Επιστημονικός Υπολογισμός

Ε.Γαλλόπουλος

ΤΜΗΥΠ, Π. Πατρών

Διάλεξη 7: 15 Νοεμβρίου 2017

Περιεχόμενα I

Περιεχόμενα II

Μελέτη σφαλμάτων: Το ((σκηνικό))

Ας θεωρήσουμε ότι το πρόβλημα αντιστοιχεί στον υπολογισμό της απεικόνισης $f : \mathcal{U} \subset \mathbb{R}^m \to \mathbb{R}^n$ για ορισμένα δεδομένα.

Προσοχή Αν τρέξουμε τον αλγόριθμο διακρίνουμε:

$x \in \mathcal{U}$ τα m στοιχεία στο πεδίο ορισμού της f;

$f(x)$ τα n στοιχεία της τιμής της συνάρτησης στο x, χωρίς λάθη υπολογισμών;

x^* τα στοιχεία εισόδου που χρησιμοποιούνται στην υλοποίηση, οπότε $x^* \in F$. Αν τα μόνα λάθη που υπάρχουν sto x^* είναι αυτά που προέρχονται από τη στρογγύλευση του x, τότε $x^* = \mathrm{fl}(x)$.

$f(x^*)$ η τιμή της $f(x^*)$, υπολογισμένη χωρίς σφάλματα (δηλ. με την αριθμητική του $\mathbb{R}$.

$f_{\mathrm{prog}}(x^*)$ η υλοποίηση του υπολογισμού της $f(x^*)$ με πεπερασμένο πλήθος διακριτών πράξεων α.κ.υ. στο $\mathcal{F}$.

Στόχος

Να εκτιμήσουμε το απόλυτο ή σχετικό σφάλμα

$$\|f_{\text{prog}}(x^*) - f(x)\| \text{ ή αν } f(x) \neq 0 \quad \frac{\|f_{\text{prog}}(x^*) - f(x)\|}{\|f(x)\|}.$$

Αν το σφάλμα είναι μικρό (π.χ. της τάξης του ϵ της μηχανής) ο αλγόριθμος, f, θεωρείται ακριβής.

Εμπρός Σφάλμα (forward error).

Το παραπάνω αποκαλείται (απόλυτο ή σχετικό) εμπρός[α΄] sf'alma.

[α΄] Ο χαρακτηρισμός ((εμπρός)) θα εξηγηθεί στη συνέχεια των διαλέξεων.

ΠΡΟΣΟΧΗ Γενικά είναι ανέφικτο να υπολογιστεί. Στόχος είναι η εκτίμησή του, π.χ. βρίσκοντας σφικτά φράγματα.

Παρατηρήσεις

- Το $\|\cdot\|$ συμβολίζει κάποια μετρική - π.χ.
- ... απόλυτη τιμή, στους βαθμωτούς,
- ... μία από τις γνωστές νόρμες, για διανύσματα και μητρώα
- ... πίνακας απολύτων τιμών (*απόσταση κατά συνιστώσες*).
- Οι έννοιες του απόλυτου και σχετικού σφάλματος είναι γενικές και ξεπερνούν την α.κ.υ.
- Αν το x^* χρησιμοποιείται ως προσέγγιση του x, τότε

$$\text{ΑΣ: } \|x^* - x\| \text{ ΣΣ αν } x \neq 0 \quad \frac{\|x^* - x\|}{\|x\|}$$

- Μερικές φορές θέτουμε ΣΣ το $\frac{\|x^* - x\|}{\|x^*\|}$
- Αν $x = 0$ τότε αυτό το ΣΣ είναι πάντα 1 όσο καλή και να είναι η προσέγγιση x^* !!! (... το άλλο ΣΣ δεν ορίζεται)
- Για προσεγγίσεις πολύ μικρών τιμών χρησιμοποιείται το απόλυτο σφάλμα.
- Χονδρικά, αν $\frac{|x^* - x|}{|x|} \leq 5 \cdot 10^{-d}$ τότε λέμε ότι οι (αριθμοί) x^* και x συμφωνούν σε d δεκαδικά ψηφία.

Από το ϵ_M στο $\mathbf{u}$ – Μονάδα στρογγύλευσης

- Στη MATLAB για οποιοδήποτε α.κ.υ. `x`, η εντολή `eps(x)` επιστρέφει την απόστασή του από τον διαδοχικό του. Στη βιβλιογραφία αυτό αναφέρεται και ως ulp(x) (units in the last place).
- Αν χρησιμοποιούμε t bits για την αναπαράσταση της ουράς (το 1ο κρυμμένο) και $x = m \times \beta^e$ τότε η απόσταση του x από τον αμέσως επόμενο α.κ.υ. είναι
  $ulp(x) = (m + 2^{-(t-1)})2^e - m2^e = 2^{e+1-t}$.
- Η μέγιστη σχετική απόσταση είναι $\frac{2^{e+1-t}}{2^e} = 2^{1-t}$.
- ... προφανώς ίση με ϵ_M.

Θα εκφράζουμε τα σφάλματα ως πολλαπλάσια της μονάδας στρογγυλευσης που θα χρησιμέυσει ως μονάδα μέτρησης των σφαλμάτων.

Μονάδα στρογγύλευσης (unit roundoff)

Η μονάδα στρογγύλευσης είναι το μέγιστο δυνατό σχετικό σφάλμα *για τον επιλεγμένο τρόπο στρογγύλευσης*. Για *στρογγύλευση προς το πλησιέστερο*,
$\mathbf{u} = \max_{z \neq 0} \frac{|z - \mathrm{fl}(z)|}{|z|} = \frac{2^{1-t}}{2}$.

IEEE double: $\mathbf{u} = 1.1102e-016$; IEEE single: $\mathbf{u} = 5.9605e-008$.

Αρχή ακριβούς στρογγύλευσης (υπενθύμιση)

Αν $\tilde{\odot}$ είναι η υλοποίηση της αριθμητικής πράξης $\odot$, τότε αν $x, y \in F$ ισχύει ότι $x\tilde{\odot}y = \mathrm{fl}(x \odot y) \in F$.

Το υπολογισμένο αποτέλεσμα είναι ακριβώς ίδιο με το να εκτελούνταν η πράξη με ((θεϊκή)) αριθμητική και μετά να εφαρμοζόταν στρογγύλευση.

ΕΠΟΜΕΝΩΣ

$$|x\tilde{\odot}y - x \odot y| \quad = \quad |\mathrm{fl}(x \odot y) - x \odot y| \leq \mathbf{u}|x \odot y|$$

άρα

$$-\mathbf{u}(x \odot y) \leq \mathrm{fl}(x \odot y) - x \odot y \leq \mathbf{u}(x \odot y)$$

$$(1 - \mathbf{u})(x \odot y) \leq x\tilde{\odot}y \leq (1 + \mathbf{u})(x \odot y)$$

Μοντέλο διάδοσης σφαλμάτων στρογγύλευσης

Μοντέλο διάδοσης

Μετά από κάθε αριθμητική πράξη $\odot$ επί δεδομένων α.κ.υ. $x, y \in F$ και δεν υπάρχει υπερ- ή υποχείλιση, ισχύει:

$$\boxed{\mathrm{fl}(x \odot y) = (1+\delta)(x \odot y), \text{ για } \delta \text{ τ.ώ. } |\delta| \leq \mathbf{u}}$$

Σχετικά με το δ:

- φράσσεται σε απόλυτη τιμή! Επομένως $-\mathbf{u} \leq \delta \leq \mathbf{u}$ (με πιο ενδελεχή ανάλυση $|\delta| < \mathbf{u}$.)
- διαφέρει για κάθε πράξη και στοιχεία x, y
- δεν απαιτείται να το γνωρίζουμε ακριβώς - χρησιμοποιούμε το μοντέλο γνωρίζοντας μόνον πώς φράσσεται το δ!
- .... το μοντέλο δεν προβλέπει πότε $\delta = 0$ (αυτή η πληροφορία μπορεί να είναι χρήσιμη, αλλά χάνεται).

Ιδιότητες αριθμητικών πράξεων στο πεδίο των πραγματικών $\mathbb{R}$

Αξιώματα πρόσθεσης

- A0 Το άθροισμα $x + y \in \mathbb{R}$.
- A1 Η πρόσθεση είναι αντιμεταθετική: $x + y = y + x$.
- A2 Η πρόσθεση είναι προσεταιριστική $x + (y + z) = (x + y) + z$.
- A3 Υπάρχει στοιχείο 0 ώστε $x + 0 = x$ για κάθε $x \in \mathbb{R}$.
- A4 Για κάθε $x \in \mathbb{R}$ υπάρχει αντίστροφο στοιχείο $-x \in \mathbb{R}$ ως προς την πρόσθεση, δηλ. $x + (-x) = 0$.

Αξιώματα πολλαπλασιασμού

- Π0 Το γινόμενο $x \times y \in \mathbb{R}$.
- Π1 Αντιμεταθετική ιδιότητα πολλαπλασιασμού: $x \times y = y \times x$.
- Π2 Προσεταιριστική ιδιότητα πολλαπλασιασμού: $x \times (y \times z) = (x \times y) \times z$.
- Π3 Υπάρχει στοιχείο 1 ώστε $x \times 1 = x$ για κάθε $x \in \mathbb{R}$.
- Π4 Για κάθε μη μηδενικό x υπάρχει αντίστροφο ως προς τον πολλαπλασιασμό $\frac{1}{x} \in \mathbb{R}$ ώστε $x \times (\frac{1}{x}) = 1$.

Ε Επιμεριστική ιδιότητα: $x \times (y + z) = x \times y + x \times z$.

Παράδειγμα: Δεν ισχύει πάντα το Α2

$$t_1 = \mathrm{fl}(x+y) \qquad s_1 = \mathrm{fl}(y+z)$$
$$t_2 = \mathrm{fl}(t_1+z) \qquad s_2 = \mathrm{fl}(x+s_1)$$

και δεν υπάρχει λόγος να ισχύει πάντα (αν και πολλές φορές ισχύει!)

$$\mathrm{fl}(\mathrm{fl}(x+y)+z) = \mathrm{fl}(x+\mathrm{fl}(y+z)).$$

Παράδειγμα: Δεν ισχύει πάντα το Α2

Listing 1: Παράδειγμα

```
(1+eps/2)+eps/2 % = 1
1+(eps/2+eps/2) % = 1.000000000000000
(-10^20+10^20)+1 % = 1
-10^20+(10^20+1) % = 0
```

Listing 2: Παράδειγμα

```
a = 0.123456700000000;
b = 4.711325195312500e+008; c = -b;
(a+b)+c % = 0.123456716537476
a+(b+c) % = 0.123456700000000
```

Παράδειγμα: Δεν ισχύει πάντα το Π4

Γενικά για $x \in F$, $\mathrm{fl}(x \cdot \mathrm{fl}(\frac{1}{x})) \neq 1$.

Listing 3: Παράδειγμα

```
index = []; for i=1:170
    if ((1/i)*i ~= 1)
        index = [index i];
    end;
end;
index
    49    98   103   107   161
```

Προειδοποιήσεις: Προσοχή για τους συγγραφείς μεταφραστών!

Επειδή στο $\mathcal{F}$ δεν ισχύουν όλες οι ιδιότητες πεδίου, δεν μπορούμε να κάνουμε τις ίδιες απλοποιήσεις και μετατροπές των αριθμητικών εκφράσεων που επιτρέπονται στο $\mathbb{R}$.
Π.χ. αν με δύο προσθετές που υπολογίζουν ταυτόχρονα το άθροισμα των α.κ.υ. x_1 , x_2 και των x_3 , x_4 το άθροισμα $x_1 + x_2 + x_3 + x_4$ μπορεί να υπολογισθεί ταχύτερα από το κλασικό $\big((x_1 \tilde{+} x_2) \tilde{+} x_3\big) \tilde{+} x_4$ χησιμοποιώντας $\big(x_1 \tilde{+} x_2\big) \tilde{+} \big(x_3 \tilde{+} x_4\big)$.

Το σφάλμα και τα αποτελέσματα μπορεί να είναι διαφορετικά!

Προειδοποίηση Οι επεξεργαστές Intel εκτελούν τις πράξεις σε εκτεταμένη διπλή ακρίβεια (80 bits) ... Αυτό ορισμένες φορές οδηγεί σε απρόσμενα αποτελέσματα (double rounding).

ΠΗΓΗ ΠΡΟΒΛΗΜΑΤΩΝ ΣΤΙΣ ±ΣΕΙΣ: Καταστροφική απαλοιφή

Προσοχή Όταν αφαιρούνται δύο αριθμοί που είναι σχεδόν ίσοι και περιέχουν μικρά σφάλματα "αναδύονται σκουπίδια" ακόμα και αν χρησιμοποιούσαμε αριθμητική άπειρης ακρίβειας!
Αν $A = A_1 + \text{θόρυβος}$, $A' = A_2 + \text{θόρυβος}$ και οι τιμές A_1, A_2 είναι σχεδόν ίσες και θέλουμε να υπολογίσουμε το (μικρό) $A_1 - A_2$ αφαιρώντας $A - A'$, τότε:

$$\boxed{\big(A_1 + \text{θόρυβος}\big) - \big(A_2 + \text{θόρυβος}\big) = \big(A_1 - A_2\big) + \text{θόρυβος}}$$

- καταστροφική απαλοιφή όταν $\big|A_1 - A_2\big| = O\big(\text{θόρυβος}\big)$ ή λιγότερο.
- Η μόλυνση από τα σκουπίδια επηρεάζει και έχει καταστροφικά αποτελέσματα αν, για παράδειγμα, τα ((ασήμαντα σκουπίδια)) πολλαπλασιαστούν με μεγάλους αριθμούς και χρησιμοποιηθούν περαιτέρω.
- Διαβάστε την ιστορία Catastrophic cancellation in the high seas της A. Langville (Lan01).

Στρογγύλευση: Από $x \in \mathbb{R}$ στο $\mathrm{fl}(x) \in F$

Αν $x \in G$ τότε μπορούμε να γράψουμε $\mathrm{fl}(x) = x(1+\delta)$ για κάποιο $|\delta| \leq \mathbf{u}$.

Σφάλμα πράξεων

Έστω ότι $x, y \in F$ και $x \odot y \in G$ και ότι $\odot \in \{\pm, \times, , \sqrt{}\}$. Τότε

$$\mathrm{fl}(x \odot y) \quad = \quad (x \odot y)(1+\delta), \text{ για κάποιο } |\delta| \leq \mathbf{u}$$

και

$$\mathrm{fl}(x \odot y) \quad = \quad \frac{x \odot y}{1+\delta}, \text{ για κάποιο } |\delta| \leq \mathbf{u}$$

Παράδειγμα

Πώς δικαιολογείται σε α.κ.υ. (IEEE 64 bits, $\mathbf{u} = 2^{-53} \approx 10^{-16}$) ότι $10^{20} \tilde{+} 10 = 10^{20}$

(θυμηθείτε παράδειγμα εισαγωγής)

$$10^{20} \tilde{+} 10 = (10^{20} + 10)(1 + \delta),\ |\delta| \leq \mathbf{u},$$

άρα για να ισχύει είναι ισοδύναμο να υπάρχει αποδεκτό δ για οποίο να ισχύει η ισότητα

$$10^{20} = (10^{20} + 10)(1 + \delta)$$

Λύνοντας ως προς δ,

$$\begin{aligned} 10^{20} &= (10^{20} + 10)(1 + \delta), \\ \cancel{10^{20}} &= \cancel{10^{20}} + 10 + 10^{20}\delta + 10\delta \\ \delta &= \frac{-10}{10^{20} + 10} \approx -10^{-19} \end{aligned}$$

επομένως το υπολογισμένο αποτέλεσμα εξηγείται καθώς το παραπάνω $|\delta| \leq \mathbf{u} \approx 10^{-16}$.

Θεωρούμε ότι οι μεταβλητές x_j περιέχουν α.κ.υ.

$$\begin{aligned}(x_1 \tilde{+} x_2) \tilde{+} x_3 &= ((x_1 + x_2)(1+\delta_1) + x_3)(1+\delta_2) \\ &= x_1(1+\delta_1)(1+\delta_2) + x_2(1+\delta_1)(1+\delta_2) + x_3(1+\delta_2)\end{aligned}$$

επομένως αν θέσουμε

$$E := (x_1 \tilde{+} x_2) \tilde{+} x_3 - (x_1 + x_2 + x_3)$$

$$E = (x_1 + x_2)(\delta_1 + \delta_2 + \delta_1\delta_2) + x_3\delta_2$$

επομένως μπορούμε να φράξουμε ως εξής:

$$|E| \leq (|x_1| + |x_2|)(2\mathbf{u} + \mathbf{u}^2) + |x_3|\mathbf{u}.$$

Με τον ίδιο τρόπο καταλήγουμε και σε άνω φράγμα για το σφάλμα στον υπολογισμό του $x_1 + (x_2 + x_3)$. Θέτουμε για συντομία $\hat{E} := x_1 \tilde{+} (x_2 \tilde{+} x_3) - (x_1 + x_2 + x_3)$.
Εντέλει έχουμε τα παρακάτω άνω φράγματα για τους δύο εναλλακτικούς τρόπους υπολογισμού:

$$\begin{aligned} |E| &\leq (|x_1| + |x_2|)(2\mathbf{u} + \mathbf{u}^2) + |x_3|\mathbf{u} \\ |\hat{E}| &\leq |x_1|\mathbf{u} + (2\mathbf{u} + \mathbf{u}^2)(|x_2| + |x_3|). \end{aligned}$$

- Αναδεικνύεται λεπτομερώς η (διαφορετική) συνεισφορά του κάθε όρου στο άνω φράγμα.
- Διαφαίνεται ότι με βάση τα x_j, θα μπορούσαμε να επιλέξουμε σειρά άθροισης που να ελαχιστοποιεί το άνω φράγμα (αλλά όχι κατ' ανάγκη το σφάλμα!).

