

**COMPUTERSIMULATIEMODEL
TER BEPALING VAN DE
TANKCAPACITEIT VAN EEN
OLIETERMINAL IN LIBANON**



DEEL 2

Afstudeerverslag

Clieterminal te Tripoli in Libanon.

Capaciteitbepaling van de opslagtanks
met behulp van computersimulatie.

deel 2 : - Analyse computerprogramma
 - Gebruikershandleiding

J.C. Schalkwijk
st.nr. 731305
afd. Civiele Techniek
TU Delft

INHOUSOPGAVE

blz.

1. ANALYSE COMPUTERPROGRAMMA.	3
1.1. Component MAIN.	5
1.1.1. Beschrijving.	5
1.1.2. Stroomschema.	11
1.2. De GENERATOREN 1 t/m 7	15
1.2.1. Procesbeschrijvingen.	15
1.2.2. Stroomschema's.	18
1.3. De HMASTER.	21
1.3.1. Procesbeschrijving.	21
1.3.2. Stroomschema.	23
1.4. Produkt SHIP.	26
1.4.1. Procesbeschrijving.	26
1.4.2. Stroomschema.	28
1.5. Crude SHIP.	30
1.5.1. Procesbeschrijving.	30
1.5.2. Stroomschema.	32
1.6. Produkt TANK.	35
1.6.1. Procesbeschrijving.	35
1.6.2. Stroomschema.	36
1.7. Crude C_TANK.	37
1.7.1. Procesbeschrijving.	37
1.7.2. Stroomschema.	38
1.8. De VERZIEKER (Raffinaderij).	40
1.8.1. Procesbeschrijving.	40
1.8.2. Stroomschema.	41
1.9. De NOTEERDER.	43
1.9.1. Procesbeschrijving.	43
1.9.2. Stroomschema.	44

2. GEBRUIKERSHANDLEIDING.	45
2.1. De dataset.	45
2.2. Aantal aanlegboeien.	47
2.3. Tussenaankomsttijden.	48
2.4. Pompkarakteristiek.	49
2.5. Andere schepen.	50
2.6. Slecht weer perioden.	51
2.7. Raffinaderij.	51
2.8. Uitvoer.	52

BIJLAGEN.

1. ANALYSE COMPUTERPROGRAMMA.

Elk PROSIM programma begint met een definitie-gedeelte. Hierin worden de verschillende te gebruiken componenten, attributen, histogrammen, variabelen etc. gedeclareerd.

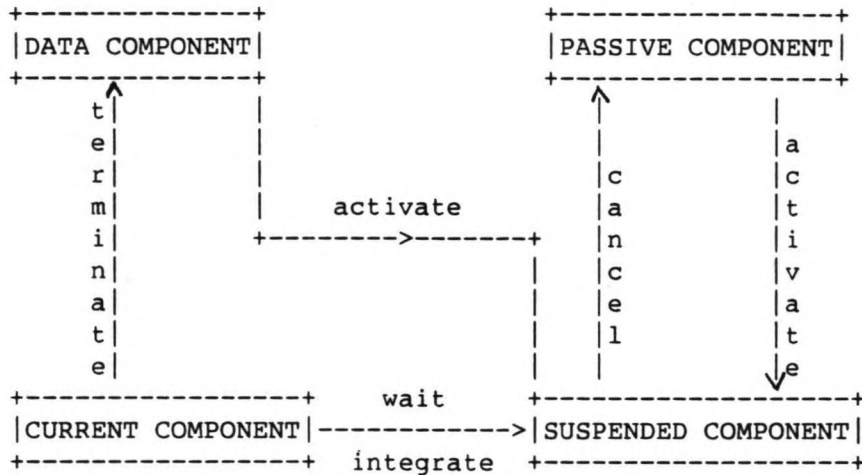
```

1      0 BEGIN;
10     1 0  DEFINE GENERATOR1 AS A COMPONENT,
          GENERATOR2 AS A COMPONENT,
          GENERATOR3 AS A COMPONENT,
          GENERATOR4 AS A COMPONENT,
          GENERATOR5 AS A COMPONENT,
          GENERATOR6 AS A COMPONENT,
          GENERATOR7 AS A COMPONENT,
          NOTEORDER AS A COMPONENT WITH ATTRIBUTES (TANK_NOT REFERENCE
            TO TANK, CTANK_NOT REFERENCE TO C_TANK),
          HMASTER AS A COMPONENT WITH ATTRIBUTE (SHIP_IN_HAND REFERENCE
            TO SHIP),
          VERZIEKER AS A COMPONENT WITH ATTRIBUTES (VERZ_TANK REFERENCE
            TO TANK, VERZ_CTANK REFERENCE TO C_TANK);
11     1 0  DEFINE SHIP AS A CLASS OF COMPONENTS EACH WITH ATTRIBUTES
          ((CONTENCE, LOSSPEED, MOORINGTIME, LEAVINGTIME) REAL,
          LEGEN CONTINUOUS (LEGEN'=-LOSSPEED) INIT(1E4),
          (BERTH, PROD, GENNR) INTEGER, TANK_IN_HAND REFERENCE TO TANK,
          CRUDETANK REFERENCE TO C_TANK, LOG2 LOGICAL),
          TANK AS A CLASS OF COMPONENTS EACH WITH ATTRIBUTES
          (LADING CONTINUOUS (LADING'=INPUT-OUTPUT) INIT(2E4),
          (INPUT, OUTPUT, B_INPUT, CAPACITY, HULP) REAL, NR INTEGER,
          LOGIC LOGICAL, (Y1, Y2) DOUBLE REAL),
          C_TANK AS A CLASS OF COMPONENTS EACH WITH ATTRIBUTES
          (C_LADING CONTINUOUS
          (C_LADING'=(A1-C_OUTPUT)+C*(C1*EXP(-C2*C_LADING)))
          INIT(1E5),
          (C_OUTPUT, C_VULLEN, A1, A2, C1, C2, C3, C_CAPACITY, AMAX, VAR)
          REAL, C INTEGER, Y3 DOUBLE REAL),
          NEWC_TANK AS A REFERENCE TO C_TANK VARIABLE,
          NEWTANK AS A REFERENCE TO TANK VARIABLE,
          NEXTSHIP AS A REFERENCE TO SHIP VARIABLE;
12     1 0  DEFINE (BOEI_ROW(4), AANLEG_ROW, OPSLAG_ROW) AS QUEUES,
          RAND(2) AS RANDOMSTREAMS WITH SEEDS (11123, 99755)
          SAMPLED FROM DISTRIBUTION UNKNOWN,
          UNIF1(8) AS RANDOMSTREAMS WITH SEEDS (377003613,
          648473574, 1396717869, 2027350275, 1356162430, 711072531),
          UNIF2(6) AS RANDOMSTREAMS WITH SEEDS (769795447,
          1752629996, 645806097, 2013311468, 1393552473, 1966641861),
          BERGING(4) AS PLOTS,
          INVOER AS A PLOT,
          INHOUD(4) AS HISTOGRAMS WITH PARAMETERS
          L_BOUND(0) N_CLASSES(35) CLASSWIDTH(1E3),
          WACHTTIJD(7) AS HISTOGRAMS WITH PARAMETERS
          L_BOUND(0) N_CLASSES(40) CLASSWIDTH(2),
          W_TIMES(7) AS HISTOGRAMS WITH PARAMETERS
          L_BOUND(0) N_CLASSES(30) CLASSWIDTH(0.5),
          (AD, P, X1, X2, X3, X4, X5, X6, X7, X8, X9, X10) AS REAL VARIABLES,
          (VERSIE, NUMBOEI, NUMRUNS, B, I, J, K, L, M, N, O, NUMSHIP, HELP)
          AS INTEGER VARIABLES,
          LOG1 AS LOGICAL VARIABLES,
          AN HOUR AS TIME_UNIT;

```

Aan de hand van de procesbeschrijvingen van de componenten wordt de betekenis van de gebruikte waarden duidelijk gemaakt. Na een uitgebreide uitleg van de processen, volgt een stroomschema van het desbetreffende proces. De nummers die tussen haakjes staan verwijzen naar de regelnummers in het programma.

Alvorens met de procesbeschrijvingen te beginnen, wordt eerst in figuur 1 aangegeven hoe en met welke comando's, de componenten geactiveerd en gepassiveerd moeten worden. Dit is afhankelijk van de staat waarin ze zich bevinden.



figuur 1

Een data component is een 'dode' component, die alleen informatie van de component bij zich draagt (b.v. attributenlijst). De andere componenten zijn 'levende' componenten.

Tijdens het simuleren is er altijd maar 1 component die de statements van zijn proces uitvoert. Dit is de current component. Bij het begin van het programma is dit MAIN.

De current component maakt van een data component of van een passive component, door middel van ACTIVATE, een suspended component. Een suspended component wordt de current component zodra de huidige current component een suspended of data component wordt door middel van wait, integrate of terminate. Indien er meerdere suspended components op het zelfde moment current component moeten worden, dan gebeurt dit in dezelfde volgorde als waarin ze suspended component zijn geworden.

Verder zijn de references nog een belangrijk hulpmiddel. In een procesbeschrijving van een component mogen alleen eigen attributen en referenties naar andere componenten gebruikt worden. In het programma zullen we zien dat hiervan veelvuldig gebruik gemaakt wordt.

1.1. Component MAIN.

1.1.1. Procesbeschrijving (130-253).

Als eerste procesbeschrijving in elk PROSIM programma zien we de procesbeschrijving van de component MAIN (130-253).

```
130 1 0 PUT LIST ('VERSIE','SIMULATIETIJD(JAREN)','AANTAL STEIGERS');
131 1 0 GET LIST (VERSIE,NUMRUNS,NUMBOEI);
132 1 0 PUT SKIP LIST (VERSIE,NUMRUNS,NUMBOEI);
133 1 0 PUT SKIP (2);
134 1 0 PUT SKIP EDIT ('PARAMETER WAARDEN VAN DE STOCHASTISCHE VERDELINGEN')
    (A);
135 1 0 GET LIST (X1,X2,X3,X4);
136 1 0 PUT SKIP LIST (X1,X2,X3,X4);
137 1 0 GET LIST (X5,X6,X7,X8,X9,X10);
138 1 0 PUT SKIP LIST (X5,X6,X7,X8,X9,X10);
139 1 0 PUT SKIP (2);
```

(130-139):

MAIN leest hier een aantal waarden van de dataset en zorgt er voor dat ze worden afgedrukt. Hiervoor gebruiken we een aantal PL1 statements, zoals: PUT SKIP, PUT SKIP EDIT, PUT LIST, PUT SKIP LIST voor het afdrukken; en GET LIST voor het inlezen. PL1 is de computertaal waaruit PROSIM ontwikkeld is. Zie hiervoor ook hoofdstuk 2 over het gebruik van het programma.

VERSIE = Situatie die gesimuleerd wordt.
1= Beginsituatie.
2= Tussenfase geen export, wel pijpleiding.
3= Eindsituatie 3 boeien.
4= Eindsituatie 4 boeien.

NUMRUNS = Aantal jaren dat gesimuleerd wordt.

NUMBOEI = Aantal boeien, dit moet overeenstemmen met de waarde voor VERSIE.

X1,X2 = Gemiddelde en standaardafwijking van tussentijden voor slecht weer perioden van 2 a 3 dagen, behorend bij verdeling RAND(1).

X3,X4 = Gemiddelde en standaardafwijking van tussentijden voor slecht weer perioden van 1 dag, behorend bij verdeling RAND(2).

X5,X6,X7 = Gemiddelde tussenaankomsttijden Erlang_k verdeeld, voor respectievelijk gasoline gas oil en fuel oil schepen. Deze waarden zijn in het huidige model niet meer nodig, maar zijn uit eerdere modellen afkomstig. In de cross reference, bijlage 2, is ook te zien dat ze verder niet in het model voorkomen.

X8,X9,X10 = Gemiddelde tussenaankomsttijden voor resp.
middle, big en small crude schepen.

```
140 1 0 RESHAPE RAND(1) WITH PARAMETERS MEAN(X1) DEVIATION(X2);
141 1 0 RESHAPE RAND(2) WITH PARAMETERS MEAN(X3) DEVIATION(X4);
142 1 0 RESHAPE INHOUD(4) WITH PARAMETERS CLASSWIDTH(3E4)
      L_BOUND(0) N_CLASSES(35);
143 1 0 NUMSHIP=0;
```

(140-142):

Hier worden de parameterwaarden veranderd voor de randomstreams RAND(1) en RAND(2). Ook worden de afmetingen van het histogram INHOUD(4) voor crude aangepast.

(143):

NUMSHIP = Aantal crude schepen dat tegelijkertijd crude onttrekt aan de crude tank. Dit is slechts een startwaarde.

```
144 1 0 IF VERSIE=1 THEN ACTIVATE GENERATOR4 FROM STARTGEN4;
145 1 0 IF VERSIE>2 THEN DO;
146 1 1   ACTIVATE GENERATOR5 FROM STARTGEN5;
147 1 1   ACTIVATE GENERATOR6 FROM STARTGEN6;
148 1 1   ACTIVATE GENERATOR7 FROM STARTGEN7;
149 1 1 END;
```

(144-149):

Afhankelijk van de versie, worden de generatoren geactiveerd vanaf hun procesbeschrijvingen.

GENERATOR4 = Generator van import crude schepen.

GENERATOR5 = Generator van crude schepen 80000 DWT.

GENERATOR6 = Generator van crude schepen 135000 DWT.

GENERATOR7 = Generator van crude schepen 50000 DWT.

```
150 1 0 ACTIVATE HMASTER FROM STARTWORK;
151 1 0 ACTIVATE NOTEERDER FROM NOTEREN;
152 1 0 ACTIVATE VERZIEKER FROM VERZIEKEN;
153 1 0 PUT SKIP EDIT ('GEGEVENS PRODUKTANKS: GASOLINE,GAS OIL,FUEL OIL')
      (A);
154 1 0 PUT SKIP LIST ('INPUT (M3/H)', 'OUTPUT (M3/H)', 'BEGININHOUD (M3)',
      'TANKGROOTTE (M3)');
```

(150-152):

Ook de HMASTER, de NOTEERDER en de VERZIEKER worden aan het werk gezet.

HMASTER = Havenmeester die de wachtende schepen naar de onbezette boeien loodst.

NOTEERDER = Noteerder noteert tankinhouden op gezette tijden in histogrammen en grafieken.

VERZIEKER = De raffinaderij die aangeeft wanneer deze wel en wanneer niet in bedrijf is (streamdays).

(153-154):

Print opdrachten voor tabelhoofden voor de waarden van de produkt tanks.

```

155 1 0 FOR I=1 TO 3 DO;
158 1 1 NEWTANK=NEW TANK;
159 1 1 NEWTANK.NR=I;
160 1 1 NEWTANK.LOGIC=TRUE;
161 1 1 GET LIST (NEWTANK.B_INPUT,NEWTANK.INPUT,NEWTANK.OUTPUT,
NEWTANK.HULP,NEWTANK.CAPACITY);
162 1 1 PUT SKIP LIST (NEWTANK.INPUT,NEWTANK.OUTPUT,
NEWTANK.HULP,NEWTANK.CAPACITY);
163 1 1 NEWTANK.LADING=NEWTANK.HULP;
165 1 1 PUT NEWTANK TO TAIL OF OPSLAG_ROW;
166 1 1 ACTIVATE NEWTANK FROM STOCKING;
168 1 1 END;

```

(155-168):

In deze herhalingsopdracht worden 3 produkt tanks gemaakt voor gasoline, gas oil en fuel oil. Statement 158 zorgt voor de creatie en vervolgens krijgen de attributen een waarde, die van de dataset worden afgelezen (161). Tenslotte wordt de tank in de OPSLAG_ROW gezet en geactiveerd (165-166).

I = Teller.
NEWTANK = Referentie naar een TANK.
NEW = PROSIM keyword, creëert een klasse component.
TANK = Produkt tank, klasse component.
Attributen van de TANK.
NR = Soort produkt dat in de tank zit.
1= gasoline 2= gas oil 3= fuel oil
LOGIC = Logische waarde die aangeeft of er wel of geen schip besteld mag worden.
TRUE= ja en FALSE= nee.
B_INPUT = Hoeveelheid produkt wat de raffinaderij aan de tank levert . m3/h
INPUT = Totale instroming in de tank. m3/h
OUTPUT = Totale uitstroming uit de tank. m3/h
HULP = Hulp variabele die nodig is om de continue variabele LADING een nieuwe waarde te geven.
CAPACITY = Tankgrootte. m3
LADING = Continue variabele, inhoud van de tank. m3

```

169 1 0 NEWC_TANK=NEW C_TANK;
170 1 0 PUT SKIP (2);
171 1 0 PUT SKIP EDIT ('POMP KARAKTERISTIEK: INPUT!=A1+') (A);
172 1 0 PUT EDIT ('C1*EXP(-C2*INHOUD)') (A);
173 1 0 PUT SKIP LIST ('A1','C1','C2','C3');
174 1 0 GET LIST (NEWC_TANK.A1,NEWC_TANK.C1,NEWC_TANK.C2,NEWC_TANK.C3,
NEWC_TANK.C_OUTPUT,NEWC_TANK.C_VULLEN,NEWC_TANK.C_CAPACITY);
175 1 0 PUT SKIP LIST (NEWC_TANK.A1,NEWC_TANK.C1,NEWC_TANK.C2,
NEWC_TANK.C3);
176 1 0 PUT SKIP (2);
177 1 0 PUT SKIP EDIT ('CRUDETANK WAARDEN :') (A);
178 1 0 PUT SKIP LIST ('OUTPUT (M3/H)','LOSSNELHEID (M3/H)',
'TANKGROOTTE (M3)');
179 1 0 PUT SKIP LIST (NEWC_TANK.C_OUTPUT,NEWC_TANK.C_VULLEN,
NEWC_TANK.C_CAPACITY);
180 1 0 NEWC_TANK.A2=NEWC_TANK.A1;
181 1 0 NEWC_TANK.AMAX=4800-165.37;
182 1 0 PUT NEWC_TANK TO TAIL OF OPSLAG_ROW;
183 1 0 ACTIVATE NEWC_TANK FROM C_STOCKING;

```

(169-183):

Na de creatie van een opslagtank voor ruwe olie (169), volgen opdrachten voor het inlezen en afdrukken van de pompkarakteristiek van de oliepijpleiding en van zijn coëfficiënten (170-175). Dan volgen de waarden voor de crude tank zelf (176-181). Verder wordt de C_TANK in de OPSLAG_ROW gezet en geactiveerd.

Alle tanks worden om programmeer technische redenen in een rij gezet. PROSIM heeft namelijk een aantal makkelijke handelingen, waarvoor de te behandelen componenten in een rij moeten staan. Zie hiervoor ook het PROSIM tekstboek.

NEWC_TANK = Referentie naar een C_TANK.

C_TANK = Ruwe olie tank.

Attributen van de C_TANK.

A1,C1,C2,C3= Coëfficiënten van de pompkarakteristiek.

C_OUTPUT = Totale uitstroming uit de tank. m3/h

C_VULLEN = Maximale laadsnelheid van een export tanker. m3/h

C_CAPACITY = Tank grootte. m3

A2 = Hulp variabele om de oude waarde van A1 niet te verliezen, als de waarde van A1 verandert.

AMAX = Maximaal mogelijke uitstroming uit de C_TANK indien deze bijna leeg is ($0.1 * C_CAPACITY$). Deze is dus gelijk aan de instroming min de uitstroming naar de raffinaderij. m3/h

```

184 1 0 FOR K=1 TO NUMRUNS DO;
187 1 1 WHILE NOW<(150+(K-1)*365) DAYS DO;
188 1 2 LOG1=TRUE;
189 1 2 WAIT RAND(1) DAYS;
191 1 2 LOG1=FALSE;
192 1 2 WAIT (2+(UNIF1(8)<0.5)) DAYS;
195 1 2 END;
196 1 1 WHILE NOW<(210+(K-1)*365) DAYS DO;
197 1 2 LOG1=TRUE;
198 1 2 WAIT RAND(2) DAYS;
200 1 2 LOG1=FALSE;
201 1 2 WAIT 1 DAY;
204 1 2 END;
205 1 1 LOG1=TRUE;
206 1 1 WAIT 365*K DAYS-NOW;
```

(184-206):

K = Teller.

NOW = PROSIM keyword, stelt de simulatietijd voor (klok).

LOG1 = Logische waarde voor slecht weer, dan zijn de golven te hoog voor de sleepboten.
TRUE= goed weer FALSE= slecht weer.

Tijdens de eerste 150 dagen van een simulatiejaar hebben de slecht weer periodes een duur van 2 a 3 dagen

en zijn de tussentijden RAND(1) dagen (187-195).
De volgende 60 dagen zijn geheel analoog aan het voorgaande, maar met andere waarden. Namelijk RAND(2) voor de tussentijden en 1 dag voor de duur (196-204). De rest van het jaar zijn er geen slecht weer perioden, dus LOG1=TRUE (205-206).

RAND(1) = Trekking uit de verdeling RAND(1).
RAND(2) = Trekking uit de verdeling RAND(2).
UNIF1(8) = Trekking uit een uniforme verdeling.
Alle verdelingen worden in het definitie gedeelte gedefinieerd.

```
208 1 1 IF K=1 THEN DO;
209 1 2   FOR O=1 TO 4 DO;
212 1 3     CLEAR INHOUD(O);
214 1 3   END;
215 1 2 END;
```

(208-215):

Het eerste jaar is een aanlooperperiode. Aan het einde ervan worden de histogrammen voor de inhouden van de tanks schoongeveegd met behulp van het statement CLEAR (212).

O = Teller.
INHOUD(O) = Histogrammen voor de tankinhouden.
1= gasoline 2= gas oil 3= fuel oil
4= crude.

```
216 1 1 IF K>1 THEN DO;
217 1 2 IF VERSIE=1 THEN DO;
218 1 3   FOR O=1 TO 4 DO;
221 1 4     TALLY WACHTTIJD(O).MEAN IN W_TIMES(O);
222 1 4     CLEAR WACHTTIJD(O);
224 1 4   END;
225 1 3 END;ELSE DO;
227 1 3   FOR O=1 TO 7 DO;
230 1 4     IF O=4 THEN TALLY WACHTTIJD(O).MEAN IN W_TIMES(O);
231 1 4     CLEAR WACHTTIJD(O);
233 1 4   END;
234 1 3 END;
```

(216-235):

Na het eerste jaar wordt afhankelijk van de versie (217,225), middels de statement TALLY, de gemiddelde wachttijd van een schip in W_TIMES(O) gezet (221,230) en wordt WACHTTIJD(O) schoongeveegd (222,231).

WACHTTIJD(O) = Histogram van de wachttijden in 1 jaar.
1= gasoline 2= gas oil 3= fuel oil
4= import crude 5= export 80000 DWT
6= export 135000 DWT 7= export 50000 DWT
W_TIMES(O) = Histogram voor de jaargemiddelde wachttijden van de verschillende schepen. Zelfde volgorde.

