CAHIERS DU SÉMINAIRE D'HISTOIRE DES MATHÉMATIQUES

J. VIGNES

Zéro mathématique et zéro informatique

Cahiers du séminaire d'histoire des mathématiques 1re série, tome 8 (1987), p. 25-42

<http://www.numdam.org/item?id=CSHM_1987__8__25_0>

ZERO MATHEMATIQUE ET ZERO INFORMATIQUE

J. VIGNES ★
Professeur à l'Université Pierre et Marie Curie et C.N.R.S.
Conseiller Scientifique à l'Institut Français du Pétrole.

I - INTRODUCTION

Une controverse existe à propos de la première apparition du zéro, dans la numération de position. Certains pensent que ce sont les babyloniens [3,4],d'autres les indiens [7] qui furent les premiers à introduire un caractère signifiant l'absence d'unité d'un certain rang dans la numération de position. Mais ce zéro n'était pas alors pensé dans le sens de rien ou de valeur nulle. Il fallut attendre plusieurs siècles pour que ce nouveau concept apparaisse. Il est très difficile de dire qui l'a introduit pour la première fois. Certains pensent qu'on le doit à *Diophante d'Alexandrie* qui vécu croit-on au 4ème siècle. Mais ces travaux restèrent quasiment inconnus. Un demi-millénaire plus tard vers 825 *Muhammad Ibn Musa AL-Khawarîsmî* écrivit un traité d'algèbre dont la traduction en latin, faite par *Abelard de Bath* vers 1120 parue sous le titre *"Liber Algorismi de numéro Indorum"*; permit de faire connaître en Occident le concept du zéro, valeur nulle, et les techniques algébriques qui en découlent. Le zéro joue un rôle capital en mathématiques, montrons ici comment il est utilisé pour contrôler la solution de problèmes d'analyse numérique.

★ Conférence donnée le 11 décembre 1985 au Séminaire d'Histoire des mathématiques.

II - ROLE DU ZERO EN TANT QU'OUTIL DE CONTROLE DE LA SOLUTION DE PROBLEMES D'ANALYSE NUMERIQUE

Un grand nombre de problèmes d'analyse numérique peuvent être classés comme "Problème à solution contrôlable". Cela veut dire que la solution étant obtenue, par une méthode quelconque, on peut contrôler sa validité en calculant la valeur d'une fonction.

Deux cas peuvent être rencontrés :

- la valeur de cette fonction est égale à zéro, alors on peut affirmer que l'on a obtenu la solution du problème cherché,

- la valeur de cette fonction est différente de zéro, alors on peut affirmer que l'on n'a pas résolu le problème.

Donnons ici quelques exemples :

a) La détection de la singularité d'une matrice carrée $\mathcal{A}$ d'ordre n.

Considérons une matrice réelle $\mathcal{A}$ d'ordre n.

Calculons par un algorithme quelconque la valeur Δ de son déterminant

- si $\Delta = 0$ alors $\mathcal{A}$ est singulière et non inversible
- si $\Delta \neq 0$ alors $\mathcal{A}$ est régulière et inversible.

b) Le calcul de la matrice $\mathcal{B}$ inverse de $\mathcal{A}$.

Soient $\mathcal{B}$ la matrice réelle d'ordre n obtenue par un algorithme quelconque de calcul de matrice inverse et I la matrice unité d'ordre n.

Calculons la matrice $\mathcal{C}$ telle que : $\mathcal{C} = \mathcal{A} . \mathcal{B} - I$

- si $\mathcal{C} = 0$ alors $\mathcal{B}$ est bien l'inverse de $\mathcal{A}$
- si $\mathcal{C} \neq 0$ alors $\mathcal{B}$ n'est pas l'inverse de $\mathcal{A}$.

c) Le calcul des valeurs propres d'une matrice $\mathcal{A}$ et des vecteurs propres associés

Soient $\lambda_i \in \mathbb{R}$ ou $\mathbb{C}$, $i=1,\ldots,n$, les valeurs propres de $\mathcal{A}$ et $v_i \in \mathbb{R}^n$ ou $\mathbb{C}^n$ les vecteurs propres associés obtenus par un algorithme quelconque.

