Università della Svizzera italiana
Institute of Computing CI

Optimization Methods **2024**

Student: Jeferson Morales Mariciano

Discussed with: Leonardo Birindelli

Midterm

Due date: Sunday, 12 May 2024, 11:59 PM

Exercise 1

Let $f : \mathbb{R}^n \to \mathbb{R}$ be given by $f = \frac{1}{2}x^T Ax - b^T x$ with A symmetric positive definite. Let x_m be the minimizer of the function f. Let v be an eigenvector of A, and let λ be the associated eigenvalue. Suppose that we use Stepeest Descent (SD) method to minimize f and the starting point for the SD algorithm is $x_0 = x_m + v$.

1.

Prove that the gradient at x_0 is $\nabla f(x_0) = \lambda v$.

Proof. Recall that for eigenvalue λ and eigenvector v of matrix A, we have the following property $Av = \lambda v$ (1). Moreover, for minimizer x_m of quadratic function f holds $\nabla f(x_m) = Ax_m - b = 0$ (2). Then:

$$
\begin{aligned}
\nabla f(x) &= Ax - b \\
\nabla f(x_0) &= A(x_m + v) - b = Ax_m + Av - b = Ax_m - b + Av \\
&\overset{(2)}{=} Av \\
&\overset{(1)}{=} \lambda v
\end{aligned}
$$

□

2.

How many iterations does the SD method take to minimize the function f if we use the optimal step length? Show the computations behind your reasoning.

It will converge in a single iteration for quadratic function f, with $x_0 = x_m + v$ where x_m is the function minimizer and v being an eigenvector with associated eigenvalue λ.

Proof. Recalling the definition of optimal step length α_{opt} for quadratic functions:

$$
\alpha_{opt} = \left(\frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T A \nabla f_k} \right)
$$

and the Steepest Descent iteration defintion of subsequent points $\{x_k\}$:

$$x_{k+1} = x_k - \alpha_k \nabla f_k$$

We compute x_1:

$$\begin{aligned} x_1 &= x_0 - \alpha_{opt} \nabla f_0 = x_m + v - \frac{\nabla f_0^T \nabla f_0}{\nabla f_0^T A \nabla f_0} \nabla f_0 \\ &\overset{1.1}{=} x_m + v - \frac{(\lambda v)^T (\lambda v)}{(\lambda v)^T A (\lambda v)} \lambda v = x_m + v - \frac{\lambda^2 v^T v}{\lambda^2 v^T A v} \lambda v = x_m + v - \frac{\|v\|^2}{v^T A v} \lambda v \\ &\overset{(1)}{=} x_m + v - \frac{\|v\|^2}{v^T \lambda v} \lambda v = x_m + v - \frac{\|v\|^2}{\lambda \|v\|^2} \lambda v = x_m + v - \frac{1}{\lambda} \lambda v = x_m + v - v \\ &= x_m \end{aligned}$$

Where x_m is the minimizer of the function f, thus the Steepest Descent method converges in the first single iteration exploiting the eigenvector v, hence moving along its direction, and using the optimal step length α_{opt}. □

Exercise 2

Given a starting point $x_0 \in \mathbb{R}^n$ and a set of conjugate directions $\{p_0, p_1, \ldots, p_{n-1}\}$, we generate the sequence $\{x_k\}$ by setting

$$x_{k+1} = x_k + \alpha_k p_k \tag{1}$$

where

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k} \tag{2}$$

and r_k is the residual, as defined in class. Consider the following theorem:

Theorem 1. *For any $x_0 \in \mathbb{R}^n$ the sequence $\{x_k\}$ generated by the conjugate gradient direction algorithm (1) and (2), converges to the solution x^* of the linear system $Ax = b$ in at most n steps.*

Prove the theorem and explain carefully every step of your reasoning.

