

K O N I N K L I J K N E D E R L A N D S
M E T E O R O L O G I S C H I N S T I T U U T

D e B i l t

Verslagen

V - 295

J.D. Opsteegh

Een barotroop model voor
96-uurs circulatieverwachtingen

De Bilt, 1977

Publikationsnummer: K.N.M.I. V-295 (M.O.)

U.D.C.: 551.509.313

EEN BAROTROOP MODEL VOOR
96-UURS CIRCULATIEVERWACHTINGEN

J.D. Opsteegh

I. Inleiding

Sinds 1968 wordt de Amerikaanse 72-uursverwachting voor 500 mbar (FUNT-FUNA) door het KNMI gebruikt om met behulp van een barotroop model deze prognose te verlengen tot 96 uur. Deze 96-uursvoorspelling wordt gebruikt bij de opstelling van de operationele verwachtingen voor de middellange termijn (3-8 dagen).

Het hiervoor gebruikte computerprogramma werd ontwikkeld op het KNMI (Van Galen 1969). Dit programma rekent echter op een rekenrooster, dat niet samenvalt met het Amerikaanse FUNT-FUNA rooster, zodat niet onmiddellijk na binnenkomst van de berichten kan worden gestart met de berekeningen.

De werkwijze tot nu toe was als volgt:

- Het verwachte stromingspatroon voor 72 uur vooruit wordt overgetrokken op een kaart waarop het rooster van het barotrope model staat afgebeeld. In het gebied van het rooster, dat buiten het FUNT-FUNA rooster ligt, tekent de meteoroloog een door hem verwacht stromingspatroon.
- Vervolgens wordt de 500 mbar hoogte van alle roosterpunten afgelezen en op een daarvoor bestemde lijst ingevuld.
- Deze lijst wordt op ponsband of ponskaarten gezet en in de computer ingevoerd. De begincondities voor een verdere barotrope voorspelling van 72 uur naar 96 uur liggen nu voor het hele rooster vast.

Deze procedure werd zes keer per week uitgevoerd, waarbij iedere keer ongeveer 400 roosterpunten moesten worden afgelezen en op ponsband gezet. Naast het arbeidsintensieve karakter van bovenomschreven werkwijze is de kans op schrijf- of ponsfouten groot.

Het grootste bezwaar echter is, dat het zeer lang duurt voordat de meteoroloog de beschikking heeft over een 96-uursverwachting.

Om deze redenen werd het door de operationele dienst wenselijk geacht de procedure te automatiseren. Deze automatisering stuitte op moeilijkheden wanneer het programma van Van Galen werd gebruikt. Besloten werd daarom het operationele programma van het barokliene 3-lagen model (BK27) als uitgangspunt te gebruiken voor de ontwikkeling van een nieuw barotroop programma (Van Galen 1969, Opsteegh 1975, Heijboer 1977).

Het rekenrooster van dit barotrope model is een snede uit het FUNT-FUNA rooster, zodat het stromingspatroon niet meer buiten het FUNT-FUNA gebied hoeft te worden geëxtrapoleerd.

Een probleem dat hierbij opduikt is het feit dat de zuidelijke rand van dit rooster over Zuid-Spanje loopt ($\pm 40^{\circ}$ NB) en dus tamelijk dicht bij Nederland in een meteorologisch actief gebied is gelegen. Aangezien er geen geschikte randcondities voor deze rand geformuleerd kunnen worden, geeft dit aanleiding tot de introductie van fouten langs de rand, die in de loop van de tijd het rekengebied binnenlopen en daar het stromingspatroon verstoren. Het betreft hier echter een prognose over een tamelijk korte termijn (24 uur), zodat gehoopt kan worden dat het effect van randbederf beperkt blijft. Gebruikers van de 96-uursprognose dienen echter op dit effect attent te zijn.

Aangezien er binnen de afdeling Meteorologisch Onderzoek mogelijk belangstelling bestaat om experimenten op te zetten met behulp van een barotroop model, werd tevens een programma ontwikkeld waarbij hetzelfde barotrope model op een hemisferisch rooster wordt gedraaid. Hiervoor werd hetzelfde rooster genomen als dat waarop het 3-lagen model rekt.

In hoofdstuk II wordt een beschrijving gegeven van het barotrope model. In hoofdstuk III worden de gebruikte differentieschema's behandeld, terwijl in hoofdstuk IV een beschrijving van het computerprogramma wordt gegeven. Tot slot wordt een complete listing gegeven van het programma zoals dat op 13-7-77 operationeel werd ingevoerd.

II. Modelvergelijkingen

Uitgegaan wordt van de barotrope vorticieteitsvergelijking waarin een diffusie-term is opgenomen.

$$\frac{\partial \zeta}{\partial t} = - \vec{V} \cdot \nabla (\zeta + f) + K \nabla^2 \zeta \quad (1)$$

Hierin is:

- ζ relatieve vorticieteit
- f Coriolisparameter
- \vec{V} horizontale windvector
- K diffusieconstante.

Wanneer in het windveld het divergente gedeelte wordt verwaarloosd, kan een stroomfunctie worden geïntroduceerd. \vec{V} en ζ kunnen in deze stroomfunctie ψ worden uitgedrukt

$$\vec{V} = \vec{K} \times \nabla \psi \quad (2)$$

$$\zeta = \nabla^2 \psi \quad (3)$$

Substitutie van (2) en (3) in (1) en transformatie naar de stereografische projectie levert:

$$\nabla^2 \left(\frac{\partial \psi}{\partial t} \right) = - J(\psi, m^2 \nabla^2 \psi + f) + K \nabla^2 (m^2 \nabla^2 \psi) \quad (4)$$

Met:

- m kaartschaalfactor
- J de jacobiaan

Stel:

$$\left(\frac{\partial \psi}{\partial t} \right)^* = \frac{\partial \psi}{\partial t} - K m^2 \nabla^2 \psi \quad (5)$$

Substitutie van (5) in (4) geeft:

$$\nabla^2 \left(\frac{\partial \psi}{\partial t} \right)^* = - J(\psi, m^2 \nabla^2 \psi + f) \quad (6)$$

In voorspelvergelijking (6) worden planetaire golven sterk retro-
graad. Om deze golven ongeveer stationair te houden, is aan de
vergelijking een Cressman-correctie toegevoegd. De uiteindelijke
voorspelvergelijking wordt dan:

$$\nabla^2 \left(\frac{\partial \psi}{\partial t} \right)^* - \left(\frac{f}{mf_0} \right)^2 q \left(\frac{\partial \psi}{\partial t} \right)^* = - J(\psi, m^2 \nabla^2 \psi + f) \quad (7)$$

met:

- q Cressman-correctie
- f₀ Coriolis-parameter op 45° N.

De randvoorwaarden, die bij de oplossing van (7) worden gebruikt,
zijn:

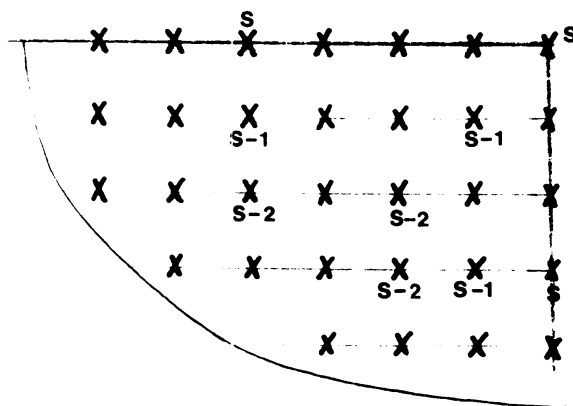
$$\left(\frac{\partial \psi}{\partial t} \right)^*_S = 0 \quad (7a)$$

$$\frac{\partial}{\partial n} \left(\frac{\partial \psi}{\partial t} \right)^*_S = 0 \quad (7b)$$

Aan het einde van iedere tijdstap wordt aan de rand op simpele wijze
gladgestreken door het stroomfunctieveld, dat 1 roosterpuntafstand
van de rand aflight, te middelen (zie fig. 1).

$$\psi_{S-1} = \frac{\psi_S + \psi_{S-2}}{2} \quad (8)$$

Figuur 1



Na iedere tijdstap wordt langs de rand S op simpele wijze gladge-
streken. Dit gebeurt door voor ψ in de punten S-1 het gemiddelde
te nemen van ψ_S en ψ_{S-2} .

De stroomfunctie ψ wordt afgeleid van de geopotentiële hoogte van 500 mbar door toepassing van de lineaire balansvergelijking.

$$f \nabla^2 \psi + \nabla f \cdot \nabla \psi = \nabla^2 \Phi_{500} \quad (9)$$

met:

$$\Phi_{500} = g z_{500}$$

Een vergelijking die praktisch gelijkwaardig is aan (9) wordt verkregen door in de tweede term van het linkerlid $\nabla \psi$ te vervangen door $\frac{\nabla \Phi_{500}}{f}$.

De vergelijking reduceert hiermee tot een simpele Poisson-vergelijking:

$$\nabla^2 \psi = \frac{\nabla^2 \Phi_{500}}{f} - \frac{\nabla f \cdot \nabla \Phi_{500}}{f^2} \quad (10)$$

Wanneer in alle roosterpunten de geopotentiële hoogte van 500 mbar bekend is, kan (10) worden opgelost. Het windveld dat op deze wijze wordt verkregen is de divergentievrije component van de geostrofische wind. Als randvoorwaarde wordt de quasi-geostrofische relatie tussen ψ en Φ_{500} gebruikt.

$$\psi = \frac{\Phi_{500}}{f_0} \quad (11)$$

Vervolgens kan voorspelvergelijking (7) worden opgelost. Oplossing van deze Helmholtz-vergelijking levert het tendensveld $\left(\frac{\partial \psi}{\partial t}\right)^*$.

Hierna kan door integratie van (5) in de tijd de verandering van de circulatie worden berekend.

