

Verbetering van het convergentie-mechanisme in het optimaliseringsprogramma DYNOPT

Citation for published version (APA):

van der Sanden, M. G. E. M. (1988). *Verbetering van het convergentie-mechanisme in het optimaliseringsprogramma DYNOPT*. (DCT rapporten; Vol. 1988.069). Technische Universiteit Eindhoven.

Document status and date:

Gepubliceerd: 01/01/1988

Document Version:

Uitgevers PDF, ook bekend als Version of Record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Verbetering van het convergentie-mechanisme
in het optimaliseringsprogramma DYNOPT

verslag van de eerste stage bij de vakgroep
Fundamentele Werktuigbouwkunde
Afdeling der Werktuigbouwkunde
Technische Universiteit Eindhoven

september - november 1987

Rapport WFW 88.069

Uitvoering : Ing. M. G. E. M. van der Sanden

Begeleiding : Dr. Ir. A. J. G. Schoofs

Inhoudsopgave

Samenvatting

1. Inleiding en probleemstelling
2. Alternatieven ter verbetering van het stapverkleiningsmechanisme
 - 2.1 Staphalvering na niet afnemen v. objectfunctie én tekenwisseling v. afgeleide
 - 2.2 Iteratie t.o.v. momentaan optimale ontwerp punt
 - 2.3 Stapregulatie m.b.v. afgeleiden / objectfunctie waarden
 - 2.4 Stapregulatie bij elke tekenwisseling van de afgeleide
 - 2.5 Stapregulatie na twee of meer tekenwisselingen v. afgeleide
 - 2.6 Staphalvering na twee of meer tekenwisselingen v. afgeleide
 - 2.7 Stapverdubbeling na gelijk teken v. afgeleide in twee of meer opeenvolgende iteraties
 - 2.8 Staphalvering bij elke tekenwisseling v.d. afgeleide
3. Evaluatie der alternatieven
 - 3.1 Staphalvering zonder meer versus onder voorwaarde van tekenwisseling
 - 3.2 Iteratie t.o.v. het laatste ontwerp punt versus t.o.v. momentaan optimale ontwerp punt
 - 3.3 Criterium voor ingrijpen in stapgrootte
 - 3.4 Stapregulatie d.m.v. afgeleiden / objectfunctie waarden versus staphalvering (aangevuld met stapverdubbeling)
4. Vergelijking der resultaten v.h. gemodificeerde programma met die v.d. oorspronkelijke versie
5. Conclusies en aanbevelingen

Literatuur

Bijlagen

- 1 a+b Invoerfile OPTIN aangepast voor het 5 elementen-model
- 2 a+b Invoerfile OPTIN voor het 19 elementen-model
- 3 Invoerfile GEOIN
- 4 Iteratie-verloop v. objektfunctie vanuit startpunt \underline{x}_A
(5 elementen-model ; oorspronkelijk / alt. 1 / alt. 2)
- 5 Iteratie-verloop v. objektfunctie vanuit startpunt \underline{x}_A
(5 elementen-model ; oorspronkelijk / alt. 1 t/m alt. 9 ; exd. alt. 2)
- 6 a+b Iteratie-verloop v. objektfunctie vanuit startpunt \underline{x}_B
(5 elementen-model ; oorspronkelijk / alt. 1 t/m alt. 8 ; exd. alt. 2)
- 7 Iteratie-verloop v. objektfunctie vanuit startpunt \underline{x}_B
(19 elementen-model ; oorspronkelijk / alt. 9 / alt. 8 + 11 herstarts)
- 8 Iteratie-verloop v. ontwerpvariabelen, startpunt \underline{x}_A
(5 elementen-model ; oorspronkelijk)
- 9 t/m 16 Iteratie-verloop v. ontwerpvariabelen, startpunt \underline{x}_A
(5 elementen-model ; respectievelijk alt. 1 t/m alt. 9)
- 17 Iteratie-verloop v. ontwerpvariabelen, startpunt \underline{x}_B
(19 elementen-model ; alternatief 8)
- 18 a+b Objektfunctie-waarden bij het iteratie-verloop.
(5 elementen-model ; startpunt \underline{x}_A respectievelijk \underline{x}_B)
- 19 Objektfunctie-waarden bij het iteratie-verloop
(19 elementen-model ; startpunt \underline{x}_B)
- 20 Eouten in het oorspronkelijke programma van DYNOPT

Samenvatting

In het optimaliseringsprogramma DYNOPT, gebaseerd op sequentiële lineaire programmering, ontmoette men convergentie-problemen bij het naderen van een lokaal minimaal ontwerp punt. Deze werden veroorzaakt door het toegepaste stapverkleiningsmechanisme van halvering van move-limits, voor alle ontwerpvariabelen tegelijk.

In dit verslag worden enige alternatieven voor modificatie van het oorspronkelijke programma geëvalueerd op hun verbetering van het convergentie-gedrag. Wijzigingen in de vorm van iteratie t.o.v. het momentaan optimale ontwerp punt in plaats van iteratie t.o.v. het laatste ontwerp punt en selectie van ontwerpvariabelen met tekenwisseling der afgeleide naar objektfunctie, waarop stap halvering toe te gaan passen, in plaats van gewoonweg op alle ontwerpvariabelen, geven reeds enige verbetering te zien. Ook het criterium voor het in werking stellen van het stapverkleiningsmechanisme speelt een rol. De methode van optimaliseren van move-limits m.b.v. afgeleiden / objektfunctiewaarden komt door de wijze waarop deze in de bestaande programmatuur van DYNOPT geïmplementeerd werd, onvoldoende tot zijn recht.

De beste modificatie binnen de onderzochte groep alternatieven is de introductie van stapverdubbeling naast de reeds aanwezige stap halvering. De verbetering in convergentie die dit opleverde, kan echter niet anders dan teleurstellend genoemd worden, wat ons

doet uitriien naar stapregulatie gebaseerd op de
methode der geconjugeerde gradiënten of eventueel
het introduceren van kwadratische programmering
binnen DYNOPT.

1. Inleiding en probleemstelling

Iedere continue constructie heeft een oneindig groot aantal eigenfrequenties, waarvan meestal alleen die met de lagere waarden in de praktijk van belang zijn. Deze interessante eigenfrequenties, zeg m stuks, worden opgeslaan in de kolommatrix $\underline{\omega}$. Veronderstel dat een beschouwde constructie te beschrijven is met n ontwerpvariabelen, gerepresenteerd door kolommatrix \underline{x} , dan zouden we graag het verband kennen tussen deze eigenfrequenties en ontwerpvariabelen:

$$\underline{\omega} = \underline{\omega}(\underline{x}) \quad (1.1)$$

Met gebruikmaking van dit veelal niet-lineaire verband moet het volgende probleem worden opgelost: Bepaal de ontwerpvariabelen \underline{x} zodat:

$$\left. \begin{array}{l} \underline{\omega}(\underline{x}) = \underline{\omega}^{\circ} \\ \text{onder de voorwaarden} \\ \underline{g}(\underline{x}) \leq \underline{0} \\ \underline{h}(\underline{x}) = \underline{0} \end{array} \right\} (1.2)$$

Hierin stelt $\underline{\omega}^{\circ}$ de kolommatrix voor met de m stuks gewenste waarden voor de eigenfrequenties. De bevestigingen voor de ontwerpvariabelen (minimale afmetingen, bepaalde ontwerpvariabelen positief vereist etc.) liggen vast met het stelsel ongelijkheden $\underline{g}(\underline{x}) \leq \underline{0}$. Het stelsel gelijkheden $\underline{h}(\underline{x}) = \underline{0}$ vertegenwoordigt eenduidige verbanden tussen de ontwerpvariabelen onderling. Het stelsel met voorwaarden uit het beschouwde probleem, wordt verondersteld niet strijdig te zijn. Strijdigheid van het totale stelsel vergelijkingen en ongelijkheden (1.2) behoort echter tot de mogelijkheden. In dat geval

wordt getracht de gewenste eigenfrequenties zo dicht mogelijk te benaderen door het minimaliseren van de volgende objektfunctie :

$$F(\underline{x}) = \sum_{j=1}^m w_j^2 \cdot \frac{\{w_j(\underline{x}) - w_j^0\}^2}{\{w_j^0\}^2} \quad (1.3)$$

onder de voorwaarden

$$g(\underline{x}) \leq 0$$

$$h(\underline{x}) = 0$$

De weegfactor w_j maakt het mogelijk bepaalde frequenties te accentueren.

Het optimaliseringsprogramma DYNOPT maakt A. b. v. het oplossen van bovenvermeld probleem gebruik van sequentiële lineaire programmering. Objektfunctie en voorwaarden worden hierbij gelineariseerd rond het huidige ontwerp-punt \underline{x}_k . Het aldus ontstane lineaire optimaliseringsprobleem levert na oplossen een nieuw ontwerp-punt \underline{x}_{k+1} .

In dit nieuwe punt wordt weer het optimaliseringsprobleem gelineariseerd en opgelost. Beëindiging van dit iteratieve proces geschiedt nadat de objektfunctie een voldoende lage waarde bereikt heeft $F(\underline{x}_k) < TOLQUA$, onvoldoende meer daalt $(F(\underline{x}_k) - F(\underline{x}_{k-1})) / F(\underline{x}_k) < TOLDEL$ of na het doorlopen van een maximaal aantal toegestane iteraties.

Convergentie-criteria en maximum aantal iteraties worden opgegeven in de invoerfile (zie bijlagen 1 en 2). Linearisatie van de objektfunctie levert op :

$$F_k(\underline{x}) = F(\underline{x}_k) + \sum_{i=1}^n (x_i - x_{ik}) \cdot \frac{\partial F(\underline{x}_k)}{\partial x_i}$$

$$= \sum_{j=1}^m w_j^2 \cdot \frac{\{w_j(\underline{x}_k) - w_j^0\}^2}{\{w_j^0\}^2} + 2 \cdot \sum_{i=1}^n (x_i - x_{ik}) \cdot \sum_{j=1}^m w_j^2 \cdot \frac{\{w_j(\underline{x}_k) - w_j^0\}}{\{w_j^0\}^2} \cdot \frac{\partial w_j(\underline{x}_k)}{\partial x_i}$$

Met weglating van constante termen in deze uitdrukking, die bij een lineair programmeringsprobleem immers geen invloed

hebben op de oplossing, vinden we nu:

Minimaliseer:

$$F_k(\underline{x}) = 2 \sum_{i=1}^n x_i \cdot \sum_{j=1}^m w_j^2 \cdot \frac{\{w_j(\underline{x}_k) - w_j^0\}}{\{w_j^0\}^2} \cdot \frac{\partial w_j(\underline{x}_k)}{\partial x_i}$$

onder de voorwaarden

$$g_k(\underline{x}) = g(\underline{x}_k) + \sum_{i=1}^n (x_i - x_{i,k}) \cdot \frac{\partial g(\underline{x}_k)}{\partial x_i} \leq 0$$

$$h_k(\underline{x}) = h(\underline{x}_k) + \sum_{i=1}^n (x_i - x_{i,k}) \cdot \frac{\partial h(\underline{x}_k)}{\partial x_i} = 0$$

(1.4)

De partiële afgeleiden van de (on)gelijkheidsbeperkingen naar de ontwerpvariabelen worden benaderd door differentiequotienten: $\frac{\partial g}{\partial x_i} \approx \frac{\Delta g}{\Delta x_i}$; $\frac{\partial h}{\partial x_i} \approx \frac{\Delta h}{\Delta x_i}$, $i=1, \dots, n$ (1.5)

Eigenfrequenties en eigenvectoren worden bepaald door oplossen van het volgende eigenwaarde-probleem, dat wordt verkregen door verdeling van de constructie in eindige elementen en partiële discretisatie van het verplaatsingsveld:

$$\{ \underline{K}(\underline{x}_k) - \omega_j^2 \cdot \underline{M}(\underline{x}_k) \} \cdot \underline{u}_j = 0 \quad (1.6)$$

waarin

$\underline{K}(\underline{x}_k)$ = symmetrische stijfheidsmatrix in ontwerppunt \underline{x}_k

$\underline{M}(\underline{x}_k)$ = " " massamatrix " "

ω_j = eigenfrequentie [rad/s]

\underline{u}_j = bij ω_j behorende eigenvector

Voor afleiding van (1.6) wordt verwezen naar lit. [1] hoofdstuk 3. In hoofdstuk 2 van dezelfde literatuuropgave wordt aangetoond dat:

$$\frac{\partial \omega_j}{\partial x_i} = \frac{\underline{u}_j^T \cdot \left\{ \frac{\partial \underline{K}}{\partial x_i} - \omega_j^2 \cdot \frac{\partial \underline{M}}{\partial x_i} \right\} \cdot \underline{u}_j}{2 \omega_j \underline{u}_j^T \cdot \underline{M} \cdot \underline{u}_j} \quad (1.7)$$

Partiële afgeleiden van massamatrix en stijfheidsmatrix naar de ontwerpvariabelen worden ook nu weer benaderd

door differentie-quotienten :

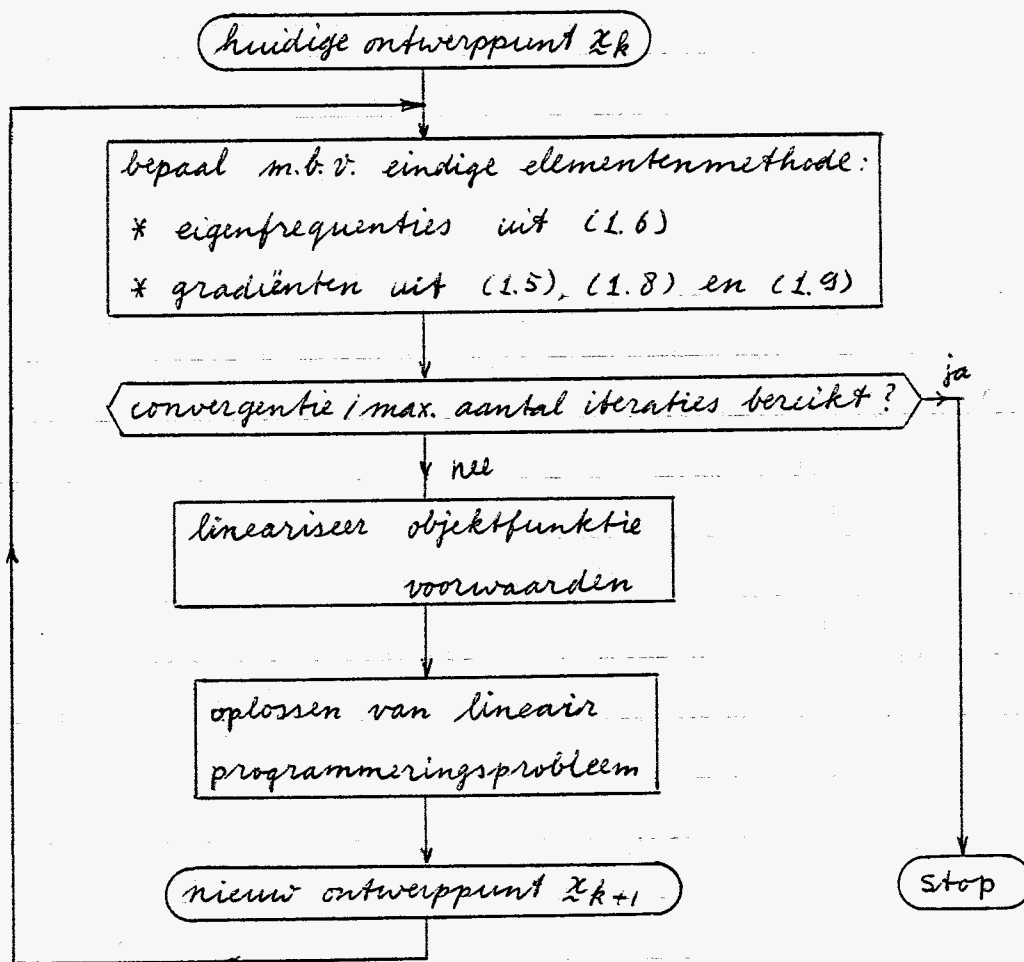
$$\frac{\partial K}{\partial x_i} \approx \frac{\Delta K}{\Delta x_i} \quad ; \quad \frac{\partial M}{\partial x_i} \approx \frac{\Delta M}{\Delta x_i} \quad , \quad i = 1, \dots, n \quad (1.8)$$

Dan ontstaat :

$$\frac{\partial w_j}{\partial x_i} \approx \frac{u_j^T \cdot \{ \Delta K - w_j^2 \cdot \Delta M \} \cdot u_j}{2 w_j \Delta x_i \cdot u_j^T \cdot M \cdot u_j} \quad \dots \quad j = 1, \dots, m \quad (1.9)$$

$i = 1, \dots, n$

In schema ziet de iteratie-cyclus binnen DYNOPT eruit zoals weergegeven in figuur 1.1.



Figuur 1.1 : Iteratie-cyclus van DYNOPT in schema

Wat de verdeling in elementen van de constructie aangaat, heeft DYNOPT de volgende beperkingen :

- Alleen axiaal symmetrische constructies kunnen worden

geanalyseerd

- Toe te passen elementtypen zijn:

1) axiaal symmetrisch, 6 knoops, isoparametrisch, periodiek element: TRIAX 6

2) axiaal symmetrisch, 8 knoops, isoparametrisch, periodiek element: QUAX 8

- Constant aantal elementen gedurende de uitvoering van het programma

- Topologie ligt vast zoals opgegeven in de invoerfile (zie bijv. bijlage 1 of bijlage 2 onder: * CONNECTIVITY) en is niet te wijzigen gedurende de uitvoering van het programma.

De term sequentiëel in de omschrijving van DYNOPT, als zijn de een optimaliseringsprogramma op basis van sequentiële lineaire programmering, houdt in dat per iteratieslag aan de ontwerpvariabelen slechts een variatie binnen op te geven grenzen rond het huidige ontwerp punt \underline{x}_k wordt toegestaan:

$$x_{ik} - |\Delta x_i| \leq x_i \leq x_{ik} + |\Delta x_i|, \quad i = 1, \dots, n \quad (1.10)$$

Deze ongelijkheidsbegrenzungen vormen een deel van het stelsel ongelijkheden $g_k(\underline{x}) \leq \underline{q}$ in (1.4). Invoering van deze maximale verandering Δx_i per iteratieslag, de zgn. move-limit, voorkomt dat de waarden van de ontwerpvariabelen heen en weer springen van de ene uiterste grens van het toegestane gebied naar de andere. Oscilleren tussen de nauwere grenzen gesteld door de move-limits is ook nu nog mogelijk, maar het berekende optimum van het gelineariseerde probleem ligt nu dichterbij het werkelijke optimale punt. Een en ander wordt verduidelijkt in voorbeeld 1.

Voorbeeld 1:

$$\text{Minimaliseer } F(x) = \frac{1}{2}x^2 - 3x + 5$$

onder de voorwaarden

$$x \geq 1 \quad ; \quad x \leq 6$$

$$x_k - 0,6 \leq x \leq x_k + 0,6$$

lineariseren rond x_k : $F_k(x) = F(x_k) + (x-x_k) \cdot \frac{\partial F}{\partial x}(x_k)$

$$= \frac{1}{2} x_k^2 - 3x_k + 5 + (x-x_k) \cdot (x_k-3)$$

$$= -\frac{1}{2} x_k^2 + 5 + (x_k-3) \cdot x$$

Het gelineariseerd probleem luidt nu:

Minimaliseer $F_k(x) = -\frac{1}{2} x_k^2 + 5 + (x_k-3) \cdot x$

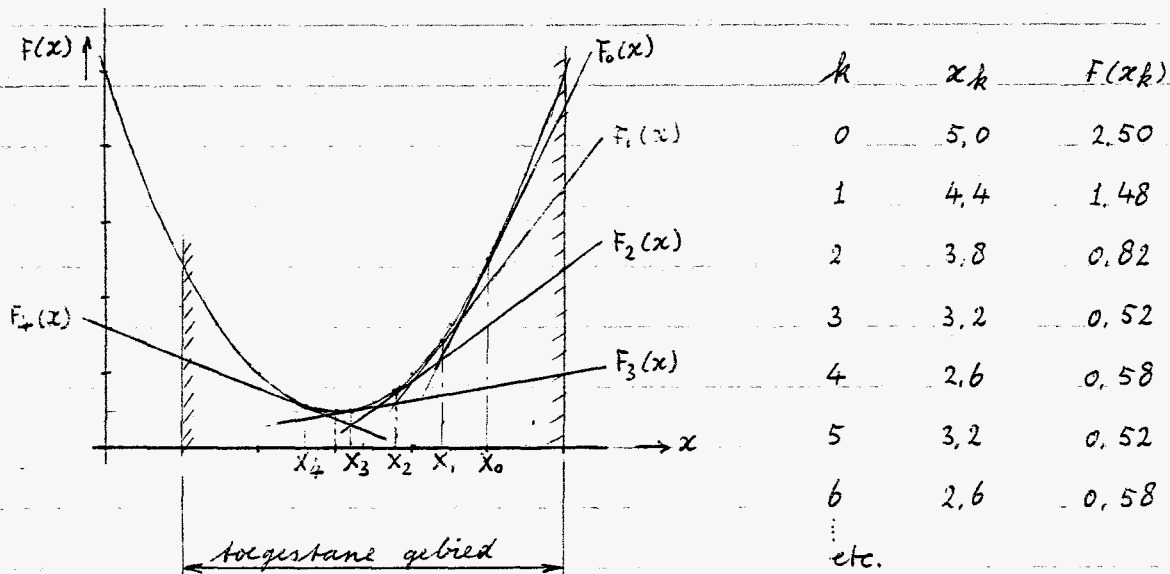
onder de voorwaarden:

$$x \geq \max(1, x_k - 0,6)$$

$$x \leq \min(6, x_k + 0,6)$$

Met k als iteratie nummer wordt x_k gezien als optimale oplossing van het laatst gelineariseerde optimaliseringsprobleem.

