

Modellering och Geometri

Thomas Ericsson

Beräkningsmatematik
Chalmers och Göteborgs universitet

2013

2: Innehåll

- Thomas Ericsson: lite kursinformation (denna sida).
- Thomas Lingefjärd: 15 min. om geometridelen (andra halvan av kursen).
- Thomas E, resten av fm och em.
 - Kort om mig och modelleringsdelen.
 - Lite om Geogebra och numeriska beräkningar.
 - Mer om Geogebra, bland annat hur man löser ODE-problem.
 - En längre modelleringsexempel om vi får tid.
 - Laboration med Geogebra.

Lunch 11:45-13:15.

Förminskade versioner av dessa sidor finns under hemsidan (**forel_n.pdf**), **n = 2, 4, 8**. Där finns också Geogebra-laborationen för em. Filerna ligger också under den öppna sidan **<http://www.math.chalmers.se/Math/Grundutb/GU/LLMA60/H13>**

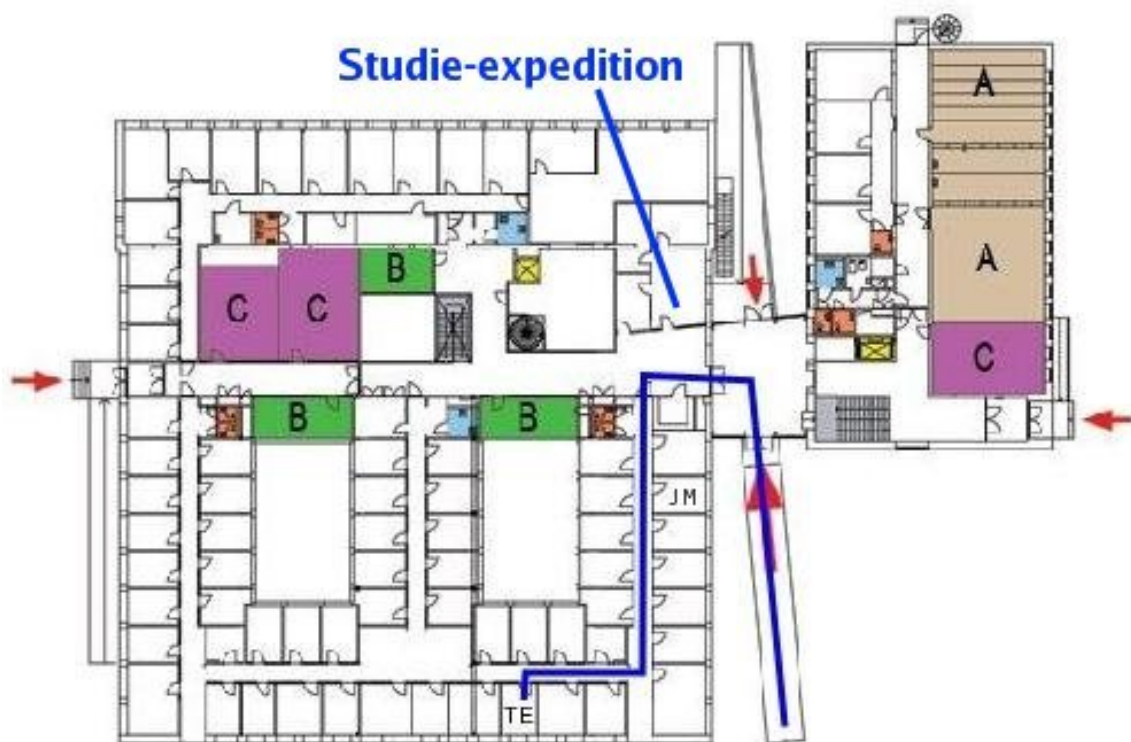
Thomas Ericsson
Beräkningsmatematik
Chalmers/GU

Telefon: 031-772 1091

E-post: **thomas@chalmers.se**

Använd e-post för kommunikation, jag lär glömma att titta i GUL.

Rum: L2075, Matematiska vetenskaper, Chalmers tvärgata 3



TE = Thomas Ericsson

1: Andra kurser och forskning

- Programmering med Matlab (och lite C++) för matematikprogrammet, GU.
- Numerisk analys för GU. Hur man löser (några) tillämpningsproblem med hjälp av dator. Algoritmer, datoraritmetik, effektivitet etc.
- Vetenskaplig visualisering. Hur man använder datorgrafik för att visualisera beräkningsresultat. Innehåller bland annat Matlab, OpenGL, ParaView. Masterskurs.
- HPC, High Performance Computing. Kodoptimering, parallella beräkningar. C & Fortran. Masterskurs.
- National Graduate School in Scientific Computing (NGSSC) (några föreläsningar). Paralleldatorcentrum vid KTH i augusti.
- Applikationsexpert (HPC-råd till forskare).

