

## ODIN

Undervisningsmateriale  
til et kursus i  
differentialligningsmodeller.

Projektrapport:

Tommy R. Andersen

Per H. H. Larsen

Peter H. Lassen

Vejleder:

Mogens Bruun Heefelt

# TEKSTER fra

# IMFUFA

**ROSKILDE UNIVERSITETSCENTER**  
INSTITUT FOR STUDIET AF MATEMATIK OG FYSIK SAMT DERES  
FUNKTIONER I UNDERVISNING, FORSKNING OG ANVENDELSER

FORORD.

Denne rapport er resultatet af et projekt lavet i efteråret 1979 og begyndelsen af foråret 1980. Den er skrevet af tre matematikstuderende, der alle har datalogi som deres andet fag. Formålet med projektet har været at udarbejde et undervisningsmateriale til brug ved et kursusforløb om differentiaalligningsmodeller. I forbindelse hermed ville vi udvikle og implementere et programmeringssprog, der skal gøre det lettere at opnå numeriske løsninger til differentiaalligningssystemer.

Da dette projekt kun er tænkt som et matematikprojekt, har vi valgt ikke at medtage implementeringsdokumentation i rapporten, ligesom vi på et tidspunkt valgte ikke at færdigimplementere sproget, da vi vurderede, at vores tid i matematikprojekt-hensende var bedre anvendt på færdiggørelse af kursusmaterialet.

	side
<u>FORORD</u> .....	1
<u>INDHOLDSFORTEGNELSE</u> .....	2
1 <u>INDLEDNING</u> .....	3
2 <u>MODELBEGREBET</u> .....	5
2.1 Introduktion af modelbegrebet .....	5
2.2 Hvad er en model ? .....	5
2.3 Hvordan opstilles en model ? .....	7
2.4 Hvordan anvendes en model ? .....	10
3 <u>DIFFERENTIALLIGNINGSMODELLER</u> .....	14
3.1 Differentialligninger .....	14
3.2 Differentialligningssystemer .....	15
3.3 Numeriske løsningsmetoder .....	16
4 <u>PROGRAMMERINGSSPROGET ODIN</u> .....	19
4.1 Præsentation af ODIN .....	19
4.2 Eks 1: Logistisk vækst ( $y' = ay(1-by)$ ) .....	20
4.3 Eks 2: En tretrinsraket .....	22
5 <u>MODELLERING OG MODELMODIFIKATION</u> .....	27
5.1 Eks 3: Væksten af en enkelt fisks biomasse .....	27
5.2 Eks 4: En rovdyr-byttedyr-model .....	30
5.3 Eks 5: En konkurrencemodel .....	38
5.4 Eks 6: En mini-Nordsømodel .....	42
6 <u>KONKLUSION</u> .....	54
Appendix 1: (Brugervejledning til ODIN) .....	55
Appendix 2: (Kursus beskrivelse) .....	76
Referenceliste .....	81

## 1 INDLEDNING.

Anvendelsen af modeller til brug ved løsning af forskellige problemer i vores hverdag er de seneste årtier øget voldsomt. Som følge heraf har man på mange undervisningsinstitutioner lagt større og større vægt på, at elever og studenter bliver bekendt med de muligheder, anvendelsen af modeller giver.

På RUC er der således i en årrække tilbudt de studerende på basisuddannelsen et kursus i differentiaalligningsmodeller.

Denne rapport forfattere har haft lejlighed til at deltage såvel som studenter som undervisere (hjælperere) på dette kursus og har bl.a. på denne baggrund fået inspiration til dette projekt, som omhandler udarbejdelse af et nyt materiale til dele af nævnte kursus.

Differentiaalligninger skal normalt ikke være særlig komplicerede, før det er uoverkommeligt at finde en algebraisk løsning. Det er derfor også i almindelighed nødvendigt at anvende datamaskiner, når der skal foretages simuleringer med differentiaalligningsmodeller. Imidlertid medfører dette et krav til modelløren om en datalogisk indsigt i form af kendskab til et programmeringssprog og dets anvendelse på en datamaskine.

Det er imidlertid vores opfattelse, at de programmeringssprog, som er til rådighed til dette formål, er for vanskelige at lære at anvende for en programmeringsukyndig.

I denne forbindelse kan det nævnes, at det netop er vores erfaring fra kurset, at anstrengelserne med at udføre de datalogiske simuleringer i højere grad koncentrerer sig om programmeringsmæssige problemer end om selve modelarbejdet. Med andre ord, aktivitetens formål fik karakter af blot at få et program til at køre, i stedet for at få udviklet en korrekt model og få denne afprøvet.

På denne baggrund formuleredes dette projekts målsætning: at udvikle et programmeringssprog, der gør numerisk simulering af differentiaalligningsmodeller tilgængelig for programmeringsukyndige. Da sproget altså skal være let at lære, må de udtryksmuligheder, man har i sproget, have en vis lighed med de udtryksformer, man er vant til at anvende i arbejdet med differentiaalligningsmodeller.

For at kunne udnytte dette modelværktøj hensigtsmæssigt i det omtalte kursus, vil vi endvidere udarbejde et nyt materiale til den del af kurset, der handler om simulering af differentiaalligningsmodeller på datamaskine.

Denne rapport er produktet af dette projektarbejde og indeholder derfor en beskrivelse af programmeringssproget ODIN\*, som er en videreudvikling af et sprog, der er introduceret i en tidligere RUC-rapport (14).

Selve rapporten er udformet således, at den kan udleveres med et kapitel af gangen og derved fungere som studiemateriale til dele af et kursus i differentiaalligningsmodeller. Af samme grund kan vi anbefale evt. læsere, som ikke skal anvende rapporten som studiemateriale, at læse Appendix 2 før den egentlige gennemlæsning.

Vi skal ligeledes gøre opmærksom på, at rapporten hovedsageligt er bygget op omkring en række eksempler, som er udvalgt for at vise nogle karakteristiske modeltyper. Det skal ligeledes nævnes, at vi i modeller og programmer har tilstræbt at vise strukturen i problemet og har lagt mindre vægt på estimering af parametre og konstanter. Derfor må flere af simuleringresultaterne (tabeller og kurver) tages med et vist forbehold.

\* Ordinære Differentiaalligningers Numeriske løsning.

## 2 MODELBEGREBET.

### 2.1 Introduktion af modelbegrebet.

Allerede flere hundrede år tilbage i tiden kan man finde eksempler på anvendelse af modeller. F.eks. kan det nævnes, at man, efter at Newton havde opstillet sine love for massetiltrækning, kunne formulere en "model" af solsystemet og ved hjælp af denne forudsige planeten Neptuns eksistens.

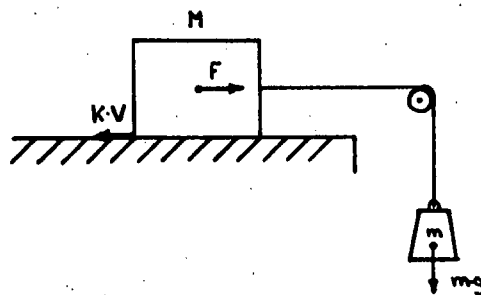
Igennem de seneste årtier er kendskabet til og anvendelsen af modeller øget voldsomt. Dette skyldes utvivlsomt, at en stigende del af de problemer, som findes i vor hverdag, har en karakter, som lettest og mest effektivt kan overskues ved hjælp af modeller. I dag er det f.eks. en selvfølge, at man i skibsindustrien laver modeller af nye skibstyper for at kunne afprøve dem i bassiner og derved kunne forudsige skibets sødygtighed og stabilitet i forskellige vind og vejrforhold.

Man kan naturligvis nævne en række andre eksempler på problemer, hvor modeller anvendes for at gøre løsningen mere overskuelig. F.eks. indenfor forurening, transport og produktion. Fælles for sådanne problemer er, at man ønsker at kende konsekvenserne af et påtænkt indgreb eller en forudset hændelse, inden de kan konstateres i virkeligheden. Et nærliggende eksempel er ønsket om at kende konsekvenserne af et uheld på et atomkraftværk. Man kan naturligvis ikke fremprovokere alle tænkelige uheld for at studere konsekvenserne, men må konstruere en model, som man så kan eksperimentere med.

Som det vil fremgå, er modeller et vigtigt værktøj ved løsning af mange af vore dages problemer. Imidlertid er det at lave modeller ikke altid en lige simpel opgave. Vi vil i dette og de efterfølgende afsnit gøre rede for, hvordan man opstiller og anvender modeller, ligesom vi vil præcisere de begreber, som indgår i dette arbejde.

### 2.2 Hvad er en model ?.

Vi vil forsøge at anskueliggøre begrebet med et eksempel og ser derfor på et mekanisk system, som vi kender fra fysikkens verden, se fig.2.1.



figur 2.1

Vi kan som bekendt opskrive kraftpåvirkningen på kassen i fig.2.1 som:

$$MV' = F = -kV + mg$$

som svarer til differentiaalligningen:

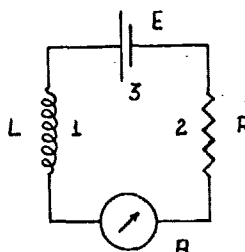
$$af'(t) = -bf(t) + c, \text{ hvor } f(t)=V(t), a=M, b=k, c=mg.$$

Lad os nu tænke os, at vi af en eller anden grund ikke kan opstille systemet, vi mangler nogle dele, eller massen af kassen er så stor, at den ikke kan være i laboratoriet. Hvis vi ønsker at undersøge eksperimentelt, hvorledes kassens hastighed ændres med tiden, må vi derfor lave en model af systemet og foretage eventuelle eksperimenter med den i stedet.

Modellen kan vi f.eks. lave af nogle elektriske komponenter, se fig.2.2

figur 2.2

1. en spole på L Henry.
2. en modstand på R Ohm.
3. et element på E volt.



Vi ved (eller tro blot på det) at :

$$E = RI + LI' \Rightarrow LI' = -RI + E$$

Vi kan altså ved at vælge  $L=M$ ,  $R=k$ ,  $E=mg$  opnå, at det elektriske system bliver en model af det mekaniske. Ved et forsøg med modellen vil amperemetret A vise en strømstyrke I, der svarer til hastigheden V, hvis vi havde lavet forsøget med vores mekaniske system.

Vi har altså fået opbygget en elektrisk model af det mekaniske system og kan nu foretage eksperimenter uden at opstille det virkelige system. Men da vi jo beskrev systemet fuldstændigt med en differentiaalligning:

$$af'(t) = -bf(t) + c, \text{ hvor } a=M, b=k, c=mg,$$

er denne ligning jo også en model af systemet. Denne form for matematisk model kalder man en differentiaalligningsmodel.

Lad os nu præcisere, hvad man skal forstå ved begrebet "en model". Hvis man ser i Gyldendals fremmedordbog, finder man, at en model er en "kopi af eksisterende ting i lille målestok".

Denne tolkning af begrebet er da også brugbar i mange af dagligdagens situationer, tænk f.eks. på modelflyvere og modeljernbaner. Men til vores brug er den ikke præcis nok. F.eks. vil en model af et molekyle jo være i "stor" målestok. Vi må derfor opstille en mere præcis definition.

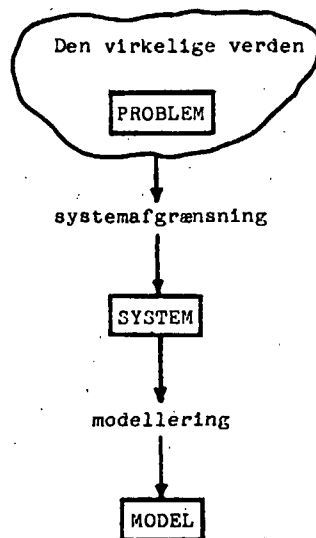
- En model er en til et ganske bestemt formål beskrevet simplificeret sammenhængsbeskrivelse af en del af et virkeligt system.
- Med et system menes her den del af den virkelige verden, som er genstand for vores interesse.

### 2.3 Hvordan opstilles en model?

Vi vil her beskrive den proces, hvor man opstiller en model til hjælp ved løsning af et givet problem. Vi vil i det følgende kalde denne proces for modelleringsprocessen.

Lad os beskrive modelleringsprocessen ved først at opstille et diagram over den og ud fra dette diskutere de begreber og handlinger, som indgår.

Den første del af modelleringsprocessen kan skitseres som følger:



#### Den virkelige verden med problemer:

Den virkelige verden er selvfølgelig den verden, som omgiver os. Det er her de problemer, som har vores interesse, findes. De kan være af videnskabelig art, - hvordan er et DNA-molekyle opbygget ? -, af miljømæssig art, - hvordan undgår vi en forurening i en given sø ? -, eller det kan være et produktionsproblem, - hvordan bygger vi et skib, som bruger mindst mulig brændstof for at transportere en given last ? -.



Der findes selvfølgelig en række problemer, hvor det ikke nødvendigvis er en hjælp at anvende modeller. Behandlingen af disse har vi fundet uinteressant i denne sammenhæng. At vi i det følgende udelukkende beskæftiger os med problemer, hvor modeller er en oplagt hjælp, skal således ikke tages som udtryk for, at alle problemer kan løses med brug af modeller.

#### Systemafgrænsning.

Denne del af modelleringsprocessen har naturligvis som mål at beskrive et system, som kan være udgangspunkt for vores model. Det er imidlertid yderst vanskeligt at give generelle retningslinier for denne afgrænsning, fordi det ofte ikke er en særlig veldefineret proces, og fordi det ofte blot er en proces, som foregår i modellørens hoved. Vi har derfor valgt at beskrive systemafgrænsningen ved at præcisere dens formål.

Formålet er at "indse", hvilke sammenhænge vores problem findes i. Sagt på en anden måde, vores system skal afgrænses sådan, at det består af alle de ting og faktorer, som har indflydelse på vores problem. Systemafgrænsningen er altså den proces, hvor man indser hvilke dele af den virkelige verden, der bør indgå i vores system.

#### System.

Vi har tidligere defineret begrebet "system"; men lad os forsøge at udbyde det en smule.

Vores system er, som det er fremgået, den del af den virkelige verden, som har indflydelse på vores problem. Sandsynligvis kender vi ikke den fuldstændige sammenhæng i systemet; men vi har indset, at de ting og faktorer, som systemet består af, har indflydelse på vores problem og dets løsning.

#### Modellering og model.

Lad os begynde med at definere nogle begreber:

- Modellering er den proces, hvor vi skaber vores model. Det kan f.eks. være at opstille nogle differentialligninger, eller det kan være at snitte et skib i et stykke træ til senere afprøvning i et vandbassin.
- Model er den genstand, vi har frembragt ved modelleringen, og som vi senere skal gøre eksperimenter med. (Vi har tidligere givet en præcis definition side 7 øverst.)
- Struktur og adfærd.  
Når vi taler om modellens eller systemets struktur, mener vi de sammenhænge, som findes mellem de enkelte faktorer. Adfærden er den måde, systemet eller modellen opfører sig på, det man kan se og måle. (en forurening, skibet gynger etc.).

Vi er nu kommet så vidt i modelleringsprocessen, at vi skal til at opstille (modellere) en model af vores system.

Det er oplagt, at udformningen af vores model afhænger af det problem, som vi ønsker at løse. Skal vi finde frem til en form på et skibsskrog, som er stabilt i søgang, kan vi f.eks. konstruere et modelskib, hvor vi kan ændre skrogets form fra eksperiment til eksperiment. Man kan således normalt ikke forvente, at en model konstrueret til et formål med held kan anvendes i andre problemløsningsituationer.

Hvad vi må gøre er at analysere vores system, således at vi kan opstille en model, som i videst muligt omfang opfylder følgende 3 krav:

- Modellen afspejler den struktur, som findes i systemet.
- Modellen har tilnærmelsesvis samme adfærd som systemet.
- Modellen er udformet således, at man kan drage nogle konklusioner fra den, der kan bruges til løsning af problemet.

Det er imidlertid også klart, at modellen skal være en simplifikation af systemet, hvorfor uvæsentlige detaljer må søges udeladt (modelskibet behøver f.eks. ikke at være malet eller have små vinduer).

Vores analyse af systemet må derfor munde ud i en opdeling af systemets faktorer i tre grupper:

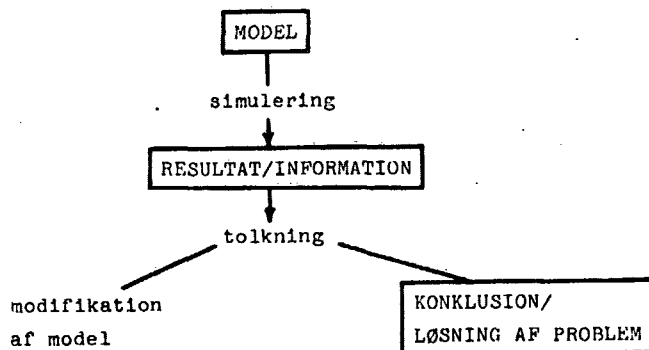
- 1.- ting, hvis effekt modelløren finder er ubetydelige, og som kan udelades i modellen.  
Vi har tidligere givet eksempler på sådanne faktorer, f.eks. vinduer i modelskibet.
- 2.- ting, som har interesse, og som modellen er designet til at studere. Denne type faktorer kaldes endogene variable (endogen = indre), output-variable eller afhængige variable. Karakteristika for sådanne variable er, at de væsentlig påvirker vores problem, men at mennesket af den ene eller anden grund ikke kan regulere på dem, selv om det måske ville løse vores problem.  
Som eksempel på sådanne endogene variable kan fra vores skibseksempel nævnes skibets rulning, som vi netop ønsker mindsket.
- 3.- ting, som påvirker systemet, og som skal med i modellen, men hvis adfærd modellen ikke er designet til at studere.  
Denne type faktorer kaldes i reglen exogene variable (exogen = ydre), uafhængige variable eller parametre. Karakteristika for denne type variable er, at vi kan regulere dem og derved finde en løsning på vores problem.  
Hvis vores problem f.eks. igen er at finde frem til et skib, der er stabilt, kan eksempler på exogene variable være skibets højde, vægt, form o.s.v., alle faktorer vi kan ændre på. Ideen er jo netop at blive ved med at regulere modellens exogene variable og observere de endogene variable, indtil vi har fundet en tilfredsstillende løsning på vores problem.

Når denne opdeling har fundet sted, skal sammenhængen mellem de forskellige variable klarlægges. (hvordan gynger et skib med en given sejlretning, som funktion af vindretningen). Man skal være opmærksom på, at problemet, man ønsker at løse, netop kan være at finde frem til en sammenhæng mellem nogle variable.

Selve valget af hvilken type model, man ønsker at arbejde med (træmodel, elektrisk kredsløb, differentiålligningsmodel), må man træffe sideløbende med, at man afgrænser systemet og finder ud af hvilke typer faktorer, der er væsentlige i modellen.

#### 2.4 Hvordan anvendes en model ?.

Vi vil på lignende måde som i foregående afsnit skitsere et diagram og ud fra dette forsøge at beskrive den del af modelleringsprocessen og de begreber, som optræder ved arbejdet med modeller.



#### Simulering og resultat.

Hensigten med at opstille en model er naturligvis at kunne udføre eksperimenter med den. Disse eksperimenter kan opfattes som simuleringer af hændelser i det virkelige system, hvorfor vi fremover vil kalde et eksperiment med en model for en simulering.

En simulering har naturligvis kun interesse, hvis man får et resultat ud af den. Et resultat kan formidles på mange måder. Det kan f.eks. i vores skibseksempel blot være en iagttagelse af, om modelskibet vælter eller ej, når vi sætter det ned i et vandbassin med bølger, eller resultatet kan være i form af udslag på nogle måleinstrumenter. Det er klart, at den form, resultatet skal fremtræde i, skal vælges allerede ved opstillingen af modellen. I det indledende eksempel, hvor vi opstillede et elektrisk kredsløb, hvor strømmen I skulle repræsentere hastigheden  $V$  i det mekaniske system, vidste vi, allerede inden vi opstillede modellen, at vi skulle benytte et amperemeter til at

"se" resultatet af simuleringen. Til simulering af differentiaalligninger anvendes ofte datamaskiner, hvorfor det er naturligt at få resultatet af en simulering udskrevet på et stykke papir, f.eks. i form af kurver eller tabeller. Vi vil i de følgende kapitler komme nærmere ind på dette.

Det er imidlertid ikke nok at beslutte, på hvilken form resultatet skal fremtræde. Man må naturligvis også tage stilling til, hvad det er for information, man ønsker at få ud af en simulering. Det, vi ønsker, er selvfølgelig information, som gør os i stand til at løse vores oprindelige problem. Lidt mere præcist kan vi sige, at den søgte information i reglen deler sig i to kategorier:

1.- Adfærdsinformation.

Information, som kan benyttes til at forudsige systemets udvikling frem i tiden.

2.- Strukturinformation.

Information, som kan benyttes til at forstå nogle sammenhænge i systemet.

Vi kan af de to kategorier se, at resultaterne af en simulering ikke altid skal give information om modellens adfærd, men også om modellens struktur (de kausale sammenhænge). Vi må derfor, allerede når vi opstiller vores model, designe resultaternes form således, at vi får den nødvendige information til løsning af vores problem.

Tolkning af resultat.

Når vi har udført en simulering og fået et resultat, må vi overveje dets brugbarhed og dets troværdighed. Med andre ord, hvordan kan vi være sikre på, at det resultat, vi har fået ved simuleringen, faktisk kan bruges til løsning af vores problem. Var modellen en nøjagtig kopi af systemet, ville resultatet naturligvis være rigtigt, såfremt simuleringen var præcis; men modellen er jo en simplificering af systemet. Så hvordan kan vi, på trods af denne simplificering, tro på resultatet?

Hvad vi kan gøre er, at vurdere modellens evne til at producere brugbare resultater, eller sagt på en anden måde, vurdere hvor god en efterligning modellen er. Finder vi, at modellen har en rimelig god validitet, vil vi tro på resultatet. Ellers må vi ændre på modellen. (modelmodifikation).

Denne modelmodifikation vil bestå i en mere eller mindre gennemgribende ændring af nogle af de præmisser, der ligger til grund for den aktuelle simulering. Der kan være tale om tildeling af nogle andre startværdier for de exogene variable, altså om andre parameterværdier, eller der kan være tale om en ændring af selve modellen, f.eks. at gøre en simplificering mindre grel. Endelig kan der være brug for at tage systemafgrænsningen op til revision, idet man måske har udelukket en faktor med signifikant betydning.

En vurdering af en models validitet kan foretages ved at rubricere den i følgende validitetskategorier:

1.- Forventningsvaliditet.

Modellens adfærd er i rimelig overensstemmelse med, hvad der er forventet (af modellen eller af systemet).

2.- Reproduktionsvaliditet.

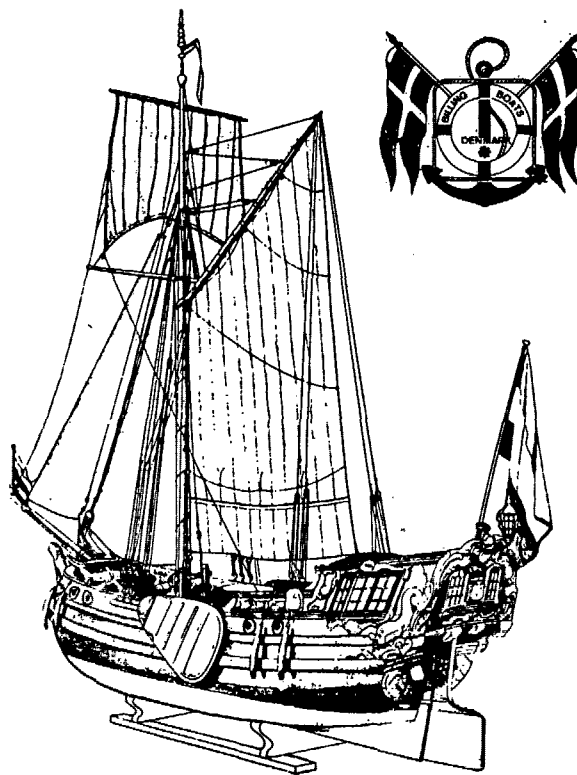
Modellen kan reproducere data, som er kendt fra systemet (data er kendt ved opstilling af modellen).

3.- Forudsigelsesvaliditet.

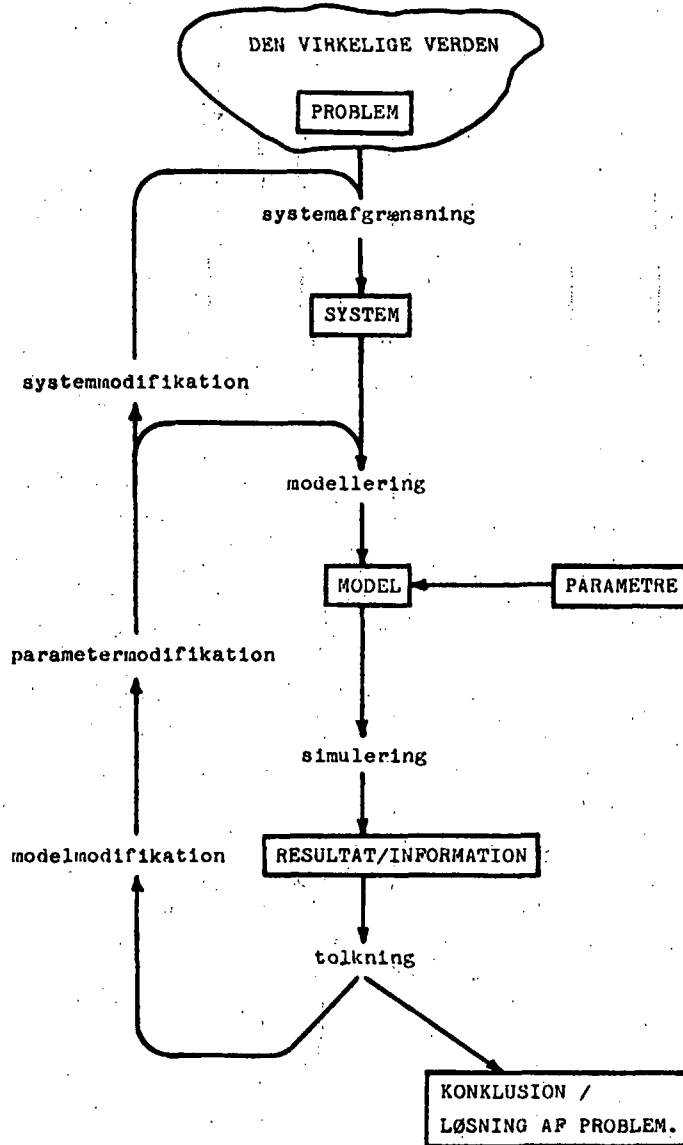
Modellen opfylder 1. og 2. og kan forudsige data, som kan findes i systemet. (Modellen er opbygget uden kendskab eller uden udnyttelse af kendskabet til disse data).

4.- Strukturvaliditet.

Udover at opfylde 1., 2. og 3. er modellens struktur en afspejling af systemets struktur.



Vi kan nu opstille det fuldstændige diagram over modelleringsprocessen:



### 3 DIFFERENTIALLIGNINGSMODELLER.

#### 3.1 Differentialligninger.

Vi har set, at man kan beskrive dele af virkeligheden ved hjælp af forskellige typer modeller, hvor det specielt er matematiske modeller, der har interesse, når kvantitative egenskaber ved den modellerede del af virkeligheden skal undersøges. I det følgende vil vi udelukkende beskæftige os med en bestemt type matematiske modeller, nemlig differentiaalligningsmodeller, der kaldes således, fordi de søger at beskrive nogle egenskaber ved en afgrænset del af virkeligheden ved hjælp af et system af differentiaalligninger.

Lad os kort repetere, hvad vi ved om differentiaalligninger:

Man kan udtrykke det forhold, at mængden af et radioaktivt stof,  $X$ , aftager (henfalder) med en hastighed, der er proportional med den resterende mængde af stoffet, ved en 1. ordens differentiaalligning:

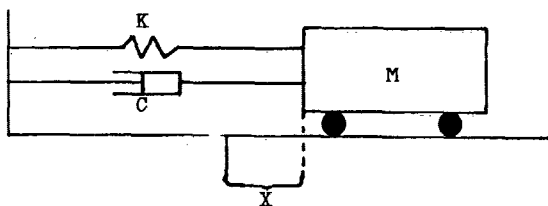
$$X' = -K X$$

hvor  $K$  er en konstant, den såkaldte henfaldskonstant.

Man kan på tilsvarende måde beskrive et system bestående af en masse,  $M$ , der bevæger sig retlinjet frem og tilbage under påvirkning af en fjeder og en dæmper, se fig. 3.1, ved en 2. ordens differentiaalligning:

$$X'' = -\frac{C}{M}X' - \frac{K}{M}X$$

der udtrykker, at kraftpåvirkningen fra dæmperen,  $-CX'$ , er modsat rettet og proportional med hastigheden, og at kraftpåvirkningen fra fjederen er modsat rettet og proportional med afstanden,  $-KX$ .



$$\begin{aligned} X(0) &= 1.0 \\ X'(0) &= 0.0 \\ M &= 1.5 \\ C &= 4.0 \\ K &= 150 \end{aligned}$$

Figur 3.1.

