ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά μαθήματα ΠΠ

Επιστημονικός Υπολογισμός Ι

Ενότητα **4** : Μοντέλο Αριθμητικής και Σφάλματα Υπολογισμού

Ευστράτιος Γαλλόπουλος
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο **«Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών»** έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σκοπός Ενότητας

- Απώλεια πληροφορίας στον επιστημονικό υπολογισμό.
- Αριθμητικό μοντέλο και πρότυπο αριθμητικής κινητής υποδιαστολής IEEE.
- Σφάλματα στρογγύλευσης και διάδοσή τους.
- Σφάλματα στρογγύλευσης και διάδοσή τους.
- Δείκτες κατάστασης προβλήματος και αλγορίθμου.
- Θεωρία και εργαλεία εκτίμησης σφάλματος και ποιότητας υπολογισμών.

Περιεχόμενα

1. Μελέτη περίπτωσης: υπολογισμός ευκλείδειας νόρμας (συνέχεια)

2. Πρότυπο IEEE-754 (γρήγορη επισκόπηση/υπενθύμιση)
   - Υποκανονικοποιημένοι αριθμοί
   - Ειδικοί αριθμοί και σύμβολα
   - Στρογγύλευση
   - Έψιλον της μηχανής

Πώς υπολογίζουμε την ευκλείδεια νόρμα;

Κώδικας 1: Απλή εκδοχή

```
1 function [s]=norm2_naive(x);
2 n = length(x);
3 s = 0;
4 for i = 1:n, s = s+x(i)^2; end
5 s = sqrt(s);
6 % faster version
7 % s = sqrt(sum(x.^2));
```

Επιθυμητές ιδιότητες συνάρτησης ((Dem97))

1. Να υπολογίζει το αποτέλεσμα με ακρίβεια, δηλ. να είναι ορθά (σχεδόν) όλα τα ψηφία της απάντησης, εκτός αν το $\|x\|_2$ είναι (σχεδόν) εκτός του συνόλου των κανονικοποιημένων α.κ.υ. του συστήματος.
2. Να είναι (σχεδόν) όσο γρήγορο θα ήταν το απλό (αλλά μη αξιόπιστο) πρόγραμμα.
3. Να λειτουργεί αξιόπιστα ακόμα και εκτός αριθμητικής IEEE εκτός αν η θεωρητική τιμή είναι (σχεδόν) μεγαλύτερη του μέγιστου αναπαραστήσιμου

Αστοχίες της απλή εκδοχής

- Αν `x=[sqrt(1.7977e+308),sqrt(1.7977e+308)]` τότε

  $$norm2\_naive(x) \rightarrow Inf$$

  αντί για

  $$1.896150381621835e+154$$

Αστοχίες της απλή εκδοχής

- Αν `x=[sqrt(1.7977e+308),sqrt(1.7977e+308)]` τότε

  $$norm2\_naive(x) \rightarrow Inf$$

  αντί για

  $$1.896150381621835e+154$$

- Αν `x=[2.2251e-308]` τότε

  $$norm2\_naive(x) \rightarrow 0$$

  αντί για

  $$2.2251e-308$$

Αστοχίες της απλή εκδοχής

- Αν `x=[sqrt(1.7977e+308),sqrt(1.7977e+308)]` τότε

  $$norm2\_naive(x) \rightarrow Inf$$

  αντί για

  $$1.896150381621835e+154$$

- Αν `x=[2.2251e-308]` τότε

  $$norm2\_naive(x) \rightarrow 0$$

  αντί για

  $$2.2251e-308$$

ΠΡΟΣΟΧΗ: `realmax = 1.7977e+308; realmin = 2.2251e-308`. Αυτές οι τιμές δεν είναι οριακές για τη συνάρτηση!

Πώς μπορούμε να αποφύγουμε τις αστοχίες!

Παράδειγμα vnorm.m (MATLAB File Exchange)

File Exchange

from **Vector norm** by Winston Smith
Returns the vector norm for a specified dimension (e.g. row/col) of a matrix

vnorm(A,varargin)

```
function y = vnorm(A,varargin)
% VNORM - Return the vector norm along specified dimension of A
%
%   VNORM(A) returns the 2-norm along the first non-singleton
%   dimension of A
%   VNORM(A,dim) return the 2-norm along the dimension 'dim'
%   VNORM(A,dim,normtype) returns the norm specified by normtype
%   along the dimension 'dim'
%   VNORM(A,[],normtype) returns the norm specified by normtype along
%   the first non-singleton dimension of A
%
%   normtype may be one of {inf,-inf,positive integer}.
%   For a given vector, v, these norms are defined as
```

Παράδειγμα από vnorm.m (MATLAB File Exchange)

```
    end
end


if isempty(ntype)
    y = sqrt(sum( abs(A).^2 , dim, ,,
elseif ntype==1
    y = sum( abs(A) , dim );
elseif isinf(ntype)
    if ntype > 0
        y=max(abs(A), [], dim);
    else
        y=min(abs(A), [], dim);
    end
elseif ntype~=floor(ntype) || ntype<1
    error(['Norm type must be one of inf,-inf or a positive ' ...
           'integer']);
else
    y = (sum( abs(A).^ntype , dim) ).^(1/ntype);
end
```

πιθανή υπερχείλιση

Τροποποίηση

Ιδέα

$$\|x\|_2 = \xi_{\max} \sqrt{\sum_{i=1}^{n} \underbrace{\left(\frac{\xi_i}{\xi_{\max}}\right)^2}_{\leq 1}}, \text{ όπου } \xi_{\max} = \max(|x|)$$

```
1  function [s] = norm_2rat(x); % author: EG
2  n = length(x); s  = 0; xmax = max(abs(x));
3  if (xmax==0), return; end
4  for i = 1:n, s = s+(x(i)/xmax)^2; end
5  s = xmax*sqrt(s);
```

