

**Paweł GÓRKA**

POLITECHNIKA RZESZOWSKA, KATEDRA PODSTAW ELEKTRONIKI

## Kontrola dokładności numerycznego przetwarzania danych pomiarowych

Dr inż. Paweł GÓRKA

Absolwent (1992) Wydziału Elektrycznego Politechniki Rzeszowskiej. Stopień doktora nauk technicznych uzyskał w 2003 roku. Pracuje jako adiunkt w Katedrze Podstaw Elektroniki Politechniki Rzeszowskiej. Zajmuje się zagadnieniami kontroli dokładności obliczeń numerycznych i symulacjami numerycznymi.



e-mail: pgorka@prz.rzeszow.pl

### Streszczenie

W artykule przedstawiono wybrane metody automatycznej kontroli dokładności obliczeń w procesie przetwarzania danych pomiarowych. Metody te powinny – w założeniu – uwzględniać dokładność wyników pomiarów, jak i błędy numeryczne. Najwięcej uwagi poświęcono omówieniu możliwości zastosowania arytmetyki przedziałowej jako najbardziej uniwersalnej metody kontroli dokładności obliczeń. Przedstawiono zasady jej stosowania, zalety jak i uwagi dotyczące ominięcia jej mankamentów.

**Słowa kluczowe:** arytmetyka przedziałowa, metody numeryczne, analiza błędów, kontrola dokładności obliczeń, przetwarzanie danych pomiarowych, błędy obliczeń, programowanie.

### Calculation accuracy check in computer processing of measurement data

#### Abstract

The paper presents some methods of the automatic accuracy check of calculations performed during computer processing of measurement data. The mentioned methods should take into account the measurement data accuracy and numerical errors. The paper discusses mainly the interval arithmetic method which appears as the most universal one. The basis of the method, its advantages and possibility to avoid some problems which can be connected with the use of interval arithmetic are presented as well.

**Keywords:** interval arithmetic, interval computations, numerical methods, error analysis, calculation accuracy check, measurement data processing, calculation error, programming.

### 1. Wstęp

Jednym z zasadniczych zagadnień metrologii jest ocena dokładności pomiarów z użyciem nowoczesnych środków modelowania tak obwodów pomiarowych, jak i mierzonych obiektów w postaci obwodów elektrycznych i elektronicznych. Zastosowanie modelowania komputerowego w procesie projektowania obwodów pozwala znacznie skrócić czas tworzenia projektu, uniknąć kosztów konstrukcji układów prototypowych oraz uniezależnić się od błędów przypadkowych w pomiarach. Konieczna jest minimalizacja błędów których źródłem są dane wejściowe jak i same metody obliczeniowe. Niezbędne jest też szacowanie dokładności modelowania obwodu. W każdym przypadku interesujące jest, w jakim stopniu wyniki modelowania są zbliżone z rzeczywistością.

Kontrola dokładności obliczeń jest jeszcze istotniejsza, gdy wyniki pomiarów są danymi wejściowymi do obliczeń numerycznych związanych np. z symulacjami rzeczywistych obiektów. Oprócz błędów wyników pomiarowych często trzeba uwzględnić tolerancje parametrów elementów projektowanego i symulowanego obiektu. W takiej sytuacji, w trakcie złożonych obliczeń błędy mogą rosnąć. Jest to tzw. znoszenie się składników, typowe dla operacji odejmowania, zwłaszcza liczb o zbliżonych wartościach [1]. W szacowaniu dokładności obliczeń często należy też uwzględnić błędy numeryczne wynikające z ograniczonej dokładności mantysy liczb zmiennopozycyjnych [2, 3].

### 2. Przenoszenie błędów

Powszechnie stosowanym sposobem szacowania błędów wyników operacji matematycznych przeprowadzanych na operandach obciążonych błędami jest reguła przenoszenia błędu. Jest ona dokładnie omawiana i dowodzona w literaturze (np. [1] i [4]) więc tu zostanie tylko przypomniana.

