

DS Informatique n°3 – Partie machine. Durée : 1 heure 30

- Faire tous les exercices dans un même fichier `NomPrenom.py` que vous sauvegarderez,
- en début de chaque exercice mettez, dans une ligne de commentaire, l'exercice que vous traitez,
- n'oubliez pas de commenter votre code,
- il est possible de demander un déblocage pour une question, mais celle-ci sera notée 0.

Ouverture et traitement des données

Le relevé de la figure 1 a été obtenu en mesurant l'accélération d'un ascenseur lors de sa montée, les données correspondantes sont stockées dans le fichier `data.csv` sous la forme `temps;acceleration`, (temps en s et accélération en m.s^{-2}) :

```
0.00;-4.7008197688875945e-06
0.01;-1.6121153364425545e-06
```

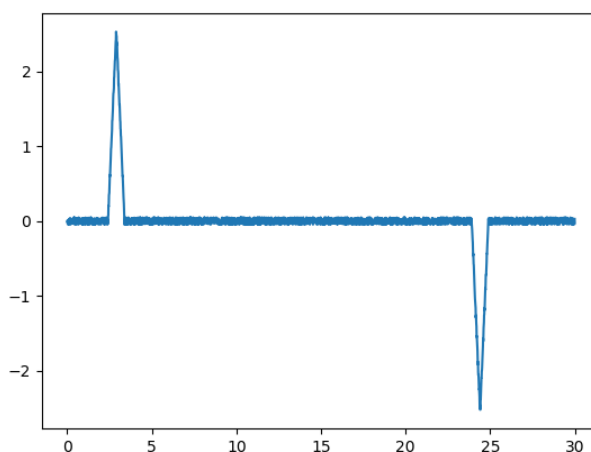


FIGURE 1 – Accélération (m.s^{-2}) de l'ascenseur au cours du temps (s)

- 1) (a) Ouvrir le fichier, traiter les données pour les stocker dans des listes `temps` et `acceleration`. Tracer `acceleration` en fonction de `temps`, cela devrait donner le même tracé que celui de la figure 1.

Solution.

```
fichier=open('data.csv','r')
contenu=fichier.read()
lignes=contenu.split('\n')
temps=[]
acceleration=[]

for ligne in lignes[1:-1]:
```

```
temps.append(float(ligne.split(";")[0]))
acceleration.append(float(ligne.split(";")[1]))
```

```
plt.plot(temps,acceleration)
```

- (b) Soit f une liste des valeurs d'une fonction à des instants définis dans une liste t . Définir une fonction `integration_num(t,f,f0)` qui renvoie une liste des différentes valeurs de l'intégrale de la fonction à partir de la méthode des rectangles. $f0$ est la valeur à l'origine de l'intégrale.

Solution.

```
def integration_num(t,f,f0):
    fi=[f0]
    for i in range(1,len(f)):
        fi.append(fi[i-1]+f[i]*(t[i]-t[i-1]))
    return fi
```

- (c) Utiliser cette fonction pour déterminer la vitesse puis la position de l'ascenseur au cours du temps sous la forme de listes `vitesse` et `position`. On prendra les conditions initiales nulles ($vitesse(0) = 0$ et $position(0) = 0$).

Solution.

```
vitesse=integration_num(temps,acceleration,0)
position=integration_num(temps,vitesse,0)
```

- (d) Tracer `position` en fonction de `temps`.

Solution.

```
plt.plot(temps,position)
plt.show()
```

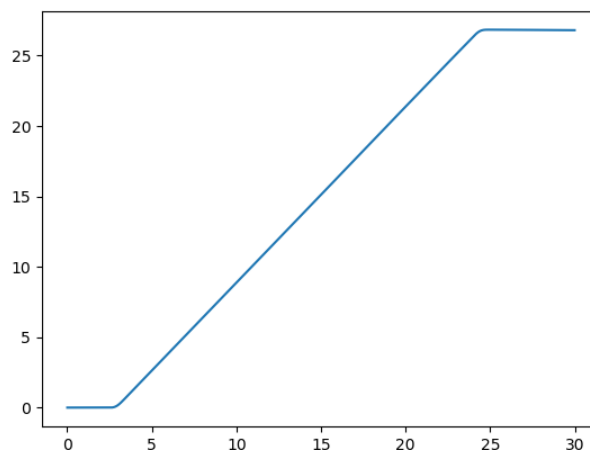


FIGURE 2 – Position (m) de l'ascenseur au cours du temps (s)

- (e) Un étage mesurant 3 m, afficher le nombre d'étages parcourus.