Θεραπεία;

1. `norm2_rat([sqrt(realmax),sqrt(realmax)])= 1.8962e+154`

ΠΡΟΣΟΧΗ εύρεση μεγίστου → 2 περάσματα από τα δεδομένα

Τροποποίηση

Ιδέα

$$\|x\|_2 = \xi_{max} \sqrt{\sum_{i=1}^{n} \underbrace{\left(\frac{\xi_i}{\xi_{max}}\right)^2}_{\leq 1}}, \text{ όπου } \xi_{max} = \max(|x|)$$

```
1  function [s] = norm_2rat(x); % author: EG
2  n = length(x); s  = 0; xmax = max(abs(x));
3  if (xmax==0), return; end
4  for i = 1:n, s = s+(x(i)/xmax)^2; end
5  s = xmax*sqrt(s);
```

Θεραπεία;

1. `norm2_rat([sqrt(realmax),sqrt(realmax)])= 1.8962e+154`
2. `norm2_rat(realmin)= 2.2251e-308`

ΠΡΟΣΟΧΗ εύρεση μεγίστου → 2 περάσματα από τα δεδομένα

Κώδικας αναφοράς BLAS-1 (Fortran)

```
1         DOUBLE PRECISION FUNCTION DNRM2 ( N, X, INCX )
2         INTEGER                          INCX, N
3         DOUBLE PRECISION                 X( * )
4   #   -- This version written on 25-October-1982. Modified ...
        on 14-October-1993 Sven Hammarling, Nag Ltd.
5
6         DOUBLE PRECISION      ONE         , ZERO
7         PARAMETER           ( ONE = 1.0D+0, ZERO = 0.0D+0 )
8         INTEGER               IX
9         DOUBLE PRECISION      ABSXI, NORM, SCALE, SSQ
10        INTRINSIC             ABS, SQRT
11  #     .. Executable Statements ..
12        IF( N<1 || INCX<1 )THEN
13           NORM  = ZERO
14        ELSE IF( N==1 )THEN
15           NORM  = ABS( X( 1 ) )
16        ELSE
17           SCALE = ZERO
```

Κώδικας αναφοράς (Fortran)

```
18              SSQ    = ONE
19              DO 10, IX = 1, 1 + ( N - 1 )*INCX, INCX
20                 IF( X( IX )!=ZERO )THEN
21                    ABSXI = ABS( X( IX ) )
22                    IF( SCALE<ABSXI )THEN
23                       SSQ   = ONE   + SSQ*( SCALE/ABSXI )**2
24                       SCALE = ABSXI
25                    ELSE
26                       SSQ   = SSQ   +     ( ABSXI/SCALE )**2
27                    END IF
28                 END IF
29        10    CONTINUE
30              NORM   = SCALE * SQRT( SSQ )
31           END IF
32           DNRM2 = NORM
33           RETURN
34           END
```

Εκδοχή MATLAB

```
1  function s = dnrm2(n,x,incx) %MATLAB BLAS-1 by J.Burkardt
2    if ( n < 1 | incx < 1 ), s = 0.0; % value = 0.0; ...
         /*correction by EG*/
3    elseif ( n == 1 ), s = abs(x(1)); %value = abs ...
         (x(1));/*correction by EG*/
4    else scale = 0.0; ssq = 1.0;
5      for ix = 1 : incx : 1 + ( n - 1 )*incx
6        if ( x(ix) ~= 0.0 )
7          absxi = abs ( x(ix) );
8          if ( scale < absxi )
9            ssq = 1.0 + ssq * ( scale / absxi )^2;
10           scale = absxi;
11         else
12           ssq = ssq + ( absxi / scale )^2;
13         end
14       end
15     end
16     s  = scale * sqrt( ssq );
```

Σχόλια

- Έξυπνος τρόπος: υπολογίζει κάθε φορά αν το νέο στοιχείο είναι μεγαλύτερο ή μικρότερο του μέχρι τώρα μεγίστου και ανάλογα προσαρμόζει τον υπολογισμό.
- Προσαρμογή:
- ένα πέρασμα από τα δεδομένα.
- ΠΩΣ; Διαβάστε τον κώδικα και επιβεβαιώστε!

Σχόλια

- Έξυπνος τρόπος: υπολογίζει κάθε φορά αν το νέο στοιχείο είναι μεγαλύτερο ή μικρότερο του μέχρι τώρα μεγίστου και ανάλογα προσαρμόζει τον υπολογισμό.
- Προσαρμογή:
- ένα πέρασμα από τα δεδομένα.
- ΠΩΣ; Διαβάστε τον κώδικα και επιβεβαιώστε!
- Αστοχία: `dnrm2([1,Inf]) = NaN`

Περισσότερες «απρόσμενες» ιστορίες

```
>> floor(0.075/0.025)
ans = 2
>> floor(0.75/0.25)
ans = 3
% observation due to C. Bekas
```

Περισσότερες «απρόσμενες» ιστορίες

```
>> floor(0.075/0.025)
ans = 2
>> floor(0.75/0.25)
ans = 3
% observation due to C. Bekas
```

```
double v = 1E308;
double x = (v * v) / v;
printf("%g %d\n", x, x==v);
```

Προσοχή (Mon08)

- με `gcc.0.1` σε Linux εκτυπώνει 10^{308}.
- με την επιλογή `-ffloat-store` εκτυπώνει $+\infty$.

Τι λένε οι κατασκευαστές

Των Corden & Kreitzer, Intel Consistency of Floating-Point Results using the Intel Compiler or Why doesn't my application always give the same answer?

Binary floating-point [FP] representations of most real numbers are inexact, and there is an inherent uncertainty in the result of most calculations involving floating-point numbers. Programmers of floating-point applications typically have the following objectives:

- Accuracy
  - Produce results that are "close" to the result of the exact calculation
    - Usually measured in fractional error, or sometimes "units in the last place" (ulp).
- Reproducibility
  - Produce consistent results:
    - From one run to the next;
    - From one set of build options to another;
    - From one compiler to another
    - From one processor or operating system to another
- Performance
  - Produce an application that runs as fast as possible

These objectives usually conflict! However, good programming practices and judicious use of compiler options allow you to control the tradeoffs.