Ενδιαφέροντα συμπεράσματα

Πώς (με ποια σειρά) αθροίζουμε 3 α.κ.υ? Η παρακάτω συζήτηση θα γίνει αποκλειστικά με βάση τα παραπάνω. Περισσότερα σε επόμενη διάλεξη αφιερωμένη στην άθροιση.

Υπάρχουν 3 μη ισοδύναμοι αριθμητικά τρόποι αθροισης (δεν ισχύει προσεταιριστικότητα):

$$(x_1 + x_2) + x_3, x_1 + (x_2 + x_3), (x_1 + x_3) + x_2$$

Οι υπόλοιποι (9) τρόποι είναι αριθμητικά ισοδύναμοι με έναν απο τους παραπάνω (γιατί ισχύει αντιμεταθετικότητα), π.χ.

$$x_3 \tilde{+} (x_1 \tilde{+} x_2), x_3 \tilde{+} (x_2 \tilde{+} x_1), (x_2 \tilde{+} x_3) \tilde{+} x_1,$$
$$(x_3 \tilde{+} x_2) \tilde{+} x_1, x_2 \tilde{+} (x_1 \tilde{+} x_3), x_2 \tilde{+} (x_3 \tilde{+} x_1),$$
$$x_1 \tilde{+} (x_3 \tilde{+} x_2), (x_3 \tilde{+} x_1) \tilde{+} x_2, (x_2 \tilde{+} x_1) \tilde{+} x_3,$$

Με ποιόν τρόπο προκύπτει το μικρότερο άνω φράγμα? Με βάση τα παραπάνω, αυτός που αφήνει το μεγαλύτερο στοιχείο τελευταίο (εκτός παρένθεσης):

- Αν $|x_3| = \max(|x_1|, |x_2|, |x_3|)$ τότε $\max_{x_j \in F} |E| \leq \max_{x_j \in F} |\hat{E}|$
- Αν $|x_1| = \max(|x_1|, |x_2|, |x_3|)$ τότε $\max_{x_j \in F} |\hat{E}| \leq \max_{x_j \in F} |E|$
- Αν $|x_2| = \max(|x_1|, |x_2|, |x_3|)$ τότε επιλέγουμε $(x_1 \tilde{+} x_3) \tilde{+} x_2$.

- Όμως: Το άνω φράγμα δείχνει την χειρότερη περίπτωση!
- Τα πράγματα μπορεί να είναι πολύ καλυτερα!!
- ... για παράδειγμα αν γνωρίζουμε ότι $x_1 = -x_2$, ή/και το αντίστοιχο δ να είναι 0.
- Αυτά δεν προβλέπονται αν εφαρμόσουμε το μοντέλο χωρίς ειδικές τροποποιήσεις.

Χαλαρώνοντας λίγο!

$$\begin{aligned} |E| &\leq (|x_1| + |x_2|)(2\mathbf{u} + \mathbf{u}^2) + |x_3|\mathbf{u} \\ &\leq (|x_1| + |x_2| + |x_3|)2\mathbf{u} + (|x_1| + |x_2|) \underbrace{\mathbf{u}^2}_{\text{αμελητέο}} \end{aligned}$$

αν αγνοήσουμε όρους με παράγοντα $\mathbf{u}^2$ (δηλ. όρους 2ης τάξης) μπορούμε να φράξουμε αμφότερους όρους $||, |\hat{E}|$

$$||, |\hat{E}| \leq (|x_1| + |x_2| + |x_3|)2\mathbf{u}.$$

ΠΡΟΣΟΧΗ Από τα παραπάνω και μόνον το σχετικό σφάλμα φράσσεται ως εξής (ομοίως και για το $|\hat{E}|$):

$$\frac{|E|}{|x_1 + x_2 + x_3|} \leq \frac{|x_1| + |x_2| + |x_3|}{|x_1 + x_2 + x_3|} 2\mathbf{u}$$

ΕΡΩΤΗΣΗ: Ποιο είναι το μειονέκτημα αυτού του φράγματος;

Ποιο είναι το μειονέκτημα αυτού του φράγματος;

- Το φράγμα εξαρτάται από τις τιμές των x_1, x_2, x_3 ...
- Ο όρος $\frac{|x_1|+|x_2|+|x_3|}{|x_1+x_2+x_3|}$ μπορεί να γίνει πολύ μεγάλος.

Αν υπολογίσουμε (μη μηδενικό) $x_1 \tilde{\times} x_2 \tilde{\times} x_3$, τότε

$$\begin{aligned}
\frac{|(x_1 \tilde{\times} x_2) \tilde{\times} x_3 - x_1 x_2 x_3|}{|x_1 x_2 x_3|} &= \frac{|(x_1 x_2)(1+\delta_1) x_3 (1+\delta_2) - x_1 x_2 x_3|}{|x_1 x_2 x_3|} \\
&= \frac{|x_1 x_2 x_3 (\delta_1 + \delta_2 + \delta_1 \delta_2)|}{|x_1 x_2 x_3|} \\
\frac{|(x_1 \tilde{\times} x_2) \tilde{\times} x_3 - x_1 x_2 x_3|}{|x_1 x_2 x_3|} &\leq 2\mathbf{u} + \mathbf{u}^2
\end{aligned}$$

Σημαντική ειδική περίπτωση

Όταν οι όροι x_j είναι ομόσημοι, $|x_1| + |x_2| + |x_3| = |x_1 + x_2 + x_3|$, επομένως

$$\frac{|E|}{|x_1 + x_2 + x_3|} \leq 2\mathbf{u}.$$

Παραδείγματα:

- άθροισμα ομόσημων όρων (π.χ. μη αρνητικών στατιστικών μεγεθών, μετρήσεων, ...)
- υπολογισμός νόρμας πραγματικών διανυσμάτων

Γενικεύσεις

Εύρεση άνω φράγματος του εμπρός σφάλματος για τον υπολογισμό των

- ... $(x_1 + x_2) + x_3$ όταν $x_j \in G$ και όχι κατ' ανάγκη α.κ.υ.
- ... αθροίσματος n αριθμών (για εναλλακτικούς τρόπους άθροισης)
- ... εσωτερικού γινομένου διανυσμάτων ...

Π.χ. στην πρώτη περίπτωση

$$\begin{aligned} \mathrm{fl}((x_1 + x_2) + x_3) &= \mathrm{fl}(\mathrm{fl}(\mathrm{fl}(x_1) + \mathrm{fl}(x_2)) + \mathrm{fl}(x_3)) \\ &= ((x_1(1+\delta_1) + x_2(1+\delta_2))(1+\delta_3) + (1+\delta_4)x_3)(1+\delta_5 \end{aligned}$$

Το παραπάνω μπορεί να ξαναγραφτεί ως:

$$x_1(1+\delta_1)(1+\delta_3)(1+\delta_5) + x_2(1+\delta_2)(1+\delta_3)(1+\delta_5) + x_3(1+\delta_4)(1+\delta_5)$$

Οι εκφράσεις γίνονται πολύπλοκες ακόμα και στην απλή αυτή μελέτη!

Παράδειγμα - το γνωστό κουίζ

$$\begin{aligned} 10^{20} - 10 - 10^{20} + 20 &= 20 \\ 10^{20} + 20 - 10^{20} - 10 &= -10 \\ -10 + 20 - 10^{20} + 10^{20} &= 0 \\ 10^{20} - 10^{20} + 20 - 10 &= 10 \end{aligned}$$

Για παράδειγμα, $10^{20} - 10 - 10^{20} + 20 = 20$?

Προσοχή: Θεωρούμε ότι οι πράξεις εκτελούνται από αριστερά προς τα δεξιά.

$$(((10^{20} - 10)(1 + \delta_1) - 10^{20})(1 + \delta_2) + 20)(1 + \delta_3)$$

$$10^{20}(1 + \delta_1)(1 + \delta_2)(1 + \delta_3) - 10(1 + \delta_1)(1 + \delta_2)(1 + \delta_3) - 10^{20}(1 + \delta_2)(1 + \delta_3) + 20(1 + \delta_3)$$

Παράδειγμα - το γνωστό κουίζ

$$\begin{aligned} 10^{20} - 10 - 10^{20} + 20 &= 20 \\ 10^{20} + 20 - 10^{20} - 10 &= -10 \\ -10 + 20 - 10^{20} + 10^{20} &= 0 \\ 10^{20} - 10^{20} + 20 - 10 &= 10 \end{aligned}$$

Για παράδειγμα, $10^{20} - 10 - 10^{20} + 20 = 20$?

Προσοχή: Θεωρούμε ότι οι πράξεις εκτελούνται από αριστερά προς τα δεξιά.

$$(((10^{20} - 10)(1+\delta_1) - 10^{20})(1+\delta_2) + 20)(1+\delta_3)$$

$$10^{20}(1+\delta_1)(1+\delta_2)(1+\delta_3) - 10(1+\delta_1)(1+\delta_2)(1+\delta_3) - 10^{20}(1+\delta_2)(1+\delta_3) + 20(1+\delta_3)$$

Ερώτημα: Είναι ίσο με 20 για επιτρεπτές τιμές των δ?

Παράδειγμα - το γνωστό κουίζ

$$\begin{aligned} 10^{20} - 10 - 10^{20} + 20 &= 20 \\ 10^{20} + 20 - 10^{20} - 10 &= -10 \\ -10 + 20 - 10^{20} + 10^{20} &= 0 \\ 10^{20} - 10^{20} + 20 - 10 &= 10 \end{aligned}$$

Για παράδειγμα, $10^{20} - 10 - 10^{20} + 20 = 20$?
Προσοχή: Θεωρούμε ότι οι πράξεις εκτελούνται από αριστερά προς τα δεξιά.

$$(((10^{20} - 10)(1+\delta_1) - 10^{20})(1+\delta_2) + 20)(1+\delta_3)$$
$$10^{20}(1+\delta_1)(1+\delta_2)(1+\delta_3) - 10(1+\delta_1)(1+\delta_2)(1+\delta_3) - 10^{20}(1+\delta_2)(1+\delta_3) + 20(1+\delta_3)$$

Ερώτημα: Είναι ίσο με 20 για επιτρεπτές τιμές των δ?

Αν $\delta_1 = 1/(10^{19} - 1), \delta_2 = \delta_3 = 0$, τότε ισχύει ισότητα.
Αφού $|\delta_j| \leq \mathbf{u}$, οι τιμές είναι επιτρεπτές, άρα το αποτέλεσμα είναι αιτιολογημένο.

Πολύ χρήσιμο εργαλείο

Για την απλοποίηση όρων όπως $p_n = \prod_{i=1}^{n}(1+\delta_i)$ όταν γνωρίζουμε ότι $|\delta_i| \leq u$. Αμέσως βλέπουμε ότι

$$(1-u)^n \leq p_n \leq (1+u)^n.$$

και ότι $p_n = 1 + nu + \mathbf{O}(u^2)$.

Ακόμα καλύτερα:

Λήμμα

Αν $|\delta_i| \leq u$ και $\rho_i = \pm 1$ για $i = 1 : n$ και $nu < 1$ τ'οτε υπάρχει κάποια τιμή θ_n τ.ώ.

$$\prod_{i=1}^{n}(1+\delta_i)^{\rho_i} = 1 + \theta_n,$$

όπου

$$|\theta_n| \leq \frac{nu}{1-nu} := \gamma_n.$$

Εφαρμογή

Εύρεση άνω φράγματος του εμπρός σφάλματος για τον υπολογισμό του

- $(x_1 + x_2) + x_3$ όταν $x_j \in G$ και όχι κατ' ανάγκη a.κ.υ.

Με βάση το Λήμμα, υπάρχουν $\theta_3, \zeta_3, \theta_2$ τέτοια ώστε

$$|\theta_3| \leq \gamma_3, |\zeta_3| \leq \gamma_3, |\theta_2| \leq \gamma_2$$

και μπορούμε να γράψουμε

$$\mathrm{fl}((x_1 + x_2) + x_3) \quad = \quad x_1(1 + \theta_3) + x_2(1 + \zeta_3) + x_3(1 + \theta_2)$$

επομένως ένα (χαλαρό) άνω φράγμα για το σφάλμα θα είναι

$$\begin{aligned} |\mathrm{fl}((x_1 + x_2) + x_3) - (x_1 + x_2 + x_3)| &\leq |x_1|\gamma_3 + |x_2|\gamma_3 + |x_3|\gamma_2 \\ &\leq (|x_1| + |x_2| + |x_3|)\gamma_3. \end{aligned}$$

Το σχετικό σφάλμα φράσσεται άμεσα από

$$\frac{|\mathrm{fl}((x_1 + x_2) + x_3) - (x_1 + x_2 + x_3)|}{|x_1 + x_2 + x_3|} \quad \leq \quad \frac{(|x_1| + |x_2| + |x_3|)}{|x_1 + x_2 + x_3|}\gamma_3.$$

Για ομόσημα το άνω φράγμα είναι $\gamma_3 = \gamma_3 = \frac{3\mathbf{u}}{1-3\mathbf{u}}$.

Κριτική

Εκκινώντας από τα στοιχεία εισόδου και παρακολουθώντας το σφάλμα σε κάθε πράξη προσπαθούμε να φράξουμε το μέγιστο απόλυτο ή σχετικό σφάλμα που θα μπορούσε να προκύψει στο τελικό αποτέλεσμα.

Η ιδέα είναι απλή

- ... η εφαρμογή της μπορεί να είναι περίπλοκη
- ... σκληρή άσκηση σε ανισότητες
- ... τεράστιες εκφράσεις, κ.λπ.

Παρατηρήσεις

Έχουν γίνει πολλές προσπάθειες αυτοματοποίησης της ανάλυσης των σφαλμάτων που υπεισέρχονται στις υπολογιστικές διαδικασίες με μέτρια ή μεγαλύτερη επιτυχία.

Εμπρός ανάλυση σφάλματος: Θεωρούμε τον αλγόριθμο ως μιά σειρά στοιχειωδών πράξεων. Σε κάθε βήμα, υπολογίζεται μία τιμή α_{k+1} με βάση προηγούμενες τιμές και στοιχεία εισόδου, π.χ. $\alpha_{k+1} = g_k(\alpha_1, \ldots, \alpha_k)$. Μερικές από τις τιμές μπορεί να είναι δεδομένα εισόδου. Στη συνέχεια, υπολογίζουμε φράγματα για τα σφάλματα στα τελικά αποτελέσματα.

Παρατηρήσεις

Έχουν γίνει πολλές προσπάθειες αυτοματοποίησης της ανάλυσης των σφαλμάτων που υπεισέρχονται στις υπολογιστικές διαδικασίες με μέτρια ή μεγαλύτερη επιτυχία.

Εμπρός ανάλυση σφάλματος: Θεωρούμε τον αλγόριθμο ως μιά σειρά στοιχειωδών πράξεων. Σε κάθε βήμα, υπολογίζεται μία τιμή α_{k+1} με βάση προηγούμενες τιμές και στοιχεία εισόδου, π.χ. $\alpha_{k+1} = g_k(\alpha_1, \ldots, \alpha_k)$. Μερικές από τις τιμές μπορεί να είναι δεδομένα εισόδου. Στη συνέχεια, υπολογίζουμε φράγματα για τα σφάλματα στα τελικά αποτελέσματα.

Ανάλυση διαστημάτων: Θεωρούμε ότι κάθε δεδομένο x εγκλείεται σε κάποιο διάστημα $[x_L, x_U)$ οπότε οι πράξεις επί των δεδομένων εκτελούνται χρησιμοποιώντας ((αριθμητική διαστημάτων)) (interval arithmetic).

Παρατηρήσεις

Έχουν γίνει πολλές προσπάθειες αυτοματοποίησης της ανάλυσης των σφαλμάτων που υπεισέρχονται στις υπολογιστικές διαδικασίες με μέτρια ή μεγαλύτερη επιτυχία.

