Problem 5 (Integration). In order to numerically approximate an integral

\[ I = \int_a^b f(x) \, dx, \]

we usually use the approach to chop up the interval into \( N \) small pieces with endpoints at \( a = x_0 < x_1 < x_2 < \cdots < x_N = b \), and use the formula

\[ I = \sum_{k=0}^{N-1} I_k, \quad I_k = \int_{x_k}^{x_{k+1}} f(x) \, dx. \]

We then try to find a simple way to approximate the individual pieces \( I_k \), for example by replacing the integral for \( I_k \) by the box rule, or the trapezoidal rule.

In the following, assume the distance between the points \( x_k \) is \( h \), i.e. for all intervals \( h = x_{k+1} - x_k \) with the same \( h \). Answer the following questions:

a) Consider the approximation formula

\[ \tilde{I}_k = hf(x_k) + \frac{h^2}{2} f'(x_k). \]  \hspace{1cm} (1)

Under the assumption that \( f \in C^3 \), prove that \( |I_k - \tilde{I}_k| = O(h^3) \).
b) If we approximate the full integral $I$ by $\tilde{I} = \sum_{k=0}^{N-1} \tilde{I}_k$, determine the convergence order of the error $I - \tilde{I}$.

c) Compare the scheme (1) with the trapezoidal rule

$$I_k^{\text{trapez}} = \frac{h}{2}f(x_k) + \frac{h}{2}f(x_{k+1}).$$

Which equation has the higher convergence order, and which may be more accurate? Which is more suitable in practice?

(4 points)

Problem 6 (Direct and iterative solvers). Assume you are given a square $N \times N$ matrix $A$ that

- is very large (i.e. $N$ is at least in the thousands),
- is positive definite,
- has a condition number $\kappa_2(A) = 10^{-4}N^2$,
- has only 17 nonzero entries per row, one of them on the diagonal.

You are supposed to solve a linear system $Ax = b$ for a given right hand side vector $b$. Explain which of the following methods is best suited for this task:

- LU decomposition,
- Jacobi iteration,
- the Conjugate Gradient (CG) method.

For a model matrix of size $N = 10^6$, determine

- the condition number of the matrix,
- the total number of entries of the matrix,
- for the LU decomposition the total number of operations,
- for the two iterative methods, the number of iterations you expect to need to reach an accuracy of $10^{-4}$,
- for the two iterative methods, the numerical effort (number of floating point additions, subtractions, multiplications and divisions) for each iteration, and for the entire solution process,
- for all three methods, how long the solution will take on a machine that can do $10^9$ floating point operations per second.
Hints: If you don’t exactly remember them, there will be three formulas you need to know: $-\log_{\epsilon} \kappa_2(A)$, $-\log_{\epsilon} \frac{\kappa_2}{\sqrt{\kappa_2(A)}}$, and $\frac{\kappa_1}{\epsilon}$. You should be able to figure out which formula goes where.

To compute the numerical effort requires you to count the number of operations in the algorithms given in the book on page 212 and 238. If you don’t immediately see how to do this, go on to the next problem.)

**Problem 7 (Convergence order of ODE solvers).** Assume for simplicity that we have an autonomous ODE

$$x'(t) = f(x(t)), \quad x(0) = x_0,$$

i.e. one where $f(\cdot)$ does not explicitly depend on the time $t$. In order to prove that an ODE solution scheme converges to the exact solution $x(t)$ as the step size $h$ goes to zero, one has to show two properties: stability and consistency. Let us only look at the consistency error and consider implicit multistep schemes using two time steps, in which $x_{k+1}$ is defined to satisfy:

$$\frac{x_{k+1} - x_k}{h} = \alpha_1 f(x_{k+1}) + \alpha_0 f(x_k).$$

If one inserts the exact solution $x(t_{k+1})$ instead of $x_{k+1}$ into this formula, and similarly $x(t_k)$ for $x_k$, the left and right hand sides will not be equal any more, but differ. This difference determines the convergence order.

**Task:** Compute this difference,

$$\frac{x(t_{k+1}) - x(t_k)}{h} - \frac{\alpha_1 f(x(t_{k+1})) + \alpha_0 f(x(t_k))}{h},$$

as a function of $h$, and determine the constants $\alpha_1, \alpha_0$ so that the difference is as small as possible. (2 points)

**Problem 8 (Norms; bonus problem).** A functional $\| \cdot \| : \mathbb{R}^n \to \mathbb{R}_0^+$ is a norm if it satisfies the following three conditions:

- $\|x\| = 0$ if and only if $x = 0$ (positivity);
- $\|\alpha x\| = |\alpha| \|x\|$ for all $\alpha \in \mathbb{R}$ and all vectors $x \in \mathbb{R}^n$ (linearity);
- $\|x + y\| \leq \|x\| + \|y\|$ for all vectors $x, y \in \mathbb{R}^n$ (triangle inequality).

Let $A$ be a symmetric and positive definite $n \times n$ matrix. Determine whether the functional

$$\|x\|_A = \sqrt{x^T A x}$$

is a norm or not. Does the situation change if $A$ is only positive definite, but not necessarily symmetric?

(Hint: The third condition above is a little tricky, don’t waste too much time on it if you can’t see how it is supposed to work.) (2 points)