Prof. Dr. F. Knechtli
M.Sc. T. Asmussen
M.Sc. S. Awwad

Homework 1
due 10.11.25

Presentation Guidelines

- Solutions will be assessed through student presentations of the delivered codes and and handwritten parts. By handwritten it is meant either by literally handwritting and scanning it or writting on the computer and uploaded as pdf.
- Each presentation should be clear and concise (approximately 5-8 minutes).
- For theoretical parts, explain your derivation logically and clearly.
- For programming parts, show the key parts of your code, display the results (tables/plots), and provide a clear interpretation.
- Be prepared to answer questions from the instructor and other students.

Presentation 1: The Computer's Number System

Objective: Investigate the limits of numerical representation in MATLAB.

1.1 Machine Limits

Write a MATLAB program that determines:

(a) The largest representable number of the form 2^{n_1} (hint: `~isinf`).

(b) The smallest positive representable number of the form 2^{-n_2}.

Your program should display the values of n_1, n_2, 2^{n_1}, and 2^{-n_2}.

1.2 Machine Precision

definition of machine precision

Write a MATLAB program that determines the smallest numbers $e_2 = 2^{-n_1}$ and $e_{10} = 10^{-n_2}$ such that $1 + e_i \neq 1$ for $i = 2, 10$. After computing e_2 and e_{10}, compare their values to the predefined MATLAB quantity `eps`. Your program should display the results for n_1, n_2, 2^{-n_1}, 10^{-n_2}, and the comparison.

1.3 Numerical Cancellation

The area of a 3D object is given by

$$A = \frac{4}{3}\pi((r+h)^3 - r^3).$$

This formula becomes numerically inaccurate for $|h| \ll |r|$.

(a) Find a mathematically equivalent expression for A which is more suitable for numerical computation.

(b) Write a MATLAB program that displays a table with the results of both the original and your improved expression for $r = 1$ and $h = 10^{-1}, 10^{-2}, \ldots, 10^{-16}$. Include a column showing the relative difference between the two results.

Presentation 2: Numerical Differentiation and Error Analysis

Objective: Derive and implement high-order differentiation formulas and analyze their convergence.

2.1 High-Order Derivative Formula

(a) Find a formula for the numerical derivative of a function $f(x)$ with an error of order h^4, using values like $f(x \pm h)$ and $f(x \pm 2h)$. **Show your derivation by hand.** (Hints: Taylor expansion)

(b) Write a MATLAB program that prints a table comparing the errors of the forward difference (asymmetric), central difference (symmetric), and your new formula for $f(x) = \cos(x)$ at $x = 1$. Use step-sizes $h = 10^{-1}, 10^{-2}, \ldots, 10^{-10}$.

2.2 Visualizing and Fitting Error

(a) Extend your program to generate a log-log plot of the absolute value of the relative error, $\delta(h)$, versus the step-size h for all three methods. Comment on the behavior you observe (e.g., the range of optimal accuracy).

(b) Assuming a power-law relation $\delta(h) = ch^{\alpha}$, determine the exponent α and the constant c by hand for the three approximations, using the errors at $h = 10^{-1}$ and $h = 10^{-2}$. Do your empirical values for α agree with the theoretical expectations?

Presentation 3: Stability of Iterative Algorithms

Objective: Compare stable and unstable iterative algorithms for evaluating sequences and analyze convergence.

3.1 Stable Recurrence Relation

Consider the integrals

$$P_n = \int_1^e [ln(x)]^n dx.$$

(a) Derive by hand a forward iteration $P_{n+1} = f(P_n)$ and a backward iteration $P_n = g(P_{n+1})$.

(b) Determine which iteration is stable.

(c) Write a MATLAB program that uses the *stable* iteration to calculate the first 20 terms $P_1, P_2, \dots, P_{20}$ and prints them.

3.2 Convergence of a Fixed-Point Iteration

The mean-field equation for the Ising model is

$$x_* = \tanh(6\beta x_*).$$

It can be solved iteratively via

$$x_{n+1} \quad = \quad \tanh(6\beta x_n)\,, \quad x_0 > 0\,,$$

We analyze the case where β is slightly larger than $1/6$:

$$6\beta \quad = \quad 1 + \epsilon \quad \text{with} \quad \epsilon \ll 1$$

leading to a small non-trivial solution $x_* \neq 0$ becomes very "small"..

(a) By hand, expand the solution x_* to leading order in ϵ. Then, to analyze convergence, write $x_n = x_* + \delta_n$ and expand the iteration formula to leading order in the error δ_n. How many iterations N are needed to reduce the error by a factor of 10?

(b) Write a MATLAB program for $\beta = 0.17$ (where $x_* = 0.24062696159732$) that calculates the estimates x_n and plots the error δ_n versus n on a semilogarithmic scale.

(c) How many iterations are needed to reach machine accuracy? Compare this number with your analytical prediction from part (a) and discuss potential causes for any differences.