Vi har tidligere set, hvorledes vi kan løse ligninger af disse typer, idet vi ved en løsning til f. eks. ligningen  $X' = -K X$  forstår en funktion  $X$ , der opfylder ligningen. I dette tilfælde  $X = Ce^{-Kt}$ . Vi vil ikke i det følgende beskæftige os med denne slags løsninger, såkaldte analytiske løsninger. Vi vil derimod vise, hvordan man kan opnå en såkaldt numerisk løsning til en differentiallyigning eller et system af differentiallyigninger. En sådan løsning vil bestå af en tabel eller en kurve, der viser den søgte funktions værdier til tilsvarende værdier for den uafhængige variabel, eller mere generelt en kurve visende sammenhængen mellem to vilkårlige funktioner, der kan beregnes ud fra systemets øjeblikkelige tilstand. En sådan numerisk løsning kræver bl.a., at man kender systemets tilstand fuldstændigt til en bestemt værdi af den uafhængige variabel, starttilstanden, hvorefter man kan beregne systemets tilstand til enhver anden værdi af den uafhængige variabel.

### 3.2 Differentiallyigningssystemer.

Vi vil nu vise, hvad vi forstår ved et system af differentiallyigninger. Hvis et radioaktivt stof henfalder til et andet radioaktivt stof med henfaldskonstanten  $\lambda$ , og dette andet stof henfalder til et stabilt stof med henfaldskonstanten  $\mu$ , kan vi, hvis mængden af det første stof til tiden  $t$  kaldes  $f(t)$  og mængden af det andet  $g(t)$ , beskrive udviklingen i henfaldet med et system af to koblede 1. ordens differentiallyigninger:

$$f'(t) = -\lambda f(t)$$

$$g'(t) = \lambda f(t) - \mu g(t)$$

Vi vil tillige vise, at en 2. ordens differentiallyigning kan omskrives til et system af to koblede 1. ordens differentiallyigninger. Formålet hermed er, at man så kun behøver at beskæftige sig med løsninger til systemer af 1. ordens differentiallyigninger.

Hvis vi har en vilkårlig 2. ordens differentiallyigning på reduceret form:

$$X'' = F(T, X, X'), \text{ hvor } F \text{ er en funktion af } T, X \text{ og } X',$$

kan vi indføre en ny funktion  $Y = X'$ , og altså  $Y' = X''$ , hvorved vi får det koblede 1. ordens differentiallyigningssystem:

$$X' = Y$$

$$Y' = F(T, X, Y).$$

Idet det tilsvarende kan vises for ligninger af vilkårlig høj orden, kan vi altså nøjes med at udvikle metoder til behandling af systemer af koblede 1. ordens differentiallyigninger.



I de differentiaalligninger, vi indtil nu har set, har kun indgået reelle funktioner. Hvis vi i stedet lader de indgående funktioner være vektorfunktioner, kan vi skrive et 1. ordens differentiaalligningssystem som:

$$\underline{X}' = \underline{G}(T, \underline{X}) .$$

Vi kan her se den førnævnte 2. ordens ligning som et specialtilfælde, nemlig ved at sætte

$$\underline{X} = (X, Y) \text{ og} \\ \underline{G}(T, \underline{X}) = (Y, F(T, X, Y)) ,$$

hvor den reelle funktion  $F$  er den fra 2. ordensligningen.

### 3.3 Numeriske løsningsmetoder.

Vi vil kort forsøge at vise, hvorledes det er muligt udfra kendskab til en starttilstand at beregne systemets tilstand til andre værdier af den uafhængige variabel, når vektorfunktionen  $\underline{G}$  i ligningen  $\underline{X}' = \underline{G}(T, \underline{X})$  opfylder visse betingelser. Disse betingelser kan kort udtrykkes som:

Hvis  $\underline{G}(T, \underline{X})$  er kontinuert i det interessante værdiområde,  $V$ , for  $(T, \underline{X})$ , og hvis  $\underline{G}(T, \underline{X})$  har kontinuerte partielle afledede (m.h.t.  $\underline{X}$ ) i  $V$ , så findes der en og kun en løsning  $\underline{f}$ , for hvilken  $\underline{f}(T_0) = \underline{X}_0$ .  
(Når  $(T_0, \underline{X}_0)$  ligger i  $V$ ).

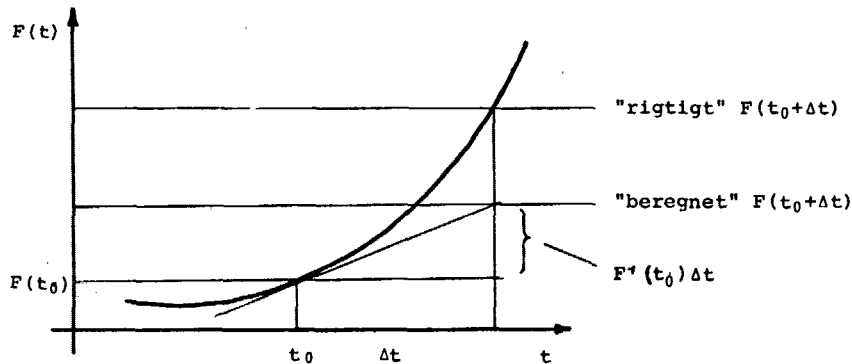
Denne formulering af eksistens-og entydighedssætningen giver os altså mulighed for at vurdere, hvornår vi kan være sikre på, at den løsning, vi finder udfra en starttilstand  $(T_0, \underline{X}_0)$ , er den eneste mulige (og eksisterer).

Den første metode, vi vil vise, kaldes EULER-integration, og går i al sin enkelhed ud på, at man opfatter løsningskurvens forløb som sammensat af liniestykker, der har samme hældning som den integrerede funktion i liniestykkets ene endepunkt.

Altså:

figur 3.2

$$F(t_0 + \Delta t) = F(t_0) + F'(t_0)\Delta t$$



Vi kan altså se, at metoden ikke giver en eksakt løsning til ligningen, idet der er forskel på den beregnede og den rigtige værdi af  $F(t_0+\Delta t)$ , og vi kan samtidig se, at små værdier for  $\Delta t$  vil gøre denne fejl lille.

Ved at foretage en Taylorudvikling, med restled, udfra  $T_0$  fås:

$$F(T_0+\Delta T) = F(T_0) + F'(T_0)\Delta T + F''(T_0+\theta\Delta T)\Delta T^2, \text{ hvor } \theta \in [0,1]$$

Fejlen ved vores metode bliver altså som restledet i Taylorrækken:  $F''(T_0+\theta\Delta T)\Delta T^2$ . Den begåede fejl er altså proportional med  $\Delta T^2$ .

Denne metode kan forbedres, forstået på den måde, at man får en mindre fejl for samme valg af  $\Delta t$ , ved også at beregne hældningen i  $(t_0+\Delta t)$ ,  $F_1'(t_0+\Delta t)$ , altså nær det punkt man gerne skulle ende i, hvor  $F_1$  er beregnet kun ved brug af hældningen i startpunktet. Man foretager så den endelige beregning af  $F(t_0+\Delta t)$  ved at tage et nyt skridt udfra  $(t_0, F(t_0))$ ; men med gennemsnittet af de to beregnede hældninger som ny hældning.

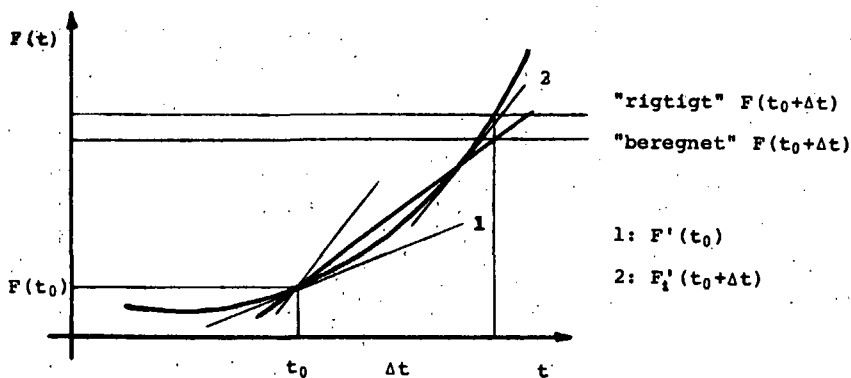
Givet differentiaalligningen  $F' = G(F, t)$ , foretager man altså følgende beregninger:

$$F'(t_0) = G(F, t_0)$$

$$F_1(t_0+\Delta t) = F(t_0) + F'(t_0)\Delta t$$

$$F_1'(t_0+\Delta t) = G(F_1(t_0+\Delta t), t_0+\Delta t)$$

$$\begin{aligned} F(t_0+\Delta t) &= F(t_0) + \frac{(F'(t_0) + F_1'(t_0+\Delta t))\Delta t}{2} \\ &= F(t_0) + (G(F(t_0), t_0) + G(F(t_0) + G(F(t_0), t_0)\Delta t, t_0))\frac{\Delta t}{2} \end{aligned}$$



$$\text{"beregnet"} : F(t_0) + \frac{(F'(t_0) + F_1'(t_0+\Delta t))\Delta t}{2}$$

I denne metode, der kaldes FORBEDRET EULER, kan den begåede fejl ved betragtninger som for EULER-metoden ses at være proportional med  $(\Delta T)^3$ .

Man kan forbedre disse metoder yderligere, og den metode, der bliver anvendt senere i dette materiale, er en metode, som udover at medtage flere led i beregningen af hældningen for det enkelte skridt, samtidigt giver mulighed for at vurdere, hvor store fejl, der er begået i det enkelte skridt. Metoden er sådan udformet, at der automatisk tages mindre skridt, hvis fejlen bliver for stor, mens skridtlængden øges, hvis den begåede fejl bliver betydelig mindre, end man kan nøjes med.

Metoden går frem i fire skridt af længden  $DT/4 = H/2$ , idet den foretager 9 beregninger af  $G(F(T), T)$ . Til beregning af  $F(T_0+DT)$  foretages følgende udregninger:

$$\begin{aligned} a_1 &= HG(F(T_0), T_0), \\ a_2 &= HG(F(T_0) + 0.5a_1, T_0 + 0.5H) \\ a_3 &= HG(F(T_0) + 0.25(a_1 + a_2), T_0 + 0.5H) \\ a_4 &= HG(F(T_0) - a_2 + 2a_3, T_0 + H) \\ F(T_0 + H) &= F(T_0) + (a_1 + 4a_3 + a_4)/6 \\ a_5 &= HG(F(T_0 + H), T_0 + H) \\ a_6 &= HG(F(T_0 + H) + 0.5a_5, T_0 + 1.5H) \\ a_7 &= HG(F(T_0 + H) + 0.25(a_5 + a_6), T_0 + 1.5H) . \end{aligned}$$

På dette punkt kan den i dette skridt maksimalt begåede fejl beregnes som:

$$E = (-a_1 + 4a_3 + 17a_4 - 23a_5 + 4a_7 - b)/90 + O(H^6),$$

hvor b beregnes som:

$$b = HG(F(T_0) + (-a_1 - 96a_2 + 92a_3 - 121a_4 + 144a_5 + 6a_6 - 12a_7)/6, T_0 + DT).$$

Hvis  $E \leq (\text{maksimalabsolutfejl}) + (\text{maksimalrelativfejl}) \cdot |F(T_0+H)|$ , for alle elementer i vektoren, er løsningen sket med den ønskede nøjagtighed. I så fald beregnes:

$$F(T_0 + DT) = F(T_0 + H) + (a_5 + 4a_7 + HG(F(T_0 + H) - a_6 + 2a_7, T_0 + DT))/6.$$

I modsat fald halveres  $DT$ , og skridtet tages forfra. Såfremt  $E$  er mindre end 2% af den tilladte fejl, fordobles skridtlængden  $DT$ .

Denne metode, som kaldes RUNGE-KUTTA-ENGLAND, betragtes i almindelighed som en god standard metode. Den begåede fejl kan ved denne metode vises at være proportional med  $(DT)^5$ .

#### 4 PROGRAMMERINGSSPROGET ODIN.

##### 4.1 Præsentation af ODIN.

Når man har opstillet en differentiallyigningsmodel (i den følgende tekst omtalt som model(len)), indset at en analytisk løsning enten er utilstrækkelig eller for krævende/umulig at opnå, og at en numerisk løsning vil være af værdi, så er man nået til et stadium, hvor det vil være rimeligt at overveje, om en datamaskine kan være en hjælp ved tilvejebringelsen af denne løsning. Det kræver imidlertid, for det første, at opgaven har et omfang, der i det hele taget berettiger til andet og mere end en manuel udregning, og, for det andet, at man har mulighed for at få adgang til en datamaskine, der råder over et programmeringssprog, som er så hensigtsmæssigt udformet, at arbejdet med at beskrive modellen i dette sprog ikke antager proportioner, som er urimelige i forhold til arbejdet med opstilling af modellen og værdien af at opnå den numeriske løsning. Hvad der er et hensigtsmæssigt sprog i så henseende afhænger naturligvis, til en vis grad, af programmørens erfaring og øvrige kvalifikationer, men enhver med blot den mindste erfaring i anvendelse af mere generelle programmeringssprog (BASIC, FORTRAN og ALGOL etc.) vil utvivlsomt medgive, at disse sprog ikke er et velegnet værktøj ved løsning af denne type problemer.

Derfor har mange simuleringssprog set dagens lys inden for de sidste 10-15 år. De fleste om ikke alle, er imidlertid ikke beregnet for programmeringsukyndige. De er enten baserede på et af de generelle programmeringssprog, eller også er de opbygget, så de ligner de generelle; men da de problemer, man ønsker løst, er af en ganske bestemt, velafgrænset type, er disse lighedspunkter med de mere generelle programmeringssprog ganske overflødige, da de kun gør sproget vanskeligere at lære og anvende.

Det er bl.a. derfor, at vi har søgt at lave et sprog, ODIN, som gør det muligt at opnå numeriske løsninger til differentiallyigninger, og som er designet med særligt henblik på, at det skal være let at forstå og anvende for, også, programmeringsukyndige.

For at gøre sproget let at lære og anvende, er det blevet udformet, så de udtryksmuligheder, som man har i sproget, har en lighed med de udtryksformer, som man er vant til at benytte i forvejen i forbindelse med denne type arbejde. I dette tilfælde den sædvanlige matematiske terminologi som man benytter, når man skriver en differentiallyigning. Sproget er desuden opdelt i de afsnit eller sektioner, som man under alle omstændigheder, mere eller mindre underforstået, altid deler en simulering op i. Det er dele som de indgående ligninger, tildeling af startværdier, rent praktiske simuleringdetaljer som simuleringsperiode og tidsskridtenes længde etc. samt angivelse af, hvad man vil have skrevet ud.

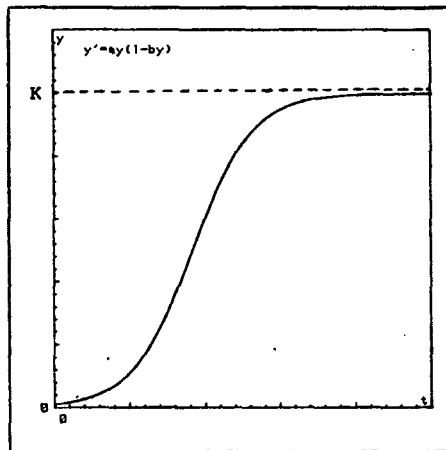
Lad os i et par eksempler se på, hvordan man benytter programmeringssproget ODIN til at beskrive en given model, så den kan simuleres af en datamaskine. For en mere omfattende og fyldig beskrivelse henvises til brugervejledningen, appendix 1.

#### 4.2 Eks. 1: Logistisk vækst ( $y' = ay(1-by)$ ).

Væksten af en population, i dette tilfælde kaniner, kan, hvis visse ikke her nærmere omtalte betingelser er opfyldt, beskrives ved flg. 1.ordens differenttialligning:

$$\text{kanin}' = \text{vækstrate} * \text{kanin} * (1 - \text{kanin}/K) \quad , \text{hvor}$$

vækstraten er fødselsrate minus dødsrate, og K er et hængningsled, der er udtryk for, at fertiliteten går ned. K kaldes bærekapaciteten, og det er den øvre grænse for populationens størrelse (se fig. 4.1).



figur 4.1

logistisk vækstkurve,  
f.eks. for en popula-  
tion kaniner.

Systemet beskrives altså af førømtalte differenttialligning, og i ODIN skrives denne ligning i en EQUATIONS -sektion (se program 4.1).

Før vi kan starte en simulering (finde en numerisk løsning), må vi have bestemt nogle startværdier, f.eks. hvor stor en population af kaniner vi ønsker at starte med, og givet nogle konstanter værdi, f.eks. vækstraten og K. Disse oplysninger angives i en ( eller flere ) sektioner med overskriften INITIAL. I program 4.1 bliver startpopulationen (kanin) sat til 10 og vækstrate til 1.5 og K til 100.

```
EQUATIONS
KANIN' = VEKSTRATE * KANIN * ( 1 - KANIN/K )

INITIAL
KANIN = 10   VEKSTRATE = 1.5   K = 100

SIMULATION
SIMTIME = 20   DTMIN = 0.001   DTMAX = 1
STOPCONDITION = KANIN <= 1

OUTPUT
PRINT(1) T, KANIN, KANIN'
PLOT(T,1) KANIN AS K

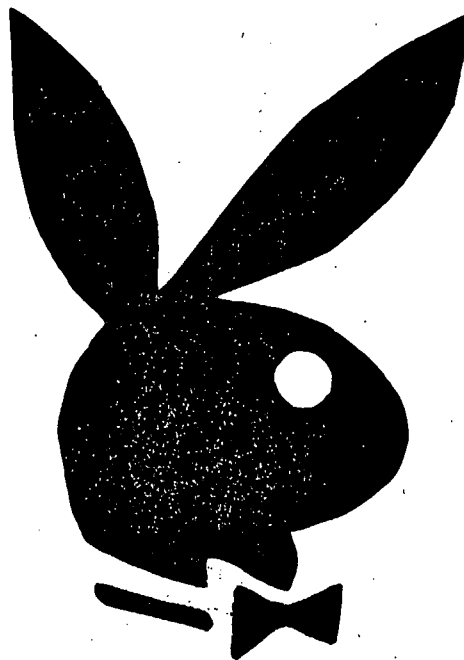
RUN

STOP
```

Program 4.1

Hvis man gerne vil se et resultat af sine anstrengelser, kan man i en OUTPUT-sektion angive, hvad man vil have skrevet ud og på hvilken form (tabeller (PRINT) og/eller kurver(PLOT)).

I en SIMULATION-sektion angives det, der har med selve simuleringen at gøre. Der kan f.eks. angives simuleringstiden (SIMTIME), grænser for tidsskridtenes størrelse (DTMIN og DTMAX) og den numeriske løsnings nøjagtighed (MAXABSERR og MAXRELERR) samt en eller flere betingelser, som, hvis de bliver opfyldt, bevirker, at programudførelsen stoppes, inden simuleringstiden udløber (STOPCONDITION). RUN er den kommando, der forårsager, at simuleringen udføres (een gang!), og STOP markerer, at der ikke skal gøres mere, og afslutter derfor programmet.



#### 4.3 Eks.2: En tretrinsraket.

Vi vil nu se på endnu en anvendelse af sproget, hvor der udnyttes lidt flere af sprogets faciliteter, og hvor systemet er lidt mere kompliceret.

Ved mange simuleringer kan det ikke lade sig gøre at beskrive en genstands bevægelse i rummet ved den samme differentiaalligning under hele simuleringsperioden. Tænk f.eks. blot på:

- en jetjager med eller uden fungerende efterbrænder,
- et automobil med registerkarburator (eller turbo),
- en faldskærmsudspringer, når skærmen folder sig ud,

eller en tretrins raket, som vi i det følgende vil opskrive bevægelsesligningen for. Og når vi har gjort det, vil vi skrive et ODIN-program, som kan give os en numerisk løsning.

Fælles ved alle disse fænomener er altså, at der sker en diskret hændelse i forbindelse med, at bevægelsesligningen bliver erstattet af en anden.

Hvis "alt" (kort for altitude, der betyder højde) er raketens højde over jorden, kan dens bevægelse, ifølge Newtons 2.lov, beskrives ved følgende differentiaalligning:

$$m \cdot \text{alt}'' = f \cdot V_0 - m \cdot g - \frac{1}{2} \rho \cdot C_d A \cdot (\text{alt}')^2 \quad \text{eller}$$

$$\text{alt}'' = \frac{f \cdot V_0 - \frac{1}{2} \rho \cdot C_d A \cdot (\text{alt}')^2}{m} - g$$

, hvor

$m$	= raketens vægt	MASS
$\text{alt}''$	= raketens acceleration	ALT''
$\text{alt}'$	= raketens hastighed	ALT'
$f$	= brændstofforbrug/sek. (letter raketten for vægt)	FLOW
$V_0$	= udstødningsgassens hastighed	VOUT
$g$	= tyngdeaccelerationen, som fås af	G
	$g = 32.17 \left( \frac{r_e}{r_e + \text{alt}} \right)^2$	, hvor
$r_e$	= jordens radius	RADERH
$\rho$	= lufttæthed, som fås af	RO
	$\rho = 0.00238 e^{-\text{alt}/24000}$	
$C_d A$	= raketens tværsnitsareal * luftmodstandskoefficient	DRAG

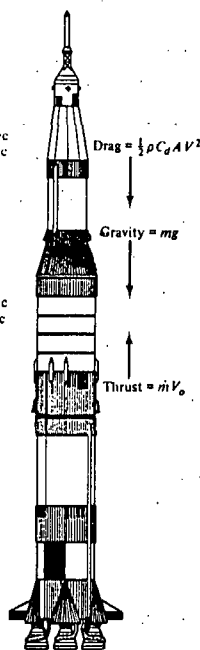
Hvis vi betragter den første ligning, kan vi se, at den kraft, der påvirker raketten bort fra jorden, er rakettsens udstødningsgas,  $f \cdot V_0$  og den kraft, der virker i den modsatte retning, består af dels tyngdekraften,  $m \cdot g$ , og modstanden,  $\frac{1}{2} \rho C_D A (\text{alt}')^2$ , (yderligere klarifikation kan muligvis opnås ved at betragte tegningen af Saturn-raketten).

Konstanterne MASS, FLOW, VOUT og DRAG er afhængige af, hvilket trin af raketten, som er ved at blive brændt af. Derfor skriver vi MASS1 for rakettsens 1. trins vægt, FLOW1 for forbruget ved afbrænding af trin 1, e.t.c. Og hvis BURNT1 og BURNT2 på samme måde angiver brændetiden for trin 1 og 2, kan før-omtalte differentiaalligning nu skrives således, idet T er tiden:

$$\text{ALT}' = \begin{cases} \frac{\text{FLOW1} \cdot \text{VOUT1} - 0.5 \cdot \text{DRAG1} \cdot \text{RO} \cdot \text{ALT}'^2}{\text{MASS1} + \text{MASS2} + \text{MASS3} - \text{FLOW1} \cdot \text{T}} - G & \text{for } T \leq \text{BURNT1} \\ \frac{\text{FLOW2} \cdot \text{VOUT2} - 0.5 \cdot \text{DRAG2} \cdot \text{RO} \cdot \text{ALT}'^2}{\text{MASS2} + \text{MASS3} - \text{FLOW2} \cdot (\text{T} - \text{BURNT1})} - G & \text{for } \text{BURNT1} < T < \text{BURNT1} + \text{BURNT2} \\ \frac{\text{FLOW3} \cdot \text{VOUT3} - 0.5 \cdot \text{DRAG3} \cdot \text{RO} \cdot \text{ALT}'^2}{\text{MASS3} - \text{FLOW3} \cdot (\text{T} - \text{BURNT1} - \text{BURNT2})} - G & \text{for } T > \text{BURNT1} + \text{BURNT2} \end{cases}$$

, og med de følgende begyndelsesværdier:

CSMP Symbol	Third stage
MASS3	Initial mass = 8137 slugs
VOUT3	Outflow velocity = 15250 ft/sec
FLOW3	Mass flow rate = 14.75 slugs/sec
BURNT3	Burn time = 479 sec
DRAG3	Drag coefficient times cross-sectional area = 360 sq ft
	Second stage
MASS2	Initial mass = 32205 slugs
VOUT2	Outflow velocity = 13805 ft/sec
FLOW2	Mass flow rate = 81.49 slugs/sec
BURNT2	Burn time = 359 sec
DRAG2	Drag coefficient times cross-sectional area = 460 sq ft
	First stage
MASS1	Initial mass = 148820 slugs
VOUT1	Outflow velocity = 8060 ft/sec
FLOW1	Mass flow rate = 930 slugs/sec
BURNT1	Burn time = 150 sec
DRAG1	Drag coefficient times cross-sectional area = 510 sq ft





Udover de allerede omtalte sektioner EQUATIONS, INITIAL, SIMULATION og OUTPUT omfatter ODIN endnu en, nemlig DEFINITIONS. I DEFINITIONS kan man definere sine egne funktioner. Hvis vi vender tilbage til vores raket, kan vi se, at tælleren i udtrykket for ALT er den samme i de 3 ligninger herover, og da det bl.a. vil få programmet til at fremstå enklere og mere overskueligt, har vi valgt, i en DEFINITIONS-sektion, at definere en funktion, F, som en funktion af FLOW, VOUT og DRAG for det pågældende rakettrin ( se program 4.2 herunder). Det betyder, at hver gang F forekommer, beregnes dens værdi ud fra udtrykket i definitionen.

Funktionerne behøver ikke at have parametre (her FLOW, VOUT og DRAG), og man kan også definere funktioner, som kun kendes i nogle punkter (tabellagte), men alt dette beskrives mere indgående i brugervejledningen for ODIN.

