

A. KIELBASIŃSKI (Warszawa)

Oszacowanie błędów w metodzie eliminacji¹

1. Wstęp. Wykonując numeryczne obliczenia często chcielibyśmy uzyskać jakąś informację o błędzie otrzymanego wyniku. Informacja taka nie zawsze musi mieć postać ograniczenia modułu (normy) błędu. Niekiedy wystarczy znać prawdopodobny rząd wielkości błędu. Nieujemne wielkości liczbowe, wyrażające informacje tego rodzaju nazywamy tu *oszacowaniami błędów*.

Poniżej opisany jest pewien sposób wyznaczania „na bieżąco” oszacowań błędów wytwarzanych i przeniesionych w algorytmie eliminacji z pełnym wyborem głównego elementu (w arytmetyce zmiennego przecinka).

Oszacowania takie mogą być wyznaczane z większą lub mniejszą subtelnością — odpowiednio większym lub mniejszym nakładem kosztów (ilość operacji, obciążenie pamięci maszyny).

Proponowany jest algorytm, wyznaczający oszacowania przenoszonych i wytwarzanych błędów kosztem niewielkiej krotności n^2 , lub nawet tylko n , działań (n — ilość równań). Skonstruowano i przetestowano procedurę algolową, realizującą ten algorytm.

2. Kontrola dokładności obliczeń „na bieżąco”

2.1. Dokładność obliczeń możemy kontrolować w następujący sposób:

- określamy możliwie realistyczne oszacowania błędów danych początkowych,
- w trakcie obliczeń wyznaczamy „na bieżąco” oszacowania błędów obciążających wyniki wykonywanych operacji (uwzględniając błędy przeniesione i wytwarzane).

Wraz z wynikiem końcowym otrzymujemy w ten sposób jakąś informację o błędzie tego wyniku. Informacja ta (jej znaczenie i wiarygodność) zależy od sposobu wyznaczania oszacowań.

2.2. Przyjmujemy zasadę jednorodności oszacowań:

- jeśli α jest oszacowaniem błędu d , to $|c| \cdot \alpha$ jest oszacowaniem błędu $c \cdot d$.

Jeśli α i β są oszacowaniami niezależnych błędów d i h — odpowiednio, to za oszacowanie sumy błędów, $d + h$, przyjmujemy jedną z następujących trzech wielkości:

- (a) $\alpha + \beta$,
- (b) $\sqrt{\alpha^2 + \beta^2}$,
- (c) $\max(\alpha, \beta)$.

¹Pierwsza wersja tego artykułu ukazała się w zeszycie 25 Sprawozdań Instytutu Maszyn Matematycznych i Zakładu Obliczeń Numerycznych Uniwersytetu Warszawskiego (1971).

Znaczenie wyboru reguły (a), (b) lub (c) wyjaśniamy następująco:

Jeśli za oszacowania błędów początkowych i wytworzonych przyjmiemy ograniczenia górne modułów tych błędów, to stosując regułę (a) otrzymamy ograniczenia górne błędów wyników (por. [2], [3]). Oszacowania, wyznaczane w oparciu o tę regułę, są dla dłuższych obliczeń prawie zawsze znacznie większe niż odpowiednie błędy. Będziemy nazywali je *oszacowaniami pesymistycznymi*.

Reguła (b) wiąże się z pewnymi wynikami statystycznej teorii błędów (por. [1], str. 122).

Głównymi przesłankami są tu następujące fakty:

- rozkłady błędów pomiarów oraz rozkłady sum zmiennych losowych są często bliskie rozkładowi normalnemu,
- jeśli d i h są niezależnymi zmiennymi losowymi o rozkładzie normalnym i o przeciętnej wartości zero, to również zmienna $d + h$ ma rozkład normalny o wartości przeciętnej zero.

Jeśli $P(|d| \leq \alpha) = r$, zaś $P(|h| \leq \beta) = q$, to

$$\min(r, q) \leq P(|d + h| \leq \sqrt{\alpha^2 + \beta^2}) \leq \max(r, q).$$

Oszacowania, konstruowane zgodnie z regułą (b), będziemy nazywali *oszacowaniami prawdopodobnymi*.

Reguła (c) była niejednokrotnie proponowana w postaci szczególnie dogodnej dla rachunku ręcznego (p. [6]). Posługując się nią możemy określić rząd wielkości błędów przenoszonych oraz uchwycić typowe przypadki numerycznej niestabilności w obliczeniach. Nie kontrolujemy natomiast w stopniu zadawalającym ewentualnej kumulacji błędów. Stąd możliwość otrzymania oszacowań znacznie mniejszych niż odpowiednie błędy. Możemy temu niekiedy zapobiec, przyjmując odpowiednio zwiększone oszacowania błędów początkowych i wytwarzanych. Oszacowania, konstruowane zgodnie z regułą (c), nazwiemy *oszacowaniami optymistycznymi*.