Jeśli określone są wartości  $x_i$  i związane z nimi błędy bezwzględne  $\Delta x_i$ , które są niezależne i przypadkowe, oraz funkcja

$$y = f(x_1, x_2, x_1 \dots x_n) \quad (1)$$

to błąd jej wyniku obliczany jest jako:

$$\Delta y \approx \sqrt{\sum_{i=1}^n \left( \left| \frac{\partial y}{\partial x_i} \right| \Delta x_i \right)^2} \quad (2)$$

Należy zwrócić uwagę na równość w przybliżeniu w (2), pomijaną w niektórych publikacjach.

W oparciu o powyższą regułę została opracowana „arytmetyka dokładnościowa” [5, 6]. Miara dokładności  $Dx$  liczby  $x$  zdefiniowana została jako

$$Dx = \lg \left| \frac{x}{\Delta x} \right| \quad (3)$$

Należy zaznaczyć, że choć zdefiniowaną tu miarę dokładności można interpretować jako liczbę dokładnych (pewnych) pozycji w zapisie dziesiętnym wartości  $x$ , to – jak wynika z wzoru (3) – jest to liczba rzeczywista, a nie całkowita. Miara dokładności może być też ujemna, gdy np. wynik odejmowania będzie mniejszy niż błąd bezwzględny tego odejmowania (wspomniane już znoszenie się składników).

W oparciu o (2) oraz zdefiniowaną miarę dokładności liczby (3) opracowane zostały wzory wykonywania obliczeń na liczbach, dla których określona jest ich dokładność. Zasady wykonywania obliczeń arytmetycznych na takich liczbach zostały nazwane „arytmetyką dokładnościową”. Wzory arytmetyki dokładnościowej dla czterech podstawowych operacji arytmetycznych:

– dodawanie i odejmowanie (wspólny wzór na dokładność)

$$c = a + b, c = a - b, Dc = Da + Db + \lg |c| - \frac{1}{2} \lg (a^2 10^{2Db} + b^2 10^{2Da}) \quad (4)$$

– mnożenie i dzielenie (wspólny wzór na dokładność)

$$c = a \cdot b, c = a / b, Dc = Da + Db - \frac{1}{2} \lg (10^{2Da} + 10^{2Db}) \quad (5)$$

Obydwie wymienione metody, tj. reguła przenoszenia błędów i arytmetyka dokładnościowa, mają zastosowanie jeśli błędy są małe i podlegają rozkładowi normalnemu. Warunki te są spełnione, jeśli operuje się bezpośrednio na wartościach pomiarowych np. obliczając sumę kilku zmierzonych długości lub moc elektryczną na podstawie pomiaru napięcia i prądu. Jeżeli obliczenia są bardziej złożone, trudno zachować warunek małej wartości błędów pośrednich – może wystąpić znoszenie składników, a gdy używane są funkcje nieliniowe – również warunek rozkładu normalnego nie będzie zachowany. Wobec zagadnienia poruszanego w niniejszym artykule – numerycznego przetwarzania danych pomiarowych – powyższe metody należy uznać za niewłaściwe do kontroli dokładności obliczeń. Wymienionych wad nie ma, omawiana dalej, metoda analizy przedziałowej.

### 3. Analiza przedziałowa

Idea arytmetyki przedziałowej i jej zasady zostały sformułowane przez R. E. Moore w latach sześćdziesiątych [7]. Znalazła ona

zastosowanie w wielu dziedzinach [8]. Metoda jest intensywnie rozwijana, pojawiło się wiele publikacji zarówno książkowych jak i artykułów, a obszerny spis literatury zebrany przez J. Garloffa znajduje się w pracach [9] i [10]. Wiele komentarzy i informacji dotyczących tematyki można znaleźć na stronie internetowej [11].

