THE DERIVE - NEWSLETTER #37

THE BULLETIN OF THE

DERIVE

USER GROUP

+ TI 92

Contents:

Revised version 2017 **March 2000**

[1] **Exploring Calculus and Differential Equations** with the TI-89 and TI-92+(+ Diskette), M B Schneider & L G Gilligan, Gilmar Publishing, ISBN 2-888808-05-5, http://www.gilmarpublishing.com , (very helpful with a lot of step by step guided problems and many additional tasks)

[2] **Prüfungsaufgaben für das Arbeiten mit DERIVE und dem TI-92/89, Bd 1,** V. Kokol-Voljc, bk-teachware SR-14, ISBN 3-901769-23-4

[3] **Mathe-Trainer I** für TI-89/92/92+(+ Diskette), Josef Böhm, bk teachware SR-15, ISBN 3-901769-24-2 (Interaktive Übungsprogramme zum Rechnen mit Termen, zum Arbeiten mit linearen Funktionen und zum Differenzieren , bzw. Integrieren)

[3] **Hand-Held Technology in Mathematics and Science Education,** Ed Laughbaum, editor, contact: http://www.math.ohio-state.edu/~laughba/

[4] **Les fonctions au lycée avec TI-89,** Jean-Alain Rodier, Èditions Pole 1999, ISBN 2-909737-43-8

[5] **Stastiques**, Daniel Vagost, a publication of T^3-France (Statistics on the TI-83+)

[6] **Modern Computer Algebra**, Joachim von zur Gathen & Jürgen Gerhard, Cambridge University Press 1999, ISBN 0 521 64176 4. (This is another "bible" for CAS algorithms like Modular algorithms and Interpolation, Fast polynomial evaluation, Primality testing, Factoring integers, Public key kryptography, Gröbner bases a.o.)

Interesting WEB sites

clem.mscd.edu/~talmanl/MathAnim.html	QuickTime animations from trig – complex analyis
www.davidparker.com/index.html	Update your DPGraph 2000 (see page 7)
www.emis.ams.org/projects/EULER/	EULER project homepage - a huge library
www.maa.org/editorial/knot/knot-index.html	Cut the Knot - Problems and Puzzles, very nice!
www.math.psu.edu/dna/complex-j.html	Graphics for Complex Analysis
www.maa.org/news/columns.html	Mathematical Association of America´s Online columns (containing Ivar Peterson´s Math Trek, Devlin´s Angle, Math Chat, ...)
www.mathgoodies.com	Interactive Math Lessons with a Problem-Solving Approach
www.math-atlas.org/	A Gateway to Mathematics
www.maths.usyd.edu.au:8000/MathSearch.html	An enormous list of links and materials
www.argonet.co.uk/oundlesch/	Internet Resouirces for the Classroom
www-math.cudenver.edu/w4t/	Download course materials, papers and books for undergraduate courses of the Colorado University of Denver
www.mcs.surrey.ac.uk./Personal/R.Knott/Fibonacci/fib.html	Fibonacci Numbers, the Golden Section and the Golden String

The links printed in red are not valid in 2017, Josef

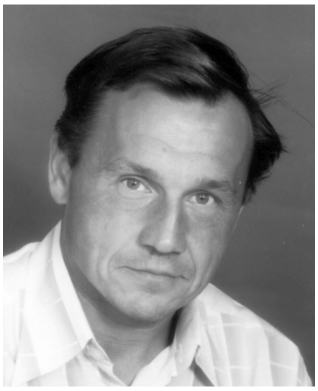
Visit `http://www.acdca.ac.at`

Austrian Center for Didactics of Computer Algebra

many materials, booklist, links, information about T^3 and download DNL-files

Dear all,

I hope that you all had a pleasent switch from the three 9´s to the three 0´s in the calendar. It is also a turn off of a 9 and a turn on of a 0 in the number of DUG-years. Our first decade has to be fulfilled, a remarkable space of time for a non-profit organisation brought together and kept alive by its enthusiastic and cooperating members. Many thanks to you all. In this year 2000 we are waiting for a new *DERIVE* and believe it or not - read page 37 - it could be that at the moment when you have this DNL in hands it might have been released.



Johann Wiesenbauer who was in Hawaii improving several routines gives insight into *DERIVE* 5´s programming facilities and Josef Lechner shows how to use logic operators to shade areas. I intended to include a fine piece of programming the TI, too, but as ever, I run short of space. So Thomas Himmelbauer, one of our best T^3-instructors will present an example in the next DNL.

Many articles in this issue demonstrate - once more - that in most cases it is not too difficult to transfer an idea presented in *DERIVE* to the TI-environment and vice versa. You can find another "experiment" from Carl and Marvin´s laboratoy which proved to be extremely successful not only in their classes. Hubert Voigt and David Bowers present different examples of the use of a CAS, while Ernö Scheiber - a warm welcome to our first author from Romania - and F. M. Fernández contribute for a more advanced level dealing with the never ending appeal of differential equations. G P Speck from New Zealand introduces an interesting *DERIVE* - statement.

At last I´d like to draw your attentiont at the two *DERIVE* and *TI*-Conferences presented below and remind you that you can download all the files used and presented in this DNL from two web sites (many thanks to W. Wegscheider - ACDCA - and B. Kaineder from bk-teachware).

http://www.acdca.ac.at/t3/dergroup/index.htm (not valid 2017)
http://www.bk-teachware.com/main.asp?session=375059 (not valid 2017)

Have you already renewed your DUG-membership for 2000?

Don´t forget to inform:

Exam Questions and Basic Skills in Technology Supported Maths Teaching
July 2nd - 5th 2000
Portoroz, Slovenia
http://www.kutzler.com/acdca-00/default.html

Computer Algebra in Mathematics Education

The Fourth International DERIVE- TI89/92 Conference
July 12th - 15th 2000
Liverpool John Moores University, Liverpool UK
http://www.cms.livjm.ac.uk/derive2k

The *DERIVE-NEWSLETTER* is the Bulletin of the *DERIVE & TI-92 User Group*. It is published at least four times a year with a contents of 44 pages minimum. The goals of the *DNL* are to enable the exchange of experiences made with *DERIVE* and the *TI-92/89* as well as to create a group to discuss the possibilities of new methodical and didactical manners in teaching mathematics.

As many of the *DERIVE* Users are also using the *TI-92/89* the *DNL* tries to combine the applications of these modern technologies.

Editor: Mag. Josef Böhm
A-3042 Würmla
D´Lust 1
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 & TI-92 Newsletter* will be.

Next issue: June 2000
Deadline 15 April 2000

Preview: Contributions for the next issues

Inverse Functions, Simultaneous Equations, Speck, NZL
Akima-Interpolation, Geruschkat, GER
A Utility file for complex dynamic systems, Lechner, AUT
Examples for Statistics, Roeloffs, NL
Quaternion Algebra, Sirota, RUS
Various Training Programs for the TI
A critical comment on the "Delayed Assignation" :==, Kümmel, GER
Sand Dunes, River Meander and Elastica, The lighter Side ....., Halprin, AUS
Type checking, Finite continued fractions, ..., Welke, GER
Kaprekar´s "Self numbers", Schorn, GER
Some simulations of Random Experiments, Böhm, AUT
Examples for Programming with DERIVE 5, Etchells, Lechner ao
Comparing statistics tools: a pie chart with DERIVE, a stem & leaf diagram on the TI,

and
Setif, FRA; Vermeylen, BEL; Leinbach, USA; Aue, GER; Koller, AUT, ...

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

Glyn D Williams, Mawnog Fach, UK

Dear sir,

I was very interested in the article on partial derivatives of trigonometrical functions which appeared in the *Sep 99* issue of the *DERIVE* bulletin, and I enclose some printouts. These were not taken with *DERIVE*, but with another symbolic algebra program called *MathView*. This provides most of the operations provided in *DERIVE*, but unlike *DERIVE* cannot work with arbitrary precision. In particular it cannot handle summation series with an infinite upper bound, and so I have set an arbitrary limit of 100 iterations. SUN(x) is very like the traditional SIN(x), but with a lower upper limit. CUS(x) settles down like COS(x), but shows an uncharacteristic value of $+\infty$ at $x = 0$. TUN(x) is not like TAN(x) at all, but is more like SIN(x) or COS(x); it never reaches infinity, but simply oscillates. However, CUT(x) is more or less similar in shape to the orthodox COT(x). The reciprocal ratios are more or less like their traditional equivalents. The hyperbolic ratios are more or less like the orthodox hyperbolic ones, except that TUNH(x) shows a slight overshoot *above* 1 before settling down at the expected 1 for high x. None of the formulae for addition of ratios work with these modified ones, nor do the forms $\text{CUS}^2x + \text{SUN}^2x = 1$ or the other sums of squares. In order to compute these ratios I started off with (see below)

Declarations

$$\text{maxiter} = 100$$

$$\text{sun}(x) = \sum_{r=0}^{\text{maxiter}} \left([-1]^r \frac{x^{2r+0.5}}{[2r+0.5]!} \right)$$

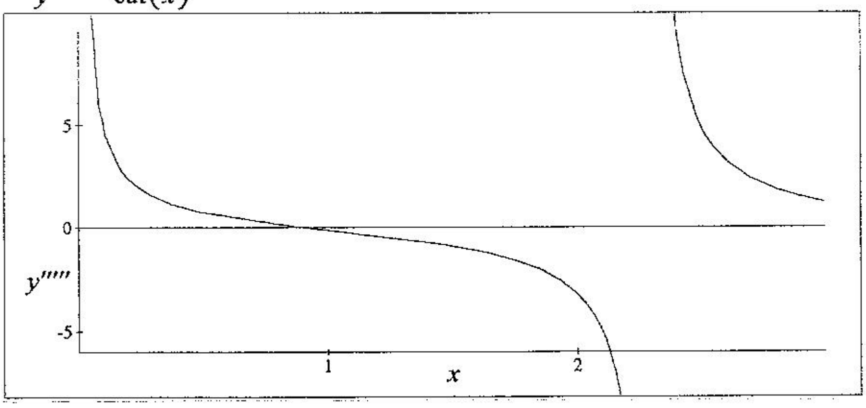
$$\text{cus}(x) \sum_{r=0}^{\text{maxiter}} \left([-1]^r \frac{x^{2r-0.5}}{[2r-0.5]!} \right)$$

(these have been copied from *MathView*, and do not therefore conform to *DERIVE* notation), and derived the other basic ratios using the relationships between the basic ratios, which (except for those involving squares) *do* work with these modified ones. For the hyperbolics I again started off with SUNH(x) and CUSH(x), which are like SUN(x) and CUS(x), but without the $[-1]^r$ term, and derived the other four basic ratios from them. (Both *DERIVE* and *MathView* interpret fractional factorials, as here, as being equal to $\Gamma(x+1)$. I cannot see any *use* for these partial derivatives, except as an exercise in algebra and nu-meric manipulation.

Yours faithfully, Glyn D Williams

$$\text{cut}(x) = \frac{1}{\text{tun}(x)}$$

$$y'''' = \text{cut}(x)$$



Luis Madureira, Portugal lmd@esoterica.pt

Can someone tell me how to solve this equation `MOD(29x,660)= 1` with `(0<x<660)`?
Thanks in advance.

```
Math Poetry: http://www.terravista.pt/Bilene/1496/index.html
Web Poll: http://homepage.esoterica.pt/~madureir/
DERIVE: http://members.xoom.com/lumadureira/
```

(all links are not valid 2017)

Ignacio Larrosa Cañestro, Spain ilarrosa@lander.es

In the 'Archive' menu, choose 'Read' ==> 'Utility' ==> `Number.mth`

Then `SOLVE_MOD(29x = 1, x, 660) = [569]` (in Exact Simplification)

Un saludo

In DERIVE 6 it is not nevessary to load the utility file. It is interesting that the equation is not solved in the first form (#1), but it is solved in the second form (#2).

#1: SOLVE_MOD(29·x = 1, x, 660) = []

#2: SOLVE_MOD(29·x − 1, x, 660) = [569]

#3: $\frac{29\cdot 569 - 1}{660} = 25$

#4: SOLVE_MOD(3·x − 30, x, 9) = [1, 4, 7]

#5: SOLVE_MOD(3·x = 30, x, 9) = [1]

#6: SOLVE_MOD(3·x − 30 = 0, x, 9) = [1]

#7: $\text{VECTOR}\left(\frac{3\cdot x - 30}{9}, x, [1, 4, 7]\right) = [-3, -2, -1]$

The second example is taken from the DERIVE Online Help. You can recognize different treatment of the equation depending on its input form. Josef

Joachim Linn, Kaierslautern, Germany linn@itwm.uni-kl.de

I would like to do some symbolic calculations involving real 6x6-matrices. In the intermediate and final results of these calculations, I would like to see the matrix elements in the general notation "`a_ij`" (I mean "a SUB [i,j]). Using the VECTOR function, I have to specify some expression involving the indices [i,j] or some term or variable independent of one or both indices (in which case I get constant symbolic entries).

Is it possible to define a function DEF_MATRIX(A, N, M) which defines A as a nonscalar object and allocates it as a symbolic N x M matrix ?

Best regards, Joachim Linn

Johann Wiesenbauer, Vienna, AUT J.Wiesenbauer@tuwien.ac.at

Applying the function

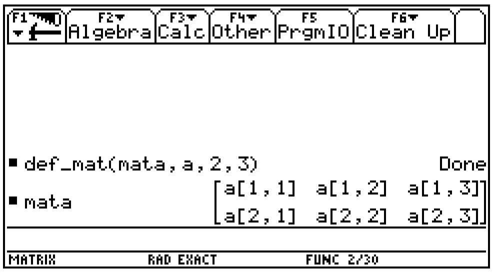
```
DEF_MATRIX(a, n, m) := VECTOR(VECTOR(a    , j, 1, m), i, 1, n)
                                       i,j
```

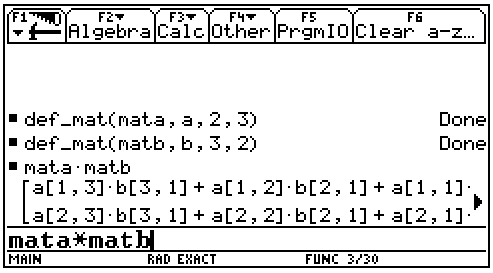
with e.g. n = 2, m = 3 yields

$$\text{DEF\_MATRIX}(b,\ 2,\ 3) = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \end{bmatrix}$$

Is this what you want?
Cheers, Johann

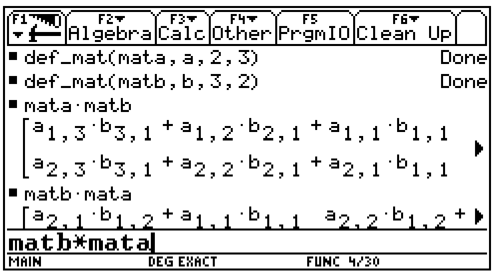
DNL: *It might happen that some TI-Users would like to have a similar function for their TI to demonstrate general rules for matrix calculations. For a similar occasion I produced a small program*:

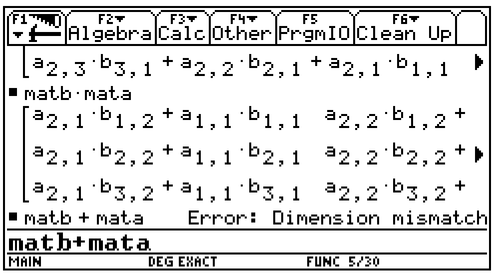


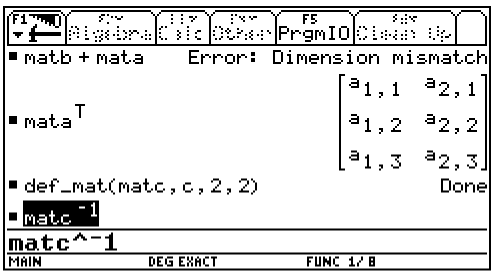


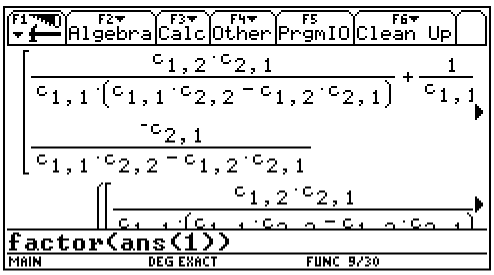

```
def_mat(mnam,el,ro_,co_)
Prgm
string(mnam)→mnam
Local i,j,aux
newMat(ro_,co_)→aux
For i,1,ro_
   For j,1,co_
      el[i,j] →aux[i,j]
   EndFor
EndFor
aux→#mnam
EndPrgm
```

It is not surprising that the Voyage 200 accepts the TI-92 code without any change. But it is very nice and comfortable as well to have subscripts for describing the matrix elements (like in DERIVE).



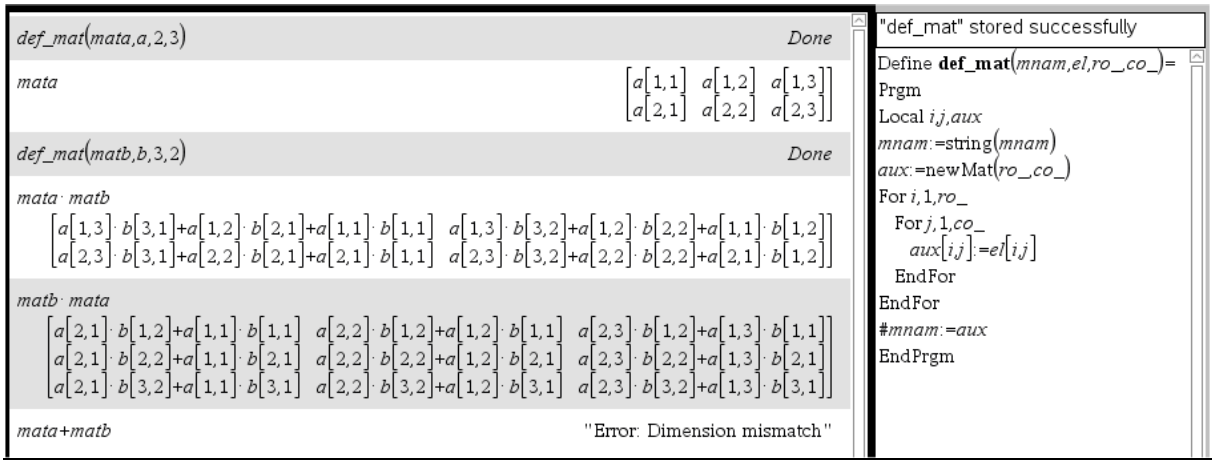






Representation of the first element in row 1 of the inverse matrix looks quite strange. Applying the factor command gives the result as expected.

We can take the same code for TI-NspireCAS, but we must miss the subscripts. At least I could not do it (even using the subscript-template.



Berthold Heinrich

Can you help me?

1. if you approximate π, you get the wrong number 3.14159292035
2. If you try to solve the square-root-equation $\sqrt{5x-56} = \sqrt{x+12} - \frac{10}{\sqrt{x+12}}$ with x := Real (0, -)

   you get the wrong solution x=-13 ([x = 13, x = -13])

Dave Stegenga, Honolulu, Hawaii

rhinodave@hawaii.rr.com

I got that value for pi when I used the default precision 6 (PrecisionDigits:=6) but increased the output digits to 12. (NotationDigits:=12).

The correct procedure is to use the Declare/Algebra State/Simpflication menu and set Digits to 12. This increases the precision digits and the output digits together which gives the desired result. Aloha, David

Johann Wiesenbauer, Vienna, Austria

J.Wiesenbauer@tuwien.ac.at

The first question has already been answered by David. The only thing I want to supplement is that the assign-ment

```
PrecisionDigits := 12
```

alone will suffice, because the variable `NotationDigits` gets the same value automatically.

As for your second question I would like to point out that (like it or lump it!) declaring the domain of variables will not affect subsequent SOLVE commands involving those variables (at least not in the way you thought!) The declaration of a variable serves a different purpose. For example, DERIVE will only simplify the expression sin(n π) to 0, if you declared n as an integer before (which makes sense, doesn't it?)

On the other hand, it's easy to get a vector of all positive solutions of your equation by simplifying

$$\text{SELECT}\left(x > 0,\ x,\ \text{SOLUTIONS}\left(\sqrt{(5\cdot x - 56)} = \sqrt{(x + 12)} - \frac{10}{\sqrt{(x + 12)}},\ x\right)\right) = [13]$$

There may be other cases though, where things are not that simple.

Cheers, Johann

Andrzej Brozi, Poland andrbrozi@ck-sg.p.lodz.pl

I´m trying to use the function RK from ODE_APPR.MTH (the Runge-Kutta-method). *DERIVE* accepts the authored expression (eg. `RK(x,[x,y],[0,0],.01,5)`, which should solve equation y´ = x). Unfortunately instead of simplifying or approximating the expression *DfW* 4.11 seems to do nothing - I´ve let run for something like 20 minutes on Celeron 333 with no result.
Does the function at all? Or is there some method to make it work? I´d appreciate some advise.
Andrzej

Johann Wiesenbauer, Vienna, AUT

Try

`RK([x],[x,y],[0,0],0.01,5)`

instead, though in this special case functions from ODE1.MTH might be in order.
Cheers, Johann

Guiseppe Ornaghi, Italy md5628@mclink.it

Can I expand a piecewise function with the function `FOURIER(y,t,t1,t2,n)`?

Boz Kempski, England B.Kempski@anglia.ac.uk

Of course, you can obtain a Fourier series representation for a periodic, piecewise defined function. I have done it many times. For example say u(t) = -3 for -3 < t < 0 and 2t for 0 < t < 5 and that u(t+8) = u(t) for all t.
Use the CHI-function to define u(t) over the periodic interval (-3,5) namely

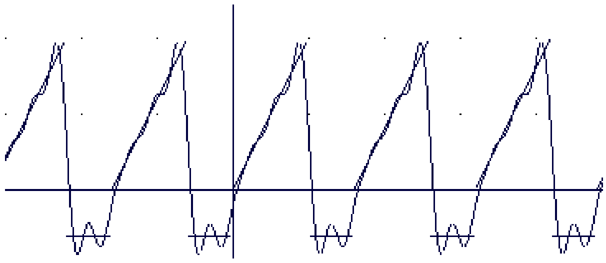
`u(t):=CHI(-3,t,0)*(-3) + CHI(0,t,5)*2t`

then issue the Fourier command i.e. `Fourier(u(t),t,-3,5,5)`. Then simplify or approximate and then plot to obtain the desired representation.
I also have a `periodic_plot` command to plot a selected number of cycles of any piecewise defined function. It is quite nice to see how the Fourier series representation compares with the original wave form.

