

**UNIVERSITE SULTAN MOULAY SLIMANE**

**ECOLE NATIONALE DES SCIENCES APPLIQUEES KHOURIBGA**

**Département : Mathématiques & Informatique**

**Filière : 1ère année Master – Big Data & Aide à la Décision**

**Sujet :**

**Optimisation sans contrainte**

**(Programmation Quadratique)**

|  |  |  |
| --- | --- | --- |
| **Réalisé par :** |  | **Supervisé par :** |
| **EL BASRI Akram**  **EL ALLAMI Ayoub**  **EL FADILI Salwa** |  | **Mr. EL HADFI Youssef** |

**Année universitaire** : 2023-2024

# TABLE DES MATIERES

[INTRODUCTION 1](#_Toc155556013)

[Applications quadratiques 2](#_Toc155556014)

[Programmation quadratique 4](#_Toc155556015)

[Exemple pratique 6](#_Toc155556016)

[Exercice en Python 7](#_Toc155556017)

[Annexes 8](#_Toc155556018)

# LISTE DE FIGURES

[Annexe 01 : main function 8](file:///C:\Users\admin\Desktop\EL_ALLAMI_EL_BASRI_EL_FADILI%5b2%5d.docx#_Toc155545252)

[Annexe 02 : input page 8](file:///C:\Users\admin\Desktop\EL_ALLAMI_EL_BASRI_EL_FADILI%5b2%5d.docx#_Toc155545253)

[Annexe 03 : Results page partie 2 9](file:///C:\Users\admin\Desktop\EL_ALLAMI_EL_BASRI_EL_FADILI%5b2%5d.docx#_Toc155545254)

[Annexe 04 : Results page part -1- 9](file:///C:\Users\admin\Desktop\EL_ALLAMI_EL_BASRI_EL_FADILI%5b2%5d.docx#_Toc155545255)

[Annexe 5 : Solve system function 9](file:///C:\Users\admin\Desktop\EL_ALLAMI_EL_BASRI_EL_FADILI%5b2%5d.docx#_Toc155545256)

[Annexe 06 : Read hessien matrix 9](file:///C:\Users\admin\Desktop\EL_ALLAMI_EL_BASRI_EL_FADILI%5b2%5d.docx#_Toc155545257)

[Annexe 07 : Compute characteristic polynomial function 9](file:///C:\Users\admin\Desktop\EL_ALLAMI_EL_BASRI_EL_FADILI%5b2%5d.docx#_Toc155545258)

[Annexe 08 : La page de saisir les paramètres (Matreice hessien & le vecteur b) part -1- 9](file:///C:\Users\admin\Desktop\EL_ALLAMI_EL_BASRI_EL_FADILI%5b2%5d.docx#_Toc155545259)

[Annexe 010 : La page des résultats - part 1- calcule les valeurs propres. 9](file:///C:\Users\admin\Desktop\EL_ALLAMI_EL_BASRI_EL_FADILI%5b2%5d.docx#_Toc155545260)

[Annexe 09 : La page de saisir les paramètres - part 2- validation d'entrées 9](file:///C:\Users\admin\Desktop\EL_ALLAMI_EL_BASRI_EL_FADILI%5b2%5d.docx#_Toc155545261)

[Annexe 11: Affichage des extremums s'elles existes. 9](file:///C:\Users\admin\Desktop\EL_ALLAMI_EL_BASRI_EL_FADILI%5b2%5d.docx#_Toc155545262)

# INTRODUCTION

L’optimisation sans contrainte (ou non contrainte) se réfère à la résolution de problèmes d'optimisation. Elle s'agit de trouver les valeurs des variables qui minimisent ou maximisent une certaine fonction, sans avoir à se conformer à des limitations spécifiques.

Les fonctions du second degré peuvent être appelées de différentes manières : trinômes, fonctions quadratiques ou encore fonctions polynomiales du second degré, avec un nombre quelconque de variables. Elles sont les fonctions les plus simples après les fonctions affines.

La programmation quadratique est considérée comme un cas particulier de la programmation non linéaire. L’optimisation quadratique est l’une des théories de la programmation mathématique la plus utilisée pour modéliser des problèmes pratiques. De nombreux domaines d’application s’appuient sur cette branche tels que : la recherche opérationnelle, la science de l’ingénieur, la physique (l’étude théorique d’une chute libre), les sciences économiques …

La représentation graphique d’une fonction du second degré est une parabole qui possède un axe de symétrie parallèle à l’axe des ordonnées. Le signe du coefficient associé au plus haut degré (ou coefficient du second degré) indique le sens de la variation de la fonction.