MEAN = Systeem attribuut, gemiddelde van een histogram.

```
235 1 2   END;
237 1 1   END;
238 1 0   PRINT INHOUD(1) HEADED BY 'GASOLINE';
239 1 0   PRINT INHOUD(2) HEADED BY 'GAS OIL';
240 1 0   PRINT INHOUD(3) HEADED BY 'FUEL OIL';
241 1 0   PRINT INHOUD(4) HEADED BY 'CRUDE';
242 1 0   PRINT W_TIMES(1) HEADED BY 'WACHTTIJDEN GASOLINE';
243 1 0   PRINT W_TIMES(2) HEADED BY 'WACHTTIJDEN GAS OIL';
244 1 0   PRINT W_TIMES(3) HEADED BY 'WACHTTIJDEN FUEL';
245 1 0   IF VERSIE=1 THEN PRINT W_TIMES(4) HEADED BY
        'WACHTTIJDEN IMPORT CRUDE';
246 1 0   ELSE DO;
247 1 1     PRINT W_TIMES(7) HEADED BY 'WACHTTIJDEN SMALL CRUDE';
248 1 1     PRINT W_TIMES(5) HEADED BY 'WACHTTIJDEN MIDDLE CRUDE';
249 1 1     PRINT W_TIMES(6) HEADED BY 'WACHTTIJDEN BIG CRUDE';
250 1 1   END;
251 1 0   PRINT BERGING(4) HEADED BY 'CRUDE' AX POSITION 0;
```

(238-251):

Deze printopdrachten spreken voor zichzelf. HEADED BY 'xxxx' zorgt voor een kop van het histogram of de grafiek.

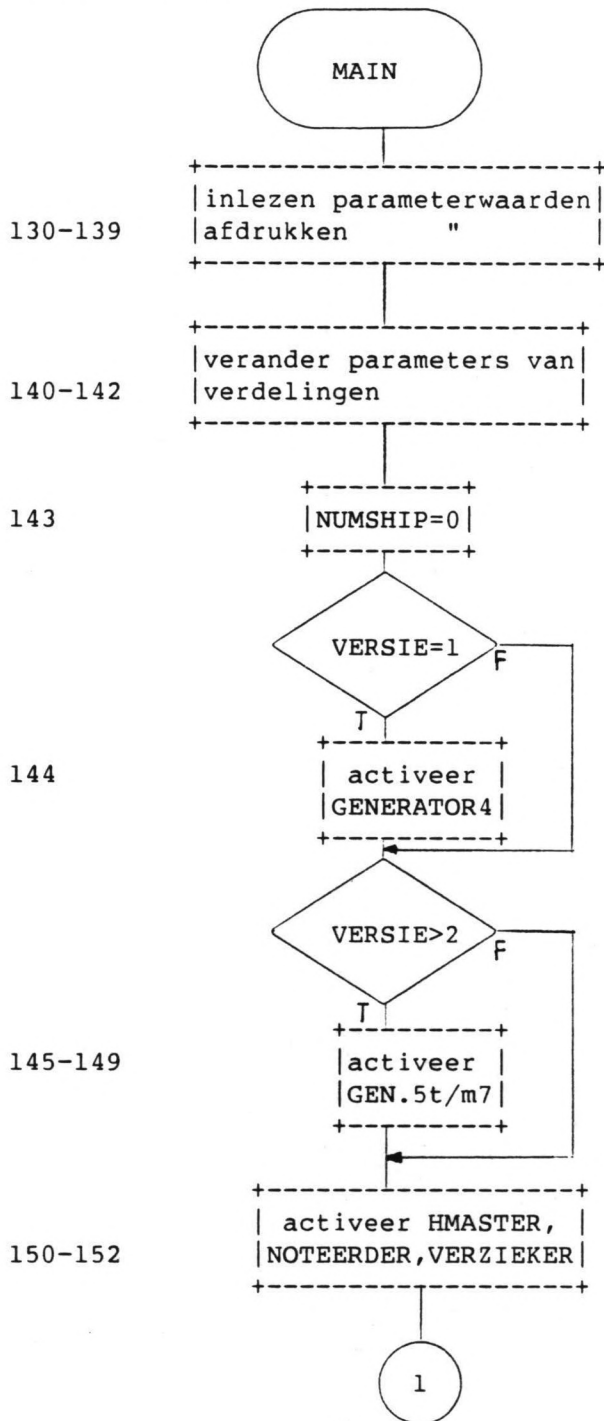
BERGING(4) = Grafiek van de inhoud van de C_TANK.

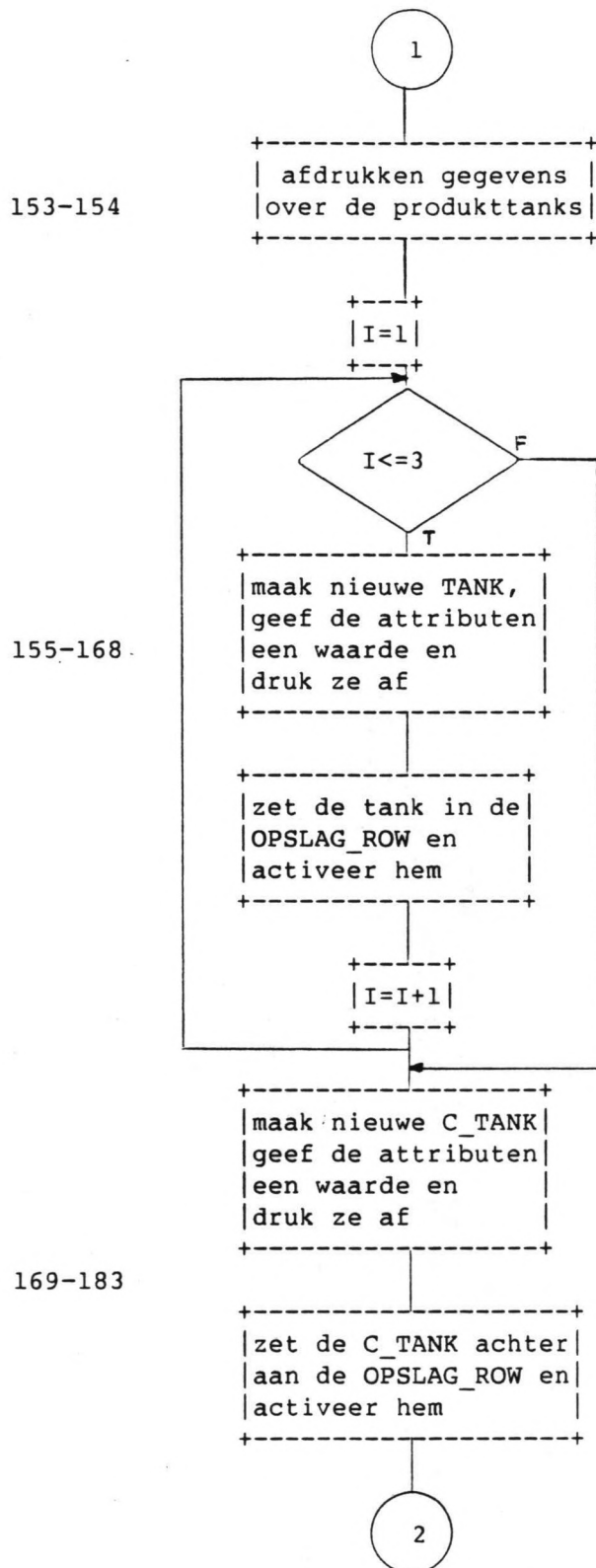
```
252 1 0   CANCEL ALL;
253 1 0   TERMINATE;
```

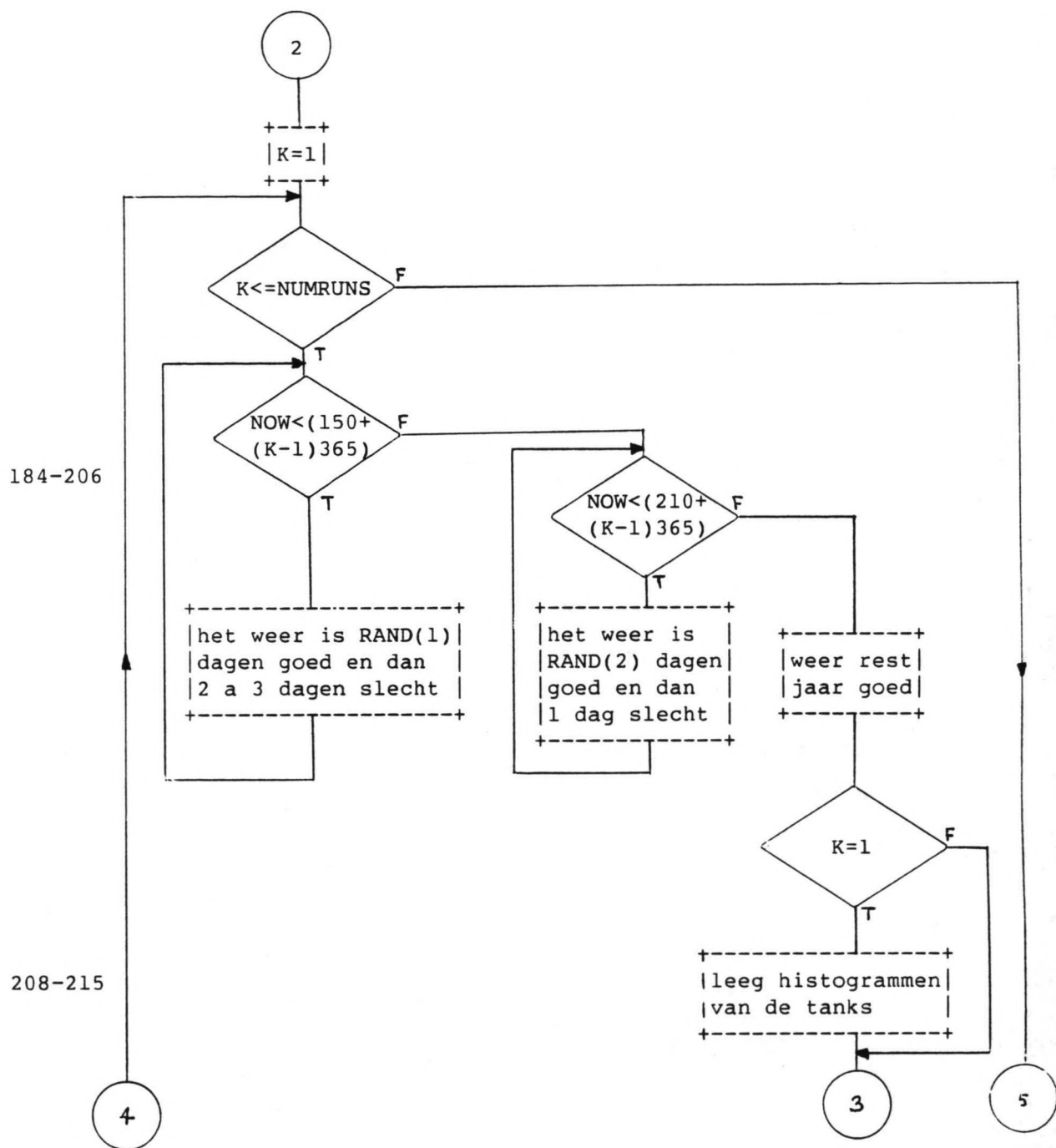
(252-253):

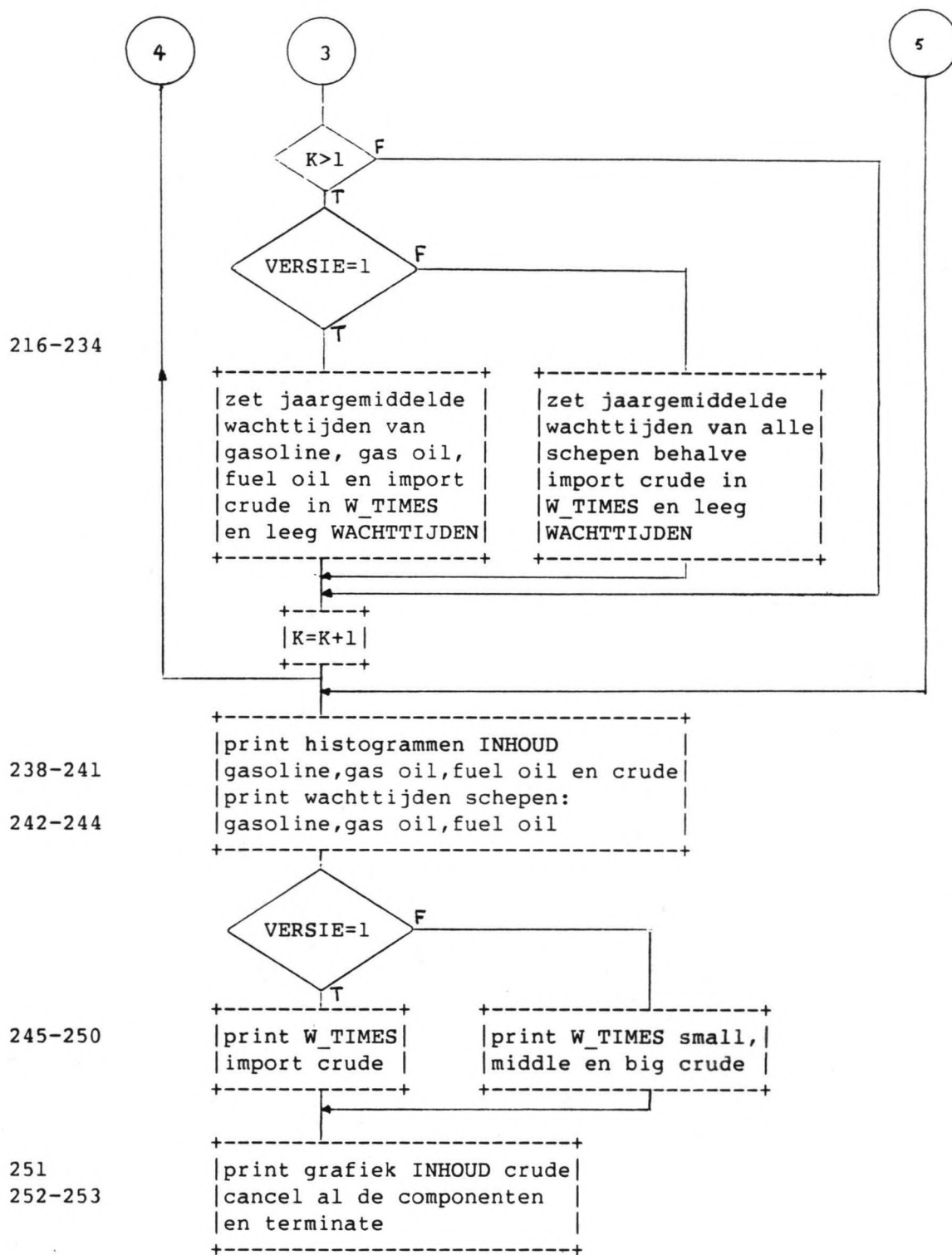
Met CANCEL ALL (252) worden alle componenten gepassiveerd. Met TERMINATE (253) wordt MAIN een data component en daarmee eindigt het programma.

1.1.2. Stroomschema.









1.2. De GENERATOREN 1 t/m 7.

1.2.1. Procesbeschrijvingen (254-358).

```
254 1 0 STARTGEN1:
      WAIT (5+2*UNIF2(1)) DAYS;
256 1 0 P=UNIF1(1);
257 1 0 NEXTSHIP=NEW SHIP;
258 1 0 NEXTSHIP.GENNR=1;
259 1 0 NEXTSHIP.PROD=1;
260 1 0 NEXTSHIP.CONTENCE=13000+3500*P;
261 1 0 NEXTSHIP.LOSSPEED=347;
262 1 0 NEXTSHIP.MOORINGTIME=4.5+P;
263 1 0 NEXTSHIP.BERTH=2;
264 1 0 PUT NEXTSHIP TO TAIL OF AANLEG_ROW;
265 1 0 TERMINATE;

266 1 0 STARTGEN2:
      WAIT (5+2*UNIF2(2)) DAYS;
268 1 0 P=UNIF1(2);
269 1 0 NEXTSHIP=NEW SHIP;
270 1 0 NEXTSHIP.GENNR=2;
271 1 0 NEXTSHIP.PROD=2;
272 1 0 NEXTSHIP.CONTENCE=10400+3000*P;
273 1 0 NEXTSHIP.LOSSPEED=296;
274 1 0 NEXTSHIP.MOORINGTIME=4.5+P;
275 1 0 NEXTSHIP.BERTH=2;
276 1 0 PUT NEXTSHIP TO TAIL OF AANLEG_ROW;
277 1 0 TERMINATE;

278 1 0 STARTGEN3:
      WAIT (5+2*UNIF2(3)) DAYS;
280 1 0 P=UNIF1(3);
281 1 0 NEXTSHIP=NEW SHIP;
282 1 0 NEXTSHIP.GENNR=3;
283 1 0 NEXTSHIP.PROD=3;
284 1 0 NEXTSHIP.CONTENCE=15000+4700*P;
285 1 0 NEXTSHIP.LOSSPEED=1579;
286 1 0 NEXTSHIP.MOORINGTIME=4.8+1.4*P;
287 1 0 NEXTSHIP.BERTH=4;
288 1 0 PUT NEXTSHIP TO TAIL OF AANLEG_ROW;
289 1 0 TERMINATE;
```

De procesbeschrijvingen van de eerste drie generatoren (254-289) zijn qua structuur precies hetzelfde.

Na te zijn geactiveerd, wacht de GENERATOR tussen de 5 en 7 dagen (254,266,278). Er wordt een trekking gedaan uit een uniforme verdeling (256,268,280) en het schip wordt gemaakt (257,269,281). Dan krijgen de attributen een waarde, waarvan er een aantal afhankelijk zijn van de waarde P. Tenslotte zet de GENERATOR het schip achter aan de AANLEG_ROW (264,276,288) en maakt zichzelf een data component (265,277,289).

UNIF1(1t/m3) = Trekkingen uit uniforme verdelingen.

UNIF2(1t/m3) = Trekkingen uit uniforme verdelingen.

P = Hulpvariabele, anders zou steeds een nieuwe trekking gedaan worden wanneer we alleen UNIF1 zouden gebruiken, terwijl de aanlegtijd afhankelijk is van de grootte van het schip.

NEXTSHIP = Referentie naar een schip.

SHIP = Een schip.

Attributen van SHIP.

GENNR = Generator nummer.
 PROD = Produktsoort. 1= gasoline 2= gas oil
 3= fuel oil 4= crude.
 CONTENCE = Ladinggrootte in m3.
 LOSSPEED = Maximum lossnelheid.
 MOORINGTIME = Aanlegtijd, van AANLEG_ROW naar
 BOEI_ROW.
 BERTH = Mogelijke aanleg boeien.
 1= alleen boei 1; 2= boei 2 of 4
 3= boei 1,2 of 3; 4= alle boeien.
 AANLEG_ROW = Wachtrij voor de schepen indien zij
 niet direct aan een boei kunnen liggen.

```

290 1 0 STARTGEN4:
      WAIT (365/13.8345) DAYS;
292 1 0 P=UNIF1(4);
293 1 0 NEXTSHIP= NEW SHIP;
294 1 0 NEXTSHIP.GENNR=4;
295 1 0 NEXTSHIP.PROD=4;
296 1 0 NEXTSHIP.CONTENCE=94675;
297 1 0 NEXTSHIP.LOSSPEED=-3550;
298 1 0 NEXTSHIP.MOORINGTIME=5.9+1.2*P;
299 1 0 NEXTSHIP.BERTH=3;
300 1 0 PUT NEXTSHIP TO TAIL OF AANLEG_ROW;
301 1 0 REPEAT FROM STARTGEN4;
```

(290-301):

De procesbeschrijving van GENERATOR4 is bijna hetzelfde als van de voorgaande processen. Dit is alleen een herhalingsproces.

De GENERATOR wacht 365/13.8345 dagen (290), maakt dan een schip met zijn attributen (293-299) en zet het in de AANLEG_ROW (300). Hierna begint hij weer opnieuw van voren af aan (301).

De waarde voor LOSSPEED is hier negatief. Dit is nodig omdat, in tegenstelling tot de export crude schepen, dit schip zijn ruwe olie aan de C_TANK levert in plaats van onttrekt (Zie hiervoor ook de procesbeschrijving C_ARRIVAL).

UNIF1(4) = Trekking uit uniforme verdeling.

```

302 1 0 STARTGEN5:
      AD=0;
303 1 0   FOR B=1 TO 2 DO;
306 1 1     AD=AD + LOG(UNIF2(4));
308 1 1   END;
309 1 0   WAIT -(X8/2)*AD DAYS;
311 1 0   P=UNIF1(5);
312 1 0   NEXTSHIP=NEW SHIP;
313 1 0   NEXTSHIP.GENNR=5;
314 1 0   NEXTSHIP.PROD=4;
315 1 0   NEXTSHIP.CONTENCE=94675;
316 1 0   NEXTSHIP.LOSSPEED=4734;
317 1 0   NEXTSHIP.MOORINGTIME=5.8+1.2*P;
318 1 0   NEXTSHIP.BERTH=3;
319 1 0   PUT NEXTSHIP TO TAIL OF AANLEG_ROW;
320 1 0   REPEAT FROM STARTGEN5;

```

```

321 1 0 STARTGEN6:
      AD=0;
322 1 0   FOR B=1 TO 2 DO;
325 1 1     AD=AD + LOG(UNIF2(5));
327 1 1   END;
328 1 0   WAIT -(X9/2)*AD DAYS;
330 1 0   P=UNIF1(6);
331 1 0   NEXTSHIP=NEW SHIP;
332 1 0   NEXTSHIP.GENNR=6;
333 1 0   NEXTSHIP.PROD=4;
334 1 0   NEXTSHIP.CONTENCE=135000/0.845;
335 1 0   NEXTSHIP.LOSSPEED=4734;
336 1 0   NEXTSHIP.MOORINGTIME=7.0+1.5*P;
337 1 0   NEXTSHIP.BERTH=1;
338 1 0   PUT NEXTSHIP TO TAIL OF AANLEG_ROW;
339 1 0   REPEAT FROM STARTGEN6;

```

```

340 1 0 STARTGEN7:
      AD=0;
341 1 0   FOR B=1 TO 2 DO;
344 1 1     AD=AD + LOG(UNIF2(6));
346 1 1   END;
347 1 0   WAIT -(X10/2)*AD DAYS;
349 1 0   P=UNIF1(7);
350 1 0   NEXTSHIP=NEW SHIP;
351 1 0   NEXTSHIP.GENNR=7;
352 1 0   NEXTSHIP.PROD=4;
353 1 0   NEXTSHIP.CONTENCE=50000/0.845;
354 1 0   NEXTSHIP.LOSSPEED=4734;
355 1 0   NEXTSHIP.BERTH=3;
356 1 0   NEXTSHIP.MOORINGTIME=5.2+1.1*P;
357 1 0   PUT NEXTSHIP TO TAIL OF AANLEG_ROW;
358 1 0   REPEAT FROM STARTGEN7;

```

(302-358):

De GENERATOREN 5 t/m 7 hebben ook eenzelfde structuur. Zij verschillen alleen in de afmetingen van de schepen. De procesbeschrijvingen beginnen met een trekking uit een Erlang-2 verdeling voor de tussenaankomsttijd (302-309,321-328,340-347). Verder doen zij precies hetzelfde als GENERATOR4, maar nu met een positieve waarde voor LOSSPEED.

UNIF1(4t/m6) = Trekking uit uniforme verdeling.

