THE DERIVE - NEWSLETTER #101

ISSN 1990-7079

THE BULLETIN OF THE

DERIVE

USER GROUP

+ CAS-TI

Contents:

March 2016

CAS for Physics Examples

Leon Magiera and Josef Böhm

Introduction

This text is devoted to solving practical problems in the field of electricity and magnetism using computer algebra systems (CAS). The problems described usually appear in the programs of standard General Physics courses at university level (engineering and science). Some of them would also be suitable for high schools.

Maxima is a powerful tool for the manipulation of symbolic and numerical expressions, including differentiation, integration, vectors, matrices, . . . and so on. It also provides commands for plotting functions, curves and data in two and three dimensions. We chose *Maxima* because it is a free software package. It can be downloaded from its website, http://maxima.sourceforge.net, where documentation in several languages can also be found.

The use of *Maxima* significantly reduces the computational work and allows a student (or teacher) to concentrate on physical ideas (which is most important), rather than on the very time consuming technical side - performing the derivations by hand.

For a first try of *Maxima*, you may wish to try the examples in **First Steps with Maxima**. Please don't understand this paper as a complete introduction in working with *Maxima*. *Maxima*-experts will certainly find ways to solve some problems in another way. *Maxima* is much more powerful that it is shown here. You can produce complex programs, your own libraries and much more.

Leon Magiera is physicist at the Wroclaw University of Science and Technology, Josef Böhm is a retired school teacher and founder of the *International DERIVE and CAS-TI User Group*. His work was putting all parts together in a common form and adding *DERIVE* and *TI-NspireCAS* treatments of some examples. *DERIVE* is off the market since several years but still widely used all over the world. *DERIVE* and *TI-NspireCAS* are running up to the latest operations systems. The *DERIVE* and *TI-NspireCAS* parts can be distinguished by another font set.

the Authors

All files are available on request.

We welcome any comments and suggestions regarding this book. You may contact us via e-mail at:

leon.magiera@wp.pl
nojo.boehm@pgv.at

In preparation:

2nd part: Magnetic Field
3rd part: Circuits
4th part: Mechanics of Charged Particles

Download *CAS for Physics Examples* from http://www.acdca.ac.at/

Dear DUG Members,

instead of an extended letter I will inform you about intended presentations to be given at TIME 2016, Mexico City.

Best regards until next time

Josef

Classical Optimization for Inventory Management and Control with Maple

Third Degree Polynomial Equations

Implicit Curves and Tangent Lines with the TI-Nspire CX CAS

CAS for Physics Problems

Wonderful World of Pedal Curves

Development of the variational thought in secondary students

The Tension between informal and formal Thinking in Geometry through a Digital Alternative of Hyperbolic Geometry

Dynamic Riemann sums to evaluate integrals and volumes with TI-Nspire

Investigating Stars in 2D and 3D with DGS and CAS

Hypothetical Learning Trajectories that use Digital technology to solve a Optimization Problem

TI-Nspire as a technological Support in Learning Conics. The Case of Ellipse

Mathematics Lessons and Classroom examples, inspired by the articles in Newspapers

The change from a calculation oriented to a problem solving oriented mathematics education supported/caused by technology

Tinkerplots – Presentation & Workshop

Study of Nonlinear Oscillator with CAS through Analytical, Numerical and Qualitative Approaches

Analysis of non-analytical smooth functions using CAS

Dynamic Geometry Software and Tracing Tangents in the Context of the mean Value Theorem: Technique and Theory Production

The use of technology and mathematics to make simulations of natural phenomena

Design and Use online Platforms to learn Mathematics and the Use of them in Simulations of Problems in Applied Science

New literacies and social practices in mathematics learning with digital technologies: a sociocultural perspective research

Reproducible Research with R

The impact of computer use on learning of quadratic functions

L'Hospital's Weight Problem: Crossing a New Border

An exploratory study of the use of digital resources in Math class: orchestration and mathematical work spaces

Teaching Mathematics and Statistics using a CAS and a Statistical Software Package - Findings from Student Surveys

Geogebra influence on Learning Analytical Geometry

Modelling in action: Examining how Students Approach Modelling real Life Situations. Model of the Movement of an Elevator vs the Movement of a Ball

Download all *DNL*-DERIVE- and TI-files from
http://www.austromath.at/dug/

The *DERIVE-NEWSLETTER* is the Bulletin of the *DERIVE* & CAS-*TI User Group*. It is published at least four times a year with a content of 40 pages minimum. The goals of the *DNL* are to enable the exchange of experiences made with *DERIVE*, *TI*-CAS and other CAS as well to create a group to discuss the possibilities of new methodical and didactical manners in teaching mathematics.

Editor: Mag. Josef Böhm
D´Lust 1, A-3042 Würmla, Austria
Phone: ++43-(0)660 3136365
e-mail: nojo.boehm@pgv.at

Contributions:
Please send all contributions to the Editor. Non-English speakers are encouraged to write their contributions in English to reinforce the international touch of the *DNL*. It must be said, though, that non-English articles will be warmly welcomed nonetheless. Your contributions will be edited but not assessed. By submitting articles the author gives his consent for reprinting it in the *DNL*. The more contributions you will send, the more lively and richer in contents the *DERIVE* & CAS-*TI Newsletter* will be.

Next issue: June 2016

Preview: Contributions waiting to be published

Some simulations of Random Experiments, J. Böhm, AUT, Lorenz Kopp, GER
Wonderful World of Pedal Curves, J. Böhm, AUT
Tools for 3D-Problems, P. Lüke-Rosendahl, GER
Hill-Encryption, J. Böhm, AUT
Simulating a Graphing Calculator in *DERIVE*, J. Böhm, AUT
Graphics World, Currency Change, P. Charland, CAN
Cubics, Quartics – Interesting features, T. Koller & J. Böhm, AUT
Logos of Companies as an Inspiration for Math Teaching
Exciting Surfaces in the FAZ / Pierre Charland´s Graphics Gallery
BooleanPlots.mth, P. Schofield, UK
Old traditional examples for a CAS – What´s new? J. Böhm, AUT
Where oh Where is It? (GPS with CAS), C. & P. Leinbach, USA
Mandelbrot and Newton with *DERIVE*, Roman Hašek, CZK
Tutorials for the NSpireCAS, G. Herweyers, BEL
Some Projects with Students, R. Schröder, GER
Dirac Algebra, Clifford Algebra, D. R. Lunsford, USA
A New Approach to Taylor Series, D. Oertel, GER
Henon & Co; Find your very own Strange Attractor, J. Böhm, AUT
Rational Hooks, J. Lechner, AUT
Statistics of Shuffling Cards, H. Ludwig, GER
Charge in a Magnetic Field, H. Ludwig, GER
Factoring Trinomials, D. McDougall, CAN

and others

Impressum:
Medieninhaber: *DERIVE* User Group, A-3042 Würmla, D´Lust 1, AUSTRIA
Richtung: Fachzeitschrift
Herausgeber: Mag. Josef Böhm

DERIVE and Windows 10

Dear Josef!

I purchased a Surface 4 with Windows 10 just recently. It it possible to install DERIVE in any way?

Best regards,

Willi

Answer from our OS-expert Günter Schödl:

Hello Willi!

Yes, it works but unfortunately it is not so easy as with earlier Windows versions. Unpack the attached archive and start install.cmd as administrator.
Much Success and best regards
Günter

I put the archive on my onedrive http://1drv.ms/1QO4aT where you can download it.

The archive `winhlp32-windows-10.zip` is contained in the package MTH101.zip, Josef

Implicit plotting with TI-NspireCAS

Dear Michel,

best regards in the first days of the New Year.

I have a question regarding plotting with TI-Nspire:

(1) Am I right, that it is not possible to perform implicit plots?

(2) Am I also right that ir is not possible to produce contour plots (it is possible with V200)?

Many thanks in advance,

Josef

This is Michel's answer:

You are absolutely right :

1) One thing many people never knew is the fact it was possible – but so long due to the slow processor – to have implicit plot on Voyage 200. Here is how : you simply type, in the Y-Editor of a 2D plot function mode window, y1(x) = zeros(f(x,y), y) where f(x,y) is your expression equal to 0. Be very very patient and you will se the implicit plot!!! Note that you will need to select the complex format (« principal branch ») and not the real one in order to get the complete graph in the cas of the third degree polynomial equation. In Nspire CAS, this won't work for a general f(x,y) but works if f(x,y) = 0 can be solved analytically for y. So it works for a third degree polynomial expression in y because they have added the analytical formulas for a third degree polynomial (but these formulas are very ugly as we all know and I strongly prefer my own function « compact_cubic » who shows something Derive is giving). But if you have to plot something as x = g(y), this is possible in Nspire CAS using text and dragging the equation onto one of the 2 axes.

2) For contour plot, one thing you can do with Nspire CAS is to use a 3D plot window and then the menu « Trace ». But this is not as having contour plots in 2D as expected.

Dear Josef, what I wrote about Nspire CAS « fake » implicit plotter was – probably, I need to check again – not correct. Here is why : with Nspire CAS OS 4, I have tried to plot the curve exp(x*y)- x -2y = 0. On the computer version, if you go to a 2D plot window in function entry mode and type, for f1(x), zeros(exp(x*y)-x-2y,y), it shows one part of the graph (the graphs consists of 2 separate curves, check with Derive). So, I suspect Nspire CAS is able to do some implicit plotting even if the equation can't be solved. If you have an old version of Nspire CAS (not OS 4, not OS 3.6, maybe OS 3.2 or earlier versons, try to plot the curve and tell me what you see).

Hi again,

I found version 3.2 on a handheld: plots also only the lower part of the curve.

Another handheld of my collection (V. 1.4): doesn't plot anything.

I don't know if implicit plotting is possible with CASIO ClassPad, I'll try.

Best regards again,
Josef

Thanks Josef! This means that TI team is doing something with their product but it seems this feature has not a big importance for them --implicit plotting is not a favorite high school subject. We are still far away from our good old Derive but I think I will focus on this for TIME 2016 and hope some TI members will be there to hear my message.

Michel

Hi again Josef, this morning I put 4 new AAA batteries in my V200 and, after many minutes (in fact about 85 minutes, I was doing something else and did not check exatly the time), the 2 parts of the implicit curve exp(x*y) – x – 2*y = 0 were on the V200 screen!

Michel

This is a part of a mail sent by our friend David Sjöstrand from Sweden

```
I am sending a copy of this mail to my old Austrian friend Josef Böhm with
whom I have cooperated a lot since 1992. Josef is the chair of DUG,
http://www.austromath.at/dug/ and has a lot of knowledge of technology in
mathematics teaching. He also has a lot of contacts with people interested
in technology in mathematics teaching all over the world.

I will let you know if I publish anything on my web site
www.davidsjostrand.com.

Have you seen this?

Investigate the sequences defined by

a)    x(n+1) = 4*x(n)+7.  x(0) = -7/3
b)    x(n+1) = 5*x(n)-3.  x(0) = 3/4

Obviously both sequences are constant, x(n) = x(0) for all n.

However if you calculate the elements of the sequences using Excel you get
an interesting result in a).  I attach an Excel file.
```

Give it a Try, Josef. Nice (easy) problem for students: Find the rule behind?

Josef Böhm: Dynamic Systems / Dynamische System

I have been interested in fractals, “Chaos“ and Dynamic Systems since many years. Treating these issues became possible for everybody with availability of computers and the respective software. Special programs like *FractInt* have been on the market since long. But now supported by spreadsheets, computer algebra and own programming it makes much more sense and fun as well to investigate these phenomena.

By a book review a came across *Hartmut Bossel*’s “*System Zoo*” book series. These books are a real repository and treasure box for applied mathematics. There was also information about the program *VENSIM*. This is a commercial simulation software free of charge for teaching purposes.

Then I purchased the *System Zoo*-CD and was very enthusiastic about the many possibilities using *VENSIM*. My ambition came up to treat a not too complex problem (Tourism and Environment) with other tools which are available in our schools. I wanted to learn about the special features, their advantages and disadvantages working through this example.

I had in mind *MS-Excel*, *DERIVE*, *WIRIS*, *TI-NspireCAS* and *GeoGebra*. All these programs offer sliders which promised making the simulations much more dynamic varying the parameters. An additional challenge was to transfer the model into a differential equation or a system of differential equations and then solving it numerically or – if possible – analytically.

I was so much fascinated by this first example that I could not resist proceeding and trying other ones. So it could happen that the paper comprises more than 100 pages finally.

The results are aesthetically appealing - at least in my opinion - and they may wake up appetite for further experimenting and discovering. The “beautiful” and “strange” attractors might make the systems of differential equations interesting even for students who are not so enthusiastic with mathematics.

Unfortunately I could not address here essential interpretation of the generated tables and diagrams. I refer to *Bossel*’s books and many other resources.

All files which are presented in this paper are available on request. Please send an email.

I wish much fun and would be very delighted receiving reactions.

Josef Böhm
nojo.boehm@pgv.at

References:

VENSIM Simulation Software
http://www.ventanasystems.co.uk/forum/index.php

Hartmut Bossel, *SystemZoo 1, 2, 3*, Books on Demand, Noderstedt
http://www.hartmutbossel.de/ezooinf.htm

Hartmut Bossel, *SystemZoo*, cotec Verlag Rosenheim (CD including VENSIM PLE)
http://www.cotec-verlag.de

Josef Böhm, *Dynamic Systems/Dynamische Systeme*,
http://rfdz.ph-noe.ac.at/acdca/materialien.html

Collapse of an Ecosystem

A more complex simulation with a historical background

Bossel cites a source which explains the collapse of the white-tailed-deer population in the Kaibab Forest (North Rim of Grand Canyon) as a consequence of shooting the predators which feed on these deer.

Prior to 1907 there was a population of approx 4000 deer living on an area of about 320 000 ha. Within a period of 15 to 20 years hunting predators (cougars, wolves and coyotes) was forced and about 8000 of them were shot. This was followed by an enormous growth of the deer population.

Sycamore Canyon, Kaibab National Forest

It was 1918 when the stock of deer had more than decupled. This caused an overexertion of food supplies. Until 1924 the deer population reached a number of 100 000 animals. Caused by lack of food 60% of the animals perished in the following two winter periods.

Vegetation was destroyed in such a way, that only half of the deer population compared with its size before this development could exist in the long run.

White-tailed-deer (Odocoilus virginianus)

1974 tried *Goodman* to simulate this system by a model which delivered results matching satisfying with the real process.

Explanation of the model

The *Deer* feed on an AREA (320 000 ha) on *Food*. *Increase Food* is governed by its *Regeneration Time*. The *Growth rate Deer* is a function of *Food Supply.* This is the amount of food available for each animal. *Food Demand* depends on the stock of *Deer* and on the DAILY REQUirement of one deer (2000 Kcal). *Food* grows again according to the MAX FOOD CAPACITY (480 Mio Kcal). *Increase Food* is determined by the *Regeneration Time* which is a function of *Vegetation Density*.

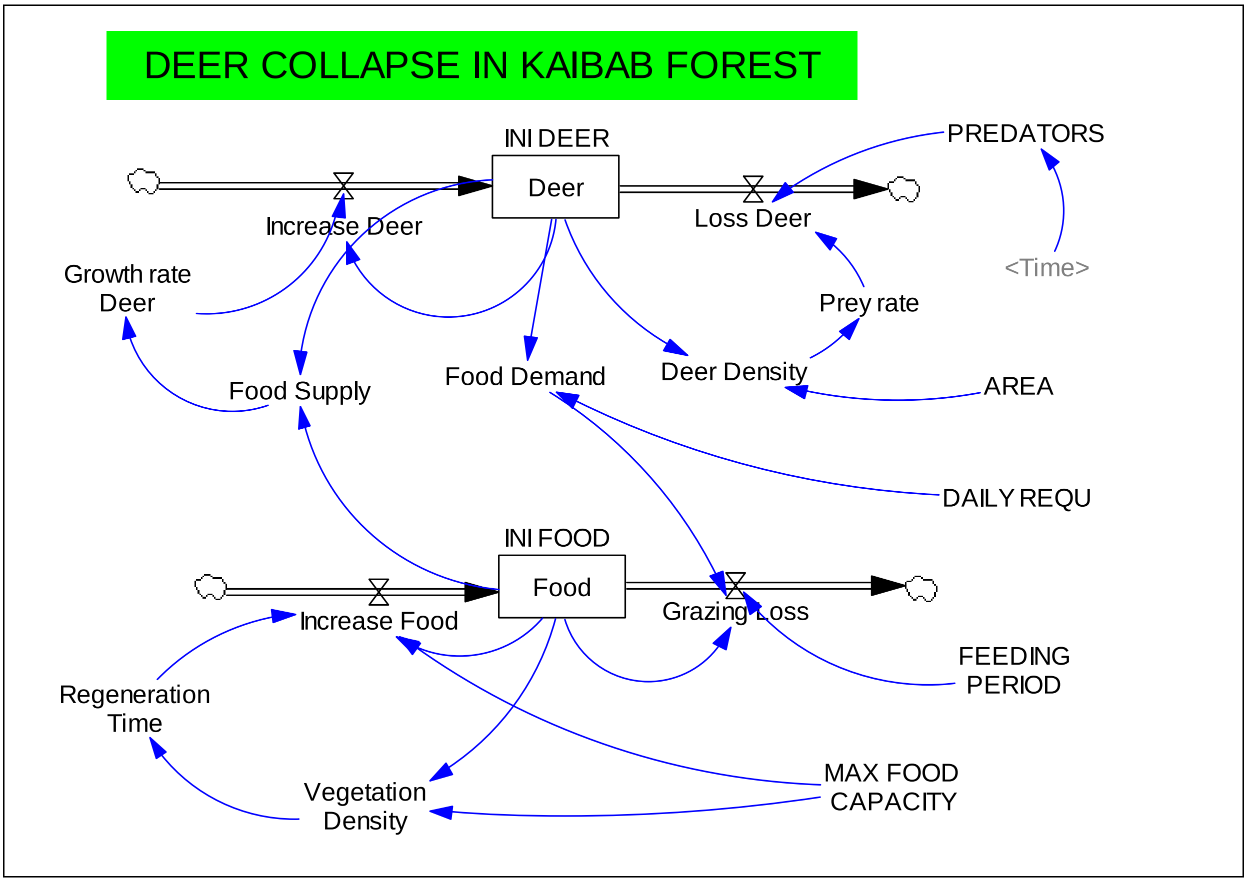
What about the predators? The *Deer* population suffers *Loss Deer* by the PREDATORS, whose number decreases linearly caused by shooting numbers. The *Prey rate* is a function of the *Deer Density* (= deer/ha).

A more detailed explanation of the parameters is given in *System Zoo*.

Especially interesting is the use of functional dependencies which are given by tables (= nodes of the describing functions). We will find these tables in the document under WITH LOOKUP.

The simulation is running for 50 years with an increment of 0.25 years.

Note the use of the IF-function with its syntax very similar to the syntax used in Computer Algebra Systems.



