THE DERIVE - NEWSLETTER #46

THE BULLETIN OF THE

DERIVE®

USER GROUP

+ TI 92

Contents:

June 2002

The book shelf is very empty n this *DNL*. I can announce one French book to appear in October 2002.

> *Calculatrices symboliques, Transformer un outil en un instrument du travail mathématique: un problème didactique*
> Coordonné par Domique Guin et Luc Trouche (avec contributions de Michèle Artigue, Paul Drijvers, Philippe Elbaz-Vincent, Jean-baptiste Lagrange, Margaret Kendal, Robyn Pierce and Kaye Stacey.

But I came across a bundle of valueable web sites:

`www.ulib.org/webRoot/Books/Numerical_Recipes/bookcpdf.html`
You can download the complete book and inform about algorithms like Gauß-Newton and Marquardt-Levenberg – mentioned in the *DNL*

`www.ericse.org/digests.html`
A collection of interesting papers

`www.dartmouth.edu/~chance/teaching_aids/profiles.html`
A rich resource of materials

`www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/book.html`
You can download a complete textbook on probability theory

`www.maths.soto.ac.uk/EMIS/ELibM.html`
Provided as part of the European Mathematical Society's European Mathematics Information Service this site offers free access to electronic material including journals, proceedings, monographs, lecture notes, and graphical works. Articles are available in HTML, PostScript and PDF formats. Journals include Advances in Geometry, Discrete Mathematics & Theoretical Computer Science, The Electronic Journal of Combinatorics, The Electronic Journal of Differential Equations, The Electronic Journal of Linear Algebra, The Electronic Journal of Probability and Electronic Communications in Probability, and Theory and Applications of Categories.

`http://education.guardian.co.uk/higher/links/mathematics`
A gate to many interesting sites

`http://gauss.mat.eup.uva.es/~alfonso`
Visit Alfonso Poblacion's home page

`http://www.mathenet.net`
`http://www.bg-bab.ac.at/~mathe/index.htm`
Günter Schödls valueable collection of tests, teaching units (for DERIVE and in German)

`http://www.cwru.edu/artsci/math/wells/pub/abouthbk.htm`
Download Version 0.9 of The Handbook of Mathematical Discourse

`http://xahlee.org/SpecialPlaneCurves_dir/specialPlaneCurves.html`
A wonderful site if you want to know about curves. Highly recommend!

`http://www.rainerwonisch.de`
Rainer Wonisch's home page (materials for *DERIVE* and *DPGraph*)

Dear DUG Members,

DNL#46 offers a couple of contributions and I hope that each of you will find something useful, interesting, inspiring or amusing among them.

The User Forum is very extensive this time, because of some reactions on Richard Schorn´s NIM-article in *DNL#45*. Johann Wiesenbauer refers to this remarkable game in his Titbits, too. In addition you can find a collection of hints for editing and testing *DERIVE* programs. I am very proud that there are very few resources for Programming with *DERIVE* (I know only one exception, which is T. Etchells website `www.cms.livjm.ac.uk/deriveprogramming`) besides the *DERIVE* Newsletter. The fact that we have a lot of registrations for a *DERIVE*-Workshop on "Introductory Examples for Programming" in the frame of VISIT-ME 2002 underlines my opinion that there is a demand on materials concerning *DERIVE*-programming.

It´s great that Valeri Anisiu´s Easter program fits excellent to Richard Schorn´s Calendar Functions and I am very delighted to meet Valeri at VISIT-ME 2002 in Vienna in the very next future.

I am happy to have finished the Statistics Package with a solution for the Logistic Regression Problem with *DERIVE* and I am very grateful for MacDonald Phillips´ very extended generalized programs for nonstandard regression lines. Based on his programs I was able to create two slim specialised versions for Sinusiodal and Logistic Regression with an estimation for the parameters – derived from the data – included. You are friendly invited to check the programs.

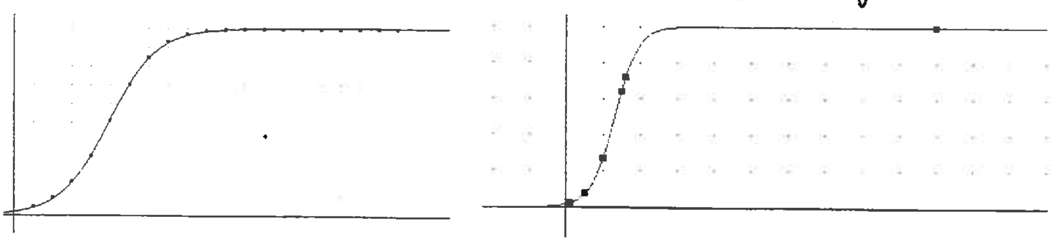
Your contributions were so rich and extended that I again have to leave a couple of even interesting articles for the next issue(s). By the way, it would be nice to receive one or the other pure *TI*-contribution for publishing. Dear *TI*-Users, please hear my call for the TIers: Offer your challenge for the *DERIVIANS*!

The book shelf is empty in this issue, but I collected a couple of very valueable websites. Full books and many, many interesting papers are ready for downloading. I am very grateful for each reference to other sites. I´d like to point to our ACDCA-website `www.acdca.ac.at` containing new materials (most of them in German).

In the last *DNL* I introduced our two granddaughters. Now you find Dominic, our oldest grandson, who welcomed some days ago his brother Moritz Jan Böhm. Moritz is very little and he will be ivited for a photo session for the next *DNL*.

After presenting the latest familiy news I wish you all a wonderful summer and I am looking forward to meeting many of you at VISIT-ME in Vienna

"Arrived" just after deadline: Logistic Regression



Download all *DERIVE*- and TI-89/92 files from

`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 44 pages minimum. The goals of the *DNL* are to enable the exchange of experiences made with *DERIVE* and the *TI*-92/89 as well as to create a group to discuss the possibilities of new methodical and didactical manners in teaching mathematics.

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

Editor: Mag. Josef Böhm
A-3042 Würmla
D´Lust 1
Austria
Phone/FAX: 43-(0)2275/8207
e-mail: nojo.boehm@pgv.at

Contributions:
Please send all contributions to the Editor. Non-English speakers are encouraged to write their contributions in English to reinforce the international touch of the *DNL*. It must be said, though, that non-English articles will be warmly welcomed nonetheless. Your contributions will be edited but not assessed. By submitting articles the author gives his consent for reprinting it in the *DNL*. The more contributions you will send, the more lively and richer in contents the *DERIVE & TI-92 Newsletter* will be.

Next issue: September 2002
Deadline 15 August 2002

Preview: Contributions waiting to be published

A Utility file for complex dynamic systems, Lechner, AUT
Tanz der Wallace-Geraden / Dance of Wallace-Lines, Baumann, GER
Various Training Programs
Type checking, Finite continued fractions, Welke, GER
Kaprekar´s "Self numbers", Schorn, GER
Examples for Statistics, Roeloffs, NL
Some simulations of Random Experiments, Böhm, AUT
Wonderful World of Pedal Curves, Böhm, AUT
Another Task for End Examination, Lechner., AUT
Tools for 3D-Problems, Lüke-Rosendahl, GER
Penrose Inverse of a Matrix, Karsten Schmidt, GER
Differential Equations in Secondary School, Günter Schödl, AUT
Putzer´s Method for the Calculation of e^{At}, Francisco Fernández, ARG
The Face Generator, Böhm, AUT

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

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

Valeriu Anisiu, Romania anisiu@math.ubbcluj.ro

Hello derivers,

Easter is coming soon!. To prove this assertion, here is a little program in *DERIVE* 5 which computes the date for this event in both western (Catholic) and eastern (Orthodox) variant.

```
easter(y,c,d,u,mao,mac,teo,tec,luo,zio,luc,zic):=~
PROG(c:=FLOOR(y,100),d:=c-2-FLOOR(c,4),u:=MOD(y,100),~
mao:=MOD(FLOOR(y,4)+y-4,7)+1,mac:=MOD(u+FLOOR(u,4)+FLOOR(c,4)-2*c,7)+1,~
teo:=MOD(11*MOD(y,19)+11,30),~
tec:=MOD(11*MOD(y,19)+9-c+FLOOR(c,4)+FLOOR(8*c+13,25),30),~
IF(tec=25,tec:=-4),IF(tec=26,tec:=-3),IF(tec>26,tec:-30),~
IF(teo>26,zio:=78-teo+d+MOD(teo-4-mao,7),~
   zio:=48-teo+d+MOD(teo-2-mao,7)),~
zic:=46-tec+MOD(tec+2-mac,7),~
luo:="Mar",luc:="Mar",~
IF(zio>31,[luo:="Apr",zio:-31]),~
IF(zio>30,[luo:="May",zio:-30]),~
IF(zic>31,[luc:="Apr",zic:-31]),~
[["Easter","Catholic:",zic,luc],[y,"Orthodox:",zio,luo]])
```

The function easter(year) can pasted in a derive session, or can be included (as it is) in a .mth file.
For example, easter(2002)
==> 31 March and 5 May for the two dates.

Cheers, and happy Easter!,
Valeriu

$$\text{easter(2002)} = \begin{bmatrix} \text{Easter} & \text{Catholic:} & 31 & \text{Mar} \\ 2002 & \text{Orthodox:} & 5 & \text{May} \end{bmatrix}$$

$$\text{easter(2003)} = \begin{bmatrix} \text{Easter} & \text{Catholic:} & 20 & \text{Apr} \\ 2003 & \text{Orthodox:} & 27 & \text{Apr} \end{bmatrix}$$

Wim de Jong WimdeJongTwo@aol.com

Valeriu,
Happy Easter to you too. "Little" seems to be an understatement for your program. I am afraid that in the absence of a documentation (which would probably be quite substantial) I cannot follow it.
According to Nachum Dershowitz and Edward M. Reingold ("Calendrical Calculations", Cambridge University Press, 1997) the calculation of Easter dates has a rich history because of the complexity of determining accurately the first full moon on or after the vernal equinox. I understand from what they say that on the basis of calculation methods now used by Catholic and Protestant churches, the most likely date of Easter is April 19 and the least likely March 22. Could this somehow be checked on the basis of your algorithm?
Cheers,
Wim de Jong

Heinz Rainer Geyer, Germany HeinzRainer.Geyer@t-online.de

I just imagine with Valerius nice programm that this is the date of EASTER.
So Nice AND PEACEFUL EASTER to you ALL OUT THERE, if there is any chance.
(by the way: Valeriu: please tell us about the math-history of finding the easter-date, i know, there was something of GAUSS, but there are others involved)

..... Back to the "wish-list":
Editing in the *DERIVE* SHEET and not in an extra window would be an even greater wish for most of us. But it may be a hard challenge in changing the concept. Maybe for the "wish-list" for SANTA-CLAUS?
Yours, Rainer [1]

[1] *See continuation on page 27*

Wim de Jong

HeinzRainer,
"*Calendrical Calculations*" by Nachum Dershowitz and Edward M.Reingold (Cambridge University Press,1997, ISBN 0-521-56474-3(paperback) may be the reference you are looking for. It is according to Ian Stewart "*One of the most fascinating books I have read all year. Takes chronology into the computer age with impressive erudition and elan*". And to quote the late Martin Gardner, "*This book is a definitive account of the world's major calendars and how to use them. It will be of interest not only to mathematicians, but also to historians and laymen.The authors are to be congratulated on a splendid research job.*"
I agree wholeheartedly with these comments. It is a beautiful book.
Cheers,
Wim de Jong

Terence Etchells & Albert Rich

From: Terence Etchells
Subject: Note on Partial Fractions

Hi Josef,
Further to your request about a note for DUG on the "Partial Fraction Anomaly" it appears that Albert has fixed the problem (see below). As you will read the identity used for functions of this form in the integrator is now employed in the Expander I have tested 1/(x^16+1) in 5.05 and it does it nicely. So I don't think it is worthy of a note in the newsletter anymore :-(

But hey, we got a result. Also it's nice to get an insight into HOW Derive works internally.
Terence

From: Terence Etchells <T.A.Etchells@livjm.ac.uk>
To: Albert Rich <adr@flex.com>
Sent: Tuesday, September 11, 2001 4:10
Subject: Partial Fractions

Hi Albert,

Is there any reason why the partial fraction expansion of 1/(1+x^8) doesn't compute? 1/(1+x^7) computes in 1.52 secs on this machine I have at work.
If it is the algorithm that is the problem, I have a neat alternative that computes the partial fractions of rational functions very quickly.

Cheers. *Terence*

From: Albert D. Rich [mailto:adr@flex.com]

Hi Terence,

If you have a straight-forward algorithm for computing partial fraction expansions of rational functions, it would certainly help out Derive's integration and other capabilities.
Aloha, Albert

From: Terence Etchells <T.A.Etchells@livjm.ac.uk>

Hi Albert,

How have you been affected by the NY disaster? It's a scary world in which we live.
On with more sanguine matters:

I don't know if this will improve Derive's integrator as my method, ironically, it uses the integrator. I discovered that Derive will exhaust memory trying to find the partial fraction expansion of 1/(1+x^8) but will integrate it very easily. So integrate the expression and split up the expression with terms and re differentiate the separate terms. We now have the partial fractions, so add the terms together without simplifying.

That is it! The attached dfw file has the Derive code that automates this.

fee() is a cute little function that adds the elements of a vector but does not simplify. i.e. fee([1,2,3])=1+2+3.

As I write this email it strikes me that we are doing too much work here:
How does Derive integrate 1/(1+x^8)? Derive quite easily factors x^8+1 into radical factors so finding the numerators of the partial fraction expansion should not be too demanding, should it? Or is Derive integrating 1/(x^8+1) by a means other than a partial fraction expansion? I strongly suspect that Derive has the capabilities of producing these expansions already, and all I've done is circumvent a problem with the straight expansion.

What do you say?
Cheers, Terence

From: Albert D. Rich [mailto:adr@flex.com]

Hi Terence,

It is scary. Let's hope the terrorists are brought to justice.
However, the long-term solution to the problem is to raise the quality of life of people around this planet to an acceptable level.

As you suspected, the Derive integrator uses a different algorithm than partial fraction expansion for integrating expressions of the form 1/(a*x^n+b). Basically it uses the fact that 1/(a*x^(2*n)+b) equals

1/(2*x^n*SQRT(a)*SQRT(-b)+2*b) - 1/(2*x^n*SQRT(a)*SQRT(-b)-2*b)

Therefore, INT(1/(a*x^(2*n)+b),x) equals

INT(1/(2*x^n*SQRT(a)*SQRT(-b)+2*b),x) - INT(1/(2*x^n*SQRT(a)*SQRT(-b)-2*b),x)

and the exponent of x has been reduced in half.

The algorithm we use for partial fraction expansion has to be more general, since it must handle more than binomial denominators. I was able to resolve the problem expanding 1/(x^8+1) and 1/(x^8-1) for the forthcoming version 5.05; however, larger exponents are still beyond Derive's ability. But we're working on it.

Hope you and your family are safe and sound.

Aloha, Albert

Note of the editor: you can find Terence´s idea among the files belonging to *DNL#46*

Don Phillips phillipsm@gao.gov

It appears that version 5.05 handles global variables differently than earlier versions. In earlier versions, a variable defined within a program was a local variable unless it was defined first outside the program; then it would be a global variable. In version 5.05, any variable defined within a program that is not designated a local variable, seems to be automatically a global variable, without having to be defined outside the program. Is this actually the case? If so, some programs that worked in earlier versions, may not work the way they were intended in 5.05.

Don Phillips

More on Wythoff's NIM

Pedro Tytgat pedro.tytgat@advalvas.be

Hi,
While reading the article Wythoff's Nim in DNL45, I was amazed to discover the safe positions are linked to the golden ratio! I immediately started a search on the Net, hoping to find the reason for this formula, but didn't find any. (If there is an easy explanation, this topic could be used in class! But I personally don't like formulas that appear without any reason.)

Can anyone point me to an explanation for it, preferably on the internet?

Thanks!
Pedro

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

http://www.ship.edu/~deensl/pgss/Day11/more-nim.html

should be sufficient, if you have some basic knowledge concerning Fibonacci sequences. As a matter of fact, this problem abounds with Fibonacci sequences for various initial values and this is where the golden ratio comes into play.

Strangely enough, Schorn didn't mention this crucial fact in his paper for some reason. (I would like to point out though that I like his articles very much otherwise for their imaginative ideas.)

Here is a slightly more "professional" version of his program along with an example:

```
wythoff(s, i_ := 1, m_ := {}, w_ := [[i, x↓i, y↓i]], x_ := 1) :=
  Loop
    If i_ > s
       RETURN REVERSE(w_)`
    w_ := ADJOIN([i_, x_, x_ + i_], w_)
    m_ := m_ ∪ {x_, x_ + i_}
    i_ :+ 1
    Loop
      If MEMBER?(x_, m_)
         x_ :+ 1
         exit
```

wythoff(20) =

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x_i	1	3	4	6	8	9	11	12	14	16	17	19	21	22	24	25	27	29	30	32
y_i	2	5	7	10	13	15	18	20	23	26	28	31	34	36	39	41	44	47	49	52

The first Fibonacci sequence is 1,2,3,5,8,13,21,34,... from the columns at the positions 1,2,5,13,...
Now skip all numbers occurring in this sequence and form the next one in the same way:

4,7,11,18,29,47,...