Det færdige raketsimuleringsprogram kan evt. se sådan ud:

Program 4.2

```
EQUATIONS
G = 32.17*((RADERH/(RADERH+ALT))**2)
RO = 0.00238*EXP(-ALT/24000.0)
ALT"= IF T<=BURNT1
      THEN (F(FLOW1,VOUT1,DRAG1)/(MASS1+MASS2+MASS3-FLOW1*T)-G)
      ELSE (IF T < BURNT1+BURNT2
            THEN (F(FLOW2,VOUT2,DRAG2)/(MASS2+MASS3-FLOW2*(T-BURNT1))-G)
            ELSE (F(FLOW3,VOUT3,DRAG3)/(MASS3-FLOW3*(T-BURNT1-BURNT2))-G))

DEFINITIONS
FUNCTION F (FLOW,VOUT,DRAG) = FLOW*VOUT-0.5*DRAG*RO*ALT**2

INITIAL
MASS1 = 148820.0    MASS2 = 32205.0    MASS3 = 8137.0
FLOW1 = 930.0      FLOW2 = 81.49     FLOW3 = 14.75
VOUT1 = 8060.0     VOUT2 = 13805.0    VOUT3 = 15250.0
DRAG1 = 510.0      DRAG2 = 460.0      DRAG3 = 360.0
BURNT1 = 150.0     BURNT2 = 359.0
ALT = 0            ALT' = 0
RADERH = 3960.0*5280.0

SIMULATION
DTMAX = 4          DTMIN = 0.00001
SIMTIME = 988

OUTPUT
HEAD $ LAUNCH OF THREE-STAGE ROCKET $
FIRST $           T$, $ ALT/5280$, $ ALT'*60/88$,
      $           G$, $ RO$
PRINT(15) T, ALT/5280.0, ALT'*60.0/88.0, ALT", G, RO
PLOT (T,SIMTIME/50) ALT'*60.0/88.0 AS V, ALT/5280 AS A
LAST ALT/5280, $ SLUT$

RUN

STOP
```

LAUNCH OF THREE-STAGE ROCKET

T	ALT/5280	VEL#60/88	ALT''	G	RO
0.0000+00	0.0000+00	0.0000+00	7.4564+00	3.2170+01	0.0000+00
1.5000+01	1.8025-01	9.1801+01	1.0554+01	3.2167+01	4.1468+01
3.0000+01	8.1148-01	2.1707+02	1.4006+01	3.2157+01	2.0180+02
4.5000+01	2.0412+00	3.7987+02	1.7931+01	3.2137+01	4.7150+02
6.0000+01	4.0380+00	5.8674+02	2.2716+01	3.2104+01	7.2496+02
7.5000+01	7.0082+00	8.4970+02	2.9027+01	3.2056+01	7.9100+02
9.0000+01	1.1223+01	1.1882+03	3.7608+01	3.1988+01	6.1192+02
1.0500+02	1.7052+01	1.6290+03	4.9127+01	3.1895+01	3.1904+02
1.2000+02	2.4987+01	2.2062+03	6.4539+01	3.1768+01	1.0213+02
1.3500+02	3.5694+01	2.9699+03	8.6168+01	3.1598+01	1.7553+01
1.5000+02	5.0114+01	4.0080+03	1.1956+02	3.1371+01	1.3397+01
1.5000+02	5.0114+01	4.0080+03	-3.4928+00	3.1371+01	1.3397+00
1.6500+02	6.6748+01	3.9781+03	-2.3554+00	3.1112+01	3.3980-02
1.8000+02	8.3281+01	3.9600+03	-1.1737+00	3.0858+01	8.8626-04
1.9500+02	9.9765+01	3.9543+03	6.5723-02	3.0608+01	2.3516-05
2.1000+02	1.1625+02	3.9616+03	1.3704+00	3.0361+01	6.2769-07
2.2500+02	1.3280+02	3.9826+03	2.7485+00	3.0116+01	1.6653-08
2.4000+02	1.4946+02	4.0181+03	4.2093+00	2.9873+01	4.3367-10
2.5500+02	1.6630+02	4.0690+03	5.7633+00	2.9629+01	1.0937-11
2.7000+02	1.8339+02	4.1363+03	7.4227+00	2.9385+01	2.6328-13
2.8500+02	2.0080+02	4.2212+03	9.2015+00	2.9140+01	5.9577-15
3.0000+02	2.1859+02	4.3250+03	1.1116+01	2.8892+01	1.2465-16
3.1500+02	2.3687+02	4.4491+03	1.3185+01	2.8641+01	2.3684-18
3.3000+02	2.5570+02	4.5953+03	1.5432+01	2.8386+01	4.0085-20
3.4500+02	2.7519+02	4.7655+03	1.7883+01	2.8125+01	5.9173-22
3.6000+02	2.9545+02	4.9619+03	2.0571+01	2.7858+01	7.4450-24
3.7500+02	3.1658+02	5.1872+03	2.3536+01	2.7583+01	7.7847-26
3.9000+02	3.3872+02	5.4444+03	2.6826+01	2.7300+01	6.5789-28
4.0500+02	3.6200+02	5.7372+03	3.0501+01	2.7007+01	4.3569-30
4.2000+02	3.8659+02	6.0699+03	3.4639+01	2.6702+01	2.1843-32
4.3500+02	4.1265+02	6.4476+03	3.9336+01	2.6385+01	0.0000+00
4.5000+02	4.4039+02	6.8768+03	4.4722+01	2.6053+01	0.0000+00
4.6500+02	4.7004+02	7.3653+03	5.0966+01	2.5706+01	0.0000+00
4.8000+02	5.0186+02	7.9230+03	5.8299+01	2.5340+01	0.0000+00
4.9500+02	5.3618+02	8.5626+03	6.7045+01	2.4955+01	0.0000+00
5.0900+02	5.7078+02	9.2481+03	7.6891+01	2.4575+01	0.0000+00
5.0900+02	5.7078+02	9.2481+03	3.0686+00	2.4575+01	0.0000+00
5.1000+02	5.7335+02	9.2502+03	3.1467+00	2.4547+01	0.0000+00
5.2500+02	6.1196+02	9.2884+03	4.3351+00	2.4134+01	0.0000+00
5.4000+02	6.5077+02	9.3390+03	5.5598+00	2.3730+01	0.0000+00
5.5500+02	6.8981+02	9.4023+03	6.8255+00	2.3333+01	0.0000+00
5.7000+02	7.2914+02	9.4788+03	8.1372+00	2.2943+01	0.0000+00
5.8500+02	7.6882+02	9.5689+03	9.5008+00	2.2560+01	0.0000+00
6.0000+02	8.0890+02	9.6733+03	1.0922+01	2.2182+01	0.0000+00
6.1500+02	8.4945+02	9.7925+03	1.2409+01	2.1810+01	0.0000+00
6.3000+02	8.9052+02	9.9274+03	1.3969+01	2.1442+01	0.0000+00
6.4500+02	9.3220+02	1.0079+04	1.5610+01	2.1078+01	0.0000+00
6.6000+02	9.7454+02	1.0247+04	1.7344+01	2.0718+01	0.0000+00
6.7500+02	1.0176+03	1.0434+04	1.9182+01	2.0361+01	0.0000+00
6.9000+02	1.0615+03	1.0640+04	2.1136+01	2.0007+01	0.0000+00
7.0500+02	1.1063+03	1.0866+04	2.3224+01	1.9654+01	0.0000+00
7.2000+02	1.1521+03	1.1115+04	2.5462+01	1.9304+01	0.0000+00
7.3500+02	1.1990+03	1.1388+04	2.7873+01	1.8955+01	0.0000+00
7.5000+02	1.2470+03	1.1686+04	3.0483+01	1.8606+01	0.0000+00
7.6500+02	1.2964+03	1.2012+04	3.3321+01	1.8259+01	0.0000+00
7.8000+02	1.3472+03	1.2369+04	3.6425+01	1.7911+01	0.0000+00
7.9500+02	1.3995+03	1.2758+04	3.9841+01	1.7563+01	0.0000+00
8.1000+02	1.4535+03	1.3185+04	4.3625+01	1.7214+01	0.0000+00
8.2500+02	1.5094+03	1.3652+04	4.7848+01	1.6864+01	0.0000+00
8.4000+02	1.5674+03	1.4165+04	5.2598+01	1.6512+01	0.0000+00
8.5500+02	1.6276+03	1.4730+04	5.7993+01	1.6158+01	0.0000+00
8.7000+02	1.6902+03	1.5354+04	6.4183+01	1.5802+01	0.0000+00
8.8500+02	1.7556+03	1.6046+04	7.1372+01	1.5443+01	0.0000+00
9.0000+02	1.8240+03	1.6818+04	7.9841+01	1.5079+01	0.0000+00
9.1500+02	1.8959+03	1.7685+04	8.9984+01	1.4711+01	0.0000+00
9.3000+02	1.9716+03	1.8667+04	1.0238+02	1.4338+01	0.0000+00
9.4500+02	2.0516+03	1.9790+04	1.1789+02	1.3959+01	0.0000+00
9.6000+02	2.1367+03	2.1093+04	1.3793+02	1.3572+01	0.0000+00
9.7500+02	2.2277+03	2.2634+04	1.6485+02	1.3176+01	0.0000+00
9.8800+02	2.3123+03	2.4230+04	1.9706+02	1.2823+01	0.0000+00

2.3123+03 SLUT

\*\*\* LAUNCH OF THREE-STAGE ROCKET \*\*\*

A=ALT  
V=ALT

0.0000E+00 A  
0.0000E+00 V

5.0000E+06 A  
1.0000E+04 V

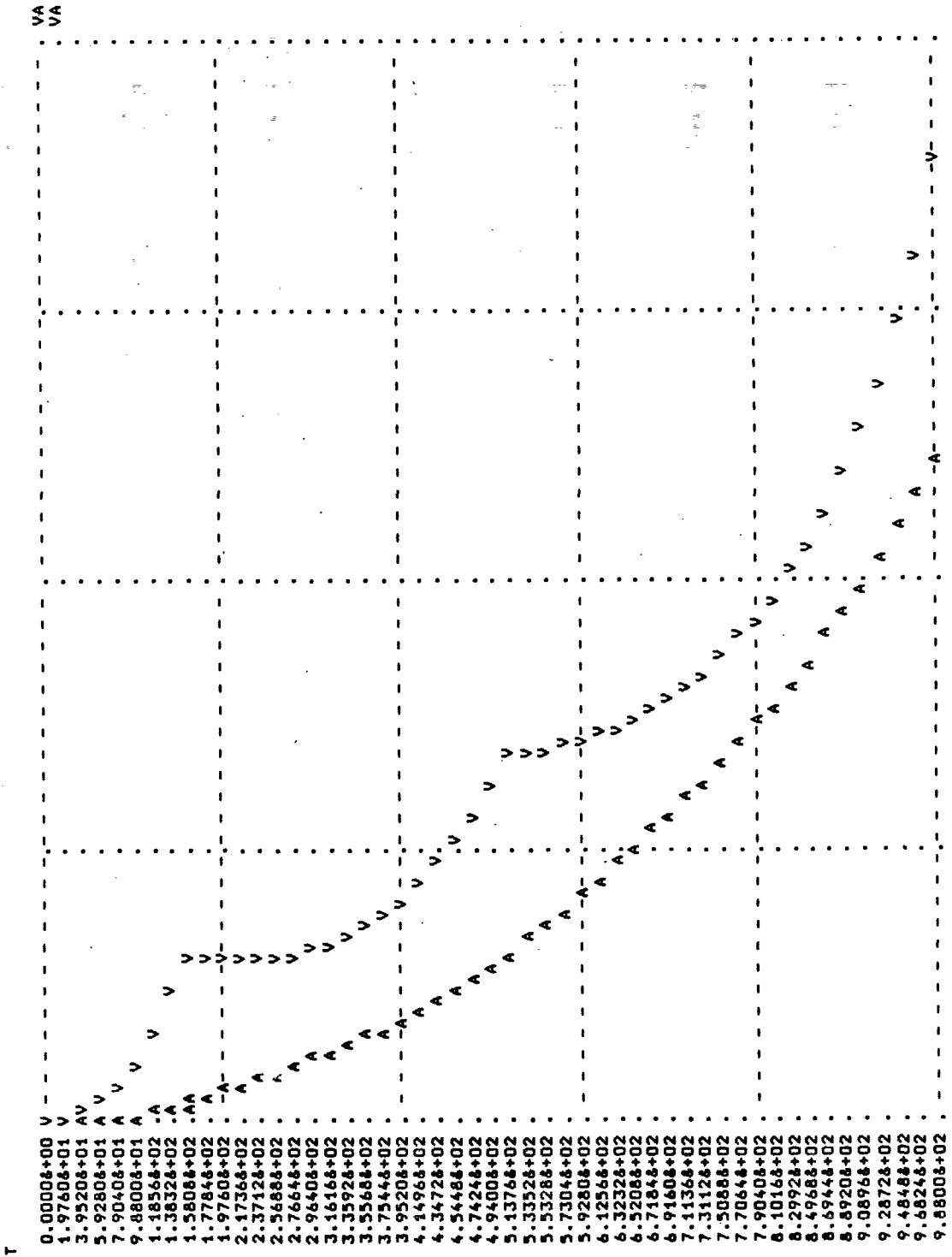
1.0000E+07 A  
2.0000E+04 V

2.0000E+07 A  
4.0000E+04 V

SCALES OF PLOT

1.0000E+07 A  
2.0000E+04 V

95 100 DUPLICATES



95 100 DUPLICATES

90 85 80 75 70 65 60 55 50 45 40 35 30 25 20 15 10 5 0

SCALES OF PLOT

1.0000E+07 A  
2.0000E+04 V

2.0000E+07 A  
4.0000E+04 V

95 100 DUPLICATES

## 5 MODELLERING OG MODELMODIFIKATION.

### 5.1 Eks.3: Væksten af en enkelt fisks biomasse.

I forbindelse med undersøgelser af fiskebestandes udvikling med henblik på en samlet model for fiskeriets indflydelse på fiskebestandene, kan man få brug for en model af udviklingen i en enkelt fisks biomasse,  $W(t)$ . Det er rimeligt at antage, at ændringen i fiskens biomasse,  $W'(t)$ , er forskellen mellem den biomasse, fisken optager, og den biomasse som den afgiver. Da en fisk udelukkende optager biomasse gennem tarmoverfladen, kan man forsøge at antage, at den optagne biomasse er proportional med tarmoverfladens størrelse, som vi vil benævne  $S(t)$ , og den optagne biomasse altså  $HS(t)$ , hvor  $H > 0$ .

Den biomasse, fisken afgiver, afgives som følge af en nedbrydning af fiskens væv, og da denne nedbrydning sker dels som et led i en konstant fornyelse af vævet, og dels ved almindeligt muskelarbejde, antages nedbrydningen at være proportional med mængden af væv, altså med den samlede vægt. Parallelt med biomasseopbygningen beskrives biomassenedbrydningen altså som  $-KW(t)$ ,  $K > 0$ . Som model for ændringen i fiskens biomasse fås altså:

$$W'(t) = HS(t) - KW(t) .$$

Hvis vi antager, at en fisk altid vokser isometrisk (lige hurtigt på alle leder), kan fiskens vægt opfattes som et rumfang, og tarmoverfladen som et areal. Da en terning med siden  $L$  jo har rumfanget  $V = L^3$ , og sidearealet  $A = L^2$ , er sammenhængen mellem arealet og rumfanget:  $A = V^{2/3}$ , vi kan altså erstatte  $S(t)$  med  $(W(t))^{2/3}$ , hvorefter differentialligningen kan skrives som:

$$W'(t) = HW(t)^{2/3} - KW(t) .$$

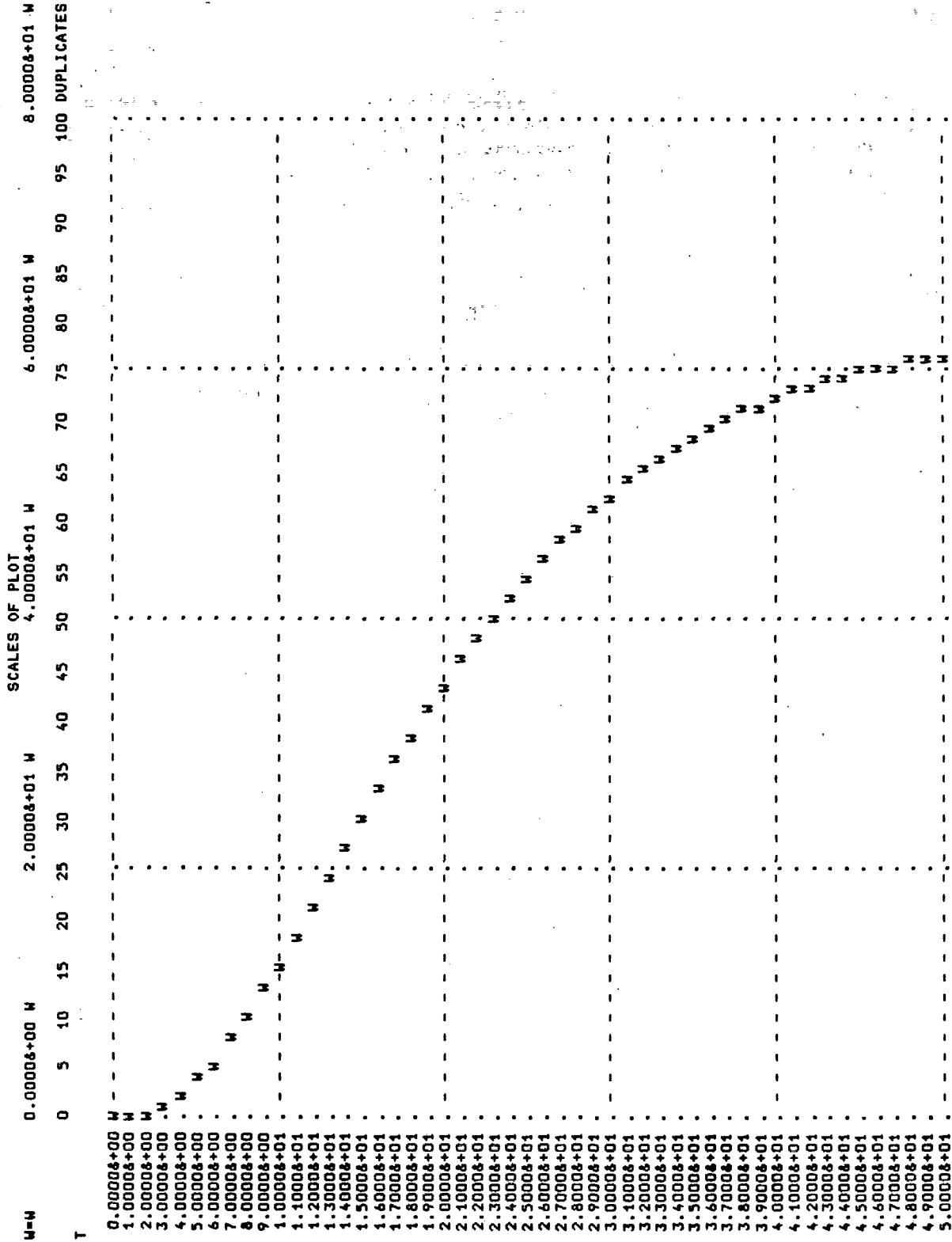
Denne ligning kaldes Bertalanffys ligning efter systemteoretikeren Ludwig von Bertalanffy, der har udviklet denne grundform for en model af ændringen i en fisks biomasse.

For at skaffe os et overblik over løsningens  $(W(T))$ 's forløb, kan vi opstille følgende ODIN-program:

```
EQUATIONS
  W' = H*W**(2/3)-K*W
INITIAL
  W = 0.001    H = 1    K = 0.25
SIMULATION
  SIMTIME = 50
OUTPUT
  PLOT(T,SIMTIME/50) W AS W
  HEAD $BERTALANFFYS MODEL FOR EN FISKS VÆGT
RUN
STOP
```

Man får følgende resultat:

\*\*\* BERTALANFFYS MODEL FOR EN FISKES VÆGT \*\*\*



T 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 DUPLICATES

Ved sammenligning med eksperimentelle måledata finder man, at løsningskurven passer dårligt med virkeligheden, og man har derfor forsøgt at forbedre modellen, så den stemmer bedre overens med måledata. Man er på denne måde (se URSIN, E (8)) nået frem til følgende model:

$$W'(t) = HW(t)^\alpha - KW(t)^\beta, \text{ hvor } 0 < \alpha < \beta < 1.$$

For de fleste fisk får man god overensstemmelse med virkeligheden, med  $\alpha = 2/3$  og  $\beta = 5/6$ , mens man for nogle laksearter får bedst overensstemmelse med måledata med  $\alpha \approx \beta \approx 0.8$ , altså  $W'(t) = (H-K) \cdot W(t)^{0.8}$ . Laksenes vækst bliver altså noget, der ligner eksponentiel vækst. At laksene alligevel ikke bliver gigantstore forklares med, at deres gydning medfører et så stort vægttab, at de kun overlever nogle få gydninger.

Vi vil nu vise, at modellen har samme kvalitative egenskaber ved forskellige valg af  $\alpha$  og  $\beta$ , og at vi derfor kan tilpasse modellen til forskellige sæt måledata ved at ændre på  $\alpha$  og  $\beta$ .

Hvis vi undersøger ligningen

$$W'(t) = HW(t)^\alpha - KW(t)^\beta$$

nærmere, ser vi at løsningskurvens tangent er vandret for

$$\begin{aligned} W'(t) &= 0 && \Leftrightarrow \\ HW(t)^\alpha &= KW(t)^\beta && \Leftrightarrow \\ W(t) &= 0 \quad \vee \quad W(t)^{\beta-\alpha} = H/K && \frac{1}{\beta-\alpha} \Leftrightarrow \\ W(t) &= 0 \quad \vee \quad W(t) = (H/K)^{\frac{1}{\beta-\alpha}} && \end{aligned}$$

For  $W(t) > 0$  fås

$$\frac{W'(t)}{W(t)^{\alpha K}} = H/K - W(t)^{\beta-\alpha}$$

Når  $W(t) < (H/K)^{\frac{1}{\beta-\alpha}}$  er  $W(t)^{\beta-\alpha} < H/K$ , altså

$$\frac{W'(t)}{W(t)^{\alpha K}} > 0 \text{ for } W(t) < (H/K)^{\frac{1}{\beta-\alpha}}.$$

Da  $W(t)^{\alpha K} > 0$ , er også  $W'(t) > 0$  for  $W(t) < (H/K)^{\frac{1}{\beta-\alpha}}$ .  
D.v.s. at et givet sæt  $\alpha$ ,  $\beta$ ,  $H$ ,  $K$  fastlægger et niveau  $W_\infty(t) = (H/K)^{\frac{1}{\beta-\alpha}}$ , for hvilket  $W' = 0$ , mens  $W' > 0$  under niveauet. Da  $W'$  kun er afhængig af  $W$ , mens  $t$  ikke indgår i udtrykket, (systemet er autonomt) kan  $W$  ikke passere  $W_\infty$ , da det ville kræve, at  $W'$  skulle blive større end 0 for  $W > W_\infty$ , men for  $W > W_\infty$  er  $W' < 0$ .

Den samme slags betragtninger kan gøres for niveauet  $W = 0$ , og resultatet bliver, at løsningskurven for ligningen

$$W' = HW^{\alpha} - KW^{\beta}$$

altid vil se ud som skitseret (se s.28), idet forskellige valg af  $\alpha$ ,  $\beta$ ,  $H$  og  $K$  blot vil flytte kurven og ændre dens stejthed. For at bestemme konstanterne  $\alpha$ ,  $\beta$ ,  $H$  og  $K$  kan man udføre visse akvarieforsøg med fisk, som det vil føre for vidt at beskrive her, da problemerne er af mere biologisk end matematisk karakter. En beskrivelse kan ses i URSIN, E (8).

#### 5.2 Eks.4: En rovdyr-byttedyr-model.

Vi opstillede i sidste afsnit en model for væksten af en art, der lever uden vekselvirkning med andre arter. Vi vil i dette afsnit vise et eksempel på en bestemt type to-artsmodeller, nemlig LOTKA-VOLTERRA-modellen. Der er tale om en model af et system, hvor to arter vekselvirker med hinanden på den måde, at den ene er føde for den anden.

Lad os som eksempel på et sådant system tænke os en isoleret biotop, f.eks. en ø, hvor der kun lever ræve og harer, og hvor der er føde nok til harerne. Hvis der kun var harer på øen, kunne vi beskrive udviklingen af harebestanden ved

$$I. \quad \text{HARER}' = A \text{ HARER} \quad , \text{ hvor } A > 0 ,$$

idet HARER betegner antallet af harer.

Hvis vi betegner antallet af ræve med RÆVE, kan vi på tilsvarende måde betegne udviklingen i en bestand af ræve, hvis der ingen harer var på øen:

$$II. \quad \text{RÆVE}' = -B \text{ RÆVE} \quad , \text{ hvor } B > 0 .$$

Hvis der både er ræve og harer på øen, således at rævne begrænser væksten i harebestanden, kunne det betyde, at harernes vækstrate,  $A$ , skulle være en funktion af antallet af ræve.  $A$  skulle så være en aftagende funktion af RÆVE, da mange ræve jo æder flere harer end få ræve, hvor den simpleste funktion ville være:

$$A = C_1 - C_2 \text{ RÆVE} \quad , \quad C_1, C_2 > 0 .$$

Vi kunne på tilsvarende måde sige, at rævenes vækstrate,  $-B$ , skal være en voksende funktion af antallet af harer, da mange harer giver føde til mange ræve, hvor det simpleste udtryk er:

$$-B = -C_3 + C_4 \text{ HARER} \quad , \quad C_3, C_4 > 0 .$$

Ligningerne I og II kan nu omskrives til:

$$\text{HARER}' = C1 \text{ HARER} - C2 \text{ R\text{E}VE} \text{ HARER}$$

III.

$$\text{R\text{E}VE}' = -C3 \text{ R\text{E}VE} + C4 \text{ HARER} \text{ R\text{E}VE}$$

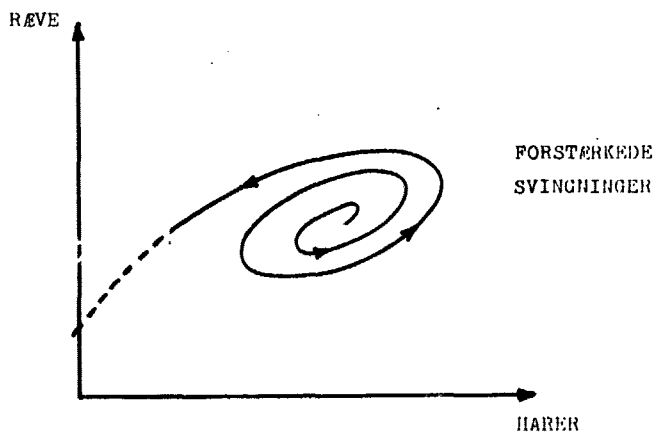
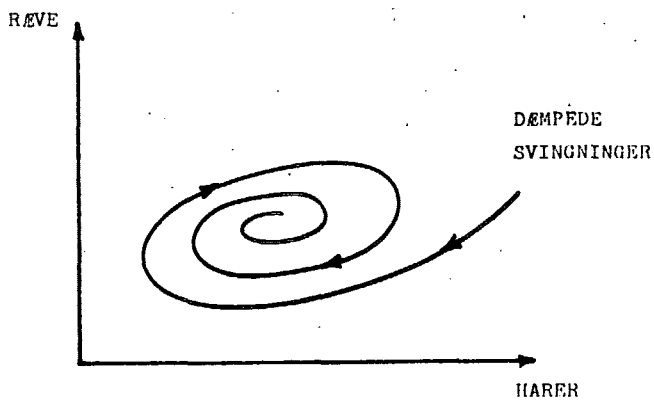
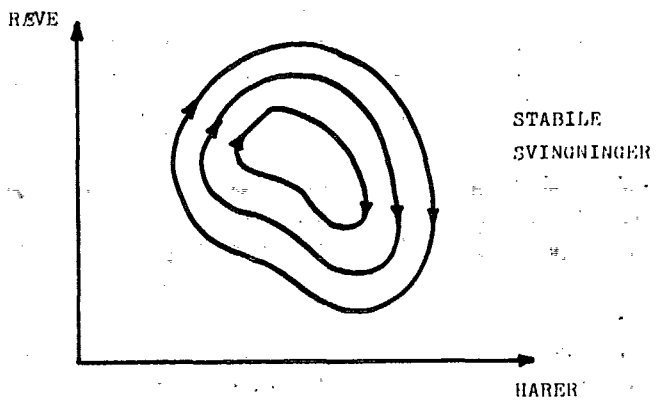
Dette system af to koblede 1. ordens differentiaalligninger kan ikke løses på traditionel vis, hvorfor vi vil søge at skaffe os et indtryk af løsningskurvernes forløb v.h.j.a. et ODIN-program. Ved numerisk løsning af to-artsmodeller angiver man tit løsningskurverne ved at afbilde den ene løsningsfunktion som funktion af den anden, altså afsætte sammenhørende værdier af de to funktioner i et koordinatsystem med de to funktioner som akser. Fasebilledet, som denne afbildning kaldes, giver nemlig et godt indtryk af, om den matematiske model, og den numeriske løsning, er stabil. Dette er interessant, da man ofte vil kræve af modeller af biologiske systemer, at de er stabile. Man har nemlig en intuitiv fornemmelse af, at naturen udviser stabilitet over kortere tidsrum, når den ikke udsættes for ydre påvirkninger.

Hvis systemet svinger stabilt, vil fasebilledet bestå af lukkede kurver, een for hver startsituation, mens et system med dæmpede svingninger vil give et fasebillede bestående af konvergerende spiraler. Hvis svingningerne i systemet derimod bliver kraftigere og kraftigere, vil fasebilledet være en spiral, der bevæger sig udad i stadig større buer, og derfor før eller siden vil skære en akse (dvs. den ene art uddør). Se figur 5.1 næste side.

Et ODIN-program til beskrivelse af vores opstillede ligningssystem, med angivelse af løsningen som et fasebillede, kan se ud som følger:

```
EQUATIONS
HARER' = C1*HARER - C2*R\text{E}VE*HARER
R\text{E}VE' = -C3*R\text{E}VE + C4*HARER*R\text{E}VE
INITIAL
R\text{E}VE = 100
HARER = 1000
C1 = 0.101
C2 = 0.00009
C3 = 0.1
C4 = 0.00008
SIMULATION
DTMAX = 1
DTMIN = 0.0001
STOPCONDITION = HARER < 5 OR R\text{E}VE < 1
SIMTIME = 500
OUTPUT
PLOT(HARER,1,100) R\text{E}VE AS R
HEAD $LOTKA-VOLTERRA MODEL$
RUN
STOP
```





Figur 5.1

\*\*\* LOTKA-VOLTERRA MODEL \*\*\*

R=REVE 0.0000&+00 R 2.0000&+03 R 4.0000&+03 R 6.0000&+03 R 8.0000&+03 R  
 SCALES OF PLOT 4.0000&+03 R 6.0000&+03 R 8.0000&+03 R 100 DUPLICATES

	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	
1.0600&+02																						
2.0600&+02																						
3.0600&+02																						
4.0600&+02																						
5.0600&+02																						
6.0600&+02																						
7.0600&+02																						
8.0600&+02																						
9.0600&+02																						
1.0060&+03																						
1.1060&+03																						
1.2060&+03																						
1.3060&+03																						
1.4060&+03																						
1.5060&+03																						
1.6060&+03																						
1.7060&+03																						
1.8060&+03																						
1.9060&+03																						
2.0060&+03																						
2.1060&+03																						
2.2060&+03																						
2.3060&+03																						
2.4060&+03																						
2.5060&+03																						
2.6060&+03																						
2.7060&+03																						
2.8060&+03																						
2.9060&+03																						
3.0060&+03																						
3.1060&+03																						
3.2060&+03																						
3.3060&+03																						
3.4060&+03																						
3.5060&+03																						
3.6060&+03																						
3.7060&+03																						
3.8060&+03																						
3.9060&+03																						
4.0060&+03																						
4.1060&+03																						
4.2060&+03																						
4.3060&+03																						
4.4060&+03																						
4.5060&+03																						
4.6060&+03																						
4.7060&+03																						
4.8060&+03																						
4.9060&+03																						

HARER 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 DUPLICATES  
 OUTPUT CONSISTS OF 502 POINTS (502 POINTS COLLECTED USING 502 CALLS OF PLOT)

Det ses, at fasebilledet bliver en lukket kurve, hvorfor modellen åbenbart beskriver biotopens udvikling som et system udvisende stabile svingninger. Dette resultat falder meget godt i tråd med en intuitiv fornemmelse af, at naturen arbejder i en fast rytme, når den ikke udsættes for ydre påvirkninger.