Εμπρός ανάλυση σφάλματος: Θεωρούμε τον αλγόριθμο ως μιά σειρά στοιχειωδών πράξεων. Σε κάθε βήμα, υπολογίζεται μία τιμή α_{k+1} με βάση προηγούμενες τιμές και στοιχεία εισόδου, π.χ. $\alpha_{k+1} = g_k(\alpha_1, \ldots, \alpha_k)$. Μερικές από τις τιμές μπορεί να είναι δεδομένα εισόδου. Στη συνέχεια, υπολογίζουμε φράγματα για τα σφάλματα στα τελικά αποτελέσματα.

Ανάλυση διαστημάτων: Θεωρούμε ότι κάθε δεδομένο x εγκλείεται σε κάποιο διάστημα $[x_L, x_U)$ οπότε οι πράξεις επί των δεδομένων εκτελούνται χρησιμοποιώντας ((αριθμητική διαστημάτων)) (interval arithmetic).

Πίσω ανάλυση σφάλματος: Στη συνέχεια!

Κριτική

Η ιδέα της εμπρός ανάλυσης είναι απλή, αλλά

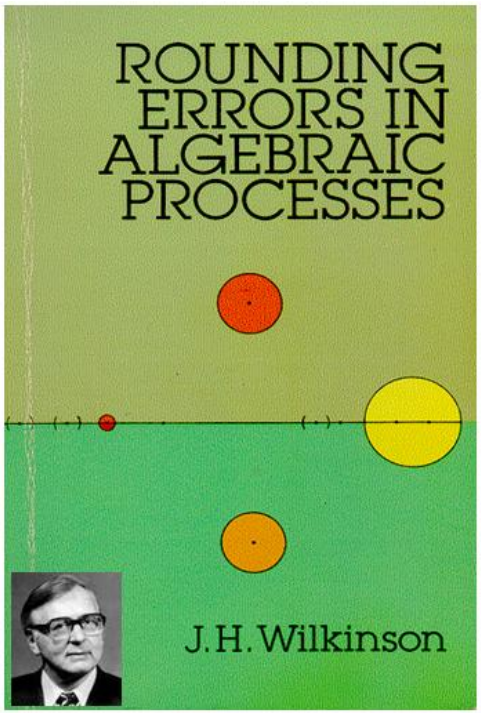
- ... η εφαρμογή της μπορεί να είναι περίπλοκη
- ... σκληρή άσκηση σε ανισότητες
- ... τεράστιες εκφράσεις, κ.λπ.
- ΠΡΟΣΟΧΗ ... δεν λέει κάτι ξεκάθαρο για τον αλγόριθμο.

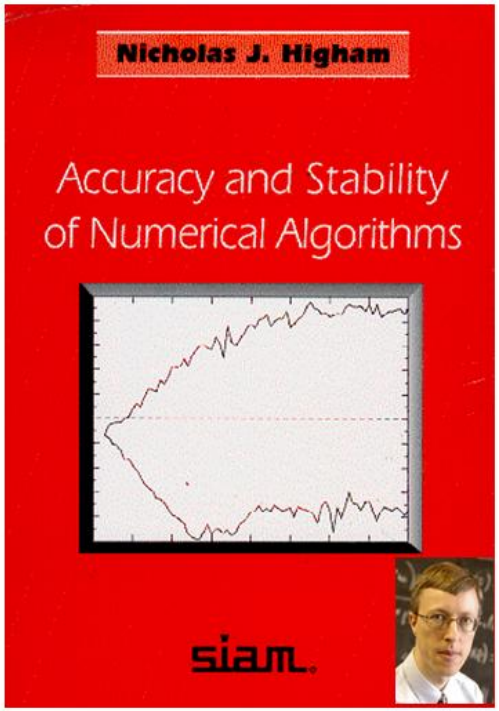
Κριτική

Η ιδέα της εμπρός ανάλυσης είναι απλή, αλλά

- ... η εφαρμογή της μπορεί να είναι περίπλοκη
- ... σκληρή άσκηση σε ανισότητες
- ... τεράστιες εκφράσεις, κ.λπ.
- ΠΡΟΣΟΧΗ ... δεν λέει κάτι ξεκάθαρο για τον αλγόριθμο.

  Vel Kahan: "In the 1950's almost no programmers knew how to distinguish numerical instability from ill condition. In other words, the distinction between an algorithm's hypersensitivity to its own internal rounding errors, and a true solution's hypersensitivity to roundoff-like errors in its problem's data, as contributors to wrongly computed solutions, was obscure. By the 1960s the distinction had been clarified through "Bacwkward Error Analyses" ...."





J.H. Wilkinson: "It is a matter of some surprise that one of the simplest methods of solution generally leads to an error, the expected value of which is precisely that resulting from random perturbations ... This means that when the original elements are not exactly representable ... the errors resulting from any initial rounding that may be necessary are as serious as those arising from any initial rounding from all the steps in the solution..."

Καταστάσεις και δείκτες τους (**conditioning**)

Έννοιες που χρησιμεύουν για τη μέτρηση και ποσοτικοποίηση της επίδρασης διαταραχών και σφαλμάτων στα δεδομένα του προβλήματος και στις πράξεις α.κ.υ. επί των υπολογισμένων αποτελεσμάτων.

Κατάσταση προβλήματος

Αφορά στην ευαισθησία/μεταβλητότητα των αποτελεσμάτων της f που οφείλονται αποκλειστικά σε (μικρές) διαταραχές των στοιχείων εισόδου.

Κατάσταση (υλοποίησης) αλγορίθμου

Αφορά στην επίδραση της αριθμητικής πεπερασμένης ακρίβειας στην υλοποίηση του αλγορίθμου (μέσω προγράμματος.)

Πρός τα πίσω ανάλυση - προς τα πίσω ευστάθεια

Ορισμός

Με τον γνωστό συμβολισμό, έστω ότι υπάρχει x_{prog} κοντά στο x τέτοιο ώστε $f_{\text{prog}}(x) = f(x_{\text{prog}})$. Τότε ο αλγόριθμος χαρακτηρίζεται (αριθμητικά) ((προς τα πίσω ευσταθής)) στο x.

Αν αυτό συμβαίνει για κάθε x στο πεδίο ορισμού της f ο αλγόριθμος αποκαλείται ((προς τα πίσω ευσταθής))

((Ένας πίσω ευσταθής αλγόριθμος απαντά με ακρίβεια σε ένα παραπλήσιο ερώτημα.))

Κρίσιμη παρατήρηση: Αν κατασκευάσουμε x_{prog} ((κοντά)) στο x ώστε αν το αποτέλεσμα που υπολογίστηκε με πράξεις α.κ.υ. είναι $z_{\text{prog}} = f_{\text{prog}}(x)$ τότε $z_{\text{prog}} = f(x_{\text{prog}})$. Τότε

$$\|z_{\text{prog}} - z\| \quad = \quad \|f_{\text{prog}}(x) - f(x)\| = \|f(x_{\text{prog}}) - f(x)\|.$$

Ιδέα

Αντί να εκτιμούμε το προς τα εμπρός σφάλμα απευθείας, εκτιμούμε πρώτα τη διαφορά $\left\| f(x_{\text{prog}}) - f(x) \right\|$ που αφορά την ευαισθησία της f σε αλλαγές.
Αν υπάρχει x_{prog} ώστε το $\left\| x_{\text{prog}} - x \right\|$ να είναι μικρό, τότε το πρόβλημα ανάγεται στο μαθηματικό (και όχι αριθμητικό) πρόβλημα της ευαισθησίας της f σε μικρές διαταράξεις (= perturbations) των στοιχείων εισόδου.

> ((Ρίξαμε το σφάλμα προς τα πίσω))[1] αναγάγοντάς το σε (εικονικές) διαταραχές των στοιχείων εισόδου. Από εκεί και πέρα δεν χρειάζεται να ασχοληθούμε με τα σφάλματα των ενδιάμεσων πράξεων!

Η f_{prog} αποτελείται από όλες τις στοιχειώδεις πράξεις (ό,τι επηρεάζει δεδομένα α.κ.υ. του προβλήματος) και με τη σειρά που εκτελούνται από το πρόγραμμα που υλοποιεί την f.

- Δεν περιμένουμε η υλοποίηση f_{prog} να λύνει το πρόβλημα με πιστότητα αν τα δεδομένα x^* είναι κοντά σε στοιχεία όπου η f είναι ((ευαίσθητη)).
- Αυτό που μας ενδιαφέρει είναι η υλοποίηση της f να μην εισάγει από μόνη της ((μεγάλα)) σφάλματα.

[1] Θυμηθείτε το συμπέρασμα του Wilkinson!

Έχουμε ήδη συναντήσει την πίσω ευστάθεια!

Θυμηθείτε την *Αρχή Ακριβούς Στρογγύλευσης:*

Αν $\tilde{\odot}$ είναι η υλοποίηση της αριθμητικής πράξης $\odot$, τότε αν $x, y \in F$ ισχύει ότι

$$x\tilde{\odot}y = \mathrm{fl}(x \odot y) = (x \odot y)(1+\delta), \text{ για κάποιο } |\delta| \leq \mathbf{u}.$$

Παράδειγμα: από άμεση επαλήθευση,

$$f_{\mathrm{prog}}(x) \quad = \quad f(x_{\mathrm{prog}})$$

όπου

$$\begin{aligned} x &= [\xi_1, \xi_2]^\top \\ x_{\mathrm{prog}} &= [\xi_1(1+\delta), \xi_2(1+\delta)] \end{aligned}$$

και

$$\|x_{\mathrm{prog}} - x\| \quad = \quad \|[\xi_1\delta, \xi_2\delta]^\top\| \leq \|x\|\mathbf{u}$$

Συμπέρασμα Όλες οι απλές αριθμητικές πράξεις ικανοποιούν την αρχή ακριβούς στρογγύλευσης επομένως είναι πίσω ευσταθείς.

Ασυμπτωτικός δείκτης κατάστασης προβλήματος (Rice' 66)

Δίνονται δύο νορμισμένοι γραμμικοί χώροι $\mathcal{X}, \mathcal{Y}$ και έστω $f : \Omega \subset X \to Y$, όπου Ω ανοικτό χωρίο. Έστω x^* σταθερή τιμή και ότι $y^* := f(x^*)$. Υποθέτουμε ότι τα x^*, y^* δεν είναι μηδέν. *Ο ασυμπτωτικός δείκτης κατάστασης της απεικόνισης f στο x^* για σχετικά απειροελάχιστες αλλαγές του x^** ορίζεται ως

$$\mathrm{cond}(f; x^*) := \lim_{\delta \to 0} \sup_{\|h\| = \delta} \left\{ \frac{\frac{\|f(x^* + h) - f(x^*)\|}{\|f(x^*)\|}}{\frac{\|h\|}{\|x^*\|}} \right\}$$

εφόσον το όριο υπάρχει. Με άλλα λόγια: Ποιά είναι η σχετική αλλαγή στην τιμή μιας συνάρτησης αν αλλάξουμε σχετικά απειροελάχιστα τις τιμές των μεταβλητών της?

Παράδειγμα: Αν $X = Y = \mathbb{R}$ και η f διαφορίσιμη τότε

$$\mathrm{cond}(f; x^*) = |f'(x^*)| \frac{|x^*|}{|f(x^*)|}$$

Προσοχή:

- Αν για κάποιο σύνολο τιμών $x^* \in X^* \subseteq \mathcal{X}$ ισχύει ότι $\mathbf{cond}(f; x^*)$ είναι μη φραγμένο ή πάρα πολύ μεγάλο, τότε ο υπολογισμός της f καθίσταται προβληματικός επί των τιμών του $\mathcal{X}$ ανεξαρτήτως του αλγορίθμου που χρησιμοποιείται.
- Για τις τιμές αυτές, η συνάρτηση f είναι "δύσκολη" στη διαχείρισή της γιατί μικρές αλλαγές στις τιμές της μεταβλητής της x^* οδηγεί σε πολύ μεγάλες αλλαγές στην τιμή της $f(x^*)$.
- Το πρόβλημα της εύρεσης του $f(x^*)$ όταν ο ασυμπτωτικός δείκτης κατάστασης είναι πολύ μεγάλος αποκαλείται κακώς τοποθετημένο.
- Τα κακώς τοποθετημένα προβλήματα και στρατηγικές για την αντιμετώπισή τους είναι παρουσιάζουν εξαιρετικό ενδιαφέρον γιατί είναι ιδιαίτερα δύσκολα αλλά απαντώνται πολύ συχνά στην πραγματικότητα.

Καλά και κακά τοποθετημένα προβλήματα I

Θυμηθείτε τις κατηγορίες των προβλημάτων (άμεσα, αντίστροφα, ταυτοποίησης). Η (κακή) τοποθέτηση ως ζήτημα συνήθως δεν αφορά τα άμεσα αλλά τις άλλες δύο κατηγορίες. Η συζήτηση συνήθως διεξάγεται για τα αντίστροφα προβλήματα.

Καλά τοποθετημένο (well-posed) πρόβλημα

Ως καλά τοποθετημένο εννούμε ένα πρόβλημα π.χ. τη λύση του $\phi(y) = x$ όπου $\phi : Y \longrightarrow X$ είναι γραμμικοί χώροι και 1) για κάθε $\hat{x} \in X$ υπάρχει λύση $\hat{y} \in Y$. 2) Η λύση είναι μοναδική, δηλαδή η συνάρτηση ϕ είναι αντιστρέψιμη, έστω $\phi^{-1} = f$. 3) Για κάθε $\bar{y}$ πλησίον της λύσης $\hat{y}$ υπάρχει $\bar{x}$ πλησίον του $\hat{x}$ τέτοιο ώστε $\phi(\bar{y}) = \bar{x}$. Ισοδύναμα, η $f = \phi^{-1}$ είναι συνεχής.

Κακά τοποθετημένο (ill-posed) πρόβλημα

Κακά τοποθετημένο λέγεται ένα πρόβλημα για το οποίο δεν ισχύει κάποιες από τις 3 παραπάνω συνθήκες.

- Οι παραπάνω ορισμοί είναι κλασικοί και παλαιοί (Hadamard'1902).

Καλά και κακά τοποθετημένα προβλήματα II

- Κακώς τοποθετημένα είναι πολλά προβλήματα σε πραγματικές εφαρμογές (δείτε παρακάτω).
- Η επίλυσή τους είναι μεγάλη πρόκληση σήμερα και απαιτεί διεπιστημονικές προσεγγίσεις που συνδυάζουν μαθηματικά, πιθανότητες, στατιστική, αριθμητική ανάλυση, HPC (high performance computing) καθώς γνώσεις της εφαρμογής όπου προκύπτει το κάθε πρόβλημα.
- Τεχνικές επίλυσης κακά τοποθετημένων προβλημάτων αποτελούν σημαντική πρόκληση για τη σημερινή έρευνα σε πολλές περιοχές.
- Το Gene Golub SIAM Summer School 2018 είναι αφιερωμένο στο ζήτημα.
- Δείτε πάλι σελ. 33-37 από το σύγγραμα Quarteroni et al. (QSS00, Ch. 2).
- (προαιρετικά) διαβάστε όσο μπορείτε από την αναφορά (Kab08).
- (προαιρετικά, ακολουθήστε τους συνδέσμους) για συζητήσεις σχετικά με ζητήματα τοποθέτησης στην επεξεργασία σήματος, στη μηχανική μάθηση.

Εφαρμογές Ι

Διαβάστε:

> *(Kab08) Attempting to understand a substantially complex phenomenon and solve a problem such that the probability of error is high, we usually arrive at an unstable (ill-posed) problem. Ill-posed problems are ubiquitous in our daily lives. Indeed, everyone realizes how easy it is to make a mistake when reconstructing the events of the past from a number of facts of the present (for example, to reconstruct a crime scene based on the existing direct and indirect evidence, determine the cause of a disease based on the results of a medical examination, and so on). The same is true for tasks that involve predicting the future (predicting a natural disaster or simply producing a one week weather forecast) or "reaching into" inaccessible zones to explore their structure (subsurface exploration in geophysics or examining a patient's brain using NMR tomography).*

J. Inv. Ill-Posed Problems **16** (2008), 317–357

DOI 10.1515 / JIIP.2008.069

Definitions and examples of inverse and ill-posed problems

S. I. Kabanikhin

Survey paper

Abstract. The terms "inverse problems" and "ill-posed problems" have been steadily and surely gaining popularity in modern science since the middle of the 20th century. A little more than fifty years of studying problems of this kind have shown that a great number of problems from various branches of classical mathematics (computational algebra, differential and integral equations, partial differential equations, functional analysis) can be classified as inverse or ill-posed, and they are among the most complicated ones (since they are unstable and usually nonlinear). At the same time, inverse and ill-posed problems began to be studied and applied systematically in physics, geophysics, medicine, astronomy, and all other areas of knowledge where mathematical methods are used. The reason is that solutions to inverse problems describe important properties of media under study, such as density and velocity of wave propagation, elasticity parameters, conductivity, dielectric permittivity and magnetic permeability, and properties and location of inhomogeneities in inaccessible areas, *etc*.

In this paper we consider definitions and classification of inverse and ill-posed problems and describe some approaches which have been proposed by outstanding Russian mathematicians A. N. Tikhonov, V. K. Ivanov and M. M. Lavrentiev.