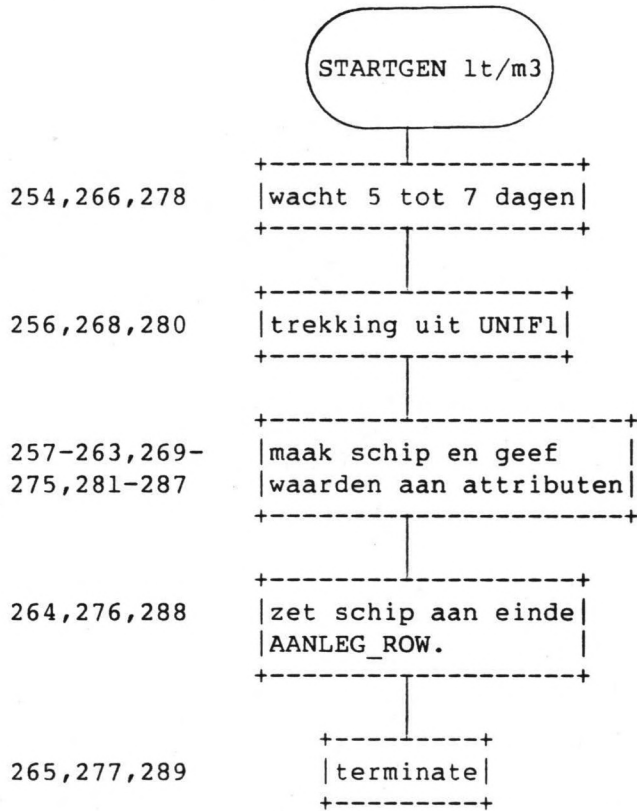
UNIF2(4t/m6) = Trekking uit uniforme verdeling.

AD = Hulpvariabele voor Erlang-k verdeling.

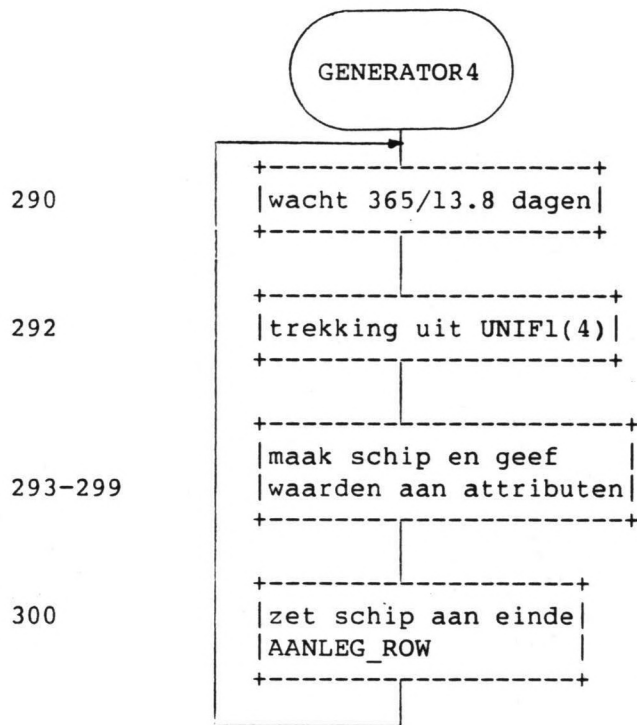
B = Teller. Ook k waarde voor Erlang-k.

1.2.2. Stroonschema's.

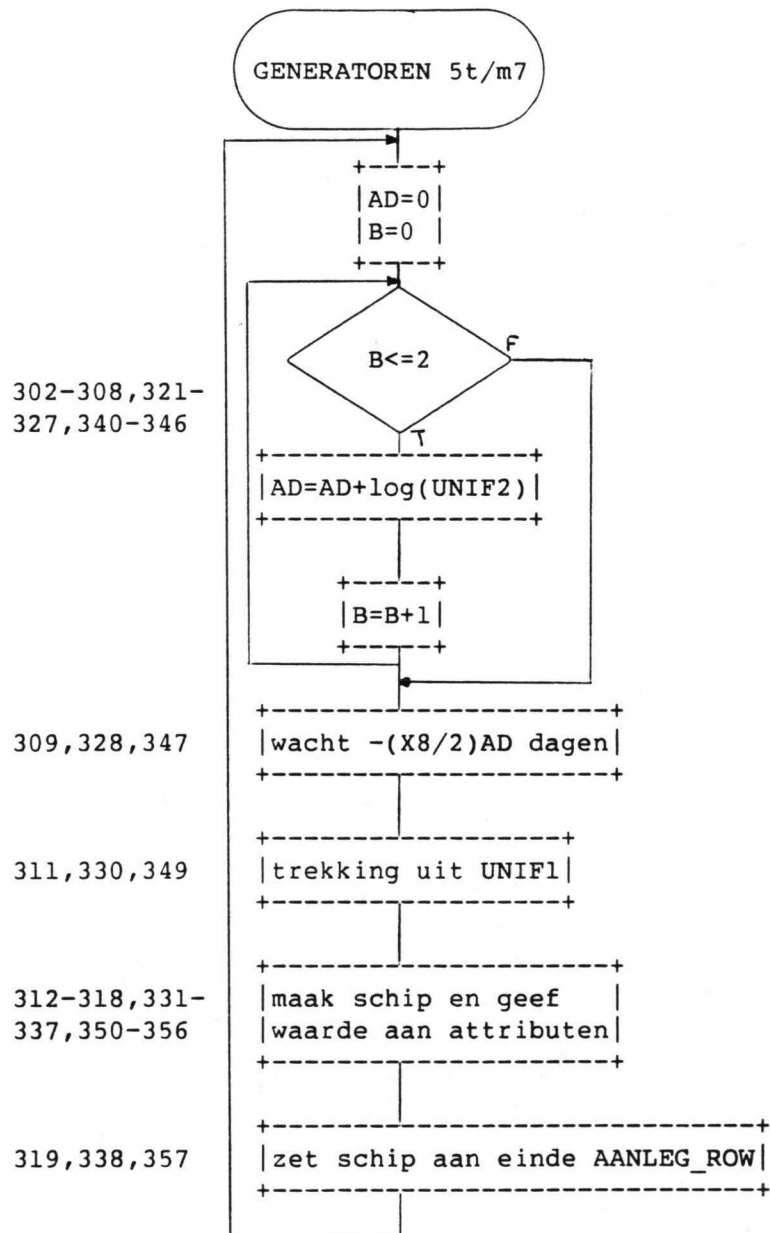
GENERATOREN 1 t/m 3



GENERATOR4



GENERATOREN 5 t/m 7



1.3. De HMASTER.

1.3.1. Procesbeschrijving (359-415).

De procesbeschrijving van de HMASTER is een activiteitenbeschrijving. Dit betekent dat iedere activiteit pas plaats vindt, wanneer er aan een aantal voorwaarden is voldaan.

```

359 1 0 STARTWORK:
      WAIT WHILE AANLEG_ROW IS EMPTY;
361 1 0 IF VERSIE=4 THEN WAIT WHILE (BOEI_ROW(1) IS NOT EMPTY &
      BOEI_ROW(2) IS NOT EMPTY & BOEI_ROW(3) IS NOT EMPTY &
365 1 0 BOEI_ROW(4) IS NOT EMPTY);ELSE WAIT WHILE (BOEI_ROW(1) IS NOT
      EMPTY & BOEI_ROW(2) IS NOT EMPTY & BOEI_ROW(3) IS NOT EMPTY);
369 1 0 WAIT WHILE LOG1=FALSE;

```

(359-369):

Hier vinden we een aantal van die voorwaarden:
Hij moet wachten indien de AANLEG_ROW leeg is. Dit betekent dat er geen schepen liggen te wachten (359).
Als VERSIE=4, dan moet hij alleen wachten als alle 4 de boeien bezet zijn, dus allemaal niet leeg. Anders moet hij alleen wachten als de 3 boeien bezet zijn (361).
Hij moet verder blijven wachten als LOG1=FALSE, dus als de sleepboten niet kunnen werken.

Als aan al deze voorwaarden voldaan is, dan is zijn strategie als volgt. Hij handelt achtereenvolgens de boeirijen 4, 1, 2 en 3 af. Na BOEI_ROW(3) begint hij weer van voren af aan (415).

```

371 1 0 IF (BOEI_ROW(4) IS EMPTY & VERSIE=4) THEN DO;
372 1 1 SHIP_IN_HAND=FIRST SHIP IN AANLEG_ROW WITH BERTH=2;
373 1 1 IF SHIP_IN_HAND=NOCOMP THEN SHIP_IN_HAND=FIRST SHIP IN
      AANLEG_ROW WITH BERTH=4;
374 1 1 IF SHIP_IN_HAND~NOCOMP THEN DO;
375 1 2 REMOVE SHIP_IN_HAND FROM AANLEG_ROW;
376 1 2 PUT SHIP_IN_HAND TO TAIL OF BOEI_ROW(4);
377 1 2 IF SHIP_IN_HAND.PROD=4 THEN ACTIVATE SHIP_IN_HAND FROM
378 1 2 C_ARRIVAL;ELSE ACTIVATE SHIP_IN_HAND FROM ARRIVAL;
379 1 2 END;
380 1 1 END;

```

(371-380):

Indien we met VERSIE=4 werken, dan komt de HMASTER het eerst bij BOEI_ROW(4) (371). Als deze boei onbezet is, dan zoekt hij een schip uit de AANLEG_ROW, dat aan deze boei behandeld kan worden. Dit moet een schip zijn met BERTH=2 (372) of BERTH=4 (373). Indien hij een schip vindt, dan loodst hij dit schip van de AANLEG_ROW naar BOEI_ROW(4) (374-376). Daarna activeert hij het schip vanaf ARRIVAL (produkt) of vanaf C_ARRIVAL (crude) (377-378).

SHIP_IN_HAND = referentie naar SHIP.
NOCOMP = PROSIM woord, geen component aanwezig.
BOEI_ROW 1t/m4 = Aanlegboeien.

```

381 1 0 IF BOEI_ROW(1) IS EMPTY THEN DO;
382 1 1 SHIP_IN_HAND=FIRST SHIP IN AANLEG_ROW WITH BERTH=1;
383 1 1 IF SHIP_IN_HAND=NOCOMP THEN SHIP_IN_HAND=FIRST SHIP IN
AANLEG_ROW WITH BERTH=3;
384 1 1 IF SHIP_IN_HAND=NOCOMP THEN SHIP_IN_HAND=FIRST SHIP IN
AANLEG_ROW WITH BERTH=4;
385 1 1 IF SHIP_IN_HAND=NOCOMP THEN DO;
386 1 2 REMOVE SHIP_IN_HAND FROM AANLEG_ROW;
387 1 2 PUT SHIP_IN_HAND TO TAIL OF BOEI_ROW(1);
388 1 2 IF SHIP_IN_HAND.PROD=4 THEN ACTIVATE SHIP_IN_HAND FROM
389 1 2 C_ARRIVAL;ELSE ACTIVATE SHIP_IN_HAND FROM ARRIVAL;
390 1 2 END;
391 1 1 END;
392 1 0 IF BOEI_ROW(2) IS EMPTY THEN DO;
393 1 1 SHIP_IN_HAND=FIRST SHIP IN AANLEG_ROW WITH BERTH=2;
394 1 1 IF SHIP_IN_HAND=NOCOMP THEN SHIP_IN_HAND=FIRST
SHIP IN AANLEG_ROW WITH BERTH=1;
395 1 1 IF SHIP_IN_HAND=NOCOMP THEN DO;
396 1 2 REMOVE SHIP_IN_HAND FROM AANLEG_ROW;
397 1 2 PUT SHIP_IN_HAND TO TAIL OF BOEI_ROW(2);
398 1 2 IF SHIP_IN_HAND.PROD=4 THEN ACTIVATE SHIP_IN_HAND FROM
399 1 2 C_ARRIVAL;ELSE ACTIVATE SHIP_IN_HAND FROM ARRIVAL;
400 1 2 END;
401 1 1 END;
402 1 0 IF BOEI_ROW(3) IS EMPTY THEN DO;
403 1 1 SHIP_IN_HAND=FIRST SHIP IN AANLEG_ROW WITH BERTH>2;
404 1 1 IF SHIP_IN_HAND=NOCOMP THEN DO;
405 1 2 REMOVE SHIP_IN_HAND FROM AANLEG_ROW;
406 1 2 PUT SHIP_IN_HAND TO TAIL OF BOEI_ROW(3);
407 1 2 IF SHIP_IN_HAND.PROD=4 THEN ACTIVATE SHIP_IN_HAND FROM
408 1 2 C_ARRIVAL;ELSE ACTIVATE SHIP_IN_HAND FROM ARRIVAL;
409 1 2 END;
410 1 1 END;
411 1 0 IF SHIP_IN_HAND=NOCOMP THEN WAIT 1 HOUR;
415 1 0 REPEAT FROM STARTWORK;

```

(381-415):

Een soortgelijke opbouw vinden we bij de overige BOEI_ROW's. Ook daar houdt de HMASTER een bepaalde volgorde van selectie aan.

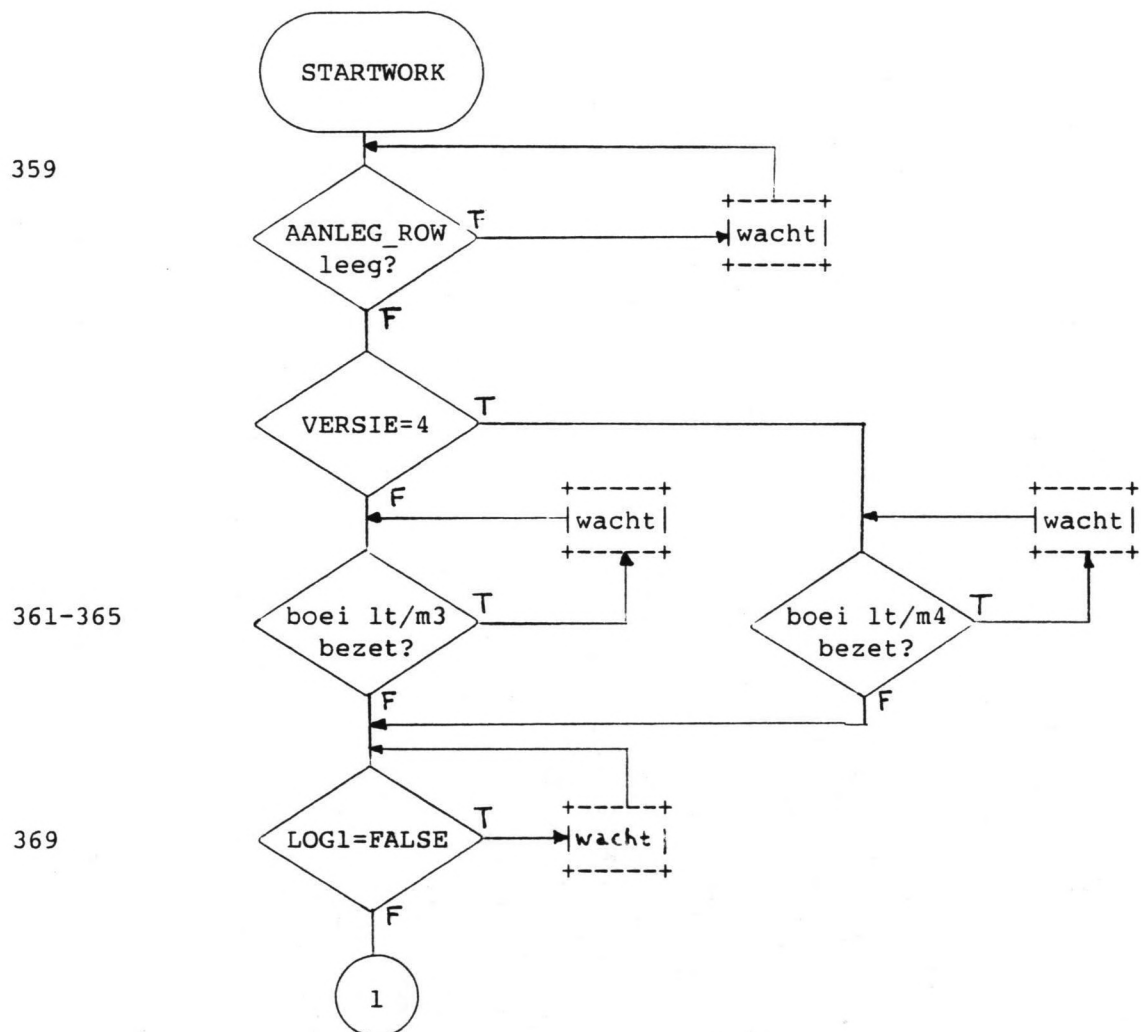
Voor BOEI_ROW(1) is deze volgorde: BERTH=1 (382), BERTH=3 (383) en BERTH=4 (384).

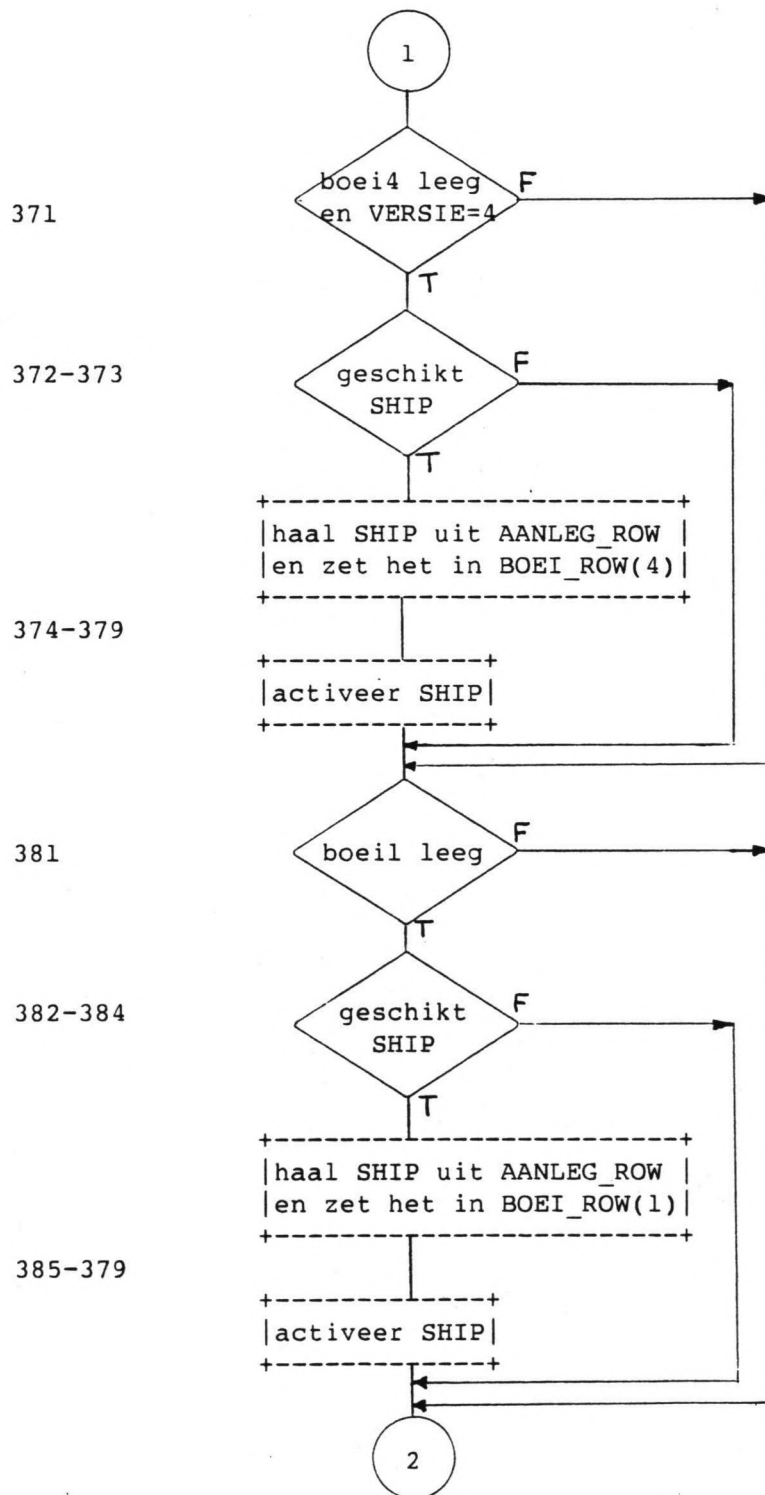
Voor BOEI_ROW(2): BERTH=2 (393) en BERTH=1 (394). We laten dus geen schepen met BERTH=3 of 4 aan deze boei liggen, terwijl dit eventueel wel mogelijk is. In deel 3, over de gevoeligheid van het programma, is deze restrictie losgelaten.

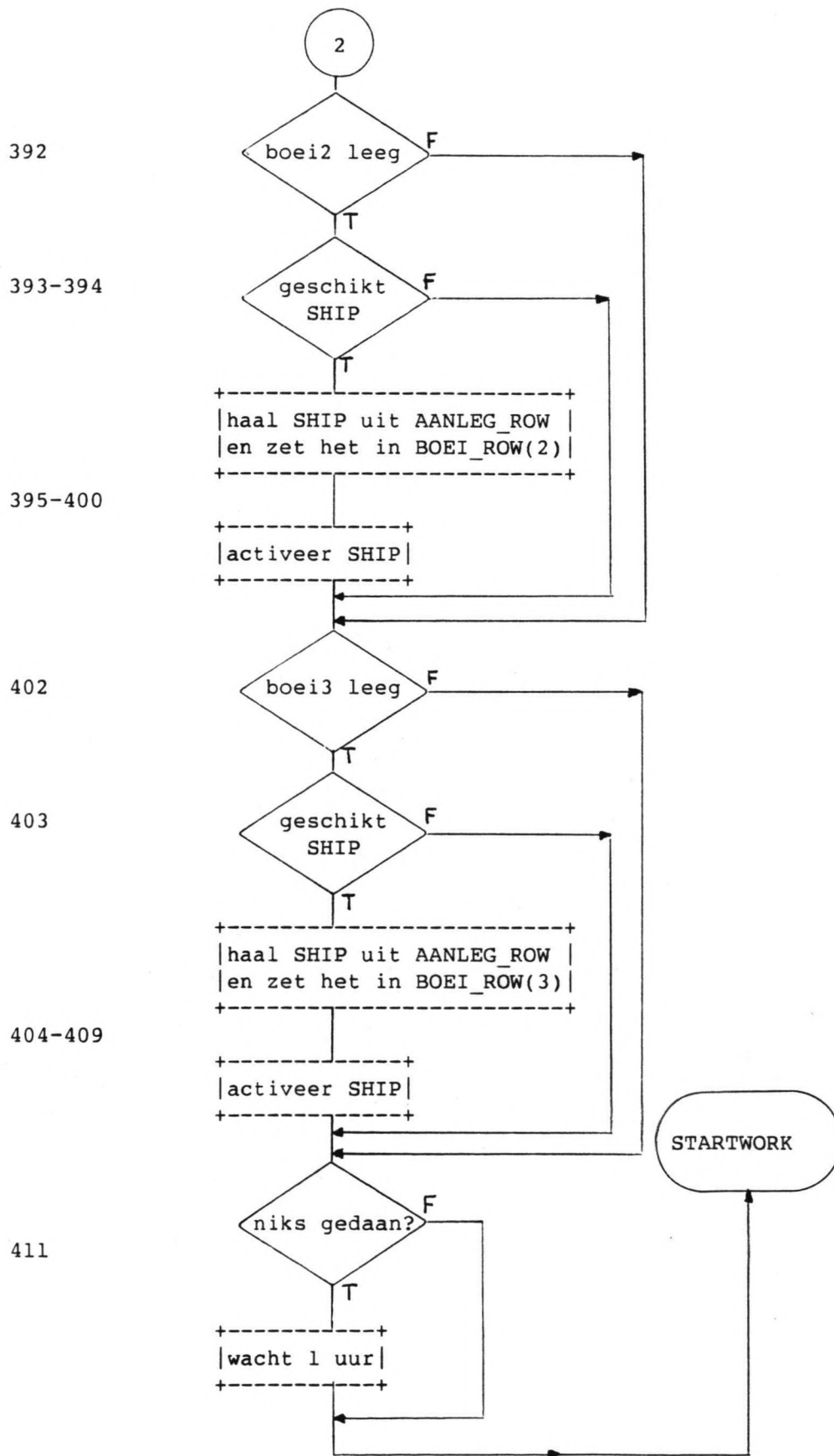
Voor BOEI_ROW(3): BERTH>2 (403).

