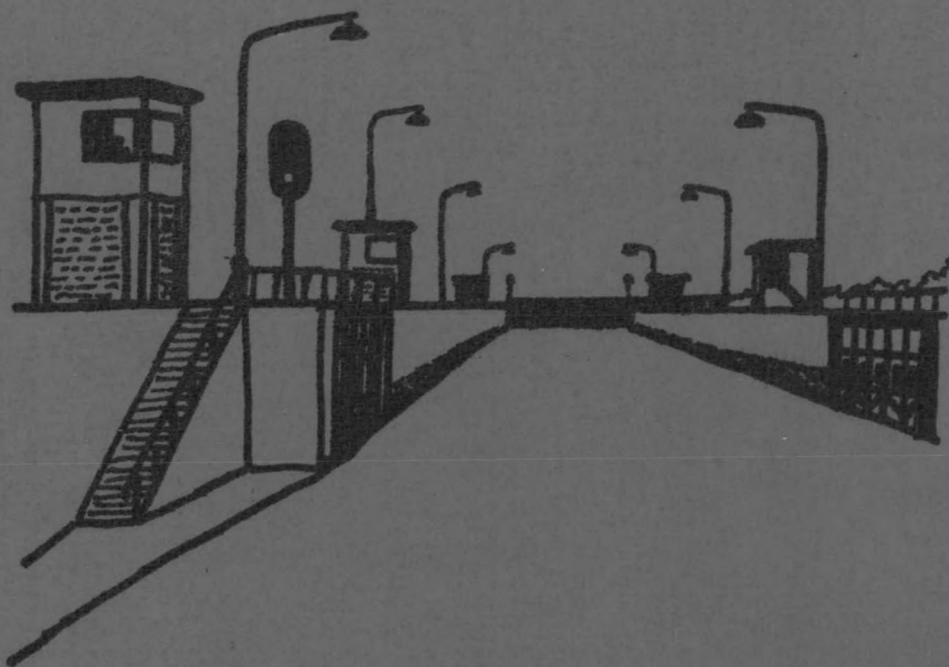


S26



VAKGROEP  
WATERBOUWKUNDE  
Afd. Civiele Techniek  
TH Delft



SLUIZEN

SLUISSIMULATIE

KANAALOPTIMALISATIE

DEEL 1

D. OOSTERVELD

FEBRUARI 1976

SLUIS-SIMULATIE MODEL

SIMULATIE VAN EEN  
NETWERK VAN KANALEN

DEEL I van het

Afstudeerverslag van

D. Oosterveld.

Delft,  
november 1975.



## INHOUDS OPGAVE :

Inleiding	pag.	1
Hoofdstuk I Simulatie m.b.v. PROSIM	pag.	11
Hoofdstuk II Algemene uiteenzetting van het model	pag.	21
Hoofdstuk III De scheepsklassen en het genereren van schepen	pag.	27
Hoofdstuk IV Het fysisch systeem bij een sluis	pag.	43
Hoofdstuk V Model analogie van het sluisgebeuren	pag.	63
Hoofdstuk VI Netwerk van kanalen en vervolg van het hoofdproces	pag.	82
Hoofdstuk VII Mogelijkheden van het model en indeling van uitvoergegevens	pag.	98

## BIJLAGEN.

A: Kapaciteitsberekeningen	pag.	48
B: Simulatie model	pag.	103
C: Tabel van scheepstypen	pag.	116
D: Schema van een sluis met één kolk	pag.	117
E: Lustijden - Invaartijden en Witvaarvolgtijden	pag.	119
F: Snelheden op de vaarweg	pag.	132
G: Meetresultaten Kolkindeling	pag.	148

## INLEIDING

In het scheepvaartverkeer constateert men regelmatig, dat schepen bewust een andere route kiezen, dan de gebruikelijke of kortste in verband met langere wachttijden bij schutsluizen, bruggen enz.

Over 't algemeen wegen de kosten en tijdwinsten om van de ene naar de andere plaats te komen zwaarder, dan de af te leggen afstand.

Ten behoeve van de aanleg van nieuwe kanaalpannen of van verbeteringen van schutsluizen, is het voor de beheerder van de scheepvaartwegen noodzakelijk een inzicht te hebben in het gedrag van het scheepvaartverkeer, in het bijzonder daar waar een verandering in het scheepvaartverkeer optreedt. Dit kan zijn een veranderd scheepsaanbod of/en andere typen schepen of enige andere oorzaak.

In Nederland is de beheerder van vrijwel alle scheepvaartwegen de Rijkswaterstaat, die, om inzicht in het gedrag van het scheepvaartverkeer te verkrijgen hiervoor legio aan gegevens verzameld. Deze dienst treedt tevens als regelaar van het scheepvaartverkeer op en heeft ten aanzien van dit scheepvaartverkeer vergaande bevoegdheden verkregen.

Het is dan ook ondenkbaar een probleem op verkeerswaterbouwkundig gebied te gaan onderzoeken zonder gebruik te maken van en samen te werken met de Rijkswaterstaat.

In het onderzoek dat in deze problematiek gedaan wordt, wordt dan ook gebruik gemaakt van de definities en reglementen, die deze dienst hanteert. Veelvuldig zal verwezen worden naar de nota's van de Hoofdafdeling Verkeerskunde van de Rijkswaterstaat. In deze nota's zijn definities en begrippen geformuleerd t.a.v. capaciteitsberekeningen en wachttijdproblemen o.a. betreffende sluizen in de binnenwateren.

Deze studie omvat drie elementen:

a. Een simulatiemodel van een schutsluis

Deze simulatie zal gedaan worden m.b.v. de computer en wel

met de simulatietaal PROSIM.

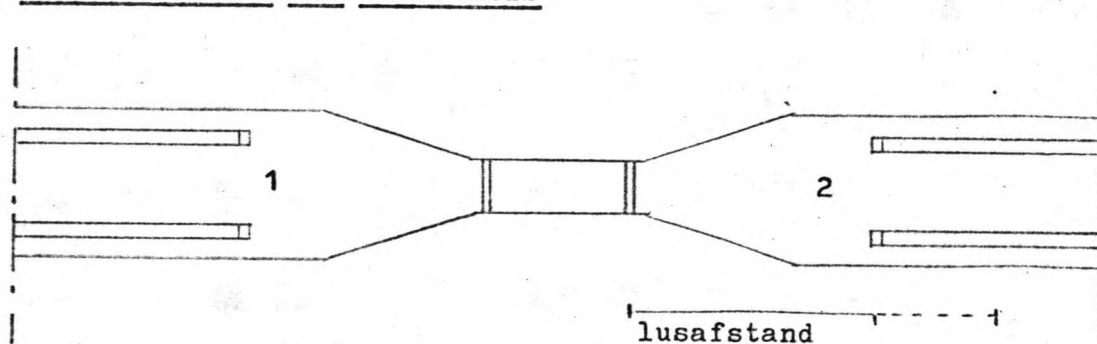
b. Een simulatiemodel van een kanaal en een netwerk van kanalen

Een kanaal bestaat uit een aantal schutsluizen, bruggen en andere "delay's". Een schip, dat uit verschillende kanalen kan kiezen, zal het kanaal kiezen, waardoor het het "goedkoopst" op de plaats van bestemming kan zijn. Dit simulatiemodel is dan ook geschikt om een scheepvaartwegennet te simuleren.

c. Een optimalisatiemodel met behulp van het in a. ontwikkelde simulatiemodel.

Enkele optimalisaties zullen in deel II van deze studie besproken worden.

SIMULATIEMODEL VAN SCHUTSLUIS



schema schutsluis met 1 sluiskolk

Procesomschrijving

Van beide zijden komen schepen bij de schutsluis aan.

De tussenaankomsttijden bij de sluis van een bepaald type schip, worden ontleend aan gegevens van de Rijkswaterstaat voor verschillende schutsluizen.

Deze verdeling van de tussenaankomsttijden kan willekeurig verdeeld zijn of gedoseerd.

Een bepaald type schip (b.v. pleziervaart) kan prioriteit bezitten t.a.v. de plaats in de wachtrij.

Het scheepsaanbod is afhankelijk van het seizoen en van de dag.

De sluiswachter begint met de schepen om te schutten aan de kant waar het waterniveau gelijk is aan die in de kolk. Indien hier geen schip is en aan de andere zijde wel, kolkt hij de sluis om, tenzij hij binnen een bepaalde tijd een schip verwacht aan betreffende zijde. In 't algemeen zal het scheepsaanbod zo groot zijn, dat hij niet "leeg" behoeft om te kolken. Elke sluis heeft een bepaalde bedieningstijd  $T_b$ .

Bij een sluis met één kolk leggen de schepen achter elkaar aan aan de zijde waar de wind vandaan komt. Is deze wachtrij "vol", dan legt men aan de andere zijde aan.

Een schip dat aankomt sluit achter in de wachtrij aan. Het schip behoudt zijn plaats in de wachtrij, met in achtneming van schepen met prioriteit. Dit aanleggen heeft gevolgen voor de invaartijden.

De sluisgrootte kan van dien aard zijn, dat niet met aantallen schepen gewerkt wordt maar met een aantal gewichtseenheden (tonnen), dat een sluis kan bevatten. In het model worden echter de schepen ook afzonderlijk in de sluis kolk "geplaatst".

Is de sluis bijna vol, dan kan de sluiswachter beslissen, of een schip, dat nog in de sluis "past" niet toegelaten wordt t.b.v. een "grotere" of twee of meer "kleinere" schepen, die samen "groter" zijn.

Voorbeeld: Sluis capaciteit: 3000 ton; ingevaren: 2700 ton; over: 300 ton; beschikbare schepen in de volgorde van de wachtrij: 150 ton, 200 ton, 100 ton, 300 ton. De sluiswachter zal beslissen dat het tweede en derde schip binnenvaren, mits aan het eerste schip geen prioriteit verleend is.

Wachttijden, in- en uitvaartijden, bedieningstijden enz. zijn afhankelijk van het schip, de weersomstandigheden, deel van de dag, de sluisomstandigheden enz.

Van al deze afhankelijkheden zijn voor het te ontwerpen simulatiemodel voldoende gegevens beschikbaar.

Opbouw van het simulatiemodel:

Het programma is onder te verdelen in vijf onderdelen:

- a. Hoofdproces met de initialisatie van het systeem en uitvoer verzorging.
- b. Het proces van de schepen generatoren.
- c. Het proces van het passeren van een sluis.
- d. Het proces van de sluismeester.
- e. Procedures.

a. Hoofdproces

b. De schepengenerator

Een schip wordt gegeneerd op het moment dat de achterstevan van een schip de sluisvoorhaven passeert.

De schepen worden in Nederland onderverdeeld in 8 typen, gerangschikt naar laadvermogen klasse:

No.	Cargo cap. class (metric tons)	Data of standard ships						
		Cargo cap. (metric tons)	length (m)	beam (m)	draft (m)	f (m <sup>2</sup> )	$\Delta$ (m <sup>3</sup> )	P (hp)
1	50- 199	125	25	4.6	1.6	7.4	165	60
2	200- 449	325	39	5.1	2.3	11.7	425	140
3	450- 749	550	50	6.6	2.5	16.5	710	260
4	750-1,149	925	67	8.2	2.5	20.5	1,180	520
5	1,150-1,549	1,350	80	9.5	2.6	24.7	1,740	730
6	1,550-2,549	2,000	95	11.5	2.7	31.0	2,600	1,100
7	2,550-4,999	4,100	175	11.4	3.0	34.2	5,150	1,200
8	> 5,000	8,800	185	22.8	3.1	73.0	10,600	2,500

De laatste twee (7 en 8) zijn duweenheden, wat voor een aantal kanalen en rivieren extra problemen geeft, daar vele scheepvaartwegen hier nog niet geschikt voor zijn. Het plan is om een aantal scheepvaartwegen hiervoor geschikt te maken.

Het willekeurige aanbod van schepen is, zoals aangetoond door prof. dr. ing. W. Leutzbach, poisson verdeeld. Vaak zullen we echter

met een gedoseerd aanbod te maken hebben, gezien de beperkte afstanden waarop de schutsluizen in vele kanalen zijn gelegen.

Er zijn meerdere mogelijkheden om een gedoseerd aanbod in het simulatiemodel te verwerken, zoals een verdeling zoeken voor het gedoseerde aanbod per schutsluis of, gezien de lange reeksen waarnemingen, het werkelijke aanbod als schepengenerator in te voeren.

De gegevens, die bij de schepengenerator ingevoerd moeten worden zijn vermeld bij het proces van de schepen.

c. Proces van de schepen

Van het schip zijn de volgende grootheden bekend:

1. tonnage
2. breedte
3. lengte
4. geladen of ongeladen
5. snelheid - als functie van het doorvaartprofiel en gemiddelde beladingsgraad

De invoer zal bestaan uit het vermelden van de klasse, de andere gegevens zijn bepaald door de klasse.

d. Het proces van de sluismeester:

Voor het sluisproces is het van belang te weten hoelang een schip nodig heeft om in- en uit te varen. Dit wordt gedeeltelijk bepaald door de sluisomstandigheden.

De belangrijke parameters van de sluis zijn:

1. lengte kolk
2. breedte kolk
3. breedte kolkdoorgang
4. diepte kolkdoorgang
5. te bevatten tonnage
6. lusafstand
7. bedieningstijd
8. kolk-voorhavengetal.

Voor de definities van deze grootheden zie bladzijde 119

e. Procedures



Het proces van het schip is bij een sluis als volgt te omschrijven:

1. Een schip komt aan zijde 1 of zijde 2 aan.
2. De sluiswachter wordt hierover ingelicht.
3. Indien de sluiswachter bezig is, moet dit schip aansluiten in de wachtrij, anders gaat de sluiswachter maatregelen nemen.
4. Het schip wacht totdat het binnen kan varen, wachttijd  $t_w$ .
5. Het schip wacht totdat het geschut is, wachttijd  $t_s$ .
6. Indien het schip niet met de eerst-volgende schutting mee kan, dan moet het schip de wachttijd  $t_o$  (overligtijd) extra wachten.
7. Uitvaren
8. Uit het systeem.

Het proces van de sluis:

1. Is er een schip? Aan welke zijde?  
Indien er geen schip is, gaat de sluiswachter naar zijn kantoor.
2. Anders:  
Zet kolk om naar de zijde waar het schip is. Bedieningstijd  $T_b$  of 0
3. 1<sup>e</sup> schip vaart in. Lustijd:  $T_l$
4. Volgende schepen varen in. Invaartijd  $T_i$ .
5. Bedieningstijd  $T_b$ .
6. Schepen varen uit. Uitvaartijd:  $T_u$ .
7. Einde proces sluis, ga verder bij punt 1.

Een uitbreiding van een simulatiemodel voor één kolk per schutsluis is dat met 2 of meerdere kolken.

Het proces van de schepengenerator blijft hetzelfde. De wachtrij verandert nu, er leggen nu zoveel schepen aan aan de zijde waar het eerst de kolk opengaat, als er schepen in die kolk kunnen, de volgende schepen leggen aan de andere zijde aan, enz.

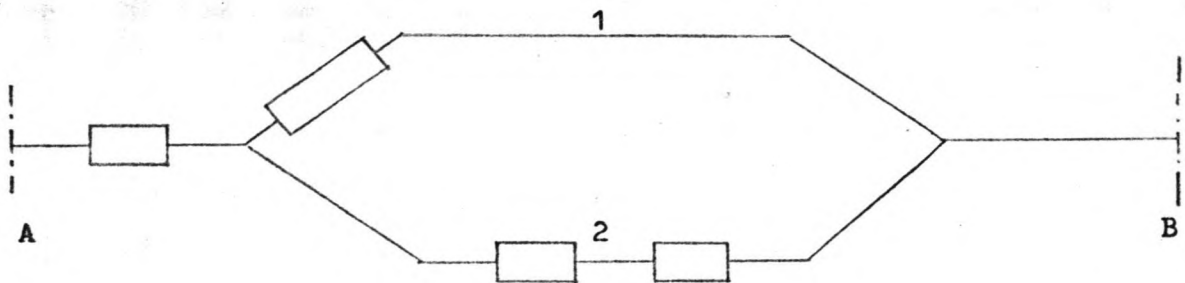
Bij een sluis met 2 kolken kan tegelijkertijd geschut worden.

In dat geval wordt water van de kolk met een hoog water niveau in de kolk met laag waterniveau ingelaten.

Het proces van de sluiskolk blijft verder hetzelfde, bij extreem laag scheepsaanbod wordt slechts één kolk gebruikt!

Bij sluizen met meer dan 2 kolken kan een soort gelijk proces gevolgd worden.

#### SIMULATIEMODEL VAN EEN KANAAL



schema van een kanaal met twee kanaal-takken

#### Procesomschrijving

Vertragingen bij het doorvaren van een kanaal kunnen ontstaan bij een aantal schutsluizen, bruggen, gedeelten waar een schip op-onthoud krijgt, aanlegplaatsen enz.

Te simuleren is nu hoe lang een schip nodig heeft om dit kanaalpand of riviergedeelte te passeren. Indien een schip een andere route kan kiezen, zal het, indien het die andere route "goedkoper" kan afleggen, die route kiezen.

Het beslissende criterium betreft dus uitsluitend de kosten, die een schip heeft om van de ene plaats naar de andere te komen. Dit geldt niet voor de pleziervaart.

Voor de eenvoud zal eerst de bovenstaande situatie bekeken worden.

Een schip wordt in A c.q. B gegenereerd. Bij de generator moet het schip meegedeeld worden of het route 1 (zie fig.) of route 2

moet kiezen, om zo goedkoop mogelijk in B (in deze beschrijving volgen we een schip van A naar B) te zijn. Hierbij is het mogelijk, dat bepaalde schepen, gezien b.v. hun afmeting of bestemming genoodzaakt zijn route 2 te kiezen, weer andere schepen laat men de keuze vrij (b.v. pleziervaart).

In de praktijk zal met de schippers niet dwingen een bepaalde route te kiezen, in dit model gaan we er van uit, dat de schippers deze keuze als een "verkeersregeling" zullen beschouwen en dan de aan te wijzen route wel zullen nemen.

Het kanaalsimulatiemodel is onder te verdelen in de passeertijd van de schutsluizen en gemakshalve, in eerste instantie, in de passeertijd van de rest van het kanaal.

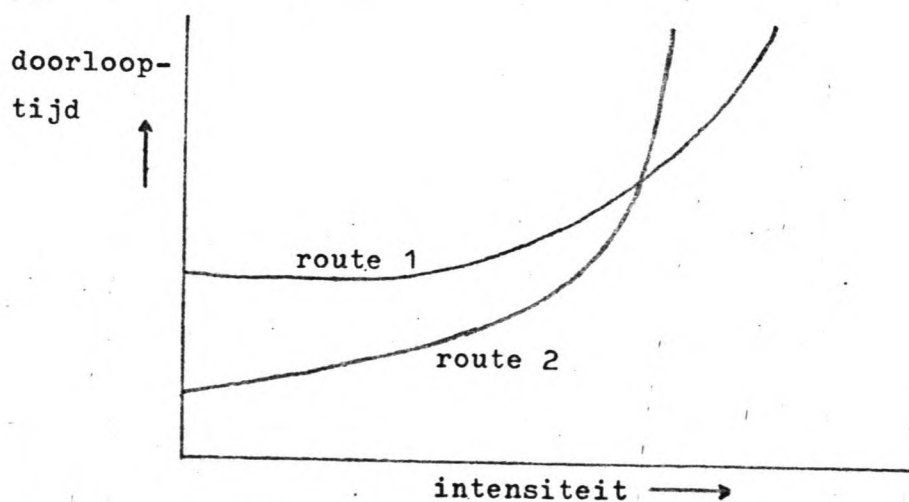
In een later stadium zullen tevens de wachttijden bij bruggen bekeken worden, terwijl dan ook het geval van een bestemming in het beschouwde gebied bekeken zal worden. Een schip kan dan binnen het systeem gegenereerd worden.

Nu is het van belang een beslissingskriterium voor de aan te wijzen route te vinden.

Hiervoor zijn verschillende mogelijkheden aan te wijzen:

1. Een relatie zoeken tussen de doorlooptijd en de intensiteit van de kanaal-panden en een schip de gunstigste doorlooptijd laten kiezen.

Voorbeeld:



Deze methode heeft het nadeel, dat voor elk kanaalpand deze lijn gezocht moet worden.

2. Een andere mogelijkheid is, doordat over 't algemeen de capaciteit van een kanaal of riviergedeelte door de sluis met de kleinste capaciteit bepaald wordt, de relatie Intensiteit/Kapaciteit (gemeeten bij de sluis met de kleinste capaciteit per kanaalvak in het simulatiemodel) tegen de doorlooptijden uit te zetten. Eventueel voor elk scheepstype afzonderlijk.

3. Bij beide voorgaande methoden ontstaat de moeilijkheid, dat de totale doorlooptijd bepalende schutsluis door de schepen pas na zeer lange tijd bereikt kan worden, terwijl dan inmiddels de doorlooptijd van die sluis sterk vergroot kan zijn.

De mogelijkheid, die het eenvoudigst is, is die, waarbij na verloop van tijd de verhoudingen waarin de schepen de verschillende routes kiezen aan te passen aan de hand van een kosten-baten berekening.

Onderzocht moet worden of deze iteratieve methode convergeert en wat de snelheid hiervan is.

Om de kanaalsimulatie uitgebreider te maken, moeten nog de doorlooptijden van de verschillende type schepen vermeld worden.

## NASCHRIFT

Deze inleiding is, uitgezonderd enkele veranderingen, de probleemstelling en de eerste "gedachte" geweest, voordat aan het simulatiemodel begonnen werd.

Het hele model is de meeste punten veel uitgebreider geworden, dan uit deze inleiding is op te maken.

Dit gedeelte werd geschreven als introductie op het maken van een kanaal simulatiemodel van het systeem, dat in deze handleiding is getekend. Het model is nu geschikt om elke willekeurig netwerk van kanalen te simuleren.

In het kader van het afstuderen is het onmogelijk gebleken een volledig model te maken. Zeer veel punten zijn schematisch opge-

nomen en voor verdere studie aan te bevelen.

Een volledige toetsing van de optimalisatie was even-  
mogelijk, maar is zeer zeker een goede aanzet om tot een  
optimalisatiemodel te komen.

november 1975

D. Oosterveld.

## Hoofdstuk I

1. Simulatie is de zo natuurgetrouw mogelijke nabootsing van de werkelijkheid.

In de praktijk valt de simulatie niet zo eenvoudig uit te voeren als deze uitspraak zou doen vermoeden, daar elke factor, die een rol speelt in de werkelijkheid, eveneens een factor behoort te zijn in de simulatie.

Voor zover aan deze laatste eis niet of niet geheel voldaan kan worden, moet men wel een beeld hebben wat de invloed van de fout op het systeem is en wat de grootte van deze fout is.

Simulatie kan op vele manieren uitgevoerd worden. De gebruikelijke vormen voor toepassing in de techniek zijn o.a.:

- a. De werkelijkheid nabootsen in een fysisch model, meestal op verkleinde schaal.
- b. nabootsen d.m.v. een elektrisch analogon.
- c. simulatie m.b.v. elektronische rekenmachines.

De laatste, simulatie m.b.v. de computer, is in dit onderzoek gebruikt om een sluis, een kanaal en een netwerk van kanalen met sluizen in haar relatie tot de scheepvaart te simuleren. In dit hoofdstuk vindt u een beschrijving van simulatie met behulp van de computer.

In de praktijk komen twee soorten fysische systemen voor, te weten continue systemen en discrete systemen.

Continue systemen laten zich het best simuleren op analoge computers, gezien het continue karakter van de verwerking van informatie door deze rekenmachines.

"Computertalen" voor continue simulatie op digitale computers zijn o.a.:

C.S.M.P. en DYNAMO.

Discrete systemen kunnen zeer goed op digitale computers gesimuleerd worden.

We spreken dan van discrete simulatie.

Bekende discrete simulatie talen zijn o.a.:

SIMULA, C.S.L., GASP, GPSS, SIMSCRIPT en PROSIM.



Gezien het discreet zijn van zowel de aankomsten van schepen bij een sluis (c.q. het passeren van een raai in een waterweg, maar gemakshalve spreken we in dit hoofdstuk slechts over het eerste gedeelte van de gesimuleerde systemen) als de doorvaart door de sluis ligt het voor de hand een discrete simulatietaal te gebruiken.

In deze studie is gebruik gemaakt van de simulatietaal PROSIM.

Discrete simulatiemethoden zijn:

a. Gebeurtenis-beschrijvingsmethode  
(event-description).

Hierin verondersteld men dat een gebeurtenis geen tijd in beslag neemt. Een gebeurtenis (event) is het begin of het einde van een handeling of een gebeuren.

Steeds als een gebeurtenis plaats vindt veroorzaakt dit een verandering in de waarde van een of meer toestandsvariabelen in een discreet systeem.

Voorbeeld:

Op het moment, dat een schip in de sluis vaart, verandert er iets aan de toestand van het systeem. Uitgezocht moet worden welke gebeurtenis het eerst volgt.

b. Aktiviteiten-methode

In deze benadering komen akties inplaats van gebeurtenissen. Net als bij een gebeurtenis verandert er tenminste één toestandvariabele door een aktie.

Voorbeeld:

Een schip moet meerdere activiteiten verrichten om een sluis te passeren. Steeds als het schip geactiveerd wordt, moet worden uitgezocht, welke activiteiten het moet verrichten.

Het verschil tussen de twee methoden is:

1. Een aktie vindt plaats en alleen dan plaats als de waarden van de toestandsvariabelen samen met de systeemtijd aan een aantal voorwaarden voldoen.
2. Een gebeurtenis vindt plaats op een tijdstip (begin of einde) van een gebeuren.

c. Proces-beschrijvingsmethode

Een proces karakteriseert het gedrag van een systeemcomponent gedurende de tijd dat deze component in het systeem vertoeft.

Een proces behoort tot een groep van processen als het op dezelfde manier beschreven wordt als een ander proces.

Zo'n proces wordt een proces-beschrijving genoemd en resulteert in een stukje programma.

Steeds als zo'n stukje programma gestart wordt met de verkregen waarden van de parameters, wordt een proces gesimuleerd. De simulatie kan echter niet zoals in een schaalmodel, twee of meer componenten tegelijkertijd volgen, terwijl het eveneens niet mogelijk is een proces in één keer achterelkaar te laten verlopen, daar gedurende het proces de gevolgde component afhankelijk is van andere, eventueel zelfs van nog niet aanwezige componenten afhankelijk is. De verandering van waarde van attribuut notities hangt af van de toestand van andere processen. Dus, als een proces gestart wordt kan het daarbij behorende programma draaien totdat een opdracht in het programma verwijst naar een later tijdstip. Het verwerken van die opdracht moet uitgesteld worden totdat geen enkel deel van enig ander proces meer plaats kan vinden vóór dat tijdstip. De verwijzing naar zo'n opdracht wordt een reactivatie punt genoemd.

Attributen kunnen toegewezen worden aan een proces (SIMULA) of aan de componenten die een proces doorlopen (PROSIM).

PROSIM is gebaseerd op deze laatste methode:

de proces-beschrijvingsmethode.

Het voordeel hiervan is dat het programma het systeem op de meest natuurlijke wijze beschrijft.

In de volgende paragraaf vindt u een beschrijving van de computertaal PROSIM.

2. Simulatie met behulp van PROSIM

2.1. PROSIM is een taal die gebaseerd is op PLI, en ontwikkeld is

om systemen door middel van proces-beschrijvingen te simuleren. Het simulatieprogramma bestaat uit een aantal proces-beschrijvingen, die in een willekeurige volgorde in het programma zijn opgenomen, voorafgegaan door de proces-beschrijving van de hoofdcomponent.

Het hoofdproces wordt door PROSIM gestart en de programmeur kan het gebruiken om attribuutnotities van de componenten en de systeemvariabelen te declareren.

Voorbeeld:

Een component is b.v. een sluis, die als attributen een sluislengte, sluisbreedte, enz. heeft.

Een systeemvariabele is b.v. het aantal runs dat men wil draaien.

Het hoofdproces moet tenminste één opdracht bevatten om een ander proces op gang te brengen.

Voor elke klasse van componenten, waarvoor de programmeur voor elk element afzonderlijk systeem faciliteiten wenst moet een BASED STRUCTURE gedeclareerd worden.

Voorbeeld:

1. ATTRLIJST BASED (SCHIP),
2. LENGTE
2. BREEDTE

• alle andere attributen van een schip.

De opdracht ALLOCATE ATTRLIJST reserveert geheugenruimte voor de attribuutnotitie van de component dat schip heet, maar voordat dit schip in een proces gebracht wordt moet het eerst aan het systeem "bekend" gemaakt worden.

Dit gebeurt d.m.v. de opdracht: SCHIP=NEW(SCHIP);

- 2.2. PROSIM bevat een aantal procedures waarvan de in het model gebruikte hieronder beschreven zijn:

- 2.2.1. PROCEDURE NOW RETURNS(FLOAT);

NOW geeft de kloktijd van het model aan.

In het model is gekozen voor de tijdseenheid minuut. Dit in verband met de invaar-en uitvaar-tijden in sluisen, die eveneens in minuten gegeven worden. De minuten worden in

decimale cijfers weergegeven, niet te verwarren met seconden.

Als een nieuwe component in het systeem gebracht wordt, vergezeld hem als attribuut zijn "arrivaltime".

De kloktijd moet als volgt worden geïnterpreteerd: Elke component heeft een eventtijdstip dat ook als attribuut wordt meegenomen.

Al deze tijdstippen staan gerangschikt naar grootte in de z.g. "sequentiewachtrij".

Het sequentiemechanisme zorgt ervoor dat elke component "aan de beurt is" als het de eerst-volgende component van de sequentiewachtrij is.

Wordt NOW aangeroepen, dan is dit altijd tevens het eventtijdstip van de component die "aan de beurt is" (current is).

#### 2.2.2. PROCEDURE ACTIVATE;

Met deze procedure wordt een component geactiveerd.

Voorbeeld: CALL ACTIVATE(NEW(SCHIP),AANKOMST);

Hierin wordt ten eerste een schip bekend gemaakt aan het systeem en ten tweede geactiveerd bij de label AANKOMST.

Binnen de proces-beschrijving van deze component mogen de attributen los van hun drager (pointer) gebruikt worden.

Voorbeeld:

In het proces van een schip is "schip" de drager en b.v. de lengte een attribuut. I.p.v. "schip lengte" mag binnen dit proces "lengte" staan.

#### 2.2.3. PROCEDURE PASSIVATE;

De current component wordt passief als het de opdracht CALL PASSIVATE tegenkomt.

Hierbij wordt geen nieuwe eventtime voor deze component vastgesteld, dus, een andere component zal deze component moeten reactiveren. Dit reactiveren gebeurt dan bij de volgende opdracht in het proces van de gereactiveerde component, tenzij deze component bij b.v. de label KOLKINDELEN moet vervolgen. Dan luidt de opdracht CALL PASSIVATE(KOLKINDELEN).

#### 2.2.4. PROCEDURE CANCEL;

De component, die van de sequentiewachtrij moet verdwijnen,

wordt verwijderd door CALL CANCEL.

Voorbeeld:

De component GENER(1,1) moet verwijderd worden:

```
CALL CANCEL(GENER(1,1));
```

2.2.5. PROCEDURE REACTIVATE;

Hiervoor gelden dezelfde regels als voor PROC. ACTIVATE met dit verschil dat een component een reactivatie punt moet hebben.

2.2.6. PROCEDURE HOLD;

Het proces van de current component wordt onderbroken.

Deze component wordt ingepast in de sequentiewachtrij.

Bij het gebruik van deze procedure moet opgegeven worden hoelang het proces van de component onderbroken moet worden en waar dan zijn reactivatie punt is. Bij het ontbreken van de laatste parameter vervolgt de component bij de volgende opdracht.

Voorbeeld:

```
CALL HOLD(DOORVAARTIJD,BIJSLUIS);
```

De current component wordt gedurende de doorvaartijd onderbroken en hervat daarna zijn proces bij de label: BIJSLUIS.

2.2.7. PROCEDURE ARRIVALTIME;

Arrivaltime is in feite een attribuut van een component, maar het wordt automatisch toegevoegd aan de attribuutlijst van alle componenten.

Het bevat de aankomsttijd van de component in het systeem.

2.2.8. PROCEDURE TERMINATE;

De relatie tussen de current component en het systeem wordt beëindigd.

Het systeem vervolgt, net als bij het passiveren van een component, bij de eerstvolgende component, die voorkomt in de sequentiewachtrij.

3. Wachtrijen van de gebruiker.

We kunnen elke wachtrij het best beschouwen als een ketting.

In een wachtrij staan alle componenten die in die wachtrij gebracht zijn middels nog te bespreken procedures. De plaats in die wachtrij kan op verschillende manieren aangewezen worden. Elke ketting bestaat uit een aantal schakels, die achter elkaar geplaatst zijn en verbonden zijn met elkaar door middel van het



aanwijzen van zijn voorganger en zijn opvolger.

Een ketting wordt gesloten door een sluiting.

Dit is een fictieve schakel, die niet tot de ketting behoort, maar wel de eerste en de laatste component van die wachtrij aanwijst en aangeeft hoeveel schakels zich in de wachtrij bevinden.

Steeds als een component in een wachtrij komt, wordt het aantal schakels met één verhoogt en omgekeerd.

Bij de initialisatie van een nieuwe wachtrij wordt tevens de sluiting aangebracht, de wachtrij is dan nog "leeg" (er liggen nog geen schepen in de wachtrij), dan geeft deze sluiting aan: NULL,NULL,0, (geen eerste, geen laatste schip en 0 schepen in de wachtrij).

Indien nu een schip met de naam BOOT in deze wachtrij komt geeft de sluiting aan:

BOOT,BOOT,1.

Vanaf de eerste en de laatste schakel heeft men toegang tot alle schakels. Op het moment dat een component in een wachtrij komt wordt dat tijdstip als attribuut voor die component genoteerd. Een reeds door PROSIM geïntialiseerde "wachtrij" is de future event list.

3.1. Het aanvragen van de aankomsttijd in de wachtrij van een component gaat als volgt:

AANKOMSTTIJDINWRIJ=QUEUETIME binnen het proces van die component, anders: b.v.

AANKOMSTTIJDINWRIJ=QUEUETIME(BOOT);

Queue time is evenals arrivaltime een door PROSIM toegevoegd attribuut aan een component.

Dit geldt uiteraard alleen voor de gebruikers-wachtrijen.

Van een wachtrij wordt verder bijgehouden:

- a. totaal aantal aankomsten,
- b. gemiddelde wachttijd, (MEANWT)
- c. maximum wachttijd,
- d. minimum wachttijd.
- e. aantal componenten met wachttijd 0,
- f. op elk moment de lengte van de wachtrij (LENGTH)
- g. de maximale en minimale lengte.



Hiervan zijn MEANWT en LENGTH op elk gewenst moment op te vragen.

Voorbeeld:

WACHTRIJLENGTE=LENGTH(RIJ(1,1));

geeft de lengte van de wachtrij RIJ(1,1).

GEMWACHTTIJD=MEANWT(RIJ(1,1));

geeft de gem. wachttijd in de wachtrij RIJ (1,1).

3.2.1. PROCEDURE NEWQUEUE;

INVAARVOLG=NEWQUEUE initialiseert een nieuwe wachtrij met de naam INVAARVOLG.

Ook een element van een array kan geïnitieerd worden:

RIJ(2,2)=NEWQUEUE.

Wachtrijen moeten gedeclareerd zijn als pointer variabelen.

3.2.2. PROCEDURE TOTAILOFQUEUE;

De current component wordt achter aan de wachtrij geplaatst door:

CALL TOTAILOFQUEUE(INVAARVOLG);

Indien niet de current component in de wachtrij INVAARVOLG geplaatst moet worden: CALL TOTAILOFQUEUE(BOOT,INVAARVOLG);

3.2.3. PROCEDURE TOHEADOFQUEUE;

Dezelfde regels als de vorige procedure, maar nu wordt de component voor aan de wachtrij geplaatst.

3.2.4. PROCEDURE SORTINQUEUE;

Een schip komt in de sluis niet persé achter of voor zijn voorganger te liggen. Het wordt er door de hulpsluiswachter "ingesorteerd". In dit model wordt de plaats van een schip in de sluiswachtrij aangegeven door de afstand van de achtersteven tot de "achterkant" van de sluis, dus middels een sorteerparameter VOLGNO.

Indien de component BOOT niet current is wordt BOOT in de sluiswachtrij gesorteerd volgens; CALL SORTINQUEUE(BOOT,SLUIS, VOLGNO);

3.2.5. PROCEDURE OUTFOFQUEUE;

CALL OUTFOFQUEUE;

De current component wordt uit de wachtrij gehaald, waarin het zich bevindt.

CALL OUTFOFQUEUE(BOOT);

De component BOOT wordt uit de wachtrij gehaald, waarin het zich bevindt.

3.2.6. PROCEDURE FIRSTOFQUEUE;

BOOT=FIRSTOFQUEUE(SLUIS);

De component die vooraan in de wachtrij SLUIS staat krijgt de naam BOOT..

N.B. Deze component blijft overigens gewoon op zijn plaats in de wachtrij.

3.2.8. PROCEDURE SUCC;

BOOT=SUCC(VAARTUIG);

BOOT is de naam van de opvolger van de component (in dit geval een schip) VAARTUIG.

Indien VAARTUIG geen opvolger heeft, dan is BOOT=NULL.

3.2.9. PROCEDURE PRED;

BOOT=PRED(VAARTUIG);

BOOT is nu de naam van de voorganger van VAARTUIG.

4. Procedures voor het gebruik van histogrammen.

PROSIM geeft de mogelijkheid een vrij uitvoerige histogram-uitvoer te gebruiken. Deze histogrammen kunnen gebruikt worden voor bijvoorbeeld de wachttijden in de wachtrijen maar even- goed voor doorlooptijden of delen hiervan.

4.1. PROCEDURE NEWHIST;

WACHTTIJD=NEWHIST(20,10,5);

De pointer variabele WACHTTIJD wordt geïnitieerd als een histogram bestaande uit 20 klassen met ondergrens 10 minuten en klaswijdte 5 minuten.

4.2. PROCEDURE HISTPUT;

CALL HISTPUT(WACHTTIJD,NOW-QUEUE TIME);

Van de current component wordt de wachttijd bij de sluis genoteerd in de histogram WACHTTIJD. De wachttijd is het verschil van het moment van invaren in de sluis en de tijd waarop het schip in de wachtrij kwam.

4.3. PROCEDURE HISTPRINT;

CALL HISTPRINT(WACHTTIJD);

Histogram WACHTTIJD wordt op de computer uitvoer uitgeprint.

4.4. PROCEDURE HISTCLEAR;

```
CALL HISTCLEAR(WACHTTIJD);
```

Histogram WACHTTIJD wordt "schoongemaakt".

5. Deze opsomming van procedures is beslist niet een volledig overzicht van de procedures van PROSIM en pretendeert ook niet een volledige behandeling te zijn van deze taal. Wil men echter het model bestuderen is het noodzakelijk van bovenstaande procedures goede nota te nemen.

Literatuur:

Delft Progress Report:

Mathematical Engineering, Mathematics and Information Engineering  
1975, blz: 85-102

Discrete Simulatie met behulp van Algol, Fortran, PL/1.

Ir. J.A.G.M. Kerbosch / R.W. Sierenberg

Samson Uitg. Alphen aan de Rijn 1973.

## Hoofdstuk II

### Algemene uiteenzetting van het model.

Het is te onderscheiden in drie onderdelen:

- a. Simulatie van een enkele sluis.
- b. Simulatie van een waterweg met sluizen.
- c. Simulatie van een netwerk van kanalen en sluizen.

#### 1. Simulatie van een enkele sluis

Bij een sluis doen zich twee soorten processen voor n.l. het proces van de schepen en het proces van de hulpsluiswachters. (Een sluis-complex kan uit meerdere sluizen bestaan. Elke afzonderlijke sluis wordt "bediend" door een hulp-sluiswachter.) Elk schip doorloopt bij een sluis het zelfde proces en behoort dan ook tot de klasse van schepen. Een hulpsluiswachter heeft ook zijn eigen proces. Indien er meer dan één hulpsluiswachter werkzaam is bij de sluis omdat de sluis meerdere kolken bevat, volgt iedere hulpsluiswachter hetzelfde proces en behoort daarom tot de klasse van "hulpsluiswachters".

Een derde proces, dat niets met de sluis te maken heeft, maar wel noodzakelijk is om het aankomstenpatroon van schepen bij die sluis te simuleren, is het proces van genereren van schepen. We gaan voorlopig uit van een Poisson-verdeling voor wat deze aankomsten betreft.

Procedure RANDOM zorgt er voor dat een Poisson-verdeelde aankomst tot stand gebracht kan worden. Elke keer, als een schip gegenereerd is, wordt in het proces van genereren van schepen vastgesteld wanneer het volgende schip aankomt en aan welke zijde van de sluis die aankomst zal plaats vinden.

Het model wordt gekarakteriseerd door deze drie processen, die in een willekeurige volgorde in het model zijn opgenomen.

De plaats in het model is niet belangrijk, daar een component zijn proces aanvangt na een label-verwijzing. Men moet er uiteraard voor zorgen dat een component nooit buiten zijn eigen pro-

ces kan komen.

Wel komt het voor dan een interactie tussen een schip en een hulpsluiswachter plaats vindt, die niet terzake doende is voor het proces van het schip.

Dit kan zijn het door de hulpsluiswachter bepalen van de plaats waar een schip in de sluis komt te liggen. Of het bepalen, door de hulpsluiswachter, of een bepaald schip nog wel in de sluis past. Hiermee wordt dat schip zelf niet uit zijn proces gehaald. Het schip wordt in feite slechts door de hulpsluiswachter opgemerkt.

Verder komt het voor dat een hulpsluiswachter beslissingen moet nemen op grond van ervaring. De hulpsluiswachter "bekijkt" van te voren hoe hij de schepen het best in de sluis kan passen. Daar dit model de praktijk tracht te simuleren, moet hieruit een gedragslijn worden bepaald en omgezet worden in een stukje programma.

Dit soort stukjes programma, die nogal een ingewikkelde structuur kunnen hebben en in wezen zich afspelen in het "hoofd" van de hulpsluiswachter worden niet in het proces van de hulpsluiswachter opgenomen, maar in aparte procedures bij het model gevoegd, met een verwijzing vanuit een proces naar zo'n procedure.

Dit is gedaan om het proces overzichtelijk te houden en bovendien geeft het proces nu slechts aan, wat een waarnemer fysisch aanschouwt.

Ondanks dit laatste streven zullen er altijd nog opdrachten in het proces voorkomen, die te maken hebben met uitvoerregistratie en beslissingscriteria.

Het model is uiteraard ook geschikt om een eventueel optimalere kolkindeling enz. te simuleren.

Het hele model wordt voorafgegaan door het zogenaamde hoofdproces.

Dit proces moet gezien worden als de besturing van de gebruiker van het model.

Hij bepaalt hierin hoe de situatie van de sluis is, hoe het aankomstenpatroon van de schepen eruitziet, hoevaak en welke uitvoer hij wenst. Hij kan hierin de processen op gang brengen, doch tenminste 1 proces.

De andere processen kunnen weer door dit ene proces op gang gebracht worden, maar bij gebruik van PROSIM zorgt PROSIM hiervoor.

Het volledige model bestaat uit:

- a. De declaraties.
- b. Het hoofdproces.
- c. Het proces van genereren van schepen.
- d. Het proces van passeren van de sluis.
- e. Het proces van de hulpsluiswachters.
- f. Een aantal procedures.

Een uitgebreide behandeling van het fysisch gebeuren bij een sluis en de modelanalogie treft u aan in de hoofdstukken IV en V.

## 2. Simulatie van een kanaal met sluizen.

Dit model is een uitbreiding van het sluismodel. De schepen doorlopen nu niet één enkele sluis met 1 of meerdere kolken, maar enkele kanaalgedeelten met 1 of meerdere sluizen.

Het proces van het doorlopen van een sluis blijft in deze simulatie hetzelfde als in het sluismodel, maar daar een schip meerdere sluizen ontmoet, moet het dit proces zo vaak herhalen als er sluizen te passeren zijn.

Het doorvaren van een kanaalgedeelte levert in het algemeen geen problemen op, hoewel ook hier een beperkte studie aan gewijd is, terwijl ook enkele onderzoek mogelijkheden hiervoor zijn opgenomen in het model.

Het doorvaren van een kanaalgedeelte betekent voor de simulatie niets anders dan dat het proces van het doorvaren van een kanaal onderbroken wordt gedurende de tijd:

$$\text{tijd} = \frac{\text{aantal kilometers kanaalpartij}}{\text{snellheid v.h. schip}}$$

Het aantal kilometers is een vast gegeven, de snelheid van een



schip hangt van een aantal factoren af, zoals doorvaart profiel en belastingsgraad. Schepen kunnen elkaar passeren én inhalen in een kanaalpand en komen dan ook in een andere volgorde bij de volgende sluis aan dan dat zij vertrokken zijn.

Zodra een sluis gepasseerd is, is het vanzelfsprekend dat we niet meer met een Poisson-verdeeld aankomstenpatroon behoeven te maken te hebben bij de eerstvolgende sluis. Over het algemeen zal hier sprake zijn van een gedoseerd aanbod van schepen. Het is dan ook van belang de schepen te genereren in het waterweg gedeelte, waar we vrij zeker zijn van een Poisson-verdeeld aanbod.

Nu is het niet nodig om het hele proces van een schip voor een tweede en eventueel volgende sluizen nog eens in het model op te nemen, daar dit proces exact hetzelfde verloopt.

Met een test en een label kan het schip bij het begin van zijn proces vervolgen. Zie hiervoor het bijgevoegd model.

In het programma moet duidelijk zijn weergegeven bij welke sluis een schip zich bevindt.

Dit is mogelijk door niet over een wachtrij te spreken, maar over een wachtrij(1) en een wachtrij(2) voor de beide zijden van de sluis en van een wachtrij(1,1) en wachtrij(1,2) als er meerdere sluizen in het systeem aanwezig zijn.

Indien een schip alle sluizen gepasseerd is geeft een test dit aan en wordt het schip uit het systeem geschreven. Het computergeheugen, dat het schip in beslag genomen had wordt weer vrijgegeven om een volgend schip in te noteren.

Een hulpsluiswachter moet weten bij welke sluis hij werkt en welke kolk hij moet bedienen. Dit krijgt hij op als hij in het systeem gebracht wordt.

De generator van schepen genereert nu geen schepen bij de sluis en geeft een schip ook geen bepaalde richting meer op.

Bij een generator komen schepen in het systeem en de generator stuurt de schepen slechts één richting uit.

Van de generator krijgt het schip o.a. op welke sluizen het moet passeren, aan welke kant het een sluis nadert: van de bovenstroomse richting of de benedenstroomse richting.

Bij een sluis kan dan ook overeenkomstig de gegenereerde richting

het schip in een wachtrij geplaatst worden.

Hoe de juiste werking van het een en ander is, is te vinden in de hoofdstukken VI en VII.

### 3. Simulatie van een netwerk van kanalen met sluizen.

Met het model is het mogelijk een netwerk van kanalen met sluizen te simuleren. Om niet een al te groot computergeheugengebied te reserveren is dit netwerk begrenst tot 5 generatoren, 25 sluizen en 25 routes waarvan van elke generator 5 routes kunnen lopen.

Het aantal schepen en het aantal hulpsluiswachters is onbegrensd evenals het aantal te draaien runs en de simulatietijd, waarbij opgemerkt dient te worden, dat wel voldoende rekentijd aangevraagd moet worden.

Mochten bovengenoemde begrenzungen tot moeilijkheden lijden, dan is het mogelijk op een eenvoudige manier deze grenzen te verruimen in het model.

Dit netwerk werkt op dezelfde wijze als het kanaalmodel.

De schepen krijgen nu bovendien op via welke route zij dienen te varen en welke sluizen zij daarin ontmoeten en van welke kant zij die naderen. Via welke route zij dienen te varen wordt weer willekeurig gekozen uit een bekend veronderstelde verdeling van de routekeuze bij een generator. Voor zover deze verdeling niet bekend is kan gebruik gemaakt worden van een in het hoofdontwerp behandelde optimalisatie procedure, dat aan dit model gekoppeld kan worden. Na iedere simulatierun worden de verhoudingen van de aantallen schepen over de verschillende routes vastgesteld aan de hand van een kosten-minimalisatie en een winst-maximalisatie. De startwaarde van de verdeling mag dan willekeurig gekozen worden.

Men dient overigens wel voorzichtig te zijn met deze methode en zich voldoende ervan te vergewissen dat deze optimalisatie ook inderdaad convergeert doordat de simulatietijd voldoende groot gekozen is.

Het netwerk zal toegelicht worden in de hoofdstukken VI en VII.

Tenslotte moet nog opgemerkt worden, dat overal waar de werkelijkheid geschematiseerd is een fout geïntroduceerd wordt ten opzichte van die werkelijkheid.

Het is noodzakelijk op de hoogte te zijn van de grootte van deze fouten en wat voor invloed zij hebben in het model.

De drie hiervoor besproken modellen zijn bijeen gebracht in één model, waarbij men wel al de drie aspecten afzonderlijk kan simuleren. Zie hiervoor de handleiding voor het gebruik van het model, dat als bijlage bij dit ontwerp is gevoegd.

Ook is er voor gezorgd dat vanuit het model verwezen kan worden naar de procedure optimalisatie.

### Hoofdstuk III

#### De scheepsklassen en het genereren van schepen.

1. In Nederland worden de scheepsklassen onder verdeeld in 8 laadvermogensklassen. (Zie bijlage C )

Voor wat het inpassen en het doorvaren van sluisen betreft, zijn van deze klassen de volgende punten van belang:

gemiddelde lengte

„ breedte

„ tonnage

„ snelheid.

Het model staat open voor maximaal 10 laadvermogensklassen (scheepstypen):

1 VAARTUIG(FLOAT),  
2 LENGTES FLOAT,  
2 BREEDTES FLOAT,  
2 TONNAGES FLOAT,  
2 SNELHEDEN FLOAT.

Wat de relaties hiervan zijn ten aanzien van het sluisgebeuren komt ter sprake in hoofdstuk IV.

In werkelijkheid hebben we niet te maken met standaardschepen maar met individuele schepen die tot een bepaalde standaardklasse gerekend kunnen worden, omdat zij binnen het definitie gebied liggen van zo'n klasse.

Binnen die klassen zijn de schepen voor elke karaktereigenschap op een bepaalde manier verdeeld. Voor middenklassen zullen zij min of meer uniform verdeeld zijn, terwijl dit voor de lagere klassen in mindere mate het geval zal zijn. Voor de standaardtypen 7 en 8 zal dit geenszins het geval zijn daar deze klassen gereserveerd zijn voor duweenheden, waarvan de duwbakken voldoen aan standaard afmetingen.

Daar hier ten aanzien van het model nog geen studie aan gewijd is, wordt in het model uitgegaan van de gemiddelde waarden, hoewel hierdoor een relatief grote fout geïntroduceerd kan worden.

Zodra deze verdelingen bekend zijn, kunnen zij zonder veel problemen in het model verwerkt worden.

Het komt, zoals in bovenstaand geval, regelmatig voor, dat facetten van de werkelijkheid nog niet in het model verwerkt zijn. Ten aanzien van het kanaalgebeuren is inzicht vereist in de snelheden, die de verschillende schepen of scheepstypen aanhouden. Hierover is door de Rijkswaterstaat, Directie Benedenrivieren, een studie aangewijd, die behandeld wordt in haar Nota 70.12.4. Zij handelt over Vaarsnelheden op scheepvaartwegen bij een lage verkeersintensiteit.

Dit laatste moet als eis gesteld worden, daar bij een hoge verkeersintensiteit congestie ontstaat, doordat de snelheid van de schepen mede afhankelijk wordt van de verrichtingen van andere schepen. In het model is het mogelijk een procedure aan te roepen, die een inzicht geeft in het aantal ontmoetingen en de plaats waar dit gebeurt in een s-t diagram, wat uitgeplot wordt na het aanroepen van deze procedure.

Wat er gebeurd bij congestie in het kanaal wordt in dit model verder niet behandeld en is ook niet bestudeerd.

Het is niet de bedoeling om in dit verslag uitgebreid in te gaan op de benaderingswijze omtrend snelheden van schepen zoals die in bovengenoemde nota is gedaan. In het model is wel gebruik gemaakt van de resultaten van dit onderzoek.

De snelheid van een schip is o.a. afhankelijk van de klasse van het schip en zijn beladingsgraad.

Opgemerkt dient te worden dat gezien de vaareigenschappen (met name t.a.v. de snelheid) van het scheepstype motorspits, deze niet behandeld kan worden als de scheepsklasse waartoe de motorspits gezien zijn overige attributen.

De bijlagen geven verder aan, hoe deze snelheden in de verschillende situaties zijn.

Dit in het kort over de snelheid van schepen in kanalen.



## 2. Attributen van schepen

1 SCHIPPARL BASED(SCHIP),  
2 LENGTE FLOAT,  
2 BREEDTE FLOAT,  
2 LUSSCHIP BIT(1),  
2 VOLGNO FLOAT,  
2 RICHTING FIXED BIN,  
2 INVAARTYD FLOAT,  
2 UITVAARTYD FLOAT,  
2 LUSTYD FLOAT,  
2 AANKOMSTYD FLOAT,  
2 GELADEN FIXED BIN,  
2 KLASSE FIXED BIN,  
2 SNELHEID FLOAT,  
2 TONNAGE FLOAT,  
2 PRIOR FIXED BIN,  
2 VOORRANG FLOAT,  
2 VAN FIXED BIN,  
2 VIA FIXED BIN,  
2 ROUTE(25) FIXED BIN,  
2 WYZER FIXED BIN,  
2 HULPSLUISW POINTER,  
2 SLUISNO FIXED BIN,

Hierna volgt een verklaring van de attributen.

### a. Lengte:

Voor de lengte van een schip wordt gerekend met de gemiddelde lengte van het scheepstype waartoe het schip behoort.

### b. Breedte:

Voor de breedte van een schip wordt gerekend met de gemiddelde breedte van het scheepstype, waartoe het schip behoort.

### c. Lusschip:

Het eerste schip, dat binnenvaart heeft geen invaarvolgtijd. Van belang voor de simulatie is de tijd die verstrijkt tussen het passeren van de achterstevan van het laatste schip dat uitvaart. Dit wordt de lustijd genoemd. We spreken gemakshalve van het lusschip. Lusschip is een boolean-variabele, die de waarde true of false krijgt.

### d. Volgno:

De volgorde van uitvaren uit een kolk hangt in sterke mate af van de plaats, die de schepen in de kolk innemen.

Deze is over het algemeen anders dan de volgorde van binnenvaren.

Een benadering hiervan is:

Het schip dat het verst met zijn achterstevan naar de achter-



kant van de sluis ligt, vaart het eerste uit. Dit proces wordt herhaalt, totdat er zich geen schepen meer in de sluis bevinden.

Deze volgorde geeft niet precies de werkelijkheid weer, maar waarnemingen, die verricht zijn bij de Volkeraksluizen (zie bijlage G ) geven een niet al te grote fout, vooral als men daarbij nog de totale uitvaartijd in beschouwing neemt. Volgno geeft de afstand van de achtersteven van de achtersteven van het schip tot de achterste sluisdeuren aan, terwijl om programmatische redenen deze afstand negatief genomen wordt.

e. Richting:

De richting van het schip geeft aan of het schip een sluis van de bovenstroomse zijde (richting=1) of van de benedenstroomse zijde (richting=2) nadert.

Bij elke sluis, die een schip moet passeren wordt opnieuw de richting vastgesteld.

f. Invaartijd:

Dit is de invaarvolgtijd van het schip, dat ook bij elke sluis opnieuw bepaald moet worden. Deze invaarvolgtijd wordt afgeleid van de gegevens, zoals die vermeld zijn in Nota 70-12-7 van de nota Vaarwegen van Rijkswaterstaat (bijlage ). Daar de invaarvolgtijd afhangt van een aantal attributen van het schip én van de sluis is het vrij complex, deze te bepalen. Daarom wordt dit in een aparte procedure gedaan in verband met de overzichtelijkheid van het proces van passeren van een sluis.

De invaarvolgtijd hangt af van:

- a. Sluisbreedte
- b. De waterhoogte ter plaatse van de sluisdeurdrempel
- c. Kolkvoorhavengetal
- d. Klasse van het schip
- e. Beladingsgraad van het schip.

De relaties, die deze variabelen met de invaarvolgtijd hebben, zijn weergegeven als lijnen in de bijlagen .

Om deze gegevens op te nemen in het model, zijn zij als polynomen bepaald. De constanten van deze polynomen zijn

als permanente dataset bij het model gevoegd. Op dit moment (1975-1976) is deze dataset opgenomen in het geheugen van de IBM-360 van de Technische Hogeschool te Delft.

Bij het gebruik van het model moet dan ook naar deze dataset verwezen worden middels een stuurkaart.

```
(//SYSIN1 DD DSN=WWWB.DATAS,DISP=SHR)
```

Gezien de nogal kleine schaal van de lusafstand, die uitgezet is langs de ordinaat-as, is ten behoeve van de nauwkeurigheid deze schaal vergroot met de factor 100, wat tot uitdrukking komt in de procedure INVAART (zie bladz. 75 )

g. Uitvaarttijd:

Ook hier is sprake van een uitvaarvolgtijd.

Hetzelfde als voor de invaartijd geldt voor de uitvaartijd (zie bijlagen E en bladz. 76 ).

h. Lustijd:

De lustijd is de tijd die omschreven staat bij de explicatie van de attribuut "lusschip".

Deze lustijd is afhankelijk van:

- a. klasse van het schip
- b. beladingsgraad van het schip
- c. breedte van het schip
- d. sluisbreedte
- e. kolkvoorhavengetal
- f. lusafstand.

Verder is een soortgelijke procedure gevolgt als voor de in- en uitvaarvolgtijden.

(Zie bijlagen E en bladz. 77 ).

i. Aankomsttijd:

Deze tijd geeft aan op welk tijdstip een schip bij een sluis is aangekomen. Deze tijd wordt genoteerd t.b.v. het vaststellen van de wachttijd van het schip bij een sluis.

j. Geladen:

Dit attribuut geeft de beladingsgraad van het schip aan.

Deze kan lopen van 0 tot 1.

0 voor ongeladen schepen,

1 voor volledig geladen schepen.

k. Klasse:

Klasse geeft de laadvermogensklasse van het schip aan.

l. Snelheid:

De snelheid wordt bepaald overeenkomstig het gestelde aan het begin van dit hoofdstuk.

m. Tonnage:

Het tonnage is het laadvermogen van het schip.

Hier wordt weer het gemiddelde tonnage van het scheepstype waartoe het schip behoort, aangehouden.

n. Prior:

De prioriteit van een schip:

Het kolkindelen vergt inzicht van de hulpsluiswachter. Bij waarnemingen bij de Volkeraksluizen is mij gebleken, dat het kolkindelen het best gesimuleerd kan worden door de grotere schepen eerst in te laten varen, waarna de kleinere schepen de "gaten" gaan vullen.