Key words. Inverse and ill-posed problems, regularization.

AMS classification. 65J20, 65J10, 65M32.

Well-posed problems	Ill-posed problems
Arithmetic	
Multiplication by a small number A $Aq = f$	Division by a small number $q = A^{-1}f \quad (A \ll 1)$
Algebra	
Multiplication by a matrix $Aq = f$	$q = A^{-1}f$, A is an ill-conditioned, degenerate or rectangular $m \times n$-matrix
Calculus	
Integration $f(x) = f(0) + \int_0^x q(\xi)\,d\xi$	Differentiation $q(x) = f'(x)$
Differential equations	
The Sturm–Liouville problem $u''(x) - q(x)u(x) = \lambda u(x)$, $u(0) - hu'(0) = 0$, $u(1) - Hu'(1) = 0$	The inverse Sturm–Liouville problem. Find $q(x)$ using spectral data $\{\lambda_n, \|u_n\|\}$
Integral geometry	
Find integrals $\int_{\Gamma(\xi,\eta)} q(x,y)\,ds$	Find q from $\int_{\Gamma(\xi,\eta)} q(x,y)\,ds = f(\xi,\eta)$
Integral equations	
Volterra equations and Fredholm equations of the second kind $q(x) + \int_0^x K(x,\xi)q(\xi)\,d\xi = f(x)$ $q(x) + \int_a^b K(x,\xi)q(\xi)\,d\xi = f(x)$	Volterra equations and Fredholm equations of the first kind $\int_0^x K(x,\xi)q(\xi)\,d\xi = f(x)$ $\int_a^b K(x,\xi)q(\xi)\,d\xi = f(x)$
Operator equations $Aq = f$	
$\exists m > 0$: $\forall q \in Q$ $m\langle q, q\rangle \le \langle Aq, q\rangle$	$A : D(A) \subset Q \to R(A) \subset F$ A is a compact linear operator with singular values $\sigma_n \searrow 0$, $n \to \infty$

Λεπτομερής μέτρηση ευαισθησίας συνάρτησης

Ιακωβιανό μητρώο

Το μητρώο $J = \left[\frac{\partial f_i}{\partial \xi_j}\right] \in \mathbb{R}^{m \times n}$ αποκαλείται Ιακωβιανό[α΄] της f στο x.

[α΄] γενίκευση της 1ης παραγώγου για συναρτήσεις πολλών μεταβλητών.

Δείκτες ευαισθησιας

Έστω $x = (\xi_1, \ldots, \xi_m)$ και $y = (\eta_1, \ldots, \eta_n) = f(x)$ και ομοίως $y^* = f(x^*)$ για κάποιο x^*. Τότε αν η f είναι διαφορίσιμη στο x, οι mn παράγοντες

$$\kappa_{ij} = \frac{\xi_j \frac{\partial f_i}{\partial \xi_j}}{f_i(x)}$$

δείχνουν την ευαισθησία της f ως προς διαταραχές των στοιχείων x.

Ειδικές περιπτώσεις: Αν $x = 0$, $f(x) = 0$ χρησιμοποιούμε δείκτη, $\left|f'(x)\right|$.

Δείκτες κατάστασης κοινών (βαθμωτών) συναρτήσεων

Άθροισμα n αριθμών $\dfrac{\sum_{j=1}^{n} |\xi_j|}{|\sum_{j=1}^{n} \xi_j|}$

Εσωτερικό γινόμενο $\dfrac{2\sum_{j=1}^{n} |\xi_j \psi_j|}{|\sum_{j=1}^{n} \xi_j \psi_j|}$

Τιμή πολυωνύμου σε μορφή δύναμης $\dfrac{\sum_{j=1}^{n} |\alpha_j||x^j|}{|\sum_{j=1}^{n} \alpha_j x^j|}$

Απλή ρίζα πολυωνύμου σε μορφή δύναμης $\dfrac{\sum_{j=1}^{n} |\alpha_j||\xi^j|}{|\xi p'(\xi)|}$

Παράδειγμα

Έστω $f(\xi_1, \xi_2) = \xi_1 - \xi_2$. Τότε $f : \mathbb{R}^2 \to \mathbb{R}$ και $\frac{\partial f}{\partial x} = [1, -1]$

$$\frac{|f(x + \Delta x) - f(x)|}{|f(x)|} = \frac{|((\xi_1 + \Delta\xi_1) - (\xi_2 + \Delta\xi_2)) - (\xi_1 - \xi_2)|}{|\xi_1 - \xi_2|}$$

$$= \frac{|\Delta\xi_1 - \Delta\xi_2|}{|\xi_1 - \xi_2|}$$

Το σχετικό σφάλμα γίνεται μεγάλο όταν η πραγματική τιμή $|f(\xi_1, \xi_2)| = |\xi_1 - \xi_2|$ είναι πολύ μικρή σε σύγκριση με τη διαφορά $|\Delta\xi_1 - \Delta\xi_2|$.

Είναι η περίπτωση της καταστροφικής απαλοιφής και αφορά στο πρόβλημα (αφαίρεση) για τα συγκεκριμένα δεδομένα (ομόσημα και σε μικρή απόσταση μεταξύ τους) και όχι στην υλοποίηση (αλγόριθμος αφαίρεσης) στην ALU.

Πίσω σφάλμα και ((δείκτης κατάστασης αλγορίθμου))

Ορισμός

Αν γνωρίζουμε ποιό είναι το x_{prog} που ελαχιστοποιεί το $\frac{\|x-x_{\text{prog}}\|}{\|x\|}$ για κάθε $x \in \mathcal{U}$ τότε μπορούμε να ορίσουμε το δείκτη κατάστασης του αλγορίθμου (στην πραγματικότητα - της υλοποίησής του). Θα είναι η μικρότερη τιμή, που συμβολίζουμε με $\mathbf{cond}(f_{\text{prog}})$, για την οποία ισχύει η ανισότητα

$$\frac{\|x - x_{\text{prog}}\|}{\|x\|} \leq \mathbf{cond}(f_{\text{prog}})\mathbf{u}.$$

για όλα τα έγκυρα x_{prog}.

- Ο δείκτης $\mathbf{cond}(f_{\text{prog}})\mathbf{u}$ δηλώνει το πίσω σφάλμα ως προς όλο το πεδίο ορισμού $\mathcal{U}$.
- Η τιμή δεν είναι ενιαία: μπορεί να υπάρχουν περιοχές του $\mathcal{U}$ που είναι μεγάλη, ενώ σε άλλες να είναι μικρή.
- Στις περιπτώσεις αυτές, μπορεί να προτιμάμε κάποιον αλγόριθμο για την "εύκολη" περιοχή και άλλον για την "δύσκολη".
- Συχνά αδυνατούμε να βρούμε την καλύτερη δυνατή τιμή του άνω φράγματος και

Παράδειγμα

Συμπεριφορά της LU για την επίλυση του $Ax = b$ και επιλογή αλγορίθμου:

- α) Αν διαγώνια κυρίαρχο, η LU χωρίς οδήγηση είναι πίσω ευσταθής.
- β) Για τα περισσότερα μητρώα, η LU συμπεριφέρεται ως πίσω ευσταθής.
- γ) Υπάρχουν ειδικές περιπτώσεις (π.χ. μητρώο `gfpp` που η LU δεν είναι πίσω ευσταθής.)

```
function A = gfpp(T, c)
%       Reference:
%       N. J. Higham and D. J. Higham, Large growth factors in
%          Gaussian elimination with pivoting, SIAM J. Matrix Analysis and
%          Appl., 10 (1989), pp. 155-164.
%       N. J. Higham, Accuracy and Stability of Numerical Algorithms,
%         SIAM, 2nd ed.,
%          Philadelphia, PA, 2002; sec. 9.4.
```

Παράδειγμα

```
function A = gfpp(T, c)
%GFPP    (Author: Nick Higham)
% Matrix giving maximal growth factor for Gaussian elim. with pivoting.
%        GFPP(T) is a matrix of order N for which Gaussian elimination
%        with partial pivoting yields a growth factor 2^(N-1).
%        T is an arbitrary nonsingular upper triangular matrix of order N-1.
%        GFPP(T, C) sets all the multipliers to C  (0 <= C <= 1)
%        and gives growth factor (1+C)^(N-1) - but note that for T ~= EYE
% ...
%        GFPP(N, C) (a special case) is the same as GFPP(EYE(N-1), C) and
%        generates the well-known example of Wilkinson.

if ~isequal(T,triu(T)) | any(~diag(T))
error('First argument must be a nonsingular upper triangular matrix.')
end
if nargin == 1, c = 1; end
if c < 0 | c > 1
error('Second parameter must be a scalar between 0 and 1 inclusive.')
end
m = length(T);
if m == 1     % Handle the special case T = scalar
n = T; m = n-1; T = eye(n-1);
else
n = m+1;
end
A = zeros(n); L = eye(n) - c*tril(ones(n), -1);
A(:,1:n-1) = L*(T; zeros(1,n-1)); theta = max(abs(A(:)));
A(:,n) = theta * ones(n,1); A = A/theta;
```

Από τους δείκτες κατάστασης στο εμπρός σφάλμα ((Gau97))

$$\begin{aligned}\frac{\|f_{\text{prog}}(x^*) - f(x)\|}{\|f(x)\|} &= \frac{\|f(x^*_{\text{prog}}) - f(x)\|}{\|f(x)\|} \\ &\leq \frac{\|f(x^*_{\text{prog}}) - f(x^*)\|}{\|f(x)\|} + \frac{\|f(x^*) - f(x)\|}{\|f(x)\|}.\end{aligned}$$

Φράσσουμε κάθε όρο ξεχωριστά:

$$\begin{aligned}\frac{\|f(x^*_{\text{prog}}) - f(x^*)\|}{\|f(x)\|} &\leq \frac{\|f(x^*_{\text{prog}}) - f(x^*)\|}{\|f(x^*)\|}\frac{\|f(x^*)\|}{\|f(x)\|} \\ &\leq \text{cond}(f; x^*)\frac{\|x^* - x^*_{\text{prog}}\|}{\|x^*\|}\frac{\|f(x^*)\|}{\|f(x)\|} \\ &\leq \text{cond}(f; x^*)\text{cond}(f_{\text{prog}})\mathbf{u}\frac{\|f(x^*)\|}{\|f(x)\|}.\end{aligned}$$

και εξ ορισμού

$$\frac{\|f(x^*) - f(x)\|}{\|f(x)\|} \leq \mathrm{cond}(f; x) \frac{\|x^* - x\|}{\|x\|} \leq \mathrm{cond}(f; x)\mathcal{E}$$

όπου

$$\frac{\|x^* - x\|}{\|x\|} \leq \mathcal{E}$$

Επομένως

$$\frac{\|f_{\mathrm{prog}}(x^*) - f(x)\|}{\|f(x)\|} \leq \mathrm{cond}(f; x^*)\mathrm{cond}(f_{\mathrm{prog}})\mathbf{u}\frac{\|f(x^*)\|}{\|f(x)\|} + \mathrm{cond}(f; x)\mathcal{E}$$

Χονδρικά, όταν ο παράγοντας $\mathcal{E}$ είναι μικρός και $\frac{\|f(x^*)\|}{\|f(x)\|} \approx 1$, ισχύει η ανισότητα:

προς τα εμπρός σφάλμα $<$ δείκτης κατάστασης προβλήματος $\times$ πίσω σφάλμα

Γενική εικόνα

Στόχοι της ((προς τα πίσω ανάλυσης σφάλματος)) (Hig02)

Ορισμός Ονομάζουμε ((προς τα πίσω ανάλυση σφάλματος)) τη διαδικασία εύρεσης φράγματος για το πίσω σφάλμα.

Στόχοι:

1. Να ερμηνεύσουμε τα σφάλματα στρογγύλευσης στους υπολογισμούς ως ισοδύναμες διαταραχές στα δεδομένα.
2. Να αναχθεί το πρόβλημα της εύρεσης φράγματος ή της εκτίμησης του εμπρός σφάλματος σε πρόβλημα της μαθηματικής θεωρίας διαταραχών.

Ισχύει ότι

$$\frac{\|f_{\text{prog}}(x^*) - f(x)\|}{\|f(x)\|} \leq \text{cond}(f; x^*)\text{cond}(f_{\text{prog}})\mathbf{u}\frac{\|f(x^*)\|}{\|f(x)\|} + \text{cond}(f; x)\mathcal{E}$$

Μεθοδολογία

Ισχύει ότι

$$\frac{\|f_{\text{prog}}(x^*) - f(x)\|}{\|f(x)\|} \leq \text{cond}(f; x^*)\text{cond}(f_{\text{prog}})\mathbf{u}\frac{\|f(x^*)\|}{\|f(x)\|} + \text{cond}(f; x)\mathcal{E}$$

Βήματα για να φράξουμε το *εμπρός σφάλμα* $\frac{\|f_{\text{prog}}(x^*)-f(x)\|}{\|f(x)\|}$ χρησιμοποιώντας την πίσω ανάλυση σφάλματος:

Υπολογισμός κατάστασης αλγορίθμου $\text{cond}(f_{\text{prog}})$. από την οποία προκύπτει το πίσω σφάλμα.

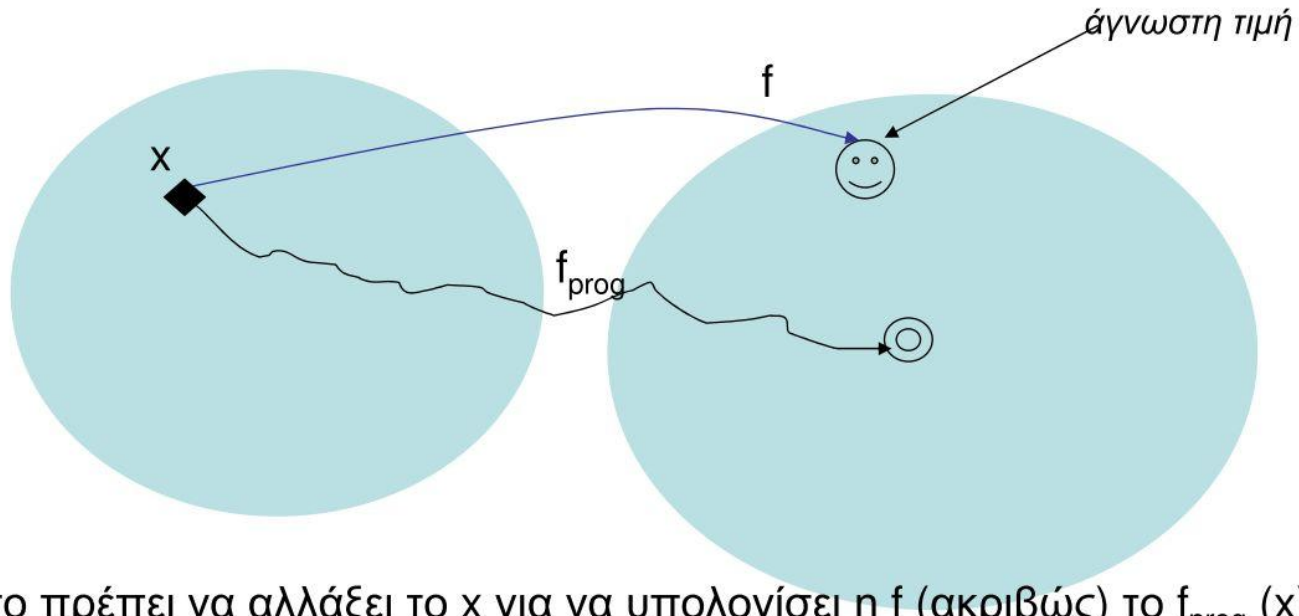
Υπολογισμός κατάστασης προβλήματος $\text{cond}(f; x)$

Εκτίμηση του $\mathcal{E}$ αν χρειάζεται.