Ce rapport commence par une introduction, ensuite une partie concernant l’application quadratique, ensuite une partie sur la programmation quadratique en faisant déclarer quelques théorèmes, puis un exemple pratique expliquant plus cette théorie, et enfin un exercice en python.

# Applications quadratiques

**Définition**

On dit que f : Rn 🡪 R est une application quadratique si elle s’agit d’un polynôme de degré **2**.

Elle s’écrit sous la forme :

* La forme quadratique d’une variable :
* La forme quadratique de deux variables :
* La forme quadratique de trois variables :

Où a, b, c, d, e et f sont des nombres réels qui ne dépendent pas des variables x, y et z, avec a ≠ 0.

La notation matricielle de l’application quadratique est sous la forme :

Avec :

* **A** une matrice réelle d’ordre n
* **c** (c est une constante)

**Théorème (**Gradient et matrice Hessienne d’une application quadratique**) :**

Soit  une application quadratique. Donc, elle est infiniment différentiable.

Matrice Hessienne :

**Démonstration :**

f polynomiale ⇒ f est infiniment différentiable.

**∀ i = 1, 2, … , n** **⇒**

**∀ i, j = 1, 2, … , n ⇒**

**Programmation quadratique**

**Théorème (**Matrice semi-définie positive, définie positive, semi-définie négative, définie négative**) :**

Si f est une forme quadratique représentée par la matrice symétrique A :

* f est semi-définie positive si toutes les valeurs propres de A sont positives et si au moins l’une d’entre elles est nulle.
* f est définie positive si et seulement si toutes les valeurs propres de A sont positives.
* f est semi-définie négative si toutes les valeurs propres de A sont négatives et si au moins l’une d’entre elles est nulle.
* f est définie négative si et seulement si toutes les valeurs propres de A sont négatives.

**Théorème (**Caractérisation de la convexité à l’ordre 2**) :**

Si f est 2 fois différentiable sur un ouvert convexe U, alors :

* **∀ x U,** est semi-définie positive **⇔**  f est convexe sur U
* **∀ x U,**  est définie positive **⇒** f est strictement convexe

**Théorème (**Convexité d’une application quadratique**) :**

La convexité d’une application quadratique est totalement caractérisée par sa matrice Hessienne. En se basant sur le théorème précédent de la caractérisation de la convexité à l’ordre 2.

Soit  une application quadratique. Alors :

* **A** **est semi-définie positive** **⇔** f **convexe.**
* **A** **est définie positive** **⇔** **f est fortement convexe**

**⇔** **f est strictement convexe.**

**Démonstration :**

On a **: = A**.

Par définition, **A** est définie positive si et seulement si **f** est fortement convexe (autrement dit, la plus petite valeur propre de **A > 0**).

**f** fortement convexe **⇒** **f** strictement convexe.

Supposons que **f** est strictement convexe ; **f** convexe **⇒ A** semi-définie positive.

Par absurde :

Supposons que **A** n’est pas définie positive : ainsi **A** admet **0** pour valeur propre et soit le sous-espace propre associé **E0 = ker A** ; il est au moins d’une dimension et **∀ x** , **.** Le développement de **Taylor-Young** à l’ordre 2 s’écrit comme suit, puisque f est quadratique :

**∀ x**

Ainsi, **∀ x ,**

Donc, la restriction de **f** à **E0** est une application **affine** **⇒ f** convexe mais non strictement convexe. Contradiction : de **f** est strictement convexe. Ainsi, **A** est définie positive.

A partir de ce qu’on a vu précédemment, nous pouvons maintenant caractériser les extrema d’une application quadratique.

**Théorème (**Programmationquadratique**) :**

Soient une application quadratique sur **Rn** et **m Rn**.

* Si **A** est semi-définie positive (resp. Semi-définie négative), alors :
* **m** **est un minimum** (resp. Maximum) **local de f ;**
* **m** **est un minimum** (resp. Maximum) **global de f ;**
* **Am = b ; m** est une solution du système d’équations linéaires **AX = b**
* Si **A** est définie positive (resp. Négative), alors f admet un unique minimum (resp. Maximum) global.
* Si **A** n’est pas semi-définie positive (resp. Semi-définie négative), alors f n’admet aucun minimum (resp. maximum) local ou global.

**Exemple pratique**

Le but de cet exemple est de déterminer si la fonction admet un extrema global. Puis, préciser ses coordonnées.

Soit , avec et .

Le polynôme caractéristique de A est :

**,** où I est la matrice identité du même ordre que A. Dans ce cas, I = I3.

Puisque **A** est symétrique réelle, alors elle est diagonalisable.

Les valeurs propres de **A** sont toutes **> 0**.

Donc, **A** est définie positive ⇒ **f** admet un **unique minimum global**. (D’après le théorème ci-dessus).