Tenslotte zorgt statement 411 ervoor dat de HMASTER niet in een lus terecht komt, als hij geen geschikt schip kan vinden. Dit gebeurt als er een schip ligt te wachten, dat niet aan een onbezette boei geholpen kan worden.

1.3.2. Stroomschema.







1.4. Produkt SHIP.

1.4.1. Procesbeschrijving (416-441).

Vanaf het moment dat de HMASTER een produkt SHIP aan een BOEI_ROW legt, is die boei bezet. Dit betekent echter niet dat dan gelijk met lossen begonnen kan worden.

```
416 1 0 ARRIVAL:
      WAIT WHILE LOG1=FALSE;
419 1 0 TALLY NOW-ARRIVALTIME IN WACHTTIJD(GENNR);
420 1 0 WAIT MOORINGTIME;
```

(416-420):

Ten eerste kan SHIP niet naar de boei varen als het weer slecht is, dus als LOG1=FALSE (416). Wanneer het weer goed is, dan is het wachten afgelopen en kan de wachttijd in het histogram WACHTTIJD genoteerd worden (419). Dan duurt het nog de MOORINGTIME voordat er met lossen begonnen kan worden.

ARRIVALTIME = Systeem attribuut, tijdstip van creatie van een klasse component.
In dit geval een SHIP.

```
422 1 0 TANK_IN_HAND=FIRST TANK IN OPSLAG_ROW WITH NR=PROD;
423 1 0 TANK_IN_HAND.INPUT=TANK_IN_HAND.INPUT+LOSSPEED;
424 1 0 LEGEN=CONTENCE;
426 1 0 INTEGRATE WHILE (LEGEN>0 &
      TANK_IN_HAND.LADING<TANK_IN_HAND.CAPACITY);
```

(422-426):

Eerst wordt de TANK opgezocht waarin gelost gaat worden (422). De INPUT van deze TANK wordt opgehoogd met LOSSPEED (423) en LEGEN krijgt de waarde CONTENCE van SHIP (424). Dan wordt de integraal LEGEN'=-LOSSPEED geïntegreerd, zolang LEGEN>0 en LADING<CAPACITY.

TANK_IN_HAND = Referentie naar TANK.

LEGEN = Continue variabele, inhoud van de scheepslading. Diff. verg.
LEGEN'=-LOSSPEED.

INTEGRATE = PROSIM keyword, nu worden alle continue attributen van de current component geïntegreerd volgens hun diff. verg.


```
428 1 0 IF TANK_IN_HAND.LADING>TANK_IN_HAND.CAPACITY THEN DO;  
429 1 1   TANK_IN_HAND.INPUT=TANK_IN_HAND.INPUT-LOSSPEED;  
430 1 1   LOSSPEED=TANK_IN_HAND.OUTPUT-TANK_IN_HAND.B_INPUT;  
431 1 1   TANK_IN_HAND.INPUT=TANK_IN_HAND.INPUT+LOSSPEED;  
432 1 1   INTEGRATE WHILE LEGEN>0;  
434 1 1 END;
```

(428-434):

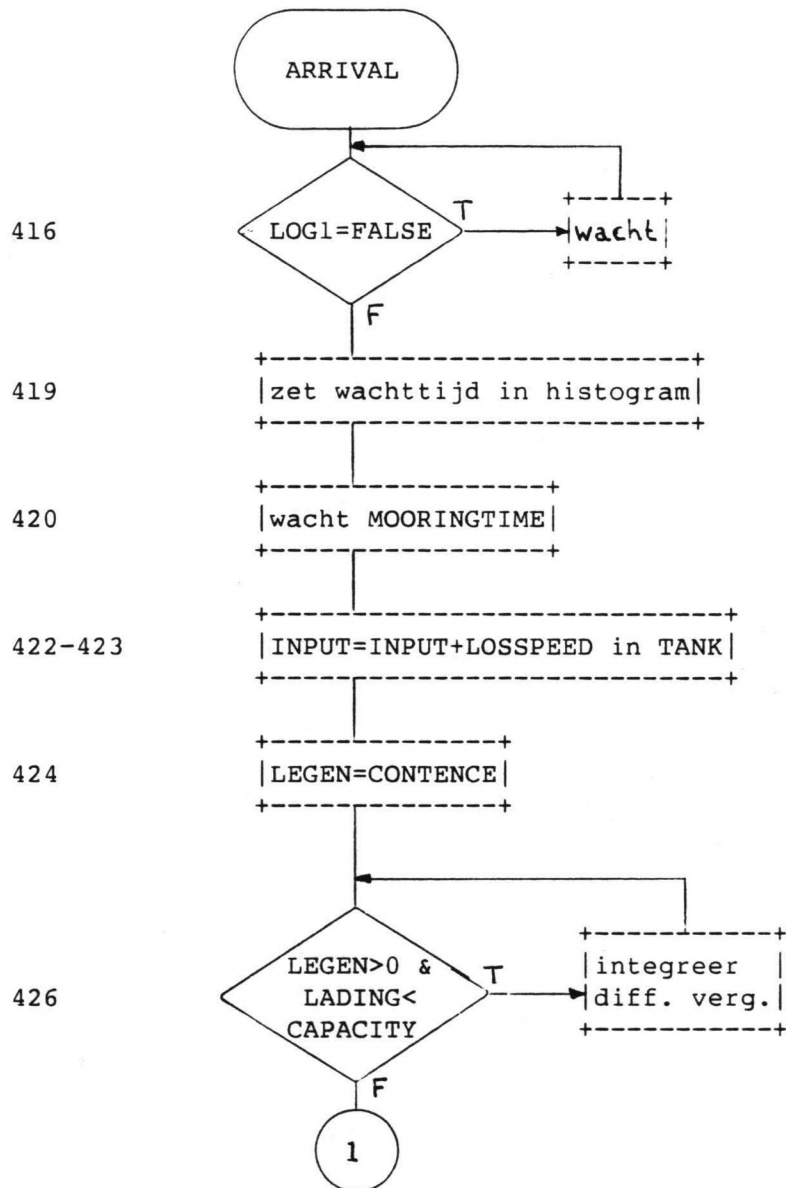
Als LADING groter wordt dan CAPACITY (428), dan moeten er maatregelen getroffen worden. Er moet voor gezorgd worden dat de INPUT in de TANK gelijk wordt aan de OUTPUT ervan, want dan blijft de LADING hetzelfde. Dit kan alleen als de waarde van LOSSPEED aangepast wordt (429-431). Met deze nieuwe waarde wordt SHIP verder leeg gepompt (432).

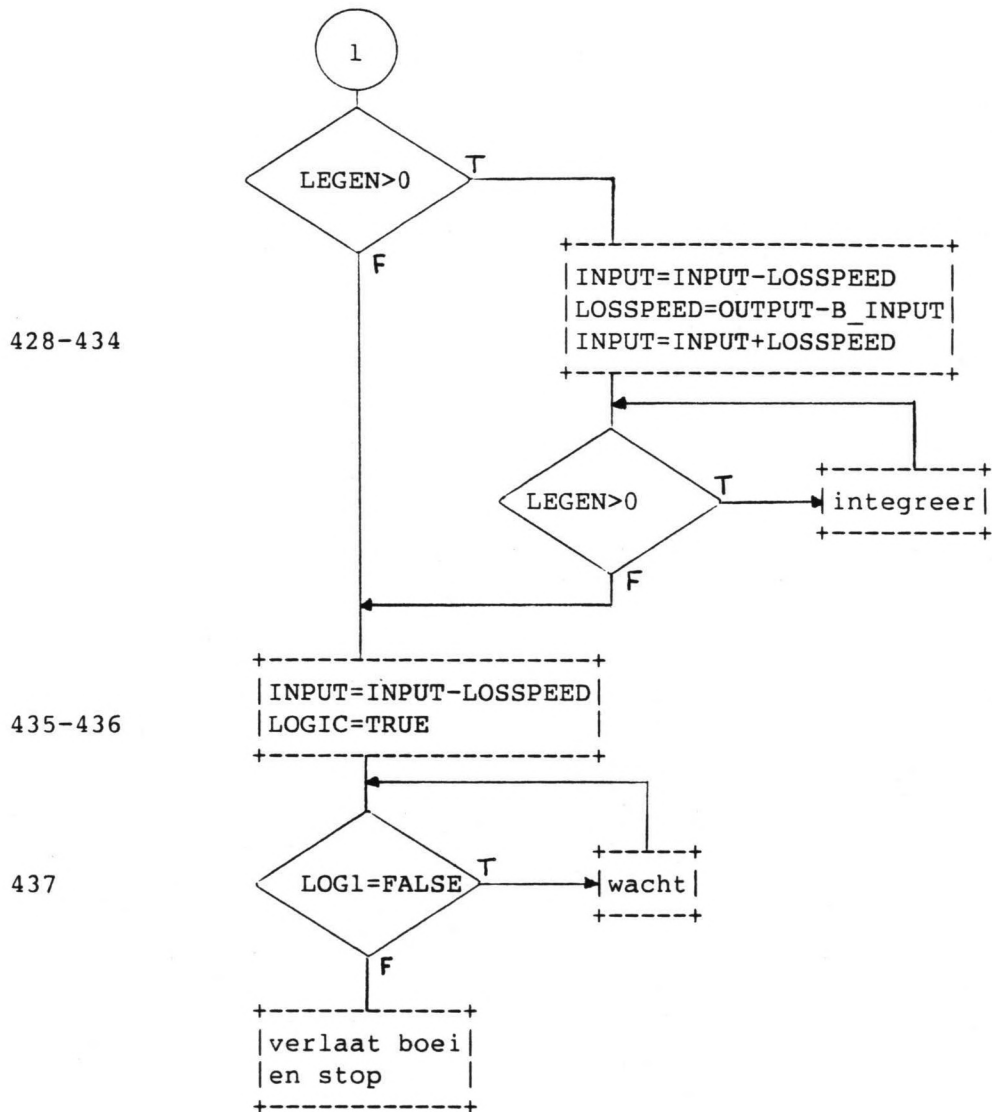
```
435 1 0 TANK_IN_HAND.INPUT=TANK_IN_HAND.INPUT-LOSSPEED;  
436 1 0 TANK_IN_HAND.LOGIC=TRUE;  
437 1 0 WAIT WHILE LOG1=FALSE;  
440 1 0 LEAVE BOEI_ROW;  
441 1 0 TERMINATE;
```

(435-441):

Na het lossen krijgt INPUT weer zijn oude waarde terug (435). Ook mag er eventueel weer een nieuw SHIP besteld worden (436). Als het weer goed is, dan mag SHIP de boei verlaten en daarmee is het proces afgerond (437-441).

1.4.2. Stroomschema.





1.5. Crude SHIP.

1.5.1. Procesbeschrijving (442-486).

```
442 1 0 C_ARRIVAL:
      WAIT WHILE LOG1=FALSE;
445 1 0 TALLY NOW-ARRIVALTIME IN WACHTTIJD(GENNR);
446 1 0 WAIT MOORINGTIME;
```

(442-446):

Dit gedeelte komt geheel overeen met de procesbeschrijving van het produkt SHIP (416-420).

```
450 1 0 CRUDETANK=OPSLAG_ROW.LAST;
451 1 0 CRUDETANK.C_OUTPUT=CRUDETANK.C_OUTPUT+LOSSPEED;
452 1 0 NUMSHIP=NUMSHIP+1;
453 1 0 IF VERSIE=1 THEN LOSSPEED=-LOSSPEED;
454 1 0 LEGEN=CONTENCE;
456 1 0 WHILE LEGEN>0 DO;
457 1 1 IF VERSIE=1 THEN INTEGRATE WHILE LEGEN>0;
461 1 1 ELSE DO;
462 1 2 LOG2=FALSE;
463 1 2 INTEGRATE WHILE (LEGEN>0 &
      CRUDETANK.C_LADING>0.01*CRUDETANK.C_CAPACITY);
465 1 2 IF CRUDETANK.C_LADING<=0.01*CRUDETANK.C_CAPACITY
      THEN DO;
466 1 3 HELP=NUMSHIP;
467 1 3 CRUDETANK.C_OUTPUT=4634.63;
468 1 3 LOG2=TRUE;
469 1 3 LOSSPEED=CRUDETANK.C_OUTPUT/HELP;
470 1 3 INTEGRATE WHILE (LEGEN>0 & NUMSHIP=HELP);
472 1 3 END;
473 1 2 END;
475 1 1 END;
```

(450-475):

De INPUT van de CRUDETANK wordt vermeerderd met de LOSSPEED en het aantal schepen aan de CRUDETANK wordt opgehoogd met 1 (450-452).

Voor VERSIE=1 moet de waarde van LOSSPEED weer positief worden, zodat dezelfde vergelijkingen als voor de overige versies gebruikt kunnen worden. LEGEN krijgt de waarde CONTENCE (453-454). Vervolgens lost SHIP zijn portie in de CRUDETANK (457).

Bij de andere versies wordt eerst de waarde LOG2=FALSE, dit is alleen om aan te geven dat de CRUDETANK nog goed gevuld is (462). Indien er inderdaad nog genoeg inzit, dan kan het crude SHIP volgepompt worden (463). Is dit echter niet zo (465), dan moeten de pompsnelheden, van de olietankers die gevuld worden, aangepast worden (466-469). Nu worden ze dus met een lagere snelheid volgepompt, zodat de OUTPUT uit de CRUDETANK gelijk is aan de INPUT ervan. Bij verandering van het aantal schepen begint elk crude SHIP weer bij statement 456.

CRUDETANK = Referentie naar een C_TANK.
HELP = Hulp variabele voor NUMSHIP.
LOG2 = Wel of geen lege CRUDETANK
TRUE=leeg FALSE=vol

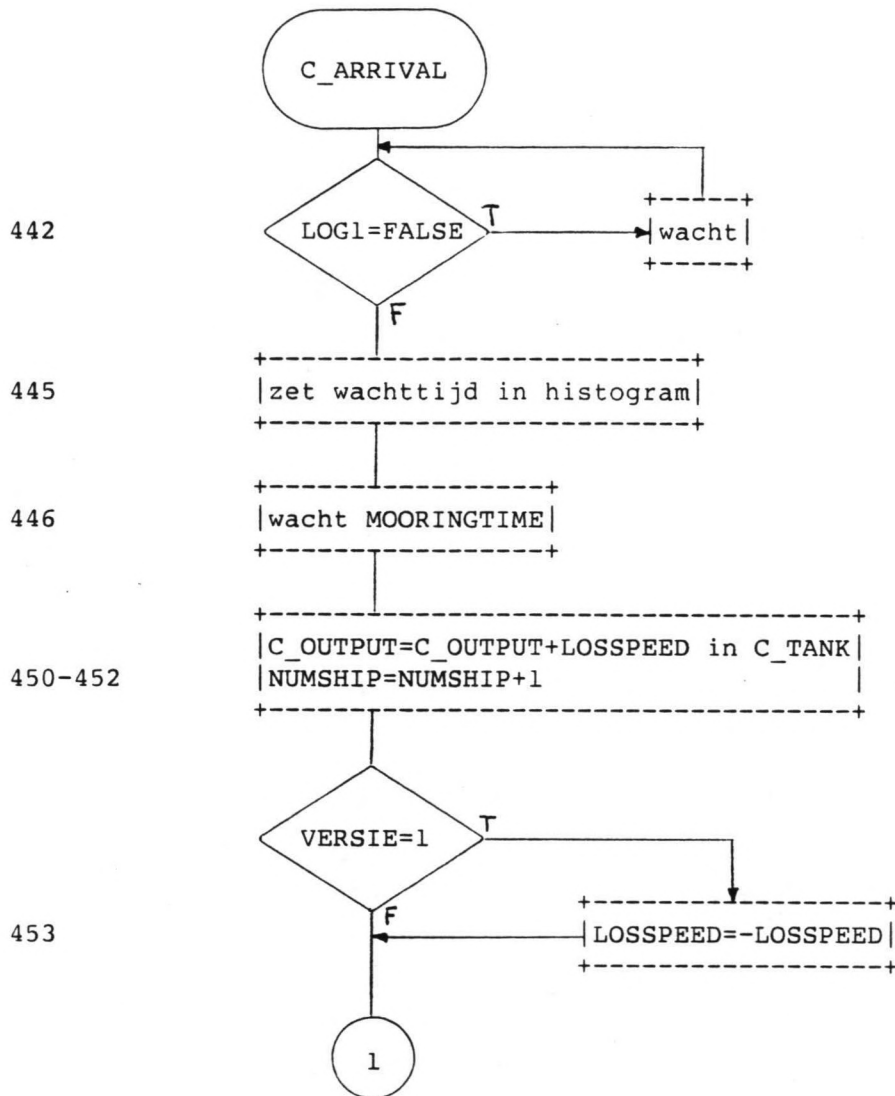
```
476 1 0 IF VERSIE=1 THEN LOSSPEED=-LOSSPEED;
477 1 0 ELSE DO;
478 1 1 IF (CRUDETANK.AMAX<4700 & LOG2=TRUE) THEN
    CRUDETANK.C_OUTPUT=NUMSHIP*LOSSPEED+165.37;
479 1 1 END;
480 1 0 CRUDETANK.C_OUTPUT=CRUDETANK.C_OUTPUT-LOSSPEED;
481 1 0 NUMSHIP=NUMSHIP-1;
482 1 0 WAIT WHILE LOG1=FALSE;
485 1 0 LEAVE BOEI_ROW;
486 1 0 TERMINATE;
```

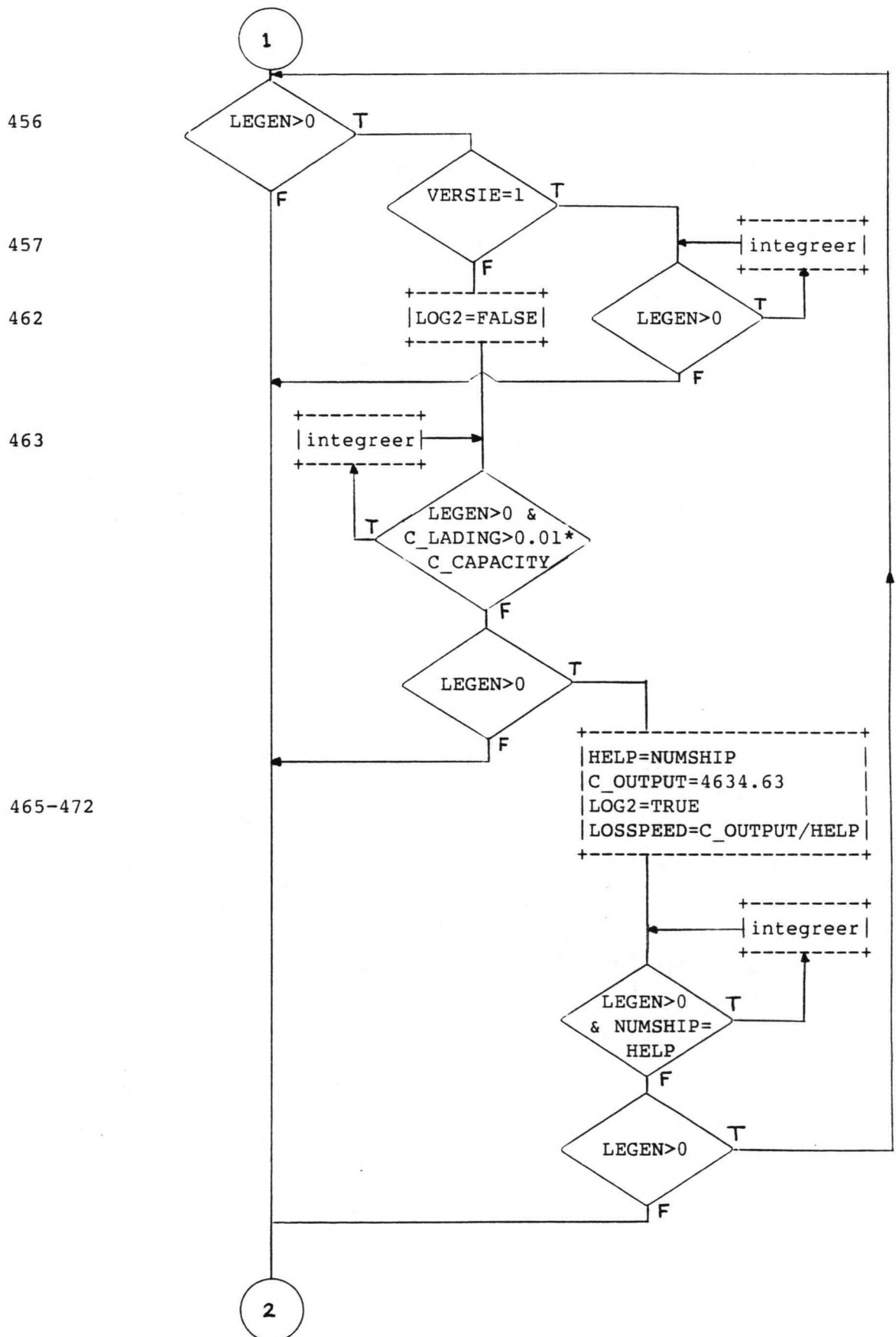
(476-486):

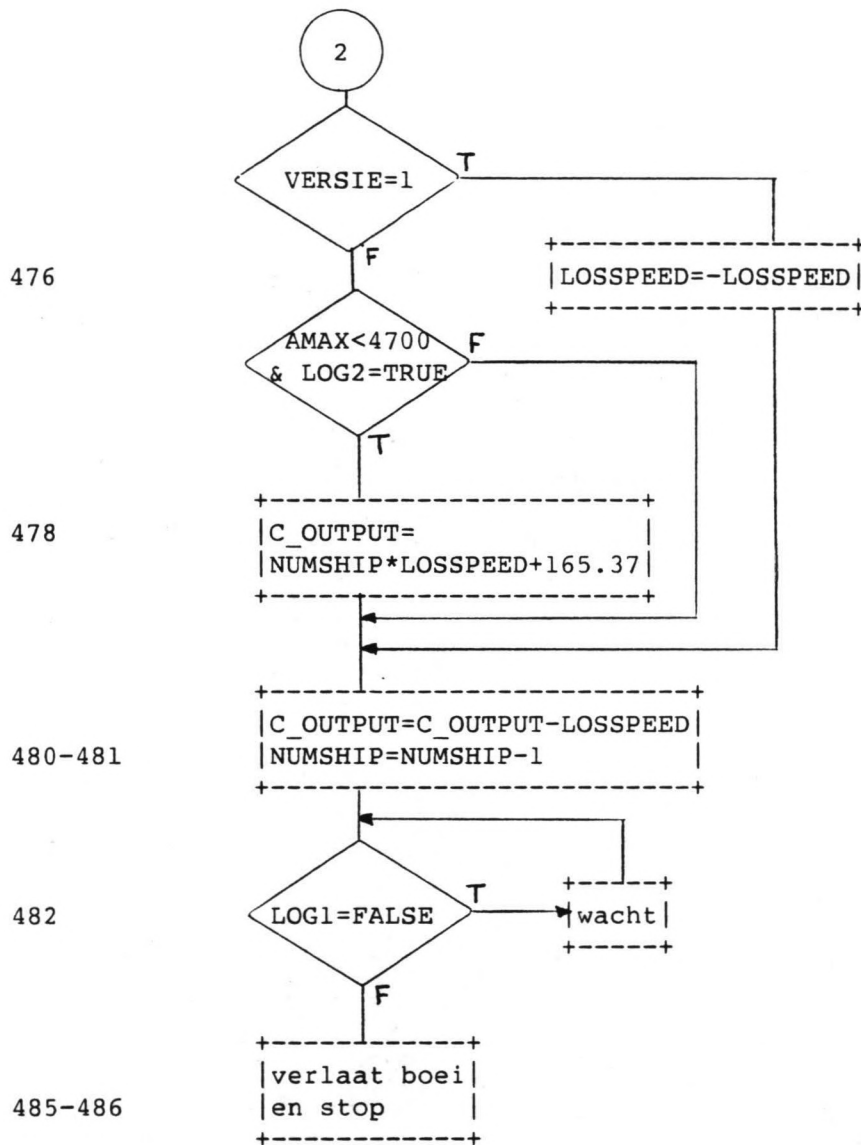
Als de VERSIE=1, dan moet LOSSPEED weer negatief gemaakt worden (476). Anders moet eerst eventueel de waarde van OUTPUT aangepast worden (478). AMAX<4700 betekent dat de raffinaderij in werking is, dus dan kan er minder crude in de schepen gepompt worden bij een bijna lege CRUDETANK. Daarna wordt deze OUTPUT weer verminderd met de LOSSPEED van SHIP (480). Ook het aantal schepen wordt 1 minder.