Główne założenie arytmetyki przedziałowej sprowadza się do tego, by parametr, którego wartości nie są znane dokładnie, opisywać nie liczbą, a najmniejszym możliwym przedziałem liczb, w którym ten parametr może przyjmować wartości. Przy tym w całym przedziale przyjmuje się jednostajny rozkład prawdopodobieństwa.

W dalszej części artykułu liczby przedziałowe zapisane będą czcionką pochyloną, pogrubioną, a same przedziały – w nawiasach kwadratowych. Tak więc liczbę przedziałową  $x$  definiuje się jako

$$x = [\underline{x}, \bar{x}] \quad (6)$$

gdzie:

$$\underline{x}, \bar{x} \text{ – odpowiednio dolna i górna granica przedziału} \quad (7)$$

W literaturze najczęściej zdefiniowane są podstawowe operacje arytmetyczne na przedziałach:

$$x + y = [\underline{x} + \underline{y}, \bar{x} + \bar{y}] \quad (8)$$

$$x - y = [\underline{x} - \bar{y}, \bar{x} - \underline{y}] \quad (9)$$

$$x \cdot y = [\min\{\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}\}, \max\{\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}\}] \quad (10)$$

$$\frac{1}{x} = \left[ \frac{1}{\bar{x}}, \frac{1}{\underline{x}} \right], \text{ gdy } \underline{x} > 0, \text{ lub } \bar{x} < 0 \quad (11)$$

$$x \div y = x \cdot \frac{1}{y} \quad (12)$$

Dla innych operacji należy najczęściej obliczyć wyniki wszystkich możliwych kombinacji dolnych i górnych granic przedziałów operandów, wybierając najmniejszą i największą z uzyskanych wartości, jako granice przedziału wyniku. Ponieważ powoduje to zwielokrotnienie działań w porównaniu z tymi samymi operacjami dla liczb rzeczywistych, należy w miarę możliwości optymalizować algorytmy, eliminując kombinacje, które na pewno nie dadzą w wyniku skrajnych wartości. Taką eliminację widać np. w zależnościach (8) i (9).

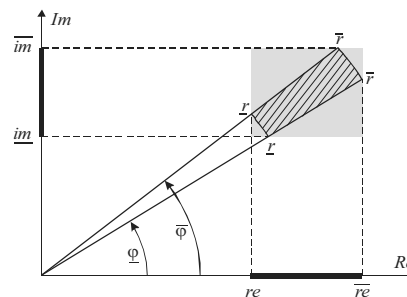
#### 4. Problemy analizy przedziałowej

Oprócz wspomnianego wyżej zwielokrotnienia operacji, duży wpływ na koszt czasowy obliczeń ma użycie funkcji nieliniowych, zwłaszcza niemonotonicznych, których argumentami są liczby przedziałowe. W takim wypadku konieczne jest znalezienie ekstremów lokalnych w przedziale określonym przez argument. Dla niektórych funkcji może być to proste i sprowadza się do sprawdzenia przedziału argumentu (np.  $f(x)=\sin(x)$ ) lub obliczenia ekstremum z ogólnie znanych wzorów (np.  $f(x)=ax^2+bx+c$ ). W wielu wypadkach może okazać się konieczne różniczkowanie funkcji. Zawsze jednak ważna jest optymalizacja algorytmu celem minimalizacji kosztu czasowego.

Oprócz czasochłonności obliczeń, istotnym problemem analizy przedziałowej jest możliwość łatwego doprowadzenia, przez niewłaściwą konstrukcję algorytmu, do sytuacji gdzie nastąpi przeszacowanie granic przedziału, to znaczy będzie on szerszy niż rzeczywistość konieczny do zawarcia wszystkich możliwych rozwiązań zadania. Jako przykład może posłużyć przekształcenie liczby zespolonej z postaci wykładniczej (lub trygonometrycznej) na arytmetyczną, gdy w tej pierwszej zarówno moduł jak i argument (lub tylko jedno z nich) określone są przedziałowo:

$$x = [\underline{r}, \bar{r}] e^{-j[\underline{\varphi}, \bar{\varphi}]} \Rightarrow x' = [\underline{re}, \bar{re}] + j[\underline{im}, \bar{im}] \quad (13)$$

Przekształcenie powyższe ilustruje rys. 1.



