

MASTER

Uitbreiding van een software-pakket voor elektrische veldproblemen

van Dorsselaer, Patrick E.G.

Award date:
1987

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

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

TECHNISCHE UNIVERSITEIT EINDHOVEN

Faculteit Wiskunde en Informatica

AFSTUDEERVERSLAG

Uitbreiding van een software-pakket
voor elektrische veldproblemen

door

Patrick E. G. van Dorsselaer

afstudeerdocent:

prof. dr. M. Rem

begeleider:

ir. H.J.F. Kloth

Juni 1987

VOORWOORD

Het voor u liggende verslag is het resultaat van het afstudeerproject dat ik heb uitgevoerd in de periode november 1986 tot en met juni 1987 bij de vakgroep EHO van de Faculteit Elektrotechniek van de Technische Universiteit Eindhoven.

Ik zou graag de volgende personen willen bedanken voor alle ondersteuning bij mijn afstuderen:

ir. H. Kloth en prof. dr. M. Rem.

SAMENVATTING

De vakgroep EHO beschikt over een software-pakket dat met m. b. v. de eindige elementen methode elektrische velden en potentialen kan berekenen in diverse ruimtelijke configuraties van dielectrische media. Aan dit pakket is een programma toegevoegd zodat het mogelijk is om het oplaadproces van deeltjes in een corona omgeving te simuleren.

Eindhoven, juni 1987

Patrick van Dorsselaer

INHOUDSOPGAVE

	pag.
0 INLEIDING	2
1 ELECTR/INTERFACE	4
1.1 BESCHRIJVING TRIQUAMESH	4
1.2 BESCHRIJVING ELECTR/INTERFACE	5
1.3 AANPASSINGEN AAN ELECTR/INTERFACE	7
1.3.1 VERANDERING AAN "KOPPELEN" EN "ZWEVENDE ELEKTRODE"	7
1.3.2 VERANDERING AAN "VOORGESCHREVEN RANDSPANNINGEN"	10
1.3.3 TOEVOEGING "COORDINAATAANPASSING"	11
2 DE REPETITIE	12
2.1 DE ELECTRICISCHE THEORIE	13
2.2 HET NIEUWE PROGRAMMA	15
2.2.1 ABSTRACTE BESCHRIJVING ELECTR/REPFEMSYS	16
2.2.2 BEPALING GEBIEDEN	18
2.2.3 TIJDSTIPPEN	21
2.2.4 DE INVOERGEGEVENS	22
2.2.5 GEBRUIK VAN ELECTR/REPFEMSYS	25
2.2.6 GLOBAAL OVERZICHT ELECTR/REPFEMSYS	27
3 ELECTR/INTERFACE/NI	30
4 VOORBEELDEN	34
4.1 DE GELEIDENDE BOL	35
4.2 DE NIET-GELEIDENDE BOL	41
4.3 CONCLUSIE	46
LITERATUUR	47
APPENDICES:	48
A1 AFLEIDING FORMULE OPPERVLAKTE GEBIED	48
A2 AFLEIDING FORMULE VOLUME GEBIED	49
A3 FORMULES VOOR ELECTRICISCHE VELD	50
A4 AFLEIDING FORMULE HOEK ALFA	51
A5 VOORBEELD FILE DATAMG	53
A6 VOORBEELD INTERFACE DIALOOG MET ELECTR/INTERFACE	54
A7 VOORBEELD FILE INTERF	57
A8 VOORBEELD FILE OUTREPFEM	58
A9 VOORBEELD FILE OUTER	59
A10 VOORBEELD INTERFACE DIALOOG MET USER/DRAW	60
A11 FILE'S DATAMG EN INTERF VOOR GELEIDENDE BOL	63
A12 PROGRAMMATEKST ELECTR/REPFEMSYS	65
A13 PROGRAMMATEKST ELECTR/INTERFACE/NI	77

O INLEIDING

De vakgroep EHO van de Faculteit Electrotechniek (TUE) beschikt over een software-pakket dat m.b.v. de eindige elementen methode elektrische velden en potentialen kan berekenen in diverse ruimtelijke configuraties van dielectrische objecten. Het in dit verslag beschreven project bestond uit het gebruikersvriendelijk maken van dit software-pakket.

Bovendien moest het mogelijk worden de programmatuur iteratief te laten werken zonder steeds opnieuw gegevens met de hand te hoeven invoeren.

Het software-pakket is alleen geschikt voor 2-dimensionale constructies en voor 3-dimensionale constructies met één symmetrie-as.

Het software-pakket bestond uit een aantal programma's, namelijk:

- TRIGUAMESH : Voor het vastleggen van de constructie en het verdelen van de constructie in elementen (zoals driehoekjes).
- ELECTR/INTERFACE : Voor het aanvullen tot een volledige invoer voor het hoofdprogramma (FEMSYS). Dit houdt in dat er allerlei randvoorwaarden kunnen worden ingevoerd (zoals randspanningen en de materiaalconstanten). Dit programma is alleen geschikt voor puur capacitieve of puur resistieve veldproblemen.
- FEMSYS : Voor het uitvoeren van de berekeningen. Een algemeen programma dat op de eindige elementen methode gebaseerde berekeningen kan uitvoeren. Om het geschikt te maken voor elektrische veldproblemen is er voor 3-dimensionale, rotatie-symmetrische problemen het programma ELECTR/ROTELEMENT en voor 2-dimensionale problemen het programma ELECTR/ELEMENT. Deze programma's koppelen de specifieke elektrische elementvergelijkingen aan FEMSYS. In deze programma's wordt er van uitgegaan dat de elementen driehoekjes zijn. De elementvergelijkingen zijn in [KOE83] afgeleid.
- ELECTR/ROTELEMENT: Start het programma FEMSYS (3-dimensionale, rotatie-symmetrische problemen).
- ELECTR/ELEMENT : Start het programma FEMSYS (2-dimensionale problemen).
- USER/DRAW : Voor het maken van grafische uitvoer.

Het programma ELECTR/INTERFACE was niet altijd even

gebruikersvriendelijk. Daarom is dit programma op een aantal punten aangepast. Deze aanpassingen worden in hoofdstuk 1 besproken.

Het invoeren van de randvoorwaarden door het interactieve programma ELECTR/INTERFACE kost nogal veel tijd. Daarom is er een niet-interactieve versie van dit programma gemaakt (ELECTR/INTERFACE/NI). Deze versie wordt in hoofdstuk 3 besproken.

De iteratie bestaat uit een aantal eindige elementen berekeningen (FEMSYS), waarbij een aantal parameters per stap van de iteratie kunnen veranderen. In dit afstudeerverslag is alleen de mogelijkheid om de "ladingsdichtheid" als parameter te kiezen uitgewerkt. Aan de software is een nieuw programma, dat de iteratie bevat, toegevoegd, genaamd ELECTR/REPFEMSYS. In hoofdstuk 2 wordt de iteratie besproken.

In hoofdstuk 4 worden twee voorbeelden uitgewerkt.

Aan het eind van het verslag zijn een aantal appendices opgenomen. De appendices A1 t/m A4 bevatten de afleidingen van een aantal formules. De appendices A5 t/m A11 bevatten een aantal in- en uitvoer files van de in hoofdstuk 4 besproken voorbeelden. Appendix A12 bevat de programmatekst van ELECTR/REPFEMSYS. Appendix A13 bevat de programmatekst van ELECTR/INTERFACE/NI.

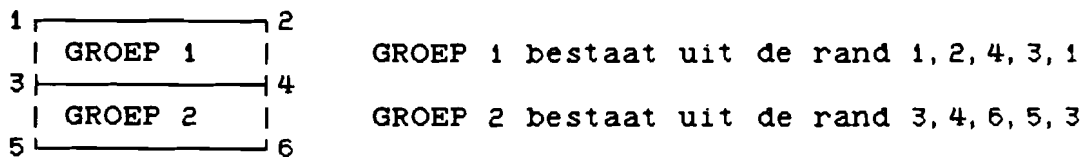
1 ELECTR/INTERFACE

In dit hoofdstuk worden de aanpassingen aan ELECTR/INTERFACE uitgelegd. Eerst wordt een beschrijving van de programma's TRIQUAMESH en ELECTR/INTERFACE gegeven.

1.1 BESCHRIJVING TRIQUAMESH

Het programma TRIQUAMESH is een programma waarmee een elementenverdeling kan worden gemaakt en waarmee de constructie kan worden vastgelegd. Het programma verwacht de gegevens over de constructie in een file DATAMG. De uitvoer komt in een file MESH, genaamd .../SAFERESULTS, te staan. Een constructie bestaat uit een aantal groepen d.w.z. de delen van de totale structuur met dezelfde materiaaleigenschappen. Een groep wordt door een rand gevormd. Een rand kan met behulp van een aantal basispunten worden vastgelegd.

vb:



De punten 1, 2, 3, 4, 5 en 6 zijn basispunten.

Eerst worden op de randen de knooppunten gegenereerd en daarna worden, uitgaande van deze knooppunten, de groepen in elementen verdeeld.

Op een snijlijn van twee groepen wordt, afhankelijk van de invoer, door TRIQUAMESH een dubbele serie samenvallende knooppunten of één serie knooppunten aangemaakt. Men moet de invoer zo maken dat er een dubbele serie knooppunten wordt aangemaakt omdat het anders problemen oplevert.

De invoergegevens bestaan uit:

- structuurnaam, kan door de gebruiker gekozen worden.
- coördinaten van de basispunten.
- contourdelen d.w.z. gerichte delen waaruit de randen opgebouwd worden. De delen kunnen beschreven worden met rechte lijnstukken tussen twee basispunten of met cirkelbogen tussen twee basispunten.
- subgebieden, deze worden beschreven met behulp van de contourdelen. Indien een zelfde contourdeel meerdere keren wordt doorlopen, genereert triquamesh slechts één serie knooppunten.
- substructuren d.w.z. delen opgebouwd uit subgebieden van gelijksoortige elementtypes (bv driehoekjes), die tesamen de totale structuur vormen.

- de grofheid van de elementen.
- een evt. hernummering van de knooppunten.

Opm: In de aangemaakte uitvoerfile is niets meer over de basispunten bekend.

Voor meer gegevens over TRIQUAMESH en de file datamg zie [BEU83] of [KOE83].

1.2 BESCHRIJVING ELECTR/INTERFACE

Het programma ELECTR/INTERFACE is een interactief programma waarmee randvoorwaarden kunnen worden ingevoerd. Dit programma levert de file INREADFEM op, waarop FEMSYS kan worden toegepast. Voor dit programma moet de file MESH en de file DATAMG bekend zijn. Deze laatste is nodig omdat een gebruiker dan in termen van de basispunten over de constructie kan blijven praten.

Wat moet men als gebruiker allemaal invoeren:

- de naam van de file datamg.
- de naam van de file mesh.
- bovenschatting voor het aantal knooppunten.
- bovenschatting voor het aantal elementen.
- de karakteristieken per groep.

Ieder groep heeft twee karakteristieken. Indien het probleem puur capacitief is dan is de eerste karakteristiek de permittiviteit en de tweede karakteristiek de ladingsdichtheid. Indien het probleem puur resistief is dan is de eerste karakteristiek de conductiviteit en de tweede karakteristiek krijgt automatisch de waarde nul (zie [KOE83]).

- evt. "randelementen"
(is niet bekeken, wordt geen gebruik van gemaakt).

- evt. "koppelen":

Op de snijlijn van twee groepen is het mogelijk dat TRIQUAMESH daar een dubbele serie samenvallende knooppunten heeft gegenereerd. Indien dit het geval is moeten deze samenvallende knooppunten aan elkaar gelijkgesteld worden. Als men dit niet doet dan worden deze groepen als onafhankelijk beschouwd en dat is niet de bedoeling.

- evt. "zwevende elektrode":

Een zwevende elektrode is een deelrand met overal op deze deelrand eenzelfde potentiaal.

- evt. "voorgescreven randspanningen".

- evt. "geaarde deelranden"

- de mogelijkheid dat er plots moeten worden gemaakt.

ELECTR/INTERFACE bevat een stukje programma dat per groep bepaalt uit welke knooppunten de rand bestaat. Indien bij een snijlijn van twee subgebieden maar één serie knooppunten is gegenereerd dan wordt deze snijlijn niet als een gedeelte van

de rand beschouwd.

In ELECTR/INTERFACE geeft men een gedeelte van de rand aan met een viertal getallen (n_1 , n_2 , n_3 , n_4). Dit is de deelrand die behoort tot groep n_1 en loopt van basispunt n_2 via basispunt n_3 naar basispunt n_4 . Indien n_2 is n_3 dan is het de kleinste rand van n_2 naar n_4 en indien n_3 is n_4 dan is het de grootste rand van n_2 naar n_4 . Dit laatste kan men beter zoveel mogelijk vermijden omdat het wel eens niet eenduidig kan zijn.

Koppelen geeft men met een vijftal punten (n_1 , n_2 , n_3 , n_4 , n_5) aan d. w. z. de rand die behoort tot groep n_1 en loopt van n_3 via n_4 naar n_5 moet met de rand die behoort tot groep n_2 en loopt van n_3 via n_4 naar n_5 worden gekoppeld. De basispunten en groepen die ingevuld moeten worden zijn consistent met de basispunten en groepen uit de invoerfile DATAMG.

1.3 AANPASSINGEN AAN ELECTR/INTERFACE

In het programma ELECTR/INTERFACE zijn een aantal veranderingen aangebracht. Waaronder een aantal kleine veranderingen voor het uithalen van een aantal onvolkomenheden. De twee grote veranderingen en de toevoeging van de mogelijkheid om de coördinaten aan te passen worden hier besproken. Verder is er nog een groot stuk aan toegevoegd om de software geschikt te maken voor de repetitie.

In het verloop van het verslag wordt nogal eens gebruik gemaakt van een kontekstvrije grammatica voor het beschrijven van een taal. Eerst wordt uitgelegd wat een kontekstvrije grammatica inhoudt.

Een kontekstvrije grammatica G is een 4-tuple (N, T, P, S) :

- N is een eindige verzameling van hulpsymbolen (non-terminals)
- T is een eindige verzameling van eindsymbolen (terminals)
- $N \cap T = \emptyset$
- P is een eindige deelverzameling van $N^* (N^* T)$, de verzameling van produktieregels
- $S \in N$, het startsymbool

De taal $L(G)$ die gegenereerd wordt door de grammatica G is de verzameling van rijen eindsymbolen die ontstaan door vervanging van hulpsymbolen door symbolen in overeenstemming met de produktieregels, beginnende met het startsymbool.

1.3.1 VERANDERING AAN "KOPPELEN" EN "ZWEVENDE ELEKTRODE"

ELECTR/INTERFACE maakt de file INREADFEM aan. Deze file moet aan een aantal voorwaarden voldoen. Bij "Koppelen" en bij "zwevende elektrode" was het mogelijk dat er niet aan de voorwaarden werd voldaan. Het programma is zo aangepast dat in alle gevallen aan deze voorwaarden wordt voldaan.

Een voorbeeld van een situatie die een fout opleverde bij de verwerking van FEMSYS: De constructie bevat een punt dat bij meer dan twee groepen behoort.

Bij "koppelen" en bij "zwevende elektrode" geeft men paren van knooppunten aan die aan elkaar gelijkgesteld moeten worden. M.a.w. paren van punten die dezelfde uitkomst bij de veld-berekening moeten opleveren. (Of in termen van elektrische veldproblemen: de knooppunten moeten dezelfde potentiaal hebben).

Een gebruiker doet dit door een aantal deelranden aan te geven (zie 1.2). In het programma worden de knooppunten bepaald die aan elkaar gelijkgesteld moeten worden.

Deze knooppunten komen in INREADFEM in een zogenaamde transformatielijst te staan. D.w.z. een lijst waarin staat welke knooppunten bij de veld-berekening dezelfde uitkomst moeten opleveren.

In het vervolg van het verslag wordt aangenomen dat er maxnode knooppunten aangemaakt zijn.

De syntax van een transformatielijst ziet er als volgt uit:

GRAMMATICA:

De 'non-terminals' :

{transformatielijst, transformatiespecs, transformatiespec,
linkerdeel, rechterdeel, knooppuntnummer}

De 'terminals' :

{"[", "TRANSFORMATIONS", ":", "X", "]", "=", "0", "1", "2",
"3", "4", "5", "6", "7", "8", "9"}

Startsymbool : transformatielijst

Grammatica-regels :

```
transformatielijst ::= TRANSFORMATIONS transformatiespecs
transformatiespecs ::= transformatiespec ;
                        transformatiespecs | e
{e = "leeg"}
transformatiespec  ::= [ linkerdeel : X ] =
                        [ rechterdeel : X ]
linkerdeel         ::= knooppuntnummer
rechterdeel        ::= knooppuntnummer
knooppuntnummer ∈ [ 1..maxnode],
```

Semantiek: Een transformatiespec houdt in dat het linkerdeel en het rechterdeel dezelfde uitkomst bij de berekening moeten opleveren.

Definities:

Laat i een knooppuntnummer zijn dan:

- 1) $Nlt(i)$ = het aantal keer dat knooppunt i als linkerdeel in een transformatielijst voorkomt.
- 2) $Nrt(i)$ = het aantal keer dat knooppunt i als rechterdeel in een transformatielijst voorkomt.

Hiermee zijn de eisen als volgt te formuleren:

- 1) $(\Delta i: i = \text{knooppuntnummer}: Nlt(i) \leq 1)$
- 2) $(\Delta i: i = \text{knooppuntnummer}: (Nlt(i) = 1) \implies (Nrt(i) = 0))$

Definitie:

Laat a, b knooppuntnummers dan

3) $K(a, b)$ = punt a en punt moeten aan elkaar gelijkgesteld worden. {"koppel a met b"}

Indien x en y knooppunten zijn en x en y moesten gekoppeld worden ($K(x, y)$) dan werd dit in de transformatielijst met x als linkerdeel en y als rechterdeel opgeslagen. Hierbij werd niet gecontroleerd of aan de eisen werd voldaan.

voorbeeld:

1		2
x		
3		

bij positie x behoren drie knooppunten, namelijk:

groep 1 -> punt a

groep 2 -> punt b

groep 3 -> punt c

Het "koppelen" van de groepen levert o. a. de koppelingen:

$K(a, b)$, $K(b, c)$ en $k(a, c)$.

Dit betekent: de bij de berekening opgeleverde uitkomsten moeten in de punten a, b en c hetzelfde zijn.

Dit werd in transformatielijst opgeslagen als:

$[a : X] = [b : X]$

$[b : X] = [c : X]$

$[a : X] = [c : X]$

Nu is aan geen van de eisen voldaan.

Het zou bv moeten zijn

$[a : X] = [b : X]$

$[c : X] = [b : x]$

Bij het invoeren van de "koppelingen" door de gebruiker wordt door het programma bijgehouden welke knooppunten dezelfde uitkomst moeten opleveren. Nadat alle "koppelingen" ingevoerd zijn, worden ze in de transformatielijst verwerkt. De verwerking is zodanig dat aan de eisen wordt voldaan.

De verwerking is als volgt:

Laat Y de grootste verzameling knooppunten zijn die dezelfde uitkomst moeten opleveren,

laat nY het aantal elementen van Y zijn,

verwerking:

Voeg "ny-1" transformatiespecs aan de transformatielijst toe, namelijk een punt uit Y als rechterdeel en de andere punten als linkerdeel.

Voorbeeld:

$Y = \{a, b, c, d\}$

verwerking:

TRANSFORMATIONS

...
 $[a : X] = [d : X]$
 $[b : X] = [d : X]$
 $[c : X] = [d : X]$

1.3.2 VERANDERING AAN "VOORGESCHREVEN RANDSPANNINGEN"

De punten die een voorgeschreven spanning krijgen komen in een zogenaamde prescribelijst en loadlijst te staan. De eerste lijst bevat de knooppunten die een voorgeschreven potentiaal-waarde moeten krijgen. De tweede lijst bevat de knooppunten, die in een prescribelijst voorkomen, met hun bijbehorende spanning. Punten die op een gearde rand (voorgeschreven spanning is nul) komen in een zogenaamde suppresslijst te staan.

De syntax van deze lijsten is als volgt:

GRAMMATICA:

De 'non-terminals':

```
{s, prescribelijst, suppresslijst, loadlijst, prescribespecs, prescribespec, knooppuntnummer, real, loadspecs, loadspec, suppressspecs, suppressspec}
```

De 'terminals' :

```
{"[", "TRANSFORMATIONS", ":", "X", "]", "=", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", ".", "E", "@", "+", "-"}
```

Startsymbool : transformatielijst

Grammatica-regels :

```
s ::= prescribelijst suppresslijst loadlijst
prescribelijst ::= PRESCRIBE prescribespecs
prescribespecs ::= prescribespec prescribespecs | e
{e = "leeg"}
prescribespec ::= [ knooppuntnummer : X ]
loadlijst ::= LOAD L1 FOR S loadspecs
loadspecs ::= loadspec ; loadspecs | e
loadspec ::= [ knooppuntnummer : X ] real
suppresslijst ::= SUPPRESS suppressspecs
suppressspecs ::= suppressspec suppressspecs | e
suppressspec ::= [ knooppuntnummer : X ]
knooppuntnummer ∈ [1..maxnode]
real is een reëel getal.
```

Voorbeeld: Laat a,b twee verschillende knooppuntnummers en r een real zijn,

```
PRESCRIBE
.... [ a : X ] ....
SUPPRESS
.... [ b : X ] ....
LOAD L1 FOR S
.... [ a : X ] r; ...
```

Dit betekent dat a de spanning r en b de spanning 0 krijgt.

Definities:

Laat i een knooppuntnummer zijn dan:

- 4) $N_p(i)$ = aantal keren dat i voorkomt in prescribelijst,
- 5) $N_s(i)$ = aantal keren dat i voorkomt in suppresslijst,
- 6) $N_l(i)$ = aantal keren dat i voorkomt in loadlijst.

De eisen zijn nu als volgt te formuleren:

- 1) ($\Delta i: i = \text{knooppuntnummer} : N_p(i) \leq 1$ and $N_l(i) \leq 1$)
- 2) ($\Delta i: i = \text{knooppuntnummer} : N_s(i) \leq 1$)
- 3) ($\Delta i: i = \text{knooppuntnummer} : N_p(i) = 0$ or $N_s(i) = 0$)
- 4) ($\Delta i: i = \text{knooppuntnummer} : N_l(i) = 1 \Rightarrow N_p(i) = 0$ and
 $N_l(i) = 0$ and
 $N_s(i) = 0$)
- 5) ($\Delta i: i = \text{knooppuntnummer} : N_p(i) = 1 \Leftrightarrow N_l(i) = 1$)

In de oude versie van ELECTR/INTERFCE werd met de eerste vier eisen geen rekening gehouden. Bijvoorbeeld: indien twee gebieden gekoppeld werden en daarna de gekoppelde snijlijn een spanning kreeg dan leverde FEMSYS foutmeldingen op.

Tijdens het invoeren van de voorgeschreven spanningen door de gebruiker wordt voor elk knooppunt bijgehouden of het al een voorgeschreven spanning moet krijgen en welke spanning het moet krijgen. Indien een knooppunt als linkerdeel in een transformatielijst voorkomt dan krijgt het bijbehorende rechterdeel de toegekende spanning. Bij de verwerking van de "koppelingen" zijn de "transformatiespecs" bijgehouden.

Bij het invoeren van de voorgeschreven spanningen is het mogelijk dat een knooppunt al een voorgeschreven spanning heeft gekregen. De laatst toegekende spanning wordt als de juiste gezien.

Nadat alle spanningen zijn ingevoerd worden de voorgeschreven spanningen in de file INREADFEM gezet. Met behulp van de bijgehouden gegevens wordt hierbij rekening gehouden met de eisen.

1.3.3 TOEVOEGING "COORDINAAT AANPASSING"

Bij het aanmaken van de constructie bij TRIQUAMESH is het niet mogelijk bij de basispunten kleine coördinaten te kiezen ($\approx 10^{-6}$). De gebruiker moet de factor aangeven waarmee de coördinaten van de knooppunten vermenigvuldigd moeten worden.

2 DE REPETITIE

In dit hoofdstuk wordt een beschrijving gegeven van de uitbreiding van de programmatuur om het systeem iteratief te laten werken. Het is de bedoeling dat FEMSYS een aantal keren wordt toegepast, waarbij het mogelijk is dat een aantal parameters per stap van de repetitie verandert. Deze parameters kunnen bijvoorbeeld de ladingsdichtheid of de spanning zijn. Alleen de mogelijkheid om de ladingsdichtheid als parameter te kiezen is uitgewerkt, omdat de interesse bij EHO hiernaar uitging.

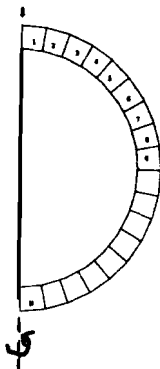
De veldproblemen, waarop de veranderlijke ladingsdichtheid als parameter kan worden toegepast, zijn de berekening van de oplading van een deeltje of object, dat zich in een extern elektrisch veld en ruimteladingsdichtheid bevindt. De vorm van het deeltje kan in principe willekeurig zijn, maar de software kan alleen rotatie-symmetrische vormen verwerken. De theorie hiervoor, wordt in paragraaf 2.1 nader toegelicht. Aan de software is een nieuw programma, dat de iteratie bevat, toegevoegd. Dit programma wordt in paragraaf 2.2 beschreven.

2.1 DE ELECTRISCHE THEORIE

In deze paragraaf worden de belangrijkste formules afgeleid, die nodig zijn om het oplaadproces van deeltjes of objecten m.b.v. de software te simuleren. De deeltjes bevinden zich in een extern elektrisch veld en ruimteladingsdichtheid.

