

Calcul Scientifique

Cours 1: Représentation des données

Alexis Lechervy



Sommaire

- 1 Présentation du cours
- 2 Représentation des données pour le calcul scientifique

Présentation du cours de Calcul Scientifique

Objectifs

- Acquérir une méthodologie de travail pour **modéliser** et **résoudre** à l'aide de l'informatique des problèmes nécessitant la mise en place de calculs scientifiques,
- Savoir **modéliser les données**, les éléments d'un **problème concret**, à l'aide d'outils à la fois **mathématiques et informatiques**,
- **Mettre en œuvre des solutions** mathématiques et algorithmique apportant une solution numérique à ces problèmes.

Organisation

- 10 CM de 1h15 le mercredi 11h-12h15,
- 10 TP de 1h15 le lundi 15h15-16h30 ou 16h45-18h00,
- 1/2 Note de CC sur les TPs + 1/2 Note de CT 1h.
- Ressources : <https://ecampus.unicaen.fr/course/view.php?id=14888>

Sommaire

1 Présentation du cours

2 Représentation des données pour le calcul scientifique

- Introduction
- Les vecteurs
- Les matrices
- Les tenseurs

Outils informatiques utilisés dans ce cours

Langage de Programmation

- **Python 3** : <https://www.python.org/>
- Documentation : <https://docs.python.org/3/>

Bibliothèque de calcul

- **Numpy** : <http://www.numpy.org/>
- Documentation : <https://docs.scipy.org/doc/numpy/index.html>
- Numpy permet la **manipulation de données sous forme de tableau** et met à disposition les mathématiques qui leurs sont associés.
- Déclaration en début de fichier : `import numpy as np`

Bibliothèque d'affichage

- **Matplotlib** : <https://matplotlib.org/>
- **Matplotlib** est librairies open source python permettant de visualiser des données sous formes de graphiques.
- Déclaration en début de fichier : `import matplotlib.pyplot as plt`

Les vecteurs

Les vecteurs en informatique

Un vecteur est un tableau de nombre (à une dimension). Il permet de regrouper sous une même appellation une liste de nombre.

Création de vecteur en numpy

- Vecteur v de 10 zéros : $v = \text{np.zeros}(10)$
- Vecteur v de 10 un : $v = \text{np.ones}(10)$
- Vecteur v de 10 deux : $v = 2 * \text{np.ones}(10)$
- Vecteur v des nombres entiers de 3 à 10 compris : $v = \text{np.arange}(3, 11)$
- Vecteur v de valeur $[42, 13, 38, 51]$: $v = \text{np.array}([42, 13, 38, 51])$

Lien avec les listes python

- Transformer une liste l en vecteur : $v = \text{np.array}(l)$
- Transformer un vecteur v en liste python : $l = \text{list}(v)$
- Attention les opérations mathématiques ne marche pas sur les listes python.

Opérations sur les vecteurs

Dimension d'un vecteur v

`v.shape`

Retourne une liste d'un élément, contenant la dimension du vecteur v .

Opération entre un nombre et un vecteur

- Additionner 2 à toutes les dimensions du vecteur : $v2 = v + 2$
- Multiplier 2 à toutes les dimensions du vecteur : $v2 = 2*v$
- Diviser 2 à toutes les dimensions du vecteur : $v2 = v/2$

Fonction de base appliquer à toutes les valeurs d'un vecteur

Liste des fonctions disponibles : <https://docs.scipy.org/doc/numpy/reference/routines.math.html>

- `np.sin`, `np.cos`, `np.tan`, `np.arcsin` ...
- `np.sinh`, `np.cosh`, `np.tanh`, `np.arcsinh` ...
- `np.exp`, `np.log`, `np.log10`, ...
- `np.sqrt`, `np.fabs`, `np.sign`, `np.square`...

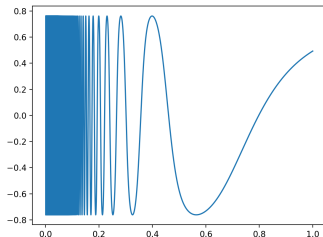
Exemple : `v2 = np.sin(v)`

Exemple d'application

Tracer une fonction donnée

- Fonction à tracer :
 $f(x) = \tanh(\cos(\frac{1}{x^2}))$
- Intervalle : Entre 0.001 et 1
- Code :

```
x = np.arange(1e-3,1,1e-6)
fx = np.tanh(np.cos(1/x**2))
plt.plot(x,fx)
plt.show()
```



⇒ L'utilisation de vecteur permet d'effectuer le calcul de la fonction en même temps sur toutes les valeurs du vecteur sans utiliser de boucle for.

Opérations sur les vecteurs

Opération sur l'ensemble d'un vecteur

- `np.sum` , `np.prod`, `np.mean`, `np.std`,`np.var` ...
- `np.min`, `np.max`, `np.argmax`, `np.argmin`,...
- `np.sort`, `np.argsort`, `np.count_nonzero`, `np.unique` ...

Opération booléenne sur les vecteurs

- Test d'égalité : `x == 1`
- Test d'inégalité : `x > 1`
- Comparaison de vecteur de même taille : `x==y`
- Sélection des valeurs d'un vecteur selon une valeur de vérité : `x[x>2]`
- Sélection des valeurs d'un vecteur selon un vecteur d'indice `i` : `x[i]`

Exemple d'utilisation

Soit `x` un vecteur contenant les notes d'examen de la promo.

- Compter le nombre d'étudiant au dessus de la moyenne de la classe :
`np.count_nonzero(x>np.mean(x))`

Les matrices

Les matrices en informatique

Une matrices est un tableau de nombre en 2D. On peut par exemple les utilisés pour stocker des images.