Calculons, pour $i=1,\ldots,n$, les valeurs D_i des déterminants des matrices $\mathcal{M}_i$ telles que $\mathcal{M}_i = \mathcal{A} - \lambda_i I$, ainsi que les vecteurs R_i définis par : $R_i = \mathcal{A} v_i - \lambda_i v_i$

- si $D_i = 0$ alors λ_i est bien valeur propre de $\mathcal{A}$
- si $D_i \neq 0$ alors λ_i n'est pas valeur propre de $\mathcal{A}$
- si $R_i = 0$ alors v_i est bien vecteur propre associé à λ_i
- si $R_i \neq 0$ alors v_i n'est pas vecteur propre associé à λ_i .

d) La résolution des systèmes d'équations

Soit un système d'équations non linéaires d'ordre n défini par

$\mathcal{F}(x) = 0$, $\mathcal{F}$ étant une application de $\mathbb{R}^n$ dans $\mathbb{R}^n$.

Soit $x^* \in \mathbb{R}^n$ le vecteur obtenu par un algorithme quelconque de résolution des systèmes non linéaires.

Calculons la valeur du vecteur $v \in \mathbb{R}^n$ défini par :

$v = \mathcal{F}(x^*)$

- si v = 0 alors x^* est une solution du système non linéaire
- si $v \neq 0$ alors x^* n'est pas une solution du système non linéaire.

e) La résolution des problèmes d'optimisation non contraints

Soit à résoudre le problème :

$\underset{\text{Max}}{\text{Min}}\ \Phi(x)$ $\quad x \in R^n$, $\quad \Phi$ étant une application

de $R^n \Longrightarrow R$, Φ est supposée unimodale et ne présentant pas de point selle.

Soit $x^* \in R^n$ le vecteur obtenu par un algorithme quelconque d'optimisation. Calculons la valeur du vecteur $u \in R^n$ défini par :

$$U = \text{grad}\ \Phi\ (x^*)$$

- si u = 0 alors x^* est bien l'extremum de Φ
- si $u \neq 0$ alors x^* n'est pas l'extremum de Φ.

Comme nous le voyons dans cette liste d'exemples, qui bien évidemment n'est pas exhaustive, l'égalité à zéro d'une ou plusieurs fonctions, calculées avec les résultats fournis par des algorithmes de résolutions, permet de dire si l'on a ou non obtenu la solution du problème cherché. Mais encore faut-il être capable de calculer correctement la valeur de ces fonctions.

Du point de vue de l'analyse numérique, les algorithmes de résolution étant généralement conçus dans le domaine du continu, pour être exécutés avec une arithmétique exacte, le problème ne se pose pas.

Du point de vue de l'informatique il en est tout autrement puisque, lorsque ces algorithmes sont mis en oeuvre sur ordinateurs, ils sont projetés dans le domaine du discret et exécutés avec une arithmétique qui n'est qu'une approximation de l'arithmétique exacte. Ainsi tout résultat de calcul est-il entaché d'une erreur. Le problème de la validité des résultats des fonctions calculées est ici crucial, c'est d'ailleurs ce qui différencie essentiellement l'informatique numérique de l'analyse numérique.

III - L'ARITHMETIQUE INFORMATIQUE ET SES CONSEQUENCES

III - 1. La représentation des éléments de $\mathbb{R}$

Tout élément $x \in \mathbb{R}$, s'écrit en virgule flottante normalisée sous la forme :

$$x = m.b^e$$

m étant la mantisse illimitée telle que $\frac{1}{b} \leqslant m < 1$

e étant l'exposant entier $e \in \mathbb{Z}$.

La représentation machine de $x \in \mathbb{R}$ est $X \in \mathbb{F}$ définie par :

$$X = M.b^E$$

M étant la mantisse limitée à p chiffres dans la base b

E = e étant l'exposant entier.

$\mathbb{F}$ est donc l'ensemble des nombres informatiques codés en virgules flottantes normalisé.

Du fait que M est limitée, X n'est qu'une approximation de x.

On peut définir les erreurs relatives α_i de ces représentations dues aux mantisses limitées par :

$$\alpha_1 = \frac{X-x}{X} = -\frac{r}{M} \qquad \alpha_2 = \frac{X-x}{x} = -\frac{r}{m}$$

r étant la partie perdue par la mantisse.