Deze veranderingen van de circulatie worden beschreven in termen van een veranderend ψ -veld. Uit dit ψ -veld kan een hoogteveld worden afgeleid door het oplossen van een lineaire balansvergelijking, maar nu in omgekeerde richting (inverteren).

$$\nabla^2 \Phi_{500} = f \nabla^2 \psi + \nabla f \cdot \nabla \psi \quad (12)$$

Als randvoorwaarde wordt wederom de quasi-geostrofische relatie gebruikt.

$$\Phi_{500} = f_0 \psi \quad (13)$$

Voor de oplossing van Poisson- en Helmholtz-vergelijkingen wordt de iteratieve Liebmann-relaxatiemethode gebruikt (Haltiner 1971). Bij deze methode is het noodzakelijk een startveld te hebben. Bij het oplossen van de lineaire balansvergelijking wordt als startveld de quasi-geostrofische relatie $\psi = \Phi/f_0$ gebruikt. Het inverteren van deze vergelijking gebeurt met als startveld $\Phi = f_0\psi$. Oplossing van voorspelvergelijking (7) vereist een startveld van de tendens $\left(\frac{\partial \psi}{\partial t}\right)^*$.

Als eerste tijdstap wordt voor dit startveld in alle roosterpunten de waarde 0 genomen, vervolgens wordt het tendensveld $\left(\frac{\partial \psi}{\partial t}\right)^*$ van de vorige tijdstap als startveld genomen.

De randvoorwaarden (11) en (13) worden bij de lineaire balansvergelijking op S (zie fig. 1) opgelegd, terwijl bij de oplossing van vergelijking (7) de randvoorwaarden (7a) en (7b) worden vertaald in

$$\left(\frac{\partial \psi}{\partial t}\right)^*_{S-1} = \left(\frac{\partial \psi}{\partial t}\right)^*_S = 0 \quad (7c)$$

De Helmholtz-vergelijking wordt opgelost in het gebied dat binnen S-1 ligt.

III. Differentieschema's

De differentieschema's, die gebruikt worden om het stelsel differentiaalvergelijkingen om te zetten in een stelsel differentievergelijkingen, zijn dezelfde als de schema's, die in het 3-lagen model worden gehanteerd. Voor een willekeurig punt i, j zien deze differentieoperatoren er als volgt uit (zie fig. 2):

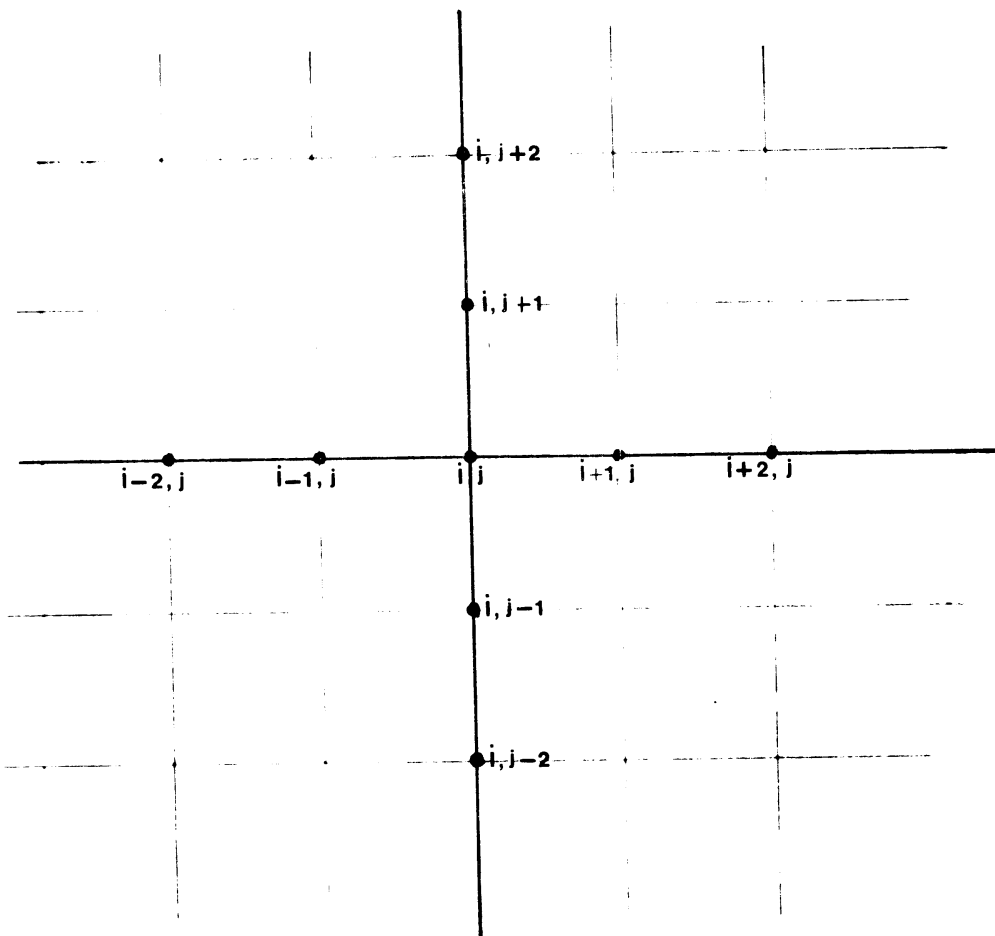
- Laplace-operator:

Hiervoor wordt een 5-punts operator met tweede-orde nauwkeurigheid gebruikt.

$$\nabla^2 \psi = (\psi_{i+1,j} + \psi_{i-1,j} + \psi_{i,j+1} + \psi_{i,j-1} - 4\psi_{i,j}) / d^2 + O(d^2) \quad (14)$$

- Jacobiaan.

Vanwege de geringe nauwkeurigheid van tweede-orde advectionsschema's wordt een vierde-orde schema gehanteerd:



Figuur 2

Gedeelte van het gebruikte rekenrooster. Voor de berekening van veranderingen in punt i, j worden grootheden van alle gemarkeerde roosterpunten gebruikt. De roosterpuntafstand $d = 381 \text{ km op } 60^\circ \text{ N.}$

$$J(\psi, \zeta)_{i,j} =$$

$$\begin{aligned} & \frac{4}{9 d^2} \left[(\psi_{i+1,j} - \psi_{i-1,j}) (\zeta_{i,j+1} - \zeta_{i,j-1}) - (\psi_{i,j+1} - \psi_{i,j-1}) (\zeta_{i+1,j} - \zeta_{i-1,j}) \right] \\ & + \frac{1}{18 d^2} \left[(\psi_{i,j+1} - \psi_{i,j-1}) (\zeta_{i+2,j} - \zeta_{i-2,j}) - (\psi_{i+2,j} - \psi_{i-2,j}) (\zeta_{i,j+1} - \zeta_{i,j-1}) \right] \\ & - \frac{1}{18 d^2} \left[(\psi_{i+1,j} - \psi_{i-1,j}) (\zeta_{i,j+2} - \zeta_{i,j-2}) - (\psi_{i,j+2} - \psi_{i,j-2}) (\zeta_{i+1,j} - \zeta_{i-1,j}) \right] \\ & + \frac{1}{144 d^2} \left[(\psi_{i+2,j} - \psi_{i-2,j}) (\zeta_{i,j+2} - \zeta_{i,j-2}) - (\psi_{i,j+2} - \psi_{i,j-2}) (\zeta_{i+2,j} - \zeta_{i-2,j}) \right] \\ & + O(d^4) \end{aligned} \quad (15)$$

- De tijdsafgeleide.

De tendens van het stroomfunctieveld bestaat uit twee componenten,

- nl. 1. tendens veroorzaakt door advectionprocessen;
2. tendens door diffusieprocessen.

Om redenen van numerieke stabiliteit kunnen diffusieprocessen wel met een voorwaartse tijdstap en niet met een centrale tijdstap worden beschreven, terwijl bij advectionprocessen een centrale tijdstap en geen voorwaartse tijdstap kan worden gemaakt.

Het differentie-analagon van (5) ziet er daarom als volgt uit:

$$\psi_{t+\Delta t} = \psi_{t-\Delta t} + 2 \Delta t \left(\frac{\partial \psi}{\partial t} \right)_t^* + 2 \Delta t K m^2 (\nabla^2 \psi)_{t-\Delta t} \quad (16)$$

De grootte van de voorwaartse tijdstap in het diffusiegedeelte is $2 \Delta t$, terwijl de grootte van de centrale tijdstap in het advectioneel gedeelte Δt bedraagt.

Om de berekeningen te kunnen starten is een aanloopprocedure nodig.

Dit gaat als volgt:

De eerste tijdstap is een voorwaartse tijdstap ter grootte $\frac{1}{2} \Delta t$.

Het gebruikte schema is in dit geval:

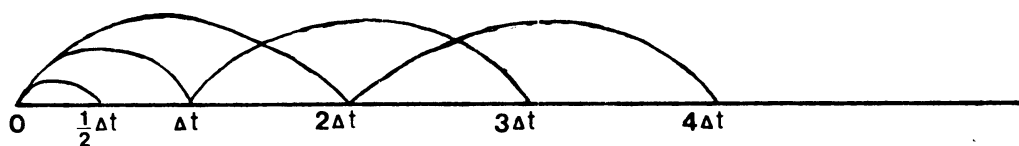
$$\psi \Delta t = \psi_0 + \Delta t \left(\frac{\partial \psi}{\partial t} \right)_0^* + \Delta t K m^2 (\nabla^2 \psi)_0 \quad (16a)$$

Voor de tweede tijdstap wordt formule (16) gebruikt, maar met tijdstapgrootte $\frac{1}{2} \Delta t$:

$$\psi_{\Delta t} = \psi_0 + \Delta t \left(\frac{\partial \psi}{\partial t} \right)_{\frac{1}{2} \Delta t}^* + \Delta t K m^2 (\nabla^2 \psi)_0 \quad (16b)$$

De derde tijdstap is de eerste tijdstap, die met formule (16) wordt berekend.

Figuur 3 geeft een schematisch beeld van bovenbeschreven start-procedure.



Figuur 3

Schema van de startprocedure, zoals die bij de numerieke tijdsintegratie wordt gebruikt.

Het rooster waarop wordt gerekend is een snede uit het FUNT-FUNA rooster.

