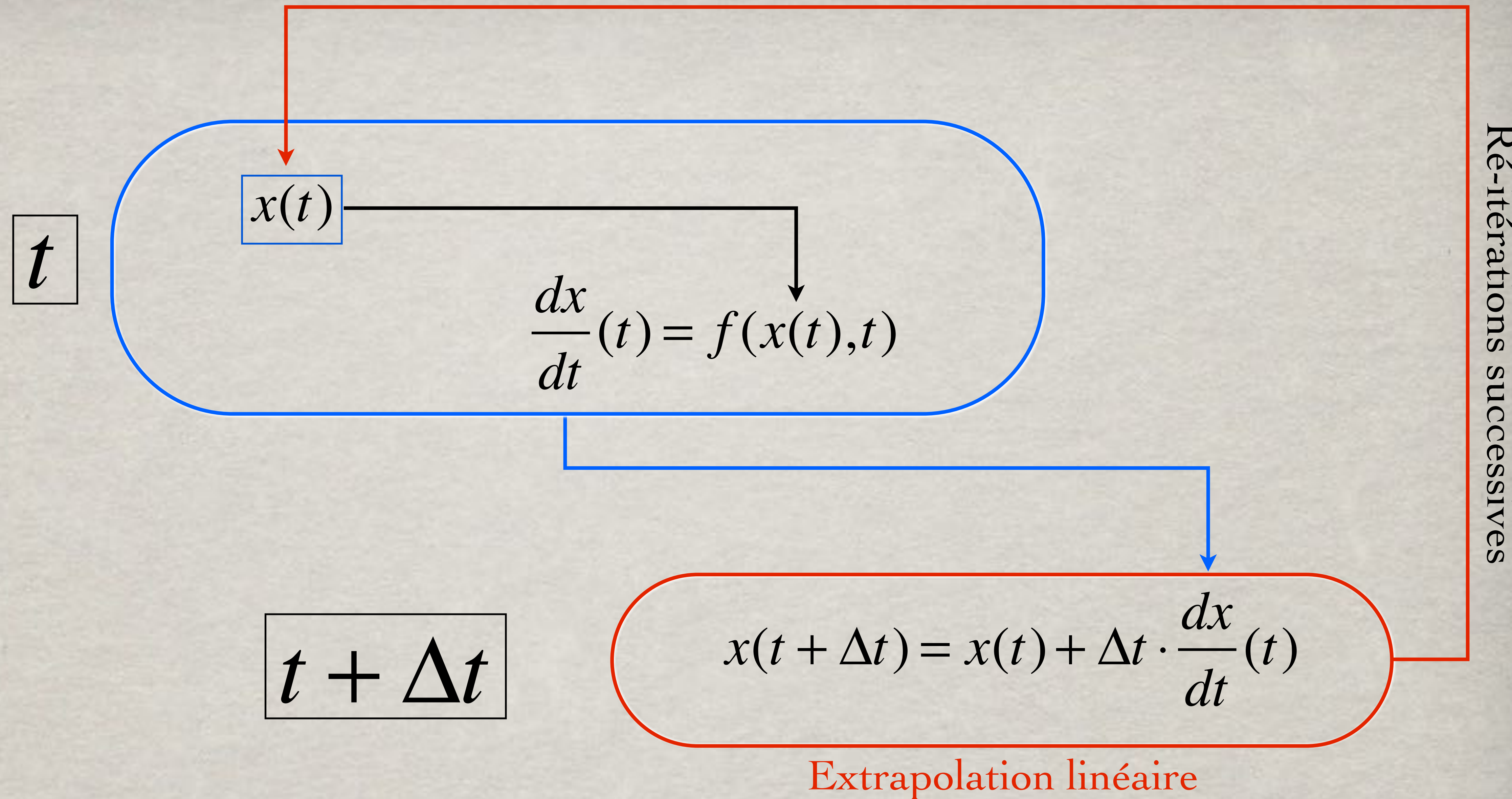
7 - Equations différentielles & méthode d'Euler

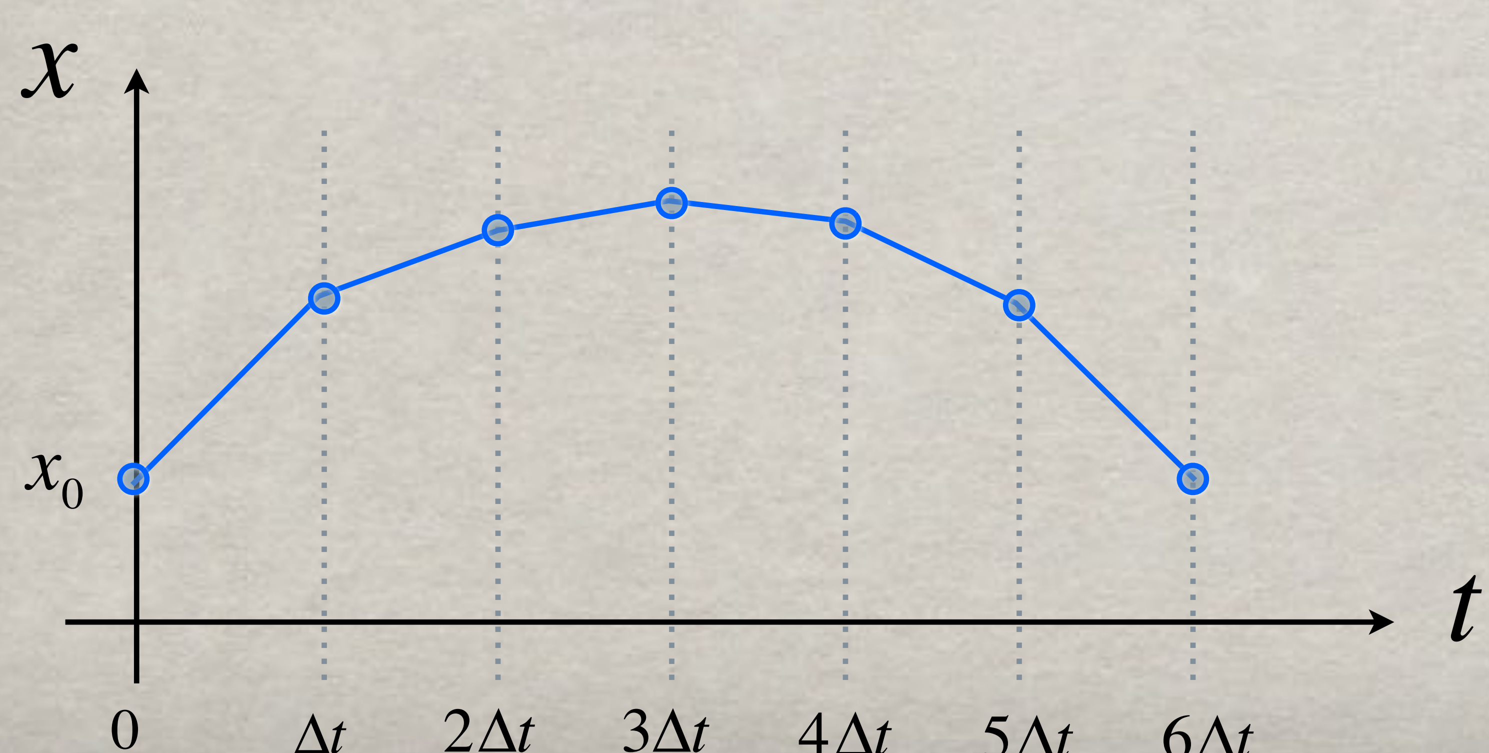
C'est avec cette méthode que les astronomes calculaient à la main les trajectoires des planètes.

C'est très long => «calcul astronomiques» mais très répétitif : l'ordinateur peut réaliser ces calculs rapidement et sans erreur.

Problème de Cauchy :

Problème de Cauchy d'ordre 1 : $\boxed{\frac{dx}{dt} = f(x,t)}$ $\boxed{x(0) = x_0}$


t
$x(t)$
$\frac{dx}{dt}(t) = f(x(t),t)$
$t + \Delta t$
$x(t+\Delta t) = x(t) + \Delta t \cdot \frac{dx}{dt}(t)$
Extrapolation linéaire
Ré-itérations successives


x
x_0
t
0
Δt
$2\Delta t$
$3\Delta t$
$4\Delta t$
$5\Delta t$
$6\Delta t$

1 - Etude d'un cas solvable analytiquemenent

La connaissance d'une solution analytique au problème permet de comparer l'écart entre la simulation numérique et le modèle.