Ten gevolge van het elektrische veld stroomt er lading naar het oppervlak van het deeltje. De ladingsstroom naar een punt op het oppervlak van het deeltje is afhankelijk van de normaalcomponent van het elektrische veld aan de buitenkant van het deeltje op dat punt. De normaalcomponent is o.a. afhankelijk van de hoeveelheid lading die al op het deeltje zit, dus de ladingsstroom naar het deeltje is tijdsafhankelijk. Er vindt geen ladingstransport op het deeltje plaats.

Beschouw nu het oppervlak van het deeltje verdeelt in N gebieden genummerd van 1 t/m N . Omdat in de software met ladingsdichtheid wordt gerekend moet elk gebiedje ook een bepaalde dikte hebben (zie figuur). Voor de nauwkeurigheid is van belang dat de gebieden niet te groot en niet te dik zijn.



Het is de bedoeling om voor een aantal tijdstippen de totale lading te bepalen die naar het deeltje is gestroomd.

Definities:

Laat $t \geq 0$ en $1 \leq n \leq N$

$i(t, n)$ = gemiddelde stroom naar gebied n op tijdstip t ,

$I(t)$ = totale stroom naar deeltje op tijdstip t ,

$q(t, n)$ = toename lading op gebied n t/m tijdstip t ,

$ro(t, n)$ = toename ladingsdichtheid in gebied n
t/m tijdstip t ,

$Q(t)$ = toename lading op deeltje t/m tijdstip t ,

$A(n)$ = oppervlakte gebied n ,

$V(n)$ = volume gebied n ,

$Er(t, n)$ = gemiddelde veldsterkte in gebied n op tijdstip t -
loodrecht op het oppervlak, de normaalcomponent,
(Er is positief indien het veld van het deeltje
af is gericht, andersom negatief).

Het is mogelijk om de stroom op een aantal discrete tijdstippen te bepalen. Hoe kleiner de tijdstappen (tijd

tussen twee tijdstippen), hoe groter de nauwkeurigheid.

Beschouw (M+1) punten op de tijd-as (tijdstippen):

$t_0 \dots t_M$, met $t_0 = 0$ en ($\Delta t_i: 1 \leq i \leq M: t_i > t_{i-1}$)

Opmerking: t_i is het (i+1) tijdstip, voor $0 \leq i \leq M$.

Hieronder volgen de formules waarmee $Q(t(M))$ kan worden afgeleid:

Laat $1 \leq n \leq N$

$$q(t_0, n) = 0$$

$$r_0(t_0, n) = 0$$

$$Q(t_0) = 0$$

Laat $0 \leq c < M$

$$Er(t_c, n) = f(\dots, r_0(t_c, 1), \dots, r_0(t_c, N), \dots)$$

met f een onbekende functie,

die met behulp van FEMSYS te berekenen is.

$$i(t_c, n) = N_0 * B * q_0 * A(n) * fn_0 * Er(t_c, n)$$

met N_0 is de ionenconcentratie,

B is de beweeglijkheid,

q_0 is de elementaire lading

$$(q_0 = 1.6 * 10^{-19} \text{ C}),$$

$$fn_0 = -1, \text{ voor } Er \leq 0,$$

$$fn_0 = 0, \text{ voor } Er > 0.$$

{indien de normaalcomponent van het deeltje af is gericht ($Er > 0$) dan is de stroom nul}

$$q(t_{c+1}, n) = q(t_c, n) + i(t_c, n) * (t_{c+1} - t_c),$$

$$r_0(t_{c+1}, n) = r_0(t_c, n) + (q(t_{c+1}, n) - q(t_c, n)) / V(n)$$

$$= q(t_{c+1}, n) / V(n).$$

Met behulp van deze formules kan $i(t_c, n)$ en $q(t_{c+1}, n)$ iteratief worden berekend, met $0 \leq c < M$.

Hiermee kan $I(t_c)$ en $Q(t_{c+1})$ worden berekend, namelijk:

$$I(t_c) = (\sum_{n: 1 \leq n \leq N: i(t_c, n)}).$$

$$Q(t_{c+1}) = (\sum_{n: 1 \leq n \leq N: q(t_{c+1}, n)}).$$

Indien in het begin nog geen lading op het deeltje zat, dan is $Q(t_M)$, voor t_M gaat naar oneindig, de verzadigingslading.

Eenheden:

Laat $t \geq 0$ en $1 \leq n \leq N$:

symbool	S. I. eenheid
$i(t, n)$	[A]
$q(t, n)$	[C]
$I(t)$	[A]
$Q(t)$	[C]
$r_0(t, n)$	[C/m ³]
$A(n)$	[m ²]
$V(n)$	[m ³]
$Er(t, n)$	[V/m]
N_0	[m ⁻³]
B	[m ² /Vs]
q_0	[C]

2.2 HET NIEUWE PROGRAMMA

Aan de software is een nieuw programma, genaamd ELECTR/REPFEMSYS, toegevoegd. Dit programma bevat de iteratie zoals beschreven in paragraaf 2.1.

In paragraaf 2.2.1 wordt een abstracte beschrijving van het programma gegeven. De paragrafen 2.2.2 t/m 2.2.7 gaan in meer detail op een aantal zaken in.

In paragraaf 2.2.2 wordt de aanmaak van de gebieden beschreven.

De tijdstippen, waarvoor de ladingsstroom naar het deeltje moet worden berekend, wordt in paragraaf 2.2.3 besproken.

De invoergegevens voor het programma worden in paragraaf 2.2.4 besproken.

Hoe een gebruiker met het programma kan werken, wordt in paragraaf 2.2.5 uitgelegd.

In paragraaf 2.2.6 wordt een globaal overzicht van het programma gegeven.

In appendix A12 staat de programmatekst van dit programma.

2.2.1 ABSTACTE BESCHRIJVING ELECTR/REPFEMSYS

Van het programma wordt een abstracte beschrijving gegeven, hierbij wordt gebruik gemaakt van de in paragraaf 2.1 afgeleide formules. De beschrijving wordt gedaan in een soort Pascal.

Bij de beschrijving wordt aangenomen dat er N gebieden zijn en de ladingsdichtheid op (maximaal) $M+1$ tijdstippen moet worden berekend, dus het programma maakt (maximaal) M stappen. De tijdstippen worden in het programma berekend.

Definitie:

$t_j =$ "het $(j+1)$ tijdstip" , voor $0 \leq j \leq M$.

Opmerking: $t_0 = 0$.

Voor het programma bestaan de volgende invarianten:

P1 : $0 \leq m \leq M$ and
 $(\Delta j: 0 \leq j < (m+1): \text{Tijd}(j) = t_j)$ and
 $(\Delta j: 0 \leq j < (m+1): \text{Lad}(j) = Q(t_j))$ and
 $(\Delta j: 0 \leq j < m : \text{Strm}(j) = I(t_j))$ and
 $(\Delta j: 1 \leq j < (N+1): \text{rho}(j) = \text{ro}(t_m, j))$
P2 : $1 \leq n \leq (N+1)$ and
 $(\Delta j: 1 \leq j < n: \text{rho}(j) = \text{ro}(t_m, j))$
P3 : $1 \leq n \leq (N+1)$ and
 $\text{Strm}(m) = (\Sigma j: 1 \leq j \leq n: i(t_m, j))$
P4 : $1 \leq n \leq (N+1)$ and
 $(\Delta j: 1 \leq j < n: \text{rho}(j) = \text{ro}(t_m, j))$ and
 $\text{dqt} = (\Sigma j: 1 \leq j \leq n: q(t_{m+1}, j) - q(t_m, j))$

Programma:

```
{ N ≥ 0 and M ≥ 0 }
n:= 1; {P2}
do n < N+1 --> rho (1):= 0; n:= n+1 {P2} od;
Lad (0):= 0; Tijd (0):= 0; m:= 0;
{P1}
do m < M
--> "compute ( $\Delta n: 1 \leq n \leq N: \text{Er}(n) = f(\text{Er}(t_m, n))$ )";
n:= 1; Strm (m):= 0;
{P3}
do n < N+1
--> h:=  $N_0 * B * q_0 * A(n) * \text{fn}_0 * \text{Er}(n)$ ;
Strm (m):= Strm (m) + h;
n:= n+1
{P3}
od;
{P3 and n = (N+1)}
"compute Tijd(m+1)";
n:=1; dqt:= 0;
{P4}
do n < N+1
```

```

--> h:= N0 * B * q0 * A(n) * fn0 * Er(n) *
      (Tijd(m+1)-Tijd(m));
dqt:= dqt + h;
rho (n):= rho (n) + h/V(n);
n:= n+1
{P4}
od;
{P4 and n = (N+1)}
Lad (m+1):= Lad (m) + dqt;
m:= m+1 {P1}
od
{(Δ1: 0 ≤ i < (M+1): Tijd (i) = ti) and
(Δ1: 0 ≤ i < (M+1): Lad (i) = G(ti)) and
(Δ1: 0 ≤ i < M      : Strm (i) = I(ti)) and
(Δ1: 1 ≤ i < (N+1): rho (i) = ro(tM, i)) }

```

De guard (m < M) is versterkt tot :

```

(m < M and not "eindcriterium" ) ,
met "eindcriterium" = m <> 0 and
                    (Strm (m-1) < delta * Strm (0)),
met 0 ≤ delta ≤ 1.

```

2. 2. 2 BEPALING GEBIEDEN

De opbouw van de gebiedjes en de daarbij behorende waarden voor de oppervlakte, het volume en de normaalcomponent worden in deze paragraaf besproken.

De bepaling van de gebieden zou in principe aan de gebruiker overgelaten kunnen worden. Omdat de gebieden niet al te groot mogen zijn, levert dit nogal wat werk voor de gebruiker op. Daarom worden de gebieden door het programma ELECTR/INTERFACE bepaald. Dit wordt in ELECTR/INTERFACE gedaan omdat de benodigde programmatuur in dit programma aanwezig is.

Van het geheel moet een constructie gemaakt worden zodanig dat deze geschikt is voor toepassing van FEMSYS. Deze constructie moet 3-dimensionaal zijn met de y-as als symmetrie-as. De rand waarop de lading komt te zitten moet op een snijlijn van twee gebieden liggen. De constructie moet zo gemaakt zijn dat er op deze rand een dubbele serie knooppunten is gegenereerd.

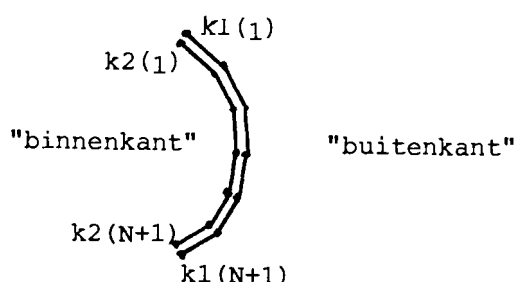
De gebruiker moet in ELECTR/INTERFACE de rand, de groep van waar lading op de rand komt te zitten (de "buitenkant") en de groep waar de lading komt te zitten (de "binnenkant") aangeven. Met behulp van deze gegevens zijn de gebieden te bepalen.

Laat de rand uit $(N + 1)$ dubbele knooppunten bestaan.

Definitie:

$k1(i)$ = nummer knooppunt aan "buitenkant" rand op positie i

$k2(i)$ = nummer knooppunt aan "binnenkant" rand op positie i



Er worden door het programma N gebieden aangemaakt, genummerd van 1 t/m N .

Aan ieder gebied n wordt een aantal elementen aan de "binnenkant" van de rand toegekend. Deze elementen bepalen het volume van het gebied. Omdat de software met driehoekjes werkt zijn deze elementen driehoekjes.

Gebied n bevat de volgende driehoekjes:

Het driehoekje dat de punten $k2(n)$ en $k2(n+1)$ bevat en alle driehoekjes die aan de volgende twee eisen voldoen:

- 1) het bevat het knooppunt $k2(n)$,
- 2) ($\Delta i: 1 \leq i \leq N$: het driehoekje bevat niet de knooppunten $k2(i)$ en $k2(i+1)$).

Het volume van een gebied ontstaat na wenteling van de

bijbehorende driehoekjes om de y-as. Dus de grootte van het volume van een gebied wordt bepaald door de grootte van de driehoekjes. De grootte van de driehoekjes kan door de gebruiker in de file DATAMG worden aangegeven. De verdeling in driehoekjes door TRIQUAMESH is zodanig dat elk gebied tenminste één driehoekje bevat. Voor de bepaling van de oppervlakte van gebied n worden de knooppunten $k_1(n)$ en $k_1(n+1)$ gebruikt. De uitvoer van het programma FEMSYS levert voor elk knooppunt een potentiaal op. Om de normaalcomponent van het elektrisch veld te berekenen, behorende tot gebied n, worden de potentialen van van het driehoekje met de knooppunten $k_1(n)$, $k_1(n+1)$ en zijn derde punt gebruikt. Dus aan elk gebied wordt nog een driehoekje aan de "buitenkant" toegekend.

Definities:

- 1) driehoek = $\{x_1, x_2, x_3\}$, waarin x_1 , x_2 en x_3 onderling verschillende knooppuntnummers zijn
- 2) driehoeken = $\{d \mid d = \text{driehoek}\}$

Opm: de verzameling driehoeken is door TRIQUAMESH aangemaakt.

Hieronder volgt een formele beschrijving van de gebieden. Laat n een gebied zijn

- $$DR_1(n) \in \text{driehoeken met } k_1(n) \in DR_1(n) \\ \text{en } k_1(n+1) \in DR_1(n)$$
- $$ADP(n) = \text{knooppuntnummer met } ADP(n) \in DR_1(n) \\ \text{en } ADP(n) \neq k_1(n) \\ \text{en } ADP(n) \neq k_1(n+1)$$
- $$DR_2(n) = \{d \in \text{driehoeken} \mid (k_2(n) \in d \text{ and } k_2(n+1) \in d) \text{ or} \\ (k_2(n) \in d \text{ and} \\ (\exists i: 1 \leq i \leq N: k_2(i) \in d \text{ and} \\ k_2(i+1) \in d))\}$$
- $$KP_1(n) = k_1(n)$$
- $$- \quad KP_2(n) = k_1(n+1)$$

Voor elk gebied n zijn de volgende gegevens van belang:

- 1) $KP_1(n)$ en $KP_2(n)$
- 2) $DR_1(n)$
- 3) $DR_2(n)$

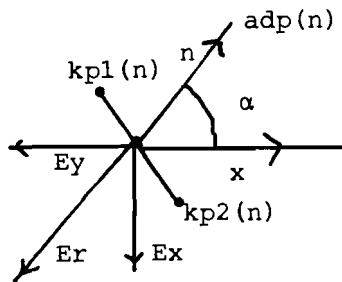
Definities:

Laat $1 \leq n \leq N$

- 3) $Opp(n)$ = de oppervlakte die ontstaat na wenteling van het lijnstuk $KP_1(n)$ en $KP_2(n)$ om de y-as. {oppervlakte van gebied n}
- 4) $Vol(n)$ = $(\sum_{d \in DR_2(n)} \text{het volume dat ontstaat na wenteling van } d \text{ om de y-as})$ {volume van gebied n}

Voor de formules van de oppervlakte zie appendix 1a.

Voor de formule van het volume zie appendix 1b.
 De berekening van het volume en van het oppervlak van een gebied wordt in het programma ELECTR/REPFEMSYS gedaan.
 Voor de berekening van de normaalcomponent moet je eerst afspreken wat positief is en wat negatief. Een positieve normaalcomponent wil zeggen dat het veld van het deeltje af is gericht. Dus indien de normaalcomponent negatief is vindt er ladingsstroom naar het deeltje plaats. Met behulp van de door FEMSYS berekende potentialen in de knooppunten van DR1(n) is het mogelijk de x en y component (Ex en Ey) van het elektrische veld in gebied n te berekenen (zie appendix 1c). Met behulp van deze componenten kan de normaalcomponent van gebied n als volgt worden berekend:
 Laat α de hoek tussen de vector n, die loodrecht op het lijnstuk van KP1(n) naar KP2(n) staat en gericht naar punt ADP(n), en de vector x, die evenwijdig met de x-as is en positief gericht is, zijn.



De projectie van Ex en Ey op vector n is:

$$ER(n) = \cos(\alpha) * Ex + \sin(\alpha) * Ey$$

Voor de afleiding van hoek α zie appendix 1d.
 De berekening van de hoek α wordt in ELECTR/REPFEMSYS gedaan.

2. 2. 3 DE TIJDSTIPPEN

De tijdstippen, waarop de ladingsdichtheid moet worden berekend, en het maximale aantal stappen, die de repetitie mag uitvoeren, moeten door de gebruiker worden aangegeven. Voor het aangeven van de tijdstippen heeft de gebruiker drie mogelijkheden, namelijk:

- 1) De tijdstappen zijn constant:
De gebruiker moet één tijdstap opgeven (dt).
- 2) De tijdstappen worden door het programma ELECTR/REPFEMSYS berekend:
De gebruiker moet de totale lading (dq), die per tijdstap naar het deeltje mag stromen, en de maximale tijdstap (mts) aangeven.
Een van de grootheden die "berekend" wordt is de totale stroom naar het deeltje op een bepaald tijdstip t: I(t). Dit bepaalt hoe groot de tijdstap op moment t moet zijn:
tijdstap = $\min(dq/I(t), mts)$.
Het volgende tijdstip is: t + tijdstap.
- 3) De tijdstappen zijn variabel:
De gebruiker moet M tijdstappen aangeven, met M is aantal stappen (dt₁, ..., dt_M).

Deze gegevens worden in een file gezet (zie 2.2.5) en deze file wordt door ELECTR/REPFEMSYS ingelezen. De tweede mogelijkheid is specifiek gericht op de verandering van de parameter ladingsdichtheid. Omdat in het begin de stroom groot is, moeten de tijdstappen in het begin klein zijn en later mogen ze wat groter zijn. Dit laatste kan ook met de derde mogelijkheid worden bereikt maar het is moeilijk voor een gebruiker te bepalen hoe groot de tijdstappen moeten zijn.

2.2.4 DE INVOERGEGEVENS

De extra gegevens, die voor de repetitie van belang zijn, worden over twee files verdeeld, genaamd INREADREPFEM1 en INREADREPFEM2. Deze files kunnen met behulp van ELECTR/INTERFACE of met behulp van ELECTR/INTERFACE/NI (zie hoofdstuk 3) worden aangemaakt.

De files INREADFEM, die de gegevens over de constructie bevat (zie 1.2), INREADREPFEM1 en INREADREPFEM2 zijn invoerfiles voor het programma ELECTR/REPFEMSYS.

De extra gegevens kunnen in twee stukken worden opgesplitst, namelijk:

1) Een aantal te berekenen gegevens:

Hieronder vallen de gegevens die bepalen hoe de gebieden eruit zien en nog een aantal constanten die in het programma ELECTR/REPFEMSYS nodig zijn voor het vastleggen van de grenzen van een aantal array's. Deze gegevens komen in de file INREADREPFEM1 te staan.

De gebruiker kan deze file aanmaken door ELECTR/INTERFACE aan te roepen en daarin voor veranderlijke ladingdichtheid kiezen. Vervolgens dient de gebruiker één of meer deelranden, die lading moeten krijgen, aan te geven. Zo'n deelrandspecificatie bestaat uit:

- a) aangeven van de groep die de buitenrand van het object bevat, de "buitenkant". Deze groep bevat de driehoekjes waarvoor de normaalcomponent van het elektrisch veld moet worden berekend.
- b) aangeven van de groep die de binnenrand van het object bevat, de "binnenkant". Deze groep bevat de driehoekjes waar lading naar toe stroomt.
- c) aangeven van een drietal basispunten die de deelrand bepalen (zie 1.2).

Per deelrandspecificatie wordt in ELECTR/INTERFACE de aangegeven deelrand in gebieden verdeeld en per gebied bepaald welke driehoekjes erbij behoren (zie 2.2.2). Deze driehoekjes worden in de file gezet. De knooppunten van de driehoekjes worden zo opgeslagen dat bekend is welke knooppunten op de rand liggen en welke niet.

Verder wordt nog het totaal aantal aangemaakte gebieden, het totale aantal driehoekjes aan de "binnenkant", het aantal door TRIQUAMESH aangemaakte knooppunten, het aantal door TRIQUAMESH aangemaakte driehoekjes en het aantal groepen, waaruit de constructie bestaat (zie 1.1), opgeslagen.

SAMENVATTEND:

de file INREADREPFEM1 bevat de volgende gegevens:

- aantal knooppunten aangemaakt door TRIQUAMESH
- aantal driehoeken aangemaakt door TRIQUAMESH
- aantal groepen
- totaal aantal aangemaakte gebieden
- totaal aantal driehoekjes aan de "binnenkant"

- per gebied
 - * driehoekje "buitenkant" met de daarbij behorende knooppunten
 - * de driehoekjes aan de "binnenkant" met de daarbij behorende knooppunten

Deze file moet m. b. v. ELECTR/INTERFACE worden aangemaakt en mag niet door de gebruiker worden gewijzigd.

2) Een aantal door de gebruiker in te voeren constanten:

Deze constanten zijn de ionenconcentratie n_0 , de beweeglijkheid B , het maximaal aantal stappen, de gegevens over de tijdstippen en de afbreekconstante δ (zie 2.2.1).

Deze gegevens worden in de file INREADREPFEM2 gezet. De gebruiker kan deze file door ELECTR/INTERFACE aan laten maken.

Indien hij in dit programma voor veranderlijke ladingsdichtheid kiest, vraagt dit programma automatisch naar deze gegevens.

De gebruiker mag in deze file veranderingen aanbrengen. Hieronder volgt de syntax van deze file:

GRAMMATICA

De 'non-terminals' :

{s, maxstap, n_0 , b, dt, dq, mt, delta, tsp1, tsp2, tsp3("integer"), dt("integer"), real, integer, unsint, decgetal, cijfers, cijfer, teken, punt, expt}

De 'terminals' : Sym U Cyfer U Teken U Expt U Dpunt

Sym = {"", "}
 Cyfer = {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9"}
 Teken = {"-", "+"}
 Expt = {"E", "e", "@"}
 Dpunt = {"."}

Startsymbool: s

Grammatica-regels:

s ::= 1, maxstap, n_0 , b, tsp1 delta |
 2, maxstap, n_0 , b, tsp2 delta |
 3, maxstap, n_0 , b, tsp3(n) delta
 {met n = maxstap}
 maxstap ::= integer
 n_0 ::= real
 {controle $n_0 > 0$ }
 b ::= real
 {controle b > 0}

```

tsp1      ::= dt,                {zie 2.3 geval 1}
tsp2      ::= dq, mts,          {zie 2.3 geval 2}
tsp3(n)   ::= e                 voor n ≤ 0
tsp3(n)   ::= tsp3(n-1) dt(n),  voor n ≥ 1
                                           {zie 2.3 geval 3}

dt        ::= real
           {controle dt > 0}

dq        ::= real
           {controle dq > 0}

mts       ::= real
           {controle mts > 0}

dt(n)     ::= real
           {controle dt(n) > 0}

delta     ::= real
           {controle 0 ≤ delta ≤ 1}

real      ::= decgetal | decgetal expt exp |
           expt exp | teken expt exp

decgetal  ::= integer | integer punt unsint | punt unsint
           | teken punt unsint | integer punt

integer   ::= teken unsint | unsint

unsint    ::= cijfer cijfers

cijfers   ::= cijfer cijfers | e

cijfer    ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" |
           "7" | "8" | "9"

teken     ::= "+" | "-"

expt      ::= "E" | "e" | "@"

exp       ::= cijfer cijfer | teken cijfer cijfer |
           cijfer | teken cijfer

punt      ::= "."

```

Opmerking:

- | is of en e is leeg.

Voorbeeld:

Gekozen wordt voor:

Maxstap = 5, $N_0 = 5 \times 10^{-14}$, $B = 2.1 \times 10^{-4}$, $\Delta = 0$ en voor
variabele tijdstappen: 1×10^{-3} , 2×10^{-3} , 3×10^{-3} , 4×10^{-3}
en 5×10^{-3} .

De file kan er als volgt uitzien:

```

FILE INREADFEM2
100 3,
200 5,
300 5E14, 2.1E-4,
400 1E-3, 2E-3, 3E-3, 4E-3, 5E-3,
500 0,

```

Opmerking: De file moet van het type data zijn.

2. 2. 5 GEBRUIK VAN ELECTR/REPFEMSYS

Aan de software is het programma ELECTR/REPFEMSYS toegevoegd. Hierin wordt FEMSYS een aantal malen aangeroepen. Dit programma maakt gebruik van ELECTR/ROTELEMENT om FEMSYS te starten.

De gebruiker kan dit nieuwe programma direct executeren of met behulp van een job, genaamd ELECTR/RFJOB, opstarten. Deze job kan als nachtjob worden opgestart. Bij het direct executeren van het programma is van belang dat de gecompileerde versie van ELECTR/ROTELEMENT aanwezig is.

De files INREADFEM, INREADREPFEM1 en INREADREPFEM2 moeten voor dit programma aanwezig zijn. De file INREADFEM is de file waarin de gegevens staan waarmee FEMSYS gaat rekenen. Van deze file wordt in het nieuwe programma een copy gemaakt, genaamd CIRF. In deze file wordt de verandering van de ladingsdichtheid aangebracht. Deze file wordt als invoerfile aan FEMSYS meegegeven. Zo is na afloop van dit nieuwe programma de file INREADFEM ongewijzigd. Hiermee is het mogelijk om vanuit de beginsituatie ELECTR/REPFEMSYS op te starten zonder INREADFEM opnieuw aan te maken. Op de laatste versie van CIRF, de versie die na afloop bestaat, is FEMSYS nog niet toegepast.

