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

CSIS Computer Simulation in Science

Homework 2
due 01.12.25

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 4: Newton-Raphson Method and Convergence Analysis

Objective: Implement and analyze the Newton-Raphson method for different functions and understand its convergence properties.

4.1 Newton-Raphson for Square Roots

Given a real number $a \neq 0$, we want to compute $x = \frac{1}{a}$. Using $f(x) = a - \frac{1}{x}$:

(a) Use the Newton-Raphson method to derive an iteration formula for calculating $\frac{1}{a}$. **Show all your derivation steps.**

(b) Write a MATLAB script that implements this iteration and tests it for 10 different values of a between 1 and 10. For each value, print the number of iterations needed to achieve a relative error below 10^{-15}.

Hints:

- The Newton-Raphson formula is: $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$.
- Choose a smart initial guessfor the value of x_0 to ensure convergence.
- The relative error can be computed as $|x_{n+1} - x_n|/|x_{n+1}|$.
- make sure your loop stops when the error tolerance is met.

4.2 Convergence Analysis of Newton-Raphson

Consider $f(x) = x^n$ for fixed n. With initial guess $x_0 = 1$:

(a) Calculate the value of n such that after m Newton-Raphson iterations, the root estimate is 10^{-15}. **Show your calculation by hand.**

(b) Write a MATLAB script that implements Newton-Raphson for $f(x)$. For $m = 1000$ iterations with your calculated n, plot the root estimates on a semi-logarithmic scale.

(c) Discuss your observations. Is Newton-Raphson efficient for large n (e.g., $n = 100$)?

Hints:

- Analyze the Newton-Raphson update rule for $f(x) = x^n$.
- The iteration can be solved exactly – look for a pattern in $x_1, x_2, \ldots$.
- For large n, consider how many iterations are needed to reach machine precision.
- The semi-log plot will reveal exponential convergence (or lack thereof).

Presentation 5: Secant Method and Comparative Analysis

Objective: Implement the secant method and its variants, and compare convergence properties with other root-finding approaches.

5.1 Secant Method Implementation

(a) Write a MATLAB program implementing the standard secant method.

(b) Write a modified secant method that always maintains a bracket around the root (like bisection).

(c) Solve $\tanh(6\beta m) - m = 0$ for $\beta = 0.17$ using both variants.

Hints:

- Start from the `bisect.m` code available on Moodle.
- The standard secant method doesn't guarantee bracketing, but your modified version should.
- Use the exact solution from previous exercises and compare the convergence behavior of both variants: $m_* = 0.24062696159732$.

5.2 Convergence Rate Analysis

(a) Generate a log-log plot of δ_{n+1} vs. δ_n for both secant method variants.

(b) Compute the convergence exponents α for both methods where $|\delta_{n+1}| \propto |\delta_n|^\alpha$.

(c) Compare the convergence rates with Newton-Raphson and bisection methods.

Hints:

- For convergence analysis: if $|\delta_{n+1}| \propto |\delta_n|^\alpha$, then on a log-log plot, the slope equals α.

Presentation 6: Numerical Integration Methods and Error Analysis

Objective: Implement and analyze numerical integration algorithms, understanding their error behavior and practical limitations.

6.1 Theoretical Foundation of Simpson's Rule

Verify through **step by step calculation** that the leading error of Simpson's rule is $O(h^5)$. [Hint: Use Taylor expansion around the interval center $x = x_1$.]

6.2 Extended Integration Methods

Write a MATLAB program that integrates a function $f(x)$ over $[a, b]$ using:

- Extended trapezoidal rule
- Extended Simpson's rule

(a) Determine the maximum k for which each method integrates x^k exactly.

(b) Numerically solve:

$$\int_0^1 \frac{\ln(1+x)}{x} dx = \frac{\pi^2}{12}$$
$$\int_0^1 x\sqrt{1-x^2} dx = \frac{1}{3}$$

(c) Create log-log plots of relative precision vs. number of subintervals N for both methods.

Hints:

- Use `trapez.m` from Moodle as a starting point.

6.3 Adaptive Integration and Precision Analysis

(a) Write a MATLAB function that integrates $f(x)$ over $[a, b]$ using extended Simpson's rule with adaptive refinement until the relative difference between successive results is below tolerance δ.

(b) How many function calls do both algorithms need to obtain relative precision of 10^{-8} for the given integrals?

(c) Consider the series:

$$\sum_{n=1}^{\infty}(-1)^n \int_{n}^{n+1} \frac{e^{-x}}{x} dx$$

(d) Plot the absolute value of terms vs. n on a semi-logarithmic scale and find the largest N such that $\sum_{n=1}^{N} a_n + a_{N+1} \neq \sum_{n=1}^{N} a_n$ in MATLAB.