Analyse Numérique

Analyse des erreurs, Suite

Laurence Lamoulie

EISTI

Ecole Internationale des Sciences du Traitement de l'Information

Erreurs de calcul

Fait 1.5 : Pour une opération quelconque

Soient a et b deux réels, le résultat de $a \otimes b$ est en fait :

$$m(a \otimes b) = m\left(m(a) \otimes m(b)\right)$$

où l'erreur relative $\frac{m(a\otimes b) - a\otimes b}{a\otimes b}$ se décompose en :

- erreur d'entrée $\eta^I(a \otimes b)$ dûe à l'erreur de représentation du résultat de l'opération
- erreur de calcul $\eta^C(a \otimes b)$ dûe au calcul lui-même, inhérente à la machine

L'erreur d'entrée $\eta^I(a \otimes b)$ dépend de l'ordre des opérations donc on peut la minimiser.

Erreurs de calcul

La suite...

On envisage dans la suite les différentes opérations les plus classiques :

- Addition
- Multiplication
- Division

Vous trouverez dans le cours le résultat pour la racine carrée.

Discussion pour la somme

Soient a et b deux réels dont on calcule la somme $S = a + b$.
On a :

$$S = a + b \Rightarrow \Delta^I S = \Delta^I a + \Delta^I b$$

Or $\eta(x) = \frac{\Delta(x)}{x}$ donc

$$\eta^I(S) = \frac{\Delta^I S}{S} = \frac{\Delta^I a}{S} + \frac{\Delta^I b}{S}$$

et $\Delta(x) = \eta(x) * x$ donc

$$\Rightarrow \eta^I(S) = \frac{a}{S}\eta^I a + \frac{b}{S}\eta^I b$$

Théorèmes pour la somme

Pour la somme de 2 réels

Théorème : Soient a et b deux réels, la somme $S = a + b$ est entachée :

- d'une erreur absolue d'entrée :

$$\Delta^I S = \Delta^I a + \Delta^I b$$

- donc d'une erreur relative d'entrée ou erreur de précision d'entrée

$$\eta^I(S) = \frac{a}{S}\eta^I(a) + \frac{b}{S}\eta^I(b)$$

Théorèmes pour la somme

Généralisation pour la somme de n réels

Théorème 1.6.1 : Généralisation
Soient $x_1, x_2, ...x_n$ n réels, la somme $S = \sum_{i=1}^{n} x_i$ est entachée :

- d'une erreur absolue d'entrée :

$$\Delta^I S = \sum_{i=1}^{n} \Delta^I x_i \qquad (1.6.3)$$

- donc d'une erreur relative d'entrée ou erreur de précision d'entrée

$$\eta^I(S) = \sum_{i=1}^{n} \frac{x_i}{S} \eta^I(x_i) \qquad (1.6.4)$$

Majoration d'erreur pour la somme

En reprenant la somme de deux réels, on peut majorer dans $\eta^I(S) = \frac{a}{S}\eta^I a + \frac{b}{S}\eta^I b$ en prenant la valeur absolue :

$$\left|\eta^I(S)\right| \leq \frac{|a|}{|S|}\left|\eta^I a\right| + \frac{|b|}{|S|}\left|\eta^I b\right|$$

donc, sachant que $\left|\eta^I x\right| \leq \varepsilon$

$$\left|\eta^I(S)\right| \leq \frac{|a|}{|S|}\varepsilon + \frac{|b|}{|S|}\varepsilon = \frac{|a| + |b|}{|S|}\varepsilon$$

Cette majoration se généralise :

$$\left|\eta^I(S)\right| \leq \frac{\sum_{i=1}^{n} |x_i|}{\left|\sum_{i=1}^{n} x_i\right|}\varepsilon \qquad (1.6.5)$$

Discussion pour le produit

Soient a et b deux réels dont on calcule le produit $P = a * b$.
On a :

$$P = a * b \Rightarrow P + \Delta^l P = m(a) * m(b) = (a + \Delta^l a) * (b + \Delta^l b)$$

