THE DERIVE - NEWSLETTER #41

THE BULLETIN OF THE

DERIVE®

USER GROUP

+ TI 92

Contents:

March 2001

[1] **Experimente aus der Physik mit CBL2 und TI-92+,** H.-D. Hinkelmann, bk-teachware. 2000, ISBN 3-901769-32-3

[2] **Mathematik mit dem TI-92 und TI-92+**, Reichel & Müller, öbv & hpt, 2000, ISBN 3-209-02447-2

[3] **Teaching Mathematics with ICT,** A. Oldknow & R.Taylor, CONTINUUM, 2000, ISBN 0-8264-4806-2

[4] **The Heart of Mathematics**, E B Burger & M Starbird, Key College Publishing, 2000, ISBN 1-55953-407-9

[5] **Mathematical Applications** for the Management, Life and Social Sciences, Harshbarger & Reynolds, Houghton Mifflin, 2000, ISBN 0-395-96142-4

Interesting WEB sites http://......

`www.kutzler.com`	Bernhars Kutzler´s personal web site
`www.bk-teachware.com`	Bernhard Kutzler´s Online Shop
`www.chartwellyorke.com/dfwind.html`	Philip Yorke´s DfW web site
`www.acdca.ac.at`	Austrian Center of Didactics of Computer Algebra
`www.mathgate.ac.uk`	A freely available Internet resource for mathematics
`www.joma.org`	Journal of Online Mathematics and its Applications
`www.mathdl.org`	Mathematical Sciences Digital Library
`www.dpgraph.com/`	download your free DPGraph Viewer
`icm.mcs.kent.edu/research/iamc.html`	DPGraph Gallery
`cage.rug.ac.be/~svw/DfW`	A new *DERIVE* site from the University of Ghent
`www.uni-klu.ac.at/ictmt5/`	Home page of the ICTMT5 Klagenfurt 2001

Our friend Paul Drijvers from the Netherland would like to inform you all about the

CAME 2001 Symposium
18 - 19 July in Utrecht
http://www.fi.uu.nl

Plenty of new materials for Teaching Mathematics supported by CAS and/or Dynamic Geometry produced by Austrian T³-Instructors covering subjects like Mathematics and Sports, Working with Matrices, Introduction in the Geometry Tool, Programming on the TI-89/92, Complex Numbers, ... are available.

They all are in German, but as they are full of screen shots and mathematics they can be useful even if you have very little knowledge in German. Browse and download at

http://www.acdca.ac.at the home site of T³-Austria

Dear friends,

I´ll start with my best regards and wishes for the new - our 11th DUG - year. Between mailing the last issue of the DERIVE Newsletter and now I received a lot of e- and other mails. Many thanks for them and for the broad positive consent to our work. And also many thanks for a bundle of proposals. One of them came from our German friend R. Baumann. He would like to encourage submitters to write their contributions sometimes in their mother tongue to give the DNL the many colours of an international forum which it really deserves. From my point of view this seems to be a good idea. Why not have by and then a French, an Italian, a Spanish, a Swedish, a .... contribution. Together with a short English summary of the content - could be done by the editor - and accompanied by mathematics, which fortunately is international enough, one should be able to follow the author´s ideas. So I´d like to ask all of you, who have "unrevealed" treas-treasures on their desks and don´t want to translate them into English: make I try and send it as it is. I must admit that I would have "minor" difficulties with a Japanese or Chinese paper.

You will find several contributions from my production in this issue. The reason is not that I am running short in contributions, oh no. But recently I was asked several times how to import data to *DERIVE* or into the TI´s memory. And then K.-H. Keuneckes paper landed in my mail box and at the same time I explored continuity and differentiability with my students on the TI-92. So it fit into the plan to combine Karl-Heinz´s contribution with one from the first DUG-days - Felix Schumm´s evolute - and with my very new one.

And as Johann Wiesenbauer alsways claims that there is so few programming stuff and we in Austria at the moment are very busy in making statistics teaching more attractive for the teachers I added a small *DERIVE*-program - several levels under Johann´s art of programming - as you can easily compare in his Titbits 19.

In this issue you will find a very extended User Forum with a lot of useful hints, because plenty of messages have accumulated since DNL#38

For the next DNL I can announce a nice short paper written by one of Josef Lechner´s students who successfully tried to overcome the lack of the ERF-Function on the TI-92 for a comfortable treating probability theory problems.

And two days ago I received an interesting paper from Colombia dealing with algorithms and recursions.

This time you will find some pages less as usual. The reason is the following: The Austrian Post increased mail charges enormously at begin of the year. They charge for mailing to US three times much as before. So we decided to save some pages in DNLs #41, #42 and #43 and to add all the missing pages to a full and rich DNL#44 by the end of the year. I hope that you will understand our position. If things will change in any way then we will come back to the usual content as soon as possible during the year.

At last I´d like to invite you registering for the ICTMT5 in Klagenfurt, Austria. As we don´t have the traditional DERIVE & TI-92 Conference this year, the event in Carinthia is a unique chance to meet the DERIVE- and TI-89/92 community in a wonderful surrounding.

We will have a Special Group "DERIVE, TI-89/92 and other CAS", chaired by B Kutzler, V Kokol-Volic and me. See you in Klagenfurt?

www.uni-klu.ac.at/ictmt5/

Best regards
Josef

Josef

Find all the *DERIVE* and *TI*-files on the following web sites

http://www.acdca.ac.at/t3/dergroup/index.htm

http://www.bk-teachware.com/main.asp?session=375059

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 40 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 - but not obliged - 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 2001
Deadline 15 May 2001

Preview: Contributions for the next issues

Inverse Functions, Simultaneous Equations, Speck, NZL
A Utility file for complex dynamic systems, Lechner, AUT
Examples for Statistics, Roeloffs, NL
Quaternion Algebra, Sirota, RUS
Various Training Programs for the TI
Sand Dunes, Halprin, AUS
Type checking, Finite continued fractions, Welke, GER
Kaprekar´s "Self numbers", Schorn, GER
Some simulations of Random Experiments, Böhm, AUT
Flatterbandkurven, Rolfs, GER
Basic Conceptions on Recursion Theory, Urrego, COL
A selfmade ERF-Function on the TI, Kremslehner, AUT
A Mistake presenting Graphs on the TI, Himmelbauer, AUT
A solution for the SNOG-Problem posed in DNL#39, Welke, GER
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

Hi, DERIVE programmers. It is a pleasure for me to present a website which is devoted to DfW5's most challenging feature: **Programming with DERIVE.** *For all of you who want to know more about this challenge and who find starting with Johann Wiesenbauer's examples too difficult I'd like to recommend David Leigh's website. He introduced himself and his website as follows:*

Hi, all
I am a student at Liverpool John Moores University, and my final year Research Project is 'Programming in Derive 5'. My project leader is Terence Etchells.

While the programming enhancements to version 5 are most welcome, the support for such programming is rather sparse. By 'support' I mean documentation of some kind - help files, textbooks, lecture notes, internet. In particular, would-be Derive Programmers who are lacking the most basics of programming techniques are not catered for.

Hence, the Web Site - address below.

To allow me to further develop the site, I would greatly appreciate some comments, from both new and experienced programmers. The emphasis of the site is on 'Basics', and I am trying to follow a 'tutorial' style - i.e. demonstrating basic routines by using simple examples.

If all goes well, the site should continue to develop after my project closes (I am pre-empting the effects of version 6, here!).

Thanks in advance,
David Leigh.

http://cmsproj.livjm.ac.uk/cmsdlei1

I received some questions concerning DERIVE-problems. As I am sure that some of you are facing the same problems I'll list them here accompanied by Al Rich's or Theresa Shelby's answers:

- **Question from Dieter Wickmann about "Hidden Functions and Variables":**

DfW5 is in Input Mode "Word" and "Sensitive". I define function BHD(w) (expr #7) in upper case. I enter bhd(0.8) - in lower case - . it appears as BHD and returns after simplifying the correct result. That is ok.
I save the worksheet and load it again. If I am entering now bhd(0.8) it appears as Bhd(0.8) - the "b" is transformed to a "B" - and there is no correct result at all. BHD(0.8) works properly.
Trying around I found a "trick": I highlight the function definition (expr #7), press ENTER twice - and bhd(0.8) will be changed in BHD(0.8) again. I save, I load and I need the same trick???

- **This is Al Rich's answer:**

After loading the file, issue the Declare > Function Definition command, and click on the pull-down menu to display a list of the currently defined function names. Note that both BHD and Bhd are included in the list of names. Thus, although the expression defining Bhd has been removed from the worksheet, the definition of the function Bhd remains active.

To remove such "hidden" function definitions and "hidden" variable values from a dfw-file, perform the following steps:

1. Start DERIVE and use the File > Open command to open the dfw-file.
2. Use the Edit > Select All command to highlight all the objects in the worksheet.
3. Use the Edit > Copy command to copy all the objects to the Windows Clipboard.
4. Use the File > Close command to close the window.
5. Use the File > New command to open a new window.
6. Use the File > Paste command to copy all objects into the new window.
7. Use the File > Save As command to save the new worksheet with either a new or the same name.

In a future version of DERIVE we will consider adding a command for clearing such "hidden" variables and functions.

- **Question from Josef Böhm "Truth values for relations":**

I am now working with equations and found out the following:

In DERIVE 4:

```
VECTOR(x^2 - 8·x < -15, x, 1, 5) = [false, false, false, true, false]
VECTOR(x^2 - 8·x = -15, x, 1, 5) = [false, false, true, false, true]
```

But now in DERIVE 5:

```
VECTOR(x^2-8·x<-15,x,1,5)=[false,false,false,true,false]
VECTOR(x^2-8·x=-15,x,1,5)=[-7=-15,-12=-15,-15=-15,-16=-15,-15=-15]
```

I´d expected to receive truth values in evaluating the equation, too??

- **Al Rich´s first answer:**

In Derive 5, equations do not simplify to true or false because many users want to solve equations manually. I fyou want to simplify an equation, call SOLVE with only the equation as an argument. For example

```
VECTOR(SOLVE(x^2-8·x=-15),x,1,5)=[false,false,true,false,true]
```

will work as you want. Hope this helps.

- **Josef was not satisfied:**

I would prefer receiving the result without using the SOLVE command. It is not clear for me, why it does work in an inequality, but does not in an equation. We don´t want to "solve" the equation, that´s the pedagogical reason why I would like to avoid the SOLVE command. The students should re-invent the natural way finding solutions by systematically guessing only - for inequalities and equations as well.

- **Al Rich´s second answer:**

After making the assigment

```
SimplifyRelations := true
```

equations as well as inequalities will simplify to true or false, if that can be determined. Also, linear equations and inequalities involving constant coefficients will also be solved when the equation is simplified. For example

```
2 x + 3 = 5
```

will simplify to `x = 1`.

- **What happened when Benno Grabinger and his students approximated π:**

Approximating 5419351/1725033 setting 100 digits for Number Display in the Output Settings gives a value for PI exact for the first 100 digits, which is obviously false, because this approximation delivers an accuracy of 12 digits only.
Working with APPROX(5419351/1725033,100) returns the correct - inaccurate - result. Why is *DERIVE* giving the impression that the fraction is a 100 digits accurate approximation in the first case? That seems to be a bad trap in which my students and I found ourselves caught for a while.

- **Al Rich:**

It is important to thoroughly understand the disitinction between the digits of notation that *DERIVE* uses to display numers AND the digits of precision that *DERIVE* uses to approximate irrational numbers. The former is set by the Declare > Output Settings command; whereas, the latter is set by the Declare > Simplification Settings command.

If the number of digits of notation is set to 100, when the rational number 1/3 is displayed in decimal, the decimal point will be followed by 100 three´s. Similarly, when the rational number 5419351/1725033 is displayed in decimal, the decimal point will be followed by 100 repeating decimal digits. This is as it should be, even though 5419351/1725033 also happens to be a 10 digit approximation of PI.

However, if you want to approximate PI to 100 digits of precision, use the Declare > Simplification Settings to change the digits of precision to 100 before simplifying. Or, better yet, use the Simplify Approximate command to temporarily set the precision to 100 digits while approximating PI.

Note that increasing the digits of precision has the side-effect of increasing the digits of notation used to display numbers. Thus, for most purposes (but not all!) the best thing to do is forget about the Declare > Output Settings command, and just use the Declare > Simplification Settings command to set the number of digits used to approximate AND display numbers. See the on-line help on both numerical precision and notation for details.

- **It is again Josef who bothers the DERIVE makers:**

Dear Al, dear Theresa,

(1) *I am preparing training units for Set Theory with DERIVE. I found strange rules when working with the difference of sets:*

(A \ B) \ C is simplified correctly (like (a - b) - c = a - b - c), but

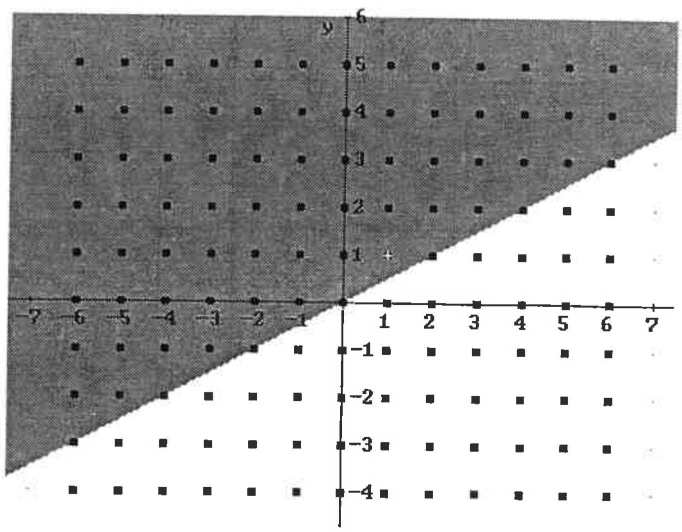
after editing A \ (B \ C) this appears as A \ B \ C in the Algebra Window and is evaluated as A \ (B \ C). (Thus a - (b - c) = a - b - c ???)

It is not easy to explain the students that A \ B \ C is interpreted by DERIVE as A \ (B \ C).

(2) *Now we are working with the Cartesian Product of sets introducing relations to proceed then to functions. It is very useful to set problems, let the students offer their suggestions and solutions and then show the results. Working with relations I don't would like to miss and their graphic representation as a powerful support.*

Given are two sets X and Y and I want to define the relation x < 2y. It works wonderful presenting the elements of X x Y as points superimposed by the shaded halfplane defined by x < 2y.

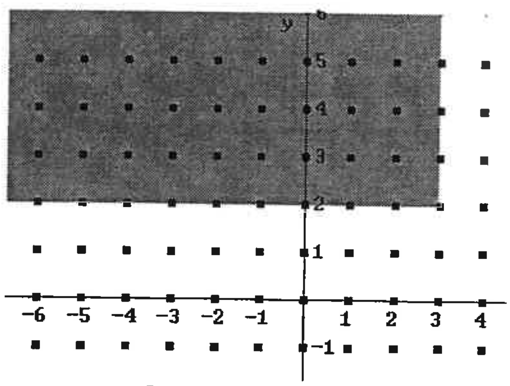
```
CaseMode := Sensitive
[X := {-6, ..., 6}, Y := {-4, ..., 5}]
[X·Y, x < 2·y]
```



Then I try

`[X·Y, x < 3 AND y ≥ 4]`

and it works properly:



But then, trying only `[X·Y, x < 3]` --> "Sorry, the highlighted expression cannot be plotted??? *If I first highlight X * Y, plot it and then highlight x < 3 and plot it, it works.*

- **Al Rich:**

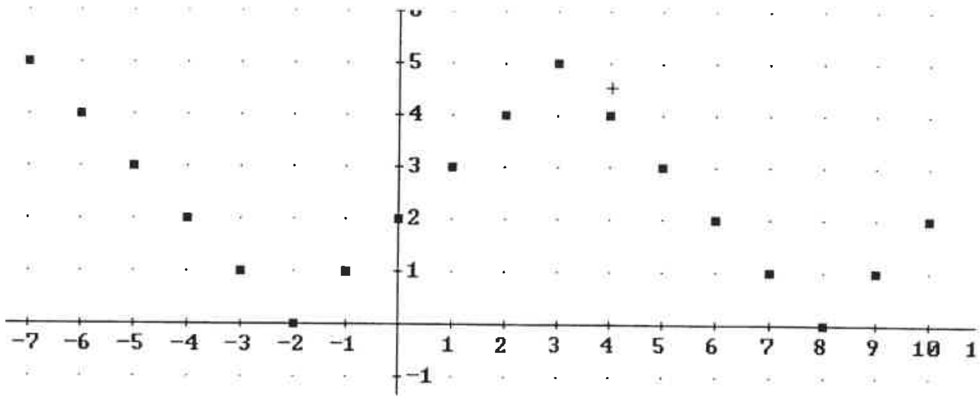
(1) Done. I have upgraded the \ (set difference) operator to be a full-fledged nary operator much like the UNION and INTERSECTION operators already are. This will give you the behavior you desire. [i.e. A \ B \ C is equivalent to (A \ B) \ C] and highlighting the operands will be easier as well.

