

MATHEMATISCH CENTRUM

2e BOERHAAVESTRAAT 49

AMSTERDAM

REKENAFDELING

"DRIJVENDE-KOMMA"-REKENTECHNIEK

(ARRA-subroutines Rd1 en Rd2)

door

E.W. Dijkstra.

MR 16

1 9 5 4

BIBLIOTHEEK MATHEMATISCH CENTRUM
AMSTERDAM

CWI BIBLIOTHEEK

3 0054 00077 3102

VOORWOORD.

In dit rapport worden de subroutines voor "Drijvende Komma"-rekentechniek beschreven, zoals deze medio 1954 ontstaan zijn.

Omdat de normering in de verschillende gevallen zoveel korter of langer duren kan, is niet bij elke aanroep de tijdsduur aangegeven. Op grond van ervaringen in Juli '54 met Rd1 opgedaan, mag men rekenen op een vertraging van 40 a 50 maal. Omdat kanaalwisselingen zeer veel optreden, mogen we verwachten, dat deze verhouding niet onbeduidend gunstiger zal komen te liggen, zodra de relais uit de selectie verdwenen zullen zijn.

Rd2 in zijn huidige vorm - met name de service-faciliteiten - is later ontstaan, naar aanleiding van de ervaringen, die we bij het werkelijk gebruik opdeden.

September 1954.

Subroutine voor "Drijvende Komma"-rekentechniek.

2+2 kanalen.

```

-----
Voorponsing:  A   1004   XO
                A   1004   XO
                +     K     X
                +     e     X
                +     f     X
-----

```

```

-----
1)      0           XO      2) -       X
        7      1     X1      15 maal X
-----

```

als K (zie onder) de orde van grootte van het kleinste en het grootste door Rd1 hanteerbare getal aangeeft;

als e het beginadres is van het eerste der twee opeenvolgende gemengde kanalen, die door Rd1 bezet worden;

als f het beginadres is van het eerste der twee opeenvolgende opdracht-kanalen, die door Rd1 bezet worden.

In de standaard representatie voor de "drijvende-komma"-rekentechniek wordt ieder getal in eerste instantie voorgesteld door $p \cdot 10^q$, waar $0,1 \leq |p| < 1$ is (in feite geldt, tengevolge van binaire afrondingen $0,1 < |p| < 1$), en $-K \leq q < 2^{14} - K$. (De mogelijkheid, K te kunnen variëren is misschien wel wat overdreven; $K = 2^{13}$ zal voor de meeste problemen de prettigste keuze zijn).

Bij het gebruik van Rd1 worden steeds twee opeenvolgende adressen voor de berging van de getallen p en q gebruikt; deze worden, als dit adressen h en h+1 zijn, geborgen volgens de conventie

$$\begin{aligned} \{h\} &= \frac{1}{2} p \\ \{h+1\} &= (q+K)2^{15} + h \text{ (of } \langle h+1 \rangle = 0/q+K; 0/h) . \end{aligned}$$

Opm.1. Het breukgedeelte staat dus in het geheugen als breuk, die in absolute waarde kleiner is dan $\frac{1}{2}$ en "dienst doet" voor het tweevoud. De factor $\frac{1}{2}$ bij het breukgedeelte is een conventie, waarvan men zich slechts bij de in- en uitvoer van getallen bewust hoeft te zijn.

Opm.2. De routines van Rd1, die doorgaans in reeksen achter elkaar uitgevoerd zullen worden, hebben alle dezelfde koppelopdracht; voordat deze aan het einde van een bewerking gehoorzaamd wordt, wordt deszelfs adres elke keer met 1 vermeerderd. Dit betekent, dat niet steeds 24/6 in het hoofdprogramma hoeft te staan, omdat, tenzij men ingrijpt, steeds de besturing twee opdrachten verder in het hoofdprogramma terugkomt. (De koppelopdracht vervult de functie van

"drijvende-komma"-opdrachtenteller!)

Opm.3. Twee werkruimte-adressen in Rd1 fungeren tezamen als quasi-accumulator "A". De rekenkundige bewerkingen verwerken steeds "A" en laten het antwoord steeds in "A" achter.

Opm.4. Elke aanroep heeft de functie van een opdracht in een een-adres code: het functiegedeelte wordt gespecificeerd door de plaats in Rd1, waar de besturing naar verwezen wordt, het adres-gedeelte wordt gespeeld door een programma-parameter, die in S wordt meegegeven; deze programma-parameter is de assemblage van de exponent en het adres van het breukgedeelte (dus (h+1), zie boven). In de quasi-accumulator "A" is de exponent van deze assemblage bevrijd.

Korte beschrijving der aanroepen.

Administratieve aanroepen.

("2-opdracht") =) 10/h+1
 7/2+f =)
 =)

Functie: Het door (h) en (h+1) gekarakteriseerde getal wordt in de quasi-accumulator "A" geplaatst. De assemblage van q+k en h (uit adres h+1 via S!) wordt ontrafeld: bij terugkomst in het hoofdprogramma is aan de inhoud van "A" niet te zien, dat deze uit h en h+1 gekomen is.

("4-opdracht") =) 10/h+1
 7/4+f =)
 =)

Functie: Het door de quasi-accumulator bevatte getal wordt in de gewenste representatie - d.w.z. na assemblage van de exponent q+K en het getal h - op h en h+1 weggeborgen. De inhoud van "A" blijft ongewijzigd!

Opm. Boven gegeven aanroep is slechts dan toegestaan, als in de a-helft van <h+1> zich 0/h bevindt, terwijl verder (h+1) positief dient te zijn. De programma-parameter mag ook wel ergens anders vandaan gehaald worden! ([S'] = +h voldoet). Dit is de enige aanroep, waarbij van de programma-parameter slechts de lage helft (en teken-cijfer) gebruikt worden.