Het programma wordt gestopt indien aan het eindcriterium is voldaan of het maximaal aantal stappen wordt bereikt (zie 2.2.1). Het programma is zo gemaakt dat vanuit de zo ontstane eindsituatie verder kan worden gegaan. Hierbij moet de naam van de file CIRF veranderd worden in INREADFEM en de file INREADREPFEM2 kan eventueel gewijzigd worden. Vanzelfsprekend moet het programma opnieuw aangeroepen worden.

Het programma levert de volgende uitvoerfiles op:

- de laatste versie van de file CIRF,
- de file OUTREADFEM:
OUTREADFEM is een copy van CIRF. Als de berekeningsgang niet goed verlopen is doordat de invoer bevat in CIRF inkorrekt is zijn aan OUTREADFEM foutmeldingen toegevoegd,
- de file OUTO:
Indien het programma m stappen ($m > 0$) heeft gemaakt dan bevat OUTO de potentialen in de knooppunten op het m-de tijdstip,
- eventueel een aantal plotfiles:
 - de plotfile .../COMPNODES,
 - de plotfile .../COORD,
 - de plotfile .../COMPCHAR,
 - Indien het programma m stappen heeft gemaakt de plofiles:
...1.../NODEDER, met $1 \leq i \leq m$. De plotfile ...1.../NODEDER bevat de berekeningsgegevens op het i-de tijdstip,
- de file OUTER bevat per gebied en per tijdstip:
 - de normaalcomponent van het elektrische veld in het gebied,
 - de totale toegestroomde lading naar het

gebied,

Voor een uitdraai van deze file zie appendix A8,

- de file OUTREPFEM bevat:

- * Indien een file niet aanwezig is een foutmelding
- * Anders -per tijdstip de totale stroom,
 - per tijdstap de toename van de totale lading,
 - per tijdstip de totale toegestroomde lading,
 - waarden van een aantal constanten zoals N_0 .

Voor een uitdraai van deze file zie appendix A9.

2.2.6 GLOBAAL OVERZICHT ELECTR/REPFEMSYS

Initialisatie:

- testen op aanwezigheid invoerfiles INREADFEM, INREADREPFEM1 en INREADREPFEM2,
- controle of INREADREPFEM2 van de goede vorm is (zie 2.2.4),
- inlezen gegevens INREADREPFEM1,
- inlezen gegevens INREADREPFEM2,
- inlezen van een aantal gegevens uit INREADFEM:
 - coördinaten knooppunten,
 - karakteristieken van de driehoekjes (ladingsdichtheid en permittiviteit),
- de identificatieregel (zie 1.2),
- nagaan of er plots moeten worden gemaakt,
- aanmaken van de file CIRF,
- berekenen van een aantal constanten:
 - volume per gebied,
 - oppervlakte per gebied,
 - de hoek α per gebied,
- "($\Delta 1: 1 \leq i \leq N: \rho(n) := 0$); Lad(0):=0; Tyd(0):= 0"
(zie 2.2.1),
- aantal gemaakte stappen wordt nul.

Per stap van de repetitie worden de volgende onderdelen automatisch uitgevoerd:

- verwijderen van de file OUTO, zodat na de uitvoering van FEMSYS na kan worden gegaan of de berekening goed is verlopen,
- uitvoeren van een eindige elementen berekening (FEMSYS met als invoerfile CIRF),
- controle of de berekening goed is verlopen door na te gaan of de file OUTO is aangemaakt,
- inlezen van de potentialen uit de file OUTO, met behulp van de potentialen worden de elektrische velden berekend,
- indien plots de naam van de plotfile .../NODEDER aanpassen, zodat deze file bij een volgende uitvoering van FEMSYS niet wordt overschreven,
- aanpassen van de variabelen "rho", "Lad", "Tyd" en "Strm" (zie 2.2.1),
- uitvoerfile OUTER uitbreiden,
- toename ladingsdichtheid per gebied omzetten in totale ladingsdichtheid per driehoek (zie Ad1),
- verandering aanbrengen in CIRF (zie Ad2),
- aantal gemaakte stappen ophogen

Na afloop van de repetitie wordt de file OUTREPFEM aangemaakt.

AD1:

FEMSYS rekent met de totale ladingsdichtheid per driehoek. Dus uit de toename van de ladingsdichtheid per gebied moet

nog de totale ladingsdichtheid per driehoek worden bepaald, namelijk:

Laat d een driehoek en t een tijdstip zijn, dan

$\text{rhodi}(d)$ = de beginladingsdichtheid van driehoek d
{kan uit de invoerfile INREADFEM worden afgelezen},

$\text{rhod}_t(d)$ = {de totale ladingsdichtheid voor driehoek d op tijdstip t }
($\exists n: 1 \leq n \leq N$ and $d \in \text{DR2}(n): \text{rho}_t(n)$) +
 $\text{rhodi}(d)$,
met $\text{rho}_t(n)$ = "toename ladingsdichtheid in gebied n t/m tijdstip t ".

Ad2:

Hoe de veranderlijke ladingsdichtheid in de file CIRF wordt aangebracht vergt nog enige uitleg.

Eerst wordt besproken hoe de karakteristieken per driehoek worden opgeslagen en hoe FEMSYS deze karakteristieken inleest. Bij het programma ELECTR/INTERFACE worden de karakteristieken per groep ingevuld. Het aantal karakteristieken bedraagt hier altijd twee (zie 1.2). Deze karakteristieken worden aan de bij de groepen behorende driehoekjes toegekent. Ieder driehoekje heeft een uniek nummer tussen 1 en ADR gekregen, met ADR het aantal aangemaakte driehoekjes.

De driehoekjes komen samen met hun karakteristieken in een zogenaamde charlijst te staan.

Hieronder volgt de syntax van een charlijst:

GRAMMATICA:

De 'non-terminals':

{charlijst, char1, char2, exceptlijst, charspecificatie, driehoekspec, driehoeknum, real}

De 'terminals':

{"CHARACTERISTICS", "EXCEPT", "(", ")", ":", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "E", "@", "e", ".", "+", "-"}.

Startsymbool: charlijst

Grammatica-regels:

```
charlijst      ::= CHARACTERISTICS char1 , char2 ;
                exceptlijst
char1          ::= real
char2          ::= real
exceptlijst    ::= EXCEPT : charspecificatie | e
charspecificatie ::= driehoekspec : char1 , char2 ;
driehoekspec   ::= driehoeknum ( unsint ) driehoeknum |
                driehoeknum
driehoeknum ∈ [1:ADR],
```

unsint is een natuurlijk getal,
real is een reëel getal.

Hieronder volgt de semantiek van de eerste driehoekspecificatie:

Laat $a, b \in [1:ADR]$ en $n \geq 0$

dan $a(n)b = \{d \mid a \leq d \leq b \text{ and } (\exists i: i \geq 0: d = a + i * n)\}$

De karakteristieken voor de driehoekjes worden nu als volgt door FEMSYS bepaald:

Ken aan elk driehoekje de eerst vermelde (onder de kop CHARACTERISTICS) karakteristieken toe. Daarna wordt per charspecificatie in de exceptlijst bekeken welke driehoekjes andere karakteristieken moeten krijgen. Dus indien in een exceptlijst een driehoek meerdere malen voorkomt dan wordt de laatst voorkomende charspecificatie de uiteindelijke.

Voorbeeld: (stel er zijn 9 driehoekjes)

CHARACTERISTICS 1, 2;

EXCEPT 3(1)6 : 3, 4; 5 : 5, 6;

Dit betekent:

driehoeken 1, 2, 7, 8, 9 hebben karakteristieken 1 en 2

driehoeken 3, 4, 6 hebben karakteristieken 3 en 4

driehoek 5 heeft karakteristiek 5 en 6.

In het geval je met ladingsdichtheid te maken hebt dan stelt de eerste karakteristiek de permittiviteit en de tweede karakteristiek de ladingsdichtheid voor.

De aanpassing van de ladingsdichtheid is als volgt:

De oorspronkelijke specificatie blijft gehandhaafd maar aan de exceptlijst worden een aantal charspecificaties toegevoegd. Per driehoek, die tot een gebied behoort, één specificatie, namelijk: het nummer van de driehoek gevolgd door de waarde van zijn permittiviteit en de nieuwe waarde van zijn ladingsdichtheid.

Opmerkingen:

- 1) Indien een file niet voorkomt wordt het programma afgebroken en in de file OUTREPFEM komt een foutmelding te staan.
- 2) Omdat het programma veel rekentijd kan kosten wordt de uitvoerfile OUTREPFEM per stap van de repetitie aangemaakt.
- 3) De repetitie bestaat o.a. uit het uitvoeren van FEMSYS. Als invoer gebruikt FEMSYS de file CIRF. Indien deze file t.o.v. de vorige stap ongewijzigd is, dan wordt de repetitie beëindigd.

3 ELECTR/INTERFACE/NI

Van het programma ELECTR/INTERFACE is een niet-interactieve versie gemaakt, genaamd ELECTR/INTERFACE/NI. Het invoeren van de randvoorwaarden gaat via een file INTERF. Hiermee is het mogelijk om het geheel van programma's in één job te zetten en het vergemakkelijkt het invoeren van de randvoorwaarden. ELECTR/INTERFACE bestaat uit een aantal blokken waarin een actie moet worden uitgevoerd (zie 1.2). De volgorde van die blokken is in de nieuwe versie gehandhaafd. De blokken worden hieronder beschreven m.b.v. een grammatica. De afzonderlijke onderdelen, die bij het interactieve gedeelte moesten worden ingevuld, komen hier in terug. De namen van de files DATAMG, MESH en INTERF moeten bij de executie worden meegegeven.

De nieuwe versie bevat een scanner die de terminals herkent. De grammatica regels worden gecontroleerd volgens het principe van de 'Recursive Descent Parser' (zie [HEM84]). Indien er een fout in de invoer zit wordt het programma gestopt. Van de invoerfile INTERF wordt een copie, genaamd INTERF/OUT, met eventuele foutmeldingen gemaakt. Indien bij de verwerking van de gegevens iets fout gaat dan wordt het programma niet afgebroken. Dus is het verstandig om in deze copy te kijken of alles goed is gegaan.

Hieronder volgt de grammatica waaraan de file INTERF moet voldoen:

GRAMMATICA

De 'non-terminals' :

```
{si, id, ka, ca, re, ko, ze, ra, gr, pl, vl, maxknopen,
maxelement, ident, kasp ("integer"), kasp, char1, char2,
mfactor, resps, resp, elnaam, achar, randsp, char,
char("integer"), groep, van, via, naar, koppelsps,
koppelsp, groep1, groep2, randsp, vssps, vssp, span,
gebiedsps, gebiedsp, vardec, maxstap, n0, b, tsp, delta, dt,
dq, mts, tsp1("integer"), dt("integer"), naam, unsint,
integer, decgetal, exp, real, cijfer, cijfers, letter,
letters, punt, expt, expt1, teken}
```

De 'terminals' :

Sym U Letter U Cyfer U Rwords U Teken U Expt U Dpunt met

```
Sym    = {"(", ")", ",", ";", ":"}
Letter = {"a", ..., "z", "A", ..., "Z"}
Cyfer  = {"0", ..., "9"}
Rword  = {"KARAKTERISTIEKEN", "RANDELEMENTEN", "KOPPELEN",
          "ZWEVENDE", "CA", "ELEKTRODE", "RANDSPANNINGEN",
```

"GEAARDE", "RANDEN", "PLOTS", "VERANDERLIJKE",
 "LADINGSDICHTHEID", "MAXSTAP", "NO", "B", "DELTA",
 "DT", "DQ", "MTS", }
 Teken = {"+", "-"}
 Expt = {"E", "e", "@"}
 Dpunt = {"."}

Startsymbool: si

Grammatica-regels

si ::= id ka ca re ko ze ra gr pl vl
 id ::= ident (maxknopen , maxelement)
 {ident is de identificatieregel,
 maxknopen is de bovengrens voor het aantal
 knooppunten en
 maxelement is de bovengrens voor het aantal
 elementen (zie 1.2)}
 ident ::= naam
 maxknopen ::= integer
 maxelement ::= integer
 ka ::= KARAKTERISTIEKEN
 Kasps (ng) , ng = aantal groepen
 kasps(n) ::= kasps(n-1) kasp ; , voor n ≥ 1
 {kasp bevat de karakteristieken van groep n}
 kasps(0) ::= e
 kasp ::= char1 , char2 | char1
 {char1 is de eerste karakteristiek,
 char2 is de tweede karakteristiek (zie 1.2)}
 char1 ::= real
 char2 ::= real
 ca ::= CA (mfactor) | e
 {CA is coördinaataanpassing (zie 1.3)}
 mfactor ::= real
 re ::= RANDELEMENTEN resps | e
 resps ::= resp ; resps | e
 resp ::= elnaam (achar)
 randsp : char(n) , met n = achar
 char(n) ::= char , char (n - 1) , voor n ≥ 2
 char(1) ::= char
 char(n) ::= e , voor n ≤ 0
 char ::= real
 achar ::= integer
 elnaam ::= naam
 randsp ::= groep , van , via , naar
 groep ::= integer
 van ::= integer
 via ::= integer
 naar ::= integer
 ko ::= KOPPELEN koppelsps | e
 Koppelsps ::= koppelsp ; koppelsps | e
 Koppelsp ::= groep1 , groep2 , van , via , naar
 groep1 ::= integer

```

groep2      ::= integer
ze          ::= ZWEVENDE ELEKTRODE randsp | e
randsp     ::= randsp ; randsp | e
ra         ::= RANDSPANNINGEN vssps | e
vssps     ::= vssp ; vssps | e
vssp      ::= randsp : span
span      ::= real
gr        ::= GEAARDE RANDEN randsp | e
pl        ::= PLOTS | e
vl        ::= VERANDERLIJKE LADINGSDICHTHEID
           gebiedsp ; gebiedsps vardec
           | e
gebiedsps ::= gebiedsp ; gebiedsps | e
gebiedsp  ::= groep1, groep2 , van , via , naar
           {groep1 is groep "buitenkant",
            groep2 is groep "binnenkant" (zie 2.2.4)}
vardec    ::= MAXSTAP : maxstap ;
           NO      : n0 ;
           B       : b ;
           tsp
           DELTA   : delta ;
maxstap   ::= integer
n0        ::= real
           {controle: n0 > 0}
b         ::= real
           {controle: b > 0}
delta     ::= real
           {controle: 0 ≤ eps ≤ 1}
tsp       ::= DT      : dt ; |
           DQ      : dq ;
           MTS     : mts ; |
           DT      : tsp1(n) ;           ,met n = maxstap
dt        ::= real
           {controle: dt > 0}
dq        ::= real
           {controle: dq > 0}
mts       ::= real
           {controle: mts > 0}
tsp1(n)   ::= e           ,voor n ≤ 0
tsp1(n)   ::= dt(n)      ,voor n = 1
tsp1(n)   ::= tsp1(n-1) , dt(n) ,voor n ≥ 2
dt(n)     ::= real      ,voor n ≥ 1
           {controle: dt(n) > 0}
naam      ::= letter letters cijfers \ Rwords
letter    ::= "a" | "b" ... | "z" | "A" ... | "Z"
letters   ::= letter letters | e
cijfers   ::= cijfer cijfers | e
cijfer    ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" |
           "7" | "8" | "9"
unsint    ::= cijfer cijfers
integer   ::= teken unsint | unsint
teken     ::= "+" | "-"
real      ::= decgetal | decgetal expt exp |
           expt1 exp | teken expt exp
expt      ::= "E" | "e" | "@"

```

```

expt1      ::= "@"
exp        ::= cijfer cijfer | teken cijfer cijfer |
              cijfer | teken cijfer
decgetal   ::= integer | integer punt unsint | punt unsint |
              teken punt unsint | integer punt
punt       ::= "."

```

Opmerkingen:

- In de gereserveerde woorden (Rwords) mogen ook kleine letters staan.
- | is of, e is leeg en \ is behalve.

Voorbeeld van een file INTERF die aan de regels voldoet:

```

FILE INTERF:
100  VOORBO (200,300)
200  KARAKTERISTIEKEN
300  1E-12, 1E-17;
400  2E-12, 1E-17;
500  KOPPELEN
600  1, 2, 3, 4, 5;
700  RANDSPANNINGEN
800  1, 1, 1, 2:5;
900  GEAARDE RANDEN
1000 2, 2, 2, 3;
1100 PLOTS
1200 VERANDERLIJKE LADINGSDICHTHEID
1300 1, 2, 3, 4, 5;
1400 MAXSTAP : 10;
1500 NO      : 2.1E15;
1600 B       : 5@-4;
1700 DT      : @-3;
1800 EPS     : 1E-3;

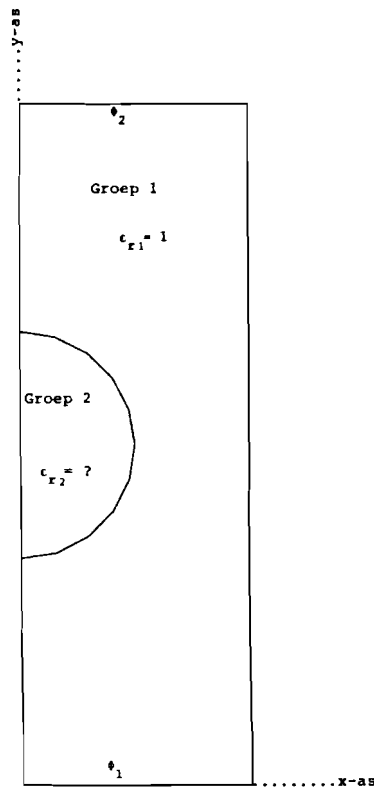
```

Opmerking: De file moet van het type data zijn.

4 VOORBEELDEN

Binnen de vakgroep EHO was de toevoeging van de repetitie aan de bestaande software nodig om de oplading van een stofdeeltje in een corona omgeving te berekenen. Het stofdeeltje bevindt zich in een elektrisch veld. Het deeltje kan zowel geleidend als niet-geleidend zijn.

Van dit geheel moet een constructie gemaakt worden waarmee de aangepaste software kan gaan rekenen. Het elektrische veld wordt gesimuleerd door het stofdeeltje tussen twee platen, die een bepaalde spanning hebben, te plaatsen. Om het veld zo goed mogelijk te simuleren moeten de platen zo ver mogelijk van elkaar staan (homogeniteit). Als stofdeeltje is een bol genomen, die in de beginsituatie ongeladen is.



figuur 1: De constructie (de y-as is de rotatie-as, ϕ_1 en ϕ_2 zijn potentialen van de platen)

4.1 DE GELEIDENDE BOL

Voor een geleidende bol is gekozen omdat hiervoor een correcte formule voor de verzadingslading van het stofdeeltje bestaat. Hiermee kan een vergelijking gemaakt worden tussen de werkelijke waarde en de door het programma berekende waarde.

Als straal van de bol is 3×10^{-6} m genomen. De afstand tussen de twee platen is 66×10^{-6} m. De bol bevindt zich in dit voorbeeld in een elektrisch veld van 1×10^5 V/m, dus dit betekent een spanningsverschil van 6.6 V ($66 \times 10^{-6} \times 10^5$) tussen de platen. ($\phi_1 = 0$ V en $\phi_2 = 6.6$ V).

Bij TRIQUAMESH is het niet mogelijke zulke kleine getallen voor de basispunten in te vullen. Daarom zijn de coördinaten van de basispunten met een factor 10^6 vermenigvuldigd. In ELECTR/INTERFACE wordt dit hersteld (zie 1.3).

Als eerste wordt een niet al te fijne verdeling van de driehoekjes genomen. De verdeling is zo gekozen dat de lengte van de ribben van de driehoekjes bij het stofdeeltje een grootte van 0.2×10^{-6} m hebben en bij de platen iets groter. Er is voor een niet al te fijne verdeling gekozen omdat dit minder rekentijd kost en al een eerste vergelijking kan worden gemaakt.

De karakteristieken van de constructie zijn de permittiviteit en de ladingsdichtheid per groep. Groep 1 is wat buiten de bol zit en groep 2 is de bol zelf. Omdat de bol geleidend is moet voor de ϵ_{r2} een groot getal gekozen worden, bv $\epsilon_{r2} = 10^5$. Dit betekent dat de permittiviteit van de bol 8.85×10^{-7} As/Vm ($\epsilon_0 \times \epsilon_{r2}$, met $\epsilon_0 = 8.85 \times 10^{-12}$ As/Vm) is. De permittiviteit buiten de bol is 8.85×10^{-12} As/Vm ($\epsilon_{r1} = 1$). In het begin zit er nog geen lading op de bol, dus de ladingsdichtheid is in de beginsituatie overal nul.

De bepaling van de tijdstappen is iets moeilijks. Te grote tijdstappen leiden tot onnauwkeurigheid, te kleine tot verspilling van rekentijd. Als gebruiker moet men het programma een aantal keren draaien, eventueel met een eenvoudigere constructie, en zo bepalen wat de tijdstappen moeten zijn. Een aantal keer draaien is ook nodig om te bekijken wat de invloed van de tijdstappen op het resultaat is. In dit voorbeeld is gekozen voor mogelijkheid 2 voor bepaling van de tijdstippen (zie 2.2.3), waarbij:

$dq = 2.5 \times 10^{-17}$ C en $mts = 1 \times 10^{-2}$ s.

De overige constanten zijn: $N0 = 5 \times 10^{14}$ m⁻³, $B = 2.1 \times 10^{-4}$ m²/Vs, maxstap = 20 en delta = 0.0001.

De berekeningsgang bestaat uit 5 stappen:

- 1) Invoerfile voor de meshgenerator maken.
- 2) Mesh genereren m. b. v. TRIQUAMESH.
- 3) Randvoorwaarden toevoegen en invoer voor ELECTR/REPFEMSYS aanmaken. Dit kan gedaan worden m. b. v. ELECTR/INTERFACE of ELECTR/INTERFACE/NI.
- 4) Een aantal eindige elementen berekeningen. Door het programma ELECTR/REPFEMSYS te runnen.

5) Grafische weergave van een aantal resultaten m. b. v. het interactieve USER/DRAW.

Deze 5 stappen worden één voor één doorlopen. Hierbij is vanuitgegaan dat de programma's ELECTR/INTERFACE, ELECTR/INTERFACE/NI, ELECTR/ROTELEMENT, ELECTR/REPFEMSYS en ELECTR/RFJOB onder de gebruiker zijn nummer aanwezig zijn. Indien dit niet het geval is dan kan hij de programma's met het volgende kommando ophalen:

COPY (ELEHAC7) naamprogramma AS naamprogramma

Stap 1: Aanmaken van de invoerfile voor TRIQUAMESH.

Deze file moet van het type data zijn. Voor wat er allemaal in moet staan zie hoofdstuk 1.1 en [KOE83]. Voor het vervolg wordt aangenomen dat deze file de naam VOORBEELD1 heeft. (Voor de tekst van deze file zie appendix A5).

Stap 2: Genereren van de mesh.

Dit kan gedaan worden met het volgende commando:

```
E$(WWTMGA1)TRIQUAMESH ON USER5; FILE DATAMG = VOORBEELD1
```

Indien de naam van de invoerfile DATAMG was i. p. v. VOORBEELD1 dan kan het stuk na de ";" worden weggelaten. Dit programma kan ook vanachter een grafische terminal worden gestart, hiermee is het mogelijk om plaatjes op het beeldscherm te laten tekenen. De elementverdeling wordt in de uitvoerfile TRIQUAMESH datum/structuurnaam/SAFERESULTS geschreven.

Voor de structuurnaam is GBOL gekozen (zie invoerfile VOORBEELD1).

De gebruiker kan TRIQUAMESH in een job zetten. Deze job kan worden aangeroepen door het kommando:

```
START naamjob (naamjob is de naam van de job).
```

In deze job moet TIQUAMESH worden aangeroepen door het kommando:

```
RUN (WWTMGA1)TRIQUAMESH ON USER5; FILE DATAMG = ???
```

Stap 3: Invoeren van de randvoorwaarden.

Dit kan gedaan worden met behulp van één van de volgende twee programma's:

- ELECTR/INTERFACE

De files VOORBEELD1 en .../SAFERESULTS zijn invoerfiles voor dit programma. Het programma kan gestart worden met behulp van de volgende kommando's:

```
GET ELECTR/INTERFACE
```

```
E
```

Voor de interface dialoog zie appendix A6.

- ELECTR/INTERFACE/NI

De files VOORBEELD1 en .../SAFERESULTS zijn hier ook weer invoerfiles voor het programma. Omdat dit programma niet-interactief is moet er nog een invoerfile, van het type data, zijn waar de gegevens in komen te staan (zie hoofdstuk 3). Laat de naam van deze file INTFVB1 zijn. Voor de inhoud van deze file zie appendix A7.

Dit programma kan gestart worden met de volgende kommando's:

```
GET ELECTR/INTERFACE/NI
```

```

E; FILE DATAMG = VOORBEELD1;FILE MESH = .../SAFERESULTS; %
FILE INTERF = INTFVB1
(% is nodig indien het kommando niet op een regel kan)
De twee programma's leveren de uitvoerfiles INREADFEM,
INREADREFEM1 en INREADREFEM2 op.

```

Stap 4: De repetitie.

De repetitie staat in het programma ELECTR/REFFEMSYS. De files INREADFEM, INREADREFEM1 en INREADREFEM2 zijn invoerfiles voor dit programma.

Het programma kan op twee manieren worden gestart, namelijk:

- Met de volgende kommando's:

```

GET ELECTR/ROTELEMENT
C
SAVE
GET ELECTR/REFFEMSYS
E

```

Voor ELECTR/REFFEMSYS is het van belang dat de gecompileerde versie van ELECTR/ROTELEMENT aanwezig is.

- Met de volgende kommando's:

```

GET ELECTR/RFJOB
% eventueel aanpassen job
SAVE
START

```

Voor de job moeten de programma's ELECTR/ROTELEMENT en ELECTR/REFFEMSYS aanwezig zijn.