***DNL*:** Jan Verhoosel wrote n an earlier DNL Jan Verhoosel wrote about *DERIVE and Plotting T-periodic Functions* (DNL#16, 1994). He used a function `PERIODIC(U(t), t, -3, 5.` His function together with Boz´s one gave the nice plot (see next page). Josef

We can use two ways to define a piecewise defined function with DERIVE 6. I added the TI-NspireCAS answer defining the *periodic*-function and using Michel Beaudin's and Frédérick Henri's great libraries for TI-NspireCAS: Kit_ETS_MB.tns and Kit_ETS_FH.tns.

```
#1:   u(t) := χ(-3, t, 0)·(-3) + χ(0, t, 5)·2·t

#2:   FOURIER(u(t), t, -3, 5, 5)

      v(t) :=
        If -3 < t ≤ 0
           -3
#3:        If 0 < t < 5
              2·t
              ?

#4:   FOURIER(v(t), t, -3, 5, 5)

      PERIODIC(f, t, a, b) :=            lim             f
#5:                             t→MOD(t - a, b - a) + a

#6:   PERIODIC(u(t), t, -3, 5)

#7:   PERIODIC(v(t), t, -3, 5)
```



$periodic(f,t,a,b):=\lim\limits_{t\to mod(t-a,b-a)+a}(f)$ *Done*

$u:=\begin{cases}-3, & -3<x\le 0\\ 2\cdot x, & 0<x<5\end{cases}$ $\begin{cases}-3, & -3<x\le 0\\ 2\cdot x, & 0<x<5\end{cases}$

$f1(x):=u$ *Done*

$periodic(u,x,-3,5)$ $\lim\limits_{x\to mod(x+3,8)-3}\left(\begin{cases}-3, & -3<x\le 0\\ 2\cdot x, & 0<x<5\end{cases}\right)$

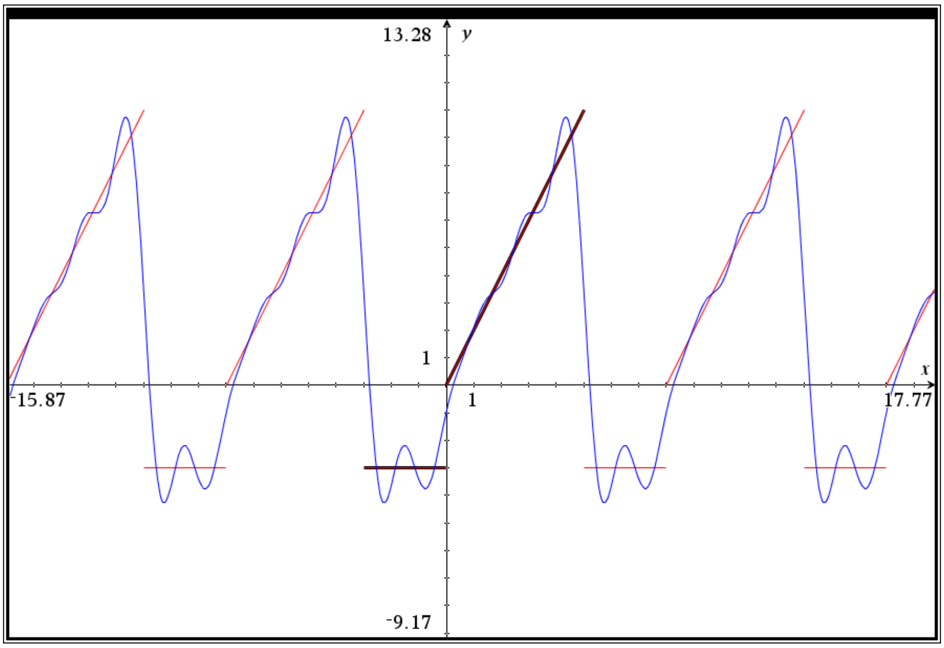
$f2(x):=\lim\limits_{x\to mod(x+3,8)-3}\left(\begin{cases}-3, & -3<x\le 0\\ 2\cdot x, & 0<x<5\end{cases}\right)$ *Done*

kit_ets_mb\fourier(u,x,-3,5,5)

$\left(\frac{13\cdot\sqrt{2}}{10\cdot\pi}+\frac{\frac{4\cdot\sqrt{2}}{25}-\frac{8}{25}}{\pi^2}\right)\cdot\cos\left(\frac{5\cdot\pi\cdot x}{4}\right)+\left(\frac{\frac{3}{5}-\frac{13\cdot\sqrt{2}}{10}}{\pi}+\frac{4\cdot\sqrt{2}}{25\cdot\pi^2}\right)\cdot\sin\left(\frac{5\cdot\pi\cdot x}{4}\right)+\left(\frac{\frac{4\cdot\sqrt{2}}{9}-\frac{8}{9}}{\pi^2}-\frac{13\cdot\sqrt{2}}{6\cdot\pi}\right)\cdot\cos\left(\frac{3\cdot\pi\cdot x}{4}\right)$ ▸

◂ $\sin\left(\frac{\pi\cdot x}{2}\right)+\left(\frac{-4\cdot\sqrt{2}-8}{\pi^2}-\frac{13\cdot\sqrt{2}}{2\cdot\pi}\right)\cdot\cos\left(\frac{\pi\cdot x}{4}\right)+\left(\frac{\frac{13\cdot\sqrt{2}}{2}+3}{\pi}-\frac{4\cdot\sqrt{2}}{\pi^2}\right)\cdot\sin\left(\frac{\pi\cdot x}{4}\right)-\frac{\cos(\pi\cdot x)}{\pi^2}+\frac{4\cdot\sin(\pi\cdot x)}{\pi}+2\to f3(x)$

Done



Reinhard Schaeck, Germany schaeck@drei-eins-vier.de

I want to solve the equation

$$\mathrm{COS}(2x) - 2\,\mathrm{COS}(x) = 0$$

algebraically and don´t now how. "SOLVE" doesn´t do anything. I´d like to underline that I want *DERIVE* to solve the equation - not myself. Which *DERIVE* command is able to solve the equation?

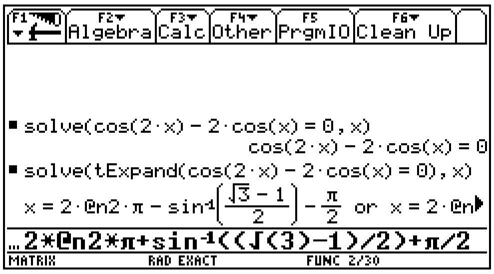
Johann Wiesenbauer, Vienna, AUT

`SOLVE` doesn´t do anything? Try again after inserting the line

```
Trigonometry:=Expand
```

Now, `SOLVE` even does a little too much for my taste. Anyway, by taking the first two solutions and adding 2kπ manually, where k is an integer, you will get all real solutions in *DfW* 4.11

$$\left[x = \mathrm{ATAN}\left(\frac{108^{1/4}\cdot(\sqrt{3}-1)}{6}\right) + \frac{\pi}{2},\ x = -\mathrm{ATAN}\left(\frac{108^{1/4}\cdot(\sqrt{3}-1)}{6}\right) - \frac{\pi}{2},\right.$$



DNL: I tried on the TI-92. Here also texpand() is necessary. @2 stands for any integer. Josef

See below how DERIVE 6 and TI-NspireCAS are behaving. It might be a nice task for students to compare the different output forms of the solutions. Josef

```
Trigonometry :≔ Expand
SOLUTIONS(COS(2·x) − 2·COS(x) = 0, x, Real)
```

$$\left[\mathrm{ACOT}\left(\frac{\sqrt{2}\cdot 3^{3/4}}{6} - \frac{12^{1/4}}{2}\right),\ -\mathrm{ACOT}\left(\frac{\sqrt{2}\cdot 3^{3/4}}{6} - \frac{12^{1/4}}{2}\right),\ \frac{3\cdot\pi}{2}\ -\right.$$

solve(cos(2· x)−2· cos(x)=0,x)	$x=2\cdot n2\cdot\pi-\sin^{-1}\left(\frac{\sqrt{3}-1}{2}\right)-\frac{\pi}{2}$ or $x=2\cdot n2\cdot\pi+\sin^{-1}\left(\frac{\sqrt{3}-1}{2}\right)+\frac{\pi}{2}$

Jean-Jaques Dahan, FRA jjdahan@wanadoo.fr

Dear friend,
I wish you first a happy new year to you and your family and I offer you the address of the website where you can discover the Conference we have given with Mathilde Arragon (she is a great specialist of Cabri in physical modelization) in Sao Paulo last October during Cabriworld 1. It is an original dialog between a Mathematician and a Physician where I use *DERIVE* and the TI-92.

www.ac-grenoble.fr/phychim/confjjma/index.htm

I will appreciate to have your opinion about our work and also about the way we have translated it in HTML pages. Thanks Jean-Jaques DAHAN

DNL: This site is a must. I strongly recommend to visit Jean-Jaques and Mathilde. (not valid 2017)

From Vladimir Rovenski (rovenski@hotmail.com):

You can see the web resorces for my book

"Geometry of Curves and Surfaces with Maple"

http://math.haifa.ac.il/rovenski-compgeom.html

The website is not valid 2017. But I found another website, where you can download a complete book in pdf-format: Simply “google” for

Modeling of Curves and Surfaces with MATLAB - ResearchGate

Josef

How to delete TI-92/89 Plots from within a program?

I wrote to Michelle Miller-Thornton:

I need some idea for my TI. Is it possible to delete a plot - created by a datasheet - from within a program. I would not like only to turn it off by using PlotsOff, but to erase it completely. Many thanks, Josef

Michelle Miller mmiller@ti.com

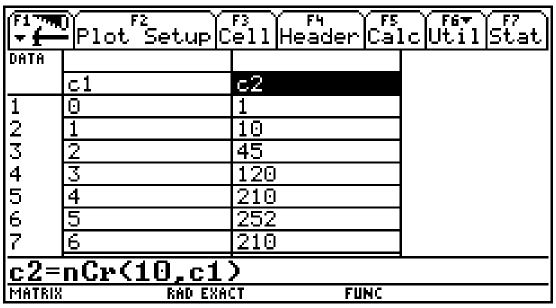
I thought about your question and asked around and this is all I can offer so far....
You cannot delete a single plot or all plots from a program command. It also appears that you are not satisfied with turning off a plot or using NewPlot to overwrite a plot. So, the only suggestion I have is to use Graph Data Bases. You can use the StoGDB command to save off your current plots, functions and window settings, and then recall these settings later (after you´ve changed them). Also, using this capability, you could create an "empty GDB" that would have to accompany your program. The one catch with the empty GDB is that it will also clear your functions, not just the plots, since the GDB reinstates functions, plots and window settings.
Hope this helps
Michelle

Hubert Voigt, Perg, Austria voigthubert@hotmail.com

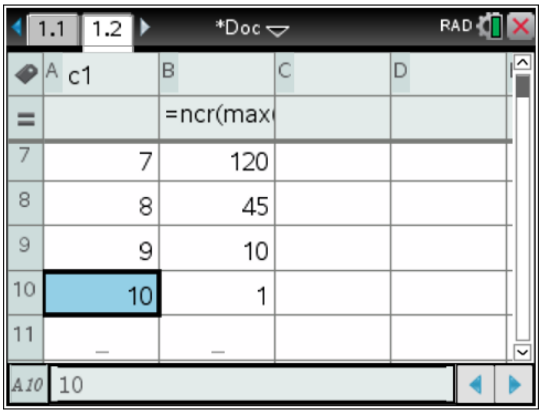
.... by the way, I created together with my students a probability table of the binomial distribution in the Data/Matrix-Editor and had to apply a really adventurous trick to bring the values of $\binom{n}{k}$ into column c2 - with numbers 0 through 10 in c1. I had to enter c2 = ncr(c1*max(c1)/c1,c1). I don´t have any idea, why it is not able to use the saved value for n = 10. It also not able to work with c=ncr(10,c1). Do you have any advice to overcome this bug?

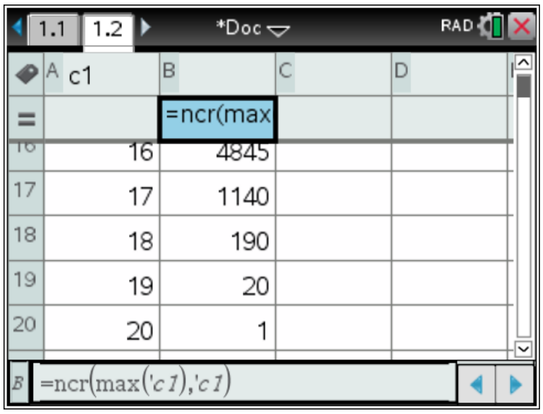
DNL: No, I don´t have. All what I can contribute is that it works on the TI-92PLUS and on the TI-89.

See the screen shot, Josef.

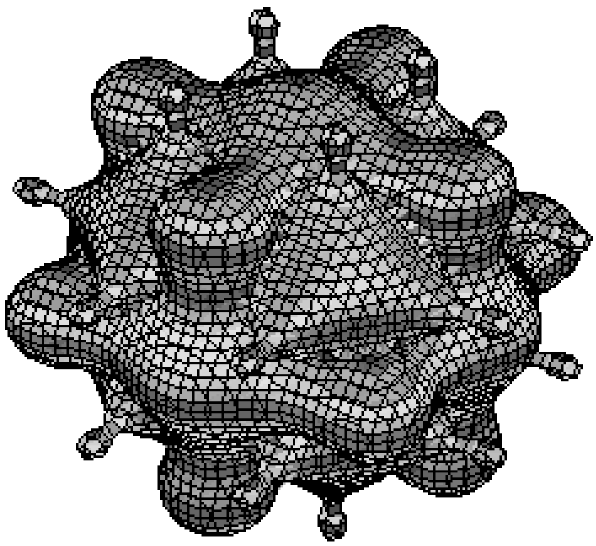


This is one possible way to do it on a TI-NspireCAS:





For working with the TIs from TI-83 through TI-89 I can recommend the Virtual TI, which can be downloaded from TI´s website. But be careful, I tried to transfer a ROM-dump from the TI-92PLUS - the real PLUS and not the PLUS-module - to the PC using the *VTI*-Wizard and had a fatal address error on my TI-92PLUS, followed by a reset which erased all my unarchived files. Make sure that you have saved all your files before trying *VTI* cooperating with the new PLUS. Fortunately did so. It might be that there is an update of *VTI* which works properly.



Intrinsically speaking, my space-time curve had a finite arclength, yet allowed me to fly off to infinity at a tangent angle.

William Whewell (1794 - 1866) England

David Halprin

ON THE SYMBOLIC COMPUTATION OF e^{At} MATRIX IN DERIVE

E. SCHEIBER*

The purpose of this note is to present a practical approach to compute symbolically the matrix e^{At}, where A is a n order square matrix with real elements and t is a real number.

The presented method may be easily implemented in *Derive* [1] and it is applicable for a matrix whose dimension is not too large (depending of the used software).

All used formulas are well known.

It's known that the columns of the matrix e^{At} are liniar independent solutions of the liniar differential system $\dot{x}(t) = Ax(t)$, whose general solution is $x(t) = e^{At}c, \quad c \in R^n$.

The matrix e^{At} is defined by

$$e^{At} = \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k = \frac{1}{2\pi i} \int_\gamma (\lambda I - A)^{-1} e^{\lambda t} d\lambda \qquad (1)$$

where γ is a smooth, closed curve deliminating a domain of the complex plane, that contains all the eigenvalues of the matrix A. The integral representation – named Dunford - Taylor integral – will be used to compute the matrix e^{At}. This integral representation may be interpreted as the inverse Laplace transform of the matrix $(\lambda I - A)^{-1}$, too.

For $\lambda \in \gamma$, the matrix $\lambda I - A$ is invertible and because $\frac{\|A\|}{|\lambda|} < 1$ we have

$$(\lambda I - A)^{-1} = \frac{1}{\lambda}(I - \frac{1}{\lambda}A)^{-1} = \frac{1}{\lambda} \sum_{k=0}^{\infty} \frac{A^k}{\lambda^k} = \sum_{k=0}^{\infty} \frac{A^k}{\lambda^{k+1}}$$

and consequently

$$\frac{1}{2\pi i} \int_\gamma (\lambda I - A)^{-1} e^{\lambda t} = \sum_{k=0}^{\infty} \frac{A^k}{k!} \left(\frac{k!}{2\pi i} \int_\gamma \frac{e^{\lambda t}}{\lambda^{k+1}} d\lambda \right) =$$

* Department of Informatics, *Transilvania* Univerity of Brasov, Romania

[1] In the core of *Derive* there isn't such a function, while *Maple* and *Mathematica* provide such kind of function.

$$= \sum_{k=0}^{\infty} \frac{A^k}{k!} \left(e^{\lambda t}\right)^{(k)}_{\lambda=0} = \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k,$$

proving (1).

If $f(\lambda)$ is the characteristic polynomial of the matrix A

$$f(\lambda) = \det(\lambda I - A) = \prod_{j=1}^{m} (\lambda - \lambda_j)^{r_j},$$

$(r_1 + r_2 + \ldots + r_m = n)$ then $(\lambda I - A)^{-1} = \frac{B(\lambda)}{f(\lambda)}$, where $B(\lambda)$ is the corresponding adjoint matrix.

Using the residue theorem, from (1), it follows that

$$e^{At} = \frac{1}{2\pi i} \int_{\gamma} \frac{B(\lambda)}{f(\lambda)} e^{\lambda t} d\lambda = \sum_{j=1}^{m} \operatorname{Res}\left(\frac{B(\lambda)}{f(\lambda)} e^{\lambda t}, \lambda_j\right) = \tag{2}$$

$$= \sum_{j=1}^{m} \frac{1}{(r_j - 1)!} \left((\lambda - \lambda_j)^{r_j} \frac{B(\lambda)}{f(\lambda)} e^{\lambda t}\right)^{(r_j - 1)}_{\lambda = \lambda_j}.$$

In order to use (2) we must compute

1. The distinct eigenvalues $\lambda_j, j = 1, 2, \ldots, m$. This may be programed in *Derive* as [2]

```
#4:  S(a) := VECTOR((EIGENVALUES(a))_k , k, 1, DIM(EIGENVALUES(a)))

#5:  S(a) := EIGENVALUES(a)
```

(#4 is the original function from year 2000. As #5 does pretty the same I replaced #4 by #5.)

2. The multiplicities r_j of λ_j, for any $j = 1, 2, \ldots, m$.

   Denoting the function $r(k, x) = \begin{cases} 1 & \text{if} \quad f^{(k)}(x) = 0 \\ 0 & \text{otherwise} \end{cases}$ for any eigenvalue λ_j, we define the vectors $w_j = (w_j^i)_{1 \le i \le n-1}$ and $W_j = (W_j^i)_{1 \le i \le n-1}$ with

$$w_j^i = \sum_{k=1}^{i} r(k, \lambda_j), \qquad W_j^i = \begin{cases} 1 & \text{if} \quad w_j^i = i \\ 0 & \text{otherwise} \end{cases}.$$

   The multiplicity of λ_j will be

$$r_j = 1 + \sum_{k=1}^{n-1} W_j^i. \tag{3}$$

The respective *Derive* code is:

```
#6:   V(a, j) := VECTOR(IF(lim_{x→(S(a))↓j} (d/dx)^k CHARPOLY(a, x) = 0, 1, 0), k, 1, DIM(a) - 1)

#7:   W(a, j) := VECTOR(Σ_{i=1}^{k} (V(a, j))↓i, k, 1, DIM(a) - 1)

#8:   R(a) := VECTOR(1 + Σ_{k=1}^{DIM(a) - 1} IF((W(a, j))↓k = k, 1, 0), j, 1, DIM(S(a)))
```

(Ernö used the ELEMENT() function in 2000. We can take the SUB-tool instead.)

3. The adjoint matrix $B(\lambda)$. We mention two ways to perform this task.

   - Let be the $n-1$ degree polynomial $C(x,y) = \frac{f(x)-f(y)}{x-y}$. Setting $x = \lambda I, y = A$ in the identity $f(x) - f(y) = (x-y)C(x,y)$ and using the Cayley-Hamilton theorem (f(A)=0) it obtains

$$f(\lambda)I = (\lambda I - A)C(\lambda I, A).$$

Thus, the adjoint matrix is $C(\lambda I, A)$.
The *Derive* code is

```
#9:   C(x, y, a) := (-1)^DIM(a)·(CHARPOLY(a, x) - CHARPOLY(a, y)) / (x - y)

#10:  B(a, x) := C(x·IDENTITY_MATRIX(DIM(a)), a, a)
```

- If $f(\lambda) = \lambda^n + p_1\lambda^{n-1} + \ldots + p_{n-1}\lambda + p_n$ and $B(\lambda) = B_0 + B_1\lambda + \ldots + B_{n-1}\lambda^{n-1}$, where $B_0, B_1, \ldots, B_{n-1}$ are n order square matrix, then from

$$f(\lambda)I = (\lambda I - A)B(\lambda)$$

it results the recurrent relations

$$B_{n-1} = I$$
$$B_{n-1-i} - AB_{n-i} = p_i I, \qquad i = 1, 2, \ldots, n-1,$$

from where

$$B_{n-1-i} = \sum_{j=0}^{i-1} p_j A^{i-j} + p_i I \qquad i = 1, 2, \ldots, n-1,$$

and consequently

$$B(\lambda) = \lambda^{n-1} I + \sum_{i=1}^{n-1} \lambda^{n-1-i} \left(\sum_{j=0}^{i-1} p_j A^{i-j} + p_i I \right).$$

The coefficients of the characteristic polynomial results from

#11:

$$P(a) := \mathrm{VECTOR}\left(\frac{\lim\limits_{x\to 0}\left(\frac{d}{dx}\right)^{\mathrm{DIM}(a)-k}\left((-1)^{\mathrm{DIM}(a)}\cdot \mathrm{CHARPOLY}(a,\ x)\right)}{(\mathrm{DIM}(a)-k)!},\ k,\ 0,\ \mathrm{DIM}(a)\right)$$

#12: `P(a) := VECTOR(POLY_COEFF(CHARPOLY(a, x), x, k), k, POLY_DEGREE(CHARPOLY(a, x), x), 0, -1)`

(#11 is Ernö's function from 2000. We can use POLY_COEFF and POLY_DEGREE together with CHARPOLY now and obtain #12.)

The *Derive* code for the adjoint matrix is

#13:

$$B(a,\ x) := x^{\mathrm{DIM}(a)-1}\cdot \mathrm{IDENTITY\_MATRIX}(\mathrm{DIM}(a)) + \sum_{i=1}^{\mathrm{DIM}(a)-1} x^{\mathrm{DIM}(a)-1-i}\cdot\left(\left(\sum_{j=1}^{i}(p\cdot a)_j\cdot a^{i-j+1}\right) + (p\cdot a)_{i+1}\cdot \mathrm{IDENTITY\_MATRIX}(\mathrm{DIM}(a))\right)$$

Finally, the *Derive* implementation of (2) is

#14:

$$U(a,\ x,\ t) := \frac{(-1)^{\mathrm{DIMENSION}(a)}\cdot B(a,\ x)\cdot e^{x\cdot t}}{\mathrm{CHARPOLY}(a,\ x)}$$

#15:

$$\mathrm{RES}(f,\ a,\ n) := \frac{\lim\limits_{x\to a}\left(\frac{d}{dx}\right)^{n-1}\left((x-a)^n\cdot f\right)}{(n-1)!}$$

#16:

$$\mathrm{EXPM}(a) := \mathrm{VECTOR}\left(\mathrm{VECTOR}\left(\sum_{k=1}^{\mathrm{DIM}(S(a))}\mathrm{RES}\left((U(a,\ x,\ t))_{i,j},\ (S(a))_k,\ (R(a))_k\right),\ j,\ \mathrm{DIM}(a)\right),\ i,\ \mathrm{DIM}(a)\right)$$

The formula of #15 represents the computation of the residue of function *f*, corresponding to the *n* order pole *a*.

