C.R. TP 13 : première partie

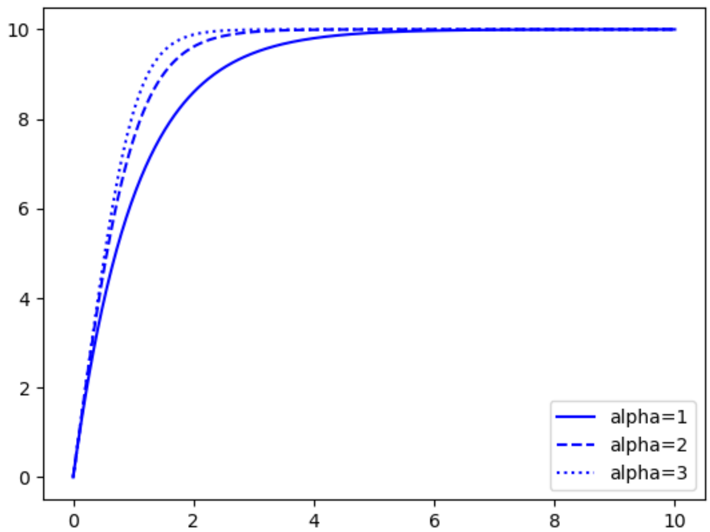
1 Vitesse de chute avec différents frottements fluides

1.1 Etude abstraite

```
import pylab as pl
from scipy import integrate
pl.figure('Vitesse de chute')
pl.clf()
v0=0
t0=0
g=9.81
style = [0,'-', '--', ':']
# lignes continues, tirets, points-tirets pour distinguer les courbes sans couleur ici
# c'est moins joli mais mieux pour les photocopies noirs et blancs
t=pl.linspace(0,10,500)
for alpha in range(1,4):
    def f(v,t):
        coefflambda= g/10**alpha
        vp=-coefflambda*v**alpha+g

        return vp
    v=integrate.odeint(f,v0,t)
    pl.plot(t,v,"b"+style[alpha],label="alpha="+str(alpha))
pl.legend()
pl.show()
```

Avec le résultat suivant :



Remarque physique : Dans un liquide, on aura plutôt $\alpha = 1$. Dans un gaz sauf si la vitesse est très faible, on aura plutôt $\alpha = 2$. Si on connaît α et la vitesse limite, on en déduit le coefficient de frottement k.

Complément plus théorique : On peut ici en fait résoudre les équations différentielles « de manière exacte » et se passer de `odeint` :

a) Pour le cas $\alpha = 1$, on a une brave E.D. linéaire :

$$v'(t) = -\lambda v(t) + g.$$

qu'on sait bien résoudre, de manière générale :

$$v(t) = C\exp(-\lambda t) + \frac{g}{\lambda}$$

où C est une constante, qu'on détermine ici avec la C.I $v(0) = 0$ ce qui donne :

$$v(t) = \frac{g}{\lambda}(1 - e^{-\lambda t})$$

On retrouve le fait que le rapport $\dfrac{g}{\lambda}$ est la vitesse limite, qu'on note v_ℓ (qu'on avait obtenu dans le sujet en prenant $\dfrac{dv}{dt} = 0$ dans l'E.D.), et on écrit la solution ;

$$\boxed{v(t) = v_\ell.(1 - e^{-\lambda t}) = v_\ell(1 - e^{-g/v_\ell.t}).}$$

b) Le cas $\alpha = 2$ demande un peu plus d'effort : L'équation différentielle devient :

$$v'(t) = -\lambda v(t)^2 + g.$$

Dans cette E.D. on veut « séparer les variables », ce que vous écrivez en physique sous la forme :

$$\frac{dv}{g - \lambda v^2} = dt. \quad (*)$$

Voyons cela au ralenti à partir de

$$\frac{dv}{dt} = \lambda v(t)^2 + g. \quad (\dagger)$$

D'abord pourquoi peut-on diviser par $-\lambda v(t)^2 + g$? Autrement dit pourquoi ce terme ne s'annule-t-il pas ?