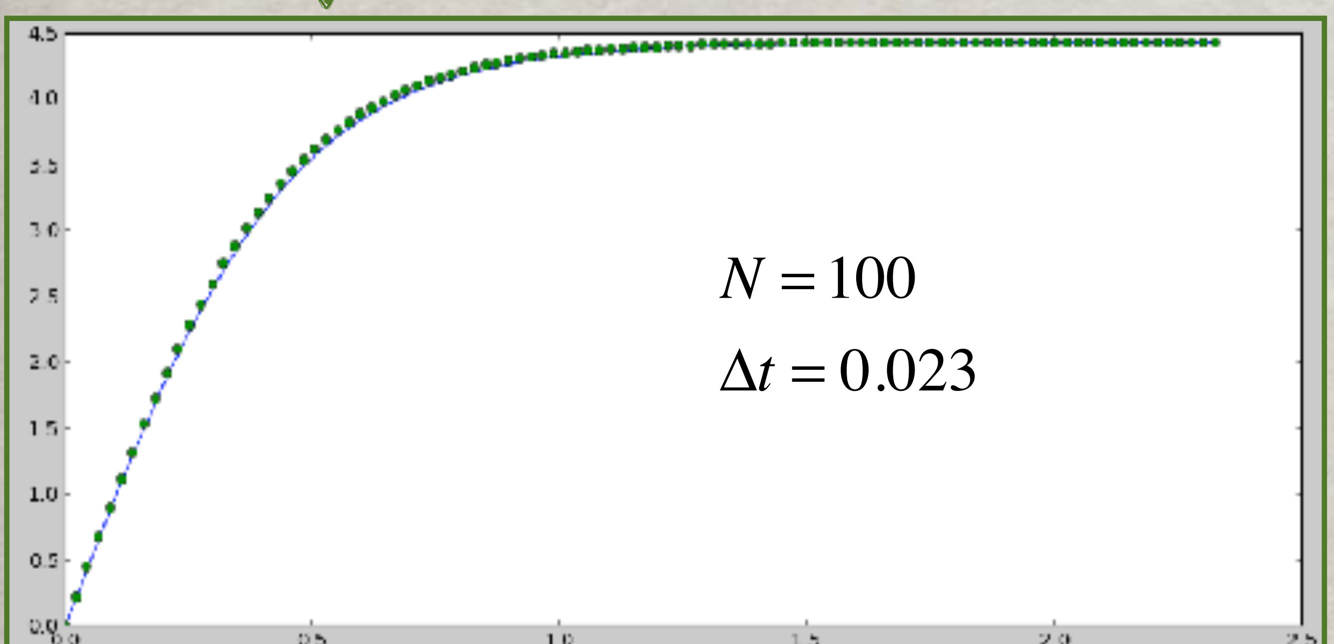
$$\boxed{\frac{dv}{dt} = g - \frac{h}{m} v^2}$$

Solution analytique

vitesse limite

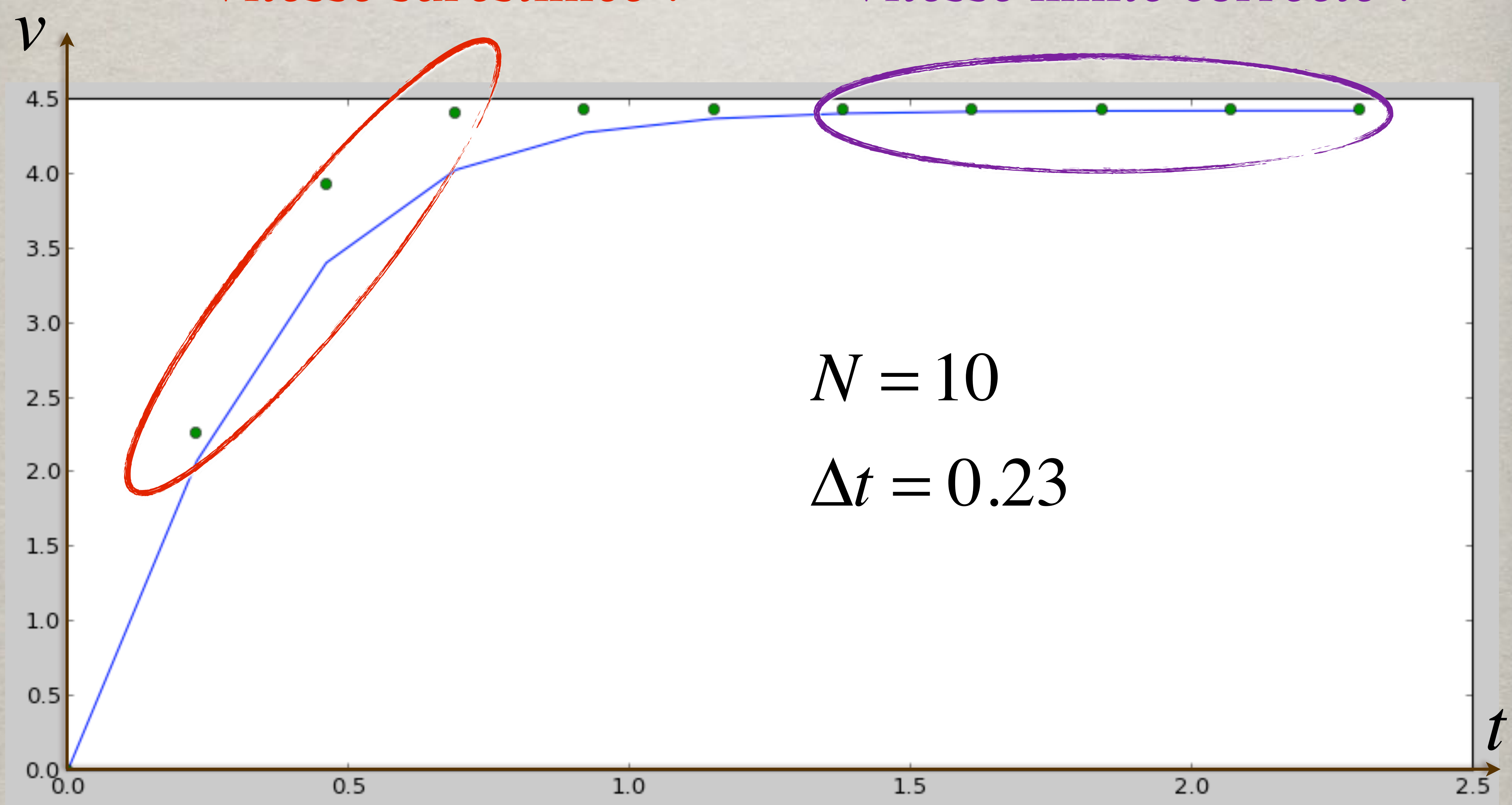
$$\boxed{v(t) = v_0 \tanh\left(\frac{h v_0}{m} t\right) \qquad v_0 = \sqrt{\frac{mg}{h}}}$$