Additieve bewerkingen.

Dit zijn de enige operaties, waartoe de aanroep verwijst naar het e-gedeelte der routine (de gemengde kanalen)

("0-opdracht") =) $10/h+1$
 $7/o+e$ =)
 =)

Functie: De inhoud van de quasi-accumulator wordt met het door (h) en (h+1) omschreven getal vermeerderd.

("1-opdracht") =) $10/h+1$
 $7/1+e$ =)
 =)

Functie: De inhoud van de quasi-accumulator wordt met het door (h) en (h+1) omschreven getal verminderd.

Opm.1. Na afloop van de berekening is het antwoord in "A" genormeerd d.w.z. het breukgedeelte ligt weer tussen 0,1 en 1, de exponent heeft de daarbij horende waarde.

Opm.2. Als het antwoord in absolute waarde te klein is, om door Rd1 nog gehanteerd te kunnen worden, wordt als antwoord in "A" achtergelaten het kleinste positieve getal, dat nog wel hanteerbaar is, nl. $0,1 \times 10^{-K} = 10^{-(K+1)}$.

Multiplicatieve bewerkingen.

("18-opdracht") =) $10/h+1$
 $7/18+f$ =)
 =)

Functie: De inhoud van de quasi-accumulator "A" wordt vermenigvuldigd met het getal door (h) en (h+1) omschreven.

("19-opdracht") =) $10/h+1$
 $7/19+f$ =)
 =)

Functie: De inhoud van de quasi-accumulator "A" wordt vermenigvuldigd met -1 maal het getal, door (h) en (h+1) omschreven.

("20-opdracht") =) $10/h+1$
 $7/20+f$ =)
 =)

Functie: De inhoud van de quasi-accumulator "A" wordt gedeeld door het getal, door (h) en (h+1) omschreven.

("21-opdracht") =) $10/h+1$
 $7/21+f$ =)
 =)

Functie: De inhoud van de quasi-accumulator "A" wordt gedeeld door -1 maal het getal, door (h) en (h+1) omschreven.

Opm. Deze routines laten het antwoord genormeerd in "A" achter. Als het antwoord in absolute waarde te klein zou zijn, om verder door Rd1 gehanteerd te kunnen worden, wordt het, ongeacht het teken der factoren, vervangen door $0,1 \times 10^{-K} = 10^{-(K+1)}$.

Controle op de constanten in Rd1.

Voor controle op de in Rd1 voorkomende constanten kent Rd1 een speciale aanroep. Deze aanroep maakt van geen programmaparameter gebruik; wat betreft zijn koppelopdracht gedraagt de controle zich als alle andere onder-routines van Rd1. Dit impliceert een "loze opdracht" in het hoofdprogramma.

Controle-aanroep: =) $7/0+f$ =)
 $24/0$ loze opdracht
 =)

Functie: Na deze aanroep worden de in Rd1 voorkomende numerieke en administratieve constanten aan een som-controle onderworpen. In geval van ongerechtigdheden stopt de machine òf op $24/2$ in $34a+e$ ($\equiv 2a + (e+32)$) òf op $24/3$ in $54a+e$ ($\equiv 22a + (e+32)$). Wordt niets bedenkelijks gesignaleerd, dan komt de besturing "gewoon" - d.w.z. 2 opdrachten verder - in het hoofdprogramma terug.

De voorbereiding van een reeks aanroepen.

Het is duidelijk, dat, wanneer de routines niet elk hun eigen koppelopdracht meekrijgen, maar steeds die van de vorige, nadat deszelfs adres met 1 vermeerderd is, gebruiken, een speciale voorzorg nodig is, om een reeks aanroepen van Rd1 in te leiden. Men dient hiervoor te weten dat de gemeenschappelijke koppelopdracht zich bevindt op de laatste plaats der twee gemengde kanalen. Bij de inloop van reeks aanroepen van Rd1 vulle het hoofdprogramma de "min-eerste" koppelopdracht in. Als de sluitletter E het adres met e (het begin-adres der gemengde kanalen) vermeerderd, luidt de inloop combinatie:

24	6	X0
4	31	E1

Ter illustratie van het bovenstaande het volgende voorbeeld: Gevraagd wordt, x door $x^3 - x^2$ te vervangen, als x zich bevindt op h en h+1.

We nemen aan:

- 1) dat deze bewerking plaats moet vinden, vlak nadat de machine "direct" gewerkt heeft, (bv. in telling of administratie); dit ter illustratie van de inloopcombinatie.
- 2) dat het gewenst is op dit moment de constanten te controleren; dit ter illustratie van de loze opdracht.
- 3) dat vermeerdering van de adressen met e, resp. f door de sluitletters E, resp. F bewerkstelligd wordt.
- 4) dat het aanroepen-reeksje bv. begint op y'a (dit had ook een b-opdracht kunnen zijn; in dat geval zou de gemeenschappelijke koppelopdracht steeds een 15-opdracht geweest zijn).

	y	24	6	X0	(31aE1 wordt nu 7/1+y; wordt <u>niet</u>
		4	31	E1	als zodanig gehoorzaamd!)
	1+y	7		FO =)	controle; zo goed 7/2+y in 31aE1
		24		X0	"loze opdracht"
zo goed =)	2+y	10	h+1	X0	x in quasi accumulator "A";
		7	2	FO =)	7/3+y $\geq$ 7/2+y in 31 a E1
=)	3+y	10	h+1	X0	
		7	18	FO =)	$x^2 \geq$ "A"
=)	4+y	10	h+1	X0	
		7	1	EO =)	$x^2 - x \geq$ "A"
=)	5+y	10	h+1	X0	
		7	18	FO =)	$x^3 - x^2 \geq$ "A"
=)	6+y	10	h+1	X0	$x^3 - x^2$ wordt op
		7	4	FO =)	h en h+1 weggeschreven
=)	7+y				

Voor de in- en uitvoer van drijvende-kommagetallen wordt naar andere routines uit de Rd-serie verwezen.

