

De ontwikkeling van een softwarepakket voor de grafische simulatie van een geometrie ten behoeve van modale analyse

Citation for published version (APA):

Veldhuizen, van, H. S. (1992). *De ontwikkeling van een softwarepakket voor de grafische simulatie van een geometrie ten behoeve van modale analyse*. (TH Eindhoven. Afd. Werktuigbouwkunde, Vakgroep Produktietechnologie : WPB; Vol. WPA1332). Technische Universiteit Eindhoven.

Document status and date:

Gepubliceerd: 01/01/1992

Document Version:

Uitgevers PDF, ook bekend als Version of Record

Please check the document version of this publication:

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

[Link to publication](#)

General rights

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

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

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

www.tue.nl/taverne

Take down policy

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

openaccess@tue.nl

providing details and we will investigate your claim.

**De ontwikkeling van een software-
pakket voor de grafische simulatie
van een geometrie ten behoeve van
modale analyse.**

H.S. van Veldhuizen

W.P.A. Rapport 1332

Dit verslag bevat de opzet van het software-pakket, dat de grafische simulatie van het HP 3566A data-acquisitie systeem beschrijft.

Augustus 1992.

TECHNISCHE UNIVERSITEIT EINDHOVEN
Faculteit Werktuigbouwkunde
Vakgroep W.P.A.

VOORWOORD

Het verslag bevat de uitwerking en de verduidelijking van mijn onderzoeksopdracht.

De onderzoeksopdracht ziet er als volgt uit:

Onderwerp: Het opzetten van een programma om de metingen verkregen met het HP 3566A data-acquisitie systeem geschikt te maken voor modale analyse.

Toelichting: De vakgroep beschikt over een 8-kanaals data-acquisitie systeem van de firma Hewlett Packard. Met dit systeem kunnen tot 8 kanalen meetsignalen worden ingelezen en met behulp van Fourier technieken worden omgewerkt van het tijds- naar het frequentie-domein. Om deze metingen geschikt te maken voor de modale analyse technieken moeten nog een aantal bewerkingen worden uitgevoerd.

Opdracht:

Ontwerp een softwarepakket waarin de geometrie van een structuur kan worden ingegeven en grafisch kan worden afgebeeld. Hiervoor is het noodzakelijk om de volgende gegevens in te kunnen voeren:

1. coördinaten van de meetpunten.
2. de wijze waarop de punten met elkaar verbonden zijn.
3. de gegevens van de afzonderlijke componenten voor wat betreft positie en oriëntatie.
4. de aanzichten van de structuur.

Het softwarepakket dient te worden beschreven in Turbo Pascal 6.0.

Om modale analyse uit te kunnen voeren is het noodzakelijk de metingen van de HP 3566A geschikt te maken voor de volgende bewerkingen:

1. meetgegevens grafisch afbeelden op het beeldscherm.
2. metingen transformeren van verplaatsing naar snelheid en versnelling en omgekeerd.
3. met de cursor uitkiezen van een te onderzoeken gebied.
4. gekozen gebied uitvergroten.
5. in dit gebied de eigenfrequenties, dempingen en residuen berekenen.

Geef de structuur aan waarbinnen deze bewerkingen plaats kunnen vinden.

De opdracht vond plaats onder supervisie van:
afstudeer hoogleraar: prof.dr.ir. A.C.H. van der Wolf.
begeleider: ir. J.A.W. Hijink.

INHOUDSOPGAVE.

	blz.
HOOFDSTUK 1: Inleiding.	4
HOOFDSTUK 2: Het programma.	5
2.1 De algemene opzet.	5
2.2 De programma opbouw.	5
2.3 De variabelen.	7
2.4 De beschrijving en toelichting.	9
2.4.1 De tabellen.	9
2.4.2 Het grafische gedeelte.	10
2.4.3 De overige units.	11
HOOFDSTUK 3: De gebruiksaanwijzing.	12
3.1 De bediening.	12
3.2 De mogelijkheden.	14
HOOFDSTUK 4: De achtergrond.	20
4.1 De keuzes.	20
4.2 De theorie.	22

HOOFDSTUK 5: Het implementeren:	24
5.1 Inleiding.	24
5.2 Het afbeelden.	24
5.3 Het transformeren.	26
5.4 Het gebied.	27
5.5 Het uitvergrooten.	27
5.6 Het omrekenen.	27
Hoofdstuk 6: Conclusies en aanbevelingen.	28
6.1 Suggesties	28
Literatuur	29

HOOFDSTUK 1: Inleiding.

Het eerste gedeelte van de onderzoeksopdracht, het softwarepakket, staat beschreven in de hoofdstukken 2, 3 en 4.

Hoofdstuk 2 bevat het theoretische gedeelte van het programma. In dit hoofdstuk staat beschreven wat de algemene opzet van het programma is en waarom het volgens een bepaalde structuur is opgebouwd. De variabelen, procedures alsmede de units worden nader op hun werking, opbouw en benaming toegelicht. Het is voor het optimale inzicht in het programma aan te bevelen om de pascalcode erbij te houden.

