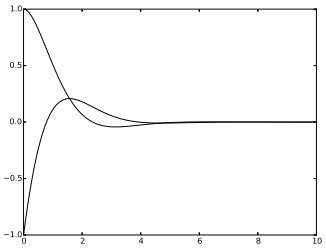
2TSI. Correction TD PYTHON 14

Exercice 01 : résolution avec *odeint*

Résoudre avec *odeint* le système différentiel $\left\{\begin{array}{rcl} x'(t) &=& -x(t)-y(t) \\ y'(t) &=& x(t)-y(t) \end{array}\right.$ sur $[0,10]$ de pas 0.01 avec $x(0)=1$ et $y(0)=-1$. On affichera ensemble $t\mapsto x(t)$ et $t\mapsto y(t)$.

Solution

```
>>> import numpy as np; import scipy.integrate as integr; import matplotlib.pyplot as plt
>>> def F(x,t) : return(np.array([-x[0]-x[1], x[0]-x[1]]))
>>> T = np.arange(0,10,0.01); X = integr.odeint(F,np.array([1,-1]),T)
>>> plt.plot(T,X[:,0]); plt.plot(T,X[:,1]); plt.show()
```



Exercice 02 : algorithme d'Euler

Considérons le système différentiel du premier ordre $\left\{\begin{array}{rcl} x'(t) &=& f(x(t),y(t),t) \\ y'(t) &=& g(x(t),y(t),t) \end{array}\right.$, où f et g sont des fonctions de $\mathbf{R}^3$ dans $\mathbf{R}$. Les conditions initiales sont $x(t0)=x0$ et $y(t0)=y0$.

1. Adapter la fonction *Euler_Affich* du TD 13 en une fonction *EulerSyst_Affich* qui a pour arguments f, g, $t0$, tn, n et $x0$, $y0$ et qui affiche la courbe intégrale $t\mapsto(x(t),y(t))$ pour $t\in[t0,tn]$ avec un pas $h=(tn-t0)/n$.

   Solution

   ```
   >>> import matplotlib.pyplot as plt
   >>> def EulerSyst_Affich(f,g,t0,tn,n,x0,y0) :
           t = t0; x = x0; y = y0; h = (tn-t0)/float(n); T = [t0]; X = [x0]; Y = [y0]
           for k in range(n) :
               x, y = x + h*f(x,y,t), y + h*g(x,y,t); t = t + h
               T.append(t); X.append(x); Y.append(y)
           plt.plot(X,Y)
   ```

2. Appliquer à $\left\{\begin{array}{rcl} x'(t) &=& x(t)(1-y(t)) \\ y'(t) &=& y(t)(x(t)-1) \end{array}\right.$ avec $[t0,tn]=[0,10]$, $x(0)=2$, $y(0)=1$ et $n=500$.

   Solution

   ```
   >>> def f(x,y,t) : return(x*(1-y)); def g(x,y,t) : return(y*(x-1))
   >>> EulerSyst_Affich(f,g,0,10,500,2,1)
   ```