Il a été montré [1,5,8] que les α_i i=1,2 pouvaient être considérées comme des variables aléatoires appartenant à des populations statistiques $P(\alpha)$ dont on a calculé la moyenne et l'écart type.

III.2 L'arithmétique virgule flottante normalisée

Lorsque l'on veut sur ordinateur effectuer une opération algébrique

$z = x \, w \, y \quad x,y,z \in \mathbb{R} \quad w \in [+, -, x, :]$

(les éléments w étant les opérateurs arithmétiques exacts)

c'est en fait :

$Z = X \, \Omega \, Y \quad X,Y,Z \in \mathbb{F} \quad \Omega \in [\oplus, \ominus, *, /]$

(les éléments Ω étant les opérateurs informatiques) qui est exécutée.

L'erreur absolue ε_Z appelée erreur d'arrondi est définie par :

$$\varepsilon_Z = Z - z = (x + \varepsilon_x) \, w \, (y + \varepsilon_y) - x \, w \, y + \alpha(X \, \Omega \, Y)$$

ε_x, ε_y étant les erreurs absolues sur X et Y et $\alpha \in P(\alpha)$,

Chaque opération informatique n'étant qu'une approximation de l'opération algébrique, au fur et à mesure du déroulement d'un calcul sur ordinateur la propagation des erreurs d'arrondi fait que tout résultat est entaché d'une erreur qui parfois peut rendre le résultat fourni complètement faux, comme nous allons le voir.

III.3. Conséquence de la propagation des erreurs d'arrondi

Considérons la matrice singulière $\mathcal{A}$ d'ordre 2 suivante :

$$\mathcal{A} = \begin{bmatrix} 3 & 5 \\ 3.10^6 & 5.10^6 \end{bmatrix} .$$

Calculons par la méthode de Gauss sur un ordinateur travaillant en arithmétique tronquée décimale avec 6 chiffres de mantisse, la valeur Δ de son déterminant. La matrice informatique A correspondant à $\mathcal{A}$ est la suivante :

$$A = \begin{bmatrix} 0.300000\ 10^1 & 0.500000\ 10^1 \\ 0.300000\ 10^7 & 0.500000\ 10^7 \end{bmatrix} .$$

Nous remarquons que les coefficients de $\mathcal{A}$ tombant ici juste en machine on a

$$A = \mathcal{A} .$$

La valeur du déterminant est donnée par :

$$\Delta = \prod_{i=1}^{2} a_{ii}^{(i)} = a_{11}^{(i)} * a_{22}^{(2)} \qquad a_{11} = 0.300000\ 10^1 = 3 .$$

L'algorithme de Gauss nous donne :

$$\begin{bmatrix} 0.100000\ 10^1 & 0.166666\ 10^1 \\ 0. & 0.500000\ 10^7 - 0.499998\ 10^7 \end{bmatrix} =$$

$$\begin{bmatrix} 0.100000\ 10^1 & 0.166666\ 10^1 \\ 0. & 0.200000\ 10^2 \end{bmatrix}$$

d'où $a_{22}^{(2)} = 0.2000000\ 10^2 = 20$ et $\Delta = 3 * 20 = 60$.

La matrice informatique A pourtant identique à $\mathcal{A}$ paraît comme régulière ce qui est tout à fait faux.

Que s'est-il passé ? La valeur du deuxième pivot $a^{(2)}_{22}$ résulte de l'effet cumulé des erreurs d'arrondi et est donc non significative. Malgré sa valeur intrinsèque cette valeur doit être considérée comme nulle. Ceci montre bien le nécessité d'élaborer un nouveau concept, celui du zéro informatique.

IV. LE CONCEPT DU ZERO INFORMATIQUE

Il est bien évidemment lié à la propagation des erreurs d'arrondi due à l'arithmétique de l'ordinateur.

Nous définissons le zéro informatique de la façon suivante :

Soit $R \in \mathbb{F}$ la valeur du résultat d'un calcul sur ordinateur

- si $R = 0$ ou bien
- si R est quelconque mais non significatif

alors R doit être considéré comme un zéro informatique, noté $\underline{0}$.

