

A. KIELBASIŃSKI (Warszawa)

Algorytm sumowania z poprawkami i niektóre jego zastosowania

1. Wstęp. Z obliczaniem sum o wielu składnikach spotykamy się w licznych zadaniach rachunku numerycznego, np. przy kwadraturach, sumowaniu szeregów, w zadaniach algebry liniowej. Jeśli wykonujemy sumowanie składników $v_1, v_2, v_3, \dots, v_n$ w standardowej arytmetyce zmiennopozycyjnej (fl), stosując najprostszy algorytm:

$$(1) \quad \begin{cases} s_0 := 0, \\ s_i := s_{i-1} + v_i \quad (i = 1, 2, \dots, n), \end{cases}$$

to otrzymujemy (por. [8], str. 28-29) zależności

$$s_n = \sum_{i=1}^n v_i (1 - A_i),$$

$$(1 - \rho)^{n-i+1} \leq 1 - A_i \leq (1 + \rho)^{n-i+1},$$

gdzie ρ jest wielkością charakterystyczną dla danej arytmetyki numerycznej, równą niewielkiej krotności 2^{-t} (t – ilość cyfr mantysy binarnej). Ograniczając n warunkiem

$$(2) \quad n \cdot \rho \leq 0.1,$$

Wilkinson otrzymuje (por. [8], str. 31) dla A_i oszacowania

$$(3) \quad |A_i| \leq (n-i+1) \cdot \rho_1,$$

$$\rho_1 = 1.06 \cdot \rho.$$

Obliczona suma s_n jest więc dokładną sumą składników $v_i \cdot (1 - A_i)$. Oszacowania (3) mogą budzić niepokój, gdy n jest duże. Wiemy bowiem, że błąd reprezentacji składnika v_i w danej arytmetyce nie przekracza wielkości $|v_i| \cdot \rho$, natomiast pozorne zaburzenie

$|v_i \cdot A_i|$ może być $(n-i+1)$ razy większe. Ponadto, przy ograniczonych składnikach v_i , oszacowanie błędu wytworzonego sumowaniem może rosnąć jak n^2 , podczas gdy sama suma rośnie co najwyżej tak, jak n . Można istotnie łatwo konstruować przykłady, w których pozorne zaburzenia prawie osiągają wskazane w (3) oszacowanie i wywierają wyraźny negatywny wpływ na dokładność obliczeń. Nie są to oczywiście przypadki niestabilności numerycznej algorytmu (por. [8]), lecz wynik kumulacji dużej ilości błędów. Warto zastanowić się, czy możliwe jest usunięcie tej wady prostego algorytmu sumowania.

Jedną z dróg, zapewne najlepszą, jest wykorzystanie możliwości, jaką w niektórych e.m.c. daje arytmometr z rejestrem sumacyjnym podwyższonej precyzji (por. np. [2]). Niestety nie wszystkie e.m.c. posiadają takie urządzenia i nie wszystkie języki programowania pozwalają na łatwe wykorzystanie tej możliwości – gdy istnieje. Dlatego znajdujemy w literaturze szereg propozycji poprawienia algorytmu sumowania (por. [3], [4], [9]) w oparciu o standardową arytmetykę zmiennopozycyjną.

Poniższa praca zawiera analizę algorytmu proponowanego w pracy Møllera [4] dla realizacji w arytmetyce maszyny GIER. Następnie opisujemy szereg zastosowań.

2. Algorytm Møllera. O. Møller przeanalizował w [4] kilka wersji algorytmu sumowania z poprawkami z punktu widzenia realizacji w arytmetyce zmiennopozycyjnej maszyny cyfrowej GIER. Poszukiwał przy tym algorytmu, który byłby równoważny kumulacji sumy na rejestrze sumacyjnym podwójnej precyzji. Okazało się, że algorytm spełniający to zadanie jest dość skomplikowany, natomiast uproszczone wersje tego algorytmu są równoważne sumowaniu na rejestrze podwójnej precyzji większości składników sumy, a stosunkowo nielicznych, – na rejestrze pojedynczej precyzji.

Dla usunięcia negatywnych skutków kumulacji błędów wystarczy jednak, by obliczona suma była dokładną sumą składników $v_i \cdot (1-B_i)$, z pozornymi zaburzeniami B_i nie przewyższającymi pewnej krotności ρ (niezależnej od n).

Okazuje się, że najprostszy z algorytmów Møllera posiada tę właściwość, jeśli realizujemy go w zmiennopozycyjnej arytmetyce z poprawnymi zaokrągleniami wyników działań. Zapisujemy go w pseudo-algolu (u, v, s, z, p – oznaczają tu zmienne rzeczywiste, i, n – zmienne całkowite):

```
(4)
      u := p := 0;
      for i := 1 step 1 until n do
      begin
        v := < i-ty składnik sumy >;
        s := u+v;
        p := u-s+v+p;
        u := s
      end;
      z := s+p;
```

Wartości s obliczane w pętli są oczywiście identyczne z wartościami s_i w (1).

Wartość p , obliczona równolegle z sumą s , stanowi poprawkę sumowania. Analizę algorytmu (4) przeprowadzimy osobno dla arytmetyki zmiennopozycyjnej z poprawnym zaokrągleniem sumy i różnicy (fl) oraz dla arytmetyki zmiennopozycyjnej e.m.c. GIER (GIER fl).

3. Analiza algorytmu Møllera w fl. Stosując reguły opisane w [8] (str. 16 i następne), otrzymamy dla wielkości zdefiniowanych algorytmem (4) następujące zależności:

$$\begin{aligned}
s &= (u + v) (1 - a) = u + v - e, \\
e &= s \cdot a / (1 - a), \quad |a| \leq \rho, \\
(5) \quad <p \text{ nowe}> = ((u-s) \cdot (1-b) + v) \cdot (1-c) + p \cdot (1-d) = \\
&= (e \cdot (1-b) \cdot (1-c) + v \cdot b \cdot (1-c) + p) \cdot (1-d), \\
&|b|, |c|, |d| \leq \rho.
\end{aligned}$$

Wprowadzając wskaźnik przy zmiennych i uwzględniając tożsamość $u_i \equiv s_{i-1}$, możemy ważniejsze zależności (4) i (5) zapisać w postaci:

$$\begin{aligned}
s_i &= (s_{i-1} + v_i) (1 - a_i) = s_{i-1} + v_i - e_i, \\
e_i &= s_i \cdot a_i / (1 - a_i), \\
(6) \quad p_i &= (e_i \cdot (1-b_i) \cdot (1-c_i) + v_i \cdot b_i \cdot (1-c_i) + p_i) \cdot (1-d_i) \\
&|a_i|, |b_i|, |c_i|, |d_i| \leq \rho.
\end{aligned}$$