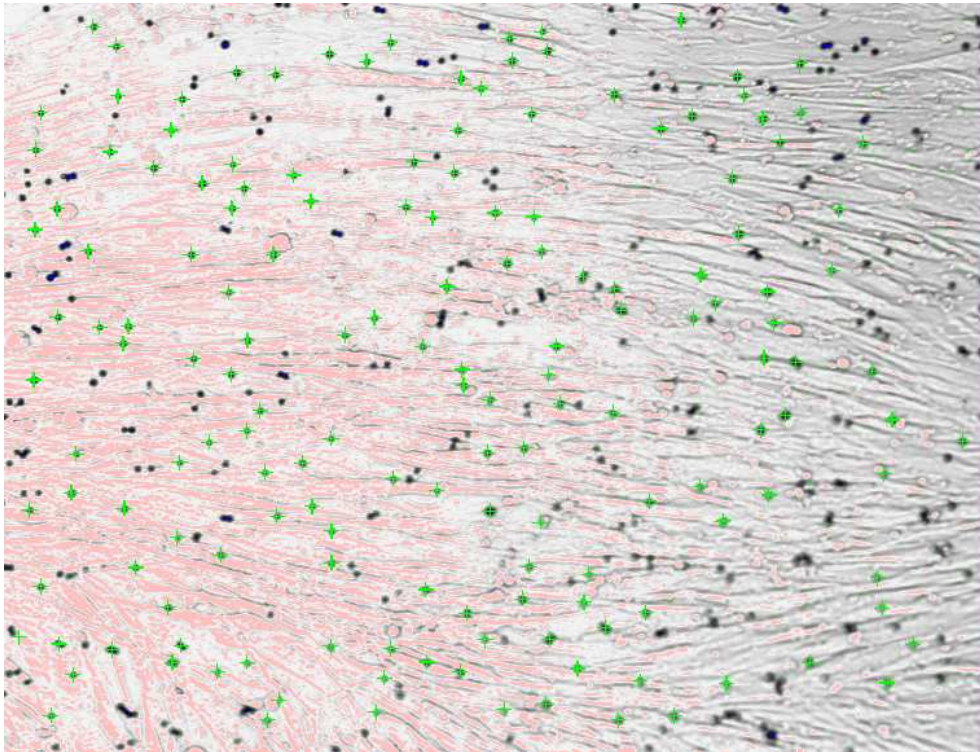
2: Andra kurser och forskning

Forskning: medicinsk fysik och lite kvantfysik.

Ett exempel: hur mekaniska egenskaper hos mänskliga lungceller påverkas av nanopartiklar.

Biomechanical effects of environmental and engineered particles on human airway smooth muscle cells

3: Andra kurser och forskning

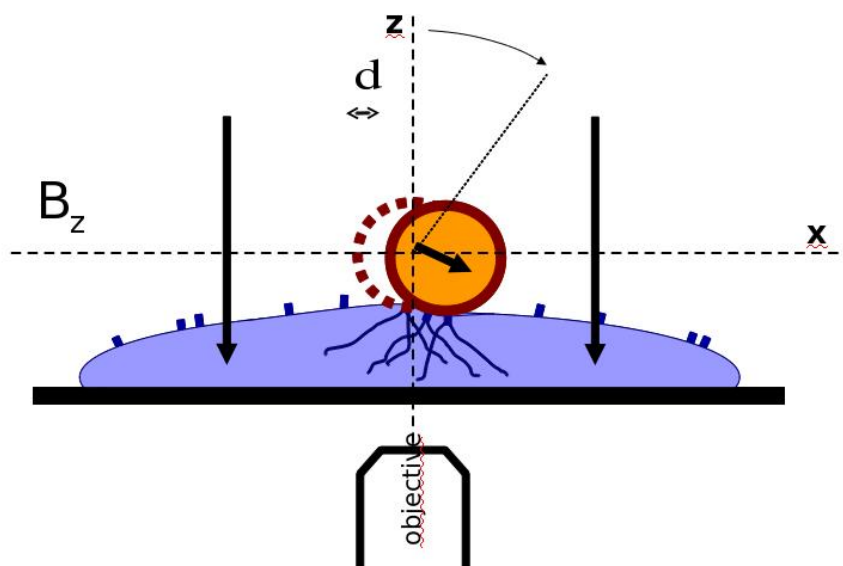


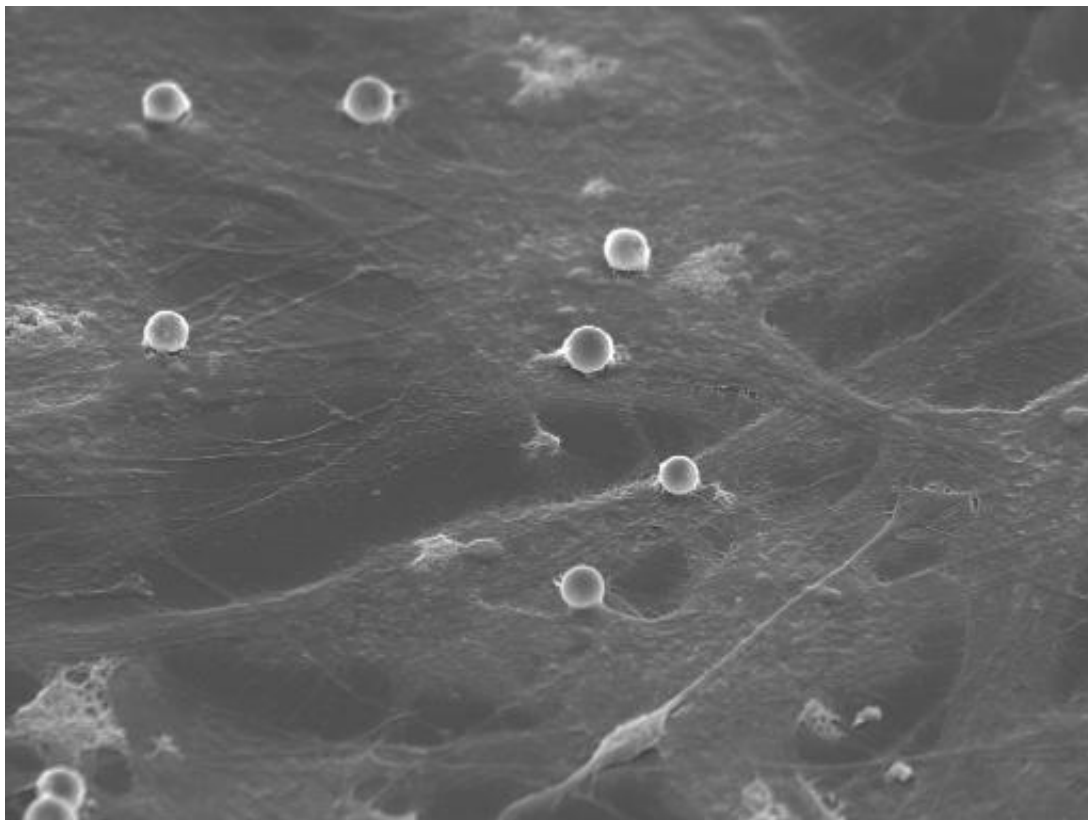
Kulorna har diametern **$4.5\mu\text{m}$** .