Ce minimum global est la solution unique du système des équations linéaires suivant : **AX = b**

**AX = b** **⇔**

**⇔** Minimum de f

**Exercice en Python**

Notre application Streamlit, intitulée "Programmation Quadratique Application", offre une interface utilisateur interactive pour explorer les propriétés d'une fonction quadratique définie par sa matrice hessienne. Cette application, entièrement implémentée en Python, fournit une approche visuelle et conviviale permettant à l'utilisateur de saisir les éléments de la matrice hessienne et du vecteur b, de visualiser les résultats et d'interpréter les implications mathématiques associées.

**Fonctionnement de l'Application :**

* **Page de Saisie des Paramètres :**

Sur cette page, l'utilisateur est invité à spécifier les dimensions de la matrice hessienne, ainsi que les éléments de celle-ci et du vecteur b. Grâce à des widgets interactifs tels que des champs de saisie numérique et des boutons, l'utilisateur peut entrer les données nécessaires pour caractériser la fonction quadratique.

Après avoir validé les données, l'application utilise des fonctions Python sous-jacentes pour calculer le polynôme caractéristique, les valeurs propres de la matrice hessienne, et présenter un aperçu des résultats.

* **Affichage des Résultats :**

Cette page présente les résultats calculés à partir des données saisies par l'utilisateur. Elle affiche la matrice hessienne, le vecteur b, le polynôme caractéristique, ainsi que les valeurs propres associées. En fonction des propriétés des valeurs propres, l'application offre une analyse qualitative de la nature de la fonction quadratique, indiquant si elle est convexe, concave, ou si les propriétés ne sont pas déterminées uniquement par les valeurs propres.

De plus, si la fonction est caractérisée comme strictement convexe, l'application résout également le système linéaire associé HX = b pour déterminer le minimum global. Réciproquement s’elle est concave elle déterminer le maximum global.

(Toutes les fonctions implémentées dans cette application se trouvent en annexe ci-dessous.)