Au concept mathématique classique de "nullité" vient s'ajouter celui de "sans signification". Une valeur $R \in \mathbb{F}$ résultant d'un calcul sur ordinateur est non significative si elle ne représente que l'effet cumulé des erreurs d'arrondi. Cela veut dire que ε etant l'erreur absolue qui l'entache, erreur due à la propagation des erreurs d'arrondi, ε est du même ordre de grandeur que la valeur de R. En effet le nombre de chiffres décimaux significatifs C_R de R est donné par :

$$C_R = \log \frac{|R|}{\varepsilon}$$

si $\varepsilon \geq |R|$ alors $C_R \leq 0$, R est bien dépourvue de sens puisque ne possédant pas un seul chiffre décimal significatif.

Remarquons que si $\varepsilon = 0$ alors $C_R = \infty$, autremant dit R est connu avec une précision infinie. En informatique la précision infinie est égale en fait au nombre C_{max} de chiffres décimaux qui peuvent être codés dans la mantisse et qui est défini par : $10^{max} = b^p$ b étant la base dans laquelle travaille la machine et p le nombre de chiffres dans cette base que l'on peut coder dans la mantisse.

Ainsi donc, pour pouvoir dire si un résultat de calcul sur ordinateur est significatif ou non, faut-il être capable de bien estimer l'erreur d'arrondi ε qui l'entache.

Nous avons créé une méthode d'analyse de la propagation des erreurs d'arrondi, appelée méthode de permutation-perturbation qui permet d'estimer l'erreur ε qui entache tout résultat d'algorithme quelle qu'en soit sa complexité.

V. LA METHODE DE PERMUTATION-PERTURBATION

Nous ne présentons ici que les idées de base de cette méthode développée par M. La Porte et J. Vignes et qui a déjà été présentée dans de nombreux articles notamment dans [2, 9, 10] et qui est aussi connue sous le nom de **CESTAC** (Contrôle et Estimation **STo**chastique des **A**rrondis de **C**alculs).

V.1. Aspect théorique de la méthode

V.1.1. La méthode de permutation

Considérons un algorithme numérique nécessitant k opérations élémentaires (affectations, +, -, x, :) et fournissant un résultat unique $r \in \mathbb{R}$ obtenu, à partir de données $d \subset \mathbb{R}$.

Comme l'arithmétique informatique ne respecte par les règles de l'arithmétique exacte (non associativité de l'addition par exemple), il n'existe pas une procédure informatique image unique de la procédure algébrique mais un ensemble $\mathcal{P}$ de procédures informatiques P_i chacune aussi représentative l'une que l'autre de la procédure algébrique. Ces procédures sont obtenues en faisant les Cop permutations possibles des opérateurs permutables. On a donc :

$$\text{Card } \mathcal{P} = \text{Cop} .$$

V.1.2. La méthode de perturbation

Pour tout résultat d'une opération arithmétique, il y a deux résultats informatiques (l'un par défaut, l'autre par excès) qui sont tous les deux aussi légitimement représentatifs du résultat arithmétique exact. Ainsi considérons une procédure informatique P_i parmi celles définies ci-dessus qui nécessite l'exécution de k opérations élémentaires pour calculer R. En fait il existe un ensemble Γ de résultats informatiques, dont tous les éléments R_j sont obtenus en faisant toutes les combinaisons des opérations arithmétiques par défaut et par excès, chacun représentant aussi légitimement le résultat arithmétique exact. Le cardinal de Γ est défini par :

$$\text{Card } \Gamma = 2^k$$

V.1.3. La méthode de permutation-perturbation

Si l'on veut obtenir tous les résultats informatiques images du résultat unique exact r, il faut appliquer la méthode de perturbation à toutes les images informatiques de la procédure algébrique. Ainsi le cardinal de l'ensemble $\mathcal{R}$ des résultats informatiques images du résultat algébrique unique r est défini par :

$$\text{Card } \mathcal{R} = 2^k \text{ Cop} .$$

V.1.4. Estimation du nombre de chiffres décimaux significatifs d'un résultat informatique

