THE DERIVE - NEWSLETTER #25

ISSN 1990-7079

THE BULLETIN OF THE

DERIVE

USER GROUP

+ TI 92

Contents:

revised 2011 March 1997

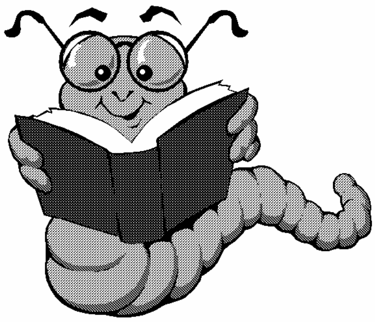
Learning Numerical Analysis through DERIVE T. Etchells, J. Berry

This book covers the major numerical methods, and their analysis, for first courses at college and undergraduate level. The relative merits of each method are covered both analytically, providing a thorough grounding in the algebraic approach, and practically, through the tried and tested computer lab-based activities.

DERIVE provides a platform on which to quickly and accurately perform many complicated numerical calculations. Also, DERIVE's ability to algebraically manipulate expressions and perform calculus operations, enhances the investigation of the convergence of numerical methods. Each chapter includes the development and algebraic analysis of the methods, lab-based activities, ideas for coursework, case studies, exercises and solutions. Free supporting utility files are downloadable via Chartwell-Bratt's web server.

Chapter 1 introduces the basic tool of numerical methods, which is recurrence relations, their solution and ill-conditioning problems. In chapter 2 we use recurrence relations methods that are used in solving equations. Chapter 3 deals with the approximation of functions by polynomials, and in particular the Taylor Polynomial, which is then used extensively in chapter 4 to analyse the errors associated with numerical methods. Chapters 5 and 6 deal with numerical approaches to the calculus of differentiation and integration. In chapter 7 we introduce and analyse numerical methods of solving differential equations.

ISBN 0-86238-468-0, 239 pages, Chartwell-Bratt, February, 1997.



Mathematical Activities with Computer Algebra, a photocopiable resource book from Etchells T, Hunter M, Monaghan J, Pozzi S and Rothery A

This photocopiable resource book is the first of a new generation of support materials for the educational use of computer algebra. Designed to be used with any computer algebra system, the authors go beyond mere button pressing and show how to harness the power of computer algebra systems for educational purposes.

Concepts are illustrated, techniques and methods presented, and modelling and applications are explained. Appendices give overviews of DERIVE, Maple, Mathematica, Theorist (MathPlus) and the new TI-92 calculator.

Activity Worksheets, Help Sheets and Teaching Notes cover a wide range of mathematical topics at school and college level.

Topics covered include; functions and graphs, differentiation, integration, sequences and series, vectors and matrices, mechanics, trigonometry, numerical methods. Activities include: Multiplying factors; Equation of a tangent; Taxing functions; The tile factory; Function and derivative - visualisation; The approximate derivative function; Sketching graphs; Pollution and population; Max cone; Optimising transport costs; Area under a curve; Enclosed areas; A function whose derivative is itself; Wine glass design; The limit of a sequence; Visualising Taylor approximations; Visualising matrix transformations; Blood groups; Circular motion; Swing safety; No turning back; Modelling the sine function; Solving equations with tangents.

20 Pounds, 96 pp, 1996, ISBN 0-86238-405-2

Both books are available in 2011 (e.g. AMAZON), Josef

In the last DNL I gave a wrong ISBN for 'An Introduction to the Mathematics of Biology'. Here is the correct one: ISBN 3-7643-3809-1.

Interesting WEB sites

http://www.swp.co.at/swhe/swhe.html (Soft Warehouse Europe)

http://derive.com (Soft Warehouse Hawaii)

http://ti.com.calc (for TI-92 programs); http://www.studli.se/chartwell.html (Publisher)

http://www.kolleg.nuernberg.de/ti92.htm (W.Pröpper´s TI-programs: ableit(), galton(), binvert())

http://www.tech.plym.ac.uk/maths/CTMHOME/CTM.HTML (John Berry in Plymouth)

http://www.cms.livjm.ac.uk/www/homepage/cmstetch/index.htm (Terence Etchells, DERIVE and TI)

All these sites are now (2011) only of historical interest, Josef

Do you know other sites? Share your knowledge with us! We will visit some DUG members´ home pages in the future. Jan Vermeylen sent a valuable collection of addresses. Wait for the next issue!! Thanks to Jan from Kapellen in Belgium. Here is one of his goodies:

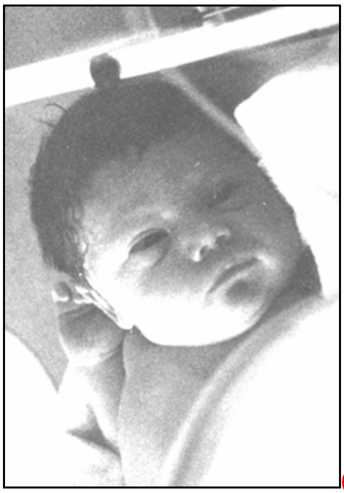
http:\\archives.math.utk.edu (Mathematical Archives with among others a lot of historical information)

Liebe DERIVE- und TI-Freunde,

Sie werden sicher über das Bild erstaunt sein. Ich bin in keinen Jungbrunnen gefallen. Das ist eines der jüngsten Mitglieder der DERIVE-Gemeinde: Kimberly, unsere erste Enkelin im zarten Alter von einer Stunde. Damit sind nun viele Fragen beantwortet (Hallo, Bärbel & Co).

Hier ist nun unsere 25. Ausgabe und ich habe aus Anlaß dieses kleinen Jubiläums SWHH und SWHE um einen Beitrag für den DNL#25 gebeten. Es sind zwei Grußworte geworden (Seite 37), für die ich mich herzlich bedanke. Ich drucke sie gerne ab, möchte aber die viele Anerkennung, die darin ausgedrückt wird, gerne mit Ihnen allen teilen.

Manchmal kommen allerdings auch leise Klagen: .... die Themen - und die zugehörigen Lösungen - sind hin und wieder von einem anderen Stern, sehr weit weg von meinen täglichen "Problemchen" mit der Mathematik .....(D.Blum)


(*)

Dazu möchte ich bemerken, daß ja Gott sei Dank nicht nur Lehrer zu unseren Mitgliedern zählen und daß es auch den Lehrern gestattet sein sollte so dann und wann nach den Sternen zu greifen. Ich möchte aber allen, die sich auch betroffen fühlen, ermuntern, über Ihre "Problemchen" zu berichten. Da kann keine Zuschrift und kein Beitrag zu wenig aufregend oder zu einfach sein. Sie alle gestalten den DNL. Und glauben Sie mir bitte, auch ich habe meine Mathe-"Problemchen" mit den 15-19jährigen, und das oft nicht zu knapp.

Ich habe mich aber bemüht, gerade diesen DNL besonders für den Lehreralltag nutzbar zu machen. Im nächsten DNL werden wir uns auch der Geometrie widmen: Abbildungen von Objekten im R^3, mit verdeckten Kanten, Fraktale usw. Auch eine ganze Schachtel voll mit TI- Fragen und Antworten steht für Sie bereit.

In vielen Fällen können DERIVE-Beiträge Anregungen für den Einsatz mit dem TI geben und umgekehrt. Wir zeigen das auch in diesem Heft.

Ich hoffe, dass Sie bereits ab dem nächsten DNL DERIVE- und TI-Dateien von einer eigenen Homepage abholen können.

Bis zum nächsten Mal
Josef

Dear *DERIVE*- and TI-friends,

Certainly you will be astonished at the photograph. No, I haven't fallen into any fountain of youth. May I introduce one of the youngest members of the *DERIVE*- and TI community: Kimberly, our first granddaughter in the tender age of one hour. This picture will answer many questions (Hello, Bärbel & friends).

Here is now our 25th issue and on this occasion I've asked SWHH and SWHE for a contribution. They have both sent greeting addresses (page 37), many thanks. It is a pleasure for me to print them, but I'd like to share their appreciation with all of you.

Sometimes I receive minor complaints: ... now and then the problems - and the accompanying solutions - seem to be from another star, very far away from my daily "Problemchens" with mathematics ... (D.Blum)

Please consider that we fortunately have a big number of non-teachers in our group and that even the teachers should be allowed to reach for the stars now and then. I'd like to encourage all of you who are feeling similar to report about their "Problemchens" (= small problems). No question, no letter, no contribution can be too simple or too unexciting. You all produce the DNL. And believe me; I have the same maths "Problemchens" with the 15-19 years´ students as you do.

I have tried to make this DNL especially useful for the teachers´ workaday routine. In the next DNL you will again find geometric items: mappings of objects in R^3, hidden lines, fractals etc. There is a box full of TI-questions with answers for you waiting to be opened.

I think that in many cases *DERIVE* contributions might inspire you for TI-applications and vice versa. You will find examples in this issue.

I hope that I will be able to offer the facility to download *DERIVES* and TI-files from my school's home page in the near future.

Until the next DNL
Josef

(*) If you are interested about the change of appearance of Kim between 1997 and now then have a look on page 48. There you can find a picture of Kim together with her 3 sisters and 3 cousins who accomplished our unique "set" of seven grandchildren, Noor and Josef (proud grandparents).

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

Editor: Mag. Josef Böhm
A-3042 Würmla
D´Lust 1
Austria
Phone: 43-(0)660 31 36 365
e-mail: nojo.boehm@pgv.at

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

Preview: Contributions for the next issues

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

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

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

Terence Etchells, Liverpool, UK t.a.etchells@livjm.ac.uk

Hi Josef,

AAGH, Did you not notice something wrong with the output of the ROMBERG_AXU_TABLE() function on page 3 of the last newsletter: It is clearly rubbish.

The problem arises with an incorrect transcription of the auxiliary functions, a – has been replaced with a * in the function ROMBERG_AUX_TABLE(). It should read

```
ROMBERG_AUX_TABLE(v,f,x,a,b,n):=ITERATES([VECTOR(2^k*v SUB (r+1)-v SUB r/
           (2^k-1),r,1,n-c),k+2,c+1],[v,k,c],[ROMBERG_START(f,x,a,b,n),4,1],n-1)
```

Also the example should have been set to 13 digits precision to show the convergence of the method in the table. (With 6 digits precision Simpson's rule gives correct result with 8 strips, for this example).

Could you please publish a correction in the DNL#25 please.

DNL: *Of course, I do. Please excuse my mistake. There were some problems in reading your email, because of a special encoding of some characters, e.g.*

"Author ROMBERG(=FBsinx,x,0,=E3,3) and approX"

#19: $\text{ROMBERG\_TABLE}\left[\frac{\text{SIN}(x)}{x}, x, 0, 4, 5\right]$

#20:

1.75804691	28.0139169	1792.45321	458861.105	469873330.6
1.758195	28.0140294	1792.45365	458861.112	""
1.75820265	28.0140362	1792.45367	""	""
1.7582031	28.0140366	""	""	""
1.75820313	""	""	""	""

Steven Schonefeld, Angola, IN, USA schonefelds@alpha.tristate.edu

Hi Josef,

I just got DNL#24 and have a couple of comments on the utility file, ROMBERG.MTH, by Terence Etchells.

(1) There was a problem with the integral of SIN(x)/x from zero to four. I think the problem involves evaluating the function at x = 0 -- it is undefined there. Perhaps you should define the function F(x):=IF(x=0,1,SIN(x)/x) and integrate this function from zero to four.

(2) The interested reader might wish to see the treatment of Romberg integration in my book

NUMERICAL ANALYSIS via DERIVE, MathWare Urbana,IL 61801 (USA) phone (800)255-2468, ISBN 0-9623629-2-1

Keep up the good work,
Steven Schonefeld.

DNL: *I'll show Terence's example treated with Steven's procedure* 4_5_ROMB.MTH:

#23: $F(x) := \frac{\text{SIN}(x)}{x}$ User

#24: ROMB(0, 4, 5) User

#25: Approx(#24)

1	4	?	"***"	"***"	"***"
2	2	?	?	"***"	"***"
4	1	?	?	?	"***"
8	0.5	?	?	?	?
16	0.25	?	?	?	?
32	0.125	?	?	?	?

#26: F(x) := IF $\left[x = 0,\ 1,\ \frac{SIN(x)}{x}\right]$ User

#27: ROMB(0, 4, 5) User

#28: Approx(#27)

1	4	1.62159	"***"	"***"	"***"
2	2	1.72009	1.75292	"***"	"***"
4	1	1.74855	1.75804	1.75838	"***"
8	0.5	1.75578	1.75819	1.75820	1.75820
16	0.25	1.75759	1.75820	1.75820	1.75820
32	0.125	1.75805	1.75820	1.75820	1.75820

It is quite interesting that Steven´s algorithm seems to need the special definition of SIN(x)/x for x = 0, while Terence´s procedure does not.