(2) Fixed, *DERIVE* will no longer interpret two-element vectors having a set for an element as a parametrically defined curve, so your expression will plot as desired.

Talking about relations and their graphic representation: I found a little function very useful to represent relations on the grid. (Josef)

```
CaseMode := Sensitive

rela(u, v1, v2, S1, S2) := VECTOR(IF(SUBST(SUBST(u, v1, e_ ), v2, e_ ), e_,
                                                        1          2
                          [?, ?], [?, ?]), e_, S1·S2)
```

```
#27:  rela(y = |5 - |x - 3||, x, y, {-10, ..., 10}, {-10, ..., 10})
```



And again, it's me

- **By the way, concerning the grid:**

Dear Derive Makers,

(1) Producing notebooklike documents in dfw-format it would be useful to include hyperlinks. Presentations would become more comfortable to present and to watch as well, because the presenter could avoid boring for- and backwards scrollings.

(2) I receive many claims with respect to scaling in the 2D-plot Window. (And I am facing this problem now in classroom in almost each lesson). Very often teachers and students expect a scaling with units appearing in equal lengths on both axes - independent of the window size. Perpendicular lines shall appear as perpendicular, circles shall appear as circles, ... So I'd like to ask if there could be an option - very like to the TI-89/92 "ZoomDec" setting - to have immediately units of equal lengths on both axes. Maybe that there is a way to achieve this now very quick, but I don't know.
I do know the Set > Aspect Ratio command, but from my point of view then I am restricted to the square formed Plot Window. Let's assume I need for my (successful) DERIVE session a split screen with the left part - approximately one third of the screen - as Algebra Window and the other two third as 2D-plot Window. It takes a lot of time to have a group of students - or not much better a group of teachers - with the same arrangement and with at least approximately equal units on both axes.

(3) It might be helpful to make the most important settings of Simplification, Output, Input, Coordinate System, ... visible as small symbols, similar to the icons in the systray. Sometimes students produce strange results and a quick inspection of "settings corner" or "settings bar" would help.

(4) And not to forget to mention: when Johann Wiesenbauer and I met at the occasion of a nice Christmas celebration at the Vienna Technical University we had a long discussion about possible DERIVE improvements. We both agreed that it would be helpful to read functions and programs containing strings if the strings would appear under quotes on the DERIVE screen. Now one can see the quotes in the edit line only, but not on the screen and not on the print out. So the listings easily can be misunderstood.

- **That´s what the DERIVE smithies answered:**

Hello Josef,
(1) We will look seriously at including hyperlinks in DERIVE worksheets for the next major release of DERIVE.

(2) Unfortunately, the current design of the 2D-plot Window makes it very difficult to implement your wish for equal lengths beween tick marks, independent of plot window size. Currently the design of the window requires that the number of horizontal and vertical tick marks be an integer (i.e. the number of intervals set by the Set > Plot Region and Set > Plot Range commands). To implement your wish will mean that the number of tick marks not necessarily be an integer.

For the next major release of DERIVE we plan to redesign the 2D-plot window to eliminate this restriction. The redesign would have an additional benefit that resizing the plot window by drawing a box using the mouse will not result in long decimal numbers for the tick mark labels.

(3) A blue icon displayed at the left of the status bar for both the 2D and 3D-plot Windows shows the current coordinate system at a glance. Settings for the Algebra Window are currently shown by the dialog boxes of the Declare > Settings commands. Perhaps in a future version, using a tabbed dialogue box would make it easier to quickly see all the current settings.

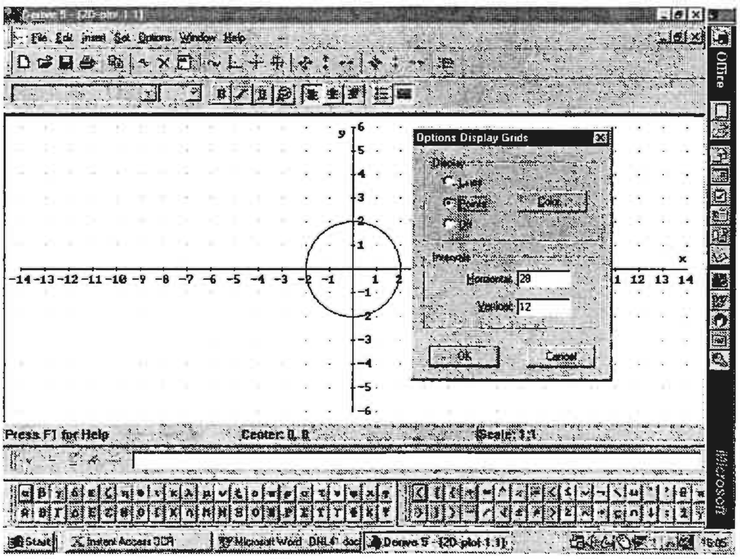
Perhaps an idea for the DUG Newsletter would be a "DERIVE 5 Tips & Techniques" column of useful, but not obvious, features that you and your readers may have discovered.

(4) Thank you for suggesting this improvement in the display of strings contained within function definitions being displayed on the screen. The next version (i.e. version 5.04) of DERIVE 5 will display such strings with quote marks. However, strings appear outside function definitions will continue to be displayed without quote marks,as they should be.

Following Al Rich´s suggestion "I proudly present":

DERIVE Tips & Techniques #1:

For a full 2D-plot screen I recommend to set in Options > Display Grids 28 horizontal and 12 vertical intervals in case of viewing the Math and Greek Symbol Toolbar. If they are switched off to have a larger plot screen then change to 28 horizontal and 14 vertical intervals.



If you want to split into Algebra / 2D Plot in a 1 : 2 ratio I recommend to set 18 and 12 intervals (+ tools bars).

You then can easily adjust other sizes using the Zoom In and Zoom Out buttons.

If you have your special tricks and techniques, then please share your experiencies with us.

Steven Schonefeld, USA

Josef, I have found a limit, which *DERIVE* cannot evaluate. It is the limit as *x* goes to positive infinity of:

#1: $f(x) := \frac{(x - 1)\cdot TANH(x) + x}{x - 2}$

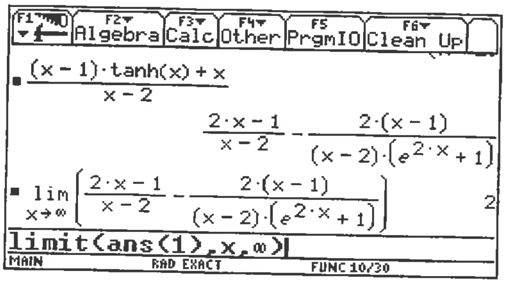
#2: $\frac{2}{(2 - x)\cdot(\hat{e}^{2\cdot x} + 1)} - \frac{2}{\hat{e}^{2\cdot x} + 1} + \frac{3}{x - 2} + 2$

#3: $\lim_{x \to \infty} f(x) = 2$

DNL:

Dear Steven,
as you can see above I can promise that future versions of DERIVE will be able to evaluate this limit. After expanding the expression the limit is eviden also for running versions

The TI-92 has learned its lection before →



By the way, Steven sent a nice joke about mathematicians and other fools. Here it is:

Simple Math

A mathematician, a statistician and an accountant were applying for the same job.

The interviewer called the mathematician and asked, "What do two plus two equal? The mathematician replied, "Four." The interviewer asked "Four, exactly?" The mathematician looked at the interviewer incredulously and said "Yes, four, exactly, definitely, alsways!"

Then the interviewer called in the statistician and asked the same question, "What do two plus two equal? The statistician said, "On average, four -- give or take ten percent, but on average, in most instances ... four."

Then the interviewer called in the accountant and posed the same question, "What do two plus two equal?" The accountant got up, locked the door, closed the shades, sat down next to the interviewer and said, "Well sir, with your current goals, tell me, what do you want it to equal?"

Editor´s note: a biologist was asked the same question the next day and responded, "2 + 2 of what?"

Wolfgang Lindner, Germany

attached a small DFW5-worksheet, showing ROW_REDUCE (GAUSSian algo) working an the special case of an 1x3-system: my students (and me too) were astonished, that ROW_REDUCE doesn't give back the reduced form in this case.

Would you please think about an 'correction' of that feature. I think ROW_REDUCE should answer in the recommended way. Or is there any reason, I have missed to know?

All the best in 2001 for you and DfW5>
Viele Gruesse

DNL: *See below the problem together with its solution.*

#1: ROW_REDUCE([10, -3, -2, 0]) = ROW_REDUCE([10, -3, -2, 0])

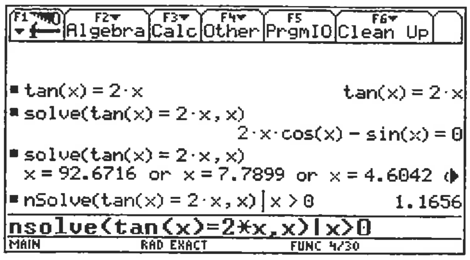
You have to enter the vector containing 4 components as a 1 x 4 matrix using double brackets!

#2: ROW_REDUCE([[10, -3, -2, 0]]) = $\left[\left[1, -\frac{3}{10}, -\frac{1}{5}, 0\right]\right]$

Hugh Porteous, England

I asked my students to find numerical solutions of tan(x) = 2x one of them typed SOLVE(TAN(x)=2x,x) into *DERIVE* and hit the approximate button on the toolbar. I was completely unable to explain the answer that *DERIVE* gave. Can anyone help?

```
#3:   SOLVE(TAN(x) = 2·x, x)

#4:   x = ± î·∞ ∨ x = 6.283185307 − î·∞ ∨ x = −3.141592653 + î·∞ ∨ x = 3.141592653 − î·∞ ∨ x
        = 3.141592653 + î·∞ ∨ x = 0

#5:   TAN(x) = 2·x

#6:   NSOLVE(TAN(x) = 2·x, x, 0.1, 10)

#7:                                          x = 1.165561185

#8:   NSOLVE(TAN(x) = 2·x, x, 1.17, 10)

#9:                                          x = 4.604216777
```



DNL: *There was an answer in the DERIVE news which blamed DfW4.11, but DERIVE 5.03 behaves in the same way. Asking for numerical solutions supported by the graph of TAN(x)-2x will help.*

Pedro Tytgat, Leuven, Belgium

Thanks a lot for the files! The way you are coaching the *DERIVE* community and the way you are caring about anyone asking for help is really phantastic.

Dear Pedro,
many thanks for your mail. We try to do our best.

Josef Lechner, Amstetten

In former *DERIVE* versions *DERIVE* returned parameter solutions for linear systems like the following, In DfW5 I miss this feature.

```
#12: SOLVE([x − y + 3·z = 10, 2·x + 4·y + 3·z = 2], [x, y, z])
```

#13: $\left[\; x = @1 \quad y = -\dfrac{@1 + 8}{5} \quad z = \dfrac{2\cdot(7 - @1)}{5} \;\right]$

DNL: *I am very proud to have an answer for Josef, who is one of our DERIVE super experts:*

```
#11:  SOLVE([x − y + 3·z = 10 ∧ 2·x + 4·y + 3·z = 2], [x, y, z]) = [2·x + 5·z = 14 ∧ 2·y −
        z + 6 = 0]
```

#12: SOLVE([x − y + 3·z = 10 ∧ 2·x + 4·y + 3·z = 2], [x, y]) = $\left[x = \dfrac{14 - 5\cdot z}{2} \wedge y = \dfrac{z - 6}{2}\right]$

BUT:

#13: SOLUTIONS([x − y + 3·z = 10 ∧ 2·x + 4·y + 3·z = 2], [x, y, z]) = $\left[\left[@1, -\dfrac{@1 + 8}{5}, \dfrac{2\cdot(7 - @1)}{5}\right]\right]$

***DERIVE* 5.03 provides full compatibility with Windows 2000. Download the free update to your existing *DERIVE* 5 software (English version only) at:**

http://www.ti.com/calc/docs/derive5update.htm

The MedMed-Regression with *DERIVE*

If you are working with a TI-83/89/92/92+ and you use one of these wonderful tools for doing statistics you will have met among other regression types the *MedMed-Regression*. Applying this type you receive a line different from the line resulting from linear regression. Once I asked myself, how to obtain this result and tried to find an answer in several statistics books, including the TI-manuals. Result: No answer at all. Do you know the answer, except the fact that it obviously seems to have some connection with the median? Then I wrote to my friend and statistics specialist Fritz Tinhof, and he sent a copy of a webpage printout, which he had downloaded some times ago

`www.washlee.arlington.k12.va.us/DEPARTMENTS/Maths/data/content3.html`

- this site doesn´t exist any more - and wished me luck.

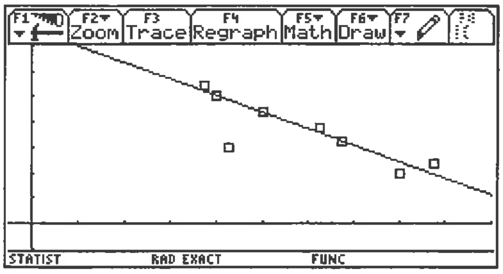
And here I found a nice description of the process. I don´t want to copy the paper but include the "recipe" into a documented *DERIVE*-program using Walter Schiller´s very useful and valueable TOM.EXE to convert a program written with a text editor into a MTH-file (find more in DNL#40).

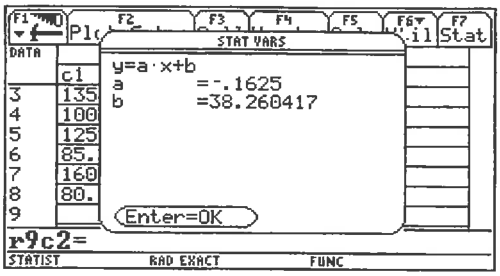
(No problems, if you don´t have TOM.EXE at your disposal, the DERIVE program can be found two pages later or you can download the files belonging to DNL#40 from ACDCA´s or bk-teachware´s DUG-page.)

Let´s take the following data, which shows data from several cars hp versus gasoline consumption:

$$hp := \begin{bmatrix} \text{hp} & 75 & 175 & 135 & 100 & 125 & 85 & 160 & 80 \\ \text{m/gal} & 27 & 12 & 16 & 22 & 19 & 15 & 10 & 25 \end{bmatrix}$$

First I will perform the process on my TI to provide a check for my desired *DERIVE* procedure:





Open any text editor and then edit the program given below. All comments can be found between /* and */. The comments will not be considered when compiling the program.

```
/* Example of a program procedure realized using TOM.EXE   */

/* Find the median of a list given by a vector              */
/* Call the function as median(list)                        */
median(liste) := prog
/* we start sorting the list in ascending order             */
/* save the sorted list as slist and find the median        */
   slist := sort(liste)
   n := dim(liste)
   if mod(n,2) = 1
            then
        slist sub( (n+1)/2)
            else
        (slist sub (n/2) + slist sub (n/2+1))/2
end
/* End procedure median                                     */
/* That will create the function median(list)               */
```

```
/* Now the MedMed regression line: Function call will be mdreg(list)  */

mdreg(list) := prog

/* sort the list wrt to the x-values and store number of elements      */

   list := sort(list)
   n := dim(list)
   fl := floor(n/3)
   r := mod(n,3)

/* We split the list into three groups, based on their x-values   */
/* If the points will not go evenly into three groups, split the  */
/* points so the 1st and 3rd group have the same number of points */

   if r = 0
          then

/* now the groups are of equal size with boundaries               */

      grg := [fl,fl+1,2*fl,2*fl+1]
          else

/* the middle group contains one elment less                      */

          if r = 1
                   then
             grg := [fl,fl+1,2*fl+1,2*fl+2]

/* and now it contains one element more than the other two        */

                   else
             grg := [fl+1,fl+2,2*fl+1,2*fl+2]

/* The groups are formed, i.e. we split up the given n x 2 matrix
   in three partial matrices.                                    */

   gra := list row [1,...,grg sub 1]
   grb := list row [grg sub 2,...,grg sub 3]
   grc := list row [grg sub 4,...,n]

/* Using the median function from above we calculate the medians of
   the x- and y-values of each matrix and store them as vectors.    */

   p1 := [median(gra col 1), median(gra col 2)]
   p2 := [median(grb col 1), median(grb col 2)]
   p3 := [median(grc col 1), median(grc col 2)]

/* Points p1 and p2 determine the slope m_ of the regression line */

   m_ := (p3 sub 2 - p1 sub 2)/(p3 sub 1 - p1 sub 1)

/* The y-intercept d1 of line(p1,p2) is given as                    */

   d1 := p3 sub 2 - m_ * p3 sub 1

/* The y-intercept d2 of the line parallel to the line above through p2*/

   d2 := p2 sub 2 - m_ * p2 sub 1

/* the MedMed-line has slope m_ and the weighed y-intercept (2 * d1 + d2)/3
   this makes sense, because d1 is determined by 2 points, d2 only by one*/

   expand(m_*x + (2*d1+d2)/3)
end
/* End procedure mdreg                                        */
```

```
/* The final result should be a table called by medreg(list). The table  */
/* should show the regression line and the sum of the squared errors.    */

medreg(list) := prog
   sse := sum((list sub i sub 2 - lim(mdreg(list),x,list sub i sub 1))^2,
i,1,dim(list))
   ["MedMed-Line:",mdreg(list);"SumSquErr^2:",sse]
end
/* End procedure medreg                                                   */


/* Because it is so easy we include two data sets: the legths of bean
   plants  */

beans:=[2.8,5.9,8.8,10.2,11.5,12,19.8,14.5,15,16.3,14,14.4,14.2,7.3,11,9.7,
10.2,11.4,16.9,13.9,12.6,12.4,8.5,9,10,13.1,8.6,11.4,11.9,17.2]

/* The data for the regression:                                           */

hp:=[75,27;175,12;135,16;100,22;125,19;85,15;160,10;80,25]
```

