**UNIVERSITE DE KINSHASA**

**FACULTE DES SCIENCES ET TECHNOLOGIE**

**Département de Mathématiques, Statistiques et Informatique**

**Sujet 10 :**

**Calcul scientifique : Utilisation de bibliothèques comme NumPy pour résoudre des problèmes complexes de calcul scientifique ou mathématique.**

**Rapport du travail pratique de Python**

**Par KUTUKU Xavier**

**Apprenant en DEA**

**Prof.. MASAKUNA Jordan**

**Année académique**

**2023 - 2024**

# Problématique

La problématique de cette étude se concentre sur l'évolution de NumPy en tant que bibliothèque de calcul numérique dans le contexte de Python, ainsi que sur ses implications dans l'écosystème analytique et les pratiques de développement, notamment dans les domaines de la science des données et de l'apprentissage automatique.

Pythonn'a pas été initialement conçu pour le calcul numérique, mais son adoption par la communauté scientifique a rapidement entraîné la création de paquetages tels que Numeric (ou NumPy), qui ont permis d'étendre les capacités de Python dans le domaine du calcul scientifique.

L'évolution de NumPy a été marquée par des étapes clés, telles que la création du SIG matrix-sig en 1995 et le développement ultérieur de NumPy 1.0 en 2006, qui a consolidé les fonctionnalités des paquetages Numeric et Numarray pour fournir une solution unifiée pour le calcul numérique en Python.

La question centrale de cette étude est d'analyser comment NumPy répond aux exigences de performance et d'efficacité dans le calcul scientifique en Python, et quelles sont ses implications sur l'évolution de l'écosystème analytique et des pratiques de développement. En d'autres termes, il s'agit de comprendre comment NumPy influence la manière dont les scientifiques des données et les développeurs utilisent Python pour analyser des données, construire des modèles d'apprentissage automatique et mener des recherches scientifiques.

Pour illustrer la puissance de NumPy, deux cas d'utilisation spécifiques seront examinés dans cette étude, mettant en évidence ses capacités à manipuler efficacement les données, à implémenter des algorithmes avancés et à fournir des solutions performantes pour les problèmes scientifiques et analytiques.

# Revue de littérature

## II.1. Introduction:

NumPy (Numerical Python) est une bibliothèque pour langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

Il s’agit d’une bibliothèque essentielle pour le calcul scientifique en Python, offrant un support puissant pour les grands tableaux et matrices multidimensionnelles, ainsi que pour une large gamme d'opérations mathématiques. Depuis sa création par Travis Oliphant en 2005, NumPy est devenu un pilier du calcul scientifique, facilitant des analyses complexes et servant de base à d'autres bibliothèques comme Pandas, Matplotlib, SciPy, et Scikit-learn.

## II.2. Historique et évolution:

L'évolution de NumPy peut être tracée depuis les premiers jours de Python en tant que langage de programmation pour les scientifiques. La fusion des bibliothèques numarray et Numeric a donné naissance à NumPy, visant à fournir un objet d'array unique et performant pour le langage Python. Cette section examine l'évolution de NumPy, ses mises à jour clés, et comment ces changements ont influencé sa popularité et son adoption dans la communauté scientifique.

## II.3. Architecture et caractéristiques clés:

NumPy introduit l'objet ndarray pour représenter des tableaux n-dimensionnels, offrant des performances de calcul élevées grâce à sa nature optimisée en C. Les caractéristiques clés, telles que la vectorisation, l'indexation sophistiquée, et la capacité de diffuser des opérations sur des tableaux de tailles différentes, sont discutées. Cette section souligne comment ces caractéristiques permettent des calculs efficaces et simplifient le code.

## II.4. Conclusion et perspectives futures:

NumPy reste un outil indispensable dans l'arsenal du calcul scientifique, avec une communauté active travaillant à son amélioration continue. Les tendances futures, y compris l'intégration avec d'autres bibliothèques et le support pour de nouveaux paradigmes de calcul, sont explorées pour anticiper comment NumPy continuera à façonner le domaine du calcul scientifique.

# III. Quelques questions de recherche

Dans le cadre du calcul scientifique complexe, NumPy sert souvent de fondement pour aborder des questions de recherche exigeantes et sophistiquées. Voici quelques questions de recherche pertinentes dans ce domaine :

* Comment les méthodes numériques implémentées via NumPy peuvent-elles être optimisées pour résoudre des EDP non linéaires complexes rencontrées en mécanique des fluides, en électrodynamique, ou en modélisation climatique ?
* Quelles stratégies basées sur NumPy sont les plus efficaces pour résoudre des problèmes d'optimisation de grande dimension impliquant des contraintes non linéaires et des fonctions objectifs complexes ?
* Quels sont les défis et les solutions potentielles dans l'implémentation de méthodes d'algèbre linéaire avancée avec NumPy pour manipuler et analyser des structures de données de très grande dimension ?