Rys. 1. Przeszacowanie przekształcenia postaci przedziałowej liczby zespolonej  
Fig. 1. Overestimation in form conversion of interval complex number

Jak widać, takie przekształcenie (odwrotne również) powoduje przeszacowanie przedziałów – obszar wyznaczony przez przedział liczby  $x'$  jest większy niż przedział liczby  $x$ . Iloraz powierzchni tych przedziałów może być miarą przeszacowania. Należy zauważyć, że przeszacowanie jest największe dla  $\varphi$  i  $\bar{\varphi}$  bliskich

$$\varphi = n \frac{\pi}{2} + \frac{\pi}{4}, \text{ gdzie } n \in \mathbb{Z} \quad (14)$$

i tym większe, im mniejszy jest przedział argumentu  $\varphi$ , a więc im dokładniej jest on określony. Najmniejsze przeszacowanie ma miejsce gdy

$$\varphi = n \frac{\pi}{2} \quad (15)$$

Należy tak zaprojektować algorytm obliczeń, by podobne konwersje nie występowały lub były jak najrzadsze – np. przez zmianę kolejności wykonywania działań oraz przekształcenia zależności matematycznych.

Sytuacją, w której może dojść do przeszacowań, są złożone obliczenia, w których te same zmienne przedziałowe występują wielokrotnie. Konieczne jest wtedy uwzględnienie faktu, że zmienna przedziałowa nie może w ciągu obliczeń (w kolejnych krokach algorytmu) przyjmować różnych wartości z przedziału. Oznacza to, że kolejne jej wystąpienia nie mogą być traktowane jako niezależne liczby przedziałowe. Konsekwencje tego zastrzeżenia zostaną przedstawione w przykładach.

Oczywista dla liczb rzeczywistych zależność  $x^2 = x \cdot x$  nie może być bezkrytycznie wykorzystana wraz z (10) do obliczenia kwadratu liczby przedziałowej. Wyjaśnia to przykład. Załóżmy, że

$$x = [-2, 3] \quad (16)$$

wtedy obliczając  $x^2$  z zależności (10) otrzymamy liczby

$$\underline{x} \cdot \underline{x} = -2 \cdot -2 = 4 \quad (17)$$

$$\underline{x} \cdot \bar{x} = -2 \cdot 3 = -6 \quad (18)$$

$$\bar{x} \cdot \bar{x} = 3 \cdot 3 = 9 \quad (19)$$

Przyjmując skrajne z powyższych wartości otrzymujemy

$$x^2 = [-6, 9] \quad (20)$$

Taki przedział jest znacznie przeszacowany, ponieważ liczba przedziałowa nie może przyjąć jednocześnie różnych wartości z przedziału, więc wynik (18) nie ma sensu. Ponadto wynik taki jak w (20) nie ma sensu z drugiego powodu – kwadrat liczby przedziałowej (przedział liczb rzeczywistych) mógłby zawierać wartości ujemne. Widać stąd, że do obliczenia kwadratu przedziału należy użyć zależności:

$$x^2 = [\min\{\underline{x}\underline{x}, \overline{x}\overline{x}\}, \max\{\underline{x}\underline{x}, \overline{x}\overline{x}\}] \quad (21)$$

a prawidłowym wynikiem jest przedział trzykrotnie mniejszej szerokości, zawierający wyłącznie dodatnie liczby:

$$x^2 = [4, 9] \quad (22)$$

Kolejny przykład również związany jest z kilkukrotnym występowaniem tej samej zmiennej w ciągu obliczeń, jednak ukazuje inne źródło potencjalnego przeszacowania wyniku końcowego. Załóżmy, że mamy zmienną przedziałową

$$x = [2, 3] \quad (23)$$

i wykonujemy ciąg obliczeń:

$$y = \frac{1}{x} \quad z = x + y \quad (24)$$

Ponieważ

$$y = \frac{1}{x} = \left[ \frac{1}{3}, \frac{1}{2} \right] \quad (25)$$

widać że nastąpiła inwersja granic przedziałów tzn.

