République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
Université Ibn Khaldoun – Tiaret
Faculté des Mathématique et de l'Informatique
Département d'Informatique

METHODES NUMERIQUES

COURS POUR :

2éme ANNEE LICENCE INFORMATIQUE

Par : Mr. KARIM MEZZOUG

karim.mezzoug@univ-tiaret.dz

Table des Matières

Remarque : **Le** Chapitre 5 : Analyse matricielle a été abordée d'une façon intégrale et progressive à travers les chapitres 2,3 et 4.

Chapitre I : Généralités sur l'Analyse Numérique et le Calcul Scientifique

1. Motivations :

L'analyse numérique est devenue indispensable dans de très nombreux domaines. Les différents disciplines qui ont recours au calcul numérique sont trop variées pour être décrient ici. Elle va de la l'astronomie à la sociologie en passant par la physique, la chimie, la biologie, la médecine, le génie civil, l'archéologie, la gestion des stocks, la psychologie expérimentale, les jeux vidéo, etc…

À travers ce chapitre, en étudiera essentiellement en premier lieu la notion d'erreur ensuite l'arithmétique en virgule flottante en passant par la propagation de l'erreur puis nous terminons par la notion de conditionnement d'un problème et la stabilité des algorithmes numériques. Tout ceci sera accompagné par quelques exemples démonstratifs.

2. Notions d'Erreurs :

La notion d'erreur n'est pas considérée au sens de faute dans la méthodologie, ou instruction fausse dans le programme. En faite elle concerne toutes erreurs inévitables. Ainsi les erreurs se distinguent en quatre catégories :

- **Les erreurs humaines** : Toutes erreurs commises à cause de l'inattention ou l'oublie, etc.. des êtres humains pendant des mesures effectuées à la main et à l'œil nue.

- **Les erreurs sur les données** : Généralement dues à l'imprécision des mesures effectuées par des capteurs ou des instruments de mesures électroniques, etc...

- **Les erreurs d'arrondi**. Un calculateur numérique ne peut représenter les nombres réels qu'avec un nombre fini de chiffres, d'où à chaque opération mathématique élémentaire, il pourra y avoir une perte de chiffres significatifs, surtout quand il s'agit de nombre d'opérations très important.

- **Les erreurs d'approximation** : ou de discrétisation. Dues essentiellement à la méthodologie et l'algorithme utilisé lui-même par exemple, lorsqu'on calcule une intégrale à l'aide d'une somme finie, une dérivée à l'aide de différences finies ou bien la somme d'une série finie à l'aide d'un nombre fini de ses termes. On analyse numérique généralement on cherche à évaluer ces erreurs de discrétisation pour chaque algorithme proposé.

3. Représentation des nombres en Machine et Arithmétique en virgule flottante :

Les opérations arithmétiques que nous effectuons tous les jours sont generalement exactes, par exemple payer sa facture au supermarché ou au guichet d'une salle de spectacle. Dans de tels cas nous manipulons des nombres entiers et des réels ayant peu de chiffres décimaux en effectuant que de simples additions ou soustraction.

Mais réellement ce n'est pas toujours le cas et se rend compte que nos calculs sont parfois trompeuses : par exemple le (19/09/2019), j'ai fait le plein d'essence routier à 41,97 DA le litre, ce qui m'a coûté 1768 DA, alors que mon réservoir a accepté 42.12 litres de carburant. J'aurais dû, en réalité, débourser 1767.7764 DA.

Mais, ai-je bien reçu la quantité affichée au compteur de la pompe ? C'est peu probable. La mesure du volume fourni est évidemment entachée d'une certaine erreur (qui varie d'ailleurs en fonction de la température ambiante, de la pression atmosphérique et du débit de fluide) et j'ai peut-être payé mon carburant à 42 DA ou 40 DA le litre !

Cet exemple concret nous permet de mettre en lumière deux sources d'erreurs différentes. L'une est directement liée à la manière d'exprimer les nombres : on utilise deux chiffres décimaux pour estimer le volume de carburant fourni, deux pour le prix au litre et aucun pour le prix total à payer. L'autre source d'erreur est directement liée aux données qui peuvent elles-mêmes n'être qu'approchées comme c'est le cas ici pour le volume de diesel effectivement délivré par la pompe.