La raison vient d'un théorème d'existence et d'unicité des solutions pour les pb. de Cauchy, qui s'applique encore à cette E.D. [1]

Si on avait un t_0 tel que $\lambda v(t_0)^2 + g = 0$, alors la solution $t \mapsto v(t)$ considérée coïnciderait au temps t_0 avec la solution constante $t \mapsto v_\ell$ (ou son opposée si la vitesse était négative), où v_ℓ est la valeur de la solution constante à l'E.D. Par thme d'unicité on aurait $v(t) = v_\ell$ pour tout t ce qui n'est pas vrai car $v(0) = 0$.

N.B. La valeur v_ℓ de la solution constante est ce qu'on a nommé vitesse-limite dans l'énoncé. « Physiquement » on s'attend à ce que la vitesse croisse dans la chute, ce qui fait grandir les frottements, jusqu'à ce qu'« à la limite » le frottement compense la force de pesanteur pour obtenir un bilan des forces nuls et une vitesse constante. *On n'utilisera pas cette intuition dans le calcul qui suit, mais on va au contraire la démontrer* : on va démontrer que $v(t) \underset{t\to+\infty}{\longrightarrow} v_\ell$.

Maintenant qu'on sait que le second membre de $(\dagger)$ ne s'annule pas, le premier membre $v'(t)$ ne s'annule pas donc $t \mapsto v(t)$ est injective et en physique on note $v \mapsto t(v)$ son application réciproque et par théorème sur la dérivée d'une réciproque : $(\dagger)$ peut se réécrire (pour tout v dans $[0, v_\ell[$) :

$$\frac{dt}{dv} = \frac{1}{g - \lambda v^2} \quad (\ddagger)$$

1. Même si les ED non linéaires ne sont pas au programme de maths, disons que ce théorème s'applique à beaucoup d'E.D. même non linéaires : celles pour lesquelles la fonction dans le second membre $(v,t) \mapsto F(v,t)$, où on décorrèle les variables v et t, est localement lipschitzienne par rapport à la variable v : ce théorème est dû à Cauchy et Lipschitz et cette condition est la raison de l'appellation « Lipschitzienne ». Ici $F(v,t) = \lambda v^2 + g$, et $v \mapsto v^2$ est localement lipschitzienne.

on a séparé les variables... c'est ce que vous écrivez avec $(*)$.... avec des notations commodes faisant intervenir des différentielles que nous n'avons pas définies en maths, mais $(\ddagger)$ et $(*)$ sont équivalentes.

Maintenant on peut utiliser $(\ddagger)$ pour calculer $t(v)$ par simple calcul de primitive. Il est plus joli de factoriser par g au dénominateur, ce qui fait apparaître d'ailleurs la vitesse-limite v_ℓ car : $1/v_\ell^2 = \lambda/g$

$$\frac{dt}{dv} = \frac{1}{g(1-v^2/v_\ell^2)} \quad (\ddagger)$$

Par D.E.S. : $\dfrac{1}{g(1-v^2/v_\ell^2)} = \dfrac{1}{2g}\left(\dfrac{1}{1-v/v_\ell} + \dfrac{1}{1+v/v_\ell}\right)$. Donc par primitivation :

$$t(v) = \frac{v_\ell}{2g}\ln\left|\frac{1+v/v_\ell}{1-v/v_\ell}\right|$$

Ici, comme dit plus haut $v(t) \neq \pm v_\ell$ par théorème d'unicité, donc $v(t) \in]-v_\ell, v_\ell[$ pour tous les temps, ainsi on peut enlever les valeurs absolues et :

$$t(v) = \frac{v_\ell}{2g}\ln(\frac{1+v/v_\ell}{1-v/v_\ell})$$

En fait la fonction $x \in]-1,1[\mapsto \dfrac{1}{2}\ln(\dfrac{1+x}{1-x})$ n'est rien d'autre que la fonction th^{-1} :

$$t(v) = \frac{v_\ell}{2g}\,\mathrm{th}^{-1}(v/v_\ell)).$$

Et en prenant t comme variable donc en passant à la fonction réciproque :

$$\boxed{v(t) = v_\ell\,\mathrm{th}(\frac{g}{v_\ell}t)}$$

Si on trace le graphe, il coïncidera avec celui donné par `odeint`.