Stąd natychmiast wynikają zależności:

$$\begin{aligned}
(7) \quad s_i &= \sum_{j=1}^i v_j \cdot (1 - A_{i,j}) = \sum_{j=1}^i (v_j - e_j), \\
p_n &= \sum_{i=1}^n (e_i \cdot (1 - E_i) + v_i \cdot V_i),
\end{aligned}$$

gdzie

$$\begin{aligned}
(8) \quad 1 - A_{ij} &= \prod_{k=j}^i (1 - a_k), \\
1 - E_i &= (1 - b_i) \cdot W_i, \\
V_i &= b_i \cdot W_i, \\
W_i &= (1 - c_i) \prod_{j=i}^n (1 - d_j).
\end{aligned}$$

Korzystając z założenia (2) zapiszemy więc dla $(1 - A_{ij})$, E_i oraz V_i oszacowania:

$$|1 - A_{ij}| \leq 1 + (i-j+1) \cdot \rho_1 \leq 1.1,$$

$$(9) \quad |E_i| \leq (n-i+3) \cdot \rho_1,$$

$$|V_i| \leq \rho \cdot (1+(n-i+2) \cdot \rho_1).$$

Wstawiając zależności (7) do ostatniej instrukcji algorytmu (4), otrzymamy:

$$(10) \quad z = (s_n + p_n) \cdot (1-h) = \sum_{i=1}^n (v_i \cdot (1+V_i) - e_i \cdot E_i) \cdot (1-h) \quad (|h| \leq \rho).$$

Sumę składników $e_i \cdot E_i$ przekształcimy osobno, korzystając z (6) i (7):

$$\begin{aligned} \sum_{i=1}^n e_i \cdot E_i &= \sum_{i=1}^n s_i \cdot a_i \cdot E_i / (1-a_i) = \sum_{i=1}^n \sum_{j=1}^i v_j \cdot (1-A_{i-1,j}) \cdot a_i \cdot E_i = \\ &= \sum_{j=1}^n v_j \sum_{i=j}^n a_i \cdot E_i \cdot (1-A_{i-1,j}). \end{aligned}$$

Zamieniając w ostatniej sumie wskaźnik i na j i vice versa, wstawiamy ją do (10), otrzymując:

$$(11) \quad z = \left(\sum_{i=1}^n v_i \cdot (1-B_i) \right) \cdot (1-h),$$

gdzie

$$B_i = -V_i + \sum_{j=i}^n a_j \cdot E_j \cdot (1-A_{j-1,i}).$$

A więc

$$(12) \quad |B_i| \leq \rho \cdot (1 + (n-i+2) \cdot \rho_1) +$$

$$+ \rho \cdot \rho_1 \sum_{j=i}^n (n-j+2) \cdot (1+(j-i) \cdot \rho_1) \leq \rho \cdot (1 + \rho_1 \cdot 0.6 (n-i+5)^2).$$

Zauważmy, że założenie (2) nie wystarcza, aby wielkości B_i miały oszacowanie niezależne od n . Należałoby przyjąć silniejszy warunek, np. postaci:

$$(13) \quad \rho \cdot (n+4)^2 \leq 0.1,$$

a nierówność (12) może być wówczas zastąpiona nierównością

$$(14) \quad |B_i| \leq \rho_1.$$

Warto zauważyć, że warunek (13) nie stanowi istotnego ograniczenia dla praktyki obliczeniowej w zakresie zadań algebry liniowej. Znacznie silniejsze ograniczenie dla n wynika z ograniczeń pamięci maszyny i czasu pracy.

Ponadto nie należy zapominać, że otrzymane oszacowania są i tak znacznie większe od prawdopodobnej wielkości błędu.

Kolejne poprawki $fl(u-s+v)$, dodawane do p w algorytmie (4), są najczęściej liczbami o „krótkich” reprezentacjach numerycznych, tzn. o rozwinięciach zawierających niezerowe cyfry na odcinku o długości $|\langle \text{cecha } u \rangle - \langle \text{cecha } v \rangle| + 1$. Jeśli rozwinięcie to odpowiada pozycjom reprezentacji p , to sumowanie poprawek odbywa się bez błędu ($d=0$).

Ponadto, w przypadku arytmetyki z poprawnym symetrycznym zaokrągleniem, szczegółowa analiza wszystkich możliwych wariantów wzajemnego położenia rozwinięć pozycyjnych u , v i s pozwala stwierdzić, że zawsze zachodzi $c=0$, że b może być różne od zera tylko wtedy, gdy $|v| > |u|$, ale wówczas d jest równe zero, i szereg innych, mniej istotnych zależności. Nie przytaczamy szczegółów tych rozważań, gdyż opis ich jest kłopotliwy, a wykorzystanie tych własności w analizie poprawiłoby oszacowanie w sposób mało istotny.

Rozważania te pozwalają jednak sądzić, że nawet w przypadkach nie spełnienia warunku (13) poprawki Møllera mogą skutecznie przeciwdziałać kumulacji błędów w procesie sumowania.

4. Analiza algorytmu Møllera w GIER — fl

1. Dla zanalizowania algorytmu Møllera w przypadku arytmetyki zmiennopozycyjnej GIER okazało się (podobnie jak w [4]) niezbędne szczegółowe rozważenie różnych możliwych przypadków wzajemnego położenia rozwinięć pozycyjnych wielkości v , s , u i p w algorytmie (4). Nie będziemy tu podawali wszystkich szczegółów tych rozważań. Zaczniemy od sformułowania najważniejszych wniosków.

- Poprawka Møllera nigdy nie pogarsza oszacowania wytworzonego błędu.
- Poprawka ta w bardzo wielu przypadkach redukuje błąd wytworzony w sumowaniu do poziomu błędu zaokrąglenia składnika v (tak jak w arytmetyce fl).
- Wyjątek stanowi sytuacja, gdy cecha sumy s jest większa niż cecha u , zaś cecha u znacznie większa niż cecha v , a ponadto, gdy ostatni bit mantysy u i odpowiedni bit mantysy v są równe 1. Wtedy poprawka Møllera nie usuwa błędów na poziomie zaokrąglenia u .
- Nawet takie błędy na poziomie zaokrąglenia u (znacznie większe, niż poziom zaokrąglenia v), można niekiedy „rozładować” na większą ilość „małych” wielkości v_i , jeśli $\sum |v_i|$ jest rzędu wielkości $|u|$.

A więc możemy zaobserwować poważną kumulację błędów w poprawionym przybliżeniu z tylko wtedy, gdy do dużego składnika dodajemy wielką ilość bardzo małych składników o różnych znakach, powodując częste wzrosty i opadanie cech sum częściowych. Wydaje się, że maksymalna kumulacja może wystąpić z mnożnikiem $n/3$ (w prostym algorytmie sumowania mnożnik ten wynosi n).