cf méca - 2

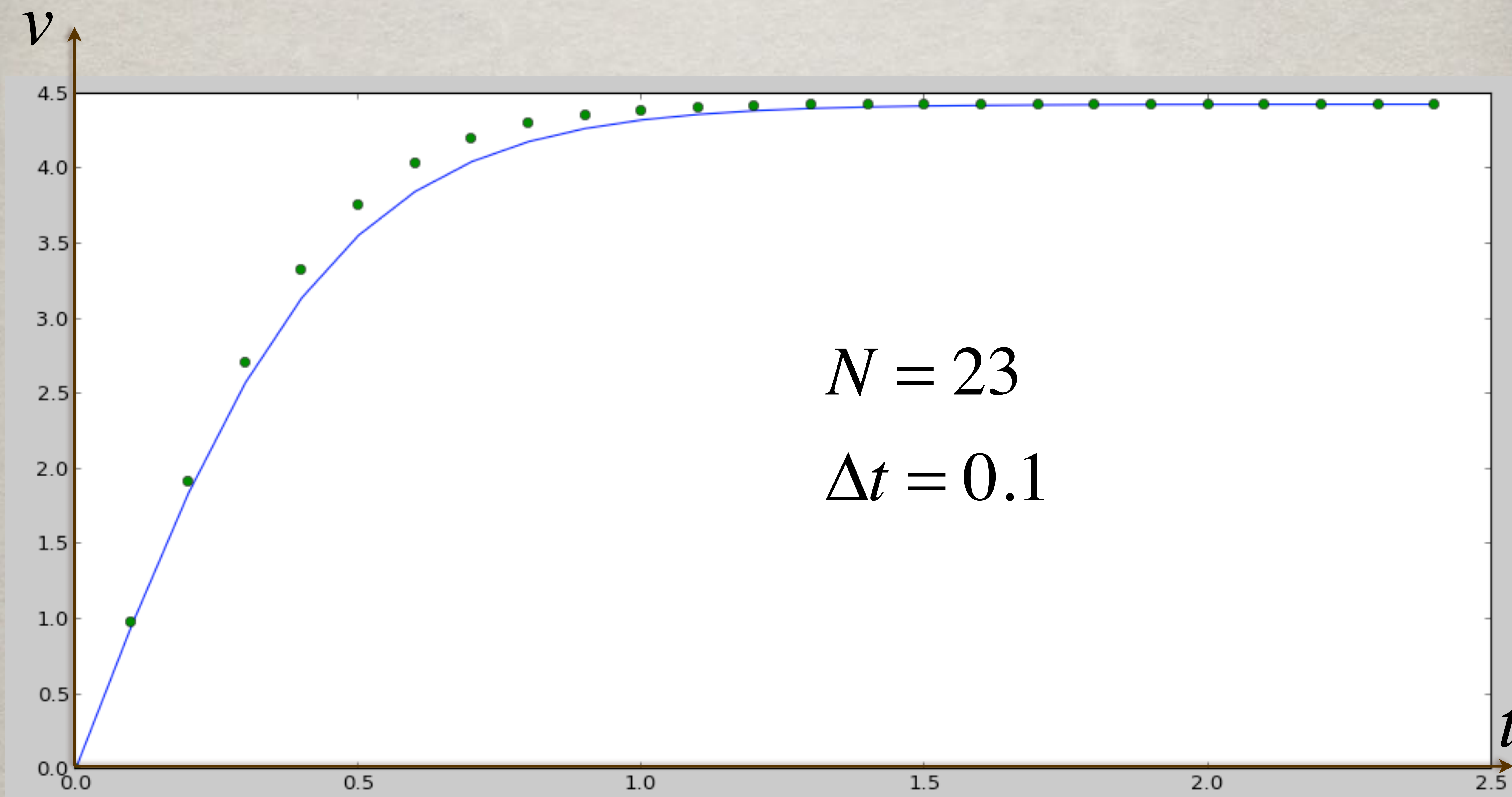


Intégration numérique par la méthode d'Euler

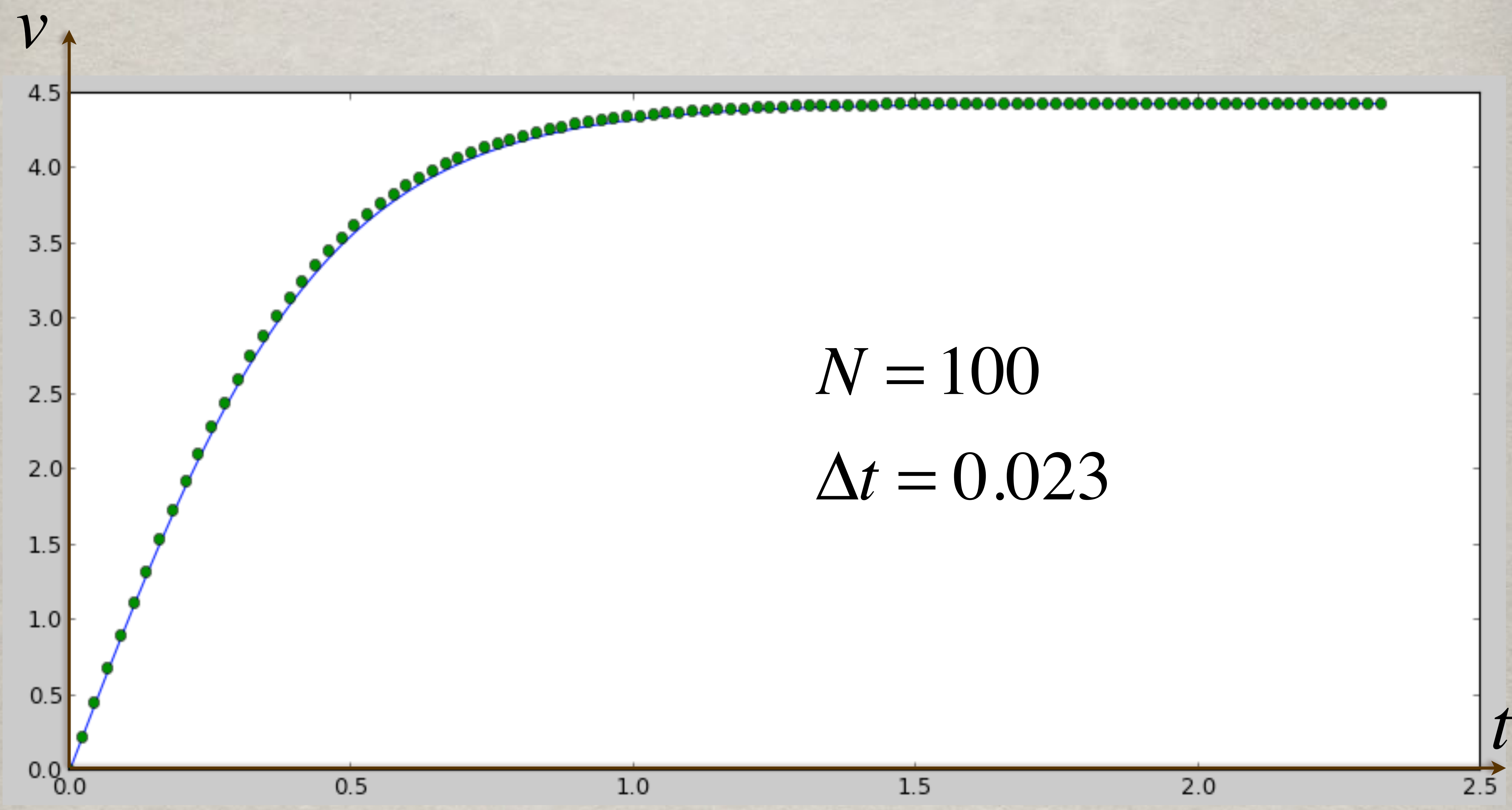
Vitesse surestimée ! Vitesse limite correcte !

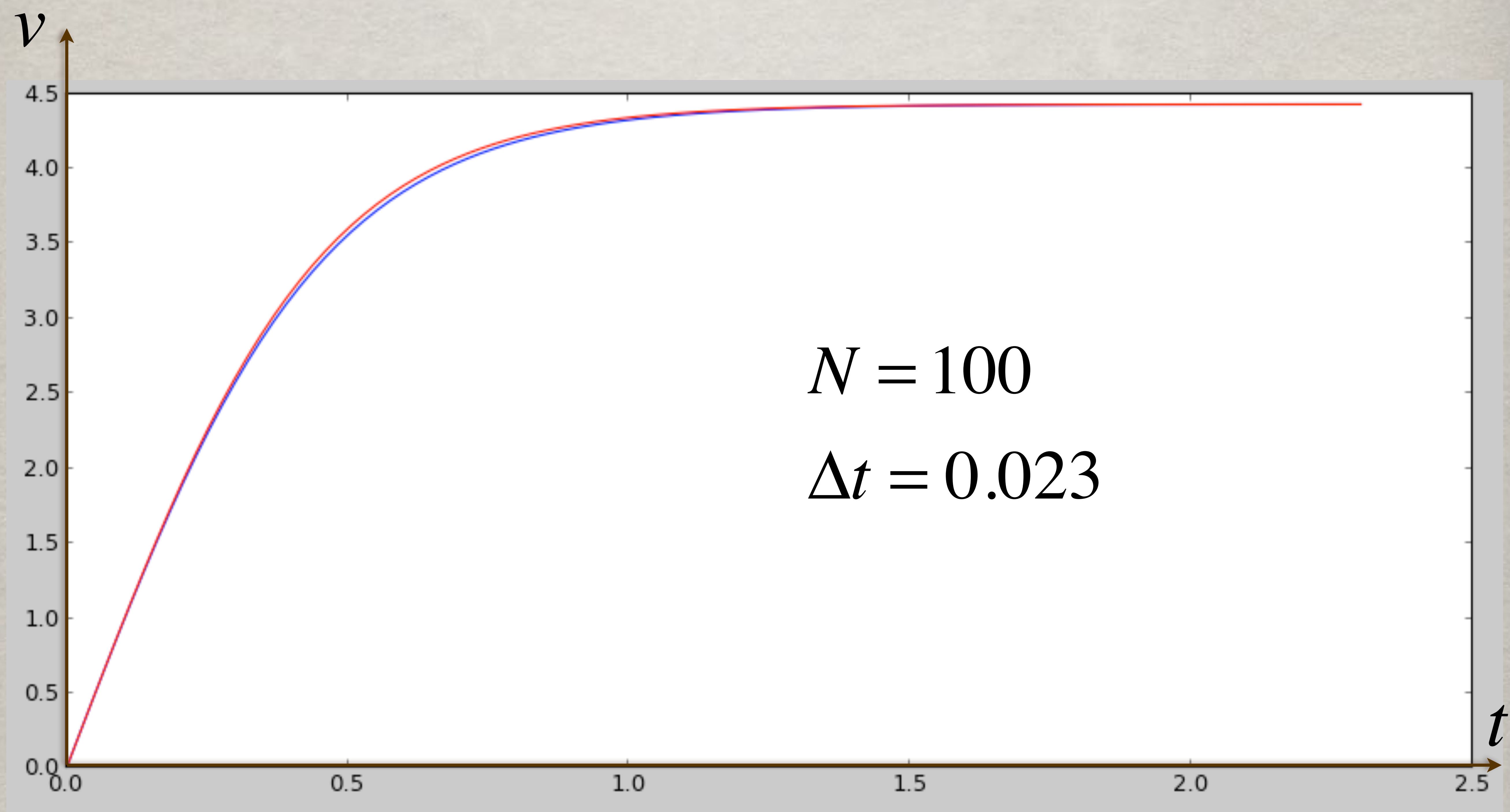


Malgré une intégration erronée, la solution converge ici vers la vitesse limite.
C'est rarement le cas avec la méthode d'Euler.



