



Institutionen för vattenbyggnad
Chalmers Tekniska Högskola

Department of Hydraulics
Chalmers University of Technology

GRAFISK PRESENTATION AV UTDATA FRÅN DATOR- PROGRAMMET MODEX

av Francisco Herrera

Examensarbete 1991:7

Göteborg 1991



Institutionen för vattenbyggnad
Chalmers tekniska högskola

Department of Hydraulics
Chalmers University of Technology

Grafisk presentation av utdata från
datorprogrammet MODEX

av

Francisco Herrera

Examensarbete
Nr. 1991:7

Göteborg 1991

Adress: Institutionen för Vattenbyggnad
Chalmers Tekniska Högskola
412 96 GÖTEBORG

Telefon: 031-72 10 00
Telefax: 031-72 21 28

Innehållsförteckning

	sid.
Förord	1
Referat	1
1 Inledning	2
2 Kort beskrivning av datorprogrammet MODEX	2
3 Ny utdatautformning	3
3.1 Inledning	3
3.2 Integrering av subrutinerna i modex	3
3.3 Självständigt program	4
4 Beskrivning av subrutinerna	6
4.1 Inledning	
4.2 Subrutin diagram	6
4.3 Subrutin minmax	8
4.4 Subrutin scala	8
4.5 Subrutin graph	8
4.6 Subrutin doaxel	8
4.7 Subrutin pline	8
4.8 Subrutin tabell	8
4.9 Subrutin tabelstd	8
4.10 Subrutin tabellpo1	9
4.11 Subrutin tabellpos	9
4.12 Subrutin textstring	9
4.13 Subrutin refile	9
4.14 Subrutin wrfile	9
4.15 Subrutin gmode	9
4.16 Subrutin clear	9
4.17 Subrutin nmode	9
5 Exempel	10
6 Förslag till fortsatt arbete	10
Referenser	10
Appendix 1	12
Appendix 2	26
Appendix 3	39

FÖRORD

Detta examensarbete har tillkommit på initiativ av Professor Lars Bergdahl vid institutionen för vattenbyggnad CTH. Han ansåg det lämpligt att bearbeta ett existerande datorprogram för dynamisk analys av förankringskablar, så att man kunde hantera de stora utdatamängderna från programmet och visa dem på skärmen i diagramform, på ett enkelt sätt.

Jag vill tacka honom för god hjälp under arbetets gång. Dessutom vill jag tacka tekn.lic. Nils Mårtensson vid Dynomar AB för materialsupport.

Göteborg November 1991

Francisco Herrera.

REFERAT

I denna skrift redovisas ett sätt att presentera stora mängder utdata, vilket är vanligt i vissa datorprogram. Detta görs i diagramform eller tabellform direkt på skärmen. Önskas det, så kan både diagram och tabell visas samtidigt. Scrollfunktioner finns inbyggda för tabeller som inte ryms på på skärmen i sin helhet. Ett antal subroutiner har tillverkats för ändamålet. Dessa är skrivna i Laheys Fortran 77 och Graphorias Graphics Library.

Ett komplett program presenteras också för att demonstrera ovanstående egenskaper. Datamängden har framställts genom körning av ett datorprogram för dynamisk analys av förankringskablar, MODEX, skrivet av Jan Lindahl vid institutionen för vattenbyggnad CTH.

1. INLEDNING.

Vid dynamisk analys av t ex förankringskablar eller risers genereras stora data-mängder, som lämpligtvis presenteras i diagramform för att göras överskådliga. Diagrammen bör först presenteras på skärmen och sedan kunna sändas till printern vid behov.

I det föreliggande arbetet har utdata från kabelodynamikprogrammet MODEX använts som exempel. Därför presenteras detta först innan det skapade grafiska presentationsprogrammet beskrivs.

Som programmeringsspråk har användts Laheys Fortran 77 (F77L) under operativ systemet MS-DOS version 3.0 och Graphorias Graphics Library version 1.0.

2. KORT BESKRIVNING AV DATORPROGRAMMET MODEX

För att hålla en flytande konstruktion vid en given position är det nödvändigt att använda någon form av förankring. Vissa konstruktioner kan ha enpunktsförankring medan andra kan utnyttja ett system av förankringskablar.

Förankringssystemet till en stor flytande konstruktion utgöres som regel av kätting eller vajer.

För att få en realistisk bild av krafterna i en förankringskabel är det nödvändigt att utföra en studie av den dynamiska spänningens förlopp i kabeln. För detta ändamål kan man använda ett datorprogram , MODEX, som har utvecklats av Jan Lindahl vid institutionen för vattenbyggnad CTH.

Den teoretiska bakgrunden till datorprogrammet MODEX är hämtad ur rapporterna "Dynamic analysis of mooring cables" (Lindahl, Sjöberg 1983) och "Implicit numerical solution of the equation of motion of a mooring cable", (Lindahl 1984).

Finita elementmetoden har utnyttjats för att transformera de partiella differentialekvationer, som beskriver rörelsen av kabeln, till ett system av ordinära tidsberoende differentialekvationer. Dessa löses i MODEX med explicit numerisk

integration. Programmet förutsätter att rörelserna eller krafterna i båda kabeländarna är kända. De kan vara en realisering av den stokastiska process som konstruktionens rörelse utgör eller några andra rörelser eller krafter som man har intresse av att studera.

Målsättningen är att kunna göra en detaljerad studie av de dynamiska krafter som kan uppkomma i en förakringskabel. Programmet förutsätter också att ett antal andra storheter är kända. Dessa är beskrivna i detalj i programmets manual. Förutom de dynamiska krafterna, kan man studera kabelns rörelser, hastigheter, accelerationer och töjningar i olika lägen och tidpunkter .

3. NY UTDATAUTFORMNING

3.1 Inledning

För att underlätta studiet av den massiva utdatamängd, som programmet genererar, kan man utforma en utvidgning till programmet så att man kan visa utdata i diagramform direkt på skärmen.

Ett annat sätt är att spara alla utdata i externa filer och utveckla ett separat datorprogram som tar hand om dessa filer och den grafiska framställningen. För detta ändamål presenteras här ett antal subrutiner som är tänkta att lösa uppgiften så enkelt som möjligt, och ett självständigt program som demonstrerar fallet när man vill se och utvärdera resultatet om man har sparat alla utdata i externa filer.

3.2 INTEGRERING AV SUBRUTINERNA I MODEX

Den enda subrutinen som användaren har behov av att komma i kontakt med för att framställa ett diagram heter just "diagram".

På lämpligt ställe i programmet kan man permanent inkorporera de rader som behövs för att kalla denna subrutin. Sedan måste programmet kompileras på nytt och under länknigen ta med hela subrutinpaketet i denna skrift och Graphorias Graphics Library. Olika tabellrutiner kan man som ovan, integrera i programmet om man önskar se resultatet i formaterade tabeller, som i princip kan vara mycket

stora, eftersom scrollfunktioner finns inbyggda och rullning av dessa är möjlig genom att använda piltangenterna, page up, page down, home, och end. Efter detta kan man framställa grafer och tabeller under programmets körning.

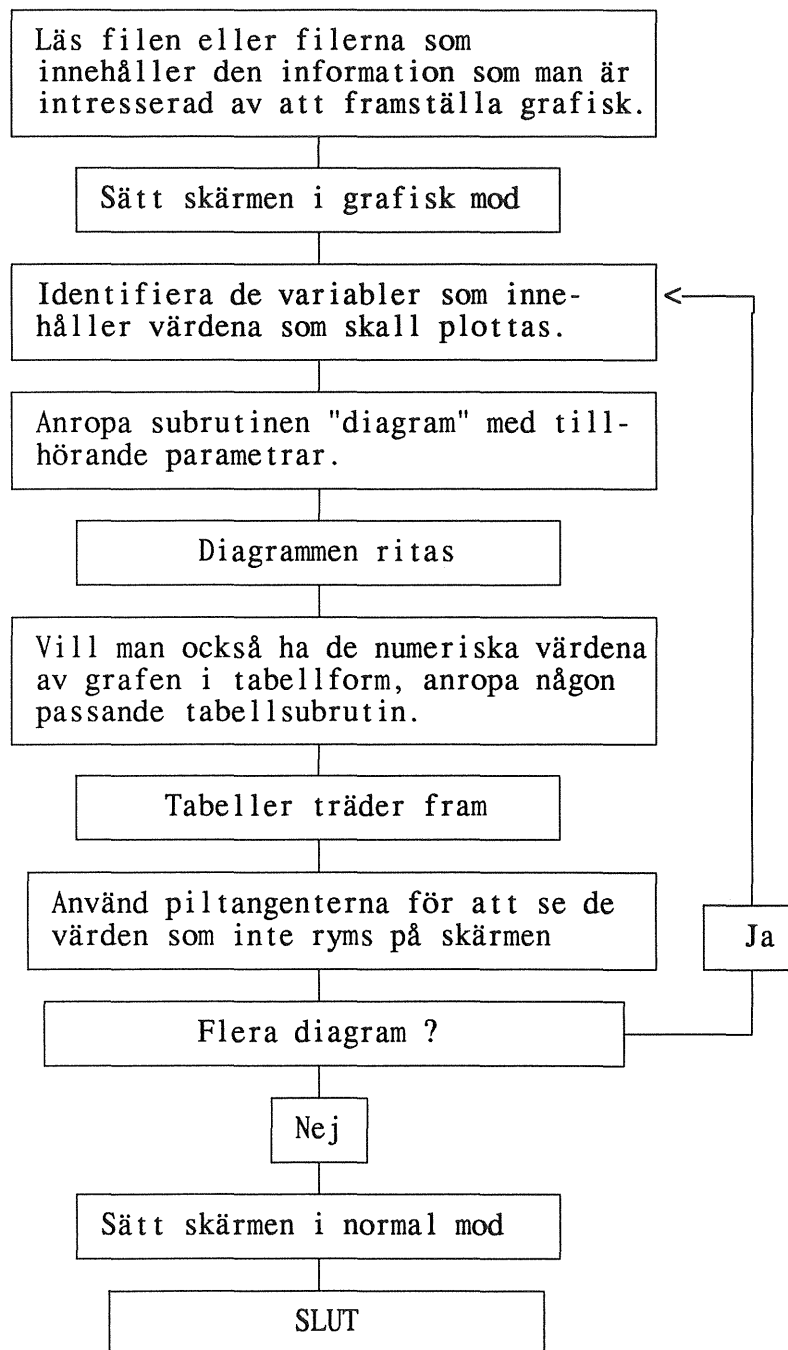
De nödvändiga aktuella parametrar som styr utseendet på grafen, position på skärmen och en del andra egenskaper finns angivna och förklarade i appendix 1 tillsammans med den kompletta listningen av subrutinerna. Subrutinen "diagram" kallar andra subrutiner men dessa är självständiga och behöver endast länkas med huvudprogrammet. Ett antal subrutiner för att framställa tabeller finns i några variationer i appendix 1. Dessa är enkla att förstå och att förändra efter eget behov.

3.2 SJÄLVSTÄNDIGT PROGRAM

Ett bra sätt att analysera resultatet av en programkörning är att göra detta med ett separat program som kan köras när man önskar det och även kunna se resultaten från tidigare körningspass.

Den enda förändring av moderprogrammet, i detta fall MODEX, som krävs är att lägga till de filer som sparar resultaten i externa filer för senare bearbetning. Att utveckla ett separat datorprogram som tar hand om dessa filer och den grafiska framställningen kräver en god kännedom av moderprogrammet. Att kunna identifiera de variabler som kan bli aktuella för senare analysering kan vara något av ett detektivjobb. Men det är också viktigt att förstå hur själva programmet är uppbyggt, de olika förutsättningarna för programmet, olika indata varianter etc. Ett sådant program presenteras här för att demonstrera ett sätt (av tusen), att visa dessa resultat. Förståelse och användning av subrutinerna är enklare att se med ett komplett program.

Programmet "DIA", tar hand om utdatan från programmet MODEX, beskrivet i avsnitt 2, och demonstrerar egenskaperna hos subrutinerna genom att visa ett antal diagram förenade med numeriska tabeller. Programmet är tänkt att arbeta som nedanstående schematiskt förenklade flödesschema visar.



En komplett listning av programmet "dia" finns angiven i appendix 2. Med de programkommentarer som har lagts in, bör det vara tillräckligt för förståelse av denna.

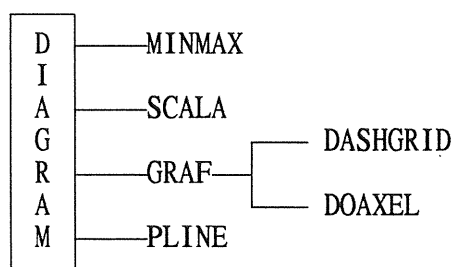
4. BESKRIVNING AV SUBRUTINERNA

4.1 INLEDNING

En kort beskrivning av alla subrutiner som har använts för att utveckla programmet "dia" kommer att visas i detta avsnitt. Huvudsubrutinen "diagram" beskrivs detaljerad.

4.2 SUBRUTIN DIAGRAM

Denna subrutin tar emot alla nödvändiga parametrar för att framställa ett diagram på skärmen.



Det enda som måste göras innan man kallar denna subrutin är att sätta skärmen i grafisk mode. Graphorias Graphics Library använder en subrutin som kallas plots. I denna framställning görs detta genom att göra "call gmode()".

4.2 Formella parametrar

title : ange diagrammets titel eller rubrik

titleinf : ange ett sifvertillägg till title(t.ex nod nr)

xlabel : x-axels rubrik

ylabel : y-axels rubrik

x1,y1 : nedersta vänstra hörnet av diagrammets fönster

x2,y2 : översta högra hörnet av diagrammets fönster

Observera att skärmens storlek är

definierad av 11.0 x 8.5 enheter i x, resp y

riktning. Nedersta vänstra hörnet är (0.0,0.0)

Översta högra hörnet är (11.0,8.5)

isegx : antal lika segment att dela x-axel
 isegy : som ovan men y-axel
 isubx : antal lika subsegment att dela varje segment i x-axel
 isuby : som ovan men y-axel
 x : vektor som innehåller x-värden att plotta
 y : som ovan men y-värdena
 istat : 0 indikerar att skalning behöver göras
 1 indikerar att man vill superponera
 grafen på den föregående i samma skala.
 npts : antal värden som innehåller x, eller y vektorn
 ichar : plottar en symbol på kurvan. De olika symbolerna
 är definierade i Graphorias Library
 manual på sidan 134. Nedan följer några
 exempel
 ichar = 0 ==> fyrkant
 1 ==> cirkel
 2 ==> triangel
 3 ==> kors
 inc : ange inkrementet för ichar ex. inc=5 betyder att
 symbolen skall plotta var femte värde.
 ndecx : antal decimaler för siffrorna i x-axel
 ndecy : som ovan men y-axel
 itypx : detta värde bestämmer hur skalning för x-värdena
 skall göras
 itypx = 0 anger att skalning skall göras
 från 0 till max- eller minimi-värdet.
 itypx = 1 anger att skalning skall göras
 från max- till minimi-värdet.
 itypy : som ovan men för y-axeln.
 icolor : färgen på kurvan
 välj 1 <= icolor <= 7 om man skall
 göra hårdkopior av skärmen med Shift+PrintScreen.

4.3 SUBRUTIN MINMAX

Denna subrutin beräknar max och min värde av en vektor

4.4 SUBRUTIN SCALA

Denna subrutin beräknar skalan, antal segment, delning av varje segment, och tiopotensen för subrutinen graph.

4.5 SUBRUTIN GRAPH

Subrutinen ritar hela utseendet och beteckningarna för diagrammet, själva kurvan plottas med en annan subrutin.

4.5 SUBRUTIN DASHGRID

Ritar en sträckad grid.

4.6 SUBRUTIN DOAXEL

Ritar en axel och indelar den i små tick.

4.7 SUBRUTIN PLINE

Plottar en linje genom ett antal punkter.

4.8 SUBRUTIN TABELL

Subrutinen skriver en tabell på skärmen med scrollfunktioner. Här anpassad för att visas tre kolumner per rad. Med några ändringar kan denna subrutin göras generell.

4.9 SUBRUTIN TABELSTD

Subrutinen skriver en tabell på skärmen med scroll funktioner. Här anpassad för max-tension, min-tension, huvud-tension och standardavvikelse.