Hoofdstuk 3 bevat het gebruikersgedeelte van het programma en dient als gebruiksaanwijzing. Het eerste gedeelte van het hoofdstuk dient als een beknopt praktisch hulpmiddel bij het ingeven van de geometrie. Het tweede gedeelte bespreekt de mogelijkheden van het programma, waardoor men enig idee krijgt wat, met behulp van het praktische eerste gedeelte van het hoofdstuk, mogelijk is. Men dient enigszins op de hoogte te zijn van de werking van de HP 5423A Structural Dynamics Analyzer [1]. Men dient te weten wat voor consequenties het invullen van de lijsten heeft en hoe punten met elkaar verbonden worden. Er is vanuit gegaan dat de gebruiker van het oude model structural dynamics analyzer nu met een kleine toelichting in hoofdstuk 3 achter de computer zijn werk kan vervolgen.

Hoofdstuk 4 vertelt waarom er in het programma bepaalde keuzes bij het programmeren zijn gemaakt. Tevens bevat het een korte toelichting van de gebruikte theorie.

Het tweede gedeelte van de onderzoeksopdracht, de structuur waarbinnen metingen verwerkt worden, wordt beschreven in hoofdstuk 5.

HOOFDSTUK 2: Het programma.

2.1 De algemene opzet.

Om het programma overzichtelijk te houden zijn er een aantal maatregelen genomen:

Alle bij elkaar horende variabelen zijn onder gebracht in een record. Hierdoor zijn de variabel-namen niet alleen in naam gelijk, maar ook in functionaliteit. Een record kan in zijn geheel bijvoorbeeld weggeschreven worden of aan een procedure meegegeven worden.

Het type van een record wordt voorafgegaan door 'type_' terwijl de als het record lokaal is afgeleid van een globale record, deze wordt voorafgegaan door 'loc_'.

Alle functioneel gelijksoortige procedures zijn onder gebracht in units.

Functies (of procedures) die slechts in één procedure worden aangeroepen, worden in deze procedure gedeclareerd. Dit geeft twee voordelen. Ten eerste hoeft de functie dan geen lokale variabelen te gebruiken, omdat voor de programmeur duidelijk is welke variabelen veranderd worden. Ten tweede kunnen in procedures die analoog aan elkaar zijn opgebouwd, ook de namen van de functies gelijk blijven. Dit laatste wordt vooral toegepast bij de tabellen.

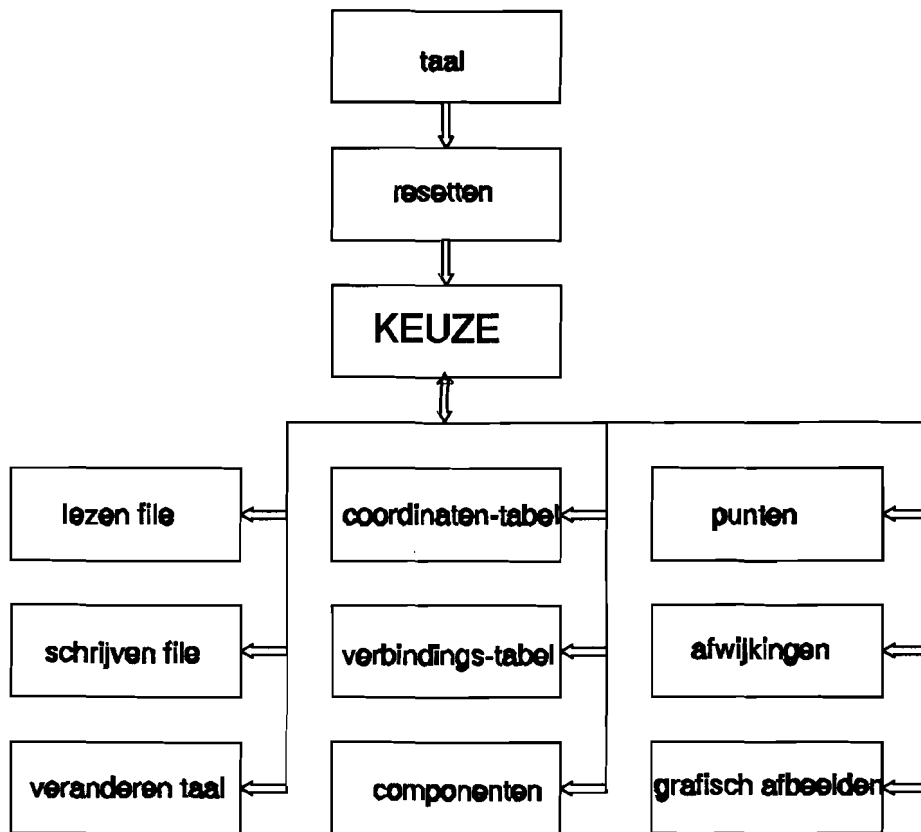
2.2 De programma opbouw.

Voor de grafische representatie van een structuur en de bewegingen daarvan, zijn een aantal gegevens nodig. Deze gegevens moeten naar disk geschreven kunnen worden om deze waarden naderhand terug te kunnen lezen.

Om dit te bereiken bestaat de kern van het programma uit een keuzemenu. Voordat het keuzemenu getoond wordt, worden (bij het opstarten) de variabelen gereset en wordt de taal bepaald.

Het keuzemenu zelf geeft de mogelijkheid om gebruik te maken van één van de vijf tabel-invoer procedures, het daadwerkelijk afbeelden van de structuur, het lezen en schrijven van alle gegevens naar disk en de taal te veranderen (zie figuur 1).

Nadat één van zo'n procedure is doorlopen komt het programma weer terug bij het keuzemenu. Tevens wordt dan steeds een backup-file gemaakt, met de naam BACKUP. Mocht het programma vastlopen, dan bevat deze file de laatste gegevens.



figuur 1 De structuur.

