THE DERIVE - NEWSLETTER #20

ISSN 1990-7079

THE BULLETIN OF THE

DERIVE®

USER GROUP

Contents:

revised Version 2009 December 1995

[1] **DERIVE-Projekte im Unterricht,** MU, Jahrgang 41, Heft 4/95
Friedrich Verlag, 30917 Seelze

[2] **Materialien zum M-Unterricht mit Computer und *DERIVE*,**
Landesmedienzentrum in Rheinland-Pfalz, 1995, Hofstr. 257, 56077 Koblenz

[3] **Learning Linear Algebra through *DERIVE***, Brian Denton,
Prentice Hall, 1995, ISBN 013 122664 9, 351 pages

[4] **Business Calculus today with Spreadsheets and *DERIVE***, R. L. Richardson,
Saunders College Publishing, 1996, ISBN 0 03 017554 2, 416 pages

[5] **Resources by Discovery,** MAA Notes 27-31, The Mathematical Association of America

[6] **transferts,** Cahiers de la Cellule Recherche Innovations Pédagogiques, Numero 7
Derive: Utilisation d´un logiciel de calcul formel
C.R.I.P. Rectorat de l´académie de Lille, 20 rue Saint Jacques, 59000 Lille, France

Dominique Lymer, one of the authors and a DUG member since long sent one copy (170 pages). It is an interesting collection of work sheets for students with accompanying support material for teachers. Merci, Dominique. Dominique has submitted a contribution which is one of my favourites for the next issue.

[7] **Matemáticas con DERIVE en la economia y la empresa,** Alfonso Gonzáles, Pareja,
ra – ma, ISBN 84 7897 201 3

Exchange for DERIVE Teaching materials in the DNL
The wheel has not to be invented twice.

Börse für DERIVE Unterrichts-materialien im DNL
Das Rad muss nicht zweimal erfunden werden.

I can offer:

Binomial Theorem, GCD & LCM, System of Coordinates (in English and in German as well), Modelling Word Problems with *DERIVE*

DERIVE Days Leeds

Spring School on Teaching and Learning
Mathematics with DERIVE and the TI-92

Trinity and All Saints College, Leeds
13 – 15 April 1996

For receiving an announcement please send a self-addressed A4 envelope to:
John Monaghan, Centre for Studies in Science and Mathematics Education,
University of Leeds, Leeds LS2 9JT

Liebe DUG-Mitglieder,

Gleich zu Beginn möchte ich uns allen zum fünfjährigen Bestehen der DUG gratulieren. Mit dem ersten Mitglied aus Namibia, das ich recht herzlich in unserem aller Namen begrüße, sind nun alle fünf Kontinente in der DUG vertreten.

Für das Jahr 1996 steht eine interessante Ausweitung des DNL in Aussicht. Bert Waits von der Ohio State University (OSU) und Bernhard Kutzler von SWHE, zwei TI-92 Spezialisten, haben angeregt, eine TI-92 Kolumne in den DNL aufzunehmen. Sie haben auch versprochen, die ersten Beiträge zu liefern und Anfragen zum TI-92 zu beantworten. Das „klassische" DERIVE für den PC wird dabei weiterhin nicht zu kurz kommen.

Die Beiträge dieser Ausgabe sind schwerpunktmäßig den Differentialgleichungen gewidmet. Das ist vielleicht für manche von Ihnen zu einseitige Kost. Ich kann Ihnen Abwechslung ankündigen: 1996 wird es einen ausgesprochen sportlichen DNL geben: eine Projektarbeit über das Schispringen, eine Geometrie des Fußballs, Baseball, ein Vergleich von sportlichen Leistungen und vielleicht ein Beitrag „DERIVE und das Tennisnetz" versprechen allerhand.

Als kleines Weihnachtsgeschenk finden Sie unter den Dateien im Unterverzeichnis <STEREO> einige Stereogramme, an denen Sie Ihren „magischen Blick" testen können. Laden Sie die Bilder in ein Bildbetrachtungsprogramm (zB IRFANVIEW). Sie lassen sich ausdrucken oder am Bildschirm betrachten. Ich wünsche Ihnen viel Spaß damit. Die Bilder wurden mit POPOUT-LITE erstellt.

(Geom. Figuren, Geckos+DUG, Geckos, SWHE-Logo, DERIVE in Rahmen und Gecko+DERIVE)

Bitte vergessen Sie nicht, Ihre Mitgliedschaft rechtzeitig zu erneuern, wenn Sie der DUG die Treue halten wollen.

Damit bleibt mir nur noch, Ihnen und Ihrer Familie in meinem und im Namen meiner Frau alles Schöne für das bevorstehende Weihnachtsfest und ein glückliches Neues Jahr 1996 zu wünschen.

Herzliche Grüße

Dear DUG Members,

Let me start with a congratulation: It is your merit that we can celebrate the 5th year of the existence of the DUG. As we have now a member from Namibia who joined us two months ago we can really say that the DUG is represented on all five continents. We will express our warm welcome.

For 1996 I am looking forward to having an interesting enrichment in the DNL. Bert Waits form the Ohio State University (OSU) and Bernhard Kutzler (SWHE) – two TI-92 specialists –suggested including a TI-92 column in the DNL. They both have promised to submit the first contributions and to answer TI-92 related requests.

Some contributions of this issue focus on differential equations. I'd like to ask our friends who are not involved so much in these applications for patience. I can promise for 1996 a really "sportive" DNL: a student's project on ski jumping, geometry of the soccer ball, base ball, a comparison of sportsmen's performances and hopefully an article "DERIVE and the tennis net".

Among the files you can find a small Christmas gift. There are some stereograms in the folder <STEREO>. Try your "magic eyes" by loading them into any graphics program (e.g. IRFANVIEW). You can either print the pictures or inspect them on the PC-screen. I used POPOUT-LITE to create the pictures.

(Geometric shapes, geckos+DUG, geckos, SWHE-Logo, DERIVE in frames and gecko+DERIVE).

Please don't forget renewing your membership for 1996.

My wife and I wish you and your family the best for Christmas and a Happy New Year 1996

Best regards

I met Joe Fiedler at a TI-92 workshop in Houston. Unfortunately I didn't realize that Joe has been a DUG member, so I didn't introduce myself. Joe, you were great. I enjoyed your and Wade Ellis´ workshop immensely. I had many nice hours at the ICTCM including the DUG-meeting. But this workshop was one of the top events for me. Many thanks.

The *DERIVE-NEWSLETTER* is the Bulletin of the *DERIVE User Group*. It is published at least four times a year with a contents of 30 pages minimum. The goals of the *D-N-L* are to enable the exchange of experiences made with *DERIVE* as well as to create a group to discuss the possibilities of new methodical and didactical manners in teaching mathematics.

Editor: Mag. Josef Böhm
A-3042 Würmla
D´Lust 1
Austria
Phone: 43-(0)2275/8207

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 *D-N-L*. It must be said, though, that non-English articles are very welcome nonetheless. Your contributions will be edited but not assessed. By submitting articles the author gives his consent for reprinting it in *D-N-L*. The more contributions you will send, the more lively and richer in contents the *DERIVE Newsletter* will be.

Preview: (Contributions for the next issues):

Graphic Integration, Probability Theory, Linear Programming, Böhm, A
LOGO in DERIVE, Lechner, A
DREIECK.MTH, Wadsack, AUS
IMP Logo and Misguided Missiles, Sawada, HAWAII
3D Geometry, Reichel, AUS
Parallel- and Central Projection, Böhm, AUS
Vector and Vector Indices Sorting, Biryukov, RUS
Algebra at A-Level, Goldstein, UK
Tilgung fremd erregter Schwingungen, Klingen, GER
Utility for Complex Dynamic Systems, Lechner, A
Notes on DERIVE 2.6 functions and limits, Speck, NZL
Ski Jump, a project with students, Scheuermann, GER
Linear Mappings and Computer Graphics, Kümmel, GER
Julia Sets, Kümmel, GER
Solving Word Problems with DERIVE, Böhm, AUT
and

Setif, FRA; Vermeylen, Belgium; Lymer, FRA; Leinbach, USA, Aue, GER; Weth, GER, Wiesenbauer, AUT; Keunecke, GER, Weller, GER, …

and messages from the derive-news@mailbase.ac.uk

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

Glynn D Williams, Gwynedd, UK

Dear Sir,

I have used **DERIVE 3.0** for just over a year now, and find that it is a significant improvement over the previous versions, both in the functions available and in speed of operation. But I feel, that there are still some loose ends which need tidying up:

1. It seems that the configurable menu system was added as an afterthought, because no error checking is done on the menu items to see that

   a) the menu tree is workable, with all brackets and quotation marks correctly matched up;

   b) all the menu items on a particular branch of the menu tree have distinct hotkeys;

   c) there is a way out, using **Quit** or its alias.

   Any of the above kinds of error can cause the program to simply hang up, requiring a reboot. This is unacceptable: errors should be indicated, with a chance to escape. I use a modified menu myself, because I like **Substitute** and **Renumber** (frequently wanted, but buried two levels down the menu-tree!) to be one-key operations; I also like **Save**, **Load** and **Merge** to be near the top of the menu.

2. There are some bugs in the code: `factor ([])` returns `[1]` instead of `[]`, making this function unworkable when used with the `factors( )` function. Of course one could write one's own `factor( )` function e.g. `myfactor(a):=if(a=[],[],factors(a))` to squash the bug, but it is messy, and such "patches" should be unnecessary.

3. The mouse could be usefully employed to select areas of graphs or expressions to be copied, pasted, or zoomed.

4. One area which needs to be attended urgently is the presentation of brackets in multi-line expressions. These are currently always represented as square; this can cause ambiguity if the function definition contains vectors or if the header accepts them as parameters. The Newsletter suffers from this; it is sometimes difficult to decide whether a bracket or a parenthesis should be keyed when typing in code. The explanation in the manual is that this is unavoidable because of the limitations of the IBM-compatible character-set.

5. How about a proper ring- or spiral-bound manual, so that one can open the book out flat: one has both hands in use when working at a keyboard, and a book which tends to close is a nuisance in these circumstances?

Yours faithfully Glyn D Williams

DNL: I sent your ideas to Soft Warehouse and I am sure you will receive an answer. In my *DERIVE* versions 3.04 and 3.06 `factor ( [] )` returns `[]` .Concerning point 4. of your complaints I must admit that you are right: two souls are fighting in my breast. The expressions written in *DERIVE* syntax are easy to type in, but they often don't represent the mathematical contents as clear as wanted. I will pay more attention to this aspect in the future. Maybe it will be possible to "mix" the two forms of representing *DERIVE* expressions.

Martin Lindsay, Footscray Campus, Melbourne, Australia

I am a beginning mathematics education PhD student who is interested in using DERIVE as a project for my research. I teach upper secondary / lower tertiary students (17 -21 year - olds) mainly precalculus and calculus topics. I would like to hear from anyone who has come across a concept (or topic) which can be investigated from an educational point of view, and which has, in their opinion the potential to contribute to our understanding of how students learn mathematics via a computer.

Regards. Martin Lindsay. e-mail: Martin Lindsay@vut.edu.au; FAX: ++6136884050

DNL: I sent some own materials to Martin and I hope that there are some other DUG members who will support his work. Are there any suggestions for projects? It could be interesting to compare projects done in different countries und under various circumstances and conditions. Hopely we will hear about results. Additionally I can recommend the **Resources for Calculus Collection** from the MAA, which I bought in Houston last week and **transferts** (in French). You will find more information in the **Book Shelf**.

In the last DNL you could find an interesting request about a square root simplification. I received two respective answers. one from the Fachhochschule Osnabrück and another one from Albert Rich, SWHH, which explains the issue.

FHS Osnabrück

The fact described by Mr. Pröpper in DNL#19 has more odd aspects. If one having defined the interval [-1,1) for *x*, tries to simplify expression #1 he will obtain the expected result (#5). Only expression #2 (derived by DERIVE) will not simplify. Although DERIVE recognizes #1 and #3 as equal, they are treated differently when they are simplified.

The next example shows the same: the declaration of the interval –1 < *x* 1 is considered in simplifying expressions #1, #9 and #11. Simplifying #13 (included #2) fails again.

#1: $\sqrt{(1+x)\cdot(1-x)}\cdot\sqrt{\left(\dfrac{1+x}{1-x}\right)}$ User

#2: $\sqrt{\left(\dfrac{x+1}{1-x}\right)}\cdot\sqrt{(x+1)\cdot(1-x)}$ Simp(#1)

#3: $\sqrt{(1-x^2)}\cdot\sqrt{\left(\dfrac{1+x}{1-x}\right)}$ Expd(#1')

#4: x :∈ Real [-1, 1) User

#5: x + 1 Simp(#1)

#6: $\dfrac{\sqrt{(1-x^2)}\cdot\sqrt{(x+1)}}{\sqrt{(1-x)}}$ Simp(#3)

```
      F(x) :=
        If x > 2
#7:        "false"                                            User
           "true"
           "unknown"

#8:   F(x) := true                                            Simp(#7)
```

```
       G(x) :=
         If x^2 > 2
#9:         "false"                                                 User
            "true"
            "unknown"

#10:   G(x) := true                                                 Simp(#9)
```

#11: $u := \mathrm{IF}\left(\sqrt{((1 + x)\cdot(1 - x))}\cdot\sqrt{\left(\frac{1 + x}{1 - x}\right)} > 3, \text{false}, \text{true}, \text{unknown}\right)$ User

#12: u := true Simp(#11)

#13: $v := \mathrm{IF}\left(\sqrt{(1 - x^2)}\cdot\sqrt{\left(\frac{1 + x}{1 - x}\right)} > 3, \text{false}, \text{true}, \text{unknown}\right)$ User

#14: v := unknown Simp(#13)

Albert Rich #1

Dear Josef,

Enjoyed reading the DUG Newsletter #19. You have surpassed the 18 Newsletters Soft Warehouse published back in the muMATH days!

In your response to Wolfgang Pröpper´s question concerning the simplification of radicals you wondered why DERIVE did not simplify his example to 1 + x even if x is declared an element of [-1,1). The reason is that DERIVE does not simplify SQRT(1 – x^2) to

SQRT(1 + x) * SQRT(1 – x)

because this requires rational factoring of 1 – x^2. DERIVE only tries square-free factoring on the argument of radicals because rational factoring can take a very long time (e.g. try rational factoring 1000 x^4 + x^3 + 1323).

To properly resolve the problem, I will teach DERIVE that factoring the difference of two squares is easy to do. (DERIVE is an obedient student but it has to be taught each and every detail!)

On another subject, your readers may be interested in the following recent additions to the utility file NUMBER.MTH:

The function CONTINUED_FRACTION(u,n) approximates to a vector of n+1 partial quotients of the continued fraction of u. For example,

```
CONTINUED_FRACTION(#e,8)
```

approximates to

```
[2,1,2,1,1,4,1,1,6]
```

If question marks (?) appear in the result, use the Options Precision command to increase the precision. CONTINUED-FRACTION is defined as follows

```
CONTINUED_FRACTION(u, n) := FLOOR(ITERATES(1/MOD(x_), x_, u, n))
```

The function PARTITIONS(n) simplifies to the number of decompositions of n integer summands without regard to order. For example, 4 = 1 + 3 = 2 + 2 = 1 + 1+ 2 = 1 + 1+ 1+ 1 so PARTITIONS(4)= = 5. The following definition of PARTITIONS was contributed by James FitzSimons:

```
PARTITIONS_AUX(n,m,mn):=
       IF(m=1,1,SUM(PARTITIONS_AUX(n-k_,m-1,k_),k_,mn,FLOOR(n,m)))

PARTITIONS(n):=SUM(PARTITIONS_AUX(n,m,1),m,1,n)

PARTITIONS(4)=4
```

Unfortunately, for n greater than about 50 the above functions take too long because of explosive fan-out on the recursive calls to PARTITIONS_AUX: Are there any DUG members who can come up with an efficient definition for PARTITIONS that Soft Warehouse can include in NUMBER.MTH? Aloha.

(Are there? Hannes and others, this is a real challenge! I have put these functions and the next ones together in one file NUMBEXT.MTH. You can merge this file to NUMBER.MTH and so improve your NUMBER.MTH. Josef)

Albert Rich #2

Dear Josef,

Enclosed is an article that appeared in a recent Science News about consecutive prime numbers that you might find interesting. I thought that these seven 97-digit primes would make a good test of DERIVE's NEXT_PRIME function. The enclosed printout shows that DERIVE does correctly recognize primes of this size.

Since my October 16, 1995 letter we have simplified and renamed the function for computing the partitions of a number. This is how my comment in this letter should read:

PARTS(n) simplifies to the number of decompositions of n into integer summands without regard to order. For example, 4 = 1 + 3 = 2 + 2 = 1 + 1+ 2 = 1 + 1 + 1 + 1 so PARTS(4) = 5.

```
PARTS_AUX(n,m)≔IF(n<2m,1,1+∑(PARTS_AUX(n-k_,k_),k_,m,FLOOR(n,2)))

PARTS(n)≔IF(n<1,0,PARTS_AUX(n,1))

PARTS(4)=5

PARTS(50)=204226
```

(needs 8.05 sec)

Recent Derive versions have another PARTS-function implemented. It looks very bulky but it is much more efficient than the earlier one - and it works without the auxiliary function:

```
PARTS(n)≔IF(n<2,1,FLOOR(APPROX(∑(-1/π√(k_/2)∑(IF(GCD(h_,k_)=1,
   COS(π(∑((i_/k_-1/2)(MOD(i_h_/k_)-1/2),i_,1,k_-1)-2nh_/k_)),0),
   h_,1,k_)2√6e^(-π√(24n-1)/(6k_))(e^(π√(24n-1)/(3k_))
   (6k_-π√(24n-1))-6k_-π√(24n-1))/(k_(24n-1)^(3/2)),
   k_,1,√n/LOG(n,11)),LOG(1/(4n√3)e^(π√(2n/3)),10)+5)+1/2))
PARTS(50)=204226
```

(needs 0.000 sec!! Josef)

DISTINCT_PARTS(n) simplifies to the number of decompositions of n into distinct integer summands without regard to order.
For example, 4 = 1 + 3 so DISTINCT_PARTS(4) = 2.

```
DISTINCT_PARTS_AUX(n,m)≔IF(n<2·m,1,1+∑(DISTINCT_PARTS_AUX(n-
k_,k_+1),k_,m,FLOOR(n,2)))

DISTINCT_PARTS(n)≔IF(n<1,0,DISTINCT_PARTS_AUX(n-1,1))

DISTINCT_PARTS(100) = 444793
```

(needs 16.5 sec)

The function which is implemented now works also without any auxiliary function. Compare the calculation time.

```
DISTINCT_PARTS(n)≔∑(IF(n-16i_+1=FLOOR(√(8n-16i_+1))^2,PARTS(i_),0),
                         i_,0,n/2)

DISTINCT_PARTS(100)=444793
```

(needs 0.047 sec)

All for now,

Aloha, Albert D. Rich, Applied Logician

(Albert's letter closes again with his challenge to improve these functions. Find now the included file for Harvey Dubner's first of seven consecutive primes separated by 210. That is a new record, the previous had been numerous examples involving six consecutive primes in arithmetic progression. Albert refers to an article from SCIENCE NEWS, vol. 148, September 1995. The article starts "Searches for patterns among prime numbers have long served as stiff tests of the ingenuity and perseverance of mathematicians." You can obviously see that not only our column writer Johannes Wiesenbauer is dealing with prime numbers. H. Dubner and H. Nelson ended up using seven computers, running continuously for about 2 weeks, to find the sequence. Now they are thinking about going to eight consecutive primes. They estimate that it would take 20 times longer - at least 2.5 computer years - to accomplish this search on their souped-up personal computers. Josef)

```
"Harvey Dubner's first of 7 consecutive primes separated by 210:"

p:=108953343124705931087578037892295773290803649299313819538521
   31055617421504473089672131417174861 51

"The following verifies the difference of the 7 consecutive primes"

"and that DERIVE's primality is working good:"

v:=ITERATES(NEXT_PRIME(n),n,p,8)

"You might simplify #5 to see the 8 huge prime numbers"

VECTOR(v SUB (n_+1)-v SUB n_,n_,DIMENSION(v)-1)

[210,210,210,210,210,210,120,52]
```

Playing-Cards Shuffling with *DERIVE*

Benno Grabinger, Neustadt, Germany

In DNL#12 Mr Chuan from Taiwan posed the following question:

> Is it a surprise that after perfectly shuffling only 8 times a deck of 52 cards, the original position of the cards is restored?