4.10 SUBROUTIN TABELPO1

Subrutinen skriver en tabell på skärmen med scroll funktioner. Här anpassad för att visa kabelposition för tiden t.

4.11 SUBROUTIN TABELPOS

Subrutinen skriver en tabell på skärmen med scrollfunktioner. Här anpassad för kabelenvelopen.

4.12 SUBROUTIN TEXTSTRING

Denna subrutin hämtar tabellens rubrik och underrubrik.

4.13 SUBROUTIN REFILE

Subrutinen läser de filer som skall plottas. De formella parametrarnas namn överensstämmer med de i programmet MODEX.

4.14 SUBROUTIN WRFILE

Denna subrutin sparar variablerna från programmet MODEX i externa filer för senare bearbetning.

4.15 SUBROUTIN GMODE

Subrutinen sätter graphics mode

4.16 SUBROUTIN CLEAR

Subrutinen rensar den grafiska skärmen

4.17 SUBROUTIN NMODE

Subrutinen sätter skärmen i normal mode

5. EXEMPEL

I Appendix 1 visas exempel på hårdkopior av skärmen. För att producera denna, krävs i den här versionen, att DOS programmet GRAPHICS.COM finns resident i minnet. Detta åstadkoms genom att skriva Graphics innan man startar applikationsprogrammet.

Genom att trycka på tangenterna shift och print screen samtidigt kan man få en kopia av skärmen direkt på printern när så önskas. Det finns möjligheter att förbättra utseendet på hårdkopian om man skriver en förbättrad version av graphics.-com eller om man använder de inbyggda subrutinerna i Graphorias Graphics Library genom att använda rätt printer driver eller plot driver.

6. FÖRSLAG TILL FORTSATT ARBETE

En genomgående förbättring av subrutinerna är möjlig genom att i många fall göra dessa mer tillämpningsinriktade, dessa kan vara lättare att använda då men det krävs mer kännedom om hur dessa är utbyggda.

Vidare kan man utarbeta dem så att användning av en plotter eller en laser printer är möjlig för att få papperskopior av skärmdiagrammen.

Ett mål man kan sätta i att bearbeta flera datorprogram så att man kan fånga in de olika svagheter som eventuellt kan finnas i subrutinerna och fullända dessa så att studium av utdata kan bli mer attraktivt och överskådligt.

REFERENSER

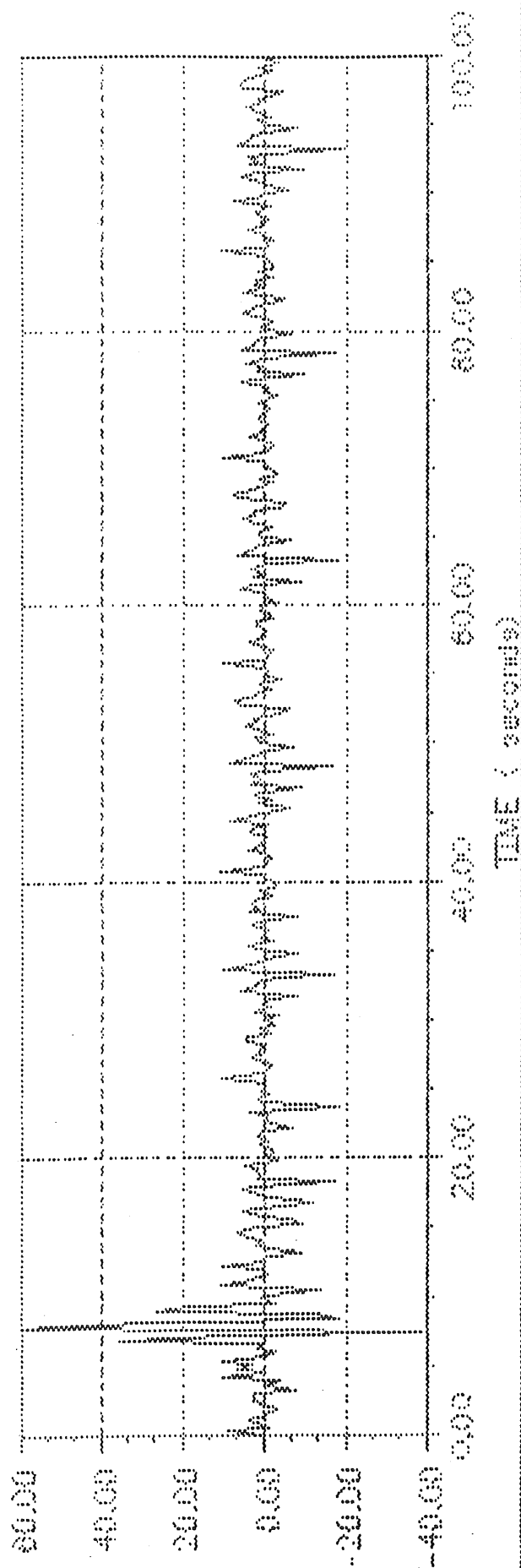
- [1] Jan Lindahl, Lars Bergdahl : " MODEX - MODIM Users Manual", 1987.
Report Series B:49. Department of Hydraulics, Chalmers
University of Technology.
- [2] Jan Lindahl, Anders Sjöberg : " Dynamic Analysis of Mooring Cables",
1983. Report Series A:9. Department of Hydraulics, Chalmers University
of Technology.

- [3] Jan Lindahl " Implicit numerisk lösning av rörelseekvationerna för en förankringskabel", 1984. Department of Hydraulics, Chalmers University of Technology.
- [4] Laheys Fortran 77 (F77L), manualer.
- [5] Graphorias Graphics Library, manual.

APPENDIX 1

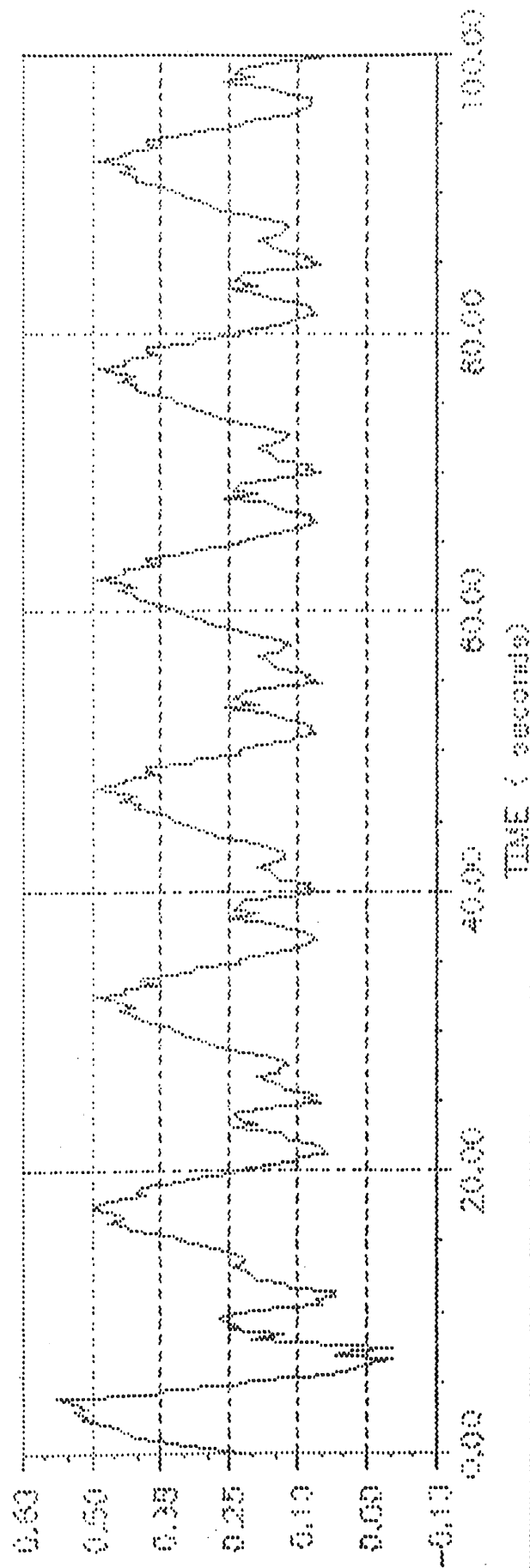
Följande sidor visar olika hårdkopior från skärmen som visar utseende av de grafer som programmet "DIA" producerar.

HORIZONTAL ACCELERATION OF NODE 13



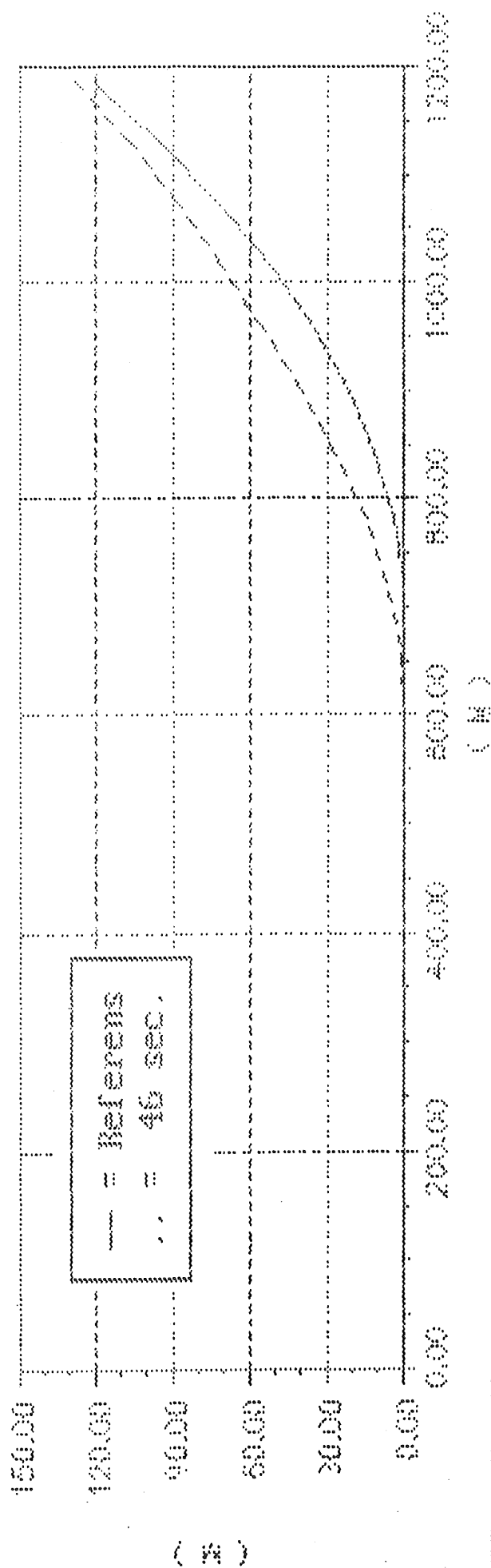
Time	Accel.	Time	Accel.	Time	Accel.
0.00	0.150E+00	3.00	0.216E+01	6.00	-0.266E+01
0.30	0.091E+01	3.30	-0.750E+01	6.30	0.200E+01
0.60	-0.164E+01	3.60	-0.344E+00	6.60	-0.164E+00
0.90	0.172E+01	3.90	-0.392E+01	6.90	0.354E+02
1.20	0.373E+01	4.20	0.102E+02	7.20	0.702E+01
1.50	-0.101E+01	4.50	-0.179E+00	7.50	-0.309E+02
1.80	-0.360E+01	4.80	0.626E+01	7.80	0.594E+02
2.10	-0.057E+00	5.10	0.310E+01	8.10	0.525E+02
2.40	0.229E+01	5.40	0.104E+02	8.40	-0.179E+02
2.70	0.214E+01	5.70	-0.760E+00	8.70	-0.126E+02

STRAIN OF ELEMENT 10



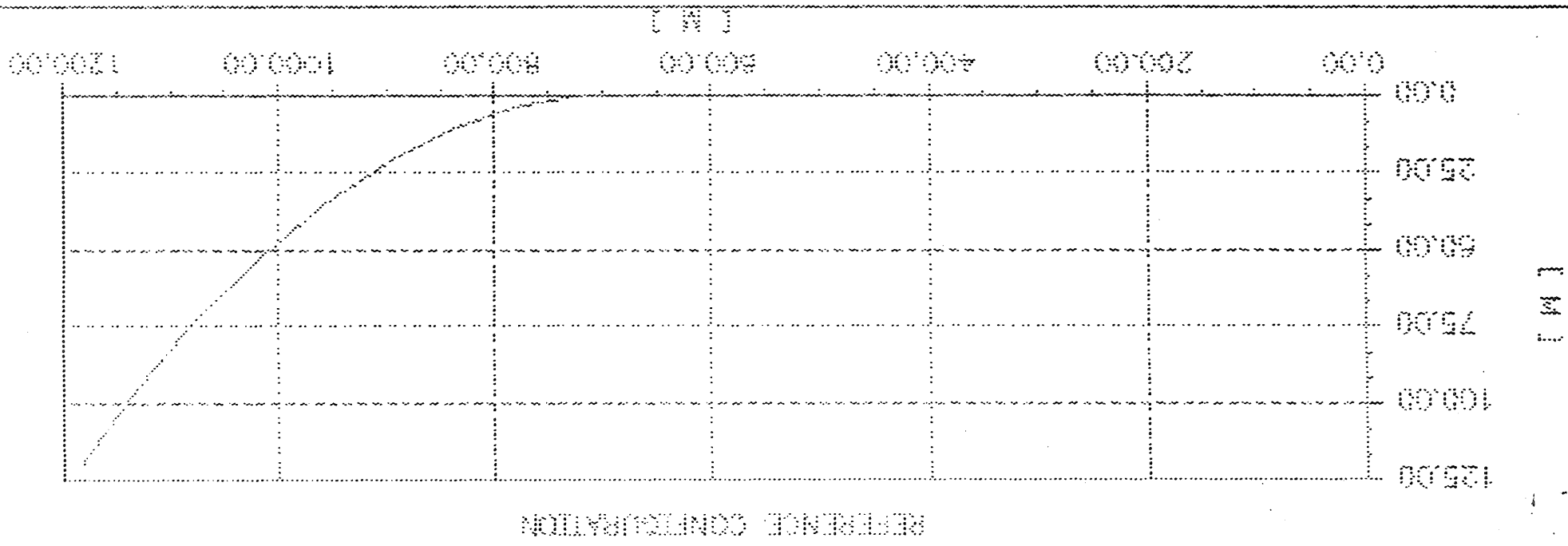
Time	Strain	Time	Strain	Time	Strain
0.00	0.220E-02	3.00	0.530E-02	6.00	0.100E-03
0.30	0.290E-02	3.30	0.510E-02	6.30	0.000E+00
0.60	0.330E-02	3.60	0.530E-02	6.60	-0.500E-03
0.90	0.380E-02	3.90	0.560E-02	6.90	0.600E-03
1.20	0.420E-02	4.20	0.380E-02	7.20	0.200E-03
1.50	0.440E-02	4.50	0.370E-02	7.50	-0.500E-03
1.80	0.450E-02	4.80	0.240E-02	7.80	0.130E-02
2.10	0.480E-02	5.10	0.210E-02	8.10	0.210E-02
2.40	0.520E-02	5.40	0.110E-02	8.40	0.150E-02
2.70	0.510E-02	5.70	0.400E-03	8.70	0.250E-02

