Computing 34, 365–373 (1985)

Computing

Short Communications / Kurze Mitteilungen

Inner Product Rounding Error Analysis in the Presence of Underflow

A. Neumaier, Freiburg i. Br.

Received October 24, 1983; revised June 7, 1984

Abstract — Zusammenfassung

Inner Product Rounding Error Analysis in the Presence of Underflow. Wilkinson's classical error analysis for sums and inner products is extended to the case where underflow may occur. This is relevant for the construction of rigorous error bounds for an inner product evaluated on computers which do not give underflow messages. The analysis also covers calculations with gradual underflow.

AMS Subject Classification: 65G05, 65G10.

Key words: Error analysis, underflow, sums, inner products.

Rundungsfehleranalyse von inneren Produkten bei Unterlauf. Die klassische Methode von Wilkinson für die Fehleranalyse von Summen und inneren Produkten wird erweitert auf den Fall des Exponentenunterlaufs. Dies ist wichtig für die Konstruktion von Fehlerschranken von inneren Produkten bei der Berechnung auf Computern, die Exponentenunterlauf nicht anzeigen. Die Methode umfaßt auch den Fall des graduellen Exponentenunterlaufs.

1. Introduction

In his now classical book on error analysis, Wilkinson [6] treats in detail the computation of inner products on a computer which performs all operations with bounded relative error. Real computers satisfy this requirement only if neither overflow nor underflow occurs. Whereas the danger of overflow is small, and usually interrupts the computation, the occurence of underflow is more likely (e.g. in the computation of small residuals), and is usually not noticed by the user. Thus, for the design of a portable routine for the computation of inner products with rigorous error bounds, Wilkinsons's error analysis has to be modified to handle underflow effects.

This is done in the present paper. The analysis given is general enough to cover fixpoint arithmetic, normalized floating point arithmetic, and floating point arithmetic with gradual underflow (the latter as described e.g. in Kahan and Palmer [1]). Moreover, emphasis is given to the problem of presenting the error bounds in such a form that they remain strict bounds even if their evaluation involves roundoff errors. The latter technique was introduced by Olver [4] in connection with absolute and relative precision; the treatment given here avoids the exponential expressions occuring in Olver's paper.

The paper is concluded with three portable algorithms (written in an informal programming language) performing three specific tasks:

SUM(x,n,s) computes an upper bound s for a sum of nonnegative numbers x_i $(i=1,\ldots,n)$:

$$\sum_{i=1}^{n} x_i \leqq s,$$

with slight overestimation only.

IPROD$(A,B,c,i,j1,j2,k,r,e)$ computes an approximate inner product r with error bound e:

$$\left| c - \sum_{j=j1}^{j2} a_{ij} b_{jk} - r \right| \leqq e,$$

with small e.

DIV(a,b,c,e) computes an approximate quotient q and a small residual error bound e such that

$$|a - cb| \leqq e.$$

These algorithms are applied in Neumaier [2] to the construction of a portable algorithm for the computation of a matrix inverse with rigorous, realistic, and componentwise error bounds with $n^3 + O(n^2)$ multiplications.

2. Roundoff Error Control: Error Analysis

We begin with some remarks concerning computer arithmetic. Let $\mathcal{M}$ be the set of machine numbers used for our calculation on a given computer. We denote by $\bar{\varphi}$ the result of an arithmetic expression φ when evaluated on the computer, and make the following assumptions on $\mathcal{M}$ and the basic arithmetical operations $\circ \in \{+,-,\cdot,/\}$.

*A*1: There are small numbers ε, η such that for all $a,b \in \mathcal{M}$ for which the results $\overline{a \circ b}$ is defined,

$$\overline{a \circ b} = (a \circ b)(1+\alpha)+\alpha', \quad |\alpha| \leqq \varepsilon, \ |\alpha'| \leqq \eta, \ \alpha\alpha' = 0.$$

The number ε is the relative precision and η the underflow threshold of the computer. We require that η (but not necessarily ε) is in $\mathcal{M}$.

*A*2: There is a large integer N such that if i,k are integers in the range $-N \leqq i,k \leqq N$ then $i,k \in \mathcal{M}$ and

$$\overline{i \circ k} = (i \circ k)(1+\alpha), \quad |\alpha| \leqq \varepsilon;$$

moreover, $\alpha = 0$ if $i \circ k$ is also an integer in this range.