De job ziet er als volgt uit:

```

ELECTR/RFJOB:
100 ?BEGIN JOB DORSSELAER/VELDBEELD;
200  QUEUE = ?;
      {Op het vraagteken moet de gebruiker het
      queuenummer invullen:
      - voor een dagjob: ? = 3,
      - voor een nachtjob: ? = 5 .}
300  MAXPROCTIME = ?;
      { Hier moet de gebruiker de maximale rekestijd
      invullen}
400  INSTRUCTION1
500  RESOURCES:
600      PROCESSTIME: LIMIT      = ? SECS.
      {bovengrens voor de rekestijd}
700      ESTIMATED = ? SECS.
      {verwachte rekestijd}
800  COMPILER OBJECT/ELECTR/ROTELEMENT WITH ALGOL LIBRARY;
900  COMPILER FILE CARD (KIND = DISK,
1000      TITLE = ELECTR/ROTELEMENT);
1100  IF file OBJECT/ELECTR/REFFEMSYS IS RESIDENT THEN
1200  BEGIN
1300      RUN OBJECT/ELECTR/REFFEMSYS
1400  END ELSE
1500  BEGIN
1600      COMPILER ELECTR/REFFEMSYS WITH ALGOL;
1700      COMPILER FILE CARD (KIND = DISK,
1800          TITLE = ELECTR/REFFEMSYS)

```



```

1900 END:
2000 REMOVE OBJECT/ELECTR/ROTELEMENT
2000 ?END JOB

```

Een indruk van wat de gebruiker voor MAXPROCTIME moet invullen:

1487 driehoekjes, 846 knooppunten en 20 stappen geeft:
 processtime = 420 (= 840 processorunits).

Een gebruiker kan zelf eventueel een job samenstellen, waarbij bijvoorbeeld ELECTR/REPFEMSYS een aantal keren wordt gedraaid. Hij kan zelfs de programma's TRIQUAMESH en ELECTR/INTERFACE/NI er in zetten zodat alles in één job staat.

De volgende uitvoerfiles zijn gecreëerd (zie 2.5):

- OUTREPFEM (zie appendix A8)
- OUTER (zie appendix A9)
- CIRF
- OUTREADFEM
- OUTO
- GBOLO1/S/GROUP/A/COORD
- GBOLO1/S/GROUP/A/COMPNODES
- GBOLO1/S/GOUP/A/COMPNODES
- GBOLO1/OO0001/X1/S/GROUP/A/NODEDER
- GBOLO1/OO0002/X1/S/GROUP/A/NODEDER
- ...
- ...
- GBOLO1/OO0020/X1/S/GROUP/A/NODEDER

Stap 5: Maken van plots.

Met behulp van het programma USER/DRAW. Het programma is interactief: telkens verschijnt een vraag of commentaar op het beeldscherm waarop een antwoord wordt verwacht. De vragen worden veelal aangeboden in de vorm van menu's (lijsten). Uit de lijst van gepresenteerde onderwerpen kan de gebruiker er één kiezen. Voor sommige onderwerpen zijn menu's op diverse niveau's gekonstrueerd. Voor meer gevens zie [BEU83]. Van de beginsituatie en de eindsituatie is een plotje getekend. Voor de dialoog met USER/DRAW en de plotjes zie appendix A10.

Resultaten:

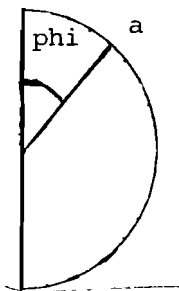
Het programma ELECTR/REPFEMSYS levert de resultaten op. Voor de normaalcomponent E_r van een ongeladen geleidende bol bestaat een formule (zie [LAA87]):

Laat a een punt op het deeltje zijn, dan geldt voor E_r in punt a:

$$E_r = -3 * E_0 * \cos(\phi)$$

met E_0 het elektrische veld en
 ϕ de hoek tussen de y-as en
 de lijn door het punt a en
 het middelpunt van de bol.

Opmerking: Een negatieve normaalcomponent wil zeggen dat het veld van de bol af is gericht.



In de volgende tabel is de werkelijke waarde van E_r en de uit het programma berekende waarde van E_r voor een aantal hoeken uitgezet ($E_0 = 1 \cdot 10^5$ V/m):

Tabel:

phi	E_r (formule) [V/m]	E_r (programma) [V/m]
2°	$-3.00 \cdot 10^5$	$-2.68 \cdot 10^5$
32°	$-2.54 \cdot 10^5$	$-2.32 \cdot 10^5$
46°	$-2.08 \cdot 10^5$	$-1.84 \cdot 10^5$
60°	$-1.50 \cdot 10^5$	$-1.44 \cdot 10^5$
88°	$-0.09 \cdot 10^5$	$-0.09 \cdot 10^5$
92°	$0.09 \cdot 10^5$	$0.05 \cdot 10^5$
120°	$1.50 \cdot 10^5$	$1.41 \cdot 10^5$
134°	$2.08 \cdot 10^5$	$1.96 \cdot 10^5$
148°	$2.54 \cdot 10^5$	$2.32 \cdot 10^5$
178°	$3.00 \cdot 10^5$	$2.68 \cdot 10^5$

Voor de verzadigingslading Q_s van een geleidende bol bestaat ook een formule:

$$Q_s = 12 \cdot \pi \cdot \epsilon_0 \cdot E_0 \cdot r^2 \quad \text{met } r = \text{straal bol en} \\ \epsilon_0 = 8.85 \cdot 10^{-12} \text{ As/Vm}$$

In dit voorbeeld:

met de formule uitgerekende waarde: $Q_s = 3 \cdot 10^{-16}$ C.

met het programma uitgerekende waarde: $Q_s = 2.55 \cdot 10^{-16}$ C

De door het programma berekende E_r en Q_s wijkt nog al wat van de werkelijke waarden af. Deze afwijkingen zijn te verklaren uit de eindigheid van de elementen. Daarom is geprobeerd de verdeling zo fijn mogelijk te maken. Dit leverde nogal wat problemen op. De constructie moest enigszins aangepast worden. De fijnst mogelijke verdeling is de verdeling zodat de lengte van de ribben van de driehoekjes die aan de bol grenzen een grootte van $0.05 \cdot 10^{-6}$ m hebben en bij de platen iets groter. Kleiner kon niet meer omdat het dan veel te veel rekentijd ging kosten. Met deze kleinere verdeling is een nieuwe berekening gestart. Voor de file DATAMG en de file INTERFACE zie appendix A11.

Tabel van normaalcomponenten in nieuwe situatie:

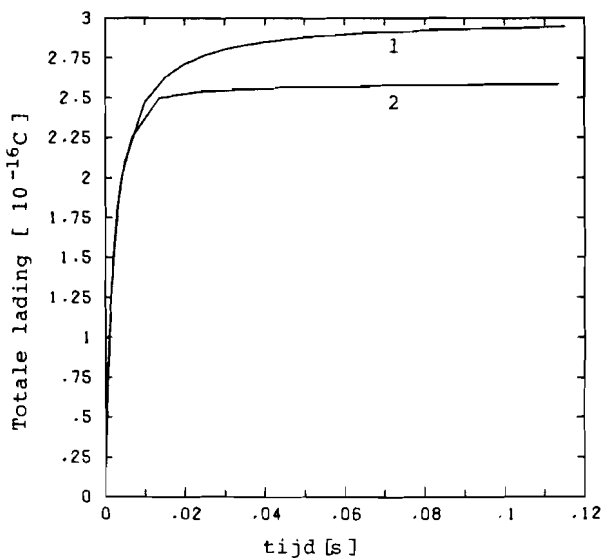
phi	E_r (formule) [V/M]	E_r (programma) [V/M]
1	-3.00×10^5	-2.87×10^5
30	-2.60×10^5	-2.53×10^5
45	-2.12×10^5	-2.10×10^5
60	-1.50×10^5	-1.42×10^5
89	-0.02×10^5	-0.06×10^5
91	0.02×10^5	0.06×10^5
120	1.50×10^5	1.45×10^5
135	2.12×10^5	2.07×10^5
150	2.60×10^5	2.51×10^5
179	3.00×10^5	2.86×10^5

De berekende Q_s is nu 2.60×10^{-16} C. Er is nog altijd een afwijking tussen de berekende en de werkelijke waarden maar iets minder dan bij de grovere verdeling.

Voor de totale lading $Q(t)$ op tijdstip t bestaat een formule:
 $Q(t) = Q_s * t / (t + \tau)$ met $\tau = 4 * \epsilon_0 / (N_0 * B * q_0)$
 en $q_0 = 1.6 \times 10^{-19}$ C.

De werkelijke oplading en de met het programma berekende oplading staan in figuur 2.

OPLADING GELEIDENDE BOL



figuur 2: lijn 1: met de formule berekende oplading,
 lijn 2: met het programma berekende oplading.

4.2 DE NIET-GELEIDENDE BOL

Voor de verzadigingslading van een niet-geleidende bol bestaat ook een formule, maar deze formule is mogelijk niet goed. Daarom is het interessant om het programma voor een niet-geleidende bol te draaien.

In dit voorbeeld is gekozen voor $\epsilon_{r2} = 5$, $E_0 = 10^5$ V/m, $r = 3 \cdot 10^{-6}$ m, $dq = 10^{-17}$ C, $\text{mts} = 10^{-3}$ s, $N_0 = 5 \cdot 10^{14}$ m⁻³, $B = 2.1 \cdot 10^{-4}$ m²/Vs, $\text{maxstap} = 20$, $\text{delta} = 10^{-4}$ en voor de kleinste mogelijk verdeling. Voor het verloop zie de geleidende bol.

Resultaten:

Voor een niet-geleidende bol bestaat ook een formule voor de normaalcomponenten in de beginsituatie:

$$E_r = -E_0 \cdot \cos(\text{phi}) \cdot (3 \cdot \epsilon_{r2} / (\epsilon_{r2} + 2))$$

(Formule E_r zie [LAA87]).

Tabel:

phi	E_r (formule) [V/M]	E_r (programma) [V/M]
1	$-2.14 \cdot 10^5$	$-2.08 \cdot 10^5$
30	$-1.86 \cdot 10^5$	$-1.82 \cdot 10^5$
45	$-1.52 \cdot 10^5$	$-1.50 \cdot 10^5$
60	$-1.07 \cdot 10^5$	$-1.02 \cdot 10^5$
89	$-0.04 \cdot 10^5$	$-0.05 \cdot 10^5$
91	$0.04 \cdot 10^5$	$0.05 \cdot 10^5$
120	$1.07 \cdot 10^5$	$1.05 \cdot 10^5$
135	$1.52 \cdot 10^5$	$1.50 \cdot 10^5$
150	$1.86 \cdot 10^5$	$1.81 \cdot 10^5$
179	$2.14 \cdot 10^5$	$2.08 \cdot 10^5$

De met het programma berekende normaalcomponenten wijken maar weinig af van de met de formule berekende normaalcomponenten.

Voor de verzadigingslading van een niet-geleidende bol bestaat een formule, maar deze formule is mogelijk niet goed:

$$Q_s = 12 \cdot \pi \cdot \epsilon_0 \cdot E_0 \cdot r^2 \cdot (\epsilon_{r2} / (\epsilon_{r2} + 2))$$

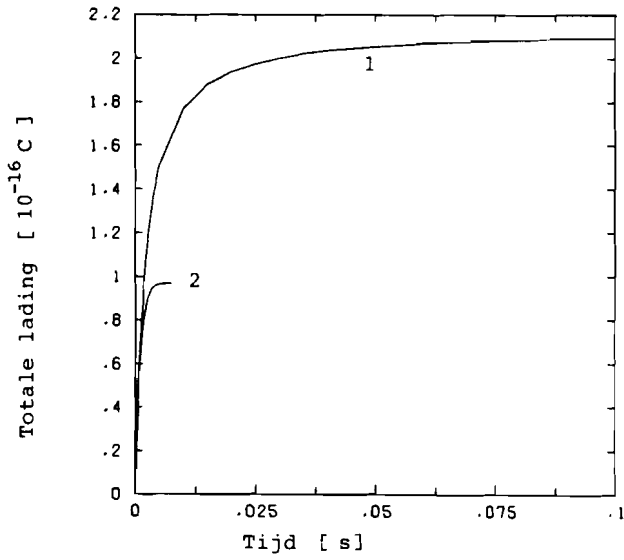
Voor $r = 3 \cdot 10^{-6}$ m en $E_0 = 10^5$ V/m levert dit :

met de formule uitgerekende waarde: $Q_s = 2.14 \cdot 10^{-16}$ C,

met het programma uitgerekende waarde: $Q_s = 0.97 \cdot 10^{-16}$ C.

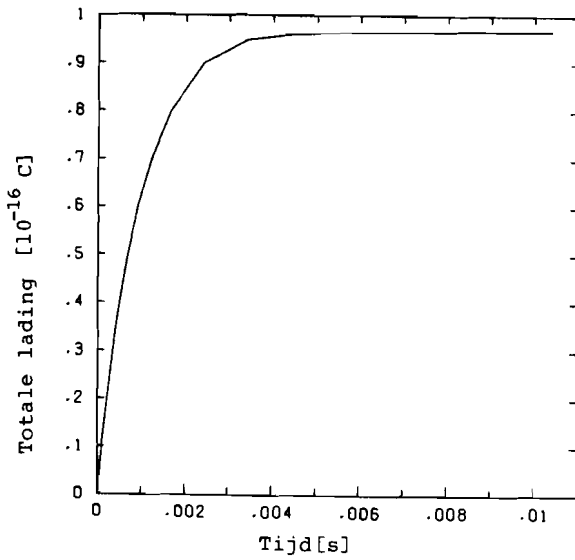
Het verloop van de oplading is nu ook weer in een grafiek uitgezet (zie figuur 3 en 4).

OPLADING NIET-GELEIDENDE BOL



figuur 3: lijn 1: met de formule berekende oplading,
lijn 2: met het programma berekende oplading.

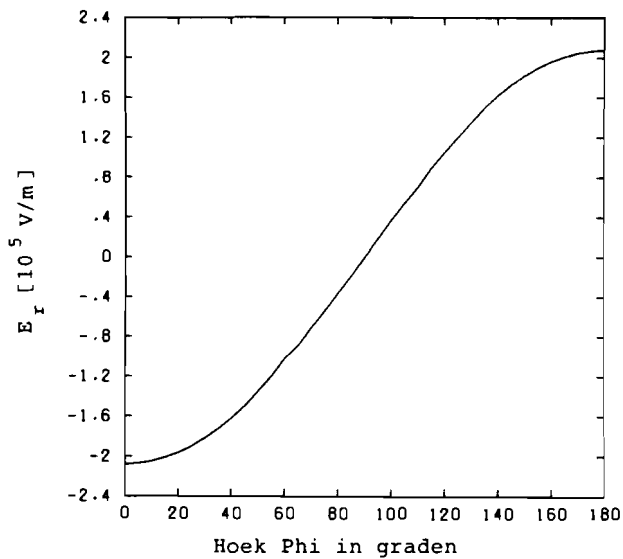
OPLADING NIET-GELEIDENDE BOL



figuur 4: met het programma berekende oplading.

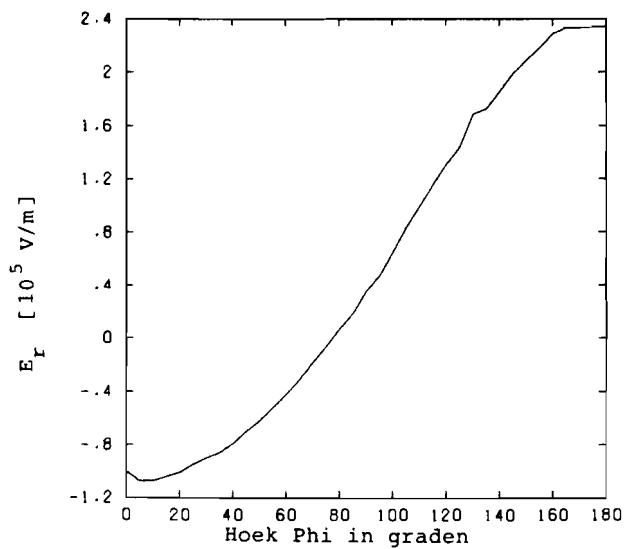
In de figuren 5 t/m 7 zijn de met het programma berekende normaalcomponenten op een aantal tijdstippen tegen de hoek phi uitgezet. In de figuren 8 t/m 10 is de oppervlakte-ladingsdichtheid op een aantal tijdstippen tegen de hoek phi uitgezet.

NORMAALCOMPONENTEN NIET-GELEIDENDE BOL



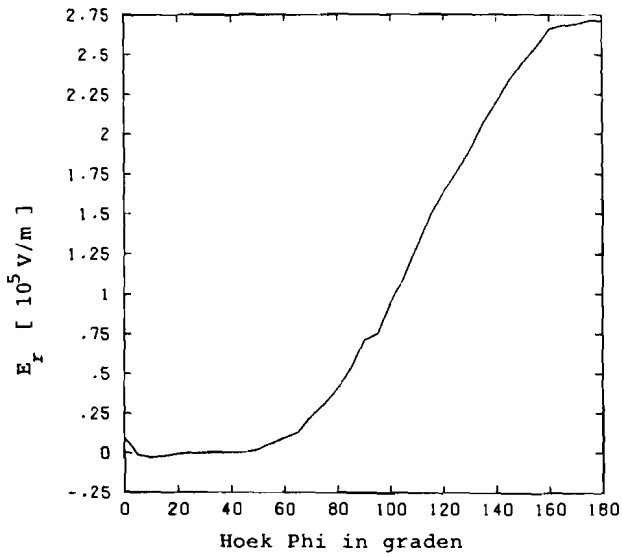
figuur 5: op tijdstip $t = 0$ s.

NORMAALCOMPONENTEN NIET-GELEIDENDE BOL



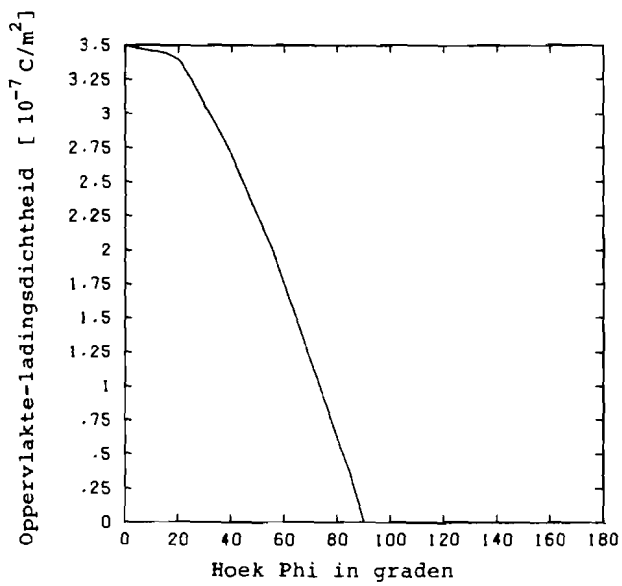
figuur 6: op tijdstip $t = 0.7 \cdot 10^{-3}$ s.

NORMAALCOMPONENTEN NIET-GELEIDENDE BOL



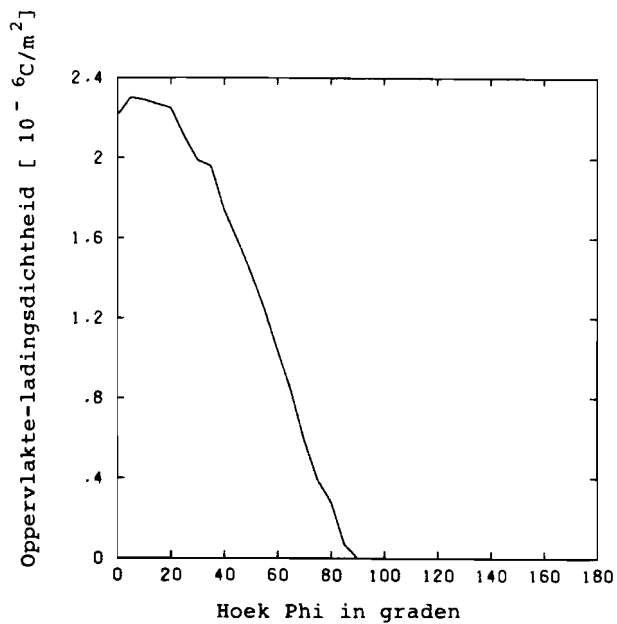
figuur 7: op tijdstip $t = 9 \times 10^{-3}$ s.

OPPERVLAKTE-LADINGSDICHTHEID NIET-GELEIDENDE BOL



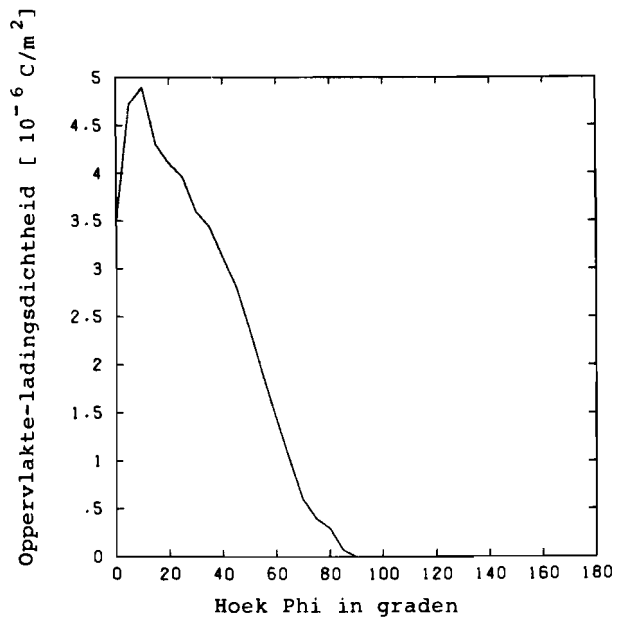
figuur 8: op tijdstip $t = 0.1 \times 10^{-3}$ s.

OPPERVLAKTE-LADINGSDICHTHEID NIET-GELEIDENDE BOL



figuur 9: op tijdstip $t = 0.7 \times 10^{-3}$ s.

OPPERVLAKTE-LADINGSDICHTHEID NIET-GELEIDENDE BOL



figuur 10: op tijdstip $t = 9 \times 10^{-3}$ s.

4.3 CONCLUSIE

De conclusie, die uit de resultaten van de geleidende en niet-geleidende bol getrokken kan worden, is dat de tot nu toe gebruikte formule voor de verzadigingslading van een niet-geleidende bol onjuist is. Bij de geleidende bol is er een afwijking t.o.v. de formule in de Qs van 15% en bij de niet geleide bol een afwijking van 55%. De afwijking van 15% kan verklaard worden uit de grofheid van de opsplitsing in elementen. Voor de niet-geleidende bol is de afwijking echter te groot om op deze wijze verklaard te worden.

Deze conclusie is in de publicatie "PROCESSES IN PULSED ELECTROSTATIC PRECIPITATORS, Submitted to: The 5th. International Symposium On High Voltage Engineering, Branschweig, August 1987" verwerkt.

De verzadigingslading van bollen, met $E_0 = 10^5$, wordt in figuur 11 getoond.

Lijn A geeft de straal-afhankelijkheid van de verzadigingslading Qs voor geleidende bollen volgens de correcte formule:

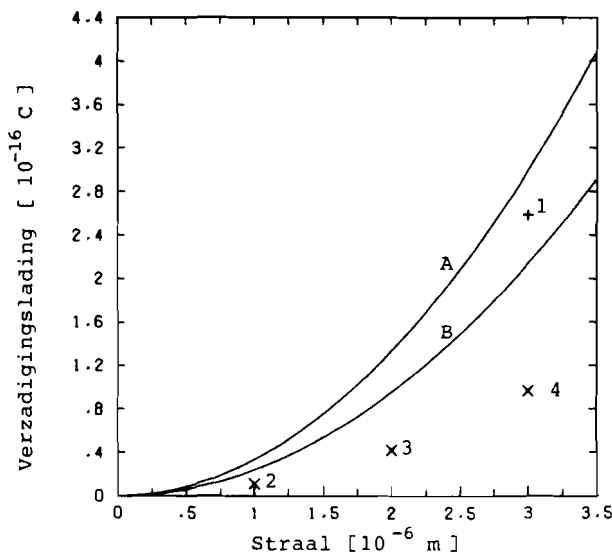
$$Q_s = 12 * \pi * \epsilon_0 * E_0 * r^2, \text{ met } r = \text{straal bol.}$$

Lijn B geeft de straal-afhankelijkheid van de verzadigingslading Qs voor niet-geleidende bollen, met $\epsilon_r = 5$, volgens de formule:

$$Q_s = 12 * \pi * \epsilon_0 * E_0 * r^2 * (\epsilon_r / (\epsilon_r + 2)), \text{ met } r = \text{straal bol.}$$

Punt 1 geeft de verzadigingslading voor een geleidende bol met $r = 3 \times 10^{-6}$, berekend door het programma.

De punten 2, 3 en 4 geven de verzadigingslading voor niet-geleidende bollen met $\epsilon_r = 5$, $r = 1 \times 10^{-6}$, 2×10^{-6} en 3×10^{-6} , berekend door het programma.



figuur 11: verzadigingslading voor geleidende en niet-geleidende bollen.