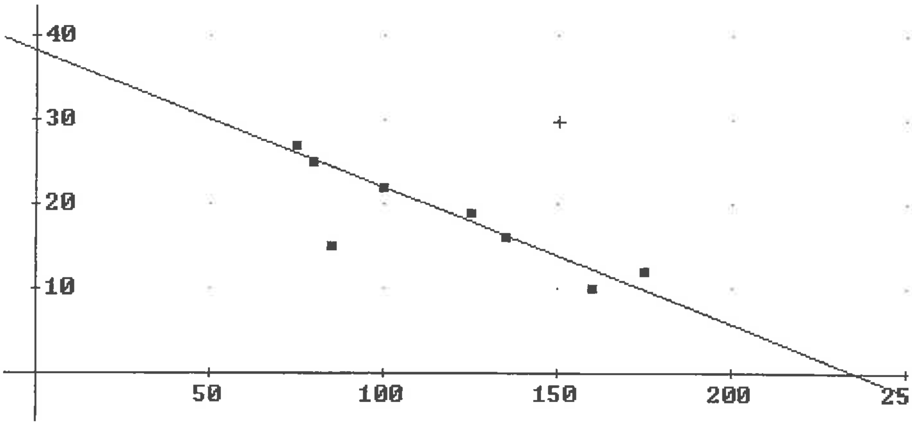
Save this file under a chosen filename. eg. medcomp.drv - drv stands for *DERIVE* - in the same folder where you have stored TOM.EXE. Then run **tom.exe medcomp.drv** and in case of not having made any typing error you will find a correct **medcomp.mth** in this folder, which you can open with *DERIVE 5* to find your file changed - see **mdreg(list,...)** below and note the many additional parameters - which you don´t need to consider using TOM.EXE.

```
       mdreg(list, d2, p3, m_, grb, r, d1, grg, f1, p2, gra, n, p1, grc) :=
         Prog
           list := SORT(list)
           n := DIM(list)
           f1 := FLOOR(n/3)
           r := MOD(n, 3)
           If r = 0
              grg := [f1, f1 + 1, 2·f1, 2·f1 + 1]
              If r = 1
                 grg := [f1, f1 + 1, 2·f1 + 1, 2·f1 + 2]
#2:              grg := [f1 + 1, f1 + 2, 2·f1 + 1, 2·f1 + 2]
           gra := list ROW [1, ..., grg↓1]
           grb := list ROW [grg↓2, ..., grg↓3]
           grc := list ROW [grg↓4, ..., n]
           p1 := [median(gra COL 1), median(gra COL 2)]
           p2 := [median(grb COL 1), median(grb COL 2)]
           p3 := [median(grc COL 1), median(grc COL 2)]
           m_ := (p3↓2 - p1↓2)/(p3↓1 - p1↓1)
           d1 := p3↓2 - m_·p3↓1
           d2 := p2↓2 - m_·p2↓1
           EXPAND(m_·x + (2·d1 + d2)/3)
```

Now lets try **median(beans)** and **medreg(hp)**:

```
#6:                              median(beans) = 11.7

#7:              medreg(hp) = [ MedMed-Line:   38.2604 - 0.1625·x ]
                              [ SumSquErr^2:         101.250      ]
```

and watch the plot:

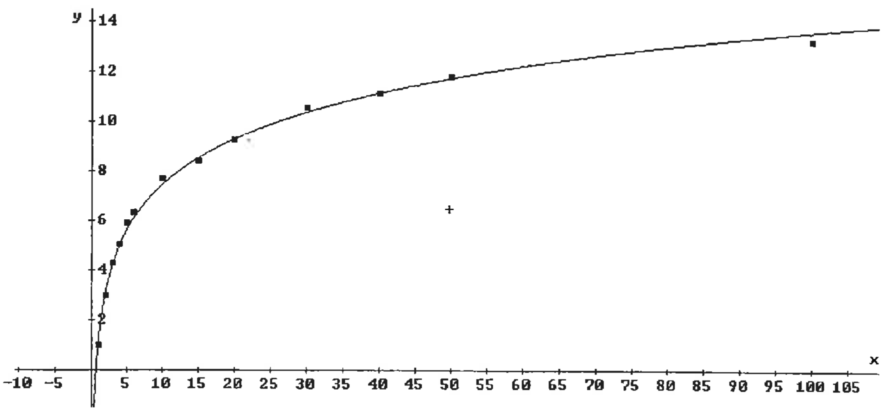


Of course, you could also edit these programs (functions) via the edit line. But I bet that most of you are not as experienced as Johann Wiesenbauer or Terence Etchells. (I am sure that there are a few more people, who will be able to do that, but not so many). But look at the "nice" program code and imagine to enter it character by character, without missing any single one:

```
median(list,slist,n):=PROG(slist:=SORT(list),n:=DIM(list),IF( MOD(n,2)=1,
slist SUB ((n+1)/2),(slist SUB (n/2)+slist SUB (n/2+1))/2))

mdreg(list,d2,p3,m_,grb,r,d1,grg,f1,p2,gra,n,p1,grc):=PROG(list:=SORT(list)
,n:=DIM(list),f1:=FLOOR(n/3),r:=MOD(n,3),IF( r=0,grg:=[f1,f1+1,2*f1,2*f1+1]
,IF( r=1,grg:=[f1,f1+1,2*f1+1,2*f1+2],grg:=[f1+1,f1+2,2*f1+1,2*f1+2])),gra:
=list ROW [1,...,grg SUB 1],grb:=list ROW [grg SUB 2,...,grg SUB 3],grc:=li
st ROW [grg SUB 4,...,n],p1:=[median(gra COL 1),median(gra COL 2)],p2:=[med
ian(grb COL 1),median(grb COL 2)],p3:=[median(grc COL 1),median(grc COL 2)]
,m_:=(p3 SUB 2-p1 SUB 2)/(p3 SUB 1-p1 SUB 1),d1:=p3 SUB 2-m_*p3 SUB 1,d2:=p
2 SUB 2-m_*p2 SUB 1,EXPAND(m_*x+(2*d1+d2)/3))

medreg(list,sse):=PROG(sse:=SUM((list SUB  i SUB 2-LIM(mdreg(list),x,list
SUB  i SUB 1))^2,i,1,DIM(list)),["MedMed-Line: ",mdreg(list);"SumSquErr^2:
",sse])
```

What is your choice?? See another example resulting from my programming experiments:
(Data from DNL#39)

```
                          ⎡ Regression line:  2.661329232·LN(x) + 1.336675569 ⎤
                          ⎢   SumSquErr^2:              0.5424936129          ⎥
#30:  lnreg(snog`) =      ⎢    CorrCoeff:               0.9983084847          ⎥
                          ⎣ Measure of Det.:            0.9966198307          ⎦
```



New Conceptions for Teaching Natural Science –
– Experiments Using CBL 2/CBR and the TI-92 Plus

Heinz-Dieter Hinkelmann, Korneuburg, Austria

1. Introduction

With this title a series has been started to enlarge the possibilities of application of the TI-92 Plus under the aspect of natural science. For that purpose 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 some experiments for natural science lessons and their mathematical analysis. If anybody has further ideas, suggestions or questions, just please contact me by using the following address:

hd.hinkelmann@aon.at

The measuring instruments will be shortly described for those who don´t know them. Today´s edition describes the new CBL 2 and the activity of freely-falling objects.

2. The new CBL 2

The CBL 2™ is easier to use and more powerful than the original CBL. It provides the easiest, most accessible way for students to collect and analyze real world data right out of the box. Students can discover how math and science affect the world around them by developing hypothesis, experimenting, and forming conclusions.

You can start collecting data in four easy steps:

- Transfer the built-in user program DataMate from the CBL 2 to your calculator with a single button push.
- Run the program.
- Plug in one of the auto-ID sensors, or set up the CBL 2 for more sophisticated applications.
- Begin collecting data.

More than 40 sensors, such as the Motion Detector, Dual Range Force Sensor, pH Sytem, or Oxygen Gas sensor, can be connected to the CBL 2. Since CBL 2 is portable and battery powered, you can take it out of the classroom for experiments such as measuring accelerations on amusement rides or monitoring the temperature, dissolved oxygen, and pH of a lake or stream. Within the classroom you can use the CBL 2 and a graphing calculator as an alternative to computers for collecting and analysing real-world data.

The CBL 2 interface includes three probes: a new stainless steel temperature probe, voltage probe, and light probe. It also includes a calculator cradle (cradle will not accept TI-92 (Plus)), a six-inch link cable, a Teacher Guide containing experiments for science and math, a TI resource CD, and 4 AA batteries. It is recommended to use the TI-9920 AC adapter (sold separately).

FLASH The upgradable Flash Memory™ of the CBL 2 allows you to update the user program with the latest software version. You can use the remaining Flash memory to store multiple experiment trials, experiment setups, or other programs.

CBL 2 has four channels for data collection: 3 analog, 1 sonic/bidirectional digital (to connect e.g. the Ultrasonic Motion Detector. You can collect approximately 12,000 data points (depending on memory) at rates of up to 50,000 points per second and channel.

3. Activity with the Motion Detector

3.1 The Ultrasonic Motion Detector

The Motion Detector functions like the automatic range finder on a Polaroid camera. This sonar device emits ultrasonic pulses and waits for an echo. The time it takes for the reflected pulses to return is used to calculate distance, velocity, and acceleration. The range is 0.4 to 6 meters. The Motion Detector has a pivoting head, rubber feet, and a clamp for mounting.

Specifications:

Frequency of the ultrasound: 49.4 kHz
Resolution: 1.1 mm
Typical Accuracy: ± 2 mm
Range: 0.45 to 6.0 meters
Power: about 51mA @ 5 V DC while running
Size: approximately 5.75" x 1.5" x 2.25"
Weight: approximately 10 oz.

3.2 The Activity

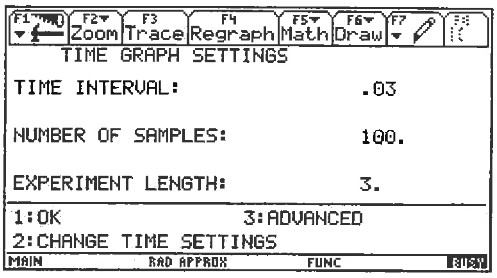
If objects are dropped from a certain height in the atmosphere down onto the Earth, air resistance and friction are at work. Light objects with greater air resistance will move towards the Earth with constant speed. A heavier object, that has less air resistance, will move towards the ground with almost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

Equipment required

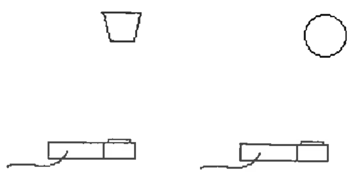
- Calculator
- Motion Detector
- Connecting cable
- Ball (light)
- Coffee filter (cone shaped) or a cone shaped paper cup

Setup procedure

- ❑ Connect the CBL 2 to the TI-92 Plus, and the Motion Detector to the sonic/bi-directional digital channel.
- ❑ Run the DataMate program on the TI-calculator.
- ❑ Choose the following time settings:

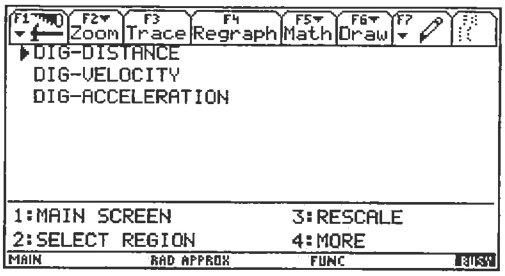


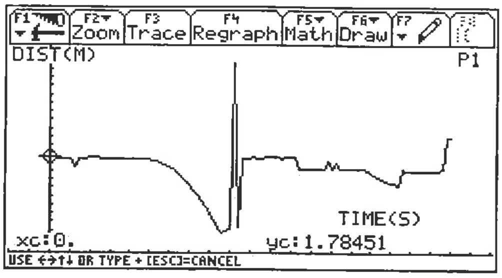
- ❑ Confirm "1:OK" twice to display the "start screen".
- ❑ In this experiment the Motion Detector is placed on the ground and a light ball (e.g. table tennis ball) is dropped onto it. Then in the second test, a coffee filter is dropped onto the detector.



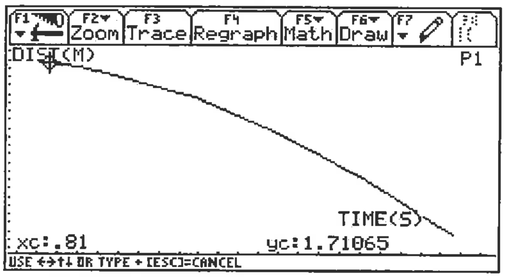
Analysis of the data obtained with the ball

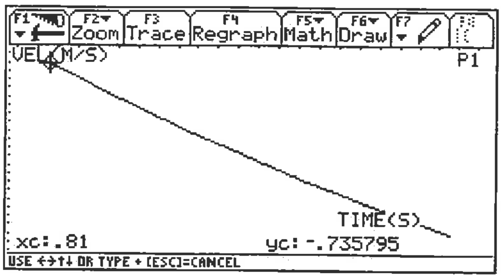
- As soon as the data collection has been completed a menu will now be displayed, where you can alternatively call up the distance-time, the velocity-time, or the acceleration-time graph.





- Now change the graph so that only the relevant part of the distance-time graph can be seen. In the same menu select "2:SELECT REGION". This will enable the determination of the left and right bounds (trace with the cursor and confirm with <Enter>) of the selected domain. After selecting the right bound (and after a brief calculation) the new distance-time graph will be displayed. In the familiar menu you can choose "DIG-VELOCITY", which initialises the velocity-time graph.

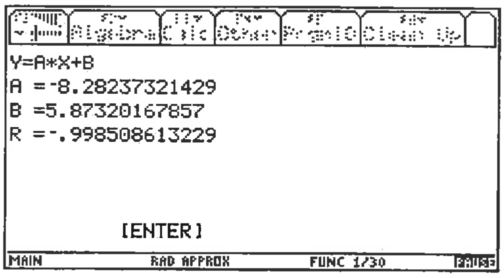


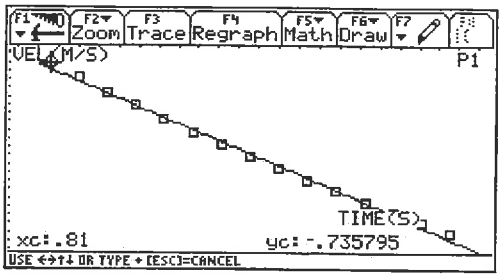


You see that the velocity of the ball is negative, but increasing in magnitude as the ball moves towards the Motion Detector. From the graph we can deduce that the acceleration appears to be almost constant. Uniform acceleration occurs if the ball offers very little air resistance.

To confirm the uniformly accelerated motion we determine the regression line of velocity.

- Choose „1:MAIN SCREEN" from the graph-selection menu. In the following main menu choose "4:ANALYZE" and then "2:CURVE FIT". Now select "5:LINEAR (VELO VS TIME)". The equation for the regression line is then displayed in a window and the values are very close to the measured data. After confirming with <Enter> you will open the graph window in which collected data and regression function are superimposed.





Analysis of the data obtained with the coffee filter

Analysis of the data obtained with the coffee filter is carried out in the same way. We will find that air resistance and friction of the filter (which is very light and has a large surface area) prevents constant velocity. "4:LINEAR (DIST VS TIME)" will confirm this.