Verder zijn er nog een aantal procedures die nodig zijn om bijvoorbeeld een string en andere karakters in te lezen.

2.3 De variabelen.

De belangrijkste variabelen in het programma zijn de records waarin alle in te voeren gegevens staan.

Deze records bevatten ook arrays. De lengte van deze arrays bepalen hoeveel punten, verbindinglijnen en componenten het programma aankan. Hiervoor zijn de volgende constanten gedeclareerd:

```
const
  max_point   : integer = 200;
  max_verb    : integer = 300;
  max_comp    : integer = 40;
```

In turbo pascal is het niet mogelijk een array te declareren met de grootte [1..max_point]. Hier moet staan [1..200].

De constanten kunnen dus gewoon gewijzigd worden (zolang de geheugen ruimte dit toelaat), maar de waarden van de arrays moet dan ook gewijzigd worden. Behalve in de *unit* declare moet dit ook in de *unit* graphic gebeuren.

Men dient er rekening mee te houden dat bij wijziging, oude files niet meer te lezen zijn!

De records zijn:

```
type_coor = record
  syst:array [1..200] of char;
  x:array [1..200] of real;
  y:array [1..200] of real;
  z:array [1..200] of real;
  r:array [1..200] of real;
  phi:array [1..200] of real;
  comp:array [1..200] of integer;
  max:integer;
end;
```

Hierin is:

- syst: Het soort coördinaten stelsel (R/C).
- x: Altijd de X-coördinaat van het punt.
- y: Altijd de Y-coördinaat van het punt.
- z: Altijd de Z-coördinaat van het punt.
- r: Soms de straal van het punt (bij cilindrische coördinaten).
- phi: Soms de hoek van het punt (bij cilindrische coördinaten).
- comp: Tot welke component het punt behoort.
- max: Het aantal punten.

```
type verb = record
  line:array [1..300] of integer;
  max:integer;
end;
```

Hierin is:

line: Het verbindings-punt.
max: Het aantal verbindings-punten.

```
type visb = record
  point:array [1..200] of char;
  comp:array [1..200] of boolean;
  max:integer;
end;
```

Hierin is:

point: De wijze van zichtbaarheid (N,Y,P).
comp: Of een punt al onzichtbaar is door component-instelling.
max: Het aantal punten (is gelijk aan *coor.max*).

```
type comp = record
  nul_x:array [1..40] of integer;
  nul_y:array [1..40] of integer;
  nul_z:array [1..40] of integer;
  rot_x:array [1..40] of integer;
  rot_y:array [1..40] of integer;
  rot_z:array [1..40] of integer;
  show:array [1..40] of char;
  used:array [1..40] of boolean;
end;
```

Hierin is:

nul_x: De X-coördinaat van de nulpuntsverschuiving van een component.
nul_y: De Y-coördinaat van de nulpuntsverschuiving van een component.
nul_z: De Z-coördinaat van de nulpuntsverschuiving van een component.
rot_x: De rotatie om de x-as in de (3-D) ruimte.
rot_y: De rotatie om de y-as in de (3-D) ruimte.
rot_z: De rotatie om de z-as in de (3-D) ruimte.
show: De wijze van zichtbaarheid (N,Y,C).
used: Of het component gebruikt wordt.

```

type_freq = record
  x:array [1..200] of real;
  y:array [1..200] of real;
  z:array [1..200] of real;
  frq:array [1..200] of real;
  damp:array [1..200] of real;
  inc_time:real;
  max:integer;
end;

```

Hierin is:

- x: De X-component van de uitwijking.
- y: De Y-component van de uitwijking.
- z: De Z-component van de uitwijking.
- frq: De frequentie van de uitwijking.
- damp: De demping van de uitwijking.
- inc_time: De verhoging van de tijd per simulatie stap.
- max: Het aantal punten (is gelijk aan coor.max).

2.4 De beschrijving en toelichting.

In de volgende paragrafen worden de belangrijkste procedures beschreven en toegelicht.

2.4.1 De Tabellen.

Alle tabel ingeef-routines zijn volgens hetzelfde schema opgebouwd.

```

begin
  reset de variabelen
  zet de layout op het scherm (procedure: layout)
  zet de waarden op het scherm (procedure: set_on_screen)
  repeat
    evalueer de toetsen (procedure: evaluate_keys)
    zet de waarden op het scherm (procedure: set_on_screen)
  until einde
end;

```

Tijdens het programma wordt (door b.v. de cursor-toetsen) steeds van plaats veranderd. Dit is te scheiden in de plaats in het array en de plaats op het scherm. Om het gemakkelijk te houden zijn deze plaatsen strikt gescheiden gehouden.

De procedure *evaluate_keys* verandert alleen de plaats in het array (en dus niet van het scherm).

De procedure *set_on_screen* berekent met behulp van deze waarde (en de waarde van de plaats in het array voor de toetsaanslag) de positie op het scherm.

De procedure *evaluate_keys* bekijkt of er een toets is ingedrukt, en voert de bijbehorende actie uit.

De procedure *set_on_screen* is overgenomen uit de programmer's guide van Hisoft-pascal (MSX) [4].

Al de procedures *get_coordinates*, *get_verbinding*, *get_components*, *get_visible* en *get_frequenties* zijn opgenomen in de unit *INPUT*.

2.4.2 Het grafische gedeelte.

De procedure *give_graphic* geeft de componenten weer op het grafische scherm. Deze procedure maakt gebruik van andere procedures.