Παράγοντες δυσκολίας

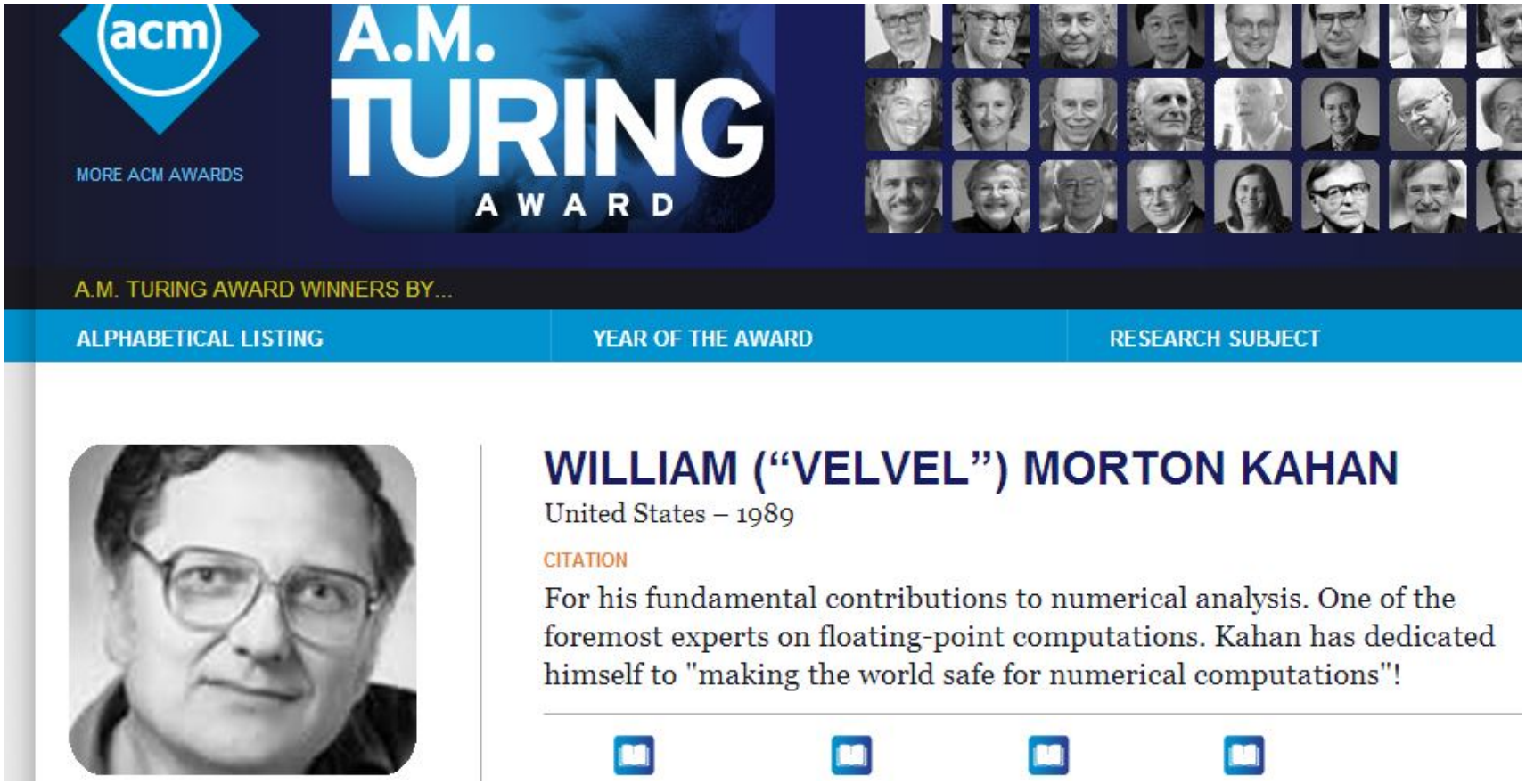
- Ακόμα και φαινομενικά αξιόπιστα προγράμματα χρειάζονται προσοχή ως προς την ορθότητα.

Παράγοντες δυσκολίας

- Ακόμα και φαινομενικά αξιόπιστα προγράμματα χρειάζονται προσοχή ως προς την ορθότητα.
- ΠΡΟΣΟΧΗ: Το «σκηνικό» περιλαμβάνει συνδυαστικά
  1. την αρχιτεκτονική (CPU, καταχωρητές και cache, μικροεντολές)
  2. το λογισμικό (γλώσσα και μεταφραστής, περιβάλλον χρόνου εκτέλεσης, αριθμητικές βιβλιοθήκες, πρόγραμμα)
  3. τον αλγόριθμο και την υλοποίησή του σε πρόγραμμα

Παρατήρηση: Η συγγραφή (ακόμα και) απλού αξιόπιστου κώδικα που είναι ταχύς και ακριβής είναι πολύπλοκη υπόθεση!

Πλαίσιο


acm
MORE ACM AWARDS
A.M. TURING AWARD
A.M. TURING AWARD WINNERS BY...
ALPHABETICAL LISTING
YEAR OF THE AWARD
RESEARCH SUBJECT
WILLIAM ("VELVEL") MORTON KAHAN
United States – 1989
CITATION
For his fundamental contributions to numerical analysis. One of the foremost experts on floating-point computations. Kahan has dedicated himself to "making the world safe for numerical computations"!

Πλαίσιο

FLOATING-POINT ARITHMETIC

AT THE MERCY OF

COMPILER WRITERS.

