Oscillations linéaires et non-linéaires

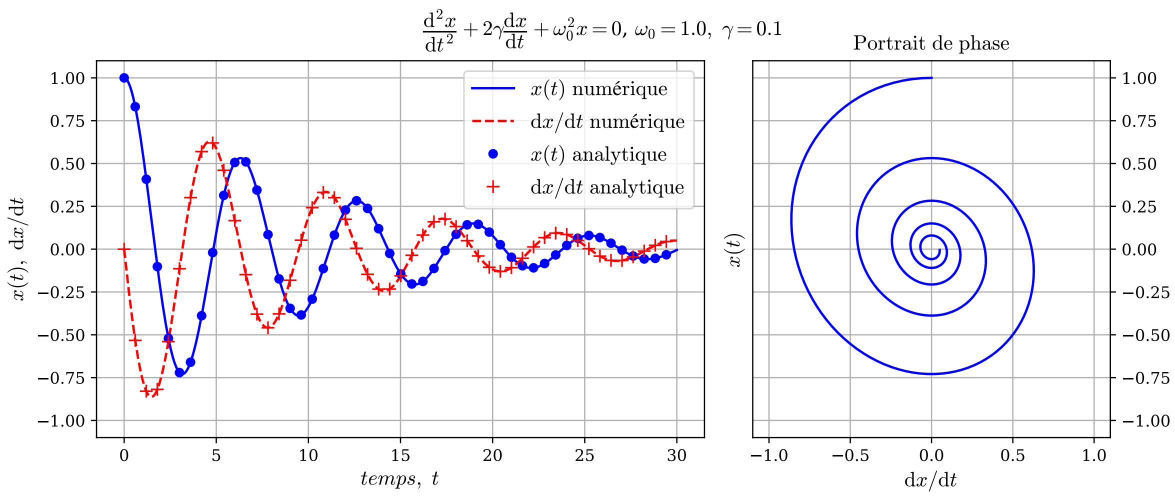
Modélisation numérique de l'oscillateur amorti, de l'oscillateur forcé et de la résonance linéaire et non-linéaire.

Pour décrire la dynamique d'un oscillateur linéaire il faut résoudre une équation différentielle d'ordre 2. L'oscillateur libre est décrit par l'équation suivante :

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + 2\gamma\frac{\mathrm{d}x}{\mathrm{d}t} + \omega_0^2 x = 0$$

L'utilisation de la fonction `odeint()` pour la solution numérique de cette équation prévoit sa séparation en deux équations d'ordre 1. Obtenez les solutions numériques pour $\omega_0 = 1$, $\gamma = 0.1$ et 0.4. Conditions initiales : $x_0 = 1$ et $v_0 = 0$. Présentez les résultats graphiquement comme fonctions du temps et comme portraits de phase : $(\mathrm{d}x/\mathrm{d}t,\, x(t))$. Pour changer la taille des subplots comme dans la figure ci-dessous utilisez `plt.subplots()` :

```
fig, (ax1,ax2) = plt.subplots(1, 2, gridspec_kw =
{ 'width_ratios':(1.7,1) } , num = 'fig name', clear = True)
ax1.plot(x, t, ... ) ; ax1.set_xlabel(r'$ temps,\ t $') ...
```



Comparez les graphiques numériques avec la solution analytique ($v_0 = 0$) :

$$x(t) = x_0\left(\cos(\omega t) + \frac{\gamma\sin(\omega t)}{\omega}\right)e^{-\gamma t},\ \frac{\mathrm{d}x}{\mathrm{d}t} = -x_0\frac{\omega_0^2}{\omega}\sin(\omega t)e^{-\gamma t},\ \omega = \sqrt{\omega_0^2 - \gamma^2}$$

Modélisez l'oscillateur avec la force extérieure :

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + 2\gamma\frac{\mathrm{d}x}{\mathrm{d}t} + \omega_0^2 x = f(t)$$

Prenez comme exemple la pulsion rectangulaire (trouvez une expression 'pythonique' pour la fonction $f(t)$ correspondante – fonction créneau). Présentez le résultat graphiquement comme fonctions du temps et comme portrait de phase : $(\mathrm{d}x/\mathrm{d}t,\, x(t))$.

Pour obtenir une résonance linéaire on applique la force harmonique $f(t) = f_0\sin(\omega_f t)$. Présentez le résultat de la modélisation numérique pour $\omega_f = 0.9$ comme fonction du temps et comme portrait de phase. Comment se manifeste la différence des fréquences ω_0 et ω_f ?

Trouvez comment on peut extraire l'amplitude finale des oscillations forcées de la solution numérique. Utilisez la boucle sur ω_f pour obtenir la forme de la résonance, les amplitudes de x et $\mathrm{d}x/\mathrm{d}t$ en fonction de ω_f. Remarquez un décalage du maximum de l'amplitude du déplacement x de $\omega_0 = 1$ ($\gamma = 0.15$).

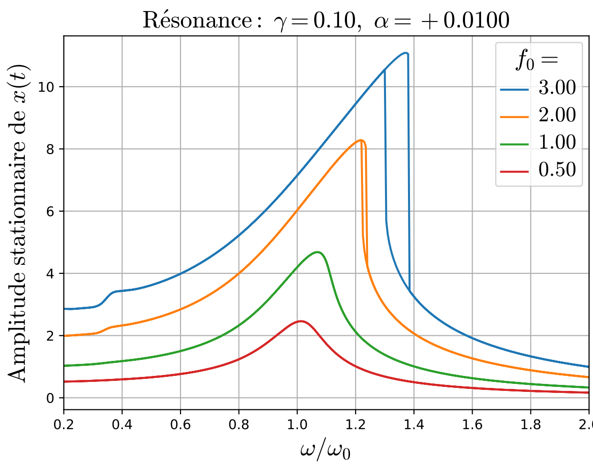
Comparez les graphiques numériques avec les solutions analytiques :

$$a_x(\omega_f) = \frac{f_0}{\sqrt{(\omega_0^2 - \omega_f^2)^2 + 4\gamma^2\omega_f^2}}, \ \ a_v(\omega_f) = \omega_f a_x(\omega_f)$$

Calculez comment la résonance change pour un oscillateur non-linéaire :

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + 2\gamma\frac{\mathrm{d}x}{\mathrm{d}t} + \omega_0^2(x + \alpha x^3) = f_0 \sin(\omega_f t)$$

Il faut reproduire un hystérésis de la résonance pour la fréquence ω_f croissante et décroissante comme dans l'exemple ci-dessous. Pour y arriver on doit, pour chaque nouvelle marche en ω_f, prendre les valeurs finales de x, $\mathrm{d}x/\mathrm{d}t$ et la phase de la force de la marche précédente comme conditions initiales.



Pour finir, vous pouvez analyser une résonance réelle – la résonance d'un circuit RLC : $L = 1$ H, $C = 0.1$ μF, $R = 350$ Ω. Téléchargez de Moodle le fichier 'resonanceRLC.txt' qui contient un tableau de mesure : fréquence (en Hz) – amplitude de tension (en V). Lisez ce fichier par Python et présentez les mesures graphiquement. Utilisez la fonction `curve_fit()` de module `scipy.optimize` pour ajuster la dépendance théorique $a_x(\omega_f)$ à l'expérience et ajoutez la courbe théorique à votre graphique expérimental. Comparez les valeurs du fit avec les valeurs théoriques :

$$\omega_0 = 1/\sqrt{LC}, \ \ 2\gamma/\omega_0^2 = RC$$