4: Andra kurser och forskning

Optical Magnetic Twisting Cytometry (OMTC)

OMTC: micro-rheometer where the adherent living cell is sheared between the substrate on which it is attached and a magnetic microbead ($5\mu\text{m}$)





Min insats utgörs i stor utsträckning av behandling av data och parameteranpassning. Vi har skapat en förenklad beskrivning, en modell, av hur cellerna uppför sig. Modellen är formulerad med hjälp av matematik, bland annat differentialekvationer. I modellen finns parametrar som man vill bestämma så att modellen passar ihop med data så bra som möjligt. Kan användas för att förstå vad som händer med cellerna.

Kurslitteratur:

- Belinda Barnes, Glenn Robert Fulford (2008) (2nd revised edition) Mathematical Modelling with case studies. Chapman & Hall/CRC. ISBN13: 9781420083484.
- Lingefjärd, Thomas, Jönsson, Per. (2012). IKT i grund- och gymnasieskolans matematikundervisning. Lund: Studentlitteratur. ISBN13: 9789144056777.

Referenslitteratur:

Philip Gerlee, Torbjörn Lundh (2012). Vetenskapliga modeller. Svarta lådor, röda atomer och vita lögner. Lund, Studentlitteratur. ISBN13: 9789144074207.

Om Mathematical Modelling:

- Lämplig för självstudier (detta är min enda föreläsning).
- Fokus på modeller och analys av resultat.
- Många intressanta exempel, borde gå att använda på gymnasienivå.
- Inriktning mot ekologi, hållbar utveckling.
- Målgruppen är nog biologer som vill lära sig modellering.
- Inte definition-sats-bevis-stilen.

Programvara: Geogebra, <http://www.geogebra.org>

Varför Geogebra?

- Enkelt att använda, ingen programmering.
- Lätt att variera begynnelsevärden och annat.
- Finns för Linux, Windows och Mac.
- Gratis. Borde gå bra att använda i skolan.

Några nackdelar (för kursen):

- Kan ej plotta i 3D.
- Begränsat vad gäller ODE. Kan till nöds klara två ekvationer (bara autonoma system dessutom).

Flertalet övningar kräver att man löser system av differential-ekvationer. För att kunna göra detta på ett smidigt sätt kommer vi att använda en beta-version av Geogebra 5.

http:

[//code.google.com/p/geogebra/downloads/list](http://code.google.com/p/geogebra/downloads/list)

Jag har testat version 4-9-189-0 (jag hämtade exe-filen). Denna version är en utökad variant av Geogebra 4 (som inte klarar av system utan en massa krångel). Jag kommer att visa hur man använder beta-versionen för att lösa system och för att rita fasporträtt. Eftersom beta-versionen är under utveckling så innehåller den dessvärre en del buggar. Det går bra att ha flera versioner av Geogebra installerade samtidigt, så du behöver inte ta bort den tidigare versionen (om du har någon).

- Eftermiddagens laboration (Geogebra + ODE).
- Några övningar ur läroboken, Mathematical Modelling.
Se någon av hemsidorna.
Är deadlines på laborationerna.
- Tentamen med dator, 2 timmar, med två problem liknande laborationerna.
- Betyget på hela kursen är en funktion av resultaten på både modellerings- och geometridelarna. Du måste dock ha minst godkänt på båda delarna för att få godkänt på hela kursen.

12: Kort om numeriska beräkningar

Modelleringsdelen skiljer sig rätt mycket från kurser i ren matematik. Problemen liknar mer enkla tillämpningsproblem från näringsliv och forskning.

Problemen i en ren matematik-kurs:

- Är klart formulerade. Beräkna volymen, gränsvärdet, ...
- Kan lösas för hand.
- Kan lösas exakt. Arean är $\pi\sqrt{2}$ a.e.
- Det finns ett facit.

Inget av detta behöver gälla för (realistiska) beräkningsproblem.