Figuur 1.2 toont het iteratie verloop.



Figuur 1.2 : Iteratie verloop bij sequentiële lineaire programmering.

Het optimale punt $x=3$, $F(3) = 0,5$ wordt niet meer dichterbij benaderd na iteratie 3, omdat vanaf dat moment oscillatie tussen de punten $x=2,6$ en $x=3,2$ optreedt.

Tegen dit heen en weer springen van de oplossing werd in de oorspronkelijke versie van DYNOPT niets gedaan in de

vorm van halvering van de move-limit, zodra de waarde van de objektfunctie was toegenomen t.o.v. diens waarde in de voorafgaande iteratie. Bij een functie van meerdere variabelen geschiedde deze staphalvering voor alle ontwerpvariabelen tegelijk. In lit [1] wordt melding gemaakt van zeer trage convergentie bij de problemen die men met DYNOPT tot een oplossing trachtte te brengen, als gevolg van het op deze wijze toepassen van stapverkleining. Het optimale punt (steeds een lokaal minimum, inherent aan de methode van sequentiële lineaire programmering) kon vaak niet meer bereikt worden binnen het toegestane aantal iteraties.

Dit is verklaarbaar als men bedenkt dat in een functie van meerdere variabelen, de richtingen met sterke kromming (grote tweede afgeleide) aanleiding geven tot halvering van de stapgrootte, terwijl dit laatste remmend werkt op benadering van het optimum in richtingen met minder sterke kromming. In voorbeeld 2 wordt dit duidelijk gemaakt.

Voorbeeld 2:

Minimaliseer $F(\underline{x}) = x_1^2 - 8x_1 + 144x_2^2 - 864x_2 + 1312$

onder de voorwaarden

$$x_1 \geq 0 \quad x_{1k} - 0.6 \leq x_1 \leq x_{1k} + 0.6$$

$$x_2 \geq 0 \quad x_{2k} - 0.6 \leq x_2 \leq x_{2k} + 0.6$$

lineariseren rond \underline{x}_k :

$$\begin{aligned} F_k(\underline{x}) &= F(\underline{x}_k) + \sum_{i=1}^2 (x_i - x_{ik}) \cdot \frac{\partial F}{\partial x_i}(\underline{x}_k) \\ &= x_{1k}^2 - 8x_{1k} + 144x_{2k}^2 - 864x_{2k} + 1312 + (x_1 - x_{1k}) \cdot (2x_{1k} - 8) + \\ &\quad (x_2 - x_{2k}) \cdot (288x_{2k} - 864) \end{aligned}$$

Het gelineariseerde probleem wordt dan:

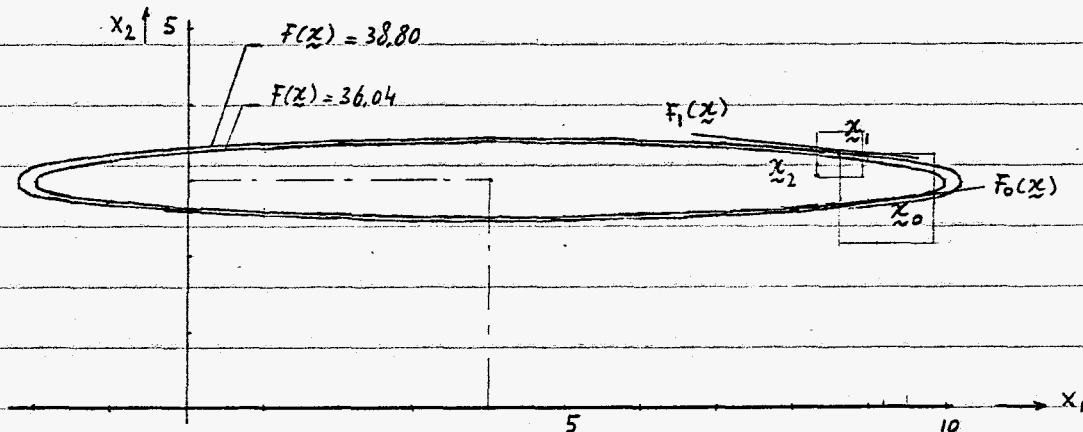
$$\text{Minimaliseer } F_k(\underline{x}) = -x_{1k}^2 - 144x_{2k}^2 + 1312 + (2x_{1k} - 8) \cdot x_1 + (288x_{2k} - 864) \cdot x_2$$

onder de voorwaarden:

$$x_1 \geq \max(0, x_{1,k} - 0,6) \quad x_1 \leq x_{1,k} + 0,6$$

$$x_2 \geq \max(0, x_{2,k} - 0,6) \quad x_2 \leq x_{2,k} + 0,6$$

Figuur 1.3 toont een deel van het iteratie-verloop.



k	x_k	$F_k(x)$	$F(x_k)$
0	(9,2 ; 2,75)	$10,4 \cdot x_1 - 72 \cdot x_2 + 138,36$	36,04
1	(8,6 ; 3,35)	$9,2 \cdot x_1 - 100,8 \cdot x_2 - 378$	38,80 : $F(x_1) > F(x_0)$
2	(8,3 ; 3,05)		18,85
...			
etc.			

halvering v. move-limits ←

Figuur 1.3 : Begin van het iteratie-verloop bij sequentiële lineaire programmering voor functie van 2 variabelen.

Door de stapverkleining in de x_1 -richting zijn meer iteraties nodig voor het bereiken van het optimum.

De vraag is nu: Hoe moet het stapverkleiningsmechanisme worden aangepast, opdat zeer langzame convergentie wordt voorkomen?

2. Alternatieven ter verbetering van het stapverkleiningsmechanisme

In dit hoofdstuk worden een aantal versies voor modificatie van het oorspronkelijke programma DYNOPT geformuleerd, waarvan verwacht wordt dat ze de convergentie-snelheid verhogen. Of dat ook inderdaad het geval is, werd steeds onderzocht door het programma te laten draaien met als invoerfile "Russische klok". De resultaten komen ter sprake in hoofdstuk 3. Nu worden eerst de belangrijkste kenmerken van elk alternatief vermeld en de wijzigingen ervan t.o.v. het oorspronkelijke programma of t.o.v. een zeer ermee verwant alternatief. Op bijlage 20 zijn de fouten vermeld die in de oorspronkelijke versie van DYNOPT aanwerpig waren. Vermelding van wijzigingen in DYNOPT.COMMON is in de volgende paragrafen achterwege gelaten omdat deze triviaal zijn: toevoegen van alle nieuw geïntroduceerde variabelen en arrays in de juiste COMMON blocks.

2.1. Staphalvering na niet afnemen v. objectfunctie en tekenwisseling v. afgeleide

Kenmerken van alternatief 1:

- Iteratie t.o.v. het laatst bepaalde ontwerp punt.
- Staphalvering indien de objectfunctie niet in waarde is gedaald t.o.v. de vorige iteratie: $F(x_k) \geq F(x_{k-1})$ maar dit alleen in die richtingen waar tekenwisseling van de afgeleide optreedt of deze nul is:
 $(\partial F / \partial x_i) | x_k * (\partial F / \partial x_i) | x_{k-1} \leq 0, i = 1, \dots, n$
- In overige richtingen blijft move-limit gelijk aan diens waarde in de vorige iteratie.

Wijzigingen in het oorspronkelijke programma t.b.v. alternatief 1:

SUBROUTINE TUNING

INTEGER INTG(1), LREA, LINT, INDEX

⋮

} ongewijzigd

\$INSERT DYNOPT. COMMON

INTEGER I

JITER = 0

DO 50 I = 1, NVAR

REA(KSTEP-1+I) = REA(KVARDE-1+I)

50 CONTINUE

100 CONTINUE

⋮

CALL DERIVA →

SUBROUTINE DERIVA

INTEGER INTG(1)

⋮

IF(NVAR .EQ. 0) GOTO 200

CALL COPYRV(REA(KDFMQL), REA(KDFMDV), NVAR)

DO 150 IVAR = 1, NVAR

⋮

150 CONTINUE

⋮

200 RETURN

END

IF(LEND) GOTO 200

CALL MATSET →

SUBROUTINE MATSET

INTEGER INTG(1)

⋮

REAL*8 ABSTEP, VARLOW, VARUP

IF(JITER .EQ. 0) GOTO 30

IF(FMIN .LT. FMINCL) GOTO 30

DO 20 I = 1, NVAR


```
IF (REA(KDFMDV-1+I) * REA(KDFMCL-1+I) .GT. Q. QDQ) GOTO 20
REA(KSTEP-1+I) = REA(KSTEP-1+I) / 2. Q DQ
20 CONTINUE
30 CONTINUE
IF (NCON .EQ. Q) GOTO 50
CONTINUE
50 DO 60 I = 1, NVAR
ABSTEP = DABS (REA (KSTEP-1+I))
VARLOW = REA (KVAR-1+I) - ABSTEP
CONTINUE
60 CONTINUE
RETURN
END
```

CALL MATSOL → *ongewijzigd*

CALL TIMEOUT

JITER = JITER + 1

GOTO 100

200 CONTINUE

CALL DATOUT (REAL), INTG (1), (2)

RETURN

END

SUBROUTINE INITI 1

INTEGER LREA, LINT

KDFMDV = KDFDN + NOPTFR * NELSEC

KDFMCL = KDFMDV + NVAR

KCONST = KDFMCL + NVAR

KFREL = KYTUN + NSECTO * NTUNFR

```
KSTEP = KREL + NOPTER  
KLASTR = KSTEP + NVAR  
:  
:  
RETURN  
END
```

Verwacht wordt dat dit alternatief een sneller dalend verloop van de objektfunctie te zien zal geven dan de oorspronkelijke versie van DYNOPT, vanaf het ontwerppunt waarin de objektfunctie voor de eerste keer in waarde is toegenomen t.o.v. de voorafgaande iteratie. Ontwerprichtingen zonder tekenwisseling van de afgeleide behouden immers hun oorspronkelijke (niet gehalveerde) waarde van de move-limit. In die richtingen moet het optimum sneller bereikt kunnen worden.

2.2. Iteratie t.o.v. momentaan optimale ontwerppunt

Kenmerken van alternatief 2:

- Iteratie t.o.v. het momentaan optimale ontwerppunt
- Staphalvering indien $F(x_k) \geq F(x_{k-1})$ en dan alleen in die richtingen waarvoor $(\partial F / \partial x_i) | x_k * (\partial F / \partial x_i) | x_{k-1} \leq 0, i=1, \dots, n$
- In overige richtingen wordt de waarde van de move-limit uit de vorige iteratie gehandhaafd.

Wijzigingen in het oorspronkelijke programma t.o.v. alternatief 2:

```
SUBROUTINE TUNING  
INTEGER INTG(1), LREA, LINT, INDEX  
:  
:  
$ INSERT DYNOPT.COMMON  
INTEGER I
```

JITER = 0

DO 50 I = 1, NVAR

REACKSTEP - I + I) = REACKVARDE - I + I)

50 CONTINUE

100 CONTINUE

CALL DATOUT

IF (NCON .GT. 0) CALL COPYRV (REACKCONDL), REACKCONST), NCON)

CALL USER

CALL DERIVA →

SUBROUTINE DERIVA

INTEGER INTG(1)

\$INSERT DYNOPT. COMMON

INTEGER IVAR, IFREQ, NOMD, NOMOLD, LENGTH, NUMB

IF (NVAR .EQ. 0) GOTO 200

CALL COPYRV (REACKDFMDL), REACKDFMDV), NVAR)

NUMB = NVAR * NCON

IF (NCON .GT. 0) CALL COPYRV (REACKDCOOL), REACKDCODV), NUMB)

DO 150 IVAR = 1, NVAR

150 CONTINUE

200 RETURN

END

IF (LEND) GOTO 200

CALL MATSET →

SUBROUTINE MATSET

INTEGER INTG(1)

\$INSERT DYNOPT. COMMON

INTEGER I, IND, NUMB

```
REAL*8 ABSTEP, VARLOW, VARUP
IF (FMIN .LT. FMINOL) GOTO 30
DO 20 I=1, NVAR
IF (REA(KDFMDV-1+I)*REA(KDFMCL-1+I) .GT. 0.0D0) GOTO 20
REA(KSTEP-1+I) = REA(KSTEP-1+I) / 2.0D0
20 CONTINUE
FMIN = FMINOL
CALL COPYRV (REA(KVAR), REA(KVAROL), NVAR)
IF (NCON .GT. 0) CALL COPYRV (REA(KCONST), REA(KCONOL), NCON)
CALL COPYRV (REA(KDFMDV), REA(KDEMOL), NVAR)
NUMB = NCON * NVAR
IF (NCONV .GT. 0) CALL COPYRV (REA(KDCOV), REA(KDCOOL), NUMB)
30 CONTINUE
IF (NCON .EQ. 0) GOTO 50
:
50 CONTINUE
DO 60 I=1, NVAR
ABSTEP = DABS (REA(KSTEP-1+I))
VARLOW = REA(KVAR-1+I) - ABSTEP
:
60 CONTINUE
RETURN
END
CALL MATSOL → SUBROUTINE MATSOL
INTEGER INTG(1)
:
LW = 9 * NVAR + 7 * NCON
CALL COPYRV (REA(KVAROL), REA(KVAR), NVAR)
IBOUND = 0
:
RETURN
END
```

```
CALL TIMEOUT
JITER = JITER + 1
GOTO 100
200 CONTINUE
:
CALL DATOUT (REAC1, INTGC1, 12)
RETURN
END

SUBROUTINE INITI 1
INTEGER LREA, LINT
:
KVAR = KCOOR + MAXCOR * NP
KVAROL = KVAR + NVAR
KVARMI = KVAROL + NVAR
:
KDFMDV = KDFDN + NPTFR * NELSEC
KDFMOL = KDFMDV + NVAR
KCONST = KDFMOL + NVAR
KCONOL = KCONST + NCON
KDCODV = KCONOL + NCON
KDCOOL = KDCODV + NCON * NVAR + NVAR
KLOWBO = KDCOOL + NCON * NVAR + NVAR
:
KREL = KYTUN + NSECTO * NTUNER
KSTEP = KREL + NPTFR
KLASTR = KSTEP + NVAR
:
RETURN
END
```

Zodra de waarde van de objectfunctie is toegenomen,

worden bij dit alternatief de waarden van objektfunctie, ontwerpvariabelen, begrenzingen (mits aanwezig) en gradiënten uit de vorige iteratie weer in de daarvoor bestemde arrays geplaatst. Iteratie vindt steeds plaats t.o.v. het ontwerp-punt met op dat moment de laagste waarde voor de objektfunctie. Het lijkt aannemelijk dat zodoende een snellere daling van de objektfunctie verkregen kan worden dan in de oorspronkelijke situatie.

2.3 Stapregulatie m.b.v. afgeleiden / objektfunctie-waarden

Kenmerken van alternatief 3:

- Iteratie t.o.v. het momentaan optimale ontwerp-punt
- Stapgrootte gestuurd door afgeleiden naar en funktiewaarden van de objektfunctie indien $F(x_k) \geq F(x_{k-1})$ alleen in die richtingen waarvoor $(\partial F / \partial x_i) | x_k * (\partial F / \partial x_i) | x_{k-1} \leq 0, i=1, \dots, n$
- In overige richtingen wordt de standaard (maximale) waarde van de move-limit gehanteerd.

Wijzigingen in het oorspronkelijk programma t.b.v. alternatief 3:

SUBROUTINE TUNING

INTEGER INTG(1), LREA, LINT, INDEX

⋮

CALL DATOUT

IF (NCON .GT. 0) CALL COPYRV (REACKCONCL, REACKCONST), NCON)

CALL USER

⋮

CALL DERIVA → SUBROUTINE DERIVA

INTEGER INTG(1)

⋮

\$INSERT DYNOPT.COMMON

```
INTEGER IVAR, IFREQ, NOMO, NOMOLD, LENGTH, NUMB
:
IF (NVAR .EQ. 0) GOTO 200
CALL COPYRV (REA(KDFMOL), REA(KDFMDV), NVAR)
NUMB = NVAR * NCON
IF (NCON .GT. 0) CALL COPYRV (REA(KDCGL), REA(KDCDV), NUMB)
DO 150 IVAR = 1, NVAR
:
150 CONTINUE
:
200 RETURN
END
```

IF (LEND) GOTO 200

CALL MATSET →

```
SUBROUTINE MATSET
INTEGER INTG(1)
REAL*8 REA(1), RSTEP
:
$INSERT DYNOPT COMMON
INTEGER I, IND, NUMB
REAL*8 ABSTEP, VARLOW, VARUP
DO 10 I = 1, NVAR
  REA(KSTEP-1+I) = REA(KVARDE-1+I)
10 CONTINUE
IF (JITER .EQ. 0) GOTO 22
IF (CFMIN .LT. FMINCL) GOTO 22
DO 20 I = 1, NVAR
  IF (REA(KDFMDV-1+I) * REA(KDFMOL-1+I) .GT. 0.0D0) GOTO 20
  CALL STEPRG (RSTEP, REA(I), INTG(I), I)
  REA(KSTEP-1+I) = RSTEP
20 CONTINUE
FMIN = FMINCL
```

```
CALL COPYRV (REA(KVAR), REA(KVAROL), NVAR)
IF (NCON .GT. 0) CALL COPYRV (REA(KCONST), REA(KCONOL), NCON)
CALL COPYRV (REA(KDEMDV), REA(KDEMDL), NVAR)
NUMB = NVAR * NCON
IF (NCON .GT. 0) CALL COPYRV (REA(KDCQDV), REA(KDCQCL), NUMB)
22 CONTINUE
IF (NCON .EQ. 0) GOTO 50
  ⋮
50 CONTINUE
DO 60 I = 1, NAR
  ABSTEP = DABS (REA (KSTEP - 1 + I))
  VARLOW = REA (KVAR - 1 + I) - ABSTEP
  ⋮
60 CONTINUE
RETURN
END
```

CALL MATSOL → Zelfde als bij alternatief 2

JITER = JITER + 1

GOTO 100

200 CONTINUE

RETURN

END

SUBROUTINE INIT1 → Zelfde als bij alternatief 2

SUBROUTINE STEPFG (RSTEP, REA, INTEG, I)

INTEGER INTEG (1), I

REAL * 8 RSTEP, REA (1)

C Calculates distance between XOPT and component X(I) of current
C iteration point. The point where the minimum value of the cubic
C function $F(x) = a + b \cdot x + c \cdot x \cdot x + d \cdot x \cdot x \cdot x$, fitted through (X1, F1)

C and (X2, F2) with respectively belonging derivatives DFDX1 and
C DFDX2 occurs, is represented by XOPT.

