Université de Limoges
Faculté des Sciences et Techniques
Département de Mathématiques

October 2023

MASTER DE MATHÉMATIQUES ET APPLICATIONS, PARCOURS ACSYON

SPLITTING METHODS FOR CONVEX OPTIMIZATION

Homework 1 .

Cleque Marlain MBOULOU

Exercice 1 :

For this exercise we want to solve numerically by using **POCS Algorithm** the following equation:

$$Ax = b, \ \ \texttt{whith} \ x \in \cap_{i=1}^{n} C_i$$

where $(C_i)_{i=1,2,\ldots,n}$ are be a finite family of nonempty closed convex sets.

$$\begin{aligned} Ax = b \Leftrightarrow a_i^T x = b_i \ \texttt{for} \ i = 1,2,\ldots,n \ \texttt{and} \ a_i \ \texttt{the row vectors of A} \\ \Rightarrow x \in [< a_i, . >= b_i] \ \texttt{for} \ i = 1,2,\ldots,n \\ \Rightarrow x \in \cap_{i=1}^{n} [< a_i, . >= b_i] \end{aligned}$$

Affine hyperplanes are closed convex sets, can be taken $C_i := [< a_i, . >= b_i] \ \texttt{for} \ i = 1,2,\ldots,n$. And for $b \in \mathcal{I}m(A), \cap_{i=1}^{n} C_i \neq \emptyset$
Using POCS Algorithm to solve this equation.

Algorithm 1: Pseudo-code de la fonction 'pocs'

Result: Solution vector x
Input: Matrix A, vector b

$n, m \leftarrow size(A)$;
$x \leftarrow zeros(m, 1)$;
$T \leftarrow 1000$;
$err \leftarrow 1e-6$;
disp(x);
for $j \leftarrow 1$ **to** T **do**
  for $i \leftarrow 1$ **to** n **do**
    $Ci \leftarrow reshape(A(i,:), [], 1)$;
    $x \leftarrow x - (dot(Ci, x) - b(i))/(dot(Ci, Ci)) * Ci$;
  end
  if $norm(A * x - b) < err$ **then**
    break;
  end
end

Bonus: We can solve this equation by the solution of normal equation given by:

$$x = (A^T A)^{-1} A^T b \tag{1}$$

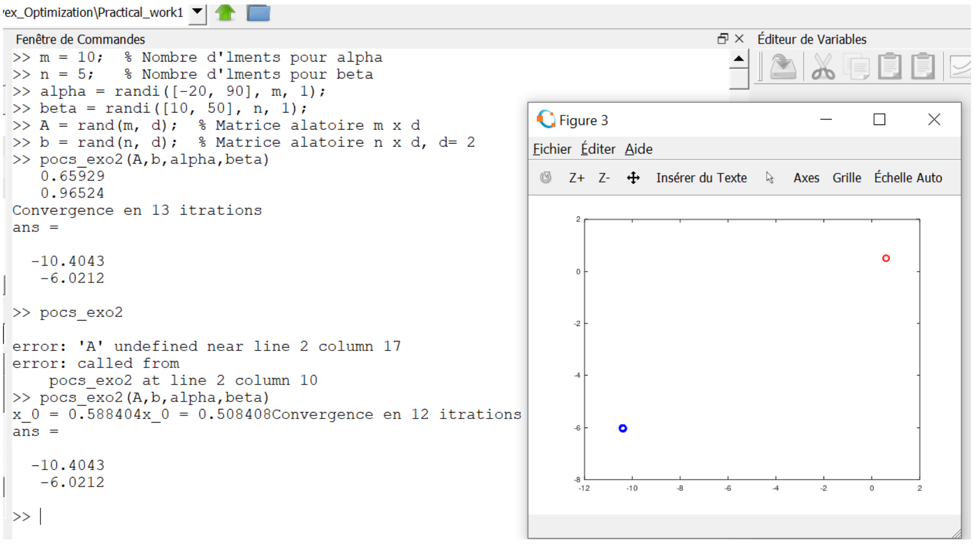
Exercice 2 :
The aim of this exercise is to provide a code based on POCS to satisfy simulatory the followings inequalities:

$$\texttt{for} \ i = 1,2\ldots,m, \sum_{k=1}^{d} a_{ik} x_a \leq \alpha_i \tag{1}$$

$$\texttt{for} \ j = 1,2,\ldots,n, \sum_{k=1}^{d} (x_k - b_{jk})^2 \leq \beta_j \Leftrightarrow ||x - b_j||_2^2 \leq \beta_j \tag{2}$$

Where the a_{ik}, b_{jk}, α_i, and β_j are given real numbers (chosen so that there exists at least one solution). We notice that (1) represents the equations of affine half-spaces and (2) the equations of closed Balls (Euclidean norm) of center b_j and radius β_j, which are closed convex sets. Moreover, there exists at least one solution, then the intersection of this closed convex sets is non empty.
Thus, we can use POCS Algorithm to solve this inequalities where the closed convex sets are affine half-spaces and closed Balls.