13: Kort om numeriska beräkningar

I numeriska beräkningar används sk flyttal (floating point numbers). I datorns CPU finns hårdvara (FPU, Floating Point Unit) som kan utföra beräkningar ($+$, $-$, $*$, $\sqrt{}$) med 64-bitars flyttal. Tal med ungefär 16 decimaler och i intervallet $\approx [-10^{308}, 10^{308}]$.

$(1 / 49) * 49 - 1$ blir $-1.1 \cdot 10^{-16}$ i Geogebra och i Matlab, Fortran, C/C++ etc.

$\text{sqrt}(2)^2 - 2$ blir $\approx 4.4 \cdot 10^{-16}$.
 $(1 + 1\text{E-}16) - 1$ blir exakt noll.

Avrundningsfel. Normalt dominerar andra approximationsfel med flera storleksordningar.

14: Kort om numeriska beräkningar

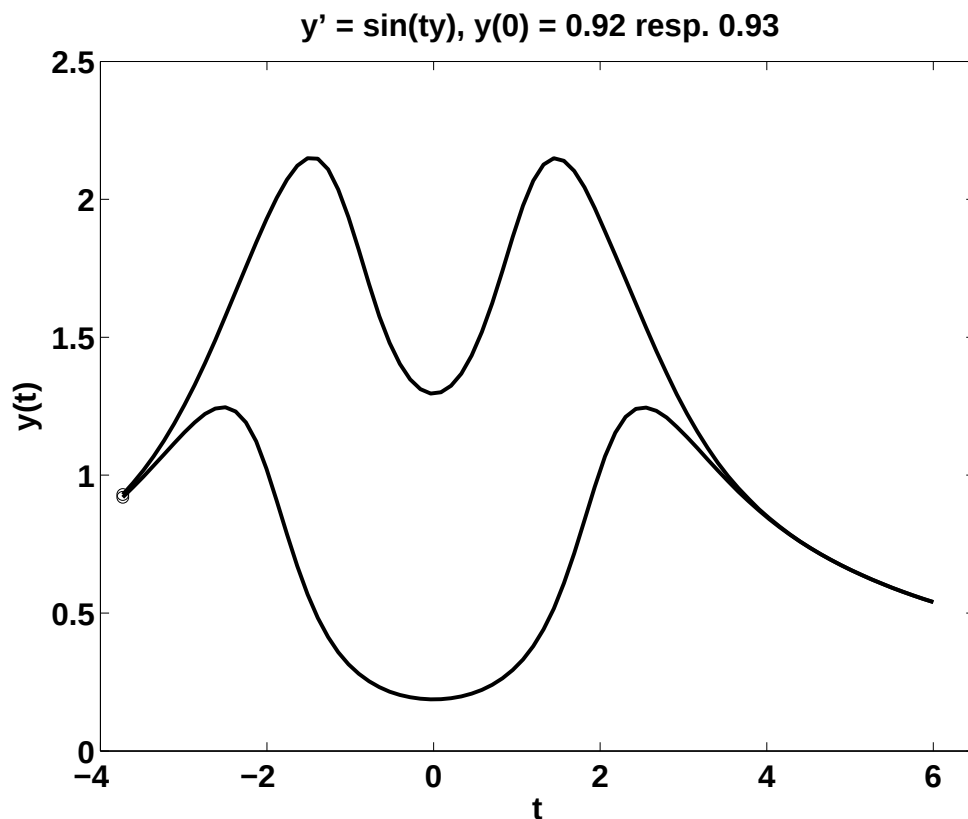
Inte så många satser och bevis i boken. Det finns en hel del matematik inom modellering, men allt är inte så praktiskt användbart. En typ av intressanta satser är störningsresultat, stabilitet.

Antag att y är lösningen till $y' = f(t, y), y(0) = a$.

Vi ändrar nu begynnelsevärdet med ϵ och får lösningen z , så $z' = f(t, z), y(0) = a + \epsilon$.

Vad är $|y(t) - z(t)|$?

Finns enbart lokala satser (för allmänna problem).



Följande gäller Geogebra version 4-9-189-0 under Windows 7 men Linux-varianten är också OK. Jag har inte testat Mac-versionen.

Observera att **NSolveODE** inte finns under version 4-2-??-? och ordningen mellan parametrar i andra ode-kommandon kan vara ändrad jämför med beta-versionen.

Geogebra har stöd för symbolisk beräkning, CAS (Computer Algebra System).

Vi kan då kanske få analytiska lösningar. Inte alla problem kan dock lösas exakt. Måste välja CAS från Visa-menyn.

Skriver man: **LösODE**[**y' = x + y**] så får man

$$y = c_1 e^x - x - 1$$

Vi vill kanske lägga på ett begynnelsevillkor, dvs. vi kräver att $y(x_0) = y_0$ för givna värden på x_0 och y_0 . T.ex.

LösODE[**y' = x + y**, 2, 0] så $y(0) = 2$.

$$y = -x + 3e^x - 1$$

Skriver man **LösODE**[**y' = exp(-x^2)**] får man

$$y = c_1 + \frac{\sqrt{\pi} \operatorname{erf}(x)}{2}$$

Anledningen är att vi försöker beräkna

$$\int e^{-x^2} dx$$

men vi vet att de primitiva funktionerna inte kan uttryckas i elementära funktioner. **erf** står för "error function".

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

Ibland får man dock nonsens-svar.