W. Kahan
Univ. of Calif.
Berkeley

FLOATING-POINT ARITHMETIC

SHORTCUTS

for Hardware Designers

TO AVOID.

W. Kahan
Univ. of Calif.
Berkeley

Mathematics Written in Sand — Version of 22 Nov. 1983

MATHEMATICS WRITTEN IN SAND -
the hp-15C, Intel 8087, etc.

W. Kahan,
University of California @ Berkeley

This paper was presented at the Joint Statistical Meeting of the American Statistical Association with ENAR, WNAR, IMS and SSC held in Toronto, Canada, August 15-18, 1983. Then the paper appeared in pp. 12-26 of the 1983 Statistical Computing Section of the Proceedings of the American Statistical Association. It had been typeset on an IBM PC and printed on an EPSON FX-80 at draft speed with an unreadable type-font of the author's devising, and then photo-reduced. The paper is reproduced here unaltered but for type fonts, pagination, and an appended Contents page.

ABSTRACT: Simplicity is a Virtue; yet we continue to cram ever more complicated circuits ever more densely into silicon chips, hoping all the while that their internal complexity will promote simplicity of use. This paper exhibits how well that hope has been fulfilled by several inexpensive devices widely used nowadays for numerical computation. One of them is the Hewlett-Packard hp-15C programmable shirt-

Αριθμοί κινητής υποδιαστολής (υπενθύμιση)

Τι είναι Ψηφιακή αναπαράσταση πραγματικών αριθμών με πεπερασμένο πλήθος ψηφίων (π.χ. 32 ή 64) ως προς κάποια βάση β (συνήθως 2) στα οποία αποθηκεύονται πρόσημο, εκθέτης και ουρά. Επειδη ο εκθέτης δεν είναι σταθερός, δίνεται η δυνατότητα αναπαράστασης μεγάλου εύρους τιμών.

Τριμερής κώδικας για κάθε αριθμό:

1. πρόσημο s (0 αν θετικός, 1 αν αρνητικός)
2. εκθέτης $e = (a_1 a_2 \ldots a_k)_2$ (πολωμένος κατά P)
3. ουρά $(b_0 b_1 b_2 \ldots b_{t-1})_2$

Τότε

$$x = (-1)^s \times 2^{e-P} \times (b_0 + b_1 \beta^{-1} + \cdots + b_{t-1} \beta^{-(t-1)})$$

Συμβολίζουμε το σύστημα των α.κ.υ. ως $\mathcal{F}(\beta, t, e_{\min}, e_{\max})$ και με F το σύνολο των εν α.κ.υ. του συστήματος.

Οι συνηθισμένοι α.κ.υ. είναι πάντα ρητοί!

