U N I V E R Z I T E T   U   B E O G R A D U

P R I R O D N O   M A T E M A T I Č K I   F A K U L T E T

B E O G R A D

MIRKO I. LUKIĆ

PRILOZI REŠAVANJU PROBLEMA TAČNOSTI IZRAČUNAVANJA U NUMERIČKOJ ANALIZI

(doktorska disertacija)

BEOGRAD, 1989 godine

MENTOR : dr Velimir Simonović

Mašinski fakultet Beograd

ČLANOVI KOMISIJE:

dr Nedeljko Parezanović

Prirodno matematički fakultet Beograd

dr Dušan Tošić

Prirodno matematički fakultet Beograd

dr Dušan Slavić

Elektrotehnički fakultet Beograd

Datum odbrane :

Datum promocije :

Prilozi rešavanju problema tačnosti izračunavanja u numeričkoj analizi

Apstrakt

U radu je razvijena aritmetika nad nizovima brojeva u pokretnom zarezu na osnovu Bohlender-ovog algoritma sumiranja. U tu svrhu uvedena je nova vrsta proširene preciznosti, RN-preciznost. Nizovi nad kojima je definisana aritmetika uređeni su u odnosu na relaciju koja zavisi od zaokruživanja . Aritmetika je primenjena na izračunavanje vrednosti polinoma i nalaženje proste nule polinoma.

U nizu teorema dokazanih u disertaciji analizirane su aritmetičke operacije u odnosu na širenje greške. Ovi rezultati se mogu primeniti na približno izračunavanje vrednosti aritmetičkog izraza. Za četiri osnovne aritmetičke operacije date su procene za poziciju prve pogrešne cifre rezultata u zavisnosti od prve pogrešne cifre argumenata operacije. Teoreme važe za proizvoljnu proširenu preciznost.

Dobiveni rezultati primenjeni su u nekim problemima linearne algebre. Predložen je algoritam eliminacije za izračunavanje vrednosti determinanata sistema od n linearnih jednačina sa n nepoznatih , koji je po broju operacija približan poznatoj Gauss-ovoj eliminaciji . Isti algoritam se koristi i za formiranje karakterističnog polinoma matrice.

Ključne reči : pokretni zarez, Bohlender-ov algoritam, tačnost, zaokruživanje, proširena RN-preciznost, približno izračunavanje.

Contributions to the Solving Accuracy Computation Problem in Numerical Analysis

Abstract

Ph.D. thesis developed an arithmetic over sequences of floating point numbers, based on Bohlender summation algorithm. That is why new extended precision is introduced, RN-precision.Sequence of floating point numbers , subject to the arithmetic , is ordered with respect to a relation dependent on rounding.The arithmetic is applied to the polynomial evaluation and computing simple zero of polynomial.

Error propagation of elementary arithmetic operations is investigated and several theorems are proved in the thesis. These theorems can be applied to error estimation in evaluation of arithmetic expressions. Position of the first wrong result digit depending on the first wrong operands digit is estimated for each of four elementary arithmetic operations. The theorems are valid for arbitrary extended precision.

Developed theory is applied to some linear algebra problems. It is developed elimination algorithm , variant of Gaussian algorithm, for determinant evaluation of n linear equations in n unknowns. The same algoritm can be applied to computation of characteristic matrix polynomial.

Key words : floating point , Bohlender algorithm , accuracy , rounding , extended precision , approximate computation.

Sadržaj

1.1 Uvod

Većina onih koji se bave numeričkom analizom **
ne pokazuju interes za računarsku aritmetiku.

B. PARLETT(1979)

Izračunavanja u pokretnom zarezu predstavljaju danas opšte prihvaćen način manipulisanja realnim brojevima u naučno-tehničkim istraživanjima i primenama. Međutim, način provere dobijenih rezultata često predstavlja veliki problem. Provera se najčešće sastoji u intuitivnoj proceni krajnjih rezultata što je nekada nemoguće i nepouzdano. Pa i kada neki rezultat smatramo logičnim i prihvatljivim, ne možemo da odgovorimo na pitanje koliko cifara dobijenog rezultata je tačno. Sve je ovo uočeno još i pre pojave računara ali upotrebom računara u obimnim proračunima ovaj problem postaje veoma važan.

Problem nepouzdanosti izračunavanja u pokretnom zarezu može da se traži u samim osnovnim aritmetičkim operacijama (sabiranje , oduzimanje , množenje i deljenje) , koje su približne. Otuda sledi i veoma jednostavna karakterizacija brojeva u pokretnom zarezu koju je dao SAMUEL JOHNSON(1750): " Približni brojevi su uvek pogrešni ", koju je citirao D. Knuth u svojoj drugoj knjizi, videti /31/ strana 213.

Međiutim, iako situacija sa realnim brojevima u pokretnom zarezu izgleda bez perspektive na samom početku, teško je predložiti neki drugi sistem za predstavljanje brojeva koji bi se pokazao efikasnijim.

Kada smo se jednog momenta odlučili za brojeve u pokretnom zarezu kao relativno prihvatljivim načinom predstavljanja realnih brojeva, sledeći problem je korektno definisanje osnovnih aritmetičkih operacija, što nije uopšte trivijalan zadatak. U našem izlaganju, za osnovne aritmetičke operacije korišćeni su algoritmi iz /33/.

Sledeći korak su algoritmi izračunavanja zbira. Osnovne ideje za ovaj problem su sadržane u /5/ i /15/. U /5/ G. Bohlender je dao algoritam sumiranja koji daje najbolju aproksimaciju, u smislu zaokruživanja, sume od n brojeva u pokretnom zarezu, a u /15/ T.J. Dekker razmatra problem izvođenja aritmetike dvostruke preciznosti pomoću aritmetika obične preciznosti. Bohlender-ov algoritam sumiranja bazira na lemi 2.3.2 (u tekstu koji sledi) , koja predstavlja osnovu za tačno sumiranje koje je razrađeno u delu 2.4.

** strana 178 u /31/

Kao osnova za tačno sumiranje služi relacija (def 2.4.3):

$$x \overset{\boxplus}{<} y \Longleftrightarrow x \boxplus y = y$$

za koju je dokazano da ima slična svojstva kao i relacija $<$. S druge strane, $\overset{\boxplus}{<}$ je relacija koja se na prirodan način uvodi u skup brojeva u pokretnom zarezu.
Kao rezultat uvađenih pojmova i teorema je drugi algoritam tačnog sumiranja, čiji je rezultat niz brojeva u pokretnom zarezu uređen u odnosu na i redukovan u odnosu na $\boxplus$ (dalja primena $\boxplus$ ne menja niz). Ovakav način predstavljanja rezultata daje ideju o gledanju na dobijeni niz kao na neki oblik proširene preciznosti. Nazovimo ovu preciznost RN (realan niz) preciznost. Ako ovu preciznost uporedimo sa proširenom preciznošću R.P. Brent-a u /8/ onda se uočava da RN proširena preciznost ima prednosti nad Brent-ovom. Kao prvo, RN-preciznost koristi postojeću realnu aritmetiku računara, dok se u Brent-ovoj preciznosti sve elementarne operacije moraju realizovati u celobrojnoj aritmetici.

Može se staviti prigovor da RN-preciznost neracionalno koristi memoriju računara, jer umesto eksponenta za svaki realan broj niza, možemo da uzmemo jedan ceo broj koji će da ima ulogu eksponenta za ceo niz.

Za neracionalno korišćenje memorije može da se stavi prigovor i kod Brent-ove preciznosti: zbog bržeg izvršavanja operacije množenja prenos pri sabiranju se vrši svaki osmi put, što onda umanjuje osnovu računanja. Uzimajući u obzir da za samo izvršavanje množenja brojevi moraju da imaju dva puta manju dužinu od formata I4, sledi da se za brojeve Brent-ove proširene preciznosti koristi manje od polovine kapaciteta celobrojne promenjive tipa I4.

RN-preciznost vrlo jednostavno komunicira sa realnom aritmetikom računara, dok je u Brent-ovoj preciznosti realizovan veliki broj potprograma za ovu svrhu: MPCRM, MPCMR, MPCDM, MPCMD, koje čak za neke vrednosti argumenata daju nekorektne rezultate, što je istaknuto u samom paketu.

Nedostatak RN-preciznosti je u relativno malom intervalu za vrednost eksponenta, jer koristi interval eksponenta realnih brojeve, koji je kod nekih računara relativno uzak. Iz ovog razloga uvedeno je prošireno predstavljanje realnog broja X u obliku mantisa i eksponent: (XM,XE). U ovom slučaju se koristi potpuni kapacitet celobrojne promenjive XE za čuvanje eksponenta.

U prilozima je razvijen čitav niz potprograma za realizaciju algoritama, razvijenih u ovom radu .Tako je Bohlender-ov algoritam, kao pomoćni algoritam, realizovan kao potprogram BSUMA(X,N,S), gde je X niz realnih brojeva koji treba da se sumira, N je dužina niza X a S je najbolja aproksimacija sume u smislu definisanog zaokruživanja.

Drugi tačan algoritam sumiranja, razvijen u ovom radu, je realizovan kao potprogram TSUMA(A,N,K) gde je A niz realnih brojeva koji treba da se sumira, N je dužina niza A, a K je dužina redukovanog niza-sume koji ostaje u A.
Odmah se uočava da drugi algoritam sumiranja ima neke prednosti u odnosu na Bohlender-ov algoritam.To je evidentno i pri samoj realizaciji, jer kao što se vidi u prilozima Bohlender-ov algoritam zahteva mnogo više potprograma.Takođe je razvijena i varijanta drugog algoritma sumiranja koja nosi naziv TSUMAP(AM,AE,N,K) . Poslednje slovo u nazivu ukazuje da se radi o sumiranju nad proširenim realnim brojevima. AM predstavlja niz mantisa a AE predstavlja niz eksponenata tipa I4. N i K imaju istu ulogu kao i u TSUMA.

U daljem razvoju teorije uveden je tačan algoritam množenja niza realnim brojem. Za definisanje ovog algoritma potrebne su operacije tačnog množenja dva broja u pokretnom zarezu i algoritam tačnog sumiranja. U prilozima su razvjene dve varijante algoritma tačnog množenja niza brojem . Prva varijanta nosi naziv TMNB(X,Y,N,XY,K). X je niz realnih brojeva dužine N, Y je realan broj kojim se množi niz X, XY je rezultujući niz dužine K. Druga varijanta ovog algoritma nosi naziv TMNBP(XM,XE,YM,YE,N,XYM,XYE,K) i predstavlja isti algoritam samo nad proširenim realnim brojevima.

Sledeći algoritam se odnosi na tačno izračunavanje vrednosti polinoma. Koristi algoritme tačnog množenja niza brojem i drugi algoritam sumiranja. Algoritam je realizovan kao potprogram POLI(X,A,N,B,K) gde je X argument za koji se izračunava vrednost polinoma,A je niz koeficijenata polinoma dužine N, B je niz realnih brojeva dužine K koji sadrži vrednost polinoma.

Imajući rešen problem izračunavanja vrednosti polinoma, sledeći korak je nalaženje nule na intervalu na čijim krajevima vrednosti polinoma imaju različite znake.Algoritam je realizovan kao potprogram NULA(A,N,AA,BB,A1,B1,C) gde su: A niz koeficijenata polinoma uređen u rastući niz u odnosu na odgovarajući stepen promenjive, (AA,BB) interval na kome se nalazi nula , (A1,B1) je najuži interval koji sadrži nulu a C je najbolja aproksimacija nule. Algoritam proverava da li na (AA,BB) postoji nula i tada metodom polovljenja nalazi najuži interval (A1,B1) i najbolju aproksimaciju C.

Za dalji razvoj aritmetike nad nizovima realnih brojeva u pokretnom zarezu potreban je algoritam tačnog množenja dva niza. Algoritam je realizovan kao potprogram TM2N(X,N,Y,M,XY ,K) gde su X dužine N, Y dužine M ulazni nizovi a XY je proizvod dužine K.

Operacija deljenja je poslednja od elementarnih operacija koja je neophodna za aritmetiku nad nizovima. Deljenje je realizovano na dva načina : pomoću recipročne vrednosti i direktno. Uvođenje deljenja preko recipročne vrednosti ima nedostatak kada je deljenje konačno.

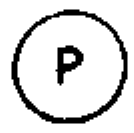
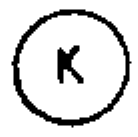
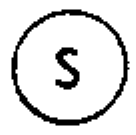
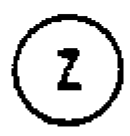
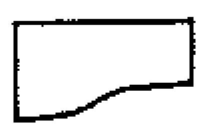
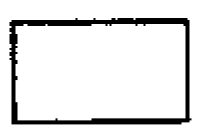
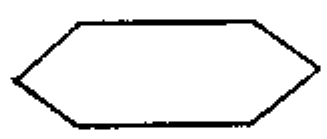
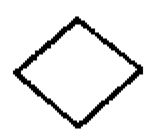
Međutim, uvođenje direktnog deljenja ukazuje na osobenost operacija u RN-preciznosti . Dok je u slučaju deljenja algoritam određivanja cifre količnika relativno složen (videti /31/ i primedbe u /46/) , kod RN-preciznosti je deljenje jednostavno i koristi deljenje obične aritmetike u pokretnom zarezu.
Svaki od algoritama je propraćen odgovarajućim teoremama koje se odnose na interval eksponenta,koji je potreban da bi se algoritam izvršio.

U trećem delu ovoga rada se razmatra problem izračunavanja vrednosti aritmetičkog izraza analizom osnovnih aritmetičkih operacija. Glavni rezultat ovoga dela može se jednostavno iskazati : ako su operandi pogrešni na nekom mestu mantise , tada greška napreduje u slučaju sabiranja i oduzimanja najviše za jednu cifru a u slučaju množenja i deljenja za najviše dve cifre,ako je osnova računanja $b > 2$.Ako je $b = 2$ tada u slučaju sabiranja i oduzimanja greška napreduje za najviše dve cifre a u slučaju množenja i deljenja najviše za tri cifre. Uvedene procene omogućavaju analizu a priori širenja greške, jer na osnovu teorema 3.1.2, 3.1.3, 3.1.4, 3.1.5, 3.1.6 i 3.1.7 možemo unapred reći koliko cifara dobijenog rezultata je pogrešno. Ove teoreme možemo smatrati fundamentalnim za analizu greške a priori .
Teorema 3.1.1 daje odgovor na pitanje o broju rezervnih cifara za slučaj sumiranja n brojeva u RN-preciznosti.

U četvrtom delu se razmatraju neke primene RN-preciznosti na probleme linearne algebre kao što su rešavanje sistema linearnih jednačina i formiranje karakterističnog polinoma matrice. Ovde su navedene dobro poznate metode modularne aritmetike za rešavanje sistema linearnih jednačina.Međutim, nedostatak ove metode predstavlja pesimistična procena broja prostih brojeva potrebnih za predstavljanje vrednosti determinante, videti /11/.
Teorema 4.3.1, koja se odnosi na rešavanje sistema linearnih jednačina pomoću determinanata je verovatno poznata, iako se do nje došlo samostalno. Ona predstavlja primenu Gauss-ove eliminacije na izračunavanje determinanata . Primena teoreme 4.3.1 uz pomoć RN-preciznosti je ilustrovana na primeru Hilbert-ove matrice 15 X 15. S druge strane, teorema 4.3.1 bi trebalo da se uvede u našu svakodnevnu univerzitetsku praksu, uz Cramer-ovo pravilo ili Gauss-ovu eliminaciju.

Na kraju se razmatra mogućnost formiranja karakterističnog polinoma matrice u RN-preciznosti. Ovakav pristup problemu nalaženja sopstvenih vrednosti se predlaže, jer su u RN-preciznosti rešana pitanja izračunavanja vrednosti polinoma i formiranja karakterističnog polinoma.

1.2 Spisak nekih oznaka

	početak algoritma
	kraj algoritma
	sastavljanje mantise i eksponenta
	normalizacija posle sabiranja
	normalizacija posle množenja
	zaokruživanje
	ulaz algoritma
	izlaz algoritma
	dodeljivanje
	pitanje
	kraj ciklusa
$\forall$	univerzalni kvantifikator
$\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$	logičke operacije
sgn	znak funkcija
$\lceil\,\rceil$	zaokruživanje na gore iz realnih brojeva u cele
$\lfloor\,\rfloor$	zaokruživanje na dole iz realnih brojeva u cele
a|b	a deli b
$\\|\ \\|$	norma matrice
$\doteq$	približno

2.1 Brojevi u pokretnom zarezu

U ovom delu se navode neka definicije i teoreme, uzete iz /33/ ,koje se odnose na realne brojeve i brojeve u pokretnom zarezu.

Teorema 2.1.1 Svaki realan broj x je na jedinstven način predstavljen u sistemu sa osnovom b razvojem oblika

$$x = *d_n\, d_{n-1} \ldots d_1\, d_0 \,, d_{-1}\, d_{-2}\, d_{-3} \ldots = * \sum_{\nu=n}^{-\infty} d_\nu b^\nu \qquad (1)$$

gde je $* \in \{+,-\}$ i $b \in N$, $b > 1$. Celi brojevi d_i, $i = n(-1)-\infty$ zadovaljavaju uslove

$$0 \leqslant d_i \leqslant b-1 \qquad \text{za } i = n(-1)-\infty \qquad (2)$$

$$d_i \leqslant b-2 \qquad \text{za beskonačno mnogo } i \qquad (3)$$

Uslovi (1)-(3) definišu jedinstven realan broj. b je osnova brojnog sistema a d_i, $i = n(-1)-\infty$ su cifre.

Množenjem odgovarajućim stepenom osnove b realan broj x, zarez možemo postaviti na željenu poziciju. Ako se zarez nalazi levo od prve cifre različite od nule , onda je to normalizovano predstavljanje broja x u osnovi b. Nula nema takvu reprezentaciju.

Sa R_b označimo sledeći skup:

$$R_b = \{0\} \cup \Big\{ x = * \; m \cdot b^{e} \mid * \in \{+,-\}, b \in N,\; b > 1,\; e \in Z,$$
$$m = \sum_{i=1}^{\infty} x[i] \cdot b^{-i} \;,$$
$$x[i] \in \{0, 1, \ldots, b-1\} \;,\; x[1] \neq 0$$
$$x[i] \leqslant b-2 \text{ za baskonačno mnogo } i \Big\}$$

gde je b osnova predstavljanja, * je znak, m je mantisa, a e je eksponent broja x.

Kako računar može da pamti samo konačan broj informacija, ceo skup R_b ne možemo predstaviti u računaru. Sledeća dva podskupa od R_b koristimo u daljem izlaganju , zavisno od ograničenja na oblast vrednosti eksponenta.

Definicija 2.1.1 Realan broj x je broj u pokretnom zarezu ako je elemenat jednog od sledećih skupova:

1. $S_{b,\ell} = \left\{ x \in R_b \mid m = \sum_{i=1}^{\ell} x[i] \cdot b^{-i} \right\}$
2. $S = S(b,l,e_1,e_2) = \left\{ x \in S_{b,\ell} \mid e_1 \leqslant e \leqslant e_2 \,,\; e_1, e_2 \in Z \right\}$

Da bi nula imala jedinstveno predstavljanje usvajamo sledeće: sgn(0) = + , mant(0) = 0,00...0 (l nula posle zareza), $\exp(0) = e_1$.

Primer 2.1.1 Da bi teorija i algoritmi u ovom radu imali realnu osnovu navodimo primer sistema u pokretnom zarezu u običnoj preciznosti DIGITAL-ove familije računara VAX-11. Prema napred navedenoj notaciji taj sistem možemo opisati sa S(2,24,-128,127). U ovom sistemu nula ima napred navedeno predstavljanje. Predstavljanje nule na ovakav način ima još jednu prednost:minimalna vrednost eksponenta(-128) omogućava da nulu razlikujemo od brojeva u pokretnom zarezu čija mantisa ima samo 1 posle zareza, a ona je sakrivena u unutrašnjem predstavljanju broja jer je uvek poznata.S druge strane , zbog sakrivene jedinice realni brojevi različiti od nule imaju opseg eksponenta $[-127\ ,\ 127]$, jer je eksponent -128 rezervisan samo za nulu.

Navodimo još neke osobine skupova $S_{b,\ell}$ i $S(b,l,e_1,e_2)$. Za oba sistema važi:

$$b^{-1} \leqslant |m| < 1.$$

Elementi oba skupa su racionalni brojevi. Skup $S(b,l,e_1,e_2)$ je konačan i svaki elemenat je jedinstven. Najveći broj predstavljiv u sistemu $S(b,l,e_1,e_2)$ je

$$B = +0,(b-1)(b-1)\ldots(b-1)\cdot b^{e_2}$$

a najmanji -B. Najmanji brojevi po apsolutnoj vrednost su

$$-0,100\ldots0\cdot b^{e_1} \quad \text{i} \quad +0,100\ldots0\cdot b^{e_1} \quad .$$

Primer 2.1.2 U sistemu iz primera 2.1.1 važi

$$2^{-1} \leqslant m \leqslant 1 - 2^{-24},$$

$$B = (1 - 2^{-24})\cdot 2^{127} .$$

Najmanji brojevi po apsolutnoj vrednosti su

$$-2^{-1}\cdot 2^{-127} \quad \text{i} \quad +2^{-1}\cdot 2^{-127} .$$

U daljem izlaganju će se koristiti sledeće oznake

$$R_b^* = R_b \cup \{-\infty\} \cup \{+\infty\}$$

$$S_{b,\ell}^* = S_{b,\ell} \cup \{-\infty\} \cup \{+\infty\}$$

$$S^* = S(b,l,e_1,e_2) \cup \{-\infty\} \cup \{+\infty\}$$

$$\overline{S} = [-B,+B]$$

Definicija 2.1.2 Funkcija $\square : R^* \longrightarrow S^*$ se naziva monotono zaokruživanje iz R^* u S^* ako je

$$(\forall x \in S)\ \square x = x.$$

$$(\forall x, y \in R)(x \leqslant y \Rightarrow \square x \leqslant \square y)$$

U daljem izlaganju ćemo koristiti sledeća zaokruživanja

$(\forall x \in R)\ \nabla x = \max \{ y \in S^* \mid y \leqslant x \}$ zaokruživanje na dole

$(\forall x \in R)\ \Delta x = \min \{ y \in S^* \mid x \leqslant y \}$ zaokruživanje na gore

$(\forall x \geqslant 0)\ \square_b x = \nabla x \wedge (\forall x < 0)\ \square_b x = -\square_b(-x)$ zaokruživanje prema nuli

$(\forall x \geqslant 0)\ \square_0 x = \Delta x \wedge (\forall x < 0)\ \square_0 x = -\square_0(-x)$ zaokruživanje od nule ili zaokruživanje prema beskonačnosti

Uvodeći oznaku

$(\forall x \geqslant 0)\ S_\mu(x) = \nabla x + (\Delta x - \nabla x)\mu/b,\ \mu = 1(1)b-1$

definišimo sledeće zaokruživanje

$$\square_\mu : R^* \rightarrow S^* \quad , \mu = 1(1)b-1$$

$$(\forall x)(x \in [0, b^{e_1-1})\ \square_\mu x = 0$$

$$(\forall x)(x \geqslant b^{e_1-1})\ \square_\mu x = \begin{cases} \nabla x & x \in [\nabla x, S_\mu(x)) \\ \Delta x & x \in [S_\mu(x), \Delta x] \end{cases}$$

$$(\forall x < 0)\ \square_\mu x = -\square_\mu(-x)$$

Ako je b parno tada $\bigcirc = \square_{b/2}$ predstavlja zaokruživanje ka najbližem broju u pokretnom zarezu.

Definicija 2.1.3 Neka $x \in R_b^*$ i neka je $\square$ zaokruživanje. $\square x$ je najbolja aproksimacija x u odnosu na $\square$.

U numeričkoj analizi za analizu greške je uobičajeno da zaokruživanje zadovoljava uslov

$$(\forall x \in R)\ \square x = x(1-\varepsilon) \text{ gde je } |\varepsilon| < \varepsilon^* = \text{const}$$

U sledećoj teoremi koju navodimo bez dokaza, navodi se da monotono zaokruživanje ima prethodnu osobinu.

Teorema 2.1.2 Neka je $S^* = S^*(b, l, e_1, e_2)$ sistem sa pokretnim zarezom i $\square : R^* \rightarrow S^*$ monotono zaokruživanje. Neka je $\delta(x) = x - \square x$ greška zaokruživanja i neka su $\varepsilon_1 = \frac{\delta(x)}{x}$ i $\varepsilon_2 = \frac{\delta(x)}{\square x}$ relativne greške zaokruživanja. Tada je

$$(\forall x \in R)(b^{e_1 - 1} \leqslant |x| \leqslant B \Rightarrow \square x = x(1 - \varepsilon_1) \quad \text{gde je } |\varepsilon_1| < \varepsilon^*$$
$$\wedge\, x = \square x(1 - \varepsilon_2) \text{ gde je } |\varepsilon_2| < \varepsilon^*$$
$$\wedge\, |x - \square x| < \varepsilon^* |x| \,\wedge\, |x - \square x| < \varepsilon^* |\square x|).$$

ε^* se naziva jedinica zaokruživanja i važi

$$\varepsilon^* = \begin{cases} \frac{1}{2} b^{1-l} & \text{za } \square = \bigcirc \\ b^{1-l} & \text{za } \square \neq \bigcirc \end{cases}$$

Uvedimo još pojmove prekoračenja odozdo i odozgo.

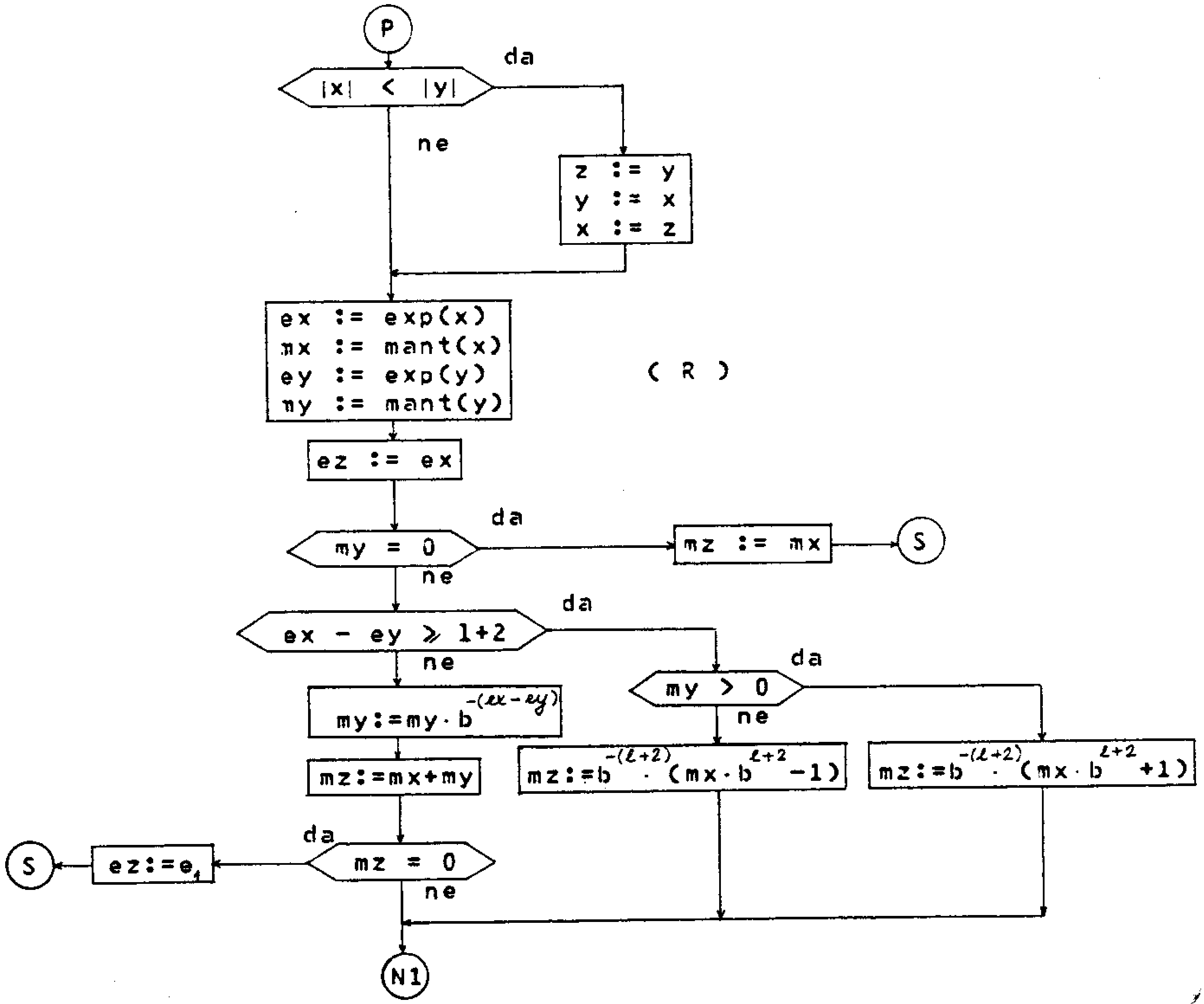
Definicija 2.1.4 Ako je $|x| \in (0, b^{e_1 - 1})$ tada govorimo o prekoračenju eksponenta odozdo. Ako je $|x| > B$, gde je B maksimalan predstavljiv broj tada govorimo o prekoračenju odozgo.

2.2 Operacije u pokretnom zarezu

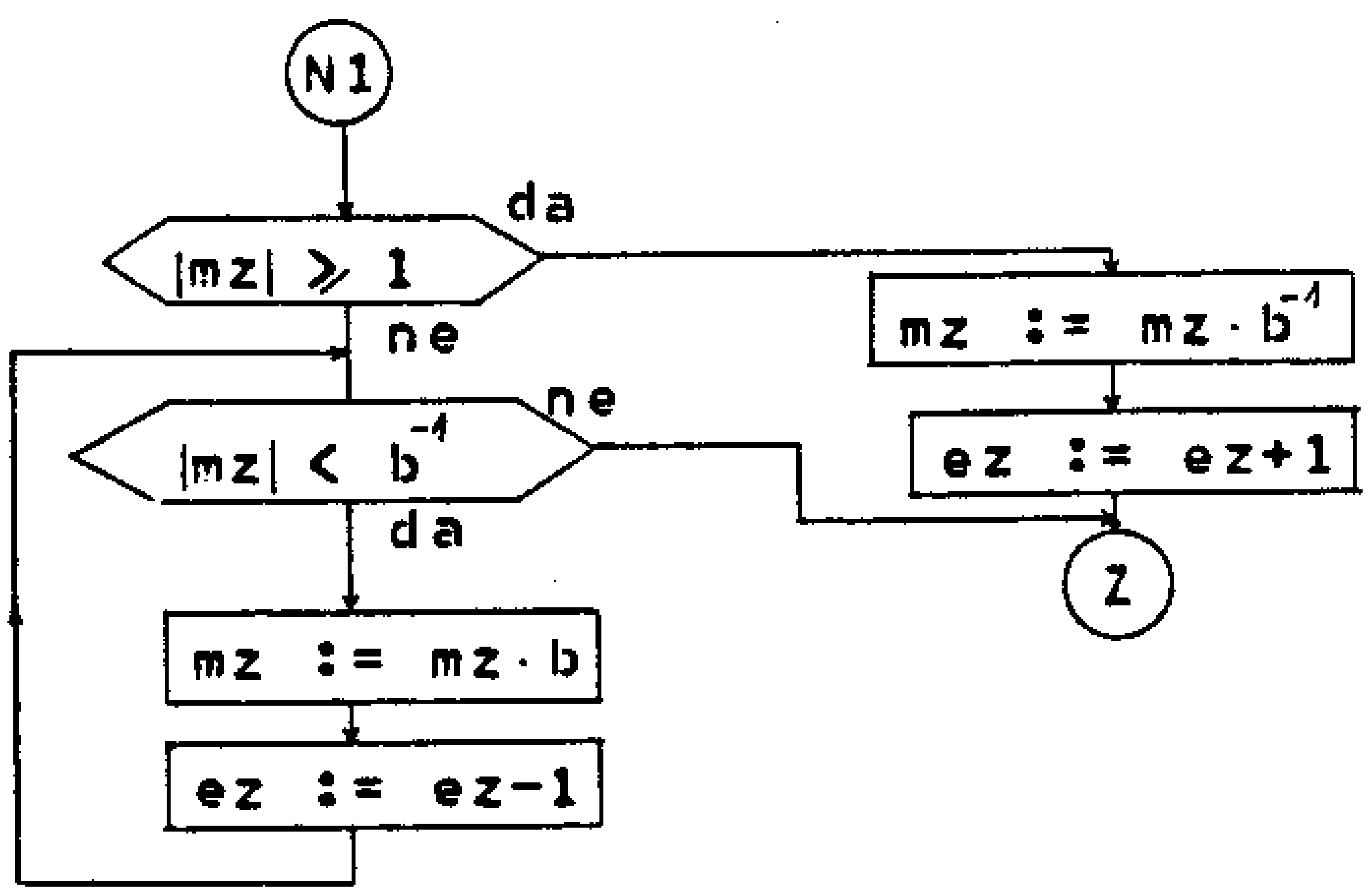
Za korektno definisanje algoritama u numeričkoj analizi neophodno je precizno definisati aritmetičke operacije u pokretnom zarezu. Iako je u /31/ to pitanje veoma detaljno razmatrano, ovo izlaganje će se uglavnom oslanjati na /33/. U /33/ su razmatrane operacije sa zaokruživanjima ∇, Δ, $\square_\mu$ $\mu = 0(1)b$, dok je u /31/ razmatrano samo zaokruživanje ka najbližem broju u pokretnom zarezu.

Četiri osnovne aritmetičke operacije u pokretnom zarezu sabiranje, oduzimanje, množenje i deljenje mogu biti realizovane koristeći akumulator dužine 2l+1 cifru u osnovi b sa jednom binarnom cifrom za znak, gde je l dužina mantise. U ovom slučaju kažemo da su algoritmi operacija relizovani pomoću dugog akumulatora. Za razliku od ove mogućnosti, pomenute algoritme možemo realizovati pomoću kratkog akumulatora, koji osim binarne cifre za znak sadrži l+2 cifre u osnovi b i još jednu binarnu cifru. Ovde ne ističemo eksplicitno binarne cifre za prekoračenje odozdo ili odozgo, ali će se ove mogućnosti takođe razmatrati u algoritmima.

