Outils de Calcul Scientifique

Objectifs

- Basés sur Python.
- Création de données formatées.
- Utilisation de fonctions prédéfinies et optimisation des calculs.
- Représentation des données/résultats.

Bibliothèques

- `Numpy` : gestion des données.
- `Scipy, Biopython, rpy` : calculs
- `Matplotlib` : graphes, histogrammes, camembert .... plutôt 2D, la 3D est assurée par `Mayavi`.
- `Pylab` permet avec un seul `import` de disposer de `numpy` et de `matplotlib` et d'interagir avec les fenêtres `matplotlib`.

Les Structures Numpy

- Utilisation des vecteurs et des opérations vectorielles :
  - Création d'un vecteur :

```
x = zeros(10)
x = 3*ones(10)
x = arange(10,20)
x = linspace(0,0,1.0,10)
```

  - Création d'une matrice :

```
M = zeros((3,4))  lignes, colonnes
M = eye(n,p)  matrice taille n,p et des 1 sur
la diagonale. M = diag(v)
```

  - Accès :

```
elem = M[0,1]  ligne, colonne
```

Opérations Matricielles

- `Numpy` optimise tous les calculs matriciels.
- Plus performant que des boucles *“maison”*
- Somme de 2 matrices `M = M1 + M2` terme à terme.
- Multiplication terme à terme `M = M1 * M2`.
- Véritable multiplication de matrices `M = dot(M1,M2)`
- Produit scalaire de 2 vecteurs `vdot(v1,v2)`

Petits exercices :

- `4 * ones((4,4)) * diag([2,3,4,5])`
- `dot(4 * ones((4,4)), diag([2,3,4,5]))`

Tests Booléens

- `Numpy` étend tous les tests booléens aux vecteurs/matrices.
- Retourne une matrice de booléens de la taille du vecteur/matrice testés.

```
b=6*np.eye(4,4)-np.diag(4+np.arange(3),1)
                      +2*np.ones((4,4))
a=np.arange(16).reshape(4,4)
(a<b)
(a==b)
(a<2)
(2<b)
```

Matplotlib

- Documentation :
  scipy-lectures.github.io/intro/matplotlib/matplotlib.html
  http://jakevdp.github.io/mpl_tutorial/
- `Pylab` permet d'inporter `numpy` et `matplotlib`
- Modèle objet :
  - `x = arange(0,10,0.2)`
  - `y = sin(x)`
  - `fig = figure()`
  - `item = fig.add_subplot(111)`
  - `item.plot(x,y)`
  - `show()`
- Exercices

Résoudre une équation différentielle

- Résolution numérique : Euler, méthode itérative
- Exemple : résoudre $\frac{dy}{dx} = -2xy$ avec
  - $y_{(x=0)} = 1$
- Solution analytique : $y = exp(-x^2)$
- Exercice : remplir le tableau pour $x = 0, 1, 2, 3, 4, 5$
- Dessiner les courbes
  - de la fonction analytique
  - des valeurs obtenues par Euler

Résoudre une équation différentielle

- Soit une équation de la forme $\frac{dy}{dt} = -2y$ entre $t = 0..10$ avec $y(t_0) = 1$

```
from scipy.integrate import odeint
def rhs (y,t) :
return -2y

t=np.linspace(0,10,100)
y= odeint(rhs,1,t)
```

- Dessiner le Résultat : `plot (t,y)`

Système à deux équations

- Lokta-Volterra (cf cookbook SciPy)
- u proies et v prédateurs
-

$$du/dt = a * u - b * u * v$$

$$dv/dt = -c * v + d * b * u * v$$

- Ecrire le modèle