Two examples are following.

1. Solve the initial value problem:

$$\dot{x} = -667x + 333y \qquad x(0) = 0$$
$$\dot{y} = 666x - 334y \qquad y(0) = 3$$

Introducing

```
#17:  Example 1:
```

$$\#18: \quad \left[a := \begin{bmatrix} -667 & 333 \\ 666 & -334 \end{bmatrix}, \; c := \begin{bmatrix} 0 \\ 3 \end{bmatrix} \right]$$

and simplifying the expression (applying the 2nd way of computation)

$$\#19: \quad \mathrm{EXPM}(a,\ t)\cdot c = \begin{bmatrix} e^{-t} - e^{-1000\cdot t} \\ 2\cdot e^{-t} + e^{-1000\cdot t} \end{bmatrix}$$

we obain the solution:

$$x = e^{-t} - e^{-1000t}$$
$$y = 2e^{-t} + e^{-1000t}$$

Finally we doublecheck the solution:

$$\#20: \quad \left[\mathrm{XX}(t) := e^{-t} - e^{-1000\cdot t},\ \mathrm{YY}(t) := 2\cdot e^{-t} + e^{-1000\cdot t} \right]$$

$$\#21: \quad \begin{bmatrix} \mathrm{XX}(0) \\ \mathrm{YY}(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$\#22: \quad \begin{bmatrix} \frac{d}{dt}\mathrm{XX}(t) = -667\cdot\mathrm{XX}(t) + 333\cdot\mathrm{YY}(t) \\ \frac{d}{dt}\mathrm{YY}(t) = 666\cdot\mathrm{XX}(t) - 334\cdot\mathrm{YY}(t) \end{bmatrix} = \begin{bmatrix} \mathrm{true} \\ \mathrm{true} \end{bmatrix}$$

2. Solve the following system:

$$\dot{x} = -7x + 3y + 4z - 8u \qquad x(0) = \alpha$$
$$\dot{y} = 6x + 4y + 4z - 8u \qquad y(0) = \beta$$
$$\dot{z} = 6x + 3y + 5z - 8u \qquad z(0) = \gamma$$
$$\dot{u} = 7x - 11y + 22z - 10u \qquad u(0) = \delta$$

Now I'd like to apply Ernö's first method. I recall `B(a,x)` from #10, enter the problem data and try to solve using `EXPM(a,t)` again.

#28: Once more with the first method:

#29: $$\left[aa := \begin{bmatrix} 7 & 3 & 4 & -8 \\ 6 & 4 & 4 & -8 \\ 6 & 3 & 5 & -8 \\ 7 & -11 & 22 & -10 \end{bmatrix},\ cc := \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix}\right]$$

#30: B(a, x) := C(x·IDENTITY_MATRIX(DIM(a)), a, a)

#31: sol := EXPM(aa, t)·cc

#32: $$sol := \begin{bmatrix} e^{2\cdot t}\cdot(4\cdot t\cdot(4\cdot\alpha + 31\cdot\beta - 32\cdot\gamma - 2\cdot\delta) - 10\cdot\alpha - 121\cdot\beta + 132\cdot\gamma) + e^{t}\cdot(11\cdot\alpha + 121\cdot\beta - 132\cdot\gamma) \\ e^{2\cdot t}\cdot(4\cdot t\cdot(4\cdot\alpha + 31\cdot\beta - 32\cdot\gamma - 2\cdot\delta) - 10\cdot\alpha - 121\cdot\beta + 132\cdot\gamma) + e^{t}\cdot(10\cdot\alpha + 122\cdot\beta - 132\cdot\gamma) \\ e^{2\cdot t}\cdot(4\cdot t\cdot(4\cdot\alpha + 31\cdot\beta - 32\cdot\gamma - 2\cdot\delta) - 10\cdot\alpha - 121\cdot\beta + 132\cdot\gamma) + e^{t}\cdot(10\cdot\alpha + 121\cdot\beta - 131\cdot\gamma) \\ e^{2\cdot t}\cdot(6\cdot t\cdot(4\cdot\alpha + 31\cdot\beta - 32\cdot\gamma - 2\cdot\delta) - 17\cdot\alpha - 197\cdot\beta + 214\cdot\gamma + \delta) + e^{t}\cdot(17\cdot\alpha + 197\cdot\beta - 214\cdot\gamma) \end{bmatrix}$$

One can read off the solution in the rows of the matrix: $x(t)$, $y(t)$, $z(t)$ and $u(t)$. We can test the correctness of the solution in #35:

#34: $$\left[xx := sol_{1,1},\ yy := sol_{2,1},\ zz := sol_{3,1},\ uu := sol_{4,1}\right]$$

#35: $$\begin{bmatrix} \frac{d}{dt}\,xx = 7\cdot xx + 3\cdot yy + 4\cdot zz - 8\cdot uu \\ \frac{d}{dt}\,uu = 7\cdot xx - 11\cdot yy + 22\cdot zz - 10\cdot uu \\ \frac{d}{dt}\,yy = 6\cdot xx + 4\cdot yy + 4\cdot zz - 8\cdot uu \\ \frac{d}{dt}\,zz = 6\cdot xx + 3\cdot yy + 5\cdot zz - 8\cdot uu \end{bmatrix} = \begin{bmatrix} true \\ true \\ true \\ true \end{bmatrix}$$

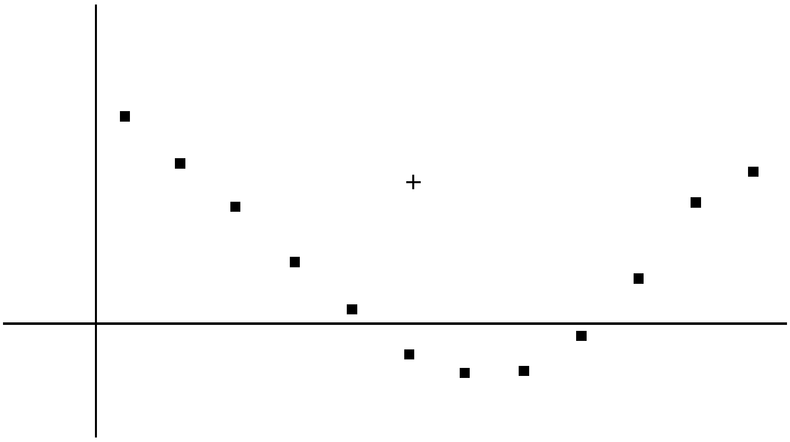
An Application of the Integral - Examining Home Heating Costs

Carl Leinbach & Marvin Brubaker
Gettysburg College, PA, USA

The home heating industry has determined that the idea of a Heating Degree Day (HDD) is a useful measure for determining the heating requirements of fuel oil customers. To determine the number of HDD in a given day, merely take the average of the daily high and daily low and subtract it from 65, the base temperature for this measure of "comfort". To determine the fuel necessary to heat a particular house for an individual, simply monitor the fuel consumption over a relatively short (say 10 day) period and keep track of the total number of daily HDD during the period. Using a professor´s home as an example, it was determined during the period from Jan 11 to Jan 21 the professor used 167 gallons of fuel and there were a total of 367 daily HDD, i.e. the professor required 0.444 gallons of fuel oil per HDD. The students then set to answering the following question:

How many gallons of fuel might the professor be expected to use during the year?

They consulted a weather almanac supplied by the local television station. This almanac gave the average daily high and low temperatures for each month of the year. The students used this information to compute the average HDD for the month. They chose the middle of the month as the "representative day" for each month and plotted the data from the almanac. This discrete plot is shown below.



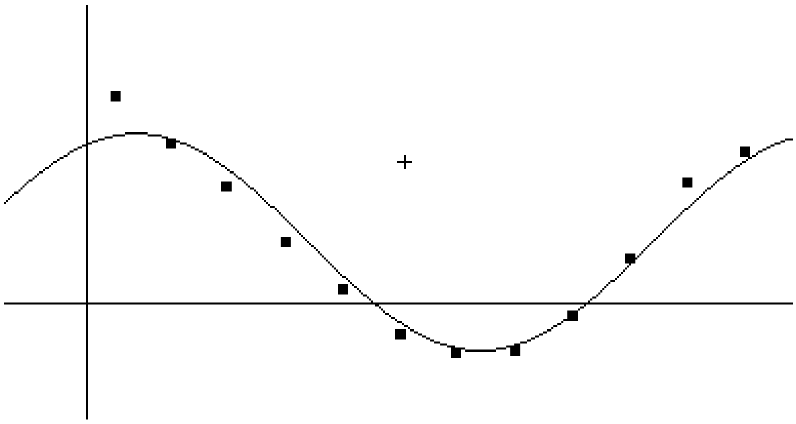
Susquehanna Valley Temperatures*

Month	HDD
Jan	44.0
Feb	34.0
Mar	25.0
Apr	13.0
May	3.0
Jun	- 6.5
Jul	- 10.5
Aug	- 10.0
Sep	- 2.5
Oct	9.5
Nov	25.5
Dec	32.0

Now the real work begins! The students noticed the obvious. The plot of the points appears to follow the graph of a Sine or Cosine function. A very brief discussion convinces them that this observation makes sense. Heating Degree Days should, indeed be a periodic function. They were also easily convinced that it made no difference whether they chose a Sine or Cosine function. To make the problem a bit more interesting, we chose a Sine function (you may, perhaps, choose to repeat this exercise using a Cosine function.) First they observed that the maximum value of the function is 36 and the minimum value is - 10.5, a net difference of 46.5. Thus it is reasonable to assume that the amplitude of the Sine term of this function is approximately 23 and the function is centered at approximately $y = 13$. The period of the function is, of course, one year or 365 days. The origin appears to be shifted to the right about 300 days. The net result of these heuristics is the function:

$$h(t) = 23\sin\left(\frac{2\pi}{365}(t - 300)\right) + 13$$

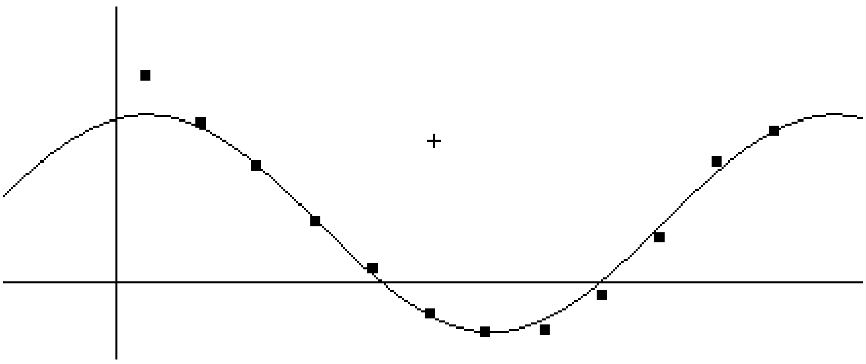
The graph of this equation and the original appears below. Note that this graph agrees, in general, with the shape of the data, but could be improved. It is tempting to think of using a curve fitting technique at this point. However, the students decide to use the tools that they have developed and have confidence in their ability to use.



Instead of introducing a sophisticated technique, the students look at the graph of the function and the data. What improvements are needed? First of all, they notice that it needs to be shifted to the left. How is this accomplished? After trying some different values for the shift of the curve, it is also determined that the graph is a little too high. They finally decide on this curve:

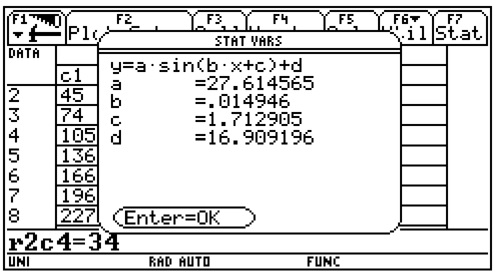
$$h2(t) \;=\; 23\sin\left(\frac{2\pi}{365}\,(t \;-\; 290)\right) + 12.5 \quad (\;\sim\; 23\cdot\mathrm{SIN}(0.0172\cdot t - 4.99) + 12.5)$$

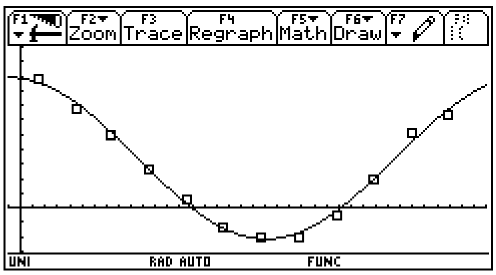
This is the graph of their function:



This graph appears to provide a reasonable fit to the data. Now the students must decide what to do with the graph.

You can compare the results from above with the fitting curve generated on the TI-92 using the same data together with its plot. Josef





If the normal heating season occurs when the Heating Degree Days lie above zero then the heating season for this area ends about day 140 (20 May) and resumes about day 253 (September 10). Furthermore, the area lying above the horizontal axis represents the total accumulated Heating Degree Days for this area. (This concept is diffucult for some students. For them area is square units and nothing else!) The idea that an area is accumulated HDD takes some time and a great deal of understanding of the meaning of the definite integral, but that is the purpose of the example.)

After the students have learned to relate the accumulated HDD during this period to the definite integral of *h2*(*t*), the next step is to relate this total to the number of gallons of fuel oil needed by the professor. This varies according to the particular home that is being heated and the habits of the people living in the home. Recall that the following data was supplied to the students: during the period from January 11 to January 21, the daily newspaper recorded 376 HDD, and the professor consumed 167 gallons of fuel. This means that the professor appears to use 0.444 gallons of fuel for each HDD. Thus, the total fuel that the professor can be expected to use during the year is 0.444 times the total accumulated Heating Degree Days. This is determined by evaluating the definite integral:

$$\int_{253}^{505} \left(12.5 + 23\cdot \mathrm{SIN}\left(\frac{2\cdot\pi}{365}\cdot(t - 290)\right)\right)\, dt = 5358.6$$

Multiplying this result by 0.444, it is determined that it will require, on the average 2 380 gallons of fuel oil to heat the professor´s home.

A more interesting question is the following: Suppose the profesor´s fuel oil tank holds 225 gallons of fuel oil. The oil company likes to deliver oil when the tank has about 25 gallons of oil remaining. If a delivery was made on February 15 (day 46 of the year), when should the next delivery be expected? Allowing *d* to denote the day of delivery, the problem is to solve the following equation for *d*.

$$\int_{46}^{d} \left(12.5 + 23\cdot \mathrm{SIN}\left(\frac{2\cdot\pi}{365}\cdot(t - 290)\right)\right)\, dt = \frac{200}{0.444}$$

This type of equation is almost never seen in a calculus course although many hours are spent impressing upon the students that the function defined by $F(x) = \int_a^x f(t)\, dt$ is the antiderivative of *f*(*x*). Thus, if *f*(*x*) is positive, *F*(*x*) is increasing, and must at least locally, have an inverse, i.e. the above equation is solvable for *d*.

Simplifying the above expression and using the approXimate operator, we have

$$-\ 1336.1\cdot \mathrm{COS}(0.0172\cdot d + 1.29) + 12.5\cdot d - 1229.7 = 450.4$$

Switching *DERIVE* to Approximate Mode and looking for a solution in the range of 46 to 150, we find that *d* has a value of approximately 61. [d = 60.6]. Thus, the next delivery can be expected on the 2nd of March, if it is not a leap year.

This application is significantly different from the standard text examples. However, it gives strong emphasis to the point that the definite integral is a tool to be used in situations where one is considering the accumulation of values of a function. The phrase “Area under the Curve” can mean something other than square units. The symbolic capabilities of *DERIVE* allowed us to make this point without becoming bogged down in calculations that can detract from our main agenda.

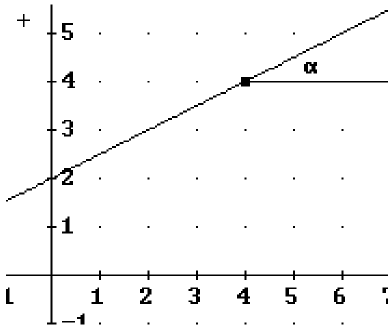
Reflections

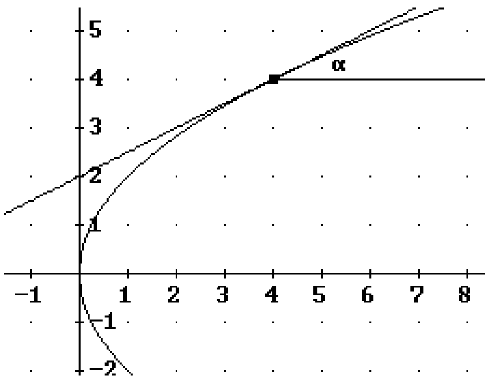
Hubert Voigt, Perg, AUT

In the final-year classes I try to find problems which are able to connect new contents with knowledge from earlier years. I was inspired to the following by a contribution in the TI-News "Mathematics in a coffee cup".Firstly we investigate lines, angles and slopes followed by trigonometric discussions.

1. Given is the line y = 0.5x + 2. A horizontal light ray coming from the right hits the line in P(4|y) and is reflected by it. What is the angle of incidence? Where is the intersection point between the reflected ray and the x-axis? Deduce the intersection angle between ray and x-axis.

   Solution: $\alpha' = 2\alpha$.





As in my type of school - a business highschool - the conics are not part of the curriculum, I use some selected examples to remind the students on their existence - well known from SI - pointing at headlights, parabolic receiver bowls, whisper galleries and others.

2. Given is the parabola $y^2 = 4x$. The light ray from above is reflected by the parabola. Some time or other we have heard, that all reflected rays meet in the focus F(e|0).

Now the curve´s tangent in P takes the role of a mirror.

$$y = 2\sqrt{x} \text{ – upper half – with } y' = \frac{1}{\sqrt{x}}$$

Show that both rays reflected in A(2|y) and B(6|y) - with A and B being points on the curve - meet in the same point on the x-axis.

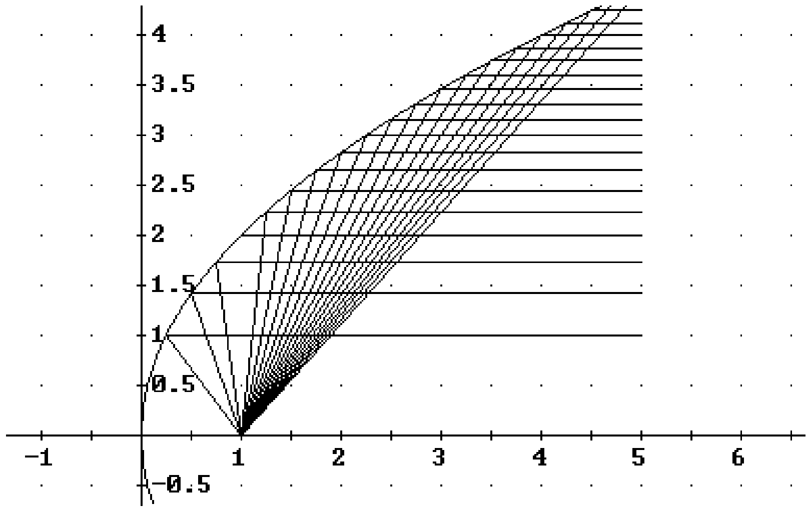
General solution: $P(x\,|\,2\sqrt{x}),\quad \alpha = \tan^{-1}\left(\frac{1}{\sqrt{x}}\right);\ \alpha' = 2\alpha = 2\tan^{-1}\left(\frac{1}{\sqrt{x}}\right)$, which gives the slope of the reflected line: $m = \tan\left(2\tan^{-1}\left(\frac{1}{\sqrt{x}}\right)\right)$. Even without any sum rule for the trig-functions a CAS like *DERIVE* is helpful:

$$\mathrm{TAN}\left(2\cdot\mathrm{ATAN}\left(\frac{1}{\sqrt{x}}\right)\right) = \frac{2\cdot\sqrt{x}}{x-1}$$

Thus we find the family of lines passing $P(x_1\,|\,2\sqrt{x_1})$: $y - 2\sqrt{x_1} = \frac{2\sqrt{x_1}}{x_1 - 1}(x - x_1)$. Setting *y* = 0 and solving for x_1 we can find very easily

$$-2\sqrt{x_1}\,(x_1 - 1) = 2\sqrt{x_1}\,(x - x_1) \Rightarrow -x_1 + 1 = x - x_1 \Rightarrow x = 1$$

All reflected rays meet in F(1|0)



Generalisation would be unable without using a CAS because my students don´t know the relations between the trig-functions and their inverses.

Discuss the case of $x_1 = 1$!!

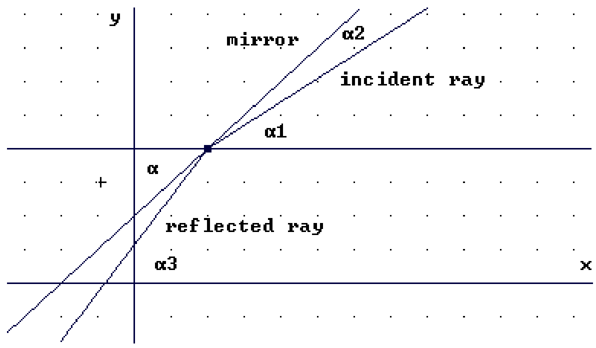
(in Calculation and Plot)

All reflected rays meet in F(1|0)

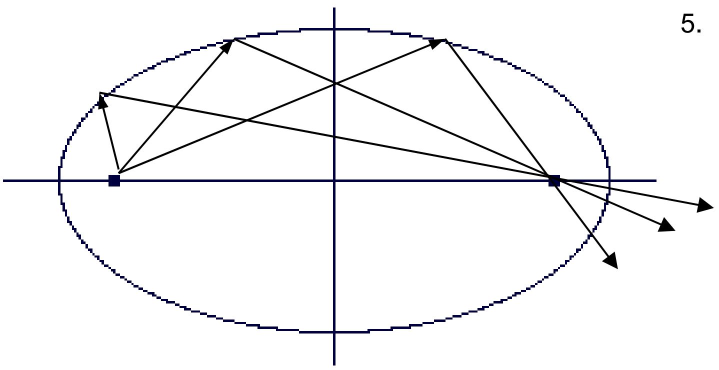
$$\text{VECTOR}\left(\begin{bmatrix} 5 & 2\cdot\sqrt{x} \\ x & 2\cdot\sqrt{x} \\ 1 & \dfrac{2\cdot\sqrt{x}}{x-1}\cdot(1-x) + 2\cdot\sqrt{x} \end{bmatrix}, x, 5, 0, -0.25\right)$$

3. Given is the line y = 0.5x + 2 and a light ray coming from the right, which has slope m = 0.7. The ray meets the line in P(2|4). Find the intersection of the reflected ray with the x-axis and its angle of incidence.