CABLE POSITION AT TIME (seconds) 46

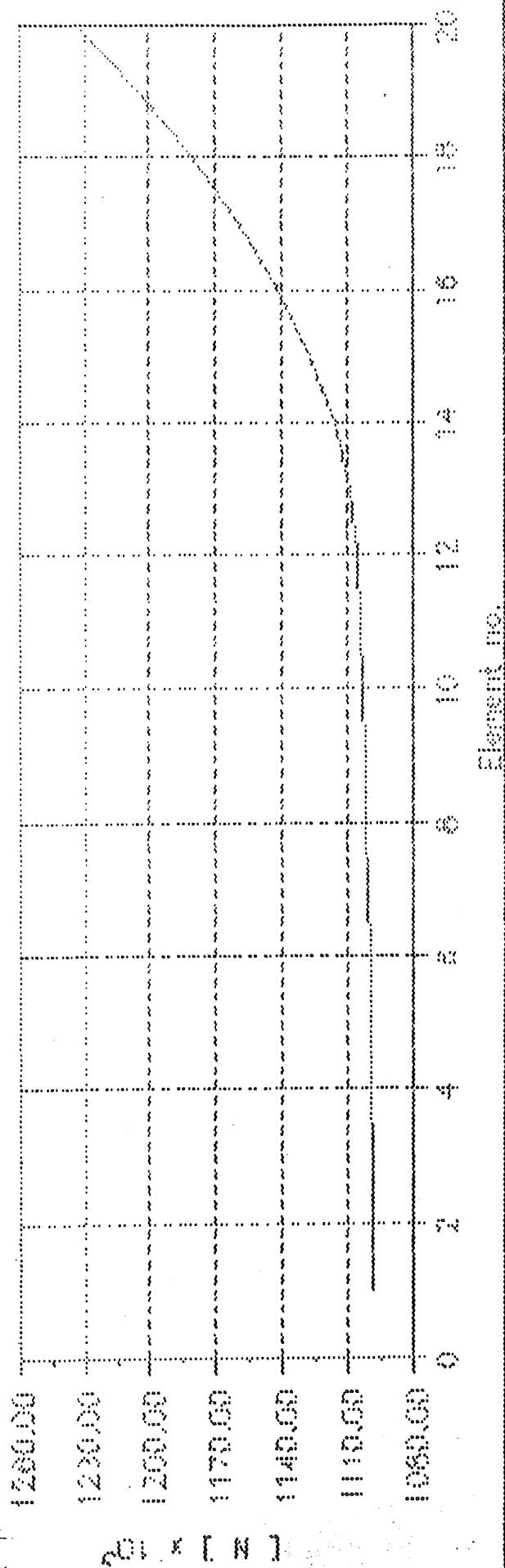


Node	Ref-xcoord	Ref-ycoord	Pos-xcoord	Pos-ycoord
1	0.000	0.000	0.000	0.000
2	60.131	-0.097	60.190	-0.097
3	120.263	-0.100	120.392	-0.100
4	180.394	-0.100	180.579	-0.100
5	240.526	-0.100	240.790	-0.100
6	300.657	-0.100	300.993	-0.100
7	360.789	-0.100	361.212	-0.100
8	420.920	-0.100	421.440	-0.017
9	481.052	-0.100	481.677	0.016
10	541.184	-0.100	541.919	-0.053

Node	Ref-xcoord	Ref-ycoord	Node	Ref-xcoord	Ref-ycoord	Node	Ref-xcoord	Ref-ycoord
1	0.000E+00	0.000E+00	11	0.601E+03	-0.998E-01	20	0.113E+04	0.346E+02
2	0.601E+02	-0.974E-01	12	0.661E+03	-0.968E-01			
3	0.120E+03	-0.998E-01	13	0.722E+03	-0.909E-01			
4	0.180E+03	-0.999E-01	14	0.782E+03	-0.877E-01			
5	0.241E+03	-0.999E-01	15	0.841E+03	-0.841E+03			
6	0.301E+03	-0.999E-01	16	0.900E+03	-0.791E+02			
7	0.361E+03	-0.999E-01	17	0.959E+03	-0.753E+02			
8	0.421E+03	-0.999E-01	18	0.102E+04	-0.521E+02			
9	0.481E+03	-0.999E-01	19	0.107E+04	0.719E+02			
10	0.541E+03	-0.999E-01	20	0.113E+04	0.346E+02			

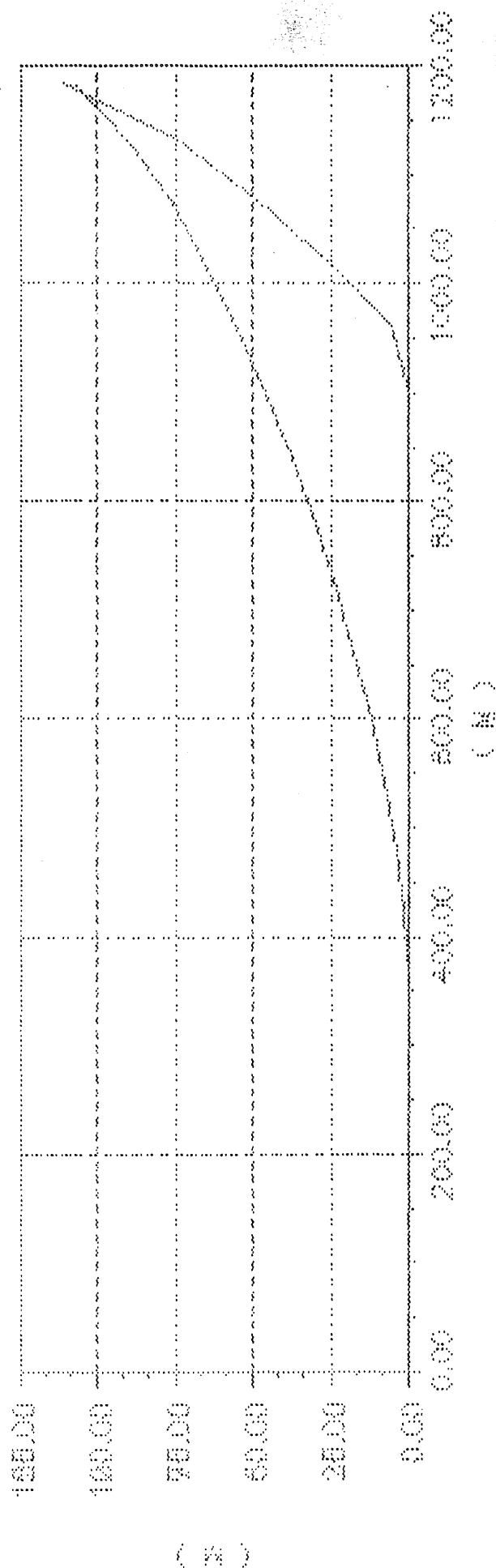


CABLE TENSION



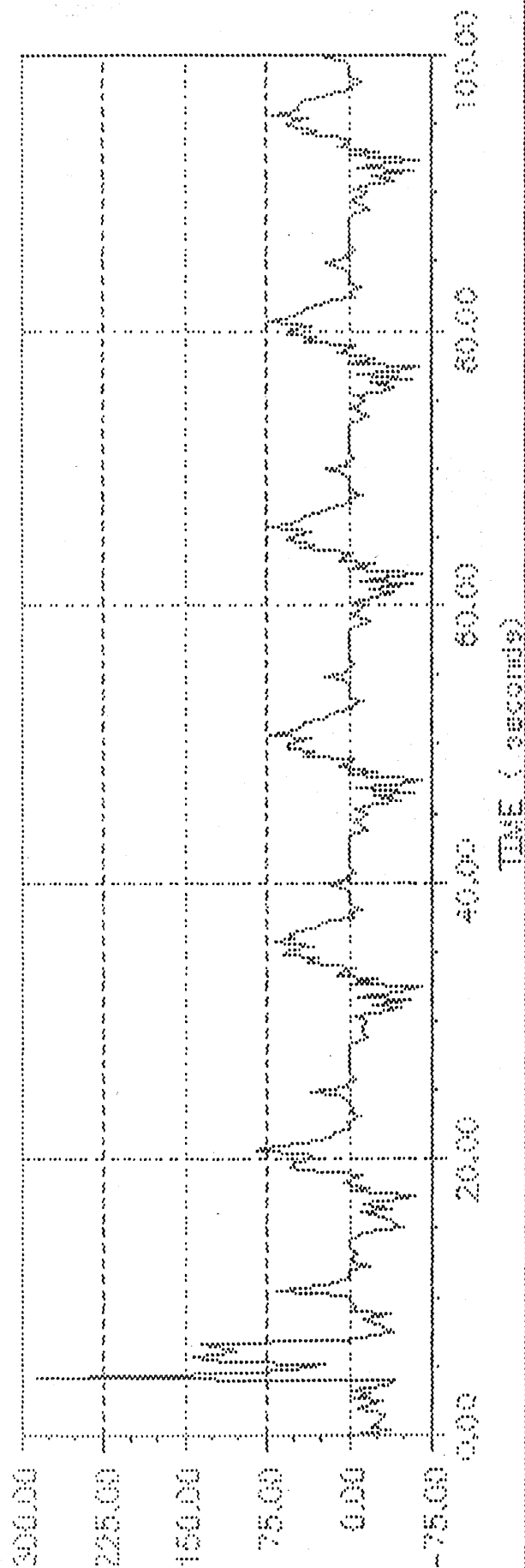
Node	Tension	Node	Tension	Node	Tension
1	0.11E+07	11	0.11E+07	21	0.12E+07
2	0.11E+07	12	0.11E+07	22	0.12E+07
3	0.11E+07	13	0.11E+07	23	0.12E+07
4	0.11E+07	14	0.11E+07	24	0.12E+07
5	0.11E+07	15	0.11E+07	25	0.12E+07
6	0.11E+07	16	0.11E+07	26	0.12E+07
7	0.11E+07	17	0.12E+07	27	0.12E+07
8	0.11E+07	18	0.12E+07	28	0.12E+07
9	0.11E+07	19	0.12E+07	29	0.12E+07
10	0.11E+07	20	0.12E+07	30	0.12E+07

CABLE ENVELOPE



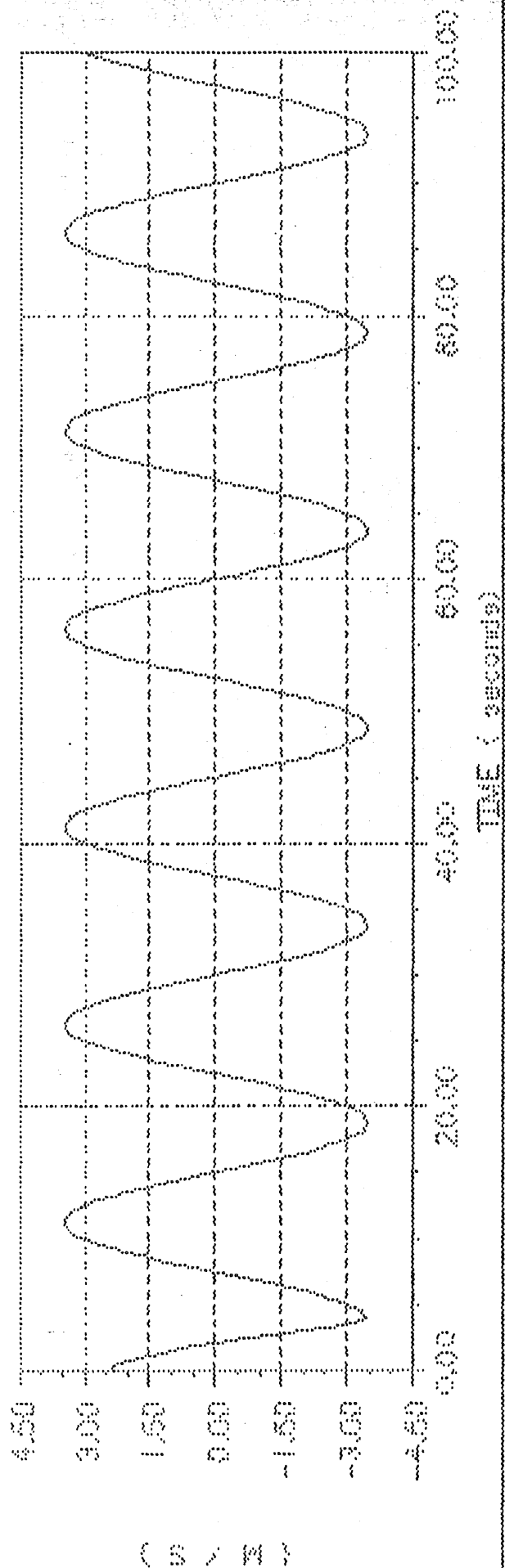
Node	Ref-xcoord	Ref-ycoord	Max-xcoord	Max-ycoord	Min-xcoord	Min-ycoord
1	0.000	0.000	0.000	0.000	0.000	0.000
2	60.131	-0.097	60.000	0.000	60.110	-0.097
3	120.263	-0.100	120.000	0.000	120.236	-0.100
4	180.394	-0.100	180.000	0.000	180.353	-0.100
5	240.526	-0.100	240.000	0.000	240.467	-0.100
6	300.657	-0.100	300.000	0.000	300.577	-0.100
7	360.789	-0.100	360.000	0.000	360.691	-0.100
8	420.920	-0.100	421.106	1.673	421.797	-0.100
9	481.052	-0.100	481.107	4.772	481.912	-0.100
10	541.184	-0.100	541.298	9.151	542.244	-0.114

TANGENTIAL DRAG FORCE DIRECTION 1 OF NODE 13



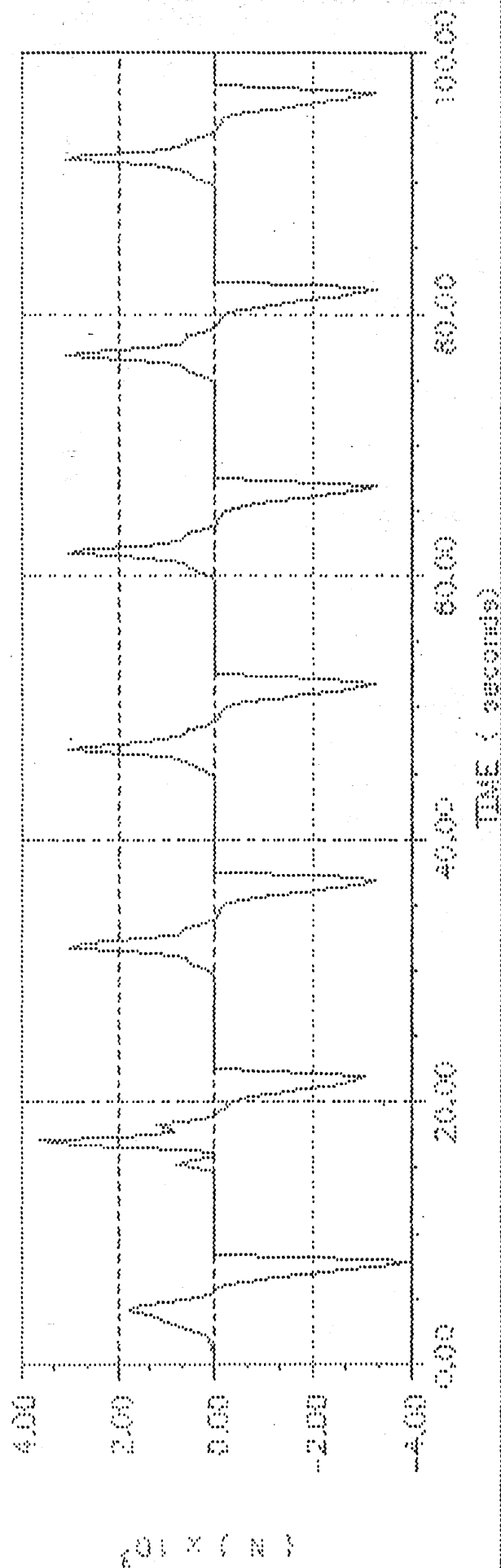
Time	Tang. Force	Time	Tang. Force	Time	Tang. Force
0.00	0.000E+00	3.00	-0.209E+02	6.00	0.102E+03
0.30	-0.372E+02	3.30	-0.439E+00	6.30	0.111E+03
0.60	-0.189E+02	3.60	-0.337E+02	6.60	0.136E+03
0.90	-0.259E+02	3.90	-0.416E+02	6.90	0.407E+01
1.20	-0.293E+02	4.20	0.284E+03	7.20	-0.229E+02
1.50	-0.313E+02	4.50	0.908E+02	7.50	-0.399E+02
1.80	-0.729E+01	4.80	0.495E+02	7.80	-0.298E+02
2.10	-0.143E+02	5.10	0.206E+02	8.10	-0.256E+02
2.40	-0.302E+02	5.40	0.112E+03	8.40	-0.131E+02
2.70	-0.596E+01	5.70	0.142E+03	8.70	-0.373E+02