Oscar Garcia, Frederiksberg, Danmark **Oscar.Garcia@flec.kvl.dk**

3D plot animation with 3dv

If you don´t have AcroSpin (or even if you do), you can still animate *DERIVE* 3D plots using **3D Viewer**. This is a freeware program that allows rotating 3-d wire frames in real time under **mouse control**.

What you need:

1. 3DV.EXE. Available in 3DV25.ZIP from many archive sites and bulletin boards. For example in SimTel´s directory /MSDOS/GRAPHICS/ (on the Web you can look in WWW:SIMTEL.NET/MSDOS/GRAPHICS/, for example)
2. An AWK interpreter. You can get MAWK by ftp from

   ftp.cdrom.com/.l/simtelnet/gnu/gnuish/mawkl22x.zip or Duff´s awk from SimTel´s directory /msdos/awk/, etc.
3. The following awk scrip to vonvert an AcroSpin file generated by *DERIVE* to 3DV format:

```
/POINTLIST/,NF==0 {if(NF==4) {point[$1]=$2" "$3" "$4;npoints++} }
/SET COLOR/ { color=$3 }
/LINELIST/,/END/ { if(NF==2)line[++nlines]=$0" "color }
END  { print npoints
       for(i in point) {print point[i];point[i]=++p}
       print nlines*2
       for(i in line)
       { split(line[i],x)
         print point[x[1]],0
         print point[x[2]],x[3]
       }
      }
```

   Save it as ACRO.AWK in your *DERIVE* directory.
4. Save the following as ACRO.BAT in your *DERIVE* directory:

```
mawk -f acro.awk acro.acd>acro.3d
c:\3dv\3dv acro.3d
```

   This assumes that you have MAWK.EXE somewhere in your path, and that 3DV.EXE is i C:\3DV\. Otherwise alter name(s) and/or path(s) as needed.

To use from DERIVE:

1. Highlight the expression and switch to a 3D-plot window. Plot on the screen if you want.
2. Do "Transfer Acrospin Save". Use "acro" for the file name.
3. Do "Options Execute". Enter "acro" and press ENTER.
4. In 3dviewer move around the object with the mouse. To exit click a mouse button.

See section 5.2.4 in the *DERIVE* manual for more details and hints.

Problems:

– Duff´s awk can run out of memory with large files (more than a 10x10 grid, say, depending on how much low memory you have available.)
– Classic *DERIVE* does not leave enough available memory to oad mawk through "Options Execute" (use Duff´s awk or run ACRO.BAT outside *DERIVE*). *DERIVE XM* is OK.
– Other awk interpreters may or may not have enough memory to run within *DERIVE* (gawk does not). The old large memory model compilation of mawk 1.0 (bmawk) does slightly better that the new version, but does not seem to be currently available from the net.

Additional notes:

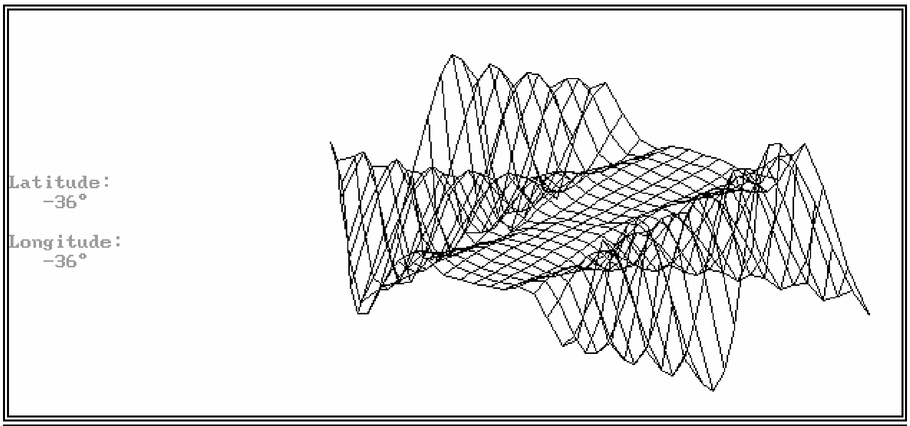
– You can save a few keystrokes by faking ACROSPIN.EXE through the following C miniprogram

```
#include<stdlib.h>
void main()
{system("acro.bat");}
```

  Compile as ACROSPIN.EXE and place in your *DERIVE* directory. Just use "acro" as the save file name.
– the 3dv data file can be used for producing high quality hardcopy on a variety of devices. Look into 3DVKIT1.ZIP, found whereever you found 3DV25.ZIP.
- The memory problems could be solved if somebody with access to an awk compiler would post a compiled version of acro.awk. Or write it in C or whatever (but be careful to allow for multiple superimposed plots in the file.)

Happy spinning!

__DNL:__ I followed Oscar Garcia´s hints and tips. And it worked. Fortunately some times ago R. Schorn had sent 3dv.exe, but I also could find it on the net (together with KIT.ZIP and MAWK). I have contact to Oscar and it seems that he is trying making other DERIVE files of 3D-objects suitable for his 3dviewer. The files produced by my ACD.EXE (DNL#24) have another format than the DERIVE-produced .ACD-files. Oscar sent also a patch to include automatic scaling. If you are interested in that, then please call or write, I´ll send it to you. I add a screen shot of 3dviewer:



(3dv is still working the DOS-environment, Josef)

__John Alexiou, USA__ **ja72@prism.gatech.edu**

I am a Mechanical Engineering graduate student at Georgia Tech. We are using DERIVE for WINDOWS for Multibody Dynamics and I was wondering if there are other people out there doing the same. I was interested in exchanging some information and maybe .MTH-files.

I have noticed that in many packages people use a `LIM(f(x),x → a)` for a substitution, unfortunately this is SLOW and cannot handle vectors very well.
I have a solution to this problem. You can use the `ITERATE` function to iterate once all the variables with their values. For example

#1: SUBEQ(eq, expr) := ITERATE(eq, LHS(expr), RHS(expr), 1) User

#2: $\mathrm{SUBEQ}\left(\left[\frac{x}{x^2+y^2}, \sqrt{(1-x\cdot y)}\right], [x = 1, y = 0]\right) = [1, 1]$ User=Simp(User)

#3: $\mathrm{SUBEQ}\left(\left[\frac{x}{x^2+y^2}, \sqrt{(1-x\cdot y)}\right], [x = a + b, y = a - b]\right)$ User

#4: $\left[\frac{a+b}{2\cdot(a^2+b^2)}, \sqrt{(-a^2+b^2+1)}\right]$ Simp(#3)

It applies to any vector/matrix size/shape, and to number/symbolic computations. In fact you can improve RK and EULER in ODE_APR.
john

Terence Etchells, Liverpool, UK **t.a.etchells@livjm.ac.uk**

Hi ALL
I´ve done a fair bit of programming in my time using *DERIVE*, and to my knowledge there are no books on it at the moment. However, I have a book on Numerical Analysis published this month. There are many activities in this book that deal with programming *DERIVE*; for example programming the Newton Raphson Method; Cobweb diagrams; Secant method; Lagrange interpolating polynomials; Simpson´s rule; Richardson´s method, Euler and Runge Kutta methods for ODEs etc. etc.

Contact Philip Yorke at Philip@chartwel.demon.co.uk for further details.
Incidently.
I have been playing around with operators recently with DFW 4.03 and I discovered for myself some interesting mathematics. Try this:
The operator D on f(x) we define as Df(x) = f(x+h) – f(x). (I would prefer to use capital delta for this but I doubt whether the character will transmit through e mail).

#1: F(x) :=

#2: $\Delta(p, x, h) := \left(\lim_{x\to x+h} p\right) - p$

Simplify Δ(F(x), x, h):

#3: $\Delta(F(x), x, h) = F(x + h) - F(x)$

Now I want the operator Δ to act on F again i.e. Δ^2 F(x): No Problem

#4: $\Delta(\Delta(F(x), x, h)) = F(x + 2\cdot h) - 2\cdot F(x + h) + F(x)$

What about Δ^3 F(x) or even Δ^10 F(x)? We can use the ITERATE command

#5: ITERATE(Δ(s, x, h), s, F(x), 3)

What do we get?

#6: $F(x + 3\cdot h) - 3\cdot F(x + 2\cdot h) + 3\cdot F(x + h) - F(x)$

#7: ITERATE(Δ(s, x, h), s, F(x), 10)

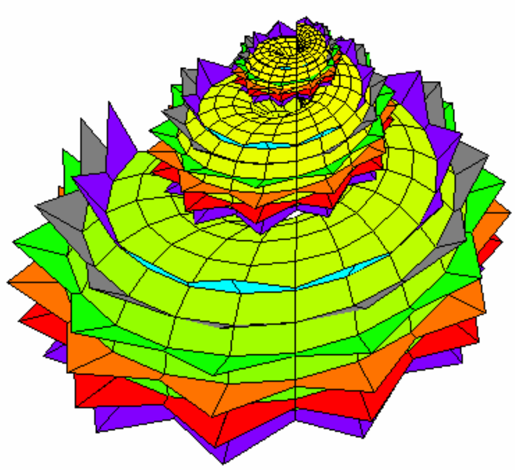
Are you surprised that we get

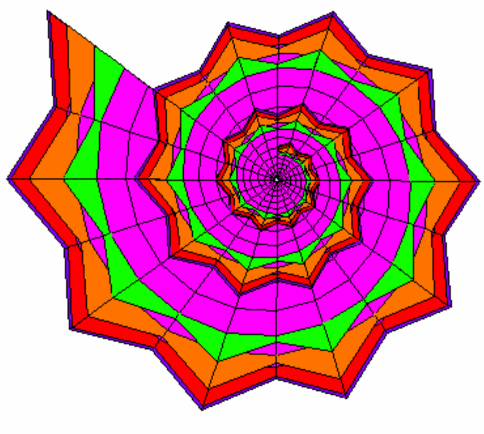
#8: $F(x + 10\cdot h) - 10\cdot F(x + 9\cdot h) + 45\cdot F(x + 8\cdot h) - 120\cdot F(x + 7\cdot h) + 210\cdot F(x + 6\cdot h) - 252\cdot F(x + 5\cdot h) + 210\cdot F(x + 4\cdot h) - 120\cdot F(x + 3\cdot h) + 45\cdot F(x + 2\cdot h) - 10\cdot F(x + h) + F(x)$

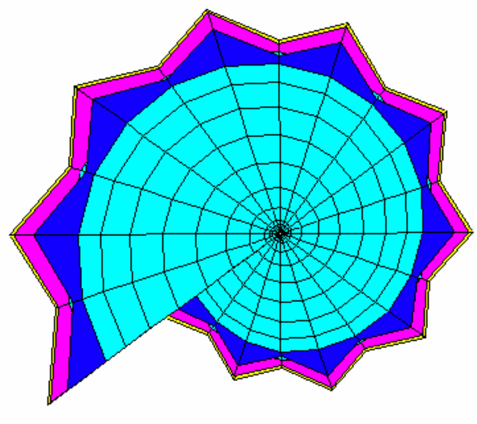
As you can see, we can use now the Δ, Josef.

Mr Schmidt sent a wonderful MATHEMATICA - graphic of a snailhouse together with the functions to produce it. I tried with DERIVE and brought the snail into life using ACD and ACROSPIN. Many thanks, Mr. Schmidt for your idea. The snail is a new object in my collection of animated graphs. (The 1997 snail can be found on page 36.)

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

[pi10 := π/10, r := 0.0972·(φ/20 - 3)^2 + 0.125]

z0 := 0.0465·(6 - φ/20)·φ·pi10

az := IF(a = 1, 1, IF(FLOOR(φ/2) ≠ φ/2, 1.1, 1.5))

HAUS(φ, θ) := VECTOR(VECTOR([COS(φ·pi10)·r·(1.1 + COS(θ)), SIN(φ·pi10)·r·(1.1 + COS(θ)),
  r·SIN(θ) + z0], φ, 0, 57), θ, 0, 2·π, pi10)

ZACK(φ, a, z) := VECTOR(VECTOR([COS(φ·pi10)·r·(1.1 + az·COS(z·pi10)), SIN(φ·pi10)·r·(1.1 +
  az·COS(z·pi10)), r·az·SIN(z·pi10) + z0], φ, 0, 57), a, 1, 1.2, 0.2)

[HAUS(φ, θ), VECTOR(ZACK(φ, a, u), u, -2, 3)]
```







The respective Graphs produced in the 3D Plot Window of DERIVE 6

Terence Etchells, Liverpool, UK t.a.etchells@livjm.ac.uk

DERIVE: The Next Generation

You may or may not be aware that I am studying for a PhD in Neural Networks. (As an aside I am writing an article for the DNL on how I programmed Derive to train a neural network.) My experiences lead to the following request.

As you know DERIVE is my number one program. If I were to have only one program on my desert island PC it would be DERIVE version 4.03. As a teacher of 16-19 mathematics it has always been the best and the Windows version is superb.

However, now as a user at the University level I can see why some universities prefer other CAS such as MAPLE(MATLAB) and MATHEMATICA. Granted they are more difficult to learn and to use; and are not very intuitive. But, they have an important facility that DERIVE does not. That is, what I call, **Vertical Programming**. i.e.