Hoe het kolkindelen en de volgorde van binnenvaren gaat, wordt in de volgende hoofdstukken besproken. Voor het schip betekent het, dat het een prioriteit krijgt als het eigenlijk al "aan de beurt" was om binnen te varen, maar dit niet kan, doordat andere schepen, gezien hun grootte, voorrang krijgen. In dat geval krijgt het schip de prioriteit 1, voor de tweede keer prioriteit 2, enz.

o. Voorrang:

Het binnenvaren in de kolk gebeurt dus met de voorrangsregel: eerst de grotere schepen, dan de kleinere. Dit komt tot uitdrukking in de "voorrang", die een schip heeft.

Heeft een kleiner schip de eerste maal "de boot" gemist, dan wordt zijn voorrang veranderd in de grootst aanwezige KLASSE pus zijn prioriteit om ervoor te zorgen dat dit schip de volgende schutting een zeer redelijke kans krijgt binnen te mogen varen.

p. Van:

Van geeft aan waar een schip vandaan komt, dat wil zeggen door welke generator het schip in het systeem is gebracht.

q. Via:

Via geeft aan via welke route vanaf zijn generator een schip

gaat varen.

r. Route(25):

Route(25) is een array bestaande uit een rij van 25 elementen. Dit getal geeft aan, dat een schip op zijn weg maximaal 25 sluizen kan tegenkomen.

Elk element geeft aan het sluisnummer, dat het schip in de volgorde, waarin dit nummer in de rij is geplaatst, moet doorlopen.

s. Wijzer:

De wijzer geeft aan bij welke sluis het schip zich bevindt of welke sluis het nadert.

B.v. heeft een schip het laatst de vijfde sluis gepasseerd, dan heeft de wijzer de waarde 6.

t. Hulpsluisw:

Een schip, dat in een wachtrij bij een sluis ligt zal eens de opdracht krijgen een kolk van die sluis binnen te varen. Het krijgt die opdracht van een hulpsluiswachter, die de betreffende kolk indeelt.

Het schip heeft voor de tijd dat het verder nog bij die sluis vertoeft alleen nog maar te maken met die hulpsluiswachter, daarom wordt die component een attribuut van de component schip, terwijl die component overigens zelf zelfstandig blijft. Daar die hulpsluiswachter niet een attribuut is, dat door een getal gerepresenteerd kan worden, wordt het gedeclareerd als een POINTER- variabele.

u. Sluisno:

Sluisno geeft aan bij welke sluis een schip zich bevindt evenals ROUTE(WIJZER), met dit verschil, dat ROUTE(WIJZER) negatief kan zijn als een schip een sluis van de benedenstroomse zijde benadert.

SLUISNO=ABS(ROUTE(WIJZER)).

Dit zijn de attributen, die een schip meekrijgt als het het systeem in wordt gestuurd. Bij aankomst in het systeem behoeven al deze attributen niet een waarde te hebben. Die kunnen zij krijgen van andere componenten of worden gedefinieerd vanuit hun eigen proces,

Bij deze lijst van attributen voegt PROSIM een aantal toe, die in hoofdstuk I reeds behandeld zijn. Een aantal attributen wordt dus meegegeven als een schip zich in het systeem begeeft.

Dit gebeurt in de zogenaamde generator van schepen.

De generator van schepen bepaalt naast de attributen van een schip eveneens de aankomst van een volgend schip bij die generator en van de klasse waartoe voorgaande schip ook behoorde.

In de volgende paragraaf worden dan ook drie onderwerpen behandeld:

- a. Aankomstenpatronen van schepen bij generatoren.
- b. Het starten van een generator.
- c. De werking van een generator en het toekennen van waarden aan attributen door de generatoren.

### 3. Generator van schepen.

3.1. Op een plaats waar schepen gegenereerd worden moet bekend zijn hoe het doorvaart- of aankomstpatroon is van schepen bij die generator.

Dit kan men op verschillende manieren inbrengen in het model. Noodzakelijk is het om óf een lange reeks waarnemingen te doen en aan de hand hiervan verdelingen samen te stellen, (eventueel in histogramvorm) óf over goede prognoses voor het toekomstige scheepvaartverkeer te beschikken.

De meest gebruikelijke vorm van verdelingen van schepen is de negatief exponentiële verdeling van aankomsten per scheeps-klasse per generator, de zo geheten POISSON-verdeling.

Hierover is reeds het een en ander onderzocht en gepubliceerd.

Bij dit model is gebruik gemaakt van publicaties hierover van prof. dr. ing. W. Leutzbach.

(Technische Hochschule Fredericiana te Karlsruhe, Instituut voor het verkeerswezen).

De kansberekening uit de POISSON-verdeling is als volgt:

$$P(x) = e^{-m} \cdot \frac{m^x}{x!}$$

m is het gemiddeld aantal vaartuigen, dat een beschouwde



raai per tijdseenheid passeert.

$P(x)$  is de kans dat voor  $x=1,2,\dots,n$  vaartuigen in het betreffende tijdsinterval die raai passeren.

Hierbij moet de vraag gesteld worden hoe groot de zekerheid is, dat de verdeling ook voldoet aan de waarnemingen.

Hierbij kan gebruik gemaakt worden van de  $\chi^2$  - toets of de berekeningen van het bereik van de natuurlijke onzekerheid. In vele gevallen zal de POISSON-verdeling gestoord zijn, daar waar zeer lange tussenaankomsten genoteerd zouden moeten worden. Hierbij is het noodzakelijk deze verstoring bij het bepalen van een nieuwe tussenaankomsttijd in rekening te brengen.

De POISSON-verdeling gaat dus uit van een dichtheid  $m$ ;  $m$ =gemiddeld aantal schepen per tijdseenheid (minuut).  
 $\lambda = \frac{1}{m}$ , waarbij  $\lambda$  de gemiddelde tussenaankomsttijd van schepen is.

Een willekeurige trekking uit de POISSON-verdeling met gemiddelde tussenaankomsttijd  $\lambda$  gaat als volgt:

Tussenaankomsttijd =  $-\lambda \cdot \text{LOG}(\text{RANDOM})$ , waarin RANDOM een willekeurig getal is tussen 0 en 1 en middels een procedure RANDOM door de computer wordt ingebracht.

Daar bij een raai verschillende sloopstypes passeren met verschillende tussenaankomsttijden moet bij elke plaats, waar schepen in het systeem komen evenveel generatoren als er sloopklassen zijn, aangebracht worden, elk met hun eigen specifieke tussenaankomsttijd!

### 3.2. Het starten van een generator.

Alvorens een generator kan gaan werken, moeten de gegevens van de schepen, wier sloopklasse deze generator vertegenwoordigt in het systeem gebracht worden.

```
1 GENERPAR BASED(GENER),
2 VERHPAR(5) FLOAT,
2 VERDELP(5) FLOAT,
2 GEM(10,2) FLOAT,
2 PR FIXED BIN,
2 KLASSEPAR FIXED BIN,
2 VANAF FIXED BIN,
2 GELADENPAR FLOAT,
```



De attribuut-lijst van een generator bestaat uit:

a. Verhpar(5):

Vanaf een generator kunnen schepen maximaal 5 routes kiezen. Van elke route moet bekend zijn, hoeveel schepen die route kiezen. Indien er minder dan 5 routes zijn, zijn de verdere waarden van dit array 0.

Verhpar(1) = 0.15 betekent dus dat 15% van de schepen route 1 kiezen vanaf de generator.

Verhpar is cumulatief genomen in verband met de aselec-te RANDOM-trekking tussen de waarden 0 en 1.

b. Verdelp(5):

Verdelp is hetzelfde als Verhpar, doch met dit verschil, dat Verdelp niet cumulatief is.

c. Gem(10,2):

Deze tweedimensionale array representeert de gemiddelde aankomsttijden  $\lambda$ .

Daar deze  $\lambda$  variabele kan zijn in de tijd stelt Gem een histogram voor van gemiddelde aankomsttijden. Maximaal zijn er 10 klassen mogelijk, waarbij de gebruiker zelf deze klassen indeelt.

Per klasse wordt aangegeven de bovengrens van de klasse en de gemiddelde tussenaankomsttijd binnen die klasse.

Het is eveneens mogelijk slechts een gedeelte van dit array te gebruiken.

d. PR:

PR stelt een variabele voor, die aanwijst welke klasse uit Gem gebruikt moet worden, gezien de kloktijd van het systeem.

De kloktijd van het systeem is de eventtime van de CURRENT component.

e. Klasse par:

Klassepar geeft aan voor welk scheepstype de generator werkt.

f. Vanaf:

Vanaf geeft aan waar de generator schepen in het systeem stuurt.

g. Geladenpar:

Geladenpar geeft aan hoe de verhouding is van geladen

schepen ten opzichte van het totaal aantal schepen.

Het inbrengen van een generator in het systeem gaat als volgt:

```
121 1 0 DO J=1 TO TOTGENER;
122 1 1 DO I=1 TO TOTKLASSEN;
123 1 2 ALLOCATE GENERPAR;
124 1 2 VERHPAR=0;VERDELP=0;PR=1;GEN(J,I)=GENER;
128 1 2 VANAF=J;KLASSEPAR=I;GEM=1;
131 1 2 PUT SKIP(3);
132 1 2 PUT SKIP EDIT('AANKOMSTEN BY GENERATOR',J,' VAN KLASSE',I)
      (A,F(2),A,F(2));

133 1 2 W=0;
134 1 2 DO K=1 BY 1 WHILE(W<0.999);
135 1 3 GET LIST(VERDELP(K));
136 1 3 W=W+VERDELP(K);
137 1 3 VERHPAR(K)=W;
138 1 3 END;
139 1 2 TABEL(1,J,I,*)=VERDELP;
140 1 2 GET LIST(GELADENPAR);
141 1 2 GET LIST(GEM(1,*));
142 1 2 DO K=1 BY 1 WHILE(GEM(K,1)<(SIMULATIETYD));
143 1 3 GET LIST(GEM(K+1,*));
144 1 3 END;
145 1 2 DO M=1 TO K;
146 1 3 PUT SKIP EDIT('GEMIDDELDE AANKOMSTTYD TOT',GEM(M,1))(A,F(6));
147 1 3 PUT EDIT(':',GEM(M,2))(A,F(8,2));
148 1 3 END;
149 1 2 PUT SKIP EDIT('GELADEN SCHEPEN / AANTAL SCHEPEN:',GELADENPAR)
      (A,F(8,2));PUT SKIP(2);
151 1 2 PUT SKIP DATA(VERDELP,VERHPAR);
152 1 2 CALL ACTIVATE(NEW(GENER),GENEREER);
153 1 2 END;
154 1 1 END;
```

- a. Er worden zóveel generatoren in het systeem gebracht als er plaatsen zijn waar schepen binnenkomen.
- b. Elke generator is onderverdeeld in zóveel subgeneratoren als er scheepstypen zijn.
- c. Reserveer per subgenerator geheugenruimte in de computer.
- d. Gem(1,1) is de naam van de eerste generator z'n subgenerator, die de laagst in het systeem aanwezige scheepstype genereert.
- e. Om programmatische redenen worden alle elementen van Verhpar en Verdelp 0 gemaakt, terwijl PR=1 aangeeft, dat aan het begin, dus bij de eerste klasse van het gemiddelde tussenaankomsten-histogram begonnen moet worden.
- f. Daar de verdeling niet voor 5 routes behoeven te gelden, wordt alleen die verdeling ingelezen, die voor de gebruiker van belang is (zie voor gebruik de bijgevoegde

handleiding)

Deze gegevens omtrent het verdelen over de verschillende routes worden verzameld in een array: TABEL.

h. In dit programmadeel worden de waarden van het histogram Gem (gemiddelde tussenaankomsten verdeeld over de tijd) ingelezen.

i. Er was geheugenruimte gereserveerd voor een generator. De attribootlijst is van deze generator ingevuld, maar deze nieuwe generator is nog niet aan het systeem bekend gemaakt.

Immers elke nieuwe component moet aan het systeem bekend gemaakt worden en een activeringspunt hebben.

Dit gebeurt met de volgende opdracht:

```
CALL ACTIVATE(NEW(GENER),GENEREER).
```

Deze nieuwe generator vangt zijn werk aan bij de label GENEREER.

Daar de computer maar 1 proces tegelijkertijd kan volgen, wordt het proces van deze nieuwe generator uitgesteld totdat het proces MAIN, waar de computer nu mee bezig is, wordt gepassiveerd of opgehouden.

Daar er meer dan 1 schepengenerator aan het werk wordt gezet op het tijdstip 0 begint de generator, die het eerst aan het systeem bekend gemaakt is, ook het eerst nadat het proces MAIN opgehouden is, ook het eerst te werken. Daarna de tweede, tenminste, als niet door een component de opdracht is gegeven een andere component voorrang te geven voor één van deze componenten. Hetgeen in het model wel het geval is, maar dit doet niets af aan de volgorde waarin de generatoren zullen beginnen te werken.

### 3.3. De werking van een generator:

```
267 1 0 /*HET PROCES VAN GENEREREN VAN SCHEPEN*/
      GENEREER:
268 1 0 ALLOCATE SCHIPPARL;
269 1 0 RICHTING=1;
270 1 0 PRIOR=0;
271 1 0 GELADEN=1;
      IF RANDOM(IVB)>GELADENPAR THEN
272 1 0 GELADEN=2;
      KLASSE=KLASSEPAR;
273 1 0 LENGTE=LENGTES(KLASSE);
274 1 0 BREEDTE=BREEDTES(KLASSE);
275 1 0 TONNAGE=TONNAGES(KLASSE);
276 1 0 SNELHEID=SNELHEDEN(KLASSE);
277 1 0 VAN=VANAF;
278 1 0 D=RANDOM(IVA);
279 1 0 DO K=1 TO 5;
      IF DK<VERHPAR(K) THEN
280 1 1 DO;ROUTE=ROUTEPAR(VAN,K,*);VIA=K;
281 1 1 GOTO ROUTEBEPAALD;
282 1 2 END;
283 1 2 END;
284 1 1 ROUTEBEPAALD:
285 1 0 WYZER=1;
286 1 0 IF ROUTE(WYZER)<0 THEN RICHTING=2;
287 1 0 SLUISNC=ABS(ROUTE(WYZER));
288 1 0 LUSSCHIP='0'B;
289 1 0 CALL ACTIVATE(NEW(SCHIP),AANKOMST);
290 1 0 DO WHILE (NOW>GEM(PR,1));PR=PR+1;END;
291 1 0 CALL HOLD(-GEM(PR,2)*LOG(RANDOM(IV)));
292 1 0 GOTO GENEREER;
293 1 0 /*EINDE VAN HET PROCES VAN GENEREREN VAN SCHEPEN
```

Daar waar de opdrachten uit het model voor zich spreken is verder geen verklaring gegeven.

Een generator vangt zijn proces aan bij de label GENEREER.

- a. De generator reserveert geheugenruimte voor een schip.
- b. Voorlopig wordt de Richting van een schip op 1 gesteld en geen enkel schip bezit van nature een prioriteit.
- c. Het schip heeft de beladingsgraad 1 (geladen). Als een willekeurig getal tussen 0 en 1 (random) een waarde aangeeft, die hoger is dan de waarde van het attribuut Geladenpar van de generator dan is de beladingsgraad 2 (ongeladen).

In het verdere vervolg van dit verslag zullen we spreken van RANDOM inplaats van een willekeurig getal tussen 0 en 1.

- d. D is een Random-getal.

De verdeling over de verschillende routes kan als volgt

in beeld worden gebracht:

Stel :	Verdelp(1) = 0.15	Verhpar = 0.15
	Verdelp(2) = 0.15	Verhpar = 0.30
	Verdelp(3) = 0.60	Verhpar = 0.90
	Verdelp(4) = 0.10	Verhpar = 1.00
	Verdelp(5) = - -	Verhpar = - -

Dit betekent dat er vier routes zijn, die vanaf een generator gevolgt kunnen worden.

15% van de schepen volgt route 1

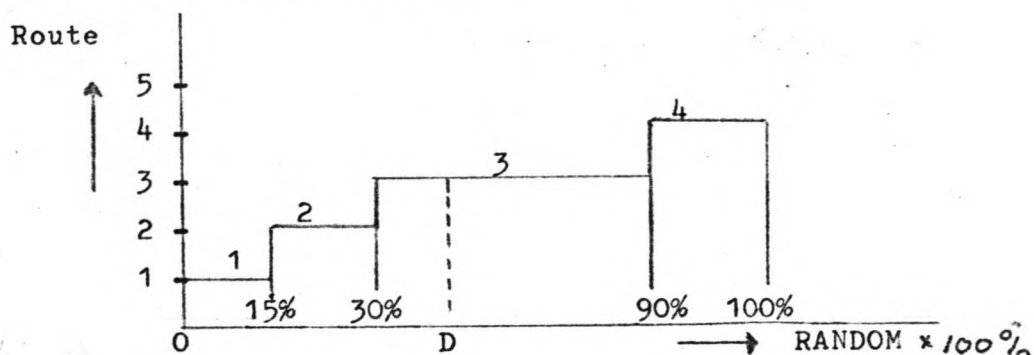
15% van de schepen volgt route 2

60% van de schepen volgt route 3

10% van de schepen volgt route 4

Route 5 is óf niet aanwezig óf er maken geen schepen gebruik van deze route.

Cumulatief uitgezet (Verhpar):



Als  $D=0.40$  zal het schip via route 3 gaan varen.

De computer moet steeds weer alle klassen van het histogram afzoeken om vast te stellen binnen welke grenzen  $D$  zich bevindt, direct daarna vorst vastgesteld welke sluizen het schip in die route zal ontmoeten:

$ROUTEPAR(van, K, *)$  is een eendimensionaal array van sluizen, die van de generator, waar het schip binenkomt en die route K volgt, gaat in de volgorde van passeren. Welnu het schip krijgt die route als zijn te varen route mee:  $ROUTE=ROUTEPAR(van, K, *)$

e. Het schip komt het eerst bij zijn eerst binnen te varen

sluis aan: WIJZER=1.

Zoals eerder besproken is, kan een sluisnummer binnen het array ROUTE negatief genoteerd zijn.

Het schip komt dan van de benedenstroomse richting of te wel van de RICHTING 2.

Het werkelijke sluisnummer: sluisno is natuurlijk een positieve waarde:

SLUISNO=ABS(ROUTE(WIJZER)).

- f. In het proces van een schip wordt getoetst of het het eerste schip is dat in de kolk vaart.

Over het algemeen zal dit niet het geval zijn, dus is SCHIP → LUSSCHIP= FALSE ('0'B).

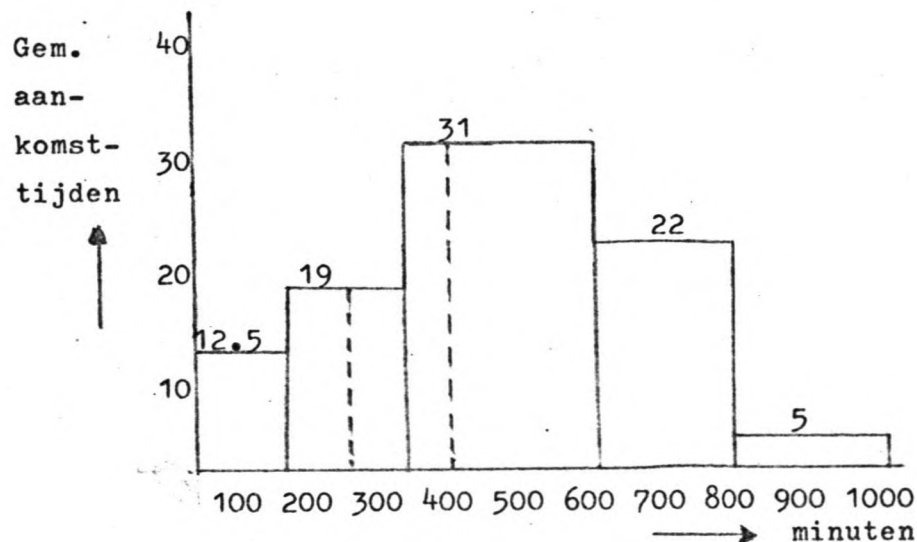
Indien het schip wel het eerste schip is dat binnenvaart, wordt het attribuut LUSSCHIP door de hulpsluiswachter veranderd in de waarde TRUE ('1'B).

- g. Het schip wordt bekend gemaakt aan het systeem en geactiveerd bij de label AANKOMST.

- h. Bepaald wordt eerst wat op dit moment de tijd is.

Mocht deze kloktijd groter zijn dan de bovengrens van de gemiddelde aankomstenverdelingsklasse, waaruit het vorige schip gegenereerd is, dan wordt opnieuw vastgesteld middels het bepalen van de bovengrens, welke histogramklasse nu genomen moet worden en welke gemiddelde aankomsttijd hier bijhoort.

Stel de gemiddelde aankomstenverdeling heeft het volgende verloop:





Een schip, dat op tijdstip 270 minuten gegenereerd is, kwam uit een POISSON-verdeling met gemiddelde aankomsttijd: 19 minuten.

De kloktijd geeft aan dat het volgende schip gegenereerd is op tijdstip 400 minuten.

De gemiddelde aankomsttijd is inmiddels veranderd en van het aankomsttijdstip van het volgende schip wordt dan ook met deze veranderde gemiddelde aankomsttijd gerekend.

Dit gebeurt op de in par. 3.1. besproken wijze.

De generator is als component verzet naar een later tijdstip en hervat dan zijn proces weer bij de label GENEREER.

Zo werken al de in het systeem gebrachte subgeneratoren onafhankelijk van elkaar en zonder elkaar te beïnvloeden.

In eerste instantie lijkt dit wat vreemd, omdat zij allen op een zelfde programma betrekking hebben.

Maar als men een vergelijking trekt met b.v. een telefoonsel en mensen die er gebruik van maken, dan stelt de telefoonsel het stukje programma voor, dat gebruikt wordt door de componenten: mensen.

Zodra de telefoneerder afgesproken heeft wanneer hij weer belt, verlaat hij de cel (het stukje programma) en een andere gebruiker gaat telefoneren. Deze maakt nu gebruik van de cel (het stukje programma).

Deze vergelijking is niet volledig als men de andere twee in het model beschreven processen beschouwd. Aan het eind van het volgende hoofdstuk zal hier aandacht aan besteed worden.

In hoofdstuk VI is het stroomdiagram van dit proces getekend.

## Hoofdstuk IV

### Het fysisch systeem bij een sluis.

#### 1. De sluis als object.

De primaire functie van een schutsluis is het "omschutten" van schepen, die zich op een waterweg bevinden met een andere waterstand dan de waterstand van de waterweg, waarin zij hun route willen vervolgen.

Was op die plaats geen sluis aanwezig of indien men de deuren van een sluis aan beide zijden opent, zou het water van "hoog" naar "laag" stromen, vandaar dat we spreken van het bovenstrooms en het benedenstrooms naderen van een sluis. Bedenk hierbij evenwel dat er steeds stroming gaande is, doordat de schutsluis een hoeveelheid water met de schepen meehevelt. Het volume van wat wordt toegevoegd aan het kanaal benedenstrooms is groot:

Opp.kolk x verschil in waterniveau aan beide zijden van de sluis.

Deze inhoud bestaat uit water en de inhoud van het door de schepen verplaatste water.

De gemiddelde stroming in het kanaal is dan:

$$\frac{\text{Opp.kolk x waterniveauverschil (m}^3\text{)}}{\text{tijd v.e. schutcyclus (sec)}} \times \frac{1}{\text{natte doorsnede v.h. kanaal (m}^2\text{)}} \cdot \text{m/sec}$$

Hierbij moet men echter bedenken, dat dit watersurplus niet constant in de tijd wordt toegevoegd, zodat er lange golf verschijnselen in beide kanaalgedeeltes optreden.

Een sluis bestaat uit 1 of meerdere kolken.

De schutkolken werken doorgaans onafhankelijk van elkaar.

De organisatie bij een sluis ligt in handen van een sluiswachter, die, bij grotere sluisen, één of meer hulpsluiswachters in dienst heeft. De sluiswachter treedt onder andere

op als cöördinator van het sluispersoneel en bemoeit zich in het algemeen niet met het kolkindelen van de sluis en de bediening van de kolk.

Dit geschiedt door de hulpsluiswachter, die aan de kant van de kolk, per kolk een hulpsluiswachter, het kolkindelen verricht in samenwerking met bedienend personeel.

Hij noteert de binnenkomsten van schepen, evenals de naam, plaats van herkomst, geladen of ongeladen en enkele andere attributen van schepen voor elk schip afzonderlijk.

De vorm van een sluis is meestal rechthoekig, terwijl de doorvaartbreedte van de sluisopening gelijk is aan de breedte van de kolk, tenminste bij alle moderne sluisen. De overgang van een kanaal in een sluis is bij moderne sluisen zeer gestroomlijnd (de hellingen naar de versmalling zijn 1 : 8), waarbij kortere invaarvolgtijden worden geconstateerd, dan bij minder of geen gestroomlijnde overgangen.

Bij een sluis is in de meeste gevallen een sluiswachterskantoor.

Verder kan een sluis uitgerust zijn met constructies om zoet en zout water te scheiden.

Op dit soort constructies wordt verder niet ingegaan, maar deze factoren kunnen wel de in- en uitvaartijden beïnvloeden alsmede de bedieningstijd van een sluis.

Veel sluisen worden gecombineerd met een brug voor het wegverkeer, wat tevens een goede bereikbaarheid van de sluis verzekert.

Op constructieve aspecten van zowel de sluis als het kanaal zal in dit ontwerp niet worden ingegaan.

De ruimte die een sluis inneemt is afhankelijk van het te verschutten aantal schepen. Die grote kan oplopen (in Nederland) tot 3 kolken van 320 m lengte en 24 m breedte en met wachtrij faciliteiten tot op een afstand van 2 km aan weerszijden van de sluis.

## 2. Het sluisgebeuren.

Bij aanvang van hun werk gaan de hulpsluiswachters eerst

naar het kantoor. Iedere hulpsluiswachter heeft een voor die dag "zijn" kolk toegewezen gekregen en blijft in afwachting van het eerstvolgende werk, dat hij verrichten moet, in het kantoor.

Een schip nadert de sluis en wordt gesignaleerd door de hulpsluiswachters of het schip geeft een signaal aan de hulpsluiswachters.

De hulpsluiswachter, die de kolk bedient, die bij matig gebruik in dienst is, verlaat het kantoor en constateert van welke kant dat schip komt. Hij zet de kolk om en opent de deuren, indien dit nodig is en geeft daarna de schipper een seintje om binnen te varen. Dit binnenvaren neemt een bepaalde tijdsduur in beslag, intussen geeft de hulpsluiswachter een volgend schip, dat in de tussentijd aangekomen is ook een seintje om binnen te varen, tenminste als dit schip nog in de sluis past, als dit schip niet past onderzoekt hij of een derde aangekomen schip nog past enz. Al deze schepen komen een bepaalde invaarvolgtijd later binnen dan hun voorganger. Indien er geen schepen meer aan die zijde van de sluis zijn, of de sluis is vol, d.w.z. er past geen enkel schip meer in, dat nog ligt te wachten aan die zijde.

De tijd, die er verstrijkt van het moment, dat de deuren gesloten worden totdat de deuren aan de andere zijde weer geopend worden heet de bedieningstijd van de kolk.

Het schip dat zover mogelijk naar voren ligt, vaart nu het eerst uit.

De tijd: "uitvaarvolgtijd" later passeert het volgende schip met zijn achterstevan de sluisdeuren enz. De hulpsluiswachter onderzoekt of er nog schepen aan die andere zijde wachten om geschut te worden. Indien dit het geval is, herhaalt zich het bovenstaande proces, indien er geen schepen liggen, gaat de hulpsluiswachter terug naar het kantoor.

Elke hulpsluiswachter heeft ditzelfde proces, de vraag is alleen: wat is het criterium voor het tijdstip, waarop een

tweede of derde hulpsluiswachter gaat werken.

Aangaande dit criterium heb ik in deze studie het volgende begrip ingevoerd:

Sorteercapaciteit.

Onder sorteercapaciteit van een kolk wordt verstaan:

De totale tonnage aan schepen, dat gesorteerd moet worden om tot een optimale kolkindeling te komen, waarbij een minimale maximum wachttijd ontstaat.

Deze capaciteit is geen echte capaciteit, maar meer een aanduiding. Dat toch het woord capaciteit is gebruikt komt doordat deze waarde sterk verbonden is met de werkelijke capaciteit van de sluis, d.w.z. de capaciteit in tonnenlaadvermogen van schepen per schutting.

Het werken met dit begrip gaat als volgt:

Op het moment, dat de sluis niet in bedrijf is wegens het ontbreken van schepen, is de sorteercapaciteit 0. Indien er één kolk in bedrijf is, is de sorteercapaciteit van de sluis gelijk aan de sorteercapaciteit van die kolk enz.

Het bepalen van deze sorteercapaciteit is gedaan aan de hand van zeer veel metingen vanuit het model. Resultaten zijn gegeven in het tweede deel van deze studie.

Het belang van deze sorteercapaciteit is tweërlei:

a. Indien het druk is bij een sluis, d.w.z. als er meer schepen aan een zijde van de sluis ligt, dan er de volgende schutting in de kolk passen naar alle waarschijnlijkheid, wordt de werkelijke capaciteit van de kolk (dus ook van de sluis op dat moment) overschreden door de intensiteit van de scheepvaart. Indien er nog een kolk niet in bedrijf is zorgt de waarde van de sorteercapaciteit ervoor, dat deze kolk in gebruik genomen wordt.

Tevens zorgt de waarde van de sorteercapaciteit ervoor dat eventueel deze laatste kolk weer aan de sluis onttrokken wordt.

b. De sorteercapaciteit is een attribuut van de kolk. Zij geeft aan hoeveel tonlaadvermogen zij kan bevatten bij

een bepaald scheepsaanbod. Verandert het scheepsaanbod, dan wijzigt eveneens deze sorteercapaciteit. Voor een optimale kolkindeling moeten een aantal schepen, die vooraan in de wachtrij liggen gesorteerd worden. Er worden dan niet meer schepen gesorteerd dan de sorteercapaciteit aangeeft.

Alvorens er gewerkt kan worden met deze sorteercapaciteit moet deze experimenteel bepaald worden (in het model is het mogelijk deze sorteercapaciteit te bepalen, waarbij dan tevens een toetsing op de juistheid van deze waarde gedaan kan worden).

3. De door de Rijkswaterstaat gehanteerde normen voor capaciteitsbepalingen

In deze paragraaf vindt u de berekening van de capaciteit van een sluis, zoals die aangegeven is in de Nota 70-12-7, welke is opgesteld door dienst Verkeerskunde, Hoofdafdeling Scheepvaart te Dordrecht.

Daar dit gedeelte niet door de samensteller van dit verslag is opgesteld maar ontleend is aan bovengenoemde nota en beschreven wordt in het afstudeerverslag van de heren Ooms en de Ruiter is dit gedeelte als bijlage opgenomen.

Gezien het belang van deze bijlage voor dit hoofdstuk volgt deze bijlage op de volgende bladzijden.

Direct daarna is opgenomen de beschouwing ten aanzien van de passeertijd van het afzonderlijke schip, zoals deze ook in dat verslag is opgenomen.



KAPASITEITSBEREKENING VAN DE BESTAANDE VAARWEG.

Voor de capaciteitsberekening wordt gebruik gemaakt van een door Rijkswaterstaat ontwikkelde methode, die uiteengezet wordt in de Rota 70.12.7. Het betreft hier een onderdeel van de vaarwegnota, dat is opgesteld door de dienst Verkeerskunde van de hoofdafdeling scheepvaart te Dordrecht en is gereedgekomen in december 1973. Omdat genoemde nota nog niet is gepubliceerd (tijd van schrijven is oktober '74), mag ik de inhoud bij de lezer niet bekend veronderstellen. Daarom volgt hieronder een korte beschrijving van de methode en een verklaring van de gebruikte symbolen.

Schutcapaciteit (Ontleend aan Nota 70.12.7)Algemene beschouwing en definitie

Als bij opeenvolgende schuttingen de sluis kolk volledig is bezet met schepen en er zich geen bijzondere vertragingen voordoen bij de bediening of de in- en uitvaart dan wordt de maximale hoeveelheid scheepvaartverkeer verwerkt. Met andere woorden de schutcapaciteit is bereikt. Ook bij volbezette kolken zal het aantal schepen per schutting, afhankelijk van de samenstelling van de vloot en de aankomst volgorde van de schepen, in het algemeen variëren. Dit heeft tot gevolg dat het maximum aantal schepen dat per schutcyclus wordt verwerkt en de schutcyclusduur zelf, niet konstant zijn. In verband hiermee wordt de schutcapaciteit met het oog op eenduidigheid, betrokken op een vaste tijdseenheid en als volgt gedefinieerd:

De capaciteit van een sluiscomplex is de maximale hoeveelheid verkeer, uitgedrukt in aantallen schepen of tonnen laadvermogen, die onder de voorkomende omstandigheden per tijdseenheid kan worden geschut als het sluisbedrijf continu volbezet werkt.

De definitie houdt dus in dat de capaciteit een gemiddelde waarde is uit een groot aantal schuttingen met volle kolken. Voor de capaciteit ( $C_s$ ) in schepen per uur geldt de volgende vergelijking:

$$C_s = \frac{2n_{\max}}{T_c} \quad (1)$$

Hierin zijn  $n_{\max}$  en  $T_c$  gemiddelde waarden van respectievelijk het aantal schepen en de cyclustijden van een groot aantal schuttingen met volbezette kolk.

De schutcyclustijd  $T_c$ .

De onderdelen van de schutcyclus volgen uit de schematische weergave van het

schutproces in figuur 1.

Een schutcyclus omvat twee opeenvolgende schuttingen:

$$T_c = T_d(\text{opvaart}) + T_d(\text{afvaart}) \quad (2)$$

De schutduur  $T_d$  is opgebouwd uit vaartijd en bedieningstijd.

$$T_d = T_i + T_u + T_b \quad (3)$$

Voor de betekenis van de toegepaste symbolen wordt verwezen naar tabel 1.

Voor sluizen die tussen twee kanaalpanden gesitueerd zijn vertoont  $T_b$  in het algemeen weinig speiding. Voor dit geval wordt de gemiddelde waarde toegepast. Voor sluizen aan of in gekanaliseerde rivieren, waar het verval sterk kan variëren, wordt uitgegaan van een frekwent voorkomende waarde van  $T_b$ .

Uit figuur 1 volgt:

$$T_i = t_1 + \sum t_i$$

$$T_u = \sum t_u$$

Hierin is:

- $t_1$ : lustijd (heeft betrekking op het eerst invarende schip)
- $t_i$ : invaarvolgtijd (overige invarende schepen)
- $t_u$ : uitvaarvolgtijd (alle uitvarende schepen)

Door middel van metingen kunnen de gemiddelde waarden van  $t_1$ ,  $t_i$ , en  $t_u$  bepaald worden voor een gegeven sluiskomplex { kolk of kolken plus voorhavens } als functie van het gemiddeld laadvermogen ( $\bar{T}$ ) en de beladingsgraad ( $Q$ ) van de passerende vloot.

Voor een bepaalde vaarrichting volgt de schutduur uit de vergelijking:

$$T_d = t_1 + (n-1)t_i + nt_u + T_b \quad (4)$$

Hierin is  $n$  het aantal schepen dat bij de schutting betrokken is. De in- en uitvaarvolgtijd en de lustijd kunnen voor gekozen waarden van het gemiddeld laadvermogen en de beladingsgraad van de vloot worden berekend voor een gegeven sluiskolk met de figuren 2 tm. 7.

Het maximum aantal schepen in de sluiskolk ( $n_{max}$ )

Voor een groot aantal, systematisch gevarieerde, kolkafmetingen is de waarde van  $n_{max}$  bepaald met behulp van kolkindelingssimulaties. Per sluiskolk zijn, afhankelijk van de afmetingen, 2 tot 5 verschillende waarden van  $\bar{T}$  toegepast. De resultaten worden weergegeven in de figuren 8 en 9. Hieruit kan worden afgeleid dat het verband tussen  $\bar{T}$  en  $n_{max}$  ongeveer de vorm heeft van de orthogonale hyperbool:

$$\bar{T} \cdot n_{max} = K \quad (5)$$

K is konstant voor een bepaald kolkoppervlak.

In figuur 10 wordt het theoretisch bepaalde verband weergegeven tussen de waarde van K (in tonnen) en het nuttige kolkoppervlak. Hierin zijn tevens een aantal punten opgenomen die gebaseerd zijn op waarnemingen in de praktijk. De afwijkingen ten opzichte van de theoretisch bepaalde lijn blijken relatief klein te zijn.

#### RECEPT VOOR DE BEREKENING.

#### 1. Bepaling van de schutkapasiteit ( $C_s$ ) als functie van het gemiddelde laadvermogen ( $\bar{T}$ )

$C_s = f(\bar{T})$  wordt bepaald door de berekening van  $C_s$  voor verschillende waarden van  $\bar{T}$  met behulp van vergelijking (1). De punten met de berekende coördinaten  $C_s, \bar{T}$  worden grafisch uitgezet en vervolgens verbonden door een vloeiende lijn.  $C_s$  kan dan worden afgelezen als functie van  $\bar{T}$ .

Voor de gekozen waarde van  $\bar{T}$  kan een berekeningsvolgorde worden aangehouden zoals hieronder puntgewijs is aangegeven.

#### A. Bepaling van $n_{max}$

1. Berekening van het nuttig kolkoppervlak O
2. Aflezen van K als functie van O in figuur 10
3. Toepassing van vergelijking (5)

#### B. Berekening van de schutcyclustijd $T_c$

1. Bepaling van het kolk-voorhavengetal KV, per vaarrichting.

KV bestaat uit drie cijfers p-q-r die respectievelijk betrekking hebben op  $t_1, t_i$  en  $t_u$ . Voor de bepaling van KV zijn geen algemeen geldende regels te geven. De waarde hangt af van de vormgeving en inrichting van de sluiskolk en voorhavens voorzover deze van invloed zijn op het vaargedrag.

Als KV=1-1-1 dan treden de kortste vaartijden op. Dit is het geval bij de meeste moderne sluiskolken. De voorhavens van deze sluizen liggen



in het verlengde van de kolken en de wachtplaatsen zijn gunstig gesitueerd ten opzichte van de sluisingen. In verband hiermee wordt voor  $p$  de waarde 1 gekozen. Voor deze sluizen geldt verder dat de vormfaktor  $O/LB=1$ ; d.w.z. dat de kolkbreedte gelijk is aan de doorvaartbreedte in de sluishoofden. Bij deze vormgeving hoeven de schepen geen tijdrovende manoeuvres uit te voeren om te kunnen afmeren na de invaart of om slaags te komen voor het sluishoofd voor de uitvaart. Voor  $q$  en  $r$  moet dan eveneens de waarde 1 worden gekozen.

Bij veel oude sluiscomplexen wijkt de vormgeving sterk af van de hiervoor beschreven moderne sluizen. Wanneer de invaart van het eerste schip sterk vertraagd wordt als gevolg van een ongunstige situering van de wachtplaats dan dient  $p$  de waarde 2 of zelfs 3 te krijgen. Als de vormfaktor van de sluis kolk duidelijk kleiner is dan 1 dan dienen  $q$  en  $r$  de waarde 2 te krijgen. (Voor de sluizen in de Zuid-Willemsvaart en de sluis te Engelen geldt een gunstige ligging van de voorhavens, maar een vormfaktor die duidelijk kleiner is dan 1 dus  $KV=1-2-2$ ).

## 2. Bepaling van de lustijd $t_1$ per vaarrichting.

- Aflezen van  $\bar{b}$  en  $\bar{l}$  als functie van de gekozen  $T$  in figuur 11.
- Bepaling van de lusafstand  $A_1$  (dat is de afstand van het voorschip tot de invardeuren).
- Berekening van  $\bar{b}/B$  ( $B$  is de doorvaartbreedte van het sluishoofd).
- Aflezen van  $t_1$  als functie van  $A_1$  en  $\bar{b}/B$  voor de gekozen  $\bar{T}$  in de figuren 2 en 3, respektievelijk voor geladen en ongeladen schepen.
- Berekening van het gewogen gemiddelde door de  $t_1$  voor geladen en ongeladen schepen te vermenigvuldigen met  $Q$  (percentage geladen schepen) en de  $t_1$  voor ongeladen schepen met  $(Q-1)$  en vervolgens beide waarden te sommeren.
- De uiteindelijke waarde van  $t_1$  wordt tenslotte gevonden door de onder e bepaalde waarde te vermenigvuldigen met 1, 1,2 of 1,4 als op grond van het kolk-voorhavengetal de waarde van  $p$  respektievelijk gelijk is aan 1,2 of 3.

## 3. Bepaling van de invarvolgtijd $t_i$ per vaarrichting.

- Bepaling van  $F$  voor het sluishoofd dat bij de invaart wordt gepasseerd.
- Het aflezen van  $t_i$  voor geladen en ongeladen schepen in figuur 4 ( $KV:q=1$ ) of 5 ( $KV:q=2$ ).
- Berekening van het gewogen gemiddelde overeenkomstig 2e voor de lustijd.

## 4. Bepaling van de uitvarvolgtijd $t_u$ per vaarrichting.

- Bepaling van  $F$  voor het sluishoofd dat bij uitvaart gepasseerd moet worden.
- b en c. Als bij  $t_i$  (figuren 6 en 7).



Tabel	<u>Notatie</u>	
Symbolen	Betekenis	Dimensie
$A_1$	lusafstand	m
$B$	doorvaartbreedte sluishoofd	m
$B_k$	sluiskolkbreedte	m
$C_s$	kapasiteit	
$C_t$	kapasiteit	tonnen laadv/uur
$C_w$	kapasiteit	schepen/week
$D$	waterdiepte boven de drempel	m
$F$	oppervl. natte dwarsdoorsnede over de drempel	$m^2$
$I_w$	intensiteit of scheepsaanbod	schepen/week
$K$	kostante	ton
$KV$	kolkvoorhavengetal	
$L$	nuttige kolk lengte	m
$N$	aantal kolken per sluiskomplex	
$O$	nuttig kolkoppervlak	$m^2$
$T$	laadvermogen	ton
$T_b$	bedieningstijd	min.
$T_c$	schutcyclustijd	min.
$T_d$	schutduur	min.
$T_i$	duur van de totale invaart	min.
$T_u$	duur van de totale uitvaart	min.
$b$	scheepsbreedte	m
$d$	diepgang	m
$f$	nat grootspantoppervlak	$m^2$
$l$	scheepslengte	m
$n$	aantal schepen per schutting	
$n_{max}$	aantal schepen bij volbezette kolk	
$t_i$	invaarvolgtijd	min.
$t_l$	lustijd	min.
$t_u$	uitvaarvolgtijd	min.
$Q$	beladingsgraad	
$y$	hoek tussen sluisas en voorhaven	

Opmerking: De gemiddelde waarde van een variabele wordt aangegeven met een streepje boven het symbool.



## SLUIZEN- DE PASSEERTIJD VAN HET AFZONDERLIJKE SCHIP. 1)

### Algemeen

De passeertijd van een te schutten schip ( $t_p$ ) begint op het tijdstip dat het schip in de voorhaven van de sluis arriveert en eindigt op het moment dat het schip de uitvaardeuren passeert.

Voor de passeertijd geldt de volgende vergelijking :

$$t_p = t_w + t_o + t_s$$

$t_w$  = wachttijd.

Deze begint op het tijdstip van aankomst in de voorhaven en eindigt op het moment dat de invaardeuren (voor het betrokken schip) voor de eerste maal na aankomst gesloten kunnen worden. Er zijn dan twee mogelijkheden.

- Het schip bevindt zich in de kolk om geschut te worden; in dat geval is het einde van de wachttijdtevens het begin van de schuttijd.
- Het schip kan niet met deze schutting mee, t.g.v. een te groot schepsaanbod, en moet overliggen in de voorhaven; het einde van de wachttijd is dan het begin van de overligtijd.

$t_s$  = schuttijd.

Deze begint op het tijdstip dat de invaart voltooid is en met het sluiten van de invaardeuren begonnen kan worden. De schuttijd eindigt op het moment dat de achterstevan van het betrokken schip de uitvaardeuren passeert.

$t_o$  = overligtijd.

Deze treedt op als het aantal schopen dat vooraf gaat aan het beschouwen de schip, gelijk is aan, of groter is dan, het aantal dat de kolk maximaal kan bevatten. Het schip moet dan tenminste overliggen tot de volgende schutting.

1) Begrippen en definities zijn ontleend aan de nota 70.12.7 van de Hoofdafdeling Verkeerskunde van Rijkswaterstaat.

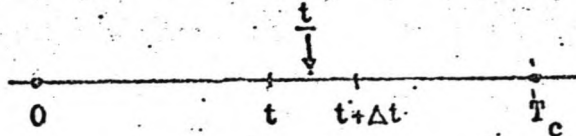
De onderdelen van de passeertijd moeten nu bepaald worden.

De wachttijd  $t_w$

Voor de bepaling van de gemiddelde wachttijd van een willekeurig schip, na aankomst bij een sluiskomplex met 1 kolk wordt uitgegaan van de volgende veronderstellingen :

1. Het schip kan op ieder willekeurig tijdstip  $t$  aankomen m.a.w.  $t$  heeft een uniforme kansverdeling.
2. Per sluiskolk is de schutcyclustijd  $T_c$  konstant en gelijk aan de gemiddelde waarde die optreedt bij een gegeven scheepsaanbod en samenstelling van de vloot ( $T_c$  is reeds bepaald bij de capaciteitsberekening).

één kolk



Het schip arriveert op een willekeurig tijdstip ( $t$ ) tussen  $t = 0$  en  $t = T_c$ . De kans dat  $t$  ligt tussen  $t$  en  $t + \Delta t$  d.i. de elementaire kans in het geval van een uniforme verdeling is :

$$P_r \left\{ t \leq \underline{t} \leq t + \Delta t \right\} = \frac{1}{T_c} \Delta t$$

De wachttijd duurt van  $t$  tot  $T_c$  en is dus  $T_c - t$ . De gemiddelde wachttijd voor alle mogelijke aankomsttijdstippen tussen 0 en  $T_c$  bedraagt :

$$t_w = \int_0^{T_c} \frac{1}{T_c} (T_c - t) dt \longrightarrow t_w = \frac{1}{2} T_c$$

De schuttijd  $t_s$

De schuttijd begint als de invaart voltooid is en de invardeuren gesloten kunnen worden. De schuttijd eindigt als de achtersteven van het betrokken schip de uitvardeuren passeert. De schuttijd bestaat dus uit de bedieningstijd  $T_b$  en een gedeelte van de uitvaartijd  $T_u$ .

In geval van een schutting met 1 schip is de schuttijd:  $t_s = T_b + t_u$   
 Omvat de schutting  $n$  schepen, dan is de gemiddelde schuttijd :

$$t_s = T_b + \frac{1}{2}(n + 1)t_u$$

De schuttijd en de wachttijd zullen voor een bepaalde sluiskolk niet al te sterk variëren. De som van beide tijden ( $t_w + t_s$ ) wordt als het konstante deel van de passeertijd berekening gezien. Dit in tegenstelling tot de gemiddelde overligtijd, die sterk afhankelijk is van de grootte en de vorm van het verkeersaanbod en daarom als het variabele deel van passeertijd wordt beschouwd.

#### De overligtijd $t_o$

Bij een gering verkeersaanbod kunnen alle schepen mee met de eerstvolgende schutting. De ondervonden vortraging is dan gelijk aan de passeertijd bij lopende vaart ( $t_p = t_w + t_s$ ).

Als de verkeersintensiteit toeneemt zal het voorkomen dat op verschillende uren het aanbod groter is dan de capaciteit. Een deel van de schepen kan dan niet direct verwerkt worden maar moet overliggen tot de volgende schutting. Bij een nog verdere toename van de intensiteit treedt accumulatie van schepen op in de voorhavens. Dit overschot wordt afgebroken in de perioden dat de intensiteit lager is dan de capaciteit.

De passeertijden van schepen bij schutsluizen worden dus duidelijk beïnvloed door de variatie in het verkeersaanbod. Het verkeersaanbod heeft deels een systematisch deels een stochastisch karakter. Een systematische opbouw van het aanbod wordt veroorzaakt door het optreden van zowel etmaalcycli als weekcycli (verschil tussen werkdagen enerzijds en zaterdag en zondag anderzijds). De bedrijfstijden van veel sluiscomplexen zijn op deze cycli afgestemd. Daarom wordt bij de bepaling van overligtijden uitgegaan van een verkeersontwikkeling op weekbasis.

De belastingsgraad wordt daarom gedefinieerd als het quotient van de weekintensiteit en de weekcapaciteit.

$$\text{belastingsgraad} = I_w / C_w$$

Het verband tussen de gemiddelde overligtijd per schip en de belastingsgraad is door Rijkswaterstaat bepaald m.b.v. schutsimulaties. Daarbij is uitgegaan van in de praktijk waargenomen verkeersaanbodspatronen.

Het resultaat van deze studie wordt samengevat in de figuur op de volgende bladzij.





$t_w + t_s$ , dat is het min of meer konstante deel vande passertijd.

afgeleid is:  $t_w = 1/2 T_c$

$$t_s = T_b + 1/2 (n+1) t_u.$$

$T_b$  is onafhankelijk van  $n$  en  $\bar{T}$ .

$T_c$  is enigszins afhankelijk van  $n$ . Om de berekening niet te ingewikkeld to maken wordt  $T_c$  bepaald voor  $n = n_{max}$  en voor  $n = 0,4 \cdot n_{max}$ . Voor waarden  $n_{min} < n < n_{max}$  wordt  $T_c$  lineair verlopend met  $n$  verondersteld.

De berekening wordt uitgevoerd voor 3 gemiddelde tonnages, achtereenvolgens voor  $\bar{T} = 400$  ton,  $\bar{T} = 500$  ton en  $\bar{T} = 600$  ton.

$$T_c = T_d \text{ opvarend} + T_d \text{ afvarend}$$

Uit de capaciteitsberekening is al gebleken dat het kleine verschil tussen  $T_d$  opv. en  $T_d$  afv. veroorzaakt wordt door een verschil in  $Y_{opv.}$  en  $Y_{afv.}$

Anders gezegd: als er in opvarende richting een andere verhouding is tussen geladen en lege schepen dan in afvarende richting ( $Y_{opv.} \neq Y_{afv.}$ ) dan is er ook een verschil tussen de schutduur opvarend en de schutduur afvarend. Om de passeertijd te bepalen moeten voor iedere sluis dus de waarden van  $Y_{opv.}$  en  $Y_{afv.}$  bekend zijn. Deze waarden zijn voor de huidige situatie bekend, maar voor iedere sluis zijn zij verschillend (zie analyse van de huidige scheepvaartbeweging, deel 1) Voor de nieuwe situatie zijn deze waarden moeilijk te voorspellen, omdat de Zuid-Willemsvaart er een geheel nieuwe functie verkrijgt, namelijk de functie van doorgaande vaarweg. Het is in dit stadium nog niet te zeggen hoe omvangrijk die doorgaande vaart zal zijn. Mede omdat de invloed van de  $Y$ -waarden op de schutduur maar gering is wordt hier afgezien van een poging om de  $Y$ -waarden te schatten. Voor alle sluisen in het kanaal werken we met  $Y_{opv} = Y_{afv} = 0,5$ , dat wil zeggen:

$$\underline{T_d \text{ opv} = T_d \text{ afv} = t_w}$$

Ter illustratie van de behandelde theorie wordt hier de passeertijd van een moderne sluis van 120 x 12 meter berekend.

<u>gegevens:</u> $L = 120$ m	$O = B \times L = 14.400$ m <sup>2</sup>	$K = 1625$ ton
$B = 12$ m	$A_{sluis} = 100$ m + scheepslengte	
$D = 3,50$ m	$T_b = 10$ minuten	
$F_{opv} = F_{afv} = 42$ m <sup>2</sup>		

De lijst van gebruikte symbolen is reeds gegeven in deel 1 op blz. 25.

De capaciteit van deze sluiskolk is reeds berekend in deel 4 bijlage 13. Aan die bijlage ontleen we de volgende lustijden, invaaryolgtijden en uitvaaryolgtijden.

$\bar{T}$	$t_1$	$t_i$	$t_u$
400	2,4	2,1	1,3
500	2,6	2,5	1,4
600	2,8	2,7	1,6

De schutduur volgt uit:  $T_d = t_1 + (n-1) \cdot t_i + n t_u + T_b$

$\bar{T}$	400	500	600
$n = n_{max}$	4,1	3,3	2,7
$n = 0,4 n_{max}$	1,6	1,3	1,1
$T_d (n = n_{max})$	24,8	23,6	22,5
$T_d (n = 0,4 n_{max})$	16,2	15,6	14,1
$t_w = 1/2 T_c = T_d$ ( $n = n_{max}$ )	24,8	23,6	22,5
( $n = 0,4 n_{max}$ )	16,2	15,6	14,1
$t_s \begin{cases} T_b \\ 1/2 (n+1) \cdot t_u \end{cases}$ ( $n_{max}$ )	10,0	10,0	10,0
( $0,4 n_{max}$ )	3,3	3,2	3,0
	1,7	1,7	1,7
$t_p = t_w + t_s$ ( $n = n_{max}$ )	38,1	36,8	35,5
( $n = 0,4 n_{max}$ )	27,9	27,3	25,8

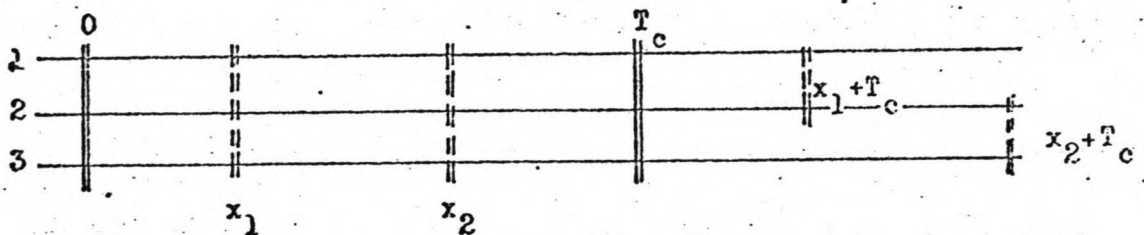
Voor een gemiddeld tonnage van 400 ton zetten we de passeertijd bij lopende vaart (dwz.  $t_p = t_w + t_s$ ) uit tegen de kolkbezettingsgraad ( $n$ ). We vinden dan een passeertijd die lineair toeneemt met de kolkbezettingsgraad. Om de werkelijke passeertijd te vinden moet de overlig-tijd hier bij opgeteld worden.

Als we naar de grafiek op blz.15 kijken zien we dat de gemiddelde overligtijd daar gegeven wordt als functie van de belastingsgraad op weekbasis. ( $I_w/C_w$ ). Bij een belastingsgraad  $I_w/C_w = 1$  hoort een kolkbezetting-tingsgraad  $n = n_{max}$ . Een bezettingsgraad van 0,5 betekent een gemiddelde kolkbezetting-tingsgraad van  $\bar{n} = 0,5 n_{max}$ . Blijkbaar mag de grafiek van blz. 15 gesuperponeerd worden op de lijn voor  $t_w + t_s$ .



De schutcyclus van de eerste kolk duurt van 0 tot  $T_c$ . De begintijdstippen van de schutcycli van de 2<sup>e</sup> en de 3<sup>e</sup> kolk zijn respectievelijk  $x_1$  en  $x_2$ . De stochastische variabelen  $x_1$  en  $x_2$  zijn uniform verdeeld op het interval  $(0, T_c)$ . De elementaire kansen met betrekking tot  $x_1$  en  $x_2$  zijn in dit geval:

Aan de nota 70.12.7 van de dienst verkeerskunde, hoofdafdeling Scheepvaart, wordt de volgende beschouwing over een complex met 3 kolken ontleend.



De schutcyclus van de eerste kolk duurt van 0 tot  $T_c$ . De begintijdstippen van de schutcycli van de 2<sup>e</sup> en de 3<sup>e</sup> kolk zijn respectievelijk  $x_1$  en  $x_2$ . De stochastische variabelen  $x_1$  en  $x_2$  zijn uniform verdeeld op het interval  $(0, T_c)$ . De elementaire kansen met betrekking tot  $x_1$  en  $x_2$  zijn in dit geval:

$$\Pr \left\{ x_1 \leq \underline{x}_1 \leq x_1 + \Delta x_1 \right\} = \frac{1}{T_c} dx_1$$

$$\Pr \left\{ x_2 \leq \underline{x}_2 \leq x_2 + \Delta x_2 \right\} = \frac{1}{T_c} dx_2$$

Een schip arriveert op een willekeurig tijdstip tussen  $t=0$  en  $t=T_c$ . De elementaire kans op de aankomst van een schip is dus, evenals bij een enkele kolk:

$$\Pr \left\{ t \leq \underline{t} \leq t + \Delta t \right\} = \frac{1}{T_c} dt.$$

Bij de berekening van  $t_w$  moeten we twee gevallen onderscheiden:  $x_2 > x_1$  en  $x_2 < x_1$ .

Geval 1:  $x_2 > x_1$

Het aankomsttijdstip,  $t$ , kan liggen tussen:

- a. 0 en  $x_1$       wachttijd is  $x_1 - t$ .
- b.  $x_1$  en  $x_2$       wachttijd is  $x_2 - t$ .
- c.  $x_2$  en  $T_c$       wachttijd is  $T_c - t$ .

Geval 2:  $x_2 < x_1$

Geval 2 is identiek aan geval 1, met dien verstande dat  $x_1$  en  $x_2$  verwisseld zijn. Beide gevallen zullen dus dezelfde uitkomst opleveren.

$$t_w = \frac{2}{T_c^3} \left[ \int_0^{T_c} \int_0^{x_2} \left\{ \int_0^{x_1} (x_1 - t) dt \right\} dx_1 \cdot dx_2 + \int_0^{T_c} \int_0^{x_2} \left\{ \int_{x_1}^{x_2} (x_2 - t) dt \right\} dx_1 dx_2 + \int_0^{T_c} \int_0^{x_2} \left\{ \int_{x_2}^{T_c} (T_c - t) dt \right\} dx_1 dx_2 \right] =$$

$$t_w = \frac{2}{T_c^3} \left[ \frac{1}{24} T_c^4 + \frac{1}{24} T_c^4 + \frac{1}{24} T_c^4 \right] = \frac{1}{4} T_c$$

N.B. Een meer algemene afleiding voor  $N$  kolken geeft als resultaat

$$t_w = \frac{T_c}{N+1}$$

Opmerking: Met dit resultaat is in principe de wachttijd bij ieder sluis-komplex te bepalen. Zoals we later zullen zien is er echter nog een praktisch probleem bij de complexen die bestaan uit kolken van verschillende afmetingen, met dientengevolge ook verschillende cyclustijden.

De tweede paragraaf van dit hoofdstuk geeft aan de procesbeschrijving van zowel de hulpsluiswachters als van het passeren van een sluis.

De waarden van de attributen van de componenten schepen en hulpsluiswachters worden ontleend aan de tabellen, die in paragraaf 3 van dit hoofdstuk zijn bijgevoegd.

