**Conférence Numpy :**

Début du séminaire : 12h30

**Objectif** **:** donner des infos par rapport à une librairie mathématique python : *Numpy*.

Cette libraire sert pour tout ce qui est calcul algébrique (matrice, vecteurs,…). Tous les exemples expliqué lors du séminaires sont sur l’uv dans les annonces. Lien : <https://github.com/odebeir/numpy_tutorialBA1>

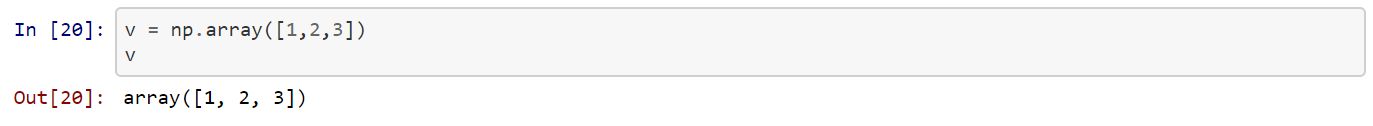
**Par rapport au tuto 1 numpy :**

<https://github.com/odebeir/numpy_tutorialBA1/blob/master/tuto_1_numpy_algebra.ipynb>

Lorsque que l’on importe une librairie, on peut donner un alias à cette librairie. Ex :

ATTENTION : importer toute la librairie avec \* est déconseillé car il encombre l’espace de travail disponible.

Pour utiliser une fct d’une librairie, on y fait appel en écrivant np.fct\_voulue.

Une des premières choses que propose la bibliothèque Numpy est la création de vecteur avec la fonction array :

v est donc un vecteur sous forme matriciel. On peut demander la taille de notre vecteur en faisant appel à la fonction v.shape() ou encore le nombre de dimension de cette matrice avec v.ndim()

On peut bien évidement comparer, additionner, multiplier des vecteurs avec les opérateur connu de python

!! on ne peut comparer que des vecteurs de tailles similaire (3,1,3) et (3,2,1) mais pas (3,1) et (3,2,4) !!

Numpy nous donne donc accès à une « toolbox » qui permet d’effectuer toute les opérations que l’on connais déjà, mais sur des vecteurs et des matrices.

Il est possible de transformer les composantes scalaires d’un vecteur en composantes booléennes de cette manière :

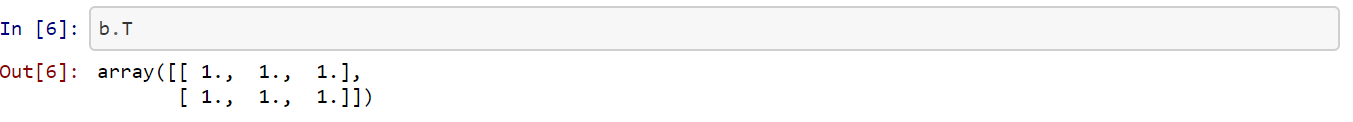
Mais Numpy permet aussi la création de matrices :

On peut initialiser une matrice nulle tel que :

Où la première composante du tuple est le nombre de lignes et la seconde le nombre de colonnes.

On peut aussi initialiser une matrice de 1 tel que :

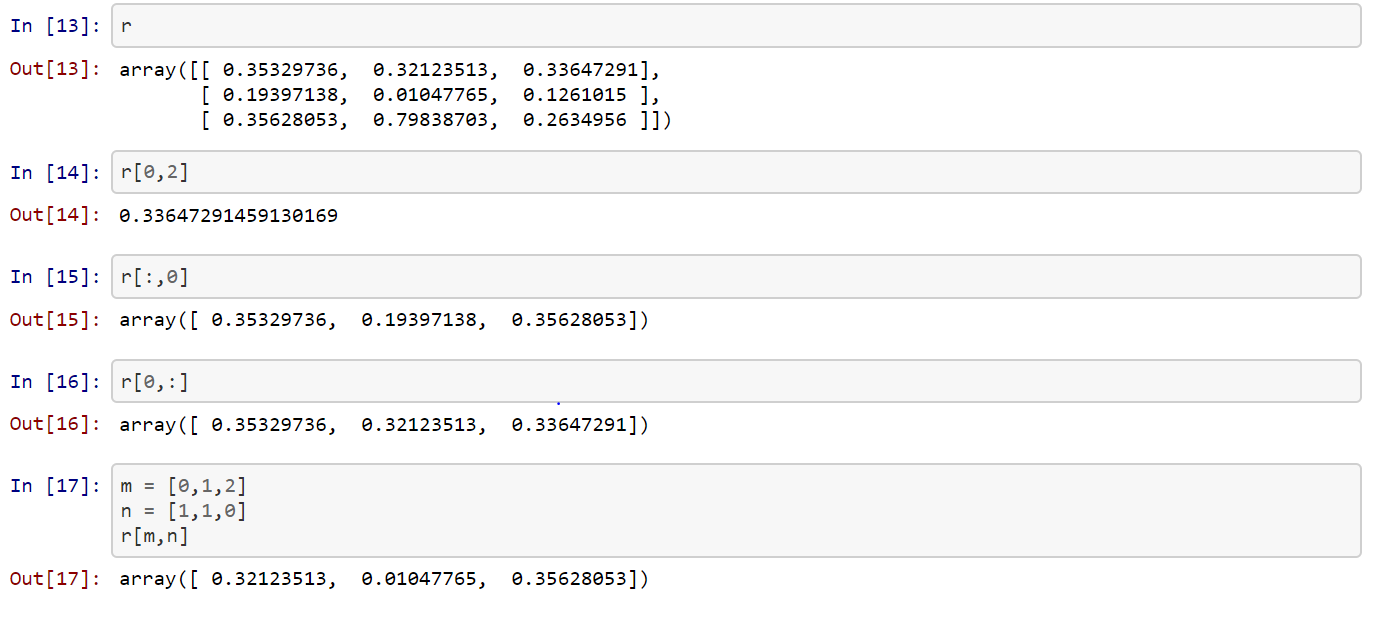
Pour créer une matrice quelconque, il faut utiliser les liste comme suit : 

