THE DERIVE - NEWSLETTER #67

ISSN 1990-7079

THE BULLETIN OF THE

DERIVE®

USER GROUP + CAS-TI

Contents:

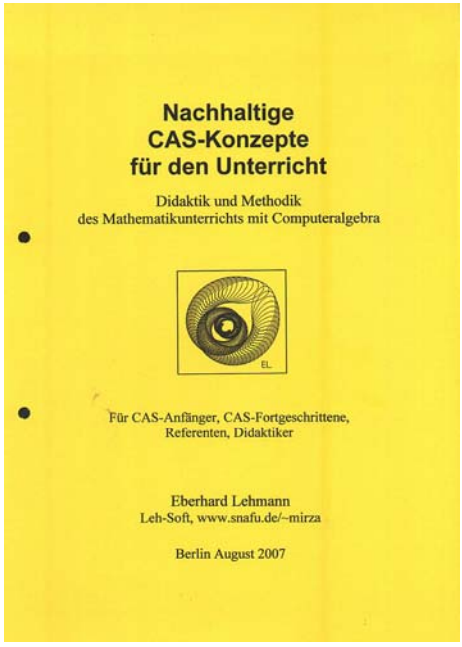
September 2007

Nachhaltige CAS-Konzepte für den Unterricht

Eberhard Lehmann

Bestellung des Buches bei:
Dr. Eberhard Lehmann
mirza@snafu.de, 030-7112420,
www.snafu.de/~mirza

20 Euro + Verpackung/Porto auf Rechnung



Another new publication is:

Günter Scheu u.a., ABITUR – Prüfungsaufgaben mit Lösungen, Mathematik CAS, Gymnasium Baden-Württemberg, STARK Verlag

If you know about CAS- related publications then please inform me. Books dealing with the use of technology in math education in general are also interesting. Josef

Dear DUG Members,

I am sorry for the delay delivering DNL#67. But I believe that its contents is worth some days overtime. We have again much more pages than the usual 40 – 44 pages. I didn´t want to leave one of the articles for the next issue. Josef Lechner´s contribution is one of the many ones which waited very long for publication, here it is at last. Nils Hahnfeld has been busy as ever and produced new packages for the handheld. As one of my final diploma papers at university dealt with conics I was very interested in his CME (*Conics Made Easy*). I could not resist making suggestions for including additional features and a very intense communication has started. You can find a selection of this discussion together with a short presentation of CME. *Differential Equations Made Easy* will be presented in a future DNL.

One highlight in this issue is doubtless Albert Rich´s visionary paper. I am very happy and proud that Albert like so many of you keep on communicating within the DUG. It would be great if DUG members would follow Albert´s invitation for supporting his vision.

Another vision is to transfer as many Derive functionalities as possible to its intended successor, the TI-Nspire. The TI-programmers promised to do their best and they are asking for the DERIVE-people´s input. The DUG can play an important role in transfering the "DERIVE spirit" to the Nspire. (Maybe it is not a pure chance that Derive ***SPIR***it and N***SPIR***e have an important part of their names in common). Bernhard Kutzler presents a more detailed view on the Nspire. Next releases should contain one or the other of the DERIVE-features. Bernhard will continue in the next DNL with a special example to demonstrate how the various applications of Nspire can cooperate.

At the occasion of a meeting with the TI-programmers some of us presented our most appreciated Derive features, which we would like to have implemented in future versions of Nspire David Sjöstrand showed an excellent example of applying the ITERATES-command. This and other interesting findings in a triangle are collected in David´s contribution in this DNL. He extended some ideas into 3D space (see next DNL).

Another gem is promised for one of the next DNLs: A. Perotti joined the DUG recently. I remembered Alessandro´s article on Groebner Bases in an earlier issue of the Intl DERIVE Journal. I wrote a mail and he immediately answered. He has adapted his DOS-Derive functions for DERIVE6 and wrote a full paper. He gave permission to publish it in the DNL and additionally promised to write an extra introduction. So we can hope that solving systems of nonlinear equations applying Buchberger´s algorithm will not remain a „Black Box" for some (or many?) of us.

You can also find a contribution from Don Phillips, bringing actuarial math on the handheld together with some additions for the former DERIVE file.

Peter Lücke-Rosendahl wrote a mail concerning some „gaps" in my DNL#15 contribution on Curve Discussions. We had an interesting exchange of mails and files and I´ll come back to this issue in the next DNL.

TIME GOES AFRICA. TIME 2008 will be held 22 – 26 September 2008 at Buffelspoort Resort, South Africa.

Michel Beaudin and I were asked to act as Co-Chairs for the DERIVE-Nspire-CAS-strand. Michel and I found it a good idea to have a completely new group of colleagues in the Program Committee. We would like to thank alle members of earlier PCs. They did a great job and they really deserve a break.

You can find the first flyer and invitation (together with another announcement) at the end of this DNL.

Finally I´d like to draw your attention to the presented books (unfortunately only in German). If anybody of you knows about interesting publications, then please let me know.

By the way, Michel Beaudin provided a paper full of exercises for his students on the EMTS (in French). We will publish it in the next DNL.

(Have a also look on SketchUp – page 20!)

Best regards to you all

The *DERIVE-NEWSLETTER* is the Bulletin of the *DERIVE* & CAS-*TI User Group*. It is published at least four times a year with a contents 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/FAX: 43-(0)2275/8207
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: December 2007
Deadline 15 November 2007

Preview: Contributions waiting to be published

Some simulations of Random Experiments, J. Böhm, AUT, Lorenz Kopp, GER
Wonderful World of Pedal Curves, J. Böhm
Tools for 3D-Problems, P. Lüke-Rosendahl, GER
ANOVA with *DERIVE* & *TI*, M. R. Phillips, USA
Financial Mathematics 4, M. R. Phillips
Hill-Encription, J. Böhm
Farey Sequences on the *TI,* M. Lesmes-Acosta, COL
Simulating a Graphing Calculator in *DERIVE*, J. Böhm
Henon & Co, J. Böhm
Do you know this? Cabri & CAS on PC and Handheld, W. Wegscheider, AUT
An Interesting Problem with a Triangle, Steiner Point, P. Lüke-Rosendahl, GER
Overcoming Branch & Bound by Simulation, J. Böhm, AUT
Diophantine Polynomials, D. E. McDougall, Canada
Graphics World, Currency Change, P. Charland, CAN
Precise Recurring Decimal Notation, P. Schofield, UK
Solving an Optimization Problem with the TI-Nspire CAS, B. Kutzler, AUT
Cubics, Quartics – interesting features, T. Koller & J. Böhm
Logos of Companies as an Inspiration for Math Teaching
Exciting Surfaces in the FAZ
BooleanPlots.mth, P. Schofield, UK
What is hiding in Dr. Pest? B. Grabinger, GER
Truth Tables on the TI, M. R. Phillips
Advanced Regression Routines for the TIs, M. R. Phillips
Directing Our Suspicions with AHP, C. & P. Leinbach, USA
Where oh Where is IT? (GPS with CAS), C. & P. Leinbach, USA
Groebner Bases, A. Perotti, IT
Embroidery Patterns, H. Ludwig, GER
Inspheres and Exspheres of a Tetrahedron, D. Sjöstrand, SWE
Compléments de mathématiques (profil Génie Mécanique), M. Beaudin, CAN

and others

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

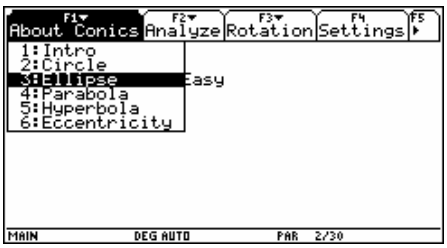
Conics Made Easy

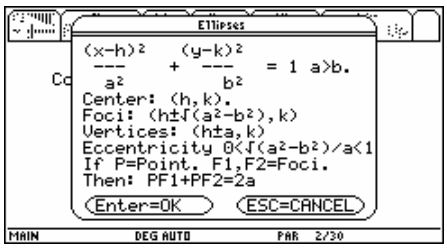
for TI-89/Titanium and TI-92+/Voyage 200

DUG-Member Nils Hanhfeld offers two new packages for the TI-handheld devices: CME (Conics Made Easy) and DME (Differential Equations Made Easy). Both products can be purchased via Internet: www.ti89.com.

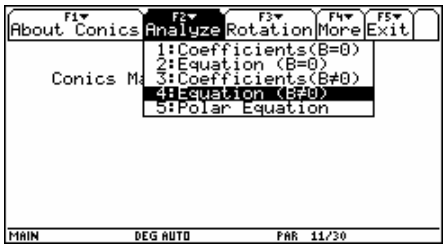
I´d like to present some screen shots from CME. I´ll take the Voyage 200 because of the larger screen which needs a lot less scrolling the screens.

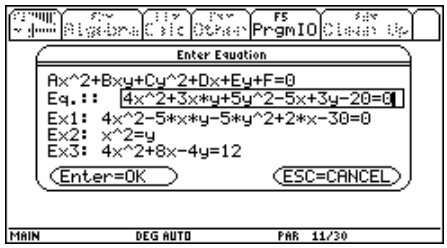
After loading the Flash Application you are offered a bar with 5 drop down menus. The first menu gives the most important information about the conics, eg. the ellipse:



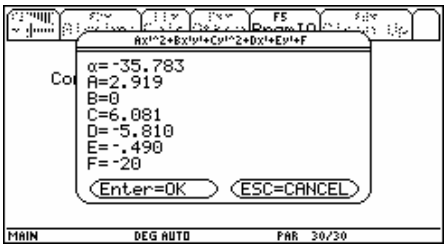


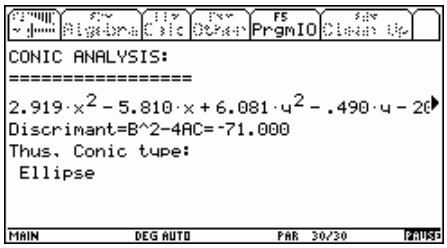
The most powerful menu is F2 Analyze. You can enter either the coefficients of the equation of a given polynomial (in rectangular coordinates only) or the full equation (see the example).

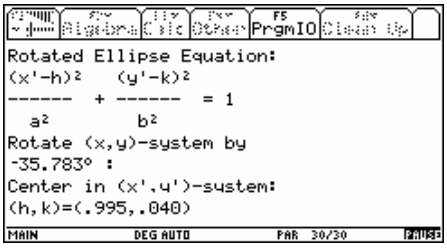


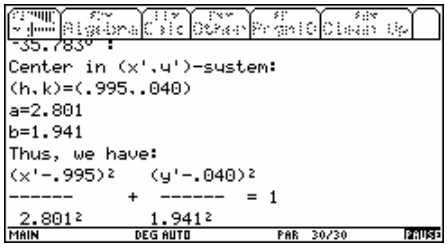


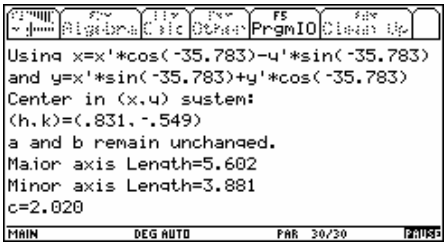
CME performs a complete analysis of the given conic followed by its plot.

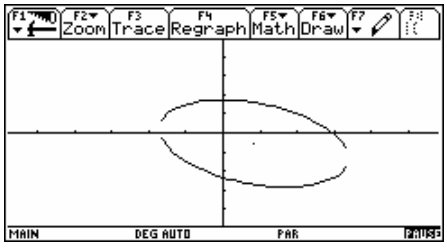






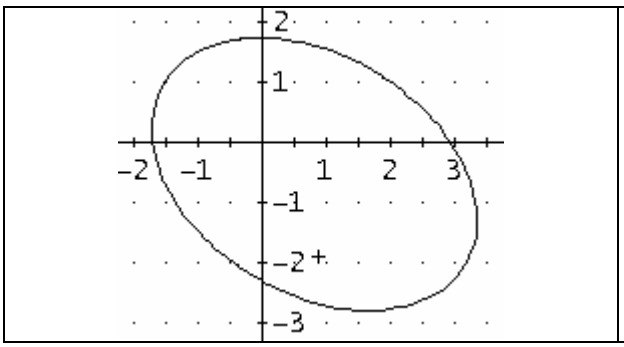


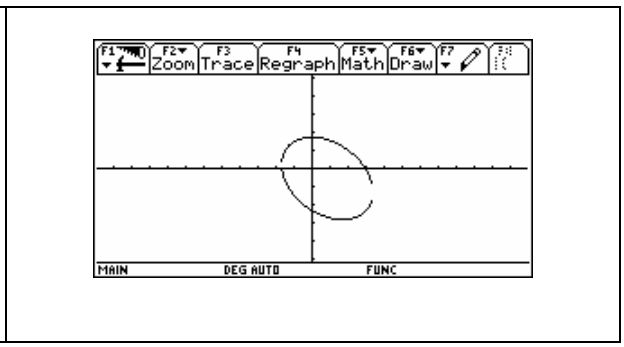




CME switches automatically to DEG-Mode.

Comparing the plot with the DERIVE plot (plotting the implicitly given expression) you will recognize that the TI-plot has sometimes a scale adapted to the dimension of the conic and that the TI-plot shows a nasty gap between the two parts, caused by problems plotting curves which are close to vertical runs. The ellipse is splitted into two parts.





This problem could be overcome by plotting the full conic in parameter form.

Here is a step by step approach for plotting the given conic in parameter form. (The last time when I performed a calculation like this was when I was 18 writing my end examination in gymnasium – several years ago ...)

General form $f(x,y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$

#1: $f(x, y) := 4 \cdot x^2 + 3 \cdot x \cdot y + 5 \cdot y^2 - 5 \cdot x + 3 \cdot y - 20$

First of all check the type!

```
ch_conic(u, dis) :=
  Prog
    dis := ∂(∂(u, x), y)^2 - ∂(u, x, 2)·∂(u, y, 2)
    If dis > 0
#2:     "hyperbola"
    If dis < 0
      "ellipse"
      "parabola"
```

#3: ch_conic(f(x, y)) = ellipse

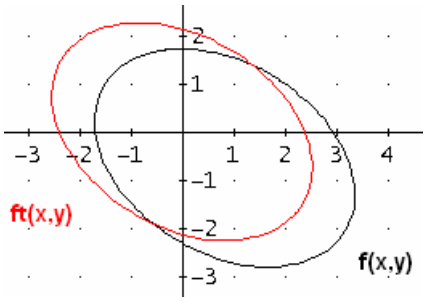
Finding the center

#4: $\left[cr := \left[\text{SOLUTIONS}\left[\frac{d}{dx} f(x, y) = 0 \wedge \frac{d}{dy} f(x, y) = 0, [x, y]\right]\right]_1\right] = cr := \left[\frac{59}{71}, -\frac{39}{71}\right]$

#5: $ft(x, y) := f(x + cr_1, y + cr_2)$

This is the conic with its center in the shifted origin:
(the equation does not contain any linear terms)

#6: $ft(x, y) := \frac{284 \cdot x^2 + 213 \cdot x \cdot y + 355 \cdot y^2 - 1626}{71}$



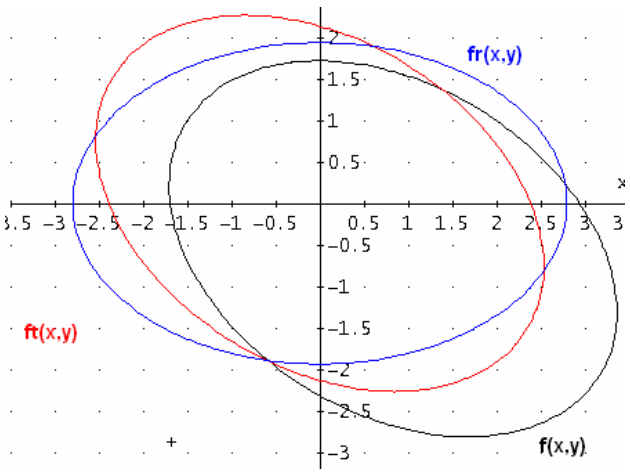
We rotate the coordinate-axes that the axes of the conic will lie on the xy-axes, finding first the rotation angle α from tan(2α) = B/(A-C):

#7: $$\left[\alpha := \frac{\mathrm{ATAN}\left(\frac{2\cdot\frac{d}{dy}\frac{d}{dx}f(x,y)}{\left(\frac{d}{dx}\right)^2 f(x,y) - \left(\frac{d}{dy}\right)^2 f(x,y)}\right)}{2}\right] = \alpha := -\frac{\mathrm{ACOT}\left(\frac{1}{3}\right)}{2}$$

We apply the rotation formulae

#8: $fr(x, y) := ft(x\cdot COS(\alpha) - y\cdot SIN(\alpha),\ x\cdot SIN(\alpha) + y\cdot COS(\alpha))$

#9: $$fr(x, y) := -\frac{71\cdot x^2\cdot(\sqrt{10} - 9) - 71\cdot y^2\cdot(\sqrt{10} + 9) + 3252}{142}$$



Finding the vertices by calculating the intersection points with the axes:

#10: $vertr(u) := APPEND(SOLUTIONS(u = 0 \wedge y = 0, [x, y]), SOLUTIONS(u = 0 \wedge x = 0, [x, y]))$

#11: $$vertr(fr(x, y)) = \begin{bmatrix} \frac{2\cdot\sqrt{(813\cdot\sqrt{10} + 7317)}}{71} & 0 \\ -\frac{2\cdot\sqrt{(813\cdot\sqrt{10} + 7317)}}{71} & 0 \\ 0 & \frac{2\cdot\sqrt{(7317 - 813\cdot\sqrt{10})}}{71} \\ 0 & -\frac{2\cdot\sqrt{(7317 - 813\cdot\sqrt{10})}}{71} \end{bmatrix}$$

#12: $$vertr(fr(x, y)) = \begin{bmatrix} 2.801072673 & 0 \\ -2.801072673 & 0 \\ 0 & 1.940611662 \\ 0 & -1.940611662 \end{bmatrix}$$

We can obtain the normal form using Eigenvalues, too. But this was not the way I did it in 1963 in gymnasium. (I´ll do this in the next DNL.)

The parameter form with coordinate axes shifted and rotated
(& length of half axes):

#13: $[a := 2.801072673,\ b := 1.940611662]$

#14: $[a \cdot \mathrm{COS}(t),\ b \cdot \mathrm{SIN}(t)]$

To obtain the parameter form of the original conic we have to
rotate and shift back:

#15: $fp(t) := [a \cdot \mathrm{COS}(t),\ b \cdot \mathrm{SIN}(t)] \cdot \begin{bmatrix} \mathrm{COS}(-\alpha) & -\mathrm{SIN}(-\alpha) \\ \mathrm{SIN}(-\alpha) & \mathrm{COS}(-\alpha) \end{bmatrix} + cr$

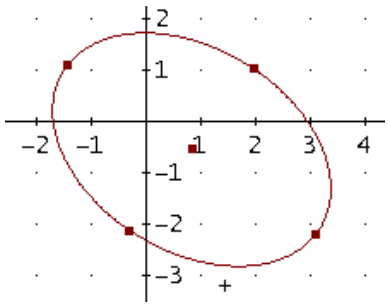
#16: $$fp(t) := \left[\frac{2801072673 \cdot \sqrt{(5 \cdot \sqrt{10} + 50)} \cdot \mathrm{COS}(t)}{10000000000} + \frac{970305831 \cdot \sqrt{(50 - 5 \cdot \sqrt{10})} \cdot \mathrm{SIN}(t)}{5000000000} + \frac{59}{71},\ - \frac{2801072673 \cdot \sqrt{(50 - 5 \cdot \sqrt{10})} \cdot \mathrm{COS}(t)}{10000000000} + \frac{970305831 \cdot \sqrt{(5 \cdot \sqrt{10} + 50)} \cdot \mathrm{SIN}(t)}{5000000000} - \frac{39}{71} \right]$$

Having plotted the conic in parameter form, we want to add the center and the vertices (and maybe for hyperbolas the asymptotes, too). You can also add the foci.