*A*3: If $a \circ b \geqq 0$ then $\overline{a \circ b} \geqq 0$.

If the computer works with normalized floating point numbers with basis B, mantissa length L, exponent range $[-E,F]$ with $E,F \geqq L$, and if the arithmetic registers have at least one guard digit then *A*1 is satisfied with $\varepsilon = B^{1-L}$ and

$\eta = B^{-1-E}$ (cf. Olver [3], Sterbenz [5], Wilkinson [6]), and $A2$ holds with $N = B^L - 1$. On some computers with gradual underflow, η is significantly smaller, $\eta = B^{-L-E}$ (cf. Kahan and Palmer [1]). If the computer works with fixpoint numbers with K digits before and L digits after the fractional point then $A1$ holds[1] with $\varepsilon = 0$ and $\eta = B^{-L}$, and $A2$ holds with $N = B^K - 1$.

We now define

$$q = (1-\varepsilon)^{-1}. \tag{1}$$

Note that usually q is not a machine number.

Proposition 1:

If $a, b \in \mathscr{M}$ and $\overline{a \circ b}$ is defined then

$$\overline{a \circ b} = (a \circ b)(1+\alpha) + \alpha', \tag{2}$$

$$a \circ b = (\overline{a \circ b})(1+\alpha'') - \alpha', \tag{3}$$

where

$$|\alpha|, |\alpha''| \leqq q-1, \; |\alpha'| \leqq \eta, \; \alpha'\alpha'' = 0. \tag{4}$$

In particular, if $a \circ b$ is nonnegative then

$$a \circ b \leqq (\overline{a \circ b})q + \eta, \tag{5}$$

$$\overline{a \circ b} \leqq (a \circ b)q + \eta. \tag{6}$$

Proof: Since $\alpha\alpha' = 0$, $A1$ implies

$$a \circ b = (\overline{a \circ b})/(1+\alpha) - \alpha' = \overline{a \circ b}\,(1+\alpha'') - \alpha'$$

with

$$|\alpha''| = |-\alpha/(1+\alpha)| \leqq \varepsilon/(1-\varepsilon).$$

Since $\varepsilon \leqq \varepsilon/(1-\varepsilon) = q-1$ the first part follows. The second part is an obvious consequence. □

Proposition 2:

If $a_i, b_i \in \mathscr{M}$ are nonnegative for $i = 1, \ldots, n$ then

$$\sum_{i=1}^{n} a_i \leqq \left(\overline{\left(\sum_{i=1}^{n} a_i\right)} + (n-1)\eta\right) q^{n-1}, \tag{7}$$

$$\sum_{i=1}^{n} a_i b_i \leqq \left(\overline{\left(\sum_{i=1}^{n} a_i b_i\right)} + (2n-1)\eta\right) q^{n}. \tag{8}$$

Proof: We assume that the sum is evaluated in the natural ordering (the result can be shown to be true also for other orderings). We denote the true and computed partial sums in (7) by

[1] Moreover, if $\circ \in \{+, -\}$ then also $\alpha' = 0$. In particular, the following considerations can be sharpened in this case.

$$r_t = \sum_{i=1}^{t} a_i, \qquad s_t = \overline{\sum_{i=1}^{t} a_i}; \tag{9}$$

then we have

$$r_1 = s_1 = a_1,\ r_{t+1} = r_t + a_{t+1},\ s_{t+1} = \overline{s_t + a_{t+1}}. \tag{10}$$

For $n=1$, (7) is obvious; assuming therefore that (7) holds for some $t \geqq 1$ in place of n we have

$$r_t \leqq (s_t + (t-1)\eta)\, q^{t-1}. \tag{11}$$

Using $q \geqq 1$ and Proposition 1 we get

$$r_{t+1} = r_t + a_{t+1} \leqq (s_t + a_{t+1} + (t-1)\eta)\, q^{t-1} \leqq (\overline{(s_t + a_{t+1})}\, q + \eta + (t-1)\eta)\, q^{t-1} =$$
$$= (s_{t+1} q + \eta t)\, q^{t-1} = (s_{t+1} + \eta t)\, q^t.$$

Therefore, (7) holds in general.