Graphic_screen schakelt het grafisch scherm in. De variabelen *Graphdriver* en *Graphmode* bepalen de schermresolutie. Voor driver 9 is de file *EGAVGA.BGI* nodig, die bij Turbo Pascal wordt meegeleverd. De bijbehorende modes zijn 0 (640 * 200), 1 (640 * 350) en 2 (640 * 480). De maximale resolutie hangt af van de grafische kaart die in de PC zit en zal dus van computer tot computer verschillen. Eventueel kan ook gebruik worden gemaakt van driver 6 (*IBM8514.BGI*) mode 0 (640 * 480).

Scaling berekent de schaalfactor. Van alle zichtbare punten wordt bekeken wat de minimale en maximale x en y coördinaten op het beeld zijn. Dit gebeurt bij een maximale positieve en negatieve uitwijking. Uit de uiteindelijk grootste en kleinste waarden kan nu de schaal worden berekend zodat de afbeelding binnen het scherm blijft.

Voor de omrekening van de (3-D) ruimte coördinaten naar de schermcoördinaten wordt gebruik gemaakt van de procedure *screen_coor*. Deze transformeert de coördinaten met behulp van de in hoofdstuk 4.2 besproken matrix. Om de simulatie sneller te doen verlopen (alhoewel een co-processor altijd uitkomst biedt) worden deze matrix-waarden *per component* in een matrix gezet, zodat deze niet iedere keer (in de simulatie) opnieuw berekend hoeft te worden.

Key evalueert de toetsen. Indien nodig wordt opnieuw de schaal berekend.

Give_mask geeft de layout. Hierin is ook de procedure *give_axis* opgenomen die het assenstelsel rechtsonder op het scherm tekent.

De procedure *lines* zet uiteindelijk de componenten op het scherm. Deze maakt weer gebruik van de procedure *visible*, die kijkt of een punt afgebeeld moet worden (geeft 0..4: 0: niet 1: normaal 2: punt een cirkel 3: lijn gekleurd 4: punt een cirkel en lijn gekleurd).

2.4.3 De overige units.

De overige procedures spreken voor zich. Zij zijn ingedeeld in een aantal units.

De unit *language* bevat de procedure *talen* die een array met strings, die gevuld wordt met zinnen in de juiste taal.

De unit *declare*, declareert alle records en constanten.

De unit *utils* een aantal hulp procedures:

Get_real_input en *get_string_input* zijn idiot-proof procedures die een real, dan wel een string inlezen. In tegenstelling tot *readln* is het hier mogelijk een eerste karakter mee te geven. Verder is het bij *get_real_input* mogelijk om een minimum en een maximum waarde in te stellen.

De procedures *normal_color*, *low_intensity* en *inverse* zorgen ervoor dat de tekst resp. normaal, in donkergrijs of geïnverteerd (zwart op wit) afgebeeld wordt.

De procedure *pressed* geeft een integer-waarde van een toets.

String_tot_char converteert het eerste element van een string naar een character.

Test kijkt of een filenaam (al) bestaat.

HOOFDSTUK 3: De gebruiksaanwijzing.

3.1 De Bediening.

Het uitgangspunt van het programma is het menu. Van hieruit kan men de gegevens over de structuur invoeren in 5 tabellen, de gegevens opslaan of uitlezen, de structuur bekijken of de taal veranderen.

Deze paragraaf beschrijft kort welke keuzes (in het programma) gemaakt kunnen worden en welke toetsen bruikbaar zijn voor het ingeven van de gegevens. Het is bij het ingeven van de gegevens handig deze beknopte praktische gebruiksaanwijzing erbij te houden.

COÖRDINATEN-TABEL.

Met de cursortoetsen gaat men naar het juiste veld en men vult de coördinaatwaarden van een punt in. Het is niet nodig deze ingave met een return af te sluiten. Bij het stelsel kan gekozen worden uit R (carthetische-coördinaten) en C (cilinder-coördinaten). De x-, y-en z-kolom staan voor de respectievelijke coördinaten. Phi en r staan voor de straal en de hoek bij het ingeven in cilinder-coördinaten. Bij component kan ingegeven worden bij welke component (1..40) het punt hoort.

Men kan het beste eerst het soort stelsel invullen, omdat bij verandering van het stelsel de coördinaten automatisch mee omgerekend worden.

cursortoetsen:	geven de bijbehorende bewegingen op het scherm
omlaag:	als de cursor in eerste kolom staat, worden de waarden gecopieerd naar de volgende rij.
insert, delete:	bijvoegen en weghalen van punten.
home, pgup, pgdn:	brenkt de cursortoets naar het eerste punt, of scrollt pagina omhoog of omlaag.
F1, F2:	F1 verhoogt een kolom met een in te stellen waarde als de cursor omlaag wordt bewogen. F2 stopt dit.
end of Esc:	terug naar menu.

VERBINDINGS-TABEL.

De toetsen zijn het zelfde als voor coördinaten-tabel m.u.v. de toetsen F1 en F2 die in deze tabel geen functie hebben.

Als men in de linkerkolom omlaag gaat wordt de rechterkolom automatisch met 1 opgehoogd.

COMPONENTEN-TABEL.

De kolommen 'nul x', 'nul y' en 'nul z' staan voor de nulpuntsverschuiving van een component in de (3-D) ruimte. 'rot x', 'rot y' en 'rot z' zorgen voor de rotatie om respectievelijk de x-, y- en z-as, van een component in de ruimte. 'zichtbaar' bepaald of een component afgebeeld wordt of niet. De waarde hiervoor kan 'N' (niet afbeelden), 'Y' (normaal afbeelden) of 'C' (gekleurd afbeelden) zijn.