Ved en matematisk analyse af ligningssystemet kan man iøvrigt vise, at det vil udvise stabile svingninger for alle valg af  $C_1, C_2, C_3$  og  $C_4$ . En nærmere gennemgang af denne analyse vil kræve flere forudsætninger, end der kræves til dette kursus, hvorfor eventuelle interesserede henvises til f.eks. MAY, s. 188-189 (4).

Vi har i de foregående overvejelser over modellens opbygning hele tiden valgt de simplest tænkelige udtryk for vekselvirkningen mellem harer og ræve. Eksperimenter med f.eks. bakterier og bakteriofager har da også vist, at modellen ikke reproducerer opnåede måleresultater særligt præcist. Vi vil derfor nu forsøge at forbedre modellen ved at benytte ideerne om logistisk vækst til en udvidelse af modellen.

Hvis vi i systemet III erstatter den positive vækstrate for harer,  $C_1$ , med det logistiske led:

$$C_1 \cdot (1 - 1/D_1 \cdot \text{HARER}),$$

får vi introduceret en bærekapacitet for harebestanden,  $D_1$ . I den negative vækstrate,  $C_2$  RÆVE, kan man indføre en ændring, der er udtryk for, at hver ræv vil spise en mindre brøkdel af harerne, når der er mange harer. Ledet kan f.eks. ændres til:

$$C_2 \cdot \frac{\text{RÆVE}}{\text{HARER} + D_2},$$

således at den første ligning i systemet III nu kan skrives som:

$$\text{HARER}' = C_1(1 - 1/D_1 \cdot \text{HARER}) \cdot \text{HARER} - \frac{C_2 \cdot \text{RÆVE} \cdot \text{HARER}}{\text{HARER} + D_2}$$

Det sidste led er således udtryk for, at rævenes indhug i harebestanden vil vokse asymptotisk mod at være proportional med antallet af ræve ( $C_2$  RÆVE), når harebestanden bliver stor.

Man kan indføre et tilsvarende logistisk led for rævenes vækst, hvor det er oplagt, at bærekapaciteten for rævene under passende omstændigheder er proportional med antallet af harer. Det logistiske led kan derfor se ud som følger:

$$C_3 \cdot (1 - \frac{\text{RÆVE}}{D_3 \cdot \text{HARER}})$$

Da vi her har indregnet harebestandens indflydelse på rævebestandens vækst, kan vi omskrive ligningen for rævenes vækst til:

$$\text{RÆVE}' = C_3 \cdot \text{RÆVE} - \frac{C_3 \cdot \text{RÆVE}}{D_3 \cdot \text{HARER}} \cdot \text{RÆVE}$$

Vi har nu modificeret LOTKA-VOLTERRA-modellen til ligningssystemet:

$$\text{HARER}' = C1 \cdot (1 - \text{HARER}/D1) \cdot \text{HARER} - \frac{C2 \cdot \text{REVE} \cdot \text{HARER}}{\text{HARER} + D2}$$

IV.

$$\text{REVE}' = C3 \cdot \text{REVE} - \frac{C3 \cdot \text{REVE}}{D3 \cdot \text{HARER}} \cdot \text{REVE}$$

Ved at ændre programmet fra før til:

```

EQUATIONS
HARER' = C1*(1-HARER/D1)*HARER - C2*REVE*HARER/(HARER + D2)
REVE' = C3*REVE - C3*REVE**2/(D3*HARER)
INITIAL
REVE = 155
HARER = 4000
C1 = 0.101
C2 = 0.5
C3 = 0.1
D1 = 4000
D2 = 2000
D3 = 0.05
SIMULATION
DTMAX = 1
DTMIN = 0.0001
STOPCONDITION = HARER < 5 OR REVE < 1
SINTIME = 100
OUTPUT
PLOT(REVE,0.1,0.5) HARER AS H
PLOT(T,SINTIME/50) HARER AS H, REVE AS R
HEAD $MODIFICERET LOTKA-VOLTERRA MODEL$
RUN
STOP
    
```

kan vi få det på næste side viste fasebillede.

Det ses af kurverne, at systemet svinger ind mod en stabil tilstand, hvor harerne og rævene er i ligevægt, altså  $\text{HARER}' = 0$  og  $\text{REVE}' = 0$ . Systemet udvikler altså dæmpede svingninger.

Man kan vise (MAY (4) s. 189-193), at en nødvendig og tilstrækkelig betingelse for stabilitet er:

$$\frac{C3}{C1} > \frac{2 \left( \frac{C2 D3}{C1} - \sqrt{\left(1 - \frac{C2 D3}{C1} - \frac{D2}{D1}\right)^2 + 4 \frac{D2}{D1}} \right)}{1 + \frac{C2 D3}{C1} + \frac{D2}{D1} + \sqrt{\left(1 - \frac{C2 D3}{C1} - \frac{D2}{D1}\right)^2 + 4 \frac{D2}{D1}}}$$

Skønt May's analyse kun viser, at systemet er lokalt stabilt, kan en grundigere analyse (MAY, ch.4. (4)) vise, at betingelsen medfører global stabilitet, altså at systemet uafhængigt af startværdier ( $> 0$ ) vil svinge ind mod ligevægtpunktet. Modellen udviser altså en opførsel, der i høj grad stemmer overens med, hvad vi vil forvente af et biologisk system.

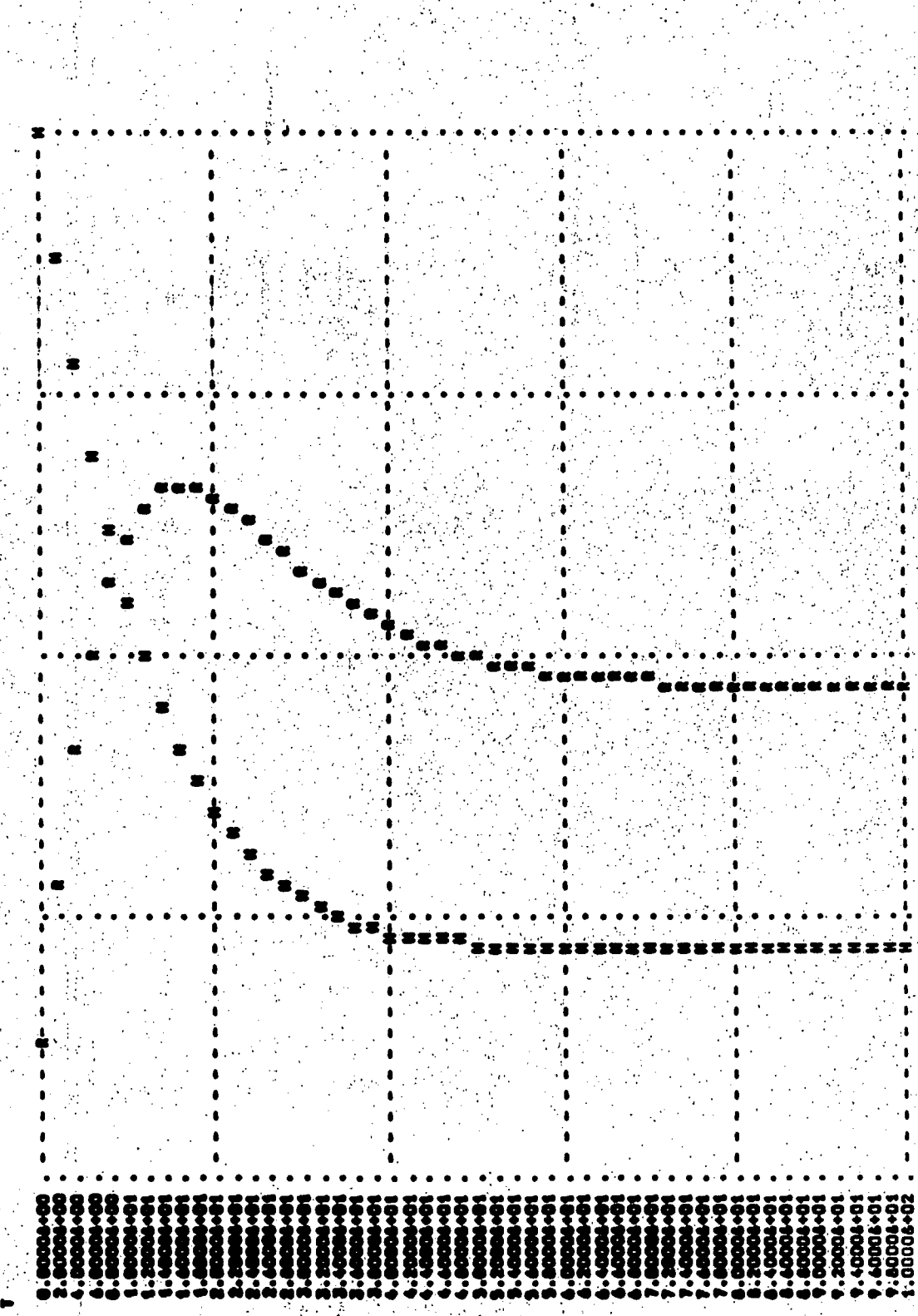
\*\*\* MODIFICERET LOTKA-VOLTERRA MODEL \*\*\*

H=HARER	3.2000e+03 H	3.4000e+03 H	SCALES OF PLOT	3.6000e+03 H	3.8000e+03 H	4.0000e+03 H																		
REVE	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	DUPLICATES		
1.5500e+02																								
1.5550e+02																								
1.5600e+02																								
1.5650e+02																								
1.5700e+02																								
1.5750e+02																								
1.5800e+02																								
1.5850e+02																								
1.5900e+02																								
1.5950e+02																								
1.6000e+02																								
1.6050e+02																								
1.6100e+02																								
1.6150e+02																								
1.6200e+02																								
1.6250e+02																								
1.6300e+02																								
1.6350e+02																								
1.6400e+02																								
1.6450e+02																								
1.6500e+02																								
1.6550e+02																								
1.6600e+02																								
1.6650e+02																								
1.6700e+02																								
1.6750e+02																								
1.6800e+02																								
1.6850e+02																								
1.6900e+02																								
1.6950e+02																								
1.7000e+02																								
1.7050e+02																								
1.7100e+02																								
1.7150e+02																								
1.7200e+02																								
1.7250e+02																								
1.7300e+02																								
1.7350e+02																								
1.7400e+02																								
1.7450e+02																								
1.7500e+02																								
1.7550e+02																								
1.7600e+02																								
1.7650e+02																								
REVE	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	DUPLICATES		

OUTPUT CONSISTS OF 1002 POINTS (1002 POINTS COLLECTED USING 1002 CALLS OF PLOT)

\*\*\* MODIFICERET LOTKA-VOLTERRA MODEL \*\*\*

H=HABER 3.2000E+03 H  
 R=REVE 1.5000E+02 R  
 SCALES OF PLOT 3.5000E+03 H  
 1.7000E+02 R  
 3.8000E+03 H  
 1.8000E+02 R  
 4.0000E+03 H  
 1.9000E+02 R  
 100 DUPLICATES



0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 DUPLICATES  
 T

### 5.3 Eks.5: En konkurrencemodel.

Vi har tidligere haft fornøjelsen af at se, hvordan væksten for en population kunne tænkes at forløbe, under visse idealiserede forhold, hvor der bl.a. ikke var taget højde for en evt. gensidig påvirkning mellem biotopens (forskellige) populationer (logistisk vækst, kap.4, s.20). Denne gensidige påvirkning mellem populationer kan og bliver ofte opdelt i følgende fire kategorier

- 1) Konkurrence (om føde)
- 2) Rovdyr versus byttedyr (incl. parasitter)
- 3) Adsler og forrådnelse
- 4) Symbiose

I dette eksempel vil vi se nærmere på den første kategori. Vi vil ved hjælp af en simpel model, hvor det system, vi kigger på, er et begrænset miljø, hvor andre effekter udelades, beskrive væksten af for overskuelighedens skyld kun 2 konkurrerende populationer ( $x_1$  og  $x_2$ ).

Den eneste begrænsning af en populations vækst i modellen for logistisk vækst

$$x' = v \cdot x \cdot (1 - x/K);$$

var begrænsningen i miljøet, bærekapaciteten  $K$ , som er et udtryk for populationstætheden eller for hvor mange individer biotopen maksimalt "kan bære". Ønsker vi at udbygge denne model, så der tages højde for den gensidige konkurrence om føden mellem  $x_1$  og  $x_2$ , må vi altså søge at finde et begrænsende led, som afhænger af den konkurrerende populations størrelse. Vælger vi dette led som en eller anden faktor ( $f_i \geq 0$ ) gange populationens størrelse ( $x_j$ , hvor  $i < j$ ), får vi nedenstående system:

$$x_1' = v_1 x_1 (1 - x_1/K_1) - f_1 x_2$$

$$x_2' = v_2 x_2 (1 - x_2/K_2) - f_2 x_1$$

Når man har opstillet en model, vil det, som tidligere omtalt, være rimeligt nøjere at undersøge, om den er i overensstemmelse med, hvad vi ved og forventer om sådanne systemers opførsel.

Modellen vil have et ligevægtpunkt ( $x_1^* = 0$ ,  $x_2^* = 0$ ), når

$$v_1 \cdot x_1 (1 - x_1/K_1) = f_1 \cdot x_2 \quad \text{og} \quad v_2 \cdot x_2 (1 - x_2/K_2) = f_2 \cdot x_1.$$

d.v.s. konkurrenceledet ( $f_1 x_2$  respektivt  $f_2 x_1$ ) og "den logistiske vækst", i hver af ligningerne, samtidigt er af modsat størrelse. Dette er bl.a. opfyldt for  $x_1 = 0$  og  $x_2 = 0$ .

Ved nogle beregninger, som det vil være for omfattende at komme ind på i denne forbindelse, kan man desuden vise, at modellen har et ligevægtpunkt for

$$f_1 f_2 < v_1 v_2.$$

når man antager, at  $x_1 > 0$  og  $x_1 < K_1$  samt, at  $x_2 > 0$  og  $x_2 < K_2$ . Det kan desuden vises, at betingelsen  $(f_1 f_2 < v_1 v_2)$  også er tilstrækkelig til at sikre  $0 < x_1 < K_1$  og  $0 < x_2 < K_2$ .

Vores model må derfor siges at udvise en matematisk set "passende opførsel", når blot produktet af konkurrencefaktorerne,  $f_1 f_2$ , er mindre end produktet af vækstraterne,  $v_1 v_2$ .

I det følgende vil vi ikke desto mindre modificere denne model, da vækstbegrænsningen ved konkurrencen kun afhænger af "den anden" populations størrelse. For at gøre modellen mere realistisk mener vi, at begrænsningen ved konkurrence skal være afhængig af begge populationers størrelse, da det er naturligt at antage, at populationerne også konkurrerer indbyrdes. I stedet for at påklistre "den logistiske vækst" et begrænsende led indsætter vi derfor, på tilsvarende måde som bærekapaciteten, leddene  $f_1 x_2$  henholdsvis  $f_2 x_1$ , således at den modificerede model får følgende udseende:

$$x_1' = v_1 \cdot x_1 \cdot (1 - x_1/K_1 - f_1 \cdot x_2)$$

$$x_2' = v_2 \cdot x_2 \cdot (1 - x_2/K_2 - f_2 \cdot x_1)$$

I denne model er populationernes vækst derfor afhængig, foruden af populationernes indbyrdes konkurrence, af konkurrencen mellem de enkelte individer i hver af populationerne, da konkurrenceledene bliver henholdsvis  $f_1 \cdot x_1 \cdot x_2$  og  $f_2 \cdot x_1 \cdot x_2$ . Faktorerne,  $f_1$  og  $f_2$ , udtrykker, i hvor høj grad de to populationer konkurrerer om den fælles føde, sådan at forstå, at det samtidigt også er et udtryk for, hvor dårlige de er til at komplettere manglende føde fra "den fælles pulje" med anden føde, som der ikke er konkurrence om.

Denne sidste model har et ligevægtpunkt ( $x_1' = 0$ ,  $x_2' = 0$ ) når  $x_1$  og  $x_2$  er lig med nul, men nok så interessant er det, at populationerne vil nå en ligevægt (ifølge N. Keyfitz (3)), når  $1/K_1 K_2 > f_1 f_2$ , hvilket med andre ord betyder, at begrænsningen i miljøet skal være større end den ved konkurrencen om føden. Hvis konkurrencen er større end begrænsningen i miljøet, vil udviklingen lede i en retning, hvor en af populationerne til sidst er udryddet. Denne, måske lidt overraskende, udvikling er faktisk i overensstemmelse med eksperimentelle resultater (af f.eks. Thomas Park, 1964 (5)). Det betyder altså, at "udstrakt konkurrence" ( $f_1 f_2 \geq 1/K_1 K_2$ ) giver ustabilitet i et begrænset miljø. Men når denne form for konkurrence alligevel forekommer i naturen, uden at den ene af populationerne bliver udryddet, skyldes det (3), at de kun delvist lever i den samme niche i biotopen. Men at et resultat forekommer "uventet" kan naturligvis også skyldes, at det er modellen, som er utilstrækkelig.



Vi kan nu på grundlag af de to sidst opstillede ligninger skrive følgende ODIN-program:

```
EQUATIONS
  X1' = V1*X1*(1 - X1/K1 - F1*X2)
  X2' = V2*X2*(1 - X2/K2 - F2*X1)

INITIAL
  X1= 50      X2= 50
  V1= 0.3     V2= 0.5
  K1= 100     K2= 500
  F1= 0.001   F2= 0.005

SIMULATION
  SIMTIME= 40
  MAXABSERR= 0.0001  MAXRELERR= 0.0001
  DTMAX= 0.5  DTMIN= 0.0001

OUTPUT
  HEAD $ KONKURRENCE-MODELLEN $
  PRINT(1) X1, X2, T
  PLOT(T,SIMTIME/50) X1 AS 1, X2 AS 2

RUN
STOP
```

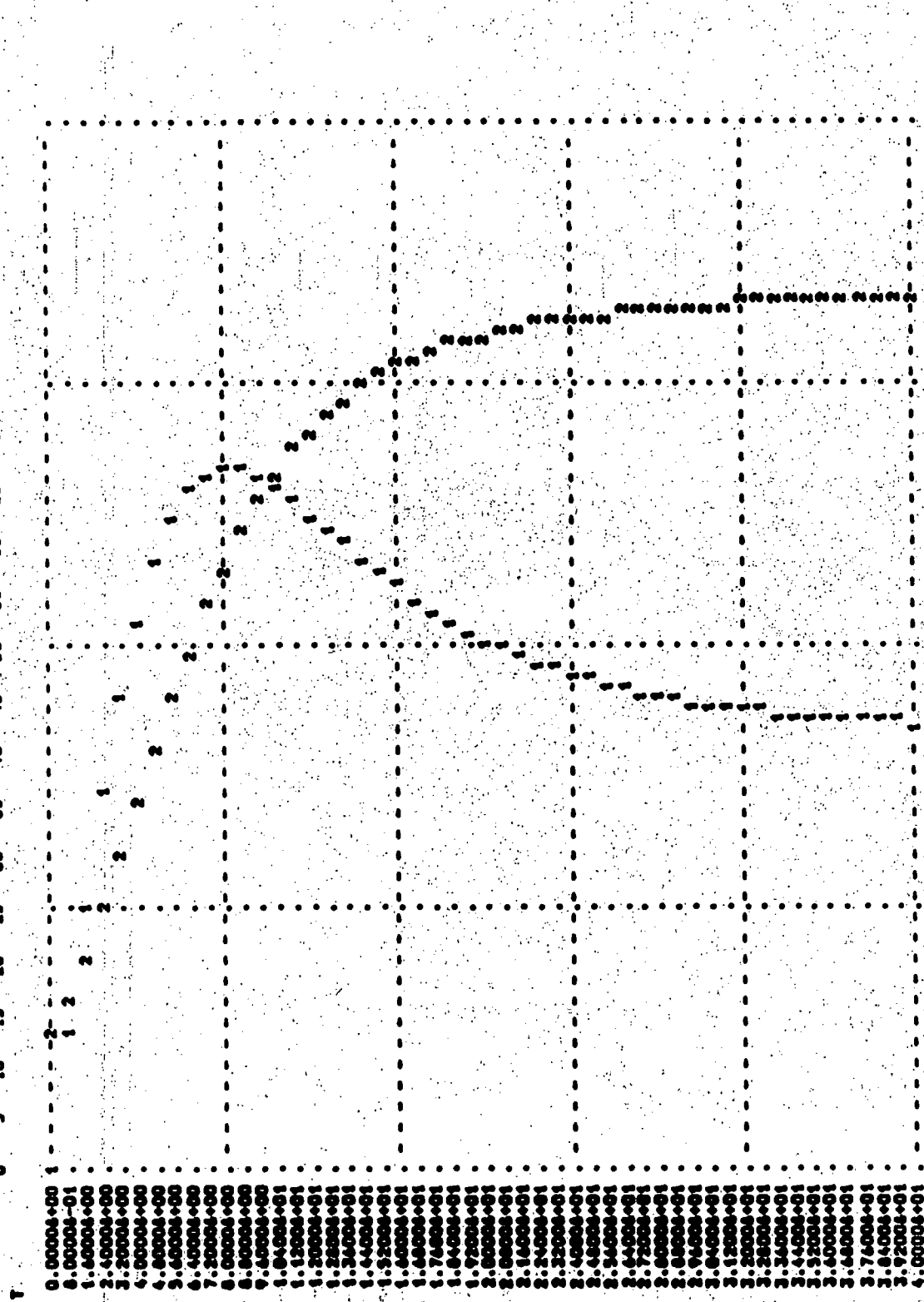
KONKURRENCE - MODELLEN

X1	X2	T
5.0000e+01	5.0000e+01	0.0000e+00
5.6515e+01	6.8037e+01	1.0000e+00
6.2332e+01	8.9376e+01	2.0000e+00
6.7194e+01	1.1325e+02	3.0000e+00
7.0974e+01	1.3849e+02	4.0000e+00
7.3679e+01	1.6379e+02	5.0000e+00
7.5418e+01	1.8793e+02	6.0000e+00
7.6360e+01	2.0998e+02	7.0000e+00
7.6690e+01	2.2944e+02	8.0000e+00
7.6582e+01	2.4616e+02	9.0000e+00
7.6183e+01	2.6023e+02	1.0000e+01
7.5607e+01	2.7193e+02	1.1000e+01
7.4936e+01	2.8158e+02	1.2000e+01
7.4231e+01	2.8954e+02	1.3000e+01
7.3528e+01	2.9609e+02	1.4000e+01
7.2853e+01	3.0151e+02	1.5000e+01
7.2218e+01	3.0602e+02	1.6000e+01
7.1632e+01	3.0979e+02	1.7000e+01
7.1097e+01	3.1296e+02	1.8000e+01
7.0611e+01	3.1564e+02	1.9000e+01
7.0174e+01	3.1793e+02	2.0000e+01
6.9782e+01	3.1988e+02	2.1000e+01
6.9432e+01	3.2157e+02	2.2000e+01
6.9119e+01	3.2302e+02	2.3000e+01
6.8841e+01	3.2428e+02	2.4000e+01
6.8594e+01	3.2537e+02	2.5000e+01
6.8374e+01	3.2633e+02	2.6000e+01
6.8180e+01	3.2716e+02	2.7000e+01
6.8007e+01	3.2790e+02	2.8000e+01
6.7854e+01	3.2854e+02	2.9000e+01
6.7718e+01	3.2910e+02	3.0000e+01
6.7598e+01	3.2960e+02	3.1000e+01
6.7491e+01	3.3003e+02	3.2000e+01
6.7397e+01	3.3042e+02	3.3000e+01
6.7314e+01	3.3075e+02	3.4000e+01
6.7240e+01	3.3105e+02	3.5000e+01
6.7174e+01	3.3132e+02	3.6000e+01
6.7116e+01	3.3155e+02	3.7000e+01
6.7065e+01	3.3176e+02	3.8000e+01
6.7019e+01	3.3194e+02	3.9000e+01
6.6979e+01	3.3210e+02	4.0000e+01

\*\*\* KONKURRENCE-MODELLEN \*\*\*

SCALES OF PLOT

1= X1 5.0000E+01 1 6.0000E+01 1 7.0000E+01 1 8.0000E+01 1 9.0000E+01 1  
 2= X2 0.0000E+00 2 1.0000E+02 2 2.0000E+02 2 3.0000E+02 2 4.0000E+02 2



T 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 DUPLICATES

#### 5.4 Eks.6: En mini-Nordsø-model.

I dette eksempel vil vi forsøge at kombinere nogle af de modelstrukturer, som vi har omtalt i de tidligere eksempler.

En del af vores inspiration til dette eksempel er hentet fra Nordsømodellen (9), hvorfor vi har valgt at opstille en model af et maritimt trearts-system, nemlig et system bestående af plankton, sild og torsk.

Lad os fastlægge vores opgave til: Ved hjælp af en model at beregne torske- og sildebestandene i et havområde igennem en årrække, når der til ethvert tidspunkt fiskes en given procent af bestandene.

Før vi begynder det egentlige modelarbejde, vil vi henlede opmærksomheden på, at vi af illustrative grunde har tilstræbt at gøre vores antagelser om systemets struktur og adfærd så simple og gennemskuelige som muligt, hvorfor de i en biologisk synsvinkel kan fremstå noget grove.

Vi har tidligere fra Eks.4 og Eks.5 set, hvordan to arter kan vekselvirke med hinanden. I dette tilfælde udvider vi modellen til at omhandle tre arter. Samtidigt vil vi opdele den enkelte art i forskellige grupper efter deres størrelse, da vi her vil antage, at store og små individer af samme art har forskellig adfærd. De spiser f.eks. ikke det samme, de små fiskes ikke, etc. Lad os nu kort og godt opskrive de antagelser, vi har gjort om systemet.

#### Opdeling af grupper:

- 1.- Sild opdeles i to grupper, store sild (S2) og små sild (S1).
- 2.- Torsk opdeles i tre grupper, store torsk (T3), mellemstore torsk (T2) og små torsk (T1).
- 3.- Plankton (P1) er summen af zooplankton og phytoplankton. Planktonbiomassen beskrives som et antal enheder med en konstant gennemsnitsvægt.

#### Føde:

- 1.- Plankton har en ubegrænset fødemængde.
- 2.- Små torsk og små sild æder kun plankton.
- 3.- Mellemstore torsk æder kun små sild.
- 4.- Store sild æder kun små torsk.
- 5.- Store torsk æder kun store sild.

Fiskning og naturlig dødsrate:

- 1.- Kun store sild, mellemstore og store torsk fiskes, og de fiskes med en fast procent af hver af bestandene.
- 2.- Torsk og sild dør af sult, hvis der er utilstrækkelig med føde.
- 3.- Alle grupper har en konstant dødsrate, som ikke skyldes hunger.

Reproduktion:

- 1.- Små sild fødes i et antal proportionalt med antallet af store sild.
- 2.- Små torsk fødes på lignende måde med et antal, der er proportionalt med antallet af store og mellemstore torsk.
- 3.- Plankton fødes kontinuerligt med en vækstrate, der giver logistisk vækst.

Den ide, som vi vil opbygge vores model efter, er at opstille to differentiaalligninger for hver gruppe. En, der skal beskrive et individs vægtændring ( $W^*$ ), og en, der skal beskrive ændringer i gruppens antal ( $N^*$ ). På denne måde kan den samlede biomasse for en gruppe beskrives som produktet af gruppens antal og et individs vægt (som er den samme for alle i gruppen).

Inden vi begynder opstillingen af differentiaalligningerne, vil vi indføre en notation for de enkelte grupper, variable og konstanter.

Vi vil benævne plankton-gruppen  $P_1$ , små sild  $S_1$ , store sild  $S_2$ , små torsk  $T_1$ , mellemstore torsk  $T_2$ , og store torsk  $T_3$ .

Ud fra dette kan vi så navngive de enkelte gruppers tilknyttede variable og konstanter. F.eks vil  $NT_1$  og  $FS_2$  være antallet af små torsk henholdsvis fiskeraten for store sild.

Denne notation vil herefter blive benyttet i teksten såvel som i det efterfølgende ODIN-program.

Fra eksempel 3 kender vi den modificerede Bertalanffy ligning:

$$dW/dt = H \cdot W^{2/3} - K \cdot W^{5/6}$$

som netop udtrykker en fisks vækst. Vi kan altså lade ligningen beskrive væksten af fisk i de enkelte grupper ved at angive en ligning for hver gruppe.  $WS_1^* = \dots$ ,  $WS_2^* = \dots$ , osv.

Imidlertid bliver f.eks små torsk, der vokser som en funktion af tiden, på et tidspunkt så store, at de skal flyttes til næste gruppe, dvs til mellemstore torsk (fra  $T_1$  til  $T_2$ ).