Soit, en négligeant les termes d'ordre 2 :

$$P + \Delta^l P = ab + \Delta^l a.b + \Delta^l b.a$$

comme $P = ab$ on a

$$\Delta^l P = \frac{\Delta^l a}{a}.ab + \frac{\Delta^l b}{b}.ab$$

$$\Rightarrow \frac{\Delta^l P}{P} = \frac{\Delta^l a}{a} + \frac{\Delta^l b}{b}$$

$$\Rightarrow \left|\frac{\Delta^l P}{P}\right| \leq \left|\frac{\Delta^l a}{a}\right| + \left|\frac{\Delta^l b}{b}\right| \leq 2\varepsilon$$

Théorèmes pour le produit

Pour le produit de 2 réels

Théorème : Soient a et b deux réels non nuls, le produit $P = a * b$ est entaché :

- d'une erreur absolue d'entrée :

$$\Delta^I P = P.\left(\frac{\Delta^I a}{a} + \frac{\Delta^I b}{b}\right)$$

- donc d'une erreur relative d'entrée ou erreur de précision d'entrée

$$\eta^I(P) = \eta^I(a) + \eta^I(b)$$

Théorèmes pour le produit

Généralisation pour le produit de n réels

Théorème 1.6.2 : Généralisation

Soient $x_1, x_2, ...x_n$ n réels non nuls, le produit $P = \prod_{i=1}^{n} x_i$ est entaché :

- d'une erreur absolue d'entrée :

$$\Delta^I P = P. \left(\sum_{i=1}^{n} \frac{\Delta^I x_i}{x_i} \right)$$

- donc d'une erreur relative d'entrée ou erreur de précision d'entrée

$$\eta^I(P) = \sum_{i=1}^{n} \eta^I(x_i)$$

Discussion pour la division

Soient a et b deux réels dont on calcule le quotient $Q = a/b$.
On a :

$$Q = \frac{a}{b} \Rightarrow Q + \Delta^I Q = \frac{m(a)}{m(b)} = \frac{a + \Delta^I a}{b + \Delta^I b}$$

$$\Rightarrow \left(Q + \Delta^I Q\right)\left(b + \Delta^I b\right) = \left(a + \Delta^I a\right)$$

Soit, en négligeant les termes d'ordre 2 :

$$Q\Delta^I b + b\Delta^I Q = \Delta^I a$$

$$\Rightarrow \Delta^I Q = \frac{b\Delta^I a - a\Delta^I b}{b^2}$$

$$\Rightarrow \left|\frac{\Delta^I Q}{Q}\right| = \left|\eta^I(Q)\right| \leq 2\varepsilon$$

Théorème pour la division

Pour la division de 2 réels

Théorème 1.6.3 : Soient a et b deux réels, b non nul, le produit $Q = a/b$ est entaché :

- d'une erreur absolue d'entrée :

$$\Delta^I Q = \frac{b\Delta^I a - a\Delta^I b}{b^2} \qquad (1.6.9)$$

- donc d'une erreur relative d'entrée ou erreur de précision d'entrée

$$\eta^I(Q) = \eta^I(a) - \eta^I(b) \qquad (1.6.10)$$

Formalisation de la notion d'algorithme

Un algorithme ?

Définition : C'est une transformation ϕ qui fait passer d'un ensemble de données $X = [x_1, x_2, ..., x_n]$ à un résultat final $Y = [y_1, y_2, ..., y_m] = \phi(X)$.

On décompose ϕ en une suite de transformations élémentaires $\phi^{(k)}, k = 1...r$:

$$\phi = \phi^{(r)} o ... o \phi^{(2)} o \phi^{(1)}$$

Si en entrée on donne $m(X)$ au lieu de X, alors à la sortie on a $m(Y) = m(\phi(m(X)))$. L'erreur $\Delta(Y)$ s'exprime pour chaque composante $y_i = \phi_i(X)$ de Y :

$$\Delta y_i \approx \sum_{j=1}^{n} \frac{\partial \phi_i(X)}{\partial x_j} \Delta x_j \quad i = 1, ...m$$