U w a g a. Opisane powyżej reguły wyznaczania oszacowań sumy błędów są szczególnymi przypadkami ogólniejszej reguły, wyrażonej wzorem:

$$(d) \quad (\alpha^p + \beta^p)^{1/p} \quad (p > 0).$$

Przypomina to znany sposób wprowadzania norm $L(p)$ w przestrzeni kartezjańskiej (por. np. [8], str. 115). Możemy prowadzić kontrolę dokładności, posługując się dowolną wartością parametru p .

Przypadki $p = 1, 2, \infty$ są jednak najłatwiejsze w realizacji praktycznej oraz posiadają prostą interpretację.

2.3. Oszacowania możemy reprezentować i obliczać posługując się mało precyzyjną arytmetyką (1–2 znaczące cyfry dziesiętne), jeśli taka realizacja jest korzystna ze względu na pracochłonność procesu obliczeniowego lub obciążenie pamięci maszyny.

Koszt realizacji w normalnej arytmetyce pełnego systemu oszacowań „na bieżąco” jest wysoki, przewyższa na ogół koszt realizacji kontrolowanego algorytmu. (Nie musi tak być, jeśli kontrolowany algorytm jest realizowany – choćby częściowo – w arytmetyce zespolonej, lub arytmetyce podwyższonej precyzji). Dlatego staramy się ograniczyć kontrolę dokładności danego algorytmu do wyznaczania oszacowań tylko tych błędów, których wielkości nie potrafimy ocenić w inny, bardziej ekonomiczny sposób.

Niekiedy możemy posłużyć się wspólnym oszacowaniem dla większej ilości błędów. Wspólne oszacowanie powinno być rzędu wielkości największego z tych błędów, może

$$(3) \quad a_{22}^{(2)}x_2 + \dots + a_{2n}^{(2)}x_n = a_{2,n+1}^{(2)},$$

$$a_{nn}^{(n)} x_n = a_{n,n+1}^{(n)},$$

który następnie rozwiązujemy, obliczając kolejno x_n, x_{n-1}, \dots, x_1 ze wzorów

$$(4) \quad x_i := \left(a_{i,n+1}^{(i)} - \sum_{j=i+1}^n a_{ij}^{(i)} x_j \right) / a_{ii}^{(i)}.$$

Przejsie od (2) do (3) jest realizowane przez wyznaczanie dla $k = 1, 2, \dots, n-1$, $i = k+1, k+2, \dots, n$, $j = k+1, k+2, \dots, n+1$ wielkości

$$(5) \quad a_{ij}^{(k+1)} := a_{ij}^{(k)} - (a_{ik}^{(k)} / a_{kk}^{(k)}) \cdot a_{kj}^{(k)}.$$

Pełny wybór elementu głównego polega na takim przenie numerowaniu (w trakcie postępowania eliminacyjnego) równań i niewiadomych, aby dla $k = 1, 2, \dots, n-1$ oraz $i, j = k, k+1, \dots, n$ była spełniona nierówność

$$(6) \quad |a_{kk}^{(k)}| \geq |a_{ij}^{(k)}|.$$

5. Błędy wytworzone i przenoszone w algorytmie eliminacji

5.1. Oznaczmy przez $A_{ij}^{(k)}$ „prawdziwe” wartości, które wystąpiłyby w algorytmie eliminacji zamiast $a_{ij}^{(k)}$, gdybyśmy rozpoczęli obliczenia z dokładnymi danymi początkowymi $A_{ij}^{(1)}$, a wszelkie operacje arytmetyczne wykonywali bezbłędnie.