Vi har i denne model valgt at lade sådanne "gruppeskift" ske hvert år efter følgende mønster:

$S_1$  overføres til  $S_2$   
 $T_1$  overføres til  $T_2$   
 $T_2$  overføres til  $T_3$

Ved samme lejlighed antager vi, at de små sild fødes afhængigt af antallet af store sild, udtrykt ved konstanten GS2, mens de små torsk fødes afhængigt af antallet af mellemstore og store torsk, udtrykt ved konstanterne GT2 og GT3.

De næste differentialligninger, som vi vil opstille, skal beskrive ændringen af antallet i en bestemt gruppe. Lad os eksemplificere dette ved at "se" nærmere på gruppen med store sild (S2).

Tilvæksten sker, som tidligere nævnt, ved en diskret hændelse en gang om året. Næmlig når de små sild er blevet så store, at de flytter over i gruppen med store sild (, hvor der muligvis er nogle i forvejen). Frafaldet i bestanden sker derimod kontinuert og er bestemt af følgende faktorer:

- det antal der fiskes,
- det antal der ædes
- det antal der dør af sult
- det antal der dør af andre grunde.

Det antal, der fiskes, vil vi beskrive som:

$$FS2 * NS2 \text{ ( fiskerate * antal store sild )};$$

$$\text{hvor } 0 \leq FS2 \leq 1.$$

Det antal store sild, som ædes af store torsk, antages at være proportionalt med det første led i Bertalanffy's vækstligning for store torsk, som jo er et udtryk for, hvormed en enkelt torsk optager af den indfangede føde, gange antallet af store torsk. Altså  $HT3 * WT3^{2/3} * NT3$ . Lad os videre antage, at proportionalitetsfaktoren ES2 ( også kaldet æderate ) er 5 gange HT3, altså at en torsk spiser 5 gange mere, end den optager.

Dette medfører, at torskenes indflydelse på bestanden af store sild vil være: Torskenes fødeforbrug divideret med den totale biomasse af store sild gange antallet af store sild, altså:

$$(ES2 * WT3^{2/3} * NT3 / (WS2 * NS2)) * NS2 \quad \text{hvor } ES2 = 5 * HT3$$

Når den totale biomasse af føde ( $NT1 * WT1$ ) er for lille til at oppebære den aktuelle population af store sild ( $NS2$ ), vil et antal af store sild dø af sult. Lad os antage, at der for et hvert forhold imellem fødebehovet og den faktiske tilstedeværende fødemængde vil dø en given procent af bestanden af sult. Vi kan f.eks sige, at vi via eksperimenter har kunne fastlægge denne sammenhæng således:

Når forholdet imellem fødebehovet og den tilstedeværende føde, målt i biomasse, er:

1/1000	dør	0.0001%	af sult om dagen
1/100	dør	0.001%	af sult om dagen
1/10	dør	10%	af sult om dagen
1/1	dør	100%	af sult om dagen

Dette viser jo tydeligt, at bestanden meget hurtigt vil blive reduceret pga. sult, hvis forholdet nærmer sig 1/10.

Hvad vi kan gøre er, at indlægge disse data i en tabel (SULT), således at vi for ethvert forhold kan få beregnet den rate, hvormed f.eks store sild dør af sult. Altså:

$$\text{SULT}((5 \cdot \text{HS2} \cdot \text{WS2}^{2/3}) \cdot \text{NS2}) / (\text{WT1} \cdot \text{NT1});$$

Endelig vil vi antage, at antallet af store sild, som dør af andre grunde: sygdom, alderdom, etc., er proportionalt med bestandens størrelse, altså:

$$\text{DS2} \cdot \text{NS2} ; \text{ hvor } 0 \leq \text{DS2} \leq 1$$

Vi kan nu opstille en differentiaalligning for ændringen af antallet af store sild:

$$\begin{aligned} d\text{NS2}/dt = & -(\text{FS2} + ((\text{ES2} \cdot \text{WT3}^{2/3}) \cdot \text{NT3}) / (\text{WS2} \cdot \text{NS2})) \\ & + \text{SULT}((5 \cdot \text{HS2} \cdot \text{WS2}^{2/3}) \cdot \text{NS2}) / (\text{WT1} \cdot \text{NT1}) + \text{DS2}) \cdot \text{NS2}; \end{aligned}$$

For plankton vil vi imidlertid vælge en lidt anden beskrivelsesform. For det første vil vi sætte vægten af en planktonenhed til at være konstant, altså  $\text{WP1}' = 0$ . For det andet vil vi (i modsætning til i de andre grupper) lade tilvæksten i antallet af plankton ske kontinuerligt. Nemlig i form af en logistisk vækst, igennem hele simuleringsperioden.

$$\text{PP1} \cdot \text{NP1} \cdot (1 - \text{NP1}/\text{BKP});$$

hvor PP1 er en koefficient for produktionen og BKP er bærekapaciteten.

Da vi antager, at plankton hverken dør eller fiskes, kan vi nu beskrive den totale ændring af planktonmængden ( $\text{NP1}'$ ), som den producerede planktonmængde minus den planktonmængde, der ædes.

$$\begin{aligned} d\text{NP1}/dt = & \text{PP1} \cdot \text{NP1} \cdot (1 - (\text{NP1}/\text{BKP})) - ((\text{EP1} \cdot (\text{WS1}^{2/3})) \cdot \text{NS1} \\ & + \text{CP1} \cdot (\text{WT1}^{2/3}) \cdot \text{NT1}) / (\text{WP1} \cdot \text{NP1}) \cdot \text{NP1} \end{aligned}$$

hvor de to led med EP1 og CP1 er udtryk for, hvor meget plankton små sild henholdsvis små torske æder.

Den biologiske argumentation for valg og estimering af konstanter og parametre vil vi spare læseren for og blot henvise til de værdier, som vi opererer med i programmet.

EQUATIONS

WP1' = 0  
NP1' = +PP1\*NP1\*(1-(NP1/BKP))  
-(( EP1\*(WS1\*\*(2/3))\*NS1 + CP1\*(WT1\*\*(2/3))\*NT1 )/(WP1\*NP1) ) \*NP1  
WS1' = HS1\*WS1\*\*(2/3) - KS1\*WS1\*\*(5/6)  
NS1' = -( ES1\*WT2\*\*(2/3)\*NT2/(WS1\*NS1) + DS1 + FS1  
+ SULT(5\*HS1\*WS1\*\*(2/3)\*NS1/(WP1\*NP1) ) ) \*NS1  
WS2' = HS2\*WS2\*\*(2/3) - KS2\*WS2\*\*(5/6)  
NS2' = -( ES2\*WT3\*\*(2/3)\*NT3/(WS2\*NS2) + DS2 + FS2  
+ SULT(5\*HS2\*WS2\*\*(2/3)\*NS2/(WT1\*NT1) ) ) \*NS2  
WT1' = HT1\*WT1\*\*(2/3) - KT1\*WT1\*\*(5/6)  
NT1' = -( ET1\*WS2\*\*(2/3)\*NS2/(WT1\*NT1) + DT1 + FT1  
+ SULT(5\*HT1\*WT1\*\*(2/3)\*NT1/(WP1\*NP1) ) ) \*NT1  
WT2' = HT2\*WT2\*\*(2/3) - KT2\*WT2\*\*(5/6)  
NT2' = -( DT2 + FT2 + SULT(5\*HT2\* WT2\*\*(2/3)\*NT2/(WS1\*NS1) ) ) \*NT2  
WT3' = HT3\*WT3\*\*(2/3) - KT3\*WT3\*\*(5/6)  
NT3' = -( DT3 + FT3 + SULT(5\*HT3\* WT3\*\*(2/3)\*NT3/(WS2\*NS2) ) ) \*NT3

INITIAL

WP1=5&13	WS1=1&-2	WS2=0.2	WT1=1&-2	WT2=0.25	WT3=1.0
NP1=1&-6	NS1=1&7	NS2=5&5	NT1=1&8	NT2=5&5	NT3=1&3
PP1=0.0015	HS1=0.0075	HS2=0.0075	HT1=0.0055	HT2=0.0055	HT3=0.0055
BKP=1&14	KS1=0.00675	KS2=0.00675	KT1=0.003	KT2=0.003	KT3=0.003
DP1=0	DS1=0.002	DS2=0	DT1=0.005	DT2=0.004	DT3=0
FP1=0	FS1=0	FS2=0.003	FT1=0	FT2=0.001	FT3=0.003
EP1=5*HS1	ES1=5*HT2	ES2=5*HT3	ET1=5*HS2		
CP1=5*HT1		GS2=1&4		GT2=1&5	GT3=1&5

SIMULATION

SIMTIME = 360  
MAXABSERR = 1&-20 MAXRELERR = 0.001  
DTMAX = 5 DTMIN = 0.001

OUTPUT

HEAD \$ MINI NORDS0-MODEL \$  
PRINT(15) NP1, NS1, NS2, NT1, NT2, NT3  
PRINT(15) WP1, WS1, WS2, WT1, WT2, WT3  
PLOT(T,3\*SIMTIME/100)  
NP1 AS A, NS1 AS B, NS2 AS C, NT1 AS D, NT2 AS E, NT3 AS F  
PLOT(T,3\*SIMTIME/100)  
WP1 AS G, WS1 AS H, WS2 AS I, WT1 AS K, WT2 AS L, WT3 AS M

DEFINITIONS

TABLEX: SULT = (0,0) (1/1000,1&-6) (1/100,1&-5) (1/10,1&-1) (1,1)

RUN

INITIAL

```
WS2 = (WS2*NS2 + WS1*NS1)/(NS2+NS1)
WS1 = 0.001
WT3 = (WT3*NT3 + WT2*NT2)/(NT3+NT2)
WT2 = WT1
WT1 = 0.001
```

```
NS0 = GS2*NS2
NS2 = NS2 + NS1
NS1 = NS0
```

```
NT0 = GT2*NT2 + GT3*NT3
NT3 = NT3 + NT2
NT2 = NT1
NT1 = NT0
```

SIMULATION

SIMSTART = T

OUTPUT

```
PRINT(15) NP1, NS1, NS2, NT1, NT2, NT3
PRINT(15) WP1, WS1, WS2, WT1, WT2, WT3
PLOT(T,3*SIMTIME/100)
  NP1 AS A, NS1 AS B, NS2 AS C, NT1 AS D, NT2 AS E, NT3 AS F
PLOT(T,3*SIMTIME/100)
  WP1 AS G, WS1 AS H, WS2 AS I, WT1 AS K, WT2 AS L, WT3 AS M
```

RUN

INITIAL

```
WS2 = (WS2*NS2 + WS1*NS1)/(NS2+NS1)
WS1 = 0.001
WT3 = (WT3*NT3 + WT2*NT2)/(NT3+NT2)
WT2 = WT1
WT1 = 0.001
```

```
NS0 = GS2*NS2
NS2 = NS2 + NS1
NS1 = NS0
```

```
NT0 = GT2*NT2 + GT3*NT3
NT3 = NT3 + NT2
NT2 = NT1
NT1 = NT0
```

SIMULATION

SIMSTART = T

OUTPUT

```
PRINT(15) NP1, NS1, NS2, NT1, NT2, NT3
PRINT(15) WP1, WS1, WS2, WT1, WT2, WT3
PLOT(T,3*SIMTIME/100)
  NP1 AS A, NS1 AS B, NS2 AS C, NT1 AS D, NT2 AS E, NT3 AS F
PLOT(T,3*SIMTIME/100)
  WP1 AS G, WS1 AS H, WS2 AS I, WT1 AS K, WT2 AS L, WT3 AS M
```

RUN

STOP



MINI NORDSØ-MODEL

T	NP1	NS1	NS2	NT1	NT2	NT3
0.0000+00	5.0000+13	1.0000+07	3.0000+05	1.0000+08	5.0000+05	1.0000+03
1.5000+01	4.9065+13	8.7739+06	9.6839+04	3.7689+07	1.6500+04	7.3251+02
3.0000+01	4.8622+13	8.2785+06	4.3473+04	2.2228+07	9.2587+03	4.1548+02
4.5000+01	4.8328+13	7.7176+06	2.6222+04	1.5296+07	6.8991+03	2.3043+02
6.0000+01	4.8104+13	7.1100+06	1.8349+04	1.1396+07	5.7284+03	1.4086+02
7.5000+01	4.7921+13	6.4937+06	1.4024+04	8.9177+06	5.0246+03	9.4434+01
9.0000+01	4.7768+13	5.8992+06	1.1357+04	7.2159+06	4.5504+03	6.8094+01
1.0500+02	4.7640+13	5.3455+06	9.5748+03	5.9842+06	4.2027+03	5.1929+01
1.2000+02	4.7534+13	4.8422+06	8.3145+03	5.0577+06	3.8985+03	4.1359+01
1.3500+02	4.7447+13	4.3917+06	7.3836+03	4.3399+06	3.6163+03	3.4086+01
1.5000+02	4.7379+13	3.9925+06	6.6722+03	3.7707+06	3.3546+03	2.8873+01
1.6500+02	4.7327+13	3.6406+06	6.1137+03	3.3108+06	3.1119+03	2.5009+01
1.8000+02	4.7292+13	3.3312+06	5.6654+03	2.9333+06	2.8867+03	2.2066+01
1.9500+02	4.7271+13	3.0593+06	5.2990+03	2.6192+06	2.6778+03	1.9770+01
2.1000+02	4.7264+13	2.8202+06	4.9948+03	2.3549+06	2.4841+03	1.7946+01
2.2500+02	4.7269+13	2.6094+06	4.7387+03	2.1303+06	2.3044+03	1.6471+01
2.4000+02	4.7286+13	2.4232+06	4.5074+03	1.9378+06	2.1377+03	1.5259+01
2.5500+02	4.7314+13	2.2583+06	4.2882+03	1.7715+06	1.9831+03	1.4240+01
2.7000+02	4.7352+13	2.1117+06	4.0802+03	1.6268+06	1.8397+03	1.3367+01
2.8500+02	4.7399+13	1.9809+06	3.8828+03	1.5001+06	1.7066+03	1.2604+01
3.0000+02	4.7455+13	1.8640+06	3.6953+03	1.3848+06	1.5832+03	1.1928+01
3.1500+02	4.7520+13	1.7590+06	3.5171+03	1.2786+06	1.4687+03	1.1321+01
3.3000+02	4.7593+13	1.6645+06	3.3477+03	1.1807+06	1.3625+03	1.0771+01
3.4500+02	4.7675+13	1.5792+06	3.1866+03	1.0905+06	1.2640+03	1.0266+01
3.6000+02	4.7765+13	1.5019+06	3.0334+03	1.0073+06	1.1726+03	9.7988+00
3.8000+02	4.7765+13	1.4334+07	2.8828+03	1.8244+08	1.0073+06	1.1824+03
3.7500+02	4.7753+13	1.8262+07	2.1801+04	7.0901+07	9.0056+03	4.2452+02
3.9000+02	4.7677+13	1.7461+07	1.2071+04	5.2061+07	3.0244+03	1.5334+02
4.0500+02	4.7537+13	1.6786+07	8.9927+03	3.8586+07	2.4952+03	8.1913+01
4.2000+02	4.7349+13	1.5700+07	7.4589+03	2.9026+07	2.2767+03	5.3482+01
4.3500+02	4.7133+13	1.4239+07	6.5251+03	2.2239+07	2.1120+03	3.9139+01
4.5000+02	4.6906+13	1.2637+07	5.8878+03	1.7367+07	1.9592+03	3.0781+01
4.6500+02	4.6682+13	1.1069+07	5.4195+03	1.3813+07	1.8175+03	2.5422+01
4.8000+02	4.6470+13	9.6377+06	5.0574+03	1.1176+07	1.6860+03	2.1747+01
4.9500+02	4.6276+13	8.3829+06	4.7666+03	9.1840+06	1.5641+03	1.9097+01
5.1000+02	4.6105+13	7.3091+06	4.5246+03	7.6536+06	1.4510+03	1.7112+01
5.2500+02	4.5957+13	6.4014+06	4.3018+03	6.4594+06	1.3461+03	1.5573+01
5.4000+02	4.5832+13	5.6382+06	4.0912+03	5.5141+06	1.2487+03	1.4339+01
5.5500+02	4.5729+13	4.9968+06	3.8918+03	4.7559+06	1.1585+03	1.3320+01
5.7000+02	4.5648+13	4.4565+06	3.7028+03	4.1402+06	1.0747+03	1.2460+01
5.8500+02	4.5586+13	3.9997+06	3.5234+03	3.6348+06	9.9701+02	1.1719+01
6.0000+02	4.5543+13	3.6116+06	3.3531+03	3.2157+06	9.2493+02	1.1069+01
6.1500+02	4.5516+13	3.2799+06	3.1913+03	2.8648+06	8.5807+02	1.0492+01
6.3000+02	4.5505+13	2.9950+06	3.0376+03	2.5685+06	7.9604+02	9.9718+00
6.4500+02	4.5509+13	2.7489+06	2.8914+03	2.3163+06	7.3849+02	9.4981+00
6.6000+02	4.5525+13	2.5308+06	2.7524+03	2.1002+06	6.8511+02	9.0626+00
6.7500+02	4.5538+13	2.3482+06	2.6201+03	1.9136+06	6.3598+02	8.6589+00
6.9000+02	4.5592+13	2.1842+06	2.4943+03	1.7516+06	5.8964+02	8.2766+00
7.0500+02	4.5641+13	2.0395+06	2.3746+03	1.6101+06	5.4702+02	7.9112+00
7.2000+02	4.5700+13	1.9113+06	2.2607+03	1.4858+06	5.0788+02	7.5620+00
7.3500+02	4.5700+13	2.2607+07	1.9135+06	5.1505+07	1.4858+06	5.1505+02
7.5000+02	4.6022+13	9.9581+06	1.0472+04	2.7876+07	3.8678+03	1.6433+01
7.6500+02	4.6281+13	9.5134+06	6.0657+03	2.5134+07	2.2550+03	5.6397+01
7.8000+02	4.6465+13	9.1639+06	4.8560+03	2.2190+07	1.8363+03	3.0948+01
7.9500+02	4.6583+13	8.8512+06	4.3293+03	1.9158+07	1.6675+03	2.1121+01
8.1000+02	4.6644+13	8.5603+06	4.0534+03	1.6301+07	1.5468+03	1.6227+01
8.2500+02	4.6657+13	8.2805+06	3.8445+03	1.3769+07	1.4349+03	1.3384+01
8.4000+02	4.6631+13	7.8884+06	3.6504+03	1.1612+07	1.3312+03	1.1532+01
8.5500+02	4.6579+13	7.3782+06	3.4658+03	9.8150+06	1.2349+03	1.0228+01
8.7000+02	4.6511+13	6.8078+06	3.2975+03	8.3358+06	1.1456+03	9.2581+00
8.8500+02	4.6436+13	6.2241+06	3.1360+03	7.1230+06	1.0628+03	8.5046+00
9.0000+02	4.6361+13	5.6598+06	2.9833+03	6.1282+06	9.8597+02	7.8997+00
9.1500+02	4.6290+13	5.1339+06	2.8386+03	5.3096+06	9.1470+02	7.4005+00
9.3000+02	4.6228+13	4.6557+06	2.7014+03	4.6328+06	8.4858+02	6.9786+00
9.4500+02	4.6176+13	4.2274+06	2.5711+03	4.0699+06	7.8724+02	6.6148+00
9.6000+02	4.6135+13	3.8475+06	2.4474+03	3.5988+06	7.3034+02	6.2954+00
9.7500+02	4.6106+13	3.5122+06	2.3299+03	3.2020+06	6.7758+02	6.0106+00
9.9000+02	4.6090+13	3.2171+06	2.2181+03	2.8656+06	6.2858+02	5.7453+00
1.0050+03	4.6085+13	2.9575+06	2.1118+03	2.5786+06	5.8315+02	5.4917+00
1.0200+03	4.6091+13	2.7288+06	2.0107+03	2.3323+06	5.4100+02	5.2493+00
1.0350+03	4.6109+13	2.5270+06	1.9146+03	2.1196+06	5.0190+02	5.0176+00
1.0500+03	4.6136+13	2.3486+06	1.8232+03	1.9350+06	4.6562+02	4.7961+00
1.0650+03	4.6173+13	2.1903+06	1.7362+03	1.7739+06	4.3178+02	4.5844+00
1.0800+03	4.6220+13	2.0494+06	1.6534+03	1.6325+06	4.0075+02	4.3821+00
1.0950+03	4.6275+13	1.9237+06	1.5746+03	1.5080+06	3.7179+02	4.1887+00

MINI NORDBØ-MODEL

T	WP1	WS1	WS2	WT1	WT2	WT3
0.0000+00	1.0000-06	1.0000-02	2.0000-01	1.0000-02	2.5000-01	1.0000+00
1.5000+01	1.0000-06	1.3294-02	2.1210-01	1.3114-02	2.6894-01	1.0378+00
3.0000+01	1.0000-06	1.7096-02	2.2440-01	1.6754-02	2.8862-01	1.0763+00
4.5000+01	1.0000-06	2.1404-02	2.3689-01	2.0944-02	3.0905-01	1.1154+00
6.0000+01	1.0000-06	2.6213-02	2.4955-01	2.5708-02	3.3022-01	1.1552+00
7.5000+01	1.0000-06	3.1516-02	2.6238-01	3.1064-02	3.5213-01	1.1956+00
9.0000+01	1.0000-06	3.7303-02	2.7536-01	3.7032-02	3.7478-01	1.2366+00
1.0500+02	1.0000-06	4.3564-02	2.8848-01	4.3628-02	3.9817-01	1.2783+00
1.2000+02	1.0000-06	5.0284-02	3.0172-01	5.0868-02	4.2230-01	1.3205+00
1.3500+02	1.0000-06	5.7451-02	3.1509-01	5.8764-02	4.4717-01	1.3634+00
1.5000+02	1.0000-06	6.5051-02	3.2856-01	6.7329-02	4.7276-01	1.4069+00
1.6500+02	1.0000-06	7.3068-02	3.4212-01	7.6575-02	4.9909-01	1.4509+00
1.8000+02	1.0000-06	8.1487-02	3.5578-01	8.6510-02	5.2615-01	1.4956+00
1.9500+02	1.0000-06	9.0292-02	3.6950-01	9.7145-02	5.5392-01	1.5409+00
2.1000+02	1.0000-06	9.9468-02	3.8330-01	1.0849-01	5.8242-01	1.5867+00
2.2500+02	1.0000-06	1.0900-01	3.9716-01	1.2054-01	6.1164-01	1.6331+00
2.4000+02	1.0000-06	1.1887-01	4.1106-01	1.3332-01	6.4156-01	1.6801+00
2.5500+02	1.0000-06	1.2906-01	4.2501-01	1.4681-01	6.7220-01	1.7276+00
2.7000+02	1.0000-06	1.3956-01	4.3898-01	1.6104-01	7.0354-01	1.7757+00
2.8500+02	1.0000-06	1.5035-01	4.5300-01	1.7601-01	7.3588-01	1.8243+00
3.0000+02	1.0000-06	1.6142-01	4.6703-01	1.9171-01	7.6831-01	1.8735+00
3.1500+02	1.0000-06	1.7275-01	4.8107-01	2.0814-01	8.0173-01	1.9232+00
3.3000+02	1.0000-06	1.8433-01	4.9512-01	2.2532-01	8.3584-01	1.9734+00
3.4500+02	1.0000-06	1.9615-01	5.0916-01	2.4324-01	8.7063-01	2.0242+00
3.6000+02	1.0000-06	2.0818-01	5.2320-01	2.6191-01	9.0610-01	2.0755+00
3.6000+02	1.0000-06	1.0000-03	2.0881-01	1.0000-03	2.6191-01	9.1579-01
3.7500+02	1.0000-06	2.0076-03	2.2106-01	1.8382-03	2.8132-01	9.5211-01
3.9000+02	1.0000-06	3.4488-03	2.3350-01	3.0183-03	3.0147-01	9.8908-01
4.0500+02	1.0000-06	5.3556-03	2.4612-01	4.5832-03	3.2237-01	1.0267+00
4.2000+02	1.0000-06	7.7490-03	2.5890-01	6.5709-03	3.4401-01	1.0650+00
4.3500+02	1.0000-06	1.0642-02	2.7184-01	9.0161-03	3.6639-01	1.1039+00
4.5000+02	1.0000-06	1.4041-02	2.8492-01	1.1950-02	3.8951-01	1.1435+00
4.6500+02	1.0000-06	1.7948-02	2.9814-01	1.5400-02	4.1337-01	1.1837+00
4.8000+02	1.0000-06	2.2361-02	3.1147-01	1.9392-02	4.3797-01	1.2246+00
4.9500+02	1.0000-06	2.7273-02	3.2491-01	2.3950-02	4.6330-01	1.2660+00
5.1000+02	1.0000-06	3.2678-02	3.3845-01	2.9094-02	4.8936-01	1.3081+00
5.2500+02	1.0000-06	3.8564-02	3.5208-01	3.4843-02	5.1615-01	1.3508+00
5.4000+02	1.0000-06	4.4921-02	3.6579-01	4.1214-02	5.4366-01	1.3941+00
5.5500+02	1.0000-06	5.1736-02	3.7957-01	4.8224-02	5.7189-01	1.4380+00
5.7000+02	1.0000-06	5.8994-02	3.9341-01	5.5885-02	6.0085-01	1.4825+00
5.8500+02	1.0000-06	6.6681-02	4.0730-01	6.4212-02	6.3051-01	1.5276+00
6.0000+02	1.0000-06	7.4783-02	4.2124-01	7.3214-02	6.6089-01	1.5732+00
6.1500+02	1.0000-06	8.3283-02	4.3521-01	8.2904-02	6.9197-01	1.6195+00
6.3000+02	1.0000-06	9.2167-02	4.4922-01	9.3289-02	7.2375-01	1.6663+00
6.4500+02	1.0000-06	1.0142-01	4.6324-01	1.0438-01	7.5623-01	1.7137+00
6.6000+02	1.0000-06	1.1102-01	4.7728-01	1.1618-01	7.8940-01	1.7616+00
6.7500+02	1.0000-06	1.2096-01	4.9132-01	1.2870-01	8.2326-01	1.8101+00
6.9000+02	1.0000-06	1.3121-01	5.0537-01	1.4194-01	8.5780-01	1.8591+00
7.0500+02	1.0000-06	1.4178-01	5.1941-01	1.5591-01	8.9302-01	1.9086+00
7.2000+02	1.0000-06	1.5263-01	5.3344-01	1.7061-01	9.2891-01	1.9587+00
7.2000+02	1.0000-06	1.0000-03	1.5308-01	1.0000-03	1.7061-01	9.4403-01
7.3500+02	1.0000-06	2.0076-03	1.6421-01	1.8382-03	1.8605-01	9.8086-01
7.5000+02	1.0000-06	3.4488-03	1.7561-01	3.0183-03	2.0222-01	1.0184+00
7.6500+02	1.0000-06	5.3556-03	1.8725-01	4.5832-03	2.1914-01	1.0565+00
7.8000+02	1.0000-06	7.7490-03	1.9911-01	6.5709-03	2.3680-01	1.0953+00
7.9500+02	1.0000-06	1.0642-02	2.1120-01	9.0161-03	2.5520-01	1.1347+00
8.1000+02	1.0000-06	1.4041-02	2.2349-01	1.1950-02	2.7434-01	1.1748+00
8.2500+02	1.0000-06	1.7948-02	2.3596-01	1.5400-02	2.9423-01	1.2153+00
8.4000+02	1.0000-06	2.2361-02	2.4861-01	1.9392-02	3.1487-01	1.2568+00
8.5500+02	1.0000-06	2.7273-02	2.6143-01	2.3950-02	3.3624-01	1.2988+00
8.7000+02	1.0000-06	3.2678-02	2.7440-01	2.9094-02	3.5836-01	1.3413+00
8.8500+02	1.0000-06	3.8564-02	2.8751-01	3.4843-02	3.8122-01	1.3856+00
9.0000+02	1.0000-06	4.4921-02	3.0074-01	4.1214-02	4.0482-01	1.4283+00
9.1500+02	1.0000-06	5.1736-02	3.1410-01	4.8224-02	4.2915-01	1.4726+00
9.3000+02	1.0000-06	5.8994-02	3.2756-01	5.5885-02	4.5422-01	1.5176+00
9.4500+02	1.0000-06	6.6681-02	3.4112-01	6.4212-02	4.8002-01	1.5631+00
9.6000+02	1.0000-06	7.4783-02	3.5476-01	7.3214-02	5.0653-01	1.6092+00
9.7500+02	1.0000-06	8.3283-02	3.6849-01	8.2904-02	5.3381-01	1.6559+00
9.9000+02	1.0000-06	9.2167-02	3.8228-01	9.3289-02	5.6179-01	1.7032+00
1.0050+03	1.0000-06	1.0142-01	3.9613-01	1.0438-01	5.9049-01	1.7510+00
1.0200+03	1.0000-06	1.1102-01	4.1003-01	1.1618-01	6.1990-01	1.7993+00
1.0350+03	1.0000-06	1.2096-01	4.2398-01	1.2870-01	6.5003-01	1.8482+00
1.0500+03	1.0000-06	1.3121-01	4.3796-01	1.4194-01	6.8086-01	1.8977+00
1.0650+03	1.0000-06	1.4178-01	4.5197-01	1.5591-01	7.1239-01	1.9476+00
1.0800+03	1.0000-06	1.5263-01	4.6599-01	1.7061-01	7.4462-01	1.9981+00