Il a été montré [2,6] que les $R_i \in \mathcal{R}$ peuvent être considérées comme une variable aléatoire approximativement gaussienne de moyenne $\overline{R}$ et d'écart type σ et que la meilleure estimation de $r \in \mathbb{R}$ est $\bar{R}$. Ainsi le nombre de chiffres décimaux significatifs de $\bar{R}$ est défini par :

$$C_{\bar{R}} = \log \frac{|\bar{R}|}{\sigma}$$

V.2. Aspect pratique de la méthode

Il a été montré [2, 6] qu'en pratique il suffit d'une sous-population de 3 voire 2 éléments pour estimer les nombres $C_{\bar{R}}$ de chiffres décimaux significatifs à 0.5 ou 1 près, avec un degré de confiance de 95%.

Ces 2 ou 3 éléments sont obtenus en faisant exécuter la même procédure image P_i, 2 ou 3 fois, en perturbant aléatoirement le résultat de chaque opération arithmétique au dernier bit et le cas échéant en permutant aléatoirement les termes des formes linéaires de la procédure. Ces permutations et perturbations aléatoirement sont faites par un logiciel qui ne nécessite qu'une trentaine d'instructions.

A partir de 2 ou 3 résultats obtenus on en calcule la moyenne qui est considérée comme le résultat informatique, l'écart type et le nombre de chiffres décimaux sur le résultat. Le logiciel complet ne nécessite qu'une cinquantaine d'instructions.

Ainsi a-t-on un outil puissant, applicable à n'importe quel algorithme aussi complexe soit-il, qui permet, pour tout résultat, d'estimer son nombre de chiffres décimaux significatifs et par conséquent de dire si une valeur informatique, quelle qu'elle soit, doit être considérée, ou non, comme un zéro informatique.

VI. UTILISATION DE LA METHODE CESTAC POUR LA DETECTION DU ZERO INFORMATIQUE

VI.1 Détection de la singularité des matrices

Pour illustrer tout d'abord l'efficacité de la méthode, utilisons-la pour la calcul du déterminant de la matrice $\mathcal{A}$ d'ordre 2 précédemment définie. Calculons 2 fois le déterminant, par la méthode de Gauss, en perturbant aléatoirement les résultats des opérations arithmétiques.

Le premier calcul nous a donné $\Delta_1 = 60$ (voir ci-dessus). Le deuxième nous donne :

$$\begin{bmatrix} 0.100000\ 10^{1} & & 0.166667\ 10^{1} \\ 0. & 0.500000\ 10^{7} - & 0.500001\ 10^{7} \end{bmatrix} =$$

$$\begin{bmatrix} 0.100000\ 10^{1} & 0.166667\ 10^{1} \\ 0. & -\ 0.100000\ 10^{2} \end{bmatrix}$$

d'où $a_{22}^{(2)} = -\ 10$ et $\Delta_2 = -\ 3 * 10 = -\ 30$ $\quad \overline{\Delta} = 15 \quad \sigma = 63.6$

$C_{\overline{\Delta}} = -\ 0.63 < 1 \Longrightarrow \overline{\Delta} = \underline{0}$.

Par conséquent la matrice A doit bien être considérée comme informatiquement singulière.

Considérons maintenant la matrice de Hilbert d'ordre n définie par :

$$\mathcal{H}_n = [a_{ij}] \in R \qquad a_{ij} = \frac{1}{i+j-1}\ , \quad i,j = 1,\ldots,n\ .$$

Du point de vue mathématique elle est toujours régulière quelle que soit la valeur de $n \in \mathbb{N}$, mais elle est de plus en plus mal conditionnée lorsque la valeur de n croît.

Du point de vue informatique elle finit par être informatiquement singulière à partir d'un certain ordre qui dépend de la précision de l'arithmétique de l'ordinateur.

