

TEKST NR 173

1989

NUMERISKE ALGORITMER

af: MOGENS BRUN HEEFELT

TEKSTER fra

IMFUFA

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

IMFUFA, Roskilde Universitetscenter, Postbox 260, 4000 Roskilde

NUMERISKE ALGORITMER

af: Mogens Brun Heefelt

Pris: 16 kr. + porto

IMFUFA tekst nr. 173/89

53 sider

ISSN 0106-6242

ABSTRACT:

Denne tekst er et udkast til en bog for det matematisk-datalogiske emne i 3. gymnasieklasses matematikundervisning.

Den endelige version vil foreligge i efteråret 1989, efter at denne tekst har været afprøvet i gymnasieklasser i foråret 1989.

FORORD

Denne bog giver en måde at udfylde rammerne for det matematisk datalogiske emne i 3. gymnasieklasses matematikundervisning. Bogens første kapitel kan dog udmærket læses i tilknytning til relevante emner i 1. og 2. gymnasieklasse.

Sigtet med bogen er at beskrive nogle numeriske algoritmer, som belyser væsentlige samspil mellem matematiske og datalogiske synsvinkler. I kapitel 1 behandles algoritmer til nulpunktsbestemmelse, og i kapitel 2 behandles algoritmer til numerisk løsning af differentiaalligninger. I appendix gennemgås Taylors formel, som benyttes i begge kapitler.

I tilknytning til bogen er fremstillet en diskette med COMAL-programmer. Derfor er der i bogen kun aftrykt relevante dele af enkelte af programmerne. Programmerne skal ligeledes belyse det omtalte samspil mellem matematiske og datalogiske synsvinkler, og de er derfor gjort korte og overskuelige. Men de er ikke nødvendigvis optimale fra et datalogisk synspunkt.

Bogen forudsætter et indledende kendskab til programmeringssproget COMAL, men den er ikke en videregående lærebog i COMAL. Programmeringsmæssige bemærkninger (eller fif) er således kun medtaget i det omfang, hvor de kan medvirke til at belyse det omtalte samspil.

IMFUFA, 1988

Mogens Brun Heefelt

Indholdsfortegnelse

	side
1. ALGORITMER TIL NULPUNKTSBESTEMMELSE	1
1. 1 Et eksempel på graftegning - brug af FUNC og PROC.	1
1. 2 Algoritmerne bisektion og regula falsi.	3
1. 3 Newton-Raphson algoritmen.	7
1. 4 Konvergens og regnehastighed.	10
1. 5 Regula falsi defineret ved FUNC	12
1. 6 Total nulpunktsbestemmelse for en funktion.	13
1. 7 Graf og nulpunkter for $f(x,y) = 0$.	14
2. ALGORITMER TIL NUMERISK LØSNING AF DIFFERENTIALLIGNINGER	20
2. 1 Taylorudvikling af en løsning til den autonome differentiaalligning.	20
2. 2 Runge-kutta algoritmer.	21
2. 3 Almindelig 4.ordens runge-kutta og dens implementering som procedure.	25
2. 4 Fejlvurdering og skridtlængde.	27
2. 5 Sammenligning af en numerisk og en analytisk løsning til Bertalanffy's ligning.	31
2. 6 Regnehastighed af forskellige algoritmer.	32
2. 7 Anden ordens differentiaalligninger og flere koblede differentiaalligninger.	33
2. 8 Numerisk og analytisk løsning af modellen for "dobbelt radioaktivt henfald".	37
2. 9 Grafisk fremstilling af numeriske løsninger.	39
2.10 Afsluttende bemærkninger.	41
APPENDIX: Taylors formel	43
Opgaver	47
Litteraturliste	53
Liste over COMAL-programmer	53

1. ALGORITMER TIL NULPUNKTSBESTEMMELSE

I dette kapitel vil der blive beskrevet tre algoritmer til bestemmelse af nulpunkter for kontinuerte og differentiable funktioner. Disse algoritmer er bisektion, regula falsi og newton-raphson. Algoritmerne vil blive introduceret på en sådan måde, at deres iterative struktur klart fremtræder. Denne indfaldsvinkel er valgt, fordi det letter en datalogisk implementering. I tilknytning til algoritmerne vil især blive fokuseret på begreberne konvergens, antal iterationer og regnehastighed. Disse begreber vil blive belyst ved såvel matematiske som datalogiske argumenter.

Kapitlet vil blive afsluttet med en belysning af, hvordan sådanne algoritmer kan indbygges som procedurer i større programmer. Dette sker dels i et program til graftegning og total nulpunktsbestemmelse for en funktion og dels i et program, der fastlægger graf og nulpunkter for en implicit given funktion $f(x,y) = 0$.

1.1 Et eksempel på graftegning - brug af FUNC og PROC.

Når man skal bestemme nulpunkter for en funktion, kan det være en fordel at få et overblik over funktionens graf. Dette kan fx. gøres ved at lave et dataprogram, som udtegner denne graf. Dette giver samtidig anledning til at sige lidt om strukturering af dataprogrammer, et emne vi skal vende mere tilbage til. Når man har funktionens graf, ved man bedre, hvor man skal lede efter nulpunkter, og hvor mange man kan håbe på at finde.

I det følgende behandles kun nulpunkter for en funktion. Fx. kan det at bestemme skæringspunkter mellem funktionerne f_1 og f_2 givet ved

$$f_1(x) = e^x \quad \text{og} \quad f_2(x) = x^3 + 1$$

omformuleres til at bestemme nulpunkter for funktionen $f = f_1 - f_2$, der er fastlagt ved

$$(1.1) \quad f(x) = e^x - x^3 - 1.$$

Når man skal lave et program, er det vigtigt at give det en struktur, som er let at overskue. Dette kan bl.a. gøres ved at definere nogle af operationerne som procedurer - dvs ved at benytte FUNC og PROC. Således kan funktionen f i (1.1) defineres ved

```
0280 FUNC f(x) CLOSED
0290   y:=EXP(x)-x*x*x-1 //      funktionen defineres
0300   RETURN y
0310 ENDFUNC f
```

Det er især en fordel at definere de anvendte funktioner som procedurer, da man så kan kalde dem på sædvanlig vis, fx.

```
0150 y:=f(x) // funktionen kaldes igen
```

Bemærk, at FUNC f er lukket (CLOSED), hvilket giver den fordel, at man i et program kan benytte de samme variabelnavne både lokalt i proceduren og globalt i programmet.

Da dette program skal tegne, må man kalde COMAL's grafikpakke ind. Dette sker ved

```
0060 USE graphics // kalder grafikpakken
```

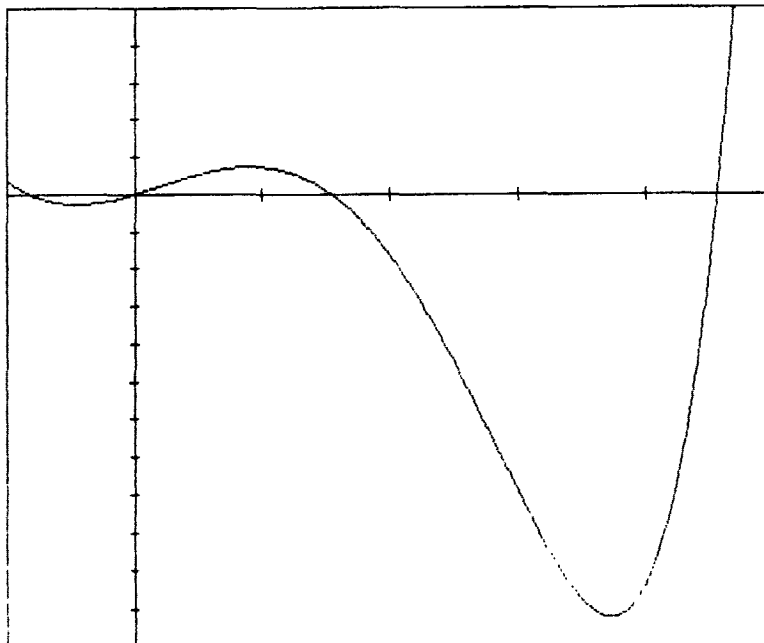
hvorefter man kan benytte alle de procedurer, som er indeholdt i grafikpakken. Ønsker man flere faciliteter i sin grafik, kan man definere det som nye procedurer. En procedure, som tegner x- og y-aksen med skæring i (0,0) vil have følgende udseende

```
0330 PROC axes(b1,b2,c1,c2) CLOSED // procedure der tegner akserne
0340   IMPORT moveto,drawto
0350   moveto(b1,0)
0360   drawto(b2,0)
0370   moveto(0,c1)
0380   drawto(0,c2)
0390 ENDPROC axes
```

Da PROC axes er CLOSED, skal man IMPORTere de procedurer (konstantanter mv.), som skal benyttes, jfr. linie 320. Den centrale del af et simpelt graftegningsprogram kan herefter være

```
0080 window(x0,x1,y0,y1) // sætter tegnerammen
0090 frame(x0,x1,y0,y1) // tegner rammen
0100 axes(x0,x1,y0,y1) // tegner akserne uden inddeling
0110
0120 y:=f(x0) // kalder funktionen
0130 moveto(x0,y) // flytter til grafens start
0140 FOR x:=x0 TO x1 STEP xstep DO
0150   y:=f(x) // funktionen kaldes igen
0160   drawto(x,y) // tegner til næste punkt af grafen
0170 ENDFOR x
```

En kørsel af programmet med den i (1.1) definerede funktion giver følgende graf



Figur 1: Graf for funktionen $f: x \rightarrow e^x - x^3 - 1$

Ved udtegnning af denne figur er dog benyttet en anden procedure til aksetegning, som også medtager akseinddeling. Dette vender vi tilbage til i afsnit 1.6, hvor der bliver præsenteret et graftegningsprogram, som tillige laver en total nulpunktsbestemmelse for funktionen.

Øvelse 1: Kald programmet "graf" og tegn - efter rettelse i FUNC f - en graf for hver af funktionerne

$$f:x \rightarrow x^3 - 4x^2 - 2x + 4$$

$$f:x \rightarrow 3x^2 - e^x - 3$$

$$f:x \rightarrow x^3 + 1 - 2x$$

1.2 Algoritmerne bisektion og regula falsi.

Betragter man figur 1, vil funktionen skifte fortegn ved passage af et nulpunkt. Dette vil gælde generelt for kontinuerte funktioner, undtagen hvis funktionen også har ekstremum i dette nulpunkt. De to algoritmer, der skal beskrives i dette afsnit, tager udgangspunkt i et sådant fortegnsskift. Algoritmen bisektion (to-delning) startes med to gæt x_0 og x_1 på nulpunktets værdi, hvor $f(x_0) \cdot f(x_1) < 0$ (dvs f har modsat fortegn i x_0 og i x_1). Dernæst bestemmes midtpunktet mellem x_0 og x_1 , dvs

$$x = (x_0 + x_1) / 2$$

(dvs intervallet $[x_0, x_1]$ deles i to lige store intervaller af x). Hvis x ligger på samme side af nulpunktet som x_0 - dvs. hvis $f(x) \cdot f(x_0) > 0$ - benyttes x som nyt x_0 - dvs $x_0 = x$ - ellers er x nyt x_1 - dvs $x_1 = x$.

Dette kan direkte oversættes til et program som

```
0120  x:=(x0+x1)/2
0130  y:=f(x)
0150  IF y*y0>0 THEN
0160    x0:=x; y0:=y
0170  ELSE
0180    x1:=x; y1:=y
0190  ENDIF
```

Ved gentagen anvendelse af denne algoritme får man en følge af intervaller $D_0, D_1, \dots, D_n, \dots$, hvor $D_0 = [x_0, x_1]$. Disse interval-opfylder, at

$$D_0 \supset D_1 \supset D_2 \supset \dots \supset D_n \supset \dots$$

Dette er grafisk illustreret i figur 2 (side 4). Benævner vi længden af intervallet D_n med d_n , kan vi af den måde, konstruktionen er sket, slutte, at

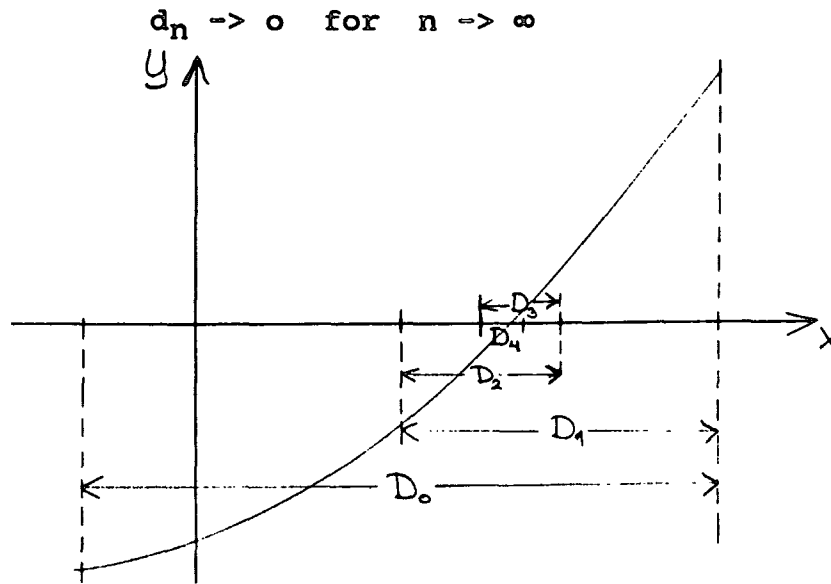
$$(1.2) \quad d_n = 1/2 d_{n-1}$$

dvs at intervallængden halveres ved hver iteration. Således er

$$d_n = (1/2)^n d_0$$

og derfor vil

(1.3)



Figur 2: De første fem intervaller i bisektion algoritmen

uanset hvordan punkterne x_0 og x_1 er valgt. Der vil således findes et fortætningspunkt x^* for intervalfølgen. Dette punkt er netop et nulpunkt for funktionen. Betragter vi følgen (x_n^+) af "positive" intervalendepunkter (dvs $f(x_n^+) > 0$) vil

$$x_n^+ \rightarrow x^* \text{ for } n \rightarrow \infty$$

(da $|x_n^+ - x^*| < d_n$), og hvis f er kontinuert på D_0 vil tillige

$$(1.4) \quad f(x_n^+) \rightarrow f(x^*) = 0 \text{ for } n \rightarrow \infty .$$

Tilsvarende kan naturligvis vises for følgen (x_n^-) af "negative" intervalendepunkter.

Ud fra (1.3) og (1.4) kan man fastlægge nogle kriterier for, hvornår et dataprogram skal stoppe beregningerne. Er den valgte nøjagtighed eps , kan man, da (1.3) gælder, benytte $d_n < \text{eps}$, og da (1.4) gælder, benytte $|f(x_n)| < \text{eps}$. Endelig kan man vælge at stoppe, når begge uligheder er opfyldt. Det sidste stopkriterium er valgt i programmet "bisek", hvor algoritmedelen har fået udseendet

```

0110 REPEAT
0120   x:=(x0+x1)/2 //          algoritmen gentages
0130   y:=f(x)
0140   n:=n+1
0150   IF y*y0>0 THEN
0160     d(n):=ABS(x-x0); x0:=x; y0:=y
0170   ELSE
0180     d(n):=ABS(x-x1); x1:=x; y1:=y
0190   ENDIF
0200 UNTIL ABS(y)<eps AND d(n)<eps // indtil f(x) er mindre end eps fra 0
0210 //          eller fejlen på x er mindre end eps

```

Eksempel: Som man så i (1.3), vil bisektions algoritmen altid konvergere, men langsomt. Af (1.2) så man, at intervallængden halveres ved hver iteration, og da

$$(1/2)^3 > 1/10 > (1/2)^4$$

vil der mellem 3. og 4. iteration være sket en intervalforkortning

med en faktor 10. Starter man med $d_0 \approx 1$, og ønskes fx $\text{eps} = 10^{-12}$, skal der benyttes ca. 40 iterationer - da

$$3 \cdot 12 < 40 < 4 \cdot 12$$

- for at opnå den ønskede nøjagtighed på nulpunktet x^* . Ønskes tilsvarende $\text{eps} = 10^{-16}$, skal man benytte ca. 50 iterationer.

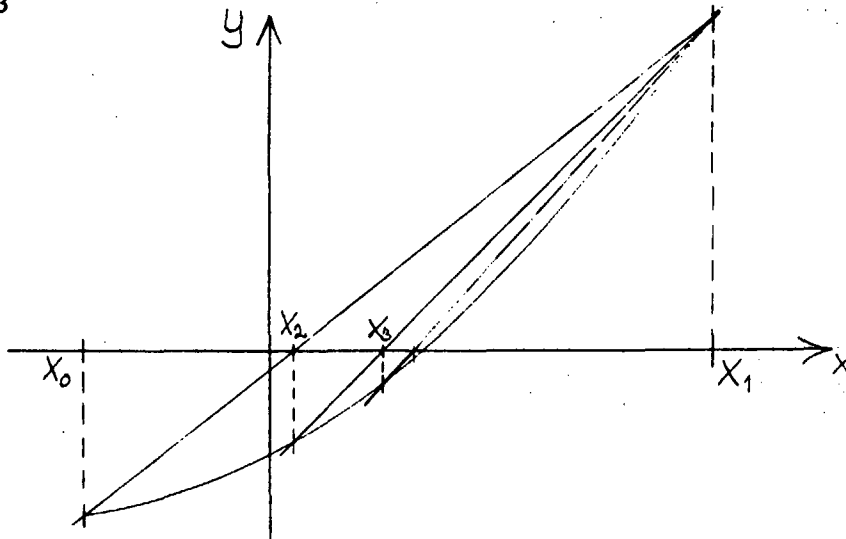
— o —

Konvergensens af bisektions algoritmen er langsom (jfr. eksemplet), og man kan være interesseret i en bedre (hurtigere) algoritme. En sådan algoritme er regula falsi (falske rette linier).

Igen starter man med to gæt x_0 og x_1 på nulpunktets værdi, hvor $f(x_0)f(x_1) < 0$. Dernæst "tegnes" en sekant til funktionen f 's graf gennem punkterne $(x_0, f(x_0))$ og $(x_1, f(x_1))$. Ligningen for denne sekant er

$$y - f(x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} (x - x_1)$$

Da funktionsværdierne i x_0 og x_1 har modsat fortegn, vil sekanten skære x -aksen ($y=0$) imellem x_0 og x_1 . Situationen fremgår af figur 3



Figur 3: De første fem punkter i regula falsi algoritmen

Skæringspunktets værdi findes ved at løse ligningen med hensyn til x (med $y=0$). Da er

$$(1.5) \quad x = \frac{x_0 f(x_1) - x_1 f(x_0)}{f(x_1) - f(x_0)}$$

Derpå gør man som i bisektion, er $f(x)f(x_0) > 0$, sættes $x_0 = x$, ellers sættes $x_1 = x$, og man gentager regula falsi algoritmen. Oversat til COMAL giver dette følgende programlinier:

```
0120  x:=(x0*y1-x1*y0)/(y1-y0)
0130  y:=f(x)
0150  IF y*y0>0 THEN
0160    x0:=x; y0:=y
0170  ELSE
0180    x1:=x; y1:=y
0190  ENDIF
```

Ved gentagen anvendelse af algoritmen får man således en følge af

x -værdier $x_0, x_1, \dots, x_n, \dots$, hvor $x_0 = x_0$ og $x_1 = x_1$, jfr fig.3. Ligger punkterne som i figur 3, kan man med brug af (1.5) få, at

$$\begin{aligned} |x_2 - x_0| &= \left| \frac{x_1 f(x_0) - x_0 f(x_1)}{f(x_0) - f(x_1)} - x_0 \right| \\ &= \left| \frac{x_1 f(x_0) - x_0 f(x_1) - x_0 f(x_0) + x_0 f(x_1)}{f(x_0) - f(x_1)} \right| \\ &= \left| \frac{f(x_0)}{f(x_0) - f(x_1)} \right| |x_1 - x_0| \\ &= \left| \frac{1}{1 - f(x_1)/f(x_0)} \right| |x_1 - x_0| < 1/2 |x_1 - x_0| \end{aligned}$$

da $f(x_1)$ og $f(x_0)$ har modsat fortegn, og da $|f(x_1)| > |f(x_0)|$

På analog måde (men mere kompliceret) kan man for

$$d_n = |x_{n+1} - x_n|$$

viser, at i "næsten" hver iteration vil gælde *)

$$d_n < 1/2 d_{n-1}$$

Man får således i det væsentlige, at

$$d_n < (1/2)^n d_0$$

og derfor vil

$$(1.6) \quad d_n \rightarrow 0 \text{ for } n \rightarrow \infty$$

uanset hvordan punkterne x_0 og x_1 er valgt. Der findes således et nulpunkt x^* således, at

$$x_n \rightarrow x^* \text{ for } n \rightarrow \infty$$

og hvis f er kontinuert på intervallet $[x_0, x_1]$, vil tillige

$$(1.7) \quad f(x_n) \rightarrow 0 \text{ for } n \rightarrow \infty$$

*) Hvis funktionen f er valgt ubehageligt, kan man risikere, at størrelsen

$$-f(x_n)/f(x_0) \ll 1.$$

Størrelsen er dog altid positiv, da de to funktionsværdier vil have modsat fortegn. Det vil derfor altid gælde, at

$$d_n < d_{n-1}$$

men dette kan medføre en meget langsom konvergens, og tit også langsommere end bisektions algoritmen. Se fx opgave 5 og 6.

På baggrund af (1.6) og (1.7) kan man helt parallelt til, hvad der blev gjort med bisektion, vælge at stoppe et dataprogram på en af betingelserne $d_n < \text{eps}$ og $|f(x_n)| < \text{eps}$ eller på dem begge. Algoritmedelen af programmet "falsi" er herefter

```

0110 REPEAT
0120   x:=(x0*y1-x1*y0)/(y1-y0) //   algoritmen gentages
0130   y:=f(x)
0140   n:=n+1
0150   IF y*y0>0 THEN
0160     d(n):=ABS(x-x0); x0:=x; y0:=y
0170   ELSE
0180     d(n):=ABS(x-x1); x1:=x; y1:=y
0190   ENDIF
0200 UNTIL ABS(y)<eps AND d(n)<eps // indtil f(x) er mindre end eps fra 0
0210 //           eller fejlen på x er mindre end eps

```

I afsnit 1.4 vil vi vende tilbage til antal iteration og til regnehastighed af de to her omtalte algoritmer.

