1. [10pt] Assume a linear least squares problem

$$\min_x \|Ax - b\|,$$  \hspace{1cm} (1)

where $A \in \mathbb{R}^{m \times n}$ has full rank, $m \geq n$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $\| \cdot \|$ is the Euclidean norm.

(a) Give the normal equations for this problem.

(b) Show how a QR-decomposition of $A$, i.e., $A = QR$ with an orthogonal matrix $Q \in \mathbb{R}^{m \times m}$, $R = [R_1, 0]^T \in \mathbb{R}^{m \times n}$ can be used to solve (1) efficiently. You can either use the normal equations or directly the optimization formulation (1).

The normal equation is $A^T Ax = A^T b$. Given a QR-factorization $A = QR$,

$$\|Ax - b\|^2 = \|Q^T (Ax - b)\|^2 = \left\| \begin{pmatrix} R_1 x - b_1 \\ -b_2 \end{pmatrix} \right\|^2 = \|R_1 x - b_1\|^2 + \|b_2\|^2,$$

where $Q^T b = [b_1, b_2]^T$. Hence, to minimize (1) we solve $R_1 x = b_1$ such that the first term in the above sum vanishes. This requires one triangular solve.

2. [15pt] Consider the quadratic optimization problems

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x - b^T x,$$  \hspace{1cm} (2)

where $Q \in \mathbb{R}^{n \times n}$ is symmetric, and $b \in \mathbb{R}^n$. For $n = 2$, give examples for $Q, b$ such that this optimization problem has

(a) a unique solution,

(b) an infinite number of solutions,

(c) no (finite) solution.

Assuming (2) has a unique solution, how many iterations does Newton’s method need to find that solution and why? Give examples of matrices $Q = Q_1$ and $Q = Q_2$, for which the steepest descent method converges fast and slow, respectively.

For $Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and any $b$, the minimization has a unique solution. For $Q = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ and $b = 0$, there are infinitely many solutions since $x_2$ is arbitrary. For $Q = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, and arbitrary $b$, the minimization problem is unbounded and thus has no finite solution. If there exists a unique solution, Newton’s method converges in 1 iteration since the problem is quadratic. The convergence of the steepest descent method depends on the ratio between
the largest and smallest eigenvalue. It converges quickly for instance for \( Q = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \), and slowly for instance for \( Q = \begin{pmatrix} 1000 & 0 \\ 0 & 1 \end{pmatrix} \) where the iterations will show a zig-zag behavior and convergence is slow.

3. [10pt] We consider the inner integral product weighted by \( \omega(t) = 1 \) on the interval \([-1, 1]\), i.e., \((p, q) := \int_{-1}^{1} p(t)q(t) dt\).

- Show that the polynomials \( 1/\sqrt{2}, t, 3/2t^2 - 1/2 \) are an orthogonal basis of the space of quadratic polynomials \( P_2 \).
- What are the optimal node locations \( t_0, t_1 \) for the quadrature rule
  \[ \hat{I}(f) = \omega_0 f(t_0) + \omega_1 f(t_1) \]
  on \([-1, 1]\) and why? Up to what polynomial order will this quadrature rule integrate polynomials exactly?

These polynomials are a basis of \( P_2 \) as they have different orders. Orthogonality requires to verify that
\[
\int_{-1}^{1} \frac{1}{\sqrt{2}} t dx = 0, \quad \int_{-1}^{1} \frac{1}{\sqrt{2}} \left( \frac{3}{2} t^2 - \frac{1}{2} \right) dx = 0, \quad \int_{-1}^{1} t \left( \frac{3}{2} t^2 - \frac{1}{2} \right) dx = 0.
\]
The optimal nodes for the quadrature are the Gauss nodes, which are the roots of the polynomial \( p_{n+1}(t) = 3/2t^2 - 1/2 \), which are \( t = \pm \frac{1}{\sqrt{3}} \). If the weights are chosen appropriately (namely \( \omega_0 = \omega_1 = 1 \)), integration is exact for polynomials up to degree \( 2n + 1 = 3 \).

4. [10pt] The divided difference scheme computes the coefficients of an interpolating polynomial with respect to the Newton basis. Here is an example for the interpolating polynomial that satisfies \( p(0) = 1, p(0.5) = 2, p(1) = 0, p(2) = 3 \).

\[
\begin{array}{c|c}
 t_i & [t_i]f = 1 \\
 0 & t_0 \end{array}
\]
\[
\begin{array}{c|c}
 0.5 & [t_1]f = 2 \\
 1 & [t_2]f = 0 \\
 2 & [t_3]f = 3 \\
 t_0t_1 \end{array}
\]
\[
\begin{array}{c|cc}
 & t_0t_1 & t_1t_2 \\
 0 & [t_0t_1]f = [t_1]f - [t_0]f = 2 \\
 1 & [t_1t_2]f = [t_2]f - [t_1]f = -4 \\
 2 & [t_2t_3]f = [t_3]f - [t_2]f = 3 \\
 t_1t_2 & t_0t_2t_3 \\
 3 & [t_0t_2t_3]f = \frac{14}{3} \\
 6 & [t_1t_2t_3]f = \frac{16}{3} \\
 \end{array}
\]

(a) Give the interpolating polynomial.

