AMATH 301 – Winter 2023
Homework 3

Due: 11:59, January 27, 2023.

Instructions for submitting:

- Coding problems are submitted to Gradescope as a python script (`.py` file) or a Jupyter Notebook (`.ipynb` file) (warning, this may not work as well as a python script). You have **8** attempts (separate submissions to Gradescope) for the coding problems.
- Writeup problems are submitted to Gradescope as a single `.pdf` file that contains text and plots. Put the problems in order and label each writeup problem. When you submit, **you must indicate which problem is which on Gradescope**. Failure to identify pages on Gradescope will lead to a **5% grading penalty. All code you used for this part of the assignment should be included at the end of your .pdf file. Failure to do so results in a 25% penalty.**

Coding problems

This section is worth 10 points, with all variables having equal weight.

1. In order to determine the half life of Plutonium-239, scientists start with a sample of approximately 100 kg of Plutonium-239 and measure the remaining amount each year for 40 years. The data is contained in the file `Plutonium.csv` which is included with the homework. `Plutonium.csv` contains a matrix M whose first row contains the number of years since the beginning of the experiment and whose second row contains the corresponding amounts of Plutonium-239 remaining measured in kg. In what follows I will call the time array `t` and the Plutonium array `P`. Let the function $P(t)$ denote the amount of remaining Plutonium as a function of time.

   (a) Compute the step size, $h = \Delta t$, for this problem. Note that we have Δt instead of Δx now because the variable is t. Save the step size, h, to the variable `A1`.

   (b) Use the first-order forward-difference formula to approximate the derivative $\frac{dP}{dt}$ at time $t = 0$. Save the result to the variable `A2`.

   (c) Use the first-order backward-difference formula to approximate the derivative $\frac{dP}{dt}$ at time $t = 40$. Save the result to the variable `A3`.

(d) We will now be using second-order finite-difference approximations. Use the second-order forward-difference formula,

$$f'(x) = \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} + \mathcal{O}(h^2) \tag{1}$$

to approximate the derivative $\frac{dP}{dt}$ at time $t = 0$. Save the result to the variable `A4`.

(e) We now want to use a similar second-order (backward-difference) finite difference scheme at the right endpoint. You will need to derive this formula. Follow what you did in the activity, along with the work in the January 20 notes where (1) is derived, to find a finite difference method which uses the points $f(x), f(x-h)$, and $f(x-2h)$ and is a second-order approximation for $f'(x)$. Use the method you derived to approximate $\frac{dP}{dt}$ at $t = 40$. Save your answer to the variable `A5`.
Hint: Break out the pen and paper and actually write this down: don't guess. This kind of thing is expected of you for the midterm.

(f) Combine the results from `A4` and `A5` along with the second-order central-difference scheme on the interior (at all times where $t \neq 0$ and $t \neq 40$) to approximate the derivative $\frac{dP}{dt}$ at all 41 times in `t`. Save the resulting array to the variable `A6`.

(g) The decay rate of Plutonium-239 at a time t is given by $-\frac{1}{P}\frac{dP}{dt}$. Use your result from `A6` to estimate the decay rate ($-\frac{1}{P}\frac{dP}{dt}$) at all 41 times in `t`. Save your result to the variable `A7`.

(h) To get the true decay rate we average over the 41 times. Save the average (mean) of these decay rates to the variable `A8`.

(i) If λ is the average decay rate that you found in part (c), then the half life of Plutonium-239, denoted by $t_{1/2}$, is given by the formula

$$t_{1/2} = \frac{\ln(2)}{\lambda}.$$

Calculate the half life, and save it to the variable `A9`.

(j) Next we want to calculate the second derivative, $\frac{d^2P}{dt^2}$ which represents the rate at which the decay rate is increasing or decreasing. Using the functions $f(x+h)$, $f(x)$, and $f(x-h)$, derive a finite-difference scheme for the second derivative, $f''(x)$, and apply it to calculate $\frac{d^2P}{dt^2}$ at $t = 22$. Save the result to the variable `A10`. To derive the scheme, follow the same steps as in the activity, but now we want to isolate the $f''(x)$ term and remove the $f(x)$ and $f'(x)$ terms.

2. The birth weight of newborn kittens is normally distributed with a mean of $\mu = 85$ grams (wow, so small!) and a standard deviation of $\sigma = 8.3$ grams. To compute the probability that a randomly selected newborn is a big one with weight between 110 grams and 130 grams, you would compute the integral

$$S = \int_{110}^{130} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)} dx$$

This integral cannot be evaluated exactly by using any of the methods you learned in Calculus class so we will evaluate it using numerical integration.