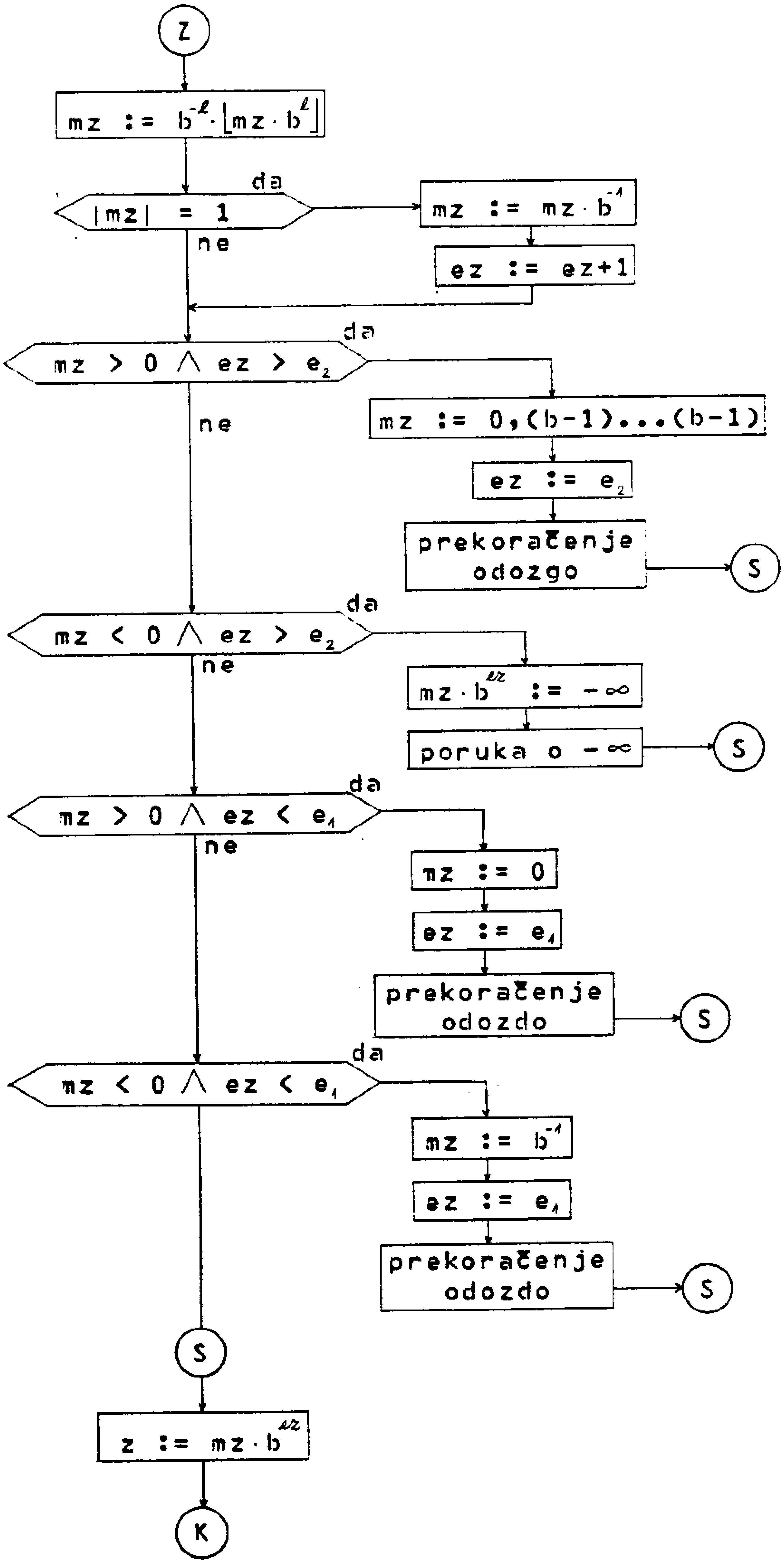
Pretpostavljamo da su operacije sabiranja , oduzimanja , množenja i deljenja u celobrojnoj aritmetici realizovane. Slede algoritmi sledećih aritmetičkih operacija u pokretnom zarezu : sabiranja (oduzimanja) , množenja i deljenja. U algoritmima ćemo koristiti sledeće oznake : R razdvajanje mantise i eksponenta broja u pokretnom zarezu , S sastavljanje mantise i eksponenta u broj u pokretnom zarezu (ista slova se koriste i za označavanje promenljivih pri definisanju algoritama ali će uloga simbola biti jasna na osnovu konteksta), N1, N2 normalizacija, Z zaokruživanje. P označava početak algoritma a K kraj algoritma. $\lfloor x \rfloor$ označava najveći ceo broj manji ili jednak x. $\lceil x \rceil$ označava najmanji ceo broj veći ili jednak x. exp(x) označava eksponent, a mant(x) označava mantisu x.



Algoritam sabiranja



Algoritam normalizacije posle sabiranja



Algoritam zaokruživanja ▽

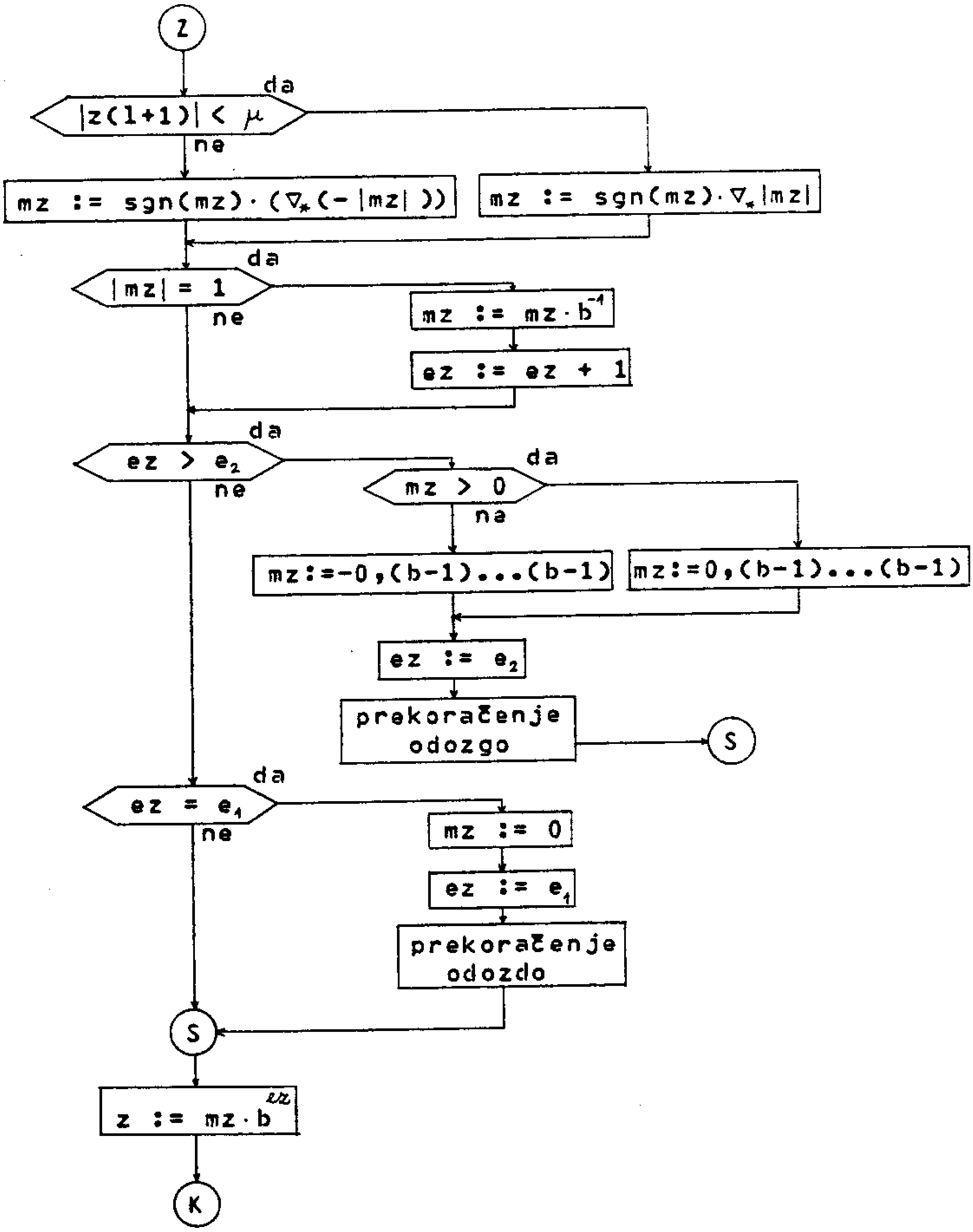
Δ možemo definisati na osnovu ∇ prema definiciji

$$(\forall z \in R)\, \Delta z = -\nabla(-z)$$

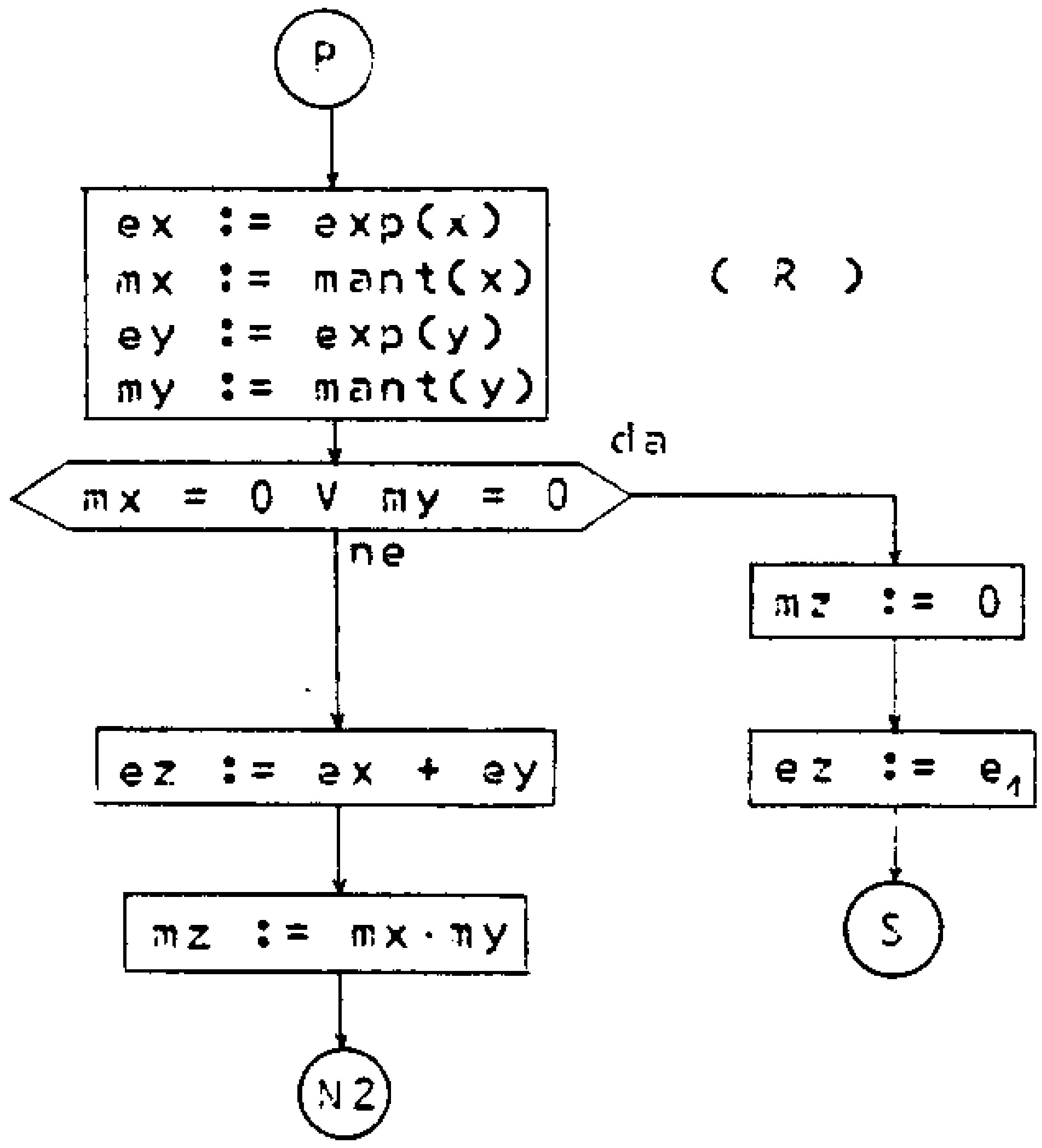
Da bismo definisali $\Box_\mu$, $0 \leqslant \mu \leqslant b$ uvedimo oznaku

$$\nabla_* mz = b^{-\ell} \cdot \lfloor mz \cdot b^{\ell} \rfloor .$$

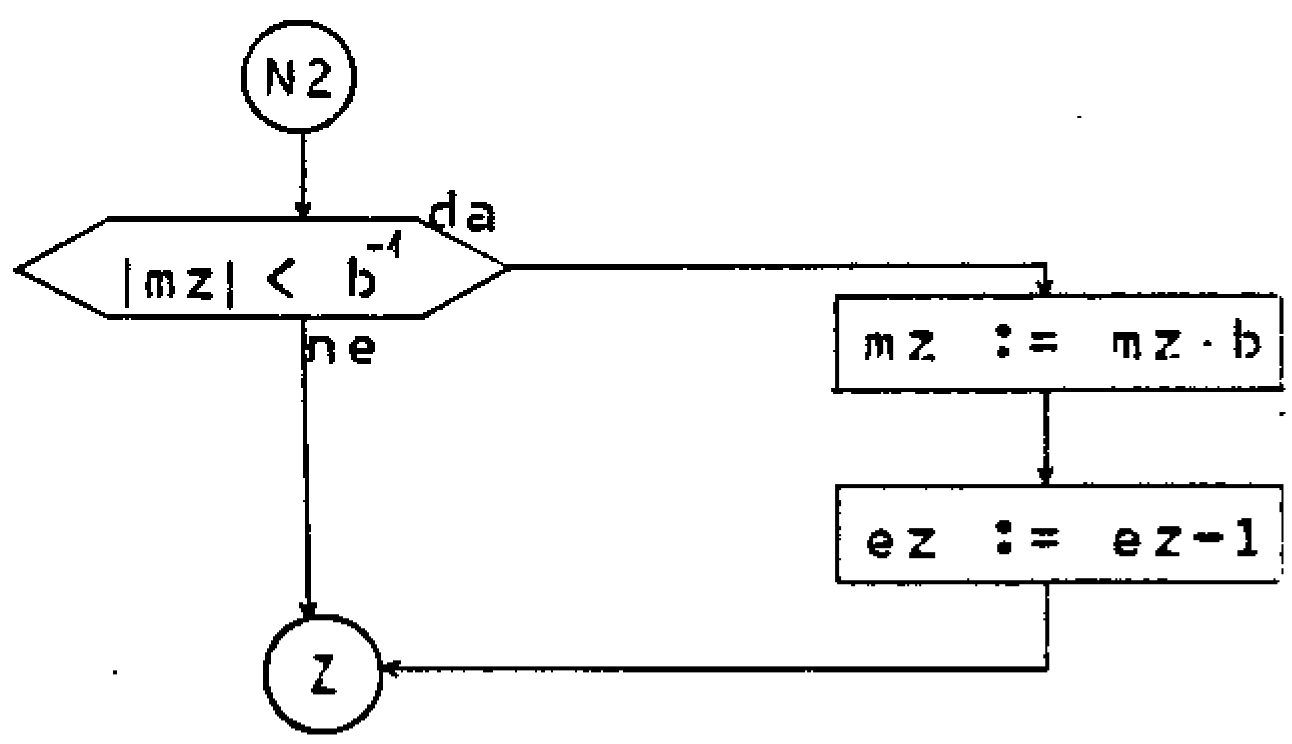
U sledećem algoritmu z(l+1) označava l+1 cifru akumulatora.



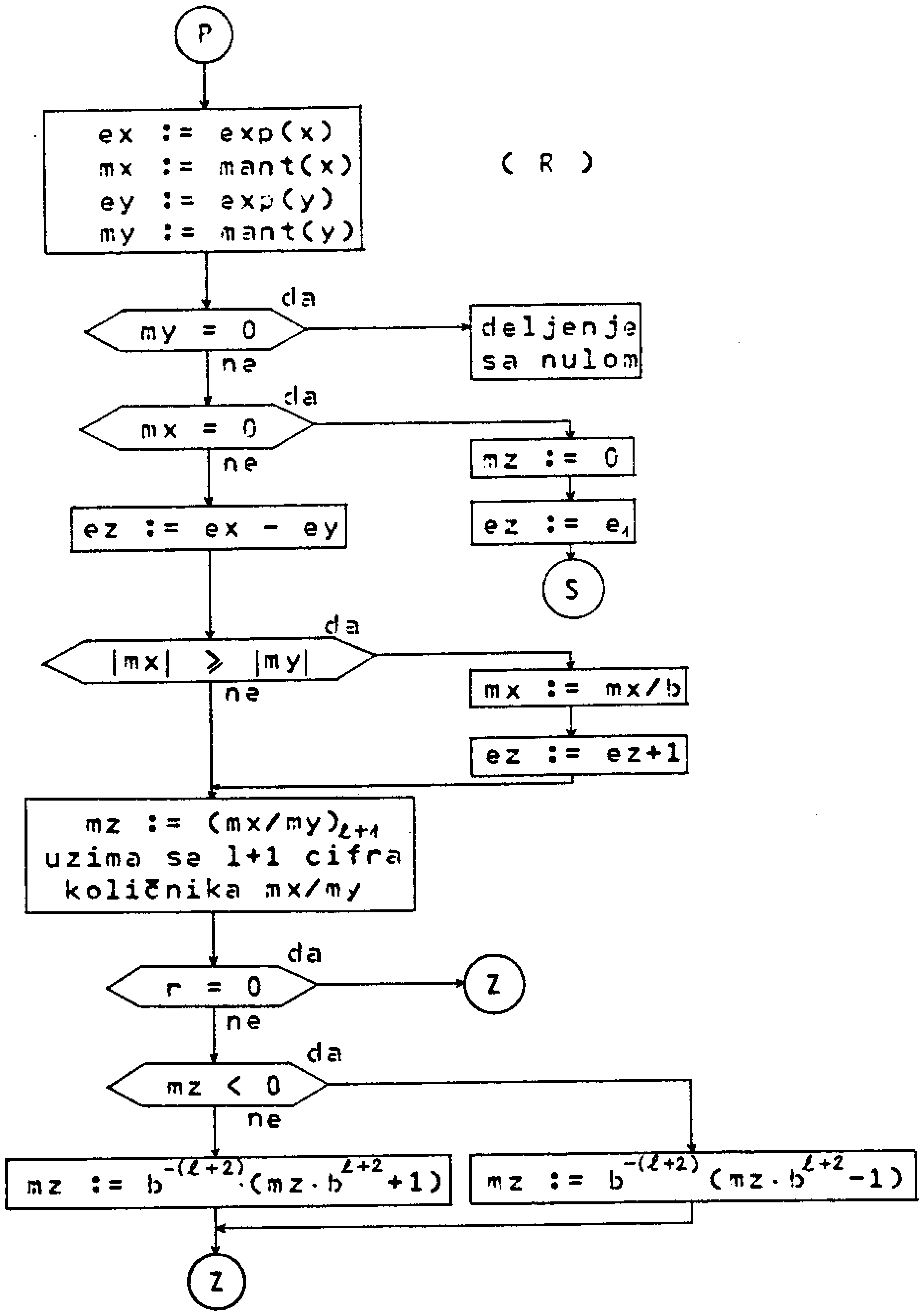
Algoritam zaokruživanja $\Box_\mu$, $0 \leqslant \mu \leqslant b$



Algoritam množenja



Algoritam normalizacije posle množenja



Algoritam deljenja

2.3 Bohlender-ov algoritam sumiranja

Izračunavanje sume n brojeva u pokretnom zarezu može dati proizvoljno veliku grešku . Zbog toga je ovaj problem privukao pažnju mnogih istraživača u oblasti računarske aritmetike i programiranja. Jedan pristup poboljšavanju tačnosti izračunavanja sume je dvostruka preciznost, ali priroda problema ostaje ista i u dvostrukoj preciznosti. Ovde navodimo algoritam Bonlender-a koji se veoma često navodi u literaturi koja se bavi ovom problematikom. Sledimo /33/ a ne originalan izvor /5/ iako oba izvora sadrže greške. Greška u /33/ je tnew := j , a trebalo bi da stoji tnew = j+1, što se zaključuje na osnovu uloge promenjive t. Greška u /5/ je u uslovu $S = 0 \vee j = 0$ a treba da stoji $S = 0$. Uvedimo sada relaciju $\prec$, koja ima važnu ulogu u pomenutom algoritmu.

Definicija 2.3.1 Neka su x i y brojevi u pokretnom zarezu, tada je

$$x \prec y \Longleftrightarrow x = 0 \vee y = 0 \vee (ex \leqslant ey \wedge y \in S_{b,ex-ey}).$$

Neformalno , $x \prec y$ u $S_b^* \setminus \{0\}$ ako i samo ako sve cifre od x imaju manje eksponente nago eksponenti cifara koje su različite od nule u y.

Za Bohlender-ov sumacioni algoritam važnu ulogu ima operacija tačno sabiranje dva broja u pokretnom zarezu x i y. Rezultat ove operacije su $s := x \oplus y$ i $r := x + y - (x \oplus y)$, tj. najbolja aproksimacija sume x i y i razlika između tačne sume i najbolje aproksimacije sume.

Sledeću lemu koja daje vezu među $\prec$, r, s navodimo bez dokaza.

Lema 2.3.1 Neka je $S = S_{b,l}$ sistem u pokretnom zarezu sa parnom osnovom i $O: R \to S$ zaokruživanje ka najbližem broju u pokretnom zarezu. Tada za prizvoljne $x, y \in S$ važi

(a) $s := x \oplus y \in S \wedge r := x + y - s \in S$
(b) $r \prec s,\ r \neq 0 \Longrightarrow es - er \geqslant 1$
(c) $(\forall z \in S)(z \prec x \wedge z \prec y \Rightarrow z \prec r \wedge z \prec s)$
(d) $(\forall z \in S)(x \prec z \vee y \prec z \Rightarrow r \prec z)$

gde je $x \oplus y = O(x + y)$.

Lema 2.3.2 Neka je $S = S_{b,l}$ i neka je $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)})$ $\in S$ n-torka brojeva u pokretnom zarezu . Polazeći od $x^{(0)}$ definišimo niz $\{x^{(k)}\}_{k=0,1,2,}$ gde je $x = (x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}) \in S^n$, na sledeći način:

$$
\begin{array}{llll}
S_1^{(k)} := x_1^{(k-1)} & & & \\
S_2^{(k)} := S_1^{(k)} \oplus x_2^{(k-1)} & ; & x_1^{(k)} := (S_1^{(k)} + x_2^{(k-1)}) - S_2^{(k)} & ; \\
S_3^{(k)} := S_2^{(k)} \oplus x_3^{(k-1)} & ; & x_2^{(k)} := (S_2^{(k)} + x_3^{(k-1)}) - S_3^{(k)} & ; \\
\vdots & & \vdots & \\
S_{n-2}^{(k)} := S_{n-3}^{(k)} \oplus x_{n-2}^{(k-1)} & ; & x_{n-3}^{(k)} := (S_{n-3}^{(k)} + x_{n-2}^{(k-1)}) - S_{n-2}^{(k)} & ; \\
S_{n-1}^{(k)} := S_{n-2}^{(k)} \oplus x_{n-1}^{(k-1)} & ; & x_{n-2}^{(k)} := (S_{n-2}^{(k)} + x_{n-1}^{(k-1)}) - S_{n-1}^{(k)} & ; \\
S_{n}^{(k)} := S_{n-1}^{(k)} \oplus x_{n}^{(k-1)} & ; & x_{n-1}^{(k)} := (S_{n-1}^{(k)} + x_{n}^{(k-1)}) - S_{n}^{(k)} & ; \\
 & & x_n^{(k)} := S_n^{(k)} ; &
\end{array}
$$

za $k = 1, 2, 3, \ldots$. Tada niz $\left\{x^{(k)}\right\}_{k=1,2,3,\ldots}$ ima sledeće osobine:

(a) $(\forall\ k \in N)\ \sum_{i=1}^{n} x_i^{(k)} = \sum_{i=1}^{n} x_i^{(0)}$

(b) $k = 1 :\quad x_{n-1}^{(1)} < x_n^{(1)}$

$k = 2 :\quad x_{n-2}^{(2)} < x_{n-1}^{(2)} < x_n^{(2)}$

$k = 3 :\quad x_{n-3}^{(3)} < x_{n-2}^{(3)} < x_{n-1}^{(3)} < x_n^{(3)}$

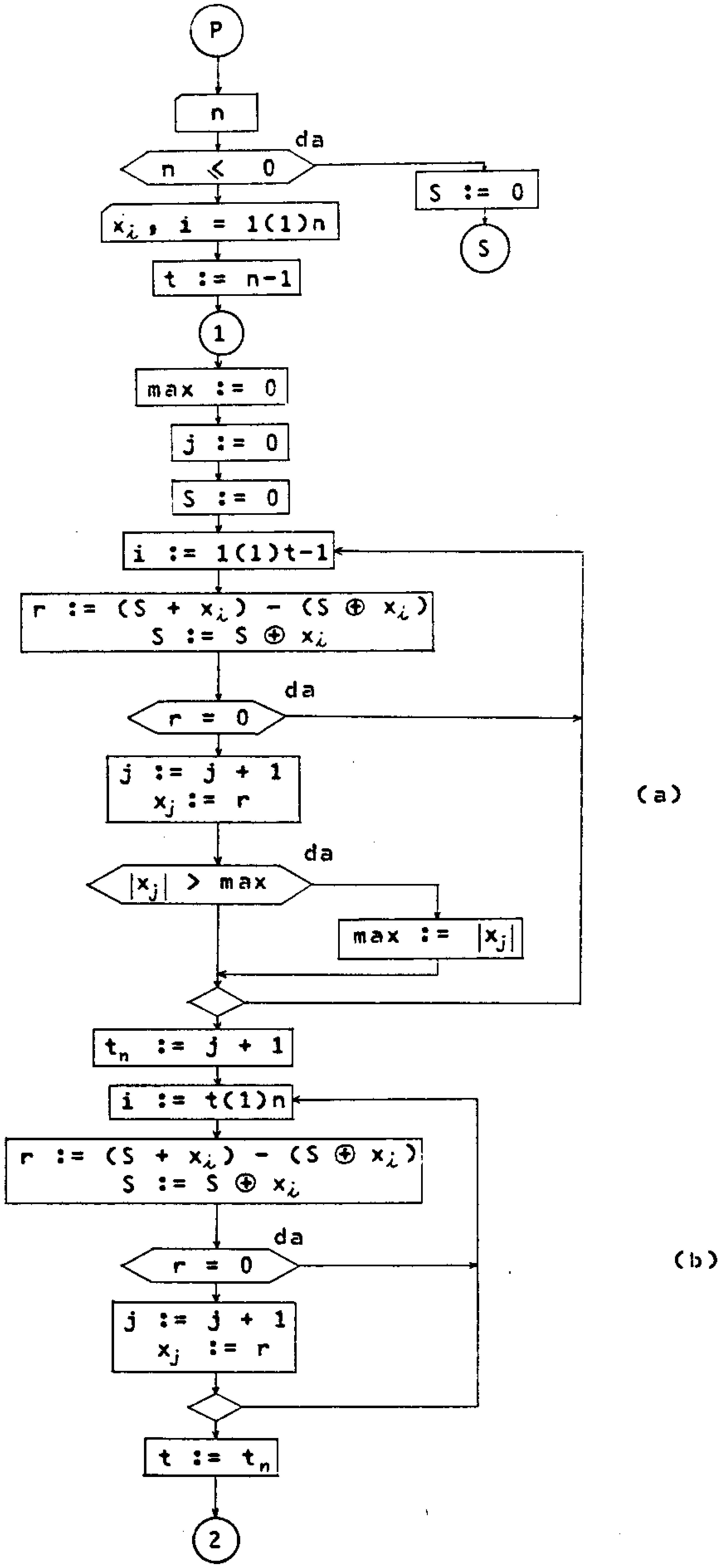
$\vdots$

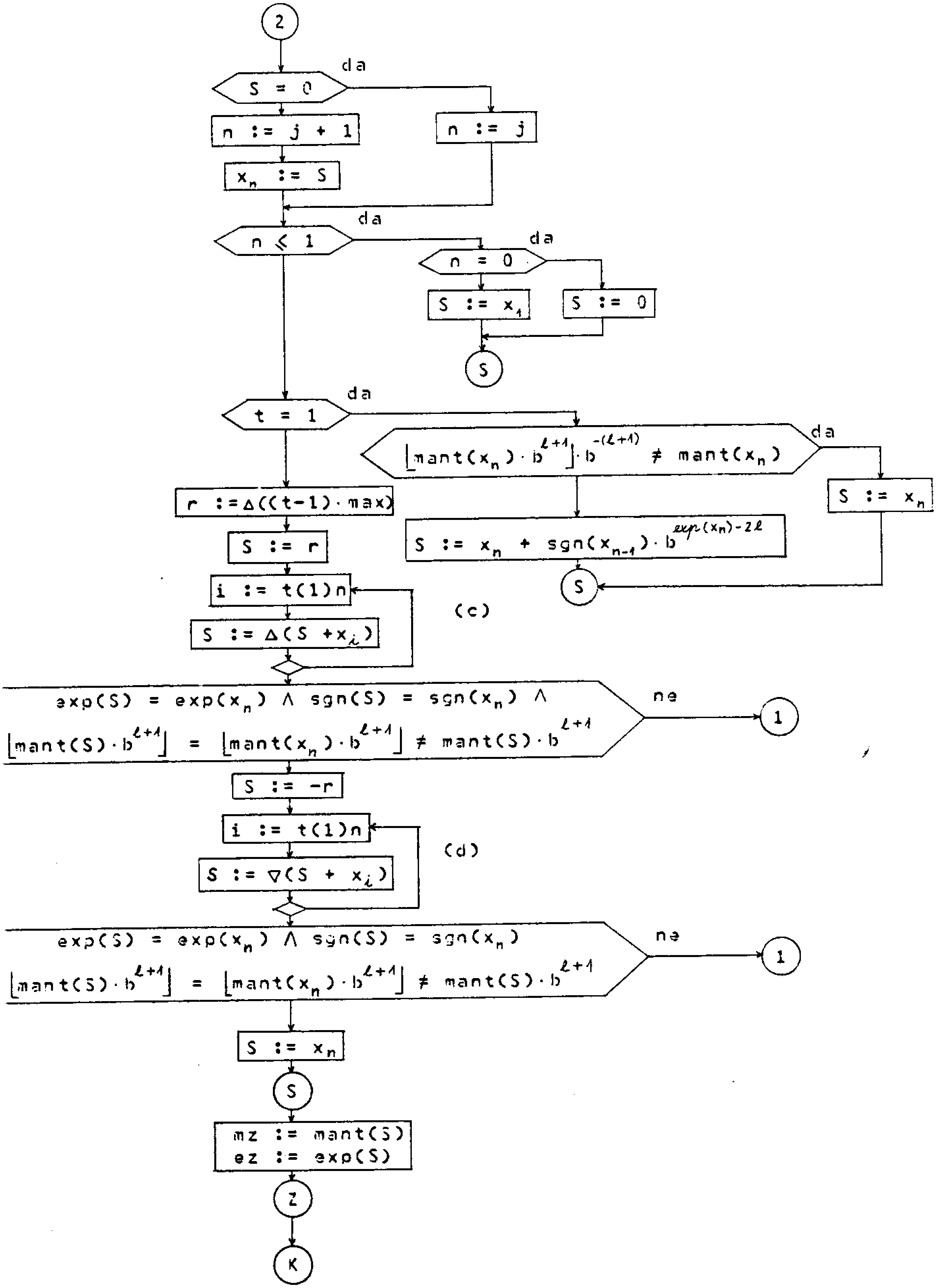
$k = n-1 :\quad x_1^{(n-1)} < x_2^{(n-1)} < x_3^{(n-1)} \ldots < x_{n-1}^{(n-1)} < x_n^{(n-1)}$.

Lemu navodimo bez dokaza. Dokaz se nalazi u /33/ i /5/.

Na osnovu prethodne lema zaključujemo da se povećavanjem k $x_i^{(k)}$ postaje uređen u rastući niz u odnosu na relaciju <. To se koristi u Bohlender-ovom algoritmu sumiranja u sistemu $S_{b,2\ell}$, odnosno sumiranju niza dvostruke dužine.

Dajemo objašnjenja koja se odnose na Bohlender-ov algoritam. Promenljiva t predstavlja n - k koje se navodi u lemi 2.3.2. Ciklus (a) određuje sumu elemenata x_i koji nisu uređeni u odnosu na < i određuje maksimalni elemenat po apsolutnoj vrednosti. Ciklus (b) nastavlja sumiranje x_i koji su već uređeni. r := Δ((t-1)·max) odredjuja gornju granicu od t-1 sabiraka x_i , koji još nisu uređeni , a max je maksimalni. Ciklus (c) određuje gornju granicu sume a ciklus (d) donju granicu sume.


P
n
n ≤ 0
da
S := 0
S
x_i , i = 1(1)n
t := n-1
1
max := 0
j := 0
S := 0
i := 1(1)t-1
r := (S + x_i) - (S ⊕ x_i)
S := S ⊕ x_i
r = 0
da
j := j + 1
x_j := r
|x_j| > max
da
max := |x_j|
(a)
t_n := j + 1
i := t(1)n
r := (S + x_i) - (S ⊕ x_i)
S := S ⊕ x_i
r = 0
da
j := j + 1
x_j := r
(b)
t := t_n
2



Bohlender-ov algoritam

Takođe treba uočiti da algoritam sumira niz čiji elementi pripadaju $S_{b,2l}$ a da je vrednost sume data u $S_{b,l}$. Računanje se izvodi u $S_{b,2l}$ radi izlaznih kriterijuma posle ciklusa (c) i (d), jer ti kriterijumi se ne bi realizovali sa mantisom dužine l.

Teorema 2.3.1 Neka je $S = S_{b,l}$ sistem u pokretnom zarezu i $l \geqslant 3$ i $x_i \in S_{b,2l}$, $i = 1, 2, \dots, n$, n brojeva u pokretnom zarezu. Neka su $\nabla, \Delta, \square_\mu, \mu = 0(1)b : R^* \to S^*$ zaokruživanja u skup brojeva čija je mantisa dužine l. Tada Bohlender-ov algoritam izračunava aproksimaciju

$$S = \widetilde{\sum_{i=1}^{n}} x_i \in S_{b,2l}$$

sume $\sum_{i=1}^{n} x_i$ sa osobinom

$$\forall (x_i) \in S^n_{b,2l} \quad \forall \square \in \left\{\nabla, \Delta, \square_\mu, \mu = 0(1)b\right\} \square \sum_{i=1}^{n} x_i = \square S.$$

Dokaz. Na osnovu leme 2.3.2 zaključujemo da je niz uređen u rastućem poretku u odnosu na relaciju <. Dakle, algoritam se zaustavlja posle najviše n-1 iteracija sa t = 1.

Da bismo dokazali tvrđenje teoreme moramo da dokažemo da se S i $\sum x_i$ poklapaju po znaku, eksponentu i prvih l+1 cifru mantise i da su l-1 cifra od S nule ako i samo ako je $S = \sum x_i$, razlikovaćemo sledeće slučajeve zaustavljanja Bohlender-ovog algoritma:

1 : $n \leqslant 1 \implies S = \sum x_i$ pa tvrđenje važi.

2 : Neka je n > 1. Ako se algoritam zaustavlja zbog t = 1 tada su x_i prema lemi 2.3.2 uređeni u odnosu na < :

$$(\forall x_i \neq 0) \quad x_1 < x_2 < \dots < x_n$$

Tada je prema lemi 2.3.1

$$\exp(x_n) - \exp(x_{n-1}) \geqslant 2l.$$

Dakle, x_n je S ako preostalih l-1 cifara mantise x_n nisu nule. U tom slučaju se dodaje ili oduzima jedinica na 2l-tom mestu x_n da bi se uzeo u obzir uticaj x_{n-1}. Kako je $l \geqslant 3$, tada je barem jedna od l-1 cifre različita od nule.

3 : Ako je $n > 1 \wedge t > 1$ tada se algoritam zaustavlja jer $x_1, x_2, \dots, x_{n-1}$ nemaju uticaja na x_n, jer ne mogu da utiču na znak, eksponent ili l+1-vu cifru mantise x_n, pa ako ne menjaju l-1 cifru x_n u nule tada je rezultat S.

2.4 Tačno sumiranje

Dosadašnje izlaganje predstavlja uvod u problem sumiranja. Na osnovu leme 2.3.2 zaključujemo da niz uređen u odnosu na $<$ sadrži tačnu vrednost sume. Prirodno je da niz $x_1, x_2, \ldots, x_n$ bude uređen opadajuće, tako da se značajniji deo vrednosti sume nalazi na početku niza. Pored toga Bohlender-ov algoritam sumira brojeve u $S_{b,2\ell}$ a vrednost sume je u $S_{b,\ell}$. Naravno, prirodno je zahtevati da ako su članovi niza iz $S_{b,\ell}$ da i vrednost najbolje aproksimacije sume bude takođe u $S_{b,\ell}$. Navedimo sada lemu 2.4.1 koja je osnova za algoritam tačnog sumiranja.