Trouver cet ordre est un problème difficile qui est très aisément résolu par la méthode **CESTAC.**

En calculant trois valeurs du déterminant de cette matrice et le nombre de chiffres décimaux significatifs de sa moyenne on trouve :

- Pour un ordinateur travaillant en arithmétique binaire tronquée avec 36 bits de mantisse pour n=9

$$C_{\overline{\Delta}_{(9)}} < 1 \implies \overline{\Delta}_{(9)} = \underline{0}$$

- Pour un ordinateur travaillant en arithmétique binaire tronquée avec 48 bits de mantisse pour n = 12

$$C_{\overline{\Delta}_{(12)}} < 1 \implies \overline{\Delta}_{(12)} = \underline{0} \ .$$

Ainsi la matrice de Hilbert apparaît-elle comme informatiquement singulière et donc non inversible, à partir de l'ordre 9 sur le 1er ordinateur, et à partir de l'ordre 12 sur le 2ème.

VI.2.Le rôle du zéro informatique dans l'implémentation des méthodes itératives

Un grand nombre de problèmes d'analyse numérique se résolvent à l'aide de méthodes itératives.

Du point de vue mathématique, ces méthodes consistent, à partir d'un élément x_0, choisi arbitrairement comme étant une approximation de la solution x_S cherchée qui peut être soit un scalaire, $x_S \in \mathbb{R}$, soit un vecteur $x_S \in \mathbb{R}^n$, soit une matrice $x_S \in \mathbb{R}^n \times \mathbb{R}^n$, à former une suite

$$x_0, x_1, \ldots, x_n \qquad x_n \to x_S \qquad n \to \infty \ .$$

Cette suite sous les conditions de convergence de la méthode tend vers la limite x_s cherchée lorsque n tend vers l'infini.

Du point de vue informatique, il faut que la suite :

$X_0, X_1, \ldots, X_n$ ($X_i \in \mathbb{F}$ ou $X_i \in \mathbb{F}^n$ ou $X_i \in \mathbb{F}^n \times \mathbb{F}^n$)

soit arrêtée au bout d'un nombre fini de calculs. Des tests d'arrêt ont été élaborés tel par exemple :

si $||F(X_n)|| < t$ alors stop.

F étant la fonction qui doit être nulle à la solution du problème,
t étant un réel fixé a priori.

Ce type de test est mauvais car : si la valeur de t est choisie trop grande le processus itératif est arrêté trop tôt sur un X_n qui est encore améliorable, si elle est choisie trop petite un nombre important d'itérations inutiles sont faites sans pour autant améliorer la précision effective de l'approximation de la solution à cause de la propagation des erreurs d'arrondi.

Le seul test d'arrêt valable appelé test d'arrêt optimal est celui qui permet d'arrêter le processus itératif dès qu'une solution informatiquement satisfaisante est atteinte, c'est-à-dire dès que : $||F(X_n)|| = \underline{0}$.

Pour illustrer l'intérêt du zéro informatique en tant qu'outil permettant d'une part d'interrompre le processus itératif sur la bonne itération, et d'autre part de contrôler la validité de la solution fournie par l'ordinateur, traitons le problème suivant :

- Soit le système non-linéaire d'ordre 2 suivant :

$$\begin{cases} F_1(X_1,X_2) = \alpha(7X_1^2 + 3X_1X_2 + 4X_1 - X_2 - 41) = 0 \\ F_2(X_1,X_2) = \alpha(10X_1^2 + 4X_1X_2 + 5X_1 - X_2 - 56) = 0 \end{cases}$$

quelle que soit la valeur $\alpha \in \mathbb{R}$, la solution est $X_1^*=2$, $X_2^*=1$. Résolvons ce système en cherchant le minimum nul de la fonction :

$$\Phi(X_1,X_2) = f_1^2(X_1,X_2) + f_2^2(X_1,X_2) .$$

Pour cela nous utilisons un logiciel d'optimisation dans lequel le test d'arrêt optimal a été implémenté et défini par :

- si $||\text{grad } \Phi|| = \underline{0}$ alors stop, sinon continuer.

Remarquons que cette fonction $\Phi(x_1,x_2)$ admet 2 minima :

- l'un relatif ne correspondant pas à la solution du système linéaire et tel que $\Phi > 0$
- l'autre absolu tel que $\Phi = 0$.

Le logiciel utilisé peut bien évidemment fournir l'un ou l'autre des minima compte tenu du test d'arrêt choisi.

