AMATH 301 – Autumn 2019
Homework 7

Due: 7:00pm, November 27, 2019.

Instructions for submitting:

- Scorelator problems are submitted as a MATLAB script (`.m` file). You should **NOT** upload your `.dat` files. The `.dat` files should be created by your script. You have 5 attempts on Scorelator.
- 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**. **All code you used for this part of the assignment should be included either at the end of the problem or at the end of your .pdf file.**

Scorelator problems

1. The file `SeaPopData.mat`, which is included with the homework, contains approximate yearly population data for the city of Seattle for the years $1890, 1891, \ldots, 2010$.

   You should load this data into MATLAB using the `load` command. Be sure that the file `SeaPopMat.mat` has been downloaded into the same directory as your script file. **You do NOT need to upload this file to Scorelator. Your code will be tested on a separate data file.** If the `load` command is successful, you will have two new vectors in your workspace, `t` and `P`. The vector `t` holds the years $1890, 1891, \ldots, 2009, 2010$. In other words, `t(0) = 1890` and `t(end) = 2010`. The vector `P` has the corresponding approximate populations.

   (a) Use the second-order left endpoint (forward) finite difference scheme derived in Activity 7 to approximate the derivative dP/dt for the year 1890. Save the result in `A1.dat`.

   (b) Use the second-order central finite difference scheme for the first derivative to approximate dP/dt for the years $t = 1891, 1892, \ldots, 2008, 2009$. Save the result as a row vector to `A2.dat`.

   (c) We now want to use a second-order right endpoint (backward) finite difference scheme similar to what was derived in Activity 7. Following what we did in class, derive a finite difference method which uses the points $f(x), f(x-h)$, and $f(x-2h)$ to find a second-order approximation for $f'(x)$. Use the method you derived to approximate dP/dt at $t = 2010$. Save your answer to `A3.dat`.

2. The birth weight of newborn babies is normally distributed with a mean of $\mu = 3.39$ kg and a standard deviation of $\sigma = 0.55$kg. To compute the probability that a randomly selected newborw has a weight between 2kg and 4kg, you would compute the integral

$$P = \int_2^4 \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)}dx$$

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 the `integral` function in MATLAB to calculate the "true" value of P. Save your answer to the file `A4.dat`.

(b) Use the left-sided rectangle rule to approximate P with step sizes of $h = 1, 2^{-1}, 2^{-2}, \ldots, 2^{-16}$. Save the approximations in a 17×1 column vector and save that to the file `A5.dat`.

(c) Use the right-sided rectangle rule to approximate P with step sizes of $h = 1, 2^{-1}, 2^{-2}, \ldots, 2^{-16}$. Save the approximations in a 17×1 column vector and save that to the file `A6.dat`.

(d) Use the midpoint rule to approximate P with step sizes of $h = 1, 2^{-1}, 2^{-2}, \ldots, 2^{-16}$. Save the approximations in a 17×1 column vector and save that to the file `A7.dat`.

(e) Use the trapezoidal rule to approximate P with step sizes of $h = 1, 2^{-1}, 2^{-2}, \ldots, 2^{-16}$. Save the approximations in a 17×1 column vector and save that to the file `A8.dat`.

(f) Use Simpson's rule to approximate P with step sizes of $h = 1, 2^{-1}, 2^{-2}, \ldots, 2^{-16}$. Save the approximations in a 17×1 column vector and save that to the file `A9.dat`.

3. The following is an example of a third-order accurate forward-difference method for the first derivative,

$$f'(x) = \frac{-\frac{11}{6}f(x) + 3f(x+h) - \frac{3}{2}f(x+2h) + \frac{1}{3}f(x+3h)}{h} + \mathcal{O}(h^3). \tag{1}$$

Assume that the error in calculating $f'(a)$ at the point $x = a$ with $h = 0.2$ is exactly 2×10^{-6}. Which of the following is most true? Save your answer to `A10.dat`.

A The error for calculating $f'(a)$ with $h = 0.02$ is **exactly** 2×10^{-3}.

B The error for calculating $f'(a)$ with $h = 0.02$ is **approximately** 2×10^{-3}.

C The error for calculating $f'(a)$ with $h = 0.02$ is **exactly** 8×10^{-6}.

D The error for calculating $f'(a)$ with $h = 0.02$ is **approximately** 8×10^{-6}.

E The error for calculating $f'(a)$ with $h = 0.02$ is **exactly** 2×10^{-7}.

F The error for calculating $f'(a)$ with $h = 0.02$ is **approximate** 2×10^{-7}.

G The error for calculating $f'(a)$ with $h = 0.02$ is **exactly** 2×10^{-9}.

H The error for calculating $f'(a)$ with $h = 0.02$ is **approximately** 2×10^{-9}.

Gradescope problems

1. This problem mirrors Scorelator Problem 2. Here we will create a plot with the errors for the five different methods used there for calculating P. You are required to turn in the one figure created in parts (a)-(d), the comments on those plots in part (f), and your code.

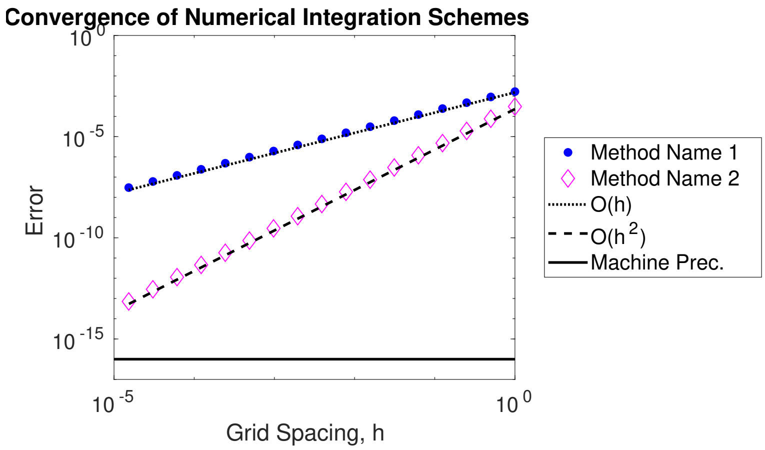
   (a) For each of the five methods used to calculate the P, create a 17×1 vector containing the absolute value of the error for the different step sizes. Do this by subtracting the "exact" solution obtained in Scorelator Problem 2 part (a) from the vector containing the 17 approximations and then taking 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.).

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

   (e) Below is a sample of what the plot might look like for just two different integration schemes and two trend lines.



   (f) Comment on what you see.

   - Which method has the highest order of accuracy?
   - What is happening to the error for Simpson's rule with a very small step size?