



# Effet Ramsauer-Townsend

**Pré-ing 2 - MI5-C**  
**Physique Moderne**  
**Référent : M. AKRIDAS**

**13 juin 2025**

**Réalisé par :**

**Ziyad HADDADI**

**Romain MICHAUT-JOYEUX**

**Nathan CHOUPIN**

# Section efficace

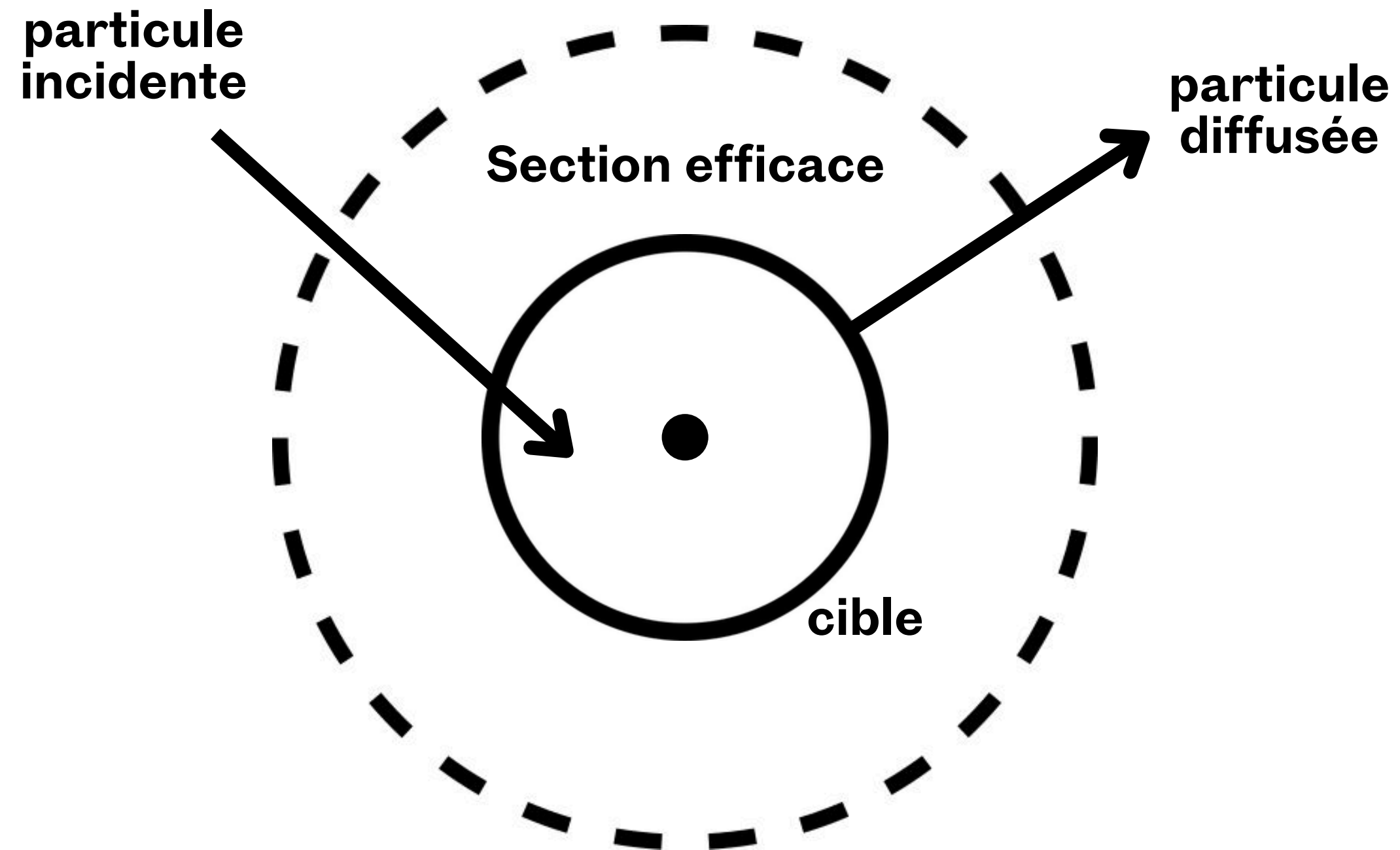
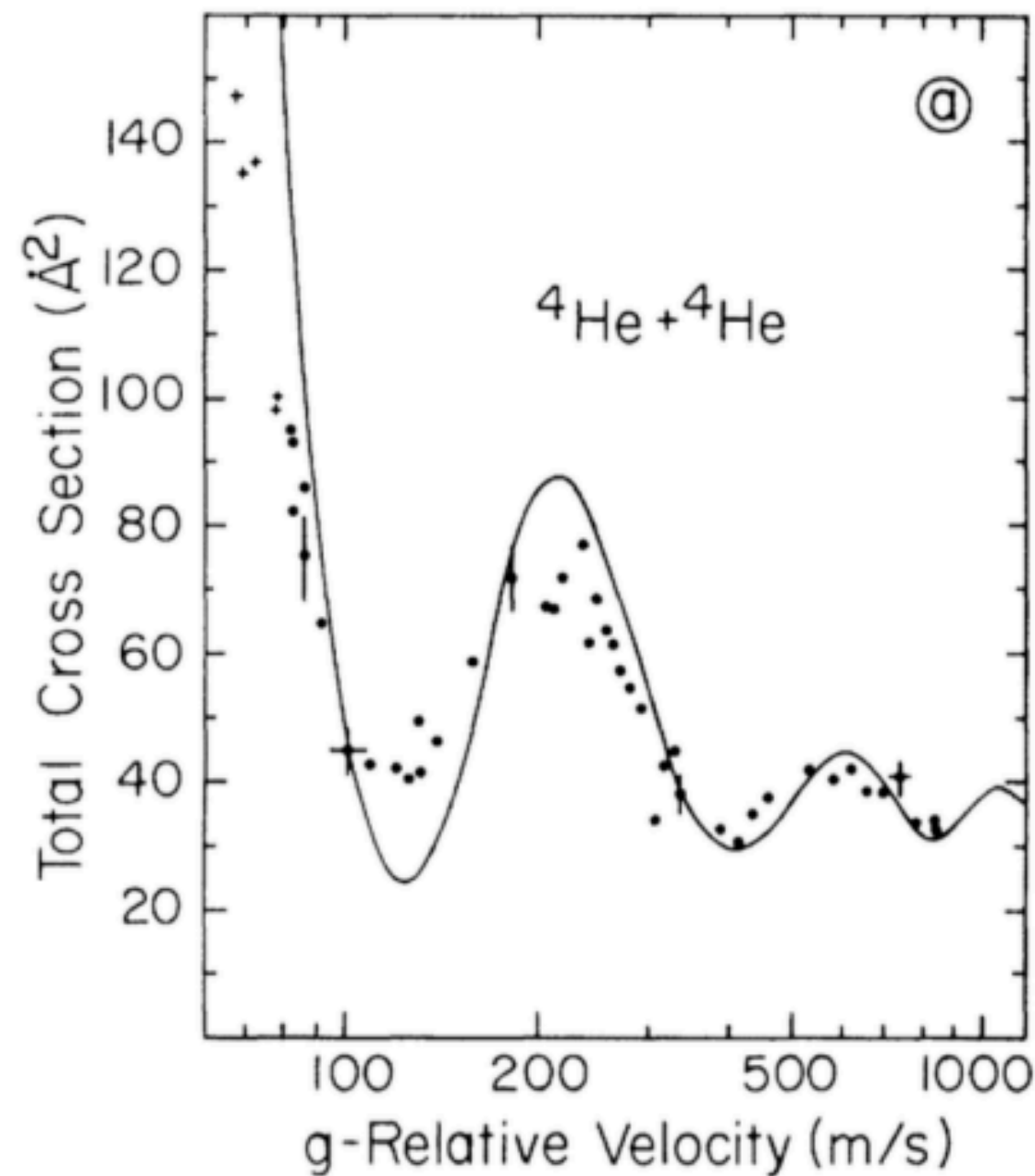


Schéma de la section efficace, en 2D,  
réalisé sur Canva.

