Norwegian University of Science and Technology
Department of Mathematical Sciences

MA2501

Deadline April 30, 2020

You must submit your work individually.

Practical information

- The project counts for 42,5% of the final mark.
- You will submit your work **individually**.
- You should produce a short report with your solutions in a pdf-file, preferably using LATEX.
- A part of the project will require the implementation (and testing) of codes in Python. A Jupiter notebook file with the code will also have to be handed in. Make sure that your code contains a meaningful documentation, in particular a reasonable number of comments in the code.
- When you present numerical results, these should be reproducible. This means that you will have to provide all relevant parameters.
- The report and the code should be submitted in Blackboard. The deadline is Thursday, April 30, 2020.
- There will be three weeks with no lectures devoted to the work and supervision of this project, starting form April 15 and until April 30, 2020.
- You are welcome to discuss with us the progress of your work. You can also contact us for help with the implementation work.

Problem 1 Nonlinear equations.

Reformulate the following equation into a fixed-point equation leading to convergent fixed-point iterations on some interval $[a, b] \subset \mathbb{R}$:

$$e^{-x} - \arccos(2x) = 0$$

Find a and b. Justify your answer.

Problem 2 Numerical linear algebra

a) Compute by hand the singular value decomposition of a 3×3 matrix A of your choice, $A = U\Sigma V^T$. Present all the steps of your procedure.

b) Make a simple code that verifies that the computed decomposition is correct and that U, V and Σ are indeed matrices satisfying the properties of the singular value decomposition. (You will have to hand in a Jupiter notebook code).

c) Let A be a real $n \times m$ matrix and $b \in \mathbb{R}^n$. Explain briefly what are the possible methods you have learned to solve the least square problem:

$$\text{Find}\, x^* \quad \text{s.t.} \quad \|Ax^* - b\|_2 = \min_{x \in \mathbb{R}^m} \|Ax - b\|_2.$$

Discuss the advantages and disadvantages of each choice of method.

Problem 3 Condition numbers.

Prove that if A is a real symmetric $n \times n$ matrix then the condition number $\mathcal{K}_2(A) := \|A\|_2 \|A^{-1}\|_2$ is

$$\mathcal{K}_2(A) = \frac{\max_{\lambda \in \sigma(A)} |\lambda|}{\min_{\lambda \in \sigma(A)} |\lambda|}$$

where we denote with $\sigma(A)$ the spectrum of A (i.e. the set whose elements are the eigenvalues of A).

Problem 4 Prove that for all A real invertible matrices $n \times n$ the condition number $\mathcal{K}_2(A) := \|A\|_2 \|A^{-1}\|_2$ is

$$\mathcal{K}_2(A) = \frac{\sigma_{\max}}{\sigma_{\min}},$$

where $\sigma_{\max}$ and $\sigma_{\min}$ are the largest and smallest singular value of A.

Problem 5

Suppose A is a $n \times n$ non-singular real matrix and let $\mathcal{K}_2(A) = \|A\|_2 \|A^{-1}\|_2$ be the condition number of A. Show that

$$\min \left\{ \frac{\|\delta A\|_2}{\|A\|_2} \mid \det(A + \delta A) = 0 \right\} = \frac{1}{\mathcal{K}_2(A)},$$

where δA is a real $n \times n$ matrix.

Hint:

- If you find this difficult, you can start by proving the statement for A and δA diagonal matrices with positive entries. Then generalise to all non-singular $n \times n$ symmetric matrices.

- The proof for non-singular $n \times n$ matrices A can be done by first proving that for all δA such that $\det(A + \delta A) = 0$ we have $\frac{\|\delta A\|_2}{\|A\|_2} \geq \frac{1}{\mathcal{K}_2(A)}$. Then proving that there is a particular δA for which the equality is attained.

Problem 6 **Divided differences**.

a) Interpolate the data

x	-2	-1	0	1
y	1	2	3	0

using divided differences and the Newton form of the approximation polynomial.

b) Using divided differences find the polynomial of lowest possible degree such that

$$p(-1) = 1/2, \quad p'(1/2) = 3, \quad p(1) = -1/2$$

Problem 7 **Divided differences**.