Formalisation de la notion d'algorithme

Ecriture sous forme matricielle

$$\Delta y_i \approx \sum_{j=1}^{n} \frac{\partial \phi_i(X)}{\partial x_j} \Delta x_j \quad i = 1, ...m \quad (1.7.1)$$

donne

$$\Delta Y = \begin{bmatrix} \Delta y_1 \\ \vdots \\ \Delta y_m \end{bmatrix} \approx \begin{bmatrix} \frac{\partial \phi_1}{\partial x_1} & \cdots & \frac{\partial \phi_1}{\partial xn} \\ \vdots & \cdots & \vdots \\ \frac{\partial \phi_m}{\partial x_1} & \cdots & \frac{\partial \phi_m}{\partial x_n} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{bmatrix}$$

$$= J\left[\phi(X)\right] \Delta X \qquad (1.7.2)$$

où $J\left[\phi(X)\right]$ est la matrice jacobienne de ϕ

Erreur relative dans un algorithme

De (1.7.1) on déduit que

$$\frac{\Delta y_i}{y_i} = \sum_{j=1}^{n} \frac{\partial \phi_i(X)}{\partial x_j} \frac{\Delta x_j}{x_j} \frac{x_j}{y_i}$$

donc

$$\eta(y_i) = \sum_{j=1}^{n} \frac{x_j}{y_i} \frac{\partial \phi_i}{\partial x_j} \eta(x_j)$$

soit

$$\eta(y_i) = \sum_{j=1}^{n} \frac{x_j}{\phi_i(X)} \frac{\partial \phi_i}{\partial x_j} \eta(x_j) \quad (1.7.3)$$

Les facteurs $\frac{x_j}{\phi_i(X)} \frac{\partial \phi_i}{\partial x_j}$ sont appelés **facteurs de conditionnement** de l'algorithme

Exemple : Calcul de $a - b^2$

$$x^{(0)} = \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} \overset{\phi^{(1)}}{\longrightarrow} x^{(1)} = \begin{pmatrix} a \\ b^2 \end{pmatrix} \overset{\phi^{(2)}}{\longrightarrow} x^{(2)} = a - b^2 = Y$$

L'algorithme compte 2 étapes donc $r = 2$. Si on note $\Psi^{(k)}$ l'application qui reste à faire après k étapes, on a ici seulement $\Psi^{(1)}$:

$$\Psi^{(1)} = \phi^{(2)} = \text{reste à faire après } \phi^{(1)}$$

soit

$$\Psi^{(1)} = \begin{pmatrix} a \\ \alpha \end{pmatrix} = a - \alpha$$

Exemple : Calcul de $a - b^2$

On décompose ϕ sous la forme

$$\phi = \phi^{(2)} o \phi^{(1)}$$

soit ici

$$\phi = \Psi^{(1)} o \phi^{(1)}$$

mais en général on a

$$\phi = \Psi^{(k)} o \phi^{(k)} o \phi^{(k-1)} ... o \phi^{(1)}$$

où

$$\Psi^{(k)} = \phi^{(r)} o \phi^{(r-1)} ... o \phi^{(k+1)}$$

Exemple : Calcul de $a - b^2$

On veut $\Delta x^{(2)}$:

$$
\begin{aligned}
\Delta x^{(2)} &= m(x^{(2)}) - x^{(2)} \\
&= m(\phi^{(2)}(m(x^{(1)}))) - \phi^{(2)}(x^{(1)}) \\
&= \underbrace{m(\phi^{(2)}(m(x^{(1)}))) - \phi^{(2)}(m(x^{(1)}))}_{\alpha} \\
&\quad + \underbrace{\phi^{(2)}(m(x^{(1)})) - \phi^{(2)}(x^{(1)})}_{\beta}
\end{aligned}
$$

On développe chaque terme séparément

Exemple : Calcul de $a - b^2$

Pour β :

$$\beta \approx J\phi^{(2)}(x^{(1)}).(m(x^{(1)}) - x^{(1)})$$

(analogue à $f(x+h) - f(x) = f'(x).h$) soit

$$\beta \approx J\phi^{(2)}(x^{(1)}).\Delta(x^{(1)})$$

Pour α

$$m[\phi^{(2)}(m(x^{(1)}))] = \phi^{(2)}(m(x^{(1)}))\left(1 + \eta^2(m(x^{(1)}))\right)$$

car $m(X) = X + m(X) - X = X(1 + \frac{m(X)-X}{X})$ et
$\eta = \frac{\Delta X}{X} = \frac{m(X)-X}{X}$