Proof. Since the directions $\{p_i\}$ are linearly independent, they must span the whole space $\mathbb{R}^n$. Hence, we can write the difference between the starting point x_0 and the solution x^* in the following way:

$$x^* - x_0 = \sum_{k=0}^{n-1} \sigma_k p_k = \sigma_0 p_0 + \sigma_1 p_1 + \ldots + \sigma_{n-1} p_{n-1}$$

for some choice of scalars σ_k. By premultiplying this expression by $p_k^T A$ and using the conjugacy property:

$$pi^T A p_j = 0 \qquad \forall i \neq j \tag{3}$$

with computations $\forall k = 0, 1, \ldots, n-1$:

$$\begin{aligned} x^* - x_0 = \sigma_k p_k &\Rightarrow p_k^T A (x^* - x_0) = p_k^T A \sigma_k p_k \\ &\Rightarrow p_k^T A (x^* - x_0) = \sigma_k p_k^T A p_k \\ &\Rightarrow \frac{p_k^T A (x^* - x_0)}{p_k^T A p_k} = \sigma_k \end{aligned}$$

we obtain:

$$\sigma_k = \frac{p_k^T A \left(x^* - x_0\right)}{p_k^T A p_k} \tag{4}$$

We now establish the result by showing that these coefficients σ_k coincide with the step lenghts α_k generated by Formula 2.

If x_k is generated by Algorithm 1, 2, then we have:

$$x_k = x_0 + \sum_{i=0}^{k-1} \alpha_i p_i = x_0 + \alpha_0 p_0 + \alpha_1 p_1 + \ldots + \alpha_{k-1} p_{k-1}$$

By premultiplying this expression by $p_k^T A$ and using the conjugacy property in Formula 3, we have that:

$$p_k^T A \underbrace{\left(x_k - x_0\right)}_{\sum_{i=0}^{k-1} \alpha_i p_i} = 0$$

and by defining the residual r_k as:

$$r_k = A x_k - b \tag{5}$$

and therefore:

$$p_k^T A \underbrace{\left(x^* - x_0\right)}_{\sum_{i=0}^{n-1} \alpha_i p_i} = p_k^T A \underbrace{\left(x^* - x_k\right)}_{\sum_{i=k}^{n-1} \alpha_i p_i} = p_k^T \left(b - A x_k\right) \overset{(5)}{=} -p_k^T r_k \tag{6}$$

By comparing this relation with the α_k and σ_k in Formula 2, 4:

$$\sigma_k \overset{(4)}{=} \frac{\overbrace{p_k^T A \left(x^* - x_0\right)}}{p_k^T A p_k} \overset{(6)}{=} \frac{-p_k^T r_k}{p_k^T A p_k} = \frac{\overbrace{-r_k^T p_k}}{p_k^T A p_k} \overset{(2)}{=} \alpha_k$$

we conclude that $\sigma_k = \alpha_k$, giving the result to prove the theorem of convergence in at most n steps, with $A \in \mathbb{R}^{n \times n}$ being SPD.

$$x^* - x_0 = \sum_{k=0}^{n-1} \sigma_k p_k = \sum_{k=0}^{n-1} \alpha_k p_k$$

$$x^* = x_0 + \sum_{k=0}^{n-1} \alpha_k p_k$$

□

Exercise 3

Consider the linear system $Ax = b$, where the matrix A is a symmetric positive definitive diagonal matrix constructed in three different ways:

$A = diag([1:10])$

$A = diag(ones(1, 10))$

$A = diag([1, 1, 1, 3, 4, 5, 5, 5, 10, 10])$

$A = diag([1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0])$

1.

How many distinct eigenvalues has each matrix?

The following matrices has the following number of distinct eigenvalues $|\Lambda|$, which correspond to the number of different elements in the diagonal of the matrix A:

$A1 = diag([1:10]) \rightarrow 10$

$A2 = diag(ones(1,10)) \rightarrow 1$

$A3 = diag([1,1,1,3,4,5,5,5,10,10]) \rightarrow 5$

$A4 = diag([1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,2.0]) \rightarrow 10$

2.

Implement the CG method (CG.m).

Matlab scripts are provided in */code* folder. Inside you will find the *CG.m* implementation.

3.

Construct a right-hand side $b = rand(10,1)$ and apply the Conjugate Gradient method to solve the system for each A.

The main file to run is *ex3.m*, which uses the *CG.m* implementation. It handles the computation of the Conjugate Gradient to the previously defined matrices.

4.

Compute the logarithm energy norm of the error (i.e. $log((x - x^*)^T A(x - x^*)))$ for each matrix and plot it with respect to the number of iteration.

The same main file *ex3.m* handles the visualization of the logarithm energy norm of the error.

