THE DERIVE - NEWSLETTER #30

ISSN 1990-7079

THE BULLETIN OF THE

DERIVE®

USER GROUP **+ TI 92**

Contents:

[1] **Analysis 1, Ein Arbeitsbuch mit *DERIVE***, Rüdeger Baumann, Klett Verlag, ISBN 3-12-739512-4.

[2] **Computereinsatz im anwendungsorientierten Unterricht**, Hellmut Scheuermann, diVerlag Franzbecker ISBN 3-88120-282-X

[3] **Lineare Algebra mit dem TI-92**, Eberhard Lehmann, Texas Instruments Deutschland, 85356 Freising

[4] **Calculus Concepts using DERIVE for Windows**, R.Freese and D. Stenenga, available at:
www.math.hawaii.edu/RDPublishing/ (still valid)

Richard Schorn wrote:
> >Who can factorize this polynomial?
> >$25x^{16}–184x^{14}+4028x^{12}–14600x^{10}+27862x^8–14600x^6+4028x^4–184x^2+25$
> >Hint: $x^4– 4x^3+6x^2+12x+5$ is one of the factors.
> ...

Albert Rich answered:
> By symmetry, I guessed and used DERIVE to verify that $x^4+4x^3+6x^2–12x+5$ is also a factor. DERIVE can then factor the resulting 8th degree polynomial using radical factoring.
> ...

Your answer shows that intelligent guessing is a valuable tool in mathematics, like DERIVE. Factoring the 8th degree polynomial (option radical) yields weird irrational coefficients. In fact $5x^4+12x^3+6x^2-4x+1$ is the third factor and now it is no problem to guess the last one. How can I achieve this result by using DERIVE??
BTW: The original polynomial of degree 16 is the square of the long diagonal in a rectangular box (cuboid??, in german "Quader") with rational edges and rational diagonals of the faces. If this polynomial is a perfect square (rational x, not 0,1,-1) one has solved an old problem.

Regards, Richard Schorn

Hi all, as for the factorization

$25x^{16}–184x^{14}+4028x^{12}–14600x^{10}+27862x^8–14600x^6+4028x^4–184x^2+25 =$

$= (x^4 + 4x^3 + 6x^2–12x + 5)\cdot(x^4 – 4x^3 + 6x^2 + 12x + 5)\cdot(5x^4 + 12x^3 + 6x^2 – 4x + 1) \cdot$
$\cdot(5x^4 – 12x^3 + 6x^2 + 4x + 1)$

I'm going to include in my next "Titbits" column in the DERIVE Newsletter #30 a detailed treatise on how to use DERIVE to compute this decomposition into irreducible polynomial over Q using powerful general algorithms.

There is only thing I would like to mention in advance. If you really use the hint given by Richard Schorn (it is not quite clear to me whether this is allowed in his opinion or not) that

$$x^4 – 4x^3 + 6x^2 + 12x + 5$$

is a factor of the given polynomial then the factorization given above is downright trivial. You don't believe it? Here you are!

(This mail is to be continued in the USER FORM 2, page 45)

Dear DERIVE and TI-friends

The DERIVE and TI-Conference 1997 has not even begun but we are preparing the ´99 Conference to be held in Austria. As I received the summary of the challenging scope and aims of the last conference in this millenium just now I decided to omit my letter and to "proudly" present the

First Announcement and Call for Papers for the Conference

Computer-Supported Mathematical Education
(Didactical, Mathematical, and Software Technological Aspects)

August, 23-25, 1999

(Organized by ACDCA - Austrian Center for Didactics of Computer Algebra
and RISC - Research Institute for Symbolic Computation)

Scope: This conference is a forum for papers on computer-supported mathematical education and emphasizes research and experimental work that tries to integrate the didactic, mathematical, and software technologic aspect of the subject.

The conference is the fifth in a sequence of conferences on computer-supported mathematical education initiated by ACDCA (Krems 1992, Krems 1993, Honolulu 1995, Kungsbacka 1997).

The aim of these conferences is twofold:

- The enormous possibilities of the new computer-based media on the improvement and innovation of teaching and learning mathematics should be promoted.
- The enormous expertise of teachers and students about the math teaching and learning process should be fully integrated into the development of new computer-based media.

The fifth conference in the series, prefers contributions that consider both aims in a balanced way.

Typical topics in the scope of the conference:

- Report about successful classroom experiments using mathematical software systems.
- The impact of math system design on mathematical thinking and problem solving.
- New ways of math teaching and learning on the basis of the new media.
- The use of network software for organizing new forms of math teaching and studying.
- Interactive and individualized generation of math learning material by the student.
- New algorithmic mathematics particularly suitable for computer-based learning.
- Improvements in the design of math software systems based on didactical experience.
- New software tools for facilitating the development of math teaching material.
- New software tools for interaction of math teachers and students via internet.
- New software for computer-supported math tutoring and evaluation.
- Examples of successful, new computer-based math texts, lectures, training units etc.

All these subjects are considered for both the secondary and the university level.

Bruno Buchberger, Dr phil, Dr hc
Professor of Mathematics
Chairman of the Research Institute for Symbolic Computation
Fax: ++43 732 2468 9930
E-mail: Buchberger@RISC.Uni-Linz.ac.at
http://www.risc.uni-linz.ac.at/people/buchberg/

Helmut Heugl, Dr.
Landesschulinspektor
Fax: ++43-2742-280-1111
E-mail: Helmut.Heugl@kem.ac.at
Helmut.Heugl@lsr-noe.gv.at

The *DERIVE-NEWSLETTER* is the Bulletin of the *DERIVE User Group*. It is published at least four times a year with a contents of 44 pages minimum. The goals of the *DNL* are to enable the exchange of experiences made with *DERIVE* as well as to create a group to discuss the possibilities of new methodical and didactical manners in teaching mathematics.
We have established a section dealing with the use of the *TI*-92 and we try to combine these modern technologies.

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

Next issue: September 1998
Deadline: 15 August 1998

Preview: Contributions for the next issues

Extracting Logic Propositions from Numerical Data, Etchells, UK
3D-Geometry, Reichel, AUT
Linear Programming, Various Projections, Word Problems, Böhm, AUT
A Utility file for complex dynamic systems, Lechner, AUT
Examples for Statistics, Roeloffs, NL
Fractals and other Graphics, Koth, AUT
Implicit Multivalue Bivariate Function 3D Plots, Biryukov, RUS
Riemann, a package for the TI-92, Böhm & Pröpper, AUT/GER
Parallel Curves, Wunderling, GER
Quaternion Algebra, Sirota, RUS
Concentric Curve Shading, Speck, NZL
Information Technologies in Geometry, Rakov & Gorokh, UKR
Parametric 3D-Plots with DERIVE and 3DV, Welke, GER
Various Training Programs for Students on the TI-92, Böhm, AUT
A Critical Comment on the "Delayed Assignation" := =, Kümmel, GER
Sand Dunes, River Meander and Elastica, The lighter side ...., Halprin, AUS
Type checking, finite continued fractions, ..., Welke, GER
Evaluation of the Modified Bessel Functions, Ibrahim, ESP
On the Resolution of the Linear Differential Equation ......, Candel, ESP

and
Setif, FRA; Vermeylen, BEL; Leinbach, USA; Weth, GER; Kirmse, GER
Aue, GER; Koller, AUT; Mitic, UK; Schorn, GER; Santonja, ESP;
Dorner, USA, Cappuccio, IT, Pröpper, GER and and and ....

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

Experimenting with GRAM - SCHMIDT

Steven Schonefeld, Auburn, USA

Some Background Information

For this discussion, we will assume that we have an inner product space, with the inner product of vectors u and v denoted by $<u,v>$. The inner product may be used to define the norm or length of a vector v as follows:

$$NORM(v) := \sqrt{<v,v>}.$$

Here are two inner products which may be defined on the vector space C[-1,1], the space of continuous functions on the interval $-1 \le x \le 1$.

IP(1) and IP(2): Suppose that $u = u(x)$ and $v = v(x)$ are in C[-1,1].

Either

$$<u,v> := \int_{-1}^{1} u(x)\, v(x)\, dx \qquad \text{IP(1)}$$

or

$$<u,v> := \int_{-1}^{1} \frac{u(x)\, v(x)}{\sqrt{1-x^2}}\, dx \qquad \text{IP(2)}$$

will define an inner product. The only difference between IP(1) and IP(2) is that the functions are multiplied by a "weight function" inside the integral in IP(2). This weight function gives more weight or importance to values of $u(x)$ and $v(x)$, for x near the ends of the interval [-1,1] and less weight in the middle of this interval. Each of these inner products permits us to compute an inner product distance between functions $u(x)$ and $v(x)$ as $NORM(u - v)$.

There is a third measure of distance in C[-1,1]. We define the **uniform distance** between functions $u(x)$ and $v(x)$ by:

$$UNIFORM\_DISTANCE(u,v) := \underset{-1 \le x \le 1}{MAX} \; |u(x) - v(x)|$$

The uniform distance is related to, but different from, the distance given by either IP(1) or IP(2). The uniform distance may be quite difficult to calculate exactly. However, we can usually approximate it from a plot of the graph of $u(x) - v(x)$ for $-1 \le x \le 1$.

The Gram-Schmidt Process

Two vectors u and v are called orthogonal if $<u,v>$ = zero. We call $V = [v_1, v_2, .., v_n]$ an orthogonal system if $<v_j, v_k>$ = zero for j different from k and each v_j is nonzero. The projection of w onto the span of the vectors in V is given by:

$$PROJECT(w,V) := \sum_{j=1}^{n} \frac{<w,v_j>}{<v_j,v_j>} v_j .$$

When distance is measured by the norm, the vector $z = PROJECT(w,V)$ is the (unique) vector in the span of V which is closest to w. In other words, the vector z satisfies:

$$NORM(z - w) \le NORM(v - w), \text{ for all } v \text{ in span}(V).$$

This vector z is the best approximation to w which can be found in the subspace spanned by V.

Suppose we have a finite set S of linearly independent vectors. We may apply the Gram-Schmidt Process to S in order to get an orthogonal system V having the same span as S. The interested reader may see any good book on Linear Algebra for a more detailed description of the Gram-Schmidt process. Procedures for automating the Gram-Schmidt process are contained in lines #4 and #5 below. When our "vectors" are functions in C[-1,1], the Gram-Schmidt process may help us approximate a fairly complicated function by a simpler function — such as a polynomial.

The Problem

Suppose that we wish to approximate the function `w(x) := ATAN(2x)`, in C[-1,1], with a polynomial $p(x)$ of degree seven so that the uniform distance between $p(x)$ and $w(x)$ is small. Here are four reasonable attempts to find such a polynomial — Attempt 3 and Attempt 4 rely on the Gram-Schmidt process.

Attempted Solutions

Attempt 1: Use a Taylor Polynomial.
The Taylor polynomial of degree seven (about $x = 0$) is shown on line #7 and is defined as TP(x). A parametric plot of the expression `[x,TP(x) - ATAN(2x)]` for parameter x going from -1 to 1 is shown in Figure 1. The Taylor polynomial is really close to ATAN(2x) for x near zero. However, the difference gets quite large at $x = \pm 1$. The uniform distance between TP(x) and ATAN(x) is approximately 14.

Attempt 2: Use an Interpolating Polynomial.
We find a polynomial which agrees with ATAN(2x) at specified x-coordinates (called nodes) by using a process called Lagrange Interpolation. (For more information on this process, see the book: **NUMERICAL ANALYSIS *via* DERIVE**, by Steven Schonefeld, MathWare, 604 E Mumford Dr., Urbana, IL 61801 Phone: 1-800-255-2468.) The interpolating polynomial using equally spaced nodes [-1, -0.75, … , 0.75, 1] (on line #14) is defined as LP(x). (We could also find this polynomial using `POLY_INTERPOLATE_EXPRESSION(u,x,a)` found in the utility file **MISC.MTH**.) A plot of `[x,LP(x) - ATAN(2x)]` is shown in Figure 2. The uniform distance between LP(x) and ATAN(2x) is approximately 0.014. You will notice that ATAN(2x) is an odd function. It makes sense that polynomials TP(x) and LP(x), when expanded, would contain only odd powers of x. Hence, we will use only odd powers of x in our last two polynomial approximations to ATAN(2x).

Attempt 3: Use IP(1)-Projection.
We define IP(1) on line #16 and apply the Gram-Schmidt process to the odd monomials `[x,x^3,x^5,x^7]` to get the odd orthogonal polynomials on line #18. The IP(1)-projection of ATAN(2x) onto the space of odd seventh-degree polynomials (on line #20) is defined as P1(x). A plot of `[x,P1(x) - ATAN(2x)]` is shown in Figure 3. The uniform distance between P1(x) and ATAN(x) is approximately 0.0068.

Attempt 4: Use IP(2)-Projection.

We define IP(2) on line #22 and again apply the Gram-Schmidt process to the odd monomials `[x,x^3,x^5,x^7]` to get the odd orthogonal polynomials on line #24. As might be expected, the orthogonal polynomials given by IP(2) are different than the ones given by IP(1). The IP(2)-projection of ATAN(2x) onto the space of odd seventh-degree polynomials (line #26) is defined as P2(x). A plot of `[x,P2(x) - ATAN(2x)]` is shown in Figure 4. The uniform distance between P2(x) and ATAN(x) is approximately 0.0041.

Remarks

Remark 1 When we use the inner product IP(1) in C[-1, 1], the vector of functions

`[1,COS(πx),SIN(πx),COS(2πx),...,COS(nπx),SIN(nπx)]`

forms an orthogonal system. Linear combinations of these trigonometric functions may be used for approximating periodic functions on the real line —— and more. The study of such approximations is included in the general category: **Fourier Analysis**.

Remark 2 When we use inner product IP(1) in C[-1, 1], the Gram-Schmidt process applied to the system `[1,x,x^2,x^3,...,x^n]` results in an orthogonal system of **Legendre polynomials** —— really scalar multiples of the Legendre polynomials. An alternate definition of the Legendre polynomial of degree n is given by:

$$LEGENDRE(x,n) := 2^n \sum_{k=0}^{[n/2]} \frac{(-1^k)(2n-2k)!}{(n-k)!k!(n-2k)!}$$

where [ ] denotes the greatest integer function (= FLOOR-function).

Remark 3 When we use inner product IP(2) in C[-1,1], the Gram-Schmidt process applied to the system `[1,x,2x^2,4x^3,...,2^(n-1) x^n]` results in the orthogonal system of **Chebyshev polynomials** —— which may be written in trigonometric form as follows:

`[1,COS(ACOS(x)),COS(2 ACOS(x)),...,COS(n ACOS(x))].`

Remark 4 When we use inner product IP(1) on C[-1,1], the Gram-Schmidt process applied to the system:

`[1,x,|x|,|2x+1|,|2x-1|,|4x+3|,|4x+1|,|4x-1|,|4x-3|]`

will yield an orthogonal system of continuous functions comprised of straight line segments. These functions may have corners at x = 0, ±1/4, ±1/2, ±3/4. The projection of a function $w(x)$ onto the span of this orthogonal system results in a piecewise linear approximation to $w(x)$.

Remark 5 Suppose that $u = u(x, y)$ and $v = v(x, y)$ are in C([0,1]^2), the space of continuous functions on the square $[0,1]^2 = \{(x, y) \mid 0 \le x \le 1, 0 \le y \le 1\}$. Then,

$$< u, v > := \int_0^1 \int_0^1 u(x,y)v(x,y)\,dy\,dx$$

will define an inner product on this space. An application of `GRAM_SCHMIDT()` to linearly independent polynomials in C([0,1]^2) may help us approximate a function $w(x, y)$ by a polynomial in x and y.

Exercises:

Exercise 1 Repeat Attempt 3 with the following modifications. This time, apply `GRAM_SCHMIDT()` to the vector of monomials `[1,x,x^2,x^3,x^4,x^5,x^6,x^7]` and project ATAN($2x$) onto the span of the resulting orthogonal set. Is your result the same as P1(x)?

Exercise 2 Using the inner product IP(1), verify that the polynomial P1(x) minimizes the norm of the difference

$$a\,x + b\,x^3 + c\,x^5 + d\,x^7 - \text{ATAN}(2x)$$

over all possible choices of a, b, c, and d as follows. Define PP(x) to be the general polynomial:

```
PP(x) := a x + b x^3 + c x^5 + d x^7
```

Author and simplify `NORM(PP(x) - ATAN(2x))^2` which is the quantity:

```
INNER_PRODUCT(PP(x) - ATAN(2x), PP(x) - ATAN(2x)).
```

The resulting expression will be a function of variables a, b, c, and d —— call it $\phi(a, b, c, d)$. Find the minimum of this function by simplifying

```
GRAD(ϕ(a,b,c,d), [a,b,c,])
```

(which gives the vector $[\phi_a, \phi_b, \phi_c, \phi_d]$ of partial derivatives of function ϕ.)
Solve for `[a, b, c, d]` and put these values into the general polynomial PP(x). The result should be P1(x).

Exercise 3 Repeat Attempt 1 - Attempt 4 with the function `w(x) := ASIN(x)` in place of ATAN($2x$).

Exercise 4 Approximate the even function `w(x) := SQRT(1 - x^2)` with a polynomial of degree eight in x. In Attempt 3 and Attempt 4, you will need to apply the Gram-Schmidt process to the vector of even monomials: `[1,x^2,x^4,x^6,x^8]` and project $w(x)$ onto the span of the resulting orthogonal system.

Exercise 5 Repeat Attempt 1 – Attempt 4 with the function `w(x) := ABS(3x - 1)` in place of ATAN($2x$). In Attempt 3 and Attempt 4, you will need to apply the Gram-Schmidt process to `[1,x,x^2,...,x^7]` and project $w(x)$ onto the span of the resulting orthogonal system. Note: this $w(x)$ has a corner at $x = 1/3$, making it somewhat difficult to approximate with a polynomial.

The DERIVE Procedures:

Gram-Schmidt process by Steven Schonefeld 1996, revised by J. Böhm 2013

Define the norm or 'length' of a vector in terms of the inner product.

```
#1:   CaseMode := Sensitive

#2:   INNER_PRODUCT(v, w) :=

#3:   NORM(v) := √(INNER_PRODUCT(v, v))
```

Projection of w onto the span of the orthogonal vectors in V.

#4: $$\text{PROJECT}(w, V) := \sum_{j=1}^{\text{DIM}(V)} \frac{\text{INNER\_PRODUCT}(V_j, w)}{\text{INNER\_PRODUCT}(V_j, V_j)} \cdot V_j$$

Apply GRAM_SCHMIDT() to a vector of independent vectors to get an orthogonal system.

#5: $$\text{GRAM\_SCHMIDT}(s) := \text{ITERATE}\left(\text{APPEND}\left(w, \left[s_{\text{DIM}(w)+1} - \text{PROJECT}(s_{\text{DIM}(w)+1}, w)\right]\right), w, [s_1], \text{DIM}(s) - 1\right)$$

Attempt 1

#6: TP(x) := TAYLOR(ATAN(2·x), x, 0, 7)

#7: $$\text{TP}(x) := -\frac{128 \cdot x^7}{7} + \frac{32 \cdot x^5}{5} - \frac{8 \cdot x^3}{3} + 2 \cdot x$$

#8: IF(|x| ≤ 1, TP(x) - ATAN(2·x), ?)

Attempt 2

#9:
```
LAGR_1(m_, i, j, x) :=
  If i = j
     1
     (x - m_↓j↓1)/(m_↓i↓1 - m_↓j↓1)
```

#10: $$\text{LAGR\_2}(m\_, i, x) := \prod_{j=1}^{\text{DIM}(m\_)} \text{LAGR\_1}(m\_, i, j, x)$$

#11: $$\text{LAGRANGE}(m\_, x) := \sum_{i=1}^{\text{DIM}(m\_)} m\_{}_{i,2} \cdot \text{LAGR\_2}(m\_, i, x)$$

#12: LAGRANGE(VECTOR([t, ATAN(2·t)], t, -1, 1, 0.25), x)

#13: LP(x) := LAGRANGE(VECTOR([t, ATAN(2·t)], t, -1, 1, 0.25), x)

#14: $$\text{LP}(x) := \frac{32 \cdot x \cdot (x+1) \cdot (1-x) \cdot (2 \cdot x+1) \cdot (2 \cdot x-1) \cdot (4 \cdot x+1) \cdot (4 \cdot x-1) \cdot \text{ATAN}\left(\frac{1}{5}\right)}{105} + \frac{x \cdot (2 \cdot x+1) \cdot (2 \cdot x-1) \cdot (4 \cdot x+3) \cdot (4 \cdot x-3) \cdot (16 \cdot x^2-15) \cdot \text{ATAN}\left(\frac{1}{3}\right)}{21} - \frac{\pi \cdot x \cdot (5120 \cdot x^6 - 9856 \cdot x^4 + 6020 \cdot x^2 - 1599)}{1260}$$

#15: IF(|x| ≤ 1, LP(x) - ATAN(2·x), ?)

Attempt 3

Redefine the inner product:

#16: $\text{INNER\_PRODUCT}(v, w) := \int_{-1}^{1} v \cdot w \, dx$

#17: $\text{P1\_}(x) := \text{GRAM\_SCHMIDT}([x, x^3, x^5, x^7])$

#18: $\text{P1\_}(x) := \left[x, x^3 - \frac{3 \cdot x}{5}, \frac{x \cdot (63 \cdot x^4 - 70 \cdot x^2 + 15)}{63}, \frac{x \cdot (429 \cdot x^6 - 693 \cdot x^4 + 315 \cdot x^2 - 35)}{429}\right]$

#19: $\text{P1}(x) := \text{PROJECT}(\text{ATAN}(2 \cdot x), \text{P1\_}(x))$

#20:

$$\text{P1}(x) := - \frac{15 \cdot x \cdot (10486905 \cdot x^6 - 18315297 \cdot x^4 + 9442895 \cdot x^2 - 1377495) \cdot \text{ATAN}\left(\frac{1}{3}\right)}{524288} - \frac{x \cdot (32175 \cdot x^6 \cdot (171115 \cdot \pi - 756152) + 3003 \cdot x^4 \cdot (14135704 - 3201975 \cdot \pi) + 385 \cdot x^2 \cdot (12876675 \cdot \pi - 56651128) - 105 \cdot (6887475 \cdot \pi - 29128952))}{73400320}$$

#21: $\text{IF}(|x| \le 1, \text{P1}(x) - \text{ATAN}(x), ?)$

Attempt 4

Redefine the inner product:

#22: $\text{INNER\_PRODUCT}(v, w) := \int_{-1}^{1} \frac{v \cdot w}{\sqrt{(1 - x^2)}} \, dx$

#23: $\text{P2\_}(x) := \text{GRAM\_SCHMIDT}([x, x^3, x^5, x^7])$

#24: $\text{P2\_}(x) := \left[x, x^3 - \frac{3 \cdot x}{4}, \frac{x \cdot (16 \cdot x^4 - 20 \cdot x^2 + 5)}{16}, \frac{x \cdot (64 \cdot x^6 - 112 \cdot x^4 + 56 \cdot x^2 - 7)}{64}\right]$

#25: $\text{P2}(x) := \text{PROJECT}(\text{ATAN}(2 \cdot x), \text{P2\_}(x))$

#26:

$$\text{P2}(x) := - \frac{x \cdot (960 \cdot x^6 \cdot (13 \cdot \sqrt{5} - 29) + 672 \cdot x^4 \cdot (78 - 35 \cdot \sqrt{5}) + 140 \cdot x^2 \cdot (95 \cdot \sqrt{5} - 211) - 2205 \cdot \sqrt{5} + 4725)}{105}$$

#27: $\text{IF}(|x| \le 1, \text{P2}(x) - \text{ATAN}(x), ?)$

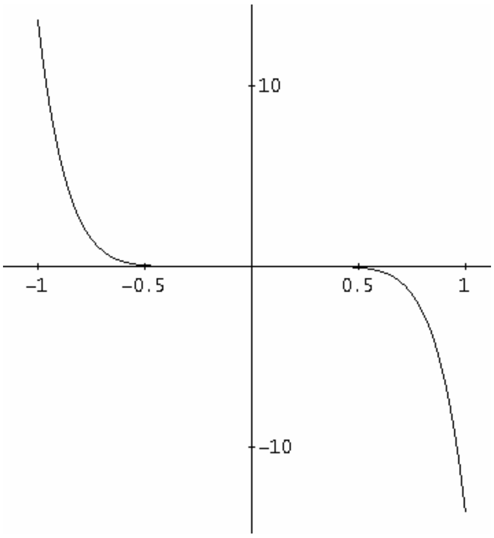


Figure 1 Graph of TP(x) – ATAN(2·x)
for $-1 \le x \le 1$

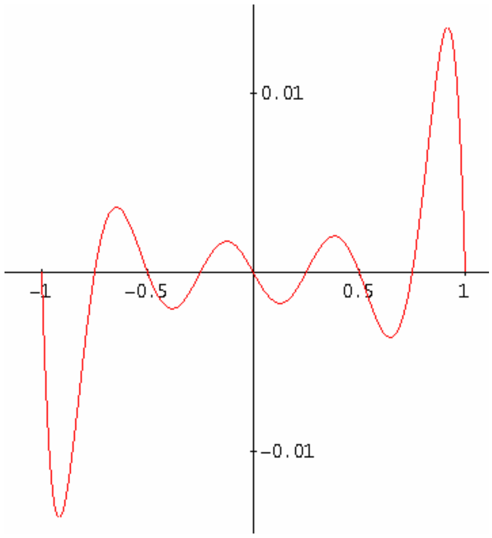


Figure 2 Graph of LP(x) – ATAN(2·x)
for $-1 \le x \le 1$

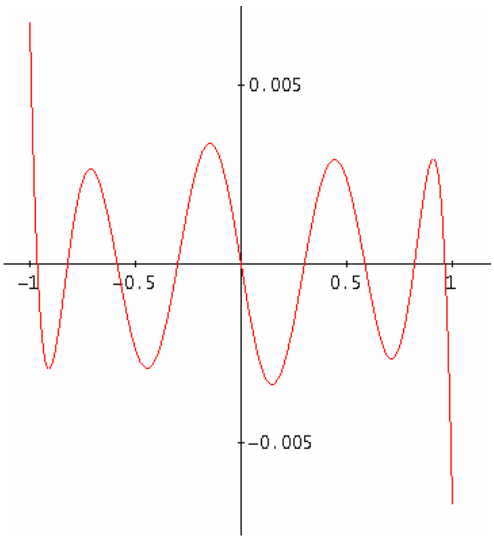


Figure 3 Graph of P1(x) – ATAN(2·x)
for $-1 \le x \le 1$

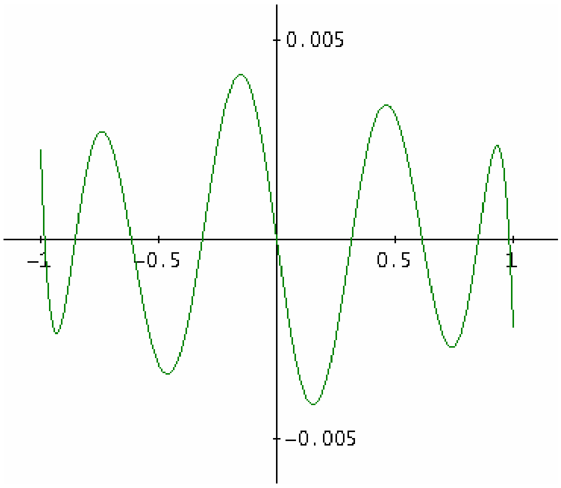


Figure 3 Graph of P1(x) – ATAN(2·x)
for $-1 \le x \le 1$

Some websites dealing with the Gram-Schmidt process:

http://www.math.hmc.edu/calculus/tutorials/gramschmidt/

http://www.math.psu.edu/sellersj/courses/math220/sp13/lecture_notes/sec6_4.pdf

http://www.math.psu.edu/mengesha/Math2025/Lecture7.pdf

http://en.wikibooks.org/wiki/Statistics/Numerical_Methods/Basic_Linear_Algebra_and_Gram-Schmidt_Orthogonalization

http://www.cs.usask.ca/~spiteri/M313/notes/Lecture8.pdf

GRAM-SCHMIDT on the TI-NspireCAS

Josef Böhm

Converting Steven's DERIVE file to a TI-NspireCAS file is interesting because of two major problems:

(1) We miss – one of my favourite DERIVE commands – the VECTOR command (#13).

(2) We also must do without ITERATE (#5).

```
vector
Define vector(v_,v,st,e,d)=
Func
Local vec
vec:= lim (v_)
      v→st
Loop
  st:=st+d
  If st>e
  Goto end
  vec:=colAugment(vec, lim (v_))
                       v→st
EndLoop
Lbl end
vec
EndFunc
```

```
gram_schmidt
Define gram_schmidt(s)=
Func
Local gs,i
gs:={s[1]}
For i,1,dim(s)-1
gs:=augment(gs,{s[dim(gs)+1]-project(s[dim(gs)+1],gs)})
EndFor
EndFunc
```

$vector([a \;\; a^2 \;\; a^3],a,1,11,2)$ $\quad \begin{bmatrix} 1 & 1 & 1 \\ 3 & 9 & 27 \\ 5 & 25 & 125 \\ 7 & 49 & 343 \\ 9 & 81 & 729 \\ 11 & 121 & 1331 \end{bmatrix}$

$vector([\sqrt{x}],x,1,2,0.2)$ $\quad \begin{bmatrix} 1 \\ 1.095445 \\ 1.183216 \\ 1.264911 \\ 1.341641 \\ 1.414214 \end{bmatrix}$

$vector(vector([a \cdot b],a,1,3,1),b,1,3,1)$ $\quad \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 4 \\ 6 \\ 3 \\ 6 \\ 9 \end{bmatrix}$

You can see that "my" *vector* command works but it does not offer the same flexibility as in DERIVE. However, it is sufficient for now.

Instead of ITERATE I have to program a respective function operating with a loop.

Take care: You have to enter the list of functions in gram_schmidt as a list – { }!

Note: Don't forget the difference between the DERIVE "IF" and the Nspire "when".

$inner\_product(v,w):=\int_{-1}^{1}(v\cdot w)\,dx$ — *Done*

$norm\_(v):=\sqrt{inner\_product(v,v)}$ — *Done*

$project(w,v\_):=\sum_{j=1}^{\dim(v\_)}\left(\frac{inner\_product(v\_[j],w)}{inner\_product(v\_[j],v\_[j])}\cdot v\_[j]\right)$ — *Done*

$tp(x):=\mathrm{taylor}(\tan^{-1}(2\cdot x),x,7,0)$ — *Done*

$tp(x)$ $\quad \frac{-128\cdot x^7}{7}+\frac{32\cdot x^5}{5}-\frac{8\cdot x^3}{3}+2\cdot x$

$$lagr\_1(m\_,i,j,x):=\text{when}\left(i=j,1,\frac{x-m\_[j,1]}{m\_[i,1]-m\_[j,1]}\right) \quad \textit{Done}$$

$$lagr\_2(m\_,i,x):=\prod_{j=1}^{\dim(m\_)[1]}\left(lagr\_1(m\_,i,j,x)\right) \quad \textit{Done}$$

$$lagrange(m\_,x):=\sum_{i=1}^{\dim(m\_)[1]}\left(m\_[i,2]\cdot lagr\_2(m\_,i,x)\right) \quad \textit{Done}$$

$$lp(x):=lagrange\left(vect\left(\left[t\quad \tan^{-1}(2\cdot t)\right],t,-1,1,\frac{1}{4}\right),x\right) \quad \textit{Done}$$

$lp(x)$

$$\frac{x\cdot\left(2048\cdot\left(6\cdot\tan^{-1}\left(\frac{2}{3}\right)-15\cdot\tan^{-1}\left(\frac{1}{2}\right)+\pi\right)\cdot x^6-896\cdot\left(18\cdot\tan^{-1}\left(\frac{2}{3}\right)-5\cdot\left(12\cdot\tan^{-1}\left(\frac{1}{2}\right)-\pi\right)\right)\cdot x^4+56\cdot\left(72\cdot\tan^{-1}\left(\frac{2}{3}\right)-49\right.}{630}$$

$$gram\_schmidt(\{x,x^3,x^5,x^7\}) \quad \left\{x,x^3-\frac{3\cdot x}{5},\frac{x\cdot(63\cdot x^4-70\cdot x^2+15)}{63},\frac{x\cdot(429\cdot x^6-693\cdot x^4+315\cdot x^2-35)}{429}\right\}$$

$$p1(x):=project\left(\tan^{-1}(2\cdot x),gram\_schmidt(\{x,x^3,x^5,x^7\})\right) \quad \textit{Done}$$

$p1(x)$

$$\frac{x\cdot\left(32175\cdot\left(342230\cdot\tan^{-1}\left(\frac{1}{2}\right)-171115\cdot\pi+378076\right)\cdot x^6-3003\cdot\left(6403950\cdot\tan^{-1}\left(\frac{1}{2}\right)-3201975\cdot\pi+7067852\right)\cdot x\right.}{36}$$

$$p1\left(\frac{3}{10}\right) \quad \text{undef}$$

My *gram_schmidt*-function seems to work, and so does the *project*-function. But there is no function value, why that? I don't know. I approximate *p1*(x) … and now it works!

$$p1(x):=-0.72196675974456\cdot x\cdot(x^6-2.5243495944637\cdot x^4+2.7204333460522\cdot x^2-2.720245700154) \quad \textit{Done}$$

$$p1(0.3) \quad 0.540419$$

For *p2*(x) I have to redefine the *inner_product* (like with *DERIVE*) and then repeat the procedure:

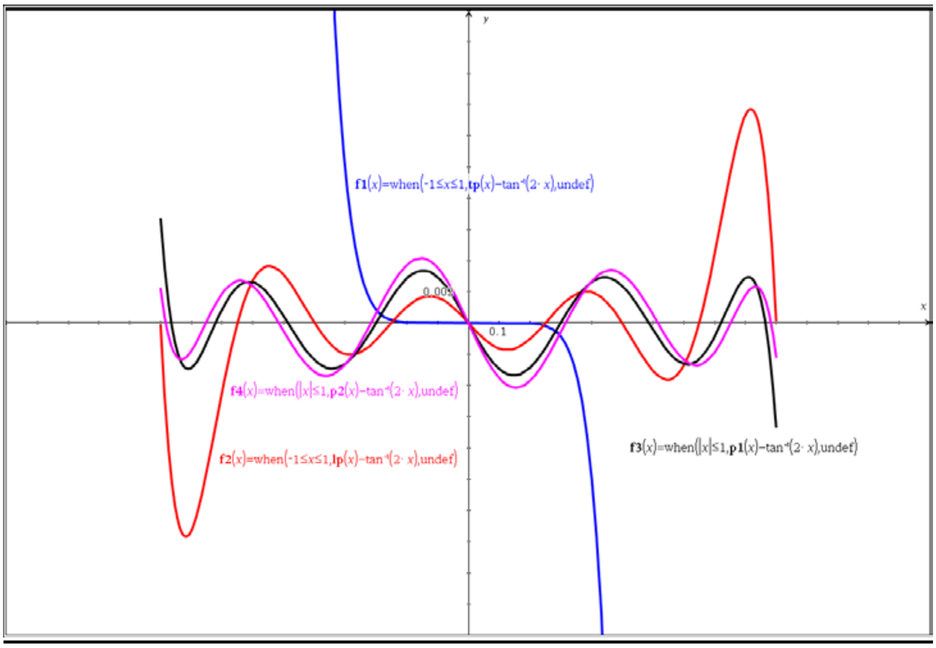
$$inner\_product(v,w):=\int_{-1}^{1}\frac{v\cdot w}{\sqrt{1-x^2}}\,dx \quad \textit{Done}$$

$$gram\_schmidt(\{x,x^3,x^5,x^7\}) \quad \left\{x,x^3-\frac{3\cdot x}{4},\frac{x\cdot(16\cdot x^4-20\cdot x^2+5)}{16},\frac{x\cdot(64\cdot x^6-112\cdot x^4+56\cdot x^2-7)}{64}\right\}$$

$$p2(x):=project\left(\tan^{-1}(2\cdot x),gram\_schmidt(\{x,x^3,x^5,x^7\})\right) \quad \textit{Done}$$

$$p2(x) \quad -0.629794\cdot x\cdot(x^6-2.666312\cdot x^4+3.019946\cdot x^2-3.108045)$$

$$p2(x):=-0.62979389712032\cdot x\cdot(x^6-2.6663118960508\cdot x^4+3.0199464091322\cdot x^2-3.1080446119652) \quad \textit{Done}$$



Finally I wanted to try my Nspire procedures (and my freshly acquired knowledge, of course) and tackled Steven's Exercise 5:

© Exercise 5

$$gram\_schmidt(\{1,x,x^2,x^3,x^4,x^5,x^6,x^7\})$$

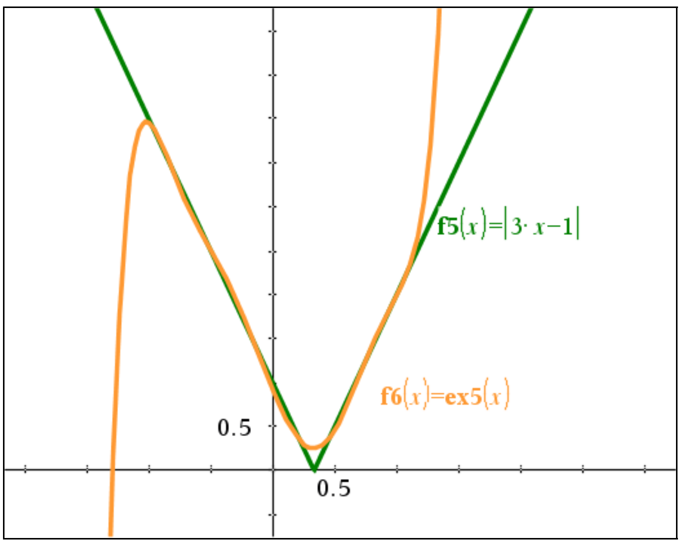
$$\left\{1,x,x^2-\frac{1}{2},x^3-\frac{3\cdot x}{4},x^4-x^2+\frac{1}{8},\frac{x\cdot(16\cdot x^4-20\cdot x^2+5)}{16},x^6-\frac{3\cdot x^4}{2}+\frac{9\cdot x^2}{16}-\frac{1}{32},\frac{x\cdot(64\cdot x^6-112\cdot x^4+56\cdot x^2 \ldots}{64}\right.$$

$$ex5(x):=project(|3\cdot x-1|,gram\_schmidt(\{1,x,x^2,x^3,x^4,x^5,x^6,x^7\}))$$ *Done*

$$ex5(x)$$

$$3.492820\cdot x^7-1.312819\cdot x^6-8.515255\cdot x^5+1.400139\cdot x^4+7.482464\cdot x^3+1.964711\cdot x^2-3.449201\cdot x+0. \ldots$$

$$ex5(x):=3.4928203441751\cdot x^7-1.3128186810168\cdot x^6-8.515255114912\cdot x^5+1.4001391895937\cdot x^4+7.4 \ldots$$ *Done*



You are invited to try the other exercises, too.

Josef

SUPER DUPER OSCULANTS 2

David Halprin, North Balwyn, Australia

Similarly, a second order contact meant that the two curves shared a gradient and curvature only, and when we got to $\frac{d\rho}{ds}$ this was the third order where they differed, and he expressed this mathematically, rather than $\frac{d\rho}{ds}$ he used $\frac{d\rho}{d\phi}$, which was the same as ρ_1, the first evolute.

Hence to go higher, even if we differentiated with regard to s, we had to convert the resulting differentials into ρ_2, ρ_3, ρ_4 etc., (to follow Cesaro's lead), which meant that we had to divide by ρ on the first time, before the next differentiation, and then we have a quotient to differentiate, viz:-

$$\rho' = \frac{d\rho}{ds} = \frac{\frac{d\rho}{d\phi}}{\frac{ds}{d\phi}} = \frac{\dot{\rho}}{\rho} = \frac{\rho_1}{\rho}; \qquad \rho'' = \frac{d\rho'}{ds} = \frac{\frac{d\rho'}{d\phi}}{\frac{ds}{d\phi}} = \frac{\rho \cdot \rho_2 - \rho_1^2}{\rho^3}$$

He did not realise that there was no advantage to be gained in doing this conversion, since one obtained exactly the same result by differentiating with respect to s and leaving the ρ' or ρ'' as is since, on substituting in the values for a nominated point, one came back to exactly the same state of affairs.

This should have shown him that he had nearly stumbled on all the other alternatives, that I shall include here, since there are a total of six possible expressions, five of which are really different from one another. With each expression one has a potential means of obtaining the osculant, and there will be one method more powerful than another in some special cases, and vice versa. This depends entirely in what form the Intrinsic equation of the curve is given. Namely, Cesaro Type 1 or 2, Whewell Type 1 or 2 and Euler Type 1 or 2.

Cesaro's published method is obviously for the Cesaro Equation Type 1, $\rho = C(s) = C_s$ that is why he differentiated with respect to s and obtained

$$\frac{dx}{ds}, \frac{d^2x}{ds^2}, \frac{d^3x}{ds^3} \ldots \ldots \frac{d^nx}{ds^n} \text{ and } \frac{dy}{ds}, \frac{d^2y}{ds^2}, \frac{d^3y}{ds^3} \ldots \ldots \frac{d^ny}{ds^n}.$$

N.B. There are two procedures that must be put together carefully:-

1) The successive differentiations of the coordinate with respect to an intrinsic variable, with the intention of their evaluation at a particular point so as to provide numerical answers, (which, thereby, can be available in the second procedure, below).

2) The construction of two Taylor series, [(one for x and one for y), in ascending powers of the intrinsic variable of choice], whose coefficients are the numerical values of the ascending series of derivatives, being evaluated at a nominated point. e.g. s = 0 and/or $\phi = 0$ or $\pi/2$.

As a consequence, one can combine these two procedural results into a completely general form by introducing a dummy variable, (say 't'), in place of the intrinsic variable in the Taylor series, leaving the original intrinsic variable only in all the terms, that were derived from the initial intrinsic equation for the base curve, instead of substituting in numerical values, as one would do if evaluating the derivatives for a particular point on the curve, such as $\phi = \phi_1$ (say).
Usually we choose $\phi_1 = 0$ since we require the same simplification, that Cesaro utilised, whereby this meant that $\sin(\phi) = 0$ when $\phi = 0$ and therefore we gladly lose approximately half the terms. However there are instances where we cannot have the intrinsic variable equal to zero, so we must evaluate about another point, bearing in mind

1) that the all the terms will be needed, there being no simplification.

2) that the first term of the Taylor series does not disappear, as it does in the Cesaro approach, hence we are displacing (translating) the curve, but this does not affect its morphology, so it is of no matter.

To use Derive or Derive XM, declare C = C(s).

Remember $C_s = \rho_s = \rho = \frac{ds}{d\phi} \therefore \phi = \int \frac{ds}{C_s}$. Differentiate with respect to 's' using Derive, as many times as necessary, (say up to 6th. or 7th. order), for greatest approximation to original (base) curve,

1.00 $\rho = C_{(s)} = C_S = C \ , \ \frac{dC}{ds} = C'$

1.01 $\frac{dx}{ds} = \cos\phi = \cos\left(\int \frac{ds}{C_S}\right)$

1.11 $\frac{dy}{ds} = \sin\phi = \sin\left(\int \frac{ds}{C_S}\right)$ and so on for higher derivatives of 'x' and 'y'.

Hence, for $\phi = 0$

1.21 $$x = t - \frac{t^3}{6C^2} + \frac{t^4 C'}{8C^3} + \frac{t^5}{120C^4}\left[4CC'' - 11(C')^2 + 1\right] + \frac{t^6}{144C^5}\left[C^2 C''' - 8CC'C'' + 10(C')^3 - 2C'\right]$$

1.22 $$y = \frac{t^2}{2C} - \frac{t^3 C'}{6C^2} - \frac{t^4}{24C^3}\left[CC''\right] - \frac{t^5}{120C^4}\left[C^2 C''' - 6CC'C'' - 6C' + 6(C')^3\right] -$$
$$-\frac{t^6}{720C^5}\left\{C^3 C'''' - 2C^2\left[4C'C''' + 3(C'')^2\right]\right\} - \frac{t^6}{720C^5}\left\{2CC''\left[18(C')^2 - 5\right] - 24(C')^4 + 35(C')^2 - 1\right\}$$

If, however you have Cesaro Type 2, $s = D(\rho) = D_\rho$, then you require

$$\frac{dx}{d\rho}, \frac{d^2x}{d\rho^2}, \frac{d^3x}{d\rho^3} \dots \dots \frac{d^nx}{d\rho^n} \quad \text{and} \quad \frac{dy}{d\rho}, \frac{d^2y}{d\rho^2}, \frac{d^3y}{d\rho^3} \dots \dots \frac{d^ny}{d\rho^n}.$$

To use Derive or Derive XM, declare D = D(ρ), differentiate with respect to ρ using Derive

2.00 $$s = D(\rho) = D_\rho = D \;,\; \frac{d\phi}{d\rho} = \frac{d\phi}{ds} \cdot \frac{dD}{d\rho} = \frac{\frac{dD}{d\rho}}{\rho}$$

2.01 $$\therefore \phi = \int \left(\frac{\frac{dD}{d\rho}}{\rho} \right) . d\rho$$

2.02 $$\frac{dx}{d\rho} = \frac{dx}{ds} \cdot \frac{ds}{d\rho} = \frac{ds}{d\rho} . cos\,\phi = \frac{dD}{d\rho} . cos\,\phi = \frac{dD}{d\rho} . cos \left[\int \left(\frac{\frac{dD}{d\rho}}{\rho} \right) d\rho \right]$$

2.11 $$\frac{dy}{d\rho} = \frac{dD}{d\rho} . sin\,\phi = \frac{dD}{d\rho} . sin \left[\int \left(\frac{\frac{dD}{d\rho}}{\rho} \right) . d\rho \right]$$ and so on for higher derivatives of 'x' and 'y'.

Hence, for $\phi = 0$

2.21 $$x = t . \frac{dD}{d\rho} + \frac{t^2}{2} . \frac{d^2D}{d\rho^2} + \frac{t^3}{6} . \left[\frac{d^3D}{d\rho^3} - \frac{1}{\rho^2} . \left(\frac{dD}{d\rho} \right)^3 \right]$$

2.22 $$y = \frac{t^2}{2} . \left[\frac{1}{\rho} \cdot \left(\frac{dD}{d\rho} \right)^2 \right] + \frac{t^3}{6} . \left[\frac{3}{\rho} \cdot \frac{dD}{d\rho} \frac{d^2D}{d\rho^2} - \frac{1}{\rho^2} \left(\frac{dD}{d\rho} \right)^2 \right] +$$
$$+ \frac{t^4}{24} \left[\frac{3}{\rho} \left(\frac{d^2D}{d\rho^2} \right)^2 - \frac{1}{\rho^3} \left(\frac{dD}{d\rho} \right)^4 + \frac{4}{\rho} \cdot \frac{dD}{d\rho} \frac{d^3D}{d\rho^3} - \frac{5}{\rho^2} \cdot \frac{dD}{d\rho} \frac{d^2D}{d\rho^2} + \frac{2}{\rho^3} \left(\frac{dD}{d\rho} \right)^2 \right]$$

If you have Whewell Type 1, $s = V(\phi) = V_\phi$, then you require

$$\frac{dx}{d\phi}, \frac{d^2x}{d\phi^2}, \frac{d^3x}{d\phi^3} \dots \dots \frac{d^nx}{d\phi^n} \quad \text{and} \quad \frac{dy}{d\phi}, \frac{d^2y}{d\phi^2}, \frac{d^3y}{d\phi^3} \dots \dots \frac{d^ny}{d\phi^n}.$$

To use *DERIVE* or *DERIVE* XM, declare V = V(ϕ), differentiate with respect to ϕ using *DERIVE*.

3.00 $$s = V(\phi) = V_\phi = V \;,\; \frac{ds}{d\phi} = \rho = \dot{V}_\phi = \dot{V}$$

3.01 $$\frac{dx}{d\phi} = \rho . cos\,\phi = \dot{V} . cos\,\phi$$

3.02 $$\frac{d^2x}{d\phi^2} = \ddot{V} . cos\,\phi - \dot{V} . sin\,\phi$$

3.03 $\dfrac{d^3x}{d\phi^3} = cos\,\phi.(\dddot{V} - \dot{V}) - 2\ddot{V}.sin\,\phi$

3.04 $\dfrac{d^4x}{d\phi^4} = cos\,\phi.\left(\dfrac{d^4V}{d\phi^4} - 3\ddot{V}\right) - sin\,\phi.(3\dddot{V} - \dot{V})$

3.05 $\dfrac{d^5x}{d\phi^5} = cos\,\phi.\left(\dfrac{d^5V}{d\phi^5} - 6\dddot{V} + \dot{V}\right) - 4\,sin\,\phi.\left(\dfrac{d^4V}{d\phi^4} - \ddot{V}\right)$

3.06 $\dfrac{d^6x}{d\phi^6} = cos\,\phi.\left(\dfrac{d^6V}{d\phi^6} - 10\dfrac{d^4V}{d\phi^4} + 5\ddot{V}\right) - sin\,\phi.\left(5\dfrac{d^5V}{d\phi^5} - 10\dddot{V} + \dot{V}\right)$

3.07 $\dfrac{d^7x}{d\phi^7} = cos\,\phi.\left(\dfrac{d^7V}{d\phi^7} - 15\dfrac{d^5V}{d\phi^5} + 15\dddot{V} \quad \dot{V}\right) - 2\,sin\,\phi.\left(3\dfrac{d^6V}{d\phi^6} - 10\dfrac{d^4V}{d\phi^4} + 3\ddot{V}\right)$

3.08 $\dfrac{d^8x}{d\phi^8} = cos\,\phi.\left(\dfrac{d^8V}{d\phi^8} - 21\dfrac{d^6V}{d\phi^6} + 35\dfrac{d^4V}{d\phi^4} - 7\ddot{V}\right) - sin\,\phi.\left(7\dfrac{d^7V}{d\phi^7} - 35\dfrac{d^5V}{d\phi^5} + 21\dddot{V} - \dot{V}\right)$

and so on for higher derivatives of 'x'

3.11 $\dfrac{dy}{d\phi} = \rho.sin\,\phi = \dot{V}.sin\,\phi$

3.12 $\dfrac{d^2y}{d\phi^2} = \dot{V}.cos\,\phi + \ddot{V}.sin\,\phi$

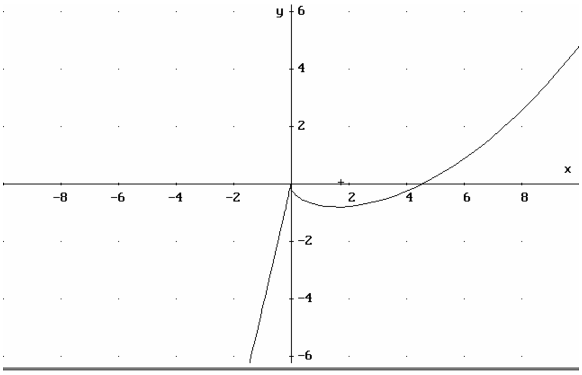
3.13 $\dfrac{d^3y}{d\phi^3} = 2\ddot{V}.cos\,\phi + (\dddot{V} - \dot{V}).sin\,\phi$

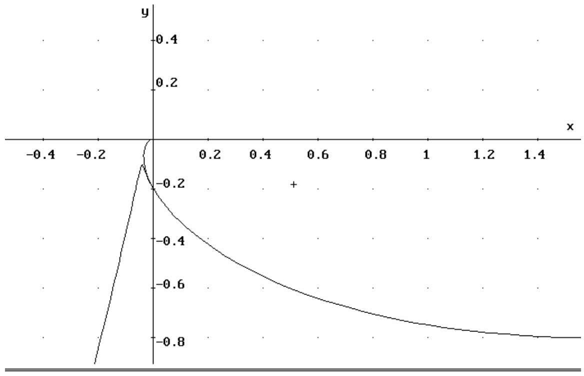
3.14 $\dfrac{d^4y}{d\phi^4} = cos\,\phi.(3\dddot{V} - \dot{V}) + sin\,\phi.\left(\dfrac{d^4V}{d\phi^4} - 3\ddot{V}\right)$

and so on for higher derivatives of 'y'.

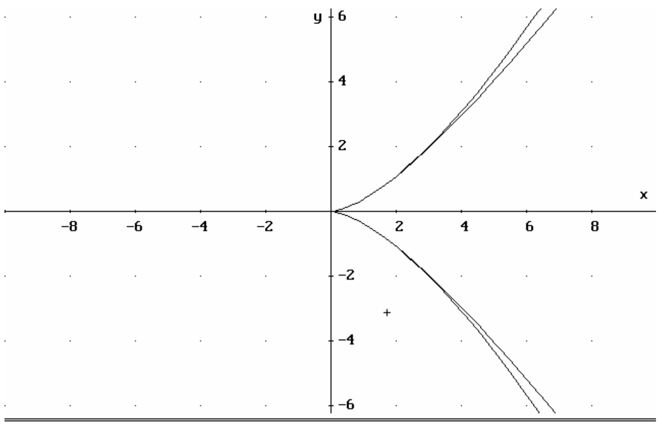
Hence, the general form, where ϕ may be chosen to be zero, but equally, we can substitute any convenient value.

(The five other cases, Cesaro-1, Cesaro-2, Whewell-2, Euler-1 and Euler-2 have been simplified by putting f = 0.)



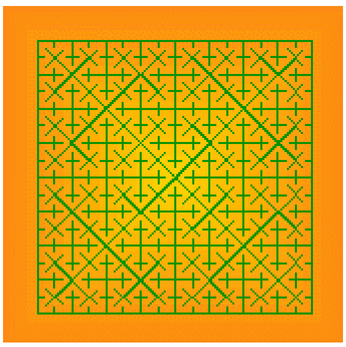


This glove-osculant resembles most of the arc of the Logarithmic Spiral, save the windings about the Pole.



This glove-osculant is a good likeness to the Tractrix on both sides of the excellent cusp.

Ernesto Cesàro (March 12, 1859 – September 12, 1906) was an Italian mathematician who worked in the field of differential geometry. This is his most important contribution which he described in *Lezione di geometria intrinseca* (Naples, 1890). This work contains descriptions of curves which today are named after Cesàro. (Wikipedia)



Free download Cesaro's book (in Italian) from

https://ia600407.us.archive.org/0/items/lezionidigeomet00cesgoog/

DRAWING in the PLANE using *DERIVE*

Agueda Mata & Carmen Torres, Madrid, Spain

INTRODUCTION

This work serves as an excellent practice for a one-semester introductory course in computer graphics for computer science students. This practice is being realized in the course "Lab of Mathematics" in the first year of Computer Science Engineering of the Polytechnic University of Madrid.

The goal of this course is the learning and handling of programs of numerical calculus, symbolic and graphic, such as *DERIVE, MAPLE V,* etc. This will help developing logical reasoning through the resolution of problems with a mathematical model.

It is the purpose of this practice to solve problems related to affine transformations applied to different objects (points, lines, conics, and plane curves in general). In this way we have the intention of improving the students´ geometric vision through *DERIVE* or *MAPLE V*. To do so, we introduce the following basic concepts.

- Implicit and parametric equations of a line and of conics with axes parallel to the coordinate axes (since there are no terms in $x\,y$):

Line: $y = a\,x + b$ $\qquad \left\{\begin{array}{l} x = t \\ y = a\,t + b \end{array}\right\}$

Circle: $(x - c_1)^2 + (y - c_2)^2 = r^2$ $\qquad \left\{\begin{array}{l} x = c_1 + r\cos t \\ y = c_2 + r\sin t \end{array}\right\}$

Parabola: $y = 2p\,(x - c)^2 + a$ $\qquad \left\{\begin{array}{l} x = t + c \\ y = 2p\,t^2 + a \end{array}\right\}$

Ellipse: $\dfrac{(x - c_1)^2}{a^2} + \dfrac{(y - c_2)^2}{b^2} = 1$ $\qquad \left\{\begin{array}{l} x = c_1 + a\cos t \\ y = c_2 + b\sin t \end{array}\right\}$

Hyperbola $\dfrac{(x - c_1)^2}{a^2} - \dfrac{(y - c_2)^2}{b^2} = 1$ $\qquad \left\{\begin{array}{l} x = c_1 + a\cosh t \\ y = c_2 + b\sinh t \end{array}\right\}$ and $\left\{\begin{array}{l} x = c_1 - a\cosh t \\ y = c_2 + b\sinh t \end{array}\right\}$

- Equation of a rotation in R^2 by angle α radians about the centre (p_1, p_2):

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \text{ with } \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \text{ so that } \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} + \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

- Equation of a symmetry regarding L, being L a straight line containing point $P(p_1, p_2)$ and denoting α the angle in radians between the positive x-axis and L.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos 2\alpha & \sin 2\alpha \\ \sin 2\alpha & -\cos 2\alpha \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \text{ with } \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \text{ so that } \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} \cos 2\alpha & \sin 2\alpha \\ \sin 2\alpha & -\cos 2\alpha \end{pmatrix}\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} + \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

- Equation of a dilatation with ratio k.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

- Equation of a translation by the vector (v_1, v_2).

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

In this paper we want to draw our attention to that part of the lab realized with the program *DERIVE for WINDOWS*.

DEVELOPMENT OF THE PRACTICE

Exercise 1: Equation of a line:

a) Draw point P(2,0) and the segment with endpoints {(1,3), (4,4)}, being v the represented vector, draw the line which passes P in direction of v.

b) Geometric meaning of the slope of a line:

Write a vector of dimension 8 with the following functions as its components: y = –5x, y = –3x, y = 2x, y = 5x, y = 8x, y = – 1/2 x, y = 0, x = 0. Represent this vector graphically. Can you recognize each of the lines? Which geometric transformation allows us to obtain each of the lines from any other one?

c) Geometric meaning of the independent term of a line:

Repeat section c) for the following lines: y = 2x + 2, y = 2x + 3, y = 2x + 8, y = 2x, y = 2x – 2, y = 2x – 6.

d) Look for eight functions so that their representation accurately reproduce the following figure (scale 2 : 2)

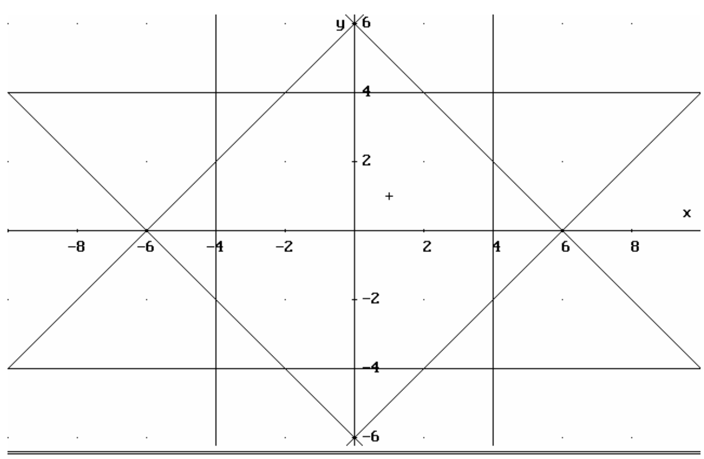


Fig. 1

Solutions:

a) Use Option Points Connect in order to plot: $\left[\begin{bmatrix}2 & 0\end{bmatrix}\begin{bmatrix}1 & 3\\ 4 & 4\end{bmatrix}\right], [2+3t, t]$.

b) `[-5x, -3x, 2x, 5x, 8x, -(1/2)x, 0, x = 0]`

c) `[2x + 2, 2x + 3, 2x + 8, 2x - 2, 2x - 6]`

d) `[x = 4, x = -4, 4, -4, x + 6, x û 6, -x + 6, -x - 6]`

Exercise 2:

Plot the following curves using the parameter form:

Circle: $(x-2)^2 + (y-1)^2 = 4$ Parabola: $y = 2x^2 - 8x + 5$

Ellipse: $\frac{(x-2)^2}{4} + \frac{(y-1)^2}{16} = 1$ Hyperbola: $\frac{(x-2)^2}{4} - \frac{(y-1)^2}{16} = 1$

Is this parabola inscribed into the circle or is the circle inscribed into the parabola?

Solution:

```
[[2+2*COS(t),1+2*SIN(t)],[t+2,2*t^2-3],[2+2*COS(t),1+4*SIN(t)],
[2+2*COSH(t),1+4*SINH(t)],[2-2*COSH(t),1+4*SINH(t)]]
```
(scale 2:2)

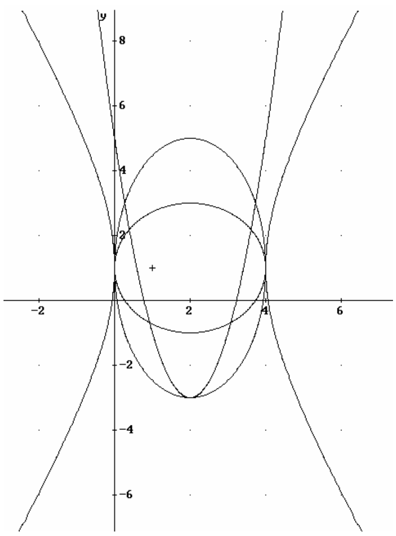


Fig. 2

Exercise 3:

a) Build an affine transformation `AFFINE(r,a,b,c,d,e,f)` which applies the transformation

whose equations are: $\begin{pmatrix}a & b\\ c & d\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix} + \begin{pmatrix}e\\ f\end{pmatrix}$, to the object r given as a matrix r := $\begin{pmatrix}x_1 & y_1\\ \dots & \dots\\ x_n & y_n\end{pmatrix}$.

b) Plot the following triangles using Option Points Connect:

$$t1:=\begin{pmatrix}-1 & 2\\ 0 & -1\\ -2 & 2\\ -1 & 2\end{pmatrix}\qquad t2:=\begin{pmatrix}0 & 0\\ 1 & 0\\ 2 & 1\\ 0 & 0\end{pmatrix}\qquad t3:=\begin{pmatrix}-1 & -3\\ -3 & -2\\ -2 & 0\\ -1 & -3\end{pmatrix}\qquad t4:=\begin{pmatrix}1 & -1\\ 2 & -2\\ 2 & -1\\ 1 & -1\end{pmatrix}$$

Using the mapping AFFINE build and plot the images of the previous triangles applying

$\begin{pmatrix}a & b\\ c & d\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix}+\begin{pmatrix}0\\ 0\end{pmatrix}$ for the following matrices: $\begin{pmatrix}-1 & 0\\ 0 & -1\end{pmatrix}$, $\begin{pmatrix}\sqrt{2}/2 & \sqrt{2}/2\\ -\sqrt{2}/2 & \sqrt{2}/2\end{pmatrix}$, $\begin{pmatrix}1/4 & 0\\ 0 & -1/4\end{pmatrix}$, $\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$.

Describe the figure´s transformation!

c) Build the affine transformation R1(r), which applies a 60° rotation about the centre (0,-2) to object r.

d) Build the affine transformation R2(r), which applies a - 60° rotation about the centre (0,-2) to obj. r.

e) Build the affine transformation S1(r) which applies a symmetry with respect to the axis x = 0 to r.

f) Plot the object `ship1`. Apply a dilatation with ratio 2/3 and a translation of vector (0,1) to `ship1`, name the new object `ship2` and plot `ship2`.

$$\text{ship1}:=\begin{pmatrix}-2 & -1 & 1 & 2 & -2 & 0 & 0 & 2 & 0\\ 1 & 0 & 0 & 1 & 1 & 1 & 3 & 5/4 & 5/4\end{pmatrix}^{T}$$

g) Plot `[ship2,R1(ship2),R2(ship2),S1(R1(ship2))].`

Solution:

a)
```
AFFINE(r,a,b,c,d,e,f):=([[a,b],[c,d]]*r`+VECTOR(VECTOR(
IF(i=1,e,f),j,1,DIMENSION(r)),i,1,2))`
```

b)
```
MOV1(r):=AFFINE(r,-1,0,0,1,0,0)
[MOV1(t1),MOV1(t2),MOV1(t3),MOV1(t4)]
```

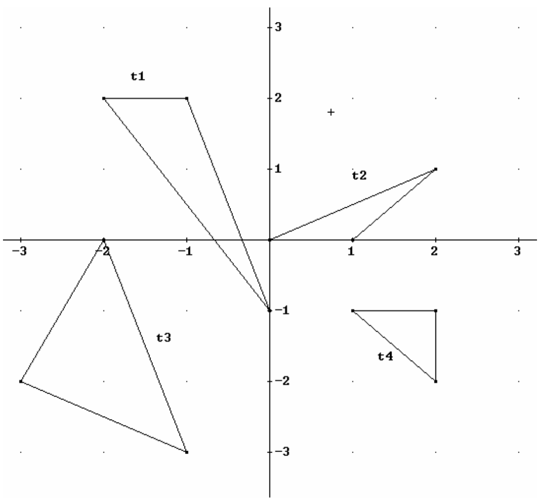


Fig. 3

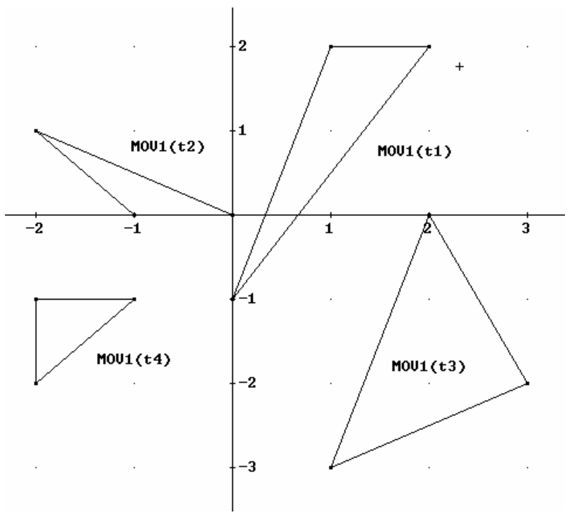


Fig. 4

```
MOV2(r):=AFFINE(r,SQRT(2)/2,SQRT(2)/2,-SQRT(2)/2,SQRT(2)/2,0,0)
[MOV2(t1),MOV2(t2),MOV2(t3),MOV2(t4)]

MOV3(r):=AFFINE(r,1/4,0,0,-1/4,0,0)
[MOV3(t1),MOV3(t2),MOV3(t3),MOV3(t4)]

MOV4(r):=AFFINE(r,0,1,1,0,0,0)
[MOV4(t1),MOV4(t2),MOV4(t3),MOV4(t4)]
```

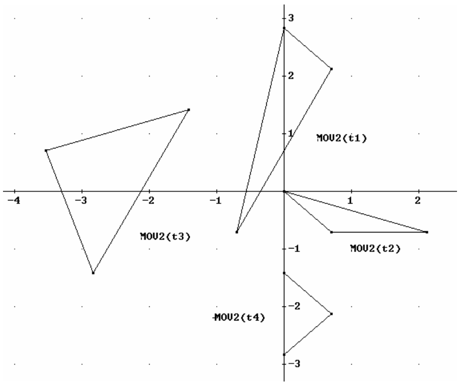


Fig.5

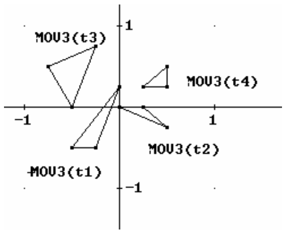


Fig. 6

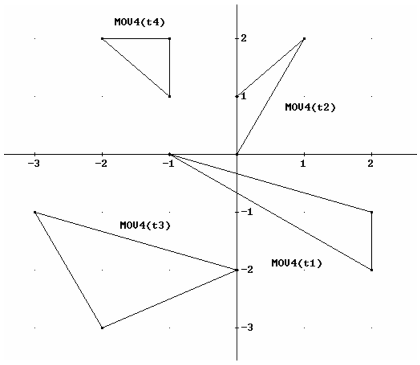


Fig. 7

c) `R1(r):=AFFINE(r,1/2,-SQRT(3)/2,SQRT(3)/2,1/2,-SQRT(3),-1)`

d) `R2(r):=AFFINE(r,1/2,SQRT(3)/2,-SQRT(3)/2,1/2,SQRT(3),-1)`

e) `S1(r):=AFFINE(r,-1,0,0,1,0,0)`

f) `ship2:=AFFINE(ship1,2/3,0,0,2/3,0,1)`

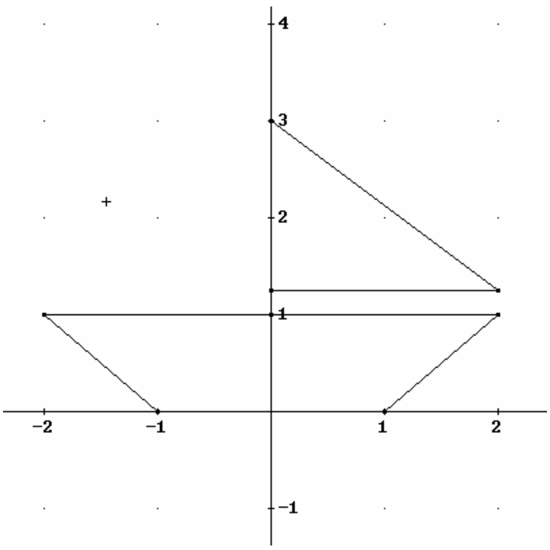


Fig. 8

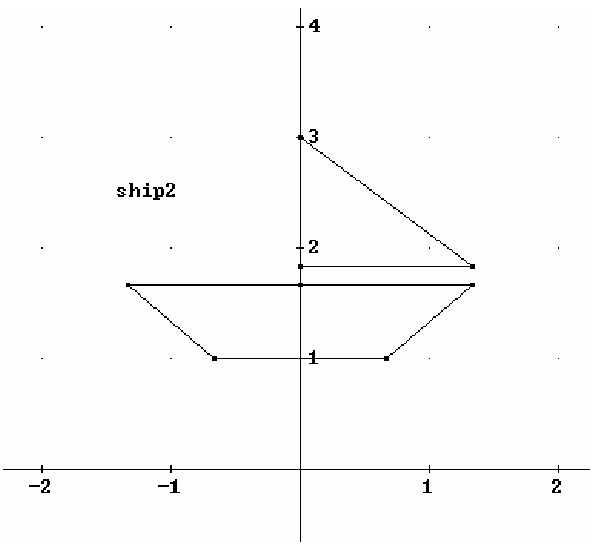


Fig. 9

g) `[ship2,R1(ship2),R2(ship2),S1(R1(ship2))]`

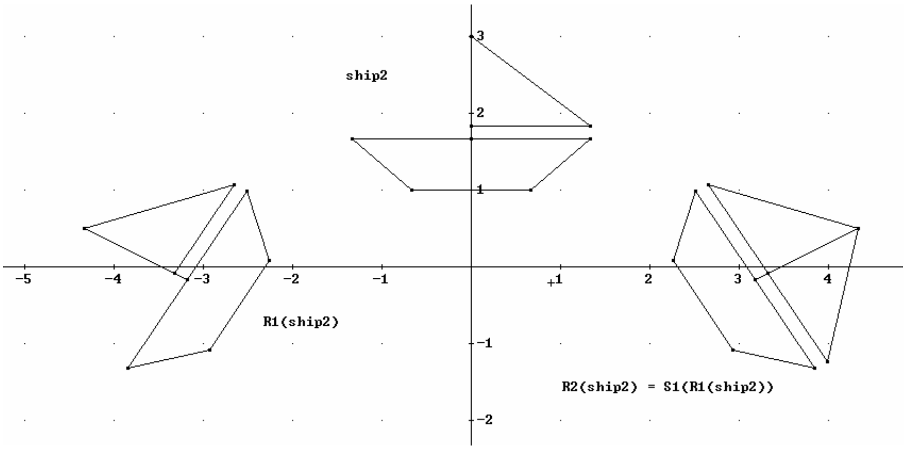


Fig. 10

Exercise 4:

a) Plot the semicircle `scir` with centre (–4,0) and radius 1, placed in the halfplane y ≥ 0.

b) Build a function G(r) which applies on that semicircle a symmetry with respect to the line y = 1 and a translation of the vector (2,0) using the function AFFINE from exercise 3. (We will call a symmetry composed of a translation of parallel vector to the line of symmetry a "sliding symmetry").

c) Plot `[scir,G(scir),G(G(scir)),G(G(G(scir))),G(G(G(G(scir))))]`

Solution:

a) `scir:=[[SIN(t)-4,COS(t)]]`

b) `G(r):=AFFINE(r,1,0,0,-1,2,2)`

c)

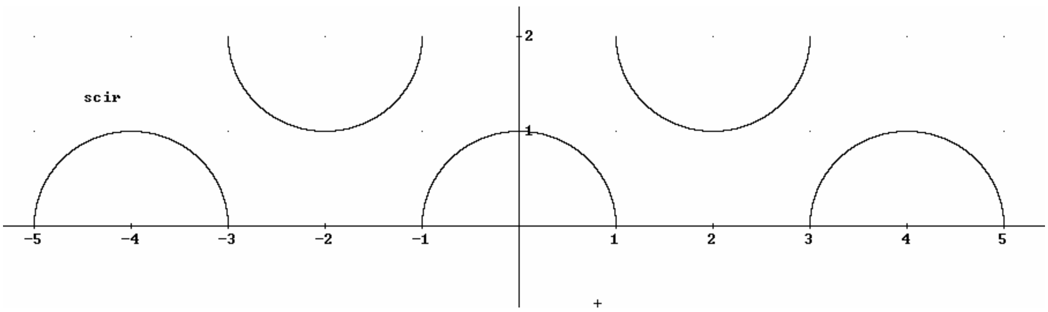


Fig. 11

Exercise 5:

a) Build a function F(r) that is a sliding symmetry with respect to the line y = 1/2 using the translation vector (1,0).

b) Plot `[scir,F(scir),F(F(scir)),F(F(F(scir))),F(F(F(F(scir)))),`
`F(F(F(F(F(scir))))),F(F(F(F(F(F(scir)))))),F(F(F(F(F(F(F(scir))))))),`
`F(F(F(F(F(F(F(F(scir))))))))]`

c) In the interval [– π/2, π/2] plot the following functions:

```
[[-COS(t),SIN(t)],[t+π/2,-1],[t+π/2,1/2-COS(2*t)/2],[COS(t)/2+π+1/2,
SIN(t)/4+3/4],[COS(t)/3+π+1/2,SIN(t)/6+1/4],[COS(t)/3+π+1/2,SIN(t)/
6-1/8],[COS(t)/3+π+1/2,SIN(t)/6-1/2],[COS(t)/4+π+1/2,SIN(t)/8-7/8]]
```

d) Apply a dilatation with ratio 1/8 and a translation with the vector (–4,0) on the previous objects. How would you call the resulting object?

e) Plot [?, F(?), F(F(?)), F(F(F(?))), F(F(F(F(?)))), F(F(F(F(F(?))))), F(F(F(F(F(F(?))))))] with ? being the resulting object of section d).

Solution:

a) `F(r):=AFFINE(r,1,0,0,-1,1,1)`

b)

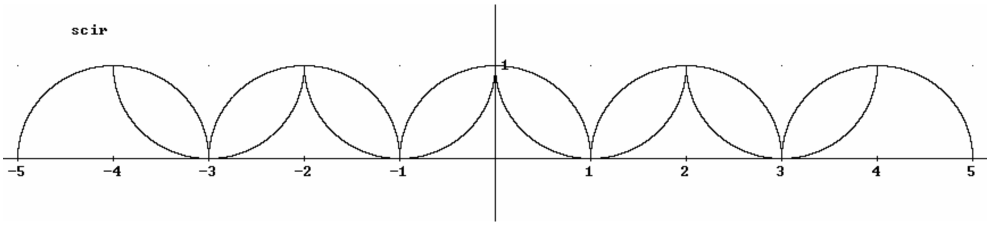


Fig. 12

c)

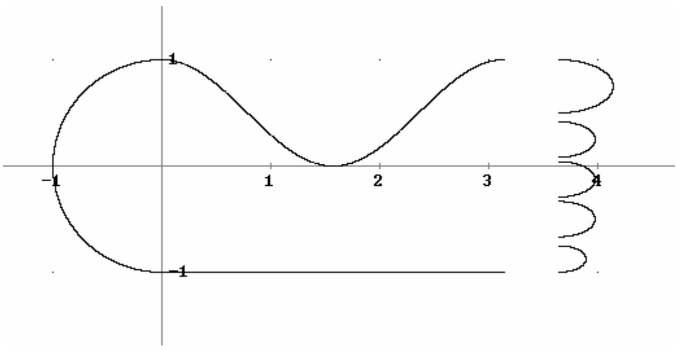


Fig. 13

d)
```
foot:=AFFINE([[-COS(t),SIN(t)],[t+pi/2,-1],[t+pi/2,1/2-COS(2*t)/2],
[COS(t)/2+pi+1/2,SIN(t)/4+3/4],[COS(t)/3+pi+1/2,SIN(t)/6+1/4],
[COS(t)/3+pi+1/2,SIN(t)/6-1/8],[COS(t)/3+pi+1/2,SIN(t)/6-1/2],
[COS(t)/4+pi+1/2,SIN(t)/8-7/8]],1/8,0,0,1/8,-4,0)
```

e)

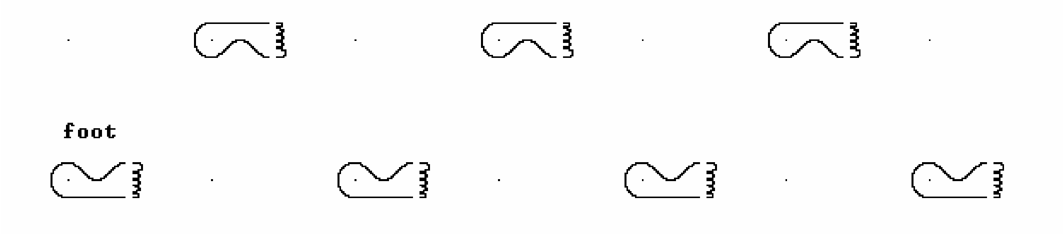


Fig. 14

Exercise 6:

a) Plot the following objects:
`parabola:=[[2*t,t^2]],`
`spiral:=[[#e^(t/4)*SIN(2*t),#e^(t/4)*COS(2*t)]],`
`trisectrix:=[[SIN(2*t)+SIN(t),-COS(2*t)-COS(t)]],`
`infinity:=[[3*SIN(t),2*SIN(2*t)]]` with $t \in [-\pi, \pi]$ and
`slinky:=[[t-11*SIN(3*t)/10,11*COS(3*t)/5]]` with $t \in [-2\pi, 2\pi]$.

b) Do this section with the graphic screen clear:

- Apply to `slinky` a symmetry with respect to y = 0, a dilatation with ratio 1/3 and a translation with a vector (0,4). Call the outcome `hair`.
- Apply to `spiral` a dilatation with ratio 1/8 and a translation with a vector (–1,3). Call the result `eyel`.
- Apply a symmetry with respect to x = 0 to `eyel` and call the result `eyer.`
- Apply a dilatation with ratio 1/8 and a translation by (0,2) to `trisectrix` giving `nose`.
- Plot `[hair, eyel, eyer, nose]` in the interval $[-\pi, \pi]$.

c) Do this section again with the graphic screen clear:

- Apply a dilatation with ratio 1/3 on `parabola`, call the outcome `face`. Plot face in the interval $[-\pi, \pi]$.
- Apply a dilatation with ratio 1/3 and a translation by (0, –1/2) on `infinity`, giving `bowtie`. Plot `bowtie`, `eyel` and `eyer` in the interval $[-\pi, \pi]$.
- Apply a symmetry wrt to x = 0 to `nose`, giving `nose2`. Plot `nose` and `nose2` for $[-\pi, 2\pi/3]$.
- Plot `hair` in the interval $[-2\pi, 2\pi]$.
- Plot a mouth in the face which you have just drawn. This will show your mood after this lab.

Solution:

a)

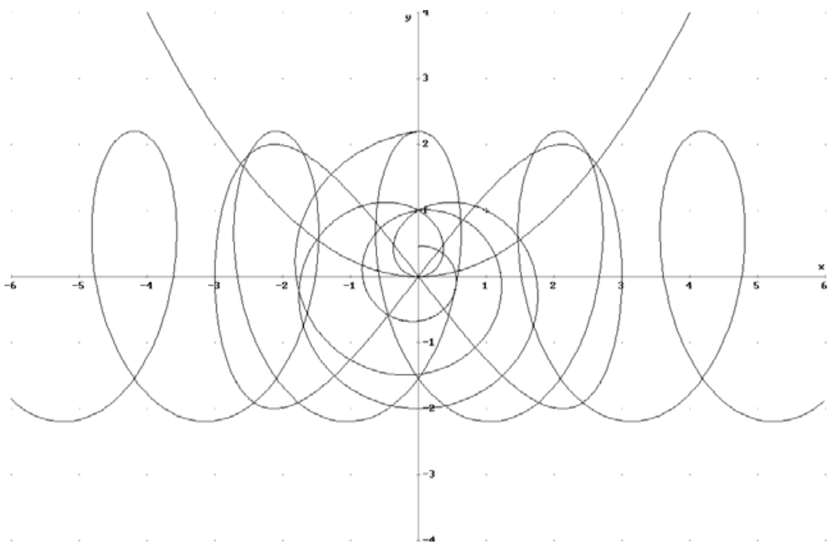


Fig. 15

b)
```
hair:=AFFINE(AFFINE(slinky,1,0,0,-1,0,0),1/3,0,0,1/3,0,4)
eyel:=AFFINE(spiral,1/8,0,0,1/8,-1,3)
eyer:=AFFINE(eyel,-1,0,0,1,0,0)
nose:=AFFINE(trisectrix,2/3,0,0,2/3,0,2)
```

c)
```
face:=AFFINE(parabola,1/3,0,0,1/3,0,0)
bowtie:=AFFINE(infinity,1/3,0,0,1/3,0,-1/2)
nose2:=AFFINE(nose,-1,0,0,1,0,0)
hyperbola:=[[SQRT(3)*SINH(t),3*COSH(t)]]
mouth:=AFFINE(hyperbola,2/9*COS(pi/13),2/9*SIN(pi/13),
       -2/9*SIN(pi/13),2/9*COS(pi/13),-1/3,1/3)
```

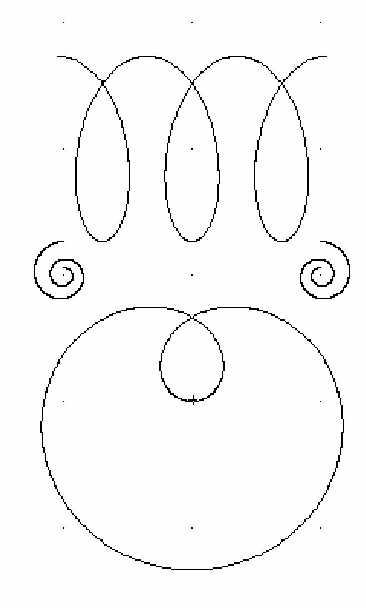


Fig 16

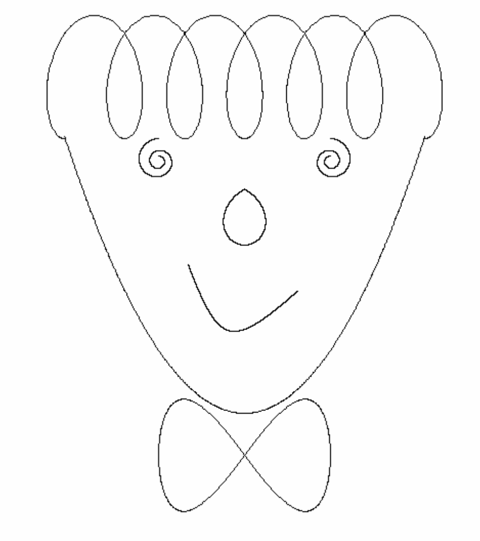


Fig. 17

BIBLIOGRAPHY

A Catalog of special Plane Curves, J.Dennis Lawrence, Dover
Metric Affine Geometry, Ernst Snapper and Robert J. Troyer. Dover
Álgebra y Geometriá, Eugenio Hernández, Addison Wesley/Universidad Autónoma de Madrid
Lecciones de Àlgebra y Geometriá, C.Asina - E.Trillas, GG.

email address of the contributors:

amata@fi.upm.es
ctorres@fi.upm.es
http://www.fi.upm.es/~dma

Agueda Mata & Carmen Torres
Dpto Matemática Aplicada
Facultad de Informática. U.P.M.
Campus de Montegancedo. s/n
28660 Boadilla del Monte, Madrid, Spain

Dear Carmen and Agueda,
*In a very recommendable book I found lots of designs and patterns and it could be for your students a challenge to find **one** function to produce the pattern given on the next page. That is you should use only one parameter running from 0 through 1. It is a nice exercise to practise parametric representations of curves, Josef. See next page.*

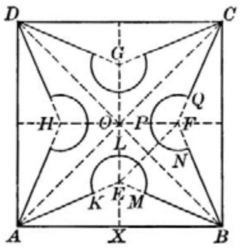


Fig. 114.

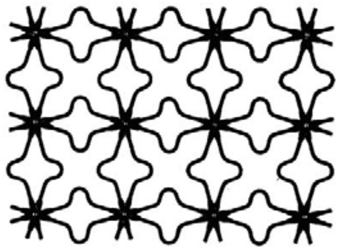


Fig. 114*a*. — Counter Railing Design.

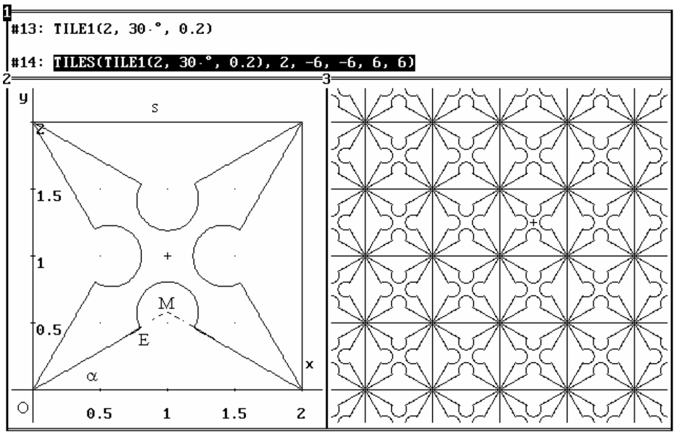
139. **In Fig. 114 ABCD is a square with diameters EG and FH and diagonals AC and BD. KE = EM = FN = FQ, etc., and ∠EAB = ∠EBA = ∠FBC = ∠FCB = ∠GCD, etc.**

This the *DERIVE*-function:

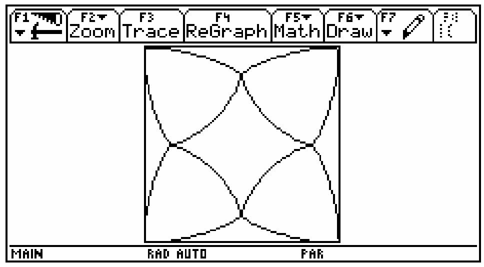
`TILE1(AB, ∠EAX, KE/AE)`

Left below is the *DOS-DERIVE* plot.

Now you can introduce sliders for the angle and the prop factor.
I did this with TI-NspireCAS (see below).

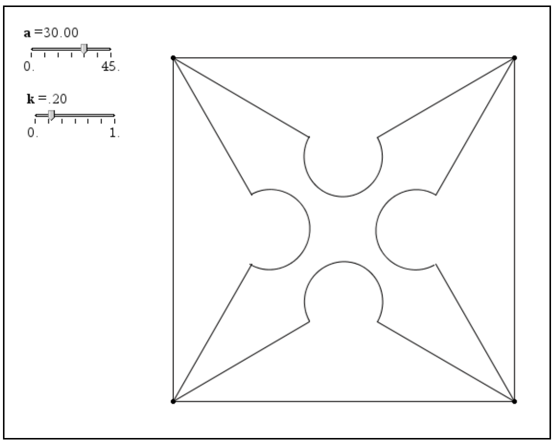


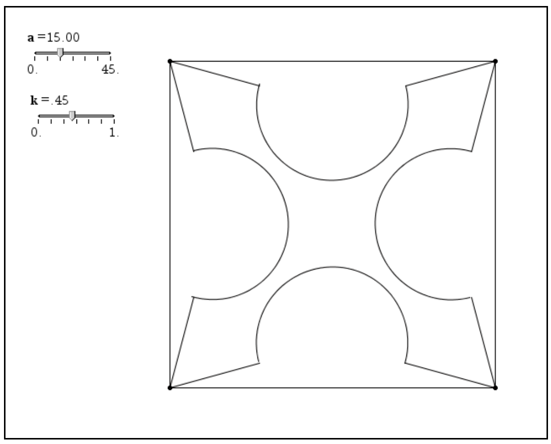
See another design presented on the *TI*-92.



(Source [2])

Variable design with sliders and NspireCAS





There are some exercises given in Sykes' pretty book, eg.

If ∠*EAB* = 30°, *KE* = 1/6 *AE*, and *AB* = *a* find the areas of (1) *AKLMBA*; (2) *AKLMBNPQC*, etc.

and next:

Find the areas mentioned above under the same conditions, except that *KE* = *AE*/n.

Answers: (1) $\dfrac{a^2}{36n^2}\left(3n^2\sqrt{3}+8\pi\right)$; (2) $\dfrac{a}{9n^2}\left(9n^2-3n^2\sqrt{3}-8\pi\right)$

and many others …

[1] *Source Book of Problems for Geometry*, Mabel Sykes, Dale Seymour
[2] *Gotische Maßwerkfenster im Geometrieunterricht*, Günter Schmidt, MUm Jg 41, Heft 3 mai 1995

6174 IS A SPECIAL NUMBER.

Sebastiano Cappuccio, Forli, Italy

In “Il Fibonacci, breve viaggio fra curiosità matematiche” (The Fibonacci, a brief journey among mathematical curiosities), by F. Conti, R. Dvornicich, T. Franzoni and S. Mortola, a poster published in 1990 by the “Scuola Normale Superiore di Pisa” for the sixth “Gara Nazionale di Matematica”, this interesting item appears:

6174 IS A SPECIAL NUMBER.

If we put its digits in decreasing order and in increasing order, we get the greatest and the lowest integer numbers made with these digits: 7641 and 1467.
Their difference is

7641 – 1467 = 6174,

the original number itself.
All this appears as a simple coincidence; if we execute the same operation with another four digit number, for example 5652, we obtain:

6552 – 2556 = 3996,

which has not anything special. But if we repeat several times the same operation, we obtain:

9963 – 3699 = 6264,
6642 – 2466 = 4176,
7641 – 1467 = 6174

and so we get the number 6174 again.
This happens with every integer number of 4 digits (not all equal, of course)!

Some questions arise: is it really true? Is 6174 the only number with this strange property, or is there some other integer number with 2, 3, 5... digits with the same property? Why?

We can use an “experimental approach” with DERIVE; we need a sorting algorithm and DNL # 13 offers us a good tool ready for that: the file SORT.MTH. Preload SORT.MTH or any other sort function.

Now we need a function which puts all digits of a given integer number n as elements of a vector:

```
n:=
DIGITS(N):=VECTOR(MOD(FLOOR(N,10^K),10),K,3,0,-1)
```

We have to sort this vector in descending and in ascending order:

```
a:=SORT(DIGITS(n))
b:=REVERSE_VECTOR(a)
```

Then, to get the difference between these numbers, we have to transform the vectors in integer numbers again:

```
NUM(v):=SUM(ELEMENT(v,k)*10^(DIMENSION(v)-k),k,1,DIMENSION(v))
```

The following function creates a new vector: its first element is the argument of the function (a four digit integer number), then it repeats the difference between the numbers sorted in descending and in ascending order until the result is the same. We saw that this leads us to a firm configuration: 6174.

```
TRY(x):=ITERATES(NUM(b)-NUM(a),n,x)
```

Now we are ready to try the algorithm:

```
TRY(4321) = [4321, 3087, 8352, 6174, 6174]
TRY(4000) = [4000, 3996, 6264, 4176, 6174, 6174]
```

Of course it does not work when the digits are all equal:

```
TRY(7777) = [7777, 0, 0]
```

This "program" runs with three-digit integer numbers too, but it works as if it has the fourth digits equals to 0:

```
TRY(300) = [300, 2997, 7173, 6354, 3087, 8352, 6174, 6174]
```

Now we can explore if there is a three-digit number which works as 6174:

```
DIGITS(n):=VECTOR(MOD(FLOOR(n,10^k),10),k,2,0,-1)
```

This number exists, it is 495:

```
TRY(123) = [123, 198, 792, 693, 594, 495, 495]
TRY(990) = [990, 891, 792, 693, 594, 495, 495]
```

It seems that there are not integer numbers with the same property and two or five ... digits:

```
DIGITS(n) := VECTOR(MOD(FLOOR(n, 10^k ), 10), k, 4, 0, -1)
TRY(12345) = [12345, 41976, 82962, 75933, 63954, 61974, 82962]
TRY(76514) = [76514, 61974, 82962, 75933, 63954, 61974]
```

Some questions to the readers: it is not very difficult to demonstrate why 495 is an "attractor" with this algorithm for three-digit integer numbers. What is the greatest number of iterations to reach 495? And what for 6174?

Why with 2, 5, 6... digits it seems that we have not a "special number", but only a "cycling set" of numbers? I was not able to get that.

Comment in 2013:
The DERIVE 6 Version uses the DERIVE-function NAME_TO_CODES:
(Code of 1 is 49, code of two is 50, etc.)

#1: a(n) := SORT(NAME_TO_CODES(n))

#2: $\text{num}(v) := \sum_{k=1}^{\text{DIM}(v)} (v_k - 48)\cdot 10^{\text{DIM}(v) - k}$

#3: try(x) := ITERATES(num(REVERSE(a(n))) - num(a(n)), n, x)

#4: a(2345) = [50, 51, 52, 53]

#5: num(a(2345)) = 2345

#6: try(2354) = [2354, 3087, 8352, 6174, 6174]

#7: try(509) = [509, 891, 792, 693, 594, 495, 495]

#8: try(12345) = [12345, 41976, 82962, 75933, 63954, 61974, 82962]

Rational Points on the Unit Circle with *DERIVE* and the *TI*-92

Peter Antonitsch, Ferlach, Austria

Andrew Wiles' proof of Fermat's last theorem has made rational points on elliptic curves a prominent topic in nowadays mathematical publications. Although being no elliptic curve itself, the unit circle offers an easy approach to some basic concepts of rational points and – because of being a bit more simple – some nice formulae to be used with CAS.

Some theory to start with[1]

As a special case of the Hilbert-Hurwitz Theorem one can get all rational points on the unit circle by finding one rational point *P* and sweeping all the secant lines through *P* of rational slope. By choosing *P* = (-1 | 0), each secant of rational slope $\frac{n}{m}$ intersects the unit circle in $\rho\left(\frac{n}{m}\right)=\left(\frac{m^2-n^2}{m^2+n^2}\;\middle|\;\frac{2mn}{m^2+n^2}\right)$, which is the well known rational parametrization of the unit circle (and contains Euklid's formulae for Pythagorean triples). This parametrization lacks in not reaching the rational point (-1 | 0) (although any other rational point (u | v) can be obtained choosing $n=v,\ m=1+u$).

Making use of Gaussian integers a »simple generalization« of this parametrization helps to overcome this defect: Defining $f(m+ni)\overset{def}{=}\left(\frac{m^2-n^2}{m^2+n^2}\;\middle|\;\frac{2mn}{m^2+n^2}\right);\ m,n\in\mathbf{Z}$ leads to $(-1\mid 0)=f(ni)$. Thus *f* is an onto map from **Z**[i]\{0} to the set $C(\mathbf{Q})$ of rational points on the unit circle, which can be made a group defining an addition by $(x_1\mid y_1)+(x_2\mid y_2)\overset{def}{=}(x_1x_2-y_1y_2\mid x_1y_2+x_2y_1)$. Therefore *f* preserves the structure of the semigroup **Z**[i]\{0} (with complex multiplication) and – which is of special interest – maps the »generating elements« of **Z**[i]\{0} onto those of $C(\mathbf{Q})$.

These »generating elements« are the prime elements of **Z**[i]\{0}:

- $1+i$ and $1-i$ are associated prime elements, where $f(1\pm i)=(0\mid \pm 1)$ (both elements in $C(\mathbf{Q})$ of order 4);
- prime numbers $p\equiv 3(4)$ are prime elements with $f(p)=(1\mid 0)$, the unit in $C(\mathbf{Q})$;
- Gaussian integers $m+ni\;\middle|\;m^2+n^2=p\equiv 1(4)$, *p* a prime number, are prime elements (of norm *p*) with $f(m+ni)=\left(\frac{m^2-n^2}{m^2+n^2}\;\middle|\;\frac{2mn}{m^2+n^2}\right)$.

Thus $\left\{\left(\frac{m^2-n^2}{m^2+n^2}\;\middle|\;\frac{2mn}{m^2+n^2}\right)\right\}_{m^2+n^2=p\equiv 1(4)}\cup\{(0\mid 1)\}$ suffices to generate all rational points on the unit circle (which can be obtained as direct sums of these »generating elements«).

Algorithms[2]

From the above it is clear, that an algorithm for the »composition« of rational points on the unit circle« has to contain a list of or a function to obtain all the generating elements that are needed and some funtions that provide the basic operations on $C(\boldsymbol{Q})$ (addition, calculation of the inverse or a multiple).
As there are infinitely many prime numbers congruent 1 mod 4 it is reasonable to choose the dynamic approach, calculating all the generating elements of $C(\boldsymbol{Q})$ from the prime elements in **Z**[i]. It helps, that every relevant prime element $m+ni$ can be identified with a prime number (2 or a prime number $\equiv 1(4)$, choose m>n to obtain uniqueness), so that a list of all the needed prime numbers (also containing their multiplicity) will do to get any rational point:

$C(\boldsymbol{Q})$-COMPOSITION:

Input: List of needed prime numbers (2 and/or prime numbers $\equiv 1(4)$), containing their multiplicities
Step 1a: Identifying 2 with $(0 \mid 1) \in C(\boldsymbol{Q})$ **Step 1b:** Splitting every prime number $\equiv 1(4)$ into a sum of sqares, $p = m^2 + n^2$, to calculate $\left(\frac{m^2-n^2}{m^2+n^2} \,\middle|\, \frac{2mn}{m^2+n^2}\right) \in C(\boldsymbol{Q})$
Step 2: Calculating the needed multiples of the generating elements
Step 3: Adding the results of step 2
Output: The rational point on the unit circle, determined by the input-list

To derive an algorithm for the »decomposition« of any rational points on the unit circle we need some results, the proofs of which are rather technical and are therefore omitted:

- Let $G_p = \left(\frac{a}{p} \,\middle|\, \frac{b}{p}\right)$ (for $p \equiv 1(4)$) be the generating element in $C(\boldsymbol{Q})$, then the denumerators of the n-times multiple $n \cdot G_p$ equal $p^{|n|}$;
- therefore, the (reduced) denumerators of an expression like $\bigoplus_j n_j \cdot G_{p_j}$ equal $\prod_j p_j^{|n_j|}$;
- multiples of $f(1+i) = G_2$ change these denumerators by factor 1.
- Every rational point on the unit circle can be written as $R_C = n_2 \cdot G_2 \oplus \left(\bigoplus_j (\pm n_j) \cdot G_{p_j}\right)$.
- All that can be combined in the »sign-rule«:

If the reduced denumerators of $R_C \oplus (-n_{j_0}) \cdot G_{p_{j_0}}$ equal $\prod_{j \neq j_0} p^{|n_j|}$, then p_{j_0} has multiplicity $+n_{j_0}$, otherwise $-n_{j_0}$.

- The multiplicity of 2 can be derived from the result of $R_C \oplus \left(\bigoplus_{j, p_j \neq 2} (\mp n_j) \cdot G_{p_j}\right)$, which has to be equal $n_2 \cdot G_2$ and leads to the rational points $(0 \mid 1)$ ($n_2 = 1$), $(-1 \mid 0)$ ($n_2 = 2$), $(0 \mid -1)$ ($n_2 = 3$) or $(1 \mid 0)$ ($n_2 = 4$) – notice, that $\langle G_2 \rangle$ is cyclic of order 4:

$C(\mathbb{Q})$-DECOMPOSITION:

Input: Rational point R_C on the unit circle that has to be »decomposed«	
Step 1: Splitting the point's reduced denumerator d into primes (considering their multiplicities as well) and selecting all prime numbers $\equiv 1(4)$	
Step 2:	For all prime numbers $p \equiv 1(4)$ Finding *m*, *n* so that $p = m^2 + n^2$ Calculating the generating element G_p as $f(m+ni)$ Calculating d/p^{n_p} and $R_C \oplus (-n_p)\cdot G_p$ to get the correct sign for the prime's multiplicity
Step 3: Calculating the multiplicity of $G_2 = f(1+i)$ from the result of $R_C \oplus \left(\bigoplus_{j,p_j \neq 2} (\mp n_j)\cdot G_{p_j} \right)$	
Output: List of prime numbers (2 and/or prime numbers $\equiv 1(4)$, containing their multiplicities) needed to calculate the elements G_{p_j} generating the »input rational point« when being added up	

These two algorithms are complementary so that the output of the one can be used as input for the other algorithm to check the result.

A closer look on the *DERIVE*- and *TI*-92-programs:

As the *DERIVE*-and *TI*-92 programs are very similar, only the *DERIVE* source-code (including the original program-remarks) is listed below, followed by explanations of certain program-parts and of some differences between the two versions.

"Part 1: Creating a list of prime factors congruent 1 mod 4 as a list of lists [prime factor, multiplicity]"

```
"List of prime factors of a number (multiplicities as exponents)"
(TI-92: function faclist(num))

FACTOR_LIST(num):=IF(num=1,[1],FACTORS(FACTOR(num)))
```

```
"Exponent of a number (to be used with the Simplify-Command)"
(TI-92: function exponent(num))

EXPONENT(num):=IF(num=1,1,(FACTORS(FACTOR(LN(num)))) SUB 2)
"Conversion: FACTOR_LIST(num) into: List of lists [prime factor, multiplicity]"
(TI-92: function primsort(num, typ) with »basis(num)«, »primlist(num)« and »mbubsort(mat,row,typ)«)

PLIST_HELP(flist):=VECTOR(IF(PRIME(FLOOR(flist SUB i)),[flist SUB i,1],
[flist SUB i^(1/EXPONENT(flist SUB i)),EXPONENT(flist SUB i)]),i,1,DIMENSION(flist))

PRIMEFACTOR_LIST(num):=IF(num=0,[[0,1]],IF(num<0,APPEND([[-1,1]],
PLIST_HELP(FACTOR_LIST(ABS(num)))),APPEND([[1,1]],PLIST_HELP(FACTOR_LIST(num)))))
```

```
"Test: Prime congruent 1 mod 4?"
(TI_92: function cong1(num))

CONG_1(num):=IF(MOD(num,4)=1,0,1)
```

```
"Conversion: List of lists [prime factor, multiplicity] into:
List of lists [prime factor congruent 1 mod 4, multiplicity]"
(TI_92: function cqlist(num))

CQ_LIST(num):=SELECT(CONG_1(k SUB 1),k, PRIMEFACTOR_LIST(num))
```

"Part 2: Definition of some operators into or on C(Q)"

```
"Mapping f: Z[i]->C(Q)"
(TI-92: function f(zli))

f(zi_num):=[(zi_num SUB 1^2-zi_num SUB 2^2)/(zi_num SUB 1^2+zi_num SUB 2^2),
2*zi_num SUB 1*zi_num SUB 2/(zi_num SUB 1^2+zi_num SUB 2^2)]
```

```
"Splitting a prime congruent 1 mod 4 into a sum of squares (helpfile for CQ_PRIME)"
(TI_92: function cutprime(pnum))

SPLIT_PRIME(pnum):=IF(pnum>1 AND CONG_1(pnum),REVERSE_VECTOR(SELECT(NOT(k=0),
k,VECTOR(IF(FLOOR(SQRT(pnum-i^2))=SQRT(pnum-i^2),i,0,0),i,1,FLOOR(SQRT(pnum))+1))),[1,0])
```

```
"Prime element of C(Q) corresponding to integer prime number p, p equals 2 or p congruent 1 mod 4"
(TI-92: function cqprime(pnum))

CQ_PRIME(p):=IF(p=2,[0,1],f(SPLIT_PRIME(p)))
```

```
"C(Q)-operators"
(TI-92:   function oncircle(cq), function cqplus(cq1,cq2), function cqinv(cq), function cqmult(m,cq)
          function cqsumh(plist,j), function cqsum(plist))

ON_UNITCIRCLE(cq):=IF([cq SUB 1^2+cq SUB 2^2=1,0,1,1])

CQ_PLUS(cq1,cq2):=[cq1 SUB 1*cq2 SUB 1-cq1 SUB 2*cq2 SUB 2,
cq1 SUB 1*cq2 SUB 2+cq1 SUB 2*cq2 SUB 1]

CQ_INVERSE(cq):=[cq SUB 1,-cq SUB 2]

CQ_MULTIPLE(m,cq):=IF(m=0,[1,0],IF(m>0,ITERATE(CQ_PLUS(cq,hz),hz,[1,0],m),
ITERATE(CQ_PLUS(CQ_INVERSE(cq),hz),hz,[1,0],-m)))

CQ_SUMHELP(plist,j):=IF(j=1,CQ_MULTIPLE(plist SUB 1 SUB 2,CQ_PRIME(plist SUB 1 SUB 1)),
CQ_PLUS(CQ_MULTIPLE(plist SUB j SUB 2,CQ_PRIME(plist SUB j SUB 1)),CQ_SUMHELP(plist,j-1)))

CQ_SUM(plist):=CQ_SUMHELP(plist,DIMENSION(plist))
```

"Part 3: Composition of points in C(Q)"

```
"Composition utility"
(TI-92: function cqcomp(list))

CQ_COMP(list):=CQ_SUM(list)
```

"Part 4: Decomposition of points in C(Q)"

"List of prime factors congruent 1 mod 4 of a C(Q)-point's denumerator (including multiplicities)"
(TI-92: function cqfactor(cq))

```
CQ_FACTORS(cq):=CQ_LIST(DENOMINATOR(cq SUB 1))
```

"Dividing a C(Q)-point's denumerator by a prime factor out of the list created by CQ_FACTORS"
(TI-92: function remdenum(cq, row))

```
REM_DENUMER(cq,row):=DENOMINATOR(cq SUB 1)/
(CQ_FACTORS(cq)) SUB [row,1]^(CQ_FACTORS(cq)) SUB [row,2]
```

"Splitting the term of sum corresponding with a prime factor out of the list created by CQ_FACTORS"
(TI-92: function differen(cq, col))

```
DIFFERENCE(cq,row):=CQ_PLUS(cq,CQ_MULTIPLE(-(CQ_FACTORS(cq)) SUB row SUB 2,
CQ_PRIME((CQ_FACTORS(cq)) SUB row SUB 1)))
```

"Determining the sign of multiplicity of a term of sum associated with a prime factor out of the list created by CQ_FACTORS"
(TI-92: function expofac(cq, col))

```
EXP_FACT(cq,row):=IF(DENOMINATOR((DIFFERENCE(cq,row)) SUB 1)=
REM_DENUMER(cq,row),1,-1)
```

"List of prime factors congruent 1 mod 4 of a C(Q)-point's denumerator (including corrected multiplicities)"
(TI-92: furnction cqfaccor(cq))

```
CQ_FACTORS_CORRECT(cq):=VECTOR(REPLACE_ELEMENT((CQ_FACTORS(cq)) SUB i SUB 2*E~
XP_FACT(cq,i),(CQ_FACTORS(cq)) SUB i,2),i,1,DIMENSION(CQ_FACTORS(cq)))
```

"Determining the multiplicity of a term of sum associated with 2"
(TI-92 function ptminsum(cq) and function c2mult(cq))

```
POINT_MINUS_SUM(cq):=CQ_PLUS(cq,CQ_MULTIPLE(-1,CQ_SUM(CQ_FACTORS_CORRECT(cq))))

C2_MULTIPLE(cq):=IF((POINT_MINUS_SUM(cq)) SUB 1=0,2-(POINT_MINUS_SUM(cq)) SUB 2,
3+(POINT_MINUS_SUM(cq)) SUB 1)
```

"Complete list of a C(Q)-point's prime factors (including C2-factors; multiplicities with correct sign)"
(TI-92: function cqdecomp(cq))

```
CQ_DECOMP(cq):=IF(ABS(cq SUB [1]+cq SUB [2])=1,[[1,1],[2,C2_MULTIPLE(cq)]],APPEND([
[1,1],[2,C2_MULTIPLE(cq)]],DELETE_ELEMENT(CQ_FACTORS_CORRECT(cq),1)))
```

One of the functions, I missed in *DERIVE* (and on the *TI*-92 as well) was one to get the list of prime factors with direct access to the multiplicities (or »exponents«). PRIMFACTOR_LIST makes use of the ability of the ln-function to split base and exponent and arranges them in a matrix; in *DERIVE* this is a (n, 2) - matrix, on the *TI*-92 (for displaying reasons) a (2,n)-matrix containing the n prime factors in the 1st column (row), the corresponding multiplicities in the 2nd column (row).

It may seem strange that these lists also contain 1 (which is not considered to be a prime). But: As the decomposition-algorithm needs to know a list of the prime factors congruent 1 modulo 4, excluding the 1 would sometimes lead to an empty list as an output of »cqlist« on the TI-92 (and some programming-problems). In *DERIVE* 1 was included for compatibility-reasons (besides: When factorizing negative numbers, 1 is replaced by -1, thus indicating the sign of the original number!)

While DERIVE sorts the primefactor-list by itself, the *TI*-92 needs »mbubsort(mat, row, typ)« which provides the bubble-sort-algorithm for matrices; »row« selects the matrix-row used as sorting-key, »typ« can either be "a" (for ascending sorting) or "d" for descending sorting).

The part of the program about decomposition of C(Q)-points corresponds to the decomposition-algorithm listed before:

- CQ_FACTORS gets a list of the prime factors congruent 1 mod 4 (see the reason for the »trick« mentioned in the last paragraph?) of the point's reduced denumerator (which should be the same in both coordinates, as in *DERIVE* and the *TI*-92 fraction-reducing happens automatically).
- REM_DENUMER gets one of the denumerators prime factors congruent 1 mod 4, say p with multiplicity n, and divides the denumerator by p^n. The <u>rem</u>aining <u>denumer</u>ator is returned.
- DIFFERENCE gets p and n like REM_DENUMER and subtracts the n-times multiple of the generating element $f(p)$ from the C(Q)-point.
- EXP_FACT compares for the same prime factor p the results of REM_DENUMER and DIFFERENCE to distinguish, wheter the corresponding multiplicity has to be multiplied by -1 or not.
- CQ_FACTORS_CORRECT uses EXP_FACT to create the list of prime factors congruent 1 mod 4 of the point's reduced denumerator with corrected multiplicities.

Examples:

...with *DERIVE for WINDOWS* (file CQDEMO.MTH on disc):

#1: "Composition and Decomposition of"

#2: "Rational Points on the Unit Circle"

#3: "(Demo-file by P.K. Antonitsch)"

#4: list1 := $\begin{bmatrix} 2 & 1 \\ 5 & 3 \\ 13 & 1 \end{bmatrix}$

#5: "Composition of corresponding rational point:"

#6: point1 := CQ_COMP(list1)

#7: $\left[\frac{1184}{1625}, -\frac{1113}{1625}\right]$

#8: "Test, if <f(2)> is cyclic of order 4"

#9: list2 := $\begin{bmatrix} 2 & 5 \\ 5 & 3 \\ 13 & 1 \end{bmatrix}$

#10: point2 := CQ_COMP(list2)

#11: $\left[\frac{1184}{1625}, -\frac{1113}{1625}\right]$

#12: "(the same point although list differs)"

#13: "Let's try some decompositions:"

#14: point3 := $\left[-\frac{219336}{2088025}, -\frac{2076473}{2088025}\right]$

#15: ON_UNITCIRCLE(point3)

#16: 0

#17: "i.e. point3 lies on the unit circle"

#18: CQ_DECOMP(point3)

#19: $\begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 5 & 2 \\ 17 & -4 \end{bmatrix}$

#20: "Try point1, just to make sure the program works:"

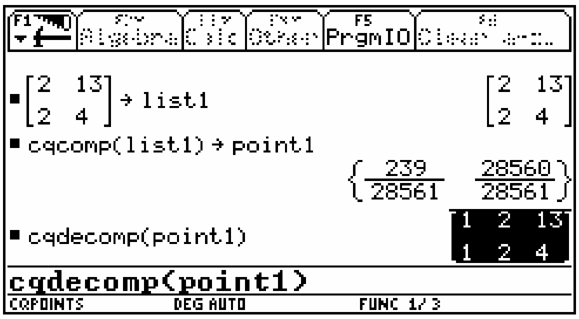
#21: CQ_DECOMP(point1)

#22: $\begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 5 & 3 \\ 13 & 1 \end{bmatrix}$

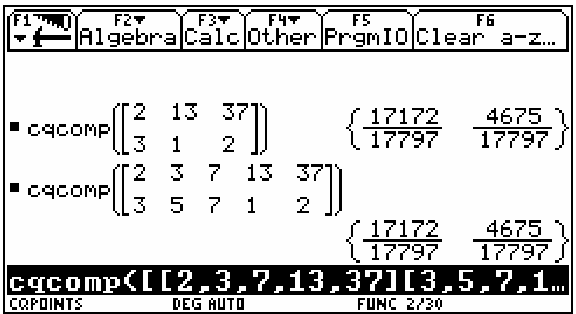
#23: "seemingly!"

...with the *TI*-92:

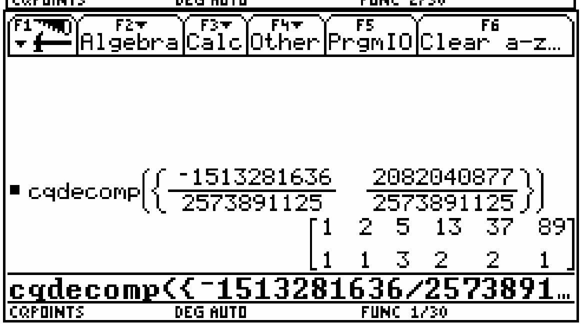
Composition and Decomposition of a rational point on the unit-circle: ↓



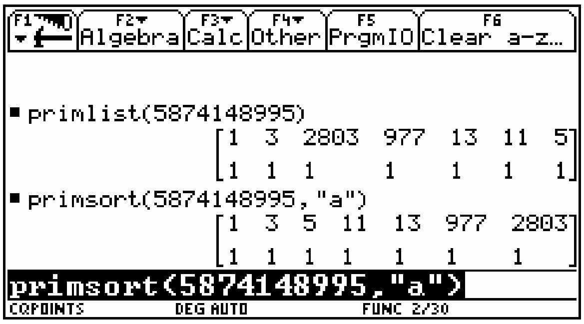
Primes ≡ 3 mod 4 don't change the point: ↓



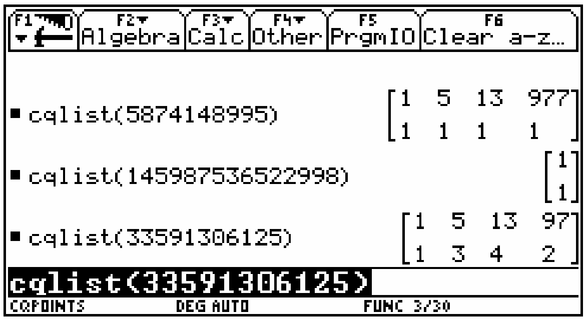
A time-consuming decomposition: →
(several minutes)



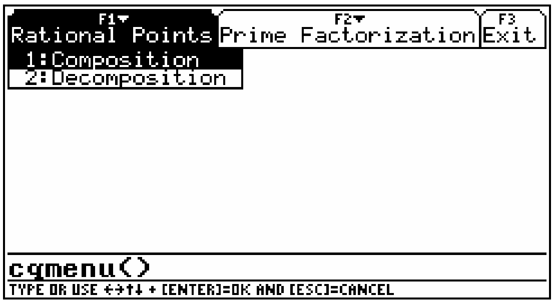
Prime-factorization: ↓

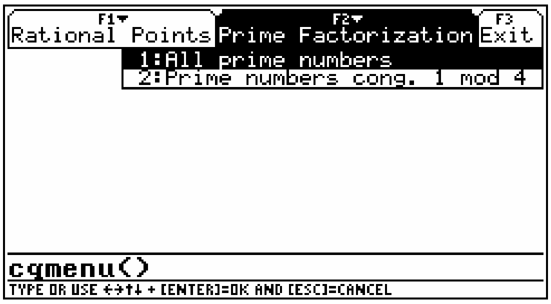


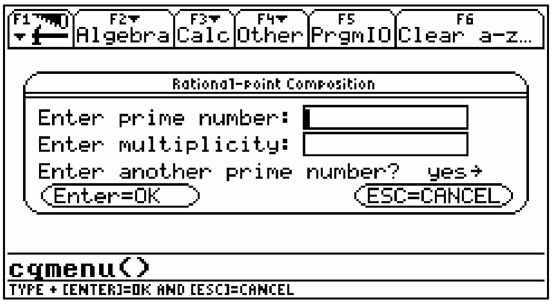
Factors ≡ 1 mod 4 only: ↓

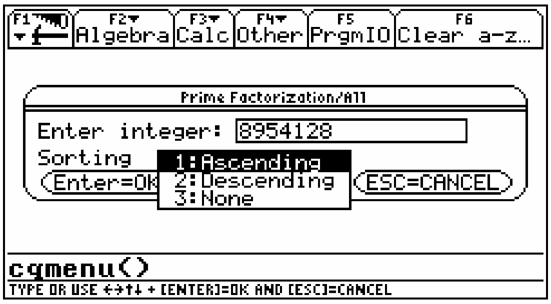


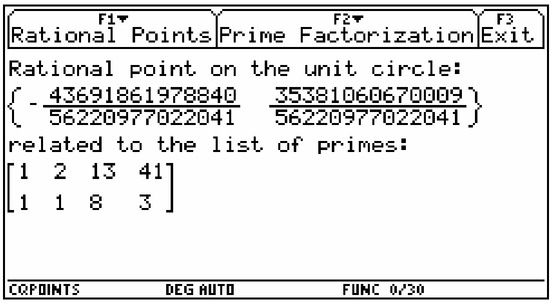
The »*TI*-92 - package« also contains the program »cqmenu()«, which provides a menu-oriented access (although I personally prefer the »line-oriented-interaction-style« making the re-use of obtained solutions much easier!?). Some screen-shots to illustrate the menu (make sure all the *TI*-92 files are stored in a folder »cqpoints« when using cqmenu()):

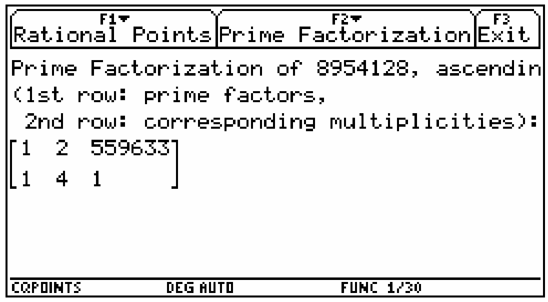












References:

[1] Lin Tan: »The Group of Rational Points on the Unit Circle« in: Mathematics Magazine Vol. 69, No. 3; June 1996

[2] P. Antonitsch: Geometrische Aspekte der Mathematik mit Computeralgebra-Systemen. Dissertation an der Universität Wien; Wien 1997

[3] I. N. Herstein: Topics in Algebra. Wiley Eastern Limited; New Delhi 1994 (16th reprint)

---

[1] more detailed presentation of the underlying concepts can be found in [1] or – with elaborated proofs – in [2].

[2] See also [2]

Titbits from Algebra and Number Theory(13)

by Johann Wiesenbauer, Vienna

Keeping an old promise these “Titbits” are devoted to purely algebraic stuff like groups and rings and what have you. Yes, it is true, there are no library functions - let alone built-in functions - on that score, but this doesn’t mean that DERIVE cannot be of any help when dealing with these topics. Unfortunately, since I cannot start from scratch for understandable reasons, I have to assume that the reader is already familiar with the basics of abstract algebra. But even if you don’t quite understand the algebraic background of the following routines, you may have a lot of fun when trying them out. Are you ready? Okay, let’s go!

First of all, I’d like to establish a polynomial arithmetic mod p (or more generally mod m, if this makes sense) that will be needed later on. Here are some useful functions in this field. The first two functions when applied to a given polynomial u in the indeterminate x will reduce its coefficients mod m using the built-in MOD- or MODS-function, respectively. (Readers who are connected to the DERIVE-mailbase will be already familiar with these definitions as well as with some others in the following.)

```
POLYMOD(u,m,x):=SUM(VECTOR(MOD(LIM(u_,x,1),m)·u_/LIM(u_,x,1),u_,
    TERMS(EXPAND(u))))

POLYMODS(u,m,x):=SUM(VECTOR(MODS(LIM(u_,x,1),m)·u_/LIM(u_,x,1),u_,
    TERMS(EXPAND(u))))
```

Furthermore, we need quite often a function that computes the degree of a given polynomial u in the indeterminate x. Although in general the following implementation is far superior to the one in MISC.MTH (check it!), it should be noted that u is supposed to be a polynomial with real coefficients due to a nasty DERIVE-bug regarding the option “Trivial” in the FACTOR-command. (Here and in the following I am always referring to version 4.09 of DfW.)

```
POLYDEG(u,x):=x·DIF(LN(DENOMINATOR(FACTOR(LIM(u,x,1/x),Trivial,x))),x)
```

What is so special about this function is that it makes no difference whether the polynomial u is in expanded form or not. (I am indebted to Alessandro Perotti for pointing out this fact to me!) The same goes for the next function which computes the leading coefficient of a polynomial u in the indeterminate x.

```
LEADCOEFF(u,x):=LIM(u/x^POLYDEG(u,x),x,inf)
```

The next new functions are basically the “mod p - versions” of the built-in standard functions QUOTIENT, REMAINDER and POLY_GCD. I also used a slightly improved form of the utility-function INVERSE_MOD from NUMBER.MTH which is included here for the sake of completeness.

```
INVERSE_MOD(a,m):=IF(GCD(a,m)=1,MOD((ITERATE(IF(MOD(α_,ß_)=0,[α_,ß_,γ_,δ_],
[ß_,MOD(α_,ß_),δ_,γ_–FLOOR(α_,ß_)·δ_]),[α_,ß_,γ_,δ_],[a,m,1,0]))SUB4,m))[1]

POLYQUOT(u,v,p,x):=ITERATE(POLYMOD(INVERSE_MOD(DENOMINATOR(q_),p)·
NUMERATOR(q_),p,x),q_,FACTOR(QUOTIENT(u,POLYMOD(v,p,x),x),Trivial,x),1)

POLYREM(u,v,p,x):=ITERATE(POLYMOD(INVERSE_MOD(DENOMINATOR(r_),p)·
NUMERATOR(r_),p,x),r_,FACTOR(REMAINDER(u,POLYMOD(v,p,x),x),Trivial,x),1)

POLYGCD(u,v,p,x):=IF(NUMBER(POLYMOD(v,p,x)),IF(POLYMOD(v,p,x),POLYQUOT(u,
LEADCOEFF(u,x),p,x),1),POLYGCD(v,POLYREM(u,v,p,x),p,x))
```

The next function is extremely useful when it comes to forming n-th powers of a polynomial u in the indeterminate x modulo both another polynomial w in x and the prime p. When looking into the code you will easily see that I just used a variant of the well-known “Square and Multiply”-algorithm.

```
POLYPOWER(u,n,w,p,x):=(ITERATE([IF(MOD(n_,2)=1,POLYREM(u_·v_,w,p,x),u_),IF(n_>0,
POLYREM(v_^2,w,p,x),v_),FLOOR(n_,2)],[u_,v_,n_],[1,u,n]))SUB1
```

[1] This function is not in the Titbits 13 file because INVERSE_MOD is implemented now in DERIVE 6, Josef.

POLYPOWER(u,n,w,p,x):=(ITERATE([IF(MOD(n_,2)=1,POLYREM(u_·v_,w,p,x),u_),IF(n_>0, POLYREM(v_^2,w,p,x),v_),FLOOR(n_,2)],[u_,v_,n_],[1,u,n]))SUB1

What's on? I see, yes, you're right! I should stop defining functions on and on and give some nice examples instead. What about computing the operation tables of some finite fields of low order? As you may recall for every q, where q is a prime power p^n, there is exactly one field $\mathbf{F}_q$ with q elements up to isomorphisms and it can be obtained by forming the factor ring

$$\mathbf{Z}_p[x]/(f(x))$$

where $\mathbf{Z}_p[x]$ is the polynomial ring over the residue class ring $\mathbf{Z}_p$ and (f(x)) is the principal ideal generated by an irreducible polynomial f(x) of degree n over $\mathbf{Z}_p$. Implementing this formula in DERIVE seems to be a tall order, doesn't it? Not at all! As a matter of fact, with our above poly-functions at hand, it's child's play!

TODIGITS(n,b,l):=IF(l,0,DELETE_ELEMENT(TODIGITS(n+b^l,b)),DELETE_ELEMENT(DELETE_ELEMENT(REVERSE_VECTOR(VECTOR(MOD(k_,b),k_,ITERATES(FLOOR(n_,b), n_,n))))))

POLYADDTABLE(w,p,x):=ITERATE(VECTOR(VECTOR(POLYMOD(TODIGITS(k_,p,d_)· VECTOR(x^i_,i_,d_-1,0,-1)+TODIGITS(l_,p,d_)·VECTOR(x^i_,i_,d_-1,0,-1),p,x),k_,0,p^d_-1), l_,0,p^d_-1),d_,POLYDEG(w,x),1)

ADDTABLE(w,p,x):=LIM(POLYADDTABLE(w,p,x),x,p)

POLYMULTTABLE(w,p,x):=ITERATE(VECTOR(VECTOR(POLYREM((TODIGITS(k_,p,d_)· VECTOR(x^i_,i_,d_-1,0,-1))·(TODIGITS(l_,p,d_)·VECTOR(x^i_,i_,d_-1,0,-1)),w,p,x),k_,0,p^d_-1), l_,0,p^d_-1),d_,POLYDEG(w,x),1)

MULTTABLE(w,p,x):=LIM(POLYMULTTABLE(w,p,x),x,p)

The first function TODIGITS(n,b,l) is an auxiliary function that is interesting on its own. It converts a number n into a vector of dimension l that contains the digits of representation of n w.r.t. the base b. If you don't want leading zeros, you should use this function in the form TODIGITS(n,b), i.e. without the third parameter. Is this the function Heinz Rainer Geyer was dreaming of in DNL #29, p.3 ? I don't know. At any rate, it is many times faster.

The second and third functions provide the addition tables of finite fields which are defined by means of the irreducible polynomial w in x over $\mathbf{Z}_p$. The only differ by the representation of the elements of the field which are polynomials or numbers, respectively. The fourth and the fifth function do the same for the multiplication of the field. Here are some examples. (At last!)

```
DisplayFormat:=Compressed

[POLYADDTABLE(x^2+x+1,2,x),POLYMULTTABLE(x^2+x+1,2,x)]

[[  0    1    x   x+1 ]   [ 0   0    0    0  ]]
[[  1    0   x+1   x  ]   [ 0   1    x   x+1 ]]
[[  x   x+1   0    1  ] , [ 0   x   x+1   1  ]]
[[ x+1   x    1    0  ]   [ 0  x+1   1    x  ]]

[ADDTABLE(x^2+x+1,2,x),MULTTABLE(x^2+x+1,2,x)]

[[ 0  1  2  3 ]   [ 0  0  0  0 ]]
[[ 1  0  3  2 ]   [ 0  1  2  3 ]]
[[ 2  3  0  1 ] , [ 0  2  3  1 ]]
[[ 3  2  1  0 ]   [ 0  3  1  2 ]]
```

POLYMULTTABLE(x^3+x+1,2,x)

0	0	0	0	0	0	0	0
0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
0	x	x^2	x^2+x	$x+1$	1	x^2+x+1	x^2+1
0	$x+1$	x^2+x	x^2+1	x^2+x+1	x^2	1	x
0	x^2	$x+1$	x^2+x+1	x^2+x	x	x^2+1	1
0	x^2+1	1	x^2	x	x^2+x+1	$x+1$	x^2+x
0	x^2+x	x^2+x+1	1	x^2+1	$x+1$	x	x^2
0	x^2+x+1	x^2+1	x	1	x^2+x	x^2	$x+1$

[ADDTABLE(x^3+x+1,2,x),MULTTABLE(x^3+x+1,2,x)]

0	1	2	3	4	5	6	7
1	0	3	2	5	4	7	6
2	3	0	1	6	7	4	5
3	2	1	0	7	6	5	4
4	5	6	7	0	1	2	3
5	4	7	6	1	0	3	2
6	7	4	5	2	3	0	1
7	6	5	4	3	2	1	0

0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7
0	2	4	6	3	1	7	5
0	3	6	5	7	4	1	2
0	4	3	7	6	2	5	1
0	5	1	4	2	7	3	6
0	6	7	1	5	3	2	4
0	7	5	2	1	6	4	3

I don't dare to give some more examples out of fear that Josef might frown on this waste of space, but you shouldn't miss having a look at the corresponding tables at least for q = 9 and q = 16.

What else can you do with that bundle of poly-functions defined above? Another important application that comes to my mind concerns factoring of polynomials with integer coefficients. A nice problem of this sort has been posed quite recently by Richard Schorn in the DERIVE-forum who was asking whether DERIVE can be used to factor the following polynomial

F(x):=25·x^16–184·x^14+4028·x^12–14600·x^10+27862·x^8–14600·x^6+4028·x^4–184·x^2+25

over **Z** (or, what amounts to the same, over **Q**).

He had encountered this polynomial when dealing with the following famous problem: Does there exist a box with integer sides such that the three face diagonals and the main diagonal all have integer lengths? As a matter of fact, the answer to this question would be "yes", if there were any integer values for x different from –1,0,1 such that f(x) is a perfect square.

The first thing we should do is to check whether f(x) has multiple roots. Because of

POLY_GCD(F(x),DIF(F(x),x))=1

this is not the case (note that we are using the built-in POLY_GCD-function here!) The same check mod p is performed by the following function:

POLYSQUAREFREE(u,p,x):=NUMBER(POLYGCD(u,DIF(u,x),p,x))

Let's select e.g. all primes below 100 such that f(x) is squarefree mod p:

SELECT(POLYSQUAREFREE(F(x),p_,x),p_,SELECT(PRIME(q_),q_,1,100))=[3,7,11,13,19,23,29, 31,37,41,43,47,53,59,61,67,71,73,79,83,89,97]

Now let's consider the following function

POLYDIVS(u,d,p,x):=POLYGCD(POLYPOWER(x,p^d,u,p,x)-x,u,p,x)

which yields for a given squarefree polynomial u in x over $\mathbf{Z}_p$ with no irreducible factors of degree < d the product of all its irreducible factors of degree d. We use it to select all primes p of the above set such that f(x) has no linear factors mod p.

SELECT(NUMBER(POLYDIVS(F(x),1,p_,x)),p_,[3,7,11,13,19,23,29,31,37,41,43,47,53,59,61,67,71, 73,79,83,89,97])=[3,7,11,13,19,23,31,43,47,59,67,71,79,83,89]

Since the output is not the empty set we may conclude from this that f(x) has no linear factors over **Z** either. Repeating this step for all primes of the above set and for factors of degree 2 and 3 we get

SELECT(NUMBER(POLYDIVS(F(x),2,p_,x)),p_,[3,7,11,13,19,23,31,43,47,59,67,71,79,83,89])= [3,7,11,23,31,71,79]

SELECT(NUMBER(POLYDIVS(F(x),3,p_,x)),p_,[3,7,11,23,31,71,79])=[3,7,11,23,31,71,79]

Again, since there are still primes p such that f(x) has no factors of degree < 4 over $\mathbf{Z}_p$ the same must be true for f(x) when viewed as polynomial over **Z**. This changes dramatically for the next possible value of d, namely d=4.

SELECT(NUMBER(POLYDIVS(F(x),4,p_,x)),p_,[3,7,11,23,31,71,79]) = [ ]

VECTOR(POLYQUOT(F(x),POLYDIVS(F(x),4,p_,x),p_,x),p_,[3,7,11,23,31,71,79]) = [1,4,3,2,25,25,25]

Moreover the second line shows that after dividing by the product of all irreducible factors of degree 4 only constants are left, i.e. our polynomial f(x) is for all remaining seven primes p a product of irreducible factors of degree 4 over $\mathbf{Z}_p$. Although it is rather likely now that the same is true for f(x) when considered as a polynomial over **Z**, strictly speaking we can only conclude from this that the degrees of all irreducible factors of f(x) are divisible by 4.

Now comes the hardest part of our computations - everything we have done so far was easy by comparison! - namely the splitting up of f(x) into irreducible factors of degree 4 for one of the primes above. Assuming that g(x) is any squarefree polynomial whose irreducible factors mod p for some odd prime p are all of degree d, then according to a theorem of Cantor and Zassenhaus we have the identity

$$g(x) = \gcd(g(x), t(x))\gcd(g(x), t(x)^{(p^d-1)/2} + 1)\gcd(g(x), t(x)^{(p^d-1)/2} - 1)$$

for all polynomials t(x). The chances are about 50% that this will result in a splitting of g(x). Choosing for t(x) a random polynomial of degree < 2d, an implementation of this idea could look like this (note that I had to resort to one of those notorious _AUX-functions - a very rare event indeed, which clearly indicates that programming this function wasn't exactly a picnic!).

```
POLYSPLIT_AUX(u,udeg,d,p,x):=IF(udeg=d,[u],ITERATE(IF(LIM(r_·(udeg-r_),r_,
POLYDEG(t_,x))>0,IF(NUMBER(s_),[POLYQUOT(u,t_,p,x),t_],[s_,t_]),[s_+1,POLYGCD(u,
POLYPOWER(SUM(RANDOM(p)·x^k_,k_,0,2·d-1),(p^d-1)/2,u,p,x)-1,p,x)]),[s_,t_],[0,1]))

POLYSPLIT(u,d,p,x):=ITERATE(APPEND(VECTOR(POLYSPLIT_AUX(e_,POLYDEG(e_,x),d,p,
x),e_,f_)),f_,[u],3)
```

The rest is plain sailing! Taking e.g. p=79 from the list above we get

RANDOM(-1)=1

POLYSPLIT(F(x),4,79,x)=[25·x^4+60·x^3+30·x^2+59·x+5,x^4+75·x^3+6·x^2+12·x+5,x^4+45·x^3+ 17·x^2+64·x+16,x^4+4·x^3+6·x^2+67·x+5]

This was actually the longest computation of all, which took 22s on my Pentium 166 PC! (Note that due to the random polynomial t(x) your factorization may look different without the initializing of the random number seed!) To get possible factors of f(x) over **Z**, we have to scale these four factors mod 79 in such a way that their first and last coefficients become divisors of 25. In addition, we take the least absolute residues mod 79 of all coefficients using POLYMODS.

(F1(x):==POLYMODS(INVERSE_MOD(5,79)·(25·x^4+60·x^3+30·x^2+59·x+5),79,x))=
5·x^4+12·x^3+6·x^2–4·x+1

(F2(x):==POLYMODS(x^4+75·x^3+6·x^2+12·x+5,79,x))=x^4–4·x^3+6·x^2+12·x+5

(F3(x):==POLYMODS(5·(x^4+45·x^3+17·x^2+64·x+16),79,x))=5·x^4–12·x^3+6·x^2+4·x+1

(F4(x):==POLYMODS(x^4+4·x^3+6·x^2+67·x+5,79,x))=x^4+4·x^3+6·x^2–12·x+5

This looks good, doesn't it? Again, we use DERIVE to show that appearances hadn't been deceptive in this case.

F(x)=EXPAND(F1(x)·F2(x)·F3(x)·F4(x))=true

When looking at the four factors you may have noticed striking similarities. Of course, this is not the working of mere chance, but reflects certain symmetries of our original polynom f(x). Concluding let's take a closer look at this phenomenon. To this end we define the following two automorphisms of the polynomial ring **Z**[x], namely

α(u,x):=LIM(u,x,-x)

ß(u,x):=x^POLYDEG(u,x)·LIM(u,x,1/x)

Because of

α(F(x))=F(x)=true
ß(F(x))=F(x)=true

f(x) is invariant under α and ß and the same must be true for the composition αß. Hence, if you can get hold of any factor h(x) of f(x), you get the three remaining ones for free by applying α, ß and αß to h(x)! (Check it!) Well, let it go at that. Email me, if you come up with some nice application of the functions above! (J.Wiesenbauer@tuwien.ac.at)

On May 27 I wrote a message to Al Rich, SWHH:

I came across an interesting problem with *DERIVE for DOS* 4.x. When I call *DERIVE* the first time after turning on my computer I receive the following message:

Loading DPMI host . . . . Unhandled exception 000D at 00EF CDEF ErrCode 0000

I ignore that message and try once more .... no problems. *DERIVE* appears on my screen as expected. Do you know any reason for that strange behaviour?

I didn´t wait long for an answer. May 28:

Hello Josef,
You might consider including the following explanation as a note in the DUG Newsletter:

DfD is a 32-bit program that must run on a 16-bit operating system (i.e. MS-DOS). Switching the processor back and forth between 16-bit and 32-bit mode requires that a DPMI host be loaded. If *DfD* is run in a "DOS box" under Windows (3.1, 95, 98, or NT), Windows automatically loads a DPMI host, and the above error does not occur.

However, if *DfD* is run directly under MS-DOS, it automatically loads the DPMI host called EDPMI.OVL, that is included for free on the *DfD* distribution diskette. Unfortunately, there seems to be a bug in EDPMI.OVL from Ergo Systems that causes that unhandled exception to occur on some computers. However, it seems to be a harmless error and simply restarting *DfD* again usually works successfully.

KEPLER'S CLOCK, OR, PERHAPS, A TART ?

Tomass Romanovskis, Riga, Latvia

Kepler's equation in astronomy

The planets move along an ellipse, in one focal point of which the Sun is situated. In order to fix the positions of a planet after equal time spans, say, after each month, one must know, how to divide the ellipse into 12 sectors of equal area with the centre in the focus. The solution of this problem may be found with the aid of Kepler's equation

$$E + \varepsilon \sin(E) = k\pi/6, \text{ with } k = 1,...,12.$$

Although Kepler's equation has been playing an important part in astronomic practice and in teaching for several centuries, there exists a tendency in high school and university programs, as well as in textbooks on physics and mathematics to avoid this theme. It may be understandable, for there did not exist a means of obtaining a rationally usable solution in an acceptable time interval. The situation has changed with the appearance of computer algebra. Kepler's equation may be inscribed by means of one expression.

How to divide a circle into sectors of equal areas, if the centre of division does not coincide with the centre of the circle ?

There exists a simple problem in geometry, which I have not so far encountered in the literature.

It is well known that a circle can easily be divided into 12 sectors of equal area, using only a pair of compasses, since this problem coincides with that of constructing a regular dodecangle inside a circle.

But how to divide a circle into sectors of equal area, if the centre of division does not coincide with the centre of the circle ?

This problem may be solved by constructing a Kepler clock, which shows the position of a planet with respect to the aphelion after equal time intervals. One may similarly divide a tart into equal parts, if the centre of the tart does not coincide with its centre.

Theory

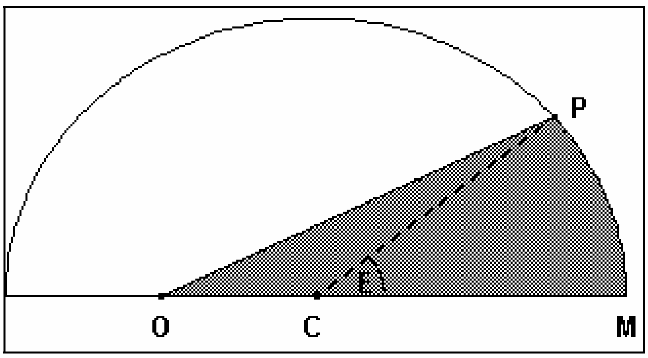


Fig. 1. How to divide the unity circle into sectors (POM) of *n* equal areas (π/n), if the centre of division does not coincide with the centre of the circle ?

Let us choose a point *P* on the circumference of a circle, which subtends an angle *E*, looking from the centre of the circle. Let us calculate the area of the sector *POM*. The area equals the sum of the area of the central sector *PCM* and that of the triangle *POC*.

$$0.5\ E + 0.5\ \varepsilon \sin(E) = S(POM).$$

The area of the unity circle equals π. We desire to divide the circle into 12 equal parts, which correspond to areas *kπ/12*, where *k = 1,...,12*. Inserting this expression for the area into the area of the sector *S(POM)*, we obtain the following equation:

$$E + \varepsilon \sin(E) = k\pi/6,$$

which is Kepler's equation. It is, as may be seen, a transcendental equation with respect to angle *E*. This is the reason that prevents the introduction of this equation into the teaching of natural science. However, applying DERIVE, the situation changes, since the solution may be written down with the aid of one expression (cf. expression #2). Since the problem is symmetrical with respect to the central axis, we may satisfy ourselves with *k = 1,2,3,4,5,6*. In astronomy Kepler's equation is usually solved for small values of ε. We choose purposedly a large value - ε = 0.5, in order to make the solution clearer.

```
#1 [Angle:=Radian, Precision:=Approximate,PrecisionDigits:=4]

                   180                        kπ     kπ
#2 E(ε):=Vector[————ITERATE[-ε sin(t)+ ——,t, ——],k,0,6]
                    π                          6      6

#3 E(0.5)=[0, 20.13, 41.14, 64.20, 91.36, 127.17, 180]
```

All that remains now is to write down two more expressions for drawing the unity circle

```
#4  x^2 + y^2 = 1
```

and for the vector of the segments, which connect the apex *O* with the points on the rim of the circle, visible at angles *E* and *-E*:

```
#5        VECTOR ([[0,-0.5],[sin(α°), cos(α°)]],α,E(0.5))
#6   VECTOR ([[0,-0.5],[sin(-α°),cos(-α°)]],α,E(0.5))
```

In order to make the picture clearer, it is turned by 90 degrees with respect to Fig.1 by means of the plot command. The second drawing shows, what the circle looks like, if it is divided into 12 parts of equal area, the apex having been displaced by half of the radius.

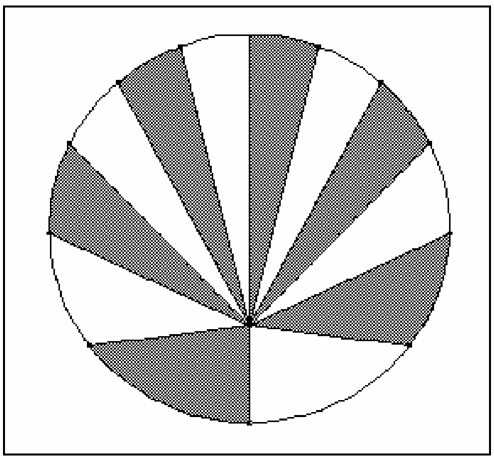


Fig.2. A tart, divided into 12 equal parts, by means of solving Kepler's equation (expression #3).

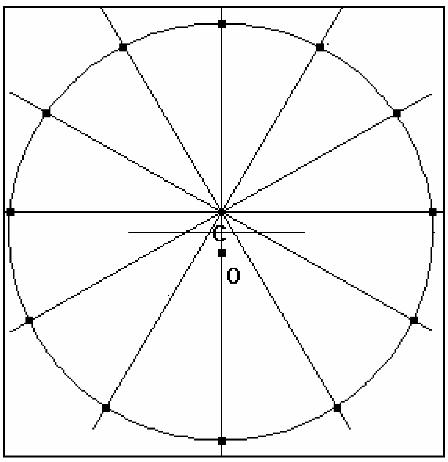


Fig.3. The hand of a Mars clock turns around a point, displaced by r/10 with respect to the centre *C*, and by r/5 with respect to the location *O* of the Sun.

The Mars clock

Kepler discovered the laws of planet motion, studying observation data of Mars. For Mars we have $\varepsilon \approx 0.1$. For such a small value of ε one may consider that Mars moves along a circular orbit, with the Sun at point *O*, which is displaced by the value $-\varepsilon$ from the centre of the circle. Applying the function *E(0.1)* = calculate and mark the location of Mars on the orbit with respect to the aphelion. After that, simplifying the command

```
VECTOR([0,0.1],[1.2SIN(α°),0.1+1.2COS(α°),α,E(0.1)),
```

draw the rays of a regular dodecangle having the centre at a distance $+\varepsilon$ from the centre. In order that the rays coming out of the centre of the circle, cross the periphery, the radius of the drawing is chosen equal to 1.2. One may see (cf. Fig. 3) that these rays, coming out of the point $+\varepsilon$, cross the orbit of Mars approximately at those points, which divide the circle into sectors of equal area, with the apex at point $-\varepsilon$. So, a Mars clock may be made in the following way. We draw a circle with centre *C*. At distance $-\varepsilon$ we put the Sun. The hand of the clock we put at point $+\varepsilon$. The point of intersection of the hand with the circumference will show, where Mars is positioned at the given time with respect to the aphelion. In order that the clock shows the position of Mars in real time, it must complete one full revolution in 687 days.

DERIVE & TI-92 User Forum 2

continued from User Forum 1

Since the given polynomial is obviously invariant under the operation of reversing the order of coefficients, every factor of it will be mapped under this automorphism onto another factor. In our case we have

$$x^4 - 4x^3 + 6x^2 + 12x + 5 \;\text{->}\; 5x^4 + 12x^3 + 6x^2 - 4\cdot x + 1.$$

Since the given polynomial is obviously invariant under the operation of reversing the order of coefficients and changing additionally the signs of coefficients of odd powers of x, every factor of it will be mapped under this automorphism onto another factor. In our case we have

$$x^4 - 4x^3 + 6x^2 + 12x + 5 \;\text{->}\; 5x^4 - 12x^3 + 6x^2 + 4\cdot x + 1.$$

Since the given polynomial is obviously invariant under the operation of changing the signs of coefficients of odd powers of x, every factor of it will be mapped under this automorphism onto another factor. In our case we have

$$x^4 - 4x^3 + 6x^2 + 12x + 5 \;\text{->}\; x^4 + 4x^3 + 6x^2 - 12\cdot x + 5.$$

There is only thing that remains to be done, namely to prove that one (and therefore all!) of the above factors are irreducible over Q. Needless to say that this is a push-over for DERIVE.

Therefore to be a real challenge this problem should be posed without any hints. Cheers, Johann Wiesenbauer

Needless to say that *DERIVE* 6 has no problems factorizing the polynomial of order 16 presented by Richard Schorn, Josef

Jan Vermeylen, Belgium **math@rhombus.be**

Take any prime number P.
What is the probability that P-1 is divisible by 6 ?
Is the answer really 1/2 ?

Johann Wiesenbauer, Austria **J.Wiesenbauer@tuwien.ac.at**

Suppose that a and d are natural numbers s.t. gcd(a,d)=1 and let

pi SUB [d,a](x) = # {primes <=x s.t. p = a mod d}

and

pi(x) = # {primes <=x}

for any real number x > 0. According to a celebrated theorem by de la Vallée Poussin the following holds

lim (pi SUB [d,a](x)/ pi(x), x, inf) = 1/euler_phi(d)

Setting a=1 and d=6 you get for that probability

1/euler_ph(6) = 1/2,

that is your guess was correct.

Cheers, Johann

Terence Etchelles, UK **T.A.Etchells@livjm.ac.uk**

Hello, All

Mmm, Not that I don't believe my good friend and superb number theorist Johann that the probability is 1/2. I like sometimes to validate some results with an experiment, a belt and braces approach:

```
ITERATES(NEXT_PRIME(n),n,1,1000)
```

Finds the first 1000 prime numbers (of course 1 is not prime)

```
VECTOR(MOD(p-1,6),p,ITERATES(NEXT_PRIME(n),n,1,1000))
```

Returns a zero for every p-1 divisible by 6

```
SELECT(m=0,m,VECTOR(MOD(p-1,6),p,ITERATES(NEXT_PRIME(n),n,1,1000)))
```

selects all the zeros in a vector

```
DIMENSION(SELECT(m=0,m,VECTOR(MOD(p-1,6),p,ITERATES(NEXT_PRIME(n),
          n,1,1000))))
```

Finds the dimension of the vector

491

Roughly 1/2. Looks like pure mathematics wins the day!!!!!

Cheers, Terence

Johann Wiesenbauer, Austria **J.Wiesenbauer@tuwien.ac.at**

Hi Terence,

Many thanx for that nice introduction of your email. Somehow I feel that I should show you my appreciation by providing a simple routine which might make counting primes a little bit easier in the future.

```
PRIMEPI(x, d, a) := IF(d, 0, LIM(SUM(PRIME(d·n_ + a), n_, 0,
           (x - a)/d), [true, false], [1, 0]),PRIMEPI(x, 6, 1) +
           PRIMEPI(x, 6, 5) + IF(x >= 3, 2,FLOOR(x, 2)))
```

This simple routine counts all primes p <= x, where x is any positive real number, such that p is of the form p = kd + a for some k>=0. (As usual, d and a are supposed to be coprime natural numbers.) PRIMEPI(x) will count all primes up to x, that is d and a can also be left out in a function call.

Here is a small example that took DERIVE 1.4s on my Pentium 166 PC:

```
[PRIMEPI(10000, 6, 1), PRIMEPI(10000, 6, 5) + 2, PRIMEPI(10000)]
[611, 618, 1229]
```

Programmers who fear neither death nor devil may try to beat these times!
(Ok, ok, I haven't yet blown my mind. That last remark was only to egg on the people out there!)

Cheers, Johann

John Alexiou **ja72@prism.gatech.edu**

Hi, here is a problem I have: Let's say we have an expression like

`#1 (Iz+m*L^2)/(C+L)^2` and by

`#2 SOLVE(DIF(#1,L)=0)` I get a list of three extrema

`#3 [L=Iz/(m*C) , L=inf, L=-inf]`

My problem is that I want to create a vector with the limit of #1 as L approaches these three values.

A vector of the substitution variables and values is obtained by

```
#4 LHS(#3) = [L,L,L]
#5 RHS(#3) = [Iz/(m*C),inf,-inf]
```

if I try the obvious

`#6 LIM( #1 , [L,L,L] ,[Iz/(m*C),inf,-inf] ) = m*Iz/(Iz+m*C^2)`

the result is a single quantity with the only first substitution done. In order to make a vector I tried

`#7 LIM( [#1,#1,#1] , [L,L,L] ,[Iz/(m*C),inf,-inf] )` = vector of three `m*Iz/(Iz+m*C^2)` which is wrong again.

It seems that I need to explicity tell it what to do with

`VECTOR( LIM(#1,L,[Iz/(m*C),inf,-inf] SUB i) , i,3 )`

but that means that I need either to know the the size of #3 or use the DIMENSION() function which requires the SOLVE for the extrema a second time just to get the size of the solution.

It would seem logical to extend the capablilities of the LIM function to handle vectorized inputs as multiple expressions and return multiple resuls. Doesn't this seem logical to everybody else??

john alexiou.

Jan Vermeylen, Belgium math@rhombus.be

Dear Mister Alexiou,
Please find attached a small Derive function that seems to solve your problem:

the function ALEX (expr , var_) simplifies to the vector of limits of the expr with var_ goining to all zeros of the derivative of expr with respect to var_

Best regards, Jan Vermeylen,

ALEX(expr,var_):=VECTOR(LIM(expr,var_,val,0),val,RHS(SOLVE(DIF(expr,var=_)=0,var_)))

ALEX((Iz+m*L^2)/(C+L)^2,L) = [Iz*m/(C^2*m+Iz),m,m]

Hellmut Scheuermann, Germany H.Scheuermann@schule.uni-frankfurt.de

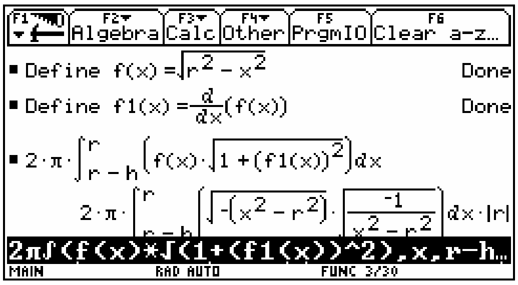
Dear Josef,

A couple of days ago together with my students I tried to find the formula for the surface area of a sphere´s segment using the *TI*-92. It can easily be done using *DERIVE*. I like that problem, because it is necessary to state some definitions to reach the wellknown formula O = 2*r*pi*h. That is a fine example for the fact that the PC - or better, the program - does not all the work and that it sometimes requires more mathematical reasoning than calculating by hands. See the *DERIVE* listing:

```
CaseMode:=Sensitive
f(x,r):=SQRT(r^2-x^2)
f1(x,r):=DIF(f(x,r),x)
;Simp(#3')
f1(x,r):=-x/SQRT(r^2-x^2)
O(h,r):=2*pi*INT(f(x,r)*SQRT(1+f1(x,r)^2),x,r-h,r)
;Simp(#5')
O(h,r):=4*pi*r^2*SIGN(h)-2*pi*SIGN(h)*ABS(r*(h-2*r))
```

At that point the students were surprised and discussion started:

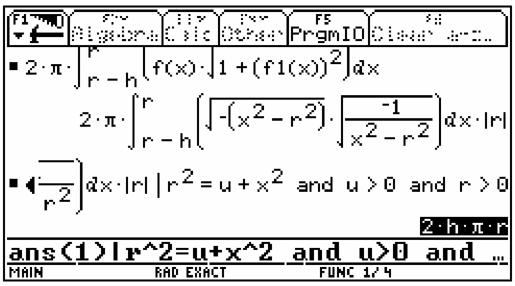
```
default:epsilonReal [0, inf)
;Simp(#5')
O(h,r):=4*pi*r^2-2*pi*r*ABS(h-2*r)
;Sub(#8)
O(h,r):=4*pi*r^2-2*pi*r*ABS(u)
u:epsilonReal (-inf, 0]
;Simp(#9)
4*pi*r^2+2*pi*r*u
;Sub(#11)
4*pi*r^2+2*pi*r*(h-2*r)
;Simp(#12)
2*pi*h*r
```



But now to my problem: I entered the following three expressions:
..... and the *TI* is not able to solve or simplify the integral. I tried to set conditions according to the *DERIVE* - listing, but it did not luck.

DNL:

You have to force the TI-92 to simplify the two square roots to -1. That needs some "bad" tricks. As you can see on the screen shot I found one "trick" to overcome the TI´s misbehaviour.
Josef



(The substitution $r^2 = u + x^2$ transforms $r^2 - x^2$ to u, u > 0 makes the root unique and r > 0 changes |r| in the result to r).

DIOPHANTINE EQUATIONS (1)

Alfonso J. Población Sáez
E.U. Politécnica, Valladolid, España

Well, as you surely have supposed, the relationship among the questions proposed in the last ACDC is that they require solutions in integers. Let us see some ways to find the solutions with *DERIVE* - I must better say, the ones that occur to me.

First Problem: Let n be the number. We need that $n \equiv 13 \pmod{17}$ and $n \equiv 5 \pmod{12}$. These conditions lead us to the equation $17x - 12y = -8$. We will solve this equation as if we don not know anything from the theory of numbers:

```
#1: [p:=17,q:=-12,r:=-8]
#2: [x:=,y:=(r-p*x)/q]
#3: ALL(a,b):=VECTOR(IF(y=FLOOR(y),[x,y],0),x,a,b)
#4: SOL(a,b):=SELECT(k/=0,k,ALL(a,b))
#5: TEST(a,b):=IF(MOD(r,GCD(p,q))/=0,"No integer solutions",SOL(a,b))
```

For example simplify TEST(1,200) and we will have all integer solutions of the equations from 1 through 200. We can see that all these are of the form [$x = 8 + 12t$, $y = 12 + 17t$]. So to find n, we can use

```
#6: n_s:=VECTOR(149+204*t,t,0,100)
#7: SELECT(n>999 AND n<10000,n,n_s)
```

After simplifying #7 (0.2 sec), we find that the smallest four-digit number is 1169 and the biggest is 9941, with 44 four-difit numbers that satisfy the stated conditions.

We can also solve the **Second Problem:** from the conditions we obtain 79y + 399z = 1900 (y represents the number of the pigs and z the number of the cows). TEST(1,100) gives y = 19, z = 1, and hence x = 80 (number of hens).

This file, although easy and fast, is not very orthodox from the mathematical point of view. It only takes advantage from the computer´s brute force. Besides this, if solutions or coefficients are big enough, we can waste a lot of time making guesses (try for example 3154x + 2971y = 45, look for a solution with TEST,..... and good luck). So I will explain another file based in continued fractions, the usual way to solve these problems (see, for example, reference [1] for details). I make use of the function which Mr Rich gave us in DNL#20 `CONTINUED_FRACTION(u,n)`. (I shortened its name to `C_F(u,n)`).

```
#1:  C_F(u,n):=FLOOR(ITERATES(1/MOD(x_),x_,u,n))
#2:  [num:=, P(n):=, Q(n):=]
#3:  w:=C_F(num,20)
#4:  P_AUX(n):=IF(n=1,w SUB 1,1+w SUB 1*w SUB 2)
#5:  P(n):=IF(n>2,P(n-1)*w SUB n+P(n-2),P_AUX(n))
#6:  Q_AUX(n):=IF(n=1,1,w SUB 2)
#7:  Q(n):=IF(n>2,Q(n-1)*w SUB n+Q(n-2),Q_AUX(n))
#8:  X(n):=P(n)/Q(n)
#9:  TABLE(k):=VECTOR([n,w SUB n,P(n),Q(n)],n,1,k)'
```

From functions P(n) and Q(n), we can illustrate and verify some results concerning continued fractions (and even to tell our students some historical aspects of the question). For example, we can build an increasing convergent sequence of fractions X(2k+1) to a real number `num`. (For instance setting `num :=` π, and simplifying `TABLE(10)` we obtain 3 < 33/106 < 103993/33102 < 208341/66317 < 833719/265381 < .... < π or a decreasing one X(2k): 22/7 > 355/113 > 104348/33215 > 312689/99532 > .... > π).

Or we can verify that $P(n)\,Q(N-1) - P(N-1)\,Q(n) = (-1)^n$. Warning: Be sure that the precicision is high enough, because if not, you may obtain mistaken results like $\pm\infty$or ?.

[1)] **AC DC** is **A**mazing (or **A**musing **C**orner of the ***DERIVER***´s **C**uriosities) is founded by Alfonso Población.

When we set `num` a rational number the former sequences are finite. This leads us to the solutions of a linear diophantine equation a x – by = ± 1: when $P_N / Q_N = a/b$, then $x = Q_{N-1}$, $y = P_{N-1}$.

Now try to solve 3145 x + 2971y = 1, and hence 3145 x + 2971 y = 45. Unfortunately we will not always reach a satisfactory solution: try this file for the equation of the second problem. A good strategy could be a mixed use of the first and the second approach.

Finally by now, a question to all of you - I have no answer: Why with approX the computing time is almost half of that using Simplify, if we work always with integer numbers? (It is referred to the second file).

References:

[1] LeVeque, William J., *Elementary Theory of Numbers,* Addison-Weslex Publishing Company, INc. (1962)

Historical resources:

[2] Boyer, Carl B., *A History of Mathematics*, John Wiley & Sons, Inc. (1968)

[3] Kline, Morris, *Mathematical thought from ancient to modern times,* Oxford University Press (1972)

A PROGRAM FOR SOLVING DIOPHANTINE EQUATIONS

Leandro Tortosa and Javier Santacruz

Let´s suppose that we want to solve a problem like this.

Determining the amount of computers that you can buy of each of the following prices : 290.000 pts and 170.000 pts. if you have a budget of 7.800.000 pts.

The raising of the problem is not difficult; we must write the equation that solves this problem, and that is

$$\mathbf{29\,x + 17\,y = 780}$$

Equations of this type are called ***Diophantine equations***.

We express a diophantine equation as

$$a\,x + b\,y = c$$

where a, b and c are integer numbers and the unknows are x, y. We are going to call **d** the greatest common divisor of the numbers (a,b). We can state and prove a sufficient condition to establish if the equation has a solution or not. The condition is that the diophantine equation will have solution if and only if d divides c, or in other way, if c = k*d , being k some integer.

The process which we are going to develop to get the solution of some diophantine equation can be summarised in the following steps :

1. We get the greatest common divisor of the two integers, d.
2. We prove if there is a solution using the above condition.
3. We get the particular solution of the equation d = a·*x + b * y using something like a backward regression as we´ll see afterwards. This solution is labeled (x_0 , y_0).
4. We get the particular solution of the diophantine equation c=a*x+b*y multiplying (x_0 , y_0) by k (remember that c = k·* d).
5. We obtain the general solution of the diophantine equation by substituing in the following expression :

Diophantic equation: $a \cdot x + b \cdot y = d \cdot n$

Let´s α, β such that

$a = \alpha \cdot d$
$b = \beta \cdot d$

The general solution is:

$x = x_0 + k \cdot \beta$
$y = y_0 - k \cdot \alpha$ with k an integer and (x_0, y_0) the particular solution.

As we can see from the last expression that give us the general solution, a diophantine equation has an infinite number of solutions. In some problems we will have to apply some conditions in order to get the particular solution in which we are concerned, but that will be after obtaining the general solution.

We have already given a description of a method to get the solution of a equation like the one we had at the begining, so that we are going to solve it following the scheme given in the above list.

$29 \cdot x + 17 \cdot y = 780$

29 = 1· 17 + 12
17 = 1· 12 + 5
12 = 2· 5 + 2
5 = 2· 2 + 1
2 = 2· 1 Then, the highest common factor is: 1.

Now, we have the greatest common factor, that is 1. We must continue the algorithm to get the whole solution by getting the solution of the Bezout identity like

1 = 5 - 2 · 2 = 5 - 2 (12 - 2 · 5) =
= (-2) 12 + 5 · 5 = (-2) 12 + 5 (17 - 12) =
= 5 · 17 + (-7) 12 = 5 · 17 + (-7) (29 - 17) =
= (-7) 29 + 12 · 17

Then, if we look at this, we realised that the particular solution of the Bezout identity is → (-7 , 12).

To get the particular solution of the equation 29 · x + 17 · y = 780 is very simple. We only have to multiply the above solution by 780. We can establish that the particular solution for the equation 29 · x + 17 · y = 780 is → (- 5460 , 9360).

Now, we are going to obtain the general solution. Following the above considerations, we have

mcd (29,17) = 1 So that $29 = \alpha \cdot d$, $17 = \beta \cdot d$ $\xrightarrow{\text{then}}$ $\alpha = 29$, $\beta = 17$

Then, the general solution is :

x = - 5460 + 17 k
y = 9360 - 29 k being k an integer.

But we have not calculate completely the solution of the problem because we can introduce in the infinite solutions we get an elementary condition : we are talking about computers and the number of conputers in anyproblem always must be a positive number , that is, we can make a restriction to numbers greater and equal to zero. Let´s do more calculations

x = - 5460 + 17 k > 0
y = 9360 - 29 k > 0 $\xrightarrow{\text{then}}$ $k \ge 321.17$, $k \le 322.75$ $\xrightarrow{\text{then}}$ $k = 322$

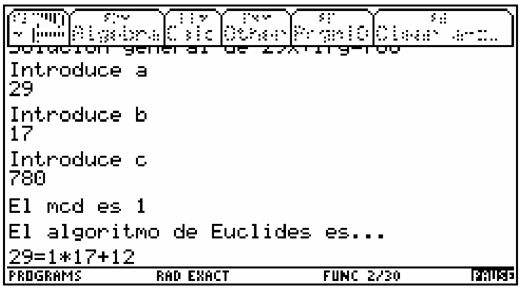
Substituing now the value of k in the expression for the general solution we obtain the final and unique solution:

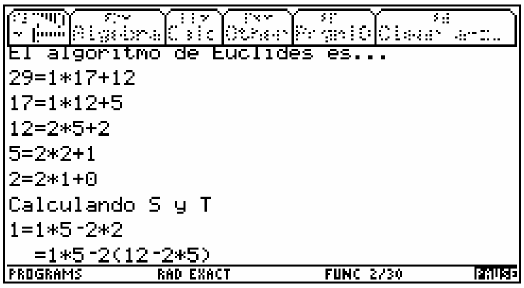
x = 14
y = 22

So we have calculated by this method the solution of the problem initialy proposed.

We have developed a program for TI-92 that describes exactly the whole process we have performed by hand, in order to get the solutions of diophantine equations.

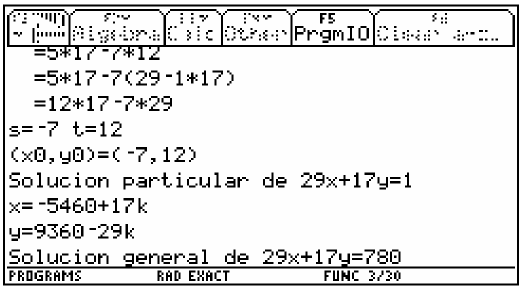
If we run the program with the data we have used in the example of the begining, we get



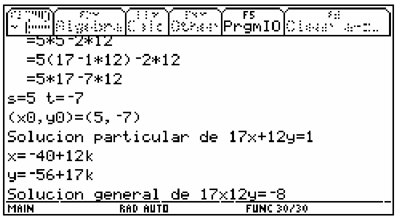


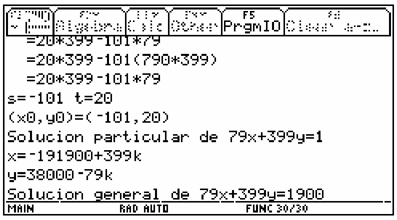
Afterwards, we can see the screen when the euclidean algorithm is developed ......

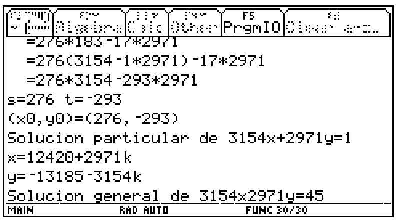
and finally we see at the screen the calculation of s and t and the particular and general solution.



See the solutions of the ACDC 7 problems using this program:







which is equivalent to 19+399k, 1–79k

Interesting WEB sites:

www.math.hawaii.edu/206L	Goodies and files from David Stenega´s calculus class
www.univie.ac.at/EMIS/	European Mathematical Information Service
www.chartwellyorke.com/	Philip Yorke´s web page with plenty of books
and as ever the **T³ site**:	www.ti.com/calc/docs/t3ww.htm

(all sites are still valid!)