# Effet Ramsauer-Townsend



**Observation de l'effet Ramsauer–Townsend lors de la collision d'atomes  $^4\text{He}$ , Robert S, 1979.**

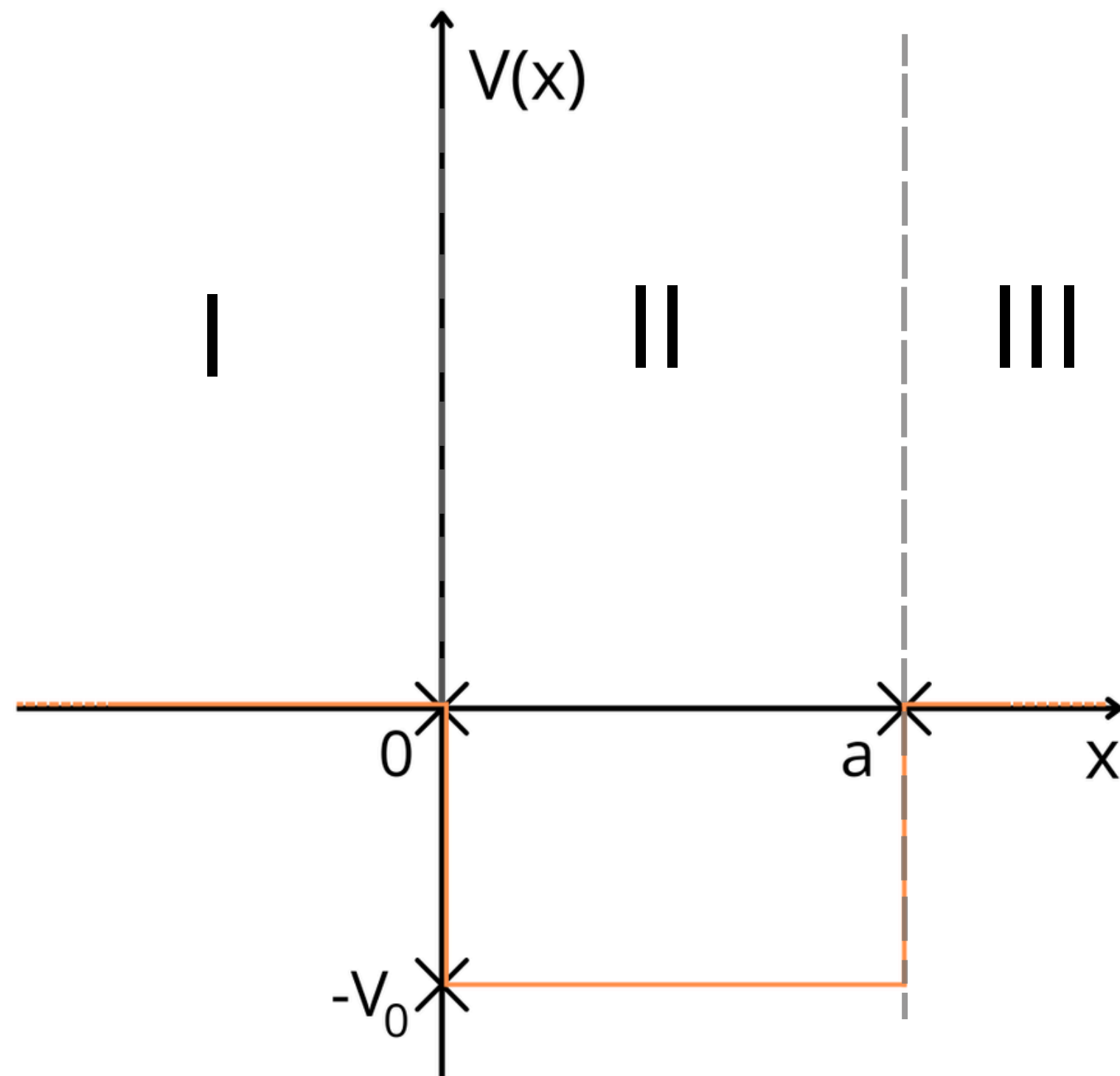


**Éclairage au xénon à partir d'un tube flash, Alex Zhang, 2020.**

# Objectifs

- **Comprendre l'effet**
- **Comparer un modèle 1D à un modèle 3D**
- **Déterminer les limites du modèle**

# Puits fini de potentiel



Électron de masse  $m$  et d'énergie  $E > 0$

$$V(x) = \begin{cases} -V_0 & \text{si } 0 \leq x \leq a \\ 0 & \text{sinon} \end{cases}$$

Potentiel (J) en fonction de la position (m),  
réalisé avec Canva.

# Partie analytique

Équation de Schrödinger indép. du temps, en une dimension :

$$\frac{-\hbar^2}{2m} \cdot \frac{d^2 \psi(x)}{dx^2} + V(x) \cdot \psi(x) = E \cdot \psi(x)$$

En région I,  $x < 0, V(x) = 0$

On déduit

$\psi_1(x) = A e^{ik_1 x} + B e^{-ik_1 x}$ , où  $A e^{ik_1 x}$  est l'onde incidente,  $B e^{-ik_1 x}$  l'onde

réfléchie, et  $k_1 = \sqrt{\frac{2mE}{\hbar^2}}$

# Partie analytique

En région II,  $x \in [0, a], V(x) = -V_0$

On déduit

$$\psi_2(x) = C e^{ik_2 x} + D e^{-ik_2 x}, \quad k_2 = \sqrt{\frac{2m(E + V_0)}{\hbar^2}}$$

En région III,  $x > a, V(x) = 0$

On déduit

$$\psi_3(x) = F e^{ik_3 x} \text{ où } \psi_3 \text{ est l'onde transmise vers } +x, \quad k_3 = \sqrt{\frac{2mE}{\hbar^2}}$$

# Partie analytique

Les conditions de continuité aux bords du puits sont :

$$(S): \begin{cases} \psi_1(0) = \psi_2(0) \\ \psi_1'(0) = \psi_2'(0) \\ \psi_2(a) = \psi_3(a) \\ \psi_2'(a) = \psi_3'(a) \end{cases}$$

Équivalent à :

$$(S) \Leftrightarrow \begin{cases} A+B = C+D & (1) \\ k(A-B) = q(C-D) & (2) \\ C e^{iqa} + D e^{-iqa} = F e^{ika} & (3) \\ q(C e^{iqa} - D e^{-iqa}) = k F e^{ika} & (4) \end{cases}, \text{ avec } k = k_1 = k_3 \text{ et } q = k_2$$



# Partie analytique

Après résolution :

$$T = \frac{|F|^2}{|A|^2} = \frac{4 k^2 q^2}{4 k^2 q^2 \cos^2(qa) + (k^2 + q^2)^2 \sin^2(qa)}$$

En injectant  $k$  et  $q$  :

$$T = \left( 1 + \frac{V_0^2}{4 E (E + V_0)} \cdot \sin^2 \left( \sqrt{(2 m (E + V_0))} \frac{a}{\hbar} \right) \right)^{-1}$$

# Résultats

Nous avons :