Subroutine Rd2: In- en uitvoer; service ten dienste van Rd1.

1+1 kanaal.

```

-----
Voorponsing:  A   1004   XO           Dit is dezelfde
                A   1004   XO           voorponsing
                +     K     X           als van
                +     e     X           Rd1.
                +     f     X
-----

```

```

1)      0           XO      2)  -
      7      1      X1      15 maal X

```

Zie voor explicatie bij de voorponsing van Rd1.

Rd2 beslaat 1 gemengd kanaal (64+e-95+e), en 1 zuiver opdrachten-
kanaal. (64+f - 95+f).

Voor de beschrijving der aanroepen wordt aangenomen, dat tgv. de voorponsing van het hoofdprogramma de sluitletters E resp. F het adres met e, resp. f vermeerderen. In die zin beslaat Rd2 dus de kanalen E2 en F2 (het gemengde kanaal van Rd2 volgend op de twee gemengde kanalen van Rd1, etc.).

Rd2 bevat 3 onderoutines, die van precies dezelfde aard zijn als de onderoutines van Rd1, d.w.z. zij delen met Rd1 de gemeenschappelijke, zichzelf ophogende koppelopdracht, zij worden dus ook - zo nodig - door

```

      24      6      XO
      4      31      E1   ingeleid.

```

In wijze van aanroepen zijn zij dus niet van de onderoutines van Rd1 te onderscheiden. Deze aanroepen zullen eerste beschreven worden.

Het bandlezen van "drijvende-komma"-getallen.

1ste methode. Hierbij wordt geen programmaparameter meegegeven; dientengevolge sta in het hoofdprogramma een loze opdracht.

```

Aanroep:  =)  24           XO
              7      17      F2  =)
              =)  . . . . .

```

Functie: Als op de band geponst staat,

```

+   h   X
+.  p   X
+   q   X

```

wordt dit drietal gelezen, en de $\pm p \times 10^{+q}$ wordt op h en h+1 geborgen volgens de conventies van Rd1.

2de methode. De eerste twee opdrachten van deze onderroutine roepen het invoerprogramma als subroutine aan, om h te lezen: de besturing komt dan met -h in de accumulator. Dit suggereert een tweede gebruik:

Aanroep: =) 3 k X0 als [k] = h
 7 18 F2 =)
 =)

Functie: Als boven, mits op de band +h X wordt weggelaten.

Opm. Hier wordt het invoerprogramma 3 (2) maal als subroutine aangeroepen. Dit impliceert o.a.:

- 1^e. Dat extra X normaal geskipt wordt;
- 2^e. Dat het invoerprogramma direct (en wel schrijvend) gaat werken, zodra een controlecombinatie op de band ontmoet wordt.

Typen en desassemblage.

Als de inhoud van h en h+1 uitgetypt dient te worden, sta in het hoofdprogramma het tweetal aanroepen:

Aanroep: na fout -+ =) 10 h+1 X0
 7 E2 =) typt
 =) 10 h+1 X0
 7 4 E2 =) controleert
 =)

Functie: Getypt wordt: breukgedeelte met teken, twee spaties, dan de macht, met teken.

Opm.1. Hier wordt gebruik gemaakt van de typroutines van het in- en uitvoerprogramma, hetwelk impliceert dat breuk en macht worden uitgetypt in de gedaante, door de betrokken constanten gespecificeerd.

Opm.2. De controle-aanroep roept op zijn beurt de Typ-controle uit het in- en uitvoerprogramma aan. Wat betreft deze routine en de stand van de tabulatorstoppen dient men te weten, dat breuk en macht samen beschouwd worden als een getal.

Opm.3. Het hoofdprogramma dient met de gebruikelijke aanroep

```

24      7    XO
      7    19   X2  =)

```

de Typcontrole op het begin van een nieuwe regel in te stellen.

Opm.4. Is het typen goed gegaan, dan komt de besturing na de controle normaal 2 opdrachten verder in het hoofdprogramma; na een fout komt hij na TWNR en het goede aantal TAB-signalen, in plaats van 2 verder, 2 terug, zodat breuk en macht samen overgetikt worden.

Service-programma's.

Incidenteel uittypen.

Als men via het bedieningspaneel h en h+1 uit wil typen, kiese men:

0 a F2 als startadres als wel de typcontrole op een nieuwe regel ingesteld worde

1 a F2 als startadres als niet de typcontrole op een nieuwe regel ingesteld worde.

BEGIN GEKOZEN OPDRACHT (machine stopt snel).

In de opdracht schakelaar 10/h+1.

SCHRIJF; DOE; BEGIN VOLGENDE OPDRACHT

(machine stopt na typen). Dan ten tweede male

SCHRIJF; DOE; BEGIN VOLGENDE OPDRACHT

(machine stopt na controle).

Daarna kan een volgend getal worden uitgetypt, door het nieuwe macht-adres in de opdrachtschakelaar te zetten, dan weer 2 x SCHRIJF-DOE, BEGIN VOLGENDE OPDRACHT. (Het "BEGIN GEKOZEN" hoeft dus maar een keer).

Inbrengen van het bedieningspaneel.

Men brengt met het handregister de ongehalveerde breuk in S. In de getalschakelaars zette men de met h geassembleerde macht m (dus $(m+K) \cdot 2^{15+h}$).

Start de machine op 11 a F2; $\{S\} \cdot 10^m$ wordt volgens de conventies van Rd1 op h en h+1 weggeborgen.