#17: $$conpts := \mathrm{VECTOR}\left(u + cr,\ u,\ \mathrm{APPEND}(vertr(fr(x, y)), [[0, 0]]) \cdot \begin{bmatrix} \mathrm{COS}(-\alpha) & -\mathrm{SIN}(-\alpha) \\ \mathrm{SIN}(-\alpha) & \mathrm{COS}(-\alpha) \end{bmatrix} \right)$$

#18: $$conpts = \begin{bmatrix} \frac{\sqrt{(77235 \cdot \sqrt{10} + 406500)}}{355} + \frac{59}{71} & -\frac{\sqrt{(4065 \cdot \sqrt{10} + 325200)}}{355} - \frac{39}{71} \\ \frac{59}{71} - \frac{\sqrt{(77235 \cdot \sqrt{10} + 406500)}}{355} & \frac{\sqrt{(4065 \cdot \sqrt{10} + 325200)}}{355} - \frac{39}{71} \\ \frac{\sqrt{(406500 - 77235 \cdot \sqrt{10})}}{355} + \frac{59}{71} & \frac{\sqrt{(325200 - 4065 \cdot \sqrt{10})}}{355} - \frac{39}{71} \\ \frac{59}{71} - \frac{\sqrt{(406500 - 77235 \cdot \sqrt{10})}}{355} & -\frac{\sqrt{(325200 - 4065 \cdot \sqrt{10})}}{355} - \frac{39}{71} \\ \frac{59}{71} & -\frac{39}{71} \end{bmatrix}$$

#19: $$conpts = \begin{bmatrix} 3.103334232 & -2.187111775 \\ -1.441362401 & 1.088520225 \\ 1.965681513 & 1.025010271 \\ -0.3037096823 & -2.12360182 \\ 0.8309859154 & -0.5492957746 \end{bmatrix}$$



When I worked the first time with CME I missed two options:

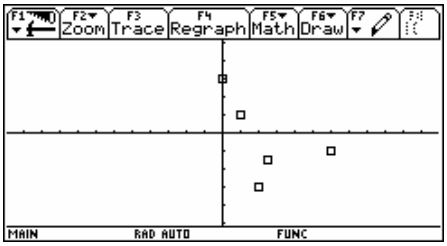
- A general conic is given by five points: Enter the points eg in form of a matrix and receive the equation of the conic containing the points – followed by its analysis + plot.

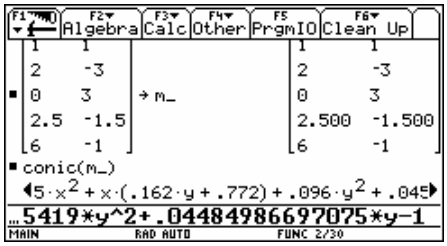
- All conics can be presented with one common function – in polar coordinates. This would be a welcome opinion to make students more accustomed with polar forms.

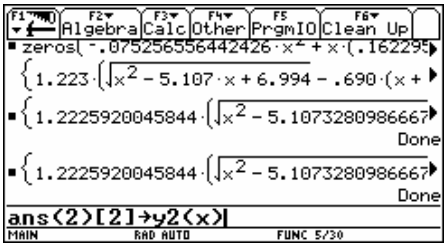
I suggested Nils to implement these features and he promised to do. I´d like to present how to find the equation of a conic which is given by five points.

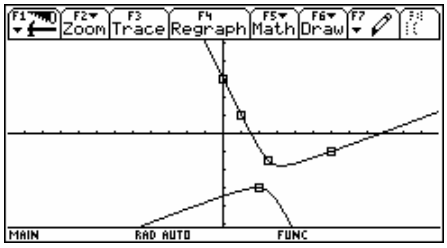
Starting with the general equation the problem is to solve a system of five linear equations with five unknowns – this should be no problem for a CAS:

The screen shot shows 5 points (matrix m_), then I call function conic(m_) and receive immediately the equation of the conic – in rectangular form. Finally I plot its graph.









Function conic() is only a transfer of a DERIVE-function. See the function and an “over-painted” picture of a fountain using the derived equations:

$$\#1:\quad con(x, y) := a\cdot x^2 + b\cdot x\cdot y + c\cdot y^2 + d\cdot x + e\cdot y = 1$$

$$\#2:\quad conic(pts) := (SOLUTIONS(VECTOR(con(v_1, v_2), v, pts), [a, b, c, d, e]))\cdot[x^2, x\cdot y, y^2, x, y] = 1$$

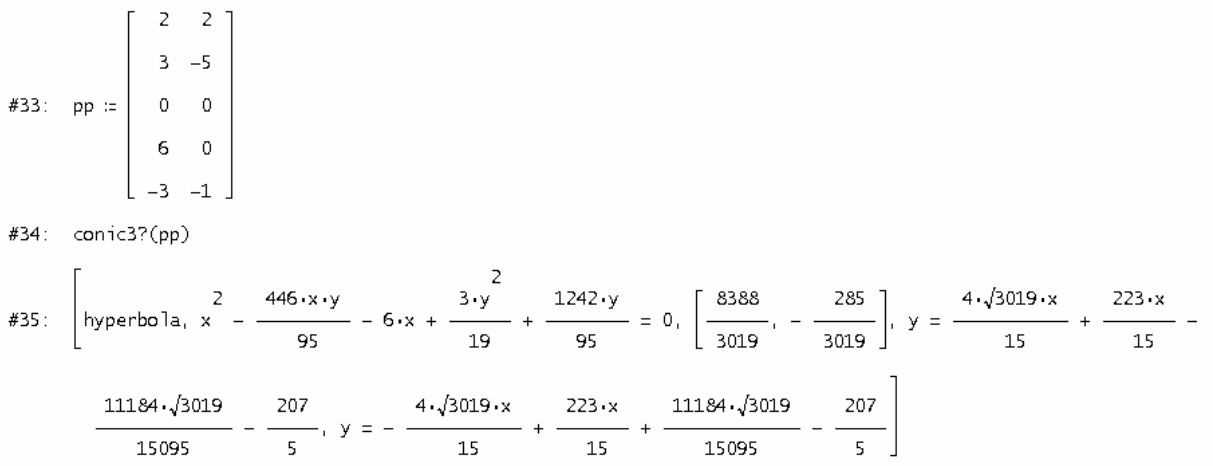
$$\#3:\quad conic(ell1)$$

$$\#4:\quad 0.0088417\cdot x^2 + x\cdot(0.0020697\cdot y + 0.00097258) + 0.11836\cdot y^2 + 0.15764\cdot y = 1$$

I was very happy with this function for a long while, but a “bug” is hidden. Do you see what is missing? No, then try to find the equation of a conic which is passing the origin!

In the meanwhile we had a fruitful cooperation and many of the ideas have been realized for an updated version of CME.

In the next DNL I´ll include a DERIVE-function (program) performing a complete analysis.



Information provided by Bernhard Kutzler:

Hi all,

Did you know that Derive has an entry in Wikipedia - with topical information about the discontinuation and replacement with TI-Nspire?

Look at: http://en.wikipedia.org/wiki/Derive_computer_algebra_system

and in German: http://de.wikipedia.org/wiki/Derive

Best regards,
bernhard

Restricted use of Slider Bars in DERIVE

Heinrich Ludwig, Germany

Preparing a visualisation in the frame of an introduction into Graph Theory for my students I wanted to turn on and off edges connecting nodes using slider bars. This works very easy presenting the edge as a segment connecting two nodes by a 2 by 2 –matrix and multiplying the matrix by a factor a with possible values 0 or 1 for a. With a = 0 the segment shrinks to length 0 and with a = 1 one can observe the full edge.

Everything seems to work properly: one can introduce many slider bars, but … only the first nine of them will be really working. I found no way to overcome this deficiency.

DNL: *I wrote to Albert Rich and Theresa Shelby. The answer confirmed this restriction. It would need changing the DERIVE code plus recompiling. Unfortunately we cannot expect a new release, so we have to live with only 9 working slider bars.*

Ignacio Larrosa Cañestro

I know that TI is not continuing the development to Derive, but I expect that this list will continue alive. Someone may have some explanation for the slowness shown by Derive executing this function:

With argument k = 2^m + 1, for values of m between 1 and 15 delays it needs only fractions of a second. But from m = 15 it needs much more time: until 182 seconds for m = 19, and even to exhaust my patience for m = 20, on a Pentium IV 2800MHz.

```
      f(k) :=
        Prog
          k := k/2
          m := 1
          Loop
#1:         If INTEGER?(k)
              RETURN m
            k := k·CEILING(k)
            DISPLAY(DIM(FLOOR(k)))
            m :+ 1
```

Really alone it should do m iterations, the answer has to be m + 1, as really gives for m less than or equal to 19.

Thanks in advance,
Ignacio Larrosa Cañestro
A Coruña (España)
ilarrosa@mundo-r.com . , .

Albert Rich

Hello Ignacio,

I don't think it strange at all that your function is so slow in computing f(2^20+1). Including the statement

DISPLAY(DIM(FLOOR(k))),

in the loop of your function displays the number of digits of k for each iteration of the loop. As you will see the number of digits doubles with each iteration. By iteration 20, multi-million digits numbers are being multiplied together!

In fact, your function is actually a good example of the high performance of Derive's infinite precision arithmetic.

Aloha from Hawaii,
Albert D. Rich
Co-author of Derive

Peter Lüke-Rosendahl, Germany

Hello Josef,

… I like to read your revised reprints. I am just trying your „Discussion of a Curve“ from revised DNL#15. I found out that “terrace points” (= points with f’ = f’’ = 0) are not recognized properly. The reason might by that within an IF-condition sign(1) and sign(0) are not distinguished …

In his next mail Peter sent his proposal using the fact that the sign of the first derivative changes in a turning point, but does not change in an extremal value with slope = 0.
I implemented Peter´s idea and by the way I tried to improve the “program” from revised DNL15 to also consider points with vanishing higher derivatives (giving “flat points” …)

Many thanks, Peter for the fruitful discussion.

All bodies are falling equally fast – Are all bodies falling equally fast?

Josef Lechner, Viehdorf, Austria

Introduction

More and more we are faced with the demand on real life applications in teaching mathematics. But teaching real life applications has the consequence to deal with complex models. As one example we will investigate the free fall under considering air resistance [1], which is often discussed in school but only rarely really treated.

Equation of motion

In contrary to free fall in vacuum a body falling free in the air is not only influenced by the gravitational force but also by a retarding resistance force, which is described by NEWTON´s formula for the resistance

$$F_W = c_w A \frac{\rho v^2}{2}.$$

Equation of motion follows

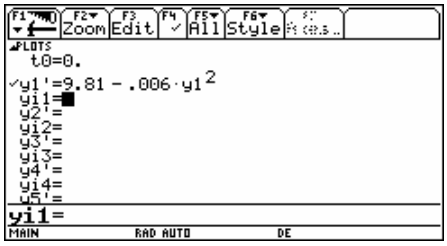
$$m \cdot a = m \cdot g - c_w A \frac{\rho \cdot v^2}{2}$$

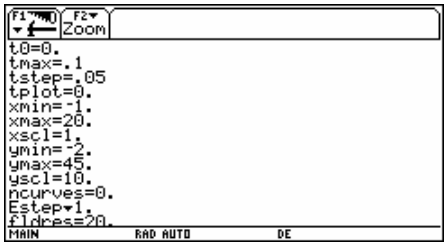
$$a = g - \frac{c_w A \cdot \rho}{2m} v^2$$

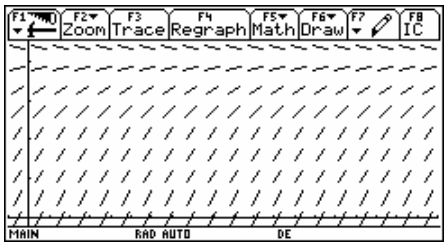
$$\frac{dv}{dt} = g - \frac{c_w A \cdot \rho}{2m} v^2 \qquad (1)$$

The numerical approach

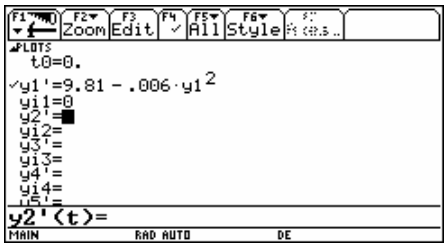
Supported by a CAS (DERIVE, Voyage 200, TI-92+ or TI-Nspire) it is easy to obtain very quick a numerical solution. (In this paper we use the TI-92+ / Voyage 200, but all procedures can be reproduced using DERIVE or the Nspire.) When we enter the equation of motion in the Y=Editor (having set the DIFF EQUATIONS Mode) we receive the respective direction field in order to have a first overview. (Drag coefficient c_w = 0.8, cross-sectional area of the falling body A = 0.8 m, air density ρ = 1.3 kg · m^{-3} and mass m = 70 kg: the resulting coefficient of the quadratic expression is 0.006.)

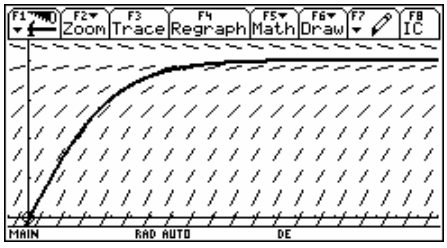






Our initial condition is $v(t=0) = 0$. So we obtain the increase of velocity which – as expected – decreases and approaches a constant value.





The end velocity which can be seen in the solution curve will be reached when restistance of air equals gravity. Equating F_g and F_W yields the end velocity

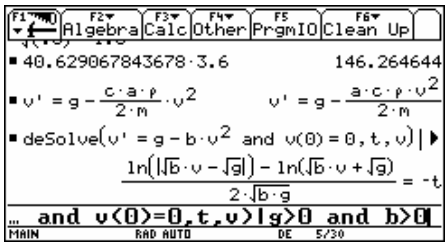
$$v_{end} = \sqrt{\frac{2m\,g}{c_w\,A\,\rho}}. \qquad (2)$$

This is in our case

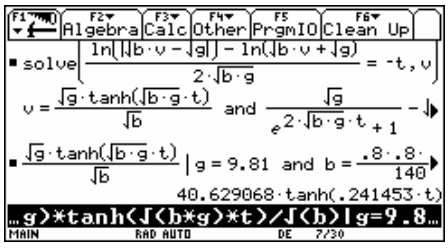
$$v_{end} \approx 40.63\,m\,s^{-1} \approx 146\,km\,h^{-1}.$$

Rate of fall as a function of time

For obtaining a presentation of the velocity function it is necessary to solve the respective differential equation. To have a better survey we set $b := \frac{c_w A \rho}{2m}$. For solving this DE the TI-devices provide the function desolve(), which makes possible to find exact solutions of most 1st and 2nd order ODEs.



Using desolve() and defining $g>0$ and $b>0$ we receive the desired expression which can be solved for the velocity v.



We can summarize the solution:

$$v(t) = \sqrt{\frac{g}{b}} \cdot \tanh\left(t\sqrt{b \cdot g}\right).$$

Back substitution under consideration of

$$\sqrt{\frac{g}{b}} = \sqrt{\frac{g \cdot 2m}{c_w A \rho}} = v_{end}$$

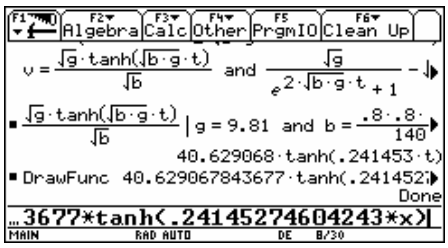
and

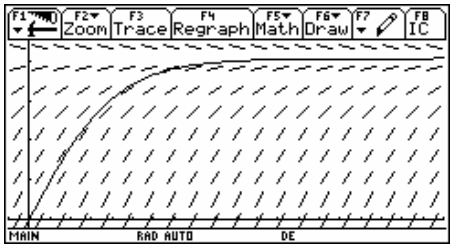
$$\sqrt{b\,g} = \sqrt{\frac{g \cdot c_w A \rho}{2m}} = \frac{g}{v_{end}}$$

results finally in the velocity function

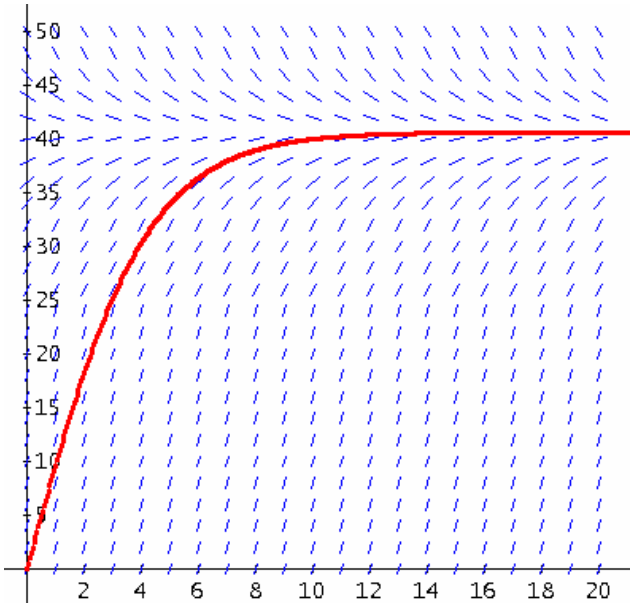
$$v(t) = v_{end} \cdot \tanh\left(\frac{g}{v_{end}} t\right). \qquad (3)$$

Working with our data we can plot the exact solution (use the Draw = F6-tool) and observe an excellent match of both curves.





Now let´s have a look how DERIVE handles the problem:



#1: DIRECTION_FIELD$\left(9.81 - \frac{1.3\cdot 0.8\cdot 0.8}{140}\cdot y^2, x, 0, 20, 20, y, 0, 50, 25\right)$

#2: SOLVE(DSOLVE1(g - b·y^2, -1, x, y, 0, 0), y)

#3: $y = \frac{\sqrt{g}\cdot(e^{2\cdot\sqrt{b}\cdot\sqrt{g}\cdot x} - 1)}{\sqrt{b}\cdot(e^{2\cdot\sqrt{b}\cdot\sqrt{g}\cdot x} + 1)}$

#4: $\text{SOLVE}\left(\text{DSOLVE1}\left(9.81 - \frac{1.3\cdot 0.8\cdot 0.8}{140}\cdot y^2, -1, x, y, 0, 0\right), y\right)$

#5: $y = \frac{40.62906784\cdot(e^{0.482905492\cdot x} - 1)}{e^{0.482905492\cdot x} + 1}$

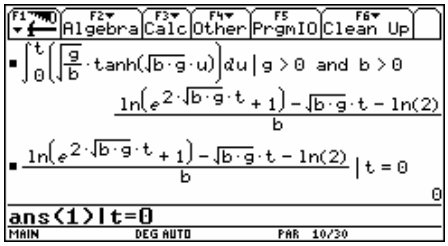
#6: $\text{TABLE}\left(\frac{40.62906784\cdot(e^{0.482905492\cdot x} - 1)}{e^{0.482905492\cdot x} + 1}, x, 0, 22, 0.05\right)$

It is interesting that DERIVE does not recognize the tanh in expression #3! We can read in the DERIVE Online Help: *The hyperbolic functions simplify to equivalent expressions involving exponentials.* That is it!

Way of the fall and its acceleration as functions of time

Now having derived the velocity function it is easy to find the distance of the fall (by integration) and its acceleration (by differentiation).

Integration of the function delivers the time--distance function with $s(t=0)=0$.

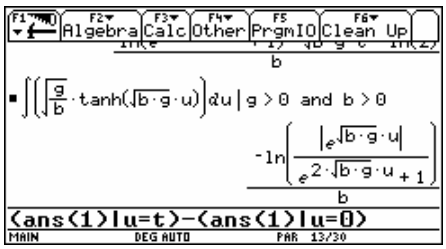


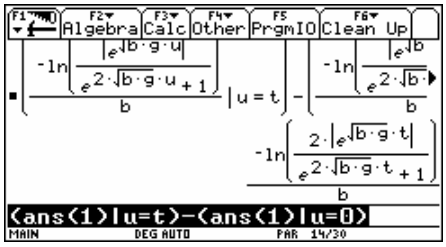
Collecting the numerator in form of a log-function enables a better and more "beautiful" presentation of the desired function:

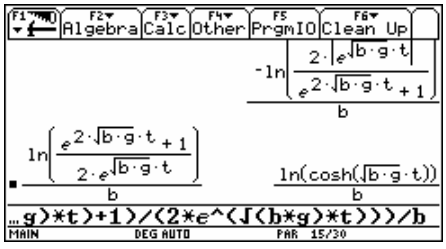
$$s(t) = \frac{1}{b}\cdot\ln\left(\frac{e^{2\sqrt{bg}\cdot t}+1}{2e^{\sqrt{bg}\cdot t}}\right).$$

The calculator does not collect the logarithms by its own, but we can force it to do so:

Let it find the indefinite integral, then substitute t and 0 for u and form the difference. The result is not the same as given in the formula and the calculator does not transform the expression into the cosh. We have to support the TI.



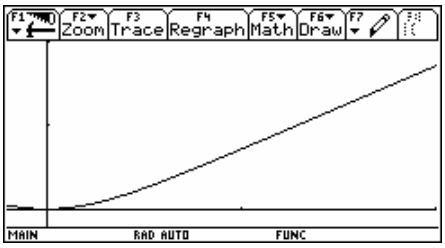




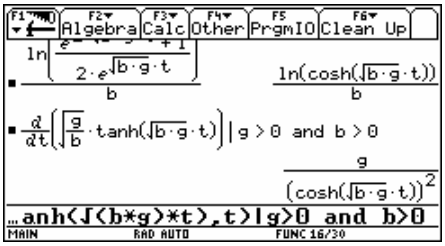
Back substitution using again v_{end} gives

$$s(t) = \frac{v_{end}^2}{g}\cdot\ln\left(\cosh\left(\frac{g}{v_{end}}\cdot t\right)\right). \qquad (4)$$

Graphic representation of our special example leads us to the typical run of a displacement function under consideration of air friction: the function is bent to left in the beginning and changes into a linear function (when the velocity becomes constant).