The document comprises all constants, all equations and all simulation parameters (originally presented numbered and in alphabetical order):

(01) AREA = 320000

(02) DAILY REQU = 2000

(03) Deer= INTEG (+Increase Deer – Loss Deer, INI DEER)

(04) Deer Density = Deer/AREA

(05) FEEDING PERIOD = 1

(06) FINAL TIME = 50

(07) Food = INTEG (+Increase Food – Grazing Loss, INI FOOD)

(08) Food Demand = DAILY REQU ∗ Deer

(09) Food Supply = Food/Deer

(10) Grazing Loss =
IF THEN ELSE(Food Demand >= (Food/FEEDING PERIOD),
Food/FEEDING PERIOD, Food Demand)

(11) Growth rate Deer = WITH LOOKUP (Food Supply, ([(0,-1)-(10000,1)],
(0,-0.5), (500,-0.15),(1000,0),(1500,0.15),(2000,0.2) ,(200000,0.2)))

(12) Increase Deer = Growth rate Deer ∗ Deer

(13) Increase Food = (MAX FOOD CAPACITY – Food)/Regeneration Time

(14) INI DEER = 4000

(15) INI FOOD = 4.7e+008

(16) INITIAL TIME = 0

(17) Loss Deer = Growth rate Pred ∗ PREDATORS

(18) MAX FOOD CAPACITY = 4.8e+008

(19) PREDATORS = WITH LOOKUP (Time, ([(0,0)-(50,300)],
(0,265),(5,245),(10,200),(15,65),(20,8),(25,0),(30,0), (35,0),(40,0),(50,0)))

(20) Prey rate = WITH LOOKUP (Deer Density, ([(0,0)-(0.35,60)], (0,0),
(0.0125,3),(0.025,13),(0.0375,28),(0.05,51),(0.0625 ,56),(0.125,56),(0.4,56)))

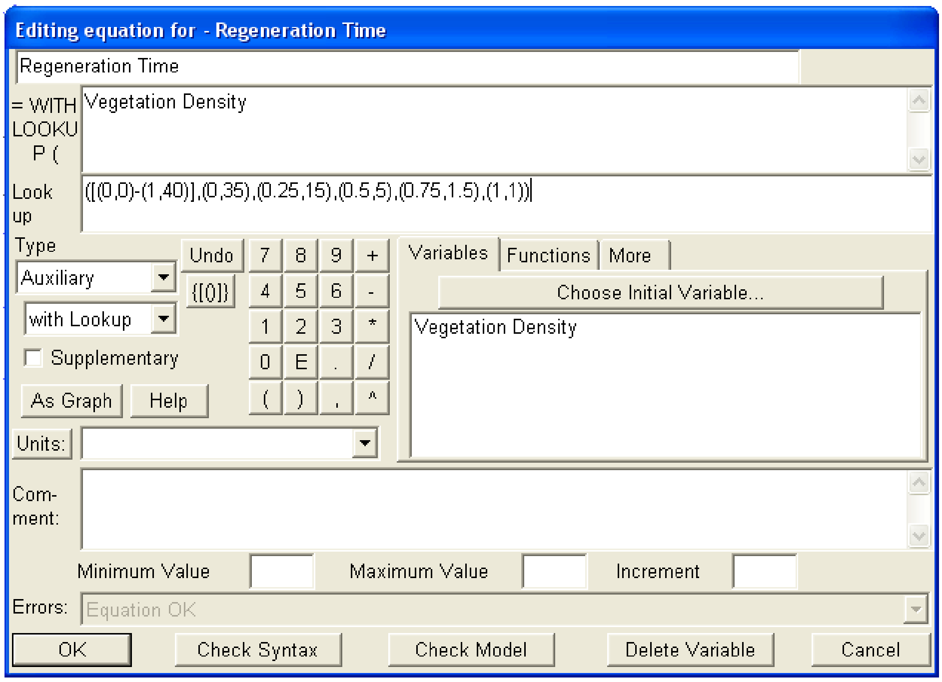
(21) Regeneration Time = WITH LOOKUP (Vegetation Density,([(0,0)-(1,40)],
(0,35),(0.25,15),(0.5,5),(0.75,1.5),(1,1)))

(22) SAVEPER = TIME STEP

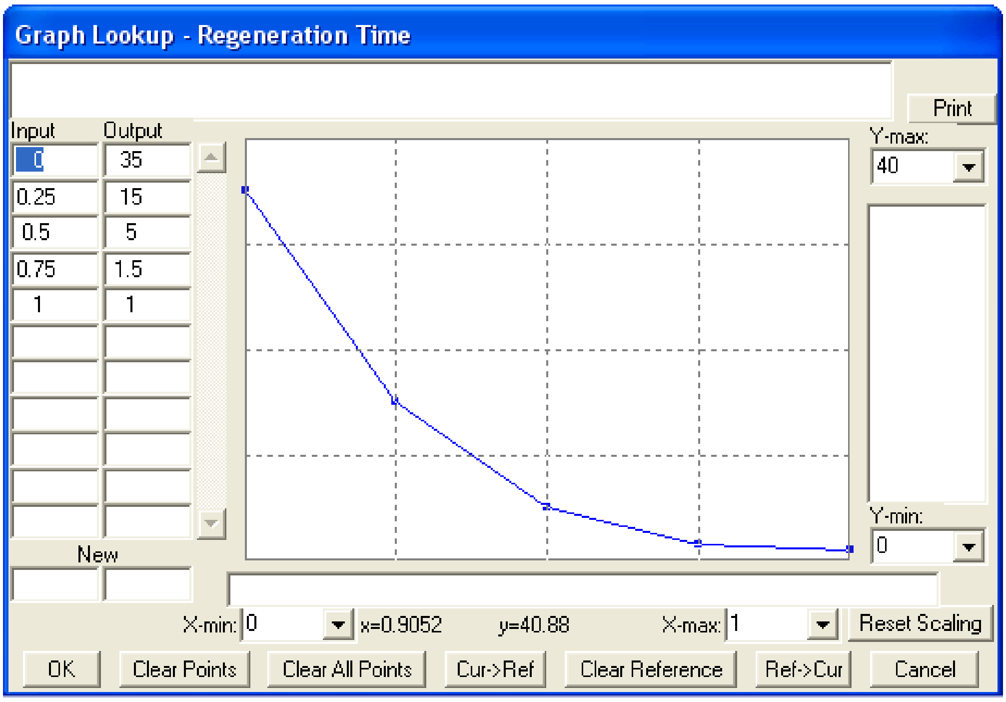
(23) TIME STEP = 0.25

(24) Vegetation Density = Food/MAX FOOD CAPACITY

Let's inspect function (21) *Regeneration Time (Vegetation Density)* as an example for working WITH LOOKUP:



After pressing the As Graph-button the graph of the piecewise defined function is presented:



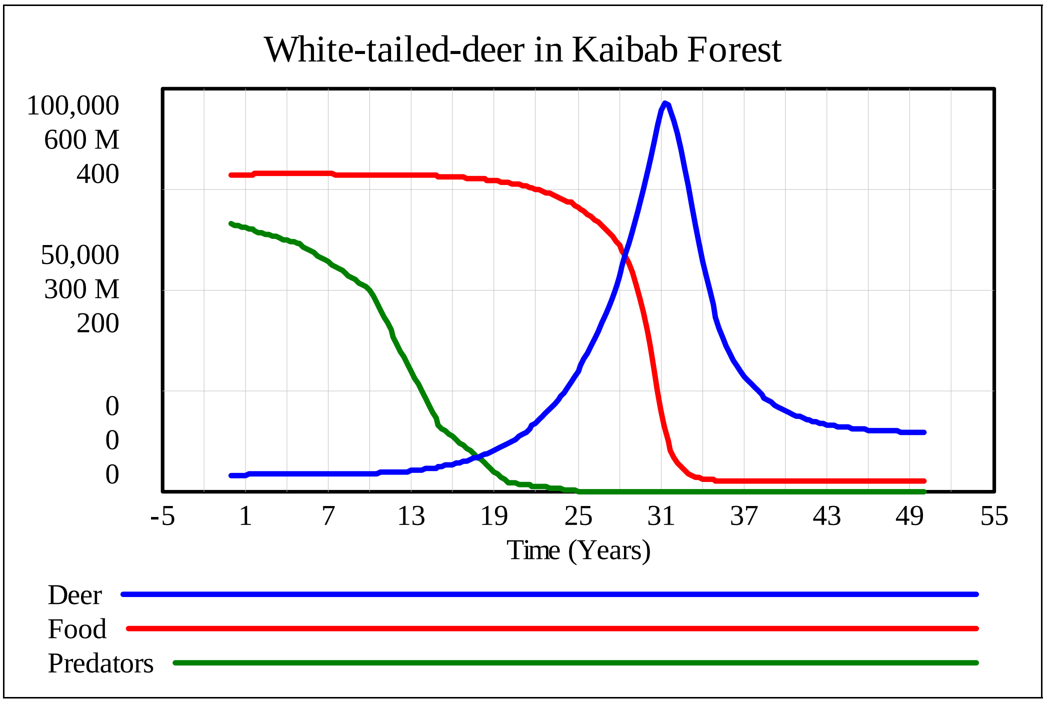
We would be able to enter the nodes directly into the grid. It is easy to recognize that there is a linear interpolation between the given points (nodes).

We run the simulation and inspect the first results.

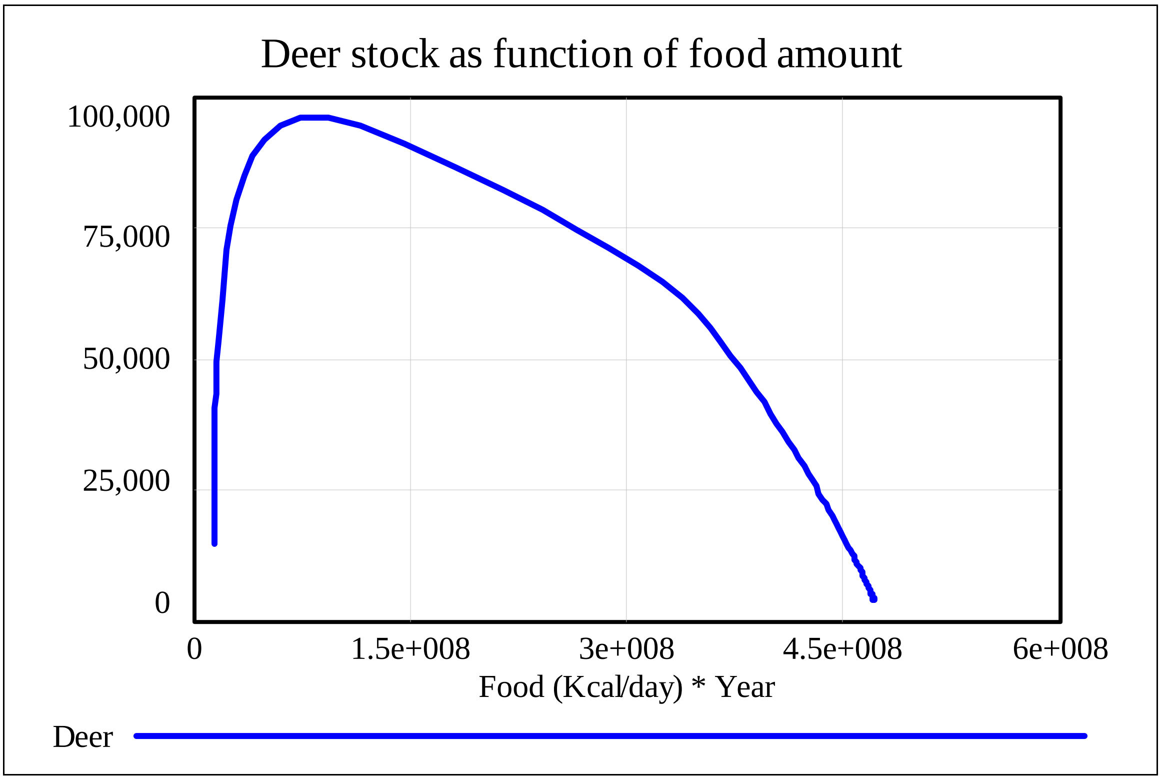
How are the deer doing?

How develops the available amount of food?

How are the predators doing? (Thanks human interaction – they are doing obviously badly!)



The second graph shows the stock of deer as function of the food amount.



Starting point is at right bottom and the development ends on the left hand side.

The result of the simulation matches with the real historical occurrence. The reduction of the vermin led to an explosion of the deer population which caused a disastrous overgrazing of the available food capacity. A huge number of deer died of hunger and finally the deer stock became stabilized on a level based on the much reduced amount of food.

As I am – unfortunately enough – no *Excel*-expert, I don't know how to realize the functional dependencies together with their connected linear interpolations in an easy way in a spreadsheet.

It would be great if any reader of these lines could accomplish this chapter performing the simulation with *Excel*. I would be very grateful for respective information.

I will come back to *MS-Excel* later.

But we can be glad having some other tools available to try with!

The *DERIVE*-Model

I accepted the challenge treating this system with *DERIVE*.

There appears the same problem: how to realize the piecewise defined functions with the linear interpolations?

As a DERIVIAN one has immediately the idea to connect the points given in a matrix using the CHI-function.

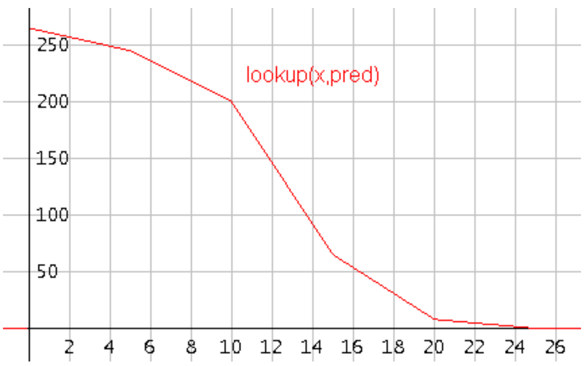
First of all the given data are fixed. Then my first attempt follows finding a function which is equivalent to the LOOKUP-function.

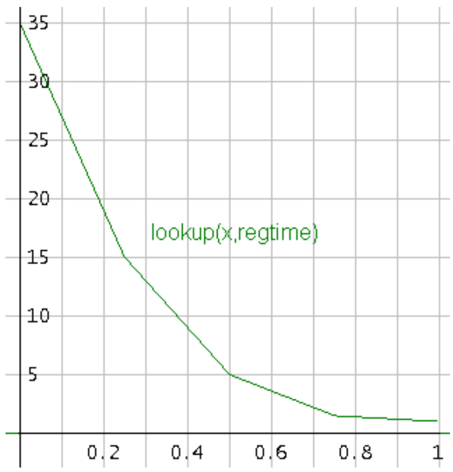
$$\left[\text{area} := 320000,\ \text{feedper} := 1,\ \text{max\_food\_cap} := 4.8\cdot 10^{8},\ \text{dayly\_requ} := 2000,\ \text{ini\_food} := 4.7\cdot 10^{8},\ \text{ini\_deer} := 4000\right]$$

$$\left[\text{pred} := \begin{bmatrix} 0 & 265 \\ 5 & 245 \\ 10 & 200 \\ 15 & 65 \\ 20 & 8 \\ 25 & 0 \\ 50 & 0 \end{bmatrix},\ \text{preyr} := \begin{bmatrix} 0 & 0 \\ 0.0125 & 3 \\ 0.025 & 13 \\ 0.0375 & 28 \\ 0.05 & 51 \\ 0.0625 & 56 \\ 0.125 & 56 \\ 0.4 & 56 \end{bmatrix},\ \text{regtime} := \begin{bmatrix} 0 & 35 \\ 0.25 & 15 \\ 0.5 & 5 \\ 0.75 & 1.5 \\ 1 & 1 \end{bmatrix},\ \text{gr\_deer} := \begin{bmatrix} 0 & -0.5 \\ 500 & -0.15 \\ 1000 & 0 \\ 1500 & 0.15 \\ 2000 & 0.2 \\ 200000 & 0.2 \end{bmatrix}\right]$$

$$\text{lookup}(x\_,\ pk) := \sum_{i=1}^{\text{DIM}(pk)-1} \chi(pk_{i,1},\ x\_,\ pk_{i+1,1})\cdot\left(\frac{pk_{i+1,2} - pk_{i,2}}{pk_{i+1,1} - pk_{i,1}}\cdot(x\_ - pk_{i,1}) + pk_{i,2}\right)$$

The graphs are looking pretty nice. Compare the graph for the regeneration time with the respective *VENSIM*-graph (page 25)! On the first glance you will not recognize any difference.





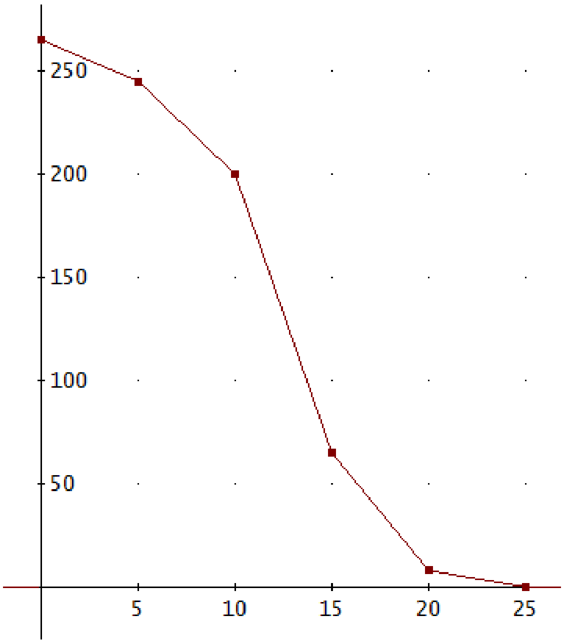
But don't be happy too early! Inspecting the value tables (e.g. the numbers of the PREDATORS) we recognize the deficiency of the CHI-function in the nodes where the function is undefined. Hence this implementation is of no need for us.

0		1	2	3	4	5	6			
±132.5 + 132.5		261	257	253	249	?	236			
24	25	26	27	28	29	30	31	32	33	34
1.6	0	0	0	0	0	0	0	0	0	0

The next function fulfils our requirements.

```
lu(x_, pk, f) :=
  Prog
    f := IF(pk↓1↓1 ≤ x_ ≤ pk↓2↓1, (pk↓2↓2 - pk↓1↓2)/(pk↓2↓1 - pk↓1↓1)·(x_ - pk↓1↓1) + pk↓1↓2, 0)
    pk := REST(pk)
    Loop
      If DIM(pk) = 1 exit
      f := f + IF(pk↓1↓1 < x_ ≤ pk↓2↓1, (pk↓2↓2 - pk↓1↓2)/(pk↓2↓1 - pk↓1↓1)·(x_ - pk↓1↓1) + pk↓1↓2, 0)
      pk := REST(pk)
    LIM(f, x, x_)
```

The graphs fit exactly (nodes and segments) and the value tables don't show any exceptions.



TABLE(lu(x, pred), x, 0, 50)'

0	1	2	3	4	5	6	7	
0	261	257	253	249	245	236	227	
24	25	26	27	28	29	30	31	32
1.6	0	0	0	0	0	0	0	0

Before programming I always tried to work through the system(s) acting as a "human spreadsheet". I'd like to recommend this way treating such systems in classroom. Then the interconnections become clear and programming and/or transfer to a "real" spreadsheet becomes very easy.

Here I benefit from the results of the *VENSIM* simulation because I can use its tables as a reference for programs and/or any other treatments.

