Retour à IS104 – Algorithmique Numérique (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=159)

Projet 1 : Méthodes de calcul numérique / Limites de la machine

Il est conseillé de lire **entièrement** chaque partie avant de s'atteler à la tâche.

Le but de ce projet consiste à évaluer les problèmes qui peuvent apparaître lors de l'utilisation d'opérations élémentaires, voire d'algorithmes plus poussés, sur des nombres flottants.

- la première partie s'intéresse à trouver des exemples dans lesquels les opérations élémentaires sont insuffisamment précises;
- la seconde partie donne des exemples d'algorithmes utilisés dans des conditions de calcul en basse précision.

Par équipe, vous pouvez vous distribuer les différentes tâches à effectuer. Idéalement, tout algorithme réalisé par une partie de l'équipe est testé par l'autre partie de l'équipe. Le rapport final doit rendre compte à la fois des algorithmes et des tests de validation effectués. **Lorsque rien n'est précisé, toute latitude vous est laissée pour décider des façons de procéder** (codage des algorithmes, choix des batteries de test).

Représentation des nombres en machine

Nous allons simuler la représentation des nombres en machine à l'aide de Python. Supposons que l'on veuille représenter ces nombres en base 10, avec une précision donnée de p décimales. Le terme « décimales » compte ici le nombre de chiffres significatifs (au sens IEEE 754) et pas le nombre de chiffres après la virgule. Comme il ne s'agit que d'une simulation, on supposera que p est faible. cette représentation est appelée ici la **représentation décimale réduite**. Tous les calculs appelés ici **réels** seront les calculs effectués par Python.

1. Ecrire une fonction qui, étant donné un nombre `x`, calcule sa représentation décimale réduite `rp(x,p)` (remarquer que l'exercice porte uniquement sur des manipulations de nombres flottants, et pas des manipulations de bits en mémoire). Les exemples suivants peuvent servir de batterie de tests :

Nombre réel	Sur 4 décimales	Sur 6 décimales
3.141592658	3.142	3.14159
10507.1823	10510	10507.2
0.0001857563	0.0001858	0.000185756

   Attention : cette fonction doit calculer un nombre (par opposition à simplement afficher un résultat).
2. Simuler les opérations usuelles que sont l'addition et la multiplication en représentation décimale réduite.
3. Considérons deux nombres `x` et `y` en représentation décimale réduite. Il est possible de calculer l'erreur relative réalisée en ajoutant `y` à `x` à l'aide de la formule :

   Erreur relative sur la somme : $\delta_s(x,y) = \frac{\left|(x+y)_{\text{réel}} - (x+y)_{\text{machine}}\right|}{\left|(x+y)_{\text{réel}}\right|}$

   Ecrire une fonction permettant de calculer cette erreur relative en fonction de `x` et `y`. Calculer de même l'erreur relative effectuée sur un produit :

   Erreur relative sur le produit : $\delta_p(x,y) = \frac{\left|(x\times y)_{\text{réel}} - (x\times y)_{\text{machine}}\right|}{\left|(x\times y)_{\text{réel}}\right|}$
4. Fixer `x` et tracer le graphe des fonctions précédentes en fonction de `y`. Essayer de choisir `x` de manière à faire apparaître une erreur relative aussi grande que possible. Donner des exemples d'erreurs relatives maximales obtenus.
5. **Calcul de log(2) :** Considérons la formule suivante permettant de calculer le logarithme népérien de 2 :

   $$\log(2) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n}$$

   Ecrire un algorithme de votre choix permettant de calculer une valeur approchée de $\log(2)$ sur `p` décimales, en utilisant la formule précédente. Cet algorithme devra en même temps évaluer l'erreur relative sur le résultat final obtenu en utilisant les fonctions précédentes.

Ce qui est évalué dans le rapport : pouvoir expliquer simplement le fonctionnement d'un algorithme simple, pouvoir dessiner un graphique, pouvoir expliquer les résultats des tests réalisés.

Algorithmes CORDIC

Dans cette partie, nous allons nous intéresser à des algorithmes utilisés dans des conditions où l'on ne possède pas de ressources importantes (mémoire, puissance de calcul) pour effectuer des calculs. Ce type de condition se présente naturellement dans les calculatrices de poche.

1. Récupérer la page de la FAQ (http://www.usenet-fr.net/fur/maths/maths-faq-3.html) du groupe de discussions

fr.sci.maths (http://groups.google.com/group/fr.sci.maths), et s'intéresser au chapitre VI.2.

2. Quelle est la représentation des nombres utilisés sur une calculatrice ? Quels avantages et inconvénients pouvez-vous voir à ce genre de représentation ?
3. Dans cette page, quatre algorithmes sont décrits pour calculer les fonctions trigonométriques et exponentielles. Quelle est la technique générale utilisée pour réaliser ces algorithmes ? En particulier, en quoi cette technique vous semble t'elle efficace lorsqu'elle est ramenée à une calculatrice ?
4. Implémenter ces quatre algorithmes sous Python, en utilisant les mêmes techniques d'optimisation que celles décrites sur cette page :
   - les fonctions exponentielle `exp` et logarithme népérien `ln` ;
   - les fonctions tangente `tan` et arctangente `arctan` .
5. Vérifier vos algorithmes en les testant par la méthode de votre choix.

La question 6 demande de lire un texte écrit en anglais disponible sur le réseau. Un lexique (files/dico.ps.gz) non exhaustif (mais pouvant être enrichi) est disponible pour la traduction de certains termes mathématiques.

6. Considérer les pages 165 à 189 du Numerical Recipes in C (http://www.nrbook.com/) (free edition, chapitres 5.0 jusqu'à 5.7, soit 25 pages en tout). Repérer **3** problèmes soulevés lors de l'évaluation de fonctions usuelles en machine, ainsi que les solutions éventuellement envisageables pour y remédier. Par exemple :

   « P. 184, le calcul des racines d'un polynôme du second degré à coefficients réels peut provoquer des erreurs d'approximation assez larges lorsque l'on utilise la méthode de calcul habituelle à l'aide du discriminant.

   $$ax^2 + bx + c = 0$$

   $$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

   Ceci est dû au fait que l'on soustrait deux quantités équivalentes. Pour éviter ce genre de problème, il vaut mieux utiliser les propriétés des racines de ces polynômes et calculer :

   $$q = \frac{1}{2}\left(b + sgn(b)\sqrt{b^2 - 4ac}\right)$$

   $$x_1 = \frac{q}{a}$$
   $$x_2 = \frac{c}{q}$$

   … avec `sgn(b)` valant +/- 1, du même signe que `b` . »

   Naturellement, l'exemple ci-dessus ne compte pas dans le décompte des 3 problèmes à présenter. Toute technique **décrite de manière incompréhensible** sera proprement ignorée.

Ce qui est évalué dans le rapport : pouvoir expliquer les problèmes rencontrés pour transposer un algorithme simple, pouvoir extraire de l'information d'une source en français, et en anglais.

Dans cette section