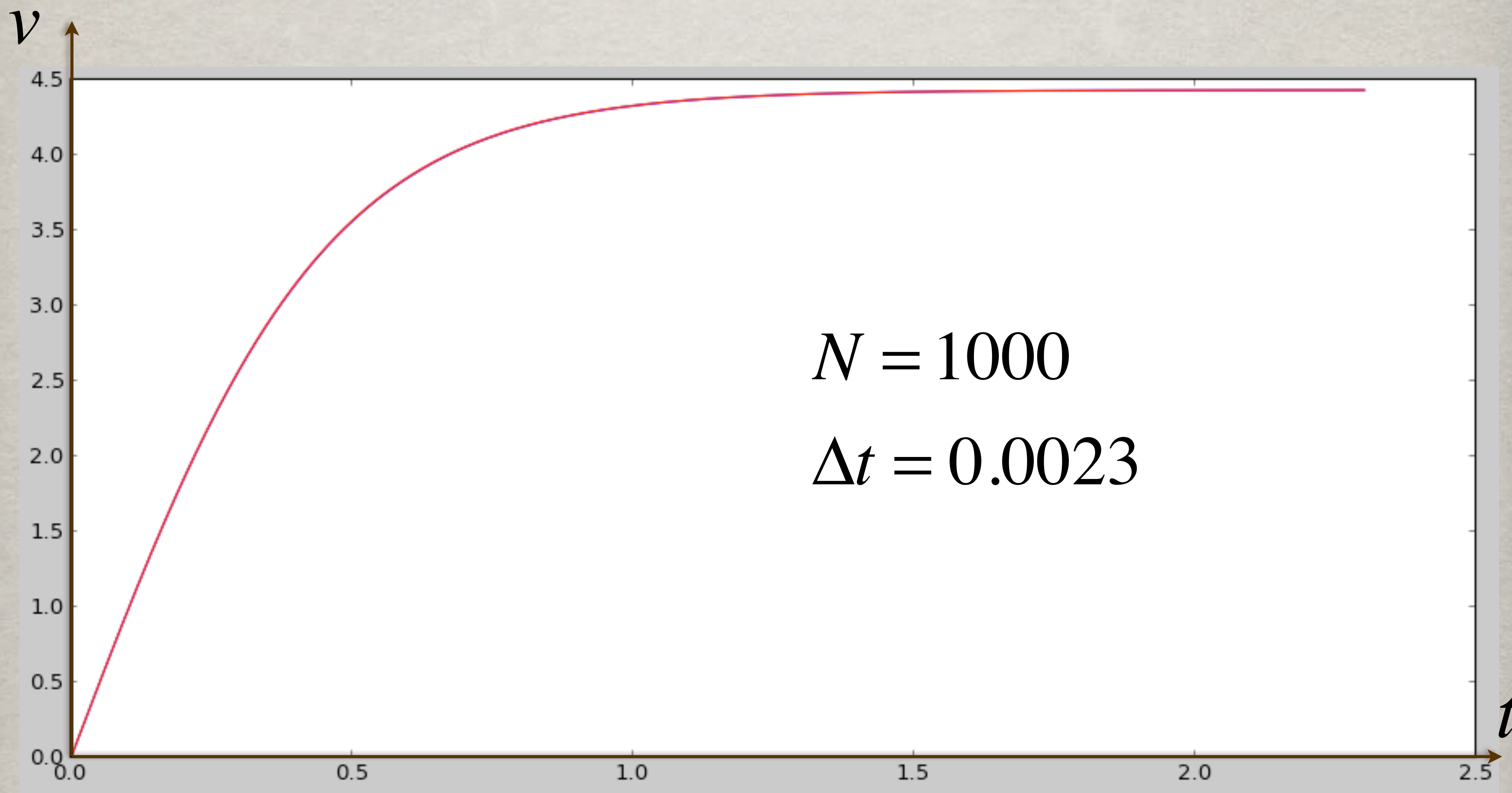
En effet, les frottements son évalués sur la base de la vitesse au point précédent or la vitesse augmente => **Les frottements sont donc sous-estimés !**


v
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
t
N = 100
Δt = 0.023



v
t
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
$N = 100$
$\Delta t = 0.023$

- Pour un nombre suffisant de points on converge vers la solution analytique.
- L'écart entre les deux peut ainsi être réduit à volonté.
  Mais cela augmente le temps de calcul, et ne convergera JAMAIS exactement.



La solution analytique dépendant des hypothèses simplificatrices du modèle, on ne peut pas la considérer exacte non plus de toute façon.

LA SOLUTION NUMÉRIQUE EST DE CE POINT DE VUE TOUT AUSSI VALIDE.

<u>Exercice :</u> équation du second ordre

- Proposer une mise en équation du pendule simple par la méthode d'Euler.
  On se place dans le cas général, et pour un rotateur rigide.

- Tracer les courbes $\theta(t)$ pour différentes conditions initiales.

- Déterminer la vitesse angulaire $\dot{\theta}(t)$

- Tracer les portraits de phase pour différentes conditions initiales

- Calculer la tension exercée sur la tige.

2 - Etude d’un cas non-solvable analytiquemenent par la méthode d’Euler

- Il n’est pas toujours possible de trouver une solution analytique à un problème.

- La modélisation mathématique trouve vite ses limites dès que la géométrie d’un problème n’est plus élémentaire : Electromagnétisme, diffusion thermique, mécanique des fluides, étude des matériaux : déformation & contraintes.

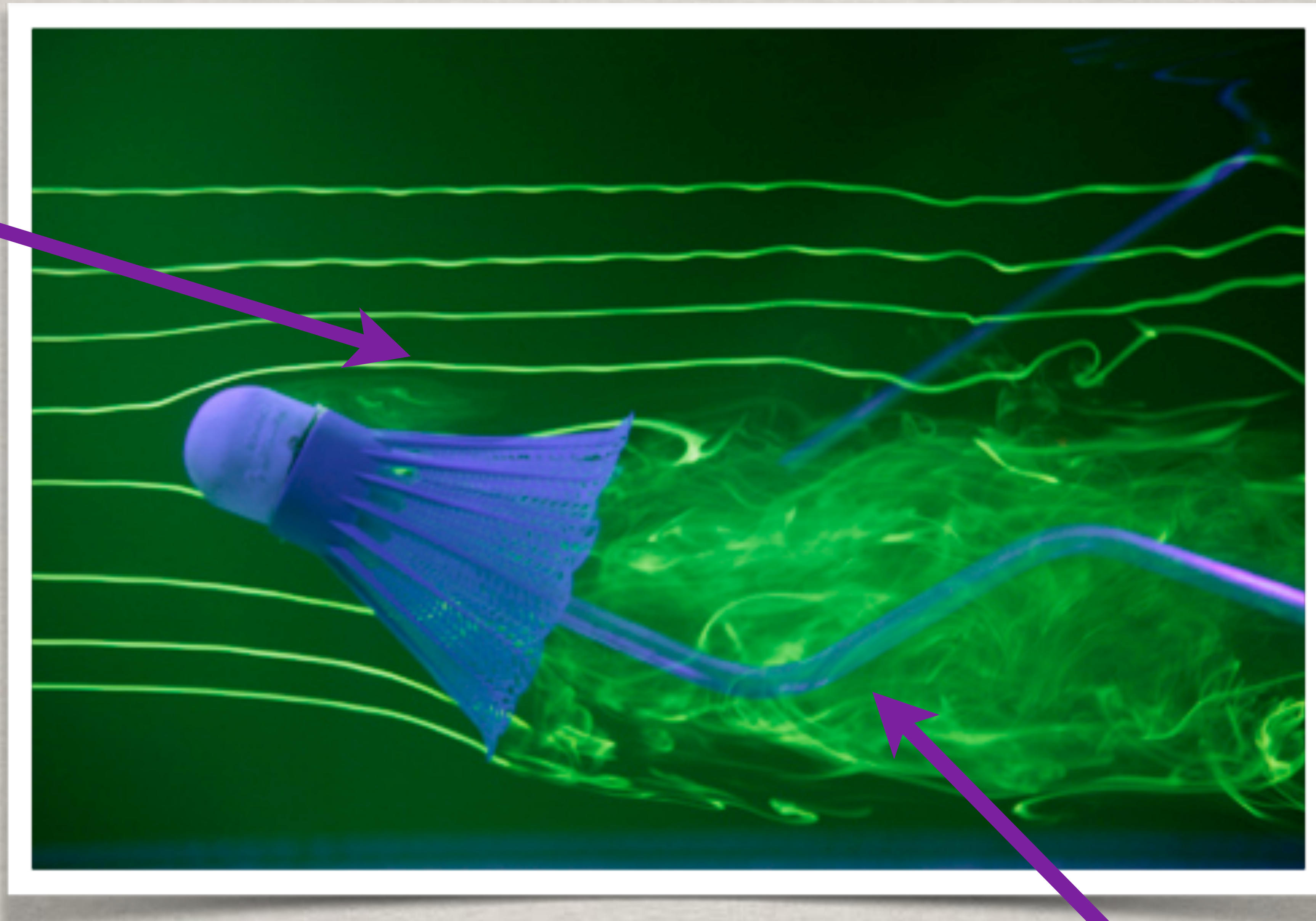
- L’exemple précédent n’a pas de solution analytique en 2D.
  De plus on obtient un système d’équations différentielles couplées et Non Linéaires.