Pomijając takie zupełnie specjalnie „złośliwe” przypadki, możemy na ogół liczyć na to, że poprawione przybliżenie z jest dokładną sumą składników $v_i(1-B_i)$, z pozornymi zaburzeniami B_i , spełniającymi nierówność

$$|B_i| \leq 3 \cdot 2^{-28}.$$

(Łatwo możemy sprawdzić, że będzie tak zawsze, gdy $n < 10000$, zaś wszystkie składniki mają ten sam znak). Można przypuszczać, że również dla innych (niż GIER fl) arytmetyk zmiennopozycyjnych (bez poprawnego zaokrąglania sumy) algorytm Møllera zapewnia poprawę oszacowania błędu sumowania i prawie zawsze skutecznie przeciwdziała kumulacji błędu.

2. Aby ułatwić odtworzenie sformułowanych powyżej spostrzeżeń podajemy zestawienie grubych oszacowań błędów, wytworzonych w algorytmie Møllera w GIER fl. Wprowadzimy wielkości e , r , m zależnościami:

$$s = \text{GIER fl}(u+v) = u + v - e,$$

$$m = \text{GIER fl}(u-s+v) = u - s + v - r = e - r.$$

Widzimy, że dodanie pojedynczej poprawki m do s spowodowałoby zastąpienie błędu e błędem r . Niech (s) , (u) , (v) oznaczają cechy binarne s , u i v , zaś

$$qu = 2^{(u)-29}, \quad qv = 2^{(v)-29}, \quad M = \max((u), (v)).$$

Poniższa tablica zawiera oszacowania błędów e , r oraz poprawki $|m|$ w czterech najważniejszych przypadkach.

Określenie przypadku		$0 \leq e \leq$	$0 \leq r \leq$	$ m \leq$	$ t \leq$
$(s) > M$	$(u) > (v)$	$2qu$	$qu+qv$	$2qu$	$qu+qv$
	$(u) \leq (v)$	$2qv$	$2qv$	qv	$2qu$
$(s) \leq M$	$(u) \geq (v)$	qu	0	qu	0
	$(u) < (v)$	qv	qv	0	qu

Uzyskanie dokładniejszych i lepszych oszacowań jest możliwe jedynie na drodze bardziej szczegółowej analizy, którą tu pomijamy.

3. O. Møller proponuje w [4] dla poprawienia sumowania w arytmetyce GIER fl algorytm nieco bardziej złożony niż (4):

(15)

```

u := p := 0;
for i := 1 step 1 until n do
begin
  v := < i-ty składnik sumy >;
  s := u+v;
  p := if |u| ≥ |v| then v-(s-u)+p
        else u-(s-v)+p;
  u := s
end;
z := s+p;
```

Użycie tak obliczonej poprawki pozwoliłoby zastąpić błąd e błędem t , dla którego oszacowania podajemy w ostatniej kolumnie tabelki.

Jak widać, oszacowania dla t są lepsze niż dla r , ale nadal nie mamy pełnej gwarancji usunięcia kumulacji błędów w sumowaniu. Gwarancję taką uzyskalibyśmy, stosując trzeci algorytm Møllera, opisany w [4], znacznie bardziej złożony niż (15).

5. Zastosowanie algorytmu Møllera w kwadraturach. Rozważmy następujące zadanie: Obliczamy przybliżoną wartość całki

$$I = \int_0^1 dt/(1+t^2) = \pi/4$$

metodą prostokątów z podziałem przedziału całkowania na n równych części:

$$P_n = \sum_{i=1}^n (1/n)/(1+(i/n)^2) = \sum_{i=1}^n n/(n^2 + i^2).$$

Stosując twierdzenie o przyrostach skończonych wyrażamy w dogodny sposób błąd metody

$$I - P_n = \sum_{i=1}^n \int_{t_{i-1}}^{t_i} \left(\frac{1}{1+t^2} - \frac{1}{1+t_i^2} \right) dt = \frac{1}{n^2} \sum_{i=1}^n \frac{w_i}{(1+w_i^2)^2},$$

gdzie

$$t_i = i/n, \quad t_{i-1} \leq w_i \leq t_i.$$

Stąd otrzymujemy już łatwo oszacowanie:

$$0 < I - P_n < 0.33/n.$$

Możemy też wskazać wyrażenie asymptotyczne dla błędu, gdyż $(I - P_n) \cdot n$ jest sumą Riemannowską całki

$$\int_0^1 w/(1+w^2)^2 dw = 1/4.$$

Zatem

$$d_n = I - P_n \sim 1/(4n) \quad (n \rightarrow \infty).$$

Wielkość d_n możemy efektywnie obliczać. Poniższa tablica zawiera wielkości d_n obliczone w arytmetyce GIERfl ze zwykłym (dn1) oraz poprawianym (dn4) sumowaniem, wyrażone w jednostkach ostatniej pozycji binarnej liczby GIERfl ($\pi/4$), (tzn. pomnożone przez 2^{29}).

$n/1000$	dn1	dn4	$2^{29}/(4n)$	dn1-dn4
1	134567	134240	135623	327
2	67762	67115	67811	647
4	34832	33556	33905	1276
8	19332	16777	16952	2555
16	13523	8389	8476	5134
32	14460	4194	4238	10266
64	22649	2097	2119	20552
128	42154	1050	1059	41104

Widzimy, że w badanym zakresie n , tylko błędy przybliżeń obliczanych z poprawianym sumowaniem zachowują się w sposób zgodny z wyrażeniem asymptotycznym dla błędu metody. Błędy dn1 zachowują się w taki sposób tylko dla $n \leq 4000$, zaś dla $n > 8000$ coraz wyraźniej odbiegają od wartości wyrażenia asymptotycznego.

Dla $n > 16000$ dn1 rośnie wraz z n zamiast maleć.

Minimalna wartość błędu dn1, równa około 13130 jednostek, jest osiągana przy $n = 20800$. Warto bliżej wyjaśnić ten nieco zaskakujący wynik eksperymentu.

W obu algorytmach składniki $n/(n^2 + i^2)$ sumy P_n były obliczane z błędem nie przekraczającym jednego zaokrąglenia (dzięki zastosowaniu arytmetyki liczb całkowitych). W rezultacie sumowane były liczby zmiennopozycyjne $v_i = n/(n^2 + i^2) \cdot (1 - e_i)$, gdzie

$$0 \leq e_i \leq 2^{-28}.$$

Wpływ tych błędów na obliczone przybliżenie całki nie przekracza poziomu jednego zaokrąglenia wyniku:

$$\sum v_i/(1 - e_i) - \sum v_i \cong \sum v_i \cdot e_i = (\sum v_i) \cdot e,$$

$$0 \leq e = (\sum v_i \cdot e_i)/(\sum v_i) \leq 2^{-28}.$$

Poza tym błędem musimy jeszcze uwzględnić błąd reprezentacji liczby $\pi/4$ w arytmetyce GIERfl oraz błąd wytworzony przy odejmowaniu $(I - P_n)$. Ponieważ błędy te na ogół różnią się znakiem, więc łączny udział tych trzech „drobnych” błędów w dn nie może przekraczać dwóch jednostek 2^{-29} . A więc wartości dn1 oraz dn4 podane w tablicy przedstawiają z tą właśnie dokładnością sumę błędu metody, $(I - P_n)$, oraz błędu wytworzonego sumowaniem.

