TP 27- Proposition de solutions

Solution 1 Fonction donnant l'approximation de $\exp(x)$ par la méthode d'Euler avec un pas de h :

```
def approx_exp(x,h):
    if x<0: h=-h
    y=1
    t=0
    while (x-t)/h>1:
        y=y+h*y
        t=t+h
    return(y)
```

Cela donne :

```
--> approx_exp(2,1e-5)
7.388908320148598
# la valeur machine est : 7.3890560989306504
```

Solution 2 Approximation de $\exp(1)$:

x	0	0,1	0.2	0,3	0,4
$y(x)$	1	1,1	1,21	1,331	1,464
$y'(x)$	1	1,1	1,21	1,331	1,464

0,5	0,6	0,7	0,8	0,9	1
1,611	1,772	1,949	2,144	2,358	2,594
1,611	1,772	1,949	2,144	2,358	2,594

L'approximation trouvé est $\exp(1) \approx 2.594$.

Solution 3 Pour calculer $\ln(x)$ il convient de poursuivre le calcul de $\exp(t)$ jusqu'à atteindre x et de retourner t.

```
def approx_ln(x,h):
    if x<1: h=-h
    y=1
    t=0
    while (x-y)*h>0:
        y=y+h*y
        t=t+h
    return(t)
```

- Le pas est positif si $x \geq 1$ (on avance vers la droite de la courbe par rapport à la condition initiale $(0,1)$) et négatif sinon.
- Le test de continuité est lorsque y n'a pas encore dépassé x. Il y a donc deux cas à considérer :

$$\big(h > 0 \text{ et } y < x\big) \text{ ou } \big(h < 0 \text{ et } x < y\big)$$

qui se regroupe en $(x-y)h > 0$.

Cela donne :

```
--> approx_ln(2,1e-5)
0.6931599999994802
# la valeur machine est : 0.69314718055994529
```

Solution 4 **Équation différentielle d'ordre 2**

1. cos vérifie l'équation : $y'' + y = 0$ avec $y(0) = 1$ et $y'(0) = 0$
2. Dérivons le vecteur de fonctions $Z = \begin{pmatrix} y \\ y' \end{pmatrix}$:

$$Z' = \begin{pmatrix} y' \\ y'' \end{pmatrix} = \begin{pmatrix} y' \\ -y \end{pmatrix}$$

On note que Z' dépend uniquement de Z (et de t). On peut écrire $Z' = f(Z,t)$ avec

$$f : \left(\begin{pmatrix} u \\ v \end{pmatrix}, t\right) \longmapsto \begin{pmatrix} b \\ -u \end{pmatrix}$$

3. On s'intéresse au double de l'abscisse du premier point ou y s'annule dans le cas où $y = \cos$.
Le schéma récurrent d'Euler devient en notant dy_k l'approximation de $y'(t_k)$:

$$\begin{cases} t_{k+1} &= t_k + h \\ y_{k+1} &= y_k + h\,dy_k \\ dy_{k+1} &= dy_k - h y_k \end{cases}$$

Ce qui donne :

```
def approx_pi(h):
    t,y,dy=0,1,0
    while y>0:
        t,y,dy=t+h,y+h*dy,dy-h*y
    return(2*t)
```

Cela donne :

```
--> approx_pi(1e-5)
3.1416000000036464
# la valeur machine est : 3.141592653589793
```


Lycée Saint-Exupery – maffre.saintexupery@gmail.com – MPSI

Solution 5 Méthode d'Euler

1. Le script de la fonction :

Méthode Euler

```
def euler(f,y0,a,b,h):
    if b<a: h=-h
    t=np.arange(a,b,h)
    n=len(t)
    T=np.zeros((n,len(y0)))
    T[0,:]=y0
    for i in range(1,n):
        T[i,:]=T[i-1,:]+h*f(T[i-1,:],t[i-1])
    return(t,T)
```

Pour tracer la courbe de la solution il faut de considérer les points d'abscisses `t` et d'ordonnées `T[:,0]` : la première colonne de `T`.

Remarque : Si le paramètre d'entré de la méthode d'Euler, n'est pas le pas mais plutôt le nombre d'itération `n`, alors il suffit d'adapter le script avec :

```
t=np.linspace(a,b,n)
h=(b-a)/(n-1)
```

2. Cas du ressort avec frottement :

a) Il faut écrire vectoriellement l'équation différentielle afin de se ramener à de l'ordre 1 :

$$y(t)=\begin{pmatrix}x(t)\\x'(t)\end{pmatrix} \quad \text{et} \quad y'(t)=\begin{pmatrix}x'(t)\\-\frac{1}{m}\left(cx'(t)+kx(t)\right)\end{pmatrix}=f(y(t),t)$$

La fonction f est : La fonction f est :

$$f:\left(\begin{pmatrix}u\\v\end{pmatrix},t\right)\mapsto\begin{pmatrix}v\\-\frac{k}{m}u-\frac{c}{m}v\end{pmatrix}$$

b) Instructions PYTHON:

```
k,m,c=10,2,1

def f(Y,t):
    return np.array([Y[1],-k/m*Y[0]-c/m*Y[1]])

a,b,h=0,10,1e-4
y0=np.array([2,0])
t,T=euler(f,y0,a,b,h)
```

c) Le comportement du ressort : $t\mapsto(t,x(t))$

```
plt.xlim(-0.5,10)
plt.ylim(-2.2,2.2)
plt.plot(t,T[:,0],color='blue')
```

- Le diagramme de phase : $t\mapsto(x(t),x'(t))$

```
plt.xlim(-2.2,2.2)
plt.ylim(-5,5)
plt.plot(T[:,0],T[:,1],color='blue')
```