To prove (8) we apply Proposition 1 and (7):

$$\sum a_i b_i \leqq \sum (\overline{a_i b_i})\, q + \eta) = (\sum \overline{a_i b_i})\, q + n\eta \leqq$$
$$\leqq (\overline{\sum a_i b_i} + (n-1)\eta)\, q^{n-1}\, q + n\eta \leqq (\overline{\sum a_i b_i} + (2n-1)\eta)\, q^n. \quad \square$$

Now we consider the evaluation of the expression

$$r = c - \sum_{j=j1}^{j2} a_{ij} b_{jk} \tag{12}$$

on the computer. The calculations proceed according to the formulae

$$s_{j1-1} = c,$$
$$p_j = \overline{a_{ij} b_{jk}},\ s_j = \overline{s_{j-1} - p_j}\ (j = j1, \ldots, j2), \tag{13}$$
$$s = s_{j2}.$$

To describe the errors we compute nonnegative quantities e_j, f_j, e by

$$e_{j1} = |p_{j1}|,\ f_{j1} = |s_{j1}|,$$
$$e_j = \overline{e_{j-1} + |p_j|},\ f_j = \overline{f_{j-1} + |s_j|}\ (j = j1+1, \ldots, j2), \tag{14}$$
$$e = (\overline{f_{j2} + e_{j2}}).$$

Then we have

Proposition 3:
If s and e are computed by (13) and (14) then with

$$n = j2 - j1 + 1, \tag{15}$$

the true value r of (12) satisfies the bound

$$|r - s| \leqq (e(q-1) + 2n\eta)\, q^n. \tag{16}$$

Proof: Define for $t=j1-1,\ldots,j2$ the partial residuals

$$r_t=c-\sum_{j=j1}^{t} a_{ij}b_{jk}-s_t. \tag{17}$$

We show by induction that for $t=j1,\ldots,j2$,

$$|r_t|\leqq[(f_t+e_t)(q-1)+2\eta(t+1-j1)]\,q^{t-j1}. \tag{18$_t$}$$

Indeed, by (13) and Proposition 1 we have

$$a_{it}b_{tk}=p_t(1+\alpha_t'')+\alpha_t',\quad s_{t-1}-p_t=s_t(1+\beta_t'')+\beta_t'$$

with quantities α_t'', β_t'' of absolute value $\leqq q-1$, and α_t', β_t' of absolute value $\leqq n$. By (17) we have

$$\begin{aligned}|r_t|&=|r_{t-1}+s_{t-1}-a_{it}b_{tk}-s_t|=\\&=|r_{t-1}+p_t+s_t(1+\beta_t'')+\beta_t'-p_t(1+\alpha_t'')-\alpha_t'-s_t|=\\&=|r_{t-1}+s_t\beta_t''-p_t\alpha_t''+\beta_t'-\alpha_t'|,\end{aligned}$$

whence

$$|r_t|\leqq|r_{t-1}|+(|s_t|+|p_t|)(q-1)+2\eta. \tag{19}$$

Since $r_{j1-1}=c-s_{j1-1}=0$, $|s_{j1}|=f_{j1}$, $|p_{j1}|=e_{j1}$ we get immediately (18$_{j1}$). For $t\geqq j1+1$ we observe that (14) and Proposition 1 give

$$e_{t-1}+|p_t|\leqq e_tq+\eta,\quad f_{t-1}+|s_t|\leqq f_tq+\eta. \tag{20}$$

Now (19), $q\geqq 1$, (18$_{t-1}$), and (20) imply that

$$\begin{aligned}r_t&\leqq[(f_{t-1}+e_{t-1})(q-1)+2\eta(t-j1)]\,q^{t-1-j1}+[(|s_t|+|p_t|)(q-1)+2\eta]\,q^{t-1-j1}\\&\leqq[(f_{t-1}+|s_t|+e_{t-1}+|p_t|)(q-1)+2\eta+2\eta(t-j1)]\,q^{t-1-j1}\\&\leqq[(f_tq+\eta+e_tq+\eta)(q-1)+2\eta+2\eta(t-j1)]\,q^{t-1-j1}\\&=[(f_t+e_t)(q-1)+2\eta+2\eta(t-j1)\,q^{-1}]\,q^{t-j1}\\&\leqq[(f_t+e_t)(q-1)+2\eta(t+1-j1)]\,q^{t-j1}.\end{aligned}$$

Therefore (18$_t$) holds for $t=j1,\ldots,j2$. In particular, (18$_{j2}$), (15), (14) and Proposition 2 show that

$$\begin{aligned}|r-s|=|r_{j2}|&\leqq[(f_{j2}+e_{j2})(q-1)+2n\eta]\,q^{n-1}\\&\leqq[(eq+\eta)(q-1)+2n\eta]\,q^{n-1}\\&\leqq[eq(q-1)+\eta q+(2n-1)\eta q]\,q^{n-1}\\&=[e(q-1)+2n\eta]\,q^n,\end{aligned}$$

as asserted. □

Our next aim is to find expressions which when evaluated on the computer provide bounds for $(a+m\eta)q^n$. We treat the fixpoint case ($\varepsilon=0$) first.

