Analyse Numérique

Multiplications vectorielle et matricielle - Complexité

Laurence Lamoulie

EISTI

Ecole Internationale des Sciences du Traitement de l'Information

Analyse active de l'erreur

Objectif

Produire en même temps que le résultat d'un calcul, une estimation (borne supérieure) de l'erreur dont il est entaché.

Lemme

La relation entre un nombre réel x et le nombre machine $m(x)$ qui le représente peut se formuler de la façon suivante :

$$m(x) = \frac{x}{1+\eta'} \quad |\eta'| \leq eps \qquad (*)$$

où *eps* représente l'epsilon machine.

Calcul de la somme des n premiers termes d'une suite $(x_k)_{k\in\mathbb{N}}$

Soit s_k la somme partielle des k premiers termes de la suite :

$$s_k = \sum_{i=1}^{n} x_i$$

Théoriquement, il suffit de faire le calcul selon les itérations :

$$\left\{ \begin{array}{c} s_0 = x_0 \\ s_k = s_{k-1} + x_k \quad k \geq 1 \end{array} \right. \tag{1}$$

Numériquement, le calcul effectif obtenu donne :

$$\left\{ \begin{array}{c} s_0 = m(x_0) \\ s_k = m(s_{k-1}) + m(x_k) \quad k \geq 1 \end{array} \right. \tag{2}$$

Calcul de la somme des n premiers termes d'une suite $(x_k)_{k\in\mathbb{N}}$

Le résultat du lemme donne pour un ε_k vérifiant $|\varepsilon_k| \leq eps$

$$\left\{\begin{array}{c} s_0 = m(x_0) \\ s_k = m(s_{k-1}) + m(x_k) = m(s_k)(1+\varepsilon_k) \quad k \geq 1 \end{array}\right. \tag{3}$$

Au rang k soit e_k l'erreur commise dans le calcul de s_k :

$$e_k = m(s_k) - s_k \tag{4}$$

$$\begin{array}{l} \Rightarrow s_k = m(s_k) + m(s_k)\varepsilon_k = e_k + s_k + m(s_k)\varepsilon_k \\ \text{or } s_k = m(s_{k-1}) + m(x_k) \\ \text{donc } s_k = e_k + s_k + m(s_k)\varepsilon_k = m(s_{k-1}) + m(x_k) \end{array} \tag{5}$$

Calcul de la somme des n premiers termes d'une suite $(x_k)_{k\in\mathbb{N}}$

Compte tenu de (*) on peut écrire pour un η_k vérifiant $|\eta_k| \leq eps$

$$s_k = e_k + s_k + m(s_k)\varepsilon_k = m(s_{k-1}) + x_k + \eta_k x_k \qquad (6)$$

et en reprenant (4) à l'ordre $k-1$ on obtient :

$$s_k = e_k + s_k + m(s_k)\varepsilon_k = e_{k-1} + s_{k-1} + x_k + \eta_k x_k \qquad (7)$$

Une simplification du fait de (1) donne

$$e_k + m(s_k)\varepsilon_k = e_{k-1} + \eta_k x_k$$

soit

$$e_k = e_{k-1} + \eta_k x_k - \varepsilon_k m(s_k) \qquad (8)$$

Calcul de la somme des n premiers termes d'une suite $(x_k)_{k\in\mathbb{N}}$

Les majorations de η_k et ε_k conduisent à

$$|e_k| \leq |e_{k-1}| + eps\left[|x_k| + |m(s_k)|\right] \tag{9}$$

On introduit maintenant la suite (δ_k) définie par

$$\left\{\begin{array}{c} \delta_0 = 0 \\ \delta_k = \delta_{k-1} + |x_k| + |m(s_k)| \quad k \geq 1 \end{array}\right. \tag{10}$$