4. Take the line from above and a ray with slope m = 1.5 coming from below, which is reflected in P(2|4). What is the equation of the reflected ray?



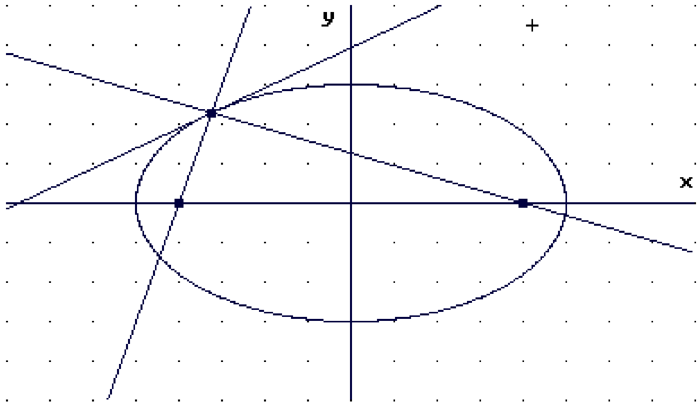
Hint: Known are α_1 and α in problem 3 and α_3 and α in problem 4. It can easily be shown by inspection only that $\mathbf{2\alpha = \alpha_1 + \alpha_3}$.



5. Given is an ellipse

$$9x^2 + 25y^2 = 225.$$

Its focal points are $F_1(-4|0)$ and $F_2(4|0)$. Show that all rays leaving F_1 and being reflected by the ellipse (its tangent in the meeting point) meet in F_2. (Principle of Whisper Galleries).



Example: Take a ray from F_1 with slope m = 3, which is equal to angle α_3 = = 71.565°. The intersection point P (line and upper half ellipse) has coordinates [−3.236, 2.286].
For the slope of the tangent use calculus:

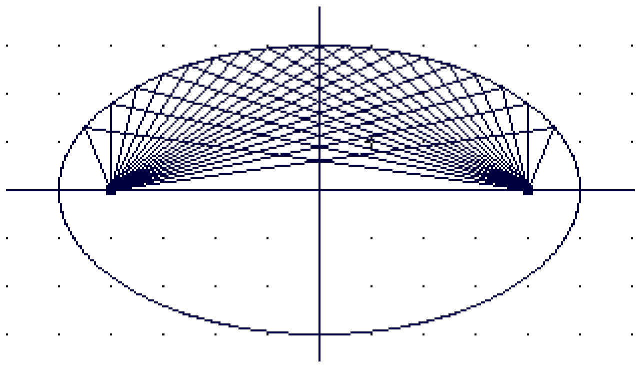
$$\frac{d}{dx} E(x) := 0.6\cdot\sqrt{(25 - x^2)}$$

$$-\frac{3\cdot x}{5\cdot\sqrt{(25 - x^2)}}$$

The slope of the tangent is 0.509939, thus α = = 27.0187. The relation derived above delivers for α_1 = -17.5275° (angle between reflected ray and *x*-axis) →

See the last lines of the student´s *DERIVE* activity.

```
TAN(−17.5275) = −0.315826

y = − 0.315826·(x + 3.23802) + 2.28593

− 0.315826·(x + 3.23802) + 2.28593 = 0

 [x = 3.99992]
```



Setting Simplification Mode to Trigonometry - Expand you get a very comfortable expression for bulky tangens of α_1, which represents the generalized slope of the reflected rays.

Using this expression it is not too difficult even for students to generate a pattern of rays together with their reflected partners.

$$\mathrm{TAN}\left(2\cdot\mathrm{ATAN}(\mathrm{F\_ABL}(x)) - \mathrm{ATAN}\left(\frac{E(x)}{x + 4}\right)\right) = \frac{3\cdot\sqrt{(25 - x^2)}}{5\cdot(x - 4)}$$

```
VECTOR([[−4,0],[x,E(x)],[4,TAN(2*ATAN(F_ABL(x))−ATAN(E(x)/(x+4)))
        *(4−x)+E(x)]],x,−5,5,0.501)
```

(You might ask for the reason for this very unusual value 0.501 as increment in the VECTOR-command).

General proof: $y - y(x_1) = \dfrac{y(x_1)}{x_1 - 4}(x - x_1)$ with $x_1 \neq 4$ and $y = 0$ gives $x = 4$ for all reflected rays,

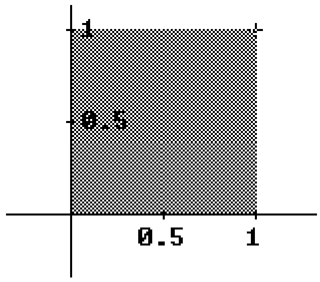
so they all will meet in F_2.

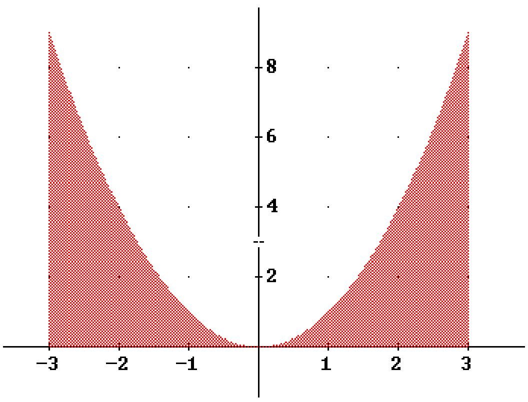
(DfW5 offers new methods to shade areas defining them by using relations and combining them by applying logic operators. Josef Lechner collected some examples and added a shadow-function. Can you assign each statement to its respective plot? What makes the difference between #4 and #7? One graph appears the same in gray for two expressions. Which one and why? Josef)

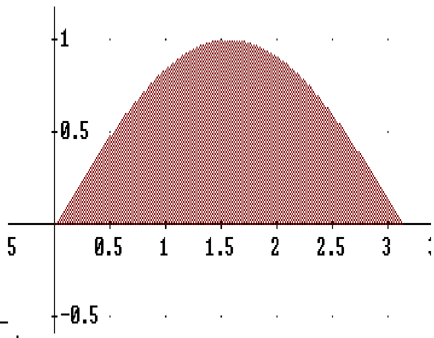
Shading in DfW5 using Logic Operators

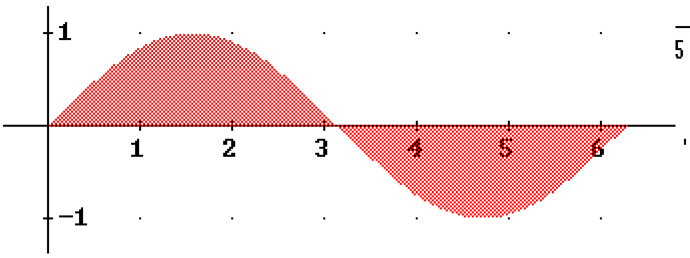
Josef Lechner, Viehdorf, AUT

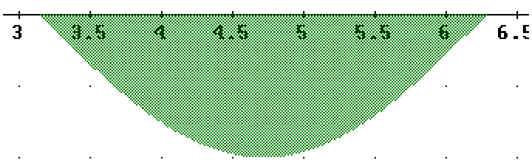
```
#1:   0<=x<=1 AND 0<=y<=1

#2:   0<=x<=pi AND 0<=y<=SIN(x)

#3:   pi<=x<=2*pi AND SIN(x)<=y<=0

#4:   [0<=x<=pi AND 0<=y<=SIN(x),pi<=x<=2*pi AND SIN(x)<=y<=0]

#5:   x>=0 AND y>=0 AND y-x/2-2<=0 AND 2*x+y-6<=0

#6:   shadow(f,int):=int SUB 1<=x<=int SUB 2 AND SUBST(IF
                   (f_>=0,0<=y<=f_,f_<=y<=0),f_,f)

#7:   shadow(SIN(x),[0,2*pi])

;Simp(#7)
#8:   x<=2*pi AND 0<=x AND IF(SIN(x)>=0,0<=y<=SIN(x),SIN(x)<=y<=0)

#9:   shadow(x^2,[-3,3])

;Simp(#9)
#10: x<=3 AND y<=x^2 AND -3<=x AND 0<=y
```

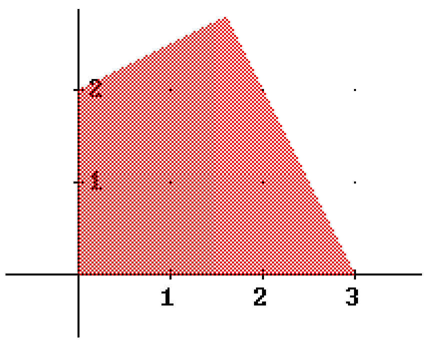




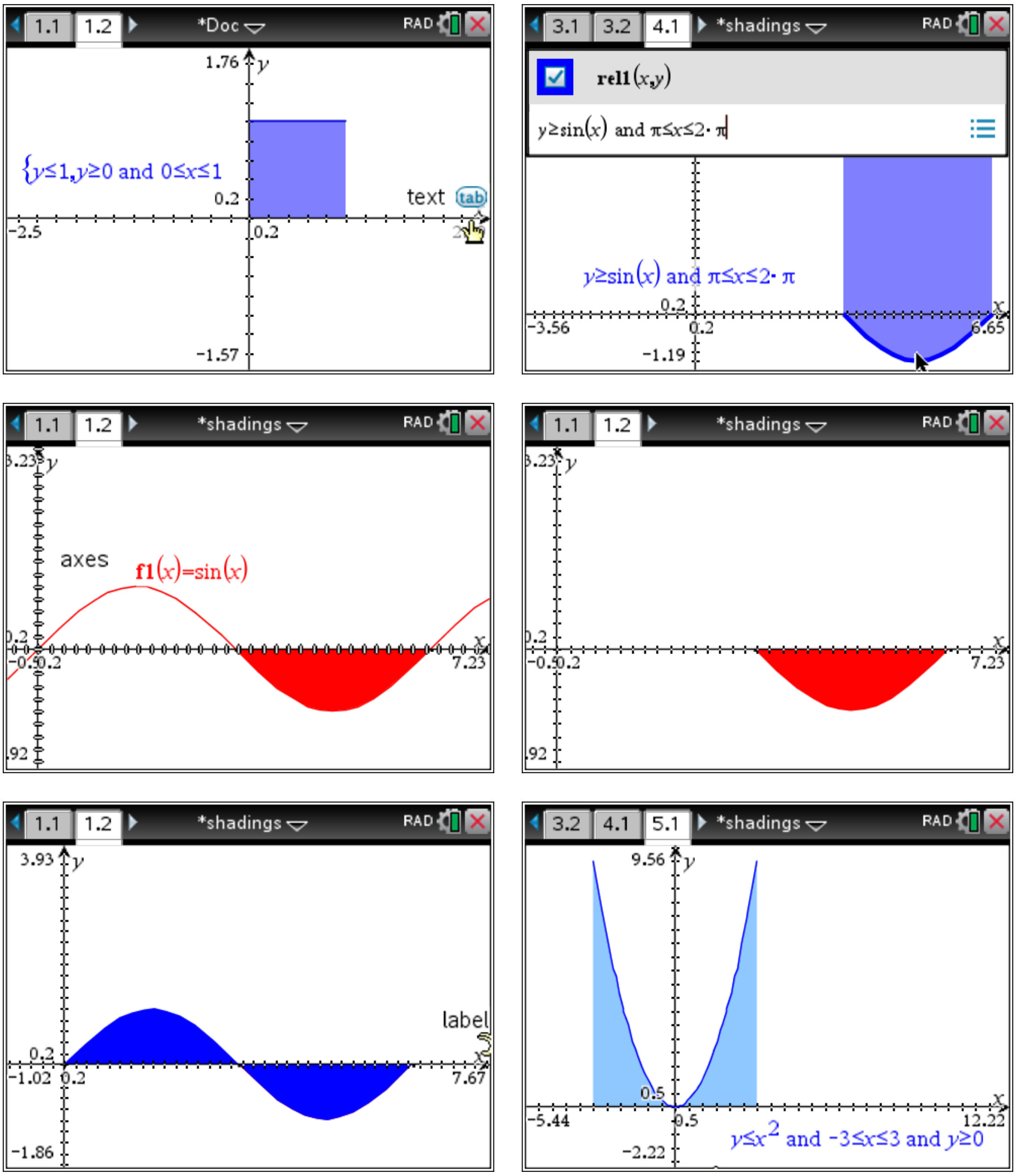








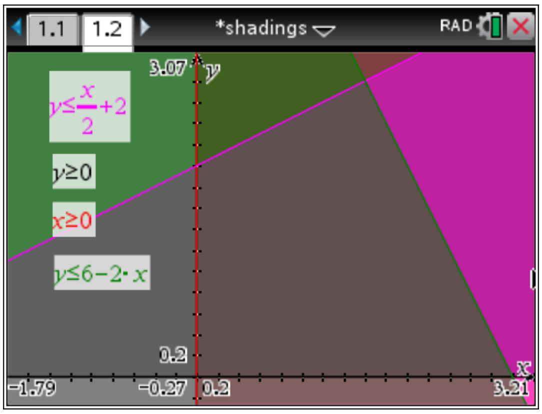
I turn on my TI-Nspire and try to shade the areas in on a Graphs page. The square works as expected. I could not convince TI-Nspire to plot y≥sin(x) and π≤x≤2*π and y≤0. It does not accept the second logic operator. So I helped myself by "Analysing the Graph" and calculating the integral which shades automatically the calculated area.

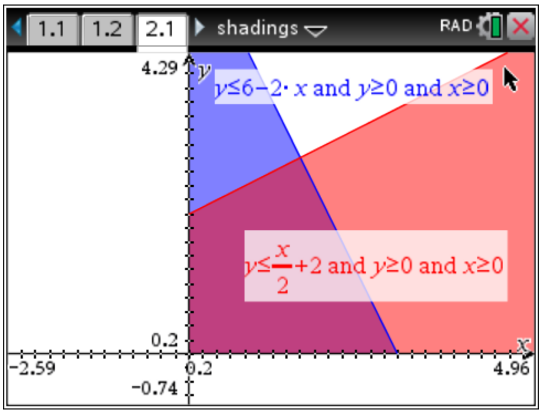


It is interesting that in case of the parabola both restrictions are accepted. Maybe that I did it wrong with the sin-function? Do you know how to do?

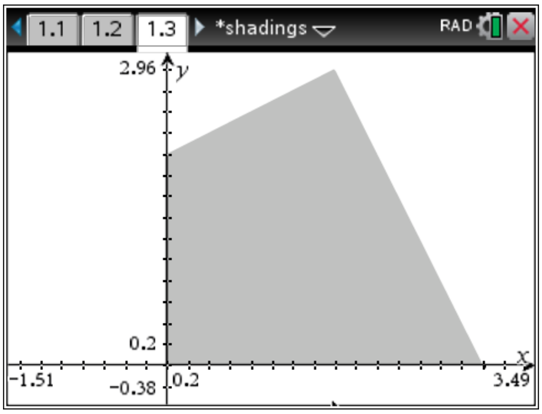
The last problem was shading the polygon – having linear programming with two variables in mind.

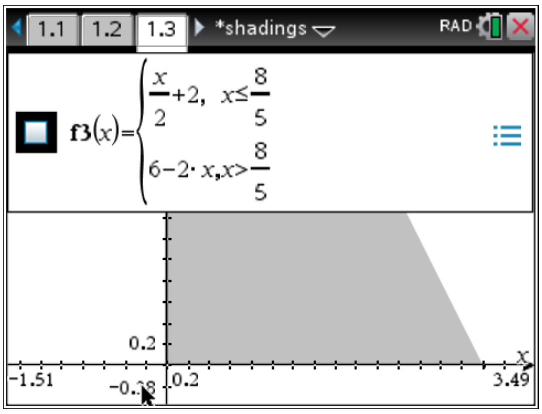
I was not able to do it in a one-line-relation. In the first row I plot the intersection of four inequalities and then better of only two inequalities.



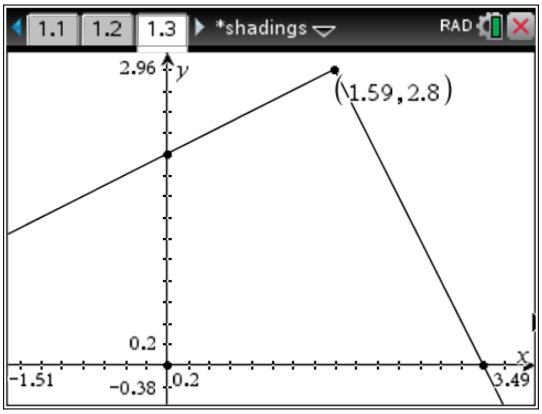


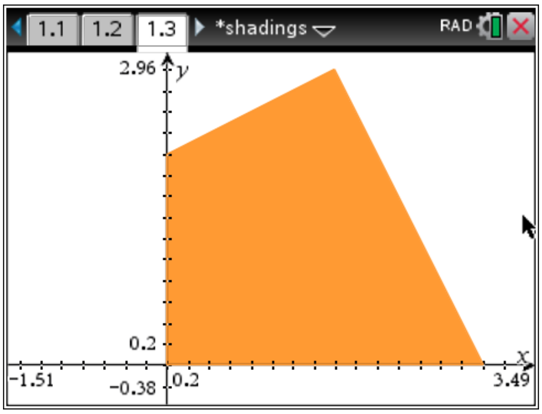
Next way is plotting the integral of the piecewise defined function. But in this case you have calculate the intersection point of the lines first.





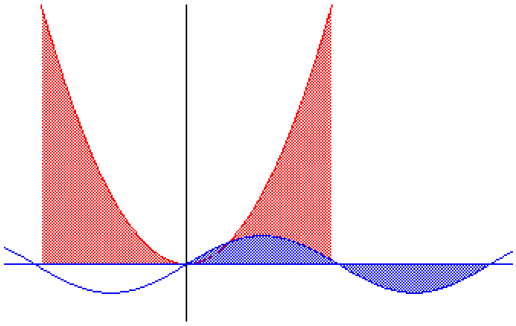
Last try: find the intersection points, produce the polygon and fill it with a color of your choice. Then hide the lines.





DERIVE 6 provides another option:

```
AreaBetweenCurves(SIN(x), 0, x, 0, 2·π, y)

                     2
AreaBetweenCurves(x , 0, x, -3, 3, y)
```



Nested IF() via a Recursive Function

G P Speck, Wanganui, New Zealand

AUTHORING a lengthy nested IF() statement can be somewhat tedious and is certainly repetitive from one application to the next.

A wide range of nested IF() applications can be formalised in the following way.

For A_(x):=[A_(x)$_1$ + A_(x)$_2$ + ... + A_(x)$_n$] and
B_(x):=[B_(x)$_1$ + B_(x)$_2$ + ... + B_(x)$_n$]

we wish to create a function T_(x) of the form:

T_(x):=IF(A_(x)$_1$,B_(x)$_1$,IF(A_(x)$_2$,B_(x)$_2$, ..., IF(A_(x)$_n$,B_(x)$_{n)}$ ...)

where there are n closing)´s on the right (= $_{n)}$).

The function T_(x) can be defined by T_(x) := S_(x,DM(A_(x))) where

DM(m) := DIMENSION(m) and S_(x,k) is given recursively by:

$$\left[\begin{array}{cc} T\_1(x) := (A\_(x))_{DM(A\_(x))} & T\_2(x) := (B\_(x))_{DM(A\_(x))} \\ T\_3(x,\ k) := (A\_(x))_{1 + DM(A\_(x)) - k} & T\_4(x,\ k) := (B\_(x))_{1 + DM(A\_(x)) - k} \end{array}\right]$$

S_(x,k):=IF(k = 1, IF(T_1(x),T_2(x)),IF(2 ≤ k ≤ DM(A_(x)),IF(T_3(x, k),T_4(x, k), S_(x, k - 1))))

At this point the reader may feel that though the above definition of a function such as T_(x) is all well and good as an academic exercise, it will be of little practical use in *DERIVE* applications because of its complexity. For instance, we know that if one attempts to PLOT even mildly 'complicated' functions via IF( ) statements in *DERIVE* , the note 'Sorry, the highlighted function cannot be plotted ' appears on the screen. However, if your computer has sufficient RAM and computational speed (as should be the case if using *DfW*) this is no significant drawback since SIMPLIFY combined with the appropriate Options - Points - Connect(yes/no) - Small/Medium/Large enables one to PLOT quite complicated IF( ) expressions more than adequately.

A NESTEDIF.MTH - utility file together with *DfW* applications of our T_(x) function follows.

#1: NESTEDIF.MTH

#2: CaseMode := Sensitive

#3: Precision := Exact

#4: [A_(x) :=, B_(x) :=, DM(m) := DIM(m)]

#5: $$\left[\begin{array}{cc} T\_1(x) := (A\_(x))_{DM(A\_(x))} & T\_2(x) := (B\_(x))_{DM(A\_(x))} \\ T\_3(x,\ k) := (A\_(x))_{1 + DM(A\_(x)) - k} & T\_4(x,\ k) := (B\_(x))_{1 + DM(A\_(x)) - k} \end{array}\right]$$

```
        S_(x, k) :=
          If k = 1
             If T_1(x)
                T_2(x)
#6:          If 2 ≤ k ≤ DM(A_(x))
                If T_3(x, k)
                   T_4(x, k)
                   S_(x, k - 1)

#7:     T_(x) := S_(x, DM(A_(x)))
```

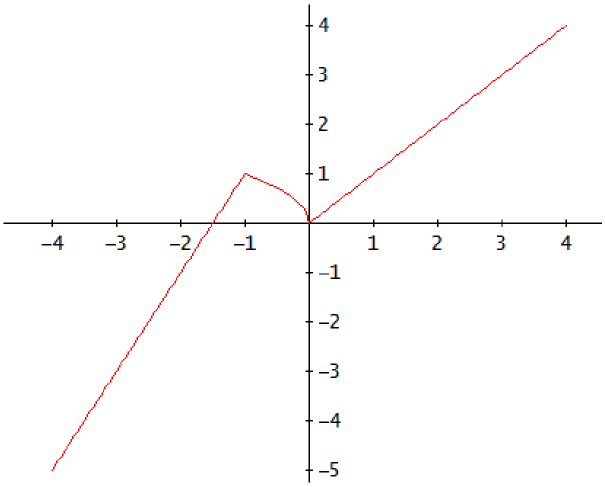
A piecewise defined function is given by the following:

#8: $\left[\begin{array}{c} A\_(x) := [x < -1,\ -1 \le x \le 0,\ x > 0] \\ B\_(x) := [2\cdot x + 3,\ \sqrt{(-x)},\ x] \end{array}\right]$

#9: $\left[T\_(-2),\ T\_\left(-\frac{1}{2}\right),\ T\_(3)\right] = \left[-1,\ \frac{\sqrt{2}}{2},\ 3\right]$

Simplify(*) and Plot the VECTOR with "Small - Connected Points".