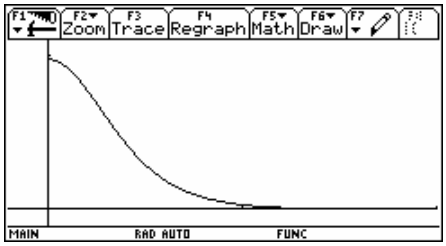
Finally we get the acceleration function as the derivative of the velocity function.



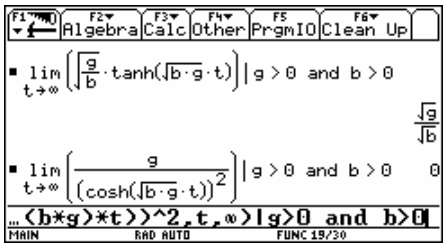
We perform back substitution using v_{end} and obtain the formula for acceleration as a function of time

$$a(t) = \frac{g}{\cosh^2\left(\frac{g}{v_{end}} \cdot t\right)}. \qquad (5)$$

The displayed graph shows the expected run. Acceleration decreases very fast from the initial value g to zero.



Of course, end acceleration a_{end} and end velocity v_{end} can be obtained analytically, too.



We turn back to DERIVE:
(Let´s assume that we don´t know about the hyperbolic functions!)

#22: $v(t) := \dfrac{\sqrt{g}\cdot\left(e^{2\cdot\sqrt{b}\cdot\sqrt{g}\cdot t} - 1\right)}{\sqrt{b}\cdot\left(e^{2\cdot\sqrt{b}\cdot\sqrt{g}\cdot t} + 1\right)}$

We substitute for b = g/v_ (with v_ = v_end):

#23: $v(t) := \dfrac{\sqrt{g}\cdot\left(e^{2\cdot\sqrt{(g/v\_)^2}\cdot\sqrt{g}\cdot t} - 1\right)}{\sqrt{\frac{g}{v\_^2}}\cdot\left(e^{2\cdot\sqrt{(g/v\_)^2}\cdot\sqrt{g}\cdot t} + 1\right)}$

#24: $v\_ : \in \text{Real}\ (0, \infty)$

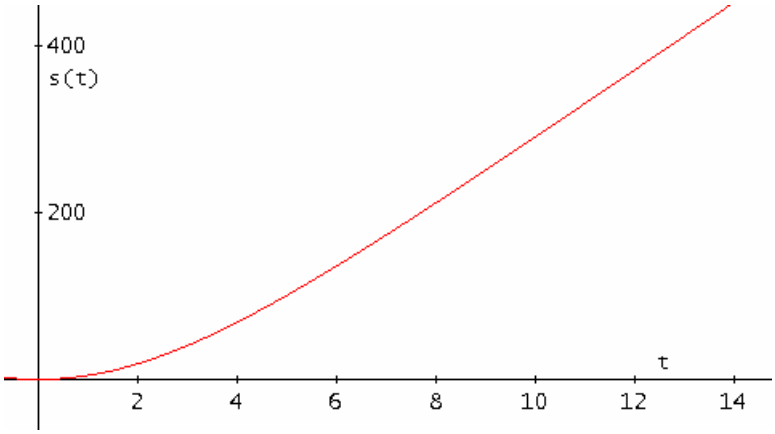
#25: $v(t) := \dfrac{v\_\cdot\left(e^{2\cdot g\cdot t/v\_} - 1\right)}{e^{2\cdot g\cdot t/v\_} + 1}$

#26: $s(t) := \int_0^t v(u)\ du$

#27: $s(t) := \dfrac{v\_^2\cdot\text{LN}\left(\frac{e^{2\cdot g\cdot t/v\_}}{2} + \frac{1}{2}\right)}{g} - t\cdot v\_$

#28: $s(0) = 0$

#29: $\dfrac{40.63^2\cdot\text{LN}\left(\frac{e^{2\cdot 9.81\cdot t/40.63}}{2} + \frac{1}{2}\right)}{9.81} - t\cdot 40.63$

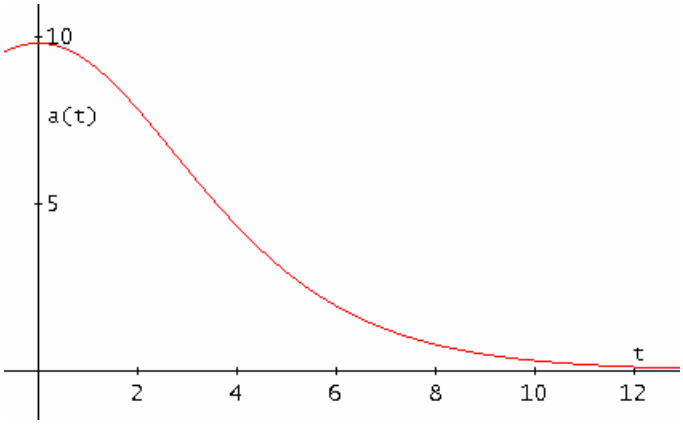


$$\#30:\quad a(t) := \frac{d}{dt}\, v(t)$$

$$\#31:\quad a(t) := \frac{4 \cdot g \cdot e^{2 \cdot g \cdot t / v\_}}{\left(e^{2 \cdot g \cdot t / v\_} + 1\right)^2}$$

$$\#32:\quad \frac{4 \cdot g \cdot e^{2 \cdot g \cdot t / v\_}}{\left(e^{2 \cdot g \cdot t / v\_} + 1\right)^2}$$

$$\#33:\quad \frac{4 \cdot 9.81 \cdot e^{2 \cdot 9.81 \cdot t / 40.63}}{\left(e^{2 \cdot 9.81 \cdot t / 40.63} + 1\right)^2}$$



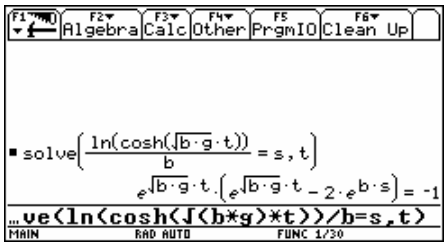
$$\#34:\quad [g :\varepsilon \text{ Real } (0, \infty),\ b :\varepsilon \text{ Real } (0, \infty)]$$

$$\#35:\quad \lim_{t \to \infty} \frac{\sqrt{g} \cdot \left(e^{2 \cdot \sqrt{b} \cdot \sqrt{g} \cdot t} - 1\right)}{\sqrt{b} \cdot \left(e^{2 \cdot \sqrt{b} \cdot \sqrt{g} \cdot t} + 1\right)} = \frac{\sqrt{g}}{\sqrt{b}}$$

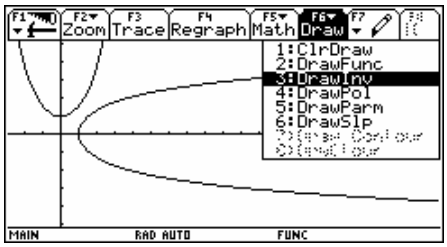
$$\#36:\quad \lim_{t \to \infty} a(t) = 0$$

Time of the fall as function of height of fall

Determination of the inverse of the time-distance-function does not lead to the desired result.



The reason for this incapability of solving the equation is the ambiguity of the arcosh(x):



We have to support the CAS by deciding for the upper branch of the arcosh-function – negative times don´t make any sense in connection with our model – then for $x \geq 1$:

$$\operatorname{arcosh}(x) = \ln\left(x + \sqrt{x^2 - 1}\right).$$

So we get from

$$\cosh(\sqrt{b\,g}\cdot t) = e^{bs}$$

$$\sqrt{b\,g}\cdot t = \ln\left(e^{bs} + \sqrt{e^{2bs} - 1}\right)$$

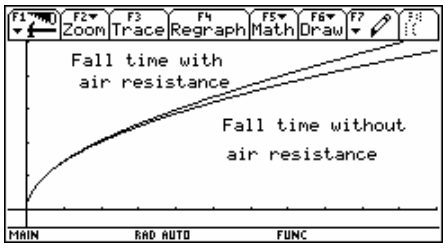
finally

$$t(s) = \frac{1}{\sqrt{b\,g}} \ln\left(e^{bs} + \sqrt{e^{2bs} - 1}\right).$$

And after introducing v_{end} we obtain the formula for the fall(=drop) time

$$t(s) = \frac{v_{end}}{g} \cdot \ln\left(e^{\frac{g\,s}{v_{end}^2}} + \sqrt{e^{\frac{2g\,s}{v_{end}^2}} - 1}\right).$$

We will compare the graphs of the functions for fall time with and without air resistance for 0 to 100 m.



Reference

[1] G. Hepperger, *Der freie Fall unter Einwirkung des Luftwiderstands,* Wissenschaftliche Nachrichten, September 1991, BMUK, Wien.

Josef received a letter from a German teacher, Madeleine:

Completion: "*Madeleine´s Problem*"

Two parachutists, *Klaus* and *Egon*, leave the plane in an altitude of 2000m and they open their parachute in an altitude of 200m above the ground. Their drag coefficient c_w is 0.45, the density of air (1.29 kg/m^3) decreases at a rate of 5% per 100 m difference of altitude. *Klaus* keeps his body vertical and his acting cross sectional area of his body is 950 cm^2. *Egon* is lying horizontally on the air bed, spreads his arms and legs reaching a cross sectional area of 7500 cm^2. They both have a weight of 80 kg. Calculate the time which they need to reach the the moment of opening their parachutes and give reasons for your procedure.

There are two forces acting on both parachutists:

$F_G = m \cdot g$ (Weight) and $F_W = c_w \cdot A \cdot \frac{\rho \cdot v^2}{2}$.

We have to consider that air density increases approaching the ground (g and c_w depend also on the altitude, but this shall be neglected here).

Leaving the plane in an altitude of H = 2000 m we can describe the density ρ as a function of the fall distance s:

$$\rho(s) = 1.29 \cdot 0.95^{\frac{2000-s}{100}}.$$

The resulting equation of motion is

$$m \cdot a = m \cdot g - c_w \cdot A \cdot \frac{\rho_0 \cdot 0.95^{\frac{H-s}{100}} \cdot v^2}{2}.$$

(H is the altitude of the jump). This leads to the following differential equation:

$$s''(t) + \frac{c_w \cdot A}{2m} \cdot \rho_0 \cdot 0.95^{\frac{H-s(t)}{100}} \cdot s'(t)^2 - g = 0.$$

For our – Madeleine´s – problem two differential equations have to be solved – one for *Klaus* and another one for *Egon*:

Klaus:

$$s''(t) + \frac{0.45 \cdot 0.095}{2 \cdot 80} \cdot 1.29 \cdot 0.95^{\frac{2000-s(t)}{100}} - s'(t)^2 - 9.81 = 0$$

Egon:

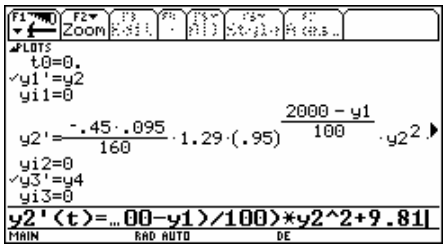
$$s''(t) + \frac{0.45 \cdot 0.75}{2 \cdot 80} \cdot 1.29 \cdot 0.95^{\frac{2000-s(t)}{100}} - s'(t)^2 - 9.81 = 0$$

The initial conditions are s(0)=0 and v(0)=0.

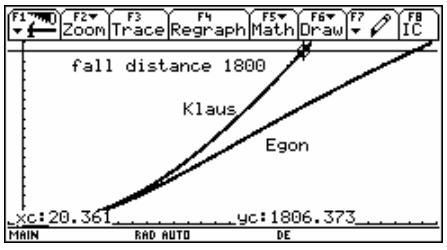
The DEs can be solved numerically (eg by applying the Runge-Kutta-method. For editing these DE´s on the TIs one has to take into account some conventions:

y1 corresponds with s_{Klaus} (1st DE)
y2 corresponds with v_{Klaus} (1st DE)
y3 corresponds with s_{Egon} (2nd DE)
y4 corresponds with s_{Egon} (2nd DE)

After entering *s*' and *v*'

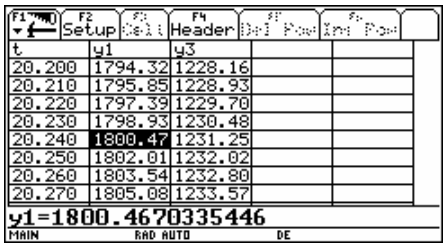


we can read off the requested times on the graphic screen – at least approximatively.



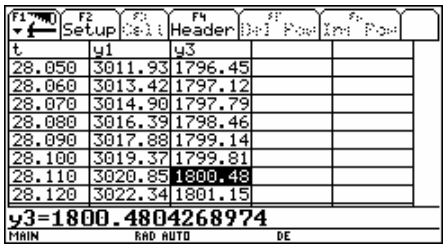
Acitvate y1' and y3' in the Y=-Editor.

We can get a more accurate estimation using the table:



t	y1	y3
20.200	1794.32	1228.16
20.210	1795.85	1228.93
20.220	1797.39	1229.70
20.230	1798.93	1230.48
20.240	1800.47	1231.25
20.250	1802.01	1232.02
20.260	1803.54	1232.80
20.270	1805.08	1233.57

y1=1800.4670335446
MAIN RAD AUTO DE



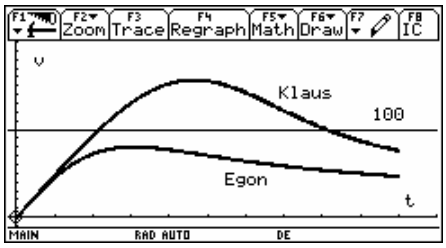
t	y1	y3
28.050	3011.93	1796.45
28.060	3013.42	1797.12
28.070	3014.90	1797.79
28.080	3016.39	1798.46
28.090	3017.88	1799.14
28.100	3019.37	1799.81
28.110	3020.85	1800.48
28.120	3022.34	1801.15

y3=1800.4804268974
MAIN RAD AUTO DE

We can read off a time of 20.2 seconds for *Klaus* and 28.1 seconds for *Egon*. y1 and y3 are the values of their falling distance. It is obvious that *Egon* needs more time

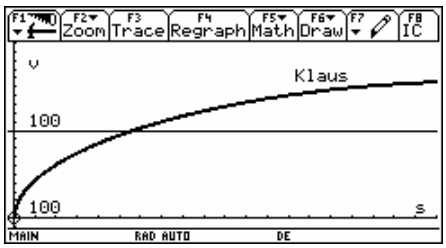
So the problem is solved – at least numerically.

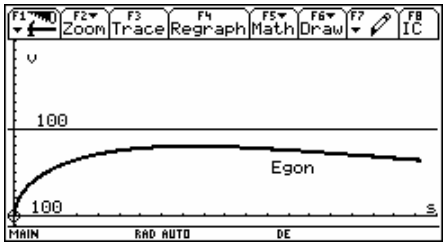
It is interesting to investigate the time – velocity diagrams of both jumpers:



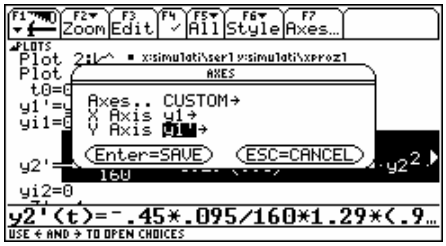
We can recognize the braking influence of the increase of the air density during fall time.

Finally we will have a look at the distance-velocity diagrams. At first it seems that we cannot have both diagrams in one graph window:

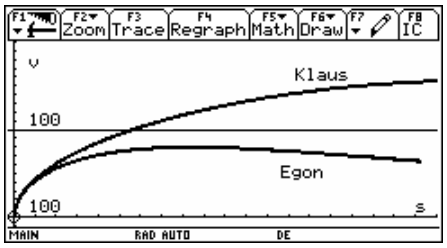




All these graphs are easy to create by using the Axes-Menu (F7) in the Y=-Editor. In Axes CUSTOM you can assign variables to the axes as you like. (The screen shot demonstrates how to set F7 Axes > CUSTOM in order to obtain the distance-velocity diagram for *Klaus*.



And with a little trick we can display both graphs on the same axes: Save both curves as separate pictures (F1-Menu > Save Copy as) and load both into an empty graph window.



The differential equations and DERIVE:

We follow the advice given in the Online Help how to treat second order differential equations in order to obtain a numerical solution.

Expressions #17 and #18 deliver a table similar to the TI-table (increment = 0.1). We can extract the coordinates for the desired solution curves or at first extract some rows of the table to have an estimation for the time needed until opening the parachute (after 1800 m):

$$\#17:\quad klaus := RK\left(\left[v,\ 9.81 - \frac{0.45 \cdot 0.095}{160} \cdot 1.29 \cdot 0.95^{(2000 - s)/100} \cdot v^2\right],\ [t, s, v],\ [0, 0, 0],\ 0.1,\ 500\right)$$

$$\#18:\quad egon := RK\left(\left[v,\ 9.81 - \frac{0.45 \cdot 0.75}{160} \cdot 1.29 \cdot 0.95^{(2000 - s)/100} \cdot v^2\right],\ [t, s, v],\ [0, 0, 0],\ 0.1,\ 500\right)$$

The first column shows the time t, the second one the distance s and the third one gives the velocity v.

We plot time the time-distance diagram, then the time-velocity curve and finally the velocity-distance graph by selecting the respective columns of the matrices `klaus` and `egon`.

$$\#19:\quad SELECT(1750 \le v_2 \le 1850,\ v,\ klaus)$$

$$\#20:\quad \begin{bmatrix} 20 & 1763.940123 & 153.3687157 \\ 20.1 & 1779.289996 & 153.6273766 \\ 20.2 & 1794.665328 & 153.8779058 \\ 20.3 & 1810.065305 & 154.1202846 \\ 20.4 & 1825.489112 & 154.3544954 \\ 20.5 & 1840.935932 & 154.5805225 \end{bmatrix}$$

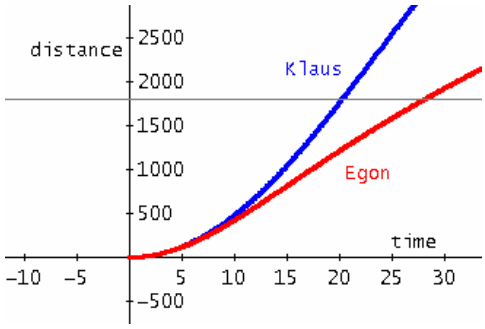
$$\#21:\quad SELECT(1790 \le v_2 \le 1810,\ v,\ egon)$$

$$\#22:\quad \begin{bmatrix} 28 & 1793.353126 & 67.26166913 \\ 28.1 & 1800.072986 & 67.13558896 \\ 28.2 & 1806.780255 & 67.00984896 \end{bmatrix}$$

$$\#23:\quad klaus\ COL\ [1, 2]$$

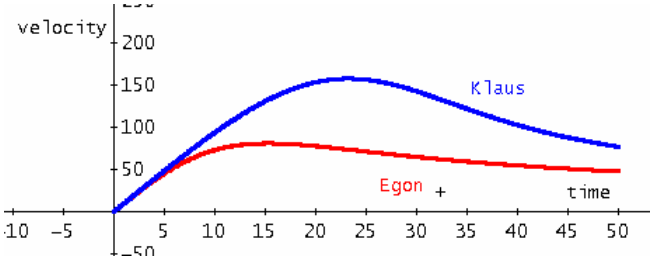
$$\#24:\quad egon\ COL\ [1, 2]$$

$$\#25:\quad 1800$$



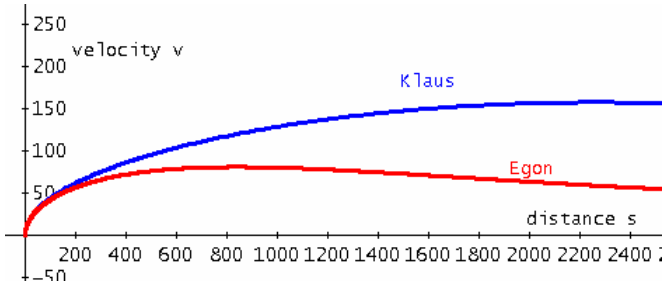
#26: klaus COL [1, 3]

#27: egon COL [1, 3]



#28: klaus COL [2, 3]

#29: egon COL [2, 3]



Endexamination 2001, Gymnasium Amstetten, Lower Austria

a) The acceleration acting on a parachutist is equal to the acceleration due to gravity minus the reverse acting brake acceleration caused by the air, which is proportional to the square of the fall velocity (prop. factor = 0.006).
Set up the respective differential equation and give its solution as $v(t)$ with $v(0) = 0$.
($g \approx 10$m/s; Work in Exact Mode!)

b) Find the respective time-distance function with $s(0) = 0$: How far did the parachutist fall until the parachute opened, if he stayed 5 seconds in free fall? Which is his fall velocity at the moment when the parachute opens?