With the help of *DERIVE* it is no problem to understand what is going on. The perfect shuffling is described by the permutation

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & \dots & 49 & 50 & 51 & 52 \\ 1 & 27 & 2 & 28 & \dots & 25 & 51 & 26 & 52 \end{pmatrix}.$$

It is well known that each permutation is a product of cycles (having no common elements). The function **CYCLE(p)** creates a product of cycles which represents the permutation p:

Applying this function to the given permutation p leads to:

[[1], [27, 14, 33, 17, 9, 5, 3, 2], [28, 40, 46, 49, 25, 13, 7, 4], [29, 15, 8, 30, 41, 21, 11, 6], [31, 16, 34, 43, 22, 37, 19, 10], [32, 42, 47, 24, 38, 45, 23, 12], [35, 18], [36, 44, 48, 50, 51, 26, 39, 20], [52]]

It can easily be seen that this product consists of 6 cycles of length 8 and one cycle of length 2. Therefore, if p has been applied 8 times the original distribution will be restored.

```
#1:   CYCLE.MTH by Benno Grabinger, 1995

#2:   n :≡ 52

#3:   u :≡ VECTOR(k, k, 1, n)

#4:   v :≡ VECTOR(IF(MOD(i, 2) = 0, i/2 + n/2, (i + 1)/2), i, 1, n)

#5:   p :≡ [u, v]

#6:   [ 1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
        1  27   2  28   3  29   4  30   5  31   6  32   7  33   8  34   9  35  10  36  11

       22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
       37  12  38  13  39  14  40  15  41  16  42  17  43  18  44  19  45  20  46  21

       42  43  44  45  46  47  48  49  50  51  52 ]
       47  22  48  23  49  24  50  25  51  26  52 ]

#7:   VALUE(i, p) :≡ ELEMENT(ELEMENT(p, 2), i)

#8:   Z(s, p) :≡ DELETE_ELEMENT(ITERATES(VALUE(j, p), j, s), 1)

      EQUAL(v, w) :≡
        If DIMENSION(w) = 0
           0
#9:        If ELEMENT(v, 1) = ELEMENT(w, 1)
              1
              EQUAL(v, DELETE_ELEMENT(w, 1))

      IN(v, z) :≡
        If DIMENSION(v) = 0
           0
#10:       If EQUAL(ELEMENT(v, 1), z) = 1
              1
              IN(DELETE_ELEMENT(v, 1), z)
```

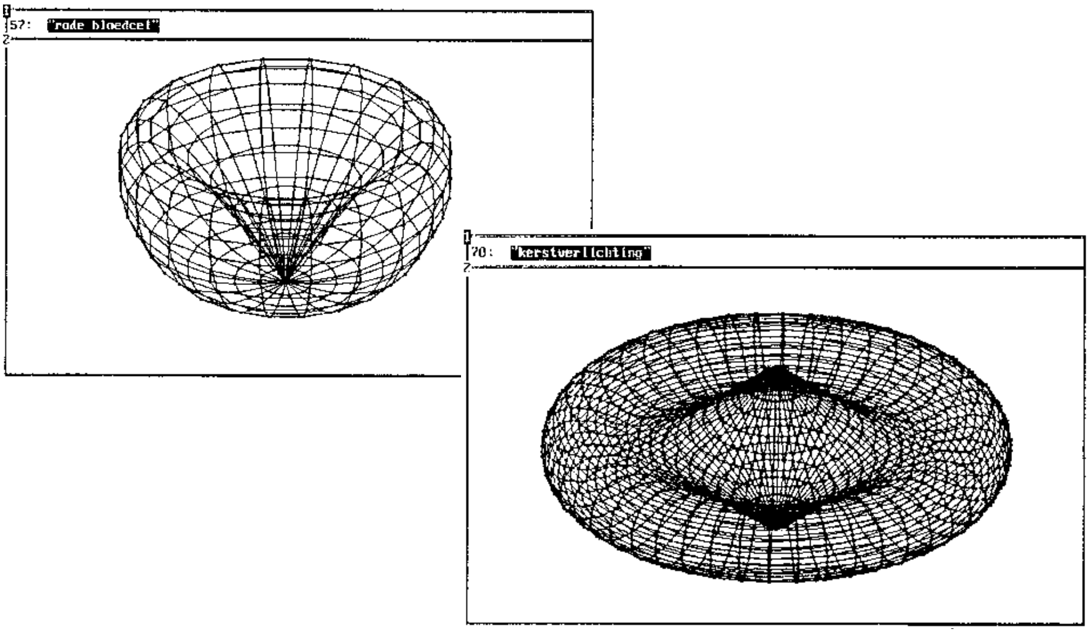
```
        CREATE(p, i, v) :=
          If i = n + 1
             v
#11:         If IN(v, Z(i, p)) = 1
                CREATE(p, i + 1, v)
                CREATE(p, i + 1, APPEND(v, [Z(i, p)]))

#12:    CYCLE(p) := CREATE(p, 1, [])

#13:    CYCLE(p)

#14:    [[1], [27, 14, 33, 17, 9, 5, 3, 2], [28, 40, 46, 49, 25, 13, 7, 4], [29, 15, 8,
          30, 41, 21, 11, 6], [31, 16, 34, 43, 22, 37, 19, 10], [32, 42, 47, 24, 38, 45,
          23, 12], [35, 18], [36, 44, 48, 50, 51, 26, 39, 20], [52]]
```

Two nice graphics from Belgium



G P Speck, the author of the contribution "Mueller´s Method" lives in Wanganui, New Zealand. The Wanganui River is the main river on the Northern Island of New Zealand. One of the famous places in the Wanganui region is the "Bridge to Nowhere". Regards to wonderful New Zealand

SOME IMPROVEMENTS ON THE RESOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS

José-Luis Llorens Fuster
Universidad Politécnica de Valencia
Departamento de Matemática Aplicada
46071-VALENCIA, SPAIN

First order equations

The general solution of the first order differential equation

$$p(x,y)\,\mathrm{d}x + q(x,y)\,\mathrm{d}y = 0 \qquad [a]$$

can be obtained using function **DSOLVE1_GEN(p,q,x,y,c)**, which is incorporated in the **ODE1.MTH file** (see [1], chap. 5, p.333; [2] chap. 9.6, p.252), always that this equation be:

- Separable
- Linear
- Homogeneous
- Exact
- Equation having an integrating factor which depends only on *x* or only on *y*.

When the differential equation is not one of the previous type, if we simplify the mentioned function we obtain "*inapplicable*". If we want to obtain the particular solution of this differential equation satisfying the initial condition (x_0,y_0), we can use the **DSOLVE1(p,q,x,y,x₀,y₀)** function.

Thus, the general solution of the equation (taking from [1], p. 353,ex. 3)

Ex. 1: $y'\sin^2 x = \cos^2 y$

can be obtained simplifying the expression DSOLVE1_GEN($\cos^2 y$, $-\sin^2 x$). The general solution will be obtained depending on the constant $c \in \mathbb{R}$.

In the ODE1.MTH file (which is now FirstOrderODEs.mth, Josef) there are some independent functions for these types of differential equations:

SEPARABLE_GEN(p, q, x, y, c) for the separable differential equation having the form $y' = p(x)\,q(y)$.

LINEAR1_GEN(p, q, x, y, c) for the linear DE having the form $y' + p(x)\,y = q(x)$.

HOMOGENEOUS_GEN(r, x, y, c) for the homogeneous,

EXACT_GEN(p, q, x, y, c) for the exact,

INTEGRATING_FACTOR_GEN(p, q, x, y, c) for these equations having an integrating factor which depends only on *x* or only on *y*. In the last three functions we suppose that the differential equation is written in the implicit form [a]. The **DSOLVE1_GEN** function *chains* the previous functions using the corresponding *tests* to identify each type of the equation. For example, if the differential equation satisfies:

$$\frac{\partial p}{\partial y} = \frac{\partial q}{\partial x}$$

then it is an exact one. Thus the definition of the function contains a sentence of the type IF(DIF(p,y) = DIF(q,x), EXACT_GEN(p,q,x,y,c),…). From this we can conclude that the efficiency of this function depends not only on the previous five definitions but also on the corresponding test.

On the other hand, it is possible to modify the way how to present the differential equation [a]. For example, we can write it as:

$$\frac{\mathrm{d}y}{\mathrm{d}x} = -\frac{p(x,y)}{q(x,y)}. \qquad \text{[b]}$$

So, in order to solve it we simplify the expression

DSOLVE1_GEN(-p(x,y)/q(x,y),–1)

and obviously we expect to obtain the same solution (or an equivalent one). However, the following illustration shows the behaviour of *DERIVE* (v. 3.01) with respect to the separable differential equation of example 1:

$$\text{DSOLVE1\_GEN}(\text{COS}(y)^2, -\text{SIN}(x)^2) = \text{"inapplicable"}$$

$$\text{DSOLVE1\_GEN}\left[\frac{\text{COS}(y)^2}{\text{SIN}(x)^2}, -1\right]$$

$$-\text{COT}(x) - \text{TAN}(y) = c$$

$$\text{SEPARABLE\_GEN}\left[\frac{1}{\text{SIN}(x)^2}, \text{COS}(y)^2\right]$$

$$\text{TAN}(y) = c - \text{COT}(x)$$

Obviously, the bug is not in the *resolution* of the equation (because, as you can see, the SEPARABLE_GEN function works correctly) but in the ***test***. But this is not an exceptional example: the following differential equations (taking from [1], p. 353, ex. 15, 18, 43, 45 and 46) are also of separable form:

Ex. 2: $x^3 y' = y^2 (x-4)$

Ex. 3: $x\,y'(2y-1) = y(1-x)$

Ex. 4: $\sec^2 x \tan y \,\mathrm{d}y + \sec^2 y \tan x \,\mathrm{d}x = 0$

Ex. 5: $x^2\, y\, y' = (1+x)\csc y$

Ex. 6: $\sin y \cos^2 y \,\mathrm{d}x + \cos^2 x \,\mathrm{d}y = 0$

$$\text{DSOLVE1\_GEN}(y^2\cdot(x-4),\ -x^3) = \text{"inapplicable"}$$

$$\text{DSOLVE1\_GEN}(y\cdot(1-x),\ -(2\cdot y-1)\cdot x) = \text{"inapplicable"}$$

$$\text{DSOLVE1\_GEN}(\text{SEC}(y)^2\cdot\text{TAN}(x),\ \text{SEC}(x)^2\cdot\text{TAN}(y)) = \text{"inapplicable"}$$

$$\text{DSOLVE1\_GEN}((1+x)\cdot\text{CSC}(y),\ x^2\cdot y) = \text{"inapplicable"}$$

$$\text{DSOLVE1\_GEN}(\text{SIN}(y)\cdot\text{COS}(y)^2,\ \text{COS}(x)^2) = \text{"inapplicable"}$$

Before following Llorens Fuster´s very valuable suggestions from 1995 let´s look at DERIVE´s solutions from today, Josef

The way how DERIVE 6 solves these differential equations:

#1: $$\text{DSOLVE1\_GEN}(y^2\cdot(x-4),\ -x^3) = \left[\frac{1}{y} = -\frac{c\cdot x^2 - x + 2}{x^2}\right]$$

#2: $$\text{DSOLVE1\_GEN}(y\cdot(1-x),\ -(2\cdot y-1)\cdot x) = (\text{LN}(y) - 2\cdot y = -\text{LN}(x) + x - c)$$

#3: $$\text{DSOLVE1\_GEN}(\text{SEC}(y)^2\cdot\text{TAN}(x),\ \text{SEC}(x)^2\cdot\text{TAN}(y)) = (\text{SIN}(y)^2 = 2\cdot c - \text{SIN}(x)^2)$$

#4: $$\text{DSOLVE1\_GEN}((1+x)\cdot\text{CSC}(y),\ x^2\cdot y) = \left(y\cdot\text{COS}(y) - \text{SIN}(y) = \text{LN}(x) - \frac{c\cdot x + 1}{x}\right)$$

#5: $$\text{DSOLVE1\_GEN}(\text{SIN}(y)\cdot\text{COS}(y)^2,\ \text{COS}(x)^2) = \left(-\text{LN}\left(\text{TAN}\left(\frac{y}{2}\right)\right) - \frac{1}{\text{COS}(y)} = \text{TAN}(x) - c\right)$$

The TI-92, Voyage 200 and TI-NSpireCAS are performing pretty the same:

$\text{deSolve}\left(y'\cdot(\sin(x))^2=(\cos(y))^2,x,y\right)$	$\tan(y)=\frac{-(\cos(x)-\mathbf{c1}\cdot\sin(x))}{\sin(x)}$
$\text{deSolve}\left(x^3\cdot y'=y^2\cdot(x-4),x,y\right)$	$y=\frac{-x^2}{\mathbf{c2}\cdot x^2-x+2}$
$\text{deSolve}\left(x\cdot y'\cdot(2\cdot y-1)=y\cdot(1-x),x,y\right)$	$2\cdot y-\ln(y)=\ln(x)-x+\mathbf{c3}$
$\text{deSolve}\left(y'=\frac{-(\sec(y))^2\cdot\tan(x)}{(\sec(x))^2\cdot\tan(y)},x,y\right)$	$\frac{-(\cos(y))^2}{2}=\frac{(\cos(x))^2}{2}+\mathbf{c4}$
$\text{deSolve}\left(x^2\cdot y\cdot y'=(1+x)\cdot\csc(y),x,y\right)$	$\sin(y)-y\cdot\cos(y)=\ln(x)-\frac{1}{x}+\mathbf{c5}$
$\text{deSolve}\left(y'\cdot(\cos(x))^2=-\sin(y)\cdot(\cos(y))^2,x,y\right)$	$\frac{\cos(y)\cdot\ln\left(\tan\left(\frac{y}{2}\right)\right)+1}{\cos(y)}=\frac{\mathbf{c6}\cdot\cos(x)-\sin(x)}{\cos(x)}$

We continue with Llorens Fuster from 1995:

The way to solve this problem is suggested from the first example. In all these cases, *DERIVE* seems to recognize more easily that the differential equation is separable if it is written in the explicit form [b] than in the implicit one [a]. So, we propose the following modification in the ODE1.MTH file:

a) We change the name of the functions DSOLVE1_GEN and DSOLVE1 respectively by **SOLVE1_GEN** and **SOLVE1**.

b) After that we add the following functions:

```
DSOLVE1(p, q, x, y, x0, y0, a_) := IF("inapplicable" = a_ := SOLVE1(p, q, x, y, x0, y0),
        SOLVE1(- p/q, -1, x, y, x0, y0), a_, a_)

DSOLVE1_GEN(p, q, x, y, c, a_) := IF("inapplicable" = a_ := SOLVE1_GEN(p, q, x, y, c),
        SOLVE1_GEN(- p/q, -1, x, y, c), a_, a_)
```

The meaning of these new functions is obvious: If the original function DSOLVE1 (now named SOLVE1) gives "*inapplicable*" then we try to apply it again writing the differential equation in the form [b]. Thus we solve all the examples from above:

$$\text{DSOLVE1\_GEN}(\cos(y)^2, -\sin(x)^2)$$
$$-\cot(x) - \tan(y) = c$$
$$\text{DSOLVE1\_GEN}(y^2\cdot(x-4), -x^3)$$
$$-\frac{1}{x} + \frac{2}{x^2} + \frac{1}{y} = c$$
$$\text{DSOLVE1\_GEN}(y\cdot(1-x), -(2\cdot y - 1)\cdot x)$$
$$\ln(x\cdot y) - x - 2\cdot y = c$$

$$\text{DSOLVE1\_GEN}(\sec(y)^2\cdot\tan(x), \sec(x)^2\cdot\tan(y))$$
$$-\frac{\sin(x)^2}{2} - \frac{\sin(y)^2}{2} = c$$
$$\text{DSOLVE1\_GEN}((1+x)\cdot\csc(y), x^2\cdot y)$$
$$-\ln(x) + y\cdot\cos(y) - \sin(y) + \frac{1}{x} = c$$
$$\text{DSOLVE1\_GEN}(\sin(y)\cdot\cos(y)^2, \cos(x)^2)$$
$$-\ln\left[\tan\left[\frac{y}{2}\right]\right] - \tan(x) - \frac{1}{\cos(y)} = c$$

We have suggested to Soft Warehouse to include "officially" this modification in the ODE1.MTH file, although some strange examples are still remaining:

Ex. 7: $(x\,y^2 - y^2 + x - 1)\,\mathrm{d}x + (x^2 y - 2x\,y + x^2 + 2y - 2x + 2)\,\mathrm{d}y = 0$

(taking from [1], p. 354, ex. 31) which admits the integrating factor $\mu(y) = e^{-2\arctan y}$. However, in this way the general solution is not obtained. The new definition of the DSOLVE1 function does not act because the given result is not "*inapplicable*". *DERIVE* identifies correctly the *type* of differential equation but is not able to obtain the solution because it cannot evaluate the integrals involved:

$$\mathrm{DSOLVE1\_GEN}(x\cdot y^2 - y^2 + x - 1,\ x^2\cdot y - 2\cdot x\cdot y + x^2 + 2\cdot y - 2\cdot x + 2)$$

$$e^{2\cdot\mathrm{ATAN}(y)}\cdot\left[\frac{x^2\cdot(y^2+1)}{2} - x\cdot(y^2+1)\right] + 2\cdot\int y\cdot e^{2\cdot\mathrm{ATAN}(y)}\,dy + 2\cdot\int e^{2\cdot\mathrm{ATAN}\ldots}$$

$$\mathrm{DSOLVE1\_GEN}\left[-\frac{x\cdot y^2 - y^2 + x - 1}{x^2\cdot y - 2\cdot x\cdot y + x^2 + 2\cdot y - 2\cdot x + 2},\ -1\right]$$

$$-\mathrm{ATAN}(y) - \frac{\mathrm{LN}((x^2 - 2\cdot x + 2)\cdot(y^2+1))}{2} = c$$

As it is shown in the third expression in the previous illustration, it is solved if we write the equation in form [b]: This is due to the fact that now the integrating factor is $\mu = \frac{y^2+1}{y+1}$, which is leading to an exact differential without appearing the mentioned integrals. Obviously this example is very rare. To solve this problem we need to modify more deeply the ODE1.MTH file.

As you can see even *DERIVE* 6 is not perfect in solving ODEs. It needs the same rewriting as in 1995.

#1: $\mathrm{DSOLVE1\_GEN}(x\cdot y^2 - y^2 + x - 1,\ x^2\cdot y - 2\cdot x\cdot y + x^2 + 2\cdot y - 2\cdot x + 2)$

#2: $e^{2\cdot\mathrm{ATAN}(y)}\cdot\left(\frac{x^2\cdot(y^2+1)}{2} - x\cdot(y^2+1)\right) + 2\cdot\mathrm{SUBST}\left(\int\frac{e^{2\cdot y}}{\mathrm{COS}(y)^2}\,dy,\ y,\ \mathrm{ATAN}(y)\right) + 2\cdot\mathrm{SUBST}\left(\int\frac{e^{2\cdot y}\cdot\mathrm{SIN}(y)}{\mathrm{COS}(y)^3}\,dy,\ y,\ \mathrm{ATAN}(y)\right) = c$

#3: $\mathrm{DSOLVE1\_GEN}\left(-\frac{x\cdot y^2 - y^2 + x - 1}{x^2\cdot y - 2\cdot x\cdot y + x^2 + 2\cdot y - 2\cdot x + 2},\ -1\right)$

#4: $\mathrm{ATAN}(y) + \frac{\mathrm{LN}((x^2 - 2\cdot x + 2)\cdot(y^2+1))}{2} = -c$

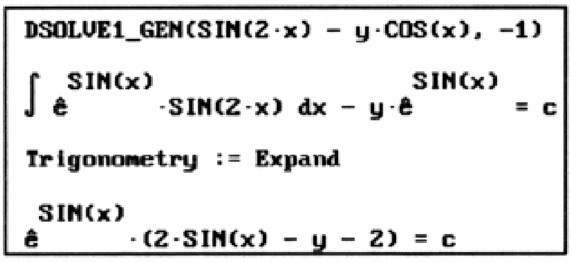
It is interesting that the TI-family shows no problem to solve this "rare" kind of differential equation without hesitating.

```
F1 F2 Algebra F3 Calc F4 Other F5 PrgmIO F6 Clean Up
■ deSolve(y' = -(x·y^2 - y^2 + x - 1)/(x^2·y - 2·x·y + x^2 + 2·y - 2·x ▸
      ln(y^2 + 1)/2 + tan⁻¹(y) = @6 - ln(x^2 - 2·x + 2)/2
■ deSolve(y'·(x^2·y - 2·x·y + x^2 + 2·y - 2·x ▸
      ln(y^2 + 1)/2 + tan⁻¹(y) = @8 - ln(x^2 - 2·x + 2)/2
...-2x+2)=-(x*y^2-y^2+x-1),x,y)
MAIN          RAD AUTO          FUNC 2/30
```

Finally we show an example that **is neither** a limitation of this function **nor** in general of the program. The linear differential equation

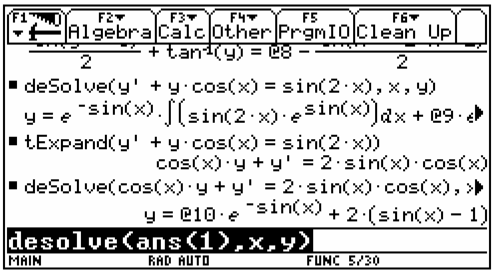
Ex. 8: $y' + y\cos x = \sin 2x$

([1], p. 366, ex. 56) is not correctly solved if we don´t select **Manage-Trigonometry-Expand:**



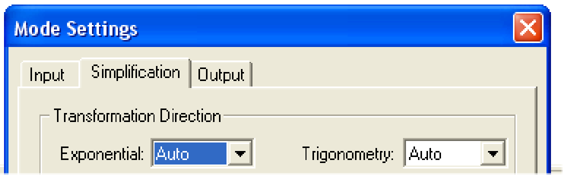
This is no problem for *DERIVE 6*.

But it is a problem for the Voyage 200 – and TI-NspireCAS as well – which can be re-solved expanding the trig expression before integrating the differential equation.