Lema 2.4.1 Neka je dat $S = S_{b,\ell}$ i neka je $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}) \in S^n$. Polazeći od $x^{(0)}$ definišimo niz $\{x^{(k)}\}_{k=0,1,2,\ldots}$, gde je $x^{(k)} = (x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}) \in S^n$, na sledeći način:

$$
\begin{array}{lll}
 & & R_1^{(k)} = x_1^{(k-1)} \\
x_1^{(k)} = R_1^{(k)} \oplus x_2^{(k-1)} & & R_2^{(k)} = (R_1^{(k)} + x_2^{(k-1)}) - x_1^{(k)} \\
x_2^{(k)} = R_2^{(k)} \oplus x_3^{(k-1)} & & R_3^{(k)} = (R_2^{(k)} + x_3^{(k-1)}) - x_2^{(k)} \\
x_3^{(k)} = R_3^{(k)} \oplus x_4^{(k-1)} & & R_4^{(k)} = (R_3^{(k)} + x_4^{(k-1)}) - x_3^{(k)} \\
\cdots\cdots & & \cdots\cdots \\
x_{n-2}^{(k)} = R_{n-2}^{(k)} \oplus x_{n-1}^{(k-1)} & & R_{n-1}^{(k)} = (R_{n-2}^{(k)} + x_{n-1}^{(k-1)}) - x_{n-2}^{(k)} \\
x_{n-1}^{(k)} = R_{n-1}^{(k)} \oplus x_n^{(k-1)} & & R_n^{(k)} = (R_{n-1}^{(k)} + x_n^{(k-1)}) - x_{n-1}^{(k)} \\
x_n^{(k)} = R_n^{(k)} & &
\end{array}
$$

za $k = 1, 2, 3, \ldots$. Tada niz $\{x^{(k)}\}_{k=0,1,2,\ldots}$ ima sledeće osobine:

(a) $(\forall k \in N) \quad \sum_{i=1}^{n} x_i^{(k)} = \sum_{i=1}^{n} x_i^{(0)}$

(b) $(\forall k \in N \wedge 1 \leqslant k \leqslant n-1) \quad x_n^{(k)} < x_{n-1}^{(k)} < \ldots < x_{n-k}^{(k)}$.

Dokaz. Osobina (a) je neposredna posledica konstrukcije niza $\{x^{(k)}\}$. Na osnovu leme 2.3.1 (b) zaključujemo da je

$$x_n^{(k)} < x_{n-1}^{(k)}$$

jer je $R_n^{(k)} < x_{n-1}^{(k)}$. Takođe važi da je $R_p^{(k)} < x_{p-1}^{(k)}$ za $p = 2, \ldots, n$. Na osnovu $x_n^{(k)} < x_{n-1}^{(k)}$ važi $x_n^{(k-1)} < x_{n-1}^{(k-1)}$. Zbog $x_n^{(k-1)} < R_{n-2}^{(k)}$ i (c) leme 2.3.1 zaključujemo da je $x_n^{(k-1)} < x_{n-2}^{(k)}$. Kako je $R_{n-1}^{(k)} < x_{n-2}^{(k)}$ i $x_n^{(k-1)} < x_{n-2}^{(k)}$ a na osnovu $x_{n-1}^{(k)} = R_{n-1}^{(k)} \oplus x_n^{(k-1)}$ i osobine (c) leme 2.3.1 zaključujemo da je $x_{n-1}^{(k)} < x_{n-2}^{(k)}$ a time je

$$x_n^{(k)} < x_{n-1}^{(k)} < x_{n-2}^{(k)} \ .$$

Analogno dokazujemo da je

$$x_n^{(k)} < x_{n-1}^{(k)} < x_{n-2}^{(k)} < x_{n-3}^{(k)}$$

i uopšte

$$x_n^{(k)} < x_{n-1}^{(k)} < \dots < x_{n-k}^{(k)}$$

za $k = 1(1)n-1$.

Definicija 2.4.1 Neka je $S = S_{b,\ell}$ sistem u pokretnom zarezu i neka $x, y \in S$. Za brojeve x i y kažemo da se ne preklapaju ako je

$$x < y \quad \text{ili} \quad y < x.$$

Definicija 2.4.2 Niz realnih brojeva u pokretnom zarezu

$$x_1, x_2, \dots, x_n$$

je normalizovan ako je

a) $x_i \neq 0$, $i = 1(1)n$,

b) x_i, x_{i+1} , $i = 1(1)n-1$ se ne preklapaju,

c) $\exp(x_n) < \exp(x_{n-1}) < \dots < \exp(x_1)$.

Teorema 2.4.1 Neka je $S = S_{b,\ell}$ sistem u pokretnom zarezu i neka $x_i \in S_{b,\ell}$, $i = 1, 2, \dots, n$ niz brojeva u pokretnom zarezu. Neka su $\nabla, \Delta, \square_\mu, \mu = 0(1)b : R^* \rightarrow S^*$ zaokruživanja. Tada prvi algoritam tačnog sumiranja daje tačnu vrednost sume u obliku niza

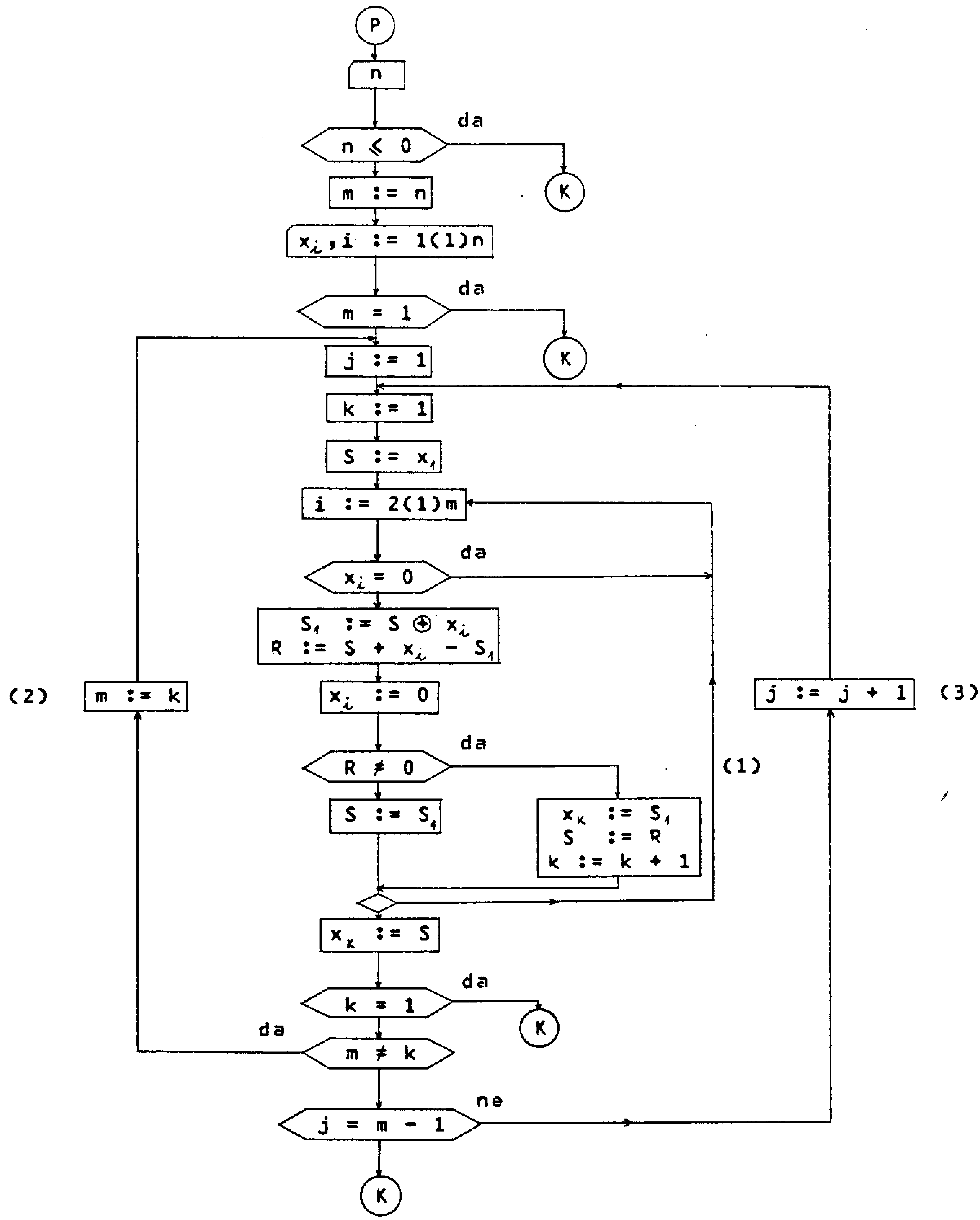
$$x_1, x_2, \dots, x_k, \ k \leqslant n$$

i dati niz je normalizovan.

Dokaz. Prema lemi 2.4.1 posle j izvršavanja ciklusa (1) prvog algoritma sumiranja važi

$$x_n^{(j)} < x_{n-1}^{(j)} < \dots < x_{n-j}^{(j)} \ .$$

Razlika je samo što u lemi 2.4.1 nisu izostavljani oni članovi niza koji su jednaki nuli, a to je u algoritmu uzeto u obzir da bi se izbegle nepotrebne operacije.



Prvi algoritam tačnog sumiranja

Izuzimajući trivijalne razloge zaustavljanja $n < 0$ i $n = 1$ prvog algoritma tačnog sumiranja, pomenuti algoritam se zaustavlja u slučajevima $k = 1$ i $j = m - 1$. m predstavlja trenutnu dužinu niza (ne računajući elemente koji su jednaki nuli) a j predstavlja broj izvršavanja ciklusa (2) u slučaju kada ne dolazi do redukovanja dužine niza. Prema lemi 2.4.1 niz će biti uređen u odnosu na $<$ posle $n - 1$ iteraciju gde je n dužina niza, pa je taj uslov uzet kao kriterijum zaustavljanja. Druga mogućnost zaustavljanja $k = 1$ je u slučaju kada se $R = 0$ nikada ne dešava.

Kako je prema lemi 2.4.1 suma stalna a niz formiran po izvršenju $n - 1$ iteracija uređen u odnosu na $<$, teorema 2.4.1 je dokazana.

Prema prvom algoritmu tačnog sumiranja zaključujemo da transformisani niz ne samo da je normalizovan u odnosu na $<$ nego i poseduje sledeće svojstvo :

$$x_i \oplus x_{i+1} = x_i \qquad i = 1, 2, \ldots, n-1.$$

Ova osobina se može da uzme kao kriterijum zaustavljanja sledećeg algoritma tačnog sumiranja, jer može da se desi da je ta osobina niza ispunjena pre $n - 1$ iteraciju leme 2.4.1. To navodi na sledeću definiciju :

Definicija 2.4.3 Neka su dati brojevi $x, y \in S$ i neka $\square \in \{ \nabla, \triangle, \square_\mu, \mu = 0(1)b \}$ je zaokruživanje. Kažemo da je $x \overset{\boxplus}{<} y$ ako je

$$x \boxplus y = y.$$

Dakle, odmah se uočava da je $\overset{\boxplus}{<}$ relativizirana prema $\boxplus$, dok je Bohlander-ova relacija $<$ nezavisna od $\boxplus$.

Razlika između relacija $<$ i $\overset{\boxplus}{<}$ je sledeća :

ako je $x \overset{\boxplus}{<} y$ tada je $x < y$, ali ako je $x < y$ tada ne mora biti $x \overset{\boxplus}{<} y$.

Primer 2.4.1 Neka je dat sistem $S(10, 2, e_1, e_2)$ gde su e_1, e_2 celi brojevi i $e_1 < e_2$, i neka je $\square$ zaokruživanje ka najbližem broju u pokretnom zarezu. Ako je $x = 1100$ a $y = 54$ tada $y < x$ prema definiciji $<$, međutim

$$1100 \boxplus 54 = 1200$$
$$(1100 + 54) - 1200 = -46 .$$

Dakle , nije $y \overset{\boxplus}{<} x$.

Posledica 2.4.1 Na osnovu definicije 2.4.3 sledi da je $0 \overset{\boxplus}{<} x$ za $x \in S$ jer je

$$x \boxplus 0 = x$$

Definišimo takođe $x \overset{\boxplus}{<} 0$.

Lema 2.4.2 Neka je $S = S_{b,\ell}$ sistem u pokretnom zarezu i $\Box \in \{\nabla, \Delta, \Box_\mu, \mu = 0(1)b\}$ zaokruživanje i $\overset{\boxplus}{<}$ relacija prema definiciji 2.4.3. Ako $x, y \in S$ i $(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}) \in S^n$, tada je

(a) $r \overset{\boxplus}{<} s$,

(b) $x_{i+2}^{(k-1)} \overset{\boxplus}{<} x_{i+1}^{(k-1)} \wedge x_n^{(k)} < x_{n-1}^{(k)} < \dots < x_{i+1}^{(k)} < x_i^{(k)} \Rightarrow x_{i+1}^{(k)} \overset{\boxplus}{<} x_i^{(k)}$

za $i = n-k(-1)1$.

Dokaz. (a) $s = x \boxplus y$, $r = x + y - (x \boxplus y)$. Formirajmo

$$s \boxplus r = (x \boxplus y) \boxplus (x + y - (x \boxplus y)) =$$
$$(x \boxplus y + x + y - (x \boxplus y))_{\boxplus} = \Box(x + y) = x \boxplus y = s.$$

Dakle, prema definiciji je $r \overset{\boxplus}{<} s$.

(b) Ako je $\Box = \nabla$ tada je $R_i^{(k)} \geq 0$ ($i = 2(1)n$) pa je prema definiciji niza

$$x_i^{(k)} \boxplus x_{i+1}^{(k)} = x_i^{(k)} \boxplus (R_{i+1}^{(k)} \boxplus x_{i+2}^{(k-1)}) .$$

Zbog $x_{i+2}^{(k-1)} < R_{i+1}^{(k)}$ sledi da je $R_{i+1}^{(k)} \boxplus x_{i+2}^{(k-1)} \geq 0$ i tada je $x_i^{(k)} \boxplus x_{i+1}^{(k)} = x_i^{(k)}$. Dakle, $x_{i+1}^{(k)} \overset{\boxplus}{<} x_i^{(k)}$.

Ako je $\Box = \Delta$ tada je $R_i^{(k)} \leq 0$ ($i = 2(1)n$) pa je prema definiciji niza

$$x_i^{(k)} \boxplus x_{i+1}^{(k)} = x_i^{(k)} \boxplus (R_{i+1}^{(k)} \boxplus x_{i+2}^{(k-1)}) .$$

Zbog $x_{i+2}^{(k-1)} < R_{i+1}^{(k)}$ sledi da je $R_{i+1}^{(k)} \boxplus x_{i+2}^{(k-1)} \leq 0$ i tada je $x_i^{(k)} \boxplus x_{i+1}^{(k)} = x_i^{(k)}$. Dakle, $x_{i+1}^{(k)} \overset{\boxplus}{<} x_i^{(k)}$.

Neka $\Box \in \{\Box_\mu, \mu = 0(1)b\}$ i neka je $x_{i+2}^{(k-1)} \overset{\boxplus}{<} x_{i+1}^{(k-1)}$. Ako su $x_{i+1}^{(k-1)}$ i $x_{i+2}^{(k-1)}$ istoga znaka, tada je prva cifra u $x_{i+2}^{(k-1)}$ različita od b-1 izuzev u slučaju $\Box = \Box_b$.

Ako su $x_{i+1}^{(k-1)}$ i $x_{i+2}^{(k-1)}$ različitoga znaka, tada je prva cifra u $x_{i+2}^{(k-1)}$ različita od b-1 izuzev u slučaju $\Box = \Box_1$ pri čemu ostale cifre u $x_{i+2}^{(k-1)}$ moraju biti 0.

Ako su $R_{i+1}^{(k)}$ i $x_{i+2}^{(k-1)}$ istoga znaka, na osnovu prethodnog i $x_{i+2}^{(k-1)} < R_{i+1}^{(k)}$ sledi da je prva cifra u $R_{i+1}^{(k)}$ ista kao i prva cifra u $R_{i+1}^{(k)} \boxplus x_{i+2}^{(k-1)}$. Zbog $R_{i+1}^{(k)} < x_i^{(k)}$ i $x_{i+1}^{(k)} = R_{i+1}^{(k)} \boxplus x_{i+2}^{(k-1)}$ sledi $x_{i+1}^{(k)} \overset{\boxplus}{<} x_i^{(k)}$.

Ako su $R_{i+1}^{(k)}$ i $x_{i+2}^{(k-1)}$ različitoga znaka tada je $|R_{i+1}^{(k)} \boxplus x_{i+2}^{(k)}| \leq |R_{i+1}^{(k)}|$ pa važi tvrđenje teoreme.

Definicija 2.4.4 Neka je $S = S_{b,\ell}$ sistem u pokretnom zarazu i neka $x, y \in S$. Za x i y kažemo da se ne preklapaju u odnosu na $\overset{\boxplus}{<}$ ako je

$$x \overset{\boxplus}{<} y \quad \text{ili} \quad y \overset{\boxplus}{<} x.$$

Definicija 2.4.5 Niz realnih brojeva u pokretnom zarezu

$$x_1, \; x_2, \ldots, \; x_n$$

je normalizovan u odnosu na $\overset{\boxplus}{<}$ ako je

a) $x_i \neq 0$, $i = 1(1)n$,

b) x_i, x_{i+1}, $i = 1(1)n-1$ se ne preklapaju u odnosu na $\overset{\boxplus}{<}$,

c) $\exp(x_n) < \exp(x_{n-1}) < \ldots < \exp(x_1)$.

Normalizovan niz nazivamo RN-preciznost (realan niz).

Lema 2.4.3 Iskaz leme 2.4.1 je tačan ako relaciju $<$ zamenimo relacijom $\overset{\boxplus}{<}$.

Dokaz. Na osnovu leme 2.4.2

Sada navodimo drugi algoritam sumiranja.

Teorema 2.4.2 Neka je $S = S_{b,\ell}$ sistem u pokretnom zarezu i neka $x_i \in S_{b,\ell}$, $i = 1, 2, \ldots, n$ niz od n brojeva u pokretnom zarezu. Neka su $\nabla, \Delta, \square_\mu, \mu = 0(1)b: R^* \longrightarrow S^*$ zaokruživanja. Tada drugi algoritam tačnog sumiranja daje tačnu vrednost sume u obliku niza

$$x_1, \; x_2, \ldots, \; x_n, \; k \leqslant n$$

i dati niz je redukovan u odnosu na $\overset{\boxplus}{<}$.

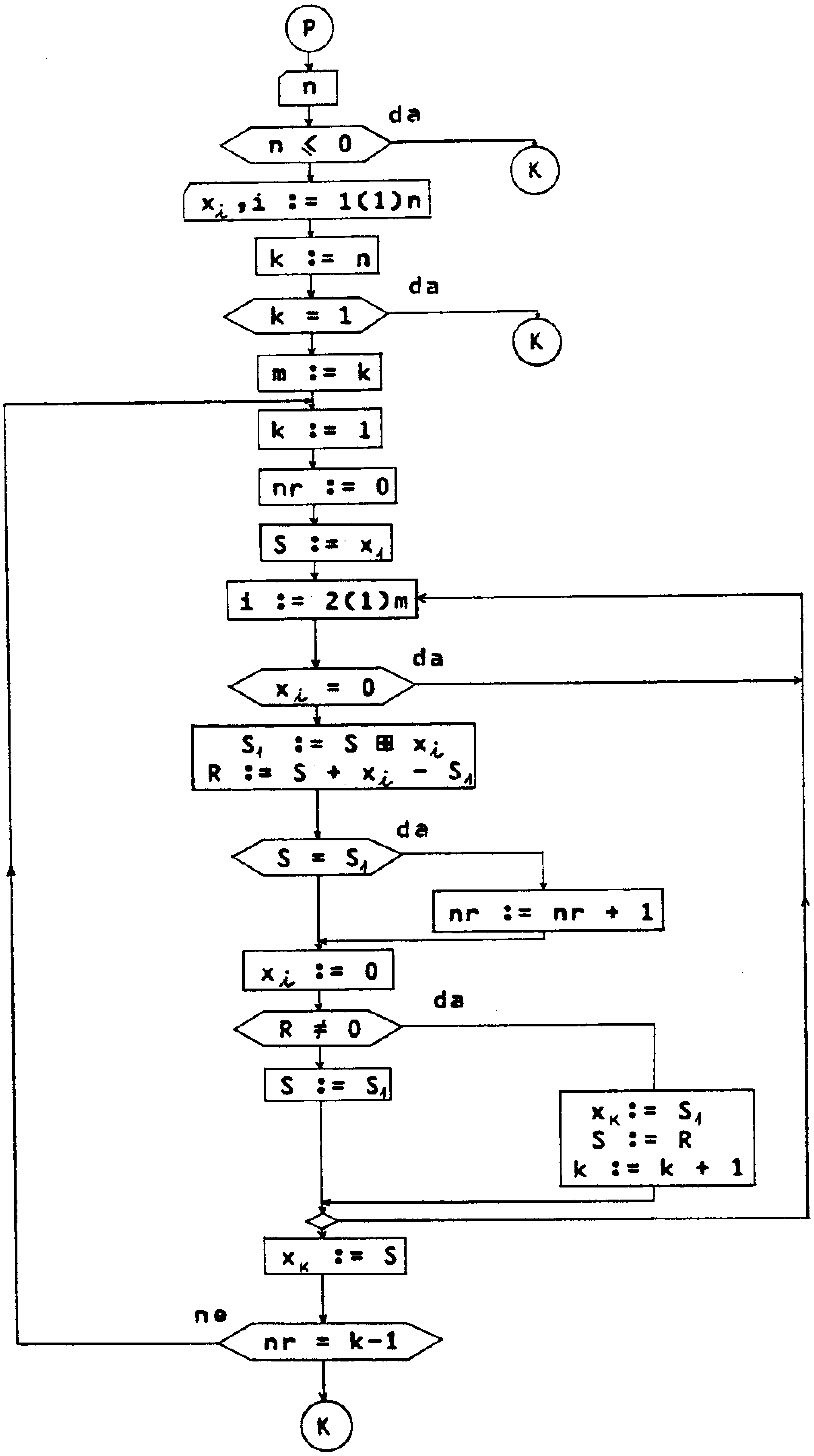
Dokaz. Prema lemi 2.4.3 posle j izvršavanja ciklusa (1) drugog algoritma tačnog sumiranja važi

$$x_n^{(j)} \overset{\boxplus}{<} x_{n-1}^{(j)} \overset{\boxplus}{<} \ldots \overset{\boxplus}{<} x_{n-j}^{(j)}.$$

Algoritam se zaustavlja ako je $k = 1$ ili $nr = k-1$. $k = 1$ nastupa u slučaju ako je $n = 1$ ili slučaj $R \neq 0$ ne nastupa pri dužini niza k.

Drugi kriterijum zaustavljanja nastupa ako je dužina niza k a broj slučajeva $x_{i-1} \boxplus x_i = x_{i-1}$ (promenljiva nr ih prebrojava) za $i = 2(1)k$ je $k-1$, što znači da je niz redukovan u odnosu na $\overset{\boxplus}{<}$.

(1)



Drugi algoritam tačnog sumiranja

Da bi algoritam bio brži članovi niza x_i jednaki nuli se izostavljaju , što inače ne bi smetalo uređenosti niza u odnosu na $\stackrel{\boxplus}{<}$.

Primer 2.4.2 Neka je dat niz x_i, i = 1,101 takav da je x_1 = 1.E+10 i x_i = 50. za i = 2,101. Uobičajeni postupak sumiranja

$$S := 0$$
$$S := S + x_i$$

daje vrednost sume S = 1.E+10, dok je tačna vrednost sume S = 10000005000. Bohlender-ov algoritam sumiranja, označen kao potprogram BSUMA(X,N,S) u prilozima, daje vrednost S = 10000005120 što predstavlja najbolju aproksimaciju tačne sume za mantisu dužine 24 i osnovu sistema 2.Tačan algoritam sumiranja označen kao TSUMA(X,N,K) u prilozima,daje vrednost tačne sume u obliku niza dužine 2: x_1= 10000005120, x_2=-120. Uočava se da x_1 predstavlja najbolju aproksimaciju vrednosti tačne sume za mantisu dužine 24 i osnovu sistema 2.

Pri izvršavanju bilo kog od napred navedenih algoritama sumiranja moguća su prekoračenja odozdo ili odozgo pa zato navodimo sledeću teoremu:

Teorema 2.4.3 Ako su $x_1, x_2, \dots, x_n \in S(b,l,e_1,e_2)$ tada se bilo koji od navedenih algoritama može izvršiti u sistemu $S_1 = S(b,l,e_1-l+1,e_2+\lceil \log_b n \rceil)$.

Dokaz. Broj sa najmanjim eksponentom dobijamo pri sumiranju brojeva

$$x_i = \pm 0,b_1 b_2 \dots b_l \cdot b^{e_1}, \quad x_j = \mp 0,b_1 b_2 \dots b_l' \cdot b^{e_1}$$

gde je $b_l \neq b_l'$. Tada je $x_i + x_j = \pm 0,(b_l - b_l') \cdot b^{e_1 - l + 1}$ pa je donja granica za eksponent $e_1 - l + 1$.

Broj sa najvećim eksponentom se dobija ako su svi članovi niza međusobno jednaki i jednaki najvećem broju u pokretnom zarezu i tada je suma

$$n \cdot (1 - b^{-l}) \cdot b^{e_2}$$

a eksponent sume je tada

$$\lceil \log_b n(1 - b^{-l}) \cdot b^{e_2} \rceil = \lceil \log_b n(1 - b^{-l}) \rceil + e_2 = \lceil \log_b n \rceil + e_2$$

pa je gornja granica za eksponent $e_2 + \lceil \log_b n \rceil$.

2.5 Tačna suma dva broja u pokretnom zarezu

U prethodnom delu su razmatrani algoritmi približnog i tačnog sumiranja koji kao jednu od operacija koriste tačno sumiranje dva broja u pokretnom zarezu. Dok je za realizaciju algoritama tačnog sumiranja bila dovoljna aritmetika dvostruke preciznosti , za realizaciju Bohlender-ovog algoritma bila je potrebna operacija tačnog sumiranja za brojeve dvostruke dužine. S druge strane da bi memorija i vreme računanja bili efikasnije upotrebljeni takođe je potrebna operacija tačnog sumiranja za brojeve dvostruke dužine. Sada komentarišamo načine na koje se operacija tačno sumiranje može realizovati. Problem je realizacija operacije $r(x,y) = x + y - (x \boxplus y)$, koja je razlika između tačne vrednosti zbira i neke aproksimacije zbira.

Za algoritam tačnog sumiranja potrebna je operacija tačno sumiranje dva broja u pokretnom zarezu, koja ja realizovana potprogramom TSXY(X,Y,S,R) navedenom u prilozima. Za realizaciju ove operacije bila je dovoljna dvostruka tačnost. Međutim, za operaciju tačno sabiranje kod Bohlender-ovog algoritma potrebno je ovu definisati nad realnim brojevima dvostruke dužine.

Ako raspolažemo nekim od programa za proširenu preciznost kao što je Brent-ov program opisan u /8/, tada bi algoritam izračunavanja $r(x,y)$ izgledao ovako:

1^0 ispitati razliku eksponenata x i y ex-ey. Ako je ex-ey $\geqslant$ $l+2$ tada je $s(x,y) = x$ ($s(x,y)$ predstavlja sumu x i y) a $r(x,y) = y$.

2^0 ako je ex-ey $< l+2$ tada treba x, y i x + y prevesti u proširenu preciznost odgovarajućim potprogramima MPCRM ili MPCDM.

3^0 naći tačnu sumu x + y i tačnu razliku $x + y - (x \boxplus y)$ i prevesti ih u tačnost izračunavanja odgovarajućim potprogramima MPCRM ili MPCMD.

Međutim, postavlja se pitanje da li u sistemu $S_{b,l}$ možemo izračunati $r(x,y)$ bez upotrebe proširene preciznosti. Odgovor na to pitanje nalazimo u /15/ i /31/ a teoremu navodimo iz /56/:

Teorema 2.5.1 Neka je aritmetika definisana sa $l+l_1$ ($l_1 \geqslant 1$) cifara u akumulatoru i sa zaokruživanjima $\square_b$ ili $\square_{b/2}$, pri čemu se pri formiranju sume $x \boxplus y$ u slučaju $b = 2$, na y ne primenjuje $\square_{b/2}$ nego $\square_b$ tada važi

$$s = x \boxplus y$$
$$yy = s \boxminus x$$
$$r = (y \boxminus yy) \boxplus (x \boxminus (s \boxminus yy)).$$

Teoremu navodimo bez dokaza.

2.6 Tačno množenje

Jedan od ciljeva ovog rada je i problem izračunavanja vrednosti polinoma. Za rešavanje ovoga problema potrebna je operacija tačnog množenja. Jvedimo prvo sledeću definiciju :

Definicija 2.6.1 Zaokruživanje $\square$ je korektno ukoliko je

$$\square x = \begin{cases} \nabla x \ ; \ x = \nabla x \\ \nabla x \text{ ili } \Delta x ; \ x \neq \nabla x \end{cases}$$

Sada navodimo rezultat iz /38/ koji se može koristiti za tačno množenje nizova iz definicija 2.4.5 normalizovanih u odnosu na $\overset{\boxplus}{<}$.

Teorema 2.6.1 Neka je dat sistem u pokretnom zarezu $S = S(b, l, e_1, e_2)$ i $x, y \in S$ i neka je $k \geqslant 2$ i $1/3 \leqslant k \leqslant 1/2$. Neka je

$$x = x_1 + x_2 \ , \ y = y_1 + u \ , \ u = u_1 + u_2$$

gde su x_1, y_1, u_1 aproksimacije x, y, u sa k cifara a x_2, u, u_2 preostali delovi dužine $l-k$ koji se ne preklapaju sa x_1, y_1, u_1 u smislu definicija 2.4.1 . Tada se tačan rezultat množenja x i y može predstaviti u obliku:

$$z_1 = \square\,(x \cdot y)$$
$$z_2 = \square\,(((((x_1 \cdot y_1 - z_1) + x_1 \cdot u) + y_1 \cdot x_2) + u_1 \cdot x_2) + u_2 \cdot x_2)$$

ukoliko je $\square$ korektno zaokruživanje.

Teoremu navodimo bez dokaza.

U prilozima je operacija tačnog množenja realizovana pomoću potprogama TPXY(X,Y,P,R) , za slučaj da su x i y dati u običnoj tačnosti , uz upotrebu dvostruke preciznosti. Ako računar raspolaže većom preciznošću od obične, tada bi za realizaciju pomenute operacije koristili teoremu 2.6.1 .

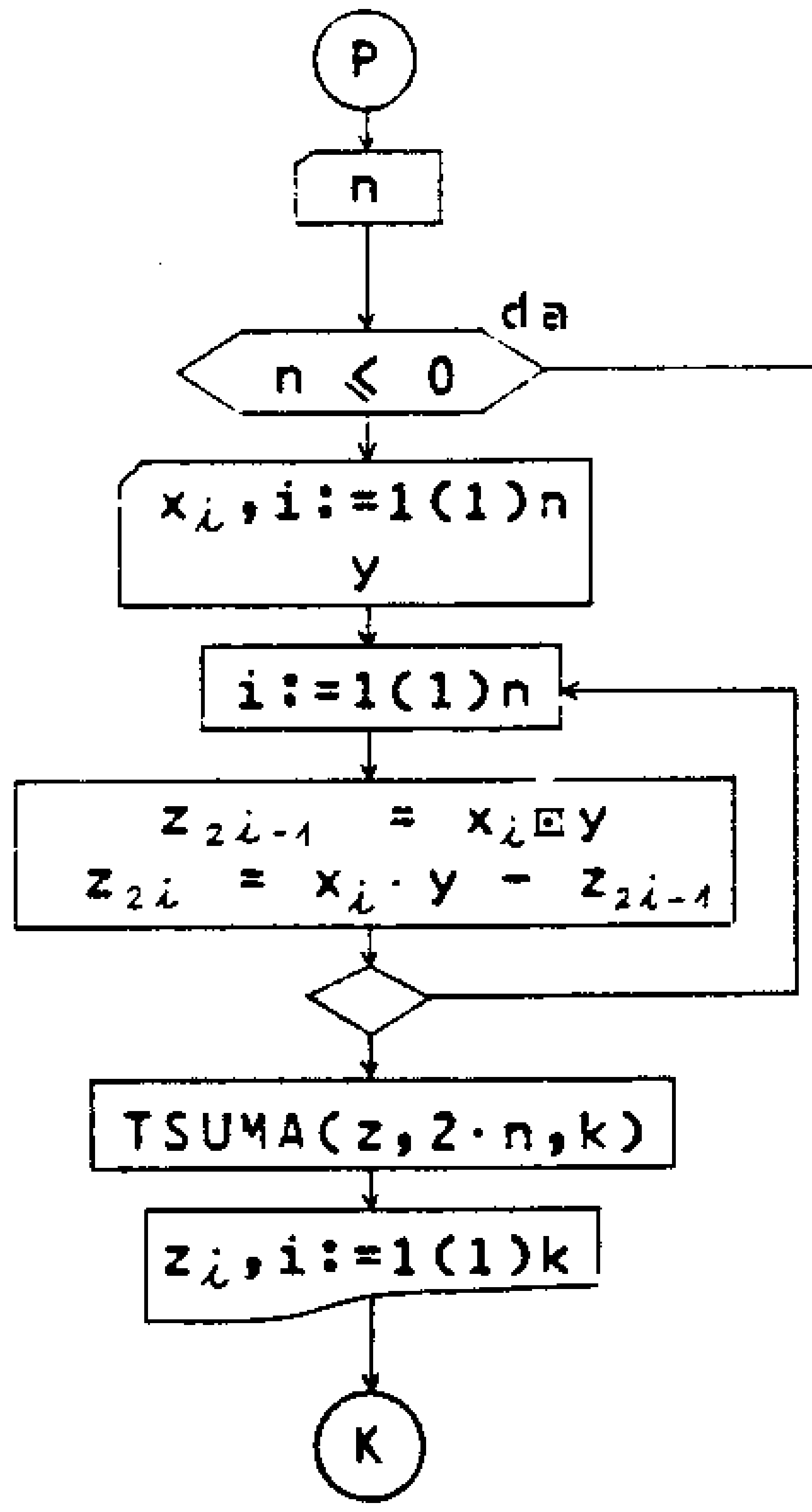
Ubuduće ćemo operaciju tačno množenje predstavljati u obliku

$$z_1 = x \boxdot y$$
$$z_2 = x \cdot y - z_1$$

Kako nam je cilj da uvedemo precizna izračunavanja navedimo sada algoritam množenja normalizovanog niza

$$x_1 \ , \ x_2 \ , \ \dots \ , \ x_n$$

u odnosu na $\overset{\boxplus}{<}$ i broja $y \in S$.



Algoritam tačnog množenja niza sa brojem u pokretnom zarezu

TSUMA označava drugi algoritam tačnog sumiranja koji je naveden napred a takođe i u prilozima. Za izvršenje algoritma tačnog množenja niza brojem važi sledeća teorema :

Teorema 2.6.3 Ako su $x_1, x_2, \ldots, x_n, y \in S(b,l,e_1,e_2)$ tada se algoritam tačnog množenja niza sa brojem u pokretnom zarezu može izvršiti u sistemu:

a) $S = S(b,l,2e_1-2l+1,2e_2)$ ako je $e_1 \geqslant 0$ ili $e_1 \leqslant 0$, $e_2 \geqslant 0$,

b) $S = S(b,l,\min(2e_2,2e_1-2l+1), e_2)$ ako je $e_2 \leqslant 0$.

Dokaz. Broj sa najmanjim eksponentom u sistemu S dobijamo množeći

$$0,(b-1)\ldots(b-1)\cdot b^{e_1} \cdot 0,(b-1)\ldots(b-1)\cdot b^{e_1} =$$

$$0,(b-1)\ldots(b-1)(b-2)_l\,0\ldots01_{2l}\cdot b^{2e_1} =$$

$$0,(b-1)\ldots(b-1)(b-2)\cdot b^{2e_1} + 0,1\cdot b^{2e_1-2\ell+1}.$$

Dakle, najmanji eksponent je $2e_1-2l+1$.

Broj sa najvećim eksponentom dobijamo množeći dva najveća broja u sistemu S

$$0,(b-1)\ldots(b-1)\cdot b^{e_2}\cdot 0,(b-1)\ldots(b-1)\cdot b^{e_2} =$$

$$0,(b-1)\ldots(b-1)(b-2)_{\ell}\,0\ldots01_{2\ell}\cdot b^{2e_2} =$$

$$0,(b-1)\ldots(b-1)(b-2)_{\ell}\cdot b^{2e_2} + 0,1\ b^{2e_2-2\ell+1}.$$

Sledi da je najveći eksponent $2e_2$.

