

Chapitre M3 – Capacité numérique – Mise en évidence d'effets non linéaires

MPSI 1, Lycée Saint Louis

Année scolaire 2025–2026

Importation des modules

```
[1]: import numpy as np
      %matplotlib inline
      import matplotlib.pyplot as plt
      import scipy.integrate as scpi

      #Reglage taille des figures
      plt.rcParams['figure.figsize']=(20.0,10.0)
      #Reglage taille des fonts des xtics et ytics des graphes
      label_size=30
      plt.rcParams['xtick.labelsize']= label_size
      plt.rcParams['ytick.labelsize']= label_size
      plt.rcParams['font.size'] = label_size
```

Définition de la fonction

```
[2]: def fct_pendule(t,Y) :
      return [Y[1],-2*np.pi*np.sin(Y[0])]
```

Paramètres

```
[3]: tab_Y0=np.vander(np.linspace(0, np.pi, 10,endpoint=False), 2)
      tab_Y0[:,1]=np.zeros(np.size(tab_Y0[:,1]))
      ti=0.
      tf=10.
      N=int(tf-ti)*100
      data_t=np.linspace(ti,tf,N+1)
```

Résolution de l'équation différentielle pour différents θ_0 et $\dot{\theta}_0 = 0$.

```
[4]: list_t=[]
      list_theta=[]
      list_omega=[]
      for k in range(np.shape(tab_Y0)[0]) :
          sol=scpi.solve_ivp(fct_pendule, [ti,tf], tab_Y0[k], t_eval=data_t, dense_output=True)
          tab_t=sol.t
          tab_theta=sol.y[0]
          tab_omega=sol.y[1]
          list_t.append(tab_t)
          list_theta.append(tab_theta)
          list_omega.append(tab_omega)
```

Tracés des solutions

```
[5]: fig=plt.figure(figsize=(20,10))

      ax_theta=fig.add_subplot(2,2,1)
      ax_omega=fig.add_subplot(2,2,3)
      ax_traj=fig.add_subplot(1,2,2,projection='polar')
      ax_traj.set_theta_zero_location('S')
      ax_omega.set_xlabel('temps en unité de $T_0$')
```

```

ax_omega.set_ylabel('\omega')
ax_theta.set_ylabel('\theta')
ax_traj.set_title("Trajectoires")
ax_traj.set_yticks([0,1,2])
ax_traj.set_ylim([0,2])

kmax=np.shape(tab_Y0)[0]
for k in range(kmax) :
    color_choice=(1-k/kmax,k*(0.5)/kmax,k*(0.5)/kmax,0.5+k*(0.5)/kmax)
    ax_theta.plot(list_t[k],list_theta[k],'-',color=color_choice)
    ax_omega.plot(list_t[k],list_omega[k],'--',color=color_choice,lw=2)
    ax_traj.plot(list_theta[k],(1+0.1*k)*np.ones(np.size(list_theta[k])),'-',color=color_choice)
plt.show()

```