5.40008+02	CF	D	B	A														CE
5.50808+02	CF	D	B	A														CE
5.61608+02	CF	D	B	A														CE
5.72408+02	CF	D	B	A														CE
5.83208+02	CF	D	B	A														CE
5.94008+02	CF	D	B	A														CE
6.04808+02	CF	D	B	A														CE
6.15608+02	CF	D	B	A														CE
6.26408+02	CF	D	B	A														CE
6.37208+02	CFD	B	A															CE
6.48008+02	CFD	B	A															CE
6.58808+02	CFD	B	A															CE
6.69608+02	CFD	B	A															CE
6.80408+02	CFD	B	A															CE
6.91208+02	CFD	B	A															CE
7.02008+02	CFD	B	A															CE
7.12808+02	CFD	B	A															CE
7.23608+02	CDE	B	A															CE DF
7.34408+02	EC	F																AB
7.45208+02	EC	F																
7.56008+02	EC	F																
7.66808+02	C	F																CE
7.77608+02	C	F																CE
7.88408+02	C	F																CE
7.99208+02	C	F																CE
8.10008+02	CF	D	B	A														CE
8.20808+02	CF	D	B	A														CE
8.31608+02	CF	D	B	A														CE
8.42408+02	CF	D	B	A														CE
8.53208+02	CF	D	B	A														CE
8.64008+02	CF	D	B	A														CE
8.74808+02	CF	D	B	A														CE
8.85608+02	CF	D	B	A														CE
8.96408+02	CF	D	B	A														CE
9.07208+02	CF	D	B	A														CE
9.18008+02	CF	D	B	A														CE
9.28808+02	CF	D	B	A														CE
9.39608+02	CF	D	B	A														CE
9.50408+02	CF	D	B	A														CE
9.61208+02	CF	D	B	A														CE
9.72008+02	CF	D	B	A														CE
9.82808+02	CF	D	B	A														CE
9.93608+02	CF	D	B	A														CE
1.00448+03	CFD	B	A															CE
1.01528+03	CFD	B	A															CE
1.02608+03	C	D	B	A														CEF
1.03688+03	C	D	B	A														CEF
1.04768+03	C	D	B	A														CEF
1.05848+03	C	D	B	A														CEF
1.06928+03	C	D	B	A														CEF
1.08008+03	C	D	B	A														CEF

T 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 DUPLICATES

OUTPUT CONSISTS OF 632 POINTS (632 POINTS COLLECTED USING 636 CALLS OF PLOT)

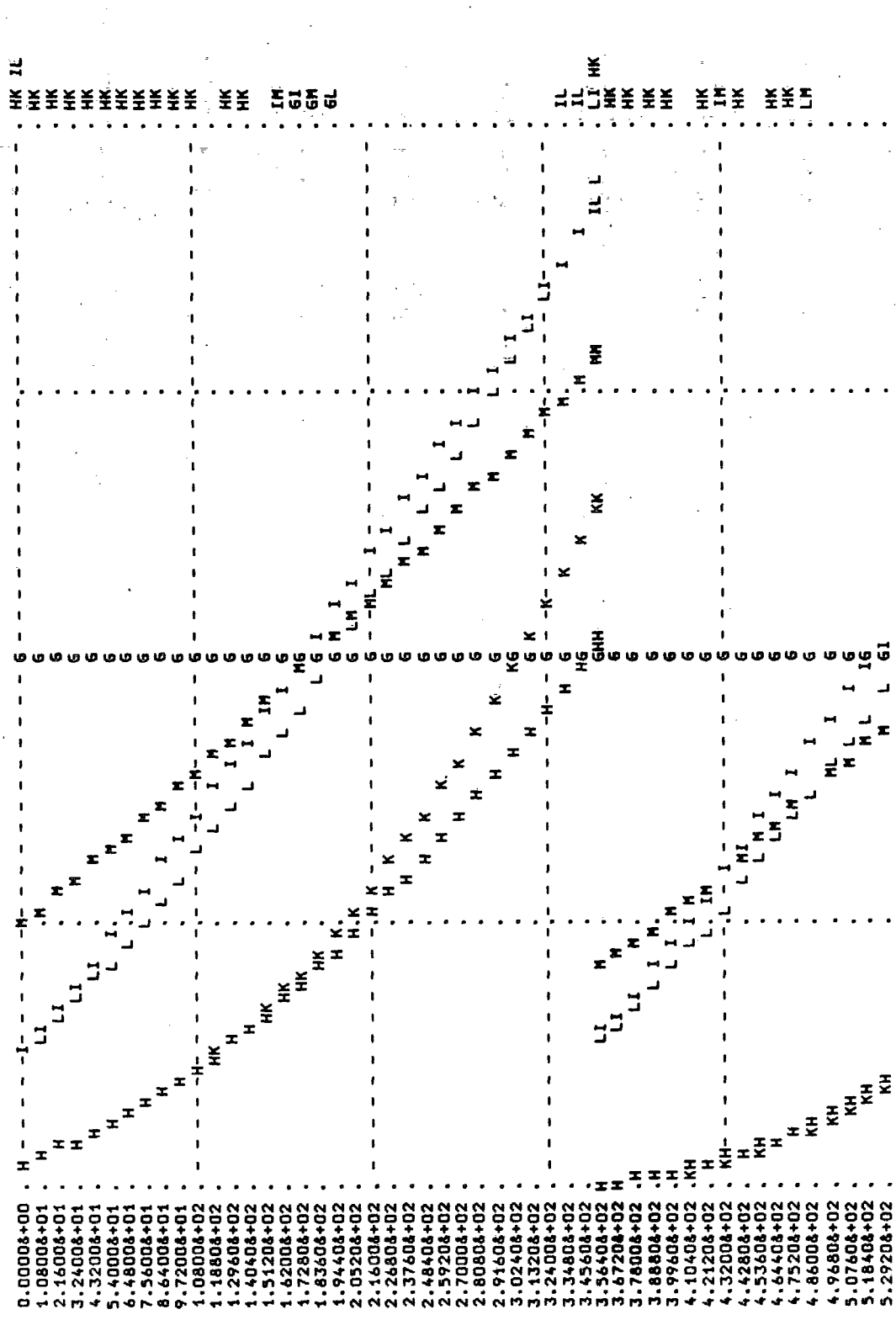
\*\*\* MINI NORDSØ-MODEL \*\*\*

G=WP1  
H=WS1  
I=WS2  
K=WT1  
L=WT2  
M=WT3

SCALES OF PLOT

0.0000&+00 G 0.0000&-07 G 5.0000&-00 G 1.5000&-06 G 1.5000&+00 M  
0.0000&+00 H 1.0000&-01 H 1.0000&-01 H 3.0000&-01 H 3.0000&-01 H  
0.0000&-01 I 2.5000&-01 I 2.0000&-01 I 4.5000&-01 I 4.5000&-01 I  
1.5000&+00 K 1.0000&+00 K 2.0000&-01 K 3.0000&-01 K 3.0000&-01 K  
1.5000&-01 L 3.5000&-01 L 5.5000&-01 L 7.5000&-01 L 7.5000&-01 L  
5.0000&-01 M 1.0000&+00 M 1.5000&+00 M 2.0000&+00 M 2.5000&+00 M

0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 DUPLICATES



	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
5.40008+02	KH																				
5.50808+02	H																				
5.61608+02	KH																				
5.72408+02	KH																				
5.83208+02	H																				
5.94008+02	KH																				
6.04808+02	H																				
6.15608+02	H																				
6.26408+02	KH																				
6.37208+02	KH																				
6.48008+02	KH																				
6.58808+02																					
6.69608+02																					
6.80408+02																					
6.91208+02																					
7.02008+02																					
7.12808+02																					
7.23608+02	IL																				
7.34408+02	IL																				
7.45208+02	IL																				
7.56008+02	IL																				
7.66808+02	IL																				
7.77608+02	IL																				
7.88408+02	IL																				
7.99208+02	IL																				
8.10008+02	IL																				
8.20808+02	IL																				
8.31608+02	IL																				
8.42408+02	IL																				
8.53208+02	IL																				
8.64008+02	IL																				
8.74808+02	IL																				
8.85608+02	IL																				
8.96408+02	IL																				
9.07208+02	IL																				
9.18008+02	IL																				
9.28808+02	IL																				
9.39608+02	IL																				
9.50408+02	IL																				
9.61208+02	IL																				
9.72008+02	IL																				
9.82808+02	IL																				
9.93608+02	IL																				
1.00448+03	IM																				
1.01528+03	IM																				
1.02608+03	IM																				
1.03688+03	IM																				
1.04768+03	IM																				
1.05848+03	IM																				
1.06928+03	IM																				
1.08008+03	IM																				

T 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 DUPLICATES  
 OUTPUT CONSISTS OF 636 POINTS (636 POINTS COLLECTED USING 636 CALLS OF PLOT)

## 6 KONKLUSION.

Det er lykkedes at udvikle et programmeringssprog, som gør numerisk simulering af differentiaalligningssystemer rimeligt tilgængeligt for programmeringskyndige.

Sproget er efter vores mening klart og overskueligt, let lært og rimeligt generelt, hvilket bl.a. kan ses af den lethed, hvormed det i vores eksempler har kunnet anvendes på en række forskellige typer modeller. Det bør kunne imødekomme de flestes behov, og specielt anvendt ved kurser er det en stor fordel, at flere ODIN-brugere kan anvende sproget samtidigt.

Grunden til, at vi anser det for vigtigt med et sprog som ODIN, er, at det efter vores mening er essentielt, at man ved denne type kurser beskæftiger sig med numeriske løsninger og i denne forbindelse med anvendelsen af datamaskiner. Denne kombination er nemlig så vigtigt et værktøj ved arbejdet med matematiske modeller, at det for kommende brugere af disse må være nyttigt at blive gjort bekendt med de muligheder, der ligger i anvendelsen af værktøjet. Desværre har vi ikke nået at få afsluttet implementeringen af oversætteren. Dette skyldes, dels at det er et ret omfattende stykke arbejde, og dels, men ikke mindst, at det resterende arbejde næsten udelukkende består af programmering, en aktivitet vi ikke kan forvente at få konteret nogen steder, da vi alle har afsluttet vores andet fag, datalogi.

I undervisnings-planer og -materialer indtager sproget en central rolle; men om det vil kunne stå sin prøve, eller om det skal "justeres lidt" hist og her, kan der naturligvis først siges noget afgørende, efter at det har været prøvet i praksis. Noget lignende kan siges om undervisnings-planer og -materialer.

APPENDIX 1.: BRUGERVEJLEDNING TIL PROGRAMMERINGSSPROGET ODIN.

Indholdsfortegnelse:

Indledning.....	55
Nogle nødvendige udtryksformer i sproget.....	56
En lille udvidelse af udtryksmulighederne.....	57
Et system med to koblede 1.ordens differentialligninger.....	58
Et system med en punktvís diskontinuert afledet.....	60
Tabellagte funktioner.....	65
Yderligere muligheder i sproget.....	70
Fejlmeddelelser fra systemet.....	70
Liste over standardfunktioner.....	72
Syntaksgrafer.....	73

Indledning:

Denne brugervejledning tjener som beskrivelse af et programmeringssprog, der gør det muligt på en let måde at opnå numeriske løsninger til differential-ligningssystemer.

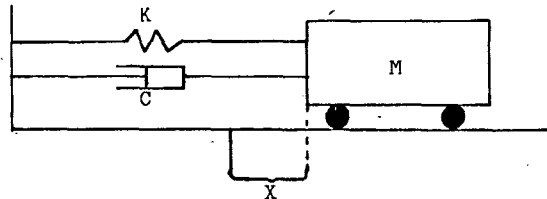
Sproget er primært tænkt som en hjælp til programmerings-ukyndige, som ellers ofte enten bruger lang tid på at lære et mere generelt programmeringssprog, eller helt opgiver at arbejde med modeller, som kræver numeriske løsninger. Det er forfatterens håb, at sproget vil vise sig så let at arbejde med, at kommende deltagere i undervisningsforløb om matematiske modeller (på RUC), vil finde det tiltalende at anvende datamaskinen som værktøj i undervisningen.



Nogle nødvendige udtryksformer i sproget.

Lad os gennem et eksempel se på, hvilke udtryksformer vi i hvert fald er nødt til at kende.

Vi vil opstille et program, der kan give en numerisk løsning til bevægelsesligningen for en masse, som bevæger sig lineært under påvirkning af en fjeder og en dæmper.



$$\begin{aligned} X(0) &= 1.0 \\ X'(0) &= 0.0 \\ M &= 1.5 \\ C &= 4.0 \\ K &= 150 \end{aligned}$$

Ved anvendelse af Newton's 2.lov kan dette system beskrives ved en 2. ordens differentiaalligning:

$$\frac{d^2X}{dt^2} + \frac{C}{M} \frac{dX}{dt} + \frac{K}{M} X = 0.$$

Vi ønsker løsningen af ligningen udskrevet som kurver visende  $x$ ,  $x'$  og  $x''$  som funktioner af  $T$ .

Programmet kommer til at se ud som følger:

```
EQUATIONS
  X'' + C/M * X' + K/M * X = 0

INITIAL
  C = 4.0  M = 1.5  K = 150  X = 1.0  X' = 0.0

OUTPUT
  PLOT(T,2) X AS X, X' AS V, X'' AS A

SIMULATION
  SIMTIME = 50

RUN
STOP
```

Det ses, at programmet er opdelt i 4 sektioner med hver deres overskrift. I sektionen med overskriften EQUATIONS står den differentiaalligning, vi ønsker løst. Ligningen behøver ikke at være ordnet med alle led på venstre side af lighedstegnet, når blot den højeste afledede af  $X$  står først på venstre side af lighedstegnet med koefficienten 1. Man må altså gerne skrive:

$$\begin{aligned} X'' + C/M * X' &= -K/M * X && \text{eller} \\ X'' &= -(C * X' + K * X)/M && \text{men ikke} \\ C/M * X' &= -K/M * X - X'' \end{aligned}$$

Vi så her, at man gerne må sætte parenteser i regneudtryk, samt at man benytter tegnene +, -, \*, / for henholdsvis addition, subtraktion, multiplikation

og division. Til potensopløftning benytter man \*\*, altså X i anden skrives som X\*\*2.

I sektionen med overskriften INITIAL gives konstanterne C, M og K værdier, og de lavere afledede af X, nemlig X og X', gives startværdier. Bemærk, at der alle steder i programmet skal benyttes punktum i stedet for decimalkomma.

I sektionen med overskriften OUTPUT angives det, hvilken type udskrift man ønsker fra programmet. Den viste PLOT-sætning vil bevirke, at der på linieskriveren bliver udskrevet et koordinatsystem med T som absците og X, X' og X'' som ordinater, markeret med henholdsvis et X, et V og et A for hver T-værdi. Afstanden mellem de enkelte T-værdier vil være 2 enheder i T.

Sektionen med overskriften SIMULATION indeholder oplysninger, som har noget at gøre med selve løsningen af ligningssystemet. I dette eksempel meddeles blot, at simuleringen skal være 50 enheder i T. Da der ikke er meddelt andet, startes med T = 0. Man kunne ellers starte med en vilkårlig T-værdi, f.eks. 17, ved at skrive SIMSTART = 17.

Når alle sektioner er skrevet, ligegyldigt i hvilken rækkefølge, skrives der RUN for at få programmet til at køre. Endelig sluttes der af med STOP, for at meddele at man ikke ønsker at ændre i programmet og køre det igen.

En lille udvidelse af udtryksmulighederne.

Vi vil se på et andet eksempel, nemlig et program som løser bevægelsesligningen for et matematisk penduls svingning. Det matematiske penduls svingning kan beskrives ved en 2.ordens differentialligning:

$$\frac{d^2y}{dt^2} + \frac{g}{m} \sin y = 0, \quad y(0) = 0 \quad y'(0) = 5 \quad g = 9.81 \quad m = 1$$

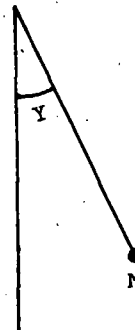
```
EQUATIONS
  Y'' + G/M + SIN(Y) = 0

INITIAL
  Y = 0  Y' = 5  G = 9.81  M = 1

OUTPUT
  PLOT(T,4) Y AS Y, Y' AS V

SIMULATION
  SINTIME = 100
  DTHIN = 4      DTHIN = 0.0001
  MAXABSERR = 0.0001  MAXRELERR = 0.0001

RUN
STOP
```



Vi ser her, at regneudtryk også kan indeholde brug af funktioner som sinus (SIN). Sidst i denne vejledning findes en liste over de funktioner, såkaldte standardfunktioner, som det er muligt at benytte. Det gælder for alle disse funktioner, at deres parametre, d.v.s. de værdier, funktionen skal beregnes udfra, skal angives i parentes efter funktionens navn.

Vi ser endvidere, at SIMULATION-sektionen har fået tilføjet nogle nye sætninger. Den første nye sætning, DTMAX = 4, angiver, at der højst må tages T-skridt af længden 4 under simuleringen, mens DTMIN = 0.0001 bestemmer størrelsen af de mindste T-skridt, der må tages. De to størrelser, MAXABSERR og MAXRELEERR, har med nøjagtigheden af den numeriske løsning at gøre. MAXABSERR angiver den største absolutte fejl, der må accepteres i et enkelt T-skridt, mens MAXRELEERR angiver den maksimale relative fejl, et enkelt T-skridt må give anledning til. Den numeriske metode, der anvendes til løsningen af differentiaalligningerne, sørger for at ændre størrelsen af T-skridtene mellem DTMAX og DTMIN på en sådan måde, at skridtene bliver så lange som muligt, samtidigt med at kravene til de maksimale absolutte og relative fejl bliver overholdt. Metoden, kaldet Runge-Kutta-Englands 4.ordens metode, er nærmere beskrevet i COMBINEDSIMULATION BRUGERHANDBOG (13).

Hvis DTMIN vælges for stor, kan man risikere, at programmet stopper, når nøjagtighedskravene ikke kan opfyldes, mens for lille DTMIN sammen med for små MAXABSERR og MAXRELEERR kan bevirke en unødigt lang køretid for programmet. Det er svært at angive generelle retningslinier for valget af DTMAX, DTMIN, MAXABSERR og MAXRELEERR, hvorfor det må anbefales, at man eksperimenterer sig frem i det enkelte tilfælde.

Hvis man intet specificerer, vælges DTMAX = 1, DTMIN = 0.00001, MAXABSERR = 0.00001 og MAXRELEERR = 0.00001.

#### Et system med to koblede 1.ordens differentiaalligninger.

Som eksempel på et system af differentiaalligninger vil vi skrive et program, som skal beskrive en isoleret population bestående af ræve og harer. Vi ønsker løsningen vist som en kurve visende sammenhængen mellem antallet af ræve og harer.

Systemet bestående af ræve og harer kan i en forenklet model beskrives ved de to koblede 1.ordens differentiaalligninger:

$$\frac{dRÆVE}{dt} = C0 \cdot RÆVE \cdot HARER - C1 \cdot RÆVE$$

$$\frac{dHARER}{dt} = C2 \cdot HARER - C3 \cdot RÆVE \cdot HARER$$

Vi vil løse ligningssystemet med følgende valg af begyndelsesværdier og konstanter:

```
RÆVE(0) = 100          HARER(0) = 1000
CO       = 0.00008     C1       = 0.1
C2       = 0.101       C3       = 0.00009
```

Vi ønsker løsningen beskrevet over 200 tidsenheder, idet vi dog vil stoppe, hvis rævebestanden kommer under 1, eller harebestanden kommer under 5.

Programmet kan se ud som følger:

```
EQUATIONS
RÆVE' = CO*RÆVE*HARER - C1*RÆVE
HARER' = C2*HARER - C3*RÆVE*HARER
INITIAL
RÆVE = 100
HARER = 1000
CO = 0.00008
C1 = 0.1
C2 = 0.101
C3 = 0.00009
SIMULATION
DTMAX = 1
DTMIN = 0.0001
STOPCONDITION = HARER < 5 OR RÆVE < 1
SIMTIME = 200
OUTPUT
PLOT(HARER,1.5) RÆVE AS R
RUN
STOP
```

Det nye i dette program er, at der i SIMULATION-sektionen er indført en ny sætning, nemlig:

```
STOPCONDITION = HARER < 5 OR RÆVE < 1
```

Denne sætning bevirker, at programmer stopper, når betingelsen på højre side af lighedstegnet er opfyldt. Det vil altså i dette tilfælde sige, når antallet af harer kommer under 5, eller antallet af ræve kommer under 1.

En betingelse kan sammensættes af vilkårlige regneudtryk sammenlignet med relationerne <, <=, =, >=, >, =/, svarende til de sædvanlige matematiske sammenligninger <, ≤, =, ≥, > og ≠.

Flore betingelser kan sammensættes til en betingelse med de logiske operatører AND og OR, svarende til  $\wedge$  og  $\vee$ , eller med EQV og IMP, svarende til biimplikation ( $\leftrightarrow$ ) og implikation ( $\Rightarrow$ ). Der må også benyttes parenteser i betingelsesudtryk.

Det skal bemærkes, at den numeriske løsning af differentiallyigningerne gør, at det er meget sjældent at et test på eksakt lighed bliver opfyldt. Betingelser med "=" alene bør derfor bruges med forsigtighed.

Der er også indført en ny type PLOT-sætning i OUTPUT-sektionen. Vi ønsker jo at få plottet den ene afhængige variabel, RÆVE, i forhold til den anden af-

hængige variabel, HARER. Dette gøres med sætningen:

PLOT(HARER,1,5) RÆVE AS R ,

hvor den første parameter til PLOT, HARER, angiver hvilken variabel, der ønskes som abscisse, mens den anden parameter, 1, specificerer med hvilke mellemrum i den uafhængige variabel, T, der skal opsamles information til plottet. Den tredje parameter, 5, angiver hvor mange enheder, der skal være mellem hver linie på abscisseaksen, altså hvor sammenpresset plottet skal være i længden.

Et system med en punktvis diskontinuert afledet.

Vi vil løse bevægelsesligningen for affyringen af en tretrinsraket, hvor accelerationen ændres diskontinuert, hver gang et afbrændt trin afkastes, og et nyt tages i brug. Raketts bevægelse kan beskrives ved følgende differentiaalligning (se (7) s.35ff for en mere dybtgående forklaring), idet ALT er raketts højde over jorden:

$$ALT'' = \begin{cases} \frac{FLOW1 \cdot VOUT1 - 0.5 \cdot DRAG1 \cdot R0 \cdot ALT'^2}{MASS1 + MASS2 + MASS3 - FLOW1 \cdot T} - G & \text{for } T \leq BURNT1 \\ \frac{FLOW2 \cdot VOUT2 - 0.5 \cdot DRAG2 \cdot R0 \cdot ALT'^2}{MASS2 + MASS3 - FLOW2 \cdot (T - BURNT1)} - G & \text{for } BURNT1 < T \leq \\ & BURNT1 + BURNT2 \\ \frac{FLOW3 \cdot VOUT3 - 0.5 \cdot DRAG3 \cdot R0 \cdot ALT'^2}{MASS3 - FLOW3 \cdot (T - BURNT1 - BURNT2)} - G & \text{for } T > BURNT1 + BURNT2 \end{cases}$$

hvor tyngdeaccelerationen G til enhver tid er givet ved

$$G = 32.17 \cdot \left( \frac{RADERH}{RADERH + ALT} \right)^2, \text{ hvor RADERH er jordens radius,}$$

og luftens massefylde, R0, er givet tilnærmelsesvist ved

$$R0 = 0.00238 \cdot e^{-ALT/24000}.$$

Konstanterne MASS1, MASS2 og MASS3 beskriver de tre trins startmasse, mens FLOW1, FLOW2 og FLOW3 er den hastighed, hvormed hvert trin forbruger masse, i form af brændstof. VOUT1, VOUT2 og VOUT3 er udstødningsgassens hastighed for hvert trin, og BURNT1 og BURNT2 er brændetiden for henholdsvis det første og det andet trin. DRAG1, DRAG2 og DRAG3 er luftmodstandskoefficienter, afhængige af hvert trins tværsnit.

Vi ønsker den numeriske løsning af bevægelsesligningen udskrevet som dels en kurve visende hastigheden som funktion af tiden, dels som en tabel over sammenhængen mellem tid, højde, hastighed, acceleration, tyngdeacceleration og lufttæthed.

Eftersom luften bliver tyndere og tyndere, jo højere man kommer op, vil  $R_0$  nærme sig uendeligt tæt til nul. Dette kan sommetider give problemer, så programmet vil være et eksempel på, hvorledes man kan stoppe kørslen, når  $R_0$  kommer tæt på nul, og derefter starte en ny kørsel med  $R_0 = 0$ , men med den sidste kørsels slutverdier som startverdier.

Programmet vil endvidere indeholde et eksempel på, hvorledes man kan definere en hyppigt forekommende udregning som en funktion, idet tællerne i de tre udtryk for  $ALT''$  kan skrives som en funktion af  $FLOW$ ,  $VOUT$  og  $DRAG$  for det pågældende rakettrin.