$$\underline{x} \rightarrow \overline{y}, \quad \overline{x} \rightarrow \underline{y} \quad (26)$$

Jeśli tej zamiany nie weźmiemy pod uwagę, to obliczając końcowe wyrażenie na zmienną  $z$  według (8) otrzymamy

$$z = x + y = \left[ \underline{x} + \underline{y}, \overline{x} + \overline{y} \right] = \left[ \underline{x} + \frac{1}{\underline{x}}, \overline{x} + \frac{1}{\underline{x}} \right] = [2.33, 3.5] \quad (27)$$

Do obliczenia każdej z granic zmiennej  $z$  użyto więc obu granic zmiennej  $x$ , co oczywiście – jak zostało już wyjaśnione – nie ma sensu, ponieważ zmienna nie może przyjmować jednocześnie różnych wartości. Prawidłowo obliczonym wynikiem jest

$$z = x + y = \left[ \underline{x} + \overline{y}, \overline{x} + \underline{y} \right] = \left[ \underline{x} + \frac{1}{\overline{x}}, \overline{x} + \frac{1}{\underline{x}} \right] = [2.5, 3.33] \quad (28)$$

Jak widać, użycie błędnych zależności (27) spowodowało znaczne przeszacowanie wyniku.

Powyższy przykład jest banalny, jednak ukazuje istotę problemu. W złożonych obliczeniach może wystąpić wiele zmiennych przedziałowych i każda może zostać wielokrotnie użyta. Oczywiście staje się konieczność kontroli nad tym, której z granic przedziału danej zmiennej należy użyć do obliczenia odpowiednio górnej i dolnej granicy wartości wyrażenia. Rozwiązać to można przez powiązanie z pośrednimi wynikami obliczeń wektora o wartościach zero-jedynkowych zawierającego informację, która granica przedziału zmiennych początkowych została użyta do obliczenia górnej/dolnej granicy w danym etapie. Wyjaśnia to poniższy, ogólny przykład.

$$a = [a, \overline{a}], b = [b, \overline{b}], c = [c, \overline{c}] - \text{zmiennne początkowe} \quad (29)$$

$$x = f(a, b, c) - \text{funkcja zmiennych przedziałowych} \quad (30)$$

$$p\overline{x} = \begin{Bmatrix} 1, 0, 1 \\ \overline{a}, \overline{b}, \overline{c} \end{Bmatrix} - \text{wektor pomocniczy} \quad (31)$$

gdzie „1” oznacza, że  $\overline{x}$  obliczono z górnej granicy odpowiedniej zmiennej początkowej („0” – z dolnej). Jeśli następnie liczymy

$$y = f(x, b, \dots) \quad (32)$$

to, na podstawie (31), do obliczenia  $\overline{y}$  należy użyć  $\overline{x}$  i  $\underline{b}$ , a nie  $\underline{x}$  i  $\overline{b}$ , ponieważ  $\overline{x}$  obliczono z  $\underline{b}$  („0” na drugiej pozycji  $p\overline{x}$ ).

