

Traitement de données : drainage de bulles de savon

15 octobre 2015

Les données seront traitées grâce à Python. Ce langage de programmation permet à la fois de traiter les données (manipuler des colonnes de données, comparer à des modèles...) et de tracer des courbes.

Vous disposez d'un fichier intitulé `donnees.txt` qui contient deux colonnes de chiffres correspondant respectivement au temps en ms et à l'épaisseur mesurée en nm.

Vous allez rédiger un fichier `NomFichier.py` qui va vous permettre d'effectuer ces opérations. Pour pouvoir ré-utiliser ce fichier ou qu'il soit utilisé par quelqu'un d'autre, pensez à commenter les différentes étapes de votre travail. Vos commentaires doivent simplement être précédés de la commande `#`.

1 Importer les données

La manipulation des données nécessite la création d'un objet numpy. On utilise ensuite la fonction `loadtext` de numpy. Effectuez les opérations suivantes :

1. Importer le tableau avec les données expérimentales
2. Afficher le tableau et son profil.
3. Créer un vecteur "temps" et un vecteur "epaisseur" contenant chacun une colonne des données correspondantes.
4. Manipuler ces deux vecteurs pour que les valeurs soient en unités du système international.
5. Représenter ces données sur un graphique en n'oubliant pas d'indiquer le titre et le libellé des axes.

2 Tracer les données, choisir le modèle

2.1 Créer les vecteurs à tracer

1. Quels sont les deux modèles proposés pour décrire les données ?
2. La seule courbe qu'un être humain est capable de reconnaître à l'œil est une droite. Ainsi, en TP d'optique, au premier semestre, pour voir si la première loi de Descartes était vérifiée, vous avez tracé $\sin i$ en fonction de $\sin r$ et vous avez vérifié qu'il s'agissait d'une droite. Dans notre cas et pour chacun des deux modèles, déterminez une fonction $y(x)$ qui sera une droite si le modèle est vérifié, i.e. dites ce qu'il faut prendre pour y et pour x pour que les données s'alignent sur une droite si le modèle est valide.
3. Pour chacun des deux modèles, créez le vecteur y et le vecteur x correspondant à partir des données expérimentales.

2.2 Tracer les deux fonctions

Pour tracer les fonctions, il vous faut créer un objet `matplotlib.pyplot` Pour chaque fonction $y(x)$ définie ci-dessus, tracer y en fonction de x . Sur chaque graphique, n'oubliez pas d'indiquer un titre et des axes en utilisant les fonctions `matplotlib.pyplot.xlabel`, `matplotlib.pyplot.ylabel` et `matplotlib.pyplot.title`.

D'après vous quel est le modèle qui décrit le mieux les données ?

2.3 Exploitation quantitative du modèle

Sur le modèle qui fonctionne le mieux, vous allez ajuster vos données par une droite pour en extraire une pente.

1. Est-ce que vous observez vraiment une droite sur toute la gamme de temps ?
2. Pour ne travailler que sur une partie de la droite, il faut créer des vecteurs *TempsRestreint* et *EpaisseurRestreint* restreints à la zone linéaire. Ainsi, si vous observez que la zone linéaire est située entre le 12^{ème} et le 28^{ème}, il faut indiquer :
« `EpaisseurRestreint=epaisseur[12 :28]` ; » et « `TempsRestreint=temps[12 :28]` ; »

Pour faire un justement, il vous faut créer un objet `scipy.optimize`. Pour faire un ajustement des données $y(x)$ par la fonction $f(x)$, procédez comme suit :

1. créer la fonction $f(x)$ (par exemple $f(x) = ax + b$) :
« `def f(x,a,b) :`
`return a*x+b` »
2. « `result=optimization.curve_fit(f,vecteurX,vecteurY,x0)` »
vous retourne un vecteur `result` tel que `result[0][0]` contient la valeur optimale du paramètre a et `result[0][1]` contient la valeur optimale du paramètre b . f est la fonction que vous avez définie, `vecteurX` est le vecteur contenant l'abscisse des données, `vecteurY` est le vecteur contenant l'ordonnée des données et `x0` est un vecteur contenant votre estimation initiale des paramètres a et b (vous pouvez mettre 0.0 pour les deux pour commencer). Effectuez l'ajustement pour le modèle qui marche le mieux.
3. En déduire la valeur de τ .
4. Est-ce que tous les paramètres de son expression sont connus ?
 - (a) Si oui, vérifiez que la valeur de τ obtenue expérimentalement est bien celle prédite théoriquement.
 - (b) Si non, déduire de vos résultats la valeur du paramètre inconnu.

2.4 Discussion des résultats

1. Discuter les erreurs.
2. Discuter les résultats, comparer la différence entre les résultats et les erreurs possibles.