Figure 1: error of A1

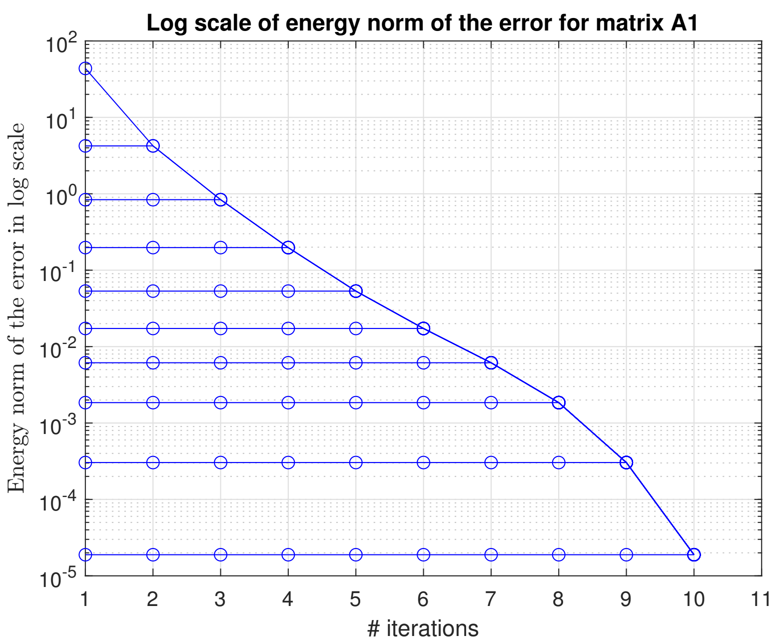


Figure 2: error of A2, (convergence in 1 iteration in log scale)