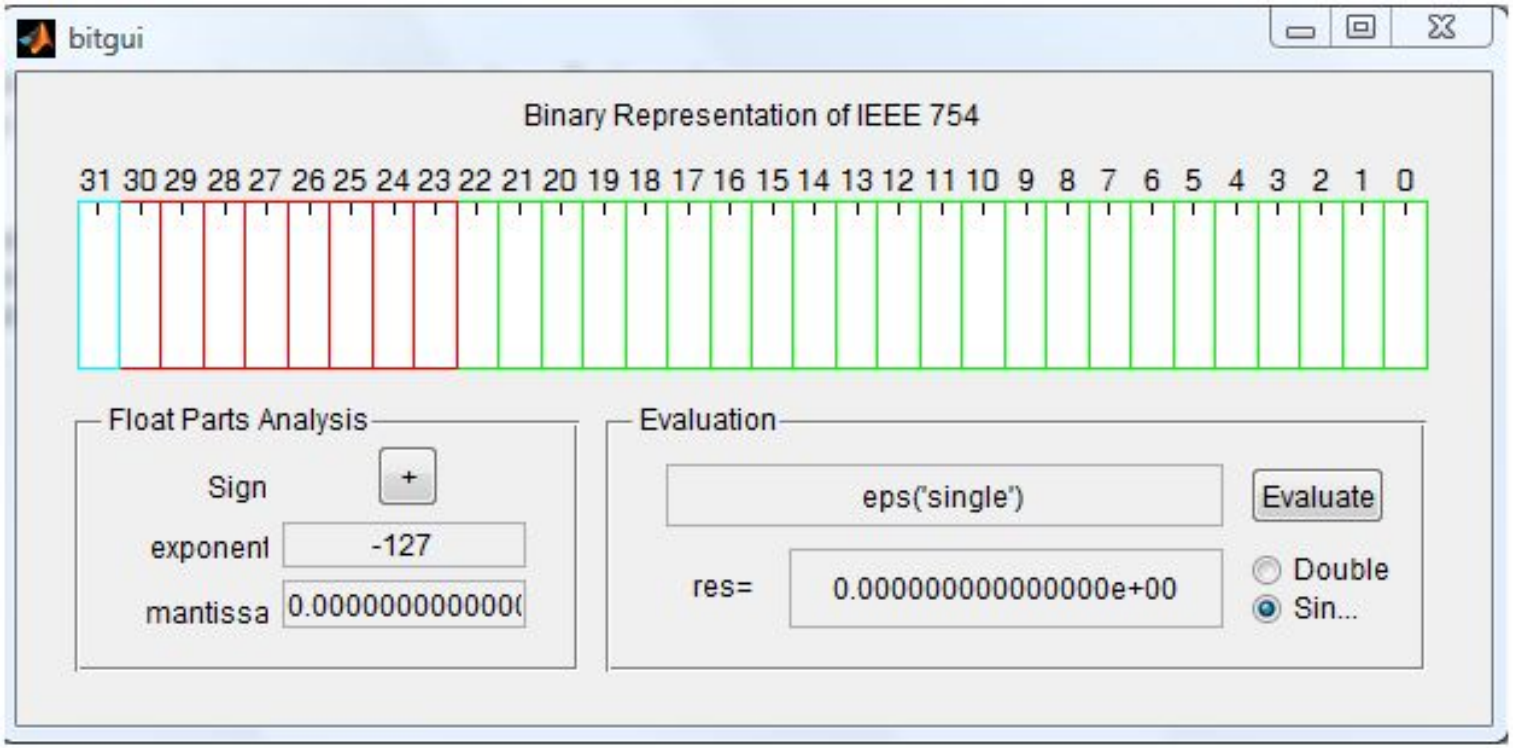
Κανονικοποίηση

- Για να αποφύγουμε την πολλαπλή αναπαράσταση του ίδιου αριθμού που επιτρέπει η «Επιστημονική Γραφή Αριθμών», $a \times 10^e$,
- π.χ. .. ότι το 350 μπορεί να γραφτεί ως 3.5×10^2, ή 35×10^1, ή 350×10^0, ...
- χρησιμοποιείται κανονικοποιημένη (normalized) αναπαράσταση: π.χ. επιλέγεται $1 \leq |a| < 10$.
- Με βάση 2, υποχρεώνουμε την ουρά a να είναι $1 \leq |a| < 2$. Τότε με βάση τα προηγούμενα

$$x = (-1)^s \times 2^{e-P} \times (1 + b_1 2^{-1} + \cdots + b_{t-1} 2^{-(t-1)})$$

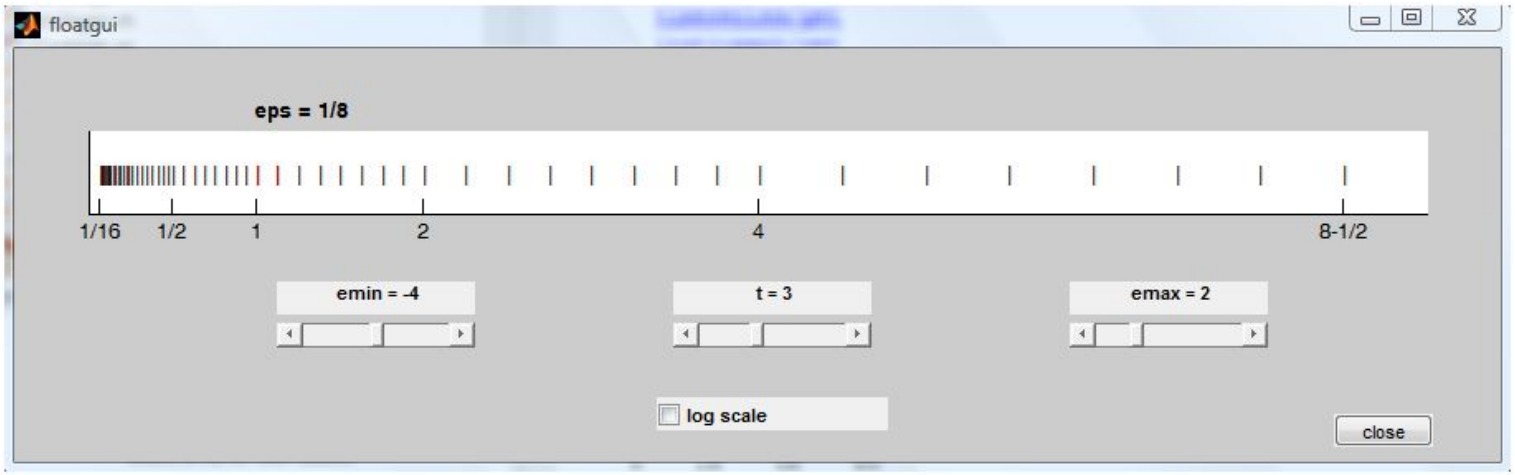
- Αν χρησιμοποιούμε κανονικοποιημένη αναπαράσταση, το μέγεθος του εκθέτη e καθορίζει άμεσα την τάξη μεγέθους του αριθμού.

Το `δικό μας' bitgui



Σχήμα: Το bitgui.m που αναπτύχθηκε στα πλαίσια του ΕΥ1 από τον Γ. Καλοφωλιά (Saarbrucken).

Κατανομή α.κ.υ.



Σχήμα: Οι α.κ.υ. είναι ανισοκατανεμημένοι στον άξονα των πραγματικών. Το παραπάνω προέρχεται από τη συνάρτηση flotgui.m που περιέχονται στη βιβλιοθήκη NCM.

- Η απόσταση διαδοχικών α.κ.υ. που έχουν τον ίδιο εκθέτη, π.χ. $d(e) = (m+1)2^e - m2^e$, είναι σταθερή.
- Η απόσταση μεταξύ διαδοχικών α.κ.υ. διπλασιάζεται κάθε φορά που ο εκθέτης αυξάνει κατά 1: $(m+1)2^{e+1} - m2^{e+1} = d(e+1) = 2d(e)$

Αριθμητική α.κ.υ.

Ευχή

... The simplest and best, though harder to attain, solution to the problem of environmental parameters is to standardize floating-point hardware, so that the values of the parameters become universal constants. (W. Miller, The Engineering of Numerical Software, 1984.)

Πραγματοποίηση (1985)

μια από τις μεγαλύτερες επιτυχίες στην επιστήμη και τεχνολογία των υπολογιστών ήταν η υιοθέτηση του προτύπου IEEE για την α.κ.υ.

Χαρακτηριστικά προτύπου IEEE 754

είδη αριθμών: πεπερασμένα σύνολα δυαδικών και δεκαδικών α.κ.υ. Συμπεριλαμβάνονται «προσημασμένο μηδενικό», «προσημασμένο άπειρο», «υποκανονικοποιημένοι αριθμοί), και η «τιμή» not a number (`NaN`) για «αόριστα αποτελέσματα».

αλγόριθμοι στρογγύλευσης: μέθοδοι για την στρογγύλευση αριθμών κατά τις αριθμητικές πράξεις και τις μετατροπές.