Zachodzą więc równości

$$(7) \quad A_{ij}^{(k+1)} = A_{ij}^{(k)} - (A_{ik}^{(k)} / A_{kk}^{(k)}) \cdot A_{kj}^{(k)}.$$

Natomiast wzory (5) oznaczają, zgodnie z konwencją zapisu w algolu (por. [5]), operacje podstawienia na zmienne $a_{ij}^{(k+1)}$ obliczonej wartości wyrażenia, zapisanego po prawej stronie. Wiadomo (por. [8] str. 16-24), że istnieją liczby $e_{ij}^{(k)}$, $f_{ij}^{(k)}$ takie, że spełnione są zależności

$$(8) \quad a_{ij}^{(k+1)} = a_{ij}^{(k)} \cdot (1 - e_{ij}^{(k)}) - (a_{ik}^{(k)} / a_{kk}^{(k)}) \cdot a_{kj}^{(k)} \cdot (1 - f_{ij}^{(k)})$$

oraz

$$(9) \quad |e_{ij}^{(k)}| \leq \rho \quad (1.5 \rho), \quad |f_{ij}^{(k)}| \leq 3 \rho \quad (3.5 \rho).$$

Niech $d_{ij}^{(k)}$ oznacza błąd przybliżenia $A_{ij}^{(k)}$ przez $a_{ij}^{(k)}$, tzn.

$$(10) \quad A_{ij}^{(k)} = a_{ij}^{(k)} + d_{ij}^{(k)}.$$

Podstawiając zależności (10) w (7) i odejmując stronami (8), otrzymamy po przegrupowaniu

$$(11) \quad d_{ij}^{(k+1)} = (d_{ij}^{(k)} + a_{ij}^{(k)} e_{ij}^{(k)}) - \left(\frac{a_{ik}^{(k)} + d_{ik}^{(k)}}{a_{kk}^{(k)} + d_{kk}^{(k)}} d_{kj}^{(k)} + \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} a_{kj}^{(k)} \cdot f_{ij}^{(k)} \right) - \\ - \frac{a_{kj}^{(k)}}{a_{kk}^{(k)} + d_{kk}^{(k)}} \left(d_{ik}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} d_{kk}^{(k)} \right).$$

5.2. Oznaczmy przez y_j błąd przybliżenia „prawdziwego” rozwiązania X_j obliczoną wielkością x_j , tzn.

$$(12) \quad X_j = x_j + y_j.$$

„Prawdziwe” rozwiązanie spełnia równania

$$(13) \quad \sum_{j=i}^n A_{ij}^{(i)} X_j = A_{i,n+1}^{(i)} \quad (i = 1, 2, \dots, n)$$

natomiast obliczone rozwiązanie spełnia równania (por. [8] str. 37 i następne):

$$(14) \quad \sum_{j=i}^n a_{ij}^{(i)} x_j = a_{i,n+1}^{(i)} + \sum_{j=i}^n (z_j^{(i)} g_j^{(i)} + a_{ij}^{(i)} x_j h_j^{(i)}),$$

gdzie

$$(15) \quad z_j^{(i)} = \sum_{k=i+1}^j a_{ik}^{(i)} \cdot x_k,$$

zaś $g_j^{(i)}, h_j^{(i)}$ spełniają nierówności:

$$(16) \quad \begin{aligned} |g_j^{(i)}| &\leq \rho & (1.5 \rho, & \text{ gdy } j < n, \text{ } 3 \rho, \text{ gdy } j = n), \\ |h_j^{(i)}| &\leq \rho & (2.5 \rho, & \text{ gdy } j > i), \\ |h_i^{(i)}| &\leq 2\rho & (2.5 \rho). \end{aligned}$$

Zastępując w (13) $A_{ij}^{(i)}$ oraz X_j wyrażeniami podanymi w (10) i (12) i odejmując stronami (14), otrzymamy po odpowiednim przegrupowaniu

$$(17) \quad y_i = \left[d_{i,n+1}^{(i)} - \sum_{j=i}^n x_j (d_{ij}^{(i)} + a_{ij}^{(i)} h_j^{(i)}) - \right. \\ \left. - \sum_{j=i+1}^n (y_j (a_{ij}^{(i)} + d_{ij}^{(i)}) + z_j^{(i)} g_j^{(i)}) \right] / (a_{ii}^{(i)} + d_{ii}^{(i)}).$$

U w a g a. W (9) i (16) podane są oszacowania pesymistyczne. Za oszacowania można przyjąć wielkości bardziej realistyczne: $\sqrt{l} \cdot \rho$ zamiast $l \cdot \rho$.

6. Kontrola dokładności algorytmu eliminacji

6.1. Z (11) i (17) widzimy, że tylko wtedy możemy uzyskać sensowne oszacowanie błędu, gdy błąd $d_{kk}^{(k)}$ jest istotnie mniejszy, niż odpowiedni element główny $a_{kk}^{(k)}$.

Warunek ten możemy w pewnym stopniu kontrolować, sprawdzając czy aktualne oszacowanie błędu $d_{kk}^{(k)}$ jest wystarczająco mniejsze niż $|a_{kk}^{(k)}|$. Gdyby tak nie było, to możemy uznać, że układ jest osobliwy (w sensie założonych błędów).

