Tecnical University of Denmark

Written 4-hour exam, 17. Decemeber 2024 Page 1 of 5

**Course name:** Introduction to Numerical Algorithms Course nr.: 02601

**Aids and materials:** All aids and materials permitted.

**Hand-in format:** All answers should be included in a single PDF file.

**Weights:** Multiple choice 40%, Question 1: 20%, Question 2: 15%, Question 3: 10%, Question 4: 15%. The weight is only a guideline. The final grade is based on overall assessment.

## Multiple choice (40%)

Each question has only ONE correct answer. In the pdf file with your answers, you just need to write the number of your answer.

**A – Errors.** What is the relative error involved in rounding 4.996 to 5.00?

1. $4.00 \times 10^{-3}$.
2. $4.00 \times 10^{-4}$.
3. $8.00 \times 10^{-4}$.
4. $8.01 \times 10^{-4}$.

**B – Interpolation: Lagrange form.** Let $x_1$, $x_2$, $x_3$ and $x_4$ be given interpolation nodes. For the interpolation polynomials in Lagrange form we use the cardinal polynomial, $l_i(x)$, such that $l_i(x_j) = 1$ if $i = j$ and zero otherwise. With the given nodes, what is the correct expression for the cardinal polynomial $l_2(x)$?

1. $l_2(x) = \frac{x-x_1}{x_2-x_1}$.
2. $l_2(x) = \frac{x-x_2}{(x_1-x_2)(x_3-x_2)(x_4-x_2)}$.
3. $l_2(x) = \frac{(x-x_1)(x-x_3)(x-x_4)}{(x_2-x_1)(x_2-x_3)(x_2-x_4)}$.
4. $l_2(x) = \frac{(x-x_1)(x-x_2)(x-x_3)(x-x_4)}{(x_2-x_1)(x_2-x_3)(x_2-x_4)}$.

**C – Secant method.** We apply secant method to find the roots of the equation $x^2 - x - 12 = 0$. If we use $x_0 = 0$ and $x_1 = 1$ as the starting points,

1. we will find the root 4.
2. we will find the root $-3$.
3. the iterate $x_k$ will tend to infinity.
4. the method will be broken down.

**D – Solving a linear system.** For a given $n$-by-$n$ matrix $A$, it takes about 0.36 seconds to solve the linear system of equations, $A\boldsymbol{x} = \boldsymbol{b}$, with $n = 200$ on a computer. In order to solve the linear system, the LU factorization followed by two triangular systems is used. Now, on the same computer in the same way we want to solve a linear system with $n = 800$. The approximate time, in seconds, that it will take to find the solution of $A\boldsymbol{x} = \boldsymbol{b}$ with $n = 800$ will be

1. 1.44.
2. 5.76.
3. 23.0.
4. 150.2.

**E – Sensitivity analysis.** Solving the linear system $A\boldsymbol{x} = \boldsymbol{b}$ with

$$A = \begin{bmatrix} 3.1 & -1.2 & 2.6 \\ 1.5 & -0.7 & 3.6 \\ -4.1 & 1.1 & 0.7 \end{bmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix} 0.654 \\ 0.765 \\ -1.042 \end{bmatrix},$$

we obtain the solution $\boldsymbol{x} = [1.2595, 3.5113, 0.3705]^T$. What is the upper bound of the relative error in the solution $\boldsymbol{x}$, if we instead use the approximate right-hand side $\bar{\boldsymbol{b}} = [0.657, 0.761, -1.039]^T$?

1. $2.828 \times 10^{-1}$.
2. $4.025 \times 10^{-3}$.
3. $6.295 \times 10^{-2}$.
4. $1.679 \times 10^{-2}$.

**F – Initial-value problems.** We want to solve an initial-value problem numerically using a large number of iterations with the step size $h$. We are considering to apply the Taylor series method of order 4 described on page 307 or the Runge-Kutta method of order 4 described on page 314 in the textbook. The truncation error after all the iterations, i.e., the global error, can be expected to be

1. of order $O(h^5)$ for both methods.
2. of order $O(h^5)$ for the Taylor series method and $O(h^4)$ for the Runge-Kutta method.
3. of order $O(h^4)$ for the Taylor series method and $O(h^5)$ for the Runge-Kutta method.
4. of order $O(h^4)$ for both methods.

## Question 1: Integration (20%)

Apply the composite Simpson's rule on $n$ equally spaced subintervals to approximate the integral

$$\int_0^1 e^{1-x^2}\,dx. \tag{1}$$

**1.1)** Use $n = 500$ in the composite Simpson's rule to compute the approximation of (1), and set this approximation as the ground-truth $\bar{S}$. State $\bar{S}$.
Then, set $n = 20, 40, 60, 80, 100$, respectively, in the composite Simpson's rule to compute the approximations of (1), and save as $S_{20}, S_{40}, \cdots, S_{100}$, respectively. Calculate the absolute errors, $e_n = |S_n - \bar{S}|$, and complete the following table with 3 significant decimal digits:

| $n$ | 20 | 40 | 60 | 80 | 100 |
|---|---|---|---|---|---|
| $e_n$ | ? | ? | ? | ? | ? |

For this question, you will need the Python function `MySimpson.py` or the Matlab function `MySimpson.m`, which are saved in the same folder as this exam assignment.