LITERATUURLIJST

- [BEU83] L. v. Beukering, L Braak, G. Peters:
"Aanroep van programma's voor berekeningen
gebaseerd op de eindige elementenmethode",
Collegedictaat, Technische Universiteit
Eindhoven, 1983.
- [EWS81] Earl. W. Swokowski: CALCULUS With Analytic
Geometry, Prindle, Weber & Schmidt, Boston,
Massachusetts, 1981.
- [HEM84] C. Hemerik: Collegedictaat Compilers 1,
Technische Universiteit Eindhoven, 1984.
- [KOE83] J. M. V. A. Koelman: "FINITE ELEMENTS IN ELECTRIC
FIELD PROBLEMS", TUE-rapport, Technische
Universiteit Eindhoven, 1983.
- [LAA87] P. v. d. Laan:
Collegedictaat Hoogspanningstechniek 1,
Technische Universiteit Eindhoven, 1987.
- [PET76] F. J. Peters: "Femsys, een systeem voor op de
eindige elementen methode gebaseerde
berekeningen, deel I", TUE-rapport, Technische
Universiteit Eindhoven, 1976.

APPENDICES

A1 AFLEIDING FORMULE OPPERVLAKTE GEBIED

Afleiding van de formule voor de oppervlakte van het lichaam dat ontstaat na wenteling van een lijnstuk a,b om de y-as (Opp).

Laat (x_a, y_a) de coördinaten van punt a zijn, met $x_a \geq 0$.

Laat (x_b, y_b) de coördinaten van punt b zijn, met $x_b \geq 0$.

Formule voor Opp:

1) $x_a < x_b$

{formule oppervlakte-integraal, zie [ESW81]}

$$\text{Opp} = \text{abs} \left(2\pi \int_{x_a}^{x_b} (x \cdot \sqrt{1 + f'(x)^2}) dx \right)$$

met f is de functie van de rechte lijn tussen de punten a en b.

$$\{ f'(x) = (y_b - y_a) / (x_b - x_a) \}$$

$$\text{Opp} = \text{abs} \left(2\pi \cdot \frac{x^2}{2} \cdot \sqrt{1 + ((y_b - y_a) / (x_b - x_a))^2} \right) \Big|_{x_a}^{x_b}$$

$$\text{Opp} = \text{abs} \left(\pi \cdot (x_b^2 - x_a^2) \cdot \sqrt{1 + ((y_b - y_a) / (x_b - x_a))^2} \right)$$

2) $x_a = x_b$

{oppervlakte van een cylinder}

$$\text{Opp} = \text{abs} \left(2\pi \cdot x_a \cdot (y_b - y_a) \right)$$

A2 AFLEIDING FORMULE VOLUME GEBIED

Afleiding van de formule voor het volume van het lichaam dat ontstaat na wenteling van een driehoek om de y-as (Vol).

Laat a, b, c de driehoekspunten zijn.

Laat (xa, ya) de coördinaten van punt a zijn, met xa ≥ 0.

Laat (xb, yb) de coördinaten van punt b zijn, met xb ≥ 0.

Laat (xc, yc) de coördinaten van punt c zijn, met xc ≥ 0.

Eerst de afleiding van de formule van de volume-integraal van het lichaam dat ontstaat na wenteling van een lijnstuk d, e om de y-as (v(d, e)).

Laat (xd, yd) de coördinaten van punt d zijn, met xd ≥ 0.

Laat (xe, ye) de coördinaten van punt e zijn, met xe ≥ 0.

Formule voor v(d, e):

1) yd = ye

$$v(d, e) = 0$$

2) yd <> ye

{formule volume-integraal, zie [ESW81]}

$$v(d, e) = \pi \int_{yd}^{ye} (f(y))^2 dy$$

met f is de functie van de rechte lijn tussen de punten d en e.

$$\{ f(y) = m * y + n \text{ met } m = (xd - xe) / (yd - ye) \text{ en } n = xd - m * yd \}$$

$$v(d, e) = \pi \int_{yd}^{ye} (m * y + n)^2 dy$$

{2 gevallen}

1) m = 0 (xd = xe)

$$v(d, e) = \pi \int_{yd}^{ye} n^2 dy$$

$$v(d, e) = \pi * n^2 * y \Big|_{yd}^{ye}$$

$$v(d, e) = \pi * xd * (ye - yd)$$

2) m <> 0 (xd <> xe)

$$v(d, e) = \pi \int_{yd}^{ye} (m * y + n)^2 dy$$

$$v(d, e) = \pi * 1 / (3 * m) * (m * y + n)^3 \Big|_{yd}^{ye}$$

$$v(d, e) = \pi / (3 * m) * ((m * ye + n)^3 - (m * yd + n)^3)$$

$$v(d, e) = \pi / (3 * m) * (xe^3 - xd^3)$$

$$v(d, e) = \pi / 3 * ((yd - ye) / (xd - xe)) * (xe^3 - xd^3)$$

Formule voor Vol: Vol = abs(v(a, b) + v(b, c) + v(c, a)).

A3 FORMULES VOOR ELECTRISCHE VELD

Formules voor de x en y component van het elektrisch veld in een driehoek (Ex en Ey).

Laat a, b, c de driehoekspunten zijn.

Laat (xa, ya) de coördinaten van punt a zijn.

Laat (xb, yb) de coördinaten van punt b zijn.

Laat (xc, yc) de coördinaten van punt c zijn.

Laat ua het potentiaal in punt a zijn.

Laat ub het potentiaal in punt b zijn.

Laat uc het potentiaal in punt c zijn.

$$Ex = -((ua-ub)*(ya-yc)-(ua-uc)*(ya-yb))/$$
$$((xa-xb)*(ya-yc)-(xa-xc)*(ya-yb))$$

$$Ey = -((ua-ub)*(xa-xc)-(ua-uc)*(xa-xb))/$$
$$((ya-yb)*(xa-xc)-(ya-yc)*(xa-xb)).$$

De formules voor Ex en Ey zijn uit [KOE83] gehaald.

A4 AFLEIDING FORMULE HOEK ALFA

Afleiding van de formule voor de hoek α tussen de vector n en de vector x . De vector n staat loodrecht op een lijnstuk a, b en is gericht naar een punt c . De vector x is evenwijdig met de x -as en heeft een positieve richting.

Laat l de lijn door de punten a en b zijn.

Laat m de lijn door het punt c en loodrecht op lijnstuk a, b zijn.

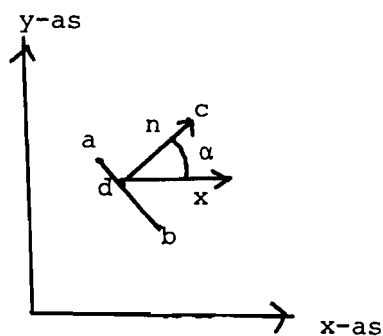
Laat d het snijpunt van l en m zijn.

Laat (x_a, y_a) de coördinaten van punt a zijn.

Laat (x_b, y_b) de coördinaten van punt b zijn.

Laat (x_c, y_c) de coördinaten van punt c zijn.

Laat (x_d, y_d) de coördinaten van punt d zijn.



Eerst volgt de berekening van x_d en y_d :

De lijn l door a en b heeft parametervoorstelling:

$$l: \begin{bmatrix} x_a \\ y_a \end{bmatrix} + \mu \begin{bmatrix} x_b - x_a \\ y_b - y_a \end{bmatrix}$$

Stel $x_d = x_a + \tau(x_b - x_a)$ en $y_d = y_a + \tau(y_b - y_a)$
{punt d ligt op de lijn l }

De lijn m door de punten c en d heeft parametervoorstelling:

$$m: \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \mu \begin{bmatrix} x_c - (x_a + \tau(x_b - x_a)) \\ y_c - (y_a + \tau(y_b - y_a)) \end{bmatrix}$$

Lijn l staat loodrecht op lijn m, dus inproduct = 0:

$$\begin{bmatrix} \begin{bmatrix} x_b - x_a \\ y_b - y_a \end{bmatrix} \\ \begin{bmatrix} x_c - x_a - r(x_b - x_a) \\ y_c - y_a - r(y_b - y_a) \end{bmatrix} \end{bmatrix} = 0$$

{regels inproduct}

$$\begin{bmatrix} \begin{bmatrix} x_b - x_a \\ y_b - y_a \end{bmatrix} \\ \begin{bmatrix} x_c - x_a \\ y_c - y_a \end{bmatrix} \end{bmatrix} - r \begin{bmatrix} \begin{bmatrix} x_b - x_a \\ y_b - y_a \end{bmatrix} \\ \begin{bmatrix} x_b - x_a \\ y_b - y_a \end{bmatrix} \end{bmatrix} = 0$$

$$r = ((x_b - x_a) * (x_c - x_a) + (y_b - y_a) (y_c - y_a)) / ((x_b - x_a)^2 + (y_b - y_a)^2)$$

Hieruit volgen de coördinaten van punt d.

Formule voor hoek α :

Laat $dy = y_c - y_d$ en $dx = x_c - x_d$,

- 1) $\alpha = \arctan (dy/dx)$, voor $dx > 0$,
- 2) $\alpha = \arctan (dy/dx) + \pi$, voor $dx < 0$,
- 3) $\alpha = \pi/2$, voor $dx = 0$ en $dy > 0$,
- 4) $\alpha = -\pi/2$, voor $dx = 0$ en $dy < 0$.

AS VOORBEELD FILE DATAMG

INVOERFILE VOORBEELD1:

```
100$INPUT "GBOL"  
200$BASISPOINTS  
300      1:  0,  0;  2:  9,  0;  
400      3:  9, 66;  4:  0, 66;  
500      5:  0, 36;  6:  0, 33;  
600      7:  0, 30;  8:  3, 33;  
700$CONTOURPIECES  
800      1(4RL5CM-6,8CM-6,7RL1RL2RL3RL4)  
900      2(5RL6RL7CM6,8CM6,5)  
1000$SUBCONTOURS  
1100     1(1)  
1200     2(2)  
1300$SUBSTRUCTURES  
1400     FEMSYS  
1500     1TRIM3(1,2)  
1600$GRADING  
1700     G1 RI 1  
1800     GV .2(5,7,8)  
1900$RENUMBER  
2000     PROFILE(1LINE 1,2)  
2100$STOP
```


A6 VOORBEELD INTERF DIALOOG MET ELECTR/INTERFACE

(U=user)

U: get ELECTR/INTERFACE

U: E

#RUNNING 8404

? TITEL VAN FILE DATAMG

#?

U: VOORBEELD1

? TITEL VAN FILE MESH (DE GEGENEREERDE MESH)

U: TRIQUAMESH060187/GBOL/SAFERESULTS

? IDENTIFICATIETEGEL VOOR FEMSYS-UITVOER

* (WANNEER DEZE REGEL NIET LEEG IS DAN WORDEN DE EERSTE

* 6 KARAKTERS NIET MEER EN NIET MINDER) GEBRUIKT ALS

* DIRECTORY VOOR DOOR FEMSYS GECREEERDE FILES;

* GEBRUIK BIJ VOORKEUR EEN SPREKENDE TEXT)

U: GBOL01

? AANTAL KNOOPPUNTEN, AANTAL ELEMENTEN (BOVENSCHATTINGEN)

U: 900, 1500

? COORDINATEN VAN DE BASISPUNTEN (Y/N)

U: N

* STRUCTUUR 1

* 2 GROEP(EN)

* GROEP 1 : 1208 ELEMENTEN, 3 KNOPEN PER ELEMENT

* GROEP 2 : 270 ELEMENTEN, 3 KNOPEN PER ELEMENT

GEEF 1 ALS HET PROBLEEM PUUR CAPACITIEF IS

GEEF 2 ALS HET PROBLEEM PUUR RESISTIEF IS

U: 1

? PERMITTIVITY(ϵ_s/ϵ_m) VOOR GROEP 1

U: 8.85E-12

? CHARGE DENSITY(ρ_s/m^3) VOOR GROEP 1

U: 0

? PERMITTIVITY(ϵ_s/ϵ_m) VOOR GROEP 2

U: 8.85E-07

? CHARGE DENSITY(ρ_s/m^3) VOOR GROEP 2

U: 0

? WILT U DE COORDINATEN AANPASSEN (Y/N)

U: Y

VERMENIGVULDIGINGSFACTOR

U: 1E-6

? WILT U UITLEG OVER DE RANDEN

U: Y

* U WORDT WAARSCHIJNLIJK VERZOCHT WAARDEN

* IN TE TIKKEN VOOR GROEP, VAN, VIA, NAAR.

* 'VAN', 'VIA' EN 'NAAR' ZIJN BASISPUNTNUMMERS

* U GEEFT DAARMEE EEN DEELRAND AAN.

* IS GEEN TUSSEN PUNT BESCHIKBAAR OM DE RICHTING VAST TE

* LEGGEN DAN TREEDT DE VOLGENDE REGELING IN WERKING :

* MET 'VAN, VAN, NAAR' BEDDELEN WE HET KORTSTE TRAJECT

* VAN 'VAN' NAAR 'NAAR'

* MET 'VAN, NAAR, NAAR' HET LANGSTE TRAJECT.

* VOOR GROEP KUNT U BIJNA ALTIJD VOLSTAAN MET DE WAARDE

* 0 (NUL).
 * ALLEEN WANNEER EEN BASISPUNT OP GROND VAN ZIJN
 * COORDINATEN NIET EENDUIDIG GELOCALISEERD KAN WORDEN,
 * MOET HET NUMMER VAN DE GROEP, WAARTOE DE DEELRAND
 * BEHOORT, WORDEN INGEVOERD.
 * U KUNT EEN OVERZICHT VAN ALLE RANDKNOOPNUMMERS KRIJGEN.
 * SUCCES !
 ? WILT U OPEENVOLGENDE RANDPUNTNUMMERS ZIEN (Y/N)
 U: N
 ? RANDELEMENTEN
 U: N
 ? KOPPELEN
 U: Y
 ? GROEP1, GROEP2, VAN, VIA, NAAR
 U: 1,2,5,8,7
 ? KOPPELEN
 U: N
 ? ZWEVENDE ELEKTRODEN
 U: N
 ? VOORGESCHREVEN RANDSPANNINGEN
 U: Y
 GROEP, VAN, VIA, NAAR, WAARDE SPANNING
 U: 1,3,3,4,6.6
 ? MEER VOORGESCHREVEN SPANNINGEN
 U: N
 ? GEAARDE RANDEN
 U: Y
 ? GROEP, VAN ,VIA, NAAR
 U: 1,1,1,2
 ? MOETEN DE RESULTATEN GEPLIT WORDEN (Y/N)
 U: Y
 ? VERANDERLIJKE LADINGSDICHTHEID AAN RAND
 U: Y
 ? WILT U UITLEG (Y/N)
 U: Y
 * U WORDT DADELIIK VERZOCHT WAARDEN IN TE TIKKEN VOOR:
 * GROEP1, GROEP2, VAN, VIA, NAAR.
 * VOOR GROEP1 DIEN T U DE GROEP IN TE VULLEN, DIE DE
 * ELEMENTEN BEVAT WAARVOOR DE NORMAALCOMPONENT VAN HET
 * ELECTRISCHE VELD MOET WORDEN BEREKEND.
 * VOOR GROEP2 DIEN T U DE GROEP IN TE VULLEN DIE DE
 * ELEMENTEN BEVAT WAAR DE LADING NAAR TOE STROOMT.

 ? GROEP1, GROEP2, VAN, VIA, NAAR
 U: 1,2,5,8,7
 ? VERANDERLIJKE LADINGSDICHTHEID AAN RAND
 U: N
 ? MAXIMAAL DOOR REPFEMSYS UIT TE VOEREN STAPPEN
 U: 20
 ? GEEF WAARDE VOOR DE IONENCONCENTRATIE 'NO' (m-3)
 U: SE14
 ? GEEF WAARDE VOOR DE BEWEEGLIJKHEID 'B' (m2/Vs)
 U: 2.1E-4
 ? WILT U VARIABELE TIJDSTAPPEN (Y/N)
 U: Y

? WILT U TIJDSTAPPEN ZELF BEPALEN (Y/N)
U: N
? GEEF WAARDE VOOR DE LADINGSTOENAME PER STAP (C)
U: 2.5E-17
? GEEF WAARDE VOOR DE MAXIMALE TIJDSTAP (S)
U: 1E-2
?GEEF WAARDE VOOR HET EINDCRITERIUM (DELTA)
U: 0.0001
ET= 4.57.4 PT= 2.1 IO= 1.5

A6 VOORBEELD INTERF DIALOOG MET ELECTR/INTERFACE

(U=user)

U: get ELECTR/INTERFACE

U: E

#RUNNING 8404

? TITEL VAN FILE DATAMG

#?

U: VOORBEELD1

? TITEL VAN FILE MESH (DE GEGENEREERDE MESH)

U: TRIQUAMESH060187/GBOL/SAFERESULTS

? IDENTIFICATIETEGEL VOOR FEMSYS-UITVOER

* (WANNEER DEZE REGEL NIET LEEG IS DAN WORDEN DE EERSTE

* 6 KARAKTERS NIET MEER EN NIET MINDER) GEBRUIKT ALS

* DIRECTORY VOOR DOOR FEMSYS GECEEERDE FILES;

* GEBRUIK BIJ VOORKEUR EEN SPREKENDE TEXT)

U: GBOL1

? AANTAL KNOOPPUNTEN, AANTAL ELEMENTEN (BOVENSCHATTINGEN)

U: 900, 1500

? COORDINATEN VAN DE BASISPUNTEN (Y/N)

U: N

* STRUCTUUR 1

* 2 GROEP(EN)

* GROEP 1 : 1208 ELEMENTEN, 3 KNOPEN PER ELEMENT

* GROEP 2 : 270 ELEMENTEN, 3 KNOPEN PER ELEMENT

GEEF 1 ALS HET PROBLEEM PUUR CAPACITIEF IS

GEEF 2 ALS HET PROBLEEM PUUR RESISTIEF IS

U: 1

? PERMITTIIVITEIT(ϵ_s/ϵ_m) VOOR GROEP 1

U: 8.85E-12

? CHARGE DENSITY(ρ_s/m^3) VOOR GROEP 1

U: 0

? PERMITTIIVITEIT(ϵ_s/ϵ_m) VOOR GROEP 2

U: 8.85E-07

? CHARGE DENSITY(ρ_s/m^3) VOOR GROEP 2

U: 0

? WILT U DE COORDINATEN AANPASSEN (Y/N)

U: Y

VERMENIGVULDIGINGSFACTOR

U: 1E-6

? WILT U UITLEG OVER DE RANDEN

U: Y

* U WORDT WAARSCHIJNLIJK VERZOCHT WAARDEN

* IN TE TIKKEN VOOR GROEP, VAN, VIA, NAAR.

* 'VAN', 'VIA' EN 'NAAR' ZIJN BASISPUNTNUMMERS

* U GEEFT DAARMEE EEN DEELRAND AAN.

* IS GEEN TUSSEN PUNT BESCHIKBAAR OM DE RICHTING VAST TE

* LEGGEN DAN TREEDT DE VOLGENDE REGELING IN WERKING :

* MET 'VAN, VAN, NAAR' BEDOELEN WE HET KORTSTE TRAJECT

* VAN 'VAN' NAAR 'NAAR'

* MET 'VAN, NAAR, NAAR' HET LANGSTE TRAJECT.

* VOOR GROEP KUNT U BIJNA ALTIJD VOLSTAAN MET DE WAARDE

* 0 (NUL).
 * ALLEEN WANNEER EEN BASISPUNT OP GROND VAN ZIJN
 * COORDINATEN NIET EENDUIDIG GELOCALISEERD KAN WORDEN,
 * MOET HET NUMMER VAN DE GROEP, WAARTOE DE DEELRAND
 * BEHOORT, WORDEN INGEVOERD.
 * U KUNT EEN OVERZICHT VAN ALLE RANDKNOOPNUMMERS KRIJGEN.
 * SUCCES !
 ? WILT U OPEENVOLGENDE RANDPUNTNUMMERS ZIEN (Y/N)
 U: N
 ? RANDELEMENTEN
 U: N
 ? KOPPELEN
 U: Y
 ? GROEP1, GROEP2, VAN, VIA, NAAR
 U: 1,2,5,8,7
 ? KOPPELEN
 U: N
 ? ZWEVENDE ELEKTRODEN
 U: N
 ? VOORGESCHREVEN RANDSPANNINGEN
 U: Y
 GROEP, VAN, VIA, NAAR, WAARDE SPANNING
 U: 1,3,3,4,6.6
 ? MEER VOORGESCHREVEN SPANNINGEN
 U: N
 ? GEAARDE RANDEN
 U: Y
 ? GROEP, VAN, VIA, NAAR
 U: 1,1,1,2
 ? MOETEN DE RESULTATEN GELOT WORDEN (Y/N)
 U: Y
 ? VERANDERLIJKE LADINGSDICHTHEID AAN RAND
 U: Y
 ? WILT U UITLEG (Y/N)
 U: Y
 * U WORDT DADELIIK VERZOCHT WAARDEN IN TE TIKKEN VOOR:
 * GROEP1, GROEP2, VAN, VIA, NAAR.
 * VOOR GROEP1 DIEN U DE GROEP IN TE VULLEN, DIE DE
 * ELEMENTEN BEVAT WAARVOOR DE NORMAALCOMPONENT VAN HET
 * ELECTRISCHE VELD MOET WORDEN BEREKEND.
 * VOOR GROEP2 DIEN U DE GROEP IN TE VULLEN DIE DE
 * ELEMENTEN BEVAT WAAR DE LADING NAAR TOE STROOMT.
 ? GROEP1, GROEP2, VAN, VIA, NAAR
 U: 1,2,5,8,7
 ? VERANDERLIJKE LADINGSDICHTHEID AAN RAND
 U: N
 ? MAXIMAAL DOOR REPFEMSYS UIT TE VOEREN STAPPEN
 U: 20
 ? GEEF WAARDE VOOR DE IONENCONCENTRATIE 'NO' (m-3)
 U: 5E14
 ? GEEF WAARDE VOOR DE BEWEEGLIJKHEID 'B' (m2/Us)
 U: 2.1E-4
 ? WILT U VARIABELE TIJDSTAPPEN (Y/N)
 U: Y

? WILT U TIJDSTAPPEN ZELF BEPALEN (Y/N)
U: N
? GEEF WAARDE VOOR DE LADINGSTOENAME PER STAP (C)
U: 2.5E-17
? GEEF WAARDE VOOR DE MAXIMALE TIJDSTAP (s)
U: 1E-2
?GEEF WAARDE VOOR HET EINDCRITERIUM (DELTA)
U: 0.0001
* ET= 4.57.4 PT= 2.1 IO= 1.5

A7 VOORBEELD FILE INTERF

FILE INTERF:

```
100 gbol01 (900, 1500)
200 karakteristieken
300 8.85e-12, 0;
400 8.85e-11, 0;
500 ca (@-6)
600 koppelen
700 1,2,5,8,7;
800 randspanningen
900 1,3,3,4:6.6;
1000 geaarde randen
1100 1,1,1,2;
1200 plots
1300 veranderlijke ladingsdichtheid
1400 1,2,5,8,7;
1500 maxstap: 20;
1600 n0 : 5E14;
1700 b : 2.1E-4;
1800 dq : 2.5E-17;
1900 mts : 1E-2;
2000 delta : 0.0001;
```

AB VOORBEELD FILE OUTREPFEM

file OUTREPFEM:

permittiviteit groep 1 = 8.85E-12
 permittiviteit groep 2 = 8.85E-7
 NO = 5.00000E+14
 B = 2.10000E-04
 DQ = 2.50000E-17
 DELTA = 1.00000E-04
 MTS = 1.00000E-02 (MAXIMALE TIJDSTAP)
 MS = 2.00000E+01 (MAXIMAAL AANTAL STAPPEN)
 AS = 2.00000E+01 (AANTAL STAPPEN)

T [S]	I [A]	T [S]	I [A]
0.	1.38403E-13	1.80633E-04	1.15760E-13
3.96596E-04	9.48683E-14	6.60119E-04	7.57338E-14
9.90223E-04	5.84533E-14	1.41791E-03	4.31107E-14
1.99782E-03	2.97514E-14	2.83811E-03	1.86064E-14
4.18174E-03	9.89169E-15	6.70911E-03	3.60353E-15
1.36467E-02	3.54329E-16	2.36467E-02	1.67466E-16
3.36467E-02	1.03208E-16	4.36467E-02	7.13546E-17
5.36467E-02	5.26086E-17	6.36467E-02	4.06553E-17
7.36467E-02	3.24508E-17	8.36467E-02	2.66999E-17
9.36467E-02	2.23782E-17	1.03647E-01	1.91120E-17

T1 [S]	T2 [S]	DQ [C]	Q [C]
0.	1.80633E-04	2.50000E-17	2.50000E-17
1.80633E-04	3.96596E-04	2.50000E-17	5.00000E-17
3.96596E-04	6.60119E-04	2.50000E-17	7.50000E-17
6.60119E-04	9.90223E-04	2.50000E-17	1.00000E-16
9.90223E-04	1.41791E-03	2.50000E-17	1.25000E-16
1.41791E-03	1.99782E-03	2.50000E-17	1.50000E-16
1.99782E-03	2.83811E-03	2.50000E-17	1.75000E-16
2.83811E-03	4.18174E-03	2.50000E-17	2.00000E-16
4.18174E-03	6.70911E-03	2.50000E-17	2.25000E-16
6.70911E-03	1.36467E-02	2.50000E-17	2.50000E-16
1.36467E-02	2.36467E-02	3.54329E-18	2.53543E-16
2.36467E-02	3.36467E-02	1.67466E-18	2.55218E-16
3.36467E-02	4.36467E-02	1.03208E-18	2.56250E-16
4.36467E-02	5.36467E-02	7.13546E-19	2.56964E-16
5.36467E-02	6.36467E-02	5.26086E-19	2.57490E-16
6.36467E-02	7.36467E-02	4.06553E-19	2.57896E-16
7.36467E-02	8.36467E-02	3.24508E-19	2.58221E-16
8.36467E-02	9.36467E-02	2.66999E-19	2.58488E-16
9.36467E-02	1.03647E-01	2.23782E-19	2.58712E-16
1.03647E-01	1.13647E-01	1.91120E-19	2.58903E-16

AS VOORBEELD FILE OUTER

Hieronder volgt een gedeelte van een file OUTER:

GEBIED	TIJD [S]	Er [U/M]	TIJD [S]	q [C/M3]
1	0.	-2.68351E+05	1.92819E-04	1.05191E-19
2	0.	-2.67971E+05	1.92819E-04	3.14691E-19
3	0.	-2.65910E+05	1.92819E-04	5.18913E-19
4	0.	-2.65835E+05	1.92819E-04	7.23227E-19
5	0.	-2.67199E+05	1.92819E-04	9.29280E-19
6	0.	-2.63182E+05	1.92819E-04	1.11077E-18
7	0.	-2.51985E+05	1.92819E-04	1.24591E-18
8	0.	-2.41667E+05	1.92819E-04	1.36468E-18
9	0.	-2.32821E+05	1.92819E-04	1.47308E-18
10	0.	-2.21921E+05	1.92819E-04	1.54841E-18
11	0.	-2.09849E+05	1.92819E-04	1.59529E-18
12	0.	-1.97241E+05	1.92819E-04	1.61536E-18
13	0.	-1.84267E+05	1.92819E-04	1.61138E-18
14	0.	-1.71403E+05	1.92819E-04	1.58705E-18
15	0.	-1.59632E+05	1.92819E-04	1.55474E-18
16	0.	-1.44456E+05	1.92819E-04	1.46963E-18
17	0.	-1.25274E+05	1.92819E-04	1.32344E-18
18	0.	-1.09883E+05	1.92819E-04	1.19903E-18
19	0.	-9.19550E+04	1.92819E-04	1.03140E-18
20	0.	-7.79065E+04	1.92819E-04	8.92915E-19
21	0.	-6.31477E+04	1.92819E-04	7.36838E-19
22	0.	-4.68272E+04	1.92819E-04	5.53543E-19
23	0.	-3.19774E+04	1.92819E-04	3.81325E-19
24	0.	-9.51139E+03	1.92819E-04	1.13886E-19
25	0.	5.07359E+03	1.92819E-04	0.
26	0.	2.26644E+04	1.92819E-04	0.
27	0.	4.18708E+04	1.92819E-04	0.
28	0.	6.16248E+04	1.92819E-04	0.
29	0.	7.82625E+04	1.92819E-04	0.
30	0.	9.32836E+04	1.92819E-04	0.
31	0.	1.09631E+05	1.92819E-04	0.
32	0.	1.26390E+05	1.92819E-04	0.
33	0.	1.41465E+05	1.92819E-04	0.
34	0.	1.52251E+05	1.92819E-04	0.
35	0.	1.86186E+05	1.92819E-04	0.
36	0.	1.82797E+05	1.92819E-04	0.
37	0.	1.96202E+05	1.92819E-04	0.
38	0.	2.08966E+05	1.92819E-04	0.
39	0.	2.21372E+05	1.92819E-04	0.
40	0.	2.32448E+05	1.92819E-04	0.
41	0.	2.41456E+05	1.92819E-04	0.
42	0.	2.51397E+05	1.92819E-04	0.
43	0.	2.62882E+05	1.92819E-04	0.
44	0.	2.68221E+05	1.92819E-04	0.
45	0.	2.66452E+05	1.92819E-04	0.
46	0.	2.67244E+05	1.92819E-04	0.
47	0.	2.69149E+05	1.92819E-04	0.
48	0.	2.67822E+05	1.92819E-04	0.

