

Python-TP4-2014

October 19, 2014

TP 4 – Informatique CM3 – Octobre 2014

1 Python @ Polytech'Lille

1.0.1 Modalités pour accomplir le TP et compte rendu

Vous aurez à écrire 2 scripts (script1.py , script2.py). Les scripts doivent être accompagnés par un document descriptif unique (README.txt). Dans ce fichier, vous devrez décrire le mode de fonctionnement des scripts et, si besoin, mettre vos commentaires. Merci d'y écrire aussi vos noms et prénoms complets.

Tous les fichiers doivent être mis dans un dossier appelé TP4-nom et ensuite être compressés dans un fichier archive TP4-nom.tgz .

Enfin vous allez envoyer ce fichier par email à l'enseignant: soit Enrico (enrico.calzavarini@polytech-lille.fr) ou Stefano (stefano.berti@polytech-lille.fr).

Vous avez 2 heures à disposition pour ce TP. Le compte rendu est à rendre avant la fin de la séance.

1.0.2 Script 1 : Vitesse maximale d'une voiture dans un virage

Un véhicule se déplace à une vitesse \mathbf{v} de module constant sur une route qui présente un virage de rayon de courbure $r = 50 \text{ m}$. Si le coefficient de frottement statique (adhérence) entre la chaussée et les pneus de la voiture est $\mu_s = 0,8$, quelle est la vitesse maximale à laquelle la courbe peut être parcourue?

A noter que pour que la voiture puisse parcourir le virage sans glisser, une condition d'adhérence doit être respectée. Notamment, le module de la force de frottement F_t (la composante tangentielle de l'action mécanique de contact) doit être inférieur à une valeur maximale $\mu_s N$ proportionnel au module de la composante normale \mathbf{N} (de l'action mécanique de contact). La composante normale équilibre le poids de la voiture \mathbf{P} , donc $N = P$. D'autre part, la force \mathbf{F}_t produit l'accélération centripète $a = \frac{|\mathbf{v}|^2}{r}$ responsable du changement de direction de la voiture (virage).

Par conséquent, la condition d'adhérence devient:

$$F_t \leq \mu_s N$$

où :

$$F_t = M a = M \frac{|\mathbf{v}|^2}{r}$$

$$N = P = M g$$

avec M la masse de la voiture (pas nécessaire dans la solution de ce problème) et $g = 9.81 \text{ ms}^{-2}$ l'accélération de la pesanteur.

Écrire un script qui demande à l'utilisateur le rayon de courbure du virage (en mètres) et la vitesse de la voiture.

L'utilisateur pourra choisir les unités qu'il préfère pour la vitesse (soit "km/h" ou "m/s") et le script devra les convertir de façon appropriée à travers une *fonction*.

Le script devra ensuite afficher la vitesse maximale admise pour la courbe en question (v_{max}) et informer l'utilisateur s'il doit ralentir (quand $v > v_{max}$).

Essayez votre script avec les valeurs $v = 60 \text{ km/h}$ et $v = 108 \text{ km/h}$ pour le module de la vitesse.

1.0.3 Bonus : Freinage sans glissement sur une route rectiligne

La même voiture va à la même vitesse sur une route rectiligne. Le conducteur freine jusqu'à l'arrêter complètement. Quelle est la distance minimale (l_{min}) nécessaire pour arrêter la voiture dans le cas d'une décélération constante?

La décélération maximale de la voiture est déterminée par la valeur maximale de la force de frottement :

$$F_t \leq \mu_s N$$

avec :

$$F_t = M a$$

$$N = P = M g.$$

Si l'accélération est constante, la distance parcourue (l) est liée à la vitesse initiale v et à l'accélération (a) par la relation :

$$l = \frac{1}{2} \frac{v^2}{a}.$$

Ajouter au script précédent l'affichage de la distance minimale d'arrêt (l_{min}).

1.0.4 Script 2 : Fonction de transfert du système de suspension d'un automobile : analyse des données issues d'une mesure.