Again skip all the numbers occurring in the first 2 sequences and form the third one:

6,10,16,26,...

etc. etc.

As you can see Fibonacci sequences (for various initial values) always and everywhere...
Hope this helps.

Cheers, Johann

Pedro Tytgat pedro.tytgat@advalvas.be

After reading , http://www.ship.edu/~deensl/pgss/Day11/more-nim.html, I must say there is still no explanation why it is the golden ratio, of all numbers, that satisfies the two rules mentioned on the top of the page (not to confound with the two facts on the page you sent me). I can clearly see that the values for x and y ARE full of Fibonacci numbers or sequences, but not WHY it is so: there must be some NECESSITY.

That was the goal of my initial question: get to know why it MUST be the golden ratio? But the link you sent me was very valueable, as it brought me to the general form of the sequences a_n and b_n (the necessary and sufficient conditions which they must satisfy).

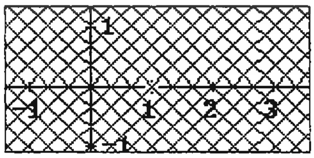
I haven't put much time in it so far, but it was quite easy to prove that it is only for the golden ration that property 2 holds: $b_n = a_n + n$ (with $a_n = [n \times g]$ and $b_n = [n \times g^2]$).

That's the kind of necessity I was looking for; now I understand what the golden ration is doing there!! I'll try to make some time this week to show that property (1) holds also for the golden ratio (I don't know yet if it holds for the golden ratio or if there are other irrational numbers that satisfy that condition). I must confess I'm not a professional mathematician (I'm civil engineer, but math teacher), so it might take some time .... :-)

Volker Loose volker@vwloose.de

Hi derivers,

I defined a function f(k, x) :=x^3/k-3x^2+2kx , then entered f(3,x)= Derive returned f(3,x)=x^3/3-3x^2+6x . I wanted to plot this and by mistake marked the whole equation. The plot was a nice pattern(see the attached file).



I tried to get a similar result and defined h(k,x):=x^2/k, entered h(3,x)=, got h(3,x)=x^2/3. This time the correct graph was plotted when I marked the whole equation. Has anyone an explanation?

Volker

Sebastiano Cappuccio scappucc@spfo.unibo.it

Hello Volkers (and Derivers),
I think that in plotting implicit equations, Derive scans the whole plot region and plots and connects all the points who have coordinates verifying the equation.
So, if you enter an univariate or a bivariate identity (e.g. x = x, x+y = y+x), you get a shaded plane with a grid.
In old versions of Derive (DfD 3, DfD 4) there was the Options Accuracy menu and you could modify the "thickness" of the grid. Now, in DfW 4 or 5, this menu disappeared.
I don't know why h(3,x)=x^2/3 appears correct: in my DfW 5 I got a grid again.
Ciao,
Sebastiano

DPGRAPH animiert *DERIVE* - - *DPGRAPH* animates *DERIVE*

Anwendung auf Funktionen zweier Variablen – Functions of two Variables

Rainer Wonisch, Germany, rainerwonisch@freenet.de

Einleitung

DERIVE 5 hat die grafischen Möglichkeiten erheblich erweitert. Es lassen sich jetzt auch 3D-Grafiken sehr gut darstellen. Was immer noch fehlt, ist die Möglichkeit, bei Funktionsscharen den Parameter online zu verändern oder sogar eine Animation zu erstellen, die den (oder die) Parameter quasi stufenlos verändert. *DPGRAPH* hat bietet beide Möglichkeiten, besitzt aber nicht die algebraischen Fähigkeiten von *DERIVE*. Beide Programme können jedoch sich einander ergänzend eingesetzt werden. Man bearbeitet seine Aufgabe mit *DERIVE*; falls eine Animation oder eine online Manipulation eines Parameters gewünscht wird, erzeugt man diese mit *DPGRAPH*.
Als Beispiele habe ich Scharen von Funktionen zweier Variablen herangezogen. Die Behandlung solcher Funktionen kann eine gute Ergänzung zur Behandlung der Differenzialrechung und der linearen Algebra sein, da sie beide Gebiete verbindet. Darüber hinaus sind die erzeugten Grafen sehr motivierend und regen zu eigenen Experimenten an.

Introduction

DERIVE 5 has extended graphic features. Although we can produce fine 3D representations we still miss the possiblity changing the parameters interactively or preparing animations. DPGraph can both, but it misses the mathematical capabilities of DERIVE. So we will combine the power of both programs. My examples are families of functions of two variables. The results are a wonderful completion in Calculus and Linear Algebra as well. Moreover the produced graphs are motivating in a high degree and may inspire for experimenting on one´s own.

Beispiel 1 – *Example 1*

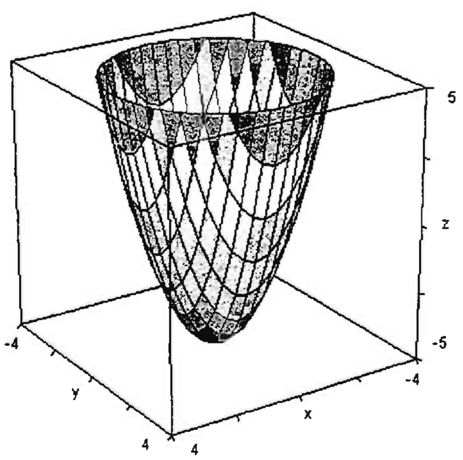
Aufgabe: Auf einem Rotationsparaboloid soll eine Tangentialebene beliebig verschiebbar sein.
Show a paraboloid of revolution together with a gliding tangent plane.

Lösungsansatz: Man definiert in *DERIVE* ein Rotationsparaboloid.

```
#1:    f(x, y) := x^2 + y^2 - 5
```

Der Zeichenbereich muss je nach dem gewählten Beispiel angepasst werden.
Für die Definition der Tangentialebene wird die entsprechende Definition für eine Tangente an eine Funktion einer Variablen übertragen. Als Form der Ebenengleichung bietet sich die Koordinatenform an.

We define the solid in DERIVE (#1) and adapt the plot region. To find the tangent plane we generalize the definition of a tangent on a one variable function and we use the coordinate-form for presenting the plane:



Für eine Funktion einer Variablen lautet die Tangentengleichung:

```
tang(x, xo) := f'(xo)·(x - xo) + f(xo)
```

Für eine Funktion zweier Variablen wird die 1. Ableitung ersetzt durch den Gradienten, und aus dem Produkt von reellen Zahlen wird das Skalarprodukt zweier Vektoren.

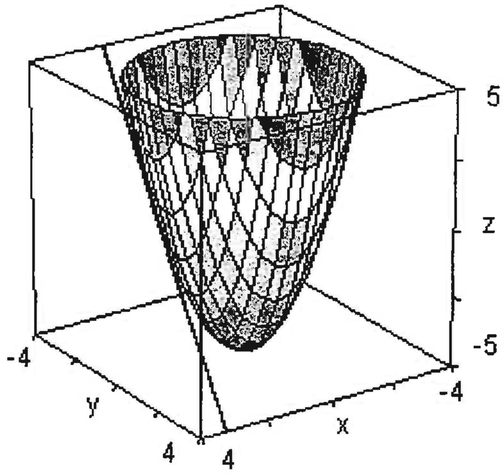
$$\#2: \quad \text{partdiff}(xo, yo) := \lim_{x \to xo} \lim_{y \to yo} \text{GRAD}(f(x, y))$$

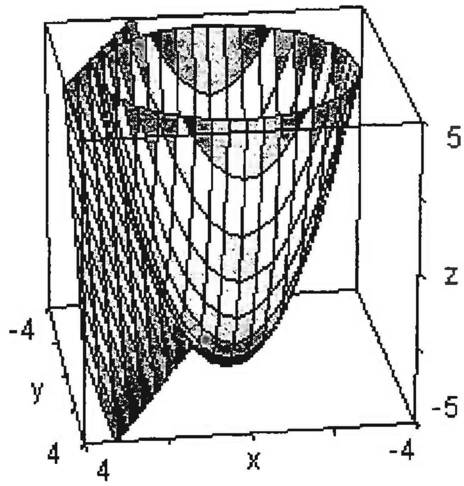
```
#3:   tangebene(xo, yo) := f(xo, yo) + partdiff(xo, yo)·[x - xo, y - yo, 0]
```

Die Tangentialebene zu einem bestimmten Punkt des Paraboloids lässt sich somit bestimmen. Ausdruck #4 zeigt ein Beispiel.

```
#4:   tangebene(2, -1) = 2·(2·x - y - 5)
```

For a function of two variables (surface) we replace the 1st derivative by the gradient and the product of real numbers changes into an inner product of two vectors. So we can easily define the tangent plane in any point of the paraboloid (see expression #4, tangebene = tangent plane).





Das linke Bild ist so gedreht, dass man die Ebene von der Kante her sieht.

Will man die Tangentialebene für einen beliebigen Punkt bestimmen, so ersetzt man die konkreten Koordinaten durch Parameter. In Anpassung an *DPGRAPH* werden als Symbole *a* und *b* gewählt.

I rotated the left picture to see the plane as a straight line. In order to cooperate with DPGraph we proceed using a and b as parameters to describe the tangent plane in any arbitrary point (expr #5). In expr #6 we use the VECTOR-command to produce a list of tangent planes. If we want to have the locus of the osculatings points on a circle around the z-axis, then we define the list as shown in expr#7.

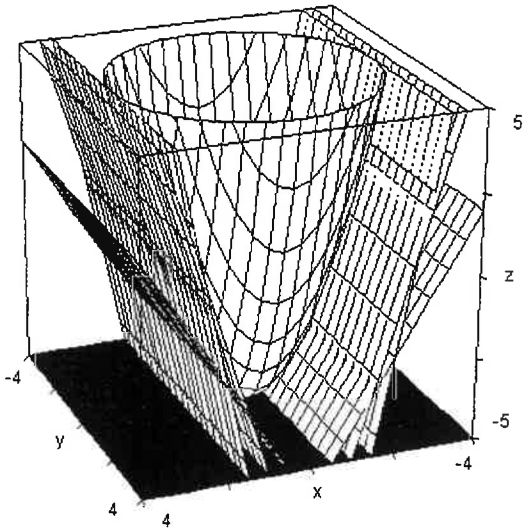
$$\#5: \quad \text{tangebene}(a, b) = 2 \cdot a \cdot x + 2 \cdot b \cdot y - a^2 - b^2 - 5$$

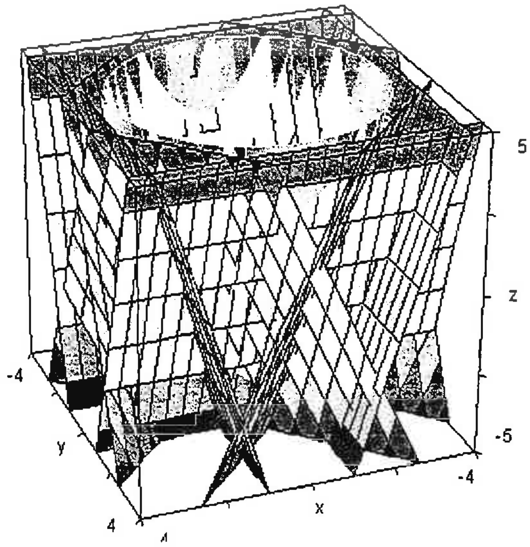
Für einen beliebigen Punkt (a, 0) lässt sich eine Liste dazugehöriger Ebenen erzeugen durch #6.

```
#6:   VECTOR(tangebene(a, 0), a, -3, 3) =

[- 2·(3·x + 7), - 4·x - 9, - 2·(x + 3), -5, 2·(x - 3), 4·x - 9, 2·(3·x - 7)]
```

Das linke Bild zeigt die entsprechende *DERIVE*-Grafik. Soll der Berührpunkt auf einem Kreis um die z-Achse liegen, so hilft Ausdruck #7.

$$\#7: \quad \text{VECTOR}\left(\text{tangebene}(2 \cdot \text{SIN}(a), 2 \cdot \text{COS}(a)), a, 0, 2 \cdot \pi, \frac{\pi}{4}\right)$$



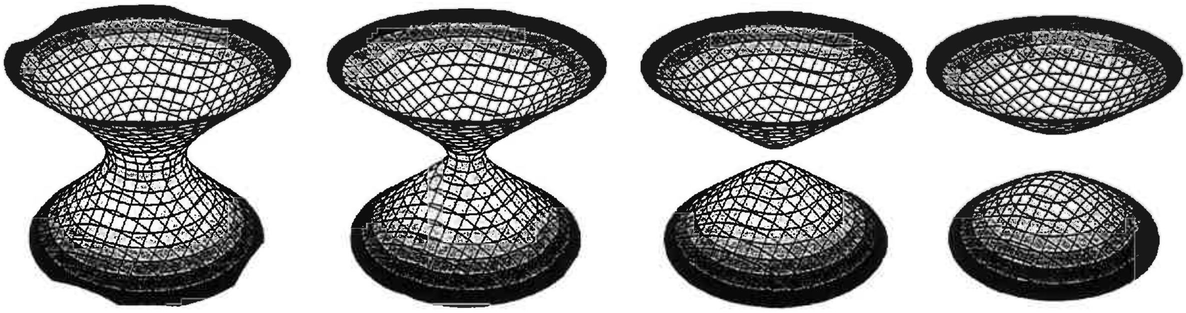


Bei beiden Bildern ist es nicht möglich, die Ebene online oder als Film wandern zu lassen. Mit *DPGRAPH* ist beides möglich.

In both plots it is not able to move the plane interactively or to produce a "movie" with a gliding plane. With DPGraph one can realize the one and the other, as well.

If you load DPGraph you immediately are presented a ready made animation. Open the Edit –Menu and inspect the default settings, which are self explaining – more or less.

Wenn Sie *DPGRAPH* starten, wird sofort eine vorgefertigte Animation sichtbar.



Wenn Sie das Edit-Menu öffnen, können Sie sich die erfolgten Eintragungen ansehen:

a := 1	Der Parameter a hat den Startwert 1 (*starting value for a = 1*)
a.minimum := -2	a hat einen Definitionsbereich von –2 bis +2 $-2 \le a \le 2$
a.maximum := 2	
b := 1	*Parameter b*
b.minimum := -2	.....
b.maximum := 2	.....
c := 1	*Parameter c*
c.minimum := -2	.....
c.maximum := 2	.....
d := 1	*Parameter d*
d.minimum := -2	.....
d.maximum := 2	.....

graph3d.box := false	Es soll kein Kasten gezeichnet werden – *plot no box*
graph3d.mesh := true	Die Gitterlinien sind sichtbar – *show the parameter lines*
graph3d.view := standard	normaler Blick – *standard view*
graph3d.background := white	Hintergrund weiß – *background white*
graph3d.perspective := true	Perspektivische Darstellung
graph3d.resolution := 21	Auflösung 21 (Anzahl der Felder)
graph3d.highlight := 0	keine Reflexe – *no reflections*
graph3d.shading := 0	keine Schattierung – *no shading*
graph3d.contrast := 1/2	Kontrastgrad 1/2
graph3d.transparency := 0	nicht transparent
graph3d.color := byheight	Farbe wird durch den Wert der z-Koordinate bestimmt
graph3d.minimumx := -3	Definitionsbereich für x,
graph3d.maximumx := 3	
graph3d.minimumy := -3	y
graph3d.maximumy := 3	
graph3d.minimumz := -3	und z
graph3d.maximumz := 3	
graph3d(a*x^2+b*y^2+sin(c*time)=d*z^2)	**eigentlicher Zeichenbefehl – *this is the Plot Command***

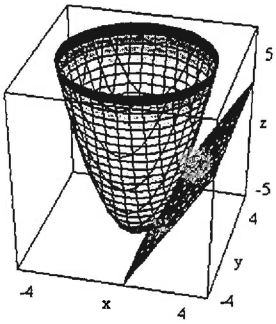
Die in dieser Datei eingetragenen Daten sind bis auf den eigentlichen Zeichenbefehl die Standardwerte von *DPGRAPH*. Ist man damit zufrieden, so können sie in der Datei für eine neue Zeichnung weggelassen werden. Offensichtlich kann *DPGRAPH* bis zu 4 Parameter verwalten; zusätzlich kann ein zeitabhängiger Wert mit Hilfe der Variablen *time* benutzt werden. Dies ermöglicht eine Animation.
Wir probieren das jetzt einmal aus für die Ebene am Paraboloid.

Obviously DPGraph is able to maintain up to four parameters and an additional parameter time, which is time dependent. This one is responsible for the animation. Let´s try with our plane osculating the paraboloid. We first set the parameters as follows:

graph3d.minimumx:=-4	Die Definitionsbereiche für x, y und z wählen wir analog *DERIVE*.
graph3d.minimumy:=-4	
graph3d.minimumz:=-5	
graph3d.maximumx:=4	*(The domain for the variables is according to DERIVE)*
graph3d.maximumy:=4	
graph3d.maximumz:=5	
a:=1	*a* soll von –3 bis 3 variieren mit dem Startwert 1
a.minimum:=-3	
a.maximum:=3	
b:=0	
b.minimum:=-3	*b* soll von –3 bis 3 variieren mit dem Startwert 0.
b.maximum:=3	

graph3d((z=x^2+y^2-5, z =2*a*x + 2*b*y - a^2 - b^2 - 5))

Der Zeichenbefehl bekommt hier eine Liste von Gleichungen übergeben. Die 1. Gleichung zeichnet das Paraboloid, die 2. Gleichung die Ebene mit den Vorgabewerten von *a* und *b*. Klickt man den Menüpunkt Scrollbar an, so öffnet sich ein Fenster, in dem man einstellen kann, welchen Parameter man mit Hilfe des Scrollbar online verändern möchte. Hier ist *a* gewählt.