Bij goed weer kan SHIP de boei verlaten en is het proces ten einde.

1.5.2. Stroonschema.







1.6. Produkt TANK.

1.6.1. Procesbeschrijving (488-502).

```
488 1 0 STOCKING:
489 1 0 Y1=INPUT;
490 1 0 Y2=OUTPUT;
490 1 0 INTEGRATE WHILE (OUTPUT=Y2 & LADING>8*24*OUTPUT &
490 1 0 INPUT=Y1);
492 1 0 IF LADING<8*24*OUTPUT THEN DO:
493 1 1 IF NR=1 & LOGIC=TRUE THEN ACTIVATE GENERATOR1 FROM
493 1 1 STARTGEN1;
494 1 1 IF NR=2 & LOGIC=TRUE THEN ACTIVATE GENERATOR2 FROM
494 1 1 STARTGEN2;
495 1 1 IF NR=3 & LOGIC=TRUE THEN ACTIVATE GENERATOR3 FROM
495 1 1 STARTGEN3;
496 1 1 LOGIC=FALSE;
497 1 1 END;
498 1 0 IF LADING<8*24*OUTPUT THEN INTEGRATE WHILE
498 1 0 (OUTPUT=Y2 & INPUT=Y1);
502 1 0 REPEAT FROM STOCKING;
```

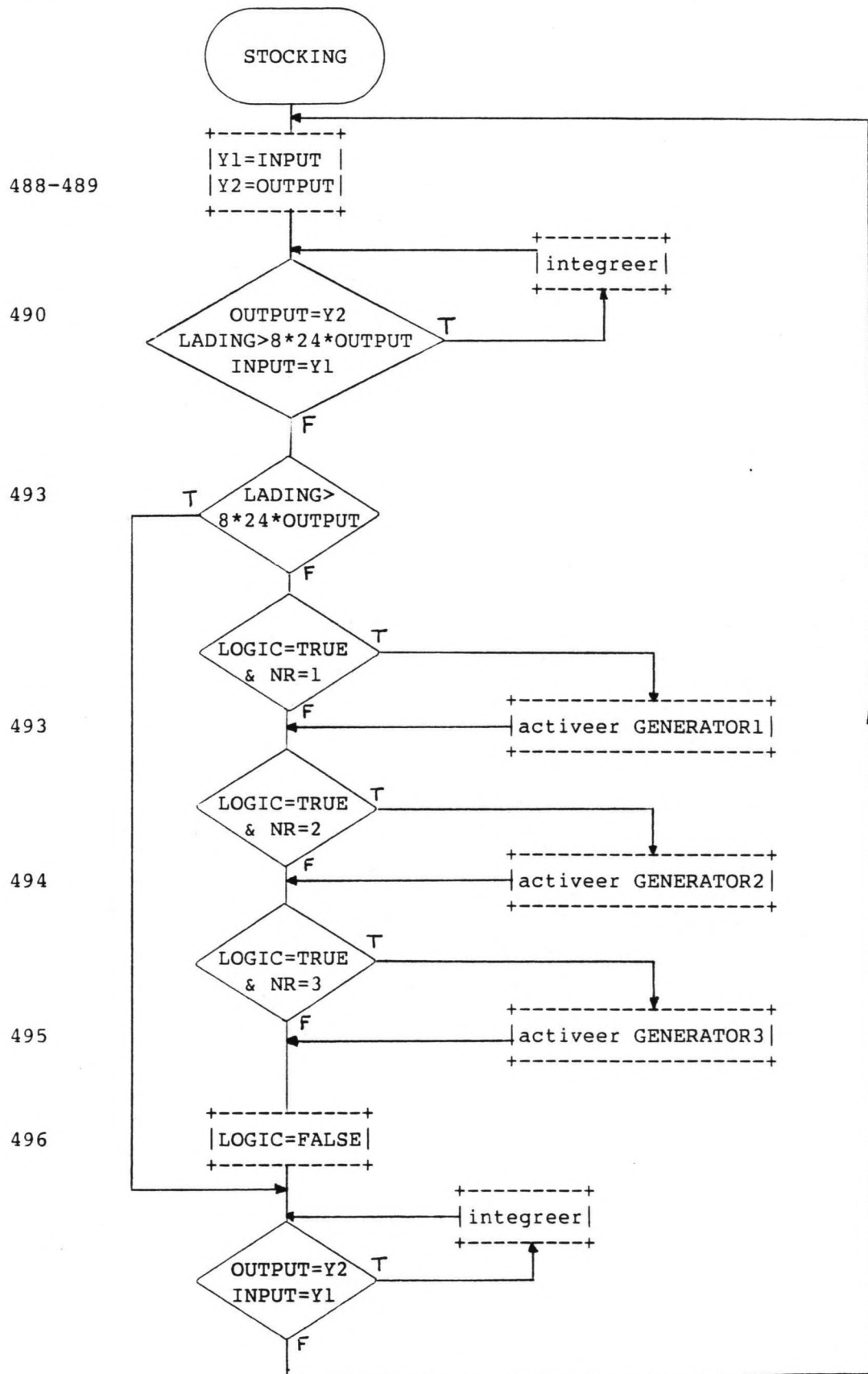
(488-502):

Dit proces zorgt er voor dat de inhoud van de TANK bijgehouden wordt door de differentiaal vergelijking:
LADING'=INPUT-OUTPUT (490 en 498).

Als de TANK nog voor minder dan 8 dagen voorraad heeft (492), dan wordt er een nieuw produkt SHIP besteld (493-495). LOGIC=FALSE zorgt er voor dat er niet 2 schepen tegelijk besteld worden (496). Als de INPUT of de OUTPUT verandert, dan zorgen de vergelijkingen Y1=INPUT en Y2=OUTPUT ervoor, dat deze waarden gewijzigd worden (488-489).

Y1 en Y2 = Hulp variabelen, voor INPUT en OUTPUT

1.6.2. Stroomschema.



1.7. Crude C TANK.

1.7.1. Procesbeschrijving (504-528).

Dit proces houdt de inhoud van de crude C_TANK bij volgens de differentiaal vergelijking:

$$C_LADING' = A1 - C_OUTPUT + C * C1 * EXP(-C2 * C_LADING).$$

```

504 1 0 C_STOCKING:
505 1 0 Y3=C_OUTPUT;
506 1 0 A1=A2;
507 1 0 C=1;
508 1 1 IF VERSIE=1 THEN DO;
      INTEGRATE WHILE (C_OUTPUT=Y3 & C_LADING<(0.5*C_CAPACITY) &
      C_LADING>(0.1*C_CAPACITY));
510 1 1 IF C_LADING>(0.5*C_CAPACITY) THEN DO;
511 1 2 A1=1E3;
512 1 2 C=0;
513 1 2 INTEGRATE WHILE C_OUTPUT=Y3;
515 1 2 END;
516 1 1 IF C_LADING<(0.1*C_CAPACITY) THEN DO;
517 1 2 A1=4.8E3;
518 1 2 C=0;
519 1 2 INTEGRATE WHILE (C_OUTPUT=Y3 & C_LADING<(0.15*C_CAPACITY));
521 1 2 END;
522 1 1 END;

523 1 0 IF VERSIE=1 THEN DO;
524 1 1 C=0;
525 1 1 INTEGRATE WHILE Y3=C_OUTPUT;
527 1 1 END;
528 1 0 REPEAT FROM C_STOCKING;
```

(504-528):

Eerst krijgen een aantal hulpvariabelen en een coëfficiënt een waarde (504-506). Als VERSIE niet 1 is, dan wordt de inhoud onder bepaalde voorwaarden geïntegreerd (508). Bij een volle C_TANK (510) wordt de D.V.:

$$C_LADING' = 1000 - C_OUTPUT \quad (511-513).$$

Bij een bijna lege C_TANK (516) wordt de D.V.:

$$C_LADING' = 4800 - C_OUTPUT \quad (517-519).$$

De D.V. krijgt weer de oude waarden, als

$$C_LADING > 0.15 * C_CAPACITY.$$

Bij VERSIE=1 wordt de volgende D.V. geïntegreerd:

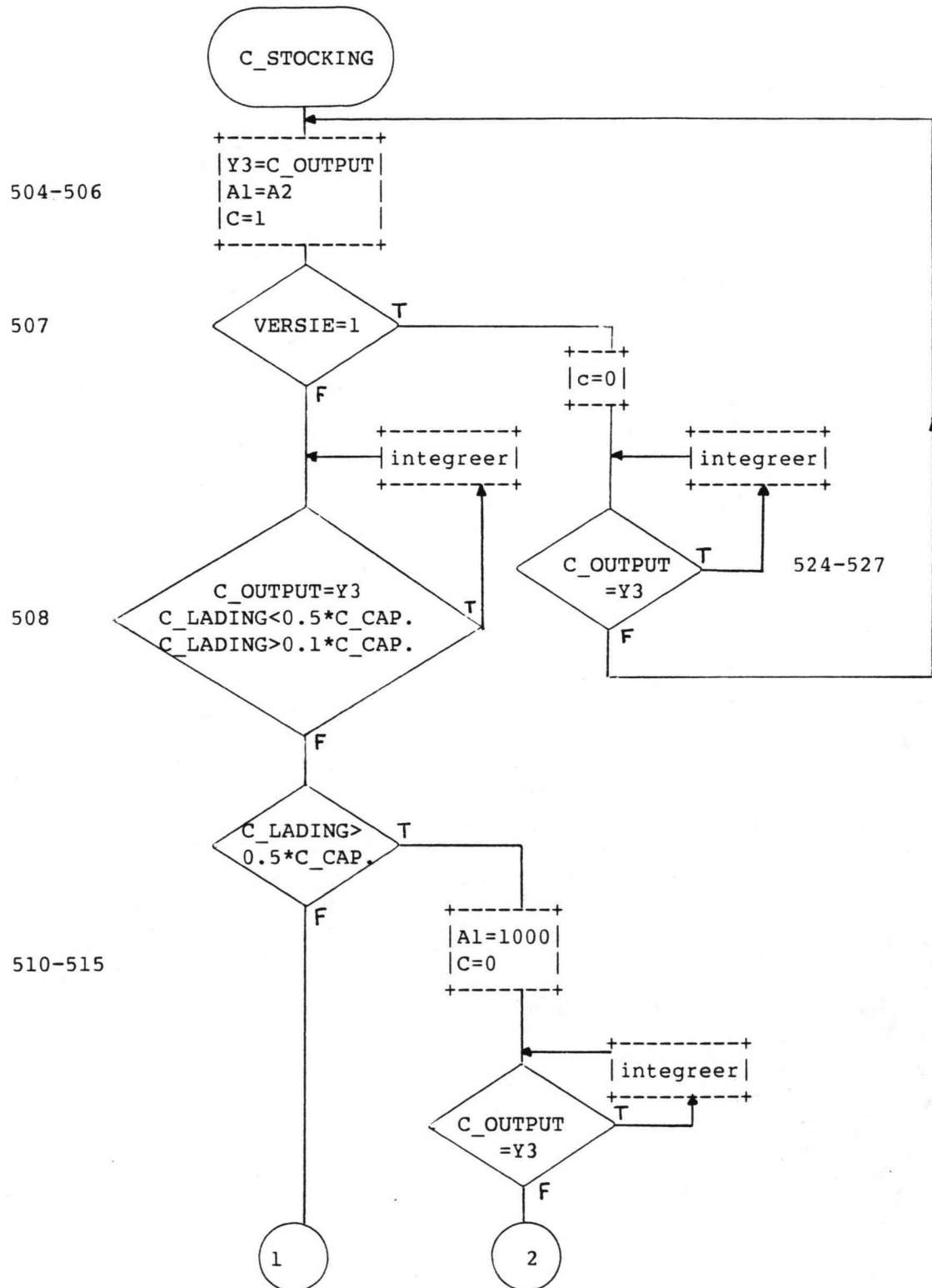
$$C_LADING = A1 - C_OUTPUT \quad (523-527).$$

Elke keer als de waarde van C_OUTPUT verandert, dan begint het proces weer van voren af aan (508,513,519,525 en 528).

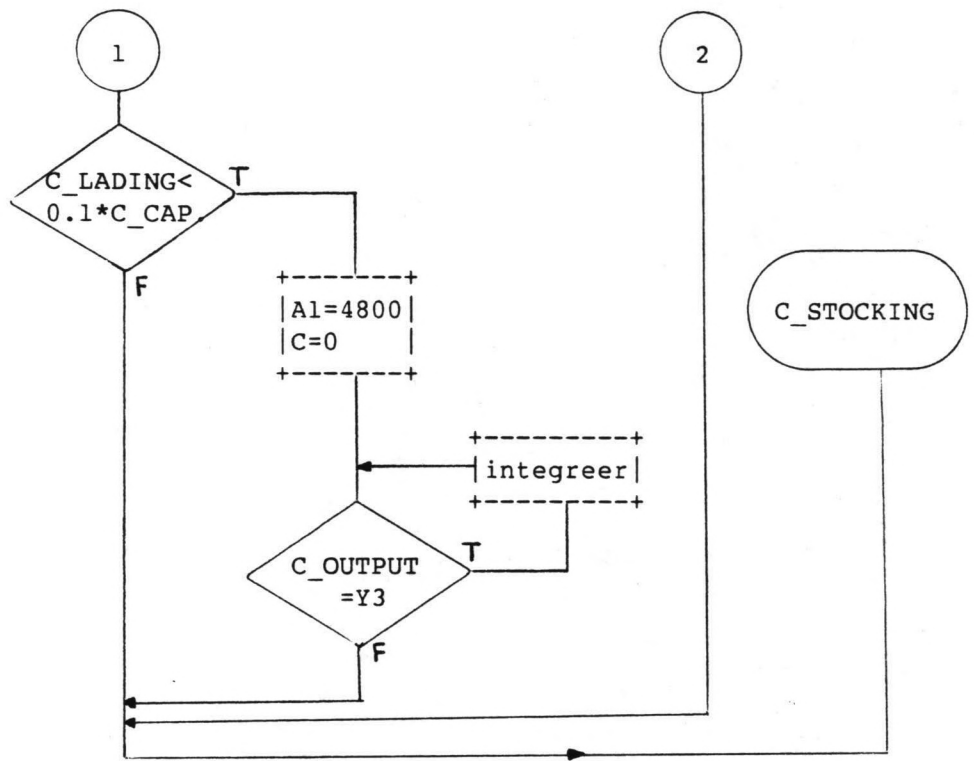
Y3 = Hulpvariabele, voor C_OUTPUT.

C = Coëfficiënt van D.V.

1.7.2. Stroonschema.



516-522



1.8. De VERZIEKER.

1.8.1. Procesbeschrijving (529-565).

De VERZIEKER bootst eigenlijk de raffinaderij na. 1 keer het proces rond is precies 1 jaar. Elk jaar is dus voor de raffinaderij hetzelfde ingedeeld.

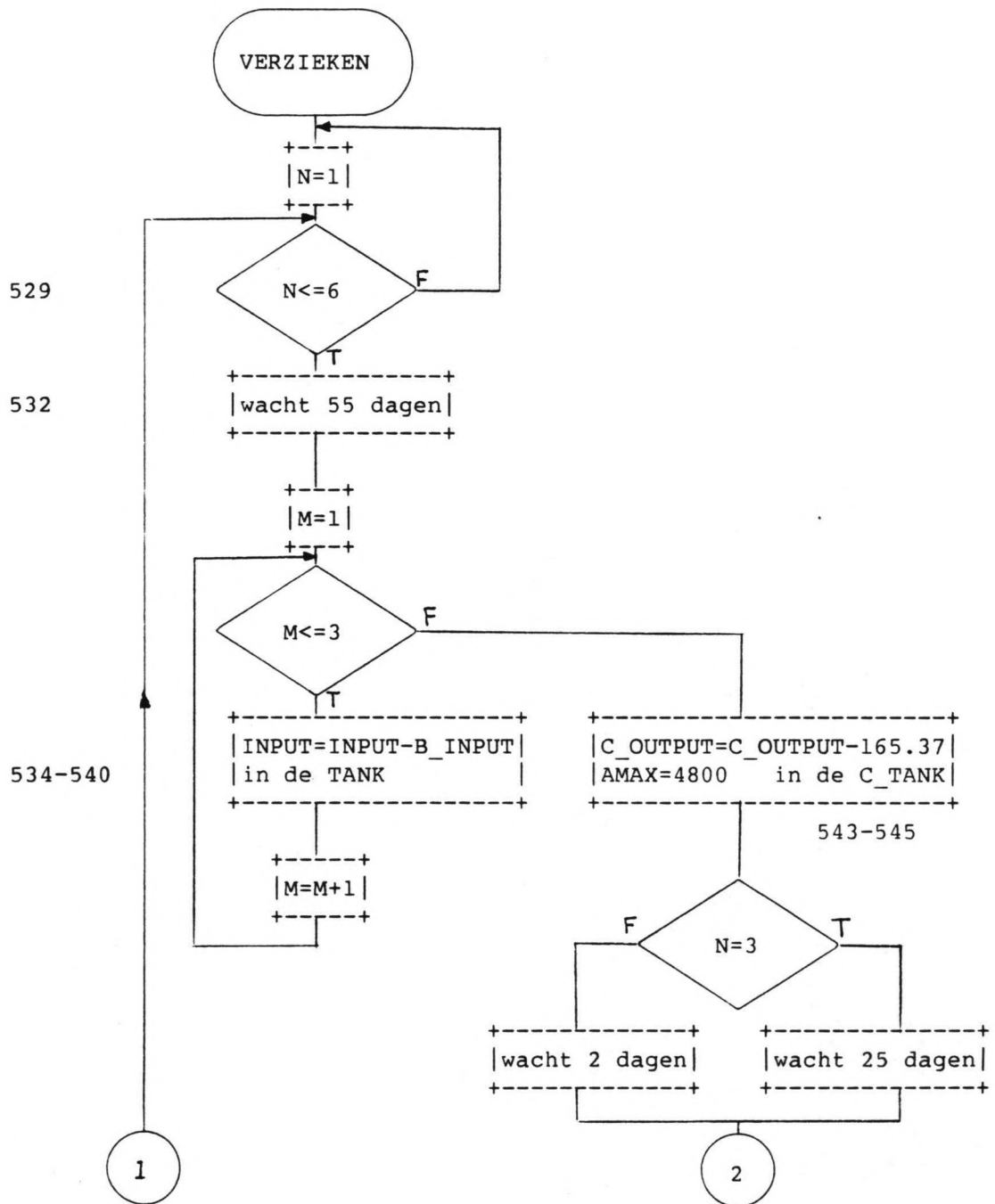
```
529 1 0 VERZIEKEN:
      FOR N=1 TO 6 DO;
532 1 1   WAIT 55 DAYS;
534 1 1   FOR M=1 TO 3 DO;
537 1 2     VERZ_TANK=FIRST TANK IN OPSLAG_ROW WITH NR=M;
538 1 2     VERZ_TANK.INPUT=VERZ_TANK.INPUT-VERZ_TANK.B_INPUT;
540 1 2   END;
543 1 1   VERZ_CTANK=OPSLAG_ROW.LAST;
544 1 1   VERZ_CTANK.C_OUTPUT=VERZ_CTANK.C_OUTPUT-165.37;
545 1 1   VERZ_CTANK.AMAX=4800;
546 1 1   IF N=3 THEN WAIT 25 DAYS; ELSE WAIT 2 DAYS;
554 1 1   FOR M=1 TO 3 DO;
557 1 2     VERZ_TANK=FIRST TANK IN OPSLAG_ROW WITH NR=M;
558 1 2     VERZ_TANK.INPUT=VERZ_TANK.INPUT+VERZ_TANK.B_INPUT;
560 1 2   END;
561 1 1   VERZ_CTANK.C_OUTPUT=VERZ_CTANK.C_OUTPUT+165.37;
562 1 1   VERZ_CTANK.AMAX=4634.63;
564 1 1   END;
565 1 0 REPEAT FROM VERZIEKEN;
```

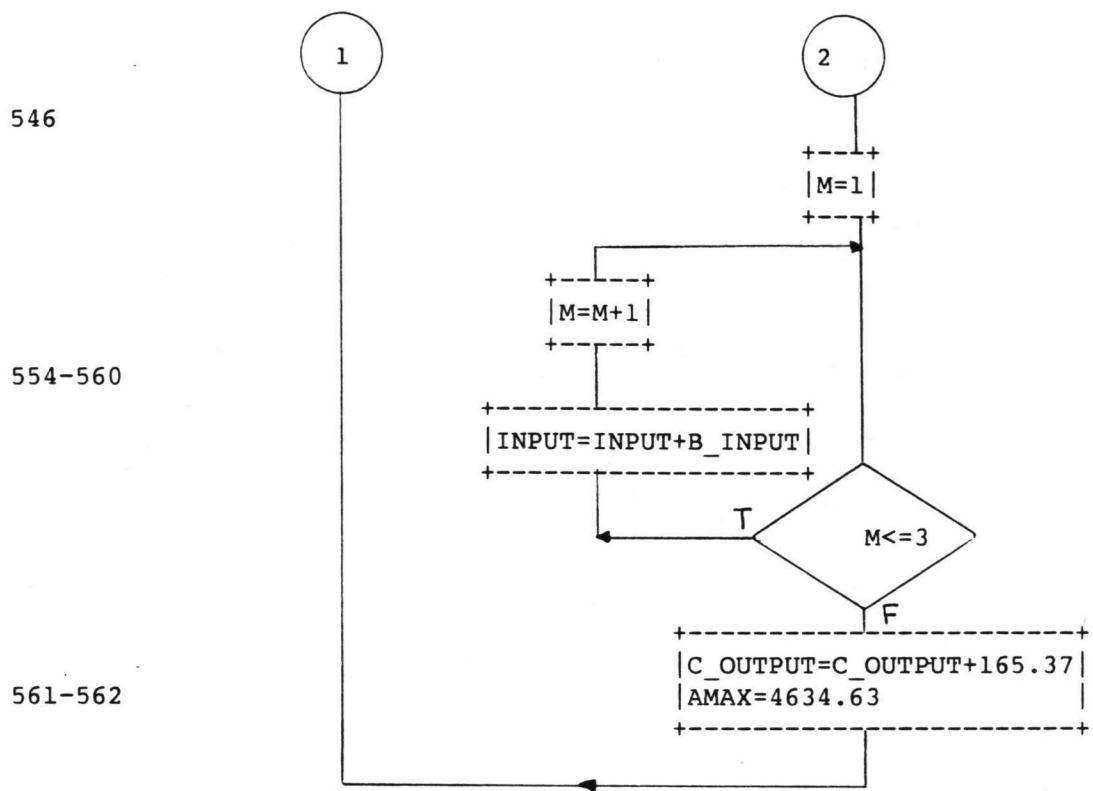
(529-565):

Er worden 6 perioden van 55 dagen gewerkt. Na elke periode van werken ligt de raffinaderij 2 dagen stil (storingen), behalve na de derde periode, dan ligt hij voor groot onderhoud 25 dagen stil (546).

