Numerisk løsning af differentialligninger

Begyndelsesværdiproblemet

Vi vil se på løsning af begyndelsesværdiproblemet:

$$\begin{cases} \dfrac{dx(t)}{dt} = x'(t) = f(x,t) \\ x(t_0) = x_0 \end{cases}$$

hvor vi vil foretage numerisk integration i trin af længde dt fra (t_0, x_0) og frem til det tidspunkt, hvor vi er interesseret i løsningsværdien.

Nøjagtigheden afhænger dels at metoden og del af antallet af skridt. Man bør derfor foretage en analyse af metodens konvergens for herudfra at kunne give en vurdering af den fejl, der er begået, men med det anvendte program er dette ikke muligt.

Da det anvendte program (Modellus v. 4.01, http://modellus.fct.unl.pt/) kun integrerer med metoden 4. ordens Runge-Kutta, vil vi primært beskæftige os med denne.

Men først en oversigt over tre forskellige metoder.

Euler

Med udgangspunkt i startpunktet beregnes næste punkt ved at antage, at tangenten i startpunktet er en god approksimation til funktionen:

$x_{n+1} := x_n + f(x_n, t_n) \cdot dt$
$t_{n+1} := t_n + dt$

Husk, at når man løser differentialligningen, er x'(t) = f(x,t).

2. ordens Runge-Kutta

Denne metode kaldes også for midtpunktsmetoden

$k_1 := f(x_n, t_n) \cdot dt$

$k_2 := f(x_n+k_1, t_n+dt) \cdot dt$

$x_{n+1} := x_n + ½ \cdot (k_1 + k_2)$
$t_{n+1} := t_n + dt$

4. ordens Runge-Kutta

$k_1 := f(x_n, t_n) \cdot dt$

$k_2 := f(x_n+½ \cdot k_1, t_n+½ \cdot dt) \cdot dt$

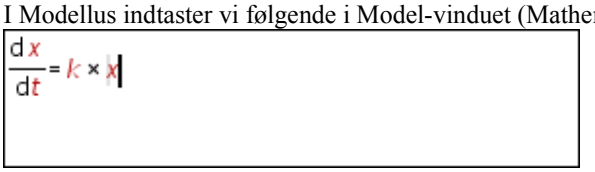
$k_3 := f(x_n+½ \cdot k_2, t_n+½ \cdot dt) \cdot dt$

$k_4 := f(x_n+k_3, t_n+dt) \cdot dt$
$x_{n+1} := x_n + (k_1 + 2k_2 + 2k_3 + k_4)/6$
$t_{n+1} := t_n + dt$

Eksempel

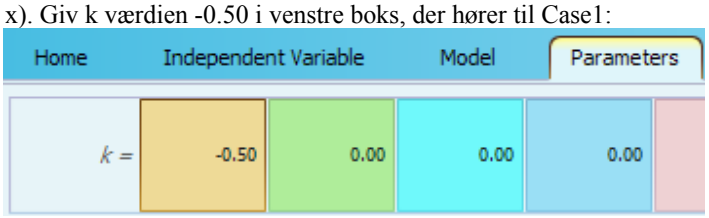
Vi vil nu løse ligningen x'(t) = -0,5·x(t), hvor x(0) = 1, idet vi her kender løsningen, der er x(t) = $1 \cdot e^{-0,5 \cdot t}$

I Modellus indtaster vi følgende i Model-vinduet (Mathematical Model):

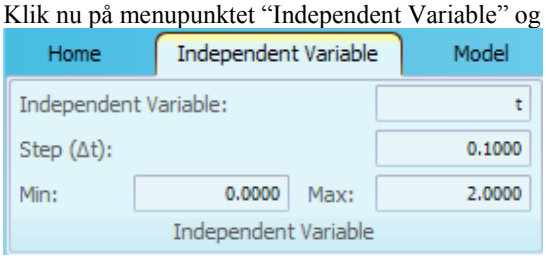


Brug knappen "Rate of change", når differentialkvotienten skal indtastes.