$T$  coef. de transmission

$R$  coef. de réflexion

$$T + R = 1$$

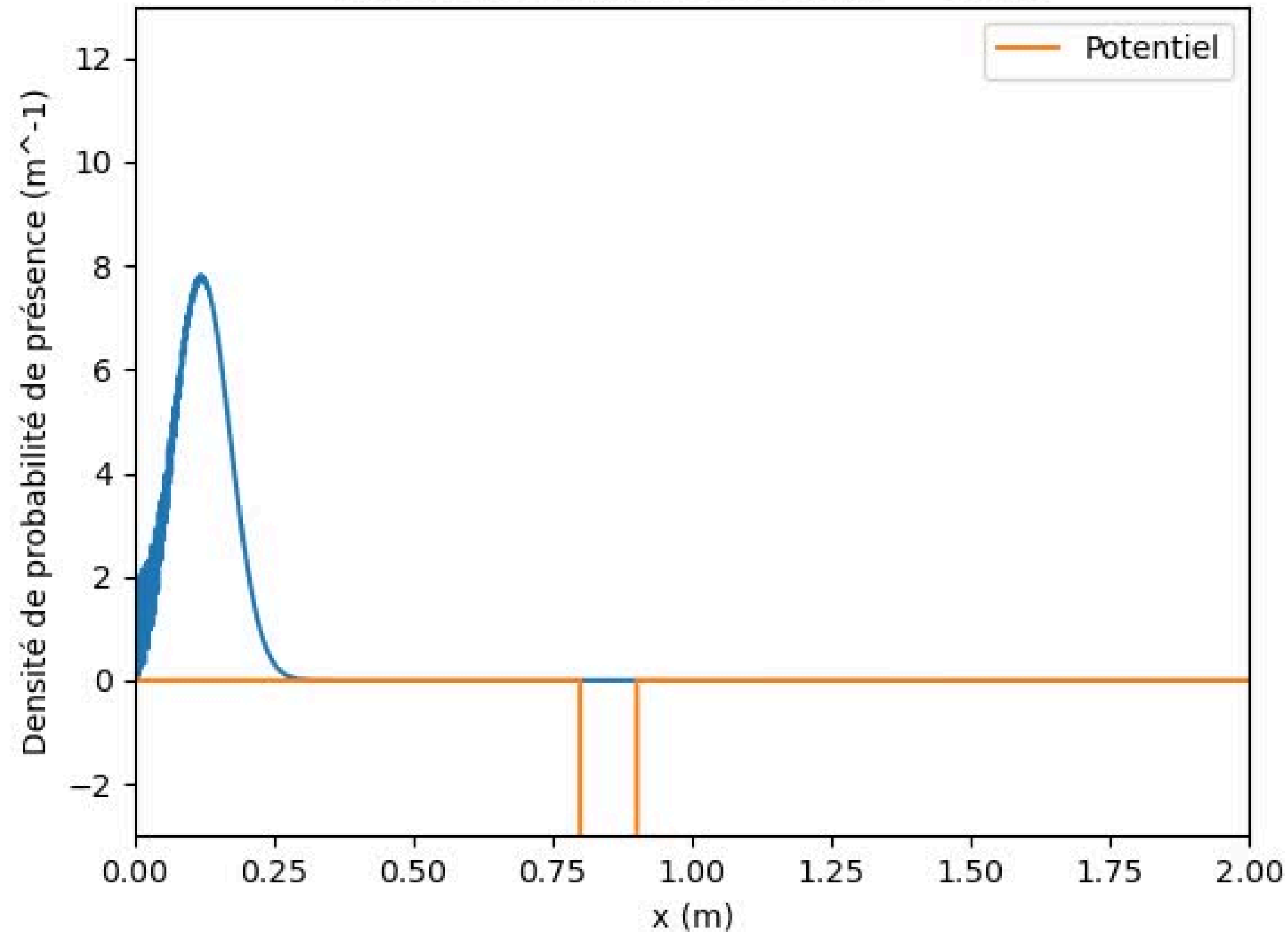
$$T = 1 \Rightarrow \sin^2(qa) = 0 \Rightarrow \sqrt{\frac{2m(E + V_0)}{\hbar^2}} a = n\pi, \quad n \in \mathbb{N}$$

# Interprétation

- **Effet pour des énergies précises**
- **Interférences**
- **Contredit la physique classique**