**1.2)** To study the behaviour of the errors, we can assume that

$$e_n \approx \beta n^{-\alpha}$$

with unknown $\alpha$ and $\beta$. By taking the logarithm on both sides, we obtain

$$\log(e_n) \approx \log(|\beta|) - \alpha \log(n) = b - \alpha \log(n).$$

Then, we are ready to find the constants $\alpha$ and $b$ by applying the least-squares fit with the data listed in the table from Question 1.1.

**1.2.1)** Set up the normal equation for the least-squares fit. State both the system matrix and the right-hand side in the normal equation.

**1.2.2)** Solve the normal equation, and state the solutions of $b$ and $\alpha$ with 3 significant digits. Include your code in the pdf file with your answers.

## Question 2: Newton's method (15%)

Consider a system of nonlinear equations:

$$\begin{aligned} -x^2 - x + 2y - 18 &= 0, \\ (x-1)^2 + (y-6)^2 - 25 &= 0. \end{aligned} \tag{2}$$

We will use Newton's method to find the solutions to the system (2).

**2.1)** Write the nonlinear system (2) in the form of $\boldsymbol{F}(\boldsymbol{X}) = \boldsymbol{0}$ with the variable $\boldsymbol{X} = [x, y]^T$. Derive the Jacobian matrix $\boldsymbol{F}'$ for the vector function $\boldsymbol{F}$, and write a Python or Matlab function to return a vector with the value of $\boldsymbol{F}(\boldsymbol{X})$ and a matrix with the values of the Jacobian matrix $\boldsymbol{F}'(\boldsymbol{X})$ for a given $\boldsymbol{X}$. Include your code in the pdf file with your answers.

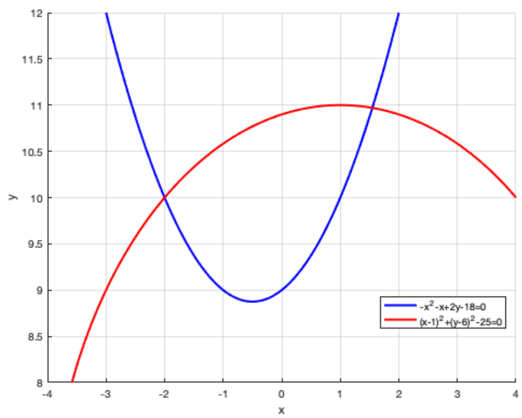


Figure 1: The function plots of (2).

**2.2)** Figure 1 shows the plots of both equations in (2). We can see that there are two solutions to the nonlinear system (2). According to Figure 1, what starting points would you choose in order to find both solutions by applying Newton's method?

**2.3)** Now, use the starting points that you suggested in Questions 2.2 to find the solutions to the nonlinear system (2). State your results with 4 significant digits. For this question, you will need the implementation of Newton's method that was used for Exercise 20 in Block 4.

## Question 3: Approximation of a function (10%)

In this question, we will approximate the function $f(x) = e^{2x}$ on the interval $I = [0, 0.1]$ and study the approximation error.

**3.1)** We know that $f$ has the values $f(0) = 1$, $f(0.05) = 1.105$ and $f(0.1) = 1.221$. Find the interpolation polynomial of the degree 2, $p_2$, in Newton form. In this question you do not need to use Python or Matlab. The derivations, intermediate results and explanations should be included in the pdf file with your answers.

**3.2)** Now, we approximate the function $f$ by a polynomial of degree at most $n$ with $n+1$ equally spaced nodes in the interval $I$. According to the second interpolation error theorem, find and state the smallest $n$ such that the approximation error $|f(x) - p_n(x)| \le 10^{-5}$ for all $x \in I$.

## Question 4: Initial-value problem (15%)

We consider the initial-value problem

$$\frac{d^3x}{dt^3} = -x \tag{3}$$

$$x(0) = 0, \quad \frac{dx}{dt}(0) = 1, \quad \frac{d^2x}{dt^2}(0) = 0$$

**4.1)** Rewrite the ordinary differential equation in (3) as a system of first-order differential equations, and implement a Python or Matlab function to return the right-hand side of the system. Include the code in the pdf file with your answers.

**4.2)** Apply the Runge-Kutta method of order 4 to solve this system of differential equation in the interval $[0, 4]$. Use $h = 2^{-k}$ with $k = 1$ and 2.

- State the number of steps $n$ and the number of function evaluations $n_f$ for $k = 1$ and 2, respectively.
- State the results of $x(4)$ at $t = 4$ for $k = 1$ and 2, respectively.

For this question, you will need the implementation of Runge-Kutta method of order 4 from Exercise 15.1 in Block 3.

END.