Flertalet tillämpningsproblem kan dock inte lösas exakt utan vi får en numerisk approximation. Säg att vi vill lösa begynnelsevärdesproblemet:

$$y'(x) = f(x, y), \quad x_0 < x \leq x_{\text{end}}, \quad y(x_0) = y_0$$

Vänsterledet får bara innehålla förstaderivatatan och högerledet $f(x, y)$ får inte innehålla någon derivata. $y(x_0) = y_0$ är ett begynnelsevillkor där x_0 (ofta noll) och y_0 måste vara kända. Vi kommer att få approximationer av lösningen i ekvidistanta x -värden $x_0, x_0 + \Delta x, x_0 + 2\Delta x, x_0 + 3\Delta x, \dots$

Så här står det i [www-hjälpen](#) (den som hoppar i upp i Geogebra-programmet är delvis felaktig, eller konstig). Man kan skriva **LösODE** i stället för **SolveODE** om man vill.

SolveODE[<f(x,y)>, <Start x>, <Start y>, <End x>, <Step>]

Så, <f(x,y)> är högerledet, <Start x> är x_0 , <Start y> är y_0 , <End x> är x_{end} och <Step> är Δx .

För att lösa problemet $y'(x) = 2x - y(x), y(0) = 2, x_{\text{end}} = 5$ skriver vi t.ex:

LösODE[2 x - y, 0, 2, 5, 0.1]

man kan skriva **2 x - y**, **2x - y**, **2*x - y**, **x 2 - y** men **inte x2 - y** (ty det refererar till variabeln **x2**). Blanktecknen kring **-** är ej nödvändiga att sätta ut.

Ett för stort värde på <Step> ger hörn på kurvan. Använd dock inte mindre <Step> än nödvändigt.

Ett alternativ är att definiera en funktion, för högerledet, först:

f(x, y) = 2 x - y
LösODE[f, 0, 2, 5, 0.1] **eller**
LösODE[f(x, y), 0, 2, 5, 0.1]

21: Några lurigheter (buggar)

$y'(x) = -1, y(0) = 1$ har lösningen $y(x) = 1 - x$.

`LösODE[-1, 0, 1, 2, 0.1]`

ger felmeddelandet "Otillåtet argument: Tal -1".

Man kan dock lura Geogebra, följande går bra:

`LösODE[0 x - 1, 0, 1, 2, 0.1]`

eller

`f(x) = -1` ritas dock ut

`LösODE[f, 0, 1, 2, 0.1]`

`f(x, y)` nedan ritas inte ut (om inte Ritområde 3D är öppet).

`f(x, y) = -1`

`LösODE[f, 0, 1, 2, 0.1]`

22: Några lurigheter (buggar)

Små/stora tal brukar man skriva med tio-potenser: `-1.45e-32` svarar **normalt** mot $-1.45 \cdot 10^{-32}$, och `1.45e87` svarar **normalt** mot $1.45 \cdot 10^{87}$.

Skriver man `-1.45e-32` i Geogebra får man svaret `-35.94` (med två decimaler) ty det varar mot $-1.45 \cdot 2.7182818 \dots - 32$ (så länge som inte **e** har omdefinierats). **pi** är också fördefinierat, men det går inte att definiera om, **pi = 3** går inte.

För att få tio-potens skriv `-1.45E-32` respektive `1.45E87`. I första fallet får man dessutom sätta om antalet gällande siffror för att inte bara få värdet noll (menyn, Inställningar, alternativet Antal decimaler). Analogt för det andra talet, annars får man se alla nollorna.

Man kan åka dit på detta även om man har så pass stora tal som `0.001` (skrivs som `0`).

Nu till system av ODE (enbart två ekvationer). Följande fungerar bara i de senaste Geogebra-versionerna. Måste finnas stöd för kommandot **NSolveODE**.

NLösODE fungerar inte (2013-08-16 i alla fall).

Låt oss lösa lärobokens exempel (predator-prey) sid 108-110. Eftersom Geogebra inte tillåter godtyckliga variabelnamn utan kräver att de sökta funktionerna benämns $y(x)$, $z(x)$ blir systemet:

$$\begin{cases} y' = y - 0.01yz \\ z' = -0.5z + 0.005yz \end{cases}, \quad \begin{matrix} y(0) = 200 \\ z(0) = 80 \end{matrix}, \quad x_{end} = 20$$

För att allt skall fungera smidigt i Geogebra föreslår jag följande arbetssätt (dock inte alltid nödvändigt att införa variabler och funktioner).

```
y0 = 200
z0 = 80
f(x, y, z) = y - 0.01 y z
g(x, y, z) = -0.5 z + 0.005 y z
ode = NSolveODE[{f, g}, 0, {y0, z0}, 20]
```