# Observations

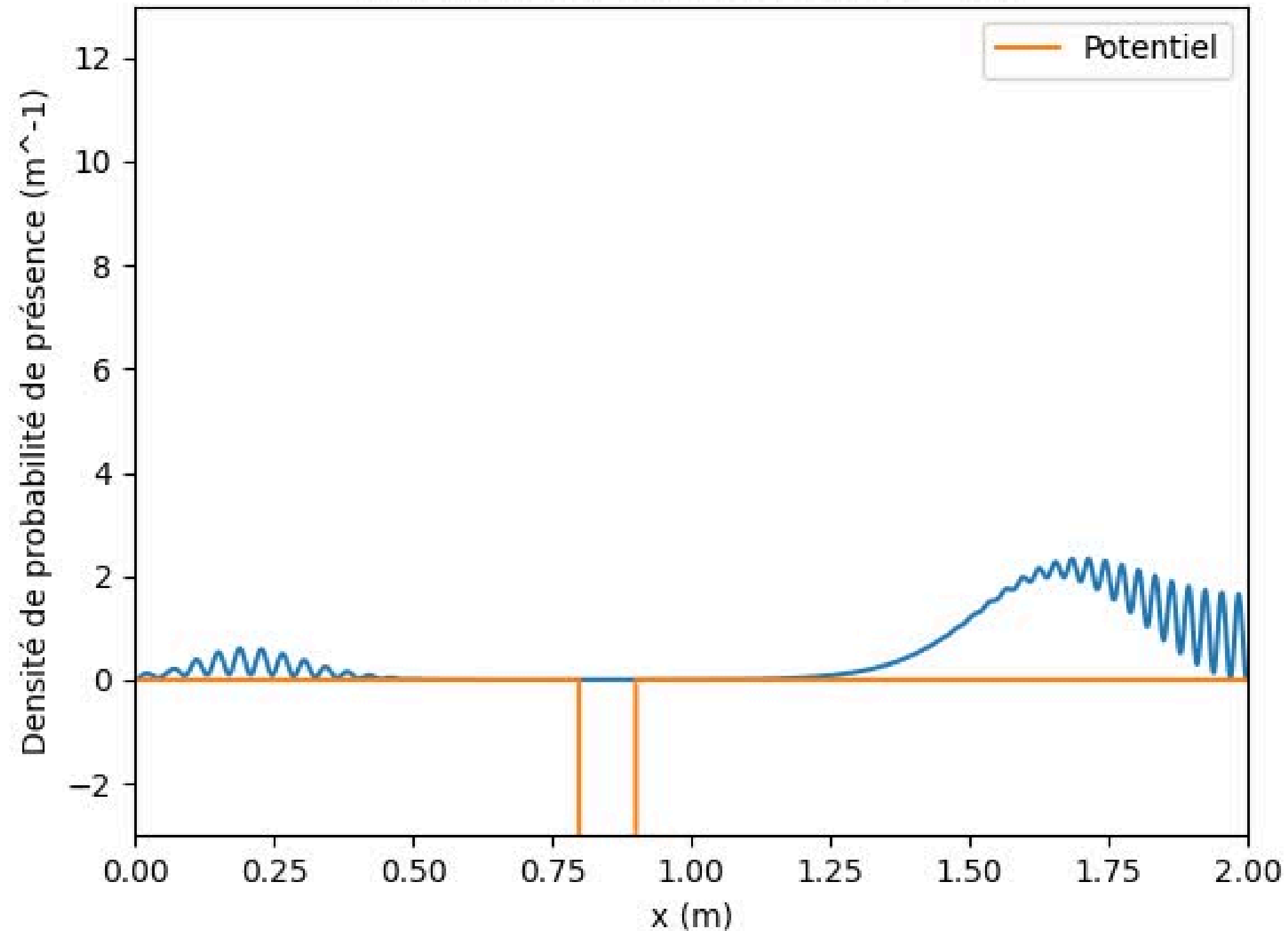
Marche Ascendante avec  $E/V_0 = 100.0$



**Modélisation de la propagation d'un paquet d'ondes, en Python, avec  $E/v_0 = 100$ .**

# Observations

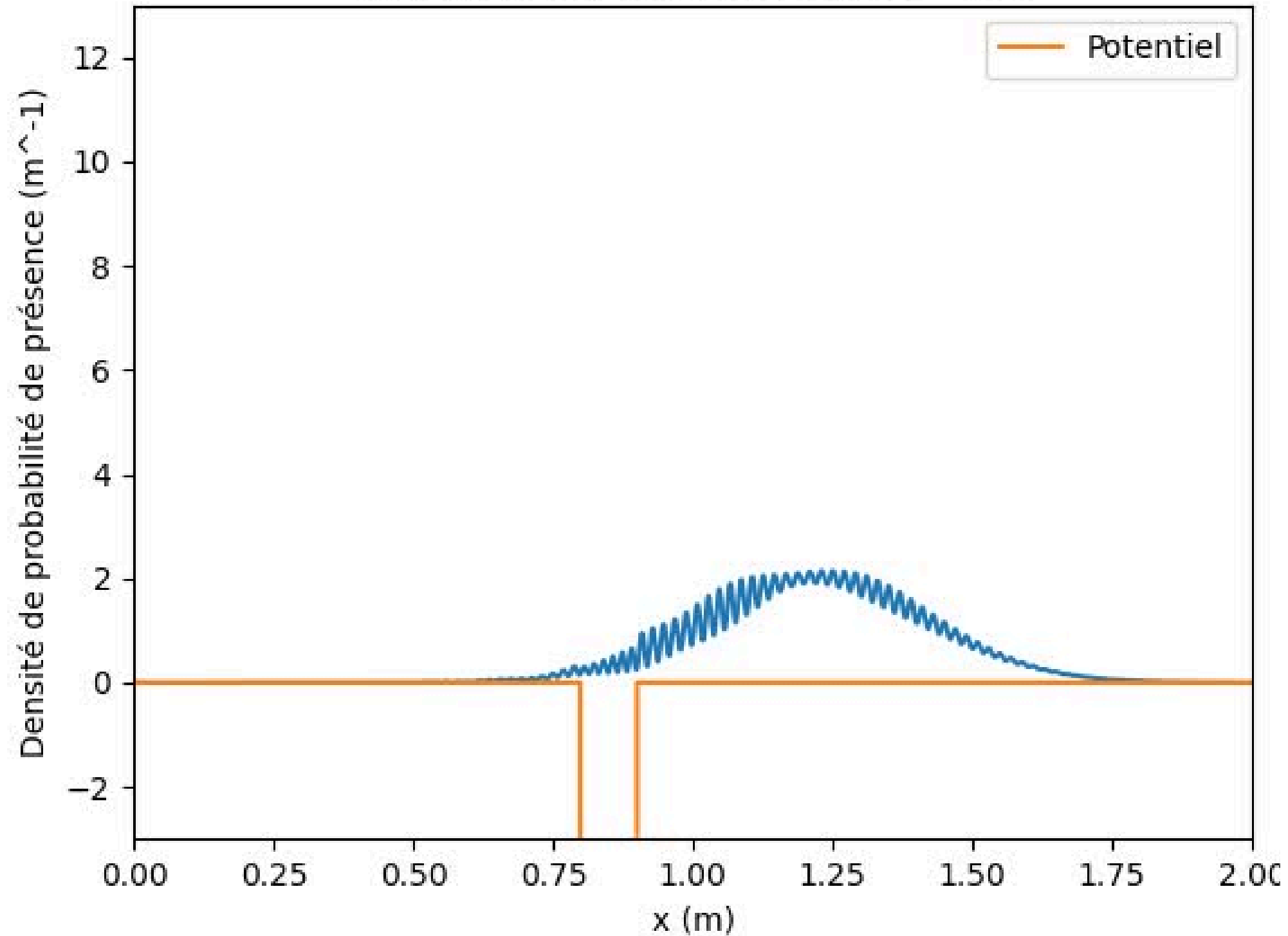
Marche Ascendante avec  $E/V_0 = 0.8$



