
    iv. **Submit your PDF file** `your_homework.pdf` to Gradescope. Do *not* submit your HTML file.
5. Problem sets are due at the beginning of the class period on the due date listed. Late problem sets will not be accepted.

Note that you can wrap lines of python code using the "\" operator to ensure that all your code is visible within the width of the page. See the following examples:

In [1]:
```python
print("demonstra\
tion")
```
```
demonstration
```

In [2]:
```python
def factorial(n):
    if n==1:
        return(1)
    return(n*\
        factorial(n-1))

factorial(4)
```
Out[2]: 24

In [3]:
```python
factorial\
(4)
```
Out[3]: 24

Problem 1 – Comparing finite-difference methods of various orders

Adapted from C&C problem 23.19

(a) By hand, compute the value of the first derivative of $f(x) = e^{-2x} - x$ at $x = 2$.

(b) Evaluate the first- and second-order centered finite-difference approximation for the first derivative of $f(x)$. Start with $\Delta x = 0.5$, *i.e.*, using $x = 2 \pm 0.5$ for the first-order approximation and $x = 2 \pm (2 * 0.5)$ for the second-order approximation. Repeat this calculation for a range of Δx down to 0.01. Plot the approximations as a function of Δx. Include a horizontal line at the true value you computed in part (a).

(c) Repeat (b) for the forward-difference and backward-difference (remember now to use up to $x = 2 \pm 2\Delta x$ for the two methods). Plot the three first-order methods together and the three second-order methods together side by side and compare. Plot the first-order centered difference against the second-order forward and backward difference approximations. How do they compare? Include legends in your plots. In what qualitative ways do the three methods differ?

(d) Plot the relative and absolute errors side-by-side in a figure. Again, include a legend. What information does this give you? What is the order of the error in forward difference approximation, in terms of Δx? What is the order in backward and centered approximations?

Problem 2 - Numerical differentiation applied to discrete data

Adapted from C&C problem 24.44

A jet fighter's position on an aircraft carrier's runway was timed during landing:

t /s	0	0.52	1.04	1.75	2.37	3.25	3.83	4.35	4.87
x /m	153	185	208	249	261	271	273	290	318

where x is the distance from the end of the carrier. Estimate (a) velocity (dx/dt) and (b) acceleration (dv/dt) using first-order and second-order forward difference numerical differentiation. Plot the position, the velocity, and the acceleration of the jet fighter versus time side-by-side in a figure.

Problem 3 – A tired graduate student

Focused Ion Beams (FIB) are common tools for making nano-photonic devices. A very tired graduate student forgets to calibrate the FIB before patterning her device and discovers that the ions have been implanted, thus creating an undesired residual electric field. She can measure the field, and uses this data to develop a model:

$$\boldsymbol{\vec{E}(x,y) = E_x\vec{x} + E_y\vec{y} = (x^3 - 4x^2 + 2x^2y)\vec{x} + (2y^3 + 2y^2 + 1.5y)\vec{y}}$$

in units of $\boldsymbol{nV/\mu m}$. This (now extremely tired) graduate student is interested in calculating the distribution of ions in her device, so that she can correctly calibrate the FIB in the future. Luckily the material is isotropic and linear, so that she can write the free charge using Gauss's Law:

$$\vec{\nabla} \cdot \vec{E} = \frac{\partial E_x(x,y)}{\partial x} + \frac{\partial E_y(x,y)}{\partial y} = \frac{\rho_{free}(x,y)}{\varepsilon}$$

(a) First, generate a contour plot of the electric field magnitude and the electric vector field side by side over the domain $[(-5,-5),(5,5)]$. Do not forget to include a color bar, as well as correct units for all three axes. Are these electric fields large, relative to human scale (Hint: plug order of magnitude numbers into Wolfram Alpha for some fun comparisons)?

(b) Using second-order centered finite difference and an appropriate Δx, calculate the free charge density (ρ_{free}), assuming the relative permittivity (ε) of this material is approximately 3.9 (a gold star for those who can correctly guess the material).

Problem 4 - Lunchtime experiments

The Very Tired Graduate Student brought a loaf of bread to the ERC so that they could prepare a sandwich for lunch each day of the week. They left the bread in the lab refrigerator but did not realize that the refrigerator was broken. Unfortunately, the bread began to spoil over time.