A10 VOORBEELD INTERFACE DIALOOG MET USER/DRAW

Het programma USER/DRAW is voor de beginsituatie (eerste tijdstip, $t = 0$) van de geleidende bol gedraaid.
Interface-dialoog met user/draw:

E(RCLC4)USER/DRAW ON USER1

.....

DEVICE: T4014=1, T4010=2, TELEVIDEO=2, HP2647A=2, A/N=3 (2):1

PLOTS (0=NO, 1=YES, 2=PREVIEW) (0) :1

A-FORMAT (1-6) (4) :4

UPPERBOUND FOR NUMBER OF NODES (1000):900

UPPERBOUND FOR NUMBER OF ELEMENTS (1000) :1500

UPPERBOUND FOR NUMBER OF STRESSES PER NODE (7) :3

>ENTER NAME OF CONNECTIONFILE (.. COMPNODES)

GBOLO1/S/GROUP/A/COMPNODES

>ENTER NAME OF COORDINATEFILE (.. COORD)

GBOLO1/S/GROUP/A/COORD

>ENTER NAME OF DISPLACEMENTSFILE (.. LHV) (OR EMPTY LINE) :

>ENTER NAME OF STRESSFILE (.. NODEDER) (OR EMPTY LINE)

GBOLO1/000001/X1/S/GROUP/A/NODEDER

> ENTER IDENTIFICATION LINE (MAX 20 CHARS)

GELEIDENDEBOL

MAIN>S

:

O - OPTIONS

B - BOUNDARY

ME - MESH

C - CONTOURS

M - MAINSTRESSES

G - GRADIENTS

L - LINE

SET>C

INDEX (1) :1

STARTLEVEL (0.) :0

INCREMENT (1.) :.1

NUMBER OF INCREMENTS (5) :100

MAIN>D

```
:  
B - BOUNDARY  
ME - MESH  
C - CONTOURS  
M - MAINSTRESSES  
G - GRADIENTS  
L - LINE
```

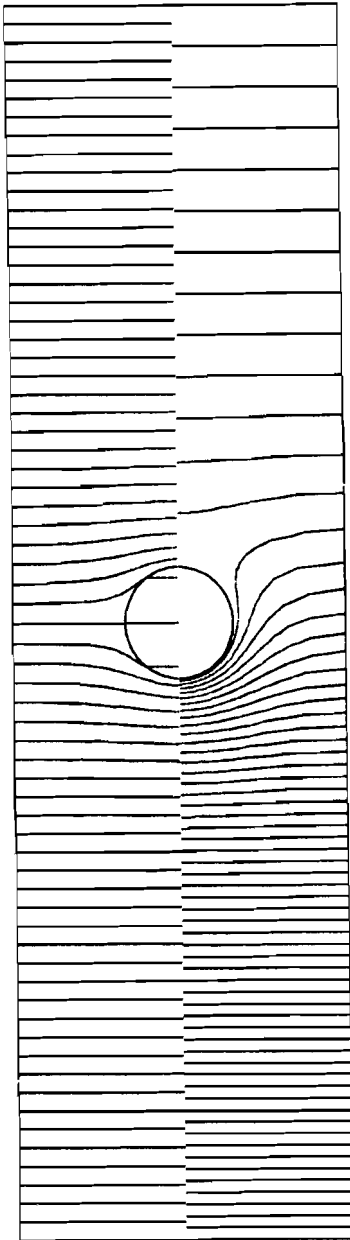
```
DRAW>C
```

Op het beeldscherm verschijnt nu de tekening van het gebied op tijdstip 1 (de beginsituatie) met equipotentiaallijnen.

```
MAIN > STOP  
#ET = 5:00 PT-2.1 IO-1.5
```

USER/DRAW is nog eens gedraaid voor de situatie op het twintigste tijdstip ($t = 0.106478E-01$ s).
De plotjes van deze twee uitdraaien zijn in figuur A10.1 gezet.

GELEIDENDE BOL



Figuur A10.1: linkerdeel: equipotentiaallijnen op tijdstip $t = 0$,
rechterdeel: equipotentiaallijnen op tijdstip $t = 0.106478$ s.

A11 FILE'S DATAMG EN INTERF VOOR GELEIDENDE BOL

Hieronder volgen de file's DATAMG en INTERF voor de geleidende bol met de kleinst mogelijke verdeling.

File DATAMG:

```
100$INPUT "GBOL"
200$BASISPOINTS
300 1: 0, 0; 2: 9, 0;
400 3: 9, 66; 4: 0, 66;
500 5: 0, 36; 6: 0, 33;
600 7: 0, 30; 8: 3, 33;
700 9: 0, 37.05; 10: 4.05, 33;
800 11: 0, 28.95; 12: 0, 35;
900 13: 2, 33; 14: 0, 31;
1000 15: 0, 36.05; 16: 3.05, 33;
1100 17: 0, 29.95; 18: 0, 35.95;
1200 19: 2.95, 33; 20: 0, 30.05;
1300$CONTOURPIECES
1400 1(4RL9CM-6, 10CM-6, 11RL1RL2RL3RL4)
1500 2(9RL15CM-6, 16CM-6, 17RL11CM6, 10CM6, 9)
1600 3(15RL5CM-6, 8CM-6, 7RL17CM6, 16CM6, 15)
1700 4(5RL18CM-6, 19CM-6, 20RL7CM6, 8CM6, 5)
1800 5(18RL12CM-6, 13CM-6, 14RL20CM6, 19CM6, 18)
1900 6(12RL6RL14CM6, 13CM6, 12)
2000$SUBCONTOURS
2100 1(1)
2200 2(2)
2300 3(3)
2400 4(4)
2500 5(5)
2600 6(6)
2700$SUBSTRUCTURES
2800 FEMSYS
2900 1TRIM3(1,2,3,4,5,6)
3000$GRADING
3100 G3 RI 1
3200 GV .05(5,7,8,15,16,17,18,19,20)
3300 GV 1(9,10,11)
3400 GV 1(6)
3500 GV 0.8(12,13,14)
3600$RENUMBER
3700 PROFILE(1LINE 1,2)
3800$STOP
```

FILE INTERF:

100 GBOL02 (5000, 5000)
200 KARAKTERISTIEKEN
300 8.85E-12, 0;
400 8.85E-12, 0;
500 8.85E-12, 0;
600 8.85E-07, 0;
700 8.85E-07, 0;
800 8.85E-07, 0;
900 CA (e-6)
1000 KOPPELEN
1100 1,2,9,10,11;
1200 2,3,15,16,17;
1300 3,4,5,8,7;
1400 4,5,18,19,20;
1500 5,6,12,13,14;
1600 RANDSPANNINGEN
1700 1,3,3,4:6.6;
1000 GEAARDE RANDEN
1100 1,1,1,2;
1300 VERANDERLIJKE LADINGSDICHTHEID
1400 3,4,5,8,7;
1500 MAXSTAP: 20;
1600 NO : 5E14;
1700 B : 2.1E-4;
1800 DQ : 2.5E-17;
1900 MTS : 1E-2;
2000 DELTA : 0.0001;


```

        IF TSK = 1 THEN HN:= 4
        ELSE IF TSK = 2 THEN HN:= 5
            ELSE HN:= MAXS + 3;
        IF READ (G [0], /, D1, D2, FOR I:= 1 TO1 HN DO D [I])
            THEN HERROR:= TRUE ELSE
        BEGIN
            FOR I:= 1 TO1 HN - 1 DO
                HERROR:= HERROR OR (D [I] <= 0);
            HERROR:= HERROR OR (D [HN] < 0) OR (D [HN] > 1);
            IF NOT READ (G [0], /, FOR I:= 1 TO1 HN + 3 DO D [I])
                THEN HERROR:= TRUE
        END
    END
END;
IF HERROR THEN WRITE (OUTREPFEM, <"FILE INREADREPFEM2 ",
    "IS NIET GOED">);
    ERROR:= ERROR OR HERROR
END;
IF NOT H.RESIDENT THEN
BEGIN
    WRITE (OUTREPFEM, <"FILE INREADFEM IS NIET AANWEZIG">);
    ERROR:= TRUE
END;
IF ERROR THEN
BEGIN
    LOCK (OUTREPFEM, CRUNCH);
    GO TO EINDE
END
END;
BEGIN
    DEFINE DQ = 1.6E-19 #;
    INTEGER POS;
    REAL DQ, B, NO, DELTA, MTS;
    ARRAY X, Y, U [0 : NNODES],
        CHAR [1 : NGROUPS, 0 : 1],
        RHO, ALFA, VOLUME, OPPERU [1 : AG],
        TLAD, STRM, TYD [0 : MAXS],
        RHODI [1 : ADB];
    INTEGER ARRAY DRH [1 : AG, 0 : 3],
        NDR [1 : ADB, 0 : 3],
        ADR [1 : AG + 1],
        EINDEL [1 : NGROUPS];
    EBCDIC ARRAY IDENT [0 : 5];
    % DQ = TOENAME LADINGSDICHTHEID PER STAP
    % B = "B"
    % NO = "NO"
    % DELTA = AFBREEKCONSTANTE
    % MTS = MAXIMALE TIJDSTAP
    % (AJ: 1 <= J <= NGOUPS:
    %     CHAR [J,0] = PERMITTIVITEIT GROEP J
    %     CHAR [J,1] = LADINGSDICHTHEID GROEP J )
    % POS = POSITIE IN CIRF WAAR LADINGSDICHTHEID MOET WORDEN
    %     AANGEBRACHT

```



```

% (AJ: 1 <= J <= NNODES:
%   XC[J] = X-COORDINAAT PUNT J
%   YC[J] = Y-COORDINAAT PUNT J)
% (AJ: 1 <= J <= AG :
%   DRH [J,0] = NUMMER DRIEHOEK GEBIED J AAN "BUITENKANT"
%   DRH [J,1], DRH [J,2] EN DRH [J,3] ZIJN DE KNOOPPUNTEN
%   VAN DE DRIEHOEK,
%   ALFA [I] = DE HOEK "ALFA" VAN GEBIED I
%   VOLUME [I] = VOLUME VAN GEBIED I
%   OPPERU [I] = OPPERVLAKTE VAN GEBIED I )
% (AJ: 1 <= J <= AG:
%   (AL: ADR [J] <= J < ADR [J+1]:
%     NDR [L,0] = NUMMER DRIEHOEK GEBIED J AAN "BINNENKANT"
%     NDR [L,1], NDR [L,2] EN NDR [L,3] ZIJN DE KNOOPPUNTEN
%     VAN DE DRIEHOEK ))
%   (AJ: 1 <= J <= ADB:
%     RHODI[J] = BEGINLADINGSDICHTHEID DRIEHOEK NDRC[J,0])
%   (AJ: 1 <= J <= NGROUPS:
%     EINDEL [I] = "HOOGSTE DRIEHOEKSNUMMER GROEP J + 1" )
%   IDENT = IDENTIFICATIETREGEL

```

```

PROCEDURE INREAD;

```

```

  % INLEZEN GEGEVENS UIT INREADREPFEM1

```

```

BEGIN

```

```

  FILE F (KIND = DISK, FILETYPE = 7, TITLE= "INREADREPFEM1.");
  INTEGER I, J;
  BOOLEAN STOP;
  INTEGER ARRAY H [0 : 3];

```

```

  READ (F); READ (F); READ (F);

```

```

  ADR [1] := 1;

```

```

  FOR I := 1 TO 1 AG DO

```

```

  BEGIN

```

```

    READ (F, <4(I6, X1)>, FOR J := 0 TO 3 DO DRH [I, J]);

```

```

    STOP := FALSE;

```

```

    ADR [I + 1] := ADR [I];

```

```

    WHILE NOT STOP DO

```

```

    BEGIN

```

```

      READ (F, <4(I6, X1)>, FOR J := 0 TO 3 DO H [J]);

```

```

      IF H [0] = 0 THEN STOP := TRUE

```

```

      ELSE BEGIN

```

```

        FOR J := 0 TO 3 DO NDR [ADR [I + 1], J] := H [J];

```

```

        ADR [I + 1] := ADR [I + 1] + 1

```

```

      END

```

```

    END

```

```

  END

```

```

END INREAD;

```

```

PROCEDURE COPYIRF (PLOTS);

```

```

  % INLEZEN GEGEVENS INREADFEM

```

```

  % COPY VAN INREADFEM MAKEN

```

```

  BOOLEAN PLOTS;

```

```

BEGIN

```

```

  FILE F (KIND = DISK, FILETYPE = 7, TITLE = "INREADFEM."),

```

```

  G (KIND = DISK, MAXRECSIZE = 14, BLOCKSIZE = 420,

```

```

        AREASIZE = 600, AREAS = 25, TITLE = "CIRF.");
DEFINE COPYL =
BEGIN
    PROW:= POINTER (ROW);
    READ (F, <A84>, PROW);
    WRITE (G, <A84>, PROW);
END COPYL #;
INTEGER I, N1, N2, H, T, ND;
BOOLEAN STOP;
ARRAY ROW [0 : 18];
POINTER PROW;

REPLACE POINTER (ROW [14]) BY "*";
COPYL;
IF PROW = "      " THEN REPLACE IDENT BY "FEMSYS" ELSE
BEGIN
    REPLACE IDENT BY "0" FOR 6;
    I:=0;
    WHILE PROW NEQ " " AND I < 6 DO
    BEGIN
        REPLACE IDENT [I] BY PROW: PROW FOR 1;
        I:= I + 1
    END
END;
STOP:= FALSE;
WHILE NOT STOP DO
BEGIN
    COPYL;
    SCAN PROW: PROW UNTIL NEQ " ";
    IF PROW = "CHARACTERISTICS" THEN STOP:= TRUE
END;
COPYL;
READ (PROW, <X18, E12.5, ",>", CHAR [1,0]);
COPYL;
READ (PROW, <X18, E12.5, ";>", CHAR [1,1]);
PROW:= POINTER (ROW);
READ (F[NO], <A84>, PROW);
SCAN PROW: PROW UNTIL NEQ " ";
IF PROW = "EXCEPT" THEN COPYL;
FOR I:= 2 TO1 NGROUPS DO
BEGIN
    COPYL;
    READ (PROW, <I10, "(1)", I4, ":", E12.5, ",>",
        EINDEL [I - 1], H, CHAR [I, 0]);
    COPYL;
    READ (PROW, <X18, E12.5, ";>", CHAR [I, 1])
END;
EINDEL [NGROUPS]:= NEL + 1;
POS:= G.RECORD + 1;
PROW:= POINTER (ROW);
READ (F [NO], <A84>, PROW);
SCAN PROW: PROW UNTIL NEQ " ";
IF PROW = "COORDINATES" THEN
BEGIN
    IF NGROUPS = 1 THEN

```

```

BEGIN
  WRITE (G, <" EXCEPT">);
  POS:= POS + 1
END;
FOR I:= 1 TO1 (2 * ADB) DO WRITE (G, <" ">);
FOR I:= 1 TO1 ADB DO
BEGIN
  T:=1;
  WHILE (NDR [I, 0] >= EINDEL [T]) DO T:= T + 1;
  RHODI [I]:= CHAR [T, 1]
END
END ELSE
BEGIN
  FOR I:= 1 TO1 ADB DO
  BEGIN
    COPYL;
    COPYL;
    READ (PROW, <X18, E12.5, ";">, RHODI [I])
  END
END;
COPYL;
DO BEGIN
  COPYL;
  READ (PROW, <2(I5, ":", E12.5, ":", E12.5, ":", ")>,
        N1, X [N1], Y [N1], N2, X [N2], Y [N2])
  END
  UNTIL (N1 = NNODES) OR (N2 = NNODES);
PLOTS:= FALSE;
PROW:= POINTER (ROW);
WHILE NOT READ (F, <A84>, PROW) DO
BEGIN
  SCAN PROW: PROW UNTIL NEQ " ";
  IF PROW = "OPTION" THEN PLOTS:= TRUE;
  PROW:= POINTER (ROW);
  WRITE (G, <A84>, PROW)
END;
LOCK (G, CRUNCH)
END COPYIRF;

PROCEDURE CALCULATE;
BEGIN
  INTEGER I, J, JJ;
  REAL U;
  ARRAY XC, YC [0 :2];

  REAL PROCEDURE HOEK (XCOORD, YCOORD);
  ARRAY XCOORD, YCOORD [*];
  BEGIN
    REAL MU, XA, XB, XC, XD, YA, YB, YC, YD,
          DY, DX, PI, H1, H2;

    PI:= 4 * ARCIAN (1);
    XA:= XCOORD [0]; XB:= XCOORD [1]; XC:= XCOORD [2];
    YA:= YCOORD [0]; YB:= YCOORD [1]; YC:= YCOORD [2];
    H1:= ((XC - XA) * (XB - XA) + (YC - YA) * (YB - YA));

```

```

H2:= ((XA - XB) * (XA - XB) + (YA - YB) * (YA - YB));
MU:= H1/H2;
XD:= XA + MU * (XB - XA); YD:= YA + MU * (YB - YA);
DY:= YC - YD; DX:= XC - XD;
IF DX = 0 THEN IF DY > 0 THEN HOEK:= PI/2
                ELSE HOEK:= -PI/2
ELSE IF DX > 0 THEN HOEK:= ARCTAN (DY/DX)
                ELSE HOEK:= PI + ARCTAN (DY/DX)
END HOEK;

REAL PROCEDURE OPP (XCOORD, YCOORD);
  REAL ARRAY XCOORD, YCOORD [*];
BEGIN
  REAL PI, X1, X2, Y1, Y2, DYDX, H;
  PI:= 4 * ARCTAN (1);
  X1:= XCOORD [0]; X2:= XCOORD [1];
  Y1:= YCOORD [0]; Y2:= YCOORD [1];
  IF X1 = X2 THEN OPP:= ABS (2 * PI * X1 * (Y2 - Y1))
  ELSE BEGIN
    DYDX:= (Y1 - Y2)/(X1 - X2);
    H:= SQRT (1 + DYDX * DYDX);
    OPP:= ABS (PI * H * (X2 * X2 - X1 * X1))
  END
END OPP;

REAL PROCEDURE VOL (XCOORD, YCOORD);
  REAL ARRAY XCOORD, YCOORD [*];
BEGIN
  INTEGER I;
  REAL U;

  REAL PROCEDURE HVOL (X1, X2, Y1, Y2);
    VALUE X1, X2, Y1, Y2;
    REAL X1, X2, Y1, Y2;
  BEGIN
    REAL DYDX, PI;
    PI:= 4 * ARCTAN (1);
    IF Y1 = Y2 THEN HVOL:= 0 ELSE
    BEGIN
      IF X1 = X2 THEN HVOL:= PI * X1 * X1 * (Y2 - Y1)
      ELSE BEGIN
        DYDX:= (Y1 - Y2)/(X1 - X2);
        HVOL:= PI * DYDX * (X2 ** 3 - X1 ** 3)/3
      END
    END
  END HVOL;

  U:= 0;
  FOR I:= 0 STEP 1 UNTIL 2 DO
    U:= U + HVOL (XCOORD [I], XCOORD [(I + 1) MOD 3],
                  YCOORD [I], YCOORD [(I + 1) MOD 3]);
  VOL:= ABS (U)
END VOL;

FOR I:= 1 TO1 AG DO

```

```

BEGIN
  FOR J:= 0 TO 1 2 DO
    BEGIN
      XC [J]:= X [DRH [I, J + 1]];
      YC [J]:= Y [DRH [I, J + 1]]
    END;
  ALFA [I]:= HOEK (XC, YC);
  OPPERU [I]:= OPP (XC, YC);
  J:= ADR [I]; U:= 0;
  WHILE J < ADR [I + 1] DO
    BEGIN
      FOR JJ:= 0 TO 1 2 DO
        BEGIN
          XC [JJ]:= X [NDR [J, JJ + 1]];
          YC [JJ]:= Y [NDR [J, JJ + 1]]
        END;
      U:= U + VOL (XC, YC);
      J:= J + 1
    END;
  VOLUME [I]:= U
END
END CALCULATE;

PROCEDURE READPOT (NS);
  VALUE NS;
  INTEGER NS;
BEGIN
  FILE F (KIND = DISK, FILETYPE = 7, TITLE = "OUTO.");
  FILE OUTREPFEM (KIND = DISK);
  INTEGER I, N1, N2, N3;
  BOOLEAN ERROR;
  ARRAY ROW [0 : 18];
  POINTER PROW;

  ERROR:= FALSE;
  IF NOT F.RESIDENT THEN
    BEGIN
      WRITE (OUTREPFEM, <"FILE OUTO IS NA BEREKENING", 16,
        " NIET AANGEMAAKT">, NS + 1);
      ERROR:= TRUE
    END ELSE
    BEGIN
      FOR I:= 1 TO 1 5 DO IF READ (F) THEN ERROR:= TRUE;
      REPLACE POINTER (ROW [14]) BY "**";
      PROW:= POINTER (ROW);
      IF READ (F, <A72>, PROW) THEN ERROR:= TRUE;
      IF NOT ERROR THEN
        BEGIN
          SCAN PROW: PROW UNTIL NEQ " ";
          IF PROW NEQ "NODE" THEN ERROR:= TRUE;
          IF READ (F) THEN ERROR:= TRUE
        END;
      IF NOT ERROR THEN
        DO READ (F, <3(I5, X4, E13.6, X4)>,
          N1, U [N1], N2, U [N2], N3, U [N3])
    END