Objectifs :

- Calculer et tracer la trajectoire d’un volant de badmington : frottements quadratiques
- Enregistrer les données physiques de la trajectoire dans un fichier.

Ecoulement autour du volant de badmington : très complexe

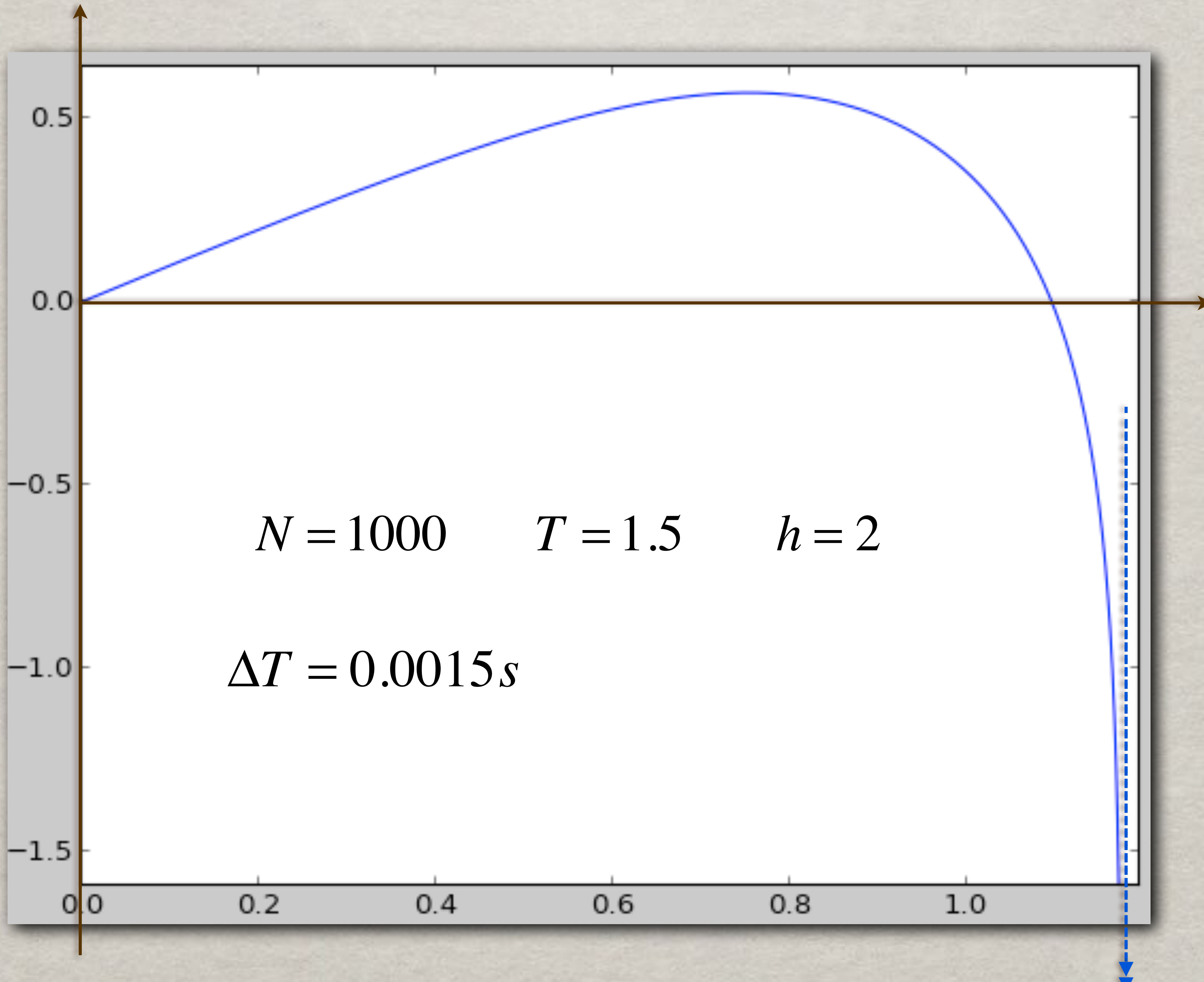
Laminaire



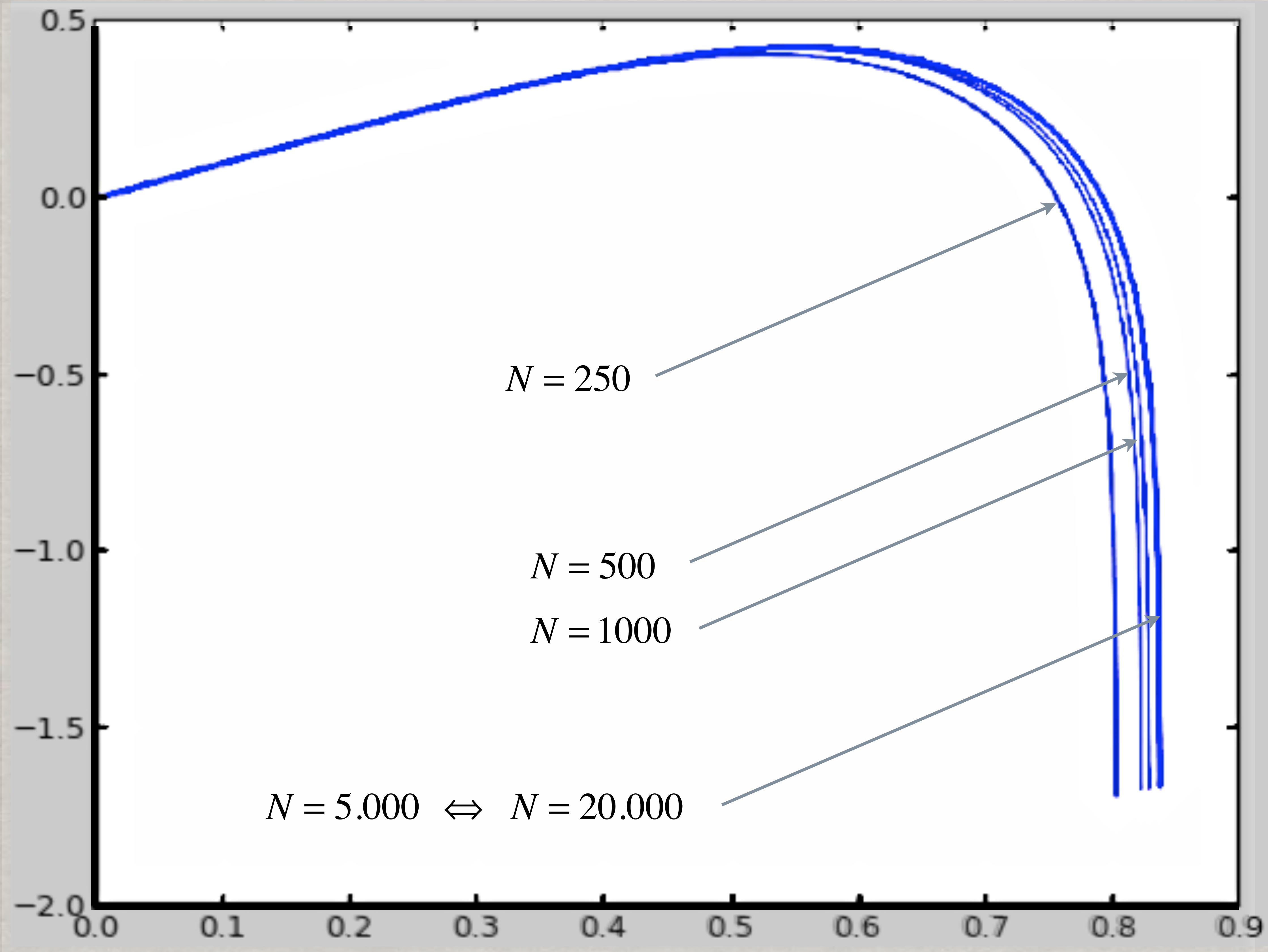
Turbulent : effet de Trainée

On résume tout cela à l'action d'une force quadratique : $\boxed{\vec{F} = -hv^2 \vec{u}_v}$