U slučaju b) donja granica je $\min(2e_2, 2e_1-2l+1)$, a gornja granica je e_2 jer je $e_2 \leqslant 0$.

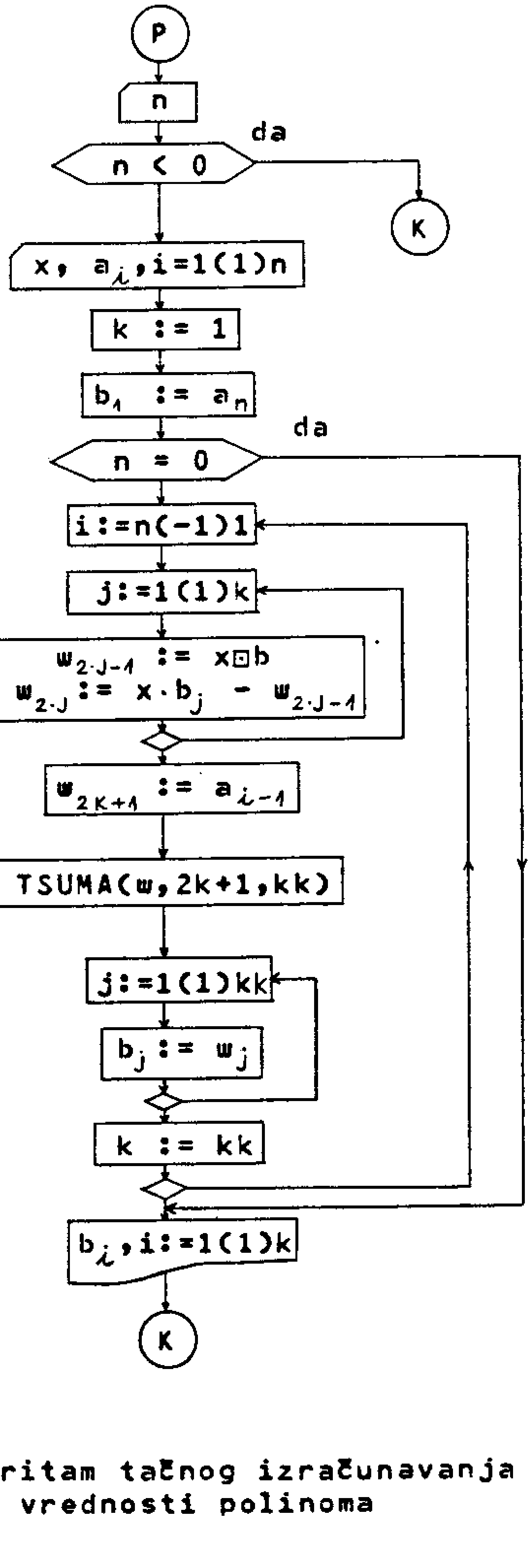
2.7 Tačna vrednost i najbolja aproksimacija vrednosti polinoma

Problem izračunavanja vrednosti polinoma nalazi primenu kod izračunavanja vrednosti funkcija definisanih pomoću polinoma. Pretpostavićemo da je polinom dat u obliku

$$p(x) = a_n x^n + a_{n-1}\cdot x^{n-1} + \ldots + a_1\cdot x + a_0.$$

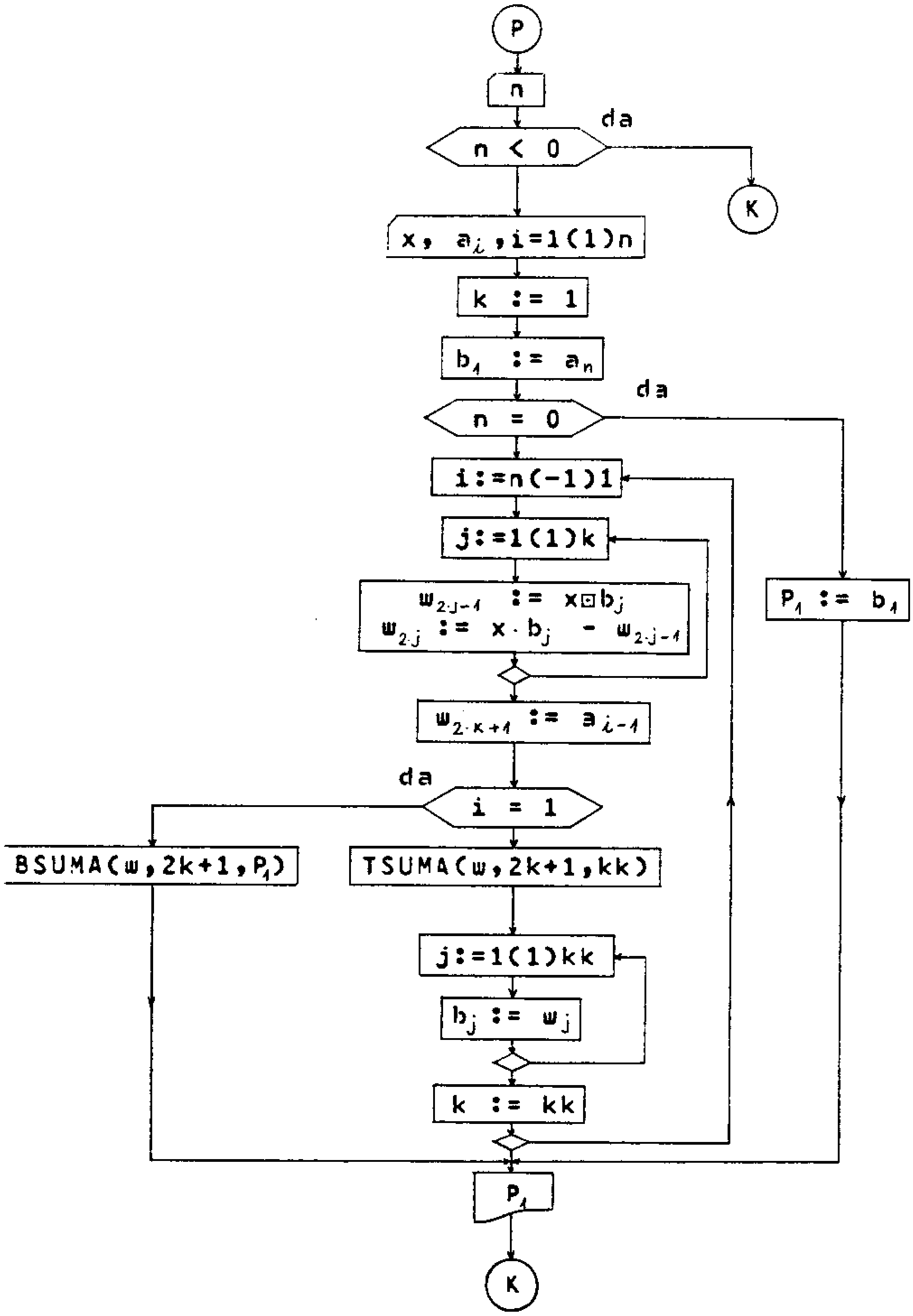
Algoritam ima kao ulaz vrednost promenjive x i niz a_i, $i=0(1)n$ koeficijenata a izlaz je niz b_i, $i=1(1)k$ koji sadrži tačnu vrednost polinoma. w_i, $i=1(1)2n+1$ pretstavlja radni niz a n označava stepen polinoma. TSUMA označava algoritam tačnog sumiranja a k predstavlja dužinu redukovanog niza posle izvršavanja algoritma tačnog sumiranja.

Sada navodimo algoritme za izračunavanje tačne vrednosti i najbolje aproksimacije vrednosti polinoma.



Algoritam tačnog izračunavanja vrednosti polinoma

Algoritam ima ulaz : n stepen polinoma, argument x i niz a_i, i=0(1)n koeficijenata. Izlaz je niz b_i, i=1(1)k koji sadrži tačnu vrednost polinoma, a k je dužina niza b. w_i,i=1(1)2k+1 predstavlja radni niz. TSUMA označava drugi algoritam tačnog sumiranja.



Algoritam izračunavanja najbolje aproksimacije polinoma

Razlika između algoritma za tačno izračunavanje i algoritma za izračunavanja najbolje aproksimacije vrednosti polinoma je sledeća: u poslednjem izvršavanju ciklusa po i za i = 1 koristi se Bohlander-ov algoritam sumiranja označen sa BSUMA a vrednost polinoma sadrži promenljiva P_1 umesto niza b. Upotreba Bohlender-ovog algoritma može znatno da ubrza izračunavanje najbolje aproksimacije u odnosu na tačno izračunavanje, kada je vrednost stepena polinoma velika.

Napred navedenim algoritmima za izračunavanje vrednosti polinoma se rešava problem izračunavanja vrednosti funkcija koje se dobro aproksimiraju pomoću polinoma.

Primer 2.7.1 Neka je dat polinom iz /7/

$$p(x) = 23616 \cdot x^5 - 161522 \cdot x^4 + 401773 \cdot x^3 - 406754 \cdot x^2 + 87511 \cdot x + 66576 .$$

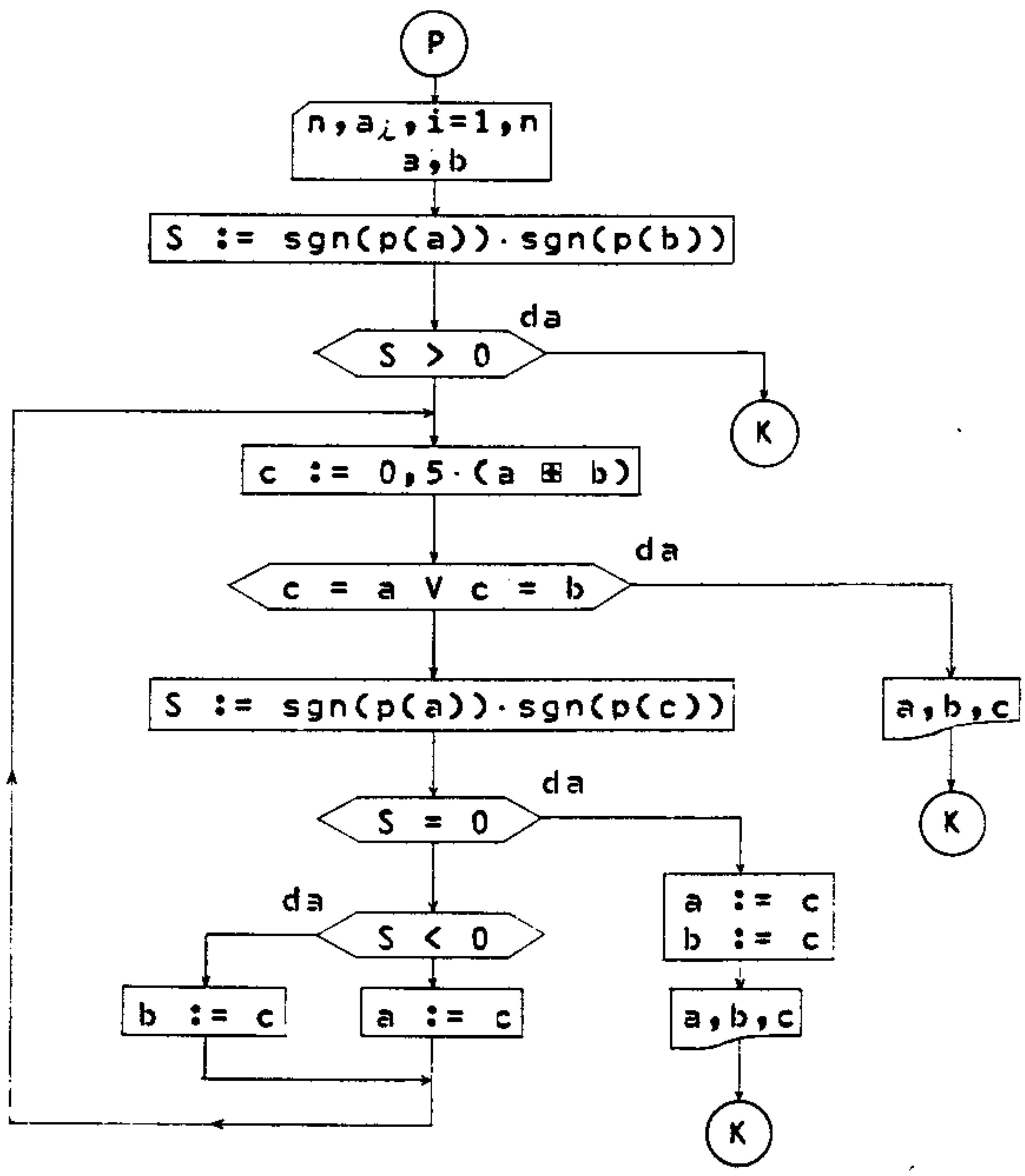
Ako izračunamo vrednost polinoma na različite načine imamo sledeće:

X	uobičajeno izračunavanje	Horner-ov algoritam	prvi član niza tačnog algoritma
0.1780000E+01	-0.1718750E+00	-0.6250000E-01	-0.5120540E-07
0.1780100E+01	-0.4687500E-01	-0.4687500E-01	-0.3409607E-07
0.1780200E+01	0.7812500E-01	-0.4687500E-01	-0.2053516E-07
0.1780300E+01	-0.4687500E-01	-0.3125000E-01	-0.1043320E-07
0.1780400E+01	0.9375000E-01	-0.1250000E+00	-0.3583786E-08
0.1780500E+01	-0.3125000E-01	-0.1562500E-01	0.3364225E-09
0.1780600E+01	-0.4687500E-01	-0.7812500E-01	0.1767312E-08
0.1780700E+01	-0.1718750E+00	-0.4687500E-01	0.1268999E-08
0.1780800E+01	-0.3125000E-01	-0.3125000E-01	-0.4873926E-09
0.1780900E+01	0.2187500E+00	-0.7812500E-02	-0.2710584E-08
0.1781000E+01	-0.3125000E-01	0.2343750E-01	-0.4492207E-08
0.1781100E+01	-0.3125000E-01	-0.5468750E-01	-0.4806776E-08
0.1781200E+01	-0.2812500E+00	-0.2343750E-01	-0.2511662E-08
0.1781300E+01	-0.3125000E-01	0.2343750E-01	0.3642758E-08
0.1781400E+01	0.9375000E-01	-0.3906250E-01	0.1504674E-07

Pod uobičajenim izračunavanjem podrazumeva se zapis polinoma kao izraza koji definiše sam polinom. Uočava se da ni Horner-ov algoritam ni uobičajeno izračunavanje ne mogu da budu upotrebljeni za nalaženje intervala u kojima se nalaze nule polinoma, a o redu veličine izračunatih vrednosti najbolji komentar daje sama tabela. Rezultati u poslednjoj koloni su isti kao u /7/.

2.8. Nalaženje jednostruke realne nule polinoma sa maksimalnom preciznošću i najbolje aproksimacije nule

Pod maksimalnom preciznošću smatramo interval između čijih krajeva se nalazi najviše jedan broj u pokretnom zarezu. Pretpostavljamo da je dat polinom $p(x)$ i da na krajevima intervala $[a,b]$ ima različite znake i da je samo jedna nula na $[a,b]$. Koristi se metoda polovljenja intervala da bi odredili najuži interval kome pripada nula, odnosno najbolju aproksimaciju nule u smislu zaokruživanja.



Algoritam izračunavanja realne nule polinoma sa maksimalnom preciznošću i najbolje aproksimacije nule

ajemo neke napomene koje se odnose na prethodni algoritam.
(x) se izračunava prema algoritmu za izračunavanje najbolje
proksimacije vrednosti polinoma ili prema tačnom algoritmu.
algoritmu koristimo sgn funkciju što je danas uobičajeno
formulaciji ovog algoritma, da bi izbegli prekoračenje
dozdo.
ostoje dva slučaja kada se algoritam zaustavlja :

° nula polinoma nije predstavljiv broj u računaru i u tom
lučaju algoritam daje najuži predstavljiv interval u
ačunaru koji sadrži nulu i kraj intervala koji je najbolja
proksimacija nule u smislu zaokruživanja.

° nula polinoma je predstavljiv broj u računaru i u tom
lučaju se maksimalna preciznost i najbolja aproksimacija
oklapaju.

rimer 2.8.1 Neka je dat polinom iz primera 2.7.1. Na osnovu
rethodnog primera evidentno je da polinom ima nule na
ledećim intervalima :
1.7804,1.7805), (1.7807,1.7808), (1.7812,1.7813).
otprogram NULA daje sledeće :

najuži interval

1.7804877758026123046875,1.780487895011901855546875)

najbolja aproksimacija

1.780487895011901855546875

najuži interval

1.7807763814926147460937 5,1.780776500701904296875 00)

najbolja aproksimacija

1.78077650070190429687500

najuži interval

(1.781249880790710449 21875,1.78125)

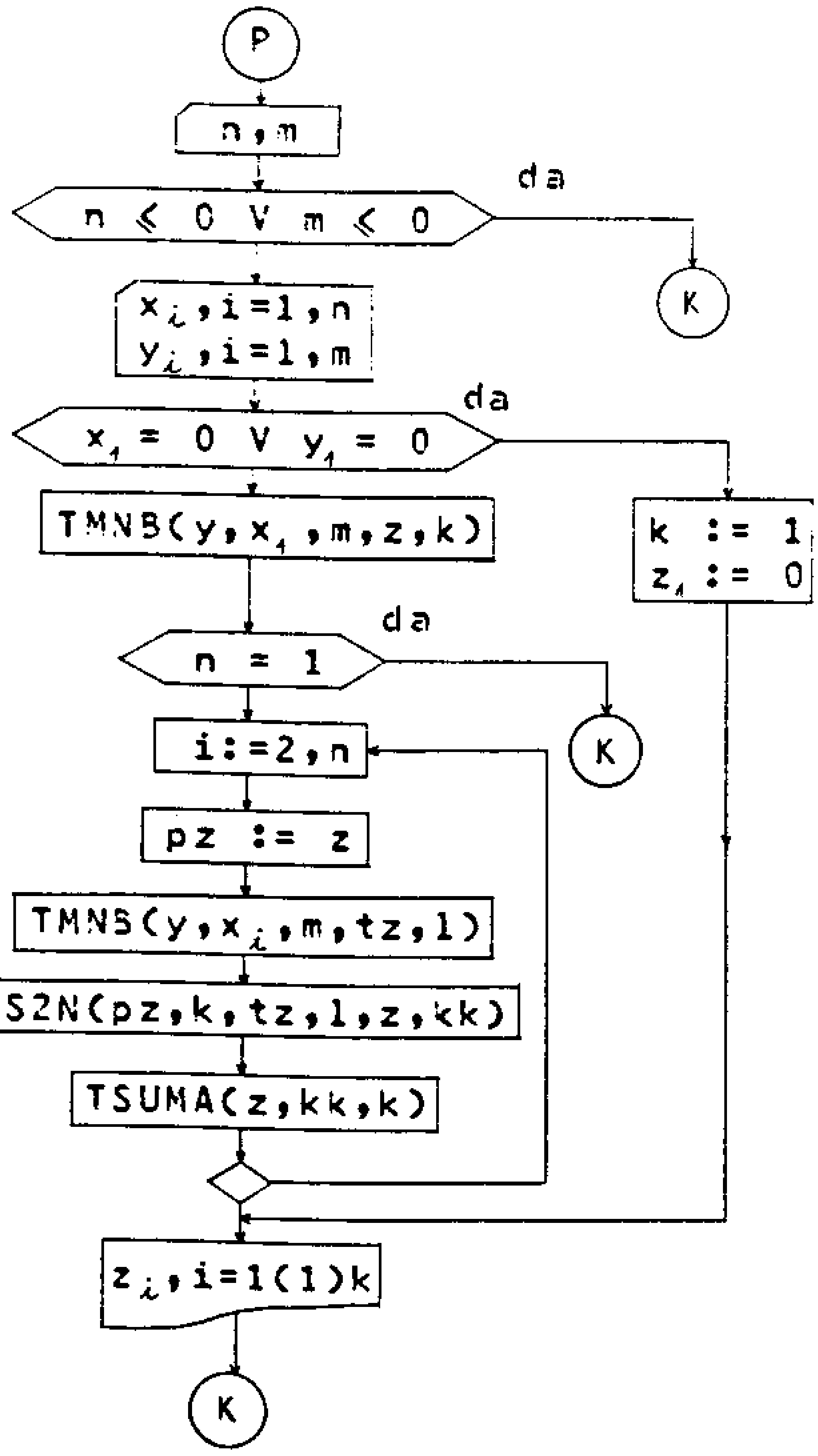
najbolja aproksimacija

1.78125 .

apominjemo da su gore navedeni intervali u decimalnom
apisu susedni brojevi u pokretnom zarezu u binarnom zapisu.

2.9. Tačno množenje dva niza

Neka su dati nizovi x_i, $i=1(1)n$, i y_i, $i=1(1)m$. Proizvod ovih nizova je niz z_i, $i=1(1)k$.



Algoritam tačnog množenja dva niza

Dajemo neke napomene koje se odnose na definisani algoritam. Ako je navedeno slovo bez indeksa onda je to oznaka za niz, a ako je navedeno slovo sa indeksom onda je to oznaka za odgovarajući član pomenutog niza.

Da bi upotrebili što manje prostora i vremena za izvršenje algoritma tačnog množenja dva niza, niz y je pomnožen sa x_i i u sledećem koraku je y pomnožen sa x_{i+1}. Zatim su prethodni i trenutni proizvod sortirani u odnosu na eksponente pomoću

S2N dajući niz z. Dakle, S2N uzima prethodni proizvod pz i trenutni proizvod tz (oba sortirana u odnosu na eksponente) dajući sortiran niz z (u odnosu na eksponente). Posle toga se poziva TSUMA za redukciju i normalizaciju z.

Ako bi pomnožili nizove x i y i tada pozivali TSUMA tada bi imali (m+1)n članova za sumiranje što je znatno, jer TSUMA deluje kao sortiranje.

2.10 Recipročna vrednost niza

Neka je dat niz x_i, i=1(1)n. Sa y_i, i=1(1)m označimo niz koji predstavlja recipročnu vrednost niza x. Recipročnu vrednost niza izračunavamo primenom poznate iterativne metode za izračunavanje recipročne vrednosti broja x :

$$y_{n+1} = y_n (2 - x \cdot y_n), \quad n = 0, 1, 2, \ldots$$

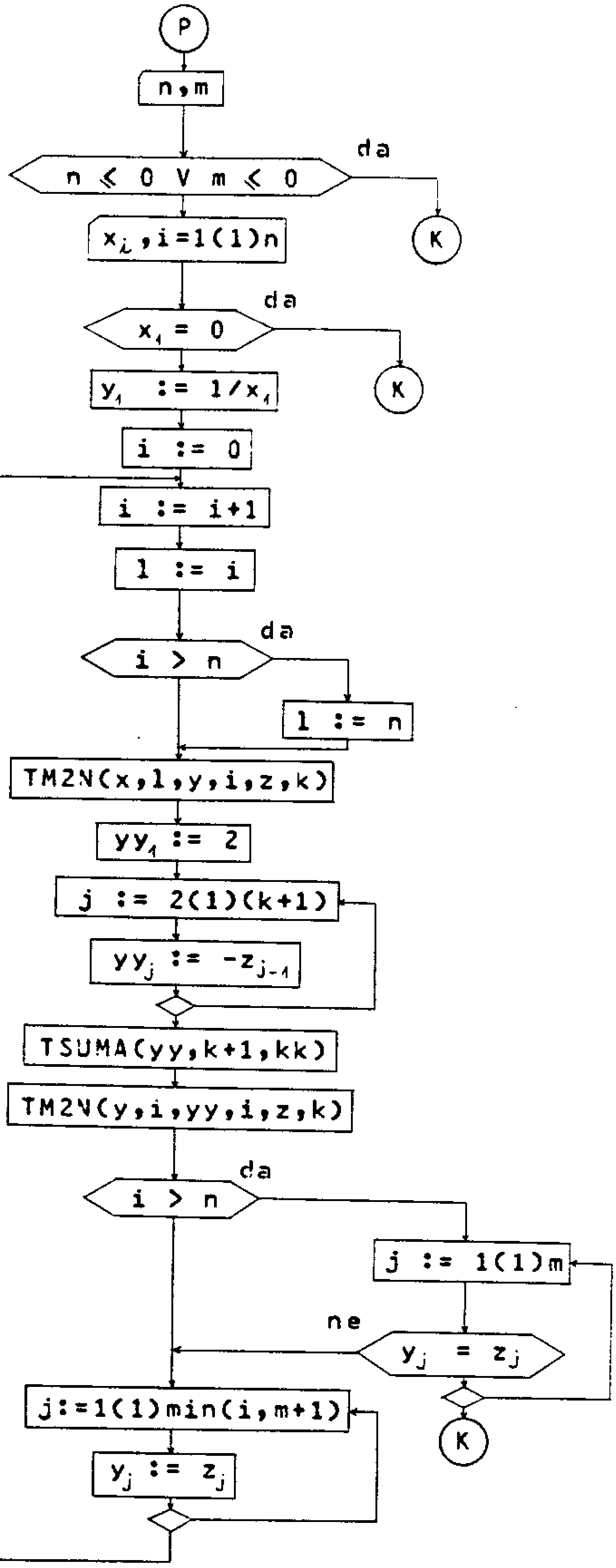
Algoritam daje recipročnu vrednost niza u obliku niza dužine m. Promenjiva i u algoritmu ima dvostruku ulogu :

1° omogućuje dinamičku promenu dužine nizova i sa procesom računanja dužina nizova se povećava do određene granice ,

2° omogućava nastavljanje iterativnog procesa do konačnog zaustavljanja algoritma sa nizom y dužine m.

U toku računanja dužina niza x predstavljena promenljivom l se povećava dok ne dostigne dužinu n. To se čini zbog toga što nije potrebno od samoga početka izračunavanja računati sa svim članovima niza x. Slično se čini i sa nizovima y, yy i z. Niz y u toku računanja ima najveću dužinu m+1 da bi se obezbedila njegova tačnost na m članova. Kriterijum za zaustavljanje se sastoji u poklapanju prethodne i sledeće iteracije na m članova što je korektan kriterijum, videti na primer /16/.

Algoritam je realizovan kao potprogram RECN(X,N,Y,M) u prilozima.



Algoritam za izračunavanje recipročne vrednosti niza

2.11 Algoritam deljenja dva niza

Uvođenje operacije recipročne vrednosti niza kao mogućnost uvođenja operacije deljena dva niza ima sledeći nedostatak: deljenje uvedeno preko recipročne vrednosti niza nije tačno u slučaju kada je operacija deljenja dva niza konačna.

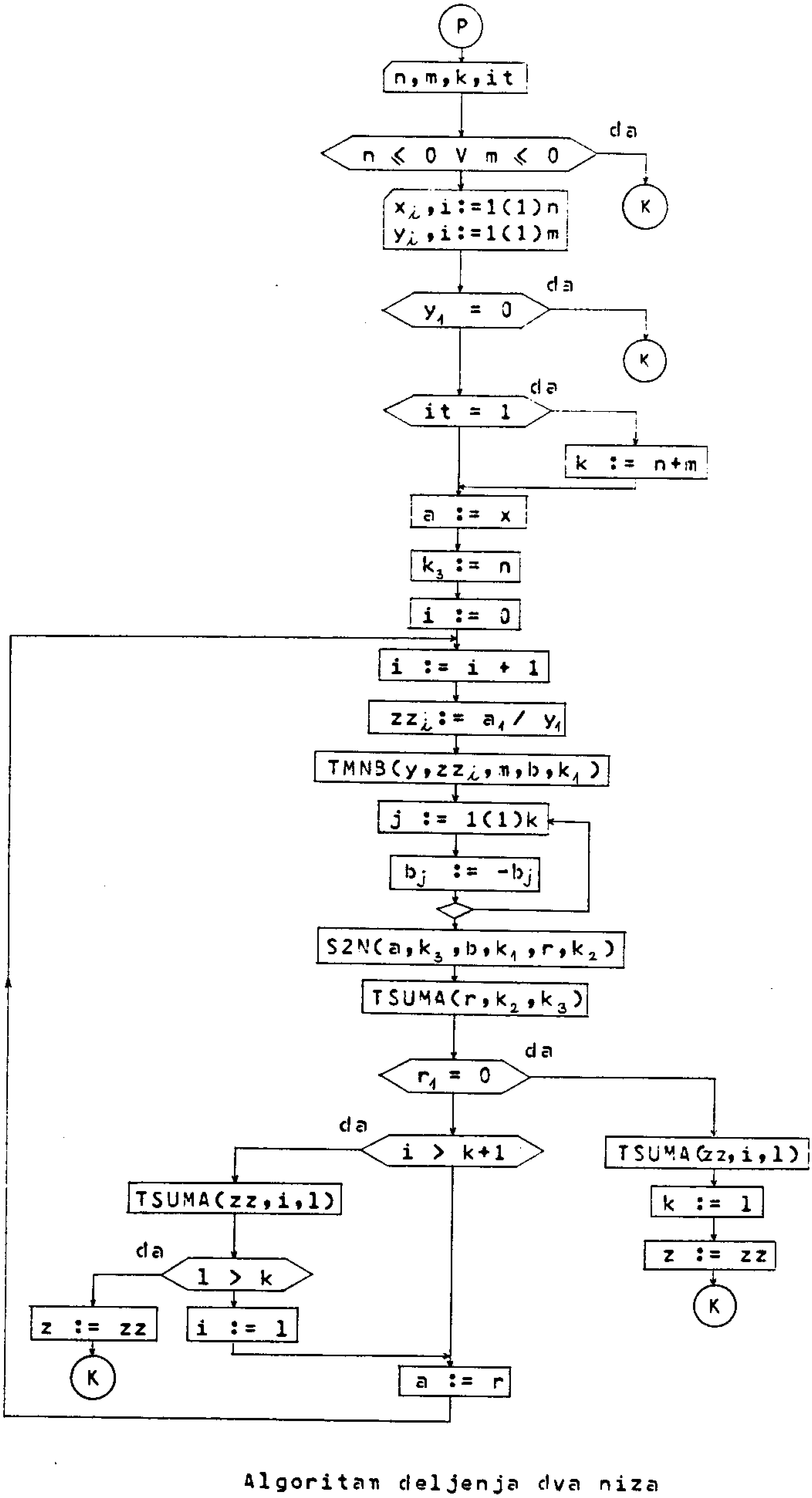
Na algoritmu deljenja dva niza uočava se bitna razlika između Brent-ove proširene preciznosti i RN-preciznosti: Brent-ova proširena preciznost zahteva definisanje deljenja ne koristeći postojeće deljenje brojeva u pokretnom zarezu. RN-preciznost koristi deljenje brojeva u pokretnom zarezu pri čemu je i pitanje rezervnih cifara relativno jednostavno rešeno što nije slučaj kod Brent-ove preciznosti.

U algoritmu važnu ulogu ima parametar it. Ako smo sigurni da je deljenje tačno (što može biti vrlo čest slučaj na primer kod raznih postupaka za tačno izračunavanje determinante) , tada stavljamo da je it = 1 i tada algoritam sam određuje dužinu izlaznog niza z, k. Ako je it ≠ 1 tada algoritam uzima parametar k (dužina izlaznog niza), kao kriterijum za zaustavljanje, ako deljenje nije konačno ili ako uzimamo neku aproksimaciju tačnog deljenja.

Pri opisu algoritma važe iste napomene kao i za prethodne algoritme nad nizovima: ako je navedena promenljiva bez indeksa onda ona označava ime niza, a ako je navedena sa indeksom onda označava odgovarajući član pomenutog niza.

Algoritam je realizovan kao potprogram D2N(X,N,Y,M,Z,K) u prilozima.

Sada navodimo algoritam deljenja dva niza.



Algoritam deljenja dva niza

3.1 Izračunavanje vrednosti aritmetičkog izraza

U drugom delu ovoga rada razmatran je problem tačnog izračunavanja vrednosti sume , proizvoda , polinoma i količnika . U slučajevima izračunavanja vrednosti sume , proizvoda i polinoma bila je dovoljna konačna memorija , te smo zbog toga mogli njihove vrednosti da tačno predstavimo. Međutim , operacija deljenja može da daje brojeve sa beskonačno mnogo cifara,te je zbog toga neophodno naći način kojim bi se vrednost aritmetičkog izraza dobila na određen broj tačnih cifara. Problemom izračunavanja aritmetičkog izraza bavi se H. Bohm u /6/ i /7/ , S. M. Rump u /50/ i P. Lohner, H. C. Fischer i drugi u /35/. Pristup koji nalazimo u pomenutim radovima je sledeći: pogodnom transformacijom se dati aritmetički izraz transformiše u sistem linearnih jednačina , čiji su koeficijenti brojevi u pokretnom zarezu. Sistem linearnih jednačina je prema /49/ i /51/ moguće rešiti sa tačnošću do poslednjeg bita (rešenja su intervali čiji su krajevi susedni brojevi u pokretnom zarezu) . Metode rešavanja sistema linearnih jednačina zahtevaju nalaženje približne inverzne matrice datog sistema da bi dobili izraze za iterativno rešavanje odgovarajućeg sistema u intervalnoj aritmetici . Pri tome ti izrazi predstavljaju skalarne proizvode za koje se zahteva da moraju biti precizni (najbolja aproksimacija tačne vrednosti na dužini mantise na kojoj računamo).
Dakle, u ovom pristupu uočavaju se dva bitna koraka:

1^o Transformacija aritmetičkog izraza u sistem linearnih jednačina i

2^o Rešavanje pomenutog sistema linearnih jednačina iterativno pomoću približne inverzne matrice.

Ovim postupkom se izračunavaju i relativno jednostavni aritmetički izrazi kao što je polinom,čije smo izračunavanje rešili u drugom delu ovoga rada.
U ovom delu će biti izložen jedan pristup za izračunavanje vrednosti aritmetičkog izraza u proširenoj preciznosti: odrediti broj cifara a priori sa kojim treba računati da bi se na kraju dobila tačnost na određen broj cifara. Na taj način se aritmetički izraz izračunava uobičajeno a time i nije potrebna približna inverzna matrica. Pri tome broj cifara za izračunavanje datog aritmetičkog izraza može biti znatno veći nego što je potrebno za konkretne vrednosti promenljivih u aritmetičkom izrazu. Međutim,to je nedostatak svih ocena a priori: na primer ako koristimo procene koje se dobijaju za broj cifara za tačno izračunavanje determinante u /3/ , /10/ , /11/ i /53/ evidentno je da su procene pesimistične, tj koristi se znatno veći broj cifara nego što je potrebno.

Pristup problemu tačnosti preko proširene preciznosti je mnogo primenjiviji sa gledišta postojećih računara : nije potrebna modifikacija niti proširenje postojeće aritmetike (nove operacije) , što se inače zahteva u intervalnoj aritmetici(videti /33/, /34/ i /35/). Takođe nije potrebno ni proširenje programskog jezika, što se inače čini za praktičnu primenu intervalne aritmetike . RN-preciznost opisana u drugom delu ovoga rada je relativno jednostavna za ugrađivanje kao što je učinjeno u prilozima. Međutim, ako je problem većeg obima , na primer broj jednačina sistema je velik, tada je vreme izvršenja algoritma znatno. Razlog tome je programsko rešenje operacija : izdvajanje i postavljanje eksponenta i tačno sabiranje i množenje . Hardversko rešenje pomenutih operacija bi znatno ubrzalo predložene algoritme. To je inače i učinjeno u primenama intervalne aritmetike jer bi i ova bila neupotrebljiva bez velikog broja hardverskih rešenja operacija.

Da bismo izložili osnove ocena a priori za proizvoljan algoritam u proširenoj preciznosti potrebne su definicije osnovnih operacija intervalne aritmetike.

Definicija 3.1.1 Podskup skupa R_b oblika

$$A = [a_1 , a_2] = \{x \mid a_1 \leqslant x \leqslant a_2 , a_1 , a_2 \in R_b\}$$

se naziva interval.

Skup intervala nad R_b označavamo $I(R_b)$.

Definišimo sada operacije nad intervalima.