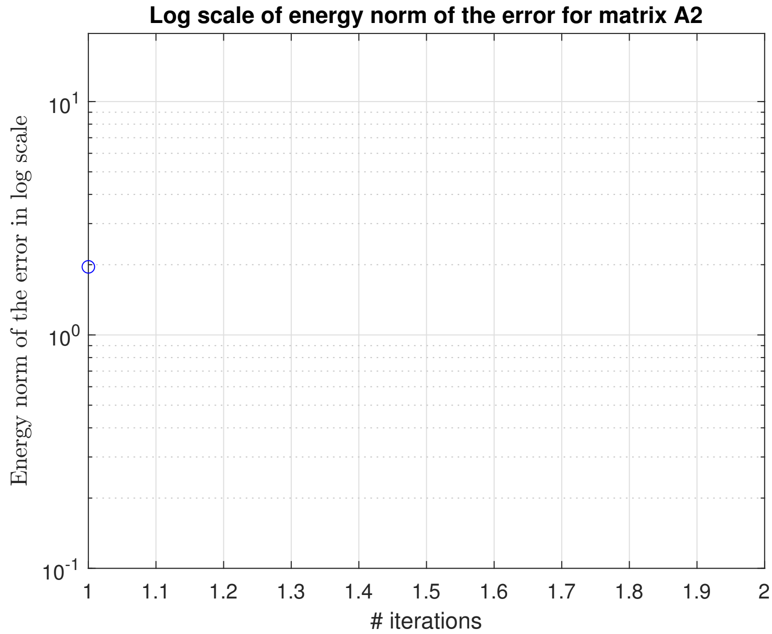


Figure 3: error of A3

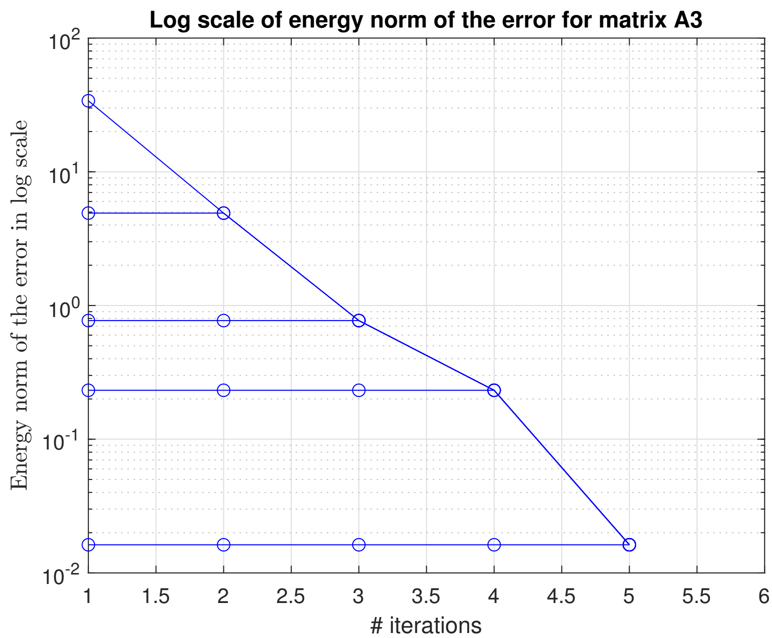
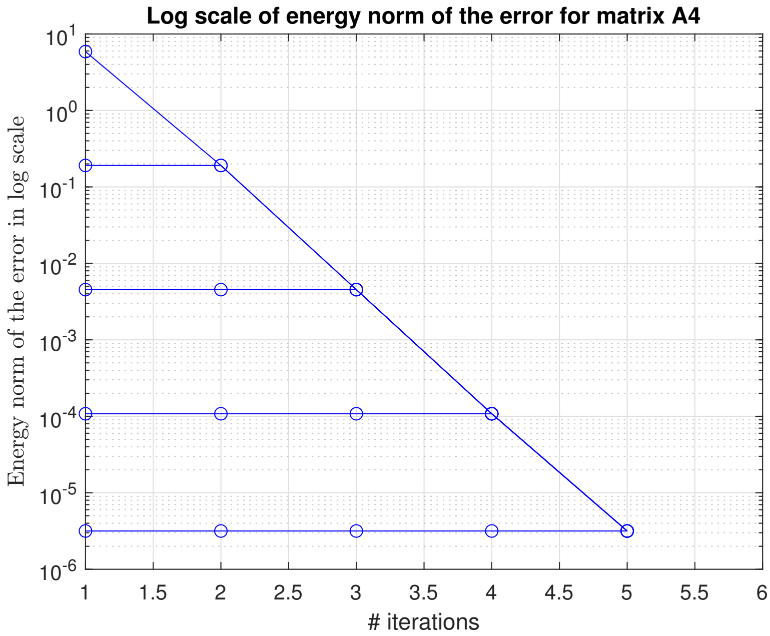


Figure 4: error of A4



5.

Comment on the convergence of the method for the different matrices. What can you say observing the number of iterations obtained and the number of clusters of the eigenvalues of the related matrix?

The convergence of the method is directly related to the number of distinct eigenvalues and their distribution in the matrix A. Uniform distribution of the eigenvalues in the matrix A leads to a balanced convergence while a skewed distribution leads to a potentially slower or higher convergence, dependent on the starting point and how many clusters of eigenvalues are there.

The convergence of the matrices are related to the distribution and the number of their eigenvalues. All convergence in less than n iterations as expected by the conjugate gradient method.