**Modélisation de la propagation d'un paquet d'ondes, en Python, avec  $E/v_0 = 0.8$ .**

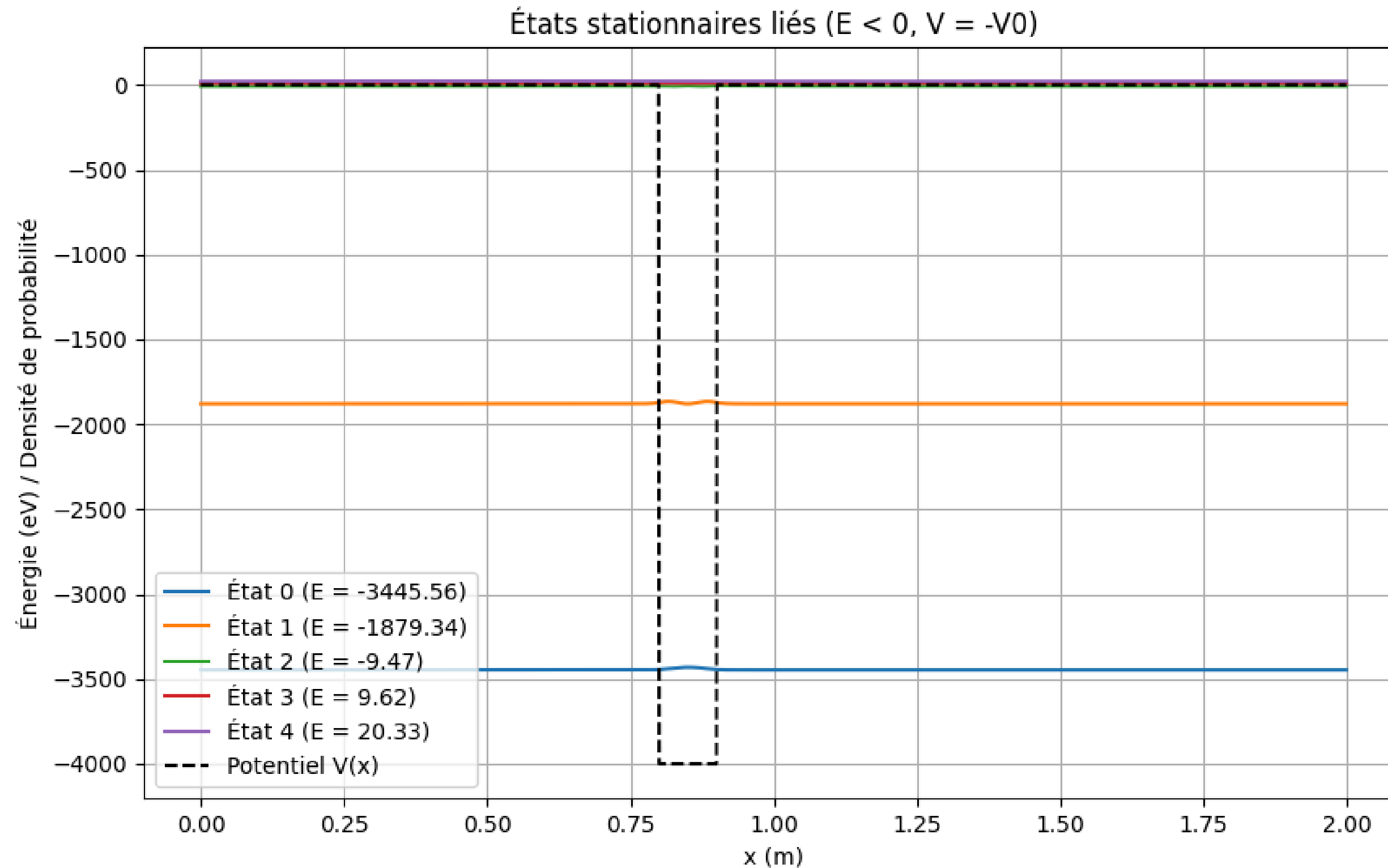
# Observations

Marche Ascendante avec  $E/V_0 = 3.0$



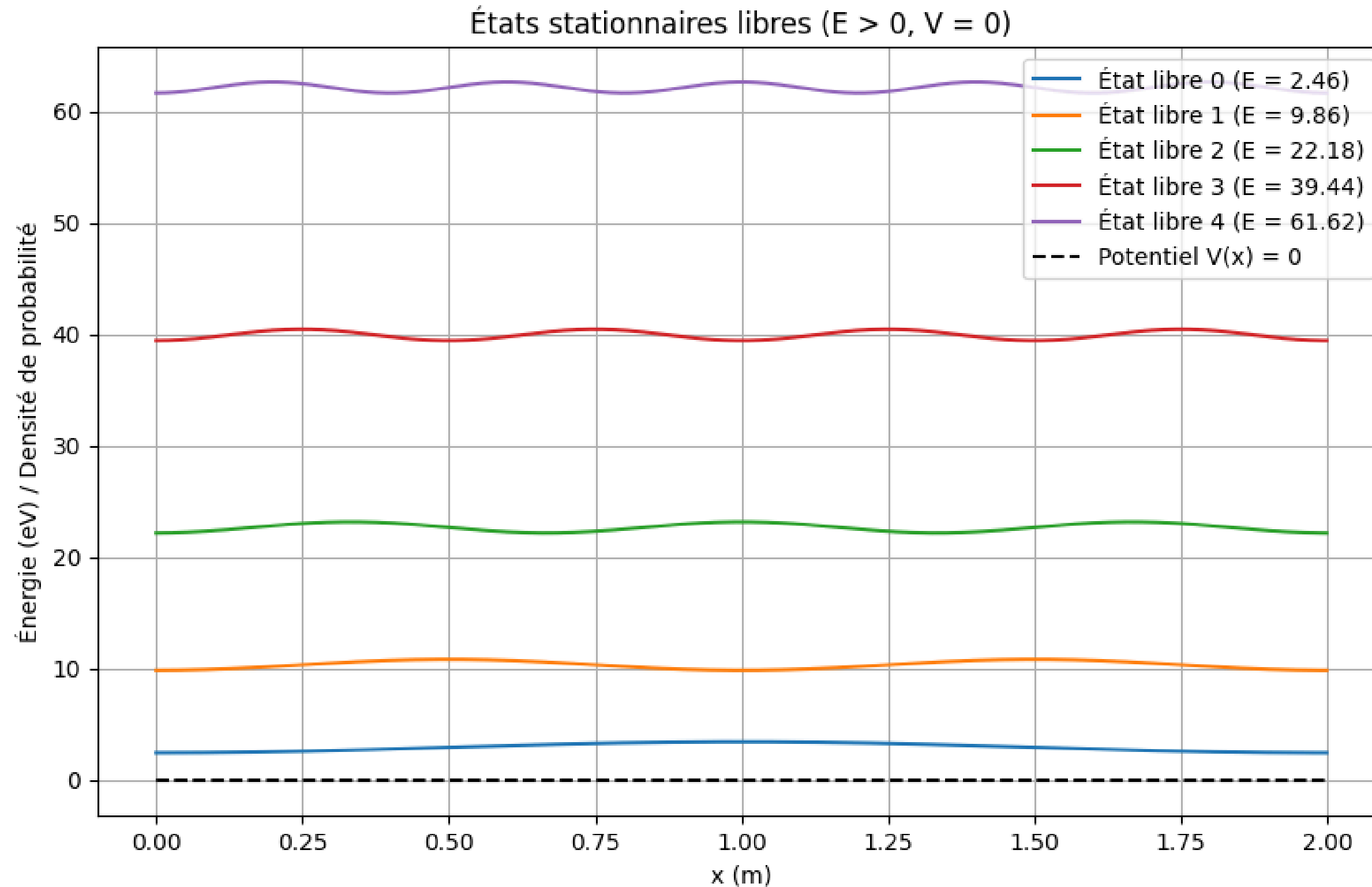
**Modélisation de la propagation d'un paquet d'ondes, en Python, avec  $E/v_0 = 3$ .**