Wel wordt aan deze tabellen een geheel andere interpretatie gegeven t.a.v. het passeren van het afzonderlijke schip.

In hoofdzaak komt dit hier op neer:

Volgens de berekening van de nota wordt bepaald wat het gemiddelde tonnage is van alle schepen, die sluis naar verwachting passeren. Hieruit wordt bepaald wat de gemiddelde invaarvolgtijd van al die schepen is.

De interpretatie, die in dit verslag aan die tabellen gegeven wordt is:

Bepaal aan de hand van het tonnage van een schip wat zijn individuele invaarvolgtijd is bij een sluis.

Een uitgebreide studie van de afwijkingen, die ontstaan, omdat van deze tabellen uitgegaan is, is niet gedaan. Daar waar categorisch afwijkingen ontstaan in de simulatie kan met vermenigvuldigingsfactor gecorrigeerd worden voor elk scheepstype afzonderlijk.

## Hoofdstuk V

### Modelanalogie van het sluisgebeuren.

1. Voordat het model kan gaan werken moeten weer eerst alle attributen en componenten gedeclareerd worden.

Die van de schepen zijn reeds besproken, die van een sluis volgen hieronder:

```
1 SCHUTSLUIS(25),  
2 NAAM CHARACTER (15) VARYING,  
2 KANTOOR POINTER,  
2 TON(2) FLOAT,  
2 CAP FLOAT,  
2 KY(2) POINTER,  
2 KANAALLENGETE(2) FLOAT,  
2 TOJKOLKEN FIXED BIN,
```

- a. Schutsluis(25)

Het model gaat er van uit, dat er maximaal 25 sluisen gepasseerd worden. We spreken in dit hoofdstuk slechts over het passeren van 1 sluis.

- b. Naam

In de getalkaarten wordt de naam van de sluis genoteerd (max. 15 karakters). In de uitvoer gaat de naam van de sluis vooraf aan zijn berekeningen.

- c. Kantoor

Kantoor wordt geïnitieëld als een wachtrij, waar de hulp-sluiswachters kunnen verblijven, indien zij geen werk te doen hebben.

- d. Ton(2)

Aan beide zijden van de sluis wordt steeds bijgehouden hoeveel tonlaadvermogen er aan schepen aanwezig zijn. (ton).

- e. Cap

Cap is de capaciteit van de sluis, tenminste van dat gedeelte, dat in gebruik is.

Elke kolk heeft zijn eigen capaciteit. Die van de sluis is het totaal van de capaciteiten van de kolken. Werkt er één kolk niet, dan wordt de capaciteit van de sluis verminderd met de capaciteit van die ene kolk. Let wel: Onder de capaci-

teit wordt hier verstaan de hoeveelheid tonlaadvermogen dat per schutting omgezet wordt.

f. Rij(2)

Aan beide zijden van de sluis komen schepen aan. Deze worden in een wachtrij geplaatst. Hiermee wordt niet bedoeld, dat die schepen ook inderdaad moeten wachten. Deze wachttijd kan ook 0 zijn.

g. Kanaallengte(2)

Aan beide zijden van de sluis bevindt zich een waterweg. Voor de eenvoud wordt gerekend, dat deze tot op een bepaalde afstand bij de sluis horen. Op deze manier kunnen sluizen aan elkaar gekoppeld worden, om uiteindelijk tot een kanaal of zelfs een netwerk simulatiemodel te komen. (km).

h. Totkolken

Totkolken geeft aan uit hoeveel kolken de sluis bestaat.

Per kolk is een hulpsluiswachter werkzaam. Deze hulpsluiswachter heeft een aantal attributen en omdat hij toch een "vaste" kolk bedient, wordt hem ook de attributen van die kolk toegewezen.

1 HULPSLUISWACHTERPARL BASED(HULPSLUIS),  
2 SLUISLENGTE FLOAT,  
2 SLUISBREEDTE FLOAT,  
2 KOLK DIEPTE(2) FLOAT,  
2 LUS FLOAT,  
2 KV FIXED BIN,  
2 BEDIENINGSTYD FLOAT,  
2 SLUIS POINTER,  
2 KOLK FLOAT,  
2 KOLKNO FIXED BIN,  
2 BOOT POINTER,  
2 NEXTSCHIP POINTER,  
2 SCHUTSLNO FIXED BIN,  
2 RAAT(50,2) FLOAT,  
2 ZYDE FIXED BIN,

a. Sluislengte

Lengte van de kolk (m).

b. Sluisbreedte

Breedte van de kolk (m).

c. Lus

De lusafstand is de afstand van de invaardeuren tot het voorschip van het eerste schip, dat kan binnenvaren.

e. KV

Kolkvoorhavengetal.



In de Nota 70-12-7 spreekt men van het kolkvoorhavengetal (KV), hetwelk aangeeft, hoe de situatie bij een sluis is wat betreft de doorvaartijden.

KV=1-1-1 geeft aan dat zowel de lustijd, als de invaarvolgtijd en de uitvaarvolgtijd gunstiger zijn dan b.v. KV=2-2-2.

f. Bedieningstijd

De gemiddelde tijd die nodig is om de invaardeuren te sluiten, de kolk om te zetten en de uitvaardeuren te openen.

g. Sluis

De schepen varen de sluis binnen, nemen de hun door de hulpsluiswachter toegewezen plaats in en wachten totdat zij weer uit kunnen varen. Hier is dus weer sprake van een wachtrij, die Sluis genoemd wordt. Het schip wordt gesorteerd in deze wachtrij naar de plaats, die zijn achterstevan aangeeft.

h. Kolk

Kolk geeft de grootte van de schutcapaciteit van de kolk aan. Als deze bekend is, kan hij reeds ingevuld worden. Zoniet, vul dan een willekeurige waarde in. Door parameter SORCAPBEP de waarde "true" ('1'B) te geven, rekent de computer de juiste waarde van de capaciteit van de kolk uit.

Hiervoor is het noodzakelijk de intensiteit van het scheepvaartverkeer groter te maken dan de te verwachten capaciteit, althans voor de eerste simulatie.

i. Kolkno

Kolkno is het nummer van de kolk.

De kolken moeten genummerd worden in de volgorde van welke het meest gebruikt wordt.

j. Boot

Iedere keer als een hulpsluiswachter een schip instructies gegeven heeft omtrent zijn ligplaats in de sluis, noemt hij het volgende schip weer BOOT. om aan te geven met welk schip hij te maken heeft.

k. Nextschip

Nextschip is de opvolger van Boot.

Uit het proces van de hulpsluiswachter zal blijken, wat hier de functie van is.

l. Schutslno

Schutslno geeft aan bij welke sluis de hulpsluiswachter werkt.

m. Raai(50,2)

Dit array is bestemd voor de kolkindeling.

We gaan er van uit, dat er niet meer dan 50 schepen in de sluis gaan.

Het eerste schip dat binnenvaart, vaart zover mogelijk door naar voren.

Achter het schip wordt een denkbeeldige lijn getrokken, een raai, die een bepaalde breedte heeft een bepaalde afstand tot de invardeuren.

Hieraan wordt getoetst of een volgend schip past of niet.

Elke hulpsluiswachter moet de mogelijkheid hebben deze raaien aan te brengen, dus is dit een attribuut van de hulpsluiswachters.

n. Zijde

Zijde geeft aan welke kant de deuren open zijn.

Zijde=1, dan staan de deuren van de kolk aan de bovenstroomse zijde open.

Dit zijn de attributen van de hulpsluiswachters.

Het initialiseren en het alloceren gebeurt op de volgende wijze:

```
68 1 0   DO K=1 TO TOTSLUIZEN;
69 1 1   GET LIST(NAAAM(K),KANAALLENGTE(K,1));
70 1 1   GET LIST(KANAALLENGTE(K,2),TOTKOLKEN(K));
71 1 1   NUM=K;TEKST3=NUM;TEKST4=NUM;SUBSTR(TEKST3,1,6)='RY  ';
75 1 1   RY(K,1)=NEWQUEUE(TEKST3);RY(K,2)=NEWQUEUE(TEKST3);
77 1 1   KANTOOR(K)=NEWQUEUE;
78 1 1   TON(K,1)=0;TON(K,2)=0;CAP(K)=0;
81 1 1   PUT SKIP(5);
82 1 1   KAN(K,*)=KANAALLENGTE(K,*);
83 1 1   IF CAPACITEITSBEPALING THEN KANAALLENGTE(K,*)=0;
84 1 1   DO N=1 TO TOTKOLKEN(K);
85 1 2   ALLOCATE HULPSLUISWACHTERPARL;
86 1 2   HULPSL(K,N)=HULPSLUI;SLUISRY=NEWHIST(20,0,1);
88 1 2   ZYDE=1;
89 1 2   SCHUTSLNO=K;
90 1 2   KOLKNO=N;COM: FORMAT(X(3),A(20),F(7,2));
92 1 2   GET LIST(SLUISENLENGTE,SLUISBREEDTE,KOLKDIEPTE,LUS,KV);
93 1 2   GET LIST(BEDIENINGSTYD);KOLK=2000;
95 1 2   IF ~CAPACITEITSBEPALING THEN DO;GET LIST (KOLK);END;
98 1 2   CAP(K)=CAP(K)+KOLK;
99 1 2   SUBSTR(TEKST4,1,5)='SLUIS  ';SLUIS=NEWQUEUE(TEKST4);
101 1 2  PUT STRING(NOT(1))EDIT('KOLKGEDEVENS KOLK',N) (R(COM));
102 1 2  PUT STRING(NOT(2))EDIT('SLUISENLENGTE=',SLUISENLENGTE) (R(COM));
103 1 2  PUT STRING(NOT(3))EDIT('SLUISBREEDTE=',SLUISBREEDTE) (R(COM));
104 1 2  PUT STRING(NOT(4))EDIT('KOLKDIEPTE 1=',KOLKDIEPTE(1)) (R(COM));
105 1 2  PUT STRING(NOT(5))EDIT('KOLKDIEPTE 2=',KOLKDIEPTE(2)) (R(COM));
106 1 2  PUT STRING(NOT(6))EDIT('LUSAFSTAND=',LUS) (R(COM));
107 1 2  PUT STRING(NOT(7))EDIT('KV=',KV) (R(COM));
108 1 2  PUT STRING(NOT(8))EDIT('BEDIENINGSTIJD=',BEDIENINGSTYD) (R(COM));
109 1 2  S=1+(30*(N-1));DO M=1 TO 8;SUBSTR(NAT(M),S, 30 )=NOT(M);END;
113 1 2  IF CAPACITEITSBEPALING THEN DO;
114 1 3  CALL ACTIVATE(NEW(HULPSLUI),AT((1/5)*SIMULATIETYD),BYKOLK);
115 1 3  END;ELSE DO;
117 1 3  CALL ACTIVATE(NEW(HULPSLUI),BYKOLK);END;
119 1 2  END;
120 1 1  END;
```

Uit de opbouw van deze initialisatie blijkt, dat per schutsluis zoveel kolken worden geïntialiseerd als de sluis kolken heeft. Deze initialisatie bestaat voornamelijk uit invoer- en uitvoer-opdrachten, terwijl er ook een aantal opdrachten instaan ten behoeve van de lay-out van de uitvoer. Verder spreken de opdrachten voor zich.

Nu de initialisatie van de componenten gerealiseerd zijn, kunnen deze processen starten.

Hoewel de procedures nog niet geïntialiseerd zijn, zullen toch eerst de processen van de hulpsluiswachters en het passeren van de sluis worden besproken:

Daar de hulpsluiswachters het eerst aan het werk worden gezet, wordt het eerst dit proces besproken.

```

343 1 0 /*PROCES VAN DE HULPSLUISWACHTERS*/
      BYKOLK:
344 1 1   IF WEINIGSCHEPENAANKEZIG THEN DO;
345 1 1   CALL SORTINQUEUE(KANTOOR(SCHUTSLNO),-KOLKNO);
347 1 1   CALL PASSIVATE;
349 1 0   CALL OUTOFQUEUE;END;
350 1 0   IF TON(SCHUTSLNO,ZYDE)->CAP(SCHUTSLNO)-KOLK THEN
351 1 1   DO;ZYDE=3-ZYDE;
353 1 1     CALL HOLD(BEDIENINGSTYD);
354 1 0   END;
355 1 0   RAAI=0;RAAI(1,1)=SLUISBREEDTE;RAAI(1,2)=SLUISLENGTE;TAL=0;
358 1 0   CALL VERANDERDEVOLGORDE;
359 1 0   EERSTESCHIP:
360 1 0     BOOT=FIRSTOFQUEUE(RY(SCHUTSLNO,ZYDE));
362 1 0     IF BOOT=NULL THEN GOTO BYKOLK;ELSE BOOT->LUSSCHIP='1'B;
      KOLKINDELEN:
363 1 0     BOOT=FIRSTOFQUEUE(RY(SCHUTSLNO,ZYDE));
364 1 1     DO WHILE(BOOT<=>NULL);
365 1 1       NEXTSCHIP=SUC(BOOT);
366 1 1       IF SCHIPPAST THEN DO;
367 1 2         BOOT->VOLGNO=-(RAAI(P,2)-BOOT->LENGTE);
368 1 2         BOOT->HULPSLUISW=HULPSLUIS;
369 1 2         BOOT->INVAARTYD=INVAART;
370 1 2         BOOT->LUSTYD=LUST;
371 1 2         BOOT->UITVAARTYD=UITVAART;
372 1 2         GEWICHT(SCHUTSLNO,KOLKNO)=GEWICHT(SCHUTSLNO,KOLKNO)+
373 1 2           BOOT->TONNAGE;TAL=TAL+1;
374 1 2         CALL REACTIVATE(BOOT,AFTER,CURRENT);
375 1 2         CALL PASSIVATE(KOLKINDELEN);END;ELSE
376 1 2         BOOT=NEXTSCHIP;
377 1 1       END;
378 1 0       TEL(SCHUTSLNO,KOLKNO)=TEL(SCHUTSLNO,KOLKNO)+KOLK;
379 1 0       CALL HISTPUT(SLUISTRY,TAL);
380 1 0       CALL HOLD(BEDIENINGSTYD);
382 1 0       ZYDE=3-ZYDE;
383 1 0       CALL REACTIVATE(FIRSTOFQUEUE(SLUIS));
384 1 0       CALL PASSIVATE(BYKOLK);
      /*EINDE VAN HET PROCES VAN DE HULPSLUISWACHTERS*/
```

Het eenvoudigst kan de bespreking gebeuren aan de hand van een beschrijvende tekst, die letterlijk het programma volgt:

Een hulpsluiswachter wordt geactiveerd bij BIJKOLK.

Als er weinig schepen zijn gaat hij naar het kantoor en wacht totdat hij weer aan het werk gezet wordt.

Hij verlaat dan het kantoor en kijkt aan welke zijde hij moet beginnen. Eventueel zal hij eerst de kolk om moeten zetten, wat een bedieningstijd lang in beslag neemt.

Er liggen dan geen schepen in de kolk, dus hij heeft ook geen denkbeeldige lijnen getrokken. Hij trekt er nu een bij de uitvaardeuren. Deze is de sluisbreedte breed en de sluislengte lang.

Hij verandert de volgorde in de wachtrij, zodanig dat de kolk optimaal gevuld wordt.

Het eerste schip noemt hij boot. Mochten er toch geen schepen meer aan die zijde liggen, omdat het schip, dat hem geactiveerd had, toch al in een kolk kon varen, gaat hij terug naar BIJKOLK (als er weinig schepen aanw..... enz.), anders is boot een lus-schip.

Nu gaat hij de kolk indelen:

Het schip dat vooraan in de wachtrij ligt noemt hij elke keer weer BOOT.

Zolang er ook werkelijk BOTEN zijn doet hij het volgende:

De opvolger van BOOT noemt hij NEXTSCHIP.

Als BOOT in de kolk past, dan krijgt dit schip een rangnummer mee, die afhangt van de plaats, die het schip in de kolk moet innemen. Deze BOOT heeft verder alleen nog maar met deze hulp-sluishwachter te maken.

Zijn invaartijd, uitvaartijd en lustijd worden berekend.

De totale hoeveelheid aan tonlaadvermogen van de schepen, dat de kolk verwerkt, neemt toe met het laadvermogen van BOOT.

Nu geeft hij het schip een seintje om binnen te varen, wat direct, zonder dat de computer ook maar iets anders nog mag doen, uitgevoerd moet worden en de hulpsluiswachter wacht intussen om, nadat het schip is ingevaren weer verder te gaan met het

kolkindelen.

Als het schip niet past, wordt gekeken of zijn opvolger past enz. Als er geen schepen meer zijn, of ze passen niet meer, dan wordt ten behoeve van de uitvoer de kolkcapaciteit gesommeerd. De hulpsluiswachter zet de kolk om, wat weer de bedieningstijd van de kolk in beslag neemt, terwijl zijn deuren daarna aan de andere zijde geopend zijn.

Hij geeft het schip, dat vooraan in de sluis ligt een seintje om uit te varen en wacht af totdat hij weer aan het werk gezet wordt. Dan begint hij van voren af aan.

In dit proces wordt op zes plaatsen naar een procedure verwezen:

- a. WEINIGSCHEPENAANWEZIG : PROC. RETURNS
- b. VERANDERDEVOLGORDE : PROC.
- c. SCHIPPAST : PROC. RETURNS
- d. INVAART : PROC. RETURNS
- e. LUST : PROC. RETURNS
- f. UITVAART : PROC. RETURNS

De procedures Returns zijn zogenaamde functieprocedures. Dit betekent, dat de naam van de procedure de uiteindelijke berekende waarde in die procedure krijgt.

De procedures zijn niet in het proces geplaatst vanwege de overzichtelijkheid van het proces.

Deze procedures worden in de volgende paragraaf besproken.

Dit was het proces van de hulpsluiswachters.

Nu volgt het proces van het passeren van de sluis:



```
/*PROCES VAN HET PASSEREN VAN EEN SLUIS*/
297 1 0 AANKOMST:
      IF SLUISNO>25 THEN
298 1 0 DO;CALL HOLD((KM(VAN,VIA)/SNELHEID)*60);
300 1 1 GOTO SYSTEEMDOORGEVAREN;
301 1 1 END;
302 1 0 CALL HOLD((KANAALLENGTE(SLUISNO,RICHTING)/SNELHEID)*60);
304 1 0 CALL TOTAILOFQUEUE(RY(SLUISNO,RICHTING));
305 1 0 AANKOMSTTYD=NOW;
306 1 0 TON(SLUISNO,RICHTING)=TON(SLUISNO,RICHTING)+TONNAGE;
307 1 0 IF HULPSLUISWACHTERINKANTOOR THEN
      CALL REACTIVATE(HULPSLUISW,AFTER,CURRENT);
308 1 0 CALL PASSIVATE;
310 1 0 INVAREN:
      TON(SLUISNO,RICHTING)=TON(SLUISNO,RICHTING)-TONNAGE;
311 1 0 IF AANKOMSTTYD>NOW THEN AANKOMSTTYD=AANKOMSTTYD-SIMULATIETYD;
312 1 0 CALL HISTPUT(WACHTTYD(SLUISNO),NOW-AANKOMSTTYD);
313 1 0 CALL OUTFQUEUE;
314 1 0 IF LUSSCHIP THEN DO;
315 1 1 INVAARTYD=LUSTYD;LUSSCHIP="0*B";END;
318 1 0 CALL HOLD(INVAARTYD);
320 1 0 INSLUIS:
      CALL SORTINQUEUE(HULPSLUISW->SLUIS,VOLGN);
321 1 0 CALL REACTIVATE(HULPSLUISW);
322 1 0 CALL PASSIVATE;
324 1 0 UITVAREN:
      CALL OUTFQUEUE;
325 1 0 CALL HOLD(UITVAARTYD);
327 1 0 IF LENGTH(HULPSLUISW->SLUIS)=0 THEN
328 1 0 CALL REACTIVATE(HULPSLUISW);ELSE
      CALL REACTIVATE(FIRSTOFQUEUE(HULPSLUISW->SLUIS));
329 1 0 CALL HOLD((KANAALLENGTE(SLUISNO,3-RICTING)/SNELHEID)*60);
331 1 0 WYZER=WYZER+1;
332 1 0 IF ROUTE(WYZER)=0 THEN GOTO SYSTEEMDOORGEVAREN;ELSE
334 1 0 DO;RICHTING=1;IF ROUTE(WYZER)<0 THEN RICHTING=2;PRIOR=0;
337 1 1 SLUISNO=ABS(ROUTE(WYZER));GOTO AANKOMST;
339 1 1 END;
340 1 0 SYSTEEMDOORGEVAREN:
      CALL HISTPUT(DOORLTYD(VAN,VIA),NOW-ARRIVALTIME);
341 1 0 FREE SCHIPPARL;
342 1 0 CALL TERMINATE;
/*EINDE VAN HET PROCES VAN HET PASSEREN VAN EEN SLUIS*/
```

Een schip komt aan bij aankomst.

Als er geen sluizen in het kanaal zijn (toetsing: SLUISNO,GR,25) vaart het schip het systeem zonder verder oponthoud door, anders kan het varen tot de eerst volgende sluis.

Komt het aan bij de sluis, sluit het schip aan, achter in de wachtrij. Zijn aankomsttijdstip is op dat moment. Doordat het in de wachtrij komt wordt het totale laadvermogen, wat ligt te wachten verhoogd met dat van het schip.

Als de hulpsluiswachter in het kantoor zit dan wordt hij, zonder dat enig ander proces kan gaan werken, aan het werk gezet en het schip wacht verder af.

Als het mag invaren neemt het tonnage aan schepen weer af in de wachtrij met het tonnage van dat schip. Als intussen de tijd

veranderd is, doordat een volgende simulatie is begonnen wordt dit verdisconteerd. Zijn wachttijd wordt genoteerd in een histogram en hij vaart de wachtrij uit.

Als het schip het eerste schip is, dat mag invaren, is zijn invaartijd gelijk aan de lustijd, anders doet het schip er de invaarvolgtijd over.

In de sluis krijgt het schip het door de hulpsluiswachter aangewezen plaats en het schip meldt de hulpsluiswachter, dat het er ligt en het schip wacht verder af.

Bij het uitvaren vaart het uit de sluis en doet over dit uitvaren de uitvaartijd. Als het het laatste schip is waarschuwt het de hulpsluiswachter anders het volgende schip om dat te laten uitvaren.

De rest van het proces is bestemd voor de kanaal- en netwerk-simulatie, wat in het volgende hoofdstuk besproken zal worden.

procedures:

HULPSLUISWACHTERINKANTOOR: PROC. RETURNS.

Om programmeertechnische redenen wordt de hulpsluiswachter gewaarschuwd om een volgend schip te laten invaren. Dit is uiteraard gestileerd. De hulpsluiswachter zal in de praktijk niet een schip een seintje geven om binnen te varen. Het schip volgt automatisch op zijn voorganger.

Dat volgende schip komt zijn invaarvolgtijd later aan in de sluis. In het model is dit ook het geval.

2. Procedures en Functieprocedures, die de processen complementeren.

2.1. SCHIPPAST: PROCEDURE RETURNS(BIT(1));

```
385 1 0 SCHIPPAST: PROCEDURE RETURNS(BIT(1));
386 2 0 P=1;
387 2 0 PASTSCHIP:
      IF RAAI(P,1)<BOOT->BREEDTE THEN
388 2 0 DO;P=P+1;GOTO PASTSCHIP;END;
391 2 0 IF RAAI(P,2)<BOOT->LENGTE THEN
      RETURN('0'B);
392 2 0 X=0;
393 2 0 GROOTSTERAAI:
      IF RAAI(P+X,1)~=0 THEN
394 2 0 DO;IF RAAI(P+X,2)<RAAI(P,2)-BOOT->LENGTE THEN
395 2 1 DO;X=X+1;GOTO GROOTSTERAAI;
397 2 2 END;
398 2 1 IF RAAI(P+X,1)<BOOT->BREEDTE THEN
399 2 1 DO;P=P+1;GOTO PASTSCHIP;
401 2 2 END;
402 2 1 X=X+1;GOTO GROOTSTERAAI;
404 2 1 END;
405 2 0 SCHIPPASTWEL:
406 2 0 X=X-1;IF RAAI(P+X,2)<RAAI(P,2)-BOOT->LENGTE THEN
407 2 0 DO;RAAI(P+X+1,1)=RAAI(P+X,1);
408 2 1 RAAI(P+X+1,2)=RAAI(P+X,2);
409 2 1 GOTO SCHIPPASTWEL;
410 2 1 END;
411 2 0 RAAI(P+X+1,2)=RAAI(P,2)-BOOT->LENGTE;
412 2 0 RAAI(P+X+1,1)=RAAI(P+X,1);
413 2 0 DO WHILE(X<0);
414 2 1 RAAI(P+X,1)=RAAI(P+X,1)-BOOT->BREEDTE;
415 2 1 X=X-1;
416 2 1 END;RETURN('1'B);
418 2 0 END SCHIPPAST;
```

Deze procedure zoekt uit of een schip nog in de sluis past en zo ja, welke plaats moet dit schip dan innemen.

Alvorens hierop in te gaan moet eerst iets over het kolk-indelen gezegd worden.

Als een schip niet past is dat vrij eenvoudig te constateren.

Verder worden de schepen zodanig in de kolk gepast, dat een optimale hoeveelheid aan schepen in de kolk kunnen. Een gedragsregel hierbij is, dat grotere schepen het eerst invaren, daarna de kleinere.

De volgorde van invaren hiervoor wordt vastgesteld in de procedure VERANDERDEVOLGORDE.

Het principe is als volgt:

Een schip, dat past vaart in en vaart zover mogelijk naar voren. Achter het schip wordt een raai gedefinieerd, waarvan de breedte wordt bepaald terwijl bovendien dit schip één of meerdere breedtes van raaien kan veranderen.

Hoe dit in zijn werk gaat kan het beste worden uitgelgd aan de hand van het volgende voorbeeld:

Stel er liggen zes schepen voor een sluis, die uit één kolk bestaat. De kolk is 200 m lang en 18 m breed.

Volgorde van schepen na het veranderen van de volgorde: klasse 6, kl 5, kl 3, kl 1, kl 1, kl 6,

U ziet dat tot een bepaalde lengte gesorteerd is.

Schip kl 6 vaart in: 95 m l, 11.5 m b.

Schip kl 5 vaart in: 80 m l, 9.5 m b.

Dit schip past niet naast het eerste schip dus gaat het er achter liggen.

Schip kl 3 vaart in: 50 m l, 6.6 m b.

Dit schip past niet naast het schip kl 6 maar wel naast het schip kl 5.

Schip kl 1 vaart in: 25 m l, 4.6 m b.

Dit schip had nog met gemak naast schip kl 6 gepast, maar zijn doorvaart naar deze plaats wordt het schip belemmerd door schip kl 3. In de praktijk had de hulp-sluiswachter dit anders opgelost:

eerst de schepen, die nog naast het schip kl 6 passen laten invaren, daarna pas het schip kl 3. Het model doet net of schip kl 3 er nog niet ligt en laat de betreffende schepen gewoon invaren.

Aan het kolkindelen doet dit niets af, wel aan de volgorde waarin de schepen uit de wachtrij gehaald worden. Naar verwachting zullen de consequenties hiervan niet groot zijn, zoals bij de proefnemingen met het model is gebleken. Toch blijft dit nog een studieobject.

Schip kl 1 vaart in: 25 m l, 4.6 m b.

Voor dit schip geldt bovenstaande relaas ook.

Schip kl 6, wat nu aan de beurt is, past niet meer in de kolk en moet overliggen.

Terwijl deze schepen aan het invaren waren, kwamen nog twee schepen bij de sluis aan: kl 6 en kl 3.

Voor het schip kl 6 is geen plaats meer, maar het schip kl 3 past nog wel in de kolk, naast schip kl 5 en achter

schip kl 3.

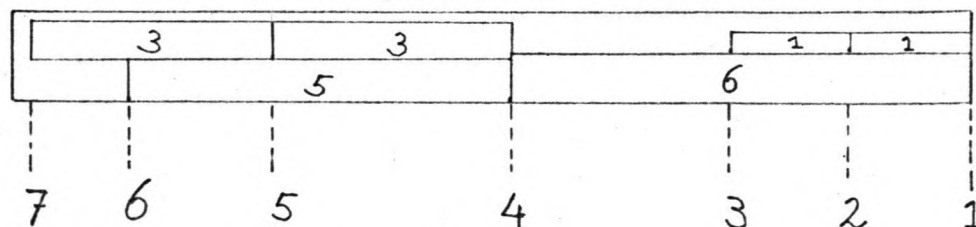
Dit schip vaart natuurlijk ook nog in.

De raainummers worden steeds weer zo aangepast dat zij van de uitvaardeuren opklimmen naar de invaardeuren.

Deze methode is op veel simulaties toegepast en blijkt redelijk te werken. Nauwkeuriger zou zijn, als de coördinaten werden vastgelegd, maar dit benadert nauwelijks beter de methode zoals die hierboven is beschreven, terwijl wel op grotere problemen gerekent kan worden, vooral daar waar het het optimaliseren van de kolkindeling aangaat. Deze methode werkt ongeveer even optimaal als wat in verschillende waarnemingen bij sluizen is waargenomen. Een stroomdiagram van deze procedure is bijgevoegd in hoofdstuk

Waarnemingen, die bij de Volkeraksluizen gedaan zijn, wat betreft de invaarvolgtijd, uitvaarvolgtijd en lustijd, alsmede het indelen van de kolk zijn in bijlage bijgesloten.

Hier vindt u een schets van de kolkindeling van bovenvermeld scheepsaanbod:



Een groot voordeel van de procesbeschrijvingsmethode is, dat de kolk niet in één keer gevuld wordt met de schepen, die ervoor liggen, maar dat het proces van het binnenvaren steeds weer wordt opgeschort totdat het schip is binnengevaren. Schepen, die tijdens dit vullen nog aankomen,



kunnen dan ook nog aan bod komen.

Wordt daarentegen met de event-beschrijvings methode of de activiteitenmethode gewerkt, worden de problemen zo enorm gecompliceerd, dat er nauwelijks een computerprogramma gemaakt kan worden, die deze methoden gebruikt, die in staat zal zijn zo'n simulatie uit te voeren.

## 2.2. INVAART: PROCEDURE RETURNS(FLOAT);

```
327 1 0 INVAART: PROCEDURE RETURNS(FLOAT);
328 2 0 Q=(SLUISBREEDTE*KOLK DIEPTE(BOOT->RICHTING))/100;
329 2 0 T=INV(BOOT->KLASSE,KV,BOOT->GELADEN,3)*(Q**2)+
      INV(BOOT->KLASSE,KV,BOOT->GELADEN,2)*Q+
      INV(BOOT->KLASSE,KV,BOOT->GELADEN,1);
330 2 0 RETURN(T);
331 2 0 END INVAART;
```

Om de individuele invaarvolgtijd van een schip vast te stellen is gebruik gemaakt van de tabellen: bijlage 2-7.

Deze tabellen werden omgevormd van haar oorspronkelijke functie tot tabellen, die geschikt zijn voor het bepalen van de individuele sluistijden van schepen.

Voor een aantal series willekeurig gekozen schepen is getoetst in hoeverre dit mogelijk was, waaruit bleek, dat de fout, die gemaakt wordt, gering is bij het anders interpreteren van de tabellen.

Voor het gebruik van deze tabellen op de huidige manier moet eerst bekend zijn, wat het scheepsaanbod is. Daar dit in de werkelijkheid nooit het geval is, bepaalt dit model eerst op het moment van binnenvaren, wat de invaarvolgtijd is van dat schip.

Stel het tonnage van dit schip is 600 ton, KV=1, het schip is geladen: Voor dit schip is de invaarvolgtijd volgens tabel 4:

$(F=60 \text{ m}^2)$  2.8 minuten.

Zo wordt van alle schepen de invaarvolgtijd bepaald. Met deze methode is het dus niet nodig van te voren op de hoogte te zijn van het scheepsaanbod.



Een controle achteraf of deze methode ten opzichte van de methode, waarvoor de tabellen oorspronkelijk zijn samengesteld geeft nagenoeg hetzelfde resultaat. Dit laatste is getoetst aan een aantal handsimulaties en daarna aan de VERSION-TRACE van het PROSIM-programma.

In de oorspronkelijke capaciteitsberekeningen werd verondersteld, dat de lijnen, die weergegeven worden in de tabellen een hyperbolisch verloop hebben, deze veronderstelling introduceert een vrij grote fout.

In dit programma zijn deze lijnen dan ook als polynomen bepaald, waarbij de factoren van de machtscomponenten vermeld staan in het array INV.

De invaarvolgtijd is afhankelijk van:

- a. breedte van de invaardeuren.
- b. drempeldiepte van de invaardeuren.
- c. scheepsklasse.
- d. kolkvoorhavengetal.
- e. beladingsgraad.

Hoe deze variabelen afhankelijk zijn blijkt uit de formule, omschreven in de procedure.

De constante factoren van de polynomen zijn overigens bepaald met de standaardprocedure KLEPOL.

### 2.3. UITVAAT: PROCEDURE RETURNS(FLOAT);

```
332 1 0  UITVAAT: PROCEDURE RETURNS(FLOAT);
333 2 0  Q=(SLUISBREEDTE*KOLKDIEPTE(B-BOOT->RICHTING))/100;
334 2 0  T=UITV(BOOT->KLASSE,KV,BOOT->GELADEN,3)*(Q**2)+
        UITV(BOOT->KLASSE,KV,BOOT->GELADEN,2)*Q+
        UITV(BOOT->KLASSE,KV,BOOT->GELADEN,1);
335 2 0  RETURN(T);
336 2 0  END UITVAAT;
```

Deze procedure is op dezelfde wijze tot stand gekomen als de procedure INVAAAT.

#### 2.4. LUST: PROCEDURE RETURNS(FLOAT);

```
337 1 0 LUST: PROCEDURE RETURNS(FLOAT);
338 2 0 U=BOOT->KLASSE;V=BOOT->GELADEN;W=BOOT->BREEDTE;
341 2 0 T=((L(U,V,1)-L(U,V,3))*(W/SLUISBREEDTE)+L(U,V,3))+(((L(U,V,2)-
L(U,V,4))*(W/SLUISBREEDTE)+L(U,V,4))*LUS);
342 2 0 RETURN(T);
343 2 0 END LUST;
```

De lustijd van een schip is eveneens op een soortgelijke manier tot stand gekomen, met dit verschil, dat de lijnen niet als polynomen berekend zijn, maar als "tangensvlakken" aan de hand van vier punten (A,B,C,D). De lustijd wordt verondersteld voor  $b/B$  lineair te zijn op de lijn  $A_1=0$ , Voorts wordt verondersteld, dat de toe of afname van de tangens behorend bij elk van de punten op de lijn  $A_1=0$  lineair is.

Er zijn niet voldoende metingen verricht om aan te tonen, dat ook deze lijnen voor deze benaderingswijze volledig geschikt zijn, hoewel verwacht mag worden, dat de resultaten een fout zullen geven in de orde van grootte van de fouten, die gevonden zijn voor de inwaar- en uitvaarvolgtijden. Mochten de fouten bij verdere experimenten onaanvaardbare vormen gaan aannemen, behoeft slechts deze procedure gewijzigd te worden.

#### 2.5. WEINIGSCHEPENAAANWEZIG: PROCEDURE RETURNS(BIT(f));

```
436 1 0 WEINIGSCHEPENAAANWEZIG: PROCEDURE RETURNS(BIT(1));
437 2 0 IF TON(SCHUTSLNO,ZYDE)->CAP(SCHUTSLNO)-KOLK THEN
438 2 0 DO;IF TON(SCHUTSLNO,3-ZYDE)->CAP(SCHUTSLNO)-KOLK THEN
439 2 1 DO;CAP(SCHUTSLNO)=CAP(SCHUTSLNO)-KOLK;
440 2 2 IF CAPACITEITSBEPALING THEN DO;
441 2 3 IF LENGTH(RY(SCHUTSLNO,ZYDE))<MEANOF(SLUISTRY) THEN DO;
442 2 4 CALL CANCEL(MAIN);CALL REACTIVATE(MAIN);END;END;
446 2 2 RETURN('1'B);
447 2 2 END;
448 2 1 END;
449 2 0 RETURN('0'B);
450 2 0 END WEINIGSCHEPENAAANWEZIG;
```

Deze procedure is eigenlijk een toetsing van de hulpsluiswachter of hij, gezien de omstandigheden nog moet werken of dat hij naar het kantoor kan gaan.

De procedure werkt als volgt:

Stel we hebben een sluis met drie kolken.

Er zijn nog geen schepen bij de sluis, dus het tonnage aan schepen, dat te wachten ligt, is 0. De capaciteit van de sluis is de capaciteit van de drie kolken samen. Als nu de eerste hulpsluiswachter arriveert, kijkt hij of er schepen zijn. Zo niet, dan gaat hij naar het kantoor en wordt de capaciteit van sluis verminderd met de capaciteit van die kolk. Dit geldt ook voor de tweede en de derde hulpsluiswachter.

De capaciteit van de sluis is nu 0.

Nu komt er een schip van een kant, die een hulpsluiswachter reactiveert. Deze hulpsluiswachter werkt zijn proces verder door en komt dan tenslotte weer aan bij deze procedure. Stel dat er nu geen schepen meer liggen aan de zijde waar de deuren nu open staan. De tonnage is dan niet groter dan de capaciteit.

Nu toetst de hulpsluiswachter of er misschien schepen aan de andere zijde liggen. Is dit het geval, dan zet de hulpsluiswachter de kolk om en gaat deze boten verschutten. Als er geen schepen meer zijn, wordt de capaciteit van de sluis weer 0 en de hulpsluiswachter gaat naar het kantoor. Stel nu dat er zoveel schepen zijn, dat al de drie hulpsluiswachters werken.

Op een bepaald moment bemerkt, tijdens het toetsen met deze procedure, de hulpsluiswachter dat het tonnage aan schepen aan de zijde, waar zijn kolk staat, minder geworden is dan de capaciteit per schutting van de sluis minus die van zijn kolk. Dit betekent, dat de schepen bij de eerst volgende schuttingen in de twee andere kolken passen. Deze hulpsluiswachter gaat terug naar het kantoor, tenminste, als het aan de andere zijde ook niet druk is.

## 2.6. HULPSLUISWACHTERINKANTOOR: PROCEDURE RETURNS(BIT(1));

```
451 1 0 HULPSLUISWACHTERINKANTOOR: PROCEDURE RETURNS(BIT(1));
452 2 0 IF LENGTH(KANTOOR(SLUISNO))=0 THEN RETURN('0'B);
453 2 0 IF TON(SLUISNO,RICHTING)<CAP(SLUISNO) THEN RETURN('0'B);
454 2 0 HULPSLUIW=LASTOFQUEUE(KANTOOR(SLUISNO));
455 2 0 BESTEZOEKEN:
      IF HULPSLUIW->ZYDE=RICHTING THEN
456 2 0 DO;CAP(SLUISNO)=CAP(SLUISNO)+HULPSLUIW->KOLK;
457 2 1 RETURN('1'B);
458 2 1 END;
459 2 0 HULPSLUIW=PRED(HULPSLUIW);
460 2 0 IF HULPSLUIW=NULL THEN
461 2 0 DO;HULPSLUIW=LASTOFQUEUE(KANTOOR(SLUISNO));
462 2 1 CAP(SLUISNO)=CAP(SLUISNO)+HULPSLUIW->KOLK;
463 2 1 RETURN('1'B);
464 2 1 END;
465 2 0 GOTO BESTEZOEKEN;
466 2 0 END HULPSLUISWACHTERINKANTOOR;
```

Een schip komt bij de sluis aan.

Als alle hulpsluiswachters werken, hoeft het schip niet anders te doen, dan achter in de wachtrij aan te sluiten. Doet nog één of meerdere hulpsluiswachters niets, dan wordt er gekeken of het de moeite waard wordt om een hulpsluiswachter te waarschuwen.

Mocht dit inderdaad het geval zijn, wordt de hulpsluiswachter, die een kolk bedient, die bij een drukker aanbod gaat werken, gewaarschuwd. Als dan ook nog zijn kolk aan de goede zijde staat, wordt de capaciteit van de sluis verhoogd met die van de kolk, anders wordt er eerst gekeken of er een andere kolk al wel goed staat. Zo ja, dan gaat die kolk werken. Mocht er geen enkele kolk goed staan, gaat de eerst genoemde hulpsluiswachter toch werken.

## 2.7. VERANDERDEVOLGORDE: PROCEDURE;

```
467 1 0 VERANDERDEVOLGORDE: PROCEDURE;
468 2 0 TONTOT=0;
469 2 0 BOOT=FIRSTOFQUEUE(RY(SCHUTSLNO,ZYDE));
470 2 0 DO WHILE (TONTOT<0.8*KOLK & BOOT->NULL);
471 2 1 NEXTSCHIP=SUC(BOOT);
472 2 1 CALL CUTOFFQUEUE(BOOT);
473 2 1 TONTOT=TONTOT+BOOT->TONNAGE;
474 2 1 BOOT->VOORRANG=-BOOT->KLASSE;
475 2 1 IF BOOT->PRIOR>0 THEN BOOT->VOORRANG=- (TOTKLASSEN+BOOT->PRIOR);
476 2 1 CALL SORTINQUEUE(BOOT,INVAARVOLGORDE,BOOT->VOORRANG);
477 2 1 BOOT=NEXTSCHIP;
478 2 1 END;
479 2 0 DO WHILE (LENGTH(INVAARVOLGORDE)~0);
480 2 1 BOOT=LASTOFQUEUE(INVAARVOLGORDE);CALL CUTOFFQUEUE(BOOT);
481 2 1 CALL TOHEADOFQUEUE(BOOT,RY(SCHUTSLNO,ZYDE));
482 2 1 BOOT->PRIOR=BOOT->PRIOR+1;
483 2 1 END;
484 2 0 END VERANDERDEVOLGORDE;
```

Het eerste schip uit de wachtrij wordt BOOT genoemd. Als het tonnage van dit schip niet het totale tonnage aan gesorteerde schepen de capaciteit van kolk met een factor 0.8 overschrijdt of als er geen schepen meer liggen te wachten, wordt dit schip uit de wachtrij gehaald en in de wachtrij invaarvolgorde geplaatst volgens een sorteerparameter. De schepen worden naar grootte gesorteerd: De grootste het eerst, de kleinste het laatst. Als het schip één keer gesorteerd is, krijgt het schip prioriteit in verband met het eventueel toch nog niet passen in de sluis. Deze prioriteit zorgt ervoor, dat de sorteerparameter van het schip zo groot wordt, dat het de volgende keer ver vooraan in de wachtrij komt te liggen. Is de rij invaarvolgorde eenmaal samengesteld wordt zij weer in die volgorde in de wachtrij voor de sluis geplaatst en kan het proces van binnenvaren gebeuren.

Dit zijn de gebruikte procedures.

De twee procedures optimaliseren en optimalisatietabel komen ter sprake in deel II van deze studie. Zij zijn overigens alleen geschikt voor het optimaliseren van de routekeuze van de schepen voor wat de kostenminimalisatie en batenmaximalisatie aangaat. Dat model gaat uit van het standpunt van de schippers als ondernemers en niet van de beheerder van de scheepvaartwegen als technicus. De resultaten van de simulatie worden gebruikt door de optimalisatie. Na elke simulatie blijken de kostenmin. en batenmax. te convergeren naar een optimum. Zie voor het gebruik ook de handleiding.

Dit voor wat betreft de sluisanalogie in een computermodel. Het kan in de vrij korte tijdsspanne, waarin dit model tot stand is gekomen, nog verre van volledig zijn. De procesbeschrijvingsmethode heeft echter als nog een



voordeel, dat verdere wijzigingen of nuances gemakkelijk toegevoegd kunnen worden, zonder dat dit de processen op andere plaatsen aantast.

## Hoofdstuk VI

### Netwerk van kanalen en vervolg MAIN-proces.

#### 1. Netwerk van kanalen

Het model is, naast het simuleren van een enkele sluis ook geschikt voor het simuleren van een kanaal of netwerk van kanalen. Dit is enerzijds gedaan om de gebruiksmogelijkheden te vergroten, anderzijds, en een zeer belangrijke rede, om tot een betere simulatie van aankomstenpatronen bij een sluis te komen. Schepen worden in de simulatie gegenereerd op een plaats, waar we kunnen aannemen, dat het aankomstenpatroon POISSON-verdeeld is. Indien nu op een beperkte afstand van de te beschouwen sluis een andere sluis is gelegen, is het aankomstenpatroon beslist niet POISSON-verdeeld meer, we spreken dan van een gedoseerd aanbod.

Indien nu die vorige sluis ook gesimuleerd is, zal het aankomstenpatroon bij de beschouwde sluis de werkelijke aankomsten kunnen benaderen.

In het proces van de hulpsluismeesters is geen uitbreiding noodzakelijk, dan alleen die, welke aangeeft, bij welke sluis de hulpsluismeester in dienst is.

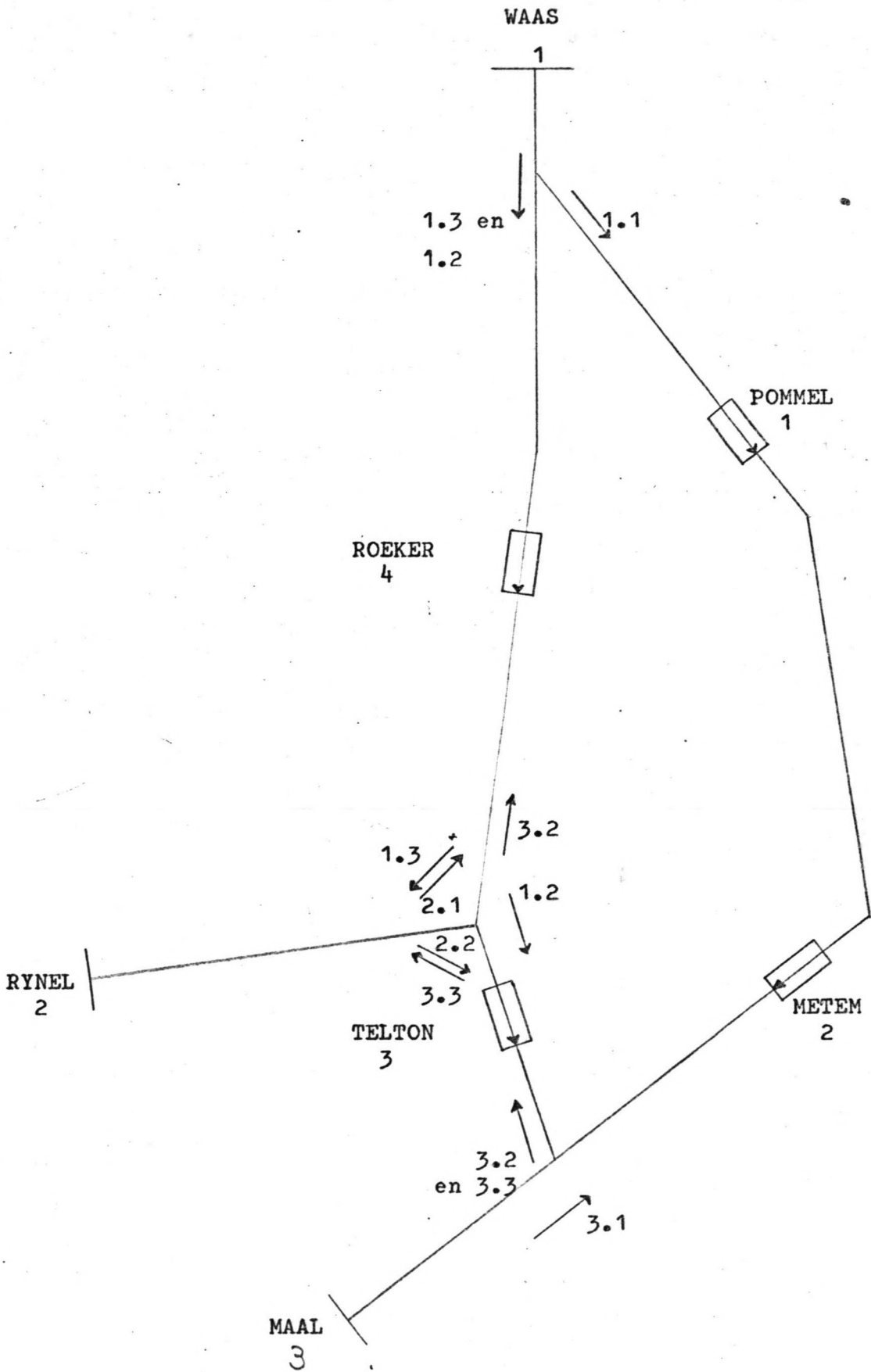
Het proces van passeren van een sluis blijft eveneens gelijk met de uitbreiding dat een schip moet weten bij welke sluis het is.

Deze extra parameters zijn besproken bij de attributen van de componenten hulpsluismeesters en schepen.

Een uitgebreide behandeling van het simuleren van een netwerk is weinig zinvol, temeer omdat bij elke sluis weer eenzelfde proces doorlopen wordt.

Het netwerkanalogie zal dan ook behandeld worden aan de hand van een voorbeeld.

Stel we hebben het volgende netwerk:



- a. Dit netwerk heeft 3 generatoren:

WAAS, RIJNEL en MAAL.

Bij elke generator is het aankomstenpatroon POISSON-verdeeld.

- b. 4 sloopstypen maken van dit netwerk gebruik.

Sloopstypen 2 tot en met 5, elk met hun specifieke attributen.

Voor het model noemen we sloopstypen 2: klasse 1. enz. (Dit komt overigens niet tot uitdrukking in de uitvoer)

Bij elke generator werken dus 4 subgeneratoren.

- c. In het netwerk komen 4 sluisen voor:

POMMEL, METEM, ROEKER, en TELTON.

- d. Voor het werken met het model moeten al deze namen nummers hebben:

WAAS=1; RIJNEL=2; MAAL=3;

POMMEL=1; METEM=2; TELTON=3; ROEKER=4;

Verder moet elke route een nummer hebben.

B.v. van generator 3 naar generator 1 via de sluisen METEM en POMMEL is de route 3.1. enz.

Als de verdeling per sloopstypen per generator over de verschillende routes bekend is, dan worden in die verhouding de schepen over de routes verdeeld.

Dit is gebeurt met een RANDOM-generator.

Stel een schip gaat van generator 1 via route 1.2. naar generator 3, dan krijgt het schip de volgende waarden voor attributen VAN, VIA en BESTEMMING en ROUTE VAN=1, VIA=2, BESTEMMING=3,

ROUTE= 4,3,0,0,...;

- e. Nu moet worden vastgesteld of een schip van de bovenstroomse dan wel van de benedenstroomse zijde een sluis nadert, m.a.w. is RICHTING=1 of is RICHTING=2.

In elke sluis is een pijl gezet. Dit geeft de stroomrichting aan.

Volgt nu een schip van generator 3 de route 3.1, dan zijn zijn attributen:

VAN=3; VIA=1; BESTEMMING=1;

ROUTE= -2,-1,0,0,...;

In het model wordt nu als een schip bij sluis -2 aankomt

de RICHTING=2 en SLUISNO=2.

- f. Om duidelijk te maken hoe een schip zich door het systeem begeeft, zullen we een schip van een generator af volgen. SCHIP 178 wordt gegenereerd op tijdstip 101 bij generator 3. Een randomgenerator geeft een getal, dat aangeeft, dat het schip route 3.1. gaat volgen.

Uit het array routepar blijkt, dat deze route 3.1. bestaat uit het array ROUTE=-2,-1,0,0,...;

In het proces van de generator van dit schip wordt vastgesteld dat RICHTING=2 en SLUISNO=2.

Het schip komt in zijn proces bij de label AANKOMST.

Het vaart tot sluis nummer 2 en gaat in de wachtrij.

Als het na verloop van tijd weer uitgevaren is, vaart het eerst nog door het kanaal, dat bij deze sluis hoort.

Op deze plaats kan het best het vervolg van het proces van een schip

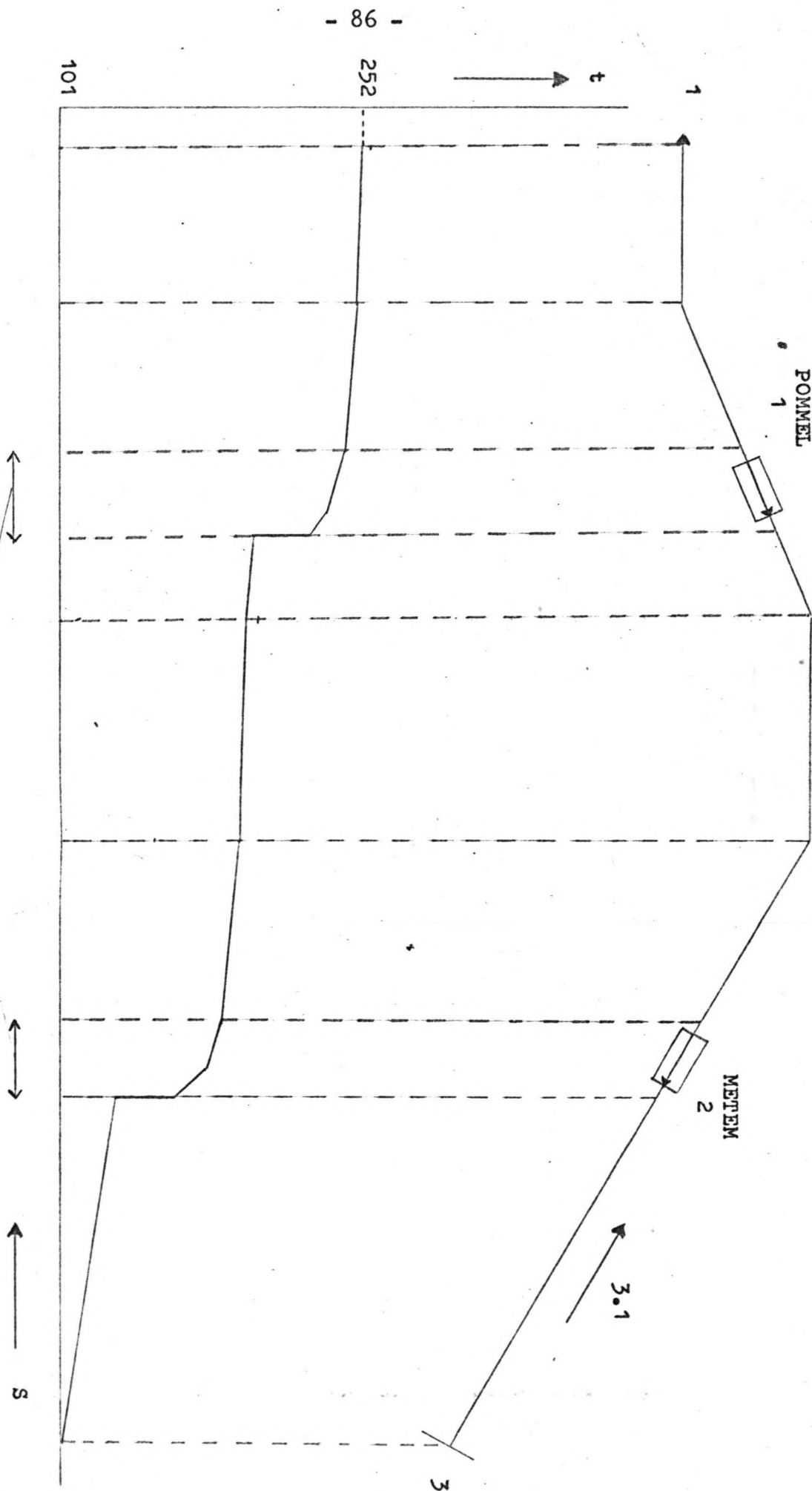
WYZER=1 (volgens de generator) WYZER wordt nu 2.; ROUTE(2) is dus het tweede element van het array ROUTE. Dit element is -1. Nu wordt RICHTING=2 en SLUISNO=1, en het schip begint van voren afaan bij de label aankomst. Komt het de volgende keer weer bij de opdracht WYZER=WYZER+1 dan is ROUTE(WYZER)=0 en het schip is het systeem doorgevaren.

In een histogram wordt deze doorvaartijd genoteerd en het schip wordt getermineerd, terwijl de geheugenruimte, die het in beslag nam, weer vrijgegeven wordt.

Een schematische gang van zaken is op de volgende bladzijde geschetst.



S-t diagram van het Schip 178 :



Als nu de prijzen van de doorvaartijden en kanaalgebruik bekend zijn, kan uit de resultaten van de simulatie van dit netwerk bepaald worden, welke route een schip het voordeligst kan kiezen. Het kiezen van een route gebeurt door een RANDOM-trekking uit verdelingen van de routekeuzes.

Door optimalisatie worden de verdelingen van routekeuzes steeds na verloop van tijd herzien. Door nu de simulatietijd groot genoeg te maken blijkt deze optimalisatie te convergeren. \* Hoe deze optimalisatie werkt wordt uiteengezet in deel II van deze studie.

## 2. Het hoofdproces (vervolg).

Voor een bepaalde tijd worden schepen door het systeem geleid (simulatietijd). Is nu deze simulatietijd verstreken, dan willen we op dat moment de uitvoergegevens hebben. Deze uitvoergegevens bestaan uit een "foto" van het systeem ten tijde "nu".

(CALL STATUS) en een aantal histogrammen met betrekking tot de wacht- en doorlooptijden, verder worden een aantal parameters weer in hun oorspronkelijke stand gezet. In de versie, die op het tijdstip van het schrijven van dit verslag gereed is, van het model worden de hulpsluismeesters voor een bepaalde tijd opgehouden, voordat zij aan het werk gaan. Hierdoor ontstaan lange wachtrijen voor de sluizen, zodat in de eerste run de intensiteit groter is dan de capaciteit, zodat deze uit het aanbod bepaald kan worden.

Mochten er nog te veel schepen in de wachtrijen liggen op het moment dat de tweede run van start gaat, worden deze verwijderd uit het systeem.