C

C Parameters

C I - I - Integer : index

C REA - I - Real * 8 : array of DIMENSION(I), used as memory

C RSTEP - 0 - Real * 8 : distance between XOPT and component X(I)

C of the momentary optimum iteration point

\$ INSERT DYNOPT, COMMON

REAL * 8 DEFDEX, DENOM, DKFDXK, DEX, FACT, RNUME, ROOT, SUMDER, SUMVAR, XOPT

C Local variables

C DEX = X2 - X1

C DEFDEX = (F2 - F1) / DEX

C DENOM = denominator of formula (2.3.5)

C DKFDXK = second derivative of cubic function

C FACT = factor appearing in numerator of formula (2.3.5)

C RNUME = numerator of formula (2.3.5)

C ROOT = factor appearing under square root in numerator of (2.3.5)

C SUMDER = DFDX1 + DFDX2

C SUMVAR = X1 + X2

C XOPT = point where minimum value of cubic function occurs

C

DEX = REA(KVAR-1+I) - REA(KVARCL-1+I) → x₂ - x₁

DEFDEX = (FMIN - FMINCL) / DEX → (F₂ - F₁) / (x₂ - x₁)

SUMDER = REA(KDFMCL-1+I) + REA(KDFMDV-1+I) → F₁' + F₂'

SUMVAR = REA(KVARCL-1+I) + REA(KVAR-1+I)

ROOT = REA(KDFMCL-1+I) * SUMDER + REA(KDFMDV-1+I) * REA(KDFMDV-1+I) +
F₁' * (F₁' + F₂') + F₂' * F₂'

& -6.0 DΦ * SUMDER * DEFDEX + 9.0 DΦ * DEFDEX * DEFDEX

IF (ROOT .LT. 0.0 DΦ) GOTO 10

DENOM = 3.0 DΦ * (SUMDER - 2.0 DΦ * DEFDEX)

IF (DENOM .EQ. 0.0 DΦ) GOTO 10

$$\text{FACT} = \frac{\left(\begin{matrix} x_1 & + & 2 & \cdot & x_2 \end{matrix} \right) \cdot F_1'}{\left(\text{REA}(\text{KVAROL}-1+I) + 2 \cdot \psi \cdot D\psi \cdot \text{REA}(\text{KVAR}-1+I) \right) \cdot \text{REA}(\text{KDFMCI}-1+I) + \left(\begin{matrix} x_2 & + & 2 & \cdot & x_1 \end{matrix} \right) \cdot F_2'} + \frac{F_2'}{\left(\text{REA}(\text{KVAR}-1+I) + 2 \cdot \psi \cdot D\psi \cdot \text{REA}(\text{KVAROL}-1+I) \right) \cdot \text{REA}(\text{KDFMDV}-1+I)}$$

$$\text{RNUME} = \text{FACT} - 3 \cdot \psi \cdot D\psi \cdot \text{SUMVAR} \cdot \text{DEFDEX} + \text{DABS}(\text{DEX}) \cdot \text{DSQRT}(\text{RCOCT})$$

$$\text{XOPT} = \text{RNUME} / \text{DENOM}$$

C' check on XOPT being a minimum

$$\text{DKFDXK} = (-2 \cdot \psi \cdot D\psi \cdot \text{FACT} + 6 \cdot \psi \cdot D\psi \cdot \text{SUMVAR} \cdot \text{DEFDEX} +$$

$$+ 6 \cdot \psi \cdot D\psi \cdot (\text{SUMVAR} - 2 \cdot \psi \cdot D\psi \cdot \text{DEFDEX}) \cdot \text{XOPT}) / (\text{DEX} \cdot \text{DEX})$$

IF (DKFDXK .LE. $\psi \cdot D\psi$) GOTO 10

$$\text{RSTEP} = \text{XOPT} - \text{REA}(\text{KVAROL}-1+I)$$

RETURN

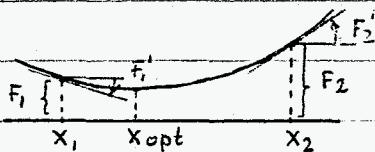
10 CONTINUE

RSTEP = DEX / 2 $\cdot \psi \cdot D\psi$ → als er iets fout gaat in de bere-

RETURN kening van XOPT, dan wordt

END staphalvering toegepast.

De methode van stapregulatie door afgeleiden naar en funktiewaarden van de objektfunctie is opgeret met de gedachte dat er een snellere manier moet zijn dan staphalvering om in een bepaalde ontwerprichting de coördinaat te vinden, tussen de betreffende componenten van twee opvolgende iteratiepunten, die de plaats van het gezochte (locale) minimum in die richting aangeeft. Benadering van de objektfunctie door een derdegraads polynoom maakt exacte bepaling van deze plaats mogelijk (zie figuur 2.3.2)



$$F(x_i) = A + B \cdot x + C \cdot x^2 + D \cdot x^3$$

→ ontwerprichting i

x_1 : component x_i van ontwerppunt x_{k-1}

x_2 : " " " " " x_k

$$F_1 = F(x_{k-1}); F_1' = (\partial F / \partial x_i) |_{x_{k-1}}; F_2 = F(x_k); F_2' = (\partial F / \partial x_i) |_{x_k}$$

Figuur 2.3.2: Benadering v. objektfunctie door derdegraads polynoom.

De coëfficiënten van de polynoom worden bepaald uit het volgende stelsel vergelijkingen:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 0 & 1 & 2x_1 & 3x_1^2 \\ 0 & 1 & 2x_2 & 3x_2^2 \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_1' \\ F_2' \end{bmatrix} \quad (2.3.1)$$

Op de plaats van het minimum geldt:

$$1) \partial F / \partial x_i = B + 2Cx + 3Dx^2 = 0 \quad (2.3.2)$$

$$\text{waarmee: } x_{\text{opt}} = \frac{-C \pm \sqrt{C^2 - 3BD}}{3D} \quad (2.3.3)$$

$$\text{en } 2) \partial^2 F / \partial x_i^2 = 2C + 6Dx > 0 \quad (2.3.4)$$

Invullen van (2.3.3) in (2.3.4) toont dat alleen het positieve teken voor het wortelteken voldoet. De coëfficiënten ingevuld in (2.3.3) levert dan als uitdrukking voor de plaats van het minimum:

$$x_{\text{opt}} = \frac{(x_1 + 2x_2) \cdot F_1' + (x_2 + 2x_1) \cdot F_2' - 3(x_1 + x_2) \cdot \frac{(F_2 - F_1)}{(x_2 - x_1)} + |x_2 - x_1| \cdot \sqrt{\frac{(F_1')^2 + F_1' \cdot F_2' + (F_2')^2 - 6(F_1' + F_2') \cdot \frac{(F_2 - F_1)}{(x_2 - x_1)} + 9 \frac{(F_2 - F_1)^2}{(x_2 - x_1)^2}}}{3 \cdot (F_1' + F_2') - 6 \cdot \frac{(F_2 - F_1)}{(x_2 - x_1)}} \quad (2.3.5)$$

Verwisseling van alle gegevens uit punt 1 met de waarden uit punt 2 blijkt geen invloed te hebben op x_{opt} (N.B. absoluut strepen van de term voor het wortelteken niet vergeten!).

Voorzieningen om een van de twee punten steeds te laten corresponderen met het vorige iteratie-punt x_k , en het andere met het huidige x_k , zijn dus niet nodig.

De stapgrootte voor een bepaalde ontwerpvariabele die in subroutine STEPRG wordt bepaald, is in dit alternatief steeds de (absolute) waarde van het verschil tussen x_{opt} en de betreffende component van het vorige iteratie-punt. Aanroep van de subroutine geschiedt alleen dan wanneer de objektfunctie in waarde is toegenomen en het vorige ite-

ratiepunt dus momentaan optimaal is (zie kenmerk: Iteratie A.o.v. momentaan optimaal ontwerppunt).

Bij de evaluatie van de alternatieven in hoofdstuk 3 wordt duidelijk dat dit alternatief in het geheel niet aan de verwachtingen heeft voldaan en om welke reden.

2.4 Stapregulatie bij elke tekenwisseling van de afgeleide

Kenmerken van alternatief 4:

- Iteratie A.o.v. momentaan optimale ontwerppunt
- Stapgrootte gestuurd door afgeleiden naar en funktiewaarden van de objektfunctie zodra: $(\partial F / \partial x_i) | x_k \neq (\partial F / \partial x_i) | x_{k-1} \leq 0, i = 1, \dots, n$
De eis dat $F(x_k) \geq F(x_{k-1})$ moet zijn is dus vervallen!
- In overige richtingen van ontwerpvariabelen wordt de standaard (maximale) stapgrootte gehanteerd.

Wijzigingen A.o.v. alternatief 3 A. b. v. alternatief 4:

```
SUBROUTINE MATSET
  INTEGER INTG(1)
  .
  .
  IF (JITER .EQ. 0) GOTO 22
  - - - - - regel verwijderen.
  DO 20 I = 1, NVAR
  .
  20 CONTINUE
  22 CONTINUE
  IF (FMIN .LT. FMINOL) GOTO 30 hiernaartoe verplaatst.
  FMIN = FMINOL
  .
  30 CONTINUE
  .
  .
```

RETURN

END

SUBROUTINE INITI1 → ongewijzigd, dus ook weer gelijk aan
uitvoering bij alternatief 2

SUBROUTINE STEPRG

INTEGER INTG(LI), I

IF (CHKFDXK LE. 0.0D0) GOTO 10

IF (FMIN.LT.FMINOL) GOTO 5

RSTEP = XOPT - REA(KVAROL - 1 + I) vorige ontwerp punt momen-
taan optimaal

RETURN

5 CONTINUE

RSTEP = XOPT - REA(KVAR - 1 + I) huidige ontwerp punt momen-
taan optimaal.

RETURN

10 CONTINUE

RSTEP = DEX / 2.0D0

RETURN

END

Op het moment van aanroep van subroutine STEPRG is nog niet duidelijk of het vorige of het huidige ontwerp punt optimaal is. Dat wordt dus in STEPRG zelf bepaald in de regel: IF (FMIN.LT.FMINOL) GOTO 5

Tot het laten vallen van de eis $F(\underline{x}_k) \geq F(\underline{x}_{k-1})$ alvorens in te grijpen in de waarde van de stapgrootte is besloten, omdat het iteratie-verloop van de ontwerpvariabelen bij de tot dusver besproken alternatieven oscillaties vertoonden, ruim voordat de objektfunctie voor het eerst toenam (zie bijlagen 9 t/m 11). Hiermee is een vertragend effect op het iteratie-verloop van de objektfunctie verdwenen.

2.5. Stapregulatie na twee of meer tekenwisselingen v. afgeleide

kenmerken van alternatief 5:

- Iteratie t.o.v. momentaan optimale ontwerpbeurt
- Stapgrootte gestuurd door afgeleiden naar en funktiewaarden van de objektfunctie zodra in twee of meer (niet noodzakelijk opvolgende) iteraties is opgetreden:
 $(\partial F / \partial x_i) | x_k * (\partial F / \partial x_i) | x_{k-1} \leq 0, i=1, \dots, n$
De eis $F(x_k) \geq F(x_{k-1})$ is ook hier vervallen.
- In overige ontwerprijtingen wordt de standaard (maximale) stapgrootte gehanteerd.

Wijzigingen t.o.v. alternatief 3, t.b.v. alternatief 5:

```
SUBROUTINE MATSET
INTEGER INTG(1)
|
|
IF (JITER .EQ. 0) GOTO 22
|
|
DO 20 I = 1, NVAR
IF (REA(KDFMDV-1+I) * REA(KDFMDL-1+I) .GT. 0.0D0) GOTO 20
INTG(KCOUNT-1+I) = INTG(KCOUNT-1+I) + 1
IF (INTG(KCOUNT-1+I) .LT. 2) GOTO 20
CALL STEPGR
REA(KSTEP-1+I) = RSTEP
20 CONTINUE
22 CONTINUE
IF (FMIN .LT. FMINOL) GOTO 30
FMIN = FMINOL
|
|
RETURN
END
```

```
SUBROUTINE INITI 1
INTEGER LREA, LINT
|
KLASTR = KSTEP + NVAR
|
KCONTY = KLOCMA + MAXDFR * NEL
KCCUNT = KCONTY + NCON + 1
KLASTI = KCCUNT + NVAR
|
RETURN
END
```

SUBROUTINE STEPRG → wijzigingen s.o.v. alternatief 3
zijn hetzelfde als beschreven in
alternatief 4.

Na uitvoering van alternatief 4 bleek uit het iteratie-
verloop der ontwerpvariabelen (zie bijlage 12) dat ingrijpen
in de stapgrootte sterk remmend werkte op de ontwikkeling
in meer optimale richting, van die ontwerpvariabelen die
gedurende het hele verloop slechts één tekenwisseling ver-
tonen. In alternatief 5 wordt de stapgrootte daarom pas
aangepast nadat tekenwisseling twee of meer keer (in
niet noodzakelijk opvolgende iteratie-punten) is opgetreden.

2.6 Staphalvering na twee of meer tekenwisselingen v. afgeleide

Kenmerken van alternatief 6:

- Iteratie s.o.v. momentaan optimale ontwerpbeurt
- Staphalvering nadat in twee of meer (niet noodzakelijk opvolgende) iteraties is opgetreden: $(\partial F / \partial x_i) | x_k \cdot (\partial F / \partial x_i) | x_{k-1} \leq 0$
- De eis $F(x_k) \geq F(x_{k-1})$ is vervallen. $i=1, \dots, n$

- In overige ontwerprichtingen behoud de move-

Wijzigingen in het oorspronkelijke programma t.b.v. alternatief 6:

SUBROUTINE TUNING

INTEGER INTG(1), LREA, LINT, INDEX

CALL DATOUT

IF (NCON .GT. 0) CALL COPYRV (REA(KCONCL), REA(KCONST), NCON)

CALL USER

CALL DERIVA

SUBROUTINE DERIVA

INTEGER INTG(1)

\$INSERT DYNOPT.COMMON

INTEGER NVAR, IREQ, NDMO, NDMOLD, LENGTH, NUMB

REAL * 8 CHANGE, FREQ, DFMIN

IF (NVAR .EQ. 0) GOTO 200

CALL COPYRV (REA(KDFMCL), REA(KDFMDV), NVAR)

NUMB = NVAR * NCON

IF (NCON .GT. 0) CALL COPYRV (REA(KDCLCL), REA(KDCLDV), NUMB)

DO 150 IVAR = 1, NVAR

150 CONTINUE

200 RETURN

END

IF (LEND) GOTO 200

CALL MATSET

SUBROUTINE MATSET

INTEGER INTG(1)

\$INSERT DYNOPT. COMMON

INTEGER I, IND, NUMB

REAL *8 ABSTEP, VARLOW, VARUP

IF (JITER .EQ. 0) GOTO 22

DO 20 I = 1, NVAR

IF (REA(KDFMDV-1+I) * REA(KDFMDL-1+I) .GT. 2.0D4) GOTO 20

INTG(KCOUNT-1+I) = INTG(KCOUNT-1+I) + 1

IF (INTG(KCOUNT-1+I) .LT. 2) GOTO 20

REA(KSTEP-1+I) = REA(KSTEP-1+I) / 2.0D4

20 CONTINUE

22 CONTINUE

IF (FMIN .LT. FMINDL) GOTO 30

FMIN = FMINDL

CALL COPYRV (REA(KVAR), REA(KVAROL), NVAR)

IF (NCON .GT. 0) CALL COPYRV (REA(KCONST), REA(KCONCL), NCON)

CALL COPYRV (REA(KDFMDV), REA(KDFMDL), NVAR)

NUMB = NVAR * NCON

IF (NCON .GT. 0) CALL COPYRV (REA(KDCDV), REA(KDCCL), NUMB)

30 CONTINUE

IF (NCON .EQ. 0) GOTO 50

!

!

50 CONTINUE

DO 60 I = 1, NVAR

ABSTEP = DABS (REA(KSTEP-1+I))

VARLOW = REA(KVAR-1+I) - ABSTEP

!

60 CONTINUE

RETURN

END

CALL MATSOL → SUBROUTINE MATSOL

INTEGER INTG(1)

!

```
      :  
      LW = 9 * NVAR + 7 * NCON  
      CALL COPYRV (REA (KVARCL), REA (KVAR), NVAR)  
      IBOUND = 0  
      :  
      :  
      RETURN  
      END
```

CALL TIMEOUT

JITER = JITER + 1

GOTO 100

200 CONTINUE

CALL DATOUT (REA (1), INTG (1), 12)

RETURN

END

SUBROUTINE INITI 1

INTEGER LREA, LINT

KVAR = KCOOR + MAXCOR * NP

KVARCL = KVAR + NVAR

KVARI = KVARCL + NVAR

KDEMDV = KDFDN + NOPTFR * NELSEC

KDFMCL = KDEMDV + NVAR

KCONST = KDFMCL + NVAR

KCENCL = KCONST + NCON

KDCODV = KCENCL + NCON

KDCCOL = KDCODV + NCON * NVAR + NVAR

KLOWBO = KDCCOL + NCON * NVAR + NVAR

KEREL = KYTUN + NSECTO * NTUNFR

KSTEP = KEREL + NOPTFR

$$KLASTR = KSTEP + NVAR$$

$$KCONTY = KLOCMA + MAXDFR * NEL$$

$$KCCUNT = KCONTY + NCON + 1$$

$$KLASTI = KCCUNT + NVAR$$

RETURN

END

In dit alternatief is afgestapt van stapregulatie door afgeleiden naar en funktiewaarden van de objektfunctie omdat deze methode in alternatief 3, 4 en 5 steeds zeer kleine stapgrootten opleverde. De verwachte grotere convergentie-snelheid heeft zich niet geopenbaard. Het principe van ingrijpen in de stapgrootte na twee tekenwisselingen is hier gebruikt omdat deze methode uit het onderlinge vergelijk tussen alternatief 3 t/m 5 (zie par. 3.3) als beste naar voren komt.

2.7. Stapverdubbeling na gelijk teken v. afgeleide in twee of meer opeenvolgende iteraties

Kenmerken van alternatief 7:

- Iteratie t.o.v. momentaan optimale ontwerpbeurt
 - Staphalvering nadat in twee of meer opeenvolgende iteraties is opgetreden: $(\partial F / \partial x_i) | z_k * (\partial F / \partial x_i) | z_{k-1} \leq 0, i=1, \dots, n$
 - Stapverdubbeling nadat in twee of meer opeenvolgende iteraties is opgetreden: $(\partial F / \partial x_i) | z_k * (\partial F / \partial x_i) | z_{k-1} > 0, i=1, \dots, n$
- (N.B. bij toepassing van halvering / verdubbeling is het teken van de afgeleide in dat geval in drie opeenvolgende ontwerpbeurten wisselend / gelijk van teken).

Wijzigingen A.o.v. alternatief 6 o.b.v. alternatief 7:

SUBROUTINE MATSET

INTEGER INTG(I)

!
\$INSERT DYNOPT.COMMON

DO 20 I = 1, NVAR

IF (REA (KDFMDV-1+I) * REA (KDFMOL-1+I) .GT. ϕ . ϕ .D ϕ) GOTO 10

INTG (KCOUNT-1+I) = INTG (KCOUNT-1+I) + 1

INTG (KSAME-1+I) = ϕ

IF (INTG (KCOUNT-1+I) .LT. 2) GOTO 2 ϕ

REA (KSTEP-1+I) = REA (KSTEP-1+I) / 2. ϕ .D ϕ

GOTO 2 ϕ

10 CONTINUE

INTG (KSAME-1+I) = INTG (KSAME-1+I) + 1

INTG (KCOUNT-1+I) = ϕ

IF (INTG (KSAME-1+I) .LT. 2) GOTO 2 ϕ

REA (KSTEP-1+I) = REA (KSTEP-1+I) * 2. ϕ .D ϕ

2 ϕ CONTINUE

IF (FMLN .LT. FMINOL) GOTO 3 ϕ

!
3 ϕ CONTINUE

IF (NCON .EQ. ϕ) GOTO 5 ϕ

!
5 ϕ CONTINUE

DO 6 ϕ I = 1, NVAR

!
6 ϕ CONTINUE

RETURN

END

SUBROUTINE INITI 1

```
INTEGER LREA, LINT
|
KCOUNT = KCONTY + NCON + 1
KSAME = KCOUNT + NVAR
KLASTI = KSAME + NVAR
|
RETURN
END
```

Het idee om ook stapverdubbeling in te bouwen in het programma ontstond nadat was gebleken dat vergroting van de waarden van de move-limits in de oorspronkelijke invoerfile, voor die ontwerpvariabelen met slechts zeer weinig tekenwisselingen, een sterke daling van de objectfunctie te zien gaf. Als gedurende twee iteraties de afgeleide niet van teken wisselt, is de kans groot dat in de daaropvolgende iteratie de ontwerpvariabele weer in dezelfde richting gevarieerd gaat worden. De objectfunctie daalt dan sneller als dit met grotere stappen geschiedt. De eis van twee keer een getijk teken / wisselend teken, alvorens stapverdubbeling / stapahalvering toe te passen dient ter voorkoming van het optreden van sterke pieken in het iteratie-verloop van de ontwerpvariabelen en daardoor mogelijk ook in de objectfunctie.

2.8 Staphalvering bij elke tekenwisseling v.d. afgeleide

Kenmerken van alternatief 8:

- Iteratie 1.0.v. momentaan optimaal ontwerp punt.
- Staphalvering elke keer nadat is opgetreden:

$$(\partial F / \partial x_i) | x_k * (\partial F / \partial x_i) | x_{k-1} \leq 0, \quad i = 1, \dots, n$$

- Stapverdubbeling nadat in twee of meer opeenvolgende iteraties is opgetreden: $(\partial F / \partial x_i) | x_k * (\partial F / \partial x_i) | x_{k-1} \leq 0$, $i = 1, \dots, n$
- In richtingen waarin niet aan een van deze twee voorwaarden voldaan is, wordt de waarde van de move-limit uit de vorige iteratie gehandhaafd.

Wijziging t.o.v. alternatief 7 t.b.v. alternatief 8:

De enige wijziging bevindt zich in de volgende regel van subroutine MATSET:

```
IF (INTG (KCOUNT-1+I) .LT. 1) GOTO 20
```

↳ i.p.v. voorheen 2

Vergelijking van alternatief 8 met alternatief 7 leert dat het sneller reageren op het optreden van tekenwisseling der afgeleiden voor een sterker dalend verloop van de objekt-functie zorgdraagt (zie bijlage 5 en vooral bijlage 6), met zelfs een geringer aantal pieken in het verloop. Dit laatste is in tegenspraak met de verwachting, uitgesproken in paragraaf 2.7.