Jak wynika z rozważań poprzedniego rozdziału, pozorne zaburzenia B_i równoważne błędowi wytworzonemu w algorytmie Møllera nie przekraczają poziomu trzech zaokrągleń, a więc cały błąd wytworzony w algorytmie Møllera nie przekracza czterech jednostek przyjętej skali. Ostatecznie więc kolumna dn4 zawiera wartości błędu metody z dokładnością do sześciu jednostek. Natomiast kolumna różnic dn1-dn4 zawiera z dokładnością do pięciu jednostek wartości błędu wytworzonego w zwykłym sumowaniu. Spostrzegamy łatwo, że błąd ten pozostaje z grubsza w liniowej zależności od n :

$$\langle \text{błąd wytworzony w algorytmie (1)} \rangle = 0.32 \cdot n \cdot 2^{-29}.$$

Zbliżony wynik możemy uzyskać na drodze teoretycznej. Realizacja algorytmu (1) w arytmetyce GIERfl w tym konkretnym przypadku da się zapisać następująco:

$$s_i = s_{i-1} + v_i - g_i \cdot s_i,$$

$$0 \leq g_i \leq 2^{-28}.$$

Wielkości g_i są względnymi błędami zaokrąglenia (ściślej mówiąc: obcięcia) dokładnej sumy $s_{i-1} + v_i$ po 29-ej pozycji binarnej. Jak łatwo można sprawdzić, średnia wartość względnych błędów zaokrągleń dodatnich liczb zmiennopozycyjnych $2^c \cdot m$ w GIER fl wynosi:

$$\frac{1}{N} \int_{b=0}^{2^{-29}} \int_{m=0.5}^1 \frac{2^c b}{2^c m} db \cdot dm = \ln 2 \cdot 2^{-29} \cong 0.7 \cdot 2^{-29}$$

($2^c \cdot b$ oznacza tu błąd bezwzględny, c — cechę, m — mantysę liczby zmiennopozycyjnej,

$$N = \int_{b=0}^{2^{-29}} \int_{m=0.5}^1 db \cdot dm = 2^{-30}.$$

A więc otrzymujemy dla błędu wytworzonego w algorytmie (1) następujące wyrażenia przybliżone:

$$\sum_{i=1}^n v_i - s_n = \sum_{i=1}^n g_i \cdot s_i \cong 0.7 \cdot 2^{-29} \cdot \sum_{i=1}^n s_i \cong 0.31 \cdot n \cdot 2^{-29},$$

gdyż

$$\sum_{i=1}^n s_i \cong \sum_{i=1}^n \sum_{j=1}^i n/(n^2+j^2) \cong n \int_0^1 dz \int_0^z dt/(1+t^2) \cong n \cdot 0.44.$$

Powyższe spostrzeżenia warto porównać z wynikami analogicznego eksperymentu z szybciej zbieżnym procesem kwadratur, np. z metodą trapezów.

Tablica zawiera odpowiednie wartości.

$n/1000$	dn1	dn4	$2^{29}/(24 n^2)$	dn1-dn4
0.25	434	359	358	75
0.5	251	91	89.5	160
1	350	23	22.4	327
2	654	7	5.6	647
4	1278	2	1.4	1276
8	2555	0	0.3	2555
16	5135	1	0.1	5134

A więc i tutaj możemy zaobserwować te same zjawiska, co w przypadku metody prostokątów.

Ze względu na szybszą zbieżność metody minimalny błąd $dn1$, rzędu tylko 240 jednostek, jest osiągany już przy $n = 520$. Widzimy tu ponadto coś nowego. Również błąd $dn4$ osiąga swą minimalną wartość (równą w tym przypadku 0) dla $n = 8000$, a dla $n = 16000$ jest już większy.

Z takich obserwacji oraz z rozważań teoretycznych wynikają oczywiste wnioski.

1° Kumulacja błędów w arytmetyce GIERfl może w poważnym stopniu zniekształcić przebieg obliczeń. Szczególnie silnie może być to odczuwane w przypadku procesów wolno zbieżnych. Niezbędne jest stosowanie środków zaradczych, np. poprawianego sumowania.

2° Dla metod nieskończonych realizowanych w arytmetyce numerycznej, istnieje graniczna (maksymalna) osiągalna dokładność. Jest ona osiągana, gdy błąd metody jest tego rzędu wielkości, co błąd wytworzony. Sytuację taką możemy nazwać „punktem krytycznym” danej metody nieskończonej. Dalsze zmniejszanie błędu metody jest niecelowe. W najlepszym przypadku całkowity błąd pozostanie na tym samym poziomie (jeśli błąd wytworzony nie rośnie), a może wzrastać (jak w przypadku $dn1$).

3° Łatwo zauważyć, że powyższe wnioski dotyczą nie tylko kwadratur, lecz również wszelkich procesów dyskretyzacyjnych.

Na zakończenie tych rozważań spróbujemy odpowiedzieć na pytanie, jak można oszacować zachowanie się błędu, wytworzonego w algorytmie (1) w rozważanych obliczeniach, w przypadku realizacji w standardowej arytmetyce fl z poprawnym, symetrycznym zaokrągleniem sumowania?

Odpowiada to sytuacji, gdy

$$-2^{-t} \leq g_i \leq 2^{-t},$$

z tym, że średnia wartość dla $|g_i|$ jest $0.35 \cdot 2^{-t}$. Zgodnie ze znanymi regułami statystycznej teorii błędów możemy więc zapisać oszacowanie:

$$\left| \sum_{i=1}^n v_i - s_n \right| \cong 0.35 \cdot 2^{-t} \left(\sum_{i=1}^n s_i^2 \right)^{1/2} \cong 0.17 \cdot 2^{-t} \sqrt{n}.$$

Łatwo możemy się zorientować, że w tym przypadku błąd wytworzony w algorytmie (1) tylko w niewielkim stopniu odbije się na przebiegu obliczeń.

Kumulacja błędów w szybko zbieżnych procesach kwadratur, realizowanych w arytmetyce fl z symetrycznym zaokrągleniem, nie wydaje się więc przedstawiać poważnego problemu obliczeniowego.

6. Zastosowanie algorytmu Möllera w metodach algebry liniowej. Arytmetyka fl. Stosowanie poprawianego sumowania w obliczeniach może być celowe tylko wtedy, gdy istotna część procesu polega na obliczaniu sum lub iloczynów skalarnych wielu składników. Wiele metod stosowanych w zadaniach algebry liniowej ma tę własność. Na przykład, podstawowe etapy metody eliminacji, a więc rozkład macierzy na iloczyn macierzy trójkątnych oraz rozwiązywanie układów o macierzach trójkątnych, mogą być sprowadzone do wielokrotnych, odpowiednio powiązanych operacji obliczania iloczynu skalarnego (por. [8], str. 137, 143).