Zakładamy więc, że

$$a_{kk}^{(k)} + d_{kk}^{(k)} \cong a_{kk}^{(k)}.$$

Dzięki temu założeniu zależności (11) i (9) pozwalają wyznaczać oszacowania (dowolnego rodzaju) błędów $d_{ij}^{(k+1)}$, gdy znane są odpowiednie oszacowania błędów $d_{ij}^{(k)}$, zaś zależności (17), (15) i (16) pozwalają wyznaczać oszacowanie błędu y_i , gdy znane są oszacowania błędów $d_{ij}^{(i)}$ oraz y_j ($j > i$).

Konstrukcja pełnego systemu takich oszacowań wymaga jednak około $2n^3$ operacji arytmetycznych oraz obciążenia pamięci maszyny przechowywaniem około n^2 oszacowań.

Tymczasem z (11) i (6) wynika natychmiast, że liczba, która jest wspólnym oszacowaniem (dla ustalonego k) błędów $d_{ij}^{(k)}$, $a_{ij}^{(k)} \cdot e_{ij}^{(k)}$, $a_{kj}^{(k)} \cdot f_{ij}^{(k)}$, jest zarazem wspólnym oszacowaniem optymistycznym błędów $d_{ij}^{(k+1)}$ ($k \leq i \leq n$, $k \leq j \leq n$).

Spróbujmy więc zastąpić indywidualne optymistyczne oszacowania błędów $d_{ij}^{(k)}$ ($k \leq i \leq n$, $k \leq j \leq n$) jednym wspólnym oszacowaniem. Pozwoli to zredukować pracochłonność kontroli dokładności do niewielkiej krotności n^2 operacji, a obciążenie pamięci maszyny – do przechowywania n oszacowań.

Musimy jednak zadbać, aby wspólne oszacowanie błędów $d_{ij}^{(k)}$ było możliwie realistyczne. Wydaje się, że można osiągnąć to w następujący sposób:

- przez wyskalowanie niewiadomych i przemnożenie równań sprowadzamy błędy współczynników układu (2) oraz poziom ich błędów reprezentacji do wspólnego przedziału $\langle -c, c \rangle$ (tzn. $|d_{ij}^{(1)}| \leq c$, $\rho |a_{ij}^{(1)}| \leq c$, $1 \leq i \leq n$, $1 \leq j \leq n$) tak, aby możliwie dużo tych błędów było tego rzędu wielkości, co ograniczenie c ;
- za wspólne oszacowanie błędów $d_{ij}^{(1)}$ przyjmujemy niewielką krotność c , np. $3c$ (wstępne „zawyżenie” oszacowania);

- wspólne oszacowanie błędów $d_{ij}^{(k)}$ powinno być nie mniejsze niż oszacowanie błędów $d_{ij}^{(k-1)}$ oraz $\rho |a_{kk}|$. Jeśli chcemy zwiększyć prawdopodobieństwo tego, by otrzymane oszacowania błędów były ich górnymi ograniczeniami, to możemy zwiększać wspólne oszacowanie błędów $d_{ij}^{(k)}$ wraz ze wzrostem k .

W przeprowadzonych eksperymentach badano między innymi przypadki wzrostu proporcjonalnego do k , oraz \sqrt{k} .

6.2. Proponowany system oszacowań jest systemem niejednolitym, oszacowania będą wyznaczane częściowo w oparciu o regułę konstrukcji oszacowań optymistycznych, częściowo w oparciu o regułę konstrukcji oszacowań prawdopodobnych. Dlatego wygodnie będzie posługiwać się kwadratami oszacowań. (Same oszacowania będą zatem pierwiastkami z tych wielkości).

Wspólne oszacowanie $\sqrt{\alpha_k}$ błędów $d_{ij}^{(k)}$ ($k \leq i \leq n, k \leq j \leq n$) wyznaczamy rekurencyjnie, zgodnie ze wzorem

$$(18) \quad \alpha_{k+1} = \max(\alpha_k \cdot (k+q)/k, 3\rho^2 (a_{k+1,k+1}^{(k+1)})^2),$$

dla odpowiednio dobranego parametru q (np. $q = 0, 1, 2$).

Dla błędów $d_{i,n+1}^{(k)}$ oraz y_i wyznaczamy indywidualne oszacowania $\sqrt{\beta_i^{(k)}}$ oraz $\sqrt{\gamma_i}$ odpowiednio.