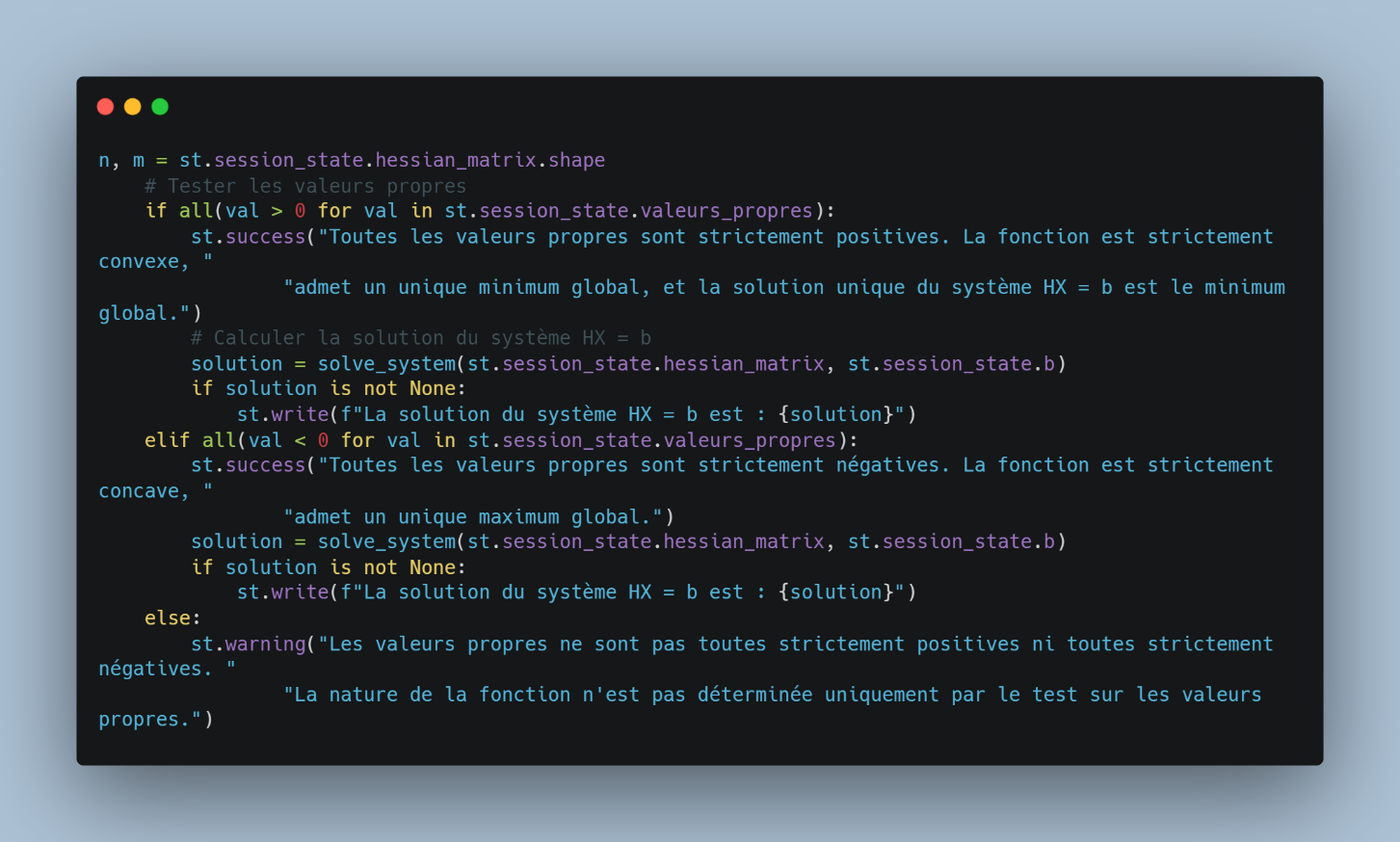
**Annexes**

Annexe 01 : La fonction principale :

Annexe 01 : main function

