PC – Lycée René CASSIN **Informatique – TP1 – Calcul approché d'intégrale et représentation graphique** **Durée : 1 h**

I. Rappels et compléments sur les bibliothèques

On importe la bibliothèque avec **import** *bibliotheque* **as** *bib* ou une fonction avec **from** *bibliotheque* **import** *fonction*.

On appelle ensuite la fonction avec *fonction* si on l'a importée ou *bib.fonction* si on a importé la bibliothèque.

On évitera en pratique **from** *bibliotheque* **import** * qui importe toutes les fonctions d'une bibliothèque, mais cela ne pose a priori pas de problème dans un sujet de concours.

dir(*bibliotheque*) donne la liste des objets de la bibliothèque.

help(*bib.fonction*) fournit de l'aide sur une fonction (ou une classe….).

Principales bibliothèques utiles en mathématiques

1. **Numpy** contient les fonctions mathématiques en version vectorielle, les objets du type array, matrix et les fonctions associées (calcul de vecteurs propres …), les fonctions sort, roots (racines de polynômes), …
2. **Scipy** contient les fonctions de résolution approchée d'équations dans scipy.optimize, d'équations différentielles et le calcul approché d'intégrales dans scipy.integrate, des fonctions statistiques dans scipy.stats, des fonctions d'algèbre linéaire (dont beaucoup sont également présentes dans numpy), d'interpolation …
3. **Matplotlib** permet les représentations graphiques d'objets mathématiques. La commande principale est la commande plot présente dans matplotlib.pyplot (que l'on importe généralement **as** plt).
   Nombreux exemples sur : http://matplotlib.org/gallery.html
4. **Pylab** (importé **as** pl) combine les fonctions de pyplot et numpy.
5. **Sympy** permet le calcul formel. Il ne fait pas parti du programme d'informatique mais peut vous être utile dans d'autres disciplines. Quelques possibilités : limit, factor, solve, diff, integrate, matrix, det, eigenvals, eigenvects, diagonalize …
   Pour plus d'informations : http://docs.sympy.org/latest/index.html

II. Représentation graphique et calcul d'intégrales

Les fonctions permettant le calcul approché d'intégrale sont quad, trapz, simps, et pour les intégrales multiples dblquad et tplquad, de la bibliothèque scipy.integrate.

plot([x_1, x_2,…, x_n],[y_1, y_2,…, y_n]) crée automatiquement la fenêtre graphique en fixant l'échelle et relie les points de coordonnées (x_i, y_i). Tester :

```
plt.plot(np.linspace(-3,3,1000), np.linspace(-3,3,1000)**2)
```

Cette commande fonctionne car l'opération **2 est vectorielle. Dans le cas contraire, on aurait dû écrire par exemple :

```
plt.plot(np.linspace(-3,3,1000),[x**2 for x in np.linspace(-3,3,1000)])
```

Activité 1 – Une intégrale impropre

Conjecturer la nature (et éventuellement la valeur) de l'intégrale de Dirichlet $\int_0^{+\infty} \frac{\sin x}{x} dx$.

Activité 2 – Une fonction définie par morceaux

On considère la fonction f définie pour tout $k \in \mathbb{Z}$ par :

- $f(x) = \cos(x)\sin(x)$ si $x \in [(2k)\pi, (2k+1)\pi[$
- $f(x) = \cos(x)$ si $x \in [(2k+1)\pi, (2k+2)\pi[$

1. Créer une fonction *trace* représentant la fonction f sur $[0, t]$, t étant un réel positif entré par l'utilisateur.
2. Calculer $\int_0^t |f(t)|dt$, à l'aide d'un programme, puis en utilisant la bibliothèque scipy.integrate.

Remarque : Dans l'optique des concours, il est nécessaire de savoir programmer les algorithmes classiques, mais pour un usage personnel, lorsqu'on a le choix, on préférera utiliser les fonctions des bibliothèques qui sont souvent écrites en C, langage compilé, donc plus rapides.