

Assignment 3

MECHTRON 3X03: Scientific Computation

Due at 11:59 PM on Friday, November 24

Fall 2023

Submission Guidelines

There are 8 problems worth a total of 100 marks. Please read all of the files included in the Assignment 3 handout as they contain useful information. Submit the following files on Avenue:

1. A completed version of `assignment3_handout.jl` containing your programming solutions. Do **not** change the name of this file. Other than `LinearAlgebra`, do **not** include (via `import` or `using` or any other means) any libraries as part of your submission (they will be detected and removed by the grading script and you will receive a mark of zero). Do **not** change the name or function signatures of any of the functions you are asked to implement. You may write helper functions to simplify your solution (include these in your completed version of `assignment3_handout.jl`).
2. A PDF called `assignment3.pdf` containing your plots and solutions to derivation/discussion questions. This PDF can be any combination of typed/scanned/handwritten, so long as it is legible (you will receive a grade of zero on any section that cannot be understood).

The handout also contains a file called `assignment3_test.jl`. This contains some (but not all) of the code that we will be using to grade your submission. Do **not** hand this file in. You are advised to use this script to test your code and modify it while debugging.

The handout contains a few more Julia files with names that start with `assignment3_plot`. You will use these scripts (along with your completed functions in `assignment3_handout.jl`) to produce the plots you need for your PDF submission. Do **not** hand these scripts in. You may find it useful to modify this code while working on your solutions, but the plots you submit should be the result of running these scripts without making any modifications.

Use of Generative AI Policy

If you use it, treat generative AI as you would a search engine: you may use it to answer general queries about scientific computing, but any specific component of a solution or lines of code must be cited (see the syllabus for citation guidelines).

This is an individual assignment. All submitted work must be your own, or appropriately cited from scholarly references. Submitting all or part of someone else's solution is an academic offence.

Problems

Problem 1 (15 points): Implement the composite midpoint rule, the composite trapezoidal rule, and the composite Simpson's rule in their respective function templates in `assignment3_handout.jl`. Note that the composite Simpson's rule requires the input r to be an even number of subintervals, and that you should apply the basic Simpson's rule $r/2$ times (i.e., you can only evaluate the integrand f on $r + 1$ points). These functions will be tested by `assignment3_test.jl` with slightly different inputs. Do not change the function signatures.

Problem 2 (10 points): Run `assignment3_plot_composite.jl` with the functions you implemented in Problem 1. This script compares your numerical quadrature methods with the analytical solution to

$$I_f = \int_0^{4\pi} e^{-x/2} \sin(x) = 0.798506 \dots \quad (1)$$

by plotting the error against $h = (a - b)/r$ for varying values of r . If your composite integration rules have been implemented correctly, you should see two parallel lines and a third line that is roughly piecewise linear in two sections. Use what we learned in lecture about the expressions for the error of each composite quadrature rule to explain:

- a) (3 points) the slope of each line;
- b) (3 points) the offset between the parallel lines; and
- c) (4 points) the roughly piecewise continuous behaviour of the third line.

Note that the plot is displayed on logarithmic axes. Submit your answers to this question in your PDF submission.

Problem 3 (15 points): Implement the adaptive Simpson's rule in its function template in `assignment3_handout.jl`. This function will be tested by `assignment3_test.jl` with slightly different inputs. Do not change its function signature.

Problem 4 (10 points): Compare your composite and adaptive Simpson's rule implementations using the script in `assignment3_plot_adaptive.jl`, which computes the integral

$$T(k) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - k^2 \sin^2 \theta}}, \quad (2)$$

where $T(k)$ is proportional to the period of a simple pendulum with starting angle θ_0 and $k = \sin \frac{\theta_0}{2}$. Submit both plots that this script outputs in your PDF submission. The blue curve on the first plot, which displays $T(k)$ approximated with the adaptive Simpson's rule, should look smooth over the plotted range $-1 < k < 1$. The red curve plots the integrals computed by the composite rule using the same number of points as the adaptive rule needed to achieve approximately satisfy the tolerance provided.

- a) (4 points) Why do the integrals computed by the composite Simpson's rule differ from the adaptive solution? Which do you think is more accurate?
- b) (6 points) This script outputs a second plot which compares the "nodes" used by the adaptive and composite rules for $k = 0.99$. Describe the distribution of nodes for each method. How does the distribution of these nodes support your answer to part a)?

Problem 5 (10 points): Consider the problem of localizing a receiver using idealized (noiseless) range measurements to transmitters with known positions $p_i \in \mathbb{R}^n$. Unless we have at least n transmitters in a non-degenerate configuration, there will be infinitely many solutions. This problem is equivalent to solving the following system of linear equations:

$$\begin{aligned} f_1(x) &= \|x - p_1\| - d_1 = 0 \\ f_2(x) &= \|x - p_2\| - d_2 = 0 \\ &\vdots \\ f_n(x) &= \|x - p_n\| - d_n = 0, \end{aligned} \quad (3)$$

where $x \in \mathbb{R}^n$ is the unknown position of the receiver, and d_i is the noiseless measurement of the distance between x and p_i . If $F(x) = 0$ (where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$) is the system of equations in Eq. 3, find the Jacobian $J(x) = D_x F$ of $F(x)$. Include your derivation of the Jacobian as part of your PDF submission.

Problem 6 (10 points): Implement Newton's method for the range-only localization (or *triangulation*) problem described in Problem 5 in the appropriate function template in `assignment3_handout.jl`. This function will be tested by `assignment3_test.jl` with slightly different inputs. Do not change its function signature.

Problem 7 (20 points): Next, consider the realistic scenario where the distance measurements d_i are corrupted by noise and we have $m \geq n$ transmitting beacons. For independent zero-mean Gaussian distributions of the noise in d_i , we can estimate the position x by solving the nonlinear least-squares optimization problem

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^m |f_i(x)|^2 \quad (4)$$

with Newton's method, where the residuals f_i are the same as in Eq. 3. We will denote the objective function of Eq. 4

$$f(x) \triangleq \sum_{i=1}^m |f_i(x)|^2. \quad (5)$$

Find the gradient ∇f and Hessian $\nabla^2 f$ of f with respect to x . Include your derivations as part of your PDF submission. **Hint:** you may find tools like <https://www.matrixcalculus.org/> useful for checking your derivations.

Problem 8 (10 points): Implement Newton's method for the unconstrained optimization problem in Eq. 4 in the appropriate function template in `assignment3_handout.jl`. Run `assignment3_plot_newton.jl` and include the plot as part of your PDF submission. Provide the following information in your PDF submission:

- a) (5 points) numerical evidence that your implementation of Newton's method has converged to a critical point; and
- b) (5 points) numerical evidence that this critical point is a local minimum.