Titbits from Algebra and Number Theory (19)

by Johann Wiesenbauer, Vienna

Time flies! Would you believe that DfW 5 has been around for more than one year now? Boy, a lot of things have changed since its advent, in particular when it comes to programming! Every time when I try to read one of those old programs written in a version before 5, I can see the difference. For example, have a look at the following program from my "Titbits(14)" in the DNL #33:

```
STIRLING1(n, k) := IF(k >= n, MAX(1 - k + n, 0), IF(k, 0, IF(k = 1, (n -1)!,  IF(k = n - 1,
COMB(n, 2), IF(k > n/2, (ITERATE([n_ DELETE_ELEMENT(v_)+ DELETE_ELEMENT(v_,
1 - n_ + n), n_ + 1], [v_, n_], ITERATE([n_APPEND(DELETE_ELEMENT(v_), [0]) + v_,  n_ + 1],
[v_, n_], ITERATE([n_APPEND(v_,[0]) + INSERT_ELEMENT(0, v_), n_ + 1], [v_, n_],  [[1], 0],
n -k), 2k - n), n - k))SUB1SUB1, (ITERATE([n_ DELETE_ELEMENT(v_, 1 - n_ + n) +
DELETE_ELEMENT(v_), n_ + 1], [v_, n_], ITERATE([n_v_ +APPEND(DELETE_ELEMENT(v_),
[0]),  n_ + 1], [v_, n_], ITERATE([n_INSERT_ELEMENT(0, v_) + APPEND(v_, [0]), n_ + 1],
[v_, n_], [[1], 0], k), n- 2·k), k))SUB1SUB1)))))
```

Believe me, I had a really hard time when trying to analyze it in order to write a new version in DfW5 and I fully understand now if you couldn't make head or tail of it then either. But now look at its "translation" into DfW5:

```
STIRLING1(n, k, n_ := 0, v_ := [1]) :=
  Prog
    If k ≥ n
       RETURN MAX(n - k + 1, 0)
    If k = 0
       RETURN 0
    If k = 1
       RETURN (n - 1)!
    If k = n - 1
       RETURN COMB(n, 2)
    If k > n/2
       Prog
         Loop
           If n_ = n - k exit
           v_ := n_·APPEND(v_, [0]) + ADJOIN(0, v_)
           n_ :+ 1
         Loop
           If n_ = k exit
           v_ := REST(n_·APPEND(v_, [0]) + ADJOIN(0, v_))
           n_ :+ 1
         Loop
           If n_ = n
              RETURN FIRST(v_)
           v_ := DELETE(REST(n_·APPEND(v_, [0]) + ADJOIN(0, v_)), -1)
           n_ :+ 1
    Loop
      If n_ = k exit
      v_ := n_·ADJOIN(0, v_) + APPEND(v_, [0])
      n_ :+ 1
    Loop
      If n_ = n - k exit
      v_ := REST(n_·ADJOIN(0, v_) + APPEND(v_, [0]))
      n_ :+ 1
    Loop
      If n_ = n
         RETURN FIRST(v_)
      v_ := DELETE(REST(n_·ADJOIN(0, v_) + APPEND(v_, [0])), -1)
      n_ :+ 1
```

Wow, what a difference! It's almost sort of embarassing that everybody can see now that I exploited a very simple idea again and again. On top of being very transparent this piece of code is also awfully fast. For example, the following computation took Derive only 2.1s on my PC, whereas Maple VI came up with the result only after 66s !

```
STIRLING1(1000, 500)

3898591727837367216096935158640785876651038507998386119589034461892038237966636723402506~
  95251610953585104919528127062876757223732918630743647185117288267265062670668130707351~
  72888942544782163964879065197918338412552523923274315890401708777454018700351846078860~
  29576155298065039322682831203107130479100929154906740657340511915593402739217723750127~
  49341938251783236265388124900819769966113275540714525635126856314949379264438044385729~
  97162978193722382173907190491884406021761586102737933407411585186205022846397142752310~
  82441790537707687478246686707174208638736297434528215596559357751596931167852214740910~
  71453904435017963913857798990861166462102333334209008581792778217958851635484464404910~
  06540324466139539625146878485043037072836334454661046330755024004887982973212624652014~
  78482053817984392992793482584299684044297208177962224139040013337793843882099466352974~
  52413642089460132594646802167253033629451381893357104538142554413259974479698159763035~
  94239063828873768951855872948875443940233586767235819328363351833708140264283041929729~
  46736079832048111944930196737312799858645290132005523887352902557151958521904827250623~
  38989047694584273948621455448883715972563151019066588288500459010021454391487947358727~
  46700430586522287683772458572138024850889879774304486561587969450422484871303569303213 8~
  38585175777705472340580028401702224069016816803706447625381968209958988515437739204950~
  44453165503147535080691800419289717887449908008344477317349034127631933523108185118400~
  82156455218388450878887203623938000367095510285823802313606131603151760436835432351960 7~
  1191700895216438940550320450235792778115936894000573334284734063208174959 5569
```

For the sake of completeness I also give you the new form of STIRLING2(n,k) which is even more beautiful due to its shortness:

```
STIRLING2(n, k, u_, v_ := [1]) :=
  Prog
    If n ≤ k
       RETURN MAX(0, n - k + 1)
    u_ := VECTOR((-1)^(k - j_)·j_^n, j_, 0, k)
    n := k
    Loop
      If k = 0
         RETURN u_·v_/n!
      v_ := ADJOIN(0, v_) + APPEND(v_, [0])
      k :- 1
```

Again its performance demolishes all rival CAS (check e.g. STIRLING1(2000,1000) in Maple or Mathematica!)

By the way speaking of Maple, it has a nice bunch of functions dealing with computations in symmetric groups going far beyond what I did in DfW 4.11 in my "Titbits(15)" (cf. the DNL #34). By coincidence I have been asked recently by Prof. J.C. Morales from the University of Colombia to rewrite some of these routines in DfW5. To be more specific, he wanted me to write the Derive versions of MULPERMS (which multiplies two permutations in disjoint cycle notation and expresses the product in disjoint cycle notation too) and of CONVERT_PERMLIST (which converts a permutation in disjoint cycle notation into list notation). Let's start with the latter function which I called toperm(p,n) for the sake of brevity:

```
toperm(p, n) :=
  Prog
    If p = []
       RETURN [1, ..., n]
    p := APPEND(VECTOR(APPEND(u_, [FIRST(u_), 0]), u_, p))
    p := SELECT(Π(p_) > 0, p_, [DELETE(p, -1), REST(p)]`)
    Loop
      If n = 0
         RETURN SORT(p)`↓2
      If ¬ MEMBER?(n, FIRST(p`))
         p := ADJOIN([n, n], p)
      n :- 1
```

Here the first argument p is a permutation in disjoint cycle notation and n is the degree of the symmetric group in question. For example, we get

```
toperm([], 6) = [1, 2, 3, 4, 5, 6]

toperm([[2, 4, 1], [7, 3]], 7) = [2, 4, 7, 1, 5, 6, 3]
```

(Note that the input [ ] denotes the identical mapping.) Just in case you prefer the "textbook notation" of a permutation, here is a variant to this end:

```
TOPERM(p, n) := [[1, ..., n], toperm(p, n)]
```

$$\text{TOPERM}([[2, 4, 1], [7, 3]], 7) = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 4 & 7 & 1 & 5 & 6 & 3 \end{bmatrix}$$

On many occasions also the inverse function of toperm(p,n) is needed, which I called tocycles(p). This time p is assumed to be given in list notation and the outcome is p in disjoint cycle notation.

```
tocycles(p, c_, p_ := [], s_) :=
  Prog
    If VECTOR?(FIRST(p))
       p := p↓2
    s_ := {1, ..., DIM(p)}
    Loop
      If s_ = {}
         RETURN REVERSE(p_)
      c_ := ITERATES(p↓k_, k_, MIN(s_))
      If DIM(c_) > 2
         p_ := ADJOIN(DELETE(c_, -1), p_)
      s_ := s_ \ MAP_LIST(c_↓k_, k_, {1, ..., DIM(c_)})
```

```
tocycles([2, 4, 7, 1, 5, 6, 3]) = [[1, 2, 4], [3, 7]]
```

$$\text{tocycles} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 4 & 7 & 1 & 5 & 6 & 3 \end{bmatrix} = [[1, 2, 4], [3, 7]]$$

(As a matter of fact, the outcome looks even "tidier" than our input above due to a clever reordering of the digits.)

Actually, this is already my third version of tocycles(p) (cf. DNL #28, p.43, and DNL #34, p32). The very first one I wrote at Carl Leinbach's request, but I'm afraid the code then was far from being very readable. (This wasn't my fault though, but rather a common feature of most Derive-programs at that time!) Well, Carl, I put my best foot forward again and hopefully you agree that at last this version leaves nothing to be desired as regards both performance and readability! (In fact, I'm inclining to say that it is "the best thing since sliced bread", "the cat's whiskers", "the bee's knees" or what have you, but for people who aren't in the know this might be too much of a self-praise!)

Having these mighty tools at hand the programing of MULPERMS is child's play now. Here is the version of MULPERMS that works exactly as the corresponding version in Maple:

```
MULPERMS(a, b, n_, s_ := 1, t_, v_) :=
  Prog
    If a = [] ∨ b = []
       RETURN APPEND(a, b)
    n_ := MAX(APPEND(APPEND(a), APPEND(b)))
    a := toperm(a, n_)
    b := toperm(b, n_)
    tocycles(VECTOR(b↓(a↓k_), k_, 1, n_))

MULPERMS([[2, 3, 4], [1, 6]], [[3, 4, 5], [1, 2]])

                                   [[1, 6, 2, 4], [3, 5]]
```

I for my part prefer an other version though where the functions a and b are carried out in the usual order, viz. "from right to left", as in the following version of MULPERMS (I leave it to you to choose the one that fits you!)

```
mulperms(a, b, n_, s_ := 1, t_, v_) :=
  Prog
    If a = [] ∨ b = []
       RETURN APPEND(a, b)
    n_ := MAX(APPEND(APPEND(a), APPEND(b)))
    a := toperm(a, n_)
    b := toperm(b, n_)
    tocycles(VECTOR(a↓(b↓k_), k_, 1, n_))

mulperms([[2, 3, 4], [1, 6]], [[3, 4, 5], [1, 2]])
                                        [[1, 3, 2, 6], [4, 5]]
```

Here some readers may raise the following question: Why the hell didn't he make use of MULPERMS(a,b) when programming mulperms(a,b) in the following obvious way:

```
mulperms(a, b) := MULPERMS(b, a)
```

Well, in my eyes this way of programming is not totally "clean", as by the use of an auxiliary function mulperms(a,b) is no longer "self-contained". (This remark is only for absolute purists when it comes to programming. If you are fond of auxiliary functions as many people are, please don't worry and stick to them!)

Now I would like to show how one can use the routines above to classify the elements of a symmetric group S_n for any given n according to various criteria. To do so we first need a program that computes the elements of S_n for any given n. (Have you any idea how such a program could look like? Try to write it yourself before looking at my solutions below!) First I give the version where the output is in list notation:

```
SYMGROUP(n, k_, n_ := 2, s_ := [[1]], t_) :=
  Loop
    If n_ > n
       RETURN SORT(s_)
    k_ := n_
    t_ := []
    Loop
      If k_ = 0 exit
      t_ := APPEND(t_, VECTOR(INSERT(n_, v_, k_), v_, s_))
      k_ :- 1
    s_ := t_
    n_ :+ 1
```

$$\text{SYMGROUP(3)} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \\ 2 & 1 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \\ 3 & 2 & 1 \end{bmatrix}$$

And now the "real thing", where the output is in disjoint cycle notation:

```
symgroup(n, k_, n_ := 2, s_ := [[1]], t_) :=
  Prog
    Loop
      If n_ > n exit
      k_ := n_
      t_ := []
      Loop
        If k_ = 0 exit
        t_ := APPEND(t_, VECTOR(INSERT(n_, v_, k_), v_, s_))
        k_ :- 1
      s_ := t_
      n_ :+ 1
    s_ := REST(VECTOR(tocycles(v_), v_, s_))
    s_ := VECTOR(ADJOIN([LCM(VECTOR(DIM(u_), u_, v_)), DIM(APPEND(v_))], v_), v_, s_)
    ADJOIN([], VECTOR(REST(v_), v_, SORT(s_)))
```

symgroup(3) = [[], [[1, 2]], [[1, 3]], [[2, 3]], [[1, 2, 3]], [[1, 3, 2]]]

symgroup(4) = [[], [[1, 2]], [[1, 3]], [[1, 4]], [[2, 3]], [[2, 4]], [[3, 4]], $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$, $\begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix}$, [[1, 2, 3]], [[1, 2, 4]], [[1, 3, 2]], [[1, 3, 4]], [[1, 4, 2]], [[1, 4, 3]], [[2, 3, 4]], [[2, 4, 3]], [[1, 2, 3, 4]], [[1, 2, 4, 3]], [[1, 3, 2, 4]], [[1, 3, 4, 2]], [[1, 4, 2, 3]], [[1, 4, 3, 2]]]

As you can see, in both versions I first generated all permutations of S_n by using the simple idea inductively that one gets all of the permutations of S_n by inserting k into all $(a_1, a_2, ..., a_{k-1}) \in S_{k-1}$ at all k possible locations. The "tricky" part is the ordering of the permutations in disjoint cycle notation, which you can see in the examples above. After racking my brains for some time I eventually came up with the solution you can see in the program, viz. I ordered them by their group order in the first place and by the number of elements that are not fixpoints in the second place, both in ascending order.

By the way, many programmers would have used recursive programming here, which often results in very short and elegant programs that are also very readable into the bargain. Even though I usually prefer the "iterative way of programming" as demonstrated above, because as a rule of thumb the overall performance in terms of memory usage and speed is better this way.

There are a lot of nice applications of our routines when dealing with permutation groups, which play an important role in abstract algebra. (In fact, by a well-known theorem every finite group G can be found among the subgroups of S_n, where n is the order of G.) As a small example we give the operation table of S_3 below

```
VECTOR(VECTOR(mulperms(a, b), b, symgroup(3)), a, symgroup(3))
```

[]	[[1, 2]]	[[1, 3]]	[[2, 3]]	[[1, 2, 3]]	[[1, 3, 2]]
[[1, 2]]	[]	[[1, 3, 2]]	[[1, 2, 3]]	[[2, 3]]	[[1, 3]]
[[1, 3]]	[[1, 2, 3]]	[]	[[1, 3, 2]]	[[1, 2]]	[[2, 3]]
[[2, 3]]	[[1, 3, 2]]	[[1, 2, 3]]	[]	[[1, 3]]	[[1, 2]]
[[1, 2, 3]]	[[1, 3]]	[[2, 3]]	[[1, 2]]	[[1, 3, 2]]	[]
[[1, 3, 2]]	[[2, 3]]	[[1, 2]]	[[1, 3]]	[]	[[1, 2, 3]]

(If you feel like trying out "bigger" examples, it might be a good idea to switch off the wrapping of new expressions in the menu "options" to see those tables in all their glory!)

There is also a nice connection of permutation groups to our Stirling numbers of the first kind. To be more precise, STIRLING1(n,k) is exactly the number of permutations in S_n having k cycles. (Note that also cycles of length 1, which are usually left out in the disjoint cycle representation, have to be counted here!!) Using the routines above, it's easy to check this (cf. DNL #34, p33). There are a lot of other relations, which you could investigate, e.g. the probability that a random permutation of size n has all its cycles of different lengths is asymtotic to $e^{-\gamma}$ as $n \to \infty$. (Here $\gamma = 0.5772...$ is the Euler-Mascheroni constant, which is one of the most important mathematical constants, known as euler_gamma in Derive.)

