Chapitre 2

Calcul numérique sur calculateur

1 Introduction :

Le calcul numérique, comme son nom peut le laisser deviner, est un grand utilisateur et manipulateur de nombres. Chacun connaît les trois catégories de nombres couramment utilisés : les entiers, dont l'ensemble $\mathbb{Z}$ forme un anneau, les rationnels et les réels qui sont des corps, respectivement notés $\mathbb{Q}$ et $\mathbb{R}$. Ces trois ensembles sont imbriqués les uns dans les autres :

$$\mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}$$

De nombreuses applications pratiques feront intervenir des nombres complexes. Leur ensemble $\mathbb{C}$ est également un corps qui contient les trois ensembles précédents.

Le calcul numérique est le but de l'analyse numérique qui peut schématiquement être partagée en trois domaines qui ne sont pas indépendants.

1. Etude mathématique du problème donné qui consiste bien généralement à savoir s'il a une solution et si, de plus, celle-ci est unique
2. Elaboration d'un algorithme de calcul d'une solution "approchée".
3. Etude de la stabilité théorique des résultats.

Les points 2 et 3 constituent le calcul numérique.

Nous avons mentionné en 2 : solution "approchée". Il est nécessaire de préciser les différents sens qui peuvent affecter le qualificatif "approché". L'algorithme conçu peut être formée d'une succession infinie d'opérations qui ne pourra bien évidemment être menée à terme. Il faut s'arrêter en temps fini. La solution ne sera qu'approchée.

Exemple 1.1

Calcul de $e = \sum_{i=0}^{\infty} \frac{1}{i!}$

On pourra calculer une valeur approchée de e par un algorithme du type suivant, en ayant pris soin de fixer N :

```
E = 0
F = 1
For I = 0 to N
E = E + 1/F
F = F * (I+1)
Next I
```

On voit également sur cet exemple qu'il y a un lien entre le nombre de décimales exactes souhaité et la valeur que l'on attribuera à N. Il illustre aussi les premiers problèmes du calcul numérique sur ordinateur. Si on suit étape après étape les calculs à effectuer, il est clair que nous obtenons les valeurs suivantes pour F :

1, 1, 2, 6, 24, $\cdots$

Pour obtenir E, nous exécuterons les opérations suivantes itération après itération :

```
E = 0 + 1 = 1
E = 1 + 1 = 2
E = 2 + 1/2 = 2.5
E = 2.5 + 1/6 = 2.5 + 0.1666...
```

Et voici qu'apparaît la deuxième signification que l'on peut donner à "approché". Il n'est pas possible de conserver la suite infinie de 6 qui apparaissent dans la représentation décimale du rationnel 1/6.

On pourrait à la rigueur concevoir de donner à chaque étape l'expression rationnelle de E.

```
E = 1
E = 2
E = 5/2
E = 32/12 = 8/3, ...
```

Puis, lorsque ce rationnel est suffisant pour obtenir le nombre souhaité de décimales exactes, la division du numérateur par le dénominateur est effectuée. Cette technique est parfois utilisée pour effectuer des calculs avec la précision maximum. Elle a l'inconvénient d'être très coûteuse en temps de calcul.

En fait, on préférera bien souvent limiter les développements décimaux des rationnels ou des réels. Il faut donc savoir apprécier le nombre de décimales à employer dans chacun des calculs intermédiaires pour être sûr que le résultat final aura un nombre donné de décimales exactes.

Exemple 1.2

Calcul de e avec 3 décimales exactes après la virgule. Nous rappelons que $e =$ 2. 718 281 828 5

$\frac{1}{F}$	E
$\frac{1}{2} = 0.5$	2.5
$\frac{1}{6} = 0.1666...$	2.6666...

Puisqu'il faut prendre un nombre limité de décimales, par exemple 3 après la virgule, doit-on considérer que 1/6 sera approché par :

0.166, ce qui constitue une troncature ou par

0.167, ce qui est un arrondi.

Le but étant de faire comprendre comment fonctionne l'arithmétique d'un calculateur, nous adopterons la règle la plus fréquemment utilisée par ceux-ci qui est l'arrondi. On augmente la dernière décimale conservée, si le premier chiffre négligé est supérieur ou égal à 5, sinon elle reste inchangée.