Programmet kan se ud som følger, idet konstanter og startverdier er valgt som i CSMP (7) på side 37.

```
EQUATIONS
G = 32.17*((RADERN/(RADERN+ALT))**2)
R0 = 0.00238*EXP(-ALT/24000.0)
ALT'' = IF T<BURNT1
      THEN F(FLOW1,VOUT1,DRAG1)/(MASS1+MASS2+MASS3-FLOW1*T)-G
      ELSE IF T<BURNT1+BURNT2
      THEN F(FLOW2,VOUT2,DRAG2)/(MASS2+MASS3-FLOW2*(T-BURNT1))-G
      ELSE F(FLOW3,VOUT3,DRAG3)/(MASS3-FLOW3*(T-BURNT1-BURNT2))-G

DEFINITIONS
FUNCTION : F(FLOW,VOUT,DRAG) = FLOW*VOUT-0.5*DRAG*R0*ALT''**2

INITIAL
MASS1 = 148820.0      MASS2 = 32205.0      MASS3 = 8137.0
FLOW1 = 930.0        FLOW2 = 81.49       FLOW3 = 14.73
VOUT1 = 8060.0       VOUT2 = 13805.0      VOUT3 = 19290.0
DRAG1 = 510.0        DRAG2 = 460.0        DRAG3 = 360.0
BURNT1 = 150.0       BURNT2 = 359.0
ALT = 0              ALT'' = 0
RADERN = 3960.0*5280.0

SIMULATION
DTMAX = 4            DTMIN = 0.00001
STOPCONDITION = R0 < 1E-38

OUTPUT
HEAD 0 LAUNCH OF THREE-STAGE ROCKET 0
FIRST 0          T0: 0          ALT/5280.0: 0          ALT''*60/880.0: 0
          0          G0: 0          R00
PRINT(15) T, ALT/5280.0, ALT''*60.0/88.0, ALT'', 0, R0
PLOT (T,SIMTIME/50) ALT''*60.0/88.0 AS V, ALT/5280 AS A

RUN

EQUATIONS
R0 = 0

INITIAL
ALT = ALT
ALT'' = ALT''

SIMULATION
SIMSTART = T
STOPCONDITION = MASS3 - FLOW3*(T-BURNT1-BURNT2) <= 1800

OUTPUT
PRINT(15) T, ALT/5280.0, ALT''*60.0/88.0, ALT'', 0
PLOT (T,SIMTIME/50) ALT''*60.0/88.0 AS V, ALT/5280 AS A
LAST ALT/5280.0 0 SLUTS

RUN

STOP
```

LAUNCH OF THREE-STAGE ROCKET

T	ALT/5280	VEL*60/88	ALT''	G	RO
0.0000+00	0.0000+00	0.0000+00	7.4564+00	3.2170+01	0.0000+00
1.5000+01	1.8025+01	9.1801+01	1.0554+01	3.2167+01	4.1468+01
3.0000+01	8.1148+01	2.1707+02	1.4006+01	3.2157+01	2.0180+02
4.5000+01	2.0412+00	3.7987+02	1.7931+01	3.2137+01	4.7150+02
6.0000+01	4.0380+00	5.8674+02	2.2716+01	3.2104+01	7.2496+02
7.5000+01	7.0082+00	8.4970+02	2.9027+01	3.2056+01	7.9100+02
9.0000+01	1.1223+01	1.1882+03	3.7608+01	3.1988+01	6.1192+02
1.0500+02	1.7052+01	1.6290+03	4.9127+01	3.1895+01	3.1904+02
1.2000+02	2.4987+01	2.2062+03	6.4539+01	3.1768+01	1.0213+02
1.3500+02	3.5694+01	2.9699+03	8.6168+01	3.1598+01	1.7953+01
1.5000+02	5.0114+01	4.0080+03	1.1956+02	3.1371+01	1.3397+00
1.5000+02	5.0114+01	4.0080+03	-3.4928+00	3.1371+01	1.3397+00
1.6500+02	6.6748+01	3.9781+03	-2.3554+00	3.1112+01	3.3980+02
1.8000+02	8.3281+01	3.9600+03	-1.1737+00	3.0858+01	8.8626+04
1.9500+02	9.9765+01	3.9543+03	6.5723+02	3.0608+01	2.3516+05
2.1000+02	1.1625+02	3.9616+03	1.3704+00	3.0361+01	6.2769+07
2.2500+02	1.3280+02	3.9826+03	2.7485+00	3.0116+01	1.6653+08
2.4000+02	1.4946+02	4.0181+03	4.2093+00	2.9873+01	4.3367+10
2.5500+02	1.6630+02	4.0690+03	5.7633+00	2.9629+01	1.0937+11
2.7000+02	1.8339+02	4.1363+03	7.4227+00	2.9385+01	2.6328+13
2.8500+02	2.0080+02	4.2212+03	9.2015+00	2.9140+01	5.9577+15
3.0000+02	2.1859+02	4.3250+03	1.1116+01	2.8892+01	1.2465+16
3.1500+02	2.3687+02	4.4491+03	1.3185+01	2.8641+01	2.3684+18
3.3000+02	2.5570+02	4.5953+03	1.5432+01	2.8386+01	4.0085+20
3.4500+02	2.7519+02	4.7655+03	1.7883+01	2.8125+01	5.9173+22
3.6000+02	2.9545+02	4.9619+03	2.0571+01	2.7858+01	7.4450+24
3.7500+02	3.1658+02	5.1872+03	2.3536+01	2.7583+01	7.7847+26
3.9000+02	3.3872+02	5.4444+03	2.6826+01	2.7300+01	6.5789+28
4.0500+02	3.6200+02	5.7372+03	3.0501+01	2.7007+01	4.3569+30
4.2000+02	3.8659+02	6.0699+03	3.4639+01	2.6702+01	2.1843+32
4.2791+02	4.0013+02	6.2630+03	3.7038+01	2.6536+01	
4.3500+02	4.1265+02	6.4476+03	3.9336+01	2.6385+01	
4.5000+02	4.4039+02	6.8768+03	4.4722+01	2.6053+01	
4.6500+02	4.7004+02	7.3653+03	5.0966+01	2.5706+01	
4.8000+02	5.0186+02	7.9230+03	5.8299+01	2.5340+01	
4.9500+02	5.3618+02	8.5626+03	6.7045+01	2.4955+01	
5.0900+02	5.7078+02	9.2481+03	7.6891+01	2.4575+01	
5.0900+02	5.7078+02	9.2481+03	3.0686+00	2.4575+01	
5.1000+02	5.7335+02	9.2502+03	3.1467+00	2.4547+01	
5.2500+02	6.1196+02	9.2884+03	4.3351+00	2.4134+01	
5.4000+02	6.5077+02	9.3390+03	5.5598+00	2.3730+01	
5.5500+02	6.8981+02	9.4023+03	6.8255+00	2.3333+01	
5.7000+02	7.2914+02	9.4788+03	8.1372+00	2.2943+01	
5.8500+02	7.6882+02	9.5689+03	9.5008+00	2.2560+01	
6.0000+02	8.0890+02	9.6733+03	1.0922+01	2.2182+01	
6.1500+02	8.4945+02	9.7925+03	1.2409+01	2.1810+01	
6.3000+02	8.9052+02	9.9274+03	1.3969+01	2.1442+01	
6.4500+02	9.3220+02	1.0079+04	1.5610+01	2.1078+01	
6.6000+02	9.7454+02	1.0247+04	1.7344+01	2.0718+01	
6.7500+02	1.0176+03	1.0434+04	1.9182+01	2.0361+01	
6.9000+02	1.0615+03	1.0640+04	2.1136+01	2.0007+01	
7.0500+02	1.1063+03	1.0866+04	2.3224+01	1.9654+01	
7.2000+02	1.1521+03	1.1115+04	2.5462+01	1.9304+01	
7.3500+02	1.1990+03	1.1388+04	2.7873+01	1.8955+01	
7.5000+02	1.2470+03	1.1686+04	3.0483+01	1.8606+01	
7.6500+02	1.2964+03	1.2012+04	3.3321+01	1.8259+01	
7.8000+02	1.3472+03	1.2369+04	3.6425+01	1.7911+01	
7.9500+02	1.3995+03	1.2758+04	3.9841+01	1.7563+01	
8.1000+02	1.4535+03	1.3185+04	4.3625+01	1.7214+01	
8.2500+02	1.5094+03	1.3652+04	4.7848+01	1.6864+01	
8.4000+02	1.5674+03	1.4165+04	5.2598+01	1.6512+01	
8.5500+02	1.6276+03	1.4730+04	5.7993+01	1.6158+01	
8.7000+02	1.6902+03	1.5354+04	6.4183+01	1.5802+01	
8.8500+02	1.7556+03	1.6046+04	7.1372+01	1.5443+01	
9.0000+02	1.8240+03	1.6818+04	7.9841+01	1.5079+01	
9.1500+02	1.8959+03	1.7685+04	8.9984+01	1.4711+01	
9.3000+02	1.9716+03	1.8667+04	1.0238+02	1.4338+01	
9.4500+02	2.0516+03	1.9790+04	1.1789+02	1.3959+01	
9.6000+02	2.1367+03	2.1093+04	1.3793+02	1.3572+01	
9.7500+02	2.2277+03	2.2634+04	1.6485+02	1.3176+01	
9.8800+02	2.3123+03	2.4230+04	1.9706+02	1.2823+01	

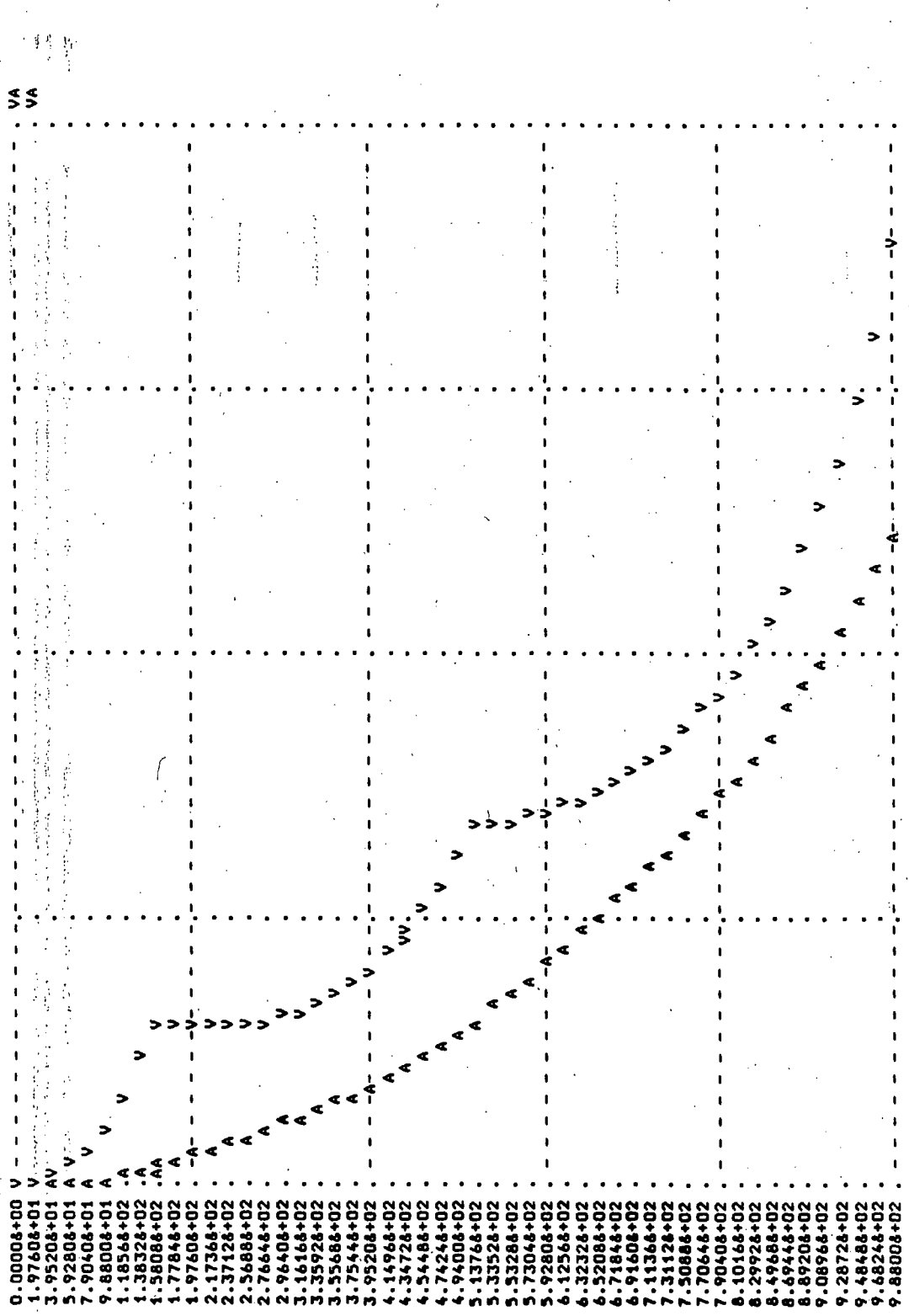
2.3123+03 SLUT

\*\*\* LAUNCH OF THREE-STAGE ROCKET \*\*\*

A=ALT 0.0000E+00 A 5.0000E+06 A 1.0000E+07 A 1.5000E+07 A 2.0000E+07 A  
 V=ALT 0.0000E+00 V 1.0000E+04 V 2.0000E+04 V 3.0000E+04 V 4.0000E+04 V

SCALES OF PLOT

0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 DUPLICATES



T 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 DUPLICATES



EQUATIONS-sektionen er her udvidet med to ligninger, som ikke er differentiaalligninger, nemlig udtrykkene for beregning af  $R_0$  og  $G$ . Det kan ses af udtrykket for  $ALT''$ , at  $G$  og  $R_0$  til enhver tid skal være beregnet, før  $ALT''$  kan beregnes. Dette forhold er det ikke nødvendigt at tage hensyn til ved skrivningen af ODIN-programmet, da systemet selv sørger for at anbringe ligningerne i den rigtige rækkefølge. Såfremt en sådan rækkefølge ikke kan opnås, vil systemet give meddelelse herom.

I programmet optræder et nyt sektionens navn, nemlig sektionen DEFINITIONS. I denne sektion kan man definere sine egne funktioner, i dette tilfælde funktionen  $F$ . Definitionen af  $F$  betyder, at hver gang  $F$  forekommer i programmet, beregnes dens værdi udfra udtrykket i definitionen, idet værdien af de parametre, der optræder i kaldet, indsættes i stedet for de tilsvarende parametre i definitionen. En funktion behøver ikke at have parametre, men kan defineres udelukkende udfra andre variable i programmet eller ved kald af funktioner.

OUTPUT-sektionen er udvidet med den sætning, der skal bruges for at få udskrevet en tabel, nemlig PRINT-sætningen. Den viste PRINT-sætning vil bevirke udskrift af en tabel, som for hver 4 tidsenheder indeholder  $T$ , højden over jorden (i miles), hastigheden (i MPH), accelerationen, tyngdeaccelerationen og lufttætheden.

Ved den første RUN-kommando vil programmet blive kørt, men betingelsen

$$\text{STOPCONDITION} = R_0 < 1 \times 10^{-38}$$

vil bevirke, at kørslen stoppes, når  $R_0$  bliver mindre end  $1 \times 10^{-38}$ . I de tidligere eksempler har RUN været efterfulgt af STOP, men i dette tilfælde ønsker vi at køre videre med  $R_0 = 0$ . Det kan gøres ved for hver sektion at skrive de linier om, der ønskes ændret. I EQUATIONS-sektionen vil vi ændre beregningen af  $R_0$ , til  $R_0 = 0$ , mens vi i INITIAL-sektionen ønsker at lade  $ALT$  og  $ALT'$  starte med de værdier, de sluttede med i sidste kørsel. Dette gøres med linierne  $ALT = ALT$  og  $ALT' = ALT'$ , hvor størrelsen på venstre side af lighedstegnet opfattes som startværdi for den nye kørsel, mens udtrykkene på højre side af lighedstegnet opfattes som resultater fra den sidste kørsel.

I SIMULATION-sektionen angiver vi, at det nye starttidspunkt skal være sidste kørsels stoptidspunkt, nemlig med linien  $SIMSTART = T$ . Vi angiver samtidig en ny stopbetingelse, som er udtryk for, at vi vil stoppe hvis tredje trins masse kommer under 100.

Mens de øvrige sektioner i hver ny kørsel kun ændres med de linier, der enten tilføjes eller ændrer en beregning, skal der i OUTPUT-sektionen angives samtlige de udskriftsfunktioner, man ønsker udført. Det er i dette tilfælde gjort ved at ændre den udskrevne tabel til ikke at indeholde  $R_0$ , samt ved at angive

en ny type udakrifthfunktion, nemlig linien

LAST ALT/5280, 8 SLUTT .

Denne linie bevirker, at der efter den sidste udakrift udskrives værdien af ALT, i miles, efterfulgt af teksten SLUT.

#### Tabellagte funktioner.

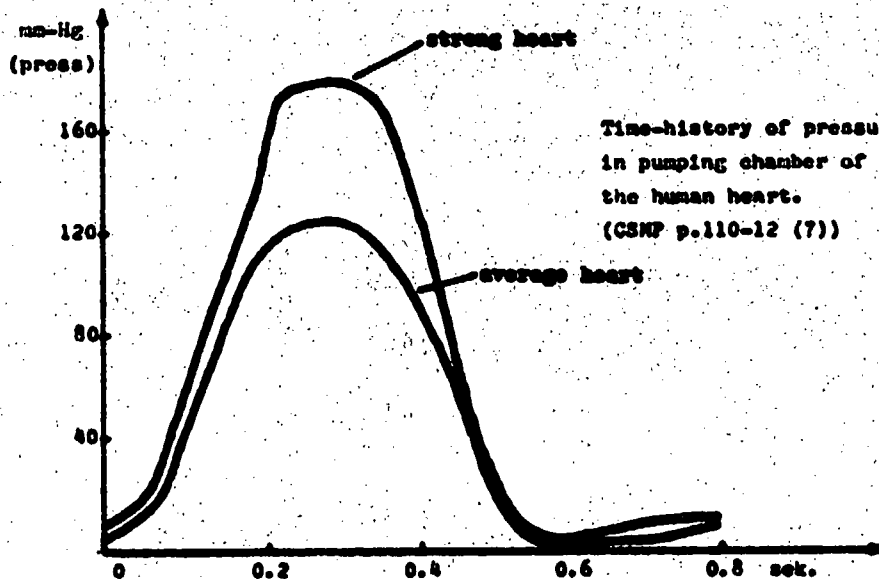
Vi ønsker at undersøge forskellen mellem den mængde blod, som et stærkt hjerte og et mere gennemsnitligt pumper i et puls-slag.

Den mængde blod, som hjertet pumper per tidsenhed (her liter/sek.), kan tilnærmelsesvis udtrykkes således:

$$\text{volumen} = \text{press}/575.$$

Press er (over-) trykket i ventriklerne, de 2 hjertekamre, målt i mm-Hg. og tallet, 575, er en eksperimentelt fastlagt konstant.

Trykket, press, er en funktion af tiden og kendes kun i enkelte (måle-) punkter, f.eks. til "tiden" 0, 0.1, 0.2, ...e.t.c. Til brug for undersøgelsen har vi 2 sådanne sæt (tabellagte) data for trykket i ventriklerne hos henholdsvis en person med et stærkt hjerte og en med et mere gennemsnitligt (se tegningen herunder).



I nedenstående ODIN-program's DEFINITIONS-sektion kan man se, hvordan det udover mulighederne for at definere funktioner udfra beregningsudtryk, som i det forudgående eksempel, er muligt at definere funktioner, som press, der kun kendes i nogle punkter, altså er tabellagte. Disse funktioners værdi i et ikke-tabellagt punkt beregnes ved lineær interpolation mellem de to kendte tabopunkter.

```
DEFINITIONS
TABLE( 0, 0.05, 16): PRESS = 8, 12, 21, 80, 140, 174, 177,
                          175, 165, 138, 86, 19, 3, 4.2, 6, 7, 8

TABLEX : PRESSURE = (0.5) (0.05,8) (0.1,18) (0.15,63) (0.2,100)
                   (0.25,120) (0.3,124) (0.4,120) (0.5, 65)
                   (0.54, 6) (0.56, 4) (0.58, 2) (0.59, 2)
                   (0.61,2.2) (0.65,2.5) (0.7, 3) (0.8, 5)

INITIAL
SHVOLUME = 0 SHVOLUME' = 0
AHVOLUME = 0 AHVOLUME' = 0

EQUATIONS
SHVOLUME' = PRESS(T)/575
AHVOLUME' = PRESSURE(T)/575

SIMULATION
SIMTIME = 0.8
MAXABSERR(SHVOLUME',AHVOLUME') = 0.0001
MAXRELERR(SHVOLUME') = 0.0001
MAXRELERR(AHVOLUME') = 0.00001

OUTPUT
HEAD $                BLOOD FLOW FOR PRESSURE-PULSE $
HEAD $ $
FIRST $                $,
$STRONG HEART         $,

$ AVERAGE HEART $
PRINT(0.04) T, PRESS(T), SHVOLUME, PRESSURE(T), AHVOLUME
PLOT(T, SIMTIME/50) SHVOLUME AS v, SHVOLUME' AS f,
                   AHVOLUME AS v, SHVOLUME' AS f
LAST $ THE FINAL VOLUME WAS:$
LAST $ $
LAST $ STRONG HEART VOLUME = $, SHVOLUME, $ LITRE$
LAST $ AVERAGE HEART VOLUME = $, AHVOLUME, $ LITRE$

RUN

STOP
```

Som det fremgår af programmet, er der mulighed for at definere to typer tabellagte funktioner, nemlig funktioner med ækvidistante indgangspunkter, PRESS, det stærke hjerte, og med ikke-ækvidistante indgangspunkter, PRESSURE, det mere gennemsnitlige hjerte.

En tabel med ækvidistante indgangspunkter defineres på følgende måde:

TABLE(<regneudtryk>,<regneudtryk>,<regneudtryk>): H = tal, tal,...,tal.

De tre parametre, <regneudtryk>, angiver startpunktet for tabellens indgangsværdier, 0 i programeksemplet, længden af intervallerne mellem indgangsværdierne, i eksemplet 0.05, samt antallet af intervaller, i eksemplet 16. Navnet efter ':', her 'H' og i programeksemplet 'PRESS', angiver tabellens navn. Rækken af tal efter lighedstegnet skal være tabelværdierne svarende til de ækvidistante indgangsværdier. Der skal altså være et tal mere end antallet af intervaller, dvs. 17 tal i programeksemplet. Værdien af `press(0.025)` for det stærke hjerte bliver 10, da de "omgivende" tabellagte nabopunkter er `PRESS(0)` og `PRESS(0.05)`, som er henholdsvis 8 og 12. Udenfor det interval, som man har defineret funktionen for, for `PRESS` er det  $(0, 16 \cdot 0.05 = 0.8)$ , vil den antage værdien for det nærmeste endepunkt.

En tabel med ikke-ækvidistante indgangsværdier defineres således:

TABLEX : TAB = (tal,tal) (tal,tal) (tal,tal)... ..(tal,tal) ,

hvor TAB bliver tabellens navn, og talparrene (tal,tal) er en indgangsværdi henholdsvis tabelværdi for hvert punkt. `PRESSURE(0.025)` for det mere gennemsnitlige hjerte bliver således 6.5. Udenfor intervallet  $(0,0.8)$  vil værdien blive den i det nærmeste endepunkt, ligesom ovenfor.

TABLEX kan naturligvis altid anvendes, uanset om punkterne er ækvidistante men hvis de er talrige og ækvidistante, så kan der spares en del skrivearbejde ved at anvende TABLE.

I SIMULATION-sektionen er det vist, hvordan man kan specificere den nøjagtighed, som skal være opfyldt ved den numeriske beregning af de to volumener. I det viste eksempel er kravet til den maksimalt tilladelige absolutte fejl for SHVOLUMEN og AHVOLUMEN altså den samme, nemlig 0.0001, mens den maksimalt tilladelige relative fejl ved den numeriske beregning er henholdsvis 0.0001 og 0.00001.

Det er, som før nævnt, næsten umuligt at sige noget generelt om valget af den absolutte og relative fejl, men det kan være en fordel at specificere en ekstra lille absolut fejl for en variabel, der bevæger sig i nærheden af nul, og det kan være en fordel med en lille relativ fejl for variable med (numerisk) høje værdier.

BLOOD FLOW FOR PRESSURE-PULSE

T	STRONG HEART		AVERAGE HEART	
	PRESS	SHVOLUME	PRESSURE	AHVOLUME
0.0000&+00	8.0000&+00	0.0000&+00	5.0000&+00	0.0000&+00
4.0000&-02	1.1200&+01	6.3616&-04	7.4000&+00	4.0936&-04
8.0000&-02	1.7400&+01	1.6103&-03	1.4000&+01	1.1254&-03
1.2000&-01	4.4600&+01	3.3477&-03	3.6000&+01	2.5745&-03
1.6000&-01	9.2000&+01	8.0415&-03	7.0400&+01	6.2953&-03
2.0000&-01	1.4000&+02	1.6117&-02	1.0000&+02	1.2231&-02
2.4000&-01	1.6720&+02	2.6773&-02	1.1600&+02	1.9727&-02
2.8000&-01	1.7800&+02	3.8932&-02	1.2240&+02	2.8119&-02
3.2000&-01	1.7600&+02	5.1258&-02	1.2400&+02	3.6709&-02
3.6000&-01	1.7300&+02	6.3407&-02	1.2320&+02	4.5319&-02
4.0000&-01	1.6500&+02	7.5157&-02	1.2000&+02	5.3740&-02
4.4000&-01	1.4340&+02	8.6028&-02	9.8000&+01	6.1448&-02
4.8000&-01	1.0680&+02	9.4858&-02	7.6000&+01	6.7499&-02
5.2000&-01	5.9200&+01	1.0063&-01	3.5500&+01	7.1296&-02
5.6000&-01	1.5800&+01	1.0335&-01	4.0000&+00	7.2849&-02
6.0000&-01	3.0000&+00	1.0393&-01	2.1000&+00	7.2884&-02
6.4000&-01	4.0000&+00	1.0405&-01	2.4250&+00	7.2843&-02
6.8000&-01	5.3333&+00	1.0438&-01	2.8000&+00	7.3025&-02
7.2000&-01	6.4000&+00	1.0479&-01	3.4000&+00	7.3238&-02
7.6000&-01	7.2000&+00	1.0527&-01	4.2000&+00	7.3501&-02
8.0000&-01	8.0000&+00	1.0579&-01	5.0000&+00	7.3823&-02

THE FINAL VOLUME WAS:

STRONG HEART VOLUME = 1.0579&-01 LITRE  
AVERAGE HEART VOLUME = 7.3823&-02 LITRE

\*\*\* BLOOD FLOW FOR PRESSURE-PULSE \*\*\*

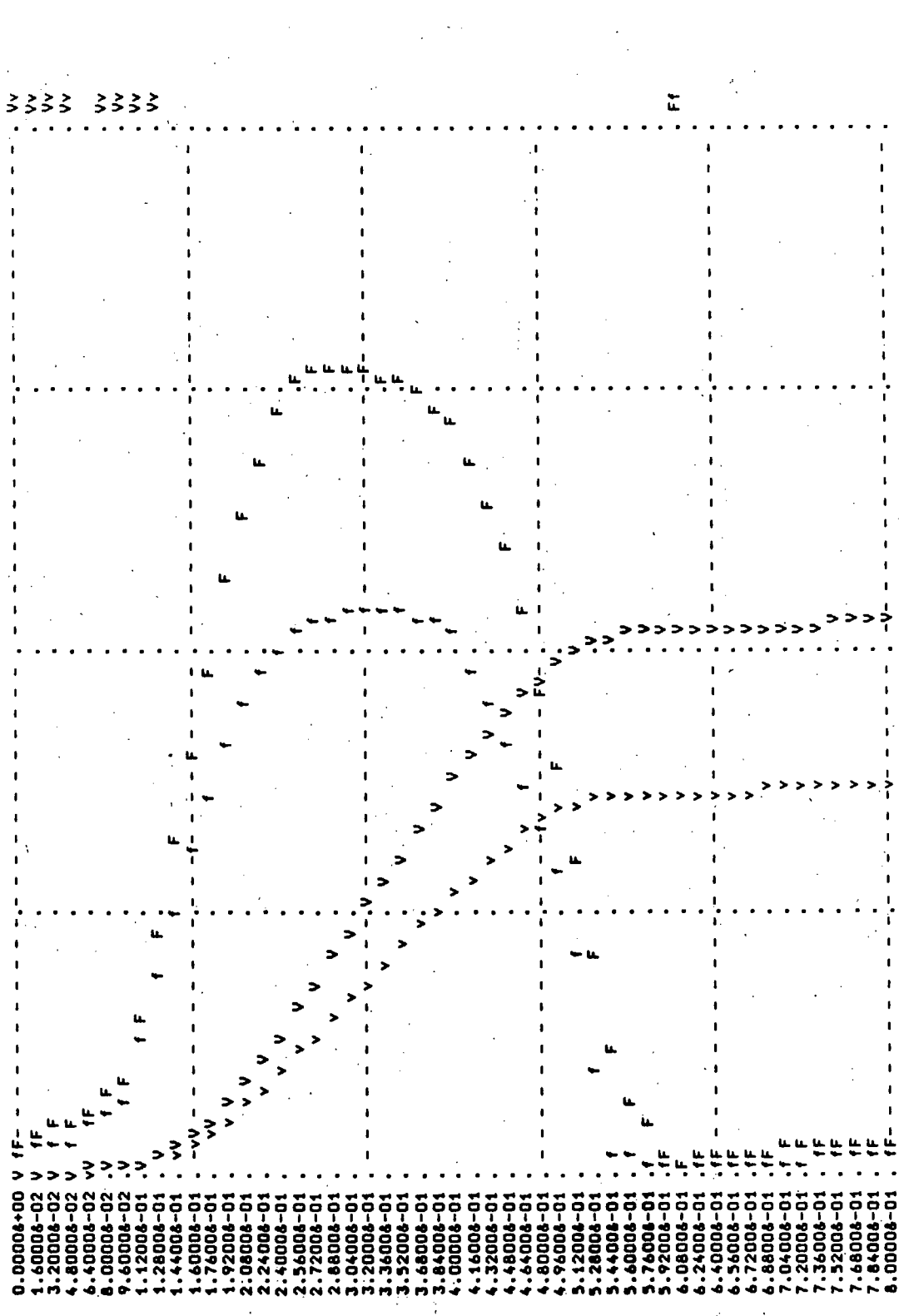
F=SHVOLUME' 0.00006+00 F  
 V=SHVOLUME 0.00006+00 V  
 f=AHVOLUME' 0.00006+00 f  
 v=AHVOLUME 0.00006+00 v

SCALES OF PLOT  
 1.00006-01 F  
 2.00006-01 V  
 1.00006-01 f  
 2.00006-01 f  
 1.00006-01 v

3.00006-01 F  
 1.50006-01 V  
 3.00006-01 f  
 1.50006-01 v

4.00006-01 F  
 2.00006-01 V  
 4.00006-01 f  
 2.00006-01 v

100 DUPLICATES



Anvendelsen og funktionen af HEAD, FIRST og LAST er forsøgt eksemplificeret af udskriften herover. I den forbindelse bør det fremhæves at HEAD giver en overskrift (kun tekster) øverst på hver side af det output, der ikke er plot, mens FIRST kun giver anledning til "en udskrift", startende på første side af output'et. PRINT-sætningen resulterer i, at der bliver udskrevet en tabel med værdien af T, PRESS(T), SHVOLUME, PRESSURE(T) og AHVOLUME, for hver 0.04 tidsenheder.

```
PLOT(T,SIMTIME/50) SHVOLUME AS V, SHVOLUME' AS F,  
AHVOLUME AS v, AHVOLUME' AS f
```

bevirker ,at der bliver udskrevet et plot (graf/kurve) med T som abscisse og SHVOLUME, SHVOLUME', AHVOLUME og AHVOLUME' som ordinat. Af plottet kan man se, at der for hvert tidsspring - tiderne står i kolonnen til venstre langs abscisseaksen - bliver printet punkter, markeret med et 'V', et 'F', et 'v' og et 'f', på graferne. Hvis man i stedet (eller også) havde ønsket at få et plot over SHVOLUME i forhold til en anden uafhængig variabel end T, for eksempel SHVOLUME' eller PRESS(T), så er det nødvendigt, at PLOT-sætningen har 3 parametre, da tiden i dette tilfælde ikke er abscisse/ordinat, og derfor skal der angives, hvor tit (i tidsenheder) d.v.s med hvilken frekvens, der skal opsamles information til plottet (jævnfør hare-ræve eksemplet på s.58). Det betyder altså, at nedenstående PLOT-sætning:

```
PLOT(SHVOLUME', 0.005, 0.2) SHVOLUME AS V
```

ville resultere i, at der blev udskrevet et plot med SHVOLUMEN' som abscisse, SHVOLUME som ordinat, og for hver 0.005 tidsenheder ville der blive opsamlet information til en markering på plottet, som ville have 0.2 enheder mellem hver linie på abscisse-aksen. Inddelingen af ordinat-aksen foretages automatisk, så linieskriverens papirbredde, med 100 printpositioner for selve plottet, udnyttes bedst muligt.