De component 'totaal' geeft een verschuiving en rotatie aan van alle componenten, terwijl de waarden van de componenten op zich relatief daaraan zijn. Als een component in de tabel grijs is afgebeeld, wordt deze niet gebruikt. De toetsen zijn het zelfde als voor coördinaten-tabel m.u.v. F1, F2, insert en delete.

PUNTEN-TABEL.

In de rechter kolom kan aangegeven worden of een punt afgebeeld moet worden. De mogelijkheden zijn 'N' (niet afbeelden), 'Y' (wel afbeelden) en 'P' (als cirkeltje afbeelden). Als een punt grijs is betekent dat deze door de instelling van de component niet zichtbaar is.

De toetsen zijn het zelfde als voor coördinaten-tabel m.u.v. F1, F2, insert en delete.

FREQUENTIE-TABEL.

De kolommen 'x', 'y' en 'z' staan voor de afwijking in de x-, y- en z-richting. 'frequentie' en 'demping' geven respectievelijk de frequentie en demping van een punt aan.

De toetsen zijn het zelfde als voor coördinaten-tabel m.u.v. F1, F2, insert en delete.

HET GRAFISCHE GEDEELTE.

Beeldt de ingegeven waarden af. Voor wijzigingen kan met F1 en F2 het component of de gehele geometrie gekozen worden. Met F3 en F4 kan de te wijziging functie gekozen worden. Met + en - kunnen de waarden verhoogd of verlaagd worden. Met F5 en F6 kan het beeld stil gezet worden, of weer lopend. Met end of esc kan het grafisch gedeelte verlaten worden.

nummer	lijn
1	1
2	2
3	3
4	4
5	1
6	-5
7	6
8	7
9	8
10	5
11	-1
12	5

linker kolom: incrementeel end: einde verbindings-tabel

figuur 3 De verbindingstabel.

COMPONENTEN-TABEL.

De componenten-tabel verzorgt alle informatie voor de visualisatie van de afzonderlijke componenten. Het is mogelijk om de componenten ten opzichte van elkaar te transleren en te roteren. Het transleren gebeurt door de componenten eigen nulpunten te geven en zodoende door nulpuntsverschuivingen, van de afzonderlijke componenten, de componenten te transleren ten opzichte van elkaar. Het roteren van de componenten gebeurt door in de tabel de rotatie om de x-, y- en z-as van de afzonderlijke componenten in te geven. De componenten kunnen ook nog afzonderlijk zichtbaar, onzichtbaar of gekleurd bekeken worden. Het is later ook nog mogelijk de componenten afzonderlijk te transleren en te roteren tijdens de grafische simulatie met behulp van de funktietoetsen (zie grafisch).

component	nulpunt			rotatie			zichtbaar
	x	y	z	x	y	z	
total	0.000	0.000	0.000	20	20	325	
1	0.000	0.000	0.000	0	0	0	C
2	0.000	0.000	0.000	0	0	0	Y
3	0.000	0.000	0.000	0	0	0	Y
4	0.000	0.000	0.000	0	0	0	Y
5	0.000	0.000	0.000	0	0	0	Y
6	0.000	0.000	0.000	0	0	0	Y
7	0.000	0.000	0.000	0	0	0	Y
8	0.000	0.000	0.000	0	0	0	Y
9	0.000	0.000	0.000	0	0	0	Y
10	0.000	0.000	0.000	0	0	0	Y
11	0.000	0.000	0.000	0	0	0	Y
12	0.000	0.000	0.000	0	0	0	Y
13	0.000	0.000	0.000	0	0	0	Y
14	0.000	0.000	0.000	0	0	0	Y
15	0.000	0.000	0.000	0	0	0	Y
16	0.000	0.000	0.000	0	0	0	Y
17	0.000	0.000	0.000	0	0	0	Y
18	0.000	0.000	0.000	0	0	0	Y
19	0.000	0.000	0.000	0	0	0	Y
20	0.000	0.000	0.000	0	0	0	Y

N:nee Y:ja C:gekleurd
linker kolom: copieren end: einde grijze componenten worden niet gebruikt
componenten-tabel

figuur 4 De componententabel.

PUNTEN-TABEL.

De punten-tabel maakt het mogelijk bepaalde punten uit de gesimuleerde geometrie op het beeldscherm op te laten lichten. Men kan in de puntentabel van alle punten in de geometrie bepalen of het punt extra groot, wel of niet oplicht. De waarden van de eerder ingevulde punten kunnen ook nu gecopiëerd worden door in linkerkolom naar beneden te gaan.

De punten-tabel maakt het mogelijk om tijdens een simulatie een of meerdere punten beter te tonen en te bestuderen.

nummer	zichtbaar
1	P
2	Y
3	Y
4	Y
5	P
6	Y
7	Y
8	Y

grijze punten worden niet afgebeeld (door component-instelling)
N: nee Y: ja P: grote punt links: copieren end: einde punten-tabel

figuur 5 De puntentabel.

FREQUENTIE-TABEL.

