Metody numeryczne i programowanie

Roman Zuber

October 12, 2020

Pierwsza część podręcznika składa się z sześciu rozdziałów. Pierwszy z nich zawiera pewne informacje wstępne, które powinny być z uczniami przedyskutowane. Drugi jest poświęcony ważnemu zagadnieniu metod numerycznych — teorii błędów. Większość omówionych w podręczniku metod numerycznych wiąże się z wielomianami. Dlatego rozdział trzeci jest poświęcony wielomianom, potraktowanym jako ważna klasa funkcji w teorii metod numerycznych. W rozdziale czwartym podane są dwie metody konstrukcji wielomianów aproksymujących funkcje ciągłe. Rozdział piąty jest poświęcony metodom rozwiązywania równań z jedną niewiadomą, w szczególności — równań algebraicznych. Rozdział szósty omawia kilka przybliżonych metod obliczania całek oznaczonych. Niektóre metody są uzasadnione za pomocą twierdzeń. Prostsze twierdzenia są udowodnione, natomiast w przypadku trudniejszych twierdzeń wskazana jest odpowiednia literatura w języku polskim. Zrozumienie dowodów niektórych twierdzeń wymaga znajomości twierdzenia o wartości średniej Lagrange'a. Warto może też zapamiętać, że w tekście używane są symbole o następującym znaczeniu: **T** — twierdzenie, **P** — przykład, **A** — algorytm, **D** — definicja.

Drugi rozdział drugiej części podręcznika jest poświęcony językowi ALGOL 60. W poszczególnych paragrafach tego rozdziału omawiane są pojęcia tego języka oraz ich zastosowanie do budowy programów w języku ALGOL. Od czasu do czasu podawane są również informacje o ALGOLu 1204 — konkretnej realizacji języka wzorcowego ALGOL 60. Chodzi bowiem o to, aby umożliwić uczniom poznane metody od razu sprawdzać na maszynie. Wydaje się, że w Polsce powinien być najłatwiejszy dostęp do maszyn ODRA 1204, które są wyposażone w translatory ALGOLu 1204.

W podręczniku podane są liczne przykłady programów realizujących te zadania, które omówione są w pierwszej części podręcznika. Ponadto, w dodatku do niniejszego podręcznika podanych jest kilka procedur lub pełnych programów w języku ALGOL 1204, realizujących niektóre algorytmy lub metody.

Część I

Teoria metod numerycznych

Rozdział 1

INFORMACJE WSTĘPNE

Co to są metody numeryczne?

Nie jest łatwo wymienić zagadnienia, którymi zajmujemy się w tym dziale matematyki. Można by powiedzieć, chociaż nie jest to określenie dokładne, że metody numeryczne zajmują się rozwiązywaniem zadań matematycznych, w których dane i wyniki są liczbami. Nie jest to dyscyplina nowa, ponieważ obliczeniami ludzkość zajmuje się od niepamiętnych czasów. Jednakże burzliwy jej rozwój rozpoczął się niedawno, z chwilą pojawienia się szybkich maszyn matematycznych. Elektroniczne maszyny cyfrowe umożliwiły rozwiązywanie takich zadań, których dawnymi metodami, nawet za pomocą arytmometrów elektrycznych, nikt nie był w stanie rozwiązać, gdyż wymagały wykonania setek tysięcy działań arytmetycznych. Wraz z nowymi możliwościami obliczeniowymi zrodziły się nowe zagadnienia. Większość zagadnień związanych z procesem obliczeń jest treścią metod numerycznych. Ważniejsze z nich omówimy w niniejszym podręczniku.

Okazało się, że wykonanie obliczeń za pomocą maszyn cyfrowych musi być poprzedzone pewnymi czynnościami przygotowawczymi. Do ważniejszych zaliczamy:

a) wybór metody numerycznej dla danego zadania,
b) analiza dokładności wyników,
c) opracowanie algorytmu,
d) opracowanie programu obliczeń.

Postaramy się każdą z wymienionych czynności opisać dokładniej.

1.1. Wybór metody numerycznej

Celem obliczeń może być na przykład rozwiązanie równania kwadratowego, rozwiązanie układu równań liniowych, obliczenie wartości całki oznaczonej itp. Dla dalszych rozważań przyjmijmy, że mamy do dyspozycji maszynę wykonującą cztery działania arytmetyczne: dodawanie, odejmowanie, mnożenie i dzielenie, oraz umiejącą dodatkowo obliczać wartości funkcji: e^x, $\log x$ (przy czym symbolem log oznaczyliśmy logarytm naturalny, to znaczy logarytm przy podstawie $e = 2{,}7182818...$), $\sin x$[1], $\cos x$, $\sqrt{x}$. Operacje te będziemy nazywali **elementarnymi operacjami maszyny.** Metodę numeryczną należy tak wybrać, aby można ją było zrealizować na maszynie. Dlatego musi być ona opisana wzorami zawierającymi tylko operacje elementarne.

Metody numeryczne można podzielić na dwie grupy: **metody dokładne** i **metody przybliżone.** Do dokładnych metod zaliczamy takie, które umożliwiają uzyskanie rozwiązania danego zadania po wykonaniu skończonej liczby operacji elementarnych. Oczywiście, metody te są dokładne w teorii, tzn. przy założeniu, że wszystkie zawarte w nich operacje są wykonywane dokładnie. W praktyce operujemy na skończonych ułamkach dziesiętnych, a wyniki działań, mające często rozwinięcia nieskończone, zaokrąglamy lub „obcinamy" na którejś cyfrze. Te zaokrąglenia lub obcięcia są w metodach dokładnych jedynym źródłem błędu wyników obliczeń, (odróżniamy tu błąd od omyłki — przestawienia cyfr w liczbie, opuszczenia znaku, fałszywej interpretacji zadania itp.). Natomiast każdą metodę, w której uzyskanie rozwiązania jest możliwe po wykonaniu nieskończenie wielu operacji, zalicza się do metod przybliżonych. Wyjaśnimy to na przykładach.

P. 1.1. Pierwiastki x_1 i x_2 równania kwadratowego

$$ax^2+bx+c = 0$$

można obliczyć ze wzorów:

$$\Delta = b^2-4ac,\quad w_1 = -\frac{b}{2a},\quad w_2 = \frac{\sqrt{\Delta}}{2a};$$

[1] Jako argument x przyjmujemy miarę teoretyczną kąta.

$$x_1 = w_1+w_2,\quad x_2 = w_1-w_2.$$

Powyższe wzory opisują, oczywiście, dokładną metodę numeryczną.

P. 1.2. Innym przykładem dokładnej metody numerycznej jest **metoda eliminacji,** stosowana dla rozwiązywania układu n równań liniowych z n niewiadomymi, np. układu

$$\begin{aligned} 3x+y-2z &= 0,\\ 2x+y+3z &= 0,\\ x+y+\ z &= 7. \end{aligned}$$

Jednakże większości zadań nie udaje się rozwiązać metodami dokładnymi.

P. 1.3. Nie znana jest dokładna metoda rozwiązywania równania

$$2\sin x-x = 0.$$

Znane są natomiast liczne metody przybliżone dla tego typu zadań. Niektóre z nich są omówione w rozdziale 5. Na przykład zastosowanie metody Newtona (5.11) prowadzi do następującego wzoru obliczeniowego dla tego zadania:

$$x_{i+1} = x_i-\frac{2\sin x_i-x_i}{2\cos x_i-1}. \tag{1.1}$$

Wzór ten wykorzystuje się w następujący sposób: Wybiera się wpierw liczbę x_0. Na ogół powinna to być liczba niedużo różniąca się od dokładnego pierwiastka. Teraz dla $i = 0$ można obliczyć wartość prawej strony wzoru (1.1) i uzyskać liczbę x_1, leżącą na ogół bliżej pierwiastka aniżeli x_0. Obliczając dalej wartość prawej strony dla x_1 otrzymamy liczbę x_2 itd. Przyjmując na przykład $x_0 = \frac{\pi}{2}$, obliczamy z (1.1)

$$\begin{aligned} x_1 &= 2{,}0000\ 0000\\ x_2 &= 1{,}9009\ 9560\\ x_3 &= 1{,}8955\ 1165\\ x_4 &= 1{,}8954\ 9427\\ x_5 &= 1{,}8954\ 9427\\ &\ldots\ldots\ldots\ldots \end{aligned} \tag{1.2}$$

Uzyskaliśmy w ten sposób ciąg liczb zbieżny do pierwiastka rozpatrywanego równania, który z dokładnością dziewięciu cyfr znaczących jest równy 1,8954 9427.[1]

[1] Pojęcie cyfr znaczących jest określone w definicji D.2.4, str. 32.

Ponieważ każda następna liczba w ciągu (1.2) leży bliżej pierwiastka od poprzedniej, metodę opisaną w przykładzie 1.3 nazywamy **metodą kolejnych przybliżeń.** Metoda opisana wzorem (1.1) nie jest dokładną metodą numeryczną, ponieważ dla uzyskania dokładnego pierwiastka musielibyśmy obliczyć nieskończenie wiele wyrazów ciągu (1.2), co w praktyce jest niemożliwe, a z drugiej strony niecelowe. Na ogół po obliczeniu kilku czy kilkunastu przybliżeń otrzymuje się przybliżoną wartość pierwiastka wystarczająco dokładną dla praktycznych potrzeb.

Rozpatrzmy jeszcze inny przykład zadania, dla którego nie istnieje dokładna metoda numeryczna.

P. 1.4. Obliczyć całkę oznaczoną $I = \int_0^1 \frac{\sin x}{x} dx$.

Numeryczne metody obliczania całek oznaczonych są omówione w rozdziale 6. Stosując na przykład metodę szeregów potęgowych, omówioną w § 6.4, możemy dokładną wartość całki przedstawić w postaci nieskończonej sumy

$$(1.3)\quad I = 1 - \frac{1}{3\cdot 3!} + \frac{1}{5\cdot 5!} - \ldots + (-1)^n \frac{1}{(2n+1)\,(2n+1)!} + \ldots$$

Również w tym przypadku nie jest praktycznie możliwe obliczenie dokładnej wartości całki, ponieważ wymaga to zsumowania nieskończenie wielu wyrazów szeregu (1.3). Okazuje się jednak, że jeśli jako przybliżoną wartość całki przyjmiemy sumę S_n n pierwszych wyrazów szeregu (1.3)

$$S_n = \sum_{i=0}^{n} (-1)^i \frac{1}{(2i+1)\,(2i+1)!},$$

to wówczas moduł[1]) różnicy pomiędzy dokładną i policzoną wartością całki jest nie większy od pierwszego nie dołączonego do sumy S_n wyrazu szeregu (1.3). Stąd wynika, że

$$(1.4)\qquad |I - S_n| < \frac{1}{(2n+3)\,(2n+3)!}.$$

[1]) Zamiast mówić: „wartość bezwzględna wyrażenia...” będziemy mówić: „moduł wyrażenia...”.

Jeśli, w szczególności, jako przybliżoną wartość całki przyjmiemy sumę pięciu wyrazów szeregu (1.3), to będzie ona różniła się od dokładnej wartości o mniej niż $\frac{1}{11\cdot 11!} \approx 2\cdot 10^{-9}$.

Doszliśmy tu do pewnego ważnego pojęcia związanego z przybliżonymi metodami numerycznymi, tzw. błędu metody. Każda bowiem przybliżona metoda numeryczna może doprowadzić do dokładnego rozwiązania po wykonaniu nieskończenie wielu kroków obliczeniowych. Ponieważ praktycznie nie jest to możliwe, musimy w pewnym miejscu obliczenia przerwać, a tym samym zdecydować się na przybliżony wynik zadania. Wówczas pojawia się bardzo ważne pytanie. Jak dużym błędem obarczone jest obliczone rozwiązanie? Błąd ten związany jest z metodą, dlatego nazywamy go **błędem metody.** Bywa on też, ze zrozumiałych względów, nazywany **błędem obcięcia.** Dla niektórych przybliżonych metod numerycznych błąd metody można dość łatwo i dość dobrze oszacować. Na przykład w przypadku metody zastosowanej w przykładzie 1.4 oszacowanie takie jest określone nierównością (1.4). Dla wielu jednakże metod oszacowanie błędu metody jest zadaniem bardzo trudnym.

1.2. Analiza dokładności wyników

W tym miejscu musimy sobie zdać sprawę z faktu, że obliczone wyniki danego zadania matematycznego są prawie zawsze obarczone błędami, a tym samym różnią się od wyników dokładnych.[1]) Takie stwierdzenie może budzić wątpliwości, ponieważ w podręcznikach szkolnych mamy wiele takich zadań, których rozwiązania są dokładne. Są to jednak specjalnie dobrane zadania, tak aby ułatwić uczniom pracę.

