Représentation des nombres, approximations et erreurs.

Méthodes numériques

2014/15

Notations : les nombres flottants normalisés sont de la forme

$$s\left(\sum_{k=1}^{r}\frac{a_k}{b^k}\right)b^e \quad \text{où :}$$

- $b>1$ est un entier (la base), la plus part du temps 2
- les $(a_1,\ldots,a_r)$ sont des entiers compris entre 0 et $b-1$ avec $a_1\neq 0$, ce sont les digits ou bits ou ... de la mantisse. (Si $b=2$ on ne stocke pas $a_1=1$).
- $s\in\{+1,-1\}$ est le signe
- e est l'exposant, $L\leq e\leq U$.

La norme IEEE-754 (langages de programmation, logiciels de calcul numérique...) utilise $b=2$, $r=53$, $L=-1021$, $U=1024$). Xcas utilise cette norme mais avec troncature des 5 bits les moins significatifs de mantisse à 0 en précision normale ($r=48$), en multiprécision r est fixé par l'utilisateur (`Digits` ou second argument de `evalf`), et L,l valent au moins 2^{30} en valeur absolue.
La saisie d'un nombre flottant se fait en écrivant la mantisse en base 10 (avec éventuellement un point décimal), suivi éventuellement de la lettre `e` et de l'exposant en base 10, même si le flottant est stocké en base 2. Par exemple `6.02e23` pour le nombre d'Avogadro. Attention, en Xcas, `10^10` est un entier exact, à ne pas confondre avec le flottant `1e10`.

Exercice 1 Ecriture en base b d'un rationnel Soit b un entier supérieur ou égal à 2. Soit p et q deux entiers strictement positifs tels que $p<q$.
Soit $\frac{p}{q}=\sum_{k=1}^{+\infty}\frac{a_k}{b^k}$ le développement en base b de $\frac{p}{q}$.

1. Montrer que la suite (a_k) est obtenue de la façon suivante :

   $r_0=p$, a_1 est le quotient de la division euclidienne de $r_0 b$ par q et r_1 est le reste

   Et si $(r_0,\cdots,r_n)$ et $(a_1,\cdots,a_n)$ sont définis alors a_{n+1} est le quotient de la division euclidienne de $r_n b$ par q et r_{n+1} est son reste.
2. Montrer que la suite (a_k) est périodique à partir d'un certain rang.
3. Donner le developpement de $x=\frac{1}{10}$ en base 2. Soit $\widehat{x}$ le flottant arrondi normalisé avec 12=11+1 bits de mantisse correspondant à x. Calculer $\widehat{x}$. Même question pour $y=2/5$. Faire la somme $\widehat{x}+\widehat{y}$.

4. Expliquez le résultat de l'opération `0.5-(0.1+0.4)` dans Xcas.

Exercice 2 Durant la première guerre du Golfe, une batterie anti-missile Patriot a raté l'interception d'un missile Scud qui causa la mort de 28 personnes. Nous allons essayer de comprendre pourquoi.
L'ordinateur de la batterie Patriot représente les nombres par des nombres à virgule fixe en base 2 avec une précision de 23 chiffres après la virgule. L'horloge de la batterie compte le temps en dixième de seconde. Pour obtenir le temps en seconde, le programme divise le temps donné par l'horloge par dix.
1) En utilisant l'écriture en base 2 de $\frac{1}{10}$ trouvée dans l'exercice précédent, donner une estimation de l'erreur d'approximation de 1/10 par la représentation des nombres de la batterie Patriot.
2) La batterie fonctionne pendant 100 heures. Donner une estimation de l'erreur introduite dans le temps en secondes.
3) Sachant qu'un missile Scud voyage à 1676 mètres par seconde, donner l'erreur de distance commise par la batterie anti-missile.

Exercice 3 Donner la valeur exacte de chaque expression suivante.

$$\sum_{n=1}^{+\infty} \frac{1}{n} \quad , \quad (10^{20}+1) - 10^{20} \quad , \quad \binom{1000}{2} = \frac{1000!}{2(998!)} \; .$$

On donne ces calculs à faire à une machine. Sa réponse correspondra-t-elle à l'expression exacte ? (Discuter selon le mode de calcul, exact ou approché, et selon la précision).

Exercice 4 Les nombres suivants peuvent se calculer selon deux manières notées A et B. Quelle est l'expression la plus judicieuse pour le calcul approché sur machine ?
– $A = 1 - \cos^2(10^{-8})$, $\quad B = \sin^2(10^{-8})$
– $A = \sum_{n=1}^{10^9} \frac{1}{n}$, $\quad B = \sum_{n=0}^{10^9-1} \frac{1}{10^9-n}$.
– $A = e^{-10} \approx \sum_{n=0}^{30} \frac{(-10)^n}{n!}$, $\quad B = \frac{1}{e^{10}} \approx \left(\sum_{n=0}^{30} \frac{10^n}{n!}\right)^{-1}$
Proposez une méthode de calcul approché précise pour calculer :

$$\frac{1}{\sqrt{10^{10}+1} - \sqrt{10^{10}-1}}$$

Exercice 5 On effectue la somme de n=100 nombres flottants de taille comparable, on suppose que chaque opération d'addition introduit une erreur absolue distribuée selon une loi uniforme sur $[-1, 1]$. Effectuer plusieurs simulations de la valeur de la somme des erreurs, observer la distribution des erreurs obtenues. Commandes Xcas utiles : `M:=ranm(n,N,uniform,-1,1)` pour générer une matrice n par N d'erreurs selon la loi uniforme, `sum(M)` effectue la somme de ces erreurs par colonnes et `histogram` permet de les représenter. On pourra comparer l'histogramme de `sum(M)/sqrt(n)` et la loi normale de moyenne nulle et d'écart-type `stddev(uniform,-1,1)` ($=\frac{1}{\sqrt{3}}$) celle de la loi uniforme considérée.