3.1. Représentation des nombres en machine :

A- La virgule flottante :

Un nombre X réel peut être écrit sous la forme standardisé comme suit :

$$x \approx \pm\, m\, b^P$$

$$\approx \pm\, .a_1a_2 \ldots a_{ms}\, b^P$$

Il est constitué d'un signe « + » ou « - » , du point décimal « . » , d'une mantisse « $a_1a_2 \ldots a_{ms}$ », d'une base « b » et d'un exposant « M », sachant que :

a. La base « b » doit être un entier ≥ 2
b. Les « a_i » doivent être des entiers tels que $0 \le a_i < b$
c. L'exposant « P » doit respecter la condition : $P_{min} \le P \le P_{max}$

Un nombre réel non nul est dit normalisé si $a_1 \neq 0$.

Exemple :

$x = -.0231 \times 10^2$ n'est pas normalisé, mais, $x = -.2310 \times 10^1$ l'est.

Le nombre de réels représentables est limité et vaut, pour m, b, P_{max} et P_{min} fixés :

$$2.(b - 1).b^{(ms-1)}.(P_{max} - P_{min}+1) + 1$$ (ici le point . exprime la multiplication)

Le nombre de réels normalisés représentables est :

Exemple :

pour, b=10 , ms=2, $P_{min}=0$, $P_{max}=1$:

$2(10-1)10^{(2-1)}(1-(0)+1)+1 = 180x2+1 = 361$

$$2(b - 1)b^{(ms-1)}(P_{max} - P_{min}+1) + 2$$

car deux représentation de zéro sont prises en compte +.0000... et - .0000...

B- Règle d'arrondissement :

Pour arrondir un nombre jusqu'à n chiffres significatifs (c.s), il faut éliminer les chiffres à droite du N éme chiffre significatif conservé si on se trouve après la virgule, sinon on remplace par des zéros :

1. Si le (N + 1) éme chiffre significatif est > 5, on augmente le n éme chiffre de 1.
2. Si le (N + 1) éme chiffre significatif est < 5, les chiffres retenus restent inchangés.
3. Si le (N + 1) éme chiffre significatif est 5, alors deux cas sont possibles :
   - ✓ Tous les chiffres rejetés, situés après le (N + 1) éme c.s, sont des zéros. les chiffres retenus restent inchangés
   - ✓ Parmi les chiffres rejetés, situés après le (N + 1) éme c.s, il existe au moins un qui soit non nul : On ajoute 1 au N éme chiffre.

Remarque Importante :

Supposons que le nombre est représenté sur machine comme suit avec N chiffres significatifs :

$$x \approx \pm m\, b^p$$

D'où :

$$b^{-1} \leq m < 1.$$

Alors la différence entre la valeur exacte d'un nombre x et la valeur x^* de sa représentation sur machine est appelée erreur d'arrondi. Elle est majorée par :

$$\tfrac{1}{2} b^{-N} b^p,$$

Soit en valeur relative :

$$\left|\frac{x - x^*}{x}\right| \leq \frac{b^{-N} b^p}{2x} \leq \frac{b^{-N} b^p}{2b^{p-1}} = \frac{b^{1-N}}{2}.$$

Exemple : voir section 1.4

3.2. Arithmétique en virgule flottante :

- **Cas 1 :**

  b = 10, ms = 3, P_{min} = -15, P_{max} = 16

  x = +.125 × 10^6, y = -.128 × 10^6, z = +.437 × 10^{12}, u = +.215 × 10^{-10}

  x + y = (+.125 × 10^6) + (-.128 × 10^6) = (+.125 - .128) × 10^6 = -.003 × 10^6

  = -.300 × 10^4 (normalisation)

  x + z = (+.125 × 10^6) + (+.437 × 10^{12}) = (+.000000125 + .437) × 10^{12} (alignement)

  = +.437000125 × 10^{12} = +.437 × 10^{12} (standardisation)

  Remarque 1 : x + z = z, avec x ≠ 0 ! ! !

  x . y = (+.125 × 10^6) × (-.128 × 10^6)

  = (+.125) × (-.128) × 106+6

= -.016000 × 10^{12} = -.160 × 10^{11} (normalisation)
x . z =(+.125 × 10^{6}) × (+.437 × 10^{12})
= (+.125) × (.437) × 10^{6+12}
= +.054625 × 10^{18} = +.546 × 10^{17} (normalisation)

Remarque 2 : l'exposant est supérieur à Mmax et le produit x . z est donc supérieur au plus grand nombre représentable (+.999× 10^{16}) on dit qu'il y a surdépassement. Ceci se traduira par un message d'erreur affiché par le calculateur indiquant qu'il y a ***overflow***.