Proposition 4:

If $\varepsilon=0$ then for all nonnegative numbers $a\in\mathscr{M}$ and integers $m,n\geqq 0$ we have

$$(a+m\eta)q^n\leqq\overline{a+(m+2)\eta}\quad \textit{if}\ \ m\leqq N-2. \tag{21}$$

Proof: We have $q=1$ and $a+(m+2)\eta \leqq a+\overline{(m+2)\eta}+\eta \leqq \overline{a+(m+2)\eta}+2\eta$; the computation of $m+2$ is error free by $A2$. Now subtract 2η. □

For the floating point case ($\varepsilon \neq 0$) we need a convenient bound for q^n. To find this we choose a large integer M such that

$$M \leqq \varepsilon^{-1}, \quad M \leqq N; \tag{22}$$

in most cases $M=\varepsilon^{-1}$ is already such an integer. Then we have the following estimate whose proof (by induction) is left to the reader.

Lemma:
If $1 \leqq n \leqq M-1$ *then*

$$q^n \leqq \frac{M}{M-n}. \quad \square \tag{23}$$

To discuss the quality of the bound (23) we assume that $M=\varepsilon^{-1}$. Then

$$q^n = \left(1+\frac{1}{M-1}\right)^n \geqq 1+\frac{n}{M-1} > 1+\frac{n}{M},$$

and for $n(n+1) \leqq M$, we have

$$\frac{M}{M-n} \leqq 1+\frac{n+1}{M}.$$

In many cases,

$$1+\frac{n+1}{M}$$

is the smallest machine number

$$> 1+\frac{n}{M},$$

whence (23) is excellent for $n(n+1) \leqq M$. For larger n we use

$$q^{\alpha M} = \left(1+\frac{1}{M-1}\right)^{\alpha M} \approx e^{\alpha}$$

for large M, which leads to the following table.

$\frac{N}{M}$	q^n	$\frac{M}{M-n}$
0.05	1.051	1.052
0.1	1.105	1.111
0.2	1.221	1.250
0.5	1.649	2.000

A still better upper bound than (23) would be $(2M-1+n)/(2M-1-n)$, valid for $n \leqq 2M-2$; but we use (23) because of its simplicity.

Proposition 5:
If $m+3, n+4 \leqq M$ *then for all nonnegative* $a \in \mathcal{M}$, *we have*

$$(a+m\eta)q^n \leqq \overline{(a+(m+4)\eta)(M/(M-3-n))}. \tag{24}$$

In particular

$$aq+\eta \leqq \overline{(a+5\eta)(M/(M-4))}. \tag{25}$$

Proof: Put $i=m+4$, $k=n+3$ (they are computed without error, by $A2$). Then by Proposition 1 and (23) for $n=1$ we have

$$\overline{i\eta} \geqq (i\eta-\eta)q^{-1} \geqq \eta(i-1)(M-1)/M \geqq (i-2)\eta = (m+2)\eta \tag{26}$$

whence by Proposition 1, $A2$, (26), and (23) we have

$$\overline{(a+i\eta)(M/(M-k))}\,q^3 \geqq \overline{(a+i\eta)}\,\overline{(M/(M-k))}\,q^2 - \eta q^2$$

$$\geqq (a+\overline{i\eta}-\eta)M/(M-k) - \eta q^2 \geqq (a+(m+1)\eta)M/(M-k) - \eta q^2$$

$$\geqq (a+(m+1)\eta)q^k - \eta q^2 \geqq (a+m\eta)q^k = (a+m\eta)q^{n+3}.$$

Now divide by q^3 to get (24). (25) follows then from $aq+\eta \leqq (a+\eta)q$ and (24) with $m=n=1$. □

We make the correction term (24) available in the procedure COR.

```
procedure COR (a, m, n);
comment valid only if m+3 ≦ M, n+4 ≦ M;
real a;
integer m, n;
begin a:=a+(m+4)*η;
      a:=a*(M/(M-3-n));
end;
```

We are now ready to describe the algorithms mentioned in the introduction. We restrict ourselves to the floating point case ($\varepsilon \neq 0$).