```
For while
If .... Then .....
End While
```

as opposed to, what I call, horizontal progamming,

```
vector(if(.... then ......),r,...)
```

I have attatched a MTH file that performs Neural network training. The effort required to program Derive to do this was considerable, I really enjoyed doing it but thats by the by, and most of the effort was in eliminating repetition of calculation through auxiliary functions. Something that I am sure, although I don't know, would be made easier with vertical programming.

Another aspect which I find a little frustrating is the way DERIVE works as a Turing machine, i.e. all calculations are done before any ouptut is given. Which means that if I run a function that requires a long compute time (say 48 hours) and after 24 hours(which is required in my work) and wish to see where I am up to, I can't because interrupting loses all the information. Similary, having continous output (e.g. to look at convergence) as the function runs and producing dynamic plots of results that change with each new calculation is not possible.

I am aware that a great deal of time and effort has been invested in the Windows version and as fan and power user of Derive I would ask (plead!) that a programming window be added in the next version. I think the programming window in the TI 92 is a good model on which to base a DERIVE programming window. I am sure, but again I don't know, that all the programming functions such as FOR, WHILE, OUTPUT, IF, THEN, PLOT are already there in muLisp but the DERIVE user can't get at them or utilise them.

I know that we users are always asking for additions and improvements and after such a mammoth effort as DfW more wishes must seem as a pain in the butt, but that is the nature of the beast.

To end a little question, is a UNIX version of DERIVE a realistic possibility? (Please don't swear!!!!).

All my hopes!

Humberto M.Pereira Silva, Rio Tinto, Portugal

Dear friend,

… Reading newsletter #23 in which V. Hermans from Holland blamed for the lack of any incentive to pupils of High School to work with electronic machines in the classroom, will it be Texas, Casio, HP or else. In Portugal there are some new ideas two years ago when the use of electronic calculators were introduced in the last three terminal years (10, 11, 12) but unlikely were forbidden in the final examinations. I personally think that would be examination to test the skill of pupils in Calculus, because I think it will be more important in their future life than some subjects that has been taught in school benches. I´m an electronic engineer who works and teaches the last form in High School and the first two years in University in Maths Analysis and Linear Algebra (matrices ...).

At the other hand we have to congratulate for the excellent quality achieved with the new *DERIVE* for WINDOWS that I have been using for a month or so, it comes with some new fine features and is even easier to work quicker and powerful.

Hoping to hear from you soon in meantime I wish you a nice and profitabel 1997. Yours H.M.P.S

Soft Warehouse Hawaii **swh@aloha.com**

Hello Power Users of DERIVE

To keep you up to date, the following is a summary of the enhancements made in DERIVE for Windows version 4.03

1. The quality of printed screen images of 2D and 3D plots has been improved by the elimination of spurious points.
2. The date/time strings on printouts now use the format (dd/mm/yy or mm/dd/yy) appropriate for the user´s own country.
3. Comments in the DMO files are now handled like they were in the DOS version of DERIVE.
4. A File Change Directory command is also included in the 2D and 3D-plot windows.
5. In 3D-plot windows, it is no longer required that an expression be plotted before issuing the Write to Acrospin command.
6. If ACROSPIN.EXE is in the \DfW directory (as it should be), then it will automatically be found by the Write To Acrospin command.
7. When multiple expressions are highlighted and copied to the clipboard by pressing Ctrl+C, the highlighted rectangle is no longer corrupted.
8. In 2D and 3D-plot windows, allow pi to be entered in upper or lower case.
9. Allow quoted strings and underscores to be used in function and variable names given in the Declare Function Definition, Declare Variable Value and Declare Variable Domain commands.
10. If a vector is highlighted and the solve icon is clicked on, the Solve System instead of Solve Algebraically command is called (this was a serious deficiency in earlier versions of DfW).

The following enhancements are in both DfW and DfD version 4.03:

11. A new built-in LOAD (file-name) function makes it possible to have DMO files automatically load utility files.
12. The new function DIF_NUMERIC(y,x,x0,h,n) defined in NUMERIC.MTH numerically calculates the nth derivative of y(x) at x = x0 using a centered finite difference with a step of size h.
13. The DSOLVE1 function in ODE1.MTH is now better able to solve separable ODEs.
14. Some problems with the DIRECTION_FIELD function defined in ODE_APPR.MTH have been corrected.
15. Recognize that SUM((ax+b)^n,x,c,inf) can be expressed in terms of the Riemann Zeta function even if n is symbolic provided that n > 1.

Horst Scheppelmann, Hameln, Germany

..... Ist Ihnen ein Buch zu DERIVE bekannt, das sich überwiegend mit vektorieller analytischer Geometrie befaßt, wie sie z.B. mit dem Lehrbuch von LAMBACHER-SCHWEIZER aus dem Klett-Verlag in der Sekundarstufe II (Klassen 11-13) unterrichtet wird.
Mit freundlichen Grüßen und Wünschen für das Neue Jahr. Ihr H.S.

DNL: *The question is directed to our German friends, but maybe there is anybody among you who does know a book dealing with vectorial analytical geometry for Secondary Schools (forms 11-13)?*

Michael S Mullen, Austin, TX, USA **msmullen@tenet.edu**

Why can *DERIVE* XM 3.01 not solve 2^x = x^2? The plot shows all three solutions including the obvious one x = 2.
I tried to declare x as a complex variable .... and I tried to expand the equation prior to solving, but *DERIVE* just yields 2^x – x^2 = 0.
Any assistance is appreciated.

DNL*: There were several answers, all of them with similar contents:*
One example: I had the same experience on the TI-92 ..... until I changed the mode to Approximate.

```
              x     2
#1:   SOLUTIONS(2  - x  = 0, x, -5, 10)

#2:   [2, 4]

#3:   [-0.7666646959]
```

DERIVE 6:
#2 is the "simplified" result and
#3 the "approximated" one.

Mike Hammet, Greenville, SC, USA **Hammet_Mike/furman@furman.edu**

I have just received and started using DfW. I want to show you something that is good for comparing its speed with other versions of *DERIVE* and for checking the speed of your computer also. It is the following math problem.

2^101 – 1 (which expands quickly gives)
2535301200456458802992406410751 (which factored slowly)
7432339208719 * 341117531003194129

Several years ago I ran this on a 286/8 MHz machine and it factored the number in 12 hours, 20 minutes cpu time (That was with *DERIVE* 1.6). I wondered if it could even do the problem before I finally let it run overnight and returned the next day to find that it worked. Since then I have tried it on some much faster computers with newer versions of *DERIVE*.
Here are the cpu times for some other computers and versions of *DERIVE*.

DERIVE for WINDOWS:	486/50 – 22.3 minutes 486/100 – 12.1 minutes
DERIVE 3:	486/50 – 47.0 minutes 486/100 – 25.4 minutes
DERIVE XM	486/50 – 24.0 minutes 486/100 – 13.6 minutes

Here *DERIVE* 3 and *DERIVE* XM were running under DOS only with WINDOWS off. By the way, running *DERIVE* 3 under WINDOWS slows it down by a factor of only about 1 or 2 percent but *DERIVE* XM is a disaster in WINDOWS, which slows it down by about a factor of five. It takes XM over two hours to factor the number of running under WINDOWS on a 486/50!

So it seems DfW is even a little faster than XM.

This is a fun problem to run on *DERIVE*. I would like to see the speed of a Pentium 200 on it. If you feel it is worth it please feel free to share it with other users.

***<u>DNL</u>**: Some answers:*

M.S.Mullen: Your test was a great idea. 2^101–1 really throws *DERIVE* into a tissy. After a little brute force I found that 2^93 – 1 is a fairly good test with a much shorter duration.
On my 486DX100 the expand - factor sequence took 14 sec for *DERIVE* XM and 11 sec for *DERIVE* loaded low. On my old 386 SX16 XM took 63 sec ....
Have fun

M.S.Mullen has a nice question in his mail:
"Who is General Failure, and why is he reading Drive C?"

Scott Guth: Hello DERIVE enthusiasts!
I thought I´d offer my results to FACTOR(2^101–1,Rational).

Using *DERIVE* for WINDOWS:	Pentium 90MHz/32MB RAM w/ WIN95:	13.6 minutes
Using Maple V Release 3:	Same computer as above;	4.02 minutes

That sure says something about the Maple kernel!

Dr.N.B.Backhouse: Dear Derivers! Let me give you some more timings of factorisations of 2^101–1:
All done on a 133 Mhz Pentium (32 Megs of RAM, which may be or may not be relevant):

DERIVE 3.00:	13.2 minutes;	
Macsyma 2.10:	9.0 minutes;	
Mathematica 3.0:	2.5 minutes	
Maple V5 Release 4:	1.75 minutes	
Ubasic:	0.75 minutes,	Didn´t they do it well?

Now can someone tell me if it is significant that both *DERIVE* and Macsyma are Lisp based, whereas Maple and Mathematica are written in C. ***(Read Al Rich´s interesting answer on page 46, Josef)***

Leon Magiera, Wroclaw, Poland magiera@rainbow.if.pwr.wroc.pl

Given:

$$\#1: \quad f := \frac{1}{\sqrt{\left(2\cdot E - \left(\frac{V0\cdot ro}{r}\right)^2 - \frac{k\cdot r^2}{m}\right)}}$$

$$\#2: \quad R1 := \sqrt{\left(\frac{E}{m} + \sqrt{\left(\left(\frac{E}{k}\right)^2 - \frac{Vo^2\cdot ro^2\cdot m}{k}\right)}\right)}$$

$$\#3: \quad R2 := \sqrt{\left(\frac{E}{m} - \sqrt{\left(\left(\frac{E}{k}\right)^2 - \frac{Vo^2\cdot ro^2\cdot m}{k}\right)}\right)}$$

$$\#4: \quad g(r) := \int f\, dr$$

DERIVE evaluates g(r):=INT(f,r) and g(R2)-g(R1) but doesn't simplify INT(f,r,R1,R2).
Best regards, Leon Magiera

A. van der Meer, Twente, Netherlands A.W.J.vanderMeer@math.utwente.nl

Consider the following DERIVE session:

$$\#1: \quad F(x) := \int \frac{1}{a + b\cdot SIN(x)}\, dx$$

Antiderivative calculated "by hand":

$$\#2: \quad G(x) := \frac{2}{\sqrt{(a^2 - b^2)}}\cdot ATAN\left(\frac{a\cdot TAN\left(\frac{x}{2}\right) + b}{\sqrt{(a^2 - b^2)}}\right)$$

$$\#3: \quad \frac{d}{dx} G(x) = \frac{1}{b\cdot SIN(x) + a}$$

$$\#4: \quad G(2\cdot\pi) - G(0) = 0$$

Conclusion: G is not the antiderivative on the whole real axis. In this case G has discontinuities on [0, 2*pi]
For this example Derive has a trick to remove these discontinuities by adding a step function:

Calculated by *DERIVE*: Simp(#1)

$$\#5: \quad F(x) := \frac{2\cdot ATAN\left(\frac{2\cdot b\cdot COS(x)}{2\cdot b\cdot SIN(x) + (\sqrt{(a - b)} + \sqrt{(a + b)})^2}\right) + x}{\sqrt{(a + b)}\cdot\sqrt{(a - b)}}$$

Which gives the correct results:

$$\#6: \quad F(2\cdot\pi) - F(0) = \frac{2\cdot\pi}{\sqrt{(a + b)}\cdot\sqrt{(a - b)}}$$

$$\#7: \quad \int_0^{2\cdot\pi} \frac{1}{a + b\cdot SIN(x)}\, dx = \frac{2\cdot\pi}{\sqrt{(a + b)}\cdot\sqrt{(a - b)}}$$

$$\#8: \quad F(4\cdot\pi) - F(0) = \frac{4\cdot\pi}{\sqrt{(a + b)}\cdot\sqrt{(a - b)}}$$

(All screenshots are from DERIVE 6.10.)

So, *DERIVE* is cleverer then you expected. In your example *DERIVE* has found that there is a possibility that certain combinations of values of the parameters may result in discontinuities in the antiderivative g(r) on the integration interval, but did not have a trick to remove these. I expect that the definite integral will be correctly calculated if you use realistic numerical values for the parameters.

Klaus Fischer, Darmstadt, Germany

Klaus Fischer suggests to include the most important and interesting WWW-sites for DERIVE and TI-92 users into the information page. Excellent idea, we will do that. Any recommendations are appreciated. See the Information page.

Jos Verhoosel, Eindhoven, Netherland **J.C.M. Verhoosel@pth.nl**

Dear *DERIVE* User,
I teach mathematics for students who will become teacher in mathematics themselves. As part of their education they invest in their 2nd year 80 hours of programming (programming structures, recurrent programming, procedures, functions, subranges etc.)

Recently I was asked if I was willing to replace this course with a new one: "Programming in *DERIVE*". I think this is a fantastic opportunity for them to extend their (and my) knowledge of *DERIVE*.

Do you know if there exists such a book or course? To my knowledge there is none.

Forced by circumstances, the course should start 3th february 1997 and concerns 2nd years´ students in mathematics and I really would like to give the course. I already found some interesting examples in some utility files of *DERIVE* and in some Titibits - files from Johann Wiesenbauer.

My question for you:

Can you give me some more programming problems in *DERIVE.* The problems should cover most of the programming facilities of *DERIVE* Also, the problems should be simple to grasp for my students and should be practical. (That means that it should be utterly clear that such a utility or subprogram can be of great use during the rest of course)

Can you give me some suggestions?
(I intend to share my experiences in the International *DERIVE* Journal if interesting)
Thanks in advance, Yours sincerely
Drs. Jos C.M.Verhoosel
Fontys PTH, Dept of Mathematics
Eindhoven, The Netherlands.

***DNL**: I could imagine that the "TURTLE GRAPHICS" article in this issue could contribute for your course and Terence´s nwe book, of course. And now, once more Terence Etchells:*

Terence Etchells, Liverpool, UK **t.a.etchells@livjm.ac.uk**

Hi Josef,

I have written a little set of functions that will produce an interpolating polynomial to approximate a function at any given points on that function. So for example, you may wish to find a polynomial that passes through the points (0,sin(0)),(pi/2,sin(pi/2)) and (pi,sin(pi)). So our function is sin(x) and the x values of the desired points are [0,pi/2,pi]. As we have 3 points the polynomial will be a quadratic. You will find attatched a MTH file INTERPOL.MTH that will do this automatically. In fact it will work for up to 20 points (i.e. a polynomial up to order 19), this can be increased by increasing the available variables in the def_vars (and the line above that resets all variables we are to use to empty) definition.

Have you found an easier way (is there? Al or Dave) of producing a vector of undefined variables, without having to type them all in. Something like VARIABLES(30,a) which would simplify to

[a,b,c,d,........,x,y,z,aa,bb,cc,dd].

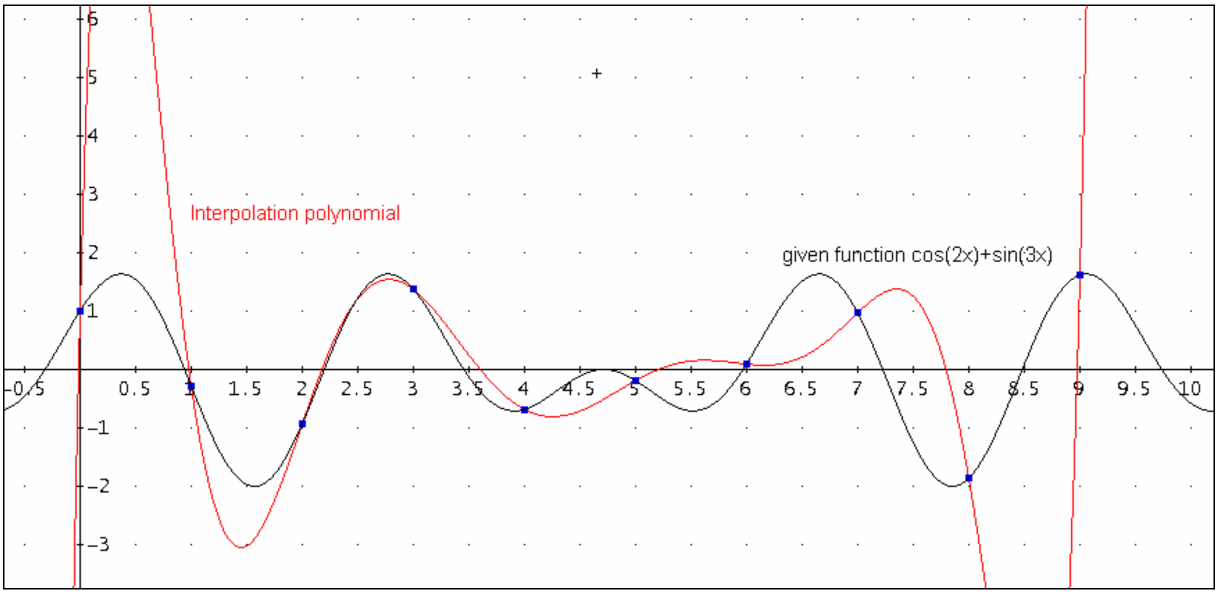
You will also notice that on line #9 the LIM function variables vector needs an EXPAND command, it doesn't work without it. Any ideas why?

If nobody has come up with a function that will do this before (don't remember seeing one in any previous DNL) you might like to use it a forth coming DNL.