Assemblage - controle.

Als men een rij drijvende-komma getallen, waarvan de machten staan op $s+1, s+1+2, \dots, s+1+2k$ wil onderwerpen aan de controle, of in de rechterhelft het juiste assemblage-resultaat staat, zet men $s+1$ in de a-positie, $s+1+2k$ in de b-positie van de getalschakelaars. Start de machine op 26a F2; als alles goed is, stopt de machine op 25/0 in 22a E2; in geval van ongerechtigdheden eerder op 24/2 of 24/3.

Het programma Rd1.

De administratieve routines spreken coor zichzelf.

De additieve routines bestaan uit twee gedeelten: de additie (in kanaal EO) en de normering (kanaal F1, adres 0 t/m 20).

In kanaal EO wordt de macht "gedesassembleerd", het breukgedeelte wordt m.b.v. een variabele opdracht (4EO) uit het geheugen aangehaald (al of niet met tekenwisseling!). Vervolgens worden de twee machten vergeleken. (A_m = de macht van de quasi-accumulator "A", V_m de macht van het getal, dat bij de inhoud van "A" opgeteld dient te worden). Nu wordt in de volgende gevallen gesplitst:

1. $V_m < A_m - 8$.

D.w.z. "V" zinkt bij "A" in het niet, de vermeerdering van "A" is in de precisie, waarin "A" wordt gerepresenteerd, gelijk aan nul. "A" wordt onveranderd gelaten. (9aEO wordt dan gehoorzaamd: $V_m - A_m + 9 \leq 0$, dus $V_m < A_m + 8$).

2. $A_m - 8 \leq V_m < A_m$.

Voor "V" dient een andere representatie gevormd te worden: V_m dient aan A_m gelijk gemaakt te worden, V_b (het breukgedeelte van "V") wordt dan "ter compensatie" door $10^{A_m - V_m}$ gedeeld: dan zijn de machten der twee bij elkander op te tellen getallen aan elkaar gelijk en er kan opgeteld worden.

(8a EO is niet gehoorzaamd, dus $V_m - A_m + 1 \leq 0 \rightarrow V_m - A_m < 0 \rightarrow V_m < A_m$; 9a EO is niet gehoorzaamd, dus $V_m - A_m + 9 > 0$ (neg. weggeschreven!) $\rightarrow V_m - A_m + 8 \geq 0 \rightarrow V_m \geq A_m - 8$; zodat aan de bedoelde ongelijkheid voldaan is, als in 9b EO de deelopdracht gevormd wordt.

3. $V_m = A_m$.

De machten der twee getallen zijn aan elkaar gelijk: er kan dus zonder meer opgeteld worden (6a EO is genegeerd dus $V_m - A_m \leq 0$; 8a EO is wel gehoorzaamd, dus $V_m - A_m + 1 > 0$; hieruit volgt de gelijkheid).

4. $A_m < V_m \leq A_m + 8$.

Nu heeft "V" de grootste macht, de macht A_m dient dus aan V_m gelijk gemaakt te worden, A_b dient door $10^{V_m - A_m}$ gedeeld te worden. Dit geschiedt in het programma door V en A van plaats te laten verwisselen, zodat na enige tijd het programma, dat geval 2 verzorgt, gebruikt kan worden. (6aEO is gehoorzaamd, dus $V_m - A > 0$; 13a EO is eveneens gehoorzaamd, dus $V_m - A_m - 8 \leq 0$ (neg. weggeschreven!) zodat aan bedoelde ongelijkheid voldaan is, als de besturing in 17bEO arriveert).

5. $V_m > A_m + 8$.

D.w.z. "V" overtreft "A" in orde van grootte volkomen, en de inhoud van de quasi-accumulator moet aan "V" gelijk worden gemaakt. (13a EO is niet gehoorzaamd, dus $V_m - A_m - 8 > 0$ (neg. weggeschreven). Adres 25E1 wordt voor het staartje gebruikt.

De optellingen konden steeds, zonder gevaar van overloop geschieden, omdat twee getallen bij elkander worden opgeteld, die elk in absolute waarde kleiner zijn dan $\frac{1}{2}$.

De normering van het antwoord, door de primaire additie afgeleverd, geschied na de gevallen 2, 3 en 4 (1 en 5 laten in de quasi accumulator een genormeerd getal achter!)

De normering heeft de volgende taak:

1. als na de primaire additie $|A_b| > \frac{1}{2}$, dan moet A_b door 10 worden gedeeld, ter compensatie moet de macht met 1 worden vermeerderd.
2. als na de primaire additie - door het wegvallen van cijfers ! - $-|A_b| < 0.05$, dient A_b zovaak met 10 vermenigvuldigd te worden, totdat $|A_b| \geq 0.05$; elke keer, dat A_b met 10 vermenigvuldigd wordt, dient de macht met 1 verminderd te worden.
3. als het antwoord absoluut te klein is, om door Rd1 nog gehanteerd te kunnen worden, wordt het vervangen door het kleinste, door Rd1 nog hanteerbare getal. Dit doet zich voor
 - a. als na de primaire additie $A_b = 0$.
 - b. als tijdens het vermenigvuldigen met machten van 10 de macht, die steeds met 1 verminderd wordt, kleiner wordt dan $-K$ (dus als $A_m = \text{macht} + K < 0$ wordt).

Als 2aF1 genegeerd, 3aF1 gehoorzaamd wordt, bevinden we ons in geval 3(a). Geval 1 wordt gedetecteerd door te onderzoeken of A_b en $2A_b$ (door additie gevormd!) verschillend teken hebben. Opdrachten 6aF1 t/m 9bF1 verzorgen de vermenigvuldiging met 0,1.