VERTICAL VELOCITY OF NODE 21



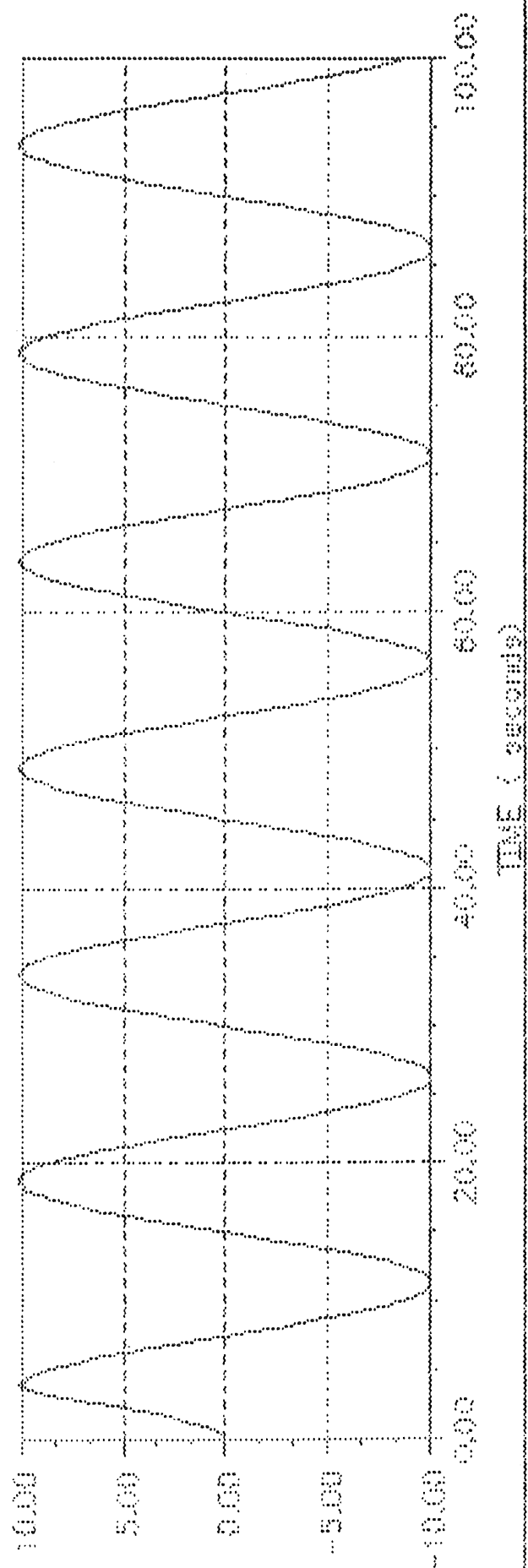
Time	Veloc.	Time	Veloc.	Time	Veloc.
0.00	0.227E+01	3.00	-0.196E+01	6.00	-0.206E+01
0.30	0.234E+01	3.30	-0.257E+01	6.30	-0.157E+01
0.60	0.214E+01	3.60	-0.319E+01	6.60	-0.128E+01
0.90	0.178E+01	3.90	-0.347E+01	6.90	-0.866E+00
1.20	0.143E+01	4.20	-0.340E+01	7.20	-0.433E+00
1.50	0.101E+01	4.50	-0.333E+01	7.50	0.000E+00
1.80	0.496E+00	4.80	-0.313E+01	7.80	0.433E+00
2.10	-0.673E-01	5.10	-0.293E+01	8.10	0.866E+00
2.40	-0.671E+00	5.40	-0.268E+01	8.40	0.128E+01
2.70	-0.130E+01	5.70	-0.237E+01	8.70	0.157E+01

2007-2008
2008-2009
2009-2010
2010-2011
2011-2012
2012-2013
2013-2014
2014-2015
2015-2016
2016-2017
2017-2018
2018-2019
2019-2020
2020-2021
2021-2022
2022-2023
2023-2024
2024-2025
2025-2026
2026-2027
2027-2028
2028-2029
2029-2030
2030-2031
2031-2032
2032-2033
2033-2034
2034-2035
2035-2036
2036-2037
2037-2038
2038-2039
2039-2040
2040-2041
2041-2042
2042-2043
2043-2044
2044-2045
2045-2046
2046-2047
2047-2048
2048-2049
2049-2050
2050-2051
2051-2052
2052-2053
2053-2054
2054-2055
2055-2056
2056-2057
2057-2058
2058-2059
2059-2060
2060-2061
2061-2062
2062-2063
2063-2064
2064-2065
2065-2066
2066-2067
2067-2068
2068-2069
2069-2070
2070-2071
2071-2072
2072-2073
2073-2074
2074-2075
2075-2076
2076-2077
2077-2078
2078-2079
2079-2080
2080-2081
2081-2082
2082-2083
2083-2084
2084-2085
2085-2086
2086-2087
2087-2088
2088-2089
2089-2090
2090-2091
2091-2092
2092-2093
2093-2094
2094-2095
2095-2096
2096-2097
2097-2098
2098-2099
2099-2100
2100-2101
2101-2102
2102-2103
2103-2104
2104-2105
2105-2106
2106-2107
2107-2108
2108-2109
2109-2110
2110-2111
2111-2112
2112-2113
2113-2114
2114-2115
2115-2116
2116-2117
2117-2118
2118-2119
2119-2120
2120-2121
2121-2122
2122-2123
2123-2124
2124-2125
2125-2126
2126-2127
2127-2128
2128-2129
2129-2130
2130-2131
2131-2132
2132-2133
2133-2134
2134-2135
2135-2136
2136-2137
2137-2138
2138-2139
2139-2140
2140-2141
2141-2142
2142-2143
2143-2144
2144-2145
2145-2146
2146-2147
2147-2148
2148-2149
2149-2150
2150-2151
2151-2152
2152-2153
2153-2154
2154-2155
2155-2156
2156-2157
2157-2158
2158-2159
2159-2160
2160-2161
2161-2162
2162-2163
2163-2164
2164-2165
2165-2166
2166-2167
2167-2168
2168-2169
2169-2170
2170-2171
2171-2172
2172-2173
2173-2174
2174-2175
2175-2176
2176-2177
2177-2178
2178-2179
2179-2180
2180-2181
2181-2182
2182-2183
2183-2184
2184-2185
2185-2186
2186-2187
2187-2188
2188-2189
2189-2190
2190-2191
2191-2192
2192-2193
2193-2194
2194-2195
2195-2196
2196-2197
2197-2198
2198-2199
2199-2200
2200-2201
2201-2202
2202-2203
2203-2204
2204-2205
2205-2206
2206-2207
2207-2208
2208-2209
2209-2210
2210-2211
2211-2212
2212-2213
2213-2214
2214-2215
2215-2216
2216-2217
2217-2218
2218-2219
2219-2220
2220-2221
2221-2222
2222-2223
2223-2224
2224-2225
2225-2226
2226-2227
2227-2228
2228-2229
2229-2230
2230-2231
2231-2232
2232-2233
2233-2234
2234-2235
2235-2236
2236-2237
2237-2238
2238-2239
2239-2240
2240-2241
2241-2242
2242-2243
2243-2244
2244-2245
2245-2246
2246-2247
2247-2248
2248-2249
2249-2250
2250-2251
2251-2252
2252-2253
2253-2254
2254-2255
2255-2256
2256-2257
2257-2258
2258-2259
2259-2260
2260-2261
2261-2262
2262-2263
2263-2264
2264-2265
2265-2266
2266-2267
2267-2268
2268-2269
2269-2270
2270-2271
2271-2272
2272-2273
2273-2274
2274-2275
2275-2276
2276-2277
2277-2278
2278-2279
2279-2280
2280-2281
2281-2282
2282-2283
2283-2284
2284-2285
2285-2286
2286-2287
2287-2288
2288-2289
2289-2290
2290-2291
2291-2292
2292-2293
2293-2294
2294-2295
2295-2296
2296-2297
2297-2298
2298-2299
2299-2300
2300-2301
2301-2302
2302-2303
2303-2304
2304-2305
2305-2306
2306-2307
2307-2308
2308-2309
2309-2310
2310-2311
2311-2312
2312-2313
2313-2314
2314-2315
2315-2316
2316-2317
2317-2318
2318-2319
2319-2320
2320-2321
2321-2322
2322-2323
2323-2324
2324-2325
2325-2326
2326-2327
2327-2328
2328-2329
2329-2330
2330-2331
2331-2332
2332-2333
2333-2334
2334-2335
2335-2336
2336-2337
2337-2338
2338-2339
2339-2340
2340-2341
2341-2342
2342-2343
2343-2344
2344-2345
2345-2346
2346-2347
2347-2348
2348-2349
2349-2350
2350-2351
2351-2352
2352-2353
2353-2354
2354-2355
2355-2356
2356-2357
2357-2358
2358-2359
2359-2360
2360-2361
2361-2362
2362-2363
2363-2364
2364-2365
2365-2366
2366-2367
2367-2368
2368-2369
2369-2370
2370-2371
2371-2372
2372-2373
2373-2374
2374-2375
2375-2376
2376-2377
2377-2378
2378-2379
23



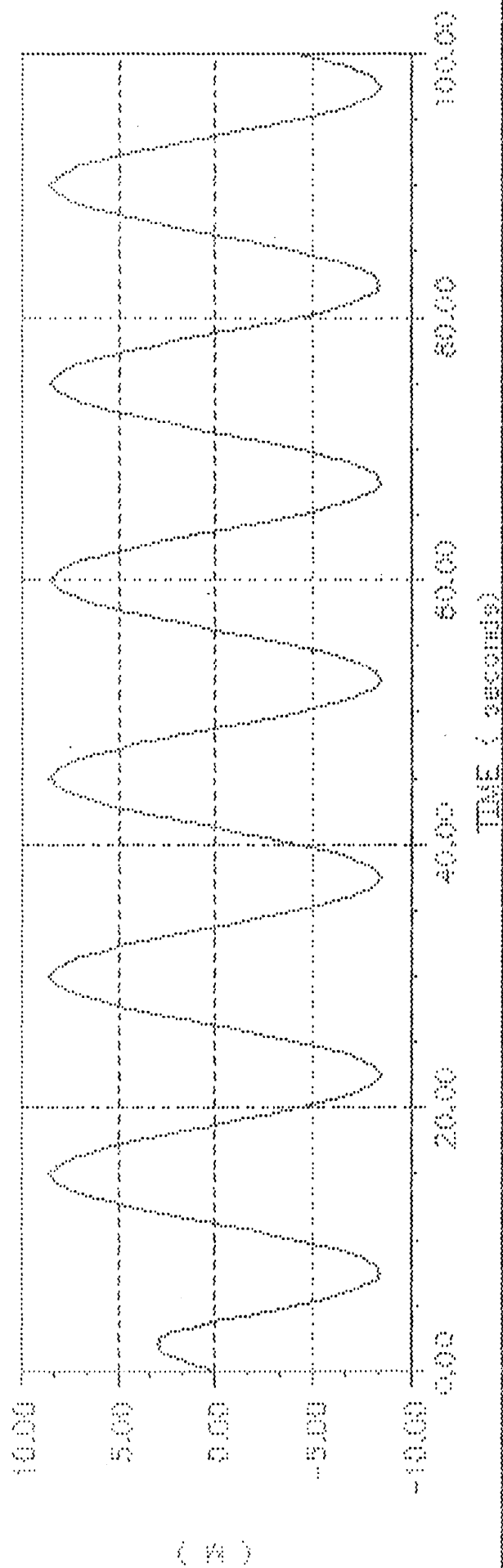
Time	Tran. Force	Time	Tran. Force	Time	Tran. Force
0.00	0.000E+00	3.00	0.054E+03	6.00	-0.223E+03
0.30	0.420E-02	3.30	0.104E+04	6.30	-0.052E+03
0.60	0.341E+01	3.60	0.121E+04	6.60	-0.124E+04
0.90	0.101E+02	3.90	0.163E+04	6.90	-0.192E+04
1.20	0.459E+02	4.20	0.179E+04	7.20	-0.262E+04
1.50	0.791E+02	4.50	0.149E+04	7.50	-0.333E+04
1.80	0.110E+03	4.80	0.944E+03	7.80	-0.397E+04
2.10	0.168E+03	5.10	0.379E+03	8.10	-0.330E+04
2.40	0.333E+03	5.40	0.599E+02	8.40	-0.210E-02
2.70	0.602E+03	5.70	-0.141E+02	8.70	-0.700E-03

HORIZONTAL DISPLACEMENT OF NODE 21



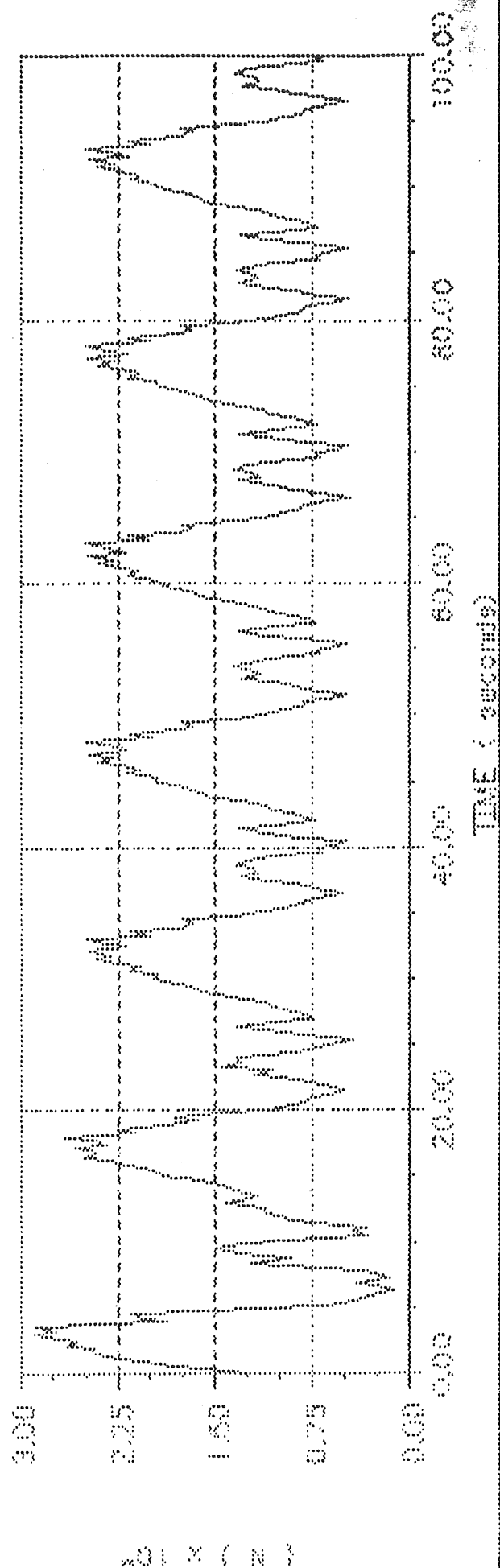
Time	Displace.	Time	Displace.
0.00	0.000E+00	3.00	0.773E+01
0.30	0.101E+00	3.30	0.868E+01
0.60	0.402E+00	3.60	0.966E+01
0.90	0.889E+00	3.90	0.101E+02
1.20	0.155E+01	4.20	0.986E+01
1.50	0.239E+01	4.50	0.966E+01
1.80	0.330E+01	4.80	0.909E+01
2.10	0.435E+01	5.10	0.851E+01
2.40	0.545E+01	5.40	0.777E+01
2.70	0.654E+01	5.70	0.687E+01
		6.00	0.597E+01
		6.30	0.484E+01
		6.60	0.371E+01
		6.90	0.251E+01
		7.20	0.126E+01
		7.50	0.000E+00
		7.80	-0.126E+01
		8.10	-0.251E+01
		8.40	-0.371E+01
		8.70	-0.484E+01

VERTICAL DISPLACEMENT OF NODE 21



Time	Displacement	Time	Displacement
0.00	0.000E+00	3.00	0.210E+01
0.30	0.667E+00	3.30	0.139E+01
0.60	0.131E+01	3.60	0.504E+00
0.90	0.100E+01	3.90	-0.525E+00
1.20	0.236E+01	4.20	-0.150E+01
1.50	0.275E+01	4.50	-0.263E+01
1.80	0.294E+01	4.80	-0.357E+01
2.10	0.301E+01	5.10	-0.452E+01
2.40	0.289E+01	5.40	-0.537E+01
2.70	0.257E+01	5.70	-0.612E+01
		6.00	-0.680E+01
		6.30	-0.736E+01
		6.60	-0.784E+01
		6.90	-0.817E+01
		7.20	-0.833E+01
		7.50	-0.850E+01
		7.80	-0.833E+01
		8.10	-0.817E+01
		8.40	-0.784E+01
		8.70	-0.736E+01