The formula

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

is well known. In this exercise we want to prove a more general formula, i.e. we want an expression for

$$S_k(n) = \sum_{i=1}^{n} i^k, \qquad k \geq 1$$

a) Prove first that $S_k(n)$ is a polynomial of degree $k+1$ in n.

Hint: Show that the divided difference $S_k[n, n+1, \ldots, n+m] = 0$ for all $n \geq 1$ when $m \geq k+2$.

b) Express $S_k(n)$ using the Newton interpolation polynomial and compute in particular $S_4(n)$. You can give your answer in Newton's form.

Problem 8 **Quadrature formulae**.

We want to approximate the following integral with Romberg's algorithm:

$$\int_0^1 f(\tau)\tau, \qquad f(\tau) = \exp(-\tau^2).$$

This integral is related to the so called error function erf

$$\operatorname{erf}(x)\sqrt{\pi}/2 = \int_0^x f(\tau)\tau.$$

This function is implemented in scipy and you may use it for verification and comparison ("from scipy.special import erf").

a) Explain briefly the idea of extrapolation and how it is related to Romberg's algorithm.

b) Find $R(3,2)$ the element with indexes $(3,2)$ of the Romberg matrix. Include the details of your calculation.

Problem 9 **Convergence of Runge-Kutta methods**. In this exercise we want to generalise the proof of convergence of the Euler method to be valid for any explicit Runge-Kutta method with s stages. Consider the initial value problem $\dot{y} = f(y)$, $y(0) = y_0$ on $[0,T]$ with $y(t) \in \mathbb{R}^m$ assume $f : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ is continuous in t and y and satisfies the Lipschitz condition with respect to y on $\mathbb{R}\times\mathbb{R}^m$ with Lipshitz constant L. Consider the one-step method

$$y_{n+1} = y_n + h\Psi_{f,h}(y_n), \quad h = \frac{T}{N}, \tag{1}$$

and N the total number of steps.

a) Assume $\Psi_{f,h}$ in (1) satisfies the Lipschitz condition with Lipschitz constant M on $\mathbb{R}\times\mathbb{R}^m$. Assume also that the one-step method (1) is consistent of order p. Prove that the method converges.

Recall that the method is consistent of order p if the local truncation error

$$\sigma(t_n, h) := y(t_n) - z_n, \qquad z_n := y(t_{n-1}) + h\Psi_{f,h}(y(t_{n-1})),$$

is $|\sigma(t,h)| = Ch^{p+1}$ for C a positive constant independent on h and $0 \le t \le T$.

Hint: Study the proof of the convergence of the Euler method (see recorded lecture and pdf file from the lecture) and adapt it to the present situation.

b) Assume now that (1) is an explicit Runge-Kutta method with 2 stages[1] and order p:

$$\begin{aligned} y_{n+1} &= y_n + h(b_1k_1 + b_2k_2), \\ k_1 &= f(t_n, y_n), \\ k_2 &= f(t_n + ch, y_n + haf(t_n, y_n)). \end{aligned}$$

Prove that the corresponding function $\Psi_{f,h}$ satisfies the Lipschitz condition and find the Lipschitz constant M. If you find this question too difficult prove the statement for Heun's method:

$$A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}, \qquad \mathbf{c} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

[1] Such method has Butcher tableau with parameters $(A, \mathbf{b}, \mathbf{c})$ where

$$A = \begin{bmatrix} 0 & 0 \\ a & 0 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \qquad \mathbf{c} = \begin{bmatrix} 0 \\ c \end{bmatrix},$$

and a, b_1, b_2 and c are real numbers.

Problem 10 Consider the Lotka-Volterra equations

$$\begin{aligned} \dot{y_1} &= \alpha y_1 - \beta y_1 y_2 \\ \dot{y_2} &= \delta y_1 y_2 - \gamma y_2 \end{aligned}$$

with $y_1(0) = y_{1,0}$, $y_2(0) = y_{2,0}$ on the time interval $[0, T]$. This is a predator prey system where $y_1(t)$ is the dynamics of increase/decrease of the prey population and $y_2(t)$ is the corresponding dynamics of the predator polupation. Here α is the growth parameter of the pray and δ the corresponding growth parameter for the predator; β is a parameter representing the frequency at which the predator and prey meet; γ is a parameter that controls the loss of predators due to death or emigration.