I try demonstrating the manual procedure – supported by the *DERIVE* made lu-functions – step by step.

The table consists of 16 columns.

In row 1 I start with Time = 0, FOOD = $4,7 \cdot 10^8$, DEER = 4000 and PREDAT = 265. The numbers in the last row indicate the order of calculation.

I go on with the entries for *Food Demand*, proceed with *Browsing Loss* and close the line with *Increase Deer*. Then we enter in row 2 (Time = 0.25) the new amount of FOOD and the new DEER population (13, 14) followed by 1 through 12. We can follow the formulae (equations) how they are listed in the document.

Row nr	Time	FoodDem	BrowsLoss	VegDens	RegTime	FoodIncr	FOOD
1	0	$8·10^6$	$8·10^6$	0.979167	1.041666	$9.6·10^6$	$4.7·10^8$ (*)
2	0.25	$8.0025·10^6$	$8.0025·10^6$	0.98	1.04	$9.23077·10^6$	$4.704·10^8$
3	0.50						$4.70707·10^8$
		①	②	③	⑥	⑦	①③

PREDAT	DeerDens	PreyRate	Loss Deer	Food Sup	GrRateDe	IncrDeer	DEER
265	0.0125	3	795	117500	0.2	800	4000
264	0.0125039	3.00312	792.824	117563	0.2	800.25	4001.25 (*)
							4003.11
⑤	④	⑧	⑨	⑩	①①	①②	①④

(*) Calculating the increases (for FOOD and DEER) one has to consider the time increment *dx*.
So for DEER(Time = 0.25) = 4000 + (800 – 795) · 0.25 = 4001.25.

The values in the columns for *Regeneration Time* (RegTime), *Predators* (PREDAT), PreyRate and *Growth rate Deer* (GrRateDe) were found using the `lu`-function (in analogy to WITH LOOKUP).

```
lu(0.979167, nachwzt) = 1.041666        lu(0.98, nachwzt) = 1.04

lu(0.0125, beuterate) = 3               lu(0.0125039, beuterate) = 3.00312

lu(117500, zuw_r_h) = 0.2               lu(0.25, raeub) = 264
```

The table from above can be transferred one by one into a *DERIVE*-program.

With *DERIVE* I collect all values in a table, too. For plotting the diagrams I have to select the respective columns.

First of all is a short function needed for the grazing or *Browsing Loss*.

```
graz_loss(x, y) :=
  If x ≥ y/feedper
     y/feedper
     x
```

The `lu`-function was introduced earlier. The full program is following.

```
kaibab(n, dx, i, tab, t, f_dem, brows_loss, veg_d, deer_d, predators,
       reg_time, food_inc, prey_r, inc_deer, food_supply, loss_deer,
       deer, food) :=
PROG(
   i := 1,
   tab := [["RNr", "Time", "FDem", "GrazL", "VegD", "RegTime",
        "FoodIncr", "PreyR", "DeerLoss", "FSupply", "DeerIncr", "Food",
        "Deer", "Predators"]],
     t := 0, [deer := ini_deer, food := ini_food],
     LOOP(
          IF(i > n, RETURN tab),
          f_dem := deer·daily_requ,
          brows_loss := graz_loss(f_dem, food),
          veg_d := food/max_food_cap,
          deer_d := deer/area,
          predators := lu(t, pred),
          reg_time := lu(veg_d, regtime),
          food_inc := (max_food_cap - food)/reg_time,
          prey_r := lu(deer_d, predr),
          loss_deer := prey_r·predators,
          food_supply := food/deer,
          inc_deer := lu(food_supply, gr_deer)·deer,
          tab := APPEND(tab, [[i, t, f_dem, brows_loss, veg_d, reg_time,
                food_inc, prey_r, loss_deer, food_supply, inc_deer, food,
                deer, predators]]),
          deer := deer + (inc_deer - loss_deer)·dx,
          food := food + (food_inc - brows_loss)·dx,
          t :+ dx, i :+ 1))
```

The first 4 rows are – very enjoyable – completely corresponding with the manually calculated table and the *VENSIM*-results as well.

```
kaibab(4, 0.25)
```

RNr	Time	FDem	GrazL	VegD	RegTime	FoodIncr	PreyR
1	0	8000000	8000000	0.9791666666	1.041666666	9600000	3
2	0.25	8002500	8002500	0.98	1.04	9230769.23	3.003125
3	0.5	8006212.5	8006212.5	0.9806397235	1.038720552	8946518.547	3.007765625
4	0.75	8011001.945	8011001.945	0.9811294662	1.037741067	8728435.7	3.013752431

DeerLoss	FSupply	DeerIncr	Food	Deer	Predators
795	117500	800	470000000	4000	265
792.825	117563.2614	800.25	470400000	4001.25	264
791.0423593	117585.4543	800.62125	470707067.3	4003.10625	263
789.603137	117573.8433	801.1001945	470942143.8	4005.500972	262

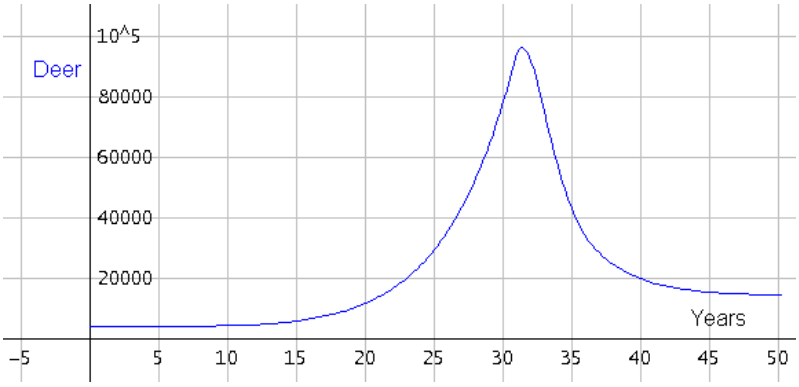
These are the values for FOOD and DEER for the last three quarters of a year.

```
(kaibab(202, 0.25))↓↓[2, 12, 13]
       [200, 201, 202]
```

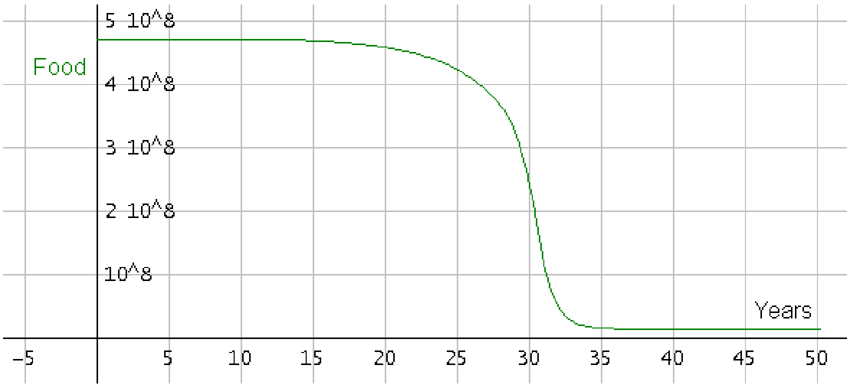
49.5	14277004.0601	14566.1776742
49.75	14277004.0192	14544.4896531
50	14277003.988	14524.4282306

Please compare the *DERIVE*-diagrams with the plots generated with *VENSIM* (pages 25, 26)

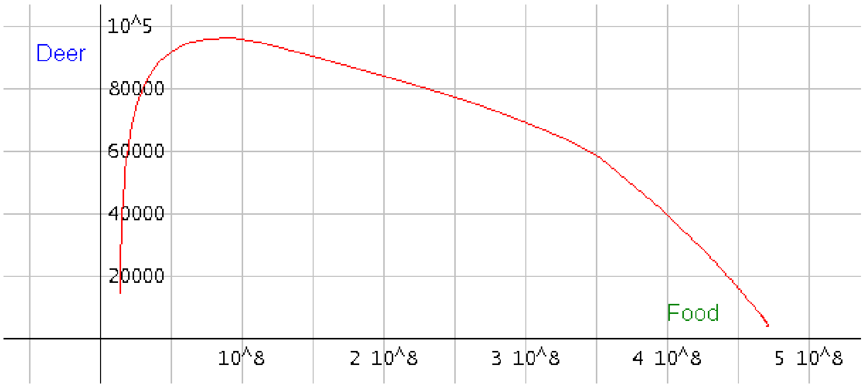
DELETE((kaibab(202, 0.25))↓↓[2, 13], 1)



DELETE((kaibab(202, 0.25))↓↓[2, 12], 1)



DELETE((kaibab(202, 0.25))↓↓[12, 13], 1)



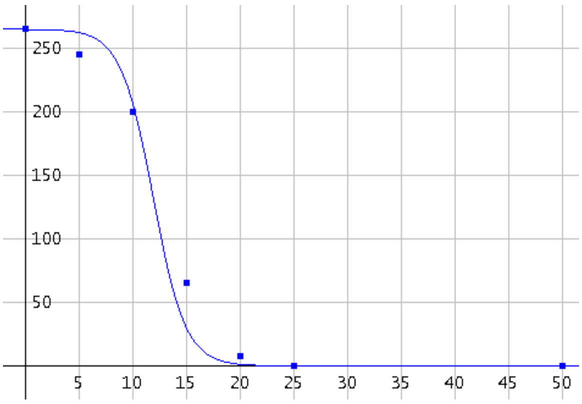
I promise to try modelling the system by using differential equations later.

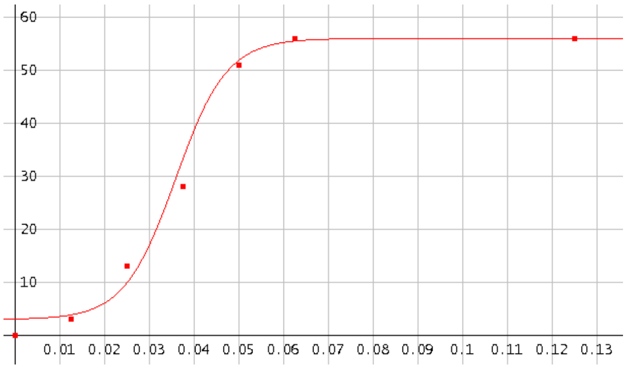
For treating this problem with a spreadsheet program it would be useful to approximate the WITH LOOKUP functions for PREDATORS, *Prey rate*, *Regeneration Time* and *Growth rate Deer* by an „ordinary“ function.

This is a nice task for its own. Sliders and meaningful considerations lead to appropriate functions.

$$\text{pred\_f}(x) := \frac{701985}{e^{0.6625\cdot x} + 2649}$$

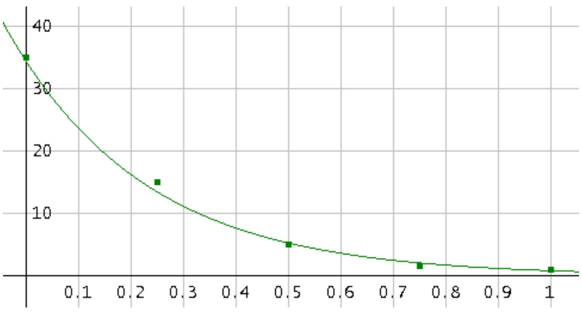
$$\text{preyr\_f}(x) := 3 + \frac{53}{1 + 529\cdot e^{-174.9\cdot x}}$$

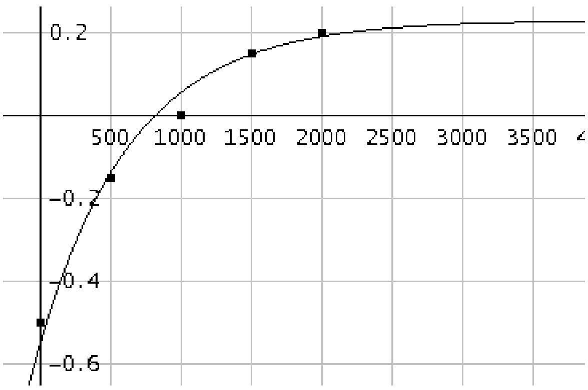




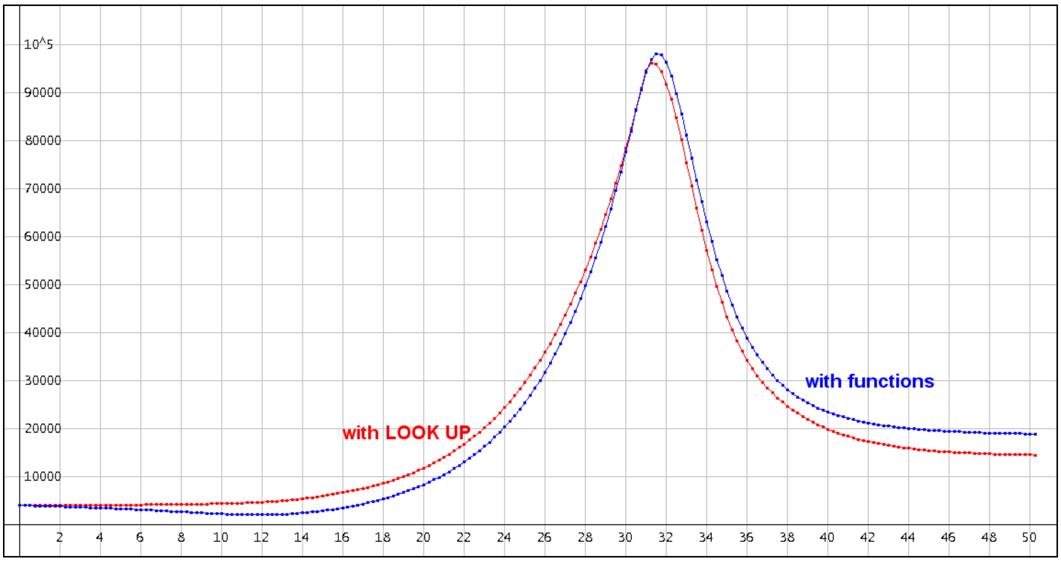
$$\text{regtime\_f}(x) := 34.4089\cdot e^{-3.7653\cdot x}$$

$$\text{gr\_deer\_f}(x) := 0.23 - 0.78\cdot e^{-0.0015\cdot x}$$





How good are these approximations? We can check this by replacing the lu-functions in program `kaibab` gaining program `kaibab_f`. Then we will compare the graphs of the deer population derived from both programs.

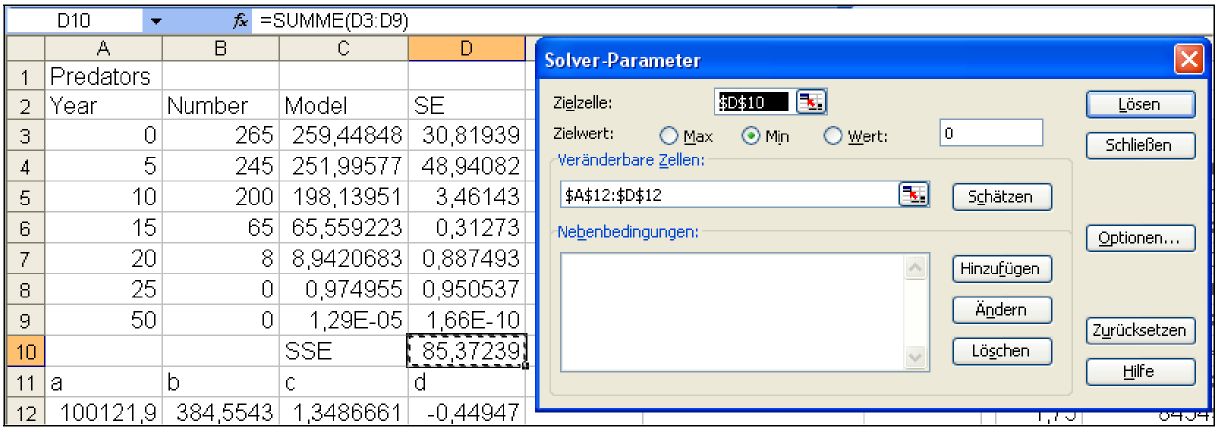


The result is impressive, isn't it? The graphs for the food are almost identical, too.

These functions make modelling with spreadsheet much more comfortable. Now let's try *MS-Excel*!

The *MS-Excel*-model

We could take over the functions from *DERIVE* but there is a “SOLVER“ available in the spreadsheet program which is a very versatile tool. We need some “inspiration” from the form of the scatter diagrams in order to make the right decision for the type of function which we should choose for the approximation.



	A	B	C	D
1	Predators			
2	Year	Number	Model	SE
3	0	265	259,44848	30,81939
4	5	245	251,99577	48,94082
5	10	200	198,13951	3,46143
6	15	65	65,559223	0,31273
7	20	8	8,9420683	0,887493
8	25	0	0,974955	0,950537
9	50	0	1,29E-05	1,66E-10
10			SSE	85,37239
11	a	b	c	d
12	100121,9	384,5543	1,3486661	-0,44947

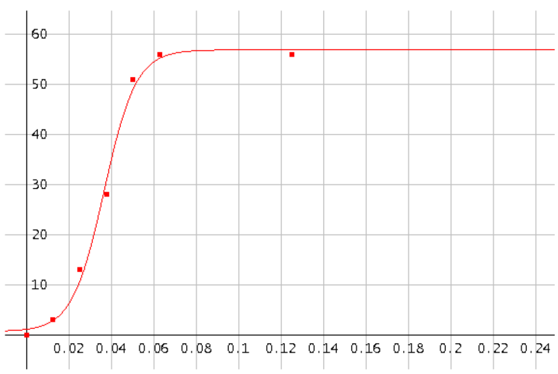
“Inspired“ by *DERIVE* I choose for the predator function the form $\dfrac{a}{b+c\cdot e^{-d\cdot x}}$ and enter in cell C3 as follows: =$A$12/($B$12+$C$12*EXP(-$D$12*A3)).

We enter initial values for the solving (= iteration-) procedure in cells A12 to D12 – and this is the trickier part of the task. However, here we can refer to earlier results again. I found approximating the growth rate of the deer the most difficult.

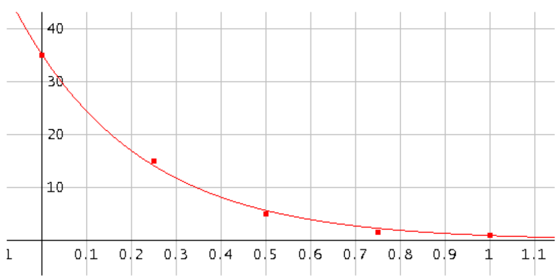
In the SE-column are the squared errors of the model values with respect to the real values. Cell D10 contains the sum of the squared errors which should become (absolutely) minimized (= 0).

Now we see that the SOLVER delivers obviously better approximating functions than we had found earlier. It doesn’t need some calculation to get this insight, just compare the graphs with the graphs on page 32!

