MENG 21200 - Principles of Engineering Analysis II ***Molecular Engineering***
Winter 2023 ***University of Chicago***

Problem Set 4

ASSIGNED: January 27, 2023
DUE: **Friday, February 3, 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]:
```
print("demonstra\
tion")
```
```
demonstration
```

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

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

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

Problem 1

Adapted from "Effects of Particle Size on Mg^{2+} Ion Intercalation into λ-MnO2 Cathode Materials", Nano Letters, 19(7), 4712-4720 (2019). https://pubs.acs.org/doi/10.1021/acs.nanolett.9b01780

The Very Tired Graduate Student is studying new materials for batteries and would like to determine the effect of particle size on Mg^{2+} ion intercalation into λ-MnO_2 cathode materials. The Very Tired Graduate Student utilizes cyclic voltammetry (CV) measurements under 5 different scan rates ν (2 mV/s, 5 mV/s, 10 mV/s, 20 mV/s, and 50 mV/s) for two kinds of particles: small-$LiMnO_2$ particles (S-LMOPs) and big-$LiMnO_2$ particles (B-LMOPs) to compare their performance.

Based on the literature, the current i (in A/g) measured in CV curves includes contributions from a diffusion-limited Faradaic process and a capacitive response. For this study on the effect of particle size, the relation between the current i and the scan rate ν at a fixed voltage V (in V) is expressed by a linear combination of two power laws:

$$i(V) = a_1\nu + a_2\nu^{0.5}$$

where a_1 and a_2 are voltage-dependent parameters, and $a_1\nu$ and $a_2\nu^{0.5}$ are the currents from the capacitive response and diffusion-limited Faradaic process, respectively.

(a) Part of The Very Tired Graduate Student's experimental (V at 10 mV/s) and processed data (a_1 and a_2) for S-LMOPs is listed below. Plot the contributions to the current from the Faradaic intercalation (red curve) and capacitive response (blue curve) in comparison to the total current (black curve) in the same figure. Indicate the anodic scan as data points connected with solid lines and the cathodic scan as data points connected with dashed lines.

(b) Usually the area within the CV loop (i.e., between the anodic and cathodic curves) represents the capacity of a battery material to store charge (i.e., capacity = current x voltage). Calculate <u>using Simpson's rules</u> the percentage contribution to the total charge storage capacity attributed to Faradaic intercalation.

(c) Given the data below at ν =10 mV/s and again <u>using Simpson's rules</u>, what is the total charge storage capacity for the B-LMOPs? How does it compare to the S-LMOPs? How does your conclusion based on your analysis compare to that of the authors?

```
# Available data for S-LMOPs for anodic scan
V_sa = [-0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2]
a1_sa = [-0.193, -0.0641, -0.00884, -0.00135, -0.00233, -0.00638,
-0.00984, -0.0075, -0.00255, 0.00511, 0.294]
a2_sa = [-0.409, -0.055, -0.0216, -0.00592, 0.0122, 0.0526,
0.213, 0.569, 1.14, 1.75, 0.461]

# Available data for S-LMOPs for cathodic scan
V_sc = [-0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2]
a1_sc = [-0.193, -0.139, -0.121, -0.056, -0.0163, -0.0241, -
0.0373, 0.0265, 0.0748, 0.121, 0.294]
a2_sc = [-0.409, -0.432, -0.386, -0.669, -0.749, -0.583, -0.519,
-0.421, -0.348, -0.155, 0.461]
```

```
# Available data for B-LMOPs for anodic scan
V_ba = [-0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2,
1.4]
a1_ba = [-0.0197, -0.00307, -0.000654, -0.0000707, -0.0000663, -
0.000131, -0.00105, 0.00229, 0.0026, -0.0074, 0.00158, 0.00144]
a2_ba = [-0.106, -0.0362, -0.00174, 0.000157, 0.0000338,
0.000332, 0.0116, 0.0283, 0.0649, 0.182, 0.225, 0.297]

# Available data for B-LMOPs for cathodic scan
V_bc = [-0.8, -0.6, -0.4, 0, 0.4, 0.8, 1.0, 1.2, 1.4]
a1_bc = [-0.0197, -0.00174, 0.00204, 0.00486, 0.00352, 0.000425,
-0.000795, 0.000963, 0.00144]
a2_bc = [-0.106, -0.15, -0.111, -0.0796, -0.0524, 0.0337, 0.08,
0.151, 0.297]
```

Problem 2 – Evaluation of the Error Function

Adapted from Chapra and Canale, Problem 22.14

There is no closed form solution for the error function:

$$\operatorname{erf}(a) = \frac{2}{\sqrt{\pi}} \int_0^a e^{-x^2} dx$$

(a) Use the two-point Gauss quadrature approach to estimate $\operatorname{erf}(1.5)$. Note that the exact value is 0.966105.

(b) Apply an existing function in python that applies Gauss quadrature to integrate the error function to $a = 1.5$.