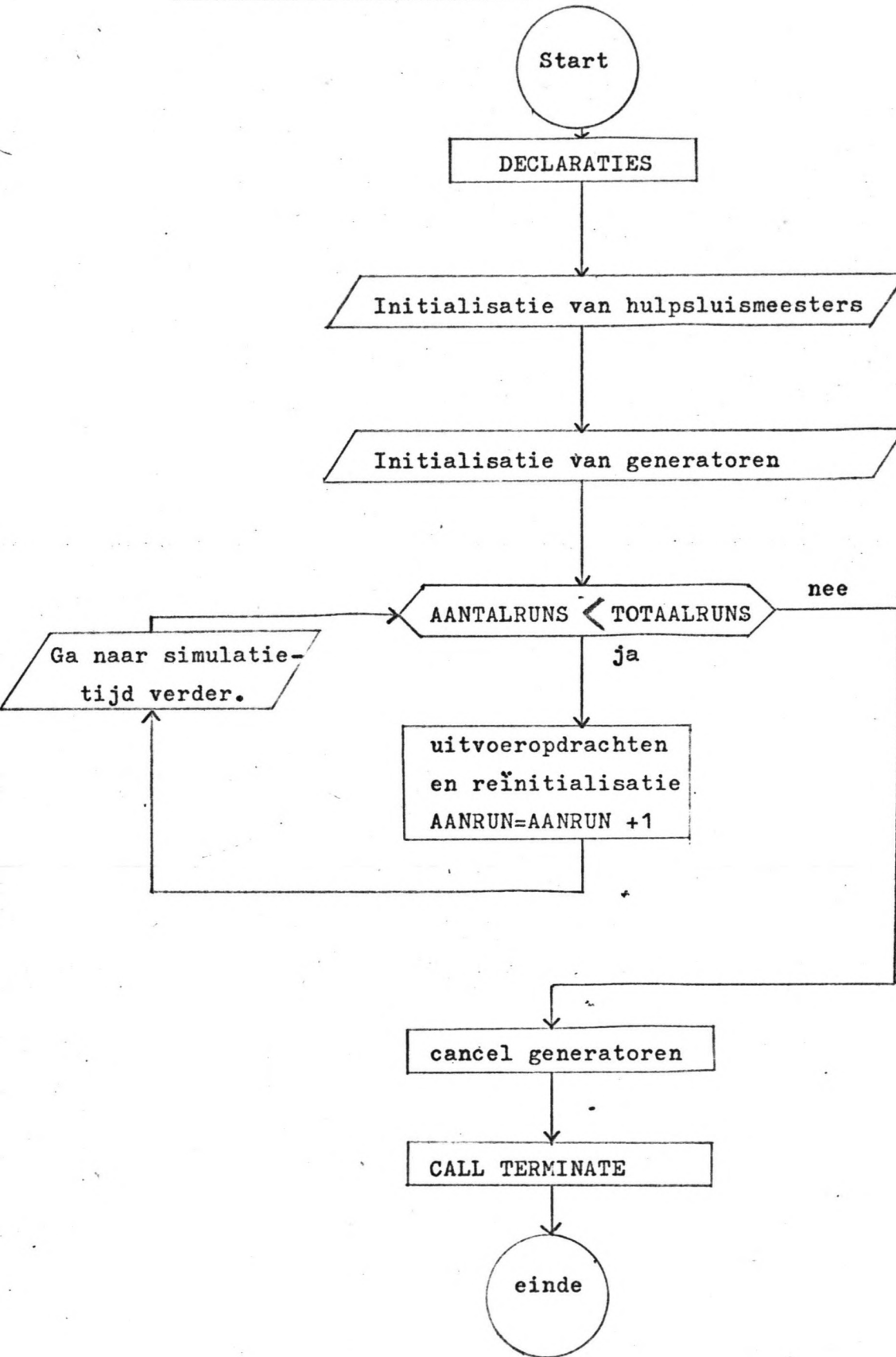
Hebben nu alle runs plaatsgevonden, dan worden alle generatoren gecancelled en de schepen, die nog aanwezig zijn in het systeem, varen dit systeem nog door, waarna de simulatie ophoudt omdat er geen enkele component meer geactiveerd kan worden (de future event list geeft geen enkele eventtijdstip meer aan).

Het bijbehorende programma gedeelte vindt u vanaf opdracht 155 in het model (blad 107).

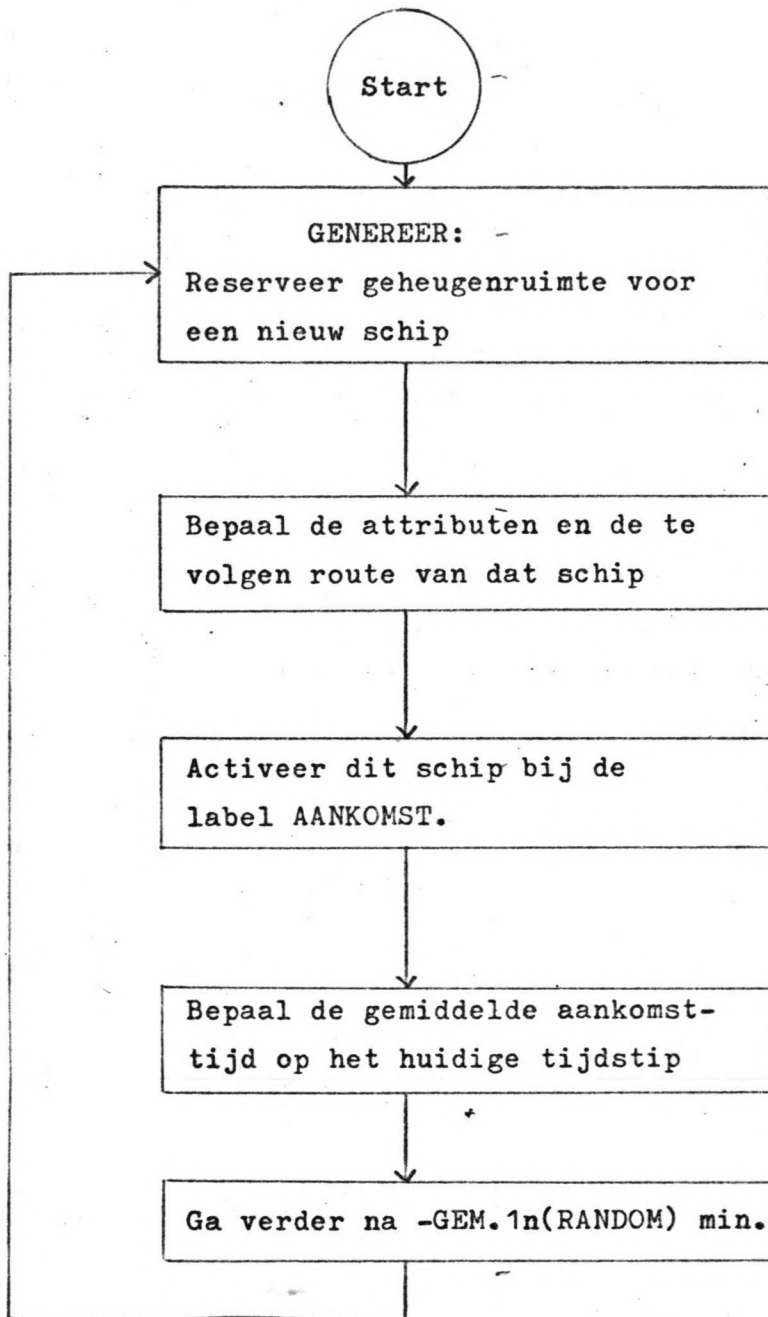
In het volgende hoofdstuk zijn de stroomdiagrammen van alle onderdelen, waaruit het programma bestaat weergegeven.

Hierna vindt u het model, zoals dit op het moment van schrijven op een computerschijf is opgeborgen en voor iedereen toegankelijk is met de stuurkaarten, die beschreven zijn in de bijgevoegde handleiding van het model.

STROOMDIAGRAM HOOFDPROCES;



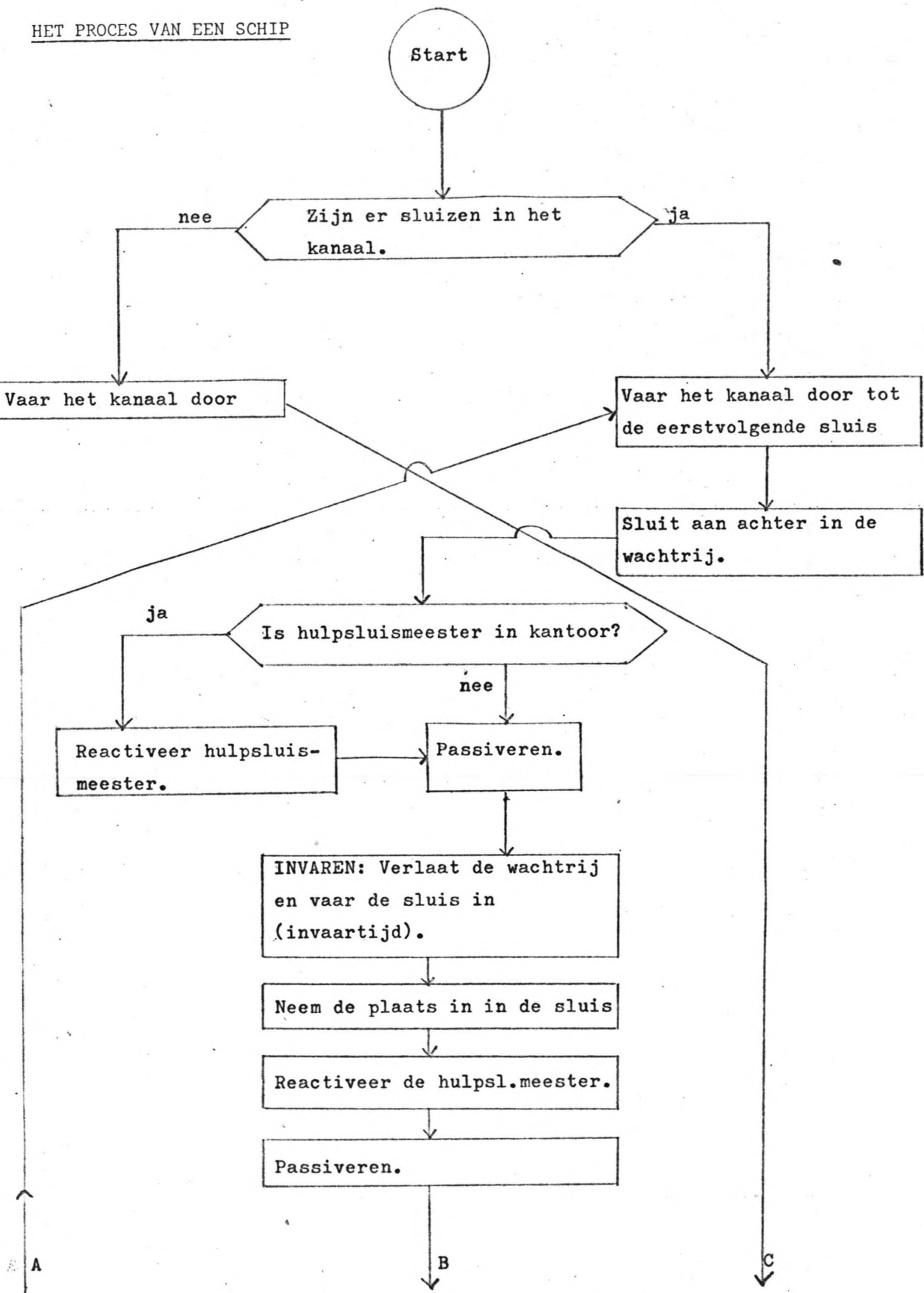
HET PROCES VAN GENEREREN VAN SCHEPEN



P.S. Elke generator wordt door de component MAIN gecancelld, zodat hier geen opdracht voor is in het proces van de generator.



HET PROCES VAN EEN SCHIP



UITVAREN: vaar uit de sluis.

wacht uitvaartijd.

nee

Zijn er nog schepen in de sluis

ja

React. hulpsluismeester

React. volgend schip.

vaar het kanaalgedeelte  
verder door

ja

Zijn er nog meer sluizen te  
passeren

nee

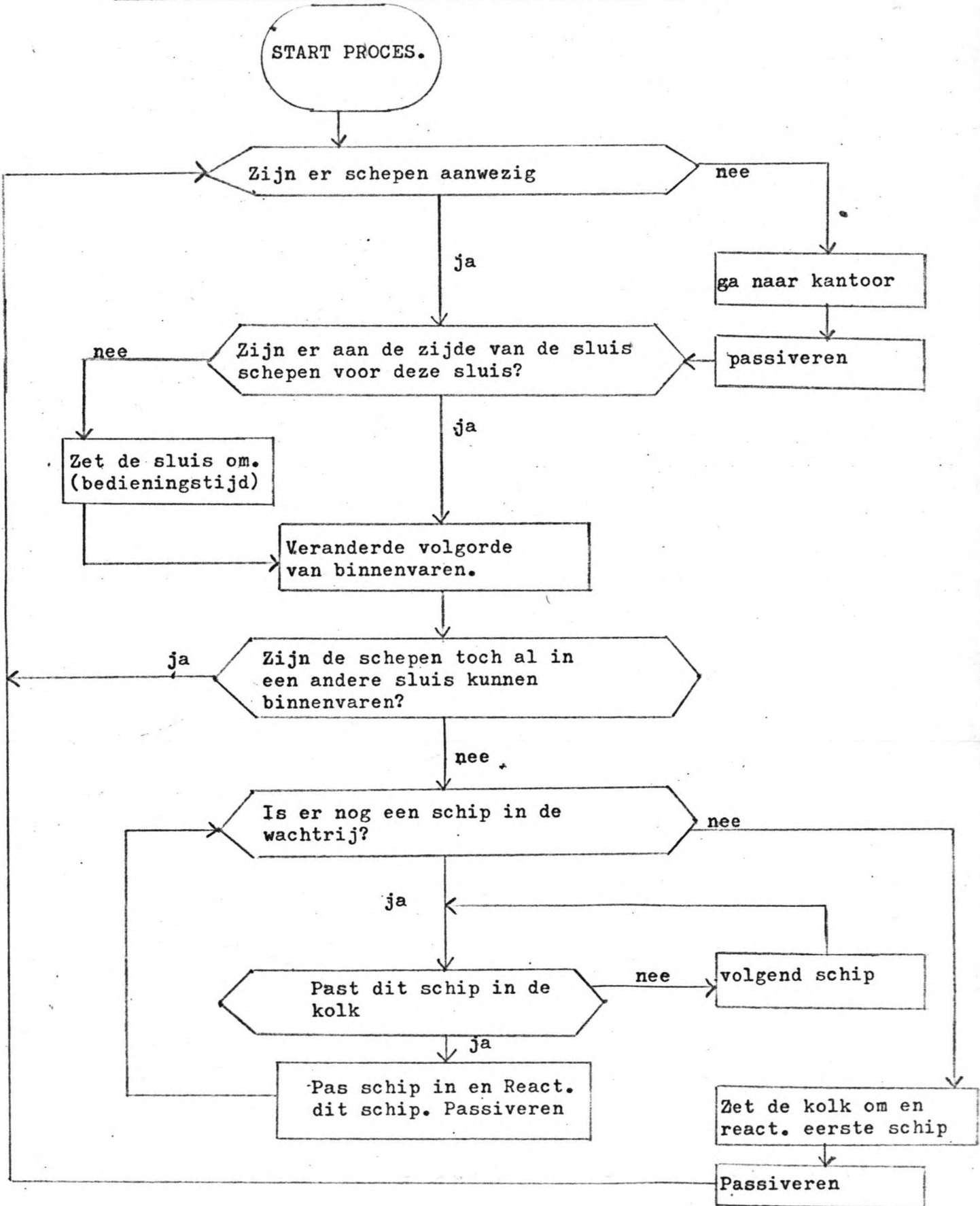
Bepaal welke sluis, rich-  
ting en prioriteit.

Notatie in histogram  
van de doorlooptijden

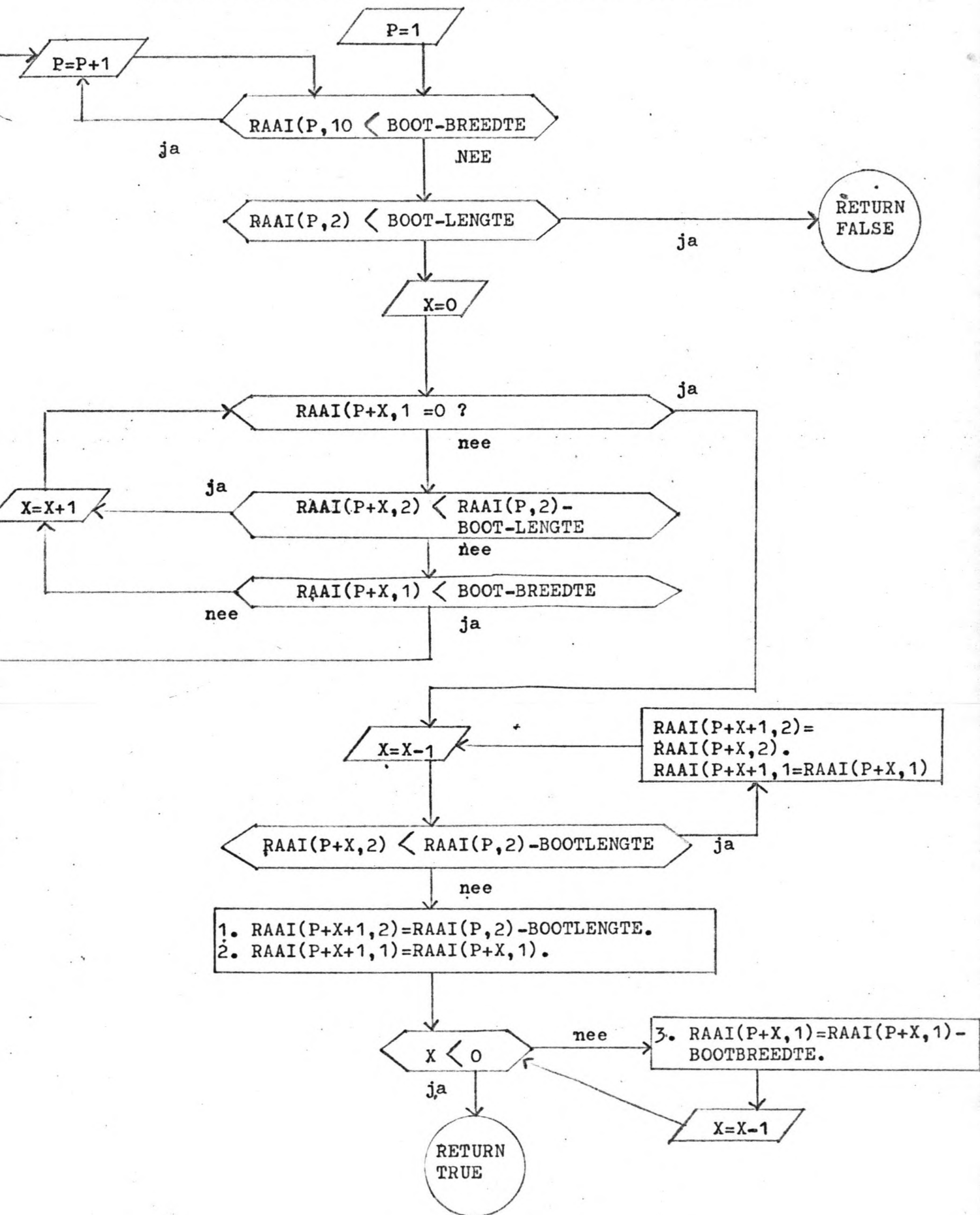
Geef geheugenruimte terug en  
beëindig levensloop schip

Einde proces.

STROOMDIAGRAM VAN HET PROCES VAN DE HULPSLUISMEESTER.

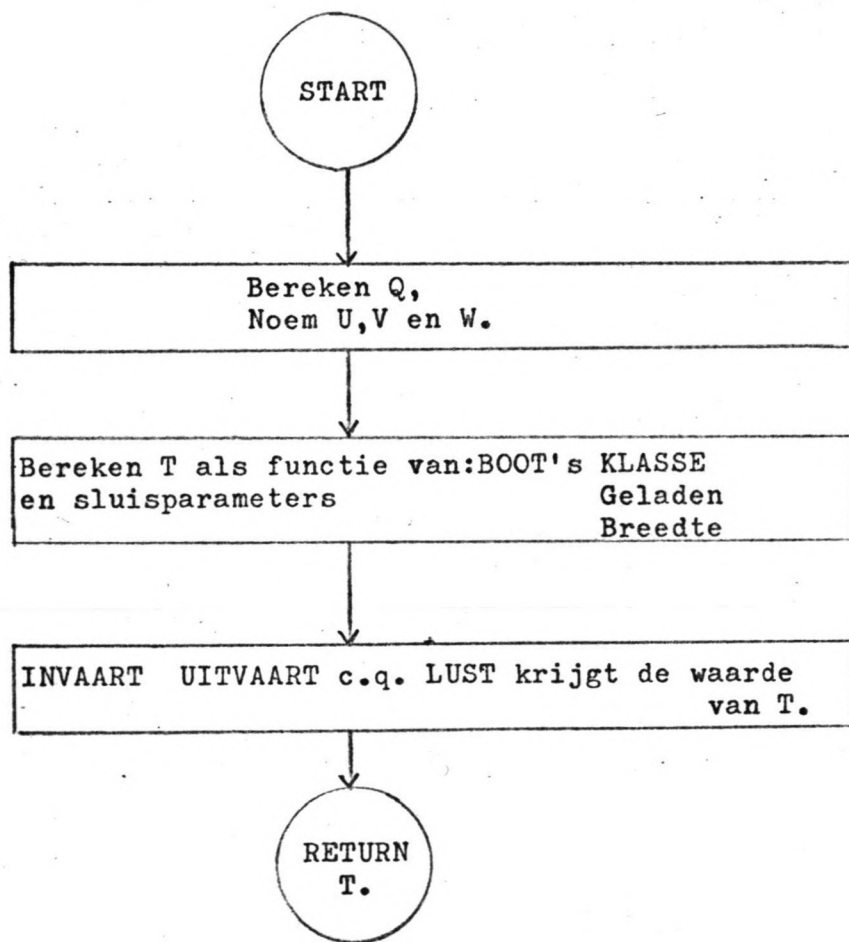


STROOMDIAGRAM: SCHIPPAST: PROCEDURE RETURNS(BIT(1));



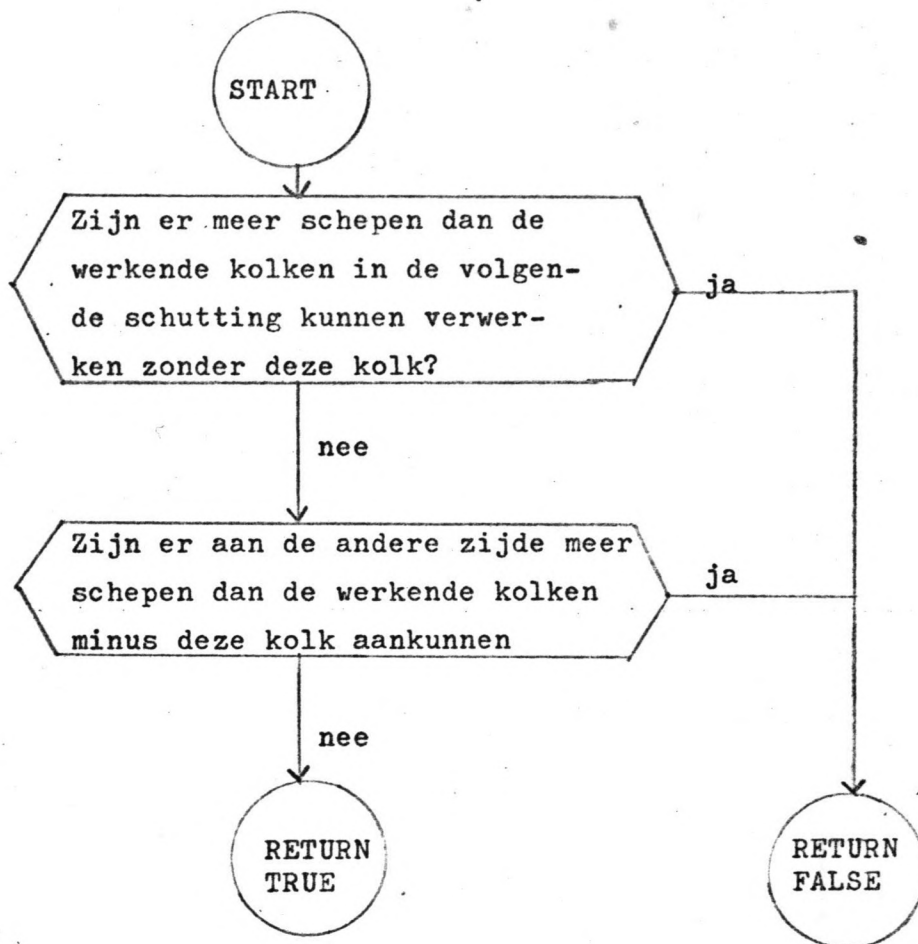
De drie procedures INVAART, UITVAART en LUST hebben min of meer eenzelfde soort structuur.

Voor deze functieprocedures geldt het volgende stroomdiagram:

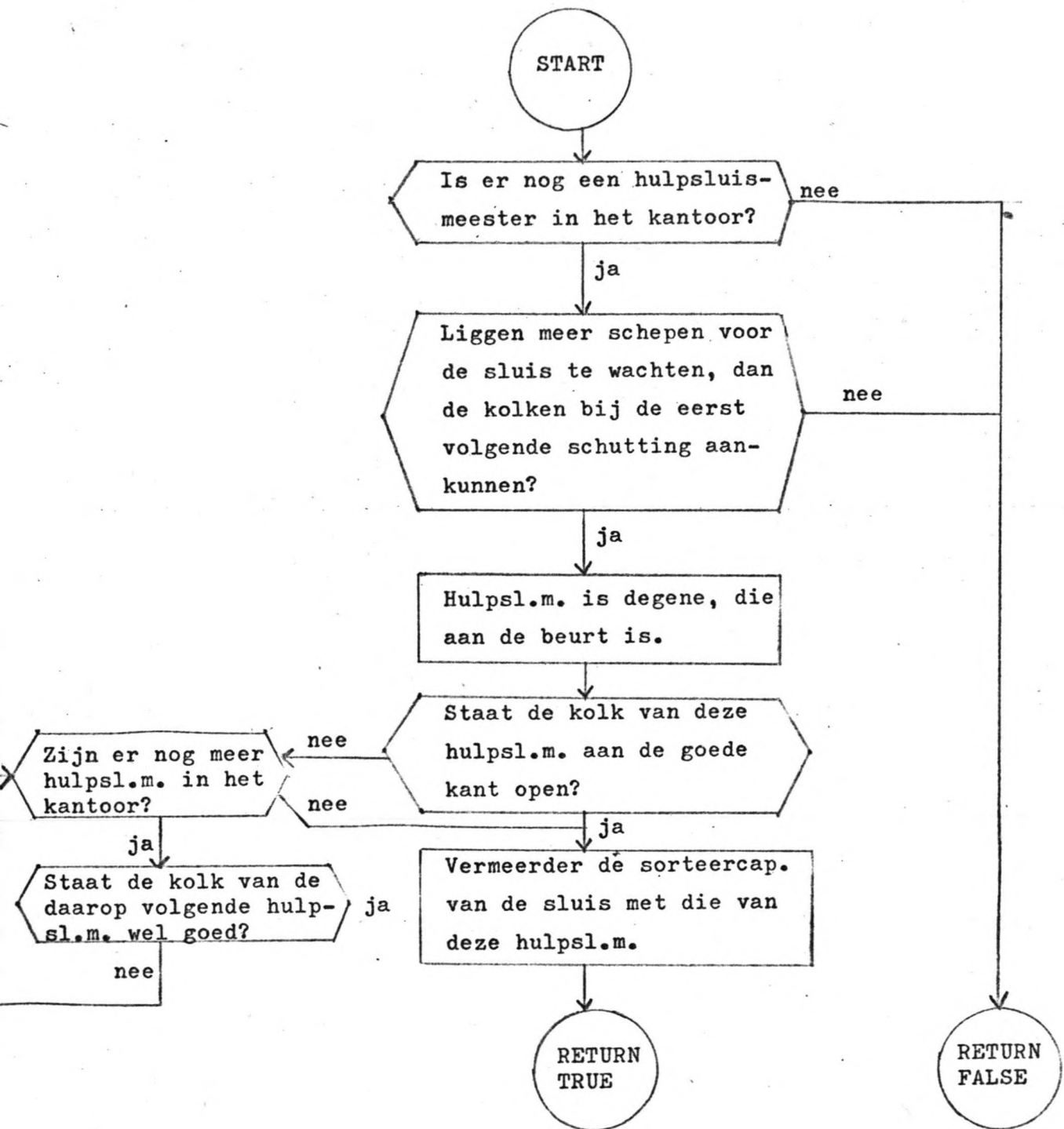




STROOMDIAGRAM: WEINIGSCHEPENAAANWEZIG: PROCEDURE RETURNS(BIT(1));



STROOMDIAGRAM: HULPSLUISMEESTERINKANTOOR: PROCEDURE RETURNS(BIT(1));



## Hoofdstuk VII

### Mogelijkheden van het model en indeling van uitvoergegevens.

#### 1. Simulatie van een sluis met één of meerdere kolken.

Hierin kunnen zich twee gevallen voordoen:

- a. De capaciteit van elke kolk afzonderlijk is niet bekend.
- b. De capaciteit van elke kolk afzonderlijk is wel bekend.

De tweede mogelijkheid levert geen problemen op, omdat de capaciteiten reeds bekend zijn en ingevoerd kunnen worden in het model.

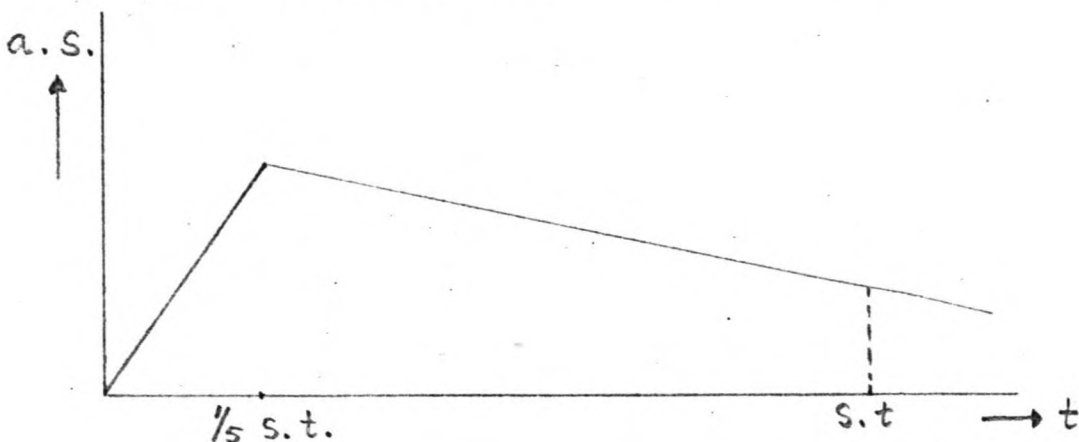
Het is noodzakelijk de capaciteiten te kennen omdat hieraan getoetst wordt, hoeveel schepen er gesorteerd moeten worden, alvorens de kolk in te delen. Stelt men geen waarde aan de volgorde, waarin de schepen binnenvaren kan men deze capaciteit de waarde 1 geven.

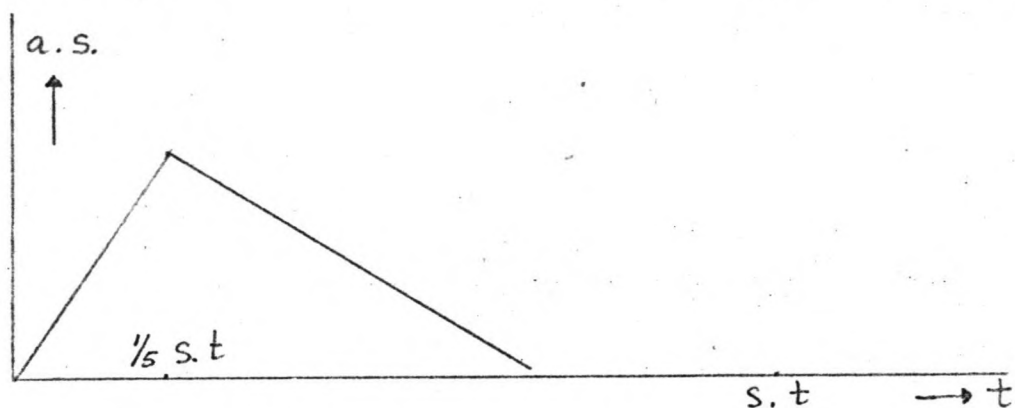
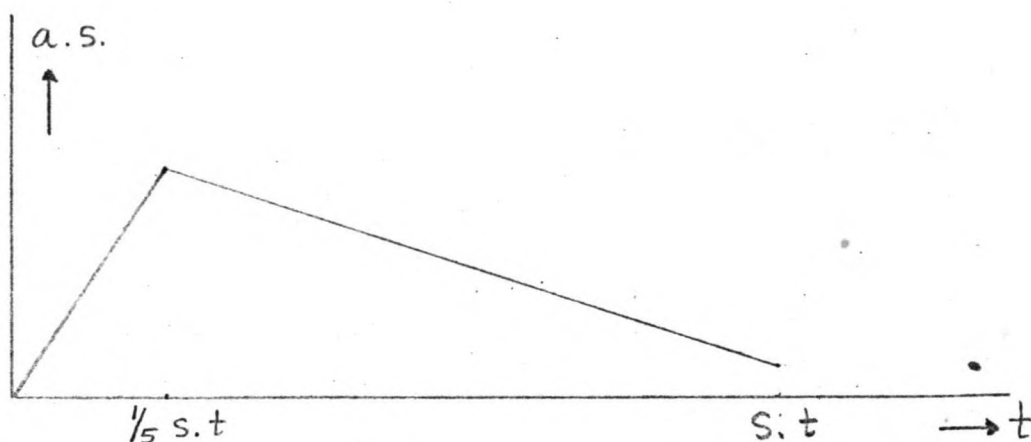
Indien de capaciteit nog niet bekend is, maar wel gewenst wordt, kan het model deze bepalen alvorens aan de werkelijke simulatie te beginnen. Om deze te kunnen bepalen, moeten we zeker zijn van een scheepsaanbod dat groter is, dan de capaciteit van de sluis.

De hulpsluismeesters worden nu in de eerste run gedurende een vijfde deel van de simulatietijd opgehouden om tot een vrij groot aanbod van schepen te komen in deze eerste run.

Het is van belang te weten voor welke tijdsperiode dit scheepsaanbod voldoende groot blijft om er een betrouwbare capaciteitsbepaling mee te kunnen doen.

Er kunnen drie gevallen onderscheiden worden:





In het eerste geval blijft het aanbod steeds groter dan de capaciteit van de sluis.

En probleem, dat hier optreedt, is het nog te grote aanbod aan het begin van de tweede run om tot een juist resultaat te komen. In de simulatie wordt dit teveel aan schepen dan ook geëlimineerd voordat aan de tweede run begonnen wordt.

In het tweede geval is het scheepsaanbod juist voldoende geweest en in het derde geval is dit aanbod aan schepen te klein.

Voor dit te kleine aanbod is in de simulatie een oplossing gevonden door de simulatie af te breken op het moment, dat er minder schepen aanwezig zijn, dan er gemiddeld in de sluis passen.

Alle bepalingen worden dan naar dat tijdstip verplaatst en de tweede run vangt dan ook op dat tijdstip aan. Bij de uitvoergegevens wordt dit tijdstip vermeld.

Een contrôle op deze methode is het nagaan of er intussen nog een hulpsluismeester naar het KANTOOR is geweest. Dit gegeven is vermeld in de status van het systeem dat eveneens op dat

tijdstip wordt gegeven.

De totale wachttijd in het kantoor moet NILL zijn.

De volgende gegevens worden verstrekt na de eerste run:

1. Het gemiddelde aantal schepen, dat in de sluis gelegen heeft per kolk.
2. Standaardafwijking hiervan.
3. Totale bevatte tonnage per schutting. (sorteer capaciteit).
4. Capaciteit per uur in tonnages.
5. Gemiddelde cyclusduur van de sluis.
6. Totale capaciteit van de sluis in tons. (Som van de capaciteiten van de kolken.)
7. Tijdstip waarop deze bepalingen gedaan zijn.

Uit de status van het systeem op dat moment blijkt het totaal aantal gegenereerde schepen.

De wachtrij INVAARVOLGORDE wordt alleen gebruikt om gesorteerde schepen in te plaatsen, die ook direct weer in de gesorteerde volgorde in hun oorspronkelijke wachtrij worden geplaatst. Hierdoor mag er aan de wachtrijen 1 en 2 ook geen waarde worden gehecht, daar deze zeer verstoord worden door dit opnieuw plaatsen in deze wachtrijen. De rest van de gegevens van STATUS spreken voor zich zelf.

In de histogram uitvoer van de wachttijd bij de sluis blijkt het otaal aantal geschutte schepen en hun wachttijden.

Aan de bezettingsgraad van de sluis mag in de eerste run bij een capaciteitsbepaling geen waarde worden gehecht, daar in deze eerste run met de rekenwaarde van 2000 ton als capaciteit van de sluis gerekend wordt. In de volgende runs is deze bezettingsgraad wel juist.

Bij een capaciteitsbepaling bestaat de uitvoer dus uit:

- 1: sluisgegevens
- 2: status van het systeem
- 3: wachttijd bij de sluis van de geschutte schepen.

De uitvoer van de tweede run is gelijk aan die van de eerste, indien er geen capaciteitsbepaling zou zijn gedaan.

Deze tweede run geeft eerst de status van het systeem op het volgend tijdstip, dan de wachttijd bij de sluis en vervolgens de doorlooptijden van de schepen bij de sluis, zowel van de



ene als van de andere zijde van de sluis.

Een variant hierop is de doorlooptijd te vervangen door de vertragingstijd, die een schip oploopt door de aanwezigheid van deze sluis.

Verdere runs hebben hetzelfde karakter.

## 2. Simulatie van een netwerk.

Deze simulatie is reeds besproken in hoofdstuk 6.

Ook hierbij bestaat de mogelijkheid om simultaan van alle sluisen tegelijkertijd de capaciteit te berekenen alvorens tot de werkelijke simulatie over te gaan.

Alle kanaalgedeeltes worden op ) gesteld wat hun lengte betreft, zodat de schepen direct bij de sluisen aankomen.

Hierbij moet evenwel bedacht worden, dat nu niet geheel op de uitkomsten van de capaciteiten vertrouwd kan worden, maar wel geven deze capaciteiten bij benadering de juiste waarde aan, zodat voor de simulatie geen verschillen optreden.

De uitvoer bestaat dan uit de sluisgegevens van alle sluisen, de status van het systeem en de wachttijden bij alle sluisen. In de tweede run komen hierbij dan alle doorlooptijden door de verschillende routes.

De simulatie van het netwerk uit hoofdstuk 6 is uitvoerig besproken in de bijgevoegde handleiding van het model.

## 3. Optimalisatie van het netwerk.

Het is mogelijk om een kostenberekening uit te laten voeren indien de prijzen bekend zijn van de schepen per tijdseenheid per klasse en per generator en de prijzen van de doorvaart per kilometer per route.

Na deze kostenberekening worden de schepen zodanig over de routes verdeeld, dat een kostenminimalisatie optreedt. Deze optimalisatie reageert op de doorlooptijden van de verschillende routes.

Tussen-resultaten worden steeds bij de doorlooptijden gegeven in verband met contrôle hierop. Achteraan in de uitvoer wordt een tabel gegeven, waarin de resultaten zodanig zijn weergegeven,

dat vertikaal deze optimalisatie gecontroleerd kan worden op convergentie. Indien deze convergentie niet of langzaam optreedt, moet de simulatietijd vergroot worden. Hoeveel groter is niet bestudeerd.

Indien naast deze optimalisatie ook nog een capaciteitsbepaling gedaan moet worden, moet op een groot aantal runs gerekend worden.

Daar de uitvoer het best besproken kan worden aan de hand van enkele voorbeelden, zal dit in de handleiding gedaan worden.

Een groot aantal berekeningen zijn in dit verslag niet expliciet vermeld omdat, wanneer men geïnteresseerd is in deze berekeningen, deze rechtstreeks uit het model afgelezen kunnen worden.

Het volgende hoofdstuk is tevens de handleiding voor het gebruik van het model met daarin enkele voorbeelden.

Dit hoofdstuk is als losse bijlage bijgevoegd.



STMT LEV NT

```

2 BREEDTE FLOAT,
2 LUSSCHIP BIT(1),
2 VOLGN0 FLOAT,
2 RICHTING FIXED BIN,
2 INVAAARTYD FLOAT,
2 UITVAARTYD FLOAT,
2 LUSTYD FLOAT,
2 AANKOMSTYD FLOAT,
2 GELADEN FIXED BIN,
2 KLASSE FIXED BIN,
2 SMELHEID FLOAT,
2 TONNAGE FLOAT,
2 PRIOR FIXED BIN,
2 VOORRANG FLOAT,
2 VAN FIXED BIN,
2 VIA FIXED BIN,
2 ROUTE(25) FIXED BIN,
2 WYZER FIXED BIN,
2 HULPSLUISM POINTER,
2 SLUISNO FIXED BIN,
1 GENERPAR BASED(GENER),
2 VERHPAR(5) FLOAT,
2 VERDEL(5) FLOAT,
2 GEM(40,2) FLOAT,
2 PR FIXED BIN,
2 KLASSEPAR FIXED BIN,
2 VANAF FIXED BIN,
2 GELADENPAR FLOAT;

/*ALGEMENE PARAMETERS*/
DECLARE
(MACHTYD(30), DOORLTYD(5,5), GEN(5,10), INVAAARVOLGORDE) POINTER,
(OPTIMALISATIEMODEL) BIT(1), (HULPSL(25,5)) POINTER, (NCT(13)) CHAR(30),
(IV, IVA, IVB) FIXED BIN(31), BP FLOAT, S FIXED BIN, (NAT(13)) CHAR(120),
TEKST1 CHAR(35), TEKST2 CHAR(33), (TEKST3, TEKST4) CHAR(8),
(AANKRN, TOTRUN, TOTKLASSEN, TOTSLUIZEN, TOTGENER, P, X, U, V, H) FIXED BIN,
(ROUJEPAR(5,5,25), BESTEMMING(5,6), F) FIXED BIN, NUM FIXED DEC(5),
(TABEL(6,5,10,5)) FLOAT, (SYSINI) FILE, CAPACITEITSBEPALING BIT(1),
(TONJOT, A(2,5), KOSTEN(5), TYD(5), C, G, STAPR(5,10,10,2), R(5)) FLOAT,
(KM(5,5), KMPRYS(5,5), B(2,5), D, BG, TEL(25,5), GEMICHT(25,5)) FLOAT,
(SIMULATIETYD, INV(6,2,2,3), UITV(6,2,2,3), L(6,2,4), M, T, Q) FLOAT;

/*BEGIN VAN HET HOOPDROCES*/
PUT EDIT('KANAALSIMULATIE- EN OPTIMALISATIEMODEL')(A);
PUT SKIP(5) EDIT('VERKEERSWATERBOUWKUNDE, (D, OOSTERVELD)')(A);
GET FILE(SYSINI) LIST(INV, UTV, L);
GET LIST(TOTRUN, IV, IVA, IVB, TOTSLUIZEN, TOTKLASSEN, TOTGENER);
GET LIST(SIMULATIETYD);

```

```

13 1 1 0
14 1 1 0
15 1 1 0
16 1 1 0
17 1 1 0

```





STMT LEV NT

```

78 1 1 1 TON(K,1)=0;TON(K,2)=0;CAP(K)=0;
81 1 1 1 PUT SKIP(5);
82 1 1 1 KAN(K,*)=KANALLENGETE(K,*);
83 1 1 1 IF CAPACITEITSBEPALING THEN KANALLENGETE(K,*)=0;
84 1 1 1 DO N=1 TO TOTKOLKEN(K);
85 1 1 1 ALLOCATE HULPSLUISSMACHTERPARL;
86 1 1 1 HULPSL(K,N)=HULPSLUISS;SLUISRY=NEWHIST(20,0,1);
88 1 1 1 ZYDE=1;
89 1 1 1 SCHUTSLND=K;
90 1 1 2 KOLKNO=N;COM: FORMAT(X(3),A(20),F(7,2));
92 1 1 2 GET LIST(SLUISELENGETE,SLUISBREEDTE,KOLKDIEPTE,LUS,KV);
93 1 1 2 GET LIST(BEDIENINGSTYD);KOLK=2000;
95 1 1 2 IF ~CAPACITEITSBEPALING THEN DO:GET LIST (KOLK);END;
96 1 1 2 CAP(K)=CAP(K)+KOLK;
98 1 1 2 SUBSTR(TEKST4,1,6)='SLUIS *';SLUIS=NEWQUEUE(TEKST4);
99 1 1 2 PUT STRING(NOT(1))EDIT('KOLKGEVENS KOLK',N) (R(COM));
101 1 1 2 PUT STRING(NOT(2))EDIT('SLUISELENGETE',SLUISELENGETE) (R(COM));
102 1 1 2 PUT STRING(NOT(3))EDIT('SLUISBREEDTE',SLUISBREEDTE) (R(COM));
103 1 1 2 PUT STRING(NOT(4))EDIT('KOLKDIEPTE 1=',KOLKDIEPTE(1)) (R(COM));
104 1 1 2 PUT STRING(NOT(5))EDIT('KOLKDIEPTE 2=',KOLKDIEPTE(2)) (R(COM));
105 1 1 2 PUT STRING(NOT(6))EDIT('LUSAFSTAND=',LUS) (R(COM));
106 1 1 2 PUT STRING(NOT(7))EDIT('KV=',KV) (R(COM));
107 1 1 2 PUT STRING(NOT(8))EDIT('BEDIENINGSTYD=',BEDIENINGSTYD) (R(COM));
108 1 1 2 (R(COM));

S=1+(30*(N-1));DO M=1 TO 8;SUBSTR(MAT(M),S, 30 )=NOT(M);END;
IF CAPACITEITSBEPALING THEN DO:
CALL ACTIVATE(NEW(HULPSLUISS),AT((1/5)*SIMULATIETYD),BYKOLK);
END;ELSE DO:
CALL ACTIVATE(NEW(HULPSLUISS),BYKOLK);END;
END;
DO J=1 TO TOTGENER;
DO I=1 TO TOTKLASSEN;
ALLOCATE GENERPAR;
VENHPAR=0;VERDELPA=0;PR=1;GEN(J,I)=GENER;
VANAFA=J;KLASSEPAR=I;GEM=1;
PUT SKIP(3);
PUT LIST EDIT('AANKOMSTEN BY GENERATOR',J,' VAN KLASSE',I)
(A,F(2),A,F(2));
W=0;
DO K=1 BY 1 WHILE(W<0.999);
GET LIST(VERDELPA(K));
W=W+VERDELPA(K);
VERHPAR(K)=W;
END;
TABELL(1,J,I,*)=VERDELPA;
GET LIST(GELADENPAR);
GET LIST(GEM(1,*));
DO K=1 BY 1 WHILE(GEM(K,1)<(SIMULATIETYD));
GET LIST(GEM(K+1,*));

```



STMT LEV NT

```

144 1 3 END;
145 1 2 DO M=1 TO K;
146 1 3 PUT SKIP EDIT('GEMIDDELDE AANKOMSTTYD TOT',GEM(M,1))(A,F(6));
147 1 3 PUT EDIT(' ',GEM(M,2))(A,F(8,2));
148 1 3 END;
149 1 2 PUT SKIP EDIT('GELADEN SCHEPEN / AANTAL SCHEPEN',GELADENPAR)
150 1 2 (A,F(8,2));PUT SKIP(2);
151 1 2 PUT SKIP DATA(VERDELP,VERKPAR);
152 1 2 CALL ACTIVATE(NE4(GENER),GENERER);
153 1 2 END;
154 1 1 DO WHILE(AANRUN->TOTRUN);
155 1 1 CALL HOLD(SIMULATIEYD);
156 1 1 IF AANRUN=1 THEN DO:PUT PAGE;
158 1 1 DO K=1 TO TOTSLUITZEN;
160 1 1 PUT EDIT('GEHEVENS VAN SLUIS',NAAM(K))(A(25),A);
161 1 3 B=0; PUT SKIP(5);IF CAPACITEITSBEPALING THEN DO:CAP(K)=0;
162 1 3 B=0; PUT SKIP(5);IF CAPACITEITSBEPALING THEN DO:CAP(K)=0;
165 1 4 DO N=1 TO TOTKOLKEN(K);
166 1 4 BG=MEANOF(HULPSL(K,N)->SLUISRY);
169 1 5 PUT STRING(NOT(9))EDIT('GEM. AANTAL SCHEPEN=',BG) (R(CAM));
170 1 5 BG=DEVIATIONOF(HULPSL(K,N)->SLUISRY);
171 1 5 PUT STRING(NOT(10))EDIT('STANDAARD AFWYKING=',BG) (R(CAM));
172 1 5 BG=GEWICHT(K,N)/(NENTRIES(HULPSL(K,N)->SLUISRY));
173 1 5 PUT STRING(NOT(11))EDIT('SORTEERCAPACITEIT=',BG) (R(CAM));
174 1 5 HULPSL(K,N)->KOLK=BG;CAP(K)=CAP(K)+BG;
176 1 5 T=NOW-((1/5)*SIMULATIEYD);
177 1 5 BG=GEWICHT(K,N)/T /60;BP=BP+BG;
179 1 5 PUT STRING(NOT(12))EDIT('CAPACITEIT=',BG) (R(CAM));
180 1 5 BG=T /NENTRIES(HULPSL(K,N)->SLUISRY)/2;
181 1 5 PUT STRING(NOT(13))EDIT('GEM. CYCLUSDUUR=',BG) (R(CAM));
182 1 5 S=1+(30*(N-1));DO M=9 TO 13;SUBSTR(NAT(M),S,30 )=NOT(M);END;
186 1 5 END;N=N-1;
188 1 4 PUT EDIT(SUBSTR(NAT(1),1,N*30))(A);PUT SKIP(2);
190 1 4 DO I=2 TO 13;PUT SKIP EDIT(SUBSTR(NAT(1),1,N*30))(A);END;END;
194 1 3 ELSE DO:N=TOTKOLKEN(K);
196 1 4 PUT EDIT(SUBSTR(NAT(1),1,30*(N))(A));PUT SKIP(2);DO I=2 TO 8;
199 1 5 PUT SKIP EDIT(SUBSTR(NAT(1),1,30*(N))(A));END;BP=CAP(K);END;
203 1 3 PUT SKIP(4) EDIT('CAPACITEIT VAN DE GEHELE SLUIS IS')(A);
204 1 3 PUT EDIT(BP,'TON')(F(8),X(2),A);
205 1 3 KANAALLENGTE(K,*)=KAN(K,*)
206 1 3 PUT SKIP EDIT('KANAALLENGTE BOVENSTROOMS=',KAN(K,1))
(A(30),F(7,3));
(A(30),F(7,3));
PUT SKIP EDIT('KANAALLENGTE BENEDENSTROOMS=',KAN(K,2))
(A(30),F(7,3));
IF CAPACITEITSBEPALING THEN DO:DO N=1,2;
DO WHILE(LENGTH(RY(K,N))=0);BOOT=FIRSTOFQUE(RY(K,N));
CALL OUTOFQUE(BOOT);CALL DESTROY(BOOT);
FREE BOOT->SCHIPPARL;
END;
END;PUT SKIP(2) EDIT('TYDSTIP VAN BEPALINGEN=',NOW,'MINUTEN')

```

208 1 3  
210 1 5  
212 1 6  
214 1 6  
215 1 6  
216 1 5

```

STAT LEV NT
218 1 4 (A,F(5),X(2),A):END:END:PUT SKIP(5);
221 1 2 END:
222 1 1 CALL STATUS;
223 1 1 PUT SKIP(3) EDIT('RESULTATEN VAN RUNNUMMER:',AANRUN)(A,F(3));
224 1 1 DO N=1 TO TOTSLUITEN;
225 1 2 PUT STRING(TEKST1)EDIT('WACHTTYD BY SLUIS',NAAM(N))(A(20),A);
226 1 2 CALL HISTPRINT(TEKST1,WACHTTYD(N));PUT SKIP(5);
228 1 2 CALL HISTCLEAR(WACHTTYD(N));
229 1 2 DO K=1 TO TOTKOLKEN(N);
230 1 3 BG=GEWICHT(N,K)/TEL(N,K);
231 1 3 PUT SKIP EDIT('BEZETTINGSRAAD VAN KOLK',K,'=',BG)
(A,F(2),A,F(5,2));
232 1 3 BG=MEANOF(HULPSL(N,K)->SLUISRY);
233 1 3 PUT SKIP EDIT('GEM. AANTAL SCHEPEN=',BG)(A(24),F(5,2));
234 1 3 CALL HISTCLEAR(HULPSL(N,K)->SLUISRY);
235 1 3 END:
236 1 2 END:
237 1 1 TEL=0:GEWICHT=0;
239 1 1 DO N=1 TO TOTGENER;
240 1 2 IF OPTIMALISATIEMODEL THEN CALL OPTIMALISEREN;
241 1 2 DO M=1 BY 1 WHILE(ROUTEPAR(N,M,1)~=0);
242 1 3 PUT STRING(TEKST2)EDIT('DOORLOOPTIJD VAN',N,' VIA ROUTE',M)
(A,F(3),A,F(3));
243 1 4 CALL HISTPRINT(TEKST2,DOORLTYD(N,M));END:
244 1 4 CALL HISTCLEAR(DOORLTYD(N,M));
246 1 3 END:
247 1 3 END:
248 1 2 PUT SKIP(5) EDIT('EINDE VAN RUNNUMMER',AANRUN)(A,F(3));
249 1 1 AANRUN=AANRUN+1;
250 1 1 CALL TIMERSET:CALL RESETQSTAT;
251 1 1 DO I=1 TO TOTGENER;
253 1 2 DO J=1 TO TOTKLASSEN;
254 1 3 GEN(I,J)->PR=1;
255 1 3 END:
256 1 3 END:CAPACITEITSBEPALING='0'B;
259 1 2 END:
260 1 1 IF OPTIMALISATIEMODEL THEN CALL OPTIMALISATIETABEL;
261 1 0 DO J=1 TO TOTGENER;
262 1 1 DO I=1 TO TOTKLASSEN;
263 1 2 CALL CANCEL(GEN(J,I));
264 1 2 END:
265 1 1 END:
266 1 0 CALL TERMINATE;
/*EINDE VAN HET HOOFDPROCES*/

```

```

267 1 0 /*HET PROCES VAN GENEREREN VAN SCHEPEN*/
GENERER:
ALLOCATE SCHIPPARL;

```

STMT LEV NT

```

268 1 0 RICHTING=1;
269 1 0 PRIOR=0;
270 1 0 GELADEN=1;
271 1 0 IF RANDOM(IVB)GELADENPAR THEN
      GELADEN=2;
272 1 0 KLASSE=KLASSEPAR;
273 1 0 LENGTE=LENGTES(KLASSE);
274 1 0 BREEDTE=BREEDTES(KLASSE);
275 1 0 TONNAGE=TONNAGES(KLASSE);
276 1 0 SNELHEID=SNELHEDEN(KLASSE);
277 1 0 VAN=VANAF;
278 1 0 D=RANDOM(IVA);
279 1 0 DO K=1 TO 5;
280 1 1 IF D<VENHPAR(K) THEN
281 1 1 DO:ROUTE=ROUTEPAR(VAN,K,*):VIA=K;
283 1 2 GOTO ROUTEBEPALD;
284 1 1
285 1 1
286 1 0
287 1 0 END;
288 1 0 ROUTEBEPALD:
289 1 0 WYZER=1;
290 1 0 IF ROUTE(WYZER)<0 THEN RICHTING=2;
291 1 0 SLUISNO=ABS(ROUTE(WYZER));
292 1 0 LUSSCHIP='0'6;
293 1 0 CALL ACTIVATE(NEW(SCHIP),AANKOMST);
294 1 0 DO WHILE(NOM>GEM(PR,1)):PR=PR+1:END;
295 1 0 CALL HOLD(-GEM(PR,2)*LOG(RANDOM(IV))):
296 1 0 GOTO GENEREER;
/*EINDE VAN HET PROCES VAN GENEREREN VAN SCHEPEN

/*PROCES VAN HET PASSEREN VAN EEN SLUIS*/
297 1 0 AANKOMST:
298 1 0 IF SLUISNO>25 THEN
299 1 1 DO:CALL HOLD((KM(VAN,VIA)/SNELHEID)*60);
300 1 1 GOTO SYSTEEMDOORGEVAREN;
301 1 1 END;
302 1 0 CALL HOLD((KANAAALLENGTE(SLUISNO,RICHTING)/SNELHEID)*60);
303 1 0 CALL TOTAALOPQUEVEERY(SLUISNO,RICHTING);
304 1 0 AANKOMSTYD=NOM;
305 1 0 TON(SLUISNO,RICHTING)=TON(SLUISNO,RICHTING)+TONNAGE;
306 1 0 IF HULPSLUISSWACHTERINKANTOOR THEN
307 1 0 CALL REACTIVATE(HULPSLUISSW,AFTER,CURRENT);
308 1 0 CALL PASSIVATE;
309 1 0 INVAREN:
310 1 0 TON(SLUISNO,RICHTING)=TON(SLUISNO,RICHTING)-TONNAGE;
311 1 0 IF AANKOMSTYD>NOM THEN AANKOMSTYD=AANKOMSTYD-SIMULATIETJD;
312 1 0 CALL HISTPUT(WACHTYD(SLUISNO),NOM-AANKOMSTYD);
313 1 0 CALL GUTDOPQUEUE;
314 1 0 IF LUSSCHIP THEN DO:
315 1 1 INVAARTYD=LUSTYD;LUSSCHIP='0'8:END;

```

STMT LEV NT

```

318 1 0 CALL HOLD(INVAARTYD);
320 1 0 INSLUIS:
CALL SORTINGQUEUE(HULPSLUISM->SLUIS,VOLGN0);
321 1 0 CALL REACTIVATE(HULPSLUISM);
322 1 0 CALL PASSIVATE;
324 1 0 UITVAREN:
CALL AUTOQUEUE;
325 1 0 CALL HOLD(UITVAARTYD);
327 1 0 IF LENGH(HULPSLUISM->SLUIS)=0 THEN
328 1 0 CALL REACTIVATE(HULPSLUISM);ELSE
CALL REACTIVATE(FIRSTOQUEUE(HULPSLUISM->SLUIS));
329 1 0 CALL HOLD((KANALLENLNGTE(SLUISNO,3-RICHTING)/SNELHEID)*60);
331 1 0 WYZER=WYZER+1;
332 1 0 IF ROUTE(WYZER)=0 THEN GOTO SYSTEEMDOORGEVAREN;ELSE
334 1 0 DO:RICHTING=1;IF ROUTE(WYZER)<0 THEN RICHTING=2;PRIOR=0;
337 1 1 SLUISNO=ABS(ROUTE(WYZER));GOTO AANKOMST;
339 1 1 END;
340 1 0 SYSTEEMDOORGEVAREN:
341 1 0 CALL HISTPUT(DOORLTYD(VAN,VIA),NOW-ARRIVALTIME);
342 1 0 FREE SCHIPPARL;
CALL TERMINATE;
/*EINDE VAN HET PROCES VAN HET PASSEREN VAN EEN SLUIS*/

/*PROCES VAN DE HULPSLUISMACHTERS*/
BYKOLK:
343 1 0 IF WEINIGSCHEPENANMEZIG THEN DO:
344 1 1 CALL SORTINGQUEUE(KANTOOR(SCHUTSLNO),-KOLKNO);
345 1 1 CALL PASSIVATE;
347 1 1 CALL OUTOQUEUE;END;
349 1 0 IF TON(SCHUTSLNO,ZYDE)->CAP(SCHUTSLNO)-KOLK THEN
350 1 0 DO:ZYDE=3-ZYDE;
351 1 1 CALL HOLD(BEDIENINGSTYD);
353 1 1 END;
354 1 0 RAAI=0;RAAI(1,1)=SLUISBREEDTE;RAAI(1,2)=SLUISLENGTE;TAL=0;
358 1 0 CALL VERANDERDEVOLGORDE;
359 1 0 EERSTESCHIP:
BOOT=FIRSTOQUEUE(RY(SCHUTSLNO,ZYDE));
IF BOOT=NULL THEN GOTO BYKOLK;ELSE BOOT->LUSSCHIP=1;B;
360 1 0 KOLKINDELEN:
BOOT=FIRSTOQUEUE(RY(SCHUTSLNO,ZYDE));
362 1 0 DO WHILE(BOOT/=NULL):
363 1 0 NEXTSCHIP=SUC(C(BOOT));
364 1 1 IF SCHIPPAST THEN DO:
365 1 1 BOOT->VOLGN0=-(RAAI(P,2)-BOOT->LENGTE);
366 1 2 BOOT->HULPSLUISM=HULPSLUIS;
367 1 2 BOOT->INVAARTYD=INVAART;
368 1 2 BOOT->LUSTYD=LUST;
369 1 2 BOOT->UITVAARTYD=UITVAART;
370 1 2 GEMICHT(SCHUTSLNO,KOLKNO)=GEMICHT(SCHUTSLNO,KOLKNO)+
371 1 2

