THE DERIVE - NEWSLETTER #68

THE BULLETIN OF THE

USER GROUP

+ CAS-TI

Contents:

[1]   *Roman Hašek*, Užití Derive ve vỳuce Matematiky, Čĕské Budĕjovicĕ 2007,
      ISBN 978–80–239–9054–6

[2]   *Pat und Carl Leinbach/Josef Böhm*, Forensische Mathematik für den Unterricht,
      bk teachware SR-57, ISBN 978–3–901769–8

[3]   *Pavel Pech*, Selected Topics of Geometry with Classical vs. Computer Proving,
      World Scientific, Singapore, ISBN 978–981–270–942–4,

If you know about CAS-related publications then please inform me. Books dealing with the use of technology in math education in general are also interesting. Josef

TIME 2008 – Call for Papers

- 30 April 2008: Submission of abstracts on or before this date

- 31 May 2008: Acceptance of abstract notification by this date

- 30 June 2008: Early bird registration on or before this date

- 1 September 2008: Final date for registration and inclusion of abstract in the programme

- 10 September 2008: Final programme published on the web-page

- 22 to 26 September 2008: Conference

- 31 October 2008: Acceptance notification for selected papers for the SAJS.

Abstracts can be submitted as e-mail attachments to Josef Böhm (nojo.boehm@pgv.at) or Michel Beaudin (Michel.Beaudin@etsmtl.ca) for the Derive/Nspire part of the conference and for the ACDCA part to Bernhard Kutzler (b@kutzler.com) or Vlasta Kokol-Voljc (vlasta.kokol@uni-mb.si).

An online submission method is being developed. E-mail attachments need not be re-submitted when the online system is up and running.

The deadline for submission is 30 April 2008 with acceptance of notification by 31 May 2008.

You can find more important information about TIME 2008 at

http://time.tut.ac.za

Many thanks to Steve Joubert and his team in South Africa.

Dear DUG Members,

At the end of a year it is always time to give a report of our activities. You can find a summary of DUG-year #17 in the box below.

Despite the fact that distributing *DERIVE* was ended we can report a lot of activities.

First of all I´d like to draw again your attention to the forthcoming Conference in South Africa. Information can be found on the website and on our Information page.

This DNL contains less contributions as usual. But you will hopefully see that they are very interesting.

There is the announced article provided by Bernhard Kutzler who demonstrates the interplay of various applications of the TI-Nspire CAS. We hope that this contribution will inspire and incite other Nspire users to share with us their – good or bad – experiences with this CAS. It has now been a considerable time on the market and we can expect one or the other reaction.

The second article is a result of a cooperation between Pat and Carl Leinbach and me. Pat and Carl gave a workshop on Forensic Mathematics at DES-TIME 2006 in Dresden. I was so fascinated by their examples that I suggested to produce a German translation adapted for the use in school. and bk teachware promised to publish the booklet. Pat and Carl provided additional activities for this collection. One of them deals with AHP

(Analytical Hierachy Process) which is a great application of matrix calculation. We, Pat, Carl and I, are very grateful for bk-teachware´s permission to present this chapter – in English – in this DNL. At the other hand this might be a mouth watering for the German readers to learn more about Forensic Mathematics (see our Book Shelf).

In the last DNL I announced an article about Gröbner Bases. Alessandro Perotti sent his paper just in time. I added two informative examples, which were accepted by Alessandro. Many thanks for your reliable cooperation.

The User Forum shows among other questions a fine programming competition.

Just recently my wife Noor and I spent three weeks in India (Rajastan). Many palaces and temples were decorated with impressive patterns. I could add many pictures to my collection of backgrounds for *DERIVE*. You can find some of them on the last page. If you are interested in one or the other picture, then let me know. I´ll send it.

Finally we will wish you all a Merry Christmas and a Happy and Healthy New Year 2008. We hope to meet many of you 2008 in South Africa.

DUG Report 2007-12-21

2007 was a sad year for DERIVE enthusiasts because DERIVE was ended in many countries. At the same time the DERIVE and CAS-TI User Group was asked by TI for cooperation to make the TI-Nspire a successor of DERIVE. DUG members participated at the first "Transition Conference" which gave the opportunity to meet the Nspire-team to deposit the wishes of the DERIVIANS for future releases of the Nspire with a special impact on didactical issues.

But there are a couple of good news for 2007, too:

- We could welcome 37 (!!) new members in the DUG
- We support organizing TIME 2008 in South Africa
- We could set up a lot of contacts between DUG-members who had special requests
- We received high quality contributions for the DNL
- DUG members were involved in national and international projects
- DUG members attended national and international conferences giving lectures and workshops
- 4 revised and 4 new DERIVE & CAS-TI Newsletter were published
- DUG members reached high positions in universities

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

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

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

Next issue: March 2008
Deadline   15 February 2008

Preview: Contributions waiting to be published

Some simulations of Random Experiments, J. Böhm, AUT, Lorenz Kopp, GER
Wonderful World of Pedal Curves, J. Böhm
Tools for 3D-Problems, P. Lüke-Rosendahl, GER
ANOVA with *DERIVE & TI*, M. R. Phillips, USA
Financial Mathematics 4, M. R. Phillips
Hill-Encription, J. Böhm
Farey Sequences on the *TI,* M. Lesmes-Acosta, COL
Simulating a Graphing Calculator in *DERIVE*, J. Böhm
Henon & Co, J. Böhm
Do you know this? Cabri & CAS on PC and Handheld, W. Wegscheider, AUT
An Interesting Problem with a Triangle, Steiner Point, P. Lüke-Rosendahl, GER
Overcoming Branch & Bound by Simulation, J. Böhm, AUT
Diophantine Polynomials, D. E. McDougall, Canada
Graphics World, Currency Change, P. Charland, CAN
Precise Recurring Decimal Notation, P. Schofield, UK
Cubics, Quartics – interesting features, T. Koller & J. Böhm
Logos of Companies as an Inspiration for Math Teaching
Exciting Surfaces in the FAZ / Pierre Charland´s Graphics Gallery
BooleanPlots.mth, P. Schofield, UK
What is hiding in Dr. Pest? B. Grabinger, GER
Truth Tables on the TI, M. R. Phillips
Advanced Regression Routines for the TIs, M. R. Phillips
Where oh Where is IT? (GPS with CAS), C. & P. Leinbach, USA
Embroidery Patterns, H. Ludwig, GER
Inspheres and Exspheres of a Tetrahedron, D. Sjöstrand, SWE
Mandelbrot and Newton with *DERIVE*, Roman Hašek, CZ
Snail-shells, Piotr Trebisz, GER
A Conics-Explorer, J. Böhm, AUT
An Optimization Problem – MS Excel vs CAS, J. Böhm
Compléments de mathématiques (profil Génie Mécanique), M. Beaudin, CAN

        and others

Volker Loose   volker@VWLOOSE.DE

Hello Derive users,

I entered the following expression 2x + 0.5y + 4z = -2 and performed a multiplication by 3.
See what happened:

```
#1:    2·x + 0.5·y + 4·z = -2

#2:    (2·x + 0.5·y + 4·z = -2)·3

        3·(4·x + y + 8·z)
#3:    ───────────────── = -6
               2
```

This is correct, but if approximating #2 I get

```
#4:    4·x + y + 8·z = -4
```

Has anyone an explanation for this strange behaviour?

Joe Frisbee   joe.h.frisbee@usa-spaceops.com

Here is what appears to be happening. Derive 6 does some automatic simplification when using
"Approximate". The second result given is correct but is obtained by multiplying the first result by
two in order to eliminate the fraction. This can be seen by selecting, from the drop down menus,
"Options/Mode Settings/Simplification" and then at the bottom of the widow select "Display Steps".
When "Approximate" is used to reduce (2·x + 0.5·y + 4·z = -2)·3 the missing step and its
explanation will be displayed. I am new to using Derive 6 so I do not know if it is possible to turn off
this feature.

```
#5:    Precision ≔ Approximate

#6:    Notation ≔ Decimal

#7:    DisplaySteps ≔ true

#8:    (2·x + 0.5·y + 4·z = -2)·3

                   y
n·x = y  →  x = ───
                   n

#9:    4·x + y + 8·z = -4
```


Andreas Lindner, Austria

I wanted to demonstrate the influence of parameters in the equation of a plane using slider bars in the
3D-Plot Window, eg *x* + 2*y* + 3*z* = *a* or *a x* + *b y* + *c z* = 4. This does not work.

DNL.: *DERIVE is unable to plot implicitly given expressions in the 3D-Plot Window. Solve the equations for z and try once more*

$$\#3: \quad z = -\frac{x + 2\cdot y - a}{3}$$

$$\#4: \quad z = -\frac{a\cdot x + b\cdot y - 4}{c}$$

Reducing a Matrix – A Chronology of emails

The Problem:

Francisco Marcelo Fernández, Argentina   `fernande@quimica.unlp.edu.ar`

Dear derivers,

I have a two-column matrix. In the left column there are some repeated elements that correspond to different elements in the right column. I want to replace the repeated items in the left column by just one element and the corresponding different elements in the right by its average. Does anybody have a simple program for doing it?

Example: reduce `[1,2;1,2.1;1,2.2;2,3;2,3.2]` to `[1,2.1;2,3.1]`

Regards,
Marcelo

John Coleman   `jcoleman@GWIA.FRANCISCAN.EDU`

The (hopefully) attached file contains a function which (hopefully) works. It returns the correct answer for your example. It is a naive algorithm that makes several passes over the data. A more efficient approach is possible of course, if efficiency is an issue.

Hope that helps

-John Coleman

m_reduce_01.dfw

```
        add_repeats(i, A) ≔
          If A = []
#1:           0
            IF(A↓1↓1 = i, A↓1↓2, 0) + add_repeats(i, REST(A))

        count_repeats(i, A) ≔
          If A = []
#2:           0
            IF(A↓1↓1 = i, 1, 0) + count_repeats(i, REST(A))
```

$$\text{\#3:}\quad \text{average\_repeats}(i, A) ≔ \left[i, \frac{\text{add\_repeats}(i, A)}{\text{count\_repeats}(i, A)} \right]$$

```
#4:     average_all_repeats(A) ≔ VECTOR(average_repeats(i, A), i, A↓↓1)

        remove_dups(v) ≔
          If v = []
              []
#5:           If MEMBER?(FIRST(v), REST(v))
                 remove_dups(REST(v))
                 ADJOIN(FIRST(v), remove_dups(REST(v)))

#6:     reduce_matrix(A) ≔ remove_dups(average_all_repeats(A))
```

$$\text{\#7:}\quad \text{reduce\_matrix} \begin{bmatrix} 1 & 2 \\ 1 & 2.1 \\ 1 & 2.2 \\ 2 & 3 \\ 2 & 3.2 \end{bmatrix} = \begin{bmatrix} 1 & 2.1 \\ 2 & 3.1 \end{bmatrix}$$

A 'Holistic' Approach to Solving an Optimization Problem with TI-Nspire CAS

by B Kutzler

TI-Nspire CAS combines several applications under one roof: A computer algebra system, a graphing software, a dynamic geometry system, a spreadsheet, a text editor, and a data & statistics package. This makes it an ideal tool for supporting the teaching and learning of topics such as optimization problems. Optimization problems can be solved geometrically, graphically, numerically, or algebraically. By showing and comparing all four approaches in parallel, students can better understand the connections. TI-Nspire CAS is ideal for that.

⧗

We look at the following optimization problem: *Given an isosceles triangle with base length 'a' and height 'h'. Inscribe a rectangle such that its base lies on the base of the triangle. Among all possible rectangles find the one with the largest area.*

⧗

We start by drawing an isosceles triangle in the Geometry application. The general **Triangle** tool is not appropriate to begin with, because one cannot enforce that two sides are equally long. Therefore we construct it as we would do with pencil, ruler, and compass. We draw a segment, construct its perpendicular bisector, and then choose the segment end points and a point on the perpendicular bisector as triangle points.
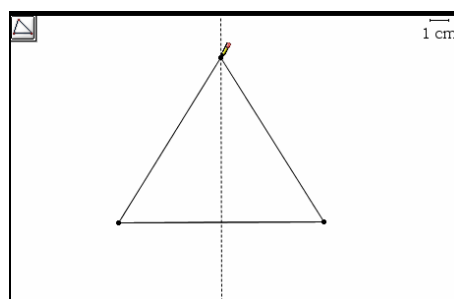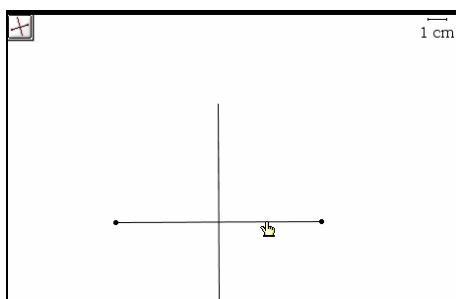

Since the triangle is symmetric with respect to the perpendicular bisector of its base and since the rectangle's base should lie on the triangle's base, the rectangle will have the same symmetry. Therefore, again, the **Rectangle** tool is not appropriate for drawing the rectangle, we need to construct it. We choose a point on the triangle base between the left vertex and the base's midpoint, and then we reflect that point at the perpendicular bisector. The point and its reflection make the base of the rectangle. In these two points we draw lines perpendicular to the triangle's base. The intersections of these perpendicular lines with the two other triangle sides give the other two rectangle vertices.
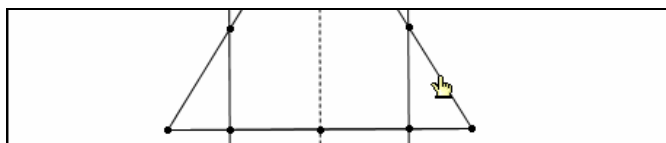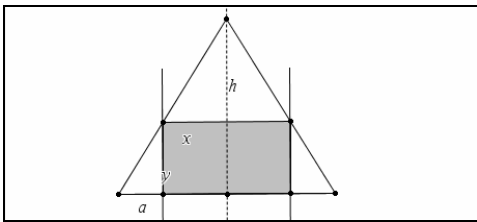

Since for the optimization problem we need to measure the area of the rectangle, we need to create the rectangle as an object. We use the **Polygon** tool, which normally would be used for drawing an arbi-
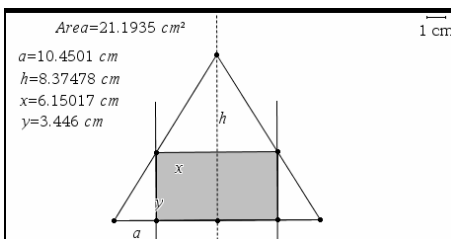
trary non-regular polygon. But since the rectangle properties already were enforced, we just use it to group the four points into a new geometrical entity.
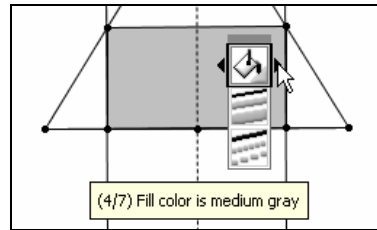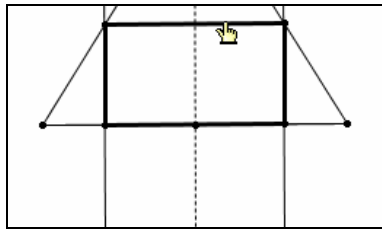
We add appropriate labels *a*, *h*, *x*, and *y* to the respective parts of the triangle and the rectangle.

We are ready to measure the lengths of a, h, x, and y, as well as the area of the rectangle.

Now we have everything we need to find a numeric solution to the optimization problem geometrically. We grab the left base point of the rectangle, and then move it. We look at the area, and then we find a position for the point at which the area is maximal.
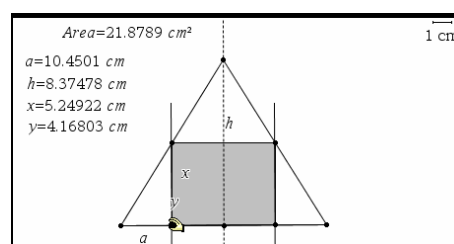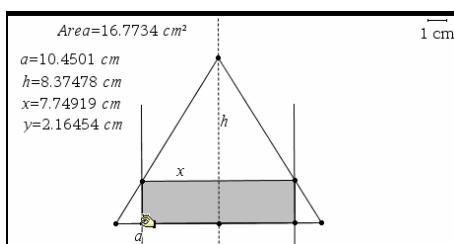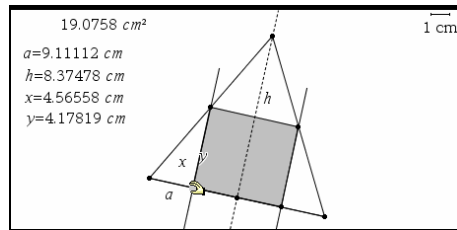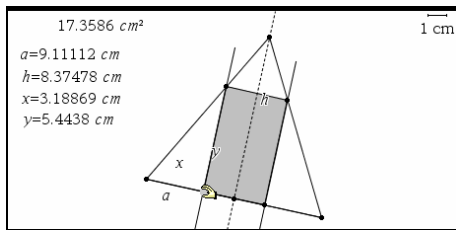
We take a note of all the numbers, or, better, we produce a printout of this page using the File>Print command. Next we create a "new" triangle by moving the triangle's left vertex.
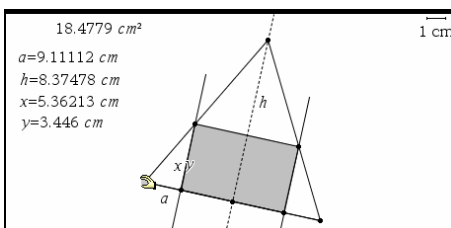
Again we move the rectangle's left base vertex to find the rectangle with maximum area for this triangle.

Again we note the resulting numbers. In the classroom we should do more rounds with more triangles, and then let the students look at the values of *a*, *h*, *x*, and *y* for a pattern.

Now is a good time to tackle this problem algebraically. We prepare for this by moving the triangle into the lower left corner, so that we have more room at the right for a Calculator window.
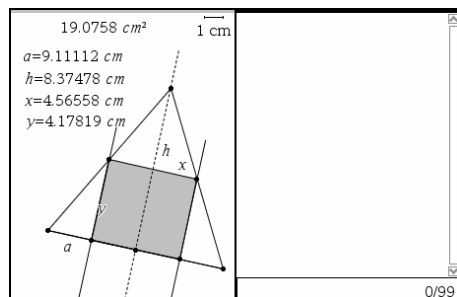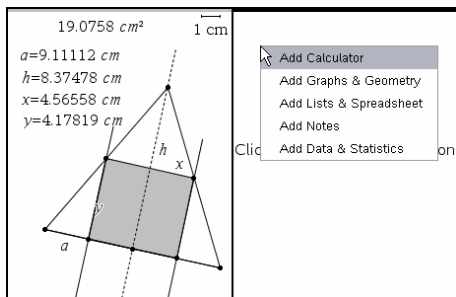
We are ready now to solve the optimization problem algebraically. We enter the function of the rectangle's area, which is the function to be maximized. We also enter the constraint on the variables which stems from the geometric construction.

$$area(x,y):=x \cdot y \qquad Done$$

The constraint on x and y is given by the fact that the upper rectangle points lie on the triangle sides. The small white rectangular triangle on the left with sides y and $\dfrac{a-x}{2}$ is similar to the "left half" of the original triangle with corresponding sides *h* and $\dfrac{a}{2}$. Using the theorem of proportional segments this gives the following constraint on *x* and y: $y : \dfrac{a-x}{2} = h : \dfrac{a}{2}$.

$$constr := \frac{y}{\dfrac{a-x}{2}} = \frac{h}{\dfrac{a}{2}}$$
$$\frac{-2 \cdot y}{x-a} = \frac{2 \cdot h}{a}$$

We solve this equation for *y*.

$$solve(constr, y)$$
$$y = \frac{-h \cdot (x-a)}{a}$$

We turn the solution into an assignment for '*y*'.

$$y := \frac{-h \cdot (x-a)}{a} \qquad \frac{-h \cdot (x-a)}{a}$$

We look at the rectangle's area, which now should depend only on the variable x (and parameters a and h).

$$area(x,y) \qquad \frac{-h \cdot x \cdot (x-a)}{a}$$

We differentiate this expression with respect to x, equate the resulting expression to zero, and then solve for x.

$$\frac{d}{dx}(area(x,y)) \qquad \frac{-h \cdot (2 \cdot x - a)}{a}$$

$$solve\left(\frac{-h \cdot (2 \cdot x - a)}{a} = 0, x\right)$$
$$x = \frac{a}{2} \text{ or } \frac{h}{a} = 0$$

We turn the "useful" part of the solution into an assignment for 'xm'.

$$xm := \frac{a}{2} \qquad \frac{a}{2}$$

We look at y when x takes the value xm. For this we use local substitution by applying the With operator.

$$y|x=xm \qquad \frac{h}{2}$$

We save this as ym. (We could have combined both steps by entering: 'y:=ym|x=xm')

$$ym := \frac{h}{2} \qquad \frac{h}{2}$$

We compute the maximum area.

$$area(xm,ym) \qquad \frac{a \cdot h}{4}$$

Next we use this formula to check if indeed we found the maximum in the Geometry window. For this we make the Geometry window's measurements available in the Calculator by performing appropriate store operations.
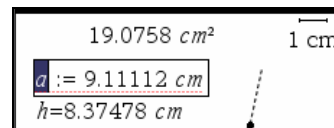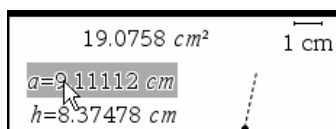
If we store this value in a, then a no longer would be a parameter in our TI-Nspire document and the calculations in the Calculator would loose (some of) their algebraic nature. Therefore, we should change the name to indicate that this is a special value for a. We use 'aa'.
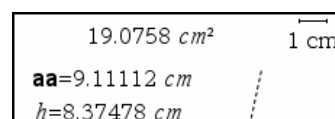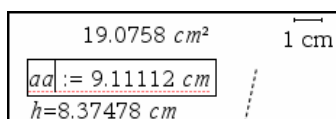
We apply the same procedure to the other three measurements.
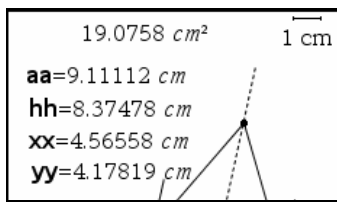
The current values of aa, hh, xx, and yy now are available in the calculator.

This number is very close to the maximum area which we found in the Geometry window. Need we check if our geometric *x* and *y* are close to the theoretical results, *a*/2 and *h*/2.

Yes, they are! With our students we may want to change the triangle again, find the maximum rectangle, and then see how "theory" and "practice", i.e. algebra and geometry again fit together.

⏳

More can be done! Let's take the numbers from the Geometry window and visualize them in a scatter plot. The two most interesting numbers for this are 'xx', the length of the rectangle base (which is controlled via moving the left rectangle base point), and the area. We still have to assign a name to the area so that we can use also this value in other applications. We call it 'ar'.

We need a Graphing window for a scatter plot – and we need a Lists&Spreadsheet page to collect data from a Geometry window and arrange it as a list to be used for the scatter plot. Currently we have one page which is split into a Geometry window and a Calculator window.

We want to use two pages for that. The first page should show Geometry and Graphing, the second page should show Calculator and Lists&Spreadsheet. This is done easily. We cut the Calculator (with $(Ctrl)+(X)$ after selecting the entire application with the **Edit>Select Application** command), and then add a Graphs&Geometry application which we leave in Graphing View. Then we add a new page, split it into two subpages, paste the Calculator into the left half, and then add a List&Spreadsheet application in the right half.
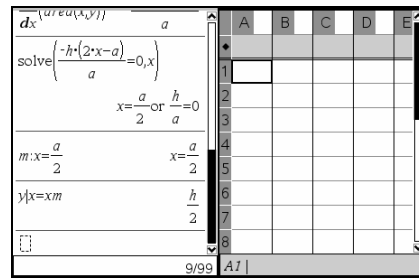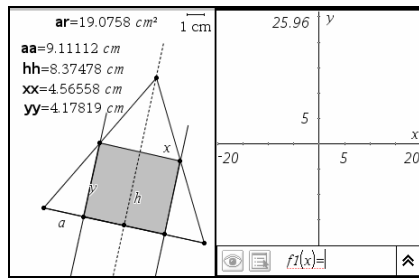
All this is done with only a few mouse clicks!

Next we prepare the spreadsheet for collecting data from the Geometry window. Column A shall carry 'xx', the length of the rectangle base, and column B shall carry the rectangle's area. We use the **Automatic Data Collection** command, and then we replace the placeholder 'var' by the variable for the rectangle base length, which is 'xx'.
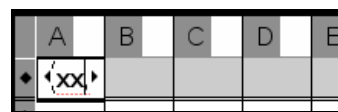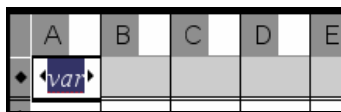
We repeat this procedure to collect the rectangle's area in column B.
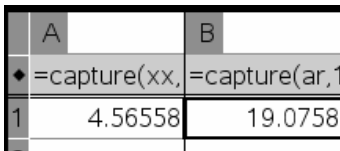
When we move the rectangle vertex again, columns A and B automatically will be filled with the respective values for 'xx' and 'ar'. But before doing so we will prepare for the values in columns A and B to be visualized as a scatter plot. This requires to give names to these two columns so that their contents will be available in other applications. We choose names 'base' and 'r_area'. (Note that the name 'area' is already used in the Calculator for the function!)
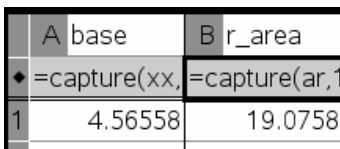
In the Graphing window we change the **Graph Type** to **Scatter Plot**. This changes the appearance of the function editor entry line.
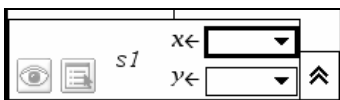
From the selection menus we choose **base** for *x* and **r_area** for *y*.
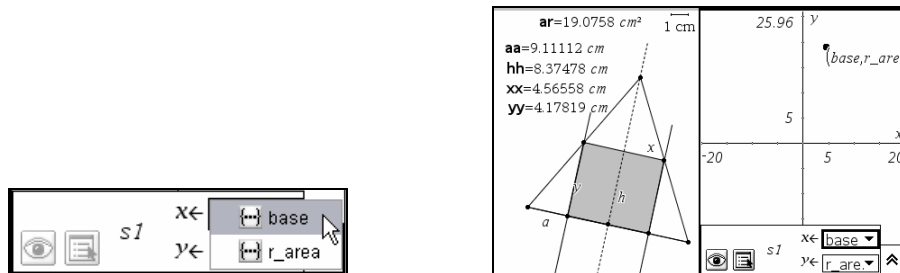
Now we have everything together. We make the Geometry window active, grab the rectangle's left base point, and then move it.
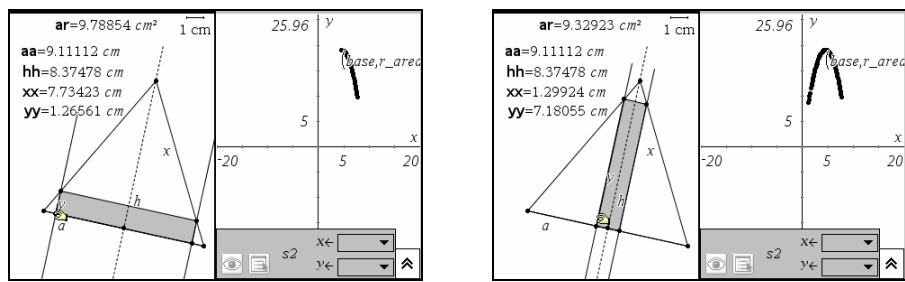
We change into the Graphing window, and then zoom in on the scatter plot with the **Window** menu's **Zoom – Data** command.
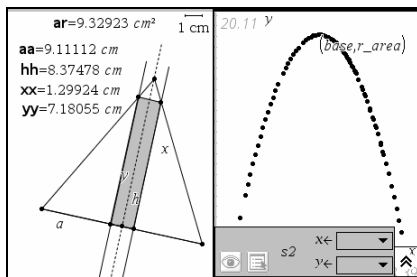
Finally we demonstrate how numbers and algebra fit together. We look at the area function again, …

$$area(x,y) \qquad \frac{-h \cdot x \cdot (x-a)}{a}$$

… and then produce a specific area function for the current values of *a* and *h* (which are available in 'aa' and 'hh'). This can be achieved using the With-operator. We enter: area(x,y)|a=aa and h=hh

$$area(x,y)|a=aa \text{ and } h=hh$$
$$-.919183 \cdot x \cdot (x-9.11112)$$

The quickest way to plot this function's graph in the Graphing window is to save this expression as f1(x), …

$$f1(x):=-.91918281247769 \cdot x$$
$$Done$$

… change into the Graphing window, change the **Graph Type** to **Function**, in the entry line scroll to **f1**, …
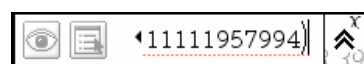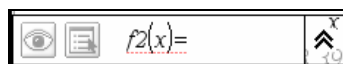
… and then use the 'Enter'-key to plot the graph of **f1**.
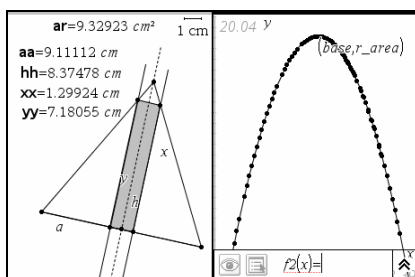
The essence of this approach is to use the data produced in the Geometry window in all other applications and see and study the connections between geometry, algebra, and graphing.

Directing Our Suspicions

Using Matrix Algebra to Organize the Results of an Investigation

Pat and Carl Leinbach

Scenario

You are called by the county police to investigate a shallow grave that has been located by two hunters at the edge of the county. You cordon off the scene and carefully uncover the remains that are buried in the grave. It is obvious that the body was not buried recently. In fact all that remains is essentially bones and some shreds of fabric. It is apparent to your experienced eye that the victim was a young female, but that is all that you can determine at the scene of the grave. You assist in a careful exhumation of the body and direct that it be taken to the county morgue.

Returning to the morgue, you do a more precise analysis. From the length of the bones and an examination of the bone ends you determine that the victim is a pre-pubescent female. A forensic dental examination reveals that the victim is a young girl who was reported missing over 20 years ago. Further analysis reveals that the victim was shot at least three times in the area of the chest. No bullets have been found on the scene, but the extent of damage to the bones indicates that the weapon was on the order of a .45 caliber pistol. You are unable to make a more exact estimate concerning the weapon.

At this point your involvement in the investigation is over. Solving the crime is the job of the police. A team of detectives is assigned to the case and they begin the tedious job of interviewing people who knew the victim and also others who had contact with her. The town where the victim lived is a small town, so the detectives begin there. They also review the local police's case notes. The prime suspect at the time of the disappearance was an African-American 14-year old boy who was the son of the girl's neighbors. His family was the only non Caucasian family in the neighborhood. The boy's sister was the girls "best friend"; however, the victim and his sister had just had a small falling out that had seemed to have been orchestrated by the victim's mother.

The Police Investigation

The police began by questioning the now grown boy and his sister. As is the case in so many investigations, the questioning revealed other individuals who needed to be visited. From their interviews, the police have narrowed the suspect list down to six individuals. Here is a very brief summary on the police findings. We will identify the suspects by alphabetical letters, A through F.

A. The African-American boy. Sister, of whom he is very protective, had just received a "rejection" from the victim; Had been seen on the victim's property the night of the disappearance; Determined to have been a fairly normal teen age boy, but resentful of the discrimination his family had to endure.

B. African-American girl. Had been a friend of the victim from the moment her family moved next door to the victim's family amidst a demonstration by the local "citizen's committee for a pure neighborhood"; Had recently had a childish falling out with the victim; Had written a note to the victim the day of her disappearance asking to meet in their "secret place."

C. Leader of the citizen's committee that lead a demonstration greeting the moving van when the African-American family moved in; Was very offended by the victim being good friend and confidant of suspect B; Quoted as saying "The only thing worse than a black is a white person who befriends one"; Known to be a stern father and husband. Known to have kept a gun in his home.

D. Real Estate agent trying to capitalize on the panic that can result from having a non Caucasian family move into the neighborhood; Was trying to buy homes cheap and sell houses in his new development on the edge of town; Perceived the public friendship of the victim and suspect B as "bad for business."

E. Teen aged boy. Son of suspect C; Prejudiced, but confused, especially by the friendship of the victim and suspect B; Had been publicly rebuked and slapped by the victim when bullying her at school; Apologized to the victim and her friend and was, in return, severely disciplined by his father. He was constantly trying to please his father.

F. Mother of victim. Was entranced by suspect D's attempts to sell her a new home, but opposed by her husband; Insisted that her daughter stop seeing suspect B; Made her daughter ignore her friend in public.

Each of the above suspects seems to have a reason for being placed on the list of suspects to the crime. The job for the detectives is to sort out these reasons and determine on whom to concentrate their efforts.

An old rule of thumb in criminal investigations is to look at three pointers: Motive, Opportunity, and Profit. Who had the most powerful motive for killing the victim? Who had the best opportunity to commit the crime? Who would profit the most from the crime? We will abbreviate these pointers as MOP. In the following investigation, we will apply the MOP pointers to the crime at hand.

Investigative Objective

- Organize investigative findings in a sensible way that will lead to the solution of the crime.

Mathematical Objectives

- Find a way to sensibly incorporate subjective evidence into a meaningful quantitative analysis.
- Assign a priority ranking and strength to the various alternatives for solving the crime.
- Using a two way comparison of alternatives to achieve the ranking.
- Learn about the concept of a dominant vector related to certain matrices.

Preliminary Mathematical Discussion

In this section we will introduce the idea of a dominant vector which can be associated with certain types of matrices. At the moment, this may seem like an irrelevant mathematical diversion, but it really has a lot to do with our attempt to solve the crime; however, the idea of a dominant vector (more properly called a dominant eigenvector) is the basis of the *Analytic Hierarchy Process* invented by the mathematician Thomas Saaty in the late 1970's. This process is a way of prioritizing several alternatives into a logical sequence of choices.

We will illustrate the idea of a dominant vector while working with a 2×2 matrix. We will assume that every entry in the matrix is positive and that the second row is not a multiple of the first row. For our example matrix we will use the matrix,

$$A = \begin{bmatrix} 7 & 8 \\ 1 & 3 \end{bmatrix}.$$

We will look at the effect of multiplying the matrix consisting of the corners of the following square by the matrix, A (from left hand side).
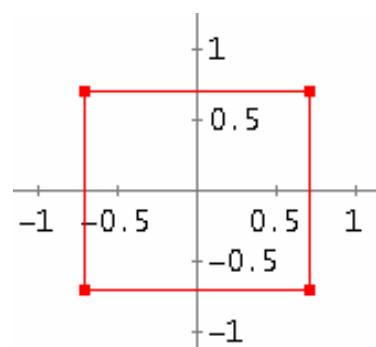
Since multiplication by the matrix A is a linear process, the lines joining the corners will be transformed into lines joining the corners of the images of the corners. Also any point in the interior of the square will be transformed into the interior of the image of the square. The square can be created by joining the following coordinate points in order:

$$Q := [-\sqrt{2}/2, -\sqrt{2}/2;\ -\sqrt{2}/2, \sqrt{2}/2;\ \sqrt{2}/2, \sqrt{2}/2;\ \sqrt{2}/2, -\sqrt{2}/2;\ -\sqrt{2}/2, -\sqrt{2}/2]$$

We multiply each of the points by A and normalize the resulting vector by dividing it by its length.

$$\text{VECTOR}\left(\frac{A \cdot v}{|A \cdot v|},\ v,\ Q\right)$$

The result is shown as the blue rectangle superimposed on the original red square. Note that the rectangle is now slanted in a particular direction and is a compressed form of the original.
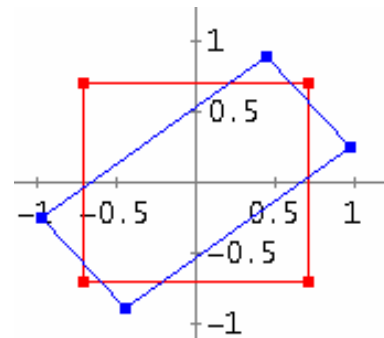
Next multiplying the corner points of the blue rectangle by A or doing the same thing by multiplying the corner points of the red matrix by A^2, we see a rectangle with a slightly different slant and even more compressed. This rectangle is shown as a green rectangle.
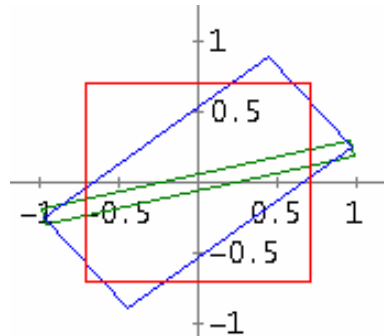
This process can repeated as often as you like. We show the result after 20 steps multiplying by A^{20} and plot the resulting "rectangle" (black).
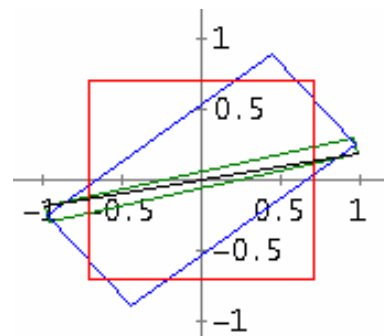
At this stage we have almost reduced the square to a single line. In fact, if we continue taking higher powers of A times the coordinates points of the square, we see that the square is reduced to a straight line segment lying on the line $y = 0.18301x$ and passing point D (0.8453 | 0.1547).

Exact calculation delivers the coordinates of point D: $\left(\dfrac{6-2\sqrt{3}}{3}, \dfrac{2\sqrt{3}-3}{3}\right)$.

We choose the vector **d** (position vector of D) lying on the line and having the property that its components sum to 1, and call it the **_dominant vector for the matrix_, A**. The idea of a **_dominant vector_** will play an important role in our analysis of the evidence collected by the police during their investigation of the suspects in the murder.

It is important to note that not every matrix has a dominant vector. For example, the matrix

$$B = \begin{bmatrix} \dfrac{\sqrt{3}}{2} & -\dfrac{1}{2} \\[2mm] \dfrac{1}{2} & \dfrac{\sqrt{3}}{2} \end{bmatrix}$$

has the property that when any vector is multiplied by successive powers of B the result is a set of 12 different vectors that keeps repeating.

#9: $B \cdot [x, y] = \left[\dfrac{\sqrt{3} \cdot x}{2} - \dfrac{y}{2}, \ \dfrac{x}{2} + \dfrac{\sqrt{3} \cdot y}{2} \right]$

#10: $B^{13} \cdot [x, y] = \left[\dfrac{\sqrt{3} \cdot x}{2} - \dfrac{y}{2}, \ \dfrac{x}{2} + \dfrac{\sqrt{3} \cdot y}{2} \right]$

#11: $\text{VECTOR}(B^{k} \cdot [2, 2], k, 20)$

The 3rd expression results in a list of the first 20 products.

Fortunately for us, we will not be dealing with matrices like B. Instead we will be dealing with matrices that have all positive entries. Two German Mathematicians, Oskar Perron (1880-1975) and Ferdinand Georg Frobenius (1849-1917), proved that *every matrix having all positive entries has an associated **dominant (Eigen)vector** having all positive terms*. They actually proved much stronger results, but the statement we have given in this paragraph is sufficient for what follows.

Additional comment for students (Josef)

In the previous paragraph we introduced the expression *Eigenvector*. This very important concept needs some explication.

With *A* being a regular quadratic matrix the equation $A \cdot v = \lambda \cdot v$ can be set up. Scalars λ and vectors *v* which fulfil this equation are called *Eigenvalues* (λ) with their respective *Eigenvectors* (*v*). Such Eigenvector only changes its length but not its direction if multiplied by matrix *A*. You can find the Eigenvectors of *A* using *DERIVE* or the CAS-calculator and then show that one of them is identical with the dominant vector from above.

#47: $\text{EIGENVALUES}(A) = [2 \cdot \sqrt{3} + 5, \ 5 - 2 \cdot \sqrt{3}]$

#48: $\text{EXACT\_EIGENVECTOR}(A, \ 2 \cdot \sqrt{3} + 5) = \left[\begin{array}{c} -2 \cdot \sqrt{3} - 2 \\ -1 \end{array} \right]$

#49: $\text{EXACT\_EIGENVECTOR}(A, \ 5 - 2 \cdot \sqrt{3}) = \left[\begin{array}{c} 2 \cdot \sqrt{3} - 2 \\ -1 \end{array} \right]$

#50: $\dfrac{[-2 \cdot \sqrt{3} - 2, \ -1]}{\Sigma([-2 \cdot \sqrt{3} - 2, \ -1])} = \left[2 - \dfrac{2 \cdot \sqrt{3}}{3}, \ \dfrac{2 \cdot \sqrt{3}}{3} - 1 \right]$

The Eigenvectors are presented in columns.

#51: $\dfrac{[-2 \cdot \sqrt{3} - 2, \ -1]}{\Sigma([-2 \cdot \sqrt{3} - 2, \ -1])} = [0.84529946, \ 0.15470053]$

Only this one with equal signs for both components can be the *dominant Eigenvector*.

#52: $\dfrac{[2 \cdot \sqrt{3} - 2, \ -1]}{\Sigma([2 \cdot \sqrt{3} - 2, \ -1])} = \left[\dfrac{2 \cdot \sqrt{3}}{3} + 2, \ -\dfrac{2 \cdot \sqrt{3}}{3} - 1 \right]$

#53: $\dfrac{[2 \cdot \sqrt{3} - 2, \ -1]}{\Sigma([2 \cdot \sqrt{3} - 2, \ -1])} = [3.1547005, \ -2.1547005]$

Compare the result of #50 with the coordinates of point *D* on page 15.

How to find the Eigenvalues?

They are the solutions of the so called characteristical equation, which is in our

case $\begin{vmatrix} 7-\lambda & 8 \\ 1 & 3-\lambda \end{vmatrix} = 0$. (Check it!)

Applying Hierarchical Ranking Methods to the Evidence

The Analytical Hierarchical Ranking Method was devised by *Thomas Saaty* in 1977. It is a method for ordering preferences of several categories or individuals relative to several criteria. In our case we have six suspects, A through E, and three criteria, Motive, Opportunity, and Profit. This technique is especially useful when we are faced with a situation that is hard to quantify, but where we have opinions or feelings when comparing the categories, individuals, or criteria, and we can rank and rate the relative strength of our feelings. In effect, the method gives us a way to quantify some things that seem to be non quantifiable.

First we need to have a scheme for comparing two items relative to a particular criterion. The method devised by Saaty uses a scale of 1 to 9 for comparing the strength of feeling about two individuals/activities relative to the criterion.

> 1 = There is **no preference** for one item over the other
>
> 3 = A **slightly stronger** preference for one over the other
>
> 5 = There is an **essential** preference for one over the other
>
> 7 = There is **some evidence** that one is preferred over the other
>
> 9 = There is **indisputable evidence** for preferring one over the other

2, 4, 6, 8 are intermediate values used as compromises for disagreements in group decisions. In short, this is the standard preference scale of 1 to 5 expanded to provide intermediate values for compromises. The idea is to store preferences when comparing two suspects in an $n \times n$ matrix $[a_{i,j}]$, where

> $a_{i,j} = s$ where s is the score of the preference of item i over item j
>
> $a_{j,i} = \dfrac{1}{s}$. This implies that we always have $a_{i,i} = 1$ for all $i = 1, \ldots, n$.

Let's try out the scheme for suspect A relative to the other suspects. We will assume that a team of at least four detectives was assigned to the case and they based their scores on their interviews.

> A compared with B
>> The detectives found no strong evidence to rank one over the other relative to the motive criteria, so they gave a score of 1. Thus,
>> $a_{1,2} = 1$ and $a_{2,1} = 1$.
>
> A compared to C
>> Although suspect C wanted the African-American family to move out of the neighborhood, the detectives believed that the hurt suffered by A upon rejection by her former friend was a stronger motive so they gave a score of 7.
>>
>> $a_{1,3} = 7$ and $a_{3,1} = \dfrac{1}{7}$.

A compared to D

> The victim was actively standing in the way of the Real Estate Agent's sales perspectives in the neighborhood for his housing project as well as the opportunity to pick up some valuable property cheaply. They all believed this was a stronger motive than A's hurt, although this was a feeling not supported by evidence so they rated it as a 5.

$$a_{1,4} = \frac{1}{5} \quad \text{and} \quad a_{4,1} = 5.$$

A compared to E

> The detectives were split on this comparison. One group believed that the public embarrassment of E by being slapped by the victim in public and his subsequent criticism by his father was clearly stronger motivation to kill the victim than A's rejection by her friend. The other group agreed that it was stronger, but not quite as strong as the first group perceived it to be. The compromise score of 8 was chosen.

$$a_{1,5} = \frac{1}{8} \quad \text{and} \quad a_{5,1} = 8$$

A compared to F

> The consensus was that the rejection that A experienced was a slightly stronger motive than the mother being thwarted in her desire for her dream house by her daughter's sway with the father not to move. They gave a score of 3 to A.

$$a_{1,6} = 3 \quad \text{and} \quad a_{6,1} = \frac{1}{3}.$$

The above assignment of values has filled in row 1 and column 1 of the Motive matrix. A similar analysis was done for the remaining suspects. Note that there are 10 more comparisons to be made in order to fill out the matrix. Here is the matrix proposed by the detectives:

$$\text{#5:} \quad \text{Motive} := \begin{bmatrix} 1 & 1 & 7 & \frac{1}{5} & \frac{1}{8} & 3 \\ 1 & 1 & 9 & 2 & \frac{1}{8} & \frac{1}{4} \\ \frac{1}{7} & \frac{1}{9} & 1 & 2 & \frac{1}{5} & 4 \\ 5 & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{3} & 2 \\ 8 & 8 & 5 & 3 & 1 & 6 \\ \frac{1}{3} & 4 & \frac{1}{4} & \frac{1}{2} & \frac{1}{6} & 1 \end{bmatrix} \qquad (1)$$

A quick view of the matrix shows that row 5 clearly dominates the others, i.e. E seems to have the strongest motives, but what about the ranking of the other suspects relative to this criterion? Also, what about the other criteria of Opportunity and Profit? We take these two questions one at a time. We first rank the suspects relative to the motive criterion. This is where the idea of a dominant vector comes in to play.

If we apply the comparisons in matrix (1) consistently to any ranking of the suspects, then the effect of the comparisons should take effect and the rankings should tend towards the line determined by the dominant vector. (Try this. Use your CAS to multiply the equal ranking vector (1,1,1,1,1,1) by the 30-th power of the Motive matrix and normalize the result by dividing by the sum of the terms so that they add up to one). If you have down loaded the accompanying *DERIVE*© file called , `AHP_Analysis.dfw`, the program, `Dom_vec`, will find the dominant vector for any positive matrix up to a given tolerance between the actual and computed value. Here is the result for the Motive matrix. Compare it with your result.

#6:   Dom_vec(Motive, 0.0001)

#7:   [0.129723, 0.145604, 0.0868161, 0.12758, 0.418666, 0.0916085]

Suspect E, corresponding to the fifth value in the list is clearly the highest ranked according to this criterion. Suspect B is a distant second followed by suspects A, D, F, and C.

Additional comment for students (Josef)

We come back to the *Eigenvector* and show that our approximative procedure can be confirmed in a very mathematical way by applying this concept.

#8:   EIGENVALUES(Motive)

#9:   [9.52509, 0.144898, −1.85023 + 4.74921·i, −1.85023 − 4.74921·i, 0.0152352

  + 2.87102·i, 0.0152352 − 2.87102·i]

#10: dv := APPROX_EIGENVECTOR(Motive, 9.52509)

#11: dv := [−0.261811, −0.293876, −0.175191, −0.257476, −0.844939, −0.184882]

#12: $\dfrac{dv}{\sum(dv)}$

#13: [0.129726, 0.145614, 0.0868066, 0.127578, 0.418664, 0.0916085]

DERIVE returns – approximatively – two real Eigenvalues (approximatively because the characteristical equation is of degree 6). The attempt with the first real Eigenvalue leads to the appropriate Eigenvector. In case of larger matrices calculating the Eigenvalues can lead to problems. It seems to be better to use an iterative process to approach the "limit". Beside the offered program you can perform the first, say 20, multiplications and either observe the whole series of vectors or just the last row of the resulting table:

#13: [0.129726, 0.145614, 0.0868066, 0.127578, 0.418664, 0.0916085]

#14: $\left(\text{ITERATES}\left(\dfrac{\text{Motive}\cdot u}{\sum(\text{Motive}\cdot u)},\ u,\ [1,\ 1,\ 1,\ 1,\ 1,\ 1],\ 20\right)\right)_{21}$

It is no problem to find the Eigenvectors - even for larger matrices – on the CAS-calculators without calculating the Eigenvalues.

The first column (which is the appropriate Eigenvector) must be converted into a list.

The four detectives compare the suspects wirh respect to the remaining criteria which are Opportunity and Profit and calculate the respective dominant vectors.

#16: Opportunity :=
$$\begin{bmatrix} 1 & \dfrac{1}{4} & 3 & 8 & 5 & \dfrac{1}{8} \\ 4 & 1 & 9 & 9 & 6 & \dfrac{1}{2} \\ \dfrac{1}{3} & \dfrac{1}{9} & 1 & 5 & \dfrac{1}{7} & \dfrac{1}{6} \\ \dfrac{1}{8} & \dfrac{1}{9} & \dfrac{1}{5} & 1 & \dfrac{1}{8} & \dfrac{1}{9} \\ \dfrac{1}{5} & \dfrac{1}{6} & 7 & 8 & 1 & \dfrac{1}{6} \\ 9 & 2 & 6 & 9 & 6 & 1 \end{bmatrix}$$

#17: Profit :=
$$\begin{bmatrix} 1 & \dfrac{1}{6} & \dfrac{1}{4} & \dfrac{1}{9} & \dfrac{1}{7} & \dfrac{1}{4} \\ 6 & 1 & \dfrac{1}{7} & \dfrac{1}{9} & 1 & \dfrac{1}{3} \\ 4 & 7 & 1 & \dfrac{1}{3} & \dfrac{1}{5} & 4 \\ 9 & 9 & 3 & 1 & 7 & 8 \\ 7 & 1 & 5 & \dfrac{1}{7} & 1 & 6 \\ 3 & 3 & \dfrac{1}{4} & \dfrac{1}{8} & \dfrac{1}{6} & 1 \end{bmatrix}$$

#18: Dom_vec(Opportunity, 0.00001)

#19: [0.133494, 0.287197, 0.0399400, 0.0183003, 0.0904892, 0.430578]

#20: Dom_vec(Profit, 0.00001)

#21: [0.0223754, 0.0667261, 0.151181, 0.486502, 0.212729, 0.0604838]

Confirm the results by calculating the Eigenvectors.

Upon examining the dominant vectors for the three rankings, you see that each criterion gives a different ranking of the three suspects. The team of detectives has one more "collective decision" to make: How does the evidence fit the crime they are investigating?   In other words, how much weight should they give to each of the three criteria relative to the circumstances of this crime?   Once again, they can use the Analytical Hierarchical Ranking Scheme to assign the weights.

$$
\#22: \quad \text{Priorities} := \begin{bmatrix} 1 & 5 & 8 \\ \dfrac{1}{5} & 1 & 4 \\ \dfrac{1}{8} & \dfrac{1}{4} & 1 \end{bmatrix}
$$

Try to explain the elements of this matrix.

With Dominant Vector:

#23: Dom_vec(Priorities, 0.00001) = [0.733384, 0.199069, 0.0675455]

It is clear that *Motive* and *Opportunity* are the dominant criteria in this case and the *Profit* has relatively little role to play in the opinion of the detectives. This should be expected in the murder of a child coming from a middle class family. Indeed, Motive is by far the prime criterion. Each individuals score for the three criteria is multiplied by the weighting given to the criteria to find a final score for each suspect.

$$
\#24: \quad \begin{bmatrix} 0.129727 & 0.133494 & 0.0223754 \\ 0.145614 & 0.287197 & 0.066726 \\ 0.0868068 & 0.03994 & 0.151181 \\ 0.127578 & 0.0183003 & 0.486502 \\ 0.418664 & 0.0904892 & 0.212729 \\ 0.0916082 & 0.430578 & 0.0604837 \end{bmatrix} \cdot \begin{bmatrix} 0.733384 \\ 0.199069 \\ 0.0675455 \end{bmatrix} = \begin{bmatrix} 0.123225 \\ 0.16847 \\ 0.0818251 \\ 0.130067 \\ 0.339423 \\ 0.156984 \end{bmatrix}
$$

The rows of the left matrix are formed by the values for Motive, Opportunity and Profit for the persons A though F. The product delivers the final evaluation, which must be sorted. As the 5th row is showing the maximum, the teen aged boy seems to be the main suspect. At the other hand the minimum value in the 3rd puts the committee leader – which is suspect C – on the last position of suspects.

This can be done using the *DERIVE*© program `MOP_Suspects` that is a part of the package supplied with this activity. The result of the program is given below.

$$
\#25: \quad \text{MOP\_Suspects(Priorities, Motive, Opportunity, Profit, 0.00001)} = \begin{bmatrix} 0.339425 & E \\ 0.168469 & B \\ 0.156985 & F \\ 0.130068 & D \\ 0.123225 & A \\ 0.0818258 & C \end{bmatrix}
$$

Upon further investigation and questioning, it turned out that suspect E had committed the murder and confessed to the murder. He was arrested, tried, and found guilty.

Exercises Related to the Activity

1. Read a mystery novel and prior to reaching the end of the novel, identify the major suspects. Use the Analytical Hierarchical Ranking Method to identify the possible perpetrator of the crime. Compare your result with the author's solution of the crime.

2. Together with a small group of friends, watch a detective show on television that has a good development of the evidence. Have everyone leave before the end of the show, but videotape the ending portion that you missed. You and your friends create the matrices for the Analytical Hierarchical Ranking Method and determine the most likely guilty suspect. Watch the end of the show and compare your result.

3. The Analytical Hierarchical Ranking Method has applications in many areas. Apply the method to making an important decision in your life, such as choosing a college to attend, choosing a summer job, choosing a destination for a family trip, etc. Obviously you will use criteria other than the ones used for crime solving.

4. Choose important criteria for success in a particular sport. Rank the teams in your league relative to these criteria using the Analytical Hierarchical Ranking Method. How do the results compare to the final rankings of the team after the end of a season?

Final Comments of the Coroner

The role of the coroner ended in this case at the point the coroner assigned an manner (murder) and means (gun shot wound to the chest) to the death of the victim. The remainder of the work was for the police. They did the interviews and gathering of evidence.
Crime solving, like the practice of medicine, is not a science. Scientific methods are applied to the evidence and conclusions are reached. However, it is seldom that the results are based on scientific laws or indisputable facts. There is a lot that requires the insight and training of the investigator. The methods used in this activity are a way for the investigators to organize their insights and "hunches". They are nothing more. They can be used to direct an investigation, not produce irrefutable evidence. In the case of the above investigation, this organization produced a focus that lead to a confession by the guilty party.

More information about AHP can be found on the following websites:

http://en.wikipedia.org/wiki/Thomas_Saaty
http://en.wikipedia.org/wiki/Analytic_Hierarchy_Process
http://www.boku.ac.at/mi/ahp/ahp.pdf   (deutsch)
http://www.boku.ac.at/mi/ahp/ahptutorial.pdf   (english)

Groebner6.dfw - a DfW Utility File for computing Gröbner Bases in DERIVE 6

Alessandro Perotti, Dept. of Mathematics, University of Trento, Italy

perotti@science.unitn.it

www.science.unitn.it/~perotti/groebner.htm

This DfW utility file contains a function that computes Gröbner bases of a set of polynomials with respect to a monomial ordering and other related functions. DERIVE 6 contains the new internal function GROEBNER_BASIS to construct a Gröbner basis for a collection of polynomials based on lexicographic ordering of the variables. We introduce new functionalities giving the possibility of choosing the monomial ordering. Besides allowing the explicit computation of Gröbner bases and normal forms, the new functions can be used, for example, to eliminate some variables between the equations and to perform the Gaussian elimination of a system depending on parameters.

We wish to thank V. Anisiu for his useful suggestions.

- **0. Introduction**
-

A *Gröbner basis* is a set of polynomials in one or more variables that has special properties, useful to solve many problems involving polynomial equations. This concept generalizes *Gaussian elimination* for linear equations and the *Euclidean algorithm* for computing the GCD of two polynomials in one variable. Using the words of B.Sturmfels [2], "Given that polynomial models are ubiquitous across the sciences and engineering, Gröbner bases have been used by researchers in optimization, coding, robotics, control theory, statistics, molecular biology, and many other fields."

To give some examples of *Buchberger's Algorithm* for the construction of a Gröbner basis, we start with a system of two linear equations in three variables

$$3x + 2y + z = 2, \quad 2x - y - 4z = 1.$$

We rewrite it as $f_1 = 3x + 2y + z - 2 = 0$, $f_2 = x - y - 4z - 1 = 0$. To eliminate the variable x from the equations, we form the new linear polynomial

$$f_3 = 1/3\, f_1 - 1/2\, f_2 = 7/6\, y + 7/3\, z - 1/6$$

and get the equivalent set of equations $f_1=0$, $f_2=0$, $f_3=0$. What we have done is: order the variables as $x>y>z$, then use the *leading terms* 3x of f_1 and 2x of f_2 to eliminate the first variable x and get the equation $f_3=0$.

The set $\{f_1, f_2, f_3\}$ that we have obtained is a Gröbner basis of the initial set $\{f_1, f_2\}$, since the linear combinations of the leading terms of f_1, f_2, f_3 give all the possible combinations of the leading terms of the original polynomials f_1, f_2. From the Gröbner basis $\{f_1, f_2, f_3\}$, a *reduced Gröbner basis* can be obtained. In the linear case it corresponds to the *row-reduced echelon form* of the system of equations. In our example a reduced basis is given by

$$\{g_1 = x - z - 4/7, \quad g_2 = y + 2z - 1/7\}$$

In the *non-linear* case, in order to make a similar construction, we have to fix an ordering for the monomials. Given an ordering for the variables, there are many orders which are compatible with the product of monomials. For example, the *lexicographic ordering* (used by the internal function GROEBNER_BASIS) for monomials in two variables $x > y$ is

$$1 < y < y^2 < y^3 < ... < x < xy < xy^2 < xy^3 < ... < x^2 < x^2y < ... < x^3 < ...$$

while the *graded lexicographic ordering* gives

$$1 < x < y < x^2 < xy < y^2 < x^3 < x^2y < y^3 < ...$$

Once the monomial ordering is fixed, every polynomial has a *leading term* (a monomial multiplied by a coefficient). For example, let F be the set

$$\{f_1 = x^2 + y^2 + z^2 + w^2, f_2 = x + z^3 - w^3, f_3 = x^2 + 2\,y^2 - y\,z - w^2\}.$$

The leading terms of f_1, f_2, f_3 w.r.t. the graded lexicographic ordering are, respectively, x^2, z^3, x^2. Now we take elements of F in pairs and cancel out their leading terms by means of *polynomial* combinations of them. We form the so-called *S-polynomials*

$$f_4 = z^3\,f_1 - x^2\,f_2 = -x^3 + w^3\,x^2 + z^3\,(y^2 + z^2 + w^2)$$

$$f_5 = f_1 - f_3 = -y^2 + y\,z + z^2 + 2\,w^2$$

$$f_6 = x^2\,f_2 - z^3\,f_3 = x^3 - w^3\,x^2 - z^3\,(2\,y^2 - y\,z - w^2)$$

We now apply the *division algorithm* and find that

$$f_4 = (w^3 - x)\,f_1 + (y^2 + w^2 + z^2)\,f_2$$

and then we don't need to add f_4 to the list $\{f_1, f_2, f_3\}$. Instead, the leading term of f_5, which is $-y^2$, can not be cancelled out by means of the leading terms of $f_1, f_2, f_3.$ Then we add f_5 to the set F and get $F' = \{f_1, f_2, f_3, f_5\}$. When we apply the division algorithm to f_6 and the new set F' we get

$$f_6 = (x - w^3)\,f_1 + (-y\,z - 2\,z^2 - 3\,w^2)\,f_2 + (x + 2\,z^3 - w^3)\,f_5$$

and so we don't add f_6 to the set F'.
If we now form the S-polynomials of pairs containing f_5 and divide them through the elements of F', we get only polynomial combinations of f_1, f_2, f_3, f_5.
Buchberger's Criterion says that F' is then a Gröbner basis of the set F: the polynomial combinations of the elements of F are the same obtained by F' and the leading terms

$$x^2, z^3, -y^2$$

of the elements of F' divide the leading terms of every polynomial combinations of the elements of F.
From F' the unique *reduced* Gröbner basis G of the set F can be obtained applying the division algorithm. It is a Gröbner basis characterized by two properties: every leading term has coefficient 1 and every non-leading term of the elements of G is not divisible by the leading monomials of F' (which are the same as those of G).
In our example, the reduced Gröbner basis w.r.t. the graded lexicographic ordering is

$$G = \{g_1 = y^2 - y\,z - z^2 - 2\,w^2, g_2 = x^2 + y\,z + 2\,z^2 + 3\,w^2, g_3 = x + z^3 - w^3\}$$

with leading terms y^2, x^2, z^3. The Gröbner basis reveals geometric properties of the set V of common zeroes of f_1, f_2, f_3. The absence of a power of w in the leading monomials of G means that there are infinite common zeroes (over the complex numbers). More precisely, it says that V is one-dimensional. This is similar to the linear case, where the number of not-leading variables in the reduced echelon equations gives the dimension of the linear variety of ze- roes of a compatible system. Besides, the set V is empty exactly when the reduced Gröbner basis is {1}.

Note that the internal function GROEBNER_BASIS returns a Gröbner basis w.r.t. the lexico- graphic ordering which is not necessarily reduced, since the coefficients of the leading terms can be different from 1.

- **1. Functions**

`Groebner(f,v,ord)`	reduced Gröbner basis for a set of polynomials
`Eliminate(f,v1,v2)`	elimination of a set of variables in equations f=0
`TotalDegree(f, v)`	total degree of a polynomial
`OrdMatrix(ord, n)`	matrices inducing relevant monomial orderings
`ElOrd(k, n)`	matrices inducing elimination orderings
`LeadingTerm(f, v, ord)`	leading term of a polynomial
`NF(f, ff, v, ord)`	normal form of a polynomial with respect to a set of polynomials
`Buchberger(f, v, ord)`	(non reduced) Gröbner basis for a set of polynomials
`System(e, v, k)`	gaussian reduction of a system depending on parameters
`Eigensystem(a, ord, v, w)`	reduction of the eigensystem Ax=λx
`RREF(a, k)`	row-reduction of a matrix depending on parameters
`RREF(a, b, k)`	row-reduction of augmented matrix depending on parameters
`LagrangeMultipliers(f,g,v)`	computation of min-max points using Lagrange multipliers

- **2. Description of the functions**

- **2.1 Computation of Gröbner Bases**

`Groebner([f1,...,fm], [x1,...,xn], ord)`

computes the reduced Gröbner basis for the set of polynomials generated by **`f1,...,fm`** with respect to the ordered variables **`x1,...,xn`** and to the monomial ordering described by **`ord`**.

The argument **ord** that defines the monomial ordering can be:

- an integer between 1 and 4 (or the variables **lex, grlex, grevlex, invlex**), correspond-
ing to **lexicographic, graded lexicographic, graded reverse lexico-
graphic, inverse lexicographic**
- a non-singular matrix with non-negative integral entries that induces the ordering (see [1] and 2.2 for
more information).

• The default ordering is **lexicographic** with respect to the variables **x1>x2>...>xn**.

• If also the second argument is omitted, the ordering is **lexicographic** with respect to the
variables appearing in **f1,...,fm** ordered by internal Derive ordering.

• **Examples**

#1: $\text{Groebner}\left(\left[x^2 + y^2 + z^2 + w^2,\ x^2 + 2 \cdot y^2 - y \cdot z - w^2,\ x + z^3 - w^3\right],\right.$

$\left.[x,\ y,\ z,\ w],\ \text{grevlex}\right)$

#2: $\left[y^2 - y \cdot z - z^2 - 2 \cdot w,\ x^2 + y \cdot z + 2 \cdot z^2 + 3 \cdot w,\ x + z^3 - w^3\right]$

#3: $\text{Groebner}\left(\left[x^2 + y^2 + z^2 + w^2,\ x^2 + 2 \cdot y^2 - y \cdot z - w^2,\ x + z^3 - w^3,\ w - 1\right]\right)$

#4: $\left[w - 1,\ z^{12} - 4 \cdot z^9 + 5 \cdot z^8 + 12 \cdot z^6 - 10 \cdot z^5 + 5 \cdot z^4 - 16 \cdot z^3 + 18 \cdot z^2 + 16,\right.$

$\left. y - \frac{z^{11}}{4} + z^8 - \frac{5 \cdot z^7}{4} - 2 \cdot z^5 + \frac{5 \cdot z^4}{2} - \frac{5 \cdot z^3}{4} + 2 \cdot z^2 - \frac{5 \cdot z}{2},\ x + z^3 - 1\right]$

#5: $\text{GROEBNER\_BASIS}\left(\left[x^2 + y^2 + z^2 + w^2,\ x^2 + 2 \cdot y^2 - y \cdot z - w^2,\ x + z^3 - w^3,\ w - 1\right],\right.$

$\left.[x,\ y,\ z,\ w]\right)$

#6: $\left[w - 1,\ z^{12} - 4 \cdot z^9 + 5 \cdot z^8 + 12 \cdot z^6 - 10 \cdot z^5 + 5 \cdot z^4 - 16 \cdot z^3 + 18 \cdot z^2 + 16,\right.$

$\left.4 \cdot y - z^{11} + 4 \cdot z^8 - 5 \cdot z^7 - 8 \cdot z^5 + 10 \cdot z^4 - 5 \cdot z^3 + 8 \cdot z^2 - 10 \cdot z,\ x + z^3 - 1\right]$

#7: $\text{Groebner}\left(\left[x^2 \cdot y^3 - x \cdot y^2,\ x^2 \cdot y - x\right],\ [x,\ y],\ \begin{bmatrix} 1 & 3 \\ 3 & 0 \end{bmatrix}\right)$

#8: $\left[x^3 - x,\ x \cdot y - x^2\right]$

Eliminate([f1,...,fm], [x1,...,xh], [y1,...,yk])
constructs equations in which some variables have been eliminated. It eliminates the first set
of variables **x1,...,xh** between the polynomial equations **f1=0,...,fm=0** in the vari-
ables **x1,...,xh,y1,...,yk**.

• If the third argument is omitted, the equations are considered with respect to all the variables appearing in `f1,...,fm`.

• As it is shown in the examples of section 3, the function can be used in some cases also when the equations are not polynomial.

• **Examples**

#9: $\text{Eliminate}\left(\left[x^2 + y^2 + z^2 + w^2, \; x^2 + 2 \cdot y^2 - y \cdot z - w^2, \; x + z^3 - w^3, \; w - 1\right], [x, w]\right)$

#10: $\left[y^2 - y \cdot z - z^2 - 2, \; y \cdot z + z^6 - 2 \cdot z^3 + 2 \cdot z^2 + 4\right]$

#11: $\text{Eliminate}\left(\left[x^2 \cdot y^3 - x \cdot y^2, \; x^2 \cdot y^4 - x\right], [x]\right)$

#12: $[\,]$

• **2.2 Other functions related to Gröbner Bases**

`TotalDegree(f, [x1,...,xn])`

computes the total degree of a multivariate polynomial `f` in the variables `x1,...,xn`.

#13: $\text{TotalDegree}(2 \cdot x^5 \cdot y^7 - 3 \cdot x \cdot y \cdot z^5)$

#14: 12

`OrdMatrix(ord, n)`

gives an integer square matrix of order `n` that induces the rational monomial ordering `ord`, which can be `lex` or 1, `grlex` or 2, `grevlex` or 3, `invlex` or 4. See [1] for more details.

#15: `MAP_LIST(OrdMatrix(ord, 3), ord, [1, ..., 4])`

#16: $\left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}\right]$

`ElOrd(k, n)`

gives an integer square matrix of order `n` that induces an elimination ordering with respect to which the first `k` variables always precede the others.

#17: `ElOrd(2, 4)`

#18: $\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

LeadingTerm(f, [x1,...,xn], ord)

computes the leading term of the polynomial **f** with respect to the ordering **ord** in the variables **x1>x2>...>xn**.

• The default ordering is lexicographic with respect to all the variables appearing in **f**.

#19: MAP_LIST(LeadingTerm(2·x²·y³ + 3·x²·y²·z² − x·y⁵, [x, y, z], ord), ord,

[lex, grlex, grevlex, invlex])

#20: $\left[2·x^2·y^3,\ 3·x^2·y^2·z^2,\ -x·y^5,\ 3·x^2·y^2·z^2\right]$

NF(f, [f1,...,fk], [x1,...xn], ord)

returns the remainder of the polynomial **f** on division by **f1,...,fk** with respect to the ordering **ord**.

If **[f1,...,fk]** is a Gröbner basis, the remainder does not depend on the ordering of the polynomials and is called **Normal Form** of the polynomial **f** with respect to **[f1,...,fk]**. It is zero if and only if **f** is a polynomial combination of **f1,...,fk**.

• The default ordering is lexicographic with respect to all the variables appearing in **f**.

#21: NF(3·x²·y, [x + z³ − w³, x² + y·z + 2·z² + 3·w², y² − y·z − z² − 2·w²],

[x, y, z, w], grlex)

#22: 3·x − 9·y·(z² + w²) − 6·w²·z − 3·w³

#23: NF(3·x²·y, [x + z³ − w³, x² + y·z + 2·z² + 3·w², y² − y·z − z² − 2·w²])

#24: y·(3·z⁶ − 6·w³·z³ + 3·w⁶)

Buchberger([f1,...,fm], [x1,...,xn], ord)

implements the Buchberger algorithm for the computation of a (non-reduced) Gröbner basis for the set of polynomials generated by **f1,...,fm** with respect to the ordered variables **x1,...,xn** and to the monomial ordering described by **ord**. The basis obtained always containes the generators **f1,...,fm**.

• The function **Groebner(f,v,ord)** uses an improved version of Buchberger algorithm for the computation of the reduced basis.
• The default ordering is lexicographic with respect to all the variables appearing in **f**.

#25: Buchberger([x² + y² + z² + w², x² + 2·y² − y·z − w², x + z³ − w³], [x, y, z, w]

[x, y, z, w], grevlex)

#26: $\left[x^2 + y^2 + z^2 + w^2,\ x^2 + 2·y^2 − y·z − w^2,\ x + z^3 − w^3,\ -y^2 + y·z + z^2 + 2·w^2\right]$

#27: Buchberger([x² + y² + z² + w², x² + 2·y² − y·z − w², x + z³ − w³])

#28: $[x^2 + y^2 + z^2 + w^2, x^2 + 2\cdot y^2 - y\cdot z - w^2, x + z^3 - w^3, -y^2 + y\cdot z + z^2 + 2\cdot w^2,$

$y\cdot z^6 + z^3 - 2\cdot w^3\cdot z^3 + 2\cdot z^2 + w^6 + 3\cdot w^2, -w^2\cdot y\cdot(w^4 + 3) + z^{11} - 4\cdot w^3\cdot z^8 + 5\cdot z^7 +$

$z^5\cdot(5\cdot w^6 + 3\cdot w^2) - 10\cdot w^3\cdot z^4 + 5\cdot z^3 - 2\cdot w^3\cdot z^2\cdot(w^4 + 3) + z\cdot(3\cdot w^6 + 7\cdot w^2),$

$z^{17} - 6\cdot w^3\cdot z^{14} + 5\cdot z^{13} + z^{11}\cdot(14\cdot w^6 + 6\cdot w^2) - 20\cdot w^3\cdot z^{10} + 5\cdot z^9 - 8\cdot w^5\cdot z^8\cdot(2\cdot w^4 +$

$3) + z^7\cdot(25\cdot w^6 + 13\cdot w^2) - 10\cdot w^3\cdot z^6 + z^5\cdot(9\cdot w^{12} + 30\cdot w^8 + 9\cdot w^4) - 2\cdot w^5\cdot z^4\cdot(5\cdot w^4 +$

$+ 13) - 2\cdot w^7\cdot z^2\cdot(w^8 + 6\cdot w^4 + 9), -z^{14} + 4\cdot w^3\cdot z^{11} - 5\cdot z^{10} - 6\cdot w^2\cdot z^8\cdot(w^4 + 1) +$

$10\cdot w^3\cdot z^7 - 5\cdot z^6 + z^5\cdot(4\cdot w^9 + 12\cdot w^5) - w^5\cdot z^2\cdot(5\cdot w^4 + 13) - w^4\cdot z^8\cdot(w + 6\cdot w^4 + 9),$

$2\cdot w^3\cdot z^{13} - 2\cdot z^{12} - 8\cdot w^6\cdot z^{10} + 18\cdot w^3\cdot z^9 - 10\cdot z^8 + z^7\cdot(12\cdot w^9 + 12\cdot w^5) -$

$4\cdot w^2\cdot z^6\cdot(8\cdot w^4 + 3) + 30\cdot w^3\cdot z^5 - 2\cdot z^4\cdot(4\cdot w^{12} + 12\cdot w^8 + 5) + z^3\cdot(18\cdot w^9 + 50\cdot w^5) -$

$2\cdot w^2\cdot z^2\cdot(5\cdot w^4 + 13) + z\cdot(2\cdot w^{15} + 12\cdot w^{11} + 18\cdot w^7) - 2\cdot w^{12} - 12\cdot w^8 - 18\cdot w^4,$

$z^{12} - 4\cdot w^3\cdot z^9 + 5\cdot z^8 + z^6\cdot(6\cdot w^6 + 6\cdot w^2) - 10\cdot w^3\cdot z^5 + 5\cdot z^4 - 4\cdot w^5\cdot z^3\cdot(w^4 + 3) +$

$z^2\cdot(5\cdot w^6 + 13\cdot w^2) + w^{12} + 6\cdot w^8 + 9\cdot w^4]$

- **2.3 Elementary applications to linear systems and matrices**

System([e1,...,em], [x1,...,xn], [k1,...,ks])
performs the gaussian reduction of the system of linear equations **e1=0,...,em=0** in the variables **x1,...,xn** depending polynomially on parameters **k1,...,ks**.

If the vector of parameters is omitted, then in the resulting equations may appear rational functions of the parameters and they may be not equivalent to the given equations for some values of the parameters.

#29: System($[x + y + z - 1, k^2\cdot x + 4\cdot y + 9\cdot z + 11, k\cdot x + 2\cdot y + 3\cdot z + 1]$, [x, y, z],

  [k])

#30: $[z\cdot(k - 3) + 3\cdot k - 9, y\cdot(k - 2) - 4\cdot k + 8, y\cdot(z + 3) - 4\cdot z - 12, x + y + z - 1]$

From the reduced system we get that if k=3 the system has infinite solutions x = -3-t, y = 4, z = t (t any real), if k=2 the system has infinite solutions x = 4-t, y = t, z = -3 (t any real), else it has a unique solution x = 0, y = 4, z = -3.

#31: System($[x + y + z - 1, k^2\cdot x + 4\cdot y + 9\cdot z + 11, k\cdot x + 2\cdot y + 3\cdot z + 1]$, [x, y, z])

#32: $[z + 3, y - 4, x]$

Eigensystem(a, ord, [x1,...,xn], w)

performs the gaussian reduction of the eigenvalues system Ax=λx associated to a square matrix **a**. The default ordering is **lex**, with default linear variables **x1,...,xn** and eigen-value variable **w**.

$$\text{\#33: } \text{Eigensystem} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 4 \\ 0 & 0 & 2 \end{bmatrix}$$

$$\text{\#34: } \left[w \cdot x3 - 2 \cdot x3, \ w^2 \cdot x2 - 5 \cdot w \cdot x2 + 5 \cdot x2, \ x2 \cdot x3, \ -w \cdot x2 + x1 + 3 \cdot x2 + 4 \cdot x3 \right]$$

$$\text{\#35: } \text{Eigensystem} \left(\begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 4 \\ 0 & 0 & 2 \end{bmatrix}, \ \text{grlex}, \ [x, y, z], \ t \right)$$

$$\text{\#36: } \left[z \cdot (t - 2), \ -x + y \cdot (t - 3) - 4 \cdot z, \ y \cdot z, \ x \cdot (t - 2) - y, \ x \cdot z + 4 \cdot z^2, \right.$$
$$\left. x^2 + x \cdot y - y^2 - 16 \cdot z^2 \right]$$

RREF(a, [k1,...,ks])

tries to compute a Row Reduced Echelon Form of a matrix **a** depending on parameters **k1,...,ks**.

RREF(a, b, [k1,...,ks])

computes the Row Reduced Echelon Form of the augmented matrix **[a,b]** depending on parameters **k1,...,ks**. The second argument **b** can be a vector or a matrix. Any null row is deleted.

If the parameters are omitted, then the internal function **ROW_REDUCE** is called for a more efficient computation. In this case, the reduced matrix may contain rational functions of the parameters and the reduction may be not valid for some values of the parameters.

$$\text{\#37: } \text{RREF} \left(\begin{bmatrix} 2 & 3 & 4 \cdot h \\ 3 & 2 \cdot h & 1 \\ 0 & 2 & 1 \end{bmatrix}, \ [h] \right)$$

$$\text{\#38: } \begin{bmatrix} 1 & 0 & \dfrac{5}{28} \\ 0 & 1 & \dfrac{1}{2} \\ 0 & 0 & h - \dfrac{13}{28} \end{bmatrix}$$

The **RREF** shows that the matrix has rank 2 for h=13/28, rank 3 otherwise.

$$\text{\#39: } \text{RREF} \left(\begin{bmatrix} 2 & 3 & 4 \cdot h \\ 3 & 2 \cdot h & 1 \\ 0 & 2 & 1 \end{bmatrix}, \ [2, 1, 1], \ [h] \right)$$

#40:
$$\begin{bmatrix} 1 & 0 & \dfrac{5}{28} & \dfrac{9}{28} - \dfrac{2 \cdot h}{7} \\[3mm] 0 & 1 & \dfrac{1}{2} & \dfrac{1}{2} \\[3mm] 0 & 0 & h - \dfrac{13}{28} & \dfrac{h}{7} - \dfrac{1}{28} \end{bmatrix}$$

#41: $\text{RREF} \begin{bmatrix} 2 & 3 & 4 \cdot h \\ 3 & 2 \cdot h & 1 \\ 0 & 2 & 1 \end{bmatrix}$

#42: $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- **2.4 Another application: the Lagrange multipliers**

`LagrangeMultipliers(f,[g1,...,gk],[x1,...,xn])`
applies the method of Lagrange multipliers for the computation of critical points of a polynomial function **`f(x1,...xn)`** subject to k polynomial constraints **`g1=0,...,gk=0`**. It returns a system of equations reduced with respect to lexicographic ordering, whose solutions are the coordinates of the critical points.

#43: $\text{LagrangeMultipliers}(z^2 - x \cdot y \cdot z + x, [x^2 + y^2 - 1, y \cdot z - 2])$

#44: $\left[z^8 - 4 \cdot z^6 - 4, \ y - \dfrac{z^7}{2} + 2 \cdot z^5, \ x - \dfrac{z^4}{2} + 2 \cdot z^2 \right]$

#45: $\text{LagrangeMultipliers}(z^2 - x \cdot y \cdot z + x, [y \cdot z - 2])$

#46: $[1]$

If the constant polynomial 1 is returned, the function has no critical points subject to the constraints.

- **3 Other examples**

In the univariate case, the Gröbner basis contains only the GCD of the polynomials.

#47: $\text{Groebner}([x^9 - 3 \cdot x^8 + x^7 - 3 \cdot x^6 - 3 \cdot x^5 + 6 \cdot x^4 + 17 \cdot x^3 - 22 \cdot x^2 - 11 \cdot x + 15,$
$3 \cdot x^7 - 9 \cdot x^6 + 5 \cdot x^5 - 15 \cdot x^4 - 4 \cdot x^3 + 3 \cdot x^2 + 48 \cdot x - 63])$

#48: $[x - 3]$

The function **Eliminate** can be applied to find the fourth degree equation of a torus starting from its parametric equations. The trigonometric functions can be considered as new variables to be eliminated, subject to conditions sin(t)^2+cos(t)^2=1 and sin(u)^2+cos(u)^2=1.

#49: $[(2 + \text{COS}(t)) \cdot \text{COS}(u), (2 + \text{COS}(t)) \cdot \text{SIN}(u), \text{SIN}(t)]$

#50: $\text{Eliminate}([x - (2 + ct) \cdot cu, y - (2 + ct) \cdot su, z - st, st^2 + ct^2 - 1, su^2 + cu^2 - 1], [ct, st, cu, su], [x, y, z])$

#51: $[x^4 + x^2 \cdot (2 \cdot y^2 + 2 \cdot z^2 - 10) + y^4 + y^2 \cdot (2 \cdot z^2 - 10) + z^4 + 6 \cdot z^2 + 9]$

In the following example the **Eliminate** function is called two times to obtain the equations of the projections of a parametric curve on the coordinate planes yz, xz, xy.

#52: $\text{gb\_xyz} := \text{Eliminate}([x - t, y - t^3, z - t^4], [t])$

#53: $\text{gb\_xyz} := [x \cdot y - z, x^2 \cdot z - y^2, y^3 - x \cdot z^2, x^3 - y]$

#54: $\text{MAP\_LIST}(\text{Eliminate}(\text{gb\_xyz}, [v]), v, [x, y, z])$

#55: $\begin{bmatrix} y^4 - z^3 \\ x^4 - z \\ x^3 - y \end{bmatrix}$

An example taken from Cox,D., Little,J., O'Shea,D., *Ideals, Varieties, and Algorithms*, Springer-Verlag, New York-Berlin-Heidelberg (1992):

#56: $\text{polys} := [x^5 + y^4 + z^3 - 1, x^3 + y^2 + z^2 - 1]$

#57: $\text{Groebner}(\text{polys}, [x, y, z], \text{grevlex})$

#58: $[x^3 + y^2 + z^2 - 1, x^2 \cdot (y^2 + z^2 - 1) - y^4 - z^3 + 1, x \cdot (y^2 + z^2 - 1) + y^4 +$
$y^2 \cdot (2 \cdot z^2 - 2) + z^4 - 2 \cdot z^2 + 1, x^2 \cdot (z^4 + z^3 - 2 \cdot z^2) + x \cdot (y^2 \cdot (2 \cdot z^2 - 2) +$
$z^4 - z^3 - 2 \cdot z^2 + 2) + y^6 - y^4 \cdot z^2 + y^2 \cdot (z^3 - 2 \cdot z^2 + 1) - z^5 - z^4 + z^3 +$
$3 \cdot z^2 - 2]$

#59: $\text{Groebner}(\text{polys}, [x, y, z], \text{grlex})$

#60: $[x^3 + y^2 + z^2 - 1, x^2 \cdot (y^2 + z^2 - 1) - y^4 - z^3 + 1, x \cdot (y^2 + z^2 - 1) + y^4 +$
$y^2 \cdot (2 \cdot z^2 - 2) + z^4 - 2 \cdot z^2 + 1, x^2 \cdot (z^4 + z^3 - 2 \cdot z^2) + x \cdot (y^2 \cdot (2 \cdot z^2 - 2) + z^4 -$

$$z^3 - 2 \cdot z^2 + 2) + y^6 - y^4 \cdot z^2 + y^2 \cdot (z^3 - 2 \cdot z^2 + 1) - z^5 - z^4 + z^3 + 3 \cdot z^2 - 2,$$

$$x \cdot (y^2 \cdot (3 \cdot z^4 - z^3 - 6 \cdot z^2 + 4) + z^6 - 3 \cdot z^5 - 3 \cdot z^4 + 3 \cdot z^3 + 6 \cdot z^2 - 4) +$$

$$y^8 - y^6 + y^4 \cdot (2 \cdot z^3 - 5 \cdot z^2 + 3) - y^2 \cdot (7 \cdot z^4 - 14 \cdot z^2 + 7) - 2 \cdot z^6 + 9 \cdot z^4 -$$

$$\left. 2 \cdot z^3 - 9 \cdot z^2 + 4 \right]$$

An assignment to a variable is a convenient way for using a lengthy result in other computations:

#61: `PROG(gb ≔ Groebner(polys, [x, y, z], lex), MAP_LIST(LeadingTerm(f), f, gb))`

#62:
$$\left[y^{12},\ x \cdot z^{11},\ 24 \cdot x \cdot y^2 \cdot z,\ x \cdot y^4,\ 12 \cdot x^2 \cdot z^4,\ x^2 \cdot y^2,\ x^3 \right]$$

This means that the lex Gröbner basis contains 7 polynomials...

#63: `MAP_LIST(DIM(TERMS(EXPAND(f))), f, gb)`

#64: $[25, 49, 53, 9, 49, 6, 4]$

...and that these polynomials contain up to 53 terms...

#65: `MAP_LIST(TotalDegree(f), f, gb)`

#66: $[12, 13, 12, 5, 12, 4, 3]$

...and their maximum total degree is 13.

#67: $NF(x^{10} \cdot y^3 \cdot z^3,\ gb)$

#68: $y^{11} \cdot z^3 + y^7 \cdot (2 \cdot z^6 - 2 \cdot z^3) + y^3 \cdot (z^9 - 2 \cdot z^6 + z^3)$

- **References**

[1] A. Perotti, *Gröbner Bases with DERIVE*, International DERIVE Journal, Vol.3,n.2, 83-98 (1996)

[2] B. Sturmfels, **What is a Gröbner Basis?**, Notices of the AMS, Vol.52, n.10, 1199-1200 (2005)

[3] J. von zur Gathen and J. Gerhard, **Modern Computer Algebra**, Cambridge University Press (1999)

I´d like to add two examples demonstrating the power of Groebner Bases (Josef)

In DNL#25 Karl Heinz Keunecke presented a system of linear equations

$[2 \cdot x + 6 \cdot y - 3 \cdot z + 6 = 0, \ 4 \cdot x + 3 \cdot y + 3 \cdot z - 6 = 0, \ 4 \cdot x - 3 \cdot y + 9 \cdot z - k = 0]$

which – according to the result of applying ROW_REDUCE – seems to have no solution:

$$\text{ROW\_REDUCE} \begin{bmatrix} 2 & 6 & -3 & -6 \\ 4 & 3 & 3 & 6 \\ 4 & -3 & 9 & k \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dfrac{3}{2} & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The Groebner Basis shows that there is a solution:

`GROEBNER_BASIS([2·x + 6·y – 3·z + 6, 4·x + 3·y + 3·z – 6, 4·x – 3·y + 9·z – k], [x, y, z, k])`

`[k – 18, y – z + 2, 2·x + 3·z – 6]`

For $k = 18$ we have $x = \dfrac{6 - 3z}{2}, \ y = z - 2$.

Let´s try to solve a polynomial system by applying Groebner Bases. The following system is to be solved:

$$\text{equs} := \begin{bmatrix} x^2 + y^2 = 5, \ x^2 + z + y = 10, \ y^2 + x^2 \cdot z^2 = 20 \end{bmatrix}$$

We use the implemented GROEBNER_BASIS function and we get

`grb := GROEBNER_BASIS([x² + y² – 5, x² + z + y – 10, y² + x²·z² – 20], [x, y, z])`

`grb := [z⁶ – 20·z⁵ + 93·z⁴ + 70·z³ – 474·z² – 50·z + 605, 15·y – z⁴ + 20·z³ – 94·z² – 35·z + 230, 15·x`
`       + z⁴ – 20·z³ + 94·z² + 50·z – 380]`

The first equation can be solved for z:

`zvals := SOLUTIONS(grb , z)`
`                      1`

`zvals := [7.557203220, 12.10993346, 1.684943157 + 0.1308187913·i, 1.684943157 – 0.1308187913·i,`
`   –1.518511499 + 0.09328222015·i, –1.518511499 – 0.09328222015·i]`

We substitute the 6 solutions for z in the 2nd equation which then is solved for y:

`yvals := VECTOR(SOLUTIONS(v, y), v, VECTOR(grb , z, zvals))'`
`                              2`
`                                          1`

`yvals := [2.175471042, –2.212919732, 0.5373527040 + 1.751128798·i, 0.5373527040 – 1.751128798·i,`
`   0.4813716390 – 2.503768862·i, 0.4813716390 + 2.503768862·i]`

We substitute the 6 solutions for z in the 2nd equation which then is solved for x:

```
xvals := VECTOR(SOLUTIONS(v, x), v, VECTOR(grb , z, zvals))'
                                           3
```

$$
\text{xvals} := \begin{bmatrix} 0.5170355277 & 0.3209147365 & 2.808901420 - 0.3349970874 \cdot i & 2.808901420 + 0.3349970874 \cdot i \\ -0.5170355277 & -0.3209147365 & -2.808901420 + 0.3349970874 \cdot i & -2.808901420 - 0.3349970874 \cdot i \end{bmatrix}
$$

$$
\begin{matrix} -3.341738775 - 0.3606635353 \cdot i & -3.341738775 + 0.3606635353 \cdot i \\ 3.341738775 + 0.3606635353 \cdot i & 3.341738775 - 0.3606635353 \cdot i \end{matrix}
$$

$$
\begin{bmatrix} \text{xvals1} := \text{xvals} , & \text{xvals2} := \text{xvals} \\ \quad\quad\quad 1 & \quad\quad\quad 2 \end{bmatrix}
$$

The equation is quadratic in *x*, so we obtain two solutions for each *z*:. Finally we have to collect the solutions considering that each pair (*z*,*y*) has two possible values for *x*.

```
[APPEND(xvals1, xvals2), APPEND(yvals, yvals), APPEND(zvals, zvals)]'
```

$$
\begin{bmatrix} 0.5170355277 & 2.175471042 & 7.557203220 \\ 0.3209147365 & -2.212919732 & 12.10993346 \\ 2.808901420 - 0.3349970874 \cdot i & 0.5373527040 + 1.751128798 \cdot i & 1.684943157 + 0.1308187913 \cdot i \\ 2.808901420 + 0.3349970874 \cdot i & 0.5373527040 - 1.751128798 \cdot i & 1.684943157 - 0.1308187913 \cdot i \\ -3.341738775 - 0.3606635353 \cdot i & 0.4813716390 - 2.503768862 \cdot i & -1.518511499 + 0.09328222015 \cdot i \\ -3.341738775 + 0.3606635353 \cdot i & 0.4813716390 + 2.503768862 \cdot i & -1.518511499 - 0.09328222015 \cdot i \\ -0.5170355277 & 2.175471042 & 7.557203220 \\ -0.3209147365 & -2.212919732 & 12.10993346 \\ -2.808901420 + 0.3349970874 \cdot i & 0.5373527040 + 1.751128798 \cdot i & 1.684943157 + 0.1308187913 \cdot i \\ -2.808901420 - 0.3349970874 \cdot i & 0.5373527040 - 1.751128798 \cdot i & 1.684943157 - 0.1308187913 \cdot i \\ 3.341738775 + 0.3606635353 \cdot i & 0.4813716390 - 2.503768862 \cdot i & -1.518511499 + 0.09328222015 \cdot i \\ 3.341738775 - 0.3606635353 \cdot i & 0.4813716390 + 2.503768862 \cdot i & -1.518511499 - 0.09328222015 \cdot i \end{bmatrix}
$$

Compare with the direct DERIVE solution of the system:

```
SOLUTIONS(equs, [x, y, z])
```
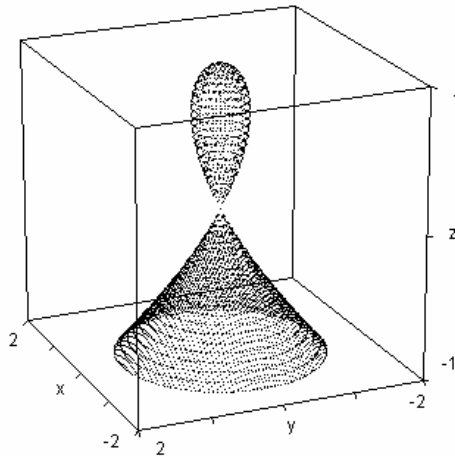
$$
\begin{bmatrix} 0.3209147343 & -2.212919730 & 12.10993346 \\ -0.3209147343 & -2.212919730 & 12.10993346 \\ 0.5170355261 & 2.175471044 & 7.557203220 \\ -0.5170355261 & 2.175471044 & 7.557203220 \\ 3.341738775 + 0.3606635353 \cdot i & 0.4813716390 - 2.503768862 \cdot i & -1.518511499 + 0.09328222015 \cdot i \\ 3.341738775 - 0.3606635353 \cdot i & 0.4813716390 + 2.503768862 \cdot i & -1.518511499 - 0.09328222015 \cdot i \\ -3.341738775 + 0.3606635353 \cdot i & 0.4813716390 + 2.503768862 \cdot i & -1.518511499 - 0.09328222015 \cdot i \\ -3.341738775 - 0.3606635353 \cdot i & 0.4813716390 - 2.503768862 \cdot i & -1.518511499 + 0.09328222015 \cdot i \\ 2.808901420 + 0.3349970874 \cdot i & 0.5373527040 - 1.751128798 \cdot i & 1.684943157 - 0.1308187913 \cdot i \\ 2.808901420 - 0.3349970874 \cdot i & 0.5373527040 + 1.751128798 \cdot i & 1.684943157 + 0.1308187913 \cdot i \\ -2.808901420 + 0.3349970874 \cdot i & 0.5373527040 + 1.751128798 \cdot i & 1.684943157 + 0.1308187913 \cdot i \\ -2.808901420 - 0.3349970874 \cdot i & 0.5373527040 - 1.751128798 \cdot i & 1.684943157 - 0.1308187913 \cdot i \end{bmatrix}
$$

Surface #3: $x^2 + y^2 = z^2 - z^3$

```
                2    2    2    3
VECTOR(ContourPts_XY(x  + y  - z  + z , 1, -1.5, 1.5, -1.5, 1.5, 0.05, 0.05), 1, -1, 1, 0.1)
```
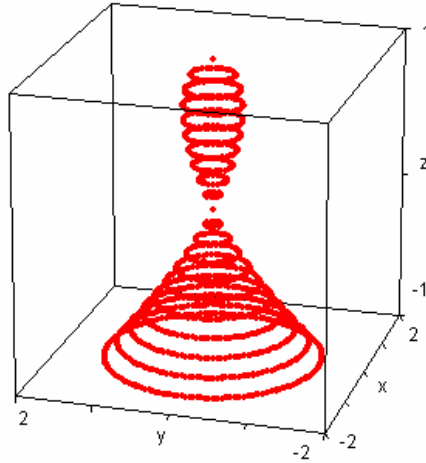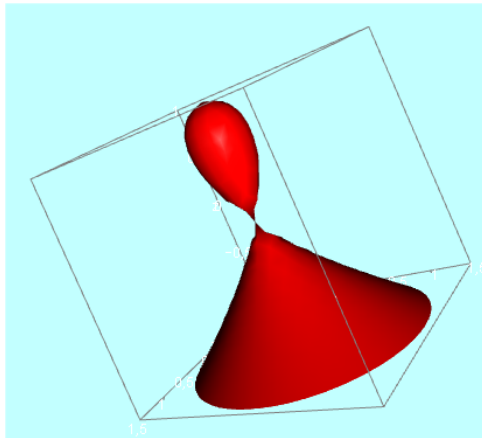


```
              2    2    2    3
ImplicitDots(x  + y  = z  - z , [-1.5, -1.5, -1], [1.5, 1.5, 1], 0.05)
```


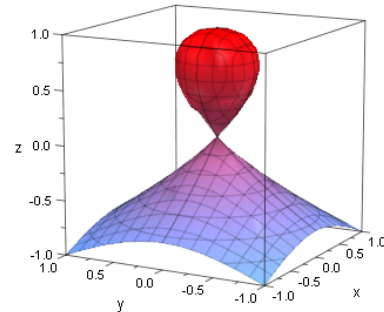Using `polycontour.mth` (DNL#63) and `IMPLICIT_Peter.mth` (DNL#64)
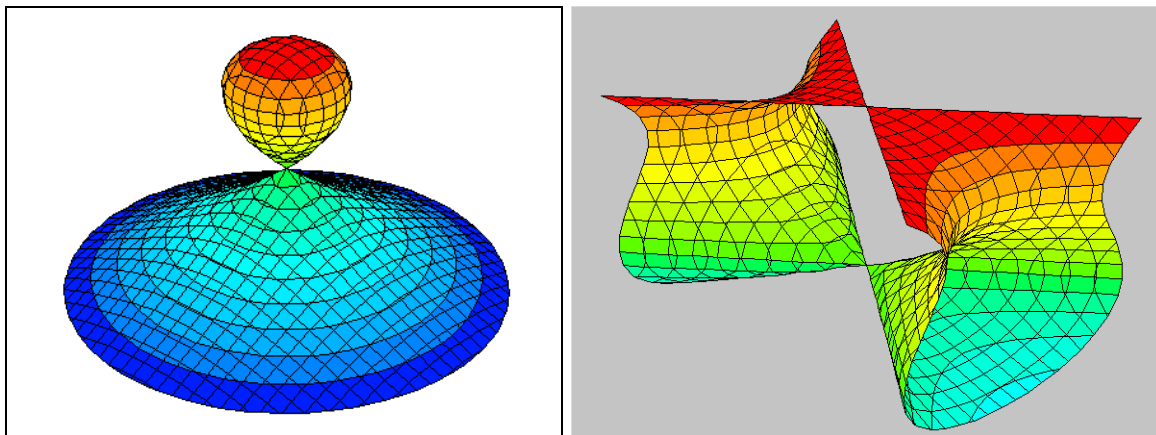
Plot with DPGraph. The right figure shows the plot of $a x^2 + b y^2 = c(z^2 - z^3)$ varying the parameters a, b and c using scroll bars.

FAZ - Fläche #3

```
plot(plot::Implicit3d(x^2+y^2=z^2-z^3,
        x=-1..1,y=-1..1,z=-1..1),
        Scaling = Constrained)
```

Autograph 3.20   MuPad

User Form (continued from page 4)

John Coleman

Greetings,

The function I came up with yesterday performs (as I suspected) poorly on large examples. For example, on my machine it took 124 seconds to evaluate reduce_matrix(A) when A was given by A:= vector([random(10),random(10)],i,1,1000). The attached file contains a programmed version which takes less than a second with the same example. Please let me know if you encounter any bugs.

m_reduce_02.dfw

```
     pre_process(v, u, f, i, j, n, e, x, w)
:=
       Prog
         u ≔ []
         f ≔ []
         i ≔ 1
         n ≔ DIM(v)
         Loop
           If i > n
              RETURN u
           e ≔ v↓i
           x ≔ e↓1
#1:        j ≔ POSITION(x, f)
           If j = false
              Prog
                f ≔ APPEND(f, [x])
                u ≔ APPEND(u, [[x, e↓2, 1]])
              Prog
                w ≔ u↓j
                w↓2 ≔ w↓2 + e↓2
                w↓3 ≔ w↓3 + 1
                u↓j ≔ w
           i ≔ i + 1
```

$$
\text{#2:} \quad av(u) ≔ \left[u_1 , \frac{u_2}{u_3} \right]
$$

$$
\text{#3:} \quad av\_all(A) ≔ \text{VECTOR}(av(A_i), i, 1, \text{DIM}(A))
$$

#4: reduce_matrix(A) ≔ av_all(pre_process(A))

#5: A ≔ VECTOR([RANDOM(10), RANDOM(10)], i, 1, 1000)

#6: reduce_matrix(A)

$$
\text{#7:} \quad \begin{bmatrix}
3 & 4.59047619 \\
7 & 4.551401869 \\
2 & 4.972222222 \\
4 & 4.885714285 \\
9 & 4.842105263 \\
1 & 4.740384615 \\
8 & 4.420454545 \\
6 & 4.034090909 \\
0 & 4.051020408 \\
5 & 4.705882352
\end{bmatrix}
$$

0.141 sec

-John Coleman

Johann Wiesenbauer   j.wiesenbauer@TUWIEN.AC.AT

Hello John,

Yes, it is true, your second version is considerably better than the first one. Even though its performance breaks dramatically down when dealing with really big lists whose number is say 10000 or more. Furthermore, people who know me won't be surprised that I also disagree with your abundant use of auxiliary functions as well as indexed variables. This may well be a matter of taste though.

Anyway, would you believe that it is possible to write a program of this sort that doesn't contain a single indexed variable? Well, it is and here comes the proof! (Note that I assumed the matrix A to be sorted. If this is not the case, then use convert(sort(A)) as in the example below.)

m_reduce_03.dfw

```
      convert(A, A_ := [], i_, s_, t_) :=
        Loop
          i_ := 0
          s_ := FIRST(A)
          t_ := FIRST(s_)
          Loop
#1:         A := REST(A)
            i_ :+ 1
            If A = []
               RETURN REVERSE(ADJOIN(s_/i_, A_))
            If FIRST(FIRST(A)) ≠ t_ exit
            s_ :+ FIRST(A)
          A_ := ADJOIN(s_/i_, A_)
```

```
#2:   A := VECTOR([RANDOM(10), RANDOM(10)], i, 1, 10000)
```

(Simplify this line and delete the output before going on!)

```
#3:   convert(SORT(A))
```

Anyone who wants to try it out himself is advised to simply copy and paste the lines above into the input line. I did only a few tests, but it seems to work and its speed is awesome. Please tell me though, if you come across any errors.

Cheers,
Johann

John Coleman

Hello Johann,
Your program is quite elegant and seems to be correct. It would be hard to break its speed record. I like your convention of using an underscore to indicate an identifier is a local variable rather than a genuine parameter.

I am a bit puzzled by what you meant by performance breaking down dramatically with larger lists. With 10,000 element examples our programs seem to run about the same (slightly over 1 second in both cases on my machine in some runs, sometimes closer to 1.5 seconds with mine, depending on how close to sorted my ultimate output is). I tried A:= vector([random(10),random(10)],i,1,100000) and got the following results:

convert(sort(A))   21.1 seconds

reduce_matrix(sort(A)) 33.0 seconds

reduce_matrix(A) 60.2 seconds

mine does seem 50% slower on large examples (in the sorted case to make them comparable). Is that what you meant by a dramatic breakdown - or did you observe even worse behavior? Just curious.

Regards,
-John Coleman

Alessandro Perotti   perotti@SCIENCE.UNITN.IT

Dear Derivers,

here is another efficient function which make the conversion asked by Marcelo Fernández:

m_reduce_04.dfw

```
        Convert(A, a_) :=
          If DIM(A) > 0
#1:           ADJOIN([A↓1↓1, AVERAGE(a_ := SELECT(IDENTICAL?(u↓1, A↓1↓1), u, A)'↓2)],
              []




          Convert(REST(A, DIM(a_))))


#2:     Convert(SORT(A))
```

On big sample data (>50000) it seems (slightly) more efficient than other solutions.

Best regards,
Alessandro Perotti

Johann Wiesenbauer

Hello John,

Thank you very much for your nice comments on my program. As to the "performance breaking down dramatically with larger lists", you are right, I should have been more precise. Below is a list of run times for various lengths k of the lists, where the lists I used are always of the form

```
A := VECTOR([RANDOM(10), RANDOM(10)], i, 1,  k)
```

The second and the third columns are the run times when approximating reduce_matrix(A) and convert(SORT(A)), respectively.

k	your version	my version	k	your version	my version
10,000	1.46 s	1.17 s	60,000	108.50 s	11.7 s
20,000	9.18 s	2.88 s	70,000	171.5 s	15.6 s
30,000	25.50 s	4.52 s	80,000	223.7 s	24.9 s
40,000	47.30 s	6.55 s	90,000	284.1 s	37.7 s
50,000	75.60 s	8.48 s	100,000	300.2 s	51.0 s

Hence, I should have rather said that the "first signs" of a breakdown of your program become visible near 10,000, but the actual breakdown is only later. Interestingly enough the relative performance difference between our programs is biggest near 70,000 and becomes smaller after that for some unknown reason. (Partly this may be due to the additional sorting of the list, which takes 7.62s for k=100,000.)

At any rate, I think that this has been a very fruitful discussion, in particular as regards the "power of indexfree programming". Thanks for your contributions.

Cheers,
Johann

Johann Wiesenbauer

Hello Alessandro,

Well, your program won't win a beauty contest, but it is admittedly the fastest one that has been been submitted so far as to this problem.
Needless to say that performance is the top priority for me, hence it's my turn again.

What about the following program that is at least as ugly, but seems to be again faster, e.g. 4.21s for the case k=100,000 mentioned in my previous email! (Note that presorting A is no longer needed as well!)

m_reduce_05.dfw

```
        conv(A) :=
          Prog
            A := REVERSE(TERMS(∑(x^FIRST(v)·(FIRST(REST(v))·y + z), v, A)))
#1:         If ∂(FIRST(A), x) = 0
                A := ADJOIN(FIRST(A) + FIRST(REST(A)), REST(REST(A)))
            VECTOR([x·∂(LN(v), x), SUBST(v, [x, y, z], [1, 1, 0])/SUBST(v, [x, y, z], [1,



        0, 1])], v, A)
#2:     conv(A)
```

Cheers,
Johann

Alessandro Perotti

Hello Johann (and all Derivers),

now another throw by me. It's a small improvement (less than 30'' on my PC for a 1 million unsorted table obtained approximating the line

```
        PROG(A:=VECTOR([RANDOM(10),RANDOM(10)],i,1,1000000),[])
```

m_reduce_06.dfw

```
#1:    PROG(A := VECTOR([RANDOM(10), RANDOM(10)], i, 1, 1000000), [])

#2:    []

       Convert(A, a_, b_, c_, t) :=
         Prog
           a_ := ∑(APPEND("x", v↓1)·(v↓2 + t), v, A)
#3:        b_ := ∂(a_, t)
           a_ := a_ – b_·t
           c_ := VARIABLES(b_)
           [MAP_LIST(DELETE(v, 1), v, c_), MAP_LIST(v↓1/v↓2, v, [GRAD(a_, c_), GRAD(b_, c_)]')]'

#4:    Convert(A)
```

Best regards,
Alessandro

Johann Wiesenbauer

Hello Alessandro,

Well, I see that you are exploting basically the same idea, but using sort of indexed variables instead of powers. In fact, this works fine as to our current "test suite" for which it seems to be "tailormade".

On the other hand, you have the serious restriction now that the first column of the matrix contains integers only as opposed to the other programs so far, which allowed even algebraic and transcedental numbers here. Hence, it's up to Marcelo now to decide whether he can live with this restriction or not.

Cheers,
Johann

Francisco Marcelo Fernández, Argentina

Dear derivers:

First of all, I want to thank you all for your interest in the problem of reducing a matrix with repeated entries.

I had a practical application in mind and was having some difficulties with my awful programs. When all the beautiful and fast proposals arrived, I had successfully prepared the experimental data for a further application of my (also poor) multi linear regression program using the first version of John Coleman's program for reducing the matrix. In my case the elements of the left column are real numbers (which Derive dutifully converts into rational ones) and are commonly sorted. The elements of the right column are also real numbers. I apologize that the left column of my example contained only integers. Efficiency is not an issue, I am patient enough to wait for a few seconds to get the reduced matrix. Besides, commonly the dimension of the experimental-data matrix is about 100-300 but my colleagues may eventually work harder and produce greater datasets.

I want to size this opportunity to say that Derive New list is a remarkable help for those who are not good programmers and want to use Derive for practical purposes.

Would you think I am a bore if I ask for a Derive program that makes linear regression with nonlinear parameters?. Mine is rather moody and decides not to converge more often than I like.

Some days ago I met Johann at the University at the occasion of the traditional meeting of the Institute. He regretted that he had not had time enough to compare calculation times of all presented versions of this matrix reducing routine. Johann, this is my Christmas present for you. I compared all six versions reducing the same matrix consisting of firstly 50,000 and then 100,000 pairs of data. This is the result:

Program	50,000 pairs	100,000 pairs
m_reduce_01	memory exhausted	memory exhausted
m_reduce_02	18.4 s	63.7 s
m_reduce_03	9.81 s	26.1 s
m_reduce_04	9.42 s	19.9 s
m_reduce_05	2.52 s	5.17 s
m_reduce_06	2.14 s	4.80 s
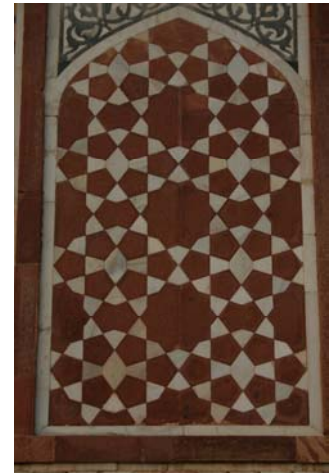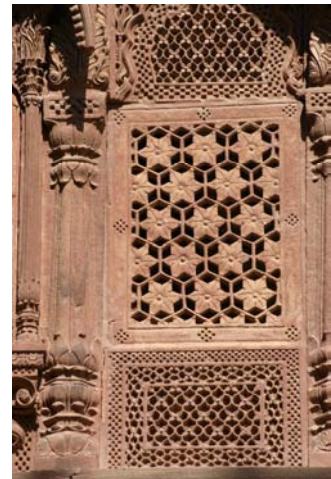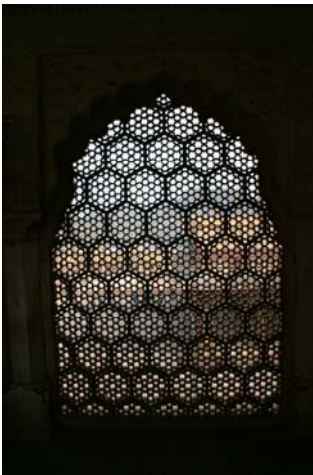
It needs only little work with a graphics program to rectificate the pictures, Josef