#5: DSOLVE1_GEN(SIN(2·x) − y·COS(x), −1)

#6: $e^{\mathrm{SIN}(x)}\cdot(2\cdot\mathrm{SIN}(x) - y - 2) = c$



This is the recent form of DSOLVE1_GEN how it is implemented in *DERIVE* 6. José-Luis suggestion for improving the version from *DERIVE* 3 was obviously accepted by Soft Ware-house and extended to include other cases, too.

```
DSOLVE1_GEN(p, q, x, y, c, a_) :=
  If "inapplicable" = a_ := INTEGRATING_FACTOR_GEN(p, q, x, y, c)
     If "inapplicable" = a_ := HOMOGENEOUS_GEN(- p/q, x, y, c)
        If "inapplicable" = a_ := SEP_GEN(p, q, x, y, c)
           GEN_HOM_GEN(- p/q, x, y, c)
           a_
           a_
        a_
        a_
     a_
     a_
```

Second order equations

When a second order differential equation $y'' = u(x, y, y')$ is **incomplete**, i.e., when some of its terms in *x*, *y*, or *y'* do not appear, sometimes it is possible to solve it directly or reduce it to two first order equations. So:

a) *y* and *x* do not appear: the equations are of the form $y'' = f(y')$. The parametric equations of the general solution as a function of the parameter *v* and the constants *c* and *k*, are:

$$x = \int \frac{\mathrm{d}v}{f(v)} + c, \quad y = \int \frac{v\,\mathrm{d}v}{f(v)} + k.$$

b) *y'* and *x* do not appear: the equations are of the form $y'' = f(y)$. The general solution can be written in one of the following expressions as a function of the constants *c* and *k*:

$$x = \int \frac{\mathrm{d}y}{\sqrt{2\int f(y)\,\mathrm{d}y + k}} + c, \qquad y = \int \sqrt{2\int f(y)\,\mathrm{d}y + k}\,\mathrm{d}x + c.$$

c) *y* does not appear: using the change $v = y'$ we transform it to a first order differential equation (in *v'*, *v*, *x*). The general solution of this equation is, obviously, another first order equation.

d) *x* does not appear: the changes $y' = v,\ y'' = v' \cdot v,$ transform it to a first order differential equation with the variables *v* (in place of *y*) and *y* (in place of *x*). The general solution of this equation is, again, another first order equation.

It is well known, that in the ODE2.MTH file are only two functions to solve the differential equations presented in a) and d). In particular:

AUTONOMOUS(r,v). We apply it to solve equation d), written in the form $y'' = r(y, v)$ with $y' = v$. *DERIVE* simplifies it solving for *v'* after applying the previous change.

AUTONOMOUS_CONSERVATIVE(r,x,y,x$_0$,y$_0$,v$_0$). We apply it to solve equation d), written in the form $y'' = r(y),$ where the initial conditions $y(x_0) = y_0,\ y'(x_0) = v_0$ are supposed. So, it cannot be used to obtain the general solution of the equation.

Our objective is to define auxiliary functions to obtain the general solution of the differential equations of kinds a), b), c) and d).

The following shows what José-Luis proposed in 1995. Then we will have a look how the utility file **SecondOrderODES.mth** is supporting solving second order DEs in our times.

➲ For case a), i.e. $y'' = u(y') = u(v)$, it is very easy to define the function:

```
FALTAYX(u,v,x,y,c,k):=[x=INT(1/u,v)+c,y=INT(v/u,v)+k]
```

➲ Similarly, for the incomplete equations of kind b), i.e. $y'' = u(y)$:

```
FALTAXV(u,v):=["Selec.:",x=INT((2∫(u,y)+k)^(-1/2),y)+c,
                   y=INT(2∫(u,y)+k^(1/2),x)+c]
```

In this way we obtain two equivalent expressions of the general solution. It is possible that one of the two solutions leads to an unsimplificable integral, depending on the form of the equation; for these reasons we present the two expressions.

➲ Finally, for the equations of type c) and d), i.e. $y'' = u(x, y)$ or $y'' = u(y, v)$ respectively, we have to define in the first step two auxiliary functions with the following purpose:

- to solve the first order differential equation obtained after *y*' = *v* as we stated previously,
- to solve for *v* in the general solution of this differential equation.

```
AU1(u,v,x,y,k):=SOLUTIONS(DSOLVE1_GEN(u,-1,x,v,k),v)

AU2(u,v,x,y,k):=SOLUTIONS(DSOLVE1_GEN(u/v,-1,y,v,k),v)
```

(I had to adapt the version from DERIVE 3 to DERIVE 6 because of another form of output for the SOLVE command.)

Now we define the corresponding main functions in which we apply the DSOLVE1_GEN function in the obtained solution with each auxiliary function. As it is possible to obtain two or more different expressions when we solve for *v*, we use the VECTOR command:

```
FALTAY(u,v,x,y,k,c):=
VECTOR(DSOLVE1_GEN((AU1(u,v,x,y,k))↓i,-1,x,y,c),i,
DIM(AU1(u, v, x, y, k)))

FALTAX(u,v,x,y,k,c):=
VECTOR(DSOLVE1_GEN((AU2(u,v,x,y,k))↓i,-1,x,y,c),i,
DIM(AU2(u, v, x, y, k)))
```

The efficiency of these functions depends on the following three factors:

- The fact that the first order differential equation obtained after doing the substitution of the variables belongs to these kind of equations which are recognized by DSOLVE1_GEN;

- The possibility of solving for *v* in the solution of this equation;

- The fact that the new differential equation obtained in this process is again of a type of equation which is recognized by DSOLVE1_GEN.

In the following functions we study the way how *DERIVE* **identifies** the kind of incomplete equation and, as a consequence, applies the right function to solve it.

```
TESTX(u,v):=IF(∂(u,x)=0,FALTAX(u,v),"inapplicable",
            "inapplicable")

TESTY(u,v):=IF(∂(u,y)=0,FALTAY(u,v),TESTX(u,v),TESTX(u,v))

TESTV(u,v):=IF(∂(u,v)=0 AND ∂(u,x)=0,FALTAXV(u,v),
             TESTY(u,v),TESTY(u, v))

ODE2I(u,v):=IF(∂(u,x)=0 AND ∂(u,y)=0,FALTAYX(u),
            TESTV(u,v),TESTV(u,v))

DSOLVE2I(u,v,w):=ODE2I((SOLUTIONS(u,w))↓1,v)
```

We apply the **ODE2I** function for the second order equation $y'' = u(x, y, v)$. The result is "inapplicable" if the equation does not belong to any of the types a) , b) c) or d). In other case we obtain the solution with the restrictions noted previously for the kinds c) and d).

We apply the **DSOLVE2I** function for the second order equation $u(w, v, x, y) = 0$, where $w = y''$. Thus, is the *main* function, where it is **not** necessary solving $w = y''$ before. Given an equation, it is sufficient to substitute $w = y'', v = y'$, as one can see in the following examples (taking from [1], p. 357):

Ex. 9 (type c): $y'' - \dfrac{y'}{x} = x \sin x$

$$\text{DSOLVE2I}\left(w - \frac{1}{x}\cdot v = x\cdot \text{SIN}(x)\right) = \left[\text{COS}(x) + x\cdot\text{SIN}(x) + \frac{k\cdot x^2}{2} + y = -c\right]$$

Ex. 10 (type c): $y' = y''x + y''^2$

$$\text{DSOLVE2I}(v = w\cdot x + w^2)$$

$$\left[y = \int \text{IF}\left(x > 0,\ \text{IF}\left((x\cdot e^k < -1 \wedge x < 0) \vee (x\cdot e^k < 1 \wedge x > 0),\ \frac{e^{-2\cdot k}}{4} - \frac{e^{-k}\cdot|x|}{2}\right)\right) dx + c\right]$$

$$\int\left(\frac{e^{-2\cdot k}}{4} - \frac{e^{-k}\cdot x}{2}\right) dx + c = -\frac{x^2\cdot e^{-k}}{4} + \frac{x\cdot e^{-2\cdot k}}{4} + c$$

Jose-Luis Fuster presents another result. So I´d like to double check the solution. I define a function y_(x) and substitute in example 10.

This solution seems to be correct.

$$y\_(x) := \frac{x\cdot e^{-2\cdot k}}{4} - \frac{x^2\cdot e^{-k}}{4} + c$$

$$y\_'(x) - (y\_''(x)\cdot x + y\_''(x)^2) = 0$$

Ex. 11 (type c): $y'\,y'' = (1+y'^2)^{1/2}$

$$\text{DSOLVE2I}(v\cdot w = (1+v^2)^{1/2})$$

$$\left[x = c + \sqrt{v^2+1},\ y = -\frac{\text{LN}(\sqrt{v^2+1}+v)}{2} + k + \frac{v\cdot\sqrt{v^2+1}}{2}\right]$$

Ex. 12 (type a): $y'' - y'^2 = 1$

$$\text{DSOLVE2I}(w - v^2 = 1) = \left[x = \text{ATAN}(v) + c,\ y = \frac{\text{LN}(v^2+1)}{2} + k\right]$$

Ex. 13 (type c): $(1-x^2)y'' - x\,y' - 2 = 0$

$$\text{DSOLVE2I}((1-x^2)\cdot w - x\cdot v - 2 = 0)$$

$$\left[\text{LN}(\sqrt{x^2-1}+x)^2 + k\cdot\text{LN}(\sqrt{x^2-1}+x) + y = -c\right]$$

Ex. 14 (type a): $y'' + \sqrt{1-y'^2} = 0$

$$\text{DSOLVE2I}(w + \sqrt{1-v^2}) = \left[x = c - \text{ASIN}(v),\ y = \sqrt{1-v^2} + k\right]$$

Ex. 15 (type d): $y\,y'' = y'^3$

$$\text{DSOLVE2I}(y\cdot w = v^3) = [y\cdot\text{LN}(y) + x - y\cdot(k+1) = -c]$$

Ex. 16 (type d): $y'^2 = y\,y''$

$$\text{DSOLVE2I}(v^2 = y\cdot w) = \left[x\cdot e^{-k} - \text{LN}(y) = c\right]$$

Ex. 17 (type d): $y'' - y'^3 + y\,y'^3 = 0$

$$\text{DSOLVE2I}(w - v^3 + y\cdot v^3) = \left[2\cdot x - \frac{y^3}{3} + y^2 - 2\cdot k\cdot y = c\right]$$

Ex. 18 (type d): $y'' = y'^3 \ln y$

$$\text{DSOLVE2I}(w = v^3\cdot\text{LN}(y)) = \left[\frac{y^2\cdot\text{LN}(y)}{2} + x - \frac{3\cdot y^2}{4} - k\cdot y = -c\right]$$

Ex. 19 (type a): $2y'y'' = 1 + y'^2$

$$\text{DSOLVE2I}(2\cdot v\cdot w = 1 + v^2) = \left[x = \text{LN}(v^2 + 1) + c,\ y = -\ 2\cdot\text{ATAN}(v) + k + 2\cdot v\right]$$

Ex. 20 (type b): $y'' = \dfrac{1}{\sqrt{y}}$

$$\text{DSOLVE2I}\left(w = \frac{1}{\sqrt{y}}\right) = \left[\text{Selec.:},\ x = \frac{(2\cdot\sqrt{y} - k)\cdot\sqrt{(4\cdot\sqrt{y} + k)}}{6} + c,\ y = x\cdot(4\cdot\sqrt{y} + \sqrt{k}) + c\right]$$

Ex. 21 (type b): $y'' = 9y$

$$\left[\text{Selec.:},\ x = \frac{\text{LN}(\sqrt{(9\cdot y^2 + k)} + 3\cdot y)}{3} + c,\ y = x\cdot(9\cdot y^2 + \sqrt{k}) + c\right]$$

$$\text{DSOLVE2}(0, -9, 0)$$

$$c1\cdot e^{3\cdot x} + c2\cdot e^{-\ 3\cdot x}$$

As the illustration shows, the differential equation in the next example can also be solved taking into account that it is a linear equation. The next example corresponds to a *not-incomplete linear* equation (although *y'* does not appear):

Ex. 22: $y'' - y = \cos 2x - 2\sin 2x$

$$\text{DSOLVE2I}(w - y = \text{COS}(2\cdot x) - 2\cdot\text{SIN}(2\cdot x)) = \text{inapplicable}$$

The two final examples show some limitations. In the first, the expression of the obtained solution seems not to be very satisfying. In the second (taking from [2], p. 261) we do not obtain a solution. In both cases the problem is caused by the resolution of the first order equations, i.e. from DSOLVE1_GEN.

Ex. 23 (type d): $1 + y'^2 = y\,y''$

$$\text{DSOLVE2I}(1 + v^2 = y\cdot w) = \left[\left[\int \frac{1}{\text{IF}(y > 0,\ e^{-k}\cdot\sqrt{(y^2 - e^{2\cdot k})})}\,dy - x = -c,\ \int \frac{1}{\text{IF}(y > 0,\ -\ e^{-k}\cdot\sqrt{(y^2 - e^{2\cdot k})})}\,dy - x = -c\right]\right]$$

Ex. 24 (type d): $y\,y'' + 1 + y'^2 = 0$

$$\text{DSOLVE2I}(y\cdot w + v^2 + 1) = \left[\int \frac{1}{\text{IF}\left(y>0,\ \frac{e^{-k}\cdot\sqrt{(1-y^2\cdot e^{2\cdot k})}}{|y|}\right)}\,dy - x = -c,\ \int \frac{1}{\text{IF}\left(y>0,\ -\frac{e^{-k}\cdot\sqrt{(1-y^2\cdot e^{2\cdot k})}}{|y|}\right)}\,dy - x = -c\right]$$

References

[1] Llorens Fuster, J.L.: "*Aplicaciones de DERIVE: Análisis Matemático-I (Cálculo)*" Servicio de Public. de la Universidad Politécnica de Valencia, 1993

[2] Soft Warehouse "*DERIVE User Manual, Version 3*", Honolulu, 1994.

Some comments for the revised version

I´d like to show how DERIVE 6 and the TIs are treating the ODEs of 2nd order. First of all I´ll solve one or the other of the presented differential equations in the traditional way. This might support the students´ understanding of the implemented CAS procedures.

Let´s start with **Ex. 9 (type c)**:

(1) $$y'' - \frac{1}{x}y' = x\sin x$$

We substitute *y*' = *v* and *y*'' = *v*' in order to reduce the given DE to a linear DE:

(2) $$v' - \frac{1}{x}v = x\ \sin x.$$

The standard technique for solving linear DEs of the form $y' + p(x)y = q(x)$ is to find an integrating factor $\rho(x)$ and then multiply both sides of the equation by this factor:

$$\rho(x) = e^{\int -\frac{1}{x}dx} = e^{-\ln x} = \frac{1}{x}.$$

$$\frac{1}{x}\cdot v' - \frac{1}{x^2}\cdot v = \sin x$$

(3) $$\left(\frac{1}{x}\cdot v\right)' = \sin x \quad |\ \text{Integrate wrt } x$$

$$\frac{1}{x}\cdot v = -\cos x + k$$

$$v = -x\cos x + k\,x \ \rightarrow\ y' = -x\cos x + k\,x \rightarrow y = -x\sin x - \cos x + \frac{k\,x^2}{2} + c$$

The last step is easy work: We can integrate directly applying integration by parts.

I proceed with **Ex. 12 (type a)**:

(1) $$y'' = y'^2 + 1$$

I found in my (old) textbooks that substituting *v* for *y'* might be successful. So let me try:

(2) $$y' = \frac{dy}{dx} = v \to dx = \frac{dy}{v}$$
$$\text{then } y'' = \frac{dv}{dx} = \frac{v \cdot dv}{dy}$$

I perform the substitution in (1) and integrate the 1st order DE by separation of variables:

(3) $$\frac{v\,dv}{dy} = v^2 + 1 \ \to \ dy = \frac{v\,dv}{v^2+1} \ | \text{ Integrate}$$
$$y = \frac{\ln(v^2+1)}{2} + k$$

(4) $$\frac{dv}{dx} = 1 + v^2 \to dx = \frac{dv}{1+v^2} \ | \text{ Integrate}$$
$$x = \tan^{-1} v + c$$

This is the parameter representation which is given as result of example 9. Finally we can try to have an explicit form of the solution:

From (4): $v = \tan(x - c)$ and substituting in (3):

$$y = \frac{\ln(\tan^2(x-c)+1)}{2} + k = \frac{1}{2}\ln\left(\frac{1}{\cos^2(x-c)}\right) + k = -\ln\left|\cos(x-c)\right| + k \ .$$

But I can also try solving the DE directly by integrating twice:

$$\frac{dv}{dx} = 1 + v^2 \to dx = \frac{dv}{1+v^2} \ | \text{ Integrate}$$
$$x + c = \tan^{-1}(v) \to v = y' = \tan(x+c)$$
$$\frac{dy}{dx} = \tan(x+c) \ | \text{ Integrate}$$
$$y = -\ln\left|\cos(x+c)\right| + k$$

Let´s have a **type b)** example: **Ex. 21**: (I don´t recognize the DE as a differential equation with constant coefficients and apply again an appropriate substitution for reducing the order of the equation.

$$v = y' \text{ and } y'' = \frac{dv}{dx} = \frac{v \cdot dv}{dy} \ \to \ 9y = \frac{v \cdot dv}{dy}$$
$$9y\,dy = v\,dv \ | \text{ Integrate}$$
$$\frac{9y^2}{2} + k = \frac{v^2}{2}$$
$$9y^2 + k_1 = v^2 \ \to \ v = \sqrt{9y^2 + k_2}$$

$$\frac{dy}{dx} = \sqrt{9y^2 + k_2} \ \to \ \frac{dy}{\sqrt{9y^2 + k_2}} = dx \ | \text{ Integrate}$$
$$x = \frac{\ln(\sqrt{9y^2 + k_2} + 3y)}{3} + c$$

I am trying to bring the solution in its explicit form.

$$\sqrt{9y^2+k_2}+3y=e^{3x-3c}$$
$$9y^2+k_2=(e^{3x-3c}-3y)^2=e^{6x-6c}-6y\cdot e^{3x-3c}+9y^2$$
$$6y\cdot e^{3x-3c}=e^{6x-6c}-k_2$$
$$y=\frac{e^{3x-3c}}{6}-\frac{k_2\cdot e^{-3x+3c}}{6}=c_1e^{3x}+c_2e^{-3x}.$$

As you can see now we obtained finally the solution in the form how it is given by solving the DE as a homogeneous DE with constant coefficients (using the characteristic equation).

Of course, one can do all the symbolic manipulation supported by CAS, too. But sometimes it´s nice to do it manually, isn´t it?

We miss an example of **type d)**, so let me choose **Ex. 16**:

$$y'^2=y\cdot y''$$
$$v=y' \text{ and } y''=\frac{\mathrm{d}v}{\mathrm{d}x}=\frac{v\cdot\mathrm{d}v}{\mathrm{d}y}\rightarrow v^2=y\cdot\frac{v\cdot\mathrm{d}v}{\mathrm{d}y} \mid \text{Separation of variables}$$
$$\frac{\mathrm{d}y}{y}=\frac{\mathrm{d}v}{v} \mid \text{Integrate}$$
$$\ln y=\ln v+\ln k\rightarrow y=k\cdot v=k\cdot\frac{\mathrm{d}y}{\mathrm{d}x} \mid \text{Separate variables again}$$
$$\frac{\mathrm{d}x}{k}=\frac{\mathrm{d}y}{y} \mid \text{Integrate}$$
$$\frac{x}{k}+\ln c=\ln y$$
$$y=c\cdot e^{\frac{x}{k}}=c\cdot e^{k_1\cdot x}$$

I would leave it to the students to compare this solution with the solution given on page 19 as the result of Llorens Fuster´s function. You can also double check the solution by substituting into the given DE.

I must say that I enjoyed comparing the traditional way resolving the equation(s) and their outcomes with the CAS results.

How does DERIVE 6 perform?

```
DSOLVE2(p,q,r,x,c1,c2) simplifies to an explicit gen-
eral solution of the linear second order ordinary dif-
ferential equation

               y" + p(x)·y' + q(x)·y = r(x)

in terms of arbitrary constants c1 and c2.  Note that
the last two arguments can be omitted if they are vari-
ables and you are satisfied with the names c1 and c2.

If no method applies or the equation cannot be con-
verted to an equivalent one having a p and a q that are
independent of x, DSOLVE2 returns the word "inapplica-
ble".(Online Help)
```

Example 9:

#1: $\text{DSOLVE2}\left(-\frac{1}{x}, 0, x\cdot\text{SIN}(x)\right) = \text{inapplicable}$

#2: $\text{DSOLVE1\_GEN}\left(-\frac{1}{x}\cdot v - x\cdot\text{SIN}(x), 1, x, v\right) = \left(\text{COS}(x) + \frac{v}{x} = c\right)$

#3: $\text{SOLVE}\left(\text{COS}(x) + \frac{v}{x} = c, v\right) = (v = c\cdot x - x\cdot\text{COS}(x))$

#4: $\text{DSOLVE1\_GEN}(c\cdot x - x\cdot\text{COS}(x), -1, x, y, k) = \left(\text{COS}(x) + x\cdot\text{SIN}(x) - \frac{c\cdot x^2}{2} + y = -k\right)$

As you can see, DSOLVE2 does not apply, one has to perform the reduction of the order.

```
AUTONOMOUS(r, v) simplifies an expression for dv/dy,
given an autonomous equation y" = r(y, v) with v repre-
senting y'.
```

Example 12:

#5: $\text{AUTONOMOUS}(v^2 + 1) = \frac{v^2 + 1}{v}$

#6: $\text{DSOLVE1\_GEN}\left(\frac{v}{v^2 + 1}, -1, v, y\right) = \left(\frac{\text{LN}(v^2 + 1)}{2} - y = c\right)$

#7: $\text{SOLVE}\left(\frac{\text{LN}(v^2 + 1)}{2} - y = c, v\right) = (v = -\sqrt{(e^{2\cdot y + 2\cdot c} - 1)} \vee v = \sqrt{(e^{2\cdot y + 2\cdot c} - 1)})$

#8: $\text{DSOLVE1\_GEN}(\sqrt{(e^{2\cdot y + 2\cdot c} - 1)}, -1, x, y, k) = (\text{ATAN}(\sqrt{(e^{2\cdot y + 2\cdot c} - 1)}) - x = -k)$

#9: $\text{DSOLVE1\_GEN}(-\sqrt{(e^{2\cdot y + 2\cdot c} - 1)}, -1, x, y, k) = (\text{ATAN}(\sqrt{(e^{2\cdot y + 2\cdot c} - 1)}) + x = -k)$

#10: $\text{SOLVE}(\text{ATAN}(\sqrt{(e^{2\cdot y + 2\cdot c} - 1)}) + x = -k, y)$

accepting some restrictions for the domain:

#11: $y = -\frac{\text{LN}(\text{COS}(x + k)^2)}{2} - c$

Example 21 is given in Fuster´s paper.