Nie wszystkie jednak algorytmy eliminacyjne wykorzystują tę własność metody. Dzieje się tak głównie dlatego, że strategia wyboru elementu głównego (szczególnie pełnego wyboru) polega na sterowaniu przebiegiem obliczeń (m.in. kolejnością eliminacji) na podstawie wartości sum częściowych obliczanych iloczynów skalarnych.

Również inne przyczyny, np. wygoda zaprogramowania niektórych etapów rachunku, skłaniają nas niekiedy do rezygnacji z obliczania w „sposób jawny” niektórych iloczynów skalarnych. Decyzja ta nie ma istotnego znaczenia, jeśli prowadzimy obliczenia w standardowej arytmetyce zmiennopozycyjnej (fl) w połączeniu z prostym algorytmem sumowania (1) dla obliczania ewentualnych sum lub iloczynów skalarnych. Sytuacja zmienia się, gdy możemy stosować algorytm Møllera, lub arytmetykę fl_2 (por. [8], str. 24).

Aby taką możliwość wykorzystać, musimy „ujawnić” iloczyny skalarne, „ukryte” w procesie obliczeniowym. Nie zawsze można to zrobić w sposób jednoznaczny i nie zawsze korzyści płynące z dokładniejszego obliczania iloczynów skalarnych uzasadniają koszty stosowania tej podwyższonej dokładności. Niemniej Wilkinson opisuje szereg procesów, w których umiejętne stosowanie arytmetyki fl_2 daje zaskakująco dobre rezultaty, (por. [8], str. 172, 214).

Dla uproszczenia sformułowań oznaczmy przez \tilde{fl} realizację obliczeń w standardowej arytmetyce zmiennopozycyjnej, ze stosowaniem algorytmu (4) przy sumowaniu iloczynów skalarnych (sum), ujawnionych w algorytmie.

Przyjmijmy zatem założenie (13), a stąd na mocy (11) i (14) zachodzi zależność:

$$(16) \quad \tilde{fl} \left(\sum_{k=1}^n a_k b_k \right) = \left(\sum_{k=1}^n a_k b_k (1 - e_k) \right) \cdot (1 - d),$$

$$|d| \leq \rho, \quad |e_k| \leq 2\rho_1.$$

W kilku następnych rozdziałach przeprowadzimy analizę niektórych typowych algorytmów algebry liniowej w celu porównania arytmetyk fl , \tilde{fl} , fl_2 .

7. Analiza algorytmu pierwiastków kwadratowych (Banachiewicza) w arytmetyce \tilde{fl}

1. Przeanalizujemy najpierw rozkład symetrycznej dodatnio określonej macierzy $A = (a_{ij})$ ($i, j = 1, 2, \dots, n$), na iloczyn macierzy $S^T \cdot S$, gdzie $S = (s_{ij})$ ($i, j = 1, 2, \dots, n$) jest macierzą trójkątną górną, tzn. $s_{ij} = 0$, gdy $i > j$ (por. [8], str. 166).

Proces obliczeniowy zapiszmy w pseudo-algolu:

```
(17)  for i := 1 step 1 until n do
      begin
        s[i,i] := sqrt(a[i,i] -  $\tilde{fl}(\sum_{k=1}^{i-1} s[k,i]^2)$ );
        for j := i+1 step 1 until n do
          s[i,j] :=  $\tilde{fl}(a[i,j] - \sum_{k=1}^{i-1} s[k,i] \times s[k,j]) / s[i,i]$ 
        end
      end
```

U w a g a. Przy obliczaniu s_{ii} nie proponujemy tu stosowania algorytmu (4) dla obliczania całej sumy pod pierwiastkiem, gdyż dzięki temu w oparciu o (16) uzyskujemy nieco lepsze oszacowanie błędu. Bardziej wnikliwa analiza algorytmu (4) pozwoliłaby jednak na uzyskanie równie dobrego oszacowania, gdybyśmy włączyli składnik a_{ii} do poprawianego sumowania (jak robimy to przy obliczaniu s_{ij}).

Stosując (16) oraz zakładając, że pierwiastek kwadratowy jest obliczony z dokładnością jednego zaokrąglenia (por. [7], rozdz. 3) otrzymamy po prostych przekształceniach zależności:

$$a_{ii} = \sum_{k=1}^i s_{ki}^2 (1 - e_i^{(k)}) = (\sum_{k=1}^i s_{ki}^2) / (1 - f_i),$$

$$|e_i^{(k)}| \leq 3(\rho + \rho^2 + \rho^3/3), \quad |f_i| \leq 3\rho_1,$$

$$a_{ij} (1 - f_{ij}) = \sum_{k=1}^i s_{ki} \cdot s_{kj} (1 - e_{ij}^{(k)}) \quad (i < j),$$

$$|e_{ij}^{(k)}| \leq 2\rho_1, \quad |f_{ij}| \leq \rho_1.$$

Możemy więc powiedzieć, że w przypadku, gdy algorytm (17) nie zostanie przerwany w wyniku pojawienia się pod pierwiastkiem liczby ≤ 0 , otrzymana macierz $S = (s_{ij})$ jest czynnikiem rozkładu $S^T \cdot S$ macierzy symetrycznej, dodatnio określonej $A - D = (a_{ij} - d_{ij})$, gdzie

$$d_{ij} = \begin{cases} a_{ii} \cdot f_i & (i = j), \\ a_{ij} \cdot f_{ij} - \sum_{k=1}^i s_{ki} s_{kj} e_{ij}^{(k)} & (i < j). \end{cases}$$

Zatem (stosując w jednym miejscu nierówność Cauchy) otrzymujemy oszacowania:

$$|d_{ii}| \leq 3\rho_1 \cdot a_{ii},$$

$$|d_{ij}| \leq \rho_1 \cdot (|a_{ij}| + 2 \sum_{k=1}^i |s_{ki} \cdot s_{kj}|) \leq \rho_1 \cdot (|a_{ij}| + 2 \sqrt{a_{ii} a_{jj}}).$$

Stąd łatwo już otrzymamy nierówności:

$$(18) \quad \begin{aligned} \|D\|_2 &\leq \rho_1 (\|A\|_2 + 2 \operatorname{spur}(A)) \leq \rho_1 (n^{1/2} + 2n) \|A\|_2, \\ \|D\|_E &\leq \rho_1 (1 + 2n^{1/2}) \|A\|_E. \end{aligned}$$

Możemy więc, podobnie jak w [6], wypowiedzieć wniosek:
Jeśli zachodzi nierówność

$$\|A\|_2 \cdot \|A^{-1}\|_2 < \rho_1^{-1} / (n^{1/2} + 2n),$$

to algorytm (17) jest wykonalny.