Ważnym, chociaż niełatwym, zadaniem metod numerycznych jest szacowanie wielkości błędów obarczających wyniki obliczeń, to zna-

[1]) Mówiąc o błędach ma się na myśli błędy nierozerwalnie związane z procesem obliczeń (np. błędy spowodowane zaokrąglaniem liczb), a nie pomyłki spowodowane takim czy innym niedopatrzeniem (np. przestawieniem dwu cyfr lub wykonaniem działań na niewłaściwych danych). Wszelkie pomyłki można łatwo wykryć i wyeliminować, np. przez powtórzenie obliczeń.

czy znajdowanie liczby ograniczającej z góry moduł różnicy pomiędzy dokładnym i obliczonym rozwiązaniem. Użyliśmy tu słowa „szacowanie", gdyż dokładne obliczenie błędu nie jest możliwe. Aby przeprowadzić sensowną analizę błędów, musimy przede wszystkim zdać sobie sprawę z tego, jakie są źródła ich powstawania.

Numeryczne wyniki zadań zawierają błędy powstające na ogół w dwóch momentach: a) przy matematycznym opisie badanego zjawiska i przygotowaniu danych liczbowych do obliczeń, b) podczas wykonywania obliczeń. Dane uzyskane z pomiarów wielkości fizycznych (temperatury, ciśnienia, długości, natężenia itp.), zwane danymi empirycznymi, są na skutek niedokładności przyrządów pomiarowych obarczone **błędami pomiarów.** Istnieją również dwa źródła powstawania błędów w czasie wykonywania obliczeń. Pierwszym z nich jest stosowanie przybliżonej metody numerycznej, o czym była mowa w § 1.1. Ten typ błędu we wszystkich jego możliwych postaciach będziemy nazywali błędem metody.

Drugim źródłem są zaokrąglenia. Liczby biorące udział w obliczeniach są zapamiętywane w pamięci maszyny. W każdej komórce pamięci może być zapisana liczba o skończonej ilości cyfr dziesiętnych, np. 10.[1] Z tego powodu wszystkie dane zawierające więcej niż 10 cyfr nie mogą być zapisane w pamięci dokładnie. Nie można na przykład dokładnie zapisać jakiejkolwiek liczby niewymiernej ($\sqrt{2}$, e, π, ...). Podobnie nie uda się zapisać dokładnych wartości liczb wymiernych o nieskończonym rozwinięciu dziesiętnym. Każda taka liczba może być wpisana do pamięci po obcięciu do 10 cyfr i zaokrągleniu. W ten sposób tworzy się **błąd zaokrąglenia.** Błąd zaokrąglenia może powstać również przy wykonywaniu operacji arytmetycznych na liczbach. Wiadomo bowiem, że iloraz dwu liczb o skończonych rozwinięciach dziesiętnych może mieć rozwinięcie dziesiętne nieskończone (np. 3:7), a ich iloczyn ma tyle cyfr ile mają oba czynniki łącznie. Aby wynik mnożenia lub dzielenia można było wpisać do pamięci, konieczne jest również odrzucenie końcowych cyfr i zaokrąglenie.

Niezależnie od przyczyny, dla której pewna wartość jest niedokładna, podczas wykonywania działań na liczbach obarczonych

[1] W rzeczywistości liczby w maszynie są zapamiętywane na ogół w systemie dwójkowym. Analiza błędów nie zależy jednak od przyjętego systemu liczenia.

błędami występuje zjawisko przenoszenia się błędów z danych na wyniki. Jest to zjawisko bardzo niebezpieczne, szczególnie przy dużej ilości działań (rzędu tysięcy lub więcej), ponieważ może spowodować bardzo znaczne zniekształcenie wyników.

P. 1.5. Zmierzono krawędź sześcianu z dokładnością 1 cm i uzyskano wynik 75 cm. Obliczyć objętość sześcianu.

Z faktu, że pomiaru dokonano z dokładnością 1 cm wynika, że długość krawędzi sześcianu może być każdą liczbą zawartą między 74 cm i 76 cm. Zatem objętość tego sześcianu może być dowolną liczbą zawartą między $74^3\text{cm}^3 = 405\,244\ \text{cm}^3$ i $76^3\text{cm}^3 = 438\,976\ \text{cm}^3$. Jeśli dla objętości przyjąć średnią arytmetyczną obliczonych wyników, to możemy wówczas powiedzieć, że objętość sześcianu jest równa $422\,110\ \text{cm}^3$ z błędem $16\,866\ \text{cm}^3$.

Z powyższego przykładu widać, jak bardzo błąd pomiaru wpłynął na wynik obliczeń. Można sobie wyobrazić jak bardzo mogą być zniekształcone wyniki, kiedy na wielu danych empirycznych wykonuje się dużą ilość działań. Sytuacja może ulec poprawie jedynie wtedy, kiedy będziemy dokładniej mierzyć. To jest możliwe wtedy, gdy będziemy mieli do dyspozycji odpowiednio dokładne przyrządy pomiarowe.

W następnym rozdziale jest przedstawiona **teoria błędów maksymalnych,** umożliwiająca określenie wielkości błędów przenoszonych z danych obarczonych błędami na wyniki operacji arytmetycznych. Należy tu wspomnieć, że oprócz teorii błędów maksymalnych istnieją inne teorie opisujące to zjawisko. Na przykład bardzo dobrze opisuje go statystyczna teoria błędów, ponieważ sam proces przenoszenia się błędów jest zjawiskiem przypadkowym.

1.3. Opracowanie algorytmu

W nowoczesnej praktyce obliczeniowej zadania wymagające wykonania dużej ilości obliczeń rozwiązuje się za pomocą maszyn cyfrowych. Rozwiązanie zadania przez maszynę musi być poprzedzone opracowaniem algorytmu, a następnie programu obliczeń.

Co to jest algorytm?

Aby odpowiedzieć na to pytanie, musimy zdać sobie sprawę z na-

stępującego ważnego faktu. Maszyna cyfrowa wprawdzie potrafi wykonywać wiele różnych działań na liczbach, jednakże ona nie myśli. Dlatego cały proces myślowy, jaki powinien być przeprowadzony w czasie rozwiązywania zadania, musi być zawarty w algorytmie. Nie będziemy podawali tu dokładnej definicji algorytmu, a treść tego pojęcia postaramy się opisać na przykładach. Szczegółowy algorytm dokładnie opisuje wszystkie operacje i czynności, podając kolejność ich wykonywania, które prowadzą do uzyskania rozwiązania. Posłużymy się przykładem.

P. 1.6. Niech będą dane trzy liczby rzeczywiste a, b, c, $(a \neq 0)$. Rozpatrzmy równanie

$$ax^2+bx+c = 0.$$

Należy wyznaczyć liczbę r różnych pierwiastków rzeczywistych tego równania i obliczyć wartości tych pierwiastków. Wiadomo, że równanie kwadratowe posiada pierwiastki rzeczywiste wtedy, gdy $\Delta = b^2-4ac \geqslant 0$. Metoda obliczania tych pierwiastków jest dobrze znana (odpowiednie wzory były podane w przykładzie 1.1). Liczbę r rzeczywistych pierwiastków równania można wyznaczyć łatwo, badając znak Δ. Mianowicie:

jeśli $\Delta > 0$, to $r = 2$,
jeśli $\Delta = 0$, to $r = 1$,
jeśli $\Delta < 0$, to $r = 0$.

Poprawny algorytm musi zawierać analizę wszystkich możliwych przypadków, jakie mogą zdarzyć się w czasie wykonywania obliczeń. Algorytm powinien także zawierać wszystkie wzory obliczeniowe, jakie należy stosować w każdym z przypadków, jakie mają miejsce w czasie obliczeń.

Przedstawimy teraz jedną z możliwych wersji algorytmu dla zadania sformułowanego w przykładzie 1.6.

A. 1.1. *Obliczanie liczby r pierwiastków rzeczywistych równania $ax^2+bx+c = 0$, $a \neq 0$, oraz wartości tych pierwiastków x_1, x_2 (gdy $r = 2$) lub x_1 (gdy $r = 1$).*

Przy rozwiązywaniu powyższego zadania należy wykonać następujące obliczenia.

1) *Obliczyć*:

$$\Delta = b^2-4ac.$$

2) *Zbadać znak Δ, przy czym*:

a) *jeśli* $\Delta > 0$, *to*

$$r = 2, \quad x_1 = \frac{-b-\sqrt{\Delta}}{2a}, \quad x_2 = \frac{-b+\sqrt{\Delta}}{2a},$$

b) *jeśli* $\Delta = 0$, *to*

$$r = 1, \quad x_1 = -\frac{b}{2a},$$

c) *jeśli* $\Delta < 0$, *to*

$$r = 0.$$

1.4. Opracowanie programu obliczeń

Algorytm określa jednoznacznie sposób rozwiązywania danego zadania. Zawiera on wszystkie wzory, które wykonane w określonej kolejności prowadzą do rozwiązania. Na podstawie algorytmu każdy, kto rozumie sens podanych tam wzorów, może już łatwo, w sposób zupełnie mechaniczny, zadanie rozwiązać. Jeśli jednak przekazujemy obliczenia na maszynę cyfrową, musimy jeszcze opracować tzw. **program obliczeń.**

Program obliczeń jest również opisem algorytmu, ale zapisanym w taki sposób, aby mogła go rozumieć maszyna cyfrowa. Każda bowiem maszyna ma swój język, zwany językiem wewnętrznym, służący do porozumiewania się z nią człowieka.(1)

Można by krótko powiedzieć, że program jest opisem algorytmu w języku wewnętrznym maszyny. Program obliczeń zawiera na ogół nieco więcej elementów aniżeli algorytm. W programie na przykład musi być określony sposób drukowania wyników obliczeń, kolejność czytania danych itp. Mówiąc krótko, program winien być tak opracowany, aby na jego podstawie „niemyśląca" maszyna mogła zupełnie automatycznie, bez ingerencji człowieka, wykonać wszystkie czynności i obliczenia konieczne dla uzyskania poprawnych wyników.

(1) Należy w tym miejscu wyjaśnić, że używanie takich zwrotów jak: „język maszyny", „porozumiewanie się z maszyną" itp. jest technicznie uzasadnione.

Programowanie, tzn. zapisywanie algorytmu w języku wewnętrznym maszyny, jest czynnością bardzo pracochłonną i na ogół — żmudną. Dlatego w ostatnich latach czynność programowania powierza się do wykonania samej maszynie.

Ostatnio w świecie, a także w Polsce, używane są uniwersalne języki algorytmiczne dla zapisywania zarówno algorytmów jak i programów. W obecnej chwili powszechnie używa się języka ALGOL 60 (ALGOrithmic Language) opracowanego w 1960 roku przez zespół 13 matematyków zachodnioeuropejskich i amerykańskich. Opis tego języka, wprawdzie niekompletny, ale wystarczający dla nauczenia się podstaw, które umożliwią nam opracowywać wiele algorytmów i programów, znajduje się w drugiej części niniejszego podręcznika. Język ten ma tę zaletę, że zapisane w nim algorytmy są czytelne i łatwo zrozumiałe dla człowieka (można go stosować w publikacjach), a również oprzeć na nim programowanie. Oczywiście maszyna może taki algorytm wtedy „zrozumieć", kiedy jest wyposażona w tzw. **translator** ALGOLu, tzn. program tłumaczący treść algorytmu zapisanego w języku ALGOL na swój własny język wewnętrzny. Polskie maszyny typu ODRA są w takie translatory wyposażone.

Z powyższego wynika, że w przypadku kiedy algorytm zostanie opracowany w języku algorytmicznym, np. ALGOLu, to wtedy już nie ma potrzeby przygotowywania programu w języku wewnętrznym (maszyna potrafi wyprodukować go sama w oparciu o algorytm zapisany w języku ALGOL). W takiej sytuacji nie ma istotnej różnicy pomiędzy algorytmem i programem. I dlatego algorytm przygotowany w języku algorytmicznym jest również programem obliczeń dla maszyny.