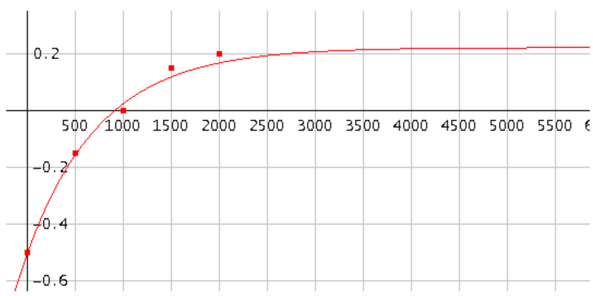
$$0.733 + \frac{317.477}{5.654 + 731.755\cdot e^{-132.95\cdot x}}$$



$$35.199\cdot e^{-3.654\cdot x}$$



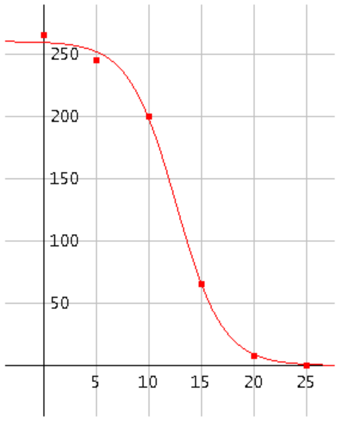
$$0.222 - 0.726\cdot e^{-0.0013\cdot x}$$



Using these functions and according to the strategy of page 30 one can fill in the *Excel* worksheet.

It is no problem to take a time increment of 0.25 years. Calculation is very fast.

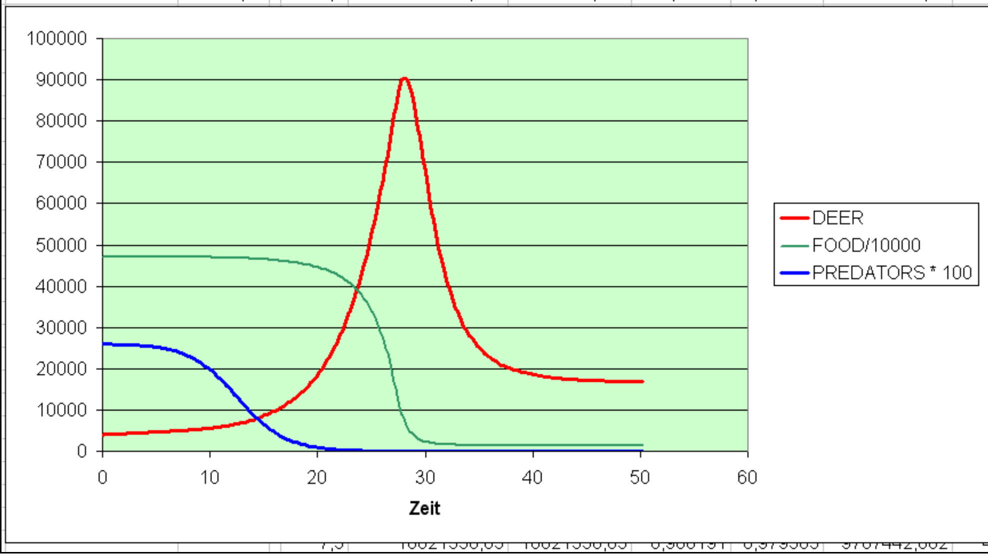
$$\frac{100122}{384.55 + 1.3487 \cdot e^{0.4495 \cdot x}}$$



The peak of the deer population is a little bit shifted but the message of the graph is quite the same as before.

See here a part of the worksheet together with the respective diagram.

Parameters:									
		Time	Food Demand	Grazing L	Veg Dens	Reg Time	Food Incr	FOOD	PREDATORS
Area	320000	0	8000000	8000000	0,979167	0,983177	10171106,17	470000000	259,4484788
Feeding Period	1	0,25	8064887,293	8064887,293	0,980297	0,979123	9658870,723	470542776,5	259,3406929
MaxFoodCap	4,8E+08	0,5	8129828,993	8129828,993	0,981128	0,976157	9279987,698	470941272,4	259,2201949
Ini-Deer	4000	0,75	8194800,688	8194800,688	0,981727	0,974023	9005115,196	471228812,1	259,0854996
Ini-Food	4,7E+08	1	8259780,262	8259780,262	0,982149	0,972522	8810711,041	471431390,7	258,9349519
Dayly Requirement	2000	1,25	8324748,316	8324748,316	0,982436	0,971503	8678181,298	471569123,4	258,7667078
dx	0,25	1,5	8389688,618	8389688,618	0,98262	0,970849	8593009,36	471657481,6	258,5787148
								1708311,8	258,3686886
								1730654,3	258,1340896
								1731390,8	257,8720959
								1715749,7	257,579575
								1687697,4	257,2530525
								1650238,2	256,8886789
								1605644,8	256,4821947
								1555632,4	256,0288926
								71501492	255,5235782
								1444191,2	254,9605298
								1384449,6	254,3334565
								1322796,8	253,6354564
								71259615	252,8589761
								1195171,4	251,9957713
								1129642,2	251,0368717
								1063130,6	249,9725516
								0995679,7	248,7923075
								70927282	247,484848
								0857886,1	246,0380972
								0787401,2	244,4392162
								70715700	242,6746487
								0642620,6	240,7301929
								70567967	238,5911079
								70491509	236,2422586



Bossel poses an interesting question and task:

> *What would have been an appropriate shooting strategy (for the predators) to achieve a stable deer population without causing the collapse of the grazing capacity?*

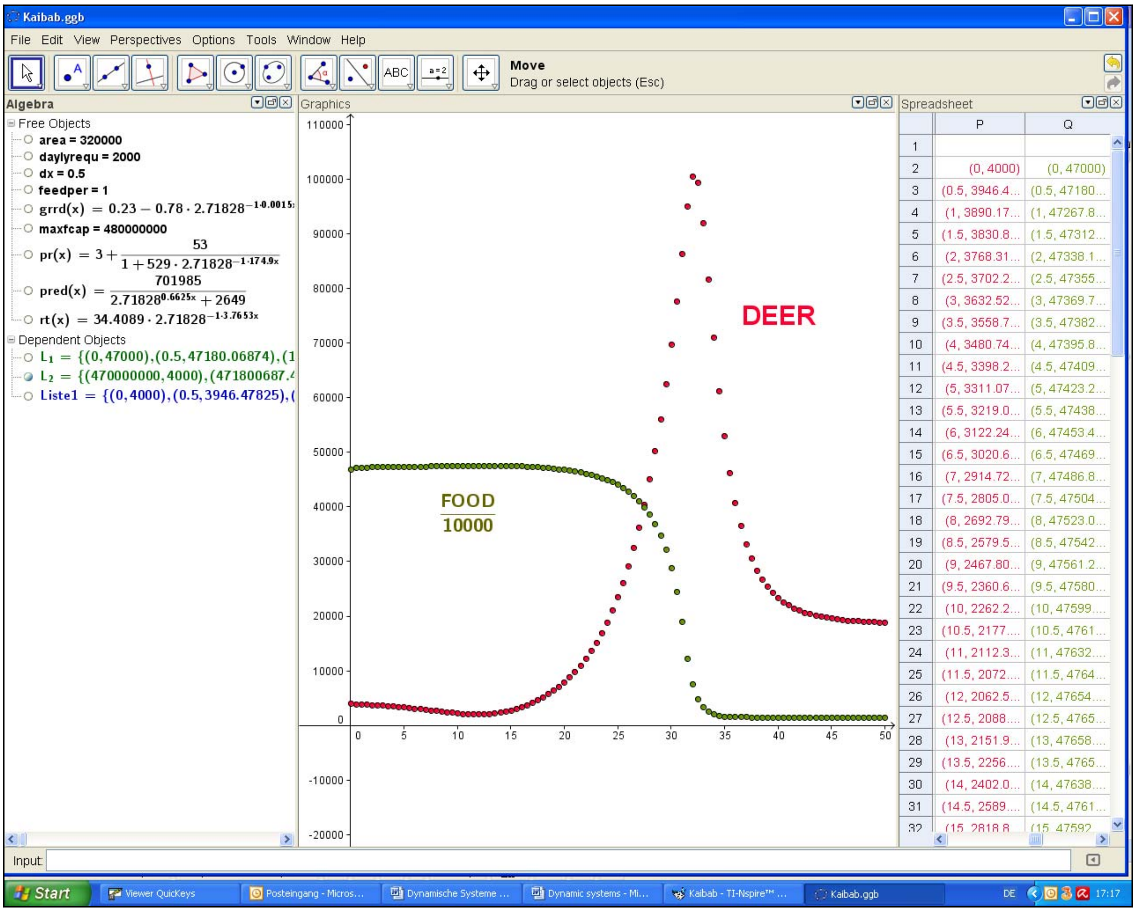
For answering a question like this application of sliders seems to be best suitable. *GeoGebra* and *TI-Nspire* (and *MS Excel*, of course) provide this valuable tool. *DERIVE* does also but we explained earlier why we cannot use the sliders with *DERIVE* in problems like this.

The *GeoGebra* Model

We use again the *DERIVE*-made approximating functions. The first model is set up with the given predator function in order to check whether the results are the expected ones.

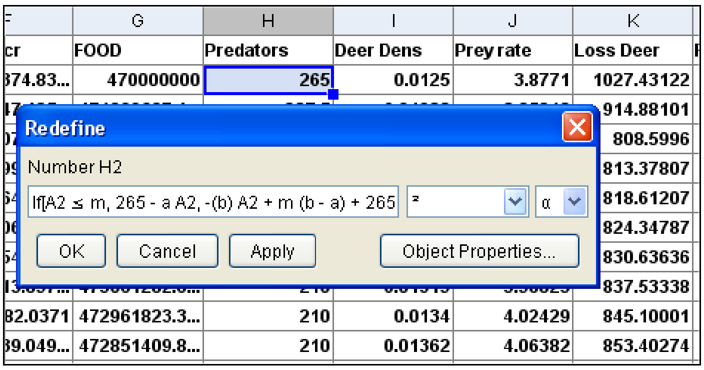
The next screen shot is a copy of the *GeoGebra*-screen with the diagram of the deer population and the scaled food stock (FOOD/10000).

As the *GeoGebra*-spreadsheet needs long calculation times I increased the time step up to 0.5 which does not cause essential changes of the results as the diagram is showing.



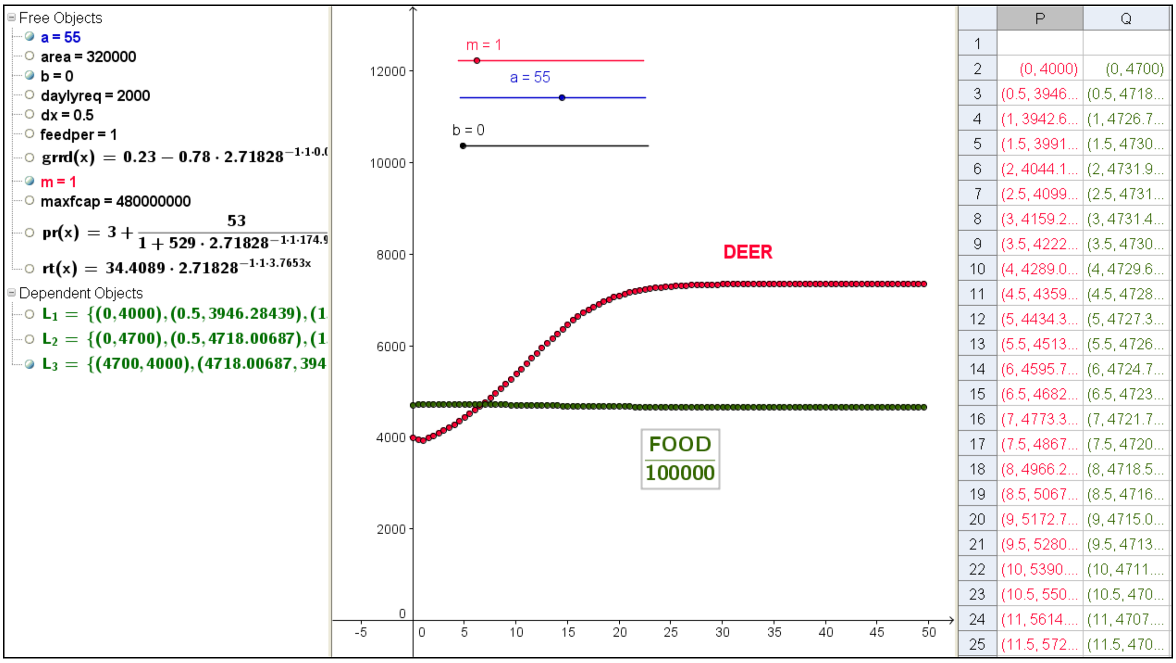
Let's try to find an answer for *Bossel*'s question. After some – exciting – attempts I decided to introduce the following shooting strategy.

I will have a radical shooting of *a* animals annually for the first *m* years followed by reduction to *b* beasts per year. The respective "predator function" is entered in cell H2 (with the corresponding time in cell A2). There is nothing else to change in the spreadsheet from above.



Calculation of the first complete table needs some time but then the diagram is reacting immediately on the change of the parameters by moving the sliders.

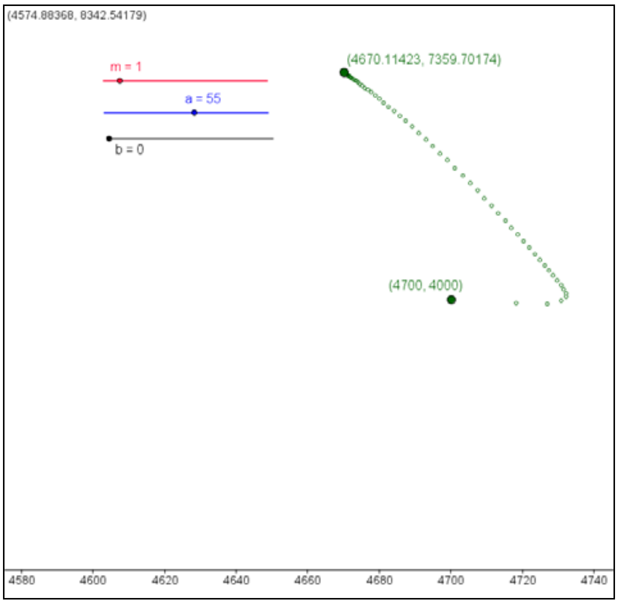
Starting with shooting 55 animals in the first year we can then keep the predator population on a level of 210 in order to achieve constant food supply for the deer. The number of deer stays stable with a stock of about 7360.



The phase diagram Food-Deer shows a significant convergence, too.

But this is not the only one possibility to obtain a stable high deer population.

It makes fun to experiment to reach a more or less stable deer population on a lower or higher level.



You can also introduce a moderate constant shooting rate or any combination. Here we have only an "exogenous" regulation of the predators. But it would also be possible to consider "endogenous" factors like natural dying rate etc and include this in the simulation.

Good Sport!

Working with *TI-NspireCAS*

In the beginning I had some troubles with the spreadsheet application but by and by it worked pretty well finally. The diagram looks the same as the *GeoGebra* graph. Calculation of the table works much faster which makes smaller time increments possible.

Transfer of the *DERIVE*-program into the *TI-NspireCAS*-language is an easy task.

luf(x,pk) is the table function corresponding with the `lu`-function in *DERIVE*:

```
Define luf(x,pk)=
Func
:Local f
:f:=when(pk[1,1]≤x_<pk[2,1],((pk[2,2]−pk[1,2])/(pk[2,1]−pk[1,1]))*(x_−pk[1,1])+pk[1,2],0)
:pk:=subMat(pk,2,1,dim(pk)[1],2)
:While dim(pk)[1]>1
:f:=f+when(pk[1,1]<x_≤pk[2,1],((pk[2,2]−pk[1,2])/(pk[2,1]−pk[1,1]))*(x_−pk[1,1])+pk[1,2],0)
:pk:=subMat(pk,2,1,dim(pk)[1],2)
:EndWhile
:f|x_=x
:EndFunc
```

See the program which provides the respective lists which are the necessary base for the graphic representations.

```
Define kaibab(n,dx)=
Prgm
:Local i,t,deer,food,f_dem,brows_loss,veg_d,deer
:Local predators,reg_time,food_inc,prey_r
:Local loss_deer,food_supply,inc_deer
:i:=1: t:=0
:deer:=ini_deer:food:=ini_food
:ld:={deer}:lf:={food}:ltime:={t}
:While i≤n
:  f_dem:=deer*daily_requ
:  brows_loss:=when(f_dem≥((food)/(feedper)),((food)/(feedper)),f_dem)
:  veg_d:=((food)/(max_food_cap))
:  deer_d:=((deer)/(area))*1.
:  predators:=luf(t,pred)
:  reg_time:=luf(veg_d,regtime)
```

```
:  food_inc:=((max_food_cap−food)/(reg_time))
:  prey_r:=luf(deer_d,preyr)
:  loss_deer:=prey_r*predators
:   food_supply:=((food)/(deer))
:  inc_deer:=luf(food_supply,gr_deer)*deer
:  deer:=deer+(inc_deer−loss_deer)*dx
:  food:=food+(food_inc−brows_loss)*dx
:  t:=t+dx
:  ld:=augment(ld,{deer})
:  lf:=augment(lf,{food})
:  ltime:=augment(ltime,{t})
:  i:=i+1
:EndWhile
:Disp "Deer in ld, scaled Food in lfs, Time in ltime"
:EndPrgm
```

The Calculator-application contains the data and the program call.

$$regtime:=\begin{bmatrix}0 & 35\\ 0.25 & 15\\ 0.5 & 5\\ 0.75 & 1.5\\ 1 & 1\end{bmatrix} \qquad \begin{bmatrix}0.000000 & 35.000000\\ 0.250000 & 15.000000\\ 0.500000 & 5.000000\\ 0.750000 & 1.500000\\ 1.000000 & 1.000000\end{bmatrix}$$

$$gr\_deer:=\begin{bmatrix}0 & -0.5\\ 500 & -0.15\\ 1000 & 0\\ 1500 & 0.15\\ 2000 & 0.2\\ 200000 & 0.2\end{bmatrix} \qquad \begin{bmatrix}0.000000 & -0.500000\\ 500.000000 & -0.150000\\ 1000.000000 & 0.000000\\ 1500.000000 & 0.150000\\ 2000.000000 & 0.200000\\ 200000.000000 & 0.200000\end{bmatrix}$$