3. Evaluatie der alternatieven

Om te komen tot een beoordeling van het effect van de modificaties voorgesteld in de verschillende alternatieven, werd het programma DYNOPT gedraaid met als invoerfile: 'Russische klok'. Het eindige elementenmodel van deze klok bestaat in oorspronkelijke uitvoering uit 19 elementen. Voor het doorrekenen van 12 iteraties met dit model bleek een CPU-tijd nodig te zijn van ± 12000 sec. Aangerien het hier gaat

om een vergelijking van de verschillende alternatieven en de exactheid van de resultaten op dit moment nog van minder belang is, werd besloten het elementenmodel van de Russische klok te reduceren tot 5 elementen. De benodigde rekentijd voor 12 iteraties was nu nog maar ± 3000 sec. De invoerfile OPTIN in oorspronkelijke vorm voor 19 elementen is weergegeven op bijlagen 2a en 2b. De hierin vermelde startwaarden worden gerepresenteerd door startpunt x_A . Op bijlagen 1a en 1b is de A. b. v. het 5 elementenmodel gewijzigde versie van OPTIN te vinden. De hier vermelde startwaarden behoren bij startpunt x_B . Voor het draaien van DYNOPT met een 5 elementen-model invoerfile moest ook subroutine GEOIN worden gewijzigd, zoals op bijlage 3 is aangegeven.

Alternatieven 1 t/m 8 zijn doorgerkend vanuit zowel startpunt x_A als startpunt x_B . Dit vergelijken der alternatieven vanuit meer dan één startpunt is nodig om te controleren of de wijze waarop het iteratie-verloop van de objektfunctie bij elk alternatief zich t.o.v. de andere verlopen verhoudt, niet toevallig startpunt afhankelijk is. De iteratie-verlopen vanuit x_A op bijlage 4 vertonen een zelfde tendens als die vanuit x_B (bijlage 5). Een grotere zekerheid van deze onafhankelijkheid kan ten koste van zeer veel rekentijd verkregen worden door vanuit nog andere startpunten een grafiek op te tekenen zoals op bijlagen 4 en 5 is weergegeven.

De keuze van 12 iteraties als het in OPTIN op te geven maximale aantal is voortgekomen uit het feit dat het oorspronkelijke objektfunctie-iteratieverloop na 6 iteraties nog geen toename in waarde te zien gaf en zodoende het stapverkleiningsmechanisme nog niet in werking was getreden. Dit laatste is essentieel noodzakelijk om een evaluatie

van de verschillende alternatieven mogelijk te maken.

3.1 Staphalvering zonder meer versus onder voorwaarde van tekenwisseling

In de oorspronkelijke versie van DYNOPT wordt van alle ontwerpvariabelen de stapgrootte gehalveerd zodra de objektfunctie in een volgende iteratie in waarde is toegenomen. Dit heeft tot gevolg dat variabelen die nog geen tekenwisseling van afgeleide naar de objektfunctie vertoond hebben of gedurende meerdere opvolgende iteraties niet meer laten zien, beperkt worden in hun ontwikkeling. Op bijlage 8 is dit bij het verloop van ontwerpvariabelen x_{14} t/m x_{17} duidelijk te zien (knik in het verloop). Alternatief 1 op bijlage 9 laat voor deze variabelen een snellere doorgroei naar meer optimale waarden zien. In de grafiek op bijlage 4 komt dit tot uiting in de vorm van een iteratie-verloop voor alternatief 1 dat beneden het oorspronkelijk verloop ligt, vanaf iteratie 7: De eerste toename van de objektfunctie en daarmee de plaats aangevend waar ingrijpen in de stapgrootte optreedt.

Bijlagen 8 en 9 vertonen overeenkomstige oscillaties bij een groot aantal variabelen. Bij deze ontwerpvariabelen is ingrijpen in de stapgrootte nodig reeds lang voordat de objektfunctie in waarde toeneemt. In alternatief 4 wordt aan deze wens tegemoet gekomen.

3.2 Iteratie t.o.v. het laatste ontwerp punt versus t.o.v. momentaan optimale ontwerp punt

Iteratie t.o.v. het momentaan optimale ontwerp punt is opgezet vanuit de gedachte dat, gegeven een begrensd toege-

stane verandering der ontwerpvariabelen per iteratieslag, de objektfunctie het snelst moet kunnen dalen door steeds uit te gaan van de plaats waar diens waarde het laagst is. Bij een toename van de objektfunctie wordt daarom in alternatief 2 teruggegrepen naar het vorige iteratie-punt, zoals op bijlage 10 bij de voortgang van iteratie 7 naar iteratie 8 te zien is. Het iteratie-verloop van de objektfunctie voor de allereerste versie van alternatief 2 ('alt. 2 (versie met fouten)' op bijlage 4) ligt inderdaad beduidend beneden de grafiek 'alt. 1 (versie met fouten)'. De ontdekking van deze fouten, vermeld op bijlage 20 en reeds aanwezig in het ongewijzigde programma van DYNOPT, werd pas gedaan bij de opzet van alternatief 4. Hun aanwezigheid had tot gevolg dat de afgeleiden van de objektfunctie naar de ontwerpvariabelen verkeerd berekend werden en een gedeelte van het array met waarden voor de ontwerpvariabelen werd overschreven. De juiste grafieken 'alt. 1' en 'alt. 2', geven een tegengesteld beeld te zien: Iteratie t.o.v. het laatste punt bij alternatief 1, levert een verloop op dat ligt beneden dat van alternatief 2. De reden om toch voor iteratie t.o.v. het momentaan optimale punt te kiezen in alle volgende alternatieven, is de constatering dat de resultaten van het alternatief dat straks als beste tevoorschijn zal komen, met het 19 elementenmodel als invoerfile betere resultaten geeft voor iteratie t.o.v. het momentaan optimale punt dan voor iteratie t.o.v. het laatste ontwerp punt. Dat houdt in dat vergelijking der alternatieven, gebaseerd op de berekeningen met het 5 elementen-model, een verkeerd beeld geeft voor zover het de keuze tussen itereren t.o.v. laatste of momentaan optimale ontwerp punt betreft.

3.3 Criterium voor ingrijpen in stapgrootte

Het onderzoek naar de invloed van verschillende criteria voor het ingrijpen in de stapgrootte is verricht op de alternatieven die als ingebouwd stapverkleiningsmechanisme hadden: Stapregulatie door afgeleiden naar en functiewaarden van de objektfunctie. Een vergelijking tussen deze methode van regulatie van de stapgrootte en die van staphalvering komt aan de orde in paragraaf 3.4.

Het iteratie-verloop der ontwerpvariabelen van alternatief 3 op bijlage 11 vertoont bij een groot aantal ontwerpvariabelen het eerder besproken heen en weer springen tussen twee waarden. Ingrijpen in de stapgrootte vindt hier pas plaats na de iteratie waarbij de objektfunctie in waarde is toegenomen. Iteraties liggend tussen de iteratie waarbij voor het eerst tekenwisseling van de afgeleide optreedt en die met de eerste objektfunctie toename zijn voor oscillerende ontwerpvariabelen nutteloos. De convergentie-snelheid van de objektfunctie, d.w.z. afname ervan per iteratie, wordt er door verlaagd. In alternatief 4 is de eis van objektfunctie toename voor ingrijpen in de stapgrootte komen te vervallen. Het bijbehorende iteratie-verloop van de ontwerpvariabelen op bijlage 12 vertoont nu veel minder oscillaties. Het verloop van de objektfunctie op bijlage 5 ligt voor alternatief 4 gemiddeld genomen onder dat van alternatief 3. Het laat in de laatste paar iteraties in de reeks van 12 ook niet zulke sterke pieken zien als bij het vorige alternatief. Vanuit startpunt x_B laat alternatief 4 helaas juist grotere pieken zien dan alternatief 3.

Het iteratie-verloop van de objektfunctie ligt voor beide alternatieven 3 en 4 zowel vanuit x_A als vanuit x_B , vanaf

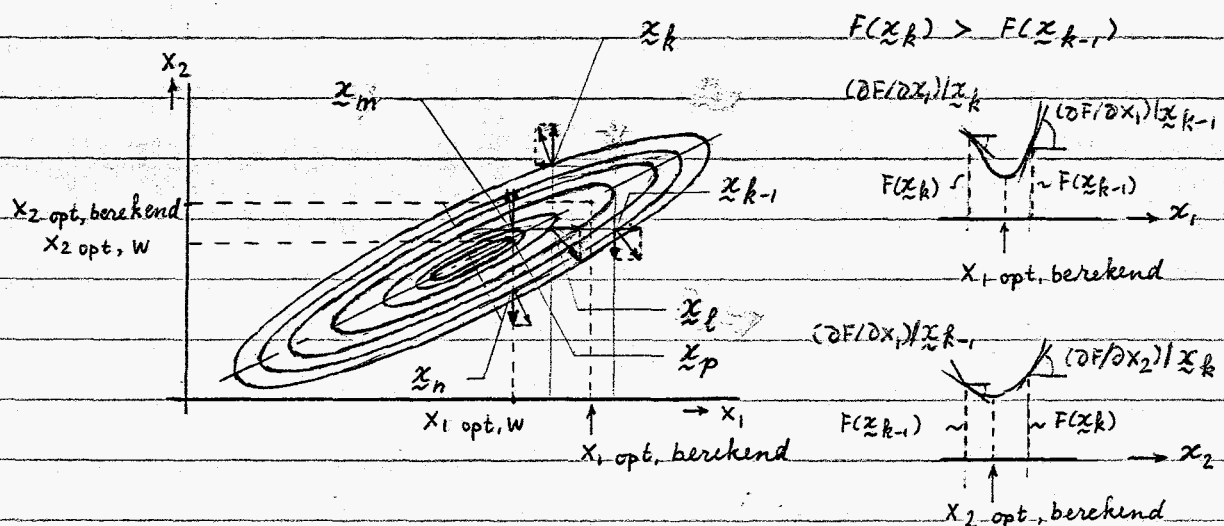
iteratie 8 boven het verloop bij het oorspronkelijke programma (waarin de fouten op bijlage 20 zijn verbeterd). De stapgrootte die wordt berekend door afgeleiden naar en functie-waarden van objektfunctie blijkt in de meeste gevallen zeer klein te zijn. Bijlage 12 toont dat dit voor de ontwerpvariabelen x_{15} , x_{16} en x_{17} die in hun helle verloop hooguit twee tekenwisselingen ontmoeten, remmend werkt op hun ontwikkeling. In alternatief 5 is dit nadeel opgeheven door pas na twee opgetreden tekenwisselingen stapregulatie toe te passen. Het iteratie-verloop van de objektfunctie bij dit alternatief ligt gemiddeld onder dat van 3 en van 4, maar nog steeds boven dat van het oorspronkelijke programma (bijlage 5). Vanuit startpunt \bar{x}_B doet alternatief 5 het in-eens veel beter (zie bijlage 6). Alternatief 3 en 4 liggen ook hier nog steeds met het laatste deel van hun objektfunctie iteratie-verloop boven het oorspronkelijk verloop. Een verklaring voor de tegenvallende resultaten van alternatieven 3 t/m 5 wordt gegeven in paragraaf 3.4. Het piekerige verloop dat de objektfunctie bij deze alternatieven vertoont, wordt veroorzaakt door het hanteren van de standaard maximale stapgrootte bij die ontwerpvariabelen waarop geen stapregulatie wordt toegepast. Deze modificatie is dan ook weer uit de nog te bespreken alternatieven verwijderd.

3.4 Stapregulatie d.m.v. afgeleiden / objektfunctie-waarden versus staphalvering (aangevuld met stapverdubbeling)

De resultaten van de alternatieven 3 t/m 5 met de daarin toegepaste stapregulatie door afgeleiden / objektfunctie-waarden vallen erg tegen: Op bijlage 5 ligt het verloop van

alle drie de alternatieven en op bijlage 6 dat van alternatief 3 en het laatste deel van alternatief 4 boven het oorspronkelijk objektfunctie-verloop. Een verklaring hiervoor luidt als volgt:

Bij de berekening van het optimale punt in een gegeven ontwerprichting, worden de afgeleiden van objektfunctie naar de bijbehorende ontwerpvariabele gebruikt uit het vorige iteratie-punt \underline{x}_{k-1} , en het huidige punt \underline{x}_k . Optimalisatie in een bepaalde ontwerprichting geschiedt dus niet met constant gehouden waarden der ontwerpvariabelen in alle overige richtingen. Aan de hand van een functie van 2 variabelen, zoals afgebeeld in figuur 3.4.1, worden de ernstige berwaren tegen deze methode verduidelijkt.



Figuur (3.4.1) : Plaats van bepaling van partiële afgeleiden t.b.v. stapregulatie door afgeleiden / objektfunctiewaarden

Optimalisatie van de stapgrootte in ontwerprichting 1, te gebruiken in de volgende iteratie vanaf momentaan optimaal punt \underline{x}_{k-1} , geschiedt m.b.v. de partiële afgeleiden $(\partial F / \partial x_1) | \underline{x}_{k-1}$ en $(\partial F / \partial x_2) | \underline{x}_k$. Het teken van deze laatste waarde is gezegesteld aan dat van $(\partial F / \partial x_1) | \underline{x}_k$ dat eigenlijk gebruikt had moeten worden (en ook $F(\underline{x}_k)$ i.p.v. $F(\underline{x}_{k-1})$)

om een juiste derdegraads polynoom-benadering te verkrijgen van de objektfunctie. Nu wordt een minimum berekend liggend tussen $x_{1,k}$ en $x_{1,k-1}$, terwijl dit in werkelijkheid links van $x_{1,k}$ ligt: $x_{1,opt,w}$. De stapgrootte, zijnde het verschil $x_{1,k-1} - x_{1,opt,w}$, berekend, is nu veel te klein. De huidige methode van stapregulatie door afgeleiden / objekt-functiewaarden berekend vervolgens m.b.v. $(\partial F / \partial x_2) | x_{k-1}$ en $(\partial F / \partial x_2) | x_k$ het optimum daartussen ($x_2 opt, berekend$) en komt met het volgende iteratie-punt in de buurt van de diagonaal van de ellips terecht. Van de punten x_{k-1} en x_k mogen we alleen dan gebruik maken bij het berekenen van de coëfficiënten voor de polynoom-vergelijking, als voor de afgeleiden de afgeleiden naar de objektfunctie in de richting van een lijn door x_{k-1} en x_k , in deze punten wordt genomen. Bij functies van meer dan 3 ontwerpvariabelen wordt dit reeds een lastige zaak.

Beter is het per ontwerprichting een optimale stapgrootte te bepalen, waarbij alle overige ontwerpvariabelen constant worden gehouden: In x_{k-1} wordt $(\partial F / \partial x_1) | x_{k-1}$ bepaald. Vervolgens wordt alleen variabele x_1 veranderd: $x_{1,k-1} : x_{1,k-1} - \Delta x_1 = x_{1,k}$. Hierna bepaalt men $(\partial F / \partial x_1) | x_k$ in het huidige ontwerp-punt x_k . Nu kan $x_{1,opt,w}$ bepaald worden, waarmee het huidige ontwerp-punt is verworden tot x_m . Herhaling van het voorgaande in de richting 2 levert ons iteratie-punt x_p . De hier beschreven methode is die der geconjugeerde gradiënten zoals beschreven in lit [2], hoofdstuk 8.

Inbouwen van deze methode in de bestaande program-matuur van DYNOPT vergt meer omvangrijke wijzigingen. Bovendien is bij toepassing ervan een groot aantal itera-ties nodig: 2 per ontwerprichting. Vooralsnog zoeken we verbetering van het convergentie-mechanisme in methoden

die hopelijk minder iteraties nodig hebben voor het opsporen van het (locale) minimum.

In alternatief 6 is weer het mechanisme van staphalving geïntroduceerd, gecombineerd met het criterium dat in de evaluatie van paragraaf 3.3 als beste uit de bus kwam. Het iteratie-verloop van de objektfunctie bij alternatief 6 vertoont zowel vanuit startpunt \underline{x}_A als vanuit startpunt \underline{x}_B een monotoon dalend verloop. De toename in convergentie-snelheid t.o.v. de oorspronkelijke situatie is echter niet noemenswaardig. Het iteratie-verloop der ontwerpvariabelen van alternatief 6 op bijlage 14 had er preciezer uit gezien bij iteratie t.o.v. het laatste bepaalde ontwerppunt, aangezien dit laatste punt steeds momentaan optimaal is. Er ontbreken nu de 'losse staarten' die ontstaan op het moment van objektfunctie-toename, bij iteratie t.o.v. het momentaan optimale ontwerppunt. Het verloop toont ook dat ontwerpvariabelen x_{10} en x_{11} na de 5^e iteratie en variabelen x_{12} , x_{13} , x_{16} en x_{17} na de 4^e iteratie geen tekenwisseling van de afgeleide meer vertonen. Zulke variabelen zijn in dat geval gebaat bij vergroting van de stapgrootte per iteratie. Voor ontwerpvariabele x_{14} , zonder ook maar één tekenwisseling geldt zulks zeker. Alternatief 7 heeft het mechanisme van stapverdubbeling ingebouwd gekregen. Het treedt in werking zodra gedurende twee opvolgende iteraties, dus in drie opvolgende ontwerppunten, het teken van de afgeleide ongewijzigd is gebleven. Het bijbehorende iteratie-verloop van de objektfunctie bereikt met \underline{x}_A als startpunt een lagere waarde dan alle eerder besproken alternatieven (zie bijlage 5). Helaas zijn er in het iteratie-verloop flinke pieken te zien, wat in het verloop vanuit startpunt \underline{x}_B

(bijlage 6) nog veel erger is. Het iteratie-verloop van de ontwerpvariabelen vertoont op bijlage 15 het vastlopen van ontwerpvariabele x_{14} tegen zijn ondergrens. Ontwerpvariabele x_{13} behoudt in iteratie 12 dezelfde waarde als in iteratie 11, terwijl x_{17} in iteratie 11 terugkeert naar zijn waarde van iteratie 9. Dit is het gevolg van niet in werking treden van staphalvering na slechts één tekenwisseling. De waarden van hun afgeleide naar de objektfunctie in iteratie 12, doet vermoeden dat het inzetten van staphalvering reeds na de eerste tekenwisseling gunstiger is voor hun ontwikkeling. Bij variabele x_{15} treedt het 'te laat' inzetten van staphalvering ook op tussen iteratie 5 en iteratie 7.

Om deze reden wordt in alternatief 8 staphalvering reeds ingezet na de eerste opgetreden tekenwisseling van de afgeleide naar de objektfunctie. Dit is overigens de enige wijziging t.o.v. alternatief 7. Op bijlage 16 is te zien dat het iteratie-verloop van de ontwerpvariabelen x_{14} , x_{15} en x_{17} eerder de juiste kant op wordt gestuurd, dankzij het sneller reageren op tekenwisseling. De objektfunctie vertoont daardoor bij alternatief 8 minder grote pieken dan bij alternatief 7 (zie bijlage 5). Vanuit startpunt Z_A blijkt alternatief 7 lagere waarden voor de objektfunctie te bereiken dan alternatief 8. Aangezien dit vanuit startpunt Z_B niet zo is, mag dat gegeven niet doorslaggevend zijn bij het aanvragen van het beste alternatief. Het laatste alternatief is daarnaast zowel vanuit Z_A als vanuit Z_B met een minder piekerig objektfunctie-verloop opgepadeld dan alternatief 7.

Een variant op alternatief 8 is nog uitgeprobeerd, waarbij iteratie t.o.v. het laatste punt wordt toegepast i.p.v. iteratie t.o.v. het momentaan optimale punt. Nogmaals

wordt duidelijk dat het 5 elementen-model aangaande deze keuze een vals beeld schept: Voor het 5 elementen-model ligt het verloop van de objektfunctie bij alternatief 9 beneden dat van alternatief 8; voor het 19 elementen-model is dat net omgekeerd (zie bijlage 7). De situatie voor wat betreft ligging der verschillende iteratie-verlopen t.o.v. elkaar bij het 19 elementen-model weegt het zwaarst, bij de keuze van het uiteindelijk als beste aan te wijzen alternatief. Een 19 elementen-model beschrijft het continue lichaam van de klok namelijk beter dan een 5 elementen-model.

Als resultaat van deze evaluatie kan dan vermeld worden dat alternatief 8 de beste modificatie, uit de onderzochte groep alternatieven, van het oorspronkelijke programma DYNOPT is, ter verbetering van het convergentiemechanisme.

4. Vergelijking der resultaten v. h. gemodificeerde programma met die v. d. oorspronkelijke versie

Ter beoordeling van de winst in convergentie-snelheid maken we gebruik van de resultaten die werden verkregen met gebruik van de invoerfile voor het 19 elementen-model (bijlage 2 a en 2 b), dat het dichtst bij de werkelijkheid staat. Aan de uitkomsten hechten we nu, in tegenstelling tot de fase van de evaluatie der alternatieven, wel absolute betekenis: Het vinden van die waarden voor de ontwerpvariabelen die een klok opleveren met frequenties, zo dicht mogelijk bij de gewenste frequenties. Beter be-

nadering van frequenties is gekoppeld aan een laagere objekt-functiewaarde (zie uitdrukking (1.3)) en aangezien startpunt \underline{x}_8 de laagste objektfunctie-waarden aan het eind van de iteratie-reeks laat zien (vergeleijk bijlage 5 met bijlage 6), geniet dit startpunt nu de voorkeur.