Show that $s(t) = \dfrac{\ln(\cosh(t\sqrt{b \cdot g}))}{b}$ also describes the distance falling down.

c) Which end velocity could be reached? Using the *t-s*-function find an expression for the end velocity.

d) Describe the run of velocity applying a restricted growth function. Choose an appropriate prop. factor that the function matches the velocity function from above as accurate as possible.

e) Extra Credit: Increasing altitude is accompanied by a decreasing air density, which even influences parachuting. Leaving the airplane in an altitude of 2000m above ground this influence can be modelled by

$$b(s) = 0.003 \cdot 0.95^{\frac{2000-s}{100}}.$$

Solve the resulting DE approximatetively in the DE-Mode and give a sketch of the resulting velocity function.

Remarks: This example uses the many features of the TI-92/V 200 in a high degree. The idea for this example gave a question posed by a certain Madeleine (German teacher) some time ago. She asked if a parachutist might become slower during his/her dropping down to earth. As part e) shows, he/she might because of the braking effect of increasing air density.

Part a) cocnsits of setting and (exact) solving a DE. This could be done without CAS, but then we had to leave other parts of the example.

Parts b) and c) aks at one hand for several basic Calculus techniques, and at the other hand students have to demonstrate their competence inshowing the equivalence of not elementary expressions. This compentence is very important in the frame of a CAS supported math education.

In part d) the students shall show their capability of modelling and finally

part e) makes use of the TI's capability of solving DEs approximatively.

A Repository of Mathematical Knowledge

Albert D. Rich, Applied Logician
17 August 2007

Having not heard from me in a while, Josef recently asked what I have been up to since Texas Instruments terminated development of ***Derive***. The following is my response:

No longer having to worry about customer expectations and upward compatibility has had a tremendously liberating influence on me. I am now free to follow the math wherever it leads, and it leads to some pretty interesting and unexpected places.

For example, it is now clear to me that a new arbitrary-element-of operator, along with open and closed intervals, is essential for representing the infinite sets that can arise when solving equations or simplifying Boolean expressions. This in turn leads to the need to algebraically simplify expressions involving infinite sets (including intervals) and the arbitrary-element-of operator. Since set theory provides the foundation for all mathematics, computer algebra systems (CAS) need to spend as much effort simplifying sets as they have traditionally done with scalars and vectors.

I am in the process of collecting and categorizing the elemental transformation rules (mathematical factoids) necessary to intelligently automate the simplification of mathematical expressions. Each rule must not only include the domain over which it is valid, but the circumstances under which it actually simplifies an expression. Otherwise, application of the rule can make things worse or even lead to an infinite regress. This issue is more fully described in my talk “Automating the Simplification of Mathematical Expressions” delivered at the Visit-ME 2002 conference held in Vienna, Austria [1].

The knowledge in current computer algebra systems is hard-coded in conventional programming languages. To add new knowledge to such systems, a mathematician first has to teach a programmer the required math. Then the programmer has to translate the math into computer code. Each step in this process is time-consuming and fraught with the possibility of error.

Instead, I propose building a repository of mathematical knowledge, the contents of which will be open and accessible to all. Rather than code, the knowledge will be abstracted in the form of transformation rules stored in a tree-structured discrimination net. Mathematicians and scientists will be able to add new rules to the net, generalize existing ones, and fix or delete invalid ones.

This is analogous to the way users add knowledge to Wikipedia, the popular on-line encyclopedia. But to solve problems using Wikipedia, or any reference manual for that matter, the reader must manually find and then apply the required knowledge. In the repository I envision, the knowledge is in a vastly more useful form since the rules appropriate for particular problems are automatically found and applied.

I have no desire to create and sell yet another system in the already crowded CAS market. Rather, my intent is to prove the *viability* of using a discrimination net to store mathematical knowledge. To provide a proof-of-concept of this paradigm shift, the following components needs to be built:

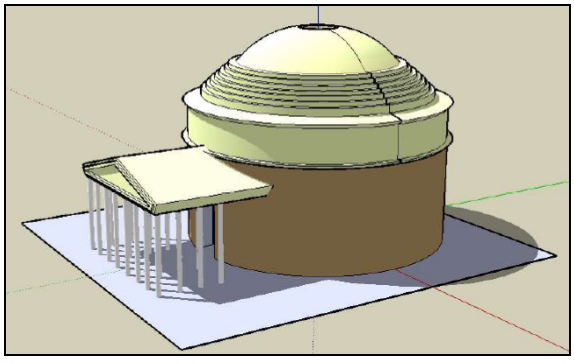
1. A "starter kit" of transformation rules sufficient for doing mathematics up to college level calculus and vector algebra. This will probably require 5000 to 6000 rules. Thus far I have collected about 4300. The rules are expressed in standard math notation as linear strings of text.
2. A computer program that can transform these text strings into parse trees, and then build a discrimination net by inserting the parse trees at the appropriate place in the net.
3. A computer program that can simplify mathematical expressions by repeatedly finding and applying applicable rules in the discrimination net until no more rules apply.

Items 2 and 3 will be a challenging, but relatively straight-forward, programming project since all the mathematical knowledge is encapsulated in item 1. They can be written in any modern programming language (e.g. Java) that provides automatic garbage collection of linked-list data structures and libraries for infinite precision arithmetic. Using such a language has the advantage of making the repository platform independent. Since the rules are stored in a discrimination net, the appropriate rule to apply can be quickly found in log n time, where n is the number of rules in the system. So efficiency should not be a problem so long as the rules are carefully crafted.

Once the advantages of such a repository become recognized, I am convinced it will be adopted as the 21st century standard for storing mathematical *and* scientific knowledge in a readily accessible form. Hopefully one or more DUG members will help me make this ambitious project a reality.

[1] You can find this lecture among the files which accompany this DNL. Thanks Bernhard Kutzler for permission to include this paper.
As I wrote in my Letter of the Editor, it would be great if DUG members would help realising Albert´s ambituous plans. Josef

Do you know Google´s SketchUp, which is a great program for modelling 3D objects. SketchUp can be downloaded for free from Google´s websites. One of my students, Dr. Erwin Rybin, delivered an excellent SketchUp tutorial and permitted its publication. You are invited to download this tutorial.It is a pdf file written in German, but it contains so many pictures that you might understand it even with little or no knowledge of German.



Pantheon in Rome, rebuilt with SketchUp.

Don Phillips provided an extended contribution on Actuarial Math with *DERIVE 6* in DNL#57. Here is the version for the handheld. Don sent a grouped file for the TI-89/TI-Titanium. It was no problem to install the files on the TI-89/Titanium and on the TI-92+/Voyage200 as well. In Don´s paper there are only TI-89 screens – according to the title of the contribution. The TI-92-screens might be a bit better readable. I don't want to print the whole paper, because there are many similarities with the DNL#57 paper but it will be among the zipped files accompanying this DNL. I´d like to give a selection of examples comparing the *DERIVE* procedures with the TI-procedures.

When Don sent the paper he wrote that he "added a few extra things that weren't in his Derive file".I tried to reproduce these "extra things" for *DERIVE*. Let´s start with Don´s introduction.

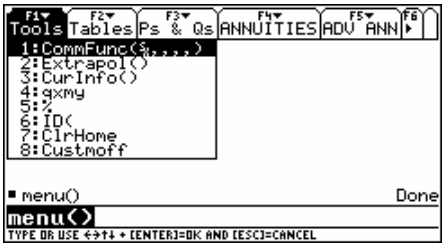
Actuarial Math on the TI-89/TI-92+/Voyage 200 – – and *DERIVE* as well

MacDonald R. Phillips
don.phillips@gmail.com
July, 2006

The TI group file, actuary.tig, provides a set of basic functions, such as D, N, S, C, M, and R, for computing the actuarial present values of life insurance and life annuities, as well as premiums and contributions. It assumes a basic working knowledge of actuarial mathematics. The calculations here represent net, not gross, premiums and contributions; i.e. they do not include the expenses of the insurance company or agent commissions, etc. (The complete TI-89 paper is among the files).

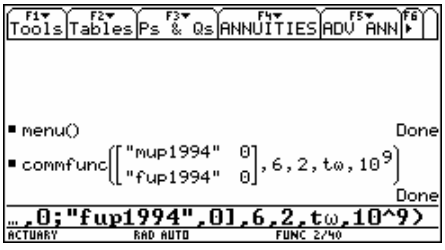
…..

To set up the custom menu for the actuarial functions, change to the ACTUARY folder and run the MENU program. Press F1, 1 to access the CommFunc().

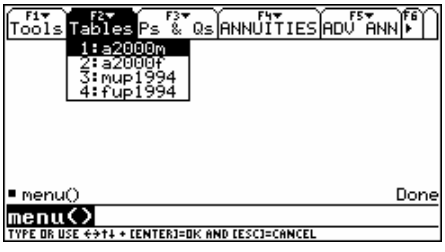


Enter the assumptions:

As noted, the mortality tables and age setbacks are entered in a 2-row matrix. Be sure to enclose the mortality table names in " ". The names of the four mortality tables included with the program are located under F2. You can use them to enter the names in the matrix. If you add other mortality tables to your calculator, you can edit the MENU program to include their names under F2.



The a2000m and a2000f tables are the annuity tables #864 and #863 in the Society of Actuaries Table Manager Software.



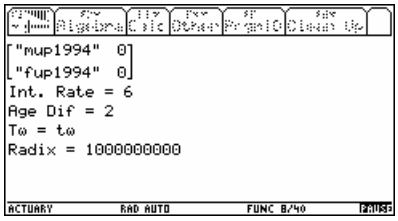
It takes a good three to five minutes to calculate all the commutation functions for X, Y, and J. As the program is running, it displays 'Processing x…', etc., until it is done.

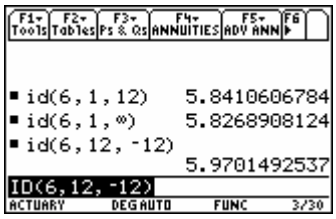
The other functions under F1: Tools are Extrapol(), CurInfo(), and ID(i, nth, mth). The Extrapol() lets you choose between a uniform distribution (udd) and a linear distribution (lin) of deaths between integral ages.

CurInfo() displays a list of the current assumptions. Press ENTER when you are done reviewing the assumptions.

ID(i, nth, mth) converts between interest rates or discount rates compounded nthly to rates compounded mthly. For instance, a 6% annual effective rate of interest is equivalent to 5.84106…% compounded monthly.

Finally, qxmy returns the age difference between x and y and % is simply the percent function. I just like having it in a convenient place and Custmoff restores the menu bar.

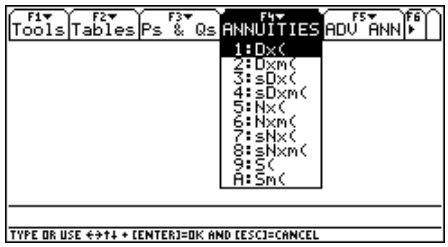




A 6% annual effective rate of interest is equivalent to 5.84…% compounded monthly, to 5.82…% compounded continuously and to a 5.97…% discount rate compounded monthly.

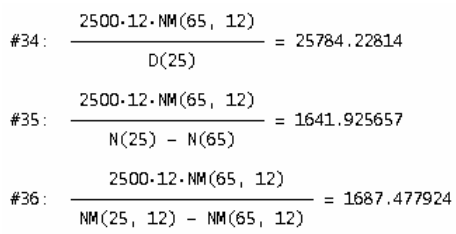
1. Life Annuities

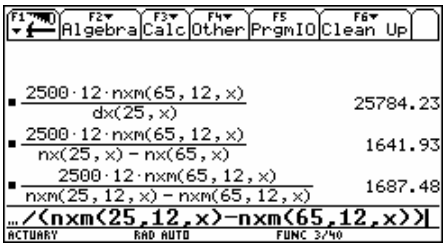
There are 10 commutation functions for basic annuities. They all take at least two arguments: age and person. Person is entered as x, y, or j and is always the last argument entered. The three basic annuity commutation functions are Dx, Nx, and S. These are used to compute annual annuities.



Dxm, Nxm, and Sm compute annuities payable mthly, e.g., monthly, quarterly, etc. sDx and sNx are used to compute annuities payable yearly that increase by a given percentage each year; the s stands for a salary scale. And sDxm and sNxm are used to compute annuities payable mthly that increase each year by a given percent.

Example 5: Suppose X, age 25, wants to provide for a retirement annuity of $2,500 a month when he retires at age 65. What is the APV (*Actuarial Present Value*) of this deferred annuity? If X wanted to

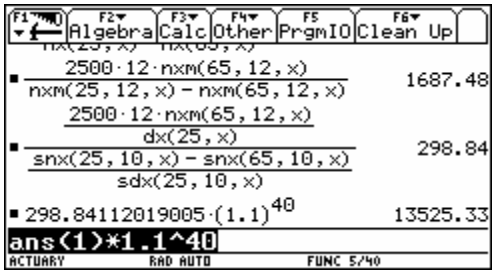




Thus, to receive $2,500 a month for life when he reaches age 65, X would only have to pay $25,784.23 at age 25. X's contribution to his retirement fund would be $1,641.93 payable at the beginning of each year until he was age 64. Or, X could make an annual contribution of $1,687.48 with $1/12^{th}$ ($140.62) payable at the beginning of each month until he was age 64 and 11 months.

Now, suppose X wanted to make annual contributions that increased by 10% a year to his deferred annuity. Here we have to use the commutation functions that incorporate a salary scale.

X's contributions begin at $298.84 when he's 25 and end at $13,525.33 when he's 64.



Let's take this calculation apart. First, the 2500*12 is the annual pension or annuity. Second, Nxm(65,12,x)/Dx(25,x) is the APV at age 25 of the pension. Next, (sNx(25,10,x) – – sNx(65,10,x))/sDx(25,10,x) is the APV at age 25 of a 40-year term annuity of 1 that increases by 10% a year. Divide the APV of the pension by the APV of the annuity of 1 gives the beginning annual contribution to fund the pension which will increase by 10% each year following the first year.

This last problem was not posed in DNL#57 and both functions snx() and sdx() don´t have their equivalents in the DERIVE utility file. I tried to add these two functions making it easier for me by assuming nontruncated tables.

$$\#40:\quad SD(age,\ sc,\ tab := x) := \left(1 + \frac{sc}{100}\right)^{age} \cdot D(age,\ tab)$$

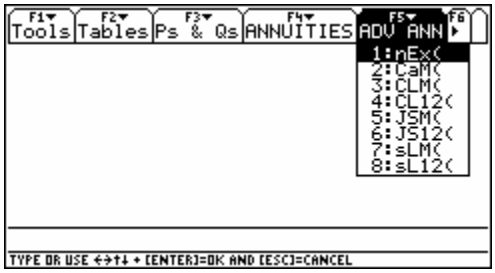
$$\#41:\quad SN(age,\ sc,\ tab := x) := \sum_{i=age}^{DIM(tab)-1} SD(i,\ sc,\ tab)$$

$$\#42:\quad \frac{\frac{2500 \cdot 12 \cdot NM(65,\ 12)}{D(25)}}{\frac{sn(25,\ 10) - sn(65,\ 10)}{sd(25,\ 10)}} = 298.8411201$$

$$\#43:\quad 298.8411201 \cdot 1.1^{40} = 13525.32662$$

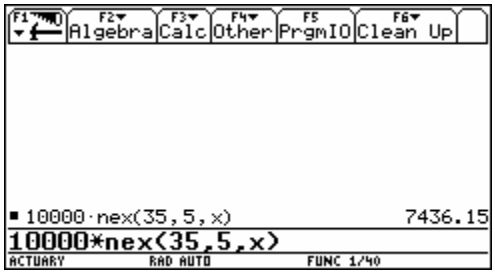
2. Advanced Annuities (all following examples are from the Handheld-Paper)

The advanced annuity functions are located under F5: ADV ANN.



Example 7: nEx(age, n, person) is the endowment function; it is the APV of 1 to be received in n years. What is the APV of a $10,000 endowment to be received by X age 35 in 5 years, if he's still alive?

nEx is not implemented in the DERIVE version, we will do this!



New Function NEX

Example 7:

$$\#74: \quad NEX(age,\ n,\ tab := x) := \frac{D(age + n,\ tab)}{D(age,\ tab)}$$

#75: 10000·NEX(35, 5) = 7436.149361

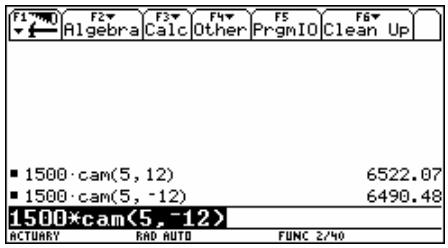
Example 8:

#76: 1500·CaM(5, 12) = 6522.070426

#77: 1500·CaM(5, −12) = 6490.477699

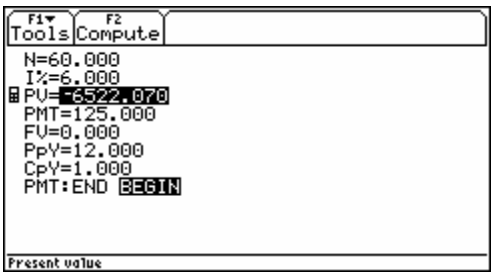
Example 8: CaM(n, mth) is the present value of an annuity due certain of 1 payable mthly. If mth is negative, it is an annuity immediate, not an annuity due. Since this is an annuity certain, it is good for all ages and persons. What is the present value of an annuity due of $1,500 payable monthly for 5 years? The annual interest rate, of course, is 6%.

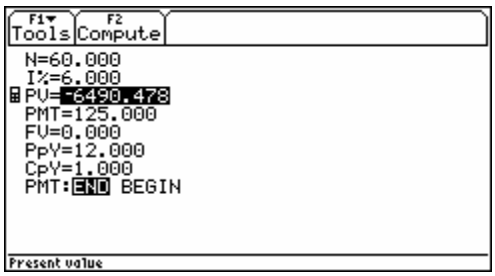
If this was an annuity immediate, the present value would be 6490.48



These values can be computed with the Finance App, too:

The annual annuity of $1,500 is payable monthly which is $125/month. It is received for 5 years or 60 months. The payments are made at the beginning of the month and the 6% interest rate is compounded annually (left) and if the payments are made at the and of the month (right):





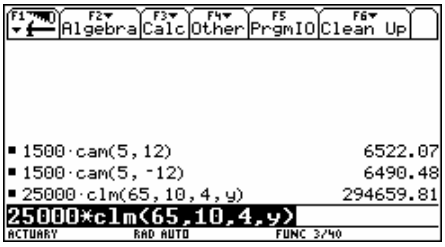
Instead of Finance App we can use TVM for Derive 6 from DNL#55:

$$TVM(60,\ 6,\ pv,\ 125,\ 0,\ 12,\ 1,\ 1) = \begin{bmatrix} N & 60 \\ IR(\%) & 6 \\ \star PV & -6522.07042711008 \\ PMT & 125 \\ FV & 0 \\ PpY & 12 \\ CpY & 1 \\ Payment & BEGIN \end{bmatrix}$$

TVM(60, 6, pv, 125, 0, 12, 1, 0) =

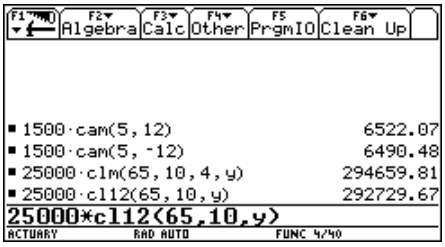
N	60
IR(%)	6
◆PV	-6490.47769871834
PMT	125
FV	0
PpY	12
CpY	1
Payment	END

Example 9: CLM(age, n, mth, person) computes the APV of an n-year certain-and-life annuity of 1 payable mthly for a person age x. What is the APV of a 10-year certain-and-life annuity of $25,000 payable quarterly for Y age 65? (In this case Y receives the $25,000 for 10 years, alive or dead, and for the rest of her life after 10 years if alive.)



The APV is $ 294 659.81.

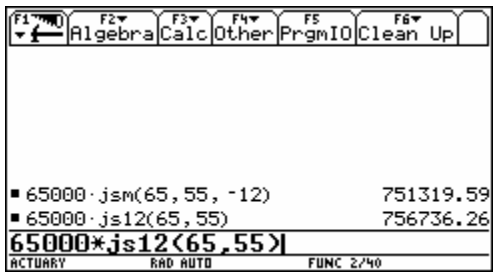
Example 10: CL12(age, n, person) computes the APV of an n-year certain-and-life annuity of 1 payable monthly for a person age x. If the above annuity were payable monthly instead of quarterly, the APV would be $292,729.67.



Example 11: JSM(age, percent, mth) computes the APV of a joint-and-survivor annuity payable mthly. It assumes that the annuity belongs to person X and person Y receives a survivor annuity if X dies. Age is the age of X, percent is the percentage of X's annuity Y receives if X dies, and mth is how often the annuity is paid each year. For example, X age 60 retires and receives a pension of $65,000 a year payable monthly at the end of each month. X's partner Y receives 55% of X's pension if X dies. X is 2 years older than Y. What is the APV of the pension?