Please, tell me, if you know of other nice applications or if you have any suggestions concerning the programs above! (j.wiesenbauer@tuwien.ac.at)

Curvature and Evolute with *DERIVE*

Felix Schumm, Germany

In [1] and [2] we find for the curvature of a function given in parameter form [$x(t)$, $y(t)$]:
(The radius $r(t)$ of curvature is its reciprocal value).

$$\kappa[\alpha](t) = \frac{x'(t)\,y''(t) - x''(t)\,y'(t)}{\left(x'^2(t) + y'^2(t)\right)^{3/2}}$$

#1: [x(t) :=, y(t) :=]

#2: $$r(t) := \frac{\left(\left(\frac{d}{dt} x(t)\right)^2 + \left(\frac{d}{dt} y(t)\right)^2\right)^{3/2}}{\left(\frac{d}{dt} x(t)\right)\cdot\left(\frac{d}{dt}\right)^2 y(t) - \left(\left(\frac{d}{dt}\right)^2 x(t)\right)\cdot\frac{d}{dt} y(t)}$$

#3: $$\left[c(t) := \frac{\frac{d}{dt} x(t)}{\sqrt{\left(\left(\frac{d}{dt} x(t)\right)^2 + \left(\frac{d}{dt} y(t)\right)^2\right)}},\ s(t) := \frac{\frac{d}{dt} y(t)}{\sqrt{\left(\left(\frac{d}{dt} x(t)\right)^2 + \left(\frac{d}{dt} y(t)\right)^2\right)}}\right]$$

Center of curvature [xc(t), yc(t)]:

#4: [xc(t) := x(t) − r(t)·s(t), yc(t) := y(t) + r(t)·c(t)]

First example: the Hyperbola [x(t) = t, y(t) = 1/t]:

#5: $$\left[x(t) := t,\ y(t) := \frac{1}{t}\right]$$

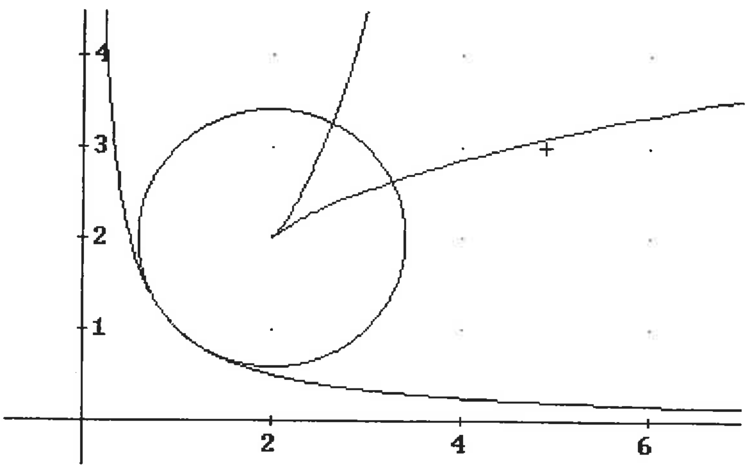
#6: $$[xc(t),\ yc(t)] = \left[\frac{3\cdot t^4 + 1}{2\cdot t^3},\ \frac{t^4 + 3}{2\cdot t}\right]$$

#7: $$\left[t,\ \frac{1}{t}\right]$$

for t = 1:

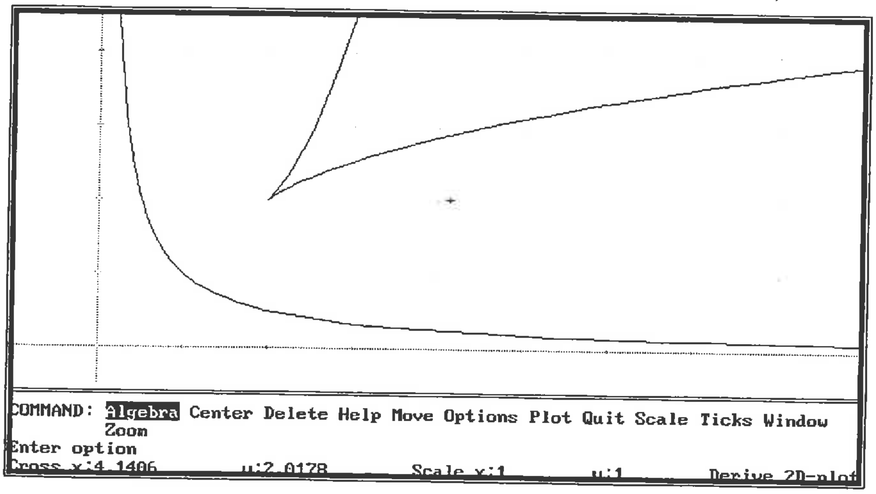
#8: $[xc(1),\ yc(1),\ r(1)] = [2,\ 2,\ \sqrt{2}]$

#9: $[2 + \sqrt{2}\cdot COS(t),\ 2 + \sqrt{2}\cdot SIN(t)]$



[1] Gray Alfred, Differentialgeometrie, Spektrum Akademischer Verlag Heidelberg 1994, ISBN 3-86025-141-4
[2] Rovenski Vladimir, Geometry of Curves and Surfaces with MAPLE, Birkhäuser, Basel 2000, ISBN 0-8176-4074-6

In a sentimental mood I reproduced the DERIVE 2.xx screenshot (loaded from a 5.25" diskette!). It looks so nice to have a DERIVE 2 - DOS screen within a DERIVE 5 worksheet, doesn´t it?. Josef



Second example: the Ellipse [x(t) = 5 cos(t), y(t) = 3 sin(t)]:

#10: $[x(t) := 5\cdot\text{COS}(t),\ y(t) := 3\cdot\text{SIN}(t)]$

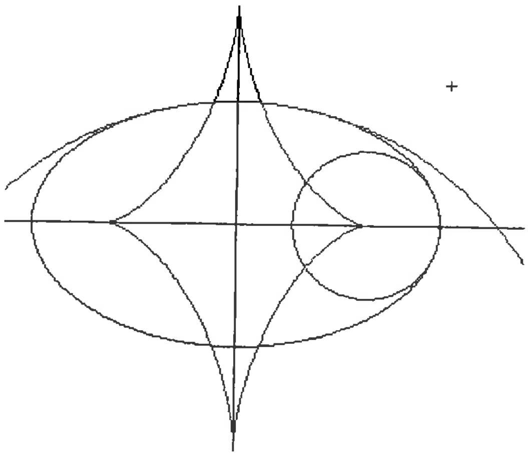
#11: $[xc(t),\ yc(t)] = \left[\dfrac{16\cdot\text{COS}(t)^3}{5},\ -\dfrac{16\cdot\text{SIN}(t)^3}{3}\right]$

We "invent" a function for the circle of curvature w.r.t parameter t:

#12: osc_circle(t_) := [xc(t_) + r(t_)·COS(t), yc(t_) + r(t_)·SIN(t)]

#13: osc_circle(0)

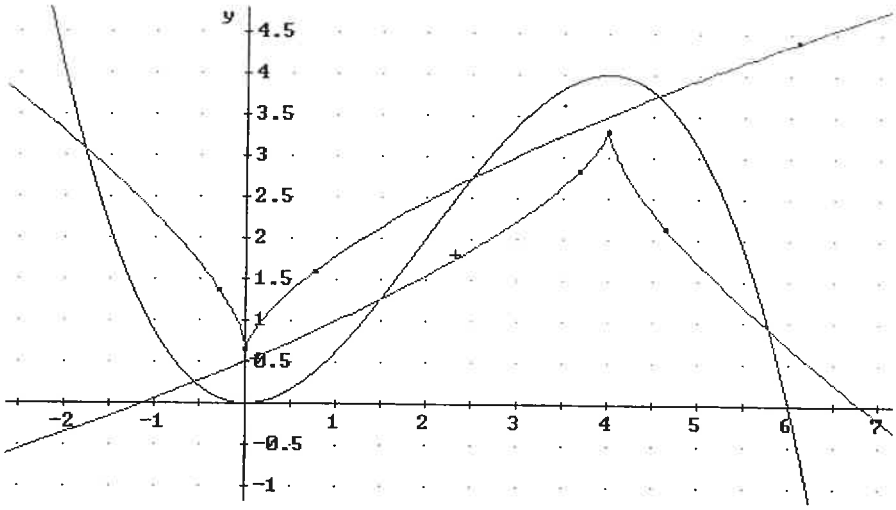
#14: $\text{osc\_circle}\left(\dfrac{\pi}{2}\right)$



One more function to compare with later results: y(x) = -x^3 / 8 + 3 x^2 / 4.
In parameter form: [x(t) = t, y(t) = -t^3/8 + 3t^2/4]

#15: $\left[x(t) := t,\ y(t) := -\dfrac{t^3}{8} + \dfrac{3\cdot t^2}{4}\right]$

#16: $[xc(t),\ yc(t)] = \left[\dfrac{13}{2\cdot(t-2)} - \dfrac{9\cdot t^5}{128} + \dfrac{45\cdot t^4}{64} - \dfrac{63\cdot t^3}{32} + \dfrac{9\cdot t^2}{16} + \dfrac{13\cdot t}{8} + \dfrac{13}{4},\ -\dfrac{13}{3\cdot(t-2)} - \dfrac{5\cdot t^3}{16} + \dfrac{15\cdot t^2}{8} - \dfrac{3\cdot t}{4} - \dfrac{3}{2}\right]$



Krümmung als Grenzwert - *Curvature as Limit*

Karl-Heinz Keunecke, Kiel, Germany

Das Unterrichtskonzept - *The didactical concept*

Über „Gefährliche Kurve" aus dem Strassenverkehr gewinnt man schnell den Einstieg in das Thema. Ein Auto, das eine Kurve durchfährt, darf diese nicht schneiden, sondern hat sie nur zu „berühren". Für die kurze Zeit, in der der Einschlag des Steuerrades nicht geändert wird, bewegt sich der Wagen auf einem Kreisbogen.

Damit sind bereits alle Begriffe zur Verfügung gestellt, die benötigt werden, um mit Hilfe des Radius des Krümmungskreises *r* bzw. durch $k = 1/r$ ein Mass für die Krümmung eines Funktionsgraphen zu definieren.
Um mit Schülerinnen und Schülern den Krümmungskreis zu konstruieren, geht man am besten zunächst von einem Kreis aus, bei dem der Mittelpunkt als gemeinsamer Schnittpunkt aller Normalen gegeben ist. Der Krümmungskreis ist hier überall gleich und mit dem Funktionsgraphen identisch. Die Methode der Konstruktion des Mittelpunktes des Krümmungskreises durch die Schnittpunkte von Kurvennormalen läßt sich auch auf beliebige Funktion übertragen. Dies wird mit Abb.1 erläutert.

A road sign "Dangerous curve" can introduce to the problem. A car driving through a curve must not "cut" but osculate the road. For a short while when the steering wheel is in a certain position the car moves on an arc of a circle. From this discussion all the expressions are available to define the curvature of a function by means of the radius r of the osculating circle as $k = 1/r$.

We try to find a circle which osculates the given graph in A(5;f(5)). Starting with the normal line of graph f in A we find neighbouring normals for x_i = 6, 5.75, 5.5, 5.25 and their intersection points with the normal line for $x = 5$. $x_4 = 5.25$ leads to center M_4 and this circle seems to be at least a good approximation for the desired circle. (Figure 1)

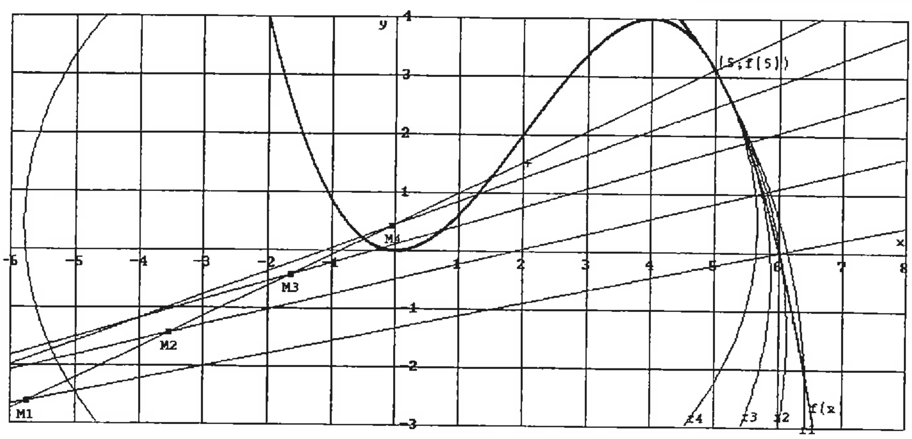


Figure 1

Es soll ein Kreis gefunden werden, der den Funktionsgraphen an der Stelle $x = 5$ berührt. Dazu wird zunächst in dem Punkt A(5;f(5)) die Normale zu dem Graphen von f und anschließend werden weitere Normale an den Stellen 5,25; 5,5; 5,75 und 6 gezeichnet. Die Schnittpunkte mit der Normalen durch A sind M_1 bis M_4. Nun werden Kreise um die Punkte M_1 bis M_4 jeweils durch den Punkt A gezeichnet. Während die Kreise mit r_1 und r_2 den Graphen noch mehrfach schneiden, so scheint der Kreis mit r_4 bereits etwa die gewünschte Eigenschaft zu haben. Man wird vermuten, dass die Näherung umso besser wird, je dichter die Normalen an die feste Normale durch A heranrücken. Hieraus kann nun im Unterricht eine Strategie für das weitere Vorgehen entwickelt werden.

1. Gleichung für die Normalen an den Stellen a und $a + h$ aufstellen.
2. Schnittpunkt dieser beiden Normalen bestimmen.
3. Untersuchen, ob es einen Grenzwert der Punktfolge für $h \to 0$ gibt.
4. Falls die Folge konvergiert, den Radius des Krümmungskreises und die Krümmung berechnen.

1. Find the equations of the perpendicular lines on places a and $a + h$.
2. Calculate the intersection points.
3. Investigate if there is a limit for the sequence of the intersection points for $h \to 0$.
4. In case of convergence calculate the radius of the osculating circle and its reciprocal value - the curvature.

First we will investigate the curvature of given functions to proceed in generalizing the procedure to obtain the well known formulae for the radius of the osculating circle and the curvature.

Mit diesem Konzept wird zunächst die Krümmung gegebener Funktionen an vorgegebenen Stellen mit Hilfe von Grenzprozessen untersucht. Das Verfahren lässt sich aber auch auf eine beliebige Funktion $f(x)$ und die Stelle $x = a$ anwenden. Dann gewinnt man die bekannten Formel für Krümmung und Krümmungskreis, die schließlich zur Bearbeitung praktischer Probleme genutzt werden .

Realisierung der UE mit *DERIVE* - *Realisation of the Teaching Unit with DERIVE*

Bei Benutzung von *DERIVE5* kann jeder Schüler ein eigenes Arbeitsblatt anlegen, in dem alle Definitionen und Befehle individuell kommentiert werden können. Außerdem kann jede Graphik in dieses Worksheet eingebunden werden. Es ist also im Grunde keine zusätzliche Mitschrift in einem Heft erforderlich, denn der Ausdruck des entsprechenden dfw-Files liefert alle erforderlichen Informationen einschließlich der Hausaufgaben mit den Lösungen.

#1: $f(x) := -\frac{x^3}{8} + \frac{3}{4} \cdot x^2$

#2: $norm(x,a) := -\frac{1}{f'(a)} \cdot (x-a) + f(a)$

#3: $nnorm(x,a,d) := -\frac{1}{f'(a+d)} \cdot (x-a-d) + f(a+d)$

#4: VECTOR(norm(x,-1+0.1·n),n,0,31)

#5: xsc(a,d):=RHS(SOLVE(norm(x,a)=nnorm(x,a,d),x))

#6: ysc(a,d):=norm(xsc(a,d),a)

#7: $\text{VECTOR}\left(\text{norm}\left(x, 5+\frac{1}{n}\right), n, 1, 20\right)$

#8: $\text{VECTOR}\left(\left[\text{xsc}\left(5, \frac{1}{n}\right), \text{ysc}\left(5, \frac{1}{n}\right)\right], n, 1, 50\right)$

#9: norm(x,5)

#10: $\left[xs(a) := \lim_{h\to 0} xsc(a,h), ys(a) := \lim_{h\to 0} ysc(a,h)\right]$

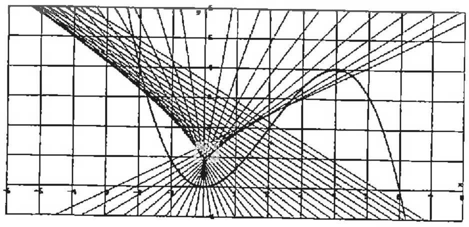


Figure 2 (derived from expression #4)

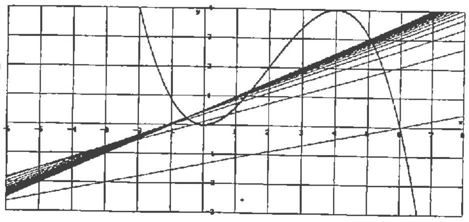


Figure 3 (derived from expression #7)

#11: $rs(a) := \sqrt{(a-xs(a))^2 + (f(a)-ys(a))^2}$

#12: $osc\_c(a) := (x-xs(a))^2 + (y-ys(a))^2 = rs(a)^2$

#13: [xs(5),ys(5),rs(5)]=[1.236,1.118,4.264]

Leider wird beim Aufruf von `norm(x,a+h)` im allgemeinen Fall der Term `f'(a+h)` nur als `f'(a)` ausgewertet. Daher ist für die benachbarte Normale eine zweite Funktion `nnorm(x,a,d)` notwendig. Die Berechnung der Koordinaten des Schnittpunktes erfolgt in den Zeilen #5 und #6. In #11 wird der Radius des Krümmungskreises ermittelt. Die Ergebnisse an der Stelle $x = 5$ finden sich anschließend.

Zunächst erhalten die Schülerinnen und Schüler mit Zeilen 1 und 2 die Möglichkeit, zum Graphen der Funktion f die Normalen zu konstruieren. Damit können sie die Lage der Schnittpunkte der Normalen zu untersuchen. Eine Normalenschar wird z.B. mit Zeile #4 erzeugt, die in Abb.2 dargestellt wird.

Darin kann aus den Schnittpunkten benachbarter Normalen auf die ungefähre Lage der Mittelpunkte der Krümmungskreise und deren Radien geschlossen werden. Die zugehörige Krümmung ist im Nullpunkt am größten und bei $x = 2$ hat sie den Wert 0, da hier die Normalen parallel verlaufen.