Øvelse 2: Der er givet funktion $f: x \rightarrow x^3 - 5$ samt punkterne $x_0 = 1$ og $x_1 = 2$. Beregn (eventuelt ved hjælp af en lommeregner) ved brug af såvel bisektion som regula falsi de fem første iterationer til bestemmelse af et nulpunkt for f . Sammenlign disse resultater med $\sqrt[3]{5} \approx 1.709976$

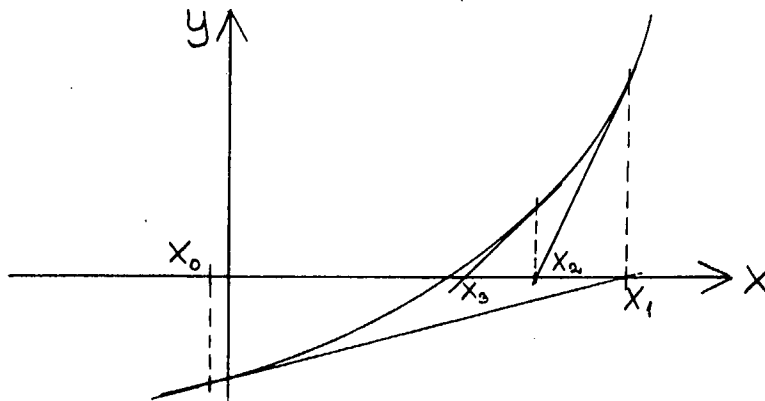
Øvelse 3: Benyt programmerne "bisek" og "falsi" til at bestemme nulpunkter for funktionen $f: x \rightarrow e^x - x^3 - 1$. Sammenlign for hvert sæt af startværdier $(x_0, x_1) = (4, 5); (1, 2); (-1, -0.5)$ konvergensens mod et nulpunkt. Vælg fx $\text{eps} = 10^{-12}$

1.3 Newton-Raphson algoritmen.

I forrige afsnit blev det vist, hvordan en sekant til funktionens graf kunne benyttes til at generere en algoritme til nulpunktsbestemmelse. En anden nærliggende angrebsvinkel er at benytte en tangent til funktionens graf.

I newton-raphson algoritmen starter man med ét gæt x_0 på nulpunktets værdi. Ligningen for tangenten til funktionen f i punktet $(x_0, f(x_0))$ er

$$y - f(x_0) = f'(x_0)(x - x_0)$$



Figur 4: De første fire punkter i newton-raphson algoritmen

Man bestemmer derpå tangentens skæring med x-aksen ($y=0$), som bliver

$$(1.8) \quad x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

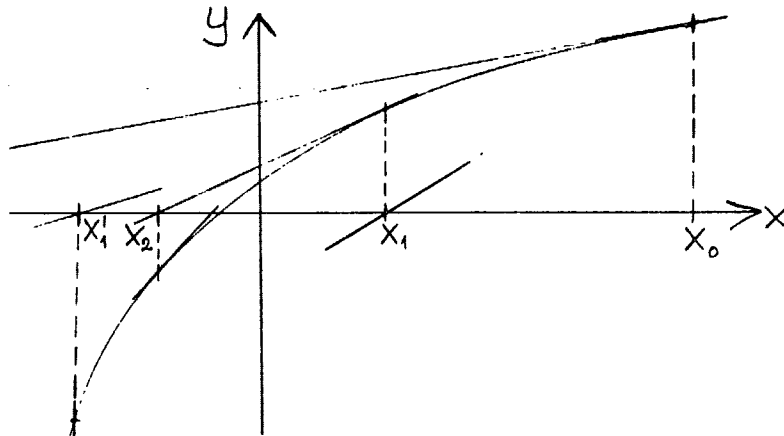
Denne x-værdi kan så benyttes som nyt gæt (dvs $x_0 = x$), og man gentager algoritmen, jfr figur 4.

Som i forrige afsnit kan man gentage algoritmen indtil $|f(x_0)| < \text{eps}$, og oversat til et COMAL-program bliver algoritmen, som følger

```
0110 y0:=f(x0); y1:=-f'(x0)
0120 REPEAT
0130   delta:=-y0/y1
0140   x1:=x0+delta
0150   x0:=x1
0160   y0:=f(x0); y1:=-f'(x0)
0170 UNTIL ABS(y0)<eps
```

I afsnit 1.2 blev beskrevet to algoritmer, som altid konvergerede - men langsomt - mod et nulpunkt, hvis man blot startede med to værdier, hvor funktionen havde modsat fortegn. Som man kan se, er newton-raphson algoritmen af en anden type, bl.a. starter vi blot med en værdi. Det viser sig ydermere, at denne algoritme ikke altid vil konvergere. I afsnit 1.4 skal vi dog vise, at hvis den konvergerer, vil den konvergere meget hurtigt.

I figur 5 kan man se en situation, hvor newton-raphson ikke sikrer, at tangenternes skæringspunkter konvergerer mod et nulpunkt.



Figur 5: Skitse af forbedret newton-raphson

En årsag er, at tangenten i det valgte punkt har en hældningskoefficient tæt på 0 (I (1.8) divideres med $f'(x_0)$).

I afsnit 1.2 viste vi, at såfremt

$$x_n \rightarrow x^* \text{ for } n \rightarrow \infty$$

hvor x^* var et nulpunkt for funktionen f , da ville

$$f(x_n) \rightarrow 0 \text{ for } n \rightarrow \infty$$

hvis f var kontinuert på et afsluttet interval, som indeholdt følgens elementer. Omvendt kan vi altså slutte, at hvis ikke

$$f(x_n) \rightarrow 0 \text{ for } n \rightarrow \infty$$

vil ej heller $x_n \rightarrow x^* \text{ for } n \rightarrow \infty$.

Således er konvergensten af $(f(x_n))$ en nødvendig betingelse for konvergensten af (x_n) . Dette kan benyttes ved en forbedring af newton-raphson.

I hver iteration i newton-raphson har man, at

$$(1.9) \quad x = x_0 - p \frac{f(x_0)}{f'(x_0)}$$

Med $p=1$. Hvis $|f(x)| > |f(x_0)|$, (dvs $(f(x_n))$ ikke konvergerer på dette trin) sættes $p = 1/2$ (dvs skridtet halveres - jfr figur 5), og $f(x)$ udregnes igen. Er fortsat $|f(x)| > |f(x_0)|$, sættes $p = 1/4$, og (1.9) beregnes, hvorefter uligheden igen undersøges. Denne halvering af skridtet fortsætter indtil

$$|f(x)| < |f(x_0)| .$$

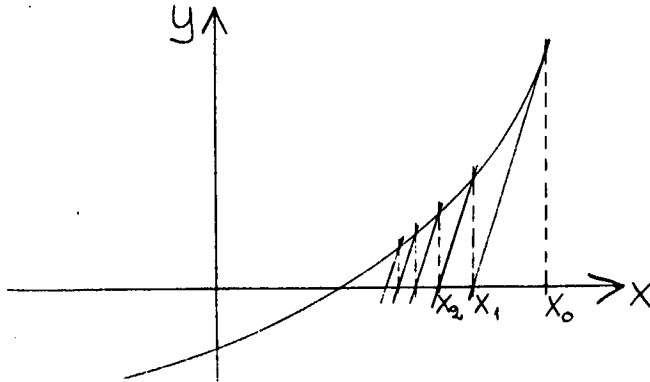
I programmet "newton" er dette lagt ind som en delalgoritme (linie 140-180) i selve newton-raphson algoritmen.

```

0110 y0:=f(x0); n:=0
0120 REPEAT // newton-raphson algoritmen gentages
0130 delta:=-y0/y1; ny:=1; n:=n+1
0140 REPEAT // skridtet halveres
0150 x2:=x0+ny*delta
0160 ny:=ny/2
0170 y2:=f(x2)
0180 UNTIL ABS(y2)<ABS(y0) // indtil algoritmen konvergerer
0190 d(n):=ABS(x2-x0)
0200 x0:=x2; y0:=y2
0210 y':=f'(x0)
0220 IF ABS(y')>100*eps THEN y1:=y'
0230 UNTIL ABS(y0)<eps // indtil f(x) er mindre end eps fra 0

```

I linie 220 er tillige suppleret med en enkelt forbedring. Hvis den numeriske værdi af tangentens hældningskoefficient er for



Figur 6: Newton-raphson med fast tangenthældning

lille, benyttes den forrige værdi af hældningskoefficienten. Af figur 6 kan man se, at algoritmen fortsat vil konvergere - blot lidt langsommere. Den her beskrevne metode kan også benyttes generelt, især hvis det er tidskrævende at beregne værdien af f' i hver iteration.

Øvelse 4: En funktion f er givet ved

$$f(x) = x^{n-a}, \quad n, a \in \mathbb{R} .$$

Vis ved indsættelse i (1.8), at newton-raphson algoritmen svarende til f bliver

$$x = ((n-1)x_0 + a/x_0^{n-1})/n$$

Øvelse 5: Der er givet funktionen $f: x \rightarrow x^3 - 5$ samt punktet $x_0 = 2$.

Beregn ved brug af newton-raphson de tre første iterationer til bestemmelse af et nulpunkt for f . Sammenlign disse værdier med de tilsvarende fra øvelse 2.

Øvelse 6: Foretag de samme beregninger som i øvelse 5, men benyt $f'(2)$ i alle tre iterationer. Hvordan påvirker dette de beregnede talværdier?

Øvelse 7: Benyt programmet "newton" til at bestemme nulpunkter for funktionen $f: x \rightarrow e^x - x^3 - 1$. Sammenlign for hver af startværdierne $x_0 = 5; 1; -1$ med de tilsvarende resultater fra øvelse 3.

1.4 Konvergens og regnehastighed.

I afsnit 1.2 blev det vist, at såvel bisektion som regula falsi algoritmen har global konvergens. Beviserne byggede på, at

$$\begin{aligned} \text{og} \quad d_n &= 1/2 d_{n-1} && \text{(bisektion)} \\ d_n &= k d_{n-1} && \text{(regula falsi)} \end{aligned}$$

hvor $k < 1/2$ (som regel). Begge algoritmer siges at have lineær konvergens, da d_n afhænger lineært af d_{n-1} .

Såfremt der gælder, at $d_n = C d_{n-1}^2$ siges konvergens at være kvadratisk. I afsnit 1.3 blev der vist situationer, hvor newton-raphson algoritmen ikke konvergerede. Man kan imidlertid vise følgende sætning:

Hvis newton-raphson algoritmen konvergerer, vil konvergens være kvadratisk. Vi definerer newton-raphson algoritmen som en ny funktion T givet ved

$$T(x) = x - \frac{f(x)}{f'(x)}$$

Starter vi med en værdi x_0 bliver der ved $x_{n+1} = T(x_n)$, $n = 0, 1, \dots$, genereret en talfølge (x_n) . Vi antager nu, at $x_n \rightarrow x^*$ for $n \rightarrow \infty$, hvor $f(x^*) = 0$.

I en omegn af x^* udvikler vi T i en taylorrække af anden orden (jfr appendix). Vi skal da udregne

$$T'(x) = 1 - \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} = \frac{f(x)f''(x)}{(f'(x))^2}$$

og

$$T''(x) = - \frac{[2f'(x)f''(x) - f'(x)f'''(x) - f(x)f^{(4)}(x)](f'(x))^2}{(f'(x))^4}$$

$$\begin{aligned}
 & + \frac{2f'(x)f''(x)[(f'(x))^2 - f(x)f''(x)]}{(f'(x))^4} \\
 = & \frac{f''(x)}{f'(x)} + f(x) \frac{f'(x)f^3(x) - 2(f''(x))^2}{(f'(x))^3}
 \end{aligned}$$

Da $f(x^*) = 0$, bliver $T(x^*) = x^*$ samt $T'(x^*) = 0$ og $T''(x^*) = f''(x^*)/f'(x^*)$. I en omegn af x^* bliver da

$$(1.10) \quad T(x) = x^* + 1/2 \frac{f''(x^*)}{f'(x^*)} (x-x^*)^2$$

Da $x_{n+1} = T(x_n)$, og sættes $d_n = |x_n - x^*|$ kan (1.10) omskrives til

$$(1.11) \quad d_{n+1} = 1/2 M^* d_n^2; \quad M^* = f''(x^*)/f'(x^*)$$

Er blot $d_n < 2/M^*$ vil $d_{n+1} < d_n$
og

$$x_n \rightarrow x^* \quad \text{for } n \rightarrow \infty$$

Hermed har vi vist, at hvis newton-raphson konvergerer, vil konvergensens være kvadratisk.

Eksempel: Antager man, at $1/2M^* \approx 1$ og $d_n \leq 10^{-1}$, da bliver $d_{n+1} \leq 10^{-2}$, $d_{n+2} \leq 10^{-4}$, $d_{n+3} \leq 10^{-8}$ og $d_{n+4} \leq 10^{-16}$. I eksemplet på side 4 så man, at bisektionsalgoritmen først når den samme nøjagtighed ved d_{n+50} .

Øvelse 8: Udledningerne i sætningen forudsætter, at $f'(x^*) \neq 0$. Man kan imidlertid vise, at konvergensens er lineær, såfremt $f'(x^*) = 0$. Gennemfør dette bevis; det anbefales at taylor-udvikle f' "passende" i en omegn af x^* . Opgave 5 og 6 illustrerer denne situation.

o

I praktiske anvendelser er man interesseret i, hvor lang regnetiden er for de benyttede algoritmer, og sjældent i hvor mange iterationer, der benyttes. Af det forrige er det klart, at bisektion og regula falsi skal benytte mange, men simple beregninger for at finde et nulpunkt. Omvendt skal newton-raphson kun benytte få, men mere komplicerede beregninger. Disse to modgående hensyn vil påvirke, hvor hurtigt en algoritme vil finde et nulpunkt for en given funktion.

Til analyse af de tre algoritmers regnehastighed er udviklet programmet "nulvalg". Programmet udskriver algoritmens totale tidsforbrug samt antal iterationer.

Anvendt på funktionen $f: x \rightarrow e^x - x^3 - 1$ med $x_0 = 4$ og $x_1 = 5$ får man

følgende resultater:

Med bisektion er
 tidsforbrug 0.7 sekunder
 nulpunktet er 4.56703683694597 med eps= 1.0E-13
 efter 46 iterationer

Med regula falsi er
 tidsforbrug 0.6 sekunder
 nulpunktet er 4.56703683694597 med eps= 1.0E-13
 efter 32 iterationer

Med newton-raphson er
 tidsforbrug 0.2 sekunder
 nulpunktet er 4.56703683694597 med eps= 1.0E-13
 efter 5 iterationer

Konklusionen her er, at newton-raphson algoritmen er langt den hurtigste, hvilket også vil kunne bekræftes med andre valg af funktioner. Som før nævnt kan man imidlertid ikke være sikker på, at newton-raphson vil konvergere. Vil man derfor lægge en sikker strategi til hurtig nulpunktsbestemmelse, skal man sammenkæde regula falsi og newton-raphson. Først bringer regula falsi følgens elementer i nærheden af nulpunktet - fx med $\text{eps} = 1/100$, og derfra kan newton-raphson overtage beregningerne. Programmet "dobbelt" kombinerer netop disse to algoritmer, og en kørsel af programmet med $f:x \rightarrow e^x - x^3 - 1$ samt $(x_0, x_1) = (4, 5)$ giver følgende resultat:

Med både regula falsi og newton-raphson er
 tidsforbrug 0.2 sekunder
 nulpunktet er 4.56703683694597 med eps= 1.0E-13
 efter 12 iterationer

Dette viser, at den konstruerede hybrid algoritme er væsentlig hurtigere end regula falsi, men måske en anelse langsommere end newton-raphson (nogle kørsler benytter 0.3 sek. og andre som her 0.2 sek.) Til gengæld er konvergensens af newton-raphson sikker, når man starter "tæt nok" på nulpunktet.

Øvelse 9: Undersøg med programmet "nulvalg" for hver af funktionerne i øvelse 1, hvor lang regnetiden bliver med de tre algoritmer.

Øvelse 10: Undersøg med programmet "dobbelt" de samme forhold som i øvelse 9.

1.5 Regula falsi defineret ved FUNC

Når man skal benytte en nulpunktsalgoritme i et større program, er det af hensyn til programmets struktur en fordel at definere algoritmen som en procedure. Da man med nulpunktsalgoritmerne ønsker at få returneret et tal - nulpunktets værdi - vil det være nærliggende at definere proceduren ved FUNC. Ønsker man en procedure for regula falsi, tager man algoritmedelen af programmet "falsi", jfr. afsnit 1.2, og bygger denne ind i en procedure således

```
0310 FUNC falsi(z1,z2) CLOSED
0320   IMPORT eps,f
0330   v1:=f(z1); v2:=f(z2)
0340   REPEAT
0350     z:=(z1*v2-z2*v1)/(v2-v1)
0360     v:=f(z)
0370     IF v*v1>0 THEN
0380       z1:=z; v1:=v
```



```

0380      z1:=z; v1:=v
0390      ELSE
0400      z2:=z; v2:=v
0410      ENDIF
0420      UNTIL ABS(v)<eps
0430      RETURN z
0440 ENDFUNC falsi

```

Her er linie 340-420 en tro kopi af algoritmedelen af "falsi". Linie 310 er procedurans hoved, hvor det specificeres, at FUNC falsi kaldes med to variable - nemlig de to gæt x_0 og x_1 . Da proceduren er CLOSED, skal man i linie 320 IMPORTere nøjagtigheden ϵ og funktionen f . Endelig RETURNeres i linie 430 nulpunktets værdi til hovedprogrammet.

Øvelse 11: Skriv en procedure (FUNC newton) for newton-raphson algoritmen og skriv dernæst en programstump, som styrer, at man starter med regula falsi og fortsætter med newton raphson analogt med programmet "dobbelt".

1.6 Total nulpunktsbestemmelse for en funktion.

Man kan nu genoptage graftegningsprogrammet "graf" fra afsnit 1.1. Dette skal udbygges med en beregning af værdien af de nulpunkter, som frekommer på den udtegnede graf. Det udbyggede program "nul- punkt" indledes med en let udvidet version af "graf" således

```

0040 USE graphics //          kalder grafikpakken
0050 REPEAT
0060   indlæsning1
0070   splitscreen
0080   graphicscreen(0) //      skærmen sættes og slettes
0090   window(x0,x1,y0,y1) //  sætter tegnerammen
0100   frame(x0,x1,y0,y1) //   tegner rammen
0110   axes(0,0,x0,x1,y0,y1,1) // tegner akserne med inddeling
0120
0130   y:=f(x0) //              funktionen kaldes
0140   moveto(x0,y) //          der flyttes til grafens start
0150   FOR x:=x0 TO x1 STEP h DO
0160     y:=f(x) //              funktionen kaldes igen
0170     drawto(x,y) //         der tegnes til næste punkt af grafen
0180   ENDFOR x
0190   INPUT "skal grafen printes (1=ja , 0=nej)?: " svar
0200   IF svar THEN printscreen
0210   INPUT "ønskes nyt plot (0=ja , 1=nej)?: " svar
0220 UNTIL svar

```

Udvidelsen består blot i, at man kan gentage graftegningen, indtil man er tilfreds med tegnerammen $[x_0,x_1] \times [y_0,y_1]$. Når man så fx har valgt intervallet $[x_0,x_1]$ så stort, at man er sikker på, at alle nulpunkter er med på den tegnede graf, fortsættes med beregning af nulpunkternes værdi.

Forud for nulpunktsbestemmelsen kan man vælge at oprette en tabel (matrix) over funktionen. Denne matrix skal først DIMensioneres. Dernæst tabelleres funktionen ved

```

0280 FOR i:=1 TO n DO
0290   x:=x0+(i-1)*h
0300   z0(1,i):=x //          der laves tabel
0310   z0(2,i):=f(x) //      over funktionen
0320 ENDFOR i

```

Programmet gennemløber så tabellen, og hver gang to på hinanden følgende funktionsværdier har modsat fortegn, kaldes FUNC falsi til bestemmelse af det nulpunkt, som ligger mellem de to tilhørende x-værdier. Dette sker i følgende programsekvens

```

0340 FOR i:=1 TO n-1 DO
0350   IF z0(2,i)*z0(2,i+1)<=0 THEN // hvis funktionen har modsat fortegn
0360     z:=falsi(z0(1,i),z0(1,i+1)) // i nabopunkter kaldes regula falsi
0370     PRINT "der er nulpunkt for funktionen i ";
0380     PRINT USING "-##.#####": z
0390   ENDIF
0400 ENDFOR i

```

Herefter har man en total nulpunktsbestemmelse - undtagen hvis et nulpunkt også er ekstremum. Udføres programmet "nulpunkt" med funktionen $f:x \rightarrow e^x - x^3 - 1$, får man dels grafen i figur 1 (jfr. side 2) og dels følgende udskrift

```

der er nulpunkt for funktionen i   -0.825155
der er nulpunkt for funktionen i    0.000000
der er nulpunkt for funktionen i    0.000000
der er nulpunkt for funktionen i    1.545007
der er nulpunkt for funktionen i    4.567037

```

I programmet kunne man naturligvis supplere med FUNC newton, jfr. øvelse 11. Herved bliver linie 360 erstattet af

```

0360     z:=newton(falsi(z0(1,i),z0(1,i+1)))

```

Denne ændring vil dog ikke gøre programmet meget hurtigere, da oprettelsen af tabellen er langt mere tidskrævende end beregningerne. Man kunne have accelereret programmet, hvis man oprettede tabellen samtidig med graftegningen. Skulle dette være gjort, kunne man imidlertid ikke gentage graftegningen, da matricen (tabellen) så skulle redimensioneres, hvilket ikke er tilladt i COMAL.

Øvelse 12: Udvid programmet "nulpunkt", således at det også kan bestemme ekstrema for en funktion. Bemærk, at f' vil skifte fortegn ved passage af et ekstremum for f . Man kan således benytte fx FUNC falsi igen på f' , når man skal finde et ekstremum. Man skal blot udvide FUNC falsi, så programmet kan styre, om der arbejdes med f eller f' . Endelig skal matricen $z0$ udvides.

Øvelse 13: Udvid programmet "nulpunkt", således at det også kan bestemme lodrette tangenter (med fortegnsskrift). Dette kan enten gøres ved at bestemme nulpunkter for $1/f$ eller gøres, hvis man fx i FUNC falsi undersøger, om $|f(x_n)| < |f(x_{n+1})|$.

1.7 Graf og nulpunkter for $f(x,y) = 0$.

I dette afsnit skal vi arbejde med funktioner af to reelle variable og forsøge at bestemme den plane kurve, som beskrives ved $f(x,y) = 0$. Til dette formål kan man benytte nogle af de algoritmer, som er omtalt.