The APV is $751,319.59. Notice that mth is negative. This is because the annuity is paid at the end of the month instead of the beginning of the month.

JS12(age, percent) computes the APV of a joint-and-survivor annuity payable mthly at the beginning of the month. What is the APV of the above pension?



The APV is $756,736.26, over 5,000 more than if it was paid at the end of the month. This is reasonable since X gets his pension sooner rather than later.

25000·CLM(65, 10, 4, y) = 294659.8150

25000·CL12(65, 10, y) = 292729.6685

65000·JSM(65, 55, -12) = 751319.5914

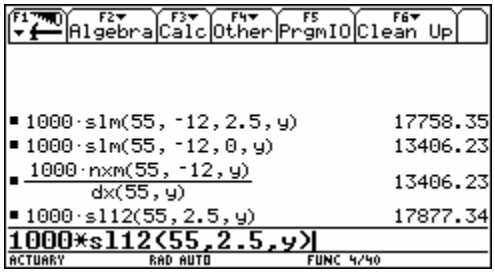
65000·JS12(65, 55) = 756736.2578

Example 12: sLM(age, mth, scale, person) computes the APV of a life annuity that increases each year by a scale factor and is payable mthly. Y is age 55 and purchases a life annuity of $1,000 that increases by 2.5 percent a year and is payable at the end of each month. What is the APV of this annuity?

The APV is $17,758.35. If there was no scale factor, the APV would be $13,406.23. This can be confirmed with the basic commutation functions.

If this annuity was paid monthly at the beginning of each month, the function sL12(age, scale, person) could be used to compute it.

As usual, the APV is a little higher because the payments come at the beginning instead of the end of the month.



SLM and SL12 are new!

$$1000 \cdot SLM(55, -12, 2.5, y) = 17758.35241$$

$$1000 \cdot SLM(55, -12, 0, y) = 13406.22575$$

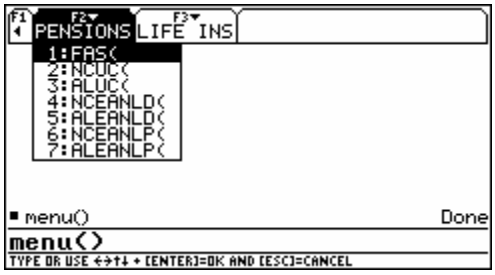
$$\frac{1000 \cdot NM(55, -12, y)}{D(55, y)} = 13406.22575$$

$$1000 \cdot SL12(55, 2.5, y) = 17877.33719$$

3. Pensions

Under F6 are some basic functions for computing the normal cost and actuarial liability of basic pension.

(These functions are not included in the actuarial file from DNL#57. Some of them make also use of the TVM-file from DNL#55.)



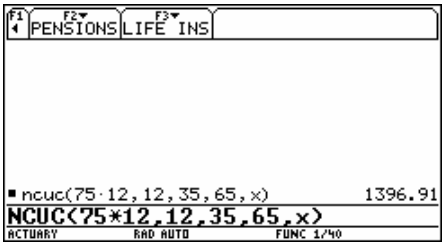
The normal cost (NC) of a pension for any given year is the actuarial value of the part of the total pension benefit assigned to the year following the valuation date, assuming valuation at the beginning of the year. The actuarial liability (AL) is the current value of past normal costs.

Example 13: Traditional Unit Credit (TUC)

The TUC cost method is usually used with pension plans that provide a flat pension benefit, such as $50 a month per year of service. The normal cost of this type plan is computed with the NCUC(bx, mth, cage, rage, person). 'bx' is the annual pension benefit earned in the following year, 'mth' is how often the pension is paid, 'cage' is the persons current age, 'rage' is the expected retirement age, and 'person' is X, Y, or J.

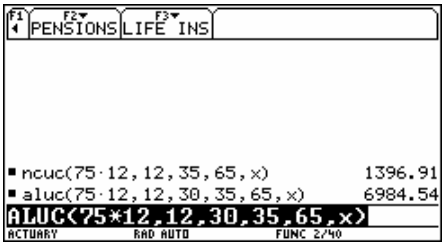
X earns a pension benefit of $75 a month for each year of service. If X age 35 expects to retire at age 65, what is the NC of his pension in the following year? What is the pension plan's actuarial liability if X was hired at age 30?

The NC is $1,396.91. Notice that this is the amount that must be put into the pension plan at the beginning of the year to fund the benefit earned during the year. Since the benefit is yet to be earned, there is no liability associated with it until the end of the year.

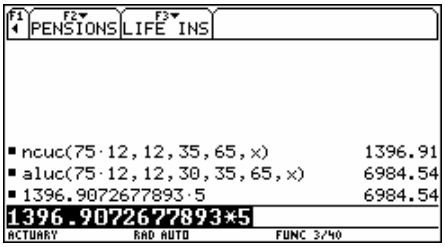


To find the actuarial liability use the

ALUC(bx, mth, eage, cage, rage, person) function where 'eage' is the entry age into the pension plan.



The AL is $6,984.54. This is 5 times the NC for X's 6th year of employment. There is no liability for the 6th year because it has yet to be earned.



And in DERIVE:

NCUC(75·12, 12, 35, 65, x) = 1396.907267

ALUC(75·12, 12, 30, 35, 65, x) = 6984.536338

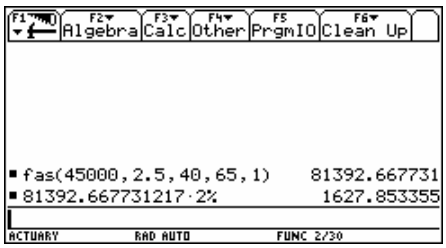
1396.907267·5 = 6984.536335

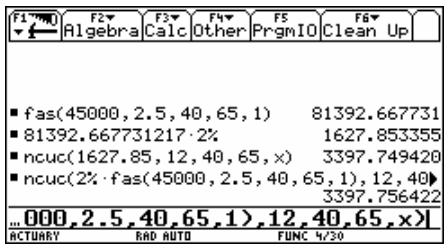
Example 14: Projected Unit Credit (PUC)

The PUC cost method adds the use of a salary scale to the TUC method. The PUC assumes that the pension benefit earned in any year is based on the final salary or final average salary. Final salary is the salary in the year preceding retirement.

X age 40 earns a pension benefit each year equal to 2% of his final salary. His current salary is $45,000 and is expected to increase by 2.5% a year. He was hired at age 30 and expects to retire at age 65. What is the NC of his benefit earned in the following year and what is the AL?

The NCUC() and ALUC() functions along with the final average salary function, FAS(cs, g, cage, rage, yrs). 'cs' is the current salary, 'g' is the expected growth rate of the salary, 'cage' is the current age, 'rage' is the retirement age, and 'yrs' is the number of year's salary to be averaged. Use 1 for 'yrs' if only the final salary is to be used.

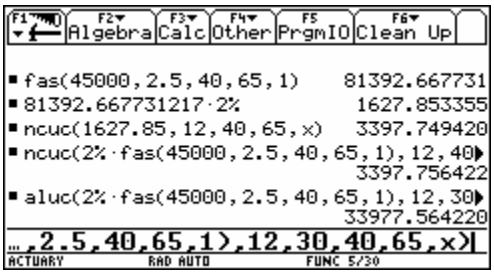




The final salary is $81 392.67. The pension benefit earned in the following year is 2% of the final salary or $1 627.85. The NC is therefore $3 397.76.

Everything can be put together at once.

The AL is $33 977.56.



Example 15: Entry Age Normal (EAN) Cost Method

The Entry Age Normal normal cost is such that at age e, the entry age into a pension, the present value of all future normal costs equals the present value of all future benefits. The entry age normal method requires an estimate of a person's full retirement benefit.

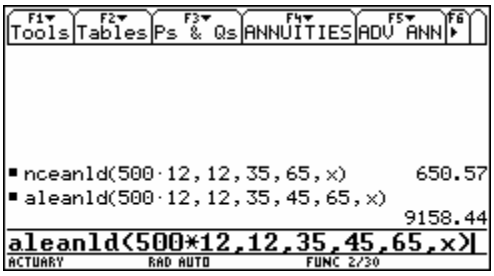
X age 45 plans to retire at age 65 and receive a pension of $500 a month. He was hired at age 35. What are the NC and AL?

To compute the NC use the NCEANLD(br, mth, eage, rate, person) function where eage is the entry age into the pension plan. The LD stands for level dollar; i.e. the normal cost will be the same amount every year (unless assumptions change and then a revised normal cost is computed). (If assumptions change and normal costs are revised, a supplemental liability may be created which may be funded through supplemental costs added to the normal costs.)

The normal cost is $650.57.

ALEANLD(br, mth, eage, cage, rage, person) is used to compute the actuarial liability.

The AL is $9,158.44.



The Entry Age Normal cost method usually uses a salary increase function when the benefit is based on a final salary or final average salary. When salaries are assumed to increase, the normal cost is defined as a level percentage (LP) of salary. When this is the case NCEANLP(br, mth, sc, eage, cage, rage, person) and ALEANLP(br, mth, sc, eage, cage, rage, person) are used to calculate the NC and AL.

X age 35 plans to retire at age 65 and receive 1% of his final monthly pay rate per year of service. His current salary is $2,000 per month and he was hired at age 25. His salary increases 5% each year. What is the normal cost and actuarial liability?

X's final average salary is $98,787.25,

His annual retirement benefit is $39,514.90 (1% of final salary times forth years of service).

And the NC for the year is $1,700.50.

This represents 7.085 percent of his current salary. Assuming salary projections do not change, the NC of future years will be 7.085 percent of the projected salary for that year.

The AL is $18,008.99.

You can check all calculations on the next screen shots. Needs some calculation time on the handheld.

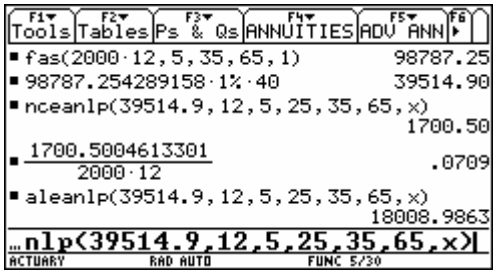
FAS(2000·12, 5, 35, 65, 1) = 98787.25428

98787.25428·1%·40 = 39514.90171

NCEANLP(39514.9, 12, 5, 25, 35, 65) = 1700.500461

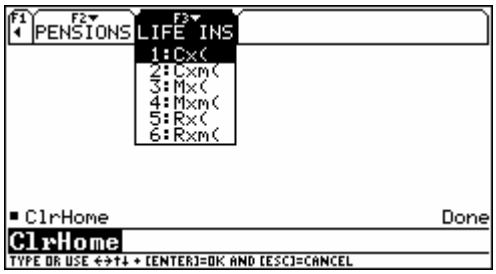
$$\frac{1700.500461}{2000\cdot 12} = 0.07085418587$$

ALEANLP(39514.9, 12, 5, 25, 35, 65) = 18008.98627



3. Life Insurance

Premiums for life insurance may be calculated with the functions under F6 and then F3: LIFE INS.



The basic life insurance functions are Cx(age, person), Mx(age, person) and Rx(age, person). Mx() is the sum of Cx()'s from age to the end of the mortality table. And Rx() is the sum of Mx()'s from age to the end of the mortality table.

Cx() is defined as $(l_x - l_{x+1})*v^{x+1}$, where l_x is the number of persons living at age x. Cx() therefore represents the number of persons dying from age x to age x + 1 times an interest rate factor evaluated at age x + 1. It is used to calculate the value of life insurance paid at the end of the year of death. Cx() is rarely used in life insurance calculations because most people buy more than just a one-year term insurance. Mx() is mostly used.

Example 16: Whole Life Insurance

What is the APV or net single premium of a whole life insurance policy of $100,000 payable at the end of the year of death for Y age 25? The APV of a whole life policy is given by Mx/Dx.

The APV of the policy is $4,460.37.

If Y wanted to pay annual premiums on this policy at the beginning of each year for as long as she lives, what would they be?

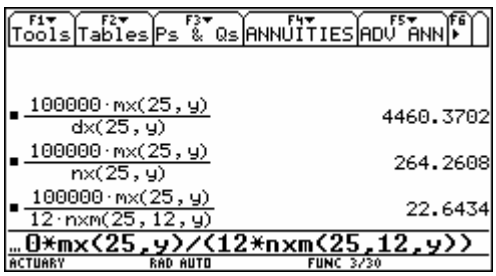
The annual premiums would be $264.26. Notice that Mx is divided by Nx where Nx represents a life annuity.

Monthly premiums payable at the beginning of each period would be $22.64.

$$\frac{100000\cdot M(25,\ y)}{D(25,\ y)} = 4460.370167$$

$$\frac{100000\cdot M(25,\ y)}{N(25,\ y)} = 264.2607926$$

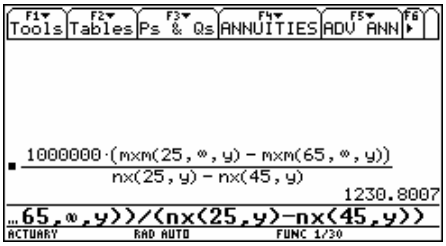
$$\frac{100000\cdot M(25,\ y)}{12\cdot NM(25,\ 12,\ y)} = 22.64336979$$



Example 17: Term Insurance

Y want to insure her life for $1,000,000 during her working years from age 25 to age 65, payable at the moment of death. But she only wants to pay annual premiums for 20 years. What is the amount of the annual premium? For this is example use the Mxm(age, mth, person) function with mth set to ∞. For life insurance, mth determines when the benefit is paid. A positive number means that it is paid at the end of the period of death. A negative number means that it is paid at the beginning of the period of death. For example, if mth = 12, the death benefit is paid at the end of the month of death. If mth = 1, it is paid at the end of the year of death. And if mth = ∞, it is assumed to be paid at the moment of death. If you noticed that the sign of mth means the reverse for life annuities, you are correct. Again, this is a consequence of the mathematical definition of the actuarial functions.

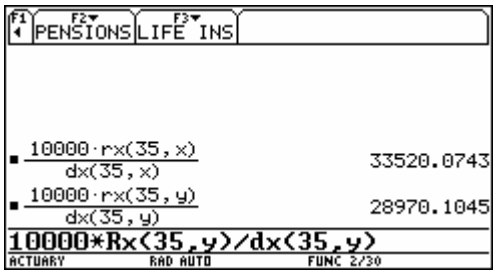
$$\frac{1000000\cdot(MM(25,\infty,y)-MM(65,\infty,y))}{N(25,y)-N(45,y)} = 1230.800745$$



The annual premium is $1,230.80.

Example 18: Increasing Insurance

What is the APV of a life insurance policy that pays $10,000 the first year if death occurs and increases by $10,000 each year after to X who is age 35? What about Y who is age 35 (DNL#57)?



$$\frac{10000\cdot R(35)}{D(35)} = 33520.07425$$

$$\frac{10000\cdot R(35,y)}{D(35,y)} = 28970.10447$$

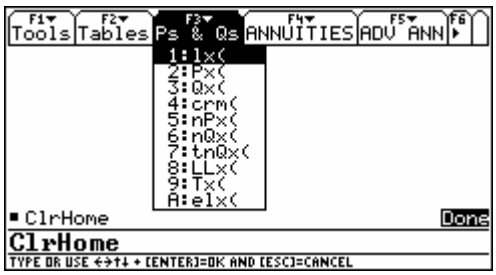
The APVs are $33 520.07 and $28970.10.

5. Non-integral Ages and the Uniform Distribution of Deaths

All the commutation functions accept non-integral ages, e.g. 50.5 or 32.3. Their values are calculated on the assumption that deaths during the year follow a uniform distribution (the variable Extrapol() is set to udd). However, if you want a straight liner interpolation between non-integral ages, set Extrapol() to lin. If non-integral ages are used with function such as Nxm() or Mxm() where abs(mth) > 1 and assuming udd, there are small relative errors in the calculations. Towards the end of the mortality table the errors can become large. If non-integral ages are used only when abs(mth) = 1 or if only integral ages are used when abs(mth) > 1, there is no error given the uniform distribution of deaths assumption.

7. Q's and P's

The Q's and P's associated with actuarial math are the probabilities of dying and living. (In actuarial textbooks they are always lower case. I use upper case because the TI-89 does not do subscripts.) The Q's and P's menu is under F3.



lx(age, person) returns the number of persons living at the beginning of age x for person X, Y or J. The ages extend from age α to age ω, the beginning and ending ages of the current mortality table. The number of persons at age α is the number chosen for the radix of the mortality table, in this case 1,000,000,000. For instance, the number of persons X living at the beginning of age 65 is 866,252,268.952.

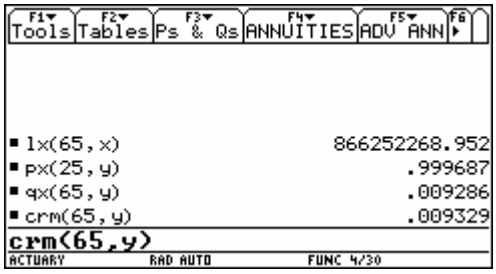
Px(age, person) gives the probability of a person of age x living one year. For Y age 25 the probability of living one year is 0.999687. $Px() = l_{x+1}/l_x$

Qx(age, person) gives the probability of a person of age x dying in the next year. For Y age 65 the probability of not being alive at age 66 is 0.009286. $Qx() = (l_x - l_{x+1})/l_x$

crm(age, person) returns the central rate of mortality of a person of age x. For Y age 65 the central rate of mortality is 0.009329316014. $crm() = (l_x - l_{x+1})/LLx(age, person)$ where LLx() is defined below.

crm for DERIVE is new!

```
lx(65) = 866252268.9
Px(25, y) = 0.9996869999
Qx(65, y) = 0.009285999913
crm(65, y) = 0.009329315926
```



nPx(age, n, person) gives the probability of a person of age x living n years. For X age 34 the probability of reaching age 58 is 0.945032212698.

$$nPx() = l_{x+n}/l_x$$

nQx(age, n, person) of course is the probability of a person of age x dying before reaching age x+n. For X age 34 the probability of dying before reaching age 58 is 0.054967787302.

$$nQx() = (l_x - l_{x+n})/l_x$$

tnQx(age, t, n, person) returns the probability of a person of age x living t years and then dying in the next n years. For Y age 34 the probability of living 10 years and they dying in the next 5 years is 0.005590571447.

$$tnQx(age, t, n\ person) = nPx(age, t, person) * nQx(age+t, n, person)$$

LLx(age, person) gives the number of life-years lived by the l_x persons who attain age x over the year from age x to x+1. On the assumption that deaths are uniformly distributed, $LLx() = (l_x + l_{x+1})/2$. For example, the number of life-years lived over the year by the X persons who attained age 45 is 970,159,731.358 years.

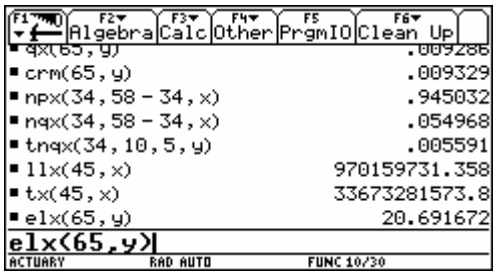
Tx(age, person) returns the total future lifetime of the l_x persons who attain age x. The total future lifetime of the X persons who attained age 45 is 3,367,321,573.8 years.

$$Tx() = l_x/2 + l_{x+1} + \ldots + l_\omega$$

elx(age, person) gives the complete expectation of life of a person age x. For Y age 65 the expected life is 20.69 years.

$$elx() = Tx()/lx()$$

```
nPx(34, 58 - 34) = 0.9450322126
nQx(34, 58 - 34) = 0.05496778731
tnQx(34, 10, 5, y) = 0.005590571399
llx(45) = 970159731.3
tx(45) = 33673281571.2
elx(65, y) = 20.69167200
```



7. Conclusion

I hope you've enjoyed this brief excursion into actuarial mathematics. This is only a taste of what the field has to offer. If you have any questions, comments, suggestions please email me at don.phillips@gmail.com.

TIME 2008 GOES AFRICA

I have the pleasure to present the program committess for the two strands of this conference. Co-Chairs are Michel Beaudin & Josef Böhm / Berhard Kutzler & Vlasta Kokol-Voljc.