Aby zrozumieć zasadę opracowywania programów w języku algorytmicznym, spróbujmy taki program opracować dla algorytmu 1.1. Program ten zapiszemy w języku nieco podobnym do ALGOLu (ponieważ ALGOLu jeszcze nie znamy). Wydaje się, że treść tego programu powinniśmy bez trudności zrozumieć. Wyjaśnimy tylko sens tzw. **instrukcji podstawienia,** którą będziemy, podobnie jak w ALGOLu zapisywali w postaci:

$$Z := W,$$

gdzie Z oznacza dowolną zmienną, a W — wyrażenie arytmetyczne. Przez wyrażenie arytmetyczne będziemy tu rozumieli po prostu pewne

wyrażenie algebraiczne, np. b^2-4ac, $\frac{\sqrt{\Delta}}{2a}$ itp. Instrukcja podstawienia poleca obliczyć wartość liczbową wyrażenia arytmetycznego W i podstawić ją pod zmienną Z. Na przykład instrukcja $Z := 1$ poleca podstawić pod zmienną Z wartość 1. Inne symbole czy zdania występujące w programie nie powinny nastręczać trudności.

Ustalimy teraz znaczenie poszczególnych zmiennych występujących w programie, czyli dokonamy tzw. opisu zmiennych. Będziemy wykorzystywali w programie zmienne rzeczywiste: a, b, c, Δ, x_1 i x_2, zgodnie ze wzorami występującymi w algorytmie A. 1.1, i dodatkowe dwie zmienne rzeczywiste w_1, w_2, które nazwiemy zmiennymi roboczymi, służącymi dla ułatwienia zapisu programu. Oprócz tego w programie wystąpi jedna zmienna całkowita, określająca ilość pierwiastków rzeczywistych.

W maszynach występuje wyraźny podział na liczby całkowite i niecałkowite, zwane rzeczywistymi. Liczby te są w maszynie przedstawiane różnie, a działania na nich są wykonywane przez inne bloki maszyny. Z punktu widzenia czystej matematyki taki podział jest niekonsekwentny, ponieważ wiadomo, że liczby całkowite są podzbiorem liczb rzeczywistych. Nieco więcej informacji o tych dwu postaciach liczb znajdzie czytelnik w 1 rozdziale II części.

Zapiszemy teraz nasz program za pomocą języka przypominającego ALGOL:

Początek programu;

Czytaj (a, b, c);
$\Delta := b^2-4ac$;
Jeśli $\Delta < 0$, to idź do 1, w przeciwnym przypadku
Jeśli $\Delta = 0$, to idź do 2, w przeciwnym przypadku

$r := 2$;

$w_1 := -\frac{b}{2a}$;

$w_2 := \frac{\sqrt{\Delta}}{2a}$;

$x_1 := w_1-w_2$;

$x_2 := w_1 + w_2$;

Idź do 3;

2: $r := 1$;

$x_1 := -\frac{b}{2a}$;

$x_2 := x_1$;

Idź do 3;

1: $r := 0$;

$x_1 := x_2 := 0$;

3: Drukuj (r, x_1, x_2);

Koniec programu.

Zauważmy, że w przypadku kiedy $\Delta = 0$ program podstawia $x_2 = x_1$. Oznacza to, że w tym przypadku zostaną wydrukowane dwa jednakowe pierwiastki. Natomiast w przypadku $\Delta < 0$ program podstawia $x_1 = x_2 = 0$. Oznacza to, że w przypadku pierwiastków zespolonych (o czym informuje wartość r), program wyprowadza trzy liczby równe zero.

Prześledzimy teraz działanie programu w przypadku: $a = 1$, $b = -3$, $c = 2$.

Po instrukcji Czytaj (a, b, c) zmienne a, b, c przyjmą konkretne wartości 1, −3, 2. Po wykonaniu pierwszej instrukcji podstawienia zostanie obliczona wartość Δ równa 1. Ponieważ $\Delta > 0$, więc program przejdzie do wykonania drugiej instrukcji „Jeśli" (gdyby $\Delta < 0$, to nastąpiłoby przejście do instrukcji oznaczonej numerem 1). Druga instrukcja „Jeśli" spowodowałaby przejście do instrukcji o numerze 2, gdyby Δ równała się zero. Następuje natomiast przejście do instrukcji podstawienia $r := 2$. Następnie oblicza się wartość $w_1 = \frac{3}{2}$, oraz wartość $w_2 = \frac{1}{2}$. Dalej obliczone zostaną wartości $x_1 = 1$ i $x_2 = 2$. Po wykonaniu tych czynności program przejdzie do wykonania instrukcji oznaczonej numerem 3. Instrukcja ta spowoduje wydrukowanie liczb: 2, 1, 2, to znaczy ilości pierwiastków rzeczywistych równania i ich wartości.

ZADANIA

1. Opisać metodę eliminacji rozwiązywania układu równań:

$$a_1x + b_1y = c_1$$
$$a_2x + b_2y = c_2,$$

przy założeniu, że $a_1 \neq 0$ i $a_1b_2 - a_2b_1 \neq 0$.

2. Opracować algorytm dla metody eliminacji rozwiązywania układu dwu równań liniowych z dwiema niewiadomymi.
3. Niech celem obliczeń będzie wyznaczenie $\sqrt{a}$. W jakich przypadkach metoda szkolna, rozpoczynająca od pogrupowania cyfr po dwie w obie strony od kropki dziesiętnej, jest metodą przybliżoną? Znaleźć oszacowanie błędu metody.
4. Pierwiastek kwadratowy z dowolnej liczby $a \geqslant 0$ można obliczać za pomocą wzoru iteracyjnego:

$$x_{i+1} = \frac{1}{2}\left(x_i + \frac{a}{x_i}\right), \tag{1.5}$$

uzyskanego ze wzoru Newtona (5.11). Obliczyć przybliżoną wartość $\sqrt{2}$, przyjmując $x_0 = 1$. Porównując kilka pierwszych wartości przybliżonych z dokładną wartością $\sqrt{2} = 1{,}4142136\ldots$, określić szybkość zbieżności ciągu kolejnych przybliżeń do pierwiastka, tzn. oszacować błąd metody.

Wskazówka. Obliczenia za pomocą wzoru (1.5) prowadzić na ułamkach zwyczajnych, a następnie zamieniać je na dziesiętne i obliczać dla kolejnych przybliżeń różnice $|x_i - \sqrt{2}|$.

5. Dane są dwie liczby rzeczywiste a i b. Opracować algorytm znajdujący większą z nich.

Rozdział 2

TEORIA BŁĘDÓW

2.1. Podstawowe pojęcia teorii błędów maksymalnych

Podstawowych pojęć teorii błędów maksymalnych jest pięć: wielkość, dokładna wartość wielkości, przybliżona wartość wielkości, błąd bezwzględny wartości przybliżonej, liczba przybliżona. Trzy pierwsze pojęcia przyjmujemy w tej teorii jako pojęcia pierwotne. Postaramy się opisać je za pośrednictwem przykładów.

Wielkością może być dowolna stała matematyczna, wynik określonego działania, pierwiastek rozpatrywanego równania itp. Na przykład π jest określona jako stosunek obwodu okręgu do jego średnicy, $e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$, $\sqrt{2}$ jest pierwiastkiem równania kwadratowego $x^2-2=0$.

Przez wartość dokładną wielkości rozumiemy wartość wynikającą wprost z definicji tej wielkości, a więc nie obarczoną żadnymi błędami.

Natomiast **wartością przybliżoną** wielkości będziemy nazywali wartość liczbową otrzymaną w wyniku obliczeń. Oczywiście, prawie nigdy nie otrzymamy w wyniku obliczeń dokładnej wartości. Zostanie ona zniekształcona skutkiem zaokrąglania lub używania przybliżonych metod obliczeniowych.

Możemy mieć do czynienia również z wielkościami fizycznymi, takimi jak: ciśnienie, temperatura, długość, stężenie, itp. Sensowne określenie dokładnej wartości wielkości fizycznej jest najczęściej niemożliwe. Dokładną wartość takiej wielkości moglibyśmy wówczas

otrzymać, gdybyśmy mogli ją zmierzyć przyrządem nieskończenie dokładnym. Niestety takich przyrządów pomiarowych praktycznie nie da się nigdy wykonać. Natomiast przybliżoną wartością wielkości fizycznej będziemy nazywali tę wartość, którą uzyskamy z pomiaru aktualnie dostępnym przyrządem pomiarowym. Formalnie rzecz biorąc każda liczba jest wartością przybliżoną dla danej wielkości. Na przykład przybliżeniem liczby π może formalnie być każda z liczb: 3,1415, 3,2, a nawet 6,25. Jednakże liczby bardzo dalekie od wartości dokładnej są w praktyce nieprzydatne. Dlatego w procesie obliczeniowym staramy się uwzględniać tylko takie wartości przybliżone, które możliwie mało różnią się od wartości dokładnych.

Ponieważ przybliżona wartość wielkości nie jest określona jednoznacznie, więc znajomość tylko wartości przybliżonej jest informacją bez znaczenia. Tu mogą nasunąć się wątpliwości. Wiadomo powszechnie, że w różnych tablicach podaje się tylko przybliżone wartości pewnych wielkości: sinusów, logarytmów, pierwiastków itp. Istnieje tu jednak pewna oczywista umowa, polegająca na tym, że jeśli dla jakiejkolwiek wielkości podana jest w tablicy jej przybliżona wartość, np. z ośmioma cyframi znaczącymi, to ona przedstawia najlepsze przybliżenie osiągalne z tą ilością cyfr.

Przypuśćmy jednak, że chcemy obliczyć przybliżoną wartość wielkości, której nie ma w żadnej tablicy, bo na przykład wielkość ta została określona po raz pierwszy. Przyjmijmy, że wybraliśmy metodę dla jej obliczania, opracowaliśmy poprawny algorytm i przekazaliśmy do obliczenia maszynie cyfrowej. Maszyna wykonała obliczenia i wydrukowała wynik $a = 5{,}87564953$. Otrzymany wynik zawiera osiem cyfr po przecinku. Czy wolno twierdzić, że obliczona liczba ma wszystkie cyfry poprawne, tzn. różni się od dokładnej nie więcej niż o $0{,}5\cdot 10^{-8}$? Byłoby to zbyt pochopne stwierdzenie. Może się bowiem okazać, że zastosowaliśmy tak mało dokładną metodę, a prócz tego sam proces obliczeniowy wprowadził tak duże błędy, że obliczona liczba jest bardzo odległa od wartości dokładnej. Oczywiście, może być również sytuacja wręcz przeciwna, tzn. obliczona liczba różni się od dokładnej dopiero na siódmym lub ósmym miejscu po przecinku. Należy oczywiście dokonać analizy dokładności i oszacować wielkość błędu, którym obarczona jest obliczona liczba. W wyniku takiej analizy będziemy mogli powiedzieć, że obliczona wartość a różni się od dokładnej wartości A nie więcej niż o Δa. W ten sposób

doszliśmy do nowego pojęcia teorii błędów maksymalnych — błędu bezwzględnego liczby przybliżonej. Pojęcie to możemy już określić za pomocą trzech poprzednich pojęć.

D. 2.1. Niech A będzie wartością dokładną, a a wartością przybliżoną pewnej wielkości. **Błędem bezwzględnym wartości przybliżonej** nazywamy każdą liczbę Δa spełniającą warunek

$$|A-a| \leqslant \Delta a,$$

to znaczy taką liczbę, że

$$a-\Delta a \leqslant A \leqslant a+\Delta a. \tag{2.1}$$

Z definicji tej wynika, że wartość przybliżona a i jej błąd bezwzględny Δa wyznaczają przedział

$$\langle a-\Delta a;\ a+\Delta a\rangle,$$

do którego należy dokładna wartość A. Z definicji wynika również, że $\Delta a \geqslant 0$. Błąd bezwzględny nie jest określony jednoznacznie.

P. 2.1. Wiemy, że $\pi = 3{,}14159265...$. Wartością przybliżoną liczby π, często używaną w rachunkach, jest liczba 3,14. Jej błędem bezwzględnym jest na przykład liczba $\Delta a = 0{,}0016$. Wynika stąd, że dokładna wartość π jest zawarta między liczbami

$$3{,}14-0{,}0016 \leqslant \pi \leqslant 3{,}14+0{,}0016,$$

to znaczy znajduje się w przedziale

$$\langle 3{,}1384;\ 3{,}1416\rangle. \tag{2.2}$$

Oczywiście można przyjąć, że błędem bezwzględnym wartości przybliżonej $a = 3{,}14$ jest liczba $\Delta a = 0{,}002$ (nie będzie to sprzeczne z definicją 2.1). Wtedy moglibyśmy napisać, że liczba π leży w przedziale $\langle 3{,}138;\ 3{,}142\rangle$, co jest również prawdą. Jednakże przedział (2.2) podaje dokładniejszą informację o liczbie π.