$area:=320000:feedper:=1:max\_food\_cap:=4.8\cdot 10^8$ 480000000.000

$daily\_requ:=2000:ini\_food:=4.7\cdot 10^8:ini\_deer:=4000$ 4000.000000

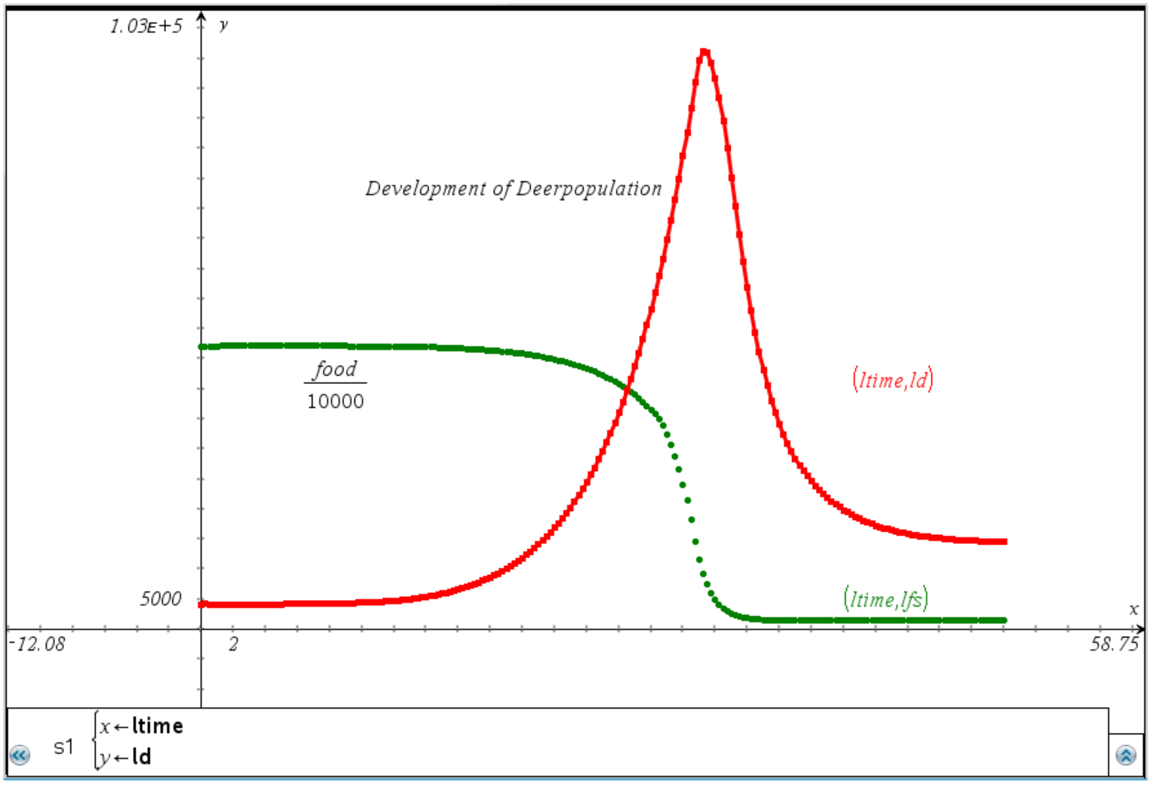
$kaibab\_var(200,0.25)$

Time in ltime, Deer in ldv, Food in lfv_scal, Predators in lprv_scal

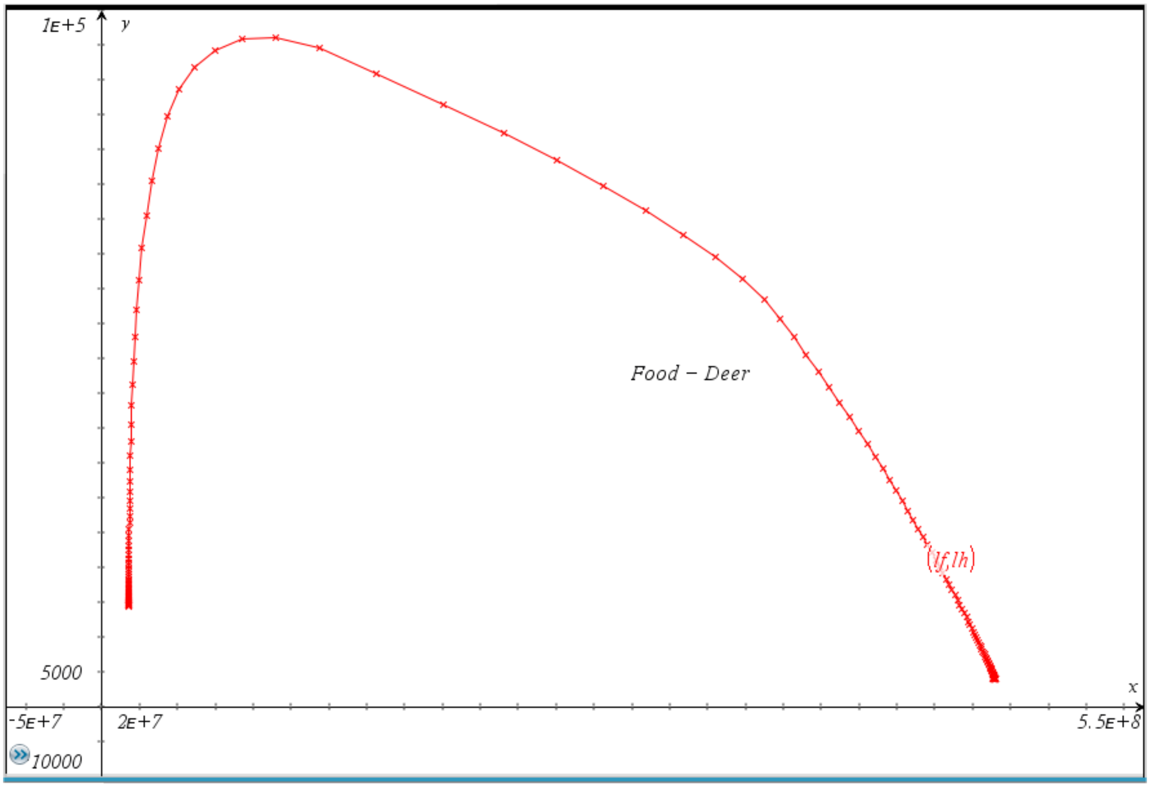
Done

Lists ltime, ld and lfs are the base for the scatter diagrams in the Graphs & Geometry application.

The next screenshot shows the already known development of the deer population together with a representation of the food available in a suitable scaling.



We have seen the phase diagram, too, produced with other tools.

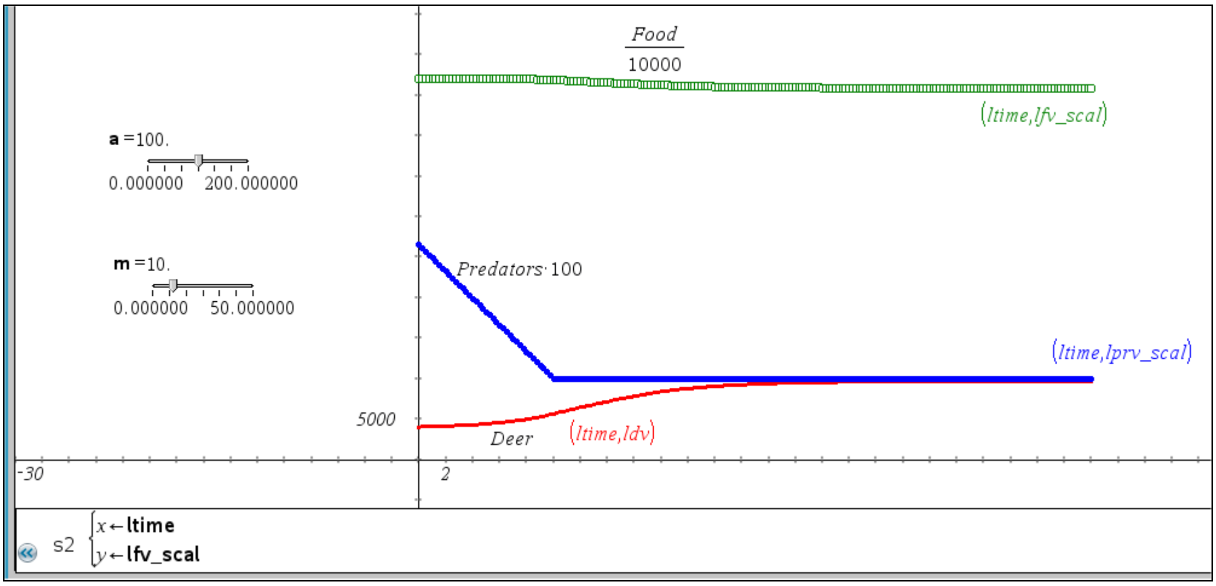


I promised to use the sliders with *TI-Nspire*. So, let us try!

I change the definition of the shooting strategy a little bit. I will keep the shooting numbers constant for the first *m* years until the predator population reaches a certain given number *a*. This number shall be kept stable. I will stick to my "philosophy" and introduce sliders for *m* and *a*. See a part of the program.

```
While i≤n
 f_dem:=deer·daily_requ
 brows_loss:=when(f_dem≥food/feedper, food/feedper, f_dem)
 veg_d:=food/max_food_cap
 deer_d:=deer/area·1.
 reg_time:=luf(veg_d,regtime)
 food_inc:=(max_food_cap−food)/reg_time
 prey_r:=luf(deer_d,preyr)
 loss_deer:=prey_r·predators
 food_supply:=food/deer
 inc_deer:=deer·luf(food_supply,gr_deer)
 deer:=deer+(inc_deer−loss_deer)·dx
 food:=food+(food_inc−brows_loss)·dx
 t:=t+dx
 predators:=when(x≤m,(a−265)/m·x+265,a)|x=t
 ldv:=augment(ldv,{deer}):  lfv:=augment(lfv,{food})
 lprv:=augment(lprv,{predators}):  ltime:=augment(ltime,{t})
 i:=i+1
EndWhile
lfv_scal:=lfv/10000:lprv_scal:=lprv·100
Disp "Time in ltime, Deer in ldv, Food in lfv_scal, Predators in lprv_scal"
EndPrgm
```

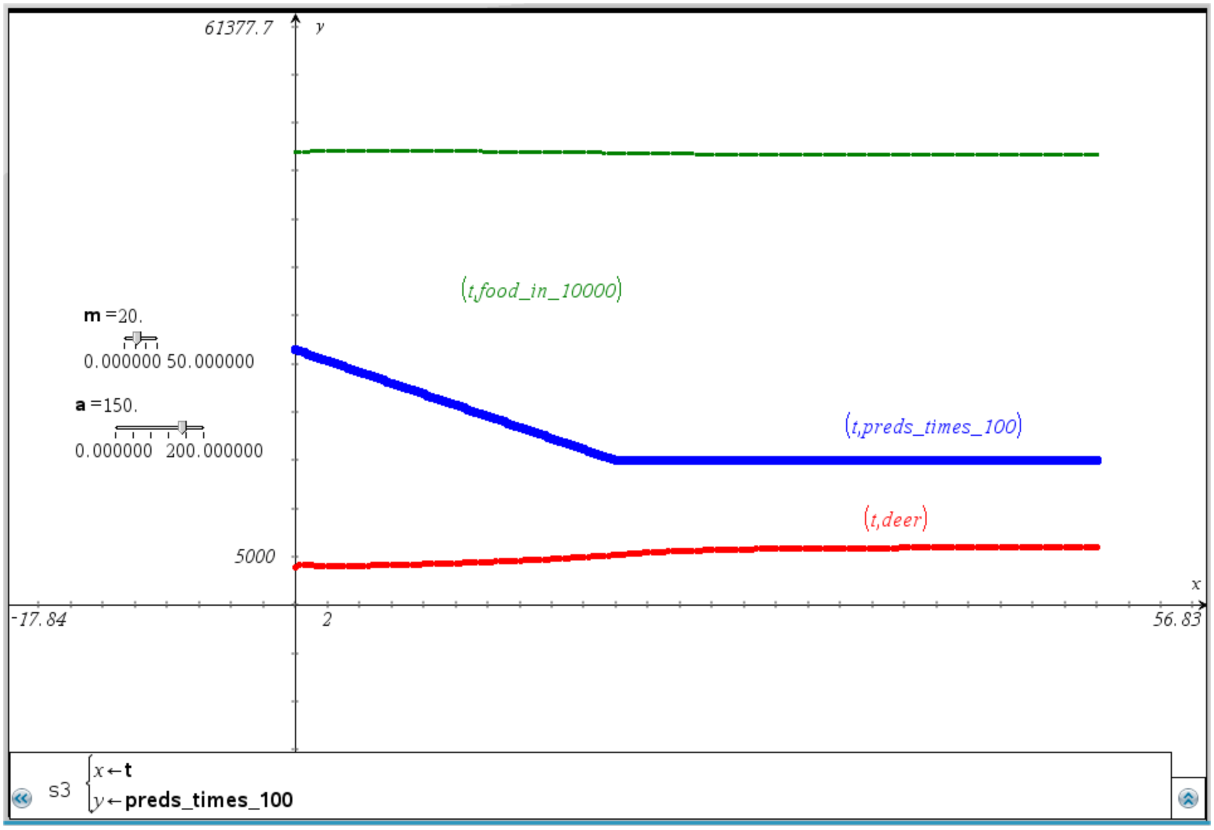
The predator function is given under *predators*:=.



Working with sliders using the program has the disadvantage that after every change of the parameters (moving the sliders) the program must be run again. Then the diagram is adapted immediately.

It is much more comfortable to use the Spreadsheet application. The graphs are manipulated directly moving the sliders. The Spreadsheet is not presented here.

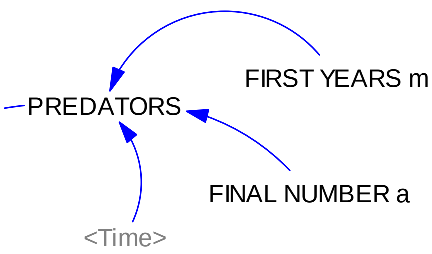
You can see two – of many others – ways to reach *Bossel*'s goal with different stable deer populations. According to the graph below we should decrease the predator stock within the first 20 years linearly down to a number of 150 and then keep this number of predators for the future.



In order to end the cycle I'd like to implement this solution into the VENSIM-model:

The stock and flow diagram changes as show at the right.

The respective new dwfinitions of the PREDATORS and the parameters *m* and *a* are given belosw.

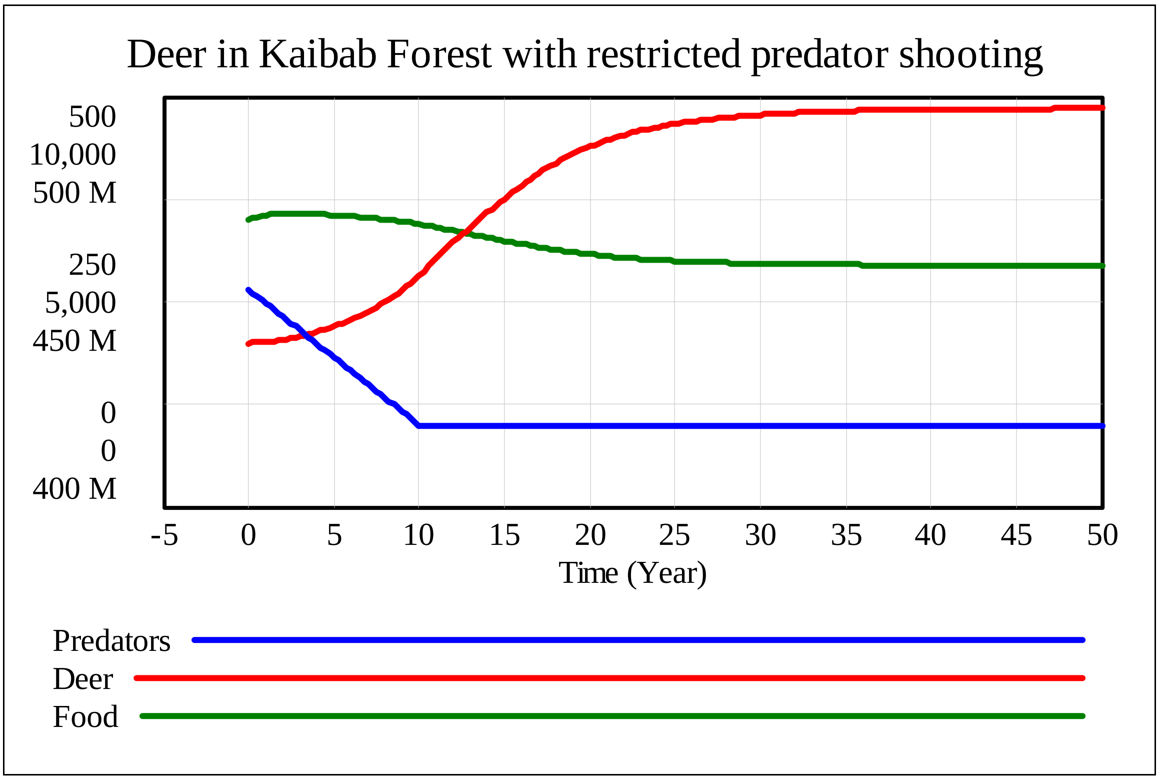


FINAL NUMBER a = 100

FIRST YEARS m = 10

PREDATORS = IF THEN ELSE(Time<=FIRST YEARS m , (FINAL NUMBER a-265)/ FIRST YEARS m*Time +265 , FINAL NUMBER a)

The *VENSIM* plot shows the expected results.



I mentioned in an earlier note my intention to try an approach via the numerical solution of the respective system of differential equations. Voila, it works as you can see in the following.

The ecological catastrophe as a system of differential equations

The form of the differential equations can be derived directly from the *VENSIM*-equations:

$$\frac{dd}{dt} = d \cdot gr\_deer\_f\left(\frac{f}{d}\right) - preyr\_f\left(\frac{d}{area}\right) \cdot pred\_f(t)$$

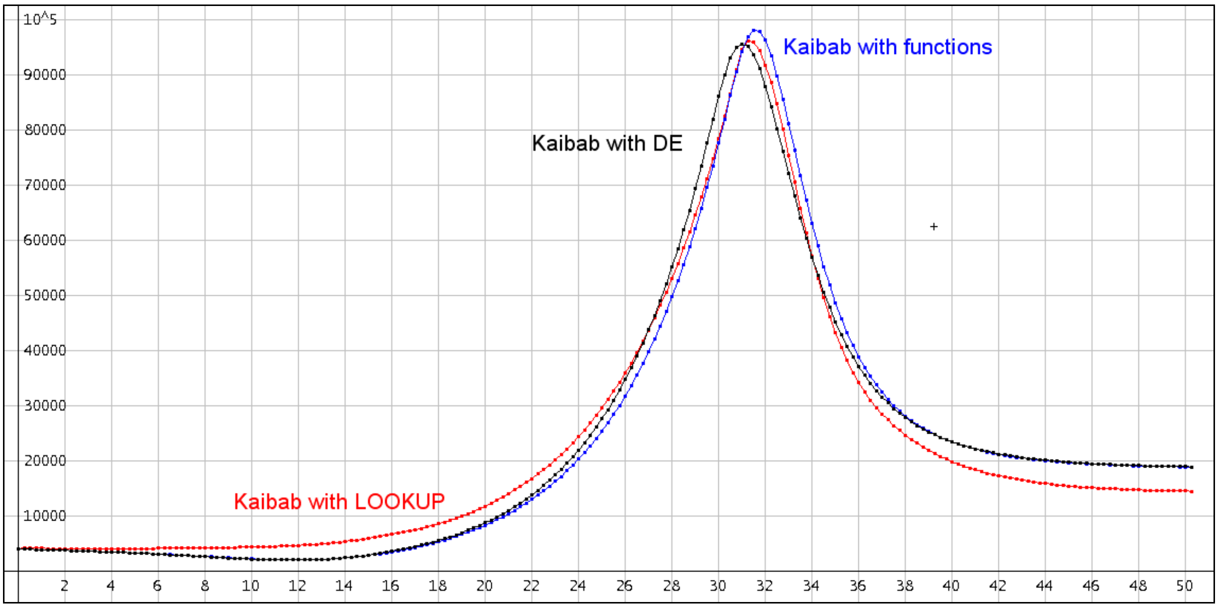
$$\frac{df}{dt} = \frac{max\_food\_cap - f}{regtime\_f\left(\frac{f}{\max\_food\_cap}\right)} - graz\_loss(dayly\_requ \cdot d)$$

I am using the Runge-Kutta-routine of *DERIVE* again.

$$\left(\mathrm{RK}\left(\left[gr\_deer\_f\left(\frac{f}{d}\right)\cdot d - preyr\_f\left(\frac{d}{area}\right)\cdot pred\_f(t),\ \frac{max\_food\_cap - f}{regtime\_f\left(\frac{f}{max\_food\_cap}\right)} - graz\_loss(dayly\_requ\cdot d,\ f)\right],\ [t,\ d,\ f],\ [0,\ ini\_deer,\ ini\_food],\ 0.25,\ 201\right)\right)\downarrow\downarrow[1,\ 2]$$

It would be possible to apply the LOOKUP-routines but RK cannot work through all 201 rows of the table in one step.