Il est aussi possible de transposer une matrice avec la fonction T : 

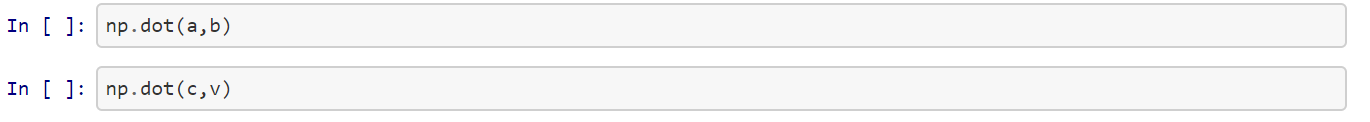
Sauvegarder une matrice dans un fichier texte : 

Où ‘random\_mat.csv’ est le nom que l’on donne à notre fichier, r est le nom de la matrice, delimiter= est la chaine de caractère qui sépare chaque élément de la matrice et fmt est la précision que l’on désire garder dans notre fichier.

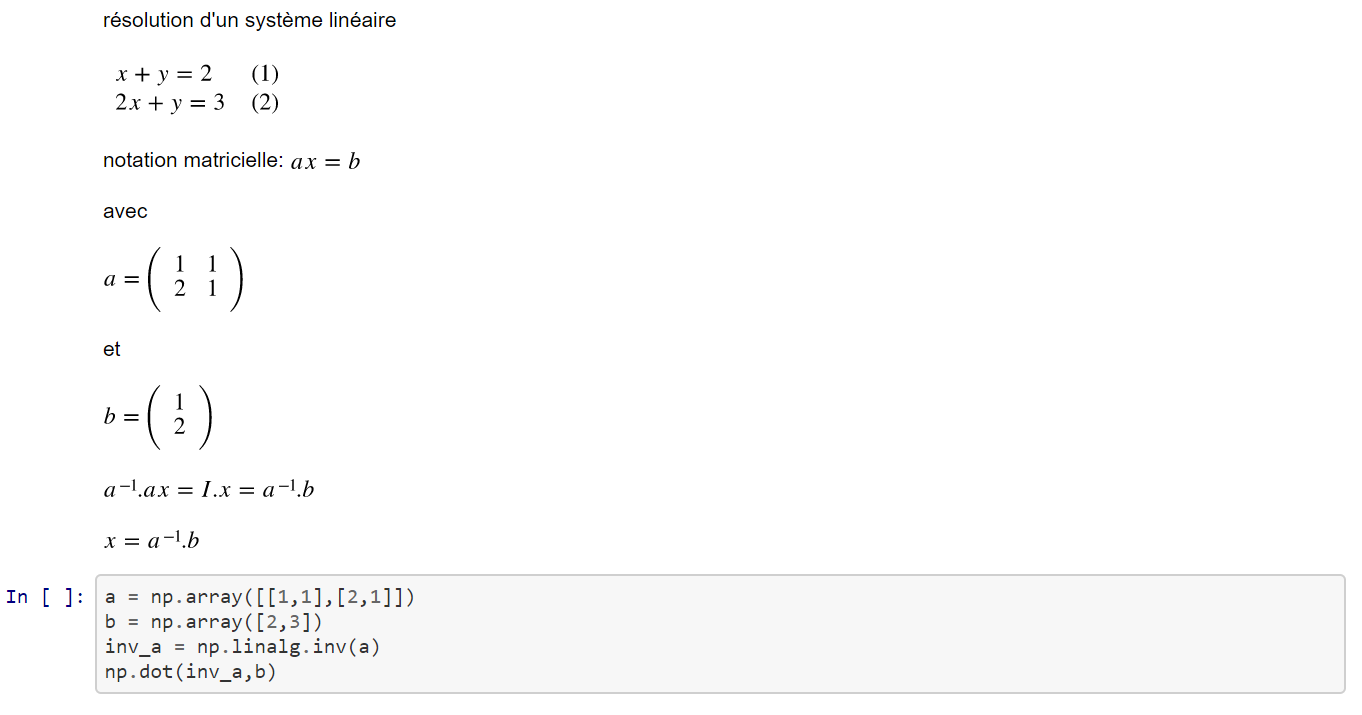
!! sauvegarder dans des fichier peut parfois faire perdre de la précision !!