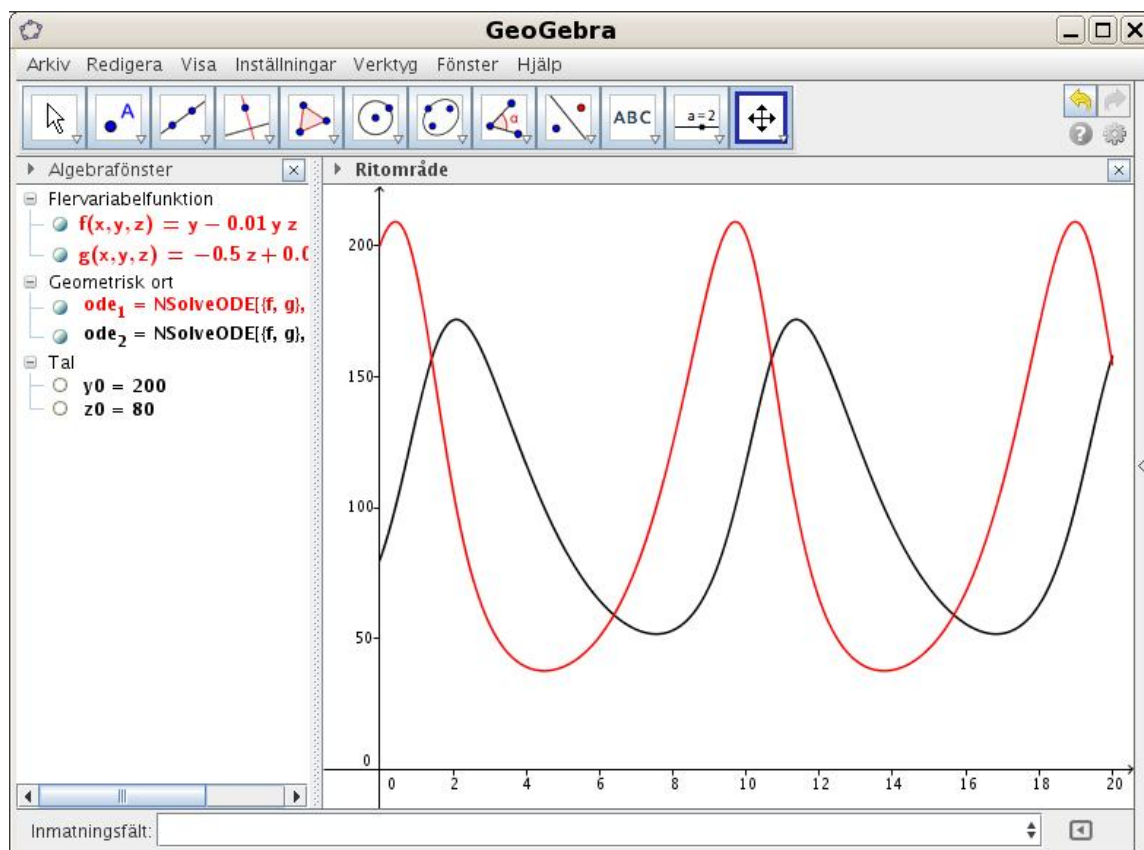
Vill man ändra sluttiden är det bra med en variabel för den också. Vi kommer att referera till **ode** senare.

För att se något i plotfönstret (Ritområde) måste man skala om x-axeln (eftersom $y(0) \gg x_{end}$). Vänster-shift-klicka (tryck på shift, klicka och håll ned) på x-axeln och drag med musen.

Skala om y-axeln analogt.

Så här kan det se, där jag dessutom ändrar färg på $y(x)$.

25: System av ODE



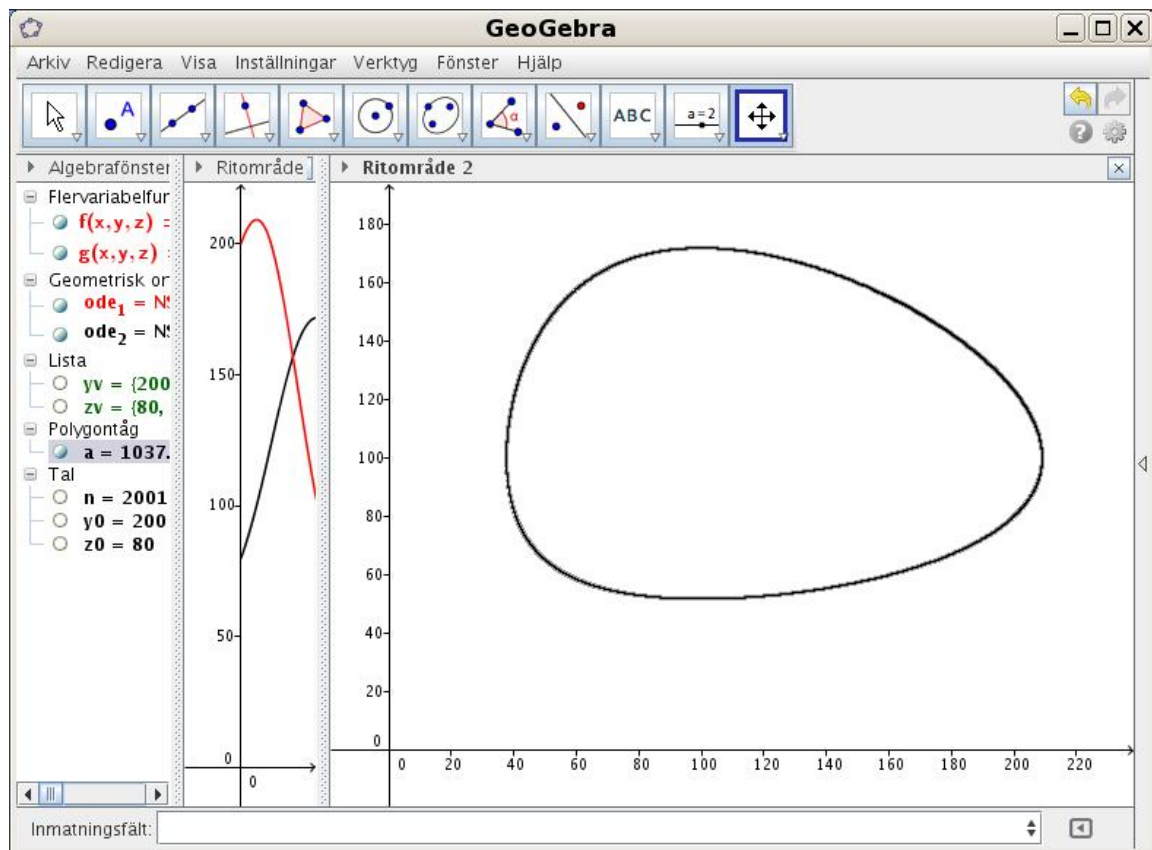
26: System av ODE

Kapitel 6 ägnas åt analys i fas-planet. Man vill då plotta z som funktion av y (x elimineras).