Implement the forward Euler and Heun's method to solve numerically this system of equations. Consider parameters α, β, δ, γ and initial values $y_1(0) = y_{1,0}$, $y_2(0) = y_{2,0}$ of your choice.

Use one of the NumPy routines for the numerical solution of ordinary differential equations to solve the Lotka-Volterra equations and to obtain a very accurate reference solution that can be used to test that your implementation of the Euler and Heun's methods is correct.

Provide numerical evidence that your implementation of the Euler method has indeed order 1 and for Heun's method order 2: compare the solution given by these numerical methods for different values of h and the reference solution obtained using the NumPy ODE routine. Choose $[0, 1]$ as time interval. Provide a `loglog` plot of the error versus the step-size h showing that you get lines of slope 1 and 2 respectively.

Problem 11 We consider the linearized pendulum equations

$$\theta''(t) + \omega^2 \theta(t) = 0, \quad 0 < t < 1, \quad \theta(0) = \alpha,\ \theta(1) = \beta, \tag{2}$$

valid for small oscillations[2]. We discretize with finite differences and obtain the numerical discretization on the grid $t_m = m\,h$, $m = 0, \ldots, M+1$ and $h = \frac{1}{M+1}$ leading to the discretized equations

$$\frac{1}{h^2}(\Theta_{m-1} - 2\Theta_m + \Theta_{m+1}) + \omega^2 \Theta_m = 0, \quad m = 1, 2, \ldots, M.$$

The linear system of equations written in a vector form is

$$G_h \Theta = \mathbf{b}, \quad \Theta = \begin{bmatrix} \Theta_1 \\ \vdots \\ \Theta_M \end{bmatrix},$$

where the matrix G_h is $G_h = A_h + \omega^2 I$, and A_h is the $M \times M$ matrix

$$A_h := \frac{1}{h^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}.$$

[2]Here $\omega^2 = \frac{g}{L}$, $g \approx 9.8$, and L is the length of the rod.

a) Find $\mathbf{b}$.

b) The truncation error vector is

$$\vec{\tau}_h := G_h \vec{\theta} - \mathbf{b}, \qquad \vec{\theta} := \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_M \end{bmatrix}, \quad \theta_j := \theta(t_j).$$

Show using Taylor theorem that the components of $\vec{\tau}_h$ satisfy

$$\tau_m = \frac{1}{12} h^2 \theta^{(4)}(t_m) + \mathcal{O}(h^4), \quad m = 1, \ldots, M.$$

This ensures that the method is consistent of order 2. What does this mean for the 2-norm of the vector $\vec{\tau}_h$? Interpret now the vector $\vec{\tau}_h$ as a piecewise constant function τ_h defined by

$$\tau_h(x) := \tau_m, \qquad x \in [t_m, t_{m+1}), \qquad m = 1, \ldots, M.$$

What can you conclude about the 2-norm (function norm) of this piecewise constant function?

c) Next we want to prove convergence. To simplify the analysis we assume $\omega^2 < \frac{\pi^2}{2}$. Find the equation relating the error $\vec{E}_h := \Theta - \vec{\theta}$ to the truncation error $\vec{\tau}_h$. Then derive an appropriate bound of the 2-norm of the error function by means of the 2-norm of the truncation error function and prove convergence.

Hint

- You may use the property that the eigenvalues of a matrix $\alpha I + B$ are $\alpha + \lambda(B)$ where, $\lambda(B)$ is an eigenvalue of B, (prove this statement if you use it).
- You may use the knowledge of the eigenvalues of the matrix A_h (see video-recorded lectures on boundary value problems and the note on boundary value problems).

d) Implement the method you have analysed in a Python code and confirm with a numerical experiment that the 2-norm of the error function goes to zero as $\mathcal{O}(h^2)$.