THE DERIVE - NEWSLETTER #26

ISSN 1990-7079

THE BULLETIN OF THE

DERIVE

USER GROUP

+ TI 92

Contents:

revised 2011 June 1997

[1] **Teaching Mathematics with *DERIVE* and the *TI-92***, Proceedings of the International DERIVE and TI-92 Conference 1996 at Schloss Birlinghoven, Bonn.
543 pages with 54 contributions (13 German, 39 English), available at ZKL, Prinzipalmarkt 38, D 48143 Muenster, Germany; email: bemtz@uni-muenster.de. (DEM 50.00)

[2] **La calcolatrici grafiche nell'insegnamento dell matematica**, G.C.Barozzi, S Cappuccio
146 pages, Pitagora Editrice Bologna 1997, ISBN 88-371-0891-5
This new booklet is addressed to Secondary School teachers and deals with graphic and programmable calculators (most TI-92) in Maths teaching and learning.
(Arithmetic, Algebraicallanguages, Calculators as a tool for Algebra, C. as an aid for learning, The discovery method, C. in problem solving, C. as a blackboard, Geometry, Locus of points: conics, Programming, Lists, Geometric transformations. A "Matura" problem solved via TI-92)

[3] **Mathematical Activities with *DERIVE***, Edited by E. Graham, J.S. Berry and A.J.P. Watkins, 207 pages, Chartwell-Bratt 1997, ISBN 0-86238-478-8
This book brings together a range of interesting examples of the use of DERIVE which have been contributed by a number of DERIVE users (22) from around the world.
(What was the question?, DERIVE and Mechanics, Linear Algebra Using DERIVE and Methods of Assessment, A Modelling Workshop, The Zeros of a Cubic Polynomial, Examples of Studies of Simple Dynamic Systems, Construction of Fractal Sets, Concepts of Differential and Integral Calculus and others.)

[4] **The State of Computer Algebra in Mathematics Education**, Edited by I.S. Berry, J. Monaghan, M. Kronfellner, B. Kutzler, 213 pages, Chartwell-Bratt 1997, ISBN 0-86238-430-3
A report of the International Symposium on Computer Algebra in Mathematics Education held in Hawaii, August 1995.
(The Curriculum, Assessment, The design of CAS written support material, Attracting Teachers to Use CAS in the Classroom, Understanding)

[5] **A Survey of Models for Tumor-lmmune System Dynamics**, Editors J.A. Adam, N. Bellomo, 344 pages, Birkhauser Boston 1997, ISBN 0-8176-3901-2
This unique book is a collection of seven interdisciplinary surveys on modeling tumor dynamics and interactions between tumors and the immune system. It is a excellent resource and survey for **applied mathematicians, mathematical biologist** and biologists interested in **modeling methods** in immunology and related sciences.
Among others you can find various Diffusion Models, The Predator-Prey Approach, A Model of Tumor Cell/Immune System Interaction, Comments on Catastrophe Theory , Mathematical Modeling of Tumor Gromh Kinetics, Tumor Gromh as a Dynamical System, Logistic Model, Multicell Spheroid Model. References for further reading

Interesting WEB sites

http://www.imag. fr/imag/cabri.html	from the Cabri developers in Grenoble
http://www.ti. com/calc/docs/cabri .htm	from Texas Instruments
http://forum.swarthmore.edu/	including The Math Forum, Ask Dr. Math, Internet Resources, Cabri Information, ... (cti, Birmingham)
http://www .cut-the-knot.com/	Interactive Mathematics and Puzzles by Alexander Bogomolny. An award winning Math site on a variety of topics, recommended by Jan Vermeylen

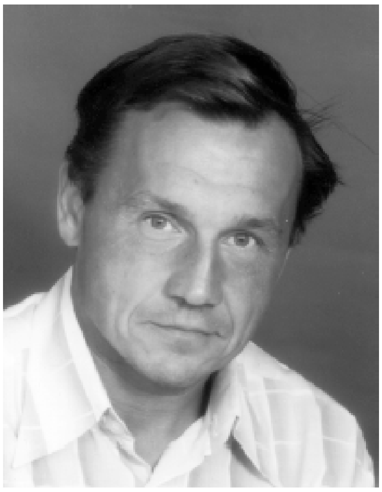
Please let us know your favourite site!

Liebe DERIVE- und TI-Freunde,

Der letzte Schritt in der Fertigstellung eines jeden DNL ist immer das Verfassen des Letters. Bis zum letzten Augenblick erreichen mich interessante Nachrichten und Noor, meine liebe Frau, drängt und drängt: „Jetzt mach doch endlich fertig!“

Aus den letzten drei Monaten habe ich besonders schöne Erinnerungen an die gelungene DERIVE- und TI-Tagung in Münster. Herzlichen Dank von dieser Stelle an die Organisatoren mit Dr. Berntzen und Bärbel Barzel an der Spitze. Auch T^3-Europe nimmt schön langsam Form und Gestalt an. Wenn Sie noch nicht wissen, was T^3 ist, dann besuchen Sie bitte die Homepage von Bert Waits.

Ich freue mich aber auch schon auf das vor uns liegende Treffen in Schweden Die FUN-Conference (Fun in Teaching Mathematics) versammelt vom 7. bis 9.August mehr als 30 Personen aus vier Kontinenten zu einem Austausch ihrer Ideen und Arbeiten. Viele Freunde werden sich wieder treffen und dort sicherlich neue Freunde gewinnen. Ich finde es sehr wichtig, dass, sich der Kreis immer wieder erweitert und erneuert. Damit ist doch sicher gestellt, dass wir nicht in unserer Routine erstarren. Da auch David Stoutemyer und Bert Waits dabei sind, können wir sicher sein, dass unsere Vorstellungen sofort an die richtige Adresse gelangen.



Vor der FUN-Konferenz gibt es am selben Ort (Kungsbacka) eine von David Sjöstrand gemeinsam mit der DUG organisierte Lehrerfortbildungsveranstaltung, bei der 50-60 Teilnehmer erwartet werden.

Und auch die „große“ Internationale DERIVE und TI-92 Konferenz 1998 in Gettysburg wirft schon ihre Schatten voraus. Beachten Sie bitte die Information ab Seite 11. Carl Leinbach hat sich bereits viel Mühe gemacht, ein wunderbares Nebenprogramm zusammen zu stellen. Mit viel Freude habe ich von ihm gehört, dass, er nach seinem Herzleiden wieder voll hergestellt ist. Carl und Pat, wir wünschen Euch alles Gute! Für den fachlichen Teil sind wir alle verantwortlich. Ich lade Sie herzlich ein, an dieser Konferenz als Vortragender, als Leiter eines Workshops oder einfach nur als Zuhörer teilzunehmen.

Allen Freunden mit Internet-Anschluss, möchte ich die angebotenen Webseiten wärmstens empfehlen. Sie entpuppen sich als wahre Fundgruben.

Ich wünsche Ihnen allen einen schönen Sommer mit erholsamen Ferien - unseren Freunden auf der anderen Halbkugel einen nicht zu unangenehmen Winter - und freue mich bereits auf den nächsten DNL mit hoffentlich vielen interessanten Neuigkeiten aus der Welt von DERIVE und TI-92.

Josef

Dear DERIVE- and TI-friends,

The last step completing each DNL is writing the Editor's Letter. Interesting news reach me until the last moment and my wife Noor is urging and urging. "It's time to come to an end!"

I have best memories on the excellent DERIVE and TI-92 Conference in Münster, Germany. Many thanks from here to the organizers led by Dr. Berntzen and Bärbel Barzel. T^3-Europe is now beginning to show shape and form. If you don't know what T^3 is then please visit Bert Waits' homepage.

I am looking forward to the meeting in Sweden ahead of us. The FUN-Conference (Fun in Teaching Mathematics) will gather more than 30 persons from four continents to exchange their ideas and work. Many friends will meet again and will make new ones. From my point of view it is very important to enlarge and to renew the circle. That will ensure that we will not stiffen in our routine. David Stoutemyer and Bert Waits will be with us, so we can be sure that our ideas and proposals will immediately reach the right address.

Just before the FUN-Conference a Teachers' Training Conference will take place in the same location (Kungsbacka) organized by David Sjöstrand and the DUG. We are expecting 50 -60 participants.

And above all, the "big" International DERIVE and TI-Conference casts its shadow ahead of it. Please note the information on page 11. Carl Leinbach has spared no effort to arrange a wonderful Spouses’ Program. I was very glad to hear from Carl that he has fully recovered from a severe heart attack. Carl and Pat, the best to you from all of us. We all are responsible for the technical part of the Conference. So I'd like to invite you all to attend to give lectures or to lead workshops or to participate and help to make this Conference as successful as the former ones have been.

I warmly recommend the websites presented on the information page. You will find great resources and many links to other sites.

I wish you all a wonderful summer with relaxing holidays - our friends living in the other hemisphere I wish a not too bad winter - and I am looking forward to the next DNL which will bring again many interesting news from the world of DERIVE and the TI-92.

Josef

The *DERIVE-NEWSLETTER* is the Bulletin of the *DERIVE User Group*. It is published at least four times a year with contents of 40 pages minimum. The goals of the *DNL* are to enable the exchange of experiences made with *DERIVE* as well as to create a group to discuss the possibilities of new methodical and didactical manners in teaching mathematics.
We include now a section dealing with the use of the TI-92 and we try to combine these modern technologies.

Editor: Mag. Josef Böhm
A-3042 Würmla
D´Lust 1
Austria
Phone: 43-(0)660 31 36 365
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 *DNL*. The more contributions you will send the more lively and richer in contents the *DERIVE Newsletter* will be.

Preview: Contributions for the next issues

3D-Geometry, Reichel, AUT
Algebra at A-Level, Goldstein, UK
Graphic Integration, Linear Programming, Various Projections. Böhm, AUT
A Utility file for complex dynamic systems, Lechner, AUT
Examples for Statistics, Roeloffs, NL
Solving Word problems (Textaufgaben) with DERIVE, Böhm, AUT
Line Searching with DERIVE, Collie, UK
About the "Cesaro Glove-Osculant", Halprin, AUS
Hidden lines, Weller, GER
Fractals and other Graphics, Koth, AUT
Experimenting with GRAM-SCHMIDT, Schonefeld, USA
Implicit Multivalue Bivariate Function 3D Plots, Biryukov, RUS

The TI-92 Section, Waits a.o.
and
Setif, FRA; Vermeylen, BEL; Leinbach, USA; Halprin, AUS; Speck, NZL; Weth, GER; Wiesenbauer, AUT; Aue, GER; Pröpper, GER; Koller, AUT; Stahl, USA; Mitic, UK; Tortosa, ESP; Santonja, ESP; Wadsack, AUT; Schorn, GER, and ....

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

<u>**Scott A. Guth, Mt San Antonio College**</u>

Hello all!

I know that this has come up recently, but I've lost the response to the question which was asked. To keep it simple, how can I fix the expression

```
[random(10), random(10)]
```

so that the same numer is returned in both positions of the vector?

<u>**pstoffel@uia.ua.ac.be**</u>

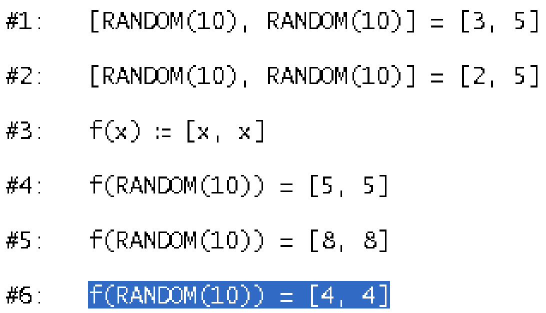
Albert D. Rich suggested the following method about a year ago (Tue 14 May 1996 for the mathematical correctness):

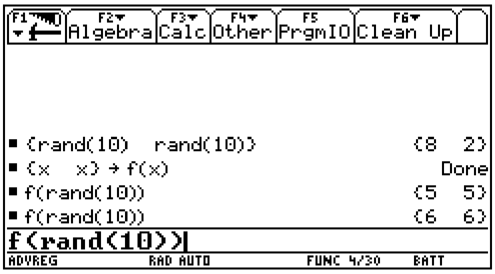
```
f(x) := [x,x]
```

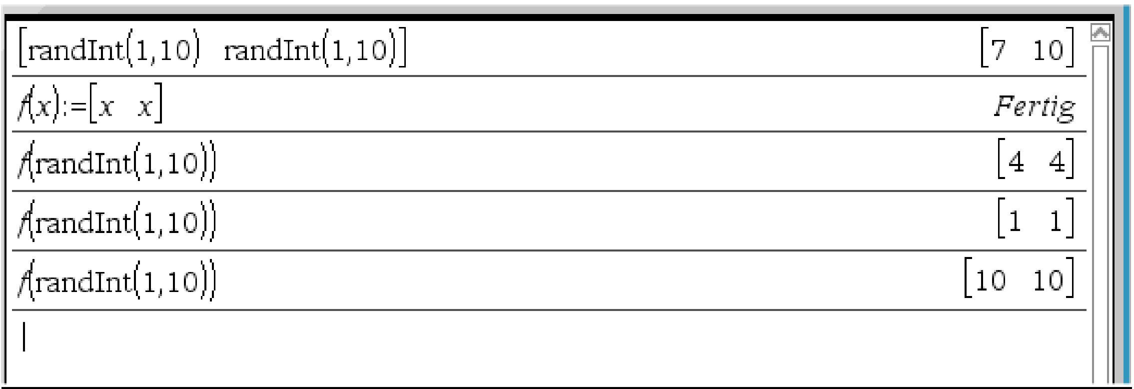
then simplify

```
f(random(10))
```

and then DERIVE returns the desired answer!







<u>DNL:</u> *As you can see the same procedure is necessary working on the TI-handhelds and with TI-NspireCAS as well.*

Jan Vermeylen, Kapellen, Belgium

1) While talking in class about singular conic sections I was embarrassed to see that $y^2 = 4$ is plotted with DfW 4.02 as two VERTICAL lines. However, $y = 2$ is correctly plotted in horizontal direction. This error does not occur in the DOS-version.
2) The expression `TERMS(x^2 + 5y^2 + î x)` is simplified to one term, and I hoped to get three of them.
3) More than once I get remarks from teachers that the `soLve`-function of equations should have the possibility to include/exclude complex solutions. With younger pupils, this is necessary. You can make this choice in the `Factor`-function, why not in `soLve`? More in general, why is a domain declaration of a variable not taken into account while solving an equation in that variable or plotting an expression in that variable?

DNL:

1) *DfW 4.05 plots $y^2 = 4$ as two vertical lines but DfW 4.04 recognizes $y^2 = 4$ as two vertical lines.*
   (In the meanwhile DERIVE 6 plots two horizontal lines.)

2) *In order to decomposite the expression $x^2 + y^2 + î·x$ into three terms you have to declare the variables as Complex. This is the trick*
   (This is not necessary in DERIVE 6.)

$$\#2:\quad \mathrm{TERMS}(x^2 + 5\cdot y^2 + \hat{\imath}\cdot x) = [x^2,\ 5\cdot y^2,\ \hat{\imath}\cdot x]$$

Jan Vermeylen, Kapellen, Belgium

I give you here some extracts of my correspondence with Theresa Shelby from SWHH about plotting problems with *DERIVE for Windows.*

Jan: Dear Theresa,
Allow me to mention a problem I have when plotting a vector of expressions:

Unlike it is said in the Online Help one of the expressions is plotted in the same color as the background. So it is invisible! You can check this with `VECTOR(y=k,k,-4,4,1/4)`.

Theresa: On my computer this does not happen. Some of the colors are very, very close to the background color, however. We have a deficiency in our algorithm that only rejects a color if exactly the same as the background; we need to add some kind of buffer. There could also be a difference in what color value is and what the device actually displays. Any way, we do need to work on this area of 2D plotting. Thanks for letting us know.

Jan: Can I add to the plot problem $y^2 = 9$ which I reported last week a similar one?
$(x + y - 1)^2 = 0$ is not plotted at all!

Theresa: Yes, I tried this also on DfD and only a small segment is plotted and only when the scale is 2 : 2. As you noticed it is not plotted at all in DfW.

I have more information on the last plotting problem you reported – this answer comes via David Stoutemyer …

This is a double root line. It is a parabolic valley $z = (x + y - 1)^2$ that barely touches $z = 0$, and none of our sample points happen to occur where $z = 0$. It is an unavoidable limitation of plotting contours from a finite number of samples. Try $(x + y - 1)^1$ or $(x + y - 1)^3$.

Jan: My remarks:

The $y^2 = 9$ problem is not yet solved or answered. I can understand the $(x + y - 1)^2 = 0$ problem but when teacing conic sections both problems give a ridiculous impression to the pupils.

Some days later …

Jan: While looking back at them another remarkable thing came up.

> $y = 3$ plots one horizontal line as it is supposed to do
>
> BUT
>
> $y^2 = 9$ or y^2 – 9 = 0 plot wrongly two vertical lines
>
> BUT
>
> **0 * $x + y^2 = 0$ plots correctly two horizontal lines!**

DNL: *So does DERIVE for DOS 4.*

Troels Ring, Aalborg, Denmark

Can anyone explain how to use *DERIVE* for exploring the general relationship between the correlation between two variables and the correlation between functions of two variables. The real problem is to see if the correlation between log(X) and log(Y) is a simple function of the correlation between X and Y.

Thanks in advance.

Julio Gonzales Cabillon, Urugay

We would like to inform you that the book titled

"Italian Research in Mathematics Education"

is now available on-line at our mathematics education server. It compiles the Italian papers presented at ICME8, Sevilla.

URL: http://157.253.25.2/servidor/em/recinf/libros.html

ctimath@bham.ac.uk

The Third Edition (March 1997) of the MathSkills Newsletter is now freely available in the Newsletter archive at the MathSills website:

URL: http://www.hull.ac.uk/mathskills/newsletters(index.html

(The link is still valid!)

warrens1@accucomm.net

Is there a way to create log-log graphs using *DERIVE*?

DNL: *Try this:*

t is a 2 columns-table → loglog-table of t

#1: $\text{loglog}(t) := \text{VECTOR}([\text{LN}(t_{i,1}), \text{LN}(t_{i,2})], i, \text{DIM}(t))$

f is a given function → loglog-transformation of this function

#2: $\text{loglogf}(f) := \text{SOLVE}(\lim_{y \to e^{\wedge}v} \lim_{x \to e^{\wedge}u} (\text{LN}(\text{LHS}(f)) = \text{LN}(\text{RHS}(f))), y)$

#3: $\text{tab} := \text{TABLE}(4.2 \cdot x^{1.57}, x, 50, 400, 50)$

#4: tab :=

50	1952.685031
100	5797.613911
150	$1.095752744 \cdot 10^4$
200	$1.7213389 \cdot 10^4$
250	$2.443515532 \cdot 10^4$
300	$3.253341553 \cdot 10^4$
350	$4.144156235 \cdot 10^4$
400	$5.110736343 \cdot 10^4$

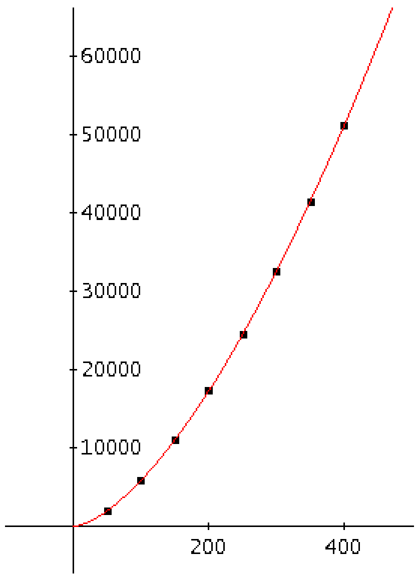
#5: loglog(tab)

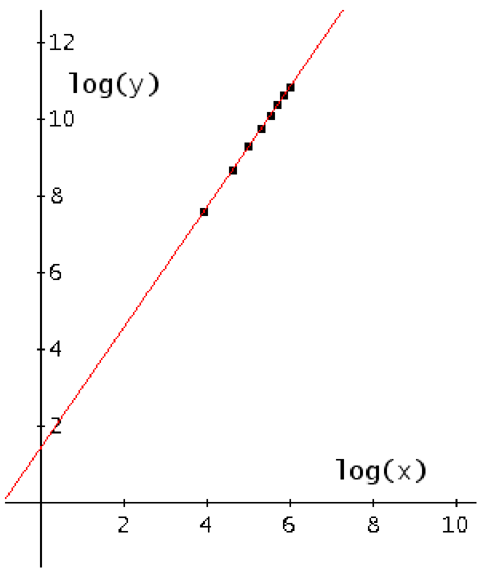
#6:

3.912023005	7.576960643
4.605170185	8.665201717
5.010635294	9.301781936
5.298317366	9.75344279
5.521460917	10.10377816
5.703782474	10.39002301
5.857933154	10.63203957
5.991464547	10.84168386

#7: $\text{loglogf}(y = 4.2 \cdot x^{1.57})$

#8: $v = 1.57 \cdot u + 1.435084525$





Scott Guth, USA

After applying some simple algebra to some trite phrases and clinches as a new understanding can be reached of the secret to wealth and success.
Here it goes.

Knowledge is Power
Time is Money and,
as every engineer knows,
Power is Work over Time.

So, substituting algebraic equations for these time-worn bits of wisdom, we get:

(1) $K = P$

(2) $T = M$

(3) $P = \frac{W}{T}$

Now, do a few simple substitutions:

Put *W*/*T* in for *P* in equation (1), which yields: (4) $K = \frac{W}{T}$

Put *M* in for *T* into equation (4), which yields: (5) $K = \frac{W}{M}$

Now we've got something. Expanding back into English, we get:

Knowledge equals Work over Money

What this MEANS is that:

1. The More you Know, the More Work You Do, and
2. The More you Know, the Less Money You Make.

Solving for Money, we get: (6) $M = \frac{W}{K}$

Money equals Work over Knowledge

From equation (6) we see that Money approaches infinity as Knowledge approaches 0, regardless of the Work done. What this MEANS is:

The More you Make, the Less you know.

Solving for Work, we get: (7) $W = M \cdot K$

Work equals Money time Knowledge

From equation (7) we see that Work approaches 0 as Knowledge approaches 0. What this MEANS is:

The stupid rich do little or no work

Working out the socioeconomic implications of this breakthrough is left as an exercise for the reader, Tosh

Should Computer Algebra Programs Use ln *x* or ln |*x*| as their Antiderivative of 1/x?

David R. Stoutemyer, SWHH, Hawaii, USA

The *DERIVE*™ computer algebra system currently uses ln *x* as its antiderivative of 1/x. About once a month we receive a suggestion that we should use ln |*x*| instead. Many current calculus texts alternate between these alternatives with no explanation of why. This article is intended to provoke sufficient debate so that implementors can decide what to do about this.

1. Logarithms in Computer Algebra Systems

For reasons of practicality, most hardware and software regards an expression such as ln(*z*) as denoting only one branch of the infinitely-branched logarithm function. With a computer algebra system, users can always specify an infinitely-branched logarithm by entering expressions such as $\ln(z) + 2n\cdot\pi\ I$, where *n* denots an indeterminate integer. After dealing with the consequent clutter they will probably decide that this is not the panacea they hoped.

If ln denotes one branch, correct use is more likely if ln consistently denotes a particular branch, such as the principal branch usually defined by

$$\mathrm{Re}(\ln z) = \ln |z| \text{ and } \mathrm{Im}(\ln z) = \mathrm{phase}(z), \text{ where } -\pi < \mathrm{phase}(z) \le \pi.$$

Among other prohibitions, consistency to this principal branch entails resisiting the temptation to "simplify" ln(exp *w*) to *w* unless the algebra system can deduce that
$-\pi < \mathrm{Im}(w) \le \pi$.

2. Why ln *z* is the better choice for complex *z*

Most computer algebra programs were not written solely to meet the needs of beginning real-variable calculus. It is a desirable goal to have integration give correct results over contours in the complex plane as well as over real intervals throughout which the integrand is well behaved. The multiple-branched logarithm is analytic except at *z* = 0, and the principal branch is applicable to contours that also don't cross the negative *x*-axis without returning.

Consequently, integration of 1/*x* from a point such as -3 to point such as +2 would yield $\ln 2 - (\ln 3 + i\cdot\pi) = \ln(2/3) - i\cdot\pi$. This is consistent with taking a contour from *x* = -3 to *x* = -*r* with *r* > 0, proceeding along a semicircle of radius *r* from angle π to angle 0, then proceeding along the positive *x*-axis to *x* = 2. There is no need to take the limit as *r* approaches 0.

In contrast , $\frac{d}{dz}\ln|z| = \frac{\mathrm{sign}(z)}{|z|} = \frac{1}{\mathrm{conj}(z)}$ where $z = \mathrm{sign}(z)\cdot|z|$.

z = conj(*z*) only for real *z*, so ln|*z*| is generally wrong for complex *z*.

3. Why ln *x* is correct even for negative real intervals

Integrating a continuous real function over a real interval should give a real result. However, this does not require that an antiderivative used to determine such an integral must be real: Valid antiderivatives can differ from each other by an arbitrary constant, and that constant can be complex, such as $i\cdot\pi$. Any such constants cancel out in a definite integral, so using ln *x* gives the desired real result when integrating between two negative values.

4. Why ln |*x*| is wrong even for real *x*

Here is an excerpt from "A First Course in Calculus" by Serge Lang (Addison-Wesley, pp. 136-137):

> "We agree throughout that indefinite integrals are defined only over intervals. Thus in considering the function $1/x$, we have to consider separately the cases $x > 0$ and $x < 0$. For x > 0, we have already remarked that log *x* is an indefinite integral. It turns out that for the interval $x < 0$ we can also find an indefinite integral, and in fact we have for $x > 0$,
>
> $$\int \frac{1}{x}\,dx = \log(-x)\,.$$
>
> Observe that when $x < 0$, $-x$ is positive, and thus $\log(-x)$ is meaningful. The fact that the derivative of $\log(-x)$ is equal to $1/x$ is true by the chain rule.
>
> For $x < 0$, any other indefinite integral is given by
>
> $$\log(-x) + C,$$
>
> where *C* is a constant.
>
> It is sometimes stated that in all cases,
>
> $$\int \frac{1}{x}\,dx = \log|x| + C\,.$$
>
> With our conventions, we do not attribute any meaning to this, because our functions are not defined over intervals (the missing point 0 prevents this). In any case, the formula would be false. Indeed, for $x < 0$ we have
>
> $$\int \frac{1}{x}\,dx = \log|x| + C_1$$
>
> and for $x > 0$ we have
>
> $$\int \frac{1}{x}\,dx = \log|x| + C_2$$
>
> in all cases.

We prefer to stick to our convention that integrals are defined only over intervals. When we deal with the log, it is to be understood that we deal only with the case $x > 0$."

Note also that the integral of $1/x$ diverges for contours that go through $x = 0$. If users are thinking of such a path, then the complex result of using ln x will probably alert them to their misuse whereas the incorrect real result of using ln $|x|$ is more likely to go unnoticed.

It is true that using ln $|x|$ fortuitously gives the correct Cauchy principal value for all real intervals. However, Cauchy principal values are customarily and wisely not taught in elementary calculus. Their valid use generally requires supporting physical justification, such as the fact that the singularities are known to be artificial canceling artifacts of the particular mathematical model of a physical problem.

5. Why some people might nonetheless prefer ln |*x*|:

Probably the reason some prefer ln $|x|$ is aesthetic: For reasons of parsimony we wish that real-variable calculus could be done entirely without excursions into complex variables. I doubt that this is possible for sufficiently complicated examples. Moreover, even for the simple example of integrating $1/x$, using ln $|x|$ to avoid complex variables unnecessarily teaches bad habits.

Calculus students are introduced to complex numbers in high-school algebra, and they are taught that antiderivatives can differ by arbitrary constants. Why not exploit logarithms of negative numbers to reinforce both ideas? This is not to say that we should generalize all elementary calculus to complex variables -- only that if a complex number is most correct and natural in a few places, exploit it rather than fight it.

It is true that the $i \cdot \pi$ from ln x does not disappear when instead of evaluating a definite integral on a negative interval we are merely substituting a single negative number into this antiderivative. However, since antiderivatives can differ by a constant, it is hard to imagine a good reason for only substituting one number into an antiderivative -- one might as well pick a random number.

On the other hand, there are good reasons to plot an antiderivative as its variable varies over some interval, and most 2-dimensional plot packages intended for real values conservatively plot nothing wherever a value is unreal. However, it is always possible to plot both the real and imaginary parts of an antiderivative such as ln x. Indeed, doing so has the educational value of helping to reveal the phase discontinuity across $x = 0$.

6. Why ln |*x*| is abhorred by computer-algebra implementors even for *x* positive or for *x* negative:

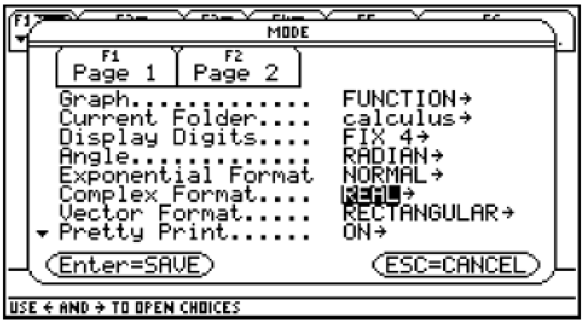
It is hard to automate powerful correct automatic simplification of absolute values. Once an expression is contaminated by them, they tend to persist unnecessarily in most

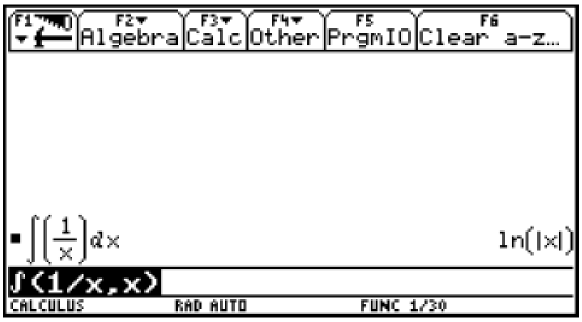
subsequent derived expressions. For example, many computer-algebra systems don't simplify the real derivative of ln |*x*| all the way to 1/*x* for real *x*. Absolute values also tend to thwart operations such as equation solving or further integration. For example, not many systems can automatically solve 2|3*x* – b| > 5 or compute antiderivatives for 1/|*x*| or
ATAN |a *x* + b|.

7. Conclusions

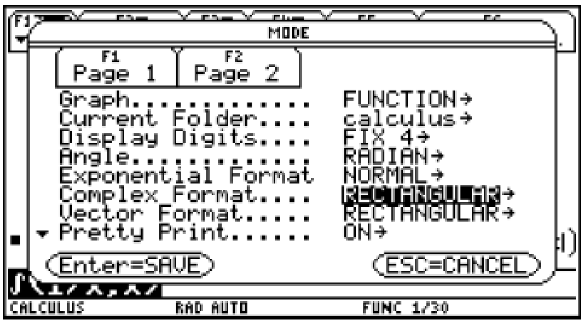
Despite the above arguments, for those users who insist on including an absolute value we are considering having a future version of *DERIVE* include a switch: When ln |*x*| is selected, *DERIVE* would use ln |*u*| in antiderivatives whenever DERIVE can deduce that *u* is real. Otherwise DERIVE would use ln *x*. Should we include such a switch? If so, what should be the default setting?

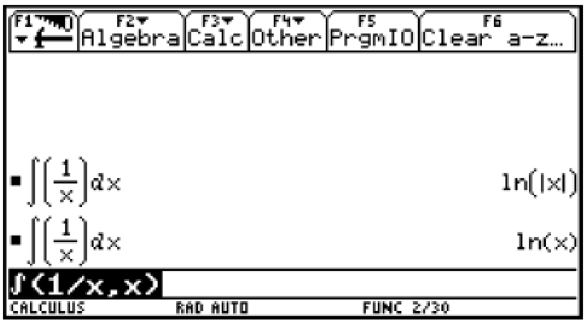
As you can see on the following screen shots, the TI-92 offers the "switch" David spoke about in the last paragraph.





Compare the settings for Complex Format!





Josef

Request for Submissions
Third International DERIVE and TI-92 Conference
July 14-17 on the Campus of Gettysburg College
Gettysburg, Pennsylvania

Submissions are requested for the Third International *DERIVE* and *TI*-92 Conference to be hedl July 14-17 on the campus of Gettysburg College in Gettysburg, Pennsylvania, USA. Submissions may be in form of proposals for 25 or 50 minute presented papers, or proposals for 2 hour workshops. All proposals will be refereed by the Conference Committee who will select from among the submissions. Acceptance of submissions will be based on the reports of the referees and the availability of program slots. In past conferences there has been an approximately 70% acceptance rate of proposals.

Topics for presented papers are expected to include discussions of at least one of the following topics: the use of *DERIVE* and/or the *TI*-92 in the classroom with an analysis of its use; applications of *DERIVE* and/or the *TI*-92 to topics in school and university level mathematics; discussions of the educational issues involved in using a Computer Algebra System (CAS) in the teaching of mathematics; the effects of using a CAS on the mathematics curriculum; or suggestions for the future design of a CAS for use in the teaching of mathematics. Other topics for papers may be considered if they are deemed appropriately related to the themes listed above by the Conference Committee.

Workshops may be either introductory tutorial, advanced tutorial, or on the use of *DERIVE* and/or the *TI*-92 in the teaching of a subject or group of subjects found in the mathematics curriculum. Preference will be given to workshops using *DERIVE* for WINDOWS as well as those using the *TI*-92. A particular preference will be given to workshops using the CBL or CBR in conjunction with the *TI*-92. Other platforms will be considered depending upon the availability of microcomputer laboratory and seminar room space. Attendance at a workshop will be based on participants´ choices at registration and be limited to 25 in addition to the instructor.

Format for Proposals and Deadline

Proposals for papers should contain the title of the paper, the names and associations of the authors, the name and preferred address of the contact person for the paper, the length of the presentation (25 min or 50 min) and an extended abstract of two pages double spaced with one standard margin.

Proposals for workshops should contain the title of the workshop, the name and associations of the presenters, the name and preferred address of the contact person, the intended audience for the workshop, the level (introductory, advanced or subject oriented) of the workshop, a brief abstract, and a detailed outline of the workshop (2 pages total.)

Four copies, collated and stapled, should be mailed to

Professor Carl Leinbach
Department of Mathematics & Computer Science
Box 402
Gettysburg College
Gettysburg, PA 17325
USA

Please mark outside of your envelope: International DERIVE & TI-92 Conference. Papers must be received in Gettysburg by November 15, 1997. Notification of acceptance will be made during January of 1998.

Keynote Speakers

During the morning of three days of the conference there will be an invited keynote address for all participants in the conference. No activities will be scheduled during the keynote address. The following individuals have been invited and tentatively accepted our invitation to be keynote speakers.

Josef Böhm	Austria
Wade Ellis	USA
Adrian Oldknow	UK

The Committee will also invite David Stoutemyer of the Soft Warehouse to make a special presentation to the conference.

Conference Committee (Tentative)

The conference organizers are

Carl Leinbach
Gettysburg College
Department of Mathematics & Computer Science
Gettysburg, PA 17325

e-mail: leinbach@cs.gettysburg.edu

Bert K. Waits
The Ohio State University
Mathematics Department
231 W.
18th Ave Columbus, OH 43210
waitsb@math.ohio-state.edu

all correspondence regarding the Third International *DERIVE* and *TI*-92 Conference should be addressed to either of above. Proposals are to be submitted to Carl Leinbach by mail or e-mail.

The following people have been invited to serve as members of the Conference Committee

Bärbel Barzel,Germany	Terence Etchells,UK
John Berry,UK	Bernhard Kutzler,Austria
Josef Böhm, Austria	Josef Lechner,Austria

Marvin Brubaker, USA	John Monaghan, UK
Paul Drijvers, Netherlands	Jeanette Palmiter,USA
Robert Hill, USA	Barbara Vincent, USA

Committee members will assist the organizers with the refereeing of papers and the general organization of the conference.

History of the Conference

The first International DERIVE Conference was held during the summer of 1994 in Plymouth, England and was organized by Professor John Berry of the University of Plymouth. The con-ference hosted over 200 participants and set the format for future conferences.

The Second International DERIVE Conference was held during the summer of 1996 at Schloss Birlinghoven in Bonn, Germany. This conference was organized by Bärbel Barzel. This conference hosted more than 250 participants and featured several presentations using the TI-92 as well as DERIVE.

For the Third International DERIVE and TI-92 Conference the TI-92 is given full partnership in the title of the Conference. It is hoped that teachers who are using hand held computer algebra systems will make several contributions to the conference in the form of presentations and workshops.

Registration Fee

The registration fee for the conference ist expected to be $150.00 US for participants, $100 for accompanying adults (includes allied program, battlefield tour, and conference trip) and $50 for children under 12 (also includes allied program, battlefield tour, and conference trip.) This fee will be confirmed when the formal registration forms are distributed. Requests for these forms may be made by contacting Carl Leinbach at his e-mail address of

"leinbach@cs.gettysburg.edu"

or by mail to him at the address given above for the submission of proposals.

Housing

Because Gettysburg is the site of an important battle of the USA Civil War (1860-1865), it is a very popular tourist attraction. Enclosed with this call is a listing of hotels/motel accomodations in the Gettysburg Area. Those marked with a + are within two blocks walking distance of Gettysburg College (the host institution for the Conference). Other motels are an easy drive from the College or a 20 - 30 minute walk. The cost of motel rooms varies, but usually runs from between $80.00 and $100.00 per day. It should be noted that Gettysburg has several restaurants ranging from fast food to fine dining.

Rooms have been reserved at the Historic Best Western - Gettysburg Hotel for the conference participants under the name of the International *DERIVE* and TI-92 Conference. Participants wishing to stay at our "head quarters" hotel need to make reservations for themselves, but must refer to the Conference.

Also a block of dormitory rooms for 160 participants (single or double occupancy) has been reserved at the College for participants. These rooms include linens (no maid service) and a full meal plan for approximately $45.00 per day per participant. The price is the same for both the participant and accompanying family members. Individuals staying in the dormitories may check in on July 13 and must check out by noon on July 17. It is recommended that a family of size larger than 2 consider accommodations other than those of the College. An individual meal plan including three meals a day will be available approximately $22.00 per day per person for participants not staying in dormitory rooms. Meals at the College Cafeteria are cafeteria style and allow for free refills.

Participants are urged to make their room reservations early. Motels do tend to fill up well in advance of the desired dates.

Spouse & Family Parallel Programm

Arrangements have been made for special programs for spouses, family members, and others attending the conference with participants. These programs will be on a daily basis and will include tours of historic and cultural attractions in the area. In addition to its connection to the USA Civil War, Gettysburg is located in one of the primary fruit production areas of the United States. It also borders the famous Pennsylvania Dutch (a crass Americanization of Deutsche) Country. Tours will take advantage of Gettysburg´s unique location and include a lunch or box lunch.

As is the custom at *DERIVE* and TI-92 Conferences, one part of a day at the conference will be devoted to a special conference activity. At the Third International *DERIVE* and TI-92 Conference, this will include several tours with dinner. Options to tours include: Washington, DC; the Inner Harbor of Baltimore, MD; tours of the Pennsylvania Dutch and Amish Country; Hershey Park Amusement Park; and Longwood Gardens.

In addition, on Friday afternoon, following the adjournment of the Conference there will be bus tours of the Gettysburg Battlefield.

Sponsorship

Sponsors of the Third International *DERIVE* and TI-92 Conference include Gettysburg College, the Soft Warehouse, and Texas Instruments Corporation.

3D-Grafik mit Adjazenzlisten
3D-Graphics Using Adjacens Lists

Hartmut Kümmel, Biedenkopf, Germany

1. Einführung

Zur Darstellung linearer Abbildungen durch Matrizen kann man die Matrix von links oder von rechts mit dem abzubildenden Vektor multiplizieren. Durch Transponieren kann eine dieser beiden Schreibweisen in die andere übersetzt werden – beide sind im Bereich der Computergrafik üblich. Im Folgenden habe ich mich für die zur gewöhnlichen Schreibweise bei Gleichungssystemen passende Darstellung $\vec{y} = A \cdot \vec{x}$ entschieden. Dabei kann das Bild einer Figur durch eine einzige Matrixmultiplikation berechnet werden, wenn die Punkte als Spalten einer „Figur-Matrix" aufgefasst werden.

Die Plot-Funktion von *DERIVE* erwartet bei Punktlisten die der Reihe nach zu zeichnenden Punkte in Zeilen einer „Figur-Matrix". Diese *DERIVE*-Syntax ist nicht „kompatibel" zur oben bevorzugten Schreibweise für Abbildungen und Gleichungssysteme, bei der die Matrix von links multipliziert wird und Punkte als Spalten aufzufassen sind. Zur Darstellung komplexerer Figuren erscheint mir die „Punkt-Reihung" ohnehin nicht so gut geeignet, weil manche Kanten mehrmals „durchlaufen" werden müssen.

1. Introduction

Linear mappings can be performed by a multiplication of the transformation matrix with the vector to be transformed from the left- or right hand side. I chose the form y = A . x - according to the presentation of simultaneous linear equations. Then the points are interpreted as columns in a "figure-matrix". We have to consider that DERIVE´s PLOT-function expects the points to be plotted as rows in a matrix. From my point of view this way does not seem not to be comfortable for more complex figures, because in most cases some edges have to be run through several times.

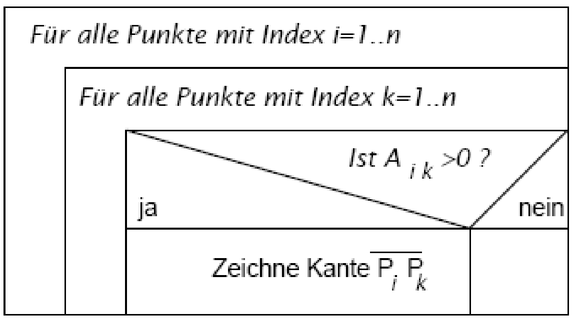
2. Repräsentation von Kantenmodellen als Graphen

Ein Graph besteht aus Knoten und Kanten. Zur Repräsentation eines Graphen ist anzugeben, welche Knoten mit einer Kante verbunden sind. Zur geometrischen (grafischen) Darstellung wird zusätzlich die Information über die Orte der Knoten benötigt. In der Graphentheorie sind hierfür (mindestens) zwei Lösungen gebräuchlich: **Adjazenz-Matrizen** und **Adjazenz-Listen**.

Das Matrixelement einer **Adjazenzmatrix** gibt an, ob die beiden Knoten mit dem Index *i* und *k* zu verbinden sind:

$$A_{ik} = \begin{cases} 1 & \text{falls } P_i \text{ mit } P_k \text{ verbunden} \\ 0 & \text{sonst} \end{cases}$$

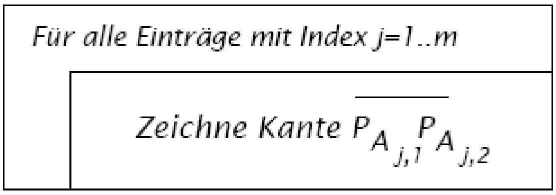
Die Darstellung des Graphen erfolgt nach dem im Struktogramm angegebenen Verfahren.



Eine **Adjazenzliste** enthält für jede Kante ein Paar der Indizes der zu verbindenden Knoten. So eine Liste kann beispielsweise durch eine zweispaltige Matrix realisiert sein:

$$\text{Zeile } A_j = \begin{cases} A_{j,1} & \text{Index des 1. Knotens} \\ A_{j,2} & \text{Index des 2. Knotens} \end{cases}$$

Die Darstellung des Graphen erfolgt nach dem im Struktogramm angegebenen Verfahren.



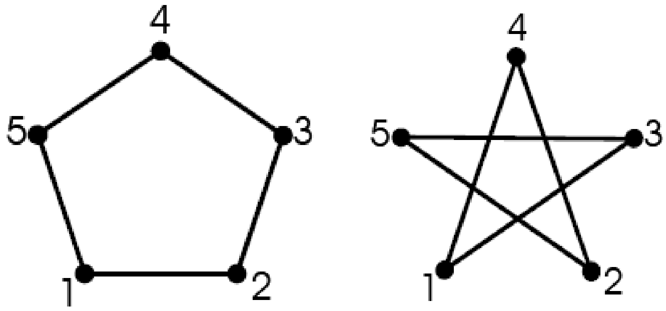
Die Darstellung als Adjazenzmatrix hat Vorteile bei der Suche nach Wegen im Graphen. Bei ungerichteten Graphen steht jedoch zuviel an Information zur Verfügung - die „halbe“ Matrix würde genügen. Für die Darstellung von Kantenmodellen eines Körpers mit den Mitteln von *DERIVE* erscheinen deshalb Adjazenzlisten als geeignete Struktur.

2. Presentation of edge-models as graphs

Usually a graph consists of nodes and edges. It must be known which nodes are connected by an edge. This information can be expressed by an adjacens matrix or by an adjacens list. The element of the matrix A_{ik} = 1, if point P_i is connected with P_k, otherwise A_{ik} = 0. The list consists of index-pairs, which describe all the edges by pairs of numbers, each of them belonging to one point.

3. Beispiel für die Darstellung eines Kantenmodells

Für ein Pentagon und ein Pentagramm ist jeweils die Adjazenzliste angegeben: Alle Adjazenz-Angaben beziehen sich nur auf die *Indizes* der Knoten. Zur grafischen Darstellung sind zusätzlich (z.B. in einer Matrix unter diesen Indizes) die *Koordinaten* der entsprechenden Punkte anzugeben.




```
penta := VECTOR([COS(Φ),SIN(Φ)],Φ,0°,360°,72°)'
```

(Die Punktliste liegt in Form einer $n \times 2$ Liste vor, *x*- und *y*-Koordinaten sind in Zeilen angeordnet.)

Pentagon	Pentagramm
Adjazenzliste: $\{(1,2),(2,3),(3,4),(4,5),(5,1)\}$	Adjazenzliste: $\{(1,3),(3,5),(5,2),(2,4),(4,1)\}$
bzw. in Matrixschreibweise: $\begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \\ 5 & 1 \end{pmatrix}$	bzw. in Matrixschreibweise: $\begin{pmatrix} 1 & 3 \\ 3 & 5 \\ 5 & 2 \\ 2 & 4 \\ 4 & 1 \end{pmatrix}$

3. An example for presentation of an edge-model and

4. How to do it in DERIVE

You can find two lists - one producing a pentagon, the other one will lead us to a pentagram, both using the same five points penta, resulting from

```
penta := VECTOR([COS(Φ),SIN(Φ)],Φ,0°,360°,72°)
```

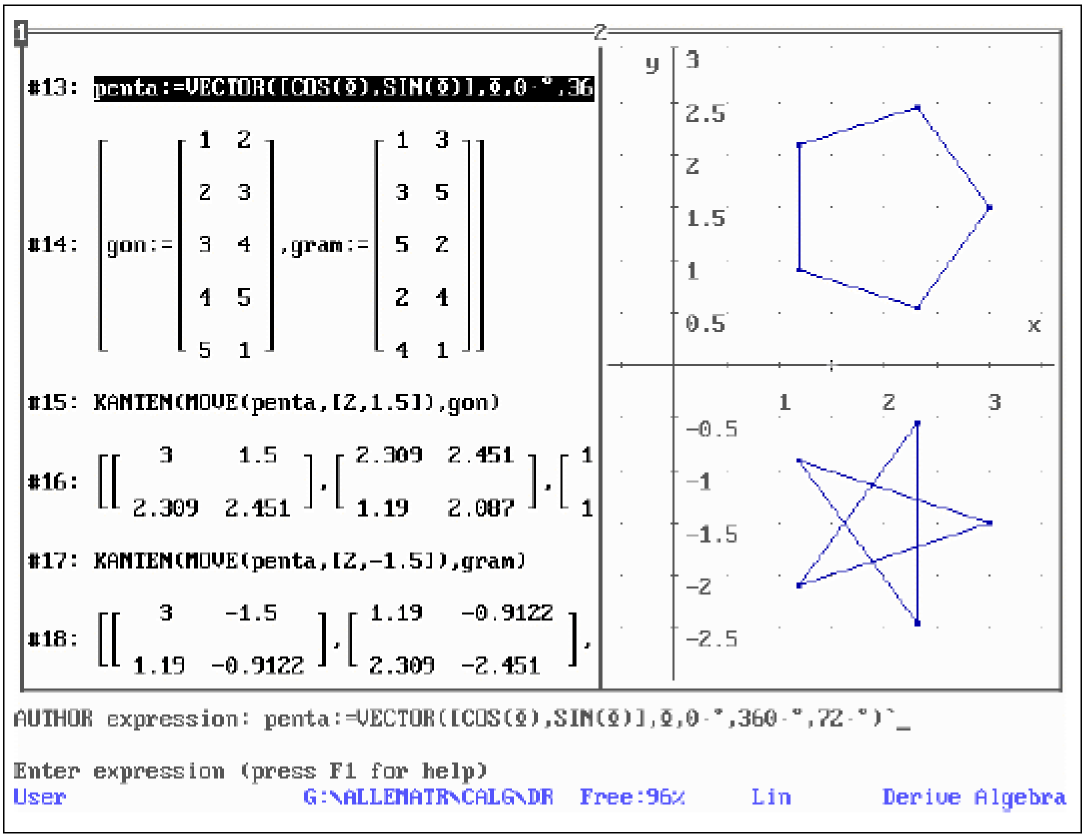
(*The point list is given in form of a n* $\times$ *2 matrix with x- and y-coordinates in two rows.*)

The two lists - matrices with 2 columns - `gon` *and* `gram` *are describing how to connect the points given in the list of points called* `penta`*. The function* `MOVE(...)` *helps to translate the figure and can be found in the DERIVE-file TOOL3D below.*

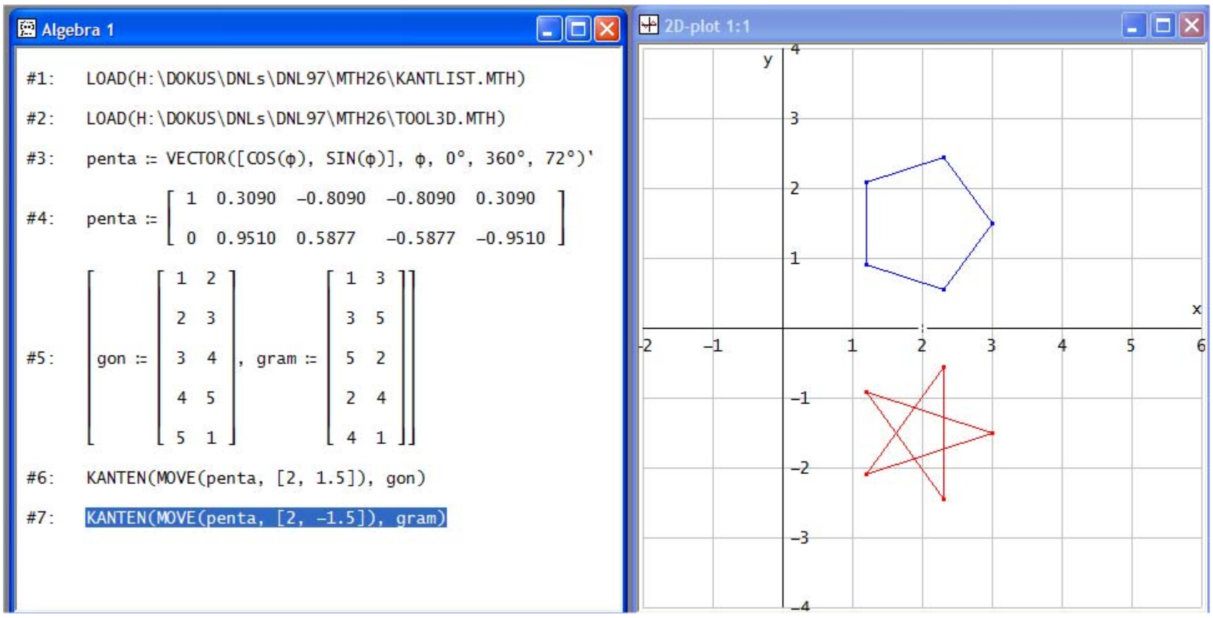
4. Umsetzen der Adjazenzlisten-Repräsentation

Die folgend definierte *DERIVE*-Funktion `KANTEN` erlaubt es, die Kanten einer Figur mit Hilfe einer Adjazenzliste zu plotten.

Dabei wird auch das hier benötigte Transponieren der spaltenweise angegebenen Punktliste in die von *DERIVE* erwartete zeilenweise Auflistung übernommen: Die Funktion `MOVE` ist weiter unten angegeben!



This is the *DERIVE* screen from 1997. Next page shows the *DERIVE 6* screen with KANTLIST.MTH and TOOL3D.MTH preloaded as utility files.



KANTLIST.MTH

```
#1:  [PrecisionDigits := 12, Precision := Approximate]

#2:  [NotationDigits := 4, Notation := Decimal]

#3:  --- KANTEN-Tool zur Darstellung von Figuren mit Adjazenzlisten ---

#4:  Eckpunkte stehen in SPALTEN von fig, M' ist die Transponierte von M

#5:  Vertices are given in columns of fig, M' is transposed of M

#6:  KANTE(fig, adj, i) := [fig'          , fig'          ]
                                adj↓i↓1        adj↓i↓2

#7:  KANTEN(fig, adj) := VECTOR(KANTE(fig, adj, i), i, 1, DIM(adj))

#8:  --- Nun Figuren (Punkte spaltenweise) und Adjazenzlisten eingeben ---

#9:  *** Enter figure (points in columns) and adjacens-list ***
```

5. Zur Verständlichkeit des KANTEN-Tools im MU

Die Funktion KANTE verwendet das Konzept der indirekten Adressierung. Das ist eine elegante, für die Informatik typische Denkweise, zugleich aber auch eine Anwendung von Matrizen. Wie aufwändig ist es, die Arbeitsweise des KANTEN-Tools durchschaubar zu machen? Im Unterricht würde ich die Auswertung des Ergebnis an einem Beispiel-Aufruf durchspielen:

Bestimme die durch KANTEN(quad,umf) *berechnete Figur* (#8):

#8: $\left[\text{quad} := \begin{bmatrix} 7 & 10 & 10 & 7 \\ 1 & 1 & 4 & 4 \end{bmatrix}, \text{umf} := \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 2 & 3 \\ 3 & 4 \\ 4 & 1 \end{bmatrix} \right]$

Die schrittweise Auswertung von KANTE(quad,umf,2) ergibt:

#9: [quad :=, umf :=]

#10: KANTE(quad, umf, 2) = $\left[quad'_{umf\downarrow 2\downarrow 1}, quad'_{umf\downarrow 2\downarrow 2}\right]$

#11: umf := $\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 2 & 3 \\ 3 & 4 \\ 4 & 1 \end{bmatrix}$

#12: KANTE(quad, umf, 2) = $\left[quad'_{2}, quad'_{4}\right]$

#13: quad := $\begin{bmatrix} 7 & 10 & 10 & 7 \\ 1 & 1 & 4 & 4 \end{bmatrix}$

#14: KANTE(quad, umf, 2) = $\begin{bmatrix} 10 & 1 \\ 7 & 4 \end{bmatrix}$

1. Auswertung: Ausdruck #12 (quad noch nicht definiert!)

2. Auswertung: Ausdruck #14

Nun folgt die Konstruktion einer Liste solcher Punktepaare:

#15: [quad :=, umf :=]

#16: KANTEN(quad, umf) = VECTOR(KANTE(quad, umf, i), i, 1, DIM(umf))

#17: umf := $\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 2 & 3 \\ 3 & 4 \\ 4 & 1 \end{bmatrix}$

#18: KANTEN(quad, umf) = $\begin{bmatrix} quad'_{1} & quad'_{2} \\ quad'_{2} & quad'_{4} \\ quad'_{2} & quad'_{3} \\ quad'_{3} & quad'_{4} \\ quad'_{4} & quad'_{1} \end{bmatrix}$

#19: $\text{quad} := \begin{bmatrix} 7 & 10 & 10 & 7 \\ 1 & 1 & 4 & 4 \end{bmatrix}$

#20: $\text{KANTEN(quad, umf)} = \left[\begin{bmatrix} 7 & 1 \\ 10 & 1 \end{bmatrix}, \begin{bmatrix} 10 & 1 \\ 7 & 4 \end{bmatrix}, \begin{bmatrix} 10 & 1 \\ 10 & 4 \end{bmatrix}, \begin{bmatrix} 10 & 4 \\ 7 & 4 \end{bmatrix}, \begin{bmatrix} 7 & 4 \\ 7 & 1 \end{bmatrix}\right]$

Hier soll nur die Arbeitsweise einer vorgelegten Funktion nachvollzogen – und verstanden – werden. Die Fähigkeit, solche Funktionen in weniger vertrautem Zusammenhang zu programmieren, ist ein völlig anderes Lernziel!

5. How to explain the tool in Math Teaching

The function KANTE *uses the concept of indirect addressing. This is an elegant way of thinking which is typical in information technology, and it is an application for matrices. I believe that it might make sense to make this tool understandable even in standard mathematics teaching. Expression #8 from above describes a special figure. How does it look like?*

I would demonstrate the concept for stepwise evaluating one call of KANTE*. In this case I want the students to understand how a ready made function is working. Developing such functions is quite a different learning goal.*

6. Räumliche Abbildungen

Als Projektion zur Darstellung von Schrägbildern wurde die senkrechte Projektion auf die 1-2-Ebene festgelegt, um nicht in der Vielfalt der Möglichkeiten unterzugehen. Die Matrizen für diese Projektion und für die Drehungen um die Koordinatenachsen sind Standardbegriffe. Zum Erzeugen der gewünschten Ansichten muss die Urfigur passend abgebildet werden!

Die Verschiebung wird hier als spezielle, nicht-lineare Abbildung eingeführt – etwas umständlich wegen der Darstellung der Punkte in Spalten der Figur-Matrix. Man könnte sie auch unter Verwendung homogener Koordinaten in die Standarddenkweisen „einbetten".

```
#1:   ------- TOOL3D: Hilfsmittel für 3D-Grafik --------

#2:   [Precision := Approximate, PrecisionDigits := 12]

#3:   [NotationDigits := 4, Notation := Decimal]
```

#4: $\left[\text{DREH1}(\alpha) := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \text{COS}(\alpha) & -\text{SIN}(\alpha) \\ 0 & \text{SIN}(\alpha) & \text{COS}(\alpha) \end{bmatrix}, \text{DREH2}(\alpha) := \begin{bmatrix} \text{COS}(\alpha) & 0 & \text{SIN}(\alpha) \\ 0 & 1 & 0 \\ -\text{SIN}(\alpha) & 0 & \text{COS}(\alpha) \end{bmatrix}\right]$

#5: $\left[\text{DREH3}(\alpha) := \begin{bmatrix} \text{COS}(\alpha) & -\text{SIN}(\alpha) & 0 \\ \text{SIN}(\alpha) & \text{COS}(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{proj12} := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}\right]$

#6: $\text{MOVE(f, v)} := \text{VECTOR(VECTOR(}f_{z,i} + v_z\text{, i, 1, DIM(}f_1\text{)), z, 1, DIM(f))}$

Die Darstellung des Einheitswürfels zeigt den Übergang zur dritten Dimension.

Das KANTEN-Tool bleibt unverändert, da die Adjazenz-Information auch für die projizierten Figuren gilt!

```
#1:    LOAD(H:\DOKUS\DNLs\DNL97\MTH26\KANTLIST.MTH)

#2:    LOAD(H:\DOKUS\DNLs\DNL97\MTH26\TOOL3D.MTH)
```

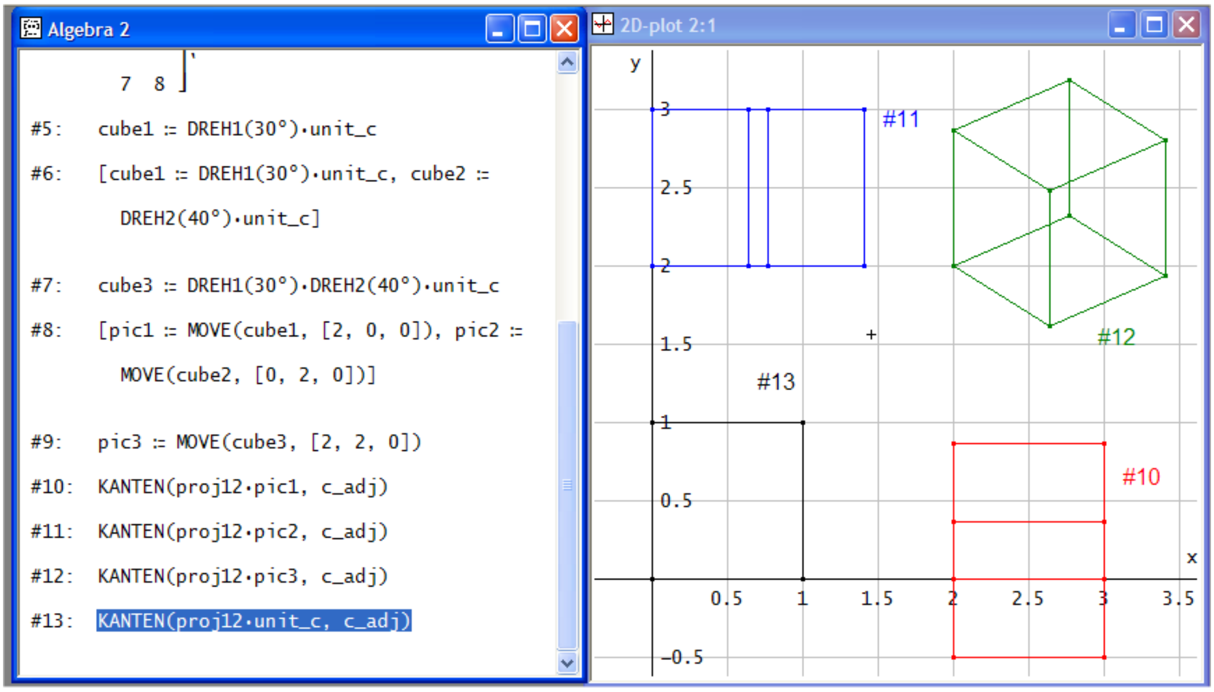
#3: unit_c :=
$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

#4: c_adj :=
$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 & 6 & 7 & 8 & 5 & 5 & 6 & 7 & 8 \end{bmatrix}'$$

6. Mappings of 3D objects

In order to produce oblique views I chose the normal projection onto the 1-2 plane. The matrices for this projection and for the rotations about the coordinate axes are standard. Translation is introduced as a special non-linear mapping. You might prefer using homogeneous coordinates.

The presentation of the unit cube shows the change to the 3rd dimension. The KANTEN-tool will remain the same because the adjacens list is also valid for projected figures.



7. Ein Abschluss-„Film“ ...

Die Herstellung des Films als Liste von Einzelbildern zeigt die typische, daten-orientierte („objektoriente“) Arbeitsweise im funktionalen Stil:

Als Grundobjekt wird ein „Turm“ durch seine Punktliste `tower` und die Adjazenzliste `tadj` definiert.

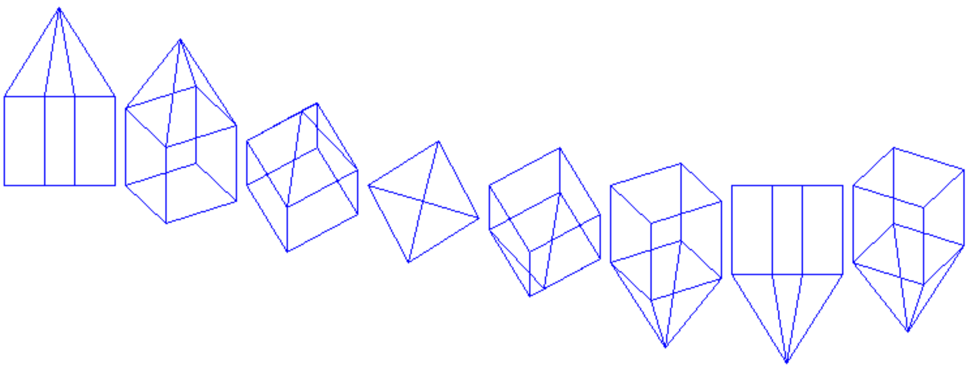
Durch Abbildungen wird zuerst jedes „i-te“ Einzelbild und dann daraus der Film als Liste der Einzelbilder hergestellt.

7. Finally a "Movie"....

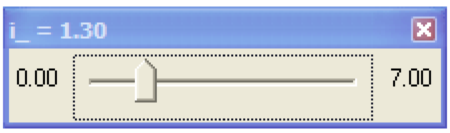
The "tower" is defined by its point list `tower` *and its adjacens list* `tadj`*. The pictures are the result of a rotation combined with a translation. The movie consists of a list of single pictures.*

(`DREH1(α)`, … *are identical to* `ROTATE_X(α)`,… *from GRAPHICS.MTH.*)

```
#1:   LOAD(H:\DOKUS\DNLs\DNL97\MTH26\KANTLIST.MTH)

#2:   LOAD(H:\DOKUS\DNLs\DNL97\MTH26\TOOL3D.MTH)

                ┌ 0  1  1  0  0  1  1  0  0.5 ┐
#3:   tower :=  │ 0  0  1  1  0  0  1  1   2  │
                └ 0  0  0  0  1  1  1  1  0.5 ┘

#4:   tadj :=  ┌ 1  2  3  4  5  6  7  8  1  2  3  4  3  4  7  8 ┐'
               └ 2  3  4  1  6  7  8  5  5  6  7  8  9  9  9  9 ┘

#5:   tow(α) := DREH1(α)·DREH2(30·1°)·tower

#6:   tower_pic(i) := MOVE(tow(15·i·1°), [0.75·i, 2, 0])

#7:   VECTOR(KANTEN(proj12·tower_pic(2·i), tadj), i, 0, 7)
```



In *DERIVE* 6 you can really have a movie using a slider bar!

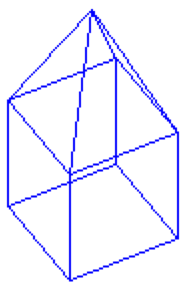



```
KANTEN(proj12·tower_pic(2·i_), tadj)
```

Introduce a slider for i_ with $0 \le i\_ \le 7$ and 70 intervalls.

Then drag the slider and observe the tower moving smoothly across the screen.

Josef



8. Adjazenzlisten für orientierte Flächen

Abschließend möchte ich zeigen, dass der von Hubert Weller im nächsten DNL erklärte Sichtbarkeitstest mit orientierten Flächen „nahtlos“ in das Adjazenzlistenkonzept eingebunden werden kann.

Zur Darstellung eines Körpers müssen zunächst die Punkte spaltenweise eingegeben werden. Als Adjanzenzlisten-Einträge für jede Fläche sind die Indizes der beteiligten Punkte so einzugeben, dass die Reihenfolge einen Linksumlauf beschreibt, wenn die Blickrichtung auf das Zentrum des Körpers zugrunde gelegt wird. Aus diesen Daten erzeugt die Funktion `flas` eine Liste von Flächenlisten zur Darstellung durch die Plot-Funktion, aus der die sichtbaren Objekte mit `show` und die unsichtbaren mit `unshow` ausgewählt werden können.

8. Adjacens lists for oriented surfaces

At the end of this article I'd like to demonstrate that Hubert Weller's visibility test (see next DNL) can be implemented into the concept of adjacens lists.

For presenting a figure one has to enter the points as columns. The entries for the adjacens lists of the single surfaces are the numbers of the points involved together with the sequence of the indeces of the points in an order describing a left hand circulation when viewing at the center of the body. Function `flas` *creates a list for plotting.* `show` *is collecting the visible objects while* `unshow` *is collecting the invisible ones.*

Utility file `fladlist.mth`

```
#1:   Verdeckte Kanten - Hidden Lines

#2:   [Precision := Approximate, Notation := Decimal]

#3:   1. Matrices for linear mappings in R3

             ┌ 1  0  0 ┐
#4:   id :=  │ 0  1  0 │
             └ 0  0  1 ┘

#5:   Translation of object f by vector v

      move(f, v) := VECTOR(VECTOR(f    + v , i, DIM(f )), z, DIM(f))
#6:                                z,i    z           1

#7:   Similar to isometric projection

#8:   iso := ROTATE_X(30°)·ROTATE_Y(35°)

#9:   Vertical projection = top view

                 ┌ 1  0  0 ┐
#10:  proj12 :=  └ 0  1  0 ┘

#11:  2. Special vector operations: project and transpose

      pr12(lst) := VECTOR(proj12·lst , k, DIM(lst))
#12:                                k
```

```
       transp(lst) := VECTOR(lst ', i, DIM(lst))
#13:                             i

#14:   3. Producing a list of areas from points and adjacens data

       fla(pts, lst) := VECTOR(pts'      , i, DIM(lst))'
#15:                               lst↓i

       flas(pts, fad) := VECTOR(fla(pts, fad ), j, DIM(fad))
#16:                                         j

#17:   4. Visibility check of oriented surfaces using determinants

       showdet(lst) := DET([lst  - lst , lst  - lst ])
#18:                          1      2     3      2

       show(flist) := VECTOR(IF(showdet(flist ) < 0, flist ), j, DIM(flist))
#19:                                          j            j

       unshow(flist) := VECTOR(IF(showdet(flist ) > 0, flist ), j, DIM(flist))
#20:                                            j            j
```

Nun sind die Körper über Punkte und orientierte Flächen zu definieren.

Now the bodies can be defined by points and oriented surfaces.

9. Orientierte Flächen und Sichtbarkeit bei konvexen Körpern

Der folgende Film wird aus einem „Turm" mit passenden Flächen-Adjazenzlisten erzeugt. Zur Demonstration ist die Definition des Körpers angegeben.

Die kompletten DERIVE-Arbeitsblätter finden sich auf der Homepage. Teilweise müssen Aufrufe von Funktion vor dem Plotten ausgewertet (vereinfacht bzw. approximiert) werden, um die Daten für die Plot-Funktion zu erzeugen! So wird Platz für umfangreiche Punktlisten eingespart.

9. Oriented surfaces and hidden lines of convex bodies

The following "movie" is created from a "Tower" with appropriate surface-adjacens lists.

You will find the complete DERIVE-listings on the home page. Some functions must be simplified before plotting to create the plot data.

```
#1:   LOAD(H:\DOKUS\DNL\DNL26\fladlist.mth)

#2:   Once more the tower

               ┌ 0  1  1  0  0  1  1  0  0.5 ┐
#3:   tower := │ 0  0  1  1  0  0  1  1   2  │
               └ 0  0  0  0  1  1  1  1  0.5 ┘

#4:   fadj := [[1, 4, 3, 2, 1], [2, 3, 7, 6, 2], [1, 2, 6, 5, 1], [1, 5, 8,
        4, 1], [5, 6, 7, 8, 5], [3, 4, 9, 3], [3, 9, 7, 3], [7, 9, 8, 7],
        [4, 8, 9, 4]]
```

#5: Generating the surfaces for the i–th picture of the movie

#6: tow(α) := ROTATE_X(α)·ROTATE_Y(30°)·tower

#7: $\text{towerpic}(i) := \text{move}\left(\text{tow}(15\cdot i°),\ \left[\frac{3\cdot i}{4},\ 2,\ 0\right]\right)$

#8: towersurf(i) := transp(pr12(flas(towerpic(i), fadj)))

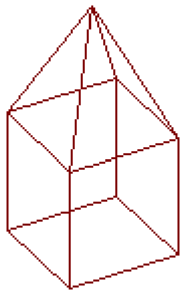
#9: Test: Plot the next expressions!

#10: test_tower := transp(pr12(flas(towerpic(2), fadj)))

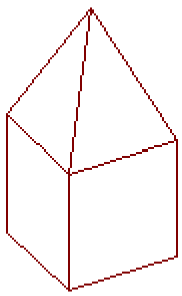
#11: view_tower := show(test_tower)

#12: Finally the movie:

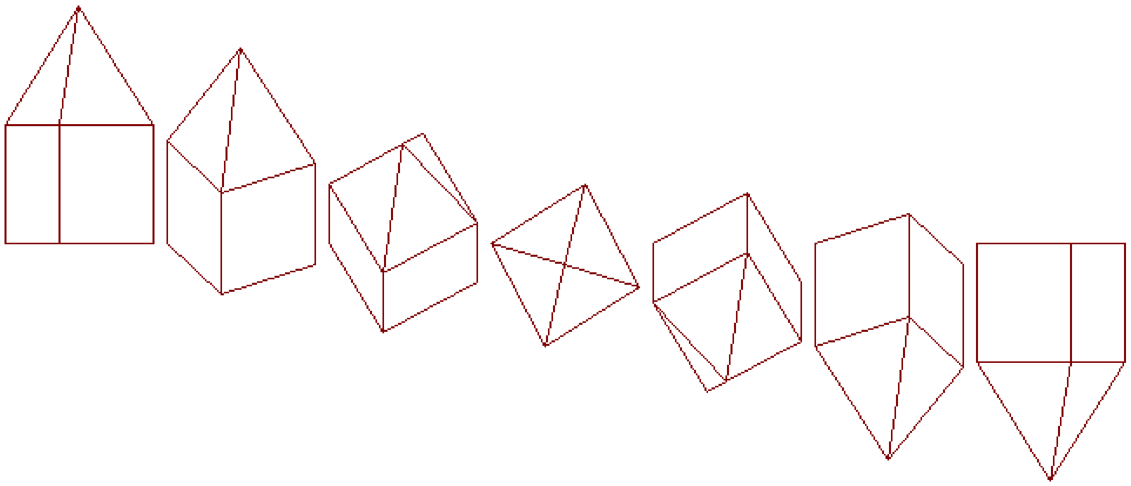
#13: VECTOR(show(towersurf(2·j)), j, 0, 6)



Plot of #10



Plot of #11



The Movie

As you might know from other DNLs, I am very keen in working with graphic representations.

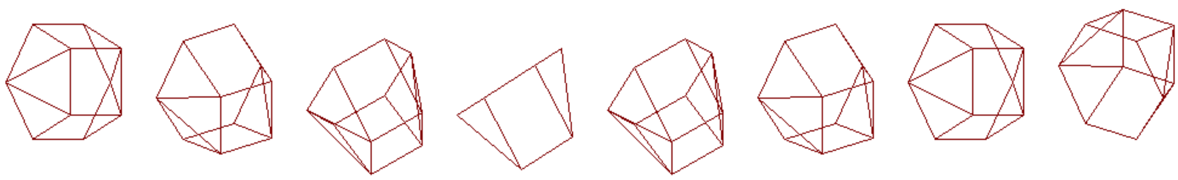
So it was a "must" for me to use Hartmut´s Toolbox.

I tried to produce my own movie with a figure having a hexagon as its base and a square as its top.

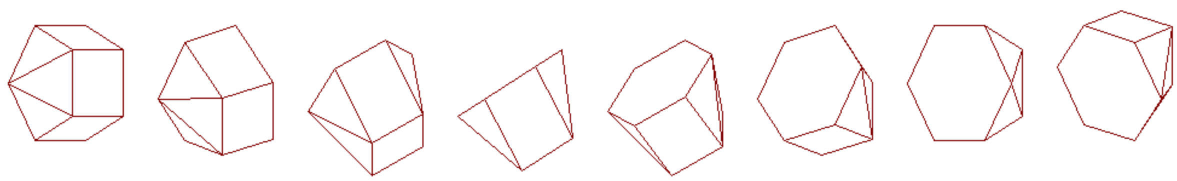
#46: $\text{body} := \begin{bmatrix} 4 & 2 & -2 & -4 & -2 & 2 & 2 & -2 & -2 & 2 \\ 0 & 2\cdot\sqrt{3} & 2\cdot\sqrt{3} & 0 & -2\cdot\sqrt{3} & -2\cdot\sqrt{3} & 2 & 2 & -2 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 5 & 5 & 5 \end{bmatrix}$

#47: $\text{badj} := \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 1 & 2 & 3 & 4 & 4 & 5 & 6 & 1 \\ 2 & 3 & 4 & 5 & 6 & 1 & 8 & 9 & 10 & 7 & 7 & 7 & 8 & 8 & 9 & 9 & 10 & 10 \end{bmatrix}$,

```
#48:  BOD(α) := ROTATE_X(α)·ROTATE_Y(30·1°)·body

#49:  BODPIC(i) := MOVE(BOD(15·i·1°), [5·i, 4, 0])

#50:  VECTOR(KANTEN(proj12·BODPIC(2·i), badj), i, 0, 7)
```



```
#51:  fb := [[1, 6, 5, 4, 3, 2, 1], [7, 8, 9, 10, 7], [1, 10, 6, 1], [1, 7, 10,
        1], [2, 7, 1, 2], [2, 3, 8, 7, 2], [4, 8, 3, 4], [4, 9, 8, 4], [4, 5, 9,
        4], [5, 6, 10, 9, 5]]

#52:  bsurf(i) := transp(pr12(flas(BODPIC(i), fb)))

#53:  VECTOR(show(bsurf(2·j)), j, 0, 7)
```



In the next DNL you will find Hubert Weller´s contribution dealing with the hidden lines. This "geometric" series" will be completed by some of my own tools for special projections (axonometric and central projection, ...)

It might be attractive to compare the various approaches. Josef

DERIVE for DOS and *DERIVE* for WINDOWS

Version 4.05

Dear DERIVErs,

On 5/12/97 Soft Warehouse, Inc. released DfD and DfW 4.05. New functions have been added to existing utility files and more contributions have been made to the user directory.

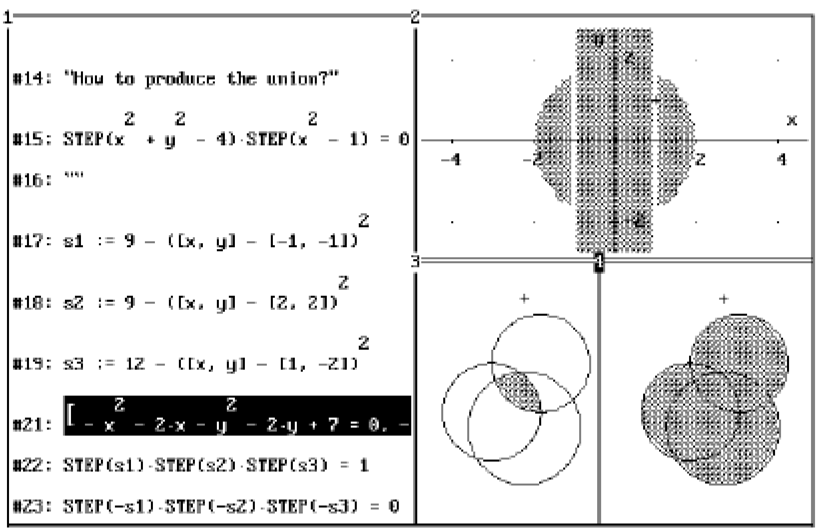
DfW 4.05 makes the following interface improvements over 4.04c:

1. Definite integrals, sums, and products (e.g. INT(SIN(x)/x, x, 0, t)) can now be plotted directly in the 2D-plot window without simplifying or approximating first.
2. When in Case Sensitive Input Mode both upper & lowercase variables can be selected in Calculus and Solve Algebraically listboxes.
3. Horizontal scrolling has been adjusted so that the smallest scrolling increment is one character. Additionally, horizontal scrolling is no longer dependent on the selected expression. The Home and End keys now scroll all the way to the left and right, respectively.
4. Parametric plots now use the given minimum and maximum parameters (in degrees) when in "Degree" angle mode.
5. If-statements using the Boolean operator NOT can now be plotted.
6. Allow vectors of the form [z=2-x-y] to be written to an AcroSpin file instead of having to select the equation within the brackets.
7. The Online Help has been updated; additional indeces and topics have been added. New topics include: Animation with AcroSpin '97, Using and Saving Algebra State Variables; DERIVE Initialization File; and Windows 3.1x Installation Questions.

Soft Warehouse, Inc. (the authors of DERIVE, A Mathematical Assistant)
3660 Waialae Avenue, Suite 304
Honolulu, HI 96816-3259 U.S.A. Tel: (808) 734-5801 Fax: (808) 735-1105
Email: swh@aloha.com Web page: http://www.derive.com

Shading regions –

- See page 43!!



Herr Peter Gräser hat den Wunsch mit einer österreichischen Schule in Kontakt zu treten, um eventuell gemeinsame DERIVE - Projekte durchzuführen. Gerne drucke ich seinen Brief hier ab und würde mich freuen, wenn über den DNL eine Zusammenarbeit zustande käme. Josef

.... ich arbeite am Willi-Graf-Gymnasium Saarbrücken. Saarbrücken ist die Hauptstadt des Saarlandes, eines kleinen Bundeslandes von Deutschland an der französischen Grenze.

Vor einigen Jahren fuhr ich im Auftrag der Fachkonferenz Mathematik/Physik/Informatik nach Wien um mich über den Einsatz von DERIVE im Unterricht kundig zu machen. Dort informierten mich Herr Dr. Heugl und Herr Klinger aufs ausführlichste.

Da unsere Schule bald einen Netzanschluß haben wird, suche ich eine Schule in Österreich, die an gemeinsamen Projekten (vor allem in Mathematik mit DERIVE - Einsatz) über das Netz interessiert ist. Bitte helfen Sie mir eine solche Schule zu finden.

Mit freundlichen Grüßen,
P.Gräser p.graeser@hit.handshake.de

Die zweite Information betrifft ein neues Diskussionsforum:

From: Erich Neuwirth, neuwirth@smc.univie.ac.at

Liebe Kollegen!

Da in Österreich Computer-Algebra ziemlich weit verbreitet im Mathematikunterricht eingesetzt wird, scheint es mir vernünftig zu sein, ein diesem Thema gewidmetes Diskussionsforum zu schaffen. daher gibts seit heute auf dem listserver auf sunsite.univie.ac.at eine neue mailing list

cau-l (computer-algebra im unterricht)

Das Forum soll dem Austausch von Ideen zu diesem Thema dienen und unter anderem erfolgreiche, aber auch weniger erfolgreiche Projekte beschreiben. Bezüglich der eingesetzten Systeme gibts keine Einschränkungen, Derive, Mathematica, Maple .... sind alle gleich willkommen.

sunsite austria hat die Sachwidmung: Computerunterstützung in mathematischer, statistischer und naturwissenschaftlicher Ausrichtung. Daher werde ich auch versuchen, auf der sunsite Sammlungen von Beispielen und Programmen für Computer-Algebra-Systeme anzulegen. Die neue mailing list soll daher auch dazu dienen, Informationen, wo auf dem Internet Resourcen über Computer-Algebra verfügbar sind zu sammeln.

Wie man auf der neuen Liste computer-algebra im unterricht subskribiert: man schickt mail an

listserv@sunsite.univie.ac.at

mit dem Inhalt(/nicht/dem Subject oder Betreff)

subscribe cau-l <Ihr Name>

wobei natürlich <Ihr Name> durch Ihren Namen ersetzt werden muss. Die Liste kann von allen Leuten subskribiert werden.

Bitte subskribieren Sie zahlreich,

Erich Neuwirth

PS.: Normalsprache auf dieser Liste sollte deutsch sein. Allerdings soll das niemanden davon abhalten, Beiträge, die auf dem Netz gefunden wurden, aber beispielsweise englisch sind, auch über die Liste zu verbreiten.

Visit my Spreadsheet home page at `http://sunsite.univie.ac.at/Spreadsite`

Comment of 2011: Some of the functions created in this article by "Mr. Titbits", Johann Wiesenbauer, have been implemented in DERIVE since 1997. Hence all of Johann's complaints are of "historical" interest. It must be mentioned that Johann contributed enormously for improving the number theory functions of DERIVE through many years, Josef

Titbits from Algebra and Number Theory (10)

by Johann Wiesenbauer, Vienna

This time I'm going to deal with a topic that I have put off several times because it scares me a little bit. It is the present state of the utility file number.mth. What gives me the jitters is the question whether I should adopt an attitude that everything is in apple-pie order (apart from, you know, minor insufficiencies that hardly deserve to be mentioned!) or risk the highly esteemed friendship with the people at the Soft Warehouse by not mincing matters. Well, as you may conclude from the last sentence I have made up my mind to choose the latter option and this is what I think of the utility file number.mth (referring to the current version 4.04 of DERIVE). It's a real mess consisting of functions that range from being very useful to totally useless, or to put it in another way, from being terrific to terrible.

"Terrible? What the hell is he talking about?", you might be asking. Well, let me give you some examples. At first two extremely simple functions where the function call is longer (!) than the definition of the function:

```
SQUARE_WAVE(x):=(-1)^FLOOR(x)

POWER_MOD(n,d,m):=MOD(n^d,m)
```

Okay, okay, I'm petty and overcritical, I know. I promise to refrain from further examples of this sort. But what about the following function INVERSE_MOD(m,n) that is supposed to return the inverse of m modulo n?

```
INVERSE_MOD(m,n):=IF(m=1,1,IF(m=n,?,
      (n*INVERSE_MOD(m-MOD(n,m),m)+1)/m))
```

As a first test I wanted to calculate the inverse of -1 mod 10007. But after simplifying

```
INVERSE_MOD(-1,10000)
```

the result was a question-mark (!). Well, obviously they didn't take into account negative numbers, but this is only a small inconvinience as it can be easily circumvented by simplifying the equivalent expression

```
INVERSE_MOD(9999,10000)
```

This time I got the message "Insufficient computation memory" after some seconds. (By the way, my Pentium 200 PC has got 32 MB RAM!) After some more attempts I found out that this routine only works if the modulus n doesn't exceed a number in the range of several thousands. Very strange, indeed! After all, I had shown them a number of times (in DNL #14, #17, #24) how it is done properly! Those routines of mine based on Euclid's Extended Algorithm (EEA) will yield the correct result within fractions of a second even if n has 100 digits or more (Check it!).

I hate to say it, but the next routine SOLVE_MOD(u,x,n) which is supposed to solve linear conguences of the form u(x) ≡ 0 mod n is an even bigger disaster, both in terms of programming and when it comes to performance.

```
SOLVE_MOD_AUX3(s,d,m):=VECTOR(s+m*k_,k_,0,d-1)

SOLVE_MOD_AUX2(c,d,m,x,n):=IF(MOD(c,d)=0,SOLVE_MOD_AUX3(
  MOD(c/d*INVERSE_MOD(m/d,n/d),n/d),d,n/d),?)

SOLVE_MOD_AUX1(u,m,x,n):=SOLVE_MOD_AUX2(m*x-u,GCD(m,n),m,x,n)

SOLVE_MOD(u,x,n):=SOLVE_MOD_AUX1(u,DIF(u,x),x,n)
```

As you can see it uses 4 (!) auxiliary functions including the notorious INVERSE_MOD. In principle, my attitude towards auxiliary functions is the following: Avoid them unless they are unavoidable (as it is often the case with recursive programming) or unless you want to make the program more readable for others, i.e. for didactic reasons. Sometimes an auxiliary function might also be interesting on its own (INVERSE_MOD would be of this kind if it worked properly). In all other cases the abundant use of auxiliary functions (like above!) is usually a sign that the programmer was either not able or too lazy to create a self-contained program.

As for performance it comes as no surprise that SOLVE_MOD inherits all the weaknesses of INVERSE_MOD, which means that it is rather slow and works only for small n. On top of it all, unsolvable congruences will yield a question-mark instead of the empty list [ ] !

How to fix this problem? Well, in the first place we need a working INVERSE_MOD. I would have liked to avoid it, believe me, but here it is (for the fourth time!):

```
INVERSE_MOD(a,m):=IF(GCD(a,m)=1,MOD((ITERATE(IF(MOD(a_,b_)=0,[a_,b_,c_,d_]
    , [b_,MOD(a_,b_),d_,c_-FLOOR(a_,b_)*d_]),[a_,b_,c_,d_],[a,m,1,0])) SUB 4,m))
```

And this is - for purists only! - my version of SOLVE_MOD without any auxiliary functions apart from INVERSE_MOD (u is supposed to be either a linear term or a linear equation in the variable x):

```
SOLVE_MOD(u,x,m):=ITERATE(IF(MOD(LIM(RHS(u)-LHS(u),x,0),t_)=0,
    ITERATES(s_+m/t_,s_, MOD(INVERSE_MOD(DIF(LHS(u),x)/t_,m/t_)*
    LIM(RHS(u)-LHS(u),x,0)/t_,m/t_),t_-1),[ ]),t_,GCD(DIF(LHS(u),x),m),1)
```

It goes without saying that this routine won't let you down even when it comes to numbers with several hundreds of digits and more (Check it!)

One more example concerning the function PELL(n) which returns the n-th number in the Pell sequence $(a_n)_{n\in\mathbb{N}_0}$ defined recursively by

$$a_{n+2} = 2a_{n+1} + a_n, \quad a_0 = 1, a_1 = 0$$

Some time ago (cf. my 'Titbits' in DNL,#21) I showed them how to program the Fibonacci sequence properly by means of the Lucas sequence. That program was polished up by Al Rich himself (cf. DNL #22, p.33) and was used to turn the former "terrible" version of FIBONACCI(n) in number.mth into a "terrific" one.

Well, you won`t be surprised, if I tell you that a similar trick also works in the case of the Pell sequence. All you have to do is to write a program that computes the numbers in the generalized Lucas sequence defined recursively by

$$L_{n+2} = pL_{n+1} - qL_n, \quad L_0 = 2, L_1 = p$$

where the integers p and q are the parameters of the sequence. Just to be on the safe side we carry out these calculations mod m (note that m=0 leaves the numbers unchanged, if it is that what you want!) which leads to the following programming code

```
GEN_LUCAS(n,p,q,m):=IF(n=0,MOD(2,m),(ITERATE(IF((n AND d_)=0,
  [MOD(a_^2-2*MOD(q^c_,m),m),MOD(a_*b_-p*MOD(q^c_,m),m),2*c_,d_/2],
  [MOD(a_*b_-p*MOD(q^c_,m),m),MOD(b_^2-2*MOD(q^(c_+1),m),m),
  2*c_+1,d_/2]), [a_,b_,c_,d_],[2,p,0,2^FLOOR(LOG(n,2))],FLOOR(LOG(n,2))+1)) SUB 1)

LUCAS(n):=GEN_LUCAS(n,1,-1,0)

FIBONACCI(n):=(2*GEN_LUCAS(n+1,1,-1,0)-GEN_LUCAS(n,1,-1,0))/5

PELL(n):=(3*GEN_LUCAS(n,2,-1,0)-GEN_LUCAS(n+1,2,-1,0))/4

PELL(10000)=836..665  (3827 digits; 0.1 sec vs. 4.5 sec with the "old" routine!)
```

Just to give an example where m is not 0, here is a nice probabilistic primality test based on generalized Lucas sequences:

```
LUCAS_PRIME(n,p):=IF(GEN_LUCAS(n,p,1,n)=p,true,false)

LUCAS_PRIME(10^100+267,1)=true  (0.3 sec !)
```

I leave it to you to experiment a little bit and find composite numbers that are "pseudo-prime" with respect to this test for some choice of p.

Would you believe that there are functions in number.mth that are never used in the literal sense of the word because they are overwritten during the loading process? There are even 3 (!) examples of this kind, namely the first definitions of MERSENNE(n), PARTS(n) and DISTINCT_PARTS(n). (In the case of the partition functions this is all the more interesting, as their second definitions are incredibly powerful leaving even the built-in functions of Mathematica and MAPLE in the dust.)

Well, I could go on and on pointing out inadequacies, but actually this would miss the point. As a matter of fact, the real weaknesses of number.mth are not the functions which are included in it, but the functions that are supposed to be there but aren't (like Euler's φ-function, Möbius' μ-function, the divisor functions σ and τ, Jacobi-symbols (a/n) and what have you. Needless to say that all these functions and many more have been defined in this column!).

Well, while writing these lines version 4.05 of DfW has just arrived. Will there be big changes in the untility-file number.mth making this article obsolete? After all, Josef is urging me to deliver it… No, all the blunders described above are still there, but a lot of new functions have been added that nobody has asked for. The input of FIBONACCI(0) or LUCAS(0) still leads to a very nasty lock-up! Are there any other users of number.mth apart from me. If so, what do they think of it? I'll try to find the answer to this question at the next DERIVE-conference in Sweden. See ya! (j.wiesenbauer@tuwien.ac.at)

Revising the next contribution was a lot of work. The DERIVE-functions from 1997 (Derive for DOS, version 3) did not work because of several reasons. One of them is that the SOLVE command was accomplished by SOLUTIONS in later versions. Another reason is that the first function REL_AUX2(w) does not work and then all further functions which depend on it …

What to do? Rewriting them and using DERIVE 6's programming facilities.

I reprint the old version and then append the DERIVE 6 file.

From Linear Inequalities to Linear Programming (1)

Josef Böhm, Würmla, Austria

I have been concerned with Linear Programming for a long time. Some time ago I produced a *DERIVE* utility for applying the SIMPLEX-method. I´ll refer to this tool at the end of this sequence of contributions (International DERIVE Journal, vol1, number 3). When you introduce Linear Programming you will start with problems which can be solved graphically. So I tried to use *DERIVE* for that approach. The graphic representation of linear inequalities is one of the im-portant basic skills connected with graphic solutions of optimisation problems. Unfortunately I didn´t know the "griddy" trick with implicit functions - I could use the poor excuse that I pro-duced this tool at times of version 2.x - so I developed functions to shade half planes according to the given inequalities.

I used Sergey Biryukow´s file from DNL#22 to handle the inequations. Unfortunately I am not so skilled in programming powerful tools without using auxiliary functions like Johann Wiesenbauer, so I apologize for my "programming style".

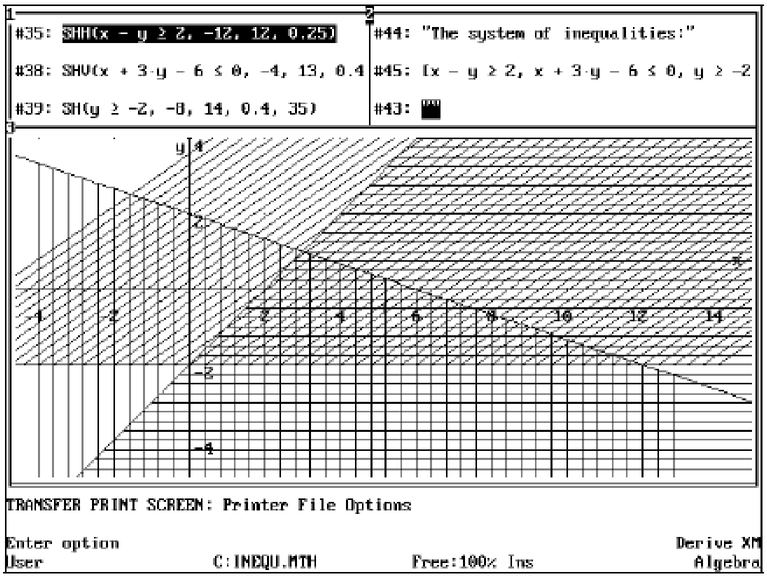
```
#1:   [InputMode := Word, Notation := Decimal, NotationDigits := 3]

#2:   REL_AUX2(w) := 100·IF(w - (LHS(w) = RHS(w))) + 10·IF(w - (LHS(w) + 1 = RHS(w))) +
        IF(w - (LHS(w) - 1 = RHS(w)))

                                                          ⎡ 100   =  ⎤
                                                          ⎢  1    >  ⎥
#3:   REL_AUX1(x) := (SELECT(x = rel_aux_ , rel_aux_,     ⎢ 10    <  ⎥ ))
                                         1                ⎢ 101   >= ⎥
                                                          ⎢ 110   <= ⎥
                                                          ⎣ 11    /= ⎦
                                                                        1,2

      REL(w) :=
        If NUMBER(REL_AUX2(w))
#4:        REL_AUX1(REL_AUX2(w))
           "="

      SGN(z) :=
        If z > 0
           1
#5:        If z < 0
              -1
              1

#6:   V(u) := DIM(VARIABLES(u))

      HELP(u) :=
        If V(u) = 2 ∨ (VARIABLES(u))↓1 = y
#7:        1
           0
           0
```

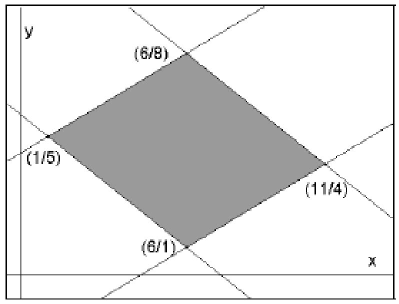
```
#8:   [TEST(u) := SOLVE(u, y), TEST2(u) := SOLVE(u, x)]

      UNH(u) :=
        If HELP(u) = 1
#9:        TEST(u)
           TEST2(u)

      AUX(u) :=
        If REL(UNH(u)) = "<=" ∨ REL(UNH(u)) = "<"
#10:       -1
           1
           1

                d
#11:  SL(u) := —— RHS(UNH(u))
                dx

#12:  G(u) := SOLUTIONS(LHS(u) = RHS(u), y)

      P(u) := [x_, lim    (G(u))  ]
#13:               x→x_          1

#14:  X_C(u) := SOLUTIONS(u, x)

      LIN(u) :=
        If HELP(u) = 1
#15:       y = (G(u))↓1
           x = X_C(u)

#16:  SHV(u, a, b, inc) := APPEND([LIN(u)], VECTOR([P(u), [x_, - 100·AUX(u)]], x_, a, b,

        inc))

      SHH(u, a, b, inc) := APPEND([LIN(u)], VECTOR([P(u), [100·SGN(SL(u))·AUX(u),
#17:

        (P(u))  ]], x_, a, b, inc))
              2

      SW(u, α) :=
        If SL(u) < 0 ∧ (α ≤ 0 ∧ SL(u) < TAN(α·1°))
           -1
#18:       If SL(u) > 0 ∧ (α ≥ 0 ∧ SL(u) > TAN(α·1°))
              -1
              1
           1

#19:  SH(u, a, b, inc, α) := APPEND([LIN(u)], VECTOR(IF(HELP(u) = 1, [P(u), [x_ + SW(u,

        α)·100·AUX(u)·SGN(α), (P(u))  + SW(u, α)·100·AUX(u)·|TAN(α·1°)|]], [
                                    2

              X_C(u)                          x_
                                                                    ]), x_, a, b, inc))
        X_C(u) + 100·AUX(u)   x_ + 100·TAN(α·1°)·AUX(u)

#20:  SHH(x - y ≥ 2, -12, 12, 0.25)

#21:  SHV(x + 3·y - 6 ≤ 0, -4, 13, 0.4)

#22:  SH(y ≥ -2, -8, 14, 0.4, 35)
```

SHV(u,a,b,inc) vertical shading for u with a ≤ x ≤ b with increment inc in x-direction

SHH(u,a,b,inc) horizontal shading for u with a ≤ x ≤ b with increment inc in y-direction

SH(u,a,b,inc,α) shading for u with a ≤ x ≤ b with increment inc in x-direction showing an angle of -π/2 < α < π/2 (with +x-direction)



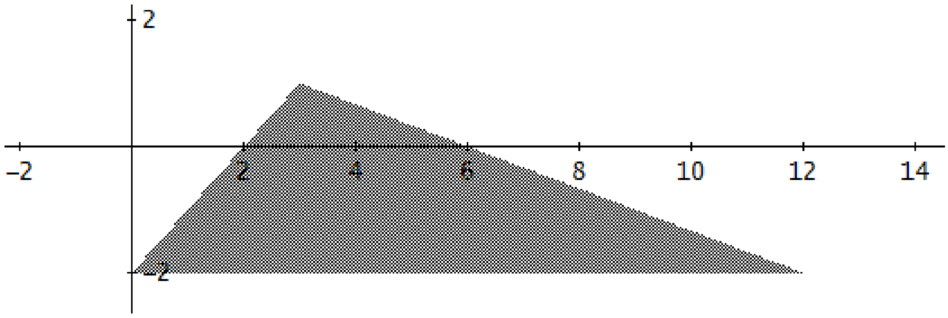


Problem for pupils:
Describe the shaded area by a system of inequalities and check your proposal.

This was the original contribution from 1997.

Yes, I know, that we can now shade parts of the xy-plane described by inequalties.

$x - y \geq 2 \wedge x + 3 \cdot y - 6 \leq 0 \wedge y \geq -2$



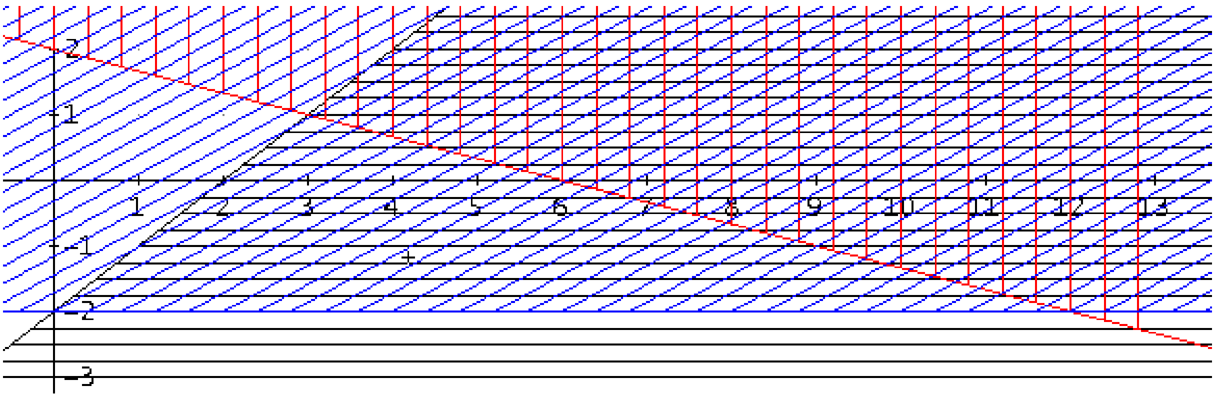
But it is my strong opinion that in many cases it would be useful demonstrating how this region (here the triangle) is created step by step adding one inequality after the other. Doing this with DERIVE 6, each plot of an inequality covers the previous ones.

How can we do it in DERIVE 6?

```
      REL(w) :=
        If (STRING(w))↓1 = "≤" ∨ (STRING(w))↓1 = "<"
#1:         1
            -1

      g(u, u_) :=
        Prog
          u_ := SOLUTIONS(LHS(u) = RHS(u), y)
#2:       If u_ = []
             [SOLVE(LHS(u) = RHS(u), x), ∞]
             [u_↓1, ∂(u_↓1, x)]

      p(u, x_) := ⎡x_,  lim   (g(u, u_))  ⎤
#3:               ⎣     x→x_           1 ⎦
```

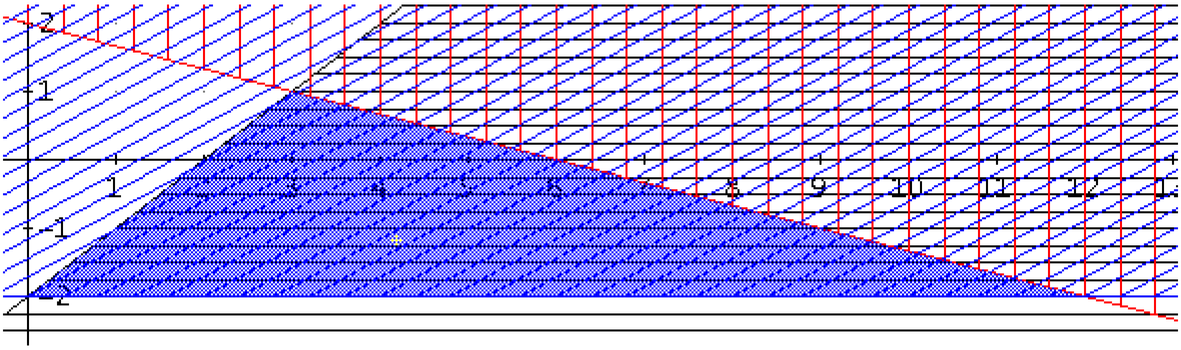
```
      shh(u, a, b, inc, sh) :=
        Prog
          If (g(u, u_))↓2 > 0 ∨ (g(u, u_))↓2 < 0
             sh := VECTOR([p(u, x_), [(p(u, x_))↓1 - REL(u)·1000, (p(u, x_))↓2]], x_, a, b, inc)
          If (g(u, u_))↓2 = 0
             Prog
               If REL(u) = 1
#4:               [a := MIN(a, b), b := (g(u, u_))↓1]
                  [a := (g(u, u_))↓1, b := MAX(a, b)]
               DISPLAY([a, b])
               sh := VECTOR(y = (g(u, u_))↓1 - i·REL(u)·inc, i, 0, (b - a)/inc)
          If (g(u, u_))↓2 = ∞
             sh := VECTOR([RHS((g(u, u_))↓1), x_; RHS((g(u, u_))↓1) - REL(u)·1000, x_], x_, a, b, inc)
          sh := APPEND([(g(u, u_))↓1], sh)

      shv(u, a, b, inc, sh) :=
        Prog
          If (g(u, u_))↓2 > 0 ∨ (g(u, u_))↓2 < 0
             sh := VECTOR([p(u, x_), [(p(u, x_))↓1, (p(u, x_))↓2 - SIGN((g(u, u_))↓2)·REL(u)·1000]], x_, a,
          If (g(u, u_))↓2 = ∞
             Prog
               If REL(u) = 1
#5:               [a := MIN(a, b), b := RHS((g(u, u_))↓1)]
                  [a := (RHS(g(u, u_)))↓1, b := MAX(a, b)]
               DISPLAY([a, b])
               sh := VECTOR(x = a + i·inc, i, 0, (b - a)/inc)
          If (g(u, u_))↓2 = 0
             sh := VECTOR([x_, (g(u, u_))↓1; x_, (g(u, u_))↓1 - REL(u)·1000], x_, a, b, inc)
          sh := APPEND([(g(u, u_))↓1], sh)



        b, inc)
```

```
#7:   shh(x - y ≥ 2, -12, 12, 0.25)

#8:   shv(x + 3·y - 6 ≤ 0, -4, 13, 0.4)

#9:   sho(y ≥ -2, -8, 14, 0.4, 35)
```



You can distinguish the regions described by the three inequalities.

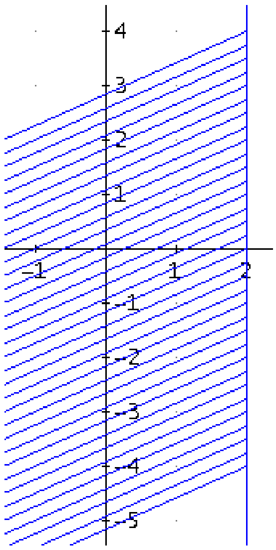
#10: $x - y \geq 2 \wedge x + 3 \cdot y - 6 \leq 0 \wedge y \geq -2$



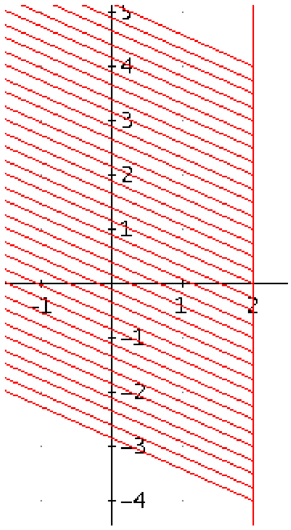
I superimpose the Boolean expression combining the given inequalities.

Here are some more examples to demonstrate my function sho:

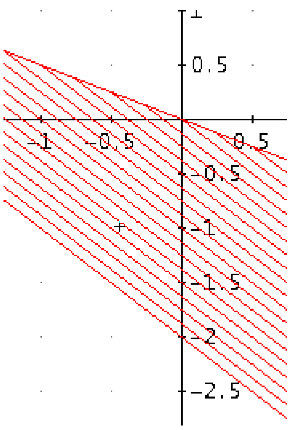
sho(x ≤ 2, −4, 4, 0.25, 30)



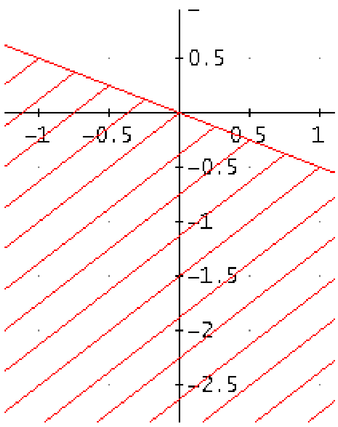
sho(x ≤ 2, −4, 4, 0.25, 30, −1)



sho(x + 2·y ≤ 0, −4, 4, 0.25, 45)



sho(x + 2·y ≤ 0, −4, 4, 0.25, 45, −1)



You can enter the angle of the shading - here 30° and 45° - and the direction – the last parameter.

Line Searching with *DERIVE*

George Collie, Dundee, Scotland, g.collie@abertay-dundee.ac.uk

Line-searching is a basic technique in numerical optimisation. A search direction is obtained say $[d_1, d_2]$ when minimising $f(x,y)$ or $[d_1, d_2, d_3]$ when minimising a function of three variables.

Searching along this direction from a starting point at (x_0, y_0) { or (x_0, y_0, z_0) } is carried out by minimising $f(x_0 + t\cdot d_1, y_0 + t\cdot d_2)$, {or $f(x_0 + t\cdot d_1, y_0 + t\cdot d_2, z_0 + t\cdot d_3)$} which is a function of ***one*** variable t and is called `LS(t)` in OPT2D_util and OPT3D_util.

The two simplest methods of obtaining a search direction for minimisation are :

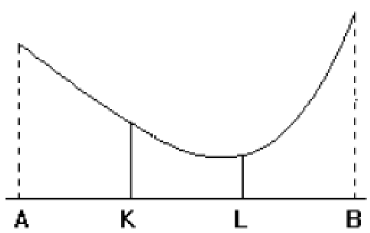
(1) Steepest descent { − Grad $F(x_0, y_0)$ or − Grad $F(x_0, y_0, z_0)$ }

(2) The Newton method { $-H^{-1}\cdot$Grad $F(x_0, y_0)$ or $-H^{-1}\cdot$Grad $F(x_0, y_0, z_0)$ }

The three methods of numerical optimisation for a function of one variable used in OPT2D_util.MTH and OPT3D_util.mth are:

(a) Golden section
(b) Fibonacci search
(c) Quadratic interpolation

The steepest descent method (STEEPD) uses the direction in which the function has the largest possible rate of decrease at (x_0, y_0, z_0). The Newton method (NEWTD) uses the Hessian matrix of second derivatives to fit a quadratic function to the function and points to the exact minimum of this quadratic function. The procedure NEWTSTEP $\left\{x_{n+1} = x_n - H_n^{-1}\cdot\nabla f(x_n)\right\}$ gives the result of the version of the method with no line searches and is equivalent to setting a value of $t = 1$ in the linesearch `LS(t)` when NEWT2D has been used. The first two univariate search methods are based on finding `LS(t)` at 2 points K and L say inside some given range (A, B) where A ≤ K ≤ L ≤ B



Assuming the function is unimodular in [A, B]: If f(L) < f(K) the interval [A, B] becomes [K, B]; otherwise it becomes [A, L].

When the method is applied to the new range, it would be useful if the best value (which is L in [K,B] or K in [A,L] could be re-used.

If AL/AB and KB/AB are each equal to the golden section ratio ($\sqrt{5} - 1$)/2 ≈ 0.618034, then the interval is always reduced in the same ratio and the best value can be re-used.

Similarly, if AL/AB and KB/AB each equal the ratio F_{n-1}/F_{2n} of two consecutive Fibonacci numbers, then this best value can be re-used in the next interval with the next ratio becoming F_{n-2}/F_{n-1}.

In OPT2D_util.MTH, although the values will be re-used, they will also be re-calculated which would not be necessary if the method was programmed in FORTRAN or PASCAL.

The third method of univariate searching fits a quadratic $P\,t^2 + Q\,t + R$ through three points $[t_1, LS(t_1)]$, $[t_2, LS(t_2)]$, $[t_3, LS(t_3)]$ and calculates the value $t^* = -Q/(2P)$ where the turning point of the quadratic lies. The method cannot be guaranteed to produce a better value than any of the t_i but (especially if $t_1 < t_2 < t_3$ and $LS(t_2) < LS(t_1)$, $LS(t_2) < LS(t_3)$), the method often produces a good value.

The utility files OPT2D_util.MTH (and OPT3D_util.MTH) were used with a class of civil engineers who had experience of computer usage but no deep interest in the mathematics behind optimisation techniques. One hour in a lecture room and one hour in a computer laboratory over a period of five weeks were allocated for this section of the course.

The first step after loading in the utilities say in OPT2D.MTH_util, is to declare a function $f(x,y)$. Then an initial point is set, a direction is chosen and a line-search function is the available.

The procedures available are:

SETX0(*x0,y0*)	sets the initial point (*x_0,y_0*) as (*x0,y0*)	*
SETXO(*x0,y0*)	sets the initial point (*x_0,y_0*) as (*x0,y0*)	*
STEEPD	sets(*d_1,d_2*) as the steepest descent direction	*
NEWTD	sets(*d_1,d_2*)as the Newton search direction	*
LS(*t*)	is the line search function: *f(x_0 + t*d_1, y0 + t*d_2)*	
GETNEWPT(t_1)	gives the result of choosing the search parameter as t_1	*
QUADSEARCH(t_1,t_2,t_3)	applies the quadratic searching method	*
GOLDEN(t_1,t_2)	applies one golden section step	
FIBON(n,t_1,t_2)	applies one Fibonacci step (n = 1, 2, ..., 16)	*
NEWTSTEP	gives the result of one step of Newton´s method	*
CHECK	shows the current function, initial point and direction *	

The procedures marked with * have to be Authored and the approXimated. The three processes of defining $f(x,y)$, choosing an initial point and selecting a search direction are crucial because thea are the only procedures which change the standard quantities $f(x,y)$, (*x_0,y_0*) and (*d_1* and *d_2*).

In the typical session shown, in session.dfw, the function $x^4 + 2x + y^4$ minimised, the method of steepest descent being used from a starting point at (1,1). A parameter value of ***t* = 1** gives a bad result (696 is much larger than 4). Then, the method of quadratic searching is used on the values 0, 0.5 and 1. This produces a good value with ***t* = 0.243323** which is used to generate a new point at (–0.459937,0.026708) and another search direction chosen. The second search is also done by a mixture of experiment and the golden section method and the third search utilises the Fibonacci method for $n = 5$.

The result after three very exact line searches (which can also be easily done using these utilities) is: $x = -\,0.791687$, $y = -\,0.178410$, $f(x,y) = -\,1.1892$. Hence, the rather haphazard method shown is competitve with this.

The utilities can also be used for Newton´s method, with and without line searching.

Other options available for line searching are graphing LS(t) or solving LS´(t) = 0.

Note:

(a) The problem of students reading SETX0 as SETXO is avoided by allowing both options.

(b) The traditional advantage of "re-using function values" in the three search methods is lost but is replaced by greater flexibility.

(c) The method of generating Fibonacci numbers is quicker than the procedure used in NUMBER.MTH though less sophisticated and always limited to some finite value of ´*n*´.

(d) The check procedure is useful in dealing with students whose answers were not correct.

The session printout will show that GOLDEN and FIBON include some hints (programmed using IF-statements) for choosing a new range of parameter values, although in practice the best technique is usually to accept the first reasonable improvement found).

OPT3D_util.MTH is an obvious extension of OPT2D_util.MTH with most of the procedures having to be changed, although `FIBON`, `GOLDEN` and `FIBONACCI` are unchanged.

#1: A set of procedures to enable line-searching to minimise F(x,y)

#2: [F(x, y) :=, x_0 :=, y_0 :=]

#3: SETX0(x0, y0) := [x_0 := x0, y_0 := y0, Function Value = , F(x0, y0)]

#4: SETX0(x0, y0) := [x_0 := x0, y_0 := y0, Function Value = , F(x0, y0)]

#5: $gg := \lim_{y\to y\_0} \lim_{x\to x\_0} \left[\frac{d}{dx} F(x, y), \frac{d}{dy} F(x, y)\right]$

#6: $hess := \begin{bmatrix} \lim_{y\to y\_0} \lim_{x\to x\_0} \left(\frac{d}{dx}\right)^2 F(x, y) & \lim_{y\to y\_0} \lim_{x\to x\_0} \frac{d}{dy}\frac{d}{dx} F(x, y) \\ \lim_{y\to y\_0} \lim_{x\to x\_0} \frac{d}{dy}\frac{d}{dx} F(x, y) & \lim_{y\to y\_0} \lim_{x\to x\_0} \left(\frac{d}{dy}\right)^2 F(x, y) \end{bmatrix}$

#7: $newtstep := \begin{bmatrix} x\_0 \\ y\_0 \end{bmatrix} - hess^{-1} \cdot \begin{bmatrix} gg_1 \\ gg_2 \end{bmatrix}$

#8: $steepd := \left[d\_1 := - gg_1, d\_2 := - gg_2\right]$

#9: $newtd := \left[d\_1 := \left(- hess^{-1} \cdot \begin{bmatrix} gg_1 \\ gg_2 \end{bmatrix}\right)_{1,1}, d\_2 := \left(- hess^{-1} \cdot \begin{bmatrix} gg_1 \\ gg_2 \end{bmatrix}\right)_{2,1} \right]$

#10: LS(t) := F(x_0 + t·d_1, y_0 + t·d_2)

#11: $GETNEWPT(t) := \begin{bmatrix} X & Y & F(X,Y) \\ x\_0 + t\cdot d\_1 & y\_0 + t\cdot d\_2 & LS(t) \end{bmatrix}$

#12: $QUADSEARCH(t1, t2, t3) := \begin{bmatrix} & t1 & t2 & t3 & \\ t & t1 & t2 & t3 & t4 = t\_s := \frac{t2 + t3}{2} - \frac{LS(t2)\cdot(t1^2 - t1\cdot(t2 + t3) + t2\cdot t3) + LS(t3)\cdot(t3 - t1)\cdot(t1 - t2)}{2\cdot(LS(t1)\cdot(t2 - t3) + LS(t2)\cdot(t3 - t1) + LS(t3)\cdot(t1 - t2))} \\ LS(t) & LS(t1) & LS(t2) & LS(t3) & LS(t4) = LS(t\_s) \end{bmatrix}$

(the last column is headed "New t")

#13: $ga := \frac{\sqrt{5} - 1}{2}$

```
                            ┌                       a                                        K
                            │
                            │     t                 a                      kk := b - ga·(b - a)
#14:  GOLDEN(a, b) :=       │
                            │ LS(t)              LS(a)                                   LS(kk)
                            │
                            └       IF(LS(kk) < LS(ll), New a  = a, )  IF(LS(kk) < LS(ll), Best point?, New a  =

                                       L                                      b          ┐
                                                                                         │
                               ll := a + ga·(b - a)                           b          │
                                                                                         │
                                      LS(ll)                                LS(b)        │
                                                                                         │
      kk)  IF(LS(kk) < LS(ll), New b  = ll, Best point?)  IF(LS(kk) < LS(ll), , New b  = b) ┘

#15:  fb := [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584]

      FIB(j) := fb
#16:              j

                              ┌                     a                                       K
                              │
                              │                                                  FIB(n - 2)·(b - a)
                              │     t               a                  kk := a + ──────────────────
#17:  FIBON(n, a, b) :=       │                                                       FIB(n)
                              │
                              │ LS(t)             LS(a)                                  LS(kk)
                              │
                              └      IF(LS(kk) < LS(ll), New a  = a, )  IF(LS(kk) < LS(ll), Best point?, New a  =

                                     L                                     b            ┐
                                                                                        │
                                 FIB(n - 1)·(b - a)                                     │
                      ll := a +  ──────────────────                        b            │
                                      FIB(n)                                            │
                                                                                        │
                                   LS(ll)                                LS(b)          │
                                                                                        │
      kk)  IF(LS(kk) < LS(ll), New b  = ll, Best point?)  IF(LS(kk) < LS(ll), , New b  = b) ┘

                 ┌     F(x,y) =       F(x, y)  ┐
                 │                             │
#18:  check :=   │   (x0, y0) =    [x_0, y_0]  │
                 │                             │
                 └  Direction =    [d_1, d_2]  ┘
```

The example is following now:

```
#1:   LOAD(I:\DOKUS\DNLs\DNL97\MTH26\OPT2D_util.mth)

                     4             4
#2:   F(x, y) := x   + 2·x + y

#3:   SETX0(1, 1) = [x_0 := 1, y_0 := 1,  Function Value =, 4]

#4:   steepd = [d_1 := -6, d_2 := -4]

#5:   LS(1) = 696

#6:   QUADSEARCH(0, 0.5, 1)

      ┌         t1   t2    t3              New t              ┐
      │                                                         │
#7:   │     t    0   0.5    1        t4 = t_s := 0.2433234421  │
      │                                                         │
      └ LS(t)    4   13    696    LS(t4) = -0.8751293389       ┘

#8:   GETNEWPT(0.2433234421)

      ┌         X                 Y               F(X,Y)      ┐
#9:   │                                                         │
      └ -0.4599406528    0.02670623145    -0.8751293389        ┘

#10:  SETX0(-0.4599406528, 0.02670623145)
```

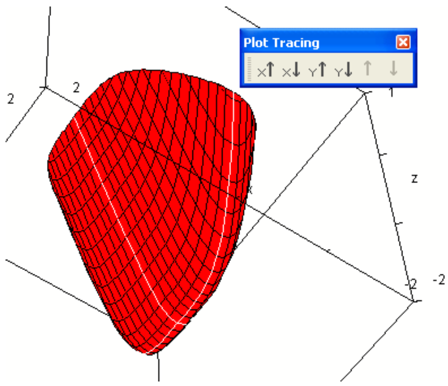
```
#11:  [x_0 := -0.4599406528, y_0 := 0.02670623145,  Function Value = , -0.8751293389]

#12:  steepd

                                                     -5
#13:  [d_1 := -1.610806674, d_2 := - 7.618997253·10   ]

#14:  LS(0.25) = -1.171522271

#15:  LS(0.5) = 0.03281904911

#16:  GOLDEN(0, 0.5)

                                  [      X              Y             F(X,Y)    ]
#17:  GETNEWPT(0.1909830056) =    [                                             ]
                                  [ -0.767577353  0.02669168046  -1.188027026   ]

#18:  SETX0(-0.767577353, 0.02669168046)

#19:  [x_0 := -0.767577353, y_0 := 0.02669168046,  Function Value = , -1.188027026]

#20:  steepd = [d_1 := -0.1910504741, d_2 := -0.00007606550317]

#21:  LS(0.2) = -1.189992455

#22:  FIBON(5, 0, 0.3)
```

#23:

	a	K	L	b
t	0	kk := 0.12	ll := 0.18	0.3
LS(t)	-1.188027026	-1.19051175	-1.190290231	-1.186775509
	New a = 0	Best point?	New b = 0.18	

In DERIVE 6 we can plot the function in the 3D-plot window and investigate the minimum applying the trace mode.

The intersection point of the parameter curves is given with (-0.8, 0, -1.904), which is pretty close to the minimum from above.



Start and end of session2.dfw which searches for the mimimum of $F(x,y,z)$:

```
#1:   LOAD(I:\DOKUS\DNLs\DNL97\MTH26\OPT3D_util.mth)

#2:   Precision := Approximate

                    2       2     4
#3:   F(x, y, z) := x  + 2·y  + z  + 3·x·y + 2·z

#4:   SETX0(1, 1, 2)

#5:   [x_0 := 1, y_0 := 1, z_0 := 2,  Function Value = , 26]


                           -13                       -13
#19:  [x_0 := - 2.85874457·10    , y_0 := - 2.85874457·10    , z_0 := 0.6176279859,

        Function Value = , 1.380771006]
```

"Special Shaded Planes"

Sebastiano Cappuccio, Forli, Italy, scappucc@spfo.unibo.it
and Nurit Zehavi, Israel, ntzehavi@wiccmail.weizmann.ac.il

On 3 April I received an e-mail from Sebastiano Cappuccio, Italy which extended to a fruitful discussion within the DERIVE Newsgroup involving other DERIVIANS, especially Nurit Zehavi from Israel. Let´s begin the story with Sebastiano´s email which might be of interest for many of you. I must admit, that the "trick with the grid" was absolutely new for me - a shame, indeed. Josef

With *DERIVE n* ($n \geq 3$), if you plot an univariate or bivariate identity (e.g. $x = x$, or $x + y = x + y$), you get a "shaded plane" with a "grid" depending on the Accuracy[1].
It is easy to explain this (*DERIVE* scans the whole plane in plotting implicit equations). If you want to plot the "grid" only for positive *x*, you have to plot an identity true only for positive *x*:

`abs(x) = x.`

Thus, if you want a "shaded" area between the straight lines $x = 2$ and $x = –2$, you can plot:

`abs((2-x)(x+2)) = (2-x)(x+2).`

If you want to get the "grid" between two curves *u* and *v*, you have to plot:

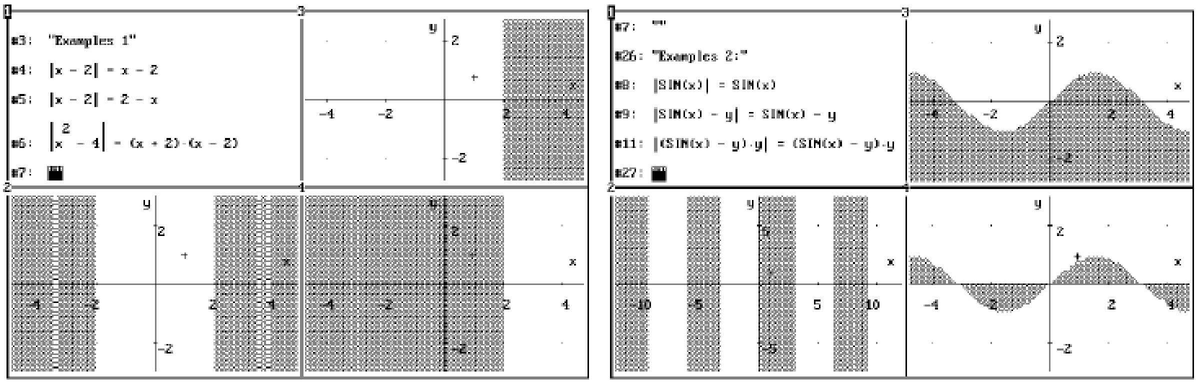
`abs((u-y)(y-v)) = (u-y)(y-v).`

For example try with

`U:=sin(x), v:= 0,` or with `u:=x^2-4, v:=0.`

My problem is: how can I get a shaded area between two curves AND within an interval (for example between $x^2 – 4$ and 0, but only with $–2 < x < 2$? It seems that DERIVE does not accept logical operators in plotting identities. Can someone help me? Thank you.

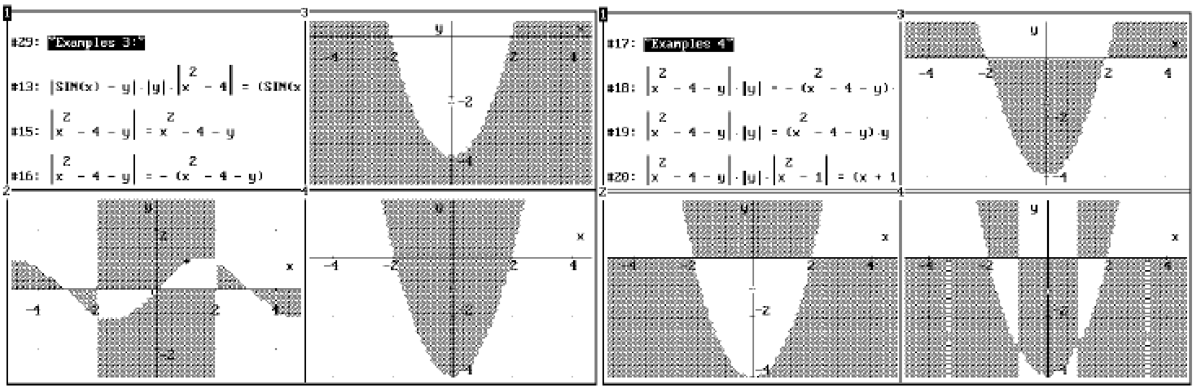
As you might imagine, I was very interested to see how this works and probably to find a solution for Sebastiano´s problem. See some screen shots:



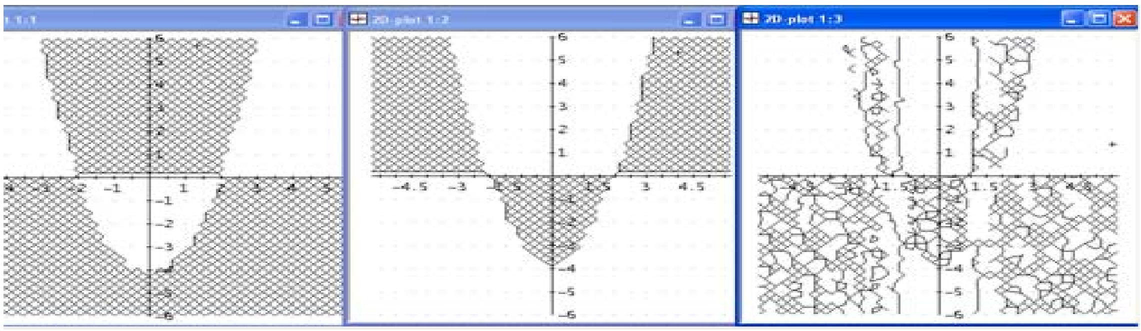
Screen shots from DERIVE for DOS

I don’t know the reason for some “big” meshes appearing in the grid.

[1] You cannot set the Accuracy in DERIVE 6, so you cannot influence the densitiy of the grid! One DERIVE 6 plot will follow after the next DfD plot from 1997, Josef



More screen shots from DERIVE for DOS



Examples 4 reproduced with DERIVE 6

Some days later we (Sebastiano and I) found an answer from Nurit in our electronic mailbox:

Hello Sebastiano,
There is a solution to your problem using DERIVE:

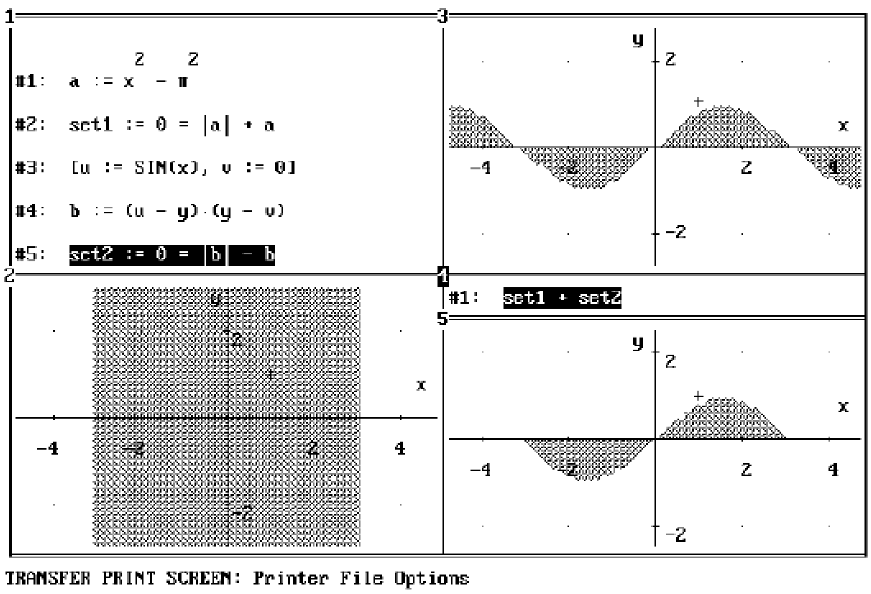
```
a:= x^2 - pi^2
set1:=0=abs(a)+a
u:=SIN(x)
v:=0
b:=(u-y)*(y-v)
set2:=0=abs(b)-b
```

plot set1
plot set2

now simplify

```
set1 + set2
```

and plot the intersection of set1 and set2.



All the best,
Nurit Zehavi and Giora Mann

By the way I have just sent an abstract to David Sjöstrand in Sweden for the "FUN"-Conference, entitled

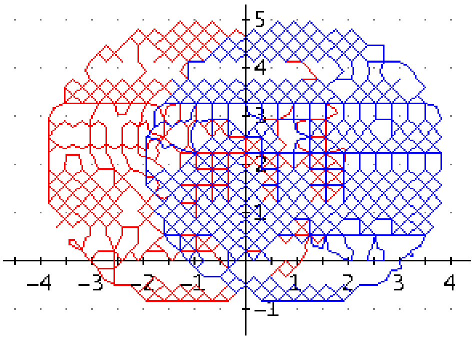
FUNtastic Implicit Plotting.

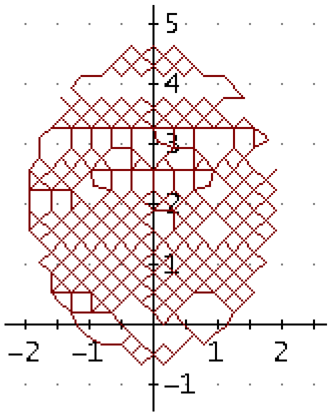
We (Giora Mann is the co-author have been using games on dots, lines and half-planes aappyling expressions similar to the examples presented by Sebastiano Cappuccio.

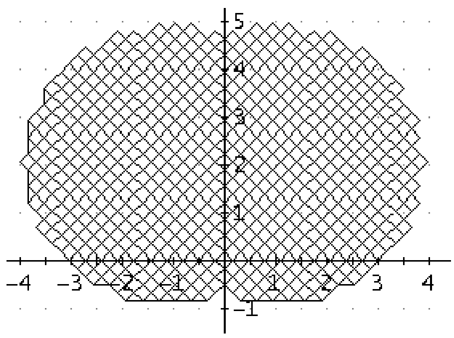
Nurit's next mail arrived on 18 May 1997:

Several *DERIVE*-news members mentioned that they couldn´t replicate my example and asked me to elaborate.
I will illustrate the method by using another ´nice´ example (taken form our teacher textbook):

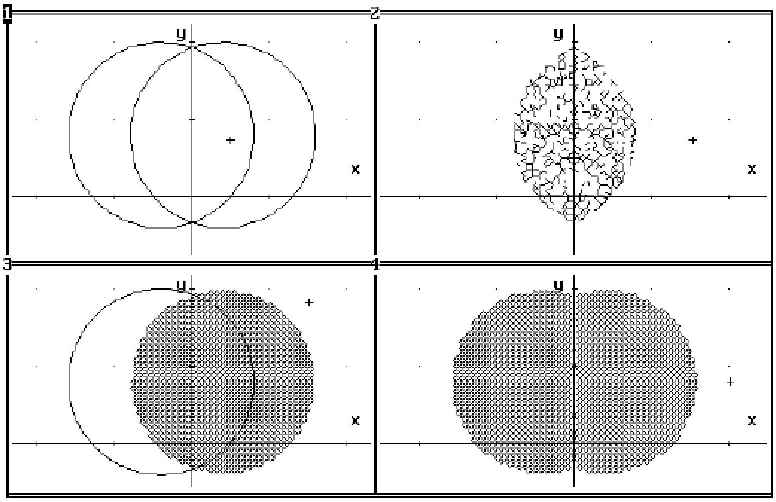
```
#1:   Define two circles

#2:   circ1 := ([x, y] - [-1, 2])^2 - 9

#3:   circ2 := ([x, y] - [1, 2])^2 - 9

#4:   Plot the next two expressions:

#5:   0 = |circ1| + circ1

#6:   0 = |circ2| + circ2

#7:   To plot the intersection:

#8:   0 = (|circ1| + circ1) + (|circ2| + circ2)

#9:   To plot the union:

#10:  0 = (|circ1| + circ1)·(|circ2| + circ2)

#11:  Predict the results of:

#12:  0 = (|circ1| - circ1) + (|circ2| - circ2)

#13:  0 = (|circ1| - circ1)·(|circ2| - circ2)
```







You can see the plots produced with DERIVE 6. The original plots from 1997 are on the next page.



Nurit wrote:

Now I have a question to SWHH: If I ask to plot circ1 (and not 0 = circ1), then the line y = 0 is plotted without considering for circ1; we have to simplify first. WHY?

23 May 1997: Another mail from Sebastiano flatters into my mailbox. Sebastiano offers additional information.

The idea of the trick for shaded areas started when a student of mine tried to plot the $y = x$ function, but for a typing error, he plotted $x = x$. I think that it occurs because DERIVE scans the whole plane in plotting implicit equations and connects points satisfying the condition (that is all the points of the plane, according to the Accuracy set): setting a lower Accuracy, you get a "large meshed net" Of course this tricks works only with DERIVE 3.x, because former releases cannot plot implicit graphs.

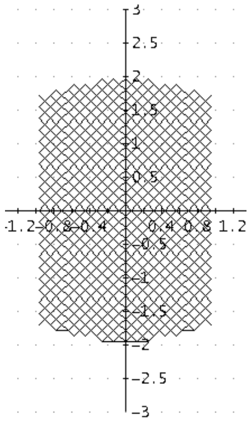
You certainly know Nurit´s messages, maybe you do not know the following messages with great ideas from England and Chile.

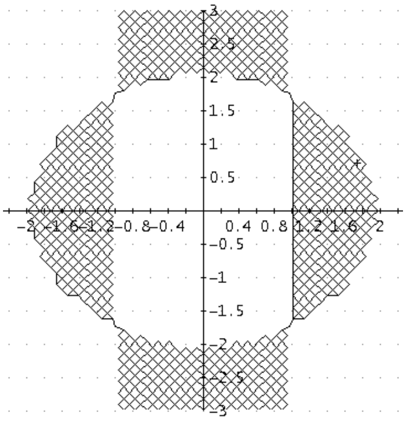
From John Alexiou

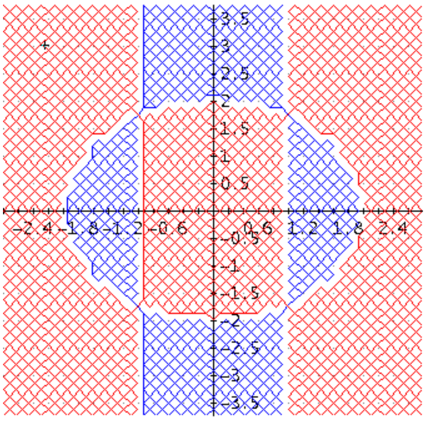
DERIVE has a step function `STEP(x)` = 1 if $x > 0$ and 0 otherwise. You can use this as a condition. Try `STEP(4 - x^2 - y^2) = 1` and it will plot the region inside a circle.

Now if you plot `STEP(4 - x^2 - y^2) * STEP(1 - x^2) = 1` then it will only shade the region with $-1 \le x \le 1$ which belongs to the circle with radius $r = 2$.

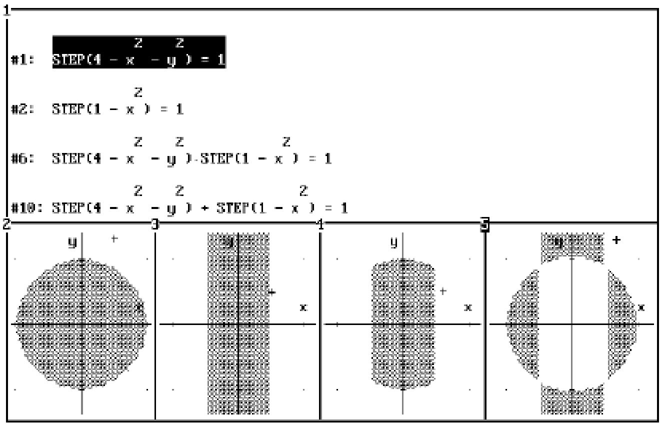
I tried and here are the results *(See first the respective DERIVE 6 plots)*:







```
STEP(4-x^2-y^2)=1
STEP(1-x^2)=1
STEP(4-x^2-y^2)=0
STEP(x^2+y^2-4)=1
"The intersection:"
STEP(4-x^2-y^2)*STEP(1-x^2)=1
"or"
STEP(4-x^2-y^2)+STEP(1-x^2)=2
"The XOR-Operation:"
STEP(4-x^2-y^2)+STEP(1-x^2)=1
"Plot the next two expressions in one Plot Window:"
STEP(4-x^2-y^2)-STEP(1-x^2)=0
STEP(4-x^2-y^2)+STEP(1-x^2)=1
"How to produce the union?"
```



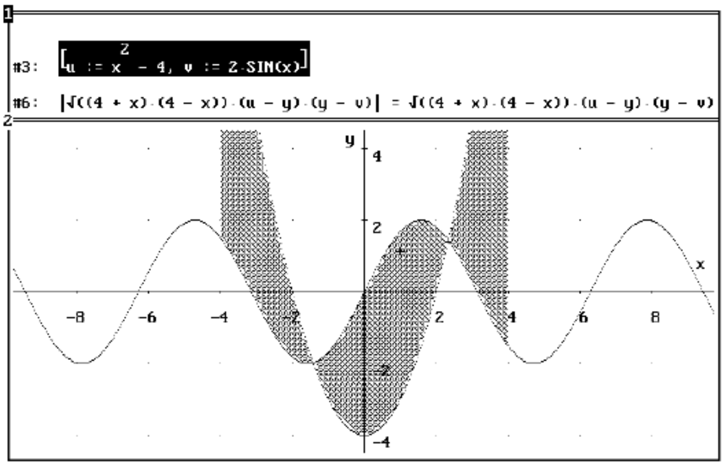
John’s STEP procedure demonstration

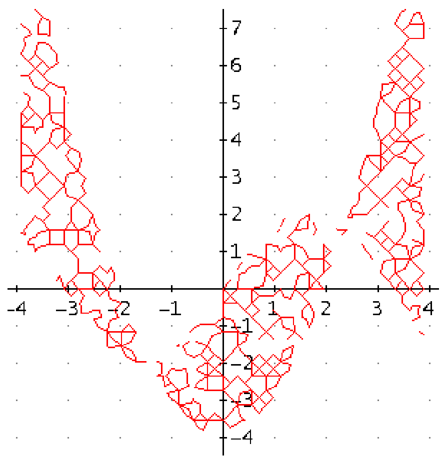
From Claudio del-Pino O.

You could try this:

`ABS(SQRT(x*(1 – x)*(u – y)*(y – v)) = SQRT(x*(1 – x)*(u – y)*(y – v)`

in order to obtain a grid between the curves, the *x*-axis form 0 to 1. I hope that this will work always.





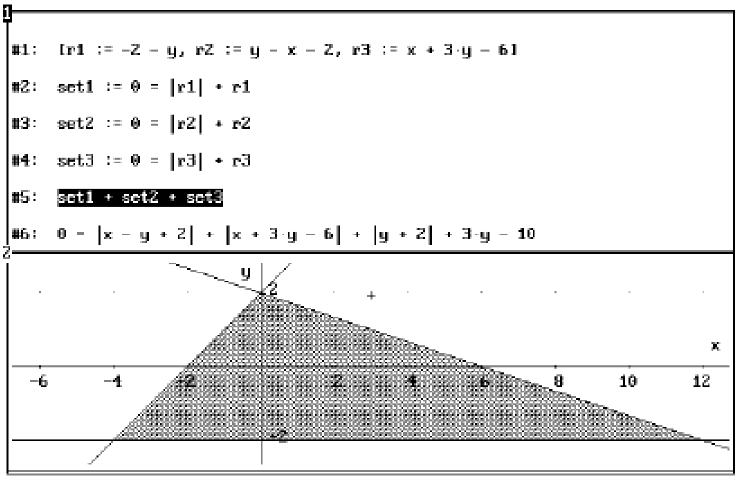
Compare DfD (left) and DERIVE 6 (right)!!

DERIVE 6 provides ready made functions for shading regions and it is easy to produce the grids by plotting appropriate inequalities. You can look at this on the next page.

This was Sebastiano´s last message and I was eager to check these suggestions, too. As you can see in another article in this DNL I was busy to visualize systems of linear inequalities and the tools to shade areas presented here are quite different from mine.

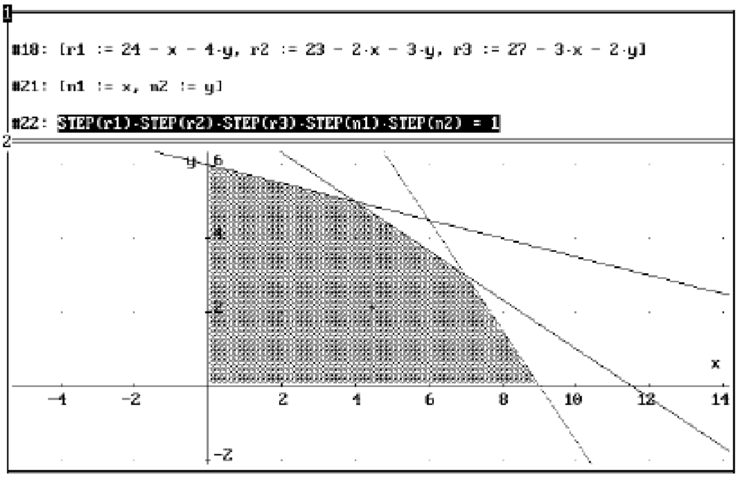
Which region in the x-y-plane is described by the following system of linear inequalities?

$$y \geq -2$$
$$x - y \geq -2$$
$$x + 3y \leq 6$$

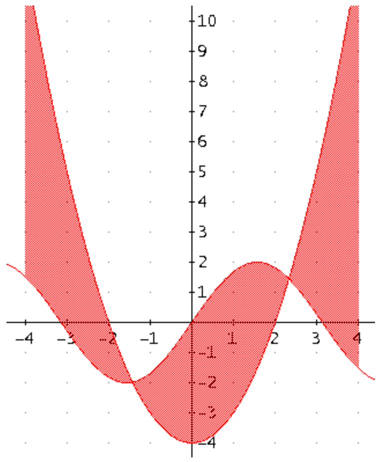


Find the region of feasible solutions considering the following restrictions (from Linear Programming):

$$x + 4y \leq 24$$
$$2x + 3y \leq 23$$
$$3x + 2y \leq 27$$
$$x, y \geq 0$$



Shading areas with DERIVE 6 using the implemented function and applying Boolean expressions (which is sometimes of more didactical value!)



```
AreaBetweenCurves(u, v, x, -4, 4, y)

[x^2 - 4, 2·SIN(x), 2·(x^2 - y - 4)·SIN(x) - y·(x^2 - y - 4) < 0 ∧ -4 ≤ x ≤ 4]

-4 ≤ x ≤ 4 ∧ (u ≤ y ≤ v ∨ v ≤ y ≤ u)
```

The SIMPLEX METHOD on the *TI*-92

Bruce Chaffee, CA, USA

This article is intended to introduce an implementation of the simplex method on the TI-92. For a more complete discussion of the method the reader is referred to INTRODUCTION TO MATHEMATICAL PROGRAMMING, by Winston.

The "Simplex Method" as presented here is often referred to as the "Big M method". The problem is to maximize an objective function subject to a set of constraints (see example below). The numbers on the right hand side of the constraints must be non-negative.

Without much difficulty, the non-negativity of the right hand side of the constraints can be overcome. The method can also be used to solve minimization problems (See example FOUR).

EXAMPLE ONE: $\leq$ CONSTRAINTS ONLY

Maximize $P = 60x_1 + 30x_2 + 20x_3$

Subj. To:

$8x_1 + 6x_2 + 1x_3 \leq 48$

$4x_1 + 2x_2 + 3/2x_3 \leq 20$

$2x_1 + 3/2x_2 + 1/2x_3 \leq 8$

$x_1, x_2, x_3 \geq 0$

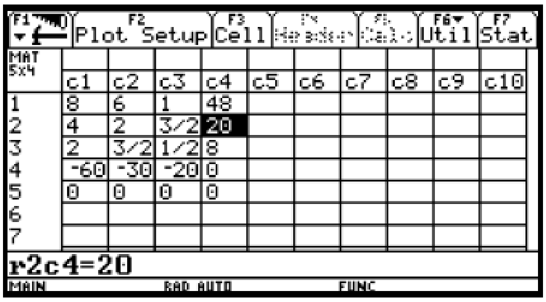
ENTER DATA INTO MATRIX AS FOLLOWS

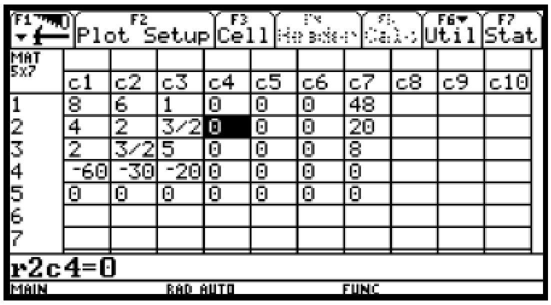
8	6	1	48
4	2	3/2	20
2	3/2	1/2	8
-60	-30	-20	0
0	0	0	0

To enter the problem, open a new matrix named MATrix EXample 1 (MATEX1) using "APPS 6, NEW, DATA ‘ MATRIX, VARIABLE MATEX1, 5 ROWS, 4 COLS. Once you get into the matrix editor you can (and will) change the number of rows and columns as you please.

Enter the coefficients of the constraints in exactly the same order as they appear on the page.

The objective function, on the other hand, is rewritten as $-60x_1 - 30x_2 - 20x_3 + P = 0$ and entered on the next-to-last line. The "P" term is in its own column, and never changes during the solution process, so it may be omitted.



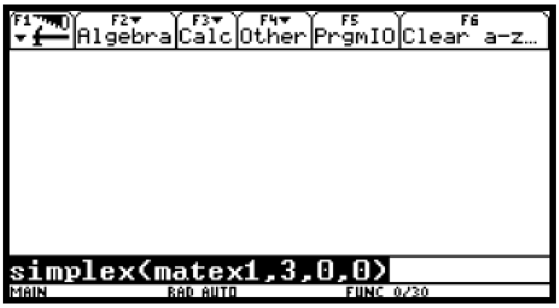


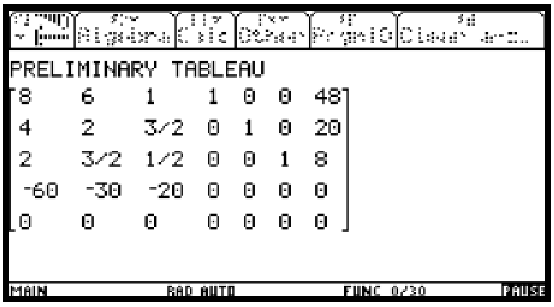
The row of zeros in the last line is required in all problems and will be used during the solution of problems with "$\geq$" or "=" constraints. For this first problem, which has only "$\leq$" constraints, the zeros will remain unchanged during the solution process.

To make room for the additional variables required for the solution, we use the rules:

For each $\leq$ constraint, add ONE column (a slack variable)
For each $\geq$ constraint, add TWO columns (a surplus variable and an artificial variable)
For each = constraint, add ONE column (an artificial variable)

Because this problem has 3 < constraints, we must add a total of 3x1 = 3 columns. Position the cursor in the last column and then use the keystrokes "F6, 1, 3" a total of 3 times to add the 3 columns. LEAVE THE COLUMNS BLANK. DO NOT ATTEMPT TO ADD IN THE ADDITIONAL VARIABLES. THE PROGRAM WILL DO IT AUTOMATICALLY.

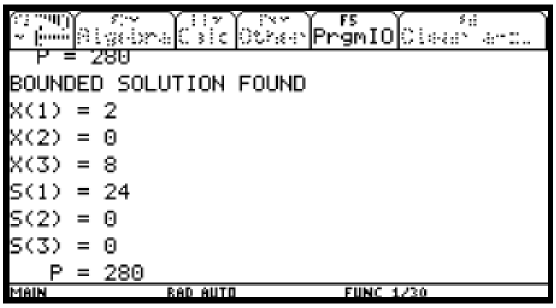


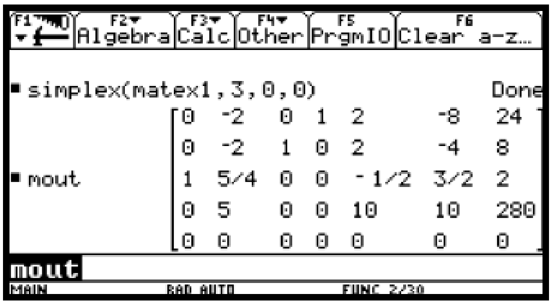


Now use "diamond", "home" to return to the home screen, where you can enter: simplex(matex1,3,0,0) and run the program. The parameters "3,0,0" tell the program that there are:

3 constraints of the "$\leq$" variety,
0 constraints of the "$\geq$" variety, and
0 constraints of the "=" variety,.

Press "ENTER" now to show the successive matrices (tableaux) as the problem is solved. When the bounded solution is found, all the variables (basic and non-basic) are displayed.





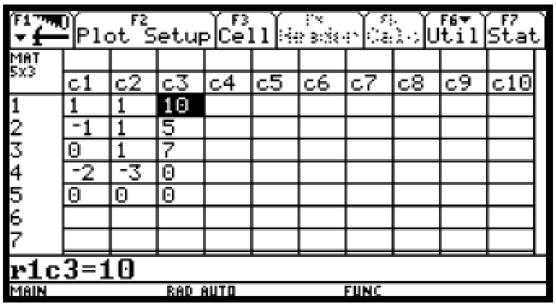
The final tableau will be stored in the variable named "mout" (Matrix OUTput). This allows you to save this matrix in a matrix variable for future reference, or simply display it on the home screen for further examination (See example FOUR).

EXAMPLE TWO: MIXED CONSTRAINTS

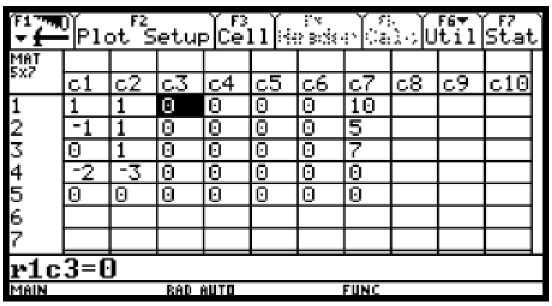
MAX P = $2x_1 + 3x_2$
Subj. To
$1x_1 + 1x_2 \leq 10$
$-1x_1 + 1x_2 \geq 5$
$0x_1 + 1x_2 = 7$
$x_1, x_2 \geq 0$

NOTICE THE CONSTRAINTS MUST BE ENTERED IN THE ORDER $\leq$, $\geq$, =.
(This is how the program knows which inequality is which type)

Load this into the matrix "matex2" as follows



Insert 4 additional columns as before

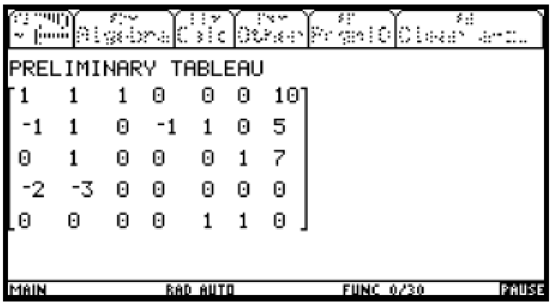


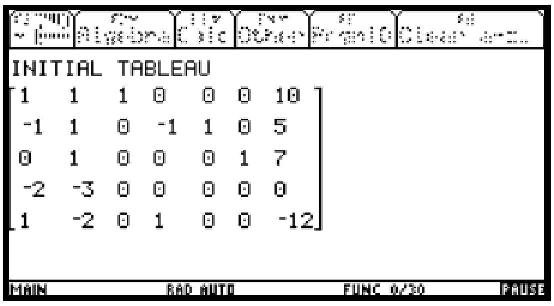
Because there is one $\leq$ constraint, one $\geq$ constraint, and one = constraint we must add 1x1col+1x2cols+1x1col=4 columns of zero's in the same manner as before (F_6, 1, 3).

Return to the home screen and enter: simplex(matex2,1,1,1)

After you press "ENTER", you will notice there are now ones in the last row of the PRELIMINARY TABLEAU. These represent an "M" in the row above. For example, a 1 in the last row, with a 0 just above it represents "0+M" in the objective function.

Press ENTER again to display the INITIAL TABLEAU. In the leftmost column, a "-2" in the next-to-last row, with a "1" below it represents a "-2 + M" as the coefficient of x_1 in the objective function.



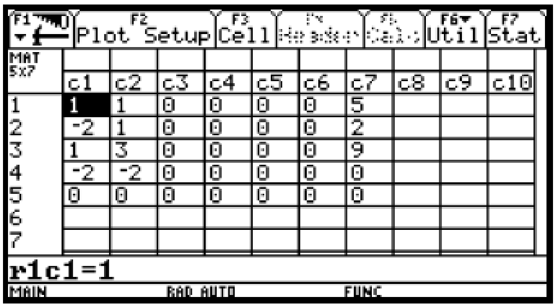


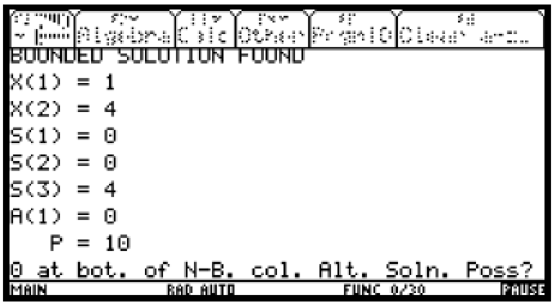
Continue pressing ENTER to step through to the solution.

When a bounded solution is found, the program checks to see if there is a "0" at the bottom of a column of any non-basic variable and alerts you. IF there is a positive value in this column, then you may pivot and see the alternate optimal solution. Because it is possible for there to be more than one such column, you are offered the option of selecting the column you want to explore - but you must be sure there is at least one positive value in the column.

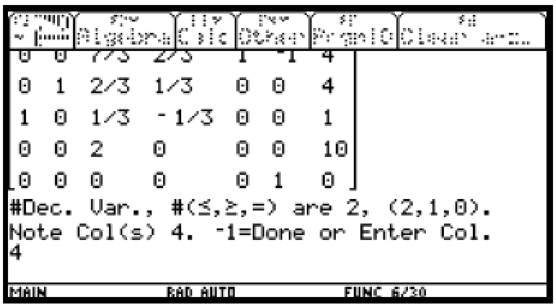
EXAMPLE THREE: MORE THAN ONE SOLUTION

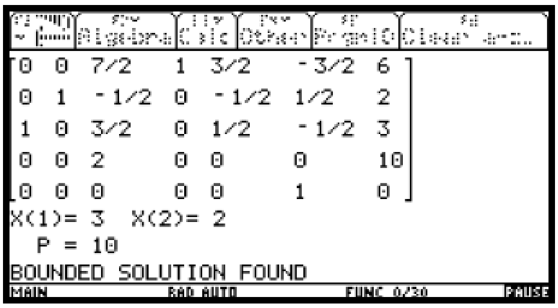
$$\begin{aligned} \text{MAX } P &= 2x_1 + 2x_2 \\ \text{Subj. To} \quad & 1x_1 + 1x_2 \le 5 \\ & -2x_1 + 1x_2 \le 2 \\ & 1x_1 + 3x_2 \ge 9 \\ & x_1, x_2 \ge 0 \end{aligned}$$





Enter the problem as before using a matrix name of your choice, such as "matex3".
Remember to enter the constraints in the order ≤, ≥, =.
You may then return to the home screen and run the program with "simplex(matex3,2,1,0)"





The first solution found is P = 10 @ (1,4). If you enter "4" to the prompt in the left screen above, the alternate solution is found P = 10 @ (3,2).

EXAMPLE FOUR: A MINIMIZATION PROBLEM

ORIGINAL PROBLEM

MINIMIZE $C = 48y_1 + 20y_2 + 8y_3$

subj. to

$8y_1 + 4y_2 + 2y_3 \geq 60$

$6y_1 + 2y_2 + 3/2y_3 \geq 30$

$y_1 + 3/2y_2 + 1/2y_3 \geq 20$

$y_1, y_2, y_3 \geq 0$

MODIFIED PROBLEM

MAXIMIZE $P = -48x_1 - 20x_2 - 8x_3$

subj. to

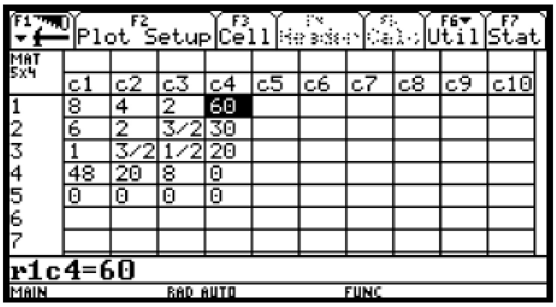
$8x_1 + 4x_2 + 2x_3 \leq 60$

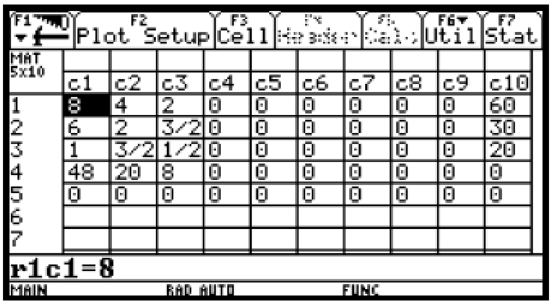
$6x_1 + 2x_2 + 3/2x_3 \leq 30$

$x_1 + 3/2x_2 + 1/2x_3 \leq 20$

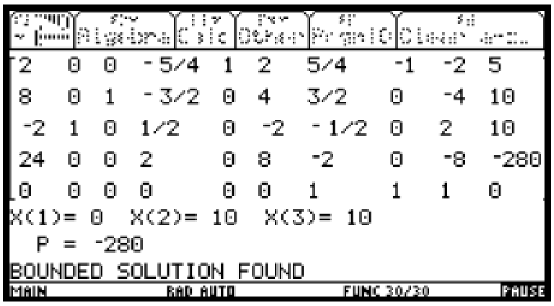
$x_1, x_2, x_3 \geq 0$

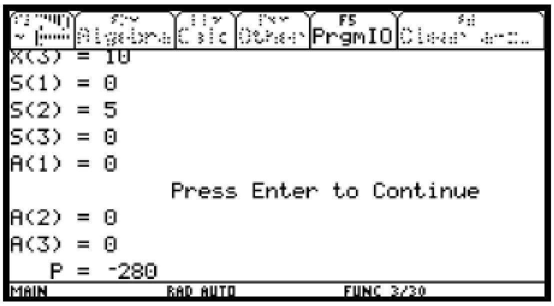
This minimum problem can be solved by finding the maximum of the MODIFIED problem, where the objective function of the modified problem is the negative of that of the original problem. The coefficients of the constraints remain the same. The MODIFIED problem is solved as follows:



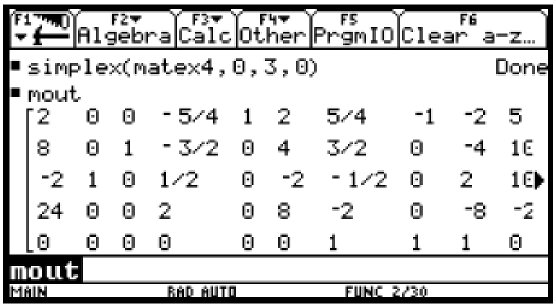


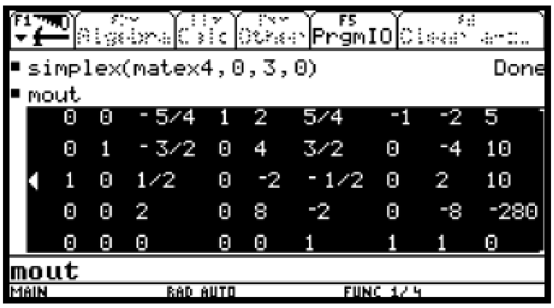
The three $\geq$ constraints require 2x3=6 additional columns. When solved, this yields the solution to the MODIFIED problem. P = -280 @(0,10,10). The solution to the ORIGINAL problem is then C = -P = +280 @(0,10,10).





When this final tableau is viewed on the home screen, the matrix is slightly too big to fit on the screen, but if you highlight the matrix, you can scroll left and right to see the entire matrix (tableau).



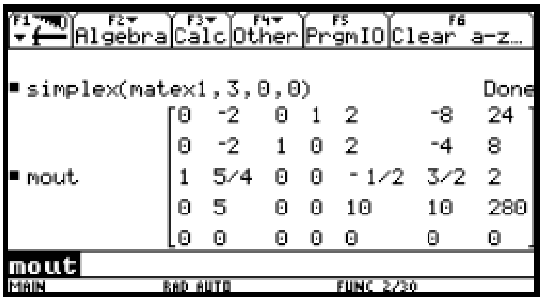


Example FOUR is, by the way, the DUAL of example ONE, which is why the value of the objective function (+280) is the same in both cases. Displaying the final matrices (using MOUT) for both problems on the home screen leads to a natural discussion of the DUAL THEOREM and SHADOW PRICES.

For example, in the left screen above, where column one is visible, notice the values in the last two rows.

In column 1 the bottom two values, "24" above a "0", stand for 24 + 0xM = 24,
in column 4 the bottom two values stand for 2+0xM = 2, and
in column 6 the bottom two values stand for 8+0xM = 8.

Compare these values to the solution to EXAMPLE ONE (P = 280 @(2,0,8), see the screen below).



Similar remarks can, of course, be made about values in the last two rows of the final tableau of example ONE, and the solution of example FOUR.

Questions related to sensitivity analysis can now be explored.
How much does the objective function change if ... ?
How much can a value change and still have the same basis remain optimal?

The program does not replace sensitivity analysis, but hopefully it will make it more understandable.

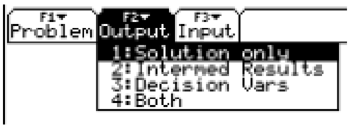
It is obvious that there is a lot of programming code copying necessary to have such a powerful tool. I do not expect that you are eager to read and to edit 6 full pages of code. So I offer to download the program from my school's homepage. And not only that. Linear Programming was and still is on our curriculum. The problem is that we often stop after having written down the goal function together with the restrictions but we refuse to start that boring transformation algorithm with more variables. Now we can use the TI for not too large, but for large enough problems.
So very soon the idea came up to add a comfortable input routine to avoid working with the Matrix Editor, although editing the matrix sometimes will be useful to make the process clearer.

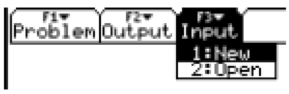
Follow my add-on surface for solving the problem below (A mixed Minimumproblem - - Blending problem)

Ein gemischtes Minimumproblem (Mischungsproblem)

$$
\begin{array}{rcrcrcrcl}
0{,}5x_1 & + & 0{,}8x_2 & + & 0{,}9x_3 & + & 1{,}2x_4 & \le & 6000 \\
0{,}75x_1 & + & 1{,}1x_2 & + & 0{,}1x_3 & + & 0{,}1x_4 & \le & 1800 \\
50x_1 & + & 50x_2 & + & 30x_3 & + & 40x_4 & \le & 200000 \\
 & & & & & & x_4 & \le & 1500 \\
x_1 & & & & & & & \ge & 200 \\
 & & x_2 & & & & & \ge & 300 \\
 & & & & x_3 & & & \ge & 450 \\
x_1 & + & x_2 & + & x_3 & + & x_4 & = & 5000 \\
10x_1 & + & 70x_2 & + & 120x_3 & + & 90x_4 & = & \text{Minimum}
\end{array}
$$

We call the program simp() and then we are offered a new menu:

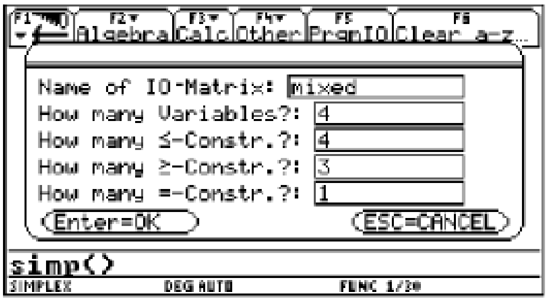


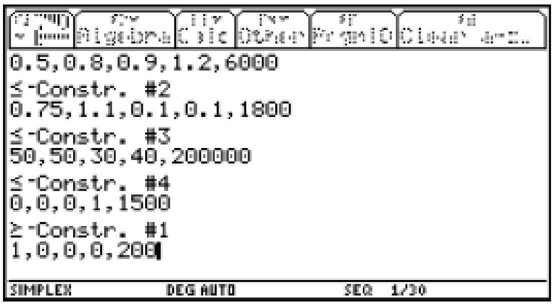


We choose Option 2: under • ; Option 1. under • •and we enter a 1:New *problem under • •*

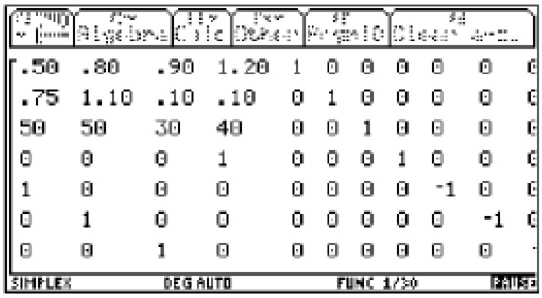
Then we fill in the data beginning with the coefficients of the restrictions and finishing with the coefficients of the goal function. Later on we will be asked to save or not to save the problem "mixed".

We enter the coefficients of the rows:

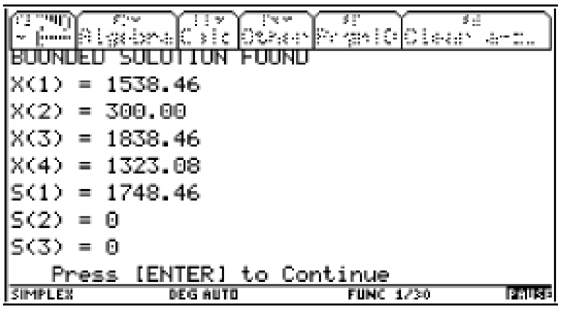




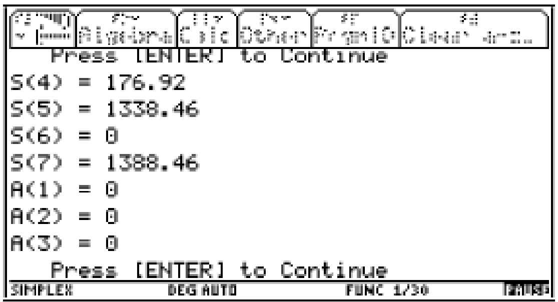
The initial tableau:........

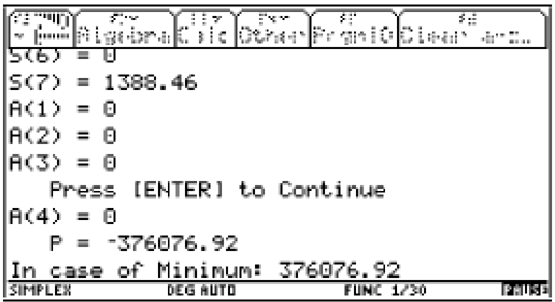


...... and the solution:



given part by part:





Now we can save the problem and load and demonstrate/calculate it next time using Option 2:Open *in menu point • •*Input*. We only have to know the problem´s name (mixed) and fix* ***Maximum*** *or* ***Minimum.*** *The parameters have been stored automatically in a list (mixedl).*

If you want to edit the matrix by hand then do it and call Bruce´s SIMPLEX-routine with one more parameter which fixes the kind of output. For the 4th parameter choose among ***n, m, d*** *or* ***b*** *for the 4 options of menu point • ; eg* SIMPLEX(mixed,4,3,1,n) *will immediately return the solution. Josef*

Some open questions for the TI-92

Some time ago I sent some questions to David Stoutemyer:

abs(x-5) | x < 5 simplifies to -(x - 5), but abs(2x-5) | x < 5/2 does not simplify?

It would be useful to switch from within a program to a data-table. Is that possible?

It is very useful to lock a program. But it makes no sense to lock a Text file (Script file). It cannot be opened and executed. It would be helpful to lock it against changes, but let it be executed.

It would be helpful to have one key (or key combination) to have a pause during plotting. Pressing the key should continue the plot-ptocedure. In programs I can help myself with a getkey() routine, but I think a key press would be useful.

David promised to look into all the problems.