Example 16 is the last in my row:

$$\#13:\quad \text{AUTONOMOUS}\left(\frac{v^2}{y}\right) = \frac{v}{y}$$

$$\#14:\quad \text{DSOLVE1\_GEN}\left(\frac{y}{v}, -1, v, y\right) = (\text{LN}(y) - \text{LN}(v) = -c)$$

$$\#15:\quad \text{SOLUTIONS}(\text{LN}(y) - \text{LN}(v) = -c, v) = \left[y\cdot e^{c}\right]$$

$$\#16:\quad \text{DSOLVE1\_GEN}(y\cdot e^{c}, -1, x, y, k) = (x\cdot e^{c} - \text{LN}(y) = k)$$

$$\#17:\quad \text{SOLVE}(x\cdot e^{c} - \text{LN}(y) = k, y) = \left(y = e^{x\cdot e^{c} - k}\right)$$

In my opinion José-Luis' tools are a great support even in times of DERIVE 6. Referring to the original DERIVE tools requires a lot of more knowledge (which is not so bad) and of more manipulating.

I wanted to compare DERIVE with the TIs and tried to solve Examples 9 through 24 using TI-NspireCAS (and the TI-92+ and Voyage 200):

© Example 9

$\text{deSolve}\left(y''-\frac{1}{x}\cdot y'=x\cdot\sin(x),x,y\right)$ $\quad y=-\cos(x)-x\cdot\sin(x)+\frac{c1\cdot x^2}{2}+c2$

© Example 10

$\text{deSolve}\left(y'=y''\cdot x+(y'')^2,x,y\right)$

$y=\left\{\frac{-e^{-2\cdot c5}\cdot x\cdot\left(e^{c5}\cdot|x|-1\right)}{4}, e^{-c5}\cdot\left(\text{sign}(x)-e^{c5}\cdot x\right)\le 0 \text{ and } x\ge 0\right. +c6 \text{ or } y=\left\{\frac{-e^{-2\cdot c7}\cdot x\cdot\left(e^{c7}\cdot|x|-1\right)}{4}, e^{-c7}\cdot(\text{sig}\right.$ ▸

© Example 11

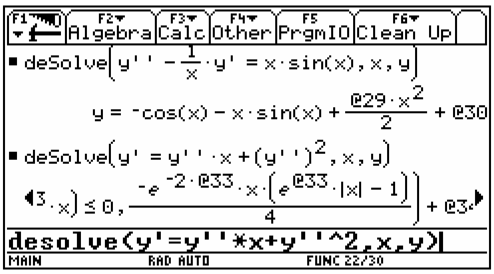
$\text{deSolve}\left(y'\cdot y''=\sqrt{1+(y')^2},x,y\right)$

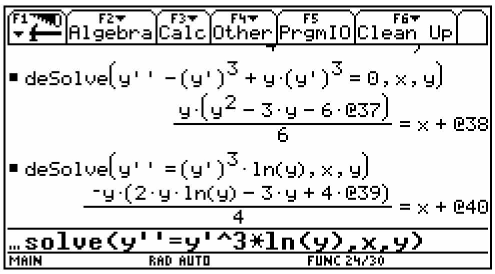
$y=\left\{\frac{\ln\left(\sqrt{x^2+2\cdot c9\cdot x+c9^2-1}+x+c9\right)-(x+c9)\cdot\sqrt{x^2+2\cdot c9\cdot x+c9^2-1}}{2}, x^2+2\cdot c9\cdot x+c9^2\ge 1 \text{ and } x+c9\ge 0\right. +c10$ ▸

© Example 12

$\text{deSolve}\left(y''-(y')^2=1,x,y\right)$ $\quad y=\left\{-\ln\left(\cos(x+c11)\right), -c11-\frac{\pi}{2}\le x\le\frac{\pi}{2}-c11\right. +c12$

As there is the same CAS implemented it is no surprise that NSpire and the handhelds are behaving pretty the same (Examples 9, 10, 17 and 18):





This is the rest of the list done with TI-NspireCAS:

© Example 13

deSolve$((1-x^2)\cdot y''-x\cdot y'-2=0,x,y)$ $\quad y=-(\ln(\sqrt{x^2-1}+x))^2+c13\cdot\ln(\sqrt{x^2-1}+x)+c14$

© Example 14

deSolve$(y''+\sqrt{1-(y')^2}=0,x,y)$ $\quad y=\{\cos(x-c15), c15-\frac{\pi}{2}\le x\le c15+\frac{\pi}{2}+c16$

© Example 15

deSolve$(y\cdot y''=(y')^3,x,y)$ $\quad -y\cdot(\ln(y)+c17-1)=x+c18$

© Example 16

deSolve$((y')^2=y\cdot y'',x,y)$ $\quad \ln(|y|)=c19\cdot x+c20$

© Example 17

deSolve$(y''-(y')^3+y\cdot(y')^3=0,x,y)$ $\quad \frac{y\cdot(y^2-3\cdot y-6\cdot c21)}{6}=x+c22$

© Example 18

deSolve$(y''=(y')^3\cdot\ln(y),x,y)$ $\quad \frac{-y\cdot(2\cdot y\cdot\ln(y)-3\cdot y+4\cdot c23)}{4}=x+c24$

© Example 19

deSolve$(2\cdot y'\cdot y''=1+(y')^2,x,y)$

$y=\{2\cdot(\tan^{-1}(\sqrt{c25\cdot e^x-1})-\sqrt{c25\cdot e^x-1}), c25\cdot e^x\ge 1+c26$ or $y=\{2\cdot\sqrt{c25\cdot e^x-1}-2\cdot\tan^{-1}(\sqrt{c25\cdot e^x-1}), c25\cdot e^x$ ▸

© Example 20

deSolve$(y''=\frac{1}{\sqrt{y}},x,y)$ $\quad \frac{(2\cdot\sqrt{y}-c28)\cdot\sqrt{4\cdot\sqrt{y}+c28}}{6}=x+c27$

© Example 21

deSolve$(y''=9\cdot y,x,y)$ $\quad y=c30\cdot e^{3\cdot x}+c29\cdot e^{-3\cdot x}$

© Example 22

deSolve$(y''-y=\cos(2\cdot x)-2\cdot\sin(2\cdot x),x,y)$ $\quad y=\frac{-\cos(2\cdot x)}{5}+\frac{2\cdot\sin(2\cdot x)}{5}+c31\cdot e^{-x}+c32\cdot e^x$

© Example 23

deSolve$(1+(y')^2=y\cdot y'',x,y)$

$\int\left(\frac{1}{\{-\sqrt{c33\cdot y^2-1}, c33\cdot y^2\ge 1}\right)dy=x+c34$ or $\int\left(\frac{1}{\{\sqrt{c33\cdot y^2-1}, c33\cdot y^2\ge 1}\right)dy=x+c34$

© I change the settings from Real or Complex Format > Real to > Rectangular:

deSolve$(1+(y')^2=y\cdot y'',x,y)$ $\quad \frac{-\ln(\sqrt{c39\cdot y^2-1}+\sqrt{c39}\cdot y)}{\sqrt{c39}}=x+c40$ or $\frac{\ln(\sqrt{c39\cdot y^2-1}+\sqrt{c39}\cdot y)}{\sqrt{c39}}=x+c40$

© Example 24

deSolve$(y\cdot y''+1+(y')^2=0,x,y)$ $\quad \int\left(\frac{-|y|\cdot i}{\sqrt{y^2-c41}}\right)dy=x+c42$ or $\int\left(\frac{|y|\cdot i}{\sqrt{y^2-c41}}\right)dy=x+c42$

You can see that the TIs are well prepared to solve 2nd order ODEs of this kind which are appearing in Llorens Fuster´s selection.

I felt challenged to check my freshly acquired (and remembered) knowledge in solving 2nd order ODEs applying my "skills" on examples 23 and 24.

Example 23:

$$1+y'^2 = y\cdot y''$$

$$v = y' \text{ and } y'' = \frac{dv}{dx} = \frac{v\cdot dv}{dy} \rightarrow 1+v^2 = \frac{y\cdot v\cdot dv}{dy}$$

$$\frac{dy}{y} = \frac{v\cdot dv}{1+v^2} \mid \text{Integrate}$$

$$\ln y = \frac{1}{2}\ln(1+v^2) + \frac{1}{2}\ln c \rightarrow y = \sqrt{c\cdot(1+v^2)}$$

$$y^2 = c + c\cdot v^2 \mid \text{Solve for } v = y'$$

$$v = \frac{dy}{dx} = \pm\sqrt{\frac{y^2-c}{c}} \mid \text{I take the positive root and separate the variables}$$

$$\frac{dy\cdot\sqrt{c}}{\sqrt{y^2-c}} = dx \mid \text{Integrate again}$$

$$\sqrt{c}\cdot\ln(\sqrt{y^2-c}+y) = x+k \mid \text{Solve for } y$$

$$y = \frac{e^{\frac{-x-k}{\sqrt{c}}}\cdot\left(e^{\frac{2x+2k}{\sqrt{c}}}+c\right)}{2} = \frac{e^{\frac{x+k}{\sqrt{c}}}+c\cdot e^{\frac{-x-k}{\sqrt{c}}}}{2} \mid \text{Substitute } c \rightarrow \frac{1}{c^2}$$

$$y = \frac{c^2\cdot e^{c(x+k)} + e^{c(-x-k)}}{2c^2}$$

The reader is friendly invited to double check the solution.

Example 24:

$$y\cdot y'' + 1 + y'^2 = 0$$

$$v = y' \text{ and } y'' = \frac{dv}{dx} = \frac{v\cdot dv}{dy} \rightarrow \frac{y\cdot v\cdot dv}{dy} + 1 + v^2 = 0$$

$$\frac{dy}{y} = -\frac{v\cdot dv}{1+v^2} \mid \text{Integrate}$$

$$\ln y = -\frac{1}{2}\ln(1+v^2) + \frac{1}{2}\ln c \rightarrow y = \sqrt{\frac{c}{1+v^2}}$$

$$y^2 = \frac{c}{1+v^2} \quad | \text{ Solve for } v = y'$$

$$v = \frac{dy}{dx} = \pm\sqrt{\frac{c}{y^2} - 1} = \pm\frac{\sqrt{c-y^2}}{y} \quad | \text{ I take the positive root and separate the variables}$$

$$\frac{dy \cdot y}{\sqrt{c-y^2}} = dx \quad | \text{ Integrate again}$$

$$-\sqrt{c-y^2} = x + k \quad | \text{ Solve for } y$$

$$y = \pm\sqrt{c-(x+k)^2}$$

I asked *MATHEMATICA* to do the job:

```
In[2]:= DSolve[y[x] * y''[x] + 1 + y'[x]^2 == 0, y[x], x]
```

Out[2]= $\left\{\left\{y(x) \to -\sqrt{-x^2 - 2c_2 x + e^{2c_1} - c_2^2}\right\}, \left\{y(x) \to \sqrt{-x^2 - 2c_2 x + e^{2c_1} - c_2^2}\right\}\right\}$

Fortunately my result from above was confirmed.

The next contribution causes problems. George Douros wrote his package for DERIVE 3 and it does not run with DERIVE 6 (solve – solutions and incompatibilities with the then used characters). I found George Douros email-address in the Internet and asked if there is a recent version available. I received an answer the following morning:

Hello Josef Boehm!

I am glad to hear from you... and sorry that I cannot be of help.

I had written a much more powerful ODE package (at the same time with the Special Functions package) for Derive 6. Some novelties in the simplification machine of Derive (I then had a few arguments with Albert Rich) prevented my new package from becoming functional. I have since (the end of 2003) stopped working on the package.

Feel free to do whatever you think useful with the article or the package itself. My new demanding interests (ancient scripts...) do not allow me to make even a vague promise that I will be of some help to you.

George Douros

You can download the files and maybe that somebody can adapt the huge package – at least partially – for the recent DERIVE versions?

I recommend the website demonstrating George´s field of interest (ancient scripts).

http://users.teilar.gr/~g1951d/

Differential Equations with *Derive*®

George Douros
Technological Education Institute of Larissa
Larissa 41110, Greece

Abstract

This is a presentation of a new *Derive* package for solving Differential Equations: `ODE.MTH`. The aim is to show how to use it and to explain some of the mathematics behind it, so that users can expand it, or use it as an educational aid. The package currently covers 1st and 2nd order Ordinary Differential Equations, both linear with variable coefficients and nonlinear.

1. Introduction

The first version of `ODE.MTH` appeared about a year ago and was written for *Derive 2.5*. Soft Warehouse found it extremely useful and encouraged the author to publicize and make it available to Derive users. In the mean time *Derive 3* appeared, with new programming functions, and `ODE.MTH` grew into an entirely new package. Nonlinear equations are now covered almost exhaustively, singular solutions are found together with primitives, and a wide class of nonhomogeneous 2nd order linear equations with variable coefficients that lead to solutions in terms of Bessel, Kummer and Hypergeometric functions is treated in detail.

The package, as well as this article, is based on *Derive 3*. The '*User Manual, DERIVE Version 3'* is the main reference. Other useful sources are listed in the *Additional Resources* appendix on pages 331-340 of the *Manual*.

2. The General Purpose Function: `ODE(w,x,y,y1,y2)`

There are over 280 new functions defined in the package. Most of these are auxiliary, *local* functions. Some can be used independently. All are, however, auxiliaries to a single function, which solves ODEs of the 1st and 2nd order. This function, in terms of the default variables, is:

```
ODE(w,x,y,y1,y2)
```

where, `w` is any equation (or function) involving the independent variable `x`, the unknown dependent variable `y`=$y(x)$ and its derivatives `y1`=$y'(x)$ and `y2`=$y''(x)$. For example, to solve the equation $y''(x)+y'(x)-3y(x)=2x^2-\cos x$, enter

```
ODE(y2+y1-3*y=2*x^2-COS(x))
```

or,

```
ODE(y2+y1-3*y-2*x^2+COS(x))
```

To override the default variables in ODE all arguments must be entered explicitly. For example, to solve the equation $u'(t)+u(t)=t^2-3$, enter

```
ODE(y1+u=t^2-3,t,u,y1)
```

or,

```
ODE(u_+u=t^2-3,t,u,u_)
```

Many of the methods in the package, use solutions of algebraic and/or transcendental equations. It is, therefore, s*trongly recommended* that the command: Manage-Branch-Any (this is actually the default DERIVE.INI setting) be issued before attempting to solve most ODEs which contain symbolic parameters. Exponents of symbolic powers x^n of the independent variable *x*, appearing in the coefficients of *y*, *y'* and *y''*, in linear 2nd order ODEs must be declared either positive or negative, since ODE needs to test the behaviour of the equation at its singular points.

2.1. Solutions in terms of Special Functions

Many 2nd order ODEs will lead to solutions involving Special Mathematical Functions. These functions (Bessel, Kummer, Hypergeometric) are defined as arbitrary in the package and will not simplify. They can be simplified by loading the additional package FUN.MTH, *after* ODE.MTH.

Loading both ODE and FUN in one session, will use up most of the available system memory in a plain 640K Derive session. In such cases the user can save the solutions, returned by ODE, and simplify them by loading FUN.MTH in a new session. No such problems occur with *DeriveXM*.

2.2. Local Variables and Functions

I know of no way to introduce local variables or functions in a Derive package. I have therefore used its ability to 'understand' almost all ASCII characters in order to 'imitate' protected local variables and functions. This is done in the file ODE.ASC, where all variables are translated to ASCII characters above 180. The default variables (x,y,y1,y2) are left unchanged only in the 'global functions': ELIMINATE, RICCATI, GENERAL, ODE and the TRANSFORM_ utilities, to reduce typing. If one uses only the default variables, it suffices, for example, to enter

```
ODE((y1-y/x)^2-x*y2)
```

instead of

```
ODE((y1-y/x)^2-x*y2,x,y,y1,y2)
```

3. Equations of the 1st Order

3.1. Equations linear in *y'(x)*

The package uses the 'standard' techniques, but also looks for *symmetries* that allow simplifying transformations. For example, the equation $f\left(\frac{xy'}{x^n},\frac{y}{x^n}\right)=0$ is invariant under the transformation $[x,y] \to [\alpha x, \alpha^n y]$. This suggests that the transformation $y \to x^n Y$ will simplify the original DE. If we transform the solved form of the above equation $\frac{x y'}{x^n} = f\left(\frac{y}{x^n}\right)$, we get

```
#1:   TRANSFORM_Y(x*y1/x^n=f(y/x^n),x^n*y)

#2:   y1*x+n*y=f(y)
```

which is a simple separable equation. ODE finds the appropriate n by computing $n = -\frac{x\frac{df}{dx}}{y\frac{df}{dy}}$. This is all done automatically without user intervention. Take, for example, the DE $y' = \frac{1}{y+\sqrt{x}} + \frac{y}{2x}$

```
#3:   ODE(y1=1/(y+√(x))+y/(2*x))

                   2
        y/√x  +  y  /(2·x)
#4:    ê
      ─────────────────────────  =  c1_
                 x
```

Extensive search for integrating factors, of various forms, is performed to make equations exact. Take for example the equation $x^2 y^3 + x(1+y^2)y' = 0$, which is not exact

```
#5:   EXACT_1ST(x^2*y^3+x*(1+y^2)*y1)

#6:   ¿
```

ODE however, finds an integrating factor $\mu(x,y) = \frac{1}{x\,y^3}$ (try it) and solves the equation

```
#7:   ODE(x^2*y^3+x*(1+y^2)*y1)

                  2
                 x         1
#8:   LN(y)  +  ────  -  ──────  =  c1_
                 2          2
                         2 y
```

3.2. Equations of degree higher than one in *y'(x)*

For equations of degree higher than one in *y'(x)* ODE tries to solve for one of the variables *x, y, y'* and proceed from there. The equation $y^2 y'^2 + 3xy' = y$ has the solution

```
#9:   ODE(y^2*y1^2+3*x*y1=y)

        ┌  4·c1_              2·c1_                                   ┐
        │ ê            3·x·ê                          2         3      │
#10:    │ ─────────  +  ──────────────  -  y  =  0,  9·x   +  4·y   = 0 │
        │    2               2                                         │
        └   y               y                                         ┘
```

or equivalently,

$$\#11:\quad [3\cdot c1\_\cdot x - y^3 + c1\_^2 = 0,\ 9\cdot x^2 + 4\cdot y^3 = 0]$$

which was actually obtained by choosing to solve the original equation for x (this is simpler than solving for y or y'). This can be seen by issuing the command

```
#11: X_SOLVABLE(y^2*y1^2+3*x*y1-y)
```

Note that ODE also appends the singular solution, when it can be found, to the primitive. In this case $9x^2+4y^3=0$ is the singular solution of $y^2y'^2+3xy'=y$.

The solution of higher degree equations is usually found in parametric form. ODE does not automatically attempt to eliminate the parameter, because, in its present form, it may lose parts of the solution. The user can in such cases use ELIMINATE to simplify the solution.

```
#12: ODE(2*y1*x-y*y1^2-y)
```

$$\#13:\quad \left[\left[x = \frac{c1\_\cdot(\alpha^2+1)}{\alpha^2},\ y = \frac{2\cdot c1\_}{\alpha}\right],\ y^3 - x^2\cdot y = 0\right]$$

```
#14: ELIMINATE(#13,α)
```

$$\#15:\quad \left[x = \frac{y^2 + 4\cdot c1\_^2}{4\cdot c1\_},\ y^3 - x^2\cdot y = 0\right]$$

4. Equations of the 2nd Order

With second order equations of some complexity, ODE starts by assuming that the equation is given in an *unnatural* form; a simple DE has undergone transformations in both the dependent and independent variables resulting in the given complicated equation. A search is initiated for the reverse transformations that will recover the original simple DE. Three major methods are used: changing the independent variable x, reducing the equation to its canonical form, and scaling transformations (symmetries) in the variables x, y, y', y''.

4.1. Changing the Independent Variable *x*

Asking ODE to solve the equation $\sin^2(x)y''+\sin(x)\cos(x)y'+4y=0$, we get

```
#16: equ_1:=SIN(x)^2*y2+SIN(x)*COS(x)*y1+4*y

#17: ODE(equ_1=0)
```

$$\#18:\quad y = c1\_\cdot \text{COS}\left[2\cdot \text{LN}\left[\text{TAN}\left[\frac{x}{2}\right]\right]\right] + c2\_\cdot \text{SIN}\left[2\cdot \text{LN}\left[\text{TAN}\left[\frac{x}{2}\right]\right]\right]$$

How did ODE proceed? When the solution algorithm reached the internal function CHANGE_X, equ_1 was transformed by letting $x \to 2\arctan e^{\frac{x}{2}}$ (computed from the coefficients of y and y''),

```
#19: INVERSE(INT(√(DIF(equ_1,y)/DIF(equ_1,y2)),x),x)

                 x/2
#20: 2·ATAN(ê       )

#21: TRANSFORM_X(equ_1,#20)

#22:  4·(y + y2)
```

which is a simple equation to solve.