Selection of the first and second column shows rise and fall of the deer population:



The plot displays all deer-plots and allows comparison..

This model fascinated me indeed, because it offers so many opportunities for treatment. Description of the piecewise defined functions (WITH LOOKUP) by one single function requires some fantasy and knowledge about possible function types. There is no "right" answer and this can be stated about most of modelling problems.

Comparing between applying a program (which must be written in advance) and spreadsheet is charming and exciting as well.

Use of sliders offers an important additional quality and provokes again interpreting the results.

Besides the mathematical point of view this model is demonstrating once more how an intervention in natural procedure (even if in best intention) can destroy the balance of environment and can result in unforeseen consequences.

Additional reference:

G. Ossimitz, *Materialien zur Systemdynamik*, hpt 1990

The problems covered in the full paper are:

Tourism and Environment

Predator and Prey (times 2)

Collapse of an Ecological System

Population Dynamics

The Reservoir is flowing over

Michaelis-Menten-Kinetics

What is a Brusselator?

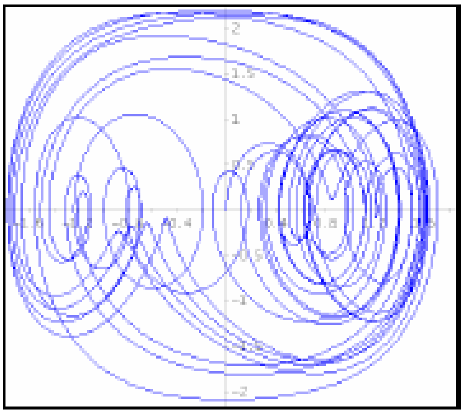
Bistable Oscillator

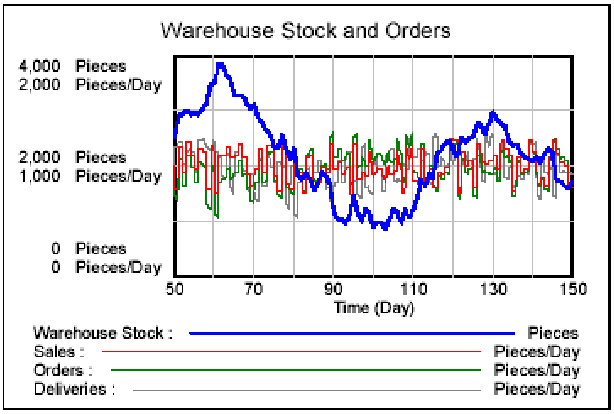
Stock-keeping with random numbers

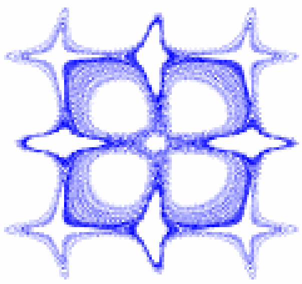
Rössler Attractor

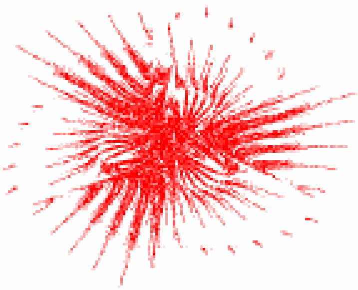
Gumowski-Mira – another "attractive Attractor"

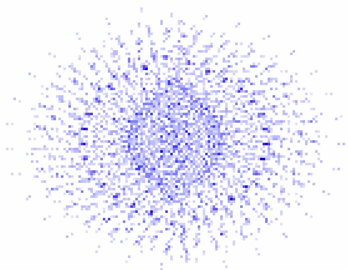
A small selection of graphs which can be found in the full paper:











Steiner Point, Steiner Circles and Triangle of Polar Lines

Peter Lüke-Rosendahl, Germany

Satz vom Steinerpunkt[1]

Sind *a*, *b*, *c*, *d* vier Gerade einer Ebene, so laufen die Umkreise der vier Dreiecke *abc*, *abd*, *acd*, *bcd* durch einen gemeinsamen Punkt, den so genannten Steinerpunkt.

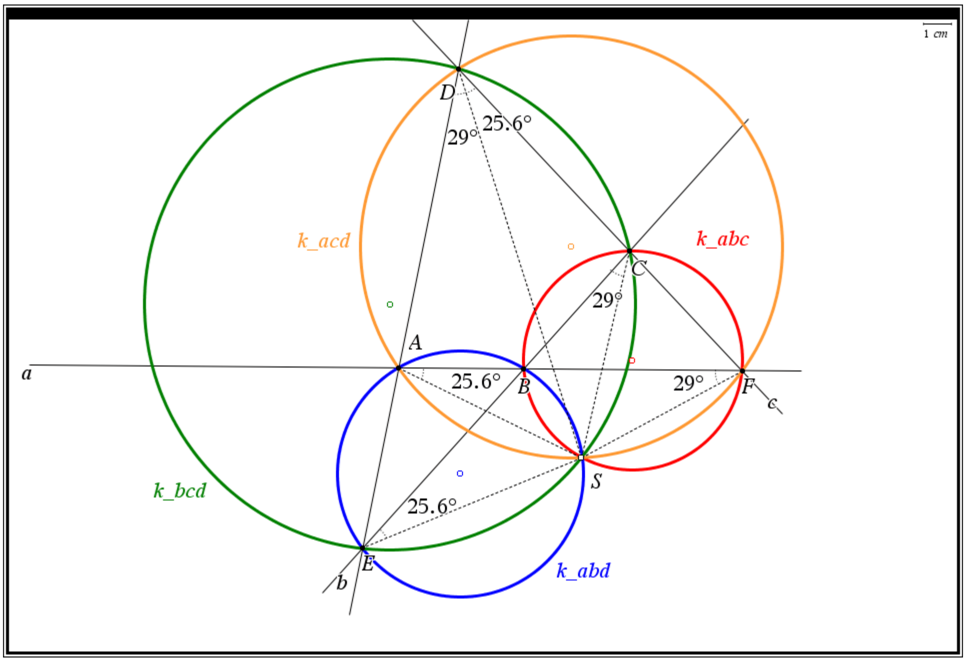
Die Umkreise werden Steinerkreise genannt. Die Mittelpunkte der vier Steinerkreise und der Steinerpunkt des entstehenden Vierseits liegen auf einem gemeinsamen Kreis – dem Steinerkreis

Theorem of the Steiner Point[1]

Given are four lines *a*, *b*, *c*, *d* in the plane. The circumcircles of the triangles *abc*, *abd*, *acd*, *bcd* have one point in common, the so called Steiner point.

The circum circles are called Steiner circles. The Steiner point and the circumcentres are lying on one common circle – the Steiner circle.

We will plot the situation on a TI-NspireCAS Geometry page containing all circumcircles together with their centres and some segments which are important for the geometric proof of the first part of the theorem. We don't plot the final circle at the moment.



$\angle SFB = \angle SDA$ (periphery or inscribed angles of circle *k_abd*) and $\angle SCB = \angle SDA$ (periphery or inscribed angles of circle *k_bcd*) $\rightarrow \angle SFB = \angle SCB \rightarrow S$ lies on the circumcircle of ΔBFC.

$\angle SAB = \angle SDF$ (periphery or inscribed angles of circle *k_acd*) and $\angle SEB = \angle SDF$ (periphery or inscribed angles of circle *k_abc*) $\rightarrow \angle SAB = \angle SEB \rightarrow S$ lies on the circumcircle of ΔABE.

The first part of theorem is proved.

Now let's proof the theorem by using CAS. This might be a good occasion for students after verifying the theorem with a Dynamic Geometry tool to apply their knowledge in analytic geometry and in applying their CAS as well (DERIVE or TI-NspireCAS or Maxima or WIRIS or …).

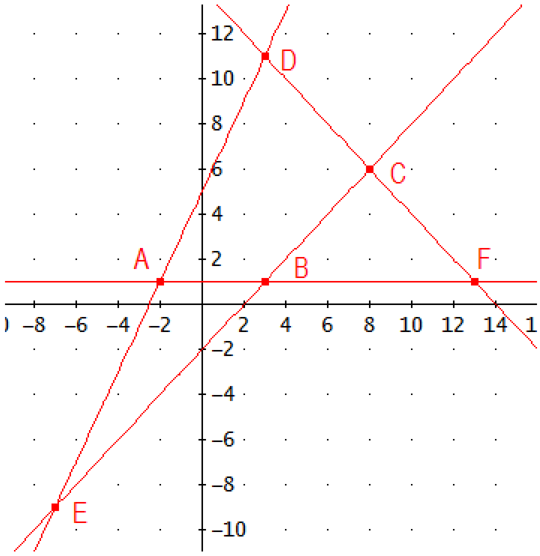
First of all we prepare all functions for later use:

```
#1:   @ auxiliary functions for later calculations

#2:   [CaseMode := Sensitive, InputMode := Word, U := [xU, yU], V := [xV, yV], W := [xW, yW]]

                      U + V
#3:   Mp(U, V) := ───────
                        2

#4:   @ Perp(U,V) : direction vector of perpendicular line wrt to segment UV

      Perp(U, V) := [- (V - U) , (V - U) ]
#5:                           2          1

#6:   @ l(U, V, s) : parameter form of line passing points U and V

#7:   l(U, V, s) := U + s·(V - U)

#8:   @ Isect(g1, g2,p,q) : position vector to intersection point

#9:   @ of lines l1 and l2 - given in parameter form

#10:  Isect(g1, g2, p, q) := SUBST(g1, p, FIRST(FIRST(SOLUTIONS(g1 = g2, [p, q]))))

#11:  @ Pb(U, V, s) : parameter form of perpendicular bisector of segment UV

#12:  Pb(U, V, s) := Mp(U, V) + s·Perp(U, V)

#13:  @ Ccc(U, V, W, p, q) : centre of circum circle of triangle UVW

#14:  Ccc(U, V, W, p, q) := Isect(Pb(U, V, p), Pb(V, W, q))

#15:  @ Ccr(U, V, W, p, q) : radius of circumcircle of triangle UVW

#16:  Ccr(U, V, W) := |U - Ccc(U, V, W, p, q)|

#17:  @ Cc(U, V, W,p,q) : equation of circumcircle of triangle UVW

                                                         2                2
#18:  Cc(U, V, W, p, q) := ([x, y] - Ccc(U, V, W, p, q))  = Ccr(U, V, W)

#19:  @ ===============================================================================
```

We could immediately start with the general proof. But for students it seems to be more appropriate to work at first stepwise with numerical data points. We fix four points, their lines forming the quadrilateral followed by calculating the missing intersection points E and F forming the complete quadrilateral.

```
#20:  @ Example:

#21:  [A := [-2, 1], B := [3, 1], C := [8, 6], D := [3, 11]]

#22:  @ the given four lines:

#23:  [l(A, B), l(B, C), l(C, D), l(D, A)]

#24:  @ the missing intersection points E and F are:

#25:  (E := Isect(l(A, D, p), l(B, C, q))) = E := [-7, -9]

#26:  (F := Isect(l(A, B, p), l(D, C, q))) = F := [13, 1]
```

This should be the first time to plot the given situation:



Using again our provided functions it is easy to find the "Steiner point" as intersection point of the circumcircles:

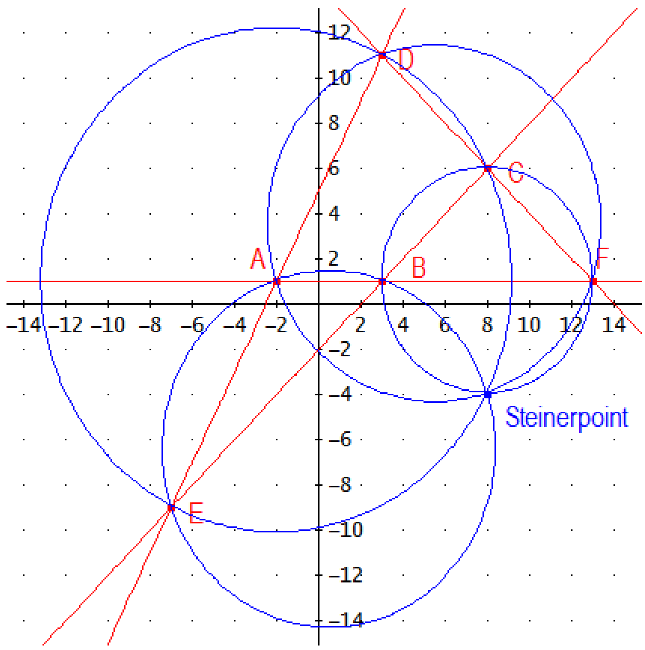
#28: [cc1 := Cc(B, C, F), cc2 := Cc(A, B, E), cc3 := Cc(F, A, D), cc4 := Cc(C, D, E)]

#29: SOLUTIONS(cc1 ∧ cc2, [x, y]) = $\begin{bmatrix} 3 & 1 \\ 8 & -4 \end{bmatrix}$

#30: SOLUTIONS(cc3 ∧ cc4, [x, y]) = $\begin{bmatrix} 3 & 11 \\ 8 & -4 \end{bmatrix}$

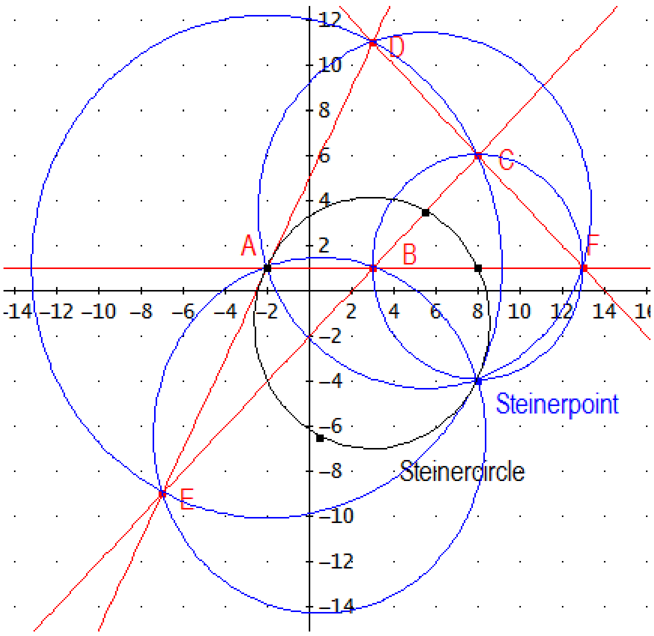
#31: @ so [8,-4] should be the desired Steiner point!

#32: S := [8, -4]



What's the next step? Plotting the circum centres followed by calculating their circumcircle and being happy observing it passing point S. We check if C4 (see below) and S are on the circumference of the Steiner circle.

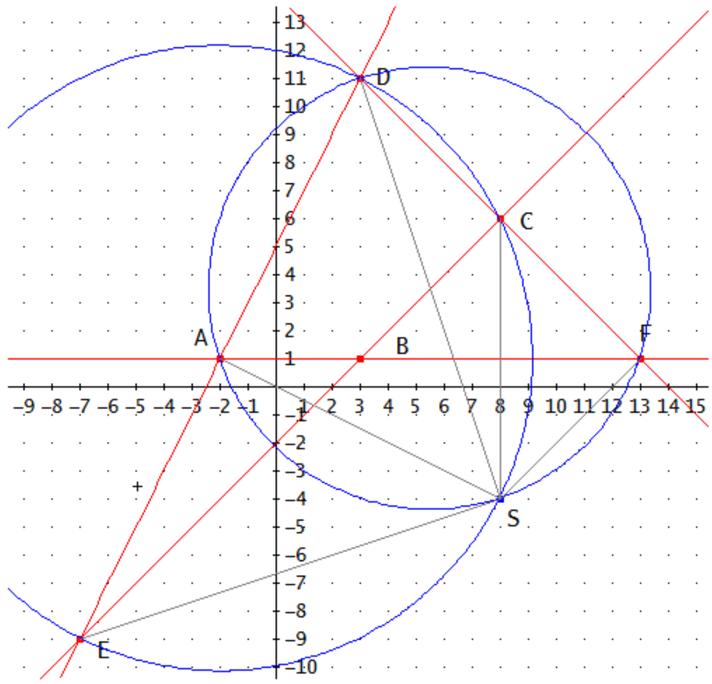
```
#38:  @ the four centres of the circumcircles are:

#39:  [C1 := Ccc(B, C, F), C2 := Ccc(A, B, E), C3 := Ccc(F, A, D), C4 := Ccc(C, D, E)]

#40:  Sc := Cc(C1, C2, C3)

#41:  SUBST(Sc, [x, y], C4) = true

#42:  SUBST(Sc, [x, y], S) = true
```



In the TI-NspireCAS plot (and with Geogebra as well) we could identify the triplets of angles of same size. Two questions for students might arise:

(1) Are there more than two triplets in the complete figure?