Jag har inte hittat något färdigt för detta, men så här kan man göra (förutsätter att vi har definierat **ode** enligt ovan). Man aktiverar lämpligen Ritområde 2 först. Det underlättar om man sparar ett exempel som en mall (en **ggb**-fil) som man sedan kan modifiera. Windows-versionen av Geogebra är buggig, så där får man krångla lite mer än under Linux (jag har brutit sista raden):

```
n = Längd[ode_1]
yv = y(Första[ode_1, n])
zv = y(Första[ode_2, n])
Polygontåg[Talföljd[(Element[yv, i],
                      Element[zv, i]), i, 1, n]]
```

Här en komprimerad bild:



28: Ett modelleringsexempel

Fysikproblem: vi släpper en metallkula från ett (mycket) högt torn med höjden h . Vilken fart har kulan när den slår i marken?

Modell: kulan har massan m (kg) och utgångsfarten 0 (m/s). Låt $s(t)$ vara positionen i höjddled med $s(0) = h$, höjden.

Accelerationen är $s''(t)$ så Newtons andra lag, $F = ma$, ger $F = ms''$. Vilka krafter påverkar kulan?

Gravitationen: $-mg$, $g \approx 9.82 \text{ m/s}^2$.

Luftmotståndet: $F_{\text{luft}}(t)$ riktad uppåt.

Inför $v = s'$. Vi har då differentialekvationen

$$mv' = -mg + F_{\text{luft}}, \quad v(0) = 0$$

Är tornet inte så högt kan vi kanske bortse från luftmotståndet, $F_{\text{luft}}(t) = 0$ vilket ger:

$$mv' = -mg, \quad v(0) = 0 \Rightarrow v(t) = -gt$$

Vi använder följande enkla modell för luftmotståndet:

$$F_{luft} = \frac{c_d A \rho_L v^2}{2}$$

A är kulans tvärsnittsarea. ρ_L är luftens densitet (antas konstant 1.2 kg/m^3 , 20° C). c_d "drag coefficient", luftmotståndskoefficient, 0.47 för en sfär, 0.04 för en strömlinjeformad kropp, 1 för cyklist med cykel (touring position).

Enkel, slarvig motivering: under tiden Δt accelereras luftmassan med volymen $A v \Delta t$ och massan $\rho_L A v \Delta t$ från stillastående till farten v (föresatt att ingen luft läcker ut på sidan).

Luftpaketet har därmed fått den kinetiska energin

$E_{kin} = (\rho_L A v \Delta t) v^2 / 2$. Arbete = kraft $\cdot$ väg, så

$F_{luft} = E_{kin} / (v \Delta t) = \rho_L A v^2 / 2$.

c_d justerar för läckaget.

30: ODE-problemet

Vårt ODE-problem blir därför:

$$m v' = -mg + \frac{c_d A \rho_L v^2}{2}, \quad v(0) = 0$$

eller (efter division med m)

$$v' = -g + c v^2, \quad \text{med } c = \frac{c_d A \rho_L}{2m}, \quad v(0) = 0$$

Låt kulan ha radie r och densitet ρ_K , då kan c skrivas

$$c = \frac{c_d \pi r^2 \rho_L}{2 \rho_K \frac{4}{3} \pi r^3} = \frac{3 c_d \rho_L}{8 r \rho_K}$$

Nu lite Geogebra där vi tyvärr måste byta $v(t)$ mot $y(x)$.

Jag har inkluderat den enklare modellen $v' = -g$.

Från plotten ser vi att:

$$\lim_{t \rightarrow \infty} v(t) = v_{\infty}$$

gränshastighet "terminal velocity". Inträffar när $mg = F_{luft}$ så att $v'(t) = 0$, ingen acceleration. Då $v' = -g + cv^2$ gäller att

$$v_{\infty} = \sqrt{g/c} = \sqrt{\frac{2mg}{c_d A \rho_L}}$$

i det allmänna fallet. En "skydiver" som faller med magen nedåt har kanske $m = 80$ kg, $c_d \approx 1$ (som för cykel+cyklist) så att

$$v_{\infty} = \sqrt{\frac{2 \cdot 80 \cdot 9.82}{1 \cdot 1.8 \cdot 0.3 \cdot 1.2}} \approx 49.5$$

Wikipedia anger 54 m/s, 195 km/h. 90 m/s med huvudet först.

32: Ett ODE-system

Vi avlossar ett gevär vertikalt uppåt.

Hur högt når kulan?

Vilken fart har kulan när den slår i marken?