Betragter man fx udtrykket

$$(y^3+x)^3 + 2(y-x) = 0$$

kan man vise, at der til hvert x vil svare netop én y -værdi, hvilket betyder, at udtrykket definerer y som funktion af x . Forsøger man at løse ligningen med hensyn til y , vil dette ikke lykkes. Man siger derfor, at y er implicit (indirekte) givet som funktion af x ved udtrykket.

Et andet eksempel af denne type er

$$x^2 + y^2 - r^2 = 0,$$

der som bekendt beskriver en cirkel med centrum i $(0,0)$ og radius r . Af dette udtryk kan man opskrive y eksplicit (direkte) som funktion af x - endda på to måder

$$y = \sqrt{r^2 - x^2} \text{ eller } y = -\sqrt{r^2 - x^2}.$$

Disse to funktioner beskriver øvre henholdsvis nedre halvcirkel. Der hvor de to halvcirkler mødes, vil disse to funktioner og det oprindelige udtryk have lodrette tangenter.

Dette eksempel viser, at man ikke umiddelbart af $f(x,y) = 0$ kan slutte, at der defineres en entydig (og implicit) given funktion $y(x)$. Man kan imidlertid vise, at blot der ikke optræder lodrette tangenter (eller dobbeltpunkter eller singulære punkter) for funktionen f , vil der være defineret en entydig funktion $y(x)$. Disse betingelser svarer til, at der for hvert fast x skal gælde, at

$$\frac{df}{dy} > 0 \quad (\text{eller } \frac{df}{dy} < 0)$$

dvs at f skal være monotont voksende for alle y (eller monotont aftagende for alle y). Er funktionen f givet ved

$$f(x,y) = (y^3 + x)^3 + 2(y - x)$$

bliver for hvert fast x

$$\frac{df}{dy} = 9y^2(y^3 + x)^2 + 2 > 0$$

for alle y .

Ser man herefter på funktionen g givet ved

$$g(x,y) = x^2 + y^2 - r^2$$

gælder for hvert fast x

$$\frac{dg}{dy} = 2y$$

Dette udtryk er 0 for $y=0$, dvs netop i de punkter, hvor de to halvcirkler mødes, og der er lodret tangent. Videre er g kun monotont voksende for $y > 0$, men monotont aftagende for $y < 0$.

Skal man forsøge at udvikle et dataprogram, som udtegner en graf for en implicit given funktion, skal man jo være sikker på en

entydig bestemmelse af det "næste" punkt på graf. Her er det af afgørende betydning, at df/dy har konstant fortegn.

For hvert fast x starter man med to gæt på det tilhørende y -værdi y_b og y_s , således at f_x

$$f(x, y_b) < 0 \text{ og } f(x, y_s) > 0$$

Da nu df/dy har samme fortegn for alle y , vil der mellem y_b og y_s findes netop ét y så $f(x, y) = 0$.

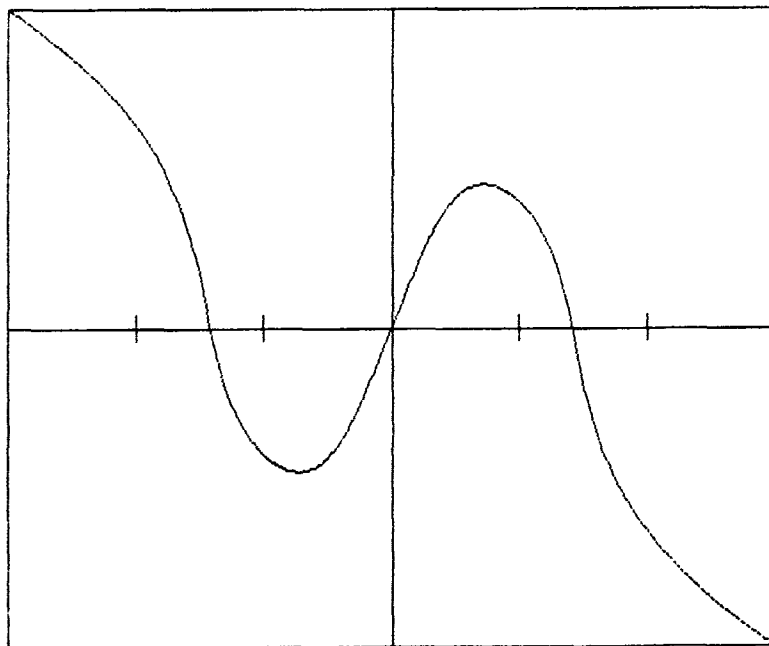
Dette lægger klart op til, at man for hvert fast x kan benytte f_x en regula falsi algoritme til at bestemme denne y -værdi. Den benyttede COMAL-procedure vil med få ændringer svare til den, der blev opstillet i afsnit 1.5.

En ændring er, at FUNC falsi kaldes med tre variable, da man også specificerer den aktuelle x -værdi. Endvidere er de indgående funktioner af to variable. Herefter kan man danne tegnedelen af programmet "falsipl"

```

0080 splitscreen
0090 window(x0,x1,y0,y1) // sætter tegnerammen
0100 frame(x0,x1,y0,y1) // tegner rammen
0110 axes(0,0,x0,x1,y0,y1,1) // akserne tegnes med akseinddeling
0120
0130 FOR i:=1 TO n DO
0140   x:=x0+(i-1)*h // x varierer i intervallet [x0,x1] med step h
0150   y:=falsi(x,yb,ys)
0160   yb:=y-2*h; ys:=y+2*h // nye startpunkter for y i x+h
0170   IF x=x0 THEN
0180     moveto(x,y) // pen flyttes til grafens start
0190   ELSE
0200     drawto(x,y) // der tegnes til næste punkt på grafen
0210   ENDIF
0220   d(1,i):=x
0230   d(2,i):=y
0240 ENDFOR i

```

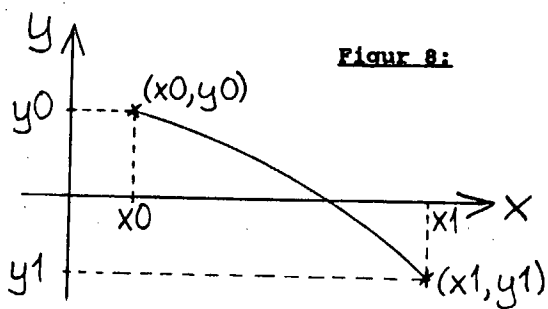


Figur 7: Graf for den ved $(y^3+x)^3+2(y-x) = 0$ givne funktion

De første linier er forklaret i programmet og svarer i øvrigt til "graf" og "nulpunkt". I linie 150 kalder man for hver ny x-værdi proceduren falsi for at få returneret den tilhørende y-værdi, og i linie 160 dannes de nye gæt - yb og ys, der skal benyttes for næste x-værdi. Disse gæt bestemmes ved at benytte den fundne y-værdi og hertil at addere ± 2 gange den valgte skridtlængde h. Hvis grafens tangent i x har for stor (numerisk) hældning (≥ 2), vil programmet stoppe, og man må da vælge en større fremskrivningsfaktor, inden programmet køres igen.

I linie 220-230 laves en tabel over funktionen. Laver man en kørsel af denne del af programmet "falsipl", vil man svarende til udtrykket $(y^3+x)^3 + 2(y-x) = 0$ få en graf som vist i figur 7; tegnerammen er $[-3,3] \times [-1,1]$.

Når man herefter vil gå over til at bestemme nulpunkter for den implicit givne funktion y, må dette ske ved at interpolere mellem punkter i den oprettede tabel. To hinanden følgende punkter (x_0, y_0) og (x_1, y_1) fra tabellen kan være placeret som vist i figur 8.



Figur 8:

Man ønsker således at finde det nulpunkt for funktionen y, som ligger mellem x_0 og x_1 , dvs en x-værdi som opfylder $f(x, 0) = 0$. Er fx $y_0 > 0$ og $y_1 < 0$ som vist på figur 8, kan vi altså benytte en regula falsi algoritme til at bestemme værdien af dette nulpunkt. Man skal dog samtidigt sikre sig (via FUNC falsi), at man forbliver på den rigtige graf

$-f(x, y) = 0$. Dette giver følgende procedure, som kaldes nulpunkt.

```

0380 FUNC nulpunkt(x0,x1,y0,y1) CLOSED
0390   IMPORT falsi,eps
0400   REPEAT
0410     x:=(y1*x0-y0*x1)/(y1-y0)
0420     y:=falsi(x,y0,y1)
0430     IF y*y0>0 THEN
0440       x0:=x; y0:=y
0450     ELSE
0460       x1:=x; y1:=y
0470     ENDIF
0480   UNTIL ABS(y)<eps
0490   RETURN x
0500 ENDFUNC nulpunkt

```

I procedurens hoved specificeres, at FUNC nulpunkt kaldes med fire variable, der er koordinaterne til to nabopunkter (fx de i figur 8 viste). I linie 410 laves en regula falsi iteration til bestemmelse af en ny x-værdi, og derpå (linie 420) kaldes FUNC falsi for at bestemme den tilhørende y-værdi. Disse to trin gentages, indtil $|y| < \text{eps}$ (linie 480). Herefter er den tilhørende x-værdi nulpunkt for funktionen y.

Den del af programmet "falsipl", der styrer nulpunktsbestemmelsen, er strukturelt opbygget som den tilsvarende del af "nulpunkt". Her gennemløbes tabellen, og hver gang to hinanden følgende y-værdier har modsat fortegn, kaldes FUNC nulpunkt, der returnerer nulpunktets værdi, og til slut udskrives resultatet.

```

0300 FOR i:=1 TO n-1 DO
0310   IF d(2,i)*d(2,i+1)<=0 THEN
0320     x:=nulpunkt(d(1,i),d(1,i+1),d(2,i),d(2,i+1))
0330     PRINT "der er nulpunkt for funktionen i ";
0340     PRINT USING "-##.#####": x
0350   ENDIF
0360 ENDFOR i

```

En kørsel af programmet "falsipl" med udtrykket

$$(y^3+x)^3 + 2(y-x) = 0$$

tegner først grafen i figur 7, og dernæst kommer følgende udskrift

```

der er nulpunkt for funktionen i   -1.414213
der er nulpunkt for funktionen i   -0.000000
der er nulpunkt for funktionen i    1.414213

```

Ved indsættelse i funktionen f givet ved

$$f(x,y) = (y^3+x)^3 + 2(y-x)$$

af $x = \sqrt{2}$, $x = 0$ eller $x = -\sqrt{2}$ viser, at disse tre x -værdier opfylder $f(x,0) = 0$, hvilket svarer til udskriften fra programkørslen.

Øvelse 14: Vis, at hver af følgende udtryk

$$2x+y+(y-x^2)^3+4 = 0$$

$$e^{x+y}+(y-x^2)^3 = 0$$

$$3y-x/2+\sin(5x+y) = 0$$

definerer en entydig implicit given funktion y (dvs at der for hvert fast x vil gælde $df/dy > 0$ for alle y).

Øvelse 15: Benyt programmet "falsipl" til at tegne graf og bestemme nulpunkter for hver af de implicit givne funktioner i øvelse 14.

Øvelse 16: Som man så i eksemplet med cirkelligningen $x^2+y^2-r^2=0$, kan man af fortegnsskift for df/dy få en opdeling i flere implicit givne funktioner. Har man ved udregning af df/dy styr på en sådan opdeling, kan man ved at ændre i "falsipl" benytte dette program til at udtegne hver af disse delfunktioner. Lav en ændring i "falsipl", således at man kan tegne flere (implicit givne) funktioner på samme figur. Tegn dernæst de ved $f(x,y) = 0$ implicit (eller eksplicit) givne funktioner, når

$$f_1:(x,y) \rightarrow x^2+y^2-1$$

$$f_2:(x,y) \rightarrow x^2-y^2-x^4$$

$$f_3:(x,y) \rightarrow y^2-x^3$$

Øvelse 17: Det generelle problem, som antydes i øvelse 16, består i, at man skal bestemme de punkter (x,y) , hvor både

$$f(x,y) = 0 \quad \text{og} \quad \frac{df(x,y)}{dy} = 0$$

Prøv at lave et program, der tager højde for dette problem.

Det er fortsat naturligt at starte med at tegne x_0 og fortsætte med at tegne $(x,y(x))$ indtil $|df/dy| < \text{eps}$. Derefter er man nødt til at skifte over til at tegne $(x(y),y)$. Dette er naturligvis muligt, hvis $df/dx > 0$ (eller $df/dx < 0$) for hvert fast y og alle x . ($df/dx = 0$, bl.a. når $f(x,y) = 0$ har vandret tangent). Når man skal tegne $(x(y),y)$, må man også have en regula falsi procedure, som finder et (x,y) på $f(x,y) = 0$ ud fra to gæt x_b og x_s . Herefter kan man

naturligt fortsætte med at tegne $(x(y), y)$ indtil $|df/dx| < \epsilon$, hvorefter man må returnere til $(x, y(x))$. Man er formentlig også nødt til at starte i x_1 og tegne "baglæns". Eventuelt skal man også prøve at starte med at tegne $(x(y), y)$ fra y_0 eller y_1 .

Øvelse 18: Afprøv det i øvelse 17 udarbejdede COMAL-program på hver af udtrykkene

$$\begin{aligned}xy &= \sin(x+y) \\x^3+y^3+xy+x+y &= 0\end{aligned}$$

2. ALGORITMER TIL NUMERISK LØSNING AF DIFFERENTIALLIGNINGER

I dette kapitel skal vi behandle en metode til at løse differentiaalligninger eller systemer af koblede differentiaalligninger numerisk. Arbejdet vil blive koncentreret om en type af algoritmer, som benævnes runge-kutta algoritmer. I den klassiske form, som bliver behandlet i denne fremstilling, er algoritmerne udviklet i perioden 1895-1901.

Gennemgangen af teorien for runge-kutta algoritmer sker udelukkende for én autonom differentiaalligning af første orden

$$y' = f(y)$$

At ligningen er autonom, betyder, at funktionen f kun afhænger af systemets egen tilstand y , men ikke af ydre forhold.

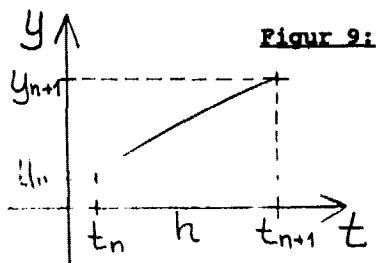
Der vil senere blive redegjort for, at teorien har gyldighed også for ikke-autonome differentiaalligninger og for systemer af koblede differentiaalligninger.

2.1 Taylorudvikling af en løsning til den autonome differentiaalligning.

Når man skal løse en differentiaalligning numerisk, og man starter i et punkt (t_0, y_0) , er det nødvendigt, at man entydigt kan fortsætte løsningen fra dette punkt. Ellers er datamaskinen for "dum" til at vide, hvordan den kommer videre. Med den autonome differentiaalligning

$$(2.1) \quad y' = f(y)$$

kan man vise, at der gennem punktet (t_0, y_0) går en entydig løsning til differentiaalligningen, hvis f er differentiabel. I det følgende får man brug for at differentiere f flere gange, når man skal taylorudvikle denne løsningsfunktion, så i realiteten skal man forudsætte f flere gange differentiabel. Da man i praktiske anvendelser som regel har vilkårligt ofte differentiable funktioner, vil antagelsen om, at f blot skal være en gang differentiabel, ikke betyde nogen reel indskrænkning.



Man betragter den autonome differentiaalligning (2.1) med start i punktet (t_0, y_0) . Der findes da en entydigt fastlagt løsningsfunktion $y(t)$, som opfylder $y(t_0) = y_0$. Denne funktion kan man approksimativt bestemme ved at foretage en taylorudvikling (jfr appendix A) af funktionen i en

række punkter. Vi betragter to t-værdier t_n og t_{n+1} og sætter $Y_n = y(t_n)$, $Y_n^{(i)} = y^{(i)}(t_n)$ samt $h = t_{n+1} - t_n$. Da bliver en Taylorudvikling af funktionen y indtil q -te orden i punktet t_n , jfr. figur 9.

$$Y_{n+1} = Y_n + Y_n' h + \frac{1}{2} Y_n'' h^2 + \dots + \frac{1}{q!} Y_n^{(q)} h^q + O(h^{q+1})$$

Da imidlertid $y' = f(y)$ bliver

$$(2.2) \quad y'' = f'(y)y' = f'(y)f(y)$$

$$\text{og} \quad y''' = f''(y)y'f(y) + f'(y)f'(y)y'$$

$$(2.3) \quad = f''(y)(f(y))^2 + (f'(y))^2 f(y)$$

osv

Eksempel: For den logistiske ligning

$$y' = ay(b-y)$$

vil man således få, at

$$y'' = (ab-2ay)(aby-ay^2)$$

$$= a^2b^2y - 3a^2by^2 + 2a^2y^3$$

og videre

$$y''' = -2a(aby-ay^2)^2 + (ab-2ay)^2(aby-ay^2)$$

$$= a^3b^3y - 7a^3b^2y^2 + 12a^3by^3 - 6a^3y^4$$

I almindelighed vil det dog være temmelig tid- og pladskrævende i hvert skridt h at beregne et passende antal afledede af f . Man vælger derfor en anden tilgangsvinkel, som vil blive præsenteret i næste afsnit. Idéen er, at man vil bestemme Y_{n+1} ud fra et gennemsnit af værdier af f beregnet i et antal punkter i intervallet $[Y_n, Y_{n+1}]$.

Øvelse 1: Beregn en løsning til den logistiske ligning

$$y' = \frac{1}{10} y(50-y)$$

gennem punktet $(0,1)$ og med skridtet $h = 0.1$. Benyt en Taylorudvikling af 2.orden og beregn frem til $t = 0.3$.

2.2 Runge-kutta algoritmer.

Idéen i disse algoritmer vil blive illustreret med en 3.ordens Runge-kutta algoritme, og det vil blive vist, hvordan man fastlægger værdien af de parametre, der indgår i algoritmen. Man tager udgangspunkt i en Taylor-udvikling af løsningsfunktionen $y(t)$ indtil 3.orden, dvs

$$Y_{n+1} - Y_n = Y_n' h + \frac{1}{2} Y_n'' h^2 + \frac{1}{6} Y_n''' h^3 + O(h^4)$$

Da løsningsfunktionen tilfredsstillers den autonome ligning (2.1), får man af (2.2) og (2.3) i afsnit 2.1, at

$$(2.4) \quad Y_{n+1} - Y_n = hf(Y_n) + 1/2h^2f'(Y_n)f(Y_n) \\ + 1/6h^3[f''(Y_n)(f(Y_n))^2 + (f'(Y_n))^2f(Y_n)] + O(h^4)$$

I 3.ordens runge-kutta vil man erstatte (2.4) med udtrykket

$$(2.5) \quad Y_{n+1} - Y_n = h(w_1k_1 + w_2k_2 + w_3k_3)$$

$$\text{hvor} \quad k_1 = f(Y_n), \quad k_2 = f(Y_n + p_1hk_1)$$

$$k_3 = f(Y_n + (p_2 - p_3)hk_2 + p_3hk_3).$$

Man skal herefter bestemme parametrene w_1, w_2, w_3 og p_1, p_2, p_3 , således at (2.4) og (2.5) stemmer overens (til og med h^3). Man skal bemærke, at beregningen k -erne sker succesivt, dvs først beregnes k_1 , dernæst benyttes den ved beregning af k_2 og endelig benyttes de begge ved beregningen af k_3 . Denne struktur kan fortsætte, hvis man skal beregne flere k -værdier, dvs bestemme runge-kutta af orden større end 3. Valget af parameteren $p_2 - p_3$ kan virke besynderlig, men de senere omregninger af k_3 vil retfærdiggøre dette valg.

Når man skal sammenligne (2.4) med (2.5), må udtrykkene for k_2 og k_3 omformuleres og taylorudvikles efter potenser af h . For k_2 får man

$$k_2 = f[Y_n + p_1hf(Y_n)] \\ = f(Y_n) + (p_1hf(Y_n))f'(Y_n) + 1/2(p_1hf(Y_n))^2f''(Y_n) + O(h^3)$$

Dette kan indsættes i k_3 , og man får

$$k_3 = f[Y_n + (p_2 - p_3)hf(Y_n) + p_3hf(Y_n + p_1hf(Y_n))] \\ = f[Y_n + (p_2 - p_3)hf(Y_n) + p_3hf(Y_n) + p_3p_1h^2f(Y_n)f'(Y_n) + O(h^3)] \\ = f(Y_n) + [p_2hf(Y_n) + p_1p_3h^2f(Y_n)f'(Y_n)]f'(Y_n) \\ + 1/2(p_2hf(Y_n))^2f''(Y_n) + O(h^3) \\ = f(Y_n) + p_2hf'(Y_n)f(Y_n) + 1/2p_2^2h^2f''(Y_n)(f(Y_n))^2 \\ + p_1p_3h^2(f'(Y_n))^2f(Y_n) + O(h^3)$$

Til slut skal udtrykkene for k_2 og k_3 indsættes i (2.5), der herved bliver

$$(2.6) \quad Y_{n+1} - Y_n = h(w_1 + w_2 + w_3)f(Y_n) \\ + h^2(p_1w_2 + p_2w_3)f'(Y_n)f(Y_n) \\ + 1/2h^3(p_1^2w_2 + p_2^2w_3)f''(Y_n)(f(Y_n))^2$$

$$+ h^3 p_1 p_3 w_3 (f'(Y_n))^2 f(Y_n) + O(h^4)$$

Vi kan nu sammenholde (2.4) og (2.6), og en sammenligning led for led giver, at

$$\begin{aligned} \text{for } hf(Y_n) \text{ er} & & w_1 + w_2 + w_3 & = 1 \\ \text{for } h^2 f'(Y_n) f(Y_n) \text{ er} & & p_1 w_2 + p_2 w_3 & = 1/2 \\ \text{for } 1/2 h^3 f''(Y_n) (f(Y_n))^2 \text{ er} & & p_1^2 w_2 + p_2^2 w_3 & = 1/3 \\ \text{for } h^3 (f'(Y_n))^2 f(Y_n) \text{ er} & & p_1 p_3 w_3 & = 1/6 \end{aligned}$$

Herved får man ialt fire ligninger til bestemmelse af de seks parametre, og der vil derfor være uendelig mange mulige løsningssæt til ligningssystemet. I litteraturen om runge-kutta algoritmer omtales stort set kun to sæt af parametre, som løser ligningerne. Disse to valg giver

Heun's metode (1900):

$$Y_{n+1} - Y_n = h(1/4 k_1 + 3/4 k_3)$$

hvor

$$k_1 = f(Y_n), \quad k_2 = f(Y_n + 1/2 h k_1)$$

$$k_3 = f(Y_n + 2/3 h k_2) \quad [\text{dvs } p_2 - p_3 = 0]$$

Kutta's metode (1901):

$$(2.7) \quad Y_{n+1} - Y_n = h(1/6 k_1 + 2/3 k_2 + 1/6 k_3)$$

hvor

$$k_1 = f(Y_n), \quad k_2 = f(Y_n + 1/2 h k_1)$$

$$k_3 = f(Y_n - h k_1 + 2 h k_2)$$

Øvelse 2: Benyt kutta's metode til at beregne det første punkt på den løsningskurve til den logistiske ligning

$$y' = \frac{1}{10} y(50 - y)$$

som starter i (0,1). Benyt $h = 0.1$

Øvelse 3: Opstil selv udtrykket for 2.ordens runge-kutta algoritmen og vis, at man efter tilsvarende (men enklere) beregninger for, at

$$w_1 + w_2 = 1, \quad p_1 w_2 = 1/2$$

dvs, at man fx kan vælge w_2 som fri parameter. For $w_2 = 1/2$ får man den algoritme, der ofte benævnes forbedret Euler metode.