Zbadajmy równość (11) dla $j = n+1$. Zakładając, że $\sqrt{\beta_i^{(k)}}$ jest oszacowaniem wielkości $|d_{i,n+1}^{(k)}|$ oraz $\sqrt{3} \cdot \rho \cdot |a_{i,n+1}^{(k)}|$, możemy przyjąć, że jest równocześnie oszacowaniem (optymistycznym) całego pierwszego nawiasu po prawej stronie równości (11). Wielkość $\sqrt{\beta_k^{(k)}} \cdot \sqrt{(a_{ik}^{(k)})^2 + \alpha_k / |a_{kk}^{(k)}|}$ jest przy podobnych założeniach oszacowaniem (optymistycznym) drugiego nawiasu, zaś $\sqrt{\alpha_k} \cdot |a_{k,n+1}^{(k)}| / |a_{kk}^{(k)}|$ jest oszacowaniem trzeciego nawiasu w (11). Kojarzając zasady konstrukcji oszacowań optymistycznych i prawdopodobnych przyjmiemy

$$(19) \quad \beta_i^{(k+1)} = \beta_i^{(k)} + (a_{ik}^{(k)} / a_{kk}^{(k)})^2 \cdot \beta_k^{(k)} + ((a_{k,n+1}^{(k)})^2 + \beta_k^{(k)}) \cdot \alpha_k / (a_{kk}^{(k)})^2.$$

W podobny sposób na podstawie (17) przyjmiemy zależność

$$(20) \quad \gamma_i = (\beta_i^{(i)} + \alpha_i v_i + s_i + \rho^2 t_i) / (a_{ii}^{(i)})^2,$$

gdzie

$$v_i = x_i^2 + \sum_{j=i+1}^n (x_j^2 + \gamma_j),$$

$$(21) \quad s_i = \sum_{j=i+1}^n (a_{ij}^{(i)})^2 \cdot \gamma_j,$$

$$t_i = \sum_{j=i+1}^n (z_j^{(i)})^2.$$

Możemy łatwo sprawdzić, że łączny koszt otrzymania oszacowań $\sqrt{\gamma_i}$ błędów y_i wynosi około $3n^2$ mnożeń i dzielen, $2n^2$ dodawań, n pierwiastków kwadratowych.

Działania te, przynajmniej częściowo, mogą być wykonywane z małą dokładnością.

Na przykład zamiast obliczać $\sqrt{\gamma_i}$, możemy za oszacowanie błędu y_i przyjąć liczbę 2^k , gdzie $k = \text{entier}((\text{cecha } \gamma_i) / 2)$.

Ponieważ wzrost elementów $|a_{kk}^{(k)}|$ jest na ogół bardzo mały (por. [8], str. 139), możemy przechowywać kolejno wyznaczone oszacowania α_k w jednej komórce, a przy wyznaczaniu γ_i posługiwać się wielkością α_i , odtwarzaną z α_{i+1} z relacji

$$\alpha_i = \alpha_{i+1} \cdot i / (i + q).$$

Pozwala to ograniczyć obciążenie pamięci maszyny do przechowywania tylko n oszacowań, $\beta_i^{(k)}$ lub γ_i .

6.3. Nieco ryzykując, możemy uprościć kontrolę dokładności obliczeń, pomijając $\beta_k^{(k)}$ w sumie $((a_{k,n+1}^{(k)})^2 + \beta_k^{(k)})$ w (19), γ_j w sumie $(x_j^2 + \gamma_j)$ w (21) oraz $\rho^2 t_i$ w (20).

To znaczy, możemy wyznaczać oszacowania ze wzorów

$$(19') \quad \beta_i^{(k+1)} = \beta_i^{(k)} + (a_{ik}^{(k)} / a_{kk}^{(k)})^2 \cdot \beta_k^{(k)} + (a_{k,n+1}^{(k)} / a_{kk}^{(k)})^2 \cdot \alpha_k,$$

$$(20') \quad \gamma_i = (\beta_i^{(i)} + \alpha_i \sum_{j=i}^n x_j^2 + \sum_{j=i+1}^n (a_{ij}^{(i)})^2 \cdot \gamma_j) / (a_{ii}^{(i)})^2.$$

Koszt takiego uproszczonego systemu kontroli wyraża się ilością $2n^2$ mnożeń $1.5 n^2$ dodawań, n pierwiastków kwadratowych.

6.4. Możemy osiągnąć dalsze zmniejszenie kosztów kontroli dokładności, rezygnując z indywidualnych oszacowań błędów $d_{i,n+1}^{(k)}$ oraz y_i .