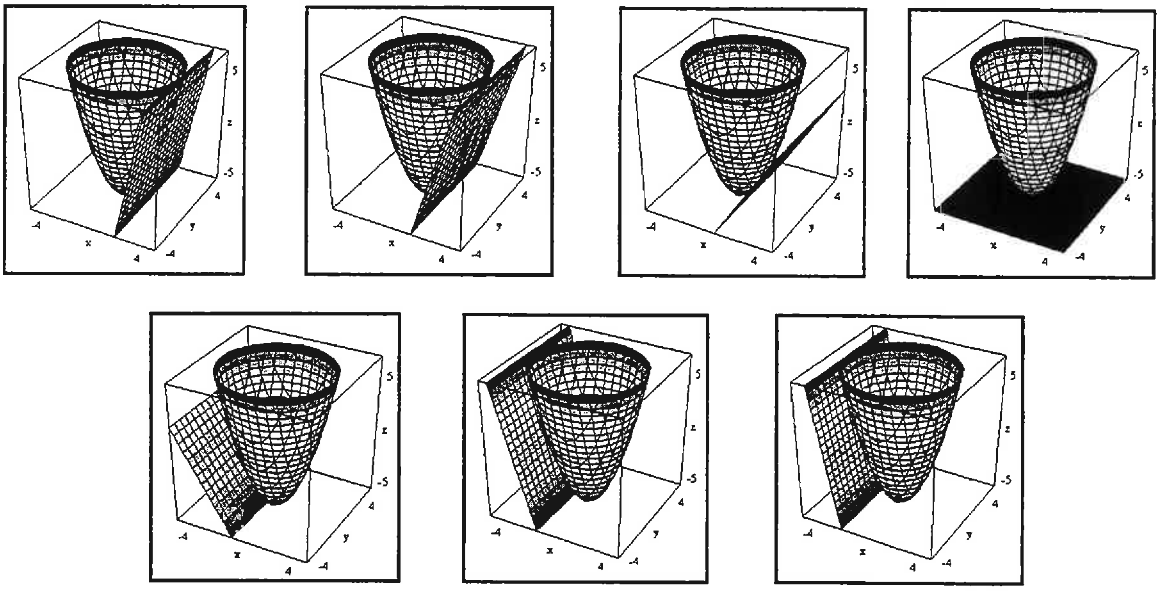
Scrollbar

Off | X slice | Y slice | Z slice

	Value	Minimum	Maximum
A variable	1	-3	3
B variable	0	-3	3
C variable	1	-2	2
D variable	1	-2	2
Contrast	.5	0	1
Surface transparency	0	0	1
Surface shading	0	0	1
Surface highlight	0	0	1
Vector shading	0	0	1
Vector highlight	0	0	1

The plot command contains a list of equations. The first one is responsible for the paraboloid and the second one for the family of planes with the given values for a and b. Clicking on the Scrollbar item in the menu bar one is offered a window to mark the parameter to be varied. The next screenshots show some positions changing variable a using the scrollbar.

Die nächsten Bilder zeigen die Ergebnisse bei Veränderung von a mit Hilfe des Scrollbars.



a variiert bei den Bildern in Einerschritten von –3 bis +3.

Die Zwischenschritte von Bild zu Bild erscheinen dabei quasi fließend, es sind umso mehr, je höher die Auflösung (resolution) eingestellt ist. Lässt man irgendeinen Wert von a stehen, wechselt in das Scrollbar-Fenster und wählt B variable, so kann man b unter Beibehaltung des Wertes von a online verändern.

Möchte man aus dem Ganzen einen Film machen, so ersetzt man einen oder beide Parameter durch eine periodische Funktion von *time*. Wenn der Berührpunkt der Ebene auf einem Kreis wandern soll, so wählt man 2**sin*(*time*) für a und 2**cos*(*time*) für b.

```
graph3d((z=x^2+y^2-5, z=2*2*sin(time)*x + 2*2*cos(time)*y - (2*sin(time))^2 - (2*cos(time))^2 - 5))
```

One can observe a smooth change from one position to the next one. Switch back to the Scrollbar-Window and choose B variable*, then you can vary b with a remaining fixed.*
*If you want to produce a movie, then replace one or both parameters by a periodic function of time. The osculation point shall move on a horizontal circle, so replace a and b by 2*sin(time) and 2*cos(time) respectively.*

Beispiel 2 – *Example 2*

Aufgabe: Untersucht werden soll die durch die folgende Gleichung definierte Funktionsschar auf Extrema:

$$f(x,y)=\left(x^2-a^2\right)\cdot\left(y^2-b^2\right).$$

The family of functions given above shall be explored with regard to extremal values.

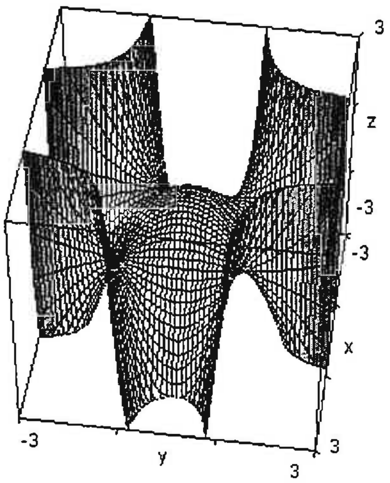
Definiert man die Funktionsschar in *DERIVE*, so lässt sich ein Graph nicht erzeugen, da keine Vorgabewerte für a und b vorhanden sind. Ausdruck #2 benutzt deshalb eine spezielle Funktion der Schar. In *DPGRAPH* kann man die Schar zeichnen lassen, da dort Vorgabewerte für a und b immer vorhanden sind, ohne nähere Festlegungen haben beide den Wert 1.

If we define the family of functions in DERIVE we cannot plot any graph, because there are not fixed values for a and b. Expression #2 uses one single member of the family. DPGraph plots the full family – or at least a part of it – because there are always default values for a and b (a = b = 1).

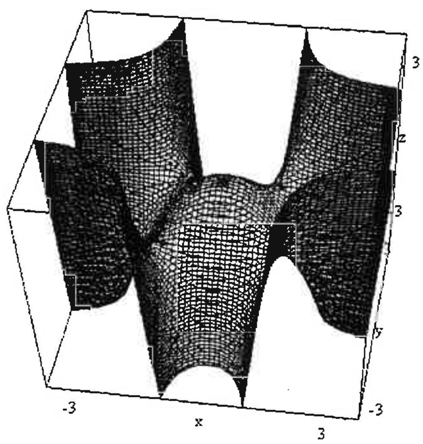
```
          2    2     2    2
#1:  f(x, y) := (x  - a )·(y  - b )

           2          2
#2:  f1(x, y) := (x  - 1)·(y  - 1)
```

```
graph3d.resolution := 61

graph3d.color:=BySteepness
graph3d(z=(x^2-a^2)*(y^2-b^2))
```

DERIVE-Plot

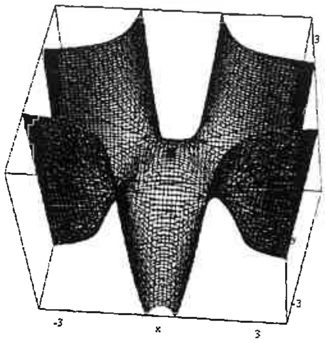


DPGRAPH-Plot

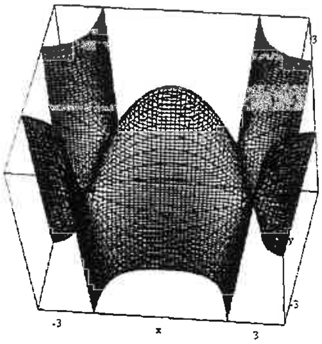


Beide Bilder ähneln sich sehr. Bei *DPGRAPH* kann man jetzt über den Menüpunkt Scrollbar z.B. a als variabel einstellen und dann im Bereich von 2 bis –2 online verändern, wobei hier wegen der Quadrate nur der Bereich 0 bis 2 interessant ist.

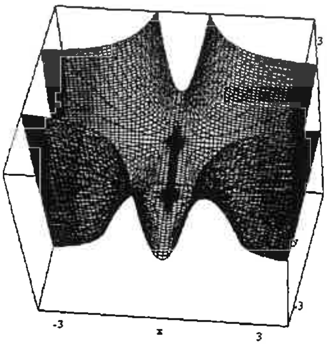
Both plots are very similar, but in DPGraph you can set a variable (in the Scrollbar Window) and change a for $-2 \le a \le +2$ *interactively. Because of the squares one can remain in the positive range for a.*



$a = \sqrt{3}$



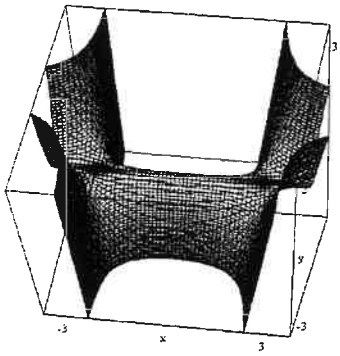
$a = \sqrt{2}$



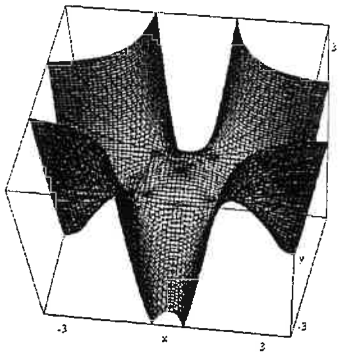
$a = 0$

Für alle Bilder hat b den Wert 1. Die Funktionen scheinen also bei (0,0) einen Hochpunkt zu haben, falls $a > 0$ ist. Der Fall $a = 0$ scheint ein Sonderfall zu sein. Außerdem haben die Funktionen für $a \neq 0$ vermutlich 4 Sattelpunkte.

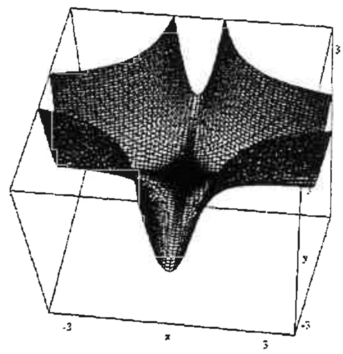
Analog lässt sich bei festem a der Wert von b verändern. Die folgenden Bilder zeigen mögliche Beispiele.



$a = 2; b = 0$



$a = \sqrt{2}; b = \sqrt{2}$



$a = 0; b = 0$

In the first row above b is fixed with b = 1. The functions seem to have a maximum at (0,0) for a ≠ 0. Case a = 0 might be a special case. Moreover the functions with a ≠ 0 seem to have 4 saddle points. In the same way we can vary b with a fixed a. The second row shows some examples of combinations. We cannot find out the exact properties of such a family using DPGraph. Bute we get an overview very quickly and conjectures about special cases. Now let´s switch back to DERIVE and use the CAS.

Die genauen Eigenschaften einer solchen Schar lassen sich mit DPGRAPH nicht ermitteln. Man bekommt aber sehr schnell einen Überblick über die Schar und erkennt auch Sonderfälle rasch.
Mit Hilfe des Gradienten der Funktion lassen sich die für Extrema in Frage kommenden Stellen ermitteln.

```
#5:   PARTDIFF(xo, yo) = [2·xo·(yo^2 - b^2), 2·yo·(xo^2 - a^2), 0]

#6:   SOLVE(2·xo·(yo^2 - b^2), [xo, yo], Real)

#7:   xo = 0 ∨ yo = -b ∨ yo = b

#8:   SOLVE(2·yo·(xo^2 - a^2), [xo, yo], Real)

#9:   xo = -a ∨ xo = a ∨ yo = 0
```

Die Hessematrix an den verdächtigen Stellen bringt eine weitere Klärung.
The Hesse Matrix clears the situation at suspicious locations.

$$\#10:\quad \mathrm{HESSE}(xo,\ yo) := \lim_{y\to yo}\ \lim_{x\to xo} \begin{bmatrix} \frac{d}{dx}\frac{d}{dx} f(x,\ y) & \frac{d}{dy}\frac{d}{dx} f(x,\ y) \\ \frac{d}{dx}\frac{d}{dy} f(x,\ y) & \frac{d}{dy}\frac{d}{dy} f(x,\ y) \end{bmatrix}$$

Sei $a \neq 0$ und $b \neq 0$.

$$\#11:\quad \mathrm{HESSE}(0,\ 0) = \begin{bmatrix} -2\cdot b^2 & 0 \\ 0 & -2\cdot a^2 \end{bmatrix}$$

Die Determinante der Matrix ist positiv, die Matrix selbst ist in diesem Punkt positiv definit. Es liegt eine isolierte Maximumstelle vor.

Determinant is > 0 → isolated Maximum

$$\#12:\quad \mathrm{HESSE}(a,\ b) = \begin{bmatrix} 0 & 4\cdot a\cdot b \\ 4\cdot a\cdot b & 0 \end{bmatrix}$$

Negative Determinante, also Sattelpunkt

Negative determinant → saddle point

$$\#13:\quad \mathrm{HESSE}(-a,\ b) = \begin{bmatrix} 0 & -4\cdot a\cdot b \\ -4\cdot a\cdot b & 0 \end{bmatrix}$$

Negative Determinante, also Sattelpunkt

Negative determinant → saddle point

$$\#14:\quad \mathrm{HESSE}(a,\ -b) = \begin{bmatrix} 0 & -4\cdot a\cdot b \\ -4\cdot a\cdot b & 0 \end{bmatrix}$$

Negative Determinante, also Sattelpunkt

Negative determinant → saddle point

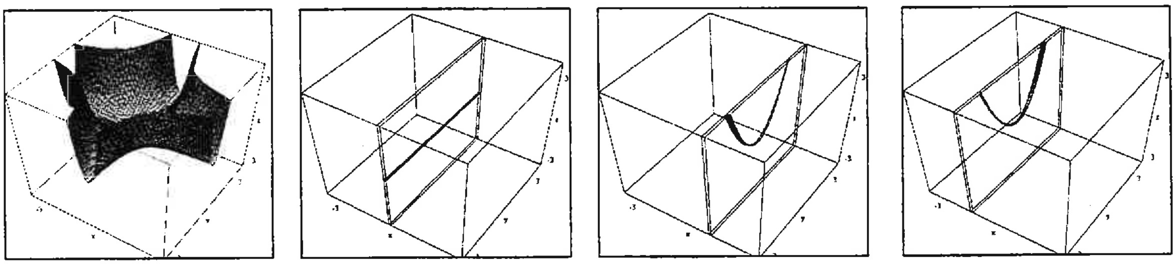
$$\#15:\quad \mathrm{HESSE}(-a,\ -b) = \begin{bmatrix} 0 & 4\cdot a\cdot b \\ 4\cdot a\cdot b & 0 \end{bmatrix}$$

Negative Determinante, also Sattelpunkt

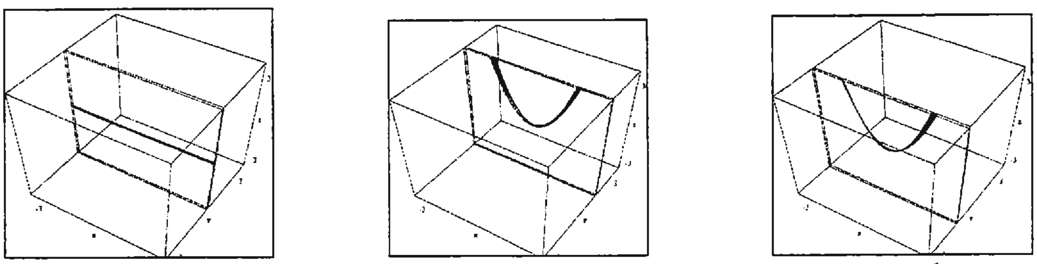
Negative determinant → saddle point

Sei $a = 0$ und $b = 0$. Die Hessematrix bringt keine Entscheidung, da ihre Determinante den Wert 0 hat. Wir schauen uns diese besondere Situation noch einmal mit *DPGRAPH* an. Dabei benutzen wir ein Feature, das sich im Menüpunkt Scrollbar befindet. Es hat Ähnlichkeiten mit dem Spurmodus in *DERIVE*, zeigt aber den Ausschnitt aus dem Bild sehr viel deutlicher.

If a and b = 0 the Hesse matrix doesn´t give an answer, because the determinant equals 0. So we investigate this special situation once more using another feature of DPGraph, which works similar to DERIVE´s trace mode, but gives more insight. Sections through the surface make clear that for the special case in point (0,0) one can find a non isolated maximum.



$x = 0$ $x = 1$ $x = -1$



$y = 0$ $y = 1$ $y = -1$

Die Schnitte durch den Graph machen deutlich, dass für diesen Sonderfall im Punkt (0,0) eine nicht isolierte Maximumstelle vorhanden ist.