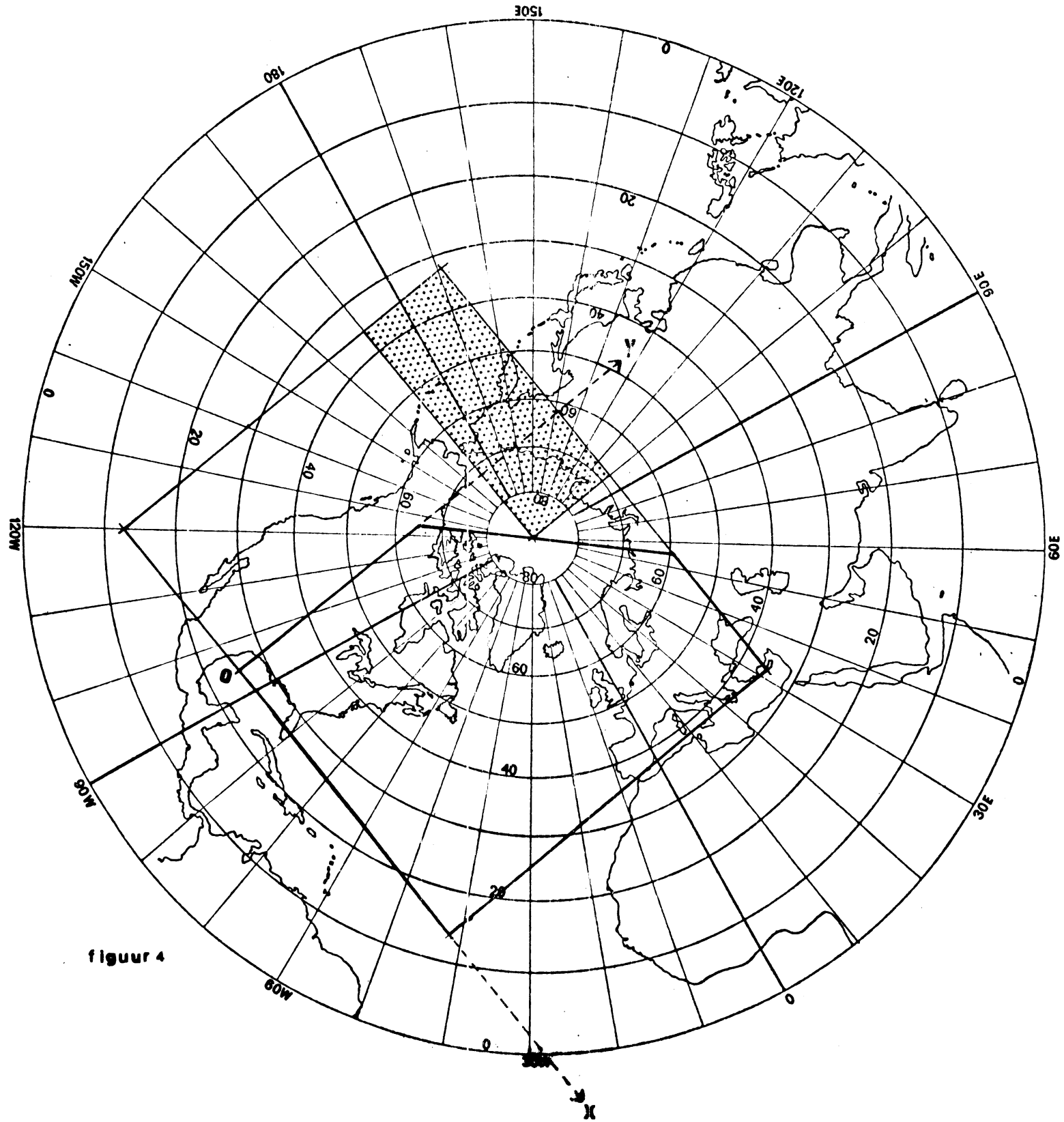
In figuur 4 is zowel het complete FUNT-FUNA rooster als het reken-rooster aangegeven. Dat er op een gedeelte van het FUNT-FUNA rooster wordt gerekend, komt omdat iedere dag de informatie uit het gearceerde gebied van het rooster ontbreekt. Een andere reden is, dat het hier een korte-termijn prognose betreft (24 uur), zodat het niet nodig is op een dergelijk groot rooster te rekenen.

De roosterpuntafstand is 381 km op 60° N en het aantal roosterpunten is 480.

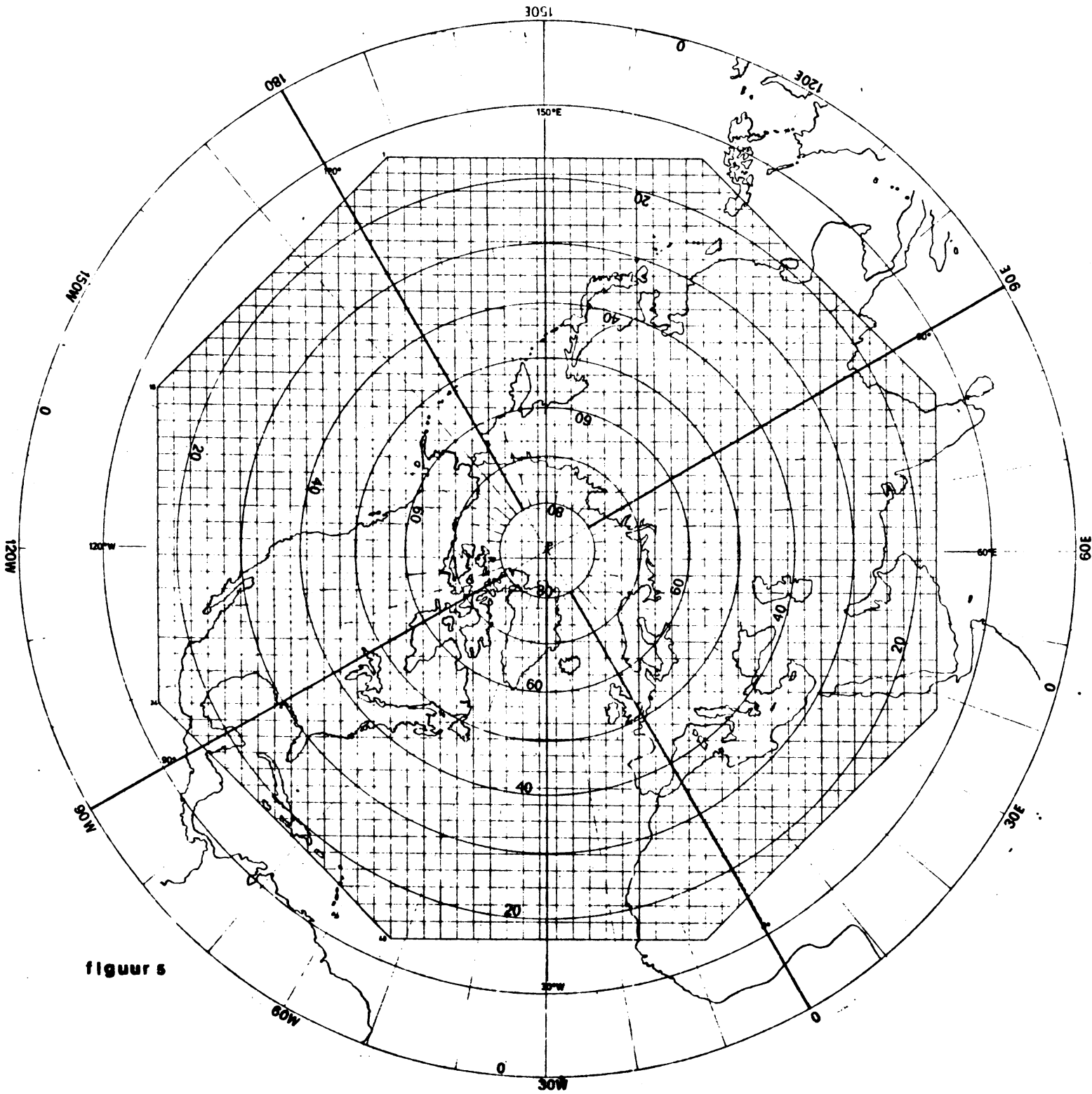
De tijd die nodig is voor een 24-uur prognose is ongeveer 1 minuut op de B6700.

Het programma dat werd ontwikkeld ten behoeve van wetenschappelijke experimenten rekt op een achthoekig rooster, waarvan de rand op ongeveer 15° N ligt. De roosterpuntafstand is 375 km op 60° N en het aantal roosterpunten bedraagt 1884.

In figuur 5 wordt een afbeelding van dit rooster gegeven.



figur 4



IV. Opbouw van het rekenprogramma BKBAR-96

Zoals in de inleiding reeds is opgemerkt, is het operationele rekenprogramma van het 3-lagen model als uitgangspunt genomen voor de ontwikkeling van een barotroop programma. De opbouw van de beide programma's vertoont hierdoor een sterke gelijkenis. Een **essentieel** verschil is, dat in het barotrope programma geen gebruik wordt gemaakt van de parallelle processing. Dit is gedaan om dit programma toegankelijk te maken en op deze manier het gebruik van het barotrope model bij experimenten gemakkelijker te maken.

Het programma is geschreven in BEA (Burroughs Extended Algol). Het is opgebouwd uit een aantal in- en uitvoer- en rekenprocedures, die in een "outer block" achtereenvolgens worden aangeroepen. Aan het einde van dit hoofdstuk wordt een complete listing van het programma gegeven.

Allereerst worden de belangrijkste constanten uit het rekenprogramma beschreven. Dan wordt beschreven op welke wijze de positie van de roosterpunten wordt vastgelegd; vervolgens wordt vermeld welke velden er in de diverse arrays zijn opgenomen. Hierna kunnen de verschillende procedures worden beschreven en tot slot wordt in een stroomdiagram het "outer block" weergegeven.