The construction of SUM exploits Propositions 2 and 5:

```
procedure SUM (x, n, s);
comment valid only if n+2 ≦ M and all x_i ≧ 0;
real array x;
integer n;
real s;
s:=0;
begin for i:=1,...,n do
      s:=s+x(i);
end;
COR (s, n-1, n-1);
end;
```

In order to get an algorithm for IPROD we apply (23) and Proposition 1 and 5 to the bound (16). We get

$$|r-s| \leqq (e(q-1)+2n\eta)q^n = (e/(M-1)+2n\eta)q^n$$
$$\leqq (\overline{(e/(M-1))}\,q+(2n+1)\eta)q^n \leqq \overline{(e/(M-1))}+(2n+1)\eta)q^{n+1}.$$

This leads to the following program (s_j is stored in r):

```
procedure IPROD (A, B, c, i, j1, j2, k, r, e);
comment valid only if j2−j1+4≦M/2;
real array A, B;
real c, r, e;
integer i, j1, j2, k;
begin real p, f;
      integer j, n;
      begin r:=c;
            p:=a[i,j1] * b[j1,k]; e:=|p|;
            r:=r−p; f:=|r|;
            for j:=j1+1,...,j2 do
            begin p:=a[ij] * b[jk]; r:=r−p;
                  e:=e+|p|; f:=f+|r|;
            end;
            e:=(f+e)/(M−1);
            n:=j2−j1+1;
            COR(e, 2 * n+1, n+1);
      end;
end;
```

To get a program for DIV we estimate $|a-cb|$ for $c=\overline{a/b}$. We have $c=(a/b)(1+\alpha)+\alpha'$ with $|\alpha| \leqq q-1 \leqq (M-1)^{-1}$, $|\alpha'| \leqq \eta$ and $\alpha\alpha'=0$. If $\alpha=0$ then

$$|a-cb| = |\alpha' b| \leqq |b|\eta \leqq \overline{|b|\eta}\cdot q+\eta$$

and if $\alpha'=0$ then

$$|a-cb| = |-a\alpha| \leqq \overline{|a|/(M-1)}\cdot q+\eta.$$

Therefore

$$|a-cb| \leqq \mathrm{Max}(\overline{|a|/(M-1)}, \overline{|b|\eta})q+\eta.$$

Combined with (25) we get the following program

```
procedure DIV(a, b, c, e);
real a, b, c, e;
begin if b=0 then
      begin c:=0; e:=|a|; end
      else
      begin c:=a/b;
            e:=Max(|a|/(M−1), |b|η);
            e:=(e+5 * η) (M/(M−4));
      end;
end;
```

Finally we mention the following lower bound for a difference; the proof is similar as for Proposition 4 and is left to the reader.

Proposition 6:
If $a, b \in \mathcal{M}$, *and*

$$d := \overline{((a-b)-5\eta)\cdot((M-4)/M)} \geqq 0$$

then $a-b \geqq d$. □

References

[1] Kahan, W., Palmer, J.: On a proposed floating-point standard. ACM SIGNUM Newsletter, Special Issue (1979), 13–21.
[2] Neumaier, A.: Strict computable error bounds for matrix inversion. Freiburger Intervall-Berichte *83/1*, 17–49 (1983).
[3] Olver, F. W. J.: A new approach to error arithmetic. SIAM J. Numer. Anal. *15*, 368–393 (1978).
[4] Olver, F. W. J.: Further developments of rp and ap error analysis. IMA J. Numer. Anal. *2*, 249–274 (1982).
[5] Sterbenz, P. H.: Floating-Point Computation. Englewood Cliffs, N. J.: Prentice-Hall 1974.
[6] Wilkinson, J. H.: Rounding errors in algebraic processes. Nat. Phys. Lab. Notes Appl. Sci. *32*. London: H. M. Stationery Office 1963.

Added in proof: Recent papers by J. Demmel (see SIAM J. Sci. Stat. Comput. *5*, 887–919 (1984) and references therein) also contain a qualitative rounding error analysis for underflow effects (higher order terms are neglected).

A. Neumaier
Institut für Angewandte Mathematik
Universität Freiburg i. Br.
D-7800 Freiburg i. Br.
Federal Republic of Germany

Printed in Austria
Druck: Paul Gerin, A-1021 Wien