In de frequentie-tabel is het mogelijk om van elk punt in de geometrie de uitwijking, of vibratie, in te voeren. Men kan van elk punt aangeven wat de grootte van de uitwijking is in respectievelijk de x-, y- en z-richting. Het is mogelijk om de ingegeven waarden van vorige punten te copieëren. De frequentie en de demping van de uitwijking van het punt zijn ook in te geven. Het ingeven van de demping heeft echter in dit programma geen werking, maar is voor eventuele latere opname vast neergezet. Men kan dus de uitwijking van elk willekeurig punt in de geometrie volledig beschrijven.

punt	x	y	z	frequentie	demping
1	1.000	2.000	3.000	4.000	3.400
2	0.000	3.000	0.000	5.000	3.700
3	1.000	2.000	3.000	4.000	9.900
4	0.000	0.000	3.000	1.000	3.400
5	2.000	0.000	0.000	1.000	2.900
6	0.000	2.000	2.000	2.000	3.400
7	0.000	1.000	0.000	3.000	5.900
8	0.000	0.000	2.000	2.000	5.100

linker kolom: copieren end: einde frequentie-tabel

figuur 6 De frequentietabel.

HET GRAFISCHE GEDEELTE.

Het grafische gedeelte geeft uiteindelijk de gesimuleerde geometrie weer met al zijn ingevoerde gegevens. Het is ook tijdens de grafische simulatie nog mogelijk veranderingen aan de gesimuleerde geometrie aan te brengen. Men kan de nulpunten en hoeken van zowel de componenten als van de gehele geometrie tijdens de simulatie met behulp van de funktietoetsen uitzoeken en met de +/- toetsen veranderen. De amplituden en frequentie van de gehele geometrie zijn ook nog in het grafische gedeelte te veranderen met de +/- toetsen. De geometrie is ook nog stil en weer in beweging te zetten met de funktietoetsen. Als de geometrie stilgezet is, zijn alle functies nog operationeel en zijn er nog veranderingen (met de funktietoetsen) aan de geometrie door te geven. Alle veranderingen die hebben plaatsgevonden met behulp van de daarvoor bestemde toetsen worden geregistreerd in de betreffende tabellen. Zo kan men als men uit het grafische gedeelte gaat in de tabellen zien wat de (met de funktietoetsen) nieuw ingestelde waarden zijn. Deze veranderingen worden dus ook meegegeven bij het weg-schrijven van de gegevens naar een file. De goed ingestelde grafische presentatie blijft dus behouden.



figuur 7 De grafische simulatie.

SCHRIJVEN FILES.

Het is mogelijk een ingevoerde geometrie op te slaan in een file. De file kan onder elke willekeurige naam opgeslagen worden. Als de filenaam al bestaat dan wordt dat gemeld. Wordt nu toch doorgedaan met het opslaan van de file onder de reeds bestaande naam, dan wordt de oude file overschreven.

LEZEN FILES.

Een opgeslagen geometrie in een bepaalde file kan opgevraagd worden in de menukeuze lezen files. Als de op te vragen filenaam niet bestaat dan wordt dit gemeld. Hierna kan men een andere naam proberen.

HOOFDSTUK 4: De achtergrond.

4.1 De keuzes.

Tijdens het programmeren moeten er altijd keuzes gemaakt worden hoe iets opgelost moet worden. Ook zijn er dingen die op het eerste gezicht niet handig lijken, maar bij nader inzien toch nut hebben. Hieronder zijn enkele 'oplossingen' beschreven:

Bij het record coor zijn er de velden x , y , r en ϕ . Op zich zou alleen veld x en y volstaan, omdat het veld *sys* de aard van de inhoud aangeeft (carthetisch of cilinder-coördinaten). Er is echter gekozen voor een overzichtelijker methode, waarbij x en y altijd de carthetische coördinaten bevatten (dus eventueel omgerekend vanuit de cilinder-coördinaten). Hierdoor heeft bijvoorbeeld de procedure *graphic* geen last meer van het soort stelsel, maar kan altijd gebruik maken van x en y .

De punten-tabel (voor het invullen van de zichtbaarheid van een punt) had samen gekund met de coördinaten-tabel.

De coördinaten-tabel wordt echter eenmaal per object ingevoerd. De zichtbaarheid veel vaker. Bij het invullen is altijd kans dat je (per ongeluk) iets veranderd, en daarom is er voor een aparte tabel gekozen.

Om de zelfde reden is ook wanneer je iets invult, de inverse kleur getoond. Mocht je per ongeluk de verkeerde toets aanslaan en met invullen beginnen, dan komt die waarde daar ook werkelijk te staan. Om dit duidelijk te maken wordt de inverse kleur getoond.

Bij de componenten-tabel zijn de niet gebruikte componenten donkergrijs afgebeeld. Het is namelijk mogelijk om b.v. component 1 en 10 te gebruiken. Hierdoor zijn 2 t/m 9 ongebruikt.

Bij het grafische gedeelte kan met een functie toets uitgekozen worden welke functie er bij de + en - toetsen hoort. Op zich is dat omslachtig. Maar als voor iedere functie een toets zou zijn gedefinieerd, zou het gebruik minder handig worden.

Bij het record *freq*, is een veld *inc_time*. Deze geeft de snelheid van de simulatie. Tijdens de simulatie wordt er een variabele *time* gebruikt (die terugkomt in de afwijking $\sin(\text{frequency} * \text{time})$). Deze *inc_time* is aan dit record toegevoegd om het gemakkelijk te kunnen saven.

Bij het opstarten van een tabel, staat de cursor in de tweede kolom. Dit is gedaan omdat je normaal gesproken naar beneden beweegt, als men daar iets wil zien. De copieerfunctie zou dan alles wissen. Om die reden kom je, als je na de rechterkolom verder gaat ook weer bij kolom 2 uit.