(b) Give a MATLAB (or PYTHON) code listing that takes as inputs vectors \( t, p \in \mathbb{R}^{n+1} \) and outputs the \( n+1 \) coefficients of the polynomial interpolant \( p(t) \in P_n \) for the Newton basis. Your code should work for any \( n \) and you can assume that the nodes \( t_0 < t_1 < \ldots < t_n \) are distinct. Stay as close to MATLAB/PYTHON syntax as possible.

The interpolating polynomial is
\[
1 + 2t - 6t(t - 0.5) + \frac{16}{3}t(t - 0.5)(t - 1).
\]
5. **[10pt]** Consider \( f : \mathbb{R}^n \to \mathbb{R} \) continuously differentiable, and \( x \in \mathbb{R}^n \) with \( \nabla f(x) \neq 0 \). For a symmetric positive definite matrix \( A \in \mathbb{R}^{n \times n} \), we define the \( A \)-weighted norm \( \| y \|_A = \sqrt{y^T A y} \). Derive the unit norm steepest descent direction of \( f \) in \( x \) with respect to the \( \| \cdot \|_A \)-norm, i.e., find the solution to the problem

\[
\min_{\| d \|_A = 1} \nabla f(x)^T d. \tag{3}
\]

We use the Choleski factorization of \( A \), i.e., \( A = B^T B \) and introduce \( e = Bd \). Then, (3) becomes

\[
\min_{\| e \| = 1} \nabla f(x)^T B^{-1} e = \min_{\| e \| = 1} (B^{-T} \nabla f(x))^T e.
\]

This is now of the form of a regular steepest descent problem, for which we know that the steepest descent direction is

\[
e = \frac{-B^{-T} \nabla f(x)}{\| B^{-T} \nabla f(x) \|},
\]

and thus

\[
d = B^{-1} e = \frac{-A^{-1} \nabla f(x)}{\| A^{-1} \nabla f(x) \|_A}.
\]

6. **[10 pt]** You are given data points \((t_i, f_i)\) for \( i = 1, \ldots, 10 \) and the following model with parameters \( x = (x_1, x_2, x_3) \) to fit this data:

\[
\varphi(t; x) = \exp(x_1) + \exp(x_2)t + x_3 t^2
\]

Formulate the corresponding nonlinear least squares problem to find the best-fitting parameters \( x \), and give the Gauss-Newton algorithm step at the initialization \( x = (0, 1, 2)^T \) explicitly. Upon convergence of your algorithm, you plot the fitting model and the data and obtain the plot shown in Figure 1. What kind of convergence speed of your Gauss-Newton algorithm did you (most likely) observe, and why?

![Figure 1: Data (dots) and best fitting model (solid line).](image-url)
The nonlinear least squares problem is

$$\min_{x \in \mathbb{R}^3} \|F(x)\|^2,$$

where

$$F(x) = \begin{pmatrix}
\exp(x_1) + \exp(x_2)t_1 + x_3^2t_1^2 - f_1 \\
\vdots \\
\exp(x_1) + \exp(x_2)t_{10} + x_3^2t_{10}^2 - f_{10}
\end{pmatrix} \in \mathbb{R}^{10}.$$

The Gauss-Newton iteration step is

$$F'(x^k)^T F'(x^k) \Delta x = -F'(x^k)^T F(x^k), \quad x^{k+1} = x^k + \Delta x, \quad (4)$$

where the Jacobian is

$$F'(x) = \begin{pmatrix}
\exp(x_1) & \exp(x_2) & 2x_3t_1^2 \\
\vdots & \vdots & \vdots \\
\exp(x_1) & \exp(x_2)t_{10} & 2x_3t_{10}^2
\end{pmatrix} \in \mathbb{R}^{10 \times 3}.$$

For the point $x = (0, 1, 2)$, the Gauss-Newton step is (4) with

$$F((0, 1, 2)^T) = \begin{pmatrix}
1 + \exp(1)t_1 + 4t_1^2 - f_1 \\
\vdots \\
1 + \exp(1)t_{10} + 4t_{10}^2 - f_{10}
\end{pmatrix}, \quad F'((0, 1, 2)^T) = \begin{pmatrix}
1 & \exp(1)t_1 & 4t_1^2 \\
\vdots & \vdots & \vdots \\
1 & \exp(1)t_{10} & 4t_{10}^2
\end{pmatrix}$$

The convergence is only linear, unless the residual at the solution $x^*$ is zero, i.e., $F(x^*) = 0$. This is not the case in the figure since the model curve does not fit all data points exactly.