Il est ainsi parfaitement évident que nous n'obtiendrons pas 3 décimales exactes après la virgule pour eux, si nous employons seulement 3 décimales après la virgule dans chaque calcul de $\frac{1}{F}$. Il en faudra quatre et nous aurons :

$\frac{1}{F}$	E
$\frac{1}{2} = 0.5$	2.5
$\frac{1}{6} \approx 0.1667$	2.6667
$\frac{1}{24} \approx 0.0417$	2.7084
$\frac{1}{120} \approx 0.0083$	2.7167
$\frac{1}{720} \approx 0.0014$	2.7181
$\frac{1}{5040} \approx 0.0002$	2.7183

2 Calcul numérique avec les nombres entiers :

2.1 Calcul avec les nombres entiers sur ordinateur

Lorsqu'on utilise de "grands" nombres entiers, les résultats informatiques (supérieurs à 32767) sont donnés par un calcul modulaire (modulo 65536).
Exemple : les variables sont déclarées réelles

```
debut
a := 2.0 * 30000
b := 2 * 30000
ecrire a, b, b + 65536
fin
```

Ce programme exécuté donne les valeurs suivantes :

```
a = 6.0E04
b = -5.536E03
b + 65536 = 6.0E04
```

Conclusion

L'expression 2.0 * 30000 est réelle et donne résultat mathématique attendu. L'expression 2 * 30000 est entière (2 et 30000 entiers) et le calcul arithmétique dépasse 32767 et renvoie un résultat modulo 65536 (avec un message d'erreur suivant le compilateur utilisé). Lorsqu'il y a ce dépassement, il faut déclarer les entiers en réels.

2.2 Calcul du plus grand entier-machine positif

Si on prend en compte ce calcul modulaire, le plus grand entier correspond au dernier entier calculé, et en rajoutant 1 à sa valeur, devient plus petit que le précé dent.
Exemple : les variables sont déclarées réelles

```
debut
a := 0
b := 1
```

```
  tant que b > a faire
    debut
      a := a + 1
      b := b + 1
    fin
  ecrire a, b
fin
```

Ce programme exécuté donne le plus grand entier positif 32767 (contenu dans **a**) et **b** a pour valeur -32768.
Le calcul s'est fait de la manière suivante :
$b = a + 1 = 32767 + 1 = 32768$ - 65536 = -32768 et l'inégalité $b > a$ n'est plus vérifiée et **a** contient le plus grand entier positif. On remarque d'une part si on modifie le programme que les entiers-machine sont dans l'intervalle $[-32768, 32767]$ et d'autre part que 32767 et 65536 sont respectivement les valeurs de $2^{15} - 1$ et 2^{16}, issues de la représentation binaire de ces entiers.

3 Calcul numérique avec les nombres réels :

3.1 Calcul avec les nombres réels sur ordinateur : le plus petit réel-machine positif

Le calcul se fait en s'approchant le plus près de zéro par valeur positive décroissante jusqu'à son obtention.
Exemple : les variables sont déclarées (**a** de type réel et **n** de type entier)

```
debut
  a := 1
  n := 0
  tant que a > 0 faire
    debut
      a := a/10
      n := n + 1
    fin
  ecrire n - 1
fin
```

Le programme exécuté donne l'ordre de grandeur du plus petit réel-machine positif (10^{-324}). Le test d'arrêt indique qu'à la fin de l'exécution, la variable **a** a pour valeur zéro.

3.2 Calcul de l'ϵ-machine

Définition : l'ϵ-machine est le plus petit réel-machine positif, noté ϵ, tel que sur machine $1 + \epsilon > 1$, noté $1 \oplus \epsilon > 1$.
En pratique, l'ϵ-machine n'est jamais le plus petit réel-machine positif. L'exemple suivant montre, en outre, la "non-associativité" de l'addition-machine, noté $\oplus$:
Soit x un réel-machine positif tel que $0 < x < \epsilon$
par définition de l'ϵ-machine on a sur machine

$$(\cdots((1 + x) + x) + \cdots + x) = 1$$

Mais si on fait d'abord suffisamment de fois l'addition-machine des x, on aura :

$$(1 + (x + \cdots + (x + x) \cdots)) > 1$$

Exemple : La variable est déclarée réelle

```
debut
  z := 1
  tant que 1 + z > 1 faire z := z/2
  ecrire 2*z
fin
```

Le programme exécuté donne l'ϵ-machine $(2.22E - 16)$. Si on modifie le programme en prenant un compteur la valeur de **2*z** est égale à 2^{-52}, issue de la représentation binaire des réels-machine.