Il est aussi possible de choisir de prendre des éléments d’une matrice ou d’un vecteur avec les slicing de la même manière qu’avec les liste : 

!! l’utilisation de \* ne fait pas la multiplication matricielle mais multiplie les termes en même places !!

Pour faire le produit scalaire, on utilise la fct dot() : 

On peut déterminer le déterminant d’une matrice avec np.linalg.det(r) et il est aussi possible d’inverser une matrice avec np.linalg.inv(r)

Exemple de résolution d’un système d’équations linéaire :

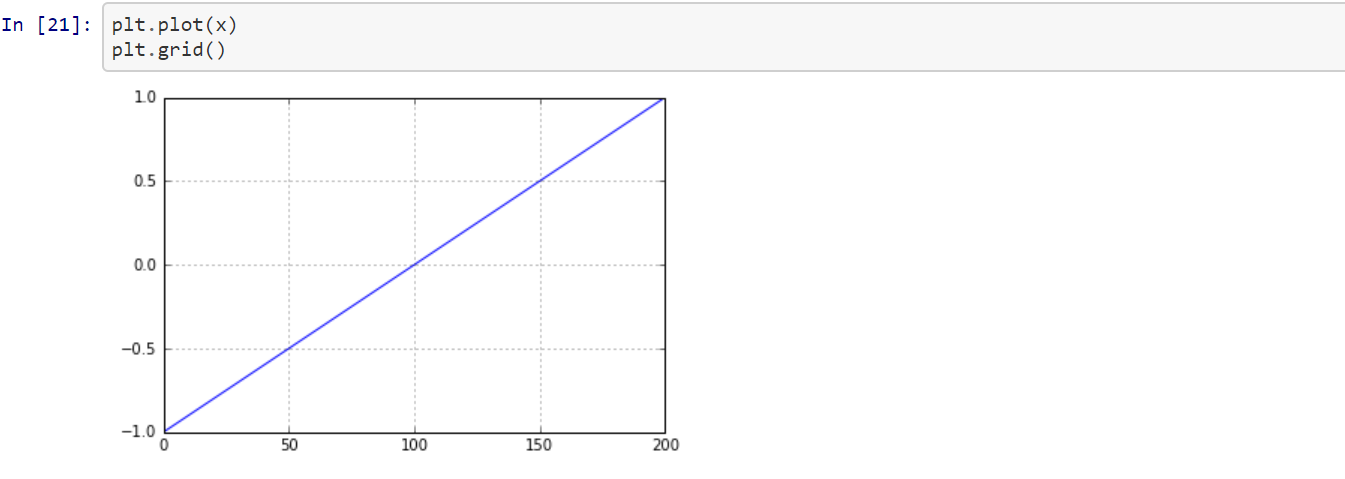
**Par rapport au tuto 2 numpy :**

<https://github.com/odebeir/numpy_tutorialBA1/blob/master/tuto_2_numpy_plot.ipynb>

On utilise la bibliothèque mathplotlib.pyplot as plt pour créer des graphiques.

arange est une fonction de numpy qui est la même que range mais qui sert pour les vecteur : 

Ici on prend tous les vecteurs de composantes x et y allant de -1 à 1 par pas de 0.01.