# Observations



**États stationnaires liés (région II) modélisés en Python.**

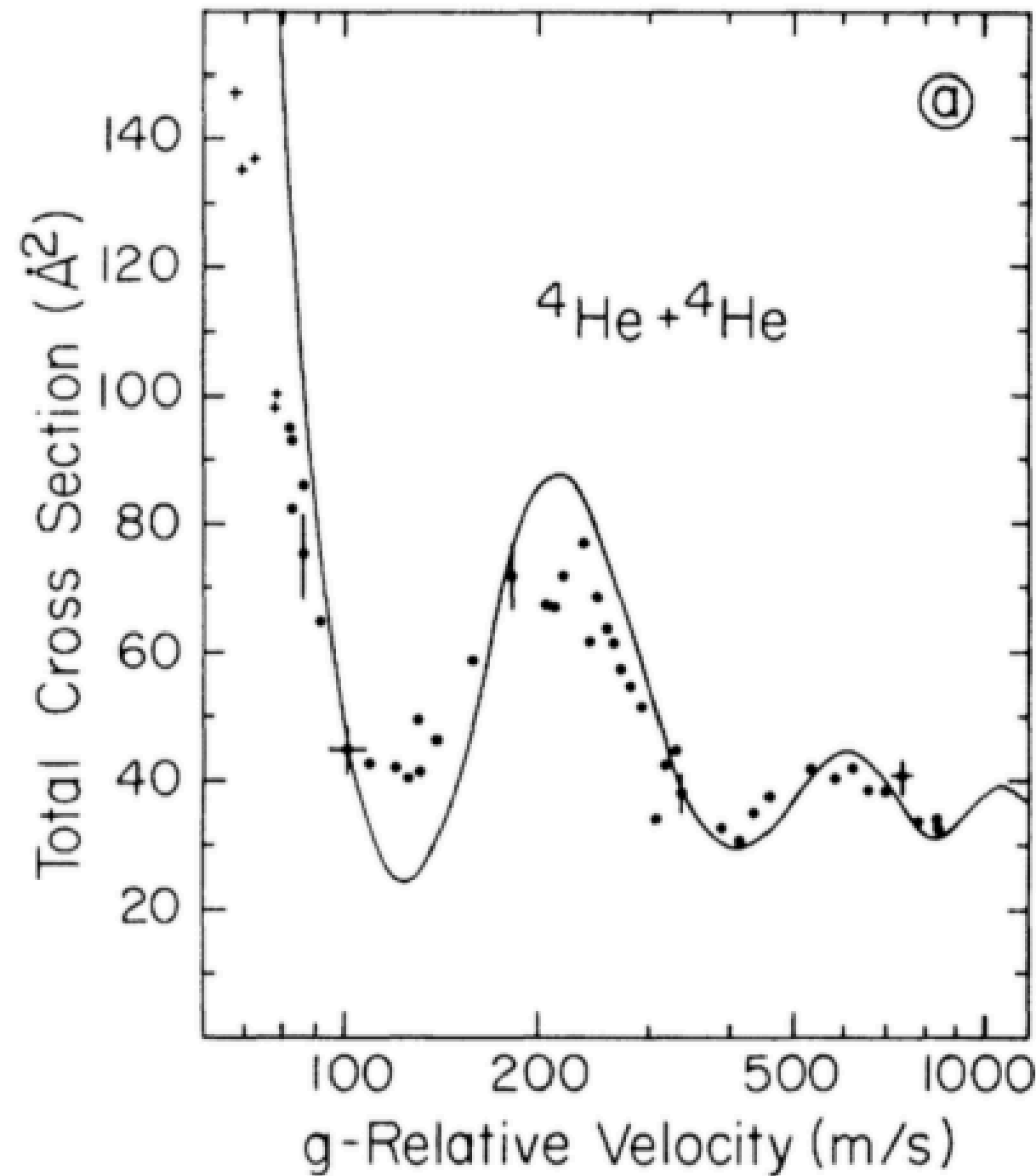
# Observations



**États stationnaires libres (région I, III) modélisés en Python.**

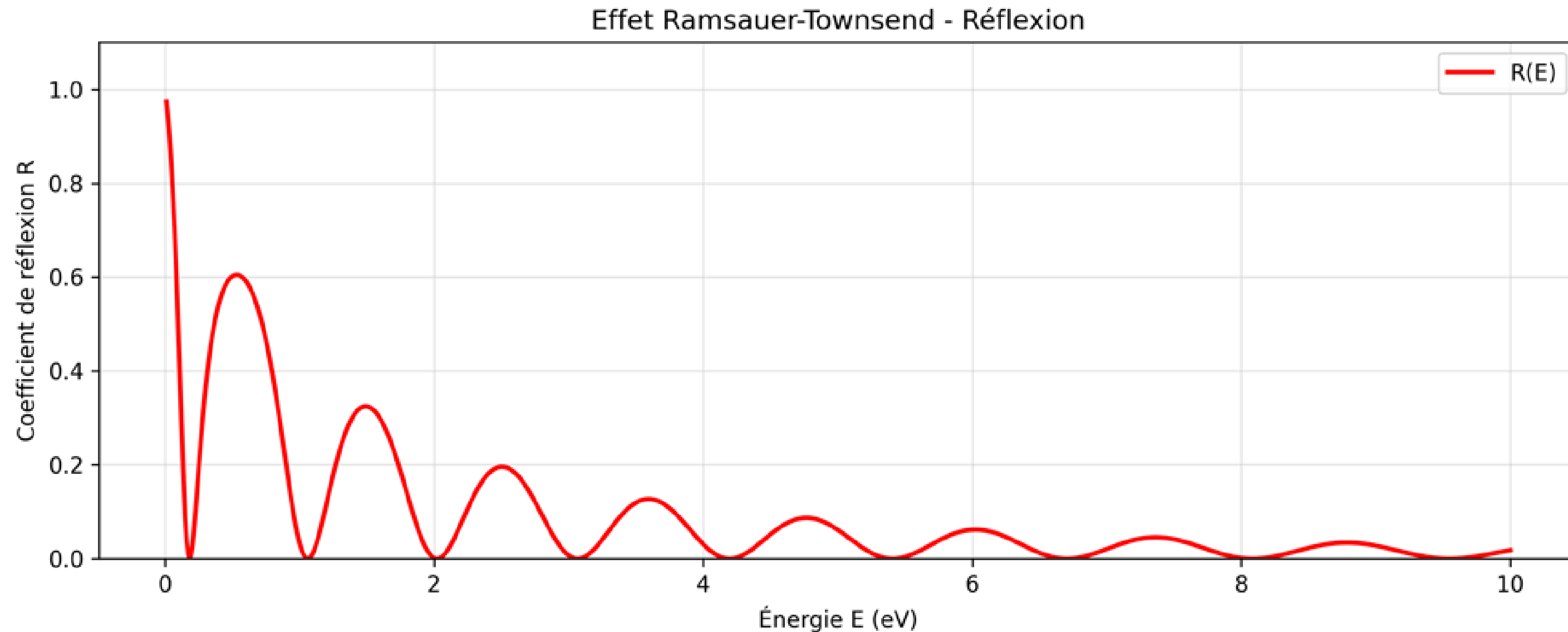


# Mesures expérimentales



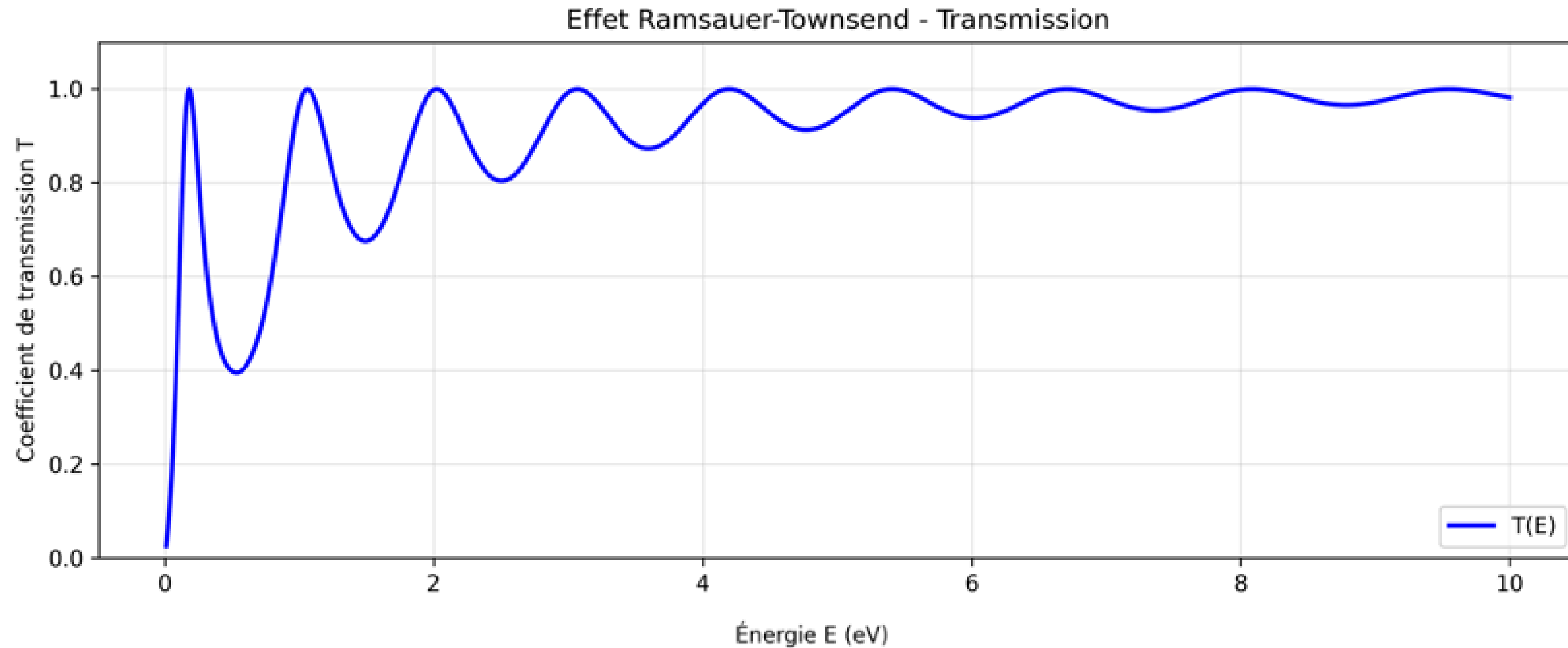
**Observation de l'effet Ramsauer–Townsend lors de la collision d'atomes  $^4\text{He}$ , Robert S, 1979.**