Als de raffinaderij plat ligt, dan levert hij geen produkten en heeft hij geen crude nodig. De INPUT van de produkt TANK's (534-540 en 554-560) en de OUTPUT van de crude C_TANK (543-544 en 561) moeten aangepast worden. Ook de maximale output moet veranderd worden (545 en 562).

1.8.2. Stroomschema.





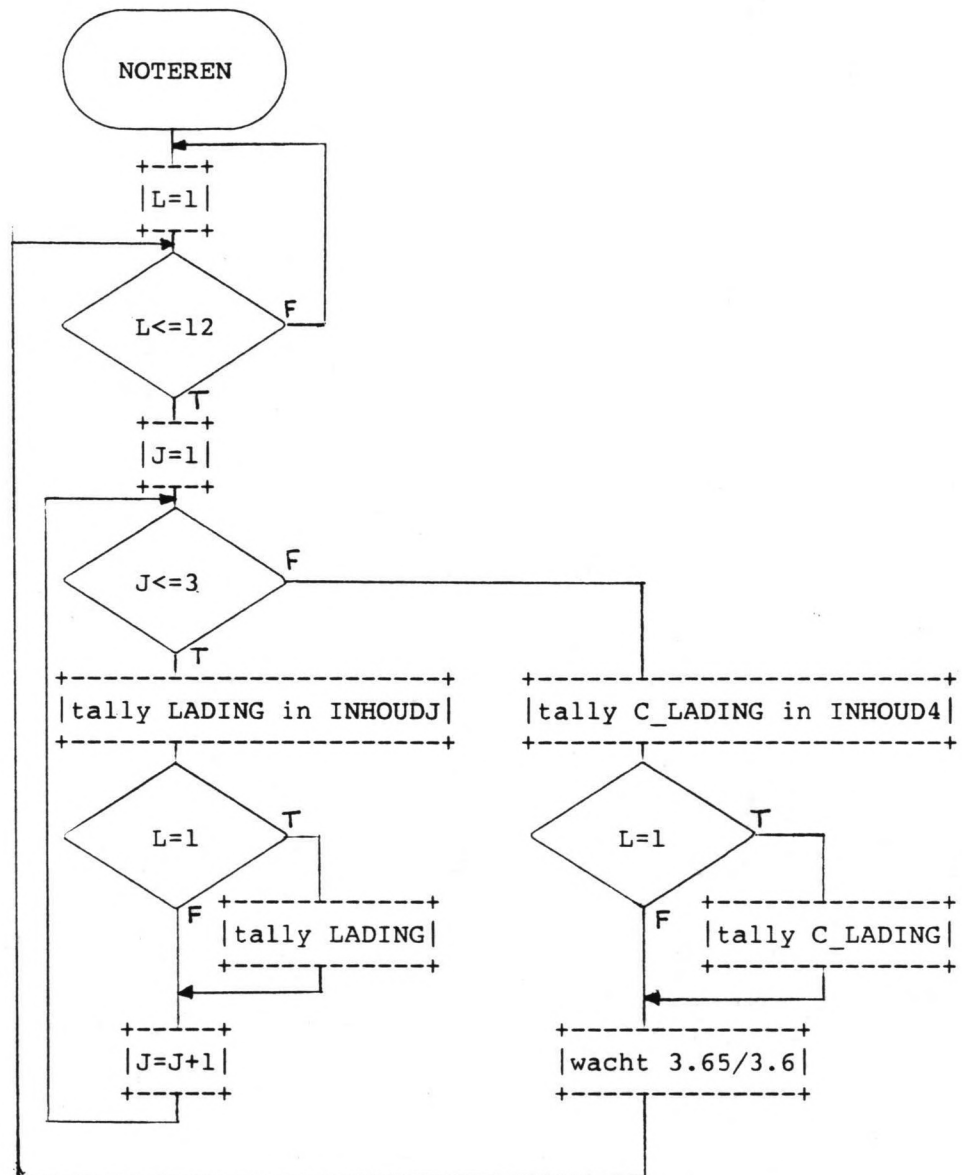
1.9. De NOTEERDER.

1.9.1. Procesbeschrijving (566-586).

```
566 1 0 NOTEREN:
      FOR L=1 TO 12 DO;
569 1 1     FOR J=1 TO 3 DO;
572 1 2         TANK_NOT=FIRST TANK IN OPSLAG_ROW WITH NR=J;
573 1 2         TALLY TANK_NOT.LADING IN INHOUD(J);
574 1 2         IF L=1 THEN TALLY TANK_NOT.LADING IN BERGING(J);
576 1 2     END;
579 1 1     CTANK_NOT=OPSLAG_ROW.LAST;
580 1 1     TALLY CTANK_NOT.C_LADING IN INHOUD(4);
581 1 1     IF L=1 THEN TALLY CTANK_NOT.C_LADING IN BERGING(4);
582 1 1     WAIT (3.650/3.6) DAYS;
585 1 1     END;
586 1 0 REPEAT FROM NOTEREN;
```

De NOTEERDER noteert de inhoud van de tanks in histogrammen (573,580), ongeveer 1 keer per dag, en in grafieken (574,581), ongeveer 1 keer in 12 dagen.

1.9.2. Stroomschema.



2. GEBRUIKERSHANDLEIDING.

Om het programma goed te kunnen gebruiken, moet eerst aan een aantal variabelen een waarde gegeven worden. Deze variabelen staan voor een gedeelte in de dataset en voor de rest in de procesbeschrijvingen van de componenten.

2.1. De dataset.

In de dataset kunnen achtereenvolgens de waarden van de volgende variabelen gewijzigd worden.

1. VERSIE. De versie die gesimuleerd wordt.
 - 1= 3 aanlegboeien, aanvoer ruwe olie met schepen, geen export van ruwe olie (huidige situatie).
 - 2= 3 aanlegboeien, aanvoer ruwe olie met pijpleiding geen export van ruwe olie.
 - 3= 3 aanlegboeien, aanvoer ruwe olie met pijpleiding wel export van ruwe olie.
 - 4= 4 aanlegboeien, aanvoer ruwe olie met pijpleiding wel export van ruwe olie.
2. NUMRUNS. Aantal simulatiejaren.

Er moet rekening mee worden gehouden dat de gegevens van het eerste simulatiejaar niet behouden blijven.
3. NUMBOEI. Aantal aanlegboeien.

Het aantal aanlegboeien moet in overeenstemming zijn met de waarde voor VERSIE.
4. X1 en X2. Gemiddelde en standaardafwijking van tussentijden voor slechtweer perioden van 2 a 3 dagen.
5. X3 en X4. Gemiddelde en standaardafwijking van tussentijden voor slechtweer perioden van 1 dag.
6. X5,X6 en X7. Dit zijn de waarden voor de gemiddelde tussenaankomst tijden voor de produkt schepen met gasoline, gas oil en fuel oil.

Deze zijn bij de nieuwe aannamen eigenlijk niet meer nodig (schepen worden besteld). Ze moeten echter wel een waarde krijgen, om verkeerd inlezen van de volgende waarden te voorkomen.
7. X8,X9 en X9. Gemiddelde tussenaankomsttijden voor ruwe olie export schepen, resp. 80.000 DWT, 135.000 DWT en 50.000 DWT.

De verdelingen zijn K-erlang. De waarde voor K moet in de procesbeschrijvingen van de generatoren veranderd worden.

8. Waarden die betrekking hebben op de produkttanks gasoline, gas oil en fuel oil. Per tank worden de volgende variabelen ingevoerd.
B_INPUT = Produktie van de raffinaderij (m3/h).
OUTPUT = Constante uitstroming uit de tank ter bevrediging van de behoeften van het eigen land (m3/h).
LADING = Begininhoud (m3).
CAPACITY = Tankgrootte (m3).
9. A1, C1, C2 en C3, zijn coëfficiënten voor de pompkarakteristiek $INPUT' = A1 + C1 \cdot \exp(-C2 \cdot INHOUD)$.
C3 is een extra coëfficiënt (moet een waarde hebben) voor een eventuele andere karakteristiek.
10. Tenslotte moeten de variabelen voor de C_TANK een waarde krijgen.
C_OUTPUT = Behoefte van de raffinaderij aan ruwe olie (m3/h).
C_VULLEN = Maximale laadsnelheid van een export schip (m3/h). (in uitvoer is deze helaas lossnelheid genoemd)
C_CAPACITY = Tankgrootte m3

Ter controle: Er zijn nu 29 waarden ingevoerd.

In de volgende paragrafen wordt aangeduid welke overige variabelen nog van waarde veranderd kunnen worden. Voorbeelden hiervan zijn te vinden in deel 3 van dit afstudeerverslag.

In de paragrafen 2.4 tot en met 2.7 worden de aannamen veranderd. Zo kan eventueel een andere terminal gesimuleerd worden. Deze moet wel enige overeenkomst vertonen met de hier gesimuleerde terminal in Tripoli.

2.2. Aantal aanlegboeien.

Het computerprogramma is in principe alleen geschikt voor simulatie met 3 of 4 aanlegboeien. Er kan echter een truc worden toegepast, waardoor simulatie met 1 of 2 aanlegboeien ook mogelijk is.

Indien er voor gezorgd wordt dat een aanlegboei continu bezet is, dan zal de HMASTER deze aanlegboei niet kunnen gebruiken. Dit is te realiseren door bijvoorbeeld een van de generatoren in een BOEI_ROW te zetten.

Aan het begin van de procesbeschrijving van de generator komt dan de volgende statement te staan:

```
IF BOEI_ROW(I) IS EMPTY THEN ENTER BOEI_ROW (I);
```

De if....then constructie is nodig om te voorkomen dat de generator, na 1 keer zijn proces doorlopen te hebben, bij activatie weer de BOEI_ROW(I) in gaat terwijl hij er al in zit. Voor I moet het nummer van de BOEI_ROW, die niet gebruikt wordt, worden ingevuld.

Een generator kan slechts in 1 rij tegelijk staan. Als er dus twee aanlegboeien niet gebruikt worden, dan moeten de procesbeschrijvingen van twee generatoren worden aangepast.

2.3. Tussenaankomsttijden.

De procesbeschrijvingen van de generatoren 5 t/m 7 beginnen met het maken van een tussenaankomsttijd. Dit is een trekking uit een K_Erlang verdeling, waarvan we de waarde van K kunnen variëren. Hoe groter de waarde van K is, des te kleiner is de spreiding. Voor K=1 vinden we de negatief exponentiele verdeling.

$$K_Erlang : f(t) = \frac{(K\mu)^K \cdot t^{K-1}}{(K-1)!} e^{-K\mu t}$$

Van de produkt schepen, generatoren 1 t/m 3, kan eventueel de tijdsduur tussen tijdstip van bestellen en tijdstip van aankomst bij de terminal worden veranderd. Deze tijdsduur is nu 5 tot 7 dagen.

Indien de tijdsduur wordt aangepast, dan is het aan te raden om ook het moment waarop besteld wordt hieraan aan te passen.

Voorbeelden van bovenstaande staan in deel 3 van het verslag.

2.4. Pompkarakteristiek.

De pomp waar het hier om gaat is de pomp van de ruwe olie pijpleiding van Irak naar Tripoli. De karakteristiek kan op drie manieren aangepast worden.

- a. Handhaving van de gemiddelde en de maximale pompcapaciteit. Alleen de vorm tussen de minimale waarde (1000 m³/h) en de maximale waarde (4800 m³/h) wordt veranderd. B.v. een lineair verband of een kwadratisch verband.
- b. Het verband wordt gehandhaaft, terwijl de gemiddelde en de maximale capaciteit anders worden. Dit houdt in dat er een andere pomp gebruikt wordt.
- c. Een combinatie van a. en b.

Bij manier b. hoeven alleen de coëfficiënten van de pompkarakteristiek veranderd te worden. Dit kan, zo blijkt uit par. 2.1., gebeuren door de waarden van de coëfficiënten in de dataset te wijzigen.

Voor de manieren b. en c. zal de differentiaal-vergelijking van de C_TANK uit het definitiegedeelte aangepast moeten worden.

diff. verg.:

$$C_LADING' = A1 - C_OUTPUT + C * C1 * \exp(-C2 * C_LADING)$$

Het rechter gedeelte van de vergelijking is gelijk aan C_INPUT - C_OUTPUT, dus de pompkarakteristiek is:

$$C_INPUT = A1 + C * C1 * \exp(-C2 * C_LADING)$$

De nieuwe pompkarakteristiek moet de volgende vorm hebben:

$$C_INPUT = A1 + C * (\text{term met coëfficiënten } C1, C2 \text{ en } C3)$$

De waarden van de coëfficiënten worden weer van de dataset afgelezen.

2.5. Andere schepen.

In de procebeschrijvingen van de generatoren worden de schepen gemaakt en krijgen de attributen ervan een waarde, waarmee de eigenschappen van het schip worden vastgelegd. De waarden van deze attributen kunnen gewijzigd worden. Zie voor de procesbeschrijvingen van de generatoren bijlage 1.

De waarde van PROD geeft aan welk produkt het schip vervoert. 1=gasoline, 2=gas oil, 3=fuel oil en 4=crude.

CONTENCE is de hoeveelheid die het schip aanvoerd. Bij de produkt schepen is de hoeveelheid afhankelijk van de waarde P (getal tussen 0 en 1). Algemeen geldt dat $CONTENCE = A + P * B$ dus de hoeveelheid ligt tussen A en A+B kl.

De crude schepen hebben een vaste hoeveelheid crude aan boord. Elk type crude schip heeft zijn eigen generator.

MOORINGTIME is de tijd die een schip nodig heeft om van de AANLEG_ROW naar de BOEI_ROW te varen en aan te leggen, zodat met laden/lossen begonnen kan worden. Deze tijd is afhankelijk van de grootte van het schip.

De waarde van LOSSPEED is afhankelijk van de pomp-capaciteit van de pijpleiding vanaf de boei naar de opslagtank. Dit is dus geen eigenschap van het schip zelf. Voor import crude schepen moet deze waarde altijd negatief zijn.

BERTH is een codering van het schip, die aangeeft aan welke aanlegboeien een schip verwerkt kan worden. Bij verandering van deze waarden of van de codering moet ook het hele proces van de HMASTER veranderd worden.

Zonder veel werk kunnen er meerdere typen crude export schepen gegenereerd worden. Dit komt neer op een extra procesbeschrijving in de vorm als die van de generatoren 5 t/m 7.

2.6. Slecht weer perioden.

Als men de slecht weer perioden en hun tussentijden wil veranderen, dan moet het gedeelte in MAIN wat hierover gaat veranderd worden. (Zie bijlage 1). De regels 140 en 141 zijn voor het gemiddelde en de standaardafwijking van de tussentijden en de regels 187 t/m 204 voor de duur van de slecht weer perioden (192 en 201). In deze laatste regels wordt ook aangegeven in welke tijd welke perioden voorkomen.

2.7. Raffinaderij.

In de procesbeschrijving van de VERZIEKER kan gespeeld worden met de perioden dat de raffinaderij wel en niet in werking is. Naar believen kunnen de regels 529,532 en 546 worden aangepast. Deze regels zorgen voor de jaarindeling. Eventueel kunnen de waarden ook variabel gehouden worden, wat wel iets meer aanpassing van het proces betekend.

2.8. Uitvoer.

De uitvoer van het programma bestaat uit drie delen.

1. Programma statements.
2. Cross reference tabel
3. Uitvoer resultaten.

Eerst wordt de tekst van het programma afgedrukt en dan een cross reference (bijlagen 1 en 2). Dit gebeurt standaard bij elk 'PROSIM' programma.

De lay out van de eigenlijke uitvoer (bijlage 3) wordt in de procesbeschrijving van MAIN vastgelegd. Hiervoor worden PL1 en PROSIM statements gebruikt.

PL1:

PUT [SKIP] LIST voor het afdrukken van tekst (tussen '...' b.v.130) of parameterwaarden (b.v.132). Een uitvoerpagina wordt in zes kolommen verdeeld. Na een komma in de uitdrukking tussen haakjes komt de volgende waarde of tekst in een volgende kolom te staan. Bij gebruik van SKIP wordt met de uitvoer op een volgende regel begonnen.

PUT [SKIP] EDIT voor het afdrukken van tekst. Ook hier wordt bij gebruik van SKIP weer op een nieuwe regel begonnen (b.v.134).

PROSIM:

STATUS

PRINT hist [HEADED BY title]

plot [HEADED BY title] [AX POSITON ap]

- "hist" is de naam van een histogram (238).
- "plot" is de naam van een grafiek (251).
- "title" is een zelf op te geven naam.
- "ap" geeft de positie van de verticale as van de grafiek aan.

Met behulp van deze opdrachten kunnen alle grafieken en histogrammen op overzichtelijke wijze worden uitgevoerd.

Bij PRINT STATUS print de computer de statussen van de componenten en statistische informatie over de wachtrijen.

STMT LEV NT

```

1      0      0      BEGIN;
10     1      0      DEFINE
GENERATOR1 AS A COMPONENT,
GENERATOR2 AS A COMPONENT,
GENERATOR3 AS A COMPONENT,
GENERATOR4 AS A COMPONENT,
GENERATOR5 AS A COMPONENT,
GENERATOR6 AS A COMPONENT,
GENERATOR7 AS A COMPONENT,
NOTEORDER AS A COMPONENT WITH ATTRIBUTES (TANK_NOT REFERENCE
TO TANK, CTANK_NOT REFERENCE TO C_TANK),
HMASTER AS A COMPONENT WITH ATTRIBUTE (SHIP_IN_HAND REFERENCE
TO SHIP),
VERZIEKER AS A COMPONENT WITH ATTRIBUTES (VERZ_TANK REFERENCE
TO TANK, VERZ_CTANK REFERENCE TO C_TANK);
SHIP AS A CLASS OF COMPONENTS EACH WITH ATTRIBUTES
((CONTENT, LOSSPEED, MOORINGTIME, LEAVINGTIME) REAL,
LEGEND CONTINUOUS (LEGEND=-LOSSPEED) INIT(1E4),
(BERTH, PROD, GENNR) INTEGER, TANK_IN_HAND REFERENCE TO TANK,
CRUDETANK REFERENCE TO C_TANK, LOG2 LOGICAL),
TANK AS A CLASS OF COMPONENTS EACH WITH ATTRIBUTES
(LADING CONTINUOUS (LADING=INPUT-OUTPUT) INIT(2E4),
(INPUT, OUTPUT, B_INPUT, CAPACITY, HULP) REAL, NR INTEGER,
LOGIC LOGICAL, (V1, V2) DOUBLE REAL),
C_TANK AS A CLASS OF COMPONENTS EACH WITH ATTRIBUTES
(C_LADING CONTINUOUS
(C_LADING'=(A1-C_OUTPUT)+C*(C1*EXP(-C2*C_LADING)))
INIT(1E5)),
(C_OUTPUT, C_VULLEN, A1, A2, C1, C2, C3, C_CAPACITY, AMAX, VAR)
REAL, C_INTEGER, V3 DOUBLE REAL),
NEW_C_TANK AS A REFERENCE TO C_TANK VARIABLE,
NEW_TANK AS A REFERENCE TO TANK VARIABLE,
NEXTSHIP AS A REFERENCE TO SHIP VARIABLE;
(BOEI_ROW(4), AANLEG_ROW, OPSLAG_ROW) AS QUEUES,
RAND(2) AS RANDOMSTREAMS WITH SEEDS (11123, 99755)
SAMPLED FROM DISTRIBUTION UNKNOWN,
UNIF1(8) AS RANDOMSTREAMS WITH SEEDS (377003613,
648473574, 1396717869, 2027350275, 1356162430, 711072531),
UNIF2(6) AS RANDOMSTREAMS WITH SEEDS (769795447,
1752629996, 645806097, 2013311468, 1393552473, 1966641861),
BERGING(4) AS PLOTS,
INVOER AS A PLOT,
INHOUD(4) AS HISTOGRAMS WITH PARAMETERS
L_BOUND(0) N_CLASSES(35) CLASSWIDTH(1E3),
WACHTTJD(7) AS HISTOGRAMS WITH PARAMETERS
L_BOUND(0) N_CLASSES(40) CLASSWIDTH(2),
W_TIMES(7) AS HISTOGRAMS WITH PARAMETERS
L_BOUND(0) N_CLASSES(30) CLASSWIDTH(0.5),
(AD, P, X1, X2, X3, X4, X5, X6, X7, X8, X9, X10) AS REAL VARIABLES,
(VERSIE, NUMBOEI, NUMRUNS, B, I, J, K, L, M, N, O, NUMSHIP, HELP)

```

Bijlage 1

STMT LEV NT

AS INTEGER VARIABLES,
LOG1 AS LOGICAL VARIABLES,
AN HOUR AS TIME_UNIT;

```

130 1 0 PUT LIST ('VERSIE', 'SIMULATIETIJD(JAREN)', 'AANTAL STEIGERS');
131 1 0 GET LIST (VERSIE, NUMRUNS, NUMBOEI);
132 1 0 PUT SKIP LIST (VERSIE, NUMRUNS, NUMBOEI);
133 1 0 PUT SKIP (2);
134 1 0 PUT SKIP EDIT ('PARAMETER WAARDEN VAN DE STOCHASTISCHE VERDELINGEN')
    (A);
135 1 0 GET LIST (X1, X2, X3, X4);
136 1 0 PUT SKIP LIST (X1, X2, X3, X4);
137 1 0 GET LIST (X5, X6, X7, X8, X9, X10);
138 1 0 PUT SKIP LIST (X5, X6, X7, X8, X9, X10);
139 1 0 PUT SKIP (2);
140 1 0 RESHAPE RAND(1) WITH PARAMETERS MEAN(X1) DEVIATION(X2);
141 1 0 RESHAPE RAND(2) WITH PARAMETERS MEAN(X3) DEVIATION(X4);
142 1 0 RESHAPE INHOUD(4) WITH PARAMETERS CLASSWIDTH(3E4)
    L_BOUND(0) N_CLASSES(35);
143 1 0 L_NUMSHIP=0;
144 1 0 IF VERSIE=1 THEN ACTIVATE GENERATOR4 FROM STARTGEN4;
145 1 0 IF VERSIE>2 THEN DO:
146 1 1 ACTIVATE GENERATOR5 FROM STARTGEN5;
147 1 1 ACTIVATE GENERATOR6 FROM STARTGEN6;
148 1 1 ACTIVATE GENERATOR7 FROM STARTGEN7;
149 1 1 END;
150 1 0 ACTIVATE HMASTER FROM STARTWORK;
151 1 0 ACTIVATE NOTEERDER FROM NOTEREN;
152 1 0 ACTIVATE VERZIEKER FROM VERZIEKEN;
153 1 0 PUT SKIP EDIT ('GEGEVENS PRODUKTANKS: GASOLINE, GAS OIL, FUEL OIL')
    (A);
154 1 0 PUT SKIP LIST ('INPUT (M3/H)', 'OUTPUT (M3/H)', 'BEGININHOUD (M3)',
    'TANKGROOTTE (M3)');
155 1 0 FOR I=1 TO 3 DO:
158 1 1 NEWTANK=NEW TANK;
159 1 1 NEWTANK.NR=I;
160 1 1 NEWTANK.LOGIC=TRUE;
161 1 1 GET LIST (NEWTANK.B_INPUT, NEWTANK.INPUT, NEWTANK.OUTPUT,
    NEWTANK.HULP, NEWTANK.CAPACITY);
162 1 1 PUT SKIP LIST (NEWTANK.INPUT, NEWTANK.OUTPUT,
    NEWTANK.HULP, NEWTANK.CAPACITY);
163 1 1 NEWTANK.LADING=NEWTANK.HULP;
165 1 1 PUT NEWTANK TO TAIL OF OPSLAG_ROW;
166 1 1 ACTIVATE NEWTANK FROM STOCKING;
168 1 1 END;