4.2. Reducing an Equation to its Canonical Form

The canonical form, $y'' + \frac{1}{4p^2}\left(2qp' - 2pq' + 4pr - q^2\right)y = 0$, of the differential equation $equ1 := py'' + qy' + ry = 0$ is obtained by letting $y \to ye^{\phi}$, where $\phi = -\frac{1}{2}\int \frac{q}{p} dx$.

It can easily be shown that equation $equ2 := p\{f\,y\}'' + q\,\{f\,y\}' + r\,f\,y = 0$ has exactly the same canonical form as $equ1$ and that its solution is a multiple of the solution of $equ1$.

This fact is used by ODE to solve equations like *equ*2, by reducing them to their canonical form and proceed from that point by pattern matching with canonical forms of equations with known solutions.

Equation hyp_ode is a simple case of a hypergeometric differential equation and is directly solved by ODE.

```
#22: hyp_:=2*x^2*(1-x)*y2-x*(2*x+3)*y1+(2*x+3)*y

#23: ODE(hyp_)
```

$$\#24:\; y = x\cdot\left[c1\_\cdot\sqrt{x}\cdot F21\left[\frac{5}{2},\frac{1}{2},\frac{3}{2},x\right] + c2\_\cdot F21\left[2,0,\frac{1}{2},x\right]\right]$$

Suppose we transform hyp_ by first letting $x \to x^k$ and then $y \to \dfrac{y}{\Omega(x)}$. The resulting equation looks awesome. This is what we previously characterized as an *unnatural* form. By reducing this equation to its canonical form, however, ODE can solve it in a natural way, as if only by making substitutions in #24.

```
#25: k^2*TRANSFORM_X(hyp_,x^k)

          2          k                  k                      2        k
#26: 2·x  ·(1 - x )·y2 - x·(2·x  + 5·k - 2)·y1 + k ·(2·x  + 3)·y

#27: Ω(x)^3*TRANSFORM_Y(#26,y/Ω(x))
```

$$\#28:\ 2x^2(1-x^k)\Omega(x)^2\,y2+$$

$$x\Omega(x)\left[2x^k\left[2x\frac{d}{dx}\Omega(x)-\Omega(x)\right]-4x\frac{d}{dx}\Omega(x)+(2-5k)\Omega(x)\right]y1+$$

$$\left[2x^k\left[x^2\,\Omega(x)\left[\frac{d}{dx}\right]^2\Omega(x)-2x^2\left[\frac{d}{dx}\Omega(x)\right]^2+x\Omega(x)\frac{d}{dx}\Omega(x)+k^2\,\Omega(x)^2\right]-\right.$$

$$\left.2x^2\,\Omega(x)\left[\frac{d}{dx}\right]^2\Omega(x)+4x^2\left[\frac{d}{dx}\Omega(x)\right]^2+x(5k-2)\Omega(x)\frac{d}{dx}\Omega(x)+3k^2\,\Omega(x)^2\right]y$$

Before solving the above equation, *k* must be declared positive (or negative) because `ODE` needs to test the behaviour of the equation at its singular points, and *Derive* is ‘reluctant’ to perform certain simplifications without this assumption for *k*. We also (optionally) declare *x* to be positive so that the solution `#31` be fully simplified to look exactly like `#24`.

```
#29: [x:εReal (0,∞),k:εReal (0,∞)]

#30: ODE(#28)
```

$$\#31:\ y = x^k\,\Omega(x)\left[c1\_\ x^{k/2}\ F21\left[\frac{5}{2},\frac{1}{2},\frac{3}{2},x^k\right]+c2\_\ F21\left[2,0,\frac{1}{2},x^k\right]\right]$$

4.3. Searching for Symmetries

The other major method used to bring an equation to its *natural* form is the search for symmetries in the dependent and independent variables. Consider, for example, the readily integrable equation `#32`, where $\Omega(t)$ is an arbitrary function. If `#32` is subjected to the transformations `#33`–`#37`, the resulting equation `#38` is in what we called an *unnatural* form. `ODE` can be impressive in solving it, only because it ‘realizes’ that `#38` is invariant under the transformations $(x,y)\to(\alpha x,\beta y)$ and automatically reverses `#33`–`#37`, to recover the original natural form.

```
#32: y2-Ω(y1)

#33: TRANSFORM_X(y2-Ω(y1),LN(x))
```

$$\#34:\ -\ \Omega(y1\cdot x)\ +\ y2\cdot x^2\ +\ y1\cdot x$$

```
#35: TRANSFORM_Y(#34,LN(y))
```

$$\#36:\ -\ \Omega\left[\frac{y1\cdot x}{y}\right]\ +\ x^2\cdot\left[\frac{y2}{y}\ -\ \frac{y1^2}{y^2}\right]\ +\ \frac{y1\cdot x}{y}$$

```
#37: NUMERATOR(FACTOR(#36,Trivial))

                         2               2 ┌ y1·x ┐
#38: x·(x·(y2·y - y1 ) + y1·y) - y ·Ω│──────│
                                           └  y   ┘

#39: ODE(#38)

          ⌠   1                                    ⌠
#40: [[ lim    │ ──── du ]-x=c1_, LN(y)=[ lim      │ƒ(t) dt ]+c2_]
       u->ƒ(x)⌡ Ω(u)                     t->LN(x) ⌡
```

5. Additional Utilities

The package makes internal use of some utilities that the end user may find useful. These utilities are:

- `ELIMINATE(v,α)` tries to eliminate α from a parametric set $v=[v_1(\alpha,\ldots),v_2(\alpha,\ldots),\ldots]$. To simplify, for example, the solution of the equation $xy'^2+4x=2yy'$, which, as given by `ODE`, is: $\left[\left[x=\alpha c_1, y=\frac{1}{2}c_1\left(\alpha^2+2\right)\right], y^2-4x^2=0\right]$, enter

  `ELIMINATE([[x=α*c1_,y=c1_*(α^2+4)/2],y^2-4*x^2=0],α)`

- `TRANSFORM_X(w,t(x),x,y,y1,y2)` will transform the equation `w`, when $x\to t(x)$. To transform, for example, the equation $y''+\cot(x)y'+4\csc^2(x)y=0$, by letting $x\to\arccos(x)$, enter

  `TRANSFORM_X(y2+COT(x)*y1+4*CSC(x)^2*y,ACOS(x))`

- `TRANSFORM_Y(w,Ω(x,y),x,y,y1,y2)` will transform the differential equation `w`, when $y(x)\to\Omega(x,Y(x))$. To transform, for example, the equation $y''-(x^2+1)y=0$, by letting $y\to e^{\frac{1}{2}x^2}Y(x)$, enter

  `TRANSFORM_Y(y2-(x^2+1)*y,y*ê^(x^2/2))`

- `TRANSFORM_P(w,x,y,y1,Θ,r,r1)` will transform the 1st order differential equation `w`, when $(x,y(x))\to r(\theta)\cdot(\cos\theta,\sin\theta)$. To transform, for example, the differential equation $(y'^2+1)\cdot(x-y)^2=(x+yy')^2$, enter

  `TRANSFORM_P((y1^2+1)*(x-y)^2=(x+y*y1)^2)`

- `RICCATI(w,s(x),x,y,y1)` will return a solution of the Riccati DE `w`, which is more general than a known particular solution, *s(x)*. For example, if $y=\frac{1}{x}$ is a known solution of the Riccati equation $y'+\frac{1}{x^2}+\frac{y}{x}=y^2$, to find a more general one, enter

  `RICCATI(y1+1/x^2+y/x=y^2,1/x)`

- `GENERAL(w,[1st(x),2nd(x)],x,y,y1,y2)` will return the general solution of $\mathrm{w} = p(x)y'' + q(x)y' + r(x)y + f(x) = 0$, if one or two solutions of the homogeneous equation are known. If two solutions are known, they must be entered as a vector `[1st(x),2nd(x)]`. If one solution is known, it may be entered as `[1st(x),0]`, or simply `1st(x)`. For example, to find the general solution of $xy''(x) + xy'(x) = y(x) + x$, if $y=x$ is a known homogeneous solution, enter

  `GENERAL(x*y2+x*y1=y+x,[x,0])`

6. Conclusion

ODE.MTH is more a mathematics than a programming package. I am a mathematician, not a programmer. I have tested it against similar packages in other symbolic mathematics programs and found it competitive. I will not make any further claims on its speed and efficiency and leave its evaluation to those who will use it. This not simple modesty; it is the *fear and dread* of the easy counter-example to any of my claims. I have been faced with many of those in writing this package. With all due respect to the power given by the programming functions in the symbolic mathematics programs, I have been taught, once more, that a mathematical problem can be solved only by *mathematics* and not by programming functions. The latter are assistants; and *Derive* is *A Mathematical Assistant* par excellence!

The hints, given here, on how ODE.MTH works are obviously not enough to help a user expand, improve, or fully exploit it in a Differential Equations course, but they have made this … a long article.

ODE.MTH, together with its accompanying files, is available via the Derive Internet Mailing List. See the *Additional Resources* appendix of the *Derive Manual*.

Office:
Prof. George Douros
T.E.I. of Larissa
Larissa 41110
GREECE

Tel: 041-611-061 … 72
Fax: 041-610-803

Home:
George Douros
Kolokotroni 3
Larissa 41223
GREECE

Tel: 041-234-866

Asymptotically stable Solutions of the Systems of Ordinary Differential Equations

A. Kozubík, Bratislava, Slovakia

A given system of ordinary differential equations of the first order can be a mathematical model for a number of mechanic, biologic, economic etc. systems. The behaviour of this system is described by the solution of the respective system. For most of the systems we require that their behaviour will be "similar" to the behaviour of some given system. This requirement is described exactly by the stability of the solution.

Any system

$$y' = g(t, y) \tag{1}$$

can be reduced by the transformation $y = x + v$ where v is any solution of (1) into the system

$$x' = f(t, x),\ \ f(t, o) = o \tag{2}$$

where

$$f(t, x) = g(t, x + v) - g(t, v).$$

System (2) has the trivial solution, which is the transform of the solution v of system (1). So, we can deal with the stability of the trivial solution of system (2).

The trivial solution of system (2) is said to be stable iff for any τ and any ε there exists $\delta = \delta(\tau,\varepsilon)$ such that for all initial values ξ satisfying $\|\xi\| < \delta$ and for all $t \geq \tau$ the solution $u(t;\tau,\xi)$ of the initial problem

$$x' = f(t, x), \quad x(\tau) = \xi$$

satisfies the inequality

$$\| u(t;\tau,\xi) \| < \varepsilon.$$

The trivial solution of system (2) is said to be uniformly stable if number δ does not depend on the initial point τ. Moreover, the trivial solution is said to be uniformly asymptotically stable, if there exists a number $\Delta > 0$ such that for all ξ; $\|\xi\| < \Delta$ the condition

$$\lim_{t\to\infty} \| u(t;\tau,\xi) \| = 0 \qquad \textit{uniformly for all } \tau$$

holds.

In this paper we deal with the linear system with constant coefficients in the form

$$x' = A x \tag{3}$$

where A is a real constant (n,n)-matrix. In this case we can apply the following assertions:

Theorem 1. The trivial solution of system (3) is uniformly asymptotically stable iff the real parts of all eigenvalues of matrix *A* are negative.

Let

$$P(x) = a_0 + a_1 x + ... + a_{n-1}x^{n-1} + a_n x^n \qquad n \geq 1, a_0 > 0, a_n \neq 0 \qquad (4)$$

is a given polynomial with real coefficients.

The matrix $\begin{pmatrix} a_1 & a_2 & 0 & 0 & 0 & ... & 0 \\ a_3 & a_2 & a_1 & a_0 & 0 & ... & 0 \\ ... & ... & ... & ... & ... & ... & ... \\ a_{2n-1} & a_{2n-2} & a_{2n-3} & ... & ... & ... & a_n \end{pmatrix}$ is said to be the Hurwitz matrix of polynomial (4).

Theorem 2. (Hurwitz Criterion) Real parts of all roots of the polynomial (4) are negative iff all main diagonal minors of the Hurwitz matrix are positive. It means that

$$D_1 = a_1 > 0$$

$$D_2 = \begin{vmatrix} a_1 & a_0 \\ a_3 & a_2 \end{vmatrix} > 0$$

$$D_3 = \begin{vmatrix} a_1 & a_0 & 0 \\ a_3 & a_2 & a_1 \\ a_5 & a_4 & a_3 \end{vmatrix} > 0$$

....

$$D_n = a_n \cdot D_{n-1} > 0$$

The following sequence is the Hurwitz criterion realised with *DERIVE*.

```
#1:   a := []

      POSITIVE(x) :=
        If x > 0
#2:        1
           0

      MAIN_MINOR(a, k, n) :=
        If k = n
#3:        a
           MINOR(MAIN_MINOR(a, k + 1), k + 1, k + 1)

#4:   Q(x, a) := CHARPOLY(a, x)

      P(x, a) :=
        If Q(0, a) > 0
#5:        Q(x, a)
           - Q(x, a)

#6:   n := DIMENSION(a)

#7:   vec_coef := VECTOR(ITERATE((d/dx)^k P(x, a) / k!, x, 0, 1), k, 0, n)
```

#8: zero_vec := VECTOR(0, k, n)

#9: base := APPEND(zero_vec, APPEND(vec_coef, zero_vec))

#10: HURWITZ_ROW(base, i, n) := VECTOR(ELEMENT(base, n + 2·i − j + 1), j, 1, n)

#11: hurwitz_matrix := VECTOR(HURWITZ_ROW(base, i, n), i, 1, n)

#12: hurwitz := VECTOR(POSITIVE(DET(MAIN_MINOR(hurwitz_matrix, k))), k, 1, n)

#13: $\text{test} := \sum_{i=1}^{n} \text{ELEMENT(hurwitz, i)}$

#14: stability? := IF(test = n, asymptotically stable solution, asymptotically unstable solution)

I declare the coefficient matrix of the system:

#15: $a := \begin{bmatrix} -4 & 1 \\ -5 & 0 \end{bmatrix}$

The characteristic polynomial of the matrix is:

#16: $P(x, a) = x^2 + 4 \cdot x + 5$

This is the vector of coefficients of the polynomial:

#17: vec_coef = [5, 4, 1]

The Hurwitz matrix of the polynomial:

#18: $\text{hurwitz\_matrix} = \begin{bmatrix} 4 & 5 \\ 0 & 1 \end{bmatrix}$

Vector (List) of the determinants of the main diagonal minors. Its elements are equal to one if the determinant is positive and equal to zero if it is not positive.

#19: hurwitz = [1, 1]

Now I will test if the trivial solution is asymptotically stable:

#20: stability? = asymptotically stable solution

Another example:

#21: $a := \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$

#22: [P(x), vec_coef, hurwitz_matrix, hurwitz]

#23: $\left[x^2 - 2 \cdot x + 2,\ [2, -2, 1],\ \begin{bmatrix} -2 & 2 \\ 0 & 1 \end{bmatrix},\ [0, 0] \right]$

#24: stability? = asymptotically unstable solution

And finally a third example:

#25: $a := \begin{bmatrix} -1 & 2 & 0 \\ -3 & -1 & 2 \\ 0 & -3 & -1 \end{bmatrix}$

#26: [P(x), vec_coef, hurwitz_matrix, hurwitz]

#27: $\left[(x + 1) \cdot (x^2 + 2 \cdot x + 13),\ [13, 15, 3, 1],\ \begin{bmatrix} 15 & 13 & 0 \\ 1 & 3 & 15 \\ 0 & 0 & 1 \end{bmatrix},\ [1, 1, 1] \right]$

#28: stability? = asymptotically stable solution

Müller´s Method for Solving a Univariate Equation

G P Speck, Wanganui, New Zealand

(A short background information [Numerical Analysis with DERIVE, St. Schonefeld, p 53 – 70] : Müller´s method is an extension of the secant method. The secant method approximates *F*(*x*) by a straight line (first degree polynomial) in order to find an approximation to a root of a function. With Müller´s method *F*(*x*) is approximated by a quadratic polynomial. Josef)

Müller´s Method (see the Reference at the end of this note) for solving univariate equations has some very significant advantages over many numerical methods available. In particular, the often troublesome problem of finding a guess reasonably "close" to a solution sought, especially in the complex number case, is not a critical issue in using Müller´s Method. One (possible) objection to the method is that convergence can be relatively slow compared to some other numerical methods – however, in many problems where this is an issue, Müller´s Method can be used with a small number of iterations to produce quite adequate initial guesses for solutions which can be found using one of the fast methods for which reasonable initial guesses are required.

In the *DERIVE* program for Müller´s Method which follows, the example $x^3 - 1000 = 0$ is given as an illustration and solved in detail to show exactly how the program is used. The exact solutions to $x^3 - 1000 = 0$ are easily computed to be 10, $-5 + 5i\sqrt{3}$, and $-5 - 5i\sqrt{3}$, so that the three solutions found in succession using Müller´s Method can be compared with these exact solutions. As a more realistic, non-trivial example, the reader could try the problem on page 192 of the *DERIVE* manual (for version 2.60: $e^z - z^2 = 0$) after viewing Müller´s Method applied to $x^3 - 1000 = 0$ on the program following. The advantages of Müller`s Method over the classical Newton-Raphson Method as employed in the manual would then be apparent.

The brief Müller program that follows could be made even more compact, but there are advantages in a more lengthy display in terms of converting formulas in Müller´s Method literature to a *DERIVE* program.

```
#1:   [Precision := Approximate, PrecisionDigits := 15]

#2:   [Notation := Decimal, NotationDigits := 10]

#3:   F(x) :=

      ⎡x0 := m , x1 := m , x2 := m , p := m ⎤
#4:   ⎣       1         2         3        4⎦

      ⎡      F(x2) - F(x1)        F(x1) - F(x0)        F(x2) - F(x1)
#5:   ⎢ca := ─────────────, cb := ─────────────, ca := ─────────────, cb :=
      ⎣        x2 - x1              x1 - x0              x2 - x1

        F(x1) - F(x0) ⎤
        ───────────── ⎥
           x1 - x0    ⎦

                               2                              2                       2
#6:   ma := IF(|c2 + √(c2  - 4·F(x2)·c)| ≥ |c2 - √(c2  - 4·F(x2)·c)|, c2 + √(c2

                                      2
        - 4·F(x2)·c), c2 - √(c2  - 4·F(x2)·c))
```

#7: $\left[x3 := x2 - \frac{2 \cdot F(x2)}{ma}, u := [x1, x2, x3, p + 1]\right]$

#8: MULLER(m0, n) := APPEND([[x1, x2, x3, p]], ITERATES(u, m, m0, n))

#9: $F(x) := 1000 - x^3$

#10: MULLER([-0.5, 0, 0.5, 0], 12)

#11:

x1	x2	x3	p
-0.5	0	0.5	0
0	0.5	4000	1
0.5	4000	0.80894382	2
4000	0.80894382	1.1774703	3
0.80894382	1.1774703	1.5253355	4
1.1774703	1.5253355	17.43907	5
1.5253355	17.43907	8.2551095	6
17.43907	8.2551095	9.6881509	7
8.2551095	9.6881509	9.9871208	8
9.6881509	9.9871208	10.000023	9
9.9871208	10.000023	10	10
10.000023	10	10	11
10	10	10	12

#12: z1 := 10

#13: $F(x) := \frac{1000 - x^3}{x - z1}$

#14: MULLER([-0.5, 0, 0.5, 0], 5)

#15:

x1	x2	x3	p
-0.5	0	0.5	0
0	0.5	-5 - 8.660254·ì	1
0.5	-5 - 8.660254·ì	-5 - 8.660254·ì	2
-5 - 8.660254·ì	-5 - 8.660254·ì	-5 - 8.660254·ì	3
-5 - 8.660254·ì	-5 - 8.660254·ì	-5 - 8.660254·ì	4
-5 - 8.660254·ì	-5 - 8.660254·ì	-5 - 8.660254·ì	5

#16: z2 := -5 - 8.660254·ì

#17: $F(x) := \frac{1000 - x^3}{(x - z1)\cdot(x - z2)}$

```
#18:  MULLER([-0.5, 0, 0.5, 0], 5)
```

#19:

x1	x2	x3	p
-0.5	0	0.5	0
0	0.5	-5 + 8.660253999·i	1
0.5	-5 + 8.660253999·i	-5 + 8.660254037·i	2
-5 + 8.660253999·i	-5 + 8.660254037·i	-5 + 8.660254037·i	3
-5 + 8.660254037·i	-5 + 8.660254037·i	-5 + 8.660254037·i	4
-5 + 8.660254037·i	-5 + 8.660254037·i	-5 + 8.660254037·i	5

```
#20:  [z1 = 10, z2 = -5 - 8.660254·i, z·3 = -5 + 8.660254·i]

                     3
#21:  F(x) := 1000 - x

#22:  MULLER([-4, 2, 5, 0], 10)
```

#23:

x1	x2	x3	p
-4	2	5	0
2	5	15.85912687	1
5	15.85912687	9.266222135	2
15.85912687	9.266222135	9.934452212	3
9.266222135	9.934452212	9.999074829	4
9.934452212	9.999074829	10.00000014	5
9.999074829	10.00000014	10	6
10.00000014	10	10	7
10	10	10	8
10	10	10	9
10	10	10	10

References

[1] Samuel D. Conte and Carl deBoor, **Elementary Numerical Analysis,** An Algorithmic Approach, McGraw-Hill Kogakusha, Ltd, Auckland, 1980, pp 120 – 127

[2] Steven Schonefeld, **Numerical Analysis via *DERIVE*,** MathWare, Urbana, 1994, pp 53 – 70.