BONUS .



Exercice 3 :

1. To compute the projection of a vector $\mathbf{y}$ (y^*) onto the column space (image) of a matrix $\mathbf{A}$ with linearly independent columns, we want to find $x \in \mathbb{R}^d$ such that $y^* = Ax$. **(**)**. We know that:
   $< y - y^*, Ax^{'} >= 0, \forall x^{'} \in \mathcal{Im}(A) \Rightarrow < A^T y - A^T y^*, x^{'} >= 0$
   However, $x^{'} \neq 0$, then $A^T y - A^T y^* = 0 \Leftrightarrow A^T y = A^T y^* \Leftrightarrow A^T y = A^T A x \Leftrightarrow x = (A^T A)^{-1} A^T y$
   Thus by replacing in **(**)**, $y^* = A(A^T A)^{-1} A^T y$

$$\text{proj}_{\text{Im}(\mathbf{A})}(\mathbf{y}) = A(A^T A)^{-1} A^T y \tag{2}$$

2.

3. (a) Diversity of Portraits: Since the matrix A represents portraits of different mathematicians, it's highly likely that these portraits are distinct and diverse. Each mathematician would have a unique appearance, and their portraits should capture those differences. As a result, the pixel values in each column of the matrix A are expected to be significantly different from each other.

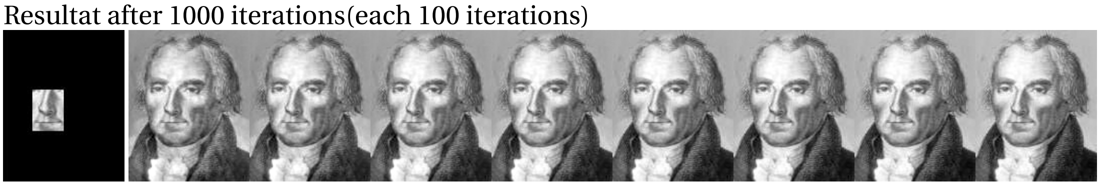
   (b) $C_1 = \mathcal{Im}(A)$

   - **Nonempty:** C_1 is the image of matrix A, which is the set of linear combinations of the columns of A. Since A has linearly independent columns, C_1 is not empty.
   - **Closed:** C_1 is a closed set because it is the range (image) of a linear transformation, which is known as a closed set.
   - **Convex:** C_1 is convex because any linear combination of vectors in C_1 remains in C_1 by defintion of C_1, particulary convex combination.
   - **Projection** ($proj_{c_1}$)**:** To numerically compute the projection of a vector x onto C_1 we need to find a vector x in C_1 that minimizes the Euclidean distance $||x - z||_2$. This is the solution of question (1): $proj_{c_1}(z) = A(A^T A)^{-1} A^T z$

   C_2 **(Set of Vectors Matching a Region of** y**):**

- **Nonempty:** C_2 is not empty if there exists at least one vector x whose components in the specified pixel range match those of y.
- **Closed:** Assuming y exists, C_2 can be considered closed as it represents a finite set of vectors that match y in specific components.
- **Convex:** C_2 is also convex. For two vectors x_1 and x_2 in C_2 , a convex combination $\lambda x_1 + (1 - \lambda)x_2$ will still have components that match y in the specified pixel range, making it a convex set.
- **Projection** ($proj_{c_2}$)**:** To numerically compute the projection of a vector z onto C_2 , we can substitute in z components corresponding to the square from pixel $60 * 30$ to pixel $100 * 60$ of y (`z(60:100, 30:60) = y(60:100, 30:60)`)

(c) Resultat after 1000 iterations(each 100 iterations)



We can conclude that the target mathematician is **math31**

(d) x^* belongs to both C_1 because it's an image formed by the linear combinations of the columns of A, and C_2 because it matches the specified components of y (`(x*(60:100, 30:60) = y(60:100, 30:60)`). Therefore, x^* is at the intersection of C_1 and C_2.