W wielokrotnych obliczeniach z wielokrotnym użyciem zmiennych początkowych może być trudne lub niemożliwe przewidzenie wpływu wyboru kombinacji progów (górnny/dolny) zmiennych początkowych na szerokość przedziału wyniku końcowego. Konsekwencją tego jest konieczność obliczania progów wyników pośrednich na podstawie wszystkich kombinacji progów zmiennych początkowych. Jeśli dane jest

$$a = [a, \overline{a}], b = [b, \overline{b}] \text{ i funkcja } x = f(a, b) \quad (33)$$

to należy obliczyć  $f(a, \underline{b})$ ,  $f(\overline{a}, \underline{b})$ ,  $f(a, \overline{b})$  i  $f(\overline{a}, \overline{b})$ , a kolejne kroki obliczeń wykonywać dla tych wszystkich wartości. Gdy użyta zostanie kolejna zmienna początkowa – liczba wyników pośrednich zwiększy się dwukrotnie. Ogólne – jeśli w ciągu obliczeń używa się  $n$  niezależnych zmiennych początkowych to wynikiem końcowym będzie  $2^{n+1}$  liczb, z których skrajne wyznaczają granice ostatecznego przedziału. Aby zminimalizować koszt obliczeniowy należy na tyle, na ile możliwe, zoptymalizować algorytm np. grupując w pojedynczym kroku wszystkie operacje związane z wybraną zmienną.

Spore możliwości przyspieszenia obliczeń w arytmetyce przedziałowej daje wykorzystanie rozszerzeń strumieniowych listy rozkazów procesorów tj. SSE (Intel) i 3DNow! (AMD). Pozwalają one na równoległe wykonanie tej samej operacji arytmetycznej na kilku argumentach [12]. Szczególnie korzystne jest zastosowanie instrukcji SSE-2 w których wprowadzono operacje równoległe na liczbach podwójnej precyzji [13].

## 5. Wnioski

Metoda analizy przedziałowej jest przydatnym sposobem kontroli dokładności wyników obliczeń, w przypadku gdy część zmiennych jest znana w przybliżeniu, a rozkład ich wartości nie jest znany. Jednak efektywne jej wykorzystanie wymaga dobrego przemyślenia algorytmu i przewidywania źródeł przeszacowania przedziałów wyników obliczeń.

## 6. Literatura

- [1] Björck Å., Dahlquist G.: Metody numeryczne. PWN, Warszawa 1987.
- [2] IEEE-754-1985, IEEE Standard for Binary Floating-Point Arithmetic, Institute of Electrical and Electronics Engineers, New York, 1985.
- [3] Tuszyński M., Gorczyński R.: Koprocesory. Komputerowa Oficyna Wydawnicza „Help”, Warszawa 1992.
- [4] Taylor J.R.: Wstęp do analizy błęd pomiarowego. Wydawnictwa naukowe PWN, Warszawa 1999.
- [5] Górka P.: Zastosowanie arytmetyki dokładnościowej w programie analizy obwodów elektrycznych; VI Konferencja. Naukowo-Techn. „Zastosowania Komputerów w Elektrotechnice”, Poznań 2001.
- [6] Dmytryshyn R., Górka P., Poprawa dokładności analizy obwodów liniowych, 4-th International Modeling School of AMSE-UAPL, Szack 2000.
- [7] Moore R.E.: Interval analysis. Prentice-Hall, Englewood Cliffs, 1966.
- [8] Kearfott R.B.: Interval Computations: Introduction, Uses, and Resources, Euromath Bulletin 2 (1), 1996.
- [9] Garloff J.: Interval mathematics: A bibliography, Freiburger Intervall-Berichte, 85(6), 1985.
- [10] Garloff J.: Bibliography on interval mathematics, continuation. Freiburger Intervall-Berichte, 87(2), 1987.
- [11] <http://cs.utep.edu/interval-comp/main.html>
- [12] Schulte M., Akkas A., Zelov V.A.: Compiler Support for Interval Arithmetic, Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference, Venice, Italy, 1999.
- [13] Lambov B.: Interval Arithmetic Using SSE-2, Reliable Implementation of Real Number Algorithms, Dagstuhl Seminar Procs., 2006.