```

111

STMT LEV NT

```

372 1 2 BOOT->TONNAGE;TAL=TAL+I;
373 1 2 CALL REACTIVATE(BOOT,AFTER,CURRENT);
374 1 2 CALL PASSIVATE(KOLKINDELEN);END;ELSE
      BOOT=NEXTSCHIP;
377 1 1 END;
378 1 0 TEL(SCHUTSLNO,KOLKNO)=TEL(SCHUTSLNO,KOLKNO)+KOLK;
379 1 0 CALL HISTPUT(SLUISTRY,TAL);
380 1 0 CALL HOLD(BEDIENINGSTYD);
382 1 0 ZYDE=3-ZYDE;
383 1 0 CALL REACTIVATE(FIRSTOFQUEUE(SLUISS));
384 1 0 CALL PASSIVATE(BYKOLK);
/*EINDE VAN HET PROCES VAN DE HULPSLUISSWACHERS*/

```

/\*PROCEDURES\*/

```

385 1 0 SCHIPPAST: PROCEDURE RETURNS(BIT(1));
386 2 0 P=I;
387 2 0 PASTSCHIP;
      IF RAAI(P,1)<BOOT->BREEDTE THEN
389 2 0 DO:P=P+1;GOTO PASTSCHIP;END;
391 2 0 IF RAAI(P,2)<BOOT->LENGTE THEN
      RETURN('0'B);
392 2 0 X=0;
393 2 0 GROOTSTERAAI;
      IF RAAI(P+X,1)~=0 THEN
394 2 0 DO:IF RAAI(P+X,2)<RAAI(P,2)-300->LENGTE THEN
395 2 1 END;
397 2 2 DO:X=X+1;GOTO GROOTSTERAAI;
398 2 1 IF RAAI(P+X,1)<BOOT->BREEDTE THEN
399 2 1 DO:P=P+1;GOTO PASTSCHIP;
401 2 2 END;
402 2 1 X=X+1;GOTO GROOTSTERAAI;
404 2 1 END;
405 2 0 SCHIPPASTWEL;
406 2 0 X=X-1;IF RAAI(P+X,2)<RAAI(P,2)-BOOT->LENGTE THEN
407 2 0 DO:RAAI(P+X+1,1)=RAAI(P+X,1);
408 2 1 RAAI(P+X+1,2)=RAAI(P+X,2);
409 2 1 GOTO SCHIPPASTWEL;
410 2 1 END;
411 2 0 RAAI(P+X+1,2)=RAAI(P,2)-BOOT->LENGTE;
412 2 0 RAAI(P+X+1,1)=RAAI(P+X,1);
413 2 0 DO WHILE(X<0);
414 2 1 RAAI(P+X,1)=RAAI(P+X,1)-BOOT->BREEDTE;
415 2 1 X=X-1;
416 2 1 END;RETURN('1'B);
418 2 0 END SCHIPPAST;
419 1 0 INVAART: PROCEDURE RETURNS(FLOAT);

```



STMT LEV NT

```

420 2 0 Q=(SLUISBREEDTE*KOLK/DIEPTE(BOOT->RICHTING))/100;
421 2 0 T=INV(BOOT->KLASSE,KV,BOOT->GELADEN,3)*(Q**2)+
      INV(BOOT->KLASSE,KV,BOOT->GELADEN,2)*Q+
      INV(BOOT->KLASSE,KV,BOOT->GELADEN,1);
422 2 0 RETURN(T);
423 2 0 END INVART;

```

```

424 1 0 UTVART: PROCEDURE RETURNS(FLOAT);
425 2 0 Q=(SLUISBREEDTE*KOLK/DIEPTE(3-BOOT->RICHTING))/100;
426 2 0 T=UITV(BOOT->KLASSE,KV,BOOT->GELADEN,3)*(Q**2)+
      UITV(BOOT->KLASSE,KV,BOOT->GELADEN,2)*Q+
      UITV(BOOT->KLASSE,KV,BOOT->GELADEN,1);
427 2 0 RETURN(T);
428 2 0 END UTVART;

```

```

429 1 0 LUST: PROCEDURE RETURNS(FLOAT);
430 2 0 U=BOOT->KLASSE;V=BOOT->GELADEN;M=BOOT->BREEDTE;
431 2 0 T=((L(U,V,1)-L(U,V,3))*(M/SLUISBREEDTE)+L(U,V,3))+((L(U,V,2)-
432 2 0 L(U,V,4))*(M/SLUISBREEDTE)+L(U,V,4))*(LUS+BOOT->LENGTE));
433 2 0 RETURN(T);
434 2 0 END LUST;

```

```

436 1 0 WEINIGSCHEPENANMEZIG: PROCEDURE RETURNS(BIT(1));
437 2 0 IF TON(SCHUTSLNO,ZYDE)->CAP(SCHUTSLNO)-KOLK THEN
438 2 0 DO:IF TON(SCHUTSLNO,3-ZYDE)->CAP(SCHUTSLNO)-KOLK THEN
439 2 1 DO:CAP(SCHUTSLNO)=CAP(SCHUTSLNO)-KOLK;
440 2 2 IF CAPCITEIISBEPALING THEN DO:
441 2 3 IF LENGHT(RY(SCHUTSLNO,ZYDE))<MEANDF(SLUSRY) THEN DO:
442 2 4 CALL CANCEL(MAIN):CALL REACTIVATE(MAIN):END:END:
443 2 2 END:
444 2 2 RETURN(*1*B);
445 2 2 END:
446 2 2 RETURN(*0*B);
447 2 2 END WEINIGSCHEPENANMEZIG;
448 2 1
449 2 0
450 2 0

```

```

451 1 0 HULPSLUISWACHTERINKANTOGR: PROCEDURE RETURNS(BIT(1));
452 2 0 IF LENGHT(KANTOGR(SLUSNO))=0 THEN RETURN(*0*B);
453 2 0 IF TON(SLUSNO,RICHTING)<CAP(SLUSNO) THEN RETURN(*0*B);
454 2 0 HULPSLUISWELASTOFQUEUE(KANTOGR(SLUSNO));
455 2 0 BESTEZOEKEN:
456 2 0 IF HULPSLUISW->ZYDE=RICHTING THEN
457 2 1 DO:CAP(SLUSNO)=CAP(SLUSNO)+HULPSLUISW->XKOLK;
458 2 1 RETURN(*1*B);
459 2 1 END:
460 2 0 HULPSLUISW=PRED(HULPSLUISW);
      IF HULPSLUISW=NULL THEN

```



STMT LEV NT

```

461 2 0 DO:HULPSLUISM=LASTOQUEUE(KANTOOR(SLUISNO));
462 2 1 CAP(SLUISNO)=CAP(SLUISNO)+HULPSLUISM->KOLK;
463 2 1 RETURN('119');
464 2 1 END;
465 2 0 GOTO 9ESTEZOEKEN;
466 2 0 END HULPSLUISMACHTERINKANTOOR;
    
```

```

467 1 0 VERANDERDEVOLGORDE: PROCEDURE;
468 2 0 TONTOT=0;
469 2 0 BOOT=FIRSTOQUEUE(RY(SCHUTSLNO,ZYDE));
470 2 0 DO WHILE(TONTOT<0.8*KOLK & BOOT<=NULL);
471 2 1 NEXTSCHIP=SUCS(BOOT);
472 2 1 CALL OUTOQUEUE(BOOT);
473 2 1 TONTOT=TONTOT+BOOT->TONNAGE;
474 2 1 BOOT->VOORRANG=-BOOT->KLASSE;
475 2 1 IF BOOT->PRIOR>0 THEN BOOT->VOORRANG=-((TOTKLASSEN+BOOT->PRIOR));
476 2 1 CALL SORTINQUEUE(BOOT,INVAARVOLGORDE,BOOT->VOORRANG);
477 2 1 BOOT=NEXTSCHIP;
478 2 1 END;
479 2 0 DO WHILE(LENGTH(INVAARVOLGORDE)~=0);
480 2 1 BOOT=LASTOQUEUE(INVAARVOLGORDE);CALL OUTOQUEUE(BOOT);
481 2 1 CALL TOHEADOQUEUE(BOOT,RY(SCHUTSLNO,ZYDE));
482 2 1 BOOT->PRIOR=BOOT->PRIOR+1;
483 2 1 END;
484 2 1 END VERANDERDEVOLGORDE;
485 2 0
    
```

```

486 1 0 OPTIMALISEREN: PROCEDURE;
487 2 0 A(1,*)=0;A(2,*)=100;
488 2 0 K=1;PUT SKIP(3);
489 2 0 DO WHILE(BESTEMMING(N,K)~=0);
490 2 1 KOSTEN(K)=(KM(N,K)*KMPRYS(N,K));
491 2 1 TYD(K)=MEANDE(DOORLTYD(N,K))/60;
492 2 1 IF TYD(K)<0 THEN TYD(K)=KM(N,K)/14;
493 2 1 PUT SKIP EDIT('KOSTEN VAN ROUTE',K,KOSTEN(K))A(F(3),F(8,2));
494 2 1 PUT EDIT(' SEMIDDELDE DOORVAARTID:',TYD(K))A(F(6,2));
495 2 1 K=K+1;
496 2 1 END;
497 2 1 DO I=1 BY 1 WHILE(BESTEMMING(N,I)~=0);
498 2 1 DO J=I+1 BY 1 WHILE(BESTEMMING(N,J)~=0);
499 2 1 IF BESTEMMING(N,J)=BESTEMMING(N,I) THEN
500 2 2 DO:IF TYD(I)=TYD(J) THEN
501 2 3 DO:IF KOSTEN(J)>KOSTEN(I) THEN
502 2 4 A(2,J)=0;ELSE A(2,I)=0;
503 2 4 END:ELSE
504 2 4 DO:C=(KOSTEN(J)-KOSTEN(I))/(TYD(I)-TYD(J));
505 2 4 IF C>100 THEN C=100;IF C<0 THEN C=0;
506 2 4 IF TYD(I)>TYD(J) THEN
507 2 5 DO:IF C>A(1,J) THEN A(1,J)=C;
508 2 5
509 2 5
510 2 5
511 2 5
    
```

STAT LEV NT

```

512 2 5 IF C(A(2,I)) THEN A(2,I)=C;
513 2 5 END;ELSE
516 2 4 DO:IF C>A(1,I) THEN A(1,I)=C;
517 2 5 IF C<A(2,J) THEN A(2,J)=C;
518 2 4 END;
519 2 3 END;
520 2 2
521 2 1 END;
522 2 0 PUT SKIP(3) DATA(A);
523 2 0 DO M=1 TO TOTKLASSEN:R=0:B=A;
526 2 1 DO I=1 BY 1 WHILE(BESTEMMING(N,I)~=0);
527 2 2 IF B(1,I)~<0 THEN
528 2 2 DO:G=GEN(N,M)->VERDELP(I);
529 2 2 DO J=I+1 BY 1 WHILE(BESTEMMING(N,J)~=0);
530 2 4 IF BESTEMMING(N,J)=BESTEMMING(N,I) THEN
G=GEN(N,M)->VERDELP(J)+G;
END;
DO K=1 TO 5;IF BESTEMMING(N,K)=BESTEMMING(N,I) THEN
DO:F=1;
DO WHILE(STAPGR(N,M,F,2)<B(1,K)):F=F+1;
END;W=0;
DO WHILE(STAPGR(N,M,F,2)<B(2,K)):
W=STAPGR(N,M,F,1)+W:F=F+1;
END;
R(K)=W*G;
B(1,K)=-1;
END;
END;

```

```

531 2 4 DO K=1 TO 5;IF BESTEMMING(N,K)=BESTEMMING(N,I) THEN
532 2 3 DO:F=1;
533 2 4 DO WHILE(STAPGR(N,M,F,2)<B(1,K)):F=F+1;
537 2 5 END;W=0;
539 2 6 DO WHILE(STAPGR(N,M,F,2)<B(2,K)):
540 2 5 W=STAPGR(N,M,F,1)+W:F=F+1;
542 2 6 END;
543 2 5 R(K)=W*G;
544 2 5 B(1,K)=-1;
545 2 5 END;
546 2 4 END;
547 2 3 END;
548 2 2 TABEL(AANRUN+1,N,M,#)=R;
549 2 1 TABEL(AANRUN+1,N,M,#)=R;
550 2 1 W=0;
551 2 1 DO K=1 TO 5;
552 2 2 W=W+R(K);GEN(N,M)->VERHPAR(K)=W;
553 2 2 END;
554 2 2 END OPTIMALISEREN;
555 2 1
556 2 0

```

OPTIMALISATIETABEL: PROCEDURE:

```

557 1 0 PUT PAGE;
558 2 0 PUT EDIT('TABEL VAN DE VERHOUDINGEN WAARIN DE SCHEPEN DE')(A);
559 2 0 PUT EDIT(' VERSCHILLENDE ROUTES KIEZEN')(A);PUT SKIP(2);
560 2 0 DO I=1 TO TOTGENER;
562 2 0 DO J=1 BY 1 WHILE(BESTEMMING(I,J)~=0);
563 2 1 ROUTE(I,J)(A,F(2),A,F(2));
564 2 2 DO K=1 TO TOTKLASSEN;
565 2 2 PUT EDIT(' KLAS',K)(A,F(2));
566 2 3 END;PUT SKIP(2);
567 2 3

```

```

569      2 2 2
570      2 2 3
571      2 2 3
572      2 2 4
573      2 2 4
574      2 2 3
575      2 2 2
576      2 2 1
577      2 2 0
578      1 0 0

```

```

      DO M=0 TO (ANRUN-1);
      PUT SKIP EDIT('NA',M,'  RUNS')(A,F(3),A(15));
      DO N=1 TO TOTKLASSEN;
      PUT EDIT(TABEL(M,I,N,J))(F(6,2));
      END;
      END;
      END;
      END OPTIMALISATIETABEL;
      END;

```

BIJLAGE C.

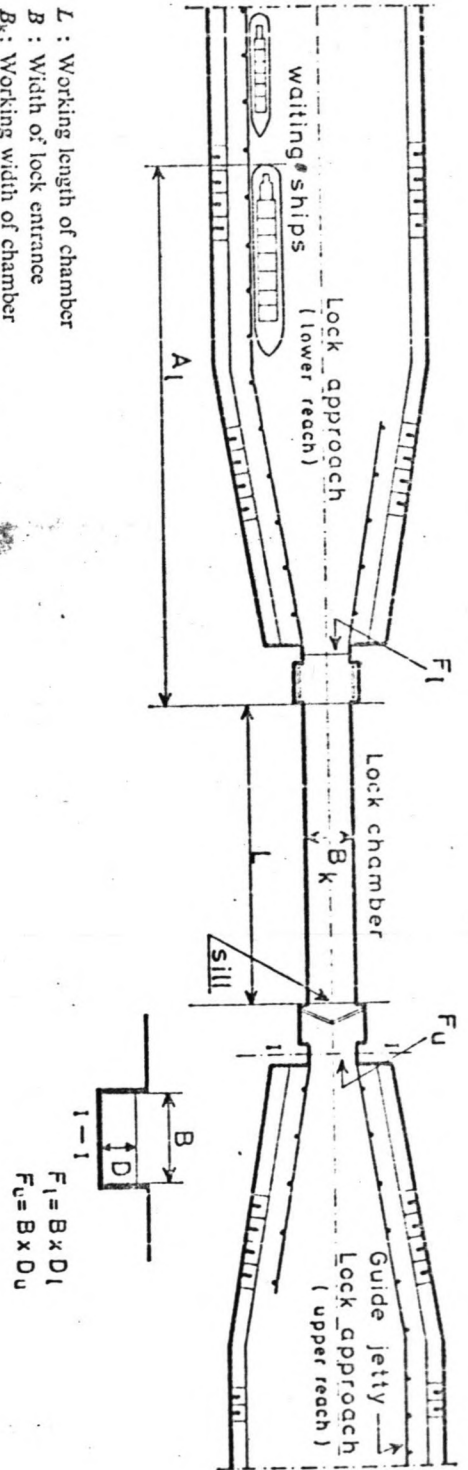
TABEL

Jaadverm. klasse nr.	Jaadverm. (tonnen)	Lengte (m)	Breedte (m)	diepgang (m)	motor- verm. (APK)	opp. max. natte dwarsdsn. (m <sup>2</sup> )		waterverpl. (m <sup>3</sup> )		l x b (m <sup>2</sup> )
						gel.	leeg	gel.	leeg	
1	125	25,0	4,6	1,6	60	7,4	1,4	165	40	115
2	325	39,0	5,1	2,3	140	11,7	2,7	425	80	200
3	550	50,0	6,6	2,5	260	16,5	4,6	710	160	330
4	925	67,0	8,2	2,5	520	20,5	6,8	1180	255	550
5	1350	80,0	9,5	2,6	730	24,7	9,3	1740	390	760
6	2000	95,0	11,5	2,7	1100	31,0	12,3	2600	600	1090
7	4100	175,0	11,4	3,0	1200	34,2	17,0	5150	1050	2000
8	8800	185,0	22,8	3,2	2500	73,0	17,0	10600	1800	4220

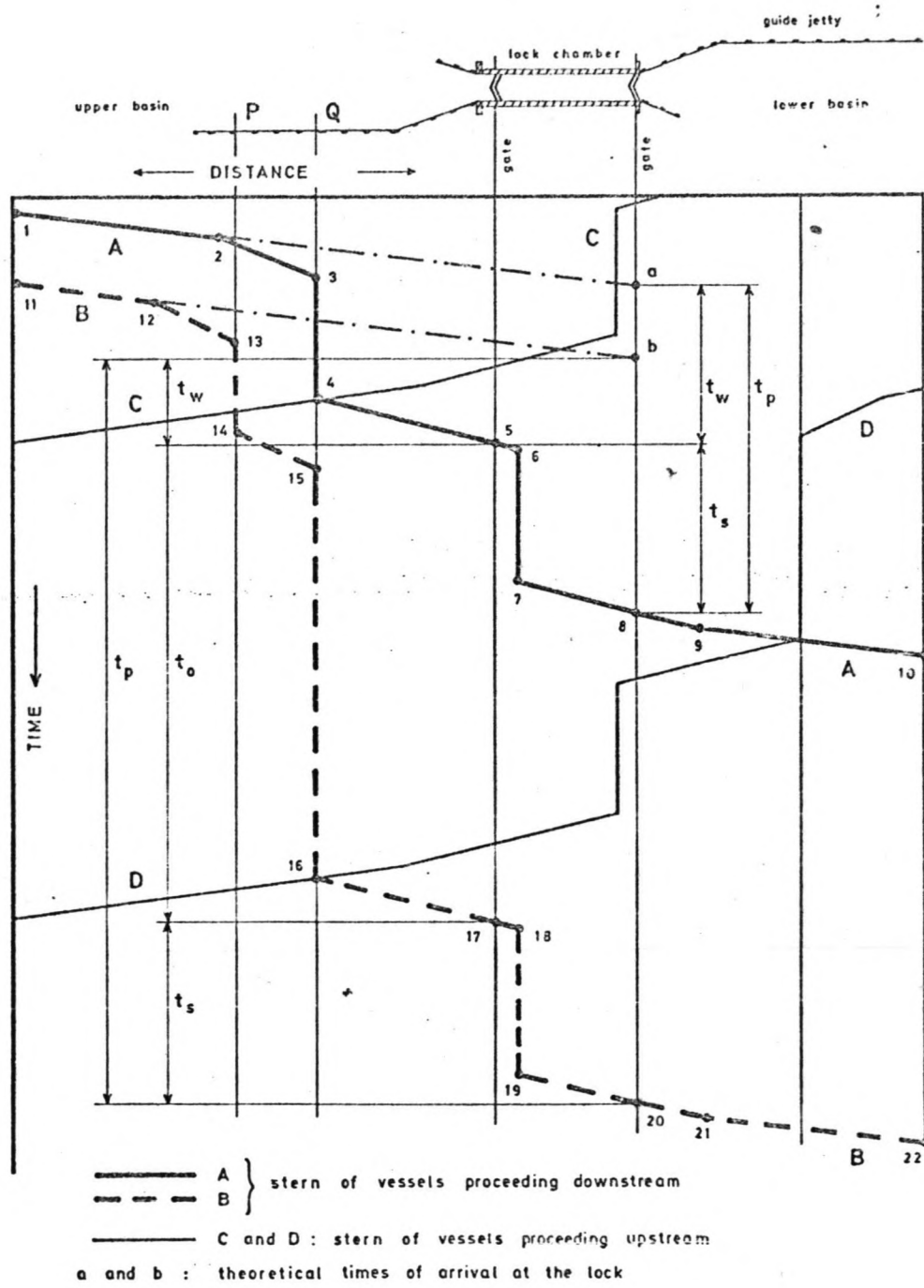
Gegevens standaardschepen.

BIJLAGE D.

- L : Working length of chamber
- B : Width of lock entrance
- B<sub>k</sub> : Working width of chamber
- F : Area of wet cross-section above sill or lock floor at the lock gate (B x D)
- A<sub>1</sub> : Switch distance (from first ship to enter - to gates)



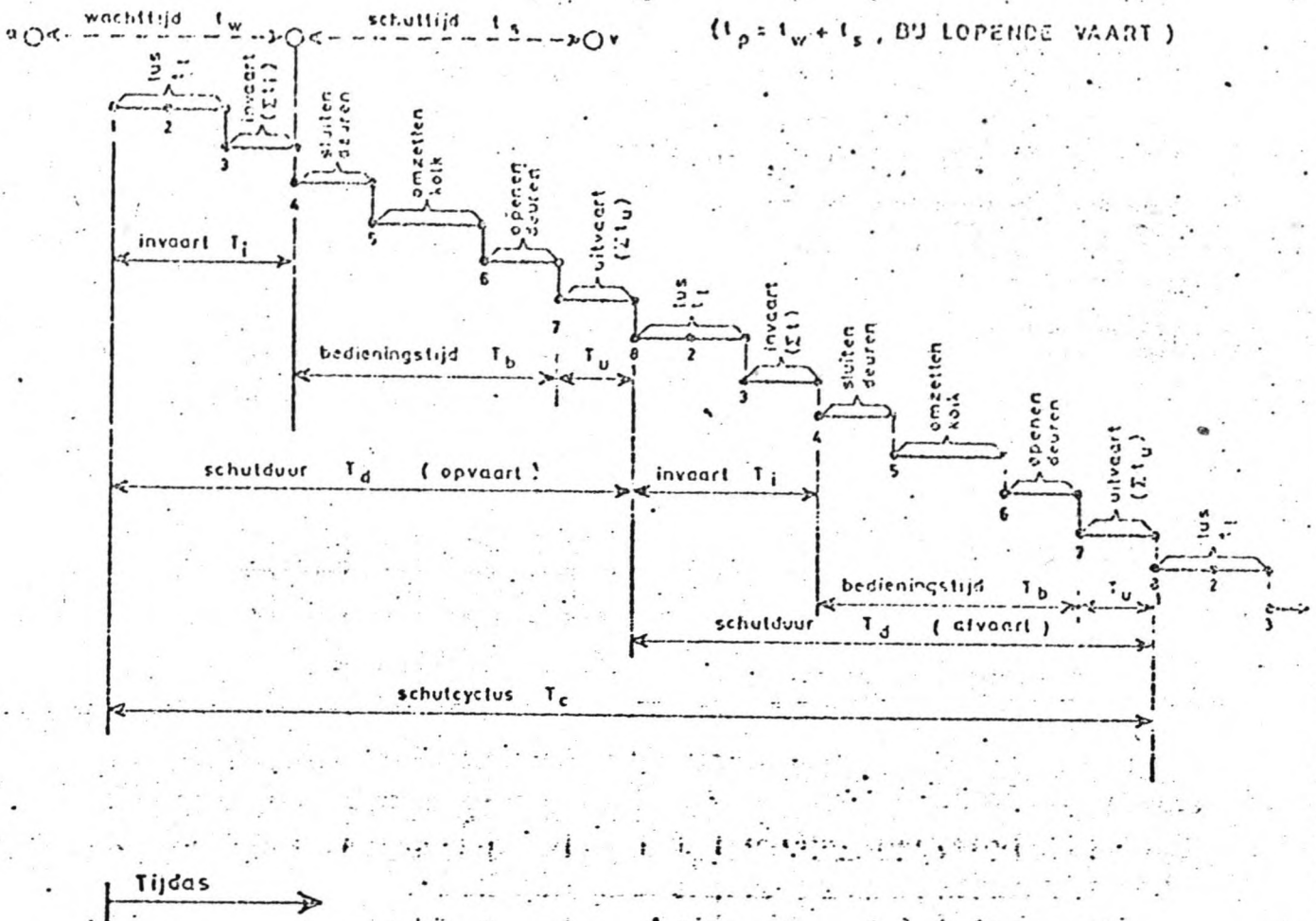




Time-distance diagram showing transit through lock.

BIJLAGE E.TabelToegepaste symbolen

Symbool	Betekenis	Dimensie
APK	Aan de schroef toegevoerd vermogen (aspaardekrachten)	-
$B_b$	Bodembreedte van de vaarweg	(m)
$B_s$	Breedte op de waterspiegel	(m)
D	Waterdiepte	(m)
F	Oppervlak van het natte dwarsprofiel	(m <sup>2</sup> )
$P_s$	Specifiek vermogen (vermogen per m <sup>3</sup> waterverplaatsing)	(APK/m <sup>3</sup> )
R	Straal van de bocht in een vaarweg	(m)
$V_s$	Vaarsnelheid t.o.v. het water	(km/u)
$V_g$	Theoretische grenssnelheid	(km/u)
b	Scheepsbreedte	(m)
d	Diepgang	(m)
$d_a$	Diepgang achterschip	(m)
$d_v$	Diepgang voorschip	(m)
f	Natte dwarsdoorsnede van het schip met het maximale oppervlak (ook wel ondergedompeld grootspantoppervlak)	(m <sup>2</sup> )
g	Versnelling van de zwaartekracht	(m/sec <sup>2</sup> )
h	Gemiddelde waterdiepte (=F/B <sub>s</sub> )	(m)
l	Scheepslengte over alles (ook wel l.o.a.)	(m)
$l_{ll}$	Lengte tussen de loodlijnen	(m)
r	Straal van de schroefcirkel (= halve diameter)	(m)
Δ	Waterverplaatsing	(m <sup>3</sup> )
λ	Beladingsgraad per schip	



Tijdstip.

Fase.

- a Aankomst van een willekeurig opvarend schip bij de sluis.
- v Achterstevan van bovengenoemd schip passeert uitvaardeuren.
- 1 Achterstevan v/h laatste uitvarende schip passeert deuren (afvaart).
- 2 Start van eerste invarende schip (opvaart) vanaf opstelruimte.
- 3 Achterstevan v/h eerste invarende schip passeert deuren.
- 4 Achterstevan v/h laatste invarende schip passeert deuren.
- 5 Invaardeuren zijn gesloten.
- 6 Uitvaardeuren beginnen te openen.
- 7 Uitvaardeuren zijn geopend.
- 8 Achterstevan v/h laatste uitvarende schip passeert deuren (opvaart).

opm. De lengte van de horizontale lijnen heeft geen verband met de duur van de aangegeven handelingen.

OPDOEW VAN DE SCHUTCYCLUS

fig. 1

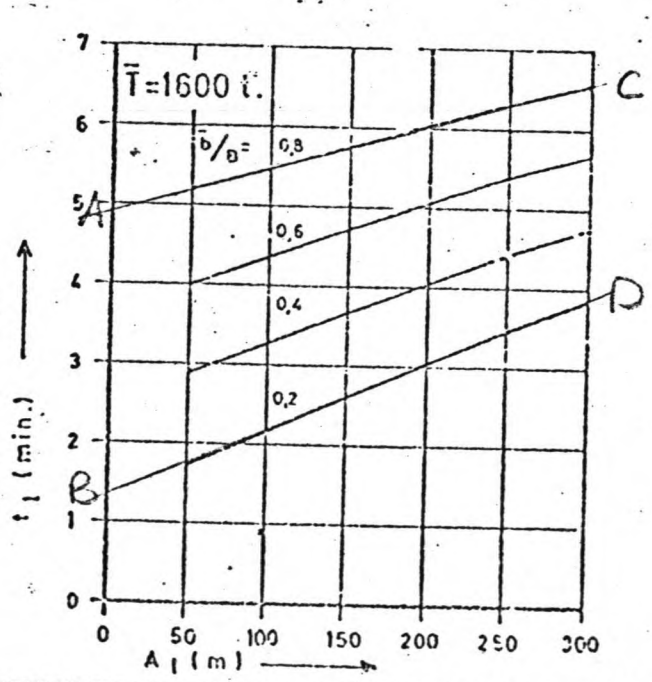
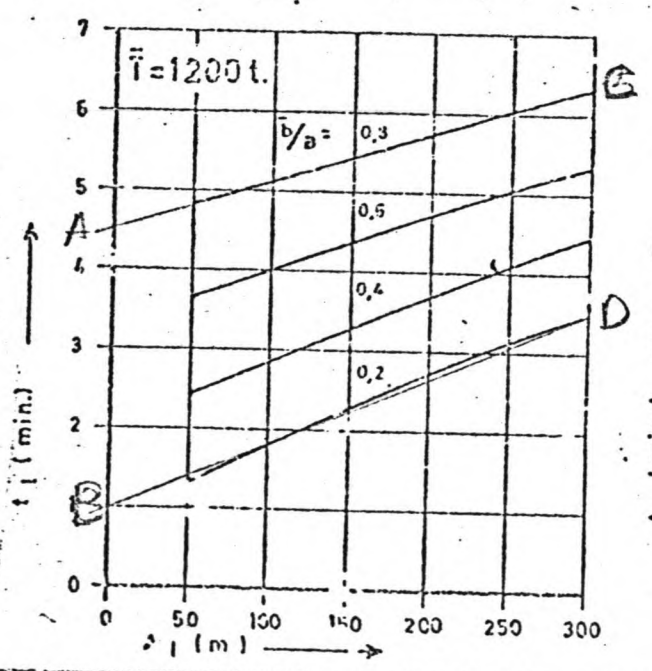
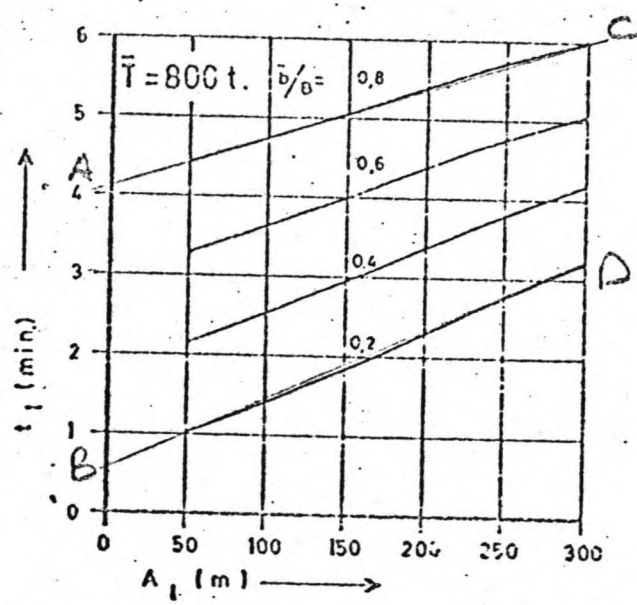
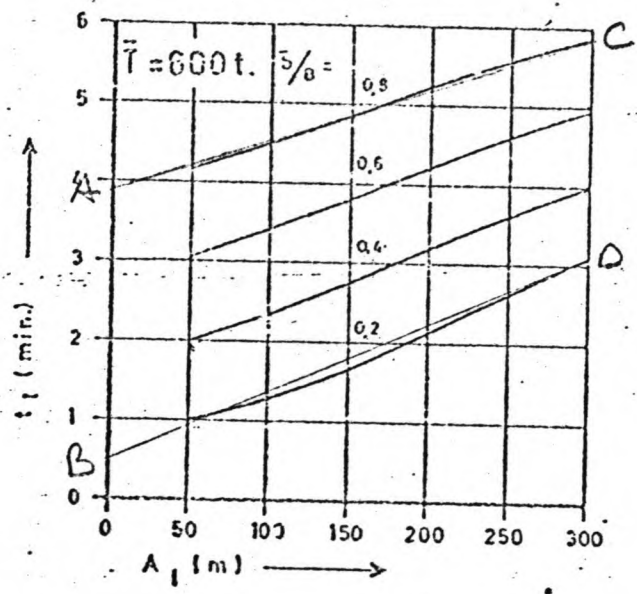
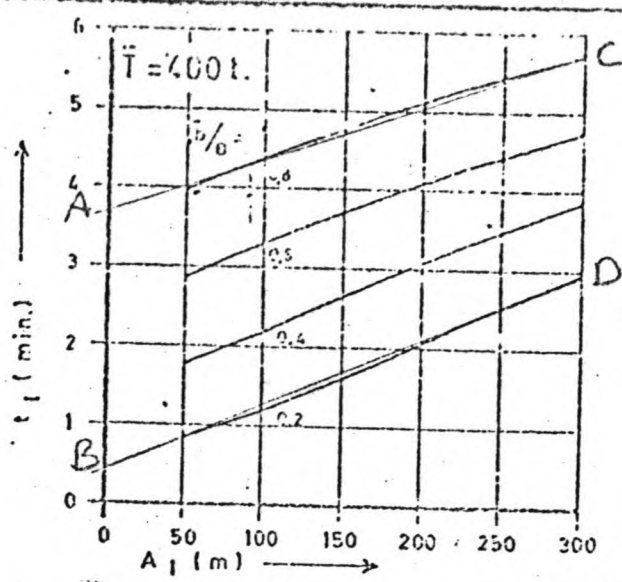
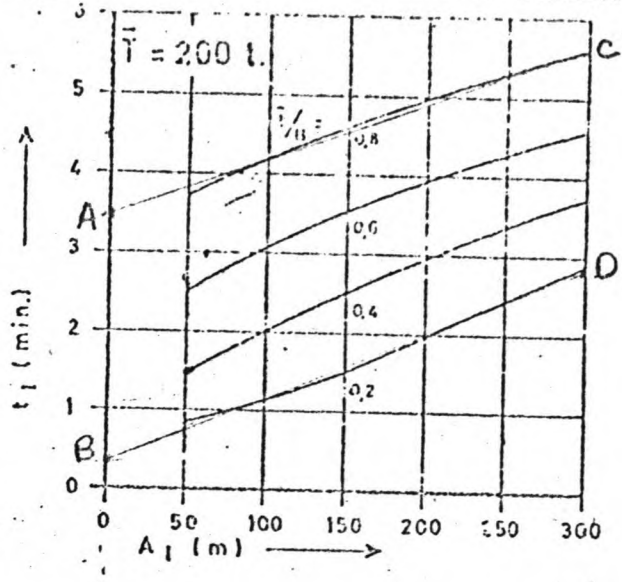
SCHAAL:

get. gez.

6-12-71

Nr. 70.12.7.

RUKSWATERSTAAT  
DIENST VERKEERSKUNDE  
HOOFDAFDIENING SCHEEPVAART



VERBAND TUSSEN LUSTUD ( $t_1$ ) EN LUSAFSTAND ( $A_1$ ) BIJ VERSCHILLENDE WAARDEN VAN  $b/a$  EN  $\bar{T}$  VOOR GELADEN MOTORSCHEPEN ( $KV=1$ )

RIJKSWATERSTAAT  
 DIENST VERKEERSKUNDE  
 HOOFDAAFDELING SCHLEPVAART

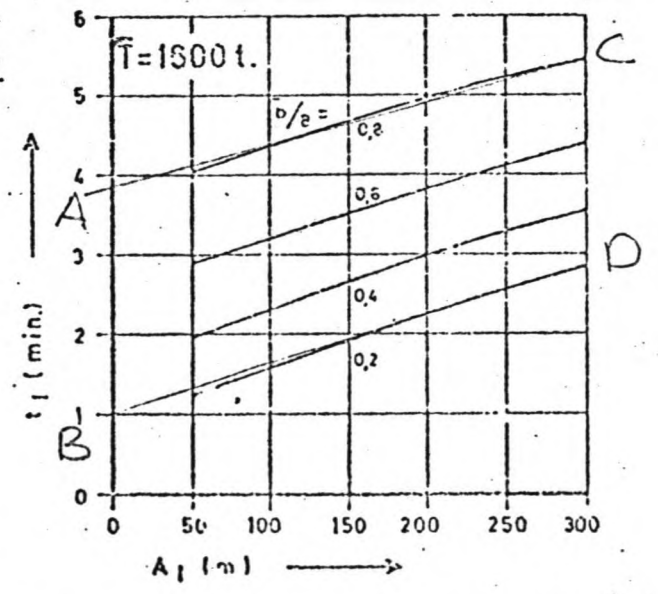
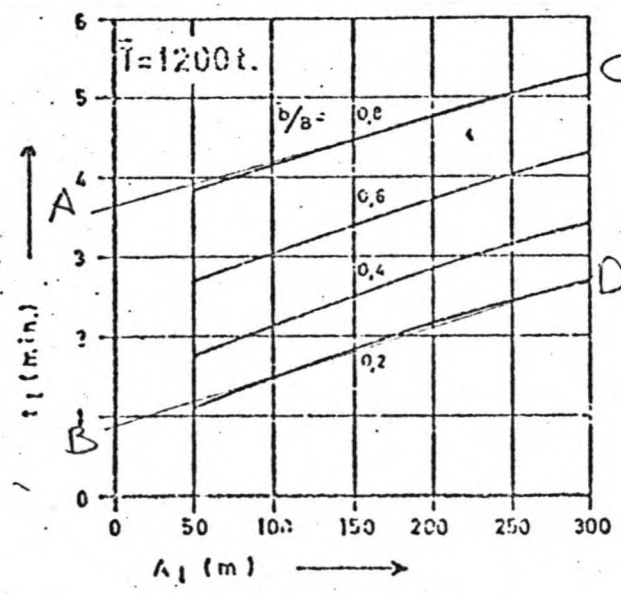
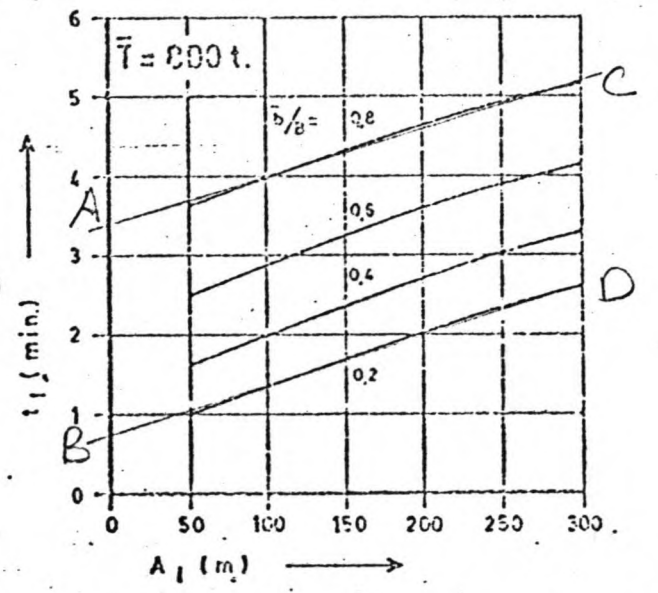
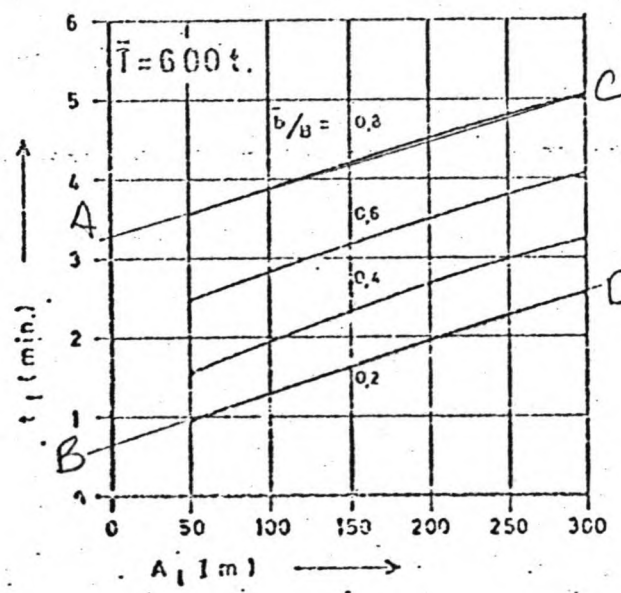
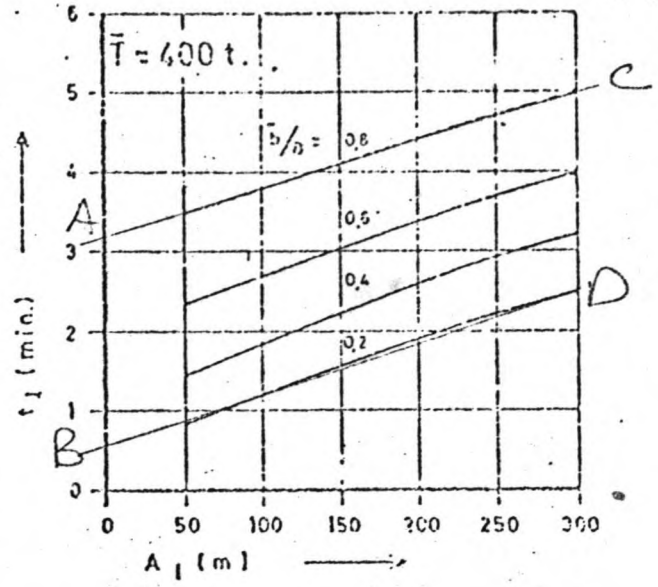
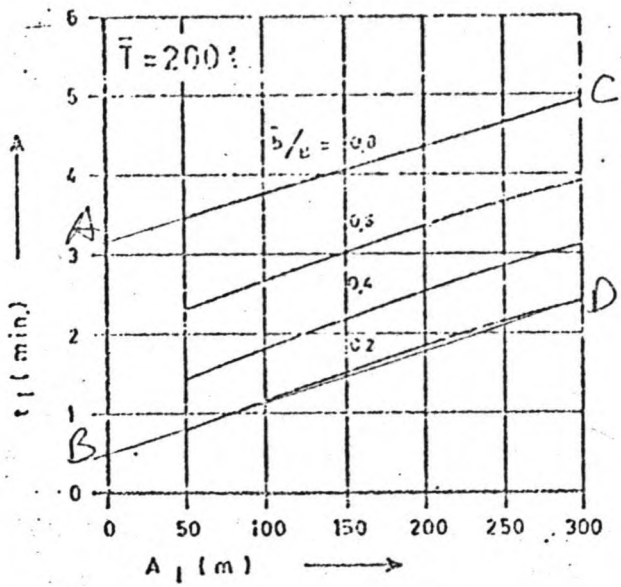
Fig. 2

SCHAAL:

SPt.	NOV.
14-2-73	
J.M.	

Nr. 70.12.





VERBAND TUSSEN LUSTUD ( $l_1$ ) EN LUSAFSTAND ( $A_1$ ) BIJ VERSCHILLENDE WAARDEN VAN  $\bar{l}/B$  EN  $\bar{T}$  VOOR ONGELADEN MOTORSCHEPEN ( $KV = 1$ )

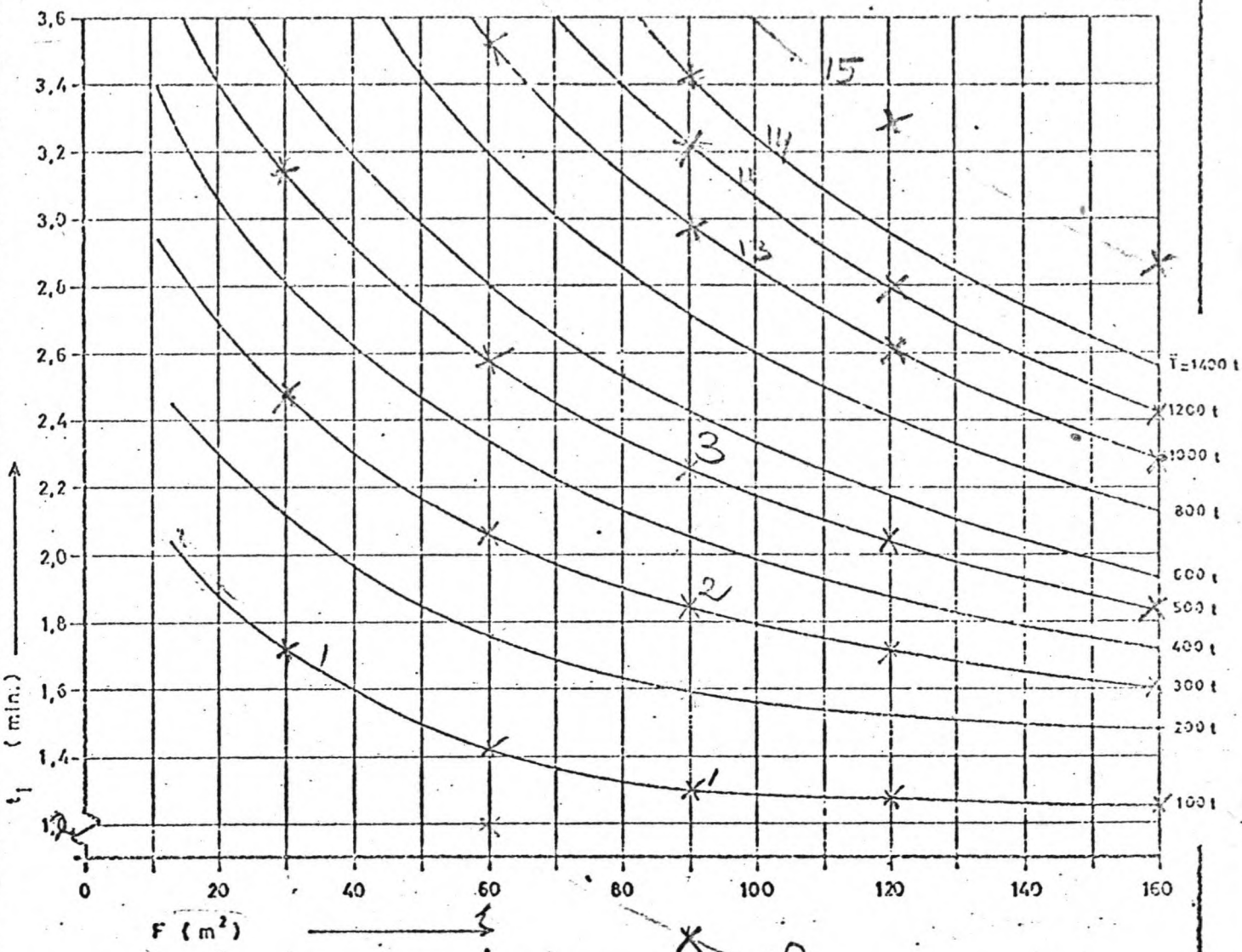
Fig. 3

RIJSWATERSTAAT  
DIENST VERKEERENDE  
ROOFDARSTELLING SCHIEPVAART

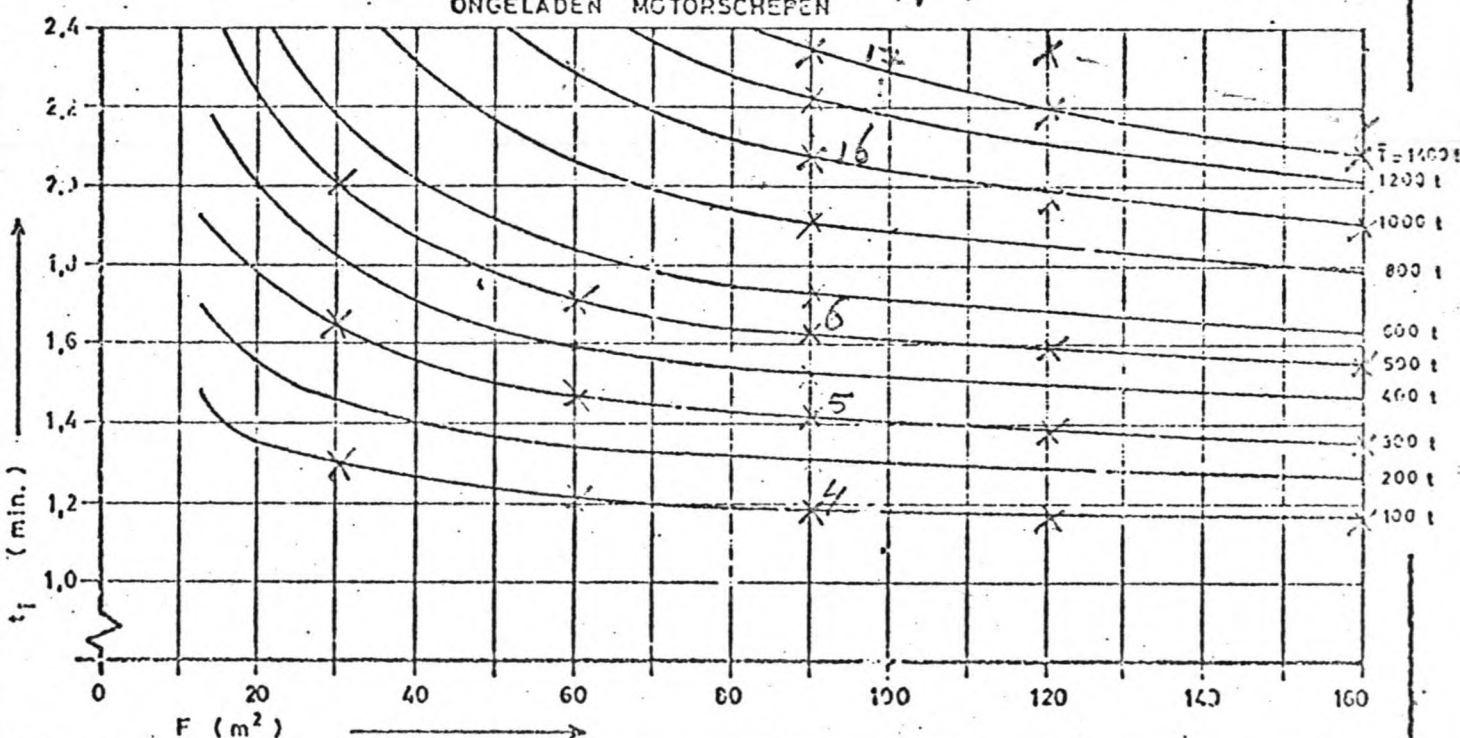
SCHAAL:

1:1  
1:4  
1:16  
1:64

Nr. 7.0.13.7.



ONGELADEN MOTORSCHEPEN



VERBAND TUSSEN INVAARVOLGTJD EN F

(KV = 1)

RIJKSWATERSTAAT  
DIENST VERKLEERSKUNDE  
HOOFDAFDELING SCHEEPVAART

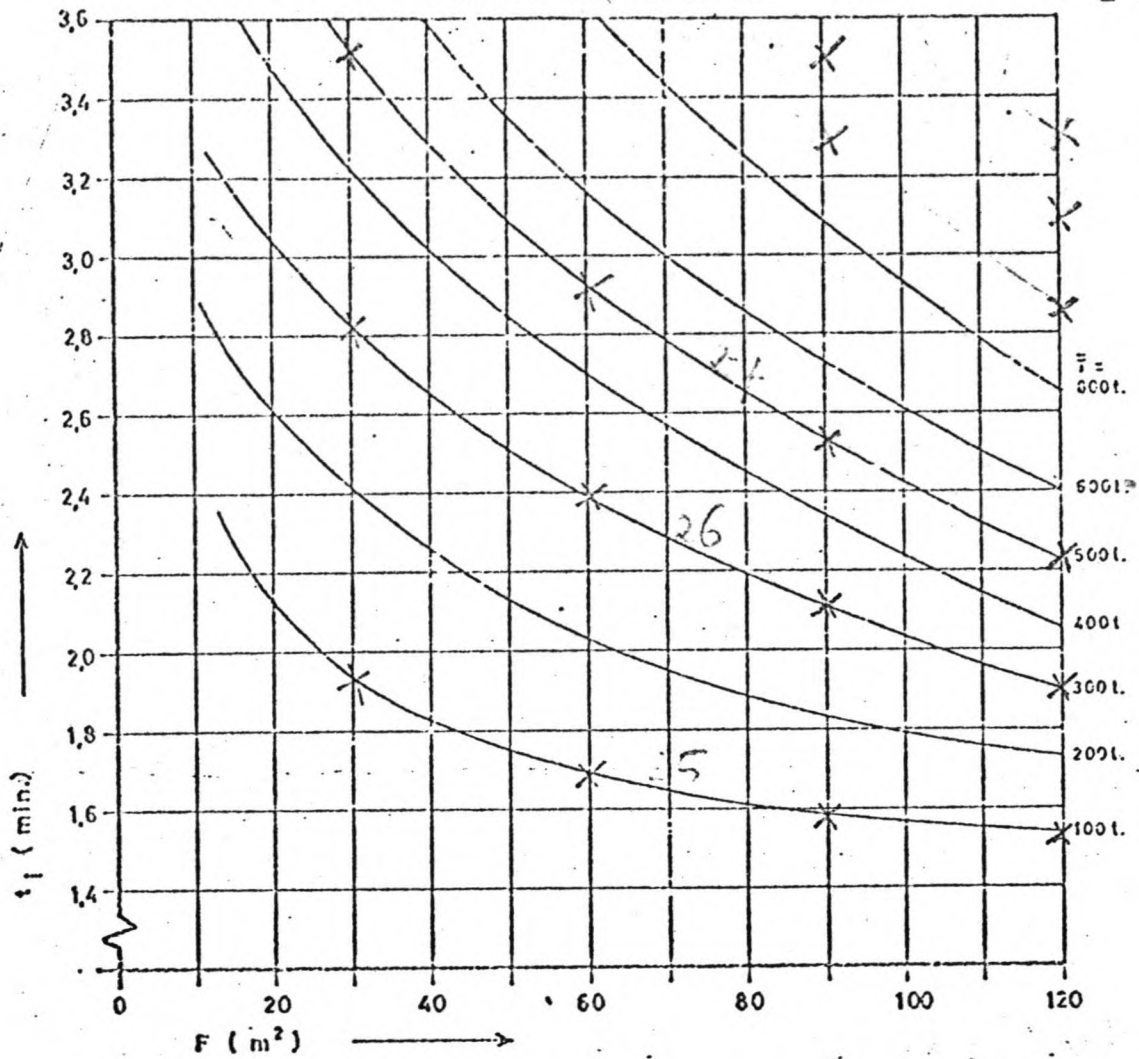
Fig. 4

SCHAAL:

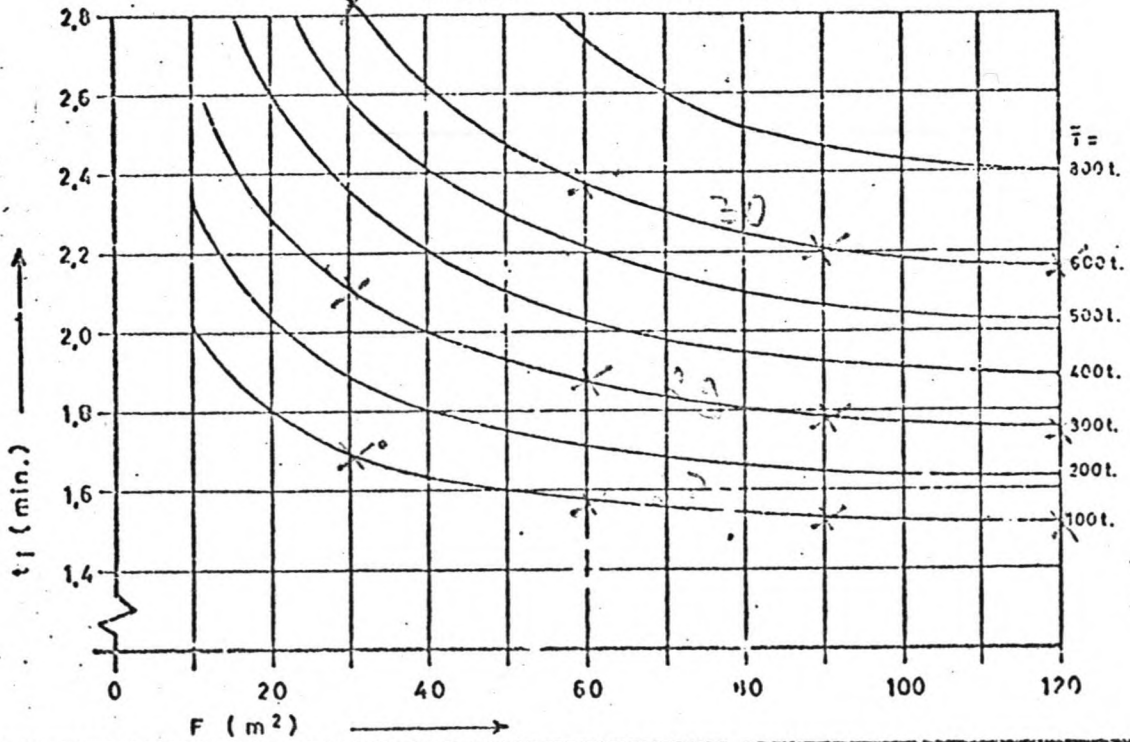
get.	get.
19.12.70	
59	

Nr. 70.12.7





ONGELADEN MOTORSCHEPEN



VERBAND TUSSEN INVAARVOLGTUD EN F  
(KV = 2)

Fig. 5

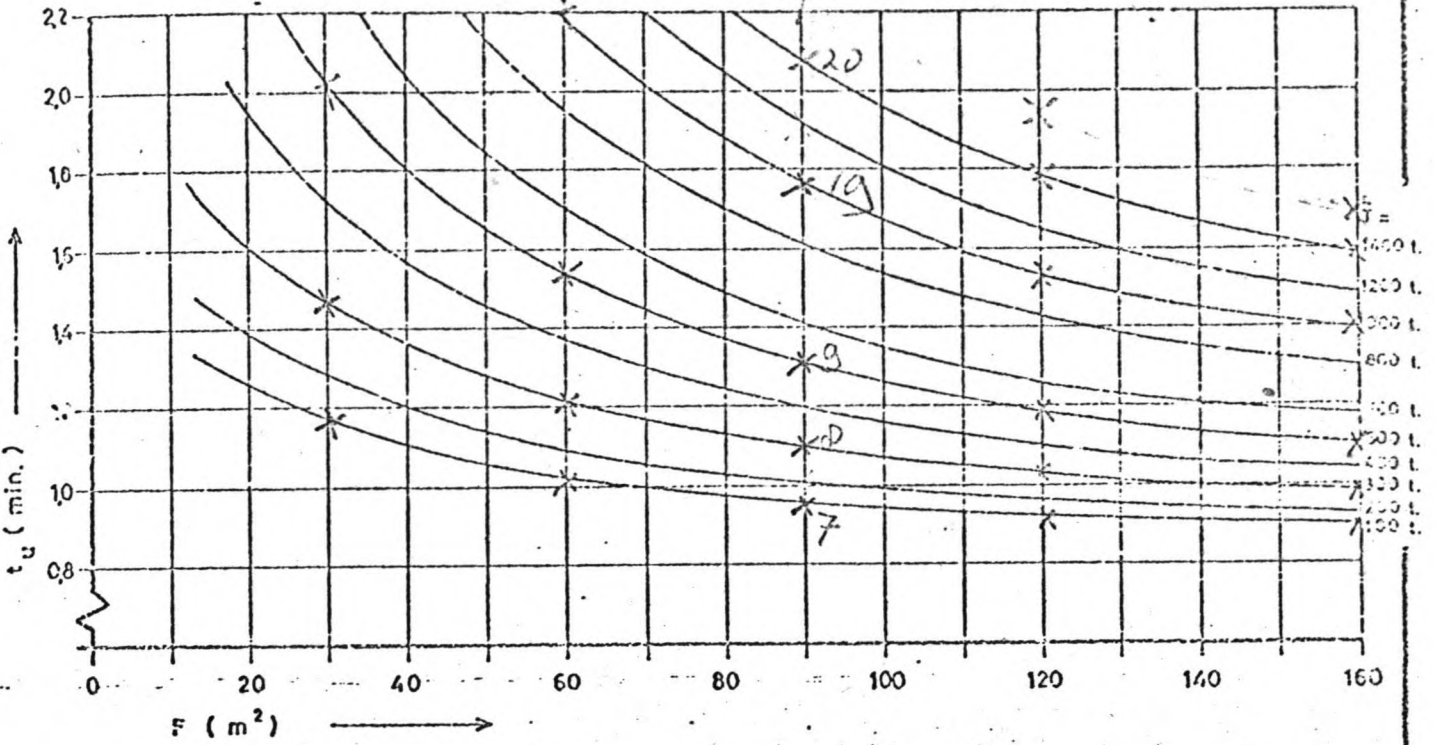
RIJKSWATERSTAAT  
DIENST VERKEERSKUNDE  
OOFGAFDELING SCHIEPVAART

SCHAAL:

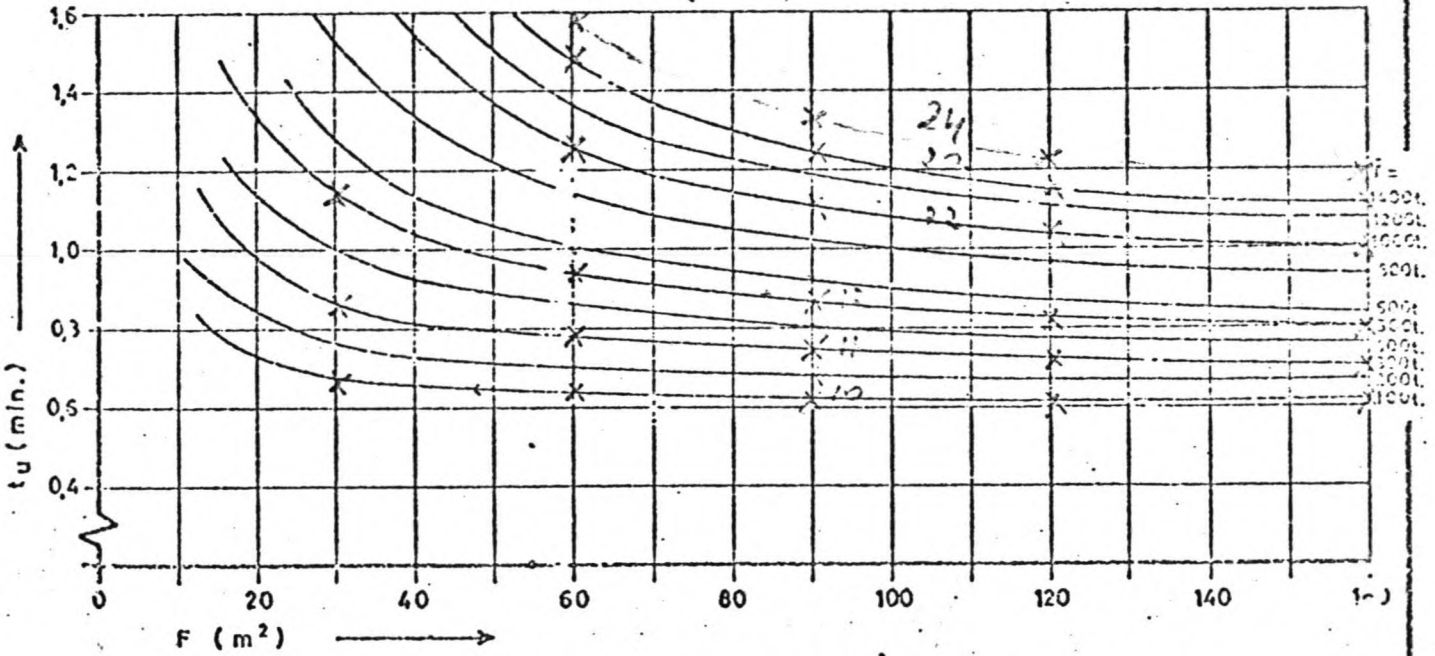
50t.	907.
19-12-73	
J.M.	