Definicija 3.1.2 Neka je $* \in \{+,-,\cdot,/\}$ binarna operacija nad skupom R_b. Ako je $A,B \in I(R_b)$, tada je

$$A*B = \{x = a*b \mid a \in A, b \in B\}$$

binarna operacija nad $I(R_b)$.

Nije teško binarne operacije nad intervalima $A = [a_1 , a_2]$ i $B = [b_1 , b_2]$ definisati eksplicitno

$$A + B = [a_1 + b_1 , a_2 + b_2]$$

$$A - B = [a_1 - b_2 , a_2 - b_1]$$

$$A \cdot B = [\min(a_1 b_1 , a_1 b_2 , a_2 b_1 , a_2 b_2), \max(a_1 b_1 , a_1 b_2 , a_2 b_1 , a_2 b_2)]$$

$$A/B = [a_1 , a_2] \cdot \left[\frac{1}{b_2} , \frac{1}{b_1}\right] \qquad \text{ako } 0 \notin B$$

Lema 3.1.1 Neka je dat sistem $S = S(b,l,e_1,e_2)$ i niz brojeva u pokretnom zarezu $x_1, x_2, \ldots, x_n \in S$ gde je $n \leqslant b$ i $n, b \in N$ i $b > 1$. Za izračunavanja sume niza $x_1, x_2, \ldots, x_n$ sa tačnošću do b-1 jedinica na zadnjem mestu mantise potrebna je jedna rezervna cifra i jedna cifra za prekoračenje.

Dokaz. Ako su svi članovi istoga znaka i niz je dužine b tada nije potrebna rezervna cifra nego cifra za prekoračenje. Jedna cifra za prekoračenje je dovoljna, jer ako suma ima maksimalnu vrednost tj. svi x_i su oblika

$$x_i = \pm 0,(b-1) \ldots (b-1)_l \cdot b^{e_2} \quad i = 1(1)b$$

tada je

$$S = \pm(b-1),(b-1) \ldots (b-1)_{l-1}\, 0_l \cdot b^{e_2}$$

gde indeks označava poziciju cifre u okviru mantise.
Ako su članovi niza različitog znaka, tada odredimo sume S_b^+ i S_b^- pozitivnih i negativnih članova niza, dovođenjem svih članova niza na maksimalni eksponent niza $x_1, x_2, \ldots, x_n$. Na osnovu prethodnog razmatranja pri formiranju suma S_b^+ i S_b^- nije potrebna rezervna cifra nego cifra za prekoračenje. Pri određivanju sume brojeva S_b^+ i S_b^- može da se pojavi 0 na prvom mestu mantise zbira ako je razlika eksponenata S_b^+ i S_b^- veća ili jednaka 1. Zbog ove mogućnosti potrebna je jedna rezervna cifra da bi pri normalizaciji dužina mantise iznosila l. Kako pri sumiranju cifara desno od l+1 cifre prenos može da bude najviše b-1 zaključujemo da se vrednost dobijene sume može najviše razlikovati za b-1 na l-tom mestu u odnosu na tačnu vrednost sume na dužini l.

Napomena 3.1.1 Rezervna cifra i cifra za prekoračenje nisu istovremeno potrebne pri sabiranju dva broja.

Teorema 3.1.1 Neka je dat sistem $S = S(b,l,e_1,e_2)$ i niz $x_1, x_2, \ldots, x_n \in S$, $n \in N$, $n > 1$. Za izračunavanje sume niza sa tačnošću do b-1 jedinica na l-tom mestu mantise potrebno je $\lceil \log_b n \rceil$ rezervnih cifara i $\lceil \log_b n \rceil$ cifara za prekoračenje.

Dokaz. Dokaz teoreme ćemo izvesti po principu regresivne indukcije, odnosno dokazaćemo da je tvrđenje tačno za beskonačno mnogo $n \in \{b, b^2, \ldots\}$ i da iz pretpostavke da je tačno za n dokazaćemo da važi za n-1.
Tvrđenje je tačno za n = b na osnovu leme 3.1.1.
Pretpostavimo sada da je tvrđenje tačno za $n = b^k$ tj. da je za formiranje sume potrebno k cifara za prekoračenje. Kako se maksimalan broj cifara za prekoračenje dobija ako su svi članovi niza jednaki maksimalnom broju u pokretnom zarezu, pretpostavićemo da imamo $n = b^{k+1}$ članova i da su svi jednaki maksimalnom broju u pokretnom zarezu. Ako b^{k+1} članova niza sumiramo po b^k članova imaćemo b suma oblika

$$S_i = \pm\underbrace{(b-1)\ldots(b-1)}_{k},(b-1)\ldots(b-1)_{l-k}\,0_{l-k+1}\ldots 0_l \cdot b^{e_2}, \quad i = 1(1)b$$

jer je po pretpostavci potrebno k cifara za prekoračenje.
Ako sada sumiramo sume S_i, $i=1(1)b$ tada će biti potrebna jedna cifra više za prekoračenje prema lemi 3.1.1. Dakle, imamo ukupno k+1 cifru za prekoračenje.
Dokažimo sada da je za izračunavanje sume niza dužine n potrebno $\lceil \log_b n \rceil$ rezervnih cifara i da se dobijena suma razlikuje od tačne sume najviše za b-1 jedinica na 1-tom mestu mantise.
Tvrđenje je tačno za $n = b$ prema lemi 3.1.1.
Pretpostavimo sada da je tvrđenje tačno za $n = b^k$ tj. da je za sumiranje b^k članova niza potrebno k rezervnih cifara i da se dobijena vrednost razlikuje od tačne vrednosti sume za najviše b-1 jedinica na 1-tom mestu.
Neka sada imamo $n = b^{k+1}$ članova niza i transformišimo sve članove niza u brojeve u pokretnom zarezu sa maksimalnim eksponentom. Izvršimo sumiranje na sledeći način: formirajmo b grupa po b^k članova i izvršimo sumiranja ovih grupa. Prema induktivnoj pretpostavci za nalaženje sume b^k elemenata sa tačnošću b-1 jedinica na 1-tom mestu mantise potrebno je k rezervnih cifara. Označimo odgovarajuće sume sa

$$S_1', S_2', \dots, S_b'.$$

Ako bismo sumirali ove sume tada bi se vrednost dobijene sume prema lemi 3.1.1 razlikovala za b-1 jedinica na 1-1 mestu mantise jer se sume $S_1', S_2', \dots, S_b'$ razlikuju od odgovarajućih tačnih suma za b-1 na 1-tom mestu mantise. Odatle sledi da sume $S_1', S_2', \dots, S_b'$ treba odrediti pomoću k+1 rezervnih cifara da bi se dobijene vrednosti ovih suma razlikovale za b-1 jedinica od odgovarajućih tačnih suma na 1+1 mestu. Dakle, ukupno je potrebno k+1 rezervnih cifara za izračunavanje sume b^{k+1} elemenata sa tačnošću do b-1 jedinica na 1-tom mestu.
Do sada smo dokazali da je tvrđenje tačno za beskonačno mnogo n oblika $n = b^k$, $k=1,2,\dots$.
Pretpostavimo da je tvrđenje tačno za n. Ako je $\lceil \log_b n \rceil = \lceil \log_b (n-1) \rceil$ tada je tvrđenje tačno za n-1.
Ako je $\lceil \log_b n \rceil \neq \lceil \log_b (n-1) \rceil$ što je tačno za $n = b^k + 1$ tada je $\lceil \log_b n \rceil = k+1$ a $\lceil \log_b (n-1) \rceil = k$ pa ako je tvrđenje tačno za n po pretpostavci, tada je tačno za n-1 jer je $n-1 = b^k$ a to je dokazano u prethodnom delu dokaza.
Dakle, prema principu regresivne indukcije teorema je tačna za svako $n > 1$.

Ako vršimo izračunavanja sa brojevima u pokretnom zarezu sa mantisom fiksne dužine zbog neprestanog svođenja rezultata na dužinu mantise, zaključujemo da se ustvari dešava sledeći proces : dobili smo rezultate na n cifara , zatim smo te rezultate zaokružili na k ($k < n$) cifara i sa tako dobijenim rezultatima nastavljamo proces izračunavanja. Zbog toga se prirodno postavlja sledeće pitanje: na kojoj cifri rezultata pri računanju sa 1 cifara postoji razlika u odnosu na rezultat pri računanju na n cifara.
Ova pitanja ćemo razmotriti u sledećih nekoliko teorema.

Definicija 3.1.3 Neka su dati $x, x' \in S(b,n,e_1,e_2)$ dva broja u pokretnom zarezu sa mantisom dužine n. Broj x' se razlikuje u odnosu na x za b_1 jedinica na l-tom mestu mantise $(1 \leqslant l \leqslant n)$ ako je

$$|x - x'| = 0{,}b_1 \ldots b_{n-l+1} \cdot b^{ex-l+1}$$

gde je $ex = \exp(x)$ i $0 < b_1 < b$.

Definicija 3.1.4 Neka su dati $x, y \in S(b,n,e_1,e_2)$ brojevi u pokretnom zarezu sa mantisom dužine n. Zbir $x + y$ je predstavljen mantisom dužine $ex - ey + n - p$, gde je $ex = \exp(x)$, $ey = \exp(y)$ i $ex \geqslant ey$. p se naziva pomeranje mantise $x + y$ i $-1 \leqslant p \leqslant n$.

Teorema 3.1.2 Neka su dati $x, y \in S(b,n,e_1,e_2)$ dva broja u pokretnom zarezu sa mantisom dužine n, pri čemu je $x \neq 0$, $y \neq 0$. Označimo sa x_1, y_1 brojeve koji se dobijaju od x i y uzimanjem prvih $l(1 < l < n)$ cifara mantise brojeva x,y. Tada se tačan zbir $x+y$ i približan zbir x_1+y_1 razlikuju najviše na

1° $l - p + 1$ cifri najviše za $b-1$ ako je $p \geqslant 0$ i x,y su različitoga znaka,

2° $l + 1$ cifri najviše za 1 ako je $p = -1$,

3° l-toj cifri najviše za 1 ako je $p = 0$ i x, y su istoga znaka.

Dokaz. Prema pretpostavci x i y možemo predstaviti u obliku

$$x = mx_1 \cdot b^{ex} + mx_2 \cdot b^{ex-l}$$

$$y = my_1 \cdot b^{ey} + my_2 \cdot b^{ey-l}$$

gde je

$$b^{-1} \leqslant mx_1 \,;\, my_1 \leqslant 1 - b^{-l}$$

$$0 \leqslant mx_2 \,;\, my_2 \leqslant 1 - b^{-(n-l)}.$$

Neka je $ex \geqslant ey$ i predstavimo zbir u obliku

$$x + y = mx_1 \cdot b^{ex} + my_1 \cdot b^{ey} + mx_2 \cdot b^{ex-l} + my_2 \cdot b^{ey-l} =$$

$$mx_1 \cdot b^{ex} + my_1 \cdot b^{-(ex-ey)} \cdot b^{ex} + mx_2 \cdot b^{ex-l} + my_2 \cdot b^{-(ex-ey)} \cdot b^{ex-l} =$$

$$(mx_1 + my_1 \cdot b^{-(ex-ey)}) \cdot b^{ex} + (mx_2 + my_2 \cdot b^{-(ex-ey)}) \cdot b^{ex-l} .$$

Ako su x,y istoga znaka prema intervalnoj aritmetici sledi procena $mx_2 + my_2 \cdot b^{-(ex-ey)}$

$$\left[0{,}1 - b^{-(n-l)}\right] + \left[0{,}1 - b^{-(n-l)}\right] \cdot b^{-(ex-ey)} = \left[0{,}1 - b^{-(n-l)}\right] +$$

$$\left[0{,}b^{-(ex-ey)} - b^{-(n-l)-(ex-ey)}\right] = \left[0{,}1 + b^{-(ex-ey)} - b^{-(n-l)} - \right.$$

$$- b^{-(n-\ell)-(ex-ey)} \Big] \subset \Big[0,2).$$

Kako su x i y istoga znaka može da bude p = -1 ili p = 0. Ako je p = -1 tada dolazi do povećavanja eksponenta u izrazu

$(mx_1 + my_1 \cdot b^{-(ex-ey)}) \cdot b^{ex}$ za 1, a na osnovu intervalne ocene za

$(mx_2 + my_2 \cdot b^{-(ex-ey)})$ sledi da je razlika eksponenata

$$ex + 1 - (ex - l + 1) = l$$

U ovom slučaju se tačan i približan zbir razlikuju najviše za 1 na l+1 cifri mantise.

Ako je p = 0 tada ne dolazi do pomeranja mantise, pa je razlika eksponenata

$$ex - (ex - l + 1) = l - 1.$$

U ovom slučaju se tačan i približan zbir razlikuju najviše za 1 na l-tom mestu mantise.

Ako su x i y različitoga znaka neka je x > 0 a y < 0 tada je

procena $mx_2 + my_2 \cdot b^{-(ex-ey)}$ sledeća

$$\Big[0,1 - b^{-(n-\ell)}\Big] - \Big[0,1 - b^{-(n-\ell)}\Big] b^{-(ex-ey)} = \Big[0,1 - b^{-(n-\ell)}\Big] + \Big[0, b^{-(ex-ey)} - b^{-(n-\ell)-(ex-ey)}\Big] = \Big[-b^{-(ex-ey)} + b^{-(n-\ell)-(ex-ey)}, -(1 - b^{-(n-\ell)})\Big]$$

Ako je p = 0 tada nema pomeranja mantise a na osnovu procene

$(mx_2 + my_2 \cdot b^{-(ex-ey)})$ sledi da je razlika eksponenata

$$ex - (ex - l) = l$$

pa se tačan i približan zbir razlikuju najviše za b-1 na l+1 mestu mantise.

Ako je p > 0 tada je razlika eksponenata

$$ex - p - (ex - l) = l - p,$$

pa se tačan i približan zbir razlikuju najviše za b-1 na l - p + 1 cifri.

Na osnovu prethodnih razmatranja sledi tvrđenje teoreme.

Razmotrimo sličan problem za slučaj množenja.

Teorema 3.1.3 Neka su dati $x, y \in S(b,n,e_1,e_2)$ dva broja u pokretnom zarezu sa mantisom dužine n, pri čemu je $x \neq 0$ i $y \neq 0$. Označimo sa x_1, y_1 brojeve koji se dobijaju od x i y uzimanjem prvih $l\,(1 < l < n)$ cifara mantise brojeva x, y.

Tada se tačan proizvod $x \cdot y$ razlikuje od približnog $x_1 \cdot y_1$ najviše za 1 na l-1 mestu mantise.

Dokaz. Prema pretpostavci x i y možemo predstaviti u obliku

$$x = mx_1 \cdot b^{ex} + mx_2 \cdot b^{ex-l}$$

$$y = my_1 \cdot b^{ey} + my_2 \cdot b^{ey-l}$$

gde je

$$b^{-1} \leqslant mx_1 ; my_1 \leqslant 1 - b^{-l}$$

$$0 \leqslant mx_2 ; my_2 \leqslant 1 - b^{-(n-l)} .$$

Proizvod $x \cdot y$ možemo predstaviti u obliku

$$x \cdot y = mx_1 \cdot my_1 \cdot b^{ex+ey} + (mx_1 \cdot my_2 + mx_2 \cdot my_1) \cdot b^{ex-ey-l} + mx_2 \cdot my_2 \cdot b^{ex+ey-2l} .$$

Zanemarujući član sa eksponentom $ex + ey - 2l$ proizvod možemo napisati u obliku

$$x \cdot y \doteq mx_1 \cdot my_1 \cdot b^{ex+ey} + mx_1 \cdot my_1 \left(\frac{mx_2}{mx_1} + \frac{my_2}{my_1}\right) \cdot b^{ex+ey-l} .$$

Da bi razlika eksponenata prvog i drugog člana bila što manja treba uzeti da je

$$mx_1 \cdot my_1 = b^{-1} \cdot (1 - b^{-l})$$

Odredimo sada najveću vrednost izraza

$$f(mx_1, my_1) = mx_1 \cdot my_2 + mx_2 \cdot my_1$$

smatrajući mx_2 i my_2 konstantama koje su jednake najvećim vrednostima

$$mx_2 = my_2 = 1 - b^{-(n-l)} .$$

Sada je

$$f(mx_1, my_1) = (1 - b^{-(n-l)})(mx_1 + my_1) .$$

Lako se dokazuje da $f(mx_1, my_1)$ uz uslov $mx_1 \cdot my_1 = b^{-1}(1 - b^{-l})$ ima lokalni minimum unutar oblasti

$$b^{-1} \leqslant mx_1 ; my_1 \leqslant 1 - b^{-l} .$$

Odredimo sada najveću vrednost koeficijenta drugog člana na osnovu prethodnog razmatranja

$$mx_1 \cdot my_2 + mx_2 \cdot my_1 = (1 - b^{-(n-l)})(b^{-1} + 1 - b^{-l}) = 0,10\ldots b^{1}$$

Razlika eksponenata prvog i drugog člana u izrazu za $x \cdot y$ je

$$ex + ay - 1 - (ex + ey - 1 + 1) = 1 - 2.$$

Dakle , tačan proizvod $x \cdot y$ i približan proizvod $x_1 \cdot y_1$ se razlikuju najviše za 1 na $l-1$ mestu mantise.

Navodimo sada teoremu za slučaj deljenja.

Teorema 3.1.4 Neka su dati $x, y \in S(b,n,e_1,e_2)$ dva broja u pokretnom zarezu sa mantisom dužine n, pri čemu je $x \neq 0$ i $y \neq 0$. Označimo sa x_1, y_1 brojeve koji se dobijaju od x i y uzimanjem prvih l $(1 < l < n)$ cifara mantise brojeva x, y. Tada se tačan količnik x/y razlikuje od približnog količnika x_1/y_1 najviše za $b-1$ jedinica na l-tom mestu mantise.

Dokaz. Prema predpostavci x i y možemo predstaviti u obliku

$$x = mx_1 \cdot b^{e_x} + mx_2 \cdot b^{e_x - l}$$

$$y = my_1 \cdot b^{e_y} + my_2 \cdot b^{e_y - l}$$

gde je

$$b^{-1} \leqslant mx_1 ;\ my_1 \leqslant 1 - b^{-l}$$

$$0 \leqslant mx_2 ;\ my_2 \leqslant 1 - b^{-(n-l)} \quad .$$

Količnik možemo predstaviti u obliku

$$x/y = (mx_1 \cdot b^{e_x} + mx_2 \cdot b^{e_x-l})/(my_1 \cdot b^{e_y} + my_2 \cdot b^{e_y-l}) =$$

$$(\tfrac{mx_1}{my_1} \cdot b^{e_x-e_y} + \tfrac{mx_2}{my_1} \cdot b^{e_x-e_y-l})(1 + \tfrac{my_2}{my_1} \cdot b^{-l}) =$$

$$(\tfrac{mx_1}{my_1} \cdot b^{e_x-e_y} + \tfrac{mx_2}{my_1} \cdot b^{e_x-e_y-l})(1 - \tfrac{my_2}{my_1} \cdot b^{-l} + (\tfrac{my_2}{my_1} \cdot b^{-l})^2 + \dots)$$

Zanemarujući članove stepena manjeg od $-l$ dobijamo

$$x/y \doteq (\tfrac{mx_1}{my_1} \cdot b^{e_x-e_y} + \tfrac{mx_2}{my_1} \cdot b^{e_x-e_y-l})(1 - \tfrac{my_2}{my_1} \cdot b^{-l}) =$$

$$\tfrac{mx_1}{my_1} \cdot b^{e_x-e_y} + (\tfrac{mx_2}{my_1} - \tfrac{mx_1}{my_1} \cdot \tfrac{my_2}{my_1}) \cdot b^{e_x-e_y-l} - \tfrac{mx_2}{my_1} \cdot \tfrac{my_2}{my_1} \cdot b^{e_x-e_y-2l}$$

Kako je treći član poslednjeg zbira zanemariv u odnosu na prva dva, količnik možemo napisati u obliku

$$x/y \doteq \tfrac{mx_1}{my_1} \cdot b^{e_x-e_y} + \tfrac{mx_1}{my_1} \cdot (\tfrac{mx_2}{mx_1} - \tfrac{my_2}{my_1}) \cdot b^{e_x-e_y-l} \quad .$$

Zbog

$$\frac{1}{b(1-b^{-l})} \leqslant \frac{mx_1}{my_1} \leqslant b(1 - b^{-l})$$

treba uzeti mx_1 i my_1 tako da bude

$$\left|\frac{mx_1}{my_1}\right| < 1$$

da bi razlika eksponenata prvog i drugog člana u izrazu za količnik bila što manja.

Koeficijent $\frac{mx_1}{my_1} \cdot (\frac{mx_2}{mx_1} - \frac{my_2}{my_1})$ treba da ima maksimalnu vrednost a to se postiže ako je

$$my_2 = 0 \;,\; mx_2 = 1 - b^{-(n-l)} \;,\; my_1 = b^{-1} + b^{-l} \;,\; mx_1 = b^{-1}$$

jer je $\left|\frac{mx_1}{my_1}\right| < 1$.

Za napred pretpostavljene vrednosti promenljivih vrednost količnika je

$$x/y \doteq \frac{1}{1 + b^{-(l-1)}} \cdot b^{ex-ey} + \frac{1 - b^{-(n-l)}}{1 + b^{-(l-1)}} \cdot b^{ex-ey-l+1} .$$

Razvijajući $\frac{1}{1 + b^{-(l-1)}}$ zbog $l > 1$ imamo

$$\frac{1}{1 + b^{-(l-1)}} = 1 - b^{-(l-1)} + b^{-2(l-1)} - b^{-3(l-1)} + \dots$$

pa x/y možemo predstaviti u obliku

$$x/y \doteq (1 - b^{-(l-1)})b^{ex-ey} + b^{ex-ey-l+1}(1 - b^{-(n-l)} + b^{-(n-l)-(l-1)} -).$$

Razlika eksponenata iznosi

$$(ex - ey) - (ex - ey - l + 1) = l - 1.$$

Kako je koeficijent uz $b^{ex-ey-l+1}$ manji od 1 zaključujemo da se tačan količnik x/y i približan količnik x_1/y_1 razlikuju na l-tom mestu za najviše $b-1$.

U procesu izračunavanja argumenti aritmetičkih operacija obično nemaju jednak broj tačnih cifara pa zbog toga navodimo sledeće teorema.

Teorema 3.1.5 Neka su dati $x, x_1, y, y_1 \in S(b,n,e_1,e_2)$, $b \geq 2$, brojevi u pokretnom zarezu sa mantisom dužine n. Neka se x_1 razlikuje u odnosu na x na l_1-tom mestu ($l_1 > 2$) za b_1 jedinica, a y_1 u odnosu na y na l_2-tom mestu ($l_2 > 2$) za b_2 jedinica, pri čemu je $l_1 \leqslant l_2$. Tada se tačan zbir $x+y$ razlikuje od približnog zbira x_1+y_1 za najviše

1° 1 na l_1-p-1 mestu ako je $ex-ey-l_1+l_2 = 0$ i $b_1 + b_2 + 1 \geqslant b$
ili ako je $ex-ey-l_1+l_2 > 0$ i $b_1 + 1 \geqslant b$,

2° $b_1 + b_2 + 1$ na l_1-p mestu ako je $ex-ey-l_1+l_2 = 0$ i $b_1 + b_2 + 1 < b$,

3° $b_1 + 1$ na l_1-p mestu ako je $ex-ey-l_1+l_2 > 0$ i $b_1 + 1 < b$,

gde je p pomeranje mantise i $0 \leqslant p < l_1$.

Dokaz. Prema definiciji 3.1.3 x i y možemo predstaviti

$$x = mx_1 \cdot b^{ex} + mx_2 \cdot b^{ex-l_1+1}$$

$$y = my_1 \cdot b^{ey} + my_2 \cdot b^{ey-l_2+1}$$

gde je

$$b^{-1} \leq mx_1 \leq 1 - b^{-(\ell_1 - 1)} \quad ; \quad 0,b_1 \leq mx_2 \leq 0,b_1 + b^{-1} - b^{-(n-\ell_1+1)}$$

$$b^{-1} \leq my_1 \leq 1 - b^{-(\ell_2 - 1)} \quad ; \quad 0,b_2 \leq my_2 \leq 0,b_2 + b^{-1} - b^{-(n-\ell_2+1)} .$$

Neka je $ex \geq ey$ tada je

$$x+y = (mx_1 + my_1 \cdot b^{-(ex-ey)}) \cdot b^{ex} + (mx_2 + my_2 \cdot b^{-(ex-ey-\ell_1+\ell_2)}) \cdot b^{ex-\ell_1+1} .$$

Neka je $\exp(mx_1 + my_1 \cdot b^{-(ex-ey)}) = -p$ tada je eksponent prvog sabirka $ex - p$.

Neka je $ex-ey-l_1+l_2 = 0$. Ako je $b_1 + b_2 + 1 \geq b$ i $b > 2$ tada je eksponent drugog sabirka $ex-l_1+2$ pa je razlika eksponenata

$$ex - p -(ex - l_1 + 2) = l_1 - p - 2.$$

Dakle, tačan i približan zbir se razlikuju najviše za 1 na $l_1 - p - 1$ mestu mantise.
Ako je $b_1 + b_2 + 1 < b$ i $b > 2$ tada je razlika eksponenata

$$ex - p - (ex - l_1 + 1) = l_1 - p - 1,$$

pa se tačan i približan zbir razlikuju najviše za $b_1 + b_2 + 1$ na $l_1 - p$ mestu mantise.
Neka je $ex - ey - l_1 + l_2 > 0$. Ako je $b_1 + 1 \geq b$ tada je eksponent drugog sabirka u zbiru $ex - l_1 + 2$, pa je razlika eksponenata $l_1 - p - 2$, i tada se tačan i približan zbir razlikuju najviše za 1 na $l_1 - p - 1$ mestu.
Ako je $b_1 + 1 < b$ tada je eksponent drugog sabirka $ex - l_1 + 1$, pa je razlika eksponenata $l_1 - p - 1$, i tada se tačan i približan zbir razlikuju najviše za $b_1 + 1$ na $l_1 - p$ mestu.'

Napomena 3.1.2 Ako je $b = 2$, $ex - ey - l_1 + l_2 = 0$ i $b_1 + b_2 + 1 \geq b$, tada se tačan i približan zbir razlikuju najviše za 1 na $l_1 - p - 2$ mestu mantise.

Teorema 3.1.6 Neka su dati $x, x_1, y, y_1 \in S(b,n,e_1,e_2)$ brojevi u pokretnom zarezu sa mantisom dužine n. Neka se x_1 razlikuje u odnosu na x na l_1-tom mestu ($l_1 > 2$) za b_1 jedinica, a y_1 u odnosu na y na l_2-tom mestu ($l_2 > 2$) za b_2 jedinica, pri čemu je $l_1 \leq l_2$. Tada se tačan proizvod $x \cdot y$ razlikuje od približnog proizvoda $x_1 \cdot y_1$ najviše za

1° 1 na $l_1 - 2$ mestu mantise ako je $b_1 + 1 \geq b$,

2° $b_1 + 1$ na $l_1 - 1$ mestu mantise ako je $b_1 + 1 < b$.

Dokaz. Prema definiciji 3.1.3 x i y možemo predstaviti

$$x = mx_1 \cdot b^{ex} + mx_2 \cdot b^{ex-\ell_1+1}$$

$$y = my_1 \cdot b^{ey} + my_2 \cdot b^{ey-\ell_2+1}$$

gde je

$$b^{-1} \leqslant mx_1 \leqslant 1 - b^{-(l_1-1)} \quad ; \quad 0,b_1 \leqslant mx_2 \leqslant 0,b_1 + b^{-1} - b^{-(n-l_1+1)}$$

$$b^{-1} \leqslant my_1 \leqslant 1 - b^{-(l_2-1)} \quad ; \quad 0,b_2 \leqslant my_2 \leqslant 0,b_2 + b^{-1} - b^{-(n-l_2+1)} .$$

Proizvod $x \cdot y$ možemo predstaviti u obliku

$$x \cdot y = mx_1 \cdot my_1 \cdot b^{ex+ey} + (mx_2 \cdot my_1 + mx_1 \cdot my_2 \cdot b^{l_1-l_2}) \cdot b^{ex-ey-l_1+1}$$
$$+ mx_2 \cdot my_2 \cdot b^{ex+ey-(l_1+l_2)+2} .$$

Ako zanemarimo treći sabirak u izrazu za proizvod , najmanja razlika u eksponentima se postiže ako je

$$mx_1 \cdot my_1 = b^{-1}(1 - b^{-(l_2-1)}) .$$

Slično rasuđivanjima kao u teoremi 3.1.3 zaključujemo da se najveća vrednost koeficijenta uz $b^{ex+ey-l_1+1}$ postiže za

$$mx_1 = b^{-1} , \; mx_2 = 0,b_1 + b^{-1} - b^{-(n-l_1+1)} ,$$
$$my_1 = 1 - b^{-(l_2-1)} , \; my_2 = 0,b_2 + b^{-1} - b^{-(n-l_2+1)} .$$

Za ove vrednosti promenljivih vrednost koeficijenta uz $b^{ex+ey-l_1+1}$ je

$$mx_2 \cdot my_1 + mx_1 \cdot my_2 \cdot b^{l_1-l_2} =$$
$$(0,b_1 + b^{-1} - b^{-(n-l_1+1)}) \cdot (1 - b^{-(l_2-1)}) + b^{-1} \cdot (0,b_2 + b^{-1} - b^{-(n-l_2+1)}) \cdot b^{l_1-l_2}$$
$$= 0,(b_1+1)\ldots \quad .$$

Ako je $b_1 + 1 \geqslant b$ tada je razlika eksponenata prvog i drugog člana u izrazu za proizvod

$$ex + ey - 1 - (ex + ey - l_1 + 2) = l_1 - 3 .$$

Dakle , u ovom slučaju je razlika na $l_1 - 2$ mestu mantise najviše za 1 .
Ako je $b_1 + 1 < b$ tada je razlika eksponenata

$$ex + ey - 1 - (ex + ey - l_1 + 1) = l_1 - 2 .$$

Dakle, razlika je na $l_1 - 1$ mestu najviše za $b_1 + 1$.

Teorema 3.1.7 Neka su dati $x, x_1, y, y_1 \in S(b,n,e_1,e_2)$ i $b > 2$, brojevi u pokretnom zarezu sa mantisom dužine n. Neka se x_1 razlikuje u odnosu na x na l_1-tom mestu ($l_1 > 2$) za b_1 jedinica, a y_1 u odnosu na y na l_2-tom mestu ($l_2 > 2$) za b_2 jedinica. Tada se tačan količnik x/y razlikuje od približnog količnika x_1/y_1

1° ako je $l_1 = l_2$ za najviše 1 na $l_1 - 2$ mestu mantise ako je $b_1 + b_2 + 1 \geqslant b$ ili za najviše $b_1 + b_2 + 1$ na $l_1 - 1$ mestu mantise ako je $b_1 + b_2 + 1 < b$,

2° ako je $l_1 < l_2$ za najviše 1 na $l_1 - 2$ mestu mantise ako je $b_1 + 1 \geqslant b$ ili za najviše $b_1 + 1$ na $l_1 - 1$ mestu mantise ako je $b_1 + 1 < b$,

3° ako je $l_1 > l_2$ za najviše 1 na $l_2 - 2$ mestu mantise ako je $b_2 + 1 \geqslant b$ ili za najviše $b_2 + 1$ na $l_2 - 1$ mestu ako je $b_2 + 1 < b$.

Dokaz. Prema definiciji 3.1.3 i da bi razlika između tačnog i približnog količnika bila najveća treba da je

$$x = mx_1 \cdot b^{ex} + mx_2 \cdot b^{ex - \ell_1 + 1}$$

$$y = my_1 \cdot b^{ey} - my_2 \cdot b^{ey - \ell_2 + 1}$$

gde je

$$b^{-1} \leqslant mx_1 \leqslant 1 - b^{-(\ell_1 - 1)} \quad ; \quad 0{,}b_1 \leqslant mx_2 \leqslant 0{,}b_1 + b^{-1} - b^{-(n - \ell_1 + 1)}$$

$$b^{-1} \leqslant my_1 \leqslant 1 - b^{-(\ell_2 - 1)} \quad ; \quad 0{,}b_2 \leqslant my_2 \leqslant 0{,}b_2 + b^{-1} - b^{-(n - \ell_2 + 1)} \quad .$$

Količnik x/y možemo predstaviti u obliku

$$x/y = (mx_1 \cdot b^{ex} + mx_2 \cdot b^{ex - \ell_1 + 1})/(my_1 \cdot b^{ey} - my_2 \cdot b^{ey - \ell_2 + 1}) =$$

$$(\frac{mx_1}{my_1} \cdot b^{ex - ey} + \frac{mx_2}{my_1} \cdot b^{ex - ey - \ell_1 + 1})/(1 - \frac{my_2}{my_1} \cdot b^{-(\ell_2 + 1)}) =$$

$$(\frac{mx_1}{my_1} \cdot b^{ex - ey} + \frac{mx_2}{my_1} \cdot b^{ex - ey - \ell_1 + 1}) \cdot (1 + \frac{my_2}{my_1} \cdot b^{-(\ell_2 - 1)} - \ldots).$$

Zanemarujući članove stepena manjeg od $1 - l_2$ dobijamo

$$x/y \doteq (\frac{mx_1}{my_1} \cdot b^{ex - ey} + \frac{mx_2}{my_1} \cdot b^{ex - ey - \ell_1 + 1})\ (1 + \frac{my_2}{my_1} \cdot b^{-(\ell_2 - 1)}) =$$

$$\frac{mx_1}{my_1} \cdot b^{ex - ey} + \frac{mx_1}{my_1}(\frac{mx_2}{mx_1} + \frac{my_2}{my_1} \cdot b^{\ell_1 - \ell_2}) \cdot b^{ex - ey - \ell_1 + 1} +$$

$$+ \frac{mx_2 \cdot my_2}{my_1 \cdot my_1} \cdot b^{ex - ey - (\ell_1 + \ell_2) + 2} .$$

Ako zanemarimo poslednji član količnik možemo zapisati

$$x/y \doteq \frac{mx_1}{my_1} \cdot b^{ex - ey} + \frac{mx_1}{my_1} \cdot (\frac{mx_2}{mx_1} + \frac{my_2}{my_1} \cdot b^{\ell_1 - \ell_2}) \cdot b^{ex - ey - \ell_1 + 1} .$$

Da bi razlika eksponenata prvog i drugog člana bila najmanja mora biti

$$\left|\frac{mx_1}{my_1}\right| < 1.$$

Koeficijent $\frac{mx_1}{my_1}(\frac{mx_2}{mx_1} + \frac{my_2}{my_1} \cdot b^{\ell_1 - \ell_2})$ postiže maksimalnu vrednost ako je

$$mx_1 = b^{-1} \; , \; mx_2 = 0{,}b_1 + b^{-1} - b^{-(n - \ell_1 + 1)} \; , \; my_2 = 0{,}b_2 + b^{-1} - b^{-(n - \ell_2 + 1)}.$$

Zbog $\left|\frac{mx_1}{my_1}\right| < 1$ i maksimalne vrednosti koeficijenta uz $b^{ex-ey-l_1+1}$ treba uzeti da je $my_1 = b^{-1} + b^{-(l_2+1)}$.