```
#1:  def_vars := [a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t]
#2:  [FUNCTION(x) :=, xvalues :=]
#3:  POLYNOMIAL_AUX(dim) := VECTOR(def_vars_r·x^(r - 1), r, 1, dim)·VECTOR(1, r, dim)
#4:  POLYNOMIAL(x) := POLYNOMIAL_AUX(DIMENSION(xvalues))
#5:  POLYNOMIAL_AUX(x, dim) := VECTOR(def_vars_r·x^(r - 1), r, 1, dim)·VECTOR(1, r, dim)
#6:  POLYNOMIAL(x) := POLYNOMIAL_AUX(x, DIMENSION(xvalues))
#7:  vars_vect := VECTOR(def_vars_r, r, 1, DIMENSION(xvalues))
#8:  co_eff := (SOLUTIONS(VECTOR(FUNCTION(xvalues_r) = POLYNOMIAL(xvalues_r), r, 1, DIMENSION(xvalues)), vars_vect))_1
#9:  interpolate :=          lim              POLYNOMIAL(x)
                     EXPAND(vars_vect)→co_eff
```

Example: Interpolate cos(2x) + sin(3x) by a polynomial at x = 0,1,2,3,4,5,6,7,8

```
#10:  FUNCTION(x) := COS(2·x) + SIN(3·x)
#11:  xvalues := [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
#12:  interpolate
#13:  TABLE(FUNCTION(x), x, xvalues)
#14:  TABLE(interpolate, x, 0, 9, 0.005)
```



DNL: *Plot #12 and obtain the interpolating polynomial. #13 results in the nine given points. #14 would plot the interpolating polynomial in form of a thick line (point size medium or large and points connected).*

AN IMPLEMENTATION OF "TURTLE GRAPHICS" IN DERIVE 3

Josef Lechner[1] , Eugenio Roanes-Lozano[2] , Eugenio Roanes-Macías[3], Johann Wiesenbauer[3]

[1]Bundesgymnasium Amstetten (Austria)
[2,3]Dept. Algebra, Univ. Complutense de Madrid (Spain)
[4]Institut für Algebra, Tech. Univ. Wien (Austria)

Introduction

If you have worked previously with any LOGO dialect you will know about the nice possibilities of the Turtle Graphics. We had already developed implementations for Turbo-Pascal [Ro4] and Maple V ([Ro1], [Ro2], [Ro3]). Now we have developed another one for DERIVE 3[1], that is presented below.

Two Remarks

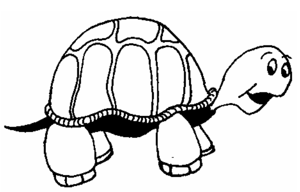
To follow the article, a certain knowledge of the possibilities of Derive and Turtle Graphics is supposed. [Ab] is a very nice introduction to the later, and [A-dS] is an impressive collection of ideas for its use.
Note that in some early versions of Derive 3 evaluation is made in such a way that this code does not work. The code has been checked in versions 3.13 (Classic and XM) and Derive for Windows Beta v. 0.1.

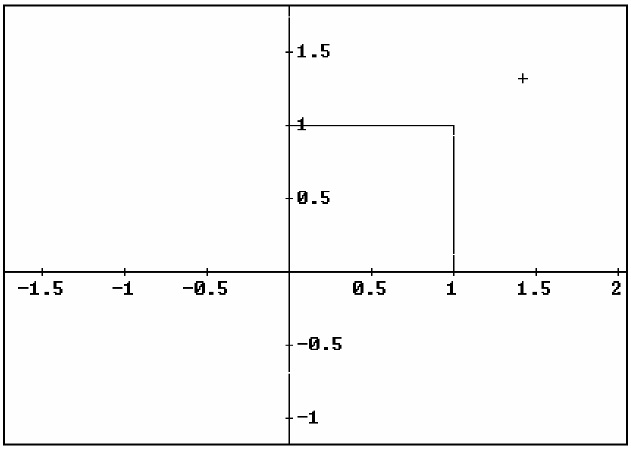
Getting Started

We should begin by setting the graphics in Connected Mode and the Color in the non Auto Mode. Then the Turtle file turtles_ut.mth should be loaded normally (or as a utility file in the background).

Example 0: For instance, to draw the square of vertices (0,0),(0,1),(1,1),(1,0) it is enough to Plot

```
[FD(1), RT(90), FD(1), RT(90), FD(1), RT(90), FD(1), RT(90)]
```





[1] Works now also for DERIVE 6, Josef

Or, even more simple, to make the assignation

```
to_rep:=[FD(1),RT(90)]
```

(and to Simplify) and Plot

```
REPEAT(4)
```

Note that the syntax adopted for the REPEAT command is not the usual in Logo.
Observe that, unfortunately, unlike in Logo, there is no visible cursor for the turtle in the graphics screen.

About the implementation (the content of the turtles_ut.mth file)

The first two lines of the turtles_ut.mth file are

```
Angle:=Degree
Precision:=Approximate
```

(in order to increase the speed; nevertheless this can -should- be omitted in some cases -see [Ro3]-).
The main problems we had to produce this implementation were related to the difficulties to change, from a programme, assignations made to variables. The following auxiliary functions are to be used

```
NEWX(u):=(xcor:=u)
NEWY(u):=(ycor:=u)
NEWD(u):=(heading:=u)
```

The turtle is, at any moment at point (xcor, ycor) and heading towards "heading" (counting clockwise from the usual halfaxis y+). The user can check the value of these variables by simplifying them.
Advancing a length l_ can be implemented easily:

```
FD(l_):=[xcor,ycor;NEWX(xcor+l_COS(90°-heading)),NEWY(ycor+l_SIN(90°-heading))]
```

and moving backwards the same distance can be obtained from the previous function

```
BK(l_):=FD(-l_)
```

When the turtle turns, only the direction (heading) is altered. The new value is reduced modulo 360 in order not to get high values for the "heading" variable after many rotations

```
RT(a_):=[xcor,NEWY(ycor+ 0·NEWD(MOD(heading+ a_°,360°)))]
LT(a_):=RT(-a_)
```

The last of the typical turtle-commands is "home" (return to the centre of the screen facing upwards):

```
home:= [NEWX(0), NEWY(0), NEWD(0)]
```

But there are some more standard turtle-commands mixing turtle ideas and Cartesian coordinates. To go to a certain position of known coordinates can be done using

```
SETPOS(x_, y_):=[xcor,ycor;xcor:=x_,ycor:=y_]0
```

The actual position is given by

```
pos := [xcor, ycor]
```

And to change only one of the coordinates of the current position can be done using

```
SETX(x_):=[[xcor,ycor],[xcor:=x_,ycor]]
SETY(y_):=[[xcor,ycor],[xcor,ycor:=y_]]
```

It is also possible to force the turtle to look in direction a_ with

```
SETH(a_):=[xcor,NEWY(ycor + 0·heading:=a_)]
```

or to look towards a certain point (x, y) with

```
SETHTOWARDS(x_,y_):=[xcor,NEWY(ycor+0·heading:=ATAN(x_-xcor,y_-ycor))]
```

Note that, to clear the screen, there is no ClearScreen command. The Delete/All in Derive's Plot menu should be used instead.

Another standard feature of Logo is the REPEAT command. We couldn't maintain the standard syntax but the idea is similar.