En remarquant que (4) donne $e_0 = m(m(x_0)) - m(x_0) = 0$ et en utilisant (9) on relie les suites (e_k) et (δ_k) par :

$$\left\{\begin{array}{c} \delta_0 = e_0 = 0 \\ |e_k| - |e_{k-1}| \leq eps\left[\delta_k - \delta_{k-1}\right] \quad k \geq 1 \end{array}\right. \tag{11}$$

Calcul de la somme des n premiers termes d'une suite $(x_k)_{k\in\mathbb{N}}$

En sommant deux à deux ces majorations on a

$$\begin{cases} \delta_0 = e_0 = 0 \\ |e_k| \leq eps.\delta_k \quad k \geq 1 \end{cases} \tag{12}$$

Algorithme : calcul et erreur sur $s = \sum_{i=1}^{n} x_i$:

$s_0 \leftarrow x_0$
$\delta_0 \leftarrow 0$
Pour $k = 1$ à n
$\quad s_k = s_{k-1} + x_k$
$\quad \delta_k = \delta_{k-1} + |x_k| + |s_k|$
FinPour
$e_n = eps.\delta_n$ // e_n est un majorant de l'erreur et non la valeur de l'erreur comme définie en (4).

Erreur sur le produit scalaire de deux vecteurs

Le produit scalaire de deux vecteurs X et Y de $\mathbb{R}^n$ vaut

$$X^T Y = \sum_{i=1}^{n} X_i Y_i$$

Question qui se pose

Sous quelle(s) condition(s) le calcul du produit scalaire est il stable ?

Autrement dit

Question qui se pose

Peut-on borner l'erreur absolue commise, par un facteur qui dépend continuement de X et Y ?

Erreur sur le produit scalaire de deux vecteurs

Le produit scalaire de deux vecteurs X et Y de $\mathbb{R}^n$ vaut

$$X^T Y = \sum_{i=1}^{n} X_i Y_i$$

Il peut être calculé itérativement à partir des sommes partielles $s_k = \sum_{i=1}^{k} X_i Y_i$. A chaque étape on introduit une erreur, les erreurs s'accumulent au fur et à mesure du calcul, et pour tout η_k vérifiant $|\eta_k| \leq eps$, on a :

$$\begin{aligned}
m(s_1) &= m(x_1 y_1) = x_1 y_1 (1 + \eta_1) \\
m(s_2) &= m(m(s_1) + m(x_2 y_2)) = (m(s_1) + x_2 y_2 (1 + \eta_2))(1 + \eta_3) \\
&= x_1 y_1 (1 + \eta_1)(1 + \eta_3) + x_2 y_2 (1 + \eta_2)(1 + \eta_3)
\end{aligned}$$

Erreur sur le produit scalaire de deux vecteurs

Si on suppose que $1 + \eta_k \simeq 1 \pm \eta$, on généralise en

$$m(s_n) = x_1 y_1 (1 \pm \eta)^n + x_2 y_2 (1 \pm \eta)^n + x_3 y_3 (1 \pm \eta)^{n-1} + \ldots + x_n y_n (1 \pm \eta)^2$$

et on montre que cela revient à

$$m(s_n) = x_1 y_1 (1 \pm \theta_n) + x_2 y_2 (1 \pm \theta'_n) + x_3 y_3 (1 \pm \theta_{n-1}) + \ldots + x_n y_n (1 \pm \theta_2)$$

On peut considérer que l'on a effectué

$$X^T (Y + \Delta Y) = \sum_{i=1}^{n} X_i (Y_i + \Delta Y_i) \qquad \text{avec } \|\Delta Y\| \leq \gamma_n \|Y\|$$

ou $$(X + \Delta X)^T Y = \sum_{i=1}^{n} (X_i + \Delta X_i) Y_i \qquad \text{avec } \|\Delta X\| \leq \gamma_n \|X\|$$