Όταν ο παράγοντας $\mathcal{E}$ είναι μικρός και $\frac{\|f(x^*)\|}{\|f(x)\|} \approx 1$:

προς τα εμπρός σφάλμα $<$ δείκτης κατάστασης $\times$ πίσω σφάλμα

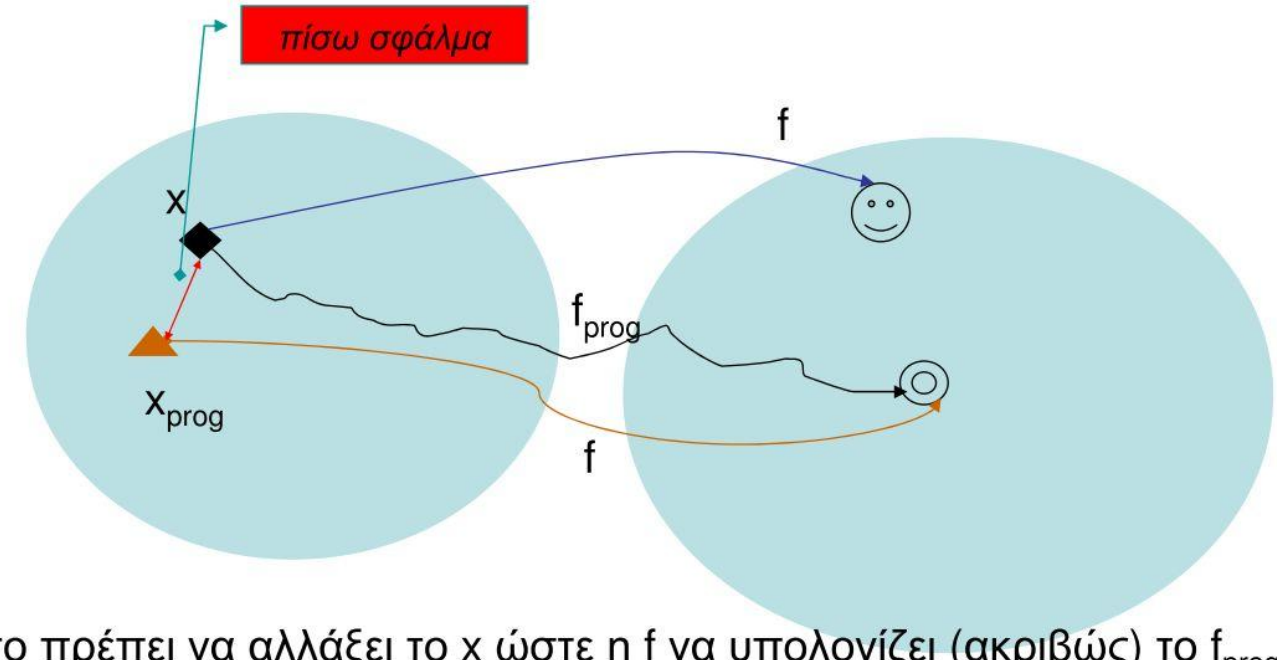
Το πρόβλημα: Να εκτιμήσουμε φράγμα για το εμπρός σφάλμα



Πόσο πρέπει να αλλάξει το x για να υπολογίσει η f (ακριβώς) το f_{prog} (x)?

$$f(x+?) = f_{prog}(x)$$

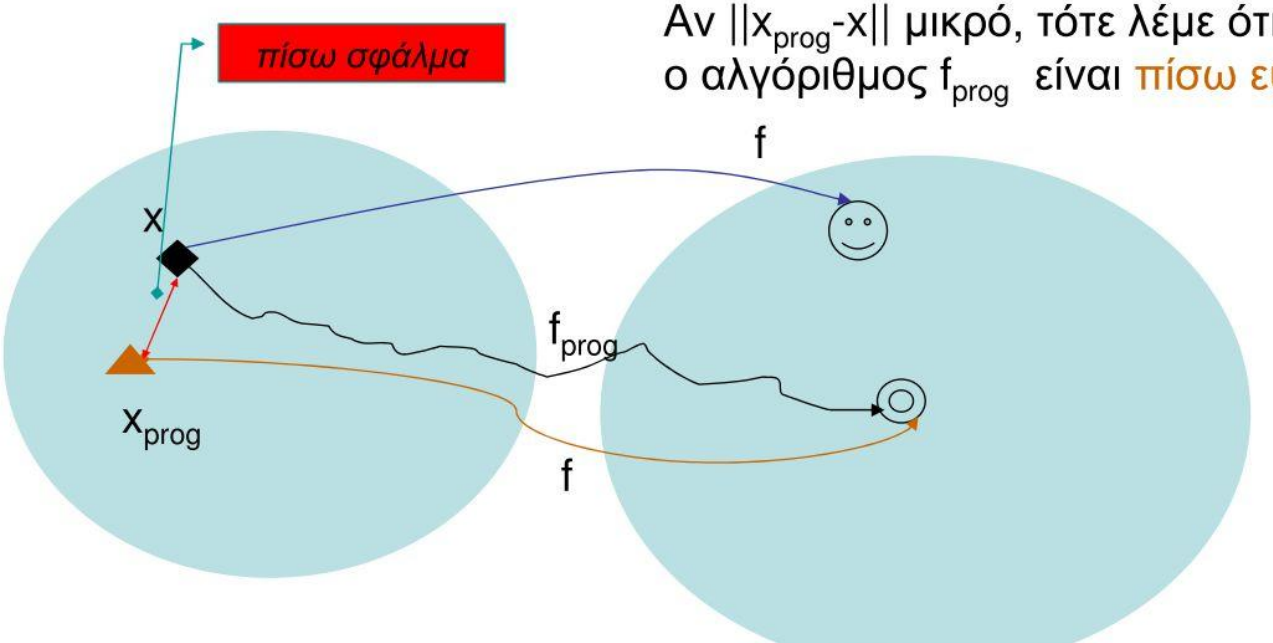
x_{prog}-x



Πόσο πρέπει να αλλάξει το x ώστε η f να υπολογίζει (ακριβώς) το f_{prog} (x)?

$$f(x+?) = f_{prog}(x)$$

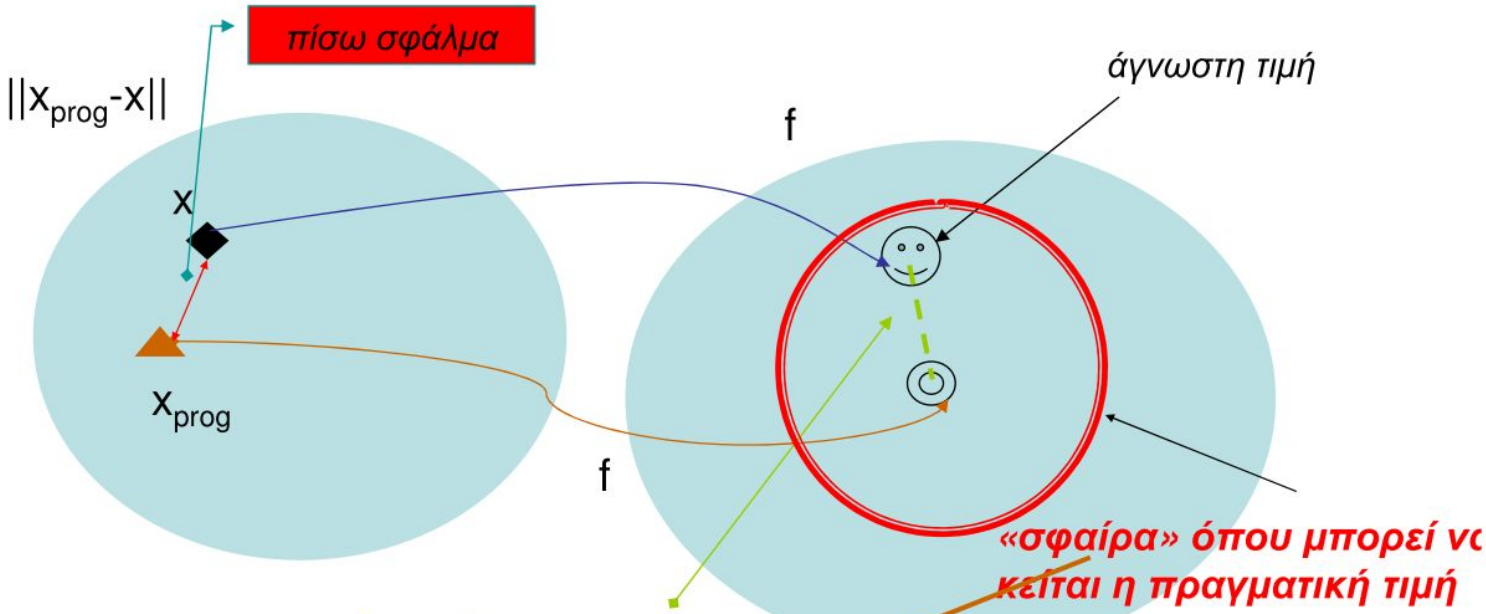
x_{prog}-x



Πόσο πρέπει να αλλάξει το x ώστε η f να υπολογίζει (ακριβώς) το f_{prog} (x)?

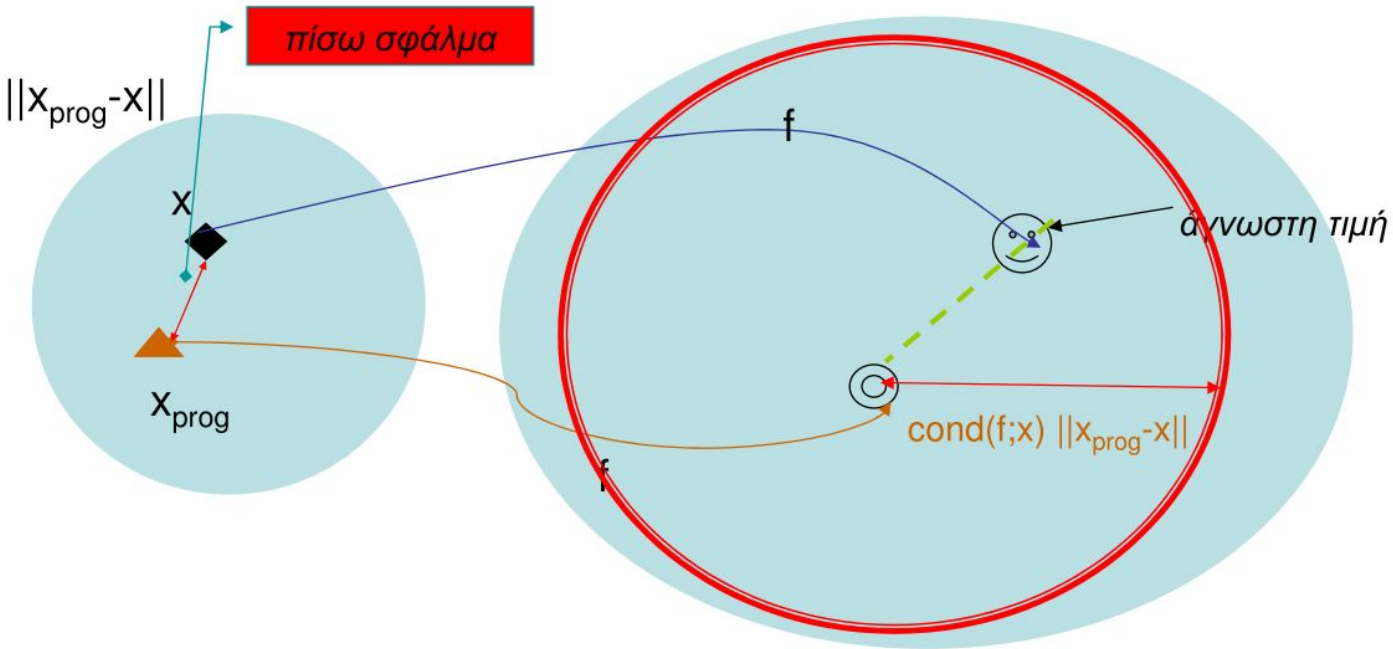
$$f(x+?) = f_{prog}(x)$$

x_{prog}-x

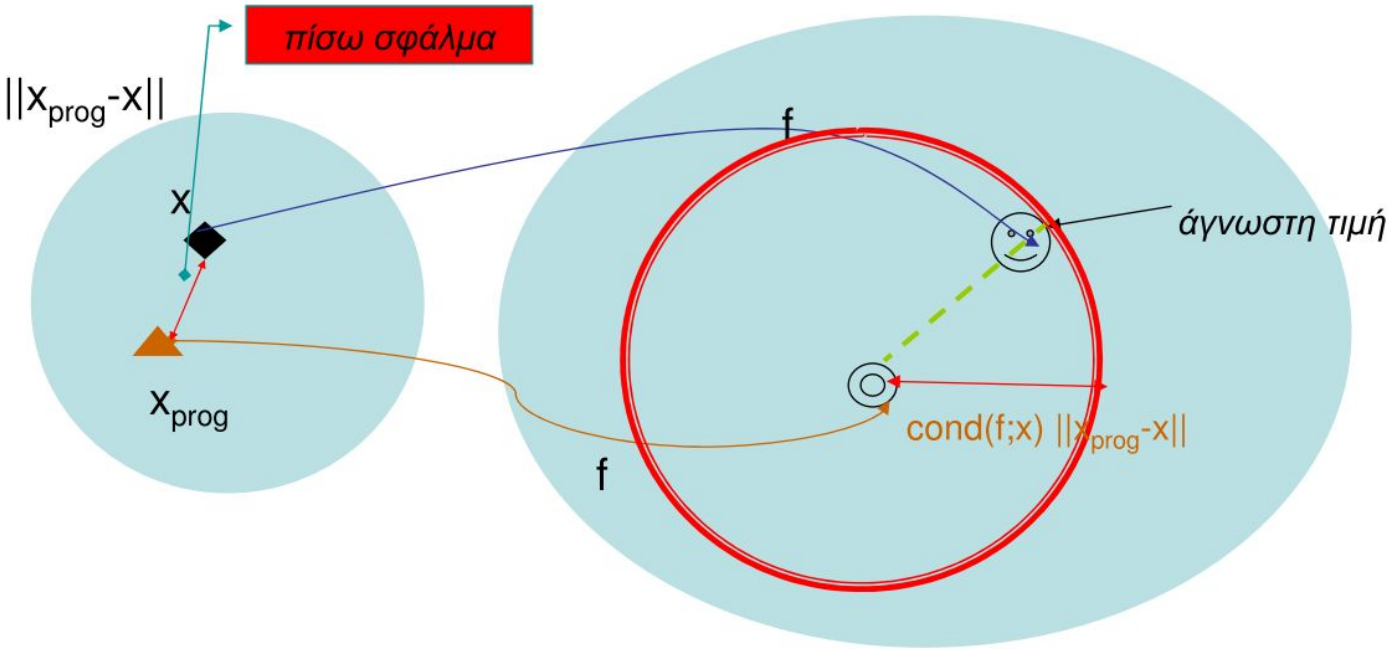


Η εκτίμηση του εμπρός σφάλματος γίνεται «μαθηματικά» βάσει
•του «δείκτη κατάστασης του προβλήματος» (δηλ. της f)
•χρησιμοποιώντας το πίσω σφάλμα.

$||f_{prog}(x) - f(x)|| \leq \mathbf{cond(f;x)}\ ||x_{prog}-x||$



Αν η συνάρτηση είναι πολύ ευαίσθητη, το cond(f;x) είναι μεγάλο (π.χ. αν $\frac{df}{dx}(x)$ μεγάλο) επομένως το f(x) μπορεί να απέχει πολύ από το $f(x_{prog})$



Για δεδομένο cond(f;x), η $||f(x_{prog}-f(x)||$ θα κυμαίνεται ανάλογα με το πίσω σφάλμα $||x_{prog}-x||$

Πίσω ανάλυση σφάλματος για DOT

Αν $s_n = x^\top y$ τότε

$$\begin{aligned}
\tilde{s}_1 &= \mathbf{fl}(\xi_1\psi_1) = \xi_1\psi_1(1+\delta_1) \\
\tilde{s}_2 &= \mathbf{fl}(\tilde{s}_1 + \mathbf{fl}(\xi_2\psi_2)) \\
&= (\xi_1\psi_1(1+\delta_1) + \xi_2\psi_2(1+\delta_2))(1+\delta_3) \\
&= \xi_1\psi_1(1+\delta_1)(1+\delta_3) + \xi_2\psi_2(1+\delta_2)(1+\delta_3)
\end{aligned}$$

όπου $|\delta_i| \leq u$.

Έχουμε

$$\tilde{s}_n = \xi_1\psi_1 \prod_{\substack{j=1 \\ j\neq 2}}^{n+1}(1+\delta_j) + \xi_2\psi_2\prod_{j=2}^{n+1}(1+\delta_j) +$$

$$\ldots\xi_3\psi_3\prod_{j=3}^{n+1}(1+\delta_j) + \cdots + \xi_n\psi_n\prod_{j=n}^{n+1}(1+\delta_j).$$

Από το Λήμμα:

$$\tilde{s}_n = \xi_1\psi_1(1+\theta_n) + \xi_2\psi_2(1+\hat{\theta}_n) +$$
$$\ldots\xi_3\psi_3(1+\theta_{n-1}) + \cdots + \xi_n\psi_n(1+\theta_2).$$

Το υπολογισμένο $\mathbf{DOT}$ *είναι το ακριβές εσωτερικό γινόμενο για τα στοιχεία* $\xi_1, \ldots, \xi_n, \psi_1(1+\theta_n), \psi_2(1+\hat{\theta}_n), \ldots, \psi_n(1+\theta_2)$, *όπου για τα* θ *ισχύει το φράγμα*

$$|\theta_j| \leq \frac{ju}{1-ju} = \gamma_j.$$

Αποδείξαμε ότι

$$\mathrm{fl}(x^\top y) = (x + \Delta x)^\top y = x^\top (y + \Delta y),$$
$$\text{όπου } |\Delta x| \le \gamma_n |x|, |\Delta y| \le \gamma_n |y|.$$

Δηλαδή το υπολογισμένο DOT είναι το ίδιο με το ακριβές DOT για στοιχεία εισόδου $x + \Delta x$, y. Επίσης

$$|\Delta x| \le \gamma_n |x| \Rightarrow \|\Delta x\|_1 \le \gamma_n \|x\|_1$$

άρα το σχετικό πίσω σφάλμα είναι φραγμένο ως εξής

$$\frac{\|\Delta x\|_1}{\|x\|_1} \le \gamma_n$$

και δείκτης κατάστασης του αλγορίθμου το $\mathbf{cond}(f_{\mathrm{prog}}) = \frac{\gamma_n}{\mathbf{u}}$.