Nr. 70.12.7.

21  
GELADEN MOTORSCHEPEN



ONGELADEN MOTORSCHEPEN



VERBAND TUSSEN UITVAARVOEGTUD EN F  
(KV = 1)

Fig 6

RIJKSWATERSTAAT  
DIENST VERKEERSKUNDE  
HOOFDOPDELING SCHIPVAART

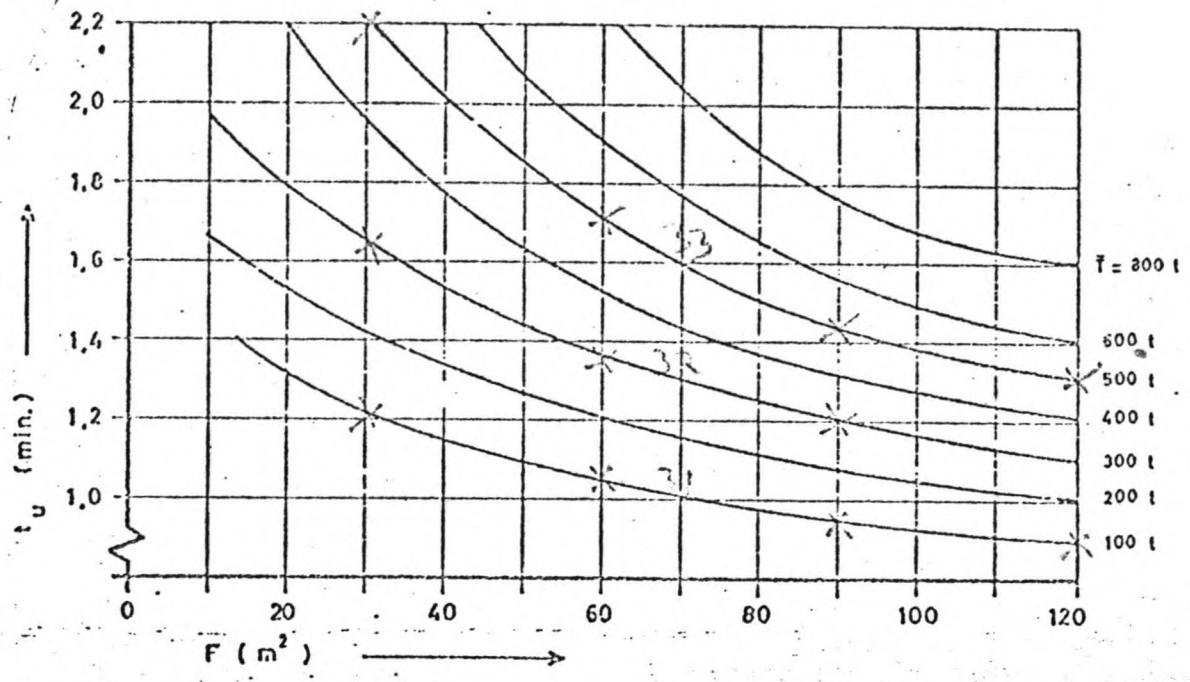
SCHAAL:

St. 19-12-75  
J.M.

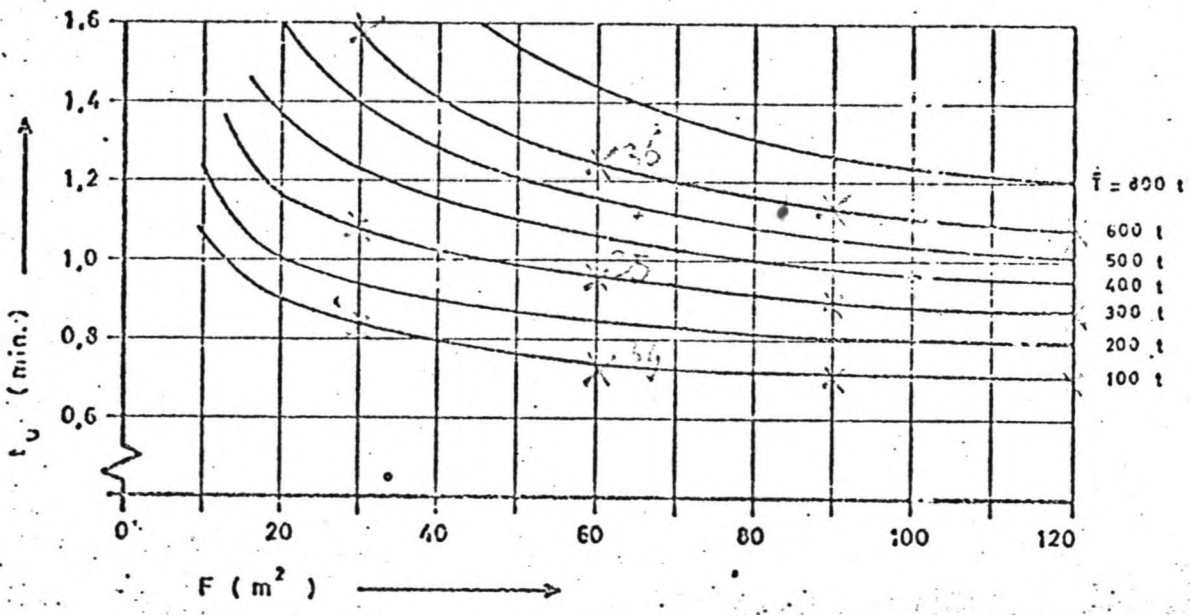
100%

Nr. 70.12.7.

GELADEN MOTORSCHEPEN



ONGELADEN MOTORSCHEPEN



VERBAND TUSSEN UITVAARVOLGTJD EN F  
(KV = 2)

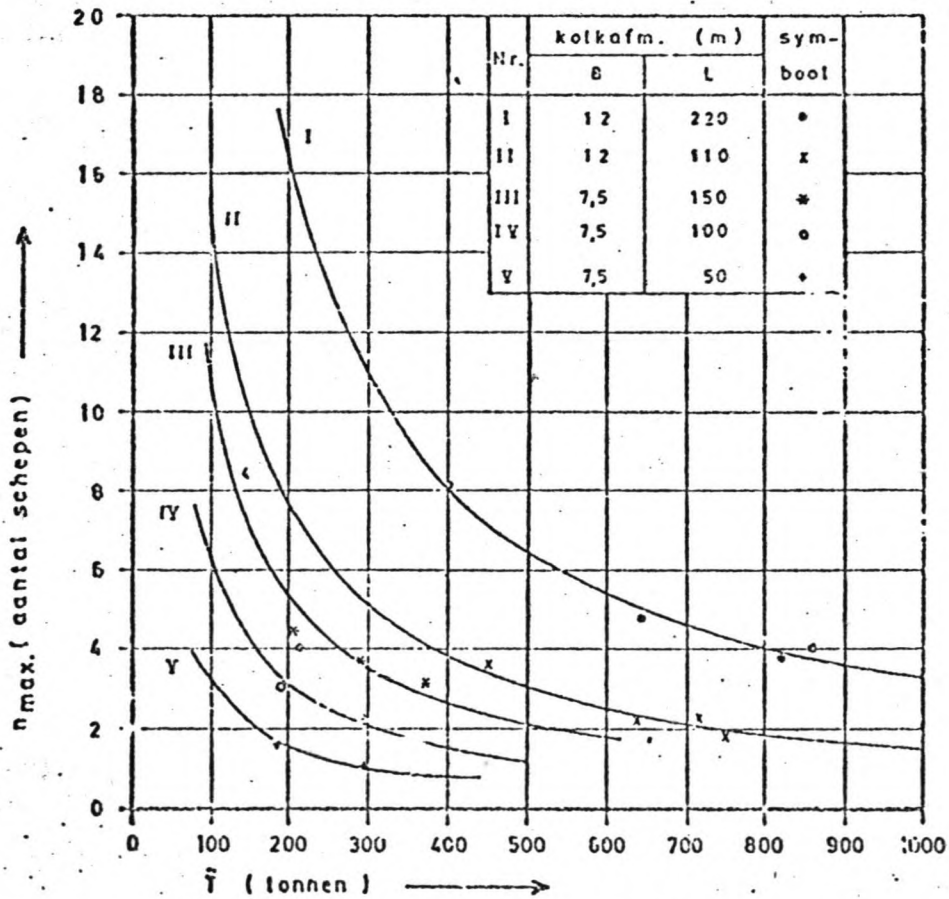
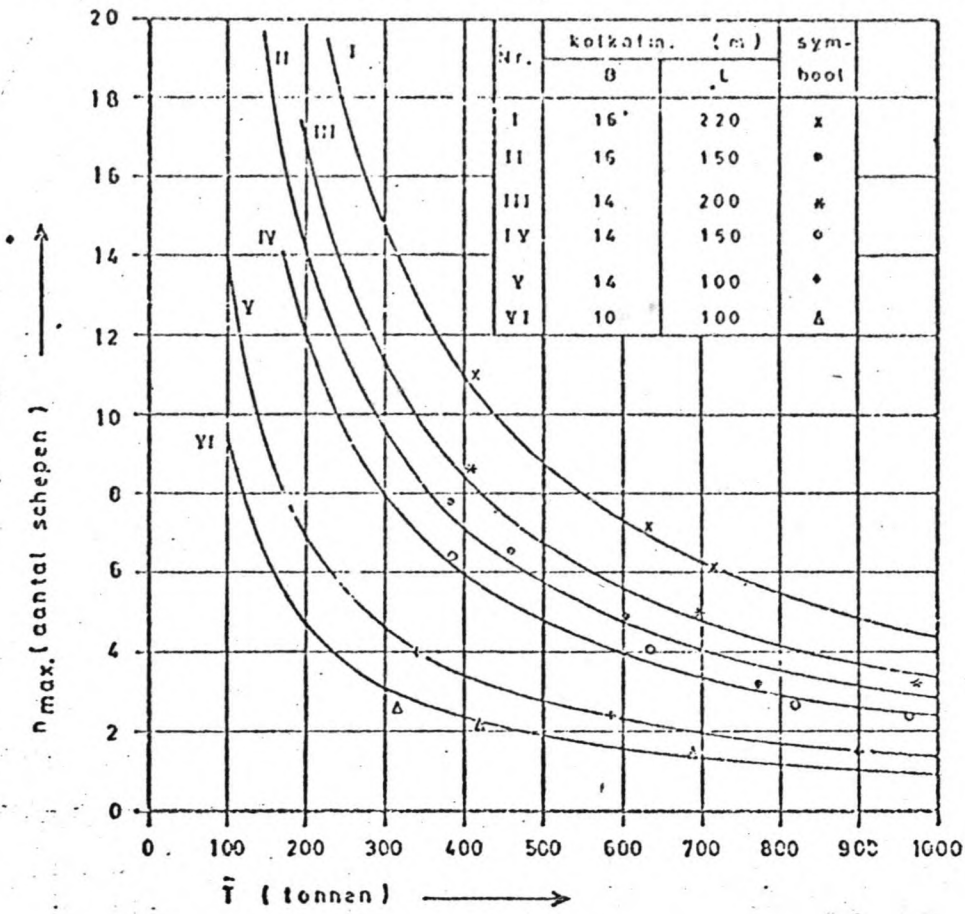
Fig. 7.

RIKSWATERSTAAT  
MIENST VERKEERSKUNDE  
HOOFDAPDELING SCHEEPVAART

SCHAAL:

981.	982.
19-12-73	
88	

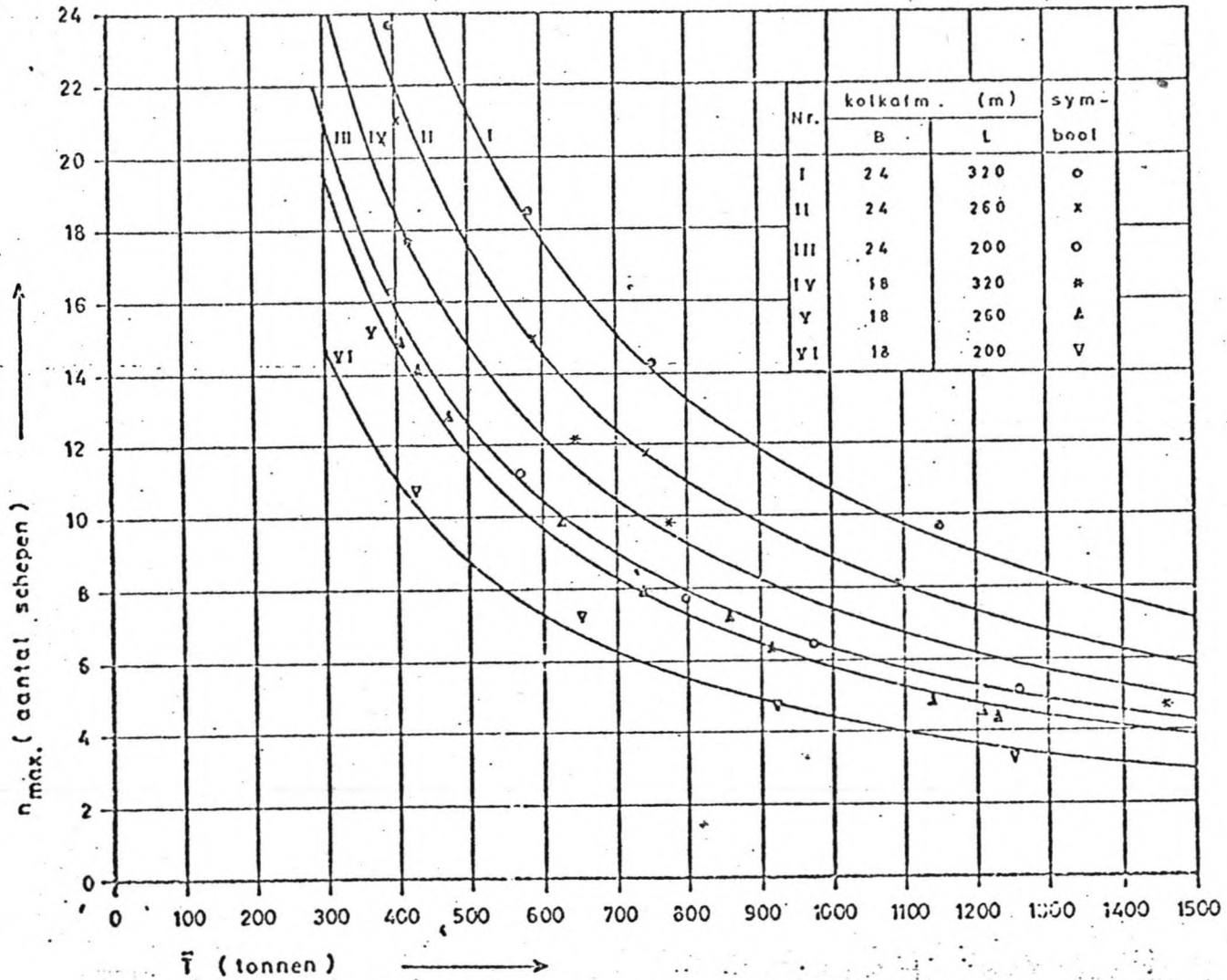
Nr. 70.12.7



MAXIMALE KOLKBEZETTING  $n_{max}$  ALS FUNKTIE VAN  $\bar{T}$  GEBASEERD OP SIMULATIES VAN DE KOLKINDELING (MIDDELGROTE EN KLEINE SLUIZEN)

Fig. 8





MAXIMALE KOLKREZETTING  $n_{max}$  ALS FUNKTIE VAN  $\bar{T}$  GEBASEERD OP SIMULATIES VAN DE KOLKINDELING (GROTE SLUIZEN)

Fig. 9

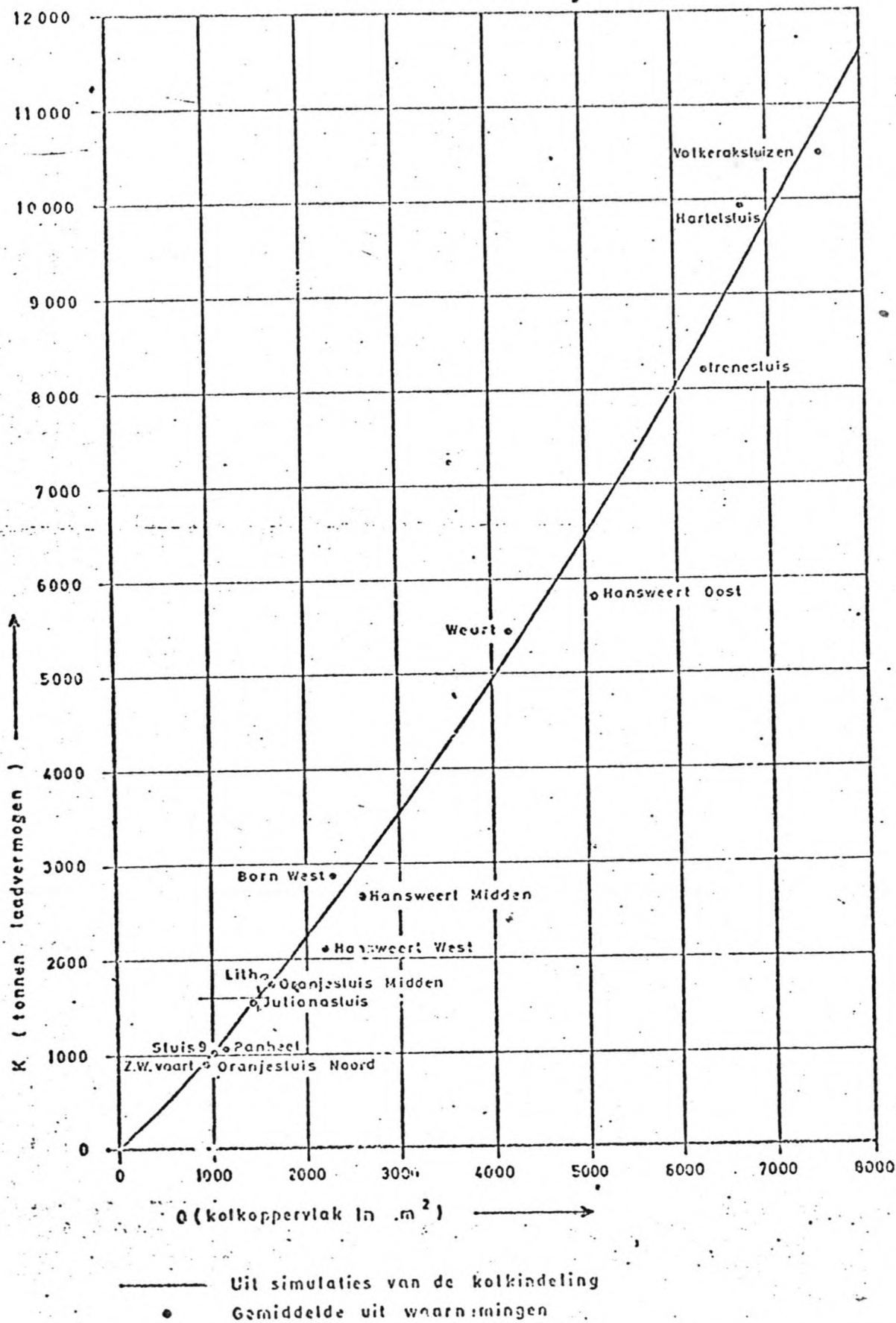
RIJKSWATERSTAAT  
DIENST VERKEERSKUNDE  
HOOFDAFDELING SCHEEPVAART

SCHAAL:

get.	gez.
10-12-73	
J.M.	

Nr. 70.12.7.





VERBAND TUSSEN DE MAXIMALE KOLKBEZETTING, IN  
TOTAAL LAADVERMOGEN, EN HET NUTTIG KOLKOPPER-  
VLAK

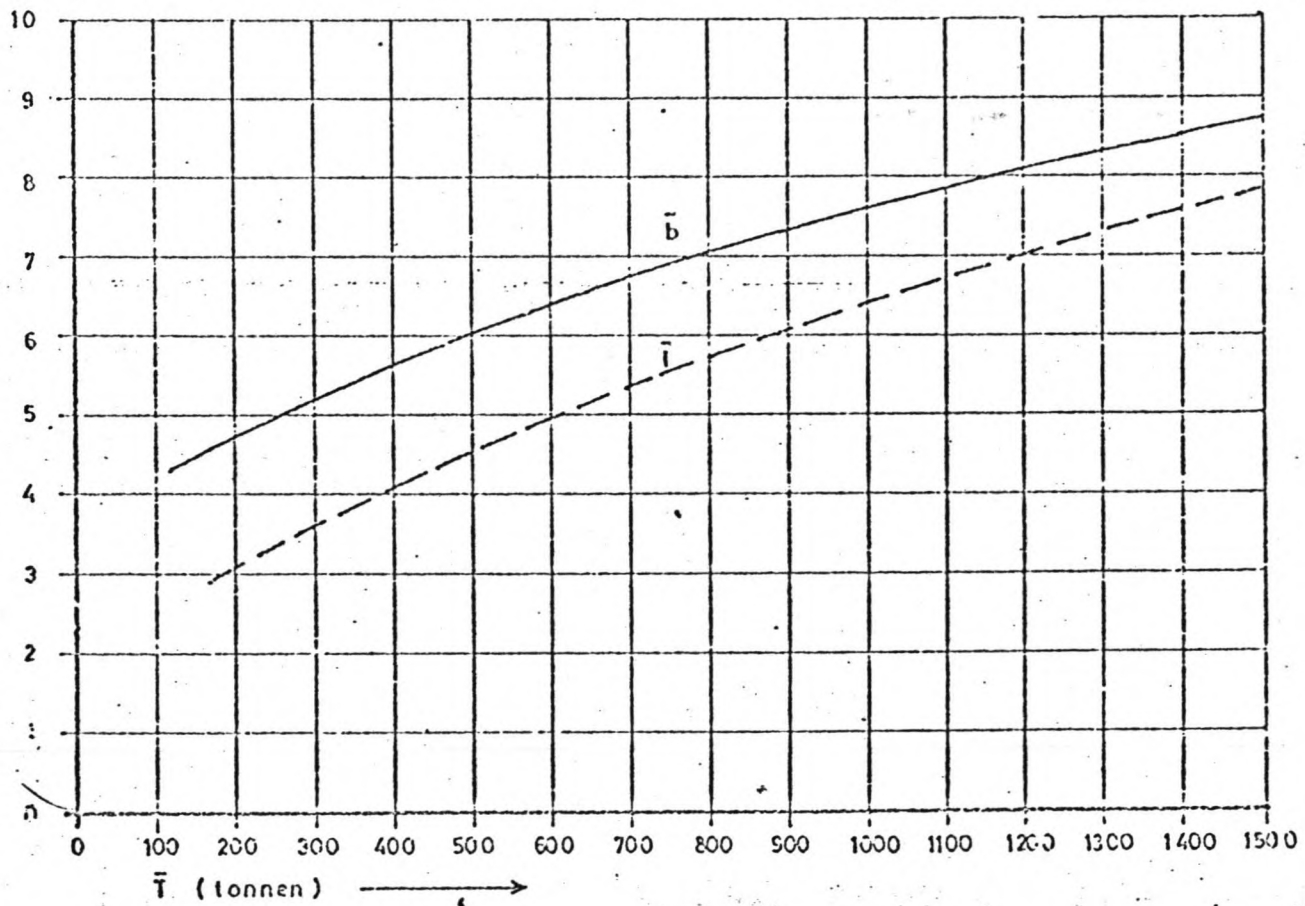
Fig. 10

RIJKSWATERSTAAT  
DIENST VERKEERSKUNDE  
HOOFDAAFDELING SCHEEPVAART

SCHAAL:

get.	gez.
15-2-73	
J.M.	

Nr. 7C.12.7.



GEMIDDELDE BREEDTE ( $\bar{b}$ ) en GEMIDDELDE LENGTE ( $\bar{l}$ ) PER SCHIP  
ALS FUNKTIE VAN HET GEMIDDELDE LAADVERMOGEN ( $\bar{T}$ )

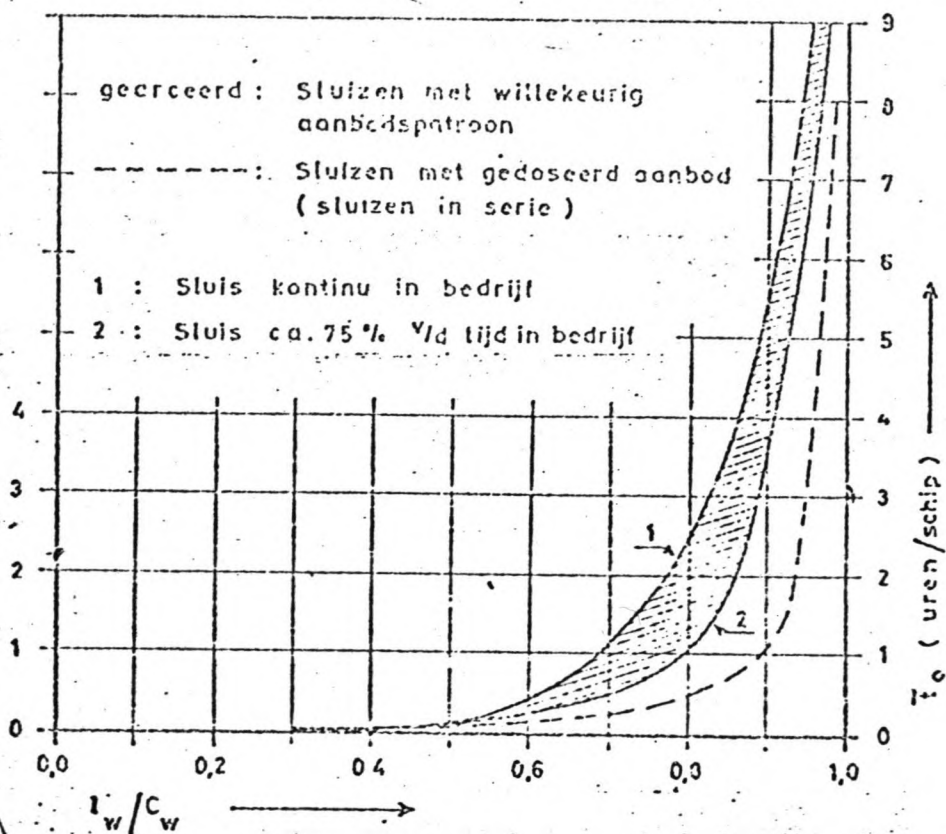
Fig. 11

SCHAAL:

RIJKSWATERSTAAT  
DIENST VERKEERSKUNDE  
HOOFDAFDELING SCHEEPVAART

10-1-77

Nr. 70. 12. 7.



VERBAND TUSSEN DE BELASTINGSGRAAD ( $I_w/C_w$ ) OF WEEKBASIS EN DE GEMIDDELDE OVERLIGTUD PER SCHIP ( $t_o$ )

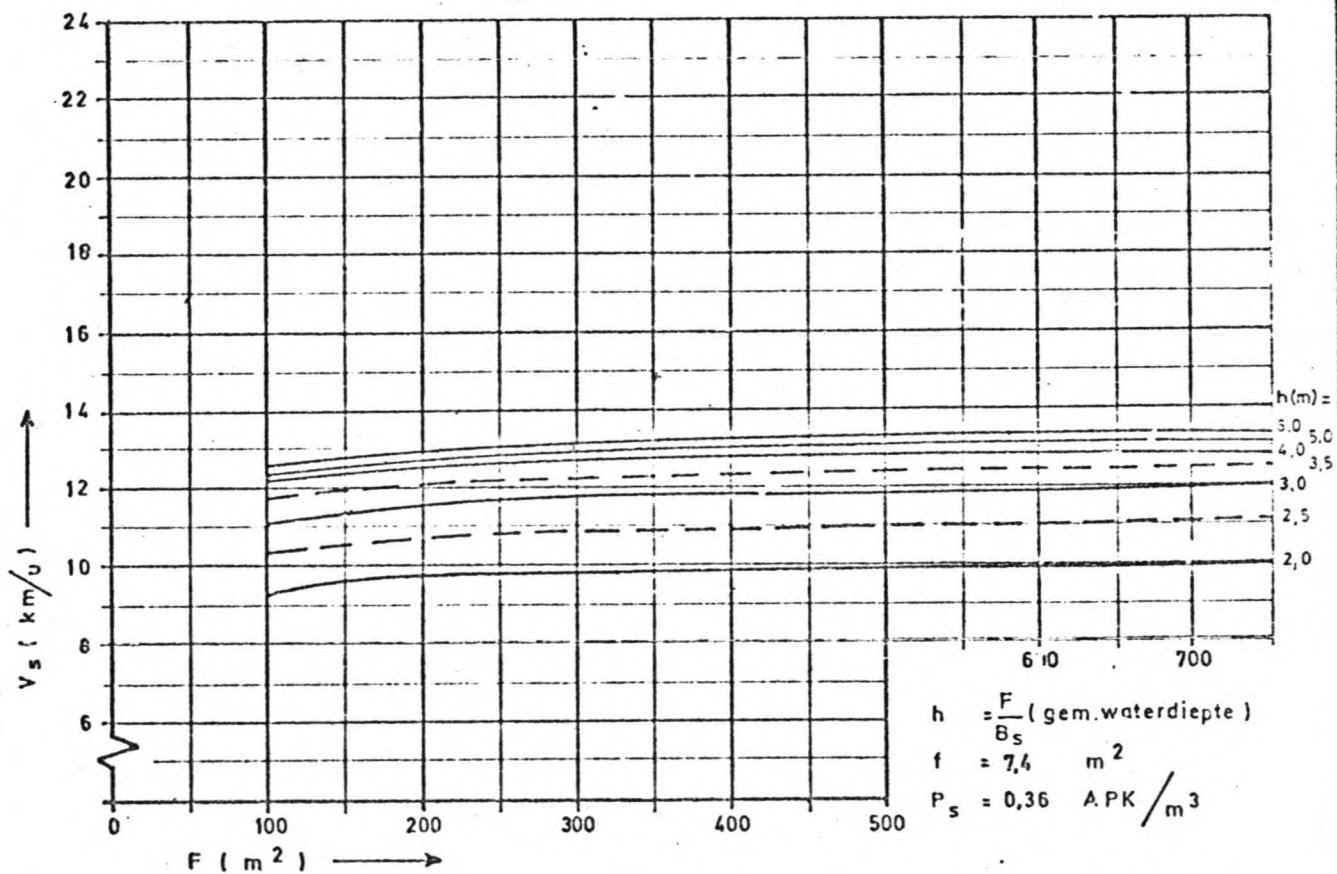
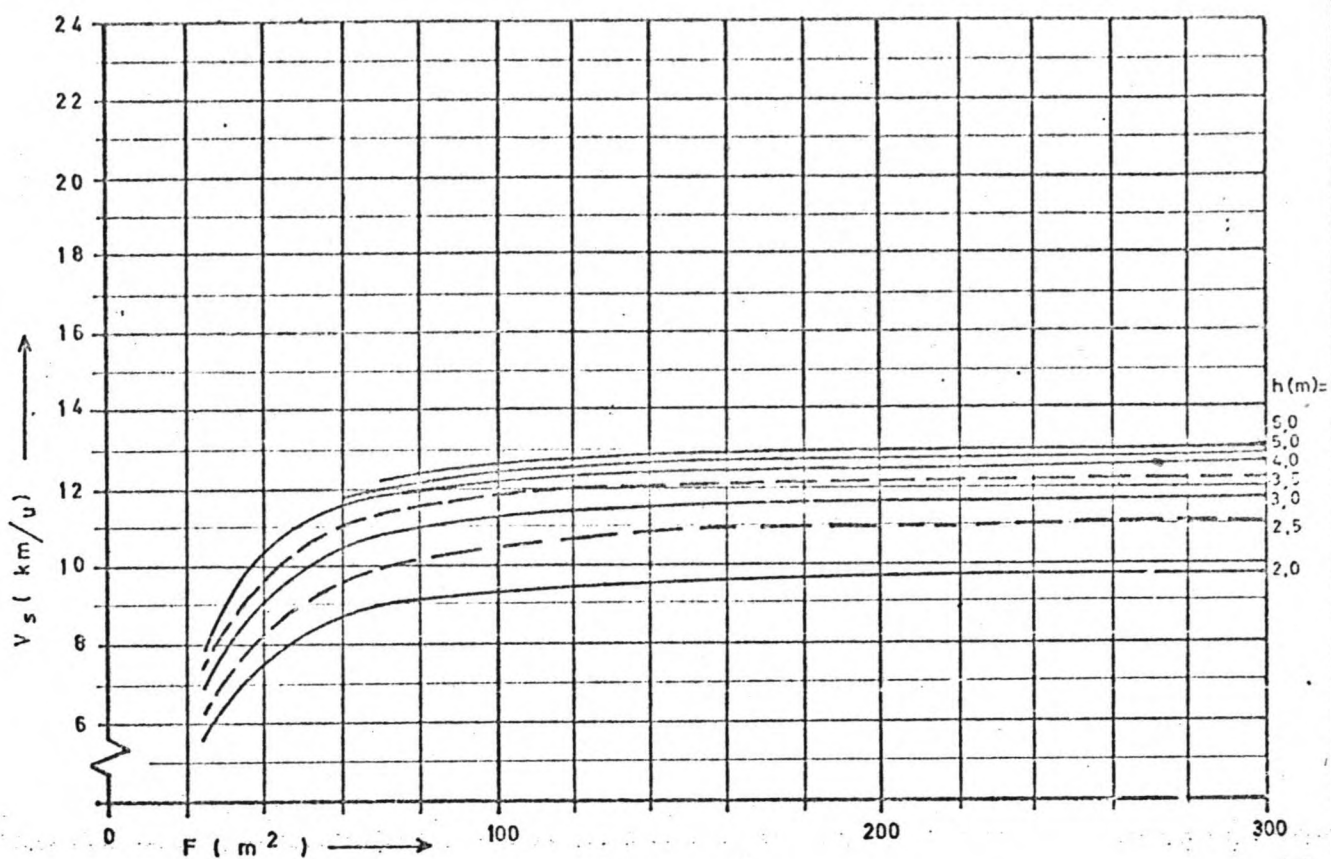
Fig. 12

RIJKSWATERSTAAT  
DIENST VERKEERGRONDE  
HOOFDDELING SCHIPVERVAART

SCHAAL:

50%	100%

Nr. 70.12.7



$h = \frac{F}{B_s}$  (gem. waterdiepte)  
 $f = 7,4 \text{ m}^2$   
 $P_s = 0,36 \text{ APK/m}^3$

( $\lambda$  = beladingsgraad)

STANDAARDSCHIP 1 ( $\lambda = 1,00$ )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG

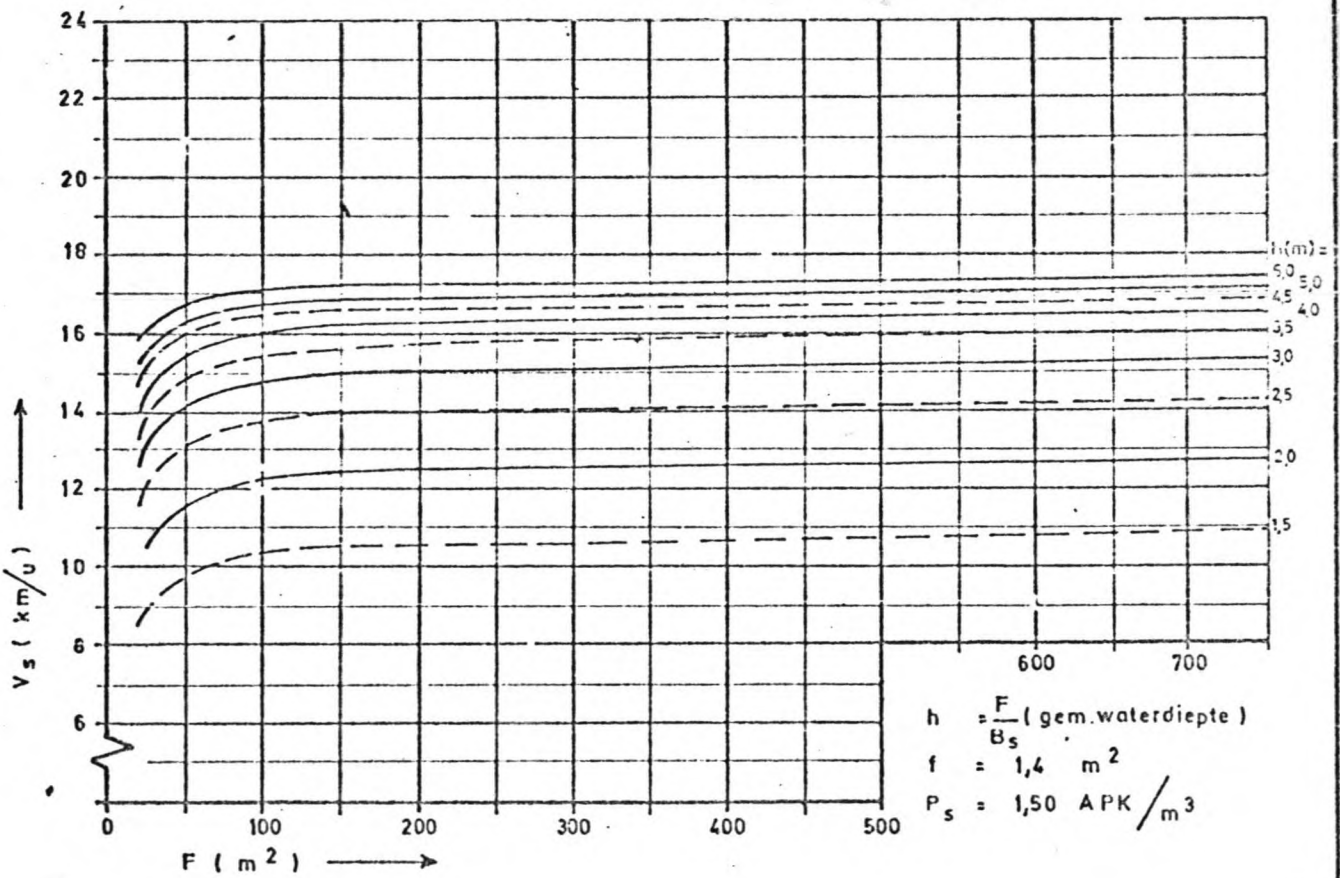
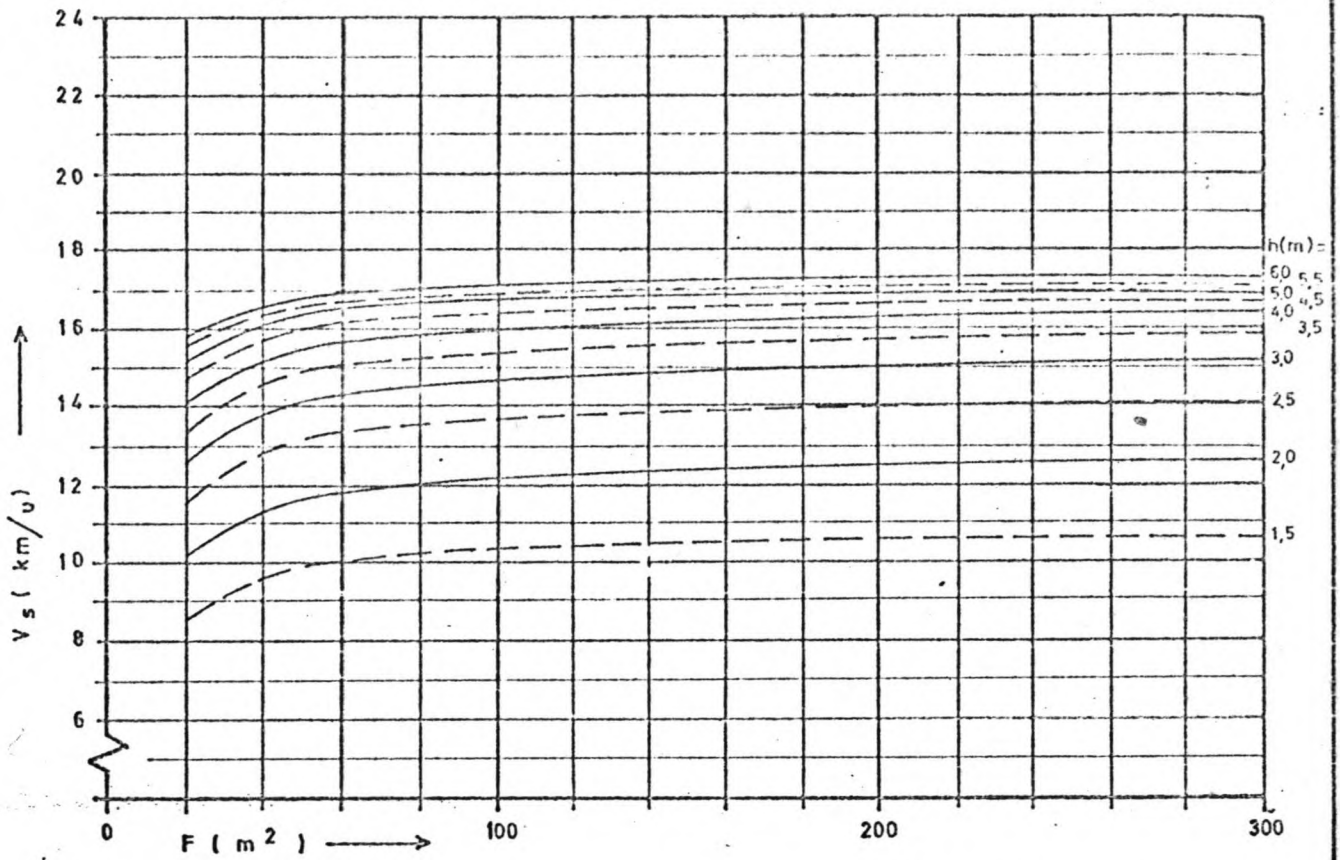
Fig. 2

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RIJKSWATERSTAAT

get.	gez.
21-10-71	
J.M.	

No. 70.12.4





STANDAARDSCHIP 1 (  $\lambda = 0,00$  )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG

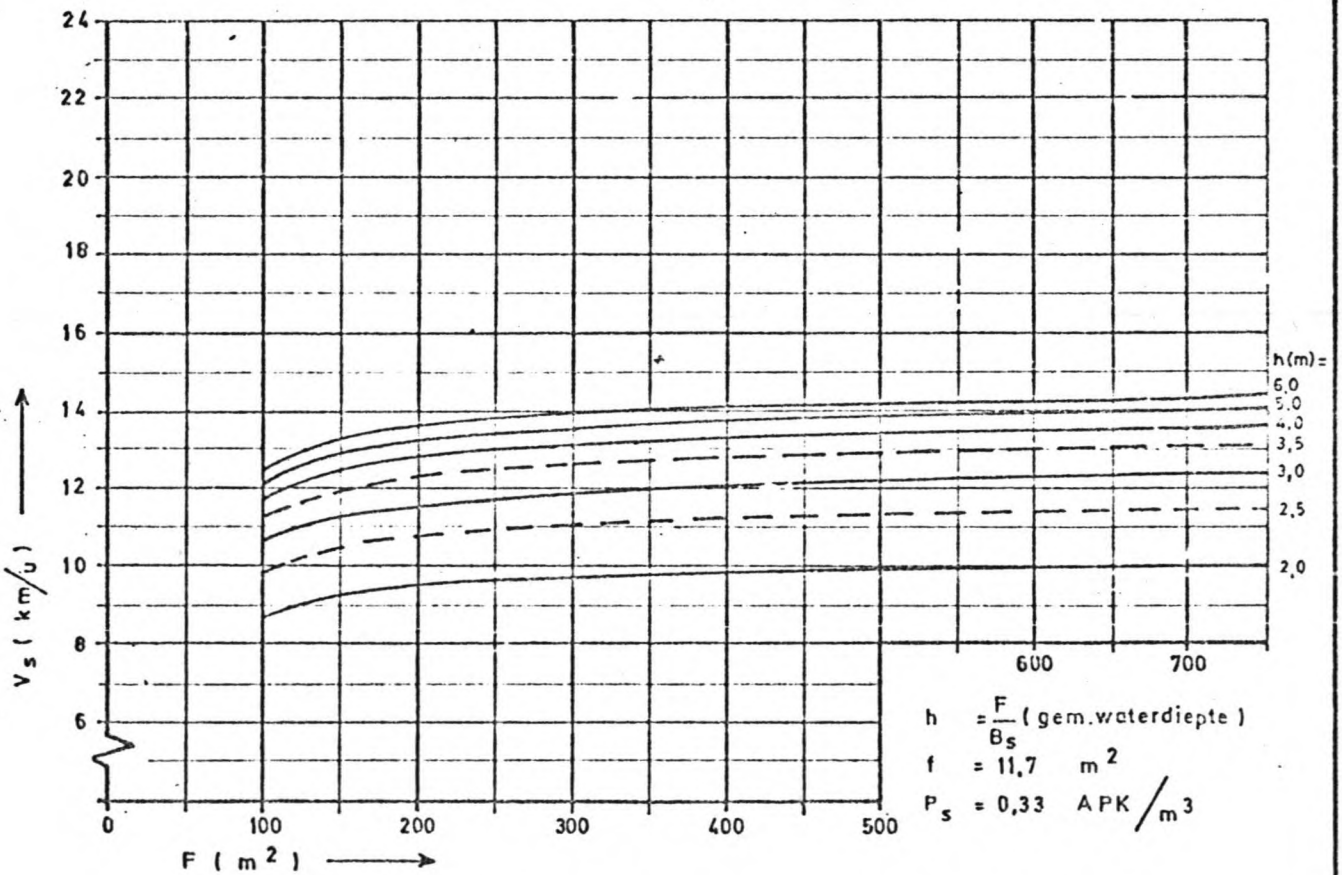
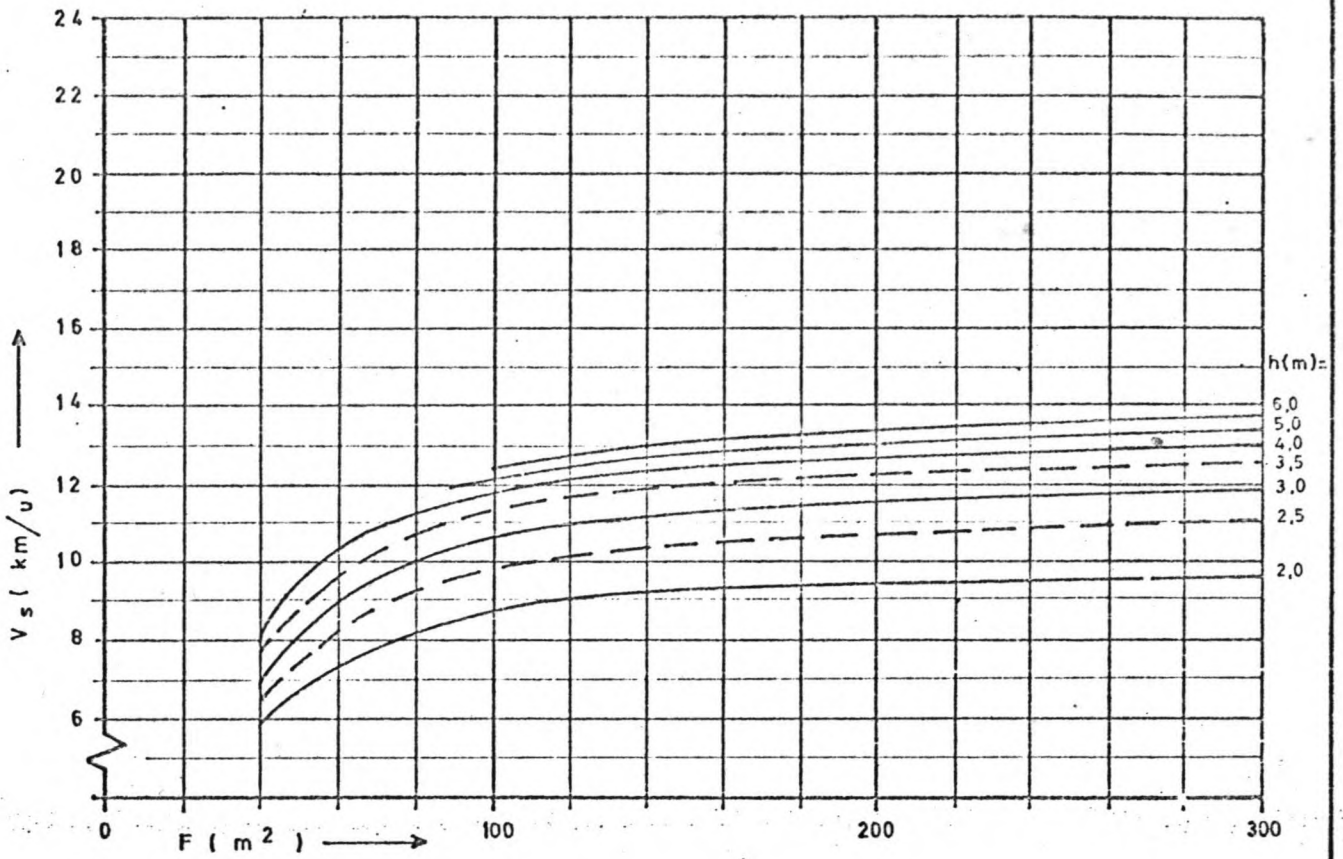
Fig. 4

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RIJKSWATERSTAAT

get. 21-10-71  
 gez. J.M.