Het iteratie-verloop van de objektfunctie voor het oorspronkelijke programma met het 19 elementen-model van de Russische klok als invoerfile, is op bijlage 7 als referentie uitgezet. Het valt op dat de range van het iteratieverloop, d.w.z. het verschil tussen $F(\underline{x}_0)$ en $F(\underline{x}_{12})$, groter is dan bij het 5 elementen-model. Objektfunctie-waarden gedurende het iteratie-verloop voor de verschillende alternatieven, welke zijn verwerkt in de grafieken op bijlagen 4 t/m 7, zijn in tabel-vorm weergegeven op bijlagen 18 en 19. Ook voor alternatief 8 neemt deze range toe bij overgang van 5 elementen-model naar 19 elementen-model. Belangrijker is echter dat de laatste objektfunctie-waarde $F(\underline{x}_{12})$ van alternatief 8 ook nu onder de corresponderende waarde van het oorspronkelijk verloop ligt. Er is dankzij de modificatie dus winst geboekt in convergentie-snelheid, maar deze is helaas niet spectaculair te noemen. Alternatief 9 geeft, zoals reeds eerder vermeld, bij het 19 elementen-model een slechtere convergentie dan alternatief 8. In de invoerfile van onze 'Russische klok' is een set van 7 stuks gewenste frequenties opgegeven (zie bijlage 1a en 1b of 2a en 2b). In de 12^e iteratie liggen bij alternatief 8 alleen de 'oktaaf' en de 'dubbeloktaaf' met hun relatieve afwijking binnen de toegestane marge van 3 cents wat overeenkomt met ongeveer 0,17%. Dit komt erop neer dat de objektfunctie nog een te hoge

waarde heeft. Met de ontwerpvariabelen uit iteratie 12 is alternatief 8 opnieuw gestart (waarden afgerond op 2 decimalen). Het bijbehorende verloop, aangeduid met B, toont dat de range nu veel kleiner is dan bij de eerste 12 iteraties: De objektfunctie neemt weliswaar nog steeds af, maar de convergentie-snelheid is sterk gedaald. In de 24^e iteratie wordt de tot dan toe laagste waarde bereikt, maar alleen de oktaaf voldoet op dat moment aan het convergentie-criterium.

Volgende herstarts beginnen pieken te vertonen aan het begin van hun iteratie-reeks: Het gevolg van de te grote waarden van de move-limits bij aanvang. Door het snelle reageren op tekenwisseling (kenmerk van alternatief 8) met staphalvering (zie bijlage 17 voor het iteratie-verloop der ontwerpvariabelen voor de eerste 12 iteraties met alternatief 8), vertoont het verloop van de objektfunctie na de 3^e iteratie reeds weer een dalende tendens. Alleen herstart F laat tussen iteratie 5 en iteratie 9 flinke schommelingen zien, wat herstart D en G aan het einde van hun verloop doen. Met zeer vlak verlopen de laatste deel van herstart H t/m herstart L doet vermoeden dat hier een (locaal) minimum in elk geval zeer dicht benaderd is. Melaas voor onze klok voldoen slechts de 'grondtoon', 'priem' en 'oktaaf' aan het convergentie-criterium.

Na 144 iteraties zijn de juiste ontwerpvariabelen voor deze lage tertsklok nog steeds niet gevonden. Dit biedt enig perspectief om toch de methode der geconjugeerde gradiënten toe te gaan passen, die immers eerder op grond van het verwachte grote aantal benodigde iteraties ter zijde was geschoven.

Bij dit alles dient wel bedacht te worden dat de keuze van het (allereerste) startpunt van grote invloed is: Het toevallig tegenkomen van een minder gunstig lokaal minimum, zet door de tekenwisselingen der afgeleiden niet zo goed staphalvering in werking, waardoor het verlaten van zo'n punt moeilijk of onmogelijk wordt. Proberen vanuit (zeer) verschillende startpunten blijft in deze noodzaak.

5. Conclusies en aanbevelingen

Het mechanisme van halvering van de move-limits bij toename van de objektfunctie, voor alle ontwerpvariabelen tegelijk, blijkt voor verbetering vatbaar te zijn. Het selecteren van ontwerpvariabelen met tekenwisseling van afgeleide naar objektfunctie en alleen daarop dit mechanisme laten werken als de objektfunctie in waarde is toegenomen (alternatief 1), blijkt reeds een geringe toename in convergentie-snelheid op te leveren. Een iets beter resultaat wordt bereikt als we de eis van objektfunctie toename voor ingrijpen in de stapgrootte inwisselen voor de eis van een keer of twee keer optreden van tekenwisseling van de afgeleide (alternatief 6).

Op basis van de resultaten verkregen met het 5 elementen-model als invoerfile, zijn geen juiste uitspraken te doen over de voordelen van iteratie t.o.v. het momentaan optimale punt in verhouding tot iteratie t.o.v. het laatste ontwerp punt. Met het 5 elementen-model levert de laatste methode de beste resultaten (vergelijk alt. 1 t.o.v. alt 2

op bijlage 4 en alt. 9 t.o.v. alt. 8 op bijlage 5). Voor het 19 elementen-model is iteratie t.o.v. het momentaan optimale ontwerppunt voordeliger (vergelijk alt. 8 t.o.v. alt. 9 op bijlage 7). De situatie zoals die zich voordoet bij het 19 elementen-model is doorslaggevend voor de te kiezen methode, aangezien dit model de werkelijkheid het best beschrijft.

Het stapverkleiningsmechanisme op basis van afgeleiden / objectfunktiewaarden komt op de manier waarop het hier in de bestaande programmatuur van DYNOPT is ingebouwd, niet tot zijn recht: Bij optimalisatie van de stapgrootte in een bepaalde ontwerprichting zijn de waarden der overige ontwerpvariabelen niet constant gehouden. Dit maakt een juiste benadering van de objectfunctie door een polynoom in de betreffende richting onmogelijk. Gevolg is een verkeerde berekening der stapgrootten (zeer kleine waarden) en daardoor een verslechtering van het convergentie-gedrag van de objectfunctie (zie alt. 3 t/m alt. 5 op bijlage 4). Toepassing van de zgn. methode der geconjugeerde gradiënten, met optimalisatie per ontwerprichting apart, verdient daarom sterk aanbeveling. Het bezwaar van vele benodigde iteraties bij deze methode (2 per ontwerprichting) kan in een ander licht worden gezien, als men bedenkt dat het hier gevonden beste alternatief ongeveer 144 iteraties nodig heeft om niet meer weg te lopen van een gevonden minimum. De beste resultaten levert, uit de groep die hier onderzocht is, alternatief 8: combinatie van staphalvering met stapverdubbeling. Trage convergentie speelt hier pas veel later op. De toename in convergentie-snelheid is echter niet overweldigend, wat ons doet uitziën naar de methode der geconjugeerde gradiënten of misschien introductie van kwadratische programmering in DYNOPT.

Literatuur

[1] F.G.J. van Asperen (1984)

Het optimaliseren van de eigenfrequenties van
axiaal-symmetrische constructies, toegepast op
een luid- of carillon-klok.

Afstudeerverslag T.H. Eindhoven ; WFW 84.012

[2] U. Kirsch

Optimum Structural Design

Mc. Graw-Hill Book Company, 1981

Invoerfile OPTIN aangepast voor het 5 elementen-model

```

(      1) *TITLE
(      2) RUSSISCHE KLOK
(      3) *ELEMENTS
(      4) 5
(      5) *NODPOINTS
(      6) 28
(      7) *NBOUNDCOND
(      8) 9
(      9) *ELTYPE
(     10) 1
(     11) 2
(     12) *NDESIGNVAR
(     13) 17
(     14) *GEOFLAG
(     15) *NFOURIER
(     16) 5
(     17) *NCPTRFREQ
(     18) 7 5
(     19) *OUTPUTOPTIONS
(     20) 0 0 1
(     21) *OPTIMAFLAG
(     22) *END PARA
(     23) *BOUNDCON
(     24) 26 1
(     25) 27 1
(     26) 28 1
(     27) 26 2
(     28) 27 2
(     29) 28 2
(     30) 26 3
(     31) 27 3
(     32) 28 3
(     33) *DESIGNVAR
(     34) 1 0.35 -3.0 2.0 0.02 'STRAAL1' /
(     35) 2 -0.03 -3.0 2.0 0.02 'STRAAL2' /
(     36) 3 -0.28 -3.0 2.0 0.02 'STRAAL3' /
(     37) 4 -0.29 -3.0 2.0 0.02 'STRAAL4' /
(     38) 5 0.28 -3.0 2.0 0.02 'STRAAL5' /
(     39) 6 0.29 -3.0 2.0 0.02 'STRAAL6' /
(     40) 7 -0.15 -3.0 2.0 0.02 'STRAAL7' /
(     41) 8 -0.17 -3.0 2.0 0.02 'STRAAL8' /
(     42) 9 -0.2 -4.0 5.0 0.08 'DIKTE1' /
(     43) 10 0.65 -4.0 5.0 0.08 'DIKTE2' /
(     44) 11 0.65 -4.0 5.0 0.08 'DIKTE3' /
(     45) 12 0.72 -4.0 5.0 0.08 'DIKTE4' /
(     46) 13 0.02 -4.0 5.0 0.08 'DIKTE5' /
(     47) 14 -0.94 -4.0 5.0 0.08 'DIKTE6' /
(     48) 15 -0.94 -4.0 5.0 0.08 'DIKTE7' /
(     49) 16 -0.74 -4.0 5.0 0.08 'DIKTE8' /
(     50) 17 -0.74 -4.0 5.0 0.08 'DIKTE9' /
(     51) *OPTIMIZA
(     52) 1 2 1 1.0 4.0348 'GRONDTOP' 'N G-2' /
(     53) 2 2 2 1.0 2.2256 'PRIEM P' '-2' /
(     54) 3 3 1 1.0 1.7461 'LAGE T' 'ERTS I-3' /
(     55) 4 3 2 1.0 1.4983 'KWINT I' 'I-3' /
(     56) 5 4 1 1.0 1.000 'OKTAAF ' 'I-4' /
(     57) 6 5 1 0.5 0.6674 'DUODECIE' 'M I-5' /
(     58) 7 6 1 0.5 0.5000 'DUBBELOK' 'TAAF I-' '6' /
(     59) *MATDATA
(     60) 0.106E 09 0.340 0.8830E-05

```

startpunt x_A

0.35
-0.03
-0.28
-0.29
0.28
0.29
-0.15
-0.17
-0.2
0.65
0.65
0.72
0.02
-0.94
-0.94
-0.74
-0.74

```
( 61) *CONNECTIVITY
( 62) 1 2 1 2 3 5 8 7 6 4 /
( 63) 2 2 6 7 8 10 13 12 11 9 /
( 64) 3 2 11 12 13 15 18 17 16 14 /
( 65) 4 2 16 17 18 20 23 22 21 19 /
( 66) 5 2 21 22 23 25 28 27 26 24 /
( 67) *ITERATION
( 68) 12 1.0E-4 1.0E-5 1.0E-6 1.0E-4 /
( 69) *END DATA
```

*TITLE

Bijlage 2a

Invoerfile OPTIN voor het 19 elementen - model

```

(      1) *TITLE
(      2) RUSSISCHE KLOK
(      3) *ELEMENTS
(      4) 19
(      5) *NEDPOINTS
(      6) 98
(      7) *NBOUNDCOND
(      8) 9
(      9) *ELTYPE
(     10) 1
(     11) 2
(     12) *NDESIGNVAR
(     13) 17
(     14) *GEOFLAG
(     15) *NFOURIER
(     16) 5
(     17) *NOPTFREQ
(     18) 7 5
(     19) *OUTPUTOPTIONS
(     20) 0 0 1
(     21) *OPTIMAFLAG
(     22) *END PARA
(     23) *BOUNDCON
(     24) 96 1
(     25) 97 1
(     26) 98 1
(     27) 96 2
(     28) 97 2
(     29) 98 2
(     30) 96 3
(     31) 97 3
(     32) 98 3
(     33) *DESIGNVAR
(     34) 1 0.22 -3.0 2.0 0.02 'STRAAL1' /
(     35) 2 -0.10 -3.0 2.0 0.02 'STRAAL2' /
(     36) 3 -0.22 -3.0 2.0 0.02 'STRAAL3' /
(     37) 4 -0.22 -3.0 2.0 0.02 'STRAAL4' /
(     38) 5 0.22 -3.0 2.0 0.02 'STRAAL5' /
(     39) 6 0.22 -3.0 2.0 0.02 'STRAAL6' /
(     40) 7 -0.22 -3.0 2.0 0.02 'STRAAL7' /
(     41) 8 -0.16 -3.0 2.0 0.02 'STRAAL8' /
(     42) 9 0.0 -4.0 5.0 0.08 'DIKTE1' /
(     43) 10 0.55 -4.0 5.0 0.08 'DIKTE2' /
(     44) 11 0.55 -4.0 5.0 0.08 'DIKTE3' /
(     45) 12 0.55 -4.0 5.0 0.08 'DIKTE4' /
(     46) 13 0.0 -4.0 5.0 0.08 'DIKTE5' /
(     47) 14 -0.55 -4.0 5.0 0.08 'DIKTE6' /
(     48) 15 -0.55 -4.0 5.0 0.08 'DIKTE7' /
(     49) 16 -0.55 -4.0 5.0 0.08 'DIKTE8' /
(     50) 17 -0.55 -4.0 5.0 0.08 'DIKTE9' /
(     51) *OPTIMIZA
(     52) 1 2 1 1.0 4.0348 'GRONDTOO' 'N 6-2' /
(     53) 2 2 2 1.0 2.2256 'PRIEM P' '-2' /
(     54) 3 3 1 1.0 1.7461 'LAGE T' 'ERTS I-3' /
(     55) 4 3 2 1.0 1.4983 'KWINT I' 'I-3' /
(     56) 5 4 1 1.0 1.000 'OKTAAF ' 'I-4' /
(     57) 6 5 1 0.5 0.6674 'DUODECIE' 'M I-5' /
(     58) 7 6 1 0.5 0.5000 'DUBBELOK' 'TAAF I-' '6' /
(     59) *MATDATA
(     60) 0.106E 09 0.340 0.8830E-05

```

startpunt x_B

34	1	0.22
35	2	-0.10
36	3	-0.22
37	4	-0.22
38	5	0.22
39	6	0.22
40	7	-0.22
41	8	-0.16
42	9	0.0
43	10	0.55
44	11	0.55
45	12	0.55
46	13	0.0
47	14	-0.55
48	15	-0.55
49	16	-0.55
50	17	-0.55

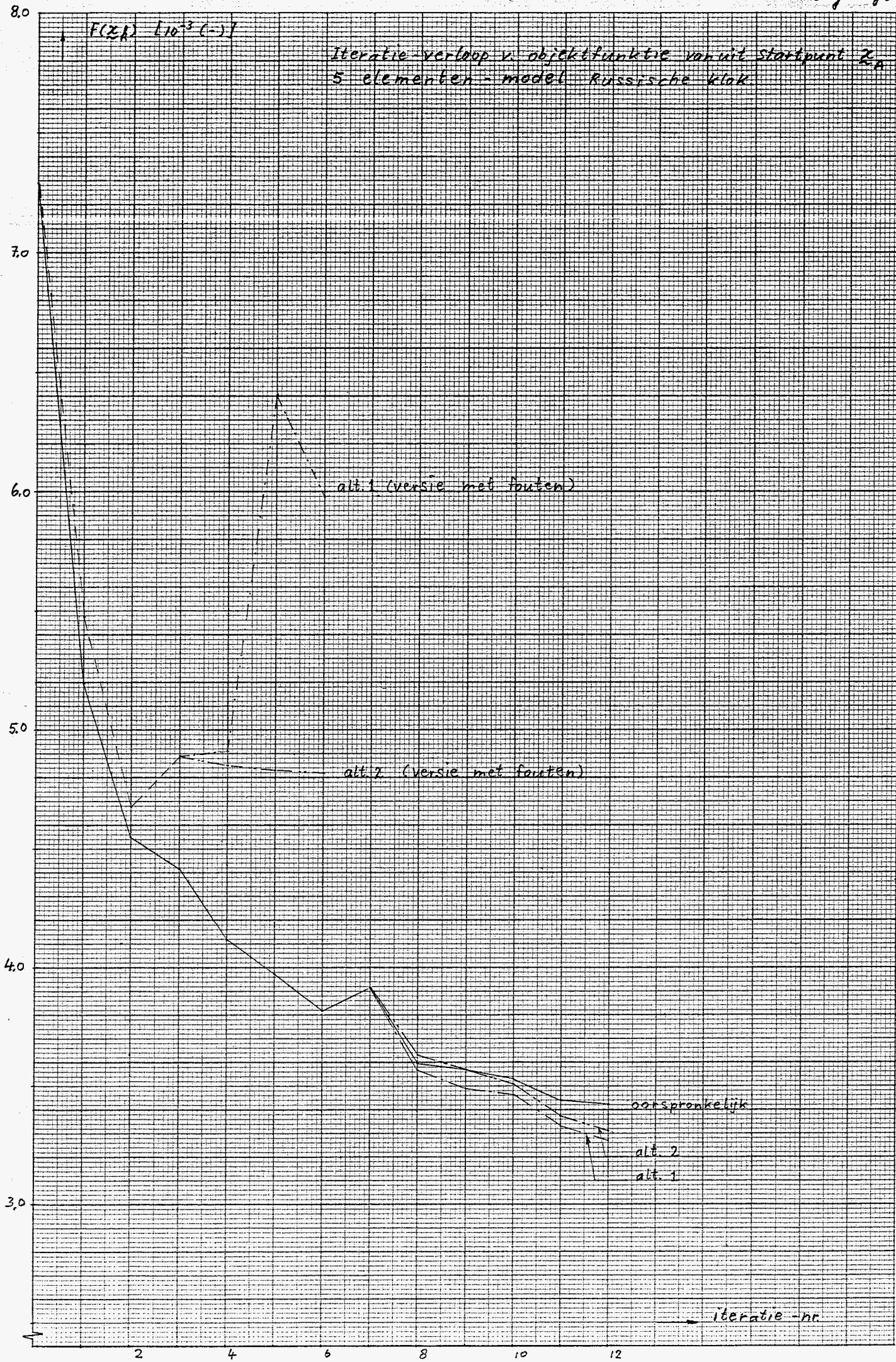

```
( 61) *CONNECTIVITY
( 62) 1 2 1 2 3 5 8 7 6 4 /
( 63) 2 2 6 7 8 10 13 12 11 9 /
( 64) 3 2 11 12 13 15 18 17 16 14 /
( 65) 4 2 16 17 18 20 23 22 21 19 /
( 66) 5 2 21 22 23 25 28 27 26 24 /
( 67) 6 2 26 27 28 30 33 32 31 29 /
( 68) 7 2 31 32 33 35 38 37 36 34 /
( 69) 8 2 36 37 38 40 43 42 41 39 /
( 70) 9 2 41 42 43 45 48 47 46 44 /
( 71) 10 2 46 47 48 50 53 52 51 49 /
( 72) 11 2 51 52 53 55 58 57 56 54 /
( 73) 12 2 56 57 58 60 63 62 61 59 /
( 74) 13 2 61 62 63 65 68 67 66 64 /
( 75) 14 2 66 67 68 70 73 72 71 69 /
( 76) 15 2 71 72 73 75 78 77 76 74 /
( 77) 16 2 76 77 78 80 83 82 81 79 /
( 78) 17 2 81 82 83 85 88 87 86 84 /
( 79) 18 2 86 87 88 90 93 92 91 89 /
( 80) 19 2 91 92 93 95 98 97 96 94 /
( 81) *ITERATION
( 82) 12 1.0E-4 1.0E-5 1.0E-6 1.0E-4 /
( 83) *END DATA
```