Za navedene vrednosti promenljivih je

$$\frac{mx_1}{my_1}\left(\frac{mx_2}{mx_1} + \frac{my_2}{my_1}\cdot b^{l_1-l_2}\right) = b\left(\frac{0,b_1+b^{-1}-b^{-(n-l_1+1)}}{1+b^{-(l_2-2)}} + \frac{0,b_2+b^{-1}-b^{-(n-l_2+1)}}{(1+b^{-(l_2-2)})^2}\, b^{l_1-l_2}\right).$$

Zbog aproksimacija

$$\frac{1}{1+b^{-(l_2-2)}} \doteq 1 - b^{-(l_2-2)} \quad \text{i} \quad \frac{1}{(1+b^{-(l_2-2)})^2} \doteq 1 - 2\cdot b^{-(l_2-2)}$$

dobijamo

$$\frac{mx_1}{my_1}\cdot\left(\frac{mx_2}{my_1} + \frac{my_2}{my_1}\cdot b^{l_1-l_2}\right) \doteq b\cdot((0,b_1 + b^{-1} - b^{-(n-l_1+1)})(1 - b^{-(l_2-2)})$$
$$+ (0,b_2 + b^{-1} - b^{-(n-l_2+1)})(1 - 2\cdot b^{-(l_2-2)})\cdot b^{l_1-l_2}.$$

Ako je $l_1 = l_2$ sledi

$$\frac{mx_1}{my_1}\cdot\left(\frac{mx_2}{mx_1} + \frac{my_2}{my_1}\right) = b(0,b_1 + 0,b_2 + 2(b^{-1} - b^{-(n-l_1+1)}) -$$
$$- b^{-(l_1-2)}(0,b_1 + 2\cdot 0,b_2 + 3\cdot b^{-1} - 3\cdot b^{-(n-l_1+1)})).$$

Ako je $b_1 + b_2 + 1 \geqslant b$ i $b > 2$ tada je razlika eksponenata

$$ex - ey - (ex - ey - l_1 + 3) = l_1 - 3$$

pa se tačan i približan količnik razlikuju za najviše 1 na $l_1 - 2$ mestu mantise.

Ako je $b_1 + b_2 + 1 < b$ tada je razlika eksponenata

$$ex - ey - (ex - ey - l_1 + 2) = l_1 - 2$$

pa se tačan i približan količnik razlikuju najviše za $b_1 + b_2 + 1$ na $l_1 - 1$ mestu mantise.

Ako je $l_1 < l_2$ tada je

$$\frac{mx_1}{my_1}\cdot\left(\frac{mx_2}{mx_1} + \frac{my_2}{my_1}\cdot b^{l_1-l_2}\right) = b\cdot 0,(b_1 + 1)\ldots\,.$$

Ako je $b_1 + 1 \geqslant b$ tada je razlika eksponenata

$$ex - ey - (ex - ey - l_1 + 3) = l_1 - 3$$

pa se tačan i približan količnik razlikuju na $l_1 - 2$ mestu mantise najviše za 1.
Ako je $b_1 + 1 < b$ tada je razlika eksponenata

$$ex - ey - (ex - ey - l_1 + 2) = l_1 - 2$$

pa se tačan i približan količnik razlikuju na $l_1 - 1$ mestu najviše za $b_1 + 1$.

Ako je $l_1 > l_2$ tada količnik napišimo u obliku

$$x/y = \frac{mx_1}{my_1} \cdot b^{e_x - e_y} + \frac{mx_1}{my_1} \cdot \left(\frac{mx_2}{mx_1} \cdot b^{l_2 - l_1} + \frac{my_2}{my_1}\right) b^{e_x - e_y - l_2 + 1} .$$

Slično se dokazuje da je najveća vrednost izraza

$$\frac{mx_1}{my_1} \cdot \left(\frac{mx_2}{mx_1} \cdot b^{l_2 - l_1} + \frac{my_2}{my_1}\right) = b \cdot 0,(b_2 + 1)\ldots$$

pa deo tvrđenja 3° sledi slično kao 2°.

Napomena 3.1.3 Ako je $b = 2$, $l_1 = l_2$ i $b_1 + b_2 + 1 \geqslant b$, tada se tačan i približan količnik razlikuju najviše za 1 na $l_1 - 3$ mestu mantise.

4.1 Sistemi linearnih jednačina

Problemi koji mogu da se jave pri rešavanju sistema linearnih jednačina su prostorna , vremenska ograničenost i tačnost rešenja . Današnji računari raspolažu relativno velikom memorijom , a i velikom brzinom izračunavanja. U poslednje vreme sve se više koriste paralelna izračunavanja, tako da vremenska i prostorna ograničenost ne predstavljaju veliki problem.
Problem tačnosti može da nastupi iz razloga da koeficijenti nisu tačno predstavljivi : obično računamo u osnovi 10 a računari u osnovi koja je neki stepen od 2. Da bi se izbegao problem grešaka pri prevođenju brojeva u poslednje vreme se proizvode računari koji izvode izračunavanja u osnovi 10.
Problem tačnosti rešavanja kod sistema linearnih jednačina nastupa zbog zaokruživanja međurezultata na broj cifara kojim računa računar.Tako da imamo primera sistema linearnih jednačina koji su veoma osetljivi na zaokruživanja i nazivaju se slabo uslovljenim sistemima. Jedna klasa takvih sistema linearnih jednačina su sistemi sa Hilbert-ovom matricom koeficijenata.
Navodimo primer Hilbert-ove matrice 15x15 da bismo pokazali uticaj grešaka zaokruživanja na konačna rešenja. Matrica je pomnožena odgovarajućim faktorom, tako da bi koeficijenti bili celi brojevi, odnosno da bi se izbegao problem netačnog prevođenja koeficijenata.

0.116454478140D+13	0.582272390700D+12	0.388181593800D+12
0.291136195350D+12	0.232908956280D+12	0.194090796900D+12
0.166363540200D+12	0.145568097675D+12	0.129393864600D+12
0.116454478140D+12	0.105867707400D+12	0.970453984500D+11
0.895803678000D+11	0.831817701000D+11	0.776363187600D+11
0.100000000000D+01		
0.582272390700D+12	0.388181593800D+12	0.291136195350D+12
0.232908956280D+12	0.194090796900D+12	0.166363540200D+12
0.145568097675D+12	0.129393864600D+12	0.116454478140D+12
0.105867707400D+12	0.970453984500D+11	0.895803678000D+11
0.831817701000D+11	0.776363187600D+11	0.727840488375D+11
0.200000000000D+01		
0.388181593800D+12	0.291136195350D+12	0.232908956280D+12
0.194090796900D+12	0.166363540200D+12	0.145568097675D+12
0.129393864600D+12	0.116454478140D+12	0.105867707400D+12
0.970453984500D+11	0.895803678000D+11	0.831817701000D+11
0.776363187600D+11	0.727840488375D+11	0.685026342000D+11
0.300000000000D+01		
0.291136195350D+12	0.232908956280D+12	0.194090796900D+12
0.166363540200D+12	0.145568097675D+12	0.129393864600D+12
0.116454478140D+12	0.105867707400D+12	0.970453984500D+11
0.895803678000D+11	0.831817701000D+11	0.776363187600D+11
0.727840488375D+11	0.685026342000D+11	0.646969323000D+11
0.400000000000D+01		

0.232908956280D+12	0.194090796900D+12	0.166363540200D+12
0.145568097675D+12	0.129393864600D+12	0.116454478140D+12
0.105867707400D+12	0.970453984500D+11	0.895803678000D+11
0.831817701000D+11	0.776363187600D+11	0.727840488375D+11
0.685026342000D+11	0.646969323000D+11	0.612918306000D+11
0.500000000000D+01		
0.194090796900D+12	0.166363540200D+12	0.145568097675D+12
0.129393864600D+12	0.116454478140D+12	0.105867707400D+12
0.970453984500D+11	0.895803678000D+11	0.831817701000D+11
0.776363187600D+11	0.727840488375D+11	0.685026342000D+11
0.646969323000D+11	0.612918306000D+11	0.582272390700D+11
0.600000000000D+01		
0.166363540200D+12	0.145568097675D+12	0.129393864600D+12
0.116454478140D+12	0.105867707400D+12	0.970453984500D+11
0.895803678000D+11	0.831817701000D+11	0.776363187600D+11
0.727840488375D+11	0.685026342000D+11	0.646969323000D+11
0.612918306000D+11	0.582272390700D+11	0.554545134000D+11
0.700000000000D+01		
0.145568097675D+12	0.129393864600D+12	0.116454478140D+12
0.105867707400D+12	0.970453984500D+11	0.895803678000D+11
0.831817701000D+11	0.776363187600D+11	0.727840488375D+11
0.685026342000D+11	0.646969323000D+11	0.612918306000D+11
0.582272390700D+11	0.554545134000D+11	0.529338537000D+11
0.800000000000D+01		
0.129393864600D+12	0.116454478140D+12	0.105867707400D+12
0.970453984500D+11	0.895803678000D+11	0.831817701000D+11
0.776363187600D+11	0.727840488375D+11	0.685026342000D+11
0.646969323000D+11	0.612918306000D+11	0.582272390700D+11
0.554545134000D+11	0.529338537000D+11	0.506323818000D+11
0.700000000000D+01		
0.116454478140D+12	0.105867707400D+12	0.970453984500D+11
0.895803678000D+11	0.831817701000D+11	0.776363187600D+11
0.727840488375D+11	0.685026342000D+11	0.646969323000D+11
0.612918306000D+11	0.582272390700D+11	0.554545134000D+11
0.529338537000D+11	0.506323818000D+11	0.485226992250D+11
0.600000000000D+01		
0.105867707400D+12	0.970453984500D+11	0.895803678000D+11
0.831817701000D+11	0.776363187600D+11	0.727840488375D+11
0.685026342000D+11	0.646969323000D+11	0.612918306000D+11
0.582272390700D+11	0.554545134000D+11	0.529338537000D+11
0.506323818000D+11	0.485226992250D+11	0.465817912560D+11
0.500000000000D+01		
0.970453984500D+11	0.895803678000D+11	0.831817701000D+11
0.776363187600D+11	0.727840488375D+11	0.685026342000D+11
0.646969323000D+11	0.612918306000D+11	0.582272390700D+11
0.554545134000D+11	0.529338537000D+11	0.506323818000D+11
0.485226992250D+11	0.465817912560D+11	0.447901839000D+11
0.400000000000D+01		
0.895803678000D+11	0.831817701000D+11	0.776363187600D+11
0.727840488375D+11	0.685026342000D+11	0.646969323000D+11
0.612918306000D+11	0.582272390700D+11	0.554545134000D+11
0.529338537000D+11	0.506323818000D+11	0.485226992250D+11
0.465817912560D+11	0.447901839000D+11	0.431312882000D+11
0.300000000000D+01		

```
0.831817701000D+11  0.776363187600D+11  0.727840488375D+11
0.685026342000D+11  0.646969323000D+11  0.612918306000D+11
0.582272390700D+11  0.554545134000D+11  0.529338537000D+11
0.506323819000D+11  0.485226992250D+11  0.465817912560D+11
0.447901839000D+11  0.431312882000D+11  0.415908850500D+11
0.200000000000D+01
0.776363187600D+11  0.727840488375D+11  0.685026342000D+11
0.646969323000D+11  0.612918306000D+11  0.582272390700D+11
0.554545134000D+11  0.529338537000D+11  0.506323818000D+11
0.485226992250D+11  0.465817912560D+11  0.447901839000D+11
0.431312882000D+11  0.415908850500D+11  0.401567166000D+11
0.100000000000D+01
```

Svakih pet uzastopnih vrsta navedene tabele predstavljaju redom 15 koeficijenata a u šestoj vrsti je desna strana sistema linearnih jednačina . Elementi matrice su dati u dvostrukoj preciznosti na 12 cifara, jer obična preciznost nije dovoljna za njihovo predstavljanje na DIGITAL-ovim računarima. Najbolja aproksimacija, u smislu zaokruživanja, rešenja datog sistema na 12 dekadnih cifara glasi:

```
-0.578999254704D-02  0.114415175303D+01 -0.565419358901D+02
 0.122731367522D+04 -0.146321958521D+05  0.107653530435D+06
-0.523134225387D+06  0.174885780475D+07 -0.411439182701D+07
 0.686925162912D+07 -0.809380818008D+07  0.657904895545D+07
-0.351015512696D+07  0.110616242295D+07 -0.156024600400D+06.
```

Ako napred navedeni sistem rešimo potprogramom SIMQ u dvostrukoj tačnosti iz DIGITAL-ove SSP biblioteke na MINC-11 računaru sa RT-11 operacionim sistemom u FORTRAN-IV jeziku , koji je realizacija Gauss-ove metode za rešavanje sistema linearnih jednačina dobijamo sledeći rezultat:

```
-0.169747204372D-02  0.200397421667D+00 -0.356028397967D+01
-0.518603667463D+02  0.205273471579D+04 -0.249040597161D+05
 0.165202999030D+06 -0.689307148836D+06  0.192096790884D+07
-0.367165815366D+07  0.483554367507D+07 -0.431676248638D+07
 0.249554601034D+07 -0.843151282051D+06  0.126425030987D+06.
```

Ako isti sistem jednačina rešimo potprogramom LEQT2F u dvostrukoj tačnosti iz IMSL biblioteka verzija za VAX/VMS operacioni sistem na računaru VAX-11/750 dobijamo sledeći rezultat:

```
-0.154305372279D-02  0.253960004357D+00 -0.103836155221D+02
 0.184536088661D+03 -0.177328084252D+04  0.102736794134D+05
-0.379413092709D+05  0.909900910096D+05 -0.138159684741D+06
 0.116017342613D+06 -0.139183379510D+05 -0.797874446831D+05
 0.871116104185D+05 -0.403613463012D+05  0.737327634682D+04.
```

Potprogram LEQT2F sadrži takođe parametar IER čija vrednost daje informaciju o tome da li je sistem slabo uslovljen ili nije. Pri rešavanju napred navedenog sistema potprogram nije ukazao na slabu uslovljenost, iako je u slučaju Hilbert-ove matrice slaba uslovljenost dobro poznata. Pored toga, LEQT2F je deklarisan kao potprogram za rešavanje sa velikom preciznošću, iako u ovom slučaju on to svojstvo ne poseduje. Za potprogram SIMQ možemo staviti primedbu da je jednostavan i nije dugo menjan. Međutim potprogrami IMSL biblioteke se redovno zamenjuju novim verzijama. Potprogram LEQT2F koristi čak i potprograme VXADD I VXMUL za četvorostruku preciznost pri izračunavanju međurezultata.
Potprogrami SIMQ i LEQT2F su uzeti kao predstavnici jer i drugi potprogrami pomenutih biblioteka ne daju bolje rezultate za napred navađeni problem.

Dakle, direktni algoritmi za rešavanje sistema linearnih jednačina,sa gledišta tačnosti predstavljaju veliki problem. Za direktno rešavanje sistema linearnih jednačina obično se koristi Gauss-ov algoritam. Međutim, deljenje a_{i1} ili a_{1j} sa $a_{11} \neq 0$, ako nije konačno , dovodi do grešaka. Zbog toga se može da predloži varijanta Gauss-ovog algoritma bez deljenja:

Teorema 4.1 Neka je dat sistem linearnih jednačina matricom

$$\begin{matrix} a_{11} & a_{12} & \cdots & a_{1n} & a_{1n+1} \\ a_{21} & a_{22} & \cdots & a_{2n} & a_{2n+1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdots & a_{nn} & a_{nn+1} \end{matrix} .$$

Tada se dati sistem transformacijom

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} \cdot a_{kk}^{(k-1)} - a_{ik}^{(k-1)} a_{kj}^{(k-1)} \quad , \; k = 1, 2, \ldots , n-1, \; i,j > k$$

svodi na trougaoni sistem

$$\begin{matrix} a_{11} & a_{12} & \cdots & a_{1n} & a_{1n+1} \\ & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & a_{2n+1}^{(1)} \\ & & \cdot\;\cdot\;\cdot & & \\ & & & a_{nn}^{(n-1)} & a_{nn+1}^{(n-1)} \end{matrix}$$

gde je $a_{ij}^{(0)} = a_{ij}$.

Evidentno je da transformacija $a_{ij}^{(k)} = a_{ij}^{(k-1)} \cdot a_{kk}^{(k-1)} - a_{ik}^{(k-1)} \cdot a_{kj}^{(k-1)}$ može da izaziva prekoračenja intervala za eksponent. Da bi se to izbeglo dovoljno je $a_{ij}^{(k)}$ podeliti ili pomnožiti sa povoljno izabranim b^n, gde je n ceo broj a b osnova sistema. Ovo deljenje ili množenje(oduzimanje ili dodavanje n eksponentu)

dovodi do svođenja vrednosti eksponenta u određeni interval a ne dovodi do grešaka izračunavanja. Dakle, postupak je tačniji za realizaciju na računaru u aritmetici pokretnog zareza nego uobičajeni Gauss-ov algoritam.
Procena broja tačnih cifara se može izvršiti na osnovu teorema 3.1.5, 3.1.6 i 3.1.7.

4.2 Primena tačnog skalarnog proizvoda kod iterativnog rešavanja sistema linearnih jednačina

Da bismo primenili algoritme iz drugog dela za iterativno rešavanje sistema linearnih jednačina navedimo poznatu teoremu:

Teorema 4.2.1 Neka je dat sistem linearnih jednačina u obliku

$$
\begin{aligned}
x_1^{(k)} &= b_{11}\cdot x_1^{(k-1)} + b_{12}\cdot x_2^{(k-1)} + \ldots + b_{1n}\cdot x_n^{(k-1)} + \beta_1 \\
x_2^{(k)} &= b_{21}\cdot x_1^{(k-1)} + b_{22}\cdot x^{(k-1)} + \ldots + b_{2n}\cdot x^{(k-1)} + \beta_2 \\
&\cdots\cdots\cdots\cdots\cdots\cdots \\
x_n^{(k)} &= b_{n1}\cdot x_1^{(k-1)} + b_{n2}\cdot x_2^{(k-1)} + \ldots + b_{nn}\cdot x_n^{(k-1)} + \beta_n
\end{aligned}
$$

za $k = 1, 2, \ldots$, pri čemu je $\|B\| < 1$. Tada je

$$\|\vec{x}^{(k)} - \vec{x}\| \leqslant \frac{\|B\|}{1-\|B\|}\cdot\|\vec{x}^{(k)} - \vec{x}^{(k-1)}\| ,$$

gde je $\vec{x}$ rešenje a $\vec{x}^{(k)}$ njegova k-ta aproksimacija.

Dakle , sa gledišta tačnosti izračunavanja problem je izračunavanje desne strane jednakosti, odnosno skalarnog proizvoda.
Na osnovu algoritama tačnog sumiranja i tačnog proizvoda, skalarni proizvod možemo tačno izračunati i predstaviti ga u RN-preciznosti. Tako da posedujući proširenu preciznost kao što je RN-preciznost, možemo tvrditi da ako rešavamo sistem linearnih jednačina iterativnom metodom, možemo ga rešiti sa proizvoljnom tačnošću.
Pored toga tačan skalarni proizvod omogućava proveru valjanosti rešenja,što u običnoj aritmetici pokretnog zareza nije uvek moguće.

4.3 Tačno rešavanje sistema linearnih jednačina

Nekada priroda problema ne dozvoljava aproksimacije, te je sistem linearnih jednačina potrebno tačno rešiti. U poslednje vreme imamo veliki broj radova koji se bavi ovim problemom : /3/, /10/, /11/, /19/, /21/, /22/, /53/ i /54/. U navedenim radovima ovaj problem se rešava izračunavanjem determinanti ili nalaženjem adjungovane matrice pa zatim množenjem sa vektorom na desnoj strani.
Međutim, čim je veća dimenzija problema, tačno izračunavanje determinante zahteva velike brojeve, što prelazi mogućnosti predstavljanja brojeva bilo kog danas poznatog računara.
Koeficijenti sistema linearnih jednačina mogu biti celi ili brojevi u pokretnom zarezu. Ako su koeficijenti celobrojni tada se za rešavanje sistema koristi modularna aritmetika i simetrično predstavljanje brojeva sa mešovitom osnovom.
U modularnoj aritmetici teškoću predstavlja razlikovanje pozitivnih i negativnih brojeva kao i problemi prekoračenja i deljenja.
Međutim , operacije sabiranja, oduzimanja i množenja se jednostavno izvode a pored toga izvršavanje množenja je reda n a ne n^2 kao kod sistema sa fiksnom osnovom. Osim toga operacije u modularnoj aritmetici mogu se izvršavati paralelno.
Zbog toga ćemo ukratko izložiti način rešavanja sistema linearnih jednačina u modularnoj aritmetici.

Definicija 4.3.1 Ako su a i b celi brojevi i ako $m \neq 0$ deli $a - b$ tada zapisujemo

$$a \equiv b \bmod m \quad .$$

Ceo broj m se naziva modul kongruencije. Kako m deli $a - b$ ako i samo ako $-m$ deli $a - b$ pretpostavićemo da je m pozitivan i da je $m > 1$.

Definicija 4.3.2 Neka je x ceo broj a m modul. Ako je

$$r \equiv x \bmod m$$

i $0 \leqslant r < m$ tada pišemo

$$r = |x|_m$$

i kažemo da je r ostatak od x po modulu m.

Definicija 4.3.3 Neka su $m_1, m_2, \ldots, m_r$ uzajamno prosti moduli . Predstavljanje broja x pomoću ostataka u osnovi $\beta = (m_1, m_2, \ldots, m_r)$ je r-torka

$$x_\beta = (|x|_{m_1}, |x|_{m_2}, \ldots, |x|_{m_r}).$$

Operacije sabiranja , oduzimanja i množenja nad brojevima predstavljenim pomoću ostataka definišemo na sledeći način :

Definicija 4.3.4

$$(a_1, a_2, \ldots, a_r) + (b_1, b_2, \ldots, b_r) =$$

$$((a_1 + b_1)\bmod m_1, \ldots, (a_r + b_r)\bmod m_r),$$

$$(a_1, a_2, \ldots, a_r) - (b_1, b_2, \ldots, b_r) =$$

$$((a_1 - b_1)\bmod m_1, \ldots, (a_r - b_r)\bmod m_r),$$

$$(a_1, a_2, \ldots, a_r) \cdot (b_1, b_2, \ldots, b_r) =$$

$$((a_1 \cdot b_1)\bmod m_1, \ldots, (a_r \cdot b_r)\bmod m_r).$$

Inverziju za množenje definišemo:

Definicija 4.3.5 Ako je m prost broj i neka je $b \neq 0$ ceo broj, tada postoji jedinstven broj c takav da je

$$|c \cdot b|_m = |b \cdot c|_m = 1.$$

Broj c nazivamo inverznim elementom u odnosu na množenje po modulu m i označavamo ga $c = b^{-1}(m)$.
Inverzan elemenat za x za sistem modula $m_1, m_2, \ldots, m_r$ definišemo:

Definicija 4.3.6 Neka je data osnova $\beta = (m_1, m_2, \ldots, m_r)$ i neka $x^{-1}(m_1), x^{-1}(m_2), \ldots, x^{-1}(m_r)$ postoje. Tada je inverzan elemenat za x u odnosu na β

$$x^{-1}(\beta) = (x^{-1}(m_1), x^{-1}(m_2), \ldots, x^{-1}(m_r)).$$

Definišimo sada simetrično predstavljanje brojeva po modulu m .

Definicija 4.3.7 Neka je x ceo broj i m modul. Ako je

$$r = x \bmod m$$

i ako je

$$-\tfrac{1}{2}\cdot m < r < \tfrac{1}{2}\cdot m$$

kažemo da je r simetričan ostatak po modulu m i označavamo ga sa $r = /x/_m$.

Definicija 4.3.8 Simetrično predstavljanje broja x pomoću ostataka u sistemu $\beta = (m_1, m_2, \ldots, m_r)$ je r-torka

$$/x/_\beta = (/x/_{m_1}, /x/_{m_2}, \ldots, /x/_{m_r}).$$

Definicija 4.3.9 Neka su dati celi brojevi $r_1, r_2, \ldots, r_t$ koje nazivamo osnove. Tada svaki ceo broj x možemo prikazati u obliku

$$x = c_1 + c_2 r_1 + c_3 r_1 r_2 + \ldots + c_t r_1 r_2 \ldots r_{t-1},$$

gde su $c_1, c_2, \ldots, c_t$ cifre predstavljanja u mešovitoj osnovi i gde je za svako i

$$|c_i| < \frac{1}{2} \cdot r_i .$$

Može se dokazati da je svaki ceo broj x jedinstveno predstavljen u intervalu $(-\frac{1}{2} \cdot R, \frac{1}{2} \cdot R)$ gde je $R = \prod_{i=1}^{t} r_i$.

Simetrično predstavljanje broja pomoću ostataka se može relativno jednostavno prevesti u simetrično predstavljanje sa mešovitom osnovom. Ovo se koristi da bi se brže dobio broj u fiksnoj osnovi od uobičajenog postupka pomoću Kineske teoreme o ostacima.
Na osnovu napred navedenih pojmova i uz pomoć odgovarajućih algoritama razvijen je program ESOLVE u /11/ za tačno rešavanje sistema linearnih jednačina čiji su koeficijenti celobrojni. Ovaj program možemo ukratko opisati :

Neka je dat sistem

$$A \cdot x = b$$

gde su A matrica koeficijenata a b vektor desne strane, celobrojni. Tada je postupak sledeći:

1° Prevesti A i b iz fiksne osnove u oblik simetričnog predstavljanja sa mešovitom osnovom,

2° Rešiti dobijeni sistem u modularnoj aritmetici za svaki prost broj p_i, $i = 1(1)MAXPRM$. MAXPRM se određuje u datom programu kao maksimalan broj prostih brojeva potrebnih za predstavljanje determinanti potrebnih za rešavanje datog sistema.

3° Prevesti dobijene determinante iz simetričnog oblika sa mešovitom osnovom u fiksnu osnovu.

Međutim, veliki problem za ovakav način rešavanja sistema linearnih jednačina predstavlja određivanje broja MAXPRM, jer se obično dobiju pesimistične procene, odnosno MAXPRM je ustvari mnogo veći nego što je stvarno potrebno.
Pored toga računanje determinanti na ovakav način nije najefikasnije. Zbog toga navodimo sledeću teoremu:

Teorema 4.3.1 Neka je dat sistem linearnih jednačina sa celobrojnim koeficijentima reda n (> 1) matricom

$$\begin{matrix} a_{11} & a_{12} & \cdots & a_{1n} & a_{1n+1} \\ a_{21} & a_{22} & \cdots & a_{2n} & a_{2n+1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdots & a_{nn} & a_{nn+1} \end{matrix}$$

gde su a_{in+1} , i = 1(1)n elementi desne strane sistema. Neka je $a_{11} \neq 0$ i primenimo sledeću transformaciju date matrice (sistema) u prvom koraku

$$a_{ij}^{(1)} = a_{11} \cdot a_{ij} - a_{i1} \cdot a_{1j} \qquad i,j \geqslant 2$$

a u svakom sledećem koraku

$$a_{ij}^{(k)} = \frac{a_{kk}^{(k-1)} \cdot a_{ij}^{(k-1)} - a_{ik}^{(k-1)} \cdot a_{kj}^{(k-1)}}{a_{k-1\,k-1}^{(k-2)}} \qquad k = 2(1)n-1, \quad i,j > k$$

ako je $a_{k-1\,k-1}^{(k-2)} \neq 0$ ($a_{ij}^{(0)} = a_{ij}$). Transformacijom se dobija sistem

$$\begin{matrix} a_{11} & a_{12} & \cdots & a_{1n} & a_{1n+1} \\ & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & a_{2n+1}^{(1)} \\ & & \cdots & & \\ & & a_{ii}^{(i-1)} & a_{in}^{(i-1)} & a_{in+1}^{(i-1)} \\ & & & \cdots & \\ & & & a_{nn}^{(n-1)} & a_{nn+1}^{(n-1)} \end{matrix}$$

pri čemu je $D = a_{nn}^{(n-1)}$, $D_n = a_{nn+1}^{(n-1)}$ i

$$D_i = \frac{1}{a_{ii}^{(i-1)}} \left(D \cdot a_{in+1}^{(i-1)} - \sum_{k=i+1}^{n} D_k\, a_{ik}^{(i-1)} \right) \quad i = n-1(-1)1,$$

gde je D oznaka za determinantu sistema a D_i su oznake za determinante koje odgovaraju nepoznatima. $a_{ij}^{(k)}$ su celi brojevi za svako k = 1(1)n-1; i,j > k.

Dokaz. Dokažimo da su $a_{ij}^{(k)}$ celi brojevi za svako k. Za k = 1 tvrđenje je tačno prema definiciji $a_{ij}^{(1)}$. Za k = 2 imamo da je

$$a_{ij}^{(2)} = \frac{(a_{11}a_{22} - a_{21}a_{12})(a_{11}a_{ij} - a_{i1}a_{1j}) - (a_{11}a_{i2} - a_{i1}a_{12})(a_{11}a_{2j} - a_{21}a_{1j})}{a_{11}} .$$

Pri množenju u brojiocu sabirci oblika $a_{21}\cdot a_{12}\cdot a_{i1}\cdot a_{1j}$ su suprotnih znakova a ostali sabirci sadrže kao faktor a_{11} pa je $a_{ij}^{(2)}$ ceo broj.

Pretpostavimo da je tvrđenje tačno za $1 \leqslant k-1$ $(k < n)$. Tada su $a_{kk}^{(k-1)}$, $a_{ij}^{(k-1)}$, $a_{ik}^{(k-1)}$, $a_{kj}^{(k-1)}$ celi brojevi i prema definiciji važi

$$a_{ij}^{(k)} = \frac{(a_{k-1\,k-1}^{(k-2)} a_{kk}^{(k-2)} - a_{k\,k-1}^{(k-2)} a_{k-1\,k}^{(k-2)})(a_{k-1\,k-1}^{(k-2)} a_{ij}^{(k-2)} - a_{i\,k-1}^{(k-2)} a_{k-1\,j}^{(k-2)})}{(a_{k-2\,k-2}^{(k-3)})^2\, a_{k-1\,k-1}^{(k-2)}} -$$

$$\frac{(a_{k-1\,k-1}^{(k-2)} a_{ik}^{(k-2)} - a_{i\,k-1}^{(k-2)} a_{k-1\,k}^{(k-2)})(a_{k-1\,k-1}^{(k-2)} a_{kj}^{(k-2)} - a_{k\,k-1}^{(k-2)} a_{k-1\,j}^{(k-2)})}{(a_{k-2\,k-2}^{(k-3)})^2\, a_{k-1\,k-1}^{(k-2)}}$$

$(a_{k-2\,k-2}^{(k-3)})^2 \mid a_{ij}^{(k)}$ jer su $a_{kk}^{(k-1)}$, $a_{ij}^{(k-1)}$, $a_{ik}^{(k-1)}$, $a_{kj}^{(k-1)}$ celi brojevi. Pri množenju u brojiocu sabirci oblika $a_{k\,k-1}^{(k-2)}\cdot a_{k-1\,k}^{(k-2)}\cdot a_{i\,k-1}^{(k-2)}\cdot a_{k-1\,j}^{(k-2)}$ su suprotnih znakova a ostali sabirci sadrže kao faktor $a_{k-1\,k-1}^{(k-2)}$ te $a_{k-1\,k-1}^{(k-2)} \mid a_{ij}^{(k)}$. Dakle, važi $(a_{k-2\,k-2}^{(k-3)})^2\, a_{k-1\,k-1}^{(k-2)} \mid a_{ij}^{(k)}$ pa je $a_{ij}^{(k)}$ takođe ceo broj.

Dokažimo sada da je vrednost determinante $D = a_{nn}^{(n-1)}$.

Ako je $n = 2$ tvrđenje je tačno na osnovu definicije $a_{ij}^{(1)}$.
Neka je sada $n > 2$, tada je

$$D = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & \cdots & \cdot \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}.$$

Neka je $a_{11} \neq 0$ tada deleći prvu vrstu sa a_{11} i oduzimajući prvu vrstu pomnoženu sa a_{i1} od i-te vrste imamo

$$D = a_{11}\cdot \begin{vmatrix} 1 & \frac{a_{12}}{a_{11}} & \cdots & \frac{a_{1n}}{a_{11}} \\ 0 & \frac{a_{11}a_{22}-a_{12}a_{21}}{a_{11}} & \cdots & \frac{a_{11}a_{2n}-a_{21}a_{1n}}{a_{11}} \\ \cdot & \cdot & \cdots & \cdot \\ 0 & \frac{a_{11}a_{n2}-a_{12}a_{n1}}{a_{11}} & \cdots & \frac{a_{11}a_{nn}-a_{n1}a_{1n}}{a_{11}} \end{vmatrix}.$$

Uzimajući u obzir oznake u formulaciji teoreme dobijamo

$$D = \frac{1}{a_{11}^{n-2}} \cdot \begin{vmatrix} a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3n}^{(1)} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n2}^{(1)} & a_{n3}^{(1)} & \cdots & a_{nn}^{(1)} \end{vmatrix} .$$

Primenjujući sličan postupak, determinantu možemo napisati

$$D = \frac{1}{\left(a_{22}^{(1)}\right)^{n-3}} \cdot \begin{vmatrix} a_{33}^{(2)} & a_{34}^{(2)} & \cdots & a_{3n}^{(2)} \\ a_{43}^{(2)} & a_{44}^{(2)} & \cdots & a_{4n}^{(2)} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n3}^{(2)} & a_{n4}^{(2)} & \cdots & a_{nn}^{(2)} \end{vmatrix} .$$

Primenjujući postupak i puta dobijamo (i = 1(1)n-1)

$$D = \frac{1}{\left(a_{ii}^{(i-1)}\right)^{n-(i+1)}} \begin{vmatrix} a_{i+1\,i+1}^{(i)} & \cdots & a_{i+1\,n}^{(i)} \\ \cdots & \cdots & \cdots \\ a_{n\,i+1}^{(i)} & \cdots & a_{nn}^{(i)} \end{vmatrix} .$$

Za i = n-1 dobijamo $D = a_{nn}^{(n-1)}$.

Vrednost x_n se dobija iz n-te jednačine transformisanog sistema pa zaključujemo da je $D_n = a_{n\,n+1}^{(n-1)}$.

Vrednost x_i (i < n) se dobija iz

$$x_i = \frac{1}{a_{ii}^{(i-1)}} \cdot \left(a_{i\,n+1}^{(i-1)} - \sum_{k=i+1}^{n} a_{ik}^{(i-1)} x_k\right)$$

a zbog $x_i = \frac{D_i}{D}$ dobijamo

$$D_i = \frac{1}{a_{ii}^{(i-1)}} \cdot \left(D \cdot a_{i\,n+1}^{(i-1)} - \sum_{k=i+1}^{n} a_{ik}^{(i-1)} D_k\right),$$

čime je dokaz teoreme završen.

Napomena 4.3.1 Pri dokazu teoreme pretpostavljali smo da je $a_{ii}^{(i-1)} \neq 0$ za i = 1(1)n. Ako to nije slučaj onda vršimo zamenu vrsta matrice što dovodi do promene znaka determinante.

Ova teorema je povoljna za izračunavanje determinanata sistema linearnih jednačina. Međutim, evidentno je da se teorema može primeniti i kada su koeficijenti brojevi u pokretnom zarezu.