Väsentligen samma modell som tidigare, $v' = -g + cv^2$, men vi behöver veta $s(t)$.

$$s'' = -g - \text{sign}(s') c (s')^2, \quad s(0) = 2.5, \quad s'(0) = 890$$

Anledningen till **sign** är att F_{luft} är motriktad rörelsen. Om kulan går uppåt är $s' > 0 \Rightarrow -\text{sign}(s') = -1$.

$s' < 0 \Rightarrow -\text{sign}(s') = +1$. Funktionen **sign** finns i Geogebra. $\text{sign}(0) = 0$.

Vi skriver om problemet som ett system av första ordningens ODE.

Låt $y = s$ och tag $z = s' = y'$. Då blir

$$s'' = z' = -g - \text{sign}(y) c z^2.$$

Systemet skrivs alltså (derivatorna i vänstra ledet, inga derivator i högra)

$$\begin{cases} y' = z \\ z' = -g - \text{sign}(z) c z^2 \end{cases}, \quad \begin{matrix} y(0) = 2.5 \\ z(0) = 890 \end{matrix}$$

I detta speciella fall skulle man först kunna lösa för z och sedan integrera för att få y . Detta går dock inte i mer allmänna fall när ekvationerna är kopplade.

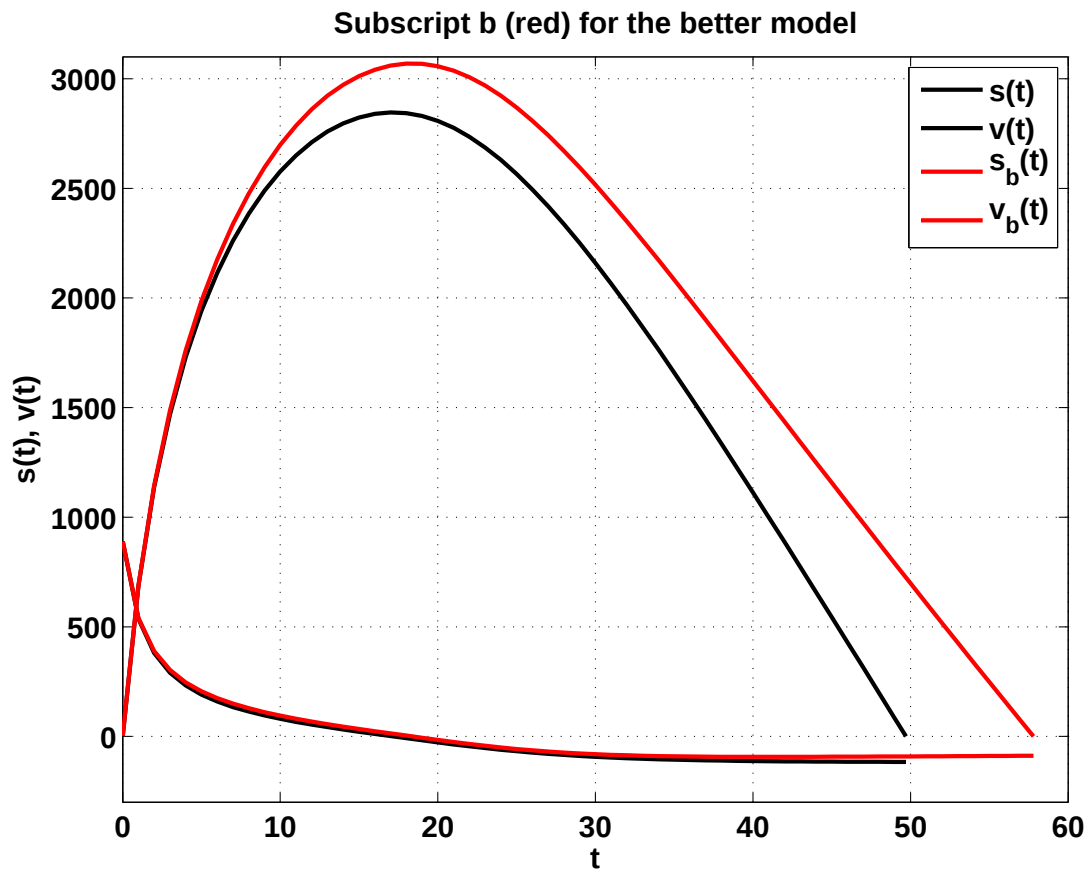
Har använt värdena $A = \pi r^2$, $r = 0.5 \cdot 7.8 \cdot 10^{-3}$, $m = 10^{-2}$ (10 gram), $c_d = 0.25$.

34: En bättre modell

- g är höjdberoende, är dock en liten effekt.
- ρ_L är höjdberoende, är viktigt att ta hänsyn till
- c_d är fartberoende, men är tämligen konstant för stora v .

För en snöflinga eller en fjäder gäller inte modellen för F_{luft} .
Luftmotståndet är då proportionellt mot v inte mot v^2 .

I följande bild, producerad i Matlab, har jag modellerat ρ_L
http://en.wikipedia.org/wiki/Density_of_air
och har utnyttjat "event-functions" för att bryta exekveringen när kulan och slår i backen och för att öka c_d på toppen ($v(t) = 0$).
Den förbättrade modellen stämmer bra med försök.



36: CFD

Om man räknar på mer komplicerad strömning, används CFD, Computation Fluid Dynamics (diskretisering + Navier-Stokes, PDE). Bilden (från ITM, 1996) visar ytnätet och en del av volymnätet för S2000.