Resümee

Die Beispiele zeigen nur einen Teil der Fähigkeiten von *DPGRAPH*, die als Ergänzung zu *DERIVE* eingesetzt werden können. Noch einmal zur Verdeutlichung: *DPGRAPH* kann in keiner Weise *DERIVE* das Wasser reichen, da es keinerlei algebraischen Fähigkeiten besitzt. Für sehr viele Aufgaben reichen die grafischen Fähigkeiten von *DERIVE* völlig aus. Nur manchmal wünscht man sich etwas mehr, hauptsächlich im Bereich der Animation und der Darstellung von implizit definierten Kurven. Wenn Sie mehr über *DPGRAPH* erfahren wollen, so schauen Sie einmal in die Homepage von David Parker.
Die Beispiele wurden so gewählt, dass man das Zusammenspiel von Derive und DPGraph erkennen kann. Weitere Beispiele dazu finden Sie auf meiner Homepage `www.rainerwonisch.de`.

Summary

The examples can only show a part of DPGraph´s capabilities, which can be used to accomplish working with Derive. DPGraph cannot reach DERIVE, because of its lack in algebraic abilities. For many problems the graphic features of DERIVE are sufficient. But sometimes one would like to have a bit more, especially on the field of animation and representation of implicitly defined curves. If you want to know more about DPGraph then please visit David Parker´s homepage. I chose examples to demonstrate cooperation between Derive and DPGraph. You can find more examples on my homepage `www.rainerwonisch.de`.

References

http://www.davidparker.com
http://odin.math.nau.edu/~jws/classes/137.2001.1/dpglinks.html
http://members.tripod.com/vismath/

Rainer Wonisch, *Animationen in Derive und DPGraph in Mathematik und Physik*, bk-teachware 2002

DPGRaph can be purchased via David Parkers website
DPGraph wird vertrieben von bk-teachware, http://shop.bk-teachware.com

Among the files belonging to this DNL-issue you will find the *DPGraph*-Viewer. It can be used to view Rainer Wonisch´s products. I recommend purchasing this wonderful small and inexpensive piece of software (only 10.54 EURO). Some days ago I had a talk with Johann Wiesenbauer, and we both agreed that we would like to find *DPGraph* implemented into *DERIVE* in one of the next versions!!??

Rüdeger Baumann´s Challenge

A Fibonacci-like sequence is defined by

$$\text{fieb}(2) = 2,\ \text{fieb}(3) = 3$$

$$\text{fieb}(k+2) = \text{fieb}(k+1)+\text{fieb}(k) + 1$$

It should be checked, if p is divisor of fieb(p) for p is a prime number.

A non – efficient checkprogram is:

```
#1: fieb(k):=IF(k<=2,2,IF(k=3,3,fieb(k-2)+fieb(k-1)+1))
#2: Pz(n):=ITERATES(NEXT_PRIME(k),k,2,n)
#3: fiebeck(n):=VECTOR([p,fieb(p),FACTORS(fieb(p))],p,Pz(n))
```

Tasks:
(a) Write an efficient *DERIVE*-procedure.
(b) Try to prove or to disprove the conjecture.

Experimente mit „Zufallspermutationen" in *DERIVE*
Experiments with Random Permuations in *DERIVE*

Franz Schlöglhofer, Traunkirchen, Austria, f.schloeglhofer@utanet.at

Using the RANDOM function one can generate pseudo random numbers. This makes possible simulations of random experiments. Applying RANDOM simulates a random experiment "with replacement". Throwing a die is an example for this.
Random experiments "without replacement" can be simulated by randomly generated permutations. Using the programming capability of *DERIVE* 5 one can produce auxiliary programs for this purpose. In the following some of such programs will be described and used to perform simulations. The advantage of a simulation is that we don't need difficult combinatoric considerations for each problem. At the other hand it requires sometimes longer calculation times. Moreover one gets relative frequencies instead of probabilites and means instead of expectations.

The *DERIVE* command RANDOM(n)

Each call of RANDOM(n) with $n \in$ N and $n > 1$ returns a random number out of $\{0,1,2,3,\ldots,n-1\}$, RANDOM(6) delivers one integer random number x with $0 \le x \le 5$.
To simulate throwing 20 times a die needs only one line:

```
VECTOR(RANDOM(6) + 1, i, 20)
          [5, 3, 3, 1, 3, 6, 2, 2, 5, 3, 5, 1, 4, 6, 5, 1, 2, 2, 2, 6]
```

Two notes of the Editor: To have at each start of the session new random numbers you should first simplify the expression RANDOM(0). On the TI-89/92 you have to set a Rand-Seed # and take care that rand(n) – with $n > 1$ - returns an integer random number x with $1 \le x \le n$.

Random Permutations

Using the self written function rperm(n,a) we can generate random permutations.

```
                         RANDOM(0) = 1285230889
rperm(10, 10)
                         [3, 2, 4, 8, 1, 7, 6, 5, 10, 9]
                         [3, 7, 9, 10, 8, 6, 5, 1, 4, 2]
                         [7, 5, 10, 3, 1, 6, 9, 8, 4, 2]
                  rperm(45, 6) = [24, 20, 15, 42, 43, 1]
                  rperm(45, 6) = [1, 28, 21, 22, 26, 3]
            SORT(rperm(45, 6)) = [2, 4, 10, 23, 27, 33]
```

Simplifying rperm(10,10) returns the numbers 1, 2, …., 10 in a random order.

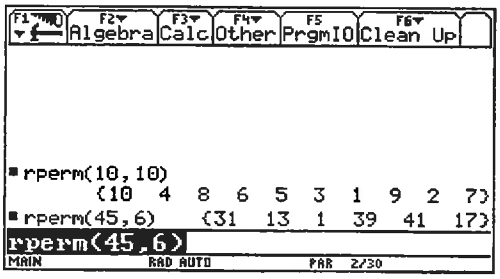
rperm(45,6) simulates Austria's most appreciated lottery "6 aus 45", where six numbers are drawn out of {1,2,3,….,45}. The elements of the list can easily be sorted.

In `rperm(n,a)` we first generate a vector (list) $v = [1,2,...,n]$ to pick out a elements in a random order.

```
rperm(n, a, u := [], v, h) :=
  Prog
    v := VECTOR(i, i, 1, n)
    Loop
      If DIM(u) < a
        Prog
          If n > 1
             h := RANDOM(n) + 1
             h := 1
          u := APPEND(u, [v↓h])
          v := DELETE(v, h)
          n := n - 1
        RETURN u
```

This is the respective TI-92 function:

```
rperm(n,a)
Func
Local  u,v,k,h
{}→u
seq(i,i,1,n)→v
For  k,1,a
  If n>1 Then
      rand(n)→h
         Else
      1→h
  EndIf
  augment(u,{v[h]})→u
  augment(left(v,h-1),
           right(v,dim(v)-h))→v
  n-1→n
EndFor
u
EndFunc
```



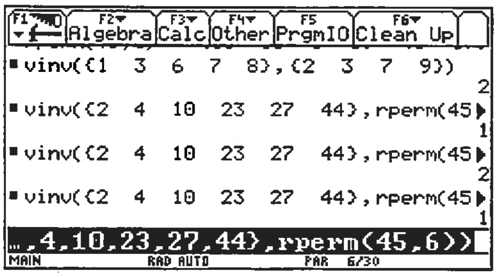
Corresponding elements in two vectors (lists)

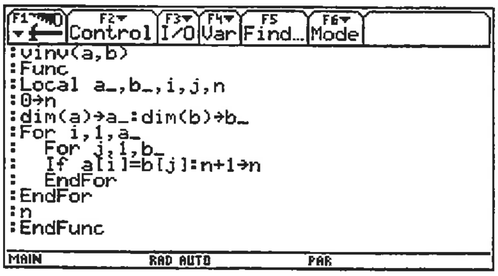
It will appear very important for our simulations to find out correspondences in two lists. Function `vinv(u,v)` returns the number of corresponding elements:

```
vinv(u, v) := Σ(VECTOR(DIM(SELECT(x = u_i, x, v)), i, 1, DIM(u)))
```

```
vinv([1, 3, 6, 7, 8], [2, 3, 7, 9]) = 2
vinv([2, 4, 10, 23, 27, 44], rperm(45, 6)) = 2
vinv([2, 4, 10, 23, 27, 44], rperm(45, 6)) = 0
```

We use our lottery-prophecy from above and simulate the Saturday and Wednesday drawing to find that we have 2 and none correct numbers on our ticket, followed by the results of 10 *DERIVE*-drawings.






```
VECTOR(vinv([2, 4, 10, 23, 27, 44], rperm(45, 6)), j, 10)
                        [1, 0, 2, 2, 0, 1, 1, 1, 0, 0]
```

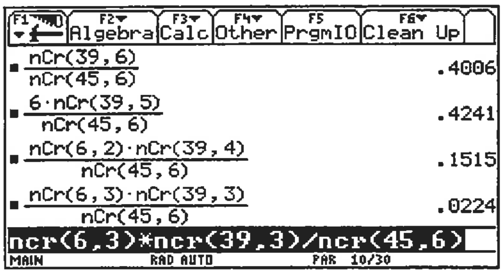
Frequency distribution

Calling `f_dist(v)` gives the distribution of the absolute frequencies of the number of correspondences.

```
f_dist(v) := VECTOR([i, DIM(SELECT(x = i, x, v))], i, MIN(v), MAX(v))
```

As an example we will simulate 10000 lottery drawings and compare the results with the numbers of our ticket. Only the occurencies from the minimum-value to the maximum-value are presented.

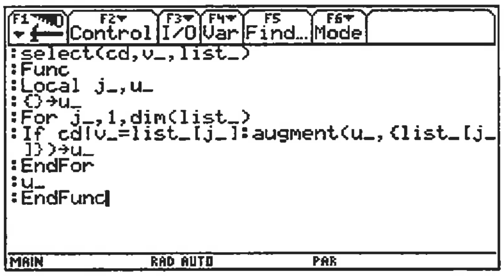
```
f_dist(VECTOR(vinv([2, 4, 10, 23, 27, 44], rperm(45, 6)), j, 10000))
```

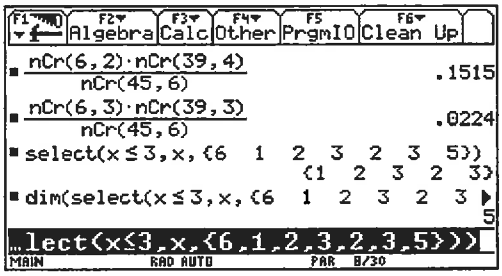


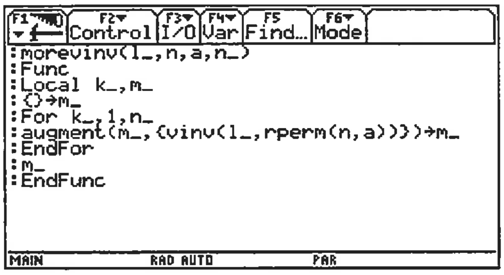
0	3998
1	4212
2	1559
3	224
4	7

Compare with the theoretical Probability!

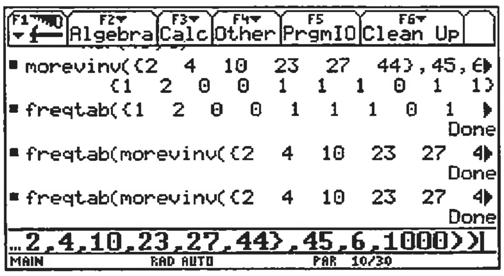
How to do with the TI-92? I am sure that there are many ways to perform this investigation on the TI – and some of them will be more sophisticated as my one. At first I implemented the `SELECT`-function on the TI:

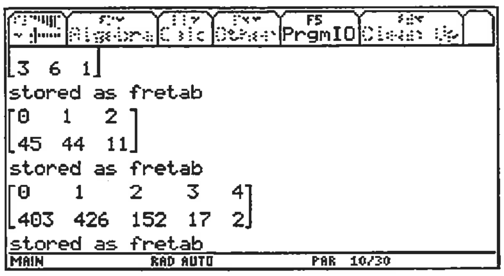






It was not so easy to create a `VECTOR`-function, which can be used as universally as the *DERIVE* `VECTOR`, so I wrote a function to produce more `vinvs` in a row, `morevinv()`.
And together with `freqtab(list)` from DNL#45 we obtain a list, very likely to the *DERIVE* generated result.





The simulation of 1000 drawings takes some time, of course.

Some more applications:

Quality control

We have a delivery of 50 devices and we know that 12 of them are not ok. For further investigation 4 devices are picked out randomly. Which is the probability that all of them are ok?

Here we have to simulate two selections: we choose 12 pieces of the given 50 pieces (which are "marked" as the bad ones) and we pick out a sample of 4 of the 50. The number of corresponding numbers is the number of bad products occuring in the sample.

`vinv(rperm(50, 12), rperm(50, 4)) = 0` In two experiments we find no bad device and
`vinv(rperm(50, 12), rperm(50, 4)) = 1` one bad piece among the four selected.

Now let´s have 20 samples:

```
VECTOR(vinv(rperm(50, 12), rperm(50, 4)), i, 20)
                    [2, 2, 2, 0, 2, 3, 2, 1, 0, 2, 0, 0, 1, 1, 1, 3, 2, 1, 1, 1]
```

Using SELECT again we can filter out the results "no bad device" – which are the zeros.

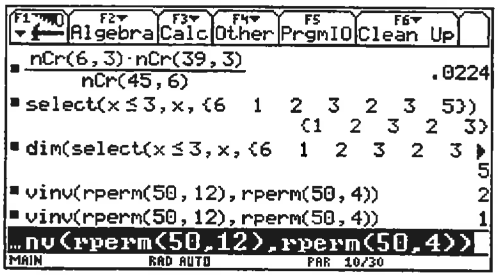
```
SELECT(x = 0, x, VECTOR(vinv(rperm(50, 12), rperm(50, 4)), i, 20))
                                   [0, 0, 0, 0, 0, 0]
DIM(SELECT(x = 0, x, VECTOR(vinv(rperm(50, 12), rperm(50, 4)), i, 1000)))
                                         326
DIM(SELECT(x = 0, x, VECTOR(vinv(rperm(50, 12), rperm(50, 4)), i, 10000)))
                                         3152
```

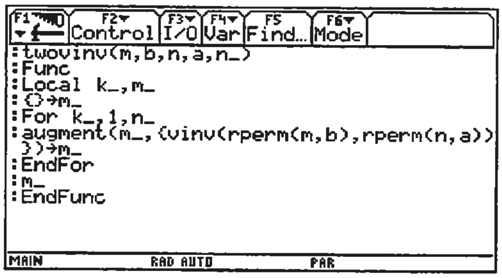
`DIM(list)` counts the elements of the list and gives the number of experiments with no corresponding numbers, i.e. there is no bad piece in the sample. The relative frequencies found by simulation are 0.33 and 0.32. What is the probability for this event?

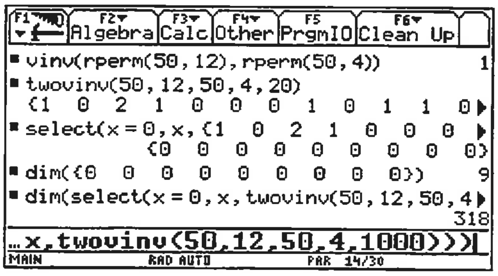
$$\frac{\text{COMB}(12,\ 0)\cdot\text{COMB}(38,\ 4)}{\text{COMB}(50,\ 4)} = 0.3205167173$$

In fact we approximate the values of a hypergeometric distribution.[1]

Let me turn on the TI and perform the same simulation:







[1] At the end of this contribution you will find a simulation for the hypergeometric distribution. It is part of a package of simulations for the TI-89/92 which can be downloaded from `www.acdca.ac.at`. Josef

Who makes himself a present?

At the occasion of a celebration 22 people bring a gift. All the gifts are distributed randomly among the 22 guests. Each of the participants receives one gift, so it can happen that one or the other will get back his own package. What is the average value of people getting back their own gift?

`VECTOR(i,i,20)` generates the list [1,2,3,…..,22], which is the sorted list of the participants. `rperm(22,22)` returns a random permutation, which describes the distribution of the gifts:

```
[VECTOR(i, i, 22), rperm(22, 22)]
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
18	14	3	2	21	1	15	8	20	6	10	16	12	9	19	5	4	11	7	17	22	13

Guest #1 receives the gift from guest#18, guests #3 and #8 will be surprised to find their own gifts. Subtracting the vectors result in another vector of same dimension and each position occupied by zero indicates that this person finds his/her own gift. Hence we apply the same trick as before and

```
DIM(SELECT(x = 0, x, VECTOR(i, i, 22) - rperm(22, 22))) = 0
DIM(SELECT(x = 0, x, VECTOR(i, i, 22) - rperm(22, 22))) = 0
DIM(SELECT(x = 0, x, VECTOR(i, i, 22) - rperm(22, 22))) = 2
```

We would like to have a series of n experiments and a variable number k of guests, so we produce an appropriate function and simulate a large number of parties.