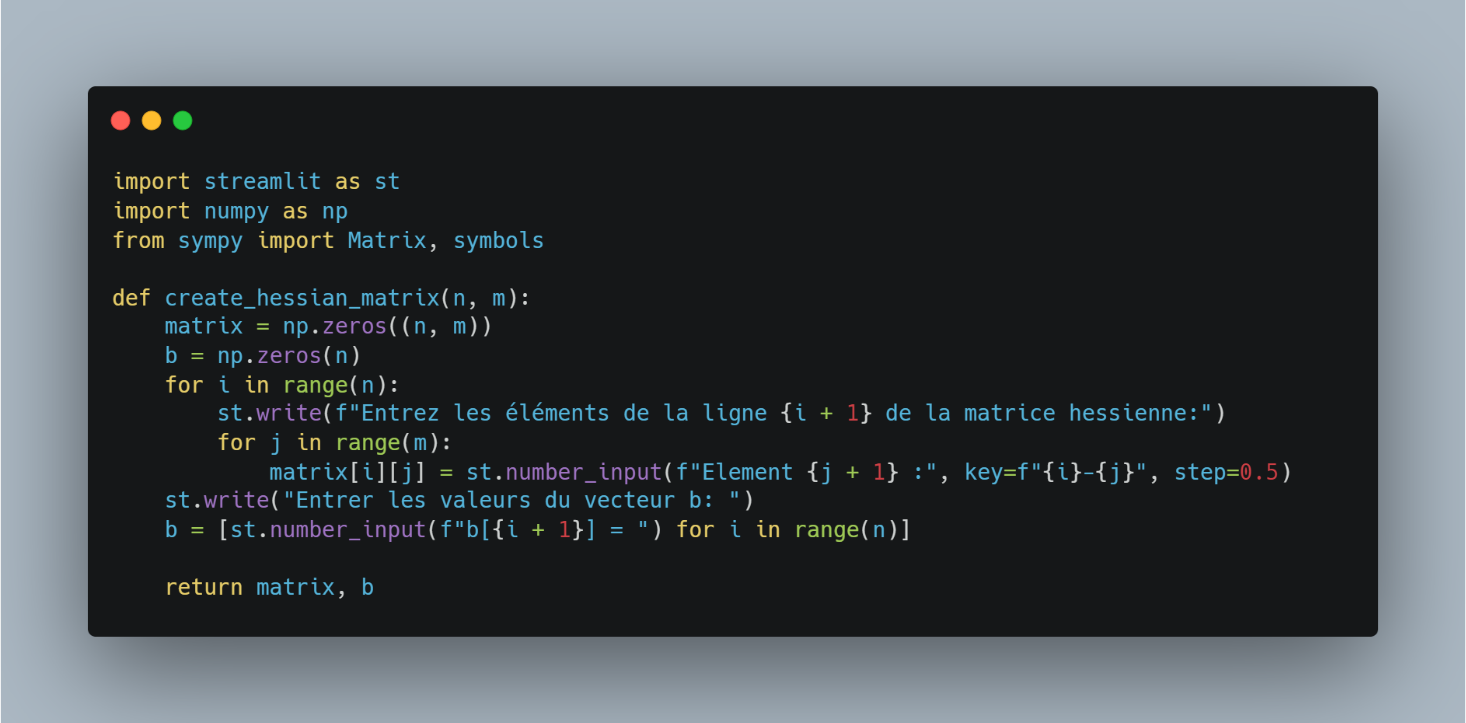
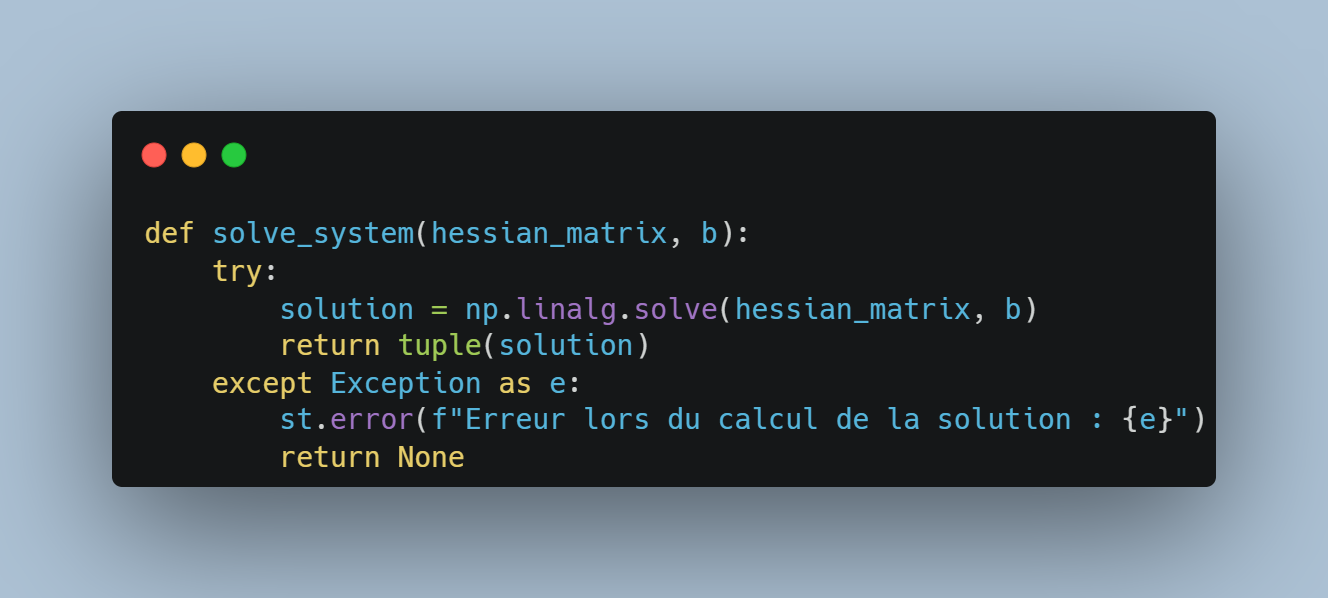
Annexe 02 : La page d’entrer:

