Final exam, Numerical Analysis (WISB251)
Tuesday, 11 April 2023, 17:00-20:00, BBG 023, 061

- Write your name on each page you turn in, and additionally, on the first page, write your student number and the total number of pages submitted.
- You may use one A4 sheet with notes while working the problems.
- For each question, motivate your answer. You may make use of results from previous subproblems, even if you have been unable to prove them.
- The maximum number of points per subproblem are given between square brackets. Your grade is the total earned points divided by 4. The final exam weighs 50% in your grade for the course.

Problem 1. [Nonlinear systems of algebraic equations]
Consider the following system of nonlinear equations for $x$ and $y$. Write $r = (x, y)^T$.

$$f(r) = \left( x + \frac{1}{2}y - \frac{\pi}{2} \right) = 0$$

Suppose we attempt to solve this system using the fixed point iteration

$$r_{k+1} = r_k - \alpha f(r_k).$$

(a) [2pts] Find (by hand) a solution $r^* = (x^*, y^*)^T$, $f(r^*) = 0$ of the nonlinear system.

(b) [4pts] What is the Jacobian matrix $f'(r^*) = Df(r^*)$ at $r^*$? What are its eigenvalues?

(c) [4pts] For what range of values $\alpha$ does the fixed-point iteration converge to $r^*$?

Problem 2. [Numerical integration]
We wish to approximate the definite integral

$$I = \int_{-1}^{1} f(x) \, dx,$$

using the values $f(c)$ and $f(-c)$ for $0 < c \leq 1$.

(a) [3pt] Construct the interpolating polynomial $p(x)$ through the points $(c, f(c))$ and $(-c, f(-c))$.

(b) [2pt] Show that the associated quadrature formula is given by

$$\bar{I} = \int_{-1}^{1} p(x) \, dx = f(c) + f(-c).$$

(c) [3pt] Derive an expression for the error $E = I - \bar{I}$ for the case $f(x)$ is a polynomial of degree $n$.

(d) [2pt] The formula is exact for polynomials up to a certain degree $n$. For what carefully chosen value of $c$ can you maximize this degree $n$?
Problem 3. [Numerical integration of ODEs]

Consider the following numerical method for solving an initial value problem \( y'(t) = f(y(t)), \ y(0) = y_0, \ y(t) \in \mathbb{R}^d, \ f : \mathbb{R}^d \to \mathbb{R}^d, \ t \in [0, T] : \)

\[
y_{n+1} = y_n + hf ((1 - \theta)y_n + \theta y_{n+1}),
\]

where \( y_n \approx y(t_n), \ n = 0, \ldots, N, \ t_n = nh, \ h = T/N. \)

(a) [5pts] Determine the truncation error for this method in the form

\[
\text{trunc. error} = Ch^q + O(h^{q+1})
\]

(i.e. determine \( q \) and \( C \)). What choice of \( \theta \) gives the best accuracy?

(Hint: Define \( \bar{y}(t) = (1 - \theta)y(t) + \theta y(t + h) \), and derive the Taylor expansion of \( \bar{y}(t) \) about \( y(t) \). Then determine the Taylor expansion of \( f(\bar{y}(t)) \) about \( y(t). \))

(b) [3pts] Compute the stability function \( R(z) \) such that \( y_{n+1} = R(h\lambda)y_n \) when the method is applied to the test problem \( y'(t) = \lambda y(t) \) for \( \lambda \) a complex number.

(c) [2pts] Sketch the stability regions \( S = \{ z \in \mathbb{C}; |R(z)| < 1 \} \) for \( \theta = 0, \ \theta = 1, \) and \( \theta = 1/2. \)

Problem 4. [Numerical differentiation formula]

In some applications it is necessary to evaluate a numerical difference formula at the midpoint between two nodes.

(a) [2pts] Write the Newton divided difference polynomial \( p_1(x) \) for the points \( (x_0, f(x_0)) \) and \( (x_1, f(x_1)) \) with \( x_1 = x_0 + h, \) and give an expression for the error \( e(x) = f(x) - p_1(x). \)

(b) [3pts] Show that the approximation of the derivative \( f'(x) = p_1'(x) \) is given by the difference formula

\[
f'(x) \approx \frac{f(x_0 + h) - f(x_0)}{h},
\]

(independent of \( x \)) and derive an upper bound on the error \( e'(x) = f'(x) - p_1'(x) \) for \( x \in [x_0, x_1]. \)

(c) [2pts] By directly expanding \( f(x_0 + h) \) in a Taylor series about \( x_0, \) derive an error bound for the above difference formula at \( x_0. \)

(d) [3pts] Instead, expand both \( f(x_0 + h) \) and \( f(x_0) \) in a Taylor series about the midpoint \( \hat{x} = x_0 + \frac{h}{2}, \) and derive an error bound for the difference formula at \( \hat{x}. \) How does the error of the approximation at the midpoint compare with your earlier bounds?