TENSION OF ELEMENT 20



Time	Tension	Time	Tension
0.00	0.123E+07	6.00	0.232E+06
0.30	0.162E+07	6.30	0.114E+06
0.60	0.181E+07	6.60	0.245E+06
0.90	0.207E+07	6.90	0.410E+06
1.20	0.226E+07	7.20	0.148E+06
1.50	0.241E+07	7.50	0.201E+06
1.80	0.244E+07	7.80	0.340E+06
2.10	0.263E+07	8.10	0.687E+06
2.40	0.258E+07	8.40	0.121E+07
2.70	0.269E+07	8.70	0.903E+06

APPENDIX 2

Följande sidor visar listningen av programmet DIA.FOR.

```

*-----*
*      Program dia.for    version 5/11/91    av Francisco Herrera      *
*-----*
*      Denna program demonstrerar hur utdata av programmet Modex      *
*      kan presenteras.                                              *
*-----*
*
character*80 filename
read(*,*)filename
call dia(filename)
stop
end

*-----*
subroutine dia(filename)
dimension iad(10,2),iav(5,2),iaa(5,2),iat(5),iae(5),idt(5,2),
+       idn(5,2),tpl(1000),
+       ad(10,1000),av(5,1000),aa(5,1000),at(5,1000),ae(5,1000),
+       adt(5,1000),adn(5,1000),
+       igem(20,5),tgem(20),xg1(100,20),xg2(100,20),
+       tkm(100),tm(100),tkmin(100),tmin(100),tmean(100),
+       tsd(100),xenvmax(100,3),xenvmin(100,3)
dimension x(1000),y(1000),rx(100),ry(100),rz(100),
+       eg(100,100,2),irikt(100)
character*80 title(5,15),titel(20),text*20,str*4,filename,name
double precision tgem
logical ja

*-----*
integer *2 sm,lm,sn,ln,sp,lp,done,one,two,three,four,five,six,
+       seven,eight,nine,zero,la,sa,lb,sb,lc,sc,ix
parameter( zero= 48,one= 49,two=50,three=51,four=52,five=53,six=54
+       ,seven=55,eight=56,nine=57,sm=77,sn=78,sp=80,lm=109,
+       ln=110,lp=112,la=97,sa=65,lb=98,sb=66,lc=99,sc=67)

*-----*
*      Läser alla filer som skall plottas, från aktuellt skivenhet.    *
*      De beteckningar som används är densamma som programmet Modex    *
*      använder. För närmare information se manualen.                  *
*-----*
call refile(ad,av,aa,at,ae,tpl,nad,kpl,nav,naa,nat,nae,
+       iad,iav,iaa,iat,iae,nadt,nadn,idt,idn,adt,adn,
+       iplot,ngem,igem,tgem,xg1,xg2,ne,nd,xenvmax,xenvmin,
+       title,rx,ry,rz,tkm,tm,tkmin,tmin,tmean,tsd,ieig,
+       filename)

*-----*
done=0
do while(done.eq.0)
call menu1(ix)
select case(ix)
case(zero)
done=1
case(one)
if(iplot.ne.0)then
call gmode
*.... Aktuella parametrar för standard avvikelse etc...
titleinf=-1 ! Ingen titelinformation.
x1=0.0      ! Nedre vänster hörn x-coord.
y1=4.5      ! Nedre vänster hörn y-coord.
x2=9.5      ! Övre höger hörn x-coord.
y2=8.5      ! Övre höger hörn y-coord.
isegx=10    ! Dela x-axel i 19 segmenter.

```

```

isegy=7      ! Dela y-axel i 7 segmenter.
isubx=0      ! Inga subsegmenter i x-axel.
isuby=2      ! Två subsegmenter i y-axel.
istat=0      ! Beräknar scalan för grafen.
ichar=0      ! Inga symboler på kurvan.
ndecx=0      ! Inga decimaler på x-axel.
ndecy=2      ! Två decimaler på y-axel.
itypx=1      ! x-axel scalas från 1 till antal element.
itypy=0      ! yaxel scalas från 0.
icolor=2     ! Färgen på kurvan grön.
npts=ne      ! Antal element
*-----*
*   Förbereder för att plotta max-tension
*-----*
    call minmax(tkm,npts,yymin,ymax)
    ymax=ymax
    call minmax(tkmin,npts,yymin,ymax)
    ymin=yymin
    xmax=npts
    xmin=1
*-----*
    do 101 i=1,npts
        x(i)=i
        y(i)=tkm(i)
101  continue
*   Plottar max-tension
    call diagram1
+ (title(1,9),titleinf,title(4,9),title(5,9),x1,y1,x2,y2,isegx,
+ isegy,isubx,isuby,x,y,istat,npts,ichar,ndecx,ndecy,
+ itypx,itypy,icolor,xmax,xmin,ymax,ymin )
*-----*
*   Förbereder för att plotta min-tension
    ichar=0
    istat=1    ! Använd samma scala och övriga aktuella parametrar.
    do 102 i=1,npts
        x(i)=i
        y(i)=tkmin(i)
102  continue
*   Plotta min-tension
    call diagram1
+ (title(1,9),titleinf,title(4,9),title(5,9),x1,y1,x2,y2,isegx,
+ isegy,isubx,isuby,x,y,istat,npts,ichar,ndecx,ndecy,
+ itypx,itypy,icolor,xmax,xmin,ymax,ymin )
*-----*
*   Förbereder för att plotta mean-tension
    ichar=1
    istat=1
    do 103 i=1,npts
        x(i)=i
        y(i)=tmean(i)
103  continue
*   Plotta mean-tension
    call diagram1
+ (title(1,9),titleinf,title(4,9),title(5,9),x1,y1,x2,y2,isegx,
+ isegy,isubx,isuby,x,y,istat,npts,ichar,ndecx,ndecy,
+ itypx,itypy,icolor,xmax,xmin,ymax,ymin )
*-----*
*   Förbereder för att plotta std-avvikelse
    ichar=3

```

```

    istat=1
    do 104 i=1,npts
        x(i)=i
        y(i)=tmean(i)+tsd(i)
104  continue
*   Plotta tmean+standardavvikelse
    call diagram1
+ (title(1,9),titleinf,title(4,9),title(5,9),x1,y1,x2,y2,isegx,
+ isegy,isubx,isuby,x,y,istat,npts,ichar,ndecx,ndecy,
+ itypx,itypy,icolor,xmax,xmin,ymax,ymin )

    ichar=3
    istat=1
    do 105 i=1,npts
        x(i)=i
        y(i)=tmean(i)-tsd(i)
105  continue
*   Plotta tmean-standardavvikelse
    call diagram1
+ (title(1,9),titleinf,title(4,9),title(5,9),x1,y1,x2,y2,isegx,
+ isegy,isubx,isuby,x,y,istat,npts,ichar,ndecx,ndecy,
+ itypx,itypy,icolor,xmax,xmin,ymax,ymin )
*-----*

    call gtext(0,70,'')
    call gtext(1,70,' — = max ')
    call gtext(2,70,' -- =+std ')
    call gtext(3,70,' .. =mean ')
    call gtext(4,70,' -- =+std ')
    call gtext(5,70,' — = min ')
    call gtext(6,70,'')
*-----*

*   Visa också plott-värdena i tabelform.
    call tabelstd(tkm,tm,tkmin,tmin,tmean,tsd,npts,iflag)
*-----*

*   Rensar det grafiska skärmen
    call clear()
    call nmode()
    end if
*-----*

    case(two)
*.... Aktuella parametrar för nod-försjukning
    if ( nad .ne.0) then
        call gmode()
        j=1
        do while(j.ge.1.and.j.le.nad)
            istat=0
            ichar=0
            x1=0.0
            y1=4.5
            x2=11.
            y2=8.5
            isegx=6
            isegy=5
            isubx=4
            isuby=3
            ndecx=2
            ndecy=2
            itypx=0
            itypy=0

```

```

        icolor=2
        titleinf=iad(j,1) ! Anger vilken nod det gäller
        n=iad(j,2)        ! Anger vilken riktning det gäller
        npts=kpl
        do 1 i=1,kpl
            x(i)= tpl(i)
            y(i)= ad(j,i)
1          continue
*      Plottar nod-förskjutning
        call diagram
+      (title(n,1),titleinf,title(4,1),title(5,1),x1,y1,x2,y2,isegx,
+      isegy,isubx,isuby,x,y,istat,npts,ichar,ndecx,ndecy,
+      itypx,itypy,icolor)
        call tabel(x,y,npts,1,iflag)
        call clear()
        if(iflag.eq.0)then
            j=100
        elseif(iflag.eq.1)then
            j=j+1
        elseif(iflag.eq.2)then
            j=j-1
        end if
        end do
        call nmode()
        end if
*-----*
*.....velocities
        case(three)
        if(nav .ne.0) then
            call gmode()
            j=1
            do while(j.ge.1.and.j.le.nav)
                istat=0
                ichar=0
                x1=0.0
                y1=4.5
                x2=11.
                y2=8.5
                isegx=6
                isegy=5
                isubx=4
                isuby=3
                ndecx=2
                ndecy=2
                itypx=0
                itypy=0
                icolor=2
                titleinf=iav(j,1)
                n=iav(j,2)
                npts=kpl
                do i=1,kpl
                    x(i)= tpl(i)
                    y(i)= av(j,i)
                end do
                call diagram
+      (title(n,2),titleinf,title(4,2),title(5,2),x1,y1,x2,y2,isegx,
+      isegy,isubx,isuby,x,y,istat,npts,ichar,ndecx,ndecy,
+      itypx,itypy,icolor)
                call tabel(x,y,npts,2,iflag)

```

```

        call clear()
        if(iflag.eq.0)then
            j=100
        elseif(iflag.eq.1)then
            j=j+1
        else if(iflag.eq.2)then
            j=j-1
        end if
    end do
    call nmode()
end if
*-----*
*.....accelerations
case(four)
if(naa .ne.0) then
    call gmode()
    j=1
    do while(j.ge.1.and.j.le.naa)
        istat=0
        ichar=0
        x1=0.0
        y1=4.5
        x2=11.
        y2=8.5
        isegx=6
        isegy=5
        isubx=4
        isuby=3
        ndecx=2
        ndecy=2
        itypx=0
        itypy=0
        icolor=2
        titleinf=iaa(j,1)
        n=iaa(j,2)
        npts=kpl
        do 3 i=1,kpl
            x(i)= tpl(i)
            y(i)= aa(j,i)
3        continue
        call diagram
        + (title(n,3),titleinf,title(4,3),title(5,3),x1,y1,x2,y2,isegx,
        + isegy,isubx,isuby,x,y,istat,npts,ichar,ndecx,ndecy,
        + itypx,itypy,icolor)
*    Plottar också tabell
        call tabel(x,y,npts,3,iflag)
        call clear()
        if(iflag.eq.0)then
            j=100
        elseif(iflag.eq.1)then
            j=j+1
        else if(iflag.eq.2)then
            j=j-1
        end if
    end do
    call nmode()
end if

```

```

*-----*
*       Resten av programmet är likartad
*-----*
*.....tension
  case(five)
  if(nat .ne.0) then
    call gmode()
    j=1
    do while(j.ge.1.and.j.le.nat)
      istat=0
      ichar=0
      x1=0.0
      y1=4.5
      x2=11.
      y2=8.5
      isegx=6
      isegy=5
      isubx=4
      isuby=3
      ndecx=2
      ndecy=2
      itypx=0
      itypy=0
      icolor=2
      titleinf=iat(j)
      npts=kpl
      do i=1,kpl
        x(i)= tpl(i)
        y(i)= at(j,i)
      end do
      call diagram
+      (title(1,4),titleinf,title(4,4),title(5,4),x1,y1,x2,y2,isegx,
+      isegy,isubx,isuby,x,y,istat,npts,ichar,ndecx,ndecy,
+      itypx,itypy,icolor)
      call tabel(x,y,npts,4,iflag)
      call clear()
      if(iflag.eq.0)then
        j=100
      elseif(iflag.eq.1)then
        j=j+1
      else if(iflag.eq.2)then
        j=j-1
      end if
    end do
    call nmode()
  end if
*-----*
*.....strain
  case (six)
  if(nae .ne.0) then
    call gmode()
    j=1
    do while(j.ge.1.and.j.le.nae)
      istat=0
      ichar=0
      x1=0.0
      y1=4.5
      x2=11.
      y2=8.5

```

```

    isegx=6
    isegy=5
    isubx=4
    isuby=3
    ndecx=2
    ndecy=2
    itypx=0
    itypy=0
    icolor=2
    titleinf=iae(j)
    npts=kpl
    do i=1,kpl
        x(i)= tpl(i)
        y(i)= ae(j,i)
    end do
    call diagram
+   (title(1,5),titleinf,title(4,5),title(5,5),x1,y1,x2,y2,isegx,
+   isegy,isubx,isuby,x,y,istat,npts,ichar,ndecx,ndecy,
+   itypx,itypy,icolor)
    call tabel(x,y,npts,5,iflag)
    call clear()
    if(iflag.eq.0)then
        j=100
    elseif(iflag.eq.1)then
        j=j+1
    else if(iflag.eq.2)then
        j=j-1
    end if
end do
call nmode
end if

```

```

*.....tangforce
case(seven)
if(nadt .ne.0) then
    call gmode()
    j=1
    do while(j.ge.1.and.j.le.nadt)
        istat=0
        ichar=0
        x1=0.0
        y1=4.5
        x2=11.
        y2=8.5
        isegx=6
        isegy=5
        isubx=4
        isuby=3
        ndecx=2
        ndecy=2
        itypx=0
        itypy=0
        icolor=2
        titleinf=idt(j,1)
        n=idt(j,2)
        npts=kpl
        do 6 i=1,kpl
            x(i)= tpl(i)
            y(i)= adt(j,i)

```

```

6      continue
      call diagram
+     (title(n,6),titleinf,title(4,6),title(5,6),x1,y1,x2,y2,isegx,
+     isegy,isubx,isuby,x,y,istat,npts,ichar,ndecx,ndecy,
+     itypx,itypy,icolor)
      call tabel(x,y,npts,6,iflag)
      call clear()
      if(iflag.eq.0)then
         j=100
      elseif(iflag.eq.1)then
         j=j+1
      else if(iflag.eq.2)then
         j=j-1
      end if
    end do
    call nmode()
  end if
*-----*
*.....tranforce
case(eight)
if(nadn .ne.0) then
  call gmode()
  j=1
  do while(j.ge.1.and.j.le.nadn)
    istat=0
    ichar=0
    x1=0.0
    y1=4.5
    x2=11.
    y2=8.5
    isegx=6
    isegy=5
    isubx=4
    isuby=3
    ndecx=2
    ndecy=2
    itypx=0
    itypy=0
    icolor=2
    titleinf=idn(j,1)
    n=idn(j,2)
    npts=kpl
    do 7 i=1,kpl
      x(i)= tpl(i)
      y(i)= adn(j,i)
7    continue
    call diagram
+   (title(n,7),titleinf,title(4,7),title(5,7),x1,y1,x2,y2,isegx,
+   isegy,isubx,isuby,x,y,istat,npts,ichar,ndecx,ndecy,
+   itypx,itypy,icolor)
    call tabel(x,y,npts,7,iflag)
    call clear()
    if(iflag.eq.0)then
      j=100
    elseif(iflag.eq.1)then
      j=j+1
    else if(iflag.eq.2)then
      j=j-1
    end if

```

```

        end do
        call nmode()
        end if
*-----*
*..... cable position
        case(nine)
1000    if ( ngem.ne.0 ) then
            call gmode()
            j=1
            do while(j.ge.1.and.j.le.ngem)
                istat=0
                ichar=1
                x1=0.0
                y1=4.5
                x2=11.
                y2=8.5
                isegx=6
                isegy=5
                isubx=4
                isuby=3
                ndecx=2
                ndecy=2
                itypx=0
                itypy=0
                icolor=2
                titleinf=tgem(j)
                write(str,'(i4)')titleinf
                k1=igem(j,3)
                k2=igem(j,4)
                npts=k2-k1+1
                do 8 i=k1,k2
                    x(i)= xg1(i,j)
                    y(i)= xg2(i,j)
8                continue
                call diagram
                + (title(1,8),titleinf,title(4,8),title(5,8),x1,y1,x2,y2,isegx,
                + isegy,isubx,isuby,x,y,istat,npts,ichar,ndecx,ndecy,
                + itypx,itypy,icolor)
* Referens
                istat=1
                ichar=0
                npts=ne+1
                x1=0.0
                y1=4.5
                x2=11.
                y2=8.5
                isegx=6
                isegy=5
                isubx=4
                isuby=3
                ndecx=2
                ndecy=2
                itypx=0
                itypy=0
                icolor=3
                titleinf=0.0
                call diagram
                + (title(1,10),titleinf,title(4,10),title(5,10),x1,y1,x2,y2,