8th DERIVE & TI-Nspire Conference

Bill Bauldry, USA
Johann Engelbrecht, South Africa
José Luis Galan, Spain
Renée Gossez, Belgium
Ansie Harding, South Africa
René Hugelshofer, Switzerland
Giora Mann, Israel
Gilles Picard, Canada
Don Porzio, USA
Wolfgang Pröpper, Germany
Otto Wurnig, Austria

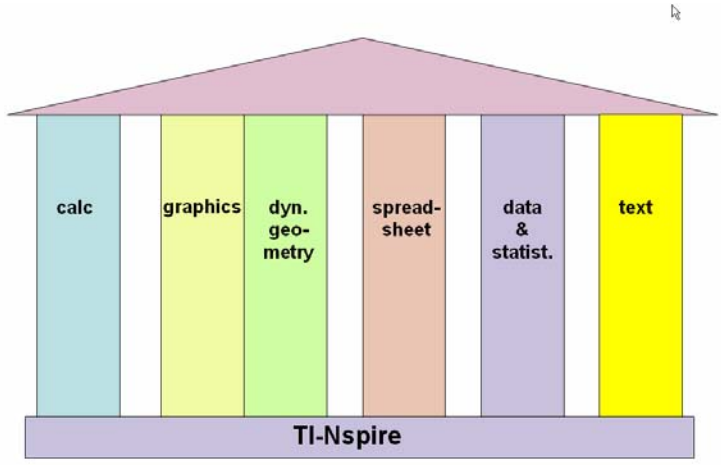
10th ACDCA Summer Academy

Rainer Heinrich, Germany
Wilfried Herget, Germany
Peter Jones, Australia
Djordje Kadijevic, Serbia
Ed Laughbaum, USA
Zsolt Lavicza, UK
Kathleen Pineau, Canada
Csaba Savari, Hungary
Spario Soon, Singapore
Marlene Torres-Skoumal, Austria
Yuzita Yaacob, Malaysia
Rose Zbiek, USA

TI-Nspire CAS = Successor of Derive!?

by B Kutzler

TI-Nspire CAS is a powerful new mathematics tool suited for teaching and learning. It combines several applications under one roof: A computer algebra system (CAS), a graphing software, a dynamic geometry system, a spreadsheet, a text editor, and a data & statistics package.



The **Calculator** application is a computer algebra system (CAS) based on the code of the handheld Voyage200.

The **Graphs & Geometry** application is a double application. It is a graphing application as known from graphing calculators such as the TI-83 or TI-84. It is also a dynamic geometry application as known from dynamic geometry products such as Cabri Geometry. Having both applications integrated with each other creates a powerful "geo-graphical" environment which goes far beyond what the individual applications usually offer.

The **Lists & Spreadsheet** application is a spreadsheet as known from programs such as Microsoft Excel combined with a list editor as known from the Voyage200.

The **Notes** application is an easy-to-use word processor which allows modes and styles particularly useful for annotating mathematical and educational work.

The **Data & Statistics** application is an easy-to-use package for statistical analysis with features as known from programs such as Fathom.

The seamless integration of these key technologies is one of the major strengths of TI-Nspire CAS. The individual technologies have been on the market for quite some years and have been tested in the classroom. Literature is available on that and there is strong evidence for each of these technologies that their proper integration into teaching and learning is beneficial for students. TI-Nspire CAS integrates all of these technologies into a powerful mathematics teaching and learning environment. This integration of well established technologies into one tool is a quantum leap and takes us into a new dimension of supporting students in understanding and learning mathematics.

⌛

TI-Nspire CAS is the proclaimed successor of Derive. Some Derive users deny this with the argument, that TI-Nspire CAS still lacks several of the Derive features. Is TI-Nspire CAS the successor of Derive – or is it not – or can it become its successor? I consider TI-Nspire CAS the successor of Derive and will provide my arguments in this article.

⌛

Some people agonize about the fact that Derive is not further developed. What does the history of Derive have to say about this?

Soft Warehouse was founded 1979 by Albert D Rich and David R Stoutemyer as a partnership, later it got incorporated as Soft Warehouse, Inc. with offices in Honolulu, Hawaii. Soft Warehouse released their first CAS product in 1979. It was called muMATH-79 and ran on 8080 and Z80 computers with as little as 48K bytes of memory running the CP/M operating system. Later versions were muMATH-80 and muMATH-83.

In October 1988 Soft Warehouse, Inc. released Derive as the successor of muMATH. Derive was a DOS program running on PC compatible computers with a minimum of 512K bytes of memory. In 1996 Derive for Windows was released. It had a GUI Windows interface, a 32-bit math engine kernel, and ran on PC compatible computers running MS Windows and NT. In 1999 David and Albert sold their company, and with it Derive, to Texas Instruments Incorporated.

The authors of Derive wrote in their *"Brief History of the muMATH/Derive CASs"* the following sentence: *"Rather than just refining and improving muMATH, we decided that an entire re-write was needed. Derive is the result. Instead of being written in muSIMP, Derive is written directly in LISP, specifically muLISP."*

After 9 years of muMATH the authors decided that it is time to do a quantum leap into a new technology by rewriting their CAS know how.

After 19 years of Derive it seems reasonable to do the same. Derive has been a key to changing the world of teaching and learning mathematics. Derive has changed the life of many teachers and students and has triggered many new developments, including the development of the legendary TI-92 handheld. But, the design of Derive is from the 80s, which makes it more and more difficult to keep up with the more and more sophisticated desires of users in the new millenium. It is time for a quantum leap into a new technology by doing another rewrite. And this has happened with the development and release of TI-Nspire CAS.

⧖

Some people agonize about the fact that some features of Derive are not (yet) available in TI-Nspire CAS.

muMATH had some nice features which were not available in Derive. For example, muMATH was an open source code product written in muSIMP. The user could look at the programs and functions and could write their own programs and functions. Derive was written in muLISP. The program code was not available any more to users, nor could they do any programming in the first version(s) of Derive.

It is true, that several Derive features are not (yet) available in TI-Nspire CAS. But it is also true that there are very many powerful features in TI-Nspire CAS which were not available in Derive, although quite a few of them were on the "Derive Wish List" for many years – and probably some of these features we never would have seen in future versions of Derive, simply because the design of Derive did not allow for their implementation. Later in this article we will look at the "Derive Wish List" and look at features on this list which are already available in TI-Nspire CAS.

In April 2007 a "Derive-TI-Nspire Transition Conference" was held in Austria with about 20 Derive experts from all around the world working with about 10 people from the TI-Nspire CAS development team, including David Stoutemyer. The goal of this meeting was to pin down those Derive features which should go into future versions of TI-Nspire CAS.

⧖

In 1988
- Derive was the proclaimed successor of muMATH,
- Derive was an entire re-write in another language,
- Some features of muMATH were not available in Derive Version 1.

In 2007
- TI-Nspire CAS is the proclaimed successor of Derive,
- TI-Nspire CAS is an entire re-write in another language,
- Some features of Derive are not available in TI-Nspire CAS Version 1.

History repeats!

⧖

A famous quote says: *"The revolution eats its children"*. This has proven true throughout history – and mathematics is not any different from other parts of life (even if non-mathematicians often doubt that; but, yes, mathematicians are humans, too!). While we see changes of products, there is also continuity. The continuity facor here is David Stoutemyer. David is the author of muMATH, David is the original author of Derive (until Albert

Rich took over, who is the original author of muLISP, the language in which Derive was written), and David is the original main author of the CAS code of TI-Nspire CAS. So "David's CAS Code" is the "immortal soul" which already sees its third reincarnation. Isn't this a proof that reincarnation exists? Why grief about death when evidently there is resurrection? The new "body" looks different, the new "body" speaks a different language, the new "body" sometimes acts different, but the new "body" has the same soul! The new body is very young and very strong, lives in a promising and supportive environment, and definitely has got everything it needs so that it can learn a lot and grow stronger and stronger in the years to come.

⌛

In the previous 19 years the activities around Derive created the "Derive Spirit". This spirit lives through people, hence it is still alive, it will stay alive, and it will continue to improve the way we teach and learn mathematics. The embodiment of the "Derive Spirit", the "Derive community" is united by the vision to use CAS for making mathematics more accessible to students. This vision is a pedagogical version of the original vision of the Derive authors *"to make computer algebra widely available to the masses on small computers"*.

One of the Derive community's main propositions was that the tool is only a means to reach the goal. We always advocated the use of Derive when its use supports the reaching of a (given) teaching goal – one should not use Derive (or any other tool) for its own sake. Let's stick to our vision – rather than to a product!

A glass half filled with beer can be considered half empty or half full, dependent on whether your attention is on the full half or on the empty half. It's the same here. The change from Derive to TI-Nspire CAS can be seen as having lost Derive or as having won TI-Nspire CAS – with all its current capabilities and with its potential for the future. We go out to teachers and preach innovation. We ask them to allow for changes in their way of teaching. Now we are challenged to allow for a change of the product, which, in fact, is a change to a new format of "David's CAS Code".

The Derive community has a great history of influencing (could we say "d(e)riving"?) the further development of Derive. Having had to speak to a small team of only two programmers made communication easy and decision chains short. Sometimes we saw a bug fix or a new feature over night – while other features never made it, because we were not "convincing enough" … Small teams have their advantages – and large teams have their advantages too. Clearly, in a large team decisions take longer, but with a large team there is more "power" behind development, so that more wishes can be fulfilled. And there is also more "power" behind marketing the product, including hot line support, teacher training, etc. TI-Nspire CAS definitely can become much bigger than Derive ever could have become. And this is good for our vision …

⌛

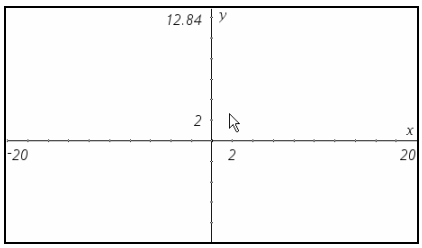
During my years of being responsible for Derive, I compiled a list of features which Derive users from around the world had asked for. Most items on this list are for the user interface and the interface to other technologies. Relatively few items are for changes to the CAS engine. This reflects the fact that many users thought that the CAS engine was pretty satisfactory for their purpose, they just wished for more comfort and more flexibility in connecting to other tools. In fact, quite a few items from this Derive Wish List would have been very difficult, if not impossible to be implemented in Derive. Some items from this list are already available in TI-Nspire CAS, others could well be seen in future versions, because the new design of TI-Nspire CAS allows for these features. Following is a selection from the Derive Wish List and my comments regarding their availability in TI-Nspire CAS.

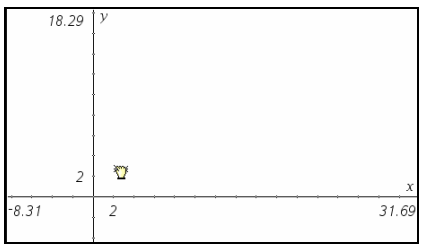
⌛

"Manipulate the graphics window with the mouse (change scales, move origin, …)":

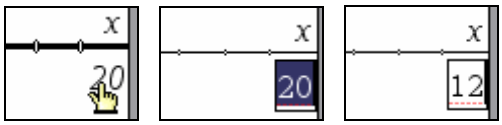
The TI-Nspire Graphing application offers a lot of options for changing the window settings.

- You can grab the plane and shift it, thus panning it within the window:

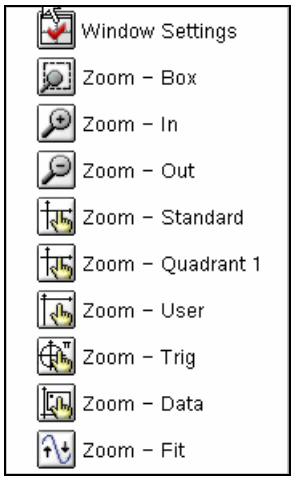




- You can click on an axis' maximum or minimum value, and then change the value via the keyboard:

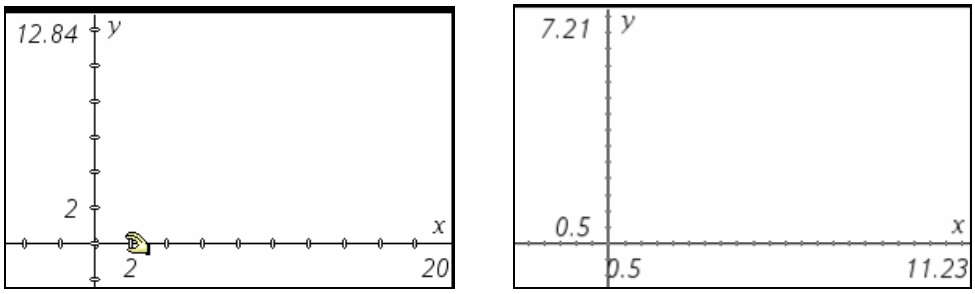


- You can use the Window Settings command from the Window menu, and then enter the values you want.

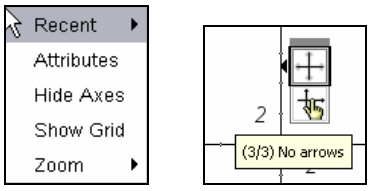


- You can use one of the many zoom commands from the Window menu.

- You can grab any of the tick marks of an axis, and then change the scale factor by dragging it:



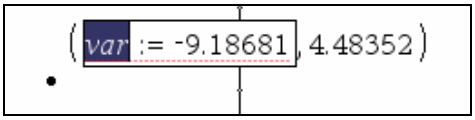
- You can click on the axis, and then right click for the context sensitive menu. The **Attributes** command opens a graphical menu in which you can change the appearance of the axis (display it with or without arrows). This menu also offers previews to the most important zoom commands from the **Window** menu.



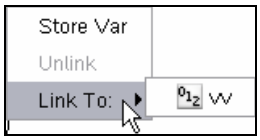
"Data exchange with Cabri products":
Cabri Geometry is the technological basis of the Graphs&Geometry application, so basically now you have both technologies (and more) within one product.

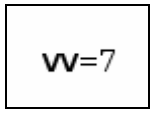
"Simple drawing options with ruler, compass, and protractor":
This is exactly what the Graphs&Geometry application offers through the Cabri technology.

"More links between the algebra and the graphing window (such as transferring graphical data)":
In an TI-Nspire Graphing or Geometry window you can store a value such as a coordinate or a measurement (of a length, an area, a slope, an angle, or an integral) in a variable.



This variable and, hence, its value, can be used in any other TI-Nspire application.
Alternatively, you can link a value in a Graphs&Geometry text box to any variable which was defined in any other TI-Nspire application.



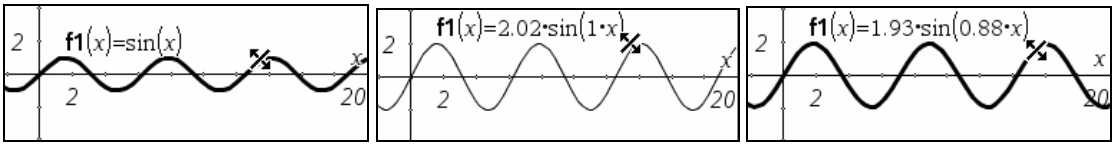


This variable/value can be transferred to geometrical objects as a distance. Hence, distances in a Graphing or Geometry window can be controlled from any other application.

⧖

***"Update the algebra expression when moving the graph"**:*

In TI-Nspire you can grab a curve and change it. The function's equation will change with it.



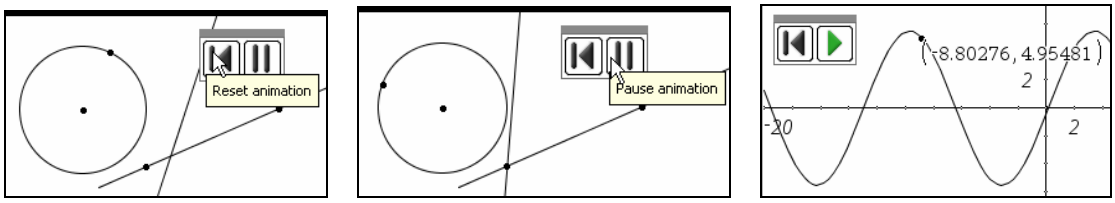
This feature is available for the following classes of functions:

- linear function $y = b$
- linear function $y = a \cdot x + b$
- quadratic function $y = a \cdot (x - b)^2 + c$
- exponential function $y = e^{(a \cdot x + b)} + c$
- exponential function $y = b \cdot e^{a \cdot x} + c$
- exponential function $y = d \cdot e^{a \cdot x + b} + c$
- logarithmic function $y = a \cdot \ln(c \cdot x + b) + d$
- sinusoidal function $y = a \cdot \sin(c \cdot x + b) + d$
- cosinusoidal function $y = a \cdot \cos(c \cdot x + b) + d$

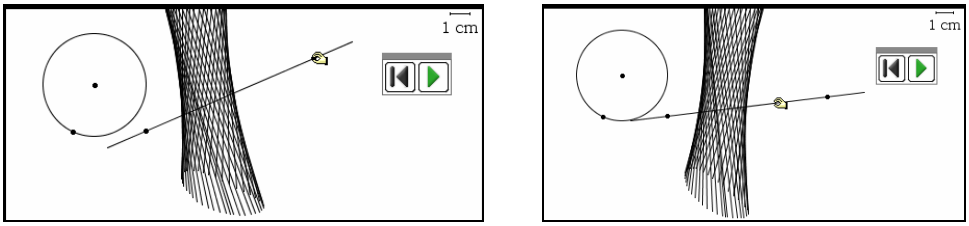
⧖

***"Animation of graphics"**:*

The Graphs&Geometry application offers a tool for animation. This tool can be applied to "independent" points on an object such as a segment, a ray, a line, a circle, or a curve. By the nature of the dynamic geometry application, all objects dependent on this point will be animated together with the point.

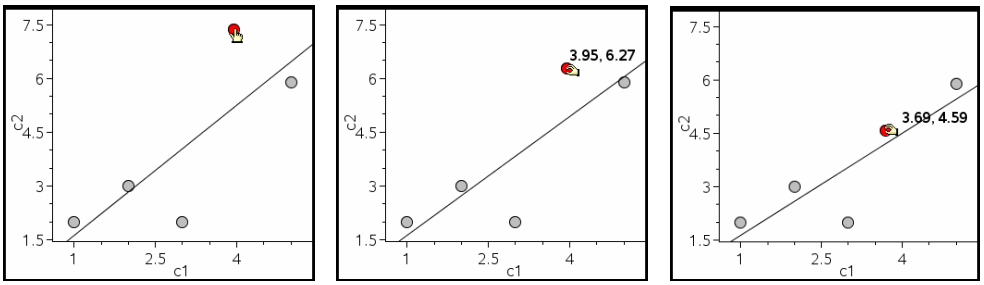


A related tool is **Locus**. It draws a family of objects generated by the movement of an object it depends on.



"Dynamic interactive graphics (change a point and the regression line changes)":

In the Data&Statistics applications you can grab points of a data plot or a scatter plot, move them, and then see the changes of dependent objects such as regression lines or regression curves. Also the numeric representations, for example in the spreadsheet, change accordingly.



"Import Excel data":

In the Lists&Spreadsheet application you can paste data which previously was copied into the clipboard from an Excel document.

	A	B
1	1	12
2	3	13
3	5	14

	A	B	C	D
1	1	12		
2	3	13		
3	5	14		

"Dynamic algebra documents (also called 'history refresh'; when you change an algebraic expression, then all objects depending on it shall be updated)":

The TI-Nspire Calculator does not offer such a feature, but one can build a problem solving template in the Lists&Spreadsheet application (in which most Calculator/CAS features are available). Therefore by the nature of a(n algebraic) spreadsheet, this gives a dynamic algebra document. The following two screen images show a template for computing the coordinates of the point of intersection of two lines given by their equations.

1	first line	2*x+3*y=5
2	second line	8*x−13*y=21
3	first variable	x
4	second variable	y
5		x=64/25 and y=...
6	x−coord intPnt	64/25
7	y−coord intPnt	-1/25

1	first line	2*a*x+b*y=7
2	second line	b*x−3*a*y=2
3	first variable	x
4	second variable	y
5		x=(21*a+2*b)/(6...
6	x−coord intPnt	(21*a+2*b)/(6*a...
7	y−coord intPnt	-(4*a−7*b)/(6*a...

The current version of TI-Nspire CAS still has limitations which prohibit an extensive exploitation of this very powerful capability. For example, cells cannot contain lists or matrices. But the mechanism is there, so all it needs is removing some of these limitations.

Other wishes which are already fulfilled in the current version of TI-Nspire CAS include:

- *"Compute all solutions of trigonometric equations"*

$$\text{solve}\left(\sin(x)=\frac{1}{2},x\right) \qquad x=2\cdot n1\cdot\pi+\frac{5\cdot\pi}{6} \text{ or } x=2\cdot n1\cdot\pi+\frac{\pi}{6}$$

- *"Offer a simple function for solving ordinary differential equations"*

deSolve(y'=y+x,x,y) $y=c1\cdot e^{x}-x-1$

It always takes courage to do quantum leaps such as from muMATH to Derive or from Derive to TI-Nspire. I guess there is nobody in the "audience" of this article who would oppose the statement that the quantum leap from muMATH to Derive was a great step forward and definitely worth having been done. This quantum leap also resulted in a significant increase of users. There are many more Derive users than there were muMATH users. With the technical and commercial power of Texas Instruments behind TI-Nspire CAS it is very likely that soon there will be many more TI-Nspire CAS users than there are Derive users. And isn't this what we wanted to achieve with all our activities of promoting the use of CAS for teaching and learning mathematics?