No. 70.12.4





(  $\lambda$  = beladingsgraad )

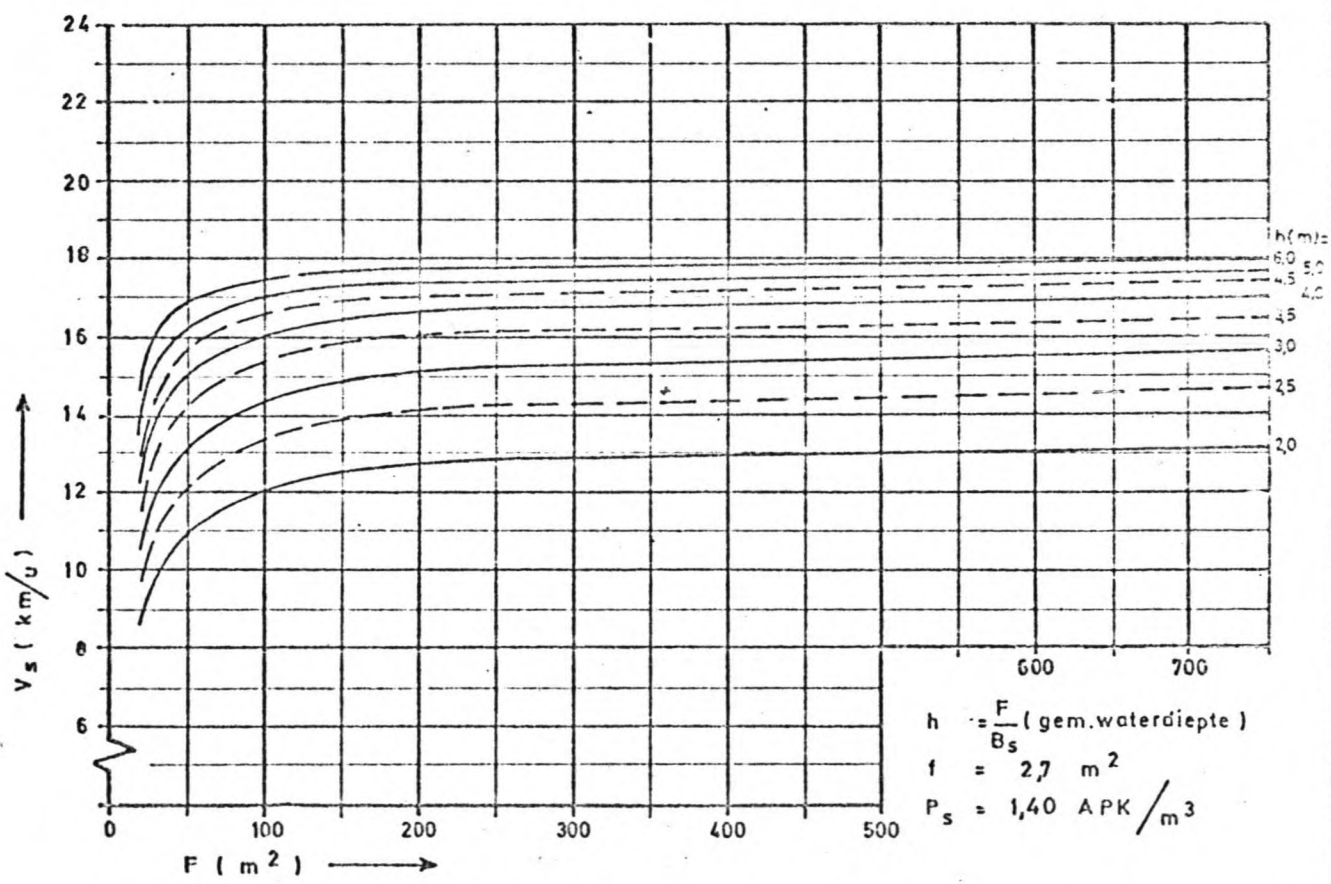
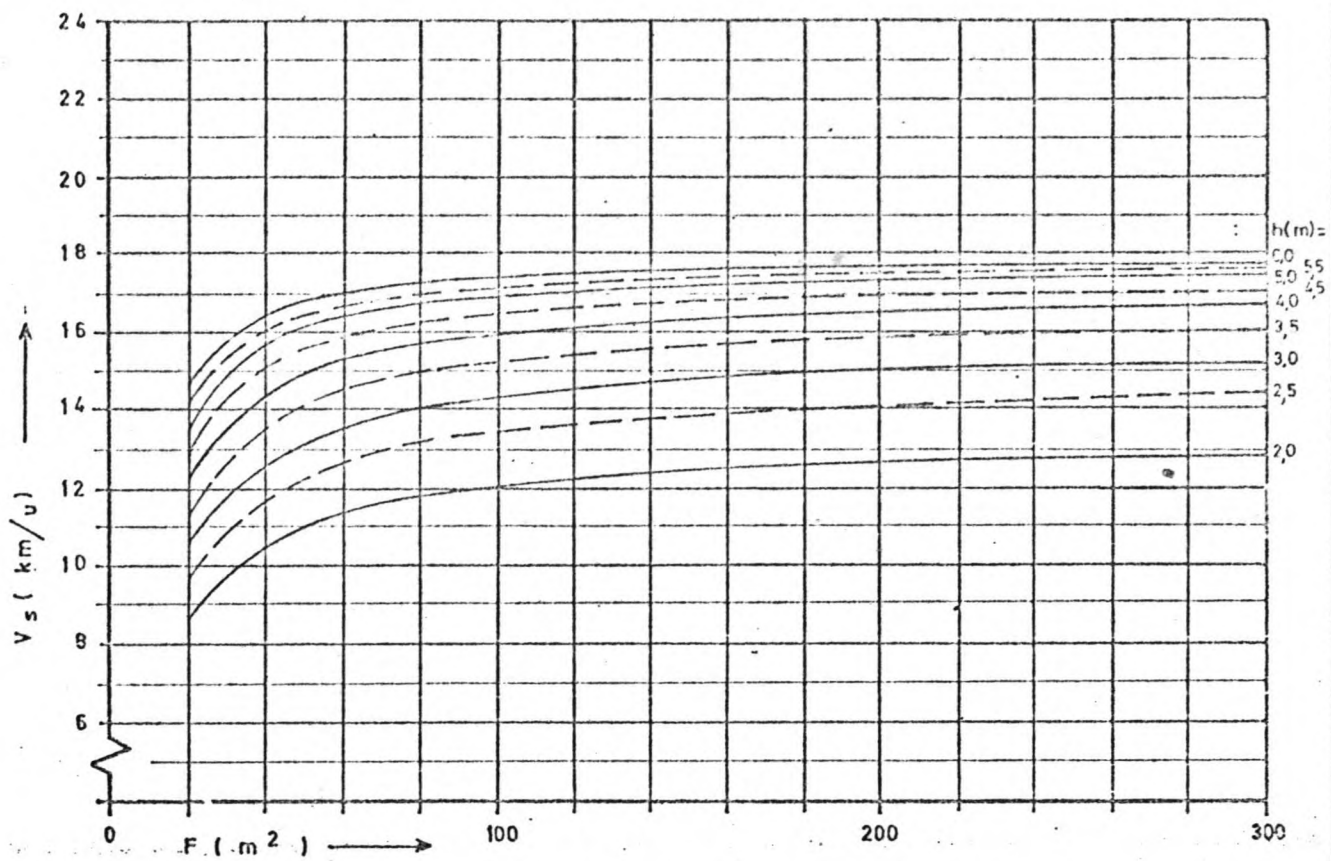
STANDAARDSCHIP 2 (  $\lambda = 100$  )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG

Fig. 5.

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RUKSWATERSTAAT

get.	gez.
21-10-71	
J.M.	

No. 70.12.4



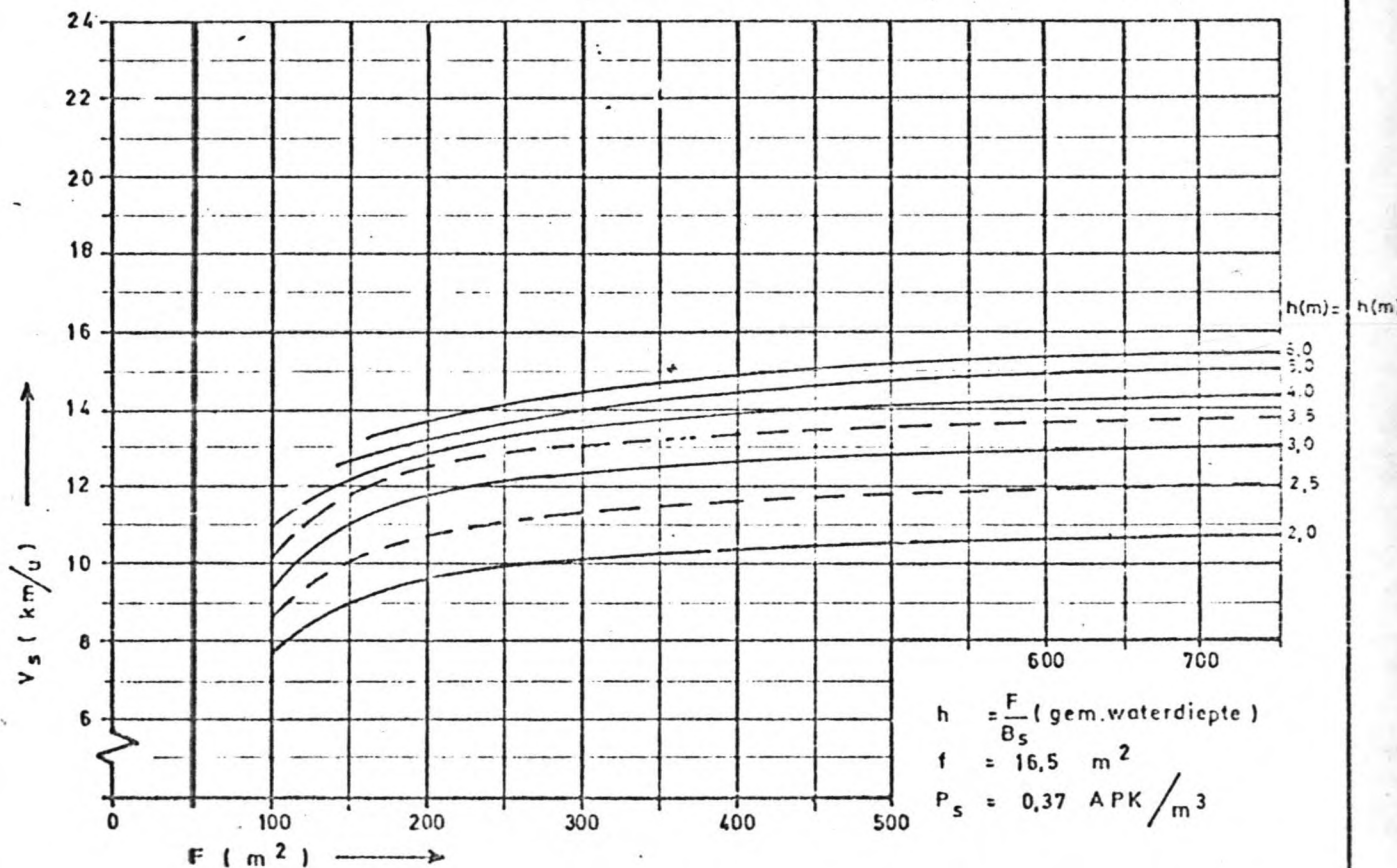
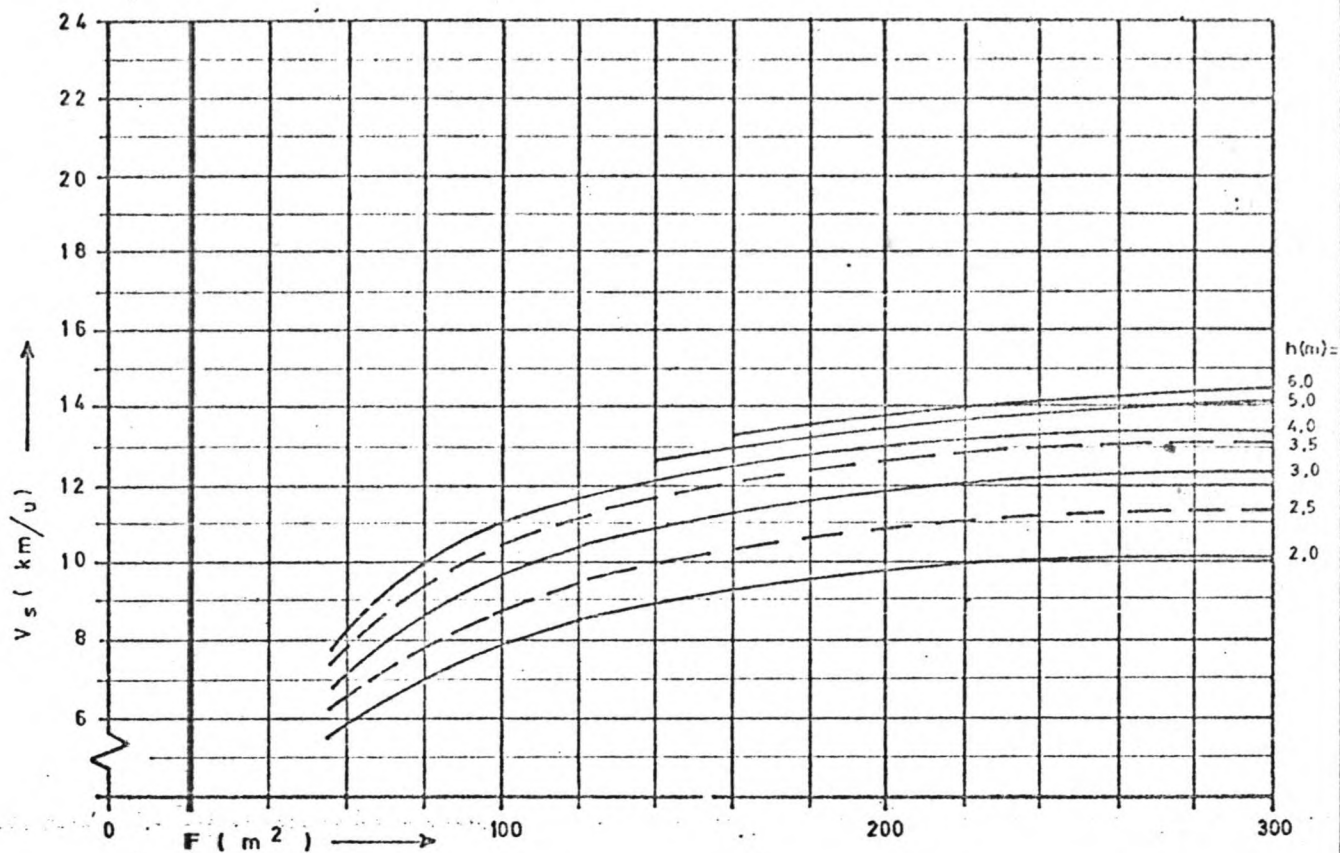
STANDAARDCHIP 2 ( $\lambda = 0,00$ )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG

Fig. 8

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RUKSWATERSTAAT

get.	gez.
21-10-71	
J.M.	

No. 70.12.4



$h = \frac{F}{B_s}$  (gem. waterdiepte)  
 $f = 16,5 \text{ m}^2$   
 $P_s = 0,37 \text{ APK/m}^3$

( $\lambda$  = beladingsgraad)

STANDAARDSCHIP 3 ( $\lambda = 1,00$ )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG.

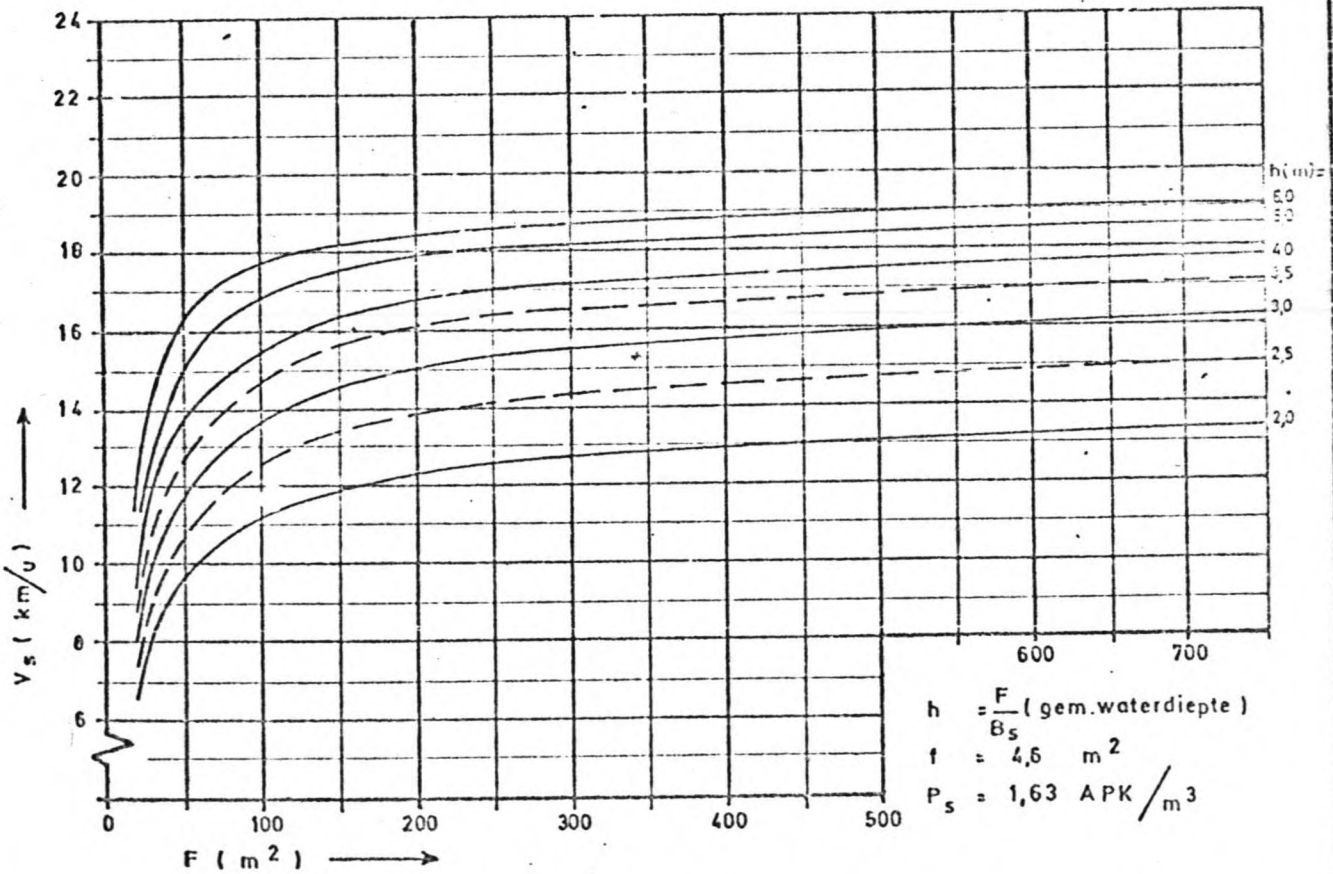
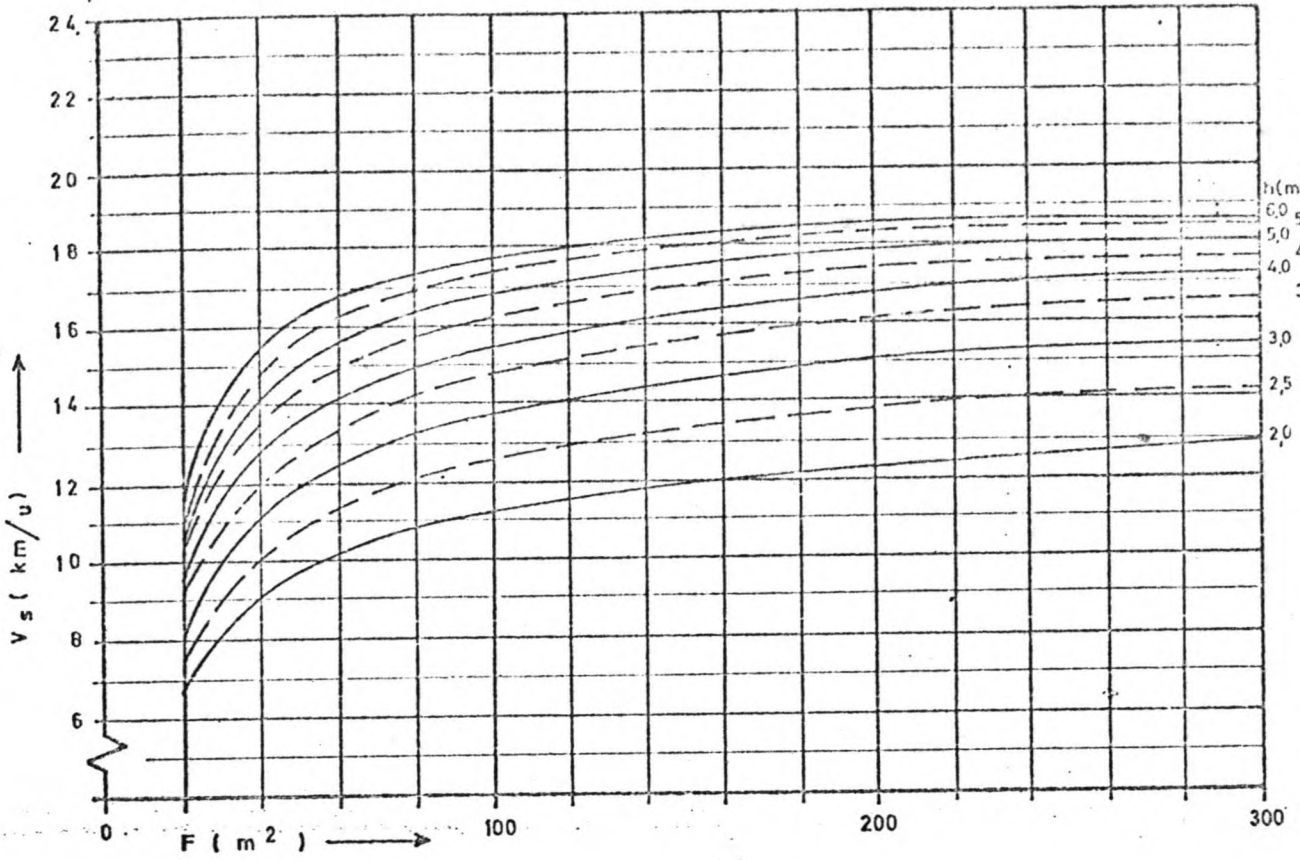
Fig. 13

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RIJKSWATERSTAAT

get.	gez.
21-10-71	
J.M.	

No. 70.12.4





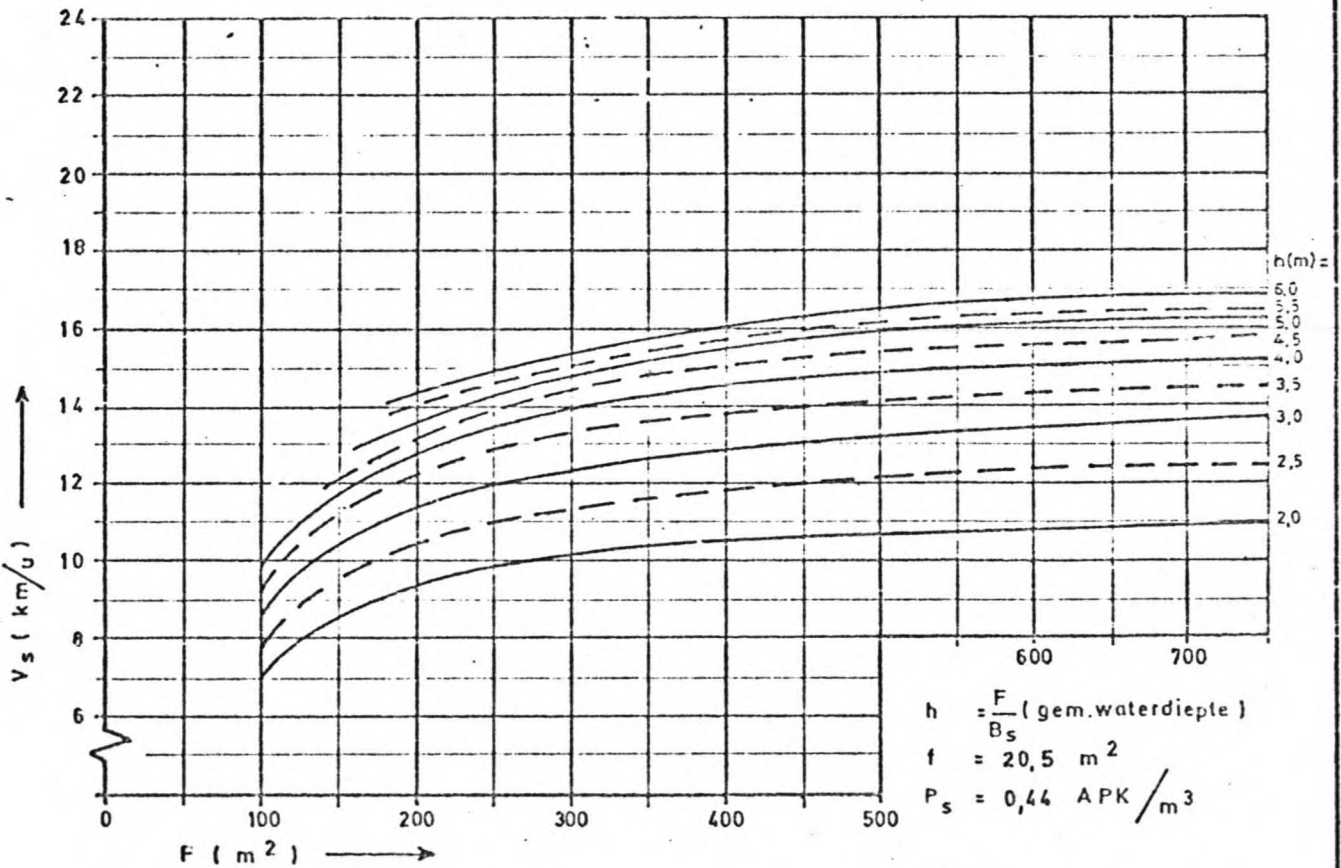
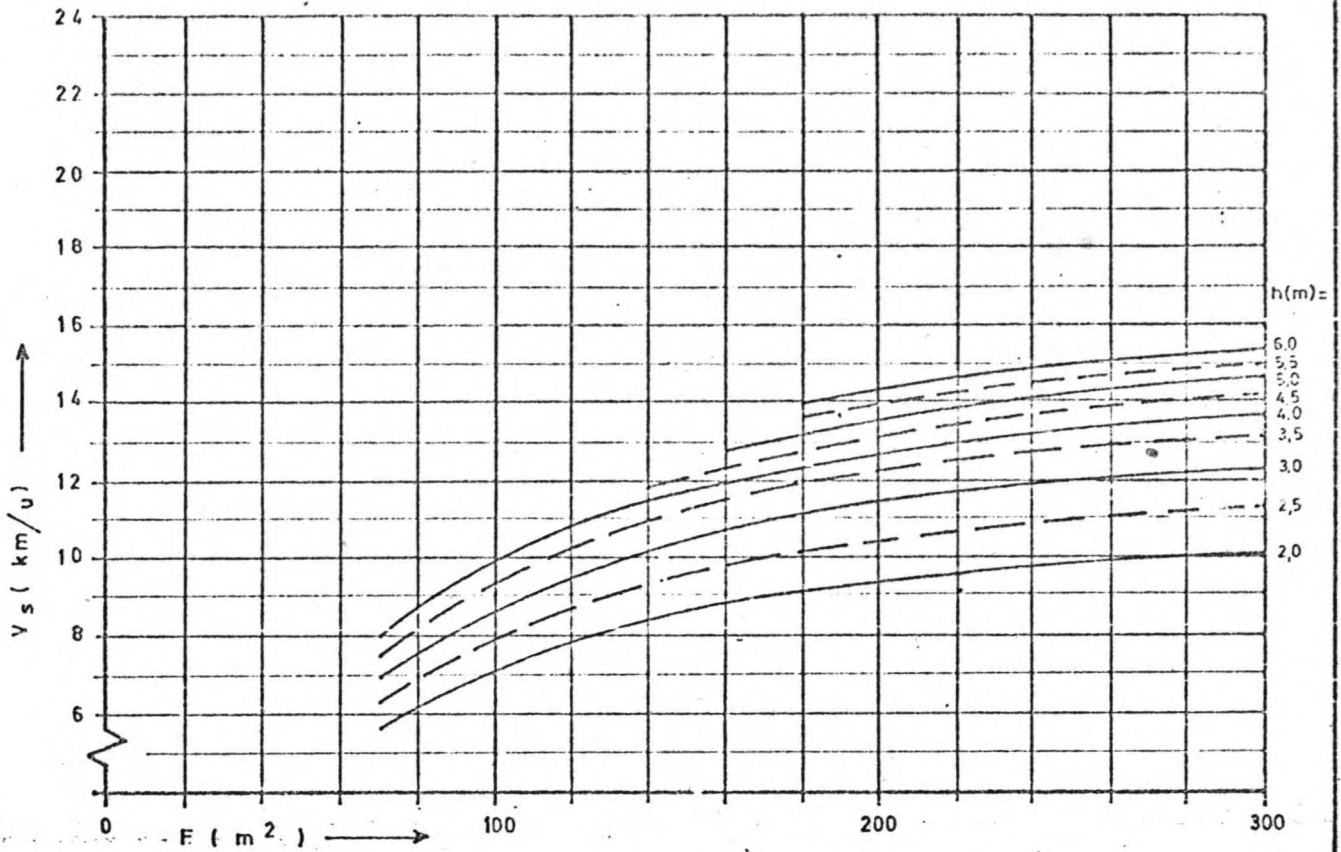
STANDAARDSCHIP 3 ( $\lambda = 0,00$ )  
 VERBAND TUSSEN VAARSNELHEID EN DWAKSPROFIEL  
 VAN DE VAARWEG

Fig. 16

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RUKSWATERSTAAT

get.	gez.
21-10-71	
J.M.	

No. 70.12.4



$h = \frac{F}{B_s}$  (gem. waterdiepte)  
 $f = 20,5 \text{ m}^2$   
 $P_s = 0,44 \text{ APK/m}^3$

( $\lambda =$  beladingsgraad)

STANDAARDSCHIP 4 ( $\lambda = 100$ )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG

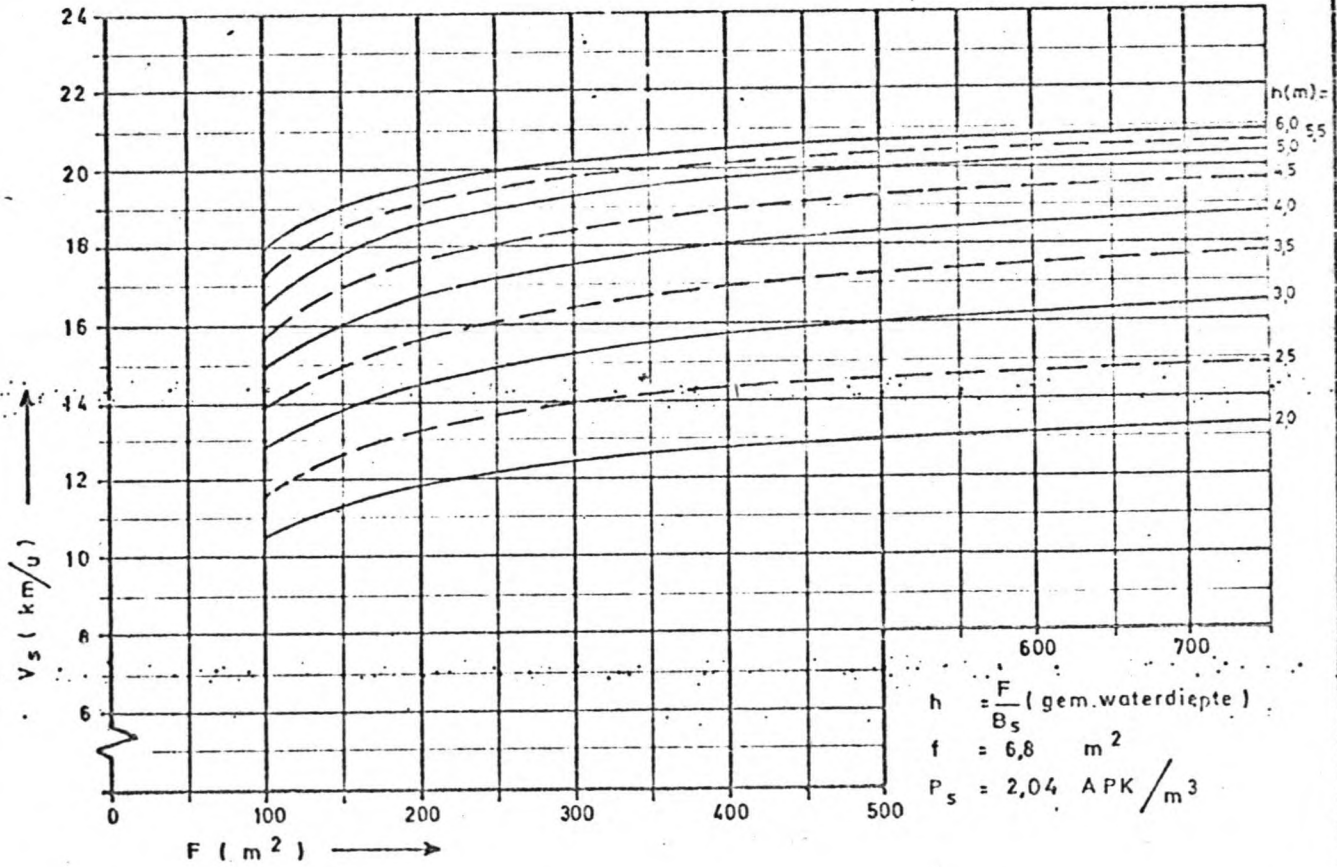
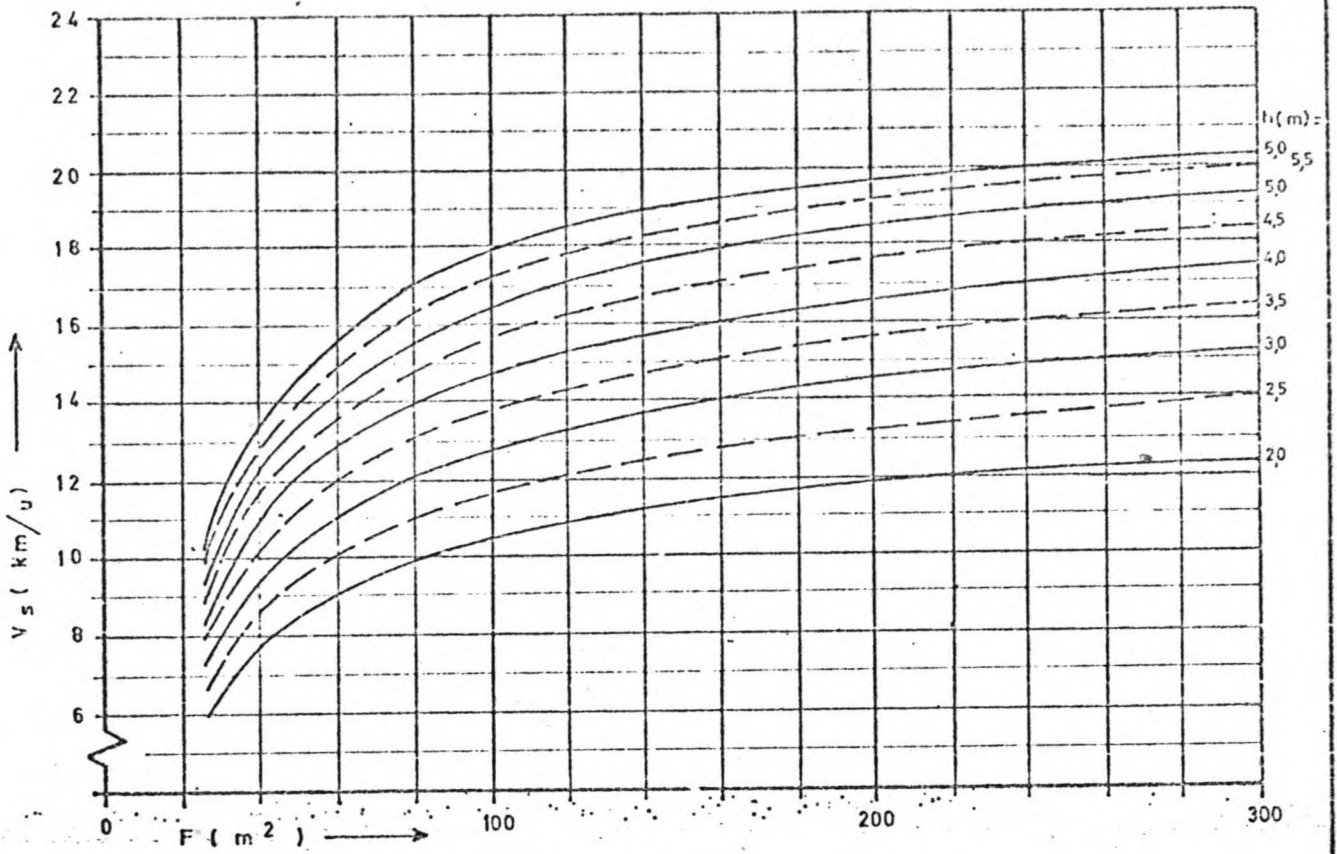
Fig. 17

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RUKSWATERSTAAT

get.	gez.
21-10-71	
J.M.	

No. 70.12.4





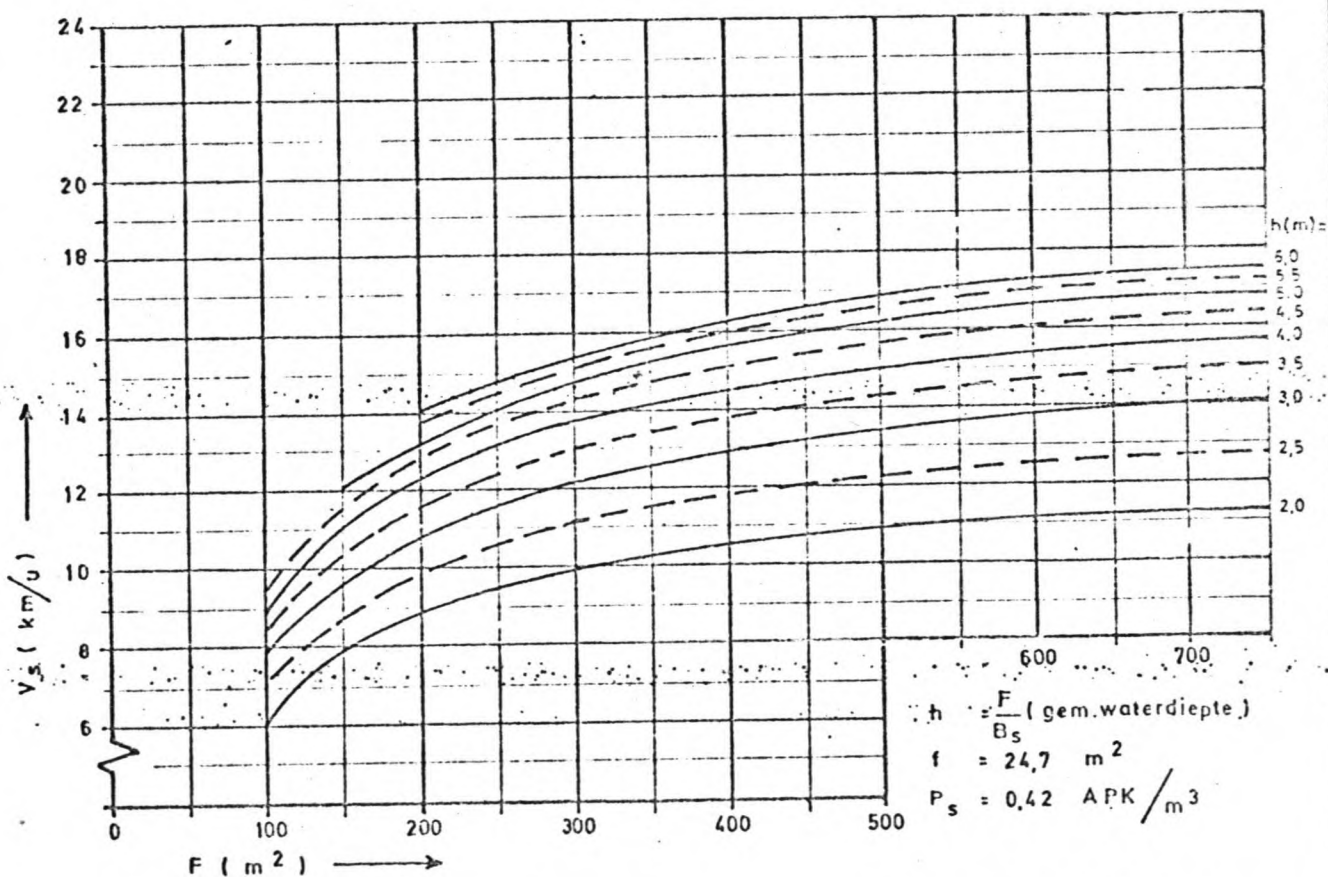
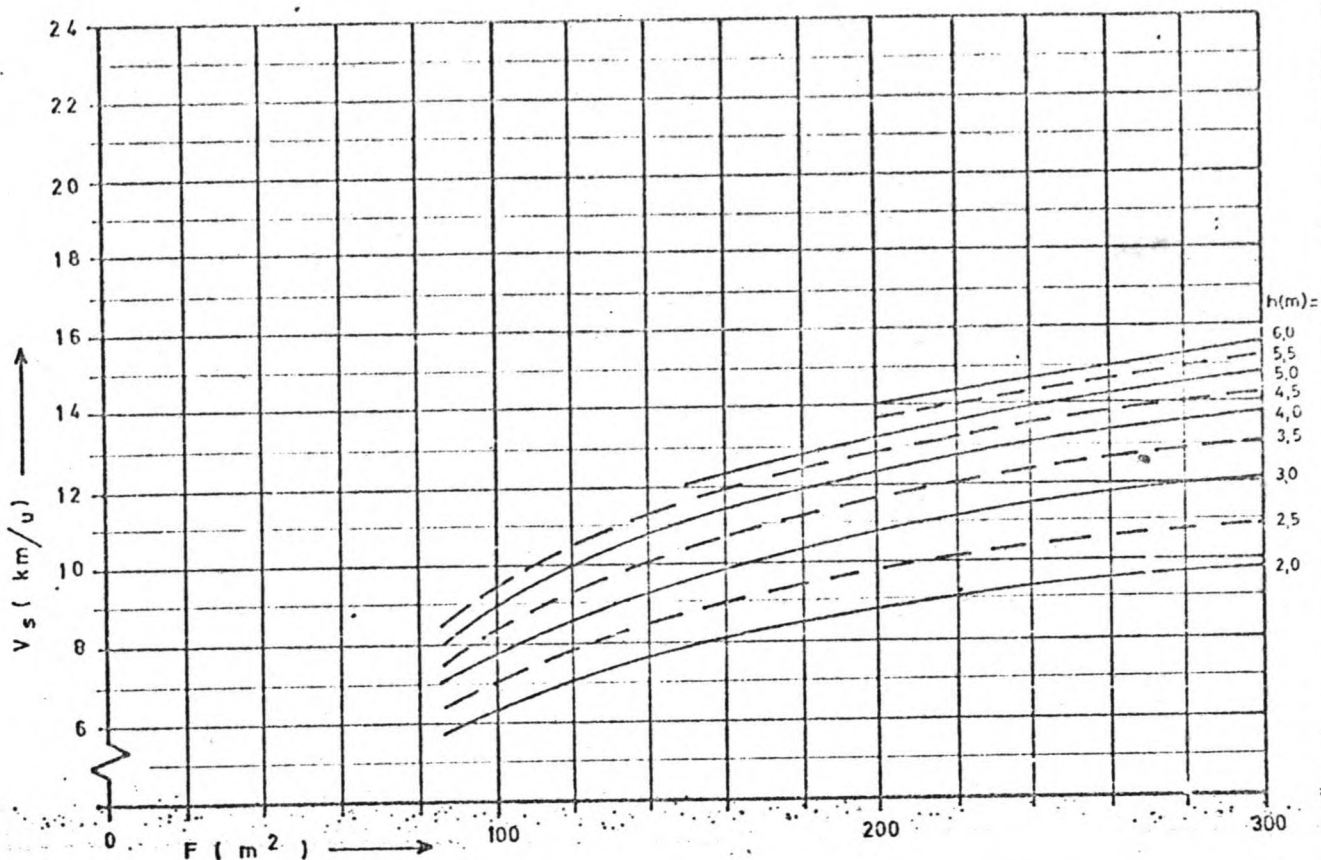
STANDAARDSCHIP 4 ( $\lambda = 0,00$ )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG

Fig.20

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RUKSWATERSTAAT.

get.	gez.
21-10-71	
J.M.	

No. 70.12.4



$h = \frac{F}{B_s}$  (gem. waterdiepte)  
 $f = 24,7 \text{ m}^2$   
 $P_s = 0,42 \text{ APK/m}^3$

( $\lambda =$  beladingsgraad)

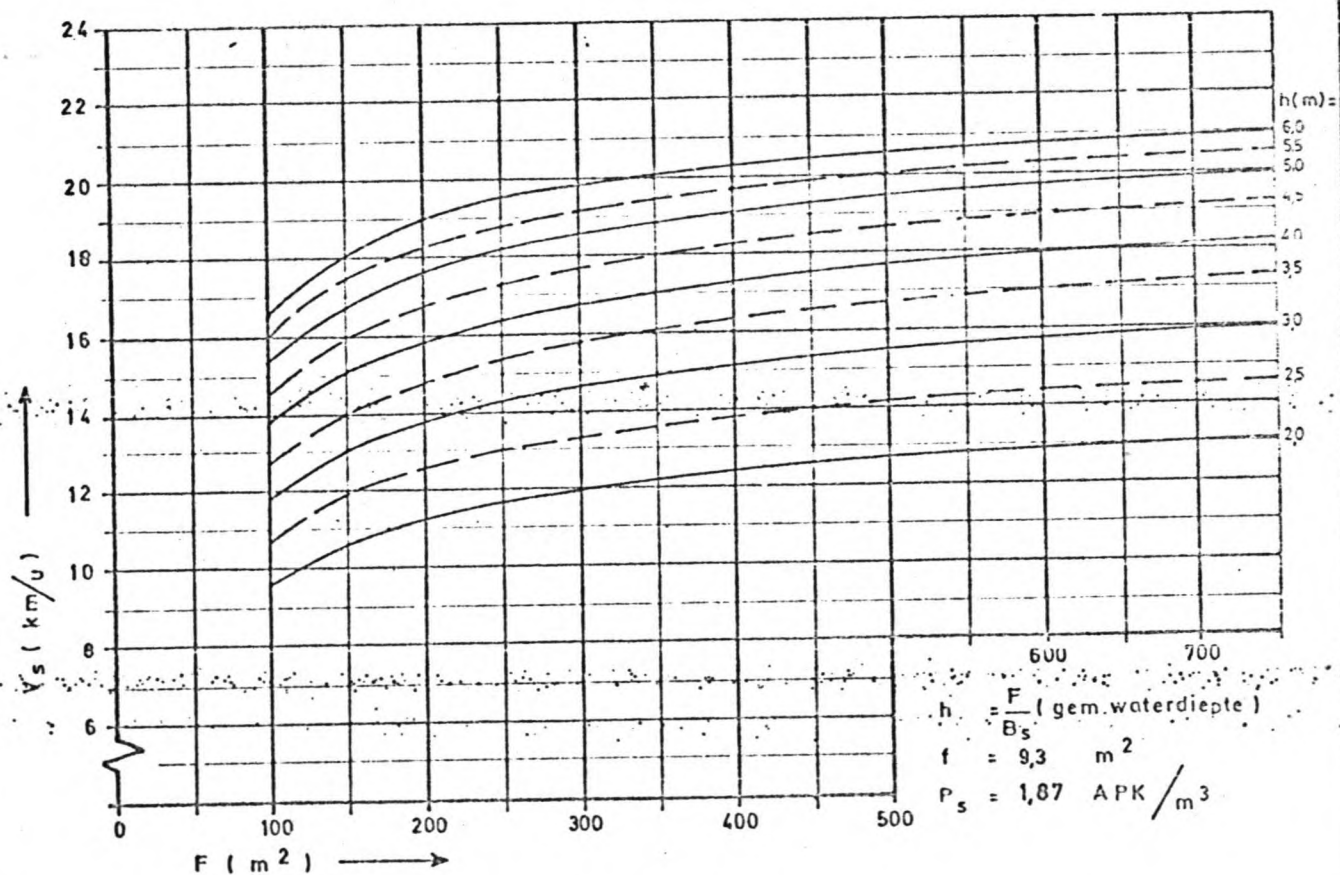
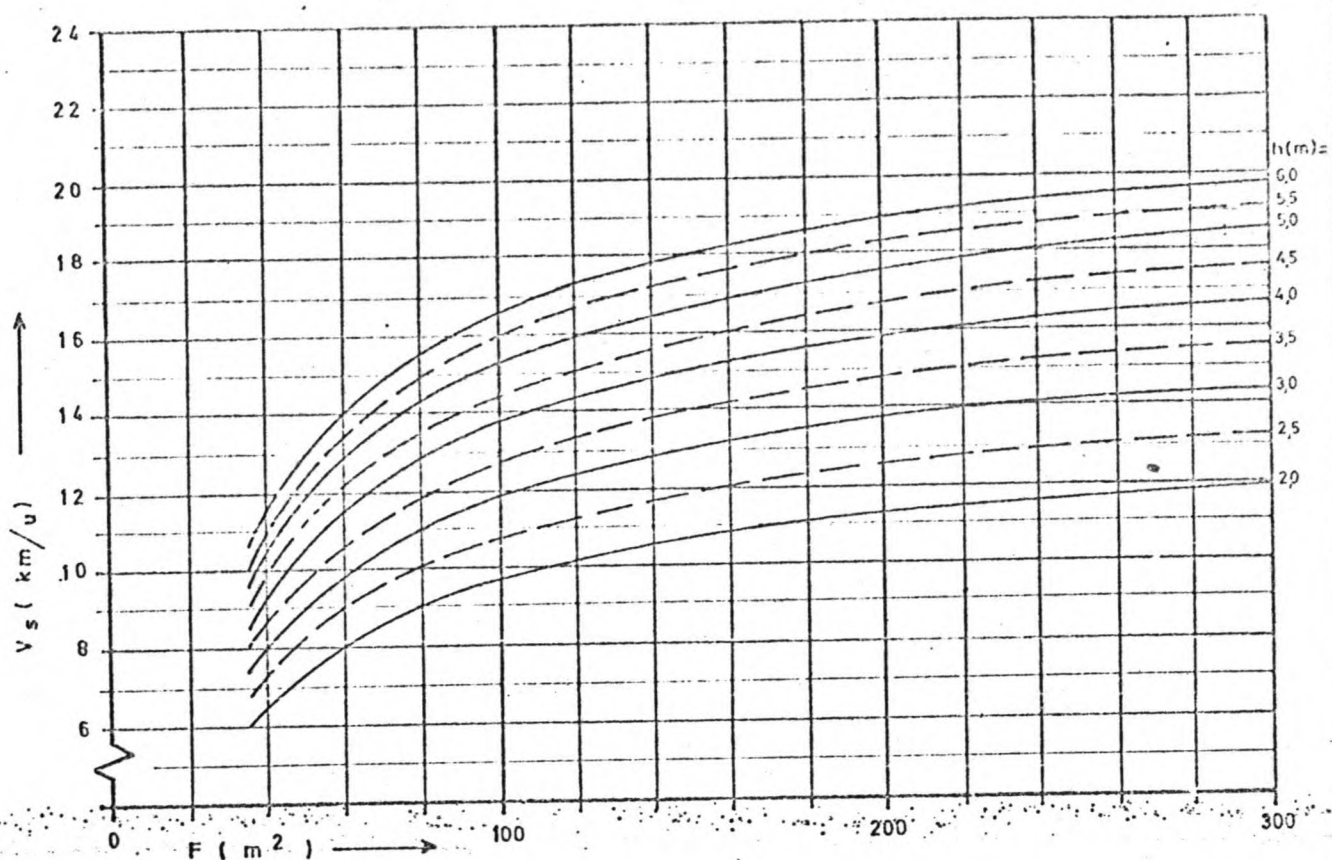
STANDAARDSCHIP 5 ( $\lambda = 100$ )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG

Fig. 21

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RIJKSWATERSTAAT

get.	gez.
21-10-71	
J.M.	

No. 70.12.4



STANDAARDSCHIP 5 ( $\lambda = 0,00$ )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG

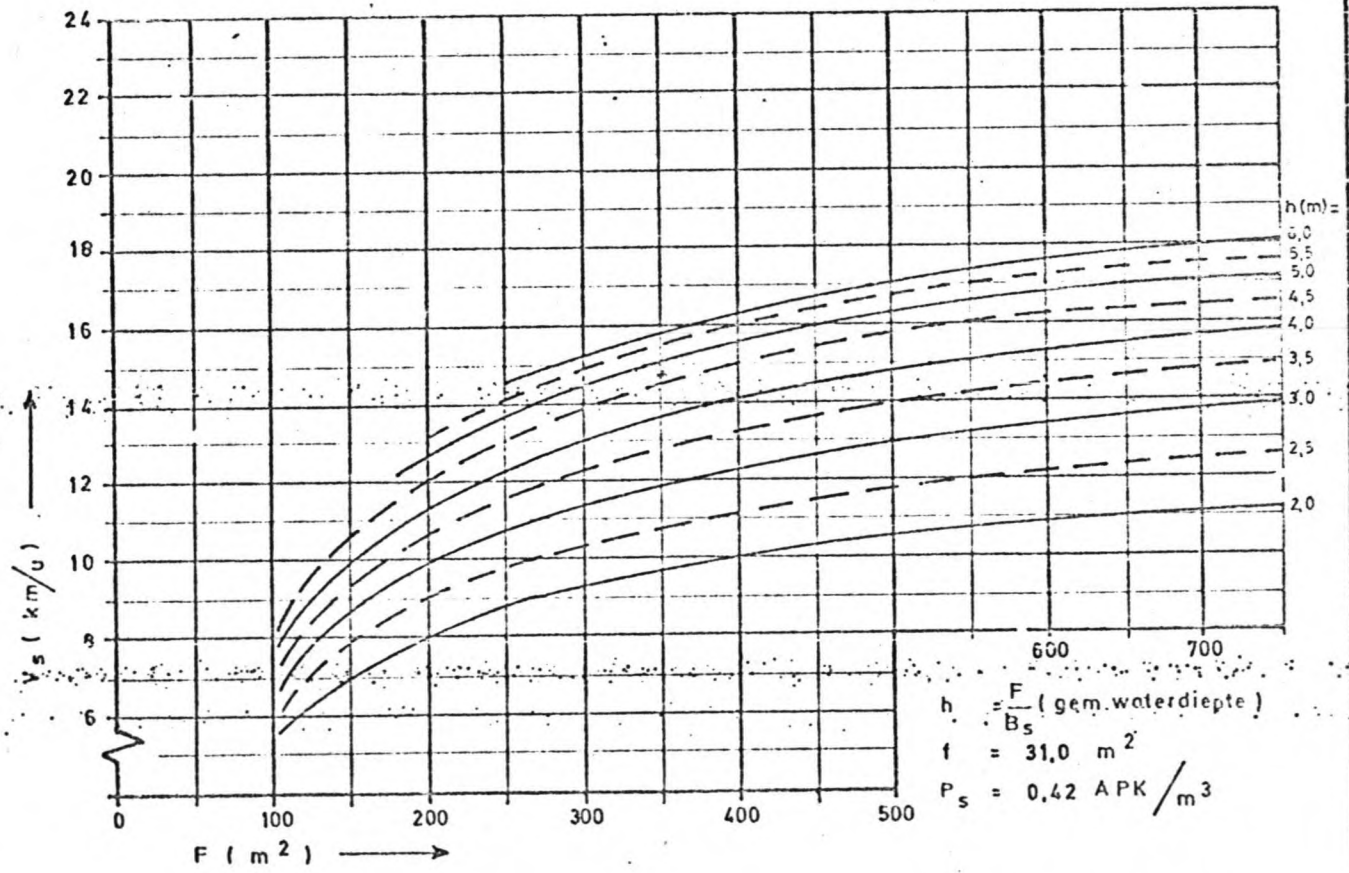
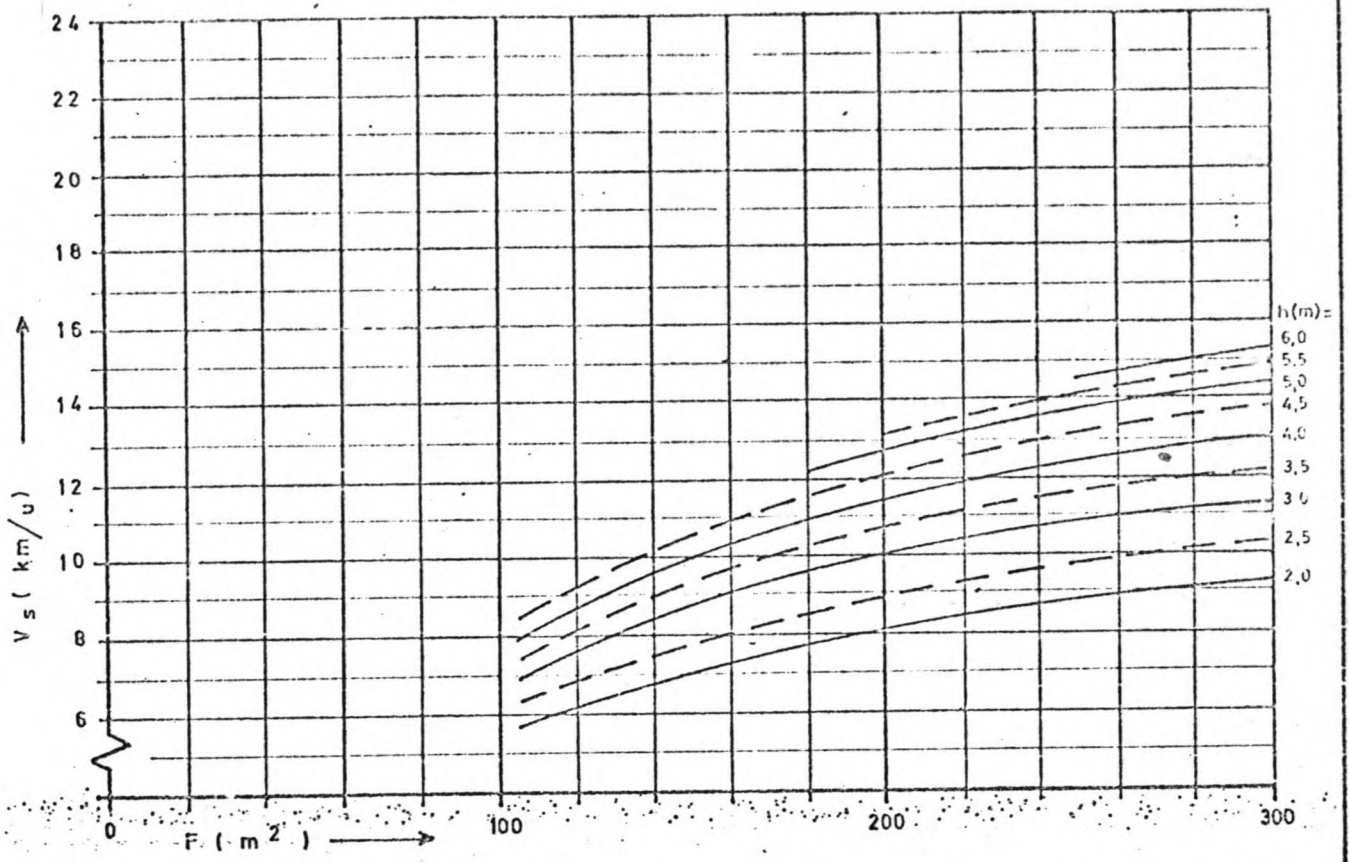
Fig. 24

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RUKSWATERSTAAT

get. 21-10-71  
 gez.  
 J.M.