```
gifts(k, n) := VECTOR(DIM(SELECT(x = 0, x, VECTOR(i, i, k) - rperm(k, k))), j, n)
gifts(22, 100)
[2, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 2, 0, 1, 1, 0, 2, 2, 1, 2, 0, 2, 1, 1, 3, 0, 1, 1, 0,
 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 3, 0, 1, 2, 0, 0, 3, 1, 0, 0, 2, 0,
 2, 0, 1, 0, 2, 0, 1, 1, 0, 0, 1, 3, 0, 3, 0, 0, 1, 0, 1, 2, 1, 1, 2, 0, 1, 3, 0, 0, 4,
 1, 3, 0, 2, 1, 0, 1, 0, 1, 1, 2, 4, 1]
```

A short function returns the average of the self-present-givers:

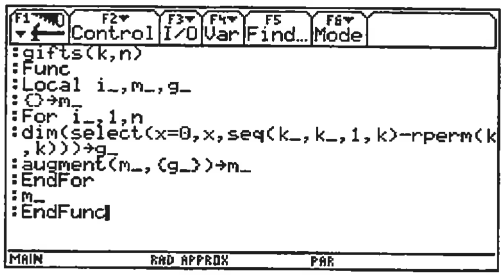
$$\mu(v) := \frac{\Sigma(v)}{DIM(v)}$$

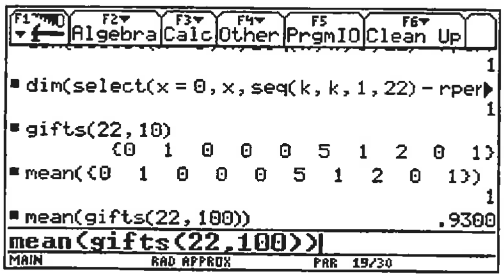
```
μ(gifts(22, 100)) = 0.94
μ(gifts(22, 100)) = 0.96
μ(gifts(10, 200)) = 0.98
μ(gifts(10, 200)) = 0.89
μ(gifts(50, 500)) = 0.908
μ(gifts(50, 500)) = 0.932
```

(or take simply `average(gifts(22,100))`).

The expectation calculated by means op probability theory is exact 1.

And here is again my TI-92 equivalent:





Dangerous Mistake in a Pharmacy

A pharmacist sells drugs to 100 customers. Later he notes that he had given two people the wrong drug. It is necessary to find out the two persons. What is the average number of people to be contacted?

In this simulation we will assign the two people to be found the numbers 1 and 2 (without restriction of generality) in a random permutation. It is not only to look for values in a list – which satisfy certain conditions (`SELECT`) but additionally we have to find out their position in the list.

For solving this problem we introduce two more auxliliary functions. `select_pos` returns the positions of elements in a vector which are satisfying a certain condition and `last_pos` gives - according to its name - the last position of occurrence of this condition.

The examples following demonstrate the use of both functions:

```
select_pos(x < 4, x, [1, 6, 3, 2, 7, 5, 4]) = [1, 3, 4]
select_pos(x < 3, x, rperm(100, 100)) = [16, 47]
last_pos(x < 4, x, [1, 6, 3, 2, 7, 5, 4]) = 4
last_pos(x ≤ 4, x, [1, 6, 3, 2, 7, 5, 4]) = 7
```

Here is `select_pos` in two versions – with and without subscripts.

```
select_pos(b, k, v, i := 1, u := [], n) :=
  Prog
    n := DIM(v)
    Loop
      If i > n
         RETURN u
      Prog
        If SUBST(b, k, v↓i)
           u := APPEND(u, [i])
        i :+ 1
```

```
select_pos2(b, k, v, u := [], i := 1) :=
  Prog
    Loop
      If v = []
         RETURN u
      If SUBST(b, k, FIRST(v))
         u := APPEND(u, [i])
      i :+ 1
      v := REST(v)
```

followed by the one-line function `last_pos`:

```
last_pos(b, k, v) := FIRST(REVERSE(select_pos(b, k, v)))
```

Now we are using `last_pos` to find the last position in any random permutation of the 100 persons which are marked by 1 and 2. In the first simulation we had to contact 29 customers, in the second one 59.

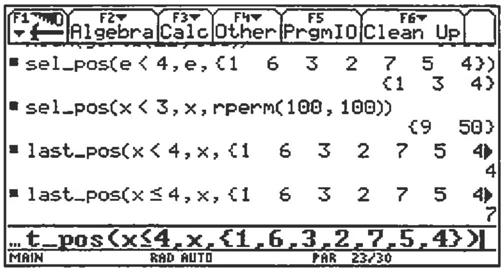
```
last_pos(x < 3, x, rperm(100, 100)) = 29
last_pos(x < 3, x, rperm(100, 100)) = 59
```

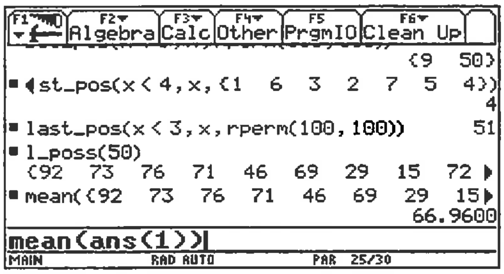
To repeat the experiment several times we construct a list formed by the results of `last_pos` and calculate the mean:

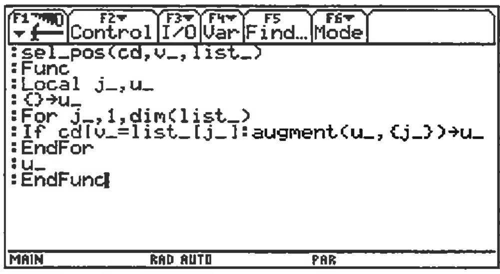
```
μ(VECTOR(last_pos(x < 3, x, rperm(100, 100)), j, 50))
                                    66.28
                                    60.04
VECTOR(μ(VECTOR(last_pos(x < 3, x, rperm(100, 100)), j, 50)), k, 10)
     [69.34, 67.42, 67.34, 65.54, 70.44, 62.84, 71.86, 64.9, 64.76, 66.82]
```

The last command generates 10 series of 50 simulations. The theoretical expectation is 67.3.

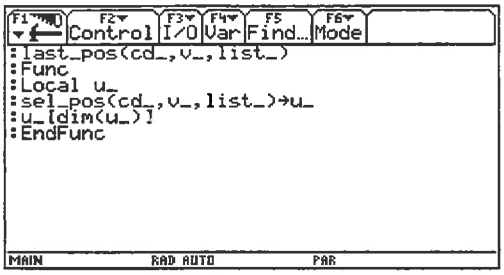
It is not too difficult to transfer the functions to the TI-92. It only needs some slight changes of earlier functions.

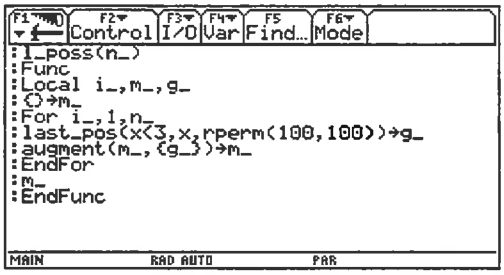






only





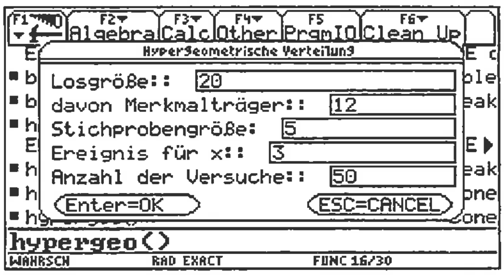
References

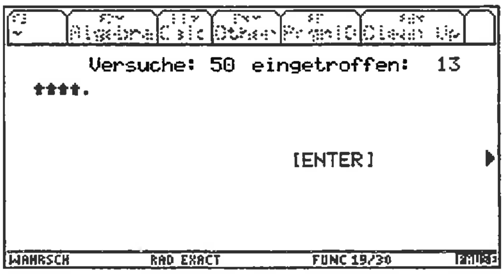
Engel, Arthur; *Mathematisches Experimentieren mit dem PC*; Klett Verlag, Stuttgart 1991

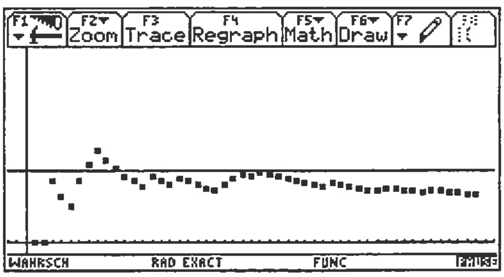
Kütting, Herbert: *Elementare Stochastik*; Akademischer Verlag heidelberg GmbH, Heidelberg-Berlin, 1999

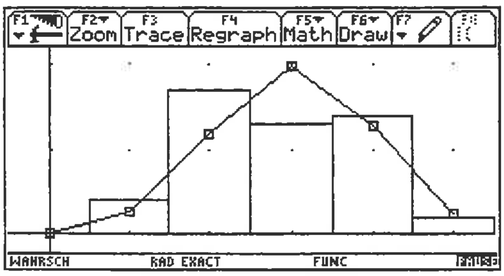
Scheid, Harald; *Stochastik in der Kollegstufe*; Bibliographisches Institut, Mannheim-Wien-Zürich, 1986

Some screen shots from my hypergeometric distribution simulation (Josef):









MONTE CARLO INTEGRATION WITH DERIVE

L. Monreal Mengual, S. Abraham Ibrahim, P. Fernández de Córdoba
Departamento de Matemática Aplicada
Universidad Politécnica de Valencia, Spain

Abstract

In these notes we introduce, with pedagogical aim, some ideas about Monte Carlo integration techniques. We illustrate them using *Derive* and we include some examples.

We introduce some simple ideas about Monte Carlo (MC) Techniques and illusrtate them using *Derive*. MC techniques include a lot of procedures using random numbers and their properties to evaluate multidimensional integrals, partial differential equations, etc., or simulate different types of problems.

In what follows we concentrate in using MC algorithms to integrate real functions in one dimension, i.e., we are going to face the evaluation of

$$\int_a^b f(x)\,dx) \qquad (1)$$

(for simplicity we are going to consider $f(x) > 0,\ \forall x \in (a,b)$)

We want to present, as an example, the evaluation of the number π. *via* the following expression

$$\pi = 4\int_0^1 1 - x^2\,dx \equiv 4\,I \qquad (2)$$

where

$$I \equiv \int_0^1 \sqrt{1-x^2}\,dx \qquad (3)$$

We are going to calculate I using MC algorithms. In particular, we are going to deal with the two basic procedures:

- The "Hit-or-Miss MC" method. This method is based in the following idea: if we consider a rectangular area including the surface under the function $f(x)$, then the integral of $f(x)$ in eq. (1) can be evaluated *via* the fraction of the rectangular area that is also under $f(x)$. Thus we generate a large number of random points in the rectangle and consider the number of them that are under $f(x)$ to calculate the integral:

  $$\int_a^b f(x)\,dx \sim \frac{NA}{NT}S$$

  where NA is the number of points under $f(x)$, NT is the total number of points generated, and S is the known area of the rectangle.

- The "Crude MC" method. This method consists in noting that eq. (1) can be related with the expecation value of the function:

  $$\int_a^b f(x)\,dx \sim \frac{b-a}{NT}\sum_i^{NT} f(x_i)$$

  where the set $\{x_i\!: i = 1,2,\ldots,NT\}$ includes NT values of the random variable x uniformly distributed in the range (a,b).

We have programmed in *Derive* both procedures to evaluate π. In **Appendix 1** we present *Derive* code including both methods. In **Appendix 2** we present the results for the evaluation of π as a function of *NT*.

In **Appendix 3** we present the Crude MC algorithm to evaluate

$$I_2 = \int_1^3 3x^2 + 3x + 2 \; dx \qquad (4)$$

(the exact result is $I_2 = 42$) and in **Appendix 4** the results for different number of trials.

Thus we can conclude that *Derive* is a useful tool to introduce MC techniques in a very easily way for the students of the first years in our universities.

The following is a *DERIVE*5 worksheet, including the tables provided by the authors.Josef

Appendix 1

```
#1:   Notation := Decimal
#2:                                    RANDOM(0) = 1307994398
```

Example: Evaluation of π

```
#3:   v(n) := VECTOR(RANDOM(1), i, n)
```

HIT-OR-MISS Method

Notice that n should be an even number

```
#4:   f(n) := VECTOR(IF((v(n))_i^2 + (v(n))_(i + 1)^2 < 1, 1, 0), i, 1, n - 1, 2)
```

$$\#5: \quad integral(n) := \frac{4 \cdot \sum_{i=1}^{n/2} (f(n))_i}{\frac{n}{2}}$$

```
#6:                                    integral(10) = 3.2
#7:                                    integral(100) = 3.36
```

CRUDE MONTE CARLO METHOD

```
#8:   Precision := Approximate
#9:   h(x) := √(1 - x^2)
```

$$\#10: \quad crudo(n) := \frac{4 \cdot \sum_{i=1}^{n} h((v(n))_i)}{n}$$

```
#11:                                   crudo(10) = 3.149004744
#12:                                   crudo(100) = 3.237198777
```

Appendix 2

NT	Hit-or-Miss Method	Crude Monte Carlo
10	3.2	2.99685
100	3.12	3.03939
1000	3.240	3.09994
10000	3.1272	3.14816

Appendix 3

#13: $v(n) := \text{VECTOR}(1 + 2\cdot\text{RANDOM}(1), i, n)$

#14: $h(x) := 3\cdot x^2 + 3\cdot x + 2$

#15: $$\text{crudo2}(n) := \frac{2\cdot \sum_{i=1}^{n} h((v(n))_i)}{n}$$

#16: crudo2(10) = 48.59232648

#17: crudo2(100) = 41.31503263

Appendix 4

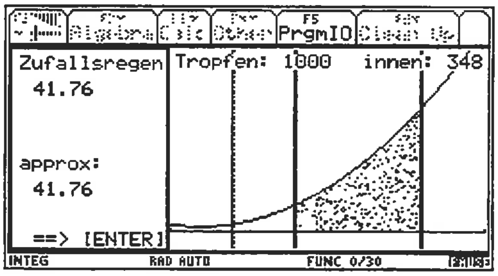
NT	Crude MC
10	44.2737
50	42.3374
100	39.6493
500	42.2144
1000	42.0075

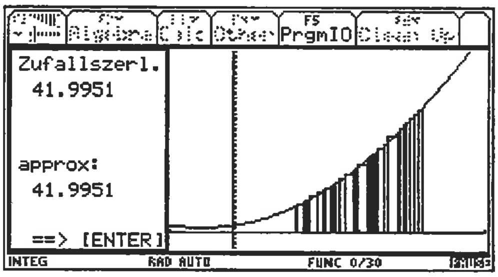
I´d like to remind you that in earlier DNLs we showed MC-Integration tools for DERIVE, too (DNL#8 & DNL#8, Terence Etchells and Josef Böhm).

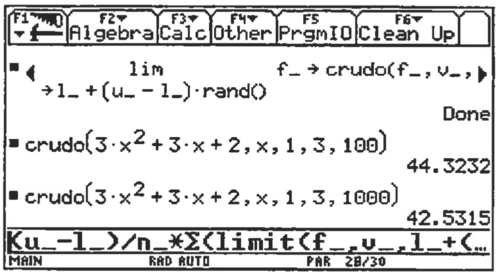
The crudo-function is easily to define on the TI.

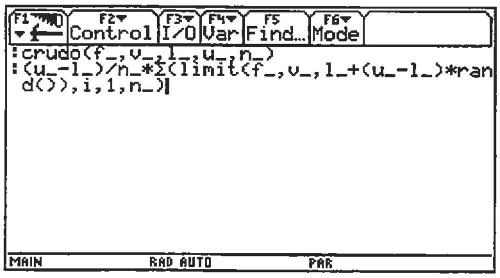
*The other screen shots are based on a program package integ(), which is available in a German and English version as well – **From Counting Raindrops to the Fundamental Theorem**.*

J.Böhm, W.Pröpper: Einführung des Integralbegriffs, 60 Seiten + Diskette
J.Böhm, W.Pröpper: Exploring Integration with the TI-89/92/92+, 60 pages + diskette
both available at bk-teachware (shop.bk-teachware.com)









Reinhard Schaeck schaeck@drei-eins-vier.de

Hello,

nice program, really. It shows some missing features in Derive, however. When I wanted to develop some functions in Derive (functions which do not just have 5 or 6 lines of code) I got nearly crazy. Take this easter-function as an example: When you want to make a small change, the only way to do it (inside Derive) is to transfer the whole code to the input line (via F3) where it becomes one very long string.

There seems to be no easy way to edit such funtions (I know I could copy the mth file to an external editor and paste it back, but that's not very comfortable, too). If you want to look at an intermediate result (let's say check the variable tec in line 4) you have problems too, if you don't want to cut the code into pieces.

I don't like general comparisons between Derive et al. but when I compare this special situation with the possibilities you have f.e. in MATLAB, then it's a big difference which kept me from using Derive in many situations (though I worked through the Website about Derive-Programmin at http://www.cms.livjm.ac.uk/deriveprogramming/)

Can we expect any improvements in this area soon?

Wim de Jong

Reinhard,

I appreciate your misgivings about the occasional complexity of a function definition presented in this discussion forum. The problem appears to be that there are, roughly speaking, two different kinds of users of DERIVE: those who want to use DERIVE to find solutions to their own mathematical problems (and may have some expertise in computer programming) and those (often without programming expertise to ' write home about') who want to use DERIVE as an educational tool. I belong to the latter category and do normally not go out of my way to formulate DERIVE functions as concisely as possible. On the contrary, I prefer multi-line definitions if they enhance the transparency of the approach. Intricacies of the system should not overshadow the process of learning mathematics. I do not object to more compact definitions as long as they are accompanied by a clear documentation.

Originally computer algebra systems such as REDUCE, AXIOM and MATHEMATICA were designed to solve research problems, but since the mid-1980's systems such as MAPLE and DERIVE were designed for the purpose of mathematics education. I am a DERIVE addict because it allows anyone with an interest in mathematics to particicipate in computer-based investigations of mathematical problems without being hampered by arithmetical or algebraic drudgery.

Despite this preference I hope that members of both the 'utilitarian' and the 'educational' camp will continue to contribute vigorously to the e-mail discussions because, at the end of the day, we are all interested in the full potential of DERIVE.

Cheers, Wim de Jong

Theresa Shelby, Hawaii tshelby@ti.com

Hi Derivers of all types,

An optional multi-line editing window is one of the top items on the "wish-list" for a future version of Derive.

Regards,
Theresa Shelby

Valeri Anisiu

Hello Reinhard,

Consider for the moment these programming tips:

1. In DERIVE 5 you can edit "in place" any (sub)expression, including arguments of a PROG or LOOP function. It is sufficient select it with the mouse (with repeated clicks if necessary) and then invoke the Edit/Derive Object in the menu (or you can use the Context Menu using a right click).

2. If you want to inspect the variable `tec`, you can change (temporarily) the expression