x ÷ z = (+.125 × 10^{6}) ÷ (+.437 × 10^{12})
= (+.125) ÷ (+.437) × 10^{6-12}
= +.286041189... × 10^{-6} = +.286 × 10^{-6} (normalisation)
u ÷ z = (+.215 × 10-10) ÷ (+.437 × 1012)
= (+.215) ÷ (+.437) × 10-10-12
= +.491990846... × 10^{-22} = +.492 × 10^{-22} (normalisation et arrondissement)

remarque 3 : l'exposant est inférieur à M_{min} et le quotient u ÷ z est donc plus petit que le plus petit nombre positif représentable (+.100 x 10^{-15}) ceci se traduira par un message d'erreur affiché par le calculateur indiquant qu'il y a ***underflow***. Dans pareil cas, selon les logiciels et matériels utilisés, le résultat est remplacé par zéro ou par ε(0). « L'ε(0) machine » dépend du calculateur et du logiciel employés et correspond au plus petit nombre distinguable de zéro (dans la version 7.0.1 de Matlab, implémentée sur PC portable, l'ε(0) machine vaut habituellement 4.940656458412465 × 10^{-324}).

- **Cas 2 :**

Soient b = 10, m = 3, x = y = +.400 × 10^{0} et z = +.100 × 10^{3}
On a :
y + z = +.100 × 10^{3}, ⇒ x + (y + z) = +.100 × 10^{3} = z
x + y = +.800 × 100, ⇒ (x + y) + z = .101 × 10^{3} ≠ x + (y + z).

Remarque 4 : l'ordre dans lequel sont effectuées les opérations peut alterner le résultat final. Par exemple quand on additionne des nombres positifs, il faut commencer par les plus petits le plus possible.

- **Cas 3 :**

La moyenne μ et la variance σ de n nombres s'obtiennent à l'aide des formules suivantes :

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \mu^2$$

Si l'on choisit b = 10, m = 3 et n = 2, les résultats obtenus pour la variance des deux nombres $x_1 = +.310\times10^2$ et $x_2 = +.320\times10^2$, sont $\sigma^2 = +.250\times10^0$ ou $\sigma^2 = -.200\times10^1$ selon la formule utilisée.

Remarque 5 : deux formules théoriquement équivalentes peuvent donner des résultats numériques très différents ! L'erreur est due à la soustraction de deux nombres à peu près égaux. On parle de **cancellassions** ou d'annulation numérique. Elle se traduit par la perte de chiffres significatifs et peut conduire, comme dans cet exemple, à des résultats aberrants : $\sigma^2 < 0$!!

- **Cas 4 :**

$A = +.4025 \times 10^1$, $B = +.4019 \times 10^1 \Rightarrow A - B = C = +.6000 \times 10^{-2}$
les zéros sont-ils significatifs ? Le six est-il significatif ?
Si A et B sont des valeurs approchées et qu'on a en réalité :
$A = +.4024786 \times 10^1$ et $B = +.4019432 \times 10^1$ on obtient alors $C = +.5354 \times 10^{-2}$

Remarque 6 : la soustraction de deux nombres proches peut se traduire par une perte de chiffres significatifs, c'est la cancellation ou l'annulation.

- **Cas 5 :**

Les racines du trinôme du second degré $ax^2 + bx + c$ sont données par les formules équivalentes :

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

$$\text{Si} \quad b^2 \gg ac \Rightarrow x = \frac{-b \pm |b|}{2a} \quad \text{et} \quad x = \frac{2c}{-b \mp |b|}$$

Remarque 7 : on obtient alors une racine «nulle» et l'autre «infinie» ce qui est évidemment aberrant et résulte à nouveau d'une cancellation.