- Programmaconstanten

In het rekenrooster van fig. 4 is de oorsprong en de ligging van het assenstelsel aangegeven. Dit coördinatenstelsel moet bekend zijn om sommige modelconstanten te kunnen begrijpen. De belangrijkste constanten, die in het programma worden gebruikt, zijn:

CRES	Cressman-correctie
D	roosterpuntafstand
KDIF	diffusieconstante
DT	tijdstap in seconden
DT2	2 maal DT
VRT	tijdstap in uren
H	2 maal de hoeksnelheid van de aarde

II2	x-coördinaat van de noordpool in aantal roosterpunten
JJ2	y-coördinaat van de noordpool in aantal roosterpunten
MM	aantal roosterpunten in x-richting
NN	aantal roosterpunten in y-richting
TOT	totale aantal roosterpunten
INVERS	regelt het inverteren van de lineaire balansvergelijking
LINEAIR	regelt het oplossen van de balansvergelijking
LABDA	overrelaxatiefactor die wordt gebruikt bij de oplossing van Poisson- en Helmholtz-vergelijkingen
MAX	maximale residu na een iteratie
EPS	voorgeschreven waarde waar MAX beneden moet komen voor voldoende convergentie
STAP	maximale aantal iteraties
SHIFT	datum-tijdgroep van de uitgangstoestand
ENDFP	prognosetermijn in uren
PROG	prognose in uren tijdens het rekenproces. Het programma wordt beëindigd als PROG=ENDFP
R	telt het aantal malen dat het "outer block" wordt doorlopen (aantal tijdstappen)
STR	straal van de aarde

- Roosterprogrammering

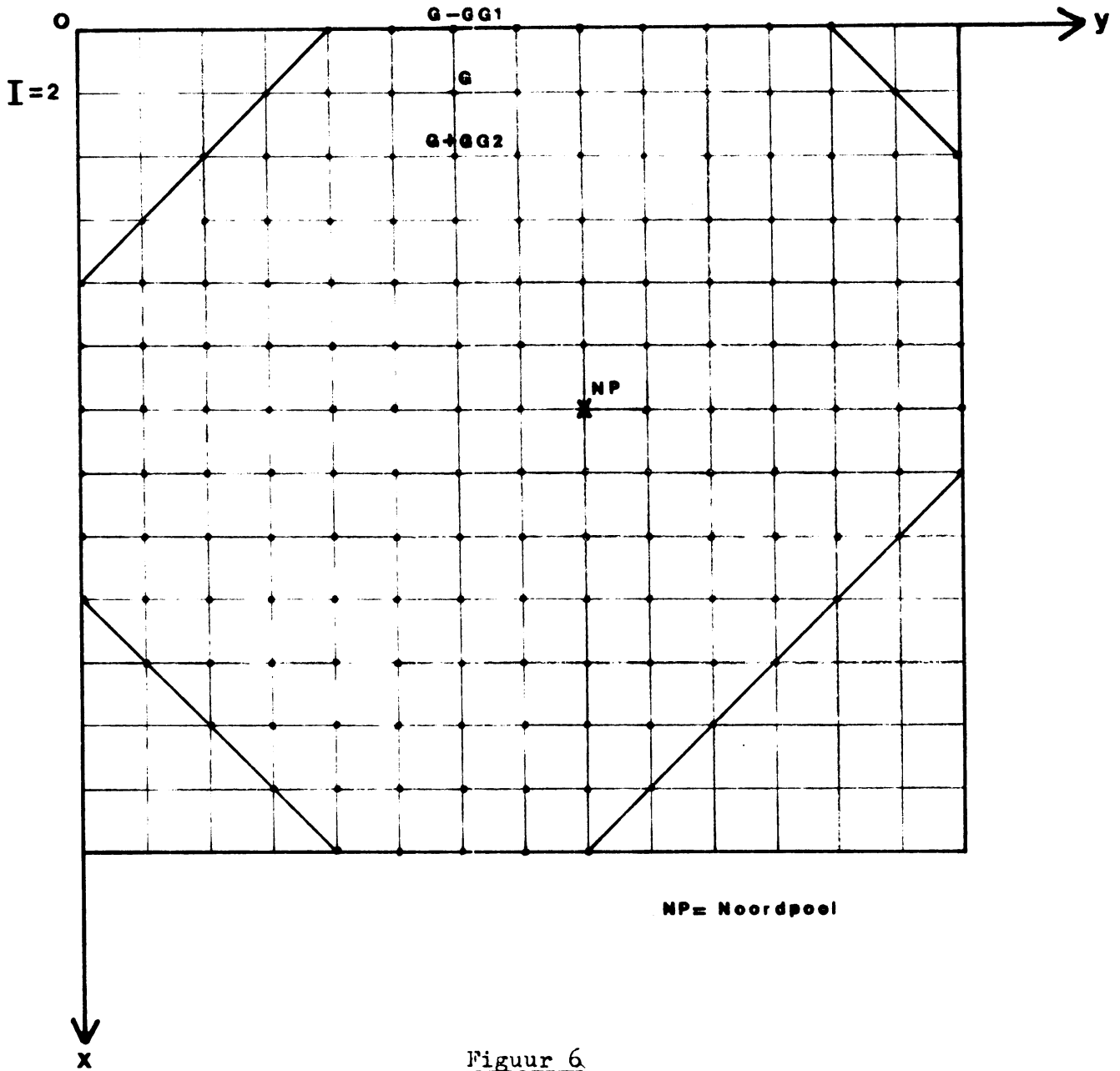
De wijze waarop het rooster wordt vastgelegd en doorlopen werd ontwikkeld door L.C. Heijboer. In principe wordt uitgegaan van een rechthoekig rooster.

De ligging van dit rooster wordt vastgelegd door de coördinaten van de noordpool II2 en JJ2 en door de richting van de x-as (zie fig. 6).

Het aantal roosterpunten in x-richting is MM, in y-richting is dat NN.

Van deze rechthoek mogen hoeken worden afgesneden. Op deze wijze kan een achthoekig rooster ontstaan. Dit wordt aangegeven met de arrays LI [I] en RE [I].

Met LI [I] wordt aangegeven hoeveel roosterpunten er zijn verdwenen van de I^e rij door het afsnijden van hoeken aan de linkerkant van het rooster. Met RE [I] wordt hetzelfde aangegeven, maar nu voor de rechterkant van het rooster.



Figuur 6

Voorbeeld van de wijze waarop de roosterpunten zijn vastgelegd.

$NN = 15; MM = 14;$

$II2 = 7; \quad CC = NN-LI[2] -RE[2] = 11 ;$

$JJ2 = 9; \quad G = TEL[2] + J = 9+4 = 13;$

$LI[2] = 3; \quad GG1 = NN-LI[2] -RE[1] = 10 ;$

$RE[2] = 1; \quad GG2 = NN-RE[2] -RE[2]-LI[3] = 12;$

$TEL[2] = 9; \quad II1 = II-II2 = 2-7 = -5$

$JJ1 = J-JJ2 + LI[2] = 4-9+3 = -2$

Het array TEL[I] geeft aan hoeveel roosterpunten er in totaal liggen op de voorgaande I-1 rijen.

De velden worden opgeslagen in één-dimensionale arrays. Het eerste roosterpunt krijgt index 1. Eerst wordt het rooster in y-richting doorlopen. Als een rij is afgewerkt, wordt naar de volgende rij gesprongen.

De index van een willekeurig roosterpunt op de i^e rij is:

$$G = \text{TEL}[I] + J$$

waarbij $J = 1$ voor het eerste roosterpunt van die rij.

Het aantal roosterpunten dat op die rij dient te worden afgewerkt bedraagt:

$$CC = NN - LI[I] - RE[I]$$

Als G de index is van een willekeurig roosterpunt, kan de index van de omliggende roosterpunten worden gevonden door

$$G+1$$

$$G-1$$

$$G-GG1 \quad \text{met} \quad GG1 = NN-LI[I]-RE[I-1]$$

$$G+GG2 \quad \text{met} \quad GG2 = NN-RE[I]-LI[I+1]$$

Om de Coriolis-parameter en de kaartschaalfactor voor een roosterpunt te kunnen berekenen, moet de geografische breedte van dat punt bekend zijn. De bepaling van de breedte gebeurt met de volgende formule:

$$\sin \varphi = \frac{1 - \frac{s^2}{r^2(1+\sin \varphi_0)^2}}{1 + \frac{s^2}{r^2(1+\sin \varphi_0)^2}} \quad (17)$$

Hierin is:

s afstand op de stereografische projectie van het roosterpunt tot de noordpool

r straal van de aarde

φ_0 60° NB

s^2 van het J^e punt van de I^e rij is:

$$s^2 = (II1 * II1 + JJ1 * JJ1) * d^2$$

met $II1 = I - II2$

$$JJ1 = J - JJ2 + LI[I]$$

- Arrays

S50 stroomfunctieveld op tijdstip t
S50 MIN1 stroomfunctieveld op tijdstip t-Δt
DS50 tendensveld $\left(\frac{\partial \psi}{\partial t}\right)^*$
F Coriolisparameter
M2 kaartschaalfactor
VORT 50 vorticiteit op 500 mbar
FORC 1 advectietermen
RVA 500 geschaald veld van de relatieve vorticiteits-
advectie, bestemd voor uitvoer
CRESCON Cressmancorrectie

VORT 50 en FORC 1 worden tevens gebruikt voor tijdelijke opslag van tussenresultaten tijdens de berekeningen.

- Procedures

IN(A); haalt de 72-uurs.
ARRAY A[0]; FUNT-FUNA prognose uit de file (OPER) DCM-
FORECASTUS en zet dit veld in array FUNTFUNA
Uit dit array wordt het veld van het uiteindelijke
rekenrooster gehaald en in het array A geplaatst.
Als FUNT en/of FUNA niet aanwezig is, wordt dit
via de printer gemeld en het rekenprogramma wordt
beëindigd.
Deze inleesprocedure is in het programma, dat op
een hemisferisch rooster rekt, vervangen door
een procedure die 500 mbar-velden uit de KNMI-
archieven haalt.

BALANS (X,A,B); als X=LINEAIR wordt het rechterlid van de lineaire
VALUE X ; balansvergelijking (10) berekend. Dit veld wordt
ARRAY A,B[0]; opgeslagen in B.
Bij de aanroep van BALANS moet in A het hoogteveld
staan.
Als X=INVERS wordt het linkerlid van (9) berekend.
Dit veld wordt wederom opgeslagen in B, terwijl
in A nu het stroomfunctieveld moet staan.