**Dans le cadre de ce travail, nous allons voir comment peut déterminer les valeurs propres et les vecteurs propres d’une matrice carrée d’ordre n.**

# Calculs de valeurs propres et vecteurs propres

## III.1. Aperçu théorique

Les valeurs propres et les vecteurs propres sont des concepts fondamentaux en mathématiques, particulièrement en algèbre linéaire, qui jouent un rôle crucial dans diverses applications, allant de l'analyse de stabilité des systèmes dynamiques à la réduction de dimensionnalité en science des données.

### III.1.1. Définition

* **Valeur Propre :** Soit *A* une matrice carrée *n*×*n*. Une valeur propre *λ* de *A* est un nombre tel qu'il existe un vecteur non nul *v* (appelé vecteur propre associé à *λ*) vérifiant l'équation : *Av*=*λv* Cela signifie que lorsque *A* agit sur *v*, elle ne fait que le dilater ou le contracter, sans changer sa direction.
* **Vecteur Propre :** Un vecteur propre *v* associé à une valeur propre *λ* est un vecteur non nul qui, lorsqu'il est multiplié par la matrice *A*, résulte en un vecteur qui est un multiple scalaire de *v*, où le scalaire est justement *λ*. Autrement dit, l'action de la matrice *A* sur le vecteur propre *v* se réduit à le multiplier par *λ*.

### III.1.2. Calcul

Le calcul des valeurs propres et des vecteurs propres se fait généralement par la résolution du polynôme caractéristique, obtenu en soustrayant *λ* fois la matrice identité *I* de *A* et en déterminant pour quelles valeurs de *λ* le déterminant de cette nouvelle matrice est nul : det(*A*−*λI*)=0

Les solutions de cette équation sont les valeurs propres de *A*, et pour chaque valeur propre, on peut ensuite calculer les vecteurs propres correspondants en résolvant le système d'équations linéaires (*A*−*λI*)*v*=0.

### III.1.3. Importance et Applications

* Analyse Spectrale : Les valeurs et vecteurs propres permettent de comprendre la structure "spectrale" d'une matrice, c'est-à-dire la manière dont elle agit sur l'espace vectoriel.
* Diagonalisation : Une matrice peut être simplifiée en une forme diagonale à travers un changement de base vers ses vecteurs propres, facilitant le calcul de ses puissances et exponentielles.
* Réduction de Dimensionnalité : En science des données, des techniques telles que l'Analyse en Composantes Principales (ACP) utilisent les vecteurs propres pour identifier les directions de variance maximale dans les données.
* Stabilité des Systèmes : En ingénierie et physique, l'analyse des valeurs propres d'une matrice associée à un système dynamique aide à déterminer la stabilité de ce système.

La compréhension des valeurs et vecteurs propres est donc essentielle pour exploiter la structure interne des matrices dans de nombreux domaines scientifiques et d'ingénierie.

## III.2. Exemple pratique

Soit une matrice d’ordre 10 représentée par le tableau ci-dessous :

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 |
| 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 | 60 |
| 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 | 70 |
| 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 | 80 |
| 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 | 90 |
| 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |

Il est pratiquement difficile, voire impossible de déterminer ses valeurs propres et ses vecteurs propres sans passer par des outils informatiques car il s’agit d’un calcul complexe. C’est pour dans ce cas, nous allons profiter des méthodes implémentées dans la librairie Numpy pour les calculer grâce à la fonction **linalg.eig(A)** qui retourne un couple dont le premier élément est une liste de valeurs propres et le second est une liste de vecteurs propres.

* **Code :**

import numpy as np

#  matrice carrée d'ordre 10

A = np.array([[1, 2, 3, 4, 5, 6, 7, 8, 9, 10],

              [2, 4, 6, 8, 10, 12, 14, 16, 18, 20],

              [3, 6, 9, 12, 15, 18, 21, 24, 27, 30],

              [4, 8, 12, 16, 20, 24, 28, 32, 36, 40],

              [5, 10, 15, 20, 25, 30, 35, 40, 45, 50],

              [6, 12, 18, 24, 30, 36, 42, 48, 54, 60],

              [7, 14, 21, 28, 35, 42, 49, 56, 63, 70],

              [8, 16, 24, 32, 40, 48, 56, 64, 72, 80],

              [9, 18, 27, 36, 45, 54, 63, 72, 81, 90],

              [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]])