Znajomość pary liczb a i Δa podaje tylko informację postaci (2.1) o wartości dokładnej. Informacja ta jest tym cenniejsza im mniejszy jest błąd bezwzględny Δa.

Mówiliśmy w § 1.2, że zjawisko przenoszenia się błędów z danych na wyniki można opisać za pomocą teorii błędów maksymalnych. Celem sformalizowania tej teorii, wprowadzimy jeszcze jedno pojęcie — liczbę przybliżoną.

D. 2.2. Jeśli a jest wartością przybliżoną dla wartości dokładnej A, obciążoną błędem bezwzględnym Δa, to parę liczb a, Δa, zapisaną w postaci

$$\overset{\Delta a}{a} \tag{2.3}$$

będziemy nazywali **liczbą przybliżoną** dla A.

Związek pomiędzy liczbą przybliżoną $\overset{\Delta a}{a}$ i dokładną wartością A wyrażamy za pomocą równości

$$A = \overset{\Delta a}{a}\ .$$

Równość tę należy rozumieć jako informację o wartości dokładnej A. Często tę równość czytamy w następujący sposób: A jest równa a z błędem Δa.

P. 2.2. Zgodnie z przykładem 2.1 możemy napisać

$$\pi = \overset{0{,}0016}{3{,}14}$$

i przeczytać: π równa się 3,14 z błędem 0,0016.

Cyfry błędu bezwzględnego (części górnej liczby przybliżonej) piszemy nad odpowiednimi cyframi wartości przybliżonej (części dolnej liczby przybliżonej), tzn. jedności nad jednościami, dziesiątki nad dziesiątkami itd. Na ogół błąd, tzn. część górna liczby przybliżonej, jest o wiele mniejszy od wartości przybliżonej. Dla uproszczenia zapisu pomijamy w części górnej zera poprzedzające pierwszą niezerową cyfrę, a znajdującą się na lewo od ostatniej cyfry części dolnej.

P. 2.3. Zamiast $\overset{0{,}0016}{3{,}14}$ piszemy $\overset{016}{3{,}14}$,

zamiast $\overset{0{,}061}{-0{,}572}$ piszemy $\overset{61}{-0{,}572}$.

2.2. Działania na liczbach przybliżonych

Zajmiemy się teraz zagadnieniem przenoszenia się błędu z argumentów operacji na wynik. Przeanalizujemy to zagadnienie tylko dla działań arytmetycznych: dodawania, odejmowania, mnożenia i dzielenia. Przed tym omówimy jeszcze pewne podstawowe przekształcenia

liczb przybliżonych, bardzo często stosowane w obliczeniach, a prowadzące do uproszczenia, czasem skomplikowanej postaci liczby przybliżonej. Mówiliśmy, że liczba przybliżona określa przedział, w którym zawarta jest wartość dokładna. Dlatego mówiąc o przekształceniach liczby przybliżonej możemy zawsze myśleć o przekształceniach przedziału. Musimy jednak zawsze pamiętać o podstawowej zasadzie teorii błędów maksymalnych.

Każda liczba przybliżona $\overset{\alpha}{a}$, związana z liczbą dokładną A musi określać przedział zawierający liczbę dokładną.

Wynika stąd, że daną liczbę przybliżoną $\overset{\alpha}{a}$, określającą przedział $\langle a-\alpha;\ a+\alpha\rangle$ można tylko tak przekształcać do nowej liczby $\overset{\beta}{b}$, aby przedział $\langle b-\beta;\ b+\beta\rangle$ zawierał w sobie poprzedni przedział.

D. 2.3. Jeśli liczby przybliżone $\overset{\alpha}{a}$ i $\overset{\beta}{b}$ są takie, że przedział $\langle a-\alpha;\ a+\alpha\rangle$ jest zawarty w przedziale $\langle b-\beta;\ b+\beta\rangle$, to mówimy, że **liczba** $\overset{\alpha}{a}$ **jest w przybliżeniu równa liczbie** $\overset{\beta}{b}$ i piszemy

$$\overset{\alpha}{a} \Rightarrow \overset{\beta}{b}. \tag{2.4}$$

Wprowadzona tu została pewna nowa relacja, zwana **relacją przybliżania.** Nie jest ona relacją zwrotną, bo z relacji $\overset{\alpha}{a} \Rightarrow \overset{\beta}{b}$ nie wynika $\overset{\beta}{b} \Rightarrow \overset{\alpha}{a}$. Strzałka wskazuje, w którym kierunku nastąpiło rozszerzenie przedziału.

Często stosowanym przekształceniem liczby przybliżonej jest jej zaokrąglanie.

T. 2.1. (Reguła zaokrąglania). *Dla dowolnej liczby przybliżonej* $\overset{\alpha}{a}$ *i dowolnej liczby rzeczywistej* b *zachodzi związek*

$$\overset{\alpha}{a} \Rightarrow \overset{\alpha+|a-b|}{b}. \tag{2.5}$$

Dowód. Należy wykazać, że

$$\langle a-\alpha;\ a+\alpha\rangle \subset \langle b-\alpha-|a-b|;\ b+\alpha+|a-b|\rangle.$$

Wiemy, że

$$-|a-b| \leqslant a-b \leqslant |a-b|, \text{ więc}$$
$$b-\alpha-|a-b| \leqslant b-\alpha+a-b = a-\alpha \leqslant a+\alpha =$$
$$= a-b+b+\alpha \leqslant |a-b|+b+\alpha.$$

Udowodniliśmy zatem, że

$$b-\alpha-|a-b| \leqslant a-\alpha \quad \text{oraz} \quad a+\alpha \leqslant |a-b|+b+\alpha.$$

Regułę zaokrąglania stosujemy wówczas, kiedy wynik działania arytmetycznego ma dużo cyfr. Wtedy można odrzucić ostatnie, zbędne, cyfry, pamiętając jednakże o odpowiednim zwiększeniu błędu bezwzględnego. Zwykle postępuje się tak, że jeśli pierwszą odrzuconą cyfrą jest 0, 1, 2, 3 lub 4, to cyfr pozostawionych w wartości przybliżonej nie zmieniamy, jeśli natomiast pierwszą odrzuconą cyfrą jest 5, 6, 7, 8 lub 9, to do pozostawionej części wartości przybliżonej dodajemy 1 na ostatnim zostawionym miejscu dziesiętnym. Taka zmiana liczby przybliżonej nazywa się **poprawnym zaokrągleniem.**

P. 2.4. Rozważmy liczbę przybliżoną $\overset{\alpha}{a} = 3{,}14159\overset{027}{\ }$. Zaokrąglając wartość przybliżoną do dwóch miejsc po przecinku, otrzymamy liczbę przybliżoną

$$\overset{\alpha}{a} \Rightarrow \overset{0{,}0000027+0{,}00159}{3{,}14} = 3{,}14\overset{015927}{\ }.$$

Zaokrąglając tę samą wartość przybliżoną do trzech cyfr po przecinku, otrzymamy

$$\overset{\alpha}{a} \Rightarrow \overset{0{,}0000027+|3{,}14159-3{,}142|}{3{,}142} = 3{,}142\overset{04127}{\ },$$

to znaczy przedział

$$\langle 3{,}1415873;\ 3{,}1424127\rangle. \tag{2.6}$$

W obu powyższych przypadkach zastosowaliśmy poprawne zaokrąglenie. Można by przy zaokrąglaniu do trzech miejsc po przecinku po prostu odrzucić pozostałe cyfry. Otrzymalibyśmy wówczas liczbę $3{,}141\overset{05927}{\ }$, to znaczy przedział

$$\langle 3{,}1404073;\ 3{,}1415927\rangle. \tag{2.7}$$

Z twierdzenia 2.1 wynika, że oba przedziały (2.6) i (2.7) zawierają dokładną wartość. Zauważmy jednak, że przedział (2.6) ma długość równą 0,0008254, natomiast długość drugiego przedziału wynosi 0,0011854. Ponieważ przedział (2.6) jest około półtora raza krótszy od przedziału (2.7), to wynika stąd, że poprawne zaokrąglenie pozostawia dokładniejszą informację o wartości dokładnej.

Stosując regułę zaokrąglania otrzymujemy liczbę przybliżoną z wielocyfrowym błędem. Ostatnie cyfry błędu nie są zbyt cenne, mało wpływają na nasze wiadomości o wartości dokładnej. Można więc pewną ilość cyfr błędu odrzucić. Należy jednak pamiętać, aby nie spowodować przy tym zwężenia przedziału. Korzysta się przy tym z tzw. reguły rozszerzania.

T. 2.2 (Reguła rozszerzania). *Jeśli* $\beta \geqslant \alpha$, *to*

$$\overset{\alpha}{a} \Rightarrow \overset{\beta}{a}.$$

Dowód twierdzenia jest oczywisty.

W praktyce z tej reguły korzysta się w ten sposób, że w górnej części liczby przybliżonej pozostawia się jedną lub dwie cyfry znaczące.

P. 2.5. Stosując regułę rozszerzania, możemy napisać:

$$\begin{array}{rcl} 015927 & & 016 \\ 3{,}14 & \Rightarrow & 3{,}14 \\ 041127 & & 05 \\ 3{,}142 & \Rightarrow & 3{,}142 \\ 05927 & & 06 \\ 3{,}141 & \Rightarrow & 3{,}141 \end{array}$$

Zajmiemy się teraz działaniami arytmetycznymi na liczbach przybliżonych. To pozwoli nam zorientować się w działaniu mechanizmu przenoszenia się błędów z danych na wyniki poszczególnych operacji. Wyjaśnimy wpierw ogólną zasadę wykonywania działań arytmetycznych na liczbach przybliżonych. Przypuśćmy, że wykonujemy działanie $\overset{\alpha}{a} \oslash \overset{\beta}{b}$, gdzie symbolem $\oslash$ oznaczyliśmy jedną z operacji arytmetycznych: $+$, $-$, $\cdot$, $:$. Wynikiem takiego działania jest znowu liczba przybliżona $\overset{\gamma}{c}$. (Zauważmy, że liczbę dokładną możemy również uważać za liczbę przybliżoną z błędem zero.) Wykonajmy teraz operacje

$x \oslash y$ na wszystkich możliwych parach x i y, gdzie: $x \in \langle a-\alpha;\ a+\beta\rangle = I$, $y \in \langle b-\beta;\ b+\beta\rangle = J$. (Symbol $x \in \langle u; v\rangle$ czytamy: x należy do przedziału $\langle u; v\rangle$.) Spośród wszystkich możliwych wyników istnieje najmniejszy

$$m = \min_{\substack{x \in I \\ y \in J}} (x \oslash y)$$

oraz największy

$$M = \max_{\substack{x \in I \\ y \in J}} (x \oslash y).$$

Przez wynik operacji arytmetycznej $\overset{\alpha}{a} \oslash \overset{\beta}{b}$ będziemy rozumieli liczbę przybliżoną $\overset{\gamma}{c}$, określającą przedział $\langle m; M\rangle$.

Mając przedział związany z liczbą przybliżoną, możemy już łatwo wyznaczyć samą liczbę przybliżoną. Mianowicie, części dolną i górną wyznaczamy ze wzorów:

$$c = \frac{m+M}{2}, \qquad \gamma = \frac{M-m}{2}.$$

P. 2.6. Pomnożyć liczby przybliżone $\overset{1}{5{,}1}$ i $\overset{1}{6{,}2}$.

Spośród wszystkich możliwych iloczynów postaci $x \cdot y$, gdzie $x \in \langle 5{,}0;\ 5{,}2\rangle$, $y \in \langle 6{,}1;\ 6{,}3\rangle$, najmniejszy jest równy 30,50, natomiast największy równa się 32,76. Zatem iloczyn liczb przybliżonych jest określony jako przedział $\langle 30{,}50;\ 32{,}76\rangle$. Przedział ten możemy zapisać w postaci liczby przybliżonej, przyjmując za część dolną jego środek, a za część górną — połowę jego długości (zgodnie z powyższymi wzorami dla c i γ). Otrzymamy wówczas

$$\overset{1}{5{,}1} \cdot \overset{1}{6{,}2} = \overset{1{,}13}{31{,}63}.$$

Stosując opisaną wyżej ogólną zasadę działań na liczbach przybliżonych, moglibyśmy łatwo wykonać dowolną operację arytmetyczną, a także każdą inną operację, na liczbach przybliżonych.

Aby jeszcze ułatwić wykonywanie działań arytmetycznych na liczbach przybliżonych, podajemy odpowiednie wzory.

