

TP 17.2 Simulation de la propagation d'une onde - python

On fait osciller continuellement l'extrémité d'une corde afin d'y propager une onde transversale.

On souhaite simuler la propagation d'une onde transversale verticale le long d'une corde horizontale sur une distance $D = 1$ m.

La longueur d'onde λ de l'onde doit être inférieure à $\frac{D}{2}$ et sa fréquence comprise entre 20 Hz et 100 Hz.

L'amplitude Y_m de l'onde est égale à $\frac{D}{10}$

La durée de la simulation doit être de 5 périodes et on représente un point tous les mm.

Pour chaque point $M(x, y)$ de la corde :

$$y(x, t) = Y_m \times \left(1 + \sin\left(\frac{2\pi \times t}{T} - \frac{2\pi \times x}{\lambda} - \frac{\pi}{2}\right) \right)$$

Les grandeurs Y_m et y sont exprimées dans la même unité. On choisira ici le mm.

Les grandeurs x et λ sont exprimées dans la même unité. On choisira ici le mm.

Les grandeurs t et T sont exprimées dans la même unité. On choisira ici la ms.



Calculer la valeur minimale et la valeur maximale de la période à utiliser dans la simulation.

À l'aide des vidéos ci-dessous et de la fiche méthode 14 p 331-334 du livre de 2nde, réaliser les différentes étapes d'un programme en langage python permettant de simuler la propagation de l'onde dans la corde.

Vous nommerez ce programme « SimulationCorde.py ».

<https://video.crdp.ac-versailles.fr/scolawebtv/4/4/1/41441.mp4>

<https://video.crdp.ac-versailles.fr/scolawebtv/4/4/2/41442.mp4>

<https://video.crdp.ac-versailles.fr/scolawebtv/4/4/3/41443.mp4>

<https://video.crdp.ac-versailles.fr/scolawebtv/4/4/4/41444.mp4>

Les variables à utiliser dans le programme sont écrites **en gras** dans les consignes.

Tester votre programme à la fin de chaque étape. Il doit être fonctionnel avant de passer à l'étape suivante.

Noter sur votre compte-rendu vos observations à la fin de chaque étape.

Étape 1 :

Importer le module pyplot de la bibliothèque matplotlib.

Importer tous les modules de la bibliothèque math.

Initialiser les variables **D**, **LongueurOnde**, **T** et **Ym**.

À l'aide de la fonction « print » afficher les valeurs de ces variables dans la console python.

Étape 2 :

Créer les listes vides **list_x** et **list_y**.

Dans une boucle « for » faisant varier l'indice **x** : (à vous de déterminer les valeurs limites de ce compteur)

À l'aide de la fonction « append », ajouter la valeur de **x** à la liste **list_x**

Calculer la variable **y** à l'aide de la formule donnée dans l'énoncé en considérant pour l'instant que $t=0$.

À l'aide de la fonction « append », ajouter la valeur de **y** à la liste **list_y**

Après la boucle « for », afficher les valeurs de **list_x** et **list_y** dans la console python à l'aide de la fonction « print ».

Étape 3 :

À l'aide de la fonction « pyplot.scatter », définir un graphique (nuage de points) à partir de **list_x** et **list_y**.

À l'aide de la fonction « pyplot.show », afficher le graphique.

Étape 4 :

A l'aide de la fonction « `pyplot.axis` », définir les valeurs minimales et maximales sur chaque axe.

A l'aide des fonctions « `pyplot.title` », « `pyplot.xlabel` » et « `pyplot.ylabel` », définir les légendes du graphique.

Attention : les fonctions précédente doivent se placer avant la fonction « `pyplot.show` »

Étape 5 :

Ajouter au programme une boucle « `for` » faisant varier l'indice **t** : (à vous de déterminer les valeurs limites de ce compteur)

Cette boucle « `for` » doit contenir les étapes 2, 3 et 4, sauf la fonction « `pyplot.show` » et la formule calculant la variable **y** doit être à présent être exprimée en fonction de **t** (**t** n'est plus égal à 0)

Étape 6 :

A l'aide de la fonction « `pyplot.pause(0.001)` », définir une pause de 1 ms avant la définition de chaque graphique (nuage de point)

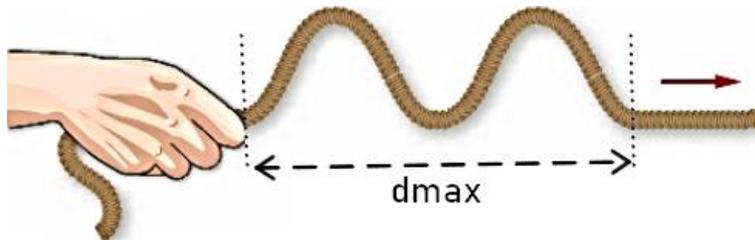
Étape 7 :

A l'aide de la fonction « `pyplot.clf()` », effacer la mémoire graphique avant la définition de chaque graphique (nuage de point)

Pour aller plus loin :

La longueur d'onde correspond à la distance parcourue par l'onde pendant une durée égale à T .

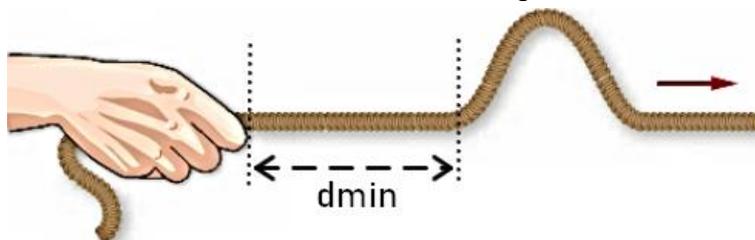
Au début de la propagation, la corde doit rester à plat tant que l'onde ne s'y est pas encore propagée.



Exprimer la distance d_{max} parcourue par l'onde pendant une durée t en fonction T et λ

A l'aide d'une condition « `si... alors... sinon...` », modifier le programme pour que **y** soit égal à 0 pour les valeurs de **x** supérieures à la distance **dmax** parcourue par l'onde à chaque instant **t**.

On fait à présent osciller l'extrémité de la corde une seule fois, pendant une durée égale à T .



Exprimer la distance d_{min} en fonction d_{max} et λ

A l'aide d'une condition « `ou` » (fonction « `or` »), modifier le programme pour que **y** soit égal à 0 pour les valeurs de **x** supérieures à la distance **dmax** ou inférieure à **dmin** à chaque instant **t**.