Invoerfile GEOIN

- (1) 5, 2
- (2) 100, 10, 10, 10, 4
- (3) 0.6, 0.4
- (4) 210.0
- (5) 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1
- (6) 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1
- (7) 2
- (8) 15
- (9) 1, 210.0, 0.0, 0.1
- (10) 2, 205.2, 4.0, -7.5
- (11) 3, 199.6, 10.0, 17.0
- (12) 4, 189.9, 20.0, 30.6
- (13) 5, 181.9, 30.0, 34.5
- (14) 6, 168.6, 45.0, 31.6
- (15) 7, 157.2, 60.0, -28.2
- (16) 8, 144.4, 80.0, 24.2
- (17) 9, 133.8, 100.0, -20.4
- (18) 10, 124.9, 120.0, 17.2
- (19) 11, 118.1, 140, -14.6
- (20) 12, 109.8, 160.0, 11.9
- (21) 13, 103.3, 240.0, 11.3
- (22) 14, 101.3, 270.0, -11.8
- (23) 15, 100.5, 295.0, 12.1
- (24) 16, 100.3, 300.0, -12.2
- (25) 8
- (26) 1, 4, 6, 8, 10, 12, 13, 15
- (27) 3, 4, 5, 6, 8, 10, 12, 13, 15
- (28) 14, 2, 3
- (29) 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0 } 3, 1, 1
- (30) 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0 } 1.0, 1.0, 1.0
- (31) 1.74545
- (32) 0.145455
- (33) 0.145455
- (34) 0.145455

wijzigingen in GEOIN t.b.v.
het 5 elementen-model



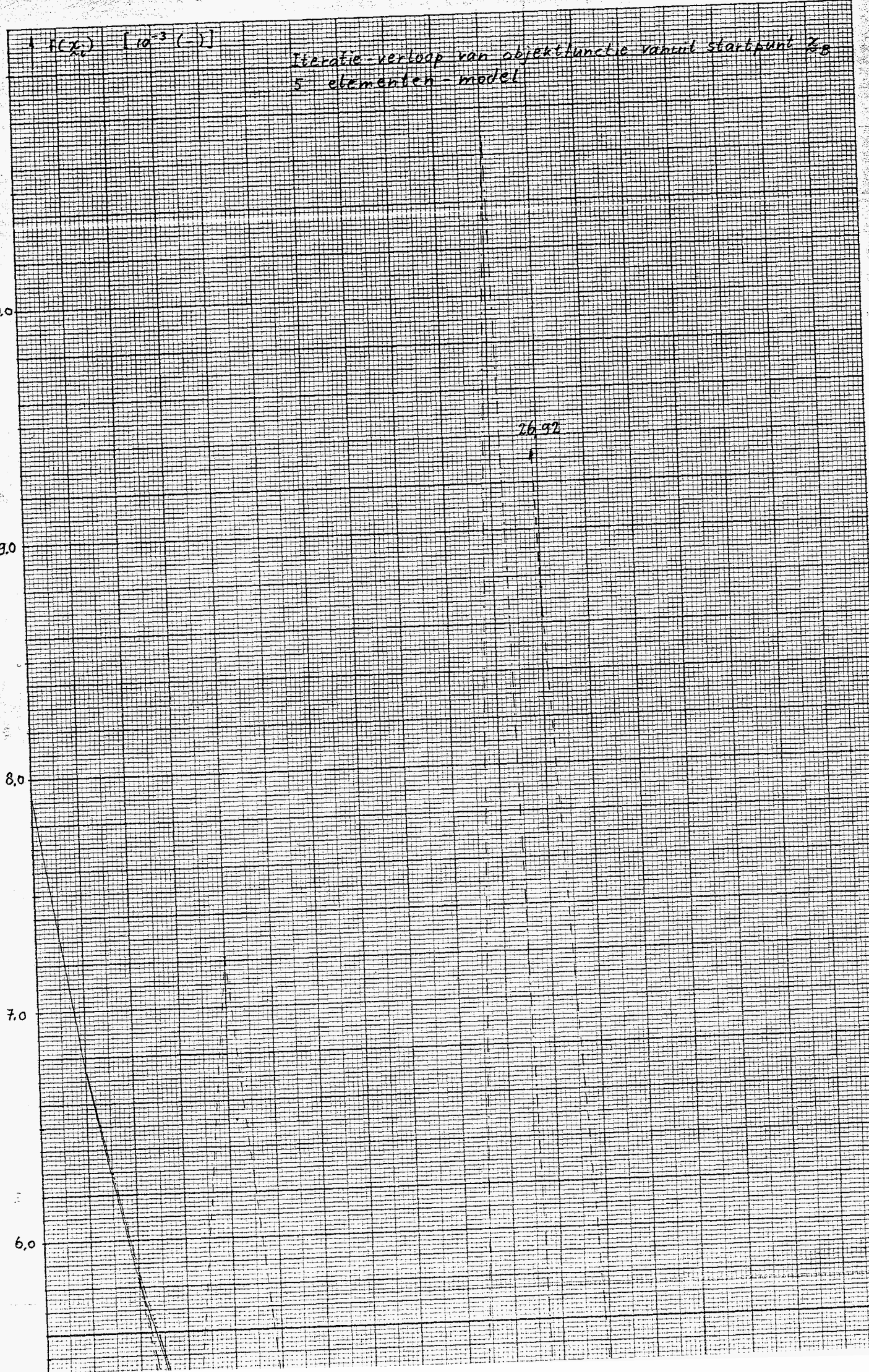


$f(x_0)$ [10^{-3} (-)]

Iteratie-verloop van objektfunctie vanuit startpunt x_0
5 elementen-model

10.0
9.0
8.0
7.0
6.0

26.92



Bijlage 6b

5.0

4.0

3.0

2.0

1.0

alt 7

alt 4

alt 3

oorspronkelijk

alt 1

alt 5

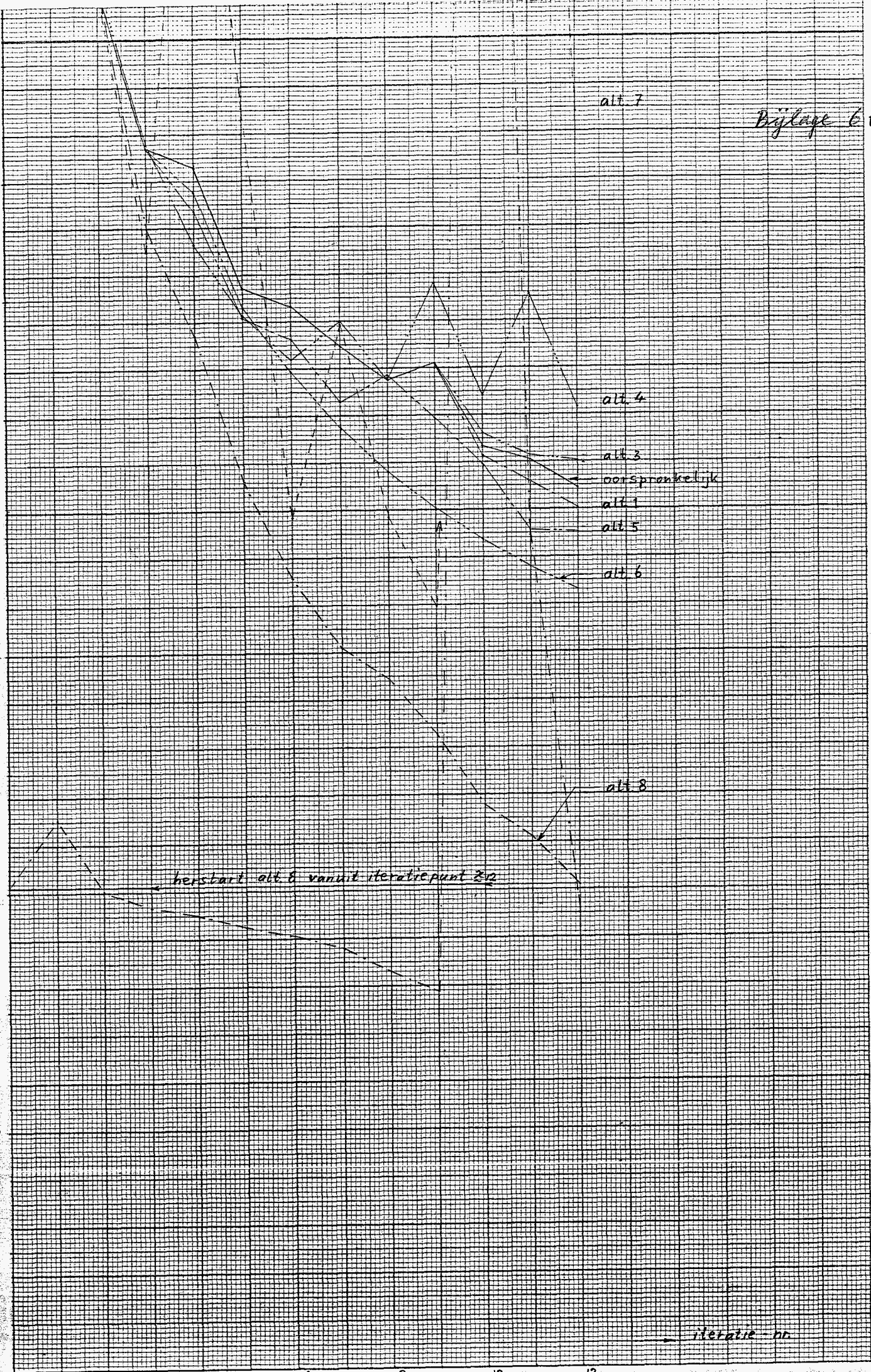
alt 6

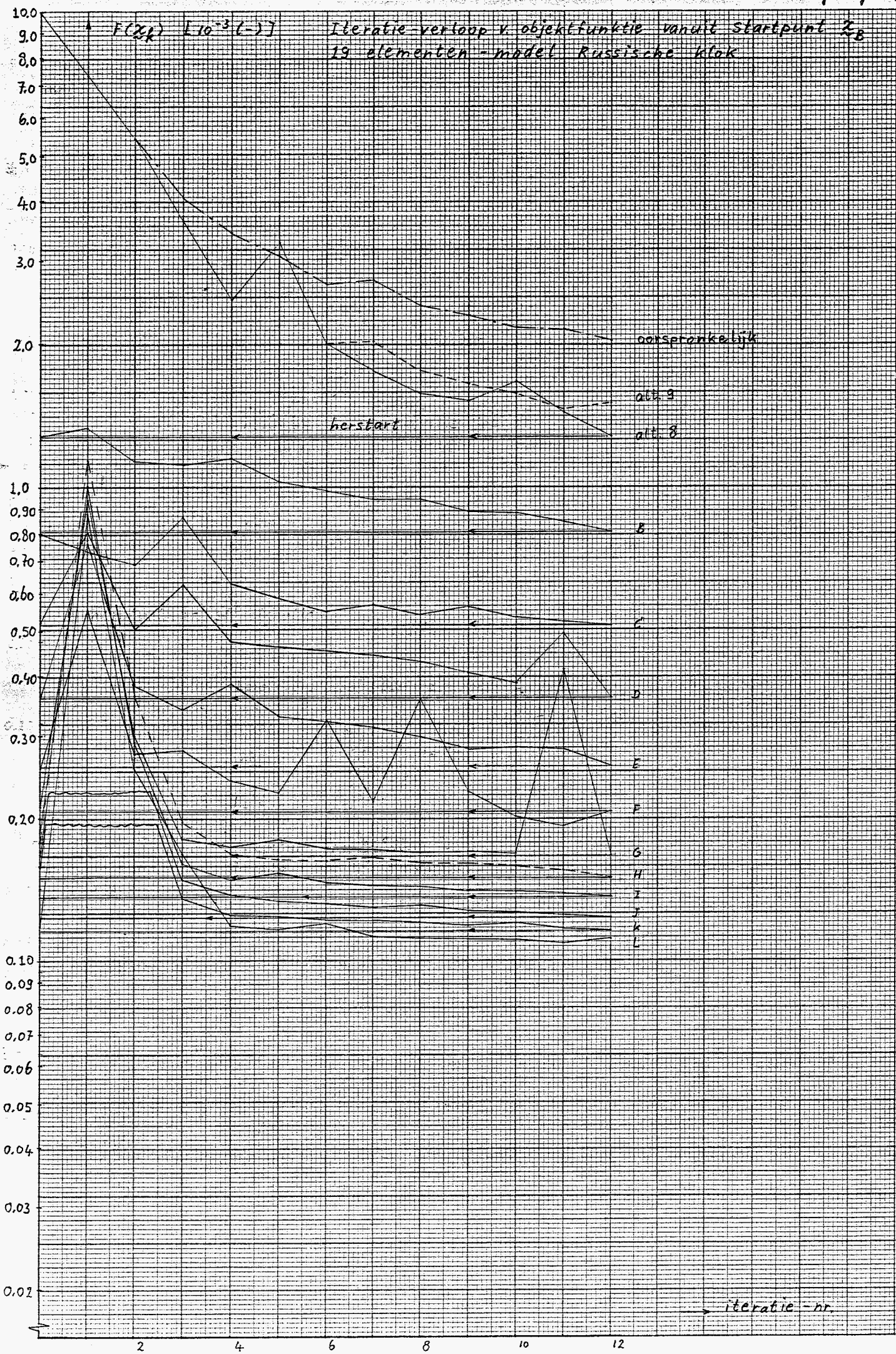
alt 8

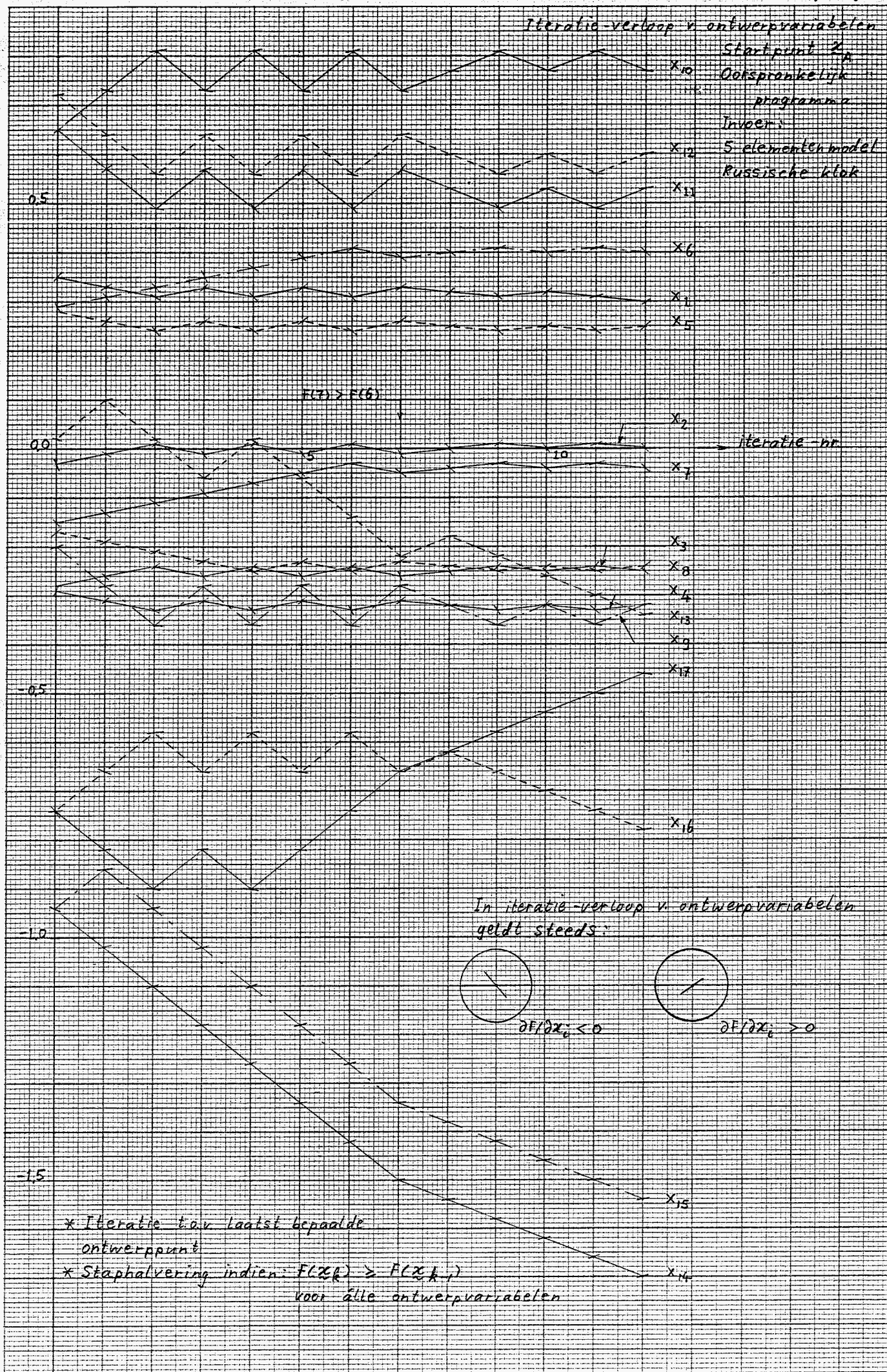
herstart alt 8 vanuit iteratiepunt 2/2

iteratie nr

0 2 4 6 8 10 12



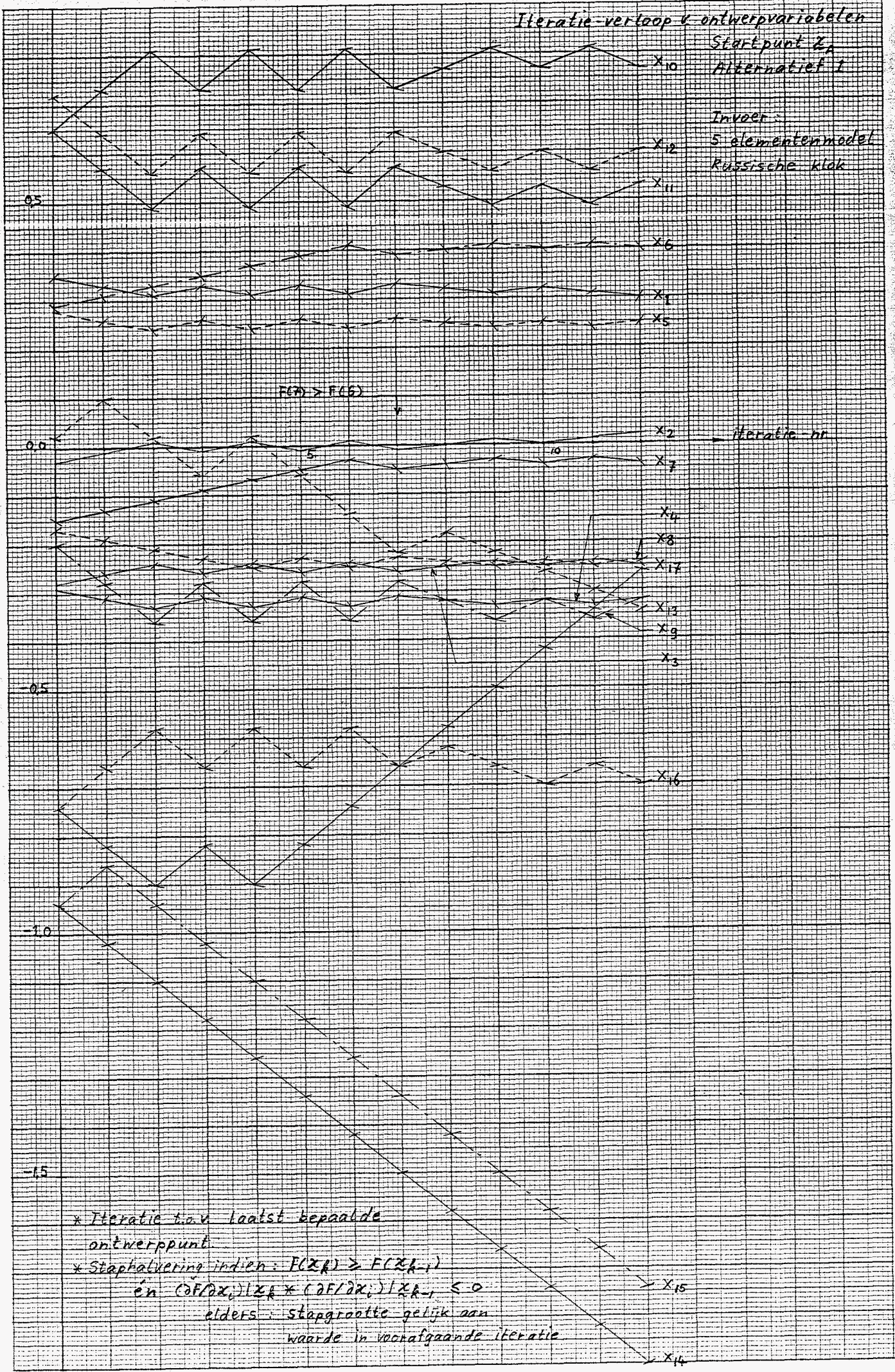




Iteratie-verloop v. ontwerpvariabelen

Startpunt Z_A
Alternatief 1

Invoer:
5 elementenmodel
Russische klok

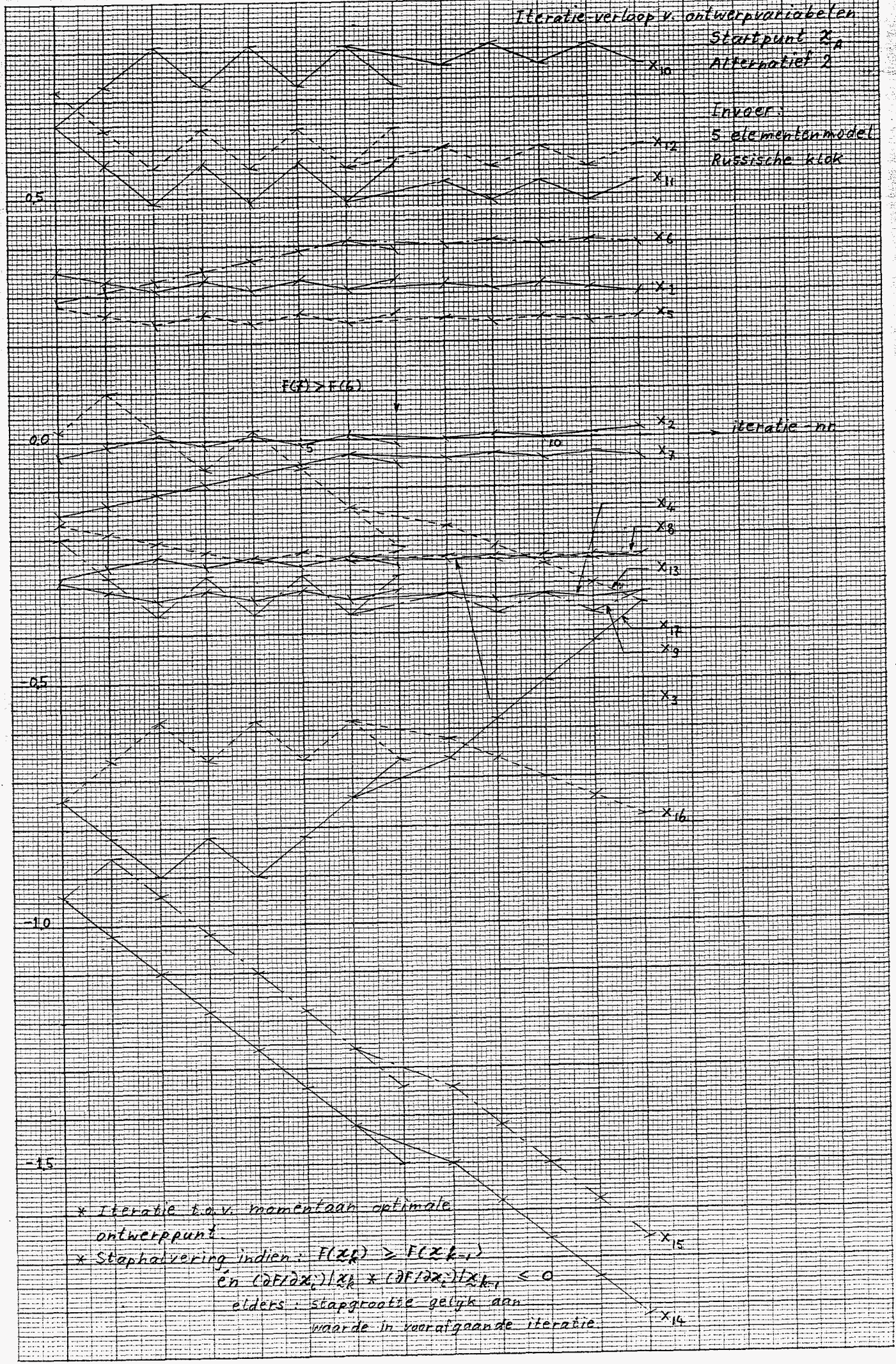


* Iteratie t.o.v. laatst bepaalde ontwerppunt
 * Staphalvering indien: $F(x_k) > F(x_{k-1})$
 en $(\partial F / \partial x_i) | x_k \times (\partial F / \partial x_i) | x_{k-1} < 0$
 elders: stapgrootte gelijk aan waarde in voorafgaande iteratie