ο παραπάνω αλγόριθμος υπολογισμού του DOT είναι προς τα πίσω ευσταθής

ΠΡΟΣΟΧΗ

1. Το ακριβές φράγμα εξαρτάται από τη σειρά υπολογισμού. Το παραπάνω φράγμα αντιστοιχεί στην άθροιση "από τα αριστερά προς τα δεξιά".
2. Στις συζητήσεις μας, οι τιμές των φραγμάτων είναι θεωρητικές. Για αξιόπιστες τιμές, πρέπει να ληφθεί υπόψη και το σφάλμα στον υπολογισμό του γ_n.
3. π.χ. αντικαθιστώντας το γ_n με $\mathbf{fl}(\gamma_{n+1}) \ge \gamma_n$.

Δείκτης κατάστασης της μαθηματικής πράξης **DOT**

$$f([x;y]) := \sum_{j=1}^{n} \xi_j \psi_j = x^\top y, \quad x, y \in \mathbb{R}^n$$

Θέτουμε $X := [x; y] \in \mathbb{R}^{2n}$ και υπολογίζουμε το δείκτη κατάστασης με βάση την ((ευαισθησία)) του αποτελέσματος ως προς κάθε στοιχειο:

$$\begin{aligned} K &= \frac{1}{|x^\top y|}\left[|\xi_1 \frac{\partial f}{\partial \xi_1}|, \ldots, |\xi_n \frac{\partial f}{\partial \xi_n}|, |\psi_1 \frac{\partial f}{\partial \psi_1}|, \ldots, |\psi_n \frac{\partial f}{\partial \psi_n}|\right] \\ &= \frac{1}{|x^\top y|}\left[|\xi_1\psi_1|, \ldots, |\xi_n\psi_n|, |\xi_1\psi_1|, \ldots, |\xi_n\psi_n|\right] \end{aligned}$$

επομένως

$$\|K\|_1 = \frac{2\sum_{j=1}^{n} |\xi_j||\psi_j|}{|\sum_{j=1}^{n} \xi_j\psi_j|}$$

Η κατάσταση εξαρτάται αρ΄ο το $|x^\top y|$, δηλ. από την τιμή $\cos(x, y)$. Υπάρχει πρόβλημα όταν $\|x\|, \|y\| \gg 0$ και $\cos(x, y) \approx 0$.

Προς τα εμπρός σφάλμα

Προς τα πίσω ανάλυση $\frac{|x^\top y - \mathrm{fl}(x^\top y)|}{|x^\top y|} \leq \gamma_n \frac{2\sum_{j=1}^n |\xi_j||\psi_j|}{|\sum_{j=1}^n \xi_j\psi_j|}$

Προς τα εμπρός ανάλυση Χρησιμοποιώντας προς τα εμπρός ανάλυση συμπεραίνουμε ότι

$$\begin{aligned} |x^\top y - \mathrm{fl}(x^\top y)| &\leq \gamma_n \sum_{i=1}^n |\xi_i||\psi_i| = \gamma_n |x|^\top |y| \\ &\leq nu|x|^\top |y| + O(u^2) \end{aligned}$$

Αν $|x^\top y| \ll |x|^\top |y|$ το άνω φράγμα μπορεί να είναι μεγάλο, επομένως δεν έχουμε καμμία εξασφάλιση όσο αφορά το μέγεθος του σχετικού σφάλματος.

Προσοχή: Όταν οι αντίστοιχες τιμές είναι ομόσημες, τότε το σχετικό σφάλμα είναι μικρό.

Μερικά εμπόδια

Η πίσω ανάλυση σφάλματος:

- Μπορεί να είναι δύσκολη
- Μπορεί να είναι ανέφικτη.

Διαστατικό επιχείρημα: Γενικά αν η διάσταση του διανύσματος εξόδου είναι μεγαλύτερη από τη διάσταση της εισόδου, τότε θα έχουμε δυσκολία να αποδείξουμε πίσω ευστάθεια.

Μερικά εμπόδια

Η πίσω ανάλυση σφάλματος:

- Μπορεί να είναι δύσκολη
- Μπορεί να είναι ανέφικτη.

Διαστατικό επιχείρημα: Γενικά αν η διάσταση του διανύσματος εξόδου είναι μεγαλύτερη από τη διάσταση της εισόδου, τότε θα έχουμε δυσκολία να αποδείξουμε πίσω ευστάθεια.
Παράδειγμα Ο υπολογισμός

$$\mathrm{fl}(C) = \mathrm{fl}(ab^\top),\ a, b \in \mathbb{R}^n$$

άρα

$$\mathrm{fl}(\gamma_{ij}) = \alpha_i\beta_j(1+\delta_{ij})$$

pou shma'inei 'oti

$$\begin{aligned} \mathrm{fl}(C) &= ab^\top + E \Rightarrow \\ |C - \mathrm{fl}(C)| &= |E| \leq \mathbf{u}ee^\top \end{aligned}$$

όπου $[E]_{ij} = \alpha_i\beta_j\delta_{ij}$ και $ee^\top$ είναι ίσο με το μητρώο με όλα τα στοιχεία του 1. Δηλ. το καλύτερο δυνατό άνω φράγμα για το εμπρός σφάλμα!

Για πίσω ευστάθεια θάπρεπε να υπάρχουν $\tilde{a}$, $\tilde{b}$ κοντά στα a, b τ.ώ.

$$\mathbf{fl}(C) = \tilde{a}\tilde{b}^\top = (a + \Delta a)(b + \Delta b)^\top$$

Τότε θα ίσχυε

$$E = \overbrace{a\Delta b^\top + \Delta a(b^\top + \Delta b^\top)}^{\text{τάξη} \leq 2}$$

Αν ο αλγόριθμος υπολογισμού εξωτερικού γινομένου ήταν πίσω ευσταθής θα ίσχυε ότι

$$n = \mathbf{rank}(E) \leq 2$$

πράγμα που είναι γενικά αδύνατο (δεν υπάρχουν αρκετά δεδομένα εισόδου στα οποία να ((αναθέσουμε)) τα σφάλματα στα αποτελέσματα)

Το εξωτερικό γινόμενο διανυσμάτων δεν είναι πίσω ευσταθής πράξη

ΠΡΟΣΟΧΗ: Αυτό δεν σημαίνει ότι ο αλγόριθμος υπολογισμού έχει πρόβλημα! Μόνον ότι δεν μπορούμε να εφαρμόσουμε πίσω ανάλυση σφάλματος για να φράξουμε το εμπρός σφάλμα

....

Παράδειγμα

Μέθοδος Horner

$s_n = \alpha_n$

for $k = n-1 : -1 : 0$

$s_k = x s_{k+1} + \alpha_k$

end

Χρησιμοποιώντας το γνωστό λήμμα:

$$
\begin{aligned}
\hat{s}_{n-1} &= (x s_n(1+\delta_1) + \alpha_{n-1})(1+\delta_2) = x\alpha_n(1+\theta_2) + \alpha_{n-1}(1+\delta_2) \\
\hat{s}_{n-2} &= (x\hat{s}_{n-1}(1+\delta_3) + \alpha_{n-2})(1+\delta_4) \\
\hat{s}_0 &= \alpha_0(1+\delta) + \alpha_1 x(1+\theta_3) + \cdots \alpha_{n-1}x^{n-1}(1+\theta_{2n-1}) + \alpha_n x^n(1+\theta_{2n}) \\
&= f_{\text{prog}}(\alpha_0, ..., \alpha_n, x) = f(\alpha_0(1+\theta_1), ..., \alpha_n(1+\theta_{2n}), x)
\end{aligned}
$$

για κάποια θ_j φραγμένα σε απόλυτη τιμή από τα αντίστοιχα γ_j.

Επομένως, το προς τα πίσω σφάλμα της μεθόδου Horner είναι μικρό.

Ειδικότερα:

> *Αφού θ_{2n} είναι ο παράγοντας που χαρακτηρίζει τη μέγιστη ασάφεια στους συντελεστές, είναι ασφαλές να συμπεράνουμε ότι αν κάθε συντελεστής δεν είναι ακριβώς γνωστός αλλά γνωρίζουμε ότι $|\alpha_j - \hat{\alpha}_j| \leq \theta_2 n|\alpha_j|$ για τον καθένα, τότε το σφάλμα που προκύπτει από την ασάφεια αυτή θα είναι σίγουρα μεγαλύτερο από οποιοδήποτε σφάλμα οφείλεται στις πράξεις α.κ.υ. που εκτελούνται από τον αλγόριθμο.*

Αυτό από μόνο του δεν είναι αρκετό για να εγγυηθεί το σφάλμα των αποτελεσμάτων αλλά μας επιτρέπει να εξετάσουμε το προς τα εμπρός σφάλμα ανεξάρτητα από τις λεπτομέρειες του αλγορίθμου.

Εύκολα αποδεικνύεται η σχέση

$$\frac{|p(x) - \hat{s}_0|}{|p(x)|} \leq \gamma_{2n} \frac{\sum_{k=0}^{n} |\alpha_k||x|^k}{|p(x)|},$$

από την οποία προκύπτει ότι δεν μπορούμε να εγγυηθούμε μικρό προς τα εμπρός σφάλμα.

Γιατί τόσο ενδιαφέρον για πολυώνυμα

Φαίνονται ((εύκολα)) αλλά επιπλέον:

- Τα πολυώνυμα είναι από τις πιο σημαντικά ((αλγεβρικά αντικείμενα)) και η ορθή χρήση τους είναι σημαντικό θέμα.
- Το πρόβλημα του υπολογισμού τιμών πολυωνύμου αυτό καθαυτό εμφανίζεται σε πολλές εφαρμογές.
- Ο αλγόριθμος αποτελεί υπολογιστικό πυρήνα για την εύρεση των ριζών.
- δυστυχώς, φαίνονται εύκολα αλλά είναι ((ύπουλα)); βλ. το άρθρο (Wil84)!

Εντολή Fused Multiply-Add (FMA) I

- Ορισμένοι επεξεργαστές περιέχουν εντολή FMA (Fused Multiply and Add).
- Πλεονέκτημα στο χρόνο εκτέλεσης: Εκτελεί την πράξη $z + x * y$ στον ίδιο χρόνο περίπου που απαιτεί μια απλή $x * y$ ή$x + y$.
- Πλεονέκτημα στην ακρίβεια: προκύπτει ένα μόνον σφάλμα στρογγύλευσης.

$$\mathrm{fl}(z + x * y) = (z + x * y)(1 + \delta), \quad |\delta| \leq \mathbf{u}.$$

αντί

$$\mathrm{fl}(z + \mathrm{fl}(xy)) = (z + xy(1 + \delta_1))(1 + \delta_2)$$

- Περιέχεται στο πρότυπο IEEE-754-2008.
- FMA3: (a,b,c) <- a + b*c; FMA4: (a,b,c,d) <- a + b*c
- Δείτε (M^+10), Wikipedia
- Αρχικά (1990) στο IBM RS-6000. Σήμερα στα Intel Haswell, AMD Piledriver, Bulldozer

Εντολή Fused Multiply-Add (FMA) II

https://software.intel.com/en-us/blogs/2011/06/13/haswell-new-instruction-descriptions-now-available/

C00 Τμήμα Μηχανικώ Google Calendar CEID Webmail :: K ΕΥΣΤΡΑΤΙΟΣ ΓΑΛ upatras eclass Ομάδες Google Ιδιώτες | Τράπεζ diary meteo.gr: Ο Κ

Floating Point Multiply Accumulate – Our floating-point multiply accumulate significantly increases peak flops and provides improved precision to further improve transcendental mathematics. They are broadly usable in high performance computing, professional quality imaging, and face detection. They operate on scalar, 128-bit packed single and double precision data types, and 256-bit packed single and double-precision data types. [These instructions were described previously, in the initial Intel® AVX specification].

	Double Precision Packed FP	Single Precision Packed FP	Double Precision Scalar FP	Single Precision Scalar FP
Fused Multiply-Add A = A x B + C C += A x B	VFMADD132PD VFMADD213PD VFMADD231PD _mm_fmadd_pd() _mm256_fmadd_pd()	VFMADD132PS VFMADD213PS VFMADD231PS _mm_fmadd_ps() _mm256_fmadd_ps()	VFMADD132SD VFMADD213SD VFMADD231SD _mm_fmadd_sd() _mm256_fmadd_sd()	VFMADD132SS, FMADD213SS VFMADD231SS _mm_fmadd_ss() _mm256_fmadd_ss()
Fused Multiply-Alternating Add/Subtract A = A x B + C | A = A x B – C C += A x B | C -= A x B	VFMADDSUB132PD VFMADDSUB213PD VFMADDSUB231PD _mm_fmaddsub_pd() _mm256_fmaddsub_pd( )	VFMADDSUB132PS VFMADDSUB213PS VFMADDSUB231PS _mm_fmaddsub_ps() _mm256_fmaddsub_ps()		
Fused Multiply-Alternating Subtract/Add A = A x B - C | A = A x B + C C -= A x B | C += A x B	VFMSUBADD132PD VFMSUBADD213PD VFMSUBADD231PD _mm_fmsubadd_pd() _mm256_fmsubadd_pd( )	VFMSUBADD132PS VFMSUBADD213PS VFMSUBADD231PS _mm_fmsubadd_pd() _mm256_fmsubadd_pd()		
Fused Multiply-Subtract A = A x B - C C -= A x B	VFMSUB132PD VFMSUB213PD VFMSUB231PD _mm_fmsub_pd() _mm256_fmsub_pd()	VFMSUB132PS VFMSUB213PS VFMSUB231PS _mm_fmsub_ps() _mm256_fmsub_ps()	VFMSUB132SD VFMSUB213SD VFMSUB231SD _mm_fmsub_sd() _mm256_fmsub_sd()	VFMSUB132SS VFMSUB213SS VFMSUB231SS _mm_fmsub_ss() _mm256_fmsub_ss()
Fused Negative Multiply-Add A = -A x B + C C += -A x B	VFNMADD132PD VFNMADD213PD VFNMADD231PD _mm_fnmadd_pd() _mm256_fnmadd_pd()	VFNMADD132PS VFNMADD213PS VFNMADD231PS _mm_fnmadd_ps() _mm256_fnmadd_ps()	VFNMADD132SD VFNMADD213SD VFNMADD231SD _mm_fnmadd_sd() _mm256_fnmadd_sd()	VFNMADD132SS VFNMADD213SS VFNMADD231SS _mm_fnmadd_ss() _mm256_fnmadd_ss()
Fused Negative Multiply-Subtract A = -A x B - C C = -A x B	VFNMSUB132PD VFNMSUB213PD VFNMSUB231PD _mm_fnmsub_pd() _mm256_fnmsub_pd()	VFNMSUB132PS VFNMSUB213PS VFNMSUB231PS _mm_fnmsub_ps() _mm256_fnmsub_ps()	VFNMSUB132SD VFNMSUB213SD VFNMSUB231SD _mm_fnmsub_sd() _mm256_fnmsub_sd()	VFNMSUB132SS VFNMSUB213SS VFNMSUB231SS _mm_fnmsub_ss() _mm256_fnmsub_ss()

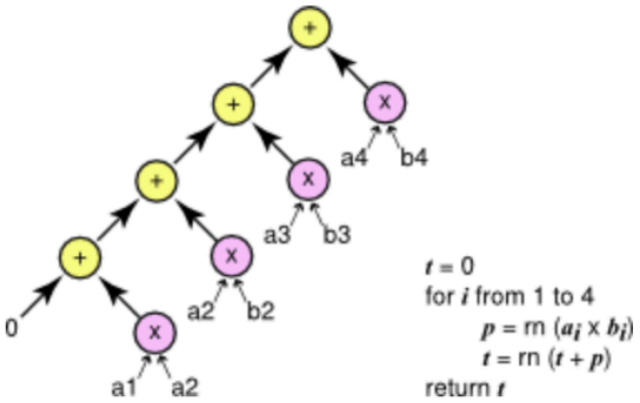
The vector instructions build upon the expanded (256-bit) register state added in Intel® AVX, and as such as supported by any operating system that supports Intel® AVX.
For developers, please note that the instructions span multiple CPUID leaves. You should be careful to check all applicable bits before using these instructions.
Please check out the specification and stay tuned for supporting tools over the next couple of months.