Geval 2 wordt verder onderzocht: inmiddels staat $2A_b$ in S; men probeert dus met 10 te vermenigvuldigen en onderzoekt of er overloop in A is. Als er geen overloop is (15 F1 e.v.) wordt A_m met 1 verminderd; onderzocht wordt, of de nieuwe A_m soms < 0 is (nl. $-A_m > 0$). Hier wordt dus onderzocht, of geval 3(b) van toepassing is.

Bij de subroutine voor de vermenigvuldiging en deling is het berekende gedeelte aanmerkelijk eenvoudiger: slechts de deling heeft hier als bijzonderheid, dat het deeltal eerst met 0,1 wordt vermenigvuldigd.

Zij hebben hetzelfde normeringsprogramma (23F1 t/m 31F1). Dit programma is eenvoudiger dan dat voor de normering na additie, omdat de A_b slechts te klein kan zijn; is dit het geval, dan is een maal met 10 vermenigvuldigen voldoende. Echter kan de macht van begin af aan al kleiner zijn dan $-K$; hier wordt onmiddellijk op getest en wordt hier het getal in "A" te klein bevonden, dan wordt ook hier "A" vervangen door $0,1 \times 10^{-K}$ (29 b F1).

Opm.1. Een kleine bespiegeling leert, dat de gangbare techniek der som-controle bij drijvende-komma-rekentechniek zijn analogon vindt in een product-controle. Het is echter gevaarlijk om te controleren of de uiteindelijke drijvende-komma-controle-grootheid genoegzaam 1 is: het breukgedeelte kan juist 0,1000.. of 0,999.. zijn. Het meest voor de hand liggend is, dit getal met behulp van Rd1 met $\frac{1}{2}$ te vermenigvuldigen. Dan kan men het breukgedeelte van de quasi accumulator vergelijken met de $\frac{1}{2}$ in Rd1 representatie) en na een shiftje controleren of de macht gelijk is aan K.

$\frac{1}{2}$ in Rd1 representatie staat daartoe op 27E2 (breuk) en 28E2 (macht).

Opm.2. Als veel getallen met de hand ingebracht moeten worden, is voor K waarschijnlijk $K = 2^{13} + 16$ of $K = 2^{13} + 32$ een handige keuze: dan hoeft men voor negatieve exponenten minder schakelaars van het handregister om te zetten.

Opm.3. De productcontrole faalt, als het getal nul in Rd1 representatie (dus $0,1 \cdot 10^{-K}$) als factor voorkomt. Doorgaan zal bv. een macht 10^{-100} klein genoeg zijn: de controle gaat dan ongehinderd!

Opm.4. In verband met de plaats in het geheugen staan achter in dit rapport kanalen behorend bij Rd1 en Rd2 afwisselend. (De scheiding in Rd1 en Rd2 doet wellicht een diepere achtergrond vermoeden: zij is slechts historisch gegroeid).

Subroutine voor "Drijvende Komma"-rekentechniek. 64 + 64 plaatsen.

		A		EO	
		A		EO	
OPTELLING =)	0EO	23	15	XO	splitst macht en adres van breuk
		7	2	EO =+	
AFTREKKING =)	1	23	15	XO	splitst macht en adres van breuk
		0	23	EO	vermeerdert functiegedeelte van adres van breuk met 1.
Ob EO -+	2	24		XO	
		0	16	EO	vormt aanhaal opdracht voor breuk
	3	4	4	EO	
		12	9	E1	schrijft macht V_m van aangehaald getal.
21a,11a EO -+	4	(-		X)	23 30 XO 20 24-31 EO 10(11) "h" 15 21 EO =+
	5	1	5	E1	vormt $V_m - A_m$
		4	27	E1	
	6	14	11	EO -+	als $V_m > A_m$
		24	1	X16	
	7	4	27	E1	
		24	8	X16	
	8	14	21	EO -+	als $V_m = A_m$ (direct optellen!)
		5	4	EO	$-(V_m - A_m + 9)$ naar het geheugen
	9	6	29	E1 -+	klaar! (V zinkt bij A in 't niet!)
		0	15	EO	er moet gedeeld worden door 10^n
	10	4	4	EO	plaats de deel opdracht
		23		XO	$A = \pm 0$, inleiding voor deling
	11	7	4	EO =+	deel door 10^n
6a EO =+		24	247	X24	vormt $V_m - A_m - 8$
	12	5	27	E1	schrijft $-(V_m - A_m - 8)$
		24	254	X24	
	13	14	17	EO -+	$A_m < V_m < A_m + 8$, dus A_b door 10^n delen.
		2	9	E1	"A" zinkt bij V in 't niet!
	14	4	5	E1	$V_m \geq A_m$
		7	25	E1 =+	nu nog $V_b \geq A_b$, dan klaar.
	15	20	23	EO	constante voor vorming van
		15	21	EO	opdracht: deel door 10^n

Subroutine voor "Drijvende Komma"-rekentechniek (vervolg).

	(A	16	EO	Kan vervallen, als de twee	
	A	16	EO)	helften van EO op 1 band	
16EO	23	30	XO	geponst worden.	
	10		XO	constante voor vorming	
17	24	256	X24	van aanhaalopdracht voor breuk	
13aEO =+	1	15	EO	= "25/0" abs.stop.	
	18	5	4	E0	vormt opdracht: deel (A_b)
		2	6	E1	door 10^n .
	19	12	6	E1	plaatst deze deelopdracht
		10	9	E1	$V_b \rightleftharpoons A_b$
	20	12	5	E1	$V_m \neq A_m$
		22	29	XO	A meteen de goede nul!
	21	7	4	E0 =+	
4b EO =+		8	6	E1	de vitale optelopdracht.
	22	12	6	E1	
		7		F1 =+	naar "normering na additie"
	23	0		XO	constante voor
		1		XO	de aftrekking
	24	+ 1 0000 0000	X		
					machten van
	25	+ 1000 0000	X		10, die
					gebruikt
	26	+ 100 0000	X		worden,
					zodra $V_m \neq A_m$,
	27	+ 10 0000	X		terwijl
					$ V_m - A_m < 9$.
	28	+ 1 0000	X		
	29	+ 1000	X		
	30	+ 100	X		
	31	+ 10	X		

Subroutine voor "Drijvende Komma"-rekentechniek (vervolg).