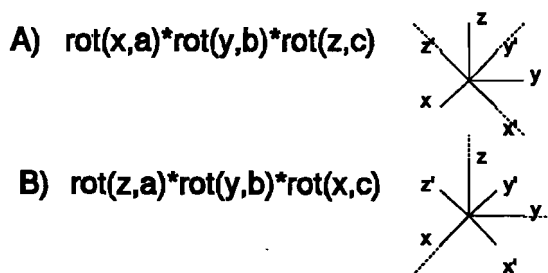
4.2 De theorie.

Het transleren van een punt in een ruimtelijk coördinatenstelsel gebeurt door de coördinaten op te tellen bij de translatiecoördinaten [3]. Het roteren van een punt om een as vindt plaats door de coördinaten van het punt te vermenigvuldigen met de rotatiematrix om de betreffende as. Rotatie om meerdere assen betekent dat men de coördinaten vermenigvuldigt met de rotatiematrices om de betreffende assen. De volgorde van de vermenigvuldigingen met de rotatiematrices is echter zeer belangrijk (zie figuur 8).

$$\text{rot}(x,a)= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(a) & -\sin(a) \\ 0 & \sin(a) & \cos(a) \end{bmatrix}$$

$$\text{rot}(y,b)= \begin{bmatrix} \cos(b) & 0 & \sin(b) \\ 0 & 1 & 0 \\ -\sin(b) & 0 & \cos(b) \end{bmatrix}$$

$$\text{rot}(z,c)= \begin{bmatrix} \cos(c) & -\sin(c) & 0 \\ \sin(c) & \cos(c) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



figuur 8.

Als men de matrices in de volgorde a vermenigvuldigd, dan roteert men ten opzichte van het bewegende assenstelsel. Als men de matrices in de volgorde b vermenigvuldigd, dan roteert men ten opzicht van het vaste, in de ruimte aanwezige, assenstelsel. In dit programma wordt altijd, bij een rotatie, rond het bewegende assenstelsel geroteerd.

HOOFDSTUK 5: Het implementeren.

5.1 Inleiding.

Om modale analyse uit te kunnen voeren is het noodzakelijk de metingen van de HP 3566A geschikt te maken voor de volgende bewerkingen:

1. meetgegevens grafisch afbeelden op het beeldscherm.
2. metingen transformeren van verplaatsing naar snelheid en versnelling en omgekeerd.
3. met de cursor uitkiezen van een te onderzoeken gebied.
4. gekozen gebied uitvergroten.
5. in dit gebied de eigenfrequenties, dempingen en residuen berekenen.

Deze bewerkingen vinden na elkaar plaats (figuur 3). De bewerkingen 1 en 2 hoeven niet noodzakelijk na elkaar plaats te vinden. In de volgende paragrafen wordt de structuur waarbinnen deze bewerkingen plaats vinden aangegeven. De programmaschetsen zijn vermeld in turbo pascal.

5.2 Het afbeelden.

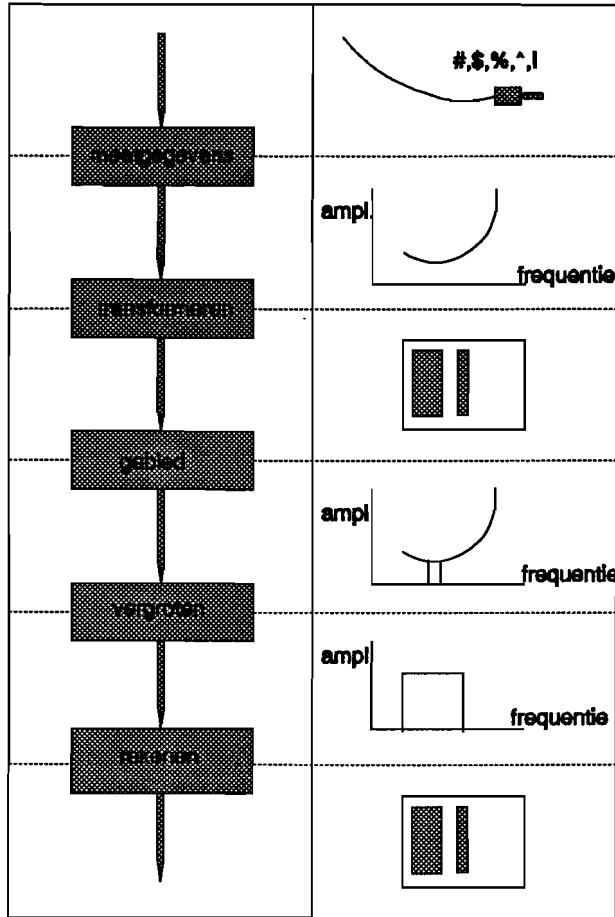
Het afbeelden van de meetgegevens op het beeldscherm gebeurt door op de x-as in de grafiek de frequentie van de exetatie neer te zetten en door de grootte van de grootste verplaatsing van het meetpunt op de y-as neer te zetten. Als men de waarde van de grootste amplitude per frequentie eenmaal heeft is het met een loop mogelijk deze waarden voor de totale frequentieband neer te zetten. Om de waarden allemaal op het beeldscherm te krijgen is het mogelijk een schaal aanpassing in de x- en y-as te verwezenlijken. Hierbij kan men hetzelfde principe als in de hierboven beschreven procedure maxschaal gebruiken.

programmaschets:

```
for i:=1 tot max.amplitude do
begin
  get max.amplitude[i];
end;

schaalx:= ( beeldafmeting in x-richting/ max. frequentie);
schaaly:= ( beeldafmeting in y-richting/ max. amplitude);

for i:=1 tot max.frequentie do
begin
  point( frq[i]*schaalx, amplitude[i]*schaaly);
  line( frq[i-1]*schaalx, amplitude[i-1]*schaaly,
        frq[i]*schaalx, amplitude[i]*schaaly);
end.
```



figuur 9.