Wspólne oszacowanie większej ilości błędów może być znacznie większe od części tych błędów. Aby to zrównoważyć, stosujemy wyłącznie regułę konstrukcji oszacowań optymistycznych (z pewnym „zawyżeniem” oszacowań błędów początkowych i wytworzonych). Zamiast oszacowaniem $\sqrt{\alpha_k}$ błędów $d_{ij}^{(k)}$ posłużymy się tu oszacowaniem δ_k , konstruowanym zgodnie z regułą

$$(22) \quad \delta_k = \max(\delta_{k-1}, 3\rho |a_{kk}^{(k)}|).$$

Przyjmując za φ_1 wspólne oszacowanie błędów $d_{i,n+1}^{(1)}$ ($1 \leq i \leq n$), wyznaczamy dla $k = 2, 3, \dots, n$ wspólne oszacowania φ_k błędów $d_{i,n+1}^{(i)}$ ($1 \leq i \leq k$) oraz $d_{i,n+1}^{(k)}$ ($k \leq i \leq n$) ze wzoru

$$(23) \quad \varphi_k = \max(\varphi_{k-1}, \delta_{k-1} \cdot |a_{k-1,n+1}^{(k-1)} / a_{k-1,k-1}^{(k-1)}|).$$

Następnie, równoległe z obliczaniem x_i , wyznaczamy wspólne oszacowania ψ_i błędów y_j ($i \leq j \leq n$) ze wzoru

$$(24) \quad \psi_i = \begin{cases} \max(\varphi_n, \delta_n \cdot |x_n| / |a_{nn}^{(n)}|), & i = n, \\ \max(\psi_{i+1}, \delta_n \cdot \max_{i \leq j \leq n} |x_j| / |a_{ii}^{(i)}|), & i < n. \end{cases}$$

Koszt realizacji tego systemu kontroli wynosi $2n$ mnożeń, $2n$ dzielen, $4n$ porównań. Obciążenie pamięci maszyn redukuje się do kilku komórek.

7. Eksperymentalne badanie kontroli dokładności

7.1. Algorytm eliminacji z pełnym wyborem głównego elementu wraz z proponowanymi systemami kontroli dokładności został zapisany w postaci procedur algolowych i przetestowany w systemie Gier Algol 4 [10].

Część eksperymentów powtórzono z kontrolą „czysto optymistyczną” oraz z kontrolą „częściowo pesymistyczną”.

Badano kontrolę przenoszono i wytwarzanego błędu dla dwu rodzajów równań liniowych:

- układów L o współczynnikach i rozwiązaniach losowych całkowitych wybieranych z pewnych losowych przedziałów (różnych na ogół dla różnych kolumn macierzy);
- układów S o losowych całkowitych prawych stronach zaś o macierzach $(2^{r_j} \cdot \min(i, j))$ $i, j = 1, 2, \dots, n$, gdzie r_j całkowita liczba losowa z pewnego przedziału.

Test przeprowadzono dla różnych ciągów liczb pseudolosowych o rozkładzie jednostajnym lub normalnym, dla różnych stopni ($1 < n \leq 60$). Badano oddzielnie kontrolę błędu wytwarzanego, kontrolę przenoszonych losowych zaburzeń macierzy współczynników (zadawanych na różnych poziomach) oraz kontrolę podobnych zaburzeń prawych stron. Ogółem przebadano blisko tysiąc indywidualnych przypadków.

U w a g a. Omawiany tu materiał doświadczalny nie obejmuje przypadków w sposób oczywisty „niekorzystnych” dla danego systemu kontroli. Łatwo np. przewidzieć, że możemy otrzymać znacznie „zawyżone” oszacowanie błędu wytworzonego, jeśli błędy reprezentacji elementów jednej choćby kolumny (lub wiersza) macierzy współczynników są znacznie mniejsze, niż wspólne oszacowanie takich błędów dla całej macierzy. Można wskazać i inne analogiczne przykłady.

7.2. W testowanych układach znane były dokładne rozwiązania, mogliśmy więc porównać błędy y_i z otrzymanymi oszacowaniami E_i . Za „wskaźnik dobroci” oszacowania przyjęto wielkość

$$w_i = \log_{10}(y_i / E_i).$$

Oszacowanie jest tym lepsze, im wskaźnik ten jest bliższy zera.

Okazało się, że dla większych n ($n \geq 10$) wartości w_1, w_2, \dots, w_n mają rozkład zbliżony do rozkładu normalnego.

Stwierdzono, że w większości przypadków wartość średnia $w = \sum w_i / n$ leży w przedziale $< -2, 0 >$, zaś odchylenie standardowe $\sigma = \sqrt{\sum_{i=1}^n (w_i - \bar{w})^2 / n}$ jest bliskie jedności (często mniejsze niż 1).