Erreur sur le produit scalaire de deux vecteurs

On a donc

$$m(X^TY) = X^T(Y + \Delta Y) = (X + \Delta X)^T Y$$

avec $\|\Delta X\| \leq \gamma_n \|X\|$ et $\|\Delta Y\| \leq \gamma_n \|Y\|$ et $\gamma_n = \frac{n.eps}{1-n.eps}$
d'où

$$\left|X^TY - m(X^TY)\right| = \left|X^T \Delta Y\right| = \left|(\Delta X)^T Y\right|$$

or $\left|X^T\Delta Y\right| = \left|(\Delta X)^T Y\right| \leq \left\|X^T\right\| . \|\Delta Y\| \leq \|X\| .\gamma_n \|Y\|$

$$\boxed{\Rightarrow \left|X^TY - m(X^TY)\right| \leq \gamma_n \|X\| \|Y\|}$$

Conclusion : Si $n.eps < 1$ alors le calcul est stable, sinon ?

Erreur sur les produits extérieur et matriciel

Produit extérieur

Si on note $A = X.Y^T$ le produit extérieur, on montre que

$$m(A) = X.Y^T + \Delta \qquad \text{avec} \ \|\Delta\| \leq eps. \left\|X.Y^T\right\|$$

Produit matriciel

Si on note $C = A.B$ le produit des matrices A et B, on montre que

$$\|C - m(C)\| \leq \gamma_n \|A\| . \|B\|$$

Complexité des algorithmes numériques (voir Brezinski pour détails)

Contexte

Etude théorique et pratique des algorithmes qui demandent le moins possible d'opérations arithmétiques pour effectuer un certain calcul.

Objectif théorique

Démonstration de l'existence d'une borne inférieure pour le nombre d'opérations arithmétiques nécessaires à la réalisation d'un calcul, puis recherche de cette borne.

Objectif pratique

Trouver un algorithme dont le nombre d'opérations arithmétiques de rapproche le plus possible de cette borne.

Complexité

Définition

On appelle **complexité** d'un algorithme numérique, le nombre d'opérations arithmétiques que son exécution nécessite.

Définition

Soit un algorithme dont le nombre d'opérations dépend de la variable n. On dit que la complexité $C(n)$ de l'algorithme est de l'ordre $f(n)$ s'il existe deux constantes N et α telles que :

$$C(n) \leq \alpha f(n), \qquad \forall n \geq N$$

Notation et Exemple

On note $C(n) = O(f(n))$.
Le produit scalaire de deux vecteurs de $\mathbb{R}^n$ est une opération en $O(n)$.

Complexité : exemple du produit matriciel

Le produit matriciel C de $A \in \mathbb{R}^{m \times n}$ et $B \in \mathbb{R}^{n \times p}$ s'effectue habituellement par

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}, \qquad i = 1..m, j = 1, ..p$$

Chaque valeur c_{ij} nécessite :

- n multiplications
- $n - 1$ additions

soit au total

- nmp multiplications
- $(n-1)mp$ additions

En fait, la complexité du produit est comprise entre $K_1 n^2$ et $K_2 n^{2.496}$.

D'autres algo. pour le calcul du produit matriciel

- S. Winograd (1970), basé sur l'idée :

  $$x_1y_1 + x_2y_2 = (x_1 + y_2)(x_2 + y_1) - x_1x_2 - y_1y_2$$

  et son extension au calcul du p.s. de deux vecteurs de dimension $n = 2k$.
  Gain : $/2$ le nombre de multiplications, $\nearrow$ additions
- V.Strassen (1969), basé sur décomposition par blocs de A et B, et une résolution récursive.
  Gain : Nombre d'opérations : $4.7n^{2.807}$

Références

- *Analyse numérique*, M. Schatzman, InterEditions, 1991,
- *Analyse numérique matricielle appliquée à l'art de l'ingénieur*, P. Lascaux R. Theodor, Masson, 1993.
- *Algèbre matricielle numérique*, C. Brezinski, document pdf, Chapitre 2