T. 2.3 (Dodawanie i obejmowanie liczb przybliżonych). *Zachodzą związki*:

$$\overset{\alpha}{a}+\overset{\beta}{b}=\overset{\alpha+\beta}{a+b} \tag{2.8}$$

$$\overset{\alpha}{a}-\overset{\beta}{b}=\overset{\alpha+\beta}{a-b}. \tag{2.9}$$

Dowód. Niech liczby przybliżone $\overset{\alpha}{a}$ i $\overset{\beta}{b}$ będą związane z wartościami dokładnymi A i B, to znaczy, że

$$a-\alpha \leqslant A \leqslant a+\alpha,$$
$$b-\beta \leqslant B \leqslant b+\beta.$$

Dodając powyższe nierówności stronami, otrzymamy

$$(a+b)-(\alpha+\beta) \leqslant A+B \leqslant (a+b)+(\alpha+\beta).$$

Z tych nierówności widać od razu prawdziwość wzoru (2.8).

Dla dowodu wzoru na odejmowanie zauważmy, że jeśli liczba przybliżona $\overset{\beta}{b}$ jest związana z wartością dokładną B, to z wartością dokładną $-B$ jest związana liczba $\overset{\beta}{-b}$, ponieważ zmiana znaku wartości przybliżonej nie może zmienić jej błędu bezwzględnego. Mamy więc

$$\overset{\alpha}{a}-\overset{\beta}{b}=\overset{\alpha}{a}+\overset{\beta}{(-b)}.$$

Korzystając teraz ze wzoru na dodawanie liczb przybliżonych, otrzymujemy natychmiast (2.9).

P. 2.7.

$$\overset{92}{14{,}521}+\overset{11}{22{,}6}=\overset{1192}{37{,}121}\Rightarrow\overset{1202}{37{,}12}\Rightarrow\overset{13}{37{,}12}.$$

Zauważmy, że po wykonaniu dodawania według wzoru (2.8), zastosowaliśmy, w celu uproszczenia wyniku, zaokrąglenie, a następnie zastosowaliśmy regułę rozszerzania. Ale tu chodzi nie tylko o uproszczenie wyniku. Zastosowaliśmy się tu do pewnej praktycznej wskazówki. Wynik dodawania jest obarczony błędem już na pierwszym miejscu po przecinku. Zaleca się, aby nie zachowywać w wartości przybliżonej (części dolnej) więcej niż jednej (co najwyżej dwu) cyfry obarczonej błędem, ponieważ dalsze cyfry nie są już pewne. Dalsze przekształcenie miało na celu uproszczenie liczby przybliżonej.

Zajmiemy się teraz wzorem na mnożenie liczb przybliżonych. Dokładny wzór dla mnożenia jest nieco skomplikowany. Dlatego ograniczymy się do podania przybliżonego wzoru dla mnożenia.

T. 2.4 (Mnożenie przybliżone liczb przybliżonych). *Zachodzi związek*

$$\overset{\alpha}{a}\cdot\overset{\beta}{b}\Rightarrow\overset{|a|\beta+|b|\alpha+\alpha\beta}{ab} \tag{2.10}$$

Twierdzenia tego nie będziemy dowodzić. Zauważmy jedynie, że w (2.10) występuje relacja przybliżania zamiast równości. Wynika to stąd, że przedział występujący po prawej stronie przybliżania może być jeszcze mniejszy. Zilustrujemy to na przykładzie, w którym wykonamy mnożenie liczb przybliżonych dwoma sposobami, raz wykorzystując wzór (2.10), a drugi raz obliczając minimum i maksimum iloczynów, jak w przykładzie 2.6.

P. 2.8. Obliczyć iloczyn $\overset{\gamma}{c}=\overset{02}{2{,}1}\cdot\overset{01}{1{,}3}$. Korzystając ze wzoru (2.10) obliczamy: $\overset{\gamma}{c}\Rightarrow\overset{472}{2{,}73}$. Otrzymany iloczyn określa przedział $\langle 2{,}6828;\ 2{,}7772\rangle$. Postępując teraz jak w przykładzie 2.6, to znaczy obliczając najmniejszy i największy z iloczynów xy, gdzie x należy do przedziału $\langle 2{,}08;\ 2{,}12\rangle$, a y — do przedziału $\langle 1{,}29;\ 1{,}31\rangle$, znajdujemy: $m=2{,}6832$, $M=2{,}7772$. Wynika stąd, że dokładny wynik mnożenia jest zawarty w przedziale $\langle 2{,}6832;\ 2{,}7772\rangle$, węższym od przedziału uzyskanego wzorem (2.10).

Dla dzielenia liczb przybliżonych istnieje również dokładny wzór, niestety jeszcze bardziej skomplikowany, aniżeli dokładny wzór dla mnożenia. Dlatego ograniczymy się również do podania przybliżonego wzoru dla dzielenia. (Oba dokładne wzory dla mnożenia i dzielenia znajdzie czytelnik w [4].)

T. 2.5 (Dzielenie przybliżone liczb przybliżonych). *Jeśli* $\beta<|b|$, *to zachodzi związek*

$$\overset{\alpha}{a}:\overset{\beta}{b}\Rightarrow\overset{\gamma}{\frac{a}{b}}, \tag{2.11}$$

gdzie

$$\gamma=\frac{\alpha+\left|\frac{a}{b}\right|\beta}{|b|-\beta}.$$

Założenie $\beta < |b|$ jest konieczne, ponieważ w przeciwnym przypadku przedział $\langle b-\beta;\ b+\beta\rangle$ zawierałby zero, przez które nie wolno dzielić.

P. 2.9. Obliczyć iloraz $1{,}42^{015} : 3^{0{,}005}$.

Obliczamy tu

$$\frac{a}{b} = 0{,}47333\ldots, \quad \frac{\alpha+\left|\frac{a}{b}\right|\beta}{|b|-\beta} < \frac{0{,}0015+0{,}4734\cdot 0{,}005}{2{,}995} =$$

$$= \frac{0{,}003867}{2{,}995} < 0{,}0013.$$

Mamy więc

$$1{,}42^{015} : 3^{0{,}005} \Rightarrow 0{,}47333\ldots^{13} \Rightarrow 0{,}4733^{14}.$$

Zauważmy, że przy obliczaniu błędu (części górnej) ilorazu wykonywaliśmy zaokrąglenia z nadmiarem, aby być pewnym, że końcowy wynik zawiera w sobie dokładną wartość wielkości.

Na zakończenie tego paragrafu dodajmy jeszcze następujące uwagi. W każdym algorytmie występują działania, które należałoby wykonywać na wartościach dokładnych. Ponieważ jednak wartości dokładnych przeważnie nie znamy, to z konieczności wykonujemy poszczególne operacje na wartościach przybliżonych.

Należy jeszcze raz uświadomić sobie, że wartości przybliżone nie dają żadnej informacji o wartościach dokładnych. Dlatego przy wykonywaniu działań na wartościach przybliżonych należy również wykonywać działania na ich błędach bezwzględnych, albo (co na jedno wychodzi) należy wszystkie operacje wykonywać na liczbach przybliżonych, stosując podane wyżej wzory.

Zatem, przy realizacji algorytmu należałoby postąpić w następujący sposób. Każdą daną zapisać w postaci liczby przybliżonej, przy czym każdą liczbę dokładną możemy uważać jako liczbę przybliżoną z błędem zero. Następnie wszystkie operacje algorytmu należy wykonywać według wzorów (2.8) — (2.11). (Zauważmy, że maszyny wykonują przeważnie tylko cztery działania arytmetyczne.) Wtedy,

po realizacji algorytmu, otrzymalibyśmy wszystkie wyniki w postaci liczb przybliżonych, a tym samym otrzymalibyśmy od razu informację o błędach wyników. W praktyce takie postępowanie jest jednak rzadko stosowane ze względu na dużą pracochłonność. Najczęściej, wykonuje się działania tylko na wartościach przybliżonych, a analizę błędów wykonuje się osobno, na ogół poza maszyną.

P. 2.10. Obliczyć wartość wielomianu

$$w(x) = a_0x^4+a_1x^3+a_2x^2+a_3x+a_4,$$

dla $x = 2{,}1$.

Przyjmijmy wpierw, że współczynniki wielomianu są liczbami dokładnymi i równają się:

$$a_0 = 2{,}3,\ a_1 = 3,\ a_2 = -4{,}5,\ a_3 = 7{,}2,\ a_4 = -0{,}1.$$

Obliczenia wykonamy dwukrotnie, raz z dokładnością do dwóch, a drugi raz — do czterech miejsc po przecinku. To znaczy, że na przykład w pierwszym przypadku wszystkie wyniki będziemy zaokrąglać do dwóch miejsc po przecinku, stosując regułę zaokrąglania T. 2.1. Będziemy również stosowali regułę rozszerzania w przypadkach, gdy błąd będzie zawierać zbyt wiele cyfr.

Obliczamy:

$$x^2 = 2{,}1^{0}\cdot 2{,}1^{0} = 4{,}41^{0},$$

$$x^3 = 4{,}41^{0}\cdot 2{,}1^{0} = 9{,}261^{0} \Rightarrow 9{,}26^{01},$$

$$x^4 = 9{,}26^{01}\cdot 2{,}1^{0} \Rightarrow 19{,}446^{21} \Rightarrow 19{,}45^{061},$$

$$2{,}3\cdot x^4 = 2{,}3^{0}\cdot 19{,}45^{061} \Rightarrow 44{,}735^{1403} \Rightarrow 44{,}74^{1903} \Rightarrow 44{,}74^{2},$$

$$3x^3 = 3^{0}\cdot 9{,}26^{01} \Rightarrow 27{,}78^{03},$$

$$-4{,}5x^2 = -4{,}5^{0}\cdot 4{,}41^{0} \Rightarrow -19{,}845^{0} \Rightarrow -19{,}85^{05},$$

$$7{,}2x = 7{,}2^{0}\cdot 2{,}1^{0} \Rightarrow 15{,}12^{0}.$$

Ostatecznie obliczamy

$$w(2{,}1) \Rightarrow 44{,}74^{2}+27{,}78^{03} -19{,}85^{05} +15{,}12^{0}-0{,}1^{0} = 67{,}69^{28}.$$

Wykonajmy teraz wszystkie obliczenia z dokładnością do czterech miejsc po przecinku:

$$x^2 = 2{,}1 \cdot 2{,}1 = 4{,}41,$$

$$x^3 = 4{,}41 \cdot 2{,}1 = 9{,}261,$$

$$x^4 = 9{,}261 \cdot 2{,}1 = 19{,}4481,$$

$$2{,}3x^4 = 2{,}3 \cdot 19{,}4481 = 44{,}73063 \Rightarrow 44{,}7306\ ,$$

$$3x^3 = 3 \cdot 9{,}261 = 27{,}783,$$

$$-4{,}5x^2 = -4{,}5 \cdot 4{,}41 = -19{,}845,$$

$$7{,}2x = 7{,}2 \cdot 2{,}1 = 15{,}12.$$

Ostatecznie obliczamy

$$w(2{,}1) \Rightarrow 44{,}7306 + 27{,}783 - 19{,}845 + 15{,}12 - 0{,}1 = 67{,}6886\ .$$

Zauważmy, że dokładność wyniku znacznie wzrosła w przypadku wykonania działań z większą dokładnością. Gdybyśmy obliczenia prowadzili jeszcze dokładniej, to otrzymalibyśmy jeszcze dokładniejszy wynik. Zwiększenie dokładności jest tu możliwe dzięki temu, że dane wejściowe są dokładne.