πράξεις: αριθμητικές και άλλες πράξεις σε αριθμητικά δεδομένα

διαχείριση εξαιρέσεων: επισήμανση ιδιαίτερων καταστάσεων (διαίρεση με 0, υπερχείλιση, κ.λπ.)

συστάσεις: για διαχείριση εξαιρέσεων, υπολογισμό εκφράσεων, υπολογισμό ιδιαίτερων συναρτήσεων (π.χ. τριγωνομετρικών), κ.λπ.

format μετατροπών: δυαδικές κωδικοποιήσεις για τη διευκόλυνση μεταφορών α.κ.υ.

Χρήσιμες αναφορές: (Mon08), (M+10) (Gol91)

Είδη α.κ.υ. στο πρότυπο IEEE-754

	single	single-ext	double	double-ext	quad-precision
μήκος α.κ.υ.	32	$\geq$ 43	64	80	128
e_{max}	+127	1023	+ 1023	+16383	+ 16383
e_{min}	-126	1022	-1022	-16382	-16382
πόλωση	+127	+1023	+1023	+16383	+16383
bits *m t*	24	$\geq$ 32	53	$\geq$ 64	113
bits *s*	1	1	1	1	1
bits *e*	8	11	11	15	15

Βαθμιαία υποχείλιση και υποκανονικοποιημένοι αριθμοί

Το πρότυπο IEEE επιτρέπει το αποτέλεσμα πράξεων μεταξύ α.κ.υ. να είναι α.κ.υ. που είναι μικρότεροι του ελάχιστου κανονικοποιημένου.

Υποκανονικοποιημένοι αριθμοί (subnormal numbers) Τότε το (κρυφό bit) είναι 0. Η κωδικοποίηση αυτών των αριθμών έχει 0 σε όλες τις θέσεις του εκθέτη. Έτσι αξιοποιούνται όλα τα bits μετά την υποδιαστολή της ουράς όταν η απόλυτη τιμή του αποτέλεσματος της πράξης είναι μικρότερη του $2^{e_{\min}}$.

Ελάχιστες τιμές

στην κανονικοποιημένη αναπαράσταση $2^{e_{\min}}$, π.χ. `2.2251e-308` σε διπλή ακρίβεια IEEE

στην υποκανονικοποιημένη αναπαράσταση $2^{e_{\min}-t+1}$, π.χ. `4.9407e-324` σε διπλή ακρίβεια IEEE. Διαίρεση με όποιο $y > 1$ επιστρέφει 0.

Στη MATLAB

Κώδικας 2: Παραδείγματα (για οικονομία έχουμε αφαιρέσει το `ans`)

```
1 format hex;
2 realmin = 0010000000000000
3 realmin/2 = 0008000000000000  %(upokanonikopoihm'enoc)
4 realmin/2^52 =  0000000000000001
5 realmin/2^53 =  0000000000000000
```

Ειδικές τιμές και σύμβολα

(Πέραν των μη κανονικοποιημένων αριθμών) το πρότυπο της IEEE προβλέπει την αναπαράσταση των παρακάτω ειδικών τιμών. Οι τιμές αυτές ανήκουν στο σύνολο F και μπορούμε να κάνουμε πράξεις με αυτές.

-0 υπάρχει επειδή η ουρά έχει αναπαράσταση σεσημασμένου προσήμου

$\pm\infty$ όπως και το σύστημα των πραγματικών αριθμών, είναι ανάγκη να επαυξήσουμε με σύμβολα για το + και - άπειρο

NαN Not a Number χρησιμοποιείται για την αναπαράσταση οποιουδήποτε αόριστου αποτελέσματος, όπως $0/0, \infty \times 0$.

Εξαιρέσεις

Εξαίρεση	Παράδειγμα	Αποτέλεσμα
invalid op	$0/0, 0 \times \mathrm{Inf}$	`NaN`
overflow		$\pm$`Inf`, $\pm$`realmax`
divide by 0		$\pm$`Inf`
underflow		$\pm$0, $\pm$`realmin`, subnormal
inexact	$\mathrm{fl}(x \oplus y) \neq x \oplus y$	στρογγύλευση

όπου `realmin`, `realmax` είναι ο ελάχιστος κανονικοποιημένος και ο μέγιστος α.κ.υ. αντίστοιχα για τν υπό συζήτηση δεδομένη αναπαράσταση.

Προσοχή `NaN == NaN` $\rightarrow$ 0

Κωδικοποίηση για ΜΟΝΗ ακρίβεια στο πρότυπο IEEE

$\pm$	$a_1 a_2 \ldots a_8$	$b_1 b_2 \ldots b_{23}$

Αν ο εκθέτης είναι	τότε η τιμή είναι
$(00000000)_2 = (0)_{10}$	$\pm(0.b_1 b_2 \ldots b_{23})_2 \times 2^{-126}$ (υποκανονικοποιημένος)
$(00000000)_2 = (0)_{10}$	$\pm(0.0 \ldots 0)_2 \times 2^{-126} = \pm 0$
$(00000001)_2 = (1)_{10}$	$\pm(1.b_1 b_2 \ldots b_{23})_2 \times 2^{-126}$
$(00000010)_2 = (2)_{10}$	$\pm(1.b_1 b_2 \ldots b_{23})_2 \times 2^{-125}$
$\vdots$	$\vdots$
$(01111111)_2 = (127)_{10}$	$\pm(1.b_1 b_2 \ldots b_{23})_2 \times 2^{0}$
$(10000000)_2 = (128)_{10}$	$\pm(1.b_1 b_2 \ldots b_{23})_2 \times 2^{1}$
$\vdots$	$\vdots$
$(11111110)_2 = (254)_{10}$	$\pm(1.b_1 b_2 \ldots b_{23})_2 \times 2^{127}$
$(11111111)_2 = (255)_{10}$	$\pm\infty$ αν $b_1 = \cdots = b_{23} = 0$, αλλοιώς `NaN`