Øvelse 4: Benyt en 2.ordens runge-kutta algoritme (fx med $w_2 =$

1/2) til at beregne den løsning til den logistiske ligning

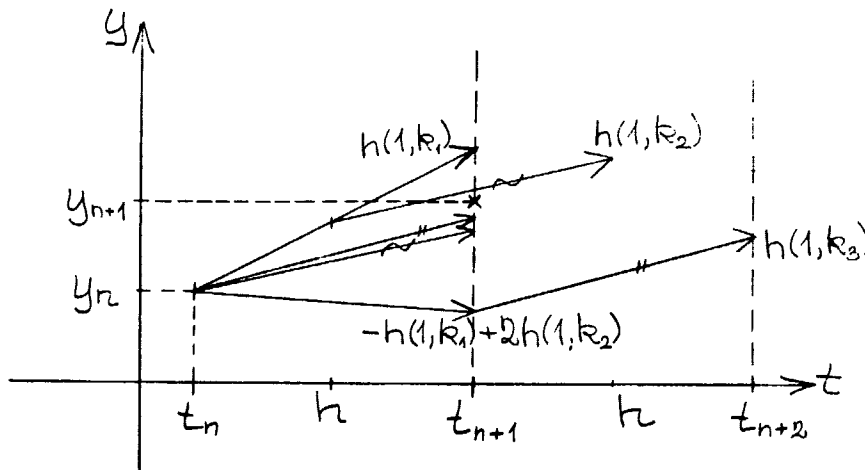
$$y' = \frac{1}{10}y(50-y)$$

som starter i $(0,1)$. Benyt $h = 0.1$ og regn frem til $t = 0.3$.

Skal man geometrisk beskrive, hvad der foregår i en runge-kutta algoritme, kan man som eksempel benytte kutta's metode (2.7). Da man arbejder med den autonome differentialligning

$$y' = f(y)$$

vil hver af k -værdierne, k_1, k_2 og k_3 beskrive hældningskoefficienter for en tangent til kurven $y(t)$, se figur 10. Derfor vil $(1, k_1)$ være en retningsvektor for tangenten med hældningskoefficient k_1 .



Figur 10: Retningsvektorerne i 3.ordens runge-kutta

~ er parallel med $(1, k_2)$
 // er parallel med $(1, k_3)$

Når man skal beregne k_2 , bevæger man sig fra punktet (t_n, y_n) i retningsvektorens retning til punktet

$$(2.8) \quad (t_n, y_n) + 1/2h(1, k_1)$$

og her udregnes $k_2 = f(y_n + 1/2hk_1)$ som hældningskoefficient for tangenten til kurven i punktet (2.8). Herefter tages en linearkombination af de to fundne retningsvektorer

$$-h(1, k_1) + 2h(1, k_2) = h(1, -k_1 + 2k_2)$$

og i punktet

$$(t_n, y_n) + h(1, -k_1 + 2k_2)$$

beregnes endnu en tangenthældning

$$k_3 = f(y_n - hk_1 + 2hk_2).$$

På baggrund af de tre retningsvektorer $(1, k_1)$, $(1, k_2)$ og $(1, k_3)$, kan man, jfr. (2.7), finde det næste punkt på løsningskurven til

$$\begin{aligned} (t_{n+1}, Y_{n+1}) &= (t_n, Y_n) + h(1, k_1)/6 + 2h(1, k_2)/3 + h(1, k_3)/6 \\ &= (t_n + h, Y_n + h(k_1/6 + 2k_2/3 + k_3/6)) \end{aligned}$$

I hvert runge-kutta trin foretager man således tre evalueringer (beregninger) af funktionen f , og det næste punkt fremkommer ved at fremskrive fra det aktuelle punkt med et vægtet gennemsnit af de tre evalueringer.

2.3 Almindelig 4.ordens runge-kutta og dens implementering som procedure.

Når man skal opbygge et dataprogram, der benytter en runge-kutta algoritme til numerisk løsning af en differentiaalligning, er det som oftest en 4.ordens algoritme, der anvendes. Grunden er - som man skal se i afsnit 2.5 - at en 4.ordens algoritme er langt hurtigere end både 3.- og 2.ordens algoritmer (hvis man tager hensyn til den relative fejl). Vi skal derfor i korte træk udvikle en 4.ordens runge-kutta algoritme. Man skal her erstatte en 4.ordens taylorudvikling af løsningsfunktionen med udtrykket

$$(2.9) \quad Y_{n+1} - Y_n = h(w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4)$$

hvor

$$\begin{aligned} k_1 &= f(Y_n) \quad , \quad k_2 = f(Y_n + p_1 h k_1) \\ k_3 &= f(Y_n + (p_2 - p_3) h k_1 + p_3 h k_2) \\ k_4 &= f(Y_n + (p_4 - p_5 - p_6) h k_1 + p_5 h k_2 + p_6 h k_3) \end{aligned}$$

Taylorudvikler man nu - som i afsnit 2.2 - k -erne efter potenser af h , og indsætter i (2.9), kan ved sammenligning med den 4.ordens taylorudvikling fås ialt otte ligninger

$$\begin{aligned} w_1 + w_2 + w_3 + w_4 &= 1 \quad , \quad w_3 p_1 p_3 + w_4 (p_1 p_5 + p_2 p_6) &= 1/6 \\ w_2 p_1 + w_3 p_2 + w_4 p_4 &= 1/2 \quad , \quad w_3 p_1 p_2 p_3 + w_4 p_4 (p_1 p_5 + p_2 p_6) &= 1/8 \\ w_2 p_1^2 + w_3 p_2^2 + w_4 p_4^2 &= 1/3 \quad , \quad w_3 p_1^2 p_3 + w_4 (p_1^2 p_5 + p_2^2 p_6) &= 1/12 \\ w_2 p_1^3 + w_3 p_2^3 + w_4 p_4^3 &= 1/4 \quad , \quad w_4 p_1 p_3 p_6 &= 1/24 \end{aligned}$$

Allerede i 1901 viste Kutta [5], side 121-122, at følgende 1-parameterfamilie (for $t > 0$) var løsning til ligningssystemet (prøv selv!)

w_1	w_2	w_3	w_4	p_1	p_2	p_3	p_4	p_5	p_6
$1/6$	$\frac{2-t}{3}$	$t/3$	$1/6$	$1/2$	$1/2$	$\frac{1}{2t}$	1	$1-t$	t

Da der er otte ligninger til at bestemme de ti parametre, må der imidlertid også findes løsninger, der ikke tilhører denne 1-parameterfamilie, fx fandt Kutta [3], side 22 også følgende sæt af løsninger:

Øvelse 6: Prøv selv at lave en procedure (PROC rk2) for forbedret Euler metode og en procedure (PROC rk3) for Kutta's metode af 3.ordens runge-kutta algoritmen.

2.4 Fejlvurdering og skridtlængde.

Når man som i afsnit 2.3 vil benytte en 4.ordens runge-kutta algoritme, vil man begå en fejl i hvert iterationsskridt. Denne fejl vil man forsøge at minimere. I [5] er givet et udtryk for fejlfunktionen for en 4.ordens runge-kutta algoritme. Man kan imidlertid få check på fejlen uden at beregne fejlfunktionen, hvilket vil blive belyst i det følgende. Men først et eksempel.

Eksempel: Her betragtes den meget simple differentiaalligning

$$y' = ay$$

hvor $a \in \mathbb{R}$. Ifølge [5], side 76 vil fejlen ved at benytte en q -te ordens runge-kutta algoritme på denne differentiaalligning i punktet (t_n, Y_n) blive

$$C_n h^{q+1} = -\frac{1}{(q+1)!} a^{q+1} Y_n h^{q+1} .$$

For en 4.ordens runge-kutta betyder det, at

$$C_n = -\frac{1}{120} a^5 Y_n .$$

Vi kan altså slutte, at C_n er uafhængig af h , og at C_n kun varierer langsomt med n , dvs $C_{n-1} \approx C_n$.

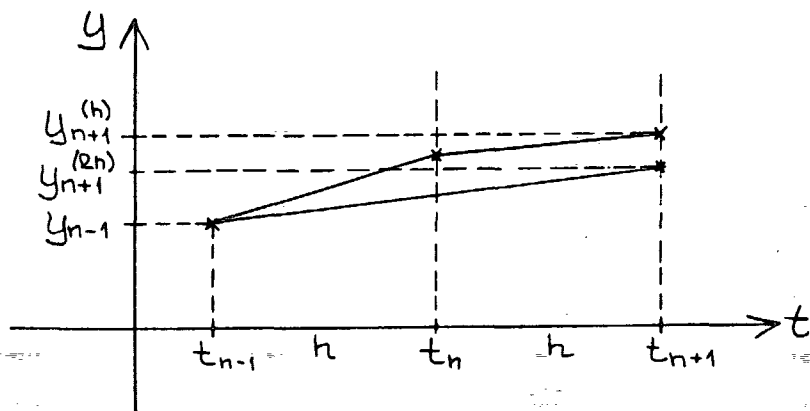
Betragter man igen den autonome differentiaalligning

$$(2.1) \quad y' = f(y)$$

kan man for små h ($h \ll 1$) med god tilnærmelse antage, at den differentiable funktion f er stykkevis lineær - dvs f er fx lineær i intervallet $[Y_{n-1}, Y_{n+1}]$. Således kan vi overføre de to konklusioner fra eksemplet også til fejlløddet for den q -te ordens runge-kutta algoritme, når den anvendes på den autonome differentiaalligning (2.1).

Idéen i fejlvurderingen er, at man sammenligner to fremskrivninger med en q -te ordens runge-kutta algoritme, jfr figur 11. Lad os antage, at den "korrekte" værdi for løsningsfunktionen $y(t)$ i punktet t_{n+1} er Y_{n+1} . Først foretages to fremskrivninger fra t_{n-1} til t_{n+1} med skridtlængde h , hvorved man får værdien $Y_{n+1}(h)$, og

$$(2.10) \quad Y_{n+1} - Y_{n+1}(h) = 2C_n h^{q+1}$$



Figur 11: De to typer fremskrivning fra t_{n-1} til t_{n+1}

da vi antog, at $C_{n-1} \approx C_n$.

Dernæst foretages en fremskrivning fra t_{n-1} til t_{n+1} med skridtlængde $2h$, hvorved man får værdien $y_{n+1}^{(2h)}$, og

$$(2.11) \quad y_{n+1}^{(2h)} - y_{n+1}^{(h)} = C_n (2h)^{q+1} = 2^{q+1} C_n h^{q+1}$$

Nu indgår $C_n h^{q+1}$ i begge ligninger, dvs (2.10) og (2.11), og ved at isolere dem af begge ligninger bliver

$$\frac{1}{2} (y_{n+1}^{(h)} - y_{n+1}^{(2h)}) = 2^{-q-1} (y_{n+1}^{(h)} - y_{n+1}^{(2h)})$$

eller

$$2^q (y_{n+1}^{(h)} - y_{n+1}^{(2h)}) = y_{n+1}^{(h)} - y_{n+1}^{(2h)}$$

eller

$$(2^q - 1) (y_{n+1}^{(h)} - y_{n+1}^{(2h)}) = y_{n+1}^{(h)} - y_{n+1}^{(2h)}$$

dvs

$$(2.12) \quad y_{n+1}^{(h)} - y_{n+1}^{(2h)} = \frac{y_{n+1}^{(h)} - y_{n+1}^{(2h)}}{2^q - 1}$$

Her er højre side i (2.12) kendte størrelser, og man kan altså benytte den som teststørrelse

$$(2.13) \quad E_q = \frac{y_{n+1}^{(h)} - y_{n+1}^{(2h)}}{2^q - 1}$$

Er således $E_q < \epsilon$ - den valgte nøjagtighed - begår vi en fejl

mindre end eps ved at benytte $y_{n+1}(h)$ i stedet for den "korrekte" værdi y_{n+1} . Hvis teststørrelsen $E_q > \text{eps}$, må man vælge en mindre h -værdi.

Et nærliggende bud på en ny h -værdi, hvis teststørrelsen (2.13) ikke godkendes, vil være at halvere h . Man kan imidlertid vælge mere rationelt. Man kan ifølge [7], side 71

$$(2.14) \quad h_{ny} = k \cdot h_{g1} \cdot (\text{eps}/E_q)^{1/q+1}$$

Når $E_q > \text{eps}$ vil $\text{eps}/E_q < 1$, og derfor er

$$(\text{eps}/E_q)^{1/q+1} < 1$$

og jo større E_q er end eps , jo mindre bliver h_{ny} i forhold til h_{g1} . Tallet k er en sikkerhedsfaktor, som skal sikre så få gennemløb af fejlberegningen ovenfor som mulig, dvs k skal vælges således, at teststørrelsen (2.13) vil blive godkendt ved næste gennemløb. I litteraturen, fx [7], side 71, foreslås $k = 0.8$, men man kan faktisk speede et program op ved at vælge k tættere på tallet 1 (herom senere). I programmer, som er knyttet til denne bog, er k valgt i intervallet $[0.90, 0.98]$.

Udtrykket (2.14) kan også anvendes, når teststørrelsen (2.13) er godkendt, dvs når $E_q < \text{eps}$. Da bliver

$$(\text{eps}/E_q)^{1/q+1} > 1$$

og man får således en større værdi af h_{ny} end af h_{g1} . Igen vil sikkerhedsfaktoren k sikre, at man ikke vælger h_{ny} så stor, at næste gennemløb af fejlberegningen måske ikke godkendes.

Disse teoretiske overvejelser kan direkte oversættes til en COMAL-procedure

```

0250 PROC rk(REF x, REF h) CLOSED //  proceduren checker fejlen i
0260   IMPORT rk4, eps //           den valgte rk algoritme
0270   x1:=x; x2:=x
0280   rk4(x1,h); rk4(x1,h) //      kalder rk algoritmen to gange med h
0290   k:=2*h; rk4(x2,k) //        kalder rk algoritmen en gang med 2h
0300   e:=(x2-x1)*(x2-x1)
0310   p:=15*eps/SQR(e)
0320   h:=0.95*h*p^0.2 //          den nye værdi af h returneres (REF)
0330   IF p>1 THEN //             hvis fejlen er lille nok
0340     x:=x1 //                  returneres (REF) 2. kald af rk med step h
0350   ELSE //                     ellers gentages
0360     rk(x,h) //                proceduren med den nye værdi af h
0370   ENDIF
0380 ENDPROC rk

```

Bemærk, at PROCEDUREHovedet præciserer, at såvel x som h er defineret som REFERENCE variable, dvs at der ved kald af denne procedure returneres nye værdier for dem begge. I linie 280 kaldes den 4.ordens runge-kutta procedure (rk4) fra afsnit 2.3 to gange med skridt h , jfr. (2.11) og i næste linie kaldes den igen én gang, men med skridt $2h$, jfr. (2.12).

I linie 310 opskrives teststørrelsen

$$p = 15 * \text{eps}/e$$

hvor jo $e/15$ netop er E_4 , idet man for $q = 4$ har, at $2^q - 1 = 15$. Hvis $p > 1$ - dvs teststørrelsen (2.13) bliver godkendt returnerer proceduren den sidste værdi af x_1 som ny x -værdi (linie 340), ellers kalder proceduren sig selv (linie 360), men med den nye (og mindre) h -værdi.

Man har således en procedure, som checker, at man i hver iteration med en 4.ordens runge-kutta algoritme kun begår en fejl, som er mindre end den valgte nøjagtighed. Der mangler nu kun den programstump, der styrer kaldet af proceduren `rk`. Den programdel kan fx være

```
0070 WHILE t<t1 DO //          gentages indtil tiden er t1
0080   PRINT USING "-###.#### ": t;
0090   PRINT USING "-###.####": y
0100   rk(y,h) //          kalder rk algoritmen med fejlcheck
0110   t:=t+2*h
0120 ENDWHILE
```

I linie 80 printes værdien af t , og i linie 90 værdien af y -men udskrevet på samme linie som t (;). I linie 100 kaldes PROC `rk` med den aktuelle værdi af y og h , og de nye værdier returneres. Det hele gentages, så længe (WHILE) $t < t_1$, jfr. linie 70.

Under FUNC `f` er fastlagt den logistiske ligning

$$y' = ay(b-y).$$

En kørsel af programmet "rk4mono" med $a = 0.01$, $b = 270$ og $t_1 = 0.5$ samt med start i $(t_0, y_0) = (0, 1)$ givet følgende resultat

0.00000	1.00000	1.38131	39.34779
0.10046	1.30766	1.44014	44.99499
0.19585	1.68925	1.49922	51.27043
0.28665	2.18146	1.55882	58.22458
0.37317	2.78132	1.61929	65.90964
0.45586	3.50371	1.68100	74.37864
0.53512	4.36596	1.74438	83.68361
0.61131	5.38704	1.80991	93.87188
0.68476	6.58748	1.87811	104.97781
0.75574	7.98952	1.94932	117.00377
0.82452	9.61715	2.02336	129.87617
0.89134	11.49616	2.09888	143.35660
0.95644	13.65430	2.17340	156.93499
1.02002	16.12129	2.24485	169.90264
1.08230	18.92895	2.31274	181.14213
1.14348	22.11127	2.37760	191.70966
1.20376	25.70450	2.44001	201.07526
1.26335	29.74725		tidsforbrug 4.4 sekunder
1.32246	34.28052		

Øvelse 7: Kald programmet "rk4mono" og kørsel programmet med differentiallyingningen

$$y' = \frac{1}{10}y(50-y)$$

gennem $(0, 1)$. (Der skal rettes i PROC-indlæsning - linie 210). Sammenlign resultatet med de tilsvarende fra øvelse 2, 4 og 5.

Øvelse 8: Skriv selv en COMAL-procedure, som styrer fejlchecket på Kutta's metode, jfr. øvelse 6.

2.5 Sammenligning af en numerisk og en analytisk løsning til Bertalanffy's ligning.

Der er nu indført en procedure, der laver et fejlcheck på den valgte runge-kutta algoritme. Det skal naturligvis også godtgøres, at den beregnede løsning opfylder den pålagte nøjagtighed. Dette kan ikke vises generelt, men kun eksemplarisk, hvor det er muligt at bestemme en analytisk løsning til en given differentiaalligning. I et sådant tilfælde kan man afgøre, om afvigelsen mellem den analytiske og den numeriske løsning er inden for den valgte nøjagtighed eps.

Som eksempel vælges Bertalanffy's ligning

$$y' = ay^{2/3} - by.$$

Denne differentiaalligning er model for vægten $y(t)$ af en fisk, hvor t angiver tiden siden fiskens "fødsel". Udledningen af de analytiske løsninger findes fx i [4], side 35-36. Herfra findes

$$(2.15) \quad y(t) = (a/b + ce^{-bt/3})^3$$

hvor $c \in \mathbb{R}$. Man vil finde den blandt løsningerne, som indeholder $(0, Y_0)$. Da er

$$Y_0 = (a/b + c)^3$$

eller

$$c = Y_0^{1/3} - a/b.$$

Indsat i (2.5), bliver den analytiske løsning gennem $(0, Y_0)$

$$(2.16) \quad y(t) = (a/b + (Y_0^{1/3} - a/b)e^{-bt/3})^3$$

Programmet "fisk" sammenligner den numeriske løsning af Bertalanffy's ligning gennem $(0, Y_0)$ med (2.16). I "fisk" er (2.16) indskrevet i FUNC f1. Ved kørsel af "fisk" udskrives hver femte punkt af den numeriske løsning sammen med det punkt af (2.16), som har samme t -værdi.

En kørsel af "fisk" med $\text{eps} = 10^{-8}$, $Y_0 = 3$ og $t_1 = 5$ giver følgende resultater

0.27952406	4.65580256
	4.65580259
0.69576221	7.66496043
	7.66496050
1.22514927	12.16384907
	12.16384918
1.91236601	18.56502402
	18.56502417
2.77815834	26.64620385
	26.64620402
3.74992752	34.87522367
	34.87522385
4.74805283	41.94756053
	41.94756071

nøjagtigheden er $1.0E-8$

Heraf ses, at fejlen på den numeriske løsning vil være mindre end 1 på ottende betydende ciffer, dvs nøjagtigheden $\text{eps} = 10^{-8}$ er overholdt.

Øvelse 9: Udled den analytiske løsning til den logistiske ligning gennem $(0, y_0)$.

Øvelse 10: Kald programmet "fisk", indskriv den logistiske ligning under PROC f, og den analytiske løsning (øvelse 9) under FUNC f1. Foretag en kørsel af "fisk" med passende valg af a, b, y_0 , t1 og eps, og undersøg, om den valgte nøjagtighed er opfyldt.

2.6 Regnehastighed af forskellige algoritmer.

Indtil nu er de omtalte COMAL-programmer udelukkende baseret på den almindeligste 4.ordens runge-kutta algoritme, jfr. afsnit 2.3. Der er i afsnit 2.2 omtalt bl.a. Kutta's metode og forbedret Euler metode, som man ved selvprøvning har kunnet se, var kortere at programmere, og havde færre beregninger i hver iteration i forhold til 4.ordens runge-kutta. Det er derfor vigtigt at undersøge, hvilke af disse algoritmer, der vil være hurtigst, hvis man stiller krav om en bestemt relativ nøjagtighed. Der er næppe tvivl om, at fx forbedret Euler metode skal benytte en mindre h-værdi end 4.ordens runge-kutta for at sikre samme nøjagtighed. Omvendt vil forbedret Euler metode behøve langt færre beregninger per iteration.