Iteratie-verloop v. ontwerpvariabelen

Startpunt Z_0
 Alternatief 2

Invoer:
 5 elementenmodel
 Russische klok



$F(5) > F(6)$

iteratie-nr

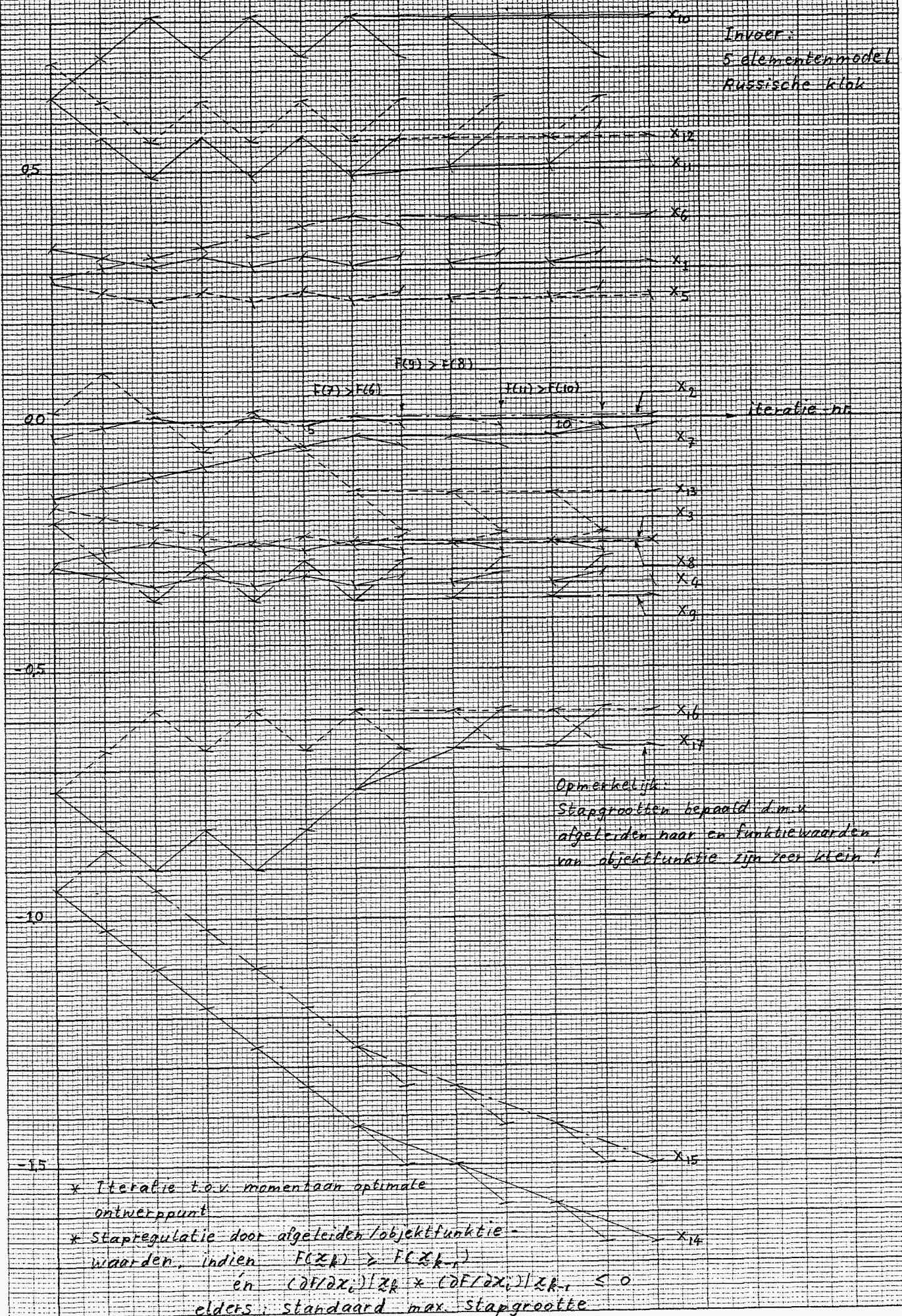
- * Iteratie t.o.v. momentaan optimale ontwerppunt
- * Staphalvering indien: $F(Z_k) \geq F(Z_{k-1})$
 en $(\partial F / \partial x_i) / x_k \times (\partial F / \partial x_i) / x_{k-1} \leq 0$
 elders: stapgrootte gelijk aan waarde in voorafgaande iteratie

1.0

Iteratie verloop v ontwerpvariabelen

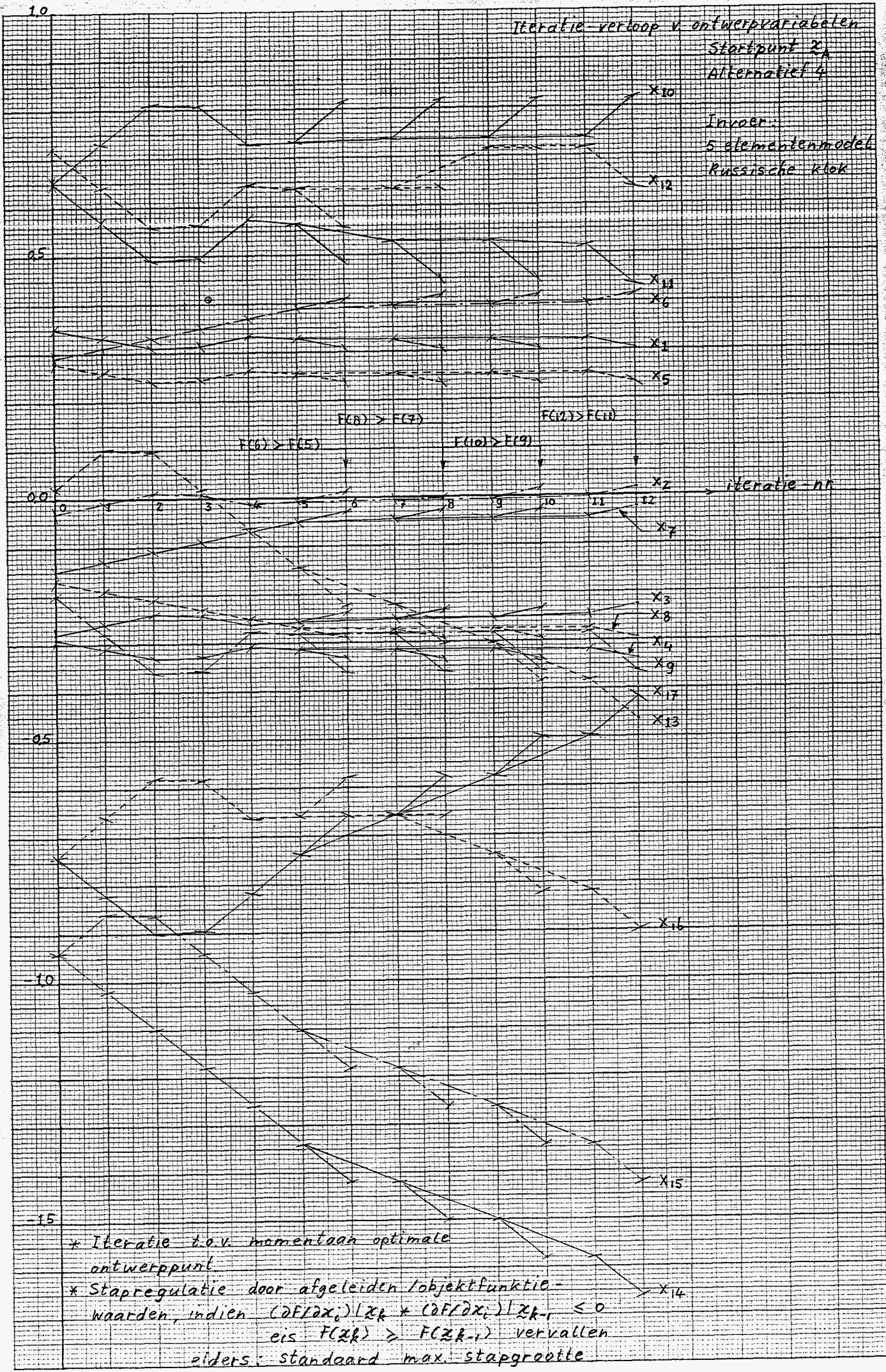
Startpunt Z_0
 Alternatief 3

Invoer:
 5 elementenmodel
 Russische klok



Opmerkelijk:
 Stapgrootten bepaald d.m.v.
 afgeleiden naar en functiewaarden
 van objektfunctie zijn zeer klein!

* Iteratie t.o.v. momentaan optimale ontwerppunt
 * Stapgrootte door afgeleiden/objectfunctie-waarden, indien $F(Z_k) > F(Z_{k-1})$
 én $(\partial F/\partial x_i)|_{Z_k} + (\partial F/\partial x_i)|_{Z_{k-1}} \leq 0$
 elders: standaard max. stapgrootte



10

Iteratie-verloop v. ontwerpvariabelen

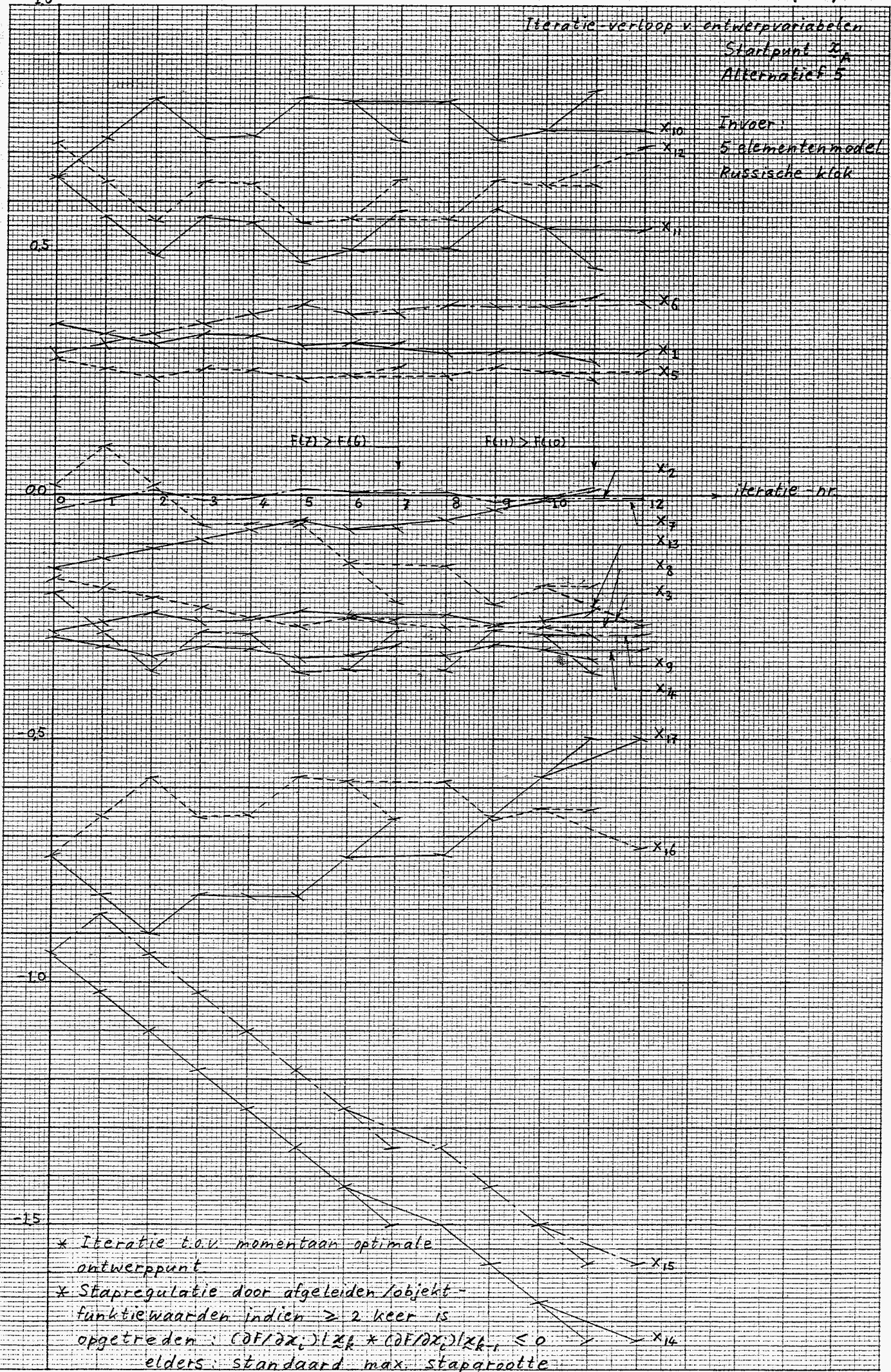
Startpunt x_0

Alternatief 5

Invoer:

5 elementenmodel

Russische klok



* Iteratie t.o.v. momentaan optimale ontwerp punt

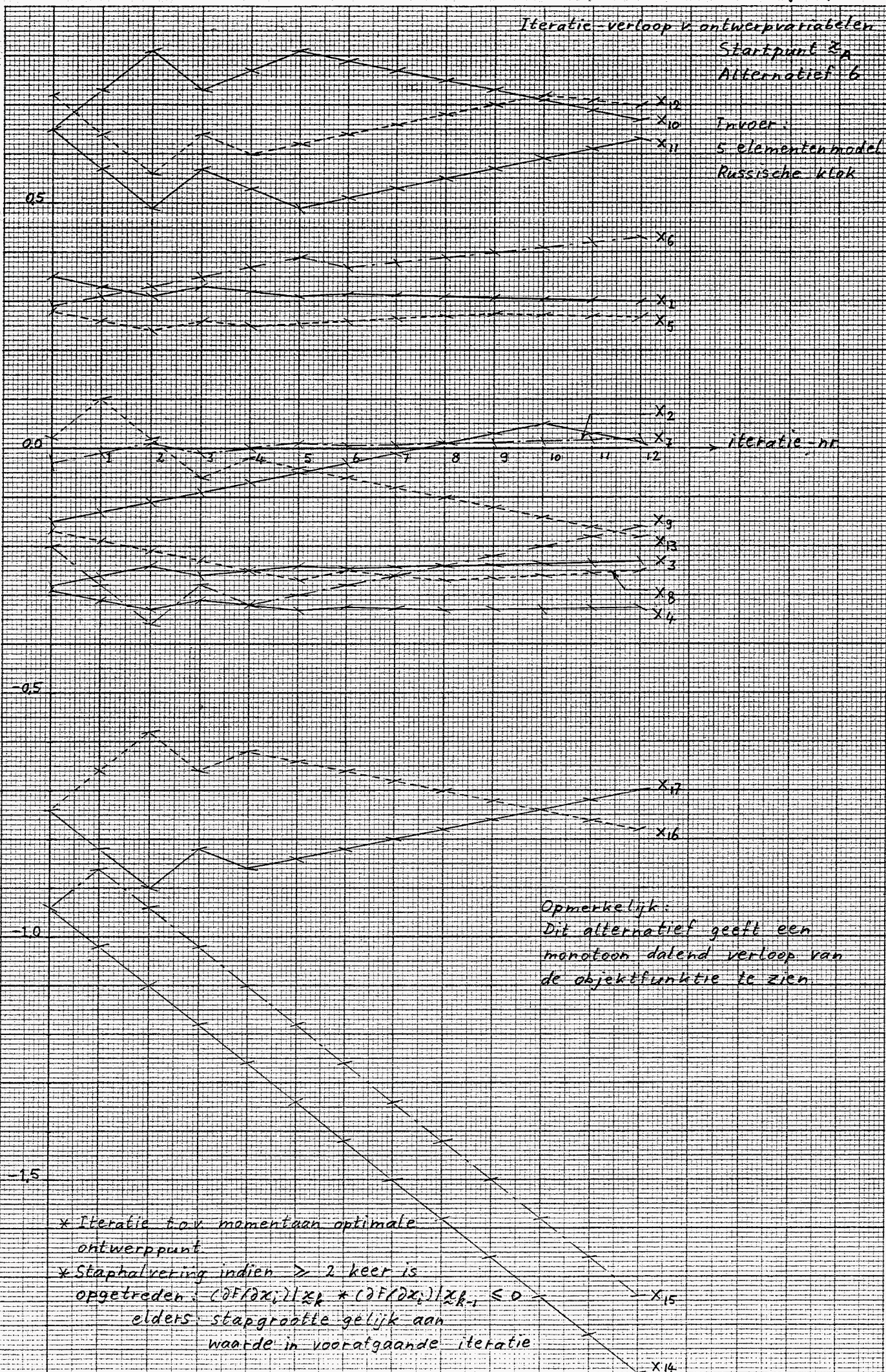
* Stapregulatie door afgeleiden / object-functiewaarden indien ≥ 2 keer is opgetreden: $(\partial F / \partial x_i) / x_k \times (\partial F / \partial x_i) / x_{k-1} \leq 0$

elders: standaard max. stapgrootte

Iteratic-verloop v ontwerpvariabelen

Startpunt x_0
 Alternatief 6

Invoer:
 5 elementenmodel
 Russische klok



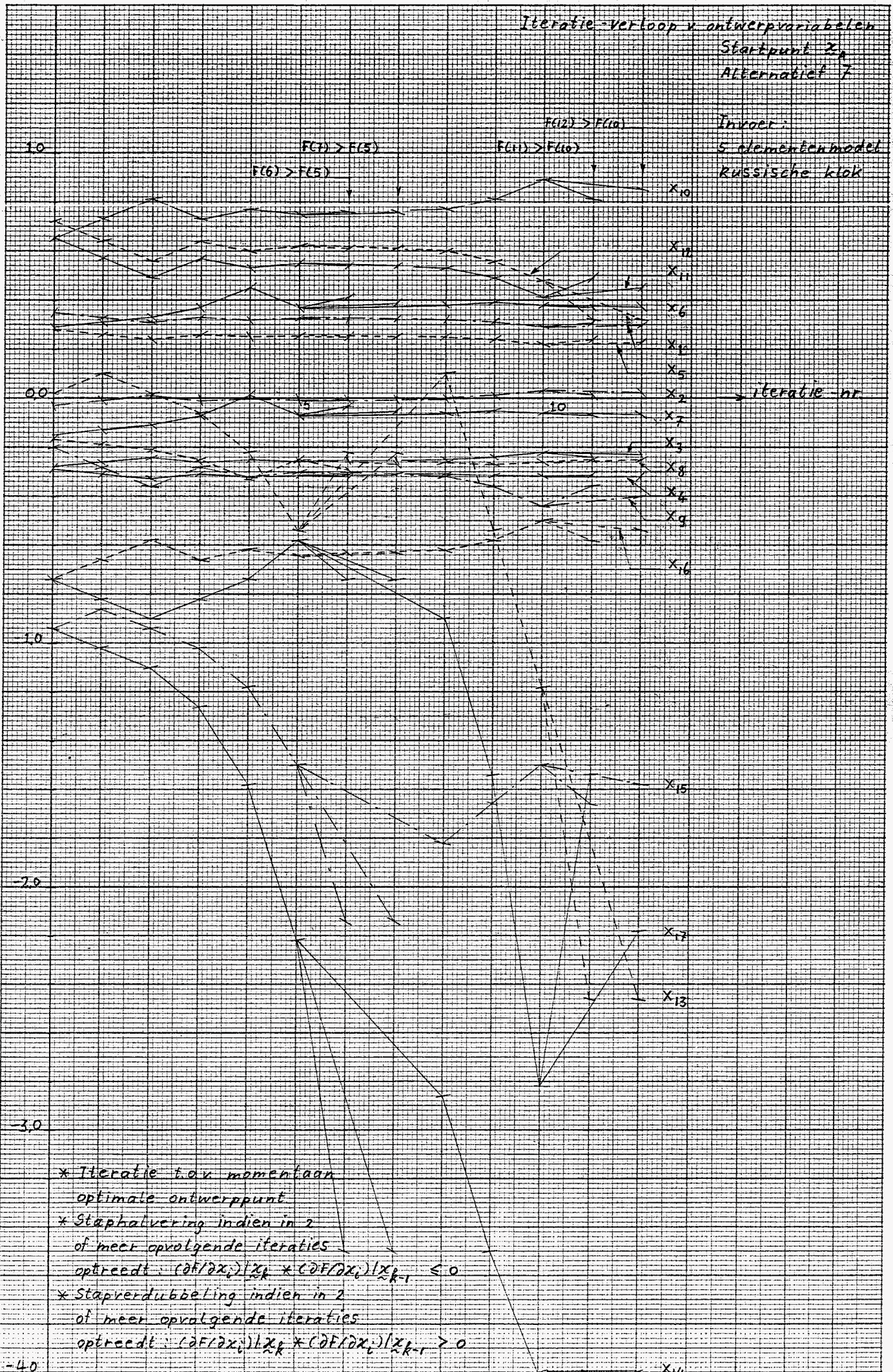
Opmerkelijk:
 Dit alternatief geeft een
 monotoon dalend verloop van
 de objektfunctie te zien.