```
tec := something
```

to

```
[tec:=something, test:=tec]
```

Then, after the call, you can inspect the (global) variable `test`.
If the variable `tec` would be inside of a loop, you could initialize at the comman level, `test:=[]`, and then

```
[tec:=something, test:=APPEND(test, [tec])]
```

to obtain all the successive values of `tec`.

3. To construct comfortably a longer program in DERIVE, you can use for example a skeleton like:

```
myprog(x,y,z) := PROG(1,2,3,4, "loop1","loop2", IF(x>0,"c1","c2"),5)
```

and then edit the expressions 1,2,3,4,"loop1","loop2" etc.

Cheers, Valeriu

Josef Böhm

Additional tipps to overcome the lack of a multiline editor

I'll take an easy example to show the principles. Let's produce a financial mathematics table. A given principal *c* is increased each period by an amount *a* earning compound interests *i* per period. Give a table showing the # of period and the respective capital. (I know that experienced users will solve the problem using a One-Line-ITERATES-Construction).

I want to force an output with two decimal digits, so I define a function `round(x,m)` to round number x on m decimal places.
That's no problem at all:

```
round(x, m) :=
FLOOR(10^m·x + 0.5)/10^m
```

If you would enter the program via the expression line, it is – although this is a very short program – not so easy not to miss any comma, parenthesis, bracket,......

FUTUREV(4000, 2000, 5, 8) =

Period	Value
0	4000
1	6200
2	8510
3	10935.5
4	13482.28
5	16156.39
6	18964.21
7	21912.42
8	25008.04

This is the program

```
FVPROG(pv,add_pmt,perc,periods,n:=0,cap,values):=PROG(cap:=pv,values:=[[n,cap]],LOO
P(n:+1,cap:=(1+perc/100)*cap+add_pmt,values:=APPEND(values,[[n,round(cap,2)]]),IF(n
=periods,RETURN(APPEND([["Period","Value"]],values))))
```

From my experience I recommend doing as follows. Open your texteditor and type in the program in a structured form, as you like. I like this structure and layout:

```
FUTUREV(pv,add_pmt,perc,periods,n:=0,cap,values):=
          PROG(
               cap:=pv, values := [[n,cap]],
               "@ Start of the loop",
               LOOP(
                 n :+ 1,
                 cap := (1+perc/100) cap + add_pmt,
                 values := APPEND(values,[[n,round(cap,2)]]),
                 IF(n=periods,RETURN APPEND([["Period",
                                          "Value"]],values)),
                "@ End of the loop"
               )
          )
```

You can set line breaks wheresoever you want. I included comments – marked by a leading @. Then paste the whole text via the clipboard into the edit line of *DERIVE*, press enter – and if the expression is syntactically correct, it will appear highlighted in the *DERIVE* screen. Now test the program.

This works very well and is not too boring, because you have the text editor and *DERIVE* open simultaneously so and can easily switch from one window to the other and back again. In this way you can transfer only one program or function after the other.

There is another way to produce a file consisting of more than only one function (program, definition, expression).

```
round(x,m):= FLOOR(10^m x+0.5)/10^m

FUTUREV(pv,add_pmt,perc,periods,n:=0,cap,values):=~
         PROG(~
              cap:=pv, values := [[n,cap]],~
              LOOP(~
                n :+ 1,~
                cap := (1+perc/100) cap + add_pmt,~
                values := APPEND(values,[[n,round(cap,2)]]),~
                IF(n=periods,RETURN APPEND([["Period",~
                                          "Value"]],values))~
              )~
         )
```

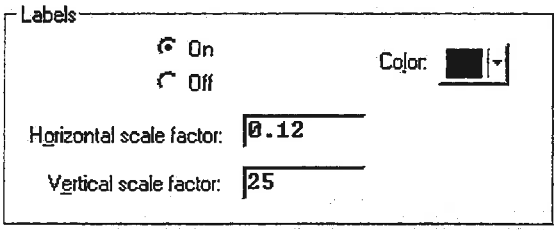
You can see the difference: the functions (programs, expressions) are separated by a blank line and the line break within a program is marked by a "~".

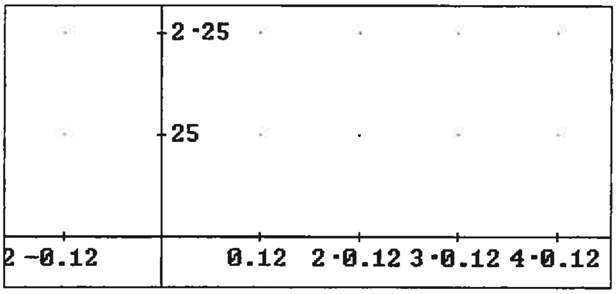
Save this file as a text file and then open it in *DERIVE*. Take care that you will not find it listed among the *.dfw and *.mth-files. It is a txt-file, but if syntactically correct, it should work.

Two more tipps for *DERIVIANs*

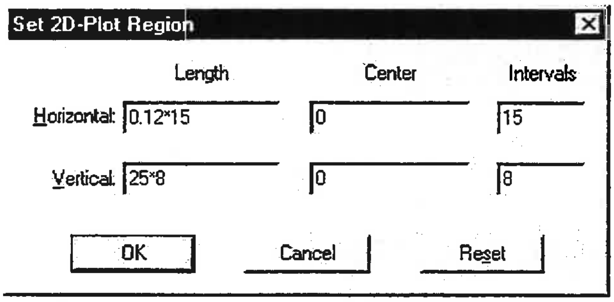
Mr Decho, a very enthusiastiC *DERIVE* user from Upper Austria calls often asking for help. One of his question was the following:

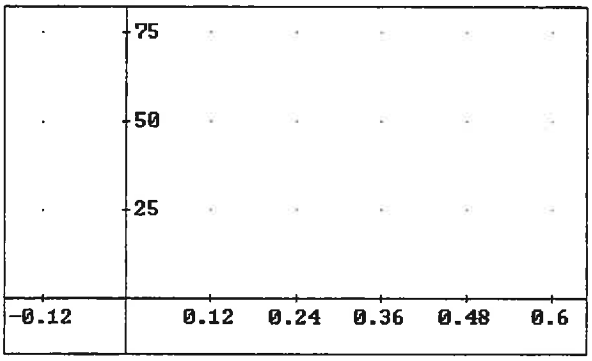
Sometimes it might be necessary to have a very special scaling on the axes, eg. 0.12 horizontal and 25 vertical. Most people try setting the scale factors in the Display Options, receiving very bulky labels.



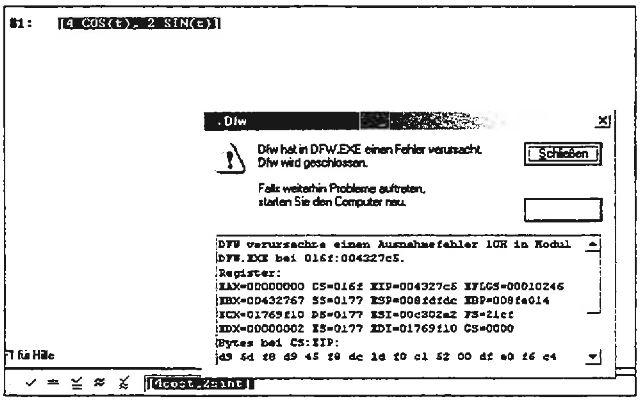


Follow the Albert Rich's advice: *Set the Scaling Factors 1 and then use the Set > Plot Region command to set the Horizontal Length to n*0.12, where n is the number of horizontal intervals which is also specified in the command. Then do accordingly for the Vertical Length.*





I received some claims on error messages at the occasion of plotting functions which are followed by a restart of *DERIVE*. Theresa Shelby's answer is:



Microsoft documentation states this about the error message:

"16 (10h): Coprocessor error Fault

This interrupt occurs when an unmasked floating-point exception has signaled a previous instruction. (Because the 80386 does not have access to the Floating Point Unit, it checks the ERROR\ pin to test for this condition). This is also triggered by a WAIT instruction if the Emulate Math Coprocessor bit at CR0 is set."

This is the same sort of error we used to experience in Derive 4 when a "naughty" printer driver would overwrite the floating point word before we used it to print the plot window. We "fixed" this problem by initializing the floating point word ourselves within Derive. However, it has recently come to my attention (post Derive 5.05) that on Windows 2000, our earlier "fix" has been "unfixed" by the operating system and we are once again at the mercy of a bad driver. The good news is that I think I have "re-fixed" the problem for the next Derive 5 point release.

So, I'd recommend to your user to update to Derive 5.05 if they are not already. If it still happens, please find out 1) the operating system the user is running and; 2) the exact steps to reproduce the error.

Four Calendar Functions

Richard Schorn, Kaufbeuren , Richard.Schorn@t-online.de

Two persons (perhaps father and son) wish to determine the date when the older is twice as old as the younger. (The question was put in a mathematical newsgroup with the birthdates January 10, 1930 and February 25, 1966)

With DERIVE it is easy to find a solution:

The first program calculates the "true" Julian Date for a given day, month and year. This number gives the total number of days that have elapsed since 4713 B.C. January 1. As this number has now reached a very high value the astronomers invented a "modified Julian Date" (MJD) starting at 1858 November 17, 00:00 hours. With MJD one can integrate hours, minutes,seconds, milliseseconds ...to give a exact date with time [1]. Some programs (i.e. Excel) use nearly the same system with a different starting date.

Program `JDAY(d,m,y)` is adapted from an ALGOL-program by Robert G.Tantzen, ACM Communications 1963, see [2, page 125, ALGOL-Algorithmen für den Julianischen Tag].

Look at the trick with `Prog(m:=m+9,y:=y-1,` when the ELSE-clause of an `IF()` must have more than one statement! Thanks to Josef for this nice idea!

```
JDAY(d, m, y) :=
  Prog
    If m > 3
       m := m - 3
       Prog
         m := m + 9
         y := y - 1
    a := MOD(y, 100)
    c := FLOOR(y/100)
    RETURN FLOOR(146097·c/4)+FLOOR(1461·a/4)+FLOOR((153·m+2)/5)+d+1721119
```

Example:

```
JDAY(27, 1, 2002) = 2452302
```

The second program **Day Of** the **Week** (`DOW(d,m,y)`) is not necessary for the solution of the original question, but I think it to be simple and nice.

```
DOW(d, m, y) :=
  Prog
    z := MOD(JDAY(d, m, y), 7) + 1
    RETURN ["Mo", "Tu", "We", "Th", "Fr", "Sa", "Su"]↓z
```

Examples:

```
DOW(27, 1, 2002) = Su

DOW(30, 4, 1777) = We "Birthday of Carl Friedrich Gauß"
```

The third program is the inverse to the first one, `JDATE(j)` calculates the current date from the Julian Date. It is also adapted from an ALGOL-Program [2].

```
JDATE(j) :=
  Prog
    j := j - 1721119
    y := FLOOR((4·j - 1)/146097)
    j := MOD(4·j - 1, 146097)
    d := FLOOR(j/4)
    j := FLOOR((4·d + 3)/1461)
    d := MOD(4·d + 3, 1461)
    d := FLOOR((d + 4)/4)
    m := FLOOR((5·d - 3)/153)
    d := MOD(5·d - 3, 153)
    d := FLOOR((d + 5)/5)
    y := 100·y + j
    h := m + 3
    If m ≥ 10
       h := m - 9
    If m ≥ 10
       y := y + 1
    m := h
    RETURN [d, m, y]
```

Here you see the clumsy way for two statements with the "same" IF, when **not** using Josefs trick mentioned earlier.

The solution of the original question is straightforward:

```
x - JDAY(10, 1, 1930) = 2·(x - JDAY(25, 2, 1966))
SOLVE(x - JDAY(10, 1, 1930) = 2·(x - JDAY(25, 2, 1966)), x)
                                  x = 2452377
                     JDATE(2452377) = [12, 4, 2002]
```

The fourth program `FRIDAY(e)` renders the solution to the well-known "Friday the Thirteenth"-Problem [3]. According to [2, page 34] all "Georgian Periods" (400 years with 7*20871 = 146097days) have the same connection "day of the week – date".
Because such a period comprises 400*12 =4800 months (**not** divisible by 7) some "thirteenths" **must** appear more often than others.
The program is a slight variation of `JDATE(j)`, only the occurrence of the weekdays falling on a 13th are counted in array T (Monday through Sunday).

The program is among the files to be downloaded from one of the homepages given on the first page.

```
FRIDAY(146097) = [685, 685, 687, 684, 688, 684, 687]
```

[1] Oliver Montenbruck – Thomas Pfleger, Astronomy on the Personal Computer, Springer-Verlag, 1991, page 12

[2] Heinz Zemanek, Kalender und Chronologie, Oldenbourg München-Wien 1981,

[3] Journal of Recreational Mathematics, vol 30, Number 3, 1999-2000, page 223, Problem 2534

Some Statistics – Tools for *DERIVE* and the *TI*-89/92 Part 2

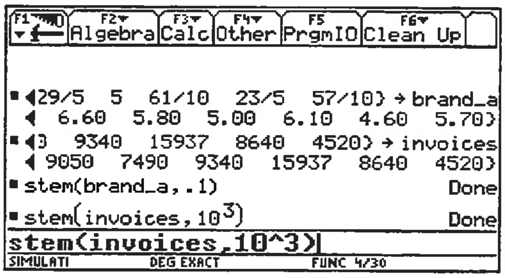
Josef Böhm, Würmla, Austria

Until now we have missed two diagram forms, the Stem & Leaf Diagram – for *DERIVE* and the TI – and the Box Plot for *DERIVE*. So let´s fill this gap.
We would like to compare the distribution of the lifetimes of two brands [1].

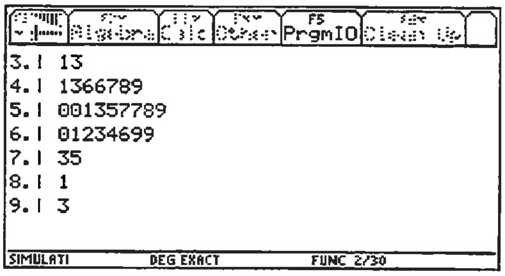
```
brand_a := [5.1, 7.3, 6.9, 4.7, 4.6, 6.2, 6.4, 5.5, 4.9, 6.9, 6, 4.8, 4.1, 5.3, 8.1,
  6.3, 7.5, 5, 5.7, 9.3, 3.3, 3.1, 4.3, 5.9, 6.6, 5.8, 5, 6.1, 4.6, 5.7]

brand_b := [5.4, 6.3, 5, 5.9, 5.6, 4.7, 6, 3.3, 6.6, 6, 5, 6.5, 5.8, 5.4, 4.9, 5.7, 6.8,
  5.6, 4.9, 6, 4.9, 5.7, 6.2, 7.5, 5.8, 6.8, 5.9, 5.3, 5.6, 5.9]

[stem(brand_a, 0.1), stem(brand_b, 0.1)]

 3 : 13              3 : 3
 4 : 1366789         4 : 7999
 5 : 001357789       5 : 003446667788999
 6 : 01234699    ,   6 : 000235688
 7 : 35              7 : 5
 8 : 1
 9 : 3
```

`stem(list,decimal_place_of_leaves)` gives a Stem & Leaf Diagram. The program was not so easy to write because of the necessary included string manipulations. I integrated a rounding procedure and now it is possible to not only consider the last decimal place of the data:

```
invoices := [10390, 7530, 10250, 4785, 10395, 11410, 12334, 3470, 2950, 21200, 8512,
  7000, 10150, 7824, 9620, 12730, 9345, 9827, 6000, 12725, 12260, 7710, 6260, 4820,
  7360, 6983, 7290, 10865, 24020, 6965, 8580, 8550, 8500, 25650, 11860, 6210, 13610,
  16042, 9240, 17140, 9340, 9748, 11644, 8030, 9050, 7490, 9340, 15937, 8640, 4520]

                          0 : 3355566677777788889999999999
stem(invoices, 1000) =    1 : 0000000112222334667
                          2 : 146
```

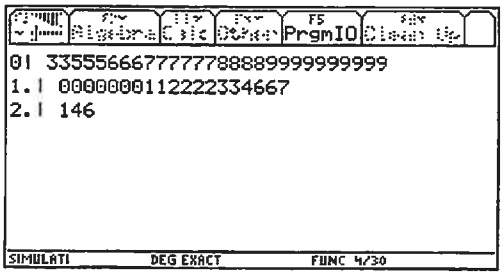


Download `stem()` from one of the two websites and you have not to do without a Stem & Leaf Diagram on the TI any longer.

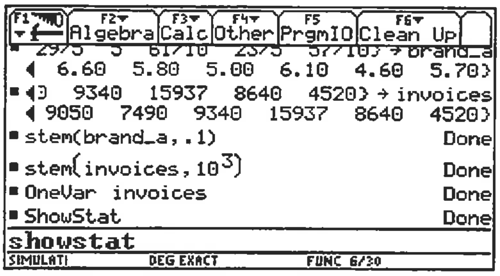


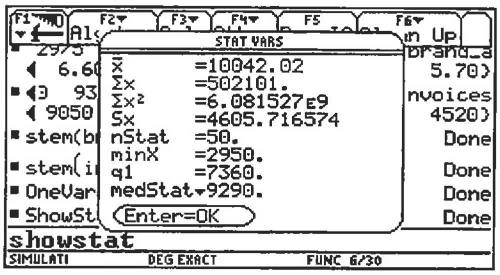
brand_a

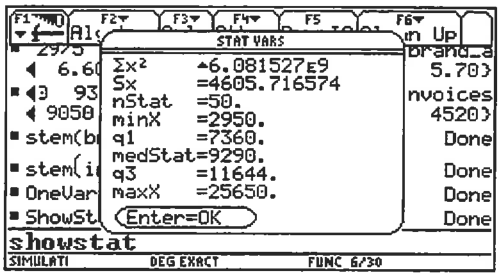
Stem & Leaf for invoices



I assume that most of you who are working with the TI-89/92 are familiar with the TI´s One Variable Statistics: You can call the One Var Stats from the `Data/Matrix Editor` or even from the Homescreen







We will tranfer this to *DERIVE* and add more useful statements about the data, as the mode – the most frequent occuring data(s), the "density region" (see below) and the number of outliers.

We provide the following functions which later will be combined to a common output.

`ONE_VAR_STATS(liste) :=`

Number of Data:	DIMENSION(liste)
Mean:	AVERAGE(liste)
Standard Deviation:	APPROX(STDEV(liste))
Minimum:	MIN(liste)
1.Quartile:	QUART1(liste)
Median:	MEDIAN(liste)
3.Quartile:	QUART3(liste)
Maximum:	MAX(liste)
dense between:	REGION(liste)
# of outliers:	OUTL(liste)
Mode:	MODUS2(liste)

The "density region" is the intervall [Q1 – 1.5 IQR, Q3 + 1.5IQR] with IQR = Interquartile Range = = Q3 – Q1. All elements lying beyond are "outliers".

`ONE_VAR_STATS(invoices) =`

Number of Data:	50
Mean:	10042.0
Standard Deviation:	4605.7
Minimum:	2950
1.Quartile:	7360
Median:	9290
3.Quartile:	11644
Maximum:	25650
dense between:	[934, 18070]
# of outliers:	3
Mode:	[9340]

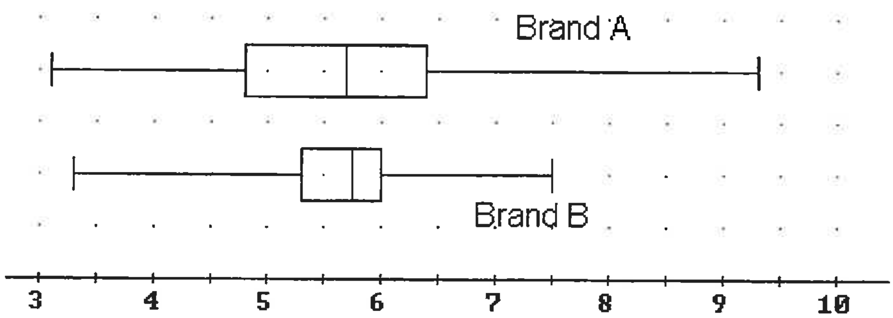
One comment on the *mode*: There are controversies in the Statsbooks defining the *mode*. Some insist on the fact that there is one unique *mode* (the most frequent element of the list) or there is non at all, but others accept that there might be more most frequent *modes* (e.g. having a bimodular distribution). `MODUS(list)` satisfies the first opinion and `MODUS2(list)` the second one.

Now we have all values in order to produce a Box Plot on the *DERIVE*-screen. Unfortunately we cannot plot the Modified Boxplot using only one single command, because we have to change the Plot Points Setting from Connected and Small to Not Connected and Large. One of points of my wishlist for the next *DERIVE* version is to have commands to set the various plot options from within a program.

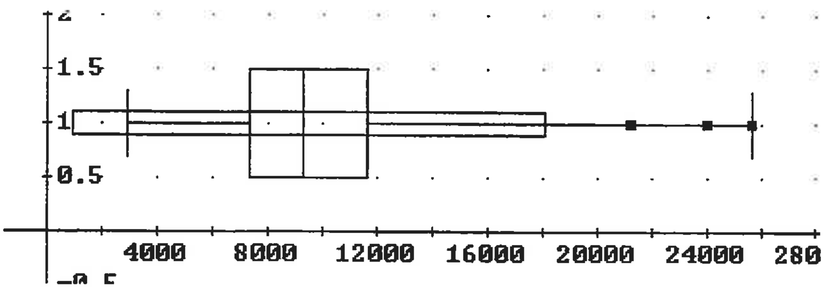
The following procedures are applicable:

`BOXPLOT(list, distance_from_x_axis)`	the common box plot diagramm
`DENSEPLOT(list, distance)`	the "density region is specially marked
`OUTPLOT(list, distance)`	list of the outliers
`MODBOXPLOT(list, distance)`	box plot & outliers (highlight and plot the two results separately)

[BOXPLOT(brand_a, 4), BOXPLOT(brand_b, 2)]

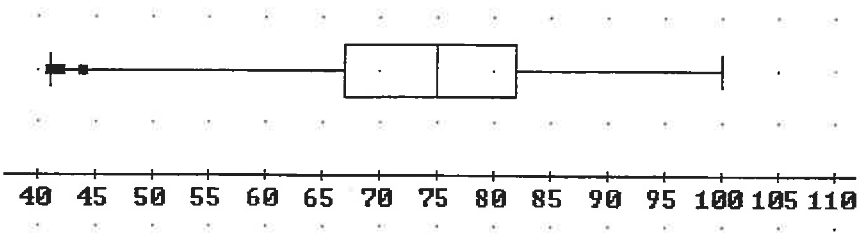


BOXPLOT(invoices, 1)
DENSEPLOT(invoices, 1)
OUTPLOT(invoices, 1)



And finally the Box Plot of the 2600 SAT Scores from DNL#45.

[BOXPLOT(scores, 2), OUTPLOT(scores, 2)]



Just in these days – collecting the stuff for *DNL*#46 – I received a mail from K.-H. Keunecke asking if there is any utility file providing easy to use functions for various regression lines – very similar to the tools on the TIs. At the moment one can use the FIT-function as described in the Help-file, but we don´t have a Log Reg, Exp Reg, Power Reg, Trig Reg and Logistic Reg on a keypress.
Here is help.
In an earlier *DNL* (*DNL* #41, March 2001) I introduced the MedMed-Regression.

We will give all Regression outputs a similar form: show the regression function, present the SSE (Sum of Squared Errors, give the Correlation Coefficient in case of linear regression (or related) and give the Coefficient of Determination. You can find the common structure of all programs in the `linreg(data_matrix)` program below.

```
linreg(a_, i, lreg, sse, av, sst, best) :=
  Prog
    lreg := FIT([x, a·x + b], a_)
    sse := Σ((a_↓i↓2 - LIM(lreg, x, a_↓i↓1))^2, i, 1, DIMENSION(a_))
    av := AVERAGE(a_ COL 2)
    sst := Σ((a_↓i↓2 - av)^2, i, 1, DIM(a_))
    best := 1 - sse/sst
    ["Regression Line:", lreg; "SSE:", sse; "CorrCoeff:", √best; "DetCoeff:", best]
```

What to do, if the data are given in rows (and not in colums as usal) together with labels?

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

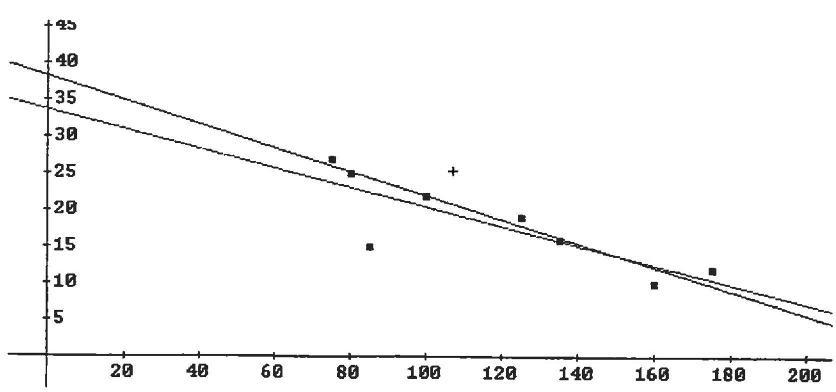
```
hp_tab := hp` ROW [2, ..., DIM(hp`)]
```

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

We compare the MedMed – Regression and Linear Regression

$$\begin{bmatrix} medreg(hp\_tab) \\ linreg(hp\_tab) \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \text{MedMed Regression Line:} & 38.2604 - 0.1625\cdot x \\ \text{SSE:} & 101.250 \end{bmatrix} \\ \begin{bmatrix} \text{Regression Line:} & 33.7073 - 0.132255\cdot x \\ \text{SSE:} & 83.7660 \\ \text{CorrCoeff:} & 0.822922 \\ \text{DetCoeff:} & 0.677201 \end{bmatrix} \end{bmatrix}$$

The MedMed-Line is the steeper one. It is much more robust with regard to the outlier(s).



To save space I give the data again in rows and present a `cubicreg(data)`. My package contains also `quadreg(data)` and `quartreg(data)`.

census :=

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
7.24	9.64	12.87	17.07	23.19	31.44	39.82	50.16	62.95	75.99	91.97	105.71	122.78	131.67	151.33	179.32	203.21	226.5

cubicreg(census`) =

CubicReg Line:	$0.00605019\cdot x^3 + 0.488870\cdot x^2 + 1.53201\cdot x + 4.05241$
SSE:	123.725
DetCoeff:	0.998516

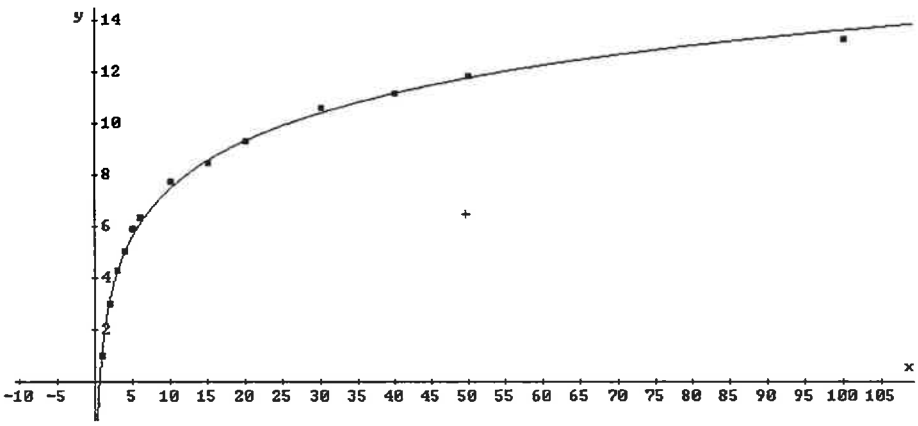
Some of you might remember the SNOG of *DNL*#39 which led to a logarithmic regression on the TI.

snog :=

1	2	3	4	5	6	10	15	20	30	40	50	100
1	3	4.3	5.04	5.92	6.35	7.7	8.43	9.27	10.56	11.12	11.82	13.22

lnreg(snog`) =

Regressionslinie:	$2.66132\cdot LN(x) + 1.33667$
SummeRes^2:	0.542493
KorrKoeff:	0.998308
BestMaß:	0.996619



The easy to perform regressions (based on linear regression) end by presenting the Exponential Function Regression line and the Power Function Regression line.

expreg(census`) =

ExpReg Line:	$8.15994\cdot ê^{0.200785\cdot x}$
SSE:	9848.98
DetCoeff:	0.881935

Pwrreg(census`) =

PowerReg Line:	$3.78842\cdot x^{1.32284}$
SSE:	6447.86
DetCoeff:	0.922706

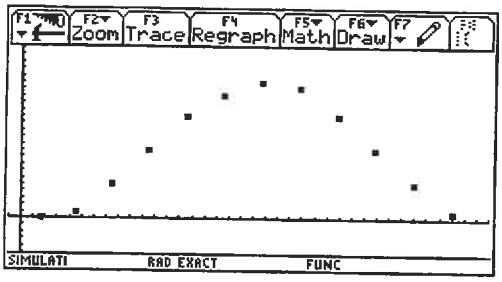
There are two more regressions implemented on the TI-92PLUS, which are really useful: the trigonometric regression `SinReg` and the `Logistic Regression`.

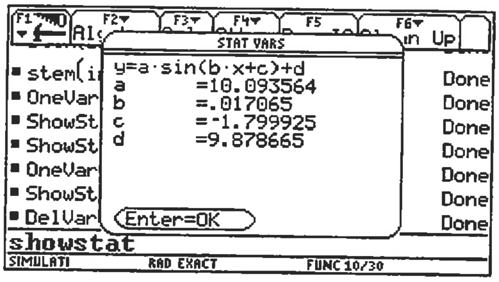
THis is a table of the average temperature of a small village near Vienna:

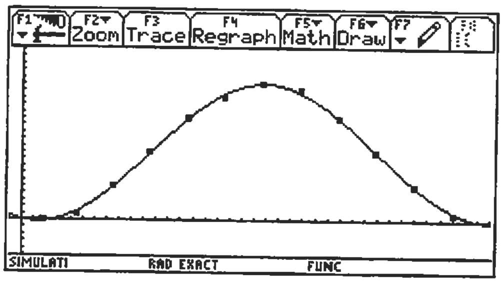
temptab :=

16	45	75	105	136	166	197	228	258	289	319	350
0	1	5	10	15	18	20	19	15	10	5	1

These are the mean temperatures for the twelve months and they are assigned to the day in the middle of the month. Day#75 is March 16. Plotting the temperature vs time, we immediately get the idea to underly a trig function.

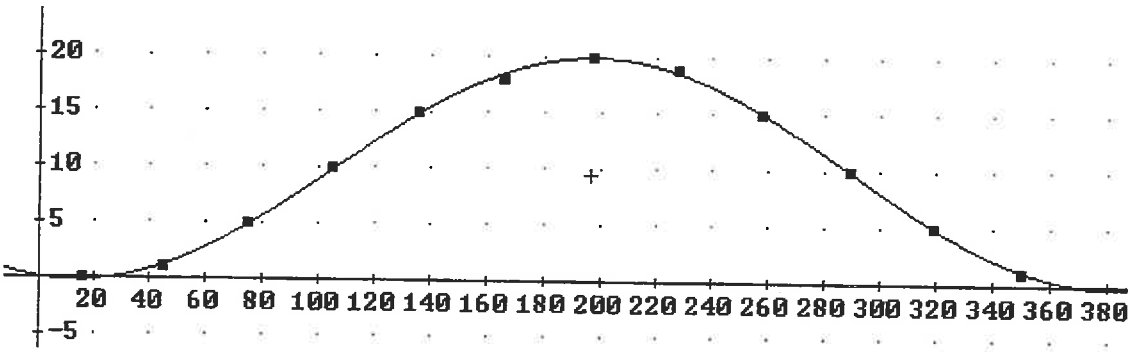






Performing a trigonometric regression is not an easy task, because of the very complicated system of nonlinear equations. There are algorithms, which can be found in Numerical Mathematics textbooks. I asked several people and received valueable advice from David Stoutemyer and MacDonald R. Phillips who sent a file "Nonlinear Regressions" programming the *Gauss-Newton-Method* and the *Marquardt-Levenberg-Algorithm*. I tried to convince Don´s very extended program to produce an output in a desired form – and it works:

```
sinreg(temptab`) = [ SinReg Line:   y = 10.0935·SIN(0.0170651·x + 4.48326) + 9.87866 ]
                   [     SSE:                          0.64126                       ]
```



Unfortunately I was not able to produce a working logistic regression until now. Maybe that there is a member within the *DERIVE* community who can help to really fill the gap Karl-Heinz spoke about. *Latest news: Things have changed. Logisitc Regression is working. You will find it included in the package.*

[1] Investigating Statistics with the TI-92, Brendan Kelly, Brendan Kelly Publishing Inc. 1977

[2] Statistik für Sozial und Wirtschaftswissenschaften, Hackl & Katzenbeisser, Oldenbourg, 2000

Titbits from Algebra and Number Theory (22)

by Johann Wiesenbauer, Vienna

Beatty sequences, do they ring a bell for you? Maybe you are thinking of film sequences with Warren Beatty, but this isn't exactly what I mean here. I'm rather talking about Sam Beatty who made the following intriguing discovery in 1929: If α and β are arbitrary irrational numbers > 1, such that

$$\frac{1}{\alpha}+\frac{1}{\beta}=1 \qquad (*)$$

then for any positive integer k exactly one of the following two statements is true:

1. There is a unique positive integer n such that $n\alpha \in (k, k+1)$.

2. There is a unique positive integer n such that $n\beta \in (k, k+1)$.

For example, let's check this for $\alpha=\sqrt{2}$ and $\beta=2+\sqrt{2}$. In the first place, (*) is fulfilled due to the following computation in Derive:

```
[α := √2, β := 2 + √2]

1     1
─  +  ─  = 1
α     β
```

Now it is easy to see that condition 1 is true for a fixed positive integer k if and only if the interval

$$I_k=\left(\frac{k}{\alpha},\frac{k+1}{\alpha}\right)$$

contains an integer n. Since the length of this interval is $1/\alpha$ and hence <1, n must be necessarily unique. Similarly condition 2 is true for the same k if and only the interval

$$J_k=\left(\frac{k}{\beta},\frac{k+1}{\beta}\right)$$

contains an integer, which must be also unique then. Now it's very important that you really understand how these checks are carried out for a given k in the following program, namely

$$\text{Cond. 1} :\Leftrightarrow I_k \cap N^* \neq \varnothing \Leftrightarrow MOD(k/\alpha) > MOD((k+1)/\alpha)$$

and

$$\text{Cond. 2} :\Leftrightarrow J_k \cap N^* \neq \varnothing \Leftrightarrow MOD(k/\beta) > MOD((k+1)/\beta)$$

respectively, where MOD(x) is the Derive-function that returns the fractional part of x (sometimes also denoted by {x} or FRAC(x)) and N* ={1,2,3,...} is the set of positive integers.

```
beatty(s, α, β, k_ := 1, u_, v_, w_ := [[k, "I"↓k, "J"↓k, "Cond 1", "Cond 2"]]) :=
  Loop
    If k_ > s
       RETURN REVERSE(w_)
    u_ := [k_, k_ + 1]
    v_ := [k_, u_/α, u_/β, MOD((u_/α)↓1) > MOD((u_/α)↓2), MOD((u_/β)↓1) > MOD((u_/β)↓2)]
    w_ := ADJOIN(v_, w_)
    k_ :+ 1
```

```
beatty(10, √2, 2 + √2)
```

k	I_k	J_k	Cond 1	Cond 2
1	[0.7071067811, 1.414213562]	[0.2928932188, 0.5857864376]	true	false
2	[1.414213562, 2.121320343]	[0.5857864376, 0.8786796564]	true	false
3	[2.121320343, 2.828427124]	[0.8786796564, 1.171572875]	false	true
4	[2.828427124, 3.535533905]	[1.171572875, 1.464466094]	true	false
5	[3.535533905, 4.242640687]	[1.464466094, 1.757359312]	true	false
6	[4.242640687, 4.949747468]	[1.757359312, 2.050252531]	false	true
7	[4.949747468, 5.656854249]	[2.050252531, 2.343145750]	true	false
8	[5.656854249, 6.363961030]	[2.343145750, 2.636038969]	true	false
9	[6.363961030, 7.071067811]	[2.636038969, 2.928932188]	true	false
10	[7.071067811, 7.778174593]	[2.928932188, 3.221825406]	false	true

When looking at this table this is the question I want you to ask the following question: How come that the **fractional parts** of the boundaries of the first interval are increasing or decreasing just in the opposite way as for the second interval?? After all, this makes sure that exactly one of the two conditions 1 and 2 above is fulfilled. Try to find this out yourself! You got it? Great! In fact, if we view I_k and J_k as 2-element vectors for the following addition, this is due to

$$I_k + J_k = \left(\frac{k}{\alpha}, \frac{k+1}{\alpha}\right) + \left(\frac{k}{\beta}, \frac{k+1}{\beta}\right) = \left(k\left(\frac{1}{\alpha}+\frac{1}{\beta}\right), (k+1)\left(\frac{1}{\alpha}+\frac{1}{\beta}\right)\right) = (k, k+1)$$

because this shows that the fractional parts of the interval boundaries obey the rules

$$\text{MOD}(k/\alpha) + \text{MOD}(k/\beta) = 1 \text{ and } \text{MOD}((k+1)/\alpha) + \text{MOD}((k+1)/\beta) = 1$$

(ckeck this in the table above!) and hence

$$\text{MOD}(k/\alpha) < \text{MOD}((k+1)/\alpha) \Leftrightarrow \text{MOD}(k/\beta) = 1 - \text{MOD}(k/\alpha) > 1 - \text{MOD}((k+1)/\alpha) = \text{MOD}((k+1)/\beta)$$

What appeared to be a mistery at first glance turned out to be actually an easy consequence of (*) !

By defining what we now call Beatty sequences, namely the integer sequences

$\lfloor\alpha\rfloor, \lfloor 2\alpha\rfloor, \lfloor 3\alpha\rfloor, \ldots$ and $\lfloor\beta\rfloor, \lfloor 2\beta\rfloor, \lfloor 3\beta\rfloor, \ldots$

we can reformulate the statement above in the following form: Under the assumption (*) these two sequences together contain ever positive integer k exactly once, i.e. the corresponding multisets

$\{\lfloor n\alpha\rfloor \mid n \in \mathbf{N}^*\}$ and $\{\lfloor n\beta\rfloor \mid n \in \mathbf{N}^*\}$

are actually sets (which means that no element occurs more than once in them!) and form a partition of the set $\mathbf{N}^*=\{1,2,3,\ldots\}$ of positive integers.

Does this ring a bell now? If not, then it is clear that you haven't read the nice article by Richard Schorn about Wythoff's Nim in the last DNL (cf. [2]), which is really a pity. (Since I occasionally refer to its contents in the following, I strongly recommend reading that paper before continuing!)

Anyway, it turned out that the Beatty sequences with $\alpha = g$ and $\beta = g^2$, where

$$g = \frac{1+\sqrt{5}}{2}$$

is the golden ratio, play a key role in the study of that game. Indeed, as the following Derive-computation shows, these values of α and β fulfill our condition (*):

$$\left[g := \frac{1+\sqrt{5}}{2},\ \alpha := g,\ \beta := g^2\right]$$

$$\frac{1}{\alpha} + \frac{1}{\beta} = 1$$

Pedro Tytgat raised the interesting question on the Derive-Forum (included somewhere in this issue!) how that ubiquitous golden ratio g comes into play here and in the following I'm going to answer it in more detail now.

First let's have a look at the first 20 safe cells (x,y) with 0<x<y of Wythoff's Nim, which were produced by the following program. (Programming novices will learn a lot, if they study the differences between Richard's version, my first version on the forum and this one!)

```
wyt(s, i_ := 1, m_ := {}, w_ := [[i, x↓i, y↓i]], x_ := 1) :=
  Loop
    If i_ > s
       RETURN REVERSE(w_)`
    w_ := ADJOIN([i_, x_, x_ + i_], w_)
    m_ := m_ ∪ {x_ + i_}
    i_ :+ 1
    Loop
      x_ :+ 1
      If MEMBER?(x_, m_)
         m_ := REST(m_)
         exit
wyt(20)
```

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x_i	1	3	4	6	8	9	11	12	14	16	17	19	21	22	24	25	27	29	30	32
y_i	2	5	7	10	13	15	18	20	23	26	28	31	34	36	39	41	44	47	49	52

Here the first line is a running number i=1,2,3,..., which is also the difference between x_i and y_i, i.e. $y_i = x_i + i$ ($i \in N^*$). Furthermore each x_i is simply the smallest positive integer that has not shown up among the x- and y-coordinates computed so far. In particular, $x_1 = 1$.

As I already pointed out in my first answer this table abounds with Fibonacci sequences, i.e. sequences $F_0, F_1, F_2,..$ obeying the recursive law $F_n = F_{n-1} + F_{n-2}$, n=2,3,4... (Note that we are committing a slight abuse of language and notation, because unlike the "classical case", where $F_0 = 0$, $F_1 = 1$, the iniatial values F_0 and F_1 may be arbitrary, though fixed nonnegative integers here!)

For example, by considering the x- and y-coordinates in the columns numbered 1,2,5,13,... you get the Fibonacci sequence 1,2,3,5,8,13,21,34,...The first pair not "covered" is (4,7). Use it to start another Fibonacci sequence 4,7,11,18,29,47,...Going on in this way it is easy to see that our table is entirely made up of Fibonacci sequences which do not overlap. Now everyone who knows the very basic facts about Fibonacci sequences will know how the golden section g comes into play. In fact, it is easy to prove that any Fibonacci sequence can be obtained by the formula

$$F_n = ag_1^n + bg_2^n$$

where $g_1 = g = (1+\sqrt{5})/2 \approx 1.618033988$ and $g_1 = (1-\sqrt{5})/2 \approx -0.6180339887$ are the two roots of the quadratic equation $x^2 - x - 1 = 0$ and a,b are constants such that

$$a + b = F_0$$

$$ag_1 + bg_2 = F_1$$

Because of $|g_2| < 1$ we have $F_n \approx ag^n$ for large n. In particular, we can always get F_{n+1} from F_n by simply rounding gF_n to the next integer, if n is sufficiently large. For example, if $gF_0 < F_1$, then we have to compute CEILING(gF_n) for even n and FLOOR(gF_n) for odd n corresponding to the alternating sign of the second term bg_2^n in the formula of F_n above.

Now suppose that $F_{n+1} = \lfloor gF_n \rfloor$ for some n. Then due to

$$\lfloor g^2F_n \rfloor = \lfloor (1+g)F_n \rfloor = F_n + \lfloor gF_n \rfloor = F_n + F_{n+1} = F_{n+2}$$

we have a simple formula for F_{n+2} as well, namely $F_{n+2} = \lfloor g^2F_n \rfloor$.

Hence we have the following surprising fact: With the possible exception of some small indices the evenindexed and the oddindexed members of any Fibonacci sequence with $gF_0 < F_1$ belong to the Beatty sequences with $\alpha = g$ and $\beta = g^2$, respectively. (For $gF_0 > F_1$ simply exchange α and β.) No wonder, we finally arrived at these Beatty sequences in Wythoff's game!

Let's turn to another topic now, which is less theoretical. A few weeks ago I bought a new Pentium 4 PC with 2GHz and 256 MB RAM and I was looking for a nice Derive-program that could take advantage of these new features. Finally I made up my mind to test some huge Mersenne numbers for primality. There is already a program in the Derive-library to test a Mersenne number $2^p - 1$, where p is supposed to be an odd prime, using the well-known Lucas Lehmer Test:

```
LUCAS_LEHMER(p, m_) :=
  Prog
    m_ := 2^p - 1
    SOLVE(ITERATE(MOD(s_^2 - 2, m_), s_, 4, p - 2) = 0)
```

As you can see this test is amazingly simple: All you have to do is to set s:=4 and perform the assignment

$$s := s^2 - 2 \mod (2^p - 1)$$

exactly (p-2) times. If s=0, then $2^p - 1$ is a prime, otherwise not. If p isn't too large, say p<10000, the present program will do. For larger p it certainly pays off to choose an approach that is a little more sophisticated.

The first thing one could do is to use the identity

$$A2^p + B = A(2^p - 1) + A + B \equiv A + B \mod 2^p - 1$$

in order to make the reduction mod $2^p - 1$ less timeconsuming. In fact, it has become a matter of "shifting and adding", which can be carried out very fast on a binary computer. Here is a program that is still very simple, but already uses this fact.

```
LLTEST(p, k_, m_, s_ := 4) :=
  Prog
    m_ := 2^p - 1
    k_ := p - 2
    Loop
      s_ := s_^2 - 2
      s_ := MOD(SHIFT(s_, -p) + (s_ ∧ m_), m_)
      k_ :- 1
      If k_ = 0
         RETURN SOLVE(s_ = 0)
```

Note that I used the undocumented SHIFT(n,c) here. SHIFT(n,c) returns the result of shifting n left by c bits. If c is negative (as in our program above!) then n is shifted right –c bits. This program is already considerably faster than LUCAS_LEHMER(p) if p is in the range of several thousands as you may conclude from the table below.

Here is yet another try, where I used representations of the form

$$a2^{(p+1)/2} + b \quad (0 \le a < 2^{(p-1)/2},\ 0 \le b < 2^{(p+1)/2})$$

for the numbers $< 2^p - 1$. Furthermore, I used the identity

$$(a2^{(p+1)/2} + b)^2 = a^2 2^{p+1} + 2ab2^{(p+1)/2} + b^2 \equiv 2ab2^{(p+1)/2} + 2a^2 + b^2 \bmod (2^p - 1)$$

and set

$$c = d2^{(p-1)/2} + e$$

in order to simplify the last expression even further to

$$e2^{(p+1)/2} + 2a^2 + b^2 + d$$

```
lltest(p, a_ := 0, b_ := 4, c_, q_, r_, u_, v_) :=
  Prog
    q_ := (p + 1)/2
    r_ := q_ - 1
    u_ := 2^q_ - 1
    v_ := (u_ + 1)/2 - 1
    p :- 2
    Loop
      c_ := 2·a_·b_
      b_ := 2·a_^2 + b_^2 + SHIFT(c_, -r_) - 2
      a_ := (c_ ∧ v_) + SHIFT(b_, -q_)
      b_ := (b_ ∧ u_) + SHIFT(a_, -r_)
      a_ := a_ ∧ v_
      p :- 1
      If p = 0
         RETURN SOLVE(a_ = v_ ∧ b_ = u_)
```

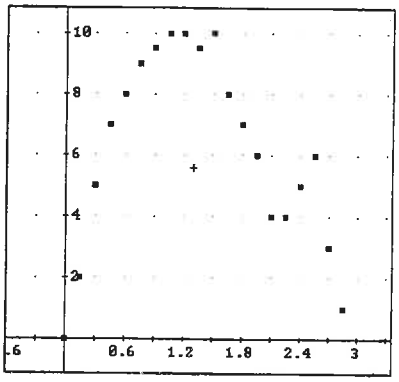
Here are some timings for various prime exponents on my 2GHz-machine:

p	LUCAS_LEHMER(p)	LLTEST(p)	lltest(p)
4423	3.21s	1.32s	1.40s
9689	23.1s	11.2s	9.29s
9941	24.9s	12.1s	9.96s
11213	35.5s	17.0s	13.8s
19937	195.3s	90.2s	70.9s

The largest number I tested was p=132049 using lltest(p) (the corresponding Mersenne prime $2^p - 1$ has already 39751 digits!) and it took 5h 16min. Don't laugh though, the story is not yet finished! Next time, I'll tell you about my experiences when using the Toom-Cook algorithm and FFT in order to speed up the squaring in the Lucas-Lehmer-Test! (j.wiesenbauer@tuwien.ac.at)

One of our DUG-members sent a mail reporting that his son in law is working as an anesthesist and he investigates the flow velocity during breathing out. He and his colleagues want to evaluate the data applying a Forier Analysis. Herwig remembering his knowledge on Numerical Mathematics wrote a little *DERIVE*-procedure – and here is it. It was not difficult to transfer the function on the TI.

```
a := [0, 2, 5, 7, 8, 9, 9.5, 10, 10, 9.5, 10, 8, 7, 6, 4, 4, 5, 6, 3, 1]
b := VECTOR(0.15·k, k, 0, 19)
b := [0, 0.15, 0.3, 0.45, 0.6, 0.75, 0.9, 1.05, 1.2, 1.35, 1.5, 1.65, 1.8, 1.95, 2.1,
  2.25, 2.4, 2.55, 2.7, 2.85]
```



Herwig refers to Bernhard Paule's *Die Mathematik des Naturforschers und Ingenieurs" Band II.* I can recommend *Numerische Methoden*, Faires & Burden, Spektrum (which is a translation of *Numerical Methods*, published by PWS 1993).

$$\mathrm{FPOINTS}(a, t, t1, t2, n) := \frac{1}{\mathrm{DIMENSION}(a)} \cdot \sum_{s=1}^{\mathrm{DIMENSION}(a)} a_s + \sum_{k=1}^{n} \frac{2}{\mathrm{DIMENSION}(a)} \cdot \left(\sum_{s=1}^{\mathrm{DIMENSION}(a)} a_s \cdot \mathrm{COS}\left(\frac{k \cdot 2 \cdot \pi \cdot (s-1)}{\mathrm{DIMENSION}(a)}\right)\right) \cdot \mathrm{COS}\left(\frac{2 \cdot \pi \cdot k \cdot t}{t2 - t1}\right) + \sum_{k=1}^{n} \frac{2}{\mathrm{DIMENSION}(a)} \cdot \left(\sum_{s=1}^{\mathrm{DIMENSION}(a)} a_s \cdot \mathrm{SIN}\left(\frac{k \cdot 2 \cdot \pi \cdot (s-1)}{\mathrm{DIMENSION}(a)}\right)\right) \cdot \mathrm{SIN}\left(\frac{2 \cdot \pi \cdot k \cdot t}{t2 - t1}\right)$$

```
FPOINTS(a, t, 0, 3, 10)
```