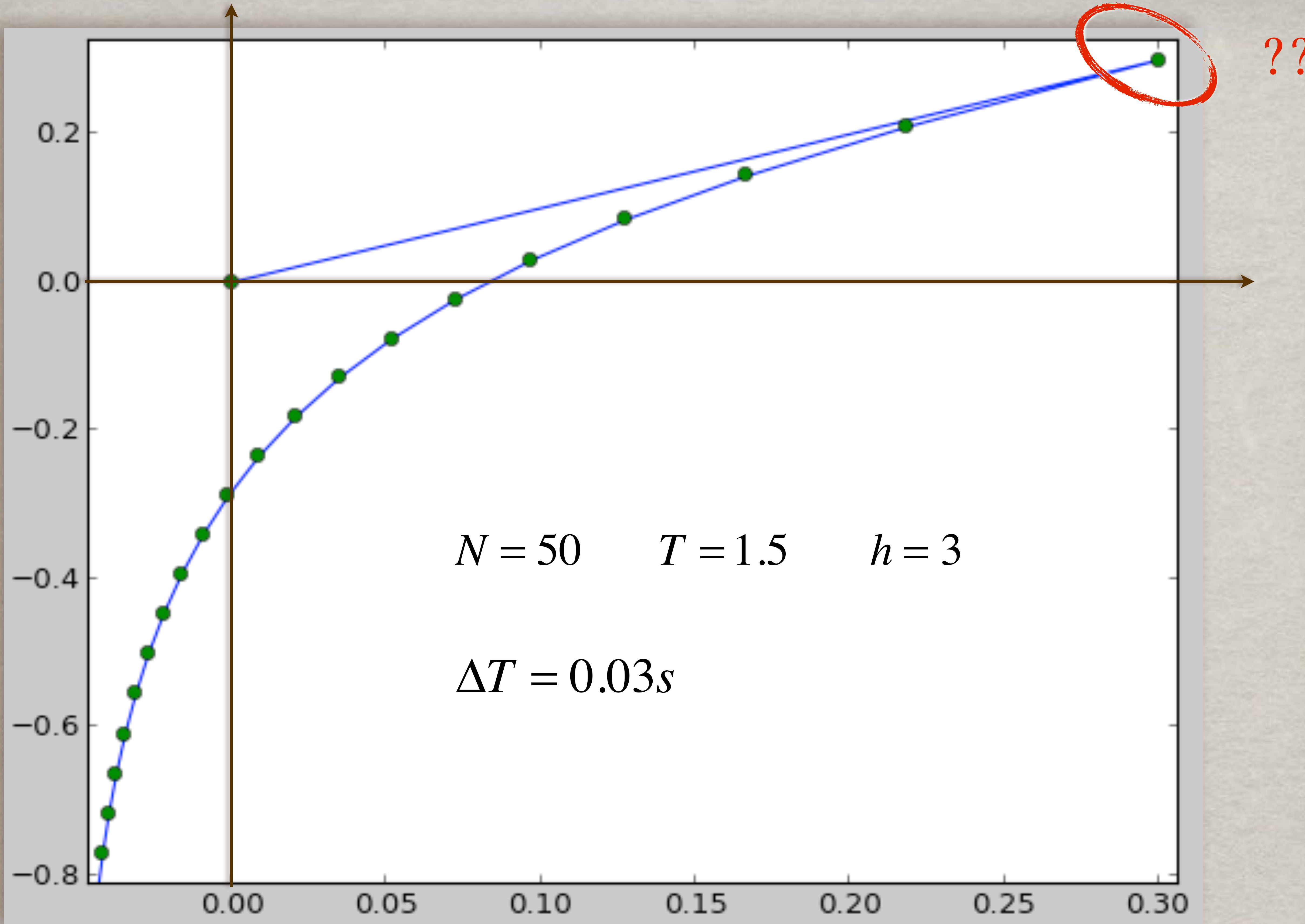
Calcul de la trajectoire balistique par la méthode d'Euler.



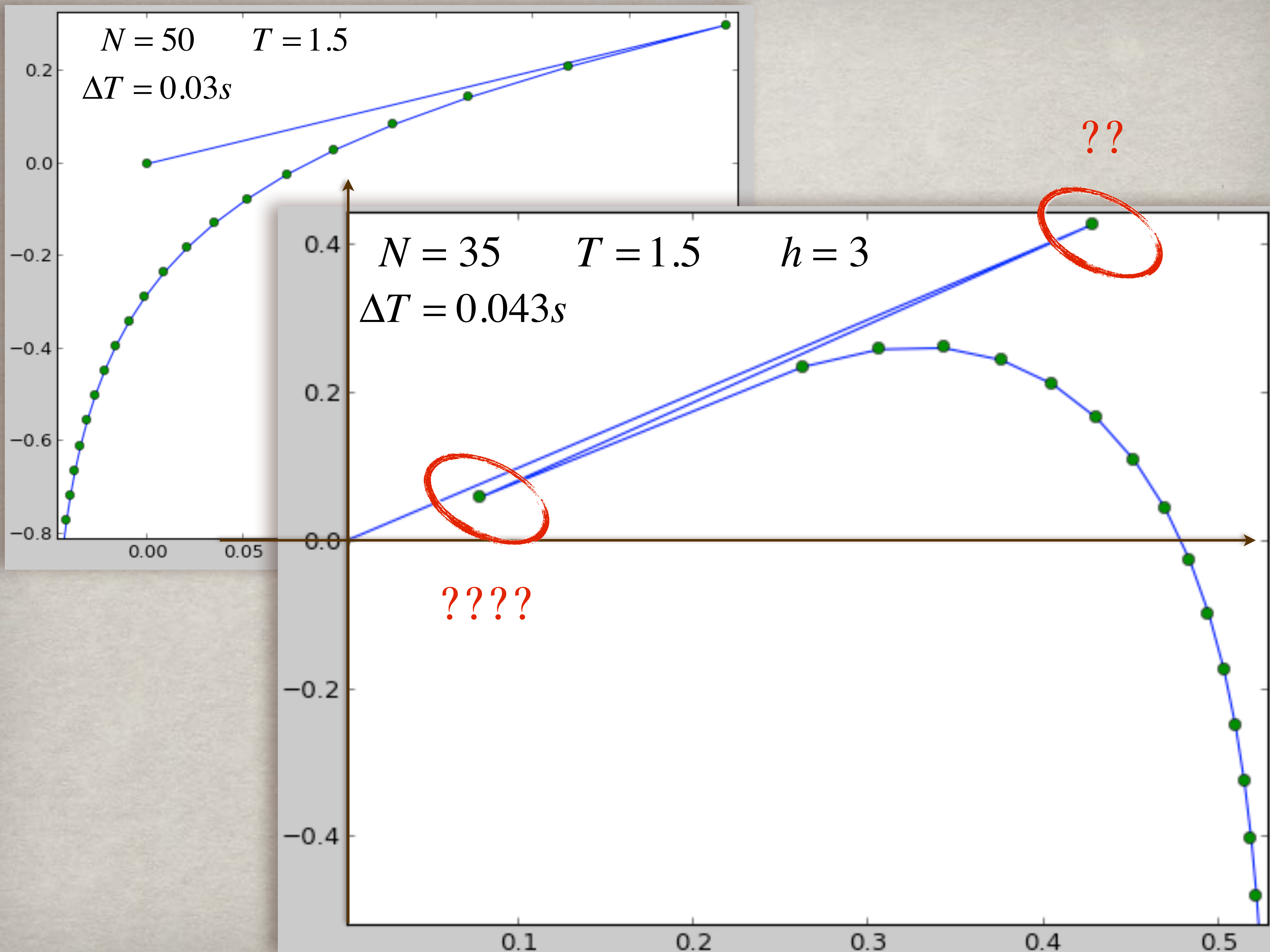
Etude de convergence numérique :



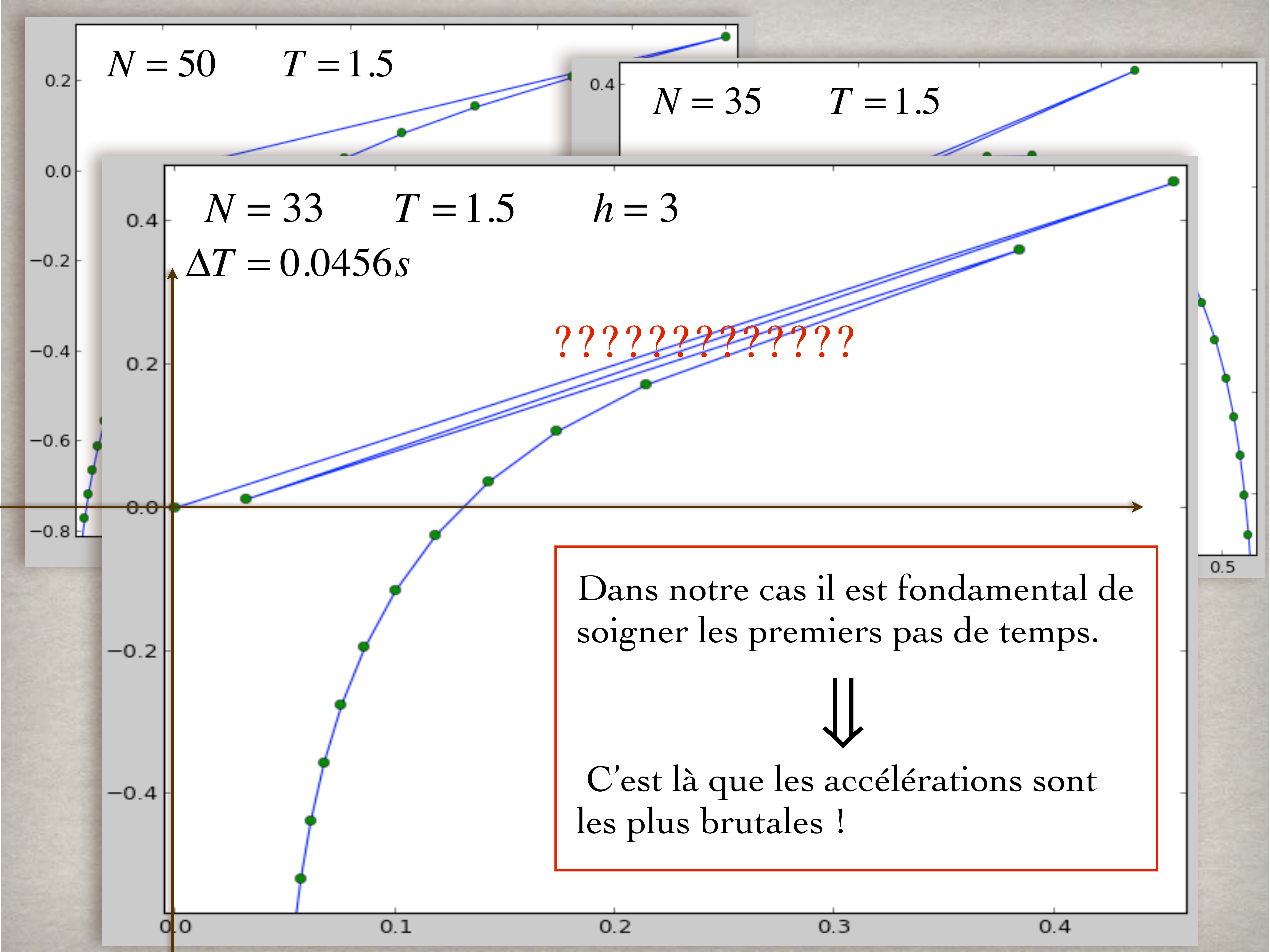
La solution numérique converge assez rapidement vers un profil unique N ~ 10.000