Jedynie w przypadku układów L przy dominującym zaburzeniu prawych stron wartość \bar{w} silnie maleje przy wzroście n . Na przykład dla $n \cong 50$ wartość \bar{w} niekiedy spada do -3 , lub nawet niżej. Oszacowania są więc w tym przypadku „silnie zawyżone”. Wydaje się, że mamy tu do czynienia ze szczególną korelacją zaburzeń prawych stron z macierzą układu, gdyż zastosowana w tych przypadkach kontrola „czysto optymistyczna” daje wyniki wyraźnie lepsze ($\bar{w} \cong -1$, $\sigma \cong 1.5$).

Kontrola czysto optymistyczna daje jednak w większości pozostałych przypadków oszacowania zbyt zaniżone (dla dużych n), zaś kontrola częściowo pesymistyczna daje z reguły oszacowania zbyt wielkie (por. [7]).

7.3. Badania eksperymentalne wykazują więc pewne zalety i wady badanego systemu kontroli. Na ich podstawie dochodzimy do następujących wniosków:

- znaczna część oszacowań jest „zawyżona” (10 lub nawet 100 krotnie);
- zdarzają się (raczej przy dużych n) przypadki szczególnej „korelacji” błędów, charakteryzujące się silnym zawyżeniem wszystkich oszacowań;
- można spokojnie posługiwać się systemem uproszczonym oszacowań, tzn. wzorami (18), (19'), (20'), (21) zamiast wzorami (18)–(21);
- oszacowania w badanych przypadkach w małym stopniu zmieniały się przy zmianie parametru q ($= 0, 1, 2$) we wzorze (18).

W procedurze Gauss Control przyjęto więc $q = 1$, oraz uproszczone oszacowania.

8. Procedura Gauss Control

real procedure Gauss Control ($n, m, A, al, E, eps, exit$);

value n, m, eps ;

integer n, m ; real al, eps ; array A, E ; label $exit$;

comment Gauss Control rozwiązuje układy równań liniowych metodą eliminacji z pełnym wyborem elementu głównego, wyznaczając równocześnie prawdopodobne oszacowania błędów otrzymanych rozwiązań. Gauss Control stanowi przeróbkę procedury Det Gauss (p. [9]). Parametry:

- | | |
|-------|---|
| n | – wartość całkowita, ilość równań (nieznanymi) w układzie. |
| m | – wartość całkowita, ilość układów. |
| A | – tablica rzeczywista dwuwymiarowa o zakresie wskaźników $[1:n, 1:n+m]$ zawiera na wejściu w $A [1:n, 1:n]$ macierz współczynników (wspólną dla wszystkich układów) zaś w $A [1:n, n+1:n+m]$ wyrazy wolne (prawe strony) układów. Na wyjściu w $A [1:n, n+1:n+m]$ znajdują się rozwiązania odpowiednich układów równań. |
| al | – zmienna rzeczywista. Na wejściu zawiera wspólne oszacowanie błędów współczynników (jeśli błędy te są co najwyżej błędami numerycznej reprezentacji, to możemy położyć $al = 0$). Na wyjściu al zawiera oszacowanie błędu względnego obliczonego wyznacznika macierzy A . |
| E | – tablica rzeczywista dwuwymiarowa o zakresie wskaźników $[1:n, n+1:n+m]$, zawiera na wejściu nieujemne oszacowania błędów odpowiednich elementów prawych stron (dla błędów reprezentacji numerycznej możemy podać oszacowanie $E [i, j] = 0$). Na wyjściu E zawiera oszacowanie błędów odpowiednich rozwiązań. |
| eps | – wartość rzeczywista, oszacowanie błędu względnego reprezentacji numerycznej w arytmetyce zmiennoprzecinkowej. |

exit — wyrażenie mianujące, określające etykietę, od której ma być kontynuowana realizacja programu w przypadku numerycznej osobliwości układu (wyjście alarmowe).