Exercise 4

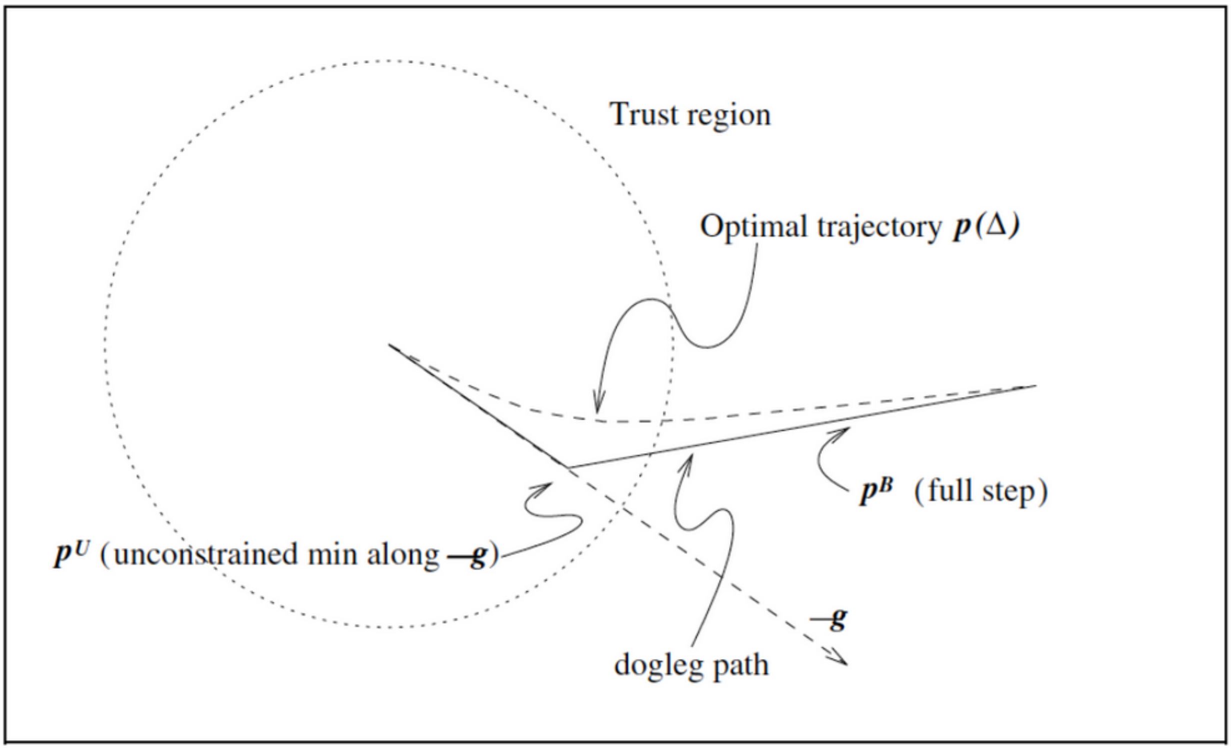
Consider the Chapter 4, "Trust-region methods" of the book *Numerical Optimization*, Nocedal and Wright.

1.

Explain Cauchy point method and Dogleg method, as well as the connection between them.

Starting with the Cauchy Point method, from the analogy of the globally convergence of linear search methods even when the optimal step length was not choosen, we have a similar situation in Trust Region methods. The Cauche Point described in Algorithm **??** is inexpensive to calculate because no matrix factorization is required although is of crucial important to understand if an approximate solution of the trus region subproblem p_k is acceptable. Specifically, the Cauchy Point method will be globally convergent if its steps p_k give a reduction in the model m_m that is at least some fixed positive multiple of the decrease attained by the Cauchy step.

Figure 5: Dogleg schema



Regarding the Dogleg method, Figure 5 shows how more sophisticated approach is to the Trust Region problem. The dogleg method finds an approximate solution by rplacing the curved trajectory for $p(\Delta)$ with a path consisting of two line segments as described in Algorithm 3. The first line segment runs from the origin to the minimizer of m along the steepest descent direction p^U. While the second line segment runs from p^U to p^B. Such trajectory is described with as $\tilde{p}(\tau)$ in Equation , where τ is a parameter that varies from 0 to 2 and choosen based on Algorithm 4.

The connection between the Cauchy Point and Dogleg methods is that the Dogleg method is a more sophisticated approach to the Trust Region problem, where the Cauchy Point method is a simpler and less expensive method to calculate the approximate solution of the Trust Region subproblem.

2.

Write down the Trust-Region algorithm, along with Dogleg and Cauchy-point computations.

Algorithm 1 Trust Region

$\hat{\Delta} > 0$: a set of authentic users
$\Delta_0 \in (0, \hat{\Delta})$: a set of authentic users
$\eta \in [0, \frac{1}{4})$: a set of authentic users
for $k = 0, 1, 2, \ldots$ **do**
    Obtain approximately of p_k using either Cauchy Point or Dogleg method
    Evaluate ρ_k from the ration between actual and predicted reduction
    if $\rho_k < \frac{1}{4}$ **then**
        $\Delta_{k+1} = \frac{1}{4}\Delta_k$
    else
        if $\rho_k > \frac{3}{4}$ **and** $\|p_k\| = \Delta_k$ **then**
            $\Delta_{k+1} = \min(2\Delta_k, \hat{\Delta})$
        else
            $\Delta_{k+1} = \Delta_k$
        end if
    end if
    if $\rho_k > \eta$ **then**
        $x_{k+1} = x_k + p_k$
    else
        $x_{k+1} = x_k$
    end if
end for

Algorithm 2 Cauchy Point method

h: hessian of the function
g: gradient of the function
Δ: radius
$p_k^S = -\frac{\Delta}{\|g\|} g$
$\tau = \begin{cases} 1 & \text{if } g^T hg \leq 0 \\ \min\left(\|g\|^3/(\Delta g^T hg), 1\right) \end{cases}$
$p_k^C = -\tau \frac{\Delta}{\|g\|} g$

Figure 3 shows the Dogleg method algorithm, where the function *chooseTau* is described in Algorithm 4. The Newton step is choosen for p^B.