```
REPEAT(k_) := IF(k_ = 1, to_rep, APPEND(REPEAT(k_ - 1), to_rep))
```

I tried successfully:

```
REPEAT(k_):=VECTOR(to_rep, i, k_)
```

Finally, the variables are initialised:

```
xcor:=0
ycor:=0
heading:=0
```

About REPEAT

What has to be repeated has to be stored in the to_rep variable (list) and the only argument of REPEAT is how many times has to be repeated what is stored in that list. Therefore, to execute repeatedly a list of commands just type

to_rep:= [*list of commands*]
REPEAT(*number*)

and Plot.
Observe that, the way it is defined, a REPEAT can not be nested inside another REPEAT. Defining a function REPEAT2 depending on a variable to_rep2 does not always work.

Some Simple Examples

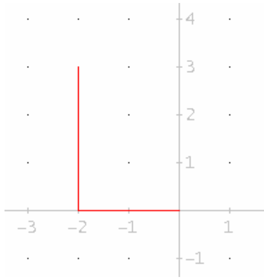
Remember:

i) To Simplify "home" and to choose Delete/All in the Plot menu in between executing the examples.

ii) (Simplify and) Plot each new expression (which is not an assignation).

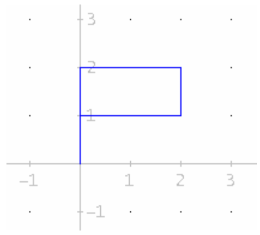
Example 1: Draw an L.

```
#2:   home
#3:   [xcor := 0, ycor := 0, heading := 0]
#4:   [LT(90), FD(2), RT(90), FD(3)]
```



Example 2: Draw a flag and return to the origin using only the Cartesian commands.

```
#5:   [xcor := 0, ycor := 0, heading := 0]
#6:   [SETY(2), SETX(2), SETY(1), SETX(0), SETY(0)]
```

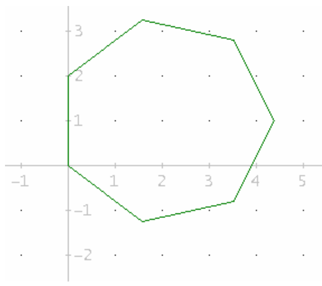


Example 3: Draw a regular heptagon

```
#7:   [xcor := 0, ycor := 0, heading := 0]
```

#8: $\text{to\_rep} := \left[\text{FD}(2),\ \text{RT}\left(\frac{360}{7}\right)\right]$

```
#9:   REPEAT(7)
```

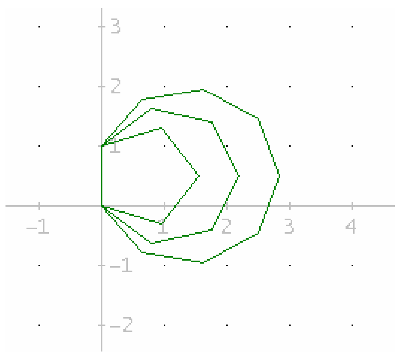


Example 4: Draw regular polygons of 5,7 and 9 sides of length 1

```
#10:  [xcor := 0, ycor := 0, heading := 0]
#11:  n_ :=
```

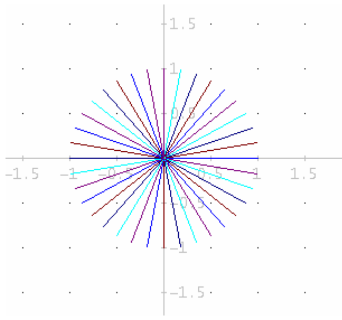
#12: $\text{to\_rep} := \left[\text{FD}(1),\ \text{RT}\left(\frac{360}{\text{n\_}}\right)\right]$

```
#13:  n_ := 5
#14:  REPEAT(5)
#15:  n_ := 7
#16:  REPEAT(7)
#17:  n_ := 9
#18:  REPEAT(9)
```



Example 5: Draw the 36 wires of a chart-wheel

```
#19:  [xcor := 0, ycor := 0, heading := 0]
#20:  to_rep := [FD(1), BK(1), RT(10)]
#21:  REPEAT(36)
```

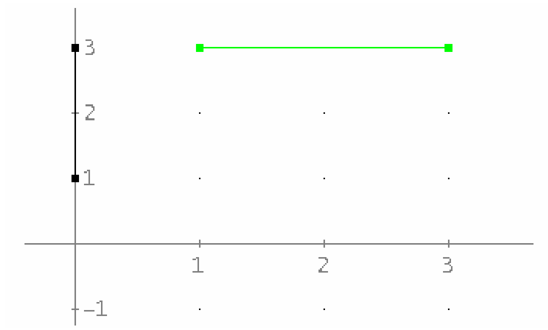


PenUp and PenDown

There is no PenUp/PenDown command in our implementation. Instead, the user has to play with only Simplifying (that only changes the position of the turtle) or Simplify and Plotting (that changes the position of the turtle and draws the correspondent segments).

Example 6: Draw two segments, the first one with endpoints (0,1) and (0,3) and the second one with endpoints (1,3) and (3,3).

```
#28:  [xcor := 0, ycor := 0, heading := 0]
#29:  FD(1)
      ⎡     0            0     ⎤
#30:  ⎣ xcor := 0    ycor := 1 ⎦
#31:  FD(2)
#32:  [RT(90), FD(1)]
                           ⎡     0            3     ⎤⎤
#33:  ⎡[0, ycor := 3],     ⎣ xcor := 1    ycor := 3 ⎦⎦
#34:  FD(2)
```



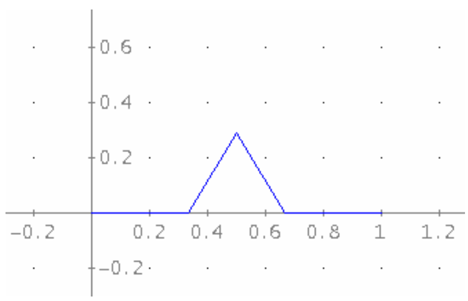
#29 and #32 must be simplified only;
#31 and #34 must be plotted.

A Recursive Example

Example 7: Function KOCH below constructs the side of a koch star with length "len" and depth "n".

```
KOCH(len, n):=IF(n > 0, [KOCH(len/3, n - 1), LT(60), KOCH(len/3, n - 1),
               RT(120), KOCH(len/3, n - 1), LT(60), KOCH(len/3, n - 1)], FD(len))
```

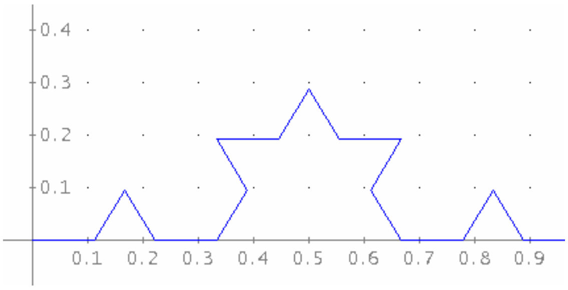
```
#36:  [xcor := 0, ycor := 0, heading := 0]
#37:  RT(90)
#38:  [0, ycor := 0]
#39:  KOCH(1, 1)
```



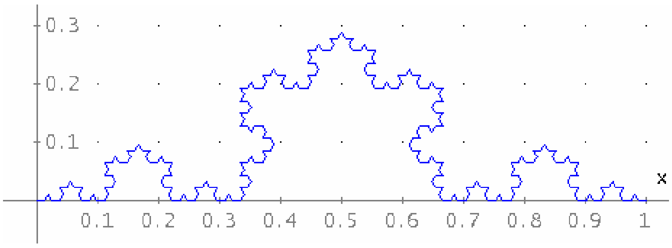
Simplify #37 (in order to have the *x*-axis as base) and then plot #39.

depth 2 and 4:

```
#40:  [xcor := 0, ycor := 0, heading := 0]
#41:  RT(90)
#42:  [0, ycor := 0]
#43:  KOCH(1, 2)
```

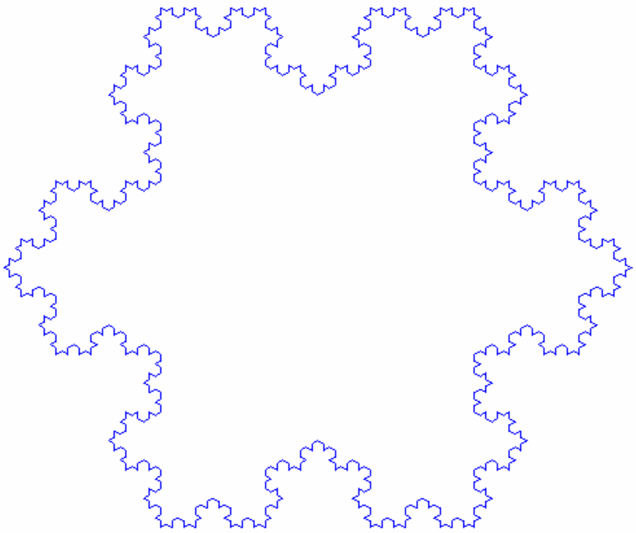



```
#44:  [xcor := 0, ycor := 0, heading := 0]
#45:  RT(90)
#46:  [0, ycor := 0]
#47:  KOCH(1, 4)
```



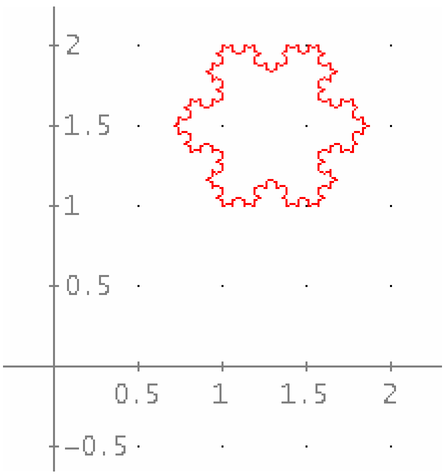
KOCH can be used to draw stars (= snow flakes):

```
#48:  [xcor := 0, ycor := 0, heading := 0]
#49:  to_rep := [KOCH(1, 4), RT(120)]
#50:  REPEAT(3)
```

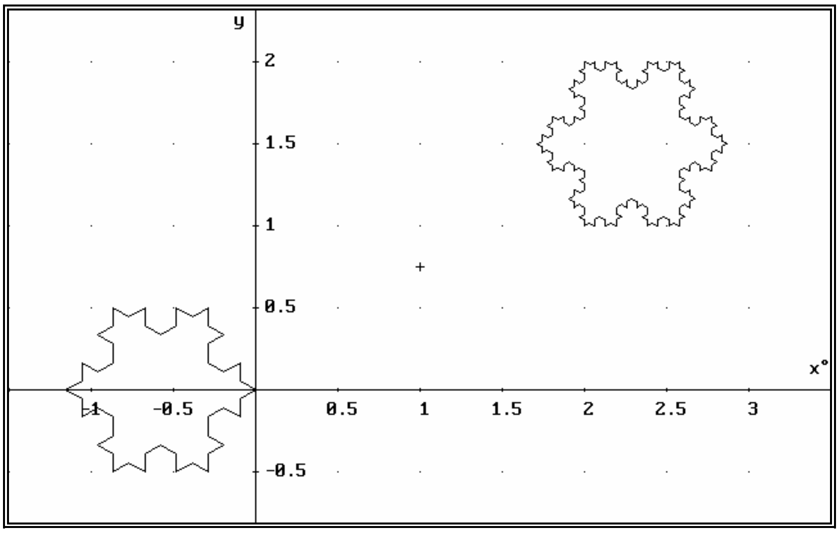


In DERIVE 6 we can write a short program, Josef:

```
       kochstar(len, n, star, n_) :=
         Prog
           star := to_rep
           n_ := 1
#51:       Loop
             If n_ = 3
                RETURN star
                star := APPEND(star, to_rep)
             n_ :+ 1

