THE DERIVE - NEWSLETTER #65

ISSN 1990-7079

THE BULLETIN OF THE

DERIVE®

USER GROUP

+ CAS-TI

Contents:

March 2007

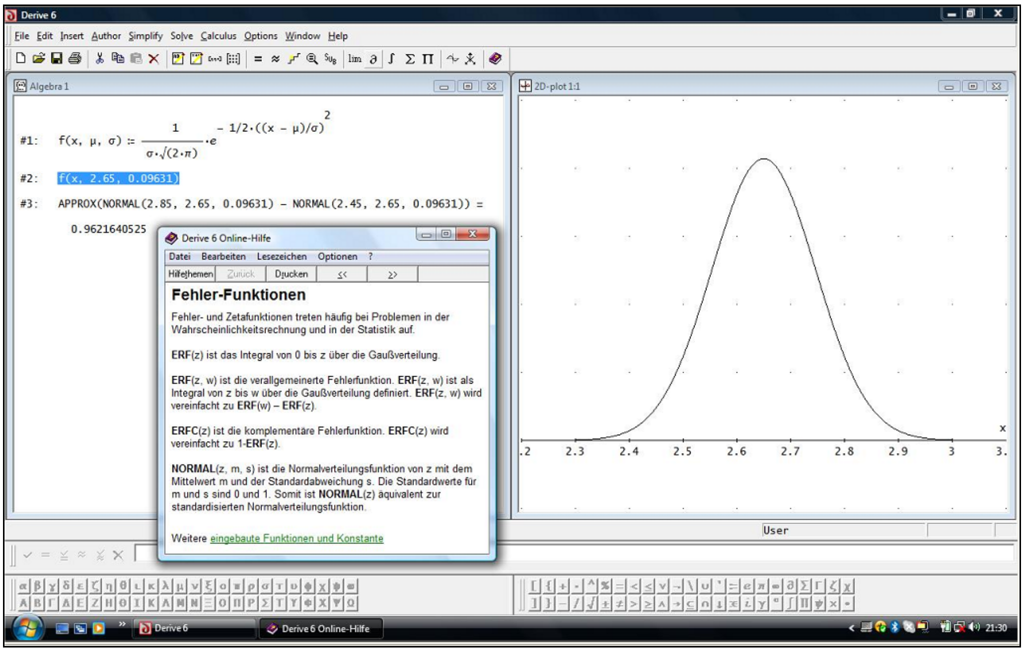
DERIVE & Windows VISTA

There were some rumors that DERIVE would not be supported by Windows Vista. We could give a VISTA-screenshot with DERIVE earlier.

Then I received the message that DERIVE is running fine but the Online-Help cannot be opened because VISTA does not support the old HTML-format.

Only two days later our DUG-member, DERIVE & IT-specialist Günter Schödl told me that there is a Microsoft patch to overcome this incompatibility. He immediately downloaded the patch an installed it on his laptop computer. You can find the URL for the patch below.

Heinz Hinkelmann asked for the URL and he sent the VISTA screenshot with DERIVE and the Online-Help open. Thanks to Günter and Heinz.



http://www.microsoft.com/downloads/details.aspx?displaylang=de&FamilyID=6ebcfad9-d3f5-4365-8070-334cd175d4bb

```
X64     for 64 Bit platforms
X86     for 32 Bit plattforms
```

`Just launch the file in Vista. Much luck, Günter.`

Congratulations

I have the pleasure to inform you that our vice president Bärbel Barzel made a great jump in her career. Bärbel – a DUG-member since its first days – made one of her first steps in using CAS in math education in Krems ("Taylor Series Expansions") and has ever been a enthusiastic propagator of CAS. (Do you remember the "Nonhigh-flyers", Bärbel?) She is now appointed professor at the Institute of Mathematics and Informatics and its Didactics at the Pedagogical Highschool in Freiburg. We are sure that Baerbel will continue her work with all her energy and enthusiasm. We wish Baerbel and her family the best for the future and are looking forward to meeting Professor Barzel at any occasion. Much Luck! Josef and the DUG-family.

Liebe DUG-Mitglieder,

endlich kommt der DNL (fast) zeitgerecht zu Ende des 1. Quartals zu Ihnen. Wir haben diesmal wieder ein sehr umfangreiches User Forum. Ich finde es besonders bemerkenswert, dass ein Problem aus dem alten DNL#13 über dessen revidierte Neufassung so großes Interesse gefunden hat. Inzwischen sind weitere Kommentare zu „R. Schorns Problem aus DNL#13“ eingelangt.

Mit MuPAD können wir einen neuen CAS-Gast begrüßen. Thomas Himmelbauer arbeitet schon seit längerer Zeit mit diesem Programm in seinem M-Unterricht. Er schickte uns eine Unterrichtseinheit zum Heronschen Verfahren zum Wurzelziehen. Ich konnte es natürlich nicht lassen, seine schöne Animation mit DERIVE und ITERATES nachzuvollziehen. Der rekursive Ansatz eignet sich auch ausgezeichnet sowohl für den Home Screen als Einstieg und für den Sequence Modus als weiteren Ausbau auf den TI-Rechnern. Leider lässt sich hier die graphische Darstellung nicht so leicht – oder nur über das Programmieren – herstellen. Die Animation mit DERIVE ist mir gelungen, weil wir von Peter Schofield ein schönes Werkzeug zur Erzeugung von „Slideshows“ erhalten haben.

Das ist aus einem Wunsch einer Kollegin entstanden, die gerne die schrittweise Approximation eines Kreises und damit eine Annäherung an π über eingeschriebene regelmäßige Polygone visualisieren wollte. Der slider bar läst sich im Zusammenhang mit dem VECTOR-Befehl oft nicht einsetzen, aber Peter hat diese Schwierigkeit mit einem einfachen Trick überwunden. Sehen Sie selbst, was man daraus machen kann.

Ich habe in diesem Zusammenhang Riemannsummen animiert. Im nächsten DNL wird Wolfgang Pröpper zeigen, wie dieses Thema mit TI-NSpire bearbeitet werden kann.

Ich möchte Sie noch besonders auf die Grafiken von Pierre Charland hinweisen. Mir fällt die Auswahl aus seinem überreichen Angebot immer sehr schwer. Eine große deutsche Zeitung (FAZ?) widmete vor einiger Zeit eine Doppelseite spannenden impliziten 3D-Flächen. Ich werde diese Flächen in den nächsten DNLs mit verschiedenen Programmen erzeugen.

Mit großer Freude habe ich wieder einen Artikel von Don Phillips aufgenommen, der unsere reiche Sammlung von wertvollen statistischen Werkzeugen erweitert. Zum Schluss bedanke ich mich gerne bei Johann Wiesenbauer, der – verlässlich wie immer –neue Titbits vorstellt, die sich wieder mit den Sudokus beschäftigen. Allen Autoren herzlichen Dank!

Dear DUG Members,

I am very glad that I can deliver DNL#65 – nearly – in time. As you will notice there is a very extended User Forum. Especially remarkable is the fact that there was and still is so much interest in “R. Schorn´s problem from DNL#13” after putting the revised DNL#13 on the website.

We welcome MuPAD as our special CAS-guest. Thomas Himmelbauer has used this CAS for a while in teaching and he sent one of his numerous teaching units: Heron´s algorithm for finding a square root. I couldn´t resist to reproduce his animation with DERIVE. This was only possible because Peter Schofield provided a *Slide Show Tool*. The recursive procedure can be demonstrated on the TI-calculators, too: as an introduction on the Home Screen and then using the Sequence Mode. Graphic representation of the process by an animation is not so easy to perform and requires some programming.

I was asked by a colleague how to apply the slider bar to visualise the approximation of a circle and subsequently of π by a sequence of inscribed regular polygons. I knew that slider bars have problems applying on VECTOR-generated lists. But Peter´s trick overcomes this difficulty. See, what can be achieved now using his versatile tool. As further examples I added the circumscribed regular polygons together with a visualisation of the approximations of π and from another field of mathematics, the Riemann sums. Wolfgang Pröpper will show how to animate Riemann sums with TI-NSpire in the next DNL.

I´d like to call your attention to Pierrre Charland´s graphics. It was not easy for me to make a choice of his rich supply of great pictures. Speaking about graphs, a well-known German newspaper (FAZ?) dedicated a double page to exciting implicit defined 3D surfaces. I intend to present these surfaces in the next DNLs generated by various programs.

It is a pleasure for me to have another contribution from Don Phillips, which extends our collection of valuable DERIVE statistics tools. We close this DNL again with Johann Wiesenbauer´s Titbits. He – reliable as ever – treated once more the Sudokus. Thank you all for your wonderful cooperation!

Download all *DNL-DERIVE*- and TI-files from

http://www.austromath.at/dug/

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

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

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

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

Next issue: June 2007
Deadline 15 May 2007

Preview: Contributions waiting to be published

Some simulations of Random Experiments, J. Böhm, AUT, Lorenz Kopp, GER
Wonderful World of Pedal Curves, J. Böhm
Another Task for End Examination, J. Lechner, AUT
Tools for 3D-Problems, P. Lüke-Rosendahl, GER
ANOVA with *DERIVE* & *TI*, M. R. Phillips, USA
Financial Mathematics 4, M. R. Phillips
Hill-Encription, J. Böhm
Farey Sequences on the *TI,* M. Lesmes-Acosta, COL
Simulating a Graphing Calculator in *DERIVE*, J. Böhm
Henon & Co, J. Böhm
Are all Bodies falling equally fast, J. Lechner
Do you know this? Cabri & CAS on PC and Handheld, W. Wegscheider, AUT
An Interesting Problem with a Triangle, Steiner Point, P. Lüke-Rosendahl, GER
Overcoming Branch & Bound by Simulation, J. Böhm, AUT
Mathematics and Design, H. Weller, GER
Diophantine Polynomials, D. E. McDougall, Canada
Challenger Matrix Problems, G P Speck, New Zealand
Graphics World, Currency Change, P. Charland, CAN
Precise Recurring Decimal Notation, P. Schofield, UK
Problems solved using the TI-Nspire, K. Stulens, BEL
Step function and Riemann Sums on the TI-NSpire, W. Pröpper, GER
Cubics, Quartics – interesting features, T. Koller & J. Böhm
Logos of Companies as an Inspiration for Math Teaching
Exciting Surfaces in the FAZ
Centroid of a Triangle – Recursively, D. Sjöstrand, SWE
BooleanPlots.mth, P. Schofield, UK

and Setif, FRA; Vermeylen, BEL; Leinbach, USA; Baumann, GER; Keunecke, GER, and others

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

Johann Wiesenbauer, Vienna Tue 06.02.2007 15:25

Hi folks,

Now that Josef Boehm is about to reissue the DNL #13, I had a look at R. Schorn's problem on page 3, namely to compute the maximal coefficient of the polynomial

$$(4x^3+3x^2+2x+1)^{20}.$$

Well, that was almost 13 years ago and it took Derive 387.2s then to find the answer 884231108759769374 5, which turns out to be the coefficient of x^40.

Just to take into account the advances of both Derive and computers since then, I would like to increase the exponent to say 100, and pose this as a new challenge. (As there are vacations at the universities right now, I thought, you might feel like a challenge!) In other words, what is the maximal coefficient in the expansion of

$$(4x^3+3x^2+2x+1)^{20}$$

and in which monomial does it occur?

I for my part also got my teeth into this nice problem and just in case you want to compare with my solution (in Derive 6.10), you will find it in the attachment.

Cheers,
Johann

OStD. R. Schorn, Kaufbeuren, Germany

In the "Journal of Recreational Mathematics" (Volume 25, Number 2 - 1993) I found the following problem:

2052. A Coefficient Problem by Charles Ashbacher, Cedar Rapids, Iowa.
On page 473 of the unsolved problems section of Index to *Mathematical Problems: 1989-1984* by Stanley Rabinowitz, there is the problem:

Menemui 5.2.1: Find the largest coefficient in the expansion of $(1 + 2x + 3x^2 + 4x^3)^{20}$.

This note started an interesting discussion between Johann and Valeriu Anisiu from Romania. I´ll skip Johann´s first version and give the last one at the end of the discussion.

The first reaction came from Spain:

Ignacio Larrosa Cañestro, A Coruña (España)

Hi Johann

I cannot open the attached file, but I spend 2.17 to develop the polynomial, 0.454 to aprox, and a few seconds more to visually inspect it.

Then the maximal coefficient is

398243667203410623024848236684603963137619260356309333266022065462985155939604195463780443664821745 ~= 3.982436672*10^98

and occurs in the term correspondent to x^200.

Best regards,
Ignacio Larrosa Cañestro, A Coruña (España)
ilarrosa@mundo-r.com

Johann Wiesenbauer, Vienna

Hi Ignacio,

Yes, your result is correct. It was meant to be a programming challenge though and getting the final result somehow doesn't matter here as much as the total time needed for this. Sorry, if my post had been misleading in that respect!

Then Valeriu joined the party:

Valeriu Anisiu, Romania Tue 06.02.2007 19:57

Hello Johann, hello everybody. My solution is slower but simpler:

```
#1:    maxterm1(u, n, x, c_, v_) :=
         Prog
           v_ := TERMS(EXPAND(u^n), x)
           c_ := SUBST(v_, x, 1)
           v_↓POSITION(MAX(c_), c_)
```

However, the result of

$$\#2: \quad \text{maxterm1}(4\cdot x^3 + 3\cdot x^2 + 2\cdot x + 1,\ 100)$$

namely

#3: 3982436672034106230248482366846039631376192603563093332660220654629851559396041954 6~
3780443664821745·x^{200}

is obtained in < 2 seconds.

Best regards,
Valeriu

Another polynomial specialist appeared:

Danny Ross Lunsford Tue 06.02.2007 18:23

$$\#1: \quad \text{PROG}\left(a := \text{EXPAND}((4\cdot x^3 + 3\cdot x^2 + 2\cdot x + 1)^{100}),\ b := \text{TERMS}(a),\ c := \text{VECTOR}\left(\frac{b_i}{x^{301 - i}},\ i,\ 1,\ 300\right),\ \text{DISPLAY}(\text{MAX}(c))\right)$$

This takes a little over 3 seconds on my ThinkPad A31p.
Answer

39824366720341062302484823668460396313761926035630933326602206546298515593960419546~
3780443664821745

101st term
-drl

Again Johann and then two replies from Valeriu:

Johann Wiesenbauer, Vienna Tue 06.02.2007 18:26

Hello Valeriu,

Great! Yes, as for brevity and elegance, your program is hard to surpass.

In particular, it makes excellent use of some built-in Derive features. Even though, I hate to say it, but it is considerably slower (by a factor of about 100 on my machine) than my program.

You know, I have been in the programming business for decades now, and nowadays I'm no longer the merciless performance hunter I used to be. If a program is so elegant and short like yours, I would certainly accept a lower performance to some degree. But here, as for me, this sacrifice of performance is too much though. (Just a hint: Try to increase the exponent n in this example up to say n=1000 and compare!)

Cheers,

Johann

Valeriu Anisiu, Tue 06.02.2007 22:16

Hello Johann,

I was aware from the beginning that your program is much faster.

But your program does not use Derive as a CAS; it manipulates lists instead of polynomials (except for a derivative, but this can be also avoided).

In an ideal CAS (an world) the manipulation of polynomials should be very efficient. My program is waiting for that moment -:).

Best reards,

Valeriu

Valeriu Anisiu Wed 07.02.2007 00:58

Hello Johann,

Another thing,

Your program seems to give wrong answers and becomes much slower than mine when the degree of p is >> 4.

Try maxterm(x^100 + 4x^3 + 3x^2 + 2x + 1, 5) or more dramatically

maxterm(x^1000 + 4x^3 + 3x^2 + 2x + 1, 5), versus maxterm1(...).

Cheers,

Valeriu

Johann ended the discussion and his file summarizes the process of finding the solution:

Johann Wiesenbauer, Vienna Tue 06.02.2007 23:27

Hello Valeriu,

Yes, you have a point there. If the degree of the base polynomial is very large, a direct application of my method fails. I don't think that this case is very typical though.

Cheers,

Johann

Johann Wiesenbauer, Vienna Wed 07.02.2007 11:36

Hello Valeriu,

Based on your remarks, I made a revision of my program (cf. the attachment), which looks much more streamlined now. Even though, your objections remains valid. In particular, as to maxterm() I have to add certain requirements regarding the base polynomial, otherwise it won't work correctly. (I'm sure, one could easily fix this problem, but I don't have the energy right now. One should remember that I wrote that routine to solve my original challenge efficiently, which it does, indeed!)

At any rate, many thanks for your valuable remarks!

Cheers, Johann

polypowers.dfw

Yet another way of computing polynomial powers

(by Johann Wiesenbauer, Vienna University of Technology, revised on February 7)

If u is any nonconstant polynomial in x, then the following routine computes the coefficient of x^k of the power y = u^n for any integer n > 1 by solving the differential equation u y' = n u' y for y in an iterative way.

```
       coeff(u, n, k, a_, b_, c_, i_ := 1, m_) :=
         Prog
           m_ := POLY_DEGREE(u)
           a_ := ADJOIN(SUBST(u, x, 0)^n, VECTOR(0, j_, m_ - 1))
           b_ := VECTOR(POLY_COEFF(u, x, k_), k_, 0, m_)
           c_ := VECTOR((j_·n + j_ - 1)·b_↓(j_ + 1), j_, 1, m_)
#1:        Loop
             If i_ > k
                RETURN FIRST(a_)
             a_ := ADJOIN(a_·c_/(i_·FIRST(b_)), DELETE(a_, -1))
             c_ :- REST(b_)
             i_ :+ 1

#2:    coeff(4·x^3 + 3·x^2 + 2·x + 1, 20, 40) = 8842311087597693745
```

In particular, we can use it in a slightly modified form to compute the monomial in u^n with the maximal coefficient under the assumption that all coefficients of u are nonnegative and nondecreasing w.r.t. their index (see below). This solves a problem posed by R. Schorn in DNL #13, p. 3 in a very efficient way from an algorithmic point of view. (Note the computation time of 0.000s on my PC!)

```
       maxterm(u, n, a_, b_, c_, i_ := 1, m_) :=
         Prog
           m_ := POLY_DEGREE(u)
           a_ := ADJOIN(SUBST(u, x, 0)^n, VECTOR(0, j_, m_ - 1))
           b_ := VECTOR(POLY_COEFF(u, x, k_), k_, 0, m_)
           c_ := VECTOR((j_·n + j_ - 1)·b_↓(j_ + 1), j_, 1, m_)
#3:        Loop
             m_ := a_·c_/(i_·FIRST(b_))
             If m_ ≤ FIRST(a_)
                RETURN FIRST(a_)·x^(i_ - 1)
             a_ := ADJOIN(m_, DELETE(a_, -1))
             c_ :- REST(b_)
             i_ :+ 1

#4:    maxterm(4·x^3 + 3·x^2 + 2·x + 1, 20) = 8842311087597693745·x^40
```

The actual superiority of the algorithm above only shows though, if n is much larger, say 100, as in my challenge on the Derive-forum.

$$\#5: \quad \text{maxterm}(4\cdot x^3 + 3\cdot x^2 + 2\cdot x + 1,\ 100)$$

$$\#6: \quad 398243667203410623024848236684603963137619260356309333266022065462985155939604195\sim$$
$$63780443664821745\cdot x^{200}$$

As for this ongoing competition on the Derive-forum, the best solution so far was contributed by Valeriu Anisiu. It is very short and elegant and looks like this:

```
       maxterm1(u, n, x, c_, v_) :=
         Prog
#7:        v_ := TERMS(EXPAND(u^n), x)
           c_ := SUBST(v_, x, 1)
           v_↓POSITION(MAX(c_), c_)
```

Alas, as for the performance of the problem above, it is not a serious competitor being slower by a factor ~ 100.

$$\#8: \quad \text{maxterm1}(4\cdot x^3 + 3\cdot x^2 + 2\cdot x + 1,\ 100)$$

$$\#9: \quad 398243667203410623024848236684603963137619260356309333266022065462985155939604195\sim$$
$$63780443664821745\cdot x^{200}$$

On the other hand, in a subsequent post Valeriu wrote:

Another thing,
Your program seems to give wrong answers and becomes much slower than
mine when the degree of p is >> 4.
Try maxterm(x^100 + 4x^3 + 3x^2 + 2x + 1, 5)
or more dramatically maxterm(x^1000 + 4x^3 + 3x^2 + 2x + 1, 5), versus maxterm1(...).

Yes, as I already wrote in my answer, he has a point there. (Compare the run times below!)

$$\#10: \quad \text{maxterm}(x^{100} + 4\cdot x^3 + 3\cdot x^2 + 2\cdot x + 1,\ 5) = 17203\cdot x^{10}$$

$$\#11: \quad \text{maxterm}(x^{1000} + 4\cdot x^3 + 3\cdot x^2 + 2\cdot x + 1,\ 5) = 17203\cdot x^{10}$$

$$\#12: \quad \text{maxterm1}(x^{100} + 4\cdot x^3 + 3\cdot x^2 + 2\cdot x + 1,\ 5) = 17203\cdot x^{10}$$

$$\#13: \quad \text{maxterm1}(x^{1000} + 4\cdot x^3 + 3\cdot x^2 + 2\cdot x + 1,\ 5) = 17203\cdot x^{10}$$

For bigger exponents the results become even wrong, because the requirement that the coefficients of the base polynomial are nondecreasing w.r.t. the index is not fulfilled here.

$$\#14:\quad \text{maxterm}(x^{1000} + 4\cdot x^3 + 3\cdot x^2 + 2\cdot x + 1,\ 10) = 1239350265\cdot x^{20}$$

$$\#15:\quad \text{maxterm1}(x^{1000} + 4\cdot x^3 + 3\cdot x^2 + 2\cdot x + 1,\ 10) = 1303773150\cdot x^{1018}$$

$$\#16:\quad \text{coeff}(x^{1000} + 4\cdot x^3 + 3\cdot x^2 + 2\cdot x + 1,\ 10,\ 1018) = 1303773150$$

Hence, when applying my routines, one should be aware of those restrictions. Many thanks to Valeriu for pointing out them!

The problem was announced as unsolved in the 1980´s. I don´t believe that it was the intention to have a computer solution. I wonder if there is a way to find this number by reasoning applying or developing something like a "super binomial formula"? Josef

The Klein Bottle and its Normals

Lester Anderson

Hello

I have a set of parametric equations defining a 3D surface, in this example a "Klein bottle", as a non-orientable surface. DFW file attached.

```
k:=4(1-cos(u)/2)

For u=0-pi

x=6cos(u)(1+sin(u))+kcos(u)cos(v)
y=16sin(u)(1+sin(u))+ksin(u)cos(u)
z=ksin(v)

For u=pi-2pi

x=6cos(u)(1+sin(u))+kcos(u)cos(v)
y=16sin(u)
z=ksin(v)

v= -pi to pi
```

What I would like to do is create a demonstration plot that shows normals to the surface (say at the mesh nodes for z=0, panels=20), to illustrate the nature of the "one-sided" surface i.e. no volume in this case. I am not too sure how to proceed on this other than setting a plane to z=0 as a starting point.

Ideally I am looking to do a cutaway view. In the above case setting v=-pi to 0 for illustration.

This is quite a complicated modelling task as not only do I need to determine the normals to the surface at a given z value, but also to make the normal vectors a small enough length to be instructive. Hopefully someone can provide some guidance on this issue as I would be looking to apply the technique to other surfaces.

The DFW file is in version 5 format, but works fine in version 6. Any advice about how to proceed will be welcome.

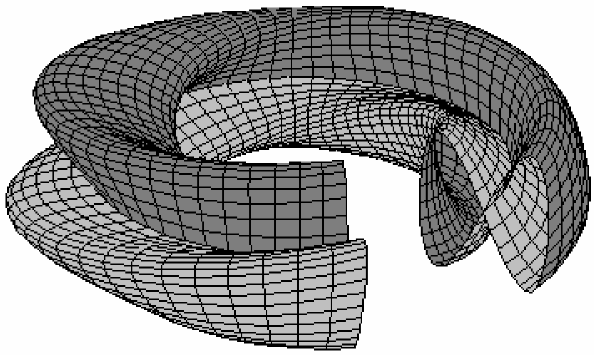
Thanks
Lester

Dear Lester,
here is bottle#2 + normals.
Best regards
Josef

The plot is embedded – You can reanimate the Bottle, Cheers!

#1: $$\left[\left(a + \cos\left(\frac{u}{2}\right)\cdot\sin(v) - \sin\left(\frac{u}{2}\right)\cdot\sin(2\cdot v)\right)\cdot\cos(u),\ \left(a + \cos\left(\frac{u}{2}\right)\cdot\sin(v) - \sin\left(\frac{u}{2}\right)\cdot\sin(2\cdot v)\right)\cdot\sin(u),\ \sin\left(\frac{u}{2}\right)\cdot\sin(v) + \cos\left(\frac{u}{2}\right)\cdot\sin(2\cdot v)\right]$$

0 <= u <= 7PI/4; 0 <= v <= 2PI; a = 3

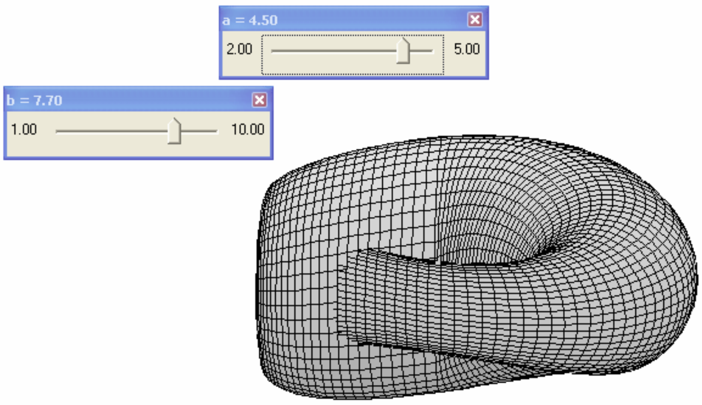


working with slider bars for a and b:

#2: $$r := 4\cdot\left(1 - \frac{\cos(u)}{2}\right)$$

#3: $$[a\cdot\cos(u)\cdot(1 + \sin(u)) + r\cdot\cos(u)\cdot\cos(v),\ b\cdot\sin(u) + r\cdot\sin(u)\cdot\cos(v),\ r\cdot\sin(v)]$$

#4: $$[a\cdot\cos(u)\cdot(1 + \sin(u)) + r\cdot\cos(v + \pi),\ b\cdot\sin(u),\ r\cdot\sin(v)]$$



Lester Anderson´s definition:

#5: $$[6\cdot\cos(u)\cdot(1 + \sin(u)) + r\cdot\cos(u)\cdot\cos(v),\ 16\cdot\sin(u) + r\cdot\sin(u)\cdot\cos(v),\ r\cdot\sin(v)]$$

#6: $$[6\cdot\cos(u)\cdot(1 + \sin(u)) + r\cdot\cos(u)\cdot\cos(v),\ 16\cdot\sin(u),\ r\cdot\sin(v)]$$

Parameter curve u = PI/4

#7: $\left[\cos(u)\cdot(6\cdot\sin(u) + 4\cdot\cos(v) + 6) - 2\cdot\cos(u)^2\cdot\cos(v),\ \sin(u)\cdot(4\cdot\cos(v) + 16) - 2\cdot\sin(u)\cdot\cos(u)\cdot\cos(v),\ 4\cdot\sin(v) - 2\cdot\cos(u)\cdot\sin(v)\right]$

#8: $\left[\cos(u)\cdot(6\cdot\sin(u) + 4\cdot\cos(v) + 6) - 2\cdot\cos(u)^2\cdot\cos(v),\ 16\cdot\sin(u),\ 4\cdot\sin(v) - 2\cdot\cos(u)\cdot\sin(v)\right]$

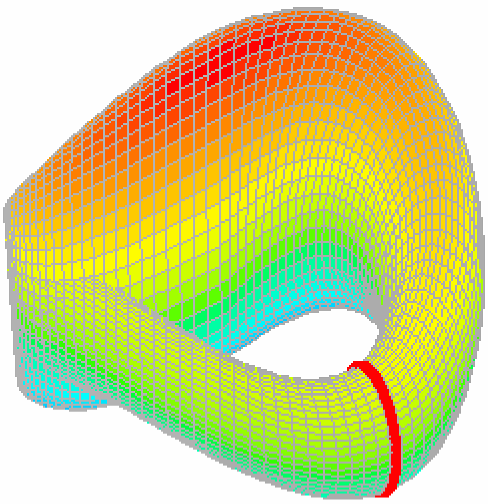
#9: $\left[\cos\left(\frac{\pi}{4}\right)\cdot\left(6\cdot\sin\left(\frac{\pi}{4}\right) + 4\cdot\cos(v) + 6\right) - 2\cdot\cos\left(\frac{\pi}{4}\right)^2\cdot\cos(v),\ \sin\left(\frac{\pi}{4}\right)\cdot(4\cdot\cos(v) + 16) - 2\cdot\sin\left(\frac{\pi}{4}\right)\cdot\cos\left(\frac{\pi}{4}\right)\cdot\cos(v),\ 4\cdot\sin(v) - 2\cdot\cos\left(\frac{\pi}{4}\right)\cdot\sin(v)\right]$

#10: $[(2\cdot\sqrt{2} - 1)\cdot\cos(v) + 3\cdot\sqrt{2} + 3,\ (2\cdot\sqrt{2} - 1)\cdot\cos(v) + 8\cdot\sqrt{2},\ (4 - \sqrt{2})\cdot\sin(v)]$

#11: par_c := $[(2\cdot\sqrt{2} - 1)\cdot\cos(v) + 3\cdot\sqrt{2} + 3,\ (2\cdot\sqrt{2} - 1)\cdot\cos(v) + 8\cdot\sqrt{2},\ (4 - \sqrt{2})\cdot\sin(v)]$

to produce a thick line:

#12: $\text{VECTOR}\left([\text{par\_c}],\ v,\ -\pi,\ \pi,\ \frac{\pi}{100}\right)$



finding the normal vector with respect to the surface

#13: $\left(\frac{d}{du}[6\cdot\cos(u)\cdot(1 + \sin(u)) + r\cdot\cos(u)\cdot\cos(v),\ 16\cdot\sin(u) + r\cdot\sin(u)\cdot\cos(v),\ r\cdot\sin(v)]\right) \times \frac{d}{dv}[6\cdot\cos(u)\cdot(1 + \sin(u)) + r\cdot\cos(u)\cdot\cos(v),\ 16\cdot\sin(u) + r\cdot\sin(u)\cdot\cos(v),\ r\cdot\sin(v)]$

#14: $\left[8\cdot\cos(u)^3\cdot\cos(v)^2 - \cos(u)^2\cdot(16\cdot\cos(v)^2 + 32\cdot\cos(v)) - \cos(u)\cdot(4\cdot\sin(u)^2\cdot\sin(v)^2 - 12\cdot\cos(v)^2 - 64\cdot\cos(v)) + 8\cdot\sin(u)^2,\ 24\cdot\cos(u)^3\cdot\cos(v) + \right.$

...

Set again u = PI/4

$$\#15:\quad \left[8\cdot\cos\left(\frac{\pi}{4}\right)^3\cdot\cos(v)^2 - \cos\left(\frac{\pi}{4}\right)^2\cdot(16\cdot\cos(v)^2 + 32\cdot\cos(v)) - \cos\left(\frac{\pi}{4}\right)\cdot\left(4\cdot\sin\left(\frac{\pi}{4}\right)^2\cdot\sin(v)^2 - 12\cdot\cos(v)^2 - 64\cdot\cos(v)\right) + 8\cdot\sin\left(\frac{\pi}{4}\right)^2, \ldots\right.$$

$$\#16:\quad \left[(9\cdot\sqrt{2} - 8)\cdot\cos(v)^2 + (32\cdot\sqrt{2} - 16)\cdot\cos(v) - \sqrt{2} + 4,\ (9\cdot\sqrt{2} - 8)\cdot\cos(v)^2 + (12\cdot\sqrt{2} - 6)\cdot\cos(v) + \sqrt{2} - 4,\ (18 - 8\cdot\sqrt{2})\cdot\sin(v)\cdot\cos(v) + (44 - 11\cdot\sqrt{2})\cdot\sin(v)\right]$$

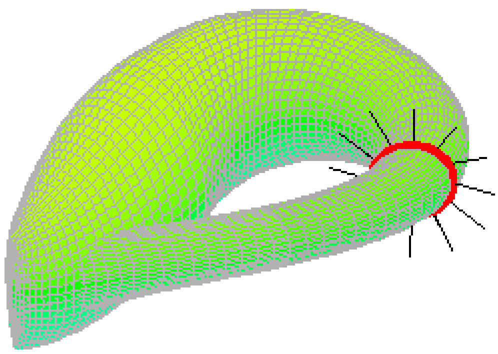
nv ist the normal unit vector of the surface for the points of the parameter curve from above:

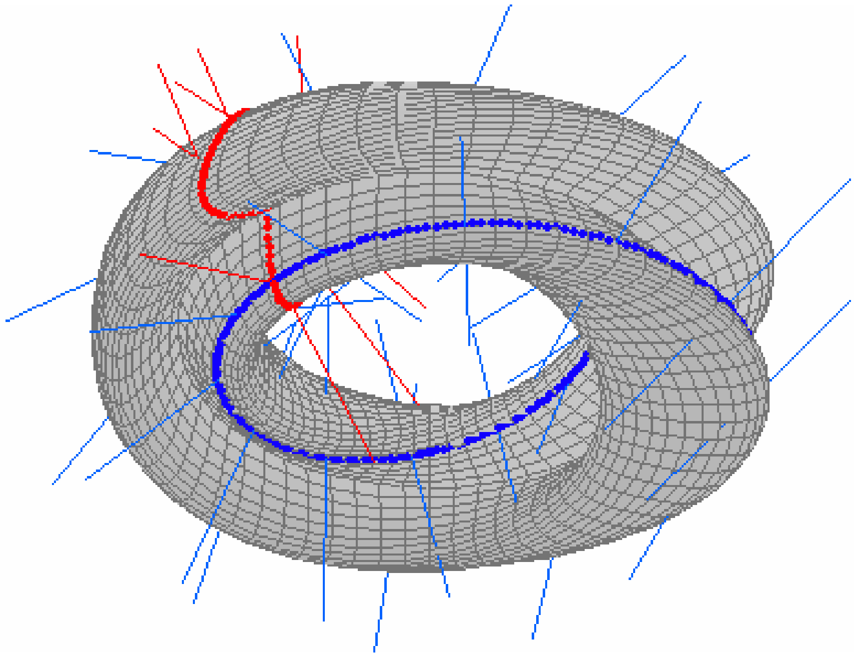
$$\#17:\quad nv := \frac{\left[(9\cdot\sqrt{2} - 8)\cdot\cos(v)^2 + (32\cdot\sqrt{2} - 16)\cdot\cos(v) - \sqrt{2} + 4,\ (9\cdot\sqrt{2} - 8)\cdot\cos(v)^2 + (12\cdot\sqrt{\sim}\right.}{\left|\left[(9\cdot\sqrt{2} - 8)\cdot\cos(v)^2 + (32\cdot\sqrt{2} - 16)\cdot\cos(v) - \sqrt{2} + 4,\ (9\cdot\sqrt{2} - 8)\cdot\cos(v)^2 + (12\cdot\sqrt{\sim}\right.\right.}$$

$$\#18:\quad nv := \left[-\frac{(4\cdot\sqrt{2} - 9)\cdot\cos(v)^2 + 8\cdot(\sqrt{2} - 4)\cdot\cos(v) - 2\cdot\sqrt{2} + 1}{\sqrt{((451 - 244\cdot\sqrt{2})\cdot\cos(v)^2 + 4\cdot(222 - 115\cdot\sqrt{2})\cdot\cos(v) - 492\cdot\sqrt{2} + 1107)}},\ -\frac{(4\cdot\sqrt{2} - 9)\cdot\cos(v)^2 + 3\cdot(\sqrt{2} - 4)\cdot\cos(v) + 2\cdot\sqrt{2} - 1}{\sqrt{((451 - 244\cdot\sqrt{2})\cdot\cos(v)^2 + 4\cdot(222 - 115\cdot\sqrt{2})\cdot\cos(v) - 492\cdot\sqrt{2} + 1107)}},\ \frac{\sin(v)\cdot(11\cdot(2\cdot\sqrt{2} - 1) - (8 - 9\cdot\sqrt{2})\cdot\cos(v))}{\sqrt{((451 - 244\cdot\sqrt{2})\cdot\cos(v)^2 + 4\cdot(222 - 115\cdot\sqrt{2})\cdot\cos(v) - 492\cdot\sqrt{2} + 1107)}}\right]$$

This creates a family of normal lines setting t from 0 to 3 after issuing Insert > Plot in the 3D PLot window

$$\#19:\quad \text{VECTOR}\left(par\_c + t\cdot nv,\ v,\ -\pi,\ \pi,\ \frac{\pi}{6}\right)$$





For the second calculation presenting another form of the Kleinbottle together with two parameter curves and their normals resulting in the graph above example follow the file kleinbottle.dfw. Josef

Lester Anderson

Dear Josef

Thanks for the help on the problem, it looks really good. It's good to have a starting point like this for topological studies; shows just what can be achieved with Derive!

Peter Schofield

Hello Joseph,

I have just managed to download and have a good look at DNL#64, and I must thank you for your really nice presentation and selection of graphics examples using my “ImplicitPts” and “ImplicitDots” Derive files (DNL#64 pp. 33-40).
I note that you have found some more interesting examples– I particularly liked the “Window of Viviani” formed by the intersection of a sphere and cylinder. I also found Tania Kollar’s article in DNL#63 (pp. 6-14) very fascinating.

In 3D, using Derive Boolean expressions, it is also possible to generate “clouds” of points and dots using the attached short file: BooleanPlots.mth. Applications are limited by the size of the list of points (dots), but the file illustrates some of the possibilities. I’m sure that, given the opportunity, Derive would have developed further graphic facilities in this direction.

Many Thanks!
Peter

BooleanPlots.mth will be presented in one of the next DNLs, Josef

Plot Range for Parametric Plots

Walter Wegscheider, Austria

Dear Josef,
this is a DERIVE related question: is it possible to influence the range of the parameter for parametric plots by a command.

Example: vertical line for $x = 2$: `g(t):=[2,t]`

If I want to have $0 \le t \le 8$ it is no problem to indicate the range in the 2D-plot dialogue. Unfortunately there is no command like `g(t):=[2,CHI(0,t,8)*t]` or similar. Is there an alterantive via the status variables or …

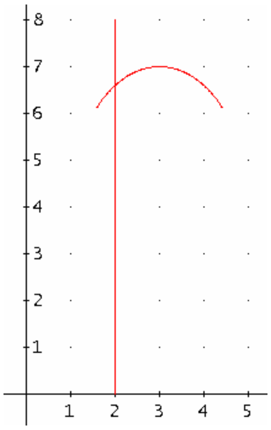
Dear Walter,

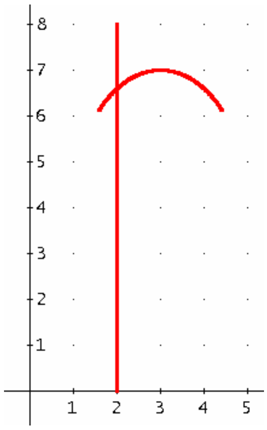
Basically there is no way to pass the dialogue box. I offer a little trick (#17) which works properly. Please inspect the attached file.

#1: (TABLE([2, t], t, 0, 8, 0.01))↓↓[2, 3]

#2: $\left(\text{TABLE}\left([2\cdot\text{COS}(t) + 3,\ 3\cdot\text{SIN}(t) + 4],\ t,\ \frac{\pi}{4},\ \frac{3\cdot\pi}{4},\ 0.01\right)\right)$↓↓[2, 3]

Applying Point Size Medium or Large you can get thick graphs.





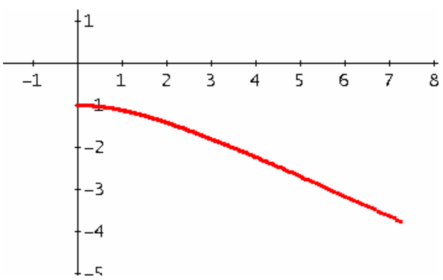
Let´s make of it an “all purpose parametric plot function”:

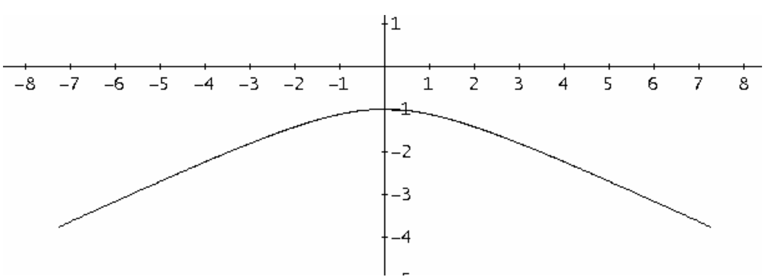
#3: paramf(pf, p_, start, end, step := 0.01) := (TABLE($[pf_1,\ pf_2]$, p_, start, end, step))↓↓[2, 3]

#4: paramf([2·SINH(u), – COSH(u)], u, 0, 2)

#5: paramf([2·SINH(u), – COSH(u)], u, –2, 2, 0.005)

If necessary one has to change the step size. In the “standard plot window” step = 0.01 is sufficiently small. Don´t forget to activate Approximate before plotting in the Plot window.





Finally I come back to the initial plots. #6 gives the plot of the vertical segment together with the segment of the ellipse.

$$\#6: \quad \left[\text{paramf}([2,\ t],\ t,\ 0,\ 8),\ \text{paramf}\left([2\cdot\text{COS}(s) + 3,\ 3\cdot\text{SIN}(s) + 4],\ s,\ \frac{\pi}{4},\ \frac{3\cdot\pi}{4}\right)\right]$$

Walter Wegscheider

Dear Josef, ,
many thanks – your "trick 17" means: program your personal plot command ☺.

The internal plot procedure works similar (including an extra check of the plot range considering the pixels of the screen).

We have discussed this question several times and have the strong opinion, that both alternatives should be provided, which are in particular:

a) the dialogue principle with intelligent default settings (like in DERIVE) which enables a first "beautiful" result without special syntax knowledge.

b) the command principle (eg MuPAD) – each command must be entered with all possible parameters to obtain a reasonable result. This requires knowledge of the syntax but has the advantage that all settings can be saved in a document and can also be used in programs (and macros) because all details can be addressed directly.

Thanks again and best regards,

Walter

Annotations!

James Gordon

I have Derive 6.
My expression #4 is sqrt(24). When I highlight #4 and execute the Command Toolbar's Approximate button, I get in the Status Bar's second pane "Approx(user)" rather than "Approx(#4). My settings are factory default. How can I get the Status Bar's second pane to read "Approx(#n), when n represents any item number.
Thank you.

Albert D. Rich

Hello James,
In order to get "Approx(#4)" use the Approximate button usually near the top of the Derive algebra window instead of the Approximate button to the left of the expression entry line.
Hope this helps.
Aloha,
Albert D. Rich, Co-author of Derive

Calculus Made Easy

CME v.9.0 is available for all TI-CAS-Calculators

http://www.ti89.com

Our friend Nils Hahnfeld released a new version with many new and exciting features. More about CME v.9.0 in the next DNL.

Francisco Marcelo Fernández

Dear Derivers:

I am calculating *Hankel determinants* with the simple function

dete(n, d) := DET(VECTOR(VECTOR(APPEND("c", i + j + d + 1), j, 0, n - 1), i, 0, n - 1))

where n is the determinant dimension, and d shifts the elements subscript. I obtain all the determinants with $n = 2,...,7$ easily and quickly, with (for example) $d = 0$. But the case `dete(8,0)` takes too long and consumes too much memory. Does anybody know why it happens?. I find exactly the same problem with Derive versions 5 and 6.

Thanks,

Marcelo

#1: dete(n, d) := DET(VECTOR(VECTOR(APPEND(c, i + j + d + 1), j, 0, n − 1), i, 0, n − 1))

#2: $\text{dete}(3, 0) = c1\cdot(c3\cdot c5 - c4^2) - c2^2\cdot c5 + 2\cdot c2\cdot c3\cdot c4 - c3^3$

#3: $\text{dete}(4, 1) = c2\cdot(c4\cdot(c6\cdot c8 - c7^2) - c5^2\cdot c8 + 2\cdot c5\cdot c6\cdot c7 - c6^3) + c3^2\cdot(c7^2 - c6\cdot c8) + c3\cdot(2\cdot c4\cdot(c5\cdot c8 - c6\cdot c7) - 2\cdot c5\cdot(c5\cdot c7 - c6^2)) - c4^3\cdot c8 + c4^2\cdot(2\cdot c5\cdot c7 + c6^2) - 3\cdot c4\cdot c5^2\cdot c6 + c5^4$

A Normal Distribution Standard Problem

Josef Lechner

Sehr geehrter Herr Dr. Lechner,
mit großem Interesse habe ich Ihren Artikel "Standardisierung der Normalverteilung - ein Anachronismus?!" in den TI-Nachrichten1/03 gelesen und stimme Ihren Urteilen zu. Bei meinem Unterricht zur Normalverteilung habe ich mich durch ihren Artikel anregen lassen und zwar auch die Gaußsche Dichtefunktion und ihre Beziehung zu anderen Dichtefunktionen normalverteilter Zufallsgrößen behandelt, für Berechnungen wollte ich aber die Dichtefunktion benutzen, so wie Sie sie auch in Ihrem Artikel angeben. Die Berechnungen waren zunächst problemlos möglich bis wir auf folgende ganz normal und harmlos aussehende Aufgabe stießen:

Eine Maschine stellt Nägel her. Die Länge der Nägel ist normalverteilt mit dem Erwartungswert μ = 8,00 cm und der Standardabweichung σ = 0,15 cm. Bei wie viel Prozent der Nägel weicht die Länge höchstens um e = 0,20 cm vom Erwartungswert μ ab?

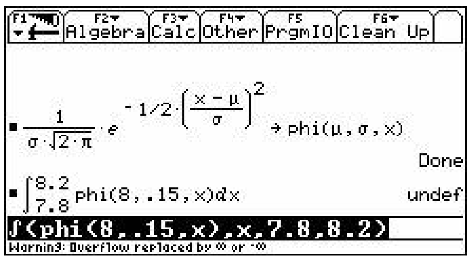
Josef Lechner forwarded the following mail which he received from Dr. Weiß, a German Voyage User. Dr. Weiß refers to Josef´s paper about "Standardisation of the Normal Distribution – An Anachronism?!". Everything was ok, but one very common problem was a "problem" for the Voyage 200.

A machine produces nails. The legth of the nails follows a normal distribution with mean μ = 8,00 cm and standard deviation σ = 0,15 cm. What percentage of the nails lies between μ ± 0,20 cm?

As you can see, the calculator doesn´t deliver any result and shows a system message

Overflow replaced by ∞ or -∞.

There were nor problems when working with the same data in mm or taking other values!



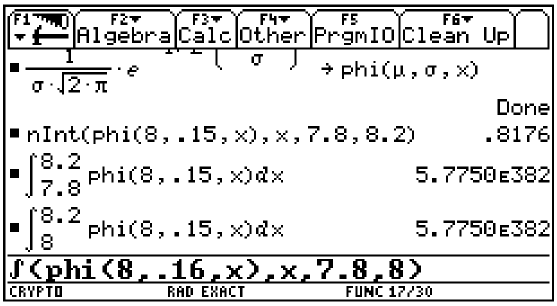
Sie sehen: Der Ti-92 meldete: Overflow replaced by unendlich or -unendlich und zeigte kein Ergebnis an.
Der Rechner lieferte problemlos das richtige Ergebnis, wenn man die Zentimeterangaben aus der Aufgabe in mm umwandelte und ihn dann rechnen ließ. Ich habe keine Erklärung für dieses Phänomen gefunden. Ist Ihnen im Unterricht mal Ähnliches passiert und/oder können Sie eventuell den Effekt erklären? Über eine Antwort würde ich mich sehr freuen.

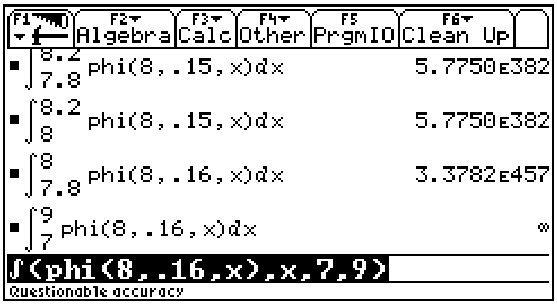
DNL:

This is very interesting. Obviously the TI does not like to accept this special value for the standard deviation (don´t ask me why). Dr Weiß (and we, of course) can overcome this bug by applying numerical integration – using **`nInt`**.

See the following TI-screens.

Interestingly enough I am lucky to produce the nice 457 digit number and infinite, but without an error message – so the result should be correct – or not? – ☺. (It´s only "questionable accuracy").





Solids of Revolutions and Infinite Series

Nils Hahnfeld, Virgin Islands

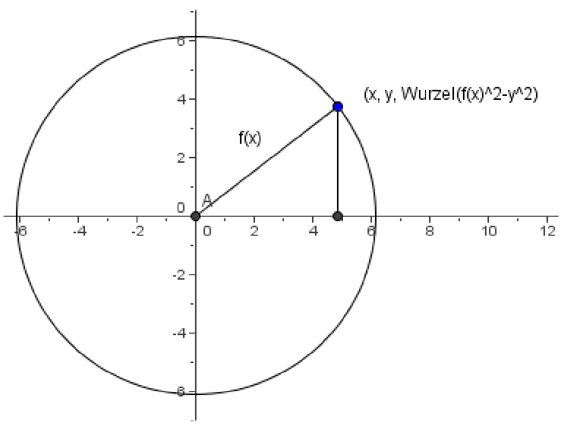
(1) Short question: how can I show my students on the TI-89 a 3D-plot which results in a rotation of sqrt(x) between [0, 4] around the x-axis?

DNL: There were several questions about representing solids of revolution with DERIVE and with the TI-devices in the past. (See DNL#35). It is no problem to produce very pretty and informative plots applying the parameter form and then it doesn´t matter which rotation axis applies. This is difficult on the TIs because we can only plot functions of form $z = f(x,y)$. So we have to distinguish between the top half and the bottom half of the surface and we can plot only one at a time.

So let´s try to do our best:

We rotate $y = f(x)$ around the x-axis.

The sketch explains the procedure: This is an intersection perpendicular to the rotation axis (= x-axis) at position x. The intersection is a circle with radius f(x). Point $(x, y, \sqrt{(f(x)^2-y^2)})$ lies on the top half of the surface and point $(x, y, -\sqrt{(f(x)^2-y^2)})$ on the bottom half.



Take the TI-89 (or the TI-92 or the Voyage 200 … or DERIVE):

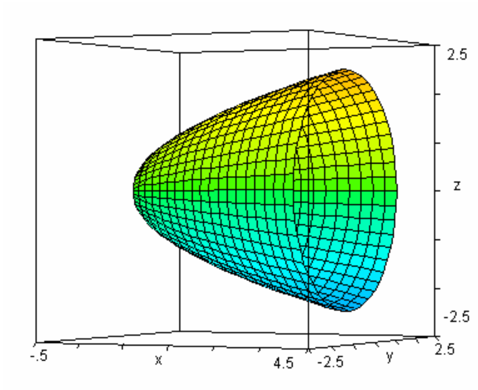
First of all I produce the paraboloid by parameter form with DERIVE:

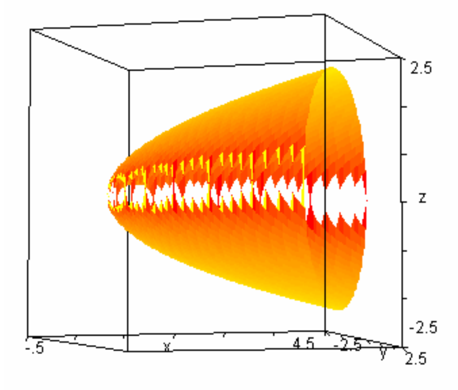
left: parameter representation

#1: [t, √t·COS(s), √t·SIN(s)]

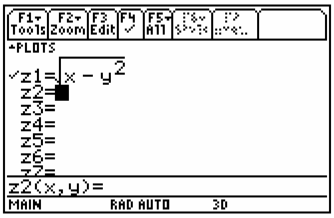
right: Two-functions-representation

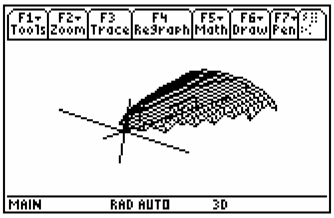
#2: $\left[\text{IF}(x \le 4, \sqrt{(x - y^2)}), \text{IF}(x \le 4, -\sqrt{(x - y^2)})\right]$

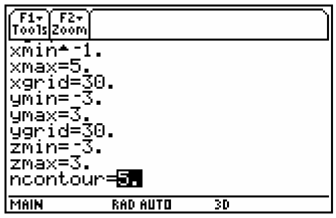




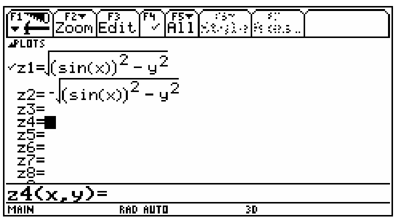
The surface on the TI-89 screen:

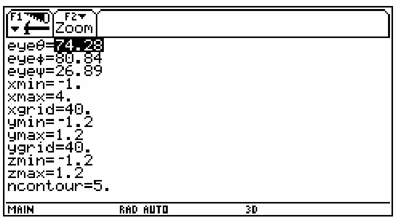


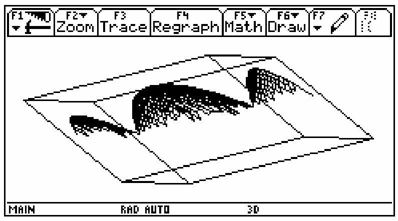




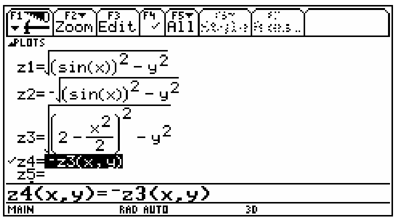
Two More examples (V 200): Rotation of the sine wave

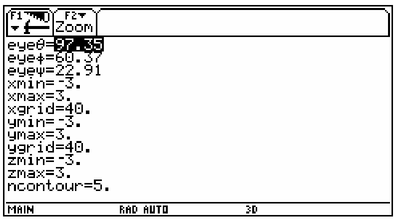


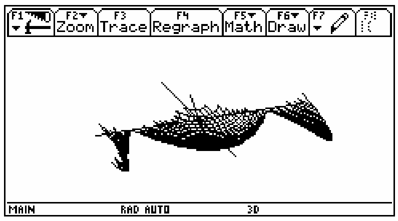




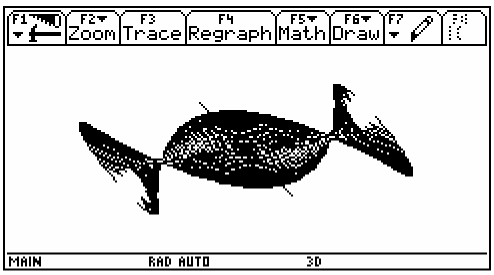
z2 shows the other half of the surface. Finally the rotation of f(*x*) = 2 – *x*^2/2 for -3 ≤ *x* ≤ 3







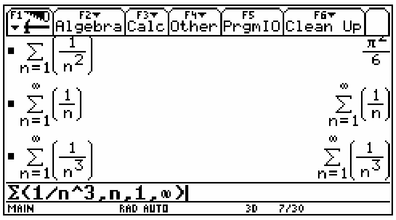
Save the figure as a picture, leave the window settings as they are and superimpose the graph of z4.

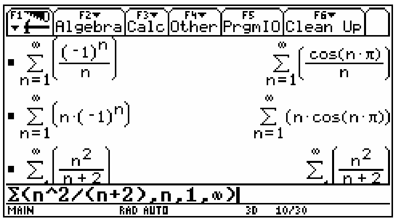


This is a fine trick to show the obtain the whole solid of revolution.

(2) The other question might be more difficult:

Can we teach the TI that the harmonic series is divergent, or does it know this?





I´d like to evaluate the end points of intervals of convergence and I need the harmonic / alternating harmonic series.

DNL: See first some sums of series calculated by the “Big Brother” DERIVE. Then I try to teach the TI some rules:

#1: $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$

#2: $\sum_{n=1}^{\infty} \frac{1}{n} = \infty$

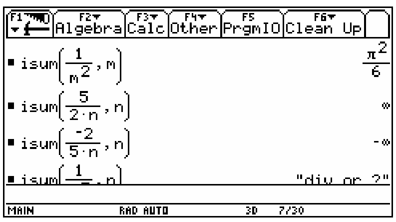
#3: $\sum_{n=1}^{\infty} \frac{1}{n^3} = \zeta(3)$

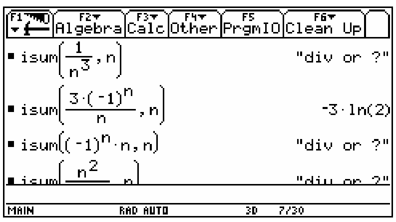
#4: $\sum_{n=1}^{\infty} \frac{(-1)^n}{n} = -\mathrm{LN}(2)$

#5: $\sum_{n=1}^{\infty} (-1)^n \cdot n = \sum_{n=1}^{\infty} n \cdot \mathrm{COS}(\pi \cdot n)$

#6: $\sum_{n=1}^{\infty} \frac{n^2}{n+2} = \infty$

```
isum(u,v)
Func
Local u_,s_,d_,k_,d2,res
u/(1/v)→u_:sign(u_)→s_
d(abs(u_),v)→d2
when(d2=0,0,1,1)→d2
string((⁻1)^v)→k_
string(s_)→d_
If (d_="1" or d_="⁻1") and d2=0 Then
   s_*∞→res: Goto end
EndIf
If d_=k_ and d2=0 Then
   u_*⁻ln(2)/(⁻1)^n→res:Goto end
EndIf
when(Σ(u,v,1,∞)=0,0,Σ(u,v,1,∞),"div or ?")→res
Lbl end
res
EndFunc
```





Heinrich Ludwig, Germany

Lieber Herr Böhm,

ich würde Sie gerne eine Kleinigkeit zu einem von DERIVEs internen Algorithmen fragen. Vielleicht wissen Sie es und es kostet Sie nicht viel Mühe, mir einen Hinweis zu geben.

Wie berechnet DERIVE die Wurzel einer ganzen Zahl? DERIVE muss einen sehr effizienten Algorithmus verwenden, um die quadratischen Faktoren einer ganzen Zahl zu erkennen.

Multipliziert man z.B. a := 2 * MERSENNE(12) * MERSENNE(12) zu einer 77-stellige Zahl aus und bildet anschließend SQRT(a), so hat sogar mein Uralt-Rechner nach 0,02 s das Ergebnis gefunden. Eine Primfaktorzerlegung von a ist mit 0,11 s auch erstaunlich schnell, aber eben nicht schnell genug. DERIVE muss andere Wege kennen.

Für einen Hinweis, auch auf weiterführende Literatur, wäre ich Ihnen sehr dankbar.

Mit freundlichen Grüßen

Heinrich Ludwig

Heinrich Ludwig asks which special algorithm is used by DERIVE for detecting quadratic factors of an integer. If you multiply a := 2 * MERSENNE(12) * MERSENNE(12) which results in a 77 digit number and then calculate SQRT(a) even Heinrich´s very old PC needs only 0.02 sec to find the result, but if you factorize a then it needs 0.11 sec, which is also very fast, but significantly slower. DERIVE must know other methods. The DNL-reader knows that Johann Wisenbauer is the specialist for questions like these and he answered:

Lieber Josef,
Das ganze fällt unter das Thema

detecting perfect powers.

Herr Ludwig sollte also diese Stichworte in google eingeben und das Material dazu sichten, insbesonderer auch die Arbeit

- Daniel J. Bernstein (1998). "Detecting perfect powers in essentially linear time". *Mathematics of Computation* **67** (223): 1253–1283.

Wenn man jetzt von algorithmischen Feinheiten absieht, steckt nicht viel mehr dahinter, als dass man für die in Frage kommenden n die nächstkleinere ganze Zahl zur n-ten Wurzel berechnet (meist mit Newton oder einer Modifikation davon) und dann schaut, ob die vorgelegte Zahl die n-te Potenz davon ist. Wenn also im Zuge des Faktorisierungsprozesses einmal eine perfekte Potenz erhalten wird, wie im obigen Beispiel nach dem Heraussieben aller Primfaktoren bis 1024, dann geht der Rest sehr schnell.
Herzliche Grüße,
Hannes

This problem is part of the objective

detecting perfect powers.

Mr Ludwig should use Google and enter these key-words. He will find some materials, in particular the paper

- Daniel J. Bernstein (1998). "Detecting perfect powers in essentially linear time". *Mathematics of Computation* **67** (223): 1253–1283.

Two-Stage Least Squares Linear Regression

by MacDonald R. Phillips

don.phillips@gmail.com

The most widely used single-equation method for estimating simultaneous systems of equations is two-stage least squares. Let **Y** be the endogenous or dependent variables in the system and **X** the exogenous or predetermined variables. The equations to be estimated are of the form:

$$\#5: \quad y = Y_1 \cdot \beta + X_1 \cdot \gamma + u$$

where **y** is an n by 1 vector of observations on the "dependent" variable, **Y**↓1 is an n by g matrix of observations on the other endogenous variables included in the equation, **β** is the g by 1 vector of coefficients associated with **Y**↓1, **X**↓1 is the n by k matrix of observations on the predetermined or instrumental variables appearing in the equation, **γ** is the k by 1 vector of coefficients associated with **X**↓1, and **u** is the n by 1 vector of disturbances in the equation.

The problem of applying OLS to the above equation is that the variables in **Y**↓1 are correlated with **u**. The essence of two-stage least-squares regression is the replacement of **Y**↓1 by a computed matrix **Y_hat**↓1, where hopefully the stochastic element is purged, and then of performing an OLS regression of **y** on **Y_hat**↓1 and **X**↓1.

The matrix **Y_hat**↓1 is computed in the first stage by regressing each variable in **Y**↓1 on <u>all</u> the instrumental variables in the complete model and replacing the actual observations on the Y variables by the corresponding regression values. Thus:

$$\#6: \quad Y\_hat_1 = X \cdot (X' \cdot X)^{-1} \cdot X' \cdot Y_1$$

where **X** = [**X**↓1 **X**↓2]. **X** is the n by K matrix of observations on <u>all</u> the instrumental variables in the complete model, **X**↓2 being the matrix of observations on those instrumental variables excluded from the equation under study.

In the second stage **y** is regressed on **Y_hat**↓1 and **X**↓1. The equation for the 2SLS estimates can be written as:

$$\#7: \quad \begin{bmatrix} Y_1' \cdot X \cdot (X' \cdot X)^{-1} \cdot X' \cdot Y_1 & Y_1' \cdot X_1 \\ X_1' \cdot Y_1 & X_1' \cdot X_1 \end{bmatrix} \cdot \begin{bmatrix} b \\ c \end{bmatrix} = \begin{bmatrix} Y_1' \cdot X \cdot (X' \cdot X)^{-1} \cdot X' \cdot y \\ X_1' \cdot y \end{bmatrix}$$

where **b** is the vector of coefficients on the other endogenous variables and **c** is the vector of coefficients on the predetermined variables in the equation, including the intercept.

The program TwoStage(eq_,endog_,exog_,data_,incept_:=1,vardef_:=1) computes the 2SLS regression coefficients, standard errors of the coefficients, t values and their probabilities, the ANOVA table, root MSE, R_square, and adjusted R_square.

The inputs are:

eq_: the equation to be solved for
endog_: a vector of the endogenous variables
exog_: a vector of the exogenous variables
data_: the name of the dataset where the first row of the data matrix contains the names of the variables
incept_: the default of 1 indicates that an intercept is to be computed for the equation; set incept:=0 if you do not want an intercept.
vardef_: the default of 1 sets the variance denominator to the degrees of freedom; changing vardef_ to 0 sets the denominator to the number of observations.

Note: According to SAS, if the no intercept option is set (incept_=0), the definition of the R_square statistic for two-stage least squares is changed to 1 - (Residal Sum of Squares/Uncorrected Total Sum of Squares).

In addition, the program Model(mm_,endog_,exog_,data_,incept_:=1,vardef_:=1) sovles for all the of equations at once. mm is a matrix of the equations with one equation per row.

Example 1:

The following data are for a simplified model designed to explain variations in the consumption and prices of food. The data are from Kmenta, pp. 563-65.

q = food consumption per head
p = ratio of food prices to general consumer prices
d = disposable income in constant prices
f = ratio of preceding year's prices received by farmers to general consumer prices
a = time in years

The endogenous variables are q and p. The exogenous variables are d, f, and a.

Estimate the following equations:

(1) $q = \gamma 0 + \beta 1 * p + \gamma 1 * d$ (the demand equation)
(2) $q = \gamma 0 + \beta 1 * p + \gamma 1 * f + \gamma 2 * a$ (the supply equation)

Enter the data and then use the Model() program:

$$\#9: \quad \text{Model}\left(\begin{bmatrix} q = p + d \\ q = p + f + a \end{bmatrix}, [q, p], [d, f, a], \text{data}\right)$$

#8: data :=

q	p	d	f	a
98.485	100.323	87.4	98	1
99.187	104.264	97.6	99.1	2
102.163	103.435	96.7	99.1	3
101.504	104.506	98.2	98.1	4
104.24	98.001	99.8	110.8	5
103.243	99.456	100.5	108.2	6
103.993	101.066	103.2	105.6	7
99.99	104.763	107.8	109.8	8
100.35	96.446	96.6	108.7	9
102.82	91.228	88.9	100.6	10
95.435	93.085	75.1	81	11
92.424	98.801	76.9	68.6	12
94.535	102.908	84.6	70.9	13
98.757	98.756	90.6	81.4	14
105.797	95.119	103.1	102.3	15
100.225	98.451	105.1	105	16
103.522	86.498	96.4	110.5	17
99.929	104.016	104.4	92.5	18
105.223	105.769	110.7	89.3	19
106.232	113.49	127.1	93	20

MODEL: q

Parameter	Value	Standard Error	t(17)	Prob(t)
β1	−0.2437076898	0.09607652762	2.536599686	0.02128843825
γ0	94.61485336	7.887363189	11.99575207	0
γ1	0.3143821013	0.04674526445	6.725432084	0.00000355719819

Source	DF	SS	MS	F	Prob(F)
Model	2	202.7676225	101.3838112	26.4446926	0.000006043944658
Residual	17	65.17469563	3.833805625		
Corrected Total	19	267.9423181			

Standard Error	R^2	Adj_R^2
1.958010629	0.7567584841	0.7281418352

$$q = -0.2437076898 \cdot p + 0.3143821013 \cdot d + 94.61485336$$

2nd part of the output of simplification of Model():

#10:

MODEL: q

Parameter	Value	Standard Error	t(16)	Prob(t)
β1	0.240567722	0.09950905774	2.417545974	0.02792589088
γ0	49.44820072	11.95947271	4.134647227	0.0007779536442
γ1	0.2528436233	0.09923148057	2.548018248	0.02148917651
γ2	0.2560236611	0.04704922378	5.441612859	0.00005432925863

Source	DF	SS	MS	F	Prob(F)
Model	3	172.1288502	57.37628342	9.581330838	0.0007369271092
Residual	16	95.81346791	5.988341744		
Corrected Total	19	267.9423181			

Standard Error	R^2	Adj_R^2
2.447108854	0.6424100957	0.5753619886

$$q = 0.240567722\cdot p + 0.2528436233\cdot a + 0.2560236611\cdot f + 49.44820072$$

Example 2:

The next example is based on Klein's model 1 (1950). The endogenous variables are c, p, w, i, x, wsum, k, and y. The exogenous variables are klag, plag, xlag, wp, g, t, yr.

yr = year - 1931
c = consumption
p = profits
w = private wage bill
i = investment
x = private production
wp = government wage bill
g = government demand
t = taxes
k = capital stock
wsum = total wage bill
plag = profits lagged
xlag = private product lagged
klag = capital stock lagged
y = c + i + g - t (national income)

Estimate the model: c = f(p,plag,wsum), i = f(p, plag, klag), and w = f(x, xlag, yr).

Full matrix contains 21 rows:

#11: klein :=

yr	c	p	w	i	x	wp	g	t	k	wsum	plag	xlag	klag	y
-10	41.9	12.4	25.5	-0.2	45.6	2.7	3.9	7.7	182.6	28.2	12.7	44.9	182.8	37.9
-9	45	16.9	29.3	1.9	50.1	2.9	3.2	3.9	184.5	32.2	12.4	45.6	182.6	46.2
-8	49.2	18.4	34.1	5.2	57.2	2.9	2.8	4.7	189.7	37	16.9	50.1	184.5	52.5
-7	50.6	19.4	33.9	3	57.1	3.1	3.5	3.8	192.7	37	18.4	57.2	189.7	53.3
-6	52.6	20.1	35.4	5.1	61	3.2	3.3	5.5	197.8	38.6	19.4	57.1	192.7	55.5
-5	55.1	19.6	37.4	5.6	64	3.3	3.3	7	203.4	40.7	20.1	61	197.8	57
-4	56.2	19.8	37.9	4.2	64.4	3.6	4	6.7	207.6	41.5	19.6	64	203.4	57.7
-3	57.3	21.1	39.2	3	64.5	3.7	4.2	4.2	210.6	42.9	19.8	64.4	207.6	60.3
-2	57.8	21.7	41.3	5.1	67	4	4.1	4	215.7	45.3	21.1	64.5	210.6	63
-1	55	15.6	37.9	1	61.2	4.2	5.2	7.7	216.7	42.1	21.7	67	215.7	53.5
0	50.9	11.4	34.5	-3.4	53.4	4.8	5.9	7.5	213.3	39.3	15.6	61.2	216.7	45.9

Apply model() again:

$$\#12:\quad \text{Model}\left(\begin{bmatrix} c = p + plag + wsum \\ i = p + plag + klag \\ w = x + xlag + yr \end{bmatrix}, [c, p, w, i, x, wsum, k, y], [klag, plag, xlag, wp, g, t, yr], klein\right)$$

I´ll show only the resulting model equations:

$$[c = 0.01730294013 \cdot p + 0.8101827436 \cdot wsum + 0.2162334038 \cdot plag + 16.55475199]$$

$$[i = 0.1502217351 \cdot p + 0.6159436535 \cdot plag - 0.1577876492 \cdot klag + 20.27821173]$$

$$[w = 0.4388586370 \cdot x + 0.1466742262 \cdot xlag + 0.1303957913 \cdot yr + 1.500299135]$$

Same data with intercept option = 0:

$$\#14:\quad \text{Model}\left(\begin{bmatrix} c = p + plag + wsum \\ i = p + plag + klag \\ w = x + xlag + yr \end{bmatrix}, [c, p, w, i, x, wsum, k, y], [klag, plag, xlag, wp, g, t, yr], klein, 0\right)$$

$$\#19:\quad [c = 0.1583375706 \cdot p + 1.121162681 \cdot wsum + 0.2651214661 \cdot plag]$$

$$\#20:\quad [i = 0.4459657099 \cdot p + 0.3685120348 \cdot plag - 0.06148091416 \cdot klag]$$

$$\#21:\quad [w = 0.4444584525 \cdot x + 0.166299164 \cdot xlag + 0.1129951702 \cdot yr]$$

References:

J. Johnson, *Econometric Methods*, 2nd ed., McGraw-Hill Book Co., New York, 1972.

Jan Kmenta, *Elements of Econometrics*, MacMillan Publishing Co., New York, 1971.

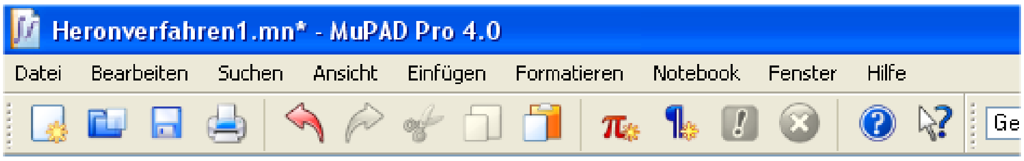
L. Klein, *Economic Fluctions in the United States: 1921-1940*, John Wileys and Sons, New York, 1950.

J.K. Binkley and Nelson, G. (1984), "Impact of Alternative Degrees of Freedom Corrections in Two and Three Stage Least Squares," *Journal of Econometrics*, 24, 3, 223-233.

Heron for Square Roots

Thomas Himmelbauer, Austria

Thomas is one of the few teachers in Austria working with MuPAD in classroom. I am very grateful that he sent some MuPad papers with one lesson among them (Schulübung = school exercise):



15.Schulübung

Heronsche Flächenformel:

Diese Formel dient zur nährungsweisen Berechnung von Quadratwurzeln.
Es handelt sich um ein rekursives Verfahren.
This formula is used for approximative calculation of square roots.
It is a recursive procedure. Demonstration of an example

Erklärung anhand eines Beispiels:

```
sqrt(51)
```

$\sqrt{51}$

1.Schritt:
Wir wählen ein Rechteck, dessen Fläche 51 groß ist.
Step 1: Choose a rectangle with area = 51.

```
a1:=1:b1:=51:a1*b1
```

51

2.Schritt:
Wir nehmen den Mittelwert der beiden Rechteckseiten als eine neue Rechteckseite,
die andere bestimmen wir so dass der Flächeninhalt jeweils 51 ist.
Step 2: We take the average of both sides as one new side and determine the second one in order to obtain again area = 51.

```
a2:=(a1+b1)/2:b2:=51/a2:a2*b2
```

51

3.Schritt:
Wir nehmen den Mittelwert der beiden Rechteckseiten als eine neue Rechteckseite,
die andere bestimmen wir so, dass der Flächeninhalt jeweils 51 ist.
Step 3: Repeat step 2.

```
a3:=(a2+b2)/2:b3:=51/a3:a3*b3
```

51

4.Schritt:
Wir nehmen den Mittelwert der beiden Rechteckseiten als eine neue Rechteckseite,
die andere bestimmen wir so, dass der Flächeninhalt jeweils 51 ist.
Step 4: Repeat step 2.

```
a4:=(a3+b3)/2:b4:=51/a4:a4*b4
```

51

```
float(b4^2)
```

33.47825896

5.Schritt:
Wiederhole Schritt 4.
Step 4: Repeat step 2.

```
a5:=(a4+b4)/2:b5:=51/a5:a5*b5
```
51

```
float(b5^2)
```
48.80600931

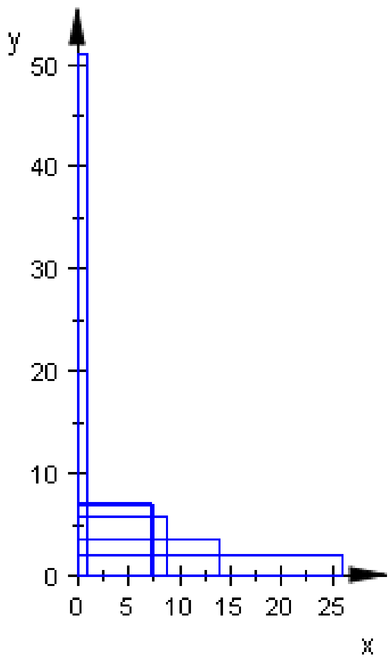
6.Schritt:
Wiederhole Schritt 5.
Step 6: Repeat step 2.

```
a6:=(a5+b5)/2:b6:=51/a6:a6*b6
```
51

```
float(a6^2)
```
51.02465678

Die Rechtecke nähern sich langsam einem Quadrat.
Die Rechteckseiten nähern sich langsam der zu berechnenden Wurzel.
The rectangles approach slowly a square and its sides approach the square root.

```
plot(plot::Rectangle(0..a1,0..b1),
plot::Rectangle(0..a2,0..b2),
plot::Rectangle(0..a3,0..b3),
plot::Rectangle(0..a4,0..b4),
plot::Rectangle(0..a5,0..b5),
plot::Rectangle(0..a6,0..b6)
,Scaling=Constrained)
```



Das Heronverfahren läßt sich durch folgendes Programm automatisieren!

The procedure can be automated by the following program:

```
Heron:=proc(A)
begin
M:=matrix(7,2);
M[1,1]:=A;
M[1,2]:=1;
for i from 2 to 7 do
M[i,1]:=float(1/2*(M[i-1,1]+M[i-1,2]));
M[i,2]:=A/float(1/2*(M[i-1,1]+M[i-1,2]));
end_for
end_proc
```

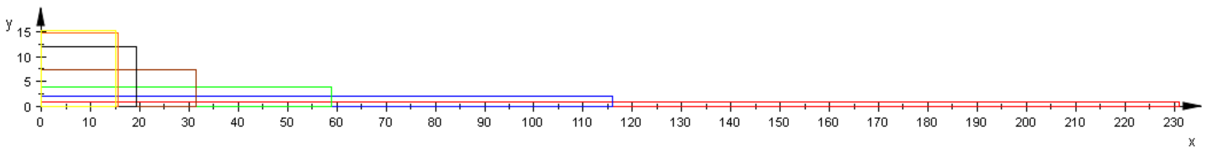
proc Heron(A) ... end

```
Heron(231)
```

15.19207849

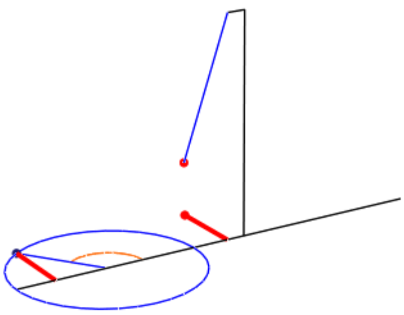
```
M
```

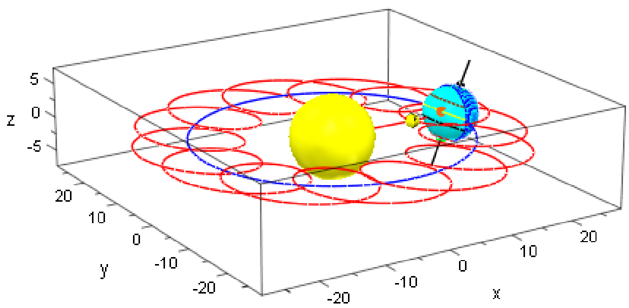
231	1
116.0	1.99137931
58.99568966	3.915540294
31.45561497	7.343680935
19.39964795	11.90743257
15.65354026	14.75704512
15.20529269	15.19207849



The following screen shots are from two MuPAD animations:

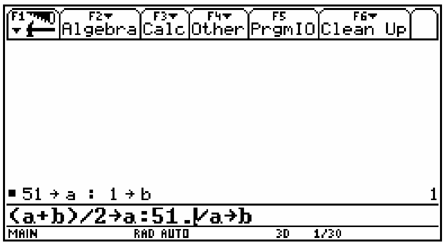
The harmonic motion and "Sun – Moon and Earth in Motion"

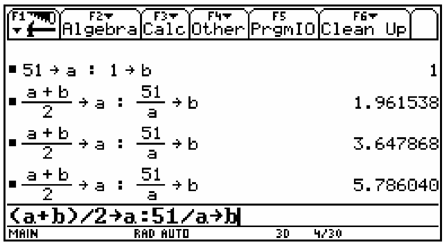


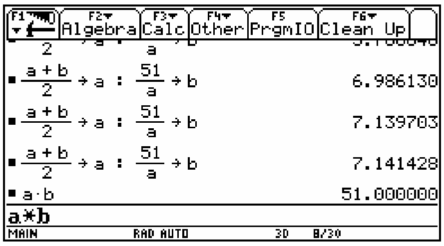


Comments on demonstrating Heron's Rule on the TI and with DERIVE.

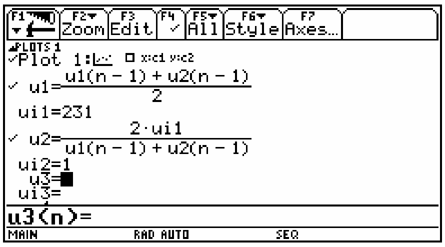
Performing the recursive procedure on the TI is very intuitive and informative as well in the Home Screen as applying the Sequence Mode.

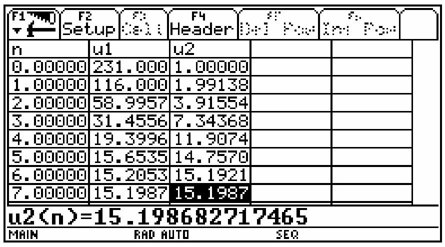






Not so easy, but of didactical value is defining the rule in the SEQUENCE Mode which unveils the recursive nature of the procedure in another way. We show how to approximate $\sqrt{231}$.





In Derive we can not only realise the recursive nature by applying the ITERATES-command, but also animate the approximating process using Peter Schofield´s *Slide Show tool*. (For details about the *Slide Show* see the following contribution.)

#1: $\text{heron}(z,\ n) := \text{ITERATES}\left(\left[\dfrac{v_1 + v_2}{2},\ \dfrac{2\cdot z}{v_1 + v_2}\right],\ v,\ [z,\ 1],\ n\right)$

#2: heron(51, 6)

#3:
$$\begin{bmatrix} 51 & 1 \\ 26 & 1.961538461 \\ 13.98076923 & 3.64786795 \\ 8.81431859 & 5.786040007 \\ 7.300179298 & 6.98612978 \\ 7.143154539 & 7.139702734 \\ 7.141428637 & 7.141428219 \end{bmatrix}$$

#4: $\sqrt{51}$

#5: 7.141428428

#6 creates the family of rectangles which converge to a square.

#6: $$\text{heron\_sq}(z, n) := \text{ITERATES}\left(\begin{bmatrix} 0 & 0 \\ \frac{v_{2,1} + v_{3,2}}{2} & 0 \\ \frac{v_{2,1} + v_{3,2}}{2} & \frac{2\cdot z}{v_{2,1} + v_{3,2}} \\ 0 & \frac{2\cdot z}{v_{2,1} + v_{3,2}} \end{bmatrix}, v, \begin{bmatrix} 0 & 0 \\ z & 0 \\ z & 1 \\ 0 & 1 \end{bmatrix}, n\right)$$

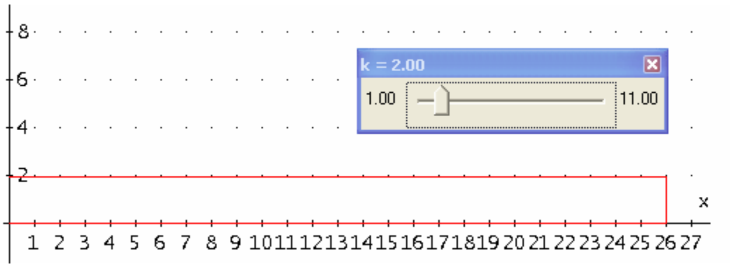
#7: $\text{Slide\_Show}(v, k) := \text{VECTOR}(v_i \cdot \text{IF}(k = i, 1, ?), i, 1, \text{DIM}(v))$

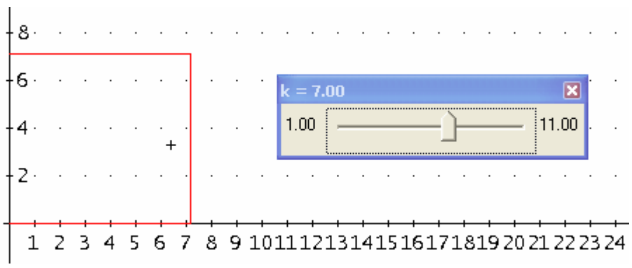
#8: $\text{Slides\_Show}(v, k) := \text{VECTOR}(v_i \cdot \text{IF}(k \ge i, 1, ?), i, 1, \text{DIM}(v))$

#9: root51 := heron_sq(51, 10)

#10: Slide_Show(root51, k)

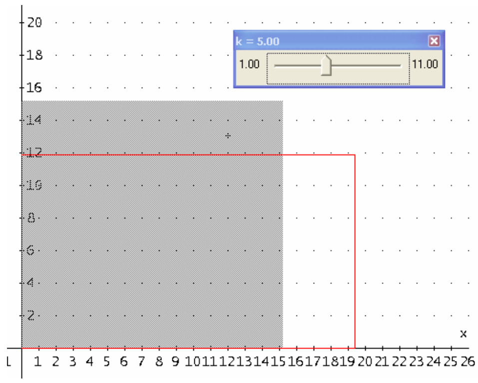
#6 creates the family of rectangles which converge to a square. Highlight #10, switch to the 2D-Plot Window, create a slider bar for *k* with $1 \le k \le 11$ with 10 intervalls. (Points small and connected)





Now you can stepwise follow how the initial rectangle changes into a square. (You could add the shaded square with sides $\sqrt{51}$ in order to make the approximating procedure more impressive.

(On the right picture we are on the way to $\sqrt{231}$.)



A Slide Show for the Slider Bar

Peter Schofield, UK

We had a seminar in Amstetten, Lower Austria, for evaluating several learning paths and discussing possible ways of integrating various technologies and media in math teaching and learning. During a break – breaks are a very important part of seminars for exchanging ideas and for receiving inspirations - one colleague asked for advice how to animate polygons inscribed in a circle to visualise approximation of π. She set up a VECTOR and could easily produce the polygons and plot one after the other.

$$\text{poly}(n) := \text{VECTOR}\left(3\cdot\left[\text{COS}\left(\frac{2\cdot\pi\cdot i}{n}\right),\ \text{SIN}\left(\frac{2\cdot\pi\cdot i}{n}\right)\right],\ i,\ 0,\ n\right)$$

```
polys(m) := VECTOR(poly(n), n, 3, m)
```

She wanted to install a slider bar for showing one polygon after the other. Each polygon appears as an element of the list polys(m). We both were not able to produce the performance of the polygons. I remembered a similar problem to show points of a list one after the other (DNL#62) but I couldn´t make David Sjöstrand´s trick working (and didn´t try Peter Schofield´s idea …). I knew that there are often problems occuring when applying the slider bar for a discrete variable. I wrote to Peter and decribed the problem. Peter came back to me after a couple of days – and here is his terrific slide show tool for two kinds of demonstrations. Josef

SlideShow.dfw: a Derive 6 file for 2D-plotting a slide-show of regular n-sided polygons leading to an approximation for a circle. March 2007.

Peter Schofield, Trinity & All Saints, LEEDS - email: p.schofield@leedstrinity.ac.uk

#1: $\text{Slide\_Show}(v, k) := \text{VECTOR}(v_i \cdot \text{IF}(k = i, ?), i, 1, \text{DIM}(v))$

#2: $\text{Slides\_Show}(v, k) := \text{VECTOR}(v_i \cdot \text{IF}(k \ge i, ?), i, 1, \text{DIM}(v))$

#3: $\text{poly}(n) := \text{VECTOR}\left(3\cdot\left[\text{COS}\left(\frac{2\cdot\pi\cdot i}{n}\right),\ \text{SIN}\left(\frac{2\cdot\pi\cdot i}{n}\right)\right],\ i,\ 0,\ n\right)$

#4: $\text{polys}(m) := \text{VECTOR}(\text{poly}(n), n, 3, m)$

#5: $$\text{polytri}(n) := \text{VECTOR}\left(3\cdot\begin{bmatrix} 0 & 0 \\ \text{COS}\left(\frac{2\cdot\pi\cdot i}{n}\right) & \text{SIN}\left(\frac{2\cdot\pi\cdot i}{n}\right) \\ \text{COS}\left(\frac{2\cdot\pi\cdot(i+1)}{n}\right) & \text{SIN}\left(\frac{2\cdot\pi\cdot(i+1)}{n}\right) \\ 0 & 0 \end{bmatrix},\ i,\ 0,\ n-1\right)$$

#6: $\text{polytris}(m) := \text{VECTOR}(\text{polytri}(n), n, 3, m)$

`Slide_Show` shows one polygon after the other and `Slides_Show` superimposes one graph after the other.

```
Select 2D-plot Window (Approx Before Plotting (ON)).
Insert>Slider-Bar> 3≤k≤36 with 33 intervals.
2D-plot and slide the following:
```

```
#7:   Slide_Show(polys(36), k - 2)

#8:   Slides_Show(polytris(36), k - 2)

#9:   Slides_Show(polys(36), k - 2)
```

If you don´t believe me then you are friendly invited to make a try – you can download all files – this works wonderful. (Don´t forget to set Approximate before Plotting on and to have the points connected, point size small is recommended.)

Peter closed his email by pointing out that this tool might find other applications, too. I took his words for serious and accomplished the inscribed polygon file with circumscribed polygons and a visualisation of the approximation process for π.

In- and circumscribed polygons with variable radius r

#10: $\text{poly\_in}(r, n) := \text{VECTOR}\left(r\cdot\left[\cos\left(\frac{2\cdot\pi\cdot i}{n}\right), \sin\left(\frac{2\cdot\pi\cdot i}{n}\right)\right], i, 0, n\right)$

#11: $\text{poly\_out}(r, n) := \text{VECTOR}\left(\frac{r}{\cos\left(\frac{\pi}{n}\right)}\cdot\left[\cos\left(\frac{2\cdot\pi\cdot i}{n}\right), \sin\left(\frac{2\cdot\pi\cdot i}{n}\right)\right], i, 0, n\right)$

#12: $x^2 + y^2 = 9$

```
#13:   poly_ins := VECTOR(poly_in(3, n), n, 3, 36)

#14:   Slide_Show(poly_ins, k - 2)

#15:   poly_outs := VECTOR(poly_out(3, n), n, 3, 36)

#16:   Slide_Show(poly_outs, k - 2)
```

#17 and #18 plot segments of height approximating π - converging from below and from above. It is recommended to plot #14 and #17 in the same colour and #16, #18 in another one.

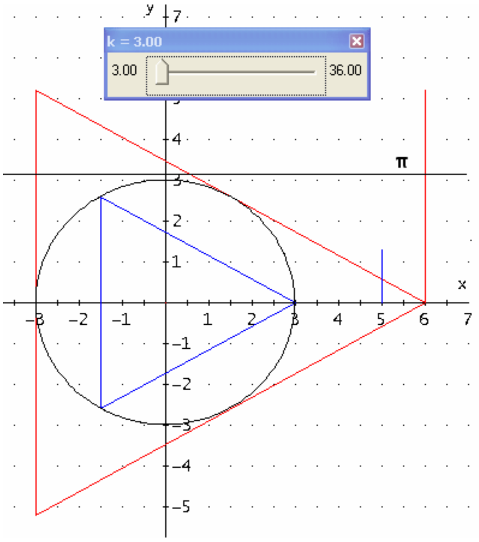
#19 gives a horizontal line y = π

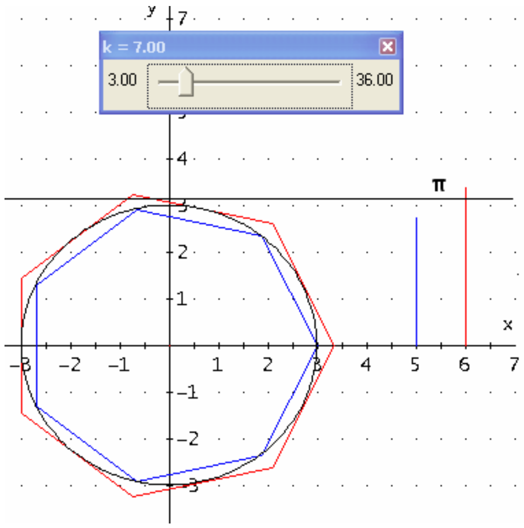
#17: $\text{pi\_in}(k) := \begin{bmatrix} 5 & 0 \\ 5 & \frac{k}{2}\cdot\sin\left(\frac{2\cdot\pi}{k}\right) \end{bmatrix}$

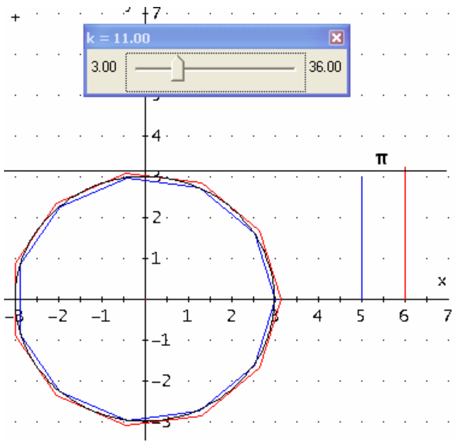
#18: $\text{pi\_out}(k) := \begin{bmatrix} 6 & 0 \\ 6 & \dfrac{\frac{k}{2}\cdot\sin\left(\frac{2\cdot\pi}{k}\right)}{\cos\left(\frac{\pi}{k}\right)^2} \end{bmatrix}$

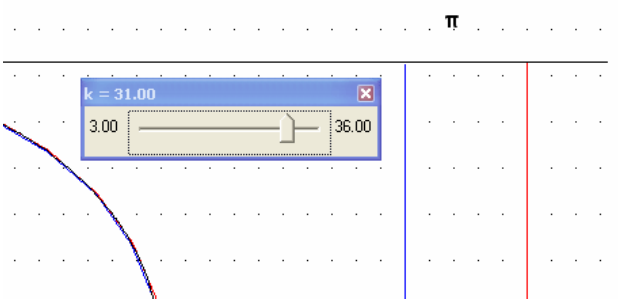
#19: π

Here are some screen shots:









When I read Peter´s note I immediately had the idea to use the slide show for demonstrating integration. More and more teaching software tools have animated Riemann sums implemented – we can now implement the animation by ourselves – thanks Peter´s slide show tool.

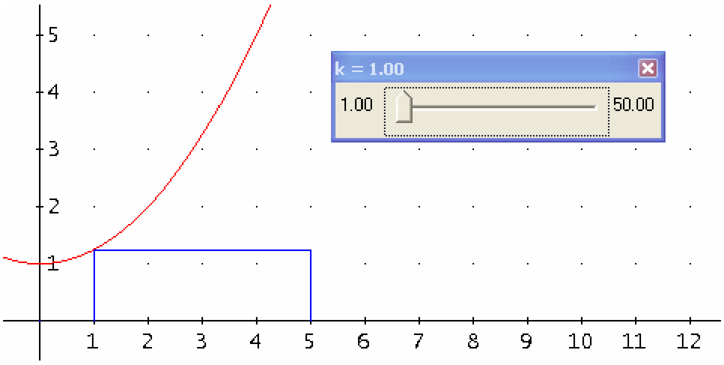
Another application:
Left and right Riemann sums

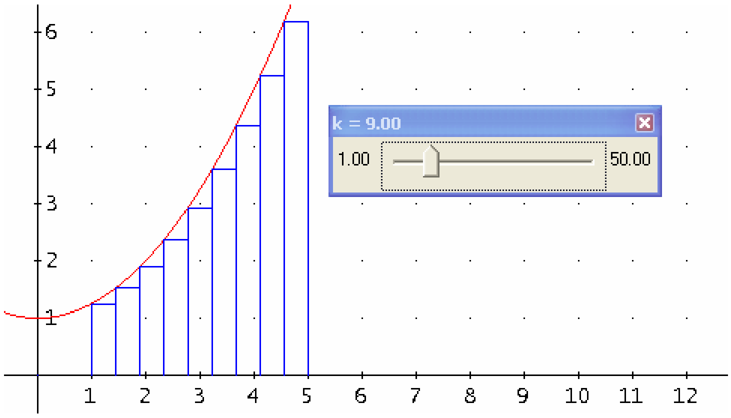
#20: $$\text{leftsums}(u, v, a, b, n) := \text{VECTOR}\left(\text{VECTOR}\left(\begin{bmatrix} a + \frac{(i-1)\cdot(b-a)}{n\_} & 0 \\ a + \frac{(i-1)\cdot(b-a)}{n\_} & \text{SUBST}\left(u, v, a + \frac{(i-1)\cdot(b-a)}{n\_}\right) \\ a + \frac{i\cdot(b-a)}{n\_} & \text{SUBST}\left(u, v, a + \frac{(i-1)\cdot(b-a)}{n\_}\right) \\ a + \frac{i\cdot(b-a)}{n\_} & 0 \end{bmatrix}, i, n\_\right), n\_, n\right)$$

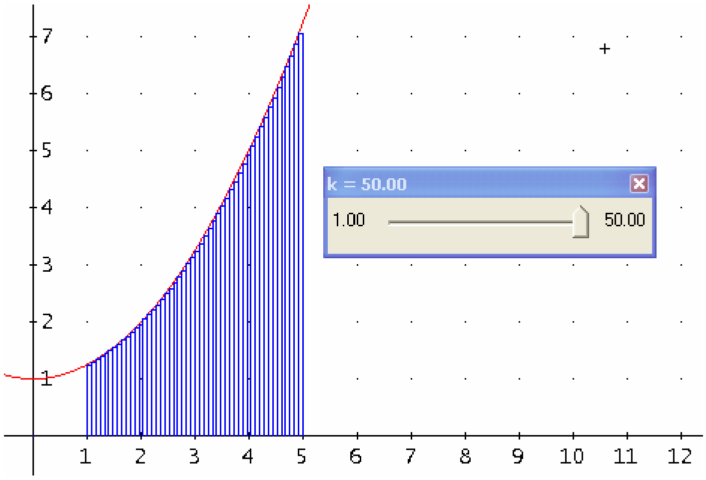
#21: $$\text{ls} := \text{leftsums}\left(\frac{x^2}{4} + 1, x, 1, 5, 50\right)$$

#22: Slide_Show(ls, k)

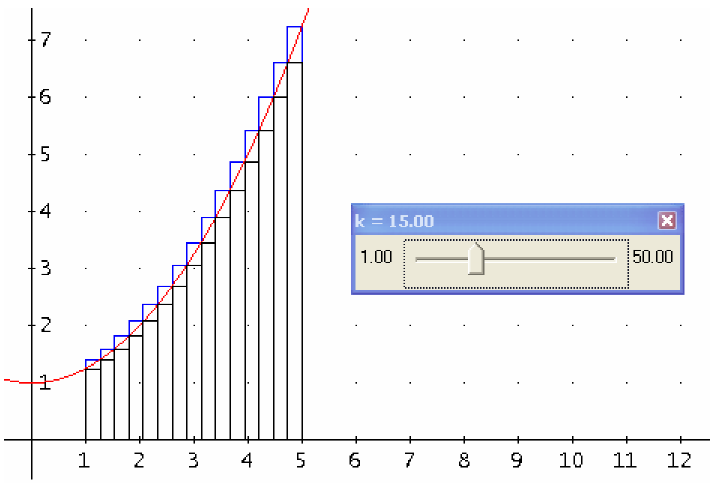
Insert a slider bar for k = 1 ... 50 with 49 intervals, plot the function and then plot #22.



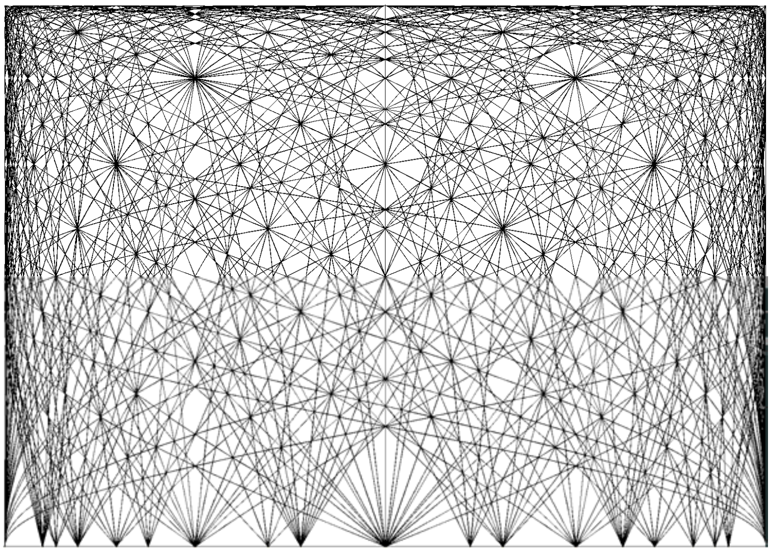


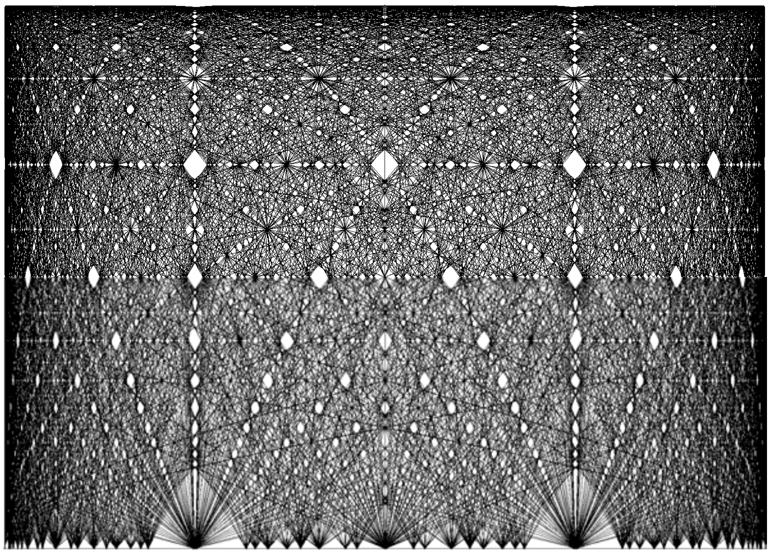


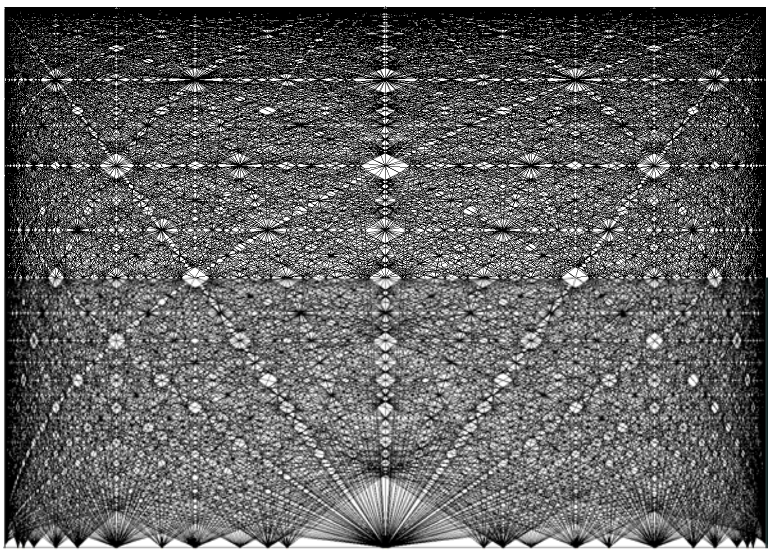
Do the same for thr right sum (or the midpoint sum or the trapezium sum) and visualise the approach from both sides to the area.



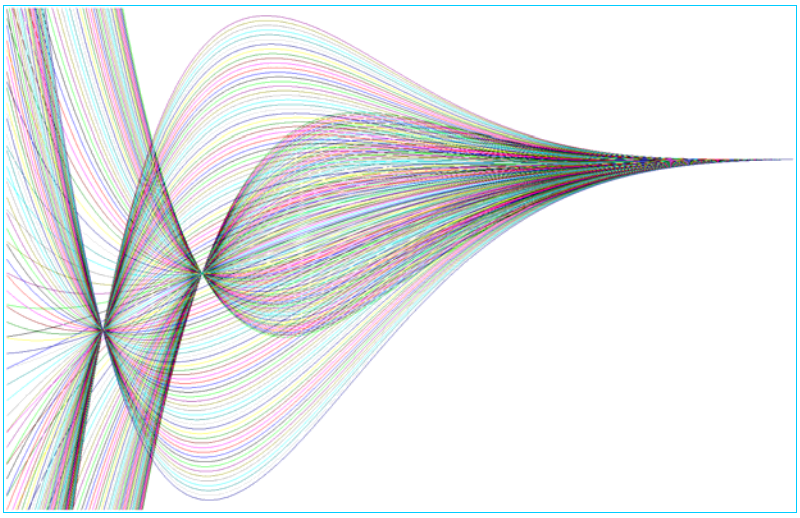
Black and White

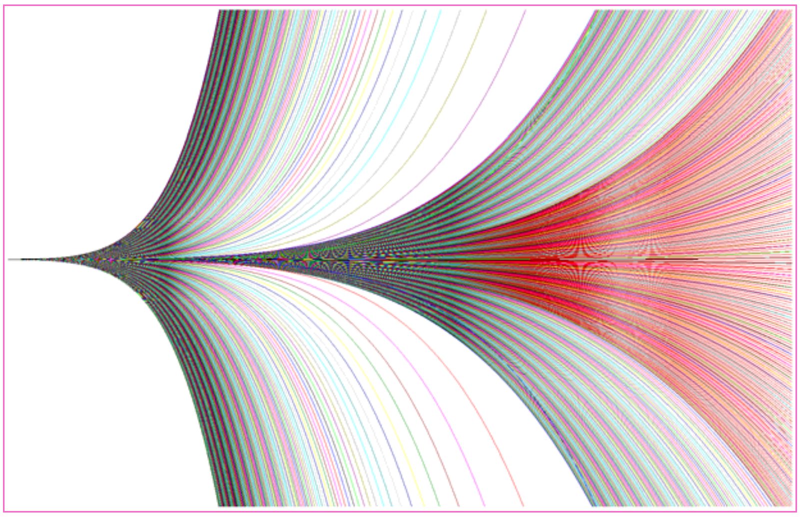


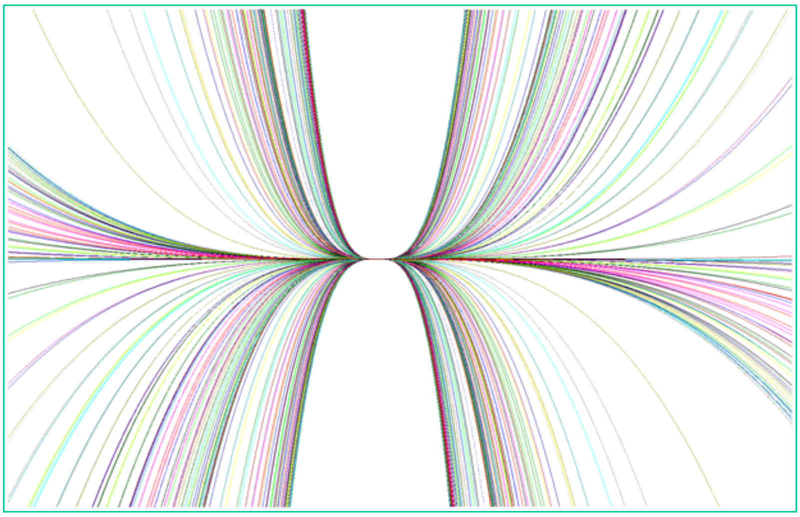




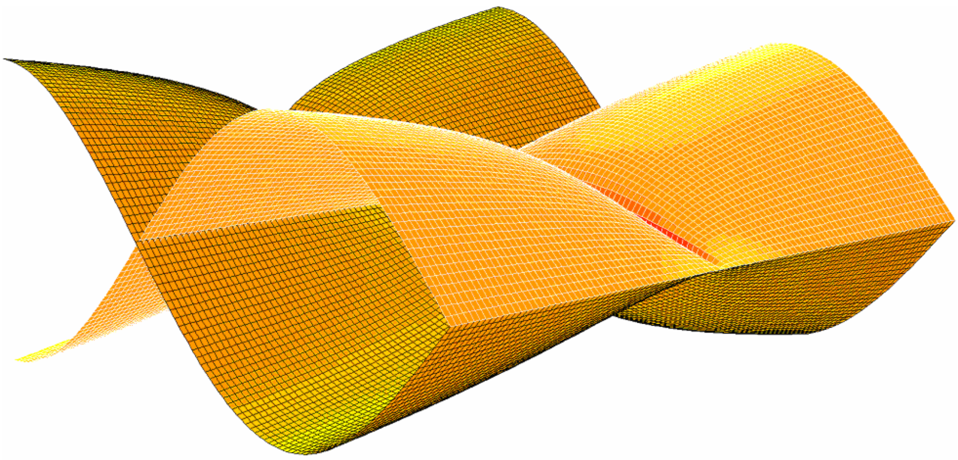
Impressions in Pastel

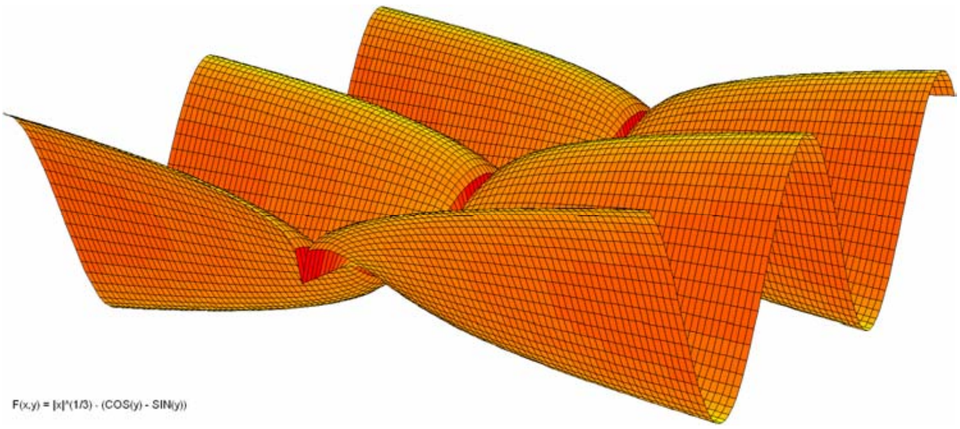


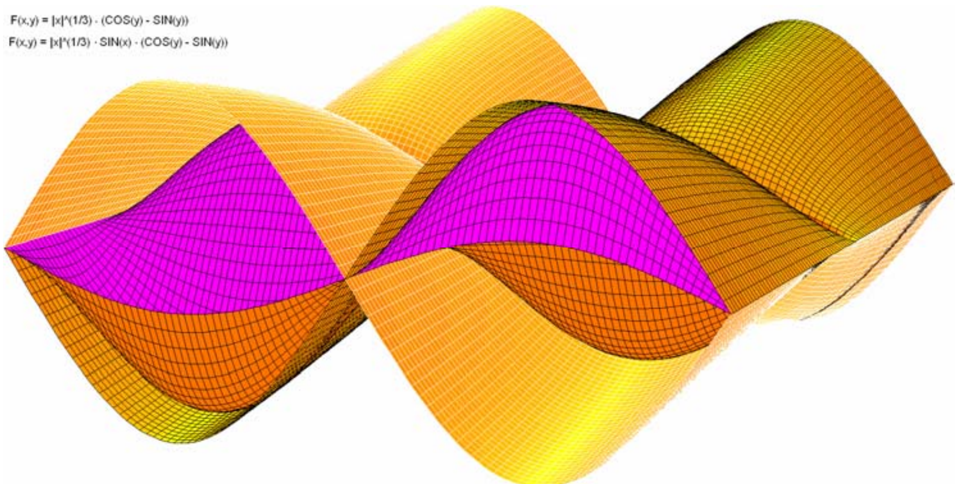




3D Heat Waves







Titbits from Algebra and Number Theory (33)

by Johann Wiesenbauer, Vienna

This column deals with permutations on a finite set, say {1,…,n}, which have received a lot of attention, as they occur in many places in mathematics and elsewhere. Just as a starter let me quote the following nice problem by Ivars Peterson at http://my.opera.com/mathmhb/blog/show.dml/415087

"The names of 100 prisoners are placed in 100 wooden boxes, one name to a box, and the boxes are lined up on a table in a room. One by one,the prisoners are led into the room. They may look into up to 50 of the boxes to try to find their own name, but must leave the room exactly as it was.The prisoners are permitted no further communication after leaving the room. They do have a chance to plot a strategy in advance. Good thing. Unless they all find their own names, they will all be executed. If each prisoner examines 50 boxes at random, the probability of the group's survival is a miniscule 1/2^100 [...]. However, there is a strategy that the prisoners can use to increase the probability of success to more than 30 percent. Incredible but true! The trick is to find this strategy."

Believe me, the solution is a real jawdropper! Hence, before reading on, it might be a good idea to stop here and have a try at this nice problem yourself!

As you may imagine by now, it has to do with special properties of permutations. First the prisoners have to agree on some bijection between the boxes and their names. For the sake of convenience, we assume that both the boxes and the prisoners are numbered with the numbers 1,2,..,100. Then each prisoner, if it is his turn, goes to the box whose number corresponds to his number, reads the name contained in it and interprets it as a pointer to the next box, again reads the name contained in it and uses it as a pointer to the next box etc.

It should be clear by now that this strategy is successful if and only if the underlying permutation (consisting exactly of the pointers above!) has got no cycles of length greater than 50. For people, who are good at combinatorics, in particular as regards the manipulation of generating functions, it is not too hard to identify the probability that a permutation on {1,2,…,n} has no cycle longer than k as the coefficient of x^n in the taylor series of $\exp(x + x^2/2+…+x^k/k)$.

This is where Derive comes into play! Again, I invite you to write a routine for the computation of this probability for general values of n and k yourself, before looking at my solution below. (It is the final solution after quite a few attempts before had ended in an overflow of memory or took too much time!)

```
prob0(n, k, n_ := 0, r_ := 1, s_ := 0, t_) :=
  Prog
    t_ := Σ(x^k_/k_, k_, 1, k)
    Loop
      s_ :+ POLY_COEFF(r_, x, n)
      n_ :+ 1
      If n_ > n
         RETURN s_
      r_ := REMAINDER(r_·t_, x^(n + 1))/n_
```

$$\text{TABLE(prob0(4, k), k, 1, 4)'} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ \frac{1}{24} & \frac{5}{12} & \frac{3}{4} & 1 \end{bmatrix}$$

Can you check this manually?

$$\text{prob0(100, 50)} = \frac{21740752665555669024605519989564940543418 3}{6972037522971247716453380893531230355680 0}$$ (3.75s)

$$\text{APPROX}\left(\frac{217407526655556690246055199895649405434183}{697203752297124771645338089353123035568 00}\cdot 100\right) = 31.18278206$$

If all this looked a little frightening to you, here comes a bit of relief. Provided that $2k+1 \geq n$, which is certainly fulfilled in the case at issue, there is a much simpler formula to compute those probabilities. For the derivation of this formula, lets consider first the probability that a given permutation on {1,2,..,n} has a cycle of length j with $j \geq n/2$. Since there are $(j-1)!$ cycles of length j (note that we may always assume w.l.o.g. that the smallest element of the cycle is also the very first one) and $n(n-1)...(n-(n-j)+1)$ is the number of possible assignments for the remaining elements, we get

$$\frac{n(n-1)...(j+1)(j-1)!}{n!}$$

for our probability, which simplifies after cancelling to the surprisingly simple expression $1/j$. Hence, by summing up these probabilities for $j = k+1,...,n$ and forming the counter probability, we get the value

$$1 - \sum_{j=k+1}^{n} \frac{1}{j} \approx 1 - \ln\frac{n}{k}$$

for the probability that there are no cycles of length greater than k for any given k with $2k+1 \geq n$. (Keep always in mind the latter restriction for the validity of this formula!) If you know the approximation $\ln 2 \approx 0.693$ (and every educated mathematician is expected to know this value by heart, isn't he?), then you get the rather good – though a little too small - approximation 0.307 for the probability we are aiming at, where $n/k = 2$.

Using this knowledge, we can implement a far better version of prob0(n,k). (Well, as its name already indicates, we had something better in mind all the time!)

```
prob(n, k) :=
  If 2·k + 1 < n
     prob0(n, k)
     1 - Σ(1/j_, j_, k + 1, n)
```

$$\text{prob(100, 50)} = \frac{217407526655556690246055199895649405434183}{697203752297124771645338089353123035568 00}$$ (0.000s !!)

This shows again that what we really need is not faster computers, but better algorithms!

Ok, let's turn to a different topic, which is the efficient traversing of all permutations in a given order (in the following always the lexicographical order). Since permutations on {1,2,..,n} have the annoying property that their total number n! increases very much with n, it is impossible for bigger values of n, say n > 9, to have a copy of each one in memory simultaneously. For most applications it suffices though, to find the "next" permutation of a given permutation wrt to the given order. Furthermore, usually certain conditions must be fulfilled, i.e. not all numbers are allowed in all positions. (Note that in the following routines the list of "forbidden" elements for each position can be found in the vector v.) Hence, we have to deal with the problem to find the next "valid" permutation.

It turns out though that we first need another important routine that will find for us the very first valid permutation under the given conditions and wrt the given order.

```
first_perm(s, v := [], f_, n_, s_, v_) :=
  Prog
    If s = []
       RETURN []
    If v = []
       v := VECTOR([], k_, DIM(s))
    f_ := SELECT(¬ MEMBER?(f__, FIRST(v)), f__, s)
    If f_ = []
       RETURN ?
    f_ := FIRST(f_)
    v_ := REST(v)
    s_ := SELECT(s__ ≠ f_, s__, s)
    If first_perm(s_, v_) = ?
       first_perm(s, ADJOIN(ADJOIN(f_, FIRST(v)), REST(v)))
       ADJOIN(f_, first_perm(s_, v_))
```

Note that if v is empty (a rather sloppy way to say that v consists of empty lists only, i.e. no elements are forbidden in any position), then the very first permutation is always the identity mapping on {1,2,…,n}.

As for next_perm(p,v), assuming that p is a valid permutations wrt v, we are looking for the rightmost position of a possible transposition. Let i and j with i<j be the corresponding positions in p. Then we first perform that transposition and then look for the very permutation that leaves the positions 1,2,..,I unchanged. For the rest, I leave it to you to find out the details.

```
next_perm(p, v := [], p_ := [], q_, r_, s_, t_, v_ := []) :=
  Prog
    If v = []
       v := VECTOR([], k_, DIM(p))
    p := REVERSE(p)
    v := REVERSE(v)
    Loop
      p_ := ADJOIN(FIRST(p), p_)
      p := REST(p)
      If p = []
         RETURN ?
      v_ := ADJOIN(FIRST(v), v_)
      v := REST(v)
      q_ := SELECT(FIRST(p) < q__ ∧ ¬ MEMBER?(q__, FIRST(v)), q__, SORT(p_))
      Loop
        r_ := ADJOIN(FIRST(p), SELECT(p__ ≠ FIRST(q_), p__, p_))
        r_ := first_perm(SORT(r_), v_)
        If r_ ≠ ? exit
        q_ := REST(q_)
        If q_ = [] exit
      If q_ ≠ []
         RETURN APPEND(REVERSE(ADJOIN(FIRST(q_), REST(p))), r_)
```

```
ITERATE(next_perm(p_), p_, [1, ..., 7], 7! - 1) = [7, 6, 5, 4, 3, 2, 1]
```

Note also the totally "index-free" style of programming in the programs above, something I'm really very proud of! In my experience, people coming from other programming languages like Pascal, VB, and diverse C-dialects etc. have often really a problem to get used to this style. Ok, here we go again, you may be thinking! You are right, I presume that you are eager by now to see these routines at work.

Hence, let's use them for example to classify all quasigroups (or latin squares, if you prefer that) on the set {1,2,…,n}. They are essentially (nxn)- matrices with the property that all rows and columns are permutations of {1,2,..,n}. In other words, each new row has the property that it is a permutation where the vector v of "forbidden" elements is exactly the transpose of the matrix built so far. The boolean variable r, which is set to true by default, determines whether we are only interested in so-called "reduced" quasigroups, which have the property that the first row and the first column are both the identity mapping. (Note that you can get all other quasigroups by first permuting the column and afterwards the rows, leaving the very first row fixed. In particular, the total number of quasigroups is bigger by a factor n!(n-1)!.)

```
quasigroups(n, r := true, s := [[]], p_, q_) :=
  Prog
    If r ∧ s = [[]]
       s := [[[1, ..., n]]]
    Loop
      q_ := FIRST(s)
      If DIM(q_) = n
         RETURN s
      s := REST(s)
      p_ := first_perm([1, ..., n], q_')
      Loop
        If p_ = ? exit
        If r ∧ FIRST(p_) > DIM(q_) + 1 exit
        s := APPEND(s, [APPEND(q_, [p_])])
        p_ := next_perm(p_, q_')
```

$$\text{quasigroups}(4) = \left[\begin{bmatrix}1&2&3&4\\2&1&4&3\\3&4&1&2\\4&3&2&1\end{bmatrix}, \begin{bmatrix}1&2&3&4\\2&1&4&3\\3&4&2&1\\4&3&1&2\end{bmatrix}, \begin{bmatrix}1&2&3&4\\2&3&4&1\\3&4&1&2\\4&1&2&3\end{bmatrix}, \begin{bmatrix}1&2&3&4\\2&4&1&3\\3&1&4&2\\4&3&2&1\end{bmatrix}\right]$$

```
DIM(quasigroups(4, false)) = 576
DIM(quasigroups(5)) = 56
```

As the notation "quasigroups" already indicates, there is a close relationship of quasigroups to groups. More precisely, every group G is a quasigroup, where additionally the so-called associativity law $(xy)z = x(yz)$ for all $x, y, z \in G$ holds. This then implies the existence of an identity element $e \in G$ such that $ex = xe = x$ for all $x \in G$ and the existence of an inverse element $x' \in G$ for each $x \in G$ such that $x'x = xx' = e$. (Try to prove this!)

Note that all 4-element reduced quasigroups above are actually groups. This is no longer true for 5-element reduced quasigroups (or slightly more generally, quasigroups with an identity element, which are also called loops), as you may check yourself referring to the complete list of those quasigroups below, which was established using a special auxiliary routine for the output of such lists in matrix form.

```
matrix(u, k) :=
  Prog
    Loop
      If MOD(DIM(u), k) = 0 exit
      u := APPEND(u, [""])
    VECTOR(u↓[k·j_ + 1, ..., k·j_ + k], j_, 0, FLOOR(DIM(u), k) - 1)
```

```
matrix(quasigroups(5), 8)
```

1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
2 1 4 5 3	2 1 4 5 3	2 1 4 5 3	2 1 4 5 3	2 1 5 3 4	2 1 5 3 4	2 1 5 3 4	2 1 5 3 4
3 4 5 1 2	3 4 5 2 1	3 5 1 2 4	3 5 2 1 4	3 4 1 5 2	3 4 2 5 1	3 5 4 1 2	3 5 4 2 1
4 5 2 3 1	4 5 1 3 2	4 3 5 1 2	4 3 5 2 1	4 5 2 1 3	4 5 1 2 3	4 3 2 5 1	4 3 1 5 2
5 3 1 2 4	5 3 2 1 4	5 4 2 3 1	5 4 1 3 2	5 3 4 2 1	5 3 4 1 2	5 4 1 2 3	5 4 2 1 3
1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
2 3 1 5 4	2 3 1 5 4	2 3 1 5 4	2 3 1 5 4	2 3 4 5 1	2 3 4 5 1	2 3 4 5 1	2 3 4 5 1
3 4 5 1 2	3 4 5 2 1	3 5 4 1 2	3 5 4 2 1	3 1 5 2 4	3 1 5 2 4	3 4 5 1 2	3 5 1 2 4
4 5 2 3 1	4 5 2 1 3	4 1 5 2 3	4 1 5 3 2	4 5 1 3 2	4 5 2 1 3	4 5 1 2 3	4 1 5 3 2
5 1 4 2 3	5 1 4 3 2	5 4 2 3 1	5 4 2 1 3	5 4 2 1 3	5 4 1 3 2	5 1 2 3 4	5 4 2 1 3
1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
2 3 4 5 1	2 3 4 5 1	2 3 5 1 4	2 3 5 1 4	2 3 5 1 4	2 3 5 1 4	2 3 5 1 4	2 3 5 1 4
3 5 2 1 4	3 5 2 1 4	3 1 4 5 2	3 1 4 5 2	3 4 1 5 2	3 4 2 5 1	3 4 2 5 1	3 5 4 2 1
4 1 5 2 3	4 1 5 3 2	4 5 1 2 3	4 5 2 3 1	4 5 2 3 1	4 5 1 2 3	4 5 1 3 2	4 1 2 5 3
5 4 1 3 2	5 4 1 2 3	5 4 2 3 1	5 4 1 2 3	5 1 4 2 3	5 1 4 3 2	5 1 4 2 3	5 4 1 3 2
1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
2 4 1 5 3	2 4 1 5 3	2 4 1 5 3	2 4 1 5 3	2 4 1 5 3	2 4 1 5 3	2 4 5 1 3	2 4 5 1 3
3 1 5 2 4	3 5 2 1 4	3 5 2 1 4	3 5 4 1 2	3 5 4 2 1	3 5 4 2 1	3 1 2 5 4	3 1 4 5 2
4 5 2 3 1	4 1 5 3 2	4 3 5 2 1	4 3 5 2 1	4 1 5 3 2	4 3 5 1 2	4 5 1 3 2	4 5 2 3 1
5 3 4 1 2	5 3 4 2 1	5 1 4 3 2	5 1 2 3 4	5 3 2 1 4	5 1 2 3 4	5 3 4 2 1	5 3 1 2 4
1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
2 4 5 1 3	2 4 5 1 3	2 4 5 3 1	2 4 5 3 1	2 4 5 3 1	2 4 5 3 1	2 4 5 3 1	2 4 5 3 1
3 5 1 2 4	3 5 4 2 1	3 1 2 5 4	3 1 4 5 2	3 1 4 5 2	3 5 1 2 4	3 5 2 1 4	3 5 4 1 2
4 3 2 5 1	4 3 1 5 2	4 5 1 2 3	4 5 1 2 3	4 5 2 1 3	4 1 2 5 3	4 3 1 5 2	4 1 2 5 3
5 1 4 3 2	5 1 2 3 4	5 3 4 1 2	5 3 2 1 4	5 3 1 2 4	5 3 4 1 2	5 1 4 2 3	5 3 1 2 4
1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
2 5 1 3 4	2 5 1 3 4	2 5 1 3 4	2 5 1 3 4	2 5 1 3 4	2 5 1 3 4	2 5 4 1 3	2 5 4 1 3
3 1 4 5 2	3 4 2 5 1	3 4 2 5 1	3 4 5 1 2	3 4 5 1 2	3 4 5 2 1	3 1 2 5 4	3 1 5 2 4
4 3 5 2 1	4 1 5 2 3	4 3 5 1 2	4 1 2 5 3	4 3 2 5 1	4 1 2 5 3	4 3 5 2 1	4 3 1 5 2
5 4 2 1 3	5 3 4 1 2	5 1 4 2 3	5 3 4 2 1	5 1 4 2 3	5 3 4 1 2	5 4 1 3 2	5 4 2 3 1
1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
2 5 4 1 3	2 5 4 1 3	2 5 4 1 3	2 5 4 1 3	2 5 4 3 1	2 5 4 3 1	2 5 4 3 1	2 5 4 3 1
3 1 5 2 4	3 4 1 5 2	3 4 2 5 1	3 4 5 2 1	3 1 2 5 4	3 1 5 2 4	3 4 1 5 2	3 4 5 1 2
4 3 2 5 1	4 3 5 2 1	4 1 5 3 2	4 3 1 5 2	4 3 5 1 2	4 3 1 5 2	4 1 5 2 3	4 1 2 5 3
5 4 1 3 2	5 1 2 3 4	5 3 1 2 4	5 1 2 3 4	5 4 1 2 3	5 4 2 1 3	5 3 2 1 4	5 3 1 2 4

But there are other well-known mathematical objects, which can also be viewed as special quasigroups, and this time they are even known to the proverbial man in the street. Just in case, that didn't ring a bell yet, I'm talking about sudokus.

Well, as you may remember, we have already dealt with sudokus in this series, but using a completely different approach. Now we are viewing them as special quasigroups on the set {1,2,..,n}, where n must be of the form $n = m^2$ for positive integer m, and try to take advantage of our routines dealing with permutations.

The parameter s is assumed to contain the usually sparse matrix with some prefilled entries, where trailing zero rows may be omitted. Note that in the case s=[], i.e. even if there are no prefilled squares, we set the first line to $[1,2,\ldots,m^2]$.

```
sudokus(m, s := [], all := true, i_ := 1, j_ := 1, n_, p_, q_ := [], r_, s_, t_ := [], u_, v_, w_ := []) :=
  Prog
    n_ := m^2
    If s = []
       s := [[1, ..., n_]]
    Loop
      If DIM(s) = n_ exit
      s := APPEND(s, [VECTOR(0, k_, n_)])
    s_ := s
    Loop
      u_ := m·FLOOR(i_ - 1, m)
      v_ := m·FLOOR(j_ - 1, m)
      r_ := {u_ + 1, ..., u_ + m}·{v_ + 1, ..., v_ + m}
      r_ := (r_ ∪ {i_}·{1, ..., n_} ∪ {1, ..., n_}·{j_}) \ {[i_, j_]}
      w_ := ADJOIN(r_, w_)
      j_ :+ 1
      If j_ > n_
         Prog
           i_ :+ 1
           j_ := 1
           t_ := ADJOIN(REVERSE(w_), t_)
           w_ := []
      If i_ > n_ exit
    t_ := REVERSE(t_)
    i_ := 1
    j_ := 0
    Loop
      v_ := []
      w_ := s_↓i_
      u_ := t_↓i_
      Loop
        If FIRST(w_) > 0
           r_ := {1, ..., n_} \ {FIRST(w_)}
           r_ := {}
        r_ := r_ ∪ MAP_LIST(s_↓(x_↓1)↓(x_↓2), x_, FIRST(u_))
        v_ := ADJOIN(SORT(r_), v_)
        w_ := REST(w_)
        u_ := REST(u_)
        If w_ = [] exit
      v_ := REVERSE(v_)
      If i_ > j_
         p_ := first_perm([1, ..., n_], v_)
         p_ := next_perm(p_, v_)
      j_ := i_
      If p_ = ?
         Prog
           s_ := REPLACE(s↓i_, s_, i_)
           i_ :- 1
           If i_ > 0
              [p_ := s_↓i_, s_ := REPLACE(s↓i_, s_, i_)]
         Prog
           s_ := REPLACE(p_, s_, i_)
           If i_ < n_
              i_ :+ 1
              [IF(¬ all, RETURN s_), q_ := ADJOIN(VECTOR(s__, s__, s_), q_)]
      WRITE(i_)
      If i_ = 0
         RETURN REVERSE(q_)
```

```
matrix(sudokus(2), 6)
```

$$\left[\begin{array}{cccccc}
\begin{bmatrix}1&2&3&4\\3&4&1&2\\2&1&4&3\\4&3&2&1\end{bmatrix} &
\begin{bmatrix}1&2&3&4\\3&4&1&2\\2&3&4&1\\4&1&2&3\end{bmatrix} &
\begin{bmatrix}1&2&3&4\\3&4&1&2\\4&1&2&3\\2&3&4&1\end{bmatrix} &
\begin{bmatrix}1&2&3&4\\3&4&1&2\\4&3&2&1\\2&1&4&3\end{bmatrix} &
\begin{bmatrix}1&2&3&4\\3&4&2&1\\2&1&4&3\\4&3&1&2\end{bmatrix} &
\begin{bmatrix}1&2&3&4\\3&4&2&1\\4&3&1&2\\2&1&4&3\end{bmatrix} \\
\begin{bmatrix}1&2&3&4\\4&3&1&2\\2&1&4&3\\3&4&2&1\end{bmatrix} &
\begin{bmatrix}1&2&3&4\\4&3&1&2\\3&4&2&1\\2&1&4&3\end{bmatrix} &
\begin{bmatrix}1&2&3&4\\4&3&2&1\\2&1&4&3\\3&4&1&2\end{bmatrix} &
\begin{bmatrix}1&2&3&4\\4&3&2&1\\2&4&1&3\\3&1&4&2\end{bmatrix} &
\begin{bmatrix}1&2&3&4\\4&3&2&1\\3&1&4&2\\2&4&1&3\end{bmatrix} &
\begin{bmatrix}1&2&3&4\\4&3&2&1\\3&4&1&2\\2&1&4&3\end{bmatrix}
\end{array}\right]$$

$$\text{sudokus}\left(3,\ \begin{bmatrix} 0&0&0&0&0&7&5&0&0\\ 0&4&0&0&0&6&3&0&0\\ 0&0&2&8&0&0&0&0&0\\ 0&1&8&0&0&0&0&0&0\\ 5&3&0&0&0&0&0&2&7\\ 0&0&0&0&0&0&9&6&0\\ 0&0&0&0&0&4&1&0&0\\ 0&0&7&9&0&0&0&8&0\\ 0&0&6&2&0&0&0&0&0 \end{bmatrix}\right) = \left[\begin{bmatrix} 8&6&1&3&4&7&5&9&2\\ 9&4&5&1&2&6&3&7&8\\ 3&7&2&8&5&9&6&1&4\\ 6&1&8&7&9&2&4&3&5\\ 5&3&9&4&6&1&8&2&7\\ 7&2&4&5&3&8&9&6&1\\ 2&8&3&6&7&4&1&5&9\\ 4&5&7&9&1&3&2&8&6\\ 1&9&6&2&8&5&7&4&3 \end{bmatrix}\right]$$

As a rule of thumb, the run time is several minutes rather than several seconds (the example above took about 6 min on my PC). Hence, this is not meant as a sudoku-solver, but rather as a demonstration that our ideas work. On the other hand, it will solve any suduko reliably and also tell you all solutions, if there is more than one. (At least, this is the default setting, which be changed though by setting the parameter all to false.) Well, and not to forget, it also works for other values of m than the “classical” case m=3, at least in principle. Let’s conclude with the following example, where m=4.

```
sudokus(4, [], false)
```

$$\begin{bmatrix} 1&2&3&4&5&6&7&8&9&10&11&12&13&14&15&16\\ 5&6&7&8&1&2&3&4&13&14&15&16&9&10&11&12\\ 9&10&11&12&13&14&15&16&1&2&3&4&5&6&7&8\\ 13&14&15&16&9&10&11&12&5&6&7&8&1&2&3&4\\ 2&1&4&3&6&5&8&7&10&9&12&11&14&13&16&15\\ 6&5&8&7&2&1&4&3&14&13&16&15&10&9&12&11\\ 10&9&12&11&14&13&16&15&2&1&4&3&6&5&8&7\\ 14&13&16&15&10&9&12&11&6&5&8&7&2&1&4&3\\ 3&4&1&2&7&8&5&6&11&12&9&10&15&16&13&14\\ 7&8&5&6&3&4&1&2&15&16&13&14&11&12&9&10\\ 11&12&9&10&15&16&13&14&3&4&1&2&7&8&5&6\\ 15&16&13&14&11&12&9&10&7&8&5&6&3&4&1&2\\ 4&3&2&1&8&7&6&5&12&11&10&9&16&15&14&13\\ 8&7&6&5&4&3&2&1&16&15&14&13&12&11&10&9\\ 12&11&10&9&16&15&14&13&4&3&2&1&8&7&6&5\\ 16&15&14&13&12&11&10&9&8&7&6&5&4&3&2&1 \end{bmatrix}$$

A Conjecture – A Theorem?

Heinrich Ludwig, Germany

Lieber Herr Böhm,

ich danke Ihnen für Ihre schnelle Auskunft! Die Arbeit von Daniel J. Bernstein konnte ich über das Internet beziehen. Mit dem Studium derselben muss ich mich leider noch gedulden. Meine Arbeit im Dienst fordert mich zur Zeit deutlich.

Vielleicht interessiert es Sie, dass ich mit DERIVEs Hilfe auf eine (mich) verblüffende Vermutung gestoßen bin. Um den Aufgabenfundus für meinen Leistungskurs zu erweitern, habe ich Kreise im 2-dimensionalen Raum (wie üblich in der Schule mit euklidischer Metrik) gesucht, die durch 3 Punkte mit Ganzzahl-Koordinaten gehen und die zugleich einen rationalen Radius haben. Man findet eine ganze Menge Lösungen. Sortiert man sie der Größe nach, so ergibt sich eine Folge von Brüchen, die einigermaßen irregulär erscheint. Neugierig geworden, habe ich DERIVE dann entsprechende Lösungen im 3-, 4-, 5- und 6-dimensionalen euklidischen Raum suchen lassen. Das Erstaunliche ist, dass im IR_5 (aber noch nicht im IR_4!) eine gewisse Vollständigkeit erreicht wird, die folgender Satz ausdrückt:

Sei q eine natürliche Zahl und p' := (q^2)/2 falls q ungerade, bzw. p' := (q^2)/4 falls q gerade, dann gibt es für jede natürliche Zahl p > p' drei Punkte im IR_5, so dass der Kreis durch diese Punkte den Radius p/q hat.

Zu einem Beweis fehlt mir leider jeder Anhaltspunkt. Trotz tagelanger Rechenzeit hat sich aber auch kein Gegenbeispiel eingestellt. - Die Umkehrung des Satzes gilt nicht, es gibt außer den eben beschriebenen Umkreisradien einige sporadische Werte, die sich einer naheliegenden Einordnung entziehen. Neugierig geworden, was höhere Dimensionen noch zu bieten hätten, untersuchte ich den IR_6 ebenso. Erstaunlich ist, dass im IR_6 keine weiteren Lösungen hinzukommen. Höherdimensionale Räume als den IR_6 sind schwer ausreichend zu untersuchen, weil die Rechenzeit mit der Dimension dramatisch wächst.

Ich räume ehrlichkeitshalber ein, dass ich nach meinen ersten Berechnungen auf Programme überging, die in Delphi (Pascal-Abkömmling) geschrieben sind, weil damit eine gegenüber DERIVE 800-fach höhere Rechengeschwindigkeit möglich war. Aber: DERIVE hat die entscheidenden Hinweise gegeben, die mich motiviert haben tiefer nachzubohren!

Mit freundlichem Gruß

Ihr Heinrich Ludwig

P.S. Meine ersten Ergebnisse habe ich (diese alle noch mit DERIVE ermittelt) in der bekannten "On-Line Encyclopedia of Integer Sequences" von Neil Sloane unter den Indizes A128006 ff. veröffentlicht.

I try to translate Heinrich´s interesting findings (Josef).

I found a (for me) surprising conjecture (supported by DERIVE). Preparing problems for my classes I looked for circles having a rational radius and passing points with integer coordinates. One can find a lot of solutions. Sorting them in increasing order one obtains a sequence or fractions which seems to be quite irregular. Extending the research for 3, 4, 5 and 6 dimensional space I noticed that in IR_5 a certain completeness seems to be reached which can be expressed as follows:

Let q a natural number and p' :=q^2/2 if q odd and p'=q^2/4 if q even, then for every natural number p > p' exist three points with integer coordinates in IR_5 such that the circle passing these points has a rational radius p/q.

Unfortunately I don´t have any idea for a proof. I worked for days and couldn´t find any counter example. Reversion of the theorem is not valid. Then I extended my search to IR_6. It is surprising that there are no additional solutions appearing. Investigations of higher dimensioned spaces are difficult, because calculation times will increase dramatically.

To be honest, I must admit, that after my first calculations I changed to programs written in Delphi, because calculation time is 800 times faster. But it was DERIVE which gave important hints, which inspired me to get deeper into the matter.

I published my first results (all of them found with DERIVE) in the well known "On-Line-Encyclopedia of Integer Sequences" of Neil Sloane under the index numbers A128006 ff.