Annexe 02 : input page

Annexe 03 : La page de résultats:

Annexe 03 : Results page partie 2

Annexe 04 : Results page part -1-

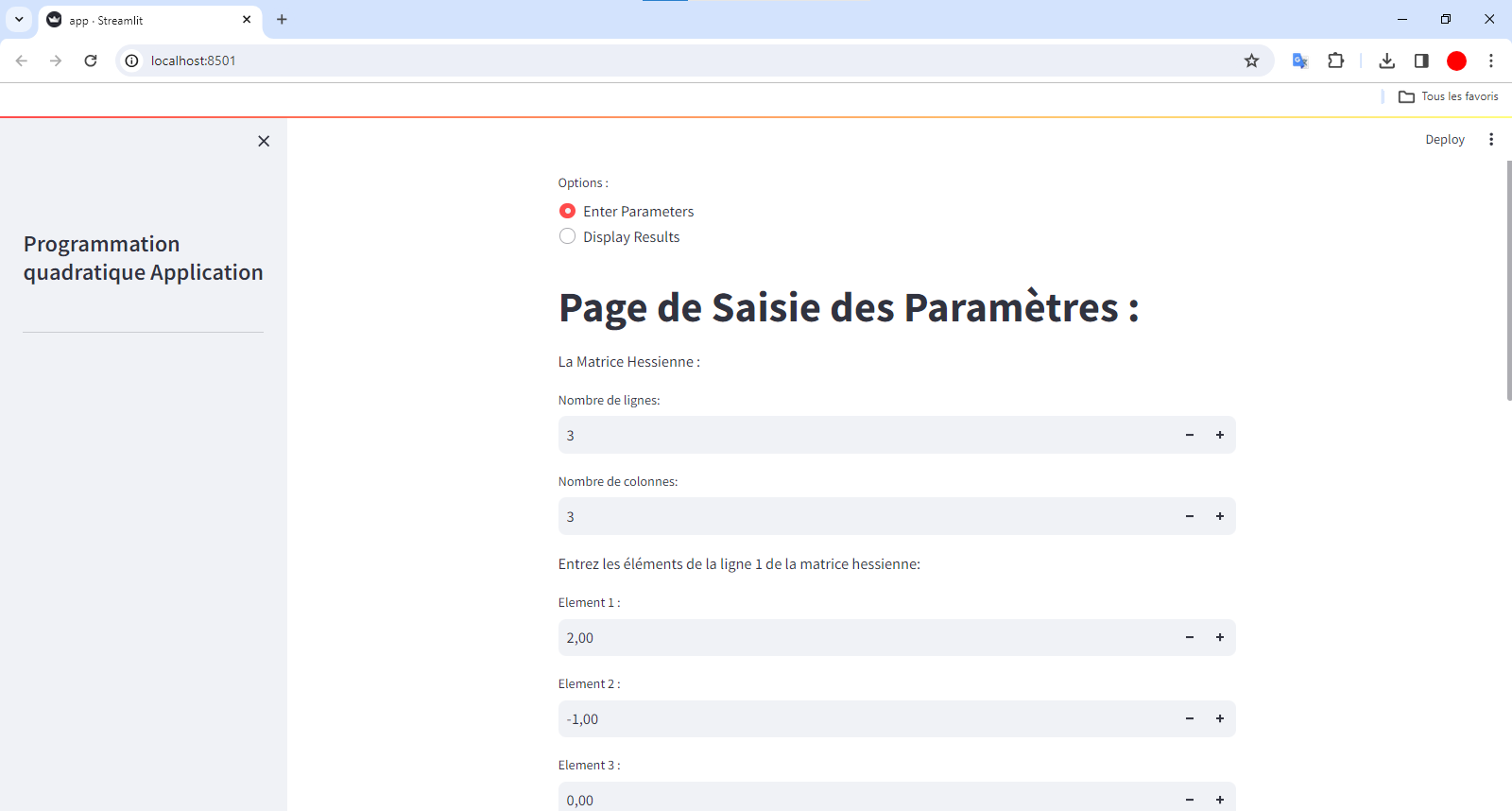
Annexe 04 : Des fonctions supplémentaires:

Annexe 5 : Solve system function

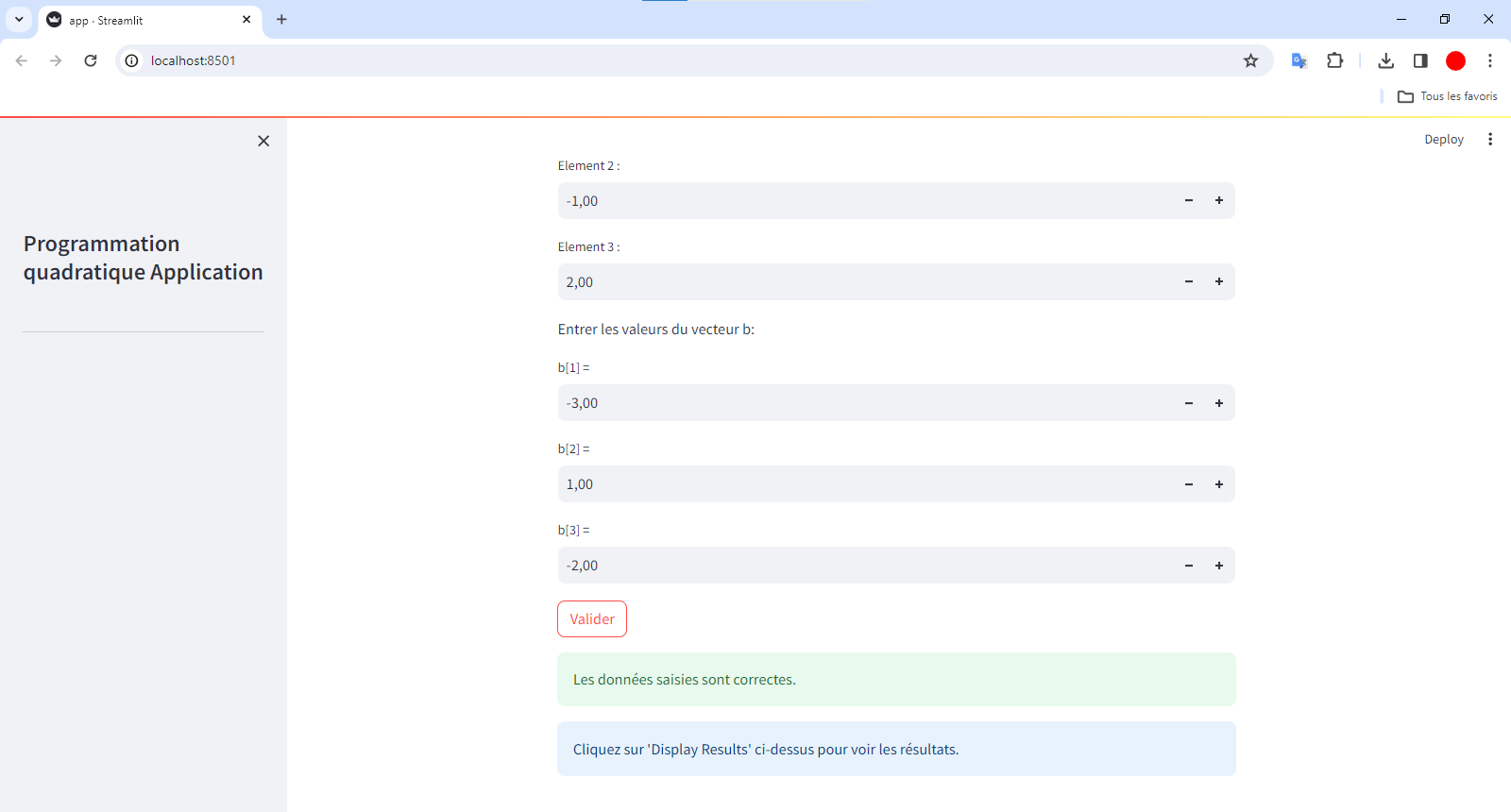
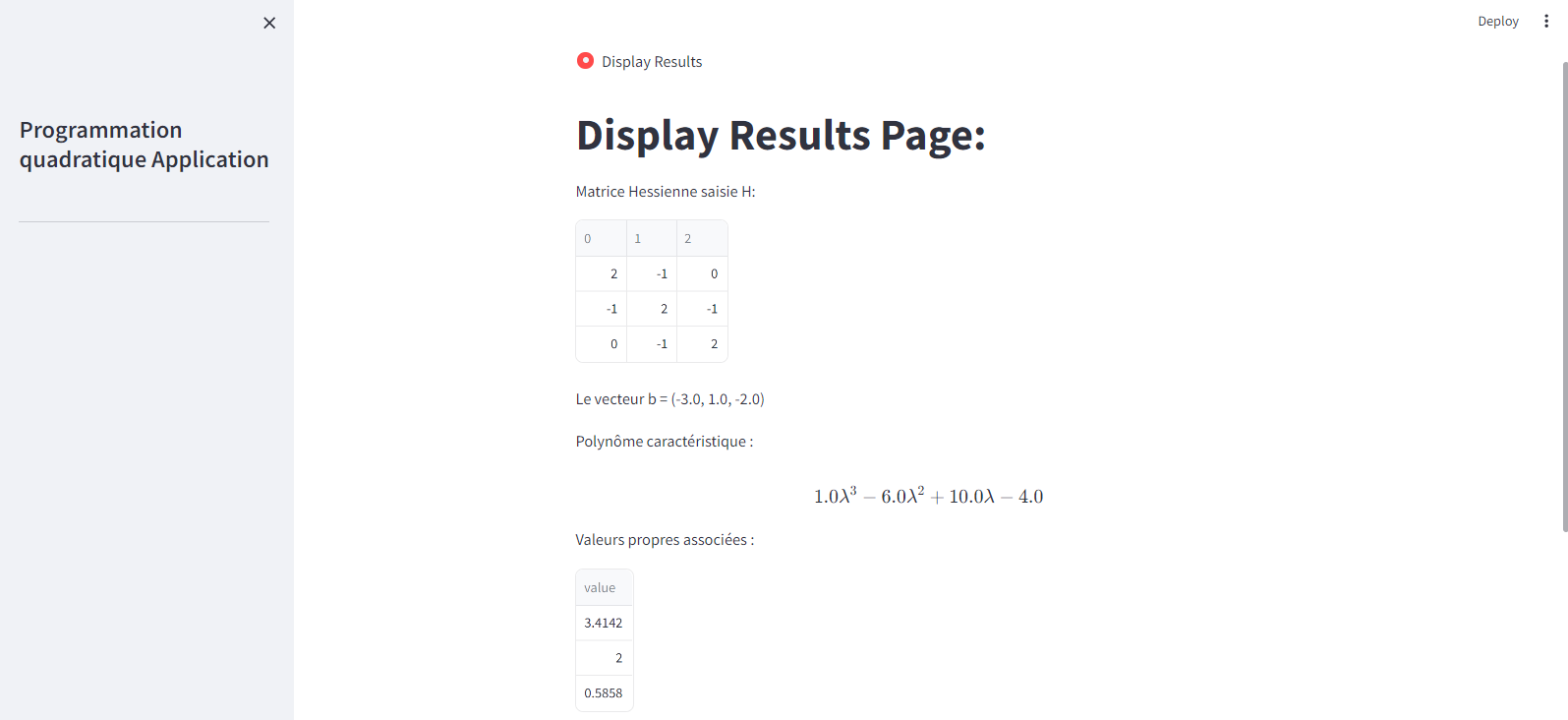
Annexe 06 : Read hessien matrix