Solution.

```
print(position[-1]/3.)
```

On trouve 9 étages (valeur approchée à cause des erreurs de mesure.)

Fabrication d'une fonction à partir d'une liste de coordonnées

Pour la suite de l'exercice, il est nécessaire que l'accélération soit stockée sous la forme d'une liste de coordonnées afin d'être manipulée et consultée comme une fonction $a(t)$ dont l'entrée serait un instant t et la sortie l'accélération correspondante (l'exemple donné présente le mode d'utilisation de la fonction mais les valeurs ne correspondent pas à ce qui doit être trouvé) :

```
print a(10)
>>> 8.999
```

La réalisation de cette fonction doit se faire en deux étapes :

- déterminer à quel index de la liste correspond le temps t donné en entrée,
- afficher l'accélération dont l'index est la valeur trouvée précédemment.

- 1) (a) Proposer une fonction `index_temps(t,data_temps)` qui renvoie l'index du temps juste inférieur à t dans la liste `data_temps`.

Solution.

```
def index_temps(t,data_temps):
    i=0
    while data_temps[i]<t:
        i=i+1
    return i
```

- (b) Proposer une fonction `a(t)` qui utilise la fonction `index_temps` afin de donner l'accélération associée à un instant t par la méthode précédente.

Solution.

```
def a(t):
    return acceleration[index_temps(t,temps)]
```

- (c) En sachant que $\frac{d v(t)}{dt} = a(t)$, déterminer grâce à la méthode d'Euler la liste `v_list` des coordonnées de la fonction vitesse $v(t)$.

Solution.

```
def methode_euler(F,y0,t):
    y = [0]*len(t)
    y[0] = y0
    for i in range(len(t)-1):
        y[i+1] = y[i]+(t[i+1]-t[i])*F(t[i])
    return y
```

```
v_list=methode_euler(a,0,temps)
```

- (d) Proposer une fonction `v(t)` qui utilise la fonction `index_temps` afin de donner la vitesse associée à un instant t par la méthode précédente en utilisant la liste `v_list`.

Solution.

```
def v(t):
    return v_list[index_temps(t,temps)]
```

- (e) En sachant que $\frac{d p(t)}{dt} = v(t)$, déterminer grâce à la méthode d'Euler la liste `p_list` des coordonnées de la fonction position $p(t)$.

Solution.

```
p_list=methode_euler(v,0,temps)
```

- (f) Proposer une fonction `p(t)` qui utilise la fonction `index_temps` afin de donner la position associée à un instant `t` par la méthode précédente en utilisant la liste `p_list`.

Solution.

```
def p(t):
    return p_list[index_temps(t,temps)]
```

- (g) Tracer les listes `p_list` et `position` en fonction de `temps`. Le résultat vous paraît-il satisfaisant ?

Solution.

```
plt.plot(temps,position)
plt.plot(temps,p_list)
plt.show()
```

Le résultat est satisfaisant car les courbes se superposent.

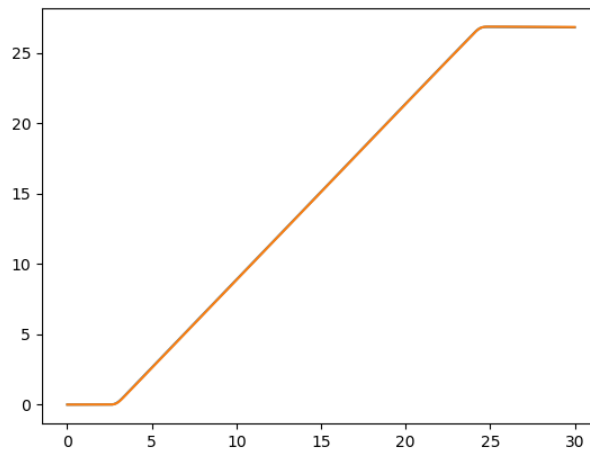


FIGURE 3 – Position (m) de l’ascenseur au cours du temps (s) par les deux méthodes

- 2) On souhaite rechercher une position particulière. A l’aide de la méthode de Newton, déterminer l’instant `t` à partir duquel l’ascenseur est à 12m du sol (4 étages).

Solution.

```
def p2(t):
    return p(t)-12.

def methode_Newton(f,df,a,p) :
    x0=a
    while abs(f(x0))>p : # ou abs(e)>p
```

```
        x1=x0-f(x0)/df(x0)
        e=x1-x0
        x0=x1
    return x1

print(methode_Newton(p2,v,5,10**(-2)))
```