Zapisując ogólnie

$$\|D\|_p \leq \rho \cdot K_p \cdot \|A\|_p,$$

możemy scharakteryzować zależność K_p od n dla normy spektralnej ($\|\cdot\|_2$), normy Frobeniusa ($\|\cdot\|_F$) oraz arytmetyki fl, fl, fl₂ tablicą (por. [6]):

(19)

	fl	fl	fl ₂
K_2	n^2	n	$n^{1/2}$
K_E	$n^{3/2}$	$n^{1/2}$	$n^{1/4}$

2. Algorytm Banachiewicza sprowadza rozwiązanie układu równań liniowych

$$A\vec{x} = \vec{b}$$

o symetrycznej, dodatnio określonej macierzy, do rozwiązania dwu układów o macierzach trójkątnych:

$$S^T \vec{y} = \vec{b},$$

$$S \vec{x} = \vec{y}.$$

Stosując arytmetykę fl, otrzymamy wektory \tilde{y} oraz \tilde{x} , które spełniają dokładnie układy

$$(S - F)^T \tilde{y} = \vec{b},$$

$$(S - G) \tilde{x} = \tilde{y},$$

przy czym

$$F = (f_{ij}), \quad G = (g_{ij}),$$

$$|f_{ij}|, |g_{ij}| \leq \rho_1 \cdot 2 \cdot |s_{ij}|.$$

Zatem \tilde{x} spełnia układ równań:

$$(A - Z) \tilde{x} = \vec{b},$$

gdzie

$$Z = D + F^T S + S^T G - F^T G$$

stąd

$$\|Z\|_E \leq \rho_1 (1 + 6n^{1/2}) \|A\|_E,$$

$$\|Z\|_2 \leq \rho_1 (n^{1/2} + 6n) \|A\|_2.$$

Porównanie tego wyniku z analogicznym wynikiem analizy dla arytmetyk fl oraz fl₂ prowadzi do wniosków przedstawionych już tablicą (19).

8. Uwagi o zastosowaniu fl̃ w algorytmach eliminacyjnych

1. Zastosowanie arytmetyki fl̃ choćby tylko do rozwiązywania układów o macierzach trójkątnych, powstających z rozkładu macierzy, pozwala na poprawienie oszacowań pozornych zaburzeń, równoważnych wytworzonemu błędowi. Na przykład, w oszacowaniu (25.14) w [8], str. 153, możemy n^3 zastąpić przez $2n^2$.

Możemy również wykorzystać fl̃ (podobnie jak fl₂) (por. [5], str. 36, 37) w procesie rozkładu trójkątnego macierzy metodą eliminacji z częściowym wyborem elementu głównego. Nie poprawi to jednak oszacowania (25.8) w [8], str. 153.

Reasumując, w algorytmach tego typu (z rozwiązywaniem układów trójkątnych) oszacowania błędu dla arytmetyki fl̃ są n -krotnie lepsze, niż dla arytmetyki fl, i na ogół n -krotnie gorsze, niż dla arytmetyki fl₂.

Przykrym rozczarowaniem jest łatwy do sprawdzenia fakt, że nie można zastąpić arytmetyki fl₂ arytmetyką fl̃ przy obliczaniu wektora residualnego $\vec{r} = \vec{b} - A\vec{x}$ w procesie iteracyjnego poprawiania rozwiązania ([8], str. 172).

2. Łatwo sprawdzimy, że realizując w arytmetyce fl̃ metodę Hymana ([8], str. 209) otrzymujemy wyznacznik

$$\det[(H - E) - \mu I] \cdot (1 - \varphi),$$

gdzie

$$E = (h_{ij} \cdot e_{ij}), \quad |e_{ij}| \leq 4\rho, \quad |\varphi| \leq 5n \cdot \rho_1,$$

$$(\|E\|_{1,\infty,E} \leq 4\rho_1 \|H\|_{1,\infty,E}).$$

Oszacowanie to jest w istocie zbliżone do oszacowania (56.5) [8], str. 212, wyprowadzonego dla arytmetyki fl₂.

9. Transformacja Householdera w arytmetyce fl̃. Zastosowanie zależności (16) w analizie transformacji Householdera, opisanej w [7], str. 153-160, prowadzi do następującego wniosku:

Jeśli P oznacza stabilną transformację Householdera, zdefiniowaną warunkiem

$$P\vec{x} = \vec{e}_1 (\pm \|\vec{x}\|_2),$$

gdzie \vec{x} dany niezerowy wektor rzeczywistej przestrzeni n -wymiarowej R^n (n ograniczone warunkiem (13)), dla dowolnej macierzy prostokątnej A ($n \times m$), numerycznie wyznaczone w arytmetyce fl, fl̃ lub fl₂ przybliżenie \tilde{B} macierzy $B = P \cdot A$ spełnia nierówność

$$(20) \quad \|B - \tilde{B}\|_E \leq \rho \cdot K \cdot \|B\|_E,$$

gdzie K jest określone w tablicy

(21)

	fl	fl̃	fl ₂
K	$3.3n+20$	20.5	10

U w a g a 1. Oszacowania podane w (21) pomijają składniki rzędu $n \cdot 2^{-2t}$, do czego upoważnia nas założenie (13).

U w a g a 2. Oszacowanie $K = 10$ dla fl_2 jest nieco lepsze, niż podane w [7], str. 160 ($K = 12.36$). Wiąże się to z lepszym oszacowaniem normy $\|P - \tilde{P}\|_2$ w (40.3) str. 156 [7].

U w a g a 3. Ten sam charakter oszacowań, co w tablicy (21) otrzymujemy dla błędu symetrycznego podobieństwa Ortegi-Householdera, PAP^T , gdy $A = A^T$ (por. [7], str. 292). Wielkości K są w tym przypadku około 2.5 razy większe niż w (21).

Uzyskane oszacowania przemawiają silniej na korzyść arytmetyki fl , niż w przypadkach poprzednio omawianych metod algebry liniowej. Dla transformacji Householdera arytmetyka fl wydaje się być prawie równie dobra, jak arytmetyka fl_2 .

Powstaje naturalnie pytanie, w jakim stopniu oszacowania wyrażone w (20) i (21) są realistyczne? Spróbujemy odpowiedzieć na nie w następnym rozdziale.

10. Eksperymentalne badanie błędu transformacji Householdera

1. Eksperymentalne badanie oszacowania (20) i (21) w przypadku arytmetyki fl i fl_2 okazało się możliwe poprzez bezpośrednie modelowanie pewnej niezależnej części błędu oraz modelowanie dokładniejszego oszacowania całego błędu.