- **Cas 6 :**

$$\sum_{n=1}^{\infty} \frac{1}{n} \text{ est une série divergente mais } \lim_{n\to\infty} \frac{1}{n} \to 0.$$

En base 10, le calcul avec un seul chiffre, (ms = 1) donne une somme qui vaut 2. Avec deux chiffres (ms = 2) on obtient 3.9

Remarque 8 : une série divergente peut sembler converger lorsqu'on calcule numériquement la somme de ses termes. Dans l'exemple précédent, en travaillant avec un seul chiffre la série semble converger vers 2. Toutefois, si on augmente le nombre de chiffres des mantisses utilisées, on obtient des sommes différentes. Tout se passe donc comme si la série admettait plusieurs limites distinctes... Contrairement aux apparences il n'y a donc pas convergence

3.3. Erreur Absolue et Erreur Relative :

Soit *x* la valeur exacte et soit x* la valeur approchée de x alors l'erreur absolue $\Delta(x)$ se calcule comme suit : $\Delta(x)=|x-x^*|$

Et l'erreur relative $\delta(x)$ se calcule par : $\delta(x) = \frac{\Delta(x)}{|x|}$

Remarque : Généralement l'erreur absolue s'avere plus efficace que l'erreur absolue. Par exemple l'erreur absolue entre x=1 et x*=1.1 et la même qu'entre y=1000 et y*=1000.1 alors que l'erreur relative 0.1/1 et différent que celle 0.1/1000.

Exemple1 : Soit x=π et (où **π** = 3.141592653589793 . . .) et π^* = 0.31415927 10^{+1} .Supposons que dans un calcul apparaisse la quantité :

$\Delta\pi = |0.3141592653589793 - 0.31415927|\ 10^{+1} = -0.46410207\ 10^{-7}$

$$\left|\frac{0.3141592653589793\ 10^{+1} - 0.31415927\ 10^{+1}}{0.3141592653589793\ 10^{+1}}\right| = \left|\frac{-0.46410207\ 10^{-8}}{0.3141592653589793}\right|$$

$$= 0.14772828\ 10^{-7} \leq \frac{10^{-7}}{2}$$

Exemple 2 : Pour la valeur exacte x = 2/ 3, la valeur approchée x^*_1 = 0.666667 est 1000 fois plus précise que la valeur approchée x^*_2 = 0.667. En effet, nous avons :

$\Delta1(x) = |x - x^*_1| = |2/3 - 0.666667| = |0.66666667 - 0.66666700| = 0.00000033 = 0.33\ 10^{-6}$

$\Delta2(x) = |x - x^*_2| = |2/3 - 0.667| = |0.66666667 - 0.66700000| = 0.00033333 = 0.33333\ 10^{-3}$

Les erreurs relatives correspondantes sont :

$\delta_1(x) = \frac{\Delta1(x)}{|x|} = \frac{0.33\ 10^{-6}}{|0.66666667|} = 0.5\ 10^{-6} = 0.5\ 10^{-4}$ %,

$\delta_2(x) = \frac{\Delta2(x)}{|x|} = \frac{0.33333\ 10^{-3}}{|0.66666667|} = 0.5\ 10^{-3} = 0.5\ 10^{-1}$ %,

3.3.1. Majoration d'Erreur :

Soit ε l'epsilon absolue machine et εr l'epsilon relative machine alors l'erreur absolue est majorée par :

$$\Delta(x) \leq \varepsilon \quad \text{et} \quad x^* - \varepsilon \leq x \leq x^* + \varepsilon \quad \text{et} \quad x = x^* \pm \varepsilon$$

Et on l'erreur relative est majorée par : $\delta(x) \leq \varepsilon r$ et $x^* = x(1 \pm \varepsilon r)$

3.3.2. Propagation de l'Erreur :

Soit f une fonction de n variable réels (x1 , x2 x3, … x xn).

Evaluons maintenant l'erreur de f(x1 , x2 x3, … x xn) :

Etant les valeurs exactes des xi sont données par la formule approchée :

x*=x+dxi sachant que dxi = x* - x

donc : $f(x*) \approx f(x) + \sum_{i=1}^{n} dx_i \frac{\partial f(x)}{\partial xi}$

et : $\Delta f(x) = |df(x)| = |f(x*) - f(x)| = \left|\sum_{i=1}^{n} \frac{\partial f(x)}{\partial x_i} dx_i\right| \leq \sum_{i=1}^{n} \left|\frac{\partial f(x)}{\partial x_i}\right| |dx_i|$

sachant que |dxi| = Δ(xi) et $df(x) = f(x) - f(x*)$

Ainsi pour l'erreur relative on aura :

$$\left|\frac{\Delta f(x)}{f(x)}\right| = \left|\sum_{i=1}^{n} \frac{x_i}{f(x)} \frac{\partial f(x)}{\partial x_i} \frac{dx_i}{x_i}\right| \leq \sum_{i=1}^{n} \left|\frac{x_i}{f(x)} \frac{\partial f(x)}{\partial x_i}\right| \left|\frac{dx_i}{x_i}\right|$$

En pratique les valeurs des dérivées de la fonction sont évaluées selon la valeur x*.