(2) How to check the equal angles with DERIVE (or Maxima or WIRIS or …)



In the figure above you can see the respective segments. Calculation is easy done:

< SFB = < SDA = < SCB

$$\#34: \left[\frac{\mathrm{ACOS}\left(\frac{(S-F)\cdot(B-F)}{|S-F|\cdot|B-F|}\right)}{1°}, \frac{\mathrm{ACOS}\left(\frac{(S-D)\cdot(A-D)}{|S-D|\cdot|A-D|}\right)}{1°}, \frac{\mathrm{ACOS}\left(\frac{(S-C)\cdot(B-C)}{|S-C|\cdot|B-C|}\right)}{1°}\right]$$

#35: [45, 45, 45]

< SAB = < SEB = < SDF

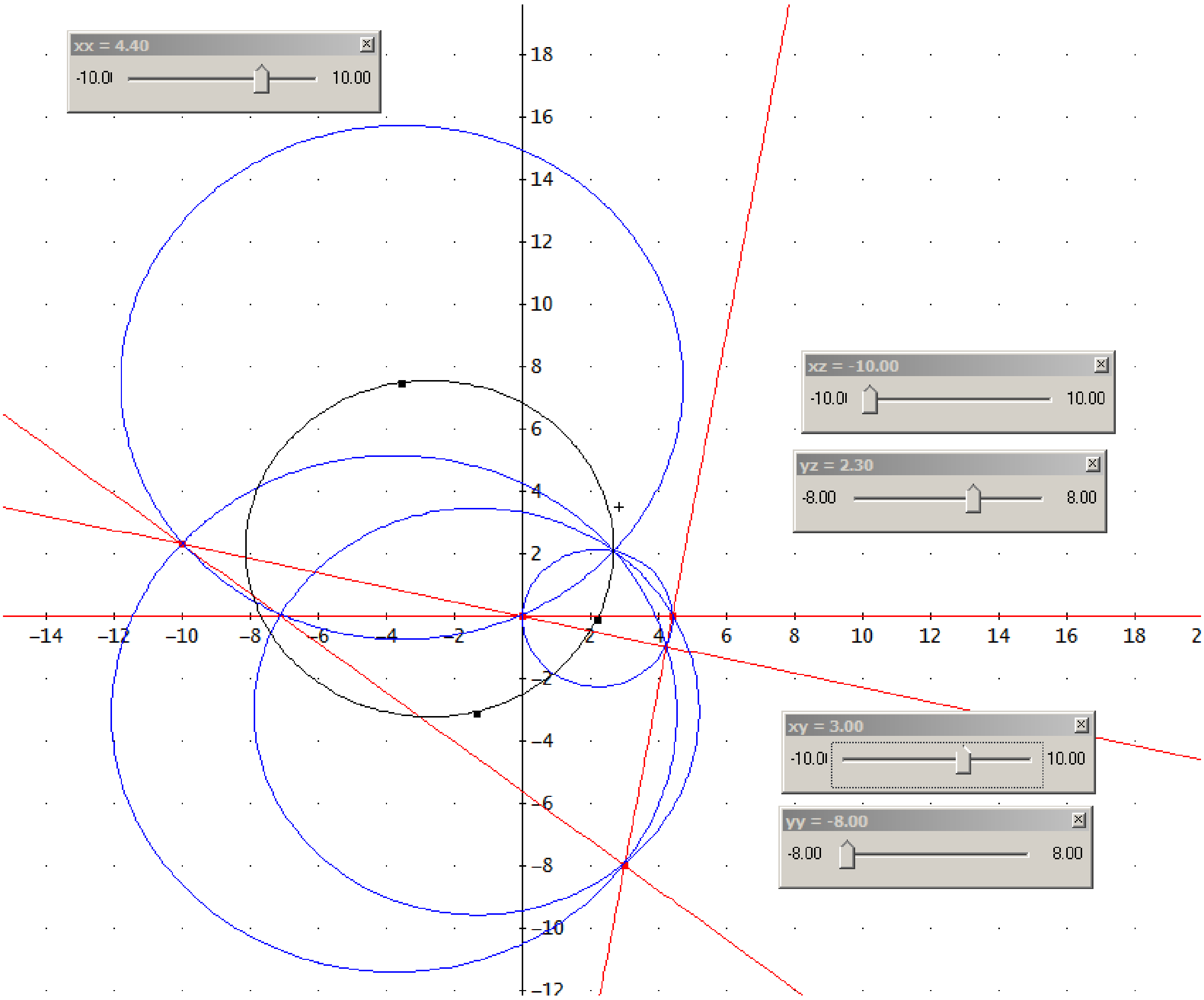
$$\#36: \left[\frac{\mathrm{ACOS}\left(\frac{(A-S)\cdot(A-B)}{|A-S|\cdot|A-B|}\right)}{1°}, \frac{\mathrm{ACOS}\left(\frac{(E-S)\cdot(E-B)}{|E-S|\cdot|E-B|}\right)}{1°}, \frac{\mathrm{ACOS}\left(\frac{(D-S)\cdot(D-F)}{|D-S|\cdot|D-F|}\right)}{1°}\right]$$

#37: [26.56505117, 26.56505117, 26.56505117]

Peter and I had an extended and fruitful email-communication regarding his great papers. I wrote that I'd like to simulate Dynamic Geometry with DERIVE by introducing sliders for the coordinates of the points. This is what he answered:

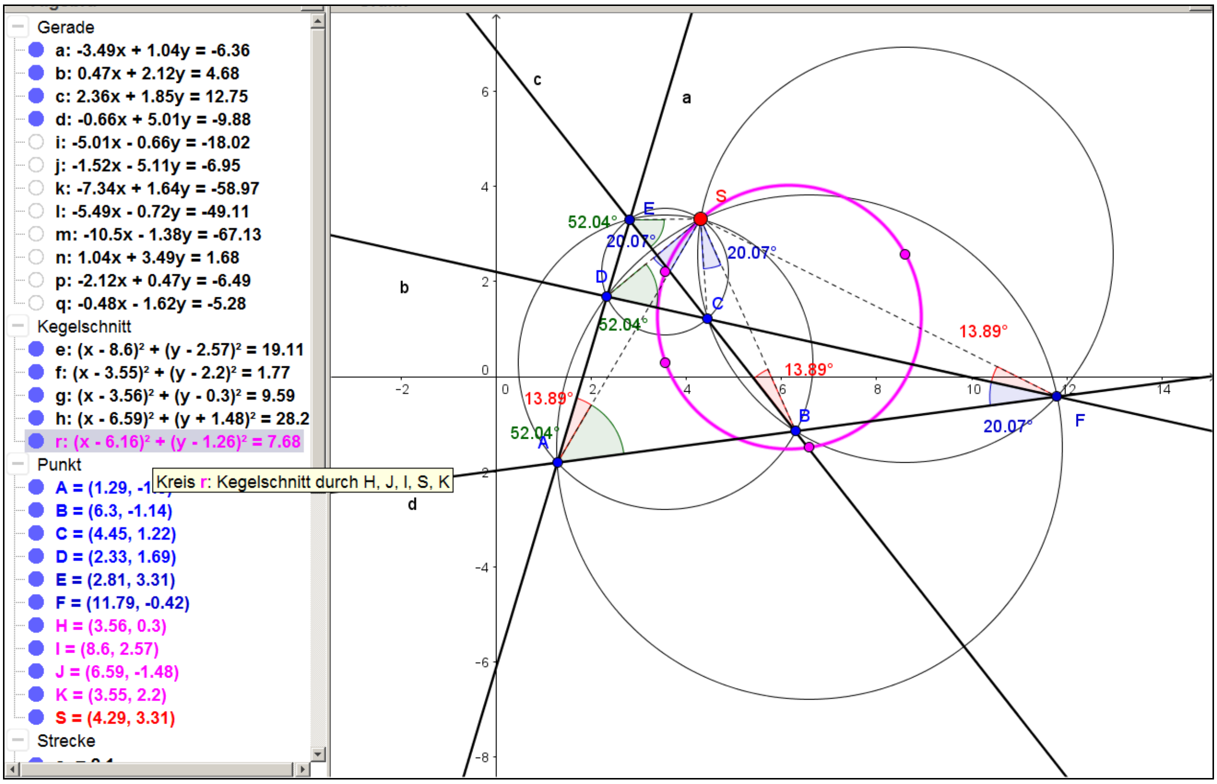
"*Since our last two or three mails I have been waiting for your so beloved sliders.*"

Well, always your servant, here they are:



With TI-NspireCAS I can plot the locus of the four circumcircles and S as a conic determined by five points, but I don't receive its equation. So you have to add the Steiner circle in the TI-Nspire graph in your imagination – its no problem at all.

GeoGebra does a little bit more: it provides the equation of the generated conic and identifies it as a circle (see below). Josef



Needless to say that the proof using analytic geometry with TI-NspireCAS doesn't cause any problems and runs more or less the same way as with DERIVE.

The proof of the second part of the theorem is not so comfortable. You can find it in [1].

Final comment: When I started preparing this DNL-issue I wanted to find some additional information about the Steiner point and undertook an internet research – but unfortunately there was nothing to find. I asked Peter about his resources. He gave [1] as reference and sent two scans with "Steinerpunkt" and "Steinerkreise". Then I wrote to Michael deVilliers, a geometry expert – now prof. emeritus living in South Africa. This is his answer:

Dear Josef

Sorry, only got this message this evening as I'm retired since 31 Jan this year and don't regularly look at my university e-mail.

As far as I can recall this result is known as a theorem of Miquel, and the point of concurrency as the Miquel point of a complete quadrangle. See for example:
http://www.cut-the-knot.org/ctk/CompleteQuadrilateral.shtml
where this result is mentioned in no. 4.

At Wikipedia at: https://en.wikipedia.org/wiki/Miquel%27s_theorem the following is said about the result:

Miquel and Steiner's quadrilateral theorem
The circumcircles of all four triangles of a complete quadrangle meet at a point M.[6] In the diagram above these are ΔABF, ΔCDF, ΔADE and ΔBCE.
This result was announced, in two lines, by Jakob Steiner in the 1827/1828 issue of Gergonne's Annales de Mathématiques,[7] but a detailed proof was given by Miquel.

So it seems more credit should maybe go to Miquel?

However, the situation is analogous to the Fermat-Torricelli-Steiner point of a triangle which is called the Fermat point in France, the Torricelli point in Italy and the Steiner point in Germany! Priority in terms of proposing the problem lies with Fermat, but Torricelli first produced a proof of the result, and much later Steiner (and others too).

Hope that helps?

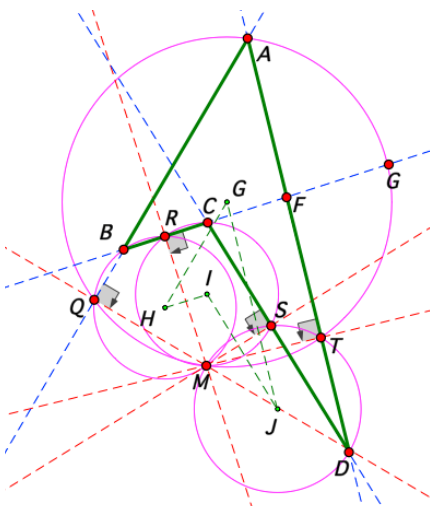
Regards, Michael

P.S. Forgot to mention: the easiest way to prove the result is by applying Miquel's theorem for a triangle to each triangle of the complete quadrangle.

And two days later there was another mail in my mailbox arriving from South Africa:

Dear Josef
Thought you might also be interested in a converse-like result I recently wrote about regarding Miquel. We can formulate the following variation of Miquel (Steiner) for 4 drawn lines:

> From an arbitrary point M construct equi-inclined lines to the lines forming quadrangle ABCD as shown below in attached pic. Then AQMT, BRMQ, CSMR and DTMS are cyclic and their circumcircles are concurrent in M, and GHIJ is similar to ABCD. This is shown in the attached pic for perpendicular lines drawn from M to the 4 lines.[2]



I'm also attaching my paper below. There is also an interactive Java sketch available at:
http://dynamicmathematicslearning.com/miquel-variation.html
Hope you and your colleagues enjoy.

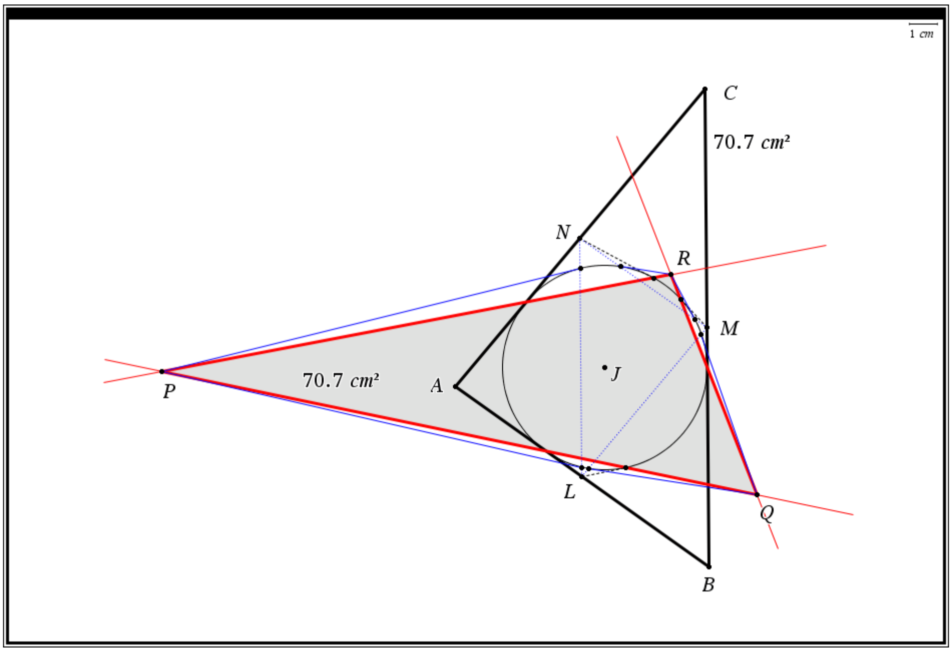
Satz von den Polaren des Inkreises[1]

Das von den Polaren der Seitenmitten eines Dreiecks *ABC* für den Inkreis dieses Dreiecks gebildete Polarendreieck *PQR* ist dem Ausgangsdreieck inhaltsgleich.

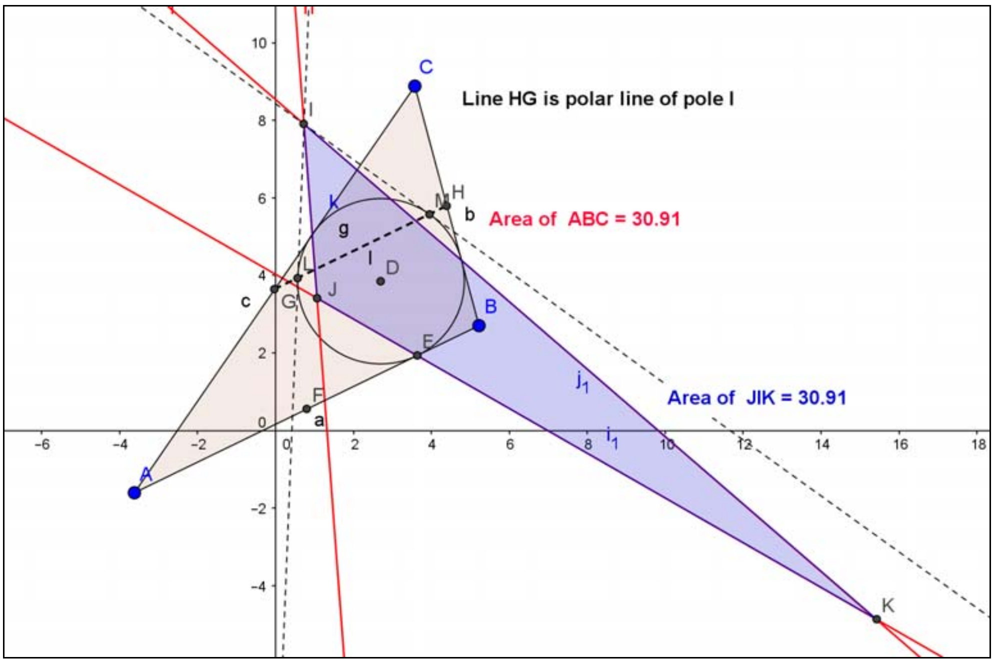
Theorem of the Polar Lines of the Incircle[1}

The polar lines of the midpoints of a triangle *ABC* with respect to its incircle form a triangle *PQR*. Both triangles have the same area.

Let us start with a Dynamic Geometry model. We can drag the vertices of the given triangle (the black one) and observe the red one together with the area of both triangles.



You can see the TI-Nspire Geometry page (above) and a Geogebra view (below).



I will present the TI-NspireCAS procedure first performed on a Notes page:

$\mathbf{p1}:=[xa\ \ ya]:\ \mathbf{p2}:=[xb\ \ yb]:\ \mathbf{p3}:=[xc\ \ yc]:\mathbf{p0}:=[xx\ \ yy]:\mathbf{pc}:=[cx\ \ cy]$ ▸ $[cx\ \ cy]$

midpoint of segment (P1,P2): $\mathbf{m}(\mathbf{p1},\mathbf{p2}):=\dfrac{\mathbf{p1}+\mathbf{p2}}{2}$ ▸ *Done*

line (P1,P2): $\mathbf{l}(\mathbf{p1},\mathbf{p2}):=\dfrac{\mathbf{p1}[1,2]-\mathbf{p2}[1,2]}{\mathbf{p1}[1,1]-\mathbf{p2}[1,1]}\cdot(x-\mathbf{p1}[1,1])+\mathbf{p1}[1,2]$ ▸ *Done*

length of segment (P1,P2): $\mathbf{le}(\mathbf{p1},\mathbf{p2}):=\mathrm{norm}(\mathbf{p2}-\mathbf{p1})$ ▸ *Done*

half perimeter of triangle (P1,P2,P3): $\mathbf{s}(\mathbf{p1},\mathbf{p2},\mathbf{p3}):=\dfrac{\mathbf{le}(\mathbf{p1},\mathbf{p2})+\mathbf{le}(\mathbf{p2},\mathbf{p3})+\mathbf{le}(\mathbf{p1},\mathbf{p3})}{2}$ ▸ *Done*

area of triangle (P1,P2,P3):
$\mathbf{area}(\mathbf{p1},\mathbf{p2},\mathbf{p3}):=\sqrt{\mathbf{s}(\mathbf{p1},\mathbf{p2},\mathbf{p3})\cdot(\mathbf{s}(\mathbf{p1},\mathbf{p2},\mathbf{p3})-\mathbf{le}(\mathbf{p1},\mathbf{p2}))\cdot(\mathbf{s}(\mathbf{p1},\mathbf{p2},\mathbf{p3})-\mathbf{le}(\mathbf{p1},\mathbf{p3}))\cdot(\mathbf{s}(\mathbf{p1},\mathbf{p2},\mathbf{p3})-\mathbf{le}(\mathbf{p2},\mathbf{p3}))}$
▸ *Done*

radius of incircle of triangle (P1,P2,P3): $\mathbf{r}(\mathbf{p1},\mathbf{p2},\mathbf{p3}):=\dfrac{\mathbf{area}(\mathbf{p1},\mathbf{p2},\mathbf{p3})}{\mathbf{s}(\mathbf{p1},\mathbf{p2},\mathbf{p3})}$ ▸ *Done*

point on angle bisector of angle (P1,P2,P3): $\mathbf{pab}(\mathbf{p1},\mathbf{p2},\mathbf{p3}):=\mathbf{p2}+\dfrac{\mathbf{p1}-\mathbf{p2}}{\mathbf{le}(\mathbf{p1},\mathbf{p2})}+\dfrac{\mathbf{p3}-\mathbf{p2}}{\mathbf{le}(\mathbf{p2},\mathbf{p3})}$ ▸ *Done*

angle bisector of angle (P1,P2,P3): $\mathbf{ab}(\mathbf{p1},\mathbf{p2},\mathbf{p3}):=\mathbf{l}(\mathbf{p2},\mathbf{pab}(\mathbf{p1},\mathbf{p2},\mathbf{p3}))$ ▸ *Done*

incentre_aux: inc(P1,P2,P3): $\mathbf{inc}(\mathbf{p1},\mathbf{p2},\mathbf{p3}):=\mathrm{linSolve}\left(\begin{cases}y=\mathbf{ab}(\mathbf{p1},\mathbf{p2},\mathbf{p3})\\ y=\mathbf{ab}(\mathbf{p2},\mathbf{p3},\mathbf{p1})\end{cases},\{x,y\}\right)$ ▸ *Done*