W przypadku, gdy macierze A , B , \tilde{B} są jednokolumnowe, a więc mogą być zapisane jako wektory \vec{a} , \vec{b} , $\vec{\tilde{b}}$, stwierdzamy, że błąd możemy rozbić na dwa składniki

$$(22) \quad \vec{b} - \vec{\tilde{b}} = E\vec{a} + F\vec{a},$$

przy czym normy spektralne macierzy kwadratowych E i F można wyrazić lub oszacować w sposób następujący:

$$(23) \quad \begin{aligned} \|E\|_2 &= \rho \cdot e_{df} = |g| + \sqrt{g^2 + d^2 (2z - z^2)}, \\ \|F\|_2 &\leq \rho \cdot f_{df} = c, \end{aligned}$$

$$1 \leq z \leq 2.$$

Parametr z jest stały dla danej transformacji P (wektora \vec{x}):

$$z = (|x_1| / \|\vec{x}\|_2 + 1) \in \langle 1, 2 \rangle.$$

Wielkości g , d i c wyrażają się odmiennie dla arytmetyki fl_2 i fl :

$$(24) \quad \begin{aligned} g &= \begin{cases} r_1 z + r_2 / z + r_3 + r_4 & \text{dla } fl_2, \\ (2r_9 (z-1) + 2r_4 + r_1 + r_2 + r_3) / (2z) + r_5 + r_6 + r_7 + r_8 z & \text{dla } fl, \end{cases} \\ d &= \begin{cases} r_1 + r_2 / z & \text{dla } fl_2, \\ r_8 + (2r_4 - r_1 - r_2 - r_3) / (2z) & \text{dla } fl, \end{cases} \\ c &= \begin{cases} |r_5| & \text{dla } fl_2, \\ |r_{10}| + 4|r_{11}| + 2|r_{12}| & \text{dla } fl. \end{cases} \end{aligned}$$

Wielkości r_i oznaczają albo pojedyncze błędy zaokrągleń, albo średnie ważone dużej ilości (około n) takich błędów. Wydaje się więc, że można je traktować jako niezależne zmienne losowe, spełniające nierówność:

$$|r_i| \leq \rho.$$

W eksperymencie modelowano r_i dwoma sposobami:

$$(i) \quad r = \rho \cdot (2q - 1) / (1 + q'),$$

$$(ii) \quad r = \rho \cdot (2q - 1),$$

gdzie q, q' oznaczają liczby losowe o rozkładzie jednostajnym w przedziale $< 0, 1 >$. Sposób (i) odpowiada zapewne wierniej przypadkowi, gdy r jest błędem względnym jednego zaokrąglenia. Sposób (ii) jest bardziej „pesymistyczny”, ostrożniejszy. Parametr z był również modelowany zgodnie z wzorem:

$$z = 1 + q.$$

Generując odpowiednią ilość liczb q , możemy modelować, zgodnie z (23) i (24), wielkości: e oraz $h = e + f$. Dla każdego z tak skonstruowanych przypadków prawdziwe są następujące zdania:

$$(25) \quad \begin{aligned} \bigvee \vec{a} : \quad \|E\vec{a}\|_2 &= \rho \cdot e \cdot \|\vec{a}\|_2, \\ \bigwedge \vec{a} : \quad \|\vec{b} - \tilde{b}\|_2 &\leq \rho \cdot h \cdot \|\vec{a}\|_2. \end{aligned}$$

Ponieważ $\|\vec{a}\|_2 = \|\vec{b}\|_2$, więc statystyka rozkładu wielkości e i h pozwala w pewnym stopniu wyjaśnić, czy oszacowania (20) i (21) są realistyczne.

Tablica zawiera zestawienie statystyczne 200000 przypadków, z wielkościami r_i modelowanymi według reguły (i).

proc.	\tilde{f}_1		f_2	
	$e \leq$	$h \leq$	$e \leq$	$h \leq$
50	1.6	4.1	1.4	1.7
70	2.3	5.0	2.0	2.4
80	2.8	5.5	2.5	2.8
90	3.5	6.3	3.1	3.5
95	4.1	7.0	3.6	4.0
99	5.2	8.3	4.6	5.0
99.9	6.4	9.7	5.5	5.9
99.99	7.2	10.6	6.2	6.6
100	8.0	12.0	6.8	7.9
∞	13.5	20.5	9.0	10.0

Natomiast dla 50000 przypadków modelowanych według reguły (ii) otrzymaliśmy tablicę (27).

(27)

proc.	\tilde{f}_1		f_2	
	$e \leq$	$h \leq$	$e \leq$	$h \leq$
50	2.3	5.9	2.0	2.5
70	3.3	7.1	2.9	3.4
80	4.0	7.9	3.5	4.0
90	4.9	9.0	4.4	4.9
95	5.8	9.9	5.1	5.6
99	7.3	11.6	6.3	6.9
99.9	8.9	13.4	7.4	8.0
100	11.0	16.1	8.2	9.0
∞	13.5	20.5	9.0	10.0

Jeśli można by więc ufać założeniom któregoś z tych dwu eksperymentów, to w tablicy (21) należałoby zastąpić oszacowania teoretyczne eksperymentalnymi:

	\tilde{f}_1	f_2
$K(i)$	12	8
$K(ii)$	16	9

Zależności (22) i (25) pozwalają na interpretację informacji zawartej w tablicach (26) lub (27). Na przykład, z pozycji eksperymentu (ii) (tablica (27)), możemy wypowiedzieć następujące spostrzeżenie:

— w 90 procentach przypadków dla błędu wytworzonego w f_2 zachodzi nierówność

$$\|\vec{b} - \tilde{b}\|_2 \leq \rho \cdot 4.9 \cdot \|\vec{b}\|_2,$$

— ale w 10 procentach przypadków nie można wykluczyć tego, że błąd $\vec{b} - \tilde{b}$ zawiera niezależną składową $E\vec{a}$, taką, że

$$\|E\vec{a}\|_2 \geq \rho \cdot 4.4 \cdot \|\vec{b}\|_2.$$

Wydaje się więc, że istotny sens uzyskanych wyników może być sformułowany w następujący sposób:

Oszacowanie błędu (20) i (21) dla arytmetyk \tilde{f}_1 i f_2 są „zawyżone” w tym sensie, że jest bardzo mało prawdopodobne, by wielkość

$$L = \|\vec{b} - \tilde{b}\|_2 / (\|\vec{b}\|_2 \cdot \rho)$$

przekraczała połowę wskazanej wielkości K . Jednakże oszacowania te są na tyle realistyczne, że jest już dość prawdopodobne, że L będzie nie mniejsze, niż np. $K/5$.

2. Drugi typ badań doświadczalnych dotyczył błędów wytworzonych przy dwukrotnej transformacji Householdera dowolnego $\vec{u} \in R^n$. Przy dokładnej realizacji powinniśmy otrzymać z powrotem ten sam wektor \vec{u} . Ze względu na błędy wytworzone otrzymujemy wektor \tilde{u} , różny na ogół od \vec{u} . Eksperyment polegał na uzyskaniu pewnej informacji o rozkładzie błędów względnych:

$$W = \|\vec{u} - \tilde{u}\|_2 / \|\vec{u}\|_2,$$

dla różnych n , dla arytmetyk GIERfl, GIERfl̃, GIERfl_{4/3}. (GIERfl_{4/3} powstaje z GIERfl przez dołączenie kumulacji iloczynów skalarnych na rejestrze sumacyjnym długości o 1/3 większej, niż długość mantysy).