Klik herefter på Interpret, der tjekker for fejl og opretter tabeller over parametre (her k) og begyndelsesbetingelser (her x). Giv k værdien -0.50 i venstre boks, der hører til Case1:



Klik nu på menupunktet "Independent Variable" og sæt følgende for t:



Klik på "Initial Conditions" og sæt startværdien for x i venstre boks (Case1)



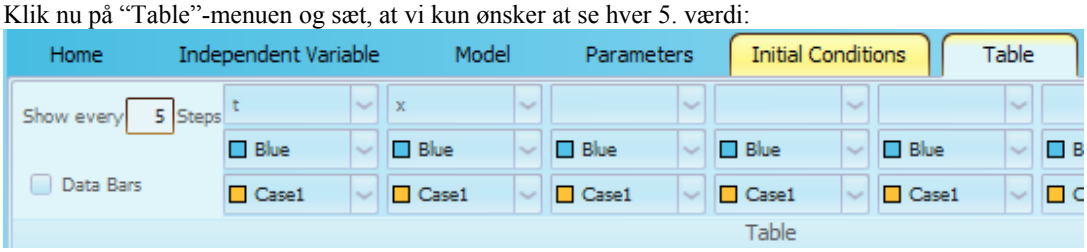
Klik på kør i nederste venstre hjørne og se graf og tabel blive udfyldt. Vi ser, at skaleringen af grafen og antallet af betydende cifre i tabellen ikke er i orden så de rettes nu.

Vælg Graph i menuen (eller klik pr graph-vinduet) og sæt flueben ved "Auto-scale". Herefter kan grafen tilrettes ved at trække i akserne eller flytte koordinatsystemet med musen.

Antallet af decimaler rettes i "Home"-meuen. Sæt det til 10, så vi kan se nøjagtigheden:



Klik nu på "Table"-menuen og sæt, at vi kun ønsker at se hver 5. værdi:



Tabellen ser nu sådan ud (venstre side) og til højre vises den korrekte taget fra TI-Interactive:

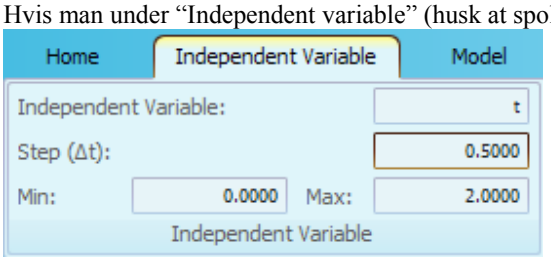
Table

t	x
0.0000000000	1.0000000000
0.5000000000	0.7788007936
1.0000000000	0.6065306762
1.5000000000	0.4723665720
2.0000000000	0.3678794611

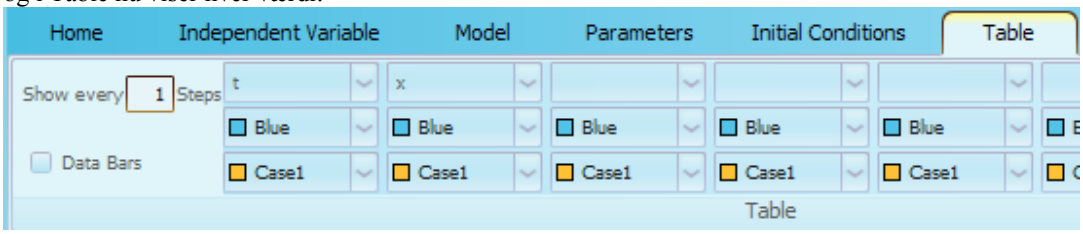
x, t	x
0	1
0.5	0.7788007831
1	0.6065306597
1.5	0.4723665527
2	0.3678794412

Så vi ser altså, at i dette tilfælde er løsningen korrekt på de første 7 cifre.