- INITIALISATIE ; - bepaalt de datum van de uitgangstoestand (SHIFT).
- bepaalt de lengte van de prognose (ENDFP).
- het Coriolis- en kaartschaalfactorveld wordt uitgerekend.
- het hoogteveld wordt met IN ingelezen.
- de balansvergelijking wordt opgelost m.b.v. BALANS en OPLOSSING POISSON.
- Het Cressmancorrectieveld (CRESCON) wordt uitgerekend.
- PLOTMAPS; verzorgt de uitvoer van prognoses van de plotter. Deze procedure is door Philippa geschreven.
- PRINTVELD verzorgt de uitvoer van prognoses van de regeldrukker. Deze procedure is als intrinsic in de B6700 aanwezig en wordt gebruikt in het programma dat op een hemisferisch rooster rekent. Een gebruiksaanwijzing van PLOTMAPS en/of PRINTVELD is bij MBW verkrijgbaar.
- VORTICITEIT (A); berekent de vorticititeit van het veld A m.b.v. ARRAY A; formule (14). Dit veld wordt in VORT 50 opgeslagen.
- FORCING 1 berekent de advectie van absolute vorticititeit m.b.v. formule (15). Dit veld wordt opgeslagen in FORC 1. Tevens wordt een geschaald veld van de relatieve vorticititeitsadvectie berekend (RVA 500), dat bestemd is voor uitvoer via PRINTVELD.
- HAAS; rekent het stroomfunctieveld op $t+\Delta t$ uit met formule (16). Als echter de programmateller R gelijk is aan 0, wordt (16a) gebruikt en wanneer $R=1$ wordt (16b) gebruikt. Hierna wordt gladgestreken langs de rand op de manier zoals dat in hfdst. I is beschreven. Dit gebeurt met IPOL.
- INVERSIE; invertteert de lineaire balansvergelijking, zodat uit het stroomfunctieveld een hoogteveld wordt verkregen. De inversie geschiedt m.b.v. BALANS en OPLOSSING POISSON.

SMOOTH (P,A);
REAL P;
ARRAY A [0];

strijkt het veld A in het binnengebied van het rooster glad met de Shuman-gladstrijker. Dit gebeurt in R. De 2e en 3e rij roosterpunten vanaf de rand worden op eenvoudige manier met IPOL gladgestreken.

Er wordt op t=0 en verder na iedere 12 uur gladgestreken. Als het aantal iteraties bij het oplossen van de Helmholtz-vergelijking groter wordt dan het aantal iteraties dat nodig was bij het oplossen van deze vergelijking tijdens de eerste rekenstap, wordt SMOOTH eveneens toegepast. Met P wordt geregeld het aantal keren dat met de Shuman-operator het veld wordt gladgestreken. Eerst worden de rijen van het rooster en vervolgens de kolommen gladgestreken.

POISSON (P,A,B);
INTEGER P;
ARRAY A,B [0];

voert 1 iteratiestap uit bij de oplossing van een Poisson- of Helmholtz-vergelijking. De gebruikte oplossingsmethode is de Liebmann-relaxatiemethode. In hfdst. I wordt beschreven welke startvelden worden genomen bij de oplossing van Poisson- en Helmholtz-vergelijkingen.
p=2 als de lineaire balansvergelijking moet worden opgelost. De randvoorwaarden worden dan op S opgelegd.
p=3 als de Helmholtz-vergelijking voor het tenveld DS 50 moet worden opgelost. Randvoorwaarden worden in dit geval op S-1 opgelegd.
Bij de aanroep van POISSON moet het rechterlid van Poisson- of Helmholtzvergelijking in B staan, terwijl in veld A de oplossing van de vergelijking wordt opgeslagen.

OPLOSSING POISSON(P,A,B);
INTEGER P;
ARRAY A,B [0];

in deze procedure wordt POISSON in een "loop" gezet. Na iedere iteratie wordt nagegaan of er nog een iteratiestap nodig is. Deze is nodig als het aantal iteraties kleiner is dan STAP en als het maximale residu (MAX) groter is dan een voorgeschreven waarde (EPS).
Voor p, A en B geldt hetzelfde als bij de beschrijving van POISSON is opgemerkt.

IPOL (K,DP);
REAL K;
ARRAY DP [0];

strijkt het veld aan de rand van het reken-
gebied op eenvoudige wijze glad door een simpele
interpolatie. IPOL wordt toegepast in HAAS op
de 2e rand (S-1) en in SMOOTH op de 2e en 3e
rand (S-2). Er wordt geïnterpoleerd met for-
mule (8).

Als K=2 wordt S-1 gladgestreken en als K=3 wordt
S-2 gladgestreken.

DP is het veld dat moet worden gladgestreken.

UITVOER

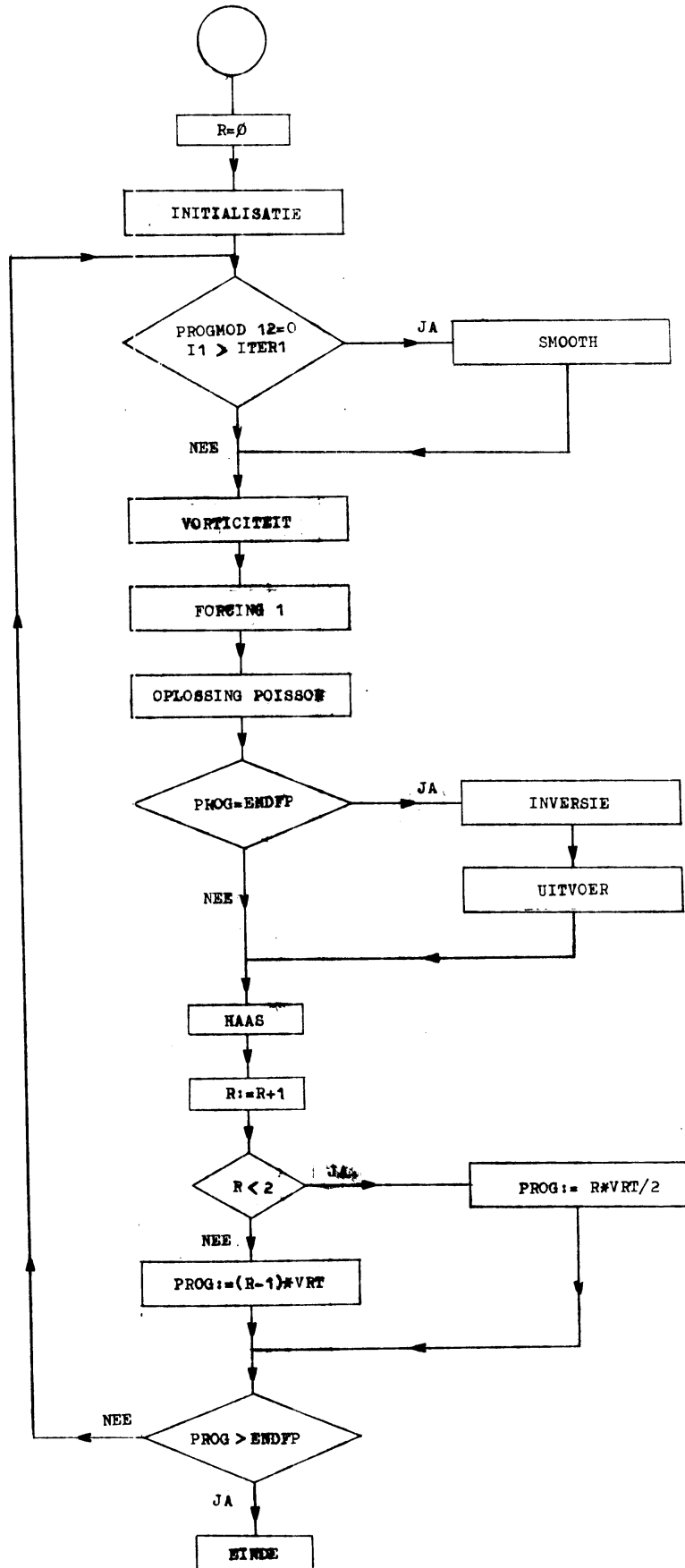
In deze procedure wordt PLOTMAPS en/of PRINTVELD
aangeropen. Deze procedures verzorgen de uit-
voer van ieder gewenst veld.

In BKBAR 96 wordt alleen de 96-uur prognose
geplot van het 500 mbar hoogteveld en van de
relatieve vorticitateitsadvectie.

Literatuur

- Galen, J. van, 1969 : Baroklien 3-parameter model. Wetenschappelijk
Rapport van het KNMI. No. W.R. 69-3, pp 73-90.
- Haltiner, G.J., 1971: Numerical Weather Prediction. John Wiley & Sons,
inc.
- Heijboer, L.C., 1977 : Design of a baroclinic three-level quasi-
geostrophic model with special emphasis on developing short
frontal waves. Uit de serie Mededelingen en Verhandelingen
van het KNMI. No. 98.
- Opsteegh, J.D., 1975: Experimenten met een 3-lagen model op een
hemisferisch rooster. Intern. memorandum van het KNMI.
No. 75-027.

STROOMDIAGRAM VAN HET "OUTER BLOCK"



PROGRAMMA SYMBOL/MDV95 ON TEST, BLAD 1

```

$SET VECTORMODE RESET LINEINFO
$SET INSTALLATION: NOHINDINFO OPTIMIZE
$LEVEL 2
$CEGURL HMBARIZ
$BEGIN
$SET OMIT

* * * * *
*
* K N M I H K B A R O T R O O P
*
* B A R C T R O P E V O O R S P E L I I N G
*
* V O O R 1 N I V E A U
*
* * * * *
$POP OMIT $SET STACK
$REAL RIZ
$POP STACK
EBCDIC ARRAY TITLE(0:255),LBUF(0:131):X
FILE LINE(KIND=PRINTER,LABELTYPE=OMITTED,UNITS=(CHARACTERS));
DEFINE
BKBA = 2 #X VOLGNUMMER
,CRES = 0.53-12 #X CRESSMANCORRECTIE
,D = 38123 #X ROOSTERPUNTSAFSTAND
,DIGITSIN(X) = (FIRSTONE(SCALERRIGHTF(X,12))-1) DIV 4+1 #X
,DT = (3600+VRT) #X
,DT2 = (7200+VRT) #X
,H = (PI/21600) #X
,I12=6#X
,INVERS = 1 #X
,JJ2=20#X
,KDIF = 0#X DIFFUSIECONSTANTE
,LINEAIR = 0 #X
,MM = 21 #X
,NEWPAGE = WRITE(LINE(SKIP 1)) #X
,NN = 26 #X
,P = 3 #X
,PI = 3.14159265 #X
,STAP = 22 #X MAXIMUM AANTAL ITERATIES POISSON & HELMHOLTZ
,TOT = 480 #X
,VRT = 1 #X
;X
REALX
I,J,PROG,ROT2,BB,CC,STR,I1,I2,G,KK1,KK2,ENCFP,SHIFTX
,SG,L
,DZ,EPS,BR,PP
    
```