Transformacja Householdera $P = I - 2\vec{z} \cdot \vec{z}^T / \|\vec{z}\|_2^2$ jest definiowana wektorem $\vec{z} \in R^n$.

Współrzędne wektorów \vec{u} i \vec{z} generujemy, jako liczby pseudolosowe o rozkładzie jednostajnym z przedziału $< -10, 10 >$.

Grube oszacowanie teoretyczne dla W podaje tablica:

	GIERfl	GIERfl̃	GIERfl _{4/3}
$W \leq$	$(1.2n+5)_{10-8}$	8_{10-8}	5_{10-8}

U w a g a. Błędy W wynikają tylko z błędu wytworzonego mnożeniami $P \cdot (P \cdot \vec{u})$, a nie z niedokładności samej operacji P . Omawiany eksperyment polega więc na uchwyceniu „dwukrotnie” tylko części błędu (22).

Niech $W_i^{(n)}$ oznacza wartość W dla i -tego generowanego przypadku (dla ustalonego n). Wprowadzamy wielkości:

$$M_k^{(n)} = \max_{1 \leq i \leq k} W_i^{(n)}; \quad S_k^{(n)} = \sqrt{\sum_{i=1}^k (W_i^{(n)})^2 / k}.$$

Pierwszą nazwiemy *błędem maksymalnym*, drugą – *błędem średnim*.

Eksperyment wykazał, że błąd maksymalny dla GIERfl_{4/3} jest około 2 razy mniejszy, niż dla GIERfl̃, oraz, że oba te błędy maleją przy wzroście n do poziomu: 3_{10-9} , co jest odpowiednikiem ρ w GIERfl (największa spostrzeżona wartość W dla GIERfl̃ wynosi 1.6_{10-8} przy $n = 50$). Natomiast dla GIERfl zarówno błąd maksymalny, jak i średni rosną jak \sqrt{n} .

Na przykład dla $k = 150$ uzyskano eksperymentalnie zależności

$$M_{150}^{(n)} \sim \sqrt{n} \cdot 10_{-8},$$

$$S_{150}^{(n)} \sim \sqrt{n} \cdot 0.3_{10-8}.$$

Poniższa tablica przedstawia niektóre uzyskane wyniki dla $k = 150$.

n	GIERfl		GIERfl̃	
	$M_{150}^{(n)}$	$S_{150}^{(n)}$	$M_{150}^{(n)}$	$S_{150}^{(n)}$
100	1.0_{10-7}	2.7_{10-8}	7.2_{10-9}	3.4_{10-9}
200	1.6_{10-7}	4.4_{10-8}	5.0_{10-9}	3.2_{10-9}
300	1.6_{10-7}	5.3_{10-8}	4.8_{10-9}	3.1_{10-9}
400	1.9_{10-7}	6.6_{10-8}	4.0_{10-9}	3.1_{10-9}
500	1.6_{10-7}	6.1_{10-8}	4.8_{10-9}	3.1_{10-9}
600	2.3_{10-7}	7.2_{10-8}	4.2_{10-9}	3.1_{10-9}
700	2.6_{10-7}	8.5_{10-8}	3.8_{10-9}	3.0_{10-9}
800	2.7_{10-7}	8.5_{10-8}	4.3_{10-9}	3.1_{10-9}
900	3.0_{10-7}	8.2_{10-8}	3.9_{10-9}	3.0_{10-9}
1000	2.5_{10-7}	8.2_{10-8}	3.8_{10-9}	3.0_{10-9}

3. Trzeci typ badań doświadczalnych dotyczył błędu wytworzonego w transformacji podobieństw Ortegi-Householdera. (Eksperyment ten był pomysłem M. Jankowskiego i został przez niego w znacznej części zrealizowany (por. [1])).

Symetryczne macierze testowe A różnych stopni sprowadzane były do postaci trójdzielnej transformacją Ortegi-Householdera przy użyciu arytmetyki: GIERfl, GIERfl̃, GIERfl_{4/3}. Następnie wyznaczano wartości własne otrzymanych macierzy trójdzielnych metodą bisekcji i porównywano otrzymane przybliżenia $\tilde{\lambda}_i$ ze znanymi wartościami własnymi λ_i macierzy testowych. Wyprowadzano błędy

$$d_i = \lambda_i - \tilde{\lambda}_i.$$

Błędy związane z bisekcją należy ocenić jako mniej istotne. Otrzymane eksperymentalnie wielkości $|d_i|$ dają nam więc z grubsza oszacowanie z dołu normy zaburzeń wprowadzonych przez transformację Ortegi-Householdera

$$|d_i| \leq \|\delta A\|_2.$$

Dla zaburzeń δA potrafimy podać teoretyczne oszacowania:

$$\|\delta A\|_E \leq Z \cdot \|A\|_E.$$

	GIERfl	GIERfl̃	GIERfl _{4/3}
Z	$(15n+n^2)_{10-8}$	$n \cdot 16_{10-8}$	$n \cdot 10_{10-8}$

Badania eksperymentalne w tym zakresie nie zostały jeszcze zakończone, wykazują jednak, że dokładności uzyskiwane w arytmetyce GIERfl̃ i GIERfl_{4/3} są tego samego rzędu, i są w większości przypadków (dla $30 \leq n \leq 85$) dziesięć – do stu razy lepsze niż w arytmetyce GIERfl (por. [1]).

Bibliografia

- [1] A. G ó r a j, M. J a n k o w s k i (i inni), *Oszacowanie błędu rozwiązania układu równań liniowych i zastosowanie poprawionego sumowania w algorytmach algebry liniowej*, Matematyka Stosowana I (1973), str. 43–46.
- [2] D. H u t c h i n s o n, *A final word on reducing truncation errors*, Comm. ACM 8/5 (1965), str. 262.
- [3] P. L i n z, *Accurate floating-point summation*, Comm. ACM 13/6 (1970), str. 361–362.
- [4] O. M ø l l e r, *Quasi-double-precision in floating-point addition*, BIT 5 (1965), str. 37–50, 251–255.
- [5] *Nowoczesne metody numeryczne*, Warszawa 1965.
- [6] J. H. W i l k i n s o n, *A priori error analysis of algebraic processes*, Proc. Int. Congr. Math., Moskwa 1966.
- [7] J. H. W i l k i n s o n, *The algebraic eigenvalue problem*, Oxford 1965.
- [8] J. H. W i l k i n s o n, *Błędy zaokrągleń w procesach algebraicznych*, Warszawa 1967.
- [9] J. M. W o l f e, *Reducing truncation-errors by programing*, Comm. ACM. 7/6 (1964), str. 355.