```

```

+       isegx, isegy, isubx, isuby, rx, ry, istat, npts, ichar, ndecx,
+       ndecy, itypx, itypy, icolor)
text='| .. ='//str//' sec. |'
call gtext(3,15,'|')
call gtext(4,15,'| — = Referens |')
call gtext(5,15,text(1:17))
call gtext(6,15,'|')
call tabelpol(rx,ry,x,y,npts,iflag)
call clear()
if(iflag.eq.0)then
    j=100
elseif(iflag.eq.1)then
    j=j+1
else if(iflag.eq.2)then
    j=j-1
end if
end do
call nmode()
end if
*-----
*----- Cable envelope.
case(la,sa)
    if(ngem.ne.0) then
        call gmode()
        istat=0
        ichar=0
        x1=0.0
        y1=4.5
        x2=11.
        y2=8.5
        isegx=6
        isegy=5
        isubx=4
        isuby=3
        ndecx=2
        ndecy=2
        itypx=0
        itypy=0
        icolor=3
        titleinf=-1
        npts=ne+1
        do 177 i=1,npts
            x(i)=xenvmax(i,1)
            y(i)=xenvmax(i,2)
177        continue
        call diagram
        + (title(1,11),titleinf,title(4,11),title(5,11),x1,y1,x2,y2,
        +   isegx, isegy, isubx, isuby, x, y, istat,
        +   npts, ichar, ndecx, ndecy, itypx, itypy, icolor)
        ichar=0
        icolor=3
        npts=ne+1
        do 178 i=1,npts
            x(i)=xenvmin(i,1)
            y(i)=xenvmin(i,2)
178        continue
        call diagram
        + (title(1,11),titleinf,title(4,11),title(5,11),x1,y1,x2,y2,
        +   isegx, isegy, isubx, isuby, x, y, istat,

```

```

+      npts, ichar, ndecx, ndecy, itypx, itypy, icolor)
call tabelpos(rx, ry, xenvmax, xenvmin, ne, nd, iflag)
call clear()
      if(iflag.eq.0)then
        j=100
      elseif(iflag.eq.1)then
        j=j+1
      else if(iflag.eq.2)then
        j=j-1
      end if
c      end do
      call nmode()
    end if

```

```

*-----
case(sb, lb)
n=nblank(filename)
nn=n-4
name=filename(1:nn)//'.egv'
inquire(file=name(1:n), exist=ja)
if(ja) then
call gmode()
open(3, name(1:n))
read(3, '(i5)')ieig
do i=1, ieig
  read(3, '(i5)')irikt(i)
  read(3, '(a)')titel(i)
  read(3, '(5x, 6e10.3)')(eg(j, i, 1), j=1, (ne-1))
  read(3, '(5x, 6e10.3)')(eg(j, i, 2), j=1, (ne-1))
end do
close(3)
x(1)=0.0
x(ne+1)=0.0
y(1)=0.0
y(ne+1)=0.0
i=1
do while(i.ge.1.and. i.le.ieig)
  x(1)=0.0
  y(1)=0.0
  x(ne+1)=0.0
  y(ne+1)=0.0
  do j=1, ne-1
    x(j+1)=eg(j, i, 1)
    y(j+1)=eg(j, i, 2)
  end do
  call minmax(x, ne, amin, amax)
  if(amax-amin.ne.0.0) scalax=20./(amax-amin)
  call minmax(y, ne, amin, amax)
  if(amax-amin.ne.0.0) scalay=20./(amax-amin)
  do j=1, ne+1
    x(j)=x(j)*scalax+rx(j)
    if(irikt(i).eq.2)then
      y(j)=y(j)*scalay+ry(j)
    end if
  end do
end do

```

```

*-----
      istat=0
      ichar=0
      x1=0.0
      y1=4.5

```

```

x2=11.
y2=7.
if(iriikt(i).eq.2) y2=8.5
isegx=6
isegy=5
isubx=4
isuby=3
ndecx=2
ndecy=2
itypx=0
itypy=0
icolor=3
titleinf=-1
npts=ne+1
icolor=3
titel(13)='forstorad scala'
titel(12)=' ( M ) '
call diagram
+   (titel(i),titleinf,titel(12),titel(13),x1,y1,x2,y2,
+   isegx,isegy,isubx,isuby,x,y,istat,
+   npts,ichar,ndecx,ndecy,itypx,itypy,icolor)
istat=1
ichar=1
icolor=2
if(iriikt(i).eq.2)then
call diagram
+   (titel(i),titleinf,title(4,10),title(5,10),x1,y1,x2,y2,
+   isegx,isegy,isubx,isuby,rx,ry,istat,
+   npts,ichar,ndecx,ndecy,itypx,itypy,icolor)
else
call diagram
+   (titel(i),titleinf,title(4,10),title(5,10),x1,y1,x2,y2,
+   isegx,isegy,isubx,isuby,rx,rz,istat,
+   npts,ichar,ndecx,ndecy,itypx,itypy,icolor)
end if
x(1)=0.0
y(1)=0.0
x(ne+1)=0.0
y(ne+1)=0.0
do j=1,ne-1
    x(j+1)=eg(j,i,1)
    y(j+1)=eg(j,i,2)
end do
call tabelref(x,y,npts,9,iflag)
call clear
if(iflag.eq.0)then
    i=100
elseif(iflag.eq.1)then
    i=i+1
else if(iflag.eq.2)then
    i=i-1
end if
end do
call nmode()
end if
end select
end do
return
end

```

APPENDIX 3

Följande sidor innehåller listningen av de subrutiner som programmet DIA.FOR använder.

```

*-----*
*      Subrutin diagram  version 4/10 1991  av Francisco Herrera      *
*-----*
*      Språk                                                         *
*      =====                                                         *
*      Laheys Fortran 77                                              *
*      Subrutinen använder Graphorias graphics library.              *
*
*      Beskrivning                                                    *
*      =====                                                         *
*      Subrutinen plottar en function  $y = f(x)$  på en grafisk skärm.    *
*      Värdena för  $x$  lagras i vektorn  $x(i)$ , och  $f(x)$  på vektorn  $y(i)$ . *
*      Genom de aktuella parametrar kontrolleras utseende på grafen.  *
*
*      Formella parametrar                                             *
*      -----                                                         *
*      title : ange diagramets rubrik                                  *
*      titleinf : ange en siffer tillägg till title( t.ex node nr)    *
*      xlabel : x-axel rubrik                                          *
*      ylabel : y-axel ::                                              *
*      x1,y1 : nedersta vänstra hörnet av diagrammets fönster         *
*      x2,y2 : översta högra hörnet av diagrammets fönster           *
*      Observera att skärmen storlek är definierad av                 *
*      11.0 x 8.5 enheter i x, resp y riktning.                       *
*      Nedersta vänstra hörnet är (0.0,0.0)                           *
*      Översta högra hörnet är (11.0,8.5)                             *
*      isegx : antal lika segmenter att dela x-axel                  *
*      isegy : :: :: :: :: y-axel                                     *
*      isubx : antal lika subsegment att dela varje segment          *
*      isuby : ::                                                      *
*      x : vektor som innehåller x-värdena att plotta                 *
*      y : :: :: :: y-värdena ::                                      *
*      istat : 0 indikerar att skalning behövs göra                   *
*      1 indikerar att man vill supperponera grafen                   *
*      på den föregående i samma skala.                               *
*      npts : antal värde som innehåller x, eller y vektorn          *
*      ichar : anger linjens utseende                                 *
*      ichar = 0 ==> hel linje                                         *
*      1 ==> streckad linje                                           *
*      2 ==> som ovan men kortare streck                             *
*      3 ==> :: ::                                                    *
*      4 ==> :: ::                                                    *
*      ndecx : antal decimaler för siffrorna i x-axel                *
*      ndecy : :: :: :: y-axel                                       *
*      itypx : detta värdet bestämmer skalning för x-värdena         *
*      -itypx = 0 anger att skalning skall göras från 0               *
*      till max eller minimi-värdet.                                  *
*      -itypx = 1 anger att skalning skall göras från max            *
*      till minimi-värdet.                                            *
*      itypy : som ovan men för y-axeln.                              *
*      icolor : färgen på kurvan                                     *
*      välj 1 <= icolor <= 7 om man skall göra hårdcopior           *
*      med Shift+PrintScreen.                                         *
*
*      Deklarationer                                                  *
*      -----                                                         *
*      Integer : isegx, isegy, isubx, isuby, istat, npts, ichar,      *
*      ndecx, , ndecy, itypx, itypy, icolor.                          *
*      Real : titleinf, x1, y1, x2, y2, x, y                          *

```

```

*      Character : title,xlabel,ylabel      *
*
*      Anmärkningar:      *
*      -----      *
*      Innan man kallar subrutinn första gången måste man      *
*      sätta skärmen i grafisk läge med call gmode(),      *
*      efter detta kan man göra uppreppade call av diagram.      *
*      Variabeln istat är användbar när man vill superponera      *
*      kurvor med samma scala (istat=1) i samma graf.      *
*      (istat=0) räknar ny scala och ritar ny graf.      *
*      Med call clear() rensar man den grafiska skärmen.      *
*      Med boxcolor(...) rensas en del av skärmen.      *
*      Gör call nmode() sätter skärmen i text läge.      *
*
*      Andra subrutinr som är incorporerade i modulen      *
*      är interna och saknar intrese för användande av modulen.
*-----*
*
*      subroutine diagram
*      +(title,titleinf,xlabel,ylabel,x1,y1,x2,y2,isegx,isegy,subx,suby
*      +,x,y,istat,npts,ichar,ndecx,ndecy,itypx,itypy,icolor)
*      =====
*
*      dimension x(1000),y(1000)
*      character*(*) xlabel,ylabel,title
*      save scalex,scaley,xxmin,yymin
*      xorig=x1+1.5          ! sätter origo så att man får plats för
*      yorig=y1+0.65        ! axlarnas siffror.
*      xalen=x2-xorig-0.5
*      yalen=y2-yorig-0.6
*      if(istat.eq.1) goto 10
*      call minmax(x,npts,xmin,xmax)
*      call scala(scalex,xmax,xmin,xalen,isegx,deltax,iexpix,itypx)
*      xxmin=xmin
*      call minmax(y,npts,ymin,ymax)
*      call scala(scaley,ymax,ymin,yalen,isegy,deltay,iexpy,itypy)
*      yymin=yymin
*      call newpen(3) ! välj cyan färg
*      call rect(x1,y1,y2-y1,x2-x1,0.0,3)
*      call graph
*      +(xorig,yorig,xalen,yalen,xlabel,ylabel,title,titleinf,xmin,ymin,
*      + deltax,deltay,iexpix,iexpy,ndecx,ndecy,isegx,isegy,subx,suby)
10  if(ichar.eq.0)then
*      call pline
*      +(xorig,yorig,x,y,npts,scalex,scaley,xxmin,yymin,icolor)
*      else
*      call plined
*      +(xorig,yorig,x,y,npts,scalex,scaley,xxmin,yymin,icolor,ichar)
*      end if
*      call newpen(3)
*      return
*      end

```

```

*-----*
*      Subroutine diagram1  version 4/10 1991  av Francisco Herrera  *
*-----*
*      Språk                                                         *
*      =====                                                         *
*      Laheys Fortran 77                                              *
*      Subrutinen använder Graphorias graphics library.             *
*      *                                                                *
*      Beskrivning                                                    *
*      =====                                                         *
*      Subrutinen arbetar på samma sätt som subrutinen diagram,      *
*      max och min värde anges som parametrar. På detta sätt kan man  *
*      plotta flera kurvor på samma diagram.                          *
*      Ange värdena för max värde som det största av alla värde och min*
*      som det minsta av alla värde som skall plottas.               *
*      *                                                                *
*-----*
      subroutine diagram1
+ (title,titleinf,xlabel,ylabel,x1,y1,x2,y2,isegx,isegy,isubx,suby
+,x,y,istat,npts,ichar,ndecx,ndecy,itypx,itypy,icolor
+,xmax,xmin,ymax,ymin )
*      =====
      dimension x(npts),y(npts)
      character*(*) xlabel,ylabel,title
      save scalex,scaley,xxmin,yymin
      xorig=x1+1.5
      yorig=y1+0.65
      xalen=x2-xorig-0.5
      yalen=y2-yorig-0.6
*
      if(istat.eq.1) goto 10
      call scala(scalex,xmax,xmin,xalen,isegx,deltax,iexpix,itypx)
      xxmin=xmin
      call scala(scaley,ymax,ymin,yalen,isegy,deltay,iexpy,itypy)
      yymin=yymin
*
      call newpen(3)
      call rect(x1,y1,y2-y1,x2-x1,0.0,3)
      call graph
+ (xorig,yorig,xalen,yalen,xlabel,ylabel,title,titleinf,xmin,ymin
+,deltax,deltay,iexpix,iexpy,ndecx,ndecy,isegx,isegy,isubx,suby)
10  if(ichar.eq.0) then
      call pline(xorig,yorig,x,y,npts,scalex,scaley,xxmin,yymin,icolor)
      else
      call plined
+ (xorig,yorig,x,y,npts,scalex,scaley,xxmin,yymin,icolor,ichar)
      end if
      call newpen(3)
      return
      end

```

```

*-----*
*      Subrutin graph      version 4/10/91      av Francisco Herrera      *
*-----*
*      Subrutinen ritar axlarna med beteckningar på ett tidigare      *
*      definierat fönster      *
*      De aktuella parametrar förklaras i subrutin diagram      *
*-----*
*
  subroutine graph
+ (xorig,yorig,xalen,yalen,xxlabel,yylabel,tttitle,titleinf,xmin,ymin
+ ,deltax,deltay,iexp,x,iepy,ndecx,ndecy,isegx,isegy,isubx,suby)
=====
*
  character*80 xlabel,yylabel,tttitle
  character*80 xlabel,ylabel,title
*  reserverar variabler
  xxmin=xmin
  yymin=ymin
  deltaxx=deltax
  deltayy=deltay
  ii=0
  jj=0
  anglex=0.0
  angley=90.0
  if(ndecx .eq. 0) ndecx=-1
  if(ndecy .eq. 0) ndecy=-1
*-----*

  if(iexp .gt. 3) then
    ii=1
    ifac=0
    ifac= iexp/3
    iexp=3*ifac
    xxmin=xxmin/10.**iexp
    deltaxx=deltaxx/10.**iexp
  end if
  if(iepy .gt. 3) then
    jj=1
    ifac=0
    ifac= iepy/3
    iepy=3*ifac
    yymin=yymin/10.**iepy
    deltayy=deltayy/10.**iepy
  end if
  if(iexp .lt. 0) then
    ii=1
    xxmin=xxmin/10.**iexp
    deltaxx=deltaxx/10.**iexp
  end if
  if(iepy .lt. 0) then
    jj=1
    yymin=yymin/10.**iepy
    deltayy=deltayy/10.**iepy
  end if
*-----*
*  ritar griden
  call dashgrid(xorig,yorig,xalen,isegx,yalen,isegy)
  call rect(xorig,yorig,yalen,xalen,0.0,3)
*-----*
*  ritar axlarna och tickar de i segment och subsegment