# Prédictions numériques



**Coef. de réflexion (sans unité) en fonction de l'énergie d'un électron (ev), modélisé en Python.**

# Prédictions numériques



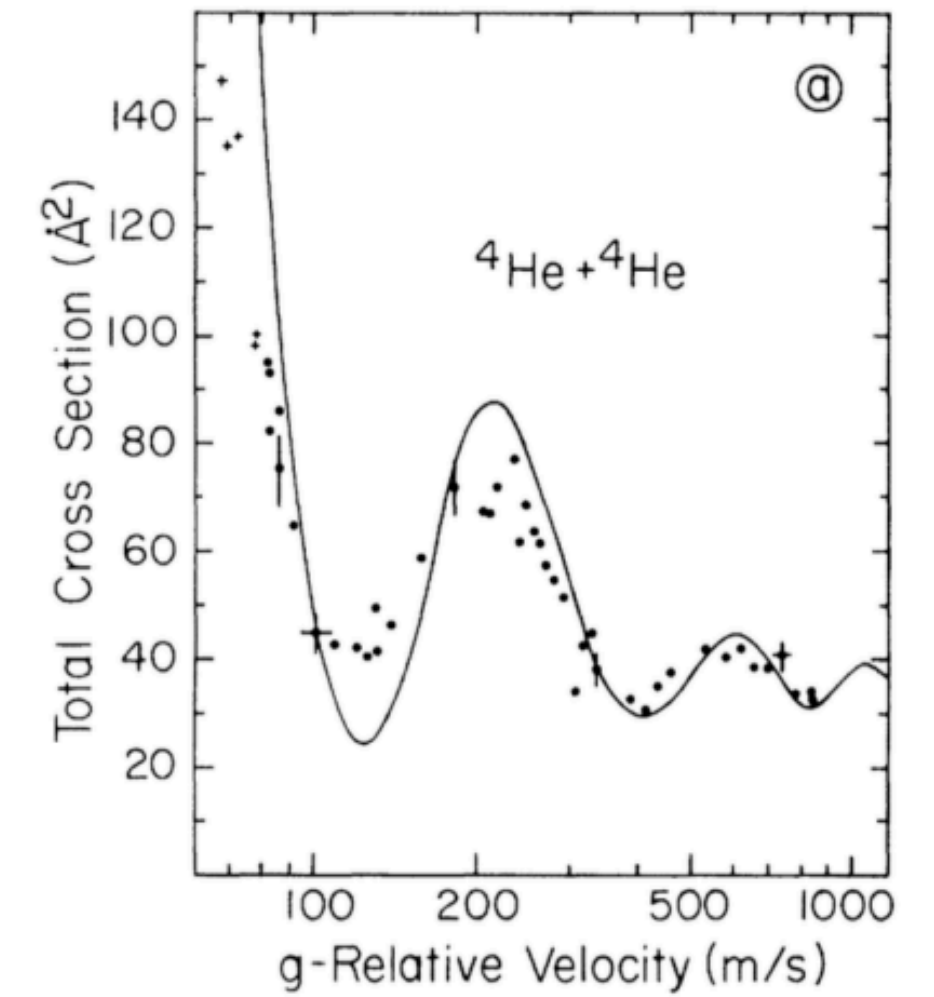
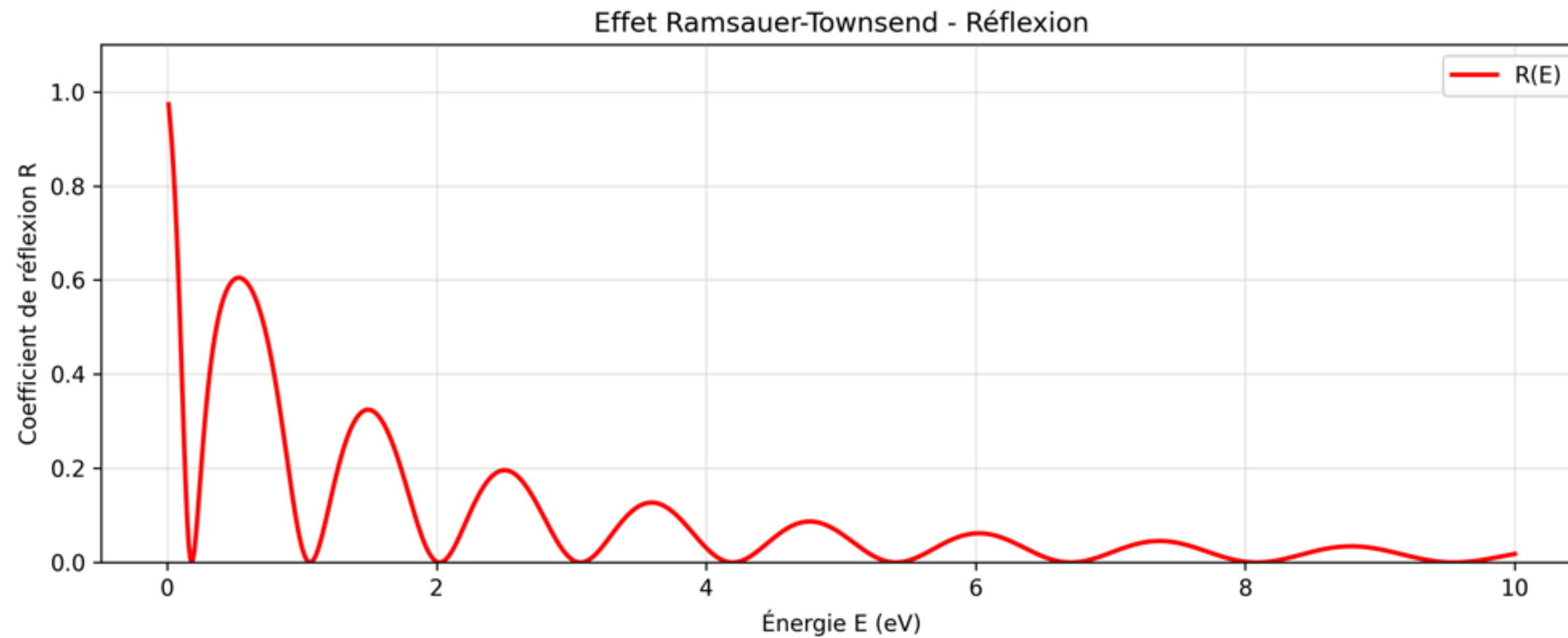
**Coef. de transmission (sans unité) en fonction de l'énergie d'un électron (ev), modélisé en Python.**

# Prédictions numériques

```
def ramsauer_townsend():  
    # Paramètres  
    v0 = 4 # Pontentiel  
    m = 9.11e-31 # Masse de l'électron  
    a = 3e-9 # Largeur du puits (nm)  
    hbar = 1.055e-34  
    eV = 1.602e-19  
  
    # Plage d'énergies  
    E_eV = np.linspace(0.01, 10, 1000)  
  
    # Calcul des coef.  
    T = np.zeros_like(E_eV)  
    R = np.zeros_like(E_eV)  
  
    for i, E_ev in enumerate(E_eV):  
        E = E_ev * eV # Conversion en Joules  
  
        q = np.sqrt(2 * m * (E + v0 * eV)) / hbar  
        terme_sin = np.sin(q * a)**2  
  
        # On utilise la formule trouvée dans l'étude analytique  
        T[i] = 1 / (1 + (v0**2) / (4 * E_ev * (E_ev + v0)) * terme_sin)  
        R[i] = 1 - T[i]
```

**Paramètres et calculs utilisés dans la prédiction,  
en Python**

# Comparaison



# Similitudes

- **Oscillations caractéristiques et amplitudes décroissantes**
- **Minimas de diffusion**
- **R dépend de l'énergie**

# Différences

- **En 1D, une trajectoire  $\Rightarrow$  oscillations plus régulières**
- **Minimas proches de 0 en 1D**

# **Limites du modèle**

- **Ne modélise pas les interactions en 3D.**
- **Néglige les forces de Van der Waals (interactions faibles d'attraction entre atomes neutres).**
- **Néglige l'angle de diffusion.**
- **Paquets d'ondes plus réalistes**



# Un modèle plus réaliste

Le potentiel de Lennard-Jones :

$$V(r) = 4\epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right]$$

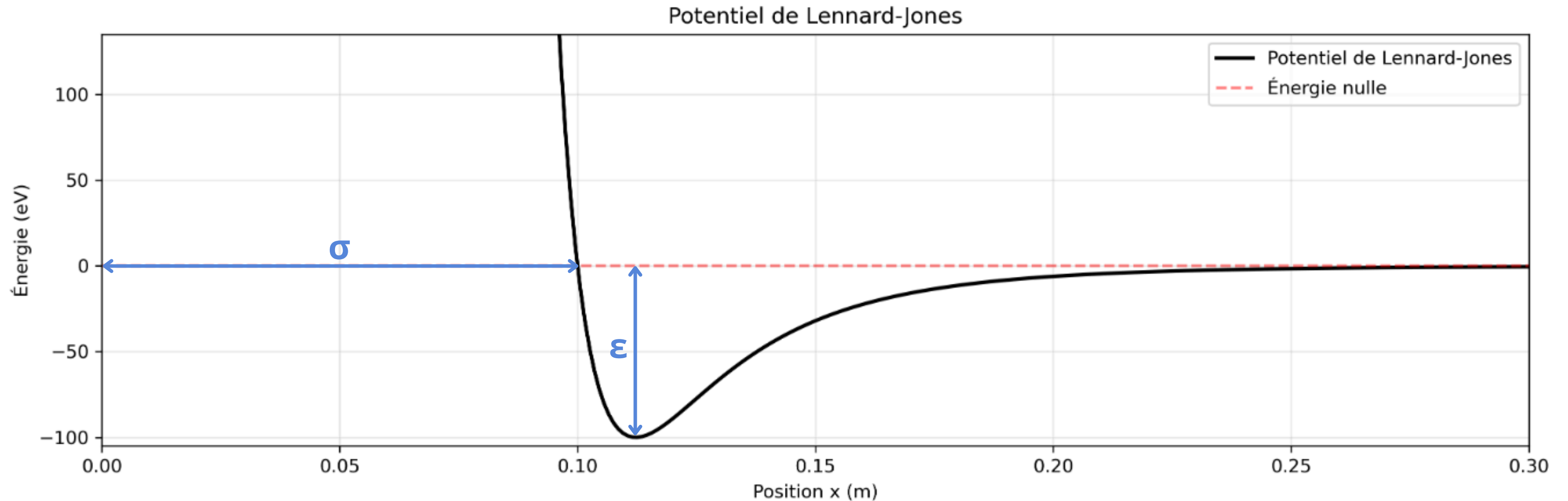
où :

- $r$  est la distance entre les deux atomes,
- $\epsilon$  est la profondeur du puits de potentiel, qui correspond à l'énergie minimale lorsque les deux atomes sont à la distance d'équilibre,
- $\sigma$  est la distance à laquelle le potentiel s'annule.

