Sup Galilée **Méthodes numériques**
MACS 1 **Année 2008-2009**

TD/TP - 1

Erreurs d'arrondi, d'annulation, instabilité numérique.

But : 1) Calculer les paramètres qui définissent l'arithmétique IEEE finie en MATLAB (epsilon machine, plus petit nombre machine normalisé, ...)
2) Observer sur des exemples l'effet très néfaste des erreurs d'arrondi d'un ordinateur.

Exercice 1. On peut utiliser une simple précision en MATLAB avec la fonction `single`. Exécuter les commandes

```
>> format hex
>> y = single(6.5)
```

et convertir le nombre hexadécimal correspondant en un nombre binaire en développant chaque chiffre hexadécimal en 4 chiffres binaires. Vérifier que le nombre en binaire avec 32 chiffres trouvé correspond bien au nombre décimal 6.5.

Exercice 2. Essayer de calculer les paramètres qui définissent l'arithmétique IEEE finie en MATLAB.

1. Calculer la précision machine *eps*. Indication : utiliser le fait que *eps* est le plus petit nombre machine tel que $1 + eps > 1$ sur la machine.

2. Calculer le plus petit nombre machine normalisé α. Comparer votre résultat avec *realmin*.

3. Calculer le plus petit nombre machine dénormalisé.

Exercice 3. Associativité. Ecrire un programme calculant

$$s = \sum_{i=1}^{N} \frac{1}{i},$$

N entier donné. On calculera s de deux façons :

$$s = 1 + 1/2 + ... + 1/N = 1/N + 1/(N-1) + ... + 1.$$

Pour $N = 1e6$, comparez vos résultats avec celui obtenu avec Maple $s_M = 14.392726722865723804$. Conclusions ? Comparer vos résultats à l'algorithme de W. Kahan

```
s=0;
c=0;
for j=1:N
  y=1/j+c;
  t=s+y;
  c=(s-t)+y;
  s=t;
end
s=s+c
```

Exercice 4. Monotonicité. On considère le polynôme

$$f(x) = x^3 - 3.000001x^2 + 3x - 0.999999.$$

Représenter f sur $[-1, 3]$ avec un pas de 0.1, puis sur $[0.998, 1.002]$ avec un pas 0.0001, puis sur $[0.999999993, 1.000000007]$ avec un pas de 0.000000000005. Que constatez-vous ? Que se passe-t-il si on représente f sous la forme (méthode de Hörner)

$$f(x) = ((x - 3.000001)x + 3) - 0.999999?$$

Exercice 3. Calcul de π. Ecrire les programmes MATLAB du calcul approché de π par les polygônes inscrits vu en cours (la version naïve, puis la version stable).

Exercice 4. Calcul de $e^{-x},\ x > 0$. Le calcul de $e^{-x},\ x > 0$ est très sensible aux erreurs d'arrondi, lorsqu'on le calcule par un développement en série de Taylor

$$e^{-x} = \sum_{j=0}^{\infty} \frac{(-x)^j}{j!} = 1 - x + \frac{x^2}{2} - \frac{x^3}{6} + \frac{x^4}{24} - \cdots$$

Sur l'ordinateur, on doit se fixer un entier N et on calcule une valeur approchée de e^{-x} par la formule

$$e^{-x} \approx \sum_{j=0}^{N} \frac{(-x)^j}{j!} \tag{1}$$

Ecrire la fonction Matlab `Expm`. Représenter sur un graphe les valeurs approchées de e^{-20} en fonction de N, pour N variant de 1 à 20. Qu'observez-vous ? Représenter dans un tableau les différentes valeurs de l'approximation de e^{-20} en fonction de N, pour N prenant les valeurs suivantes : 20,40,60,80,100,120. Ajouter dans ce tableau la valeur de e^{-20} obtenue avec Matlab. Quel problème observez-vous sur la précision obtenue avec vos calculs ? Identifier d'où vient ce problème, et trouver un moyen d'y palier.

Exercice 5. Instabilité numérique. Il s'agit de phénomènes d'amplification des erreurs d'arrondi, qui se produisent souvent dans le cas de calculs récurrents ou itératifs. Considérons le cas du calcul itératif suivant : calculer une suite (u_n) définie par sa valeur initiale u_0 et la relation

$$u_{n+1} = f(u_n), \quad n \geq 0$$

où f est une fonction donnée. Par exemple, prenons $u_0 = 2$ et $f(x) = |ln(x)|$.

a) Ecrire un programme permettant le calcul approché de u_k pour $k = 1, 2, \ldots, n$ où n est un entier dont la valeur sera fixée au début du programme. Quelle valeur obtenez-vous en matlab pour u_{30} ?
Ce calcul est-il précis, compte tenu des erreurs d'arrondi ? Pour cela, on pertube la donnée initiale, en prenant $u_0 = 2 - 1e - 9$. Quelle nouvelle valeur obtenez-vous pour u_{30} ? Conclusions ?
b) Ecrire un programme permettant le calcul d'un tableau à 9 lignes et 3 colonnes, défini de la façon suivante :

1. $1^{\text{ère}}$ colonne (dans l'ordre) : valeurs de $u_0, u_5, u_{10}, u_{15}, u_{20},\ u_{24}, u_{25}, u_{26}, u_{30}$ partant de la valeur $u_0 = 2$,
2. $2^{\text{ème}}$ colonne (dans l'ordre) : valeurs de $u_0, u_5, u_{10}, u_{15}, u_{20},\ u_{24}, u_{25}, u_{26}, u_{30}$ partant de la valeur $u_0 = 2 - 1e - 9$,
3. $3^{\text{ème}}$ colonne (dans l'ordre) : valeurs de $e_0, e_5, e_{10}, e_{15}, e_{20},\ e_{24}, e_{25}, e_{26}, e_{30}$ où e_n représente l'erreur relative commise entre la première et la deuxième colonne.

Qu'observez-vous sur l'erreur relative commise entre la première et la deuxième colonne ? La valeur de u_{30} est-elle précise à 10^{-9} près ? A partir de quelle valeur de u_n a-t-on une "catastrophe numérique". Pourquoi ?