Um die Schnittpunkte algebraisch zu bestimmen wird die Gleichung

`norm(x,a) = nnorm(x,a,d)` für $d \neq 0$

in #5 gelöst und in #6 der zugehörige y-Wert berechnet. Mit $d = 1/n$ werden in #7 für $a = 5$ und für $n = 1,...,20$ die zugehörigen Normalen berechnet. Aus der graphischen Darstellung in Abb.3 kann man vermuten, dass die Schnittpunkte der Normalenschar mit der Normalen an der Stelle $a = 5$ einem Grenzwert zustreben. Dies erkennt man noch deutlicher, indem man für die gleichen Bedingungen nur die Punktfolge $\left[xsc\left(5,\frac{1}{n}\right), ysc\left(5,\frac{1}{n}\right)\right]$ einzeichnet, wie es in Abb.4 geschehen ist.

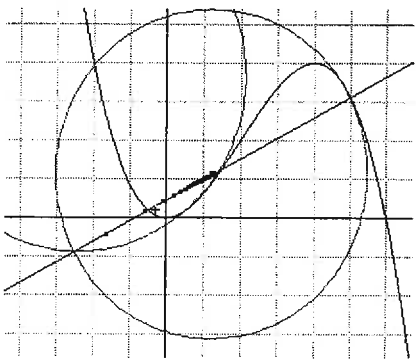


Figure 4

Als nächster Schritt sind nun die Grenzwerte für `xsc(a,d)` und `ysc(a,d)` zu berechnen (#10) Man erhält:

$$(xs; ys) \approx (1.237; 1.118)$$

Aus dem Abstand dieses Punktes von A errechnet man (#11) den Radius des Krümmungskreises $r_s \approx 4.265$. In #12 wird die Gleichung des Krümmungskreises als Funktion von a festgelegt. Er ist in Abb.4 eingezeichnet worden und ist somit ein für Schülerinnen und Schüler wichtiger Nachweis, dass die durchgeführten Berechnungen richtig sind.

Unfortunately f'(a+h) doesn´t evaluate in an expression using a generic function f(x) correctly. It appears as f'(a). That is the reason that nnorm(x,a,d) must be defined The intersection point is calculated in #5 and #6, the radius of the circle is result of #11.

The students are encouraged to produce a family of perpendicular lines (expr #4, Figure 2). Using a sequence for h (eg k = 1/n) with a = 5 they create a family of perpendicular lines approaching the normal line in a = 5.(expr #7, Figure 3) They can estimate the position of the "last" intersection point and confirm their conjectures by evaluating and plotting the sequence of points in #8 (figure 4). The next step is performing the limit process in #10, observing the convergence, followed by calculating the radius of the osculating circle (#11). #12 generates the osculating circle as a function of a.

The sequence of points may be divergent. Approaching a = 2 using norm(x,2+h) or creating a point list similar to that given in #7 results in infinite large numbers. The students should discover that in a point of inflection the radius of the osculating circle is infinite, thus curvature is 0.

Having worked out sufficient examples one can proceed generalizing the problem. Working with a generic function f(x) and an arbitrary position a delivers results in # 20 and #21 immediately.

Nicht immer sind die entstehenden Punktfolgen konvergent. Nähert man der Normalen im Wendepunkt $a = 2$ eine Normale `nnorm(x,2+h)` immer mehr an, so entfernt sich der Schnittpunkt der beiden Normalen immer mehr. Das ist bereits aus Abb.3 zu vermuten, weil an dieser Stelle die Normalen offensichtlich parallel verlaufen.

Eine Punktliste ähnlich #8 mit $a = 2$ anstelle von $a = 5$ - womöglich mit einer stärker konvergierenden Folge für k - ist ein deutlicher Hinweis auf die Divergenz

$$\begin{bmatrix} 1.300000000 \cdot 10^{20} & -8.666666666 \cdot 10^{19} \\ 1.300000000 \cdot 10^{21} & -8.666666666 \cdot 10^{20} \end{bmatrix}$$

Wenn eine ausreichende Anzahl von Beispielen im Unterricht bearbeitet worden ist, dann kann die Bestimmung der Krümmung für eine beliebige Funktion f an der Stelle a nach der gleichen Methode durchgeführt werden. Dies wird anschließend gezeigt.

Von diesen Lösungstermen sind nun die Grenzwerte für $h \to 0$ zu berechnen. Das leistet *DERIVE* ebenso wie viele andere CAS [1] nicht. Bekanntlich sind diese Terme so umzuformen, dass die Differenzenquotienten von f und f' auftreten, deren Grenzwerte dann f' und f'' sind. Der folgende Term zeigt nochmals die Rechnerlösungen für `xsc(a,h)` und `ysc(a,h)`.

```
#19:  f(x) :=

                    f'(a)·(f'(a + h)·(f(a + h) - f(a)) + a + h) - a·f'(a + h)
#20:  xsc(a, h) = ─────────────────────────────────────────────────────────────
                                     f'(a) - f'(a + h)

                              f'(a + h)·(f(a + h) - f(a)) + h
#21:  ysc(a, h) = f(a) - ─────────────────────────────────────
                                   f'(a) - f'(a + h)
```

Daraus erhält man nach der beschriebenen Umformung für die *x*-Koordinate:

DERIVE (and other CAS) are unable to discover the derivatives in these expressions - so rediscover your pre-DERIVE manipulating skills to rewrite expr #20 and #21 to calculate the limits leading to the center coordinates of the circle and its radius - the last step can be done supported by DERIVE.

$$\frac{a\,\dfrac{f(a+h)-f(a)}{h}-\left(f(a+h)\cdot f(a)\dfrac{f(a+h)-f(a)}{h}+\dfrac{f(a)\cdot h}{h}\right)}{\dfrac{f(a+h)-f(a)}{h}}$$

Der Grenzwert ist dann:

$$xk(a)=\lim_{h\to 0} xsc(h)=a-\frac{\left(f'^2(a)+1\right)\cdot f'(a)}{f''(a)}$$

Für die zugehörige Ordinate erhält man

$$yk(a)=f(a)+\frac{f'^2(a)+1}{f''(a)}$$ und für die Krümmung $$kr=\frac{f''(a)}{\left(\sqrt{1+f'^2(a)}\right)^3}.$$

Für diese Rechnungen kann man bereits wieder das CAS verwenden.

```
                             2
                     (f'(a)  + 1)·f'(a)
#24:  xc(a) := a - ────────────────────
                          f''(a)

                             2
                        f'(a)  + 1
#25:  yc(a) := f(a) + ────────────
                          f''(a)

                                  2                2
#26:  rc(a) := √((a - xc(a))  + (f(a) - yc(a)) )

                ⎛          2     3 ⎞
                ⎜ (f'(a)  + 1)     ⎟
#27:  rc(a) = √ ⎜ ──────────────── ⎟
                ⎜            2     ⎟
                ⎝     f''(a)       ⎠

                    3
                   x      3  2
#28:  f(x) := - ──── + ───·x
                   8      4

#29:  rc(5) = 4.264
```

Literatur:

[1] Keunecke, Karl-Heinz: Behandlung der Begriffe Folge, Grenzwert und Ableitung am Beispiel der Krümmung. Praxis der Mathematik 5/42, Oktober 2000

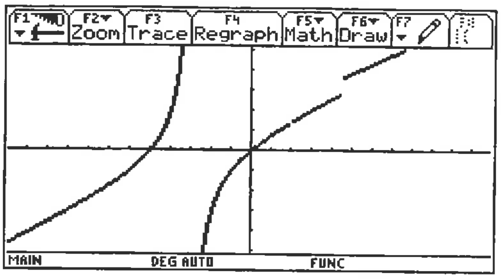
From Pole to Pole
on the TI-92 or TI-89´s Data-Matrix Editor.

Josef Böhm, Würmla

The TI´s `Data/Matrix`-editor is an excellent tool to support concepts of accumulation points, limits, continuity, average and instantaneous rates of change.

I´ll demonstrate the possibilities using an "exotic" function.

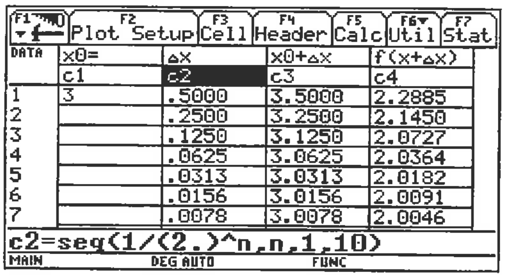
$$f(x) = \frac{x^2 - 4}{2x - 4} + \frac{3\,\mathrm{sign}\,(2x - 9)}{x + 3}$$



Which values for x are excluded from the domain?

To keep the exploration as general as possible I recommend to store the function as `f(x)` in the Home Screen. Then open a new Data Sheet via the `Data/Matrix` Editor, say `analysis`.

1 Continuity Behaviour at $x_0 = 3$



DATA	x0=	Δx	x0+Δx	f(x+Δx)
	c1	c2	c3	c4
1	3	.5000	3.5000	2.2885
2		.2500	3.2500	2.1450
3		.1250	3.1250	2.0727
4		.0625	3.0625	2.0364
5		.0313	3.0313	2.0182
6		.0156	3.0156	2.0091
7		.0078	3.0078	2.0046

c2=seq(1/(2.)^n,n,1,10)

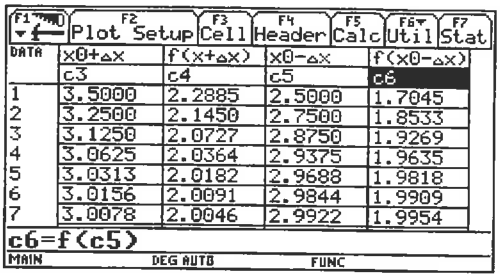
In cell `c1` we enter the value fo x_0. In column `c2` we define a 0-sequence to obtain in `c3` a sequence of x-values with rightsided limit 3. Subsequently in c4 we can see the sequence of function values which should tend to $f(x_0)$.

(Experiment changing the sequence in `c2`).

For `c3` enter: `c1[1]+c2`
For `c4` enter: `f(c3)`

In columns `c5` and `c6` we generate the left sided approach.

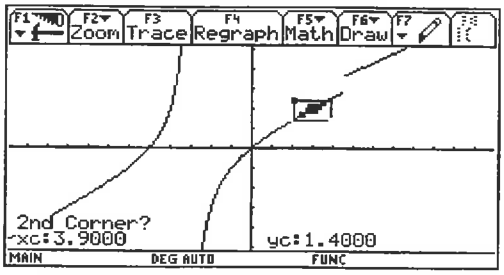
How should we change `c2` to accelerate the convergence?

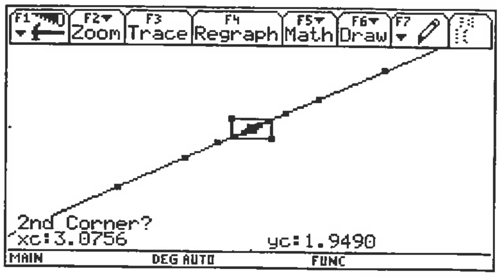


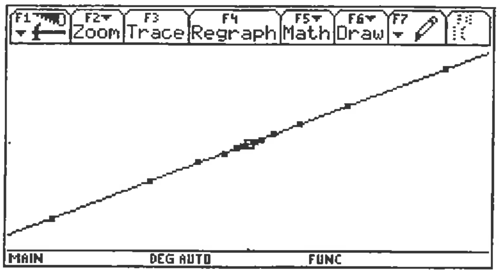
DATA	x0+Δx	f(x+Δx)	x0−Δx	f(x0−Δx)
	c3	c4	c5	c6
1	3.5000	2.2885	2.5000	1.7045
2	3.2500	2.1450	2.7500	1.8533
3	3.1250	2.0727	2.8750	1.9269
4	3.0625	2.0364	2.9375	1.9635
5	3.0313	2.0182	2.9688	1.9818
6	3.0156	2.0091	2.9844	1.9909
7	3.0078	2.0046	2.9922	1.9954

c6=f(c5)

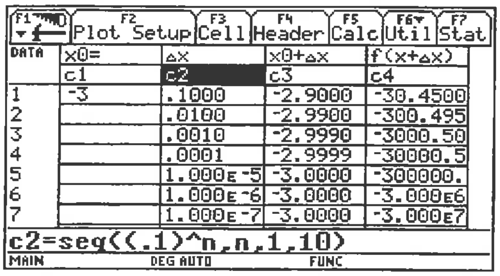
Using the plot-facilities together with zooming in we can visualize the convergence process:





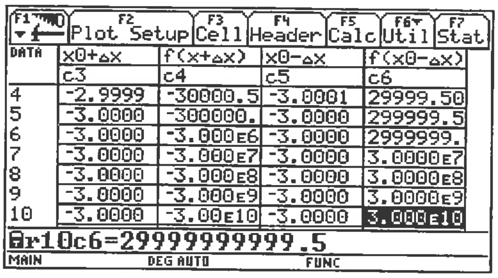


2 Continuity Behaviour at $x_0 = -3$



DATA	x0=	Δx	x0+Δx	f(x+Δx)
	c1	c2	c3	c4
1	-3	.1000	-2.9000	-30.4500
2		.0100	-2.9900	-300.495
3		.0010	-2.9990	-3000.50
4		.0001	-2.9999	-30000.5
5		1.000E-5	-3.0000	-300000.
6		1.000E-6	-3.0000	-3.000E6
7		1.000E-7	-3.0000	-3.000E7

c2=seq((.1)^n,n,1,10)

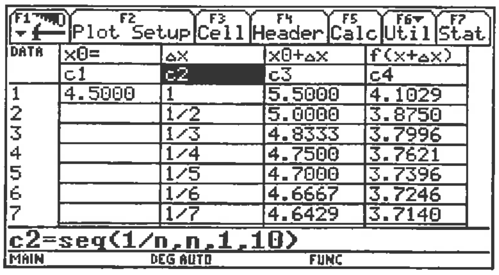


DATA	x0+Δx	f(x+Δx)	x0-Δx	f(x0-Δx)
	c3	c4	c5	c6
4	-2.9999	-30000.5	-3.0001	29999.50
5	-3.0000	-300000.	-3.0000	299999.5
6	-3.0000	-3.000E6	-3.0000	2999999.
7	-3.0000	-3.000E7	-3.0000	3.0000E7
8	-3.0000	-3.000E8	-3.0000	3.0000E8
9	-3.0000	-3.000E9	-3.0000	3.0000E9
10	-3.0000	-3.00E10	-3.0000	3.000E10

r10c6=29999999999.5

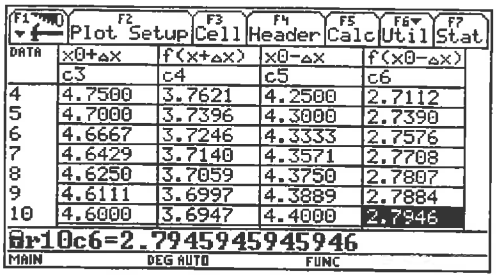
The results in columns c4 and c6 are very informative!! We can see what happens at a pole.

3 Continuity Behaviour at $x_0 = 4.5$



DATA	x0=	Δx	x0+Δx	f(x+Δx)
	c1	c2	c3	c4
1	4.5000	1	5.5000	4.1029
2		1/2	5.0000	3.8750
3		1/3	4.8333	3.7996
4		1/4	4.7500	3.7621
5		1/5	4.7000	3.7396
6		1/6	4.6667	3.7246
7		1/7	4.6429	3.7140

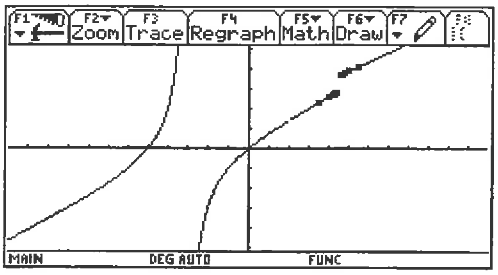
c2=seq(1/n,n,1,10)

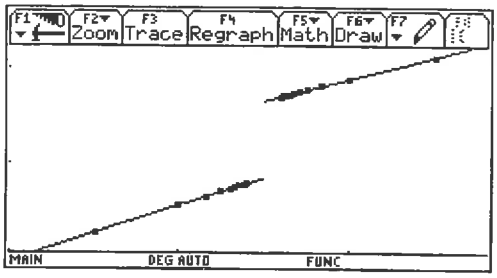


DATA	x0+Δx	f(x+Δx)	x0-Δx	f(x0-Δx)
	c3	c4	c5	c6
4	4.7500	3.7621	4.2500	2.7112
5	4.7000	3.7396	4.3000	2.7390
6	4.6667	3.7246	4.3333	2.7576
7	4.6429	3.7140	4.3571	2.7708
8	4.6250	3.7059	4.3750	2.7807
9	4.6111	3.6997	4.3889	2.7884
10	4.6000	3.6947	4.4000	2.7946

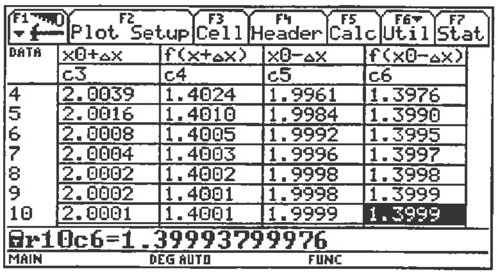
r10c6=2.7945945945946

Discuss the outcomes in columns c4 and c6. Change the sequence in c2. Discuss possible consequences. Produce a graphic representation.