* Iteratie tot momentaan optimale
 ontwerppunt
 * Staphalvering indien > 2 keer is
 opgetreden: $(\partial F / \partial x_i) | x_k * (\partial F / \partial x_i) | x_{k-1} \leq 0$
 elders: stapgrootte gelijk aan
 waarde in voorafgaande iteratie

Iteratie-verloop v. ontwerpvariabelen

Startpunt x_0
 Alternatief 7

Invoer:
 5 elementenmodel
 Russische klok

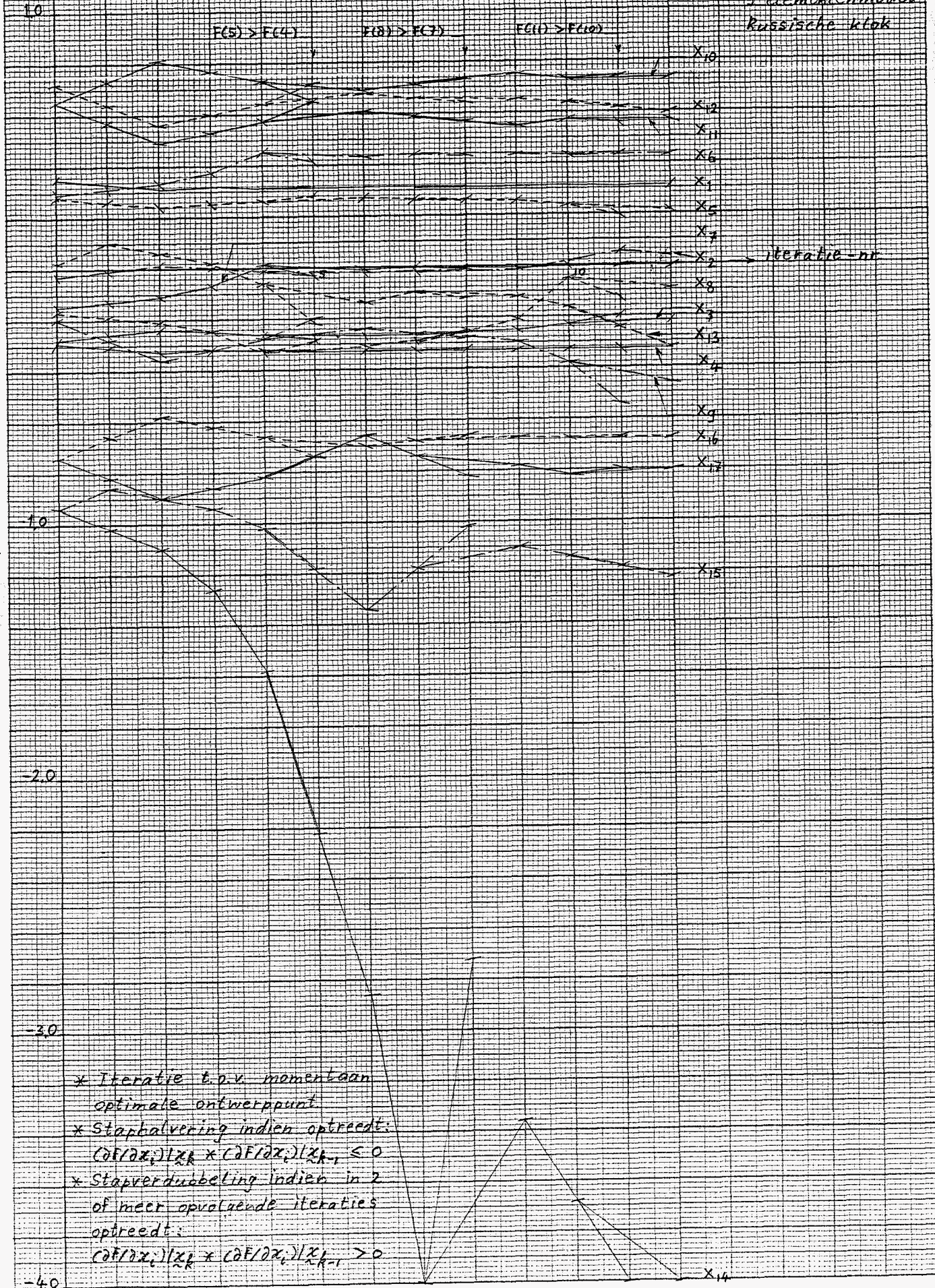


- * Iteratie t.o.v. momentaan optimale ontwerppunt
- * Staphalvering indien in z of meer opeenvolgende iteraties optreedt: $(\partial F / \partial x_i) | x_{R-1} \times (\partial F / \partial x_i) | x_{R-1} \leq 0$
- * Stapverdubbeling indien in z of meer opeenvolgende iteraties optreedt: $(\partial F / \partial x_i) | x_{R-1} \times (\partial F / \partial x_i) | x_{R-1} > 0$

Iteratie-verloop v ontwerpvariabelen

Startpunt Z_0
 Alternatief 8

Invoer:
 5 elementenmodel
 Russische klok



- * Iteratie t.o.v. momentaan optimale ontwerppunt
- * Staphalvering indien optreedt:
 $(\partial F / \partial x_i) | x_k \times (\partial F / \partial x_i) | x_{k-1} \leq 0$
- * Stapherdubbeling indien in 2 of meer opeenvolgende iteraties optreedt:
 $(\partial F / \partial x_i) | x_k \times (\partial F / \partial x_i) | x_{k-1} > 0$

Iteratie-verloop v. ontwerpvariabelen

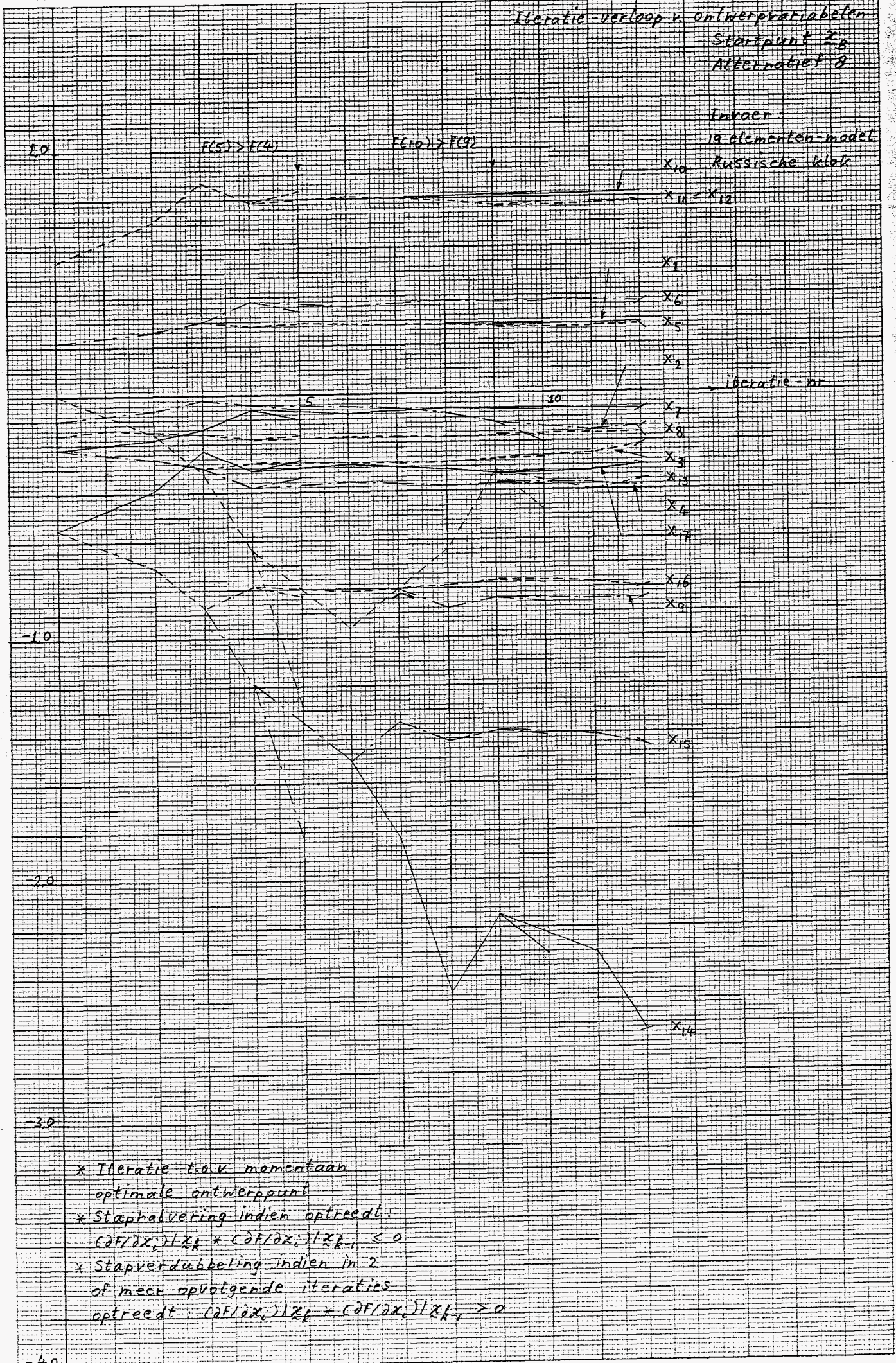
Startpunt Z_0

Alternatief 2

Invoer:

18 elementen-model

Russische klok



- * Iteratie t.o.v. momentaan optimale ontwerppunt
- * Staphalvering indien optreedt:
 $(\partial F / \partial x_i) | Z_k \times (\partial F / \partial x_i) | Z_{k-1} < 0$
- * Staphalvering indien in 2 of meer opeenvolgende iteraties optreedt:
 $(\partial F / \partial x_i) | Z_k \times (\partial F / \partial x_i) | Z_{k-1} > 0$

Objektfunctie-waarden in 10^{-3} (-) bij het iteratie-verloop
5 elementen-model ; startpunt Z_A .

	Oorspronkelijk	Alternatief 1	Alternatief 2	Alternatief 3	Alternatief 4
F_0	7,29003	7,29003	7,29003	7,29003	7,29003
F_1	5,19714	5,19714	5,19714	5,19714	5,19714
F_2	4,54868	4,54868	4,54868	4,54868	4,64910
F_3	4,41762	4,41762	4,41762	4,41762	4,34734
F_4	4,11954	4,11954	4,11954	4,11954	4,15007
F_5	3,97599	3,97599	3,97599	3,97599	3,89558
F_6	3,81938	3,81938	3,81938	3,81938	3,42332
F_7	3,91728	3,91728	3,91728	3,91728	3,72480
F_8	3,59704	3,56946	3,62608	3,66246	3,95753
F_9	3,56936	3,49241	3,56936	4,00882	3,64637
F_{10}	3,52982	3,46370	3,50676	3,57219	3,72092
F_{11}	3,44280	3,33518	3,37833	3,88119	3,54332
F_{12}	3,42389	3,27205	3,31218	3,49421	3,65455

	Alternatief 5	Alternatief 6	Alternatief 7	Alternatief 8	Alternatief 9
F_0	7,29003	7,29003	7,29003	7,29003	7,29003
F_1	5,19714	5,19714	5,19714	5,19714	5,19714
F_2	4,54868	4,54868	4,54868	4,59768	4,59768
F_3	4,41762	4,41762	4,14112	4,07752	4,07752
F_4	4,07541	3,98534	3,78487	3,79438	3,79438
F_5	4,02887	3,88425	3,70190	3,96043	3,96043
F_6	3,80081	3,74506	4,27354	3,42323	3,57909
F_7	4,02314	3,61468	3,96856	3,28355	3,27546
F_8	3,69489	3,50808	3,24633	3,33656	3,12999
F_9	3,68817	3,41928	2,94332	3,12530	2,93212
F_{10}	3,48462	3,34487	2,68134	2,99051	2,64558
F_{11}	3,72601	3,27577	3,31279	3,12971	2,68030
F_{12}	3,41411	3,21840	2,89163	2,83617	3,34239

Objektfunctie-waarden in $[10^{-3} (-)]$ bij het iteratie-verloop.
5 elementen-model ; startpunt x_B

	Oorspronkelijk	Alternatief 1	Alternatief 2	Alternatief 3	Alternatief 4
F_0	7,91025	7,91025	7,91025	7,91025	7,91025
F_1	6,73749	6,73749	6,73749	6,73749	6,73749
F_2	5,82006	5,82006	5,82006	5,82006	5,84200
F_3	5,14210	5,14210	5,14210	5,14210	5,15140
F_4	5,05752	5,05752	5,05752	5,05752	4,72920
F_5	4,54141	4,54141	4,54141	4,54141	4,42460
F_6	4,46055	4,46055	4,46055	4,46055	4,23770
F_7	4,29600	4,29600	4,29600	4,29600	4,40040
F_8	4,14933	4,14933	4,14933	4,14933	4,14640
F_9	4,22002	4,22002	4,22002	4,22002	4,55810
F_{10}	3,86360	3,82760	3,90234	3,92268	4,08280
F_{11}	3,81100	3,71950	3,73053	3,83324	4,51640
F_{12}	3,69300	3,60420	3,61947	3,80201	4,02700

	Alternatief 5	Alternatief 6	Alternatief 7	Alternatief 8	Herstart B
F_0	7,91025	7,91025	7,91025	7,91025	2,02167
F_1	6,73680	6,73680	6,73680	6,73680	2,30567
F_2	5,81983	5,81983	5,81983	5,83100	2,00573
F_3	5,14073	5,14073	4,70107	4,79112	1,93577
F_4	4,76085	4,81770	7,26213	4,37773	1,90042
F_5	4,34302	4,33762	5,33725	3,71075	1,84773
F_6	4,33978	4,17473	3,56499	3,32517	1,81086
F_7	4,01679	4,00003	4,42001	3,02708	1,75834
F_8	4,01718	3,84466	3,58615	2,82565	1,66071
F_9	3,78690	3,69608	3,19476	2,65302	1,57530
F_{10}	3,58983	3,55342	26,92270	2,36524	10,67750
F_{11}	3,55091	3,42185	2,45808	2,23145	3,48857
F_{12}	3,47495	3,30101	5,53602	2,01807	1,89977

Objektfunctie-waarden in $[10^{-3}(-)]$ bij het iteratie-verloop.
19 elementen-model ; startpunt \bar{x}_B .

	Oorspronkelijk	Alternatief 9	Alternatief 8	Herstart B	Herstart C	Herstart D	Herstart E
F_0	9,98973	9,98973	9,98973	1,27732	0,79636	0,51722	0,36147
F_1	7,34443	7,34443	7,34443	1,33409	0,73084	0,80634	0,76055
F_2	5,38106	5,39778	5,39778	1,13457	0,68948	0,50144	0,38273
F_3	4,07236	3,63275	3,63275	1,11602	0,86718	0,62490	0,34013
F_4	3,42936	2,47568	2,47568	1,15182	0,62717	0,47160	0,38628
F_5	3,07938	3,27506	3,27506	1,02393	0,58558	0,46058	0,32916
F_6	2,67782	2,00049	2,00049	0,98195	0,54951	0,45197	0,31983
F_7	2,73487	2,01884	1,76228	0,94211	0,56431	0,44240	0,31149
F_8	2,40662	1,76228	1,58191	0,94444	0,53845	0,42780	0,29751
F_9	2,29629	1,65461	1,52422	0,88853	0,55887	0,40553	0,27992
F_{10}	2,18124	1,58177	1,66969	0,88393	0,53078	0,38640	0,28192
F_{11}	2,14665	1,46345	1,43803	0,84976	0,52096	0,49123	0,28085
F_{12}	2,03820	1,50581	1,27723	0,80586	0,51269	0,35958	0,25720

	Herstart F	Herstart G	Herstart H	Herstart I	Herstart J	Herstart K	Herstart L
F_0	0,25494	0,21254	0,17852	0,18635	0,16656	0,16194	0,11853
F_1	0,55162	0,94209	1,14100	1,00735	1,00308	1,00050	0,85908
F_2	0,27450	0,29724	0,36287	0,29020	0,28663	0,26788	0,25589
F_3	0,27776	0,18122	0,19555	0,16054	0,14840	0,13499	0,16726
F_4	0,24021	0,17402	0,16824	0,14830	0,13798	0,12462	0,11862
F_5	0,22628	0,18016	0,16455	0,15356	0,13420	0,12381	0,11629
F_6	0,32385	0,17312	0,16259	0,14664	0,13186	0,12199	0,12012
F_7	0,21602	0,17211	0,16512	0,14449	0,12960	0,12156	0,11226
F_8	0,35857	0,16976	0,16135	0,14400	0,13120	0,12039	0,11192
F_9	0,22760	0,17008	0,16026	0,14110	0,12707	0,11868	0,11121
F_{10}	0,20085	0,16891	0,15855	0,13958	0,12624	0,11998	0,11033
F_{11}	0,19288	0,41242	0,15559	0,13837	0,12514	0,11729	0,10916
F_{12}	0,20554	0,16678	0,15043	0,13610	0,12370	0,11592	0,11154

Eouten in het oorspronkelijke programma van DYNOPT

```
(1 ) SUBROUTINE INITI 1 (LREA, LINT)
(2 ) INTEGER LREA, LINT
      |
(81 ) KFREL = KYTUN + NSECTO * NTUNFR      moet zijn:
(82 ) IF (NOPTTO .GT. 0) KLASTR = KFREL + NOPTFR } KLASTR = KFREL + NOPTFR
(83 ) KLASTR = KFREL + NOPTTO
      |
(111 ) RETURN
(112 ) END
```

In de invoerfile 'Russische klok' geldt: Aantal te optimaliseren frequentie-verhoudingen = $NOPTFR = 7$; Nummer van de referentie-frequentie = $NOPTTO = 5$. In regel (83) worden voor het array-gedeelte waarin de frequentie-verhoudingen op te slaan slechts 5 plaatsen gereserveerd, i.p.v. benodigde 7 stuks. In alt. 1 en alt 2. verder daardoor voor alle iteraties x_1 en x_2 verkeerd berekend, bij een eerste test-run.

```
(1 ) SUBROUTINE DFUDVA (DFU, WG, FREQ, FREQOP, DFDVAR, NDESVA, NFR, NTO)
(2 ) INTEGER NDESVA, NFR, NTO
      |
(46 ) IF (NTO .GT. 0) GOTO 30 } regel 47 moet vóór regel 46
(47 ) CALL ZERORV (DFU, NDESVA) staan!
      |
      |
(60 ) RETURN      DFU  $\triangleq$  REA (KDFEMDV): array met afgeleiden v.
      |      |      |      |      |      |      |      |      |      |
      |      |      |      |      |      |      |      |      |      |
(61 ) END      DFU maakt array-inhoud nul. objectfunctie naar ont-
                werpvariabelen.
```

Omdat $NTO \triangleq NOPTTO = 5$ groter dan 0 wordt regel (47) nooit bereikt. In eerste iteratie is inhoud nul van DFU; daarna nooit meer gewist maar nieuwe waarden bij oude opgeteld.