In a follow-up article I will demonstrate how easy it is in TI-Nspire CAS to combine its applications to create a "holistic" experience of solving an optimization problem geometrically, graphically, and algebraically.

As Bernhard mentions in his last paragraph he sent a second paper demonstrating the Nspire. At this occasion I´d like to invite all of you who have been working and experimenting with the Nspire (either handheld or PC-version) to share your findings with us. I also repeat asking for TI-92+, TI-89 and V 200 papers. Just recently checking Nils Hahnfeld´s CME (see page 3 and reproducing Josef Lechner´s "Falling Bodies" I was again fascinated by the power of the handheld. Come on and share your treasuries (and your problems, of course).
I send my "hand held regards",
Josef

I met David Sjöstrand in March when we discussed which features of DERIVE we would like to find in future versions of TI-Nspire. One of many points was, that we miss ITERATE and ITERATES as a powerful tool – even in times of programming, because of its didactical value. David presented a fine example and I asked him leaving it for the DNL. He agreed without hesitating and he sent another paper dealing also with triangles:

Hi Josef,

here is my paper again. I would be interested to know if Corollary 1 is already known. I didn´t know this until I was working with this paper. Perhaps you or some of the readers of the DNL know.

Best regards,
David

Incenter, Excenters, Orthocenter, Centroid and common Intersection Points in a Triangle

David Sjöstrand, Onsala, Sweden

Let *ABC* be a triangle. In this paper we often identify a point *P* with the vector $\overrightarrow{OP}$.

Definition 1:

line(A, B, x, y) := (A − B)·(x − A) − (A − B)·(y − A) = 0
                    2   2   1   1   1   2

D is a point lying between the points *A* and *B* on the line passing *A* and *B* dividing the segment *AB* in two parts in the ratio *a*/*b*, where $a,b>0$, counted from *B*.

Then $b(B-D)=a(D-A)\Leftrightarrow D=\frac{aA+bB}{a+b}$. It is obvious that the point $\frac{aA+bB}{a+b}$ is on the line passing *A* and *B* for any real numbers *a* and *b* such that $a+b\neq 0$

Definition 2:

A and *B* are two points in the plane and *a* and *b* are real numbers *a* and *b* such that $a+b\neq 0$, then the point $D=\frac{aA+bB}{a+b}$ divides the line segment *AB* in the ratio *a*/*b* counted from *B*.

In Defintion 2 *D* is an arbitrary point on the line passing *A* and *B*.

We will use Derive to prove

Theorem 1

The lines

$line\left(A,\frac{bB+cC}{b+c},x,y\right)$, $line\left(B,\frac{cC+aA}{c+a},x,y\right)$ and $line\left(C,\frac{aA+bB}{a+b},x,y\right)$ are concurrent and their common point of intersection is $\frac{aA+bB+cC}{a+b+c}$.

Proof:

We make the following definitions in Derive

[A := [a1, a2], B := [b1, b2], C := [c1, c2], a := a, b := b, c := c]

If we simplify

SOLUTIONS([line(C, (b·B + a·A)/(b + a), x, y), line(B, (a·A + c·C)/(a + c), x, y)], [x, y])

and

SOLUTIONS([line(A, (b·B + c·C)/(b + c), x, y), line(C, (a·A + b·B)/(a + b), x, y)], [x, y])

we receive in both cases

[[(a·a1 + b·b1 + c·c1)/(a + b + c), (a·a2 + b·b2 + c·c2)/(a + b + c)]]

which is equal to $\frac{aA+bB+cC}{a+b+c}$.

Q.E.D.

There is a converse of Theorem 1:

Theorem 2

A_1 is on the line passing *B* and *C,* B_1 is on the line passing *C* and *A* and C_1 is on the line passing *A* and *B*.

If the three lines $line(A, A_1), line(B, B_1)$ and $line(C, C_1)$ are concurrent, there are three real numbers *a*, *b* and *c*, such that

$A_1 = \frac{bB + cC}{b + c}$, $B_1 = \frac{cC + aA}{c + a}$ and $C_1 = \frac{aA + bB}{a + b}$

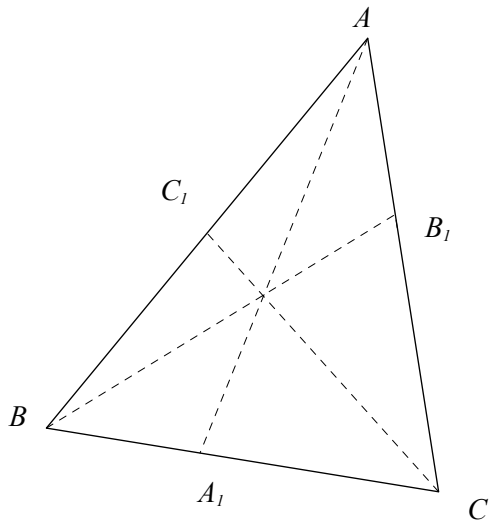


Figure 1

Proof:

Since A_1 is on the line passing *B* and *C* there exist real numbers *b* and *c* such that $A_1 = \frac{bB + cC}{b + c}$.

Since B_1 is on the line passing *C* and *A* there exist real numbers c_1 and a_1 such that

$B_1 = \frac{c_1 C + a_1 A}{c_1 + a_1} = \frac{\frac{c}{c_1}(c_1 C + a_1 A)}{\frac{c}{c_1}(c_1 + a_1)} = \frac{cC + aA}{c + a}$ where $a = \frac{c}{c_1} a_1$.

The lines $line(A, A_1), line(B, B_1)$ and $line\left(C, \frac{aA + bB}{a + b}\right)$ are concurrent according to Theorem 1.

Therefore $C_1 = \frac{aA + bB}{a + b}$.

Q.E.D.

Corollary 1

The lines $line(A, A_1), line(B, B_1)$ and $line(C, C_1)$ are concurrent if and only if $\frac{AC_1}{C_1 B} \cdot \frac{BA_1}{A_1 C} \cdot \frac{CB_1}{B_1 A} = 1$

Proof:

It follows from Definition 2 and Theorem 1 and Theorem 2 that lines $line(A, A_1), line(B, B_1)$ and $line(C, C_1)$ are concurrent if and only if there are real numbers *a*, *b* and *c* such that $\frac{bB + cC}{b + c}$ divides *BC* in the ratio *b*/*c* counted from *C*.

$\frac{cC + aA}{c + a}$ divides *CA* in the ratio *c*/a counted from *A*.

$\frac{aA+bB}{a+b}$ divides *CA* in the ratio *a*/*b* counted from *B*.

Since $\frac{AC_1}{C_1B}\cdot\frac{BA_1}{A_1C}\cdot\frac{CB_1}{B_1A}=\frac{b}{c}\cdot\frac{c}{a}\cdot\frac{a}{b}=1$ Corollary 1 is proved.

Q.E.D.

Corollary 2

If $a=|B-C|, b=|C-A|$ and $c=|A-B|$, the incenter of the triangle *ABC* is $\frac{aA+bB+cC}{a+b+c}$

Proof:

$\frac{bB+cC}{b+c}$ divides *BC* in the ratio *b*/*c* counted from *C*.

$\frac{cC+aA}{c+a}$ divides *CA* in the ratio *c*/a counted from *A*.

$\frac{aA+bB}{a+b}$ divides *AB* in the ratio *a*/*b* counted from *B*.

If $a=|B-C|, b=|C-A|$ and $c=|A-B|$ then the lines $line\left(A,\frac{bB+cC}{b+c},x,y\right)$,

$line\left(B,\frac{cC+aA}{c+a},x,y\right)$ and $line\left(C,\frac{aA+bB}{a+b},x,y\right)$ are the angle bisectors of the triangle *ABC*.

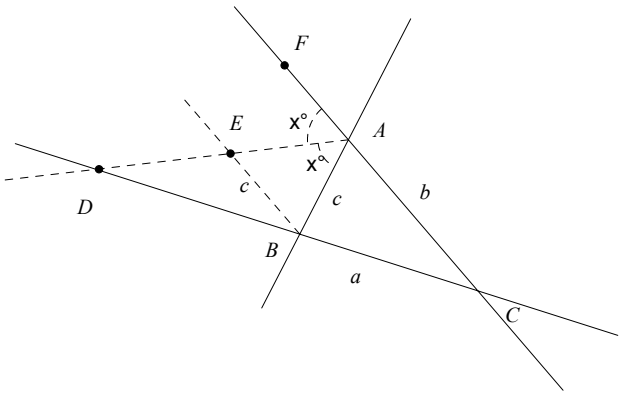
The incenter of a triangle is the common intersection point of the angle bisectors of the triangle.
Now the Corollary 1 follows from Theorem 1.

Q.E.D.

Lemma

Let *ABC* be a triangle and $a=|B-C|,\ b=|C-A|$ and $c=|A-B|$

If *AD* is the bisector of the exterior angle *BAF,* then $D=\frac{-bB+cC}{-b+c}$.



Proof:

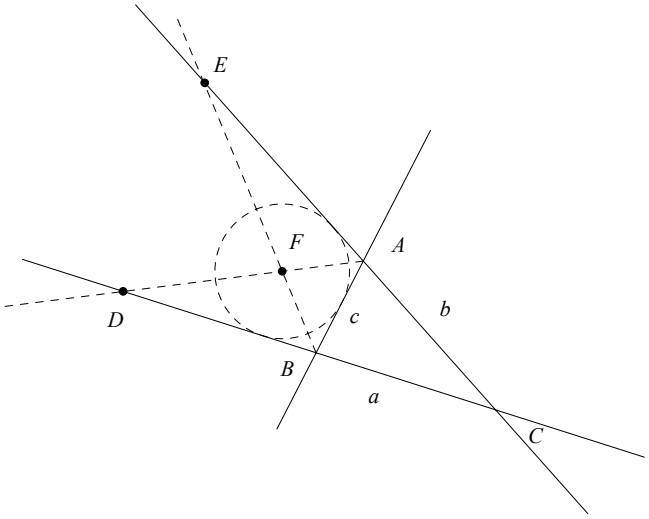
BE is parallel to *CF*. Therefore *BE* = *BA* and the triangles *BED* and *DCA* are similar which gives $\frac{CD}{BD}=\frac{b}{c}$ and consequently:

$$c(D-C)=b(D-B)\Leftrightarrow D=\frac{-bB+cC}{-b+c}$$

Q.E.D.

Corollary 3

The excenters of the triangle ABC are $\frac{-aA-bB+cC}{-a-b+c}$, $\frac{aA-bB-cC}{a-b-c}$ and $\frac{-aA+bB-cC}{-a+b-c}$.



Proof:

The excenter, F, is the intersection point of the bisector of $\angle DBA$, $line\left(A,\frac{-bB+cC}{-b+c},x,y\right)$ and the bisector of $\angle BAF$, $line\left(B,\frac{-aA+cC}{-a+c},x,y\right)$.

According to Theorem 1 the intersection point of $line\left(A,\frac{-bB+cC}{-b+c},x,y\right)$ and $line\left(B,\frac{-aA+cC}{-a+c},x,y\right)$ is $\frac{-aA-bB+cC}{-a-b+c}$.

We have to take care of the case that $b = c$. In this case the bisector of the exterior angles of A is parallel to BC. The equation of this bisector is

$$(C_2 - B_2)\cdot(x - A_1) - (C_1 - B_1)\cdot(y - A_2) = 0$$

You receive the excenter as the intersection point between this bisector and $line\left(C,\frac{bB+aA}{b+a},x,y\right)$ using Derive:

$$\text{SOLVE}\left(\left[\text{line}\left(C,\frac{a\cdot A+b\cdot B}{a+b},x,y\right),\ (C_2 - B_2)\cdot(x - A_1) - (C_1 - B_1)\cdot(y - A_2) = 0\right],\ [x,\ y]\right)$$

$$\left[x = \frac{a\cdot a1+b\cdot(b1-c1)}{a} \wedge y = \frac{a\cdot a2+b\cdot(b2-c2)}{a}\right]$$

$$\left[\frac{a\cdot a1+b(b1-c1)}{a},\frac{a\cdot a2+b(b2-c2)}{a}\right]=\left[\frac{a\cdot a1+b\cdot b1-c\cdot c1}{a+b-c},\frac{a\cdot a2+b\cdot b2-c\cdot c2}{a+b-c}\right]=$$

$$\frac{-aA-bB+cC}{-a-b+c}$$

By symmetry we conclude that the formulas for the excenters are as we have stated in Corollary 2.

Q.E.D.

Remark

If you let $a = b = c$ in Theorem 1 you receive the well known formula for the intersections point of the medians of a triangle, $\frac{aA+bB+cC}{a+b+c}=\frac{aA+aB+aC}{a+a+a}=\frac{A+B+C}{3}$

Corollary 4

The altitudes of a triangle are concurrent.

Proof:

Let AA_1, BB_1 and CC_1 be the altitudes of the triangle ABC.

Then

A_1 divides BC in the ratio $\frac{c\cos B}{b\cos C}$ counted from B.

B_1 divides CA in the ratio $\frac{a\cos C}{c\cos A}$ counted from C.

and

C_1 divides AB in the ratio $\frac{b\cos A}{a\cos B}$ counted from A.

Since $\frac{c\cos B}{b\cos C}\cdot\frac{a\cos C}{c\cos A}\cdot\frac{b\cos A}{a\cos B}=1$ the altitudes are concurrent according to Corollary 1.

Using ITERATES for finding the midpoint of a Triangle

Plot a sequence of triangles, Tn, where the vertices of Tn are the midpoints of the sides of Tn–1. T0 = P is a given triangle.

#1: $P := \begin{bmatrix} -1 & 3 \\ 2 & 3 \\ -2 & -2 \\ -1 & 3 \end{bmatrix}$

If we make the below defintion of f we have that Tn = f(Tn–1).

#2: $f(P) := \left[\frac{P_1 + P_2}{2}, \frac{P_2 + P_3}{2}, \frac{P_3 + P_1}{2}, \frac{P_1 + P_2}{2}\right]$

If we plot the below expression we receive the triangles T0, T1,... T20.

#3: ITERATES(f(x), x, P, 20)

The below expression gives the median from A to BC in the triangle ABC.

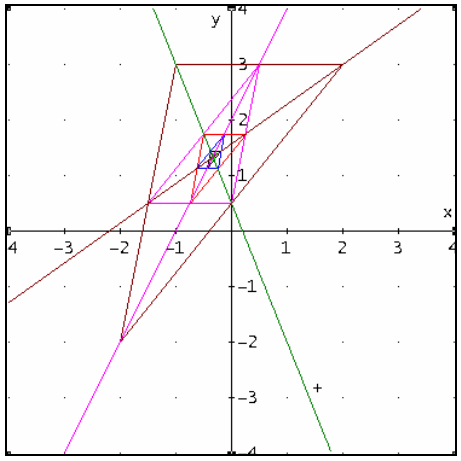
$$\#4:\quad MEDIAN(A, B, C) := y = \frac{\left(\frac{B + C}{2}\right)_2 - A_2}{\left(\frac{B + C}{2}\right)_1 - A_1} \cdot (x - A_1) + A_2$$

The below expression gives the medians of T0.

$$\#5:\quad medians := \left[MEDIAN(P_1, P_2, P_3), MEDIAN(P_2, P_3, P_1), MEDIAN(P_3, P_1, P_2)\right]$$

The medians of T0 are obviously the medians of Tn for all n.
Let Sn be point in the interior or on the boundary of Tn.
It is obvius that the sequence Sn is convergent. Let lim Sn = M. It is obvious that M is the unique point belonging to all Tn. Since all the medians intersect Tn for all n they must all pass M.
(See the plot of #5 on the next page.)

Thus we have proved that the medians of a triangle intersect in one point, M.



Let Pn,1, Pn,2 and Pn,3 be the vertices of Tn. We have that
Pn,1+Pn,2+Pn,3 = (Pn-1,1+Pn-1,2)/2 + (Pn-1,2+Pn-1,3)/2 + (Pn-1,3+Pn-1,1)/2 =
= Pn-1,1+Pn-1,2+Pn-1,3 = P0,1+P0,2+P0,3.

On the other hand lim Pn,k = M.
Therefore 3M = lim(Pn,1+Pn,2+Pn,3) = lim(P0,1+P0,2+P0,3) = P0,1+P0,2+P0,3.
We have thus proved the wellknown formula M=(A+B+C)/3.

Let M be the intersection point of the medians.
We can see that lim Tn = M when n tends to infinity.

$$\#6:\quad \frac{P_1 + P_2 + P_3}{3} = \left[-\frac{1}{3}, \frac{4}{3}\right]$$

We can verify that M is given by #6 by verifying a vertice of T20 with #6

$$\#7:\quad (ITERATE(f(x), x, P, 20))_1 = \left[-\frac{349523}{1048576}, \frac{1398103}{1048576}\right]$$

$$\#8:\quad \frac{P_1 + P_2 + P_3}{3} - (ITERATE(f(x), x, P, 20))_1 = \left[-\frac{7}{3145728}, -\frac{5}{3145728}\right]$$

Titbits 34

or Polynomial Arithmetic for the Advanced

(c) Johann Wiesenbauer, Vienna University of Technology

On quite a few occasions I have been dealing with polynomial arithmetic in the past (for example, cf. my Titbits 13, 17, 25, 26). As this is a very important and fundamental topic, indeed, hopefully you won't object to my resuming it once more. To be more precise, I would like to put together a number of basic polynomial routines that could be used in a Derive-implementation of the celebrated Schoof-algorithm for point counting on elliptic curves of the form y^2 = x^3 +ax +b mod p for some prime p, although this algorithm (and some refinements like the SEA-algorithm) are covered in depth only in a future issue of this series. Hence, to some extent this is also a continuation of my Titbits 30, where I already listed a number of important tools regarding computations on those elliptic curves.

Let's start with the 4 fundamental routines polyrem(u,v,p,x), polyquot(u,v,p,x), polygcd(u,v,p,x) and polypower(u,n,v,p,x). In all cases u and v are two polynomials in **Z**p[x], i.e. the polynomial ring over the residue class ring **Z**p for the prime p. While the first two are supposed to compute the remainder and quotient of the division u:v, the output of the third and fourth should be the gcd(u,v) and the polynomial power u^k mod v, both again in the ring **Z**p[x]. As for the last routine we assume w.l.o.g. that v is normed, i.e. has got 1 as the leading coefficient.

Using basically the routines from the Titbits 25 quoted above, a first implementation could look like this:

```
      polyrem(u, v, p, x) :=
        Prog
#1:       u := FACTOR(REMAINDER(u, v, x), Trivial, x)
          POLY_MOD(NUMERATOR(u)·INVERSE_MOD(DENOMINATOR(u), p), p)

      polyquot(u, v, p, x) :=
        Prog
#2:       u := FACTOR(QUOTIENT(u, v, x), Trivial, x)
          POLY_MOD(NUMERATOR(u)·INVERSE_MOD(DENOMINATOR(u), p), p)

      polygcd(u, v, p, x, r_) :=
        Prog
          u := POLY_MOD(u, p)
          v := POLY_MOD(v, p)
          Loop
#3:         If v = 0
              RETURN u
            r_ := polyrem(u, v, p, x)
            WRITE(POLY_DEGREE(r_, x))
            u := v
            v := r_
```

```
      polypower(u, k, v, p, x, u_ := 1) :=
        Loop
          If k = 0
            RETURN u_
#4:       If ODD?(k)
            u_ := POLY_MOD(REMAINDER(u·u_, v, x), p)
          u := POLY_MOD(REMAINDER(u·u, v, x), p)
          k := FLOOR(k, 2)
```

In fact, they do work, as the examples below show.

```
#5:   w := POLY_MOD(3·RANDOM_POLY(x, 3, 7), 7)

              3      2
#6:   w := 3·x  + 6·x  + 6·x + 5

#7:   u := POLY_MOD(4·w·RANDOM_POLY(x, 7, 7), 7)

              10    9      8      6      5      4    3      2
#8:   u := 5·x   + x  + 3·x  + 2·x  + 2·x  + 4·x  + x  + 3·x  + 6·x + 6

#9:   v := POLY_MOD(2·w·RANDOM_POLY(x, 7, 7), 7)

              10      8      7      5      4      3      2
#10:  v := 6·x   + 6·x  + 3·x  + 6·x  + 4·x  + 3·x  + 2·x  + 6·x

                                    3      2
#11:  polygcd(u, v, 7, x) = 6·x  + 5·x  + 5·x + 3

#12:  polygcd(polyquot(u, w, 7, x), polyquot(v, w, 7, x), 7, x) = 2

#13:  v := POLY_MOD(INVERSE_MOD(6, 7)·v, 7)

           10    8      7    5      4      3      2
#14:  v := x   + x  + 4·x  + x  + 3·x  + 4·x  + 5·x  + x

                                      9     8     7     6     5    4     3     2
#15:  polypower(u, 10, v, 7, x) = 2·x  + 2·x  + 5·x  + 4·x  + 5·x  + x  + 2·x  + 6·x  + 6·x + 1

                          10                  9     8     7     6     5    4     3     2
#16:  POLY_MOD(REMAINDER(u  , v, x), 7) = 2·x  + 2·x  + 5·x  + 4·x  + 5·x  + x  + 2·x  + 6·x  + 6·x + 1
```

There are a number of objections though. A minor one is that the outcome of polygcd(u,v,p,x) is usually not normed, i.e. it hasn't got 1 as the leading coefficient, as this is a common convention for gcd's of polynomials. There is an easy remedy for this problem, namely by "norming" the divisor polynomial before each division in the euclidean algorithm, as this was already carried out for v above before the computation of polypower(u,10,v,7,x)

```
      polygcd(u, v, p, x, d_, r_) :=
        Prog
          u := POLY_MOD(u, p)
          If u = 0
            RETURN POLY_MOD(v, p)
          Loop
            v := POLY_MOD(v, p)
            If v = 0
#17:          RETURN u
            d_ := POLY_DEGREE(v, x)
            WRITE(d_)
            r_ := LIM(v/x^d_, x, ∞)
            v := POLY_MOD(INVERSE_MOD(r_, p)·v, p)
            r_ := REMAINDER(u, v, x)
            u := v
            v := r_
```

```
                                    3      2
#18:  polygcd(u, v, 7, x) = x  + 2·x  + 2·x + 4
```

A far more serious problem with respect to our goal is the poor performance of the routines above, if u and v are polynomials of a "large" degree, say of degree about 1000 or even more. (For example, the corresponding computations für polygcd(u,v,p,x) would take some quarters of an hour, rather than some seconds, as the interested reader can easily check himself by following the pattern above!)

