Assignment 3

4 November, 2022
Due date: 18 November

Instructions

- Submit to Avenue a **PDF file** containing your solutions.
  Do not submit files in other formats e.g. IMG, PNG.
- Name your MATLAB files **exactly** as specified.
- Name your PDF file **Lastname-Firstname-studentnumber.pdf**.
- Do not submit zipped files.

Problem 1 [10 points] [6 points] Derive formulas for the **smallest number** of n required to approximate

$$\int_0^{\pi/2} e^{2x}\sin(x)dx$$

with absolute error of at most `tol`, where `tol` is a given tolerance, using

a. Composite Trapezoidal rule

b. Composite Simpson's rule

c. Composite Midpoint rule

[4 points] Implement a MATLAB function `errors(tol)` that outputs for given tolerance the number of points and the corresponding errors. For example, my function outputs (digits are replaced by x)

```
tol=1e-07
trapezoid n=  xxxxx, error=xxe-08
midpoint  n=  xxxxx, error=xxe-08
Simpson   n=     xx, error=xxe-08
```

Submit

- Avenue: your MATLAB code. The function should be stored in a file with name `erorrs.m`. You can have other files called by this one.
- PDF: your derivations, `erorrs.m`, and the output with `tol=1e-4` and `tol=1e-12`.

Problem 2 [2 points] A particle of mass m moving through a fluid is subjected to a viscous resistance R, which is a function of the velocity v. The relationship between R, v, and time t is

$$t = \int_{v(t_0)}^{v(t)} \frac{m}{R(u)}du.$$

Assume $R(v) = -v\sqrt{v}$ for a particular fluid. If $m = 10$ kg and $v(0) = 10$ m/s, approximate the time for the particle to slow down to $v = 5$ m/s.

Problem 3 [4 points] Write a MATLAB program that does the following. Using composite Simpson, evaluate the integral

$$\int_{0.1}^{1} \sin\frac{1}{x}dx$$

for $n = 2, 4, 8, \cdots$ until the absolute error is within 10^{-4}. Apply the adaptive Simpson to the same integral, so the the error in the computed value is within 10^{-4}. How does the number of points compare? You program should output with the format specification

`'composite n=%d  error=%e  adaptive n=%d error=%e\n'`

Plot $\sin(1/x)$ versus x for $x \in [0.1, 1]$ and also mark on the x axis the points adaptive Simpson selects. What conclusion(s) can you make from this plot?

Submit

- Avenue: all MATLAB code. Name your main program `main_simpson.m`.
- PDF: the output of `main_simpson.m`, plot

Problem 4 [5 points] The MATLAB function `trapz` uses the composite trapezoid rule to evaluate an integral. Theoretically, the error in the trapezoid rule behaves like ch^2, for some constant $c > 0$. Here you want to check empirically that the power of h is indeed 2. Design a numerical experiment that finds c and k in ch^k for this function. Write a MATLAB function `[c,k]=error_trapz(f,a,b)` that take as input a function handle `f` and interval $[a, b]$ and outputs c and k.

Submit

- PDF: description of your algorithm, `error_trapz.m`, and the output of the scrript script `main_error_trapz.m`.
- Avenue: `error_trapz.m`

Problem 5 [8 points] Consider adding two $n \times n$ matrices using the functions below.

```
function C = addR(A, B)
    [n, ~] = size(A);
    for i = 1:n
        C(i, 1:n) = A(i, 1:n) + B(i, 1:n);
    end
end
function C = addC(A, B)
    [n, ~] = size(A);
    for j = 1:n
        C(1:n, j) = A(1:n, j) + B(1:n, j);
    end
end
```

(a) [4 points] Write a MATLAB program `timeadd.m` that times their execution for at least 10 distinct values of n. You can use the `timef` function in `timef.m`. Select values for n such that when plotted on a loglog scale, the data points (n, time) are nearly on a straight line. Theoretically, adding two $n \times n$ matrices takes $O(n^2)$ operations. Your program should find values for c and k in cn^k using least squares and output these values.

Denote the constants you compute for `addR` by c_{row} and k_{row}, and for `addC` by c_{col} and k_{col}.

Report the values for n you use and the computed four constants.

(b) [2 points] Report the computed constants when `n=1000:500:5000`.

Bonus In (b), my `timeadd` computes k_{row} close to 3, which does not make sense. That is an $O(n^2)$ algorithm runs in cubic time.

[2 points] If you are also getting k_{row} close to 3, can you conjecture what the reason might be (k_{row} should be ≈ 2).

[6 points] Justify theoretically and if possibly empirically your conjecture.

Submit

- PDF: (a) values of n and the computed constants, (b) computed constants; `timeadd.m`, answers to bonus part
- Avenue: `timeadd.m`

Problem 6 [4 points] The function $F(l) = k(l - l_0)$ is the force required to stretch a spring l units, where the constant $l_0 = 5.3$ is the length of the unstretched spring.

(a) Given the measurements

l	7	9.4	12.3
$F(l)$	2	4	6

determine k using least squares.

(b) Additional measurements are made giving the data

l	8.3	11.3	14.4	15.9
$F(l)$	3	5	8	10

Determine k from these data.

Which of the fits in (a) and (b) fits best the whole data

l	7	9.4	12.3	8.3	11.3	14.4	15.9
$F(l)$	2	4	6	3	5	8	10

Give sufficient detail supporting your claim.