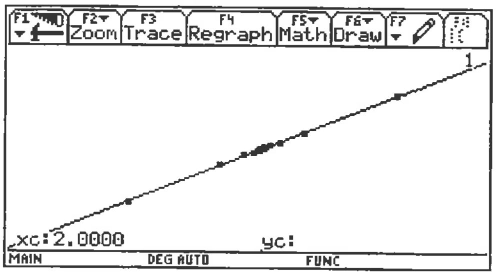


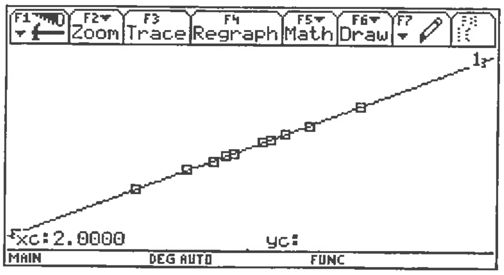
4 The last -and maybe most - interesting position is $x_0 = 2$



DATA	x0+Δx	f(x+Δx)	x0-Δx	f(x0-Δx)
	c3	c4	c5	c6
4	2.0039	1.4024	1.9961	1.3976
5	2.0016	1.4010	1.9984	1.3990
6	2.0008	1.4005	1.9992	1.3995
7	2.0004	1.4003	1.9996	1.3997
8	2.0002	1.4002	1.9998	1.3998
9	2.0002	1.4001	1.9998	1.3999
10	2.0001	1.4001	1.9999	1.3999

r10c6=1.39993799976





Discuss **all** the messages given on the screen.

Remember the hole in the function graph.

So we found a pole
a jump
and a hole
as different kinds of discontinuities

5 A numerical approach to the differential quotient.

Now it is easy to extend this table for further use in calculus teaching. We only have to add some columns for the absolute changes and then for the rates of change leading to the average rate of change and to its limit.

Let´s have another "artificial" function. Do you know the "**Marilyn Monroe Curve**"?

$$f(x) = \left(\frac{|x|}{2} - 2\right)^2$$

$x_0 = 2$:

DATA	rs.af	rs.af/Δx	ls.af	ls.af/Δx
	c7	c8	c9	c10
4	-.0615	-.9844	-.0635	-1.0156
5	-.0310	-.9922	-.0315	-1.0078
6	-.0156	-.9961	-.0157	-1.0039
7	-.0078	-.9980	-.0078	-1.0020
8	-.0039	-.9990	-.0039	-1.0010
9	-.0020	-.9995	-.0020	-1.0005
10	-.0010	-.9998	-.0010	-1.0002

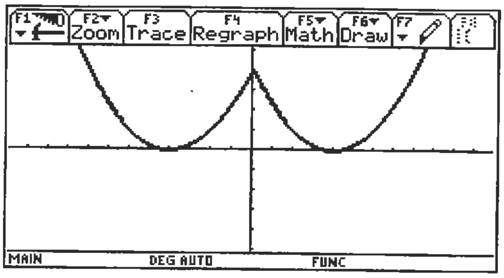
c7=c4-f(c1[1])

$x_0 = 0$:

DATA	rs.af	rs.af/Δx	ls.af	ls.af/Δx
	c7	c8	c9	c10
4	-.1240	-1.9844	.1240	1.9844
5	-.0623	-1.9922	.0623	1.9922
6	-.0312	-1.9961	.0312	1.9961
7	-.0156	-1.9980	.0156	1.9980
8	-.0078	-1.9990	.0078	1.9990
9	-.0039	-1.9995	.0039	1.9995
10	-.0020	-1.9998	.0020	1.9998

r10c10=1.9997558593536

Do you recognize the difference in the behaviour of the rate of changes progress?
Use various sequences for approaching $x = 0$. Is the function continuous for $x = 0$?



Further explorings:

Try another function, eg $f(x) = -\frac{x^3}{8} + \frac{3x^2}{4}$. Perform the same investigation like above, but finally enter t for the location in c1[1]:

DATA	ls.af(x)	ls.af/Δx	expanded
	c9	c10	c11
5	-.000004...	-.375000...	-.375000...
6	-3.75000...	-.375000...	-.375000...
7	-3.75000...	-.375000...	-.375000...
8	-3.75000...	-.375000...	-.375000...
9	-3.75000...	-.375000...	-.375000...
10	-3.75000...	-.375000...	-.375000...
11			

r10c11=-.375*t^2+1.500000000...

Attach one more column to expand the results from c10 or c8.

Copy and paste content of cell r10c11 into the Home Screen and try to make conclusions for a general rule to find a formula for the instantaneous rate of change.

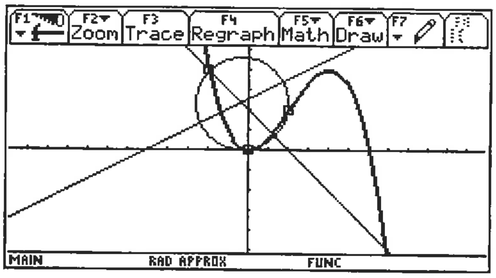
If you cannot copy the content in earlier *TI*-versions then switch to the Home Screen and enter analysis[11][10] to see the result.

```
■ -.375·t^2 + 1.5000000000375·t - 7.500000▸
                  -.375·t^2 + 1.500·t - 7.500E-11
...000375*t-7.5000000001249E-11
MAIN            RAD APPROX          FUNC 5/30
```

Describe also the "last" elements in columns c4 and c6 - repeating and deepening continuity.

6 Curvature on the Spreadsheet

It is a challenge to find the circle of curvature in this way, too, i.e without treating the problem in the usual way determining the center of the osculating circle as the limit position of the intersection point of two neighbouring normals of the curve. I want to extend the Δx approach - without assuming the existence of a derivative, so I cannot set up the equation of a normal (compare K.H. Keunecke's contribution).

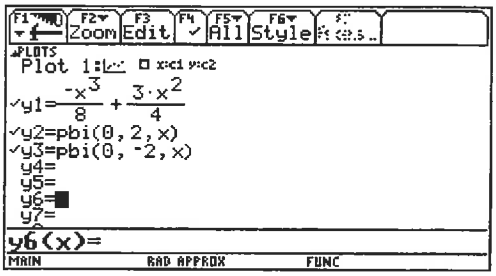


We will take the function from before and make further use of our created data table `analysis`. We want to find the osculating circle on locations $x_0 = 0$, $x_0 = 5$ and generalized on $x_0 = t$.

Take three points on the graph $(x_0-\Delta x, f(x_0-\Delta x))$, $(x_0, f(x_0))$ and $(x_0-\Delta x, f(x_0-\Delta x))$ and find the circumcircle. If h tends to 0 the circle will have three points in common with the graph - the osculating circle.

Where is the center of this circle? What is its radius?

The center is the limit point of the intersection point of the two perpendicular bisectors (see the sketch). Again we extend the data table. But now we make use of the power of a CAS. Define a function `pbi(xØ,h,x)` for the perpendicular bisector and calculate the first coordinate of the intersection point. Store the result as `xco(xØ,h)` (h is equivalent to Δx).




```
-x^3/8+3*x^2/4 → f(x)

-h/(f(xØ+h)-f(xØ))*(x-
(2*xØ+h)/2)+(f(xØ)+f(xØ+h))/2→pbi(xØ,h,x)

zeros(pbi(xØ,h,x)-pbi(xØ,-h,x),x)[1]

ans(1) → xco(xØ,h)
```

We could now switch to the `Data/Matrix` Editor and attach two more columns `c12` and `c13`. To save memory and computation time as well it is better to open a new data table `curv`.

DATA	x0	Δx	xCO	yCO	radius
	c1	c2	c3	c4	c5
1	0	1.000	.038	1.052	1.053
2		.250	.005	.690	.690
3		.111	.001	.671	.671
4		.063	3.2E-4	.668	.668
5		.040	1.3E-4	.667	.667
6		.028	6.4E-5	.667	.667
7		.020	3.5E-5	.667	.667

c2=seq(1./n^2,n,1,10)

Table for $x_0 = 0$ (`c1[1] = Ø`)

The header of column c3 is c3 = `xco(c1[1],c2)` and the header of `c4` is `pbi(c1[1],c2,c3)`.

In `c5` you find the radius of the osculating circle (distance between point on curve and center point). All values in `c3`, `c4` and `c5` seem to converge (0, 2/3, 2/3).

DATA	x0	Δx	xCO	yCO	radius
	c1	c2	c3	c4	c5
1	5	.500	1.342	.925	4.268
2		.250	1.263	1.070	4.265
3		.125	1.243	1.106	4.265
4		.063	1.239	1.115	4.265
5		.031	1.237	1.117	4.265
6		.016	1.237	1.118	4.265
7		.008	1.237	1.118	4.265

c2=seq((.5)^n,n,1,10)

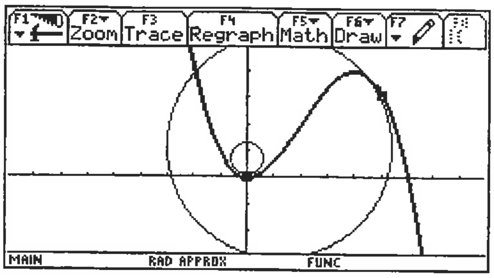
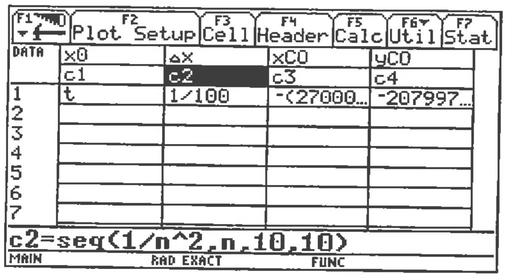
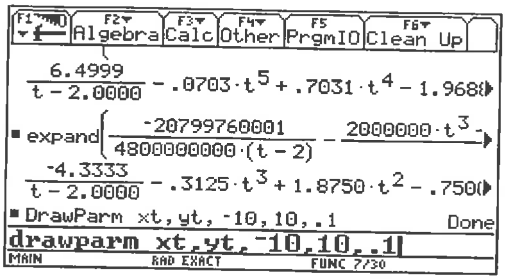


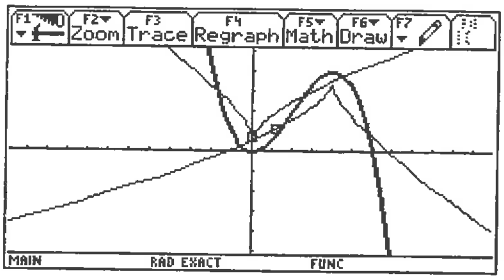
Table for $x_0 = 5$ (with `c1[1] = 5`) and the graph together with the approximations of the circles.

Now we will start completely generalized calculation and enter `t` in `c1[1]`. I changed the sequence to only make visible the last elements and inspected the outcomes in the Home Screen.





For switching here and back between Home Screen and `Data/Matrix` Editor you should switch Auto Calculation Off. (Compare with the exact results from page 26).



Now it should be a challenge for the students to investigate the behaviour of this curve for $t = 2$. Without having mentioned an "inflection point" ever before they have the chance to explore this very special point of a graph on their own.

Find the asymptote of the "evolute".

This is the moment our roundtrip closes.

We started investigating discontinuities and we finish investigating a pole again but on a much higher level. And I think that we made an interesting journey into analysis. It is clear that we have not reached our destination point yet. Working with limits we obtain exact answers to confirm our numerical and graphical outcomes. (One can do this at the end of each part or at the end of the whole procedure repeating the process once more from the beginning from the analytical point of view).

It is my experience that students like to follow this very "inner mathematical" reasoning. They feel their adventures in their heads and feel a bit infected by the spirit of mathematics.

I want to give full credit to David Bowers who gave a marvellous workshop in San Francisco and in Liverpool as well showing so many possibilities how to use the TI´s Data/Matrix Editor in a very meaningful and manyfold way.

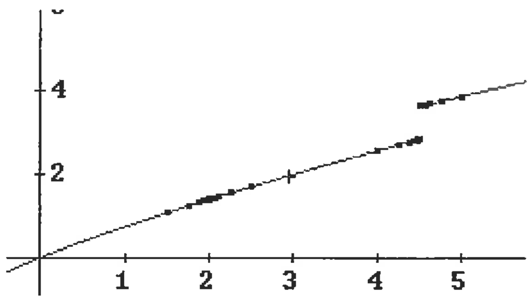
I also want to give credit to participants of the 1st T³ Winter Academy in Austria (1 - 6 January 2001), who gave the idea of "absolute adressing" a cell in the Data/Matrix Editor. (K.H.-Keunecke, DetlevKirmse, ManfredGrote).

I´d like to dedicate this paper to Detlev Kirmse who was attacked by an apoplexy of the brain after coming home from our meeting. All friends wish Detlev the best for his recovering and hope to discuss and to go skiing with him again next winter.

Additional comment: it is easy to transfer the whole paper to the DERIVE environment. Instead of using the spread sheet, one makes intensive use of the VECTOR command to produce the tables (or the TABLES command in DfW5). Some details are added. You can download the complete DfW5 file from the home pages of the DUG.

To give some impressions of the *DERIVE* version I add some sections from a DfW-file.

Investigation of discontinuities. (Unfortunately you cannot distinguish the different colours, right)

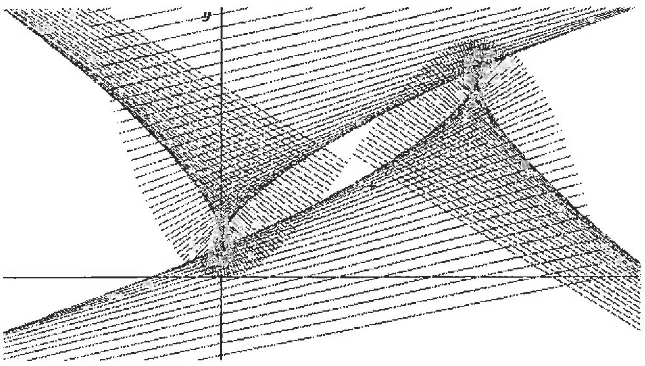


Numerical derivation of "our" cubic. (below)

#81: $f(x) := -\dfrac{x^3}{8} + \dfrac{3 \cdot x^2}{4}$

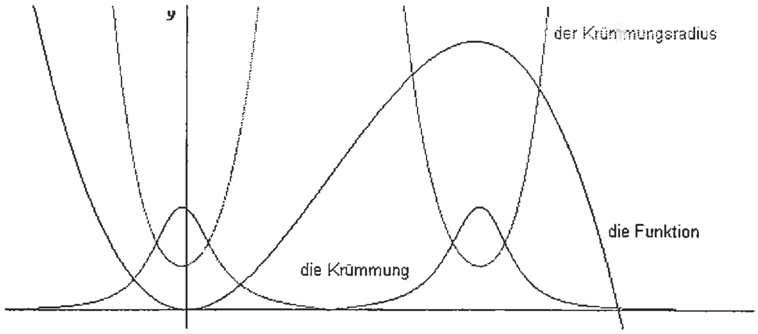
#82: $\text{EXPAND}(\text{diff\_sh}(f(x),\ x,\ x,\ 0.1^n,\ 19,\ 20,\ 1))$

#83: $$\begin{bmatrix} \Delta x & (f(x0+\Delta x)-f(x0))/\Delta x & (f(x0)-f(x0-\Delta x))/\Delta x \\ 10^{-19} & -0.375 \cdot x^2 + 1.499 \cdot x + 7.499 \cdot 10^{-20} & -0.375 \cdot x^2 + 1.5 \cdot x - 7.5 \cdot 10^{-20} \\ 10^{-20} & -0.375 \cdot x^2 + 1.499 \cdot x + 7.499 \cdot 10^{-21} & -0.375 \cdot x^2 + 1.5 \cdot x - 7.5 \cdot 10^{-21} \end{bmatrix}$$



The family of radii of the circles of osculation appears as a nice pattern of lines (left)

Add the function of the radius of curvature ("Krümmungsradius") and the function of the curvature ("Krümmung") und you receive a wonderful representation of the dependencies.



$$\text{EXPAND}\left(\text{TABLE}\left(\left[x0n\left(t, \frac{1}{n^2}\right), y0n\left(t, \frac{1}{n^2}\right)\right], n, 15, 20, 5\right)\right)$$

$$\begin{bmatrix} 15 & \dfrac{6.499}{t-2} - 0.07031 \cdot t^5 + 0.7031 \cdot t^4 - 1.968 \cdot t^3 + 0.5625 \cdot t^2 + 1.625 \cdot t + 3.249 & \dfrac{4.333}{t-2} - 0.3125 \cdot t^3 + 1.875 \cdot t^2 - 0.7500 \cdot t - 1.499 \\ 20 & \dfrac{6.499}{t-2} - 0.07031 \cdot t^5 + 0.7031 \cdot t^4 - 1.968 \cdot t^3 + 0.5625 \cdot t^2 + 1.625 \cdot t + 3.249 & \dfrac{4.333}{t-2} - 0.3125 \cdot t^3 + 1.875 \cdot t^2 - 0.7500 \cdot t - 1.499 \end{bmatrix}$$

Compare again with the results from page 26.