I believe that the comments will guide you through the calculation process.

incentre_aux: inc(P1,P2,P3): $\mathbf{inc}(\mathbf{p1},\mathbf{p2},\mathbf{p3}):=\mathrm{linSolve}\left(\begin{cases}y=\mathbf{ab}(\mathbf{p1},\mathbf{p2},\mathbf{p3})\\ y=\mathbf{ab}(\mathbf{p2},\mathbf{p3},\mathbf{p1})\end{cases},\{x,y\}\right)$ ▸ *Done*

incentre: icc(P1,P2,P3): $\mathbf{icc}(\mathbf{p1},\mathbf{p2},\mathbf{p3}):=[\mathbf{inc}(\mathbf{p1},\mathbf{p2},\mathbf{p3})[1]\ \ \mathbf{inc}(\mathbf{p1},\mathbf{p2},\mathbf{p3})[2]]$ ▸ *Done*

polar line of pole P0 wrt to circle (C,r_):

$\mathbf{pl}(\mathbf{p0},\mathbf{pc},r\_):=(\mathbf{p0}[1,1]-\mathbf{pc}[1,1])\cdot(x-\mathbf{pc}[1,1])+(\mathbf{p0}[1,2]-\mathbf{pc}[1,2])\cdot(y-\mathbf{pc}[1,2])-r\_^2$ ▸ *Done*

the polar lines are:

$\mathbf{pl1}(\mathbf{p1},\mathbf{p2},\mathbf{p3}):=\mathbf{pl}(\mathbf{m}(\mathbf{p1},\mathbf{p2}),\mathbf{icc}(\mathbf{p1},\mathbf{p2},\mathbf{p3}),\mathbf{r}(\mathbf{p1},\mathbf{p2},\mathbf{p3}))$ ▸ *Done*

$\mathbf{pl2}(\mathbf{p1},\mathbf{p2},\mathbf{p3}):=\mathbf{pl}(\mathbf{m}(\mathbf{p1},\mathbf{p3}),\mathbf{icc}(\mathbf{p1},\mathbf{p2},\mathbf{p3}),\mathbf{r}(\mathbf{p1},\mathbf{p2},\mathbf{p3}))$ ▸ *Done*

$\mathbf{pl3}(\mathbf{p1},\mathbf{p2},\mathbf{p3}):=\mathbf{pl}(\mathbf{m}(\mathbf{p2},\mathbf{p3}),\mathbf{icc}(\mathbf{p1},\mathbf{p2},\mathbf{p3}),\mathbf{r}(\mathbf{p1},\mathbf{p2},\mathbf{p3}))$ ▸ *Done*

their intersection points are:

$\mathbf{ip\_aux}(l1,l2):=\mathrm{linSolve}\left(\begin{cases}l1\\ l2\end{cases},\{x,y\}\right)$ ▸ *Done*

$\mathbf{ip}(l1,l2):=[\mathbf{ip\_aux}(l1,l2)[1]\ \ \mathbf{ip\_aux}(l1,l2)[2]]$ ▸ *Done*

the vertices of the triangle of polars are:

$\mathbf{pp1}(\mathbf{p1},\mathbf{p2},\mathbf{p3}):=\mathbf{ip}(\mathbf{pl1}(\mathbf{p1},\mathbf{p2},\mathbf{p3})=0,\mathbf{pl2}(\mathbf{p1},\mathbf{p2},\mathbf{p3})=0)$ ▸ *Done*

$\mathbf{pp2}(\mathbf{p1},\mathbf{p2},\mathbf{p3}):=\mathbf{ip}(\mathbf{pl1}(\mathbf{p1},\mathbf{p2},\mathbf{p3})=0,\mathbf{pl3}(\mathbf{p1},\mathbf{p2},\mathbf{p3})=0)$ ▸ *Done*

$\mathbf{pp3}(\mathbf{p1},\mathbf{p2},\mathbf{p3}):=\mathbf{ip}(\mathbf{pl3}(\mathbf{p1},\mathbf{p2},\mathbf{p3})=0,\mathbf{pl2}(\mathbf{p1},\mathbf{p2},\mathbf{p3})=0)$ ▸ *Done*

We start with two triangles with given numerical data. Even here we see that the CAS behaves differently: the first triangle is constructed such that the vertices of the polar triangle have rational coordinates – no problem. The second triangle gives irrational coordinates (lots of roots). The area of the polar triangle can only be calculated by approximation.
Finally, the generalized case doesn't work at all because we don't get a workable expression for the centre of the incircle. At least I could not do it (Josef).

pp3(p1,p2,p3):=ip(pl3(p1,p2,p3)=0,pl2(p1,p2,p3)=0) ▸ *Done*

$a:=\left[\frac{-13}{4}\ \ -8\right]$: $b:=\left[\frac{25}{2}\ \ -4\right]$: $c:=[2\ \ 10]$ ▸ $[2\ \ 10]$

area(a,b,c) ▸ $\frac{525}{4}$ area(pp1(a,b,c),pp2(a,b,c),pp3(a,b,c)) ▸ $\frac{525}{4}$

Next example:

$u:=[-3\ \ 0]$: $v:=[4\ \ -4]$: $w:=[3\ \ 6]$ ▸ $[3\ \ 6]$

area(u,v,w) ▸ 33

only by approximating, in exact mode too many roots – system hangs up!

area(pp1(u,v,w),pp2(u,v,w),pp3(u,v,w)) ▸ 33.

Generalized:

$u1:=[0\ \ 0]$: $v1:=[a\_\ \ 0]$: $w1:=[b\_\ \ c\_]$ ▸ $[b\_\ \ c\_]$

area(u1,v1,w1)|a_>0 and b_>0 and c_>0 ▸ $\frac{a\_\cdot c\_}{2}$

icc(u1,v1,w1)|a_>0 and b_>0 and c_>0

$$\blacktriangleright \left[\left\{\frac{a\_\cdot\left(\sqrt{b\_^2+c\_^2}+b\_\right)}{\sqrt{a\_^2-2\cdot a\_\cdot b\_+b\_^2+c\_^2}+a\_+\sqrt{b\_^2+c\_^2}}, \sqrt{a\_^2-2\cdot a\_\cdot b\_+b\_^2+c\_^2}\cdot b\_-(a\_-b\_)\cdot\sqrt{b\_^2+c\_^2}\neq 0\right.\right.$$

The great benefit of DERIVE is – besides the more powerful CAS – that we can calculate and plot more or less simulteously.

#1: [CaseMode := Sensitive, InputMode := Word]

#2: $\left[A := \left[-\frac{13}{4}, -8\right], B := \left[\frac{25}{2}, -4\right], C := [2, 10]\right]$

#3: [A, B, C, A]

m(U,V): slope of line passing points U, V

#4: $m(U, V) := \frac{U_2 - V_2}{U_1 - V_1}$

l(U,V): implicite form of equation of line (U,V)

#5: $l(U, V) := y - U_2 - m(U, V)\cdot(x - U_1) = 0$

IP(l1,l2): Intersection point of lines l1, l2

#6: $IP(l1, l2) := (\text{SOLUTIONS}(l1 = l2, [x, y]))_1$

A_triangle(U,V,W): Area of triangle UVW

#7: $A\_triangle(U, V, W) := \frac{1}{2}\cdot|CROSS(V - U, W - U)|$

pl(P0,C1,r): Polar line of point P0 with respect to circle (centre C1 and radius r)

#8: $pl(P0, C1, r) := (P0_1 - C1_1)\cdot(x - C1_1) + (P0_2 - C1_2)\cdot(y - C1_2) = r^2$

Mp(U,V): Midpoint of segment (U,V)

#9: $Mp(U, V) := \frac{U + V}{2}$

Abp(U,V,W): Point on angular bisector of angle < UVW

#10: $Abp(U, V, W) := V + \frac{U - V}{|U - V|} + \frac{W - V}{|W - V|}$

Ab(U,V,W): Angular bisector of angle < UVW

#11: Ab(U, V, W) := l(V, Abp(U, V, W))

Icc(U,V,W): Centre of incircle of triangle UWV

#12: Icc(U, V, W) := IP(Ab(U, V, W), Ab(W, U, V))

Icr(U,V,W): Radius of incircle of triangle UWV, applying the Hesse form

#13: $n0(U, V) := \frac{[U_2 - V_2, V_1 - U_1]}{|[U_2 - V_2, V_1 - U_1]|}$

#14: $Icr(U, V, W) := |(U - Icc(U, V, W))\cdot n0(U, V)|$

Equation of the incircle: Inc(C,r)

#15: $Inc(U, V, W) := (x - (Icc(U, V, W))_1)^2 + (y - (Icc(U, V, W))_2)^2 = Icr(U, V, W)^2$

pl1, pl2, pl3 are the polar lines

#16: pl1(U, V, W) := pl(Mp(U, V), Icc(U, V, W), Icr(U, V, W))

#17: pl2(U, V, W) := pl(Mp(U, W), Icc(U, V, W), Icr(U, V, W))

#18: pl3(U, V, W) := pl(Mp(V, W), Icc(U, V, W), Icr(U, V, W))

U1, V1, W1 are the vertices of the triangle of polar lines

#19: U1(U, V, W) := IP(pl1(U, V, W), pl2(U, V, W))

#20: V1(U, V, W) := IP(pl1(U, V, W), pl3(U, V, W))

#21: W1(U, V, W) := IP(pl2(U, V, W), pl3(U, V, W))

We compare the areas:

#22: A_triangle(A, B, C) = $\frac{525}{4}$

#23: A_triangle(U1(A, B, C), V1(A, B, C), W1(A, B, C)) = $\frac{525}{4}$

We take another triangle, plot and calculate again:

#24: [A := [-3, 0], B := [4, -4], C := [3, 6]]

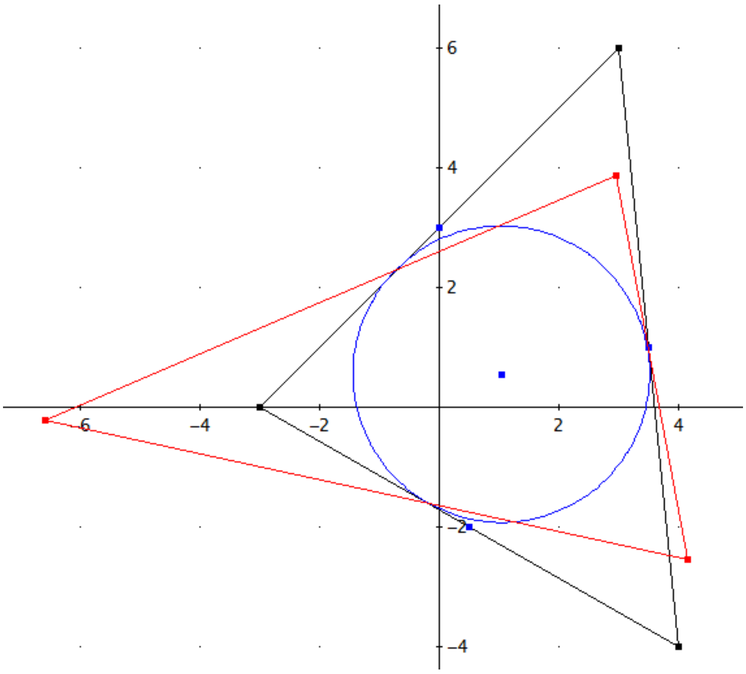
#25: [A, B, C, A]

#26: [Mp(A, B), Mp(A, C), Mp(B, C)]

#27: Inc(A, B, C)

#28: Icc(A, B, C)

#29: [U1(A, B, C), V1(A, B, C), W1(A, B, C), U1(A, B, C)]



You can see the "nice" coordinates of the vertices. No problem for DERIVE to find the area!

#30: [A_triangle(A, B, C), A_triangle(U1(A, B, C), V1(A, B, C), W1(A, B, C))]

#31: [33, 33]

#32: [U1(A, B, C), V1(A, B, C), W1(A, B, C)]

#33:
$$\begin{bmatrix} -\frac{\sqrt{6565}}{22}-\frac{2\cdot\sqrt{202}}{11}-\frac{5\cdot\sqrt{130}}{11}+\frac{107}{22} & \frac{\sqrt{6565}}{22}-\frac{7\cdot\sqrt{202}}{22}-\frac{\sqrt{130}}{22}+\frac{25}{22} \\ \frac{\sqrt{6565}}{22}-\frac{2\cdot\sqrt{202}}{11}+\frac{5\cdot\sqrt{130}}{11}-\frac{47}{22} & -\frac{\sqrt{6565}}{22}-\frac{7\cdot\sqrt{202}}{22}+\frac{\sqrt{130}}{22}+\frac{113}{22} \\ -\frac{\sqrt{6565}}{22}+\frac{2\cdot\sqrt{202}}{11}+\frac{5\cdot\sqrt{130}}{11}-\frac{25}{22} & \frac{\sqrt{6565}}{22}+\frac{7\cdot\sqrt{202}}{22}+\frac{\sqrt{130}}{22}-\frac{107}{22} \end{bmatrix}$$

But problems are to occur now when generalizing the calculation:

Generalization:

```
#34:  [A1 := [0, 0], B1 := [a, 0], C1 := [b, c]]

#35:  [b :∈ Real (0, ∞), c :∈ Real (0, ∞), a :∈ Real (0, ∞)]
```

#36: A_triangle(A1, B1, C1) = $\dfrac{a\cdot c}{2}$

```
#37:  A_triangle(pt1, pt2, pt3)
```

Out of Memory!!

Peter knows the trick: we shift the triangle in such a way that its incenter falls into the origin and then we will try calculating the area once more:

```
#38:  [A0 := A1 - Icc(A1, B1, C1), B0 := B1 - Icc(A1, B1, C1), C0 := C1 - Icc(A1, B1, C1)]

#39:  Icc(A0, B0, C0) = [0, 0]

#40:  Icc(A1, B1, C1)

#41:  A_triangle(A0, B0, C0)
```

#42: $\dfrac{a\cdot c}{2}$

Yes, it works perfectly, the areas are the same!!

You are invited to simplify `Icc(A1,B1,C1)` and compare with #39!! Due to the strange TI-Nspire output for the incenter we cannot perform this trick with TI-NspireCAS. At least, I can not.

The proof given by *Dörrie* in his book starts with creating the polar triangle in another way: the poles of the connecting lines of the midpoints of the sides form the triangle, which then turns out to be also the triangle formed by the polar lines of the midpoints.

As an exercise for the students we will do this the other way round: we will show that the vertices of the polar triangle are the poles of the sides of the midpoint triangle.

#43: $\left[A := \left[-\dfrac{13}{4}, -8\right], B := \left[\dfrac{25}{2}, -4\right], C := [2, 10]\right]$

The vertices of the "polar triangle" ...

#44: (vert := [U1(A, B, C), V1(A, B, C), W1(A, B, C)]) = vert := $\begin{bmatrix} -4 & -7 \\ 16 & -\dfrac{9}{2} \\ 2 & \dfrac{55}{8} \end{bmatrix}$

... and the lines passing the midpoints of the sides ...

```
#45:  plines := [l(Mp(A, B), Mp(A, C)), l(Mp(A, B), Mp(B, C)), l(Mp(A, C), Mp(B, C))]
```

#46: plines := $\left[4\cdot x + 3\cdot y = \dfrac{1}{2}, 24\cdot x - 7\cdot y - 153 = 0, 16\cdot x - 63\cdot y + 73 = 0\right]$

```
#47:  VECTOR(pl(pll, Icc(A, B, C), Icr(A, B, C)), pll, vert)
```

#48: $\left[4\cdot x + 3\cdot y = \frac{1}{2},\ 24\cdot x - 7\cdot y = 153,\ 16\cdot x - 63\cdot y - 127 = -200\right]$

... form pairs of pole and polar lines with respect to the incircle of the given triangle.

finding the pole with given circle (by its equation) and a line
(by its equation): pol(circle, line)

```
       pol(circle, pl, center, pole, k, m, r_2) :=
         Prog
           circle := LHS(LHS(circle) - RHS(circle))
           pl := LHS(LHS(pl) - RHS(pl))
           center := [SOLUTIONS(∂(circle, x), x), SOLUTIONS(∂(circle, y), y)]`↓1
           pl_ := (SOLUTIONS(pl, y))↓1
#49:       k := ∂(pl_, x)
           m := IF(∂(pl, y) = 0, y - center↓2, IF(∂(pl, x) = 0, x - center↓1, y - center↓2 + 1/k·(x - center↓1)))
           r_2 := circle - (x - center↓1)^2 - (y - center↓2)^2
           pole := (SOLUTIONS(pl ∧ m, [x, y]))↓1
           pl := (x - center↓1)·(pole↓1 - center↓1) + (y - center↓2)·(pole↓2 - center↓2) + r_2
           (SOLUTIONS(pl ∧ m, [x, y]))↓1
```

```
#50:  pol(Inc(A, B, C), l(Mp(A, B), Mp(A, C))) = [-4, -7]
```

#51: VECTOR(pol(Inc(A, B, C), pline), pline, plines) = $\begin{bmatrix} -4 & -7 \\ 16 & -\frac{9}{2} \\ 2 & \frac{55}{8} \end{bmatrix}$

other version of pol() working by comparing the coefficients:
pol2(circle, line)

```
       pol2(circle, pl, center, pl2, xp, yp, r_2, k1, k2) :=
         Prog
           circle := LHS(LHS(circle) - RHS(circle))
           pl := LHS(LHS(pl) - RHS(pl))
           k1 := SUBST(pl, [x, y], [0, 0])
           pl := pl/k1
#52:       center := [SOLUTIONS(∂(circle, x), x), SOLUTIONS(∂(circle, y), y)]`↓1
           r_2 := circle - (x - center↓1)^2 - (y - center↓2)^2
           pl2 := (x - center↓1)·(xp - center↓1) + (y - center↓2)·(yp - center↓2) + r_2
           k2 := SUBST(pl2, [x, y], [0, 0])
           pl2 := pl2/k2
           (SOLUTIONS(∂(pl, x) = ∂(pl2, x) ∧ ∂(pl, y) = ∂(pl2, y), [xp, yp]))↓1
```

#53: VECTOR(pol2(Inc(A, B, C), pline), pline, plines) = $\begin{bmatrix} -4 & -7 \\ 16 & -\frac{9}{2} \\ 2 & \frac{55}{8} \end{bmatrix}$

Another final comment: Inspecting the graphs I noticed that obviously the sides of the two triangles intersect in the tangency points of the incircle and the given triangle. Can you prove this?
Josef

References:

[1] *Heinrich Dörrie*, Mathematische Miniaturen, Hirt, Breslau, 1943
new edition, Sändig, Wiesbaden, ISBN 3253021157

[2] Published in The Mathematical Gazette, 98(542), July 2014, pp. 334-339. All rights reserved by *The Mathematical Association*, http://www.m-a.org.uk/jsp/index.jsp?lnk=620