	(A	E1	Deze controlecombinatie kan ver-
	A	E1)	vallen, als E1 na E0 op dezelfde
OE1	(+	X)	band geponst wordt.
			werkruimte. bij deling 18 13 E1(xQ1)
			bij CONTROLE wordt dit 20(21) h
1	23	1	XO halveer (om factor $\frac{1}{2}$ er bij te
			maken)
	15	15	E1 =+
31b F1 =+	2	24	2 XO + cond.stop; somcheck faalt?
	24	1	X16
3	4		E1
	7	22	E1 =+
4	18	13	E1 constante voor
	20		XO "vorm deel-opdracht"
5	(+	X)	werkruimte: macht A_m van de
			quasi-accumulator "A"
6	(+	X)	werkruimte: breuk A_b van de
			quasi-accumulator "A"
7	+	1	D
8	24		XO constante voor
	2		XO "SCHOON IN"
9	(+	X)	werkruimte: V_m van aangehaalde
			getal
9b FO =+	10	(+	X) wordt $4/h$
			$12/h+1$ bij: VUIL UIT.
	11	7	29 E1 =+ klaar!
		24	XO
	12	24	XO constante voor
		18	XO "vorm vermenigvuldig opdracht"
	13	+ 5368 7092	X = 0,1, naar boven afgerond
	14	(+	X) wordt $24/0$
17b FO =+			$18(19)/h$
	15	22	29 XO voorlopige A_b in S
1b E1 =+		3	5 E1 voor onderzoek A_m

Subroutine voor "Drijvende Komma"-rekentechniek (vervolg).

	(A	16	E1	Deze controle-combinatie kan ver-
	A	16	E1)	vallen, als de 2de helft van E1 di-
16E1	12	6	E1	rect na de 1ste geponst wordt.
	8	6	E1	schrijft voorlopige A_b
				verdubbelt deze vast
17	24		X0	
	24	255	X24	om te testen of $A_m > 0$.
18	4	24	E1	
	7	23	F1 =+	voortzetting van de additie
19	0		X0	constante voor - deling
	1		X0	en - vermenigvuldiging.
20	4		X0	constante voor
	12	1	X0	"VUIL UIT"
21	20	4	X24	constante, ingelast ten
	11	189	E24	dienste van de som-controle
3b E1 =+	22	24	3	X0 - cond.stop; somcheck faalt?
	7	29	E1 =+	klaar, alles in orde!
23	(+		X)	werkruimte normering na verm. en
				deling
24	(+		X)	werkruimte normering na additie
				normering na verm.deling
14b EO =+	25	12	6	E1 $V_b \geq A_b$
	7	29	E1 =+	
26	+		D =	K
3b FO =+	27	(+	X)	werkruimte bij 24 X0 bij "SCHOON IN"
				additie; bij "VUIL UIT" 2/h
28	12	5	E1	plaatst macht bij "SCHOON IN" in
				"A"
	4	6	E1	plaatst breuk bij "SCHOON IN" in
				"A"
29	2	31	E1	ophogen met 1
	24	1	X16	van het adres
30	4	31	E1	van de koppelopdracht
	24		X0	
31			=)	gemeenschappelijke
				koppelopdracht

Subroutine Rd2: In- en uitvoer; service ten dienste van Rd1.

32 + 32 plaatsen.

		A		E2	
		A		E2	
	OE2	23	15	X0	<u>Typroutine.</u> isoleert h en de macht
		9	26	E1	macht minus K
1		0	23	E2	vormt variabele opdracht
		4	2	E2	
2		(-		X)	wordt 12 12 E2 schrijf macht weg 2/h haal breuk
3		22	28	X0	verdubbel breuk
		15	4	F2 +=	voortzetting in kan F2
	4	23	15	X0	<u>controle op typen</u>
		0	26	E2	vormt variabele opdracht
5		9	6	E2	macht, minus "macht na typen"
		4	6	E2	plaatst var. opdracht
6		(-		X)	9 12 E2 minus "breuk na typen" 8 h plus "breuk uit geheugen"
7		9	26	E1	verminder dit met K. S= - 0!
		24	6	X0	
8		7		X2 =)	naar de "echte" typcontrole
		14	30	F2 -+	als een fout is gemaakt
9		7	29	E1 +=	typen is goed gegaan.
16a F2 +=		24	7	X0	<u>Einde: Inbrengen met de hand.</u>
10		(-		X)	wordt 12/h 4/h+1
11		24	2	X0	
		24	3	X0	
25bF2+= 12		(-		X)	wordt 13/h+1 Einde bandlezen. 5/h
13		7	29	E1 +=	klaar, terug naar hoofdprogramma.
		0		X0	<u>Vervolg Assemblage-controle.</u>
30bF2+= 14		(-		X)	3/h+1 22 14 X0
15		23	15	X0	nu hopen we -h in S geïsoleerd te heb- ben
		8	29	E2	+h+1 (de lopende).