Mark Buxton
Software Engineer
Intel Corporation


NVIDIA
PRECISION AND PERFORMANCE:
FLOATING POINT AND IEEE 754
COMPLIANCE FOR NVIDIA GPUS
TB-06711-001_v5.5 | July 2013
White paper

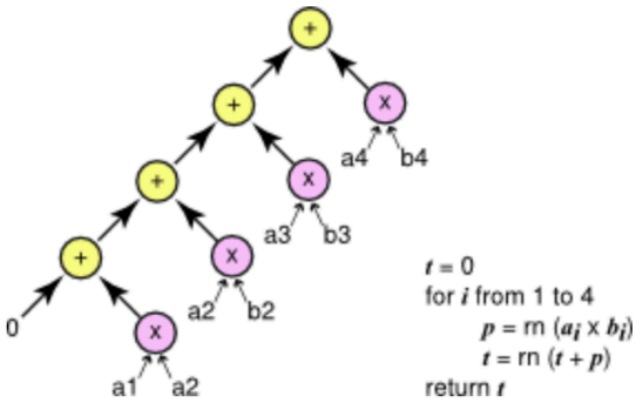
Εσωτερικό γινόμενο

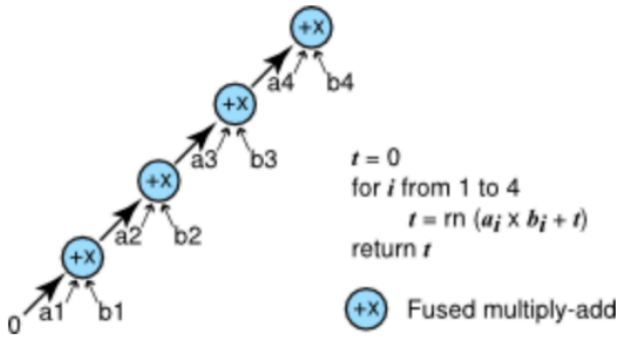
```
a = (1.907607, -.7862027, 1.148311, .9604002);   b = ...
    (-.9355000, -.6915108, 1.724470, -.7097529)
```



Εσωτερικό γινόμενο

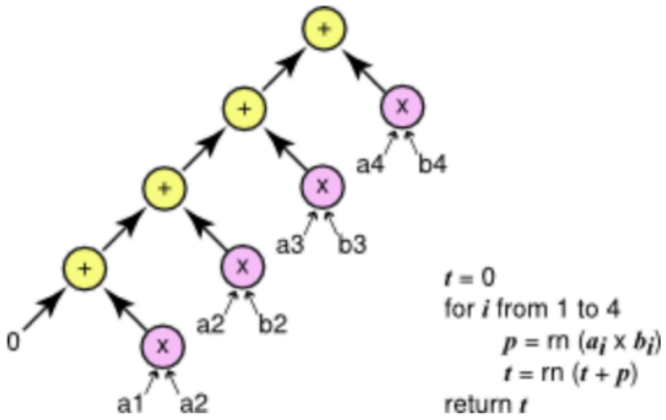
```
a = (1.907607, -.7862027, 1.148311, .9604002); b = ...
    (-.9355000, -.6915108, 1.724470, -.7097529)
```



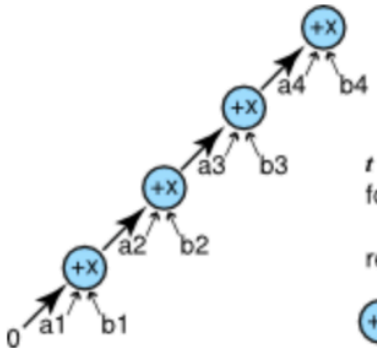


Εσωτερικό γινόμενο

```
a = (1.907607, -.7862027, 1.148311, .9604002);   b = ...
    (-.9355000, -.6915108, 1.724470, -.7097529)
```

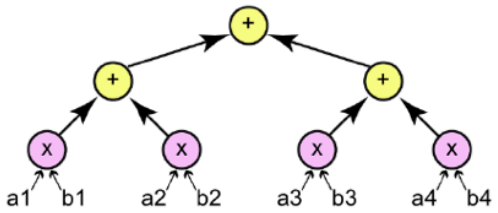


$t = 0$
for i from 1 to 4
$p = \text{rn}(a_i \times b_i)$
$t = \text{rn}(t + p)$
return t



$t = 0$
for i from 1 to 4
$t = \text{rn}(a_i \times b_i + t)$
return t

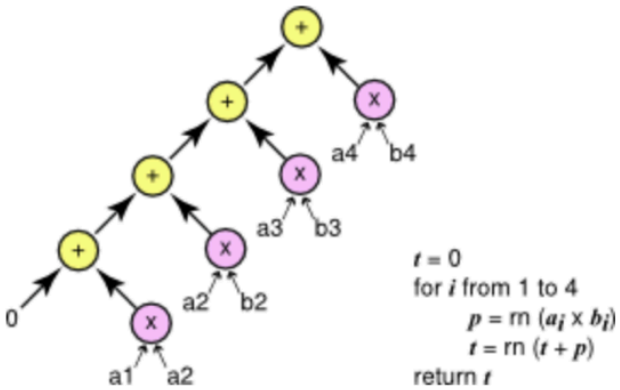
+x Fused multiply-add

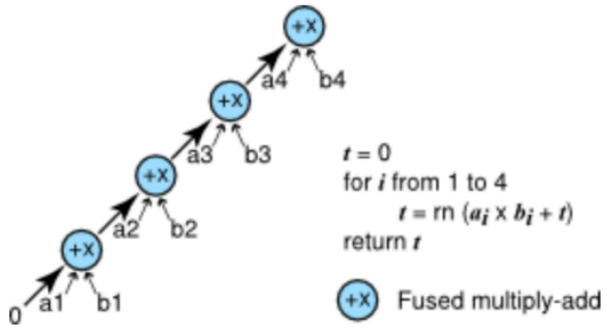


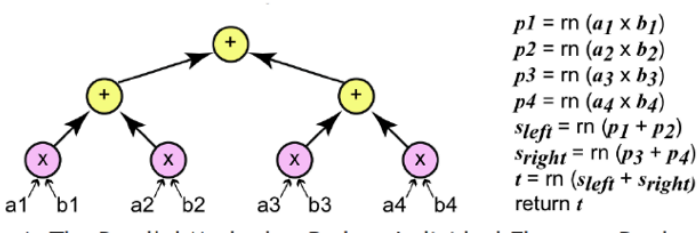
$p1 = \text{rn}(a_1 \times b_1)$
$p2 = \text{rn}(a_2 \times b_2)$
$p3 = \text{rn}(a_3 \times b_3)$
$p4 = \text{rn}(a_4 \times b_4)$
$s_{left} = \text{rn}(p_1 + p_2)$
$s_{right} = \text{rn}(p_3 + p_4)$
$t = \text{rn}(s_{left} + s_{right})$
return t

Εσωτερικό γινόμενο

```
a = (1.907607, -.7862027, 1.148311, .9604002);   b = ...
    (-.9355000, -.6915108, 1.724470, -.7097529)
```





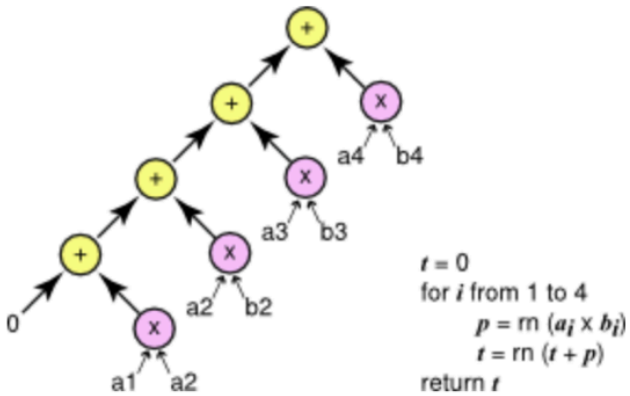


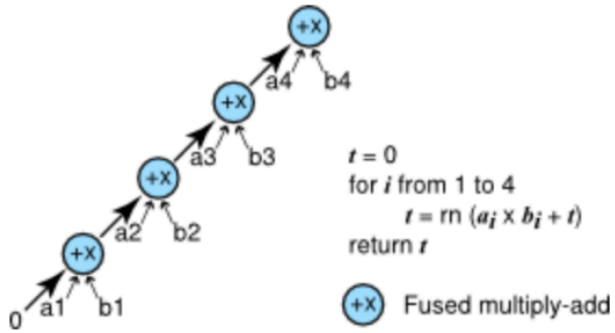
method	result	float value
exact	.0559587528435...	0x3D65350158...
serial	.0559588074	0x3D653510
FMA	.0559587515	0x3D653501
parallel	.0559587478	0x3D653500

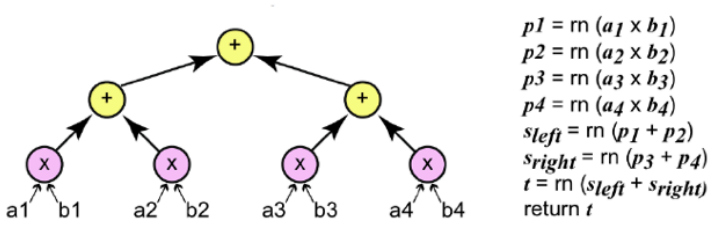
Figure 5 Algorithms Results vs. the Correct Mathematical Dot Product

Εσωτερικό γινόμενο

```
a = (1.907607, -.7862027, 1.148311, .9604002);   b = ...
    (-.9355000, -.6915108, 1.724470, -.7097529)
```







method	result	float value
exact	.0559587528435...	0x3D65350158...
serial	.0559588074	0x3D653510
FMA	.0559587515	0x3D653501
parallel	.0559587478	0x3D653500

Figure 5 Algorithms Results vs. the Correct Mathematical Dot Product

Το πρόβλημα της άθροισης (Higham'02)

Δίδεται σύνολο αριθμών $\mathcal{S} = \{\xi_1, ..., \xi_n\}$ και ζητούμε το άθροισμα $\mathbf{sum}(\mathcal{S})$.

- Μια από τις πιο συχνές πράξεις σε επιστημονικούς και εμπορικούς υπολογισμούς
- π.χ. συνοπτικά στατιστικά χαρακτηριστικά, π.χ. μέσος όρος, απόκλιση, νόρμα, ...
- Αξίζει τον κόπο μια πιο προσεκτική θεώρηση
- ... σε σχέση με την *ταχύτητα* και την *ακρίβεια*

Αναδρομική άθροιση

$s = 0$

for $i = 1 : n$

$s = s + \xi_i$

end

Σχετικά με την αναδρομική άθροιση

- Πολύ απλή υλοποίηση
- Ιδιαίτερα χρήσιμη για ομόσημα στοιχεία μετά από ταξινόμηση σε αύξουσα σειρά κατ' απόλυτη τιμή
- Αποφυγή προβλημάτων ((απορρόφησης)) από μεγάλα στοιχεία,
- ... πρέπει να είναι γνωστά τα στοιχεία και να ταξινομήσουμε πριν την άθροιση.

Πίσω σφάλμα άθροισης

Εύκολα βλέπουμε ότι

$$\begin{aligned} \mathrm{fl}(s) &= (\cdots((\xi_1+\xi_2)(1+\delta_1)+\xi_3)(1+\delta_2)\cdots+\xi_n)(1+\delta_{n-1}) \\ &= \xi_1(1+\theta_{n-1})+\xi_2(1+\hat{\theta}_{n-1})+\xi_3(1+\hat{\theta}_{n-2})+\cdots+\xi_n(1+\theta_1) \end{aligned}$$

όπου ως συνήθως

$$|\theta_j| \leq \gamma_j = \frac{j\mathbf{u}}{1-j\mathbf{u}}$$

Επομένως, το πίσω σφάλμα εξαρτάται άμεσα από το n, όπως θα περιμέναμε...

Μερικές ακόμα μέθοδοι

- Αναγωγική άθροιση ανά ζεύγη[2] πολύ χρήσιμη για παράλληλη άθροιση
- Ενθετική άθροιση
- Αντισταθμισμένη άθροιση και νεώτερες παραλλαγές

Χρήσιμα χαρακτηριστικά

- είναι οι αριθμοί ομόσημοι?
- είναι η ακολουθία διαθέσιμη από την αρχή?
- ποιό είναι το εύρος (μέγιστο, ελάχιστο)?
- είναι η ακολουθία διατεταγμένη/ταξινομημένη?

[2] paiwise / cascade / fan-in

Ενθετική άθροιση

1. ταξινόμηση του $\mathcal{S}$ κατ' απόλυτη τιμή σε αύξουσα σειρά
   $\mathcal{L} := \xi_1 \leq \xi_2 \leq \xi_3 \leq \cdots \leq \xi_n$
2. Διαγραφή του $\xi_1 + \xi_2$ απο την $\mathcal{L}$ και ένθεση στην $\mathcal{L}$ στην κατάλληλη θέση ώστε να παραμείνει μονοτονική. Επαναρίθμηση του $\mathcal{L}$ και αν περιέχει 2 ή περισσότερα στοιχεία, επιστροφή στο (2).
3. Η σειρά των προσθέσεων δημιουργεί ένα δένδρο με φύλλα τους αρχικούς αριθμούς $\{\xi_1, \ldots, \xi_n\}$ υπό πρόσθεση και ρίζα την άθροιση από την οποία προκύπτει το τελικό άθροισμα. Οι υπόλοιπο κόμβοι αντιστοιχούν στις ενδιάμεσες αθροίσεις.

Ενδιαφέρον Θυμίζει την κωδικοποίηση Huffman.

(KW00)

Αν τα δεδομένα έχουν μικτό πρόσημο (θετικά και αρνητικά στοιχεία) η εύρεση του αθροίσματος που ελαχιστοποιεί το μέγιστο σφάλμα στρογγύλευσης για όλους τους δυνατούς τρόπους άθροισης είναι NP-hard.

Συστάσεις

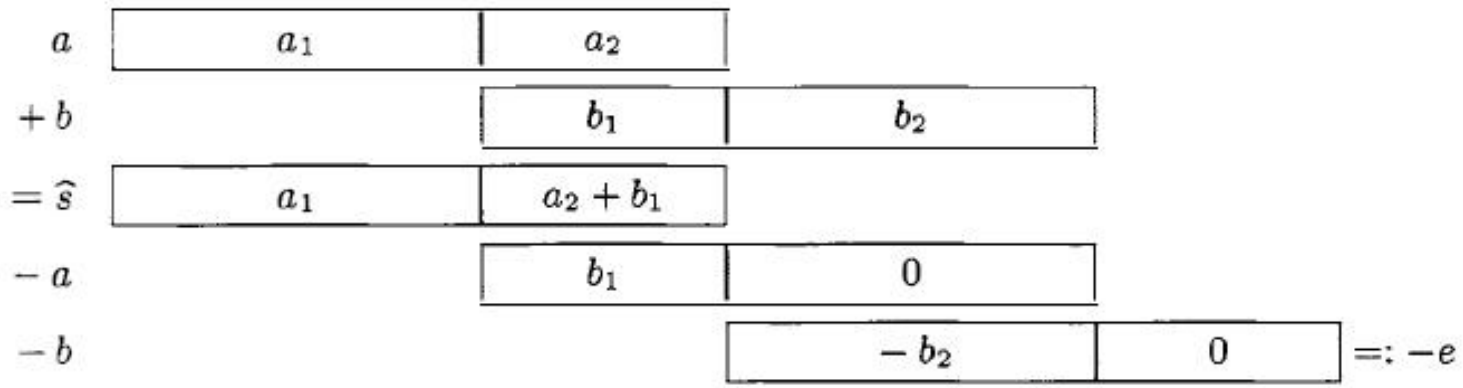
Εφόσον το πρόβλημα κατασκευής του βέλτιστου δένδρου άθροισης είναι πολύ δύσκολο:

- *χαλαρώνουμε* τις απαιτήσεις και αναζητούμε κάτι λιγότερο φιλόδοξο,
- *αξιοποιούμε* ό,τι πληροφορία υπάρχει για το $\mathcal{S}$.

- Αν χρειάζεται μεγάλη ακρίβεια, σκεφτείτε τη χρήση αντισταθμισμένης άθροισης ή άθροιση με εκτεταμένη ακρίβεια
- Για τις περισσότερες μεθόδους, το σφάλμα είναι στη χειρότερη περίπτωση ανάλογο με το n. Για πολύ μεγάλο n καθίσταται ενδιαφέρουσα η χρήση της αντισταθμισμένης άθροισης ή της αναγωγικής άθροισης
- Αν υπάρχει πιθανότητα σημαντικής (ως καταστροφικής) απαλοιφής λόγω μεικτών προσήμων, η αναδρομική άθροιση με φθίνουσα διάταξη είναι συνήθως καλύτερη.
- Αν το σύνολο ομόσημο, όλες οι μέθοδοι προσφέρουν σχετικό σφάλμα το πολύ $n\mathbf{u}$ με καλύτερη την επανορθωμένη άθροιση[3] (compensated summation).