Disse to modgående hensyn er det vanskeligt at vurdere teoretisk. Man er nødt til at sammenligne de faktiske regnetider på en bestemt opgave.

Til denne analyse er fremstillet programmet "diffvalg". Dette program giver den tid, fire algoritmer benytter til at udregne samme numeriske løsning. De fire algoritmer er mersons version af 4.ordens runge-kutta med eget fejlcheck ([3], side 24-25 og [7], side 84), almindelig 4.ordens runge-kutta, kutta's metode og forbedret Euler-metode. "diffvalg" løser differentialligningen

$$y'' + 2ay' + b^2y = m \sin(rt)$$

gennem punktet (t_0, y_0, y_0') . En kørsel af "diffvalg" med $a = 0.1$, $b = 1$, $m = 4$, $r = 1$ og $t_1 = 4$ gennem $(t_0, y_0, y_0') = (0, 5, 0)$ giver resultatet

sikkerhedsfaktorerne $k_m = 0.98$ og $k_4 = 0.9$

Med mersons algoritme er
tidsforbrug 5.5 sekunder
skridtlængde ved stoptid er 0.102268

Med 4. ordens runge-kutta er
tidsforbrug 6.4 sekunder
skridtlængde ved stoptid er 0.109387

Med 3. ordens runge-kutta er
tidsforbrug 12 sekunder
skridtlængde ved stoptid er 0.039516

Med 2. ordens runge-kutta er
tidsforbrug 33.9 sekunder
skridtlængde ved stoptid er 0.011095

Dette viser, at de to 4.ordens algoritmer er langt de hurtigste. Med et ændret valg af startpunkt så $(t_0, y_0, y_0') = (0, 5, 3)$ bliver

sikkerhedsfaktorerne $k_m = 0.98$ og $k_4 = 0.9$

Med mersons algoritme er
tidsforbrug 5.7 sekunder
skridtlængde ved stoptid er 0.101000

Med 4. ordens runge-kutta er
tidsforbrug 5.6 sekunder
skridtlængde ved stoptid er 0.126047

Med 3. ordens runge-kutta er
tidsforbrug 10.8 sekunder
skridtlængde ved stoptid er 0.044528

Med 2. ordens runge-kutta er
tidsforbrug 31.9 sekunder
skridtlængde ved stoptid er 0.011291

Her er situationen den samme, og tilmed er de to 4.ordens algoritmer lige hurtige.

På baggrund af disse kørsler er det nærliggende at konkludere, at en 4.ordens algoritme er langt hurtigere end både 3. og 2.ordens algoritmer. Ser man på skridtlængder ved sluttid i kørslerne, bekræfter de, at der i 4.ordens algoritmer foretages langt færre iterationer.

Programmet "diffvalg" kan også benyttes til at undersøge regnehastighedens afhængighed af sikkerhedsfaktorerne, jfr (2.14). I de to viste kørsler er $k_m = 0.98$ (svarende til merson) og

$k_4 = 0.90$. Gentages imidlertid den første kørsel med ændrede sikkerhedsfaktorer, får man jfr. øvelse 12 længere regnetid end den oprindelige kørsel. Årsagerne er, at man med for små sikkerhedsfaktorer får mindre h-værdier end optimalt, og at man med for store sikkerhedsfaktorer får så store h-værdier, at fejlchecket gennemløbes flere gange end rimeligt.

Øvelse 11: Kald programmet "diffvalg" og indtast i linie 1390 den logistiske ligning - her skrevet som

$$w(1) := a * z(1) * (b - z(1))$$

og i linie 1400 $w(2) := 0$

samt i linie 1280 $a := 0.1$, $b := 50$

Undersøg for $\epsilon = 10^{-8}$ regnetider for de fire algoritmer, også med forskellige valg af sikkerhedsfaktorer.

Øvelse 12: Gentag den første kørsel af "diffvalg", men med $k_m = 0.90$ og $k_4 = 0.82$ og sammenlign regnetiderne. Gør det samme med $k_m = k_4 = 0.99$.

2.7 Anden ordens differentiaalligninger og flere koblede differentiaalligninger.

Indtil nu har vi udelukkende arbejdet med autonome differentiaalligninger som

(2.1) $y' = f(y)$
 Langt fra alle differentiaalligninger er imidlertid af denne form.
 En mere generel differentiaalligning af første orden vil være på
 formen

$$(2.17) \quad y' = f(s, y)$$

hvor s kan være en vilkårlig ydre, variabel faktor fx tiden.
 Supplerer man med differentiaalligningen

$$(2.18) \quad s' = 1$$

vil (2.17) og (2.18) tilsammen beskrive et autonomt system af
 to koblede differentiaalligninger.

Indfører man vektoren

$$\vec{x} = (s, y)$$

og vektorfunktionen

$$\vec{F}(s, y) = (1, f(s, y))$$

kan dette system skrives kort som den autonome vektordifferen-
 tiaalligning

$$\vec{x}' = \vec{F}(\vec{x}).$$

Som nævnt i afsnit 2.6 løste programmet "diffvalg" differential-
 ligningen

$$(2.19) \quad y'' + 2ay' + b^2y = m\sin(rt)$$

Denne ligning af 2.orden kan ligeledes omskrives til et
 (autonomt) system af koblede differentiaalligninger af 1.orden.
 Sættes nu

$$\begin{aligned} y' &= z \\ \text{bliver} \quad y'' &= z' \end{aligned}$$

men af (2.19) kan man få, at

$$y'' = -2ay' - b^2y + m\sin(rt)$$

og indføres z er

$$z' = -2az - b^2y + m\sin(rt).$$

Da den variable t også indgår i ligningen, kan man som før
 indføre en ekstra differentiaalligning

$$t' = 1.$$

Således har man de tre koblede differentiaalligninger

$$(2.20) \quad \begin{aligned} t' &= 1, & y' &= z \\ z' &= -2az - b^2y + m\sin(rt). \end{aligned}$$

Indfører man vektoren

$$\vec{x} = (t, y, z)$$

og vektorfunktionen

$$\vec{f}(t, y, z) = (1, z, -2az - b^2y + m\sin(rt))$$

kan de tre ligninger i (2.20) skrives som den autonome vektor differentialligning

$$\vec{x}' = \vec{f}(\vec{x})$$

Øvelse 13: Omskriv hver af differentialligningerne til et autonomt system af første ordens differentialligninger

$$\begin{aligned} y'' + a(y^2 - 1)y' + y &= 0 \\ y' + ty' - y &= 0 \\ y'' + 3y'' + 3y' + y &= \sin(2t) \end{aligned}$$

I det følgende har vi den autonome vektor differentialligning

$$\vec{x}' = \vec{f}(\vec{x})$$

hvor

$$\vec{x} = (x_0, x_1, \dots, x_n)$$

og

$$\vec{f} = (1, f_1, \dots, f_n)$$

dvs differentialligningen

$$\begin{aligned} x_0' &= 1 \\ \text{samt de første ordens differentialligninger} \\ x_1' &= f_1(x_0, x_1, \dots, x_n) \\ x_2' &= f_2(x_0, x_1, \dots, x_n) \\ &\vdots \\ &\vdots \\ x_n &= f_n(x_0, x_1, \dots, x_n) \end{aligned}$$

For dette system kan man vise, at de parametre, som blev udledt i afsnit 2.2 og 2.3, uden videre kan overføres. Dette beror på, at man kan differentiere hver funktion f_1, \dots, f_n for sig med hensyn til en af de variable x_1, \dots, x_n for sig. Man vil for hver funktion for sig få de samme parametre som for den enlige ligning i afsnit 2.2 og 2.3.

o

Skal disse betragtninger overføres til et program, der løser et system af første ordens differentialligninger, skal det

eksisterende program blot modificeres lidt. Man skal erstatte en variable x med en vektorvariabel $x()$ med $n+1$ koordinater. En sådan vektor skal dimensioneres, hvilket sker med

```
1240 DIM x(0:n), x0(0:n) //      vektorer dimensioneres
```

Videre skal man ændre den procedure (FUNC f), der fastlægger differentiaalligning, idet den nu kaldes med en vektor og skal returnere en vektor. Man benytter derfor PROC, og da man skal returnere en vektor, skal vektoren i procedurehovedet være en REFERENCE variable. Ligningen i (2.19) er fx fastlagt som

```
1350 PROC f(REF z()) CLOSED //      procedure der fastlægger
1360   IMPORT m,n,a,b,r //          differentiaalligningssystemet
1370   DIM w(0:n)
1380   w(0):=1
1390   w(1):=z(2)
1400   w(2):=-2*a*z(2)-b*b*z(1)+m*SIN(r*z(0))
1410   z():=w() //                  ny vektor returneres (REF)
1420 ENDPROC f
```

Bemærk, at tilordningen i linie 1410 ikke er tilladt i alle COMAL-versioner, og den kan da erstattes af

```
FOR i=0 TO n DO z(i):=w(i)
```

Efter samme retningslinier skal man i fejlchecket (PROC rk) og i 4.ordens runge-kutta (PROC rk4) erstatte en variabel med en vektorvariabel, således at rk4 bliver

```
0340 PROC rk4(REF z(),h) CLOSED //  alm. 4. ordens rk
0350   IMPORT n,eps,f
0360   DIM z1(0:n), z2(0:n), z3(0:n), z4(0:n), y(0:n)
0370   y():=z()
0380   f(y()); z1():=y() //          1. evaluering
0390   FOR i:=0 TO n DO y(i):=z(i)+h*z1(i)/2
0400   f(y()); z2():=y() //          2. evaluering
0410   FOR i:=0 TO n DO y(i):=z(i)+h*z2(i)/2
0420   f(y()); z3():=y() //          3. evaluering
0430   FOR i:=0 TO n DO y(i):=z(i)+h*z3(i)
0440   f(y()); z4():=y() //          4. evaluering
0450   FOR i:=0 TO n DO y(i):=z(i)+h*(z1(i)+2*z2(i)+2*z3(i)+z4(i))/6
0460   z():=y() //                  ny vektor returneres (REF)
0470 ENDPROC rk4 //                  h reterneres ikke, ej REF-variabel
```

og tilsvarende rk

```
0150 PROC rk(REF z(),REF h) CLOSED // procedure der checker fejlen i
0160   IMPORT n,rk4,eps //          den valgte rk algoritme
0170   DIM x1(0:n), x2(0:n), y0(0:n)
0180   x1():=z(); x2():=z(); e:=0
0190   rk4(x1(),h); rk4(x1(),h) //   kalder rk algoritmen to gange med h
0200   k:=2*h; rk4(x2(),k) //       kalder rk algoritmen en gang med 2h
0210   FOR i:=0 TO n DO
0220     y0(i):=x2(i)-x1(i)
0230     e:=e+y0(i)*y0(i)
0240   ENDFOR i
0250   p:=15*eps/SQR(e)
0260   h:=0.94*h*p^0.2 //           ny værdi af h returneres (REF)
0270   IF p>1 THEN //              hvis fejlen er lille nok
0280     z():=x1() //              returneres 2. kald af rk med skridt h
0290   ELSE //                      ellers gentages
0300     rk(z(),h) //              rk med den nye værdi af h
0310   ENDIF
0320 ENDPROC rk
```

I rk er dog kommet en væsentlig ændring. Ved beregning af fejlestimatet E_4 (jfr. (2.13)) skal vi nu beregne længden af vektoren

$$(2.21) \quad \frac{\begin{matrix} \rightarrow (h) \\ x_{n+1} \end{matrix} - \begin{matrix} \rightarrow (2h) \\ x_{n+1} \end{matrix}}{15}$$

Fra rumgeometri ved man, at længden af

$$\vec{z} = (z_1, z_2, z_3)$$

bliver

$$\|\vec{z}\| = (z_1^2 + z_2^2 + z_3^2)^{1/2}$$

Dette udtryk kan uden videre generaliseres til en vilkårlig endelig dimension n , dvs er

$$\vec{z} = (z_1, z_2, \dots, z_n)$$

bliver længden af z

$$\|\vec{z}\| = (z_1^2 + z_2^2 + \dots + z_n^2)^{1/2}$$

I linie 210-240 udregnes differencen i (2.21) koordinatvis, hvorefter disse kvadreres, og endelig summeres de. I linie 250 beregnes til slut kvadratroden af denne sum.

Øvelse 14: Omformulér de i øvelse 6 fremstillede procedurer rk2 og rk3, således de kan anvendes på vektordifferentialligninger. Hvilke ændringer skal der laves i proceduren rk (fejlchecket), for at den kan anvendes sammen med rk3?

Øvelse 15: Opskriv proceduren f for hver af differentialligningerne i øvelse 13.

2.8 Numerisk og analytisk løsning af modellen for "dobbelt radioaktivt henfald".

Med muligheden for at løse systemer af første ordens differential- ligninger kan man tillige sammenligne sådanne løsninger med den tilsvarende analytiske løsning (hvis man kan bestemme den).

Her skal vi arbejde med de to differentialligninger

$$(2.22) \quad x' = -ax, \quad y' = ax - by$$

Disse to ligninger er en simpel model for, hvordan et radioaktivt stof (x) henfalder til et nyt stof (y), som igen er radioaktivt og derfor også henfalder (kort: dobbelt radioaktivt henfald). Der findes flere sådanne henfaldskæder. Den her valgte er omtalt i [4], side 97-98.

Det første stof er bly-214 med halveringstid $T_1 = 27$ min (= 9/20

time), og det andet stof er bismuth-214 med halveringstid $T_2 = 20$ min (= 1/3 time). Da sammenhæng mellem henfaldskonstant og halveringstid er som $aT_1 = \ln 2$ bliver

$$a = \frac{20}{9} \ln 2 \quad \text{og} \quad b = 3 \ln 2$$

I [4], side 97-98 er den analytiske løsning tillige udledt. Der startes (til $t_0 = 0$) med x_0 atomer bly-214 isoleret (og derfor intet bismuth-214, dvs $y_0 = 0$). Da bliver den analytiske løsning

$$x(t) = x_0 e^{-at}$$

$$y(t) = \frac{ax_0}{b-a} (e^{-at} - e^{-bt})$$

I programmet "radio" er ligningerne i (2.22) indskrevet i PROC f således

```
0710 PROC f(REF z()) CLOSED //      procedure der fastlægger
0720   IMPORT n,a,b //              differentiaalligningerne
0730   DIM w(0:n)
0740   w(0):=1
0750   w(1):=-a*z(1)
0760   w(2):=a*z(1)-b*z(2)
0770   z():=w()
0780 ENDPROC f
```

og den analytiske løsning i PROF f1

```
0800 PROC f1(REF z()) CLOSED //     procedure der fastlægger
0810   IMPORT a,b,n,r //             de analytiske løsninger
0820   DIM w(n)
0830   w(1):=r*EXP(-a*z(1))
0840   w(2):=a*r*(EXP(-a*z(1))-EXP(-b*z(1)))/(b-a)
0850   z():=w()
0860 ENDPROC f1
```

Programmet "radio" udskriver hvert tiende punkt af den numeriske løsning og sammenligner det med det punkt af den analytiske løsning, som har samme t-værdi (= x(0)). En kørsel af "radio" med $x_0 = 100$ og $\text{eps} = 10^{-8}$ frem til $t_1 = 5$ giver

0.16148909	77.97779964	18.57629870
	77.97779963	18.57629875
0.35520272	57.86089785	28.81079048
	57.86089784	28.81079057
0.56730785	41.73460127	31.42000190
	41.73460126	31.42000201
0.80194521	29.07597598	29.18000489
	29.07597595	29.18000501
1.06494062	19.39108317	24.20020976
	19.39108315	24.20020987
1.36492664	12.21587996	18.18081600
	12.21587993	18.18081610
1.71561109	7.11758151	12.27138547
	7.11758148	12.27138555
2.14068326	3.69809505	7.23400601
	3.69809501	7.23400607
2.68047883	1.61019635	3.51608878
	1.61019630	3.51608879
3.35467168	0.56999864	1.36165944
	0.56999858	1.36165940
4.14340697	0.16913950	0.43148750
	0.16913945	0.43148744

nøjagtigheden er 1.0E-8

I hver koordinat er afvigelsen således mindre end 1 på ottende betydende ciffer, indtil man når til de små værdier til slut. Dette skyldes, at der undervejs har oparbejdet sig meget små fejl, som på grund af tallenes størrelse får en relativ større betydning.

Øvelse 16: Bestem den analytiske løsning til differentiaalligningen

$$(2.23) \quad y'' + 0.2y' + y = 5\sin t$$

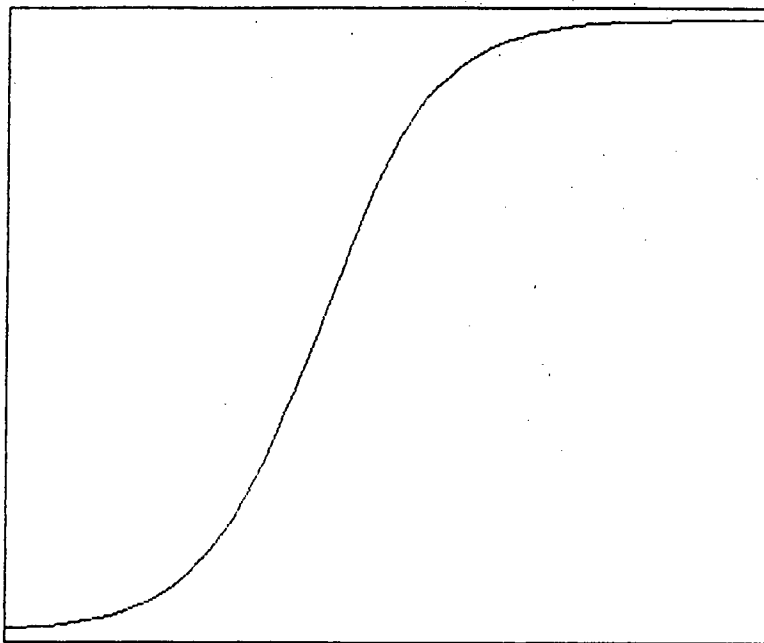
gennem punktet $(t_0, Y_0, Y_0') = (0, 0, -5)$. Se fx [4], side 114-120.

Øvelse 17: Kald programmet "radio" og indskriv (2.23) under PROC f, og den i øvelse 16 fundne analytiske løsning under PROC fl. Lav en kørsel af "radio" med start i $(t_0, Y_0, Y_0') = (0, 0, -5)$ og med $\text{eps} = 10^{-8}$.

2.9 Grafisk fremstilling af numeriske løsninger.

Dette afsnit skal afrunde fremstillingen om numerisk løsning af differentiaalligninger og vise, hvordan numeriske løsninger kan fremstilles grafisk.

I afsnit 2.4 blev en del af programmet "rk4mono" gennemgået. Dette program løste en differentiaalligning og udskrev en tabel over løsningsfunktionen. Imidlertid kan man i programmet også vælge at lave en graf for løsningen, således at man i stedet forbinder tabellens punkter med rette linier. Vælger man at tegne grafen svarende til tabellen på side 30, får man figur 12.



Figur 12: Graf for en løsning til $y' = \frac{1}{100}y(270-y)$

Skal man ligeledes i programmet "rk4" supplere med muligheden for grafisk fremstilling, opstår der flere problemer. Dels er den aktuelle tegneramme ikke kendt - dvs valget af y_{\min} og y_{\max} , dels

kan man vælge mellem flere mulige grafer. Med modellen for dobbelt radioaktivt henfald i afsnit 2.8 kan man således vælge mellem tre grafer - nemlig et tx-, et ty- eller et xy-kordinat-system. Har man ialt n differentiallyigninger af første orden (samt $x_0'=1$), kan man vælge mellem

$$n(n+1)/2 \text{ grafer}$$

(bevis det!)

I programmet "sving" er der derfor valgt den indgangsvinkel, at der først oprettes en fil, hvori man indlæser en tabel over løsningsfunktionen, jfr følgende del af "sving".

```

0070 OPEN FILE 1,fil$,WRITE //  filen åbnes for skrivning
0080 WHILE x(0)<t1 DO //      indtil tiden er t1
0090   WRITE FILE 1: x() //   skrives der på filen
0100   FOR i:=1 TO n DO
0110     IF x(i)>xmax(i) THEN xmax(i):=x(i) // finder max og min i
0120     IF x(i)<xmin(i) THEN xmin(i):=x(i) // alle n koordinater
0130   ENDFOR i
0140   rk1(x(),h)
0150 ENDWHILE
0160 xmax(0):=t1-1
0170 CLOSE FILE 1 //      filen lukkes

```

Her åbnes i linie 70 filen for skrivning (WRITE), og man skriver i filen (linie 90), indtil tiden er t1 (linie 80). Videre fastlægger man i linie 100-130 største- og mindsteværdi, som løsningsfunktionen antager i hver koordinat. Derefter lukkes filen i linie 170. Når først tabellen ligger på en fil, kan man kalde den så ofte, man ønsker, for at udtegne grafer. Dette sker i følgende programdel

```

0190 USE graphics
0200 REPEAT
0210   INPUT "Hvilke to koordinater skal plottes mod hinanden?": j,k
0220   graphicscreen(0) //   skærmen sættes og slettes
0230   splitscreen
0240   p1:=INT(xmin(j))
0250   p2:=INT(xmax(j))+1
0260   q1:=INT(xmin(k))
0270   q2:=INT(xmax(k))+1
0280   window(p1,p2,q1,q2) // tegnerammen bestemmes
0290   frame(p1,p2,q1,q2) //  rammen tegnes
0300   axes(0,0,p1,p2,q1,q2,1) // akserne tegnes gennem (0,0)
0310   OPEN FILE 2,fil$,READ // filen åbnes for læsning
0320   READ FILE 2: x0() //   der læses fra filen
0330   moveto(x0(j),x0(k)) //  pen flyttes til startpunkt
0340   WHILE NOT EOF(2) DO //  indtil "end of file"
0350     READ FILE 2: x0() //   læses næste punkt fra filen og
0360     drawto(x0(j),x0(k)) // pennen tegner til dette punkt
0370   ENDWHILE
0380   CLOSE FILE 2 //      filen lukkes
0390   INPUT "skal grafen printes (1=ja,0=nej)?: svar
0400   IF svar THEN printscreen
0410   INPUT "nyt plot (1=nej,0=ja)?: svar
0420   textscreen
0430 UNTIL svar

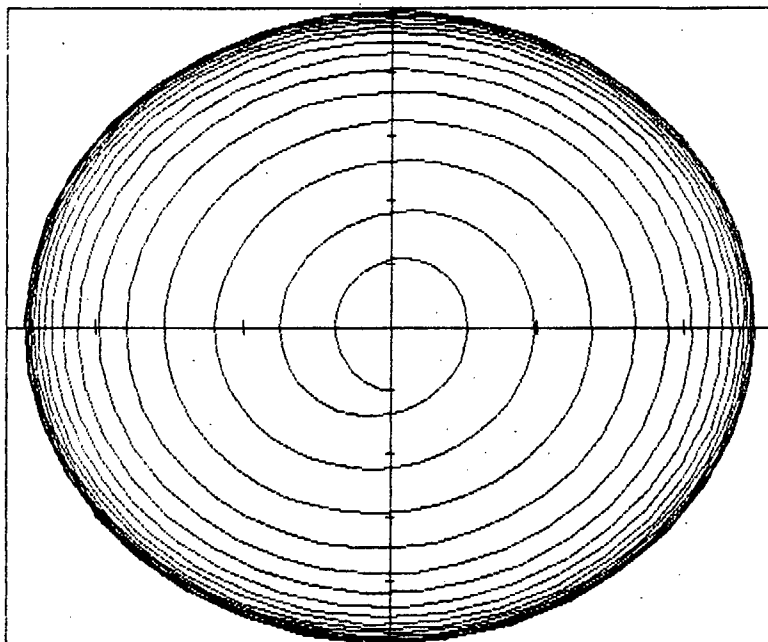
```

I linie 240-270 beregnes heltalsværdier for tegnerammen, som fastlægges i linie 280 og tegnes i linie 290. Akserne tegnes i linie 300. I linie 310 åbnes filen for læsning (READ), og man fortsætter med at læse (linie 350) og at tegne en linie (linie 360), indtil man når til enden af filen - EOF (linie 340). Herefter lukkes filen (linie 380).