Subroutine Rd2; In- en uitvoer; service ten dienste van Rd1 (vervolg)

	(A	16	E2	kan vervallen, als deze pag. onmid-
	A	16	E2)	dellijk achter de eerste helft geponst
				wordt.
16E2	12	30	E2	we hopen op wegschrijven van +1
	2	30	E2	
17	24	3	X0	Stopt als fout in assemblage
	24	254	X24	
18	4	30	E2	
	24	2	X0	Stopt als fout in assemblage
19	2	29	E2	ophogen van de
	24	2	X16	lopende h+1
20	4	29	E2	met <u>2</u> !
	1	10	E2	test of de laatste al geweest is!
21	5	30	E2	
	14	28	F2	-+ nog niet klaar
22	24	256	X24	= 25/0, als alles goed is gegaan
	0		X0	stopt de machine hier
23	12	12	E2	
	2		X0	
24	12		X0	
	4	1	X0	
25	13	1	X0	
	5		X0	
26	9	12	E2	
	8		X0	
27	+. 2500	0000	X	$\frac{1}{2}$ in Rd1-representatie
28	0	27	E2	
	0		D0	
29	(+		X)	werkruimte
30	(+		X)	werkruimte
31	3		X0	
	22	14	X0	

Subroutine voor "Drijvende-Komma"-rekentechniek (vervolg).

		A		FO	
		A		FO	
CONTROLE =)	OF0	2	15	EO	optelling
		0	30	EO	voor de
	1	0	16	EO	somcontrole
		7	10	FO =+	
"SCHOON IN"=)	2	23	15	XO	splitst macht en adres van de breuk
		0	8	E1	vormt opdracht "haal breuk"
	3	4	27	E1	plaatst opdracht
		15	27	E1 =+	ga naar die opdracht
"VUIL UIT"=)	4	23	15	XO	(splitst macht en) vormt adres van de breuk
		22	15	XO	
	5	4	27	E1	bergt "h"
		22	14	XO	
	6	8	27	E1	vormt opdracht voor
		8	20	E1	VUIL UIT
	7	12	10	E1	
		2	5	E1	haal macht uit "A"
	8	22	14	XO	schuif deze in goede positie
		2	6	E1	haal breuk uit "A"
	9	8	27	E1	voltooi de assemblage van h en de macht;
		7	10	E1 =+	ga naar de wegschrijfoopdrachten
1b FO =+	10	0	28	EO	
		0	24	EO	voortzetting van de optelling
	11	0	29	EO	voor de som-
		0	25	EO	controle
	12	0	27	EO	
		0	26	EO	
	13	15	25	FO =+	
19b FO =+		0	19	E1	(bij -VERM: vermeerder functie- gedeelte met 1)
	14	9	26	E1	verminder de macht met K
		0	12	E1	vorm vermenigvuldig opdracht
	15	4	14	E1	
		8	5	E1	het optellen der machten

Subroutine voor "Drijvende-Komma" rekentechniek (vervolg).

	(A	16	FO	Deze controlecombinatie kan ver-
	A	16	FO)	vallen, als men voor FO één
	16FO	12	5	E1 band maakt.
		10	6	E1 macht voorlopig klaar
	17	8	6	E1 de verdubbeling om
		15	14	E1 een factor $\frac{1}{2}$ kwijt te raken!
				E1 ==+ ga vermenigvuldigen
+ VERMENIG- =)	18	23	15	XO splitst macht en adres van de
VULDIGING		7	14	FO ==+ breuk
- VERMENIG- =)	19	23	15	XO splitst macht en adres van de
VULDIGING		15	13	FO ==+ breuk
+ DELING =)	20	23	15	XO splitst macht en adres van de
		7	22	FO ==+ breuk
- DELING =)	21	23	15	XO splitst macht en adres van de
		0	19	E1 breuk
				E1 hoog functiegedeelte bij h met
				1 op!
20b FO -+	22	9	7	E1 verminder macht met K+1 (ver-
		0	4	E1 schil +K,+1, vanwege 0,1)
				E1 vorm deel-opdracht
	23	4		E1 plaats deelopdracht
		9	5	E1 minus voorlopige macht in S
	24	13	5	E1 macht voorlopig klaar
		10	6	E1
	25	7		E1 ==+
13a FO ==+		0	23	EO
	26	0	31	EO
		0	8	E1 verdere
	27	0	12	E1 voortzetting
		0	13	E1 van de
	28	1	21	E1 optelling
		1	4	E1 ter som-
	29	0	20	E1 controle
		0	19	E1
	30	1	26	E1
		0	7	E1
	31	4		E1 schrijve -0 weg
		7	2	E1 ==+

Subroutine voor "Drijvende-Komma"-rekentechniek (vervolg).

	(A	F1	Deze controlecombinatie kan
	A	F1)	vervallen, als F1 onmiddellijk
			achter F0 geponst wordt.
22b EO =+	0F1	2 6	E1
		24 255	X24
	1	4 24	E1
		24 1	X16
	2	14 3	F1 --+ $A_b > 0$
		4 24	E1
	3	14 16	F1 --+ $A_b = 0!$
2a F1 --+		24	X0
	4	13 24	E1
		8 6	E1 in $S = A_b + A_b$; zelfde teken?
	5	14 21	F1 --+ $A_b < 0$
		12 27	E1
	6	14 10	F1 --+ zelfde teken; probeer met 10
			te vermenigvuldigen!
22a F1 --+		2 5	E1 vermenigvuldig met 0,1; A_m met
			1 vermeerderen!
	7	24 1	X16 $A_m + 1 \geq A_m$
		4 5	E1
	8	10 13	E1
		18 6	E1 $0,1 A_b \geq A_b$
	9	4 6	E1
		7 29	E1 --+ klaar!
	10	24 256	X24 = "25/0"
22b, 16a, 6aF1 =+		12 24	E1 $ A_b $ is te klein!; schrijf A_b
			weg, voor als x 10 teveel geeft
	11	24 17	X0 probeer maal 10 te maken
		24 255	X24 eventuele +0 $\rightarrow$ -0.
	12	4 27	E1
		24 1	X16
	13	14 18	F1 --+ overloop: laatste keer. maal
			10 was te veel!
		5 27	E1
	14	14 18	F1 --+ overloop: laatste keer. maal
			10 was te veel!
		24	X0 nu 13 [A] = +1.
	15	1 5	E1
		5 5	E1 $A_m - 1 \geq A_m$

Subroutine voor "Drijvende-Komma"-rekentechniek (vervolg).