Attention : la méthode d’Euler propage les erreurs sans garde fou !



N = 50 T = 1.5
ΔT = 0.03s
0.2
0.0
−0.2
−0.4
−0.6
−0.8
0.00
0.05
N = 35 T = 1.5 h = 3
ΔT = 0.043s
0.4
0.2
0.0
−0.2
−0.4
0.1
0.2
0.3
0.4
0.5
??
????



N = 50 T = 1.5
N = 35 T = 1.5
N = 33 T = 1.5 h = 3
ΔT = 0.0456s
??????????????
Dans notre cas il est fondamental de soigner les premiers pas de temps.
⇓
C'est là que les accélérations sont les plus brutales !

```
#JCourtin 03/14 : Tir balistique avec frottements quadratiques
from math import *
from matplotlib import pyplot as plt

##PARAMETRES
h=2;    m=1;    g=9.81;    v0=14.14;    alpha=pi/4   #données arbitraires ....

##CONDITION INITIALES
time=[0.]
x=[0];    vx=[v0*cos(alpha)];     ax = 0.0;
y=[0];    vy=[v0*sin(alpha)];     ay = -g

N=1000;  Tmax=1.5;  dt=Tmax/N

##Création du fichier -- Ecriture d'un entête
myfile='/Users/jcourtin/Desktop/Python/ENV/Pyzo/Workspace_PROJET/Workspace_Euler/data.csv'

fichier=open(myfile,'w')
fichier.write("{},  {},  {},  {},  {},  {},  {},  {}\n".format('i',  'time',   'x[i]','y[i]',  'vx[i]',  'vy[i]',  'ax',  'ay'))

#Ecriture des conditions initiales --> Attention :  gestion des formats numériques
line=" {:5d},   {:10.4e},   {:10.4e},   {:10.4e},   {:10.4e},   {:10.4e},   {:10.4e}, {:10.4e}
\n".format(0,time[0], x[0], y[0], vx[0], vy[0], ax, ay)

print(line);               fichier.write(line)
```

```
## BOUCLE D'INTEGRATION
for i in range(1,N):

    time+=[time[-1]+dt]        #actualisation du temps --> Euler

    x+=[x[-1] + vx[-1]*dt]     #actualisation de la position --> Euler
    y+=[y[-1] + vy[-1]*dt]

    ax = -h/m*(vx[-1]**2+vy[-1]**2)**(0.5)*vx[-1]   #calcul de l'accélération (cf PFD)
    ay = -g - h/m*(vx[-1]**2+vy[-1]**2)**(0.5)*vy[-1]

    vx+=[vx[-1]+ax*dt]         #actualisation de la vitesse --> Euler
    vy+=[vy[-1]+ay*dt]

    #Ecriture des données --> Attention :  gestion des formats numériques
    line=" {:5d},   {:10.4e},   {:10.4e},   {:10.4e},   {:10.4e},   {:10.4e},   {:10.4e},   {:10.4e}
        \n".format(i,time[i],x[i],y[i],vx[i],vy[i], ax, ay)

    print(line);    fichier.write(line)

fichier.close()

#Tracé des courbes
plt.close();     plt.plot(x,y);     plt.plot(x,y,'o');        plt.show()
```

Etude du pendule simple

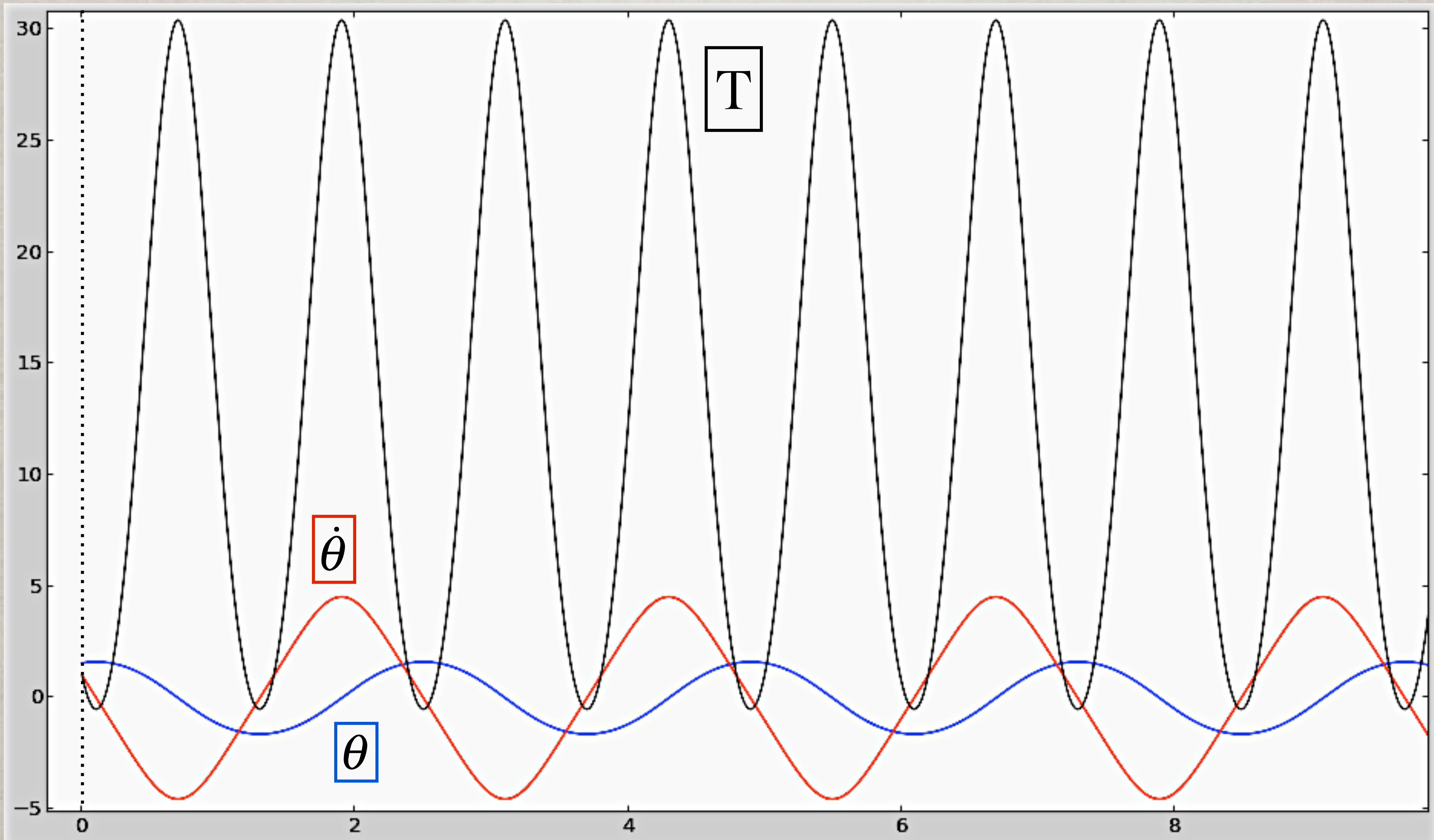
par intégration numérique d'Euler

Mise en équation :

Méthode d'Euler :