Algorithm 3 Dogleg method

h: hessian of the function
g: gradient of the function
Δ: radius
$p^B = -h^{-1}g$
if $\left\|p^B\right\| \leq \Delta$ **then**
    $p_k^D = p^B$
    return
end if
$p^U = -\frac{g^T g}{g^T h g} g$
if $\left\|p^U\right\| \geq \Delta$ **then**
    $p_k^D = \frac{\Delta p^U}{\|p^U\|}$
    return
end if
$p_k^D = p^U +$ chooseTau$(p^U, p^B, \Delta) \cdot (p^B - p^U)$

Algorithm 4 chooseTau

Solve $\left\|p^U + \tau(p^N - p^U)\right\|^2 = \Delta^2$ ▷ Choose Tau such that $\|p\|^2 = \Delta^2$
$\tau = \max(\tau_1, \tau 2)$

3.

Consider the following lemma (Lemma 4.2, page 75 Numerical optimization, Nocedal and Wright)

Lemma 4.2. *Let B be positive definite. Then,*

(a) $||\tilde{p}(\tau)||$ is an increasing function of τ, and

(b) $m(\tilde{p}(\tau))$ is a decreasing function of τ

Read carefully the proof and explain in detail how each step is obtained.

Recalling:

$$\tilde{p}(\tau) = \begin{cases} \tau p^U & 0 \le \tau \le 1 \\ p^U + (\tau - 1)\left(p^B - p^U\right) & 1 \le \tau \le 2 \end{cases}$$

First, let's prove that $\|\tilde{p}(\tau)\|$ is an increasing function with. $\tau \in [0,1]$.

Proof. Let's define $\phi(\tau) = \|\tilde{p}(\tau)\|$. By computing $\frac{\partial \phi}{\partial \tau}$, since $\left\|p^U\right\|$ is positive, the derivative is always positive and the function is monotonically increasing.

$$\begin{aligned} \phi(\tau) &= \|\tilde{p}(\tau)\| = |\tau| \cdot \left\|p^U\right\| = \tau\left\|p^U\right\| \\ \frac{\partial \phi}{\partial \tau} &= \left\|p^U\right\| \ge 0 \end{aligned}$$

thus, $\phi(\tau)$ is an increasing function with $\tau \in [0,1]$. □

Second, let's prove that $m(\tilde{p}(\tau))$ is a decreasing function with $\tau \in [0,1]$.

Proof. Let's expand the function $m(\tilde{p}(\tau))$. After computing its $\frac{\partial m}{\partial \tau}$, we expand p^U by its definition and simplify the expression. Remember B is SPD. At the end, since $\tau \in [0,1]$, the expression $(\tau - 1) \le 0$, thus the function is monotonically decreasing.

$$\begin{aligned} m(\tilde{p}(\tau)) &= f + g^T \tilde{p}(\tau) + \frac{1}{2}\tilde{p}(\tau)^T B \tilde{p}(\tau) = f + \tau g^T p^U + \frac{1}{2}\tau^2 (p^U)^T B p^U \\ \frac{\partial m}{\partial \tau} &= g^T p^U + \tau (p^U)^T B p^U \\ &= g^T \cdot \left(-\frac{g^T g}{g^T B g} g\right) + \tau \left(-\frac{g^T g}{g^T B g} g\right)^T B \left(-\frac{g^T g}{g^T B g} g\right) \\ &= g^T \cdot \left(-\frac{\|g\|^2}{g^T B g} g\right) + \tau \left(-\frac{\|g\|^2}{g^T B g} g\right)^T B \left(-\frac{\|g\|^2}{g^T B g} g\right) \\ &= -\frac{\|g\|^4}{g^T B g} + \tau \frac{\|g\|^4}{(g^T B g)^2} g^T B g = -\frac{\|g\|^4}{g^T B g} + \tau \frac{\|g\|^4}{g^T B g} \\ &= \underbrace{(\tau - 1)}_{\le 0} \underbrace{\frac{\|g\|^4}{g^T B g}}_{\ge 0} \le 0 \end{aligned}$$

thus, $m(\tilde{p}(\tau))$ is a decreasing function with $\tau \in [0,1]$. □

Third, let's prove that $\|\tilde{p}(\tau)\|$ is an increasing function with $\tau \in [1,2]$.