# Lennard-Jones

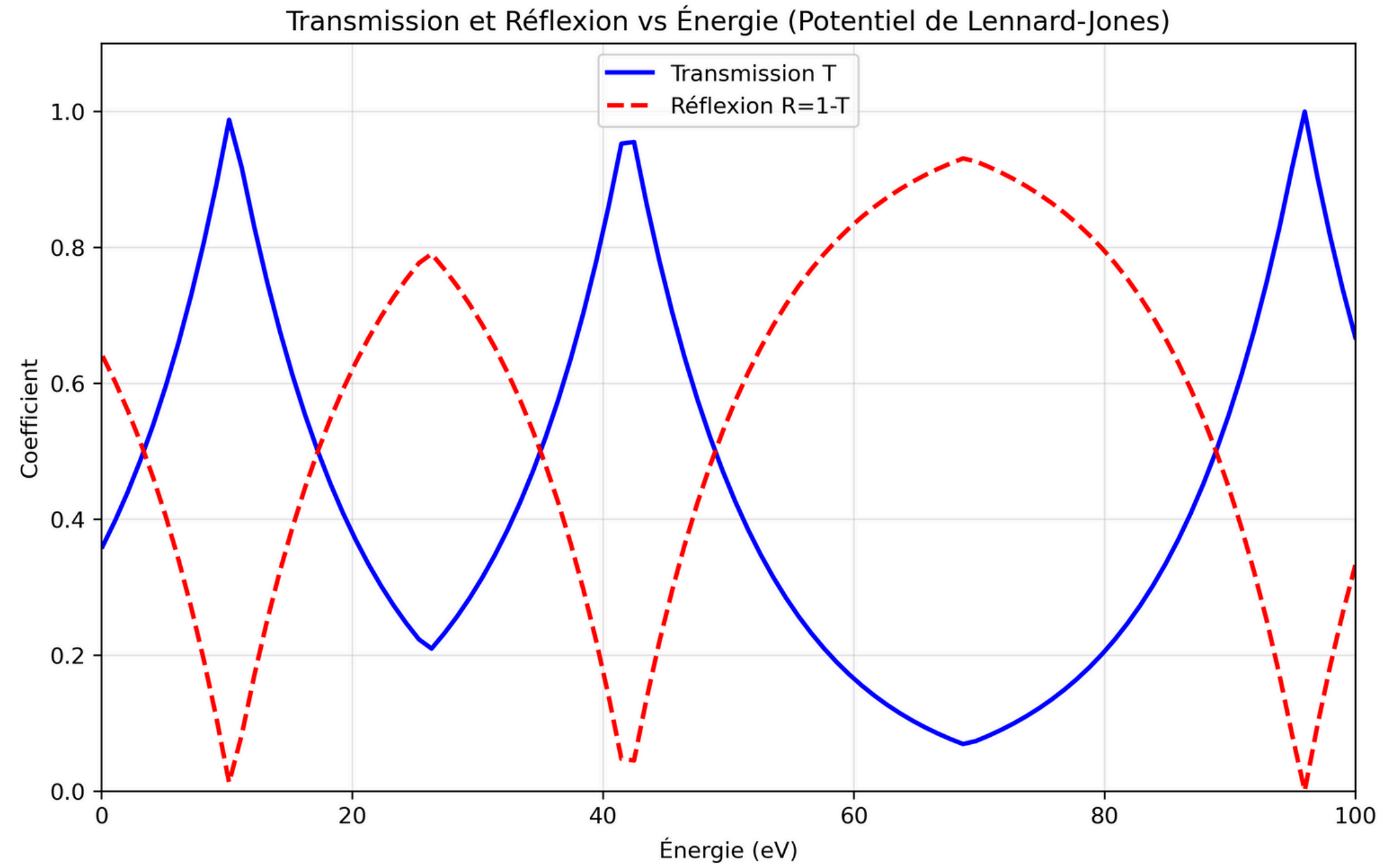
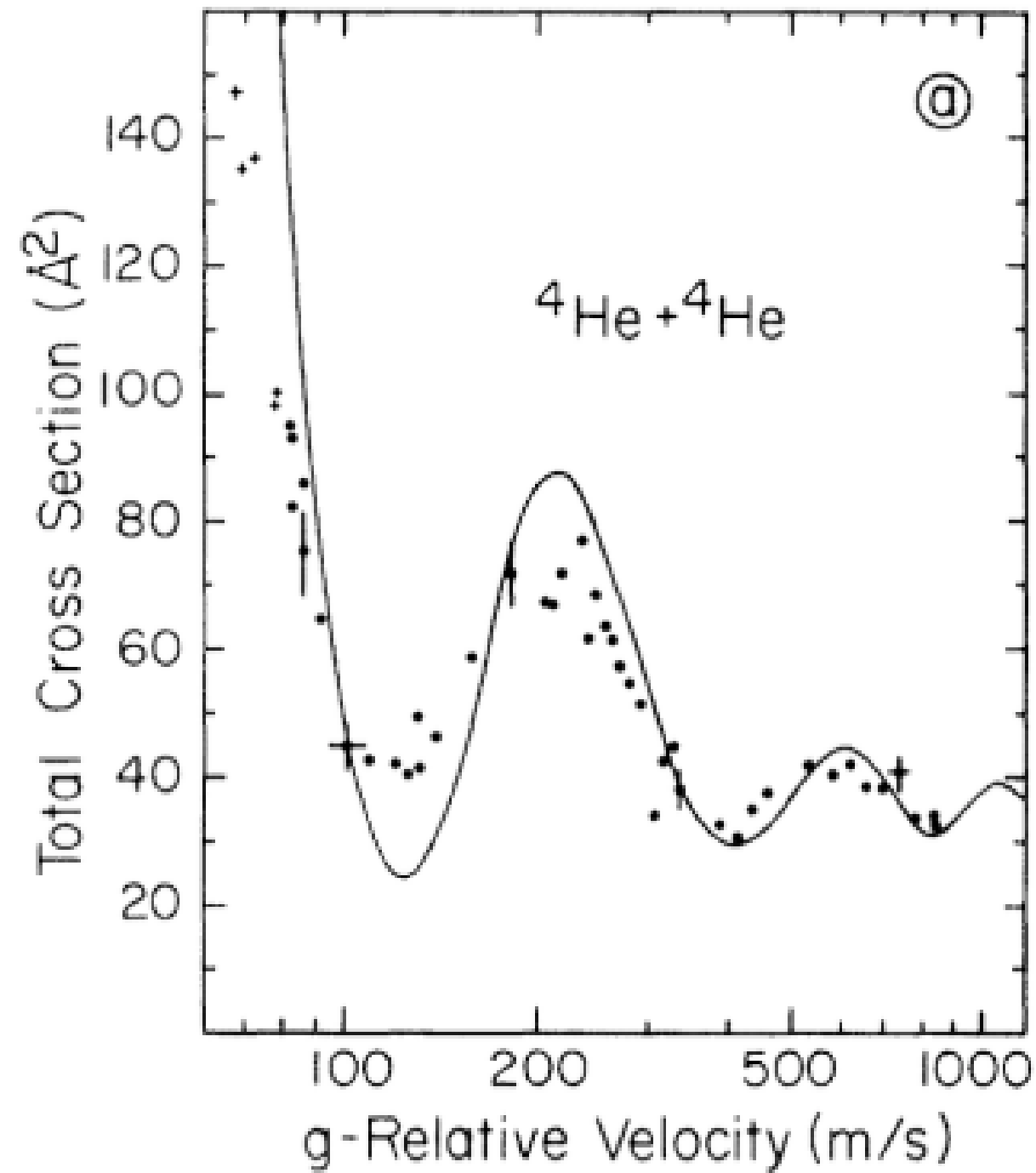
- **Potentiel continu avec attraction/répulsion (forme  $r^{-6}$  et  $r^{-12}$ ).**
- **Prend en compte la physique réelle des interactions atomiques.**
- **Explique la cohésion des gaz nobles et les zones de «transparence».**
- **Plus proche des expériences sur l'effet Ramsauer–Townsend.**

# Lennard-Jones



**Énergie (eV) en fonction de la position (m), en utilisant le modèle de Lennard-Jones, modélisé en Python.**

# Lennard-Jones



**Coef. de transmission et réflexion en fonction de l'énergie (eV), modélisé en Python.**

# Conclusion