#10: VECTOR([x, T_(x)], x, -4, 4, 0.05)



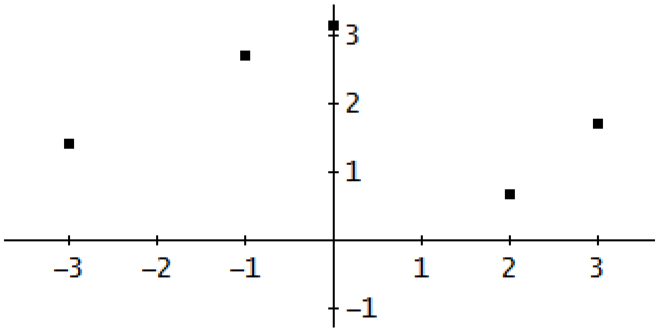
Finite sequence T_(x) is given:

#11: $\left[\begin{array}{c} A\_(x) := [x = -3,\ x = -1,\ x = 0,\ x = 2,\ x = 3] \\ B\_(x) := \left[\sqrt{2},\ e,\ \pi,\ \frac{2}{3},\ 1.7\right] \end{array}\right]$

#12: VECTOR(T_(k), k, [-4, -3, -1, 0, 2, 2.5, 3, 4]) = $\left[?,\ \sqrt{2},\ e,\ \pi,\ \frac{2}{3},\ ?,\ \frac{17}{10},\ ?\right]$

Simplify(*) and Plot the following VECTOR with "Large - Non-Connected Points

#13: VECTOR([x, T_(x)], x, -3, 3)

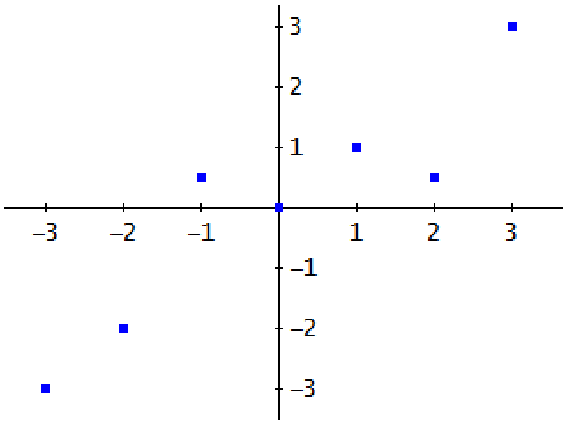


Sequence T_(x) defined on the integers is given, check specific T_(x)-values and then plot large points without connecting them. (Note: for T_(x) restricted to the positive integers, use `IF(x>0,T_(x))`

#14: $$\left[\begin{array}{c} A\_(x) := \left[\frac{x+2}{3} = \text{FLOOR}\left(\frac{x+2}{3}\right), \frac{x+1}{3} = \text{FLOOR}\left(\frac{x+1}{3}\right), \frac{x}{3} = \text{FLOOR}\left(\frac{x}{3}\right)\right] \\ B\_(x) := \left[x, \frac{1}{2}, x, \frac{1}{3}, x\right] \end{array}\right]$$

#15: $$\text{VECTOR}(T\_(x), x, -2, 2, 0.5) = \left[-2, ?, \frac{1}{2}, ?, 0, ?, 1, ?, \frac{1}{2}\right]$$

#16: VECTOR([x, T_(x)], x, -3, 3)



Another piecewise defined function, its value table and plot:

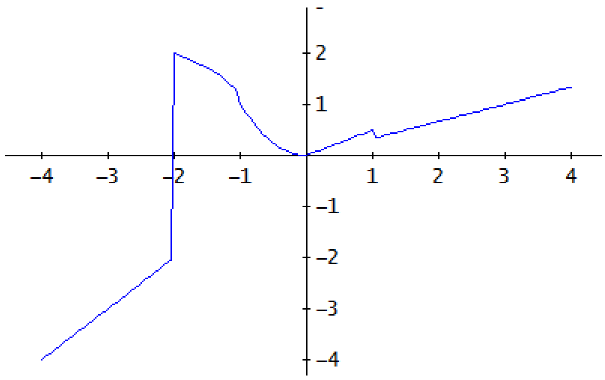
#17: $$\left[\begin{array}{c} A\_(x) := [x < -2, -2 \le x < -1, -1 \le x \le 0, 0 < x \le 1, x > 1] \\ B\_(x) := \left[x, 1 + \sqrt{(-1 - x)}, x^2, \frac{x}{2}, \frac{x}{3}\right] \end{array}\right]$$

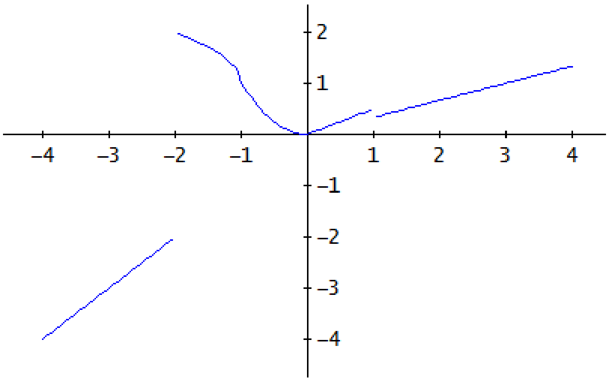
#18: $$\text{VECTOR}(T\_(x), x, [-3, -1.5, 0, 0.5, 2]) = \left[-3, \frac{\sqrt{2}}{2} + 1, 0, \frac{1}{4}, \frac{2}{3}\right]$$

#19: VECTOR([x, T_(x)], x, -4, 4, 0.05)

#20: $\text{VECTOR}([x, \text{IF}(x \ne -2 \wedge x \ne 1, T\_(x))], x, -4, 4, 0.05)$

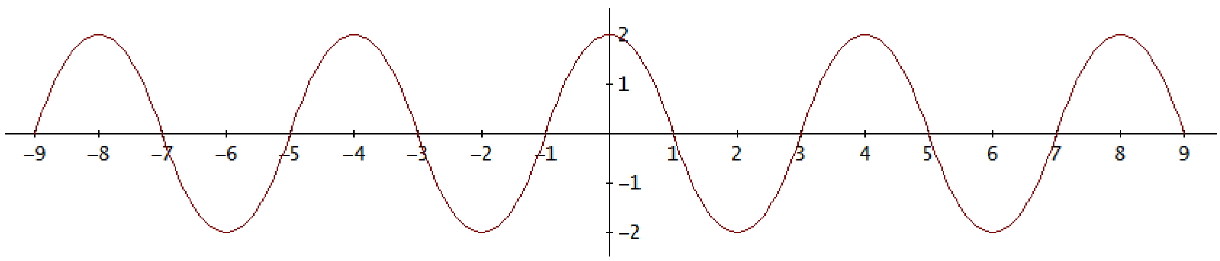
See the difference:





(I could not resist to try this utility file to produce a chain of parabolas. Both the ranges and the curves are produced by a VECTOR-statement, Josef)

```
#21:  Try this, Josef

      [          A_(x) := VECTOR(2·k - 1 ≤ x ≤ 2·k + 1, k, -4, 4)                    ]
#22:  [                               k + 1            2         k                  ]
      [ B_(x) := VECTOR((-1)     ·2·(x - 2·k)  + (-1) ·2, k, -4, 4)                ]

#23:  VECTOR([x, T_(x)], x, -9, 9, 0.05)
```



(It took me 10.1sec to produce the list of points. I tried also with DERIVE XM from 1995 and had no problems, it took me 8.0sec. This was in 2000.)

(*) It is not necessary to simplify (or approximate) before plotting with DERIVE 6. Go to the plot window and plot the expressions.

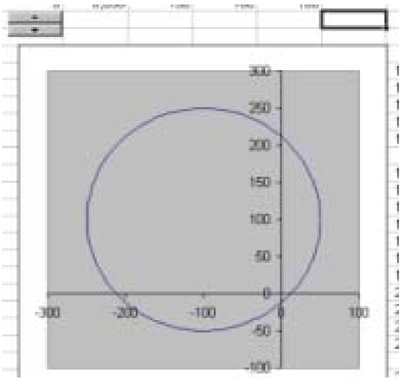
A Metamorphosis in MS-EXCEL, DERIVE and on the TIs

David Sjöstrand, SWE & Josef Böhm, AUT

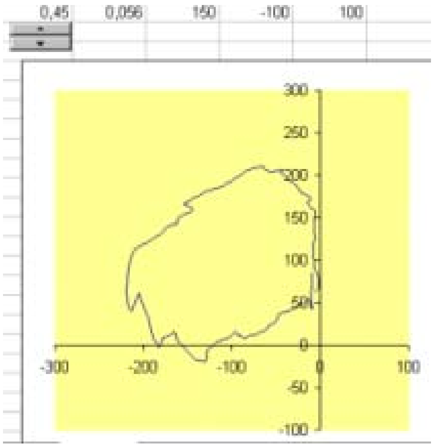
Could you imagine to perform morphing even on the TI-92. I´ll present a nice example of using vectors, functions, coordinates and a bit phantasy. Another pretty thing is to show again that in many cases it is possible to transfer an idea from one software to the other.

The start:

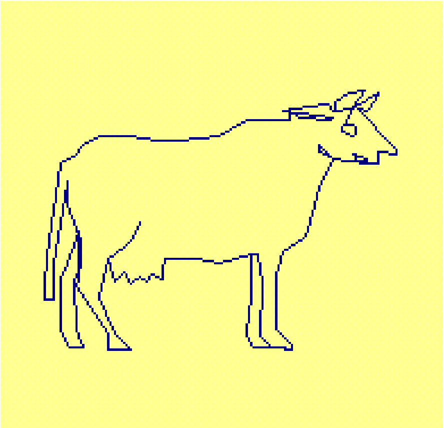
In the first days of February I received a mail from my friend David. He wrote that he used "weighted mean values" to transform one 2D-object into another one. I found an Excel-file as attachment. David asked me to press the buttons. I had no idea what to do with "weighted mean values", but I loaded the Excel-file with the inviting file name "COW.XLS" and was faced by a circle and two buttons.



And I pressed the buttons - what happened?? Very smoothly the circle changed its contour ......



.... and finally .....



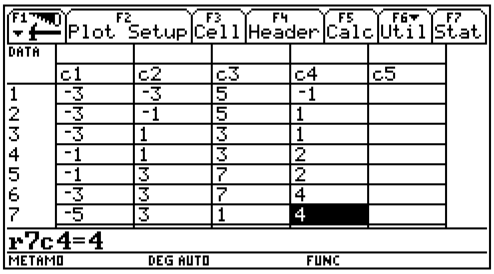
.... a cow appeared on the screen. I was fascinated, but I didn´t discover the machines behind the scenery. e-mailing is much more comfortable and I pressed another button on my computer to answer and asked David for the "trick" or recipe how to do that.

The next step, David´s answer:

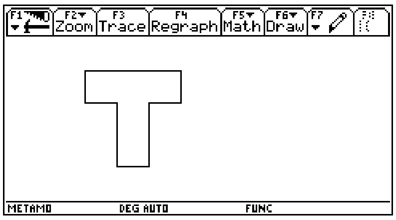
Hi, Josef, if s varies from 0 to 1 the curve (1 – s)[x1(t),y1(t)] + s*[x2(t),y2(t)] varies from the curve [x1(t),y1(t)] to the curve [x2(t),y2(t)] passing all "weighted mean values" of the two curves. In the same way (1-s)*A + s*B (= A + s * (B - A)) varies from a pig to a rabbit if A and B are matrices containing coordinates for a pig and a rabbit respectively. This means if I only had a coordinate matrix for two animals I could easily do the same with them as I did with the circle and cow.*
Best regards, David.
PS: I received the coordinates for the cow from Tomass Romanovskis in Särö-97.

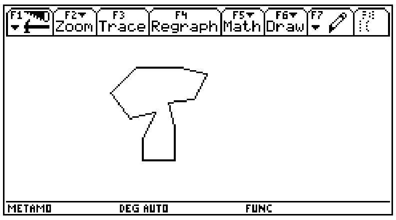
The TI-92 appears:

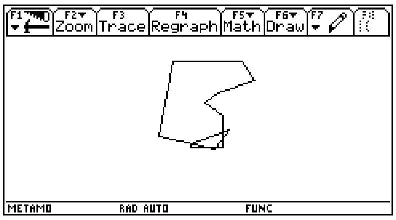
That was it. Immediately I felt the strong demand to transfer this idea onto my TI-92 to have an additional attraction for my pupils to animate their LOGOs (DNL#32). I designed two letters in coordinates (they need to be generated by the same number of points) and created a new DATA-sheet named d2 - because of 2 objects. (Columns c1 and c2 contain the coordinates of a "T" and c3 together with c4 describe an "E". Then I wrote some lines of program code and it worked:

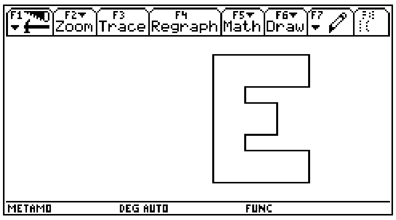


	c1	c2	c3	c4	c5
1	-3	-3	5	-1	
2	-3	-1	5	1	
3	-3	1	3	1	
4	-1	1	3	2	
5	-1	3	7	2	
6	-3	3	7	4	
7	-5	3	1	4	



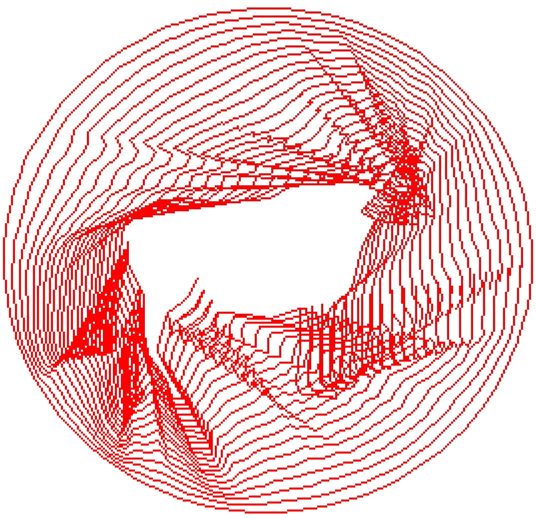







```
meta2()
Prgm
Local t
PlotsOff : FnOff
For t,0,1,.02
d2[1]+t*(d2[3]-d2[1])»lx
d2[2]+t*(d2[4]-d2[2])»ly
NewPlot 1,2,lx,ly,,,,5
DispG
EndFor
DelVar lx,ly
PlotsOff
EndPrgm
```

I sent the program to David and he introduced the next partner.



The DERIVE cow is on the screen:

Find attached the MTH-file containing the matrices COW and CIRCLE and a GIF-file containing the result of a plot of

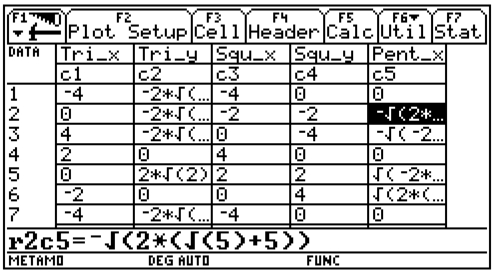
VECTOR(t cow + (1-t) circle, t, 0, 1, 0.05).

COW_CIRCLE.MTH.

I think how to gradually transform a matrix or a curve C1 to a matrix or a curve C2 and then transform C2 to C3 and so on. If you have a chain of curves C1, C2, C3, ... Cn, you need to derive a certain polynomial of degree n – 1. You really need computer algebra to derive this polynomial you will soon hear from me. Regards, David

The multiple metamorphosis:

Now it was my turn again. I felt the challenge to produce a - more or less - smooth change from a triangle - via square and regular pentagon - to a regular hexagon. I designed the figures - using 6 poiints for each of them.

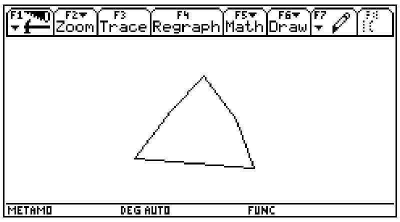


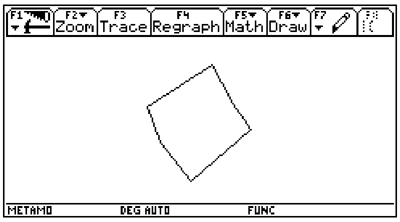
DATA	Tri_x	Tri_y	Squ_x	Squ_y	Pent_x
	c1	c2	c3	c4	c5
1	-4	-2*√(...	-4	0	0
2	0	-2*√(...	-2	-2	-√(2*...
3	4	-2*√(...	0	-4	-√(-2...
4	2	0	4	0	0
5	0	2*√(2)	2	2	√(-2*...
6	-2	0	0	4	√(2*(...
7	-4	-2*√(...	-4	0	0

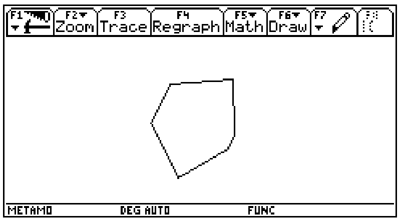
r2c5=-√(2*(√(5)+5))

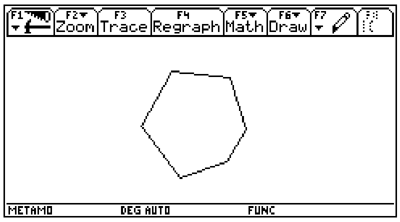
Then I changed my program using one more for-next-loop and transformed one figure into the other. It worked properly with a reasonable speed.

Program call: meta(4) - because of 4 objects. You find the data in d4.









A toe is not a TOE

Josef, find attached the file TOE.XLS and TOE.MTH. The idea is to find a 2nd degree polynomial, f(t), such that f(0) = A12, f(0.5) = C12 and f(1) = E12 (suitable cells in an Excel sheet). If you send me the letters H, I, J, S and F I could create a polynomial of degree 6 and send you an Excel message HI JOSEF. Regards, David.

#1: InputMode := Word

#2: $f(t) := d\cdot t^2 + e\cdot t + g$

#3: $\text{SOLVE}([f(0) = obj1,\ f(0.5) = obj2,\ f(1) = obj3],\ [d,\ e,\ g])$

#4: $[d = 2\cdot obj1 - 4\cdot obj2 + 2\cdot obj3 \wedge e = -3\cdot obj1 + 4\cdot obj2 - obj3 \wedge g = obj1]$

#5: $f(t) := (2\cdot obj1 - 4\cdot obj2 + 2\cdot obj3)\cdot t^2 + (-3\cdot obj1 + 4\cdot obj2 - obj3)\cdot t + obj1$

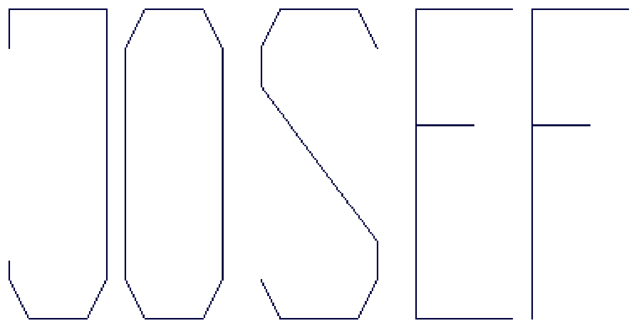
#6: $[f(0),\ f(0.5),\ f(1)] = [obj1,\ obj2,\ obj3]$

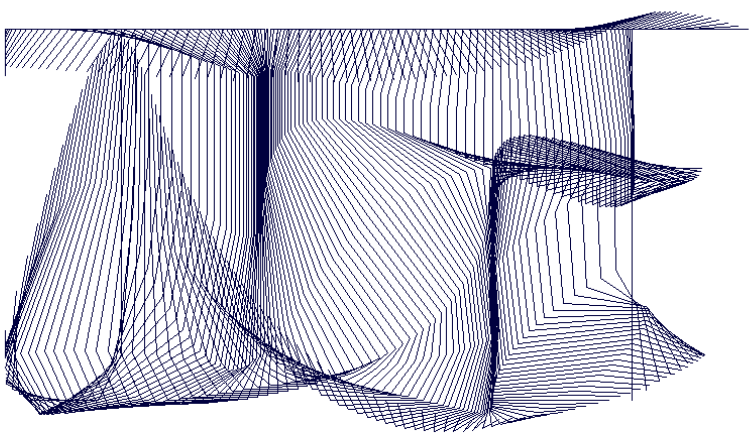
You can find one example with objects = t_, o, and e in file toe.mth which transforms a “T” via an “O” to an “E”.

You can find below the *DERIVE* file how to find the function for generalising a morph-function for four objects followed by the resepective TI-program. But the TI-program doesn´t work very fast, although it is a very nice and attractive algorithm for applying one single parameter and vector calculations.

```
F(x):=a*x^3+b*x^2+c*x+d

SOLVE([F(0)=c1,F(1/3)=c3,F(2/3)=c5,F(1)=c7],[b,c,a,d])

[[b=4.5*(2*c1-5*c3+4*c5-c7),c=-0.5*(11*c1-18*c3+9*c5-2*c7),
a=-4.5*(c1-3*c3+3*c5-c7),d=c1]]
```

```
metaf()
Prgm
Local  t
PlotsOff  : FnOff
For  t,0,0.99,0.01
ª9/2*(d4[1]-3*d4[3]+3*d4[5]-d4[7])*t^3+9/2*(2*d4[1]-5*d4[3]+4*d4[5]-d4[7])*t^2-1/2*(11*d4[1]-18*d4[3]+9*d4[5]-2*d4[7])*t+d4[1]»lx
ª9/2*(d4[2]-3*d4[4]+3*d4[6]-d4[8])*t^3+9/2*(2*d4[2]-5*d4[4]+4*d4[6]-d4[8])*t^2-1/2*(11*d4[2]-18*d4[4]+9*d4[6]-2*d4[8])*t+d4[2]»ly
NewPlot  1,2,lx,ly,,,,5
DispG
EndFor
d4[7]»lx:d4[8]»ly
NewPlot  1,2,lx,ly,,,,5
DispG
DelVar  lx,ly
PlotsOff
EndPrgm
```



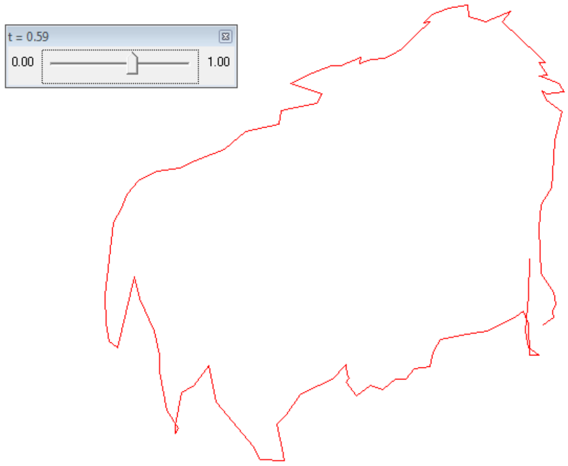


I produced "JOSEF" in coordinates and found the function to change from "J" through all letters to "F".

This results in a nice pattern on the *DERIVE* Plot Window and in a smooth moving and changing of letters on the TI-Graph Screen.

I had a lot of fun revising this DNL from 2000 based on DfW4 (with a glimpse in DERIVE 5 provided by Johann Wiesenbauer in his Titbits 16) and the TI-92. What can we do now with DERIVE 6 and TI-NspireCAS CX (and what can we do not)?