[3] Στις επόμενες διαφάνειες.

Ακριβής άθροιση $\longrightarrow$ επανορθωμένη άθροιση



Σχήμα: Ιδιοφυής ιδέα (Gill'51, Kahan'65) από (Higham'02)

Για κάθε α.κ.υ. $|a| \geq |b|$ μπορούμε να υπολογίσουμε

$$\hat{s} = \mathrm{fl}(a + b), \quad \hat{e} = \mathrm{fl}((a - \hat{s}) + b)$$

χρησιμοποιώντας στρογγύλευση προς το πλησιέστερο. Το υπολογισμένο $\hat{e}$ είναι το ακριβές σφάλμα τής άθροισης! Ειδικότερα,

$$a + b = \hat{s} + \hat{e}$$

Παρόλα αυτά, $\mathrm{fl}(\hat{s} + \hat{e}) = \hat{s}$ γιατί το αποτέλεσμα $\hat{s} = \mathrm{fl}(a + b)$ είναι το καλύτερο δυνατό (λόγω αρχής ακριβούς στρογγύλευσης)

Σχόλια

Ιδέα Να το κρατήσουμε για την επόμενη πρόσθεση (αν υπάρχει). Αν πρέπει να προσθέσουμε και άλλον αριθμό, c, τότε (ίσως) μπορούμε να ((επανορθώσουμε)) με το άθροισμα των κρατούμενων.
Πλεονεκτήματα επανορθωμένης άθροισης Γενικά είναι πιο ακριβής από την συνηθισμένη άθροιση.

Εμπρός σφάλμα (Knuth, Kahan)

φράσσεται ως

$$|\sum_{j=1}^{n}\xi_j - \mathrm{fl}\left(\sum_{j=1}^{n}\xi_j\right)| \leq (2\mathbf{u} + O(n\mathbf{u}^2))\sum_{j=1}^{n}|\xi_j|.$$

Πίσω σφάλμα

$$\mathrm{fl}(s) = \sum_{j=1}^{n}(1+\mu_j)\xi_j, \quad \text{όπου} \quad |\mu_j| \leq 2\mathbf{u} + O(n\mathbf{u}^2)$$

δηλ. όσον αφορά στους όρους 1ης τάξης (που εξαρτώνται από το $\mathbf{u}$) το πίσω σφάλμα είναι ανεξάρτητο του n.

Επανορθωμένη αριθμητική?

A Floating-Point Technique for Extending the Available Precision

T. J. DEKKER*

Received July 26, 1971

Abstract. A technique is described for expressing multilength floating-point arithmetic in terms of singlelength floating point arithmetic, i.e. the arithmetic for an available (say: single or double precision) floating-point number system. The basic algorithms are exact addition and multiplication of two singlelength floating-point numbers, delivering the result as a doublelength floating-point number. A straightforward application of the technique yields a set of algorithms for doublelength arithmetic which are given as ALGOL 60 procedures.

Let x and y be singlelength floating-point numbers and let

$$z = fl(x+y);$$

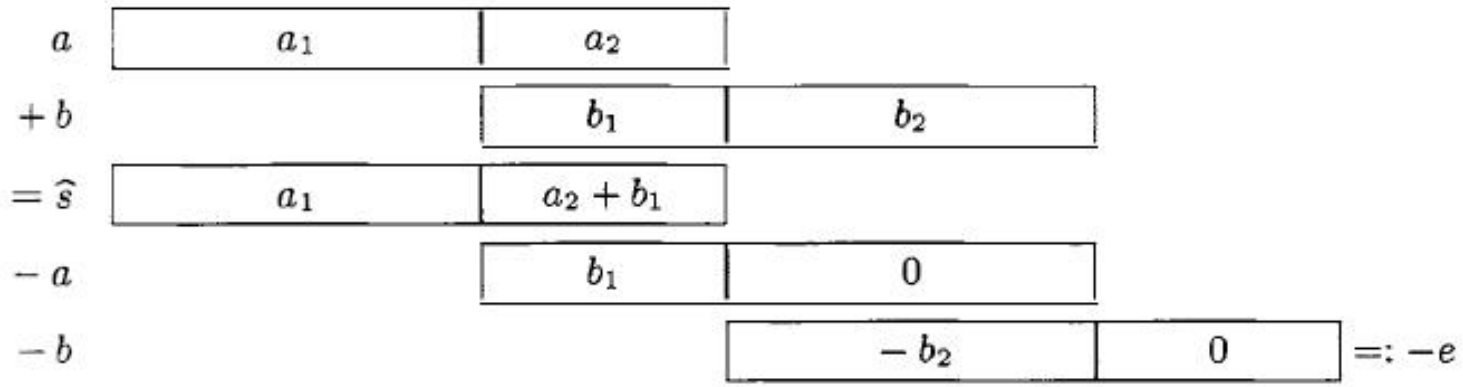
i.e. z is the result of a singlelength floating-point addition of x and y. Let zz be the correction term exactly satisfying

$$z+zz=x+y.$$

It will be shown that, under various conditions, zz can be obtained by the formula

$$zz=fl((x-z)+y).$$

Ακριβής άθροιση $\longrightarrow$ επανορθωμένη άθροιση



Σχήμα: Ιδιοφυής ιδέα (Gill'51, Kahan'65) από (Higham'02)

Για κάθε α.κ.υ. $|a| \geq |b|$ μπορούμε να υπολογίσουμε

$$\hat{s} = \mathrm{fl}(a + b), \quad \hat{e} = \mathrm{fl}((a - \hat{s}) + b)$$

χρησιμοποιώντας στρογγύλευση προς το πλησιέστερο. Το υπολογισμένο $\hat{e}$ είναι το ακριβές σφάλμα τής άθροισης! Ειδικότερα,

$$a + b = \hat{s} + \hat{e}$$

Παρόλα αυτά, $\mathrm{fl}(\hat{s} + \hat{e}) = \hat{s}$ γιατί το αποτέλεσμα $\hat{s} = \mathrm{fl}(a + b)$ είναι το καλύτερο δυνατό (λόγω αρχής ακριβούς στρογγύλευσης)

Σχόλια

Ιδέα Να το κρατήσουμε για την επόμενη πρόσθεση (αν υπάρχει). Αν πρέπει να προσθέσουμε και άλλον αριθμό, c, τότε (ίσως) μπορούμε να ((επανορθώσουμε)) με το άθροισμα των κρατούμενων.
Πλεονεκτήματα επανορθωμένης άθροισης Γενικά είναι πιο ακριβής από την συνηθισμένη άθροιση.

Εμπρός σφάλμα (Knuth, Kahan)

φράσσεται ως

$$|\sum_{j=1}^{n}\xi_j - \mathrm{fl}\left(\sum_{j=1}^{n}\xi_j\right)| \leq (2\mathbf{u} + O(n\mathbf{u}^2))\sum_{j=1}^{n}|\xi_j|.$$

Πίσω σφάλμα

$$\mathrm{fl}(s) = \sum_{j=1}^{n}(1+\mu_j)\xi_j, \quad \text{όπου} \quad |\mu_j| \leq 2\mathbf{u} + O(n\mathbf{u}^2)$$

δηλ. όσον αφορά στους όρους 1ης τάξης (που εξαρτώνται από το $\mathbf{u}$) το πίσω σφάλμα είναι ανεξάρτητο του n.

Εφαρμογή

Θα δοκιμάσουμε βήμα-βήμα την επανόρθωση στον γνωστό υπολογισμό

$$((10^{20} - 10) - 10^{20}) + 20$$

```
a1=10^20;  b1=-10; s1  =  a1+b1;
e1=-((a1-s1)+b1);  % επιστρέφει   10
a1=-10^20;  b1=s1; s2=a1+b1;
e2=-((a1-s2)+b1);  % επιστρέφει  0
a1=s2;  b1=20;  s3  =  a1+b1;
e3=-((a1-s2)+b1);  επιστρέφει  20
s  =  s3+e1+e2+e3
   =  10
```

- Η επανόρθωση "διόρθωσε" το αποτέλεσμα και το "επανέφερε" στην ακριβή του τιμή.
- Πρόκειται για το best case scenario, δεν συμβαίνει πάντα!!!

Listing 4: Επανόρθωση στην άθροιση 2 αριθμών

```
function [s,e] = kahan_cs(a,b);
if (abs(a)< abs(b))
   temp =a; a=b; b=temp;
end
s = a+b;
e = (a-s)+b;
```

Listing 5: Επανόρθωση χωρίς ελέγχους στην άθροιση 2 αριθμών

```
function [s,e] = nocomp2sum(x,y);
% following Muller et al.
s = a+b;
a_dot = s-b; b_dot = s-a_dot;
da = a-a_dot; db = b-b_dot;
e = da + db;
```

Άθροιση $n \geq 3$ στοιχείων I

Listing 6: Επανόρθωση στην άθροιση

```
function [s] = casc_sum(x);
s = x(1);
e = 0;
for i=2:length(x);
    [s,c(i)] = kahan_cs(s,x(i));
    e = e + c(i);
end
s = s+e;
```

Παρατήρηση: Εναλλακτικά μπορούμε να εφαρμόσουμε αναδρομικά τον ίδιο αλγόριθμο για την άθροιση των στοιχείων του `C`.

ΠΑΡΑΔΕΙΓΜΑ: (όπως πριν)

x = [1e20, -10, -1e20, 20]

ΑΠ	sum(x)	casc_sum(x)
10	20	10

Άθροιση $n \geq 3$ στοιχείων II

Listing 7: Επανόρθωση στην άθροιση

```
function [s] = casc_sum(x);
s = x(1);
e = 0;
for i=2:length(x);
    [s,c(i)] = kahan_cs(s,x(i));
    e = e + c(i);
end
s = s+e;
```

ΠΑΡΑΔΕΙΓΜΑ: `x = 1/n * ones(n,1)` Θεωρητικά `sum(x) = 1`

x = 1/n*ones(n,1)

Παράδειγμα

Έστω `x = 1/n * ones(n,1)` Θεωρητικά πρέπει να ισχύει ότι `sum(x) = 1`

`n`	`sum(x)`	`casc_sum(x)`	`abs(1-sum(x))`
10	9.999999999999999e-01	1	1.1102e-016
100	1.000000000000001e+00	1	6.6613e-016
10^4	9.999999999980838e-01	1	1.9162e-012

Ενθετική άθροιση

1. Διάταξη $\mathcal{S}$ σε αύξουσα απόλυτη τιμή $\mathcal{L} := \xi_1 \leq \xi_2 \leq \xi_3 \leq \cdots \leq \xi_n$
2. Διαγραφή των ξ_1, ξ_2 από το σύνολο $\mathcal{L}$ και ένθεση του $\xi_1 + \xi_2$ στο $\mathcal{L}$ στη σωστή θέση ώστε η ακολουθία να παραμείνει διατεταγμένη κατ' αύξουσα τιμή.
3. Επαναρίθμηση των στοιχείων του $\mathcal{L}$ και επιστροφή στο (2) αν έχει τουλάχιστον 3 στοιχεία.

Summation algorithms can be represented as binary tree $\mathcal{T}$, where the leaves are the inputs, the root the output and the internal nodes are the partial sums computed in the course of the algorithm.

Ενδιαφέρον - "χαζή" παρέμβαση "έξυπνου" μεταφραστή

Δείτε την τελευταία παράγραφο:

matteo-frigo commented on Mar 31, 2014

Just to make it more interesting, here are a couple of issues with FMA, in addition to those pointed out by stevengj.

First, the FMA implemented by PowerPC and by Intel differ in (at least) one subtle way. PowerPC computes +/-(a*b* +/- *c), whereas Intel implements* +/-(a*b*) +/- c. The two differ when the result is -0; for example, -(+0+*0* - +*0) = -0, whereas +0* - +*0*+0 = +0. So a pedantic (I dare not say correct) compiler cannot translate the PowerPC expression into the x86 expression.

Second, common-subexpression elimination makes things interesting. For example, FFTs tend to compute pairs of expressions of the form a +/- b*c, which reduces to two fma's. However, a sufficiently smart compiler will tend to compute t=b*c; a +/- t, which is three flops. Historically in fftw, we were able to defeat gcc-3.x with some hackish workaround, but gcc-4.[0-2] had a more formidable optimizer that was always isolating the common subexpression no matter what. I haven't tried more recent gcc's or llvm in this respect, but that's something to watch for.

Σφάλματα και επανόρθωση στον πολλαπλασιασμο? I

- Οι παραπάνω ιδέες μπορούν να επεκταθούν και στον πολλαπλασιασμό α.κ.υ.
- Δείτε το άρθρο του Dekker που αναρτήθηκε νωρίτερα και θυμηθείτε και τη συζήτηση στην τάξη.
- Επιγραμματικά μόνον αναφέρουμε ορισμένες από τις ιδέες.
- Περισσότερα στις αναφορές.

Ερώτημα Μπορούμε να υπολογίσουμε το ακριβές σφάλμα στον πολλαπλασιασμό α.κ.υ.?

Αναζητούμε τα εξής: Δοθέντων 2 α.κ.υ. x, y να υπολογίσουμε το γινόμενο σε α.κ.υ. και α.κ.υ. b τ.ω. $xy = \mathrm{fl}(xy) + b$ (σε αριθμητική άπειρης ακρίβειας).

Απαιτείται πρώτα να διαχωρίσουμε την είσοδο σε 2 μέρη: Αν $eps = 2^{-p}$ τότε $s = \lceil p/2 \rceil$. Για παράδειγμα αν $p = 53$ τότε $s = 27$. Ο αλγόριθμος του Dekker διαχωρίζει τον α.κ.υ. a σε δύο τμήματα x, y ώστε $a = x_h + x_l$ αλλά $|x_l| \leq |x_h|$ και τα x_h, x_l δεν έχουν επικάλυψη.

Σφάλματα και επανόρθωση στον πολλαπλασιασμο? II

Listing 8: Διαχωρισμός του Veltkamp

```
function [x_h,x_l]=SplitV(x,s)
% split precision-p radix-B fp into two floats x_h, x_l
% so that for given s<p, the sifgnificand of x_h fits
% in p-s digits and the significand of x_l fits into s digits
% and x=x_h+x_l exactly
F = B^s+1;
g = F*x;
d = x-g;
x_h = g + d; x_l = x-x_h;
```

Παράδειγμα (M^+10)

Για βάση 10, ακρίβεια $p = 8$ ψηφία, διαχωρισμός σε 2 τμήματα $s = 4$ ψηφίων.

Καλούμε $\mathrm{SplitV}(1.2345678,4)$

$\mathrm{F} = 10^s{+}1; \Rightarrow \mathrm{F}{=}10001$

$\mathrm{g} = \mathrm{F}{*}\mathrm{x}; \Rightarrow \mathrm{g}{=}12346.9125678 \Rightarrow \mathrm{g}{=}12346.913$

$\mathrm{d} = \mathrm{x}\text{-}\mathrm{g}; \Rightarrow \mathrm{x}\text{-}\mathrm{g}{=}\text{-}12345.6784322 \Rightarrow \mathrm{d}{=}\text{-}12345.678$

$\mathrm{x\_h} = \mathrm{g} + \mathrm{d}; \Rightarrow \mathrm{g}{+}\mathrm{d}{=}1.235 \Rightarrow \mathrm{x\_h}{=}1.235$

$\mathrm{x\_l} = \mathrm{x}\text{-}\mathrm{x\_h}; \Rightarrow \mathrm{x}\text{-}\mathrm{x\_h}{=}\text{ -}0.0004322 \Rightarrow \mathrm{x\_l}{=}\text{-}0.0004322$

Προσέξτε ισχύει ακριβώς ότι

$\mathrm{x\_h}{+}\mathrm{x\_l} = 1.235 \text{ - } 0.0004322 = 1.2345678$


W. Gautschi.

Numerical Analysis: An Introduction.

Birkhauser, Boston, 1997.


N.J. Higham.

Accuracy and Stability of Numerical Algorithms.

SIAM, Philadelphia, 2nd edition, 2002.


S I Kabanikhin.

Definitions and examples of inverse and ill-posed problems.

J. Inverse and Ill-posed Problems, 16(4), 2008.


Ming-Yang Kao and Jie Wang.

Linear-time approximation algorithms for computing numerical summation with provably small errors.

SIAM J. Comput., 29(5):1568–1576, 2000.


A.N. Langville.

Catastrophic cancellation on the high seas.

The Pi Mu Epsilon Journal, 11(4):205–208, 2001.

http://www4.ncsu.edu/~ipsen/ma798I/langville.pdf.


J.-M. Muller et al.

Handbook of Floating-Point Arithmetic.

Birkhäuser Boston, 2010.


A. Quarteroni, R. Sacco, and F. Saleri.

Numerical Mathematics.

Springer, New York, 2000.


J.H. Wilkinson.

The perfidious polynomial.

In G.H. Golub, editor, *Studies in Numerical Analysis*, volume 24, pages 1–28. Mathematical Association of America, 1984.