Some ideas for the revised Version of this DNL:

(1) G P Speck writes about a “program” for demonstrating and performing Müller´s Method. In 1995 it was a huge progress to have a list of functions – calling each other – working in the sense of a program. Now we can write real programs as a whole. I´ll collect all procedures in one program without loosing the insight into the single steps. Application of ITERATES has an advantage in demonstrating the iterative nature of the process but the disadvantage that its syntax is sometimes difficult for students to understand. A simple loop might be easier to follow. This realized in my program version.

```
      m_meth(u, v, n, ca, cb, c, c2, ma, x4, dis, table) :=
        Prog
          table := ["x1", "x2", "x3", "p"; v↓1, v↓2, v↓3, 0]
          n_ := 1
          Loop
            If n_ > n
               RETURN table
            ca := (SUBST(u, x, v↓3) - SUBST(u, x, v↓2))/(v↓3 - v↓2)
#1:         cb := (SUBST(u, x, v↓2) - SUBST(u, x, v↓1))/(v↓2 - v↓1)
            c := (ca - cb)/(v↓3 - v↓1)
            c2 := ca + c·(v↓3 - v↓2)
            dis := √(c2^2 - 4·SUBST(u, x, v↓3)·c)
            ma := IF(ABS(c2 + dis) ≥ ABS(c2 - dis), c2 + dis, c2 - dis)
            x4 := v↓3 - 2·SUBST(u, x, v↓3)/ma
            v := [v↓2, v↓3, x4]
            table := APPEND(table, [[v↓1, v↓2, v↓3, n_]])
            n_ :+ 1
```

#2: $(\text{m\_meth}(1000 - x^3, [-0.5, 0, 0.5], 12))_{[12, 13, 14]}$

#3: $\begin{bmatrix} 9.987121004 & 10.00002340 & 10 & 10 \\ 10.00002340 & 10 & 10 & 11 \\ 10 & 10 & 10 & 12 \end{bmatrix}$

#4: $\text{m\_meth}\left(\dfrac{1000 - x^3}{x - 10}, [-0.5, 0, 0.5], 5\right)$

#5: $\begin{bmatrix} x1 & x2 & x3 & p \\ -0.5 & 0 & 0.5 & 0 \\ 0 & 0.5 & -5 - 8.660254037\cdot i & 1 \\ 0.5 & -5 - 8.660254037\cdot i & -5 - 8.660254037\cdot i & 2 \\ -5 - 8.660254037\cdot i & -5 - 8.660254037\cdot i & -5 - 8.660254037\cdot i & 3 \\ -5 - 8.660254037\cdot i & -5 - 8.660254037\cdot i & -5 - 8.660254037\cdot i & 4 \\ -5 - 8.660254037\cdot i & -5 - 8.660254037\cdot i & -5 - 8.660254037\cdot i & 5 \end{bmatrix}$

The next equation has only complex zeros. See the first steps

#10: $\text{m\_meth}(x^4 + 7\cdot x^2 + 10, [-10, 0, 10], 20)$

#11: $\text{FIRST}(\text{REVERSE}(\text{m\_meth}(x^4 + 7\cdot x^2 + 10, [-10, 0, 10], 20)))$

#12: $[7.633765896\cdot 10^{-108} + 1.414213562\cdot i, 4.034339544\cdot 10^{-118} + 1.414213562\cdot i, 2.132092571\cdot 10^{-128} + 1.414213562\cdot i, 20]$

#13: $\text{FIRST}\left(\text{REVERSE}\left(\text{m\_meth}\left(\dfrac{x^4 + 7\cdot x^2 + 10}{x - 1.414213562\cdot i}, [-10, 0, 10], 20\right)\right)\right)$

#14: $[- 5.066365831\cdot 10^{-14} - 1.414213561\cdot i, - 3.683475968\cdot 10^{-14} - 1.414213561\cdot i, - 1.0368848\cdot 10^{-15} - 1.414213561\cdot i, 20]$

Neglect the tiny real part, then the solutions are {-$i\sqrt{2}$, $i\sqrt{2}$, -$i\sqrt{5}$, $i\sqrt{5}$}.

In the next example the imaginary part can be taken as zero.

$(m\_meth(e^x - x^2, [-10, 0, 10], 20))_{[20, 21, 22]}$

$$\begin{bmatrix} -0.70346742 + 1.4981314\cdot10^{-19}\cdot i & -0.70346742 - 2.7887403\cdot10^{-19}\cdot i & -0.70346742 - 6.5523512\cdot10^{-28}\cdot i & 18 \\ -0.70346742 - 2.7887403\cdot10^{-19}\cdot i & -0.70346742 - 6.5523512\cdot10^{-28}\cdot i & -0.70346742 - 2.0078468\cdot10^{-20}\cdot i & 19 \\ -0.70346742 - 6.5523512\cdot10^{-28}\cdot i & -0.70346742 - 2.0078468\cdot10^{-20}\cdot i & -0.70346742 - 3.9241406\cdot10^{-20}\cdot i & 20 \end{bmatrix}$$

$(m\_meth(x^3 - 10 + 5\cdot i, [-10, 0, 10], 20))_{[21, 22]}$

$$\begin{bmatrix} 2.2094163 - 0.34420843\cdot i & 2.2094163 - 0.34420843\cdot i & 2.2094163 - 0.34420843\cdot i & 19 \\ 2.2094163 - 0.34420843\cdot i & 2.2094163 - 0.34420843\cdot i & 2.2094163 - 0.34420843\cdot i & 20 \end{bmatrix}$$

(2) I wanted to illustrate the process by deriving the quadratic functions and plotting them together with the newly found approximation.

#24: $q(x) := a\cdot x^2 + b\cdot x + c$

#25: $f(x) := 1000 - x^3$

Initial guesses are [-4,2,5] (see page 42):

#26: $(\text{SOLUTIONS}(q(-4) = f(-4) \wedge q(2) = f(2) \wedge q(5) = f(5), [a, b, c]))_1$

These are the coefficients of the parabola:

#27: $[-3, -18, 1040]$

You may use the FIT-function to obtain the parabola, too:

#28: $\text{FIT}\left(\left[x, a\cdot x^2 + b\cdot x + c\_\right], \begin{bmatrix} -4 & f(-4) \\ 2 & f(2) \\ 5 & f(5) \end{bmatrix}\right) = -3\cdot x^2 - 18\cdot x + 1040$

#29: $\text{SOLUTIONS}(-3\cdot x^2 - 18\cdot x + 1040 = 0, x) = [15.85912687, -21.85912687]$

Which zero of the approximating parabola is the appropriate one?

#30: $[|5 - 15.85912687|, |5 - -21.85912687|] = [10.85912686, 26.85912686]$

We take the solution with the smaller distance to x3, which is 15.859...
The next parabola should pass the points (2,f(2)), (5,f(5)) and (15.859...),f(15.859...))

#31: $\text{FIT}\left(\left[x, a\cdot x^2 + b\cdot x + c\_\right], \begin{bmatrix} 2 & f(2) \\ 5 & f(5) \\ 15.85912687 & f(15.85912687) \end{bmatrix}\right)$

#32: $-22.85912686\cdot x^2 + 121.0138880\cdot x + 841.4087313$

#33: $\text{SOLUTIONS}(-22.85912686\cdot x^2 + 121.013888\cdot x + 841.4087313, x) = [-3.972324047, 9.266222136]$

#34: $[|15.85912687 - -3.972324047|, |15.85912687 - 9.266222136|] = [19.83145091, 6.592904733]$

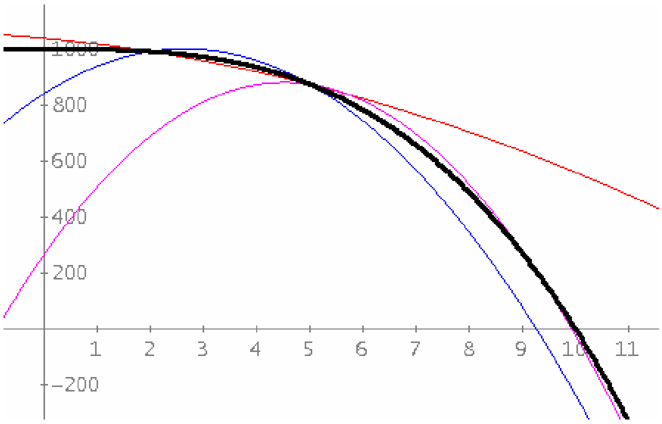
This is the third parabola:

```
#35:  FIT([x, a·x^2 + b·x + c_], [5, f(5); 15.85912687, f(15.85912687); 9.266222136, f(9.266222136)])

#36:  - 30.12534900·x^2 + 272.5809374·x + 265.2290376

#37:  SOLUTIONS(- 30.125349·x^2 + 272.5809374·x + 265.2290376, x) = [9.934452211, -0.8862271639]
```

The next figure shows the function together with the first three parabolas:



(3) Finally I´d like to work with the TI-Nspire using the spread sheet and the slider:

$f(x):=1000-x^3$ Done

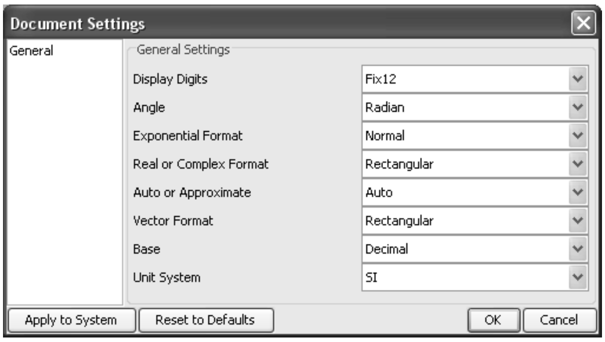
1/99

	A	B	C x1	D x2	E x3	F ca	G cb	H c	I c2	J dis	K ma
◆											
1	x1	-0.5...	-0.500...	0	0.500...	0	0	0	0	0	0
2	x2	0	0	0.5000...	4000....	-0.250...	-0.25...	0.000...	-0.250...	0.2500...	-0.500...
3	x3	0.50...	0.5000...	4000.0...	0.808...	-16002...	-0.25...	-4000...	-3200...	4472.3...	-32006...
4			4000.0...	0.8089...	1.177...	-16003...	-1600...	-4001...	-1237....	4186.6...	-5424....
5			0.8089...	1.1774...	1.525...	-2.993...	-1600...	-4001...	-1477....	4262.1...	-5739....
6			1.1774...	1.5253...	17.43...	-5.509...	-2.99...	-3.51...	-6.730...	118.50...	-125.2...
7			1.5253...	17.439...	8.255...	-333.0...	-5.50...	-20.1...	-653.5...	283.61...	-937.1...
8			17.439...	8.2551...	9.688...	-516.2...	-333....	-27.2...	-266.2...	344.25...	-610.5...
9			8.2551...	9.6881...	9.987...	-241.9...	-516....	-35.3...	-292.6...	313.84...	-606.5...
10			9.6881...	9.9871...	10.00...	-290.3...	-241....	-27.9...	-298.7...	299.43...	-598.1...
11			9.9871...	10.000...	9.999...	-299.6...	-290....	-29.6...	-299.9...	299.99...	-599.9...
12			9.9999999996868	9999...	10.00...	-300.0...	-299....	-29.9...	-299.9...	299.99...	-599.9...
13			9.9999...	10.000...	10.00...	-300.0...	-300....	-29.9...	-300.0...	300.00...	-600.0...
14			10.000...	10.000...	#UND...	#UND...	-300....	#UN...	#UND...	undef+...	piece...
15			10.000...	#UND...	#UND...	#UND...	#UN...	#UN...	#UND...	undef+...	piece...
16			#UND...	#UND...	#UND...	#UND...	#UN...	#UN...	#UND...	undef+...	piece...
17											

C13 =d12

The "undefined – messages" starting in row 14 are caused by the restricted accuracy. The values of x_2 and x_3 are so close that their difference is internally rounded to zero and then the division in column F gives an undefined result. All other error-messages are the consequence.
However, we have result in the last complete row#13.

I recommend the Document Settings as shown on the right:



The entries for the cells are:

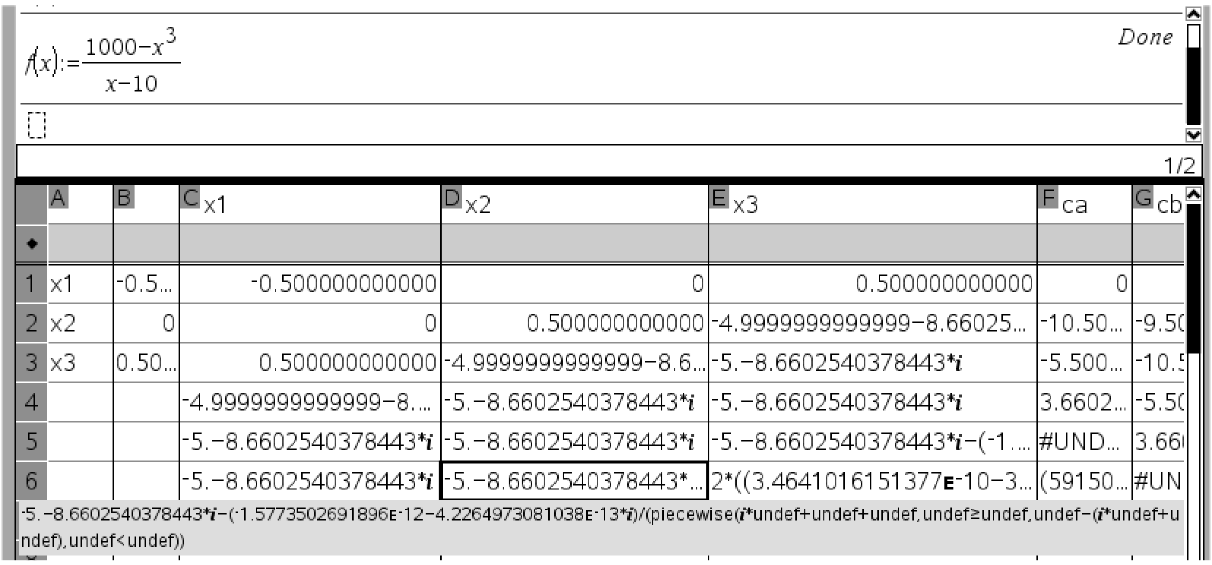
C1: =b1
D1: =b2
E1: =b3
C2: =d1
D2: =e1
F2: =(f(e1)-f(d1))/(e1-d1)

G2: =(f(d1)-f(c1))/(d1-c1)
H2: =(f2-g2)/(e1-c1)
I2: =f2+h2*(e1-d1)
J2: =√(i2^2–4*h2*f(e1))
K2: when(abs(i2+j2) ≥ abs(i2-j2),i2+j2,i2–j2)
E2: =e1–2*f(e1)/k2

I try to perform the next step according to G P Speck´s DERIVE routine in order to find the next (complex) root:

I recommend deleting some rows in the table before changing the function and/or the initial guesses. The expressions in the cells might become too bulky and the system might hang up. (See below the complete entry of cell D6 – in grey.)

It is better to start with less rows and then proceed by copying down row for row. Don´t forget to save between the steps.



$f(x):=\frac{1000-x^3}{x-10}$ Done

1/2

	A	B	C x1	D x2	E x3	F ca	G cb
◆							
1	x1	-0.5…	-0.500000000000	0	0.500000000000	0	
2	x2	0	0	0.500000000000	-4.9999999999999-8.66025…	-10.50…	-9.5(
3	x3	0.50…	0.500000000000	-4.9999999999999-8.6…	-5.-8.6602540378443*i	-5.500…	-10.5
4			-4.9999999999999-8.…	-5.-8.6602540378443*i	-5.-8.6602540378443*i	3.6602…	-5.5(
5			-5.-8.6602540378443*i	-5.-8.6602540378443*i	-5.-8.6602540378443*i-(-1.…	#UND…	3.66(
6			-5.-8.6602540378443*i	-5.-8.6602540378443*…	2*((3.4641016151377E-10-3…	(59150…	#UN

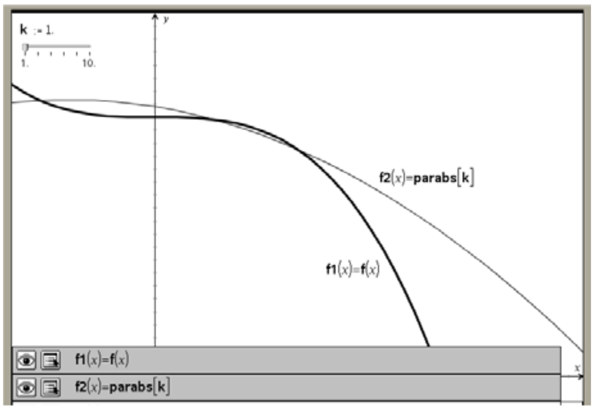
-5.-8.6602540378443*i-(-1.5773502691896E-12-4.2264973081038E-13*i)/(piecewise(i*undef+undef+undef,undef≥undef,undef-(i*undef+undef),undef<undef))

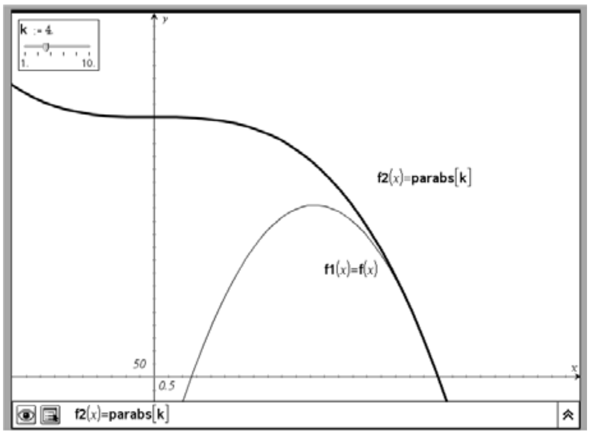
Finally I´d like to visualize the single steps of the algorithm by plotting the approximating parabolas supported by a slider in the Graphs & Geometry Application. For this purpose I calculate the parabolas passing the three points using my polreg-program which is part of my personal statistics library, called statistik (see DNL#...).

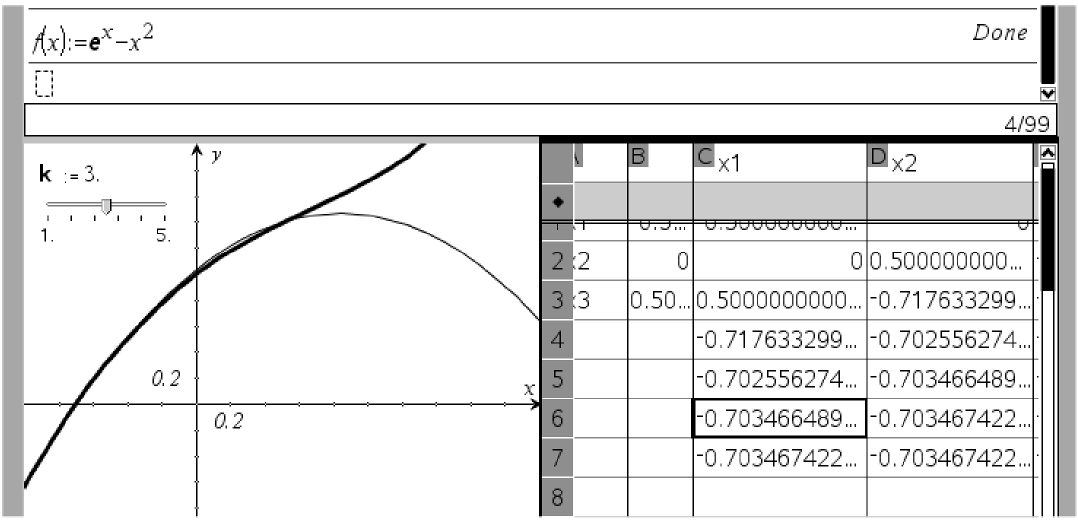
		D x2	E x3	F ca	G cb	H c	I c2	J dis	K ma	L parabs	M
◆											
1	00...	2.000000...	5.00000...	0	0	0	0	0	0	-3.*x^2−18.000000000005*x...	
2	00...	5.000000...	15.8591...	-39....	-12....	-3.0...	-48....	113.1...	-16...	-3.*x^2−18.000000000005*x+1040.	
3	00...	15.85912...	9.26622...	-35...	-39....	-22....	-60...	302.6...	-90...	-30.12534901459*x^2+272.5...	
4	12...	9.266222...	9.93445...	-48...	-35...	-30....	-28...	325.9...	-61...	-35.059801219*x^2+396.560...	
5	22...	9.934452...	9.99907...	-27...	-48...	-35....	-30...	304.5...	-60...	-29.19971*x^2+284.04306*x...	
6	52...	9.999074...	10.0000...	-29...	-27...	-29....	-29...	299.9...	-59...	#ERR	

L1 =statistik\polreg(2,{c1,d1,e1},{f(c1),f(d1),f(e1)})

I introduce a slider and visualize the approximating parabolas step by step. This works pretty well until the accuracy of the system has reached its limit.