The most important improvement is the possibility to introduce sliders for the parameter in DERIVE and TI-Nspire as well:

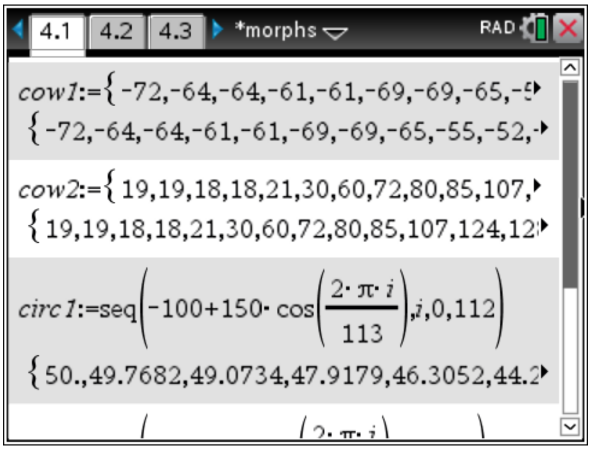


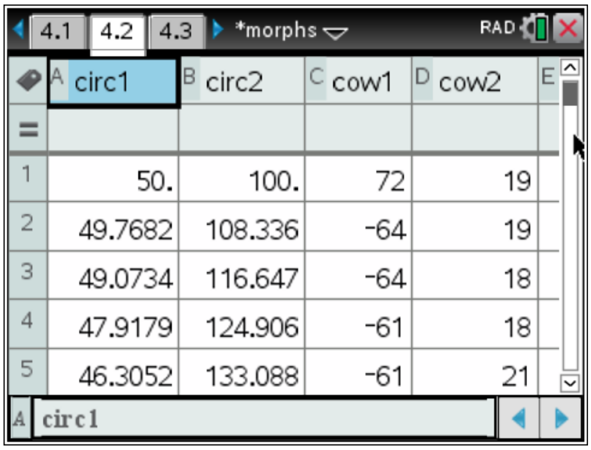
t·cow + (1 – t)·circle

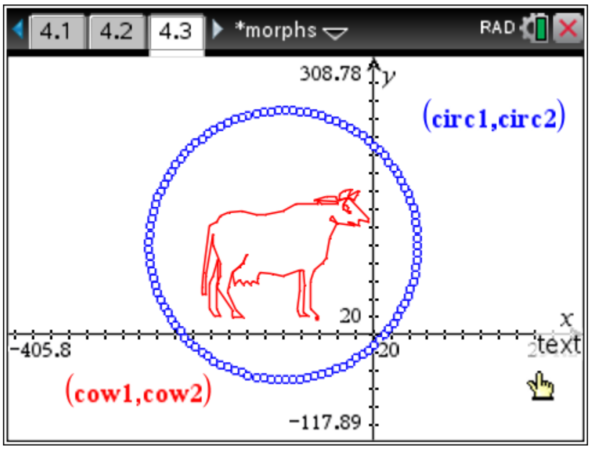
Now we can watch the transformation from a circle to a cow. We cannot animate the slider – what we can do with TI-Nspire.

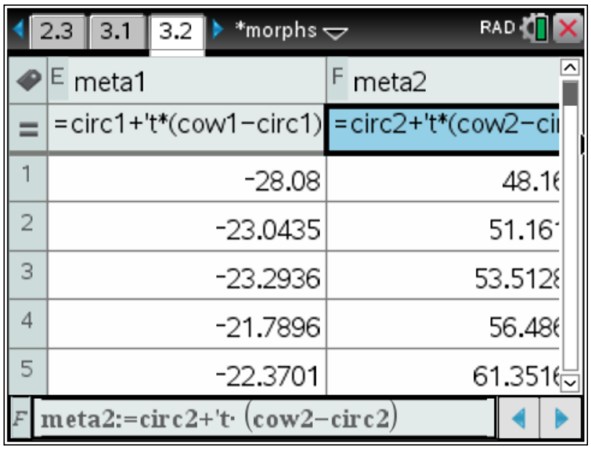
Circle and cow are defined by 113 points. It would be very boring copying 452 numbers one after the other from the DERIVE file or Excel-file to TI-Nspire.

Copy and paste supported by the computer makes it an easy work. See below how to transfer and create the list of coordinates for cow and circle. Unfortunatly we cannot program the metamorphosis because TI-NspireCAS does not support programming graphics. We put the lists into a Lists & Spreadsheet page in order to obtain two scatter diagrams for the two objects.







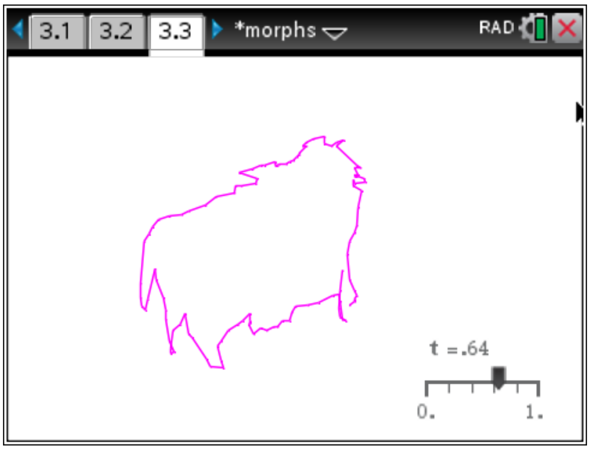


	A circ1	B circ2	C cow1	D cow2	E meta1	F meta2	G ccc1	H ccc2
=					=circ1+'t*(co	=circ2+'t*(cow	=(2*circ1−4*co	=(2*circ2−
1	50.	100.	-72	19	-43.94	37.63	-36.4248	42.6196
2	49.7682	108.336	-64	19	-37.8333	39.5473	-30.8252	45.0504
3	49.0734	116.647	-64	18	-37.9931	40.6887	-31.0278	46.7654
4	47.9179	124.906	-61	18	-35.9489	42.5883	-29.2395	49.1737
5	46.3052	133.088	-61	21	-36.3198	46.7802	-29.7098	53.6848
6	44.2402	141.167	-69	30	-42.9547	55.5685	-35.9791	62.4164
7	41.7294	149.12	-69	60	-43.5322	80.4976	-36.7113	85.9874
8	38.7806	156.921	-65	72	-41.1305	91.5317	-34.7376	96.7628
9	35.4028	164.545	-55	80	-34.2074	99.4454	-28.6386	104.653
10	31.6064	171.971	-52	85	-32.7705	105.003	-27.6204	110.361
11	27.4033	179.173	-47	107	-29.8872	123.6	-25.304	128.046
12	22.8063	186.131	-39	124	-24.7845	138.29	-20.9773	142.117
13	17.8298	192.823	-47	128	-32.0891	142.909	-28.0956	146.902
14	12.4891	199.228	-46	131	-32.5475	146.692	-28.9446	150.895
15	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

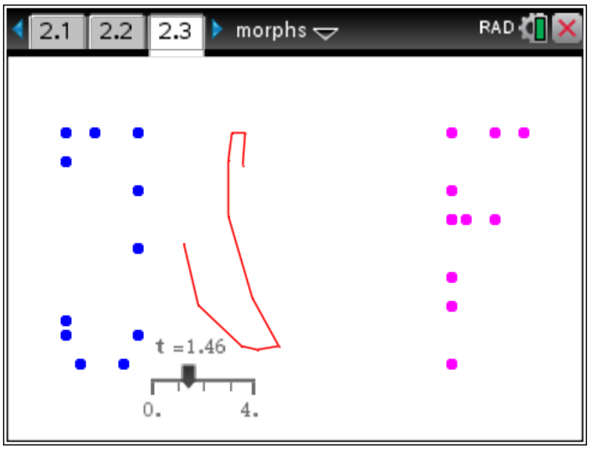
G ccc1:=(2· circ1−4· cow1+2· circ1)· 't^2+(−3· circ1+4· cow1−circ1)· 't+circ1

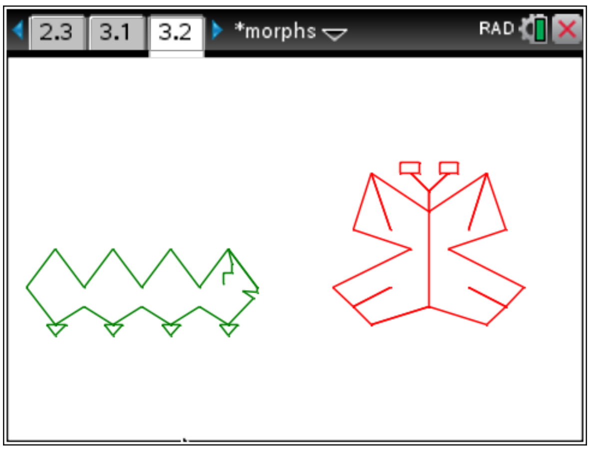
I wanted to produce an "infinite" animation circel – cow – circle – cow …. So, I applied the formula from page 32 on the three objects circle, cow and again cicle - and it worked (right screen shot).

Then I tried for five objects (with 0 ≤ t ≤ 4) and switched through the letters of "JOSEF".

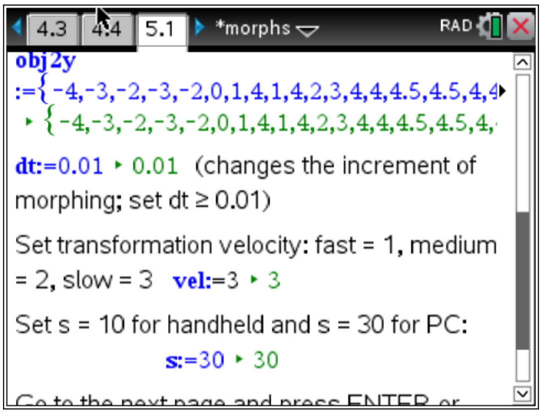


See finally a creation of my students from many years ago presenting the most wonderful metamorphis in nature.

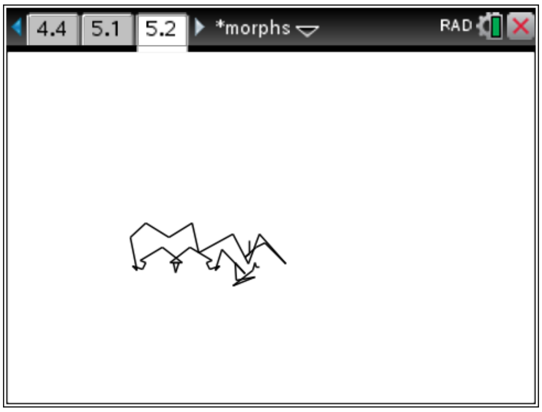




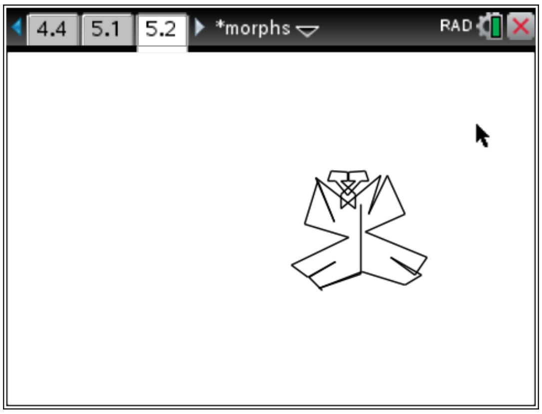
As I complained earlier it is not possible to program graphing with TI-NspireCAS. You can overcome this indeficieny by programming with LUA. So I had the ambition to replace the Nspire program by a LUA script. It needed some tries and a few emails to and from Steve Arnold and finally it worked. You can enter all necessary parameters including the lists of coordinates on a notes page and then immediately admire the transformation process. Much fun with MORPHING, Josef. (See also the screen shot on page 52.)




```
platform.apilevel='2.2'
local screen = platform.window
local w, h
local dt = var.recall("dt") or 0.1
local play = true
local t,n,dummy,x3,y3 = 0,0,0,{},{}

function on.construction()
  h=screen:height(); w=screen:width()
end

function on.resize()
  h = screen:height(); w = screen:width()
play = false; screen:invalidate();end
```




```
function on.enterKey()
  play = not play;   dt = var.recall("dt") or 0.1
  if play == true then timer.stop() else timer.start(dt) end
screen:invalidate();end

function on.mouseUp()
  play = not play;   dt = var.recall("dt") or 0.1
  if play == true then timer.stop() else timer.start(dt) end
screen:invalidate();end

function on.paint(gc)
     s=var.recall("s") or 30; vel=var.recall("vel") or 1
     gc:setColorRGB(0,0,0)
     for i=1,n-1 do
       gc:drawLine(w/2+s*x3[i],h/2-s*y3[i],w/2+s*x3[i+1],h/2-s*y3[i+1])
     end
     for j=1,(vel-1)*1000000 do
        dummy=dummy+1
     end;dummy=0
end
```




```
function on.timer()
    x1=var.recall("obj1x") or {};y1=var.recall("obj1y") or {}
    x2=var.recall("obj2x") or {};y2=var.recall("obj2y") or {}
    n = math.min(#x1,#x2,#y1,#y2)
    if t < 1 then
      t = t + dt
      for k = 1, n do
        x3[k] = x1[k] + t*(x2[k] - x1[k])
        y3[k] = y1[k] + t*(y2[k] - y1[k])
      end
      else t = 0 end
screen:invalidate();end
```

Titbits from Algebra and Number Theory(16)

by Johann Wiesenbauer, Vienna

This is going to be a very special column in this series. In fact, by courtesy of TI, the new owner of Soft Warehouse, it will be the very first publication containing programs written in the new powerful programming language of the forthcoming Derive for Windows 5. Although I had to use the current beta version of DfW5, there should be no essential differences to the final version, which should be out or at least on the way by the time you read these lines.

As you will see yourself before long, there will be a lot of great improvements in Derive 5, such as a permanent input line, a wrapping of long expressions on the screen, the option of embedding OLE-objects, a new spectacular 3D-graphics and many more. In fact, this could well be the most decisive change since the introduction of Derive back in 1988. In this article though I am going to deal only with some new features of Derive in the field of programming. Josef, the editor of this journal, told me to provide something like a programming course for beginners, and he urged me (on his knees!) to make the examples as simple as possible. Well, I'll do my best!

Let's start with a really simple problem from number theory. As you may know, every nonzero integer n can be represented uniquely as the product of an odd number, called the "odd part" of n, and a power of 2. How could a program look like that computes the odd part of a nonzero integer n? First the solution in DfW 4.11, which of course still works in DfW 5:

$$\text{odd\_part}(n) := \text{ITERATE}\left(\text{IF}\left(\text{MOD}(n\_, 2) = 1, n\_, \frac{n\_}{2}\right), n\_, n\right)$$ #

Here is the first rule: ITERATE-constructs are usually replaced by the more versatile LOOP-constructs in DfW 5, e.g.

```
odd_part(n) :=
  Loop
    If MOD(n, 2) = 1
       RETURN n
    n := n/2
```

As you can see, DERIVE-programs have got a nice structure now using several lines and indentations. This is the good news. The bad news is that currently you have to take care of commas and parentheses when typing the program in the input line. Thus in our example, the actual input is

ODD_PART(n):=loop(if(mod(n,2)=1,return(n)),n:=n/2)

I was told by Al Rich though, that a multi-line editor is in the cards, which will make the writing and editing of programs much more comfortable! For this reason and to make the listings more readable, I'll always give them only in the clearly arranged form here as they will appear in the algebra window.

We have seen above a way of exiting a loop my means of RETURN( ), which exits the function altogether and yields the argument of RETURN( ) as value of the function. Another way is to use the key word EXIT, which will cause the program to continue with the first program line after the loop in question or to stop and return the last computed value, if there is no such line left. (Note that due to the functional concept of Derive, LOOP( ) is also a function and its value is the value of the last expression that was computed before exiting the loop.)

As an example I would like to implement a probabilistic primality test, the so-called Rabin-Miller test, which is also used by Derive internally. If n >1 is an odd integer and a any integer in the range 0 < a < n, then n passes the Rabin-Miller test, if and only if the sequence

$$a^s, a^{2s}, a^{4s}, \ldots, a^{(n-1)/2},$$

where s is the odd part of n-1, starts with ±1 mod n or contains –1 mod n. Although there are many examples, where composite numbers pass this test, e.g. n=2047 for a=2, according to a celebrated theorem by Rabin-Monier, the probability that a composite number n will pass the Rabin-Miller test for k bases chosen at random in the range above is less than 4^{-k}.

In the following implementation of this test, first the odd part s_ of n – 1 is computed. Next we compute $a^{s\_}$ mod n using the built-in function MODS(x,m), which computes the least absolute remainder of x mod m. By taking the negative absolute value, we change the change the sign of MODS($a^{s\_}$,n), in case it is positive and the result becomes our new a. (Note that in view of the subsequent squarings, a change of sign does not matter!) Using this little trick, we only have to check the condition, whether the sequence above contains –1 mod n.

```
rabin_miller(n, a, s_) :=
  Prog
    s_ := n - 1
    Loop
      s_ :/ 2
      If MOD(s_, 2) = 1 exit
    a := - ABS(MODS(a^s_, n))
    Loop
      If a = -1
         RETURN true
      s_ :* 2
      If s_ = n - 1
         RETURN false
      a := MODS(a^2, n)
```

Also new in Derive 5 are the commands s_:/2 and s_:*2, which stand for halfing and doubling s_ and are simply abbreviations of s_:=s_/2 and s_:=s_*2, respectively. Furthermore, we took advantage of the fact that RETURNs (as well as EXITs) can occur more than once and in any place of a loop.

It might be a good idea – take this as a proposal – to underscore auxiliary variables like s_ above in order to distinguish them from “genuine” variables, which are used in the call of the function. It goes without saying that the number of these auxiliary variables should be as small as possible. (I would go so far as to say that one can tell a proficient programmer by the small number of auxiliary variables he uses!)

Although the routine RABIN_MILLER(n,a) above is very fast, e.g. the computation

```
              100
rabin_miller(10    + 267, 2) = true
```

takes only 0.01s on my Pentium 450 PC, it usually pays off to do some trial divisions by all primes below a certain bound B before applying it. At the very least, this must be done for the prime p=2, as n is supposed to be odd in the Rabin-Miller test. Furthermore, the exception n=1 must be treated beforehand.

```
pprime?(n, k := 6, d_ := 2, s_, t_ := 5) :=
  Prog
    If GCD(n, 6) > 1
       RETURN MEMBER?(n, [2, 3])
    s_ := MIN(FLOOR(√n), 1021)
    Loop
      If t_ > s_ exit
      If MOD(n, t_) = 0
         RETURN false
      t_ :+ d_
      d_ := 6 - d_
    If n < 1062961
       RETURN SOLVE(n > 1)
    RANDOM(-n)
    Loop
      If k = 0
         RETURN true
      If ¬ rabin_miller(n, RANDOM(n - 3) + 2)
         RETURN false
      k :- 1
```

#

In the routine PPRIME?( ) above, which emulates to a certain extain the built-in primality test of Derive before version 5 (cf. [2]) we first check, whether n is divisible by 2 or 3, and then trial-divide by all numbers that are not divisible by 2 or 3 up to 1021 or the square-root of n, whichever comes first. If n "survives" these trial divisions and is not smaller than 1062961 ($=1031^2$), then up to k Rabin-Miller tests are carried out (k= 6 by default). Note that before selecting the random bases a in the range 0 < a < n (excluding a = 1 and a = n - 1 for obvious reasons) the random number generator is initialized by computing RANDOM(-n) in order to make sure that every run of the program for fixed n and k yields the same result. Our routine is astonishingly fast, e.g. the computation

```
            100
pprime?(10     + 267) = true
```

takes only 0.12s on my PC (vs. 0.05s with the built-in function PRIME?( ), which is a lot safer though by combining a number of Rabin-Miller tests for n with a so-called Lucas test).

Let's revisit now the topic of the very first column in this series (cf. DNL #13,27-29). It dealt with the Euclidean algorithm which is used to determine the gcd(a,b) for two integers a and b. It has been called by D. Knuth the "granddaddy of all algorithms, because it is the oldest algorithm that has survived to the present day" (cf. [1]). I used a version of Derive then - I think it was Derive for DOS 2.56 - that has become incredibly obsolete by now. What has changed in the meantime?

Well, in the first place we could use the new operator ' to generate a nice table consisting of the equations that appear in the Euclidean algorithm for the integers a and b . (Because of gcd(0,0)=0, gcd(a,b)=gcd(b,a) and gcd(a,b)=gcd($|a|,|b|$) , we may assume w.l.o.g. that $a \ge b > 0$ in the following.)

```
ea(a, b, q_, r_, s_ := []) :=
  Loop
    If b = 0
       RETURN s_
    [q_ := FLOOR(a, b), r_ := a - q_·b]
    If r_ > 0
       s_ := INSERT('[a = q_·b + r_], s_, 0)
       s_ := INSERT('[a = q_·b], s_, 0)
    [a := b, b := r_]
```

And this is a small example that shows the appearance of a function call on the screen:

$$\text{ea}(8991,\ 3293) = \begin{bmatrix} 8991 = 2\cdot 3293 + 2405 \\ 3293 = 2405 + 888 \\ 2405 = 2\cdot 888 + 629 \\ 888 = 629 + 259 \\ 629 = 2\cdot 259 + 111 \\ 259 = 2\cdot 111 + 37 \\ 111 = 3\cdot 37 \end{bmatrix}$$

What might strike you is the fact that the quotients in this example are relatively small. According to [1] the probabilty P(q) that a quotient q occurs in one of the equation for two random numbers a and b should be given by the formula

$$\mathrm{P}(q) = \lg \frac{(q+1)^2}{(q+1)^2 - 1} ,$$

where lg x denotes the logarithm for the base 2. In particular, we get P(1)=0.41504..., P(2)=0.16992..., P(3)=0.09311..., P(4)=0.05890..., which means that the probability of small quotients is very high, indeed!