Les résultats obtenus sont les suivants :

1) $\alpha=10^{-20}$ et $X_{1,o} = -5$, $X_{2,o} = -2$

$X_{1S} = -2.0253...$ $X_{2S} = -2.6155...$ $\Phi = 5.25954...10^{-40}$

$C_\Phi = 6 \Longrightarrow \Phi > \underline{0}$

D'où X_{1S}, X_{2S} n'est pas solution du système non-linéaire

2) $\beta=10^{+30}$ et $X_{1,o}=-1$ $x_{2,o}=50$

$X_{1S}=2.00000...$ $X_{2S}=1.00000...$ $\overline{\Phi}=6.6....10^{+36}$

$C_{\vec{\Phi}} < 1 \Longrightarrow \overline{\Phi} = \underline{0}$.

D'où X_{1S}, X_{2S} est solution du système non-linéaire.

Nous pouvons remarquer que :

- dans le premier cas malgré la très faible valeur intrinsèque apparente de $\overline{\Phi}$ cette valeur est significative et connue avec 6 chiffres décimaux exacts, ce n'est pas un zéro informatique;

- dans le deuxième cas malgré la grande valeur intrinsèque apparente de $\overline{\Phi}$, cette valeur est non significative, c'est un zéro informatique.

Le zéro informatique permet de contrôler la validité des solutions fournies par l'ordinateur dans les algorithmes quelconques et de stabiliser les algorithmes itératifs, d'optimiser les algorithmes de types différences ou éléments finis. Il est d'un intérêt capital en informatique numérique.

VII - CONCLUSION

Le zéro mathématique, d'abord en tant que caractère désignant l'absence d'unité d'un certain rang dans la numération de position, a démistifié le calcul numérique, et a rendu l'arithmétique accessible à tous. Puis le zéro, valeur nulle, a permis le développement de l'algèbre et de l'algorithmique.

Avec les ordinateurs qui utilisent une représentation approchée des valeurs numériques et une arithmétique qui n'est qu'une approximation de l'arithmétique exacte, il est nécessaire d'introduire un nouveau concept, celui du zéro informatique. Comme

nous l'avons montré, ce concept transcende la notion de valeur nulle pour atteindre celle de valeur non significative, c'est-à-dire de valeur entachée d'une erreur d'arrondi du même ordre de grandeur que la valeur numérique elle-même. Ainsi en informatique une valeur numérique résultant d'un calcul sur ordinateur, doit être considérée comme un zéro informatique si son nombre de chiffres décimaux significatifs associés est inférieur à 1, et cela quelle que soit sa valeur intrinsèque.

Ce concept, par les possibilités qu'il offre de stabiliser , d'optimiser les algorithmes et de contrôler la validité des solutions qu'il fournit, me paraît être un apport décisif en informatique numérique et être la clef de voûte du développement, du contrôle et de la validité du calcul scientifique sur ordinateurs.

BIBLIOGRAPHIE

[1] R. Alt, Etude Stochastique de l'erreur numérique d'affectation sur ordinateur : Rapport Institut de Programmation n°76, février 1976.

[2] J.P. Faye and J. Vignes, Stochastic Approach of the Permutation-Perturbation Method for Round-Off Error Analysis. Applied Numerical Mathematics, Vol.1, n°4 (1985).

[3] G. Guitel, Histoire Comparée des numérations écrites, Flammarion, Paris (1975).

[4] G. Ifrah, Histoire Universelle des Chiffres, Seghers, Paris (1981).

[5] M. La Porte et J. Vignes, Etude Stochastique des erreurs dans l'arithmétique des ordinateurs. Applications au contrôle des résultats des algorithmes numériques. Numerische Mathematik 23, (1974).

[6] M. Maillé, Some Methods to estimate accuracy of measurements or Numerical Computation. Processings of Maths. for Comp. Congrès Paris (1982).

[7] D.E. Smith, History of Mathematics Dover Publ. (1958).

[8] Ton That Long, About Stochastic of Round-Off Error in Floating-Point Arithmétic. Applied Numerical Mathematics (à paraître).

[9] J. Vignes and M. La Porte, Error Analysis in Computing. Proceedings of IFIP Congress Stockholm (1974).

[10] J. Vignes, New Methods for Evaluating the Validity of the Results of Mathematical Computation. Mathematics and Computer in Simulation, vol.20, n°4 (1978).