Zupełnie inaczej wygląda sprawa wtedy, kiedy dane wejściowe są obarczone błędami. Oczywiste jest, że wówczas zwiększanie dokładności obliczeń nie może zwiększyć dokładności wyników. Jeżeli na przykład dane są obarczone błędami już na drugim miejscu po przecinku, to niemożliwe jest uzyskanie wyników obarczonych błędami dopiero na trzecim czy dalszych miejscach po przecinku. Aby przekonać się o tym, wykonajmy jeszcze raz wszystkie obliczenia, zakładając, że współczynniki są obarczone błędami i równają się:

$$a_0 = 2{,}3\ ,\ a_1 = 3,\ a_2 = -4{,}5\ ,\ a_3 = 7{,}2\ ,\ a_4 = -0{,}1\ .$$

Wykonajmy wpierw obliczenia z dokładnością do dwóch miejsc po przecinku. Potęgi x będą takie jak policzono wyżej, ponieważ

obliczenia prowadzimy dla dokładnej wartości $x = 2{,}1$. Dalej obliczamy:

$$2{,}3\ x^4 = 2{,}3 \cdot 19{,}45 \Rightarrow 44{,}735 \Rightarrow 44{,}74 \Rightarrow 44{,}74,$$

$$3x^3 = 3 \cdot 9{,}26 \Rightarrow 27{,}78\ ,$$

$$-4{,}5\ x^2 = -4{,}5 \cdot 4{,}41 \Rightarrow -19{,}845 \Rightarrow -19{,}85 \Rightarrow -19{,}85\ ,$$

$$7{,}2\ x = 7{,}2 \cdot 2{,}1 \Rightarrow 15{,}12\ .$$

I ostatecznie

$$w(2{,}1) \Rightarrow 44{,}74 + 27{,}78 - 19{,}85 + 15{,}12 - 0{,}1 = 67{,}69 \Rightarrow 67{,}69.$$

Wykonajmy teraz te same obliczenia z dokładnością czterech miejsc po przecinku:

$$2{,}3 \cdot 19{,}4481 \Rightarrow 44{,}73063 \Rightarrow 44{,}7306 \Rightarrow 44{,}7306,$$

$$3 \cdot 9{,}261 = 27{,}783,$$

$$-4{,}5 \cdot 4{,}41 \Rightarrow -19{,}845\ ,$$

$$7{,}2 \cdot 2{,}1 \Rightarrow 15{,}12\cdot$$

Ostatecznie obliczamy

$$w(2{,}1) \Rightarrow 67{,}6886 \Rightarrow 67{,}69 \Rightarrow 67{,}69.$$

Zauważmy, że w przypadku kiedy dane wejściowe były dokładne, to zwiększenie dokładności obliczeń z dwóch do czterech miejsc po przecinku spowodowało prawie tysiąckrotne zmniejszenie błędu bezwzględnego. Natomiast w przypadku, gdy dane wejściowe były obarczone błędami, wówczas zwiększenie dokładności obliczeń prawie nie zmniejszyło błędu bezwzględnego obliczeń.

Takiego wyniku należało się spodziewać. Już w przykładzie P. 2.7 zwróciliśmy uwagę na to, że w czasie obliczeń zaleca się w wartości przybliżonej (części dolnej) zachowywać co najwyżej dwie cyfry

obarczone błędem, ponieważ dalsze cyfry są już niepewne. Mając to na uwadze, powinniśmy byli od razu zrezygnować z obliczeń z dokładnością do czterech miejsc po przecinku w przypadku, gdy w danych już drugie miejsce po przecinku było obarczone błędem.

Na zakończenie tego paragrafu wprowadzimy jeszcze jedno pojęcie — **cyfry znaczące.** W powyższym przykładzie żądaliśmy, aby obliczenia były prowadzone z dokładnością kilku miejsc po przecinku. W niektórych przypadkach można tak określać dokładność obliczeń. Bardziej sensownym jednak jest określanie dokładności obliczeń za pomocą ilości cyfr znaczących.

Niech liczba A będzie dokładną wartością pewnej wielkości, posiadającą na ogół nieskończone rozwinięcie dziesiętne. Cyfry tego rozwinięcia numerujemy zgodnie z ich znaczeniem pozycyjnym: k-ta cyfra po przecinku (k-ta pozycja) ma numer $-k$, cyfra jednostek ma numer 0, cyfra dziesiątek — numer 1 itd.

P. 2.11.

Numery cyfr (pozycji) :	2	1	0	−1	−2	−3	−4	−5	−6
Liczba A		3	7	4,	0	2	7	6	9	4

D. 2.4. Jeśli a jest przybliżeniem dokładnej wartości A, to k-tą cyfrę dziesiętną liczby a nazwiemy **znaczącą** wtedy, gdy

$$|A-a| \leqslant \frac{1}{2}10^k \quad \text{oraz} \quad |a| \geqslant 10^k.$$

Wynika stąd, że każda cyfra poprawnie zaokrąglonej liczby, począwszy od pierwszej różnej od zera, jest znaczącą.

P. 2.12. Niech wartość dokładna $A = 0{,}07658676\ldots$. W wyniku poprawnego zaokrąglenia do 6 miejsc po przecinku otrzymamy wartość przybliżoną $a = 0{,}076587$. Ostatnia cyfra jest znacząca, ponieważ

$$|A-a| \leqslant 0{,}00000024 < \frac{1}{2}10^{-6}, \quad |a| \geqslant 10^{-6}.$$

Z definicji D. 2.4 wynika, że w liczbie a cyframi znaczącymi są także cyfry o numerach: −2, −3, −4, −5. Możemy też powiedzieć, że liczba a ma pięć cyfr znaczących.

Liczbę określającą ilość cyfr znaczących bardzo często używa się jako miary dokładności obliczeń. Jest to inna miara dokładności aniżeli błąd bezwzględny. Bardziej związana jest z błędem względnym, o którym mówimy w następnym paragrafie.

2.3. Błąd względny

W paragrafie 2.2 wprowadziliśmy pojęcie błędu bezwzględnego. Ta miara dokładności wartości przybliżonej nie jest jedyną i nie zawsze najbardziej wygodną. Jeśli na przykład zmierzymy długość pokoju z dokładnością do 1 mm = 0,001 m i otrzymamy wynik 5,762 m oraz z taką samą dokładnością zmierzymy długość ostrza żyletki i otrzymamy wynik 34 mm = 0,034 m, to oczywiście pierwszy pomiar jest o wiele dokładniejszy od drugiego. Stąd wynika, że z dwu liczb przybliżonych $\overset{1}{5{,}762}$ i $\overset{1}{0{,}034}$ pierwsza jest dokładniejsza od drugiej, pomimo że obie są obarczone takim samym błędem bezwzględnym. Określimy teraz dokładniej w jakim sensie pierwsza z podanych liczb przybliżonych jest dokładniejsza od drugiej. W tym celu jako miarę dokładności liczb przybliżonych wprowadzimy błąd względny.

D. 2.5. Błędem względnym wartości przybliżonej a obarczonej błędem bezwzględnym Δa nazywamy liczbę

$$\varepsilon a = \frac{\Delta a}{|a|}.$$

Błąd względny jest więc określony tylko dla niezerowych wartości przybliżonych i — podobnie jak błąd bezwzględny — jest zawsze liczbą nieujemną, którą w razie potrzeby można dowolnie powiększyć. Łatwo obliczamy, że błąd wartości przybliżonej 5,762 m, obarczonej błędem bezwzględnym 0,001 m, jest równy 0,00017, natomiast błąd względny liczby przybliżonej 0,034 m, obarczonej takim samym błędem bezwzględnym, jest większy około 170 razy i wynosi 0,029.

Należy podkreślić, że błąd bezwzględny jest liczbą mianowaną i ma ten sam wymiar co wartość przybliżona a, natomiast błąd względny, będąc ilorazem dwu liczb mianowanych, jest liczbą niemianowaną i bywa często wyrażany w procentach.

P. 2.13. Zważono 5 t pszenicy z dokładnością do 10 kg oraz 1 g pewnej substancji, potrzebnej dla sporządzenia lekarstwa, z dokładnością 0,05 g. Który z pomiarów jest dokładniejszy?

Ilość pszenicy $a = 5000$ kg jest wartością przybliżoną, obarczoną błędem bezwzględnym $\Delta a = 10$ kg. Błąd względny

$$\varepsilon a = \frac{\Delta a}{|a|} = \frac{10}{5000} = 0{,}002 = 0{,}2\%.$$

Natomiast ilość substancji $b = 1$ g jest wartością przybliżoną, obarczoną błędem bezwzględnym $\Delta b = 0{,}05$ g. Zatem

$$\varepsilon b = \frac{0{,}05}{1} = 0{,}05 = 5\%.$$

Wynika stąd, że pierwszy pomiar jest 25 razy dokładniejszy od drugiego. Można by właściwie powiedzieć, że ważenie pszenicy odbywało się zbyt dokładnie. Natomiast ważenie 1 g części składowej lekarstwa z dokładnością 5% jest na pewno zbyt niedokładne. Tak sporządzone lekarstwo może się okazać szkodliwe dla organizmu.

Jeśli na przykład zażądalibyśmy, aby 1 g substancji odważyć z dokładnością 0,1%, to wówczas z równości

$$\frac{\Delta b}{1} = 0{,}001$$

obliczamy, że $\Delta b = 0{,}001$. Ważenie musi się odbywać z dokładnością do 1 mg.

Zaletą błędu względnego jest to, że przy jego stosowaniu niektóre wzory na błędy wartości wyników działań arytmetycznych są prostsze niż przy stosowaniu błędu bezwzględnego.

T. 2.6. *Zachodzi wzór*

$$\varepsilon(ab) = \varepsilon a + \varepsilon b + \varepsilon a \cdot \varepsilon b. \tag{2.12}$$

Wzór ten należy interpretować następująco: błąd względny wartości przybliżonej iloczynu wartości dokładnych AB, oznaczony symbolem $\varepsilon(ab)$, wyraża się wzorem (2.12) przez błędy wartości przybliżonych czynników A i B.

Dowód T. 2.6 wynika wprost z T. 2.4 i D. 2.5:

$$\varepsilon(ab) = \frac{|a|\beta + |b|\alpha + \alpha\beta}{|ab|} = \frac{\beta}{|b|} + \frac{\alpha}{|a|} + \frac{\alpha}{|a|} \cdot \frac{\beta}{|b|} = \varepsilon b + \varepsilon a + \varepsilon a \cdot \varepsilon b.$$

T. 2.7. *Jeśli* $\varepsilon b < 1$, *to*

$$\varepsilon\left(\frac{a}{b}\right) = \frac{\varepsilon a + \varepsilon b}{1 - \varepsilon b}. \tag{2.13}$$

Dowód przeprowadza się podobnie jak w przypadku T. 2.6, z tym, że należy tu skorzystać z T. 2.5 i D. 2.5.

Dodajmy jeszcze na zakończenie rozdziału następujące uwagi. Twierdzenia występujące w teorii błędów maksymalnych pozwalają wyznaczyć maksymalny błąd (względny lub bezwzględny), jaki może wystąpić po wykonaniu działań na liczbach przybliżonych. W rzeczywistości może się zdarzyć, że nie wystąpi tak duży błąd, jak to wynika z teorii błędów maksymalnych.

P. 2.14. Załóżmy, że wartości dokładne 0,753821 i 0,616205 zaokrąglamy do dwóch cyfr po przecinku, a następnie dodajemy. Otrzymamy

$$\begin{array}{ccc} 03821 & 03795 & 07616 \\ 0{,}75 + & 0{,}62 & = 1{,}37 \end{array}.$$

Z powyższego rachunku wynika, że przybliżona suma 1,37 różni się od sumy dokładnej co najwyżej o 0,007616. W rzeczywistości dokładna suma podanych liczb jest równa 1,370026 i różni się od wyliczonej sumy przybliżonej tylko o 0,000026. Zatem oszacowanie błędu za pomocą teorii błędów maksymalnych jest w tym przypadku bardzo niedokładne.

Był to przykład, gdzie nastąpiła redukcja błędów zaokrągleń. Oczywiście, tak samo może się zdarzyć w wielu innych przypadkach. Nie jest jednak łatwo wyróżnić te przypadki, ponieważ zjawisko przenoszenia się błędów jest przypadkowe. Dlatego to zjawisko lepiej opisuje statystyczna teoria błędów. Teoria ta uwzględnia możliwość znoszenia się błędów, co zauważyliśmy w przykładzie 2.14. Teoria statystyczna bada przy określonych założeniach nie największe możliwe błędy, ale wartości oczekiwane tych błędów, prawdopodobieństwa występowania błędów określonej wielkości itd. Nie będziemy tej teorii rozważać w niniejszym podręczniku. Pewne informacje na ten temat można znaleźć w [1], [3] i [4].

1. Dane są wartości dokładne następujących wielkości: $\pi = 3{,}1415926\ldots$, $\sqrt{2} = 1{,}4142136\ldots$, $\sqrt{3} = 1{,}7320508\ldots$, $e = 2{,}7182818\ldots$, $\log 2 = 0{,}6931472\ldots$, $85\pi = 267{,}0354$.

(a) Dokonać poprawnego zaokrąglenia każdej z podanych wartości do pięciu cyfr znaczących i wyznaczyć błąd bezwzględny każdej z otrzymanych w ten sposób wartości przybliżonej.

(b) Zbudować liczby przybliżone związane z wartościami dokładnymi podanymi w 1, przyjmując jako wartości przybliżone ich zaokrąglenia do pięciu miejsc znaczących.