- **Exemple d'application sur l'addition** :

  Soit la fonction : f(x,y) = x ± y.

  Alors : d(x±y) ≤ d(x) + d(y) et Δ (x±y) ≤ Δ (x) + Δ (y)

$$\left|\frac{d(x \pm y)}{(x \pm y)}\right| \leq \left|\frac{x}{x \pm y}\right| \left|\frac{dx}{x}\right| + \left|\frac{y}{x \pm y}\right| \left|\frac{dy}{y}\right|$$

Exemple de calculs : Soient x∗ = 255 et y∗ = 250 avec δx = δy = 0, 1 % = 10^{-3}.
Question : δ(x-y) =?
Nous avons d'abord : Δx = x∗.δx = 0,255, Δy = y∗.δy = 0,250.
et puis x∗ - y∗ = 5 avec une erreur relative :
δ(x-y) =Δ(x - y) / |x∗ - y∗| ≤ Δx + Δy / |x∗ - y∗| = 0.101 10^0 = 10,1%.
On remarque que x∗ et y∗ sont 101 fois plus précis pour x et y (respectivement) que (x∗-y∗) l'est pour (x – y).

- **Exemple d'application 2 :** reprendre l'exemple précédent en appliquant le calcul suivant : $\delta(x.y)$ = ?

4. Stabilité et analyse d'erreur des méthodes numériques et conditionnement d'un problème

4.1. Conditionnement

Le conditionnement décrit la sensibilité de la valeur d'une fonction à une petite variation de son argument, c'est-à-dire :

$$\frac{f(x) - f(x*)}{f(x)} \text{ en fonction de } \frac{x - x*}{x}$$

Lorsque ($x - x*$) est petit. Pour une fonction suffisamment régulière, on a évidemment :

$$\left| \frac{f(x) - f(x*)}{f(x)} \Big/ \frac{x - x*}{x} \right| \simeq \left| \frac{xf'(x)}{f(x)} \right|.$$

D'où on tire :

Définition : On appelle conditionnement d'une fonction numérique f de classe C^1 en un point x, le nombre :

$$cond(f)_x := \left| \frac{xf'(x)}{f(x)} \right|.$$

Exemple :

$$f(x) = \sqrt{x}$$

$$\left| \frac{xf'(x)}{f(x)} \right| = \frac{1}{2}.$$

Ceci correspond à un bon conditionnement, puisque l'erreur relative sur f sera au plus moitié d'une erreur relative sur x.

Exemple : $f(x) = a - x$

$$\left| \frac{xf'(x)}{f(x)} \right| = \left| \frac{x}{a - x} \right|.$$

Ici, le conditionnement est très mauvais si x est voisin de a.

4.2. Stabilité

Définition : La stabilité décrit la sensibilité d'un algorithme numérique pour le calcul d'une fonction f(x).

Exemple :

$$f(x) = \sqrt{x+1} - \sqrt{x}.$$

Le conditionnement de cette fonction est égal à :

$$\left| \frac{xf'(x)}{f(x)} \right| = \left| \frac{x\left(\sqrt{x} - \sqrt{x+1}\right)}{2\sqrt{x(x+1)}} - \frac{1}{\sqrt{x+1} - \sqrt{x}} \right| = \frac{1}{2}\sqrt{\frac{x}{x+1}}$$

Cette dernière expression étant proche de 1/2 pour x grand. Donc, si x est grand, le conditionnement de f est bon. Cependant, dans un calcul à 6 chiffres significatifs, on a :

$$f\,(12345) = \sqrt{12346} - \sqrt{12345} = 111,113 - 111,108 = 0,500000.10^{-2}.$$

Or un calcul précis donne : f(12345) = 0,4500032.... 10^{-2}. On a donc une erreur relative de 10% ce qui est important et peu en accord avec le bon conditionnement de f. Ceci est dû à l'algorithme utilisé dans ce calcul que l'on peut expliciter comme suit :

$$\begin{aligned} x_0 &: = 12345 \\ x_1 &: = x_0 + 1 \\ x_2 &: = \sqrt{x_1} \\ x_3 &: = \sqrt{x_0} \\ x_4 &: = x_2 - x_3 \end{aligned}$$

Il y a quatre fonctions à intervenir et, à priori, même si le conditionnement de f est bon, il se peut que le conditionnement d'une ou plusieurs fonctions utilisées dans l'algorithme soit supérieur à celui de f. C'est ce qui se produit ici pour la fonction $x_3 \rightarrow x_4 = x_2 - x_3$ (x_2 étant supposé fixe) dont le conditionnement est grand lorsque x_3 est voisin de x_2 comme on l'a vu précédemment.