Remarque : on prouve ainsi que v_ℓ est bien la vitesse-limite comme attendu par le physicien (voir le N.B. plus haut) et utilisée dans l'énoncé comme telle. Mais le raisonnement que nous venons de faire pour la résolution n'a pas eu besoin de cette intuition physique donnée par l'expérience a priori, on a seulement utilisé le théorème d'unicité des solutions au pb. de Cauchy considéré et v_ℓ désignait simplement la valeur de la solution constante (positive) à l'E.D.

c) Comparaison mathématique des solutions trouvées au a) et au b)

En notant $\tau = v_\ell/g$ (homogène à un temps), les deux solutions s'écrivent respectivement

$$\begin{cases} v(t) = v_\ell(1-e^{-t/\tau}) & (a), \\ v(t) = v_\ell\,\mathrm{th}(t/\tau) & (b) \end{cases}$$

La variable τ est parfois appelée *temps caractéristique*. Ces deux fonctions croissent assez vite pour qu'au bout d'un temps de l'ordre de quelques τ, on soit déjà proche de v_ℓ : pour la première avec $t = \tau$ on est a 63%, avec $t = 2\tau$, on est à 86%.

La position relative des courbes peut aussi se démontrer à partir de ces formules explicites : pour $u = t/\tau$, on a :

$$\mathrm{th}(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}} = \frac{1-e^{-2u}}{1+e^{2u}}$$

On veut montrer que $\forall u \in \mathbb{R}^+, \mathrm{th}(u) \geq 1 - e^{-u} \quad (C)$. Or en posant $x = e^{-u}$, on a $x \in]0,1]$ et :

$$(C) \Leftrightarrow \frac{1-x^2}{1+x^2} \geq 1 - x \Leftrightarrow \frac{1+x}{1+x^2} \geq 1 \Leftrightarrow x \geq x^2,$$

ce qui est vrai puisque $x \in]0,1]$.

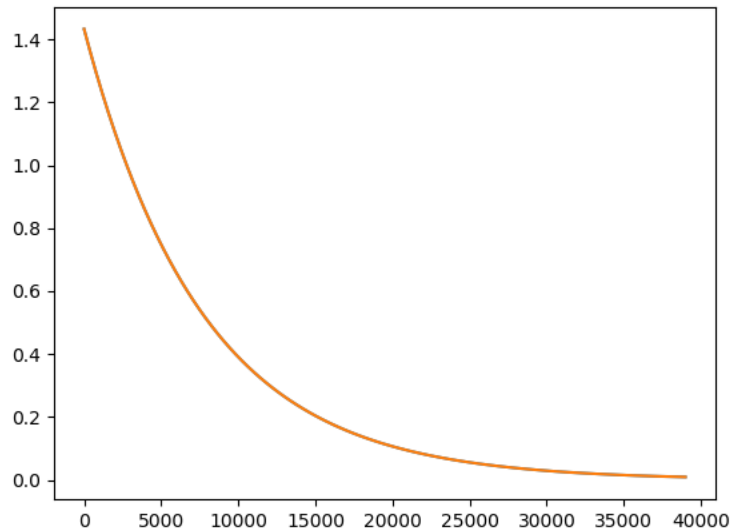
1.2 Le saut de Felix Baumgartner

a) Comme on choisit l'axe (Oz) orienté vers le bas, $v = -\dfrac{dh}{dt}$. De même le poids $\overrightarrow{P} = \dfrac{GMm}{(R+h)^2}\vec{e}_z$ alors que la force de frottement est $\overrightarrow{F} = -\dfrac{1}{2}\rho(h)AC_x v^2\vec{e}_z$.
La relation fondamentale de la dynamique s'écrit alors, en projection sur $\vec{e}_z$:

$$m\frac{dv}{dt} = -\frac{1}{2}\rho(h)AC_x v^2 + \frac{GMm}{(R+h)^2}.$$

b) Loi de ρ :

```
h=np.linspace(0,39000,100)
pl.figure("Loi de rho");pl.clf;
pl.plot(h,1.433*pl.exp(-0.00013*h))
pl.show()
```



On a une jolie décroissance exponentielle de la densité avec l'altitude, on voit que sur cette plage d'altitude, ρ ne doit *vraiment pas* être considérée comme constante.