2. Dokładna wartość A pewnej wielkości jest zawarta w przedziale $\langle 0{,}3763;\ 0{,}37634868\rangle$. Zbudować liczbę przybliżoną związaną z A. Zaokrąglić ją do 6 miejsc po przecinku. Dlaczego nie ma sensu pozostawiać w przybliżonej wartości więcej niż 6 miejsc po przecinku? Do otrzymanego wyniku zastosować jeszcze regułę rozszerzania.

3. Korzystając z twierdzeń: 2.3, 2.4 i 2.5, wyprowadzić wzory dla następujących działań: (i) $\overset{\alpha}{a}\cdot\overset{0}{b}$, (ii) $\overset{\alpha}{a}:\overset{0}{b}$, (iii) $(\overset{\alpha}{a})^2$, (iv) $1:\overset{\alpha}{a}$, (v) $\overset{\alpha}{a}\cdot\overset{\beta}{b}\cdot\overset{\gamma}{c}$.

4. Dane są liczby przybliżone: $\overset{1}{a_1} = 1{,}3503$, $\overset{5}{a_2} = 2{,}3750$, $\overset{2}{a_3} = 3{,}4728$, $\overset{3}{a_4} = 2{,}8506$.

Obliczyć dwoma sposobami sumę tych liczb:

(i) zaokrąglić wpierw każdą liczbę do dwu miejsc po przecinku, a potem dodać,

(ii) najpierw dodać wszystkie liczby, a potem zaokrąglić do dwu miejsc po przecinku.

Który z powyższych sposobów jest dokładniejszy?

5. Dane są liczby przybliżone:

$\overset{1}{a_1} = 1{,}22$, $\overset{2}{a_2} = 100{,}11$, $\overset{1}{a_3} = 0{,}01$.

Obliczyć dwa następujące iloczyny: $a_3a_1a_2$ oraz $a_2a_1a_3$, dokonując po każdym mnożeniu zaokrąglenia do dwu miejsc po przecinku. Czy wartość iloczynu zależy od kolejności czynników? Który z dwu obliczonych iloczynów jest dokładniejszy? W jakiej kolejności należy pomnożyć powyższe trzy liczby, aby uzyskać iloczyn obarczony najmniejszym błędem?

6. Obliczyć sumę

$$s = 1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{10},$$

przedstawiając każdy ze składników w postaci liczby przybliżonej, poprawnie zaokrąglonej do czterech miejsc po przecinku.

7. Niech $f(x) = \dfrac{2x-0{,}3}{4x+6}$.

Obliczyć błąd bezwzględny oraz błąd względny funkcji $f(x)$ dla $x = \overset{1}{1{,}01}$.

8. Za pomocą linijki, na której jest podziałka milimetrowa, zmierzono długości dwu prętów o przekroju 1 cm² i uzyskano wyniki: $l_1 = 362$ mm, $l_2 = 34$ mm. Zakładając, że błąd pomiaru może wynosić co najwyżej 0,5 mm (połowa najmniejszej podziałki), obliczyć błędy względne obu przybliżonych wartości. Obliczyć masę każdego z prętów, wiedząc, że pręty zbudowane są z miedzi, której ciężar właściwy wynosi $8{,}93\ \overset{05}{\ }$ G/cm³. Obliczyć błędy względne każdej z uzyskanych mas.

9. Udowodnić prawdziwość wzoru 2.13.

10. Prędkość w ruchu jednostajnym obliczamy ze wzoru

$$v = \frac{s}{t}.$$

Przyjmijmy, że droga s zmierzona z dokładnością do 1 m wynosi 1736 m, natomiast czas t zmierzony z dokładnością do 0,1 s, wynosi 137,4 s. Obliczyć błąd względny prędkości v.

11. Wielkość r oblicza się ze wzoru

$$r = 16t^2.$$

Określić dopuszczalny błąd względny $\varepsilon(t)$ taki, żeby błąd względny wartości przybliżonej r był mniejszy od 0,01.

Część II

Maszyny matematyczne i programowanie

Rozdział 1

MASZYNY MATEMATYCZNE

1.1. Arytmetyczne podstawy maszyn cyfrowych

A. Systemy liczenia

W codziennym życiu posługujemy się najczęściej dwoma systemami liczenia: dziesiętnym systemem pozycyjnym i systemem rzymskim. W systemie rzymskim liczby zapisuje się za pomocą cyfr: I, V, X, L, C, D, M itd. (jeden, pięć, dziesięć, pięćdziesiąt, sto, pięćset, tysiąc itd.). Na przykład zapis MCMLXXII oznacza liczbę 1972. Ponieważ wykonywanie działań arytmetycznych na liczbach w systemie rzymskim nie jest łatwe, dlatego w tym systemie nie wykonuje się żadnych rachunków. Wygodnym dla obliczeń jest dziesiętny układ pozycyjny, w którym dla zapisania dowolnej liczby używa się cyfr: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. W systemie dziesiętnym wyróżnia się pozycje: jedności, dziesiątek, setek, tysięcy itd., a na prawo od przecinka dziesiętnego: dziesiętnych, setnych, tysięcznych itd.

P. 1.1. Liczba dziesiętna $l = 3765{,}762$ ma wartość

$$l = 3\cdot 10^3+7\cdot 10^2+6\cdot 10+5+7\cdot 10^{-1}+6\cdot 10^{-2}+2\cdot 10^{-3}.$$

Każdą liczbę w systemie dziesiętnym możemy zapisać w postaci

$$a_n a_{n-1} \ldots a_1 a_0,\ a_{-1}a_{-2} \ldots a_{-s}, \tag{1.1}$$

gdzie n, s — dowolne liczby całkowite nieujemne, natomiast liczby a_i mogą przyjmować wartości: 0, 1, 2, ..., 9.
Wartość liczby l oblicza się ze wzoru:

$$l = a_n 10^n+a_{n-1}10^{n-1}+ \ldots +a_1 10+a_0+a_{-1}10^{-1}+ \ldots +a_{-s}10^{-s}.$$

Liczbę dziesięć nazywamy w tym systemie **podstawą liczenia.** Możemy oczywiście prawą stronę (1.1) traktować jako liczbę przedstawioną w systemie o dowolnej podstawie r (r — liczba naturalna), z tym, że wówczas a_i może przyjmować wartości: 0, 1, 2, ..., $r-1$, które w tym przypadku nazywamy cyframi systemu o podstawie r. Wartość liczby (1.1) w systemie o podstawie r oblicza się ze wzoru:

$$l = a_n r^n+a_{n-1}r^{n-1}+ \ldots +a_1 r+a_0+a_{-1}r^{-1}+ \ldots +a_{-s}r^{-s}.$$

P. 1.2. W układzie trójkowym wykorzystuje się tylko trzy cyfry: 0, 1, 2. Liczba

$$l = 154 = 1\cdot 3^4+2\cdot 3^3+2\cdot 3^2+0\cdot 3+1$$

przyjmie w układzie trójkowym postać:

$$l = 12201.$$

W elektronicznych maszynach cyfrowych bardzo wygodnym okazał się **system pozycyjny dwójkowy,** zwany też **systemem binarnym,** w którym cyfry oznacza się symbolami: 0, 1. Każdą liczbę w systemie binarnym możemy również zapisać w postaci (1.1), z tym, że a_i w tym przypadku mogą przyjmować tylko dwie wartości: 0, 1. Wartość liczby zapisanej w systemie dwójkowym oblicza się ze wzoru:

$$l = a_n 2^n+a_{n-1}2^{n-1}+ \ldots +a_1 2+a_0+a_{-1}2^{-1}+ \ldots a_{-s}2^{-s}.$$

Poszczególne cyfry liczby binarnej nazywane są **bitami.**

P. 1.3. Liczbę

$$l = 54{,}25 = 1\cdot 2^5+1\cdot 2^4+0\cdot 2^3+1\cdot 2^2+1\cdot 2^1+0\cdot 2^0+0\cdot 2^{-1}+1\cdot 2^{-2}$$

możemy zapisać w systemie binarnym w postaci

$$l = 110110{,}01.$$

P. 1.4. Kolejne liczby naturalne w systemie binarnym mają postać:

1	1	6	110	11	1011	16	10000
2	10	7	111	12	1100	17	10001
3	11	8	1000	13	1101	18	10010
4	100	9	1001	14	1110	19	10011
5	101	10	1010	15	1111	20	10100

Działania arytmetyczne na liczbach w systemie dwójkowym wykonuje się bardzo łatwo. Wystarczy zapamiętać dwie bardzo proste tabliczki działań.

Tabliczka dodawania	Tabliczka mnożenia
$0+0=0$	$0\cdot 0=0$
$0+1=1$	$0\cdot 1=0$
$1+0=1$	$1\cdot 0=0$
$1+1=10$	$1\cdot 1=1$

P. 1.5. Algorytmy działań arytmetycznych w systemie binarnym są takie same jak w systemie dziesiętnym, przy czym wykorzystuje się bardzo proste tabliczki dodawania i mnożenia. Podajemy niżej przykłady wszystkich czterech działań arytmetycznych. Szczegóły pozostawiamy do samodzielnego przeanalizowania.

Dodawanie	Odejmowanie	Mnożenie
1100111,011	10110,1101	10,11
+10011,111	−10001,1111	110,10
1111011,010	100,1110	1011
		1011
		1011
		10001,111

Dzielenie

```
11011101101:1001 = 11000101
1001
 1001
 1001
    1011
    1001
      1001
      1001
      0000
```


B. Arytmetyka stałoprzecinkowa i zmiennoprzecinkowa

Pamięć[1]) maszyny cyfrowej jest najczęściej podzielona na pewną liczbę oddzielnych fragmentów zwanych słowami. Każde **słowo maszynowe** zawiera tę samą liczbę bitów (cyfr binarnych). Słowo maszynowe może zawierać określoną informację, np. liczbę, rozkaz itp.

Istnieją różne sposoby przedstawiania liczb w pamięci. Na przykład na pierwszym bicie wpisany jest znak liczby (zero dla liczb dodatnich, jedynka dla liczb ujemnych), a na pozostałych bitach słowa wpisany jest moduł liczby. W niektórych maszynach, np. w maszynie ODRA 1204, liczby są przedstawiane w tzw. postaci uzupełnieniowej. Nie będziemy się jednak tym zajmować.

W dalszych rozważaniach, celem łatwiejszego zrozumienia, będziemy zakładali, że jedno słowo pamięci maszyny zawiera liczby dziesiętne o określonej długości (np. 10-cyfrowe). Rozważania te nietrudno przenieść na system binarny.

Maszyny cyfrowe są przeważnie tak zaprojektowane, że każda liczba d-cyfrowa ma z założenia przecinek dziesiętny przed pierwszą cyfrą, zaraz po znaku. Wobec tego zakłada się, że wszystkie liczby w maszynie są co do modułu mniejsze od 1. Dlatego przy dodawaniu, odejmowaniu i dzieleniu liczby powinny być takie, aby ich suma, różnica lub iloraz były również mniejsze od 1. Jeśli wynik działania nie jest mniejszy od jedynki, to — jak mówimy — powstaje nadmiar. Gdy maszyna wykonuje działania arytmetyczne na liczbach opisanych wyżej to mówimy, że maszyna pracuje w arytmetyce stałoprzecinkowej. Nietrudno dojść do wniosku, że planowanie obliczeń w arytmetyce stałoprzecinkowej jest żmudne, ze względu na żądanie, aby wszystkie dane, wyniki pośrednie i wyniki końcowe były mniejsze od 1. Zachodzi wówczas konieczność skalowania, czego nie będziemy tu dokładnie wyjaśniać. Nadmienimy tylko, że operacja skalowania umożliwia na przykład traktowanie liczb stałoprzecinkowych jako liczb całkowitych. Celem ułatwienia planowania obliczeń, wszystkie maszyny mogą wykonywać również tzw. działania zmiennoprzecinkowe. Działania zmiennoprzecinkowe realizuje się albo przez odpowiednie programy działań zmiennoprzecinkowych, albo za pomocą odpowiednich układów elektronicznych.

[1]) O pamięci maszyny powiemy więcej w § 1.2.

W przypadku działań zmiennoprzecinkowych, d-cyfrowe słowo jest podzielone na dwie części x i y, mające odpowiednio m cyfr i $d-m$ cyfr. To słowo jest interpretowane jako liczba postaci $x \cdot 10^y$ (w maszynach binarnych $x \cdot 2^y$). Przy tym zakłada się, że część x, zwana **mantysą,** jest znowu liczbą co do modułu mniejszą od 1, a y — **cecha** liczby — jest liczbą całkowitą.