Da bismo ilustrovali primenjivost teoreme za izračunavanja determinanti , primenimo je na sistem linearnih jednačina sa Hilbert-ovom matricom koeficijenata , koji je napred naveden . Pored toga ovaj primer pokazuje univerzalnost primene RN-preciznosti :

	mantisa u osnovi 10	stepen od 2 u osnovi 10
	0.6627536416053771972656 25	205
	0.72463756799697875976 5625	178
	0.78548240661621093750 0000	152
D	-0.60250550508499145507 8125	125
	-0.81945788860321044921 8750	98
	-0.83175569772720336914 0625	70
	-0.81889259815216064453 1250	45
	-0.77880859375000000000 0000	15
	-0.98235869407653808593 7500	197
	-0.96008294320785522460 9375	170
	0.56541097164154052734 3750	145
D(1)	0.52932125329971313476 5625	120
	0.93604618310928344726 5625	92
	-0.89797431230545043945 3125	67
	0.80744946002960205078 1250	42
	0.60188293457031250000 0000	17
	0.75829076766967773437 5000	205
	-0.69199484586715698242 1875	180
	0.95697522163391113281 2500	154
D(2)	-0.56850332021713256835 9375	129
	-0.86950373649597167968 7500	104
	-0.56252008676528930664 0625	79
	-0.76516819000244140625 0000	54
	0.51696121692657470703 1250	29
	-0.58552145957946777343 7500	211
	-0.84595960378646850585 9375	185
	0.80154812335968017578 1250	156
D(3)	-0.50269587175750732421 8750	130
	-0.70710051059722900390 6250	104
	-0.67954081296920776367 1875	79
	-0.92574775218963623046 8750	53
	-0.59375751018524169921 8750	27

	mantisa u osnovi 10	stepen od 2 u osnovi 10
	0.794342398643493652343750	215
	-0.508743166923522949218750	187
	0.752321893937225341796875	162
D(4)	-0.534836351871490478515625	136
	0.615510821342468261718750	111
	-0.940726518630981445312500	82
	-0.636612296104431152343750	56
	-0.553257048130035400390625	30
	-0.591890931129455566406250	219
	-0.610381484031677246093750	193
	0.871524870395660400390625	168
	-0.637456622455596923828125	143
D(5)	-0.853294610977172351562500	116
	-0.520784139633178710937500	91
	-0.595164299011230468750000	66
	0.749817609786987304687500	41
	0.564575195312500000000000	16
	0.544340312480926513671875	222
	-0.903412282466888427734375	197
	-0.627291321754455566406250	171
	0.896606266498565673828125	146
D(6)	0.808515548706054687500000	120
	-0.595142841339111328125000	91
	-0.917410492897033691406250	66
	-0.886448502540588378906250	41
	-0.902343750000000000000000	14
	-0.661295175552368164062500	224
	0.592113137245178222656250	199
	0.928684651851654052734375	174
	-0.969364643096923828125000	149
D(7)	-0.681598663330078125000000	123
	-0.638385593891143798828125	97
	0.772289752960205078125000	71
	-0.888674736022949218750000	46
	0.766937255859375000000000	19
	0.552683770656585693359375	226
	0.844367265701293945312500	199
	0.607805205749511718750000	172
	0.601423025131225585937500	145
D(8)	-0.603705240653991699218750	120
	0.606840193271636962890625	91
	-0.857409586906433105468750	65
	-0.740270912647247314453125	37
	0.968750000000000000000000	11

	mantisa u osnovi 10	stepen od 2 u osnovi 10
	-0.650126516819000244140625	227
	0.983092010021209716796875	201
	-0.504446208477020263671875	175
	0.994257748126983642578125	150
D(9)	0.895798861980438232421875	125
	0.769439101219177246093750	99
	-0.612452268600463867187500	71
	-0.515552341938018798828125	46
	-0.608652114868164062500000	21
	0.542714774608612060546875	228
	0.555112183094024658203125	202
	0.719886422157287597656250	177
D(10)	0.937697720527648925781250	152
	0.718120932579040527343750	127
	-0.801196992397308349609375	99
	-0.948928773403167724609375	73
	-0.958159069088439941406250	48
	0.880371093750000000000000	22
	-0.639462590217590332031250	228
	0.884004712104797363281250	202
	0.983649551868438720703125	176
	-0.512457013130187988281250	149
D(11)	0.667729854583740234375000	124
	-0.782320141792297363281250	99
	0.883344948291778564453125	74
	-0.877006947994232177734375	49
	0.771332740783691406250000	24
	0.519786894321441650390625	228
	0.982159793376922607421875	202
	-0.966936945915222167968750	176
	0.818073391914367675781250	151
D(12)	0.728874683380126953125000	125
	-0.877319931983947753906250	97
	0.939694941043853759765625	68
	0.766196548938751220703125	43
	-0.724609375000000000000000	17
	-0.554649353027343750000000	227
	-0.895546019077301025390625	202
	0.656187951564788818359375	176
	-0.767841398715972900390625	146
D(13)	-0.502943217754364013671875	120
	0.867361664772033691406250	94
	0.560799598693847656250000	67
	-0.767716109752655029296375	40
	0.978271484375000000000000	15

	mantisa u osnovi 10	stepen od 2 u osnovi 10
	0.699151217937469482421875	225
	-0.960833311080932617187500	198
	-0.675047159194946289062500	173
	0.834237754344940185546875	148
D(14)	-0.950527131557464599609375	123
	0.779732227325439453125000	98
	0.780215620994567871093750	73
	0.810007870197296142578125	46
	0.859375000000000000000000	14
	-0.788924217224121093750000	222
	0.528063840503692626953125	197
	-0.927234649658203125000000	172
	-0.910792469978332519531250	147
D(15)	0.636222422122955322265625	121
	0.722272276878356933593750	94
	0.564495325088500976562500	69
	0.551249861717224121093750	44
	0.503112792968750000000000	18

4.4 Formiranje karakterističnog polinoma matrice

Imajući rešen problem sistema linearnih jednačina pomoću vrednosti odgovarajućih determinanata u RN-preciznosti, tada problem sopstvenih vrednosti možemo rešiti formiranjem karakterističnog polinoma metodom Krilov-a.Na osnovu teoreme 4.3.1 možemo izračunati determinante odgovarajućeg sistema linearnih jednačina , čije su vrednosti nizovi u RN-preciznosti.Na dobijeni polinom primenimo neki od algoritama za nalaženje nula polinoma, jer smo pitanje izračunavanja vrednosti polinoma rešili u delu 2.7.

4.5 Zaključak

U radu je uvedena RN-preciznost i pomoću nje je razmatran problem tačnosti izračunavanja. Za razliku od uobičajene proširene preciznosti kao što je Brent-ov MP-program, RN-preciznost je praktičnija za tačna izračunavanja. Primena MP-programa za tačna izračunavanja nameće problem određivanja a priori broja cifara krajnjeg rezultata, dok je kod RN-preciznosti to automatizovano, jer se čuva dužina niza za vreme izračunavanja.

Ako prihvatimo modularnu aritmetiku za rešavanje relativno uske oblasti problema kao što je sistem linearnih jednačina, onda se i tu susrećemo takođe sa proširenom preciznošću. Pored toga, pri određivanju broja prostih brojeva potrebnih za predstavljanje vrednosti determinante (MAXPRM), mora se dati ocena vrednosti determinanata na osnovu Hadamard-ove nejednakosti, što ponovo uključuje proširenu preciznost, jer vrednosti determinanata zahtevaju velike vrednosti za eksponent.

Istraživanja koja su izvršena u ovom radu imala su u vidu pre svega, tip računara sa klasičnom računarskom aritmetikom, koji su dostupni širokom krugu korisnika (ne postavljaju se zahtevi za uvođenjem posebne aritmetike kao u slučaju intervalne aritmetike) . Imajući takvo opredeljenje u istraživanju, uvedena je relacija $\overset{\boxplus}{<}$(definicija 2.4.2), koja je relativizirana prema aritmetici računara (zaokruživanju). Sledeći korak je definicija RN-preciznosti (definicija 2.4.5). Osnovu za aritmetiku u RN-preciznosti predstavlja drugi agoritam tačnog sumiranja , razvijen u 2.4 , koji proizvoljan niz brojeva u pokretnom zarezu redukuje u normalizovan u odnosu na $\overset{\boxplus}{<}$. Za drugi algoritam tačnog sumiranja važnu ulogu ima operacija tačno sumiranje dva broja u pokretnom zarezu, koju možemo označiti simbolično

$$x + y = (x_1 , y_1),$$

gde je x_1 najbolja aproksimacija, u odnosu na zaokruživanje, tačnog zbira x + y, a y_1 je razlika između x + y i x_1. Za dalji razvoj aritmetike u RN-preciznosti potreban je algoritam tačnog množenja niza sa brojem u pokretnom zarezu, razvijen u 2.6. Za ovaj algoritam je važna operacija tačnog množenja dva broja u pokretnom zarezu , koju možemo označiti simbolično

$$x \cdot y = (x_1 , y_1),$$

gde je x_1 najbolja aproksimacija, u odnosu na zaokruživanje, tačnog proizvoda $x \cdot y$, a y_1 je razlika između $x \cdot y$ i x_1. Na osnovu drugog algoritma tačnog sumiranja i algoritma tačnog množenja niza brojem u pokretnom zarezu, razvijeni su algoritmi u 2.7, za tačno izračunavanje vrednosti polinoma i

najbolje aproksimacije, u odnosu na zaokruživanje, vrednosti polinoma . Algoritmi za izračunavanje vrednosti polinoma mogu se koristiti za nalaženje jednostruke realne nule polinoma u obliku najbolje aproksimacije nule ili u obliku intervala, čiji su krajevi susedni brojevi u pokretnom zarezu. Ovaj algoritam je razvijen u 2.8.
Aritmetika u RN-preciznosti je kompletirana algoritmima za tačno množenje dva niza (razvijen u 2.9) i za deljenje dva niza (razvijen u 2.11).
Svi algoritmi su realizovani u VAX-FORTRAN-u u obliku potprograma, i mogu se koristiti(uz eventualne modifikacije) za razvoj kompleksnijih algoritama velike tačnosti.

U zavisnosti od preciznosti sa kojom računamo , vreme izvršavanja algoritama se menja. Međutim, mikroprogramska realizacija operacija izdvajanja i postavljanja eksponenta, tačno sabiranja i tačno množenja , bi doprinela bržem izvršavanju algoritama,što se i inače čini pri modifikaciji aritmetike (modifikacija je znatnija u slučaju intervalne atitmetike, napr u /33/ ili /34/).

U trećem delu rada se razmatraju približna izračunavanja aritmetičkog izraza. Za procenu a priori broja tačnih cifara mogu se primeniti teoreme 3.1.5, 3.1.6 i 3.1.7.
Te teoreme tvrde da u slučaju sabiranja i oduzimanja greška napreduje najviše za jednu, a u slučaju množenja i deljenja najviše za dve cifre, ako je osnova sistema b > 2.
U četvrtom delu su navedene primene algoritama iz drugog dela na rešavanje sistema linearnih jednačina. Teorema 4.3.1 tvrdi da se determinante sistema linearnih jadnačina mogu izračunati pomoću Gauss-ove eliminacije , što je veoma povoljno za primenu.
Međutim, u RN-preciznost nisu za sada uvedena zaokruživanja na dole i gore za realizaciju intervalnih operacija što je učinjeno u MP-programu.
Jedan od mogućih pravaca daljeg istraživanja može da bude uvođenje zaokruživanja na dole i gore u RN-preciznost, što bi omogućilo primenu u intervalnoj aritmetici.
Od interesa za dalje istraživanje može da bude i varijanta Bohlender-ovog algoritma za RN-preciznost, odnosno algoritma sumiranja , koji bi dati niz dužine n redukovao u RN-preciznost dužine m ($m \ll n$).

Spisak literature

/1/ Accurate Scientific Computations, Proceedings, 1985, Eds Miranker, W. L. and Toupin, R. A., Springer-Verlag, 1986

/2/ Alefeld, G. and Herzberger, J., Introduction to Interval Computations, Academic Press, 1983

/3/ Bareiss, E.H.,Computational Solutions of Matrix Problems Over an Integral Domain, J. Inst. Maths. Applics, 1972, 10, 68-104

/4/ Blue, J. L., A Portable Fortran Program to Find the Euclidean Norm of a Vector, ACM Trans. Math. Soft., 4, 1 (Mar 1978), 15-23

/5/ Bohlender, G., Floating point computations of functions with maximum accuracy, IEEE Trans. Comput., C-26, No 7, 621-632, 1977.

/6/ Bohm H.,Berechnung von Polynomnullstellen und Auswertung Aritmetischer Ausdrucke mit Garantierter Maximaler Genauigkeit - Dissertation, Universitat-Karlsruhe 1983.

/7/ Bohm H., Evaluation of aritnmetic expressions with maximum accuracy in New Approuch to Scientific Computations, editors Kulisch , U. and Miranker, W.L, Academic Press, 1983.

/8/ Brent, R. P. A FORTRAN multiple - precision arithmetic package. ACM Trans.Math.Soft. 4, 1(Mar 1978), 57-70.

/9/ Bus, J. C.P. and Dekker, T. J., Two Efficient Algorithms with Guaranteed Convergence for Finding a Zero of a Function, ACM Trans. Math. Soft., 1,4(Dec 1975), 330-345

/10/ Cabay, S. and Lam, T.P.L., Congruence Techniques for the Exact Solution of Integer Systems of Linear Equations, ACM Trans. Math. Soft., 3, 4(Dec 1977), 386-397

/11/ Cabay , S. and Lam , T.P.L., ALGORITHM 522 ESOLVE , Congruence Techniques for the Exact Solution of Integer Systems of Linear Equations, ACM Trans. Math. Soft., 3, 4(Dec 1977), 404-410

/12/ Clenshaw, C.W. and Olver, F.W.J., Beyond Floating Point, JACM, V-31, No. 2,(Apr 1984), 319-328

/13/ Cody W. J., Analysis of Proposals for the Floating Point Standard, Computer, March 1981, 63-68

/14/ Cody W.J.,Floating-point Parameters,Models and Standards, The Relation Between Numerical Computation and Programming Languages, J. K. Reid (ed), North-Holland, IFIP 1982

/15/ Dekker, T. J. A Floating - point technique for extending the available precision, Numer. Math. 18, 224-242, 1971

/16/ Demidovich, B. P. and Maron, I. A., Computational mathematics, Mir Publishers-Moscow, 1981.

/17/ Demmel J.W. and Kruckeberg F., An Interval Algorithm for Solving Systems of Linear Equations to Prespecified Accuracy, Computing 34, 117-129, (1985)

/18/ Forsythe G. and Molar C.B., Computer Solution of Linear Algebraic Systems, Prentice-Hall, Englewood Cliffs, 1967

/19/ Gregory, R. T. and Krishnamurthy, E. V. , Methods and Applications of Error-free Computation, Springer-Verlag, 1984

/20/ Hahn, W., Mohr, K. and Shauar, U., Some Techniques for Solving Linear Equation Systems with Guarantee,Computing 34, 375-379(1985)

/21/ Howell, J.A. and Gregory, R.T., An Algorithm for Solving Linear Algebraic Equations Using Residue Arithmetic I, BIT 9(1969), 200-224

/22/ Howell, J.A. and Gregory R.T., Solving Linear Equations Using Residue Arithmetic-Algorithm II, BIT 10(1970), 23-37

/23/ Hull T.E., The Use of Controled Precision , The Relation Between Numerical Computation and Programming Languages, J.K. Reid (ed), North-Holland, IFIP 1982

/24/ Hull, T.E. and Abraham, A. , Properly Rounded Variable Precision Square Root, ACM Trans. Math. Soft. 11, 3 (Sep 1985), 229-237

/25/ Interval Mathematics 1985, Proceedings,1985,Ed K. Nickel, Springer-Verlag 1985

/26/ Jansen, P. and Weidner, P. , High-Accuracy Arithmetic Software - Some Tests of ACRITH Problem-Solving Routines, ACM Trans. Math. Soft. 12, 1(Mar 1986), 62-70

/27/ Jovanović B.S., Numerička analiza, Beograd 1984

/28/ Kaucher E.W. and Rump S.M.,Generalized Iteration Methods for Bounds of the Solution of Fixed Point Operator-Equations, Computing 24, 131-137(1980)

/29/ Kaucher, E.W. and Rump, S.M. , E-Methods for fixed Point Equation f(x) = x, Computing 28, 31-42(1982)

/30/ Kaucher E.W. and Miranker W.L., Self-Validating Numerics for Function Space Problems, Academic Press, 1984

/31/ Knuth, D.,The art of Computar Programming,Vol.2. Addison Wesley, Reading, Massacnusetts, 1980.

/32/ Kulisch U.,An Axiomatic Approach to Rounded Computations, Numer. Math. 18,1-17(1971)

/33/ Kulisch, U. and Miranker, W. L., Computer Arithmetic in Theory and Practice, Academic Press, 1981.

/34/ Kulisch, U.W. and Miranker, W.L. , The Arithmetic of the Digital Computer: A New Approach, SIAM Review vol. 28,No 1., March 1986

/35/ Kulisch, U.W. and Stetter, H.J. (editors) , Scientific Computation with Automatic Result Verification, Springer Verlag 1988

/36/ Kurepa S. , Konačno dimenzioni vektorski prostori i primjene, Sveučilišna naklada Liber, Zagreb, 1986

/37/ Lange E., Implementation and Test of the ACRITH Facility in a System/370, IEEE Trans. Computers, C-36, No. 9, Sep 1987(1088-1096)

/38/ Linnainmaa, S. Software for doubled - precision floating point Computations. ACM Trans.Math.Soft. 7, 3(Sept 1981) 272-283.

/39/ Matula D.W. and Kornerup P., Finite Precision Rational Arithmetic: Slash Number systems, IEEE Trans. Computers, C-34, No. 1, Jan 1985,(3-18)

/40/ Milovanović G.V. , Numerička analiza I , Naučna knjiga, Beograd, 1985

/41/ Mitrinović D.S., Indukcija-binomna formula-kombinatorika, Matematička bibliotaka 26 , Zavod za izdavanje udžbenika Srbije, Beograd, 1963

/42/ Mitrinović D.S. i Đoković D.Ž., Polinomi i matrice, Građevinska knjiga, Beograd, 1986

/43/ Moore, R.E., Interval Analysis, Prentice-Hall, Englewood Cliffs, New Jersey, 1966

/44/ Moore, R. E., Methods and Aplications of Interval Analysis, SIAM Studies in Applied Mathematics. SIAM, Philadelphia, Pennsylvania, 1979.

/45/ Ratschek H. and Rokne J., Computer Methods for the Range of Functions, Ellis Horwood Limited, 1984

/46/ Regener E. , Multiprecision Integer Division Examples Using Arbitrary Radix, ACM Trans. Math. Soft., 10, 3(Sep 1984), 324-338

/47/ Rump S.M.,Polymomial Minimum Root Separation,Mathematics of Computation, Vol 33, Num 145, Jan 1979, 327-336

/48/ Rump, S. M., Kleine Fehlerschranken bei Matrixproblemen, Dissertation, Universitat-Karlsruhe 1980

/49/ Rump, S. M. , Solving Nonlinear Systems with Least Significant Bit Accuracy, Computing 29, 183-200(1982)

/50/ Rump S.M. and Bohm H. , Least Significant Bit Evaluation of Arithmetic Expresions in single-precision , Computing 30, 189-199(1983)

/51/ Rump S.M., Solving Algebraic Problems with High Accuracy in New Approach to Scientific Computations , ed. Kulisch U. and Miranker W.L., Academic Press 1983

/52/ Rump, S.M., Solution of linear Systems with verified Accuracy, Appl. Num. Math. 3(1987), 233-241

/53/ Springer J., Exact Solution of General Integer Systems of Linear Equations, ACM Trans. Math. Soft., 12, 1(Mar 1986)

/54/ Szabo, N.S. and Tanaka, R.I., Residue Arithmetic and Its Applications to Computer Technology, Mc Graw-Hill, 1967

/55/ Tasković M.R., Osnove teorije fiksne tačke , Matematička biblioteka 50 , Zavod za udžbenike i nastavna sredstva , Beograd 1986

/56/ Virkkunen, J. A unified approach to floating point rounding with applications to multiple - precision summation. Rep. A - 1980-1. Dep. of Computer Sci., Univ. of Helsinki, Finland, 1980

/57/ Yohe J.M., Roundings in Floating-point Arithmetic , IEEE Trans. Computers, C-22, No 6, June 1973(577-586)

/58/ Yohe J.M.,Software for Interval Arithmetic: A Reasonable Portable Package, ACM Trans. Math. Soft. 5, 1(Mar 1979), 50-63