#52:  [xcor := 0, ycor := 0, heading := 0]
#53:  SETPOS(1, 1)
#54:  to_rep := [KOCH(1, 3), RT(120)]
#55:  kochstar(2, 3)
```



This is the original DOS-DERIVE screen from 1997:



Conclusions

Many geometric designs and drawings can be produced in a more convenient way using Turtle Graphics instead of Cartesian coordinates. Therefore, we think that this can be another useful tool for *DERIVE*rs.

References

[A-dS] H. Abelson, A. diSessa: *Turtle Geometry. The Computer as a Medium for exploring Mathematics*. M.I.T., 1981.

[Ab] H. Abelson: *Apple Logo*. Byte Books / McGraw-Hill, 1992.

[Ro1] E. Roanes M., E. Roanes L.: *An Implementation of "Turtle Graphics" in Maple V.2*. CAN Nieuwsbrief 12, March 1994, pages 43-48.

[Ro2] E. Roanes M., E. Roanes L.: *An implementation of "Turtle Graphics" in Maple V*. Maple Tech, Special Issue, 1994, pages 82-85 (reprint of the article above).

[Ro3] E. Roanes L., E. Roanes M.: *"Turtle Graphics" in Maple V.2*. In: Robert J. Lopez (Editor): Maple V: Mathematics and Its Application. Birkhäuser, 1994 (pages 3-12).

[Ro4] E. Roanes L.: *Automatización e implementación de algunos problemas algebraicos y geométricos*. Tesis Doctoral. Univ. Politécnica de Madrid, 1993.

[RS] A. Rich, J. Rich, T. Shelby, D. Stoutemyer: *DERIVE User Manual*. SoftWarehouse, 1994.

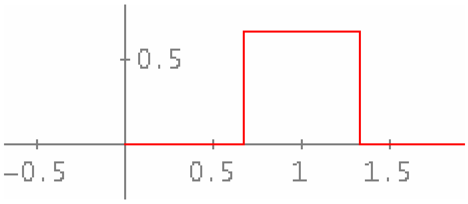
Now is 2011 and we have DERIVE 6.10 which is much more powerful than the DOS-Version from 1997.

I remember my LINDENMAYER-Systems contribution from DNL#51 and DNL#52. This has a tight connection to Turtle-Graphics.

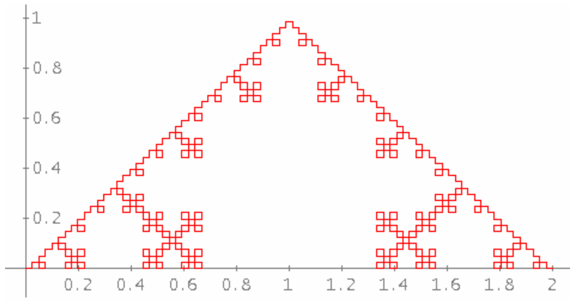
I use turtles_ut.mth for reproducing some examples:

```
hilb(len, n) := IF(n > 0, [hilb(len/3, n - 1), LT(90), hilb(len/3, n - 1),
RT(90), hilb(len/3, n - 1), RT(90), hilb(len/3, n - 1), LT(90), hilb(len/3,
n - 1)], FD(len))
```

```
#58:  [xcor := 0, ycor := 0, heading := 0]

#59:  RT(90)

#60:  [0, ycor := 0]

#61:  hilb(2, 1)
```

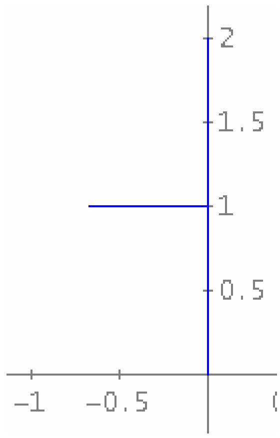



```
#62:  [xcor := 0, ycor := 0, heading := 0]

#63:  [0, ycor := 0]

#64:  hilb(2, 4)
```

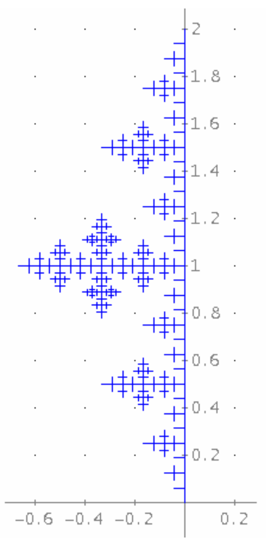



```
pat2(len, n) := IF(n > 0, [pat2(len/2, n - 1), LT(90), pat2(len/3, n - 1),
RT(180), pat2(len/3, n - 1), LT(90), pat2(len/2, n - 1)], FD(len))
```

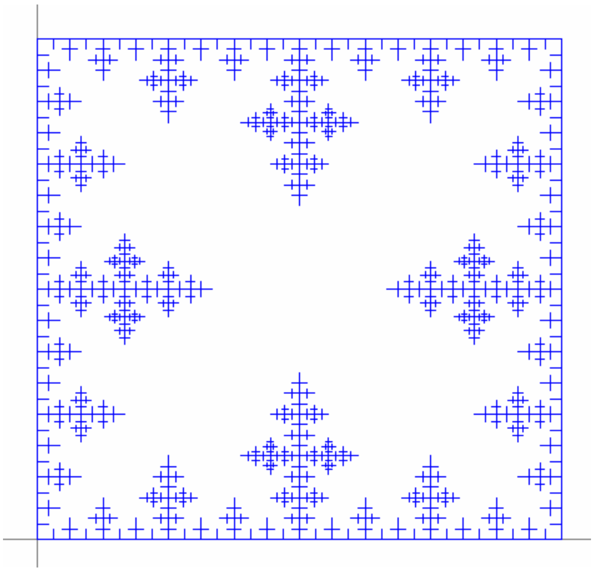
```
#67:  [xcor := 0, ycor := 0, heading := 0]

#68:  pat2(2, 1)
```




```
#69:  [xcor := 0, ycor := 0, heading := 0]

#70:  pat2(2, 5)
```

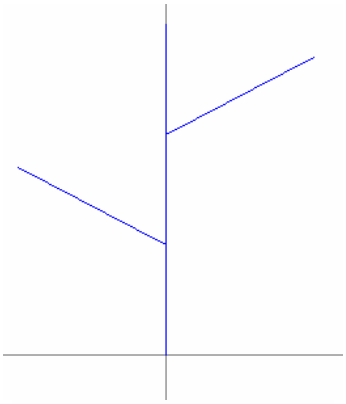


```
#71:   [xcor := 0, ycor := 0, heading := 0]

#72:   RT(90) = [0, ycor := 0]

#73:   VECTOR([pat2(2, 5), LT(90)], i, 4)
```

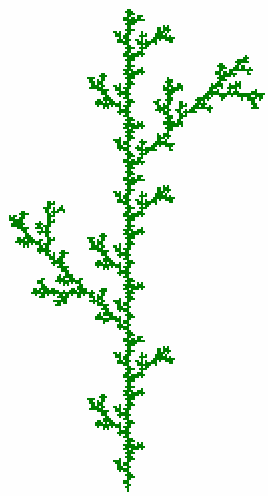


```
twig(len, n) := IF(n > 0, [twig(len/3, n - 1), LT(45), twig(len/3, n - 1),
BK(len/3), RT(45), twig(len/3, n - 1), RT(45), twig(len/3, n - 1), BK(len/3),
RT(315), twig(len/3, n - 1)], FD(len))
```

```
#75:   [xcor := 0, ycor := 0, heading := 0]

#76:   twig(3, 1)
```

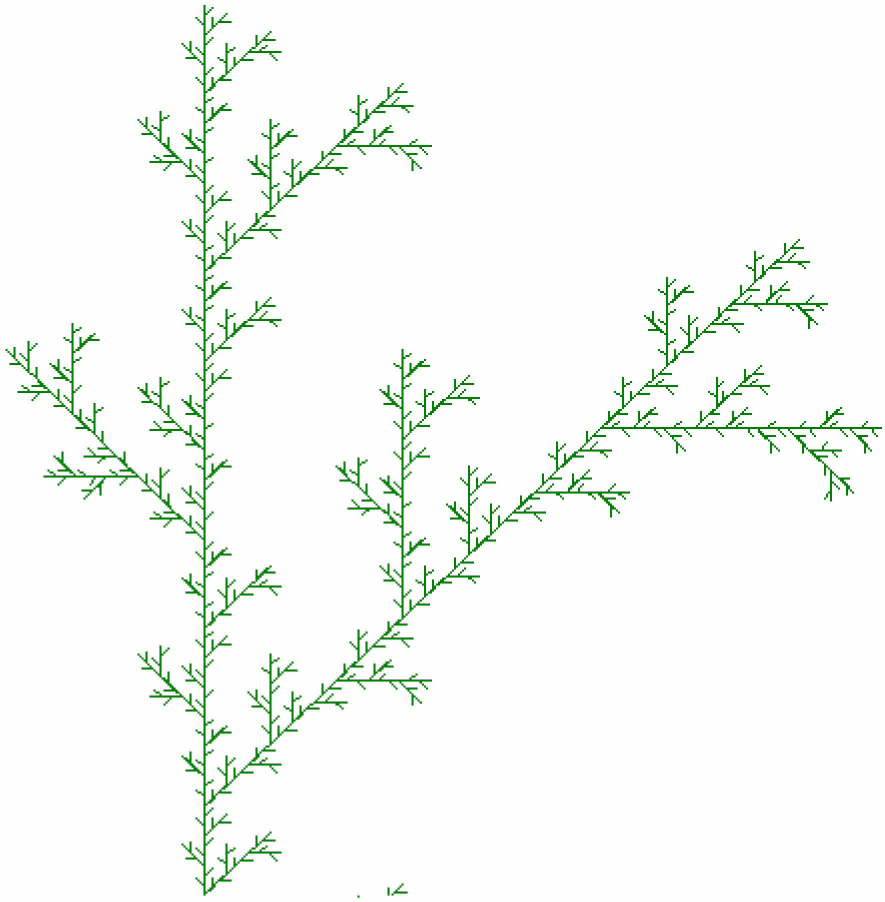


```
#77:   [xcor := 0, ycor := 0, heading := 0]

#78:   twig(3, 5)
```



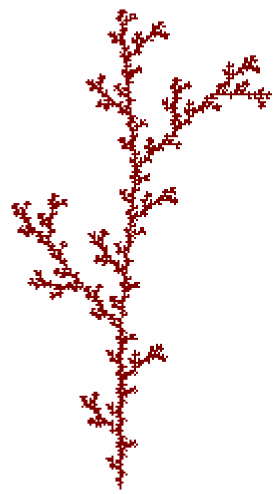
More References:

Lindenmayer Systems: DNL#51 and DNL#52
Snowflake a.o.: DNL#33 and DNL#34 (Contribution from Matia Koth)
LOGO in DERIVE: DNL#38 (Josef Lechner)



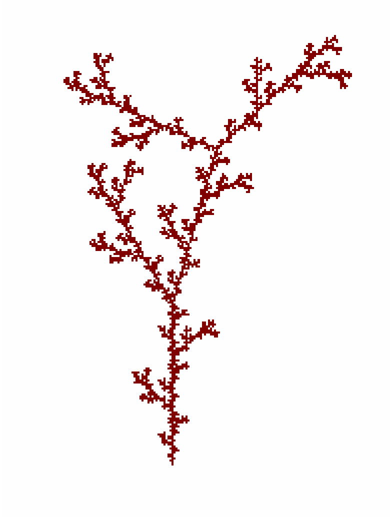
```
twig2(len, n) := IF(n > 0, [twig(len/3, n - 1), LT(40), twig(len/3, n - 1),
BK(len/3), RT(45), twig(len/3, n - 1), RT(40), twig(len/3, n - 1), BK(len/3),
RT(310), twig(len/3, n - 1)], FD(len))
```

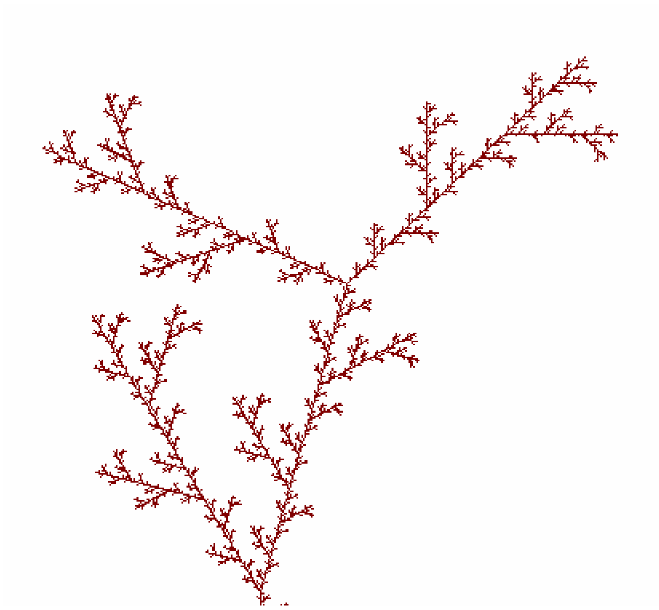
```
#80:  [xcor := 0, ycor := 0, heading := 0]

#81:  twig2(3, 5)
```

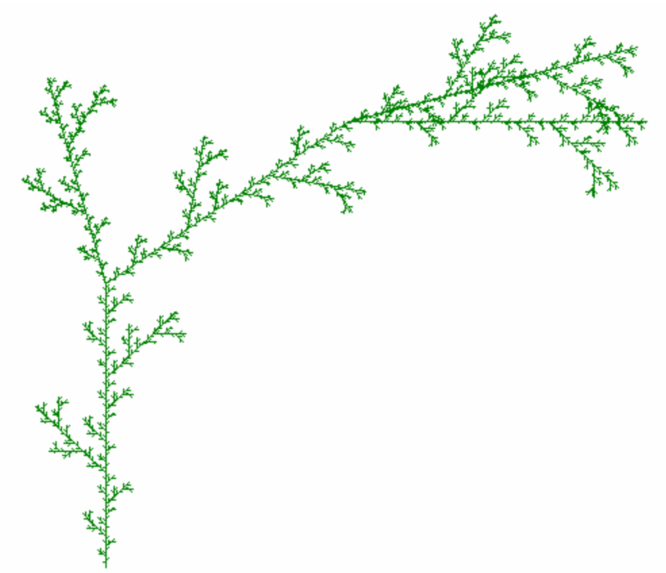


```
twig3(len, n) := IF(n > 0, [twig(len/3, n - 1), LT(30), twig(len/3, n - 1),
BK(len/3), RT(45), twig(len/3, n - 1), RT(30), twig(len/3, n - 1), BK(len/3),
RT(250), twig(len/3, n - 1)], FD(len))
```