```

```

        UNTIL N1 = 0
        ELSE WRITE (OUTREPFEM, <"IN OUTO STAAT EEN FOUTMELDING">)
    END;
    IF ERROR THEN
    BEGIN
        LOCK (OUTREPFEM, CRUNCH);
        GO TO EINDE
    END
END READPOT;

PROCEDURE ADJUSTVAR (NS, CV);
    VALUE NS;
    INTEGER NS;
    BOOLEAN CV;
BEGIN
    INTEGER C;
    REAL X1, X2, X3, Y1, Y2, Y3, U1, U2, U3,
        EX, EY, DT, DRHO;
    ARRAY ER, I [1 : AG];

    SIRM [NS] := 0; TLAD [NS + 1] := TLAD [NS];
    FOR C := 1 TO 1 AG DO
    BEGIN
        X1 := X [DRH [C, 1]]; Y1 := Y [DRH [C, 1]];
        U1 := U [DRH [C, 1]];
        X2 := X [DRH [C, 2]]; Y2 := Y [DRH [C, 2]];
        U2 := U [DRH [C, 2]];
        X3 := X [DRH [C, 3]]; Y3 := Y [DRH [C, 3]];
        U3 := U [DRH [C, 3]];
        EX := -((U1 - U2) * (Y1 - Y3) - (U1 - U3) * (Y1 - Y2)) /
            ((X1 - X2) * (Y1 - Y3) - (X1 - X3) * (Y1 - Y2));
        EY := -((U1 - U2) * (X1 - X3) - (U1 - U3) * (X1 - X2)) /
            ((Y1 - Y2) * (X1 - X3) - (Y1 - Y3) * (X1 - X2));
        ER [C] := EX * COS (ALFA [C]) + EY * SIN (ALFA [C]);
        IF ER [C] < 0 THEN
            I [C] := -1 * NO * DQ * B * ER [C] * OPPERU [C]
        ELSE
            I [C] := 0;
        SIRM [NS] := SIRM [NS] + I [C]
    END;
    IF ISK = 2 THEN
    BEGIN
        IF SIRM [NS] NEQ 0 THEN DT := MIN (MTS, DQ / SIRM [NS])
        ELSE DT := MTS;
        TYD [NS + 1] := TYD [NS] + DT;
    END ELSE DT := TYD [NS + 1] - TYD [NS];
    CV := FALSE;
    FOR C := 1 TO 1 AG DO
    BEGIN
        DRHO := I [C] * DT / VOLUME [C];
        TLAD [NS + 1] := TLAD [NS + 1] + I [C] * DT;
        IF DRHO > ((50-8) * RHO [C]) THEN CV := TRUE;
        RHO [C] := RHO [C] + DRHO;
        WRITE (OUTER, <I6, " !", 2(E12.5, " ") "!", 2(E12.5, " ")>,
            C, TYD [NS], ER [C], TYD [NS+1],

```

RHO [C] * VOLUME [C])

END
END ADJUSTUAR;

PROCEDURE ADJUSTINR;
BEGIN

```
FILE F (KIND = DISK, FILETYPE = 7, MYUSE = IO,  
        TITLE = "CIRF.");  
INTEGER I, J, T, ND;  
ARRAY RHOD [1 : NEL];  
FOR I:= 1 TO1 NEL DO RHOD [I]:= 0;  
FOR I:= 1 TO1 AG DO  
BEGIN  
  J:= ADR [I];  
  WHILE J < ADR [I + 1] DO  
  BEGIN  
    RHOD [NDR [J, 0]]:= RHOD [NDR [J, 0]] + RHO [I];  
    J:= J + 1  
  END  
END;  
FOR I:= 0 TO1 ADB - 1 DO  
BEGIN  
  ND:= NDR [I + 1, 0];  
  T:= 1;  
  WHILE (ND > EINDEL [T]) DO T:= T + 1;  
  WRITE (F [POS + 2 * I], <I17, ":", E12.5, ", ">,  
        ND, CHAR [T, 0]);  
  WRITE (F [POS + 2 * I + 1], <X18, E12.5, "; ">,  
        RHODI [I + 1] + RHOD [ND])
```

END
END ADJUSTINR;

PROCEDURE INITUAR;
BEGIN

```
INTEGER I;  
FOR I:= 1 TO1 AG DO RHO [I]:= 0;  
TLAD [0]:= 0;  
WRITE (OUTER, <"GEBIED !", X2, "TIJD [S]" X5, "Er [U/M]",  
        X3, "!", X2, "TIJD [S]", X6, "q [C/M3]">);  
WRITE (OUTER, <60("-")>)
```

END INITUAR;

PROCEDURE READCON;

% INLEZEN GEGEVENS UIT INREADREPFEM2

```
BEGIN  
FILE INC (KIND = DISK, FILETYPE = 7, TITLE= "INREADREPFEM2.");  
INTEGER I, D1, D2;  
REAL DT;  
ARRAY DIA [1 : MAXS];  
  
TYD [0]:= 0;  
IF TSK = 1 THEN  
BEGIN  
  READ (INC, /, D1, D2, NO, B, DT, DELTA);  
  FOR I:= 1 TO1 MAXS DO TYD [I]:= TYD [I - 1] + DT
```

```

END ELSE
IF ISK = 2 THEN
BEGIN
  READ (INC, /, D1, D2, NO, B, DQ, MTS, DELTA)
END ELSE
IF ISK = 3 THEN
BEGIN
  READ (INC, /, D1, D2, NO, B,
        FOR I:= 1 TO1 MAXS DO DTA [I], DELTA);
  FOR I:= 1 TO1 MAXS DO
    TYD [I]:= TYD [I - 1] + DTA [I]
  END
END READCON;

BOOLEAN PROCEDURE DOFEM (NS, CV);
  VALUE NS, CV;
  INTEGER NS;
  BOOLEAN CV;

BEGIN
  BOOLEAN EC;
  EC:= FALSE;
  IF NS > 0 THEN EC:= (STRM [NS-1] < (DELTA * STRM [0]));
  DOFEM:= (NS < MAXS) AND (NOT EC) AND CV
END DOFEM;

PROCEDURE UITVOER (NS, CV);
  VALUE NS, CV;
  INTEGER NS;
  BOOLEAN CV;
BEGIN
  FILE OUT (KIND = DISK, MAXRECSIZE = 14, BLOCKSIZE = 420,
            AREASIZE = 600, AREAS = 25, TITLE = "OUTREPFEM.");
  INTEGER I;
  FOR I:= 1 TO1 NGROUPS DO
    WRITE (OUT, <"PERMITTIIVITEIT VOOR GROEP ", I*, " : ",
           E12.5>, ENTIER (LOG (I)) + 1, I, CHAR [I, 0]);
  WRITE (OUT, <"MS = ", E12.5, " (MAXIMAAL AANTAL STAPPEN)">,
        MAXS);
  WRITE (OUT, <"AS = ", E12.5, " (AANTAL STAPPEN)">, NS);
  WRITE (OUT, <"NO = ", E12.5>, NO);
  WRITE (OUT, <" B = ", E12.5>, B);
  IF ISK = 1 THEN
  BEGIN
    IF MAXS > 0 THEN WRITE (OUT, <"DT = ", E12.5>, TYD [1])
  END ELSE
  IF ISK = 2 THEN
  BEGIN
    WRITE (OUT, <"DQ = ", E12.5>, DQ);
    WRITE (OUT, <"MTS = ", E12.5, " (MAXIMALE TIJDSTAP)">, MTS)
  END ELSE
  BEGIN
    FOR I:= 1 TO1 NS DO
      WRITE (OUT, <"DT", I*, " = ", E12.5>,
            ENTIER (LOG (I)) + 1, I, TYD [I] - TYD [I - 1])
    END
  END

```



```

END;
WRITE (OUT, <"DELTA = ", E12.5>, DELTA);
WRITE (OUT, <" ">);
IF NS > 0 THEN
BEGIN
  IF STRM [NS - 1] < (DELTA * STRM [0]) THEN
    WRITE (OUT, <"DE STROOMSTERKTE IS KLEINER DAN ",
      " DELTA MAAL DE BEGINSTROOM">);
  ELSE IF NOT CV THEN
    WRITE (OUT, <"DE INVOERFILE VOOR FEMSYS IS ",
      "T.O.U. DE VORIGE STAP ONGEWIJZIGD">);
END;
WRITE (OUT, <" ">); WRITE (OUT, <" ">);
WRITE (OUT, <X3, 2("!", X5, "T [S]", X5, "!", X5, "I [A]",
  X5), "! ">);
WRITE (OUT, <X3, 65("-")>);
I:= 0;
WHILE I < (NS - 1) DO
BEGIN
  WRITE (OUT, <X3, "!", 4(X1, E12.5, X2, "! ">, TYD [I],
    STRM [I], TYD [I + 1], STRM [I + 1]);
  I:= I + 2
END;
IF I = NS - 1 THEN
  WRITE (OUT, <X3, "!", 2(X1, E12.5, X2, "!"),
    2(X15, "! ">, TYD [I], STRM [I]);
WRITE (OUT, <X3, 65("-")>);
WRITE (OUT, <" ">); WRITE (OUT, <" ">);
WRITE (OUT, <X3, "!", X4, "T1 [S]", X5, "!", X4, "T2 [S]",
  X5, "!", X4, "DQ [C]", X5, "!", X5, "Q [C]",
  X5, "! ">);
WRITE (OUT, <X3, 65("-")>);
FOR I:= 0 TO NS - 1 DO
BEGIN
  WRITE (OUT, <X3, "!", 4(X1, E12.5, X2, "! ">, TYD [I],
    TYD [I + 1], TLAD [I+1] - TLAD[I], TLAD[I+1]);
END;
WRITE (OUT, <X3, 65("-")>);
LOCK (OUT, CRUNCH)
END UITVOER;

PROCEDURE CHFP (N);
  VALUE N;
  INTEGER N;

BEGIN
  EBCDIC ARRAY ONAAM, NNAAM [0 : 40];

  REPLACE ONAAM BY IDENT FOR 6;
  REPLACE ONAAM [6] BY "/X1/S/GROUP/A/NODEDER.";
  REPLACE NNAAM BY IDENT FOR 6;
  REPLACE NNAAM [6] BY "/";
  REPLACE NNAAM [7] BY (N+1) FOR 6 DIGITS;
  REPLACE NNAAM [13] BY "/X1/S/GROUP/A/NODEDER.";
  CHANGEFILE (ONAAM , NNAAM)

```

```

END CHFP;

BEGIN
  INTEGER NSTEPS;
  BOOLEAN CHVAR, PLOTS;
  % NSTEPS = AANTAL GEMAAKTE STAPPEN
  % CHVAR = VERANDERING IN DE FILE CIRF ?
  % PLOTS = MOETEN ER PLOTS GEMAAKT WORDEN ?

  TASK I;
  PROCEDURE FEMSYS; EXTERNAL;
  REPLACE I.NAME BY "OBJECT/ELECTR/ROTELEMENT.";
  REPLACE I.FILECARDS BY "FILE INREADFEM (KIND= DISK,
                          TITLE= CIRF)";
  REPLACE I.FILECARDS BY "FILE OUTREADFEM (KIND= DISK)";
  REPLACE I.FILECARDS BY "FILE OUTO (KIND= DISK)" 48"00";

  READCON;
  NSTEPS:= 0; CHVAR:= TRUE;
  IF DOFEM (NSTEPS, CHVAR) THEN
  BEGIN
    INREAD;
    COPYIRF (PLOTS);
    CALCULATE;
    INITVAR;
    WHILE DOFEM (NSTEPS, CHVAR) DO
    BEGIN
      REMOVEFILE ("OUTO.");
      CALL FEMSYS [I];
      READPOT (NSTEPS);
      IF PLOTS THEN CHFP (NSTEPS);
      UITVOER (NSTEPS, CHVAR);
      ADJUSTVAR (NSTEPS, CHVAR);
      ADJUSTINR;
      NSTEPS:= NSTEPS + 1
    END
  END;
  UITVOER (NSTEPS, CHVAR)
END
END;
EINDE: LOCK (OUTER, CRUNCH)
END.

```

A13 PROGRAMMATEKST ELECTR/INTERFACE/NI

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% M.b.v. dit programma kan de gebruiker aan een door
% TRIQUAMESH gecreëerde file die gegevens toevoegen,
% waardoor een geschikte invoerfile voor FEMSYS gemaakt
% kan worden. Het programma is geheel in termen van
% elektrische veldproblemen gesteld. Het is alleen geschikt
% voor puur capacatieve of puur resistieve veldproblemen.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
$SET LINEINFO
$SET INSTALLATION
BEGIN
  FILE MESH (KIND = DISK, FILETYPE = 7),
  DATAMG (KIND = DISK, FILETYPE = 7),
  F (KIND = DISK, MAXRECSIZE = 14, BLOCKSIZE = 420,
    AREASIZE = 600, AREAS = 25, TITLE = "INREADFEM."),
  INTERF (KIND = DISK, FILETYPE = 7),
  UTU (KIND = DISK, TITLE = "INTERF/OUT.");

$ INCLUDE "(RCLC4)INTERFACE ON USER1"

DEFINE DEELRAND (DR) -
BEGIN
  FOR I:- 1 TO1 3 DO
    INN [I]:- FIND (RAND, X, Y, PRES, RANDBP [0],
      XBP [RANDBP [I]], YBP [RANDBP [I]], EPS);
    PARTOFBOUND (DR, OK, RAND, INN [1], INN [2], INN [3]);
    IF NOT OK THEN WARN (1);
  END DEELRAND #;

DEFINE T01 = STEP 1 UNTIL #;
DEFINE NORW = 19 #;
INTEGER NNODES, NEL, PC, NUMI;
REAL NUMR;
BOOLEAN EOFI;
EBCDIC ARRAY LINE [0 : 72],
  NAAM [0 : 71],
  SYM [0 : 20],
  WOORD [1 : NORW, 1 : 20];

POINTER P;
TRUTHSET
  LETTERS ("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"),
  NUMMERS ("0123456789"),
  TEKEN ("+-"),
  REALT (".Ee@"),
  EXPT ("Ee@"),
  REST ("(),;:");
LABEL STOP;

```

```

PROCEDURE NEXTSYM;
BEGIN
  INTEGER I, J, K, UNSINT1, UNSINT2, NUM1, NUM2;
  REAL NUM3;
  BOOLEAN APUNT;
  EBCDIC ARRAY A [0 : 71];

  PROCEDURE NEXTCH;
  BEGIN
    IF PC = 73 THEN
      BEGIN
        P := LINE;
        IF NOT READ (INTERF, <A72>, P) THEN
          BEGIN
            PC := 1;
            WRITE (UTV, <A72>, P)
          END
        ELSE
          BEGIN
            REPLACE P BY "*";
            EOFI := TRUE
          END
        END
      END
    ELSE
      BEGIN
        PC := PC + 1;
        P := P + 1
      END
    END NEXTCH;

  REPLACE SYM BY " " FOR 20;
  WHILE P = " " DO NEXTCH;
  IF P IN LETTERS THEN
    BEGIN
      K := 0;
      REPLACE NAAM BY " " FOR 72;
      WHILE P IN LETTERS DO
        BEGIN
          IF P = "a" THEN REPLACE P BY "A" ELSE
          IF P = "b" THEN REPLACE P BY "B" ELSE
          IF P = "c" THEN REPLACE P BY "C" ELSE
          IF P = "d" THEN REPLACE P BY "D" ELSE
          IF P = "e" THEN REPLACE P BY "E" ELSE
          IF P = "f" THEN REPLACE P BY "F" ELSE
          IF P = "g" THEN REPLACE P BY "G" ELSE
          IF P = "h" THEN REPLACE P BY "H" ELSE
          IF P = "i" THEN REPLACE P BY "I" ELSE
          IF P = "j" THEN REPLACE P BY "J" ELSE
          IF P = "k" THEN REPLACE P BY "K" ELSE
          IF P = "l" THEN REPLACE P BY "L" ELSE
          IF P = "m" THEN REPLACE P BY "M" ELSE
          IF P = "n" THEN REPLACE P BY "N" ELSE
          IF P = "o" THEN REPLACE P BY "O" ELSE
          IF P = "p" THEN REPLACE P BY "P" ELSE
          IF P = "q" THEN REPLACE P BY "Q" ELSE

```

```

IF P = "r" THEN REPLACE P BY "R" ELSE
IF P = "s" THEN REPLACE P BY "S" ELSE
IF P = "t" THEN REPLACE P BY "T" ELSE
IF P = "u" THEN REPLACE P BY "U" ELSE
IF P = "v" THEN REPLACE P BY "V" ELSE
IF P = "w" THEN REPLACE P BY "W" ELSE
IF P = "x" THEN REPLACE P BY "X" ELSE
IF P = "y" THEN REPLACE P BY "Y" ELSE
IF P = "z" THEN REPLACE P BY "Z";
REPLACE NAAM [K] BY P FOR 1;
K:= K + 1;
NEXTCH
END;
WHILE P IN NUMMERS DO
BEGIN
REPLACE NAAM [K] BY P FOR 1;
K:= K + 1;
NEXTCH
END;
I:= 1; J:= NORW;
DO
BEGIN
K:= (I + J) DIV 2;
IF NAAM LEQ WOORD [K, 1] FOR 20 THEN J:= K - 1;
IF NAAM GEQ WOORD [K, 1] FOR 20 THEN I:= K + 1
END UNTIL I > J;
IF (I - 1) > J THEN REPLACE SYM BY WOORD [K, 1] FOR 20
ELSE REPLACE SYM BY "NAAM"
END ELSE
IF (P IN NUMMERS) OR (P IN TEKEN) OR (P IN REALT) THEN
BEGIN
UNSINT1:= (IF P = "-" THEN -1 ELSE 1);
IF P IN TEKEN THEN NEXTCH;
K:= 0;
WHILE P IN NUMMERS DO
BEGIN
REPLACE A [K] BY P FOR 1;
K:= K + 1;
NEXTCH
END;
IF (K = 0) AND (NOT P IN REALT) THEN REPLACE SYM BY "NUL"
ELSE
BEGIN
IF K = 0 AND (P IN EXPT) THEN NUM1:= UNSINT1
ELSE NUM1:= UNSINT1 * INTEGER (A [0], K);
IF K = 0 AND P = "." THEN APUNT:= TRUE ELSE APUNT:= FALSE;
IF NOT P IN REALT THEN
BEGIN
REPLACE SYM BY "INTEGER";
NUM1:= NUM1
END
ELSE
BEGIN
IF P IN EXPT THEN
BEGIN

```

```

NEXTCH;
UNSINT2:= (IF P = "-" THEN -1 ELSE 1);
IF P IN TEKEN THEN NEXTCH;
K:= 0;
WHILE P IN NUMMERS DO
BEGIN
    REPLACE A [K] BY P FOR 1;
    K:= K + 1;
    NEXTCH
END;
IF (K = 0) OR (K > 2) THEN REPLACE SYM BY "NUL"
ELSE
BEGIN
    NUM2:= UNSINT2 * INTEGER (A [0], K);
    NUMR:= NUM1 * (10 ** NUM2);
    REPLACE SYM BY "REAL"
END
END
ELSE
BEGIN
    NEXTCH;
    K:= 0;
    WHILE P IN NUMMERS DO
    BEGIN
        REPLACE A [K] BY P FOR 1;
        K:= K + 1;
        NEXTCH
    END;
    IF (K = 0) AND APUNT THEN REPLACE SYM BY "NUL"
    ELSE
    BEGIN
        NUM3:= UNSINT1 * INTEGER (A [0], K) * (10 ** (-K))
        IF (P = "e") OR (P = "E") OR (P = "@") THEN
        BEGIN
            NEXTCH;
            UNSINT2:= (IF P = "-" THEN -1 ELSE 1);
            IF P IN TEKEN THEN NEXTCH;
            K:= 0;
            WHILE P IN NUMMERS DO
            BEGIN
                REPLACE A [K] BY P FOR 1;
                K:= K + 1;
                NEXTCH
            END;
            IF (K = 0) OR (K > 2) THEN REPLACE SYM BY "NUL"
            ELSE
            BEGIN
                NUM2:= UNSINT2 * INTEGER (A [0], K);
                NUMR:= (NUM1 + NUM3) * (10 ** NUM2);
                REPLACE SYM BY "REAL"
            END
        END
    END
    ELSE
    BEGIN
        REPLACE SYM BY "REAL";

```

```

                                NUMR := NUM1 + NUM3
                                END
                                END
                                END
                                END
                                END ELSE
                                IF P IN REST THEN
                                BEGIN
                                REPLACE SYM BY P FOR 1;
                                NEXTCH
                                END ELSE
                                IF P = "*" AND EOFI THEN REPLACE SYM BY "END"
                                ELSE
                                BEGIN
                                REPLACE SYM BY "NUL";
                                NEXTCH
                                END
                                END NEXTSYM;

PROCEDURE ERROR (A);
    VALUE A;
    INTEGER A;
BEGIN
    WRITE (UTU, <*( " "), "*">, PC - 2);
    CASE A OF
    BEGIN
    1: WRITE (UTU, <" ERROR: INTEGER VERWACHT">);
    2: WRITE (UTU, <" ERROR: REAL VERWACHT">);
    3: WRITE (UTU, <" ERROR: NAAM VERWACHT">);
    4: WRITE (UTU, <" ERROR: STRUCTUUR NIET CORRECT GELEZEN">);
    5: WRITE (UTU, <" ERROR: ELEMENTEN KUNNEN NIET IN EEN GROEP">);
    6: WRITE (UTU, <" ERROR: GEEN CORRECTE RAND GEVONDEN">);
    7: WRITE (UTU, <" ERROR: KARAKTERISTIEKEN VERWACHT">);
    8: WRITE (UTU, <" ERROR: ', ' OF '; ' VERWACHT">);
    9: WRITE (UTU, <" ERROR: VOLGORDE VAN DE VOORGESCHREVEN ",
        "WOORDEN IS NIET GOED OF", /, X7, " VOORGESCHREVEN
        "WOORD KOMT AL VOOR IN DE INVOERFILE">);
    10: WRITE (UTU, <" ERROR: EINDE VAN DE INVOERFILE OF ", /,
        "
        VOORGESCHREVEN WOORD VERWACHT">);
    11: WRITE (UTU, <" ERROR: 'NO' VERWACHT">);
    12: WRITE (UTU, <" ERROR: 'B' VERWACHT">);
    13: WRITE (UTU, <" ERROR: 'MAXSTAP' VERWACHT">);
    14: WRITE (UTU, <" ERROR: 'MIS' VERWACHT">);
    15: WRITE (UTU, <" ERROR: 'DQ' OF 'DI' VERWACHT">);
    16: WRITE (UTU, <" ERROR: WAARDE MOET POSITIEF ZIJN">);
    17: WRITE (UTU, <" ERROR: 'O <= EPS <= 1'">);
    18: WRITE (UTU, <" ERROR: ER IS VOOR VERANDERLIJKE ",
        "LADINGSDICHTHEID GEKOZEN ", /,
        "
        MAAR ER IS GEEN GOEDE RAND INGEVOERD">);
    END;
    GO TO STOP
END ERROR;

PROCEDURE WARN (A);

```

```

    VALUE A;
    INTEGER A;
BEGIN
    WRITE (UTU, <*(" "), "*">, PC - 2);
    CASE A OF
    BEGIN
        1: WRITE (UTU, <"WARNING: DEELRAND ONVINDBAAR">);
        2: WRITE (UTU, <"WARNING: ZINLOSE KOPPELING">);
        3: WRITE (UTU, <"WARNING: DEELRANDEN NIET COMPATIBEL">);
        4: WRITE (UTU, <"WARNING: (GEDEELTE VAN) RAND AL EENS GEHAD">);
        5: WRITE (UTU, <"WARNING: GROEP MAG NIET KLEINER DAN 1 ZIJN">);
    END;
END WARN;

PROCEDURE TERM (CH);
VALUE CH;
REAL CH;
BEGIN
    BOOLEAN TC;
    IF CH = "(" THEN TC := (SYM = "(" );
    ELSE IF CH = ")" THEN TC := (SYM = ") " );
    ELSE IF CH = "," THEN TC := (SYM = ", " );
    ELSE IF CH = ";" THEN TC := (SYM = "; " );
    ELSE IF CH = ":" THEN TC := (SYM = ": " );
    IF NOT TC THEN
    BEGIN
        WRITE (UTU, <*(" "), "*">, PC-2);
        WRITE (UTU, <"ERROR: ' ", A1, "' VERWACHT">, CH);
        GO TO STOP
    END
END TERM;

PROCEDURE INT (A);
INTEGER A;
BEGIN
    IF SYM = "INTEGER " THEN A := NUMI
    ELSE ERROR (1)
END INT;

PROCEDURE PREAL (X);
REAL X;
BEGIN
    IF SYM = "INTEGER " THEN X := NUMI
    ELSE IF SYM = "REAL " THEN X := NUMR
    ELSE ERROR (2)
END PREAL;

PROCEDURE RSP (N, RP);
VALUE N;
INTEGER N;
INTEGER ARRAY RP [*];
BEGIN
    INTEGER I;
    FOR I := 0 TO 1 N - 1 DO
    BEGIN

```



```

        INT (RP [I]); NEXTSYM;
        TERM (" , "); NEXTSYM
    END;
    INT (RP [N])
END RSP;

BEGIN
    BOOLEAN FERROR;
    EBCDIC ARRAY FTITEL [0 : 80];
    FERROR = FALSE;
    IF NOT INTERF.RESIDENT THEN
    BEGIN
        WRITE (UTU, <"FILE 'INTERF' IS NIET AANWEZIG">);
        REPLACE FTITEL BY INTERF.TITLE;
        WRITE (UTU, <"TITEL: ", A60>, FTITEL);
        FERROR = TRUE
    END;
    IF NOT MESH.RESIDENT THEN
    BEGIN
        WRITE (UTU, <"FILE 'MESH' IS NIET AANWEZIG">);
        REPLACE FTITEL BY " " FOR B1;
        REPLACE FTITEL BY MESH.TITLE;
        WRITE (UTU, <"TITEL: ", A60>, FTITEL);
        FERROR = TRUE
    END;
    IF NOT DATAMG.RESIDENT THEN
    BEGIN
        WRITE (UTU, <"FILE 'DATAMG' IS NIET AANWEZIG">);
        REPLACE FTITEL BY " " FOR B1;
        REPLACE FTITEL BY DATAMG.TITLE;
        WRITE (UTU, <"TITEL: ", A60>, FTITEL);
        FERROR = TRUE
    END;
    IF FERROR THEN GO TO STOP
END;