```

STMT LEV NT

```

169 1 0 NEWC_TANK=NEW C_TANK;
170 1 0 PUT SKIP (2);
171 1 0 PUT SKIP EDIT ('POMP KARAKTERISTIEK: INPUT:=A1+') (A);
172 1 0 PUT EDIT ('C1*EXP(-C2*INHOUD)') (A);
173 1 0 PUT SKIP LIST ('A1','C1','C2','C3');
174 1 0 GET LIST (NEWC_TANK.A1,NEWC_TANK.C1,NEWC_TANK.C2,NEWC_TANK.C3,
NEWC_TANK.C_OUTPUT,NEWC_TANK.C_VULLEN,NEWC_TANK.C_CAPACITY);
175 1 0 PUT SKIP LIST (NEWC_TANK.A1,NEWC_TANK.C1,NEWC_TANK.C2,
NEWC_TANK.C3);
176 1 0 PUT SKIP EDIT ('CRUDETANK WARDEN :') (A);
177 1 0 PUT SKIP LIST ('OUTPUT (M3/H)','LOSSNELHEID (M3/H)',
178 1 0 'TANGROORTE (M3)');
179 1 0 PUT SKIP LIST (NEWC_TANK.C_OUTPUT,NEWC_TANK.C_VULLEN,
NEWC_TANK.C_CAPACITY);
180 1 0 NEWC_TANK.A2=NEWC_TANK.A1;
181 1 0 NEWC_TANK.AMAX=4800-165.37;
182 1 0 PUT NEWC_TANK TO TAIL OF OPSLAG_ROW;
183 1 0 ACTIVATE_NEWC_TANK FROM C_STOCKING;
184 1 0 FOR K=1 TO NUMRUNS DO;
187 1 1 WHILE NOW<(150+(K-1)*365) DAYS DO;
188 1 1 LOG1=TRUE;
189 1 2 WAIT RAND(1) DAYS;
190 1 2 LOG1=FALSE;
191 1 2 WAIT (2+(UNIF1(8)<0.5)) DAYS;
192 1 2 END;
193 1 1 WHILE NOW<(210+(K-1)*365) DAYS DO;
194 1 1 LOG1=TRUE;
195 1 2 WAIT RAND(2) DAYS;
196 1 2 LOG1=FALSE;
197 1 2 WAIT 1 DAY;
198 1 2 END;
199 1 1 LOG1=TRUE;
200 1 1 WAIT 365*K DAYS-NOW;
201 1 1 IF K=1 THEN DO;
202 1 1 FOR O=1 TO 4 DO;
203 1 1 CLEAR INHOUD(O);
204 1 1 END;
205 1 1 LOG1=TRUE;
206 1 1 WAIT 365*K DAYS-NOW;
207 1 1 IF K=1 THEN DO;
208 1 1 FOR O=1 TO 4 DO;
209 1 1 CLEAR INHOUD(O);
210 1 1 END;
211 1 1 END;
212 1 3 END;
213 1 3 IF K>1 THEN DO;
214 1 3 IF VERSIE=1 THEN DO;
215 1 3 FOR O=1 TO 4 DO;
216 1 3 TALLY WACHTTIJD(O),MEAN IN W_TIMES(O);
217 1 3 CLEAR WACHTTIJD(O);
218 1 3 END;
219 1 3 END;ELSE DO;
220 1 3 FOR O=1 TO 7 DO;
221 1 3 IF O=4 THEN TALLY WACHTTIJD(O),MEAN IN W_TIMES(O);
222 1 3 CLEAR WACHTTIJD(O);
223 1 3 END;
224 1 3 END;
225 1 3 FOR O=1 TO 7 DO;
226 1 3 IF O=4 THEN TALLY WACHTTIJD(O),MEAN IN W_TIMES(O);
227 1 3 CLEAR WACHTTIJD(O);
228 1 3 END;
229 1 3 END;
230 1 4 END;
231 1 4 FOR O=1 TO 7 DO;
232 1 4 IF O=4 THEN TALLY WACHTTIJD(O),MEAN IN W_TIMES(O);
233 1 4 CLEAR WACHTTIJD(O);
234 1 4 END;

```

STMT LEV NT

```

235 1 2      END:
237 1 1      PRINT INHOUT(1) HEADED BY 'GASOLINE';
238 1 0      PRINT INHOUT(2) HEADED BY 'GAS OIL';
239 1 0      PRINT INHOUT(3) HEADED BY 'FUEL OIL';
240 1 0      PRINT INHOUT(4) HEADED BY 'CRUDE';
241 1 0      PRINT W_TIMES(1) HEADED BY 'WACHTTIJDEN GASOLINE';
242 1 0      PRINT W_TIMES(2) HEADED BY 'WACHTTIJDEN GAS OIL';
243 1 0      PRINT W_TIMES(3) HEADED BY 'WACHTTIJDEN FUEL';
244 1 0      IF VERSIE=1 THEN PRINT W_TIMES(4) HEADED BY
245           'WACHTTIJDEN IMPORT CRUDE';
246 1 0      ELSE DO:
247           PRINT W_TIMES(7) HEADED BY 'WACHTTIJDEN SMALL CRUDE';
248 1 1      PRINT W_TIMES(5) HEADED BY 'WACHTTIJDEN MIDDLE CRUDE';
249 1 1      PRINT W_TIMES(6) HEADED BY 'WACHTTIJDEN BIG CRUDE';
250 1 1      END:
251 1 0      PRINT BERGING(4) HEADED BY 'CRUDE' AX POSITION 0;
252 1 0      CANCEL ALL;
253 1 0      TERMINATE;

```

```

/*****
/* PROCESS OF THE GENERATORS */
*****/

```

```

254 1 0      STARTGEN1:
255           WAIT (5+2*UNIF2(1)) DAYS;
256 1 0      P=UNIF1(1);
257 1 0      NEXTSHIP=NEW SHIP;
258 1 0      NEXTSHIP.GENNR=1;
259 1 0      NEXTSHIP.PROD=1;
260 1 0      NEXTSHIP.CONTENTCE=13000+3500*P;
261 1 0      NEXTSHIP.LOSSPEED=347;
262 1 0      NEXTSHIP.MOORINGTIME=4.5+P;
263 1 0      NEXTSHIP.BERTH=2;
264 1 0      PUT NEXTSHIP TO TAIL OF AANLEG_ROW;
265 1 0      TERMINATE;

266 1 0      STARTGEN2:
267           WAIT (5+2*UNIF2(2)) DAYS;
268 1 0      P=UNIF1(2);
269 1 0      NEXTSHIP=NEW SHIP;
270 1 0      NEXTSHIP.GENNR=2;
271 1 0      NEXTSHIP.PROD=2;
272 1 0      NEXTSHIP.CONTENTCE=10400+3000*P;
273 1 0      NEXTSHIP.LOSSPEED=296;
274 1 0      NEXTSHIP.MOORINGTIME=4.5+P;
275 1 0      NEXTSHIP.BERTH=2;
276 1 0      PUT NEXTSHIP TO TAIL OF AANLEG_ROW;
277 1 0      TERMINATE;

```

STMT LEV NT

```

278 1 0 STARTGEN3:
280 1 0 WAIT (5+2*UNIF2(3)) DAYS;
281 1 0 P=UNIF1(3);
282 1 0 NEXTSHIP=NEW SHIP;
283 1 0 NEXTSHIP.GENNR=3;
284 1 0 NEXTSHIP.PROD=3;
285 1 0 NEXTSHIP.CONTENTENCE=15000+4700*P;
286 1 0 NEXTSHIP.LOSSPEED=1579;
287 1 0 NEXTSHIP.MOORINGTIME=4.8+1.4*P;
288 1 0 NEXTSHIP.BERTH=4;
289 1 0 PUT NEXTSHIP TO TAIL OF AANLEG_ROW;
    TERMINATE;

```

```

290 1 0 STARTGEN4:
292 1 0 WAIT (365/13.8345) DAYS;
293 1 0 P=UNIF1(4);
294 1 0 NEXTSHIP=NEW SHIP;
295 1 0 NEXTSHIP.GENNR=4;
296 1 0 NEXTSHIP.PROD=4;
297 1 0 NEXTSHIP.CONTENTENCE=94675;
298 1 0 NEXTSHIP.LOSSPEED=-3550;
299 1 0 NEXTSHIP.MOORINGTIME=5.9+1.2*P;
300 1 0 NEXTSHIP.BERTH=3;
301 1 0 PUT NEXTSHIP TO TAIL OF AANLEG_ROW;
    REPEAT FROM STARTGEN4;

```

```

302 1 0 STARTGEN5:
303 1 0 AD=0;
306 1 1 FOR B=1 TO 2 DO;
308 1 1 AD=AD + LOG(UNIF2(4));
309 1 0 END;
311 1 0 WAIT -(X8/2)*AD DAYS;
312 1 0 P=UNIF1(5);
313 1 0 NEXTSHIP=NEW SHIP;
314 1 0 NEXTSHIP.GENNR=5;
315 1 0 NEXTSHIP.PROD=4;
316 1 0 NEXTSHIP.CONTENTENCE=94675;
317 1 0 NEXTSHIP.LOSSPEED=4734;
318 1 0 NEXTSHIP.MOORINGTIME=5.8+1.2*P;
319 1 0 NEXTSHIP.BERTH=3;
320 1 0 PUT NEXTSHIP TO TAIL OF AANLEG_ROW;
    REPEAT FROM STARTGEN5;

```

STMT LEV NT

```

321 1 0 STARTGEN6:
      AD=0;
      FOR B=1 TO 2 DO;
322 1 0       AD=AD + LOG(UNIF2(5));
325 1 1       END;
327 1 1       WAIT -(X9/2)*AD DAYS;
328 1 0       P=UNIF1(6);
330 1 0       NEXTSHIP=NEW SHIP;
331 1 0       NEXTSHIP.GENNR=6;
332 1 0       NEXTSHIP.PROD=4;
333 1 0       NEXTSHIP.CONTENCE=135000/0.845;
334 1 0       NEXTSHIP.LOSSPEED=4734;
335 1 0       NEXTSHIP.MOORINGTIME=7.0+1.5*P;
336 1 0       NEXTSHIP.BERTH=1;
337 1 0       PUT NEXTSHIP TO TAIL OF AANLEG_ROW;
338 1 0       REPEAT FROM STARTGEN6;
339 1 0

```

```

340 1 0 STARTGEN7:
      AD=0;
      FOR B=1 TO 2 DO;
341 1 0       AD=AD + LOG(UNIF2(6));
344 1 1       END;
346 1 1       WAIT -(X10/2)*AD DAYS;
347 1 0       P=UNIF1(7);
349 1 0       NEXTSHIP=NEW SHIP;
350 1 0       NEXTSHIP.GENNR=7;
351 1 0       NEXTSHIP.PROD=4;
352 1 0       NEXTSHIP.CONTENCE=50000/0.845;
353 1 0       NEXTSHIP.LOSSPEED=4734;
354 1 0       NEXTSHIP.BERTH=3;
355 1 0       NEXTSHIP.MOORINGTIME=5.2+1.1*P;
356 1 0       PUT NEXTSHIP TO TAIL OF AANLEG_ROW;
357 1 0       REPEAT FROM STARTGEN7;
358 1 0

```

```

/*****
/* PROCESS OF THE HARBOURMASTER */
*****/

```

```

359 1 0 STARTWORK:
      WAIT WHILE AANLEG_ROW IS EMPTY;
361 1 0       IF VERSIE=4 THEN WAIT WHILE (BOEI_ROW(1) IS NOT EMPTY &
      BOEI_ROW(2) IS NOT EMPTY & BOEI_ROW(3) IS NOT EMPTY &
365 1 0       BOEI_ROW(4) IS NOT EMPTY);ELSE WAIT WHILE (BOEI_ROW(1) IS NOT
      EMPTY & BOEI_ROW(2) IS NOT EMPTY & BOEI_ROW(3) IS NOT EMPTY);
369 1 0       WAIT WHILE LOG1=FALSE;

```


STMT LEV NT

```

371 1 0 IF (BOEI_ROW(4) IS EMPTY & VERSIE=4) THEN DO:
372 1 1 SHIP_IN_HAND=FIRST SHIP IN AANLEG_ROW WITH BERTH=2;
373 1 1 IF SHIP_IN_HAND=NOCOMP THEN SHIP_IN_HAND=FIRST SHIP IN
    AANLEG_ROW WITH BERTH=4;
374 1 1 IF SHIP_IN_HAND=NOCOMP THEN DO:
375 1 2 REMOVE SHIP_IN_HAND FROM AANLEG_ROW;
376 1 2 PUT SHIP_IN_HAND TO TAIL OF BOEI_ROW(4);
377 1 2 IF SHIP_IN_HAND.PROD=4 THEN ACTIVATE SHIP_IN_HAND FROM
378 1 2 C_ARRIVAL;ELSE ACTIVATE SHIP_IN_HAND FROM ARRIVAL;
379 1 2 END:
380 1 1 IF BOEI_ROW(1) IS EMPTY THEN DO:
381 1 0 SHIP_IN_HAND=FIRST SHIP IN AANLEG_ROW WITH BERTH=1;
382 1 1 IF SHIP_IN_HAND=NOCOMP THEN SHIP_IN_HAND=FIRST SHIP IN
383 1 1 AANLEG_ROW WITH BERTH=3;
    IF SHIP_IN_HAND=NOCOMP THEN SHIP_IN_HAND=FIRST SHIP IN
384 1 1 AANLEG_ROW WITH BERTH=4;
    IF SHIP_IN_HAND=NOCOMP THEN DO:
385 1 1 REMOVE SHIP_IN_HAND FROM AANLEG_ROW;
386 1 2 PUT SHIP_IN_HAND TO TAIL OF BOEI_ROW(1);
387 1 2 IF SHIP_IN_HAND.PROD=4 THEN ACTIVATE SHIP_IN_HAND FROM
388 1 2 C_ARRIVAL;ELSE ACTIVATE SHIP_IN_HAND FROM ARRIVAL;
389 1 2 END:
390 1 2 IF BOEI_ROW(2) IS EMPTY THEN DO:
391 1 1 SHIP_IN_HAND=FIRST SHIP IN AANLEG_ROW WITH BERTH=2;
392 1 0 IF SHIP_IN_HAND=NOCOMP THEN SHIP_IN_HAND=FIRST
393 1 1 SHIP IN AANLEG_ROW WITH BERTH=1;
394 1 1 IF SHIP_IN_HAND=NOCOMP THEN DO:
    IF SHIP_IN_HAND=NOCOMP THEN DO:
395 1 1 REMOVE SHIP_IN_HAND FROM AANLEG_ROW;
396 1 2 PUT SHIP_IN_HAND TO TAIL OF BOEI_ROW(2);
397 1 2 IF SHIP_IN_HAND.PROD=4 THEN ACTIVATE SHIP_IN_HAND FROM
398 1 2 C_ARRIVAL;ELSE ACTIVATE SHIP_IN_HAND FROM ARRIVAL;
399 1 2 END:
400 1 2 IF BOEI_ROW(3) IS EMPTY THEN DO:
401 1 1 SHIP_IN_HAND=FIRST SHIP IN AANLEG_ROW WITH BERTH>2;
402 1 0 IF SHIP_IN_HAND=NOCOMP THEN DO:
403 1 1 REMOVE SHIP_IN_HAND FROM AANLEG_ROW;
404 1 1 PUT SHIP_IN_HAND TO TAIL OF BOEI_ROW(3);
405 1 2 IF SHIP_IN_HAND.PROD=4 THEN ACTIVATE SHIP_IN_HAND FROM
406 1 2 C_ARRIVAL;ELSE ACTIVATE SHIP_IN_HAND FROM ARRIVAL;
407 1 2 END:
408 1 2 IF SHIP_IN_HAND=NOCOMP THEN WAIT 1 HOUR:
409 1 1 REPEAT FROM STARTWORK;
410 1 1
411 1 0
415 1 0

```

STMT LEV NT

```

/*****
/* PROCESS OF A PRODUCT SHIP */
/*****

```

```

416 1 0 ARRIVAL:
419 1 0 WAIT WHILE LOG1=FALSE;
420 1 0 TALLY NOW-ARRIVALTIME IN WACHTTIJD(GENNR);
422 1 0 WAIT MOORINGTIME;
423 1 0 TANK_IN_HAND=FIRST TANK IN OPSLAG_ROW WITH NR=PROD;
424 1 0 TANK_IN_HAND.INPUT=TANK_IN_HAND.INPUT+LOSSPEED;
426 1 0 LEGEN=CONTENCE;
428 1 0 INTEGRATE WHILE (LEGEN>0 &
429 1 1 TANK_IN_HAND.LADING<TANK_IN_HAND.CAPACITY);
430 1 1 IF TANK_IN_HAND.LADING>TANK_IN_HAND.CAPACITY THEN DO;
431 1 1 TANK_IN_HAND.INPUT=TANK_IN_HAND.INPUT-LOSSPEED;
432 1 1 TANK_IN_HAND.INPUT=TANK_IN_HAND.INPUT+LOSSPEED;
433 1 1 INTEGRATE WHILE LEGEN>0;
434 1 1 END;
435 1 0 TANK_IN_HAND.INPUT=TANK_IN_HAND.INPUT-LOSSPEED;
436 1 0 TANK_IN_HAND.LOGIC=TRUE;
437 1 0 WAIT WHILE LOG1=FALSE;
440 1 0 LEAVE BOEI_ROW;
441 1 0 TERMINATE;

```

```

/* PROCES OF A CRUDE SHIP */

```

```

442 1 0 C_ARRIVAL:
445 1 0 WAIT WHILE LOG1=FALSE;
446 1 0 TALLY NOW-ARRIVALTIME IN WACHTTIJD(GENNR);
450 1 0 WAIT MOORINGTIME;
451 1 0 CRUDETANK=OPSLAG_ROW.LAST;
452 1 0 CRUDETANK.C_OUTPUT=CRUDETANK.C_OUTPUT+LOSSPEED;
453 1 0 NUMSHIP=NUMSHIP+1;
454 1 0 IF VERSIE=1 THEN LOSSPEED=-LOSSPEED;
455 1 0 LEGEN=CONTENCE;
456 1 0 WHILE LEGEN>0 DO;
457 1 1 IF VERSIE=1 THEN INTEGRATE WHILE LEGEN>0;
461 1 1 ELSE DO;
462 1 2 LOG2=FALSE;
463 1 2 INTEGRATE WHILE (LEGEN>0 &
465 1 2 CRUDETANK.C_LADING>0.01*CRUDETANK.C_CAPACITY);
466 1 3 IF CRUDETANK.C_LADING<=0.01*CRUDETANK.C_CAPACITY
467 1 3 THEN DO;
468 1 3 HELP=NUMSHIP;
469 1 3 CRUDETANK.C_OUTPUT=4634.63;
470 1 3 LOG2=TRUE;
471 1 3 LOSSPEED=CRUDETANK.C_OUTPUT/HELP;
472 1 3 INTEGRATE WHILE (LEGEN>0 & NUMSHIP=HELP);
473 1 2 END;

```


STMT LEV NT

```

522 1 1 END:
523 1 0 IF VERSIE=1 THEN DO:
524 1 1 C=0:
525 1 1 INTEGRATE WHILE V3=C_OUTPUT:
527 1 1 END:
528 1 0 REPEAT FROM C_STOCKING;

```

/* PROCESS OF THE VERZIEKER */

```

529 1 0 VERZIEKEN:
FOR N=1 TO 6 DO:
532 1 1 WAIT 55 DAYS:
534 1 1 FOR M=1 TO 3 DO:
537 1 2 VERZ_TANK=FIRST TANK IN OPSLAG_ROW WITH NR=M:
538 1 2 VERZ_TANK.INPUT=VERZ_TANK.INPUT-VERZ_TANK.B_INPUT:
540 1 2 END:
543 1 1 VERZ_CTANK=OPSLAG_ROW.LAST:
544 1 1 VERZ_CTANK.C_OUTPUT=VERZ_CTANK.C_OUTPUT-165.37:
545 1 1 VERZ_CTANK.AMAX=4800:
546 1 1 IF N=3 THEN WAIT 25 DAYS: ELSE WAIT 2 DAYS:
554 1 1 FOR M=1 TO 3 DO:
557 1 2 VERZ_TANK=FIRST TANK IN OPSLAG_ROW WITH NR=M:
558 1 2 VERZ_TANK.INPUT=VERZ_TANK.INPUT+VERZ_TANK.B_INPUT:
560 1 2 END:
561 1 1 VERZ_CTANK.C_OUTPUT=VERZ_CTANK.C_OUTPUT+165.37:
562 1 1 VERZ_CTANK.AMAX=4634.63:
564 1 1 END:
565 1 0 REPEAT FROM VERZIEKEN:

```

/* PROCESS OF THE NOTERORDER */

```

566 1 0 NOTEREN:
FOR L=1 TO 12 DO:
569 1 1 FOR J=1 TO 3 DO:
572 1 2 TANK_NOT=FIRST TANK IN OPSLAG_ROW WITH NR=J:
573 1 2 TALLY_TANK_NOT.LADING IN INHOUD(J):
574 1 2 IF L=1 THEN TALLY_TANK_NOT.LADING IN BERGING(J):
576 1 2 END:
579 1 1 CTANK_NOT=OPSLAG_ROW.LAST:
580 1 1 TALLY_CTANK_NOT.C_LADING IN INHOUD(4):
581 1 1 IF L=1 THEN TALLY_CTANK_NOT.C_LADING IN BERGING(4):
582 1 1 WAIT (3.650/3.6) DAYS:
585 1 1 END:
586 1 0 REPEAT FROM NOTEREN:
682 1 0 END:

```

END OF SOURCE TEXT

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
12	AANLEG_ROW	QUEUE 91,264,276,288,300,319,338,357,372,373,382,383,384,393,394,403,588
12	AD	REAL VARIABLE 102,302,306,306,309,321,325,325,328,340,344,344,347 PL/I - BUILTIN 22,25,29,33,36,42,55,68,83,86,89,90,91,92,93,94,95,96,97,98,99,100,101,102, 103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121, 122,123,124,125,126,140,141,142
*****	ADDR	REAL VARIABLE, ATTRIBUTE OF C_TANK 181,478,545,562
11	AMAX	REAL VARIABLE, ATTRIBUTE OF C_TANK 181,478,545,562
416	ARRIVAL	/* STATEMENT LABEL CONSTANT */ 378,389,399,408
11	A1	REAL VARIABLE, ATTRIBUTE OF C_TANK 174,175,180,505,511,517 71
11	A2	REAL VARIABLE, ATTRIBUTE OF C_TANK 180,505
12	B	INTEGER VARIABLE 117,303,304,304,305,322,323,323,324,341,342,342,343
11	B_INPUT	REAL VARIABLE, ATTRIBUTE OF TANK 161,430,538,558
12	BERGING	(4) PLOT 97,251,574,581
11	BERTH	INTEGER VARIABLE, ATTRIBUTE OF SHIP 263,275,287,299,318,337,355,600,603,606,609,612,615,618,621
12	BOEI_ROW	(4) QUEUE 90,371,376,381,387,392,397,402,406,591,591,591,591,594,594,594
11	C	INTEGER VARIABLE, ATTRIBUTE OF C_TANK 506,512,518,524 71
442	C_ARRIVAL	/* STATEMENT LABEL CONSTANT */ 377,388,398,407
11	C_CAPACITY	REAL VARIABLE, ATTRIBUTE OF C_TANK