In the following I'm trying to fix this problem by setting up my own polynomial arithmetic based on the the vector representations of the involved polynomials. In order to ease the change between the polynomial representation and the vector representation of a polynomial I will first provide the routine polytovec(u,p,x) as well as its inverse vectopoly(u,x).

```
      polytovec(u, p, x) :=
        Prog
          u := POLY_MOD(u, p)
          If u = 0
             RETURN []
#19:      u := MOD(SUBST(TERMS(u + p·(x^POLY_DEGREE(u) - 1)/(x - 1)), x, 1), p)
          Loop
            If FIRST(u) > 0
               RETURN u
            u := REST(u)
```

```
                                          j_
#20:  vectopoly(u, x) := u·VECTOR(x   , j_, DIM(u) - 1, 0, -1)
```

Furthermore, we need some basic operations for polynomials represented as vectors, such as the product vecmult(u,v,p) of two polynomials u and v and the remainder vecrem(u,v,p) of the division u:v (as always in Zp[x] !)

```
      vecmult(u, v, p, s_) :=
        Prog
          If DIM(u) < DIM(v)
             [s_ := u, u := v, v := s_]
          s_ := 0·u
          v := REVERSE(v)
          Loop
            If v = [] exit
#21:        s_ := ADJOIN(0, s_ + FIRST(v)·u)
            u := APPEND(u, [0])
            v := REST(v)
          s_ := MOD(s_, p)
          Loop
            s_ := REST(s_)
            If FIRST(s_) ≠ 0
               RETURN s_
```

```
      vecrem(u, v, p, k_ := 0) :=
        Prog
          If DIM(u) < DIM(v)
             RETURN u
          Loop
            If DIM(v) = DIM(u) exit
            k_ :+ 1
            v := APPEND(v, [0])
          Loop
            u := MOD(u - FIRST(u)·v, p)
#22:        u := REST(u)
            k_ :- 1
            If k_ < 0 exit
            v := DELETE(v, -1)
          Loop
            If FIRST(u) > 0
               RETURN u
               0
               RETURN u
            u := REST(u)
```

Now I'm going to replace the routines polygcd(u,v,p,x) and polypower(u,k,v,p,x) above with the following new implementations.

```
polygcd(u, v, p, x, k_ := 0, t_, v_) :=
  Prog
    u := polytovec(u, p, x)
    v := polytovec(v, p, x)
    If DIM(u) < DIM(v)
       [t_ := u, u := v, v := t_]
    Loop
      If v = []
         RETURN vectopoly(u)
      t_ := INVERSE_MOD(FIRST(v), p)
      v := MOD(t_·v, p)
      Loop
        If DIM(v) < DIM(u)
           Prog
             k_ :+ 1
             v := APPEND(v, [0])
#23:       exit
      t_ := u
      Loop
        If FIRST(t_) > 0
           t_ := MOD(t_ - FIRST(t_)·v, p)
        t_ := REST(t_)
        If k_ = 0 exit
        v := DELETE(v, -1)
        k_ :- 1
      u := v
      v := t_
      Loop
        If v = []
           exit
           If FIRST(v) > 0 exit
        v := REST(v)
```

$$\#24:\quad polygcd(u, v, 7, x) = x^3 + 2\cdot x^2 + 2\cdot x + 4$$

```
polypower(u, k, r, p, x, y, u_ := [1]) :=
  Prog
    u := polytovec(u, p, x)
    r := polytovec(r, p, x)
    Loop
#25:  If k = 0
         RETURN vectopoly(u_)
      If ODD?(k)
         u_ := vecrem(vecmult(u, u_, p), r, p)
      u := vecrem(vecmult(u, u, p), r, p)
      k := FLOOR(k, 2)
```

$$\#26:\quad polypower(u, 10, v, 7, x) = 2\cdot x^9 + 2\cdot x^8 + 5\cdot x^7 + 4\cdot x^6 + 5\cdot x^5 + x^4 + 2\cdot x^3 + 6\cdot x^2 + 6\cdot x + 1$$

Again we got the correct result for our examples above and already now a noticable increase of performance can be seen, though the full power of these small gems of programming will become only visible when dealing with huge polynomials as the reader might again check himself.

As for our goal, namely to provide all necessary tools for a later implementation of Schoof's algorithm, we need some further tools concerning computations on elliptic curves. The first routine of this sort computes the so-called division polynomials $\psi(n,a,b,p,x,y)$. They are of the form $nf(x)y$, if n is even and of the form $f(x)$, if n is odd, where $f(x)$ is some normed polynomial in x, whose zeros are exactly the x-coordinates of all n-torsion points (x,y) of E with $y \neq 0$ in the algebraic closure of Zp.

In particular, it easily follows from this that the degree of f(x) grows like n^2/2. (To be more precise, this degree is at most (n^2-1)/2, if n is odd, and at most (n^2-4)/2, if n is even. Furthermore, those bounds are sharp if and only if p is not a divisor of n.)

```
      ψ(n, a, b, p, x, y, u_) :=
        Prog
          If n < 3
            RETURN n·y^MAX(0, n - 1)
          If n = 3
            RETURN POLY_MOD(3·x^4 + 6·a·x^2 + 12·b·x - a^2, p)
          If n = 4
            RETURN POLY_MOD((x^6 + 5·a·x^4 + 20·b·x^3 - 5·a^2·x^2 - 4·a·b·x - 8·b^2 - a^3)·4·y, p)
#27:      If EVEN?(n)
            Prog
              u_ := ψ(n/2 + 2, a, b, p, x, y)·ψ(n/2 - 1, a, b, p, x, y)^2
              u_ := ψ(n/2 - 2, a, b, p, x, y)·ψ(n/2 + 1, a, b, p, x, y)^2
              POLY_MOD(ψ(n/2, a, b, p, x, y)·u_·(p + 1)/(2·y), p)
            Prog
              u_ := ψ((n + 3)/2, a, b, p, x, y)·ψ((n - 1)/2, a, b, p, x, y)^3
              u_ := ψ((n + 1)/2, a, b, p, x, y)^3·ψ((n - 3)/2, a, b, p, x, y)
              POLY_MOD(REMAINDER(u_, y^2 - x^3 - a·x - b, y), p)
```

$$\#28:\quad \psi(6,1,2,7,x) = 6\cdot x^{16}\cdot y + 4\cdot x^{14}\cdot y + x^{9}\cdot y + 3\cdot x^{7}\cdot y + 5\cdot x^{2}\cdot y + y$$

$$\#29:\quad \psi(7,1,2,7,x) = 6\cdot x^{21} + 5\cdot x^{14} + x^{7} + 1$$

Among other things, those division polynomials can be used to compute the x- and y-coordinate of the additive power nP of a point P=(x,y) on E over the algebraic closure of Zp, where P must not be an n-torsion point. Since x and y are variables here, those coordinates are rational polynomials in x and y, where the exponent of y is again at most 1. They are computed by the following routines.

```
      nPx(n, a, b, p, x, s_, t_, y_) :=
        Prog
          s_ := ψ(n - 1, a, b, p, x, y_)·ψ(n + 1, a, b, p, x, y_)
          If ODD?(n)
#30:        s_ := (x^3 + a·x + b)/y_^2
          t_ := ψ(n, a, b, p, x, y_)^2
          If EVEN?(n)
            t_ := (x^3 + a·x + b)/y_^2
          POLY_MOD(x·t_ - s_, p)/POLY_MOD(t_, p)

      nPy(n, a, b, p, x, y, s_, t_) :=
        Prog
          s_ := ψ(n + 2, a, b, p, x, y)·ψ(n - 1, a, b, p, x, y)^2
          s_ := POLY_MOD(s_ - ψ(n - 2, a, b, p, x, y)·ψ(n + 1, a, b, p, x, y)^2, p)
#31:      t_ := 4·y·ψ(n, a, b, p, x, y)^3
          If EVEN?(n)
            t_ := (x^3 + a·x + b)/y^2
          s_/POLY_MOD(t_, p)
```

$$\#32:\quad nPx(3,1,2,7,x) = \frac{x^9 + 2\cdot x^7 + 4\cdot x^6 + 2\cdot x^5 + x^4 + 4\cdot x^3 + 5\cdot x^2 + x + 3}{2\cdot x^8 + x^6 + 4\cdot x^5 + 2\cdot x^4 + x^3 + 4\cdot x^2 + x + 1}$$

$$\#33:\quad nPy(3,1,2,7,x) = \frac{y\cdot(4\cdot x^{12} + 4\cdot x^{10} + 3\cdot x^9 + 5\cdot x^8 + 4\cdot x^7 + 5\cdot x^5 + 5\cdot x^3 + 2\cdot x^2 + x + 5)}{3\cdot x^{12} + 4\cdot x^{10} + 2\cdot x^9 + 5\cdot x^8 + x^7 + 2\cdot x^5 + 5\cdot x^3 + 6\cdot x^2 + x + 3}$$

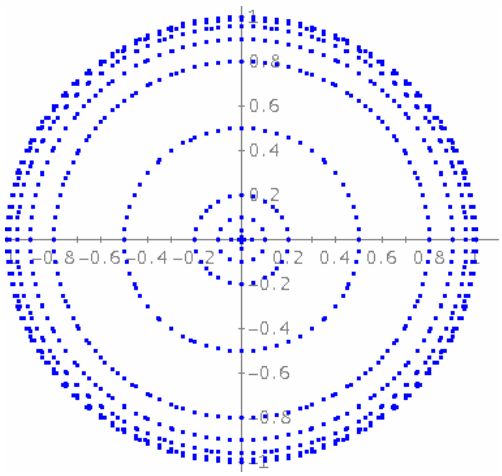
By now, we have accumulated the most important ingredients of the Schoof's celebrated algorithm for point counting on the elliptic curve E:y^2=x^3+ax+b mod p. However, as already mentioned, due to lack of time and space I will postpone a discussion and an efficient implementation of this algorithm until the next issue.

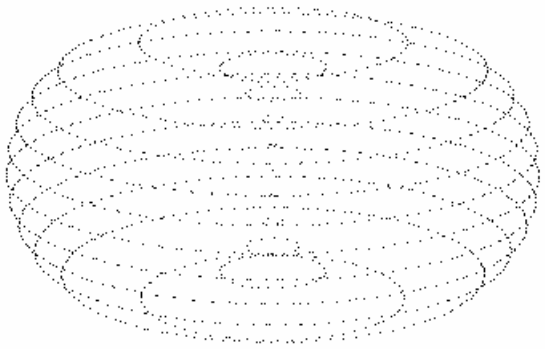
Surface #2: $(x^2+y^2+z^2)^2 = x^2+y^2$

Surface #2 in DERIVE presented applying functions from `polycontour.mth` (DNL#63) and `IMPLICIT_Peter.mth` (DNL#64).

```
                                2    2    2 2    2    2
VECTOR(ContourPts_2D((x  + y  + z )  - x  - y , 1, -1, 1, -1, 1, 0.05, 0.05), 1, -0.5, 0.5, 0.1)
```

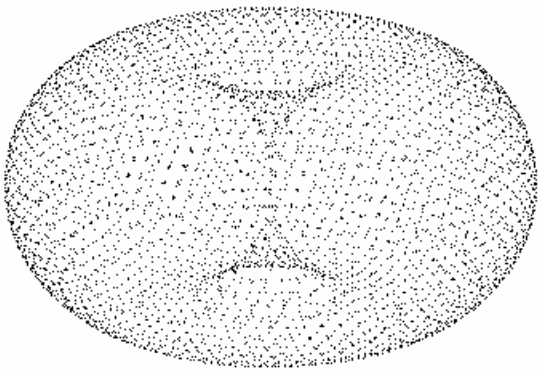
```
                                2    2    2 2    2    2
VECTOR(ContourDots_XY((x  + y  + z )  - x  - y , 1, -1, 1, -1, 1, 0.05, 0.05), 1, -0.5, 0.5, 0.1)
```

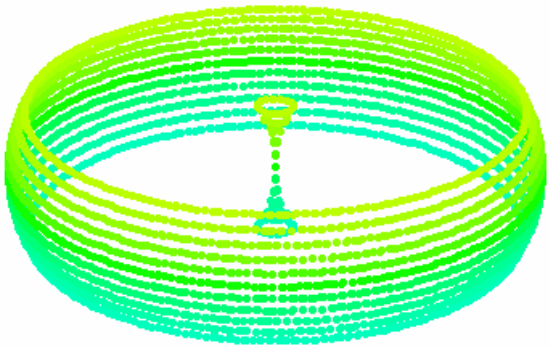




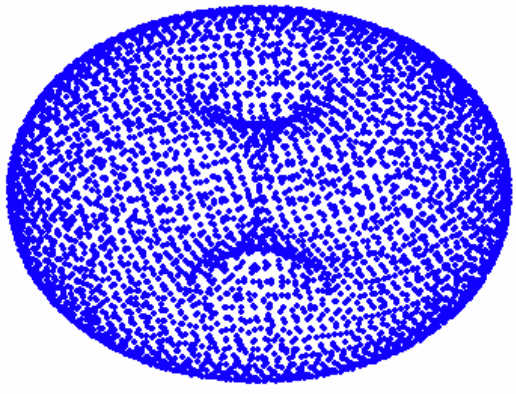
```
                    2    2    2 2    2    2
ImplicitDots((x  + y  + z )  = x  + y , [-1, -1, -0.5], [1, 1, 0.5], 0.05)
```

```
                                2    2    2 2    2    2
VECTOR(ContourPts_XY((x  + y  + z )  - x  - y , 1, -1, 1, -1, 1, 0.025, 0.025), 1, -0.25, 0.25, 0.05)
```

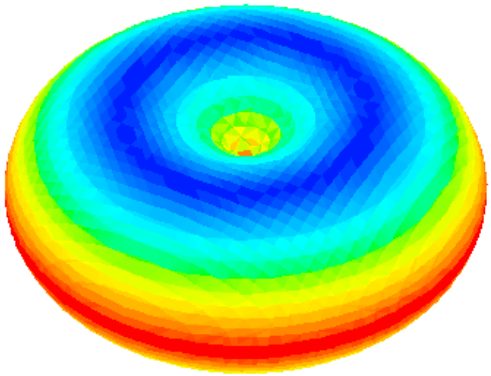




```
                    2    2    2 2    2    2
ImplicitPts((x  + y  + z )  = x  + y , [-1, -1, -0.5], [1, 1, 0.5], 0.05)
```



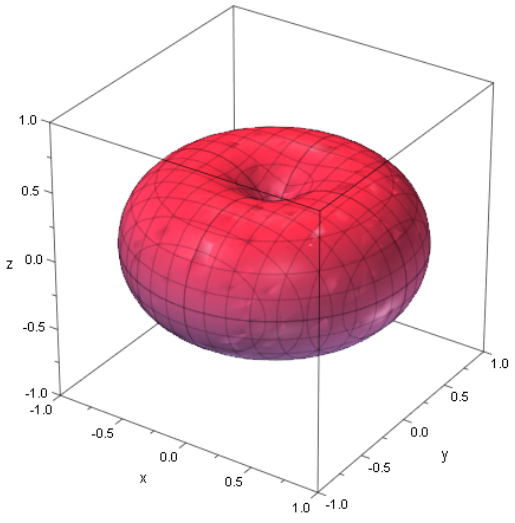
Surface #2 in DPGraph



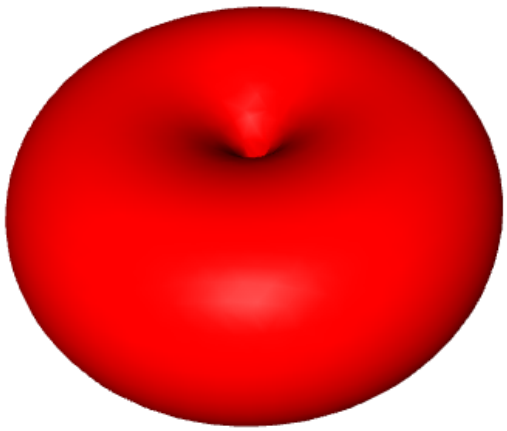
Surface #2 in MuPAD

FAZ - Fläche #2

```
plot(plot::Implicit3d((x^2+y^2+z^2)^2=x^2+y^2,
                      x=-1..1,y=-1..1,z=-1..1,
                      Mesh = [21,21,21],AdaptiveMesh = 3),
                      Scaling = Constrained)
```



Surface #2 in Autograph 3.20



The Mathematics Education into the 21st Century Project

together with

The University of Applied Sciences (FH), Dresden (Germany)

are proud to announce our

10th (Anniversary!) International Conference

"Models in Developing Mathematics Education"

September 11 – 17, 2009
Dresden, Saxony, Germany

in cooperation with

Saxony Ministry of Education

Chairman
Dr. Alan Rogerson, International Coordinator of the Mathematics in Society Project (Poland).
Prof Dr Fayez Mina, Dept. of Curriculum & Instruction, Faculty of Education, Ain Shams University (Egypt).

You are invited to attend our project conference to be held in the historic and beautyful city of Dresden, Germany.
The chairman of the Local Organising Committee will be Prof. Dr. Ludwig Paditz of the Dresden University of Applied Sciences.
For ALL further conference details and updates please email arogerson@inetia.pl .

Purpose of the Conference

In 1992 ACDCA (the Austrian Center for Didactics of Computer Algebra) started a conference series which has become a driving force in bringing technology, in particular computer algebra systems (CAS), into the classroom.

The conference series comprises two strands: the ACDCA Summer Academies, which are more oriented towards didactical questions connected with the use of technology for teaching and learning and the International Derive Conferences, which are more oriented towards concrete applications of Derive and similar tools for teaching in schools and at universities.

The conference program will consist of keynote lectures, contributed presentations, and workshops. Other formats will be considered if suggested by participants.

Conference Co-Chairs

Steve Joubert and Josef Boehm

Organizing Committee

Steve Joubert
Tshwane University of Technology
South Africa
JoubertSV@tut.ac.za

Program Committee Co-Chairs

ACDCA Summer Academy:
Bernhard Kutzler & Vlasta Kokol-Voljc

DERIVE & TI-Nspire Conference:
Michel Beaudin & Josef Boehm

Website

To be announced shortly.

Call for Submissions

Deadlines will be announced shortly

10th ACDCA Summer Academy

Topics include:

- CAS-based curricula and teaching methods
- Dynamic geometry systems as teaching tools
- Assessing with technology
- Internet as a teaching aid
- Dispensable and indispensable mathematical skills and abilities
- CAS as pedagogical tools for compensation and amplification
- New classroom examples using CAS
- The next generation: PeCAS = Pedagogical CAS

8th Derive & TI-Nspire Conference

Topics include:

- Applications in mathematics, the natural sciences, research & development, economy, social sciences, industry, ...
- Teaching and learning with Derive, TI-Nspire, TI-89/92, Voyage200
- Programming in Derive, in TI-Nspire, TI-89/92, Voyage200
- Interfaces to other programs/tools
- Producing and using utility files
- Problems and limitations
- Web resources – going online
- Connecting computer algebra and computational logic

Please mark this event on your calendar and submit using the respective form at the conference website that will be announced shortly.

TIME-2008

Int'l Symposium on

Technology and its Integration into Mathematics Education

This symposium combines the

10th ACDCA Summer Academy

and the

8th Int'l Derive&TI-Nspire Conference

and is organized by

Tshwane University of Technology

22-26 Sept 2008

Venue: ATKV Buffelspoort
(1 hour from Pilansberg Game Reserve and Sun City)
South Africa

1st Announcement