Exercice 6 : évaluation d'un polynôme en un réel. On veut calculer la valeur en x d'un polynome $\sum_{k=0}^{n} a_k X^k$. On suppose que $a_0, \dots, a_n, x$ sont représentés exactement par des nombres flottants (on pourra aussi réfléchir aux erreurs introduites si la représentation n'est pas exacte !).

1. On fait le calcul de la manière suivante : $s_0 = a_0, u_1 = x, v_1 = a_1 u_1$ et pour $k \geq 1$

$$s_k = s_{k-1} + v_k, \quad u_{k+1} = u_k x, \quad v_{k+1} = u_{k+1} a_{k+1}$$

   Combien a t-on fait de multiplications et d'additions ?

2. Méthode de Hörner
   Le principe est d'écrire $P(X) = \sum_{k=0}^{n} a_k X^k = a_0 + X(a_1 + X(\cdots + X(a_{n-1} + X a_n)))$.
   On pose pour $0 \leq k \leq n$,

$$p_k = a_k + a_{k+1} x + \cdots + a_n x^{n-k}$$

   On a alors $P(x) = p_0$. On calcule les termes par une récurrence descendante
   $p_n = a_n$ et $p_{k-1} = a_{k-1} + x p_k$
   Combien a t-on fait de multiplications et d'additions ?

3. Ecrire deux programmes évaluant un polynome donné par ses coefficients en x. Comparer en utilisant `time()` la rapidité d'execution.
   Dans Xcas, la fonction `horner(P,x)` évalue en x le polynome P par la méthode de Hörner. On peut définir un polynome par son expression symbolique ou la liste de ses coefficients par ordre décroissant. On peut faire le calcul avec des rationnels et arrondir la valeur exacte, que gagne-t-on alors ? Qu'observe-t-on en terme de temps d'exécution ?

4. Bonus : Estimation des erreurs par ces deux méthodes (attention, question délicate !).
   On note u l'erreur d'arrondi/troncature machine. On rappelle que si x et y sont des flottants, si $* \in \{+, \times\}$, alors il existe ϵ tel que $x \circledast y = (1+\epsilon)(x * y)$ et $|\epsilon| \leq u$.
   En utilisant la première méthode, on a : $\widehat{s_0} = a_0; \widehat{u_1} = x; \widehat{v_1} = a_1 \otimes \widehat{u_1}$ et pour $k \geq 1$,

$$\widehat{s_k} = \widehat{s_{k-1}} \oplus \widehat{v_k}, \quad \widehat{u_{k+1}} = \widehat{u_k} \otimes x, \quad \widehat{v_{k+1}} = \widehat{u_{k+1}} \otimes a_{k+1}$$

   La valeur donnée par la machine est donc $\widehat{s_n}$.
   On note $\gamma_n(u) = (1+u)^{n+1} - 1$. Montrer par récurrence que

$$|\sum_{k=0}^{n} a_k x^k - \widehat{s_n}| \leq \gamma_n(u) \ \sum_{k=0}^{n} |a_k||x|^k$$

   (On pourra montrer $|\hat{v}_n / v_n| \leq (1+u)^n$, $|\hat{v}_n - v_n| \leq \gamma_{n-1}(u)|v_n|$, $|\hat{s}^n| \leq (1+u)^{n+1} sum|a_k x^k|$). Peut-on en déduire une information sur l'erreur relative du résultat renvoyé, en faisant éventuellement des hypothèses sur les signes ?

Les processeurs de calcul flottant utilisent souvent une représentation plus précise que le standard IEE-754, avec 64 bits de mantisse (+1 implicite) au lieu de 52 bits (+1 implicite) (ou des 47+1 bits de Xcas). Si on effectue les calculs intermédiaires de cette manière (ou plus généralement en multi-précision), qu'y gagne-t-on ?
En notant $\widehat{p_k}$, le flottant obtenu par la méthode de Hörner, on peut montrer que

$$|P(x) - \widehat{p_0}| \leq \sum_{k=0}^{n} \gamma_{2k+2}(u)|a_k||x|^k$$

Cette majoration sera meilleure que celle obtenue par la première méthode si les termes $a_k x^k$ décroissent vers 0, mais elle reste du même type.

Exercice 7 On souhaite calculer l'intégrale

$$I_n = \int_0^1 \frac{x^n}{10+x} dx$$

pour $n \in \mathbb{N}$.

1. Calculer I_0.
2. Montrer que pour tout $n \in \mathbb{N}^*$,

$$I_n = \frac{1}{n} - 10 I_{n-1}.$$

   En déduire une procédure pour calculer I_n pour n quelconque.
3. Utiliser la procédure de (2) pour calculer en flottant I_5, I_{10}, I_{20} avec Xcas. Les estimations obtenues vous paraissent-elles raisonnables (noter par exemple que, pour tout n, $0 \leq I_n \leq 1$) ?
4. Montrer que l'on peut renverser la procédure ci-dessus :

$$I_{n-1} = \frac{1}{10n} - \frac{I_n}{10}.$$

   A partir d'une estimation grossière de I_{30}, utiliser cette procédure inverse pour estimer I_{20}. Comparer le résultat avec la valeur de l'intégrale fournie par Xcas.