This is a 2[nd] example (the equation mentioned by GP Speck on page 40).

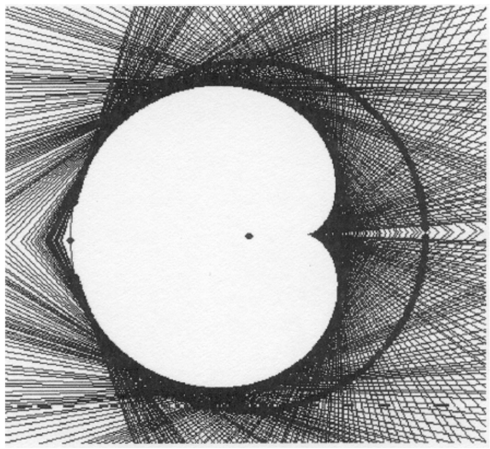
Ebene Algebraische und Transzendente Kurven (7)

Thomas Weth, Würzburg, Germany

Die Kardiode – *The Cardiod*

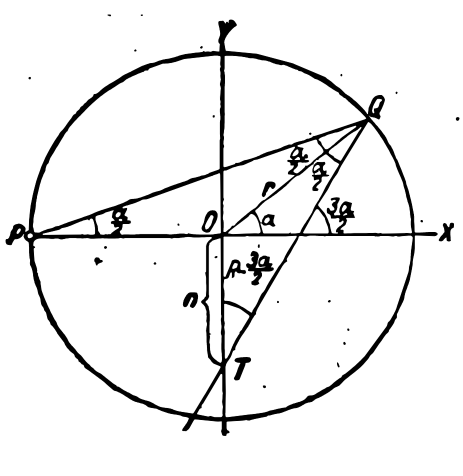
Legt man einen zylindrischen Ring auf eine ebene Tischfläche, so werden einfallende Sonnenstrahlen am Inneren des Rings reflektiert. Die reflektierten Strahlen hüllen näherungsweise eine „Brennkurve" ein, die wegen ihres Aussehens unter dem Namen Kardioide oder Herzkurve (vom griechischen καρδια) bekannt ist. Dieselbe Kurve lässt sich auch als helle Linie in einer mit Tee gefüllten Tasse beobachten. Genau genommen handelt es sich bei diesen Kurven um (halbe) Nephroiden.

Ozanam erwähnt die Kardioide in seinem „mathematischen Wörterbuch" *(Dictionnaire mathematique,* Amsterdam, 1691); die heute noch gültige Namensgebung geht allerdings auf Castillon zurück *(De curva cardioide,* 1741). Die Kardioide ist mit der „fraktalen Geometrie" zu neuer Aktualität und Popularität gelangt - sie bildet (in grober Näherung) die Umrandung der "Ikone" der Chaostheoretiker, nämlich der Mandelbrotmenge (volkstümlich „Apfelmännchen").



If you put a cylindrical ring on a plane table then the falling in sun rays will be reflected and they form approximately a "focal curve" which is called cardiod or heart curve. One can observe this curve in a cup filled with tee or coffee. Actually the cardiod consists of two (half) nephriods. In a rough approximation this curve forms the border of the chaos theorists´ "icon" - the Mandelbrot set.

Konstruktion der Kurve und Herleitung ihrer Gleichung
Construction of the curve and derivation of its equation



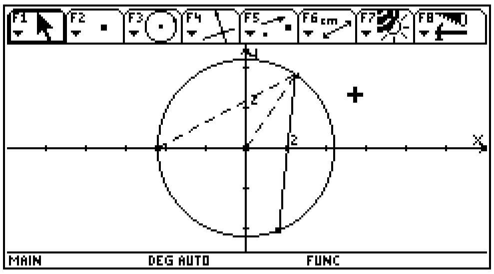
Eine genaue Konstruktion ergibt sich mit folgender Vorschrift:

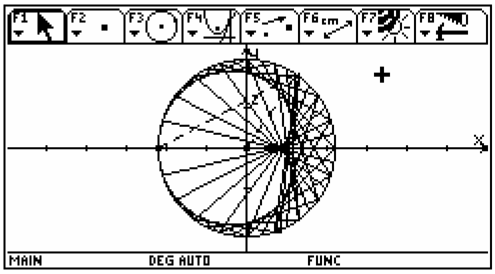
Konstruktion der Kardioide als ***katakaustische*** *Linie – als Einhüllende einer Geradenschar*:
Gehen von einem Punkt auf dem Umfang eines Kreises Strahlen aus, die an der Kreislinie reflektiert werden, so umhüllen die reflektierten Strahlen die Kardioide.

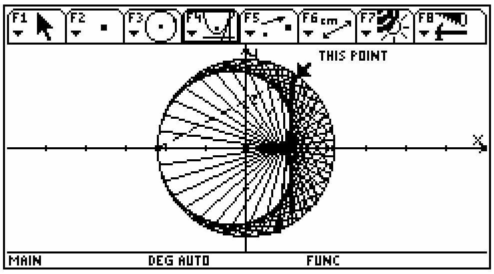
This is the instruction to find the cardiod as a ***catacaustic*** *line:*

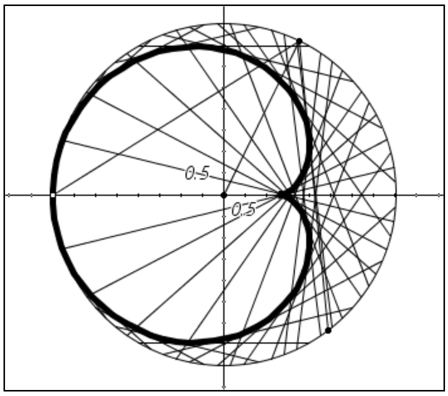
A point on the perimeter of a circle is the initial point of rays which are reflected at the perimeter. The family of the reflected rays form the cardiod as their envelope (see the sketch).

Now in 2009 we can produce the locus of the reflected segments on the graph screen of the TIs working with the Cabri-application. (TI-Nspire offers the same occasion.)









The right graph shows the cardiod on the TI-NSpire´s Graphs & Geometry Screen.

We proceed with Thomas Weth´s text from 1995:

Zunächst soll nun eine **kartesische Koordinatendarstellung** der Kurvenpunkte hergeleitet werden. Dabei ist zu beachten, dass die Kurvenpunkte genau die Berührpunkte mit den Kurventangenten sind. P sei der Punkt, von dem die Strahlen ausgehen. In obiger Figur wird der Strahl in Q reflektiert. Dann gilt für die Steigung des Strahls QT: $m = \tan\frac{3\alpha}{2}$.

Außerdem gilt nach dem Sinussatz im Dreieck OQT und unter Berücksichtigung des Vorzeichens für den Achsenabschnitt n der Geraden QT: $\frac{n}{r} = \frac{\sin\frac{\alpha}{2}}{\sin\left(\frac{\pi}{2} - \frac{3\alpha}{2}\right)}$, also $n = -\frac{r\sin\frac{\alpha}{2}}{\cos\frac{3\alpha}{2}}$.

Für die Kurventangente QT lautet also die Geradengleichung: $y = x\tan\frac{3\alpha}{2} - \frac{r\sin\frac{\alpha}{2}}{\cos\frac{3\alpha}{2}}$. Die partielle Ableitung nach dem Parameter führt zur Gleichung der Hüllkurve. (Siehe Erklärung im Anhang.)

The family of rays with their initial point on a fixed point on a circle form if they reflected at this circle the cardiod. Pay attention to the fact that the points of the curve are exactly the osculation points of the tangents. Let P the fixed point on the circle (the origin of the rays), Q is the intersection point of the ray and the circle. Then QT is the reflected ray with slope m. In triangle OQT we apply the sine rule to obtain the y-intercept n of the reflected ray. Using DERIVE we find the equation of the family of lines (I) with angle α as parameter. Partial differentiation wrt to the parameter (II) leads to the equation of the envelope. (See appendix and the accompanying file).

Formt man nun mit DERIVE um, so liefert die Multiplikation mit $\cos\frac{3\alpha}{2}$:

(I) $$F(x,y,\alpha) = y\cos\frac{3\alpha}{2} - x\sin\frac{3\alpha}{2} + r\sin\frac{\alpha}{2} = 0.$$

(II) $$\frac{\partial f}{\partial\alpha} = -3y\sin\frac{3\alpha}{2} - 3x\sin\frac{3\alpha}{2} + r\cos\frac{\alpha}{2} = 0.$$

Löst man diese Gleichungen nach x und y auf, so erhält man (nach mehreren Versuchen für die Einstellungen des *Simplification Mode* für *Trigonometry* und *TrigPowers* die Koordinaten der Kurvenpunkte als:

$$x = \frac{r}{3}(2\cos\alpha - \cos 2\alpha) \;\; und \;\; y = \frac{r}{3}(2\sin\alpha - \sin 2\alpha).$$

Damit liefert DERIVE aus der Parameterdarstellung die gesuchte Kurve.

#1: $x^2 + y^2 = r^2$

#2: P := [r·COS(t), r·SIN(t)]

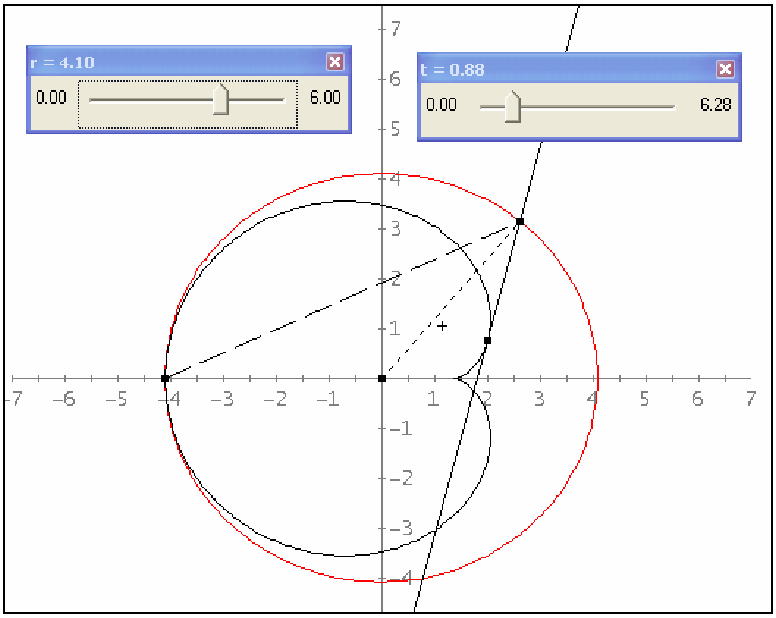
#3: $y = x\cdot \mathrm{TAN}\left(\frac{3\cdot t}{2}\right) - \frac{r\cdot \mathrm{SIN}\left(\frac{t}{2}\right)}{\mathrm{COS}\left(\frac{3\cdot t}{2}\right)}$

#4: $\left[\frac{r}{3}\cdot(2\cdot \mathrm{COS}(t) - \mathrm{COS}(2\cdot t)),\ \frac{r}{3}\cdot(2\cdot \mathrm{SIN}(t) - \mathrm{SIN}(2\cdot t))\right]$

#5: $\left[\frac{r}{3}\cdot(2\cdot \mathrm{COS}(t\_) - \mathrm{COS}(2\cdot t\_)),\ \frac{r}{3}\cdot(2\cdot \mathrm{SIN}(t\_) - \mathrm{SIN}(2\cdot t\_))\right]$

#6: [[-r, 0], P]

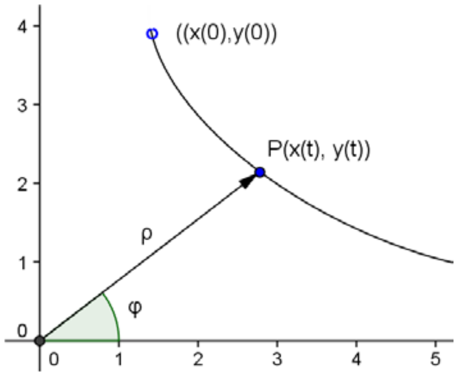
#7: [[0, 0], P]



With DERIVE 6 we can improve the representation by introducing slider bars for the radius of the circle and for the angle which is formed by the ray PQ and the x-axis.

Zur Herleitung der **Polardarstellung** der Kurve bedarf es zunächst einiger Theorie:

> Gegeben sei eine Kurve $(x(t), y(t))$ mit dem Kurvenparameter t. Gesucht ist zu jedem Kurvenpunkt P der Abstand ρ zum Ursprung des Koordinatensystems und der Winkel φ, den die Halbgerade OP mit der x-Achse einschließt (vgl. Skizze).



Zu jedem φ benötigt man also zunächst den zugehörigen Parameterwert t. Erhält man zwischen t und φ eine umkehrbar eindeutige Beziehung (mit gewissen Differenzierbarkeitseigenschaften), kann man t als Funktion von φ darstellen („$t = t(\varphi)$“) und kann die Kurve umparametrisieren. Man erhält dann $(x(t(\varphi)), y(t(\varphi)))$ oder kürzer $(x(\varphi), y(\varphi))$.

Prinzipiell kann man folgendermaßen vorgehen:
Für den Winkel φ gilt:

$$\varphi = \arctan\frac{y(t)}{x(t)} \quad \text{oder} \quad \tan\varphi = \frac{y(t)}{x(t)}.$$

Diese Gleichung löst man – falls es gelingt – nach t auf und erhält t in Abhängigkeit von φ, also eine Funktion $\varphi \to t(\varphi)$. Einsetzen liefert dann $(x(t(\varphi)), y(t(\varphi)))$. Bildet man noch den Betrag $\rho = |(x(t(\varphi)), y(t(\varphi)))|$, so kann man mit *DERIVE* die Kurve in Polarform $(\rho(\varphi),\varphi)$ zeichnen.

Wendet man dieses Verfahren nun auf die oben hergeleitete Parameterdarstellung der Kardiode $\left(\frac{r}{3}(2\cos\alpha - \cos 2\alpha), \frac{r}{3}(2\sin\alpha - \sin 2\alpha)\right)$ an, erhält man nach mehreren Umformungen eine Gleichung 4. Grades mit der „Variablen“ cos (α), die *DERIVE* nicht lösen kann.

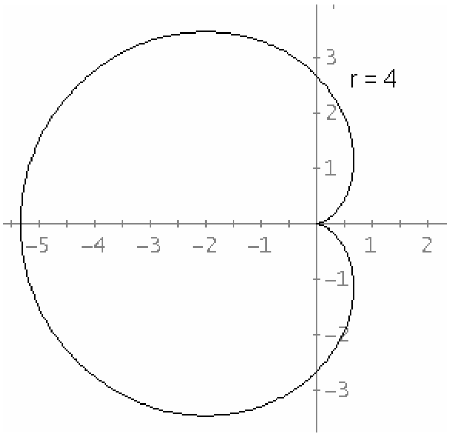
Verschiebt man allerdings die gesamte Kurve so, dass ihre Spitze in den Koordinatenursprung fällt, $\left(\frac{r}{3}(2\cos\alpha - \cos 2\alpha) - \frac{r}{3}, \frac{r}{3}(2\sin\alpha - \sin 2\alpha)\right)$ und schlägt den oben angedeuteten Weg ein, erhält man: $\tan\varphi = \frac{\sin\varphi}{\cos\varphi} = \frac{\sin\alpha}{\cos\alpha}$. Um den Radius für die Polarform zu erhalten, kann man also bei dieser speziellen Lage der Kurve direkt den Betrag der ursprünglichen Parameterdarstellung verwenden und erhält (wieder nach einigen Umformungen mit *DERIVE*) die einfache Polardarstellung der Kardiode:

$$\textbf{(III)} \qquad \rho = \left|\frac{r}{3}(2\cos\alpha - \cos 2\alpha) - \frac{r}{3}, \frac{r}{3}(2\sin\alpha - \sin 2\alpha)\right| = \ldots = \frac{2r}{3}(1 - \cos\alpha).$$

*Next we will find the **polar form** of this curve. We need for each point P(t) $\in$ curve its distance ρ from the origin and the angle φ (see the sketch above). If there exists a reversible unique relation between t and φ then t can be presented as a function of φ: t = t(φ) which leads to (x(t(φ)),y(t(φ))) or (x(φ), y(φ)) with* $\varphi = \arctan\frac{y(t)}{x(t)}$ *or* $\tan\varphi = \frac{y(t)}{x(t)}$.

If it is possible then we solve this equation for t and substitute t(φ) for t. With $\rho = |(x(t(\varphi)), y(t(\varphi)))|$ we are able to plot this curve with DERIVE in polar form.

Performing this procedure with the given parameter form we obtain a quartic with the "variable" cos α which even DERIVE is unable to solve. But after performing a translation of the curve such that its vertex is laying in the origin the next attempt turns out to be successful and we can plot the cardiod in polar form (see below).



Der zugehörige *DERIVE*-Polar-Plot is in nebenstehender Abbildung dargestellt.

Im Nachhinein ist man immer schlauer:

[Trigonometry := Expand, Trigpower := Auto]

$$\left[\frac{2\cdot r\cdot \mathrm{COS}(\alpha)}{3} - \frac{2\cdot r\cdot \mathrm{COS}(\alpha)^2}{3}, \frac{2\cdot r\cdot \mathrm{SIN}(\alpha)}{3} - \frac{2\cdot r\cdot \mathrm{SIN}(\alpha)\cdot \mathrm{COS}(\alpha)}{3}\right]$$

$$\frac{2\cdot r\cdot (\mathrm{COS}(\alpha) - 1)\cdot [- \mathrm{COS}(\alpha), - \mathrm{SIN}(\alpha)]}{3}$$

Wenn man nun beachtet, dass allgemein für die kartesische und die Polardarstellung immer gilt: $(x(\varphi), y(\varphi)) = (\rho \cos \varphi, \rho \sin \varphi)$, erhält man durch Vergleich $\rho = \frac{2r}{3}(1-\cos\alpha)$.

Afterwards it looks so easy to obtain the parameter form. We should know the general relation between the rectangular and the polar form: $(x(\varphi), y(\varphi)) = (\rho \cos \varphi, \rho \sin \varphi)$ *and then find by comparison* $\rho = \frac{2r}{3}(1-\cos\alpha)$.

Da die Kurve sicher eine algebraische ist – sie wurde elementar mit Zirkel und Lineal punktweise erzeugt – ermitteln wir ihre **algebraische Gleichung**.

Der Einfachheit halber verzichten wir auf den Streckfaktor $\frac{1}{3}$ in der Parameterdarstellung und verwenden $(x, y) = (2r \cos\alpha\,(1-\cos\alpha), 2r \sin\alpha\,(1-\cos\alpha))$.

Nun ist ein Polynom $F(x, y, r)$ mit $F(x, y, r) = 0$ zu erstellen. Wie man sofort sieht, gilt:

$$x^2 + y^2 = (2\cdot r\cdot \mathrm{COS}(\alpha)\cdot (1 - \mathrm{COS}(\alpha)))^2 + (2\cdot r\cdot \mathrm{SIN}(\alpha)\cdot (1 - \mathrm{COS}(\alpha)))^2$$

$$x^2 + y^2 = 4\cdot r^2\cdot (\mathrm{COS}(\alpha) - 1)^2$$

Zur Elimination von α aus dieser Gleichung löst man noch die in cos α quadratische Gleichung $x = r \cos \alpha\,(1 - \cos \alpha)$ nach cos α auf und setzt ein. *DERIVE* liefert bei diesem Vorgehen die algebraische Kurvengleichung der Kardiode:

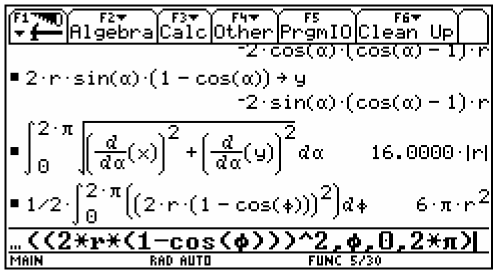
$$(x^2 + y^2)\cdot (x^2 + 4\cdot r\cdot x + y^2) = 4\cdot r^2\cdot y^2$$

Die entsprechende Polarform lautet:

$$\rho = 2\cdot r\cdot (1 - \mathrm{COS}(\phi))$$

We found all representation forms for the Cardioid. You may follow the calculation in the respective DERIVE-file.

Calculating the length of the curve and the enclosed area gives interesting results. An interpretation of these results can be found investigating the cardiod as a special epi-cycloid.



Merkwürdige Ergebnisse ergeben sich beim Berechnen der Länge der Kurve und des von ihr eingeschlossenen Flächeninhalts. Mit dem CAS berechnet man:

$l = \int_0^{2\pi} \sqrt{\dot{x}(\alpha)^2 + \dot{y}(\alpha)^2}\,\mathrm{d}\alpha = 16r.$ Damit ist die Länge der Kurve ein ganzzahliges Vielfaches des zu ihrer Erzeugung verwendeten Kreises.

$A = \frac{1}{2}\int_0^{2\pi} \rho(\alpha)^2\,\mathrm{d}\alpha = 6r^2\pi,$ also wieder ein ganzzahliges Vielfaches der Fläche des Kreises. Eine vollständige Interpretation dieser Ergebnisse erhalten wir, wenn wir die Kardiode als eine spezielle Epizykloide, eine Kreisrollkurve betrachten.