	(A	16	F1	Deze controlecombinatie kan ver-
	A	16	F1)	vallen, als de twee helften van
16F1	14	10	F1 -+	F1 achter elkaar geponst worden.
				$A_m \text{ nog } \geq 0$, dus doorgaan.
	10	13	E1	0,1 in S
17	23		X0	
	4	5	E1	macht wordt minimaal ingesteld
18	7	19	F1 =+	
13a, 14a F1 =+	10	24	E1	haal laatste voor x 10, die
				teveel was.
18a F1 -+	19	23	1	X0
				halveer in verband met deel-
	24		X0	centventie.
20	12	6	E1	nieuwe A_b wordt in quasi-accu-
				mulator ingevuld.
	7	29	E1 =+	klaar!
21	24	256	X24	= "25/0" abs.stop
5a F1 =+	12	27	E1	
22	14	6	F1 -+	verschillend teken, dus $ A_b \geq \frac{1}{2}$!
	15	10	F1 =+	
18b E1 =+	23	6	29	F1 -+ $A_m < 0$
	24	17	X0	
24	24	255	X24	
	4	23	E1	
25	6	29	E1 -+	er is overloop en $A_m \geq 0$; klaar!
	24	1	X16	
26	5	23	E1	
	6	29	E1 -+	er is overloop en $A_m \leq 0$; klaar!
27	3	5	E1	
	24	1	X16	
28	5	5	E1	
	14	30	F1 -+	na vermindering $A_m - 1 \geq A_m$, is $A_m \geq 0$.
23a F1 -+	29	23	X0	
	10	13	E1	
30	4	5	E1	
28b F1	23	1	X0	
31	12	6	E1	
	7	29	E1 =+	klaar!

Subroutine Rd2: In- en uitvoer; service ten dienste van Rd1.(ver-
volg)

		A		F2	
		A		F2	
☆		OF2	24	6	X0
			7	19	X2 =)
					<u>Incidenteel typen onder</u>
					<u>TWNR controle van het bedienings-</u>
					<u>paneel.</u>
=)	1	24	6	X0	
		4	31	E1	
	2	24	2	X0	10/h+1 via bedieningspaneel.
		7		E2 =)	ga typen (h,h+1)
	3	24	2	E1	10/h+1 via bedieningspaneel
		7	4	E2 =)	controleer
	4	7	1	F2 =+	klaar voor het volgende getal.
3b E2 =+		24	6	X0	<u>Voortzetting typroutine:</u>
	5	7	16	X3 =)	typ de breuk (met teken)
=)		23	31	X0	halveer deze weer
	6	10	12	E2	haal de macht
		4	12	E2	schrijf de - gehalveerde breuk -
					weer weg voor controle
	7	23		X0	A schoon
		24	255	X0	A=-0
	8	24	8	X0	spatie
		24	8	X0	spatie
	9	24	6	X0	
		15	2	X3 =)	type de macht (met teken)
	10	12	6	E2	schrijf de macht weg (voor con-
					trole)
		7	29	E1 =+	naar gem.koppelopdracht
☆	11	23	1	X0	<u>Inbrengen met de hand</u>
		24	7	X0	lees macht-assemblage
	12	12	29	E2	berg de gehalveerde breuk even
		22	14	X0	
	13	12	12	E2	bergt (h).2 ¹⁵
		23	15	X0	
	14	8	24	E2	vormt var.opdracht
		8	12	E2	
	15	12	10	E2	
		10	29	E2	gehalveerde breuk weer in S

Subroutine Rd2: In- en uitvoer; service ten dienste van Rd1. (ver-
volg)

	(A	16	F2	Kan vervallen, als tweede
	A	16	F2)	helft onmiddellijk achter de eerste
	16F2	15	9	E2 ==+ wordt geponst.
		0		naar var. opdracht
			X0	
17	24	6	X0	<u>Bandlezen van floating-getallen.</u>
	15	31	X0 =)	naar invoer.
18	4	6	E2	bergt -h
	22	14	X0	
19	8	6	E2	
	9	25	E2	vormt de var.dubbele wegbergop-
				dracht (neg.)
20	13	12	E2	plaatst deze
	24	6	X0	
21	15	31	X0 =)	naar invoerprogramma. lees breuk
	22	1	X0	halveer deze
22	4	2	E2	berg halve breuk even
	24	6	X0	
23	15	31	X0 =)	naar invoerprogramma; lees macht.
	1	26	E1	vermeerder deze met K(-macht!)
24	22	14	X0	shift deze in positie
	8	6	E2	assembleer met h.
25	2	2	E2	nu breuk in A
	7	12	E2 ==+	naar variabele wegbergopdracht
26	24	7	X0	<u>Assemblage-controle</u>
	22	15	X0	
27	4	10	E2	plaatst "laatste h+1"
	23	14	X0	
28	12	29	E2	plaats "lopende h+1"
-+	2	29	E2	vorm variabele opdracht
29	0	31	E2	
	4	14	E2	
30	7	14	E2 ==+	
8b E2 ==+	2	31	E1	spring terug in hoofd-
31	24	254	X24	programma na een
	7	30	E1 ==+	type-out!