

## 1 Tracés de courbes

### 1.1 Avec des listes et avec numpy

On souhaite tracer la fonction  $\text{ch}$  sur l'intervalle  $[-4, 4]$ . Importer les modules dont vous allez avoir besoin.

#### 1. Utilisation des listes.

- Définir la fonction  $\text{ch}$ .
- Définir une liste de 10000 abscisses entre  $-4$  et  $4$  et la liste des images par  $\text{ch}$  qui correspond.
- Faire le tracé de la courbe et l'afficher.

#### 2. Utilisation de numpy.

- Définir à l'aide de la commande *linspace* un tableau de 10000 abscisses entre  $-4$  et  $4$ .
  - Définir le tableau des ordonnées correspondantes, on utilisera la fonction *exp* du module *numpy*.
  - Faire le tracé de la courbe et l'afficher.
3. La fonction  $\text{ch}$  existe déjà dans le module *math*, consulter la documentation de ce module pour la trouver et refaire la question 1. en vous en servant.

### 1.2 La fonction inverse

Tracer la fonction inverse sur le domaine de définition  $[-3, 3] \setminus \{0\}$  en prenant 200 points. Quel est le problème du graphe que vous obtenez ? Comment le corriger ?

### 1.3 Loi de Benford

- Créer la liste des  $2^k$  pour  $k$  allant de 0 à 1000.
- Créer la liste des premiers chiffres de chaque nombre de la liste précédente (le premier chiffre de 256 est 2). On pourra remarquer que la commande *str*( $n$ ) convertit le nombre  $n$  en une chaîne de caractères et que la commande *int* permet de convertir une chaîne de caractères en nombre, quand cela a un sens.
- Tracer l'histogramme correspondant à la liste précédente.
- Refaire l'expérience précédente avec les puissances de 3, les puissances de  $\pi$ , les puissances de 1.5, les puissances de  $\sqrt{2}$ . Que remarquez-vous ?

## 2 Création d'un module simple

- Écrire une fonction *max2*( $a$ ,  $b$ ) qui prend en paramètres deux nombres  $a$  et  $b$  et qui renvoie le maximum des deux nombres.
- Sauvegarder ce fichier sous le nom *mafonction.py*.
- Dans un nouveau fichier, importer le module que vous venez de créer et servez-vous de la fonction *max2* pour créer une fonction *max4*( $a$ ,  $b$ ,  $c$ ,  $d$ ) qui renvoie le maximum des 4 nombres  $a$ ,  $b$ ,  $c$  et  $d$ .

## 3 Utilisation du module time

On considère les trois fonctions suivantes qui créent la liste des valeurs de la fonction sinus pour les  $n$  premiers entiers naturels.

```
1 import math as m
2 def f1(n):
3     L = []
4     for i in range(n):
5         L = L + [m.sin(i)]
6     return(L)
7
8 import math as m
9 def f2(n):
10    L = []
11    for i in range(n):
12        L.append(m.sin(i))
13    return(L)
14
15 import math as m
16 def f3(n):
17    return([m.sin(i) for i in range(n)])
```

Comparer les temps d'exécution des trois fonctions pour différentes valeurs de  $n$ . Que peut-on en conclure ?

## 4 Mélange et tri d'une liste

### 1. Mélange d'une liste.

- Écrire une fonction  $crea(n)$  qui prend en paramètre un entier naturel non nul  $n$  et renvoie la liste des entiers de 1 à  $n$ .
- Écrire une fonction  $mel(L)$  qui mélange la liste  $L$ . Le principe est le suivant :
  - on crée une liste  $M$  qui va être la liste  $L$  mélangée
  - on choisit un indice  $i$  au hasard entre 0 et  $len(L) - 1$
  - on met l'élément d'indice  $i$  dans la liste  $M$  et on enlève cet élément dans la liste  $L$  avec la commande  $L.pop(i)$ .
  - on recommence l'opération jusqu'à ce que la liste  $L$  soit vide.
- Dans le module *random*, il existe une fonction permettant de mélanger une liste, allez voir dans la documentation et trouver son nom. Dans la suite de l'exercice, on se servira de cette fonction qui est plus rapide que la nôtre.

### 2. Temps d'exécution du tri bulles.

- Dans le répertoire du tp3, vous trouvez un module *tri*. Importer ce module afin de pouvoir vous servir de la fonction *tri.bulles* qui y figure.
- On considère la liste des entiers de 1 à 100 que l'on mélange et on calcule le temps mis par la fonction *tri.bulles* pour la trier. Répéter 10 fois cette opération pour avoir le temps moyen mis pour faire ce tri.
- Tracer la courbe représentant le temps moyen (toujours sur 10 essais) mis par le tri bulles pour trier une liste de  $i$  éléments pour  $i$  allant de 10 à 200. Les abscisses sont ces valeurs de  $i$  et les ordonnées les temps correspondants. Que remarquez-vous ?

## 5 On s'amuse comme on peut

Pour s'occuper, la petite Alice, 6 ans, joue au jeu suivant :

- elle prend un jeu de 52 cartes,
- elle lance toutes les cartes en l'air et enlève les cartes qui sont retombées face visible,
- elle recommence ce procédé jusqu'à ce que toutes les cartes soient éliminées.

Écrire une ou plusieurs fonctions en Python permettant d'avoir une valeur approchée du nombre moyen de lancers qu'Alice doit effectuer. Quel est ce nombre ?