(a) Use `scipy.integrate.quad` function to calculate the "true" value of S. Save your answer to the variable `A11`.

(b) Use the left-sided rectangle rule to approximate S with step sizes of $h = 2^{-1}, 2^{-2}, \ldots, 2^{-16}$. Save the approximations in an array with 16 elements and save that to the variable `A12`.

(c) Repeat part (b) with the right-sided rectangle rule. Save the result to the variable `A13`.

(d) Repeat part (b) with the midpoint rule. Save the result to the variable `A14`.

(e) Repeat part (b) with the trapezoidal rule. Save the result to the variable `A15`.

(f) Repeat part (b) with Simpson's rule. Save the result to the variable `A16`.

Writeup problems

This section is worth 10 points, each problem is worth 5 points.

1. This problem mirrors Coding Problem 1. You should turn in the two figures created, answers to the questions, and your code.

   (a) Load in the data and plot the data as black markers with black lines in between (`'-ok'`). Add axis labels, legends, and a title.

   (b) Create a new figure (you can use `plt.figure()`). On it, plot the derivative, $\frac{dP}{dt}$, that you calculated using the second order methods, `A6`. Plot this using green markers with green lines between the markers (`'-og'`). Include axis labels, legends, and a title.

   (c) Comment on what you see in the plot of the derivative. You should see something that looks strange/confusing. Why is that? Why isn't it a smooth curve?

   (d) Based on what you see, why is it a good idea to use the *mean* of the calculated decay rate to calculate the half life.

2. This problem mirrors Coding Problem 2. Here we will create a plot with the errors for the five different methods used for calculating S. You are required to turn in the one figure created in parts (a)-(e), the comments on those plots in part (g), and your code. You will be graded on how easy it is to see and interpret your plot and how well it illustrates the orders of each method.

   (a) For each of the five methods (not including `scipy.integrate.quad`) used to calculate S, create an array of length 16 containing the absolute value of the error for the different step sizes. Do this by subtracting the "exact" solution obtained in Coding Problem 2 part (a) from the array containing the 16 approximations and then taking the absolute value.

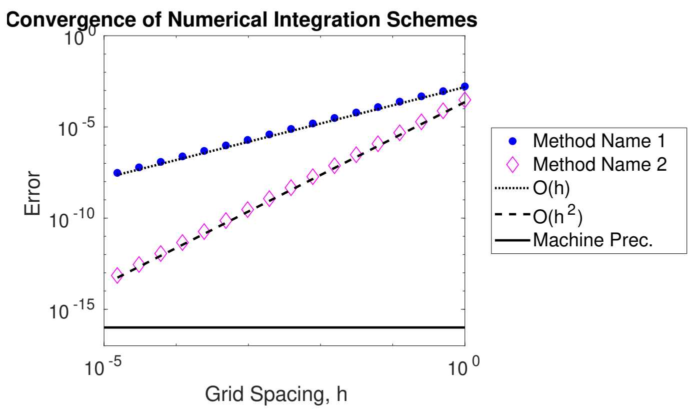
(b) Plot the errors versus the step size for each of the five methods using a log-log plot. Use a different color and marker type for each method.

(c) On the same figure, plot a trend line that represents $\mathcal{O}(h)$ by plotting $c \cdot h$ versus h on the log-log plot. Choose the constant c so that the trend line falls near your error points. Also include trend lines for $\mathcal{O}(h^2)$ and any other orders that are represented by the numerical integration methods in your plot. Plot these in a similar way: plot $C \cdot h$ versus h on the log-log plot and choose the constant C so that it matches your data. Use different line styles for each trend line (e.g. `':'`,`'-.'`,`'-'` etc.). Set the linewidth of each line to be at least 2 so that the lines can be easily seen.

(d) Add a black horizontal line at 10^{-16} which is (approximately) "machine precision". This is the lowest you could reasonably expect the error of one of the methods to be because of rounding error.

(e) Add appropriate labels to the x and y axes, a legend, and a title.

(f) Below is a sample of what the plot might look like for just two different integration schemes and two trend lines. (Note that this sample does not come from this problem, so your numbers will look different).



(g) Comment on what you see.

i. Which method has the highest order of accuracy? How can you tell from the plot?

ii. What is happening to the error for Simpson's rule with a very small step size? Why does the error stop decreasing here?