PROGRAMMA SYMBOL/MDV96 ON TEST, BLAD 2

```

, LABDA X RELAXATIEFACTOR
, I1,JJ1,TEMP
, S
, Q1,Q2,Q3,K
, ITER1
, MP,SP
, MAX
;
LABEL NIET;
EVENT SMOOTHING;
BOOLEAN HEADING;
VALUE ARRAY
LI(22(0)),
RE(12,11,10,9,8,7,6,5,4,3,2,1,10(0)),
TEL(0,0,15,31,48,66,85,105,126,148,171,195,220,
246,272,298,324,350,376,402,428,454);
PROCEDURE RTLINE:X
BEGIN
WRITE(LINE,132,LBUF):X
REPLACE LBUF BY " " FOR 22 WORDS;
END:XP R O C E D U R E R T L I N E
PROCEDURE IN(A)ARRAY A(0);
BEGIN
REAL I,J,K,L,COUNT:X
ARRAY FUNTFUNAC(831),FORECASTSAR(416),FPRESENT(0:9);
LABEL EOF:X
EBCDIC ARRAY TITLE(0:252);
FILE FORECASTSFILE(KIND=DISK,FILETYPE=7):X
COUNT=-1;FILL FPRESENT WITH 9(-1);
REPLACE TITLE BY "(OPER)DCH/FORECASTUS/",SHIFT FOR 8 DIGITS,* ON OPER.*
REPLACE FORECASTSFILE,TITLE BY TITLE:X
IF FORECASTSFILE.RESIDENT THENX
WHILE TRUE DOX
BEGINX
READ(FORECASTSFILE,417,FORECASTSAR)(EOF):X
IF FORECASTSAR(0)<9 THEN FPRESENT(FORECASTSAR(0)):=COUNT:=+1:X
END:X
EOF:
IF FPRESENT(2)>-1 AND FPRESENT(8)>-1 THENX
BEGINX
FOR L:=2,9 DOX
BEGINX
K:=IF L=8 THEN 0 ELSE 16:Z
READ(FORECASTSFILE)(FPRESENT(L),417,FORECASTSAR):X
FOR I:=0 STEP 1 UNTIL 25 DOX
FOR J:=0 STEP 1 UNTIL 15 DOX
FUNTFUNAC(K+J*I*32):=FORECASTSAR(I+16+J*I):Z
END:
    
```

PROGRAMMA SYMBOL/MDV96 ON TEST, BLAD 3

```

FOR I:=1+I+1 WHILE I LEQ MM DO
BEGIN
  CC:=NN-RE(I);GG:=TE(I);
  FOR J:=1+J+1 WHILE J LEQ CC DO
  BEGIN
    G:=G+J;
    A(I):=FUNTFUNA(I+10*(J-1)*32);
  END;
END;
END ELSEY
BEGIN
  WRITE(LINE, <<"FUNT EN/DE FUNA IS NIET AANWEZIG">>);
  MYSELF.STATUS:=-1;
END;
END;

PROCEDURE IPOL(K,DP);
  VALUE K; REAL K; ARRAY DP(0);
  FORWARD;
PROCEDURE OPLOSSINGPOISSON(P,A,B);
  VALUE P; REAL P; ARRAY A,B(0);
  FORWARD;

LONG ARRAY
  S50 X
  , S50MINI X
  , DS50 X
  , F X CDRIOLISPARAMETER
  , M2 X SCHAALFACTOR
  , VORT50 X
  , FORC1 X
  , RVA50G X
  , CRESCON X
  (U:TOT);

PROCEDURE BALANS(X,A,B);
  VALUE X; REAL X; ARRAY A,B(0);
  BEGIN
    REAL AID,AID1,AID2,AID3,AID4,H1,X
      I,J,CC, KK1, KK2, G, GG, GG1, GG2, C, D;
    DEFINE LINEAIREBALANS=(C*H1-D/2)*98.1/4.**2#;
      INVERSEBALANS=(C*H1+D/2)#;
    BB:=MM-1;
    FOR I:=2 STEP 1 WHILE I LEQ BB DO
    BEGIN
      CC:=NN-(KK1:=LI(I))-(KK2:=RE(I))-1;
      GG1:=NN-KK1-RE(I-1);
      GG2:=NN-KK2-LI(I+1);
      GG:=TE(I);
      FOR J:=2 STEP 1 WHILE J LEQ CC DO

```

PROGRAMMA SYMBOL/MDV96 ON TEST, BLAD 4

```

BEGIN
  C:=(AID):=A(G:=GG+J)*4+(AID1:=A(G+1))+
    (AID2:=A(G-GG1))+(AID3:=A(G-1))+AID4:=A(G+GG2);
  D:=(F(G+1)-H1:=F(G))*(AID1-AID)+(F(G-1)-H1)*(AID3-AID);
  (F(G-GG1)-H1)*(AID2-AID)+(F(G+GG2)-H1)*(AID4-AID);
  B(G):=CASE X OF (LINEAIREBALANS, INVERSEBALANS);
END;
END;
END;
PROCEDURE BALANS
PROCEDURE INITIALISATIE
  REAL HULP,HULP1,HULP5,SI;
  SHIFT:=MYSELF.TASKVALUE;
  SHIFT:=* DIV 100*100;
  ENDFP:=96;
  PROG:=72;
  REPLACE LBUF BY "VRT,VOORSPELPERIODE IN UREN ZIJN RESP.:";
  TEMP:=ENTIER(VRT) FOR 1 DIGITS;"", (VRT-TEMP)*22 FOR 2 DIGITS;
  " ", ENDFP FOR 2 DIGITS;
  RTLINE;
  RTLINE;
  HULP:=(D2:=D**2)/STR:=6371229**2*BR:=(SIN(PI/3)+1)**2;
  FOR I:=1 STEP 1 UNTIL MM DOX
  BEGIN
    CC:=NN-(KK1:=LI(I))-KK2:=RE(I);
    GG:=TE(I);
    II:=I-II2;
    FOR J:=1 STEP 1 UNTIL CC DOX
    BEGIN
      JJ1:=J-JJ2+KK1;
      PP:=(II*II+JJ1+JJ1)*HULP;
      FG:=GG+JJ:=(SI:=(1-PP)/(1+PP))*H;
      M2(G1:=BR/(SI+1)**2);
    END;
  END;
  IN(S50);
  BALANS(LINEAIR, S50, FORC1);
  FOR G:=1 STEP 1 UNTIL TOT DO
  BEGIN
    S50(G1:=**9.8125;
  END;
  EPS:=.04;
  LABDA:=0.45887;
  OPLOSSINGPOISSON(2, S50, FORC1);
  EPS:=.01;
  IF CRES NEQ 0 THEN LABDA:=0.375;
  FOR G:= 1 STEP 1 UNTIL TOT DO
  BEGIN
    S50MINI(G):=S50(G);

```


PROGRAMMA SYMBOL/MDV96 ON TEST. BLAD 5

```

HULP5:=F(G)/2-4;
CRESCON(G):=HULP5+HULP5*GRES*D2/M2(G);
END;
END: P R O C E D U R E I N I T I A L I S A T I E
PROCEDURE VORTICITEIT(A);
ARRAY A(0);
BEGIN
  REAL X1,X2,X3,H,H1,H2,H3,
  I,J,CC,KK1,KK2,G,GG,GG1,GG2;X
  M:=MM-1;
  FOR I:=2 STEP 1 WHILE I LEQ BB DO
  BEGIN
    CC:=NN-(CKK1:=LI(I))-(CKK2:=RE(I))-1;X
    GG1:=-RE(I-1)+NN-KK1;X
    GG2:=-LI(I+1)+NN-KK2;X
    X2:=A(H1:=(GG:=TEL(I))+1);
    X3:=A(H2:=GG+2);
    FOR J:=2 STEP 1 WHILE J LEQ CC DO
    BEGIN
      H:=M2(G:=GG+J)/D2;X
      X1:=READLOCK(READLOCK(A(H1:=G+1),X3),X2);X
      VORT50(G):=(A(H2:=G-GG1)+A(H3:=C+GG2)-4*X2+X1+X3)*H;
    END;X
  END;X
END: P R O C E D U R E V O R T I C I T E I T
PROCEDURE FORCING1;
BEGIN
  REAL
  GS50X,GS50Y,GV50X,GV50Y,GFY,GFY,JAC1,JAC2,
  H50S1,H50S2,H50S3,H50S4,
  H50V1,H50V2,H50V3,H50V4,
  HF1,HF2,HF3,HF4,
  HULP1,HULP3,HULP4,M,
  I,J,CC,KK1,KK2,GG1,GG2,GG3,GG4,G,GG,
  GMIN1,GMIN2,GPLUS1,GPLUS2,
  GGG1,GGG2,GGG3,GGG4;X
  BOOLEAN BOOL1;X
  M:=MM-2;
  FOR I:=3 STEP 1 WHILE I LEQ BB DO
  BEGIN
    CC:=NN-(CKK1:=LI(I))-(CKK2:=RE(I))-2;X
    GG3:=(GG1:=-RE(I-1)+NN-KK1)-LI(I-1)-RE(I-2)+NN;X
    GG4:=(GG2:=-LI(I+1)+NN-KK2)-RE(I+1)-LI(I+2)+NN;X
    BOOL1:=I=3 OR I=BB;X
    H50S2:=S50(GMIN2:=(GG:=TEL(I))+1);
    H50S4:=S50(GPLUS1:=GG+4);
    H50V2:=VORT50(GMIN2);
    H50V4:=VORT50(GPLUS1);