```
#83:   [xcor := 0, ycor := 0, heading := 0]

#84:   twig3(3, 5)
```







(Two more examples on page 28)

In an earlier Letter of the Editor I mentioned that the DERIVIAN Turtle is one result of the many fruitful talks at the DERIVE Conference 1996 in Bonn. (The most important are the breaks.) After this conference began a very busy exchange of ideas via e-mail. Following the TURTLE.MTH file you might ask yourself: 'what is that, there are two := assigments in one expression?. So did I.

Fortunately Josef Lechner is very conscientious and he collects many things concerning Computer Algebra in general and concerning DERIVE in special. When I asked Josef to give an additional explanation for this strange expression he took two sheets of paper from his huge desk and asked me: "Will that be enough? I summarized my e-mails with Johann Wiesenbauer." Many thanks, Josef. I think that will do it. So find here instead of his TITBITS, J.Wiesenbauers "double assignation".

Comment of 2011: The following problems are from 1997. Later versions of DERIVE don´t behave the same. I recommend experimenting with a := expression(a) applying the various possibilities of "Simplify" from the menu- and the edit bar as well !!

Is it able to implement TURTLE graphics in *DERIVE*?

Not completely!

The reason is that *DERIVE* does not support self assignments. In *DERIVE* we meet a strict functional programming style. Especially users, which are not accustomed with that way or programming might be frustrated by the result of the assignment

```
a := expression(a)
```

One example: The user enters

```
#1:  a := 3                                   User
#2:  a := a + 7                               User
```

and expects a = 10, but he obtains a = a + 7, and simplifying again and again:

```
#3:  a + 7                                    Simp(#2)
#4:  a + 14                                   Simp(#3)
#5:  a + 21                                   Simp(#4)
```

So, what can be done? Let´s try with the assignment NEWX(u) : = x := u with u being an expression in x. That works because expressions in function parameters are evaluated before passing. That is an (interesting) alternative to the ITERATES command.

```
#7:  NEWA(u) := (a := u)                      User
#8:  a := 1                                   User
```

#9: $\text{NEWA}\left[\frac{1}{2}\cdot\left[a + \frac{2}{a}\right]\right]$ User

```
#10: 1.5                                      Simp(#9)
#11: a = 1.5                                  User=Simp(User)
#12: 1.41666                                  Simp(#9)
#13: a = 1.41666                              User=Simp(User)
#14: 1.41421                                  Simp(#9)
#15: a = 1.41421                              User=Simp(User)
```

Obviously the variable has changed its value!!

Let´s try once more. Set back the value for a. Watch the annotations, please.

```
#17: a := 1                                                              User
#18: 1.5                                                            Simp(#9)
#19: 1.41666                                                        Simp(#9)
#20: 1.41421                                                        Simp(#9)
#21: 1.41421                                                        Simp(#9)
#23: [NEWX(u) := (x := u), NEWY(u) := (y := u), NEWA(u) := (a := u)]   User
#24: FWD(l, w) := [NEWX(x + l·COS((a + w)·°)), NEWY(y + l·SIN((a + w)·°))
     + 0·NEWA(a + w)]
```

It seems to work:

```
#25: home := [x := 0, y := (a := 0)]                                     User
#26: quadrat := [home, FWD(2, 0), FWD(2, 90), FWD(2, 90), FWD(2, 90)]   User
```

$$\#27:\quad \begin{bmatrix} 0 & 0 \\ 2 & 0 \\ 2 & 2 \\ 0 & 2 \\ 0 & 0 \end{bmatrix} \qquad \text{Simp(\#26)}$$

The big disillusion:

```
#28: home = [0, 0]                                            User=Simp(User)
```

$$\#29:\ \text{VECTOR(FWD(2, 0), i, 1, 4)} = \begin{bmatrix} 2 & 0 \\ 2 & 0 \\ 2 & 0 \\ 2 & 0 \end{bmatrix} \qquad \text{User=Simp(User)}$$

```
#30: InputMode := Word                                                   User
```

$$\#31:\ \text{ITERATES(FWD(2, 0), turtlepos, home, 4)} = \begin{bmatrix} 0 & 0 \\ 2 & 0 \\ 2 & 0 \\ 2 & 0 \\ 2 & 0 \end{bmatrix} \qquad \text{User=Simp(User)}$$

What is possible in a list, does not work in a VECTOR!!

#33: a := 1 User

#34: $\left[\text{NEWA}\left[\frac{1}{2}\cdot\left[a+\frac{2}{a}\right]\right],\ \text{NEWA}\left[\frac{1}{2}\cdot\left[a+\frac{2}{a}\right]\right],\ \text{NEWA}\left[\frac{1}{2}\cdot\left[a+\frac{2}{a}\right]\right]\right]$ User

#35: [1.5, 1.41666, 1.41421] Simp(#34)

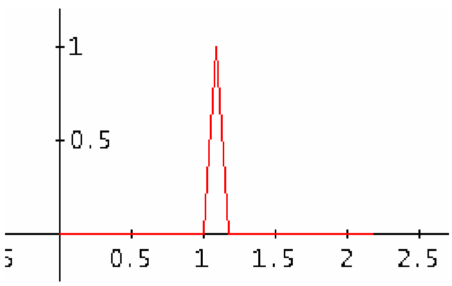
#36: a := 1 User

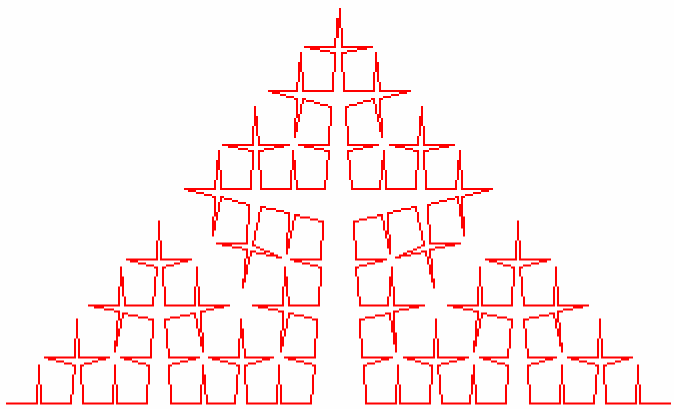
#37: $\text{VECTOR}\left[\text{NEWA}\left[\frac{1}{2}\cdot\left[a+\frac{2}{a}\right]\right],\ i,\ 1,\ 3\right] = [1.5,\ 1.5,\ 1.5]$ User=Simp(User)

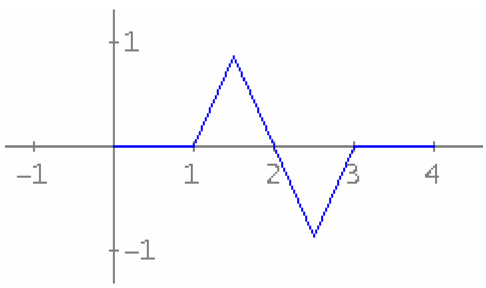
Conclusion: To implement a Turtle graphic in *DERIVE* it would be necessary for the program to support a self assigment of the variables and/or the the support of composition of functions would be improved.

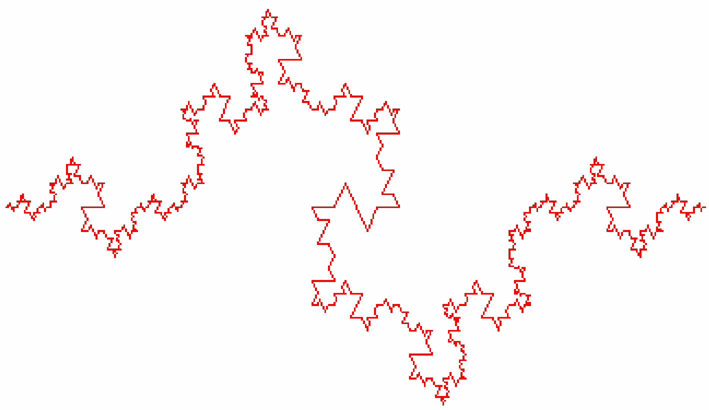
So I will close: And yet it works. I tried to translate Josef (L)´s comments. I hope that he will recognize the sense of his comments in my words. Josef (B).

J.Wiesenbauer has promised to check the DERIVE utility files for improvements using the new built in DfW and DfD4 functions and capabilities. Do you have ideas?







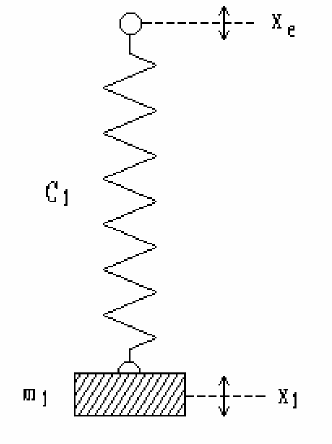


Tilgung fremderregter Schwingungen durch abgestimmte Ankopplung

Cancellation of Separately Excited Oscillations by a Tuned Connection

Leo H. Klingen, Bonn, GER

Eine rotierende Maschine mit exzentrischen beweglichen Teilen steht auf einem Fundament, das beim Lauf der Maschine möglichst in Ruhe bleiben soll. Nicht immer stellt eine möglichst große Masse des Fundaments eine optimale Lösung dar. Bei schwingungsfähigem Fundament m_1 haben wir den wohlbekannten Fall der Federschwingung (Federkonstante c_1) mit periodischer Fremderregung x_e, die hier beim prinzipiellen Ansatz von Figur 1 am Aufhängepunkt der Feder angreift.



Figur 1

A rotating machine with eccentric moveable parts is placed on a basement which should keep as motionless as possible when the machine is running. Sometimes the optimal solution is not to have the mass of the foundation as big as possible. With an oscilllating (vibrating) basement m_1 we recognize the well known case of a spring oscillation (spring constant c_1) with periodical separate excitation which acts in the spring´s suspension point (figure 1).

Due to Hooke´s Law and Newton´s Dynamic Fundamental Law we obtain the motion equation when neglecting damping.

Mit den Daten $c_1 = 1$, $m_1 = 2$ und $x_e = 4\cos(\sqrt{3}\,t)$ und der realen Federdehnung $(x_1 - x_e)$ ergibt sich die Bewegungsgleichung nach dem Hookeschen Gesetz und Newtons dynamischen Grundgesetz, wenn Dämpfung vernachlässigt wird, zu

$$2\ddot{x}_1 = -(x_1 - x_e)$$

oder / or

$$\ddot{x}_1 + 0.5x_1 = 2\cos\left(\sqrt{3}\,t\right)$$

Man kann sie exakt lösen mit