What about a program that does some kind of statistics for us as far as the quotients and their absolute and relative frequencies are concerned?

```
ea_statistics(l, a_, as_, n_, p_, ps_, q_, qs_, r_, s_) :=
  Prog
    n_ := DIM(l)
    s_ := REVERSE(TERMS(Σ(x_^l_, l_, l)))
    l := []
    as_ := 0
    Loop
      If s_ = []
         RETURN APPEND([["Quot.", "Abs.F.", "Σ", "Rel.F.", "Σ", "Prob.", "Σ"]], l)
      q_ := x_·∂(LN(s_↓1), x_)
      a_ := LIM(s_↓1, x_, 1)
      as_ :+ a_
      p_ := APPROX(LOG((q_ + 1)^2/((q_ + 1)^2 - 1), 2))
      ps_ := APPROX(LOG(∏((k_ + 1)^2/((k_ + 1)^2 - 1), k_, 1, q_), 2))
      l := APPEND(l, [[q_, a_, as_, a_/n_, as_/n_, p_, ps_]])
      s_ := DELETE(s_)

ea_quotients(a, b, l_ := [], r_) :=
  Loop
    If b = 0
       RETURN ea_statistics(l_)
    l_ := APPEND(l_, [FLOOR(a, b)])
    r_ := MOD(a, b)
    a := b
    b := r_

             40
(a := RANDOM(10   )) = a := 8353277490109198729488714160267139225088

             40
(b := RANDOM(10   )) = b := 4086189197847449071543091710137126824l
```

```
ea_quotients(a, b)
```

Quot.	Abs.F.	Σ	Rel.F.	Σ	Prob.	Σ
1	25	25	0.357142	0.357142	0.415037	0.415037
2	15	40	0.214285	0.571428	0.169925	0.584962
3	8	48	0.114285	0.685714	0.0931094	0.678071
4	3	51	0.0428571	0.728571	0.0588936	0.736965
5	3	54	0.0428571	0.771428	0.0406419	0.777607
6	1	55	0.0142857	0.785714	0.0297473	0.807354
7	4	59	0.0571428	0.842857	0.02272	0.830074
10	1	60	0.0142857	0.857142	0.0119726	0.874469
11	1	61	0.0142857	0.871428	0.0100536	0.884522
12	2	63	0.0285714	0.9	0.00856201	0.893084
13	1	64	0.0142857	0.914285	0.00737953	0.900464
23	1	65	0.0142857	0.928571	0.00250685	0.941106
32	1	66	0.0142857	0.942857	0.00132539	0.956931
47	1	67	0.0142857	0.957142	0.000626305	0.970252
204	1	68	0.0142857	0.971428	0.0000343298	0.992979
284	1	69	0.0142857	0.985714	0.0000177618	0.994946
289	1	70	0.0142857	1	0.0000171546	0.995033

# #

In the table above you can see for two random numbers a and b with about 40 digits and a>b for all occurring quotients their absolute frequencies, their relative frequencies and their probabilities according to the formula above. Each of these values is also cumulated in a column of its own that is always to the right of the column it refers to.

I had to choose the numbers a and b relatively small so that the resulting table would fit this page. The coincidence between the actual relative frequencies and the corresponding probabilties becomes even more impressive if you choose larger numbers, say with several hundred digits. (Check it!)

Just a few words to the programming techniques used in the programs above. The first argument L of EA_STATISTICS( ) is the list of quotients provided by EA_QUOTIENTS( ). For each of the quotients in this list L we have to determine the number of occurrences. To this end, we form the polynomial

$$\sum_{q \in L} x^q$$

and take advantage of the fact that Derive will internally combine all powers x^q for a fixed q into one expression of the form ax^q, where a denotes the number of occurrences of q in the list L. Moreover, these expressions are sorted by descending values of q. Hence, the first assignment of s_ in EA_STATISTICS( ), namely

```
                     l_
s_ := REVERSE(TERMS(∑(x_  , l_, 1)))
```

yields for s_ a list of expressions of the form ax^q, where q runs through all different quotients in L in ascending order. Now we have come across a small problem: How do we get the a's and q's from the terms ax^q? First try to find this out yourself, then look at the programm listing and compare your solution with the solution given in the lines where a_ and q_ are assigned! You got it? Great! The rest should be easy.

Taking a look at the resulting table again, the last number in the third column, which is 77 here, tells us how many divisions in the Euclidean algorithm were necessary. But what is its expected value? It is

$$\frac{12\ln 2}{\pi^2}\ln 10^{40} + 0.14 \approx 77.76$$

(cf. [1]) and therefore surprisingly close to the actual value.

You might wonder how the ubiquitious π comes into play in the formula above. Well, there is not enough space to go into details, but it is mainly because of the well-known formula

$$\sum_{k=1}^{\infty}\frac{1}{k^2}=\frac{\pi^2}{6}$$

By the way, maybe the most elementary way to prove this formula is by computing the double integral

$$\int_0^1\int_0^1 \frac{dx\ \ dy}{1-xy}$$

For one thing, the value of this integral should be the sum on the left hand side of the formula (you can see this easily by expanding the integrand into an infinite geometric series with quotient q=xy and integrating this series term by term!), for another this integral can be computed directly by means of clever substition (try it!) yielding the right hand side of the formula. Can DERIVE cope with this integral? You bet! (Check it!) I remember how I tried to compute this integral in the old days of Derive 2.xx to no avail. At some time since then, it … no, SHE (as I was told by Al Rich himself!) has learnt how to do this and many, many nice things more!

References

[1] D.E.Knuth, The Art of Computer Programming, Vol 2: Seminumerical Algorithms, 3rd ed., Addison-Wesley, Reading, Mass.,1998

[2] A.D.Rich and D.R.Stoutemyer, Inside the DERIVE Computer Algebra System, The International DERIVE Journal, Vol.1 No.1,3-17.

Experiments Using CBL/CBR and the TI-92 – New Conceptions for Teaching Natural Science

Heinz-Dieter Hinkelmann, Korneuburg, Austria

1. Introduction

With this title a new series had been started to enlarge the possibilities of application of the TI-92 under the aspect of natuaral science. For that means experiments based on CBL™ (Computer Based Laboratory) and CBR™ (Computer Based Ranger), both products of *Texas Instruments*, are described as well as the measuring sensors of CBL-basic equipment and the sensors and probes of the *Vernier* company.

We will show one or two experiments for the natural science lessons and their mathematical analysis. If anybody has more ideas, suggestions or questions, just please contact me by using the following adress:

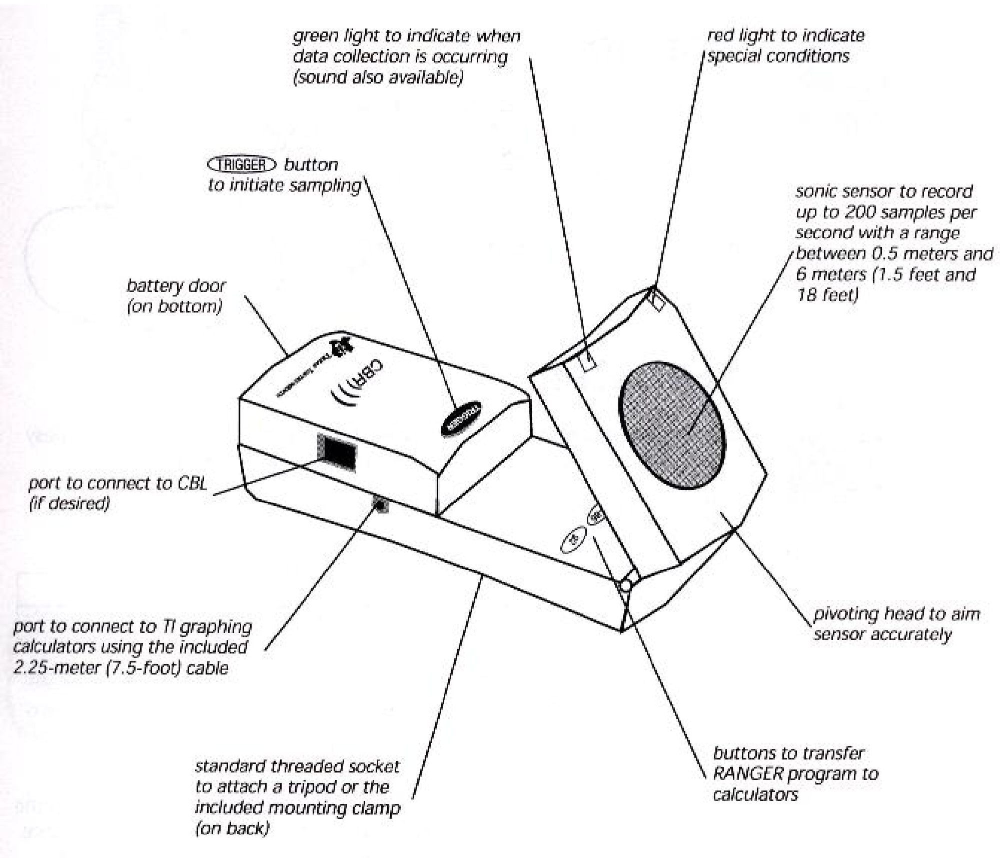
heinz-dieter.hinkelmann@telecom.at

The measuring instruments will be shortly described for those, who don´t know them. Today´s edition describes CBR and the following editions of DNL will explain the sensors and probes.

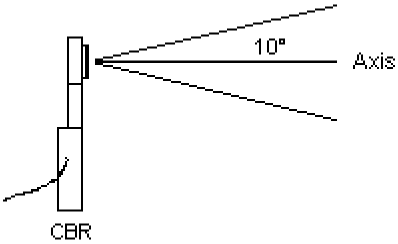
2. How does the CBR work?

The picture below describes the features of the CBR[1]:

#



The sonic motion detector sends out an ultrasonic pulse, which returns after bouncing off an object. The wave fills a cone-shaped space with an opening angle of 20°. The motion detector then receives the reflected pulse. Using the time interval between the emitted and reflected pulses and the speed of sound the CBR determines how far the object is away. Then the CBR computes the first and second



derivatives of the distance data, with respect to time, in order to obtain the velocity and acceleration data. The time, distance, velocity and acceleration data are then stored in the calculator as list variables and can be processed automatically with the help of the RANGER program.

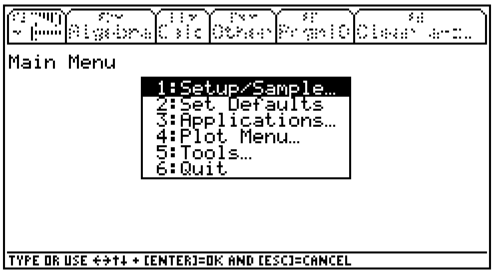
The sonic motion detector calculates the distance from the closest object that produces a sufficiently strong echo.

The sensitivity of the echo detection automatically increases, in steps, every few milliseconds as the ultrasound travels outward. This is to allow for echoes being weaker from more distant objects. However, the range of the CBR is limited to a distance of between 0.5 and 6 meters (between 1.5 and 18 feet) as the echo produced by an object that is farther away is usually too weak to be reliably measured.

3. Start and setups

The CBR connects directly to a TI-calculator (the TI-82, TI-83, TI-85, TI-86 and the TI-92) using the black cable. The CBR contains a set of programs (RANGER) that allows communication between the calculators and the CBR. These programs are downloaded to the calculators from the CBR by pressing the appropriate button on the CBR. RANGER also includes programs MATCH and BALL BOUNCE. The programs for the TI-89 are not included within the CBR and hence must be downloaded to the calculator from a personal computer by using the TI-Graph Link™. This version of RANGER you will find on the TI-website. For downloading the file RANGER.xxP there will need at least 10 KBytes of free memory on your TI-calculator.

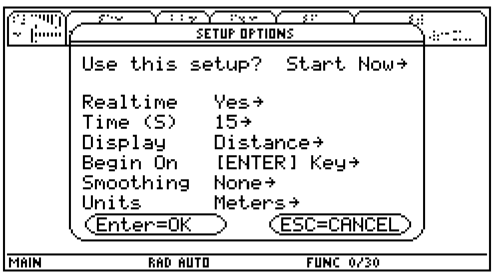
If you enter the program code, **ranger()**, the setup screen is displayed. By pressing <Enter> the "MAIN MENU" together with some sub-menus is shown. This menu provides the following options:



1: View and change the set-up options before data collection
2: Reverts to the original standard set-up
3: List of applications, e.g. "Match" & "Ball Bounce"
4: Graph set-up options
5: Auxiliary programs for data transfer
6: Exit the RANGER program

When you choose the "SETUP OPTIONS" the default-settings will appear:

The following options for changing the setup are provided:



Realtime	Yes	No		
Time (S)				
Display	Distance	Velocity	Acceleration	
Begin On	[Enter] Key	[Trigger] Button	10 Sec. Delay	
Smoothing	Light	Medium	Heavy	None
Units	Feet	Meters		

4. Two Selected Activities

4.1 Activity 1: Match the graph

This activity helps the student understand partially defined linear functions. The program uses a "Distance versus Time" concept consisting of three combined linear functions. By moving the CBR to and fro horizontally, whilst directed towards a wall, a student attempts to physically match the graph plotted by the TI-92.

After the students have mastered this task, we then set them the more difficult task involving "Velocity versus Time" graphs.

The program randomly generates the graphs.

Equipment required: Calculator, CBR, connecting cable, masking tape, meter stick, marker pens

Setup procedure

We connect the CBR to the TI-92 with the connecting cable.

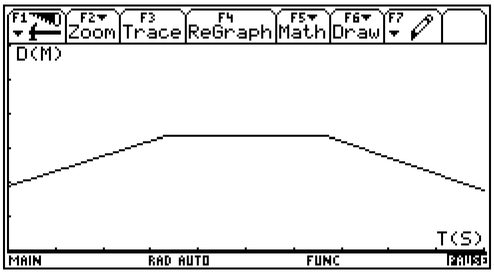
The student conducting the test holds the CBR, which is aimed directly at a wall, in one hand, and the calculator in the other. As an alternative the CBR could also be positioned on a table at chest height and pointed directly at the student. We mark the floor in front of the wall (or table) at distances of 50 cms (up to a total distance of 4 m).

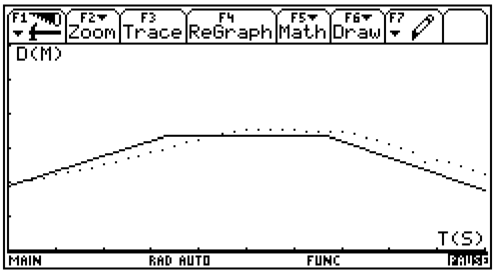
We run the RANGER program on the TI-calculator. From the "MAIN MENU" we choose "3:Applications..." and then "2:Meters". The "Match" programs are not affected by other setup variations.

Distance-Time match

To examine the distance-time-concept now we choose "1:Distance Match", again from the "Applications" menu and the start window will open. After pressing the <Enter> key a graph is displayed. The student then ascertains from the graph: where she should start; and how she should move.

A typical graph may look like one given below:



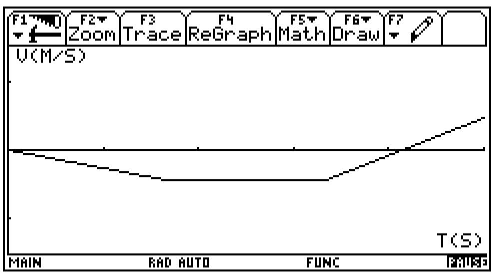


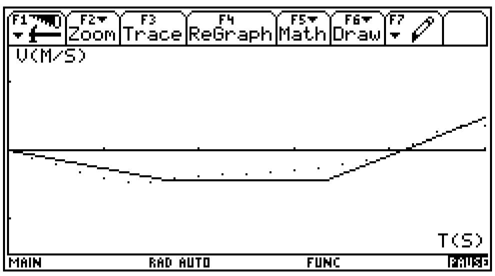
The student should move away from the wall from 1.9 m to 3.5 m in 3.2 s, remain in this position for approximately 3.5 s and then move towards the wall up to a distance of 1.8 m in the following 3.3 s.

Sampling is started in "Realtime" mode by pressing the <Enter> key. The dotted line, which shows the movements of the tester, is superimposed over the previously plotted graph.

Velocity-Time match

In the same way we can choose "2:Velocity Match" from the "Applications" menu and do the experiment. It is, of course, much more difficult to match the velocity-time graph. A typical match could look like the following:





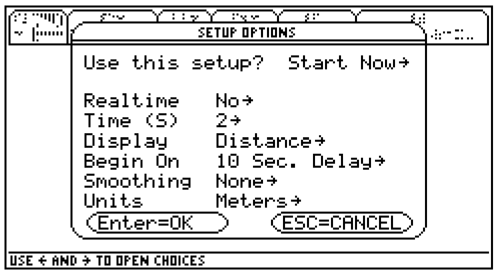
4.2 Activity 2: Free-falling objects

If objects are dropped from a certain height in the atmosphere down onto the earth, air resistance and friction are at work. Light objects with greater air resistance will move towards the earth with constant speed. A heavier object, that has less air resistance, will move towards the ground with almost constant acceleration. These facts can be demonstrated using a cone shaped paper cup and a light ball (which can bounce on the CBR without damaging it).

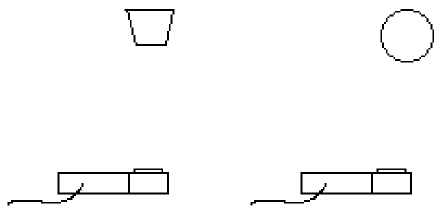
Equipment required: Calculator, CBR, connecting cable, ball (light), paper cup (cone shaped)

Setup procedure

We connect the CBR with TI-92, run the RANGER program on the TI-calculator and choose the setup on the right ("10 Sec. Delay" will delay data collection for 10 seconds after starting up the CBR).

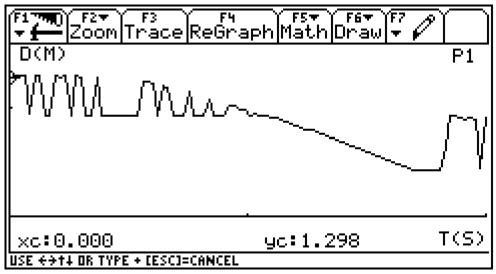


In this experiment the CBR is placed on the ground and a paper cup is dropped onto it. Then, in the second test, a light ball (e.g. table tennis ball) is dropped onto the CBR.



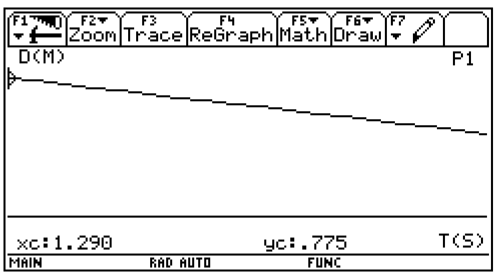
Analysis of the data obtained with the paper cup

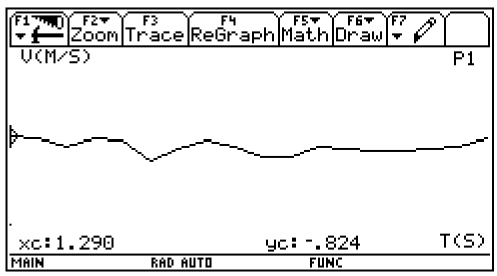
As soon as the data collection has been completed, the distance-time graph of the cup's fall is displayed.



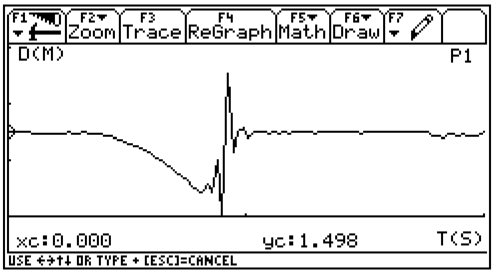
It is possible to change the graph so that only the relevant parts can be seen.

After selecting the left and the right bound the new distance-time graph will be displayed. After pressing the <Enter> key in the follwing menu you can choose "2:Velocity-Time", which initialises the velocity-time graph.

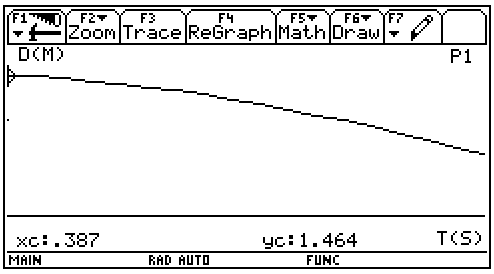


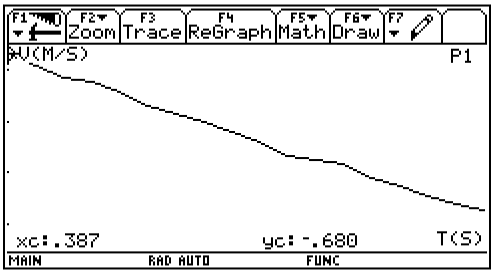


Air resistance and friction of the cup (which is very light and has a large surface area) prevents constant velocity.



As with the paper cup experiment, we display only the relevant domain of the graph.





Velocity of the ball is negative, but increasing in magnitude as the ball moves towards the CBR. From the graph we can deduce that the acceleration appears to be almost constant. Uniform acceleration occurs if the ball offers very little air resistance. In fact, the ball offers hardly any air resistance.

DERIVE Version 5??

One question for many others which are very similar and the answer:

Hello Everyone!

I've been waiting for the release of DERIVE ver 5 since last year. I was told it's coming out soon. But, I still have no news of it.

Can the DERIVE creators brief us on this? The others may also be interested on its new/upcoming features.

I got the feeling that DERIVE will no longer be sold as an independent software but will only be part of TI calculator's CAS component. Correct me if I'm mistaken.

Thanks.

Dear all,

I am happy to report that Derive 5 should be available by the end of this month (March). The new features in version 5 will be described on the Derive web site at www.derive.com when the software is released. Texas Instruments will continue to sell Derive as a stand-alone software product for MS Windows. It is my understanding that TI intends to distribute it on a CD ROM.

I think Derive fans will find the many new features and capabilities of Derive 5 well worth the wait. Thank you for your patience.

Aloha,
Albert D. Rich
Co-author of Derive

ORDINARY DIFFERENTIAL EQUATIONS WITH CONSTANT COEFFICIENTS

Francisco M. Fernández, CEQUINOR (CONICET), Facultad de Ciencias Exactas, UNLP, Calle 47 y 115, Casilla de Correo 962, 1900 La Plata, Argentina. marcelo@diquite.edu.ar / framfer@isis.unlp.edu.ar

1. Introduction

In DNL#33 Candel and Basauri proposed an interesting algorithm for the solution of ordinary homogeneous differential equations of order n with constant coefficients. They argued that such solutions could then be used to solve inhomogeneous equations. Here I show an alternative procedure that applies directly to inhomogeneous equations. In my opinion the present approach offers some advantages: first, it appears to be simpler, second, it gives the general solution of the homogeneous equation as a particular case of the inhomogeneous one, third, it bypasses the explicit intermediate construction of the linearly independent solutions of the homogeneous equation. My approach is recursive, and, thereby suitable for computer algebra.

2. Ordinary Differential Equation

Following Candel and Basauri's notation I write the differential equation as

$$\hat{P}Y(x) = f(x), \tag{1}$$

where $\hat{P}$ is the differential operator with constant coefficients a_j

$$\hat{P} = P(\hat{D}) = \sum_{j=0}^{n} a_j \hat{D}^j , \tag{2}$$

and $\hat{D} = \dfrac{d}{dx}$. Without loss of generality we choose $a_n = 1$. In order to solve equation (1) one first obtains the roots $\{r_1, r_2, \ldots, r_n\}$ of the characteristic polynomial

$$P(r) = 0 \tag{3}$$

and factorizes the operator $P(\hat{D})$ as follows:

$$P(\hat{D}) = \prod_{j=1}^{n} (\hat{D} - r_j). \tag{4}$$

Next I define the set of functions

$$Y_0(x) = Y(x), Y_s(x) = \prod_{j=1}^{s} (\hat{D} - r_j) Y(x), s = 1,2,\ldots,n, \tag{5}$$

so that $Y_n(x) = f(x)$. On solving the first-order differential equation

$$Y_{s+1}(x) = (D - r_{s+1}) Y_s(x) \tag{6}$$

for $Y_s(x)$ one obtains

$$Y_s(x) = \exp(r_{s+1}x)(c_{s+1} + \int_{x_0}^{x} \exp(-r_{s+1}u)Y_{s+1}(u)du), \quad (7)$$

where c_{s+1} is an arbitrary integration constant. If one applies equation (7) recursively for s = n–1, n–2, …, 0, then the general solution of equation (1) appears at the last step as $Y_0(x)$ which depends on n integration constants $c_1, c_2, \ldots c_n$, and the lower integration limit x_0. For concreteness I choose x_0 from now on.