Annexe 07 : Compute characteristic polynomial function

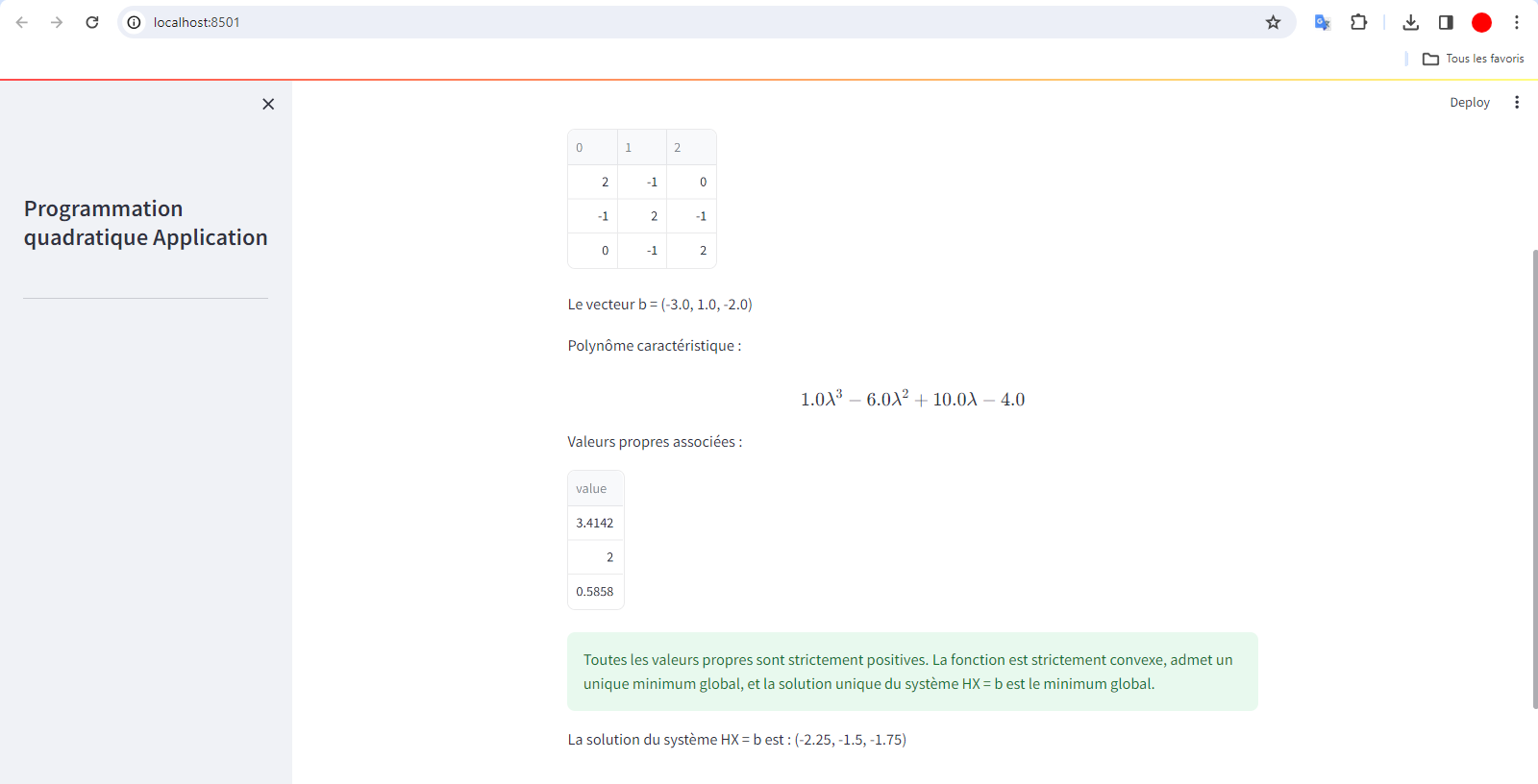
Annexe 06 : Les interfaces de notre application python :

Annexe 08 : La page de saisir les paramètres (Matreice hessien & le vecteur b) part -1-



Annexe 09 : La page des résultats - part 1- calcule les valeurs propres.

Annexe 010 : La page de saisir les paramètres - part 2- validation d'entrées



Annexe 11: Affichage des extremums s'elles existes.