La fonction plot permet d’afficher les axes x et y et la fonction grid de faire une grille afin de faciliter la lecture des valeurs : 

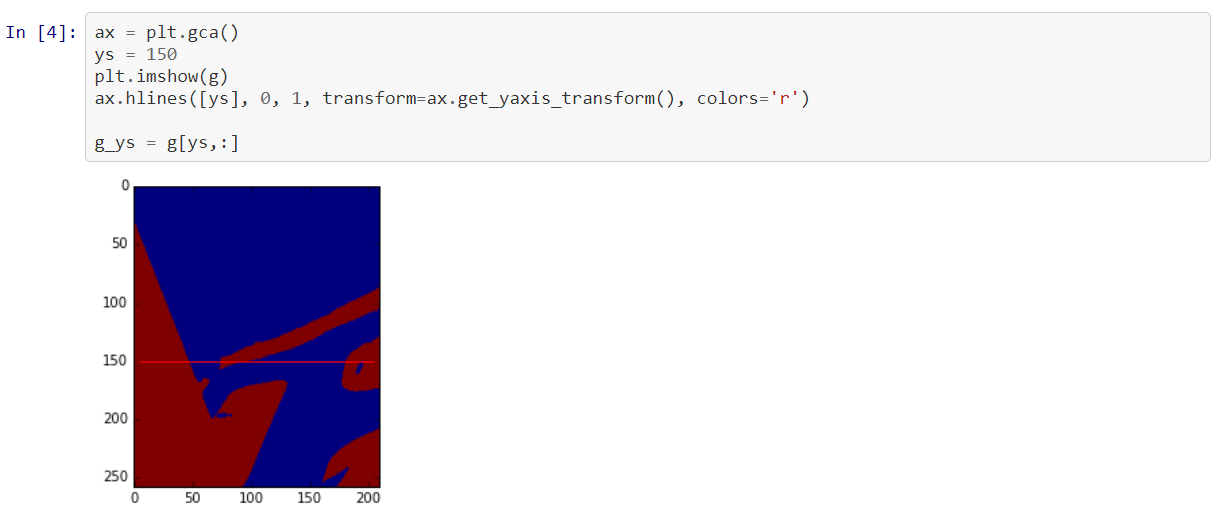
Un exemple d’application de cette partie de Numpy est la détermination d’un champ électrique de vecteurs en physique (voir dans le tuto lien au-dessus car cette partie est assez longue et ne nous servira pas lors de notre projet)

**Par rapport au tuto 3 numpy :**

<https://github.com/odebeir/numpy_tutorialBA1/blob/master/tuto_3_numpy_image.ipynb>

L’analyse d’image est une partie importante dans notre projet. Il faut bien comprendre qu’une image est lue par l’ordinateur comme une matrice. Si l’image est en nuance de gris, chaque pixel sera un point de cette matrice variant de 0 pour les parties noir a 1 pour les parties blanches. C’est donc grâce au matrice que l’on va pouvoir déterminer quelles parties du tracé est la route.

La fonction plt.imread permet de lire une image en niveaux de gris et il est possible d’afficher cette image grâce à la fonction plt.imshow() : 

On peut, comme pour les liste et comme expliquer avant pour les matrice sélectionner une partie d’image en utilisant le slicing de la matrice qui la compose : 

Où la ligne rouge clair représente la partie choisie de la matrice qui représente l’image.

NB : ici le rouge représente le blanc et le bleu le noir

Il nous est donné dans les codes de la machine virtuelle une technique d’interpolation 2D pour trouver le chemin qui parait le plus court. Cette focntion est appelée profile. La fonction profile prend en paramètre une image et échantillonne le long d’une droite. Cet échantillonnage donne la valeur des niveaux de gris le long de ce vecteur qui traverse l’image. On utilise encore une autre librairie appelée scipy pour avoir la fonction d’interpolation.

La route idéale est donc définie par les 0 contenu dans la matrice image. Pour connaitre la position de toutes les valeurs d’une matrice égale à 0 on utilise la fonction np.where(r>0) : 

**Tuto 4 : comment les images couleurs sont codées ? :**

Pour les images en couleur, pour chacun des pixels, il faut trois valeurs , une pour le rouge une pour le vert et une pour le bleu. Pour cela, il faut une matrice a trois dimensions et donc une matrice cubique 3x3x3 où chaque ‘plan’ de la matrice représente l’intensité d’une des trois couleur du rgb.

Le triplet des trois intensité donne la couleur réel de l’image.

Histogramme d’une image : graphique qui reprend pour chacune des valeur des couleurs le nombre de pixels qui ont cette valeur et donc cette couleur.

Typiquement, pour chaque panneau, l’orange pi va recevoir des petite matrice qui les représentent.

Dans notre librairie utilisée par le projet, par convention, les plan de couleur sont dans l’ordre suivant : bleu, vert et rouge.

Lorsque l’on voit un panneau, il faut donc voir si c’est le bleu qui prédomine pour les flèches ou le rouge pour les stop.