A group of undergrads observed this, and rather than notifying The Very Tired Graduate Student, they decided to characterize the growth of microorganisms in the bread over time. The data that each student collected is provided below. It is organized by undergrad student number, with each undergrad reporting the time (in hours) and measured bacteria count per slice of bread (in thousands). For example, Undergrad 1 counted 1.7 thousand bacteria at 1.2 hours.

Undergrad 1						
Time (hours)	1.2	19.6	28.2	40.4	50.2	60.0
Bacteria Count (thousands)	1.7	1.7	3.6	11.2	22.2	44.7
Undergrad 2						
Time (hours)	13.5	25.7	34.2	44.1	55.1	58.8
Bacteria Count (thousands)	0.16	1.2	5.2	9.9	31.3	39.9
Undergrad 3						
Time (hours)	4.9	11.0	26.9	36.7	47.8	53.9
Bacteria Count (thousands)	2.5	2.7	4.7	4.7	15.8	28.8
Undergrad 4						
Time (hours)	2.4	18.4	29.4	38.0	42.9	57.6
Bacteria Count (thousands)	0.3	1.6	3.2	9.0	15.7	40.0
Undergrad 5						
Time (hours)	6.1	24.5	31.9	39.2	45.3	52.7

Bacteria Count (thousands)	1.2	5.1	5.3	6.7	16.8	25.4
Undergrad 6						
Time (hours)	8.6	22.0	33.1	41.6	51.4	56.3
Bacteria Count (thousands)	2.5	6.7	7.2	11.8	23.1	36.3

(a) Create a publication-quality scatterplot of the data for bacteria count versus time. Each undergrad's data should be presented using a different color.

(b) The undergraduates anticipate that the bacteria count can be modeled using an exponential function. For <u>only</u> the data from Undergrad 1, find the parameters of the exponential model using a linear regression.

(c) Compare the linearly regressed model expression to the results from scipy's `curve_fit` function. Plot the original data along with the fitting of both exponential models (*i.e.*, from the linearized regression and `curve_fit`) for Undergrad 1

(d) Which of the two approaches yields a better fit? Explain <u>how</u> these two approaches fit the empirical data to the exponential model (i.e., what is the numeric computation being performed) and <u>why</u> different outcomes result.

(e) As more data points are collected, it might be expected that the difference between the best fit model expressions from the linear regression and `curve_fit` become smaller. The undergrads decide to test this out.

For the `curve_fit` regression, plot the standard error of the estimate versus n, where the fitting is performed for the datasets provided by the first n undergraduates (e.g., when n=3, use the data from Undergrads 1, 2, and 3). The standard error of the estimate is given as

$$s_{y/x} = \sqrt{\frac{S_R}{m-2}}$$

where S_R is the sum of squared residuals and m is the number of data points that have been regressed. Briefly explain your observations.

Assuming the best quality fit is achieved using the `curve_fit` algorithm applied to the data from all the undergrads, plot the absolute error in the exponential rate constant computed from the linearized regression as a function of n.

(f) Using the fitting parameters from the `curve_fit` algorithm applied to the data from all the undergrads, provide the exponential model equation.

(g) The Very Tired Graduate Student accidentally ate a slice of the bread at the 40-hour mark. What was the bacteria count at that time?

Problem 5 – Have we seen this before?

Adapted from C&C problem 24.40

The rate of cooling of a body can be expressed as

$$\frac{dT}{dt} = -k(T - T_a)$$

where T = temperature of the body (°C), T_a = temperature of the surrounding medium (°C), and k = a proportionality constant (per minute). Thus, this equation (called Newton's law of cooling) specifies that the rate of cooling is proportional to the difference in the temperatures of the body and of the surrounding medium. If a metal ball heated to 80°C is dropped into water that is held constant at $T_a = 20$ °C, the temperature of the ball changes, as in

Time, min	0	5	10	15	20	25
T, °C	80	44.5	30.0	24.1	21.7	20.7

(a) Plot dT/dt versus $T - T_a$.
(b) Determine k by applying an appropriate regression method.
(c) Plot T versus time (t) including the measured data as discrete data points and the model fit as a continuous curve.