```

```

call doyaxel(xorig,yorig,yalen,isegy,isuby)
call doxaxel(xorig,yorig,xalen,isegx,isubx)
*-----*
*   skriver siffror i y-axel
    sc=yalen/isegy
    do 10 i=1,isegy+1
      fpn=yymin+(i-1)*deltayy
      x=0.2*fpnlen(fpn,ndecy)
      call number(xorig-0.2-x,yorig-0.09+(i-1)*sc,0.2,fpn,anglex,ndecy)
10    continue
*-----*
*   skriver siffror i x-axel
    sc=xalen/isegx
    iii=isegx+1
    do 20 i=1,iii
      fpn=xxmin+(i-1)*deltaxx
      x= 0.2*fpnlen(fpn,0)/2
      call number(xorig+(i-1-x)*sc,yorig-0.3,0.2,fpn,anglex,ndecx)
20    continue
*-----*
*   skriver title
    title=tttitle
    ntitle= len(charnb(title))+1
    hight=0.2
    iside=12
    if(titleinf .lt.0.0) iside=02
    dist=0.0
    ndigit=-1
    call label(xorig,yorig+yalen+0.25,xorig+xalen,yorig+yalen+0.25,
&              title,ntitle,hight,iside,dist,titleinf,ndigit)
*-----*
*   skriver ylabel
    ylabel=yylabel
    nchary = len(charnb(ylabel))
    hight=0.2
    iside=02
    dist=1.2
    rnum=0.0
    ndigit=-1
    if(jj.eq.1) then
      ylabel(nchary+1:nchary+6)= ' x 10'
      nchary=nchary+5
    end if
    call label(xorig,yorig,xorig,yorig+yalen,
&              ylabel,nchary,hight,iside,dist,rnum,ndigit)
    if(jj .eq. 1) then
      call number(xorig-dist-0.1,999.0,0.18,real(iexpy),angley,-1)
    end if
*-----*
*   skriver xlabel
    xlabel=xxlabel
    ncharx= len(charnb(xlabel))
    hight=0.2
    iside=01
    dist=0.4
    rnum=0.0
    ndigit=-1
*-----*
    if(ii.eq.1) then

```

```

        xlabel(ncharx+1:ncharx+6)=' x 10'
        ncharx=ncharx+5
    end if
    call label(xorig,yorig,xorig+xalen,yorig,
&            xlabel,ncharx,height,inside,dist,rnum,ndigit)
    if(ii.eq.1) then
        call number(999.0,yorig-dist-0.12,0.18,real(iexp),anglex,-1)
    end if

```

```

*-----*
return
end

```

```

*-----*
*      Subrutin scala      version 4/10/91      av Francisco Herrera      *
*-----*
*      Denna subrutin beräknar skalan, antal segmenter, delning      *
*      av varje segment, och tio potensen för subrutine graph      *
*-----*
*
  subroutine scala (scale,gmax,gmin,alen,iseg,delta,ipot,ityp)
=====
  real max,min
  integer iseg
  double precision b,delning,eps
  ipot=0
  iflag=0
  eps=0.001
  max=gmax
  min=gmin
*-----*

  range=max-min
  if(range .lt. 1.e-30) then
    max=1.0
    min=-1.0
  end if
  if ( max*min .lt. 0.0) then
    ii=1
  else
    if ( ityp .eq. 0 ) then
      if(max .gt.0.0) then
        min=0.0
        ii=2
      else
        max=0.0
        ii=3
      end if
    else
      if(max .gt. 0.0) then
        ii=4
      else
        ii=5
      end if
    end if
  end if

  range=max-min
  delning=(range/iseg)
  b=dlog10(delning)
  ib=b
  b=b-ib
  if(b.lt.0) then
    b=b+1
    ib=ib-1
  end if
  delning=10.**b
  if(delning .gt.7.5+eps) then
    delning= 10.*10.**ib
  elseif(delning.gt.5.+eps) then
    delning=7.5*10.**ib
  elseif(delning.gt.3.+eps) then
    delning=5.*10.**ib

```

```

elseif(delning.gt.2.5+eps) then
    delning=3.*10.**ib
elseif(delning.gt.2.0+eps) then
    delning=2.5*10.**ib
elseif(delning.gt.1.5+eps) then
    delning=2.0*10.**ib
elseif(delning.gt.1.25+eps) then
    delning=1.5*10.**ib
elseif(delning.gt.1.00+eps) then
    delning=1.25*10.**ib
elseif(delning.ge.0.+eps) then
    delning=1.0*10.**ib
endif

```

```

if(ii .eq. 1) then
    i=max/delning
    max=delning*i
    if(max .lt. gmax-0.1*delning) then
        i=i+1
        max=delning*i
    end if
    i= min/delning
    min=delning*i
    if(min .gt. gmin+0.1*delning) then
        i=i-1
        min=delning*i
    end if
    gmax=max
    gmin=min
else if(ii .eq. 2) then
    i=max/delning
    max=delning*i
    if(max .lt. gmax) then
        i=i+1
        max=delning*i
    end if
    gmax=max
    gmin=min
else if(ii .eq. 3) then
    i=min*delning
    if (min .gt.gmin) then
        i=i-1
        min=delning*i
    end if
    gmax=max
    gmin=min
else if(ii .eq. 4) then
    i=min/delning
    min=i*delning
    i=(max-min)/delning
    max= delning*i+min
    if(max .lt. gmax) then
        i=i+1
        max=delning*i+min
    end if
    gmax=max
    gmin=min
else if(ii .eq.5) then
    i=max/delning

```

```
max=i*delning
i=(max-min)/delning
min=max-delning*i
gmax=max
gmin=min
end if
```

```
*-----*
```

```
ipot=ib+1
delta= real(delning)
iseg= anint((gmax-gmin)/delta)
scale= alen/(gmax-gmin)
return
end
```

```

*-----
*      Subrutinen ritar x-axeln
*-----
*
      subroutine doxaxel(xorig,yorig,xalen,iseg,isub)
*=====
      bigtick=0.1
      smalltick=0.04
      call plot(xorig,yorig,3)
      call plot(xorig+xalen,yorig,2)
      if(iseg .gt.0) then
        do 10 i=0,iseg
          x= xorig+(xalen/iseg)*i
          call plot(x,yorig,3)
          call plot(x,yorig-bigtick,2)
          if(isub.gt.0 .and. i.lt.iseg) then
            do 20 j=1,isub
              x=x+(xalen/iseg/(isub))
              call plot(x,yorig,3)
              call plot(x,yorig-smalltick,2)
20          continue
            end if
10        continue
          end if
      return
      end
*
*-----
*      Subrutinen ritar y-axeln
*-----
*
      subroutine doyaxel(xorig,yorig,yalen,iseg,isub)
*=====
      bigtick=0.1
      smalltick=0.04
      call plot(xorig,yorig,3)
      call plot(xorig,yorig+yalen,2)
      if(iseg .gt.0) then
        do 10 i=0,iseg
          y= yorig+(yalen/iseg)*i
          call plot(xorig,y,3)
          call plot(xorig-bigtick,y,2)
          if(isub.gt.0 .and. i.lt.iseg) then
            do 20 j=1,isub
              y=y+(yalen/iseg/(isub))
              call plot(xorig,y,3)
              call plot(xorig-smalltick,y,2)
20          continue
            end if
10        continue
          end if
      return
      end

```

```

*      Subroutine skriver en tabell på skärmen med scroll funktioner.      *
*      Här anpassad för att visa tre kolumner per rad                    *
*-----*
      subroutine tabel(x,y,ne,nr,iflag)
      =====
*      x och y = vektorer som innehåller de numeriska värde
*      ne= antal x och y värden
*      nr= tabell-titel-nummer
*      iflag returnera ett tillstånd för senare behandling

      dimension x(ne),y(ne)
      character*74 string,string1,string2
      character*80 mess
      n=0
      done=0
      iradmin=16 ! raden som tabellen börjar
      iantrad=10 ! antal raden som skall visas
      iantcol=3
      iant=iantrad*iantcol
      icol=5      ! Kolumn som tabellen börjar
      mess=' M = Menu      N = Next      P = Previous      '//
+      ' Move :      PGUP PGDN HOME END'

c      hämtar rubrikerna för tabellen
c
c      call textstring(string1,string2,nr)
      call gtext(29,1,mess(1:78))
      call rect(0.0,0.0,0.35,11.0,0.0,3)
      call gtext(iradmin-1,icol,string1)
      call gtext(iradmin,icol,string2)
      call rect(0.0,0.8,3.6,11.0,0.0,3)
*-----*

      do while(done .eq. 0)
c.... Gör tal till strängar
      do i=1,iantrad
          ind1=i+n
          ind2=i+n+iantrad
          ind3=i+n+2*iantrad
          if(ind1 .le. ne) then
              write(string (1:8),100) x(ind1)
              write(string(11:23),200) y(ind1)
          else
              write(string (1:8),300) ' '
              write(string(11:23),400) ' '
          end if
          if (ind2 .le.ne) then
              write(string(25:33),100) x(ind2)
              write(string(36:48),200) y(ind2)
          else
              write(string(25:33),300) ' '
              write(string(36:48),400) ' '
          end if
          if(ind3 .le.ne) then
              write(string(50:58),100) x(ind3)
              write(string(61:73),200) y(ind3)
          else
              write(string(50:58),300) ' '
              write(string(61:73),400) ' '
          end if
      end do
      done=.true.
  end do

```

```

        call gtext(iradmin+i,icol,string)
    end do
100  format(f8.2)
200  format(e12.3)
300  format(a7)
400  format(a10)
*-----*
    ikey=ixkey()
    select case(ikey)
        case(1072) !up
            n=n-1
        case(1080) !down
            n=n+1
        case(1073) !pgup
            n=n-iant
        case(1081) !pgdwn
            n=n+iant
        case(1071) !home
            n=0
        case(1079) !end
            n=ne
        case(77,109) !M,m
            done=1
            iflag=0
        case(78,110) !N,n
            done=1
            iflag=1
        case(80,112) !P,p
            done=1
            iflag=2
    end select
    if(n.ge.ne-mod(ne,iant)) n=ne-mod(ne,iant)
    if(n.lt.0) n=0
end do
return
end

```

```

*      Subroutine skriver en tabell på skärmen med scroll funktioner.      *
*      Här anpassad för standard avvikelse etc.                          *
*      -----*
subroutine tabelstd(tkm,tm,tkmin,tmin,tmean,tsd,ne,iflag)
=====
dimension tkm(ne),tm(ne),tkmin(ne),tmin(ne),tmean(ne),tsd(ne)
character*78 string,string1,string2
character*80 mess
n=0
done=0
iradmin=17 ! raden som tabellen börjar
iantrad=10 ! antal raden som skall visas
mess=' M = Menu      N = Next      P = Previous          '//
+   ' Move :      PGUP  PGDN  HOME  END'
string1=' Elem      maximun      time      minimun      '//
+   ' time      mean      deviation'
string2=' ====='
+   ' ====='
call gtext(29,1,mess(1:78))
call rect(0.0,0.0,0.35,11.0,0.0,3)
call gtext(iradmin-2,1,string1)
call gtext(iradmin-1,1,string2)
call rect(0.0,0.8,3.6,11.0,0.0,3)
do while(done .eq. 0)
c.... Gör tal till strängar
do 1 i=1,iantrad
write(string (1:5),100) i+n
write(string (9:19),200) tkm(i+n)
write(string(20:26),300) tm(i+n)
write(string(30:40),200) tkmin(i+n)
write(string(41:47),300) tmin(i+n)
write(string(51:61),200) tmean(i+n)
write(string(65:75),200) tsd(i+n)
call gtext(16+i,1,string)
1 continue
ikey=ixkey()
select case(ikey)
case(1072) !up
n=n-1
case(1080) !down
n=n+1
case(1073) !pgup
n=n-iantrad
case(1081) !pgdwn
n=n+iantrad
case(1071) !home
n=0
case(1079) !end
n=ne
case(77,109) !M,m
done=1
iflag=0
case(78,110) !N,n
done=1
iflag=1
case(80,112) !P,p
done=1
iflag=2
end select

```

```
      if(n.gt.ne-iantrad) n=ne-iantrad
      if(n.lt.0) n=0
      end do
100    format(i5)
200    format(e10.4)
300    format(f6.1)
      return
      end
```

```

*-----*
*      Subrutinen sätter graphics mode
*      -----
*      subroutine gmode ()
*      =====
*      call plots(0,1,0)
*      return
*      end
*-----*
*      Subrutinen rensar skärmen
*      -----
*      subroutine clear()
*      =====
*      call plot(0.0,0.0,-999)
*      return
*      end
*-----*
*      Subrutinen sätter normal mode
*      -----
*      subroutine nmode()
*      =====
*      call plot(0.0,0.0,999)
*      return
*      end
*-----*
*      Subrutinen ritar ett streckad grid
*      -----
*      subroutine dashgrid(xorig,yorig,xalen,isegx,yalen,isegy)
*      =====
*      sc=yalen/isegy
*      do i=1,isegy
*          call plot(xorig,yorig+i*sc,3)
*          call dashp(xorig+xalen,yorig+i*sc,0.05)
*      end do
*      sc=xalen/isegx
*      do i=1,isegx
*          call plot(xorig+i*sc,yorig,3)
*          call dashp(xorig+i*sc,yorig+yalen,0.05)
*      end do
*      return
*      end
*-----*
*      Subrutinen färgar ett rektangel i önskad färg
*      -----
*      subroutine boxcolor(xx,yy,h,w,icolor,icolor1)
*      =====
*      dimension x(4),y(4)
*      x(1)=xx
*      x(2)=xx+w
*      x(3)=xx+w
*      x(4)=xx
*      y(1)=yy
*      y(2)=yy
*      y(3)=yy+h
*      y(4)=yy+h
*      call newpen(icolor)
*      call rect(xx,yy,h,w,0.0,3)
*      call fill(4,x,y)

```

```

call newpen(icolor1)
call rect(xx,yy,h,w,0.0,3)
return
end

```

```

*-----*

```

```

* Subrutinen beräknar maxvärde och minvärde av en vektor

```

```

*-----*

```

```

* subroutine minmax(a,n,amin,amax)

```

```

* =====

```

```

real a(n),amin,amax

```

```

amin=a(1)

```

```

amax=a(1)

```

```

do i=2,n

```

```

    if(a(i) .lt. amin) amin=a(i)

```

```

    if(a(i) .gt. amax) amax=a(i)

```

```

end do

```

```

return

```

```

end

```

```

* Subroutine skriver en tabell på skärmen med scroll funktioner.
* Här anpassad för att visa kabel position för tiden t.
* -----
* subroutine tabelpol(refx,refy,x,y,ne,iflag)
* -----
dimension refx(ne+1),refy(ne+1),x(100),y(100)
character*65 string,string1,string2
character*80 mess
n=0
done=0
iradmin=16 ! raden som tabellen börjar
iantrad=10 ! antal raden som skall visas
icol=10
mess=' M = Menu      N = Next      P = Previous      '//
PGUP PGDN HOME END'
string1=' Node      Ref-xcoord      Ref-ycoord'//
+      '      Pos-xcoord      Pos-ycoord'
string2=' ====      =====      ====='//
+      '      =====      ====='
call gtext(29,1,mess(1:78))
call rect(0.0,0.0,0.35,11.0,0.0,3)
call gtext(iradmin-1,icol,string1)
call gtext(iradmin,icol,string2)
call rect(0.0,0.8,3.6,11.0,0.0,3)
do while(done .eq. 0)
c.... Gör tal till strängar
do 1 i=1,iantrad
write(string (1:5),100) i+n
write(string (9:19),200) refx(i+n)
write(string(23:33),200) refy(i+n)
write(string(37:47),200) x(i+n)
write(string(51:61),200) y(i+n)
call gtext(iradmin+i,icol,string)
1 continue
ikey=ixkey()
select case(ikey)
case(1072) !up
n=n-1
case(1080) !down
n=n+1
case(1073) !pgup
n=n-iantrad
case(1081) !pgdwn
n=n+iantrad
case(1071) !home
n=0
case(1079) !end
n=ne+1
case(77,109) !M,m
done=1
iflag=0
case(78,110) !N,n
done=1
iflag=1
case(80,112) !P,p
done=1
iflag=2
end select
if(n.gt.ne+1-iantrad) n=ne+1-iantrad