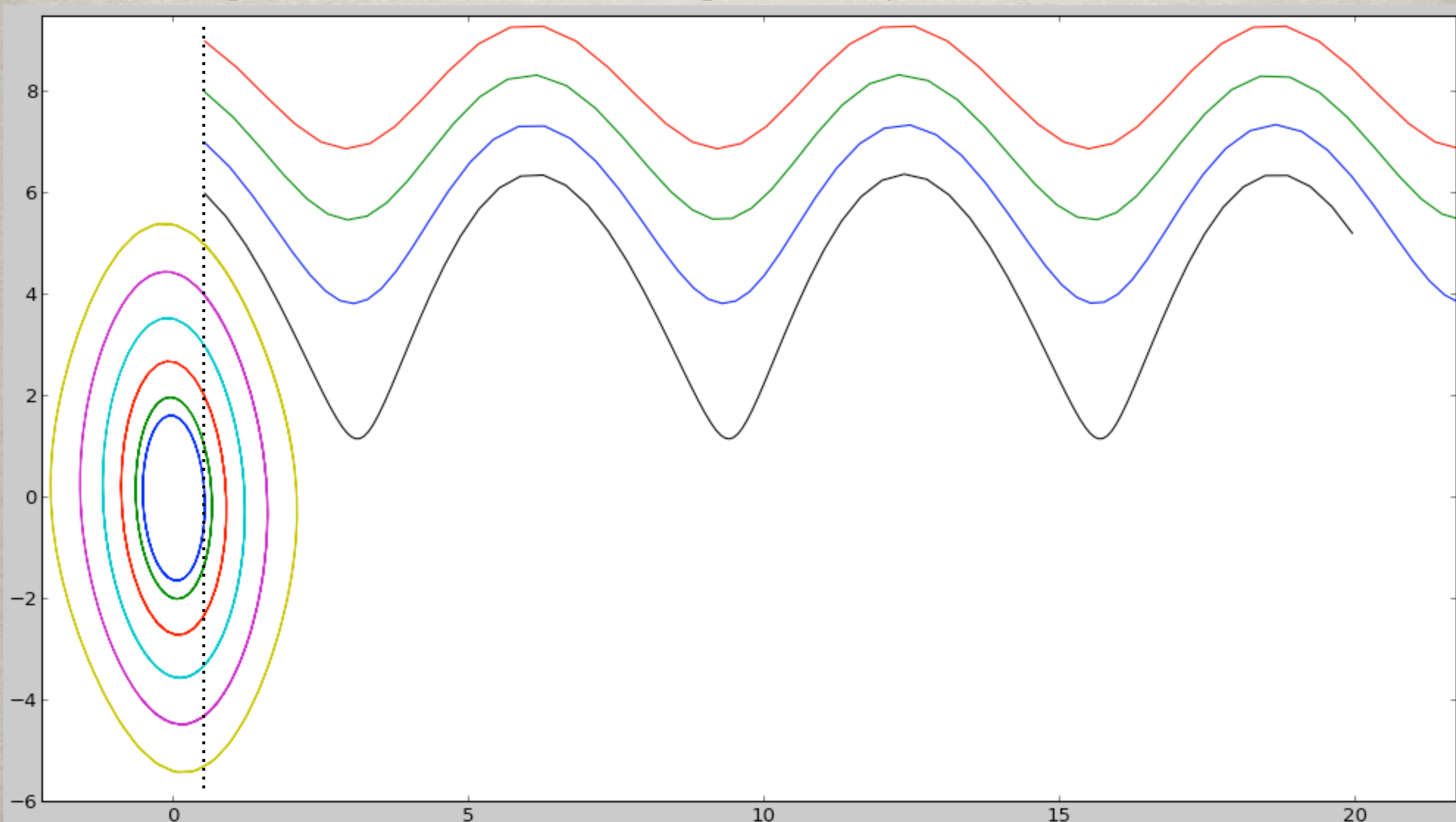
```
#paramètres physiques
m=1;   R=1;    v0=7;    g=9.81;   k=0.3

#param. simulation.                                  #conditions initiales
N=501;   Tmax=10.01;    dt=Tmax/N;                   theta0=pi/2;    theta_dot0=v0/R


############### Fonction pendule pour le rotateur rigide #############
def pendule(Tmax, N, v0=7, theta0=pi/2, k=0.3, g=9.81, R=1, m=1):
   #conditions initiales
   time=[0.];    theta=[theta0];   dt=Tmax/N

   theta_dot0=v0/R;   theta_dot=[theta_dot0]

   T=[m*g*cos(theta[-1])+m*R*(theta_dot[-1])**2]

   #boucle d'intégration d'Euler
   for i in range(1,N):

      theta_dotdot = -g/R*sin(theta[-1]) -k/m*R*theta_dot[-1]  #calcul theta point point

      time+=[time[-1]+dt]

      theta+=[theta[-1]+theta_dot[-1]*dt]  #Euler theta

      theta_dot+=[theta_dot[-1]+theta_dotdot*dt] #Euler theta point point

      T+=[m*g*cos(theta[-1])+m*R*(theta_dot[-1])**2] #calcul tension

   return (time, theta, theta_dot, T)
```

SOL = pendule(Tmax, 10000, v0=1, theta0=pi/2, k=0., g=9.81, R=1, m=1)



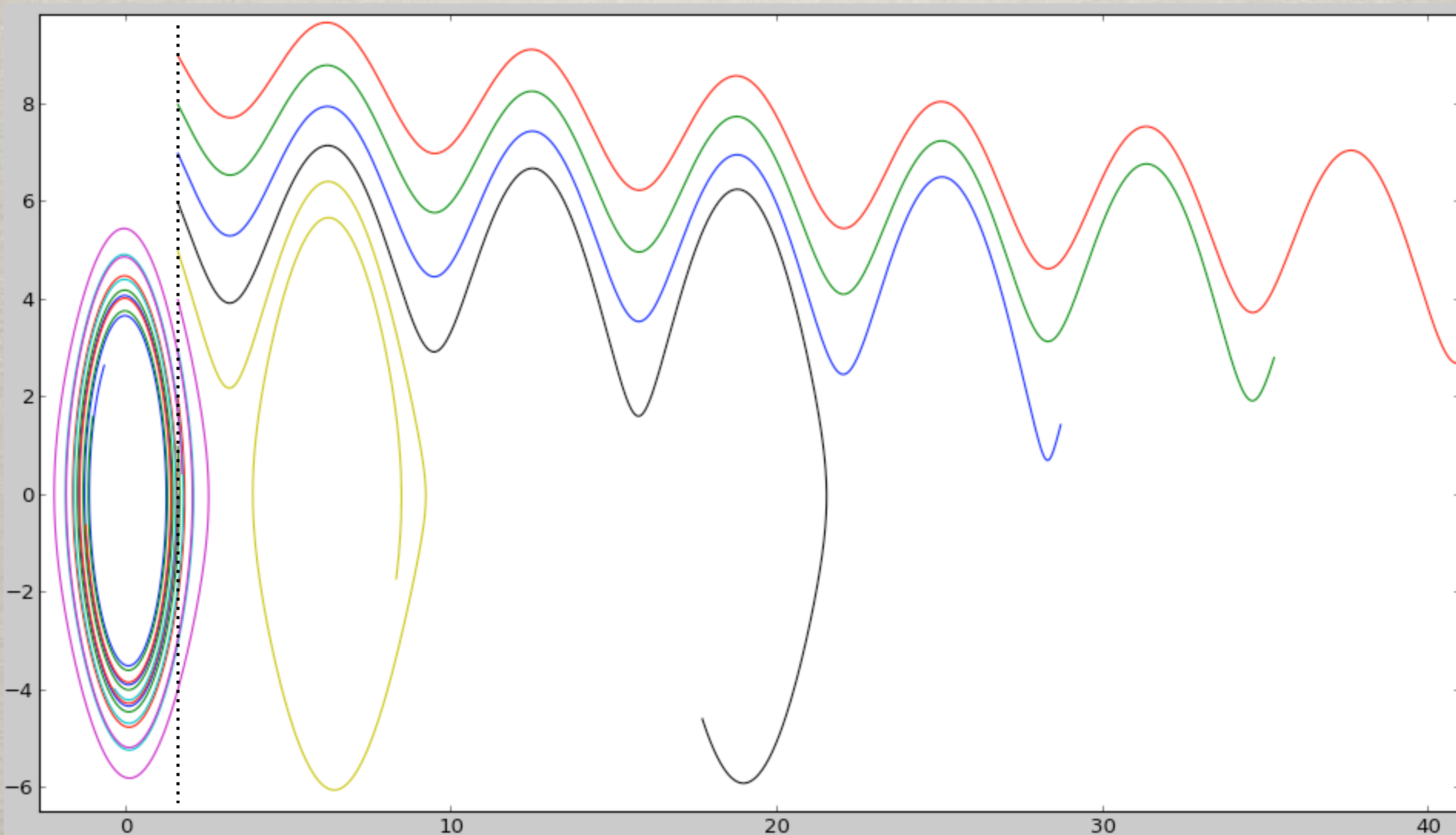
Protrait de phase sans frottement

SOL = pendule(6, 100, v0=0+i, theta0=pi/6, k=0.0, g=9.81, R=1, m=1)



Protrait de phase avec frottements

SOL = pendule(Tmax = 6, 6*100, v0=0+i, theta0=pi/2, k=0.1, g=9.81, R=1, m=1)



Protrait de phase avec frottements

SOL = pendule(Tmax = 6, 6*100, v0=0+i, theta0=pi/2, k=0.1, g=9.81, R=1, m=1)