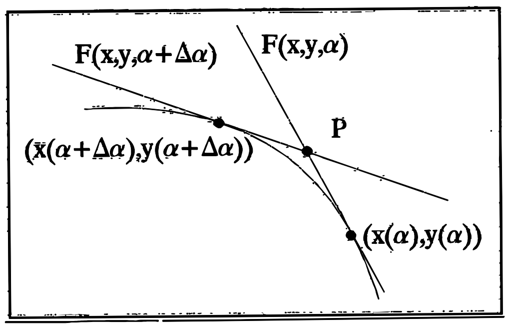
Anhang – *Appendix*

Definition von Hüllkurven – *Definition of envelopes*

Sei $F(x,y,\alpha)$ eine Funktion, so dass $F(x,y,\alpha) = 0$ für jedes α eine Kurve beschreibt. $F(x,y,\alpha)$ bestimmt dann eine Kurvenschar mit dem Scharparameter α. (Im obigen Fall sind die $F(x,y,\alpha)$ die algebraischen Gleichungen von Geraden). Die Koordinaten der Schnittpunkte P zweier „benachbarter" Kurven erfüllen dann die Gleichung

$F(x,y,\alpha+\Delta\alpha) - F(x,y,\alpha) = 0$ und speziell auch

$$\frac{F(x,y,\alpha+\Delta\alpha) - F(x,y,\alpha)}{\Delta\alpha} = 0.$$



Für $\Delta\alpha \to 0$ geht die Gleichung über in: $\frac{\partial F(x,y,\alpha)}{\partial\alpha} = 0.$ Erhält man nun durch Elimination von α aus den Gleichungen $F(x,y,\alpha) = 0$ und $\frac{\partial F(x,y,\alpha)}{\partial a} = 0$ wieder die Gleichung $f(x,y) = 0$ einer Kurve, so nennt man diese die Hüllkurve der Kurvenschar $F(x,y,\alpha)$.

If F(x,y,α) is a family of curves with parameter α then we can find the envelope of these curves by eliminating the parameter from the two equations

$F(x,y,\alpha) = 0$ *and* $\frac{\partial F(x,y,\alpha)}{\partial\alpha} = 0.$ *If this procedure gives an equation f(x,y) of a curve then this curve is the envelope.*

I had the intention to place here a short Utility provided by Sergey Biryukov. As I have the strong impression that the contents of this issue has become "heavy enough", I will do this in DNL#21. Instead of this I reprint an international product: Belgian & Austrian. When I was in Belgium Jan Vermeylen gave me a lot of work sheets and *DERIVE* files. Among them were some very interesting attempts to use the random number generator for creating exercise examples. These files were (be patient, you will find them in one of the next year ´s *DNL*):

poly_frac.mth	to practise factoring polynomials
rekrij.mth	to solve problems with arithmetic series
quadr_eq.mth	to create various quadratics

At the *DERIVE* conference in Honolulu our working group had the idea to produce a "How it could look" work sheet for quadratics and factoring polynomials. We decided that I should add a demo file in combination with some exercises. I remembered Jan´s quadr_eq.mth and this is the result:

First load `VIETAUTI.MTH` as Utility file:

```
#1:  [z1 :≡ RANDOM(21) - 10, z2 :≡ RANDOM(31) - 15, z3 :≡ RANDOM(11) - 5]

#2:  list :≡ [(a - z1)·(a - z2), (b - z1)·(b - z2), (d - z1)·(d - z2), (e - z1)·(e -
       z2), (f - z1)·(f - z2), (g - z1)·(g - z2), (h - z1)·(h - z2), (i - z1)·(i - z2),
       (k - z1)·(k - z2), (o - z1)·(o - z2), (p - z1)·(p - z2), (t - z1)·(t - z2), (x -
       z1)·(x - z2), (y - z1)·(y - z2), (z - z1)·(z - z2)]

     VIE(n, va) :≡ [VECTOR(EXPAND( lim  (x_ - z1)·(x_ - z2)) = 0, n_, 1, n)]'
#3:                                 x_→va

     VIETA(n) :≡ EXPAND([VECTOR(list                  = 0, j, 1, n)])'
#4:                                  RANDOM(15) + 1

     SOLU(v) :≡ VECTOR(IF(DIM(SOLUTIONS(v_, (VARIABLES(v_)) )) = 2, SOLUTIONS(v_,
#5:                                                         1
       (VARIABLES(v_)) ), APPEND(SOLUTIONS(v_, (VARIABLES(v_)) ), SOLUTIONS(v_,
                      1                                       1
       (VARIABLES(v_)) ))), v_, v)
                      1
```

(Some details of the 1995 version for *DERIVE* 3 have been changed to make the file suitable for *DERIVE* 6.10.)

Then produce the following VIETADEM.DMO demo file. You can do it with any text editor or using the Edit > Annotation facility of the recent *DERIVE* versions. (More details can be found in *DNL*#4.)

The comments are the lines with the leading semi-colon. The must be entered as Annotation (without the semi-colon) together with the respective *DERIVE* commands:

```
;This initializes the random number generator. Press ENTER.
RANDOM(0)

;Try to solve the equations. Continue with ENTER.
task:=VIE(3,x)

;Congratulations if you are right. Press ENTER for the next bundle.
SOLU(task)
```

```
;Check your solutions pressing ENTER.
task:=VIE(5,a)

;Let’s finish this sequence with a package of 10.
SOLU(task)

;ENTER will present all solutions.
task:=VIETA(10)

;If more than two answers are wrong, go on practising. You can then run the
DEMO-file again.
SOLU(task)

;Interrupt the demo and check the solution using the SOLVE-command.
VIETA(3)

VIETA(5)

;After checking the results, you may run the DEMO again.
VIETA(10)
```

If you are doing in with *DERIVE* 6 you might face problems saving the file as a “demo” file (with extensions .dmo only). I´d like to remind you that you have to save it as “vietadmo.dmo” – including the quotes!!

$$\text{VIETA}(5) = \begin{bmatrix} x^2 - 17\cdot x = -72 \\ g^2 + g = 42 \\ x^2 + 7\cdot x = 8 \\ z = -7 \lor z = 7 \\ t^2 - 11\cdot t = -28 \end{bmatrix}$$

Is it a feature or is it a bug?

The function works – but if by chance – we have two solutions with $z_1 = -z_2$ then *DERIVE* automatically solves the equation instead of only expanding the product of the two respective polynomials.

DERIVE 5 & 6 allow programming. So we can collect the single procedures in a compact program which excludes the special cases $z_1 = -z_2$ and z_1 or $z_2 = 0$:

```
#7:   list_(z1, z2) := [(a - z1)·(a - z2), (b - z1)·(b - z2), (d - z1)·(d - z2), (e -
        z1)·(e - z2), (f - z1)·(f - z2), (g - z1)·(g - z2), (h - z1)·(h - z2), (i -
        z1)·(i - z2), (k - z1)·(k - z2), (o - z1)·(o - z2), (p - z1)·(p - z2), (t -
        z1)·(t - z2), (x - z1)·(x - z2), (y - z1)·(y - z2), (z - z1)·(z - z2)]

      VIETA_(n, i_, tbl, p) :=
        Prog
          i_ := 1
          tbl := []
          Loop
#8:         If i_ > n
               RETURN tbl
            p := [z1, z2]
            If p↓1 + p↓2 ≠ 0 ∧ p↓1·p↓2 ≠ 0
               tbl := APPEND(tbl, [[EXPAND((list_(p↓1, p↓2))↓(RANDOM(15) + 1)) = 0]])
            i_ :+ 1
```

Titbits from Algebra and Number Theory (6)

by Johann Wiesenbauer, Vienna

This time I would like to deal with a purely algebraic problem to make up for the fact that we have concerned mostly with number theoretic issues so far. And what could be more algebraic – at least from the historical point of view – than the solution of polynomial equations! As you all know DERIVE can easily cope with polynomial equations in one variable up to degree 4. Therefore to be a real challenge for DERIVE the degree of the polynomial in question has to be greater than 4. What about the polynomial equation

$$z^{17} - 1 = 0$$

which was solved by Gauß when he was only 19 years old? (Actually Gauß started his famous diary on March 30, 1796, with the following entry: "*Principia quibus innititur section circuli, ac divisibilitas eiusdem geometrica in septemdecim partes etc.*" cf. [1]). Of course the application of the built-in SOLVE to this equation is definitely out of question. But what we could do is to follow the thoughts of genius Gauß and enjoy their originality and elegance leaving all the drudgery to DERIVE. Are you ready for it? Then here we go.

To begin with, it suffices to determine the solution

$$z = e^{i\varphi} = \cos\varphi + i\sin\varphi \quad \text{with} \quad \varphi : \frac{2\pi}{17}$$

of the equation above since all other solutions are merely powers of it. In the outline of the solution given by Gauß in his letter to Gerling he actually determined cos φ which amounts to the same thing. To do so he first introduces the notations

$$a := \cos\varphi + \cos 4\varphi$$
$$b := \cos 2\varphi + \cos 8\varphi$$
$$c := \cos 3\varphi + \cos 5\varphi$$
$$d := \cos 6\varphi + \cos 7\varphi$$

as well as $e := a + b$ and $f := c + d$.

Then he states that "according to a wellkown theorem" the equality

$$e + f = -\frac{1}{2}$$

holds. Using

$$\cos k\varphi = \frac{1}{2}\left(z^k + z^{-k}\right) = \frac{1}{2}\left(z^k + z^{17-k}\right)$$

this can be derived (without DERIVE!) in the following way:

$$e + f = \sum_{k=1}^{8} \cos k\varphi = \frac{1}{2}\sum_{k=1}^{16} z^k = \frac{1}{2}\frac{z^{17}-1}{z-1} = -\frac{1}{2}.$$

Gauß continues by claiming that

$$e\,f = -1.$$

Again we could calculate

$$e\,f = (\cos\varphi + \cos 2\varphi + \cos 4\varphi + \cos 8\varphi)(\cos 3\varphi + \cos 5\varphi + \cos 6\varphi + \cos 7\varphi) =$$
$$= \frac{1}{2}\left(z + z^{16} + z^2 + z^{15} + z^4 + z^{13} + z^8 + z^9\right)\frac{1}{2}\left(z^3 + z^{14} + z^5 + z^{12} + z^6 + z^{11} + z^7 + z^{10}\right)$$

and using $z^{17} = 1$ as well as

$$\sum_{k=1}^{16} z^k = -1 \qquad (*)$$

after a lot of tedious computations we could finally arrive at the desired result. (Try it out by yourself!) Is there a way to make DERIVE do these calculations for us? You won´t be taken by surprise if I tell you there is actually one. The following utility function can be used quite generally to reduce a polynomial expression u in one variable modulo an equation v, which amounts to adding the rule v to the other rules for the handling of polynomials.

```
RED(u, v) := ITERATE(RHS(v)·QUOTIENT(u_, LHS(v)) + REMAINDER(u_, LHS(v)), u_, u)
```

By setting $u := e\,f$ and taking (*) for v we get:

$$e := \frac{z + z^2 + z^4 + z^8 + z^9 + z^{13} + z^{15} + z^{16}}{2}$$

$$f := \frac{z^3 + z^5 + z^6 + z^7 + z^{10} + z^{11} + z^{12} + z^{14}}{2}$$

$$\text{RED}\left(e \cdot f,\ \sum_{k\_=1}^{16} z^{k\_} = -1\right) = -1$$

Therefore e and f are both roots of the quadratic equation

$$x^2 - (e + f)x + e\,f = x^2 + \frac{1}{2}x - 1 = 0$$

which has the solutions

$$x_{1,2} = \frac{-1 \pm \sqrt{17}}{4}.$$

Now we have come across a small problem, namely is

$$e = x_1,\ f = x_2 \qquad (**)$$

or vice versa?

Again DERIVE can be of help in showing that the first alternative (**) is valid:

$$\text{APPROX}\left(\lim_{\phi \to 2\cdot\pi/17} (\cos(\phi) + \cos(2\cdot\phi) + \cos(4\cdot\phi) + \cos(8\cdot\phi))\right) = 0.780776$$

Do you see the achievement? The polynomial equation (*) of degree 16 has been broken up into two polynomial equations (**) with fewer terms! It takes no Gauß to have the idea of trying the same trick once more:

$$\left[a := \frac{z + z^4 + z^{13} + z^{16}}{2},\ b := \frac{z^2 + z^8 + z^9 + z^{15}}{2}\right]$$

$$\text{RED}\left(a \cdot b,\ \sum_{k\_=1}^{16} z^{k\_} = -1\right) = -\frac{1}{4}$$

Therefore a and b are both roots of the quadratic equation $x^2 - (a + b)x + a\,b = x^2 - e\,x - \frac{1}{4} = 0.$

Again we leave it to DERIVE to calculate its solutions:

$$\text{SOLUTIONS}\left(x^2 - e\cdot x - \frac{1}{4} = 0,\ x\right)$$

$$\left[\frac{\sqrt{(34 - 2\cdot\sqrt{17})}}{8} + \frac{\sqrt{17}}{8} - \frac{1}{8},\ -\frac{\sqrt{(34 - 2\cdot\sqrt{17})}}{8} + \frac{\sqrt{17}}{8} - \frac{1}{8}\right]$$

By applying APPROXIMATE one finds out that the first solution is *a* and the second one is *b*:

```
[1.024740588, -0.2439641824]
```

$$\text{APPROX}\left(\lim_{\phi\to 2\cdot\pi/17}(\text{COS}(\phi) + \text{COS}(4\cdot\phi))\right) = 1.02474$$

In analogous manner the values of *c* and *d* are calculated:

$$\left[c := \frac{z^3 + z^5 + z^{14} + z^{12}}{2},\ d := \frac{z^6 + z^7 + z^{11} + z^{10}}{2}\right]$$

$$\text{RED}\left(c\cdot d,\ \sum_{k\_=1}^{16} z^{k\_} = -1\right) = -\frac{1}{4}$$

$$f := -\frac{\sqrt{17}}{4} - \frac{1}{4}$$

$$\text{SOLVE}\left(x^2 - f\cdot x - \frac{1}{4} = 0,\ x\right)$$

$$x = -\frac{\sqrt{(34 - 2\cdot\sqrt{17})}}{8} + \frac{\sqrt{17}}{8} - \frac{1}{8}\ \vee\ x = \frac{\sqrt{(34 - 2\cdot\sqrt{17})}}{8} + \frac{\sqrt{17}}{8} - \frac{1}{8}$$

$$\left[x = \sqrt{\left(\frac{\sqrt{17}}{32} + \frac{17}{32}\right)} - \frac{\sqrt{17}}{8} - \frac{1}{8},\ x = -\sqrt{\left(\frac{\sqrt{17}}{32} + \frac{17}{32}\right)} - \frac{\sqrt{17}}{8} - \frac{1}{8}\right]$$

```
[x = 0.172075, x = -1.45285]
```

$$\text{APPROX}\left(\lim_{\phi\to 2\cdot\pi/17}(\text{COS}(3\cdot\phi) + \text{COS}(5\cdot\phi))\right) = 0.172075$$

The rest is a piece of cake. Because of

$$\cos\varphi\cos 4\varphi = \frac{1}{2}\left(z + z^{16}\right)\frac{1}{2}\left(z^4 + z^{13}\right) = \frac{1}{4}\left(z^5 + z^{14} + z^{20} + z^{29}\right) =$$
$$= \frac{1}{4}\left(z^3 + z^{14} + z^5 + z^{12}\right) = \frac{1}{2}\left(\cos 3\varphi + \cos 5\varphi\right) = \frac{c}{2}$$

we have

$$(x - \cos\varphi)(x - \cos 4\varphi) = x^2 - (\cos\varphi + \cos 4\varphi)x + \cos\varphi\cos 4\varphi =$$
$$= x^2 - a\,x + \frac{c}{2} = 0.$$

We let DERIVE put the finishing touches:

$$\left[a := \sqrt{\left(\frac{17}{32} - \frac{\sqrt{17}}{32}\right)} + \frac{\sqrt{17}}{8} - \frac{1}{8},\ c := \sqrt{\left(\frac{\sqrt{17}}{32} + \frac{17}{32}\right)} - \frac{\sqrt{17}}{8} - \frac{1}{8}\right]$$

$$\text{SOLUTIONS}\left(x^2 - a\cdot x + \frac{c}{2} = 0,\ x\right)$$

$$\left[\frac{\sqrt{(-\sqrt{(38\cdot\sqrt{17} + 170)} + 3\cdot\sqrt{17} + 17)}}{8} + \frac{\sqrt{(34 - 2\cdot\sqrt{17})}}{16} + \frac{\sqrt{17}}{16} - \frac{1}{16},\ -\frac{\sqrt{(-\sqrt{(38\cdot\sqrt{17} + 170)} + 3\cdot\sqrt{17} + 17)}}{8} + \frac{\sqrt{(34 - 2\cdot\sqrt{17})}}{16} + \frac{\sqrt{17}}{16} - \frac{1}{16}\right]$$

[0.9324722294, 0.09226835946]

$$\text{APPROX}\left(\lim_{\phi\to 2\cdot\pi/17} \text{COS}(\phi)\right) = 0.932472$$

That´s it! There is only one thing that is slightly disturbing, namely that DERIVE didn´t return the term for cos φ in its most beautiful form which is according to Gauß:

$$\cos\varphi = -\frac{1}{16} + \frac{1}{16}\sqrt{17} + \frac{1}{16}\sqrt{34 - 2\sqrt{17}} + \frac{1}{8}\sqrt{17 + 3\sqrt{17} - \sqrt{34 - 2\sqrt{17}} - 2\sqrt{34 + 2\sqrt{17}}}\,.$$

But isn´t this a bit of a tall order in view of this huge formula?

Far more important is the fact that both forms of cos φ show that apart from the basic operations only square roots are used for its calculation. From this follows easily the stunning fact that the regular polygon with 17 edges can be constructed by using compass and straight-edge only! (cf [2] for the details!) As Gauß was able to show this in his "*Disqusitiones Arithmeticae"* (1801) this is possible for a regular polygon with n edges if and only if n is of the form

$$n = 2^r f_1 f_2 \ldots f_s,\ r \ge 0, s \ge 0,$$

where $f_1, f_2, \ldots f_s$ are pairwise distinct Fermat primes! Thus we've returned happily to number theory where we will continue next time!

References

[1] Wußing H. Mathematisches Tagebuch 1796-1814 von Carl Friedrich Gauß, Akedemische Verlagsgesellschaft, Leipzig 1981

[2] Strubecker K., Carl Friedrich Gauß – Princeps Mathematicorum, Bild der Wissenschaft 5, 1977 (118 – 126)

Three additional comments from the editor:

- The file TITIBITS6.MTH is among the accompanying files and it contains the *DERIVE* functions and the complete calculation.

- Some time ago a DUG member sent a *DERIVE* file calculating the 17 roots of the equation. He refers to an article written by Friedrich Freytag, DdM 3, 1992 (pp 188 – 213). I recommend reading this article. There one can find a description how to construct the side of a regular 17-edge. I intend incuding this construction in DNL#21.
- Johann Wiesenbauer has invented a remarkable function `RED(u,v)`. We compute a polynomial expression under consideration of an additional condition (which is in my opinion a kind of special substitution). Johannes told me that he had thought that this would be an advantage of other CAS packages, and he wanted to make this possible with *DERIVE*, too.

Three easy to follow examples for applying `RED(u,v)`:

$$[x := 2\cdot a + c,\ y := c + 3\cdot b]$$

$$\text{RED}(x\cdot y^2 + 3\cdot x\cdot y,\ a + b = 10)$$

$$-18\cdot b^3 + 3\cdot b^2\cdot(54 - c) + b\cdot(4\cdot c^2 + 123\cdot c + 180) + c\cdot(c + 3)\cdot(c + 20)$$

$$\left[x := 3\cdot a^2 + c,\ y := c + 3\cdot b^2\right]$$

$$\text{RED}\left(3\cdot(x + y),\ a - b = \frac{1}{2}\right)$$

$$\text{RED}\left(3\cdot(x + y),\ a - b = \frac{1}{2}\right) = 18\cdot b^2 + 9\cdot b + \frac{3\cdot(8\cdot c + 3)}{4}$$

$$\text{RED}(x + y,\ a - 2\cdot b = 5) = 15\cdot b^2 + 60\cdot b + 2\cdot c + 75$$

We would like to rewrite $x + y = 3a^2 + 3b^2 + 2c$ under consideration that $a - 2b = 5$. Let´s reproduce Johann Wiesenbauer´s RED-function step by step (= iteration by iteration):

1st step: Long division and its consequence

$$\left.\begin{array}{l}(3a^2+3b^2+2c):\underbrace{(a-2b)}_{5}=3a+6b\\ \text{remainder: } 15b^2+2c\end{array}\right\}3a^2+3b^2+2c=5(3a+6b)+15b^2+2c=15a+15b^2+30b+2c$$

2nd step: We can perform the long division of the polynomials again:

$$\left.\begin{array}{l}(15a+15b^2+30b+2c):\underbrace{(a-2b)}_{5}=15\\ \text{Rem}:15b^2+60b+2c\end{array}\right\}15a+15b^2+30b+2c=3a^2+3b^2+2c=15\cdot5+15b^2+60b+2c$$

No more division is possible, the next quotient would be zero and the result of the procedure remains the same $75 + 15b^2 + 60b + 2c$. The iteration process has come to an end. Compare with the result of the RED-function from above! Things will change if we change the variable order:

```
VariableOrder := [b, a]
```

$$\text{RED}(x + y,\ a - 2\cdot b = 5) = \frac{15\cdot a^2}{4} - \frac{15\cdot a}{2} + \frac{8\cdot c + 75}{4}$$

(Who can write a respective function/program for the TI´s?)