P R I L O Z I

```
      INTEGER FUNCTION IEXS(X)

POTPROGRAM ODREDJUJE VREDNOST EKSPONENTA
REALNOG BROJA X TIPA R4 I DODELJUJE
PROMENJIVOJ IEXS TIPA I4

X REALAN BROJ TIPA R4 - ULAZ
IEXS EKSPONENT TIPA I4 REALNOG BROJA X - IZLAZ
      REAL*4 X, XP
      INTEGER*2 EX, I, J
      LOGICAL*1 Y(4), Z(2)
      EQUIVALENCE (XP,Y(1)), (I,Z(1))
      IEXS = -128
      IF(X .EQ. 0.)RETURN
      XP = X
      Z(2) = Y(1)
      J = I
      I = 0
      Z(1) = Y(2)
      IF(X .LT. 0) I = I - 128
      IF(J .LT. 0) EX = 2*I + 1
      IF(J .GE. 0) EX = 2*I
      IEXS = EX - 128
      RETURN
      END
      INTEGER FUNCTION IEXD(X)

POTPROGRAM ODREDJUJE VREDNOST EKSPONENTA
REALNOG BROJA X TIPA R8 I DODELJUJE
PROMENJIVOJ IEXD TIPA I4

X REALAN BROJ TIPA R8 - ULAZ
IEXD EKSPONENT TIPA I4 REALNOG BROJA X - IZLAZ

      REAL*8 X, XP
      INTEGER*2 EX, I, J
      LOGICAL*1 Y(8), Z(2)
      EQUIVALENCE (XP,Y(1)), (I,Z(1))
      IEXD = -128
      IF(X .EQ. 0.)RETURN
      XP = X
      Z(2) = Y(1)
      J = I
      I = 0
      Z(1) = Y(2)
      IF(X .LT. 0) I = I - 128
      IF(J .LT. 0) EX = 2*I + 1
      IF(J .GE. 0) EX = 2*I
      IEXD = EX - 128
      RETURN
      END
```

```
      SUBROUTINE TSXY(X,Y,S,R)

C POTPROGRAM ODREDJUJE TACNU SUMU BROJEVA X, Y
C TIPA R4 I PREDSTAVLJA SUMU U OBLIKU APROKSIMACIJE
C S = X + Y I RAZLIKE TACNE SUME I APROKSIMACIJE R

C X I Y REALNI BROJEVI TIPA R4 - ULAZ
C S I R REALNI BROJEVI (TACNA SUMA) TIPA R4 - IZLAZ
      INTEGER EX,EY,D,L
      REAL X,Y,S,R
      REAL*8 S1
      DATA L/24/
      D = IEXS(X) - IEXS(Y)
      IF(IABS(D) .GT. (L+1)) GO TO 10
      S = X + Y
      S1 = X
      S1 = S1 + Y
      S1 = S1 - S
      R = S1
      RETURN
10    IF( D .LT. 0 ) GO TO 20
      S = X
      R = Y
      RETURN
20    S = Y
      R = X
      RETURN
      END
      REAL*4 FUNCTION PEXS(X,EX,IGR)

C POTPROGRAM DODELJUJE EKSPONENTU BROJA
C X TIPA R4 VREDNOST PROMENLJIVE EX I
C DOBIJENI BROJ DODELJUJE PEXS

C X REALAN BROJ TIPA R4 - ULAZ
C EX CEO BROJ TIPA I4 - ULAZ
C IGR CEO BROJ TIPA I4 (INDIKATOR GRESKE) - IZLAZ
C PEXS REALAN BROJ TIPA R4 - IZLAZ

      REAL*4 X,XP
      INTEGER EX
      INTEGER*2 I
      LOGICAL*1 Y(4), Z(2)
      EQUIVALENCE (XP,Y(1)),(I,Z(1))
      IGR = 0
      I = EX + 128
      IF(0 .LE. I .AND. I .LE. 255) GO TO 20
      TYPE 10, I
10    FORMAT(' VREDNOST EKSPONENTA VAN [-128,127] ',I6)
      IGR = 1
      RETURN
20    XP = X
      I = I/2
      IF(X .LT. 0.) I = I + 128
```

```
      Y(2) = Z(1)
      I = 0
      Z(1) = Y(1)
      IF(I .GE. 128) I = I - 128
      IF(MOD(EX,2) .NE. 0) GO TO 30
      Y(1) = Z(1)
      PEXS = XP
      RETURN
30    I = I + 128
      Y(1) = Z(1)
      PEXS = XP
      RETURN
      END
      REAL*8 FUNCTION PEXD(X,EX,IGR)
C POTPROGRAM DODELJUJE EKSPONENTU BROJA
C X TIPA R8 VREDNOST PROMENLJIVE EX I
C DOBIJENI BROJ DODELJUJE PEXD

C X REALAN BROJ TIPA R8 - ULAZ
C EX CEO BROJ TIPA I4 - ULAZ
C IGR CEO BROJ TIPA I4 (INDIKATOR GRESKE) - IZLAZ
C PEXD REALAN BROJ TIPA R8 - IZLAZ

      REAL*8 X,XP
      INTEGER EX
      INTEGER*2 I
      LOGICAL*1 Y(8), Z(2)
      EQUIVALENCE (XP,Y(1)),(I,Z(1))
      IGR = 0
      I = EX + 128
      IF(0 .LE. I .AND. I .LE. 255) GO TO 20
      TYPE 10, I
10    FORMAT(' VREDNOST EKSPONENTA VAN [-128,127] ',I6)
      IGR = 1
      RETURN
20    XP = X
      I = I/2
      IF(X .LT. 0.D0) I = I + 128
      Y(2) = Z(1)
      I = 0
      Z(1) = Y(1)
      IF(I .GE. 128) I = I - 128
      IF(MOD(EX,2) .NE. 0) GO TO 30
      Y(1) = Z(1)
      PEXD = XP
      RETURN
30    I = I + 128
      Y(1) = Z(1)
      PEXD = XP
      RETURN
      END
```

```
      REAL*4 FUNCTION SLPR(X,I)

C POTPROGRAM ODREDJUJE SLEDECI ILI PRETHODNI BROJ
C U POKRETNOM ZAREZU ZAVISNO DA LI JE I > 0 ILI
C I < 0 ZA REALAN BROJ X TIPA R4

C X REALAN BROJ TIPA R4 - ULAZ
C I CEO BROJ TIPA I4 - ULAZ
C SLPR REALAN BROJ TIPA R4 - IZLAZ

      REAL*4 X,X1,X2,X3
      INTEGER IGR,EX,EX1
      DATA X2/'80'X/
      IF(I .NE. 0) GO TO 20
      TYPE 10, I
10    FORMAT(' I = 0 U POTPROGRAMU SLPR ',I2)
      STOP
20    CONTINUE
      IF(X .NE. 0.) GO TO 30
      IF(I .GT. 0) SLPR = X2
      IF(I .LT. 0) SLPR =-X2
      RETURN
30    CONTINUE
      IF(X .NE. -X2) GO TO 40
      IF(I .LT. 0) GO TO 40
      SLPR = 0.
      RETURN
40    EX = IEXS(X)
      X1 = PEXS(X,0,IGR)
      X3 = PEXS(0.D0,-23,IGR)
      IF(I .GT. 0) SLPR = X1 + X3
      IF(I .LT. 0) SLPR = X1 - X3
      EX1 = IEXS(SLPR) + EX
      SLPR = PEXS(SLPR,EX1,IGR)
      RETURN
      END
      SUBROUTINE CNV(X)

C POTPROGRAM ZAPISUJE REALAN BROJ
C TIPA R4 U HEKSADECIMALNOM OBLIKU

      REAL*4 X, X1
      LOGICAL*1 Y(4)
      EQUIVALENCE (X1,Y(1))
      X1 = X
      TYPE 10, (Y(2*I),Y(2*I-1),I=1,2)
10    FORMAT(' '4Z2)
      RETURN
      END
```

```
      SUBROUTINE CNVD(X)

C POTPROGRAM ZAPISUJE REALAN BROJ
C TIPA R8 U HEKSADECIMALNOM OBLIKU

      REAL*8 X, X1
      LOGICAL*1 Y(8)
      EQUIVALENCE (X1,Y(1))
      X1 = X
      TYPE 10, (Y(2*I),Y(2*I-1),I=1,4)
10    FORMAT(' '8Z2)
      RETURN
      END
      SUBROUTINE TSUMA(A,N,K)

C POTPROGRAM ODREDJUJE TACNU SUMU NIZA A DUZINE N
C I UREDJENU VREDNOST SUME ZADRZAVA U A DUZINE K.

C A NIZ DUZINE N TIPA R4 - ULAZ, IZLAZ
C N CEO BROJ DUZINA NIZA A NA ULAZU, TIPA I4 - ULAZ
C K CEO BROJ DUZINA NIZA A NA IZLAZU, TIPA I4 - IZLAZ

      REAL*4 A(N),S,S1,R
      INTEGER I,K,M
      IF(N .GT. 0) GO TO 20
      TYPE 10, N
10    FORMAT(' N <= 0 U POTPROGRAMU TSUMA ',I10)
      RETURN
20    K = N
30    IF(K .LE. 1) RETURN
      M = K
      K = 1
      NR = 0
      S = A(1)
      DO 40 I = 2,M
      IF(A(I) .EQ. 0.) GO TO 40
      CALL TSXY(S,A(I),S1,R)
      IF(S .EQ. S1) NR = NR + 1
      A(I) = 0.
      IF(R .NE. 0.) GO TO 50
      S = S1
      GO TO 40
50    CONTINUE
      A(K) = S1
      S = R
      K = K + 1
40    CONTINUE
      A(K) = S
      IF(NR .EQ. (K-1)) RETURN
      GO TO 30
      END
```

```
      SUBROUTINE TSUMAP(AM,AE,N,K)
C POTPROGRAM ODREDJUJE TACNU SUMU NIZA A DUZINE N
C I UREDJENU VREDNOST SUME DODELJUJE NIZU B DUZINE K.
C KORISTI SE U SLUCAJU DA INTERVAL ZA EKSPONENT
C [-128,127] NIJE DOVOLJAN ZA REALIZACIJU ALGORITMA.

C AM NIZ MANTISA NIZA A DUZINE N, TIPA R4 - ULAZ, IZLAZ
C AE NIZ EKSPONENATA NIZA A DUZINE N TIPA I4 - ULAZ, IZLAZ
C N CEO BROJ DUZINA NIZA A NA ULAZU, TIPA I4 - ULAZ
C K CEO BROJ DUZINA NIZA A NA IZLAZU, TIPA I4 - IZLAZ

      REAL*4 AM(N),SM,S1M,RM
      INTEGER AE(N),SE,S1E,RE
      INTEGER I,K,M
      IF(N .GT. 0) GO TO 20
      TYPE 10, N
10    FORMAT(' N <= 0 U POTPROGRAMU TSUMAP ',I10)
      RETURN
20    CONTINUE
      K = N
30    IF(K .LE. 1) RETURN
      M = K
      K = 1
      NR = 0
      SM = AM(1)
      SE = AE(1)
      DO 40 I = 2,M
      IF(AM(I) .EQ. 0.) GO TO 40
      CALL TSXYP(SM,SE,AM(I),AE(I),S1M,S1E,RM,RE)
      IF(SM .EQ. S1M .AND. SE .EQ. S1E) NR = NR + 1
      AM(I) = 0.
      IF(RM .NE. 0.) GO TO 50
      SM = S1M
      SE = S1E
      GO TO 40
50    CONTINUE
      AM(K) = S1M
      AE(K) = S1E
      SM = RM
      SE = RE
      K = K + 1
40    CONTINUE
      AM(K) = SM
      AE(K) = SE
      IF(NR .EQ. (K-1)) RETURN
      GO TO 30
      END
```

```
      SUBROUTINE RAZD(X,XM,XE)

C POTPROGRAM PREDSTAVLJA REALAN BROJ X TIPA R4
C U OBLIKU MANTISE XM TIPA R4 I EKSPONENTA XE
C TIPA I4

C X REALAN BROJ TIPA R4 - ULAZ
C XM MANTISA BROJA X, TIPA R4 - IZLAZ
C XE EKSPONENT BROJA X, TIPA I4 - IZLAZ

      REAL*4 X, XM
      INTEGER XE,IEXS,IGR
      IF(X .NE. 0.) GO TO 10
      XM = 0.
      RETURN
10    XE = IEXS(X)
      XM = PEXS(X,0,IGR)
      RETURN
      END
      SUBROUTINE RAZDN(A,AM,AE,N)
      REAL*4 A(N), AM(N)
      INTEGER AE(N)
      IF(N .LE. 0) THEN
                   TYPE 10,N
10                 FORMAT(' N <= 0 U POTPROGRAMU RAZDN',I5)
                   RETURN
                   END IF
      DO 20 I = 1,N
      CALL RAZD(A(I),AM(I),AE(I))
20    CONTINUE
      RETURN
      END
      SUBROUTINE SAST(XM,XE,X,IGR)

C POTPROGRAM SPAJA XM, XE U REALAN BROJ X

C XM MANTISA BROJA X, TIPA R4 - ULAZ
C XE EKSPONENT BROJA X, TIPA I4 - ULAZ
C X DOBIJEN SPAJANJEM XM I XE, TIPA R4 - IZLAZ
C IGR INDIKATOR GRESKE, TIPA I4 - IZLAZ

      REAL*4 XM, X, PEXS
      INTEGER XE
      IGR = 0
      IF(-128 .LE. XE .AND. XE .LE. 127) GO TO 20
      TYPE 10, XE
10    FORMAT(' VREDNOST EKSPONENTA VAN [-128,127] ',I10)
      IGR = 2
      RETURN
20    IF(XM .NE. 0.) GO TO 30
      X = 0.
      RETURN
30    X = PEXS(XM,XE,IGR)
      RETURN
      END
```

```
      SUBROUTINE SASTN(AM,AE,A,N,K)

C POTPROGRAM SPAJA NIZ MANTISA AM I NIZ EKSPONENATA
C AE DUZINE N U NIZ REALNIH BROJEVA A DUZINE K.
C K <= N JER SVI CLANOVI NIZA (AM,AE) NE MORAJU
C BITI PREDSTAVLJIVI KAO REALNI BROJEVI.

C AM NIZ MANTISA DUZINE N, TIPA R4 - ULAZ
C AE NIZ EKSPONENATA DUZINE N, TIPA I4 - ULAZ
C N DUZINA AM I AE, TIPA I4 - ULAZ
C A NIZ REALNIH BROJEVA DUZINE K, TIPA R4 - IZLAZ
C K DUZINA NIZA A, TIPA I4 - IZLAZ

      REAL*4 A(N), AM(N)
      INTEGER AE(N), N, K
      IF(N .LE. 0) THEN
                   TYPE 10,N
10                 FORMAT(' N <= 0 U POTPROGRAMU SASTN', I5)
                   RETURN
                   END IF
      DO 40 I = 1,N
      CALL SAST(AM(I),AE(I),A(I),IGR)
      IF(I .EQ. 1 .AND. IGR .NE. 0) THEN
                                    TYPE 20
20                                  FORMAT(' NIZ NIJE PREDSTAVLJIV')
                                    RETURN
                                    END IF
      IF(IGR .EQ. 0) THEN
                     K = I
                     ELSE
                     TYPE 30, I
30                   FORMAT(' 'I5,' CLAN NIZA NIJE PREDSTAVLJIV')
                     RETURN
                     END IF
40    CONTINUE
      RETURN
      END
      SUBROUTINE SXY(XM,XE,YM,YE,ZM,ZE)

C POTPROGRAM ODREDJUJE SUMU BROJEVA X I Y KADA SU
C OVI DATI U OBLIKU MANTISE TIPA R4 I EKSPONENTA
C TIPA I4 I PREDSTAVLJA JE U OBLIKU MANTISE ZM I
C EKSPONENTA ZE

C XM MANTISA X, TIPA R4 - ULAZ
C XE EKSPONENT X, TIPA I4 - ULAZ
C YM MANTISA Y, TIPA R4 - ULAZ
C YE EKSPONENT Y, TIPA I4 - ULAZ
C ZM MANTISA Z, TIPA R4 - IZLAZ
C ZE EKSPONENT Z, TIPA I4 - IZLAZ
```

```
      REAL*4 XM, YM, ZM, X1, Y1, Z1, T
      INTEGER XE, YE, ZE, IX, IY, IZ, IR, IGR
      ZE = MAX(XE,YE)
      X1 = XM
      IX = XE
      Y1 = YM
      IY = YE
      IF(XE .GE. YE) GO TO 10
C
C      ZAMENI MESTA
C
      Z1 = XM
      IZ = XE
      X1 = Y1
      IX = IY
      Y1 = Z1
      IY = IZ
10    CONTINUE
      IR = IY - IX
      T = PEXS(Y1,IR,IGR)
      ZM = XM + T
      ZE = ZE + IEXS(ZM)
      ZM = PEXS(ZM,0,IGR)
      RETURN
      END
      SUBROUTINE PXYP(XM,XE,YM,YE,ZM,ZE)
C POTPROGRAM ODREDJUJE PROIZVOD BROJEVA X I Y KADA
C SU OVI DATI U OBLIKU MANTISA TIPA R4 I EKSPONENT
C TIPA I4 I PREDSTAVLJA U OBLIKU MANTISE ZM I
C EKSPONENTA ZE

C XM MANTISA X, TIPA R4 - ULAZ
C XE EKSPONENT X, TIPA I4 - ULAZ
C YM MANTISA Y, TIPA R4 - ULAZ
C YE EKSPONENT Y, TIPA I4 - ULAZ
C ZM MANTISA Z, TIPA R4 - IZLAZ
C ZE EKSPONENT Z, TIPA I4 - IZLAZ

      REAL*4 XM, YM, ZM, T
      INTEGER XE, YE, ZE, IGR
      IF(XM .EQ. 0. .OR. YN .EQ. 0.) THEN
                                      ZM = 0.
                                      RETURN
                                      END IF
      ZE = XE + YE
      T = XM * YM
      ZE = ZE + IEXS(T)
      ZM = PEXS(T,0,IGR)
      RETURN
      END
```

```
      SUBROUTINE TPXYP(XM,XE,YM,YE,PM,PE,RM,RE)
C POTPROGRAM ODREDJUJE TACAN PROIZVOD BROJEVA
C X I Y KADA SU OVI DATI U OBLIKU MANTISE I
C EKSPONENTA I PREDSTAVLJA PROIZVOD U OBLIKU
C PROIZVODA P I OSTATKA R

C XM MANTISA X, TIPA R4 - ULAZ
C XE EKSPONENT X, TIPA I4 - ULAZ
C YM MANTISA Y, TIPA R4 - ULAZ
C YE EKSPONENT Y, TIPA I4 - ULAZ
C PM MANTISA P, TIPA R4 - IZLAZ
C PE EKSPONENT P, TIPA I4 - IZLAZ
C RM MANTISA P, TIPA R4 - IZLAZ
C RE EKSPONENT R, TIPA I4 - IZLAZ

      REAL*8 XMD, YMD, PMD, PMDD, PEXD
      REAL*4 XM, YM, PM, RM
      INTEGER XE, YE, PE, RE, IEXS, IGR
      IF(XM .EQ. 0. .OR. YM .EQ. 0.) THEN
                                     PM = 0.
                                     RM = 0.
                                     RETURN
                                     END IF
      PE = XE + YE
      PM = XM * YM
      PMDD = PM
      PE = PE + IEXS(PM)
      PM = PEXS(PM,0,IGR)
      XMD = XM
      YMD = YM
      PMD = XMD * YMD
      PMD = PMD - PMDD
      IF(PMD .EQ. 0.D0) THEN
                        RM = 0.
                        RETURN
                        END IF
      RE = XE + YE + IEXD(PMD)
      RM = PEXD(PMD,0,IGR)
      RETURN
      END
      SUBROUTINE TPXY(X,Y,P,R)
C POTPROGRAM ODREDJUJE TACAN PROIZVOD BROJEVA X, Y TIPA R4
C I PREDSTAVLJA PROIZVOD U OBLIKU APROKSIMACIJE P = X*Y I
C RAZLIKE TACNOG PROIZVODA I APROKSIMACIJE P, R

C X REALAN BROJ TIPA R4 - ULAZ
C Y REALAN BROJ TIPA R4 - ULAZ
C P REALAN BROJ TIPA R4 - IZLAZ
C R REALAN BROJ TIPA R4 - IZLAZ
```

```
      REAL*4 X, Y, P, R
      REAL*8 P1
      P = X*Y
      P1 = X
      P1 = P1*Y - P
      R = P1
      RETURN
      END
      SUBROUTINE TSXYP(XM,XE,YM,YE,SM,SE,RM,RE)
C POTPROGRAM ODREDJUJE TACNU SUMU BROJEVA X I Y KADA
C SU OVI DATI U OBLIKU MANTISA TIPA R4 I EKSPONENT
C TIPA I4. OVO JE VARIJANTA POTPROGRAMA TSXY.

C XM MANTISA X, TIPA R4 - ULAZ
C XE EKSPONENT X, TIPA I4 - ULAZ
C YM MANTISA Y, TIPA R4 - ULAZ
C YE EKSPONENT Y, TIPA I4 - ULAZ
C SM MANTISA S, TIPA R4 - IZLAZ
C SE EKSPONENT S, TIPA I4 - IZLAZ
C PM MANTISA P, TIPA R4 - IZLAZ
C RE EKSPONENT R, TIPA I4 - IZLAZ
      INTEGER XE, YE, SE, RE, L, D, IS, XE1, YE1
      REAL*4 XM, YM, SM, RM, X, Y, R, S
      DATA L/24/
      IF(XM .EQ. 0.) THEN
                     SM = YM
                     SE = YE
                     RM = 0.
                     RETURN
                     END IF
      IF(YM .EQ. 0.) THEN
                     SM = XM
                     SE = XE
                     RM = 0.
                     RETURN
                     END IF
      D = XE - YE
      IF(IABS(D) .GT. (L+1)) GO TO 10
      IS = (XE + YE) / 2
C TRANSLIRANJE EKSPONENTA DA BI SE OPERACIJE IZVELE
C SA RASPOLOZIVOM ARITMETIKOM POKRETNOG ZAREZA
      XE1 = XE - IS
      YE1 = YE - IS
      CALL SAST(XM,XE1,X,IGR)
      CALL SAST(YM,YE1,Y,IGR)
      CALL TSXY(X,Y,S,R)
      CALL RAZD(S,SM,SE)
      CALL RAZD(R,RM,RE)
      IF(S .EQ. 0.) RETURN
```

```
C VRACANJE NA STVARNE VREDNOSTI EKSPONENATA

      SE = SE + IS
      IF(R .NE. 0.) RE = RE + IS
      RETURN
10    IF(D .LT. 0) GO TO 20
      SM = XM
      SE = XE
      RM = YM
      RE = YE
      RETURN
20    SM = YM
      SE = YE
      RM = XM
      RE = XE
      RETURN
      END
      REAL*8 FUNCTION SLPRD(X,I)

C POTPROGRAM ODREDJUJE SLEDECI ILI PRETHODNI BROJ
C U POKRETNOM ZAREZU ZAVISNO DA LI JE I > 0 ILI
C I < 0 ZA REALAN BROJ X DUZINE 6 BAJTOVA

C X REALAN BROJ TIPA R8 - ULAZ
C I CEO BROJ TIPA I4 - ULAZ
C SLPRD REALAN BROJ TIPA R8 - IZLAZ

      REAL*8 X, X1, X2, ZA, EPS, CZ, PEXD
      INTEGER I, I1, IGR, EX
      LOGICAL TEST1, TEST2
      DATA L/48/, CZ/'80'X/
      IF(I .NE. 0) GO TO 10
      SLPRD = X
      RETURN
10    CONTINUE
      IF(X .NE. 0.D0) GO TO 20
      IF(I .GT. 0) SLPRD =  CZ
      IF(I .LT. 0) SLPRD = -CZ
      RETURN
20    CONTINUE
      TEST1 = X .EQ. -CZ .AND. I .GT. 0
      TEST2 = X .EQ.  CZ .AND. I .LT. 0
      IF(.NOT. TEST1) GO TO 30
      SLPRD = 0.D0
      RETURN
30    CONTINUE
      IF(.NOT. TEST2) GO TO 40
      SLPRD = 0.D0
      RETURN
40    CONTINUE
      EX = IEXD(X)
      X1 = PEXD(X,0,IGR)
      EPS = PEXD(0.D0,1-L,IGR)
      IF(I .GT. 0) X2 = X1 + EPS
```

```
      IF(I .LT. 0) X2 = X1 - EPS
      I1 = IEXD(X2) + EX
      SLPRD = PEXD(X2,I1,IGR)
      RETURN
      END
      SUBROUTINE ZA(X,XN,XD,XG)
C POTPROGRAM ZAOKRUZUJE X TIPA R8 NA MANTISU DUZINE
C 6 BAJTOVA DAJUCI XN - NAJBOLJA APROKSIMACIJA OD X,
C XD - X ZAOKRUZENO NA DOLE I XG - X ZAOKRUZENO NA
C GORE

C X REALAN BROJ TIPA R8 - ULAZ
C XN REALAN BROJ TIPA R8 - IZLAZ
C XD REALAN BROJ TIPA R8 - IZLAZ
C XG REALAN BROJ TIPA R8 - IZLAZ

      REAL*8 X, XN, XD, XG, X1, X2, Z, PEXD, EPS
      INTEGER I, I1, IGR, L, EX
      INTEGER*2 J
      LOGICAL*1 Y(8), YY(2)
      EQUIVALENCE (Z,Y(1)), (J,YY(1))
      DATA L/48/
      Z = X
      J = 0
      YY(1) = Y(7)
      Y(7) = 0
      XN = Z
      IF(J .EQ. 0) THEN
                   XD = Z
                   XG = Z
                   RETURN
                   END IF
      EX = IEXD(XN)
      X1 = PEXD(XN,0,IGR)
      EPS = PEXS(0.D0,1-L,IGR)
      IF(J .LT. 128 .AND. X .LT. 0.D0) THEN
                                       XG = X1
                                       XD = X1 - EPS
                                       XN = X1
                                       END IF
      IF(J .LT. 128 .AND. X .GT. 0.D0) THEN
                                       XG = X1 + EPS
                                       XD = X1
                                       XN = X1
                                       END IF
      IF(J .GE. 128 .AND. X .LT. 0.D0) THEN
                                       XG = X1
                                       XD = X1 - EPS
                                       XN = XD
                                       END IF
      IF(J .GE. 128 .AND. X .GT. 0.D0) THEN
                                       XG = X1 + EPS
                                       XD = X1
                                       XN = XD
                                       END IF
```

```
      XN = PEXD(XN,IEXD(XN)+EX,IGR)
      XD = PEXD(XD,IEXD(XD)+EX,IGR)
      XG = PEXD(XG,IEXD(XG)+EX,IGR)
      RETURN
      END
      SUBROUTINE OP(X,Y,I,Z,ZD,ZG)
C POTPROGRAM ZA ZADATO X I Y TIPA R8 U ZAVISNOSTI
C OD I ODREDJUJE: I = 1 NAJBOLJU APROKSIMACIJU ZBIRA
C Z I DONJU GRANICU ZD I GORNJU GRANICU ZG. ZA I = 2
C ODREDJUJE RAZLIKU A ZA I = 3 ODREDJUJE PROIZVOD.
C X, Y REALNI BROJEVI TIPA R8 - ULAZ
C I CEO BROJ TIPA I4 - ULAZ
C Z, ZD, ZG REALNI BROJEVI TIPA R8 - IZLAZ
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER I, D
      A = X
      B = Y
      IF(DABS(A) .LT. DABS(B)) THEN
                               C = A
                               A = B
                               B = C
                               END IF
      IF(I .EQ. 1) GO TO 10  ! SAB
      IF(I .EQ. 2) THEN      ! ODU
                   B = -B
                   GO TO 10
                   END IF
      IF(I .EQ. 3) GO TO 20  ! MNO
      TYPE 30, I
0     FORMAT(' I = ',I4,' NEDEFINISANA OPERACIJA')
      STOP
0     CONTINUE
      IF(A .EQ. 0.D0) THEN
                      Z  = B
                      ZD = Z
                      ZG = Z
                      RETURN
                      END IF
      IF(B .EQ. 0.D0) THEN
                      Z  = A
                      ZD = Z
                      ZG = Z
                      RETURN
                      END IF
      D = IEXD(A) - IEXD(B)
      IF(D .GT. (L+1))THEN
                      Z = A
                      IF(A .GT. 0.D0 .AND. B .GT. 0.D0) THEN
                                                        ZD = A
                                                        ZG = SLPRD(A,1)
                                                        RETURN
                                                        END IF
```

```
                    IF(A .GT. 0.D0 .AND. B .LT. 0.D0) THEN
                                                      ZD = SLPRD(A,-1)
                                                      ZG = A
                                                      RETURN
                                                      END IF
                    IF(A .LT. 0.D0 .AND. B .GT. 0.D0) THEN
                                                      ZD = A
                                                      ZG = SLPRD(A,1)
                                                      RETURN
                                                      END IF
                    IF(A .LT. 0.D0 .AND. B .LT. 0.D0) THEN
                                                      ZD = SLPRD(A,-1)
                                                      ZG = A
                                                      RETURN
                                                      END IF
                    ELSE
                    Z1 = A + B
                    CALL ZA(Z1,Z,ZD,ZG)
                    RETURN
                    END IF
      CONTINUE
      Z1 = A*B
      CALL ZA(Z1,Z,ZD,ZG)
      RETURN
      END
      SUBROUTINE TSXYD1(X,Y,S,R)

POTPROGRAM ODREDJUJE TACNU SUMU BROJEVA X, Y CIJA JE MANTISA
DUZINE 6 BAJTOVA I PREDSTAVLJA SUMU U OBLIKU APROKSIMACIJE
S = X + Y I RAZLIKE TACNE SUME I APROKSIMACIJE R, GDE SU S I
R SA MANTISAMA TAKODJE DUZINE 6 BAJTOVA. OVAJ SE POTPROGRAM
KORISTI PRI REALIZACIJI BOHLENDER-OVOG ALGORITMA.

X, Y REALNI BROJEVI TIPA R8 - ULAZ
S, R REALNI BROJEVI TIPA R8 - IZLAZ

      REAL*8 X, Y, S, R, Z, T, PEXD, Y1
      INTEGER D, L, IGR
      INTEGER*2 I
      LOGICAL*1 XY(8), YY(2)
      EQUIVALENCE (Z,XY(1)), (I,YY(1))
      DATA L/48/
      D = IEXD(X) - IEXD(Y)
      IF(IABS(D) .GT. (L+1)) GO TO 20
      Z = X + Y
      I = 0
      YY(1) = XY(7)
      XY(7) = 0
      S = Z
      IF(I .LT. 128) GO TO 10
      T = PEXD(0.5D0,IEXD(Z)-L+1,IGR)
      S = Z + T
      CONTINUE
      Y1 = S - X
```

```
      R = (Y - Y1) + (X - (S - Y1))
      RETURN
      IF(D .LT. 0) GO TO 30
      S = X
      R = Y
      RETURN
      S = Y
      R = X
      RETURN
      END
      SUBROUTINE BSUMA(X,N,S)

POTPROGRAM PREDSTAVLJA REALIZACIJU BOHLENDER-OVOG
ALGORITMA

X NIZ REALNIH BROJEVA DUZINE N, TIPA R8 - ULAZ
N DUZINA NIZA X, TIPA I4 - ULAZ
S REALAN BROJ, TIPA R4 - IZLAZ

      REAL*8 XD(200),SD,S1,R,MAX,MANT,MANT1,T1,Z,ZL,ZU,PEXD,Y1,Y2,Y3
      REAL*4 X(N), S
      INTEGER T, TNEW, L, M, ES, EXN, IGR
      DATA L/48/
      DO 5 I = 1,N
      XD(I) = X(I)
      IF(N.GT.0) GO TO 10
      S = 0.
      RETURN
      T = N-1
      CONTINUE


      MAX = 0.D0
      J = 0
      SD = 0.D0
      DO 30 I = 1,T-1
      CALL TSXYD1(SD,XD(I),S1,R)
      IF(R .EQ. 0.D0) GO TO 25
      J = J + 1
      XD(J) = R
      IF(DABS(XD(J)) .GT. MAX) MAX = DABS(XD(J))
      SD = S1
      CONTINUE
      TNEW = J + 1


      DO 40 I = T,N
      CALL TSXYD1(SD,XD(I),S1,R)
      IF(R .EQ. 0.D0) GO TO 35
      J = J + 1
      XD(J) = R
      SD = S1
      CONTINUE
      T = TNEW
```

```
      IF(SD .NE. 0.D0) GO TO 50
      N = J
      GO TO 60
)     N = J + 1


      XD(N) = SD
)     CONTINUE
      IF(N.GT.1) GO TO 70
      IF(N .EQ. 0) S = 0.
      IF(N .NE. 0) S = XD(1)
      RETURN
)     IF(T .NE. 1) GO TO 90
      MANT1 = DINT(PEXD(XD(N),L+1,IGR))*PEXD(0.D0,-L,IGR)
      MANT  = PEXD(XD(N),0,IGR)
      IF(MANT1 .NE. MANT) GO TO 80
      M = IEXD(XD(N))
      S = XD(N) + DSIGN(1.D0,XD(N-1))*PEXD(0.D0,-2*L+1,IGR)*
     $     PEXD(0.D0,M+1,IGR)
      RETURN
)     S = XD(N)
      RETURN
)     T1 = T-1
      CALL OP(T1,MAX,3,Z,ZL,R)
      SD = R
      DO 100 I = T,N
      CALL OP(SD,XD(I),1,Z,ZL,SD)
)0    CONTINUE
      ES = IEXD(SD)
      EXN = IEXD(XD(N))
      Y1 = DINT(PEXD(SD,L+1,IGR))
      Y2 = DINT(PEXD(XD(N),L+1,IGR))
      Y3 = PEXD(SD,L+1,IGR)
      IF(ES.EQ.EXN .AND. DSIGN(1.D0,SD).EQ.DSIGN(1.D0,XD(N)) .AND.
     $Y1.EQ.Y2 .AND. Y2.NE.Y3) GO TO 110


      GO TO 20
:0    SD = -R
      DO 120 I = T,N
:0    CALL OP(SD,XD(I),1,Z,SD,ZU)
      ES = IEXD(SD)
      EXN = IEXD(XD(N))
      Y1 = DINT(PEXD(SD,L+1,IGR))
      Y2 = DINT(PEXD(XD(N),L+1,IGR))
      Y3 = PEXD(SD,L+1,IGR)
      IF(ES.EQ.EXN .AND. DSIGN(1.D0,SD).EQ.DSIGN(1.D0,XD(N)) .AND.
     $Y1.EQ.Y2 .AND. Y2.NE.Y3) GO TO 130


      GO TO 20
;0    S = XD(N)
      RETURN
      END
```

```
      SUBROUTINE TMNB(X,Y,N,XY,K)
C POTPROGRAM MNOZI NIZ X DUZINE N REALNIM BROJEM
C Y I PREDSTAVLJA PROIZVOD U OBLIKU NIZA XY DUZINE K

C X NIZ REALNIH BROJEVA DUZINE N, TIPA R4 - ULAZ
C Y REALAN BROJ TIPA R4 - ULAZ
C N CEO BROJ TIPA I4 - ULAZ
C XY NIZ REALNIH BROJEVA DUZINE K, TIPA R4 - IZLAZ
C K CEO BROJ TIPA I4 - IZLAZ

      DIMENSION X(N), XY(200), Z(200)
      IF(N .LE. 0) THEN
                   TYPE 5, N
5                  FORMAT(' N <= 0 U POTPROGRAMU TMNB ', I10)
                   RETURN
                   END IF
      IF(Y .EQ. 0.) THEN
                    K = 1
                    XY(1) = 0.
                    RETURN
                    END IF
      DO 10 I = 1,N
      CALL TPXY(X(I),Y,Z(2*I-1),Z(2*I))
0     CONTINUE
      CALL TSUMA(Z,2*N,K)
      DO 20 I = 1,K
0     XY(I) = Z(I)
      RETURN
      END
      SUBROUTINE TMNBP(XM,XE,YM,YE,N,XYM,XYE,K)

  POTPROGRAM PREDSTAVLJA VARIJANTU POTPROGRAMA TMNB
  GDE SU ODGOVARAJUCI ARGUMENTI DATI U OBLIKU MANTISA
  I EKSPONENT

  XM NIZ REALNIH BROJEVA DUZINE N, TIPA R4 - ULAZ
  XE NIZ EKSPONENATA DUZINE N, TIPA I4 - ULAZ
  YM REALAN BROJ TIPA R4 - ULAZ
  YE CEO BROJ TIPA I4 - ULAZ
  N CEO BROJ TIPA I4 - ULAZ
  XYM NIZ REALNIH BROJEVA DUZINE K, TIPA R4 - IZLAZ
  XYE NIZ EKSPONENATA DUZINE N, TIPA I4 - IZLAZ
  K CEO BROJ TIPA I4 - IZLAZ

      REAL*4 XM(N), XYM(200), ZM(200)
      INTEGER XE(N), XYE(200), ZE(200), YE
      IF(N .LE. 0) THEN
                   TYPE 5, N
                   FORMAT(' N <= 0 U POTPROGRAMU TMNBP ', I10)
                   RETURN
                   END IF
      IF(YM .EQ. 0.) THEN
                    K = 1
                    XYM(1) = 0.
                    RETURN
                    END IF
```

```
      DO 10 I = 1,N
      CALL TPXYP(XM(I),XE(I),YM,YE,ZM(2*I-1),ZE(2*I-1),ZM(2*I),ZE(2*I))
0     CONTINUE
      CALL TSUMAP(ZM,ZE,2*N,K)
      DO 20 I = 1,K
      XYM(I) = ZM(I)
0     XYE(I) = ZE(I)
      RETURN
      END
      SUBROUTINE POLI(X,A,N,B,K)

  POTPROGRAM IZRACUNAVA VREDNOST POLINOMA CIJI SU
  KOEFICIJENTI DATI U OPADAJUCEM NIZU U ODNOSU NA
  VREDNOST STEPENA KOD ODGOVARAJUCEG SABIRKA U
  POLINOMU. NAPR. A(0) JE SLOBODAN CLAN A A(N) JE
  KOEFICIJENT UZ X**N. X JE ARGUMENT A VREDNOST
  POLINOMA JE PREDSTAVLJENA U OBLIKU NIZA B DUZINE K

  X REALAN BROJ TIPA R4 - ULAZ
  A NIZ KOEFICIJENATA DUZINE N+1, TIPA R4 - ULAZ
  N CEO BROJ TIPA I4 - ULAZ
  B NIZ REALNIH BROJEVA DUZINE K, TIPA R4 - IZLAZ
  K CEO BROJ TIPA I4 - IZLAZ

      REAL*4 X, A(0:N), B(200), BM(200), WM(200)
      INTEGER XE, BE(200), WE(200)
      IF( N .LT. 0) THEN
                     TYPE 10, N
0                    FORMAT(' N NEGATIVNO U POTPROGRAMU POLI',I10)
                     RETURN
                     END IF
      K = 1
      B(1) = A(N)
      IF(N .EQ. 0) RETURN
      CALL RAZD(X,XM,XE)
      CALL RAZD(B(1),BM(1),BE(1))
      DO 20 I = N,1,-1
      DO 30 J = 1,K
      CALL TPXYP(XM,XE,BM(J),BE(J),WM(2*J-1),WE(2*J-1),WM(2*J),WE(2*J))
0     CONTINUE
      CALL RAZD(A(I-1),WM(2*K+1),WE(2*K+1))
      CALL TSUMAP(WM,WE,2*K+1,KK)
      DO 40 J = 1,KK
      BM(J) = WM(J)
0     BE(J) = WE(J)
      K = KK
0     CONTINUE
      CALL SASTN(BM,BE,B,KK,K)
      RETURN
      END
```

```
      SUBROUTINE NULA(A,N,AA,BB,A1,B1,C)

POTPROGRAM NALAZI NULU POLINOMA CIJI SU
KOEFICIJENTI PREDSTAVLJENI NIZOM A I CIJI
JE STEPEN N NA INTERVALU [AA,BB]. UKOLIKO
NULA NIJE TACNO PREDSTAVLJIVA U DATOM
SISTEMU SA POKRETNIM ZAREZOM, POTPROGRAM
ODREDJUJE NAJUZI INTERVAL [A1,B1] KOJI
SADRZI NULU A PREDSTAVLJIV JE U DATOM
SISTEMU, I NAJBOLJU APROKSIMACIJU NULE C.

A NIZ KOEFICIJENATA DUZINE N+1, TIPA R4 - ULAZ
N CEO BROJ TIPA I4 - ULAZ
AA , BB SU LEVA I DESNA GRANICA INTERVALA NA
KOME SE NALAZI NULA, TIPA R4 - ULAZ
A1 , B1 SU LEVA I DESNA GRANICA NAJUZEG INTERVALA
KOJI SADRZI NULU, TIPA R4 - IZLAZ
C JE NAJBOLJA APROKSIMACIJA NULE, TIPA R4 - IZLAZ

      REAL*4 A(0:N), V(200)
      A1 = AA
      B1 = BB
      CALL POLI(A1,A,N,V,K)
      T1 = V(1)
      CALL POLI(B1,A,N,V,K)
      T2 = V(1)
      S = SIGN(1.,T1)*SIGN(1.,T2)
      IF(S .GT. 0.) THEN
                    TYPE 10, A1, B1, T1, T2
                    FORMAT(' NEMA NULU NA [',E15.7,',',E15.7,']',2E15.7)
                    RETURN
                    END IF
      C = (A1 + B1)/2.
      IF(C .EQ. A1 .OR. C .EQ. B1) RETURN
      CALL POLI(C,A,N,V,K)
      T2 = V(1)
      S = SIGN(1.,T1)*SIGN(1.,T2)
      IF(S .EQ. 0.) THEN
                    A1 = C
                    B1 = C
                    RETURN
                    END IF
      IF(S .LT. 0.) THEN
                    B1 = C
                    ELSE
                    A1 = C
                    END IF
      GO TO 20
      END
```

```
      SUBROUTINE TM2NP(XM,XE,N,YM,YE,M,XYM,XYE,K,IT)

POTPROGRAM ODREDJUJE TACAN PROIZVOD NIZOVA X
DUZINE N I Y DUZINE M KOJI SU PREDSTAVLJENI
U PROSIRENOM OBLIKU (XM,XE) I (YM,YE) DAJUCI
PROIZVOD XY DUZINE K TAKODJE U PROSIRENOM OBLIKU.

XM NIZ MANTISA DUZINE N NIZA X, TIPA R4 - ULAZ
XE NIZ EKSPONENATA DUZINE N NIZA X,TIPA I4 - ULAZ
N DUZINA XM I XE, TIPA I4 - ULAZ
YM NIZ MANTISA DUZINE M NIZA Y, TIPA R4 - ULAZ
YE NIZ EKSPONENATA DUZINE M NIZA Y, TIPA I4 - ULAZ
M DUZINA YM I YE, TIPA I4 - ULAZ
XYM NIZ MANTISA DUZINE K NIZA XY, TIPA R4  - IZLAZ
XYE NIZ EKSPONENATA DUZINE K NIZA XY, TIPA I4 - IZLAZ
K DUZINA XYM I XYE, TIPA I4 - IZLAZ
IT PARAMETAR TACNOSTI: IT = 1 OPERACIJA TACNA
                       IT = 0 OPERACIJA ZAOKRUZENA NA K
                              BROJEVA U POKRETNOM ZAREZU

      REAL*4 XM(N), YM(M), XYM(200), ZM(200), ZM1(200)
      INTEGER XE(N), YE(M), XYE(200), ZE(200), ZE1(200)
      IF(N .LE. 0 .OR. M .LE. 0) THEN
                                  TYPE 10, N, M
                                  FORMAT(' N ILI M NEGATIVNO',2I10)
                                  RETURN
                                  END IF
      IF(XM(1) .EQ. 0. .OR. YM(1) .EQ. 0.) THEN
                                                K = 1
                                                XYM(1) = 0.
                                                RETURN
                                                END IF
      CALL TMNBP(YM,YE,XM(1),XE(1),M,XYM,XYE,L)
      IF(N .EQ. 1) GO TO 40
      DO 20 I = 2,N
      DO 30 J = 1,L
      ZM(J) = XYM(J)
      ZE(J) = XYE(J)
      CONTINUE
      CALL TMNBP(YM,YE,XM(I),XE(I),M,ZM1,ZE1,K1)
      CALL S2NP(ZM,ZE,L,ZM1,ZE1,K1,XYM,XYE,KK)
      CALL TSUMAP(XYM,XYE,KK,L)
      CONTINUE
      CONTINUE
      IF(IT .EQ. 1) THEN
                    K = L
                    RETURN
                    ELSE
                    IF(K .GT. L) K = L
                    RETURN
                    END IF
      RETURN
      END
```

```
      SUBROUTINE RECNP(XM,XE,N,YM,YE,M)
C
C POTPROGRAM ODREDJUJE RECIPROCNU VREDNOST NIZA
C X DUZINE N KOJI JE PREDSTAVLJEN U PROSIRENOM
C OBLIKU (XM,XE) DAJUCI RECIPROCNU VREDNOST U
C OBLIKU NIZA Y DUZINE M U PROSIRENOM OBLIKU (YM,YE).
C
C XM NIZ MANTISA DUZINE N NIZA X, TIPA R4 - ULAZ
C XE NIZ EKSPONENATA DUZINE N NIZA X, TIPA I4 - ULAZ
C N DUZINA XM I  XE, TIPA I4 - ULAZ
C YM NIZ MANTISA DUZINE M NIZA Y, TIPA R4 - IZLAZ
C YE NIZ EKSPONENATA DUZINE M NIZA Y, TIPA I4 - IZLAZ
C M DUZINA YM I YE, TIPA I4 - IZLAZ
C
      REAL*4 XM(N),YM(200),YYM(200),ZM(200)
      INTEGER XE(N),YE(200),YYE(200),ZE(200)
      IF(N .LE. 0) THEN
                    TYPE 10, N
10                  FORMAT(' N <= 0 U POTPROGRAMU RECNP',I10)
                    RETURN
                    END IF
      IF(XM(1) .EQ. 0.) THEN
                         TYPE 20, XM(1)
20                       FORMAT(' X(1) = 0 U POTPROGRAMU RECNP')
                         RETURN
                         END IF
      Y1 = 1./XM(1)
      YM(1) = PEXS(Y1,0,IGR)
      YE(1) = IEXS(Y1) - XE(1)
      I = 0
30    I = I + 1
      L = I
      IF(I .GT. N) L = N
      CALL TM2NP(XM,XE,L,YM,YE,I,ZM,ZE,K,1)
      YYM(1) = 0.5
      YYE(1) = 2
      DO 40 J = 2,K+1
      YYM(J) = -ZM(J-1)
40    YYE(J) =  ZE(J-1)
      CALL TSUMAP(YYM,YYE,K+1,KK)
      CALL TM2NP(YM,YE,I,YYM,YYE,I,ZM,ZE,K,1)
      IF(I .GT. M) THEN
                    DO 50 J = 1,M
                    IF(YM(J).NE.ZM(J) .OR. YE(J).NE.ZE(J)) GO TO 60
50                  CONTINUE
                    RETURN
                    END IF
60    CONTINUE
      DO 70 J = 1,MIN(I,M+1)
      YM(J) = ZM(J)
70    YE(J) = ZE(J)
      GO TO 30
      END
```

```
      SUBROUTINE S2NP(XM,XE,N,YM,YE,M,ZM,ZE,K)

 POTPROGRAM SPAJA NIZOVE X DUZINE N I Y DUZINE M
 KOJI SU PREDSTAVLJENI U PROSIRENIM OBLICIMA (XM,XE)
 I (YM,YE) DAJUCI NIZ Z DUZINE K KOJI JE SORTIRAN
 U OPADAJUCI NIZ U ODNOSU NA EKSPONENTE. Z JE TAKODJE
 PREDSTAVLJEN U PROSIRENOM OBLIKU (ZM,ZE).
 KORISTI SE ZA BRZE IZVRSAVANJE TSUMAP.

 XM NIZ MANTISA DUZINE N NIZA X, TIPA R4 - ULAZ
 XE NIZ EKSPONENATA DUZINE N NIZA X,TIPA I4 - ULAZ
 N DUZINA XM I XE, TIPA I4 - ULAZ
 YM NIZ MANTISA DUZINE M NIZA Y, TIPA R4 - ULAZ
 YE NIZ EKSPONENATA DUZINE M NIZA Y, TIPA I4 - ULAZ
 M DUZINA YM I YE, TIPA I4 - ULAZ
 XYM NIZ MANTISA DUZINE K NIZA XY, TIPA R4  - IZLAZ
 XYE NIZ EKSPONENATA DUZINE K NIZA XY, TIPA I4 - IZLAZ
 K DUZINA XYM I XYE, TIPA I4 - IZLAZ

      REAL XM(N), YM(M), ZM(200)
      INTEGER XE(N), YE(M), ZE(200)
      IF(N .EQ. 0 .OR. M .EQ. 0) THEN
                                 TYPE 10, N, M
0                                FORMAT(' N ILI M <= 0 U S2NP', 2I10)
                                 RETURN
                                 END IF
      IX = 1
      IY = 1
      IZ = 1
      K = N+M
0     CONTINUE
      IF(IX .GT. N) THEN
                   DO 30 I = IZ,N+M
                   I1 = I - IZ + IY
                   ZM(I) = YM(I1)
                   ZE(I) = YE(I1)
0                  CONTINUE
                   RETURN
                   END IF
      IF(IY .GT. M) THEN
                   DO 40 I = IZ,N+M
                   I1 = I - IZ + IX
                   ZM(I) = XM(I1)
                   ZE(I) = XE(I1)
0                  CONTINUE
                   RETURN
                   END IF
      IF(XE(IX) .GT. YE(IY)) THEN
                             ZM(IZ) = XM(IX)
                             ZE(IZ) = XE(IX)
                             IZ = IZ + 1
                             IX = IX + 1
                             ELSE
                             ZM(IZ) = YM(IY)
```

```
                              ZE(IZ) = YE(IY)
                              IZ = IZ + 1
                              IY = IY + 1
                              END IF
      GO TO 20
      END
      SUBROUTINE D2NP(XM,XE,N,YM,YE,M,ZM,ZE,K,IT)
      REAL XM(N), YM(M), ZM(200)
      REAL AM(200), BM(200), RM(200), ZZM(200)
      INTEGER XE(N), YE(M), ZE(200)
      INTEGER AE(200), BE(200), RE(200), ZZE(200)

C POTPROGRAM ODREDJUJE KOLICNIK NIZOVA X DUZINE N
C I Y DUZINE M KOJI SU PREDSTAVLJENI U PROSIRENIM
C OBLICIMA (XM,XE) I (YM,YE) RESPEKTIVNO, DAJUCI
C KOLICNIK U OBLIKU NIZA Z TAKODJE U PROSIRENOM
C OBLIKU

C XM NIZ MANTISA DUZINE N NIZA X, TIPA R4 - ULAZ
C XE NIZ EKSPONENATA DUZINE N NIZA X, TIPA I4 - ULAZ
C N DUZINA XM I XE, TIPA I4 - ULAZ
C YM NIZ MANTISA DUZINE M NIZA Y, TIPA R4 - ULAZ
C YE NIZ EKSPONENATA DUZINE M NIZA Y, TIPA I4 - ULAZ
C M DUZINA YM I YE, TIPA I4 - ULAZ
C ZM NIZ MANTISA DUZINE K NIZA Z, TIPA R4 - IZLAZ
C ZE NIZ EKSPONENATA DUZINE K NIZA Z, TIPA I4 - IZLAZ
C K DUZINA ZM I ZE, TIPA I4 - IZLAZ
C IT PARAMETAR TACNOSTI: IT = 1 OPERACIJA TACNA
C                        IT = 0 OPERACIJA ZAOKRUZENA NA K
C                               BROJEVA U POKRETNOM ZAREZU

      IF(N .LE. 0 .OR. M .LE. 0) THEN
                                  TYPE 10, N, M
10                                FORMAT(' N ILI M <= 0 U D2NP')
                                  RETURN
                                  END IF
      IF(YM(1) .EQ. 0.) THEN
                         TYPE 15
15                       FORMAT(' DELJENJE NULOM')
                         RETURN
                         END IF
      IF(IT .EQ. 1) K = N+M
      DO 20 I = 1,N
      AM(I) = XM(I)
20    AE(I) = XE(I)
      K3 = N
      I = 0
30    I = I + 1
      Q = AM(1)/YM(1)
      ZZM(I) = PEXS(Q,0,IGR)
      ZZE(I) = IEXS(Q) + (AE(1) - YE(1))
      CALL TMNBP(YM,YE,ZZM(I),ZZE(I),M,BM,BE,K1)
      DO 40 J = 1,K1
40    BM(J) = -BM(J)
```

```
      CALL S2NP(AM,AE,K3,BM,BE,K1,RM,RE,K2)
      CALL TSUMAP(RM,RE,K2,K3)
      IF(RM(1) .EQ. 0.) THEN

                               CALL TSUMAP(ZZM,ZZE,I,L)
                               K = L
                               DO 50 J = 1,K
                               ZM(J) = ZZM(J)
50                             ZE(J) = ZZE(J)
                               RETURN
                                                            END IF
      IF(I .GT. (K+1)) THEN
                             CALL TSUMAP(ZZM,ZZE,I,L)
                             IF(L .GT. K) THEN
                                          DO 60 J = 1,K
                                          ZM(J) = ZZM(J)
60                                        ZE(J) = ZZE(J)
                                          RETURN
                                          ELSE
                                          I = L
                                          END IF
                             END IF
      DO 70 J = 1,K3
      AM(J) = RM(J)
70    AE(J) = RE(J)
      GO TO 30
      END
```