```

PROGRAMMA SYMBOL/MDV96 ON TEST. BLAD 6

```

HF2:=F(GMIN2);
HF4:=F(GPLUS1);
FOR J:=3 STEP 1 WHILE J LEQ CC DO
BEGIN
  H50S1:=READLOCK(S50(GMIN1:=(G:=GG+)-1),H50S2);X
  H50S3:=READLOCK(S50(GPLUS2:=G+2),H50S4);X
  H50V1:=READLOCK(VORT50(GMIN1),H50V2);X
  H50V3:=READLOCK(VORT50(GPLUS2),H50V4);X
  HF1:=READLOCK(F(GMIN1),HF2);X
  HF3:=READLOCK(F(GPLUS2),HF4);X
  GS50X:=S50(GGG2:=G+GG2)-S50(GGG1:=G-GG1);
  GS50Y:=H50S3-H50S2;
  GV50X:=VORT50(GGG2)-VORT50(GGG1);
  GV50Y:=H50V3-H50V2;
  GFY:=F(GGG2)-F(GGG1);
  GFY:=HF3-HF2;
  IF J=CC OR J=3 OR BOOL1 THEN M:=1/4 ELSE
  BEGIN
    M:=1/14;X
    GGG3:=G+GG3;
    GGG4:=G+GG4;
    GS50X:=GS50X*8-S50(GGG4)+S50(GGG3);
    GS50Y:=GS50Y*8-H50S4+H50S1;
    GV50X:=GV50X*8-VORT50(GGG4)+VORT50(GGG3);
    GV50Y:=GV50Y*8-H50V4+H50V1;
    GFY:=GFY*8-F(GGG4)+F(GGG3);
    GFY:=GFY*8-HF4+HF1;
  END;X
  JAC1:=(GS50X*GV50Y-GS50Y*GV50X)*M;
  JAC2:=(GS50X*GFY-GS50Y*GFY)*M;
  FORC1(G):=-JAC1-JAC2;
  RVA500(G):=-HULP4:=M2(G)/D2)+JAC1+Q10;
END;X J = L O O P
END;X I = L O O P
END: P R O C E D U R E F O R C I N G 1
PROCEDURE HAAS;
BEGIN
  REAL H1,H2,H3,G;
  FOR G:=1 STEP 1 UNTIL TOT DO VORT50(G):=0;
  VORTICITEIT(S50MIN1);
  IF R<2 THEN
  BEGIN
    ROT2:=(R+1)*DT/2;X
    FOR G:=1 STEP 1 UNTIL TOT DO
    BEGIN
      S50(G):=(VORT50(G)*KOIF+DS50(G))*ROT2+S50MIN1(G);X
    END;X
  END ELSE

```

PROGRAMMA SYMBOL/MDV96 ON TEST. BLAD 7

```

BEGINX
  FOR G:=1 STEP 1 UNTIL TOT DO
    BEGINX
      S50MINI(G):=H1:=Z
      READLOCK((VORT50(G)+KDIF+DSSC(G))*D(2+S50MINI(G),S50(G));
    ENDX
  ENDX
  IPOL(2,S50);
ENDX LEA P F R D G
00301000
00302000
00303000
00304000
00305000
00306000
00307000
00308000
00309000
00310000
00311000
00312000
00313000
00314000
00315000
00316000
00317000
00318000
00319000
00320000
00321000
00322000
00323000
00324000
00325000
00326000
00327000
00328000
00329000
00330000
00331000
00332000
00333000
00334000
00335000
00336000
00337000
00338000
00339000
00340000
00341000
00342000
00343000
00344000
00345000
00346000
00347000
00348000
00349000
00350000

PROCEDURE INVERSIE;
BEGINX
  BALANS(INVERS,S50,VORT50);
  FOR G:=1 STEP 1 UNTIL TOT DO
    BEGIN
      FORC1(G):=S50(G)+3-4;
      CRESCON(G):=0;
    END;
  EPS:=1;
  LABDA:=0.45887;
  DPLLOSSINGPOISSON(2,FORC1,VORT50);
  EPS:=0.1;
  IF CRES NEQ 0 THEN LABDA:=0.375;
  FOR G:=1 STEP 1 UNTIL TOT DOX
    BEGINX
      FORC1(G):=*/98.1;X
      CRESCON(G):=(FCG/2-4)**2*CRES*D2/M2(G);
    ENDX
  ENDX P R O C E D U R E I N V E R S I E
00329000
00330000
00331000
00332000
00333000
00334000
00335000
00336000
00337000
00338000
00339000
00340000
00341000
00342000
00343000
00344000
00345000
00346000
00347000
00348000
00349000
00350000

PROCEDURE SMOOTH(P,A); VALUE P; REAL P; ARRAY A(0);X
BEGIN
  REAL I;
  PROCEDURE R(U,OP);X
  VALUE U; REAL U; ARRAY DP(0);X
  BEGINX
    REAL I,J,CC,G,GG,KK1,KK2,GG1,GG2,GG3,GG4,GG5,GG6;X
    BB:=MM-3;
    FOR I:=4 STEP 1 WHILE I LEQ BB DOX
      BEGINX
        CC:=NN-(KK1:=LI(I))-(KK2:=RE(I))-3;X
        GG:=TELI;X
        G:=GG+4;X
        J:=GG+CC;X
        CASE U OF X
          BEGINX
            O: DO FORC1(G):=(DP[G-1]+DP[G+1])*0.22049-(DP[G-2]+DP[G+2]);X

```

PROGRAMMA SYMBOL/MDV96 ON TEST. BLAD 8

```

  *11317+(DP[G-3]+DP[G+3])*0.2886-DP[G]*0.27236;X
  UNTIL G:=*+1>J;X
  1: DO DP(G):=**FORC1(G) UNTIL G:=*+1>J;X
  2: GG1:=NN-KK1-RE[I-1];
  GG2:=NN-KK2-LI[I+1];
  GG3:=GG1+NN-LI[I-1]-RE[I-2];
  GG4:=GG2+NN-RE[I+1]-LI[I+2];
  GG5:=GG3+NN-LI[I-2]-RE[I-3];
  GG6:=GG4+NN-RE[I+2]-LI[I+3];
  DO FORC1(G):=(DP[G-GG1]+DP[G+GG2])*0.22049-(DP[G-GG3]+X
  DP[G+GG4])*0.11317+(DP[G-GG5]+DP[G+GG6])*0.2886-DP[G];X
  *0.27236;X
  UNTIL G:=*+1>J;X
  ENDX
ENDX
PROCEDURE R
PROCEDURE(SMOOTHING);X
THRU P DO FOR I:=0,1,2,1 DO R(I,A);
LIBERATE(SMOOTHING);X
IPOL(3,A);
IPOL(2,A);
ENDX P R O C E D U R E S M O O T H
00351000
00352000
00353000
00354000
00355000
00356000
00357000
00358000
00359000
00360000
00361000
00362000
00363000
00364000
00365000
00366000
00367000
00368000
00369000
00370000
00371000
00372000
00373000
00374000
00375000
00376000
00377000
00378000
00379000
00380000
00381000
00382000
00383000
00384000
00385000
00386000
00387000
00388000
00389000
00390000
00391000
00392000
00393000
00394000
00395000
00396000
00397000
00398000
00399000
00400000

Q1:=A[G-1];
Q3:=A[G];
PP:Q2:=Q3;
G1:=G+1;
Q3:=A[G1];
HULP1:=CRESCON(G);
K:=(A[G+GG2]+A[G-GG1]+Q1+Q3)-(4+HULP1)*Q2-ECI)*L/(4+
HULP1);
Q1:=Q2+K;

```

```

PROGRAMMA SYMBOL/MOV96 ON TEST. BLAD 9

      A(G):=01;
      K:=ABS(K);
      IF MAX<K THEN MAX:=K;
      G:=G1;
      IF G LEQ J THEN GO TO PP;
    END
END%Z P R O C E D U R E P U I S S O N
PROCEDURE PLOUSSINGPOISSON(P,A,B);
  VALUE P;
  REAL A;
  ARRAY A,B(O);
  BEGIN
    LABEL TERUG;
    L:=LA90A*4;
    S:=0;
    TERUG:MAX:=0;
    POISSON(P,A,B);
    S:=S+1;
    IF S<STAP AND MAX>EPS THEN GO TO TERUG
  END%Z O P L O S S I N G P O I S S O N

PROCEDURE IPOL(K,DP);
  VALUE K; REAL K; ARRAY DP(O);%
  BEGIN%
    REAL I,J,BR,CC,KK1,KK2,G,GG,GG1,GG2;%
    BB:=MM-K+1;%
    FOR I:=K STEP 1 WHILE I LEQ BB DO%
      BEGIN%
        CC:=NN-(KK1:=LI(I))-(KK2:=RE(I))-K+1;%
        GG1:=NN-KK1-RE(I-1);
        GG2:=NN-KK2-LI(I+1);
        GG:=TEL(I);
        IF I=K OR I=BB THEN
$SET OMIT = VECTORMODE
          FOR J:=K STEP 1 WHILE J LEQ CC DO
            DP[G:=GG+J]:=DP[G+GG1]+DP[G+GG2])/2%
$FOP OMIT SET OMIT = NOT VECTORMODE
          DO VECTORMODE(DP[G:=GG+K],DPG1:=DP[G+GG1],DPG2:=DP[G+GG2],
            FOR CC=K+1)%
            BEGIN%
              DP:=(DPG1+DPG2)/2;%
              INCREMENT DP,DPG1,DPG2;%
            END%
$FOP OMIT
          ELSE%
            BEGIN%
              DPG:=GG+K:=IF LI(I) NEQ 0 THEN
                (DP[G+GG1]+DP[G+GG2]+DP[G-1]+DP[G+1])/4 ELSE
                (DP[G-1]+DP[G+1])/2;
              DPG:=GG+CC:=IF RE(I) NEQ 0 THEN

