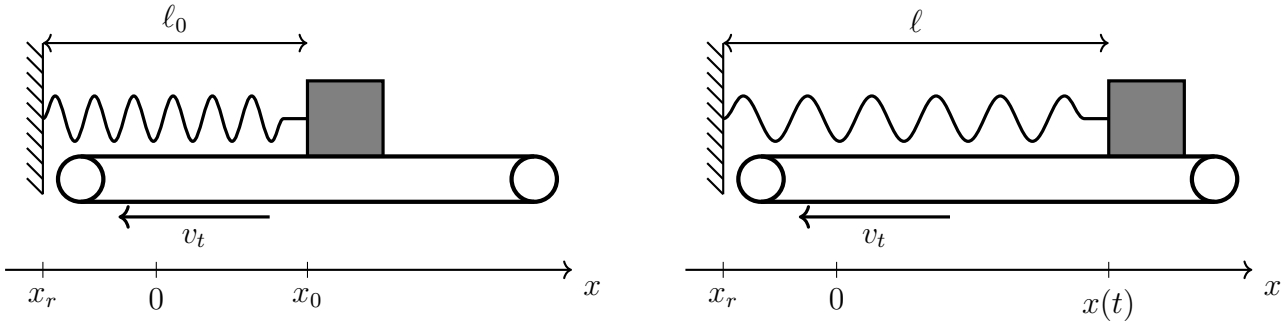
TP IPT n°3
Mouvement de collé-glissé (stick-slip)

On considère un solide parallépipédique de masse m pouvant se déplacer selon l'axe horizontal (Ox). Il est attaché à un ressort de constante de raideur k, de longueur à vide ℓ_0 dont l'autre extrémité est fixée à un support immobile à la position x_r. Le solide est posé sur un tapis roulant se déplacant à la vitesse v_t constante et le contact entre le solide et le plan se fait avec frottements caractérisés par les coefficients statique f_s et dynamique f_d. Initialement, le solide est tel que le ressort est au repos. Le but du TP est de modéliser l'évolution du système.



Le mouvement se compose d'une alternance de deux phases : une première phase dite statique durant laquelle le bloc est immobile par rapport au tapis puis une phase dynamique lors de laquelle le bloc est ramené vers sa potition initiale par le ressort.

1 - Mise en équation

- Exprimer la norme de la force de frottement dans le cas dynamique. En déduire l'équation du mouvement dans ce cas.
- Déterminer la poition x_{max} maximale atteinte par le bloc au cours de la phase statique.

On cherche la solution au problème entre les temps $T_0 = 0$ et T_{max} avec un pas de temps dt sous forme de tableaux dont le premier, `X[]`, représente la position pour chacun de ces temps et le deuxième, `V[]`, la vitesse. Les valeurs pour chaque pas de temps seront ajoutées au fur et à mesure de la simulation. Ces différents tableaux sont initialisés de sorte que $x(t = 0) = x_0 = x_r + \ell_0$ et $v(t = 0) = v_t$ (phase statique). L'ensemble des pas de temps considérés est enregistré dans le tableau `T[]`.

2 - Résolution

- Dans le cas statique, comment déterminer la position et la vitesse à l'instant $t + \mathrm{d}t$ en connaissant celles à l'instant t ?

Pour résoudre l'équation différentielle correspondant au cas dynamique écrite précédemment, on utilise la fonction `odeint` du module `scipy.integrate`. Cette fonction permet de résoudre une équation différentielle de la forme $\dot{y} = f(y, t, ...)$ avec y pouvant être un vecteur. Par exemple, pour résoudre l'équation

$$\ddot{x} + \alpha\dot{x} + \omega^2 x = \gamma$$

on pose un vecteur Y dont les deux composantes sont x et $\dot{x}$. Ainsi cette équation est équivalente à :

$$\dot{Y} = \begin{pmatrix} \dot{x} \\ \ddot{x} \end{pmatrix} = \begin{pmatrix} \dot{x} \\ -\alpha\dot{x} - \omega^2 x + \gamma \end{pmatrix} = f(Y,t)$$

qui peut être résolue par `odeint`. En supposant que les paramètres α, γ et ω sont définis de manière globale dans le code, il faut définir une fonction `Der(Y,t)` qui calcule le vecteur $\dot{Y}$ à partir de l'équation différentielle :

```
def Der(Y,t):
    x,v = Y
    return [ v, -alpha*v - omega**2 * x + gamma]
```

En considérant que le vecteur initial `Y0` est connu, l'équation différentielle se résout par exemple entre les instants 0 et 10 avec 5 pas de temps avec le code :

```
T = np.linspace(0, 10, 5)
sol = odeint(Der, Y0, T)
```

la solution `sol` étant ici une matrice $(5, 2)$. Les deux colonnes représentent respectivement la position et la vitesse pour les différents pas de temps.

- Écrire la fonction dont l'entête est

  ```
  def Der(Y:np.array,t:np.array) -> np.array:
  ```

  qui réalise la dérivation du vecteur `Y` par rapport au temps `t` d'après l'équation différentielle qui nous occupe et renvoie le tableau correspondant.
- Coder la boucle temporelle permettant de résoudre le problème.
- Tracer l'évolution de la position du solide au cours du temps ainsi que le portrait de phase.

3 - Bonus

- Ce modèle permet d'expliquer le principe de l'archer sur une codre de violon. Quel lien peut-on faire entre le déplacement du solide (de la corde) et le son émis ? Tracer le spectre de ce son. Pourquoi le son d'un violon est-il qualifié de riche ?
- En adaptant votre code, tracer l'évolution du système de l'exercice du **DM11**.