REPLACE WOORD [1 , 1] BY "B           ";
REPLACE WOORD [2 , 1] BY "CA           ";
REPLACE WOORD [3 , 1] BY "DELTA          ";
REPLACE WOORD [4 , 1] BY "DQ           ";
REPLACE WOORD [5 , 1] BY "DT           ";
REPLACE WOORD [6 , 1] BY "ELEKTRODE       ";
REPLACE WOORD [7 , 1] BY "GEARDE          ";
REPLACE WOORD [8 , 1] BY "KARAKTERISTIEKEN ";
REPLACE WOORD [9 , 1] BY "KOPPELEN         ";
REPLACE WOORD [10, 1] BY "LADINGSDICHTHEID ";
REPLACE WOORD [11, 1] BY "MAXSTAP          ";
REPLACE WOORD [12, 1] BY "MIS              ";
REPLACE WOORD [13, 1] BY "NO               ";
REPLACE WOORD [14, 1] BY "PLOTS            ";
REPLACE WOORD [15, 1] BY "RANDELEMENTEN   ";
REPLACE WOORD [16, 1] BY "RANDEN             ";
REPLACE WOORD [17, 1] BY "RANDSPANNINGEN   ";
REPLACE WOORD [18, 1] BY "VERANDERLIJKE    ";
REPLACE WOORD [19, 1] BY "ZWEUENDE         ";

```

```

REPLACE LINE BY " " FOR 73;
P:= LINE [72];
PC:= 73; EOFI:= FALSE;
NEXTSYM;

```

```

IF SYM NEQ "NAAM " THEN ERROR (3);
WRITE (F, <A72>, NAAM); NEXTSYM;
TERM ("("); NEXTSYM;
INT (NNODES); NEXTSYM;
TERM (","); NEXTSYM;
INT (NEL); NEXTSYM;
TERM (")");

```

```

BEGIN

```

```

    DEFINE MAXAB = 1000 #,
           MAXGR = 25 #;

```

```

    INTEGER EL, AEL, AELN, G, N, I, J, K, RNNODES,
           ACHAR, NGROUPS, CC;

```

```

    BOOLEAN OK, PW;

```

```

    REAL EPS, CMF;

```

```

    INTEGER ARRAY RAND, DR, DR2, PRES, NN [0 : NNODES],
                SUCC, LAST [1 : NNODES],
                TOP [0 : NEL, 0 : 8],
                EINDEL, GAELN [0 : MAXGR], RANDBP, INN [0 : 3],
                HRBP [0 : 4];

```

```

    ARRAY X, Y [0 : NNODES + 1],
           XBP, YBP [0 : MAXAB],
           CHAR [0 : MAXGR, 1 : 2];

```

```

    EPS:= 6@-5;

```

```

    BASISPOINTS (XBP, YBP, DATAMG);

```

```

    FOR I:= 1 TO1 NNODES DO

```

```

        BEGIN

```

```

            PRES [I]:= 0;

```

```

            SUCC [I]:= I; LAST [I]:= I

```

```

        END;

```

```

    WRITE (F, <"BEGIN OPTION 9-1;"> );

```

```

    WRITE (F, <" ELEMENT TRIL3,1(1,1) FREEDOMS [1(1)3:X] CHAR 2"/X24
           "COORD 2,2,2 DERIV 0,3"/"ENDELEMENT"/>);

```

```

    WRITE (F, <" ELEMENT TRIL6,2(1,0) FREEDOMS [1(1)6:X] CHAR 2"/X24
           "COORD 2,2,2,2,2,2 DERIV 0,3"/"ENDELEMENT"/>);

```

```

    READTRIQUASTRUCTURE (MESH, EINDEL, GAELN, TOP, X, Y, PRES,
                        NGROUPS, NNODES, OK);

```

```

    IF NOT OK THEN ERROR (4);

```

```

    FOR G:= 2 TO1 NGROUPS DO

```

```

        IF GAELN [G] NEQ GAELN [1] THEN OK:= FALSE;

```

```

    IF NOT OK THEN ERROR (5);

```

```

    FINDBOUND (RAND, OK, TOP, NNODES, NGROUPS, EINDEL, GAELN);

```

```

    IF NOT OK THEN ERROR (6);

```

```

    WRITE (F, <"STRUCTURE S"> );

```

```

WRITE (F, <" GROUP A TYPE TRIL3">);

AELN:= GAELN [1];
K:= 80 DIV (6 + 5 * AELN);
AEL:= EINDEL [NGROUPS];
EL:= 0;
WHILE EL < AEL DO
  WRITE (F, <80 (15, ":", * (14, ","), 14, ";") >,
        THRU MIN (K, AEL - EL) DO
    [ TOP [EL:= EL + 1, 0], AELN - 1,
      FOR I:= 1 TO1 AELN DO TOP [EL, I] ] );

NEXTSYM;
IF SYM NEQ "KARAKTERISTIEKEN " THEN ERROR (7);
NEXTSYM;
FOR G:= 1 TO1 NGROUPS DO
  BEGIN
    PREAL (CHAR [G,1]); NEXTSYM;
    IF SYM = "," THEN
      BEGIN
        NEXTSYM;
        PREAL (CHAR [G, 2]); NEXTSYM;
        TERM (",")
      END ELSE
        IF SYM = ";" THEN CHAR [G, 2]:= 0
        ELSE ERROR (8);
    NEXTSYM
  END;
WRITE (F, <" CHARACTERISTICS "/X18, * (E12.5, ","/ X18),
      E12.5, ";">, 1, CHAR [1, 1], CHAR [1, 2]);
IF NGROUPS > 1 THEN
  WRITE (F, <" EXCEPT",
        * (/ ,110, "(1)", 14, ":", * (E12.5, ","/ X18), E12.5, ";">>
        NGROUPS - 1,
        FOR G:= 2 TO1 NGROUPS DO
          [ EINDEL [G - 1] + 1, EINDEL [G],
            1, CHAR [G, 1], CHAR [G, 2]
          ] );

IF SYM = "CA " THEN
  BEGIN
    NEXTSYM;
    TERM ("("); NEXTSYM;
    PREAL (CMF); NEXTSYM;
    TERM (")"); NEXTSYM
  END
ELSE CMF:= 1;
J:= 0;
FOR I:= 1 TO1 NNODES DO IF PRES [I] > 0 THEN
  BEGIN J:= J + 1; NN [J]:= I END;
RNNODES:= J;
WRITE (F, <" COORDINATES"/
      (2 (15, ":", E12.5, ",", E12.5, ";")) >,
      FOR I:= 1 TO1 J DO
        [ NN [I], CMF * X [NN [I]], CMF * Y [NN [I]] ] );

```

```

IF SYM = "RANDELEMENTEN " THEN
BEGIN
  J:= 0;
  NEXTSYM;
  WHILE SYM = "NAAM " DO
  BEGIN
    NEXTSYM;
    TERM ("("); NEXTSYM;
    INT (ACHAR); NEXTSYM;
    TERM (")"); NEXTSYM;
    RSP (3, RANDBP);
    DEELRAND (DR);
    NEXTSYM;
    TERM (":"); NEXTSYM;
    IF ACHAR > 0 THEN
    BEGIN
      ARRAY PCHAR [0 : ACHAR -1];

      FOR I:= 0 TO1 ACHAR - 2 DO
      BEGIN
        PREAL (PCHAR [I]); NEXTSYM;
        TERM (","); NEXTSYM
      END;
      PREAL (PCHAR [ACHAR -1]); NEXTSYM;
      IF OK THEN
      BEGIN
        J:= J + 1;

        WRITE (F, <"GROUP R", I1, " TYPE ", A23 /
          (4(I6, ":", I4, ":", I4, ";"))>,
          J, NAAM,
          FOR I:= 2 TO1 DR [0] DO
            [ 10000*J-1+I, DR [I-1], DR [I] ]);
        WRITE (F, <" CHARACTERISTICS",
          * (E12.5, ":", " / X17), E12.5, ";">,
          ACHAR - 1, PCHAR);
        WRITE (F, <" COORDINATES"/
          (2 (I6, ":", E12.5, ":", E12.5, ";"))>,
          FOR I:= 1 TO1 DR [0] DO
            [ DR [I], X [DR [I]], Y [DR [I]] ]);

          END
        END;
        TERM (";"); NEXTSYM
      END
    END;
  BEGIN
    DEFINE CONNECT (AA, BB) =
    BEGIN
      LA:= LAST [AA];
      LAST [AA]:= LAST [BB];
      H:= SUCC [AA];
      WHILE H NEQ AA DO
      BEGIN

```

```

        LAST [H]:= LAST [BB];
        H:= SUCC [H]
    END;
    H:= SUCC [LA];
    SUCC [LA]:= SUCC [LAST [BB]];
    SUCC [LAST [BB]]:= H
END CONNECT #;

INTEGER AA, BB, LA, H;
BOOLEAN TRANSF;
TRANSF:= FALSE;
IF SYM = "KOPPELEN " THEN
BEGIN
    NEXTSYM;
    WHILE SYM = "INTEGER " DO
    BEGIN
        RSP (4, HRBP);
        RANDBP [0]:= HRBP [0]; RANDBP [1]:= HRBP [2];
        RANDBP [2]:= HRBP [3]; RANDBP [3]:= HRBP [4];
        DEELRAND (DR);
        RANDBP [0]:= HRBP [1];
        DEELRAND (DR2);
        IF OK THEN
        BEGIN
            % CHECK FOR CONSISTENCY
            IF DR [0] = DR2 [0] THEN
                IF DR [1] NEQ DR2 [1] THEN
                BEGIN
                    TRANSF:= TRUE;
                    FOR I:= 1 TO1 DR [0] DO
                    BEGIN
                        AA:= DR [I]; BB:= DR2 [I];
                        IF LAST [AA] NEQ LAST [BB] THEN
                            CONNECT (AA, BB)
                    END
                END ELSE WARN (2)
                ELSE WARN (3)
            END;
            NEXTSYM;
            TERM (""); NEXTSYM
        END
    END;
END;

IF SYM = "ZWEUENDE " THEN
BEGIN
    NEXTSYM;
    IF SYM = "ELEKTRODE " THEN
    BEGIN
        NEXTSYM;
        WHILE SYM = "INTEGER " DO
        BEGIN
            RSP (3, RANDBP);
            DEELRAND (DR);
            IF OK THEN
            BEGIN

```

```

        TRANSF:= TRUE; BB:= DR [1];
        FOR I:= 2 TO1 DR [0] DO
        BEGIN
            AA:= DR [I];
            IF LAST [AA] NEQ LAST [BB] THEN
                CONNECT (AA, BB)
            END
        END;
        NEXTSYM;
        TERM (";"); NEXTSYM
    END
END;

IF TRANSF THEN
BEGIN
    WRITE (F, <" TRANSFORMATIONS">);
    FOR I:= 1 TO1 RNNODES DO
    BEGIN
        IF LAST [I] NEQ I THEN
            WRITE (F, <"      [", I4,":X] = [", I4,":X];">,
                I, LAST [I])
        END
    END
END;

BEGIN
    REAL SPAN;
    INTEGER AA, AUS, AGR;
    ARRAY USPAN, PRSU [1 : RNNODES];

    AUS:= 0;
    FOR I:= 1 TO1 RNNODES DO PRSU [I]:= 0;
    IF SYM = "RANDSPANNINGEN " THEN
    BEGIN
        NEXTSYM;
        WHILE SYM = "INTEGER " DO
        BEGIN
            RSP (3, RANDBP);
            DEELRAND (DR);
            NEXTSYM;
            TERM (":"); NEXTSYM;
            PREAL (SPAN);
            IF OK THEN
            BEGIN
                FOR I:= 1 TO1 DR [0] DO
                BEGIN
                    AA:= LAST [DR [I]];
                    IF PRSU [AA] = 0 THEN AUS:= AUS + 1;
                    PRSU [AA]:= 1;
                    USPAN [AA]:= SPAN
                END
            END;
            NEXTSYM;
            TERM (":"); NEXTSYM
        END
    END

```

```

        END
    END;

    AGR:= 0;
    IF SYM = "GEARDE " THEN
    BEGIN
        NEXTSYM;
        IF SYM = "RANDEN " THEN
        BEGIN
            NEXTSYM;
            WHILE SYM = "INTEGER " DO
            BEGIN
                RSP (3, RANDBP);

                DEELRAND (DR);
                IF OK THEN
                BEGIN
                    FOR I:= 1 TO1 DR [0] DO
                    BEGIN
                        AA:= LAST [DR [I]];
                        IF PRSU [AA] NEQ 2 THEN AGR:= AGR + 1;
                        IF PRSU [AA] = 1 THEN AUS:= AUS - 1;
                        PRSU [AA]:= 2
                    END
                END;
                NEXTSYM;
                TERM (";","); NEXTSYM
            END
        END
    END;

    IF AUS > 0 THEN
    BEGIN
        WRITE (F, <" PRESCRIBE">);
        WRITE (F, <<(X6, 5(*("[", 14, ": X]"))>>,
            FOR I:= 1 TO1 RNNODES DO
                IF PRSU [I] = 1 THEN [1, I])
    END;

    IF AGR > 0 THEN
    BEGIN
        WRITE (F, <" SUPPRESS">);
        WRITE (F, <<(X6, 5(*("[", 14, ": X]"))>>,
            FOR I:= 1 TO1 RNNODES DO
                IF PRSU [I] = 2 THEN [1, I])
    END;

    WRITE (F, <"ENDSTRUCTURE"> );

    IF AUS > 0 THEN
    BEGIN
        WRITE (F, <"LOAD L1 FOR S">);
        WRITE (F, <<(X1,2 (*("[", 14, ": X] ",E12.5, ";"))>>,
            FOR I:= 1 TO1 RNNODES DO
                IF PRSU [I] = 1 THEN [1, I, USPAN [I]]);

```

```

        WRITE (F, <"ENDLOAD">);
        WRITE (F, <"LHU X1 OF L1 ENDLHU">)
    END
END;

IF SYM = "PLOTS " THEN
BEGIN
    WRITE (F, <X1/ "OPTION 74-1, 75-1, 76-1, 68-1;"> );
    NEXTSYM
END;
WRITE (F, <X1 /"INSTRUCTION 100 S (100 - 14)"> );
WRITE (F, <"END"> );

IF SYM = "VERANDERLIJKE " THEN
BEGIN
    NEXTSYM;
    IF SYM = "LADINGSDICHTHEID " THEN
    BEGIN
        FILE G1 (KIND = DISK, MAXRECSIZE = 14, BLOCKSIZE = 420,
                AREASIZE= 600, AREAS= 25, TITLE= "INREADREPFEM1.")
        G2 (KIND = DISK, MAXRECSIZE = 14, BLOCKSIZE = 420,
            AREASIZE= 600, AREAS= 25, TITLE= "INREADREPFEM2.")
        REAL KEUZE, NO, B, DT, MTS, DQ, SC;
        INTEGER ASG, ATB, H, JJ, MAS;
        INTEGER ARRAY NSG1 [1 : NNODES, 0 : 3],
                    ASG2 [1 : NNODES + 1],
                    NSG2 [0 : NEL, 0 : 3],
                    CGB [1 : EINDEL [NGROUPS], 0 : 1];

        PROCEDURE DRIEHOEKEN1 (DR, EDP, LDP, DHK);
            VALUE EDP, LDP;
            INTEGER EDP, LDP;
            INTEGER ARRAY DR [*],
                        DHK [* , *];

        BEGIN
            INTEGER I, J, H1, H2, HMIN, HMAX;
            BOOLEAN CYC;
            INTEGER ARRAY PRAND [1 : NNODES];

            FOR I:= 1 TO1 NNODES DO PRAND [I]:= 4 * DR [0];
            FOR I:= 1 TO1 DR [0] - 1 DO PRAND [DR [I]]:= I;
            CYC:= FALSE;
            IF DR [0] > 3 THEN
                IF DR [1] = DR [DR [0]] THEN CYC:= TRUE;
            IF NOT CYC AND DR [0] > 0 THEN
                PRAND [DR [ DR [0]]]:= DR [0];
            FOR I:= EDP TO1 LDP DO
            BEGIN
                FOR J:= 1 TO1 3 DO
                BEGIN
                    H1:= PRAND [TOP [1, J]];
                    H2:= PRAND [TOP [1, (J MOD 3) + 1]];
                    HMIN:= MIN (H1, H2);
                    HMAX:= MAX (H1, H2);
                END
            END
        END
    END
END

```



```

        IF HMAX - HMIN = 1 THEN
        BEGIN
            DHK [ HMIN, 0] := I;
            DHK [ HMIN, 1] := DR [HMIN];
            DHK [ HMIN, 2] := DR [HMAX];
            DHK [ HMIN, 3] := TOP [I, (J + 1) MOD 3 + 1]
        END ELSE
        IF CYC AND HMAX - HMIN = DR [0] - 2 THEN
        BEGIN
            DHK [ HMAX, 0] := I;
            DHK [ HMAX, 1] := DR [HMAX];
            DHK [ HMAX, 2] := DR [HMIN];
            DHK [ HMAX, 3] := TOP [I, (J + 1) MOD 3 + 1]
        END
    END
END
END DRIEHOEKEN1;

```

```

PROCEDURE DRIEHOEKEN2 (DR, EDP, LDP, DHK, DHK1);
    VALUE EDP, LDP;
    INTEGER EDP, LDP;
    INTEGER ARRAY DR, DHK1 [*];
    INTEGER ARRAY DHK [*, *];

BEGIN
    INTEGER I, J, H1, H2;
    INTEGER ARRAY HDH [EDP : LDP, 0 : 1],
                HDH1, Q [1 : DR [0]],
                NSG [1 : DR [0], 0 : 3],
                PRAND [1 : NNODES];
    BOOLEAN ARRAY DPRES [EDP : LDP];

    DRIEHOEKEN1 (DR, EDP, LDP, NSG);
    FOR I := 1 TO1 DR [0] DO HDH1 [I] := 0;
    FOR I := EDP TO1 LDP DO HDH [I, 0] := 0;
    FOR I := 1 TO1 NNODES DO PRAND [I] := DR [0];
    FOR I := 1 TO1 DR [0] - 1 DO PRAND [DR [I]] := I;
    FOR I := EDP TO1 LDP DO DPRES [I] := FALSE;
    FOR I := 1 TO1 DR [0] - 1 DO DPRES [NSG [I, 0]] := TRUE;
    FOR I := EDP TO1 LDP DO
    BEGIN
        IF NOT DPRES [I] THEN
        BEGIN
            H1 := DR [0];
            FOR J := 1 TO1 3 DO
                IF PRAND [TOP [I, J]] < H1 THEN
                BEGIN
                    H1 := PRAND [TOP [I, J]];
                    H2 := J
                END;
            IF H1 < DR [0] THEN
            BEGIN
                HDH [I, 0] := H1;
                HDH [I, 1] := H2;
            END
        END
    END

```

```

        HDH1 [K1]:= HDH1 [K1] + 1
    END
END
END;
DHK1 [1]:= 0;
FOR I:= 2 TO1 DR [0] DO DHK1 [I]:= DHK1 [I - 1] + 1 +
        HDH1 [I - 1];
FOR I:= 1 TO1 DR [0] DO Q [I]:= DHK1 [I];
FOR I:= 1 TO1 DR [0] - 1 DO
BEGIN
    FOR J:= 0 TO1 3 DO DHK [Q [I], J]:= NSG [I, J];
    Q [I]:= Q [I] + 1
END;
FOR I:= EDP TO1 LDP DO
    IF HDH [I, 0] NEQ 0 THEN
    BEGIN
        DHK [Q [HDH [I, 0]], 0]:= 1;
        FOR J:= 1 TO1 3 DO DHK [Q [HDH [I, 0]], J]:=
            TOP [I, (HDH [I, 1] - 2 + J) MOD 3 + 1];
        Q [HDH [I, 0]]:= Q [HDH [I, 0]] + 1
    END
END DRIEHOEKEN2;

WRITE (G1, /, RNNODES, EINDEL [NGROUPS], NGROUPS);
ASG:= 0; AIB:= 0;
NEXTSYM;
WHILE SYM = "INTEGER " DO
BEGIN
    RSP (4, HRBP);
    RANDBP [0]:= HRBP [0]; RANDBP [1]:= HRBP [2];
    RANDBP [2]:= HRBP [3]; RANDBP [3]:= HRBP [4];
    IF (HRBP [0] >= 1) AND (HRBP [1] >= 1) THEN
    BEGIN
        DEELRAND (DR);
        IF OK THEN
        BEGIN
            RANDBP [0]:= HRBP [1];
            DEELRAND (DR2);
            IF OK THEN
            BEGIN
                IF DR [0] = DR2 [0] THEN
                BEGIN
                    FOR I:= 1 TO1 ASG DO
                        FOR J:= 1 TO1 DR [0] - 1 DO
                            IF (CGB [I, 0] = DR [J] AND
                                CGB [I, 1] = DR [J + 1]) OR
                                (CGB [I, 0] = DR [J + 1] AND
                                    CGB [I, 1] = DR [J])
                                THEN OK:= FALSE;
                            IF OK THEN
                            BEGIN
                                IF ASG = 0 AND DR [0] > 1 THEN
                                BEGIN
                                    WRITE (G1, <"1,">);
                                    WRITE (G1, <" ">);
                                END
                            END
                        END
                    END
                END
            END
        END
    END
END

```

```

END;
DRIEHOEKEN1 (DR,EINDEL [HRBP [0] - 1 ] +
              EINDEL [ HRBP [0]], NSG1);
FOR I:= 1 TO1 DR [0] - 1 DO
BEGIN
  CGB [ASG + I, 0]:= NSG1 [I, 1];
  CGB [ASG + I, 1]:= NSG1 [I, 2]
END;
ASG:= ASG + DR [0] - 1;
H:= EINDEL [HRBP [1] - 1] + 1;
DRIEHOEKEN2 (DR2, H, EINDEL [HRBP [1]],
              NSG2, ASG2);
ATB:= ATB + ASG2 [DR [0]];
FOR I:= 1 TO1 DR [0] -1 DO
BEGIN
  WRITE (G1,<4(I6, X1)>,FOR J:= 0 TO1 3 I
        NSG1 [I, J]);
  J:= ASG2 [I];
  WHILE J < ASG2 [I+1] DO
  BEGIN
    WRITE (G1, <4(I6, X1)>,
          FOR JJ:= 0 TO1 3 DO NSG2 [J, JJ]);
    J:= J + 1
  END;
  WRITE (G1, <" ">);
END
END ELSE WARN (4)
END ELSE WARN (3)
END
END
END ELSE WARN (5);
NEXTSYM;
TERM (";"); NEXTSYM
END;
IF ASG = 0 THEN ERROR (18);
WRITE (G1 [2], /, ASG, ATB);
IF SYM NEQ "MAXSTAP " THEN ERROR (13);
NEXTSYM;
TERM (";"); NEXTSYM;
INT (MAS); NEXTSYM;
TERM (";"); NEXTSYM;
IF SYM NEQ "NO " THEN ERROR (11);
NEXTSYM;
TERM (";"); NEXTSYM;
PREAL (NO); NEXTSYM;
IF NO <= 0 THEN ERROR (16);
TERM (";"); NEXTSYM;
IF SYM NEQ "B " THEN ERROR (12);
NEXTSYM;
TERM (";"); NEXTSYM;
PREAL (B); NEXTSYM;
IF B <= 0 THEN ERROR (16);
TERM (";"); NEXTSYM;
IF SYM = "DQ " THEN
BEGIN

```

```

WRITE (G2, <"2,">); WRITE (G2, /, MAS);
WRITE (G2, /, NO, B);
NEXTSYM;
TERM (":"); NEXTSYM;
PREAL (DQ);
IF DQ <= 0 THEN ERROR (16);
WRITE (G2, /, DQ); NEXTSYM;
TERM (":"); NEXTSYM;
IF SYM NEQ "MTS " THEN ERROR (14);
NEXTSYM;
TERM (":"); NEXTSYM;
PREAL (MTS);
IF MTS <= 0 THEN ERROR (16);
WRITE (G2, /, MTS); NEXTSYM;
TERM (":"); NEXTSYM;
END ELSE
IF SYM = "DI " THEN
BEGIN
NEXTSYM;
TERM (":"); NEXTSYM;
IF (SYM = ";") AND (MAS <= 0) THEN
BEGIN
WRITE (G2, <"3 ">); WRITE (G2, /, MAS);
WRITE (G2, /, NO, B);
NEXTSYM
END ELSE
BEGIN
PREAL (DI);
IF DI <= 0 THEN ERROR (16);
NEXTSYM;
IF SYM = "; " THEN
BEGIN
WRITE (G2, <"1,">); WRITE (G2, /, MAS);
WRITE (G2, /, NO, B);
WRITE (G2, /, DI);
NEXTSYM;
END ELSE
IF SYM = ", " THEN
BEGIN
WRITE (G2, <"3,">); WRITE (G2, /, MAS);
WRITE (G2, /, NO, B);
WRITE (G2, /, DI);
FOR I:= 2 TO1 MAS DO
BEGIN
TERM (","); NEXTSYM;
PREAL (DI);
IF DI <= 0 THEN ERROR (16);
WRITE (G2, /, DI); NEXTSYM
END;
TERM (":"); NEXTSYM;
END ELSE ERROR (8)
END
END ELSE ERROR (15);
IF SYM = "DELTA " THEN
BEGIN

```

```

        NEXTSYM;
        TERM (":"); NEXTSYM;
        PREAL (SC);
        IF (SC < 0) OR (SC > 1) THEN ERROR (17);
        NEXTSYM;
        TERM (";"); NEXTSYM
    END ELSE SC: = 0;
    WRITE (G2, /, SC);
    LOCK (G1, CRUNCH);
    LOCK (G2, CRUNCH)
END
END;

IF SYM NEQ "END " THEN
BEGIN
    PW: = FALSE;
    FOR I: = 1 TO 1 NORW DO
        IF WOORD [I, 1] = SYM FOR 20 THEN PW: = TRUE;
    IF PW THEN ERROR (9)
    ELSE ERROR (10)
END;
END;
LOCK (F, CRUNCH);
STOP: LOCK (UTV, CRUNCH)
END.

```