Bylage 2

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
11	C_LADING	174, 179, 465, 510, 516, 645, 660, 666 CONTINUOUS ATTRIBUTE, ATTRIBUTE OF C_TANK 465, 510, 516, 580, 581 71, 78, 645, 660, 660, 666
11	C_OUTPUT	REAL VARIABLE, ATTRIBUTE OF C_TANK 174, 179, 451, 451, 467, 469, 478, 480, 480, 504, 544, 544, 561, 561 71, 660, 663, 666, 669
503	C_STOCKING	/* STATEMENT LABEL CONSTANT */ 183, 528
11	C_TANK	CLASS 1, 68, 169, 504, 504, 505, 505, 506, 510, 510, 511, 512, 516, 516, 517, 518, 524 71, 71, 71, 71, 71, 71, 660, 660, 660, 660, 660, 660, 663, 663, 666, 666, 666, 669, 669 REAL VARIABLE, ATTRIBUTE OF C_TANK 174, 179
11	C_VULLEN	REAL VARIABLE, ATTRIBUTE OF TANK 161, 162, 428, 630
11	CAPACITY	REAL VARIABLE, ATTRIBUTE OF SHIP 260, 272, 284, 296, 315, 334, 353, 424, 454
11	CONTENCE	REFERENCE TO COMPONENT, ATTRIBUTE OF SHIP 450, 451, 451, 465, 465, 467, 469, 478, 478, 480, 480, 645, 645
10	CTANK_NOT	REFERENCE TO COMPONENT, ATTRIBUTE OF NOTEORDER 25, 579, 580, 581
11	C1	REAL VARIABLE, ATTRIBUTE OF C_TANK 174, 175 71
11	C2	REAL VARIABLE, ATTRIBUTE OF C_TANK 174, 175 71
11	C3	REAL VARIABLE, ATTRIBUTE OF C_TANK 174, 175
*****	EXP	PL/I - BUILTIN 71
*****	FALSE	PROSIM - BUILTIN 191, 200, 462, 496, 597, 624, 636, 639, 651

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
10	GENERATOR1	COMPONENT 13,493
10	GENERATOR2	COMPONENT 14,494
10	GENERATOR3	COMPONENT 15,495
10	GENERATOR4	COMPONENT 16,144
10	GENERATORS	COMPONENT 17,146
10	GENERATOR6	COMPONENT 18,147
10	GENERATOR7	COMPONENT 19,148
11	GENNR	INTEGER VARIABLE, ATTRIBUTE OF SHIP 258,270,282,294,313,332,351,419,445
12	HELP	INTEGER VARIABLE 126,466,469,648
10	HMASTER	COMPONENT 30,150
11	HULP	REAL VARIABLE, ATTRIBUTE OF TANK 161,162,163
12	I	INTEGER VARIABLE 118,155,156,157,159
12	INHOUUD	(4) HISTOGRAM 99,142,212,238,239,240,241,573,580
11	INPUT	REAL VARIABLE, ATTRIBUTE OF TANK 161,162,423,423,429,429,431,431,435,435,488,538,538,558,558 58,654,657
12	INVOER	PLOT 98
12	J	INTEGER VARIABLE 119,569,570,570,571,573,574,678

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
12	K	INTEGER VARIABLE 120,184,185,185,186,187,196,206,208,216
12	L	INTEGER VARIABLE 121,566,567,567,568,574,581
11	LADING	CONTINUOUS ATTRIBUTE, ATTRIBUTE OF TANK 164,428,492,498,573,574 65,630,654
11	LEAVINGTIME	REAL VARIABLE, ATTRIBUTE OF SHIP
11	LEGEN	CONTINUOUS ATTRIBUTE, ATTRIBUTE OF SHIP 425,455,456 52,630,633,642,645,648
*****	LOG	PL/I - BUILTIN 306,325,344
11	LOGIC	LOGICAL VARIABLE, ATTRIBUTE OF TANK 160,436,493,494,495,496
12	LOG1	LOGICAL VARIABLE 188,191,197,200,205,597,624,636,639,651
11	LOG2	LOGICAL VARIABLE, ATTRIBUTE OF SHIP 462,468,478
11	LOSSPEED	REAL VARIABLE, ATTRIBUTE OF SHIP 261,273,285,297,316,335,354,423,429,430,431,435,451,453,453,469,476,476,478, 480 45
12	M	INTEGER VARIABLE 122,534,535,535,536,554,555,555,556,672,675
11	MOORINGTIME	REAL VARIABLE, ATTRIBUTE OF SHIP 262,274,286,298,317,336,356,420,446
12	N	INTEGER VARIABLE 123,529,530,530,531,546
11	NEWC_TANK	REFERENCE TO COMPONENT 83,169,174,174,174,174,174,174,174,175,175,175,175,175,179,179,180,180,181, 182,183
11	NEWTANK	REFERENCE TO COMPONENT 86,158,159,160,161,161,161,161,161,162,162,162,162,163,164,165,166

DCL NO. IDENTIFIER

ATTRIBUTES AND REFERENCES

11 NEXTSHIP

REFERENCE TO COMPONENT
 89,257,258,259,260,261,262,263,264,269,270,271,272,273,274,275,276,281,282,
 283,284,285,286,287,288,293,294,295,296,297,298,299,300,312,313,314,315,316,
 317,318,319,331,332,333,334,335,336,337,338,350,351,352,353,354,355,356,357

10 NOTEORDER

COMPONENT
 26,151

566 NOTEREN

/* STATEMENT LABEL CONSTANT */
 151,586

11 NR

INTEGER VARIABLE, ATTRIBUTE OF TANK
 159,493,494,495,627,672,675,678

***** NULL

PL/I - BUILTIN
 6,11,11,11

12 NUMBOEI

INTEGER VARIABLE
 115,131,132

12 NUMRUNS

INTEGER VARIABLE
 116,131,132,186

12 NUMSHIP

INTEGER VARIABLE
 125,143,452,452,466,478,481,481,648

12 O

INTEGER VARIABLE
 124,209,210,210,211,212,218,219,219,220,221,221,222,227,228,228,229,230,230,
 230,231

12 OPSLAG_ROW

QUEUE
 92,165,182,422,448,537,541,557,572,577

11 OUTPUT

REAL VARIABLE, ATTRIBUTE OF TANK
 161,162,430,489,492,498
 58,654,654,657

12 P

REAL VARIABLE
 103,256,260,262,268,272,274,280,284,286,292,298,311,317,330,336,349,356

11 PROD

INTEGER VARIABLE, ATTRIBUTE OF SHIP
 259,271,283,295,314,333,352,377,388,398,407,627

12 RAND

(2) DISTRIBUTION(UNKNOWN)
 93,94,140,141,189,198

11 SHIP

CLASS
 1,42,257,269,281,293,312,331,350,419,420,422,423,423,423,424,425,428,428,
 429,429,429,430,430,430,431,431,431,435,435,435,436,445,446,450,451,451,451,

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
10	SHIP_IN_HAND	453,453,454,455,456,462,465,465,467,468,469,469,476,476,478,478,478,480,480,480 45,600,603,606,609,612,615,618,621,627,630,630,630,633,642,645,645,648 REFERENCE TO COMPONENT, ATTRIBUTE OF HMASTER 29,372,373,373,374,375,376,377,378,382,383,383,384,384,385,386,387,388, 388,389,393,394,394,395,396,397,398,398,399,403,404,405,406,407,407,408,411 /* STATEMENT LABEL CONSTANT */ 493
254	STARTGEN1	/* STATEMENT LABEL CONSTANT */ 494
266	STARTGEN2	/* STATEMENT LABEL CONSTANT */ 495
278	STARTGEN3	/* STATEMENT LABEL CONSTANT */ 144,301
290	STARTGEN4	/* STATEMENT LABEL CONSTANT */ 146,320
302	STARTGEN5	/* STATEMENT LABEL CONSTANT */ 147,339
321	STARTGEN6	/* STATEMENT LABEL CONSTANT */ 148,358
340	STARTGEN7	/* STATEMENT LABEL CONSTANT */ 150,415
359	STARTWORK	/* STATEMENT LABEL CONSTANT */ 166,502
487	STOCKING	EXTERNAL FILE 131,135,137,161,174
*****	SYSIN	EXTERNAL FILE PRINT 130,132,133,134,136,138,139,153,154,162,170,171,172,173,175,176,177,178,179
*****	SYSPRINT	CLASS 1,55,158,488,488,489,489,492,492,493,493,494,494,495,495,496,498,498 58,58,627,654,654,654,654,654,657,657,657,672,675,678
11	TANK	REFERENCE TO COMPONENT, ATTRIBUTE OF SHIP 422,423,423,428,428,429,429,430,430,431,431,435,435,436,630,630
11	TANK_IN_HAND	REFERENCE TO COMPONENT, ATTRIBUTE OF NOTEORDER
10	TANK_NOT	

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
*****	TRUE	22,572,573,574
12	UNIF1	PROSIM - BULLTIN 160,188,197,205,436,468,478,493,494,495
12	UNIF2	(8) RANDOMSTREAM 95,192,256,268,280,292,311,330,349
11	VAR	(6) RANDOMSTREAM 96,254,266,278,306,325,344
12	VERSIE	REAL VARIABLE, ATTRIBUTE OF C_TANK
10	VERZ_CTANK	INTEGER VARIABLE 114,131,132,144,145,217,245,361,371,453,457,476,507,523
10	VERZ_TANK	REFERENCE TO COMPONENT, ATTRIBUTE OF VERZIEKER 36,543,544,544,545,561,561,562
529	VERZIEKEN	REFERENCE TO COMPONENT, ATTRIBUTE OF VERZIEKER 33,537,538,538,538,557,558,558,558
10	VERZIEKER	/* STATEMENT LABEL CONSTANT */ 152,565
12	W_TIMES	COMPONENT 37,152
12	WACHTTIJD	(7) HISTOGRAM 101,221,230,242,243,244,245,247,248,249
12	X1	(7) HISTOGRAM 100,221,222,230,231,419,445
12	X10	REAL VARIABLE 104,135,136,140
12	X2	REAL VARIABLE 113,137,138,347
12	X3	REAL VARIABLE 105,135,136,140
12	X4	REAL VARIABLE 106,135,136,141
		REAL VARIABLE 107,135,136,141

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
12	X5	REAL VARIABLE 108,137,138
12	X6	REAL VARIABLE 109,137,138
12	X7	REAL VARIABLE 110,137,138
12	X8	REAL VARIABLE 111,137,138,309
12	X9	REAL VARIABLE 112,137,138,328
11	V1	REAL VARIABLE, ATTRIBUTE OF TANK 488,654,657
11	V2	REAL VARIABLE, ATTRIBUTE OF TANK 489,654,657
11	V3	REAL VARIABLE, ATTRIBUTE OF C_TANK 504,660,663,666,669

Bijlage 3.

VERSIE	4	SIMULATIE TIJD(JAREN)	6	AANTAL STEIGERS	4
PARAMETER WAARDEN VAN DE STOCHASTISCHE VERDELINGEN					
6.16999E+00	2.94999E+00	2.10000E+01	1.03999E+01	9.12500E+00	1.01399E+01
1.59279E+01	2.89569E+01	2.31119E+01	2.28125E+01		
GEGEVENS PRODUKTANKS: GASOLINE, GAS OIL, FUEL OIL					
INPUT (M3/H)		OUTPUT (M3/H)		BEGIN INHOUD (M3)	
3.22269E+01	6.77209E+01	2.00000E+04	3.00000E+04	2.00000E+04	3.00000E+04
3.30409E+01	4.69959E+01	2.00000E+04	3.00000E+04	2.00000E+04	3.00000E+04
6.14119E+01	8.68019E+01	2.00000E+04	3.00000E+04	2.00000E+04	3.00000E+04
POMP KARAKTERISTIEK: INPUT=A1+C1*EXP(-C2*INHOUD)					
A1	C1	C2	C3		
9.10500E+02	9.99200E+03	1.88699E-05	0.00000E+00		
CRUDE TANK WAARDEN :					
OUTPUT (M3/H)		LOSSNHELHEID (M3/H)		TANKGROOTTE (M3)	
1.65369E+02	4.73400E+03	5.00000E+05			

Bijlage 3

HOOFDPROGRAMMA

VERSIE 3
SIMULATIETIJD(JAREN) 10
AANTAL STEIGERS 3

PARAMETER WAARDEN VAN DE STOCHASTISCHE VERDELINGEN
6.16999E+00 2.94999E+00 2.10000E+01 1.03999E+01
1.59279E+01 2.89569E+01 2.31119E+01 2.28125E+00 9.12500E+00 1.01399E+01

GEGEVENS PRODUKTANKS: GASOLINE,GAS OIL,FUEL OIL
INPUT (M3/H) OUTPUT (M3/H) BEGININHOUD (M3) TANKGROOTTE (M3)
3.22269E+01 6.77209E+01 2.00000E+04 3.00000E+04
3.30409E+01 4.69959E+01 2.00000E+04 3.00000E+04
6.14119E+01 8.68019E+01 2.00000E+04 3.00000E+04

POMP KARAKTERISTIEK: INPUT=A1+C1*EXP(-C2*INHOUD)
A1 C1 C2 C3
9.10500E+02 9.99200E+03 1.88699E-05 0.00000E+00

CRUDETANK WAARDEN :
OUTPUT (M3/H) LOSSNELHEID (M3/H) TANKGROOTTE (M3)
1.65369E+02 4.73400E+03 5.00000E+05

EXTREME WAARDEN VAN DE CRUDETANKINHOUD
2.51540E+05
2.58624E+05
2.63675E+05
2.68865E+05
2.55405E+05
2.20268E+05
2.53591E+05
2.15469E+05
2.21855E+05

ENTRIES	3240
ZERO ENTRIES	0
MEAN	13411.2
STD.DEVIATION	4221.524
MINIMUM	-3381.806
MAXIMUM	22658.13

FREQ	ENTRS	PERC	CUM PERC
<	0.	0.09	0.09
<	1000.	0.00	0.09
<	2000.	0.15	0.25
<	3000.	0.22	0.46
<	4000.	0.43	0.90
<	5000.	0.80	1.70
<	6000.	1.23	2.93
<	7000.	2.13	5.06
<	8000.	4.72	9.78
<	9000.	7.35	17.13
<	10000.	7.50	24.63
<	11000.	7.41	32.04
<	12000.	7.56	39.60
<	13000.	241	47.04
<	14000.	246	54.63
<	15000.	237	61.94
<	16000.	234	69.17
<	17000.	232	76.33
<	18000.	217	83.02
<	19000.	207	89.41
<	20000.	172	94.72
<	21000.	117	98.33
<	22000.	45	99.72
<	23000.	9	100.00

ENTRIES	3240
ZERO ENTRIES	0
MEAN	11810.59
STD. DEVIATION	3602.74
MINIMUM	-2226.51
MAXIMUM	19464.82

[illegible]

ENTRIES	3240
ZERO ENTRIES	0
MEAN	20293.3
STD. DEVIATION	5492.64
MINIMUM	1444.154
MAXIMUM	30013.9

FREQ	CLASSES	ENTRS	PERC	CUM
<	0.	0	0.00	0.00
<	1000.	0	0.00	0.00
<	2000.	2	0.06	0.06
<	3000.	2	0.06	0.12
<	4000.	5	0.15	0.28
<	5000.	5	0.15	0.43
<	6000.	15	0.46	0.90
<	7000.	10	0.31	1.20
<	8000.	12	0.37	1.57
<	9000.	14	0.43	2.01
<	10000.	20	0.62	2.62
<	11000.	36	1.11	3.73
<	12000.	35	1.08	4.81
<	13000.	86	2.65	7.47
<	14000.	179	5.52	12.99
<	15000.	192	5.93	18.92
<	16000.	192	5.93	24.85
<	17000.	197	6.08	30.93
<	18000.	187	5.77	36.70
<	19000.	193	5.96	42.65
<	20000.	190	5.86	48.52
<	21000.	182	5.62	54.14
<	22000.	189	5.83	59.97
<	23000.	161	4.97	64.94
<	24000.	188	5.80	70.74
<	25000.	182	5.62	76.36
<	26000.	162	5.00	81.36
<	27000.	178	5.49	86.85
<	28000.	148	4.57	91.42
<	29000.	131	4.04	95.46
<	30000.	83	2.56	98.02
<	31000.	64	1.98	100.00

ENTRIES	9
ZERO ENTRIES	0
MEAN	10.2679
STD. DEVIATION	3.38451
MINIMUM	3.729274
MAXIMUM	13.69137

FREQ	CLASSES	ENTRS	PERC	CUM	101	201	301	401	501	601	701	801	901	1001
<	0.0	0	0.00	0.00										
<	0.5	0	0.00	0.00										
<	1.0	0	0.00	0.00										
<	1.5	0	0.00	0.00										
<	2.0	0	0.00	0.00										
<	2.5	0	0.00	0.00										
<	3.0	0	0.00	0.00										
<	3.5	0	0.00	0.00										
<	4.0	1	11.11	11.11										
<	4.5	0	0.00	11.11										
<	5.0	0	0.00	11.11										
<	5.5	0	0.00	11.11										
<	6.0	0	0.00	11.11										
<	6.5	1	11.11	22.22										
<	7.0	0	0.00	22.22										
<	7.5	0	0.00	22.22										
<	8.0	0	0.00	22.22										
<	8.5	0	0.00	22.22										
<	9.0	1	11.11	33.33										
<	9.5	0	0.00	33.33										
<	10.0	0	0.00	33.33										
<	10.5	0	0.00	33.33										
<	11.0	1	11.11	44.44										
<	11.5	2	22.22	66.67										
<	12.0	0	0.00	66.67										
<	12.5	0	0.00	66.67										
<	13.0	1	11.11	77.78										
<	13.5	1	11.11	88.89										
<	14.0	1	11.11	100.00										
REMAINDER EMPTY														

WACHTTIDEN GAS OIL

ENTRIES	9
ZERO ENTRIES	0
MEAN	15.90044
STD. DEVIATION	4.51126
MINIMUM	8.64987
MAXIMUM	21.65196

[illegible]

ENTRIES	9
ZERO ENTRIES	0
MEAN	5.35158
STD.DEVIATION	4.5583
MINIMUM	0.4250564
MAXIMUM	14.1439

FREQ	CLASSES	ENTRS	PERC	CUM	PERC	101	201	301	401	501	601	701	801	901	1001
0.0		0	0.00	0.00											
0.5		1	11.11	11.11											
1.0		1	11.11	22.22											
1.5		0	0.00	22.22											
2.0		1	11.11	33.33											
2.5		0	0.00	33.33											
3.0		0	0.00	33.33											
3.5		1	11.11	44.44											
4.0		0	0.00	44.44											
4.5		0	0.00	44.44											
5.0		0	0.00	44.44											
5.5		1	11.11	55.56											
6.0		1	11.11	66.67											
6.5		0	0.00	66.67											
7.0		0	0.00	66.67											
7.5		1	11.11	77.78											
8.0		0	0.00	77.78											
8.5		0	0.00	77.78											
9.0		0	0.00	77.78											
9.5		0	0.00	77.78											
10.0		1	11.11	88.89											
10.5		0	0.00	88.89											
11.0		0	0.00	88.89											
11.5		0	0.00	88.89											
12.0		0	0.00	88.89											
12.5		0	0.00	88.89											
13.0		0	0.00	88.89											
13.5		0	0.00	88.89											
14.0		0	0.00	88.89											
14.5		1	11.11	100.00											
REMAINDER EMPTY															

ENTRIES	9
ZERO ENTRIES	0
MEAN	8.42414
STD. DEVIATION	3.83733
MINIMUM	4.2265
MAXIMUM	16.1287

FREQ CLASSES	ENTRS	PERC	CUM PERC
< 0.0	0	0.00	0.00
< 0.5	0	0.00	0.00
< 1.0	0	0.00	0.00
< 1.5	0	0.00	0.00
< 2.0	0	0.00	0.00
< 2.5	0	0.00	0.00
< 3.0	0	0.00	0.00
< 3.5	0	0.00	0.00
< 4.0	0	0.00	0.00
< 4.5	1	11.11	11.11
< 5.0	0	0.00	11.11
< 5.5	0	0.00	11.11
< 6.0	1	11.11	22.22
< 6.5	1	11.11	33.33
< 7.0	2	22.22	55.56
< 7.5	0	0.00	55.56
< 8.0	0	0.00	55.56
< 8.5	1	11.11	66.67
< 9.0	1	11.11	77.78
< 9.5	0	0.00	77.78
< 10.0	0	0.00	77.78
< 10.5	0	0.00	77.78
< 11.0	0	0.00	77.78
< 11.5	0	0.00	77.78
< 12.0	0	0.00	77.78
< 12.5	0	0.00	77.78
< 13.0	0	0.00	77.78
< 13.5	1	11.11	88.89
< 14.0	0	0.00	88.89
< 14.5	0	0.00	88.89
< 15.0	0	0.00	88.89
< 15.5	1	11.11	100.00

WACHTT IJDEN MIDDLE CRUDE

ENTRIES	9
ZERO ENTRIES	0
MEAN	6.73527
STD. DEVIATION	1.781865
MINIMUM	5.00988
MAXIMUM	10.47575

[illegible]

WACHTIJUDEN BIG CRUDE

ENTRIES	9
ZERO ENTRIES	0
MEAN	5.98644
STD.DEVIATION	1.826738
MINIMUM	3.132487
MAXIMUM	8.8143

FREQ	CLASSES	ENTRS	PERC	CUM	PERC	CUM	101	201	301	401	501	601	701	801	901	1001
<	0.0	0	0.00	0.00	0.00	0.00										
<	0.5	0	0.00	0.00	0.00	0.00										
<	1.0	0	0.00	0.00	0.00	0.00										
<	1.5	0	0.00	0.00	0.00	0.00										
<	2.0	0	0.00	0.00	0.00	0.00										
<	2.5	0	0.00	0.00	0.00	0.00										
<	3.0	0	0.00	0.00	0.00	0.00										
<	3.5	2	22.22	22.22	22.22	22.22										
<	4.0	0	0.00	0.00	22.22	22.22										
<	4.5	0	0.00	0.00	22.22	22.22										
<	5.0	0	0.00	0.00	22.22	22.22										
<	5.5	1	11.11	33.33	33.33	33.33										
<	6.0	0	0.00	33.33	33.33	33.33										
<	6.5	2	22.22	55.56	55.56	55.56										
<	7.0	3	33.33	88.89	88.89	88.89										
<	7.5	0	0.00	88.89	88.89	88.89										
<	8.0	0	0.00	88.89	88.89	88.89										
<	8.5	0	0.00	88.89	88.89	88.89										
<	9.0	1	11.11	100.00	100.00	100.00										

REMAINDER EMPTY

MSG06: SIMULATION STOPS, BECAUSE THE STATUS OF THE SYSTEM -GIVEN BELOW- WILL REMAIN UNCHANGED