En conclusion, le choix d'un bon algorithme numérique est essentiel. Par exemple, ci dessus, un meilleur algorithme est obtenu en utilisant :

$$f\,(x) = \sqrt{x+1} - \sqrt{x} = \frac{1}{\sqrt{x+1} + \sqrt{x}}.$$

Dans ce cas, toujours avec 6 chiffres significatifs :

$$f(12345) = \frac{1}{\sqrt{12346} + \sqrt{12345}} = \frac{1}{222,221} = 0,450002.10^{-2}$$

ce qui donne une erreur relative de 0.0003 %.

Exemple d'instabilité : Calcul de e^{-12} à l'aide de sa série de Taylor

$$e^x = \sum_{n=0}^{N} \frac{x^n}{n!} \,(= S_N) \ \text{ pour } N \text{ grand.}$$

Exercice : Calculer à l'aide de Scilab les valeurs successives de cette somme en fonction de *N* pour *x* = -12. À dix chiffres significatifs. Puis à refaire les calculs avec la formule suivante :

$$e^{-x} = \frac{1}{e^x}$$

Laquelle des deux formule est plus stables ?

1. Conclusion :

Dans la vie réelle plusieurs catastrophes ont été généralement le résultat inévitable d'une mauvaise gestion de l'arithmétique des ordinateurs ou calculateurs (erreurs d'arrondis, d'annulation ou concellation). Voici quelques exemples qui ont marqués la vie humaine :

1.1. Missile Patriot

En février 1991, pendant la Guerre du Golfe, une batterie américaine de missiles Patriot, à Dharan (Arabie Saoudite), a échoué dans l'interception d'un missile Scud irakien. Le Scud a frappé un baraquement de l'armée américaine et a tué 28 soldats. La commission d'enquête a conclu à un calcul incorrect du temps de parcours, dû à un problème d'arrondi. Les nombres étaient représentés en virgule fixe sur 24 bits, donc 24 chiffres binaires. Le temps était compté par l'horloge interne du système en 1/10 de seconde. Malheureusement, 1/10 n'a pas d'écriture finie dans le système binaire : 1/10 = 0,1 (dans le système décimal) = 0,00011001100110011001100110011... (dans le système binaire). L'ordinateur de bord arrondissait 1/10 à 24 chiffres, d'où une petite erreur dans le décompte du temps pour chaque 1/10 de seconde. Au moment de l'attaque, la batterie de missile Patriot était allumée depuis environ 100 heures, ce qui avait entraîné une accumulation des erreurs d'arrondi de 0,34s. Pendant ce temps, un missile Scud parcourt environ 500 m, ce qui explique que le Patriot soit passé à côté de sa cible. Ce qu'il aurait fallu faire c'était redémarrer régulièrement le système de guidage du missile.

1.2. Explosion d'Ariane 5

Le 4 juin 1996, une fusée Ariane 5, a son premier lancement, a explosé 40 secondes après l'allumage. La fusée et son chargement avaient coûté 500 millions de dollars. La commission d'enquête a rendu son rapport au bout de deux semaines. Il s'agissait d'une erreur de programmation dans le système inertiel de référence. À un moment donné, un nombre codé en virgule flottante sur 64 bits (qui représentait la vitesse horizontale de la fusée par rapport à la plate-forme de tir) était converti en un entier sur 16 bits. Malheureusement, le nombre en question était plus grand que 32768 (overflow), le plus grand entier que l'on peut coder sur 16 bits, et la conversion a été incorrect.

1.3. Bourse de Vancouver

En 1982, la bourse de Vancouver a crée un nouvel indice avec une valeur nominale de 1000. Après chaque transaction boursière, cet indice était recalculé et tronqué après le troisième chiffre décimal et, au bout de 22 mois, la valeur obtenue était 524.881, alors que la valeur correcte était 1098.811. Cette différence s'explique par le fait que toutes les erreurs d'arrondi étaient dans le même sens : l'opération de troncature diminuait à chaque fois la valeur de l'indice.