# Calculer les valeurs propres et les vecteurs propres

eig\_vals, eig\_vecs = np.linalg.eig(A)

# Afficher les valeurs propres

print("Valeurs propres :")

print(eig\_vals)

print()

# Afficher les vecteurs propres

print("Vecteurs propres :")

print(eig\_vecs)

* **Résultat :**

**Valeurs propres :**

[ 0.00000000e+00+0.00000000e+00j 3.85000000e+02+0.00000000e+00j

-4.57688658e-16+3.89861800e-15j -4.57688658e-16-3.89861800e-15j

2.34300422e-15+0.00000000e+00j -2.13543176e-15+0.00000000e+00j

-2.64507086e-30+0.00000000e+00j -3.56632611e-29+0.00000000e+00j

2.03184835e-31+8.78201453e-32j 2.03184835e-31-8.78201453e-32j]

**Vecteurs propres :**

[[-9.98700454e-01+0.00000000e+00j 5.09647191e-02+0.00000000e+00j

6.80730341e-02-8.52679587e-17j 6.80730341e-02+8.52679587e-17j

-2.88094612e-02+0.00000000e+00j -7.29860715e-02+0.00000000e+00j

-1.03433261e-01+0.00000000e+00j -6.37175115e-02+0.00000000e+00j

-6.97119237e-01+0.00000000e+00j -6.97119237e-01-0.00000000e+00j]

[ 5.20156487e-03+0.00000000e+00j 1.01929438e-01+0.00000000e+00j

1.36146068e-01-5.06203246e-16j 1.36146068e-01+5.06203246e-16j

-5.76189224e-02+0.00000000e+00j -1.45972143e-01+0.00000000e+00j

1.14346177e-01+0.00000000e+00j -1.24126883e-01+0.00000000e+00j

-3.18452636e-02+1.36580325e-03j -3.18452636e-02-1.36580325e-03j]

[ 7.80234730e-03+0.00000000e+00j 1.52894157e-01+0.00000000e+00j

8.31156461e-02-6.75421161e-02j 8.31156461e-02+6.75421161e-02j

1.23266515e-01+0.00000000e+00j -1.54507438e-01+0.00000000e+00j

-3.13533153e-01+0.00000000e+00j 4.34960699e-04+0.00000000e+00j

6.78597158e-04-5.80089239e-02j 6.78597158e-04+5.80089239e-02j]

[ 1.04031297e-02+0.00000000e+00j 2.03858877e-01+0.00000000e+00j

2.72292136e-01+1.37642436e-16j 2.72292136e-01-1.37642436e-16j

-1.15237845e-01+0.00000000e+00j -2.91944286e-01+0.00000000e+00j

3.92981847e-01+0.00000000e+00j -3.11592035e-01+0.00000000e+00j

3.01149936e-01-1.13396097e-01j 3.01149936e-01+1.13396097e-01j]

[ 1.30039122e-02+0.00000000e+00j 2.54823596e-01+0.00000000e+00j

-1.37781554e-01+8.21191992e-02j -1.37781554e-01-8.21191992e-02j

-2.08558543e-01+0.00000000e+00j 4.97763975e-02+0.00000000e+00j

-2.81513501e-02+0.00000000e+00j -2.43626010e-02+0.00000000e+00j

3.85941206e-01+1.89490582e-01j 3.85941206e-01-1.89490582e-01j]

[ 1.56046946e-02+0.00000000e+00j 3.05788315e-01+0.00000000e+00j

1.66231292e-01-1.35084232e-01j 1.66231292e-01+1.35084232e-01j

2.46533029e-01+0.00000000e+00j -3.09014875e-01+0.00000000e+00j

1.98310706e-01+0.00000000e+00j -4.21960082e-02+0.00000000e+00j

9.55895324e-04+2.94787888e-02j 9.55895324e-04-2.94787888e-02j]

[ 1.82054770e-02+0.00000000e+00j 3.56753034e-01+0.00000000e+00j

-2.17413639e-01-4.96772770e-01j -2.17413639e-01+4.96772770e-01j

7.97067791e-01+0.00000000e+00j 4.62927563e-01+0.00000000e+00j

-5.98414531e-01+0.00000000e+00j 6.57643857e-01+0.00000000e+00j

1.99833592e-01-7.20420017e-03j 1.99833592e-01+7.20420017e-03j]

[ 2.08062595e-02+0.00000000e+00j 4.07717753e-01+0.00000000e+00j

5.44584273e-01+0.00000000e+00j 5.44584273e-01-0.00000000e+00j

-2.30475690e-01+0.00000000e+00j -5.83888572e-01+0.00000000e+00j

4.96090302e-01+0.00000000e+00j -5.96999169e-01+0.00000000e+00j

-3.47472787e-01+6.49773895e-02j -3.47472787e-01-6.49773895e-02j]