En kørsel af "sving" med differentialligningen

$$y'' + 0.2y' + 4y = 5\sin 2t$$

og med start i $(t_0, y_0, y'_0) = (0, 0, -5)$ frem til $t_1 = 50$ giver figur 13, hvis man vælger et yy' -koordinatsystem.



Figur 13: Graf i $y-y'$ for en løsning til $y'' + 0.2y' + 2y = 5\sin 2t$

Øvelse 18: Da den benyttede regnenøjagtighed som regel er "større" end den nødvendige tegnenøjagtighed, kan det af pladshensyn være en fordel kun at gemme fx hvert femte punkt i tabellen. Forsøg selv at lave en sådan ændring i "sving".

Øvelse 19: For nogle differentialligninger vil løsningskurven efter kortere eller længere tid bevæge sig i en "lukket" kurve (kaldet en grænsecykel). Det kan derfor være en fordel at tegne grafen og vente med at gemme data, indtil denne indsvingning er sket. Lav de nødvendige ændringer i "sving", således at dette er muligt. Husk også, at man skal kunne fortælle programmet, når det skal begynde at gemme data.

Øvelse 20: Med de i øvelse 18 og 19 foreslåede ændringer skal man nu løse Lorenz' ligninger

$$\begin{aligned}x' &= a(y-x) \\y' &= rx - y - xz \\z' &= xy - bz\end{aligned}$$

Her er $a = 10$ og $b = 2.66667$. Foretag en kørsel af "sving" svarende til følgende r -værdier $r \in \{28, 92.5, 100, 155, 250\}$.

PS: For $r = 28$ vil der ikke opstå en grænsecykel.

2.10. Afsluttende bemærkninger.

De i dette kapitel omtalte runge-kutta algoritmer er blandt den type algoritmer til numerisk løsning af differentialligninger,

der kaldes "single step" metoder. De er karakteriseret ved, at man på baggrund af en række evalueringer foretager en frem-skrivning et trin. En særlig fordel ved runge-kutta algoritmerne er - som vi har set - at man kan variere skridtlængden ved hvert trin samt, at algoritmerne er selvstartende, blot man kender begyndelsespunktet.

Beslægtet med runge-kutta findes nogle algoritmer, som også involverer den afledede af f i beregningerne af k 'erne. Herved kan man reducere antallet af evalueringer for at opnå samme nøjagtighed som med fx 4.ordens runge-kutta. Denne type algoritmer er bl.a. velegnet til at løse "stive lignings-systemer". Sådant benævnes differentiaalligningssystemer, hvor de indgående konstanter er af forskellig størrelsesorden (fx $a \approx 10^{-5}$, $b \approx 10^3$). Denne type differentiaalligninger optræder især ved modeller for kemiske systemer. Her vil de indgående reaktioner typisk have meget forskellig reaktionshastighed (varierende fra millisekunder til flere minutter).

En anden hovedtype af numeriske algoritmer til løsning af differentiaalligninger benævnes "multistep" metoder. Idéen i disse algoritmer er, at løsningsfunktionen approksimeres med et polynomium af grad $p > 1$ udfra $p + 1$ punkter på løsningskurven. Denne type algoritmer er som regel hurtigere end tilsvarende "single step" metoder, men har to svagheder. For at starte en sådan algoritme skal man kende $p + 1$ punkter, dvs man fx skal bruge en runge-kutta algoritme til at beregne disse punkter. Ydermere er det i almindelighed vanskeligt at variere skridtlængde, da de fleste "multistep" metoder forudsætter, at de $p + 1$ punkter ligger ækvivalent (dvs med samme afstand). De mest populære "multistep" metoder er Adam-Bashforth algoritmerne [6], side 240-257.

APPENDIX: Taylors formel

Med henblik på anvendelser i kapitel 1 og kapitel 2 skal her vises, hvordan man kan approksimere en funktion med et polynomium. Der gælder følgende vigtige relationer, Taylors formel:

Sætning: Hvis funktionen f er defineret på intervallet $[a, x]$ og er differentiabel af indtil $n+1$ 'te orden på intervallet, da gælder, at

$$f(x) = f(a) + f'(a)(x-a) + \frac{1}{2} f''(a)(x-a)^2 + \dots + \frac{1}{n!} f^{(n)}(a)(x-a)^n + R_n$$

hvor det n -te restled R_n er givet ved

$$R_n = \frac{1}{n!} \int_a^x f^{(n+1)}(t)(x-t)^n dt$$

Bevis: Som bekendt gælder, at

$$\int_a^x f'(t) dt = [f(x)]_a^x = f(x) - f(a)$$

eller

$$(A1) \quad f(x) = f(a) + \int_a^x f'(t) dt$$

I det følgende skal vi gentagne gange benytte sætningen om partiel integration dvs at

$$(A2) \quad \int_a^x h(t)g(t) dt = [H(t)g(t)]_a^x - \int_a^x H(t)g'(t) dt$$

hvor H er en stamfunktion til h .

Først sættes $g(t) = f'(t)$ og $h(t) = 1$, og vi vælger stamfunktionen $H(t) = -(x-t)$. Da får man indsat i (A1), at

$$(A3) \quad f(x) = f(a) + [-f'(t)(x-t)]_a^x + \int_a^x f''(t)(x-t) dt \\ = f(a) + f'(a)(x-a) + \int_a^x f''(t)(x-t) dt$$

Herefter benyttes (A2) med $g(t) = f''(t)$ og $h(t) = x-t$, og man vælger stamfunktionen $H(t) = -(x-t)^2/2$. Anvendt på integralet i (A3) får man

$$\begin{aligned} \int_a^x f''(t)(x-t)dt &= [-f''(t)\frac{1}{2}(x-t)^2]_a^x + \frac{1}{2} \int_a^x f'''(t)(x-t)^2dt \\ &= \frac{1}{2} f''(a)(x-a)^2 + \frac{1}{2} \int_a^x f'''(t)(x-t)^2dt \end{aligned}$$

Indsat i (A3) bliver det til

$$(A4) \quad f(x) = f(a) + f'(a)(x-a) + \frac{1}{2} f''(a)(x-a)^2 + \frac{1}{2} \int_a^x f'''(t)(x-t)^2dt$$

Denne konstruktion kan fortsættes. Lad os antage, at der for $n = p$ gælder, at

$$(A5) \quad f(x) = f(a) + f'(a)(x-a) + \dots + \frac{1}{p!} f^{(p)}(a)(x-a)^p + R_p$$

hvor restleddet

$$(A6) \quad R_p = \frac{1}{p!} \int_a^x f^{(p+1)}(t)(x-t)^p dt$$

Denne gang benyttes $g(t) = f^{(p+1)}(t)$ og

$$h(t) = \frac{1}{p!} (x-t)^p, \text{ og vi vælger } H(t) = -\frac{1}{(p+1)!} (x-t)^{p+1}$$

Da får man indsat i (A6), at

$$\begin{aligned} R_p &= [-f^{(p+1)}(t)\frac{1}{(p+1)!}(x-t)^{p+1}]_a^x + \frac{1}{(p+1)!} \int_a^x f^{(p+2)}(t)(x-t)^{p+1} dt \\ &= \frac{1}{p+1!} f^{(p+1)}(a)(x-a)^{p+1} + \frac{1}{(p+1)!} \int_a^x f^{(p+2)}(t)(x-t)^{p+1} dt \end{aligned}$$

Indsættes dette udtryk for restleddet i (A5) får man

$$(A7) \quad f(x) = f(a) + f'(a)(x-a) + \frac{1}{2} f''(a)(x-a)^2 + \dots \\ \dots + \frac{1}{(p+1)!} f^{(p+1)}(a)(x-a)^{p+1} + R_{p+1}$$

hvor restleddet

$$R_p = \frac{1}{(p+1)!} \int_a^x f^{(p+2)}(t) (x-t)^{p+1} dt$$

Vi har nu godtgjort, at sætningen er rigtig for $n = 1$ (i A3) og for $n = 2$ (i A4). I (A5) antages, at sætningen er rigtig for $n = p$ og til slut vises, at sætningen også gælder for $n = p+1$. Et bevis, som er opbygget på denne måde, kaldes et induktionsbevis. Da vi nu har vist sætningen for $n = 2$, kan vi af induktionstrinet slutte, at den gælder for $n = 3$, og derfor vil den ifølge induktionstrinet også gælde for $n = 4$, osv for alle positive heltal n .

Udtrykket for restleddet kan imidlertid omformuleres en smule. Ifølge integralregningens middelværdisætning, gælder det hvis funktionen h er kontinuert, findes der mindst et tal $x_0 \in [a, x]$, således at

$$\int_a^x h(t)g(t)dt = h(x_0) \int_a^x g(t)dt.$$

Anvendes dette på restleddet i sætningen, vil

$$R_n = f^{(n+1)}(x_0) \frac{1}{n!} \int_a^x (x-t)^n dt$$

eller

$$(A8) \quad R_n = \frac{1}{(n+1)!} f^{(n+1)}(x_0) (x-a)^{n+1}$$

Når man skal anvende Taylors formel, ønsker man at gøre restleddet lille, hvorved approksimationen med et polynomium vil være "god". Da nu f er differentiabel af indtil $n+1$ 'te orden på $[a, x]$, vil $f^{(n+1)}$ være kontinuert på dette interval, og derfor vil $f^{(n+1)}$ antage såvel største - som mindsteværdi på dette interval, dvs $f^{(n+1)}(x_0)$ er begrænset. Dvs "blot x er tæt nok ved a , kan R_n gøres så lille som ønsket". I kapitel 1 og 2 får vi brug for denne indgangsvinkel. Når man der skriver $f(x) = O(h^4)$, vil det betyde, at " R_4 vil være tæt ved 0" eller måske lidt mere præcis " R_4 vil konvergere mod 0 med fjerde potens af $h(=|x-a|)$ "

————— o —————

Man kan dog også benytte Taylors formel til at beregne det approksimerende polynomium til en kendt funktion. Af hensyn til det følgende eksempel og til øvelserne skal først vises en

sætning: Givet følgen (u_k) af positive tal, hvor $u_k \rightarrow u < 1$ for $k \rightarrow \infty$. Da vil følgen (v_n) , hvor $v_n = u_1 \cdot u_2 \cdot \dots \cdot u_n$ opfylde, at $v_n \rightarrow 0$ for $n \rightarrow \infty$.

Bevis: Først vælges r således, at $u < r < 1$ samt et N således, at $u_n \leq r$ for $n \geq N$. Da vil

$$0 < v_n = u_1 u_2 \dots u_N u_{N+1} \dots u_n \leq u_1 u_2 \dots u_N \cdot r^{n-N}$$

eller

$$0 < v_n \leq (u_1 u_2 \dots u_N / r^N) r^n$$

I parentesen står et fast tal, og da $r < 1$, vil $r^n \rightarrow 0$ for $n \rightarrow \infty$, og derfor vil også $v_n \rightarrow 0$ for $n \rightarrow \infty$.

Eksempel: Er $u_k = |x|/k$, vil $u_k > 0$ for alle k og tillige vil $u_k \rightarrow 0$ for $k \rightarrow \infty$. Ifølge sætningen vil således

$$\frac{|x|^n}{n!} = \frac{|x|}{1} \cdot \frac{|x|}{2} \dots \frac{|x|}{n} \rightarrow 0 \text{ for } n \rightarrow \infty$$

Dette vil vi benytte i følgende

Eksempel: Funktionen $f:t \rightarrow e^{at}$ betragtes på intervallet $[0, x]$. f er vilkårligt ofte differentiabel, og

$$f'(t) = ae^{at}, f''(t) = a^2 e^{at}, \dots, f^{(p)}(t) = a^p e^{at}$$

Da nu $f^{(p)}(0) = a^p$ giver Taylors formel

$$e^{ax} = 1 + ax + \frac{1}{2} a^2 x^2 + \frac{1}{6} a^3 x^3 + \dots + \frac{1}{n!} a^n x^n + R_n$$

hvor

$$R_n = \frac{1}{(n+1)!} (ax_0)^{n+1} e^{ax_0}$$

Men da $|ax_0|^{n+1}/(n+1)! \rightarrow 0$ for $n \rightarrow \infty$, kan e^{ax} approksimeres vilkårligt godt med et sådant polynomium.

Øvelse A1: Betragt funktionen $f:t \rightarrow \cos t$ på intervallet $[0, x]$. Beregn det til $\cos x$ svarende polynomium og vis, at restleddet kan gøres vilkårligt lille, blot der medtages led nok i polynomiet.

Øvelse A2: Gør som i øvelse A1 blot for funktionen $f:t \rightarrow \sin t$.

Opgaver til kapitel 1:

1. Ved et fixpunkt for en funktion ϕ forstås et tal x som opfylder:

$$\phi(x) = x$$

Geometrisk betyder det, at fixpunktet findes der, hvor grafen for ϕ skærer linien $y=x$. At bestemme fixpunkt for ϕ svarer til at bestemme nulpunkt for funktionen f givet ved

$$(*) \quad f(x) = \phi(x) - x$$

Tegn ved brug af programmet "graf" for hver af funktionerne

$$\phi_1: x \rightarrow (x^3+2)/3$$

$$\phi_2: x \rightarrow 4\ln x$$

$$\phi_3: x \rightarrow 1 - \cos 2x$$

den tilhørende f -funktion (jfr. (*)) i et passende interval.

2. Når en funktion ikke er defineret i et punkt, vil programmet "graf" naturligvis ikke kunne tegne funktionens graf gennem dette punkt. Lav en ændring i "graf", som tager højde for denne situation. Dette kan fx gøres ved at tegne til punktet $(x-1/10, f(x-1/10))$, flytte pennen til punktet $(x+1/10, f(x+1/10))$ og tegne videre.

Tegn derefter hver af funktionerne i et passende interval

$$f_1: x \rightarrow e^x - 1 - 1/x$$

$$f_2: x \rightarrow x - e^{1/x}$$

$$f_3: x \rightarrow x - \ln(x^2 - 1).$$

3. Bestem ved brug af "falsi" eller "newton" fixpunkter for hver af funktionerne ϕ_1 , ϕ_2 og ϕ_3 i opgave 1.

4. Bestem ved brug af "falsi" eller "newton" nulpunkter for en eller flere af funktionerne i øvelse 1 og opgave 2.

5. Betragt funktionen f givet ved

$$f(x) = x^3 - 3x + 2$$

der jo har $x-1$ som faktor to gange, dvs at $f'(1) = 0$. Forsøg med både "falsi" og "newton" at bestemme et nulpunkt i intervallet $[0.7, 1.3]$. Hvorfor vil dette ikke lykkes med "falsi"?

Betragt dernæst funktionen

$$g: x \rightarrow x^3 - 3x + 1.99$$

og forsøg at bestemme de to nulpunkter i intervallet $[0.7, 1.3]$ med "nulvalg".

Bemærk, at antal iterationer for newton-raphson ved beregning på f er større end ved beregning på g (jfr. øvelse 8)

6. Betragt de tre funktioner f_i givet ved

$$\begin{aligned} f_1(x) &= x^4 - 2x^3 + 2x - 1 \\ f_2(x) &= x^4 - 2x^3 + 2x - 0.9999999 \\ f_3(x) &= x^4 - 2x^3 + 2x - 0.999999 \end{aligned}$$

Foretag en faktoropløsning af f_1 . Bestem ved brug af "newton" et nulpunkt for hver af de tre funktioner med start i intervallet $[0.9, 1.1]$. Bemærk igen antallet af iterationer, der benyttes ved f_1 i forhold til det antal, der benyttes ved f_2 og f_3 (jfr. øvelse 8). Tilmed er nøjagtigheden i bestemmelsen af nulpunktet ($x=1$) til f_1 meget upræcis, hvorfor?

Bestem tillige nulpunkterne for f_2 og f_3 ved brug af "bisek" og "falsi". Hvorfor benytter "falsi" så mange iterationer i forhold til "bisek"?

7. Skitsér grafen for funktionen

$$f: x \rightarrow x^3 - 3x + \frac{5}{2}$$

Forsøg på denne figur med start i $x_0 = 0$ at bestemme et nulpunkt for f ved brug af newton-raphson algoritmen. Forsøg det samme med den forbedrede newton-raphson.

8. Tegn ved brug af "graf" en graf for funktionen

$$f: x^4 - \frac{8}{3}x^3 - 2x^2 + 8x - \frac{7}{3}$$

Forsøg på denne figur med start i $x_0 = -0.8$ at bestemme et nulpunkt for f ved brug af newton-raphson og forbedret newton-raphson.

9. Overfør de i opgave 2 foreslåede ændringer i "graf" til programmet "nulpunkt" og foretag derpå en total nulpunktsbestemmelse af funktionerne i opgave 2.

10. Bestem samtlige fixpunkter for ϕ -funktionerne i opgave 1.

11. Foretag en total nulpunktsbestemmelse af funktionerne i øvelse 1.

12. Foretag på baggrund af øvelse 12 en total nulpunkts- og ekstremumsbestemmelse af funktionerne

$$\begin{aligned} f_1: x \rightarrow e^x - x^3 - 1 \\ f_2: x \rightarrow x^3 - 4x^2 - 2x + 4 \\ f_3: x \rightarrow x^3 - 3x + 2 \end{aligned}$$

13. Foretag for hver af funktionerne

$$g_1: x \rightarrow \frac{1}{x-1} - \frac{4}{x^2}$$

$$g_2: x \rightarrow e^x - 1 - 1/x$$

en funktionsundersøgelse. Undersøg med "nulpunkt" ændret i overensstemmelse med øvelse 12 og 13 om programmet kan give den samme analyse.

Opgaver til kapitel 2:

1. Betragt den autonome differentialligning

$$y' = f(y)$$

Et punkt y kaldes et ligevægtspunkt for differentialligningen, hvis

$$y' = 0 \text{ (dvs } f(y) = 0 \text{)}.$$

Vis, at de to differentialligninger

$$\begin{aligned} y' &= a^2 y^{2/3} - b^2 y \\ y' &= a y^{2/3} - b y^{5/6} \end{aligned}$$

vil have de samme ligevægtspunkter, $a, b > 0$.

Tegn ved hjælp af programmet "rk4mono" en løsningskurve for hver af differentialligningerne gennem punktet $(t_0, y_0) = (0, 1)$. Vælg fx $a=3$ og $b=1$. Vil der være kvalitativ(?) forskel på de to løsningskurver?

2. Vis, at hver af følgende differentialligninger

$$\begin{aligned} y' &= ay(b-y) \\ y' &= ay(b-y)^{3/2} \\ y' &= aby \cos\left(\frac{\pi y}{2b}\right) \\ y' &= aby \left(1 + \cos\left(\frac{\pi y}{b}\right)\right) \end{aligned}$$

vil have de samme ligevægtspunkter (jfr. opgave 1). Tegn ved brug af programmet "rk4mono" en løsningskurve for hver differentialligning gennem punktet $(t_0, y_0) = (0, 1)$. Vælg fx $a = 0.01$ og $b = 270$. Disse løsningskurver skulle gerne udvise kvalitativt samme forløb!

Prøv selv at foreslå differentialligninger, som udviser de samme træk.

3. Betragt igen den autonome differentialligning

$$y' = f(y)$$

Det (eller de) punkt(er) på en løsningskurve $y(t)$ til differen-

tialligningen, hvor væksthastigheden er størst, findes som bekendt af $y''(t) = 0$. Da imidlertid $y'' = f'(y)y'$, og da $y' > 0$, må der altså gælde, at

$$f'(y) = 0$$

Vis, at den største væksthastighed på en løsningskurve til $y' = ay(b-y)$

vil forekomme i punktet $y = b/2$. Bestem også for de øvrige differentiaalligninger i opgave 2 den y -værdi, som giver størst væksthastighed. Bemærk, at man i de to sidste tilfælde må benytte en nulpunktsalgoritme fra kapitel 1; vælg fx $a = 0.01$ og $b = 270$.

4. Bestem for differentiaalligningerne i opgave 1 den største væksthastighed.

5. Bestem den fuldstændige løsning til differentiaalligningen

$$y' = y \ln 2 + 2t - t^2 \ln 2$$

(Vis, fx at funktionen $t \rightarrow t^2$ er én løsning til differentiaalligningen) og skitsér den løsningskurve, som går gennem punktet $(t_0, y_0) = (0, -1)$.

6. Benyt programmet "rk4mono" til at bestemme numeriske løsninger til differentiaalligningen i opgave 5 (selv om denne ligning ikke er autonom, kan programmet benyttes, da man i linie 550 har IMPORTeret t).

Tegn et passende antal løsningskurver til differentiaalligningen, når $t_0 = 0$ og $y_0 \in [-2, 0]$. Vælg fx $t_1 = 7$. Vurdér på baggrund af disse løsningskurver, hvordan løsningskurverne vil forløbe, når $y_0 < -2$ eller $y_0 > 0$.