Proof. Let's define $\alpha = \tau - 1$, and $\phi(\alpha) = \frac{1}{2}\|\tilde{p}(\alpha)\|^2$ with $\alpha \in [0,1]$. Inside $\phi(\alpha)$ we defined the squared norm $\|\cdot\|^2$ to delete the square root from the calculations. Then, simplify the expression by collecting using the square of sum which is the reason why we have the scalar $\frac{1}{2}$ in $\phi(\alpha)$. Remember B is SPD. Afterwards, we compute $\frac{\partial \phi}{\partial \alpha}$ and then prove $\left(p^U\right)^T\left(p^B - p^U\right) \ge 0$. To do so, we use the Cauchy-Schwarz inequality and exploit the B matrix SPD properties, raising it to the square root to easy the computation of the norm of u, v variables. Finally, this concludes the proof to show that ϕ is monotonically increasing.

$$
\begin{aligned}
\phi(\alpha) &= \frac{1}{2}\|\tilde{p}(\alpha)\|^2 = \frac{1}{2}\left\|p^U + \alpha\left(p^B - p^U\right)\right\|^2 \\
&= \frac{1}{2}\left\|p^U\right\|^2 + \alpha\left(p^U\right)^T\left(p^B - p^U\right) + \frac{1}{2}\alpha^2\left\|p^B - p^U\right\|^2 \\
\frac{\partial\phi}{\partial\alpha} &= \left(p^U\right)^T\left(p^B - p^U\right) + \underbrace{\alpha\left\|p^B - p^U\right\|^2}_{\geq 0} \geq \left(p^U\right)^T\left(p^B - p^U\right)
\end{aligned}
$$

$$
\begin{aligned}
\left(p^U\right)^T\left(p^B - p^U\right) &= -\left(p^U\right)^T\left(p^U - p^B\right) \\
&= \frac{g^Tg}{g^TBg}g^T \cdot \left[\left(-\frac{g^Tg}{g^TBg}g\right) - \left(-B^{-1}g\right)\right] \\
&= \frac{\|g\|^2}{g^TBg}g^T \cdot \left(B^{-1}g - \frac{\|g\|^2}{g^TBg}g\right) \\
&= \frac{\|g\|^2}{g^TBg}g^TB^{-1}g \cdot \left(1 - \frac{\|g\|^2}{g^TBgB^{-1}g}g\right) \\
&= \underbrace{\frac{\|g\|^2}{g^TBg}g^TB^{-1}g}_{\geq 0} \cdot \left(1 - \frac{\|g\|^4}{\left(g^TBg\right)\left(g^TB^{-1}g\right)}\right)
\end{aligned}
$$

$$
\begin{aligned}
\text{let } u = B^{-\frac{1}{2}}g, v = B^{\frac{1}{2}}g &\Rightarrow \left|u^Tv\right|^2 \leq \|u\|^2 \cdot \|v\|^2 \\
&\Rightarrow \left|g^TB^{-\frac{1}{2}}B^{\frac{1}{2}}g\right|^2 \leq \left\|B^{-\frac{1}{2}}g\right\|^2 \cdot \left\|B^{\frac{1}{2}}g\right\|^2 \\
&\Rightarrow \left|g^Tg\right|^2 \leq \left(g^TB^{-\frac{1}{2}}B^{-\frac{1}{2}}g\right)\left(g^TB^{\frac{1}{2}}B^{\frac{1}{2}}g\right) \\
&\Rightarrow \|g\|^4 \leq \left(g^TB^{-1}g\right)\left(g^TBg\right) \\
&\Rightarrow \frac{\|g\|^4}{\left(g^TBg\right)\left(g^TB^{-1}g\right)} \leq 1 \\
&\Rightarrow 1 - \frac{\|g\|^4}{\left(g^TBg\right)\left(g^TB^{-1}g\right)} \geq 0
\end{aligned}
$$

thus, $\phi(\tau)$ is an increasing function with $\tau \in [1, 2]$. □

Lastly, let's prove that $m(\tilde{p}(\tau))$ is a decreasing function with $\tau \in [1, 2]$.