Création d'une matrice en numpy

- Matrice m de 10x5 zéros : $m = \text{np.zeros}((10,5))$
- Matrice m de 10x5 un : $m = \text{np.ones}((10,5))$
- Matrice identité m de taille 11 deux : $m = \text{np.eye}(11)$
- Matrice m de valeur $\begin{bmatrix} 31 & 29 \\ 16 & -1 \end{bmatrix}$: $m = \text{np.array}([[31,29],[16,-1]])$

Attention aux vecteurs

Attention une matrice a une seule dimension est différent d'un objet vecteur sous numpy.

Convertir une matrice en vecteur : $m[:,0]$ ou $m[0,:]$

Convertir un vecteur en matrice : $v[:,\text{np.newaxis}]$ ou $v[\text{np.newaxis},:]$

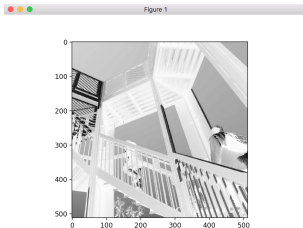
Une image vu comme une matrice

Lecture d'une image

- Utilisation de la librairie scipy :
- `import scipy as sc`
`import scipy.ndimage`
`m = sc.ndimage.imread('mon_image.png')`

Affichage d'une matrice comme une image noir et blanc

```
plt.imshow(m, cmap='gray')  
plt.show()
```



Opérations sur les matrices

Dimension d'une matrice `v`

`m.shape`

Retourne une liste de deux éléments, contenant les dimensions de la matrice `m`.

Opération entre un nombre et une matrice

- Additionner 2 à toutes les dimensions de la matrice : $m2 = m + 2$
- Multiplier 2 à toutes les dimensions de la matrice : $m2 = 2 * m$
- Diviser 2 à toutes les dimensions de la matrice : $m2 = m / 2$

Fonction de base appliquer à toutes les valeurs d'une matrice

Liste des fonctions disponibles : <https://docs.scipy.org/doc/numpy/reference/routines.math.html>

- `np.sin`, `np.cos`, `np.tan`, `np.arcsin` ...
- `np.sinh`, `np.cosh`, `np.tanh`, `np.arcsinh` ...
- `np.exp`, `np.log`, `np.log10`, ...
- `np.sqrt`, `np.fabs`, `np.sign`, `np.square`...

Exemple : `m2 = np.cos(m)`

Opérations sur les matrices

Opération sur l'ensemble d'une matrice

- `np.sum` , `np.prod`, `np.mean`, `np.std`,`np.var` ...
- `np.min`, `np.max`, `np.argmax`, `np.argmin`,...
- `np.sort`, `np.argsort`, `np.count_nonzero`, `np.unique` ...

Opération sur un axe donnée

Exemple : `np.min(m,axis=0)`

Opération booléenne sur les vecteurs

- Test d'égalité : `x == 1`
- Test d'inégalité : `x > 1`
- Comparaison de matrice de même taille : `x==y`
- Sélection des valeurs d'une matrice selon une valeur de vérité : `x[x>2]`
- Sélection des valeurs d'un axe selon un vecteur de boolean : `x[v!=0, :]` ;
- Sélection des valeurs d'une matrice uniquement sur certaine ligne `x[v, :]`.

Exemple d'utilisation

Binarisation d'une image noir et blanc.

On souhaite avoir une image où :

- Tout les pixels > 128 sont à 255,
- Tout les pixels ≤ 128 sont à 0,

Code

```
im[im>128] = 255
```

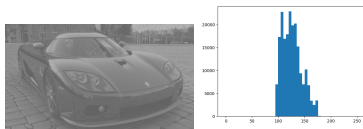
```
im[im<=128] = 0
```

Résultat Avant/Après



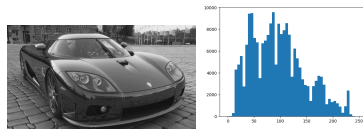
Exemple d'utilisation : Étirement d'histogramme

Avant



$\text{np.min}(x) \rightarrow 93$ $\text{np.max}(x) \rightarrow 185$

Après



$\text{np.min}(x) \rightarrow 0$ $\text{np.max}(x) \rightarrow 255$

Code

```
x2=255*(x-np.min(x))/(np.max(x)-np.min(x))
```

Les tenseurs

Les tenseurs en informatique

Un tenseur peut être vu comme un tableau de nombre en n-dimension. On peut par exemple les utiliser pour stocker des images couleurs ou pour stocker une base d'image de même dimension.

Création d'une matrice en numpy

- Tenseur t de 10x5x3 zéros : `t = np.zeros((10,5,3))`
- Tenseur s de 10x5x3x9 uns : `m = np.ones((10,5,3,9))`
- Tenseur t composé des valeurs $\begin{bmatrix} 31 & 29 \\ 16 & -1 \end{bmatrix}$, $\begin{bmatrix} 13 & 92 \\ 61 & 1 \end{bmatrix}$:
`m = np.array([[[[31,29],[16,-1]], [[13,92],[61,1]]])`

Attention aux dimensions

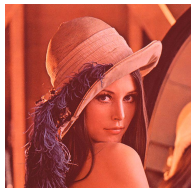
Un tenseur de dimension (10,5,2) se comportera différemment du tenseur de dimension (10,5,2,1).

Exemple d'utilisation : Augmenter le rouge d'une image

Avant



Après



Code

```
import numpy as np
import scipy as sc
import scipy.ndimage
im=np.array(sc.ndimage.imread('lena.jpg'),dtype='uint16') # lecture de l'image
im[:, :,0] += 50 # rougir
im[im>255] = 255 # seuillage
```