7. Indlæs den analytiske løsning og differentiaalligningen fra opgave 5 i programmet "fisk" og sammenlign den numeriske og den analytiske løsning gennem punktet $(t_0, y_0) = (0, -1)$. Vælg fx $\text{eps} = 10^{-8}$ og $t_1 = 7$.

8. En simpel model for volumen af en kræftsvulst kan beskrives ved differentiaalligningen

$$y' = ae^{-bt}y$$

hvor $a, b > 0$, jfr. [4], side 95-97. Bestem den fuldstændige løsning til denne differentiaalligning. Vis ud fra denne løsning at der vil finde et maksimalt volumen og en maksimal væksthastighed.

9. Benyt programmet "rk4mono" til at tegne en løsning til differentiaalligningen i opgave 8 gennem punktet $(t_0, y_0) = (0, 1)$. Vælg fx $a = 1$, $b = 0.2$ og $t_1 = 20$. Sammenlign de fundne maksimale værdier i opgave 8 med den udtegnede kurve.

10. Indlæs den analytiske løsning og differentiaalligningen fra opgave 8 i programmet "fisk" og sammenlign den analytiske og den numeriske løsning gennem punktet $(t_0, y_0) = (0, 1)$. Vælg fx $a = 1$, $b = 0.2$ og $t_1 = 20$.

11. Bestem såvel en analytisk som en numerisk løsning til hver af differentiaalligningerne

$$\begin{aligned}y' &= y/t-1 \\y' &= (y/t)^2+2 \\y' &= t/(2y)\end{aligned}$$

12. Differentiaalligningen

$$y''+a(y^2-1)y'+y=0$$

kaldes van der Pols ligning, [1], side 151-156. Ligningen er en model for et elektrisk kredsløb, hvori modstanden har ikke-lineær karakteristik. Omskriv ligningen til et sæt af autonome differentiaalligninger af 1.orden.

Foretag ved brug af programmet "sving" en numerisk løsning af van der Pols ligning. Vælg $a \in]0, 1]$. Hvordan vil løsningskurverne variere med a ?

13. En differentiaalligning af 2.orden givet ved

$$y''+g(y)y'+h(y) = f(t)$$

kan imidlertid også på anden måde omskrives til et sæt af autonome differentiaalligninger af første orden. Sættes $x' = -h(y)$, kan man ved integration af ligningen (mht t) få

$$y'+G(y)-x = F(t)$$

hvor G og F er stamfunktioner til g hhv f (prøv selv!). Således bliver ligningssystemet

$$\begin{aligned}t' &= 1 \\x' &= -h(y) \\y' &= x-G(y)+F(t)\end{aligned}$$

Dette sæt af ligninger kaldes Liénards ligninger, [1], side 179-180. Omskriv van der Pols ligning (jfr. opgave 12) til de tilhørende Liénards ligninger og løs ligningerne numerisk ved brug af programmet "sving". Vælg $a \in]0, 1]$.

14. Der er udviklet en lang række "rovdyr-byttedyr" modeller; den første af disse er Lotka-Volterra modellen (1925-26)

$$\begin{aligned}x' &= ax-bxy \\y' &= -cy+dxy\end{aligned}$$

hvor $a, b, c, d > 0$, jfr. [1], side 140-146. Denne model er som regel for simpel og er siden blevet videreudviklet i mange varianter. En af de bedre blandt disse modeller er Holling-Tanner modellen

$$x' = ax(1-x/N) - \frac{bxy}{x+D}$$

$$y' = cy(1-y/x)$$

hvor $a, b, c, N, D, K > 0$, jfr [1], side 147-151.

Vælg værdier for de indgående konstanter i Lotka-Volterra modellen og foretag ved brug af programmet "sving" en kørsel af denne model. Det vil vise sig, at alle løsningskurver vil være lukkede kurver, dvs at selv små fejl i modelforudsætningerne kan resultere i, at løsningskurverne vil divergere eller vil konvergere mod en ny grænsesituation (grænsecykel eller -punkt).

Vælg tilsvarende de resterende værdier i Holling-Tanner modellen og foretag ved brug af programmet "sving" en kørsel af denne model. Overvej om systemets opførsel i denne situation er mere rimelig.

Vis, at begge modeller vil have ligevægtspunkter (dvs hvor såvel $x' = 0$ som $y' = 0$).

15. Omskriv hver af differentiaalligningerne

$$y'' + ty' - y = 0$$

$$y'' + 2/t^2 y' - y = 0$$

$$y'' + 2/t y' - y = 1/t$$

til et autonomt system af differentiaalligninger af 1.orden, og løs derpå ved brug af programmet "sving" hver af dem gennem punktet (1,1).

16. Lav en ændring af programmet "sving", således at differentiaalligningerne kan læses "baglæns", dvs for aftagende t . Løs derpå ligningerne i opgave 15 gennem punktet (1,1), fx indtil $t_1 = 0.01$.

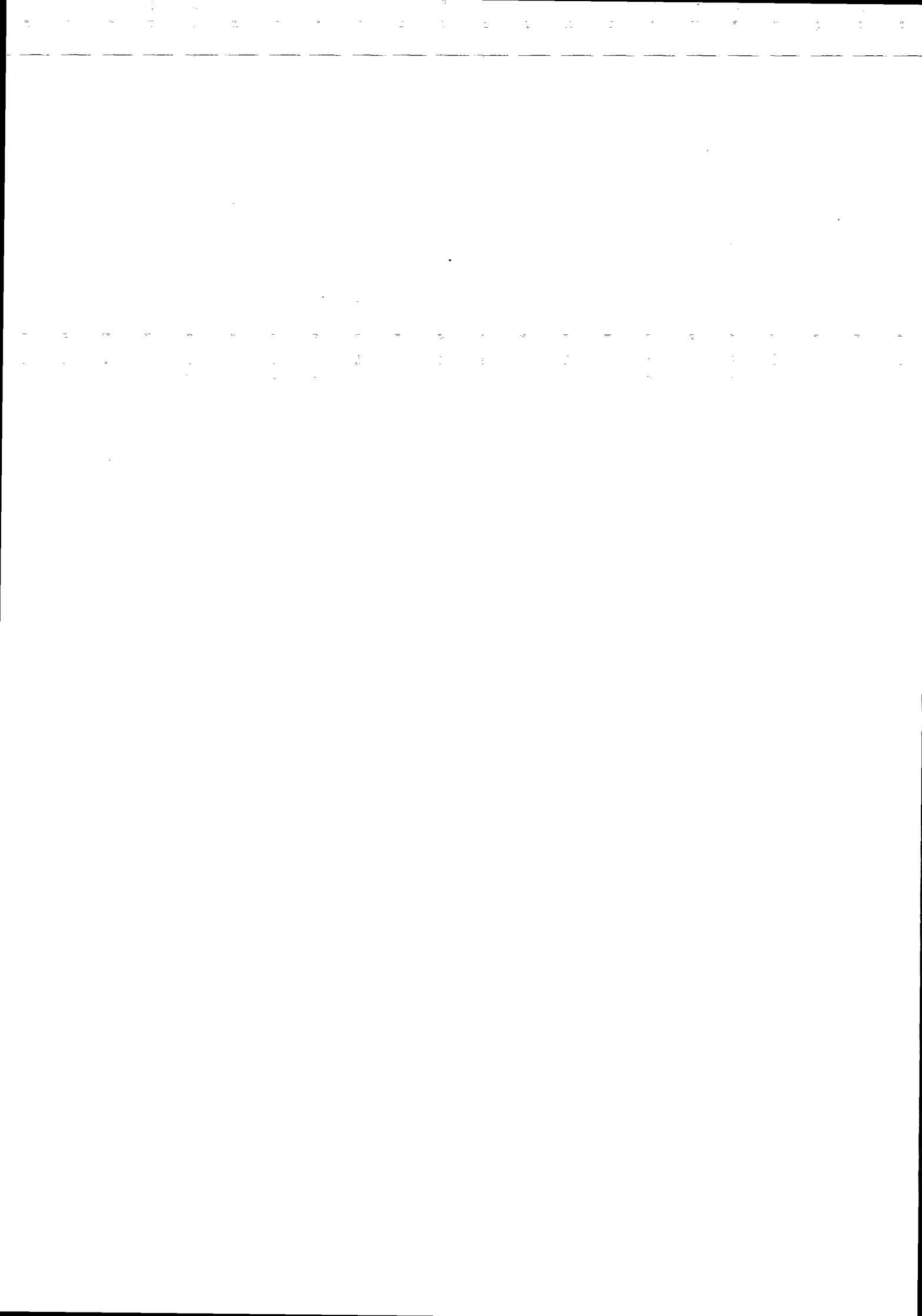
Litteraturliste:

- [1] D.K. Arrowsmith and C.M. Place: "Ordinary Differential Equations" (Chapmann and Hall, 1982)
- [2] E. Christiansen: "Introduktion til numerisk analyse" (Odense Universitetstrykkeri, 1982)
- [3] L. Fox: "Numerical Solutions of Ordinary and Partial Differential Equations" (Pergamon Press, 1962)
- [4] M.B. Heefelt: "Differentielligningsmodeller", Gyldendal, 1980)
- [5] P. Henrici: "Discrete variable Methods in Ordinary Differential Equations" (J. Wiley & Sons, 1962)
- [6] F. Hildebrand: "Introduction to Numerical Analysis" (McGraw-Hill, 1974)
- [7] M. Kubicek and M. Marek: "Computational Methods in Bifurcation. Theory and Dissipative Structures" (Springer, 1983)

Liste over programmer på disketten:

Til kapitel 1: "graf", "bisek", "falsi", "newton", "nulvalg", "dobbel", "dobbfkt", "nulpunkt", "extremum", "falsipl".

Til kapitel 2: "rk4mono", "fisk", "diffvalg". "rk2", "radio", "sving", "lorenz", "rkmono", "rkm".



- 1/78 "TANKER OM EN PRAKSIS" - et matematikprojekt. Projekt rapport af: Anne Jensen, Lena Lindenskov, Marianne Kesselhahn og Nicolai Lomholt. Vejleder: Anders Madsen
- 2/78 "OPTIMERING" - Menneskets forøgede beherskelsesmuligheder af natur og samfund. Projekt rapport af: Tom J. Andersen, Tommy R. Andersen, Gert Krenøe og Peter H. Lassen. Vejleder: Bernhelm Booss.
- 3/78 "OPCAVESAMLING", breddekursus i fysik. Af: Lasse Rasmussen, Aage Bonde Kræmmer og Jens Højgaard Jensen.
- 4/78 "TRE ESSAYS" - om matematikundervisning, matematiklæreruddannelsen og videnskabsrindalismen. Af: Mogens Niss. Nr. 4 er p.t. udgået.
- 5/78 "BIBLIOGRAFISK VEJLEDNING til studiet af DEN MODERNE FYSIKS HISTORIE". Af: Helge Kragh. Nr. 5 er p.t. udgået.
- 6/78 "NOGLE ARTIKLER OG DEBATINDLÆG OM - læreruddannelse og undervisning i fysik, og - de naturvidenskabelige fags situation efter studenteroprøret". Af: Karin Beyer, Jens Højgaard Jensen og Bent C. Jørgensen.
- 7/78 "MATEMATIKKENS FORHOLD TIL SAMFUNDSØKONOMIEN". Af: B.V. Gnedenko. Nr. 7 er udgået.
- 8/78 "DYNAMIK OG DIAGRAMMER". Introduktion til energy-bond-graph formalismen. Af: Peder Voetmann Christiansen.
- 9/78 "OM PRAKSIS' INDFLYDELSE PÅ MATEMATIKKENS UDVIKLING". - Motiver til Kepler's: "Nova Stereometria Doliorum Vinarium". Projekt rapport af: Lasse Rasmussen. Vejleder: Anders Madsen.
-
- 10/79 "TERMODYNAMIK I GYMNASIET". Projekt rapport af: Jan Christensen og Jeanne Mortensen. Vejledere: Karin Beyer og Peder Voetmann Christiansen.
- 11/79 "STATISTISKE MATERIALER". Af: Jørgen Larsen.
- 12/79 "LINEÆRE DIFFERENTIALLIGNINGER OG DIFFERENTIALLIGNINGSSYSTEMER". Af: Mogens Brun Heefelt. Nr. 12 er udgået.
- 13/79 "CAVENDISH'S FORSØG I GYMNASIET". Projekt rapport af: Gert Kreinøe. Vejleder: Albert Chr. Paulsen.
- 14/79 "BOOKS ABOUT MATHEMATICS: History, Philosophy, Education, Models, System Theory, and Works of". Af: Else Høyrup. Nr. 14 er p.t. udgået.
- 15/79 "STRUKTUREL STABILITET OG KATASTROFER i systemer i og udenfor termodynamisk ligevægt". Specialeopgave af: Leif S. Striegler. Vejleder: Peder Voetmann Christiansen.
- 16/79 "STATISTIK I KRÆFTFORSKNINGEN". Projekt rapport af: Michael Olsen og Jørn Jensen. Vejleder: Jørgen Larsen.
- 17/79 "AT SPØRGE OG AT SVARE i fysikundervisningen". Af: Albert Christian Paulsen.
- 18/79 "MATHEMATICS AND THE REAL WORLD", Proceedings af an International Workshop, Roskilde University Centre, Denmark, 1978. Preprint. Af: Bernhelm Booss og Mogens Niss (eds.)
- 19/79 "GEOMETRI, SKOLE OG VIRKELIGHED". Projekt rapport af: Tom J. Andersen, Tommy R. Andersen og Per H.H. Larsen. Vejleder: Mogens Niss.
- 20/79 "STATISTISKE MODELLER TIL BESTEMMELSE AF SIKRE DOSER FOR CARCINOGENE STOFFER". Projekt rapport af: Michael Olsen og Jørn Jensen. Vejleder: Jørgen Larsen.
- 21/79 "KONTROL I GYMNASIET-FORMÅL OG KONSEKVENSER". Projekt rapport af: Crilles Bacher, Per S. Jensen, Preben Jensen og Torben Nysteen.
- 22/79 "SEMIOTIK OG SYSTEMEGENSKABER (1)". 1-port lineært response og støj i fysikken. Af: Peder Voetmann Christiansen.
- 23/79 "ON THE HISTORY AF EARLY WAVE MECHANICS - with special emphasis on the role af reality". Af: Helge Kragh.
-
- 24/80 "MATEMATIKOPFATTELSE HOS 2.C'ERE". a+b 1. En analyse. 2. Interviewmateriale. Projekt rapport af: Jan Christensen og Knud Lindhardt Rasmussen. Vejleder: Mogens Niss.
- 25/80 "EKSAMENSOPGAVER", Dybdemodulet/fysik 1974-79.
- 26/80 "OM MATEMATISKE MODELLER". En projekt rapport og to artikler. Af: Jens Højgaard Jensen m.fl.
- 27/80 "METHODOLOGY AND PHILOSOPHY AF SCIENCE IN PAUL DIRAC'S PHYSICS". Af: Helge Kragh.
- 28/80 "DILENTRISK RELAXATION - et forslag til en ny model bygget på væskernes viscoelastiske egenskaber". Projekt rapport af: Gert Kreinøe. Vejleder: Niels Boye Olsen.
- 29/80 "ODIN - undervisningsmateriale til et kursus i differentiaalligningsmodeller". Projekt rapport af: Tommy R. Andersen, Per H.H. Larsen og Peter H. Lassen. Vejleder: Mogens Brun Heefelt.
- 30/80 "FUSIONSENERGIEN - - - ATOMSAMFUNDETS ENDESTATION". Af: Oluf Danielsen. Nr. 30 er udgået.
- 31/80 "VIDENSKABSTEORETISKE PROBLEMER VED UNDERVISNINGSSYSTEMER BASERET PÅ MÅNGDELERE". Projekt rapport af: Troels Lange og Jørgen Karrebæk. Vejleder: Stig Andur Pedersen. Nr. 31 er p.t. udgået.
- 32/80 "POLYMERE STOFFERS VISCOELASTISKE EGENSKABER - BELYST VED HJÆLP AF MEKANISKE IMPEDANSMÅLINGER MØSSBAUEREFFEKTIVLIGNINGER". Projekt rapport af: Crilles Bacher og Preben Jensen. Vejledere: Niels Boye Olsen og Peder Voetmann Christiansen.
- 33/80 "KONSTITUERING AF FAG INDEN FOR TEKNISK - NATURVIDENSKABELIGE UDDANNELSER. I-II". Af: Arne Jakobsen.
- 34/80 "ENVIRONMENTAL IMPACT AF WIND ENERGY UTILIZATION". ENERGY SERIES NO. I. Af: Bent Sørensen. Nr. 34 er udgået.

- 35/80 "HISTORISKE STUDIER I DEN NYERE ATOMFYSIKS UDVIKLING".
Af: Helge Kragh.
- 36/80 "HVAD ER MENINGEN MED MATEMATIKUNDERVISNINGEN?".
Fire artikler.
Af: Mogens Niss.
- 37/80 "RENEWABLE ENERGY AND ENERGY STORAGE".
ENERGY SERIES NO. 2.
Af: Bent Sørensen.
-
- 38/81 "TIL EN HISTORIE TEORI OM NATURERKENDELSE, TEKNOLOGI OG SAMFUND".
Projektrapport af: Erik Gade, Hans Hedal, Henrik Lau og Finn Physant.
Vejledere: Stig Andur Pedersen, Helge Kragh og Ib Thiersen.
Nr. 38 er p.t. udgået.
- 39/81 "TIL KRITIKKEN AF VÆKSTØKONOMIEN".
Af: Jens Højgaard Jensen.
- 40/81 "TELEKOMMUNIKATION I DANMARK - oplæg til en teknologivurdering".
Projektrapport af: Arne Jørgensen, Bruno Petersen og Jan Vedde.
Vejleder: Per Nørregaard.
- 41/81 "PLANNING AND POLICY CONSIDERATIONS RELATED TO THE INTRODUCTION OF RENEWABLE ENERGY SOURCES INTO ENERGY SUPPLY SYSTEMS".
ENERGY SERIES NO. 3.
Af: Bent Sørensen.
- 42/81 "VIDENSKAB TEORI SAMFUND - En introduktion til materialistiske videnskabsopfattelser".
Af: Helge Kragh og Stig Andur Pedersen.
- 43/81 1. "COMPARATIVE RISK ASSESSMENT OF TOTAL ENERGY SYSTEMS".
2. "ADVANTAGES AND DISADVANTAGES OF DECENTRALIZATION".
ENERGY SERIES NO. 4.
Af: Bent Sørensen.
- 44/81 "HISTORISKE UNDERSØGELSER AF DE EKSPERIMENTELLE FORUDSÆTNINGER FOR RUTHERFORDS ATOMMODEL".
Projektrapport af: Niels Thor Nielsen.
Vejleder: Bent C. Jørgensen.
-
- 45/82 Er aldrig udkommet.
- 46/82 "EKSEMPLARISK UNDERVISNING OG FYSISK ERKENDELSE-1+11 ILLUSTRERET VED TO EKSEMPLER".
Projektrapport af: Torben O. Olsen, Lasse Rasmussen og Niels Dreyer Sørensen.
Vejleder: Bent C. Jørgensen.
- 47/82 "BARSEBÄCK OG DET VÆRST OFFICIELT-TÆNKELIGE UHELD".
ENERGY SERIES NO. 5.
Af: Bent Sørensen.
- 48/82 "EN UNDERSØGELSE AF MATEMATIKUNDERVISNINGEN PÅ ADGANGSKURSUS TIL KØBENHAVNS TEKNIKUM".
Projektrapport af: Lis Eilertzen, Jørgen Karrebæk, Troels Lange, Preben Nørregaard, Lissi Pedersen, Laust Rishøj, Lill Røn og Isac Showiki.
Vejleder: Mogens Niss.
- 49/82 "ANALYSE AF MULTISPEKTRALE SATELLITBILLEDER".
Projektrapport af: Preben Nørregaard.
Vejledere: Jørgen Larsen og Rasmus Ole Rasmussen.
- 50/82 "HERSLEV - MULIGHEDER FOR VEDVARENDE ENERGI I EN LANDSBY".
ENERGY SERIES NO. 6.
Rapport af: Bent Christensen, Bent Hove Jensen, Dennis B. Møller, Bjarne Laursen, Bjarne Lillethorup og Jacob Mørch Pedersen.
Vejleder: Bent Sørensen.
- 51/82 "HVAD KAN DER Gøres FOR AT AFHJÆLPE PICERS BLOKERING OVERFOR MATEMATIK ?"
Projektrapport af: Lis Eilertzen, Lissi Pedersen, Lill Røn og Susanne Stender.
- 52/82 "DESUSPENSION OF SPLITTING ELLIPTIC SYMBOLS".
Af: Bernhelm Booss og Krzysztof Wojciechowski.
- 53/82 "THE CONSTITUTION OF SUBJECTS IN ENGINEERING EDUCATION".
Af: Arne Jacobsen og Stig Andur Pedersen.
- 54/82 "FUTURES RESEARCH" - A Philosophical Analysis of Its Subject-Matter and Methods.
Af: Stig Andur Pedersen og Johannes Witt-Hansen.
- 55/82 "MATEMATISKE MODELLER" - Litteratur på Roskilde Universitetsbibliotek.
En biografi.
Af: Else Høytrup.
Vedr. tekst nr. 55/82 se også tekst nr. 62/83.
- 56/82 "EN - TO - MANGE" -
En undersøgelse af matematisk økologi.
Projektrapport af: Troels Lange.
Vejleder: Anders Madsen.
-
- 57/83 "ASPECT EKSPERIMENTET" -
Skjulte variable i kvantemekanikken?
Projektrapport af: Tom Juul Andersen.
Vejleder: Peder Voetmann Christiansen.
Nr. 57 er udgået.
- 58/83 "MATEMATISKE VANDRINGER" - Modelbetragtninger over spredning af dyr mellem småbiotoper i agerlandet.
Projektrapport af: Per Hammershøj Jensen og Lene Vagn Rasmussen.
Vejleder: Jørgen Larsen.
- 59/83 "THE METHODOLOGY OF ENERGY PLANNING".
ENERGY SERIES NO. 7.
Af: Bent Sørensen.
- 60/83 "MATEMATISK MODEKSPERTISE" - et eksempel.
Projektrapport af: Erik O. Gade, Jørgen Karrebæk og Preben Nørregaard.
Vejleder: Anders Madsen.
- 61/83 "FYSIKS IDEOLOGISKE FUNKTION, SOM ET EKSEMPEL PÅ EN NATURVIDENSKAB - HISTORISK SET".
Projektrapport af: Annette Post Nielsen.
Vejledere: Jens Høytrup, Jens Højgaard Jensen og Jørgen Vogelius.
- 62/83 "MATEMATISKE MODELLER" - Litteratur på Roskilde Universitetsbibliotek.
En biografi 2. rev. udgave.
Af: Else Høytrup.
- 63/83 "CREATING ENERGY FUTURES: A SHORT GUIDE TO ENERGY PLANNING".
ENERGY SERIES No. 8.
Af: David Crossley og Bent Sørensen.
- 64/83 "VON MATEMATIK UND KRIEG".
Af: Bernhelm Booss og Jens Høytrup.
- 65/83 "ANVENDT MATEMATIK - TEORI ELLER PRAKSIS".
Projektrapport af: Per Hedegård Andersen, Kirsten Habekost, Carsten Holst-Jensen, Annelise von Moos, Else Marie Pedersen og Erling Møller Pedersen.
Vejledere: Bernhelm Booss og Klaus Grünbaum.
- 66/83 "MATEMATISKE MODELLER FOR PERIODISK SELEKTION I ESCHERICHIA COLI".
Projektrapport af: Hanne Lisbet Andersen, Ole Richard Jensen og Klavs Frisdahl.
Vejledere: Jørgen Larsen og Anders Hede Madsen.
- 67/83 "ELEPSOIDE METODEN - EN NY METODE TIL LINEÆR PROGRAMMERING?".
Projektrapport af: Lone Billmann og Lars Boye.
Vejleder: Mogens Brun Heefelt.
- 68/83 "STOKASTISKE MODELLER I POPULATIONSGENETIK" - til kritikken af teoriladede modeller.
Projektrapport af: Lise Odgård Gade, Susanne Hansen, Michael Hviid og Frank Mølgård Olsen.
Vejleder: Jørgen Larsen.