```

```

PROGRAMMA SYMBOL/MOV96 ON TEST. BLAD 10

      (DP[G+GG1]+DP[G+GG2]+DP[G-1]+DP[G+1])/4 ELSE
      (DP[G-1]+DP[G+1])/2;
    END%Z
  END%Z
END%Z P R O C E D U R E I P O L

PROCEDURE PRINTVELD(A,B,STRINGA)
  "COORDINATEN MERKTEKENS="(LS),BR1,LS2,BR2)
  " "(X1,X2,Y1,Y2,SH,PM,SHIFT,PROG);%
  VALUE A,B,LS1,BR1,LS2,BR2,X1,X2,Y1,Y2,SH,SHIFT,PROG;%
  REAL A,B,LS1,BR1,LS2,BR2,X1,X2,Y1,Y2,SH,SHIFT,PROG;%
  ARRAY PM(O);%
  STRING STRINGA;%
  FILE FF;%
  BEGIN%
    LABEL EINDREG;
    INTEGER REG1,REG2,POS1,POS2,QK;%
    REAL K,KK,REGEL,R2,P1,P2,P3,P4,P5,I2,I3,J2,J3,IX,TY,TY2,TY3,TY4,%
RN1,RN2,%
    LN1,LN2,INDEX,SH1,SH2,S,X,Y,P,PP,H1,H2,[4],[5],STL,%
    V1,V2,V3,V4,V5,K1,K2,K3,K4,K5,AB,I,J,G,G1,GG2;%
    EBCDIC ARRAY DPM(O:TOT),LBUF(O:131);%
    ARRAY F(O:TOT);%
    BOOLEAN BOOL,R2,R3,R4,GG6,GG7,RAND,HARK;%
    PROCEDURE RTLINE;%
    BEGIN%
      WRITE(FF,132,LBUF);%
      REPLACE LBUF BY " " FOR 22 WORDS;%
    END%Z P R O C E D U R E R T L I N E
    PROCEDURE KERN(BOOL); VALUE BOOL; BOOLEAN BOOL;%
    BEGIN%
      REAL KRN,M;%
      INTEGER PROCEDURE GEM(A,KK);
      ARRAY A(O);
      REAL KK;
      BEGIN%
        REAL S;%
        FOR M:=1 STEP 1 UNTIL 3 DO S:=A[KK+M]+S;%
        GEM:=QK-S/7
      END%Z
      V1:=GEM(F,K1);
      V2:=GEM(F,K2);
      V3:=GEM(F,K3);
      V4:=GEM(F,K4);
      V5:=GEM(F,K5);
      IF V1>0 AND V2>0 AND V3>0 AND V4>0 AND V5>0 AND BCOL
      THEN KRN:="H" ELSE

```

```

00401000
00402000
00403000
00404000
00405000
00406000
00407000
00408000
00409000
00410000
00411000
00412000
00413000
00414000
00415000
00416000
00417000
00418000
00419000
00420000
00421000
00422000
00423000
00424000
00425000
00426000
00427000
00428000
00429000
00430000
00431000
00432000
00433000
00434000
00435000
00436000
00437000
00438000
00439000
00440000
00441000
00442000
00443000
00444000
00445000
00446000
00447000
00448000
00449000
00450000

00451000
00452000
00453000
00454000
00455000
00456000
00457000
00458000
00459000
00460000
00461000
00462000
00463000
00464000
00465000
00466000
00467000
00468000
00469000
00470000
00471000
00472000
00473000
00474000
00475000
00476000
00477000
00478000
00479000
00480000
00481000
00482000
00483000
00484000
00485000
00486000
00487000
00488000
00489000
00490000
00491000
00492000
00493000
00494000
00495000
00496000
00497000
00498000
00499000
00500000

```


PROGRAMMA SYMBOL/MDV96 ON TEST. BLAD 13

```

                                F(P3),F(P2));X                                00601000
                                END;X                                        00602000
REPLACE DPM(INDEX): IF R366 THEN P5 ELSE IF R2 THEN P1 ELSE                00603000
IF R3 THEN P2 ELSE IF R4 THEN P3 ELSE P4) BY KK.[7:48] FOR 1;X           00604000
                                F(INDEX);S;X                               00605000
                                END;X                                        00606000
END;X * * * * P O S I T I E;X                                           00607000
IF RGG6 THEN X                                                            00608000
BEGIN X                                                                    00609000
    K:=4;X                                                                    00610000
    DO X                                                                      00611000
    BEGIN X                                                                    00612000
        K5:=(K4:=(K3:=(K2:=(K1:=K)+TY)+TY)+TY)+TY;X                       00613000
        IF DPM(K3-3) NEQ "*" FOR 1 AND DPM(K1+3) NEQ "*" FOR 1 THEN X     00614000
        BEGIN X                                                                00615000
            OK:=F(K3);X                                                         00616000
            V1:=OK-F(K1);X                                                       00617000
            V2:=OK-F(K5);X                                                       00618000
            V3:=OK-F(K3-3);X                                                     00619000
            V4:=OK-F(K3+3);X                                                     00620000
            IF NOT (AB=10 OR AB=60) THEN X                                       00621000
            IF V1>0 AND V2>0 AND V3>0 AND V4>0 THEN KERN(TRUE) ELSE X     00622000
            IF V1<0 AND V2<0 AND V3<0 AND V4<0 THEN KERN(FALSE);X         00623000
        END;X                                                                    00624000
    END UNTIL K:=*+1>TY-3;X                                                  00625000
    IF REGEL=REG1 THEN REPLACE DPM(POS1) BY "*" FOR 1 ELSE X               00626000
    IF REGEL=REG2 THEN REPLACE DPM(POS2) BY "*" FOR 1;X                     00627000
    REPLACE LBUF BY (IF MARK:=SH=30 AND (REGEL=9 OR REGEL=10)                00628000
    THEN "=" ELSE " ").[7:48] FOR 1;X                                         00629000
    DPM(TY2+1) FOR TY3-TY2;X                                                  00630000
    (IF MARK THEN "=" ELSE " ").[7:48] FOR 1;X                                00631000
    RTLINE;X                                                                    00632000
END;X                                                                        00633000
IF R2 OR R3 OR REGEL=TX THEN X                                              00634000
BEGIN X                                                                    00635000
    BOOL:=REGEL=TX;X                                                           00636000
    DO X                                                                      00637000
    BEGIN X                                                                    00638000
        REPLACE LBUF BY (IF MARK:=SH=30 AND (REGEL=9 OR REGEL=10)           00639000
        THEN "=" ELSE " ").[7:48] FOR 1;X                                       00640000
        DPM(IF R2 THEN 0 ELSE IF R3 THEN TY ELSE X                             00641000
        IF BOOL THEN TY3 ELSE TY4)+1) FOR TY;X                                 00642000
        (IF MARK THEN "=" ELSE " ").[7:48] FOR 1;X                            00643000
    END UNTIL READLOCK(FALSE,BOOL);X                                         00644000
    END;X                                                                        00645000
    EINDREG;X                                                                  00646000
END;X * * * * R E G E L;X                                                  00647000
END;X                                                                        00648000
RTLINE;X                                                                    00649000
                                00650000

```

PROGRAMMA SYMBOL/MDV96 ON TEST. BLAD 14

```

REPLACE LBUF BY X                                                            00651000
"STIPPENBAND VAN ",A FOR DIGITSIN(A) DIGITS," TOT ",A+B FOR X              00652000
DIGITSIN(A+B) DIGITS,CASE REAL(PROG=0) OF (" PROG ",AAL " ),X             00653000
SHIFT DIV 100 FOR 6 DIGITS," ",SHIFT MOD 100 FOR 2 DIGITS X                00654000
," " ,PROG FOR 2 DIGITS," ",STRINGA+6 FOR REAL(STRINGA,6);X                00655000
RTLINE;X                                                                    00656000
NEWPAGE;X                                                                    00657000
END;X P R I N T V E L D;X                                                    00658000
PROCEDURE UITVOER;X                                                         00659000
BEGIN X                                                                    00660000
    REAL H2;X                                                                    00661000
    NEWPAGE;X                                                                    00662000
    PRINTVELD(LINE,552,6,"500 HBAR",50,60,0,40,8,21,8,26,30,X             00663000
    FORC1,SHIFT,PROG);X                                                         00664000
    PRINTVELD(LINE,0,10,"RVA 500 HBAR",50,60,0,40,8,21,8,26,X             00665000
    30,RVA500,SHIFT,PROG);X                                                    00666000
END;X P R O C E D U R E U I T V O E R;X                                       00667000
                                00668000
SPACE;X                                                                    00669000
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 00670000
ZZ O U T E R B L O C K;X                                                       00671000
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 00672000
                                00673000
BEGIN X                                                                    00674000
REPLACE LBUF BY " " FOR 22 WORDS;X                                           00675000
NEWPAGE;X                                                                    00676000
REPLACE LBUF BY "BKFUNTFUNA "-" BKGA FOR DIGITSIN(BKGA) DIGITS,X           00677000
", ",COMPILETIME(15);X DATE COMPILED;X                                       00678000
", ",COMPILETIME(1) DIV (3600*60) FOR 2 DIGITS;X                             00679000
"; ",COMPILETIME(1) MOD (3600*60) DIV 3600 FOR 2 DIGITS;X                   00680000
RTLINE;X                                                                    00681000
WRITE(LINE[SPACE 15]);X                                                       00682000
NAME(LINE,"SHV WEERKAMER");X                                                  00683000
END;X                                                                        00684000
                                00685000
INITIALISATIE;X                                                             00686000
DOX T I M E L O D P;X                                                         00687000
BEGIN X                                                                    00688000
    IF I1>ITER1 OR PROG MOD 12=0 THEN X                                       00689000
    BEGIN X                                                                    00690000
        SMOOTH(P,S50);X                                                         00691000
        RTLINE;X                                                                00692000
        REPLACE LBUF BY "SMOOTH, PROG = ",PROG FOR 2 DIGITS;X               00693000
        RTLINE;X                                                                00694000
        RTLINE;X                                                                00695000
        HEADING:=TRUE;X                                                         00696000
    END;X                                                                        00697000
    VORTICITEIT(S50);X                                                         00698000
    FORCING1;X                                                                  00699000
    OPLOSSINGPOISSON(3,C550,FORC1);X                                           00700000

```

PROGRAMMA SYM40L/MOV96 ON TEST. BLAD 15

```

IF R=0 THEN I1:=ITER1:=S-1 ELSE I1:=S;
MP:=MAX;
SP:=S;
IF READLOCK(FALSE,HEADING) THEN;
BEGIN;
REPLACE LBUF BY;
" PROG SP MP";
RTLINE;
RTLINE;
END;
REPLACE LBUF(1) BY;
INTEGRT(PRG) FOR 2 DIGITS,"*"*PROG MOD 22*22 FOR 2 DIGITS,*
" *SP FOR 3 DIGITS." 0,"*MP*24 FOR 4 DIGITS;
RTLINE;
IF PROG MOD 24=0 THEN INVERSIE;
IF PROG MOD 24 = 0 THEN UITVOER;
HAAS;
END UNTIL PRG:=(IF R:=*+1<2 THEN R/2 ELSE R-1)*VRT+72>ENDFP;
NIET: NEWPAGE;
END.

```

```

00701000
00702000
00703000
00704000
00705000
00706000
00707000
00708000
00709000
00710000
00711000
00712000
00713000
00714000
00715000
00716000
00717000
00718000
00719000
00720000

```