How to import Data from a Spreadsheet Program to CAS

Josef Böhm, Würmla

You can use `File Load Data` to load a Data Matrix into *DERIVE* for further use. In the best case you have nothing to do but loading the file - this is the case if the format of the given data fits perfect to the format requested by *DERIVE*. If the original file´s extension is .DAT or .TXT this might by worth a try. You will fail importing titles of columns and/or rows. I try loading a file called comptime.dat [1].
In all the following examples I truncated the files to save space.

$$\begin{bmatrix} 1 & 15 & 16 & 10 \\ 2 & 21 & 10 & 5 \\ 3 & 30 & 12 & 7 \\ 4 & 17 & 9 & 9 \end{bmatrix}$$

It worked perfect.
Now let´s assume that we created a Excel-worksheet and we want to proceed working in *DERIVE*.
The next example includes two "difficulties": we want to extract columns 2 and 3 and - unfortunately we worked on a German DUG member´s PC using commas, instead of decimal points.
One click to the *DERIVE* Online Help should help!

Each line of the data file must be a list of numbers separated by spaces and/or commas. The numbers must be in standard integer, rational, or floating point notation with an optional E or D followed by an exponent (e.g. -2.325E-7). DERIVE interprets each row of the data file as a row of a matrix, with blank lines used to separate matrices.

The Excel table

1	15,25	16,38	10
2	31,34	10,45	5
3	30,16	12,36	7

Highlight the inner two columns and copy and paste them to your text processor. Then save in text-format with extension .dat, say `comma.dat` and then load the data-file into *DERIVE*.

```
15,25    16,38
31,34    10,45
30,16    12,36
```

That´s how the table looks like in text-processing. program

$$\begin{bmatrix} 15 & 25 \\ 16 & 38 \\ 31 & 34 \\ 10 & 45 \\ 30 & 16 \\ 12 & 36 \end{bmatrix}$$

Data with Commas - from German versions of Excel - look very strange in *DERIVE*. Follow the *DERIVE* Online Help. Our German speaking users working with a Spreadsheet Program, which uses decimal commas or working with any other data files using commas, have to replace the commas by points applying the appropriate text processor´s procedure to do this in one single step. Save then as .dat file in **text format** and try again:
(You could also change the general Windows settings in Excel!)

```
15.25    16.38
31.34    10.45
30.16    12.36
```

$$\begin{bmatrix} 15.25 \\ 16.38 \\ 31.34 \\ 10.45 \\ 30.16 \\ 12.36 \end{bmatrix}$$

The data are ok now, but we lost their arrangement in rows and columns. According to the Online Help again, we do need the commas again to separate the data. Convert the data-table in your text-processing program to a text with a comma separating the cells. The text should look like that now. (Take care that there are no tabs or anything else left). Save and then load again to receive #4.

```
15.25,16.38
31.34,10.45
30.16,12.36
```

$$\#4: \quad \begin{bmatrix} 15.25 & 16.38 \\ 31.34 & 10.45 \\ 30.16 & 12.36 \end{bmatrix}$$

Next time we find in [2] a data table `sales.txt` and we would like to use it for a *DERIVE* session. We try loading `sales.txt` and we receive the error message that *DERIVE* is unable parsing something in the file. We open the file with an ordinary text editor and we can read:

```
Sales sq footage  Shopping centers
1.232000000e+002  1.600000000e+000  1.320000000e+001
2.115000000e+002  2.300000000e+000  1.750000000e+001
3.855000000e+002  3.000000000e+000  2.210000000e+001
4.751000000e+002  3.400000000e+000  2.550000000e+001
6.411000000e+002  3.900000000e+000  3.260000000e+001
```

Remove all titles - they are no numerical data - and substitute the tab-spaces by commas to obtain:

```
1.232000000e+002,1.600000000e+000,1.320000000e+001
2.115000000e+002,2.300000000e+000,1.750000000e+001
3.855000000e+002,3.000000000e+000,2.210000000e+001
4.751000000e+002,3.400000000e+000,2.550000000e+001
6.411000000e+002,3.900000000e+000,3.260000000e+001
```

Save as a text-file with extension .dat and load the file. After Simplification or Approximation you will recognize that your conversion went well.

$$\#5: \begin{bmatrix} 1.232\cdot 10^{+2} & 1.6\cdot 10^{+0} & 1.32\cdot 10^{+1} \\ 2.115\cdot 10^{+2} & 2.3\cdot 10^{+0} & 1.75\cdot 10^{+1} \\ 3.855\cdot 10^{+2} & 3\cdot 10^{+0} & 2.21\cdot 10^{+1} \\ 4.751\cdot 10^{+2} & 3.4\cdot 10^{+0} & 2.55\cdot 10^{+1} \\ 6.411\cdot 10^{+2} & 3.9\cdot 10^{+0} & 3.26\cdot 10^{+1} \end{bmatrix}$$

$$\#6: \begin{bmatrix} 123.2 & 1.6 & 13.2 \\ 211.5 & 2.3 & 17.5 \\ 385.5 & 3 & 22.1 \\ 475.1 & 3.4 & 25.5 \\ 641.1 & 3.9 & 32.6 \end{bmatrix}$$

1	64	1
2	86	1
3	85	1
4	57	1
5	63	0
6	94	1
7	71	0

From [3] I loaded 2600 data (results from an US-wide calculus test and you can see the first seven rows of 2600 of the table). The data had a special form with spaces between the data fields and I wanted to import only the middle column. The easiest way was to import the data to Excel

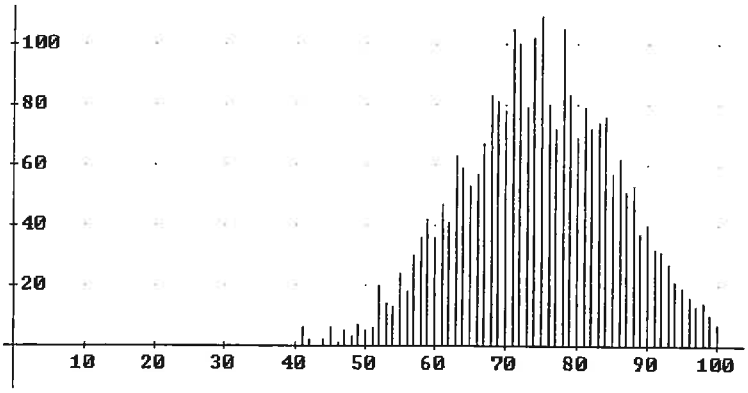
Save the whole file as text-file from within Excel as calculus.dat or copy the middle column to another Excel table and save it as middle.dat.

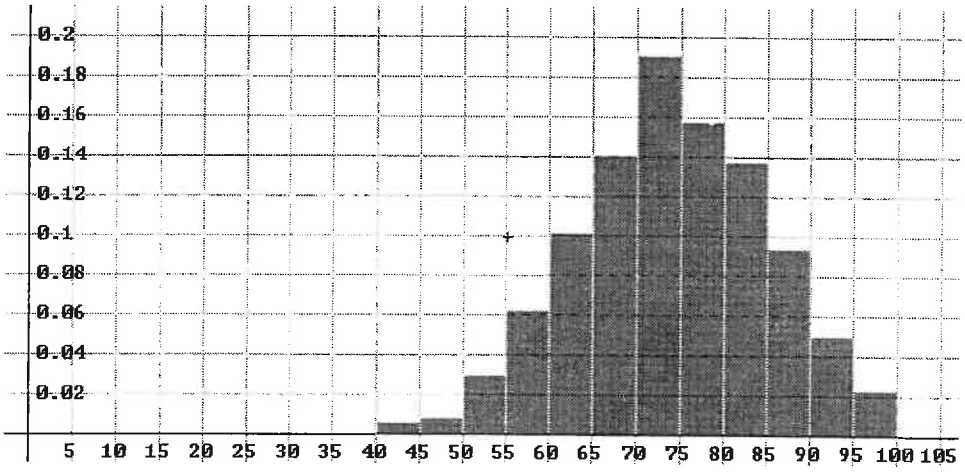
$$\#7: \begin{bmatrix} 1 & 64 & 1 \\ 2 & 86 & 1 \\ 3 & 85 & 1 \\ 4 & 57 & 1 \\ 5 & 63 & 0 \\ 6 & 94 & 1 \\ 7 & 71 & 0 \end{bmatrix} \qquad \#8: \begin{bmatrix} 64 \\ 86 \\ 85 \\ 57 \\ 63 \\ 94 \\ 71 \end{bmatrix}$$

See some results of my efforts treating the enormous set of data on the next page: the frequency diagram, a histogram with grouped data and a modified boxplot.

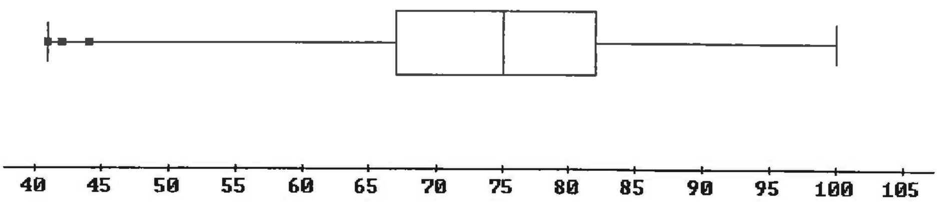
[1] Statistics for Business and Economics, Mc Clave ao, Prentice Hall, 2001
[2] Elementary Statistics - Picturing the World, Larson & Farber, Prentice Hall, 2000
[3] Applied Statistics - A first Course in Inference, Graybill ao, Prentice Hall, 1998

#40: FREQDIAG(scores)

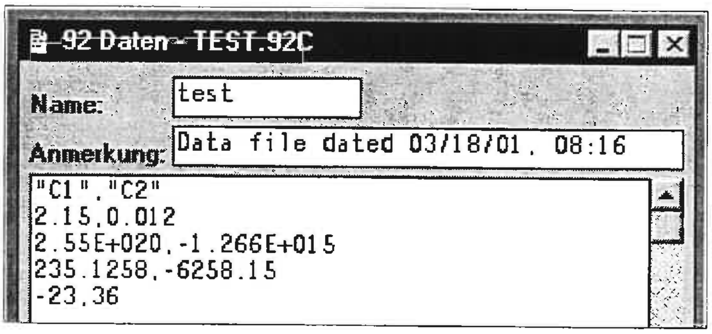




The modified BoxPlot (showing the outliers)

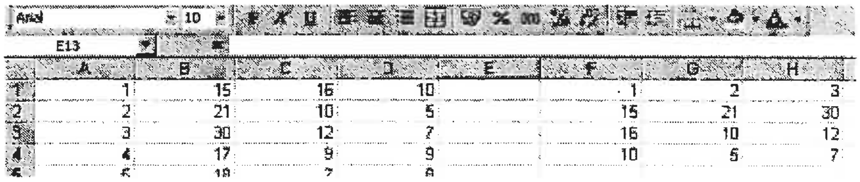


Now let´s try to bring the data into a TI. (I´ll not do it with the 2600 rows of the calculus scores). We will start analysing the format of the data stored in any Data table. Using *GraphLink* we open a data file (extension .92C or .9XC) and copy the *GraphLink* output to a text editor:



I couldn´t find a way to transform the given data in a way that the TI accepted the files as a data file as a whole. (extension .92C or .9XC). The file had to be converted in a text-file. See my solution:

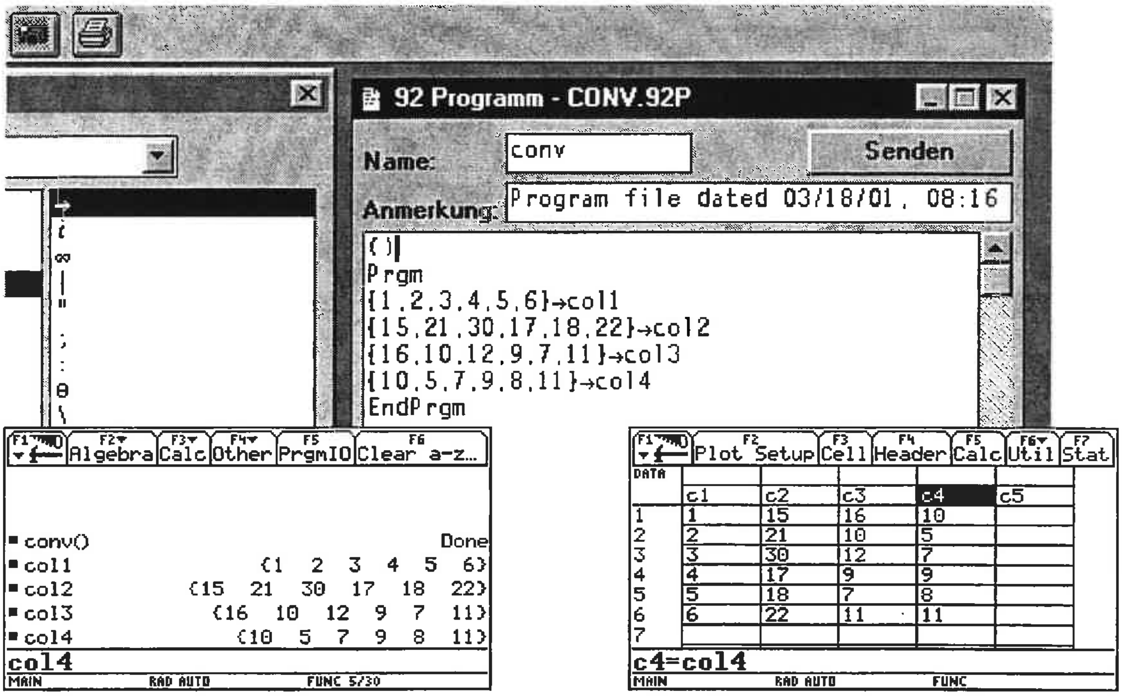
I wrote a small program conv( ) to convert the columns of the table into lists. But in a first step you have to transpose the table in your Spreadsheet program.



Then copy the transposed table into your text processing program and perform the same changes as shown above to receive finally a sequence of data lines. Edit the program conv() in *GraphLink* and copy and paste the lines between braces to save the lines as lists in your TI. Transmit conv() to the TI and run it.

```
1,2,3,4,5,6
15,21,30,17,18,22
16,10,12,9,7,11
10,5,7,9,8,11
```

Then open a new data table and insert into the headers the names of the newly created lists. You should find your data in the columns now as shown below.

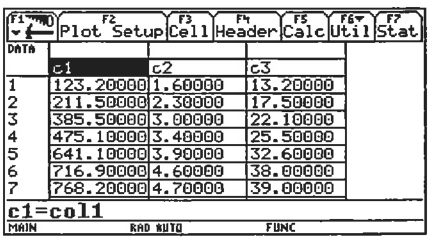


You will face new problems with data given in floating point notation like the following:

`1.2320E+02,2.1150E+02,3.8550E+02,4.7510E+02, ..................`

The procedure is the same but additionally you have to take care of the "E+" or "E-". Replace "E+" by "*10^" and "E-" accordingly. You can perform the replacements in the *GraphLink* Editor:

```
Prgm
{1.2320*10^02,2.1150*10^02,3.8550*10^02,4.7510*10^02,6.4110*10^02,7.1690*10
^02,7.6820*10^02,8.0660*10^02,8.5130*10^02,8.9380*10^02,9.3390*10^02}→col1
{1.6000*10^00,2.3000*10^00,3.0000*10^00,3.4000*10^00,3.9000*10^00,4.6000*10
^00,4.7000*10^00,4.8000*10^00,4.9000*10^00,5.0000*10^00,5.1000*10^00}→col2
{1.3200*10^01,1.7500*10^01,2.2100*10^01,2.5500*10^01,3.2600*10^01,3.8000*10
^01,3.9000*10^01,3.9600*10^01,4.0400*10^01,4.1200*10^01,4.2100*10^01}→col3
EndPrgm
```



Run conv( ) and open the Data/Matrix Editor, ...
If you know a better way to bring data into the TI´s brain, then please share your experience with us.

Note: There are rumours about an Excel compatible TI-89/92+ application to be released in the next future.