Gauss Control — przy wyjściu obliczona wartość wyznacznika macierzy układu;

```

begin
  integer i,j,k,i0,j0,r;
  real u,w,s,z,t,d;
  integer array perm [1:n];
  array norm [1:n];
  m := n + m;
  d := 1;
  eps := eps ↑ 2;
  al := al ↑ 2;
  if al = 0 then
    begin
      for j := 1 step 1 until n do
        begin s := 0;
          for i := 1 step 1 until n do
            s := s + A [i,j] ↑ 2;
            if s = 0 then go to exit;
            if s > 1 ∨ s < .25 then
              begin
                s := norm [j] := 2 ↑ (−entier(ln(s)/1.3863 + 1));
                for i := 1 step 1 until n do
                  A [i,j] := A [i,j] × s .
                end else norm [j] := 1
              end
            end
          end j;
        for i := 1 step 1 until n do
          begin s := 0;
            for j := 1 step 1 until n do
              s := s + A [i,j] ↑ 2;
              if s = 0 then go to exit;
              s := 2 ↑ (−entier(ln(s)/1.3863 + 1)); d := d/s;
              if s > 1 ∨ s < .25 then
                for j := 1 step 1 until m do
                  begin A [i,j] := A [i,j] × s;
                    if j > n then E [i,j] := E [i,j] × s
                  end
                end
              end
            end j
          end i
        end al = 0 else
        for i := 1 step 1 until n do norm [i] := 1;
        for k := 1 step 1 until n do
          begin s := 0; r := if k = 1 then m else n;
            for i := k step 1 until n do
              for j := k step 1 until r do
                begin t := A [i,j] ↑ 2;
                  if j > n then
                    begin z := E [i,j] ↑ 2;

```

```

      E [i,j] := if z < t × eps then t × eps else z
    end else
      if t > s then
        begin s := t; i0 := i; j0 := j end
      end;
    t := s × eps; al := al × (k + 1)/k;
    if al < t then al := t;
    if s ≤ al then go to exit;
    if i0 > k then
      begin d := -d;
        for j := k step 1 until m do
          begin s := A [k,j];
            A [k,j] := A [i0,j]; A [i0,j] := s;
            if j > n then
              begin s := E [k,j];
                E [k,j] := E [i0,j]; E [i0,j] := s
              end
            end
          end;
        if j0 > k then
          begin d := -d;
            for i := k step 1 until n do
              begin s := A [i,k];
                A [i,k] := A [i,j0]; A [i,j0] := s
              end
            end;
          s := A [k,k]; d := d × s; perm [k] := j0; w := al/s ↑ 2;
          for i := k + 1 step 1 until n do
            begin t := A [i,k]/s; z := t ↑ 2;
              for j := k + 1 step 1 until m do
                begin u := A [k,j]; A [i,j] := A [i,j] - t × u;
                  if j > n then E [i,j] := E [i,j] + z × E [k,j] + u × u × w
                end j
              end i
            end k;
          for k := n + 1 step 1 until m do
            begin t := 0; u := al;
              for i := nstep-1 until 1 do
                begin r := perm [i]; s := z := 0;
                  for j := i + 1 step 1 until n do
                    begin w := A [i,j]; z := z - w × A [j,k];
                      s := s + w ↑ 2 × E [j,k]
                    end;
                  w := A [i,i]; z := (z + A [i,k])/w; t := t + z × z;
                  s := (E [i,k] + u × t + s)/w ↑ 2; u := u × (i - 1)/i;
                  if r > i then
                    begin E [i,k] := E [r,k]; A [i,k] := A [r,k] end;
                    E [r,k] := s; A [r,k] := z
                  end i
                end k;
              for i := 1 step 1 until n do
                begin u := norm [i]; d := d/u;

```



```
for j := n + 1 step 1 until m do
begin A [i,j] := A [i,j] × u;
      E [i,j] := sqrt (E [i,j]) × u
end
end i;
Gauss Control := d;
al := sqrt(al)/abs(A[n,n])
end GC;
```

Bibliografia

- [1] M. Fisz, *Rachunek prawdopodobieństwa*, Warszawa 1954.
 - [2] J. Łukasiewicz, M. Warmus, *Metody numeryczne i graficzne* t. 1. Warszawa 1956.
 - [3] R. E. Moore, *Interval Analysis*, Prentice Hall, 1966.
 - [4] W. Pankiewicz, *Kontrolowany algorytm...*, Sprawozdanie IMM i ZON UW (20), 1969.
 - [5] S. Paszkowski, *Język Algol 60*, Warszawa 1965.
 - [6] М. Н. Швец, *О приближенных числах*, Киев 1968.
 - [7] J. K. Westlake, *A handbook of numerical matrix inversion*, John Wiley, 1968.
 - [8] J. H. Wilkinson, *Błędy zaokrągleń w procesach algebraicznych*, Warszawa 1967.
 - [9] J. Zachariassen, *Linear Equations*, GIER System Library, no 162, Copenhagen 1963.
 - [10] *A manual of Gier Algol 4*, Regnecentralen, Copenhagen 1967.
-