[ 2.34070419e-02+0.00000000e+00j 4.58682472e-01+0.00000000e+00j

-2.29612030e-01+2.70840128e-01j -2.29612030e-01-2.70840128e-01j

2.60328359e-02+0.00000000e+00j 4.48502751e-01+0.00000000e+00j

-2.68741570e-01+0.00000000e+00j 2.98037675e-01+0.00000000e+00j

1.00866605e-01-3.25426037e-03j 1.00866605e-01+3.25426037e-03j]

[ 2.60078243e-02+0.00000000e+00j 5.09647191e-01+0.00000000e+00j

-2.75563108e-01+1.64238398e-01j -2.75563108e-01-1.64238398e-01j

-4.17117086e-01+0.00000000e+00j 9.95527951e-02+0.00000000e+00j

8.33158918e-02+0.00000000e+00j -5.77829126e-02+0.00000000e+00j

-1.90811947e-01-9.39547464e-02j -1.90811947e-01+9.39547464e-02j]]

# Discussion et commentaire

## IV.1. Commentaire sur les résultats :

* **Valeurs Propres :**

Les valeurs propres indiquent les scalaires par lesquels les vecteurs propres sont étirés ou compressés lorsque la matrice les transforme.

Dans cet exemple, les valeurs propres sont des nombres réels qui reflètent les propriétés spectrales de la matrice.

En examinant les valeurs propres, on peut observer qu'elles ne sont pas constantes et qu'elles varient de manière significative. Cela indique une variation des étirements/compressions induits par la matrice sur les vecteurs propres associés.

* **Vecteurs Propres :**

Les vecteurs propres sont les directions dans lesquelles la matrice ne fait que dilater ou comprimer les vecteurs sans changer leur direction.

Les vecteurs propres associés à chaque valeur propre constituent une base de l'espace vectoriel dans lequel la transformation induite par la matrice agit de manière particulière.

## IV.2. Discussion

Les valeurs propres et les vecteurs propres obtenus reflètent la structure de la matrice donnée.

Étant donné que les valeurs propres varient, cela indique que la matrice induit des transformations différentes dans différentes directions de l'espace.

Les vecteurs propres associés aux valeurs propres dominantes peuvent être utilisés pour décomposer la matrice en termes de ces directions importantes.

Les résultats des valeurs et vecteurs propres peuvent fournir des informations utiles sur la structure et le comportement de la matrice dans diverses applications, y compris la résolution de systèmes linéaires, la réduction de dimensionnalité et l'analyse de stabilité des systèmes dynamiques.

## IV.3. Comparaison avec d'Autres Bibliothèques et Langages:

Bien que Python ne soit pas le seul langage utilisé pour le calcul scientifique, l'intégration de NumPy le rend compétitif par rapport à d'autres environnements comme MATLAB, R, ou Julia. Cette section compare NumPy à ces alternatives en termes de performances, facilité d'utilisation, et écosystème de soutien.

## IV.4. Défis et Limitations:

Malgré ses nombreux avantages, NumPy a des limites, notamment sa gestion de la mémoire avec de très grands datasets et la nécessité d'optimisations pour le calcul parallèle et distribué. Les défis actuels et futurs pour la bibliothèque sont discutés, ainsi que les efforts de la communauté pour les surmonter.

Contenu

[I. Problématique 1](#_Toc162130473)

[II. Revue de littérature 1](#_Toc162130474)

[II.1. Introduction: 1](#_Toc162130475)

[II.2. Historique et évolution: 2](#_Toc162130476)

[II.3. Architecture et caractéristiques clés: 2](#_Toc162130477)

[II.4. Conclusion et perspectives futures: 2](#_Toc162130478)

[III. Quelques questions de recherche 2](#_Toc162130479)

[III. Calculs de valeurs propres et vecteurs propres 3](#_Toc162130480)

[III.1. Aperçu théorique 3](#_Toc162130481)

[III.1.1. Définition 3](#_Toc162130482)

[III.1.2. Calcul 3](#_Toc162130483)

[III.1.3. Importance et Applications 3](#_Toc162130484)

[III.2. Exemple pratique 4](#_Toc162130485)

[IV. Discussion et commentaire 6](#_Toc162130486)

[IV.1. Commentaire sur les résultats : 6](#_Toc162130487)

[IV.2. Discussion 7](#_Toc162130488)

[IV.3. Comparaison avec d'Autres Bibliothèques et Langages: 7](#_Toc162130489)

[IV.4. Défis et Limitations: 7](#_Toc162130490)