Problem 3 - Galactic disc

Consider a disc of space dust and rocks of radius $R = 100$ m and total mass $M = 10{,}000$ kg. Assume that the disc can be approximated as infinitesimally thin. Next consider a 1 kg point mass placed directly above the center of the disc in a coordinate frame with origin located on the center of the disc and z-axis perpendicular to the plane of the disc. The force experienced by the point mass under the gravitational attraction of the disc resolved in the z-direction is given by:

$$f_z(z) = G\sigma z \int_{-R}^{R} dx \int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} dy \frac{1}{(x^2+y^2+z^2)^{3/2}}$$

where $G = 6.67408 \times 10^{-11}\ \mathrm{m^3 kg^{-1} s^{-2}}$ is the gravitational constant and σ is the mass per unit area of the disc.

(a) Integrate the above expression over the range $z = [50, 1000]$ m using `scipy.integrate.dblquad`.

(b) Make a plot of $f_z(z)$ in pN over the same range of z.

At large z, the integral can be approximated as $\widetilde{f_z}(z) = \frac{G\sigma\pi R^2}{z^2}$.

(c) Make a plot of $\widetilde{f_z}(z)$ and $f_z(z)$ in the same plot. Explain any discrepancy that you observe.

Problem 4 – Comparison of integration methods

(a) Write your own code to implement Euler's method: `euler(fn, xi, xf, y0, h)` where `fn` is a differential function $\boldsymbol{dy/dx}$ to be solved, `xi` is the initial x value, `xf` is the final x value, `h` is the step size, and `y0` is the initial value. Your `euler` function should return a list of all the computed y values from `xi` to `xf` inclusive.

(b) Write your own code to implement Heun's method: `heun(df, xi, xf, y0, h)` where `fn` is a differential function $\boldsymbol{dy/dx}$ to be solved, `xi` is the initial x value, `xf` is the final x value, `h` is the step size, and `y0` is the initial value. Your `heun` function should return a list of all the computed y values from `xi` to `xf` inclusive.

(c) Given the function $\boldsymbol{y' = \sin(x) - y}$ and initial condition of $\boldsymbol{y(x = 0) = 1.7}$, solve with the Euler's and the Heun's methods for the domain $\boldsymbol{x = [0, 2\pi]}$ using step sizes that correspond to $\boldsymbol{[2\,, 20, 200, \ldots, 2 \times 10^6]}$ number of steps. Also solve using the RK23, RK45, and LSODA methods via the python library function `scipy.integrate.solve_ivp`. Remember that these library methods are <u>adaptive</u> methods, and automatically select the step size to be taken.
The analytical solution to this differential equation is $\boldsymbol{y = 2.2e^{-x} + sin(x)/2 - cos(x)/2}$. Plot this analytical solution for $\boldsymbol{y}$ versus $\boldsymbol{x}$, the $\boldsymbol{\pi/10}$ step size solutions from your Euler's and Heun's methods, and the `solve_ivp` solutions, on a single plot. Be sure to label axes, provide a title, make each data set its own color, and provide a legend.

(d) How does the error associated with each method scale with the step size taken? To understand the effect of step size in these methods, plot in a single figure the absolute value of the fractional relative true error at $\boldsymbol{x = 2\pi}$ as a function of step size for all 5 numerical methods. Plot using the average step size for the adaptive methods.
Note that you may find it beneficial to plot your results on a semilog or log-log scale. Explain the differences observed in the global errors between the methods.

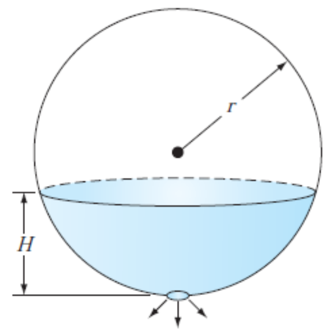
Problem 5

Adapted from Chapra and Canale, Problem 25.20

A spherical tank has a circular orifice in its bottom through which the liquid flows out. The flow rate through the hole can be estimated as

$$Q_{out} = CA\sqrt{2gH}$$

where Q_{out} = outflow (m^3/s), C = an empirically-derived coefficient, A = the area of the orifice (m^2), g = the gravitational constant (= 9.81 m/s^2), and H = the depth of liquid in the tank. <u>Implement and use the fourth-order Runge-Kutta numerical method</u> to determine how long it will take for the water to flow out of a 3 m diameter tank with an initial height of 2.75 m. Note that the orifice has a diameter of 3 cm and C = 0.55.



Problem 6

The following equations developed by the American meteorologist Edward Lorenz are a simple model for atmospheric fluid dynamics:

$$\frac{dx}{dt} = -\sigma x + \sigma y$$

$$\frac{dy}{dt} = rx - y - xz$$

$$\frac{dz}{dt} = -bz + xy$$

Lorenz developed these equations to relate the intensity of atmospheric fluid motion, x, to temperature variations y and z in the horizontal and vertical directions, respectively. These equations are nonlinear as provided from the simple multiplicative terms (xz and xy).

Use numerical methods to obtain solutions for these equations, with $\sigma = 10$, $b = 2.666667$, $r = 28$, and initial conditions of $(x, y, z) = (5, 5, 5)$, then

(i) Plot the results to visualize how the dependent variables change temporally for t from 0 to 20 (*i.e.*, provide plots of x, y, and z versus t).

(ii) What differences do you observe if the initial conditions are slightly perturbed to $(x, y, z) = (5.001, 5, 5)$?

(iii) In addition, plot the dependent variables versus each other to see whether any interesting patterns emerge (*i.e.*, plot in 3D with axes of x, y, and z).