5.3 Het transformeren.

Het transformeren van verplaatsing naar snelheid en versnelling en omgekeerd worden berekend in het frequentiedomein.

Met behulp van fourieranalyse zijn de transformaties eenvoudig uit te voeren.

De uitwijking wordt gegeven door:

$$u = \operatorname{Re}(Ae^{-i\omega t})$$

Waarbij:

$$e^{-i\omega t} = \cos(\omega t) + i\sin(\omega t)$$

Het reële gedeelte is dan:

$$u = \cos(\omega t)$$

De snelheid is de afgeleide van de plaats.

$$\dot{u} = v = \operatorname{Re}(-i\omega t e^{-i\omega t})$$

en

$$v = \omega \sin(\omega t)$$

De versnelling is de tweede afgeleide van de uitwijking.

$$\ddot{u} = a = \operatorname{Re}(-\omega^2 e^{-i\omega t})$$

en

$$a = -\frac{1}{\omega^2} \cos(\omega t)$$

waarin:

u = uitwijking [m].

v = snelheid [m/s].

a = versnelling [m/s²].

A = amplitude [m].

ω = frequentie* 2π [1/s]

5.4 Het gebied.

Het uitkiezen met de cursortoetsen van een te uitvergroten gebied gebeurt op de zelfde manier zoals er in de coördinatenlijst en de verbindingslijst bewogen wordt. Elke beweging van een cursortoets heeft tot gevolg dat de x-coördinaten veranderen. Door dit op het scherm te tekenen ziet men "het gebied" bewegen.

```
repeat
  hj:=readkey;           ...uitlezen toets.
  oo:=ord( readkey)+256; ...waarde toets.
  if oo=0 then           ...waarde cursortoets.
    aa:= ord( readkey)+256;
    if aa=333 then       ...naar rechts.
      begin
        x:=x+1;
        gotoxy( x, 0);
        line( x, 0, x, hoogte gebiedslijn);
        line( x+gebiedsbreedte, 0, x+gebiedsbreedte, hoogte
              gebiedslijn);
      end;
until toets is ingedrukt;
```

5.5 Het uitvergroten.

Het uitvergroten van een gebied is gewoon het vergroten van het gekozen gebied over de hele scherm breedte. Dit gebeurt door de schaal te vergroten.

```
schaalx:=( beeldafmeting in x-richting/ gebiedsbreedte);
for i:=x tot x+gebiedsgroote do
begin
  point( frq[i]*schaalx, amplitude[i]*schaaly);
  line( frq[i-1]*schaalx, amplitude[i-1]*schaaly,
        frq[i]*schaalx, amplitude[i]*schaaly);
end.
```

5.6 Het omrekenen.

De eigenfrequenties, dempingen en residuen van de metingen in het gebied zijn verschillende berekeningen. De eigenfrequentie is de grootste waarde in het gebied. De grootste waarde krijgt men zoals ook de grootste amplitude wordt gevonden bij het afbeelden. De demping en residuen zijn berekeningen voor het betreffende gebied die van hetzelfde principe zijn als de hierboven beschreven berekeningen.

HOOFDSTUK 6: Conclusies en aanbevelingen.

6.1 Suggesties.

Bij het programmeren is het aan te bevelen eerst in de literatuur na te kijken of een bepaalde procedure al niet eerder geprogrammeerd is. Scroll-procedures bijvoorbeeld zijn al meerdere keren op verschillende manieren geprogrammeerd, dus het is zonde om veel tijd en energie in een eigen scroll-programma te steken.

De volgende stap in het programma is het implementeren van de uitwijkingen van de punten uit een meting. Nu wordt met de hand de record freq opgevuld in de frequentietabel. Als gelijk in het programma de record freq opgevuld wordt vervalt de frequentietabel en dient de record freq verbonden te worden met de data die de uitwijkingen van de punten beschrijven.

Bij grotere geometrieën is het mogelijk dat de simulatie traag verloopt. Een aanbeveling is het om een snellere processor aan te schaffen of een coprocessor van het merk cyrix, die rotatiematrices standaard sneller berekent. Turbopascal ondersteunt de coprocessor niet, maar een eenvoudig programma in machinetaal maakt het toch mogelijk.

Het huidige VGA-beeldscherm met de grote van (640*480) punten is door zijn resolutie eigenlijk te onnauwkeurig. Als suggestie zou een beeldscherm van minstens (800*600) punten en 16 kleuren voor de grafische simulatie aan te bevelen zijn.

Literatuur.

- [1] Hewlett packard, User's guide 5423a structural dynamics analyzer introduction tot operation volume 1.
- [2] Hewlett packard, the fundamentals of signal analysis.
- [3] Paul R.P., Robotic manipulators: mathematics, programming and control, the M.I.T.-press, 1984.
- [4] Hisoft, Programmer's guide van Hisoft Pascal version 1.0, 1986.