CS/SE 4X03 — Assignment 3

3 November, 2021

Due date: 16 November

Instructions

- If you write your solutions by hand, please ensure your handwriting is legible. We may subtract marks for hard-to-read solutions.
- Submit to Avenue a **PDF file** containing your solutions and the **required** MATLAB **files**.

  Assignments in other formats, e.g. IMG, PNG, **will not be marked**.

  Name your MATLAB files **exactly** as specified.
- Name your PDF file **Lastname-Firstname-studentnumber.pdf**.
- Submit **only what is required**.
- Do not submit zipped files. We will **ignore any compressed file** containing your files.
- We will **deduct marks** for not following the required naming conventions.

Problem 1 [5 points] Consider the integral $\int_0^1 \sin(\pi x^2/3)dx$. Suppose that we wish to integrate it numerically with an error of magnitude less than 10^{-8} and using equally spaced points and the trapezoidal rule. Derive how many points are needed to achieve this accuracy.

Problem 2 [3 points] Approximate $\int_{-1}^{1}(x-0.5)^2\,\mathrm{d}x$ using the Simpson's rule with 5 equally spaced points and calculate the error in this approximation.

Problem 3 [3 points] A common problem in surveying is to determine the altitudes of a series of points with respect to some reference point. The measurements are subject to error, so more observations are taken than are necessary to determine the altitudes, and the resulting overdetermined system is solved in the least square sense to smooth out the error. Suppose that there are four points whose altitudes x_1, x_2, x_3, x_4 are to be determined. In addition to direct measurements of each x_i, with respect to a reference point, measurements are taken of each point with respect to all of the others. The resulting measurements are:

$$\begin{aligned} x_1 &= 2.95 & x_2 &= 1.74 \\ x_3 &= -1.45 & x_4 &= 1.32 \\ x_1 - x_2 &= 1.23 & x_1 - x_3 &= 4.45 \\ x_1 - x_4 &= 1.61 & x_2 - x_3 &= 3.21 \\ x_2 - x_4 &= 0.45 & x_3 - x_4 &= -2.75 \end{aligned}$$

From these data, find the best values for the altitudes. How do your computed values compare with the direct measurements?

Problem 4 [10 points] Implement the composite Simpson's rule and adaptive Simpson. Construct an example of a function $f(x)$, interval $[a, b]$ and tolerance `tol` such that, to achieve about the same accuracy, the composite Simpson's rule on a uniform mesh requires at least 100 times more function evaluations than your adaptive Simpson.

For this purpose consider the following steps.

- Use Matlab's `quad` function to compute an accurate approximation. For example, you can use tolerance `1e-2*tol` in `quad`.

  To measure the error produced by your implementations, you can subtract from the `quad`'s result and take absolute values.

- Select an n in your composite Simpson such that the error is about `tol`.
  Count the number of function evaluation for this n; denote them by C_1.

- In the adaptive Simpson, use `tol` and compare with the result from `quad`, to ensure that your computed result is within the tolerance.
  Count the the number of function evaluations; denote them by C_2.

  Then $C_1 \geq 100C_2$ should hold.

Name the program implementing the above `main_simpson.m`. It should output the errors of the composite and adaptive Simpson, C_1 and C_2, and should also plot $f(x)$ versus x.

Submit

- PDF: your function (in math notation), the plot, the errors of composite and adaptive Simpson, and C_1 and C_2
- Avenue: `main_simpson.m` and all the files it uses (if there are such)

Problem 5 [10 points] The error function is defined as

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

In MATLAB, it can be evaluated as `erf(x)`.

(a) Use the midpoint, trapezoid, and Simpson composite quadrature rules to compute an approximate value for erf(1).

Using `loglog`, plot on the same figure the error in each of these methods versus $h = 1/2^i$, where $i = 1, 2, \ldots, 10$.

(b) The errors in these rules behave like ch^k for constants c and k, where k is the order of a method. Using least squares, estimate these constants for each of the methods using the error data. Output the computed constants for each method as e.g.

```
fprintf('trapezoid %.2e*h^%.2f\n', c, k);
```

You must use this format specification.

Name the main program `main_integration_error.m` implementing the above. It should produce the plot and output the constants.

Submit

- PDF: plot, the computed constants
- Avenue: `main_integration_error.m` and any files it uses

Problem 6 [10 points] You are given the data file `nbody.dat` with positions of Jupiter, Saturn, Uranus, Neptune and Pluto.

- The first column is time. Each time unit is 100 days. This file contains data up to time 5000, which gives $5000 \times 100/365 \approx 1369, 86$ years.
- Columns 2, 3, 4 contain the coordinates (x, y, z) of Jupiter, then next three columns contain the coordinates of Saturn, and so on. Distance is measured from the sun in astronomical units (AU), where 1 AU is the mean radius of the earth's orbit.

A planet follows an elliptical orbit, which can be represented in a Cartesian (x, y) coordinate system by the equation

$$ay^2 + bxy + cx + dy + e = x^2. \tag{1}$$

Write a Matlab program that uses linear least squares to find for each of the planets the coefficients a, b, c, d, e in (1). Your program should output them in the from

	a	b	c	d	e
Jupiter					
Saturn					
Uranus					
Neptune					
Pluto					

Name your program `main_orbit.m`.

Note. If `x` and `y` are the x, y coordinates of a planet, and if you have computed a, b, c, d, e correctly, the plots from

```
plot(x,y);
hold on;
[xs, ys] = meshgrid(min(x)-1:0.1:max(x)+1, min(y)-1:0.1:max(y)+1);
contour(xs, ys, a*ys.^2+b*xs.*ys+c*xs+d*ys+e-xs.^2, [0, 0], 'k--', 'LineWidth', 1);
```

should overlap.

Submit

- PDF: the above table
- Avenue: `main_orbit.m`