Στρογγύλευση

Έστω F το σύνολο των α.κ.υ. που αναπαρίστανται στο $\mathcal{F}$. Σημαντικό ρόλο παίζει η συνάρτηση στρογγύλευσης

$$\mathrm{fl} : \mathbb{R} \rightarrow F$$

που για κάθε $z \in \mathbb{R}$ επιστρέφει κάποια τιμή $\mathrm{fl}(z) \in F$ σύμφωνα με κάποια προσυμφωνημένη μέθοδο στρογγύλευσης για την οποία πρέπει να ισχύουν οι εξής ιδιότητες:

- Αν $z \in F \Rightarrow \mathrm{fl}(z) = z$.
- Αν $z_1 \leq z_2$ τότε $\mathrm{fl}(z_1) \leq \mathrm{fl}(z_2)$.

Περιπτώσεις

- Το z είναι δεδομένο που θα αποτελέσει είσοδο στο πρόγραμμα.
- Το z είναι αποτέλεσμα πράξης μεταξύ δύο α.κ.υ.

Επίσης έστω ότι $G \supset F$

$$G := \{x \in \mathbb{R} \ : m \leq |x| \leq M\} \cup \{0\} \subset \mathbb{R} \ ,$$

όπου $m \in F$ είναι ο ελάχιστος μη μηδενικός και $M \in F$ ο μέγιστος θετικός (κανονικοποιημένος) α.κ.υ. στο $\mathcal{F}$. Τότε ισχύει ένα από τα εξής:

$z \in F$ άρα $\mathrm{fl}(z) = z$.

$z \in G$ και $z \notin F$ άρα $\mathrm{fl}(z) \neq z$ (στρογγύλευση, η διαφορά $|\mathrm{fl}(z) - z|$ είναι το απόλυτο σφάλμα στρογγύλευσης).

$z \notin G$ και $|\mathrm{fl}(z)| > M$ υπερχείλιση, το αποτέλεσμα $\in (-\infty, -M, M, \infty)$, εξαρτάται από τη μέθοδο στρογγύλευσης.

$z \notin G$ και $|\mathrm{fl}(z)| < m$ υποχείλιση, $0 < |\mathrm{fl}(z)| < m$. Το πρότυπο IEEE περιέχει υποκανονικοποιημένους αριθμούς και υποστηρίζεται η βαθμιαία υποχείλιση.

Η πιο συνηθισμένη μέθοδος στρογγύλευσης

> *Round-to-nearest mode:* *In this mode, the representable value nearest to the infinitely precise result shall be delivered; if the two nearest representable values are equally near, the one with its least significant bit equal to zero shall be delivered.*

Στις στρατηγικές στρογγύλευσης ενός $z \notin F$, έχουν σημασία ο πλησιέστερος α.κ.υ. μικρότερος του z και ο πλησιέστερος α.κ.υ. μεγαλύτερος του z. Έστω ότι τους συμβολίζουμε ως $z_-, z_+ \in F$ αντίστοιχα.

Στρογγύλευση προς το πλησιέστερο άρτιο

Θέτουμε $\mathrm{fl}(z) = \arg\min_{y \in \{z_-, z_+\}} |y - z|$ και αν $z - z_- = z_+ - y$ τότε θέτουμε $\mathrm{fl}(z)$ το ένα από τα z_-, z_+ που έχει στην ουρά του το 0 ως τελευταίο bit.

Συμβολισμός Το σύμβολο $\arg\min_{x \in X} f(x)$[1] είναι η τιμή ή το σύνολο τιμών $x \in X$ στις οποίες ελαχιστοποιείται η $f(x)$. Π.χ. αν $f(x) = x^2 + 1$ τότε $\min_{x \in \mathbb{R}} f(x) = 1$ ενώ $\arg\min_{x \in X} f(x) = 0$.

[1] Προέρχεται από το argument of the mininimum και προφέρεται (άργκ-μίν).

Υπάρχουν άλλοι τρόποι στρογγύλευσης;

ΒΕΒΑΙΩΣ: στο πρότυπο IEEE προβλέπονται 5 τρόποι που είναι χρήσιμοι σε ορισμένες περιπτώσεις[2]

1. Προς τον πλησιέστερο, ισοπαλίες προς ζυγό (default)
2. Προς τον πλησιέστερο, ισοπαλίες μακρύτερα από το 0
3. Προς το 0 (αποκοπή)
4. Προς το ∞
5. Προς το $-\infty$

Δεν προσφέρεται πάντα εύκολη λογισμική υποστήριξη. Στη MATLAB υπάρχει τρόπος αλλά είναι «κρυμμένος».

Κώδικας 3: Και όμως - η «μυστική» εντολή feature

```
>> feature('setround',n) % JETOUME n SE 0.5 (round to ...
    nearest even), 0, + H - Inf (round to ....)
```

[2] π.χ. στην ανάλυση διαστημάτων - δείτε επόμενη διάλεξη.

Έψιλον της μηχανής

Ορισμός

Το έψιλον της μηχανής είναι η απόσταση του 1.0 από τον αμέσως μεγαλύτερο α.κ.υ.

Σημ. Στην IEEE-754: single precision $\epsilon_M \approx 1.1921e-007$· double precision $\epsilon_M \approx 2.2204e-016$. Στη MATLAB δείτε τις εντολές `eps` και `eps('single')`.

Ενδιαφέροντα φαινόμενα:

Είσοδος	`format short.`	`format hex`
$1+$`eps`$/2$	1	3*ff*0000000000000
$1+$`eps`	1.0000	3*ff*0000000000001
$1+$`eps`$/2+$`eps`	1.0000	3*ff*0000000000001
$1+$`eps`$+$`eps`$/2$	1.0000	3*ff*0000000000002
$1+2*$`eps`	1.0000	3*ff*0000000000002
$1+2*$`eps`$+$`eps`$/2+$`eps`$/4$	1.0000	3*ff*0000000000002

Έψιλον της μηχανής

Ορισμός

Το έψιλον της μηχανής είναι η απόσταση του 1.0 από τον αμέσως μεγαλύτερο α.κ.υ.