No. 70.12.4





$h = \frac{F}{B_s}$  (gem. waterdiepte)  
 $f = 31,0 \text{ m}^2$   
 $P_s = 0,42 \text{ APK/m}^3$

( $\lambda$  = beladingsgraad)

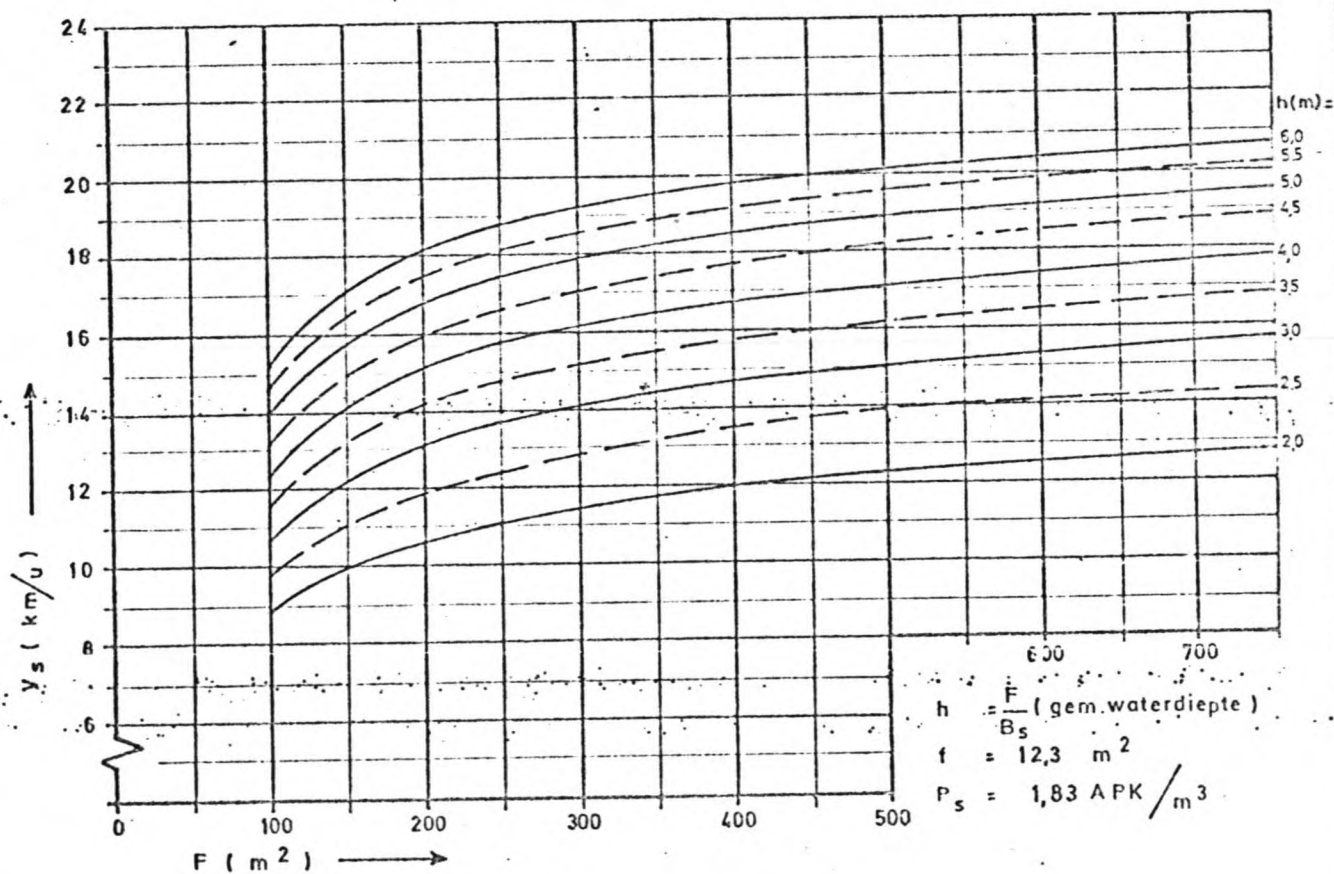
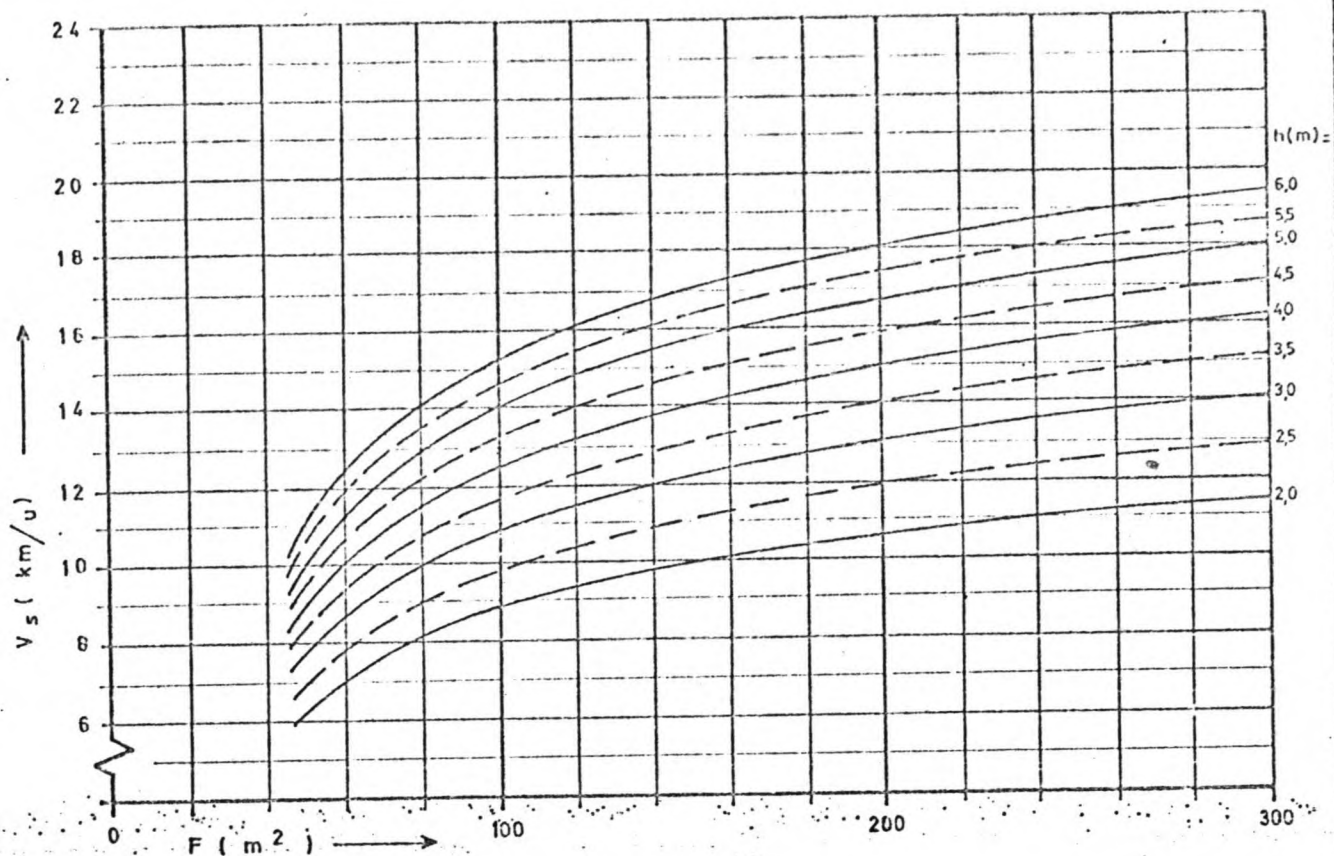
STANDAARDSCHIP 6 ( $\lambda = 1,00$ )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG

Fig. 25

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RIJKSWATERSTAAT

get  
 21-10-71  
 J.M.

gez.  
 No. 70.12.4



STANDAARDSCHIP 6 ( $\lambda = 0,00$ )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG

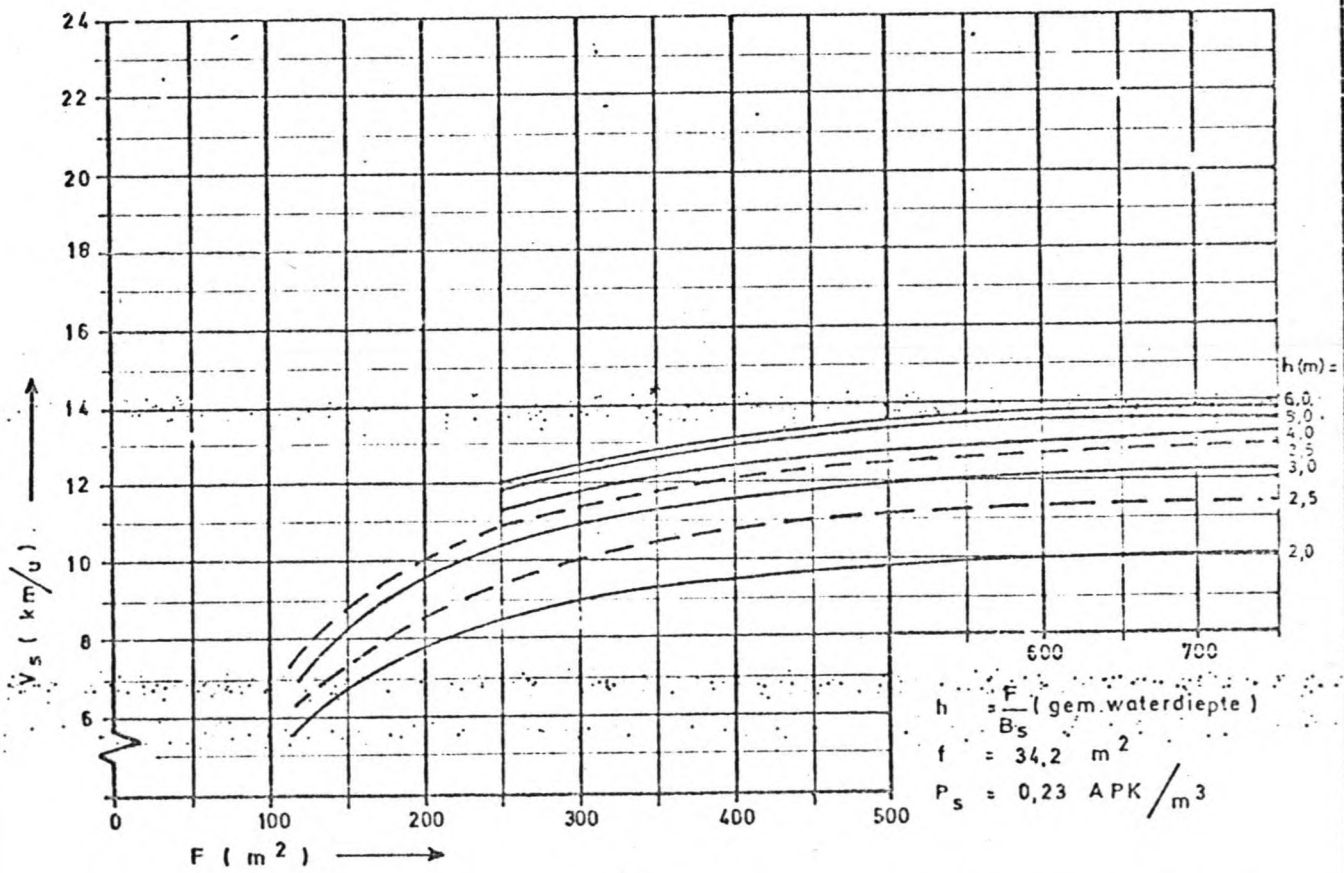
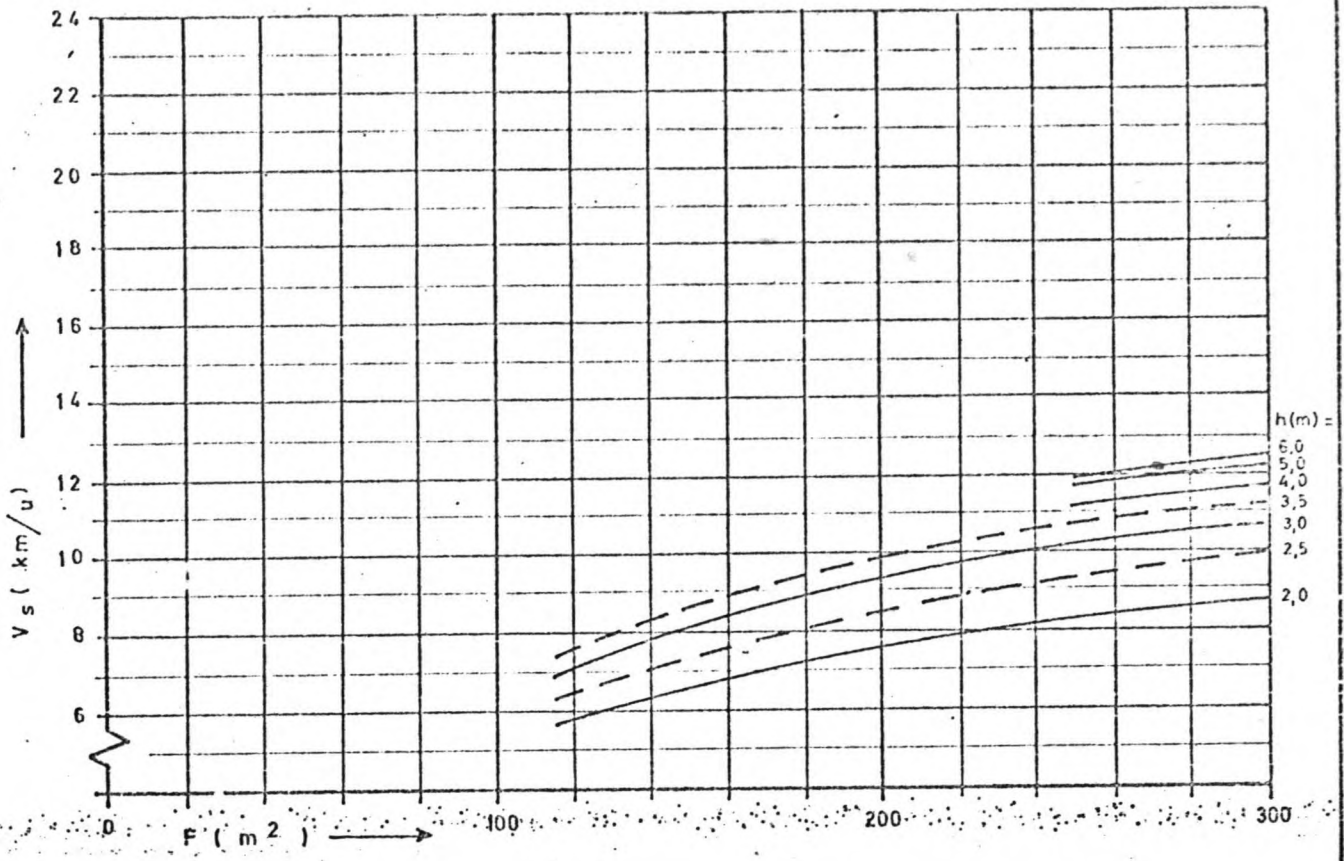
Fig. 27

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RUKSWATERSTAAT

get.	gez.
21-10-71	
J.M.	

No. 70.12.4





$h = \frac{F}{B_s}$  (gem. waterdiepte)  
 $f = 34,2 \text{ m}^2$   
 $P_s = 0,23 \text{ APK/m}^3$

( $\lambda$  = beladingsgraad)

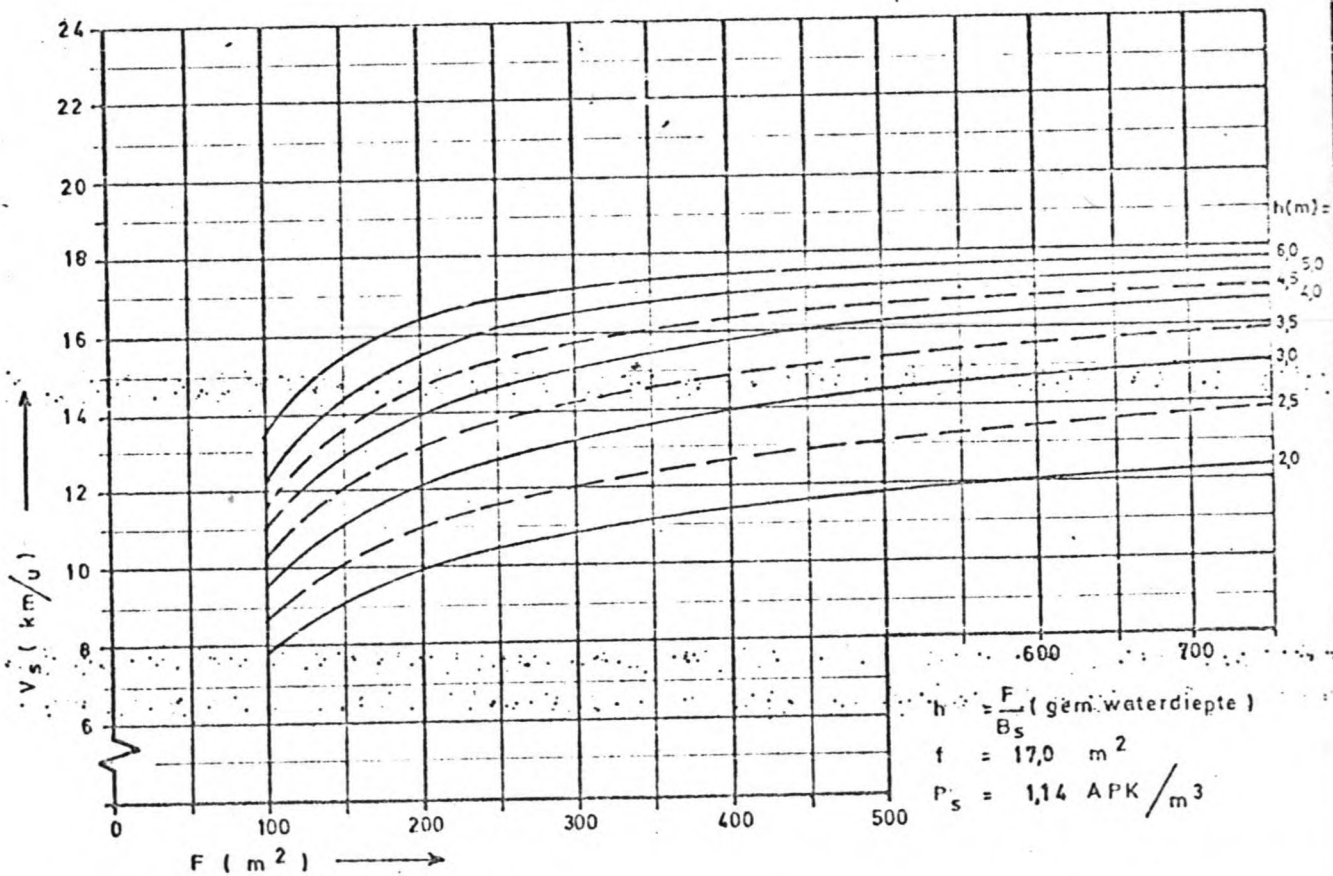
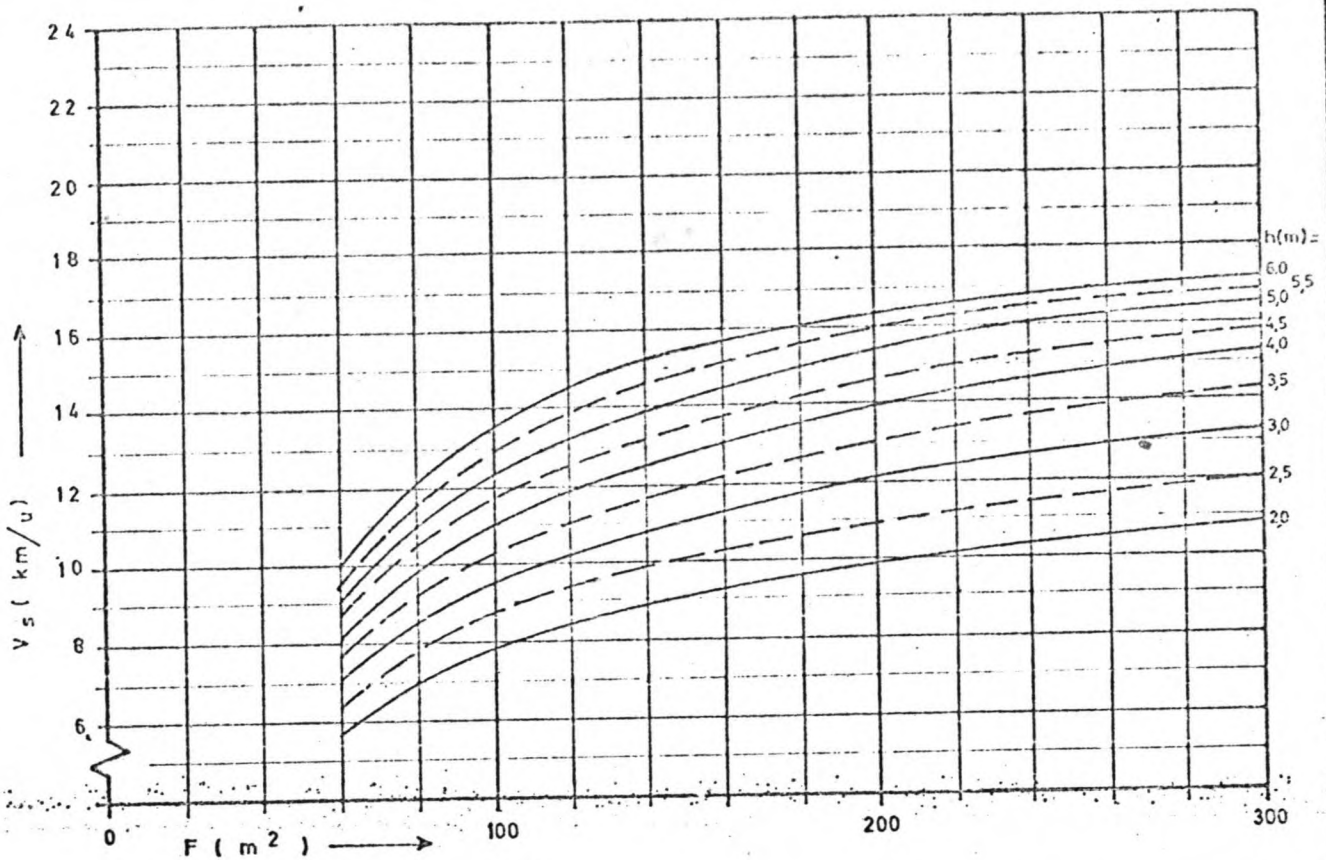
STANDAARDSCHIP 7 ( $\lambda = 1,00$ )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG

Fig. 28

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RIJKSWATERSTAAT

get. 21-10-71  
 J.M.

No. 70.12.4



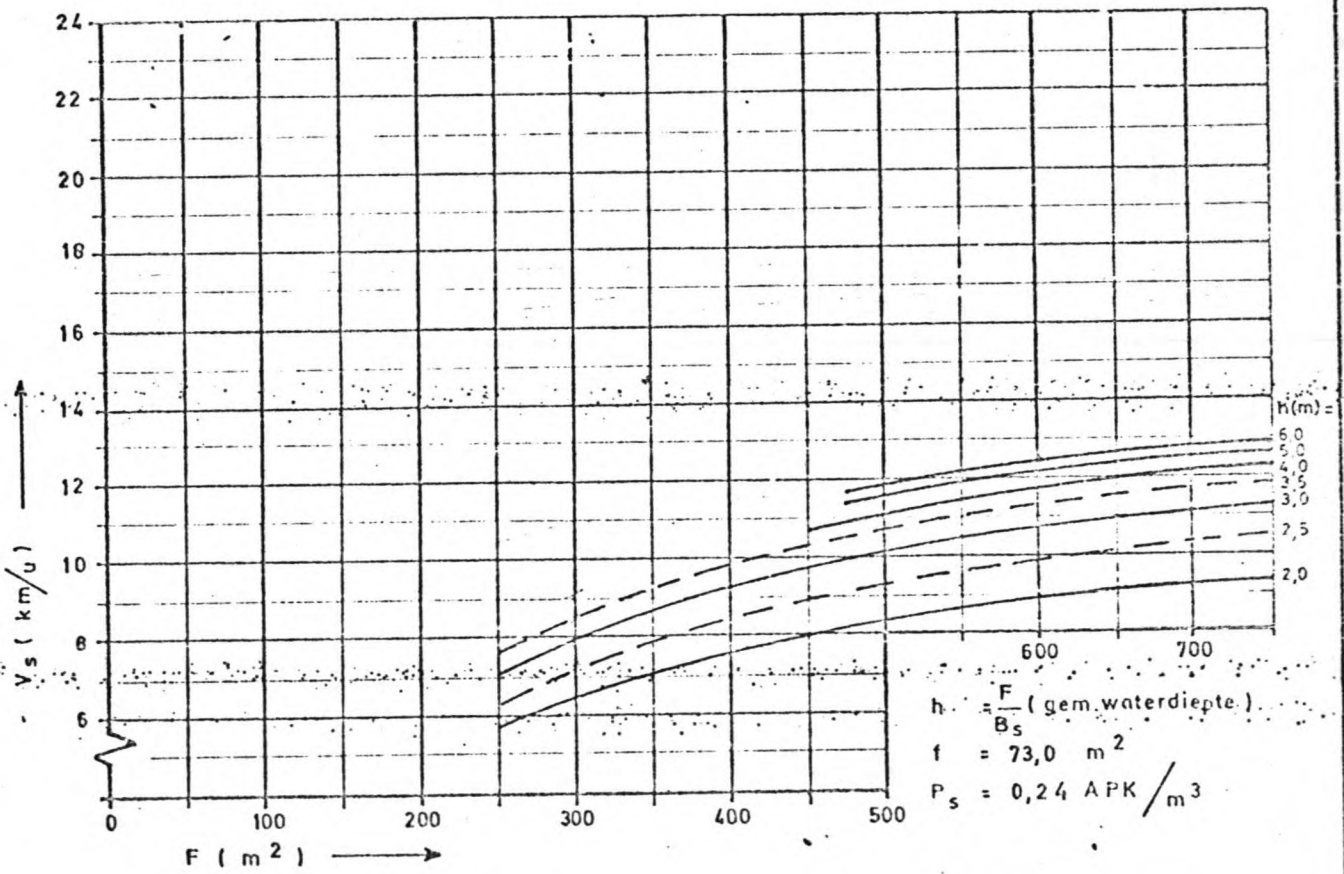
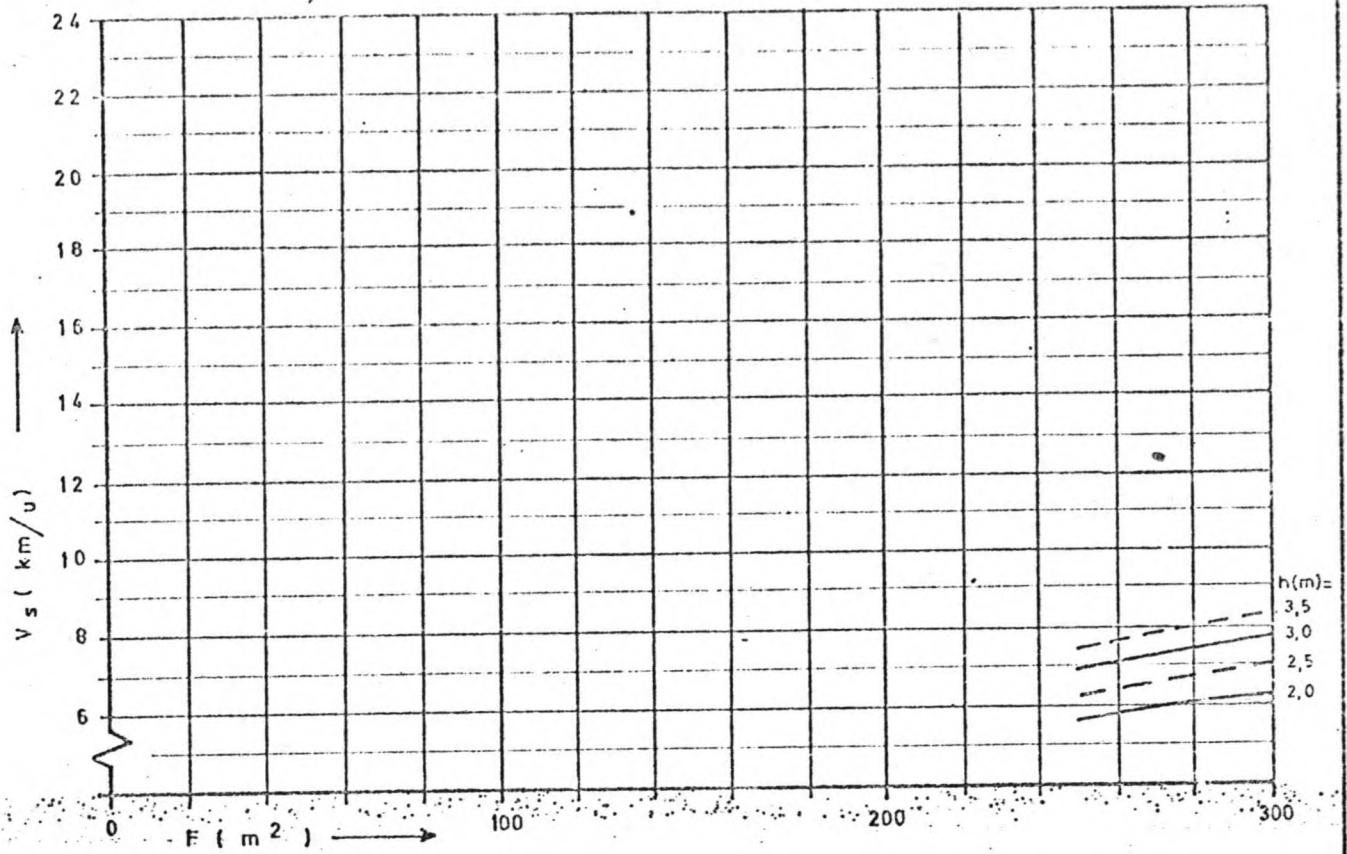
STANDAARDSCHIP 7 ( $\lambda = 0,00$ )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG

Fig. 30

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RUKSWATERSTAAT

get.	gez.
21-10-71	
J.M.	

No. 70.12.4



$\lambda =$  beladingsgraad

STANDAARDSCHIP 8 ( $\lambda = 1,00$ )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG

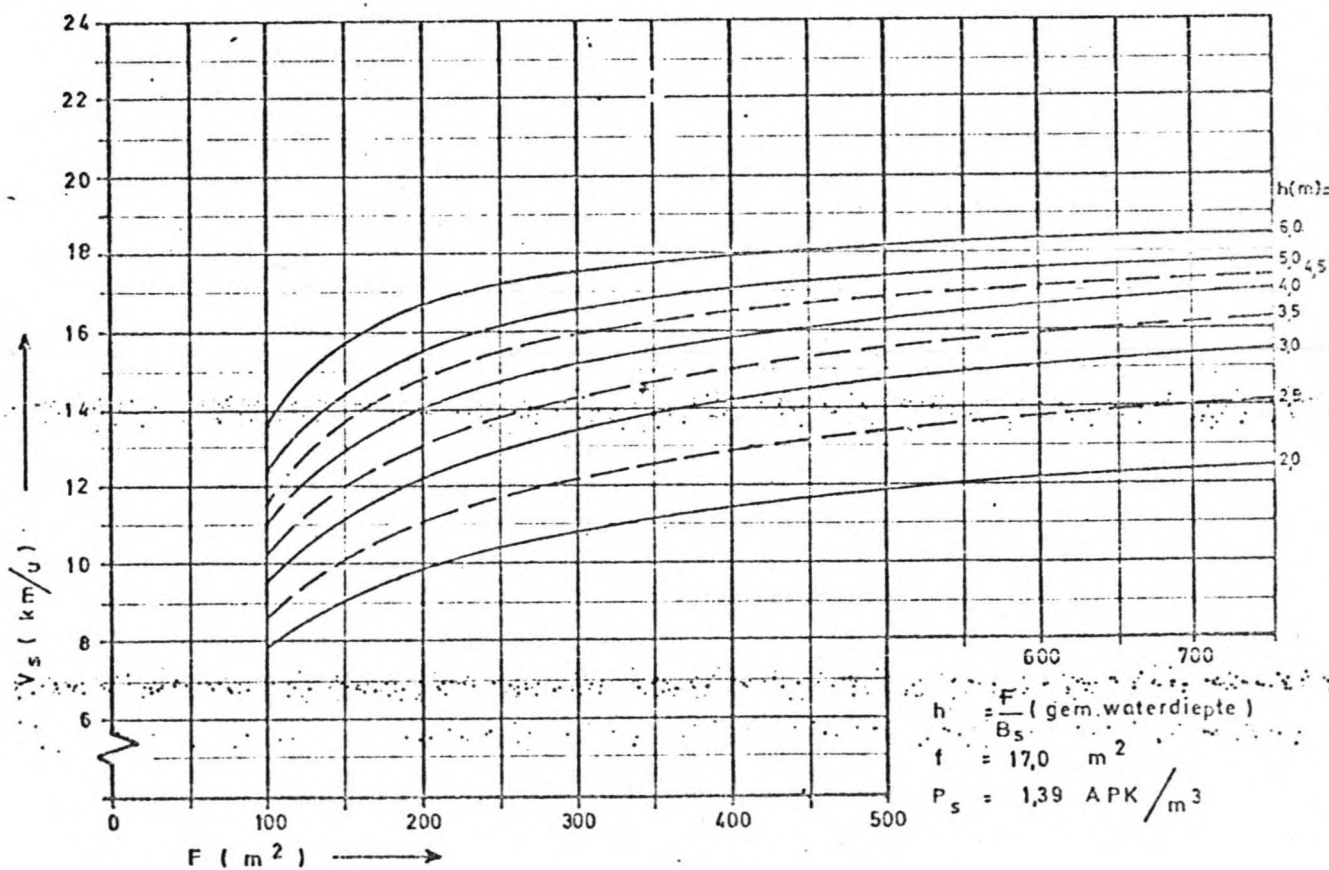
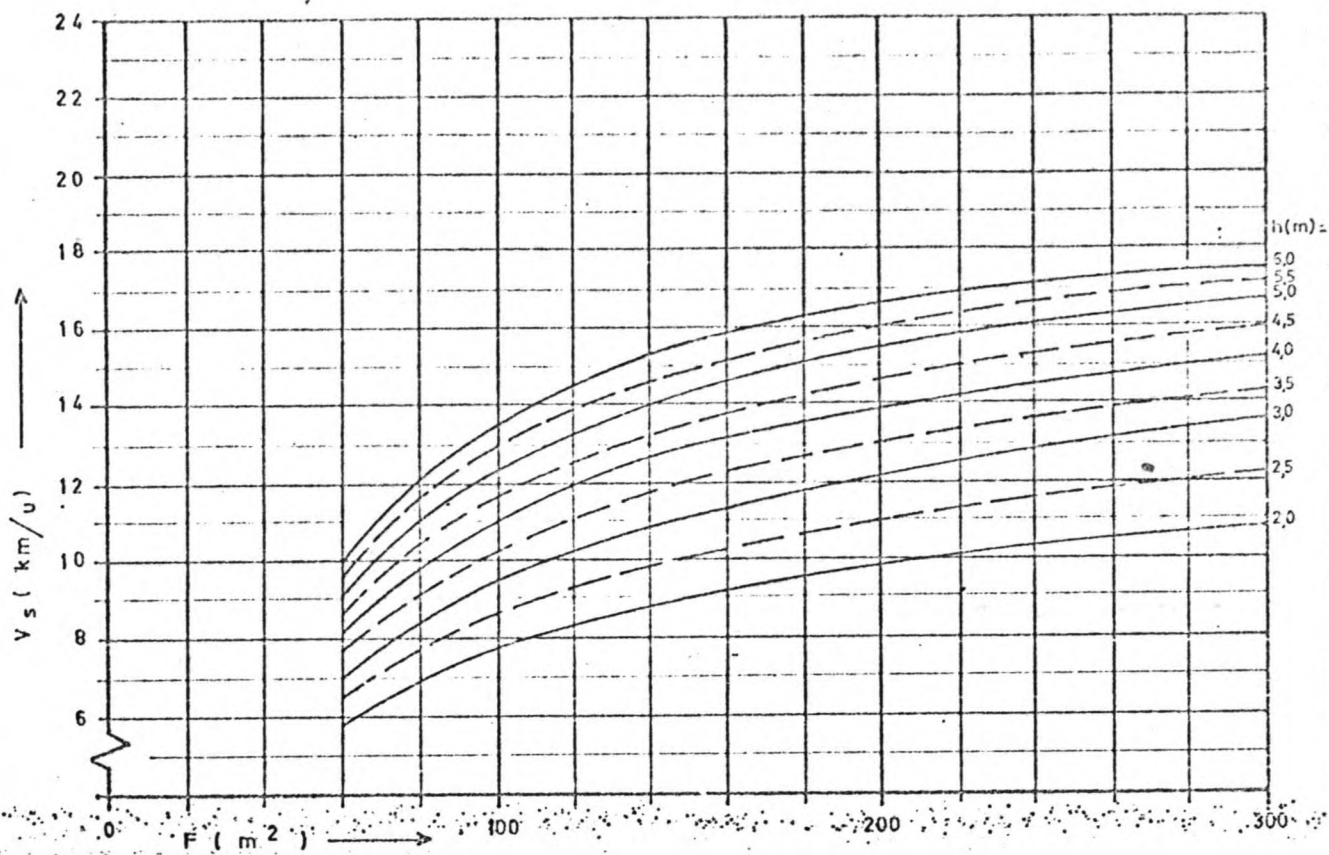
Fig. 31

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RIJKSWATERSTAAT

get. 21-10-71  
 J.M.

No. 70.12.4





STANDAARDSCHIP 8 ( $\lambda = 0,00$ )  
 VERBAND TUSSEN VAARSNELHEID EN DWARSPROFIEL  
 VAN DE VAARWEG

Fig. 33

STUDIEDIENST VERKEER TE WATER  
 DIRECTIE BENEDENRIVIEREN  
 RIJKSWATERSTAAT

get.	gez.
21-10-21	
J.M.	

No. 70.12.4

BIJLAGE G.

Meetresultaten kolkindeling,  
8 juli 1975 bij de Volkeraksluizen.



Schuifling no: 1

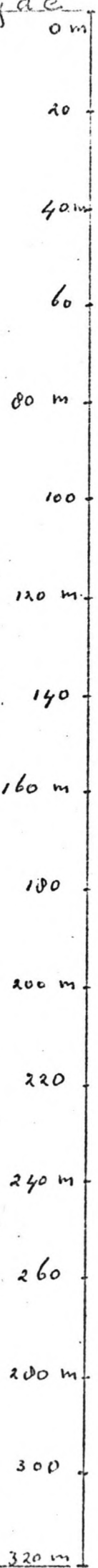
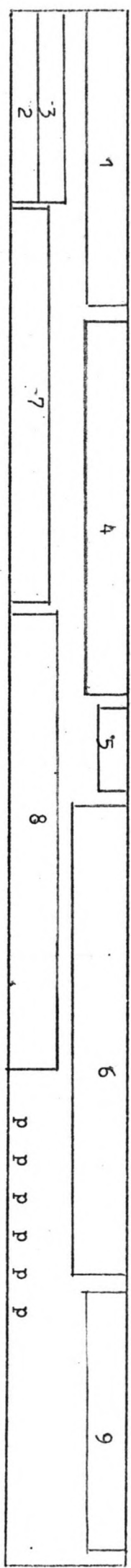
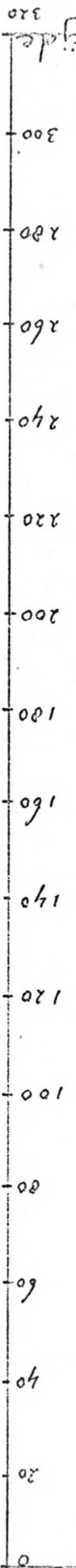
Onderzoek Kolkindeling Volkeraksluizen

	Naam	schip + plaats	gela- den	tonnage	tijd van aan Komst	tijd van vertrek
18	WESTKOLK	Terneuzen	-	619	10.44.45	11.29.45/3
19	LISA		+	350	10.45.30	11.28.20/1
20	SPEFICA		+	332	10.46.00	11.29.00/2
21	CALENDULA VII	Rotterdam	-	1300	10.50.50	11.30.05/4
22	SPES (sleeboot)	Rotterdam	n.v.	33	10.58.20	11.35.00/7
23	KALAHARI (geslept)	Rotterdam	+	2827	11.01.10	11.36.20/8
24	ZAGRI 20	Rotterdam	+	970	11.05.50	11.31.50/5
25	RICY III	Latem	+	2000	11.09.40	11.33.05/6
26	MANDELZOOM	Gent	-	800	11.10.20	11.37.40/9
27	PLEZIERVAART - 6	+5 min	n.v.	n.v.	11.17.30	11.40.45/10
28						
29						
30						
31						
32						
33						

Schutting no : 1

sluis dicht	11.17.45
sluis open	11.27.55
bedienings tijd	10.10

Hollandsch diep zijde



Vaarrichting : ←

Hollandsch diep zijde

Onderzoek Kolkindeling Volkeraksluizen

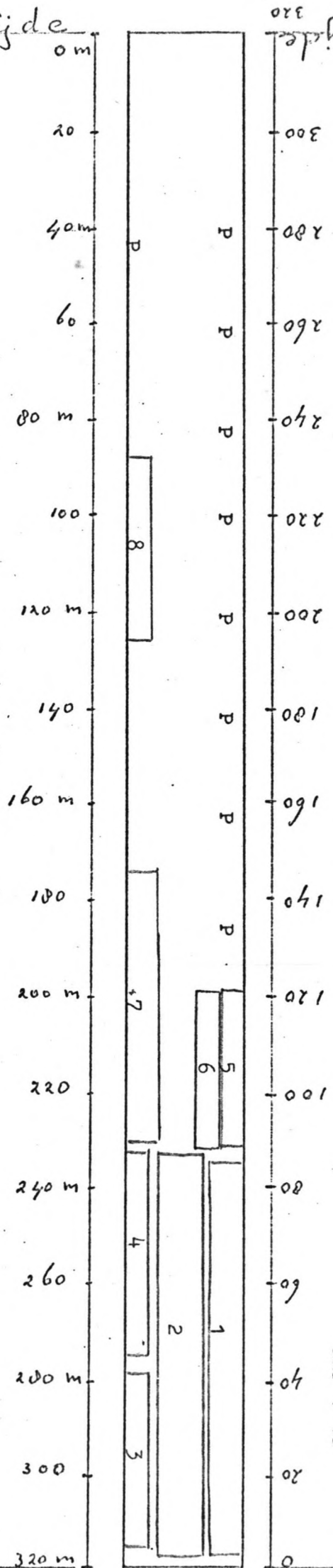
Volkerak zijde

	Naam schip + plaats	gela- den	tonnage	tijd van aan komst	tijd van verbrek
18	1 ARO 4	+	1448	11.45.20	12.15.00/3
19	2 DUDENBUETTEL	-	1439	11.49.40	12.13.40/1
20	3 NOORD HINDER	+		11.50.15	12.14.00/2
21	4 LEJOMA	-	260	11.51.40	12.15.30/4
22	5 SIRENELLA	+	380	11.54.00	12.17.00/6
23	6 CONSTANCE II (pleziervisboot)	n.v.	16	11.55.30	12.16.10/5
24	7 BERTHA	+	600	11.56.50	12.18.00/7
25	8 PIETTER B	+	300	11.59.10	12.20.00/8
26	9 PLEZIERVAART - 9 ± 2 min.	n.v.		12.02.40	12.21.20/9
27	10				
28	11				
29	12				
30	13				
31	14				
32	15				
33	16				
34	17				

Schutting no. 2

datum: 8-7-1975

Sluis dicht	12.03.30
sluis open	12.11.00
bedieningsijd	7.30



NAP + 10

Vaarrichting: →

Onderzoek Kolkindeling Volkeraksluizen

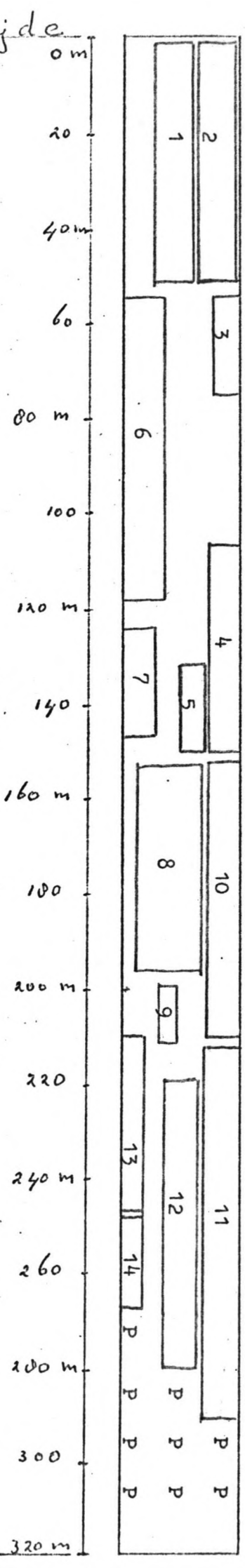
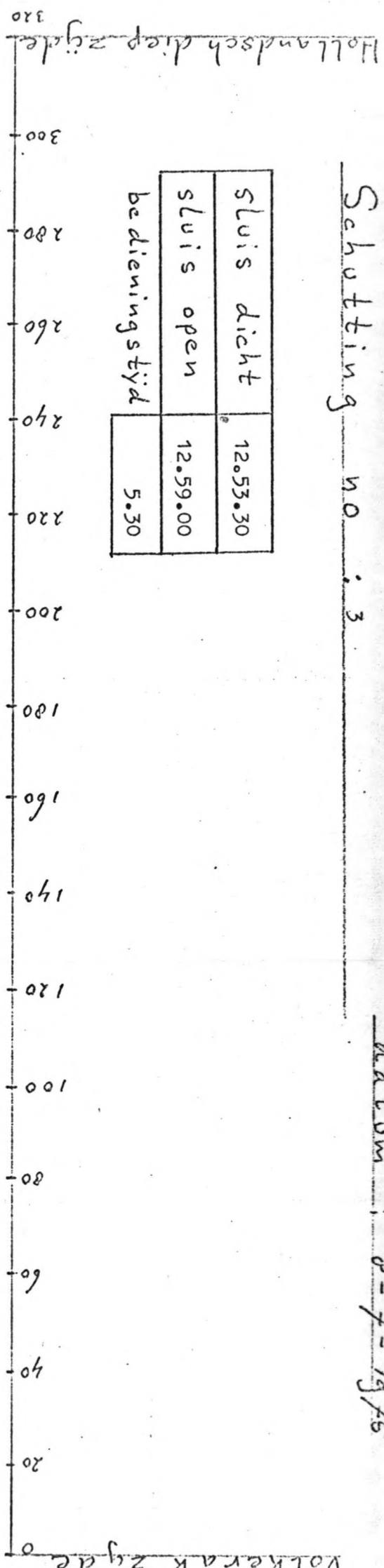
Volkerak zijde

	Naam schip + plaats	W. stad	geladen	tonnage	tijd van aankomst	tijd van vertrek
18	JAN B	W. stad	-	400	12.28.00	13.01.00/1
19	SJAAK B	W. stad	-	450	12.28.00	13.01.00/2
20	DUW SLEEPBOOT	W. stad	n.v.	50	12.28.10	13.01.10/3
21	BB 130	Overveen	-	330	12.29.40	13.02.50/5
22	ENAVANT (sleeboot)	Dordrecht		50	12.29.40	13.02.50/6
23	ZWERVER	Werkendam	-	251	12.30.45	13.02.00/4
24	BROEDERTROUW VII (sleeboot)				12.35.40	13.03.20/7
25	GROTE KRAAN				12.36.25	13.03.45/8
26	KLEIN DUW SLEEPBOOTJE				12.36.26	13.03.50/9
27	ELJO D 73 m	Zwijndrecht	+	916	12.39.35	13.06.00/10
28	TRANSITORIUS 65 m	Werkendam	-		12.41.55	13.08.10/12
29	FRATERNUS 55 m	Zeebrugge	-	930	12.41.40	13.08.10/13
30	JAN Sr. 52 m	Rotterdam	-		12.44.55	13.07.00/11
31	EMMANUEL	Rotterdam	+		12.46.45	13.11.40/14
32	PLEZIERVAART				12.47.10	13.12.20/15
33						
34						



Schutting no : 3

sluis dicht	12.53.30
sluis open	12.59.00
bedieningstijd	5.30



Vanrichting : ←

Hollandsch diep zijde

Onderzoek Kolkindeling Volkeraksluizen

Volkerak zijde

	Naam	schip + plaats	gela- den	tonnage	tijd van aankomst	tijd van vertrek
18	1	JOHANNA 80 m Merkssem (B)	+	1014	13.17.45	13.38.30/1
19	2	MARIA 50 m Merkssem	-		13.18.55	13.41.35/2
20	3	MARIE JOSEPIA 50 m Rupermonde	+		13.20.10	13.40.35/3
21	4	STEVINGROEP (sleeboot) 10 m Rupermonde	n.v.		13.20.20	13.40.20/4
22	5	ERAN 64 m	+		13.22.30	13.42.40/5
23	6	BRANTAS 60 m Rotterdam	-		13.23.30	13.43.20/6
24	7	NAVI 40 m	-		13.25.20	13.44.20/7
25	8	NIEUWE ZORG 32 m Rotterdam	+	209	13.26.20	13.45.15/8
26	9	PLEZIERVAART 6 + 5 min			13.30.10	13.47.25/9
27	10					
28	11					
29	12					
30	13					
31	14					
32	15					
33	16					
34	17					

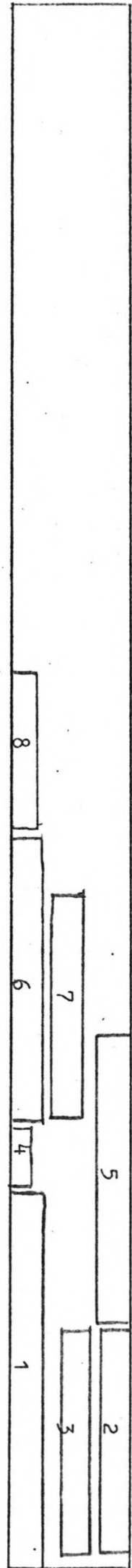
Hollandsch diep zijde

0 m  
20  
40 m  
60  
80 m  
100  
120 m  
140  
160 m  
180  
200 m  
220  
240 m  
260  
280 m  
300  
320 m

Vaarrichting :  $\longrightarrow$

Onderzoek Kolkindeling Volkeraksluizen

Volkerak zijde



Hollandsch diep zijde

320  
300  
280  
260  
240  
220  
200  
180  
160  
140  
120  
100  
80  
60  
40  
20  
0

Sluis dicht	13.30.40
Sluis open	13.37.35
bedieningstijd	6.55

Schutting no : 4

datum : 8-7-1975

Schutting no. 5

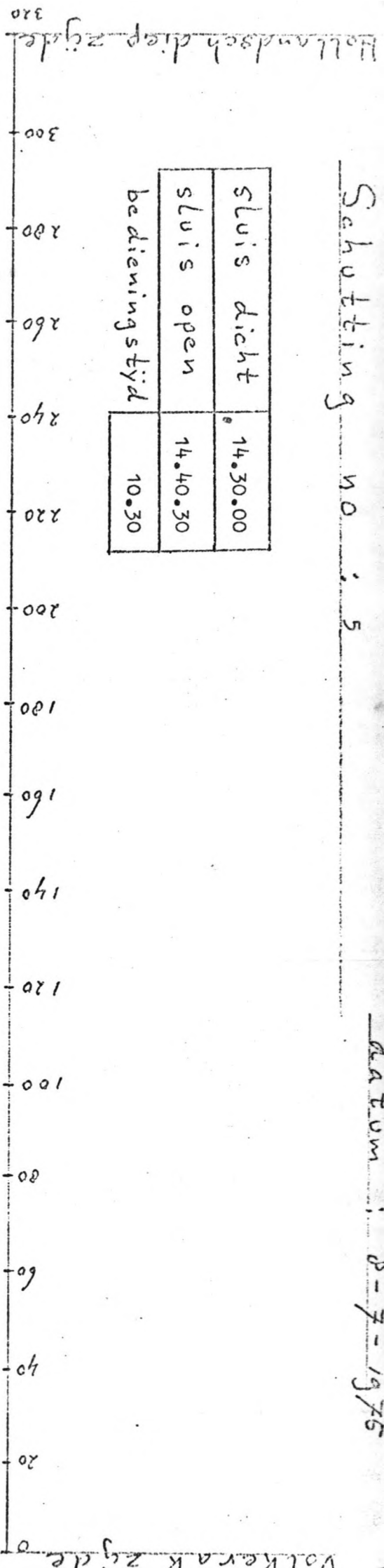
Onderzoek Kolkindeling Volkeraksluizen

	Naam	schip + plaats	gela-	tonnage	tijd van	tijd van	
			den		aan Komst	verbrek	
18	MARION	80 m	Ruhrort (D)	+	1220	13.51.40	14.44.30/2
19	CORONA	80 m	Antwerpen (B)	+	1326	13.54.20	14.46.30/6
20	ZAGRI 21	64 m	Rotterdam	+	654	13.57.20	14.43.05/1
21	SLEEPBOOTE	26 m	Zwijndrecht	n.v.	6	13.58.50	14.44.40/3
22	BOORPONTON	grondmij	Haarlem	n.v.	45	13.59.20	14.45.05/5
23	HENJA	32 m	Werkendam	-	280	14.01.30	14.45.00/4
24	VERENA WEKING	80 m	Bevergern (D)	+	1320	14.04.20	14.48.00/8
25	AMBULANT	44 m	Oude Tonge	+	352	14.06.02	14.49.15/10
26	ANDRE	80 m	Dordrecht	+	1299	14.08.15	14.51.00/11
27	RIVAL	48 m	Sas van Gent	+	350	14.09.50	14.47.05/7
28	JOHANNA	31 m	Goes	-	130	14.13.10	14.48.40/9
29	ELFRA (sleeboot)		Maasbracht	n.v.	23	14.15.30	14.59.50/13
30	BIJZONDER BUIS TRANSPORT		BREEDTE + 8 m LENGTE + 80 m	n.v.		14.21.00	15.01.10/14
31	WALVIS (sleeboot)		Dordrecht	n.v.	27	14.21.00	15.01.10/15
32	PLEZIERVAART	14 + 11 min		n.v.		14.29.20	14.56.20/12
33							
34							

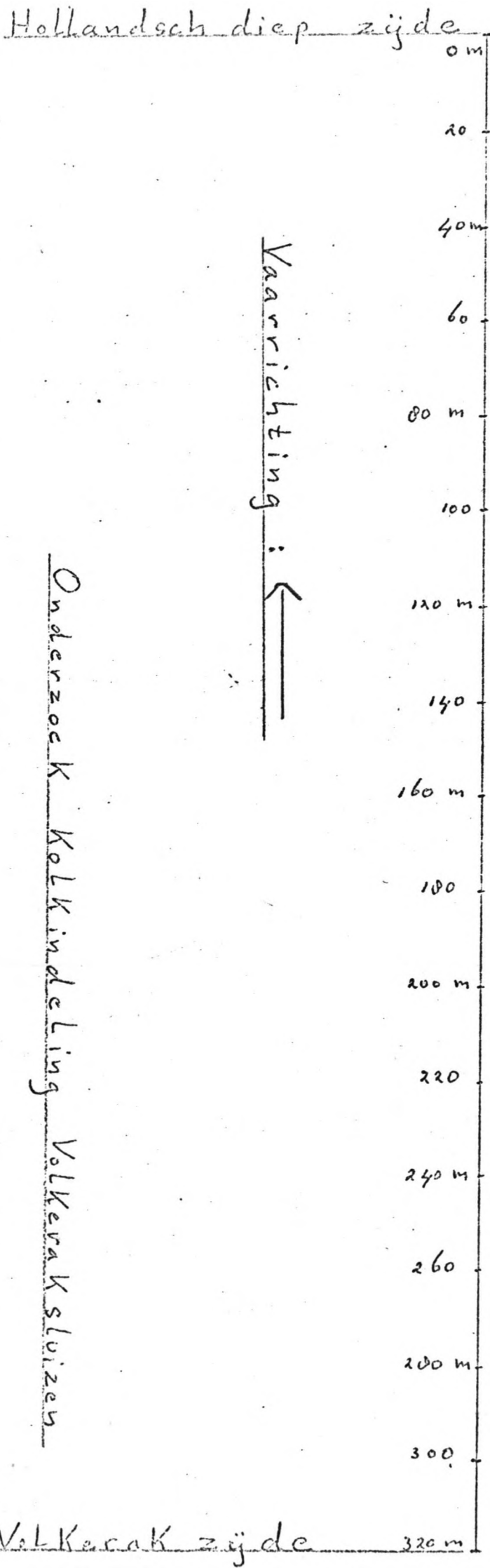
Schutting no. 5

datum: 8-7-1975

sluis dicht	14.30.00
sluis open	14.40.30
bedieningstijd	10.30



Voarriehhting: ←



Onderzoek Kolkindeling Volkeraksluizen

Volkerak zijde



Schulding no. : 6

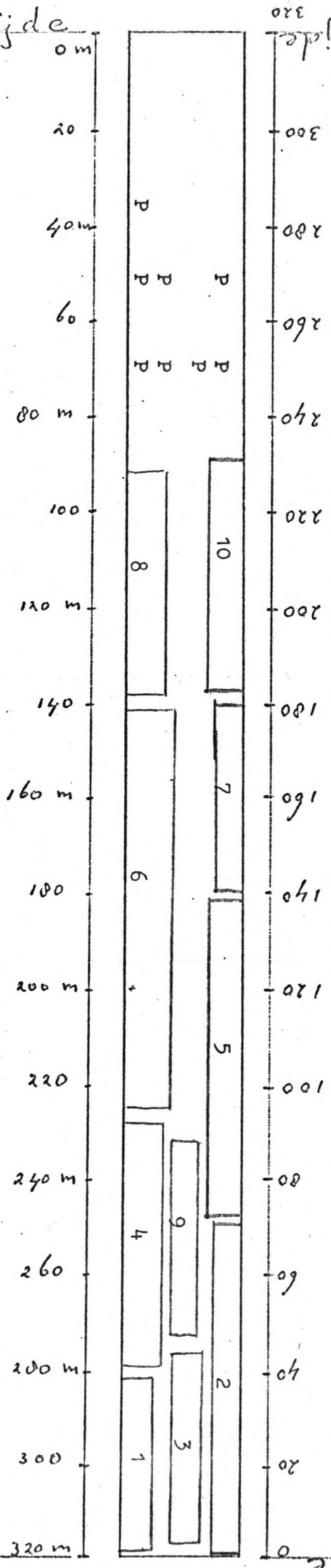
Onderzoek Kolkindeling VolKeraKsluizen

	Naam	schip +	plaats	gela-	tonnage	tijd van	tijd van	
				den		aan Konst	ver trek	
18	1	ANTRUSO	40 m	Geertruidenberg	-		15.05.30	15.34.50/2
19	2	HEIN	73 m	Zwijndrecht	+	972	15.06.50	15.35.10/3
20	3	DUBIO	48 m	Terneuzen	+		15.07.50	15.34.30/1
21	4	CHRISTING	55 m	Rotterdam	+	325	15.09.30	15.37.35/4
22	5	ASTRID	67 m	Schiedam	-	910	15.10.50	15.38.00/5
23	6	TALING	90 m	Rotterdam	+		15.13.30	15.39.20/6
24	7	ZEEUWSE STROMEN	55 m	Oude Tonge	+	388	15.14.40	15.42.40/9
25	8	EUGENE	50 m	Gent	+	858	15.15.10	15.40.35/7
26	9	CLAUDE	48 m	Colin (F)	-		15.15.33	15.41.20/8
27	10	WATERPOORT	48 m	Stiens	-	410	15.18.40	15.43.20/10
28	11	PLEZIERVAART 8	+ 1.30 min		n.v.	n.v.	15.22.30	15.46.30/11
29	12							
30	13							
31	14							
32	15							
33	16							

Schutting no : 6

datum : 8-7-1975

sluis dicht	15.23.00
sluis open	15.34.20
bedieningstijd	11.20



Vaarrichting : →

Onderzoek Kolkindeling Volkeraksluizen

Volkerak zijde