c) . L'E.D. du second ordre donnée par la R.F.D. devient une E.D. du premier ordre pour l'inconnue vectorielle $X(t) = \begin{pmatrix} v(t) \\ h(t) \end{pmatrix}$. Précisément : sachant que $X'(t) = \begin{pmatrix} v'(t) \\ h'(t) \end{pmatrix}$ et que $h'(t) = -v(t)$, l'E.D. devient :

$$X'(t) = F(X(t), t),$$

où pour $X = \begin{pmatrix} X_0 \\ X_1 \end{pmatrix}$, $F(X,t) = \begin{pmatrix} -\dfrac{1}{2m}\rho(X_1)AC_x X_0^2 + \dfrac{GM}{(R+X_1)^2} \\ -X_0 \end{pmatrix}$

N.B. En fait $F(X,t)$ ne dépend pas de t, mais on doit donner ce paramètre pour appliquer `odeint`.
Voici l'implémentation en Python de cette fonction `F` :

$\boxed{\text{Donner des noms pour les constantes numériques, } m,\ A \text{ etc, c'est plus lisible !}}$

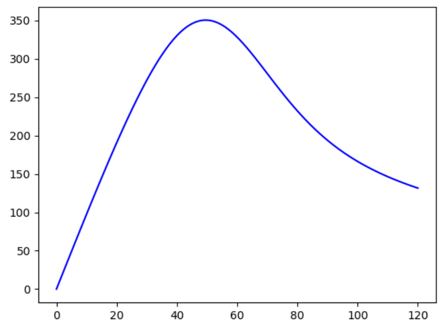
```
def F(X,t):
    m=80.;A=0.45;Cx=0.8;
    #print(X)
```

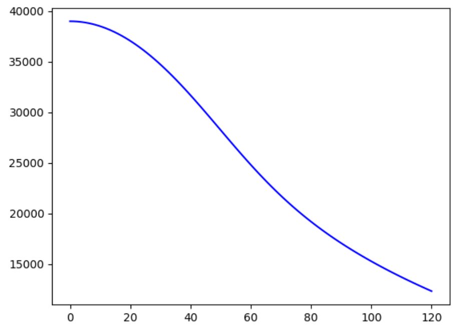
```
    v=X[0];
    h=X[1]; G=6.73e-11;M=6.e24; R=6371.e3;
    rho=1.433*pl.exp(-0.00013*h);
    dX=np.zeros(2)
    dX[0]=1/m*(-1. /2*rho*A*Cx*v**2 + G*M*m/(R+h)**2);
    dX[1]=-v;
    return dX;
```

Puis le calcul avec `odeint` et l'affichage

```
# Integration avec odeint
T=pl.linspace(0,120,100)
X0=[0,39000]
X=integrate.odeint(F,X0,T)

# Affichage des courbes
pl.figure("Courbe vitesse"); # à gauche ci-dessous
pl.plot(T,X[:,0],"b")
pl.show()
##
pl.figure("Courbe altitude"); à droite ci-dessous
pl.plot(T,X[:,1],"b")
pl.show()
```





Remarque : on voit que la vitesse maximale est atteinte au bout d'environ 55s et qu'elle est d'environ 350 m./s. A ce moment-là, Félix est à environ 30 km d'altitude, la densité de l'air est, suivant la loi du b), d'environ $0,03 kg.m^{-3}$. Il manque la température de l'air pour pouvoir calculer la vitesse du son dans l'air à cet endroit-là. A vous de me chercher ça ! Mais je pense bien qu'il dépasse la vitesse du son !

d) La méthode `split` avec l'argument `(,)` permet, sur une chaîne de caractères contenant des éléments séparés par des virgules, de renvoyer une liste dont les entrées sont les différents sous-chaîne de la chaîne précédente.

Par exemple si `L="café,lait,chocolat"`, `L.split(",")` renvoie :

```
['café', 'lait', 'chocolat']
```

```
# Initialisation des tableaux de données
T=pl.zeros([len(text)])
V=pl.zeros([len(text)])
H=pl.zeros([len(text)])

# Conversion des données textes en flottants
for i in range(len(text)):
    ligne=text[i].split(",")
    T[i]=float(ligne[0])
```

```
    V[i]=float(ligne[1])
    H[i]=float(ligne[2])

# Superposition aux courbes simulées et mesurées
pl.figure("Courbe vitesse");
pl.plot(T,V,"r")

pl.figure("Courbe altitude");
pl.plot(T,H,"r")
```