Nous avons déjà vu que l'expression de la fonction de transfert d'un système de suspension d'une voiture a l'expression suivante:

$$T(u) = A \frac{1}{\sqrt{(1 - u^2)^2 + u^2 / Q^2}}$$

avec l'amplitude $A = F/(m\omega_0^2)$, la pulsation adimensionnée $u = \omega/\omega_0$, et le facteur de qualité Q .

Nous disposons d'une série des données, issues d'une mesure, de la fonction de réponse d'un système de suspension d'automobile.

Les mesures sont organisées en deux fichiers : "pulsation_adim.txt" et "transfert.txt". Le premier contient les valeurs u_i d'échantillonnage de la pulsation adimensionnelle, le second les valeurs de l'intensité de la fonction de transfert T_i pour les valeurs de u_i considérées.

La valeur du facteur de qualité Q est inconnue, et notre objectif est de l'estimer à partir de l'analyse des mesures.

Les mesures, comme dans toute autre expérience, sont affectées par des erreurs aléatoires (bruit expérimental). Par conséquent, afin d'estimer Q , nous devons utiliser un indicateur de vraisemblance. Nous choisissons d'utiliser ici la méthode des moindres carrés, qui est brièvement décrite ci-dessous.

A partir d'une série de N données T_i , échantillonnées à u_i (avec $i = 0, N - 1$, où N est le nombre de lignes dans le fichier), nous définissons le coefficient de vraisemblance S :

$$S = \sum_{i=0}^{N-1} (T(u_i) - T_i)^2$$

La valeur de Q qui minimise le coefficient de vraisemblance S représente la meilleure estimation du facteur de qualité Q . Donc c'est la valeur que nous cherchons, et nous l'appelons Q_{best} .

Le but de cet exercice est d'écrire un script qui effectue les actions suivantes.

- a) Lire les données des deux fichiers et tracer un graphique de T_i en fonction de u_i .

- b) Effectuer l'analyse des moindres carrés, à l'aide d'une *fonction* qui calcule S , et d'une *boucle* (“while” ou “for”) pour déterminer la valeur Q qui minimise S dans l'intervalle $Q \in [0.1, 3]$. Dans la boucle on pourra incrémenter les valeurs de Q par addition d'une quantité petite, par exemple $\Delta Q = 0.1$. Enfin, afficher la valeur de Q_{best} . On rappelle que T_i sont les valeurs mesurées et $T(u_i)$ sont les valeurs calculées (à partir des valeurs expérimentales de u_i) avec la fonction $T(u)$. Dans cet exercice on prendra $A = 2$.
- c) Tracer un graphique de T_i en fonction de u_i ainsi que de la courbe de transfert $T(u)$ correspondant à la valeur $Q = Q_{best}$.

Conseils :

- 1) Pour la lecture des données à partir des fichiers vous pouvez suivre les informations dans le TP1 (et dans votre script 3 du TP1).
- 2) Il pourrait être aussi utile de se rappeler que la commande “len” restitue le nombre d'éléments d'une liste (ou d'un array), voir indications dans le TP1.
- 3) Après avoir lu les données, il sera nécessaire de les transformer en des tableaux numpy. Pour ce faire il est possible d'utiliser la commande “np.asarray”. Supposons que vous avez lu les données du tableau “data”. Ce tableau sera de type “chaîne de caractères”. Afin de le convertir en un tableau numpy, vous pouvez utiliser:

```
data = np.asarray(data, dtype=float)
```

- 4) Pour calculer la fonction de transfert $T(u)$, vous pouvez réutiliser une partie de votre script 2 du TP2.
- 5) Pour calculer le minimum de S avec une boucle sur les valeurs de Q : initialiser la valeur du minimum à une valeur très grande (p.ex. $S_{min} = 1000.0$) avant de rentrer dans la boucle; calculer S pour la valeur actuelle de Q et vérifier si $S < S_{min}$; en cas affirmatif S devient le nouveau minimum; itérer sur toutes les valeurs de Q .