Exemple : Calcul de $a - b^2$

Pour α

$$\begin{aligned}
& \phi^{(2)}(m(x^{(1)}))\left(1+\eta^2(m(x^{(1)}))\right) - \phi^{(2)}(m(x^{(1)})) \\
= & \ \eta^2(m(x^{(1)})).\phi^{(2)}(m(x^{(1)})) \\
\approx & \ \eta^2(m(x^{(1)})).\phi^{(2)}(x^{(1)}) \text{ car on néglige les termes d'ordre 2} \\
= & \ \eta^2(m(x^{(1)})).x^{(2)}
\end{aligned}$$

donc

$$\begin{aligned}
\Delta x^{(2)} &= \eta^{(2)}(m(x^{(1)})).x^{(2)} + J\phi^{(2)}(x^{(1)}).\Delta(x^{(1)}) \\
&\approx \eta(a-B).x^{(2)} + J\Psi^{(1)}(x^{(1)}).\Delta(x^{(1)})
\end{aligned}$$

où $H_r y = \eta(a-B).x^{(2)}$
Sur $\Delta(x^{(1)})$ on renouvelle le même raisonnement

Exemple : Calcul de $a - b^2$

Pour $\Delta(x^{(1)})$:

$$
\begin{aligned}
\Delta x^{(1)} &= \eta^{(1)}(m(x^{(0)})).x^{(1)} + J\phi^{(1)}(x^{(0)}).\Delta(x^{(0)}) \\
&= \begin{pmatrix} 0 & 0 \\ 0 & \eta(b^2) \end{pmatrix} \begin{pmatrix} a \\ b^2 \end{pmatrix} + \begin{pmatrix} \frac{\partial \phi_1^{(1)}}{\partial a} & \frac{\partial \phi_1^{(1)}}{\partial b} \\ \frac{\partial \phi_2^{(1)}}{\partial a} & \frac{\partial \phi_2^{(1)}}{\partial b} \end{pmatrix} \begin{pmatrix} \Delta a \\ \Delta b \end{pmatrix} \\
&= \begin{pmatrix} 0 \\ \eta(b^2).b^2 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 2b \end{pmatrix} \begin{pmatrix} \Delta a \\ \Delta b \end{pmatrix} \\
&= \begin{pmatrix} 0 \\ \eta(b^2).b^2 \end{pmatrix} + \begin{pmatrix} \Delta a \\ 2b\Delta b \end{pmatrix} \\
&= \begin{pmatrix} \Delta a \\ \eta(b^2).b^2 + 2b\Delta b \end{pmatrix}
\end{aligned}
$$

Exemple : Calcul de $a - b^2$

Finalement :

$$\begin{aligned}
\Delta x^{(2)} &= \eta(a-B).(a-b^2) + J\Psi^{(1)}(x^{(1)}).\begin{pmatrix} \Delta a \\ \eta(b^2).b^2 + 2b\Delta b \end{pmatrix} \\
&= \eta(a-B).(a-b^2) + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} \Delta a \\ \eta(b^2).b^2 + 2b\Delta b \end{pmatrix} \\
&= \underbrace{\eta(a-b^2).(a-b^2) - \eta(b^2).b^2}_{\substack{\text{propagation des erreurs sur les} \\ \text{op. faites en cours d'algo.}}} \\
&+ \underbrace{\Delta a - 2b\Delta b}_{\text{erreur sur la donnée}}
\end{aligned}$$

Généralités

Dans l'erreur globale, on a mis en évidence deux contributions :

- L'erreur sur la donnée répercutée sur le résultat par l'intermédiaire de ϕ
- Les effets des erreurs d'arrondi sur le résultat final que l'on note $E_r(A, X)$ (dont on trouvera l'expression en (1.8.20))

Seule la deuxième varie d'un algorithme à l'autre, c'est pourquoi on adopte la définition suivante :

FAIT1.7 : Définition

Un algorithme A est plus **crédible** qu'un algorithme A' si $E_r(A, X) \leq E_r(A', X)$ pour un même ensemble de données X

Généralités

Dans la contribution $E_r(A, X)$ le facteur $H_r Y$ étant majoré, l'erreur d'entrée $\Delta^I Y$ est :

$$\Delta^I Y = [|J\phi(X)|\,|X| + |Y|]\,\varepsilon$$

On adopte alors la Définition suivante :

Stabilité numérique

Un algorithme est dit **numériquement stable** si les erreurs intermédiaires d'arrondi sont de même ordre de grandeur que l'erreur inhérente $\Delta^I Y$

Références