```

```
      if(n.lt.0) n=0
      end do
100   format(i5)
200   format(f10.3)
      return
      end
```

c-----

```

subroutine tabelpos(refx,refy,xmax,xmin,ne,ndim,iflag)
=====
* dimension refx(ne+1),refy(ne+1),xmax(100,3),xmin(100,3)
character*78 string,string1,string2
character*80 mess
n=0
done=0
iradmin=16 ! raden som tabellen börjar
iantrad=10 ! antal raden som skall visas
icol=1
mess=' M = Menu      N = Next      P = Previous      '//
PGUP PGDN HOME END'
string1=' Node   Ref-xcoord  Ref-ycoord  '//
+       ' Max-xcoord  Max-ycoord  '//
+       ' Min-xcoord  Min-ycoord  '
string2=' ====  =====  =====  '//
+       ' =====  =====  '//
+       ' =====  =====  '
call gtext(29,1,mess(1:78))
call rect(0.0,0.0,0.35,11.0,0.0,3)
call gtext(iradmin-1,icol,string1)
call gtext(iradmin,icol,string2)
call rect(0.0,0.8,3.6,11.0,0.0,3)
do while(done .eq. 0)
c.... Gör tal till strängar
do 1 i=1,iantrad
write(string (1:5),100) i+n
write(string (8:18),200) refx(i+n)
write(string(20:30),200) refy(i+n)
write(string(32:42),200) xmax(i+n,1)
write(string(44:54),200) xmax(i+n,ndim)
write(string(56:66),200) xmin(i+n,1)
write(string(68:78),200) xmin(i+n,ndim)
call gtext(iradmin+i,icol,string)
1 continue
ikey=ixkey()
select case(ikey)
case(1072) !up
n=n-1
case(1080) !down
n=n+1
case(1073) !pgup
n=n-iantrad
case(1081) !pgdwn
n=n+iantrad
case(1071) !home
n=0
case(1079) !end
n=ne+1
case(77,109) !M,m
done=1
iflag=0
case(78,110) !N,n
done=1
iflag=1
case(80,112) !P,p
done=1
iflag=2
end select

```

```
      if(n.gt.ne+1-iantrad) n=ne+1-iantrad
      if(n.lt.0) n=0
      end do
100   format(i5)
200   format(f10.3)
      return
      end
```

c-----

```

c.... Subroutine som sparar de filer som skall plottas
c.....
      subroutine wrfile(ad,av,aa,at,ae,tpl,nad,kpl,nav,naa,nat,nae,
&                      iad,iav,iaa,iat,iae,nadt,nadn,idt,idn,adt,adn,
&                      iplot,ngem,igem,tgem,xg1,xg2,ne,nd,ieig,filename)
c.....
      dimension iad(10,2),iav(5,2),iaa(5,2),iat(5),iae(5),idt(5,2),
&              idn(5,2),tpl(1000),
&              ad(10,1000),av(5,1000),aa(5,1000),at(5,1000),ae(5,1000),
&              adt(5,1000),adn(5,1000),
&              igem(20,5),tgem(20),xg1(100,20),xg2(100,20)
c.....
      double precision tgem
      character*80 filename,name*80
c.....
      n=nblank(filename)
      nn=n-4
      name=filename(1:nn)//'.inf'
      open(3,name(1:n))
      write(3,100) nad,nav,naa,nat,nae,nadt,nadn,ngem,kpl-1,iplot,ne,nd,
+      ieig
      if(nad .ne. 0)then
          write(3,100) (iad(i,1),i=1,nad)
          write(3,100) (iad(i,2),i=1,nad)
      end if
      if(nav .ne. 0) then
          write(3,120) (iav(i,1),i=1,nav)
          write(3,120) (iav(i,2),i=1,nav)
      end if
      if(naa .ne.0) then
          write(3,120) (iaa(i,1),i=1,naa)
          write(3,120) (iaa(i,2),i=1,naa)
      end if
      if(nat .ne. 0) write(3,120) (iat(i),i=1,nat)
      if(nae .ne. 0) write(3,120) (iae(i),i=1,nae)
      if(nadt .ne.0) then
          write(3,120) (idt(i,1),i=1,nadt)
          write(3,120) (idt(i,2),i=1,nadt)
      end if
      if( nadn .ne. 0)then
          write(3,120) (idn(i,1),i=1,nadn)
          write(3,120) (idn(i,2),i=1,nadn)
      end if
      if(ngem .ne.0) write(3,115)(tgem(i),i=1,ngem)
      do 8 i=1,ngem
          write(3,110)(igem(i,j),j=1,5)
8      continue
      write(3,150)(tpl(i),i=1,kpl-1)
      close (3)
c.....
      if (nad .ne. 0) then
          name=filename(1:nn)//'.dis'
          open(3,name(1:n))
          write(3,'(a)') 'HORIZONTAL DISPLACEMENT OF NODE'
          write(3,'(a)') 'VERTICAL DISPLACEMENT OF NODE'
          write(3,'(a)') 'DISPLACEMENT OUT OF PLANE OF CABLE OF NODE'
          write(3,'(a)') 'TIME ( seconds)'
          write(3,'(a)') '( M )'
          do 1 k=1,kpl

```

```

        write(3,200) (ad(i,k),i=1,nad)
1      continue
    end if
    close(3)
c.....
    if (nav .ne. 0) then
        name=filename(1:nn)//'.vel'
        open(3,name(1:n))
        write(3,'(a)') 'HORIZONTAL VELOCITY OF NODE'
        write(3,'(a)') 'VERTICAL VELOCITY OF NODE'
        write(3,'(a)') 'VELOCITY OUT OF PLANE OF THE CABLE OF NODE'
        write(3,'(a)') 'TIME ( seconds)'
        write(3,'(a)') '( M / S )'
        do 2 k=1,kpl
            write(3,300) (av(i,k),i=1,nav)
2      continue
    end if
    close(3)
c.....
    if (naa .ne. 0) then
        name=filename(1:nn)//'.acc'
        open(3,name(1:n))
        write(3,'(a)') 'HORIZONTAL ACCELERATION OF NODE'
        write(3,'(a)') 'VERTICAL ACCELERATION OF NODE'
        write(3,'(a)') 'ACCELERATION OUT OF PLANE OF THE CABLE OF NODE'
        write(3,'(a)') 'TIME ( seconds)'
        write(3,'(a)') '( M / S ** 2 )'
        do 3 k=1,kpl
            write(3,300) (aa(i,k),i=1,naa)
3      continue
    end if
    close(3)
c.....
    if (nat .ne. 0) then
        name=filename(1:nn)//'.ten'
        open(3,name(1:n))
        write(3,'(a)') 'TENSION OF ELEMENT'
        write(3,'(a)') ' '
        write(3,'(a)') ' '
        write(3,'(a)') 'TIME ( seconds)'
        write(3,'(a)') '( N )'
        do 4 k=1,kpl
            write(3,400) (at(i,k),i=1,nat)
4      continue
    end if
    close(3)
c.....
    if (nae .ne. 0) then
        name=filename(1:nn)//'.str'
        open(3,name(1:n))
        write(3,'(a)') 'STRAIN OF ELEMENT'
        write(3,'(a)') ' '
        write(3,'(a)') ' '
        write(3,'(a)') 'TIME ( seconds)'
        write(3,'(a)') ' '
        do 5 k=1,kpl
            write(3,300) (ae(i,k),i=1,nae)
5      continue
    end if

```

```

close(3)
c.....
if (nadt .ne. 0) then
  name=filename(1:nn)//'.tgf'
  open(3,name(1:n))
  write(3,'(a)') 'TANGENTIAL DRAG FORCE DIRECCION 1 OF NODE'
  write(3,'(a)') 'TANGENTIAL DRAG FORCE DIRECCION 2 OF NODE'
  write(3,'(a)') 'TANGENTIAL DRAG FORCE DIRECCION 3 OF NODE'
  write(3,'(a)') 'TIME ( seconds)'
  write(3,'(a)') '( N )'
  do 6 k=1,kpl
    write(3,300) (adt(i,k),i=1,nadt)
6    continue
  end if
  close(3)
c.....
if (nadr .ne. 0) then
  name=filename(1:nn)//'.trf'
  open(3,name(1:n))
  write(3,'(a)') 'NORMAL DRAG FORCE DIRECCION 1 OF NODE'
  write(3,'(a)') 'NORMAL DRAG FORCE DIRECCION 2 OF NODE'
  write(3,'(a)') 'NORMAL DRAG FORCE DIRECCION 3 OF NODE'
  write(3,'(a)') 'TIME ( seconds)'
  write(3,'(a)') '( N )'
  do 7 k=1,kpl
    write(3,300) (adr(i,k),i=1,nadr)
7    continue
  end if
  close(3)
c.....
if (ngem.ne.0) then
  name=filename(1:nn)//'.pox'
  open(3,name(1:n))
  write(3,'(a)') 'CABLE POSITION AT TIME (seconds)'
  write(3,'(a)') ' '
  write(3,'(a)') ' '
  write(3,'(a)') '( M )'
  write(3,'(a)') '( M )'
  name=filename(1:nn)//'.poy'
  open(7,name(1:n))
  write(7,'(a)') ' '
  write(7,'(a)') ' '
  write(7,'(a)') ' '
  write(7,'(a)') '( M )'
  write(7,'(a)') '( M )'
  do 9 i=1,ne+1
    write(3,115) (xg1(i,j),j=1,ngem)
    write(7,115) (xg2(i,j),j=1,ngem)
9    continue
  close(7)
  close(3)
  end if
c.....
100  format(13i5)
110  format(20i5)
115  format(20f12.4)
120  format(5i5)
150  format(10f10.4)
200  format(10f12.4)

```

```
300  format(5f12.4)
400  format(5f16.2)
      return
      end
```

```

c.... Subroutine som läser filerna som skall plottas
C.....
      subroutine refile(ad,av,aa,at,ae,tpl,nad,kpl,nav,naa,nat,nae,
&                      iad,iav,iaa,iat,iae,nadt,nadn,idt,idn,adt,adn,
&                      iplot,ngem,igem,tgem,xg1,xg2,ne,nd,xenvmax,
&                      xenvmin,title,rx,ry,rz,tkm,tm,tkmin,tmin,tmean,
&                      tsd,ieig,filename)
C.....
      dimension iad(10,2),iav(5,2),iaa(5,2),iat(5),iae(5),idt(5,2),
&              idn(5,2),tpl(1000),
&              ad(10,1000),av(5,1000),aa(5,1000),at(5,1000),ae(5,1000),
&              adt(5,1000),adn(5,1000),
&              igem(20,5),tgem(20),xg1(100,20),xg2(100,20),
&              tkm(100),tm(100),tkmin(100),tmin(100),tmean(100),
&              tsd(100),xenvmax(100,3),xenvmin(100,3)
C.....
      dimension rx(100),ry(100),rz(100)
      double precision tgem
      character*80 title(5,15)
      character filename*80,name*80
      logical ja
      n=nblank(filename)
      nn=n-4
      name=filename(1:nn)//'.inf'
      open(3,name(1:n))
      read(3,100) nad,nav,naa,nat,nae,nadt,nadn,ngem,kpl,iplot,ne,nd,
+      ieig
      if(nad .ne. 0)then
        read(3,100) (iad(i,1),i=1,nad)
        read(3,100) (iad(i,2),i=1,nad)
      end if
      if(nav .ne. 0) then
        read(3,120) (iav(i,1),i=1,nav)
        read(3,120) (iav(i,2),i=1,nav)
      end if
      if(naa .ne.0) then
        read(3,120) (iaa(i,1),i=1,naa)
        read(3,120) (iaa(i,2),i=1,naa)
      end if
      if(nat .ne. 0) read(3,120) (iat(i),i=1,nat)
      if(nae .ne. 0) read(3,120) (iae(i),i=1,nae)
      if(nadt .ne.0) then
        read(3,120) (idt(i,1),i=1,nadt)
        read(3,120) (idt(i,2),i=1,nadt)
      end if
      if( nadn .ne. 0)then
        read(3,120) (idn(i,1),i=1,nadn)
        read(3,120) (idn(i,2),i=1,nadn)
      end if
      if(ngem .ne.0) read(3,115)(tgem(i),i=1,ngem)
      do 8 i=1,ngem
        read(3,110)(igem(i,j),j=1,5)
8      continue
      read(3,150)(tpl(i),i=1,kpl)
      close (3)
100  format(13i5)
110  format(20i5)
115  format(20f12.4)
120  format(5i5)

```

```

150  format(10f10.4)
*-----
      name=filename(1:nn)//'.ref'
      open(3,name(1:n))
      read(3,'(a)')(title(i,10),i=1,5)
      do 19 i=1,ne+1
        read(3,'(3f12.4)') rx(i),ry(i),rz(i)
19    continue
      close(3)
c.....
      if(iplot .eq.0) goto 999
      name=filename(1:nn)//'.std'
      open(3,name(1:n))
      read(3,'(a)')(title(i,9),i=1,5)
      do 20 i=1,20
        read(3,500) j,tkm(i),tm(i),tkmin(i),tmin(i),tmean(i),tsd(i)
20    continue
      500 format(1X,I4,2(2X,E11.4,1X,F7.2),2X,E11.4,2X,E11.4)
c 500  format(3X,i5,2(4X,e11.4,1X,f9.2),5X,e11.4,3X,e11.4)
      close(3)
c.....
      if(nad .ne.0) then
        name=filename(1:nn)//'.dis'
        open(3,name(1:n))
        read(3,'(a)')(title(i,1),i=1,5)
        do 11 j=1,kpl
          read(3,'(10f12.4)') (ad(i,j),i=1,nad)
11    continue
        close(3)
      end if
c.....
      if(nav .ne.0) then
        name=filename(1:nn)//'.vel'
        open(3,name(1:n))
        read(3,'(a)')(title(i,2),i=1,5)
        do 12 j=1,kpl
          read(3,'(5f12.4)') (av(i,j),i=1,nav)
12    continue
        close(3)
      end if
c.....
      if(naa .ne.0) then
        name=filename(1:nn)//'.acc'
        open(3,name(1:n))
c      open(3,'acceler.out')
        read(3,'(a)')(title(i,3),i=1,5)
        do 13 j=1,kpl
          read(3,'(5f12.4)') (aa(i,j),i=1,naa)
13    continue
        close(3)
      end if
c.....
      if(nat .ne.0) then
        name=filename(1:nn)//'.ten'
        open(3,name(1:n))
        read(3,'(a)')(title(i,4),i=1,5)
        do 14 j=1,kpl
          read(3,'(5f16.4)') (at(i,j),i=1,nat)
14    continue

```

```

        close(3)
        end if
c.....
        if(nae .ne.0) then
            name=filename(1:nn)//'.str'
            open(3,name(1:n))
            read(3,'(a)')(title(i,5),i=1,5)
            do 15 j=1,kpl
                read(3,'(5f12.4)') (ae(i,j),i=1,nae)
15        continue
            close(3)
            end if
c.....
        if(nadt .ne.0) then
            name=filename(1:nn)//'.tgf'
            open(3,name(1:n))
            read(3,'(a)')(title(i,6),i=1,5)
            do 16 j=1,kpl
                read(3,'(5f12.4)') (adt(i,j),i=1,nadt)
16        continue
            close(3)
            end if
c.....
        if(nadn .ne.0) then
            name=filename(1:nn)//'.trf'
            open(3,name(1:n))
            read(3,'(a)')(title(i,7),i=1,5)
            do 17 j=1,kpl
                read(3,'(5f12.4)') (adn(i,j),i=1,nadn)
17        continue
            close(3)
            end if
c.....
999    if(ngem .ne.0) then
            name=filename(1:nn)//'.poy'
            open(7,name(1:n))
            read(7,'(a)')(title(i,8),i=1,5)
            name=filename(1:nn)//'.pox'
            open(3,name(1:n))
            read(3,'(a)')(title(i,8),i=1,5)
            do 18 i=1,ne+1
                read(3,'(20f12.4)') (xg1(i,j),j=1,ngem)
                read(7,'(20f12.4)') (xg2(i,j),j=1,ngem)
18        continue
            close(7)
            close(3)
            end if
            name=filename(1:nn)//'.ema'
            inquire(file=name(1:n),exist=ja)
            if(ja) then
                open(3,name(1:n))
                read(3,'(a)')(title(i,11),i=1,5)
                name=filename(1:nn)//'.emi'
                open(7,name(1:n))
                read(7,'(a)')(title(i,11),i=1,5)
                do 9 i=1,ne+1
                    read(3,'(3f12.4)') xenvmax(i,1),(xenvmax(i,j),j=2,nd)
                    read(7,'(3f12.4)') xenvmin(i,1),(xenvmin(i,j),j=2,nd)
9                continue

```

```
close(7)
close(3)
end if
```

```
C.....
```

```
    return
    end
```

```
C.....
```