Proof. Let's define again $\alpha = \tau - 1$, for $m(\tilde{p}(\alpha))$ with $\alpha \in [0, 1]$.

$$
\begin{aligned}
\phi(\alpha) &= m(\tilde{p}(\alpha)) = f + g^T\tilde{p}(\alpha) + \frac{1}{2}\tilde{p}(\alpha)^T B\tilde{p}(\alpha) \\
&= f + g^T\left[p^U + \alpha\left(p^B - p^U\right)\right] + \frac{1}{2}\left[p^U + \alpha\left(p^B - p^U\right)\right]^T B\left[p^U + \alpha\left(p^B - p^U\right)\right] \\
&= f + g^T p^U + \alpha g^T\left(p^B - p^U\right) + \frac{1}{2}\left(p^U + \alpha p^B - \alpha p^U\right) B\left(p^U + \alpha p^B - \alpha p^U\right) \\
&= f + g^T p^U + \alpha g^T\left(p^B - p^U\right) + \frac{1}{2}[(p^U)^T Bp^U + \alpha(p^U)^T Bp^B - \alpha(p^U)^T Bp^U + \alpha(p^B)^T Bp^U \\
&+ \alpha^2(p^B)^T Bp^B - \alpha^2(p^B)^T Bp^U - \alpha(p^U)^T Bp^U - \alpha^2(p^U)^T p^B + \alpha^2(p^U)^T Bp^U] \\
&= f + g^T p^U + \alpha g^T\left(p^B - p^U\right) + \frac{1}{2}(\alpha^2\left[(p^B)^T Bp^B - (p^B)^T Bp^U - (p^U)^T Bp^B + (p^U)^T Bp^U\right] \\
&+ \alpha\left[(p^U)^T Bp^B - (p^U)^T Bp^U + (p^B)^T Bp^U - (p^U)^T Bp^U\right] + (p^U)^T Bp^U) \\
&= f + g^T p^U + \alpha g^T\left(p^B - p^U\right) + \frac{\alpha^2}{2}\left[(p^B)^T Bp^B + (p^U)^T Bp^U - 2(p^B)^T Bp^U\right] \\
&+ \alpha\left[(p^U)^T Bp^B - (p^U)^T Bp^U\right] + (p^U)^T Bp^U \\
\frac{\partial\phi}{\partial\alpha} &= g^T\left(p^B - p^U\right) + \alpha\left[(p^B)^T Bp^B + (p^U)^T Bp^U - 2(p^B)^T Bp^U\right] + \left[(p^U)^T Bp^B - (p^U)^T Bp^U\right] \\
&= g^T p^B - g^T p^U + \left[(p^B)^T Bp^U - (p^U)^T Bp^U\right] + \alpha\left[(p^B)^T Bp^B - (p^U)^T Bp^B + (p^U)^T Bp^U - (p^B)^T Bp^U\right] \\
&= (p^B)^T g - (p^U)^T g + (p^B)^T Bp^U - (p^U)^T Bp^U + \alpha\left[(p^B)^T Bp^B - (p^U)^T Bp^B + (p^U)^T Bp^U - (p^B)^T Bp^U\right] \\
&= \left(p^B - p^U\right)^T\left(g + Bp^U\right) + \alpha\left(p^B - p^U\right) B\left(p^B - p^U\right) \\
\frac{\partial\phi}{\partial\alpha} &\leq \left(p^B - p^U\right)^T\left(g + Bp^U + B\left(p^B - p^U\right)\right) \\
&= \left(p^B - p^U\right)^T\left(g + Bp^B\right) \\
&= \left(p^B - p^U\right)^T\left[g + B\left(-B^{-1}g\right)\right] = \left(p^B - p^U\right)^T\cdot(g - g) = 0
\end{aligned}
$$

thus, $m(\tilde{p}(\tau))$ is a decreasing function with $\tau \in [1, 2]$. □