#### Yderligere muligheder i sproget.

I DEFINITIONS-sektionen er der også mulighed for at specificere en funktion som værende en extern procedure, hvorved det er muligt at benytte programmer, skrevet i andre programmeringssprog, til nogle af beregningerne. Dette vil kun være nødvendigt i helt specielle tilfælde, og for den interesserede bruger vil fremgangsmåden fremgå af syntaksgraferne sidst i denne brugervejledning.

#### Fejlmeddelelser fra systemet.

Der skelnes mellem tre principielt forskellige slags fejl. Den ene type fejl er brud på reglerne for, hvorledes man skal udtrykke sig i sproget, såkaldte syntaksfejl. Den anden type fejl opstår, hvis systemet ikke er i stand til at

anbringe ligningerne i en sådan rækkefølge, at alle størrelser er beregnet, inden de indgår i en beregning. Disse fejl kaldes algebraiske løkker. Den tredje type fejl, der kaldes runtime-fejl, kan opstå under selve kørslen af programmet. Der kan f.eks. være tale om, at nævneren i en beregning efterhånden bliver så lille, at der til sidst sker division med nul. Da oversætteren knapt er færdig, er det endelige udsende af disse fejlmeddelelser endnu ikke fuldstændigt fastlagt.



Liste over standardfunktioner.

Aritmetiske funktioner.

ABS(X)	den numeriske værdi af X.
ARCCOS(X)	$\cos^{-1}(X)$ , resultat i radianer i intervallet $(0, \pi)$ .
ARCSIN(X)	$\sin^{-1}(X)$ , resultat i radianer i intervallet $(-\pi/2, \pi/2)$ .
ARCTAN(X)	$\tan^{-1}(X)$ , resultat i radianer i intervallet $(-\pi/2, \pi/2)$ .
ARCTAN(X,Y)	$\tan^{-1}(X/Y)$ , resultatet i radianer i intervallet $(0, 2\pi)$ .
CBRT(X)	kubikroden af X.
COS(X)	X i radianer.
COSH(X)	hyperbolsk cosinus, X i radianer.
ENTIER(X)	det største heltal, som ikke er større end X.
EXP(X)	exponentialfunktionen.
LN(X)	den naturlige logaritme.
LOG(X)	titalslogaritmen.
MOD(X,Y)	X og Y heltal, X mod Y når X og Y er positive.
SIGN(X)	1 for X > 0, 0 for X = 0, -1 for X < 0.
SIN(X)	X i radianer.
SINH(X)	hyperbolsk sinus, X i radianer.
SQRT(X)	kvadratroden af X.
TAN(X)	X i radianer.
TANH(X)	hyperbolsk tangens, X i radianer.

Tilfældig-tals generatorer.

De tilfældige tal fra hver funktion udgør en sekvens af tal, hvor startpunktet i sekvensen beregnes ud fra parametren U. Man får altså samme sekvens hver gang for samme U. U skal have ulige startværdi.

DRAW(A,U)	returnerer TRUE med sandsynligheden A, FALSE med sandsynligheden 1-A.
ERLANG(A,B,U)	trækker et tal fra ERLANG-fordelingen med middelværdi 1/A og med standardafvigelse $1/(A \cdot \text{SQRT}(B))$ .
NEGEXP(A,U)	trækker et tal fra en negativ exponentialfordeling med middelværdi 1/A.
NORMAL(A,B,U)	trækker et tal fra en normal-fordeling med middelværdi A og standardafvigelse B.
POISSON(A,U)	trækker et tal fra Poisson-fordelingen med parametren A.
RANDINT(A,B,U)	trækker et heltal i intervallet (A,B) med ligestør sandsynlighed. A og B skal være heltal.
UNIFORM(A,B,U)	trækker et tal ligeligt fordelt i intervallet (A,B).

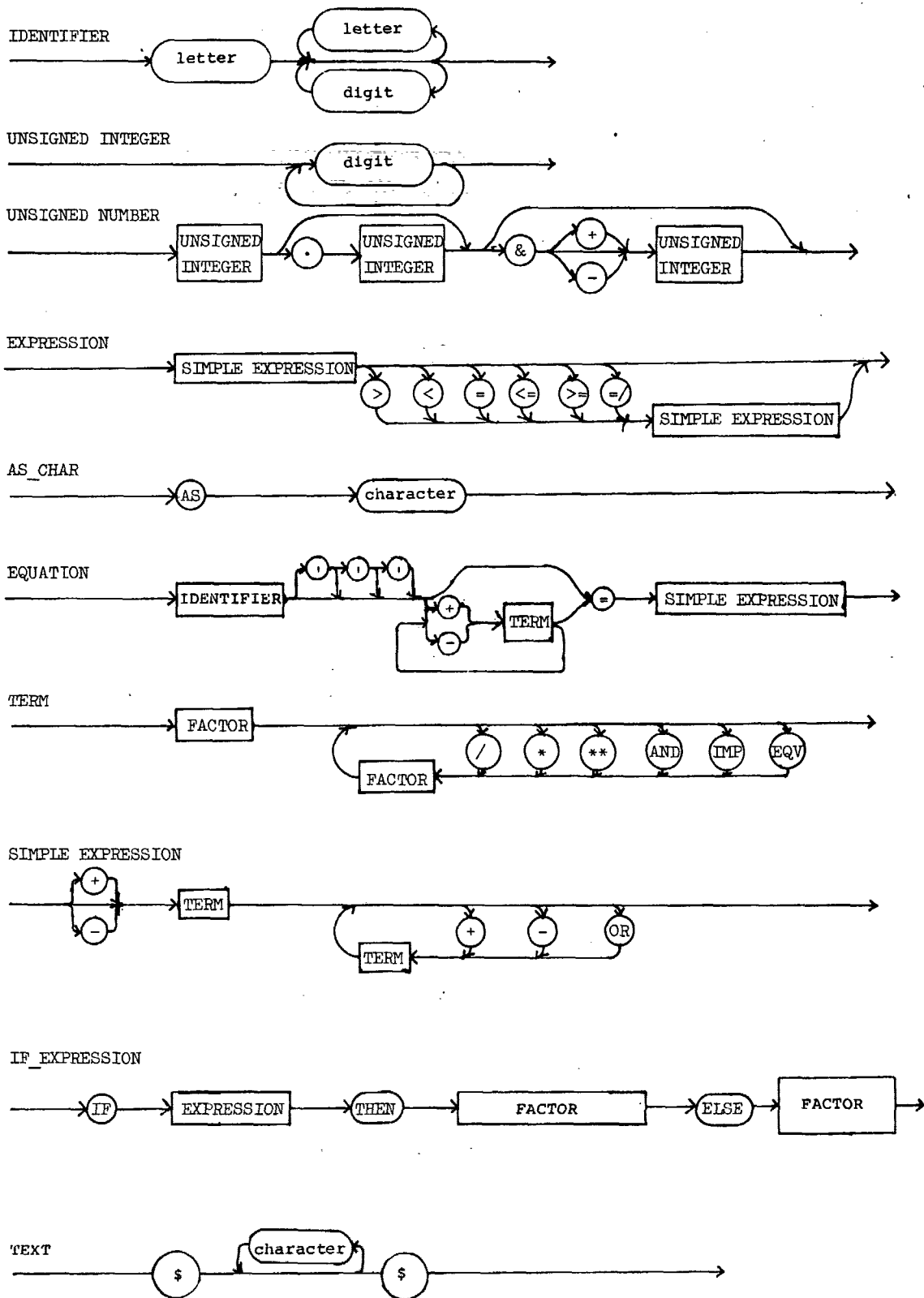
Se evt. SIMULA, PROGRAMMER'S REFERENCE MANUAL, UNIVAC EXEC 8, Universitetsforlaget, Oslo 1972, for yderligere forklaring af standardfunktionerne.

### Syntaksgrafer.

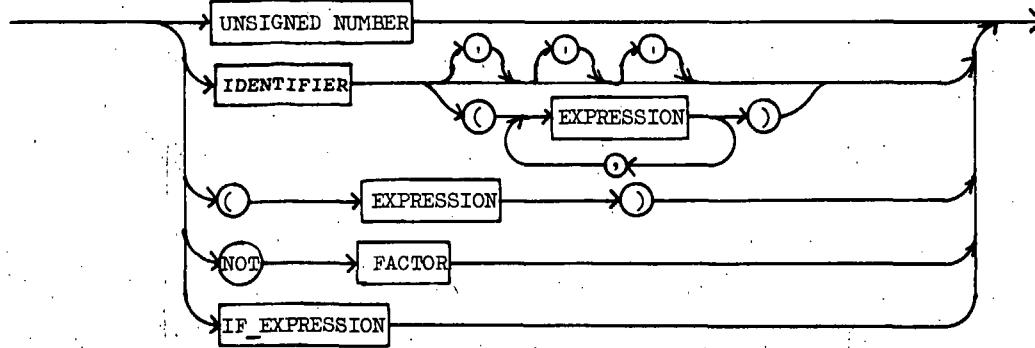
Ved hjælp af syntaksgraferne "PROGRAM" samt de tilhørende undergrafer herunder kan man undersøge, om de syntaktiske regler, gældende for ODIN-programmer, er overholdt. Graferne kan altså ikke bruges til at afgøre, om de semantiske regler er overholdt.

Når man vil undersøge, om et program overholder de syntaktiske regler, så starter man med grafen "PROGRAM" og undersøger, idet man gennemløber graferne, om det er muligt at danne en tekst svarende til den undersøgte, ved at gennemløbe graferne på passende måde. Hvis det er muligt, har man et såkaldt syntaksfejlfrigt ODIN-program. Når man gennemløber graferne, skal man være opmærksom på følgende: Alle (firkantede) kasser erstattes med undergrafer, som har det navn, der står i kassen, og efter gennemløbet af en undergraf fortsættes på det oprindelige sted i grafen, men lige efter "kassen". Alle "ovale kasser" eller cirkler skal, helt bogstaveligt, læses/opfattes som de(t) tegn, der står i den. Dog skal det med små typer skrevne, som "letter", "digit" og "character", opfattes som et vilkårligt element af den beskrevne type.

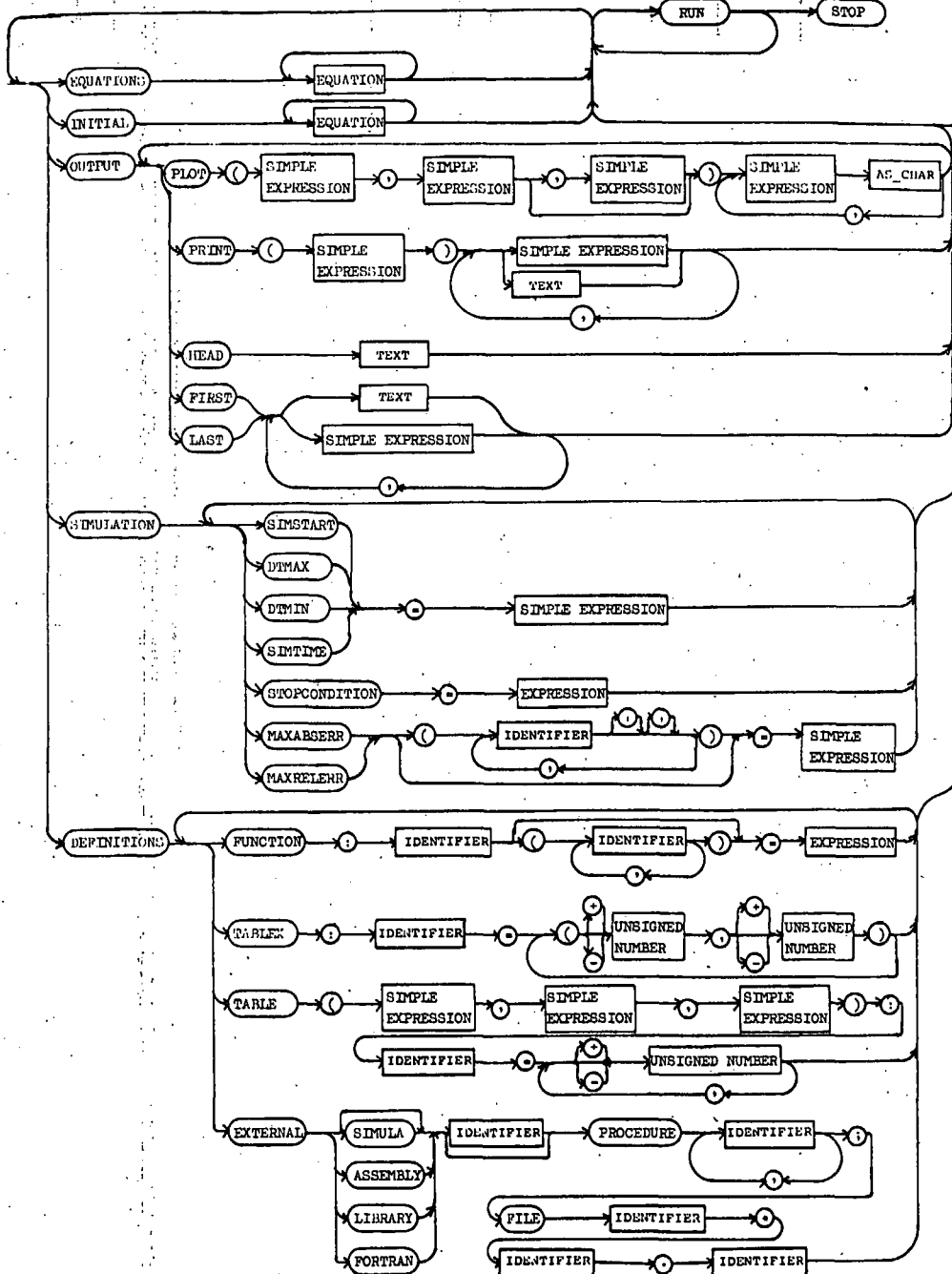
For en nøjere beskrivelse af tydningen og konstruktionen af syntaksgrafer henvises til Niklaus Wirth (12).



FACTOR



PROGRAM



## APPENDIX 2.: KURSUSBESKRIVELSE.

Denne rapport er tænkt som delmateriale til et kursus i differentiaalligningsmodeller, som afholdes på RUC-NAT-Basis. Imidlertid er materialet kun udarbejdet til brug ved den afsluttende del af kurset, hvor opstilling af modeller og simulering af samme indføres. Anvendelsen af dette kursusmateriale forudsætter således, at kursUSDeltagerne har et elementært kendskab til differentiaalligninger og deres algebraiske løsninger, på et niveau som mindst svarer til specialelæsning af differentiaalligninger i 3.g-Mat/Fys. Det forudsættes ligeledes, at kursUSDeltagerne er bekendt med begrebet kausaldiagrammer.

Dette kursus har som formål at give deltagerne en indsigt i differentiaalligningsmodellers anvendelse og samtidigt sætte dem i stand til at kunne drage konklusioner udfra simple modeller, de selv har opstillet.

Dette delkursus er opdelt i 4 perioder A, B, C og D, som igen er opdelt i henholdsvis 1, 2, 3 og 4 lektioner af ca. 2.5 time. Deltagerne tænkes opdelt i grupper på 3-4 personer. Endvidere påtænkes en vejleder at cirkulere imellem grupperne og styre evt. afsporede diskussioner i en konstruktiv retning.

PERIODE A, har som formål at give deltagerne en fornemmelse af hvorfor og hvor, modeller anvendes. Specielt at differentiaalligningsmodeller i visse situationer kan være et kraftfuldt værktøj.

PERIODE B, har som formål at give deltagerne en forståelse for, hvordan man anvender differentiaalligningsmodeller. Her, specielt, at det oftest er nødvendigt at anvende numeriske løsningsmetoder for at kunne foretage simuleringer. Desuden vil der i denne periode blive introduceret differentiaalligningssystemer.

PERIODE C. Programmeringssproget ODIN introduceres, og der opstilles en model på baggrund af et simpelt differentiaalligningssystem. Modellen afprøves og køres på en datamaskine.

PERIODE D. På baggrund af et givet problem - eks.3 bytte-/rovdyr - skal deltagerne selv opstille system og model og udfra simuleringer med modellen drage konklusioner angående systemet.

De følgende "lektioner" uddeles ved de enkelte lektioners begyndelse.

Periode A, lektion 1.

Forudgående litteratur: ingen.

1. Hvor og hvorfor anvendes modeller?

Diskuter hvor modeller kan anvendes, og prøv at begrunde hvorfor. Hvordan kan man f.eks. tænke sig, at DSB kunne anvende modeljernbaner med fordel, eller Lindø-værftet skibsmodeller?

2. Hvad er en matematisk model og herunder specielt en differentiaalligningsmodel? På hvilken måde indgår f.eks. tiden i disse beskrivelser? (Hvad beskriver de meget omtalte fiskerimodeller? og hvad bruges de til?).

Udlevering af litteratur: Kapitel 2.

Periode B, lektion 2.

Forudgående litteratur: Kapitel 2.

1. Opstil en differentiaalligningsmodel over rygtespredning (sladder) i en population på  $N$  individer.

I systemafklaringen må I bl.a. tage stilling til:

- hvordan spredes rygтет?
- hvem spreder rygter (er det alle, eller er de lige gode til det)?

Problemet er at finde antallet af individer, der kender rygтет til tiden  $T$ , altså som funktion af tiden ( $s(T)$ ). Vælg selv værdier for  $N$  og de konstanter, som evt. indgår. (givet: eet individ kender rygтет til tiden 0;  $s(0) = 1$ ).

2. Når differentiaalligning-(n/rne) er opstillet, vil I sikkert kunne konstatere, at de(n) ikke (umiddelbart) kan løses algebraisk. Vi må derfor anvende en anden metode, en såkaldt numerisk metode.

Derfor skal I nu udvikle en sådan metode og ved hjælp af den løse ovenstående problem.

3. Det, I skal gøre, er at skitsere kurven  $s(T)$ 's forløb grafisk udfra de ting, som I ved om funktionen.

Fig.1

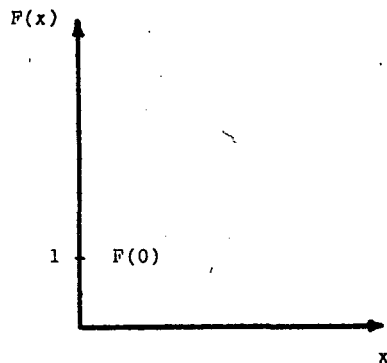


Fig.2

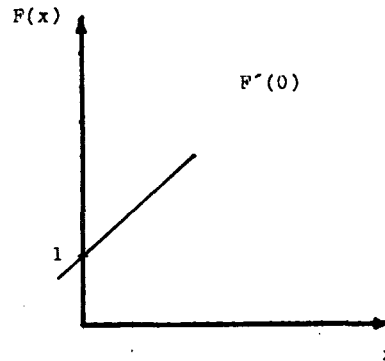


Fig.3

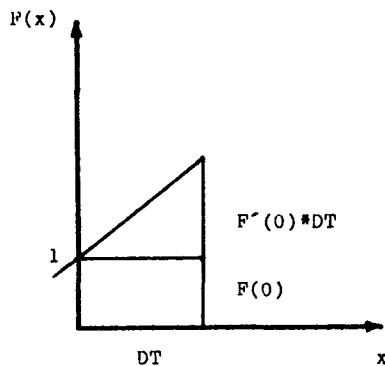
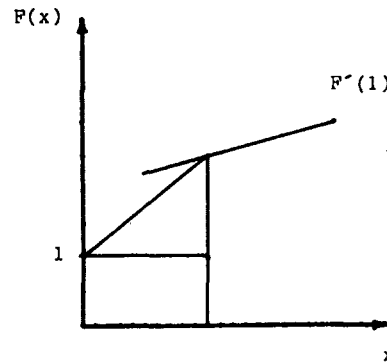


Fig.4



- 1)  $s(0) = 1$
- 2) Udfra den differentiaalligning, som I har opstillet, kan  $s'(0)$  beregnes.
- 3) Ved at tage et "tidsskridt" (DT) f.eks. på 1 tidsenhed og antage at funktionsværdien ( $s(1)$ ) ligger på tangenten ( $s'(0)$ ) kan vi beregne  $s(1)$  som  $s(0) + s'(0) \cdot DT$ , hvor DT her er 1. Er antagelsen rimelig?
- 4) Når vi nu kender  $s(1)$ , kan vi beregne  $s'(1)$  og derved finde  $s(2)$ ...etc.

4. I har sikkert iagttaget, at metoden ikke er særlig nøjagtig. Men hvad sker der, hvis I gør skridtlængden større henholdsvis mindre?

Udlevering af litteratur: Kapitel 3.

Periode B, lektion 3.

Forudgående litteratur: Kapitel 2 og 3.

1. Diskuter evt. problemer i de udleverede kapitler.
2. Færdiggør det i den foregående lektion stillede rygte-problem.
3. Diskuter om resultatet af rygte-problemet
  - ser rimeligt ud og
  - om modellen virker troværdig.

Udlevering af litteratur: Kapitel 4.

Periode C, lektion 4.

Forudgående litteratur: Kapitel 2,3 og 4.

1. Diskuter evt. problemer i kapitel 4.
2. Opstil en differentiaalligningsmodel for dobbelt radioaktivt henfald (find selv et eksempel).
3. Løs ligningssystemet algebraisk.
4. Skriv et program i ODIN, som kan simulere situationen, d.v.s. find en numerisk løsning.

Udlevering af litteratur: Appendix 1 (brugervejledning).

Periode C, lektion 5+6.

Forudgående litteratur: Kapitel 2, 3, 4 og appendix 1.

1. Diskuter evt. problemer ved læsningen af brugervejledningen.
2. Færdiggør henfaldsprogrammet og kød det.
3. Kontroller om det simulerede resultat stemmer overens med det algebraiske. Ellers find fejlen!

Udlevering af litteratur: ingen.

Periode D, lektion 7.

Forudgående litteratur: Kapitel 2, 3, 4 og appendix 1.

1. På en ellers øde ø eksisterer der 3 levende organismer, græs, geder og hjorte. Både gederne og hjortene lever af græsset, de konkurrerer altså om føden.



Simuler dette økologiske system og afgør, om der kan opstå en ligevægtssituation, hvor ingen af organismene er uddøde. Skriv et ODIN-program.

Udlevering af litteratur: Kapitel 5.

Periode D, lektion 8+9+10.

Forudgående litteratur: Kapitel 2, 3, 4, 5 og appendix 1.

1. Diskuter hvordan de i kap. 5 anvendte teknikker kan hjælpe jer videre.
2. Hvilke lighedspunkter er der mellem jeres system og de i kap.5 omtalte?
3. Prøv i lyset af ovenstående diskussion at diskutere om det kan bruges i jeres fortsatte arbejde med modellen.

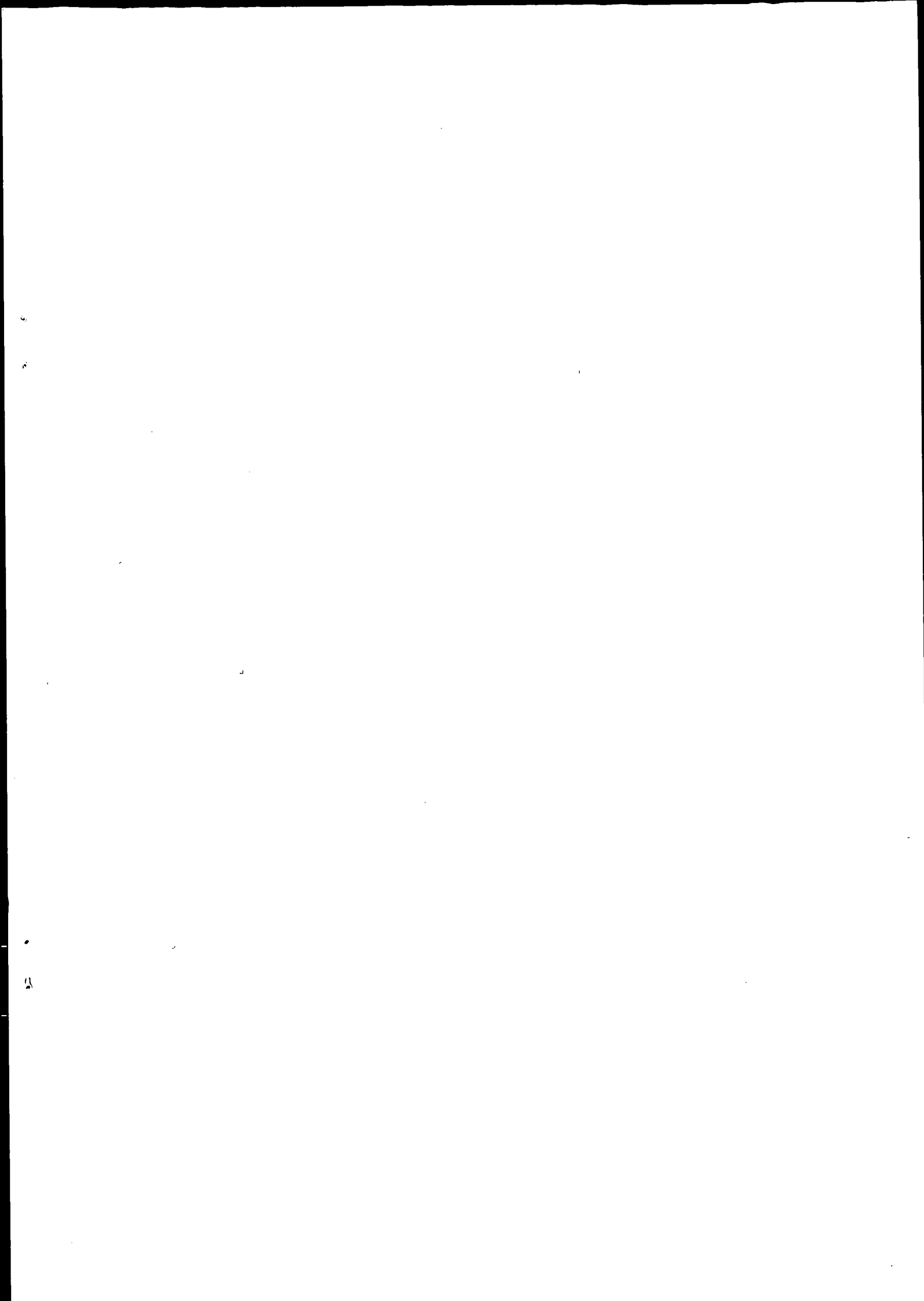
Ingen yderligere udlevering af litteratur.

Foreslået evalueringskriterium: aflevering af en rapport over rygte-problemet og det økologiske problem.

Referenceliste.

1. Ludwig von Bertalanffy : Perspective on General Systems Theory, George Braziller, 1975, s.1-52.
2. Bernard P. Ziegler : Theory of Modeling and Simulation, John Wiley, 1976.
3. Nathan Keyfitz : Introduction to the mathematics of population, Addison-Wesley, 1968, s.271-292.
4. Robert M. May : Stability and Complexity in Model Ecosystems, Princeton University Press, 1973.
5. Thomas Park : "Beetles, competition, and population" in W.E. Hazen, ed., Readings in Population and Community Ecology, W.B.Saunders Co., 1964, s.132-142.
6. G.A.Korn, J.V.Wait : Digital Continuous System Simulation, Prentice-Hall, 1978.
7. F.H.Speckhart, W.L.Green : A guide to using CSMP, the continuous system modeling program, Prentice-Hall, 1976.
8. Erik Ursin : En fysiologisk vækstmodel baseret på opbygning og nedbrydning, ikke udg. piece, Charlottenlund, 1.6.1972.
9. K.P.Andersen, E.Ursin : "A Multispecies Extension to the Beverton and Holt Theory of Fishing ...." i Meddelelser fra Danmarks fiskeri og havundersøgelser, vol.7, s.319-435, DIFMR, 1977.
10. G.M.Birtwistle et al. : SIMULA BEGIN, Studentlitteratur, 1973.
11. R.W.Hamming : Numerical Methods for Scientists and Engineers, 2nd Edition, McGraw-Hill, 1973, s.657-676.
12. Niklaus Wirth : Algorithms + Data Structures = Programs, Prentice-Hall, 1976, s.280-291.
13. K. Hølsgaard : COMBINEDSIMULATION. Brugerhåndbog., Intern publikation, RUC, oktober 1978.
14. Peter Lassen : ET DIFFERENTIALIGNINGSSPROG, En 3. moduls datalogirapport, RUC, maj 1979.







Rex Subis  
A 7 I K K  
rglmpa 4M  
IBNI