Calcul Symbolique en Python : La Bibliothèque SymPy

MEFFO Lea 22U2194
KEMBOU Richel 22U2118
KELODJOU Ivana 22T2894
TSEMEGNE Martin 22U2080

Université de Yaoundé I

(jecy.meffo, richel.kembou, ivana.kelodjou, yvan.tsemegne)@facsciences-uy1.cm

October 2, 2025

SOMMAIRE

1. Introduction et Contexte
2. SymPy : La Boîte à Outils
3. Application : Le Calcul du Gradient
4. Conclusion

La Problématique

Comment les ordinateurs gèrent-ils *vraiment* les résultats des calculs ?

L'approche Numérique

Calcule avec des approximations. **Inconvénient :** Perte de précision.

`sqrt(8)**2` $\rightarrow$ `8.000000...2`

L'approche Symbolique

Manipule les expressions. **Avantage** : Exactitude garantie.

$(\sqrt{8})^2 \rightarrow 8$

SymPy est la solution Python pour le calcul symbolique.

Calcul Symbolique vs. Calcul Numérique

Calcul Numérique (Ex: NumPy)

- Travaille avec des nombres (approximations flottantes).
- **Exemple :** $\sqrt{2} \approx 1.41421356$
- **Idéal pour :** Calculs intensifs, simulations, analyse de données.

Calcul Symbolique vs. Calcul Numérique

Calcul Numérique (Ex: NumPy)

- Travaille avec des nombres (approximations flottantes).
- **Exemple :** $\sqrt{2} \approx 1.41421356$
- **Idéal pour :** Calculs intensifs, simulations, analyse de données.

Calcul Symbolique (Ex: SymPy)

- Travaille avec des symboles et des expressions mathématiques.
- **Exemple :** $\sqrt{2}$ reste $\sqrt{2}$. L'expression est exacte.
- **Idéal pour :** Algèbre, calcul différentiel, résolution exacte d'équations.

Présentation de SymPy

Qu'est-ce que SymPy ?

Une bibliothèque Python **pure** pour les mathématiques symboliques.

- **Open-source** et activement développée.
- **Légère** et sans dépendances lourdes.
- S'intègre parfaitement dans l'écosystème scientifique Python.
- Permet de traiter les mathématiques comme un mathématicien.

SymPy est l'outil pour des mathématiques exactes en Python.

Fonctionnalités Clés:

Algèbre

- Simplification : `simplify(sin(x)**2 + cos(x)**2)` → 1
- Factorisation : `factor(x**2 - 1)` → `(x - 1)(x+1)`
- Résolution d'équations : `solve(x**2 - 4, x)` → `[-2, 2]`

Calcul Différentiel

- Dérivées : `diff(x**3, x)` → `3*x**2`
- Intégrales : `integrate(cos(x), x)` → `sin(x)`

Calcul Matriciel

- Matrices symboliques
- Déterminant, inverse...

Le Gradient : Concept et Utilité

Définition

Le gradient d'une fonction $f(x_1, \ldots, x_n)$ est le vecteur de ses dérivées partielles :

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

Le Gradient : Concept et Utilité

Définition

Le gradient d'une fonction $f(x_1, \ldots, x_n)$ est le vecteur de ses dérivées partielles :

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

Utilité Fondamentale : L'Optimisation

- Le gradient pointe toujours dans la **direction de la plus forte pente**.
- C'est le moteur de la **Descente de Gradient**, un algorithme essentiel en Machine Learning.

Cas Pratique : Minimiser une Fonction de Coût

Objectif

Trouver le minimum de la fonction $L(x, y) = 2x^2 + 3y^2 - 4x + 6y + 10$.

Cas Pratique : Minimiser une Fonction de Coût

Objectif

Trouver le minimum de la fonction $L(x, y) = 2x^2 + 3y^2 - 4x + 6y + 10$.

Démarche

Le point minimum d'une fonction est atteint lorsque son gradient est nul.

1. Calculer le gradient symbolique $\nabla L(x, y)$.
2. Résoudre l'équation $\nabla L(x, y) = \vec{0}$.

Cas Pratique : Minimiser une Fonction de Coût

Objectif

Trouver le minimum de la fonction $L(x, y) = 2x^2 + 3y^2 - 4x + 6y + 10$.

Démarche

Le point minimum d'une fonction est atteint lorsque son gradient est nul.

1. Calculer le gradient symbolique $\nabla L(x, y)$.
2. Résoudre l'équation $\nabla L(x, y) = \vec{0}$.

SymPy est l'outil parfait pour la première étape.

Interprétation et Résolution

Résultat Symbolique Obtenu avec SymPy

SymPy nous donne l'expression exacte du gradient :

$$\nabla L(x, y) = \begin{pmatrix} 4x - 4 \\ 6y + 6 \end{pmatrix}$$

Interprétation et Résolution

Résultat Symbolique Obtenu avec SymPy

SymPy nous donne l'expression exacte du gradient :

$$\nabla L(x, y) = \begin{pmatrix} 4x - 4 \\ 6y + 6 \end{pmatrix}$$

Trouver le Point Critique

On résout le système $\nabla L(x, y) = 0$:

- $4x - 4 = 0 \implies x = 1$
- $6y + 6 = 0 \implies y = -1$

Conclusion de l'Application

SymPy a transformé un problème d'optimisation en une simple résolution de système d'équations.

Conclusion

Ce qu'il faut retenir

- Le **calcul symbolique** est essentiel pour obtenir des résultats mathématiques **exacts**.
- **SymPy** est la bibliothèque de choix en Python : puissante, légère et intuitive.
- Il permet de résoudre des problèmes complexes de manière **élégante et précise**.

Pensez à SymPy lorsque la précision prime sur la vitesse de calcul brute.

Questions ?

Merci de votre attention.