- 69/83 "ELEVFORUDSÆTNINGER I FYSIK"
- en test i l.g med kommentarer.
Af: Albert C. Paulsen.
- 70/83 "INDLÆRINGS - OG FORMIDLINGSPROBLEMER I MATEMATIK PÅ VOKSENUNDERVISNINGSNIVEAU".
Projekt rapport af: Hanne Lisbet Andersen, Torben J. Andreasen, Svend Åge Houmann, Helle Glerup Jensen, Keld Fl. Nielsen, Lene Vagn Rasmussen.
Vejleder: Klaus Grünbaum og Anders Hede Madsen.
- 71/83 "PIGER OG FYSIK"
- et problem og en udfordring for skolen?
Af: Karin Beyer, Sussanne Blegaa, Birthe Olsen, Jette Reich og Mette Vedelsby.
- 72/83 "VERDEN IFØLGE PEIRCE" - to metafysiske essays, om og af C.S Peirce.
Af: Peder Voetmann Christiansen.
- 73/83 "'EN ENERGIANALYSE AF LANDERUG"
- økologisk contra traditionelt.
ENERGY SERIES NO. 9
Specialeopgave i fysik af: Bent Hove Jensen.
Vejleder: Bent Sørensen.
-
- 74/84 "MINIATURISERING AF MIKROELEKTRONIK" - om videnskabeliggjort teknologi og nytten af at lære fysik.
Projekt rapport af: Bodil Harder og Linda Szkotak Jensen.
Vejledere: Jens Højgaard Jensen og Bent C. Jørgensen.
- 75/84 "MATEMATIKUNDERVISNINGEN I FREMTIDENS GYMNASIUM"
- Case: Lineær programmering.
Projekt rapport af: Morten Blomhøj, Klavs Frisdahl og Frank Mølgaard Olsen.
Vejledere: Mogens Brun Heefelt og Jens Bjørneboe.
- 76/84 "KERNEKRAFT I DANMARK?" - Et hørings svar indkaldt af miljøministeriet, med kritik af miljøstyrelsens rapporter af 15. marts 1984.
ENERGY SERIES No. 10
Af: Niels Boye Olsen og Bent Sørensen.
- 77/84 "POLITISKE INDEKS - FUP ELLER FAKTA?"
Opinionsundersøgelser belyst ved statistiske modeller.
Projekt rapport af: Svend Åge Houmann, Keld Nielsen og Susanne Stender.
Vejledere: Jørgen Larsen og Jens Bjørneboe.
- 78/84 "JÆVNSTRØMSLEDNINGSEVNE OG GITTERSTRUKTUR I AMORFT GERMANIUM".
Specialrapport af: Hans Hedal, Frank C. Ludvigsen og Finn C. Physant.
Vejleder: Niels Boye Olsen.
- 79/84 "MATEMATIK OG ALMENDANNELSE".
Projekt rapport af: Henrik Ooster, Mikael Wennerberg Johansen, Povl Kattler, Birgitte Lydholm og Morten Overgaard Nielsen.
Vejleder: Bernhelm Booss.
- 80/84 "KURSUSMATERIALE TIL MATEMATIK B".
Af: Mogens Brun Heefelt.
- 81/84 "FREKVENSafhængig ledningsevne i amorft germanium".
Specialrapport af: Jørgen Wind Petersen og Jan Christensen.
Vejleder: Niels Boye Olsen.
- 82/84 "MATEMATIK - OG FYSIKUNDERVISNINGEN I DET AUTOMATISEREDE SAMFUND".
Rapport fra et seminar afholdt i Hvidovre 25-27 april 1983.
Red.: Jens Højgaard Jensen, Bent C. Jørgensen og Mogens Niss.
- 83/84 "ON THE QUANTIFICATION OF SECURITY":
PEACE RESEARCH SERIES NO. 1
Af: Bent Sørensen
nr. 83 er p.t. udgået
- 84/84 "NOGLE ARTIKLER OM MATEMATIK, FYSIK OG ALMENDANNELSE".
Af: Jens Højgaard Jensen, Mogens Niss m. fl.
- 85/84 "CENTRIFUGALREGULATORER OG MATEMATIK".
Specialrapport af: Per Hødegård Andersen, Carsten Holst-Jensen, Else Marie Pedersen og Erling Møller Pedersen.
Vejleder: Stig Andur Pedersen.
- 86/84 "SECURITY IMPLICATIONS OF ALTERNATIVE DEFENSE OPTIONS FOR WESTERN EUROPE".
PEACE RESEARCH SERIES NO. 2
Af: Bent Sørensen.
- 87/84 "A SIMPLE MODEL OF AC HOPPING CONDUCTIVITY IN DISORDERED SOLIDS".
Af: Jeppe C. Dyre.
- 88/84 "RISE, FALL AND RESURRECTION OF INFINITESIMALS".
Af: Detlef Laugwitz.
- 89/84 "FUERNVARMOPTIMERING".
Af: Bjarne Lillethorup og Jacob Mørch Pedersen.
- 90/84 "ENERGI I L.G - EN TEORI FOR TILRETTTELÆGGELSE".
Af: Albert Chr. Paulsen.
-
- 91/85 "KVANTETEORI FOR GYMNASIET".
1. Lærervejledning
Projekt rapport af: Biger Lundgren, Henning Sten Hansen og John Johansson.
Vejleder: Torsten Meyer.
- 92/85 "KVANTETEORI FOR GYMNASIET".
2. Materiale
Projekt rapport af: Biger Lundgren, Henning Sten Hansen og John Johansson.
Vejleder: Torsten Meyer.
- 93/85 "THE SEMIOTICS OF QUANTUM - NON - LOCALITY".
Af: Peder Voetmann Christiansen.
- 94/85 "TREENIGHEDEN BOURBAKI - generalen, matematikeren og ånden".
Projekt rapport af: Morten Blomhøj, Klavs Frisdahl og Frank M. Olsen.
Vejleder: Mogens Niss.
- 95/85 "AN ALTERNATIV DEFENSE PLAN FOR WESTERN EUROPE".
PEACE RESEARCH SERIES NO. 3
Af: Bent Sørensen.
- 96/85 "ASPEKTER VED KRAFTVARMEFORSYNING".
Af: Bjarne Lillethorup.
Vejleder: Bent Sørensen.
- 97/85 "ON THE PHYSICS OF A.C. HOPPING CONDUCTIVITY".
Af: Jeppe C. Dyre.
- 98/85 "VALGMULIGHEDER I INFORMATIONSDEREN".
Af: Bent Sørensen.
- 99/85 "Der er langt fra Q til R".
Projekt rapport af: Niels Jørgensen og Mikael Klintorp.
Vejleder: Stig Andur Pedersen.
- 100/85 "TALSYSTEMETS OPBYGNING".
Af: Mogens Niss.
- 101/85 "EXTENDED MOMENTUM THEORY FOR WINDMILLS IN PERTURBATIVE FORM".
Af: Ganesh Sengupta.
- 102/85 OPSTILLING OG ANALYSE AF MATEMATISKE MODELLER, BELYST VED MODELLER OVER KØRS FODEROPTAGELSE OG - OMSÆTNING".
Projekt rapport af: Lis Eilertzen, Kirsten Habekost, Lill Røn og Susanne Stender.
Vejleder: Klaus Grünbaum.

- 103/85 "ØDSLE KOLDKRIGERE OG VIDENSKABENS LYSE IDEER".
Projekt rapport af: Niels Ole Dam og Kurt Jensen.
Vejleder: Bent Sørensen.
- 104/85 "ANALOGREGNEMASKINEN OG LORENZLIIGNINGER".
Af: Jens Jäger.
- 105/85 "THE FREQUENCY DEPENDENCE OF THE SPECIFIC HEAT OF THE GLASS TRANSITION".
Af: Tage Christensen.

"A SIMPLE MODEL OF AC HOPPING CONDUCTIVITY".
Af: Jeppe C. Dyre.
Contributions to the Third International Conference on the Structure of Non - Crystalline Materials held in Grenoble July 1985.
- 106/85 "QUANTUM THEORY OF EXTENDED PARTICLES".
Af: Bent Sørensen.
- 107/85 "EN MYG GØR INGEN EPIDEMI".
- flodblindhed som eksempel på matematisk modellering af et epidemiologisk problem.
Projekt rapport af: Per Hedegård Andersen, Lars Boye, Carsten Holst Jensen, Else Marie Pedersen og Erling Møller Pedersen.
Vejleder: Jesper Larsen.
- 108/85 "APPLICATIONS AND MODELLING IN THE MATHEMATICS CURRICULUM" - state and trends -
Af: Mogens Niss.
- 109/85 "COX I STUDIETIDEN" - Cox's regressionsmodel anvendt på studenteroplysninger fra RUC.
Projekt rapport af: Mikael Wennerberg Johansen, Poul Kattler og Torben J. Andreasen.
Vejleder: Jørgen Larsen.
- 110/85 "PLANNING FOR SECURITY".
Af: Bent Sørensen
- 111/85 "JORDEN RUNDT PÅ FLADE KORT".
Projekt rapport af: Birgit Andresen, Beatriz Quinones og Jimmy Staal.
Vejleder: Mogens Niss.
- 112/85 "VIDENSKABELIGGØRELSE AF DANSK TEKNOLOGISK INNOVATION FREM TIL 1950 - BELYST VED EKSEMPLER".
Projekt rapport af: Erik Odgaard Gade, Hans Hedal, Frank C. Ludvigsen, Annette Post Nielsen og Finn Physant.
Vejleder: Claus Bryld og Bent C. Jørgensen.
- 113/85 "DESUSPENSION OF SPLITTING ELLIPTIC SYMBOLS 11".
Af: Bernhelm Booss og Krzysztof Wojciechowski.
- 114/85 "ANVENDELSE AF GRAFISKE METODER TIL ANALYSE AF KONTIGENSTABELLER".
Projekt rapport af: Lone Billmann, Ole R. Jensen og Arne-Lise von Moos.
Vejleder: Jørgen Larsen.
- 115/85 "MATEMATIKKENS UDVIKLING OP TIL RENESSANCEN".
Af: Mogens Niss.
- 116/85 "A PHENOMENOLOGICAL MODEL FOR THE MEYER-NELDEL RULE".
Af: Jeppe C. Dyre.
- 117/85 "KRAFT & FJERNVARMEOPTIMERING".
Af: Jacob Mørch Pedersen.
Vejleder: Bent Sørensen
- 118/85 "TILFÆLDIGHEDEN OG NØDVENDIGHEDEN IFØLGE PEIRCE OG FYSIKKEN".
Af: Peder Voetmann Christiansen
- 120/86 "ET ANTAL STATISTISKE STANDARDMODELLER".
Af: Jørgen Larsen
- 121/86 "SIMULATION I KONTINUERT TID".
Af: Peder Voetmann Christiansen.
- 122/86 "ON THE MECHANISM OF GLASS IONIC CONDUCTIVITY".
Af: Jeppe C. Dyre.
- 123/86 "GYMNASIEFYSIKKEN OG DEN STORE VERDEN".
Fysiklærerforeningen, IMFUFA, RUC.
- 124/86 "OPGAVESAMLING I MATEMATIK".
Samtlige opgaver stillet i tiden 1974-jan. 1986.
- 125/86 "UVBY, 6 - systemet - en effektiv fotometrisk spektral-klassifikation af B-, A- og F-stjerner".
Projekt rapport af: Birger Lundgren.
- 126/86 "OM UDVIKLINGEN AF DEN SPECIELLE RELATIVITETSTEORI".
Projekt rapport af: Lise Odgaard & Linda Szkoták Jensen
Vejledere: Karin Beyer & Stig Andur Pedersen.
- 127/86 "GALOIS' BIDRAG TIL UDVIKLINGEN AF DEN ABSTRAKTE ALGEBRA".
Projekt rapport af: Pernille Sand, Heine Larsen & Lars Frandsen.
Vejleder: Mogens Niss.
- 128/86 "SMÅKRYB" - om ikke-standard analyse.
Projekt rapport af: Niels Jørgensen & Mikael Klintorp.
Vejleder: Jeppe Dyre.
- 129/86 "PHYSICS IN SOCIETY"
Lecture Notes 1983 (1986)
Af: Bent Sørensen
- 130/86 "Studies in Wind Power"
Af: Bent Sørensen
- 131/86 "FYSIK OG SAMFUND" - Et integreret fysik/historie-projekt om naturanskuelsens historiske udvikling og dens samfundsmæssige betingethed.
Projekt rapport af: Jakob Heckscher, Søren Brønd, Andy Wierød.
Vejledere: Jens Høyrup, Jørgen Vogelius, Jens Højgaard Jensen.
- 132/86 "FYSIK OG DANNELSE"
Projekt rapport af: Søren Brønd, Andy Wierød.
Vejledere: Karin Beyer, Jørgen Vogelius.
- 133/86 "CHERNOBYL ACCIDENT: ASSESSING THE DATA. ENERGY SERIES NO. 15.
AF: Bent Sørensen.
-
- 134/87 "THE D.C. AND THE A.C. ELECTRICAL TRANSPORT IN AsSeTe SYSTEM"
Authors: M.B.El-Den, N.B.Olsen, Ib Høst Pedersen, Petr Visčor
- 135/87 "INTUITIONISTISK MATEMATIKS METODER OG ERKENDELSESTEORETISKE FORUDSÆTNINGER"
MATEMATIKSPECIALE: Claus Larsen
Vejledere: Anton Jensen og Stig Andur Pedersen
- 136/87 "Mystisk og naturlig filosofi: En skitse af kristendommens første og andet møde med græsk filosofi"
Projekt rapport af Frank Colding Ludvigsen
Vejledere: Historie: Ib Thiersen
Fysik: Jens Højgaard Jensen
- 137/87 "HOPMODELLER FOR ELEKTRISK LEDNING I UORDNEDE FASTE STOFFER" - Resume af licentiatafhandling
Af: Jeppe Dyre
Vejledere: Niels Boye Olsen og Peder Voetmann Christiansen.
- 119/86 "DET ER GANSKE VIST - - EUKLIDS FEMTE POSTULAT KUNNE NOK SKABE RØRE I ANDEDAMMEN".
Af: Iben Maj Christiansen
Vejleder: Mogens Niss.

- 138/87 "JOSEPHSON EFFECT AND CIRCLE MAP."
Paper presented at The International Workshop on Teaching Nonlinear Phenomena at Universities and Schools, "Chaos in Education". Balaton, Hungary, 26 April-2 May 1987.
By: Peder Voetmann Christiansen
- 139/87 "Machbarkeit nichtbeherrschbarer Technik durch Fortschritte in der Erkennbarkeit der Natur"
Af: Bernhelm Booss-Bavnbek
Martin Bohle-Carbonell
- 140/87 "ON THE TOPOLOGY OF SPACES OF HOLOMORPHIC MAPS"
By: Jens Gravesen
- 141/87 "RADIOMETERS UDVIKLING AF BLODGASAPPARATUR - ET TEKNOLOGIHISTORISK PROJEKT"
Projektrapport af Finn C. Physant
Vejleder: Ib Thiersen
- 142/87 "The Calderón Projektor for Operators With Splitting Elliptic Symbols"
by: Bernhelm Booss-Bavnbek og
Krzysztof P. Wojciechowski
- 143/87 "Kursusmateriale til Matematik på NAT-BAS"
af: Mogens Brun Heefelt
- 144/87 "Context and Non-Locality - A Peircan Approach
Paper presented at the Symposium on the Foundations of Modern Physics The Copenhagen Interpretation 60 Years after the Como Lecture. Joensuu, Finland, 6 - 8 august 1987.
By: Peder Voetmann Christiansen
- 145/87 "AIMS AND SCOPE OF APPLICATIONS AND MODELLING IN MATHEMATICS CURRICULA"
Manuscript of a plenary lecture delivered at ICMTA 3, Kassel, FRG 8.-11.9.1987
By: Mogens Niss
- 146/87 "BESTEMMELSE AF BULKRESISTIVITETEN I SILICIUM"
- en ny frekvensbaseret målemetode.
Fysikspeciale af Jan Vedde
Vejledere: Niels Boye Olsen & Petr Višćor
- 147/87 "Rapport om BIS på NAT-BAS"
redigeret af: Mogens Brun Heefelt
- 148/87 "Naturvidenskabsundervisning med Samfundsperspektiv"
af: Peter Colding-Jørgensen D.H
Albert Chr. Paulsen
- 149/87 "In-Situ Measurements of the density of amorphous germanium prepared in ultra high vacuum"
by: Petr Višćor
- 150/87 "Structure and the Existence of the first sharp diffraction peak in amorphous germanium prepared in UHV and measured in-situ"
by: Petr Višćor
- 151/87 "DYNAMISK PROGRAMMERING"
Matematikprojekt af:
Birgit Andresen, Keld Nielsen og Jimmy Staal
Vejleder: Mogens Niss
- 152/87 "PSEUDO-DIFFERENTIAL PROJECTIONS AND THE TOPOLOGY OF CERTAIN SPACES OF ELLIPTIC BOUNDARY VALUE PROBLEMS"
by: Bernhelm Booss-Bavnbek
Krzysztof P. Wojciechowski
- 153/88 "HALVLEDERTEKNOLOGIENS UDVIKLING MELLEM MILITÆRE OG CIVILE KREFTER"
Et eksempel på humanistisk teknologihistorie
Historiespeciale
Af: Hans Hedal
Vejleder: Ib Thiersen
- 154/88 "MASTER EQUATION APPROACH TO VISCOUS LIQUIDS AND THE GLASS TRANSITION"
By: Jeppe Dyre
- 155/88 "A NOTE ON THE ACTION OF THE POISSON SOLUTION OPERATOR TO THE DIRICHLET PROBLEM FOR A FORMALLY SELFADJOINT DIFFERENTIAL OPERATOR"
by: Michael Pedersen
- 156/88 "THE RANDOM FREE ENERGY BARRIER MODEL FOR AC CONDUCTION IN DISORDERED SOLIDS"
by: Jeppe C. Dyre
- 157/88 "STABILIZATION OF PARTIAL DIFFERENTIAL EQUATIONS BY FINITE DIMENSIONAL BOUNDARY FEEDBACK CONTROL: A pseudo-differential approach."
by: Michael Pedersen
- 158/88 "UNIFIED FORMALISM FOR EXCESS CURRENT NOISE IN RANDOM WALK MODELS"
by: Jeppe Dyre
- 159/88 "STUDIES IN SOLAR ENERGY"
by: Bent Sørensen
- 160/88 "LOOP GROUPS AND INSTANTONS IN DIMENSION TWO"
by: Jens Gravesen
- 161/88 "PSEUDO-DIFFERENTIAL PERTURBATIONS AND STABILIZATION OF DISTRIBUTED PARAMETER SYSTEMS: Dirichlet feedback control problems"
by: Michael Pedersen
- 162/88 "FISIK & FYSIK - OG MEGET MERE"
AF: Karin Beyer, Susanne Blegaa, Birthe Olsen,
Jette Reich, Mette Vedelsby
- 163/88 "EN MATEMATISK MODEL TIL BESTEMMELSE AF PERMEABILITETEN FOR BLOD-NETHINDE-BARRIEREN"
Af: Finn Langberg, Michael Jarden, Lars Frellesen
Vejleder: Jesper Larsen
- 164/88 "Vurdering af matematisk teknologi
Technology Assessment
Technikfolgenabschätzung"
Af: Bernhelm Booss-Bavnbek, Glen Pate med
Martin Bohle-Carbonell og Jens Højgaard Jensen
- 165/88 "COMPLEX STRUCTURES IN THE NASH-MOSER CATEGORY"
by: Jens Gravesen

166/88 "Grundbegreber i Sandsynligheds-
regningen"

Af: Jørgen Larsen

167a/88 "BASISSTATISTIK 1. Diskrete modeller"

Af: Jørgen Larsen

167b/88 "BASISSTATISTIK 2. Kontinuerte
modeller"

Af: Jørgen Larsen

168/88 "OVERFLADEN AF PLANETEN MARS"

Laboratorie-simulering og MARS-analoger
undersøgt ved Mössbauerspektroskopi.

Fysikspeciale af:

Birger Lundgren

Vejleder: Jens Martin Knudsen

Fys.Lab./HCØ

169/88 "CHARLES S. PEIRCE: MURSTEN OG MØRTEL
TIL EN METAFYSIK."

Fem artikler fra tidsskriftet "The Monist"
1891-93.

Introduktion og oversættelse:

Peder Vostmann Christøansen

170/88 "OPGAVESAMLING I MATEMATIK"

Samtlige opgaver stillet i tiden
1974 - juni 1988

171/88 "The Dirac Equation with Light-Cone Data"

af: Johnny Tom Ottesen