P. 1.6. Jeśli przyjmiemy $d = 10$, $m = 8$, to liczba

$$l = \underbrace{5767010006}_{y \quad y}$$

będzie interpretowana jako liczba

$$l = 0{,}576701 \cdot 10^6.$$

Mantysa i cecha mogą być ujemne i dodatnie, ale przeważnie w liczbie zmiennoprzecinkowej jest dopuszczalny tylko jeden znak. Aby obejść tę sprzeczność często cechę zwiększa się o pewną stałą liczbę. Na przykład w przypadku $d = 10$, $d-m = 2$, można zwiększyć cechę o 50. Wtedy liczbę zmiennoprzecinkową o częściach x (mantysa) i y (cecha) interpretuje się jako $x \cdot 10^{y-50}$.

P. 1.7. Jeżeli zwiększymy cechę o 50, liczba z przykładu 1.6 będzie interpretowana jako

$$0{,}576701 \cdot 10^{-44}.$$

Natomiast liczbę $0{,}576701 \cdot 10^6$ należy wówczas zapisać w postaci:

$$\underbrace{5767010056.}_{x \quad y}$$

Przy planowaniu obliczeń i układaniu programu oraz przygotowywaniu danych nie ma potrzeby zapisywać liczb z podziałem na mantysę i cechę. Byłoby to bardzo uciążliwe, szczególnie w systemie binarnym. Wszystkie dane przygotowuje się, na ogół, w postaci ogólnie przyjętej (z pewnymi tylko modyfikacjami). Przedstawienie liczb w maszynie w odpowiedniej postaci (stało- czy zmiennoprzecinkowej) odbywa się automatycznie, za pomocą odpowiednich programów.

Podamy jeszcze kilka danych o organizacji zapisu liczb w maszynie ODRA 1204. W pamięci maszyny ODRA 1204 mogą być umieszczane następujące liczby:

a) stałoprzecinkowe krótkie,
b) stałoprzecinkowe długie,
c) zmiennoprzecinkowe.

Liczba stałoprzecinkowa krótka zajmuje 24 pozycje binarne, ponumerowane od 0 do 23 (z lewej do prawej). Znak liczby zapisuje się na pozycji zerowej (1 oznacza liczbę ujemną, 0 — liczbę dodatnią).

Liczba stałoprzecinkowa długa składa się z 48 pozycji binarnych, ponumerowanych od 0 do 47. Znak liczby znajduje się również na pozycji zerowej.

Liczba zmiennoprzecinkowa zajmuje 48 pozycji binarnych, ponumerowanych od 0 do 47. Mantysa m zajmuje 38 pozycji (od 0 do 37), przy czym na zerowej pozycji jest umieszczony znak mantysy. Cecha c zajmuje 10 pozycji (od 38 do 47). Mantysa przyjmuje wartości od -1 do $1-2^{-37}$, a cecha zwiększona o 512 przyjmuje wartości od 1 do 1023. Z przyjętej struktury wynika, że liczby zmiennoprzecinkowe

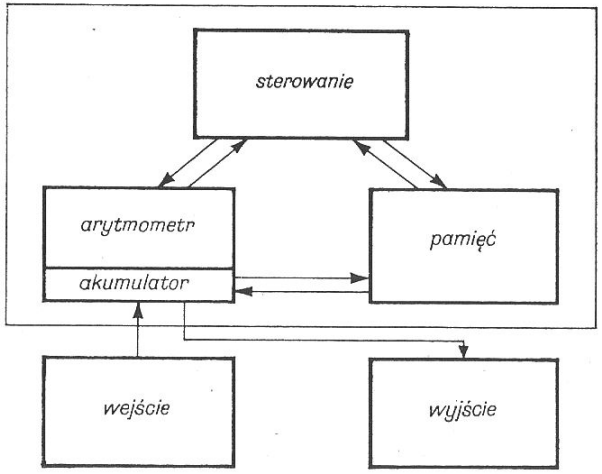
$$l = m \cdot 2^c$$

są zawarte w przedziale $-2^{511} \leqslant l \leqslant (1-2^{-37})2^{511}$,

to znaczy w przedziale $-2^{511} \leqslant l \leqslant 2^{511}-2^{474}$.

1.2. Organizacja maszyn cyfrowych

W każdej nowoczesnej maszynie cyfrowej można wyróżnić pięć podstawowych bloków: *wejście, wyjście, pamięć, arytmometr, sterowanie.*



Rys. 1.1. Schemat blokowy maszyny cyfrowej

Blok wejścia służy do wprowadzenia do maszyny wszelkiej informacji niezbędnej w procesie obliczeń.

Blok wyjścia służy do wyprowadzania wyników obliczeń.

Nośnikami informacji wprowadzanej są najczęściej: papierowa taśma perforowana, karty perforowane, taśma magnetyczna, ołówki świetlne itp. Na tych samych nośnikach wyprowadza się wyniki z maszyny. Powszechnie stosuje się także bezpośrednie drukowanie wyników lub wyświetlanie ich na ekranach.

Pamięć służy do przechowywania informacji zarówno wejściowej jak i pośredniej, wytworzonej w procesie liczenia, a niezbędnej dla jego kontynuacji. Można wyróżnić dwa rodzaje pamięci: operacyjną (wewnętrzną) oraz zewnętrzną.

Pamięć operacyjna charakteryzuje się szybkim do niej dostępem (zapisanie informacji do takiej pamięci lub jej odczytanie trwa około 10^{-6}s). Pamięć operacyjna (zrealizowana najczęściej na rdzeniach ferrytowych) składa się z komórek o określonej długości, ponumerowanych od zera do pewnej liczby N. Numer przyporządkowany komórce nazywamy jej **adresem.** Na przykład w maszynie ODRA 1204 jedna komórka zawiera 24 bity, czyli tzw. słowo krótkie. W takiej komórce może być wpisana liczba stałoprzecinkowa krótka lub rozkaz. Natomiast dla zapisu liczb stałoprzecinkowych długich oraz liczb zmiennoprzecinkowych wykorzystuje się w maszynie ODRA 1204 dwie komórki o kolejnych numerach (np. $k-1$ i k). Pamięć operacyjna jest na ogół nieduża. Maksymalna pamięć operacyjna ODRY 1204 może mieć pojemność 64 K komórek krótkich ($K = 1024$), ponumerowanych jednolicie od 0 do 65535.

Pamięć zewnętrzna maszyny służy jako magazyn informacji, która w odpowiednim momencie może być stamtąd pobrana do pamięci operacyjnej. Pojemność tej pamięci może być nieograniczona. Sięga ona miliardów bajtów (1 bajt = 8 bitów). Pamięci zewnętrzne zrealizowane są na bębnach magnetycznych, taśmach magnetycznych lub dyskach.

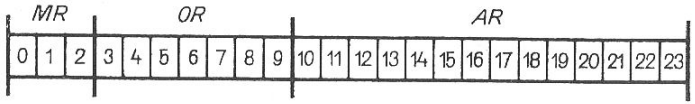
Arytmometr jest blokiem wykonującym wszystkie operacje, jakie maszyna może wykonać. Odgrywa on taką samą rolę jak arytmometr elektryczny w obliczeniach ręcznych. Jedną z ważnych części arytmometru jest tzw. **akumulator** — rejestr biorący udział w każdej operacji maszyny.

Operacje maszyny wygodnie jest podzielić na grupy:

1) **Operacje arytmetyczne.** Maszyny cyfrowe mogą wykonywać operacje arytmetyczne na dwóch dowolnych liczbach, z których jedna jest zazwyczaj umieszczona w akumulatorze, a druga w pamięci. Do operacji arytmetycznych zalicza się: dodawanie, odejmowanie, mnożenie i dzielenie. Operacje arytmetyczne dzielą się na stałoprzecinkowe i zmiennoprzecinkowe, w zależności od typu liczb biorących udział w operacji. Obie te grupy operacji mają zupełnie różne realizacje w maszynie.
2) **Operacje logiczne i sterujące.** Na podstawie takich kryteriów jak znak liczby lub porównanie dwu liczb, jako następna może być wykonana jedna z dwu dowolnych instrukcji, z których każda rozpoczyna inny wariant obliczeń. Na przykład jeśli wyróżnik równania kwadratowego jest nieujemny, wówczas przechodzi się do obliczenia pierwiastków rzeczywistych, w przeciwnym przypadku oblicza się pierwiastki zespolone. W tym przypadku badanie znaku wyróżnika będzie operacją logiczną, natomiast wybranie jednego wariantu obliczeń — operacją sterującą.
3) **Operacje przesyłania danych.** Operacje te przesyłają dane z jednego miejsca pamięci do drugiego.
4) **Operacje wejścia i wyjścia** sterują wprowadzaniem informacji do maszyny i ich wyprowadzaniem z maszyny.

Sterowanie. Rozwiązanie określonego zadania odbywa się zgodnie z uprzednio przygotowanym programem. Program ten zapisany w tzw. języku wewnętrznym maszyny musi być wpierw wprowadzony do pamięci maszyny. Program zapisany w kolejnych komórkach maszyny przedstawia sobą ostatecznie ciąg poleceń, które wykonane w określonej kolejności prowadzą do rozwiązania zadania. Każde polecenie nazywamy **rozkazem.** W maszynie cyfrowej ODRA 1204 rozkaz jest słowem krótkim. Słowo rozkazowe dzieli się na trzy części.

Trzy pierwsze pozycje binarne (0, 1, 2) stanowią **część modyfikacyjną** MR rozkazu. Dalsze 7 bitów (od 3 do 9) stanowi **część ope-**



Rys. 1.2. Budowa rozkazu adresowego w maszynie ODRA 1204

racyjną OP rozkazu. Pozostałe 14 bitów (od 10 do 23) stanowi **część adresową** AR rozkazu.

Część operacyjna rozkazu określa rodzaj wykonywanej czynności (np. odejmowanie, mnożenie, przesyłanie itp.). Część adresowa wskazuje adres liczby biorącej udział w wykonywanej czynności (jedna z liczb znajduje się zawsze w akumulatorze). Roli części modyfikacyjnej nie będziemy tu omawiali.

Zadaniem bloku sterowania jest pobieranie kolejnych rozkazów z pamięci maszyny do specjalnego rejestru, zwanego rejestrem rozkazów, rozszyfrowanie treści tego rozkazu, tzn. ustalenie wykonywanej czynności i odszukanie w pamięci liczby, której adres jest zawarty w części adresowej rozkazu, oraz uruchomienie odpowiedniej części arytmometru wykonującej tę czynność. Arytmometr po zakończeniu wykonywania operacji przesyła do bloku sterowania sygnał zakończenia operacji. Wówczas blok sterowania pobiera następny do wykonania rozkaz, z którym postępuje analogicznie jak z poprzednim. Czynności te powtarzają się aż do momentu pobrania do rejestru rozkazów rozkazu STOP, który spowoduje zatrzymanie pracy maszyny.

Rozdział 2

JĘZYK ALGOL 60

2.1. Informacje wstępne

W pierwszej części niniejszego podręcznika określone zostały pojęcia algorytmu i programu. Odróżnienie obu tych pojęć jest, w zasadzie, pewną historyczną konsekwencją z czasów, kiedy nie były jeszcze rozwinięte języki algorytmiczne, a programy obliczeń pisało się w kodzie wewnętrznym maszyny. Programowanie w kodzie wewnętrznym nie zostało całkowicie wyeliminowane i można by podać wiele przykładów, gdzie celowo używa się języka wewnętrznego. Jednakże programowanie w kodzie wewnętrznym jest bardzo żmudne, czasochłonne, wymaga dużej uwagi i systematyczności. Okazało się, że programowanie w kodzie wewnętrznym można nieco ułatwić, jeśli przedtem przygotuje się algorytm w pewnym łatwo czytelnym języku i na jego podstawie opracowuje się program obliczeń. To było powodem powstania i rozwoju języków algorytmicznych. Były to wpierw języki wykorzystujące pewne graficzne symbole (języki schematów blokowych). Okazało się, że programowanie w kodzie wewnętrznym, zwane też kodowaniem, w oparciu o dobrze opracowany algorytm wykonuje się prawie automatycznie. To nasunęło myśl przerzucenia ciężaru programowania, przynajmniej częściowo, na same maszyny cyfrowe. Było to tym bardziej konieczne, że zastosowania maszyn rosną szybciej niż możliwości kształcenia programistów. Z tego powodu rozpoczęła się rozwijać teoria języków algorytmicznych z jednej strony oraz teoria budowy translatorów, tzn. programów tłumaczących algorytmy z języka algorytmicznego na język wewnętrzny maszyny,