Σημ. Στην IEEE-754: single precision $\epsilon_M \approx 1.1921e-007$· double precision $\epsilon_M \approx 2.2204e-016$. Στη MATLAB δείτε τις εντολές `eps` και `eps('single')`.
Ενδιαφέροντα φαινόμενα: `feature('setround',Inf)`

Είσοδος	`format short.`	`format hex`
`1+eps/2`	1.0000	3ff0000000000001
`1+eps`	1.0000	3ff0000000000001
`1+eps/2+eps`	1.0000	3ff0000000000002
`1+eps+eps/2`	1.0000	3ff0000000000002
`1+2*eps`	1.0000	3ff0000000000002
`1+2*eps+eps/2+eps/4`	1.0000	3ff0000000000004

Θεμελιώδης αρχή

Αρχή ακριβούς στρογγύλευσης (ΑΑΣ)

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.
- Η υλοποίηση δεν είναι απλή! Πέρασαν δεκαετίες μέχρι να βρεθεί οικονομικός και ορθός τρόπος υλοποίησης και τους κατασκευαστές να την υιοθετήσουν.
- Φαινόταν ότι θα απαιτούνταν πολύ μεγάλοι καταχωρητές
- αλλά αρκούν 3 επιπλέον ψηφία (guard digit, rounding digit, sticky bit)!

ΠΡΟΣΟΧΗ στο double rounding πολλών επεξεργαστών.

Ιστορικά

Οι κατασκευαστές γενικά ακολουθούν το πρότυπο IEEE-754 εδώ και καιρό αλλά ενίοτε έχουμε παραβάσεις (ακόμα και πρόσφατα)...

GPU Floating-Point Paranoia

Karl E. Hillesland
University of North Carolina at Chapel Hill *

Anselmo Lastra
University of North Carolina at Chapel Hill *

1 Introduction

Up until the late eighties, each computer vendor was left to develop their own conventions for floating-point computation as they saw fit. As a result, programmers needed to familiarize themselves with the peculiarities of each system in order to write effective software and evaluate numerical error. In 1987, a standard was established for floating-point computation to alleviate this problem, and CPU vendors now design to this standard [IEEE 1987].

Today there is an interest in the use of graphics processing units, or GPUs, for non-graphics applications such as scientific computing. GPUs have floating-point representations similar to, and sometimes matching, the IEEE standard. However, we have found that GPUs do not adhere to IEEE standards for floating-point operations, nor do they give the information necessary to establish bounds on error for these operations. Another complication is that this behavior seems to be in a constant state of flux due to the depen-

Operation	R300/arbfp	NV30/fp30
Addition	[-1.000, 0.000]	[-1.000, 0.000]
Subtraction	[-1.000, 1.000]	[-0.750, 0.750]
Multiplication	[-0.989, 0.125]	[-0.782, 0.625]
Division	[-2.869, 0.094]	[-1.199, 1.375]

Table 1: Floating-Point Error in ULPs (Units in Last Place). Note that the R300 has a 16 bit significand, whereas the NV30 has 23 bits. Therefore one ULP on an R300 is equivalent to 2^7 ULPs on an NV30. Division is implemented by a combination of reciprical and multiply on these systems. Cg version 1.2.1. ATI driver 6.14.10.6444. NVIDIA driver 56.72.

Schryer [Schryer 1981]. By testing all combinations of these numbers, we include all the test cases in Paranoia, as well as cases that push the limits of round-off error and cases where the most work must be performed, such as extensive carry propagation. Table 1 gives results for some example systems.

Βιβλιογραφία I

J.W. Demmel.

Applied Numerical Linear Algebra.

SIAM, Philadelphia, 1997.

D. Goldberg.

What every computer scientist should know about floating point arithmetic.

ACM Comput. Surveys, pages 5--48, 1991.

J.-M. Muller et al.

Handbook of Floating-Point Arithmetic.

Birkhäuser Boston, 2010.

David Monniaux.

The pitfalls of verifying floating-point computations.

ACM Trans. Program. Lang. Syst., 30(3):12:1--12:41, May 2008.

Ε. Γαλλόπουλος.

Επιστημονικός Υπολογισμός Ι.

Πανεπιστήμιο Πατρών, 2008.

Χρήση Έργου Τρίτων Ι

1. http://www.mathworks.com/matlabcentral/fileexchange/10708-vector-norm (βλ. σελ 7-8)
2. https://software.intel.com/sites/default/files/managed/33/49/FP_Consistency_080814_0.pdf (βλ. σελ 15)
3. http://amturing.acm.org/award_winners/kahan_1023746.cfm (βλ. σελ 17)
4. http://www.cs.berkeley.edu/ wkahan/MathSand.pdf (βλ. σελ 17)
5. http://www.eecs.berkeley.edu/Faculty/Photos/Homepages/kahan.jpg (βλ. σελ 17)
6. http://www.mathworks.com/matlabcentral/fileexchange/33874-bitgui--a-graphical-explorer-of-the-ieee-754-floating-point-formats (βλ. σελ 20)
7. http://www.mathworks.com/moler/ncmfilelist.html (βλ. σελ 21)
8. http://www.cs.unc.edu/ ibr/projects/paranoia/gpu_paranoia.pdf (βλ. σελ 36)

Σημείωμα Αναφοράς

``Επιστημονικός Υπολογισμός Ι'', Έκδοση: 1.0, Πάτρα 2013-2014.

Διαθέσιμο από τη δικτυακή διεύθυνση: https://eclass.upatras.gr/courses/CEID1096/

Τέλος Ενότητας

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης