THE DERIVE - NEWSLETTER #35

# THE BULLETIN OF THE DERIVE USER GROUP + TI 92

Contents:



revised version 2015 September 1999

**[1]** **Mathematik lernen und üben**, Computer-Algebra mit den Rechnern TI-89 und TI-92
Günter Scheu, Dümmler-Verlag, Dümmlerbuch 45151
Zusätzlich erhältlich: Diskette mit Lösungsvorschlägen, Aufgabenblättern, Skripten und Programmen, Dümmlerbuch 45162 und 45166

**[2]** **Projekte und Aufgaben zur Analytischen Geometrie**, (DERIVE, TI-89/92 Sek. II),
Benno Grabinger, Schroedel-Verlag ISBN 3-507-73224-6

**[3]** **Mathematiques concrètes, illustrées par la TI-92 et la TI-89,** J.M.Ferrard – H. Lemberg,
Springer France 1998, ISBN 2-287-59647-X

**[4]** **Explorations – Statistics Handbook for the TI-83,** Larry Morgan, Texas Instruments 1997
ISBN 1-886309-07-8

**[5]** **Neue Technologien – Neue Wege im Mathematikunterricht,** Ergebnis eines EU-COMENIUS Projekts.
**New Technologies – New Means of Mathematics Teaching**, Result of an EU-COMENIUS Project.
Barzel (GER), Böhm (AUT), Drijvers (NED), Janssens (BEL), Sjöstrand (SWE), Watkins (ENG), ublished by the Pedagogical Institute of Lower Austria 1999. (approx. 100 pages, 13 €)
Avaible at the Editor (write, call, FAX, email) or http://www.bk-teachware.com or http://www.ydsa.se (Scandinavia)

**[6]** **Computer Algebra Systems,** A Practical Guide**,** Michael Wester ed., Wiley 1999,
ISBN 0-471-98353-5.
A "Must" for a CAS-enthusiast. On 418 pages you can find a comparison of the most popular CAS – including DERIVE and have a look behind the curtain of how they do work. Experts like Al Rich, David Jeffrey, Wolfram Koepf , Frank Postel, Stanley Steinberg and others contributed articles like: "Let´s Do Some Analysis", Computing Limits in CAS", "CAS in Maths Education", "On Lovelace, Babbage and the Origins of Computer Algebra". Three appendices comprise Resources, Synonyms and CAS for various purposes.

## A Collection of interesting web sites:

| | |
|---|---|
| `http://www.mathematik-verstehen.de/` | Dörte Haftendorn´s site |
| `archives.math.utk.edu:80/topics/` | A wonderful collection |
| `www.ecmc.de/eenet` | European Experts´ Network for Educational Technology |
| `wmy2000.math.jussieu.fr/ma2000.htm` | World Mathematical Year 2000 |
| `www.bham.ac.uk/ctimath/gateway/packages.htm` | DERIVE links |
| `https://sites.google.com/site/franzklement/` | CAS-Links (in German only) |
| `www.schroedel.de` | Publisher, books and materials (German) |
| `home.eduhi.at/member/j.kliemann/ti83/index.htm` | Teaching Materials |
| `www.treasure-troves.com` | Eric Weisstein´s Treasure Box |

The red links are not valid any longer.

Dear friends,

there is so much to report, that I really don´t know where to start. This time I have to do without a German version and I hope that my many German speaking friends will forgive me.

The ACDCA-Summer Academy in Gösing, Lower Austria was a wonderful meeting of CAS users and –creators: David Stoutemyer and Theresa Shelby presented *DERIVE* Version 5, Frank Postel showed an interesting educational application of MuPAD and Danny Gremillion gave an impressive introduction to TI-Interactive. 65 delegates from 19 countries attended 35 lectures - a full program for three days, accomplished by a nostalgic train ride and a hike in the mountains.

The picture shows Josef together with Carl Leinbach at the "End of the Trail". One or two cool drinks helped them to recover.

Thank you all for coming and sharing your experience. The papers will be presented on ACDCA´s homepage very soon. We also plan to produce a CD with the submitted papers – depending on how many of the lecturers will deliver their contributions.

In this DNL you can find the "true Titibits 15". Unfortunately I repeated Johann Wiesenbauer´s Titbits #14 as #15 in the last DNL. Please study the new one carefully. It extends Richard Schorn´s trip into Group Theory from DNL#33. (By the way, Richard took many pictures in Gösing and he promised to send some to put them also on the Summer Academy CD). As a reparation for my mistake I added two pages more of content and apologize once more for my mistake.

Have you ever had the idea to calculate the "half derivative" of a function? David Halprin did. Follow his findings in the "Lighter Side of Operational Calculus". David also sent an amusing collection of "Epi-taphs/grams for Mathematicians of Desparate Fame". I hope to find some space in the next DNLs to present some pearls of his collection.

"The News" of this summer were doubtless TI´s take-over of *DERIVE*. Many messages full of fear, disappointment, but also full of hope and good ideas could be found in the news groups. I am happy to have the statements of two very competent people in the heart of this issue. Our good wishes for a prospering future accompanies the just married CAS-couple.

In the next DNL you will find an interesting astronomic article dealing with the "3 bodies´ problem", contributed by Ludwig Fritsch, Vienna. He uses *DERIVE* together with CYCLONE 98 to visualize this famous problem. Page 10 shows another CYCLONE 98 created object and announces its even more powerful successor.

I´d like to draw your attention to the outcome of an European Union supported COMENIUS project. The book shelf contains an English and a German version [5]. It is very likely that you will know some of the project group´s partners.

You can see that it was a very busy summer for the DERIVIANS and TI-ers. And I didn´t even mention the Plymouth Conference – I couldn´t attend this conference because we had to organize the Summer Academy, I didn´t mention the eclipse, I didn´t mention .......

Best regards until December,

Josef

The *DERIVE-NEWSLETTER* is the Bulletin of the *DERIVE & TI-92 User Group*. It is published at least four times a year with a contents of 44 pages minimum. The goals of the *DNL* are to enable the exchange of experiences made with *DERIVE* and the *TI-92/89* as well as to create a group to discuss the possibilities of new methodical and didactical manners in teaching mathematics.

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

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

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

Next issue: December 1999
Deadline 15 November 1999

**Preview: Contributions for the next issues**

3D-Geometry, Reichel, AUT
Graphic Integration, Linear Programming, Various Projections a.o., Böhm, AUT
A Utility file for complex dynamic systems, Lechner, AUT
Examples for Statistics, Roeloffs, NL
Quaternion Algebra, Sirota, RUS
Various Training Programs for the TI
A critical comment on the "Delayed Assignation" :==, Kümmel, GER
Sand Dunes, River Meander and Elastica, ....., Halprin, AUS
Type checking, Finite continued fractions, Winding number, Welke, GER
Share Holder´s Considerations using a CAS, Böhm, AUT
Kaprekar´s "Self numbers", GER
Implicit Multivalue Bivariate Function 3D Plots, Biryukov, RUS
LU-Decomposition, Morales, COL
ODEs with Constant Coefficients, Fernandez, ARG
Three Bodies Problem (DERIVE & Cyclone98), Fritsch, AUT
Calendar Problems (Poblacion), ESP

and
Setif, FRA; Vermeylen, BEL; Leinbach, USA; Speck, NZL; Biryukow, RUS
Wiesenbauer, AUT; Aue, GER; Koller, AUT, Ibrahim & Córdoba, ESP, .......

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

# A Substitution Function for Variables

by *Stefan Welke,* Spwelke@aol.com

**Abstract**

*DERIVE* has no built-in function to perform substitutions but offers the possibility to substitute variables and subexpressions via menu commands. This article describes a function that performs substitution of variables.

## Introduction

Consider the following problem: Given the equation of an algebraic curve γ, e.g. $y^2 - x = 0$, find an equation for γ in polar coordinates, i.e. $r := f(\tau)$. A handmade solution with *DERIVE* is: Enter the equation, use the SUBSTITUTION - command to substitute $x := r\,cos(t)$ and $y := r\,sin(t)$, then solve for *r*.

$$\#1: \quad y^2 - x$$

$$\#2: \quad (r\cdot SIN(t))^2 - r\cdot COS(t)$$

$$\#3: \quad SOLVE((r\cdot SIN(t))^2 - r\cdot COS(t),\ r) = \left[r = 0,\ r = \frac{COS(t)}{SIN(t)^2}\right]$$

Comment 2015: Later DERIVE versions present the output of SOLVE in another form. Additionally there is also the SOLUTIONS-function available:

$$SOLVE((r\cdot SIN(t))^2 - r\cdot COS(t),\ r) = \left(r = \frac{COS(t)}{SIN(t)^2} \vee r = 0\right)$$

$$SOLUTIONS((r\cdot SIN(t))^2 - r\cdot COS(t),\ r) = \left[0,\ \frac{COS(t)}{SIN(t)^2}\right]$$

This way is particulary satisfactory for educational purposes because the student is aware of each step of calculation. But at a more advanced level one would like to automate this procedure. A solution of this problem should be a function TO_POLAR(expr_) which returns an expression for $f(\tau)$. The aim of this article is to provide a function SUBST which works like the SUBSTITUTION – procedure for variables:

$$SUBST(y^2 - x,\ [x,\ y],\ [r\cdot COS(t),\ r\cdot SIN(t)]) = r^2\cdot SIN(t)^2 - r\cdot COS(t)$$

The syntax is: SUBST(expr_, var_list, substitution_list). In the expression denoted by expr_ each element of var_list is substituted by the corresponding element of substitution_list. The announced function TO_POLAR now reads:

```
TO_POLAR(expr_, var_) := RHS(SOLVE(SUBST(expr_, var_, [r·COS(t), r·SIN(t)]) = 0, r))
```

$$\text{TO\_POLAR}(y^2 - x, [x, y]) = \left[0, \frac{\text{COS}(t)}{\text{SIN}(t)^2}\right]$$

Comment: This is the TO_POLAR-function in DERIVE 6 syntax:

```
to_polar(expr_, var_) := SOLUTIONS(SUBST(expr_, var_, [r·COS(t), r·SIN(t)]), r)
```

$$\text{to\_polar}(y^2 - x, [x, y]) = \left(r = \frac{\text{COS}(t)}{\text{SIN}(t)^2} \vee r = 0\right)$$

## Simple Substitution

A list of expressions is evaluated from left to right. Thus the following simple construction works:

#4: $[x := y,\ y^2 - x,\ x := x] = [y,\ y^2 - y,\ x]$

Note that we must clear variable *x*, because otherwise *x* keeps its value *y* for all future calculations. The second element of the RHS in #4 is what we really want and therefore we try the next command:

#5: $[x := w^2,\ y^2 - x,\ x := x]_2 = y^2 - x$

The reason for the failure is that only the second element of the LHS list of #5 is evaluated. A way around is to use a self-defined function which does nothing else but □□□. This may look strange but it works because in user-defined functions the argument is evaluated first.

#6: $\text{PART}(v\_, n\_) := v\_{}_{n\_}$

#7: `SIMPLE_SUBST(expr_, var_, svar_) := PART([var_ := svar_, expr_, var_ := var_], 2)`

#8: $\text{SIMPLE\_SUBST}(a·x^3 + b,\ x,\ z) = a·z^3 + b$

This is exactly what we need as substitution of a single variable.

Comment: #4 to #8 do not work in DERIVE 6. Use the implemented SUBST-function instead:

$$\text{SUBST}(a·x^3 + b,\ x,\ z) = a·z^3 + b$$

## Substitution of several variables

We can use `SIMPLE_SUBST` repeatedly to substitute several variables in an expression. To do this we utilize the `ITERATE`-function. Consider the list `v_:=[k,expr_]` where `k` ∈ ℕ is a counter and expr_ is the expression on which we perform the substitution. We now iterate the following function *S* over the argument `v_`. For brevity `a_` denotes the variable list and `b_` the substitution list.

#9: $\text{S(v\_, a\_, b\_)} := \left[1 + \text{v\_}_1,\ \text{SIMPLE\_SUBST}\left(\text{v\_}_2,\ \text{a\_}_{\text{v\_}\downarrow 1},\ \text{b\_}_{\text{v\_}\downarrow 1}\right)\right]$

Notice that the counter increases by one when *S* is applied to `v_`. We observe the effect of repeated application of *S*:

#10: $\text{S}\left(\left[1,\ x^2 - y\cdot z\right],\ [x,\ y,\ z],\ \left[w^3,\ a^3,\ 5\right]\right) = \left[2,\ w^6 - y\cdot z\right]$

#11: $\text{S}\left(\left[2,\ w^6 - y\cdot z\right],\ [x,\ y,\ z],\ \left[w^3,\ a^3,\ 5\right]\right) = \left[3,\ w^6 - a^3\cdot z\right]$

#12: $\text{S}\left(\left[3,\ w^6 - a^3\cdot z\right],\ [x,\ y,\ z],\ \left[w^3,\ a^3,\ 5\right]\right) = \left[4,\ w^6 - 5\cdot a^3\right]$

The function S2 uses `ITERATE` to produce the final result. The number of iterations equals the number of variables to substitute. S3 removes the counter:

#13: `S2(expr_, a_, b_) := ITERATE(S(w_, a_, b_), w_, [1, expr_], DIMENSION(a_))`

#14: $\text{S2}\left(x^2 - y\cdot z,\ [x,\ y,\ z],\ \left[w^3,\ a^3,\ 5\right]\right) = \left[4,\ w^6 - 5\cdot a^3\right]$

#15: `S3(expr_, var_, svar_) := PART(S2(expr_, var_, svar_), 2)`

#16: $\text{S3}\left(x^2 - y\cdot z,\ [x,\ y,\ z],\ \left[w^3,\ a^3,\ 5\right]\right) = w^6 - 5\cdot a^3$

Comment: #10 to #16 do not work in DERIVE 6 – and it is notnecessary. Use the implemented SUBST-function instead:

$$\text{SUBST}\left(x^2 - y\cdot z,\ [x,\ y,\ z],\ \left[w^3,\ a^3,\ 5\right]\right) = w^6 - 5\cdot a^3$$

## Intermediate Variables

Let us substitute $x+ y$ for $x$ and $2 + x$ for $y$ in the product $p := x\,y$. The result should be

$$p' := (x + y)\ (2 + x).$$

But function S3 gives a different result:

#17: $\text{S3}(x\cdot y,\ [x,\ y],\ [x + y,\ 2 + x]) = 2\cdot(x + 1)\cdot(x + 2)$

The reason is that S3 works sequentially. First $x$ is substituted, then $y$ in both places:

$$x\,y \;\rightarrow\; (x+y)\,y \;\rightarrow\; (x+(x+2))\,(x+2) \;=\; 2\,(x+1)\,(x+2)$$

But only the second $y$ should be replaced by $x + 2$. We can avoid this dilemma by the introduction of as many intermediate variables as that number of variables is that we have to substitute:

#18: $\text{S3}(x\cdot y,\ [x,\ y,\ a,\ b],\ [a,\ b,\ x + y,\ 2 + x]) = (x + 2)\cdot(x + y)$

First the variables *x* and *y* are renamed. Doing so they cannot be confused with the variables *x* and *y* in the substituting expressions. Then the variables *a* and *b* are replaced by *x* + *y* resp. 2 + *x*.

If we want to perform this procedure within a function we must ensure that the intermediate variables are not accidentally used in that expression that is used for substitution. Since *DERIVE* does not have local variables a good way to prevent this difficulty is to use underscored variables □__, b__, etc.:

```
#19:  v_ := [a__, b__, c__, d__, e__, f__, g__, h__]
```

The functions `AUX_1` and `AUX_2` enlarge the given variable and substitution lists by the correct number of intermediate variables from the list `v_`:

```
      AUX_1(u_, v_) := APPEND(u_, VECTOR(v_ , i, 1, DIMENSION(u_)))
#20:                                       i

#21:  AUX_1([k, l, m, n], v_) = [k, l, m, n, a__, b__, c__, d__]

      AUX_2(u_, v_) := APPEND(VECTOR(u_ , i, 1, DIMENSION(v_)), v_)
#22:                                   i

#23:  AUX_2(v_, [a, b, c]) = [a__, b__, c__, a, b, c]
```

These examples illustrate the use of `AUX_1` and `AUX_2`. Now we are well prepared to define the function `SUBST` we were looking for:

```
#24:  SUBST(expr_, var_, svar_) := S3(expr_, AUX_1(var_, v_), AUX_2(v_, svar_))

#25:  SUBST(x·y, [x, y], [x + y, 2 + x]) = (x + 2)·(x + y)

                2                                       2        2
#26:  SUBST(y  - x, [x, y], [r·COS(t), r·SIN(t)]) = r ·SIN(t)  - r·COS(t)
```

#25 and #26 show the expected results. The number of variables in `v_` is a theoretical limitation to the application of SUBST. But `v_` can be enlarged to math practically every purpose. Note that the number of elements of `var_` must match the number of elements of `svar_`.

## Applications

Our first application is the function `TO_POLAR` which was mentioned in the introduction.

```
#27:  TO_POLAR(expr_, var_) := RHS(SOLVE(SUBST(expr_, var_, [r·COS(t), r·SIN(t)]) = 0, r))

                  2             ┌       COS(t) ┐
      TO_POLAR(y  - x, [x, y]) = │0,  ───────── │
#28:                            │            2 │
                                └     SIN(t)   ┘

                  2    2              ┌           1                         1          ┐
      TO_POLAR(x  - y  + 1, [x, y]) = │ ──────────────────── , - ──────────────────── │
#29:                                  │                  2                        2 │
                                      └ √(1 - 2·COS(t) )        √(1 - 2·COS(t) )     ┘
```

Comment: See the TO_POLAR-function in DERIVE 6 syntax on the next page:

$$\text{to\_polar}(x^2 - y^2 + 1, [x, y]) = \left( r = -\frac{\text{SIGN}(2\cdot\cos(t)^2 - 1)}{\sqrt{(1 - 2\cdot\cos(t)^2)}} \lor r = \frac{\text{SIGN}(2\cdot\cos(t)^2 - 1)}{\sqrt{(1 - 2\cdot\cos(t)^2)}} \right)$$

$$\text{to\_polar}(x^2 + y^2 - 1, [x, y]) = (r = -1 \lor r = 1)$$

$$\text{to\_polar}(x\cdot y - 1, [x, y]) = \left( r = -\frac{1}{\sqrt{(\sin(t))}\cdot\sqrt{(\cos(t))}} \lor r = \frac{1}{\sqrt{(\sin(t))}\cdot\sqrt{(\cos(t))}} \right)$$

A second application is the inversion of an algebraic curve with respect to a circle. Formally inversion with respect to the unit circle consists of a substitution of variables in the defining polynomial of the algebraic curve:

$$(x, y) \to \left( \frac{x}{x^2 + y^2}, \frac{y}{x^2 + y^2} \right)$$

More mathematical background is presented in [1] und [2]. Consider the parabola $y^2 - 2x = 0$ as an example:

$$\#30: \quad \text{SUBST}\left( y^2 - 2\cdot x, [x, y], \left[ \frac{x}{x^2 + y^2}, \frac{y}{x^2 + y^2} \right] \right) = -\frac{2\cdot x^3 + 2\cdot x\cdot y^2 - y^2}{(x^2 + y^2)^2}$$

The numerator of #30 is the defining polynomial of the image curve under inversion.

$$\#31: \quad \text{NUMERATOR}\left( -\frac{2\cdot x^3 + 2\cdot x\cdot y^2 - y^2}{(x^2 + y^2)^2} \right) = -2\cdot x^3 - y^2\cdot(2\cdot x - 1)$$

The following auxiliary function `AUX` computes the substitution list for the inversion with respect to a circle with radius *r* and center *m_*.

$$\#32: \quad \text{AUX}(\text{var\_}, \text{m\_}, \text{r\_}) := \text{m\_} + \frac{\text{r\_}^2\cdot(\text{var\_} - \text{m\_})}{(\text{var\_} - \text{m\_})\cdot(\text{var\_} - \text{m\_})}$$

$$\#33: \quad \text{AUX}([x, y], [-2, 1], 1)$$

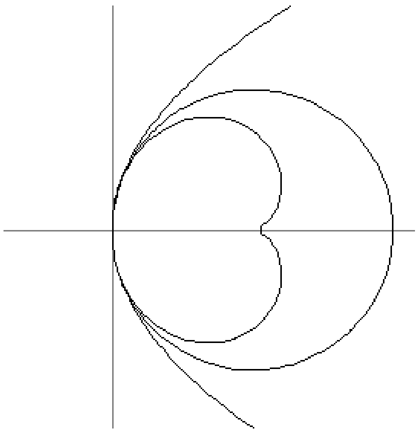
$$\#34: \quad \left[ \frac{x + 2}{x^2 + 4\cdot x + y^2 - 2\cdot y + 5} - 2, \frac{x^2 + 4\cdot x + y^2 - y + 4}{x^2 + 4\cdot x + y^2 - 2\cdot y + 5} \right]$$

We combine all steps in one function. Note that we can use the function value as substitution list. Again the reason is that `AUX` is evaluated before `SUBST`. The `FACTOR` function is necessary to guarantee that the result of `SUBST` has fractional form.

```
INVERSION(expr_, var_, m_, r_) := NUMERATOR(FACTOR(SUBST(expr_, var_, AUX(var_, m_, r_))))
```

$$\text{INVERSION}(y^2 - 2\cdot x, [x, y], [1, 0], 1)$$

$$-2\cdot x^4 + 6\cdot x^3 - 2\cdot x^2\cdot(2\cdot y^2 + 3) + 2\cdot x\cdot(3\cdot y^2 + 1) - y^2\cdot(2\cdot y^2 + 1)$$



**Remarks**: My efforts to define a substitution function were motivated by the second example. When writing [1] I missed the possibility to use substitution of variables as part of a user-defined function. *Mathematica* and *Maple* offer such a function and I wondered if it was possible to write a function for *DERIVE* with the polynomial of the image curve under inversion as the function value.

Further information on inversion of algebraic curves is accessible in [2].

## Conclusions

*DERIVE´s* strength lies in its restriction to a small set of really essential programming functions. This feature is particularly useful in education. But *DERIVE´s* programming functions are powerful enough to achieve a well working substitution function with only nine lines of programming code. So substitution of variables can become a part of user-defined functions. Two additional lines of programming code provide the function `INVERSION` which manipulates polynomials in a nontrivial way.

## References

[1] S. Welke, *Inversion of Elementary Algebraic Curves with Respect to a Circle*, Proceedings of the International DERIVE and TI-92 Conference, Schloß Birlinghoven, July 1996. ZKL Uni Münster

[2] Xah Lee, http://www.best.com/~xah/SpecialPlaneCurves_dir/Inversion_dir/inversion.html (with many interesting links)

Comment of the Editor: DERIVE 5 will provide a lot of additional programming tools. At the ACDCA Summer Academy David Stoutemyer and Theresa Shelby gave an exciting presentation of DERIVE 5 which will contain many new features.

Comment for the revised version 2015: As mentioned above we have now the SUBST-command implemented. Stefan's INVERSION-function works pretty well. I'd like to propose a small improvement. When plotting the inversion of the parabola given above you will receive an error message because you have to change to:

$$-2\cdot x^4 + 6\cdot x^3 - 2\cdot x^2\cdot(2\cdot y^2 + 3) + 2\cdot x\cdot(3\cdot y^2 + 1) - y^2\cdot(2\cdot y^2 + 1) = 0$$

My proposal is:

```
INVERSION(expr_, var_, m_, r_) := NUMERATOR(FACTOR(SUBST(expr_, var_,
           AUX(var_, m_, r_)))) = 0
```

Examples:

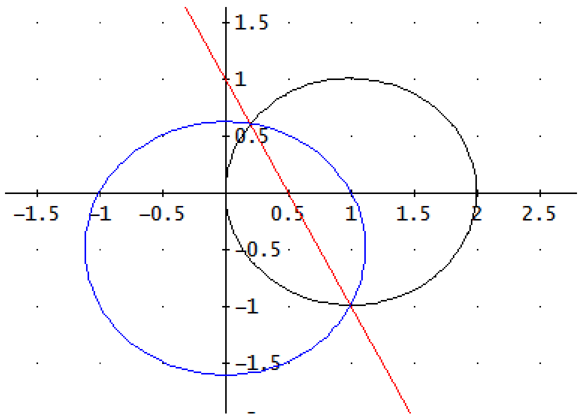
Inversion of a straight line wrt to a circle [(1,0);1] gives a circle passing the origin and vice versa and 2nd example is the inversion of an ellipse wrt to the circle [(1,0);2] and vice versa.

INVERSION(2·x + y − 1, [x, y], [1, 0], 1)

$x^2 + y^2 + y = 1$

INVERSION($x^2 + y^2 + y - 1$, [x, y], [1, 0], 1)

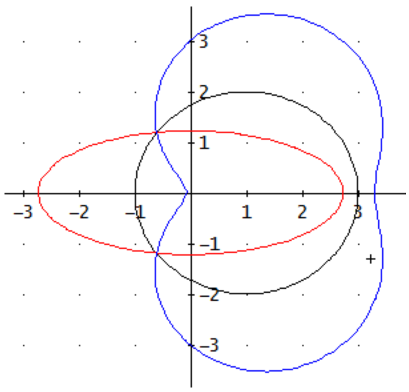
$2\cdot x + y = 1$



$(x - 1)^2 + y^2 = 4$

INVERSION($2\cdot x^2 + 10\cdot y^2 - 15$, [x, y], [1, 0], 2)

$13\cdot x^4 - 68\cdot x^3 + 2\cdot x^2\cdot(13\cdot y^2 + 47) - 4\cdot x\cdot(17\cdot y^2 + 9) + 13\cdot y^4 - 118\cdot y^2 = 3$



## Morris Waehlti´s "Ten Section Method"

In July I had a telephone talk with Morris who wanted to create a *DERIVE* – program to demonstrate an algorithm for finding a root of an equation numerically similar to the Bisection-Method, but using a "Ten Section". Some days after I had a FAX, which described accurately what he had in mind:

As per our telephone conversation, here, on paper, is what I was thinking about. To create a *DERIVE* "program" to find the square root of 5 (or any other number) by the TEN SECTION METHOD.

| NUMBER | NUMBER belongs to VECTOR | SINCE (test) |
|---|---|---|
| 1 | no | $1^2 < 5$ and $2^2 < 5$ |
| 2 | yes | $2^2 < 5$ and $3^2 > 5$ |
| 2.0 | no | $2.0^2 < 5$ and $2.1^2<5$ |
| 2.1 | no | $2.1^2 < 5$ and $2.2^2 < 5$ |
| 2.2 | yes | $2.2^2 < 5$ and $2.3^2 > 5$ |
| 2.20 | no | $2.20^2 < 5$ and $2.21^2 < 5$ |
| 2.21 | no | $2.21^2 < 5$ and $2.22^2 < 5$ |
| 2.22 | no | $2.22^2 < 5$ and $2.23^2 < 5$ |
| 2.23 | yes | $2.23^2 < 5$ and $2.24^2 > 5$ |
| 2.230 | no | |
| 2.231 | no | |
| .... and so on .... | | |

The vector in the PROCESS of GROWING would look like this:

**[2, 2.2, 2.23, 2.236, 2.2360, ....]**

It would STOP GROWING at some desired degree of precision.

If one wanted to find $\sqrt{59}$ , perhaps one could give the "program" 7 as an initial value to speed the process up a bit.

Finding $\sqrt{5}$ is like solving the equation $x^2 – 5 = 0$ for its positive root. With some modifications to the "program" could one use it to find a root of some less simple equation f(x) = 0 by the TEN SECTION METHOD, perhaps by giving it an initial value?? A number, say 3, would belong to the solution vector if f(3) < 0 but f(4) > 0 i.e.

| NUMBER | NUMBER belongs to VECTOR | SINCE (test) |
|---|---|---|
| 3 | yes | f(3) < 0 and f(4) > 0 |
| 3.0 | no | f(3) < 0 and f(3.1) < 0 |
| 3.1 | yes | f(3.1) < 0 and f(3.2) > 0 |
| 3.10 | no | f(3.10) < 0 and f(3.11) < 0 |
| 3.11 | yes | f(3.11) < 0 and f(3.12) > 0 |
| 3.110 | yes | f(3.110) < 0 and 3.111 > 0 |

The solution vector in the PROCESS of GROWING would look like this:

**[3, 3.1, 3.11, 3,110,......]**

It would STOP GROWING at some desired degree of precision.

I appreciate that the TEN SECTION METHOD is NOT a very efficient way to do anything. BUT I am regarding it more as an EXERCISE in programming with DERIVE. As a TEST, if you like, of what one CAN and CAN NOT do with DERIVE functional programming.

For instance, is it possible to have a DERIVE Program PAUSE and accept user input???
(Hi Al, Dave and Theresa, it is to you to answer this question!)

If you think that all this will turn out to be just a time consuming bother, PLEASE just forget about it.

Best regards, Morris Waehlti, Geneva, Switzerland.

I didn´t forget and tried my best, but I was not as successful as with the "Look and Say Sequence" (page 34):

```
                                                                  -k_
TENSEC(u, x, n) := VECTOR(MAX(SELECT(u < 0, x, VECTOR(0 + n_·10      , n_, 0,

    k_ + 1
  10       ))), k_, 0, n)

         2
TENSEC(x   - 5, x, 4)

[2, 2.2, 2.23, 2.236, 2.236]

in 1999 80 sec, now only 6 sec

         2
TENSEC(x   - 5, x, 5)

[2, 2.2, 2.23, 2.236, 2.236, 2.23606]

in 1999: I couldn't wait longer!, now: needs 133 sec
```

As Johann Wiesenbauer – my man for all problems connected with programming in DERIVE – was in Hawaii to work together with the Soft Warehousers to improve DERIVE 5 I wrote to one of his competitors in programming skills Terence Etchells and posted Morris´ TEN SECTION PROBLEM. Very soon I received the following file:

```
"Ten Section Method"
"Written by Terence Etchells, Liverpool John Moores University,UK"
"For Morris Waehlti and the DUG"
PrecisionDigits:=10
Notation:=Decimal
SIGNS(w):=IF(w=0,1,SIGN(w))
TEST_FOR_ROOT(f,x,a,b):=IF(SIGNS(ITERATE(f,x,a,1))=SIGNS(ITERATE(f,x,b,1)),
         0,1,0)
FIND_NEXT_SECTION(f,x,s_,i):=ITERATE(IF(TEST_FOR_ROOT(f,x,a,a+i)=1,a,a+i,
         a+i),a,s_)
TEN_SECTION(f,x,s_int,s_val,n):=ITERATES([i/10,
         FIND_NEXT_SECTION(f,x,s_,i)],[i,s_],[s_int,s_val],n)` SUB 2
```

f is the equation f(x) = 0; x is the variable; s_int is the initial interval size. s_val is the starting point; n is the number of times you wish to perform the decimal search,

e.g. $x^2 – 5=0$, starting at x = 1 with a starting interval of size 1, 10 times.

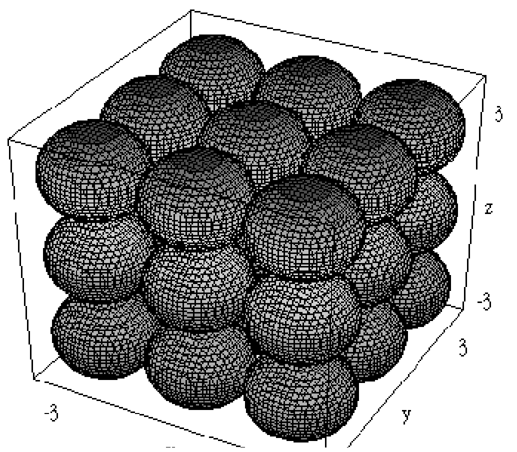
```
TEN_SECTION(x^2-5,x,1,1,10)
;Simp(#13)
[1,2,2.2,2.23,2.236,2.236,2.23606,2.236067,2.2360679,2.23606797,
 2.236067977]
```

N.b. you must use a high precision level if n is large!!!
E.g. $x^2 – 5=0$, starting at x = -20 with a starting interval of size 10, 10 times.

```
TEN_SECTION(x^2-5,x,10,-20,10)
;Simp(#17)
[-20,-10,-3,-2.3,-2.24,-2.237,-2.2361,-2.23607,-2.236068,-2.236068,
 -2.23606798]
```

The interval size is divided by 10 for each iteration; this can be changed by editing the `TEN_SECTION()` function where the `i/10` appears.

---




This nice object consisting of 27 spheres – it is really only **ONE** surface - was presented by David Sjöstrand at the ACDCA-Summer Academy 1999.

He used DERIVE to edit and simplify the expression below and imported this expression to Cyclone98 to plot this box full of spheres.

$$\prod_{c=-1}^{1} \prod_{b=-1}^{1} \prod_{a=-1}^{1} ((x - 2*a)^2 + (y - 2*b)^2 + (z - 2*c)^2 - 1) = 0$$

There is a wonderful new version of Cyclone99 – DPGraph2000 (see page 33):
http://www.ydsa.se/DPGraph2000/DPGraph2000/DPGraph2000.html

**Peter Witthinrich, Lübeck, Germany** Pwitthinrich@t-online.de

Dear Josef,

Alfonso Poblacion's algorithm shows another way to compute pi, but what is the basics, the theoretical background?
I didn't intend to show a high speed algorithm but the spigot algorithm found by Rabinowitz/Wagon. And Poblacion's algorithm is not a spigot algorithm, which produces the decimal places digits by digits (so the digits 'drop' one after the other).
Another important difference lies in the fact that the spigot algorithm uses only integer arithmetic, i.e. there are no rounding errors apart from integer overflow (which can be avoided by suitable routines or exception handling). So the spigot algorithm uses low level mathematics, but I think the PIAPPROX algorithm uses Derive' s internal floating point routines. They are good I know, they were already good in old muMath years ago. Anyway the TI-92 is much slower and lacks memory for computing large numbers.

My proposal: Alfonso Poblacion describes us the theoretical background of his mentioned algorithm. There are several algorithms computing pi: I would like to learn about another one.

Best wishes
Viele Grüße aus Lübeck

**Alfonso Poblacion, Valladolid, Spain** alfonso@gauss.mat.eup.uva.es

Dear Josef and Peter,

By the days I read DNL #33, I was working with other colleagues on the writing of a Derive Secondary level book. In this book we include several miscellaneous chapters combining maths, history and courious things. One of them is about squaring the circle and the calculation of PI. So I read some articles and as I found interesting (because of the speed) algorithms to compute PI, I sent you one of them for the DNL relating this with the appearance of the Spigot algorithm. But my aim was not to say something like "This is a better algoritn because is faster". I know both are different, so please Peter, forgive me if I caused you that bad impression, surely deriving from my poor English.

The algorithn (that Josef gently named PIPOB, but of course is not mine) was taken from an article in the Spanish version of Scientific American titled "Ramanujan and PI" that appeared in Spain in the issue of April 1988, whose authors were Jonathan M. Borwein and Peter B. Borwein. They explained on that article that they deduced the algorithm modifying another one from Gauss that was later modified by Richard Brent and Eugene Salamin (called Gauss-Brent-Salamin algorithm). After the Borwein & Borwein's changes, they said that was used by Yasumasa Kanada in 1983 to compute thousands of digits of PI (As you probably know, Kanada is an expert and recordman about calculating Pi digits).So the explanations I can give you are the ones from that article that also includes another two iterative algorithms.

They explained that these algorithms are based on modular equations. A modular function L(q) is one that can relate by an algebraical expression called modular equation involving the same function expressed by the same variable q raised up to an integer power L(q^p). The integer p determines the "order" of the modular equation. For example (I will write mathematical expressions like if they would be written with DERIVE, but do not try to translate them to DERIVE; I only use it like a word processor)

```
L(q)=16q PRODUCT(((1+q^(2n))/(1+q^(2n-1)))^8, n, 1, inf)
```

Its associated equation is of $7^{th}$ order and it relates L(q) with L(q^7):

```
(L(q)L(q^7))^(1/8)+((1-L(q))(1-L(q^7)))^(1/8)=1
```

The solutions of the modular equation that satisfy added conditions are called singular values. One of these values is deriving from a sequence of values K sub p (I don not know how to use this editor to write subindices), where

```
K sub p = sqrt(L(e^(-pi sqrt(p))))
```

These values has the property that -2/sqrt(p) log(K sub p /4) coincide with a lot of digits of PI. The number of common digits increases with p.

Ramanujan used this general procedure to build some series related with PI. On the algorithm described in the DNL, the first step (the calculation of y(n)) belongs to the calculation of one of the terms of a sequence of singular values that is obtaining when you solve a modular equation of a suitable order (the authors did not say which). The $2^{nd}$ step (calculation of a(n)) belongs to the logarithm of the singular value.

This is more or less what the authors say about the background of the iterative algorithms in the referred article. I hope this would be what you asked me for.

Have excellent holidays.
Alfonso J. Poblacion.

# DtL: A *DERIVE* to LaTeX-Translator

**Gary Bishop, bishop@cs.unc.edu**

**August 1999**

### 1 What is it?

DtL is a little program I wrote to address the problem of translating equations from Derive into the form required by LaTeX. Manually translating anything more than trivial expressions is error prone and frustrating. With DtL I can work out an equation in Derive and immediately paste it into my LaTeX document to get publication quality formatting.

### 2 How do I use it?

In Derive for Windows, select an expression or sub-expression and copy it to the clipboard with `Copy Expressions' from the `Edit' menu or by holding the Control key and typing C. Next run DtL.exe; a shortcut on the desktop, or a `shortcut key' is great for this. DtL will replace the Derive expression on the clipboard with the equivalent input for LaTeX. Finally, switch to the editor you use for entering LaTeX and paste the result.
You can also use DtL with the ` .mth' files produced by Derive. Run DtL with the command line: DtL.exe -f -F < myfile.mth > myfile.tex to convert all the expressions in myfile.mth into LaTeX format.

### 3 How does it work?

When you select an equation in Derive for Windows, the text of the equation is placed on the clipboard in two different formats. The format you have access to in most programs is formatted text using the DfW printer font. You can paste this form into a document to get a representation just like you saw in the Derive window but the quality of the result is well below what you get with a serious typesetter like LaTeX.
The other format, `DfW Expression' is the one operated on by DtL. When you run DtL it reads the `DfW Expression' from the clipboard and translates it into an equivalent LaTeX representation. It then exits leaving the input for LaTeX on the clipboard. Then you can paste this text into whatever text editor you use. I prefer Emacs with the Auctex package but any editor that supports pasting text from other applications should work.

I used flex and bison from the gnu family of compiler building tools to do the lexical scanning and parsing of the input. These are the free software equivalents of lex and yacc from Unix. The remainder of the code is pretty straight-forward C++. It would be easy to modify DtL to produce input for TeX or Matlab.

### 4 How do I get DtL?

Although I wrote DtL to meet my own needs, I have made the source and executable available at http://www.cs.unc.edu/~gb/DtL.html. You can download either a zip file or a tar.gz file. Download the format that you prefer, unpack it with a program like WinZip or pkunzip and put the executable DtL.exe somewhere on your path. I put a shortcut on my desktop so that I can easily invoke DtL whenever I need it.

### 5 Where can I get the other software mentioned here?

Emacs, the world's greatest and most programmable editor, can be downloaded from http://www.cs.washington.edu/homes/voelker/ntemacs.html. The auctex package customizes emacs for editing TeX and LaTeX; you can download it from http://sunsite.auc.dk/auctex/. The MikTeX package provides the LaTeX processor and other programs for previewing and printing; you can download it from http://www.snafu.de/~cschenk/miktex/index.html. All of the gnu tools I used were part of the cygwin package of Unix tools for Win32 available from http://sourceware.cygnus.com.

### 6 Who supports DtL?

*Not me*. If you find bugs in DtL, please send me email. I am likely to fix it but I won't promise to fix any problem. Feel free to fix it yourself and let me know what you changed; that is why I included the source. As I said earlier, extending or modifying DtL to change the output style or language should be easy for anyone who knows how to use the tools. If you have never used flex (lex) or bison (yacc), I highly recommend you learn to use them. These powerful tools make translators such as DtL a snap to program.

---

File translated from TeX by TTH, version 2.30.
On 03 Aug 1999, 15:36.

**Gerhard Bardach** g.bardach@gmx.at

Does someone know a function in Derive to create a chain of generalized eigenvectors?

I need it for the creation of invariant subspaces for a matrix.

**Sebastiano Cappuccio, Italy** p.cappuccio@fo.nettuno.it

Dear Joseph,

please, see the URL:

http://people.ciram.unibo.it/~barozzi/Net_Elenco_schede.html

Here everybody can download a whole book of exercises of Calculus 1 using Derive in PDF format.

Unfortunately files are in Italian only.

I look forward to meeting you in Goesing next August.

Best regards

Sebastiano

The URL given above is valid. See also http://people.ciram.unibo.it/~barozzi/MI2/PDF/

You are friendly invited to visit the various home pages of T$^3$-Europe institutions. They provide a lot of CAS-materials and informations and you can also find many interesting links to related sites. (all from 2015)

**United Kingdom and Ireland**
http://www.tcubed.org.uk/home/

**Finland:**
https://education.ti.com/fi/suomi/nonproductsingle/opettaja_koulutus

**Sweden:**
https://education.ti.com/sv/sverige/larare/larar_fortbildning/kostnadsfria-larartraffar

**Germany:**
http://www.t3deutschland.de/

**Switzerland (German & French)**
http://www.t3schweiz.ch/
http://www.t3suisse.ch/

**T3 Worldwide**
**education.ti.com/en/us/pd/community/t3-ww**

**Austria**
http://www.t3oesterreich.at/

**France:**
https://education.ti.com/fr/france/formations/t3france_home

**Portugal:**
https://education.ti.com/pt/portugal/professor/professor
http://www.apm.pt/apm/t3/t3.htm

**Italy:**
https://education.ti.com/it/italia/professori/aggiornamento

**United States and Canada:**
https://education.ti.com/en/us/pd

**Belgium (Flemish and Walloon):**
http://www.t3vlaanderen.be/
http://www.t3wallonie.be/

**Holland**
http://www.t3nederland.nl/home/

**Denmark**
https://education.ti.com/da/danmark/nonproductsingle/laerere_t3

**Norway**
https://education.ti.com/no/norge/nonproductsingle/lt3

**Australia and New Zealand**
https://education.ti.com/en-GB/aus-nz/pd/online-learning

Dear Joseph,

Hello again from autumnal heading-into-winter Australia. Firstly I want to thank you for the first half of my paper on Glove Osculants in DNL # 29, received last month.

Today I am sending you a paper on fractional derivatives using the original idea of operational calculation by Oliver Heaviside, but developed by me with the auspices of Derive XM and its excellent plotting facilities as well as its calculation of the semi-derivatives of a Taylor series. I have developed these semi-derivatives as though they art new functions and have attempted with a little success to relate them to one another with some identities. Perhaps there is more to be found, yet untapped.

Herzlichst, Yours sincerely,
David Halprin

# THE LIGHTER SIDE OF OPERATIONAL CALCULUS; OLE!

David Halprin, North Balwyn, Australia, email: davidlaz@net2000.com.au

Let $D = \frac{d}{dx}$ and therefore $D^p = \frac{d^p}{dx^p}$ 1is the linear functional differential operator in the simplified operational calculus to follow.

$$D\cdot x^r = \frac{d(x^r)}{dx} = r\cdot x^{r-1} = \frac{r!}{(r-1)!}\cdot x^{r-1}$$

$$D^2\cdot x^r = \frac{d^2(x^r)}{dx^2} = r(r-1).x^{r-2} = \frac{r!}{(r-2)!}\cdot x^{r-2}$$

Hence

$$D^{1/2}\cdot x^r = \frac{d^{1/2}(x^r)}{dx^{1/2}} = \frac{r!}{(r-1/2)!}\cdot x^{r-1/2} = F_x$$

$$D^{1/2}\cdot F_x = D^{1/2}\cdot D^{1/2}\cdot x^r = \frac{r!}{(r-1/2)!}\cdot\frac{(r-1/2)!}{(r-1)!}\cdot x^{r-1} = \frac{r!}{(r-1)!}\cdot x^{r-1}$$

Q.E.D.

Now we can advance to the trigonometric ratios, starting with sin x and cos x by expressing them as a sum to n terms where the rth term is all that is needed on which to derive the operation and choose as many terms as one requires, even though it is an infinite series. I shall choose the first 11 terms mostly in this exercise, but occasionally more will be needed for a better graphic representation of the series in question.

$$\sin\ x = \sum_{r=0}^{n}\frac{(-1)^r\cdot x^{2r+1}}{(2r+1)!}$$

$$D^{1/2}.\sin x = \sum_{r=0}^{n}\frac{(-1)^r\cdot x^{(2r+1/2)}}{(2r+1/2)!} = \sqrt{x}.\left(\frac{1}{(1/2)!} - \frac{x^2}{(5/2)!} + \frac{x^4}{(7/2)!} - \frac{x^6}{(13/2)!} + .. - ..\right)$$

$$D^{1/2}\cdot D^{1/2}.\sin x = \sum_{r=0}^{n}\frac{(-1)^r\cdot x^{2r}}{(2r)!} = \cos x$$

$$\cos x = \sum_{r=0}^{n}\frac{(-1)^r\cdot x^{2r}}{(2r)!}$$

$$D^{½}.\cos x = \sum_{r=1}^{n} \frac{(-1)^r . x^{2r-½}}{(2r - ½)!} = - \sqrt{x}.\left( \frac{x}{(3/2)!} - \frac{x^3}{(7/2)!} + \frac{x^5}{(11/2)!} - \frac{x^7}{(15/2)!} + .. - .. \right)$$

$$D^{½}.D^{½}.\cos x = \sum_{r=1}^{n} \frac{(-1)^r . x^{2r-1}}{(2r - 1)!} = - \sin x$$

$$\tan x = \sum_{r=1}^{n} \frac{-(-1)^n . 2^{2n} \left(2^{2n} - 1\right). B_{2n} . x^{2n-1}}{(2n)!}$$

$$D^{½}.\tan x = \sum_{r=1}^{n} \frac{-(-1)^n . 2^{2n} \left(2^{2n} - 1\right). B_{2n} . x^{2n-1.5}}{2n.(2n - 1.5)!}$$

Where B2n is equal to the Derive XM definition of BERNOULLI(2n) in the file NUMBER.MTH and is the nth term in some tables of Bernoulli numbers.

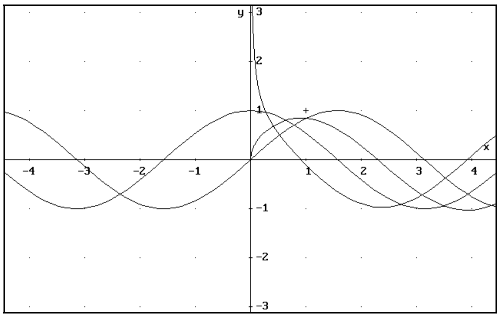
$$D^{½}.\sin x = \text{sun } x \ , \ D^{½}.\cos x = \text{cus } x \ , \ D^{½}.\tan x = \text{tun } x$$

$$D^{½}.\text{cosec } x = \text{cosuc } x \ , \ D^{½}.\sec x = \text{suc } x \ , \ D^{½}.\text{cotan } x = \text{cotun } x$$

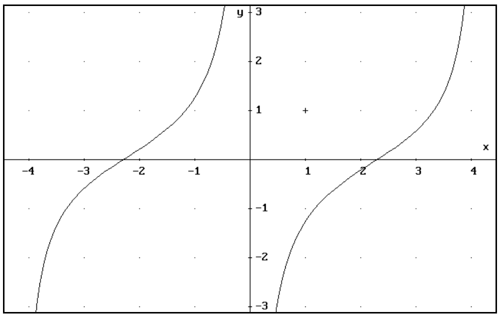
$$\frac{1}{\text{sun } x} = \text{cosuc}_2 \ x \ , \ \frac{1}{\text{cus } x} = \text{suc}_2 \ x \ , \ \frac{\text{sun } x}{\text{cus } x} = \text{tun}_2 \ x$$

$$\frac{1}{\text{sunh } x} = \text{cosuch}_2 \ x \ , \ \frac{1}{\text{cush } x} = \text{such}_2 \ x \ , \ \frac{\text{sunh } x}{\text{cush } x} = \text{tunh}_2 \ x$$

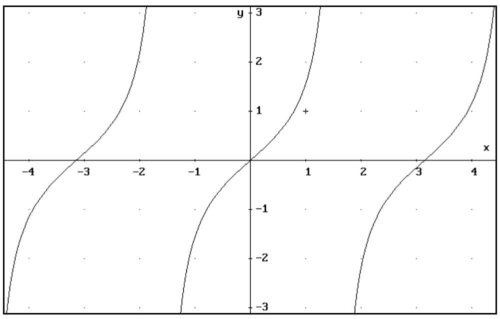
$$D^{½}.\ln(1 + x) = \text{lun}(1 + x)$$



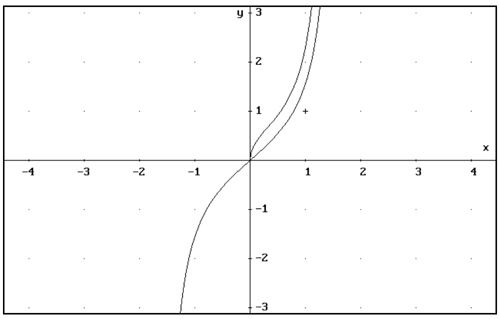
SIN x, SUN x, COS x & CUS x



SUN x / CUS x (Taylor expansions)

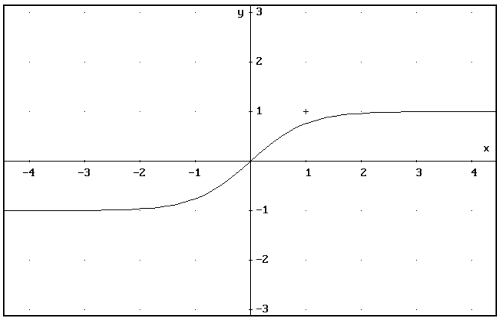


TAN x

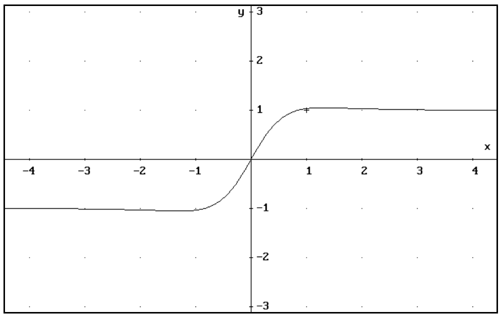


TAN x and TUN x

This shows that the second semi-derivative of sin x converts it to cos x, while the second semi-derivative of cos x looks different from what we might expect, so we substitute (r+1) for r and we have the correct expression, save for the exponent of `-1' effecting a change in its sign, as expected.




TANH x = SINH x / COSH x



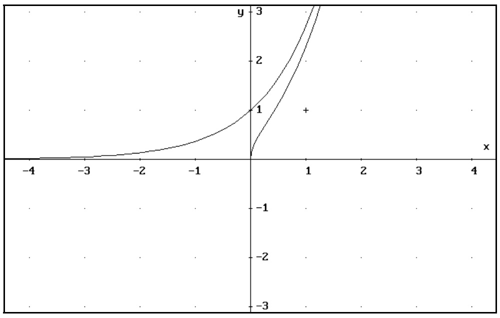

SUNH x / CUSH x

Now to the exponential series:

$$e^x = \sum_{r=0}^{n} \frac{x^r}{r!}$$

$$D^{\frac{1}{2}} . e^x = \sum_{r=0}^{n} \frac{x^{r-\frac{1}{2}}}{(r - \frac{1}{2})!} = E_{(x)}$$

$$D^{\frac{1}{2}} .E_{(x)} = D^{\frac{1}{2}} . D^{\frac{1}{2}} . e^x = \sum_{r=0}^{n} \frac{x^{r-1}}{(r-1)!} = e^x \quad \text{Q.E.D.}$$




$e^x$ and $E_{(x)}$

Naturally the hyperbolic functions follow from the exponential series:

$$\sinh x = \sum_{r=0}^{n} \frac{x^{2r+1}}{(2r+1)!}$$

$$D^{\frac{1}{2}} . \sinh x = \sum_{r=0}^{n} \frac{x^{(2r+\frac{1}{2})}}{(2r+\frac{1}{2})!} = \sqrt{x} . \left( \frac{1}{(1/2)!} + \frac{x^2}{(5/2)!} + \frac{x^4}{(9/2)!} + \frac{x^6}{(13/2)!} + ...... \right)$$

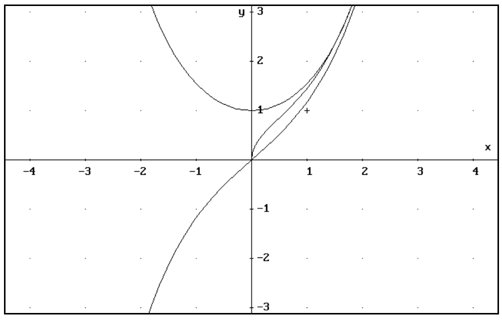
$$D^{\frac{1}{2}} . D^{\frac{1}{2}} . \sinh x = \sum_{r=0}^{n} \frac{x^{2r}}{(2r)!} = \cosh x$$

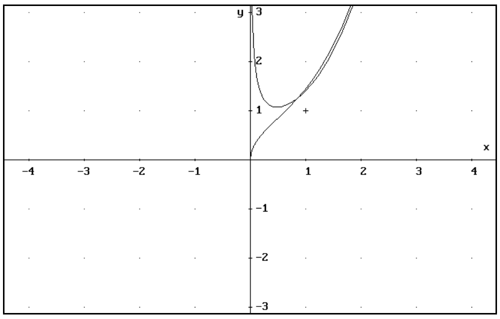
$$\cosh x = \sum_{r=0}^{n} \frac{x^{2r}}{(2r)!}$$

$$D^{\frac{1}{2}} . \cosh x = \sum_{r=1}^{n} \frac{x^{2r-\frac{1}{2}}}{(2r-\frac{1}{2})!} = \sqrt{x} . \left( \frac{x}{(3/2)!} + \frac{x^3}{(7/2)!} + \frac{x^5}{(11/2)!} + \frac{x^7}{(15/2)!} + ...... \right)$$

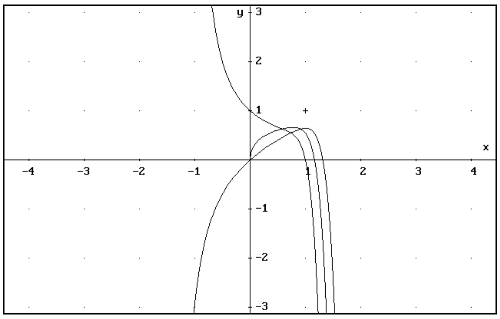
$$D^{\frac{1}{2}} . D^{\frac{1}{2}} . \cosh x = \sum_{r=1}^{n} \frac{x^{2r-1}}{(2r-1)!} = \sinh x$$

SINH x, SUNH x COSH x





SUNH x and CUSH x



LN(1 + x), LUN(1 + x) and 1/(1+x)

Now to a logarithmic series:

$$\ln(1+x)=\sum_{r=1}^{n}\frac{-(-1)^r.x^r}{r}$$

$$D^{\frac{1}{2}}.\ln(1+x)=\sum_{r=1}^{n}\frac{-(-1)^r.r!.x^{r-\frac{1}{2}}}{r.(r-\frac{1}{2})!}$$

$$D^{\frac{1}{2}}.D^{\frac{1}{2}}.\ln(1+x)=\sum_{r=1}^{n}\frac{-(-1)^r.r!.x^{r-1}}{r.(r-1)!}=\sum_{r=1}^{n}-(-1)^r.x^{r-1}=\frac{1}{1+x}\quad \text{Q.E.D.}$$

Now to some tests for distributivity, associativity and commutativity:

$$D^{\frac{1}{2}}.x^r=\frac{r!.x^{r-\frac{1}{2}}}{(r-\frac{1}{2})!}=F_x$$

$$D.F_x=\frac{(r-\frac{1}{2}).r!.x^{r-1.5}}{(r-\frac{1}{2})!}=\frac{r!.x^{r-1.5}}{(r-1.5)!}$$

$$D.x^r=r.x^{r-1}$$

$$D^{\frac{1}{2}}.(D.x^r)=\frac{r.(r-1)!.x^{r-1.5}}{(r-1.5)!}=\frac{r!.x^{r-1.5}}{(r-1.5)!}$$

$$D^{1.5}.x^r=\frac{r!.x^{r-1.5}}{(r-1.5)!}$$

$$D^{\frac{1}{2}}.D=D.D^{\frac{1}{2}}=D^{1.5}$$

e.g. $D^{1.5}.\sin x=D^{\frac{1}{2}}.\cos x$ etc.

Now to try to find a chain rule for a product of two functions:

$$x^m.x^n=x^{m+n}$$

$$D.(x^m.x^n)=mx^{m-1}.x^n+x^m.nx^{n-1}=x^{m-1}.x^{n-1}(mx+nx)$$

$$=x^{m+n-2}(m+n).x=(m+n).x^{m+n-1}\quad \text{Q.E.D.}$$

$$D^{\frac{1}{2}}.x^{m+n}=\frac{(m+n)!x^{m+n-\frac{1}{2}}}{(m+n-\frac{1}{2})!}$$

$$x^n.D^{\frac{1}{2}}.x^m+x^m.D^{\frac{1}{2}}.x^n=\frac{x^n.m!.x^{m-\frac{1}{2}}}{(m-\frac{1}{2})!}+\frac{x^m.n!.x^{n-\frac{1}{2}}!}{(n-\frac{1}{2})!}$$

$$=x^{m+n-\frac{1}{2}}.\left[\frac{m!}{(m-\frac{1}{2})!}+\frac{n!}{(n-\frac{1}{2})!}\right]\neq D^{\frac{1}{2}}.x^{m+n}$$

This shows, not unexpectedly, that the usual product rule does NOT apply for semi-derivatives, since their is no semblance of commutativity with the variable upon which it operates, and therefore it is not a general property of a linear operator. In other words both the D operators, (whether with integral or fractional exponents), are associative, commutative and distributive with each other preceding the function, but `in toto' they have left distributivity when associated with a variable and are non-commutative for multiplication with the variable. The essential difference between those operators with fractional exponents and those with integral exponents is exemplified when applied to a product of 2 or more variables, where we can only differentiate the product by the usual rules for a product when the exponent is an integer.

Let p and q be integers:

$$D^p(u+v) = D^p u + D^p v \ , \ D^p(u.v) = u.D^p v + v.D^p u$$

$$D^{\frac{p}{q}}(u+v) = D^{\frac{p}{q}} u + D^{\frac{p}{q}} v \ , \ D^{\frac{p}{q}}(u.v) \neq u.\left(D^{\frac{p}{q}} v\right) + v.\left(D^{\frac{p}{q}} v\right) 2$$

$$\left( D^{1/2}.e^x \right)^2 = \left( D^{1/2}.e^x \right)\left( D^{1/2}.e^x \right) = E^2_{(x)} \neq D.e^{2x}$$

Now to some consequences of the associative properties of the D operator and some newly defined and named functions and their inter-relationships.

Let $D^{1/2}.\sin x = \text{sun } x \ , \ D^{1/2}.\cos x = \text{cus } x \ , \ \text{tun}_2 \, x = \dfrac{\text{sun } x}{\text{cus } x}$

Let $\text{cosuc}_2 \, x = \dfrac{1}{\text{sun } x} \ , \ \text{suc}_2 \, x = \dfrac{1}{\text{cus } x} \ , \ \text{cotun}_2 \, x = \dfrac{\text{cus } x}{\text{sun } x}$

$$D.\text{sun } x = D.D^{1/2}.\sin x = D^{1/2}.D.\sin x = D^{1/2}.\cos x = \text{cus } x$$

$$D.\text{cus } x = D.D^{1/2}.\cos x = D^{1/2}.D.\cos x = - D^{1/2}.\sin x = - \text{sun } x$$

$$D.\text{tun}_2 \, x = \frac{\text{cus}^2 x + \text{sun}^2 x}{\text{cus}^2 x} = 1 + \text{tun}_2{}^2 \, x \neq \text{suc}_2{}^2 \, x$$

$$D.\text{cosuc}_2 \, x = \frac{- \text{cus } x}{\text{sun}^2 x} = - \text{cosuc}_2 \, x . \text{cotun}_2 \, x$$

$$D.\text{suc}_2 \, x = \frac{\text{sun } x}{\text{cus}^2 x} = \text{suc}_2 \, x . \text{tun}_2 \, x$$

$$D.\text{cotun}_2 \, x = \frac{\text{sun}^2 x + \text{cus}^2 x}{- \text{sun}^2 x} = - (1 + \text{cotun}_2{}^2 \, x) \neq - \text{cosuc}_2{}^2 \, x$$

Let $D^{1/2}.\sinh x = \text{sunh } x \ , \ D^{1/2}.\cosh x = \text{cush } x$

$$E_{(x)} = \sqrt{x}.\left( \frac{1}{(1/2)!} + \frac{x}{(3/2)!} + \frac{x^2}{(5/2)!} + \frac{x^3}{(7/2)!} + ....... \right)$$

$$\therefore \ E_{(x)} = (\text{sunh } x + \text{cush } x)$$

$$E_{(-x)} = \sqrt{x}.\left( \frac{1}{(1/2)!} - \frac{x}{(3/2)!} + \frac{x^2}{(5/2)!} - \frac{x^3}{(7/2)!} + ..-.. \right)$$

$$\therefore \ E_{(-x)} = \text{sunh } x - \text{cush } x$$

$$\therefore \ \text{sunh } x = \tfrac{1}{2}.(E_{(x)} + E_{(-x)}) \ , \ \text{cush } x = \tfrac{1}{2}.(E_{(x)} - E_{(-x)})$$

$$\therefore \ \text{cush}^2 x + \text{sunh}^2 x = \tfrac{1}{2}.(E^2_{(x)} + E^2_{(-x)}) \ , \ \text{cush}^2 x - \text{sunh}^2 x = - E_{(x)}.E_{(-x)}$$

$$E_{(ix)} = \sqrt{x}\ \ \sqrt{i}.\left(\frac{1}{(1/2)!} + \frac{i\,x}{(3/2)!} - \frac{x^2}{(5/2)!} - \frac{i\,x^3}{(7/2)!} + .. + .. - .. - ..\right)$$

$$\therefore\ E_{(ix)} = \sqrt{i}.(\text{sun } x - i.\text{cus } x)$$

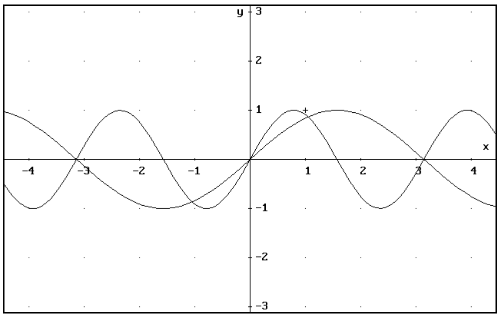
$$E_{(-ix)} = \sqrt{x}\ \ \sqrt{i}.\left(\frac{i}{(1/2)!} + \frac{x}{(3/2)!} - \frac{i\,x^2}{(5/2)!} - \frac{x^3}{(7/2)!} + .. + .. - .. - ..\right)$$

$$\therefore\ E_{(-ix)} = \sqrt{i}.(i.\text{sun } x - \text{cus } x)$$

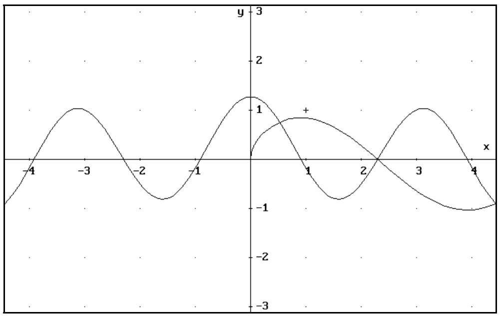
$$\therefore\ i.E_{(ix)} + E_{(-ix)} = 2i\sqrt{i}.\text{sun } x$$

$$i.E_{(ix)} - E_{(-ix)} = 2\sqrt{i}.\text{cus } x$$

$$\therefore\ \text{cus}^2 x + \text{sun}^2 x = -\ E_{(ix)}.E_{(-ix)}\ ,\ \text{cus}^2 x - \text{sun}^2 x = \tfrac{1}{2}i\ \left(E^2_{(ix)} - E^2_{(-ix)}\right)$$



SIN x and SIN 2x



SUN x and 2 SUN x CUS x

$$D.e^{mx} = \frac{d\,(e^{mx})}{d\,x} = m.e^{mx}$$

$$D^2.e^{mx} = \frac{d^2\,(e^{mx})}{d\,x^2} = m^2.e^{mx}$$

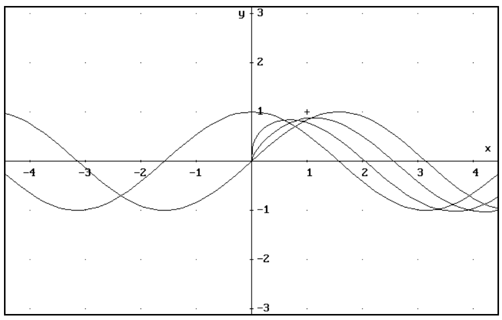
Hence

$$D^{\frac{1}{2}}.e^{mx} = m^{\frac{1}{2}}.\sum_{r=1}^{n}\frac{(mx)^{r-\frac{1}{2}}}{(r-\frac{1}{2})!}$$

$$D^{\frac{1}{2}}.D^{\frac{1}{2}}.e^{mx} = m.\sum_{r=1}^{n}\frac{(mx)^{r-1}}{(r-1)!} = m.e^{mx}\qquad \text{Q.E.D.}$$

$$2\,\text{sun } x.\text{cus } x = \tfrac{1}{2}\ \left(E^2_{(ix)} + E^2_{(-ix)}\right)$$

Now to demonstrate graphically one-third derivatives.



SIN x, D^(1/3) SIN x, D^(2/3) SIN x and COS x

See a short section of David´s HEAVISID.MTH

#1: File HEAVISID.MTH, (c) 1998 by David Halprin, PO Box 460, Nth. Balwyn

#2: Melbourne, Victoria, Australia. A file to investigate fractional derivatives.

#3: Notation := Rational

#4: $$\sum_{n=0}^{10} \frac{(-1)^n \cdot x^{2\cdot n + 1}}{(2\cdot n + 1)!}$$

#5: Simplify to receive the series for sin x

#6: $$\sum_{n=0}^{10} \frac{(-1)^n \cdot x^{2\cdot n}}{(2\cdot n)!}$$

#7: Simplify to receive the series for cos x

#8: $$\frac{n!}{\left(n - \frac{1}{2}\right)!} \cdot x^{n - 1/2}$$

#9: This is the operator to obtain the semi-derivative of x^n.

#10: Now to apply it to the general terms in the trig series.

#11: $$\sum_{n=0}^{10} \frac{(-1)^n \cdot x^{2\cdot n + 1/2}}{\left(2\cdot n + \frac{1}{2}\right)!}$$

#12: Simplify to obtain the semi-derivative of sin x = SUN x

#13: $$\sum_{n=0}^{10} \frac{x^{2\cdot n + 1}}{(2\cdot n + 1)!}$$

#14: This is the Taylor series for sinh x to 11 terms.

#15: $$\sum_{n=0}^{10} \frac{x^{2\cdot n + 1/2}}{\left(2\cdot n + \frac{1}{2}\right)!}$$

#16: Simplify to obtain the semi-derivative of sinh x = SUNH x

**BIBLIOGRAPHY**

Heaviside, Oli. Proceedings, Royal Society of London,
Pt.I Feb 2 1893 Pp 504/529, Pt.II June 15 1893 Pp 105/143

Poritsky, H. American Mathematical Monthly June 1936 Pp 331/344

Peacock, G. British Association Report 1833 Pp 211/221 & 241/249

**David Halprin, P.O.Box 460**
**North Balwyn, Victoria 3104, Australia**

It was a nice day in August. I opened my electronic mailbox and found:

```
On Tue, 10 Aug 1999, Terence Etchells wrote:

Hello Fellow Derivers
I strongly suggest that you go to the site
http://www.ti.com/calc/docs/news/cg-504.htm
for some rather Earth shattering news.
```

## Texas Instruments Acquires DERIVE

Many angry, sad, friendly, disappointed, satisfied and expectant messages followed. Let´s read the opinion of two important and competent people from the two companies:

To all DERIVE Users:

On 6 August 1999 Soft Warehouse, Inc. and all its assets, including DERIVE, were acquired by Texas Instruments Incorporated. There has been much concern expressed on this NewsGroup and elsewhere about the impact this will have on the future of DERIVE. As one of the principal authors of the system, I would like to respond to these concerns. The views expressed are my own, not TI's. However, note that I am working with TI as an independent consultant.

At the risk of over anthropomorphizing DERIVE, I make the following metaphor: More than one person has told me that DERIVE has many of the traits of a woman (intuitive, persistent, eager to help, etc.). I have been working on DERIVE and her predecessor muMATH for 20+ years now. And like any 20 year old, she is ready to go out into the world on her own. Like any father would be, I am proud and elated that she has come of age, but also deeply concerned about her future.

The following facts have convinced me that passing the DERIVE torch to TI is the right thing to do: Soft Warehouse, mostly through the efforts of my partner David Stoutemyer, has had a close working relationship with the Educational and Productivity Solutions (E&PS) division (formerly the Calculator division) of TI for more than seven years.

During that time E&PS has successfully developed and marketed products (the TI-92 and TI-89 calculators) based in large part on computer algebra (CA) software developed in cooperation with Soft Warehouse.

The investment TI made to acquire Soft Warehouse provides clear evidence that TI appreciates the value and potential of CA in general and DERIVE in particular.

TI faces strong competition in the graphing calculator market. To remain competitive in this market, TI will need to provide math and science educators with an integrated set of math tools (both hardware and software) available on a broad range of platforms.

Writing and debugging completely new software for each product would be prohibitively expensive, difficult to support, and not provide the required integration between products. Instead, it makes much more sense to write an interface specially designed for each product, but to base all the products on a single math engine. TI appears to want to base its products on the math engine that powers DERIVE.

Theresa Shelby designed (in collaboration with me) and implemented the DERIVE for Windows interface (both version 4 and the forthcoming version 5). Theresa and David Stoutemyer are now employees of TI. I am a consultant working with TI. Presumably we were hired by TI for our expertise in developing compact, efficient CA systems, as well as for our experience in marketing mathematical software over the last 20+ years. Therefore, we have a unique opportunity to significantly influence the direction E&PS takes in the future.

Our agreement with TI makes it clear that the top priority for Theresa and me is to complete, thoroughly test, and release version 5 of DERIVE for Windows as soon as possible. Thereafter I will continue to work on the math engine that powers DERIVE.

Freed from the responsibilities of running a company, I can devote more of my professional time and creative efforts to extending the mathematical and programming powers of DERIVE.

TI has the resources, personnel, and know-how required to effectively promote DERIVE in the current highly competitive global software market in a way that Soft Warehouse could never hope to do.

In addition to its own considerable programming talent, TI has the stature to attract first-rate programmers and mathematicians to enhance DERIVE's performance and reliability, and to extend the range of its capabilities. Also authors will be more likely to write and publishers more likely to publish math and science books based on DERIVE if TI's name is on the product.

Several users have expressed concern that DERIVE will be transformed from an efficient and compact CA system into a behemoth much like our competitors'. E&PS pioneered development of the graphing calculator, and more recently (in cooperation with Soft Warehouse) has produced calculators having symbolic math capabilities. The relatively modest computing power and memory size of such portable platforms gives E&PS a vested interest in preserving DERIVE's compact size and high performance. Also, the calculator paradigm that DERIVE uses fits in well with that used on products designed for math and science education.

muLISP (the LISP system in which DERIVE is written) and DERIVE (and its predecessor muMATH) have steadily evolved over the past 20+ years. The mistakes and successes I made in the process have taught me a great deal about how to implement a computer algebra system. I am not going to live forever. I am determined to pass this knowledge on to others more capable than me so that DERIVE and/or her successors will continue to be used and improved. Thus, as soon as version 5 is released, one of my primary consulting duties will be to transfer this knowledge to software engineers at E&PS. I and the other authors greatly appreciate your loyalty and enthusiasm for DERIVE, and your concern about her future. But, now it's time to get back to work on version 5!

Aloha,
Albert D. Rich
Co-author of DERIVE

P.S. We recently received the following email from my friend and colleague, Julio Valella of Texas Instruments:

==================================================================

Al, David, and Theresa:

I am delighted to be assigned as the Texas Instruments DERIVE Product Manager.

In my new role I have a monumental challenge, to continue your long established excellence in development and support of the Derive family of products. And to help you complete Derive 5.0 and ship, promote, and support it worldwide with our renown TI-Cares™ Educator Support services.

We are a mere few days into this challenge! We are learning, organizing, training our staff, and marshalling resources required to maintain uninterrupted delivery and support of DfW 4.11 while we prepare to launch and support DfW5.0 worldwide. Some of the forthcoming TI-Cares ™ services Derive enthusiasts should enjoy in the near future include:

+ Workshop Loan Program - Texas Instruments free loan service to educators and professionals who wish to conduct training, academic presentations, teacher workshops, and on site institutional demonstrations.

+ Conference Exhibits - Texas Instruments exhibits at most math and science conferences worldwide. Beginning this November Derive will be displayed with opportunity for hands on demonstrations, literature, applications and support services information.

+ World Wide Web information dedicated about Derive will continue at http://www.derive.com/ Soon global exposure and support for Derive expands at http://www.ti.com/calc/docs/software.htm

As the Team Leader of TI-92 development my emotional connections with Derive and its development team run deep. Now I am eager to get to know and work with Derive customers and enthusiasts worldwide.

With a little help from all our friends I'm confident we will meet our challenge and measure up to the high expectations you have conditioned in Derive users!

Cheers,
Julio Valella
Derive Product Manager
Texas Instruments Educational & Productivity Solutions
jvalella@ti.com
http://www.ti.com/calc/

This wrote Heinz Rainer Geyer from Germany:

Hey Albert,

A wonderful letter.
it's all about math, we know, but your comparison really got something!
So, what is our (DERIVERS) play: I thing we all fear to loose that beautiful woman student, we (they all) all in love with, to that tall, rich and smart guy, who soon will leave her alone, broken and ruined.

But sometimes that guy appears to be her real big love, and when they got married the future will start in their baby with even greater capacities: the cleverness of the grandfather combined with the smartness of the father.

Back to reality: a few days ago at ACDCA conference in Goesing Danny Gremillion of TI presented some features of that new TI InterActive, which has no CAS in Version 1 and some other important features are also missing, but it looks like that tough guy.
And DERIVE 5 appears dressed up, so he has to look at her.

In our days education always has to have a look on fashion. Pupils got to have new designs like worksheets with free shifting entries and one-click-connection to embedded objects, it also should have internet facilities to present your work on schools homepage and all that snick-snack.

But one day the both (TI-InterActive and DERIVE) will have their baby, which usually get the name of that grandfather.

Name it DERIVE InterActive

So, wipe your tears away and take a drink with the grandfather in law.

Cheers
Rainer

The last words again to the person, who started and encouraged the discussion on the internet, Terence Etchells:

Well said Sir!!!

Cheers

Terence

# ON THE SOLUTION OF A LINEAR DIFFERENTIAL EQUATION OF THE ORDER *n* WITH CONSTANT COEFFICIENTS (3)

**Emilio Defez Candel , Vicente Soler Basauri ,** Departamento de Matemática Aplicada
Universidad Politécnica de Valencia, P. O. BOX 22.012. Valencia. Spain

## 4 Complete linear differential equation of the *n*th order with constant coefficients.

Function `DIFERENCIAL_HOMOGENEA_COEFICIENTES_CONSTANTES(v,x)`

can also be used to calculate solutions of the complete linear differential equation of the *n*th order with constant
coefficients.

$$a_n\, y^{n)}(x)+a_{n-1}\, y^{n-1)}(x)+\ldots+a_1 y'(x)+a_0 y(x)=f(x)\ . \tag{20}$$

Let us suppose that $y_p(x)$ is a particular solution of (20). Then the change $y=z+y_p$ transforms (20) into a type (6) homogeneous differential equation. Let us suppose that $y_H(x)$ is the general solution of the homogeneous equation associated to (20) given by

$$a_n\, y^{n)}(x)+a_{n-1}\, y^{n-1)}(x)+\ldots+a_1 y'(x)+a_0 y(x)=0 \tag{21}$$

It is then inferred that the general solution of equation (20) is given by

$$y(x)=y_H(x)+y_p(x) \tag{22}$$

This solution contains the *n* arbitrary constants that must appear in the general solution of an differential equation of the *n*th order, since they will be contained in the expression of $y_H(x)$. Based on the theorem of existence and uniqueness of solutions 2.1 it can be demonstrated that (22) is actually the general solution of (20). Thus, the problem of solving (20) is reduced to the problem of finding a $y_p(x)$ particular integral of (20). We are going to see two methods, the method of undetermined coefficients and the method of variation of constants, which will enable us to obtain this $y_p(x)$ particular solution of (20).

### 4.1 Method of undetermined coefficients.

This method consists of writing a similar expression to that of *f(x)* as an $y_p(x)$ possible particular solution, but with undetermined coefficients. These coefficients will be determined in such a way that when $y$ is substituted by $y_p(x)$in (20), both members of the equality are identical. The following rules enable the each form. $y_p(x)$ac cording to the function *f(x)*.

1. If $f(x)=P_m(x)$, where $P_m(x)$ is an m-degree polynomial in *x*, we will state

$$y_p(x)=\left(A_0+A_1 x+\ldots+A_m x^m\right)x^k \tag{23}$$

*k* being the multiplicity order of 0 as a root of the characteristic polynomial associated to equation (21) and the coefficients $\{A_0, A_1, \ldots, A_m\}$are the constants to be determined.

2. If $f(x)=e^{\alpha x}P_m(x)$, where $P_m(x)$ is an m-degree polynomial in x and $\alpha \in R$, we will state

$$y_p(x)=e^{\alpha x}\left(A_0+A_1x+\ldots+A_mx^m\right)x^k \tag{24}$$

$k$ being the multiplicity order of $\alpha$ as a root of the characteristic polynomial associated to equation (21) and the coefficients $\{A_0, A_1, \ldots, A_m\}$ are the constants to be determined.

3. If $f(x)=\begin{cases} P_s(x)\sin(\beta x) \\ P_c(x)\cos(\beta x) \\ P_s(x)\sin(\beta x)+P_c(x)\cos(\beta x)\end{cases}$, where in each one of the cases $P_s(x)$ and $P_c(x)$ are polynomials in x in s and c degrees and $\beta \in R$, let m be $m=\begin{cases} s \\ c \\ \max\{s,c\}\end{cases}$ according to which the one corresponding to $f(x)$ is. $m$ thus defined, in any of the three cases we will state

$$y_p(x)=\left[\left(A_0+A_1x+\ldots+A_mx^m\right)\sin(\beta x)+\left(B_0+B_1x+\ldots+B_mx^m\right)\cos(\beta x)\right]x^k \tag{25}$$

$k$ being the order of multiplicity of $i\beta$ as a root of the characteristic polynomial associated to equation (21) and the coefficients $\{A_0, A_1, \ldots, A_m, B_0, B_1, \ldots, B_m\}$ are the constants to be determined.

4. If $f(x)=e^{\alpha x}\left[P_s(x)\sin(\beta x)+P_c(x)\cos(\beta x)\right]$, where $P_s(x)$ and $P_c(x)$ denote polynomials in $x$ of $s$ and $c$ respectively, $\alpha,\beta \in R$. Let $m$ be $m=\max\{s,c\}$, we will state

$$y_p(x)=e^{\alpha x}\left[\left(A_0+A_1x+\ldots+A_mx^m\right)\sin(\beta x)+\left(B_0+B_1x+\ldots+B_mx^m\right)\cos(\beta x)\right]x^k \tag{26}$$

$k$ being the order of multiplicity of $\alpha+i\beta$ as a root of the characteristic polynomial associated to equation (21) and the coefficients $\{A_0, A_1, \ldots, A_m, B_0, B_1, \ldots, B_m\}$ are the constants to be determined so that $y_p(x)$ actually be a particular solution of equation (20).

Let us observe that each one of these particular cases is already included in this last case. We have described them in this way, however, to facilitate their identification.

**Example 4.1** *We want to find the complete solution of the differential equation:*

$$2y''(x)-y'(x)-y(x)=x\sin(3x)+2\cos(3x)\ . \tag{27}$$

*For that purpose, we will first solve the homogeneous equation*

$$2y''(x)-y'(x)-y(x)=0\ , \tag{28}$$

*with DERIVE, where in this case* `v = [2, -1, -1]`, *obtaining:*

(#1 to #33 are functions provided in CONST.MTH from DNL#33.)

```
#34:   DHCC(v, x) := DIFERENCIAL_HOMOGENEA_COEFICIENTES_CONSTANTES(v, x)
```

$$\#35:\quad \mathrm{DHCC}([2, -1, -1], x) = e^{x}\cdot c_{1,1} + e^{-x/2}\cdot c_{2,1}$$

*Taking into account the expression of f(x), we must find a solution of the form*

$$y_p(x) = [P_m(x)\sin(3x) + P_m(x)\cos(3x)]x^k\,, \tag{29}$$

*being* $m = \max\{1,0\} = 1$ *and k the multiplicity of* $3i$ *as a root of the characteristic polynomial of equation (28). This characteristic polynomial is given by the function* `AUX_1([2, -1, -1])`.

```
#36:   AUX_1([2, -1, -1]) = 2·x^2 - x - 1

#37:   2·(3·i)^2 - 3·i - 1 = -19 - 3·i
```

*As we can see,* $3i$ *is not a root of the characteristic equation obtained, so that k = 0. Thus, we look for a solution*

$$y_p(x) = (Ax + B)\sin(3x) + (Cx + D)\cos(3x), \tag{30}$$

*where coefficients A, B, C, D have to be determined. This being a second order differential equation, we calculate the two first derivatives of (30):*

```
#38:   Y_P(x) := (a·x + b)·SIN(3·x) + (c·x + d)·COS(3·x)

#39:   d/dx Y_P(x) = (3·a·x + 3·b + c)·COS(3·x) - (3·c·x - a + 3·d)·SIN(3·x)

#40:   (d/dx)^2 Y_P(x) = - 3·(3·c·x - 2·a + 3·d)·COS(3·x) - 3·(3·a·x + 3·b + 2·c)·SIN(3·x)
```

*Replacing in differential equation (27), we obtain*

```
#41:   2·(d/dx)^2 Y_P(x) - d/dx Y_P(x) - Y_P(x)

#42:   - (x·(3·a + 19·c) - 12·a + 3·b + c + 19·d)·COS(3·x) - (x·(19·a - 3·c) + a + 19·b +
        3·(4·c - d))·SIN(3·x)
```

*Now equalling the coefficients obtained with function f(x), a system of equations is obtained, which we solve:*

```
#44:   SOLVE([12·a - 3·b - c - 19·d = 2, - 3·a - 19·c = 0, - 19·a + 3·c = 1, -a - 19·b -
        3·(4·c - d) = 0], [a, b, c, d])

#45:   [a = - 19/370 ∧ b = - 809/34225 ∧ c = 3/370 ∧ d = - 9199/68450]
```

*We check that the function obtained is a particular solution of (27)*

```
#46:   Y_P(x) := (- 19/370·x - 809/34225)·SIN(3·x) + (3/370·x - 9199/68450)·COS(3·x)

#47:   2·(d/dx)^2 Y_P(x) - d/dx Y_P(x) - Y_P(x) = 2·COS(3·x) + x·SIN(3·x)
```

*The general solution of (27) is given by the addition of the particular solution and the general solution of the homogeneous (28):*

```
#48:  Y(x) := Y_P(x) + DHCC([2, -1, -1], x)
```

$$\#49:\quad Y(x) := e^{x}\cdot c_{1,1} + e^{-x/2}\cdot c_{2,1} + \left(\frac{3\cdot x}{370} - \frac{9199}{68450}\right)\cdot COS(3\cdot x) - \left(\frac{19\cdot x}{370} + \frac{809}{34225}\right)\cdot SIN(3\cdot x)$$

Let DERIVE 6.10 do the job.

in 0.09 seconds:

$$\#50:\quad DSOLVE2\left(-\frac{1}{2}, -\frac{1}{2}, \frac{2\cdot COS(3\cdot x) + x\cdot SIN(3\cdot x)}{2}, x\right)$$

$$\#51:\quad c1\cdot e^{x} + c2\cdot e^{-x/2} + \frac{(555\cdot x - 9199)\cdot COS(3\cdot x)}{68450} - \frac{(3515\cdot x + 1618)\cdot SIN(3\cdot x)}{68450}$$

© Let's try with TI−NspireCAS:

$\text{deSolve}(2\cdot y''-y'-y=x\cdot \sin(3\cdot x)+2\cdot \cos(3\cdot x),x,y)$

$$y=c3\cdot e^{\frac{-x}{2}}+c4\cdot e^{x}+\left(\frac{3\cdot x}{370}-\frac{9199}{68450}\right)\cdot \cos(3\cdot x)+\left(\frac{-19\cdot x}{370}-\frac{809}{34225}\right)\cdot \sin(3\cdot x)$$

## 4.2 Method of variation of constants.

The method of variation of constants or Lagrange method is more general than the method of undetermined coefficients. It consists of starting from the general solution of the homogeneous problem associated to (20):

$$y_H(x) = C_1\, y_1 + C_2\, y_2 + \ldots + C_n\, y_n \quad , \tag{31}$$

and deal with the constants $\{C_1, C_2, \ldots, C_n\}$ as **functions** of *x*, imposing them certain conditions so that the function thus obtained

$$y_H(x) = C_1(x) y_1 + C_2(x) y_2 + \ldots + C_n(x) y_n \quad , \tag{32}$$

*be* a particular solution of (20). The way to find the functions $C_i(x)$ is supplied by the following result:

**Theorem 4.1** *If* $\{y_1, y_2, \ldots, y_n\}$ *constitutes a fundamental set of solutions of homogeneous differential equation (21) associated to equation (20), and if functions* $\{C_1(x), C_2(x), \ldots, C_n(x)\}$ *verify the system of equations:*

$$\left.\begin{array}{c} C'_1(x)y_1 + C'_2(x)y_2 + \ldots + C'_n(x)y_n = 0 \\ C'_1(x)y'_1 + C'_2(x)y'_2 + \ldots + C'_n(x)y'_n = 0 \\ C'_1(x)y''_1 + C'_2(x)y''_2 + \ldots + C'_n(x)y''_n = 0 \\ \vdots \\ C'_1(x)y^{n-2)}{}_1 + C'_2(x)y^{n-2)}{}_2 + \ldots + C'_n(x)y^{n-2)}{}_n = 0 \\ C'_1(x)y^{n-1)}{}_1 + C'_2(x)y^{n-1)}{}_2 + \ldots + C'_n(x)y^{n-1)}{}_n = \dfrac{f(x)}{a_n} \end{array}\right\} \quad (33)$$

*then* $y(x) = C_1(x)y_1 + C_2(x)y_2 + \ldots + C_n(x)y_n$ *,is a particular solution of equation (20).*

Let us observe that the system of equations (33) can be written in matrix way:

$$\begin{pmatrix} y_1 & y_2 & y_3 & \cdots & y_n \\ y'_1 & y'_2 & y'_3 & \cdots & y'_n \\ y''_1 & y''_2 & y''_3 & \cdots & y''_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y^{n-1)}{}_1 & y^{n-1)}{}_2 & y^{n-1)}{}_3 & \cdots & y^{n-1)}{}_n \end{pmatrix} \begin{pmatrix} C'_1(x) \\ C'_2(x) \\ C'_3(x) \\ \vdots \\ C'_n(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \dfrac{f(x)}{a_n} \end{pmatrix}, \quad (34)$$

where the invertibility of the matrix

$$\begin{pmatrix} y_1 & y_2 & y_3 & \cdots & y_n \\ y'_1 & y'_2 & y'_3 & \cdots & y'_n \\ y''_1 & y''_2 & y''_3 & \cdots & y''_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y^{n-1)}{}_1 & y^{n-1)}{}_2 & y^{n-1)}{}_3 & \cdots & y^{n-1)}{}_n \end{pmatrix}, \quad (35)$$

is guaranteed for $\{y_1, y_2, \ldots, y_n\}$ being a fundamental set of solutions. In order to Apply the method of variation of constants, we will use the following auxiliary functions of *DERIVE* included in the file **CONST.MTH**

$$\text{V\_C\_AUX\_2}(v, c, i, j) := \left(\frac{d}{dx}\right)^j c_i$$

$$\text{V\_C\_AUX\_3}(v, c) := \text{VECTOR}\left(\left(\frac{d}{dx}\right)^j c,\ j,\ 0,\ \text{DIM}(v) - 2\right)$$

$$\text{V\_C\_AUX\_4}(v, c) := \text{V\_C\_AUX\_3}(v, c)^{-1}$$

$$\text{V\_C\_AUX\_5}(v, c, f) := \text{VECTOR}\left(\frac{f}{v_1}\cdot\text{KRONECKER}(j, \text{DIM}(v) - 2),\ j,\ 0,\ \text{DIM}(v) - 2\right)$$

```
V_C_AUX_6(v, c, f) := V_C_AUX_4(v, c)·V_C_AUX_5(v, c, f)

V_C_AUX_7(v, c, f) := ∫ V_C_AUX_6(v, c, f) dx

V_C_AUX_8(v, c, f) := VECTOR(1, j, 0, DIM(v) - 2)

V_C_AUX_9(v, c, f) := V_C_AUX_7(v, c, f)·c

SOLUCION_PARTICULAR_COMPLETA(v, c, f) := V_C_AUX_9(v, c, f)
```

In vector `c` we write a fundamental set of solutions of the homogeneous differential equation (21), which can be obtained by means of the function:

`DIFERENCIAL_HOMOGENEA_COEFICIENTES_CONSTANTES(v,x).`

Function `V_C_AUX_3(v,c)` calculates matrix (35). The rest of the functions serve in order to solve the matrix system (34).

**Example 4.2** *We want to find a complete solution of the differential equation:*

$$y'''(x)+y'(x)=\mathrm{cosec}(x). \qquad (36)$$

*For that purpose, we firstly solve the homogeneous equation*

$$y'''(x)+y'(x)=0, \qquad (37)$$

*with DERIVE , where in this case* $v=[1,0,1,0]$ *thus obtaining the solution:*

```
#52:  DHCC([1, 0, 1, 0], x) = COS(x)·d     + SIN(x)·r     + c
                                      2,1            2,1     1,1

#53:  Y_H(x) := COS(x)·d     + SIN(x)·r     + c
                        2,1            2,1     1,1
```

*A fundamental solution set of solution of the associated homogeneous equation (37) is given by the terms that accompany the constants in the solution obtained:*

$$\{1, \sin(x), \cos(x)\}.$$

*We apply the function of DERIVE*

`SOLUCION_PARTICULAR_COMPLETA(v, c, f)`

*where in this case* $v=[1,0,1,0]$, $c=[1,\sin(x),\cos(x)]$ *and* $f(x)=\cos ec(x)$, *obtaining the particular solution:*

```
#54:  SOLUCION_PARTICULAR_COMPLETA([1, 0, 1, 0], [COS(x), SIN(x), 1], CSC(x))

#55:  - LN(COS(x) + 1) + LN(SIN(x))·(1 - COS(x)) - x·SIN(x)

#56:  Y_P(x) := - LN(COS(x) + 1) + LN(SIN(x))·(1 - COS(x)) - x·SIN(x)
```

*Adding to this particular solution the one previously obtained of the homogeneous equation, we obtain:*

```
#57:  Y(x) := - LN(COS(x) + 1) + LN(SIN(x))·(1 - COS(x)) - x·SIN(x) + COS(x)·d     + SIN(x)·r     + c
                                                                           2,1            2,1     1,1
```

*which, as we can see, is actually a solution of equation (36):*

$$\#58: \quad \left(\frac{d}{dx}\right)^3 Y(x) + \frac{d}{dx} Y(x) - CSC(x) = 0$$

**REMARK**: It is important to point out that when the dimension of the system is large, **Derive** will have plenty of problems in order to obtain the inverse of the matrix (35). This is why the method of undetermined coefficients is preferable.

## References.

**J.S. Berry, E. Graham y A. J. P. Watkins** (1993). *Learning Mathematics through* **Derive** , Ed. Ellis Horwood. London.

**E. Defez**, (1997). Cálculo de la matriz exponencial con **Derive** . Revista DELTA, editada por la Asociación de usuarios de **Derive** en España. Valencia. No. 1. 45-59. (In Spanish)

**E. Defez**, (1997). Cálculo del polinomio mínimo de una matriz con **Derive** . Revista EPSILON, editada por la S.A.E.M. "THALES". N.37 Vol. 13(1).11-20 . (In Spanish)

**T. M. Flett**, (1980). *Differential Analysis,* Cambridge University Press, Cambridge,.

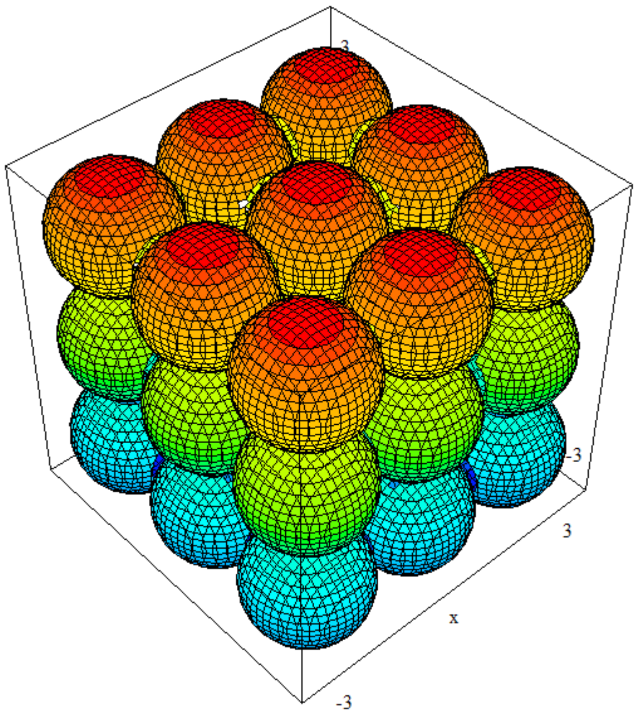
**A. Kiselov, M. Krasnov** and **G. Makarenko** (1979). *Problemas de ecuaciones diferenciales ordinarias.*, MIR Moscú,. (In Spanish)

**J. L. Llorens Fuster**, (1993). *Introducción al uso del* **Derive**, Publicaciones Universidad Politécnica de Valencia, 95.683.. (In Spanish)

**C. Paulogorrán y C. Pérez**, (1994). *Cálculo matemático con DERIVE para P.C.* Ed. RAMA, Madrid. (In Spanish)

**G. E. H. Reuter**, (1962). *Elementary differential equations and operators,* Routledge and Kegan Paul, London.

**Soft Warehouse INC**, (1994). **Derive** user manual**.** Version 3, Honolulu.



In DNL#33 and #34 Rüdeger Baumann posted two sequences:

(1) **[1, 3, 7, 12, 18, 26, 35, 45, 56, 69, ....]** (Hofstadter-Sequence, which forms together with the sequence of its differences the set of the natural numbers without zero).

and asked for a *DERIVE*-program to produce this sequence.

Johann Wiesenbauer was in the mood to produce a very short DERIVE-file:

```
P.S: As for the problem posed by Ruedeger Baumann in the DNL #33, here
is a possible solution

F(n) := REVERSE_VECTOR(IF(n = 1, [1], IF(n = 2, [3, 1],
ITERATE(INSERT_ELEMENT(IF(SELECT(v_ = f_sub1 - f_sub2 + 1, v_, f_) = [],
2f_sub1 - f_sub2 + 1, 2f_sub1 - f_sub2 + 2), f_), f_, [3, 1], n - 2))))

F(10) = [1, 3, 7, 12, 18, 26, 35, 45, 56, 69]

(If you only want the nth element of this sequence, it should be easy to
change this definition accordingly!)
```

(2) The second problem was the "**Look´n Say Sequence**":

**[1, 11, 21, 1211, 111221, "three ones, two twos, one one" = 312211, 13112221, ......]**

and in DNL#34´s User Forum Rüdiger wondered if it would be easy to realize this sequence in *DERIVE*, because one has to consider the order of the numbers. I tried and it took me one interesting evening to solve the problem (Josef).

Look and Say: (it works, indeed!)

```
H(v_):=SELECT(u SUB 1/=0,u,APPEND(VECTOR(IF(v_ SUB (i+1)/=v_ SUB i,[i,
       v_ SUB i],[0,0]),i,1,DIMENSION(v_)-1),[[DIMENSION(v_),
       v_ SUB DIMENSION(v_)]]))

H3(v_):=APPEND(APPEND([(H(v_)) SUB 1],VECTOR([(H(v_)) SUB (i+1) SUB 1-
        (H(v_)) SUB i SUB 1,(H(v_)) SUB (i+1) SUB 2],i,1,
        DIMENSION(H(v_))-1)))

LOOKNSAY(st,n):=ITERATES(H3(u_),u_,[st],n)

LOOKNSAY(1,5) = [[1],[1,1],[2,1],[1,2,1,1],[1,1,1,2,2,1],[3,1,2,2,1,1]]

LOOKNSAY(3,6)

[[3],[1,3],[1,1,1,3],[3,1,1,3],[1,3,2,1,1,3],[1,1,1,3,1,2,2,1,1,3],[3,1,1,3
,1,1,2,2,2,1,1,3]]

POT(v):=VECTOR(10^(DIMENSION(v)-i),i,1,DIMENSION(v))

ZF(st,n):=VECTOR(w . POT(w),w,LOOKNSAY(st,n))

ZF(1,5) = [1,11,21,1211,111221,312211]

ZF(1,10) = [1,11,21,1211,111221,312211,13112221,1113213211,31131211131221,
13211311123113112211,11131221133112132113212221]

ZF(2,5) = [2,12,1112,3112,132112,1113122112]
```

The question for me – Josef – was:
How could a function/program look like which generates the Hofstadter sequence?

Is it possible to transfer Johann's tricky function from above into the Nspire language?
I found out that I would miss REVERSE, ITERATE, INSERT and SELECT among the Nspire functions. Reinventing these functions together with some others collected under *Vector Manipulating Functions* in the DERIVE Online Help could be the first step. ITERATE will be substituted by a loop.

So let's start. I create a library for manipulating lists (not vectors, because these objects cannot be compared with the Nspire vectors) called *listops*.

Library **listops** containing list operations (DERIVE like):

**select1(u,k,l)** returns a list of elements **k** of list **l** for which **u(k)** is true.
Example:
```
select1(a>10 and isPrime(a)=true,a,seq(k,k,1,30)) ▸ {11,13,17,19,23,29}
```

**insert(u,l,n)** returns a copy of list **l** with expression **u** inserted before the **n-th** element.
Example:
```
insert("inserted",seq(k,k,1,10),4) ▸ {1,2,3,"inserted",4,5,6,7,8,9,10}
```

**reverse(l)** returns a copy of list **l** in reverse order.
Example:
```
reverse(seq(k,k,5,15)) ▸ {15,14,13,12,11,10,9,8,7,6,5}
```

```
select1(a>10 and isPrime(a)=true,a,seq(k,k,1,30))
                                   {11,13,17,19,23,29}
insert(3,seq(k,k,1,10),4)      {1,2,3,3,4,5,6,7,8,9,10}
reverse(seq(k,k,5,15))
                        {15,14,13,12,11,10,9,8,7,6,5}
```

reverse 1/1
```
Define LibPub reverse(l)=
Func
seq(l[k],k,dim(l),1,-1)
EndFunc
```

select1 6/8
```
Define LibPub select1(c_,v_,l_)=
Func
Local ll,i,lc
ll:={[]}
For i,1,dim(l_)
lc:=c_|v_=l_[i]
If lc=true
 ll:=augment(ll,{l_[i]})
 EndFor
 ll
EndFunc
```

insert 1/1
```
Define LibPub insert(u,l,n)=
Func
augment(augment(seq(l[k],k,1,n-1),{u}),seq(l[k],k,n,dim(l)))
EndFunc
```

As there is a SELECT-function in a TI-Nspire library (*numtheory*) I took the function name select1. Unfortunately it caused some problems which I don't understand. The first argument should be a Boolean expression. The first two select1 calls are working, but combining both expressions by an AND fails. As you can see above one has to accomplish the second expression to isPrime(x)=true to make it work. Don't ask me why. numtheory\select works without this accomplishment, but only accept x as a variable and the B.e. must be entered as a string.

| | |
|---|---|
| *select1*(*a*>10,*a*,seq(*k*,*k*,1,30)) | {11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30} |
| *select1*(isPrime(*z*),*z*,seq(*k*,*k*,1,30)) | {2,3,5,7,11,13,17,19,23,29} |
| *select1*(*a*>10 and isPrime(*x*),*x*,seq(*k*,*k*,1,30)) | "Error: Argument must be a Boolean expression or integer." |
| *numtheory**select*(seq(*k*,*k*,1,30),"x>10 and isPrime(x)") | {11,13,17,19,23,29} |
| *numtheory**select*(seq(*k*,*k*,1,30),"a>10 and isPrime(a)") | "Error: Argument must be a Boolean expression or integer." |

Now it is time to "translate" Johann's DERIVE-threeliner. We cannot expect to do it in three lines, too. The IF-construct alone does need more paper work. Anf then we have to set up the loop (here performed as a While – EndWhile – loop).

```
hofstadt(10)
                        {1,3,7,12,18,26,35,45,56,69}
                                                Done
hofstadt(20)
  26,35,45,56,69,83,98,114,131,150,170,191,213,236,260}
                                                Done
```

```
"hofstadt" stored successfully
Define hofstadt(n)=
Prgm
Local f,k,newel
If n=1 Then
 f:={1}
Else
   If n=2 Then
     f:={3,1}
   Else
     k:=0: f:={3,1}
     While k<n-2
       If listops\select1(x=f[1]-f[2]+1,x,f)={[]} Then
         newel:=2·f[1]-f[2]+1
       Else
         newel:=2·f[1]-f[2]+2
       EndIf
       f:=augment({newel},f)
       k:=k+1
EndWhile:EndIf:EndIf
Disp listops\reverse(f)
EndPrgm
```

We compare with the DERIVE – results and can be satisfied – translation works. I wonder if anybody of the Nspire-programmers could do it in a more elegant way. It is not necessary to refer to Johann's function but to the definition of the Hofstadter Sequence (DNL#33). Josef

## Detlev Kirmse: Desktops, Notebooks, or a Graphing Calculator? - A Mathematical Answer to an Economic Question[1]

February 2001. The assembly line is buzzing and clicking. The PC department of the large Augsburg electronics company Compunix installs desktop and notebook computers (a) with Hekatium™-processors[2]. Supply contracts provide up to 100 processors for installation in each 12-hour-shift (1'). The remaining material is enough to assemble 200 notebooks, a desktop uses 2½ times the amount that a notebook requires (2'). A desktop computer can be assembled in 5 minutes, whereas a notebook computer requires 9 minutes (3'). The company makes 150 € per desktop and 200 € per notebook (z').

A young businesswoman Eva Sailor is sitting in her office. The pale winter sun shines on the paper in front of her. She writes:

| | | | | | |
|---|---|---|---|---|---|
| (a) | desktops: | $x$ | (units) | | |
| | notebooks: | $y$ | (units) | | |
| (1') | $x + y \leq 100$ | | (processors) | (1) | $y \leq -x + 100$ |
| (2') | $y + 2.5x \leq 200$ | | (additional material) | (2) | $y \leq -2.5x + 200$ |
| (3') | $5x + 9y \leq 12 \cdot 60$ | | (assembly time in minutes) | (3) | $y \leq -\frac{5}{9}x + 80$ |
| | | | | (4) | $y \geq 0$(clear!) |
| | | | | (5) | $y \geq 0$ (ditto) |
| ($z'$) | $f_t(x,y) = 150x + 200y$ | | (profits in Euro. "Maximize!" says the boss) | ($z$) | $y = -\frac{3}{4}x + k$ (maximize $k$) |

Ms. Sailor chews on her pencil: which production numbers are to be planned? That is which pair of positive integers ($x$,$y$) promises the largest profits described in the target function ($f_t$) under the given boundary conditions (1) - (5)?

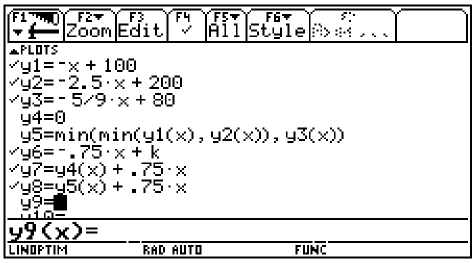
A typical problem of the linear optimization[3], formerly known as linear programming, a subfield of operational research. This term was first used by George B. Dantzig in 1949, who formulated this type of problem along with its mathematical solution [3][4]. This work was motivated, in part, by World War II, when the U.S. Air Force required clear planning procedures for the operation of airplanes and their supplies. Naturally there had also been preliminary studies in the civilian field, e. g. by the famous mathematician John von Neuman [1] and by his Russian colleague Kantorovitch [2]. In the meantime, this area of research has been established as a standard branch in applied mathematics.

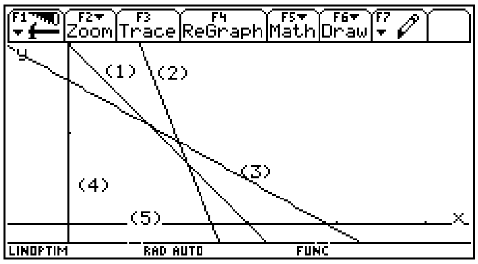
Our businesswoman turns round to the monitor and tries to start the company-owned PPS[4]-softwarepackage for production planning. Unfortunately, the in-house data network has been blocked for the half last hour because of heavy traffic. So, instead, she takes up her old school-calculator TI-92[5] and rummages through the mathematic knowledge she acquired in high school.

---

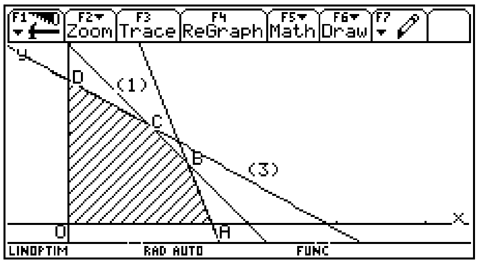
[1]This article was published first in BUS 34, Augsburg, 1998
[2]ηεκατον (gr.) = hundred
[3]"ein Verfahren, mit dem der kleinste oder größte Wert einer von zwei Variablen linear abhängenden Funktion bestimmt wird, deren Definitionsbereich wiederum durch ein lineares Ungleichungssystem bestimmt ist." [6]
[4]PPS: Produktionsplanung und -steuerung [7] *production planning and controlling*
[5]first symbolic graphing calculator, introduced in 1995 by Texas Instruments, cf. BUS 29, july 1995, p. 48.

She inputs the equations for the boundary conditions (1) - (4) ((5) is the y axis) in the `Y= Editor` of the TI-92 [8] (fig. 1). After adjusting the window for the coordinate system, she sees the graph shown in fig. 2:

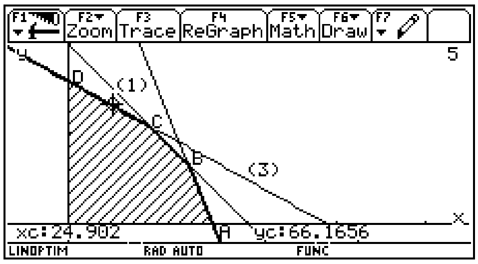


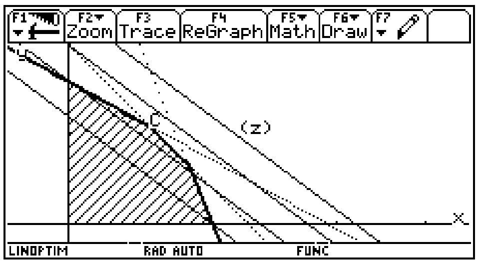

The five lines (1) - (5) define the region *OABCD* (fig. 3). She knows that all points within or on the edge of this convex planning polygon are solutions (*x*,*y*)[6] that meet the boundary conditions. She defines the right top border of the polygon with the function y5(x)=min(min(y1(x),y2(x)),y3(x)) and hatches the inside with the command
Shade  y4(x),y5(x),0,150,4,6.



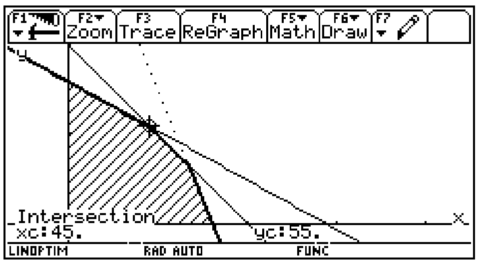

Which point gives the optimum value? Surely an "extreme value", i. e. a point on the edge. On possibility Ms. Sailor considers to "trace" this curve y5(x) by typing □ Trace, and to use given coordinates to experimentally find an optimal point (fig. 4).




But optimal according to which criterion? According to the target function ($f_t$) of course! She inputs the equation using the equation editor: y6=-.75*x+k. This function, however, contains the parameter *k* and therefore represents a family of lines. To look at the dependance of the target function of *k,* Ms. Sailor arbitrarily stores the set of values `{60, 80, 100, 120}` → `k` and draws the 4 selected copies of the target function (fig. 5).




The desired solution is the line with the maximum value of *k* that still contains a common point with the set of possible solutions. As one can see, the optimal point is the vertex *C*. Its coordinates can be calculated from the intersection of lines (1) and (2) using the command □ Math, 5: Intersection resulting in the coordinates (45,55) (fig. 6). Finished.

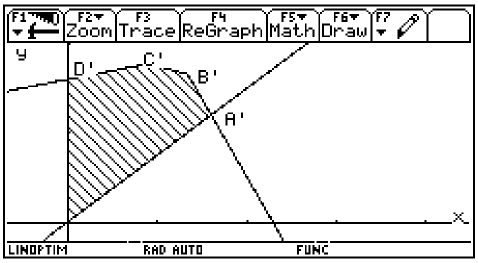


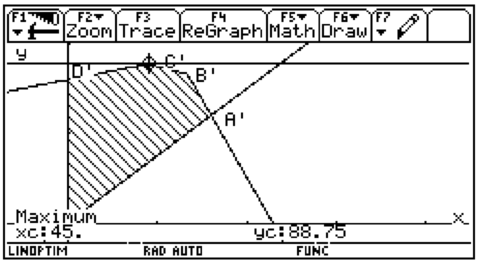

The procedure, however, has one weakness: The optimum point is determined empirically. But this can also be formalized.

---

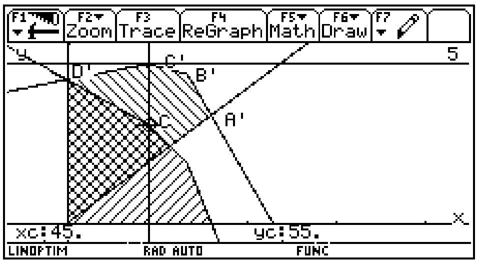
[6]In this case, only integer solutions are permissible, i. e. all grid points on or within the solution polygon

The polygon of possible solutions and the target function are mapped on the *y* axis as a function of the slope of the target function (fig. 7). Arithmetically, a line through the origin with the slope equal to that of the target function is subtracted from the functions `y4` (lower edge) and `y5` (top margin of the planning polygon) (in the # Editor the equations y7 and y8). By this geometric mapping the target function becomes horizontal (that is the trick!). The optimum point becomes the maximum of the path y7. Its maximum *C*' can easily be determined automatically by `F5` Math, 4: Maximum (fig. 8).




Mapping back *C*' on the original planning polygon to *C* is only a formality. Its coordinates (*x,y*), fortunately integers, now supply the optimal production numbers for one shift: 45 desktop computers and 55 notebook computers and a maximized profit of *z*(45,55) = 17750€(fig. 9).

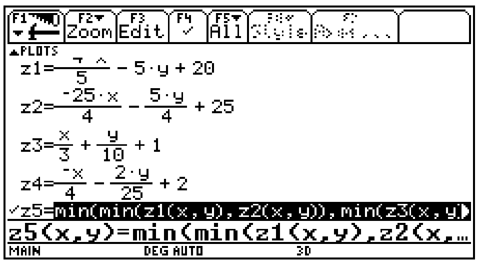



Our businesswoman puts her TI-92 aside and leans back. The whole procedure took hardly more than 8 minutes. She solved the linear optimization problem graphically - using a numeric tool. The problem is representative for a set of similar problems. Their solutions can be integers as in this case or real numbers. The problems are usually subject to more than five boundary conditions and lead in general to solution polygons with correspondingly more vertices. The target function may need to be maximized or minimized and the solution may be a single point (vertex) or a whole segment (edge).

Further examples include transportation problems (the famous "problem of the travelling salesman"), menu planning, traffic planning, controlling of machine production, and above all, economic optimization. In all cases the occurring relations are either linear or can be approximated as linear.

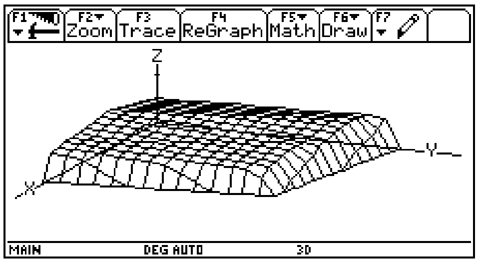
The graphic solution presented here can only be applied in this form with two variables (*x* and *y*). In the case of three variables, the diagram is already more complex:

| | | |
|---|---|---|
| consider the target function: | (z) | $f_z(x,y,z) = x + 10y + 100z$ |
| with the boundary conditions: | (1) | $4x + 25y + 5z \le 100$ |
| | (2) | $25x + 5y + 4z \le 100$ |
| | (3) | $10x + 3y - 30z \ge -30$ |
| | (4) | $25x + 8y + 100z \le 200$ |

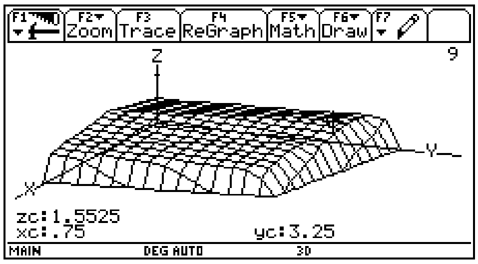



(fig. 10)

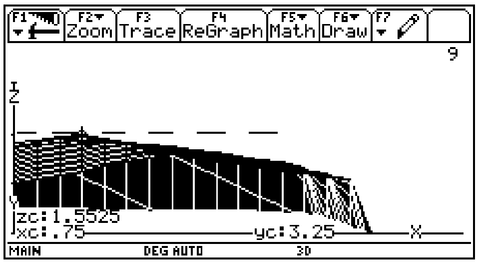
These inequatilities describe semi-spaces, which are cut off by planes in three-dimensional space. The solution space, given by the intersection of the semi-spaces, is a polyhedron (fig. 11).




Even here, the TI-92 can be used for visualization, although it scarcely offers a numeric assistance during the optimization with the graphic procedure. Using `F3`: Trace, one can trace on the grid lines of the polyhedron and approach the solution, determined by the resolution of the grid (fig. 12). However, the identification of the correct corner point is left to the eye.




A trick helps here also. If one rotates the axes such that the plane of the target function is horizontal, then the maximum shows up as "the highest elevation of the solution mountains" (fig. 13).




Inequalities with more than 3 variables cannot yet be visualized, not even in the year 2001. They represent hyperpolyhedrons in a space of the dimension higher than 3, which are limited by hyperlevels of one dimension less. For such optimization tasks one remains set back on the simplex procedure [5], a numeric algorithm which goes back to George B. Dantzig [4].

Then the telephone rings. It's the department manager. "Ms. Sailor, I just remembered we have two assembly lines that are no longer being used. What if we convert them to ...? Can you calculate it again? ... - say max. 15 per shift - and above that the processors for the new pocket calculators... ". With a soft "ping" Ms. Sailor's monitor reports back. In the menu of her PPS software she selects... linear optimization... simplex procedures...

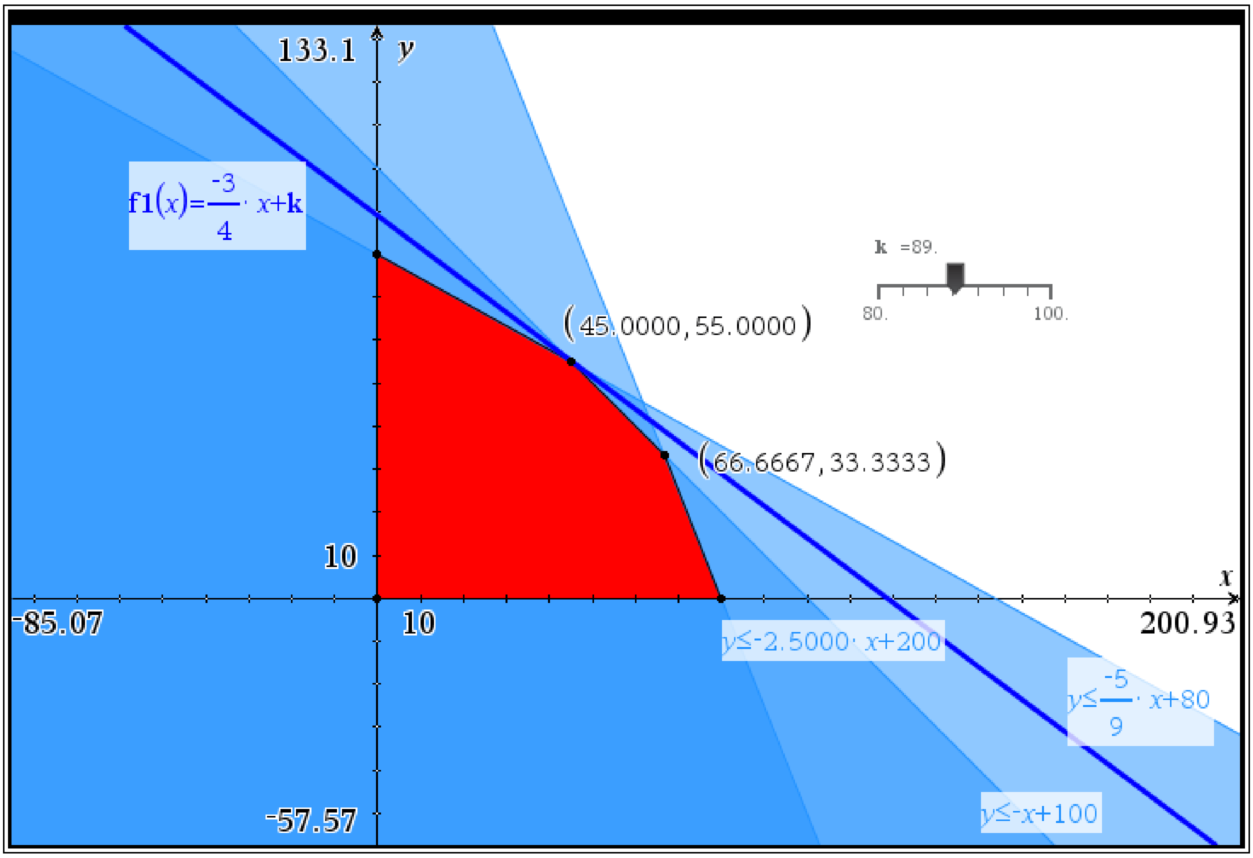
## Literature:

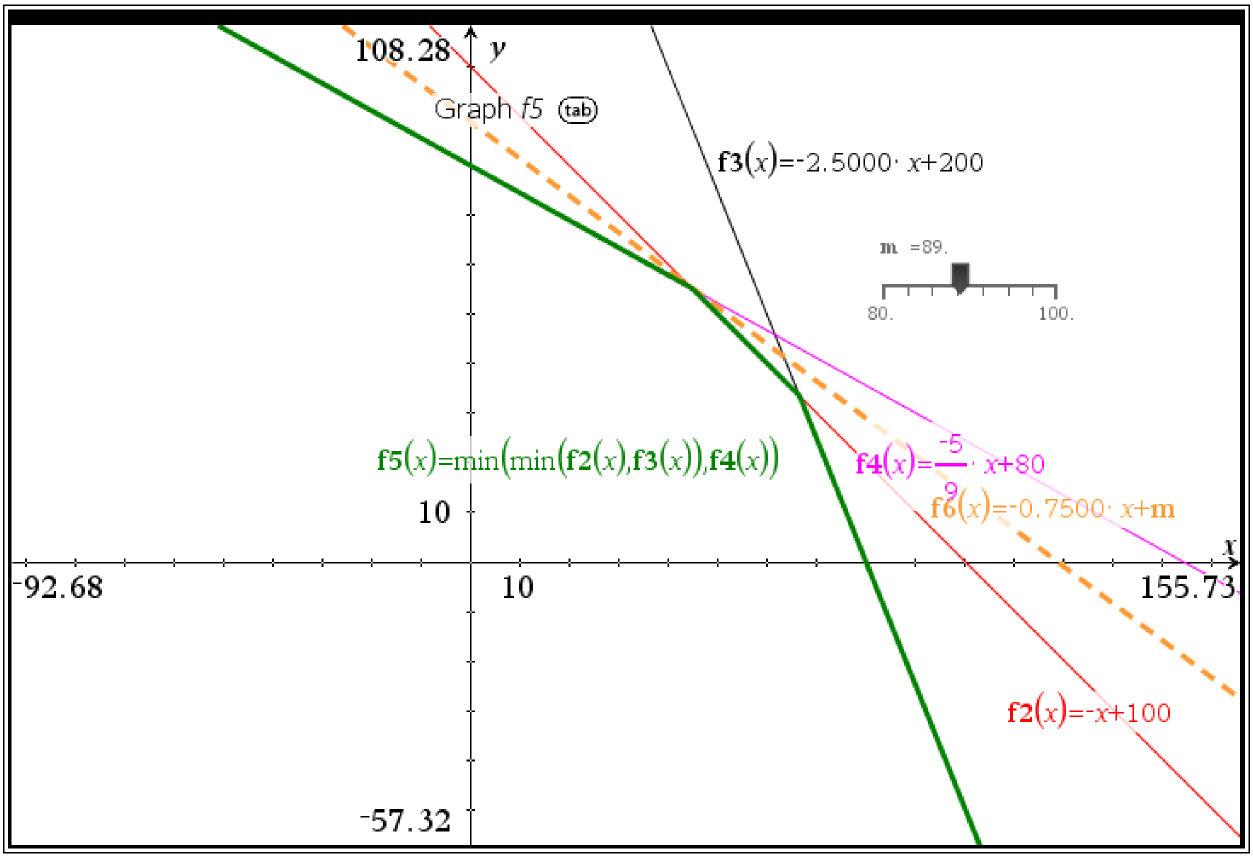
[1] von Neumann, John: Über ein ökonomisches Gleichungssystem und eine Verallgemeinerung des Brouwerschen Fixpunktsatzes. Ergebn. Math. Koll. 8, 1937.
[2] Kantorowitsch, L. V.: Mathematische Methoden in der Organisation und Planung der Production. Moskau, 1939.
[3] Dantzig, George B.: Programming in a linear structure. Washington D. C., USAF, 1948.
[4] Dantzig, George B.: Lineare Programmierung und Erweiterungen, dt. von Jaeger, A.. Berlin Heidelberg New York, 1966.
[5] Bartsch, H.-J.: Mathematische Formeln. Leipzig, 1969.
[6] Reck, Jürgen (ed.): Herder Lexikon Mathematik. Freiburg im Breisgau, 1975.
[7] Schulze, Hans Herbert: Computer Enzyklopädie. Hamburg, 1989.
[8] Bellemain, Franck a. o.: TI-92 handbook. Texas Instruments, 1996.

Detlev Kirmse
Zentralstelle für Computer im Unterricht
Schertlinstraße 9
D-86159 Augsburg

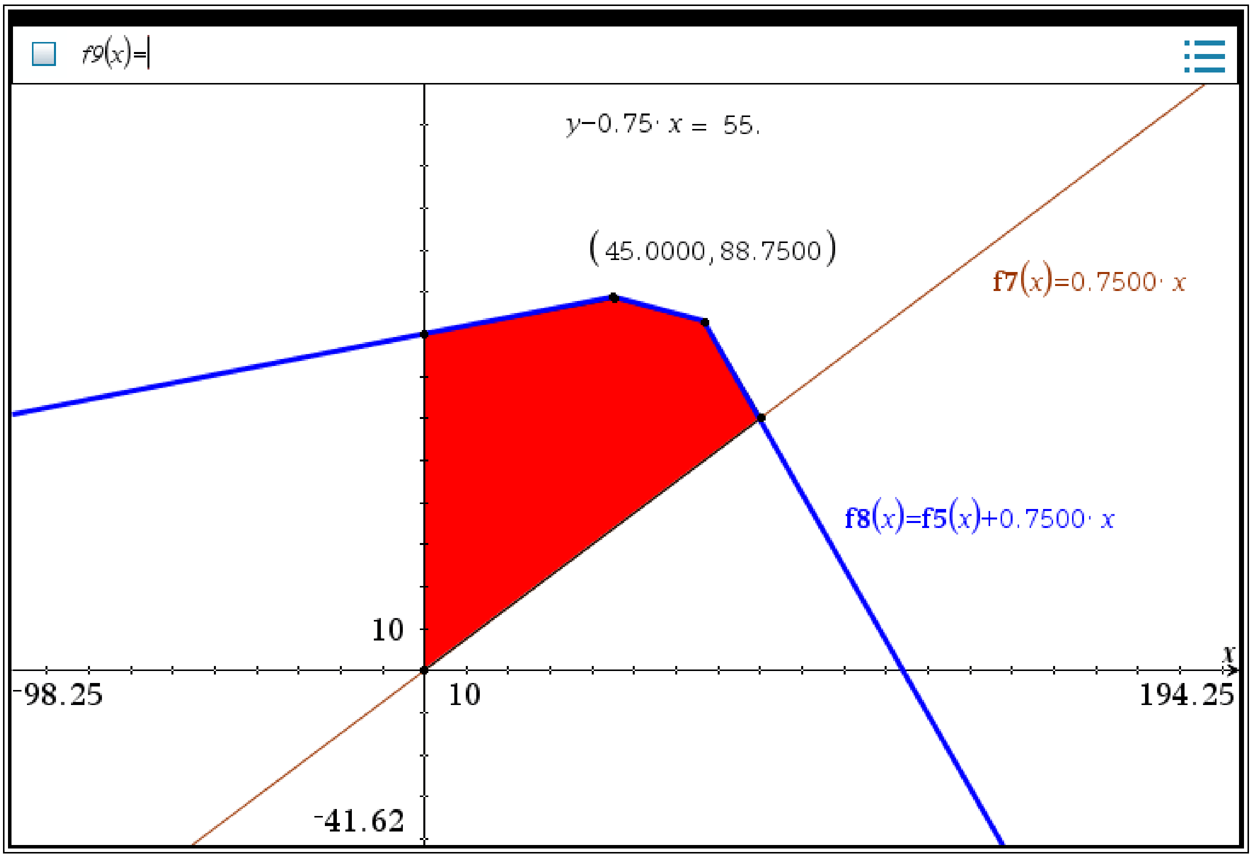
http//:www.zs-augsburg.de
kirmse@zs-augsburg.de

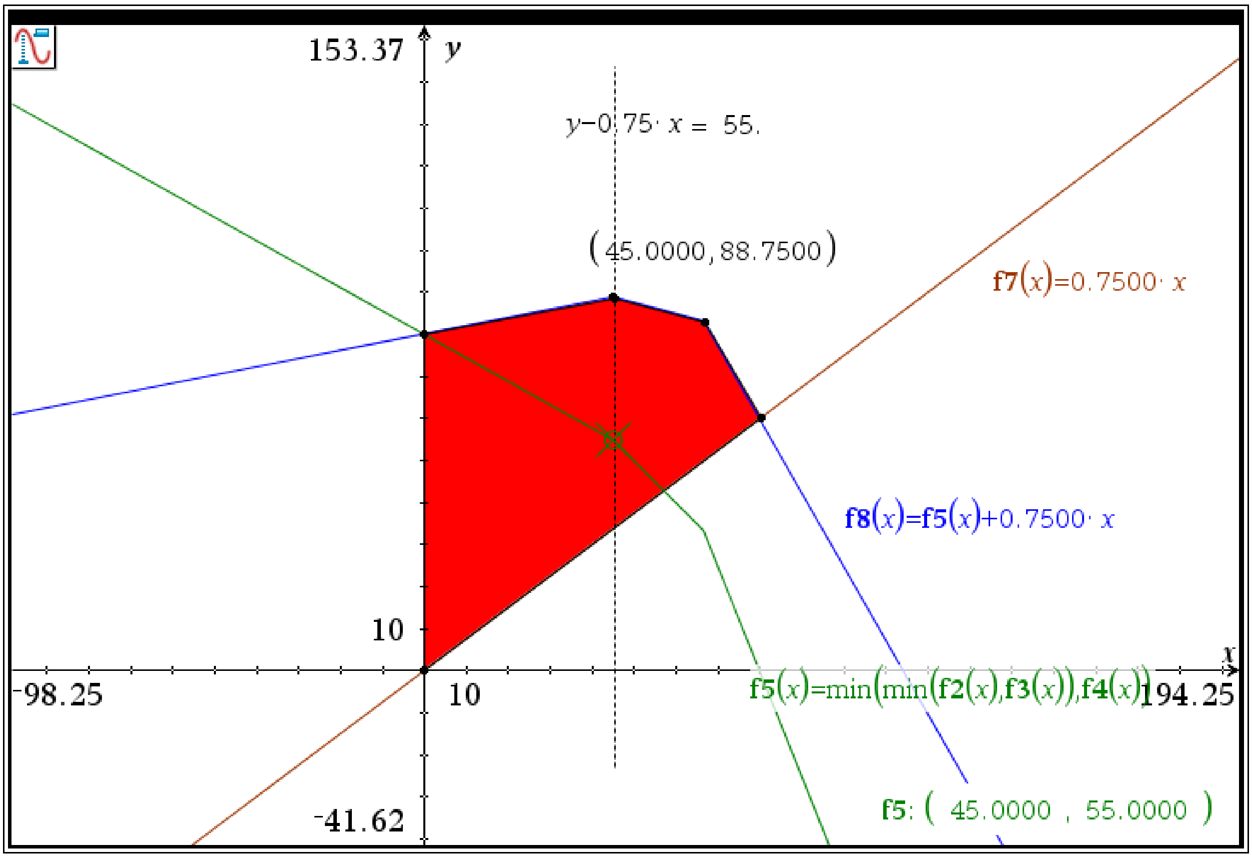
I append the graphic approach for treating 2 dimensional linear programming problems with the TI-Nspire CX CAS:



The procedure is very similar to the way to work with the TI-92. Unfortunately I was not able to reproduce the 3D-representation of Detlev's second problem. Josef



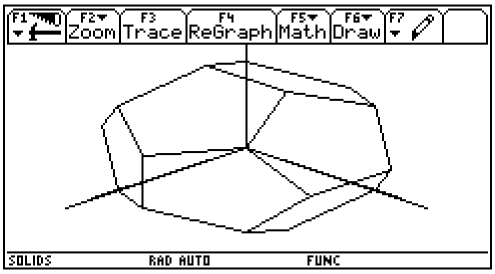



# Die Platonischen Körper - Platonian Solids - - Hidden Lines with TI-92 -

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

Inspired by Hubert Wellers´s presentation of polyhedrons (see DNL #29, p20-p25) we spent a lot of time in our school (in the scope of a working group CoMa - Computer-Mathematics - at the Gutenbergschule in Wiesbaden) with Platonian Solids.

A dodecahedron in isometric representation ➢




It was easy to realize the Hidden-Line-Algorithm using *DERIVE*. Working with the TI-92 we had to overcome some "hurdles", because the needed matrix operations – better operations applied on lists of matrices – are not possible. Creation of a new matrix from rows of an existing one has to be done element by element. Fortunately all boundary planes of Platonian Solids show the same number of vertices, so we could easily solve this problem.

We begin defining the solid by **one** matrix of 3D-coordinates. This matrix contains the closed sequence of points of each boundary plane. Matrix icosa(a) for the icosahedron consists of 20*4 = = 80 rows (20 planes with (3+1) points each).

In our realisation variable a has various meanings: it is the length of an edge (Tetrahedron and cube) or the length of the base-cube´s edge (octahedron, dodecahedron and icosahedron).

Entering the matrices based on our *DERIVE*-results was a bit laborious but it was also a useful training for our concentration and finally we obtained the matrix. I will not explain now the calculation of the coordinates.

The program allin(body,proj,n) draws all edges of solid body using the matrix of projection proj. n is the number of rows per face.

(Try: allin(dodeka(4),kav1,6))

```
:allin(m3,pro,n)
:Prgm
:Local m,l,i
:
:@ Start MainProg
:@ Transfer 3D-Points to 2D
:m3*pro→m
:For l,1,rowDim(m),n
:  For i,0,n-2,1
:  Line m[l+i,1],m[l+i,2],m[l+i+1,1],m[l
 +i+1,2],1
:  EndFor
:EndFor
:EndPrgm
SOLIDS          RAD AUTO          FUNC
```

Matrix proj converts matrix body by multiplication to a matrix representing a 2-dimensional object – a set of points in the plane -, which are all drawn using the line - - command.

hidd(body,proj,n) performs for each face the determinant-test before drawing. Because of the problems mentioned above the determinants are calculated in a subroutine. ( hidd(dodeka(4), kav1, 6) )

(Don´t forget: the sequence of points defining the faces need positive orientation).

```
:hidd(m3,pro,n)
:Prgm
:Local m,l,i,test
:@ Testroutine with Determinant
:Define test(mm,k)=Func
:(mm[k+2,1]-mm[k+1,1])*(mm[k,2]-mm[k+1,2
 ])-(mm[k+2,2]-mm[k+1,2])*(mm[k,1]-mm[k+
 1,1])
:EndFunc
:
:@ Start MainProg
:@ Transfer 3D-Points to 2D
:m3*pro→m
:
:For l,1,rowDim(m),n
:If test(m,l)>0 Then
:  For i,0,n-2,1
:  Line m[l+i,1],m[l+i,2],m[l+i+1,1],m[l
 +i+1,2],1
:  EndFor
:EndIf
:EndFor
:EndPrgm
SOLIDS          RAD AUTO          FUNC
```

On the defined solids you can apply appropriate projections. Multiplications of matrices can be performed without any problems.

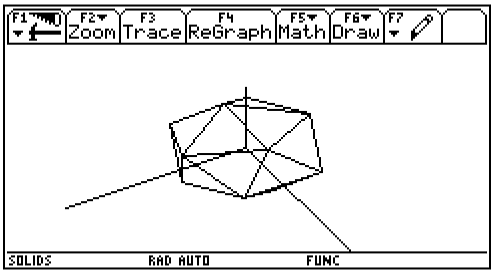
If you want to translate the solid you should add a constant vector v of same dimension to each row body[i] of the matrix. On the TI-92 PLUS such program works, the ordinary TI-92 produces a Domain Error, although a singular calculation body[i]+v can be performed on the Home-Screen.

Again the solution is an addition element per element which can easily be followed in the program trans(body,vec): The solid body is translated by the vector v.

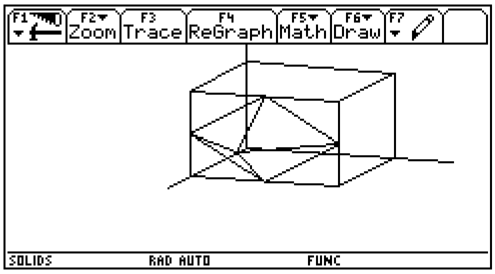
```
:axes(xm,ym,zm,pro)
:Prgm
:Local m
:[[xm,0,0][0,0,0][0,ym,0][0,0,0][0,0,zm]
 ]*pro→m
:plotm(m)
:EndPrgm
```

```
:trans(m,v)
:Func
:Local i,k
:For i,1,rowDim(m),1
:  For k,1,colDim(v),1
:    m[i,k]+v[1,k]→m[i,k]
:  EndFor
:EndFor
:Return m
:EndFunc
```

Program axes(xm,ym,zm,pro) represents the axes. Parameters are the oriented lengths of the three axes and the projection matrix in use.




Ikosaeder in "Military Perspective"




Cube together with a translated Octahedron in dimetric projection

```
■ axes(7,7,7,mil2)                       Done
■ hidd(icosa(5),mil2,4)                  Done
■ axes(7,7,7,dime)                       Done
■ hidd(cube(5),dime,5)                   Done
■ hidd(trans(okta(5),[2.5  0  0]),dime,4)
                                         Done
```

Using all these tools one can do a lot in Linear Algebra. Working with nonregular solids you will miss *DERIVE's* wonderful VECTOR-command.

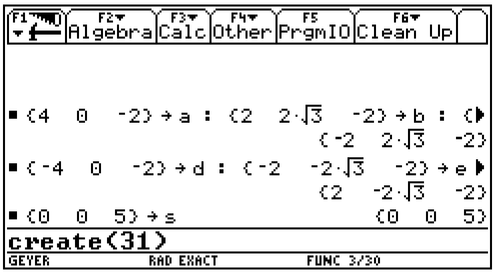
The projection matrices used here are:

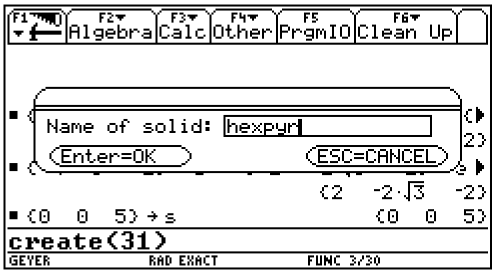
$$kav1 = \begin{bmatrix} -1/2 & -1/2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad mil1 = \begin{bmatrix} -\sqrt{3}/2 & -1/2 \\ 1/2 & -\sqrt{3}/2 \\ 0 & 1/5 \end{bmatrix}; \quad mil2 = \begin{bmatrix} -\sqrt{3}/2 & -1/2 \\ 1/2 & -\sqrt{3}/2 \\ 0 & 1/2 \end{bmatrix}$$

**References:**

Hubert Weller: Polyhedrons – their Representation, DNL#27
Hubert Weller: Platonic Solids – A Hidden Lines Algorithm, DNL#29
Richard Schorn: C60 – The Buckyball, DNL#21
P Familiar Ramos: POLIEDROS 1.0, DNL#24

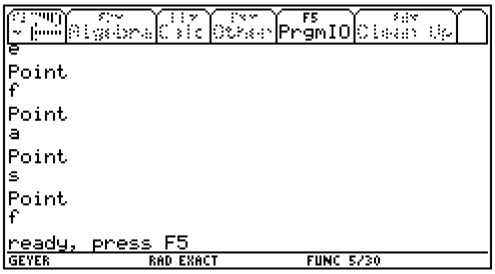
Heinz Rainer wrote about nonregular solids. I faced that challenge and modified - extended – his programs for presenting other solids on the TI-screen.
Let´s take a pyramide with a hexagonal base.
The base is defined by six points (A,B,C,D,E,F). We first enter each single point as a list consisting of its three coordinates, followed by the vertex S(0,0,4):

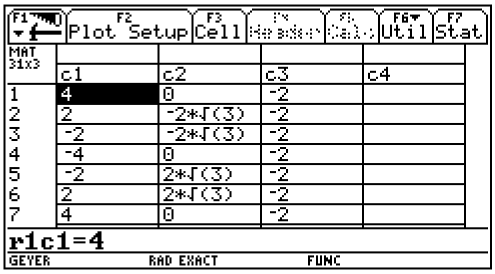


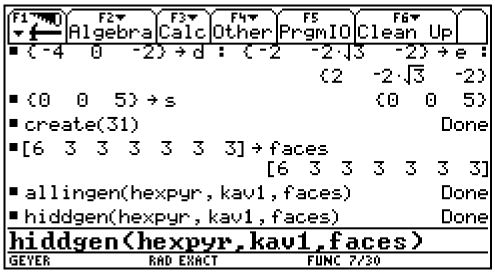

As Heinz Rainer mentioned before editing a matrix consisting of 7 + 6 * 4 = 31 points is very boring. I use the program `create(number_of_points)` to produce the matrix, describing the pyramid:

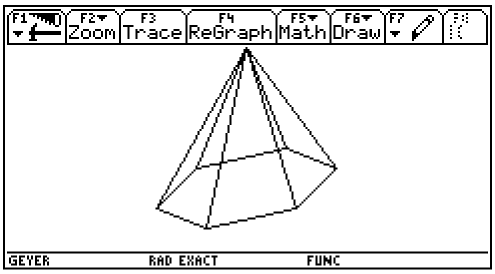
Take care of the order of the points and enter 31 points:
A,F,E,D,C,B,A, A,B,S,A, B,C,S,B, C,D,S,C, D,E,S,D, E,F,S,E, F,A,S,F

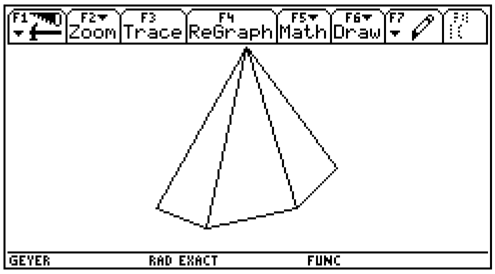



If you want to change some values, then open matrix `hexpyr` with the Matrixeditor and perform your changes. Define a vertex-vector with the numbers of the faces´ vertices and then call
`allingen(body_name, projection, v_vector)` or
`hiddgen(body_name, projection, v_vector)`.



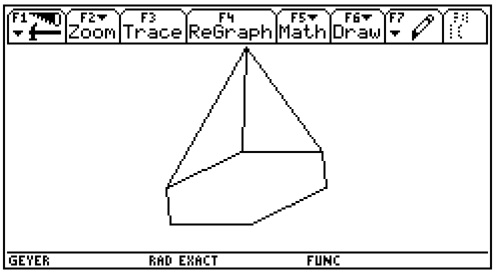





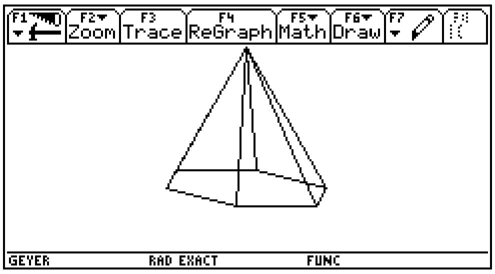
Try some other projections:

$$\text{iso} = \begin{bmatrix} -1 & -1/2 \\ 1 & -1/2 \\ 0 & 1 \end{bmatrix};\quad \text{or oblique view } \text{obl}(sf, \Phi) = \begin{bmatrix} 1 & 0 \\ sf\cdot\cos(\Phi) & -sf\cdot\sin(\Phi) \\ 0 & 1 \end{bmatrix}$$

**The TI-programs:**



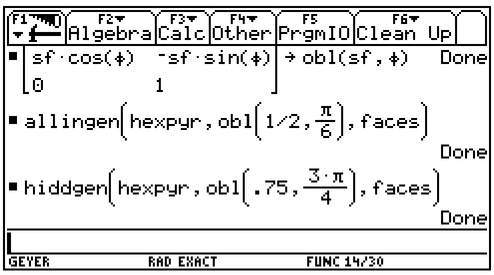

```
create(num)
Prgm
Local j_,num,bo,pt
Request "Name of solid",nams
{}→bo
ClrIO
For j_,1,num
InputStr "Point",pt
augment(bo,expr(pt))→bo
EndFor
list▸mat(bo,3)→bo
bo→#nams
Disp "ready, press F5"
EndPrgm
```

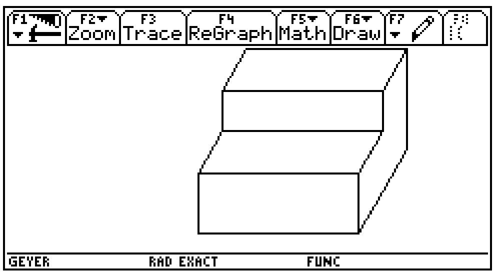



```
hiddgen(m3,pro,f)
Prgm
ClrDraw
Local  m,l,i,j,st,test,k_,fla
© Testroutine with Determinant
Define  test(mm,k)=Func
(mm[k+2,1]-mm[k+1,1])*(mm[k,2]-mm[k+1,2])-(mm[k+2,2]-mm[k+1,2])*(mm[k,1]-
mm[k+1,1])
EndFunc

ClrIO
© Start MainProg
© Transfer 3D-Points to 2D
m3*pro→m:1→st:dim(f)[2]→fla
For  k_,1,fla,1
If test(m,st)>Ø Then
  For  i,Ø,f[1,k_]-1,1
   Line  m[st+i,1],m[st+i,2],m[st+i+1,1],m[st+i+1,2],1
  EndFor
EndIf
st+f[1,k_]+1→st
EndFor
EndPrgm
```

For `allingen(m3,pro,f)` you have only to remove the hidden line test.
I add another object stairs (produced with a vector fstair).

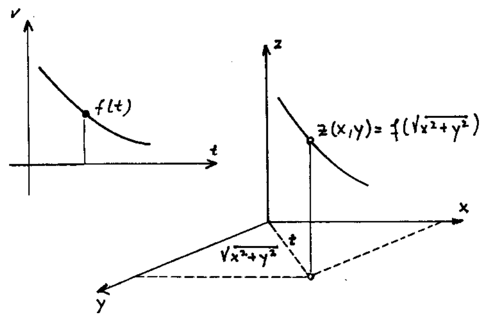


On many occasions I am asked how to produce 3D-pictures of solids of revolution on the TI´s 3D-screen. Students often show an enormous deficiency in their 3D-imagination. Producing their own representations might improve their visualization of 3D-solids. The procedure is very easy to perform. A simple sketch describes how to do it.

Let´s take any function $v = f(t)$ which should be rotated around the $v$-axis. Then the $v$-axis changes to the $z$-axes and argument $t = \sqrt{x^2 + y^2}$ . Function $z(x, y) = f(\sqrt{x^2 + y^2})$ describes the solid.

For an easy example we want to rotate the parabola $y = \frac{x^2}{2}$ around the $y$-axis.




1. Step:

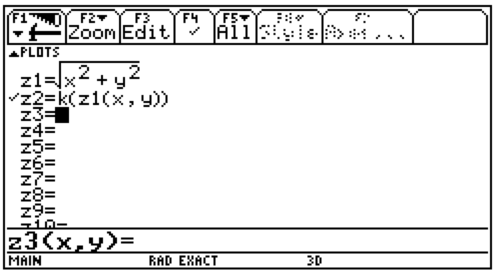
In the Home Screen define function

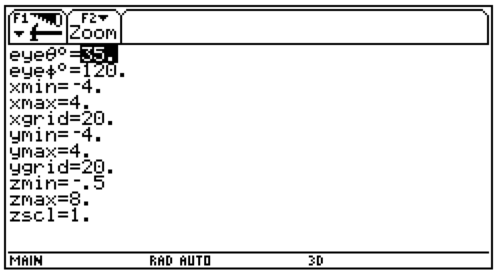
x^2/2 → k(x).

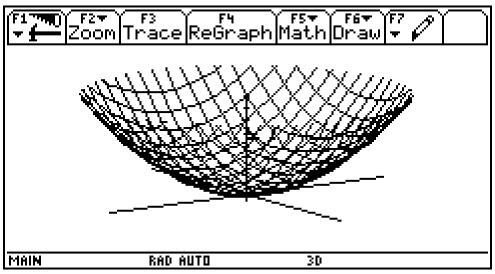
2. Step: Switch the Graph Mode to 3D.

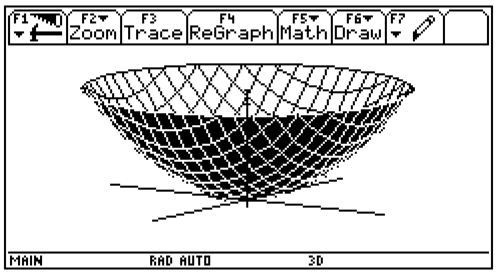
3. Step:

Define in the #-Editor the auxiliary function z1(x,y) and then the surface z2(x,y). Set the ∃-values. Press % and wait for the graph.



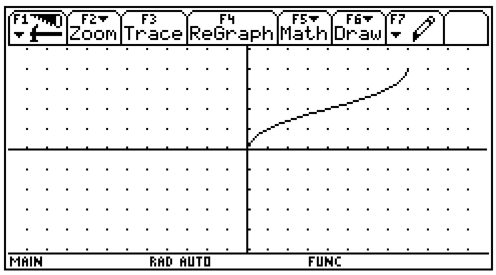



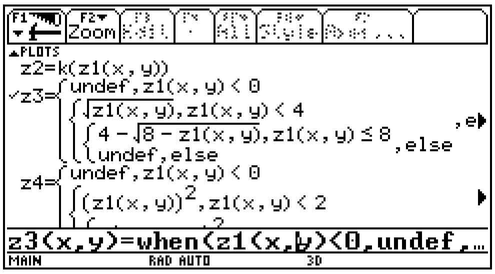

The next example is more difficult: we take a piecewise defined function and want to produce the solids resulting from rotations around both axes.

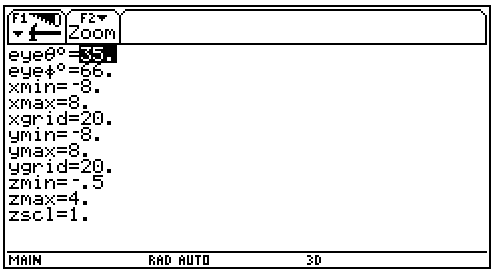
The function is composed of two parabolas:

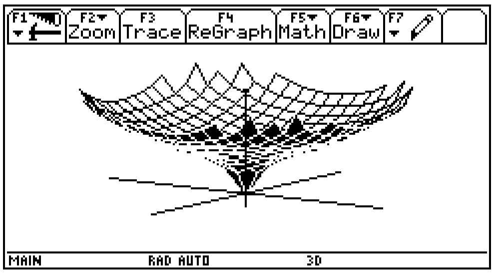
$$y = \begin{cases} \sqrt{x} & \text{for } 0 \le x < 4 \\ 4 - \sqrt{8 - x} & \text{for } 4 \le x \le 8 \\ \textit{undefined} & \text{else} \end{cases}$$




```
when(x<Ø, undef,when (x < 4, √(x), when (x ≤8,4 - √(8-x),undef)))
```



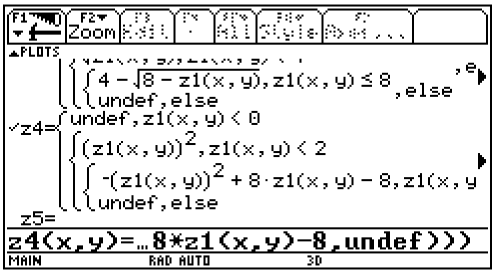



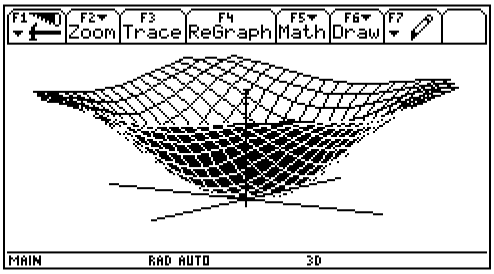

Rotating the curve around the x-axis should result in a bell-shaped surface.

In the #-Editor edit function z3(x,y) according to the function given above , replacing x by z1(x,y).

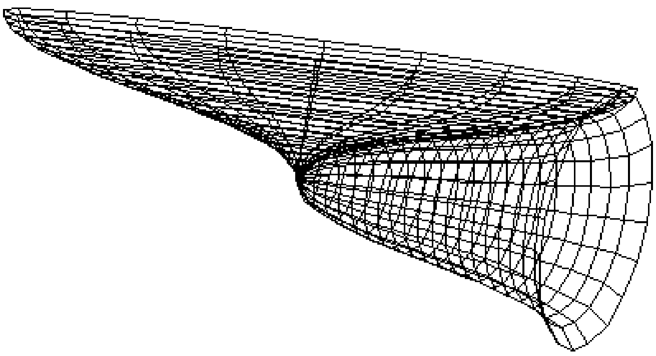
Take care of the ∃-settings and plot the graph.

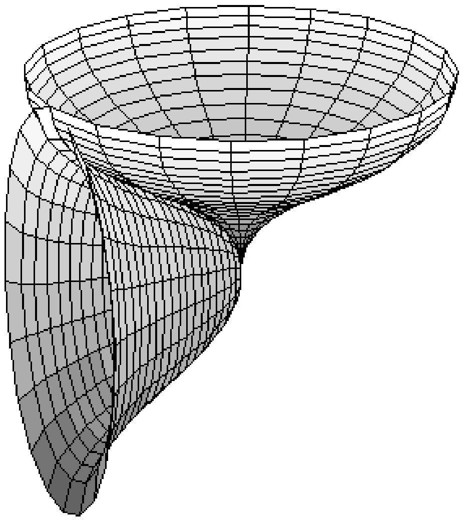
As this is not a function, we must use a trick: we rotate the inverse function around the y-axis to obtain a visualization of the bell. We also have to adapt the ∃-settings. Take into account that the scaling is not the same in both windows. So the bell doesn´t look as slim as it should.



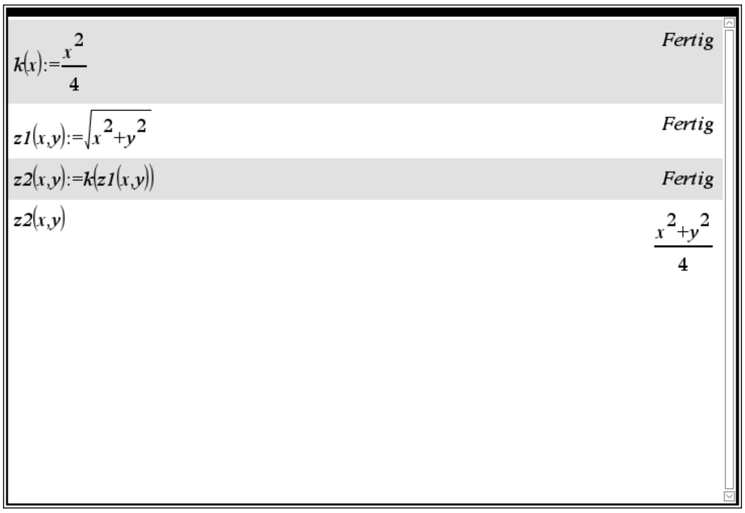

Finally let me show, how the "Chinese Hat" and the "Bell" appear in DERIVE.

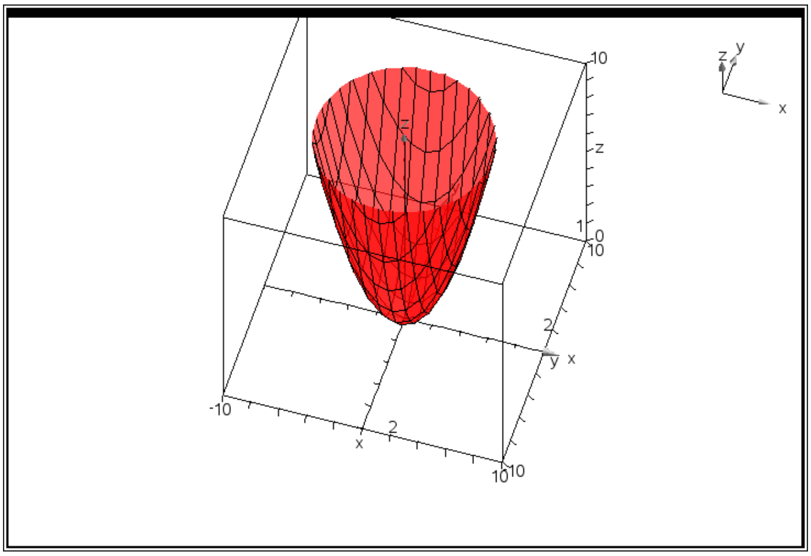




The right picture is produced with an α-version of DERIVE 5, which was presented by David Stoutemyer and Theresa Shelby at the ACDCA-Summer Academy. It should be an appetizer for the new DERIVE-age.

Let's try to present the solids of revolution on the TI-Nspire CX CAS screen. The rotation of the parabola is an easy job – the procedure performed on the TI-92/Voyage 200 can be transferred on the Nspire without any problems:

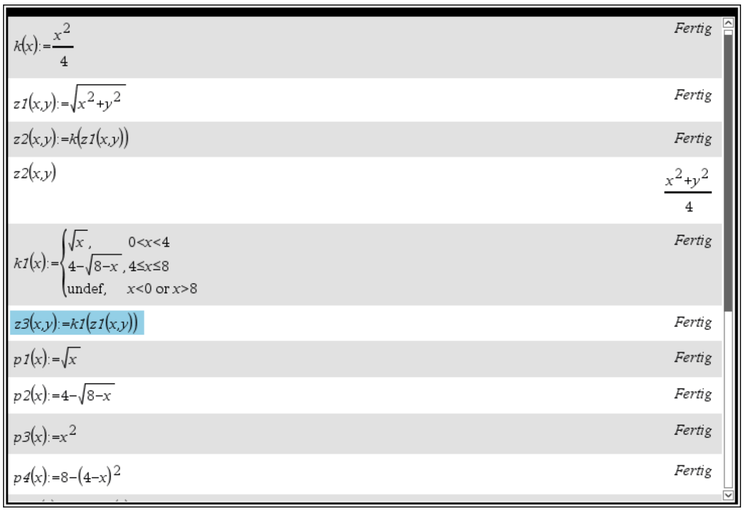


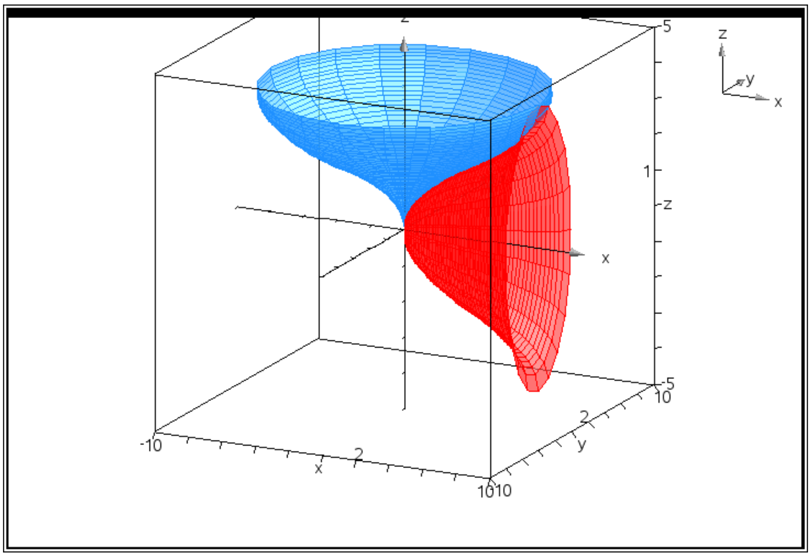

I was not able to repeat the procedure with the piecewise defined function – the Chinese Hat and the Bell.

I failed with all my tries and efforts.

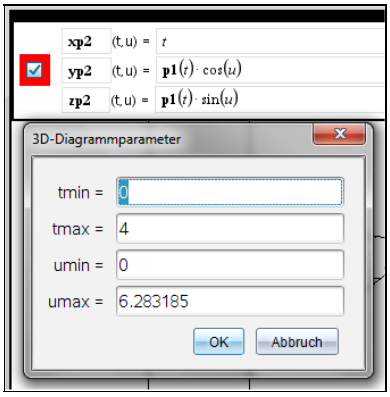
If anybody of you will have more success, then please keep me informed.




Next experiment was to define the two parts of both solids in parameter form. This works as you can see on the next screen shot.

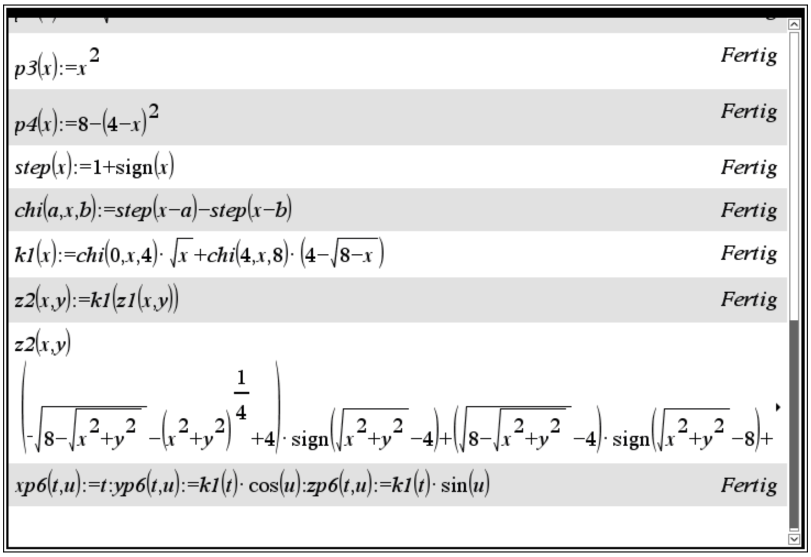



p1($t$) and p2($t$) form the red solid and p3($t$) with p4($t$) form the blue one.




Using an idea of Michel Beaudin and his colleagues from Montreal we can "invent" *DERIVE*'s CHI-function for defining a piecewise defined function.

Then we can use this function which is called $k1(x)$ in my screen to plot the $z2(x,y)$ created solid (rotation around the $z$-axis) and the parameter form created solid (rotation around the $x$-axis) using only one function.



$p3(x):=x^2$ Fertig

$p4(x):=8-(4-x)^2$ Fertig

$step(x):=1+\text{sign}(x)$ Fertig

$chi(a,x,b):=step(x-a)-step(x-b)$ Fertig

$k1(x):=chi(0,x,4)\cdot\sqrt{x}+chi(4,x,8)\cdot\left(4-\sqrt{8-x}\right)$ Fertig

$z2(x,y):=k1(z1(x,y))$ Fertig

$z2(x,y)$

$\left(\left(-\sqrt{8-\sqrt{x^2+y^2}}-\left(x^2+y^2\right)^{\frac{1}{4}}+4\right)\cdot\text{sign}\left(\sqrt{x^2+y^2}-4\right)+\left(\sqrt{8-\sqrt{x^2+y^2}}-4\right)\cdot\text{sign}\left(\sqrt{x^2+y^2}-8\right)+\right.$

$xp6(t,u):=t{:}yp6(t,u):=k1(t)\cdot\cos(u){:}zp6(t,u):=k1(t)\cdot\sin(u)$ Fertig