3. Program for the Solution of the Differential Equation (1)

The program for the solution of equation (7) reduces to one recursive function:

```
[CaseMode := Sensitive, InputMode := Word]

Y(r, s, f) :=
  If s = DIM(r)
     f
     EXP(r↓(s + 1)·x)·(c↓(s + 1) + ∫(LIM(EXP(- r↓(s + 1)·x)·Y(r, s + 1, f), x, u), u, 0, x))
```

The arguments are: the vector $r = [r_1, r_2, \ldots, r_n]$ with *all the roots* of the characteristic polynomial (3), $s = 0$, and the inhomogeneous term $f(x)$. Notice that repeated roots should be entered explicitly in r. If desired, one obtains intermediate solutions by simply setting $0 < s < n$. Those who prefer the use of ITERATE instead of LIM in functions like $Y(r, s, f)$ may substitute

```
Y2(r, s, f) :=
  If s = DIM(r)
     f
     EXP(r↓(s + 1)·x)·(c↓(s + 1) + ∫(ITERATE(EXP(- r↓(s + 1)·x)·Y(r, s + 1, f), x, u, 1), u, 0, x))
```

for $Y(r, s, f)$ above.

As an illustrative example consider the differential equation discussed by Candel and Basauri:

$$y'''(x) + 3y''(x) - 4y'(x) - 12y(x) = 0, \quad (8)$$

which in Derive form reads

DIF(yy,x,3)+3*DIF(yy,x,2)-4*DIF(yy,x)-12*yy and appears on the DERIVE 6 screen as

$$\left(\frac{d}{dx}\right)^3 yy + 3\cdot\left(\frac{d}{dx}\right)^2 yy - 4\cdot\frac{d}{dx} yy - 12\cdot yy$$

Using Derive SOLUTIONS command one obtains the roots of the characteristic equation

$$\text{SOLUTIONS}(r^3 + 3\cdot r^2 - 4\cdot r - 12, r) = [2, -2, -3]$$

Application of equation (7)

```
yy1 := Y([2, -2, -3], 0, 0)
```

yields

$$yy1 := \frac{e^{2\cdot x}\cdot(c_3 + 5\cdot c_2 + 20\cdot c_1)}{20} - \frac{e^{-2\cdot x}\cdot(c_3 + c_2)}{4} + \frac{e^{-3\cdot x}\cdot c_3}{5}$$

which differs from the one obtained by Candel and Basauri just in the way of collecting the independent solutions. In order to verify this result substitute it into the differential equation and simplify:

$$\left(\frac{d}{dx}\right)^3 yy1 + 3\cdot\left(\frac{d}{dx}\right)^2 yy1 - 4\cdot\frac{d}{dx} yy1 - 12\cdot yy1 = 0$$

Next I consider an inhomogeneous version of Candel and Basauri's equation arbitrarily choosing $f(x) = \cos(x)$. Applying the algorithm above

```
yy2 := Y([2, -2, -3], 0, COS(x))
```

one obtains

$$yy2 := \frac{e^{2\cdot x}\cdot(5\cdot c_3 + 25\cdot c_2 + 100\cdot c_1 + 2)}{100} - \frac{e^{-2\cdot x}\cdot(5\cdot c_3 + 5\cdot c_2 - 2)}{20} + \frac{e^{-3\cdot x}\cdot(10\cdot c_3 - 3)}{50} - \frac{3\cdot COS(x)}{50} - \frac{SIN(x)}{50}$$

Again, to verify the result substitute it into the differential equation

$$\left(\frac{d}{dx}\right)^3 yy2 + 3\cdot\left(\frac{d}{dx}\right)^2 yy2 - 4\cdot\frac{d}{dx} yy2 - 12\cdot yy2 = COS(x)$$

Just let me try just a little more complex example:

$$y^{IV}(x) - 2y'''(x) + 2y''(x) - 2y'(x) + y(x) = sin(x)$$

Solutions of the characteristic equation:

$$SOLUTIONS(r^4 - 2\cdot r^3 + 2\cdot r^2 - 2\cdot r + 1,\ r) = [1,\ i,\ -i]$$

but:

$$FACTOR(r^4 - 2\cdot r^3 + 2\cdot r^2 - 2\cdot r + 1) = (r - 1)^2\cdot(r^2 + 1)$$

shows that the roots are $r = [1, 1, -i, i]$; therefore I obtain the solution as

$$yy3 := \frac{e^{x}\cdot(2\cdot(x - 1)\cdot c_4 + 2\cdot x\cdot c_3 + 4\cdot x\cdot c_2 + 4\cdot c_1 + x - 2)}{4} + \frac{(5\cdot x + 1)\cdot COS(3\cdot x)}{4000} - \frac{3\cdot(5\cdot x + 1)\cdot SIN(3\cdot x)}{4000} - \frac{(5\cdot x + 1)\cdot COS(x)^3}{1000} + \frac{3\cdot(5\cdot x + 1)\cdot SIN(x)\cdot COS(x)^2}{1000} + \frac{COS(x)\cdot(2000\cdot c_4 + 15\cdot x + 2003)}{4000} - \frac{SIN(x)\cdot(2000\cdot c_3 - 985\cdot x - 997)}{4000} + i\cdot\left(e^{x}\cdot\left(\frac{x}{2} - \frac{1}{2}\right)\cdot c_3 - \frac{COS(5\cdot x)}{86528} + \frac{5\cdot SIN(5\cdot x)}{86528} - COS(3\cdot x)\cdot\left(\frac{c_4}{400} + \frac{9}{12800}\right) + SIN(3\cdot x)\cdot\left(\frac{3\cdot c_4}{400} + \frac{27}{12800}\right) + \frac{COS(x)^5}{5408} - \frac{5\cdot SIN(x)\cdot COS(x)^4}{5408} + COS(x)^3\cdot\left(\frac{c_4}{100} + \frac{349}{135200}\right) - SIN(x)\cdot COS(x)^2\cdot\left(\frac{3\cdot c_4}{100} + \frac{1047}{135200}\right) - COS(x)\cdot\left(\frac{3\cdot c_4}{400} - \frac{c_3}{2} + \frac{2219}{1081600}\right) + SIN(x)\cdot\left(\frac{3\cdot c_4}{400} + \frac{2219}{1081600}\right)\right)$$

This looks really very strange and complicated. We trust on DERIVE's art manipulating with trig expressions and try:

```
Trigonometry := Expand

Y([1, 1, i, -i], 0, SIN(x))
```

$$e^{x}\cdot\left(\left(\frac{x}{2}-\frac{1}{2}\right)\cdot c_{4}+\frac{x\cdot c_{3}}{2}+x\cdot c_{2}+c_{1}+\frac{x}{4}-\frac{1}{2}\right)+\mathrm{COS}(x)\cdot\left(\frac{c_{4}}{2}+\frac{1}{2}\right)$$

$$-\frac{\mathrm{SIN}(x)\cdot(2\cdot c_{3}-x-1)}{4}+i\cdot\left(e^{x}\cdot\left(\frac{x}{2}-\frac{1}{2}\right)\cdot c_{3}+\frac{c_{3}\cdot\mathrm{COS}(x)}{2}\right)$$

This looks quite better.

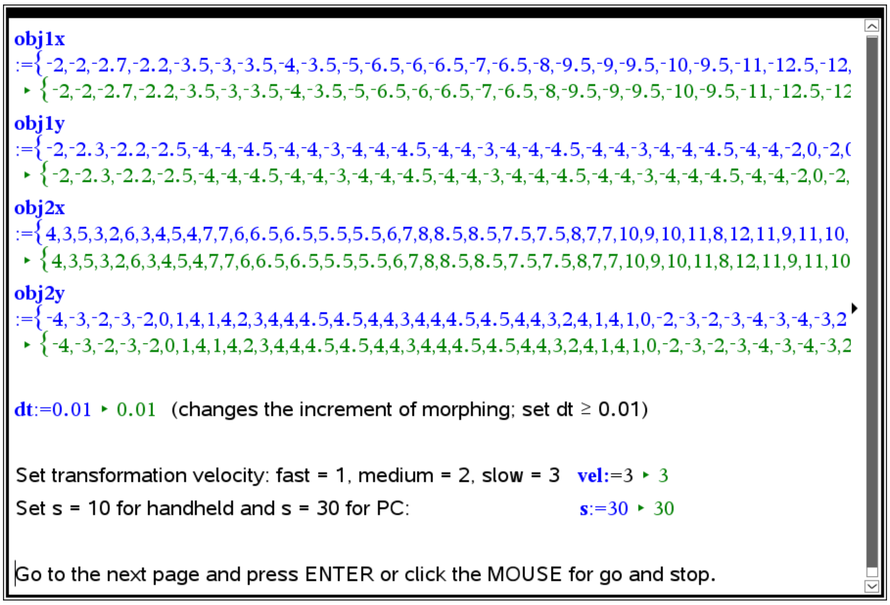
Then verify the result substituting it back into the differential equation:

$$\left(\frac{d}{dx}\right)^{4} yy3-2\cdot\left(\frac{d}{dx}\right)^{3} yy3+2\cdot\left(\frac{d}{dx}\right)^{2} yy3-2\cdot\frac{d}{dx} yy3+yy3=\mathrm{SIN}(x)$$

The algorithm seems to work properly.

I wanted to supplement the function just described with an efficient algorithm for the calculation of the vector r. The one I developed was based on Derive SOLVE command, following the lines of Candel and Basauri. However, it resulted to be rather slow because I am not a good programmer and my program carried out certain calculations several times unnecessarily. Moreover, it was unsuitable for the case that Derive was unable to obtain exact roots. Therefore, I prefer to give only the algorithm for the systematic solution of the differential equation leaving further improvements and additions to more proficient Derivers who may be interested in the present method. I think that the algorithm is simple and elegant.

See here a screenshot of the notes page of the LUA-supported morphing decribed on page 37.



At the 12th ICTCM in San Francisco David Bowers gave a workshop on the use of a very special feature of the TI-92/89. He published his workshop on his web site

http://ourworld.compuserve.com/homepages/davidbowers/Sanfra99/Spread2.htm.(not valid 2017)

He gave the permission to present a selection of his innovating activities in the DNL. Many thanks for that. As I am convinced that in many cases not the special technology but the idea behind is the important fact, I add how I tried to transfer his ideas to DERIVE - keeping in mind, that David did this in a similar way in a lecture held at the DERIVE-Conference in Bonn 1996(*). Josef

Algebraic Spreadsheets - the Best of two Worlds?

David Bowers, Ipswich, UK

Spreadsheets have long been a useful tool for analysing problems numerically. More recently, computer algebra systems have hit the scene. We demonstrate how the TI-92/89 goes some way to incorporate computer algebra into a simple spreadsheet environment. Examples of some unexpected applications will be investigated.

Activity 4: Investigating surds

The rules for manipulating surd quantities are often taken on trust. CAS allow arithmetics to be done exactly, and answers can be given in simplified surd form. Combining this with the tabular approach of a spreadsheet, the student can generate a variety of examples quickly, and investigate the application of the underlying rules.

We focus here on the result $\sqrt{m} * \sqrt{n} = \sqrt{m\,n}$ where $\sqrt{m\,n}$ may simplify further.

Start a new datasheet called surds in your special folder.

We will make use here of the function rand(n), which generates a random integer between 1 and *n*. Therefore a random integer between 5 and 8 inclusive will be given by rand(4)+4, and a random integer between 10 and 15 inclusive will be given by rand(6)+9. These two sets of numbers will generate appropriate data for our investigation.

Firstly, set the 3 to Exact. Then:

c1 = seq(rand(4) + 4, i, 1, 20)
c2 = seq(rand(6) + 9, i, 1, 20)
c3 = √(c1)
c4 = √(c2)
c5 = c3 * c4

F1 | F2 Plot Setup | F3 Cell | F4 Header | F5 Calc | F6 Util | F7 Stat

DATA	c2	c3	c4	c5
1	13	√(6)	√(13)	√(78)
2	13	√(5)	√(13)	√(65)
3	13	√(6)	√(13)	√(78)
4	14	√(5)	√(14)	√(70)
5	12	√(7)	2*√(3)	2*√(21)
6	10	√(6)	√(10)	2*√(15)
7	10	√(5)	√(10)	5*√(2)

c5=√(c1*c2)

MAIN RAD EXACT FUNC

Scrolling down the lists, the student should justify the values in column c5 on the basis of the values in cols c3 and c4. Sometimes it is "obvious"- as in rows 1 and 2 in the screendump - but sometimes it needs further explanation - as in rows 5 - 7. *Note that a fresh screen of data is automatically generated if you highlight and enter any column header (which re-invokes the rand( ) function).*

The connection with the factorisation of the product *m n* can be made clear by adding

c6 = factor(c1*c2).

(*) http://rfdz.ph-noe.ac.at/acdca/konferenzen/bonn-1996.html

The student should be encouraged to produce a table of results similar to the one above, and to explain each of the values in column c5. As a self-test, the student could scroll back across to "hide" columns c5 and c6, and generate a new table and then work out what s/he thinks the values in c5 will be given. S/he could also change the argument of the rand( ) function to work with larger numbers to become more confident in his conjectures and work.

TASK: Logarithms also have underlying rules which students often find difficult to internalise.
For example: **log(a b) = log(a) + log(b)**.
Produce a spreadsheet to demonstrate these rules, which will allow students to investigate their properties.

The DERIVE implementation is a bit tricky because of simplification of *DERIVE´*s RANDOM function at each call in a statement. (see DNL #27). But Johann Wiesenbauer found a very easy way to overcome that problem. You have to fix the random numbers before using them for any further purpose.

```
"Activity 4 with Johann`s support:"

SURD(n) := VECTOR(APPEND(v, [√(v_1), √(v_2), √(v_1·v_2)]), v, VECTOR([RANDOM(4)
           + 4, RANDOM(6) + 9], i, 1, n))
```

$$\text{SURD}(3) = \begin{bmatrix} 4 & 13 & 2 & \sqrt{13} & 2\cdot\sqrt{13} \\ 5 & 10 & \sqrt{5} & \sqrt{10} & 5\cdot\sqrt{2} \\ 5 & 9 & \sqrt{5} & 3 & 3\cdot\sqrt{5} \end{bmatrix}$$

```
"My own attempt refering to DNL#27"

[z1 := RANDOM(4) + 4, z2 := RANDOM(6) + 9]

TEST(n) := (a :== VECTOR([z1, z2], i, n))

SURD2(v_) := VECTOR([v_i,1, v_i,2, √(v_i,1), √(v_i,2), √(v_i,1·v_i,2)],
             i, DIMENSION(v_))
```

$$\text{TEST}(3) = \begin{bmatrix} 4 & 12 \\ 7 & 9 \\ 7 & 12 \end{bmatrix}$$

$$\text{SURD2}(a) = \begin{bmatrix} 4 & 12 & 2 & 2\cdot\sqrt{3} & 4\cdot\sqrt{3} \\ 7 & 9 & \sqrt{7} & 3 & 3\cdot\sqrt{7} \\ 7 & 12 & \sqrt{7} & 2\cdot\sqrt{3} & 2\cdot\sqrt{21} \end{bmatrix}$$

Activity 6: Differentiation by first principles

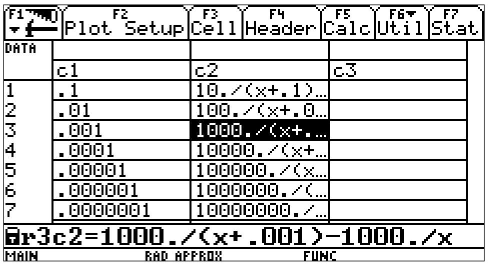
Spreadsheets allow the generation of numerical sequences which tend to a limit. This can be incorporated with the algebraic facilities of the TI-92/89 to allow a visualisation of the limiting behaviour of the gradient function.

Start a new datasheet called "diff1" in your folder.

We wish to consider the limit of $\frac{f(x+h) - f(x)}{h}$ as h tends to zero.

In column c1 generate a sequence of values for *h*: c1 = seq(0.1^i,i,1,10). Start off by considering the functions $f(x) = x^2$ and $f(x) = x^3$. We will show here the process for $f(x) = \frac{1}{x}$.

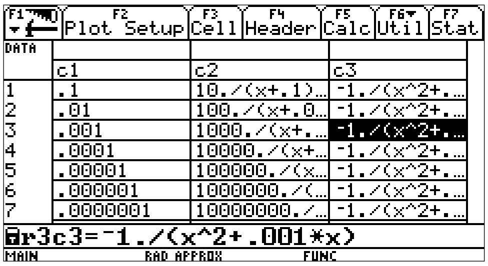
So edit c2 = (1/(x+c1)-1/x)/c1



	c1	c2	c3
1	.1	10./(x+.1)…	
2	.01	100./(x+.0…	
3	.001	1000./(x+.…	
4	.0001	10000./(x+…	
5	.00001	100000./(x…	
6	.000001	1000000./(…	
7	.0000001	10000000./…	

The output in column c2 is not obvious, but scrolling down the column and keeping an eye on the entry line, we see that we have the sum of two fractions. In such a case the function comdenom( ) comes in handy:

c3 = comdenom(c2)



	c1	c2	c3
1	.1	10./(x+.1)…	-1./(x^2+.…
2	.01	100./(x+.0…	-1./(x^2+.…
3	.001	1000./(x+.…	-1./(x^2+.…
4	.0001	10000./(x+…	-1./(x^2+.…
5	.00001	100000./(x…	-1./(x^2+.…
6	.000001	1000000./(…	-1./(x^2+.…
7	.0000001	10000000./…	-1./(x^2+.…

and we see that the result simplifies to the form

$$\frac{-1}{x^2 + k\,x},$$

where *k* tends to zero, giving the expected result.

Finally there is a very interesting example: *f*(*x*) = sin(*x*) using c2 = (sin(x+c1)-sin(x))/c1 and to expand the compound angle with c3 = texpand(c2).

See an attempt for a the DERIVE implementation - I used 0.1^(2i) steps:

F(x) :=

$$\text{diff1} := \text{VECTOR}\left(\left[0.1^{2\cdot i}, \frac{F(x + 0.1^{2\cdot i}) - F(x)}{0.1^{2\cdot i}}\right], i, 1, 5\right)$$

$$F(x) := \frac{1}{x}$$

$$\text{diff1} = \begin{bmatrix} 0.01 & -\dfrac{100}{x\cdot(100\cdot x + 1)} \\ 0.0001 & -\dfrac{10000}{x\cdot(10000\cdot x + 1)} \\ 0.000001 & -\dfrac{1000000}{x\cdot(1000000\cdot x + 1)} \\ 0.00000001 & -\dfrac{100000000}{x\cdot(100000000\cdot x + 1)} \\ 0.0000000001 & -\dfrac{10000000000}{x\cdot(10000000000\cdot x + 1)} \end{bmatrix}$$

In DERIVE some phantasy is required to make the desired result visible clear enough.
Perform each fraction such that the numerator becomes 1 and expand the denominator.

TASK: Use the spreadsheet approach to investigate other derivatives by first principles.

Try to visualise $f'(x) = \lim_{h \to 0} \dfrac{f(x+h) - f(x-h)}{2h}$.

Activity 8: Reduction formulae

Sometimes it is possible to express an integral in terms of a related, but simpler integral. This is known as a reduction formula. A common application is for high powers of trigonometric functions. For example, *If* $I_n = \int \cos^n x\,dx$, *then* $n\,I_n = \cos^{n-1} x \sin x + (n-1)\,I_{n-2}$.

A spreadsheet approach is suitable for investigating reduction formulae, since it is possible to list results for all values of n (n = 1, 2, 3, 4, ...)

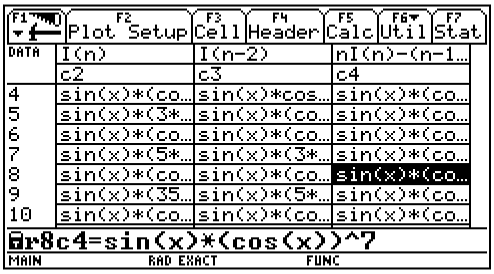
Firstly, let us verify the above reduction formula, starting a new datasheet called reduct.

c1 = seq(i,i,1,10)

c2 = ∫(cos(x))^c1,x)

c3 = ∫(cos(x))^(c1-2),x)

c4 = c1 * c2 - (c1 - 1)*c3

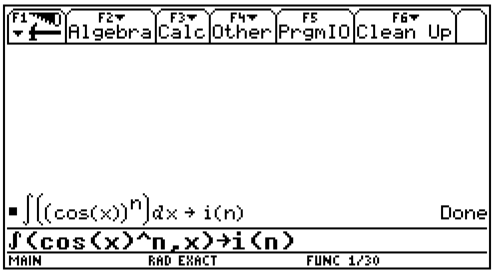


By scrolling down column c4, we see that it indeed appears to be the case that

$$n\,I_n - (n-1)\,I_{n-2} = \sin x \cos^{n-1} x.$$

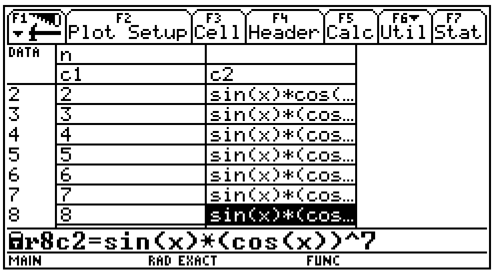
thus verifying the formula.

It is always useful to remember that functions defined in the Home screen retain their definition in the Data/Matrix Editor. A more succinct implementation of the above is:



c1 = seq(i,i,1,10)

c2 = c1 * i(c1) – (c1-1)*i(c1-2)



I followed David´s shorter approach for a DERIVE-version. In the file you will also find the realisation of the first datasheet. The statement:

```
REDUCT(n):=VECTOR(INSERT_ELEMENT(v SUB 1*v SUB 2-(v SUB 1-1)*v SUB 3,v,4),
           v,VECTOR([i,INT(COS(x)^i,x),INT(COS(x)^(i-2),x)],i,1,n))
```

gives:

$$\text{REDUCT2(5)} = \left[\text{SIN(x)}, \text{SIN(x)}\cdot\text{COS(x)}, \text{SIN(x)}\cdot\text{COS(x)}^2, \text{SIN(x)}\cdot\text{COS(x)}^3, \text{SIN(x)}\cdot\text{COS(x)}^4\right]$$

At last I strongly recommend to download David´s paper containing numerous ideas and tasks.