Hvis man under "Independent variable" (husk at spole tilbage med , da man ellers ikke kan rette) retter til:



og i Table nu viser hver værdi:



får vi i stedet følgende:

Table

t	x
0.0000000000	1.0000000000
0.5000000000	0.7788085938
1.0000000000	0.6065428257
1.5000000000	0.4723807651
2.0000000000	0.3678941994

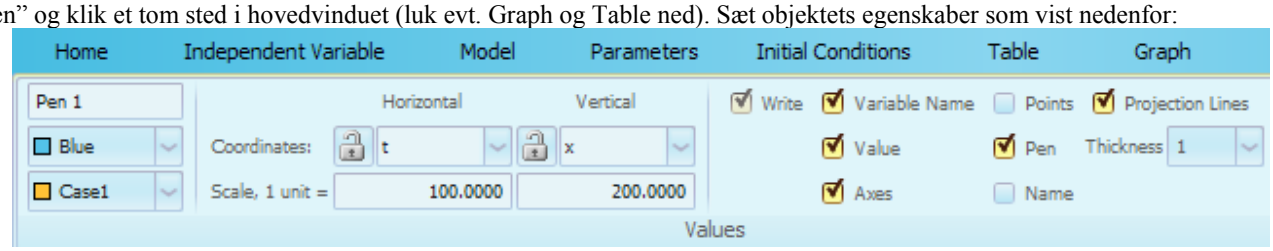
Det ses, at vi nu kun har 4 korrekte cifre.

At vi benytter en 4. ordens metode skal forstås sådan, at fordobler vi skridtlængden vil vi som udgangspunkt få en fejlgrænse, der er 2^4 gange større. Da vi i ovenstående hat øget skridtlængden med en faktor 5, øges fejlgrænsen derfor med en faktor $5^4 = 625$, svarende til ca. 3 cifre, hvilket er helt i overensstemmelse med eksemplet ovenfor.

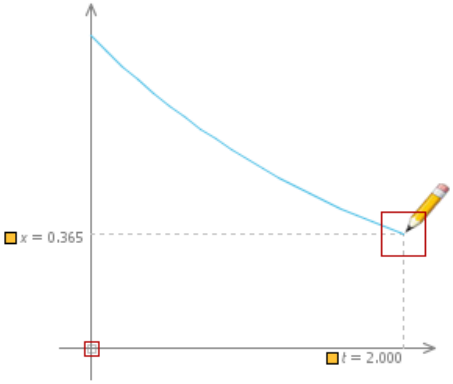
I dette eksempel kender vi løsningen, og dermed fejlen, men man benytter netop numeriske metoder til ligninger, hvor løsningen er ukendt, så fejlen skal naturligvis kunne vurderes særskilt. Heldigvis er dette muligt, se f.eks. Edmund Christiansen: Numerisk Analyse, Odense Universitets Trykkeri, 1989, men vi vil ikke gå ind i dette og blot håbe, at vore løsninger ikke er behæftet med for store fejl.

Objekter i Modellus

Sæt først antallet af decimaler tilbage til 3 og steplængden til 0.1 og klik derefter på "Objects"-menupunktet. Vælg "Pen" og klik et tom sted i hovedvinduet (luk evt. Graph og Table ned). Sæt objektets egenskaber som vist nedenfor:



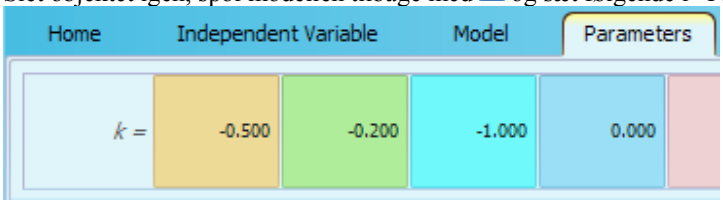
Scale angiver hvor mange pixels på skærmen, der bruges til en enhed. Kør evt. modellen igen. Grafen ser nu sådan ud:



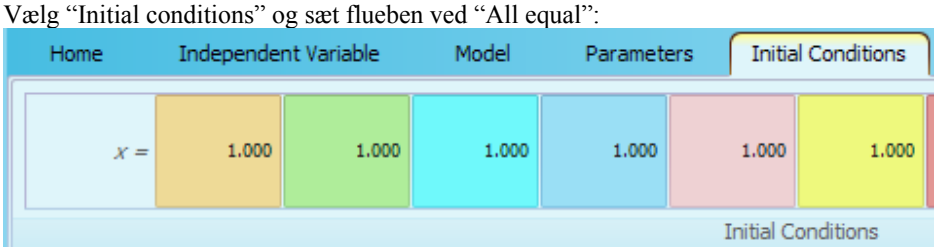
På denne måde kan man vælge forskellig præsentation af løsningen ved at benytte de forskellige objekttyper.

Cases i Modellus

Slet objektet igen, spol modellen tilbage med og sæt følgende i "Parameters":



Vælg "Initial conditions" og sæt flueben ved "All equal":



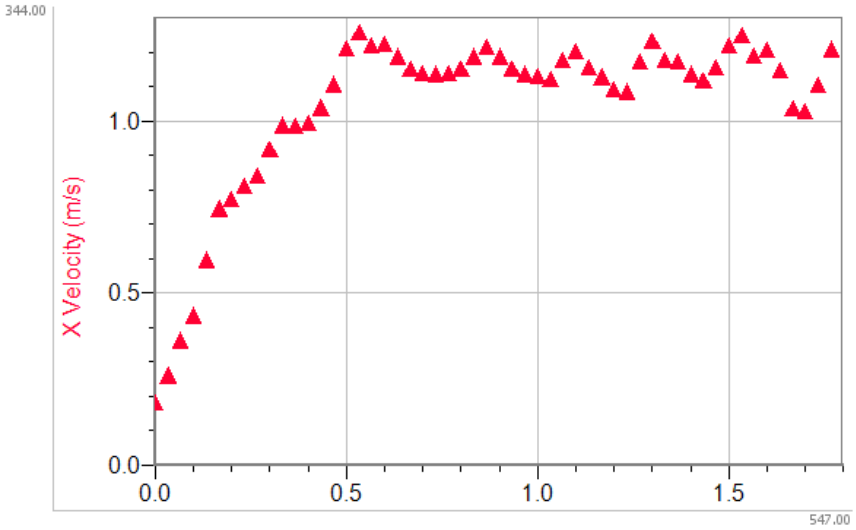
Sæt de tre Cases op i Graph-menupunktet:



Kør modellen og se alle tre løsninger. Tilsvarende teknikker kan anvendes for tabeller og objekter.

Tilpasning til data

Da man ikke kan importere måledata som en tabel vil vi i stedet importere dem som et "Image"-objekt og lægge et "Pen"-objekt ovenpå. Eksemplet nedenfor tager udgangspunkt i forsøget med kageformene, hvor (t, v)-grafen er udmålt med LoggerPro og ultralydsafstandsmåleren:

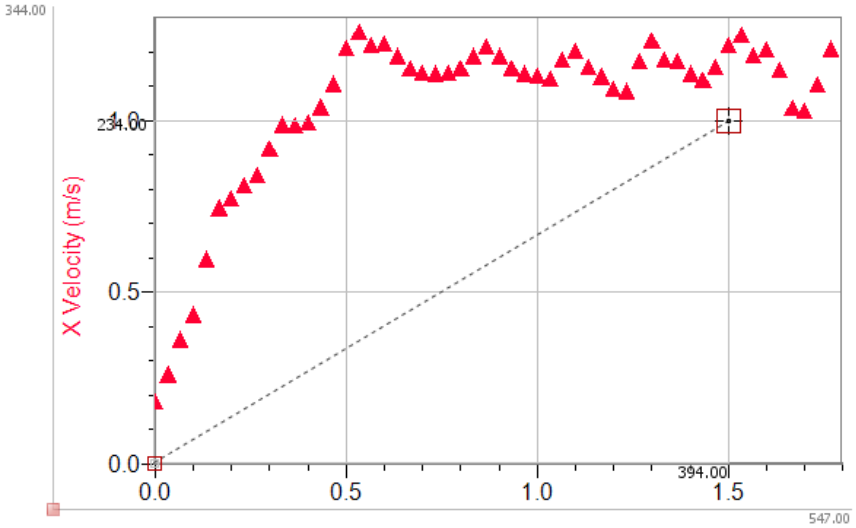


Ifølge opg. M27 kan v(t) skrives som:

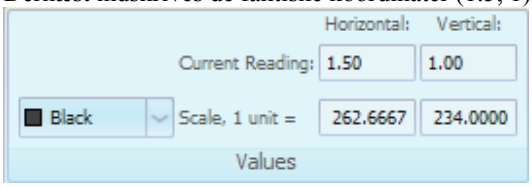
$$v(t) = v_0 \cdot \frac{e^{b \cdot t} - 1}{e^{b \cdot t} + 1}$$

hvor v_0 er den maksimale hastighed og $b = 2 \cdot g / v_0$, så den funktion vil vi tilpasse til ovenstående graf. Først importeres billedet som et Image-objekt. Bemærk, at der er nogle detaljer som skal være på plads, før man har billedet fra LoggerPro, da LoggerPro som standard grafer på udklipsholderen som vektor-objekter og de kan ikke indsættes. De skal først konverteres til bitmap-filer, hvilket f.eks. kan ske i IrfanView. Tryk "PrintScreen", åben IrfanView og sæt ind. Beskær billedet og gem som grafik-fil i png-format.

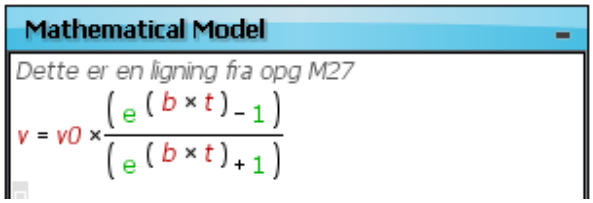
Når billedet er sat ind skal vi finde en oversættelse mellem pixels og de koordinater, som faktisk anvendes. Dette gøres ved at anbringe et "Measure coordinates"-objekt ovenpå grafen og tilpasse skaleringen:



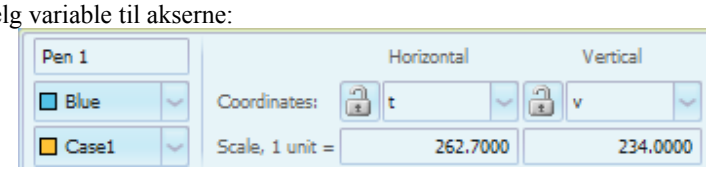
Dernæst indskrives de faktiske koordinater (1.5, 1) og Scale (262.7, 234.0) noteres til senere brug:



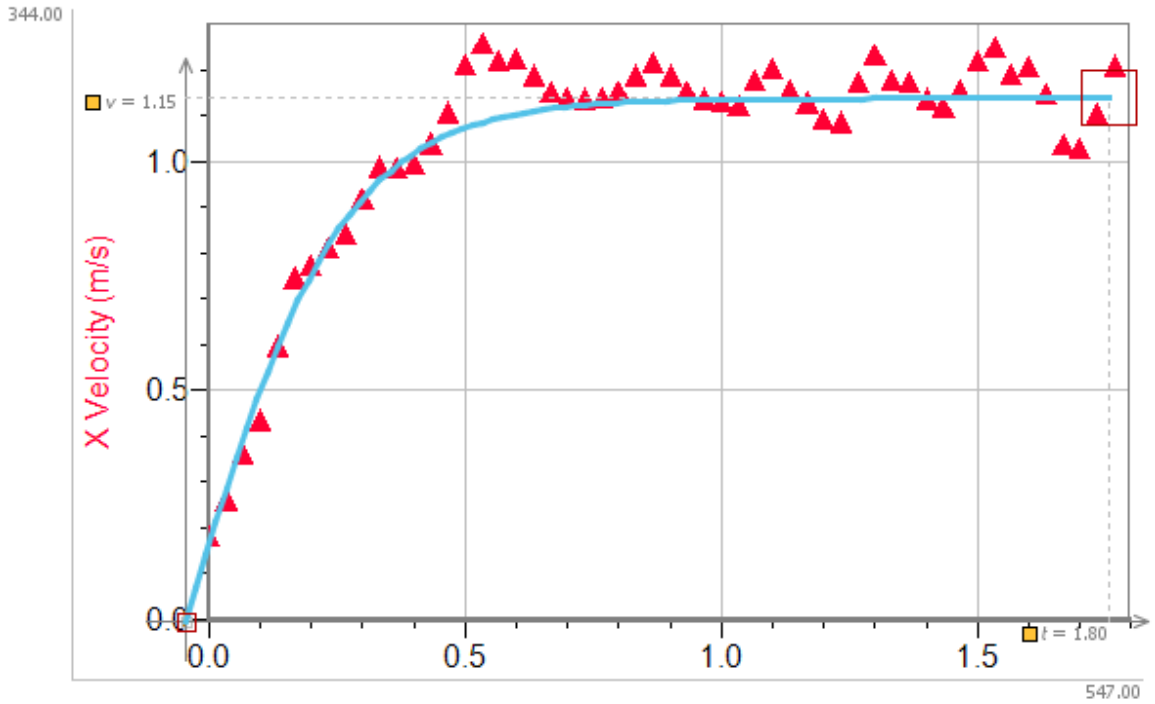
Indskriv nu den matematiske model:



Slet "Measure coordinates"-objektet og indsæt i stedet et "Pen"-objekt. Sæt Scale i dette objekt til det noterede og vælg variable til akserne:



Sæt i "Parameters" v0 til 1.15 og b til 17.4 og kør modellen. Ret på b og ryk med "Pen"-koordinatsystemet indtil der er tidslig overensstemmelse:



Bevægelsesligninger

Det er vigtigt at gøre sig klart, at variable kun kan skrives med cifre, "_" og bogstaver; de skal starte med et bogstav. Der skelnes mellem store og små bogstaver!

Da der ikke regnes med enheder, er det jeres ansvar, at alle størrelser indgår i de korrekte enheder. Det er sikrest at anvende SI-enheder overalt.

Indfør som i eksemplerne konstanter for alle størrelser, som du er vant til i fysik. Det gør det nemmere at overskue, om du har tænkt rigtigt.

Kommentarer skrives på særskilte linjer ved at skrive et ";" og derefter kommentaren.

1 dimensional bevægelse

Hent *bevægelse i en dimension.modellus*, hvor du kan se, at

Fres:= -k*x-c*vx+Fo*sin(w*t)

hvilket er summen af F_{elast}, F_{frik} og F_{driv}}

F_{elast} er fjederkraften, F_{frik} er en friktionskraft, der er proportional med farten (bemærk fortegn) og F_{driv} er en udefrakommende påvirkning af fjederen (en såkaldt tvangskraft).

Undersøg modellen for forskellige værdier af parametrene og steplængde.

Andre kræfter, som skal bruges sammen med kageformene:

Konstant kraft: $F_{res} = m*a$

Friktionskraft: $F_{frik} = -½*c_w *\rho_{luft}*A *vx*abs(vx)$, $c_w \approx 0.25 \ldots 1.00$, A er frontarealet.

2 dimensional bevægelse

Hent *bevægelse i to dimensioner.modellus*, hvor du kan undersøge de skrå kast med $\mathbf{F_{res}} = (0,-g*m)$.

Andre kræfter:

Friktion: $F_x = -½*c_w *\rho_{luft}*A *vx*v$, $F_y = -½*c_w *\rho_{luft}*A *vy*v$, $v = \sqrt{vx^2 + vy^2}$

Gravitation: $F_x = -G*M*m*x/r^3$, $F_y = -G*M*m*y/r^3$