```
#3:   DSOLVE2_IV(0, 0.5, 2·COS(√3·t), t, 0, 0, 0)
```

#4:

$$-\left(\frac{2\cdot\sqrt{6}}{5}+\frac{2}{5}\right)\cdot\cos\left(t\cdot\left(\sqrt{3}-\frac{\sqrt{2}}{2}\right)\right)\cdot\cos\left(\frac{\sqrt{2}\cdot t}{2}\right)+\left(\frac{2\cdot\sqrt{6}}{5}+\frac{2}{5}\right)\cdot\sin\left(t\cdot\left(\sqrt{3}-\frac{\sqrt{2}}{2}\right)\right)\cdot\sin\left(\frac{\sqrt{2}\cdot t}{2}\right)+\left(\frac{2\cdot\sqrt{6}}{5}-\frac{2}{5}\right)\cdot\cos\left(t\cdot\left(\sqrt{3}+\frac{\sqrt{2}}{2}\right)\right)\cdot\cos\left(\frac{\sqrt{2}\cdot t}{2}\right)+\left(\frac{2\cdot\sqrt{6}}{5}-\frac{2}{5}\right)\cdot\sin\left(t\cdot\left(\sqrt{3}+\frac{\sqrt{2}}{2}\right)\right)\cdot\sin\left(\frac{\sqrt{2}\cdot t}{2}\right)+\frac{4\cdot\cos\left(\frac{\sqrt{2}\cdot t}{2}\right)}{5}$$

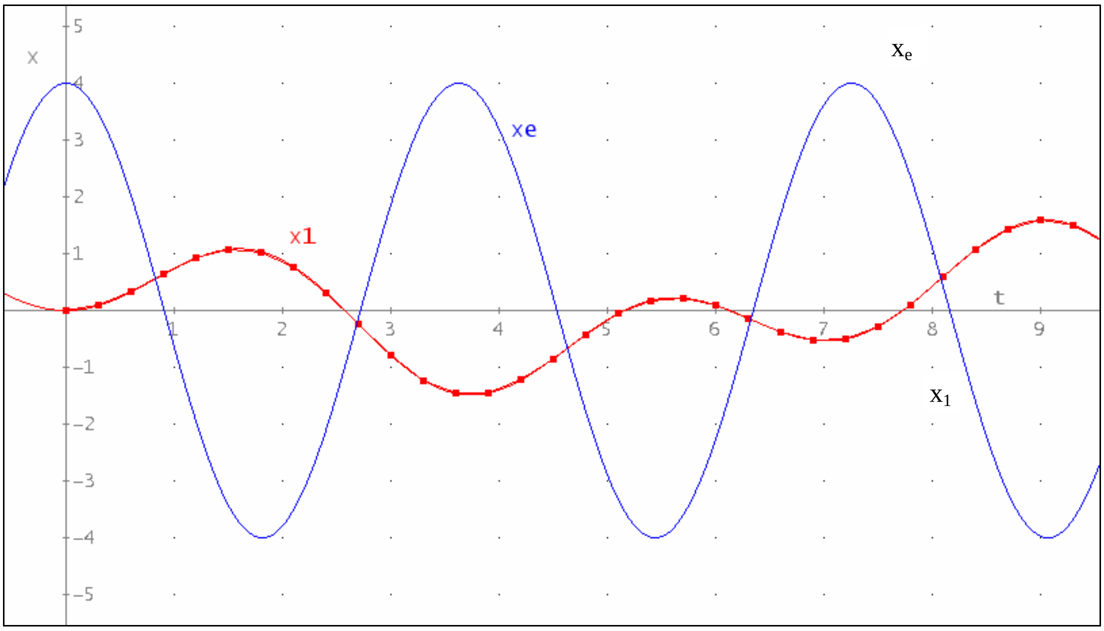
Oder man wendet nach Substitution $\dot{x}_1 = x_3$ die Runge-Kutta-Methode an:

```
#4:    (RK([x3, - 0.5·x1 + 2·COS(√3·t)], [t, x1, x3], [0, 0, 0], 0.3, 50))↓↓[1, 2]

#5:    4·COS(√3·t)
```

Das Ergebnis zeigt Figur 2 mit einer erheblichen Schwingungsantwort des Fundaments.

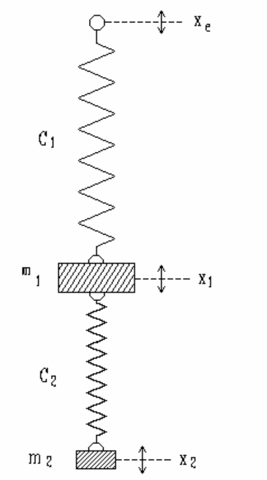
We can find an exact solution using DSOLVE2() or an approximative solution using the Runge-Kutta-Method. The result in figure 2 shows a serious vibration response of the basement.

We can achieve an interesting extinction of the resulting oscillation by connecting an additional mass m_2 using a second spring (spring constant c_2). See figure 3.



Figur 2

Eine interessante Auslöschung der resultierenden Schwingung kann man durch Ankopplung einer weiteren Masse m_2 über eine zweite Feder mit der Federkonstanten c_2 erreichen. Vgl. die Prinzipskizze Figur 3.



Figur 3

Mit den zusätzlichen Daten $m_2 = 1$, $c_2 = 3$, die so gewählt sind, dass sich als Eigenfrequenz die Frequenz des Störers ergibt, erhält man das System (unter Berücksichtigung richtiger Vorzeichen nach dem Prinzip actio = reactio)

The additional data ($m_2 = 1$, $c_2 = 3$) are chosen in such a way that the proper frequency results from the frequency of the disturber. (principle of actio = reactio).

Then we use again Runge-Kutta.

$$\ddot{x}_1 = -0.5(x_1 - x_e) - 1.5(x_1 - x_2)$$

$$\ddot{x}_2 = -3(x_2 - x_1)$$

oder

$$\ddot{x}_1 + 2x_1 - 1.5x_2 = 2\cos(\sqrt{3}\,t)$$

$$\ddot{x}_2 - 3x_1 + 3x_2 = 0$$

Mit den Substitutionen $\dot{x}_1 = x_3$ und $\dot{x}_2 = x_4$ kann man Runge-Kutta ansetzen.

```
#6 :  (RK([x3, x4, - 2·x1 + 1.5·x2 + 2·COS(√3·t), 3·x1 - 3·x2],
                 [t, x1, x2, x3, x4], [0, 0, a, 0, 0], 0.3, 50))↓↓[1, 2]
```

First of all we have to find the initial values for a (that is the constant amplitude of the connected oscillation). We plot for two values: a_1 = -2/3 and a_2 = -4/3 (negative, because opposite to the spurious oscillation). Using a_1 we recognize a remaining vibration whereas using a2 the effect of the additional oscillator causes a complete extinction of the basement´s vibrations.

Zuvor muss der Anfangswert a (das ist bei vernachlässigter Dämpfung die konstante Amplitude der angekoppelten Schwingung) bestimmt werden. Wir plotten für zwei Werte a_1 = -2/3 und a_2 = -4/3 (negativ wegen der Gegenphasigkeit zur Störschwingung). Beim Wert a_1 bleibt, wie die Grafik verrät, eine Restschwingung des Fundaments übrig. Beim Wert a_2 ist die Gegenwirkung des Zusatzschwingers so groß, dass sich eine vollständige Auslöschung der fremderregten Schwingung des Maschinenfundaments ergibt!

x1 for a1 = -2/3

$$\#7: \left(RK\left([x3, x4, -2\cdot x1 + 1.5\cdot x2 + 2\cdot COS(\sqrt{3}\cdot t), 3\cdot x1 - 3\cdot x2], [t, x1, x2, x3, x4], \left[0, 0, -\frac{2}{3}, 0, 0\right], 0.3, 50\right)\right)\downarrow\downarrow[1, 2]$$

x1 for a2 = -4/3

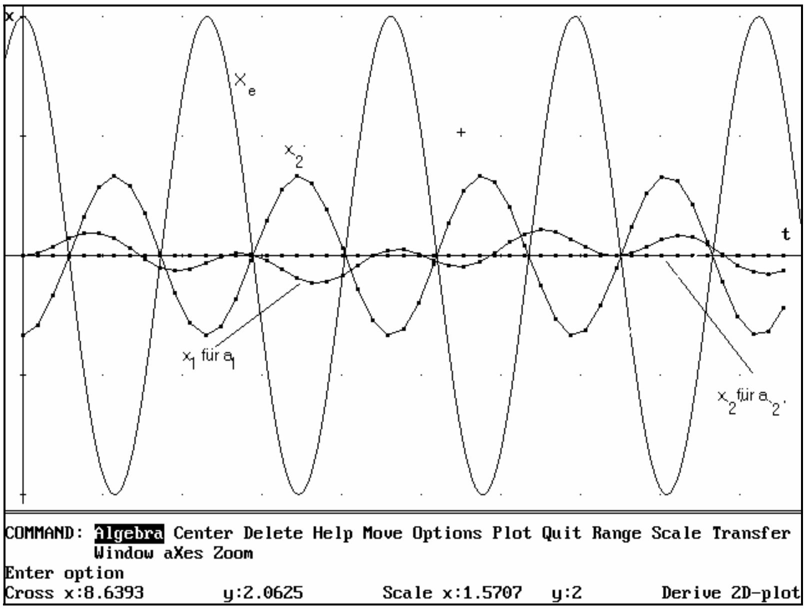
$$\#8: \left(RK\left([x3, x4, -2\cdot x1 + 1.5\cdot x2 + 2\cdot COS(\sqrt{3}\cdot t), 3\cdot x1 - 3\cdot x2], [t, x1, x2, x3, x4], \left[0, 0, -\frac{4}{3}, 0, 0\right], 0.3, 50\right)\right)\downarrow\downarrow[1, 2]$$

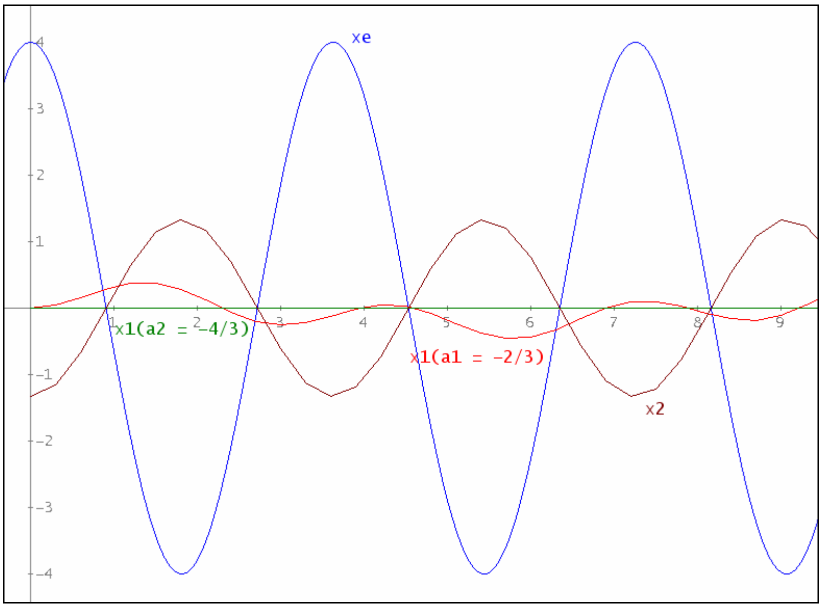
x2

$$\#9: \left(RK\left([x3, x4, -2\cdot x1 + 1.5\cdot x2 + 2\cdot COS(\sqrt{3}\cdot t), 3\cdot x1 - 3\cdot x2], [t, x1, x2, x3, x4], \left[0, 0, -\frac{4}{3}, 0, 0\right], 0.3, 50\right)\right)\downarrow\downarrow[1, 3]$$

Natürlich ist die Bestimmung von a nicht nur durch Plotexperimente, sondern auch rechnerisch leicht möglich. Man unterstellt naheliegend $x_1 = X_1 \cos(\sqrt{3}\,t)$ bzw. $x_2 = X_2 \cos(\sqrt{3}\,t)$ als partikuläre Lösung und berechnet für $X_1 = 0$ für alle t

$$X_2 = a_2 = -4/3.$$





Figur 4

Für die notwendigen zweiten Ableitungen und algebraischen Umformungen hilft *DERIVE*, Rechenfehler zu vermeiden. Übrigens ist auch ein Realexperiment ohne sonderlichen Aufwand für Liebhaber oder im Rahmen eines Schülerprojektes ohne weiteres möglich.

Ein elektrisches Analogon besteht in einem Antennen-Schwingkreis mit (mehreren) von Sendern fremderregten Schwingungen, darunter vielleicht eine besonders starke eines Lokalsenders. Wenn diese unerwünscht ist, kann man sie "heraussaugen" durch Ankopplung eines abgestimmten zweiten Schwingungskreises mit eventuell verstärkter Frequenz des Lokalsenders, sodass das Frequenzgemisch des Restes ungestört in der üblichen Weise weiterverarbeitet werden kann. Man kann versuchen, auch diese Situation zu simulieren.

An electronic analogon is an antenna oscillation circuit with (some) undesirable frequency(ies) - local station. You could "suck it out" by connecting a second circuit with possibly amplified frequency. You could try to simulate this situation.

Probability Distributions Proof and Computations (2)

Peter Mitic, Medstead, UK

MEANS AND VARIANCES OF CONTINUOUS RANDOM VARIABLES

Similar problems are encountered with continuous distributions, for which the moment generating function is $M(t) = \int_s e^{xt} f(x)\,dx$ where *f(x)* is a probability density function. When calculating the mean and variance of some continuous random variables it appears that *DERIVE* is more successful in evaluating integrals directly rather than via a moment generating function. The *DERIVE* session below shows that integrals which do not involve the parameter t can be evaluated easily. Integrals which contain this parameter cannot always be evaluated, even if the range of the parameter is restricted such that the integral is convergent (t < 1/2 in the expressions # 5 –.# 7 below). In particular, it is easy to check that the area under the graph of a probability density function is 1 (expressions # 2 to # 4 below). The probability density function used (F(x) in expression # 2) is that of a $\chi^2(5)$ random variable.

#1: $F(x) := \dfrac{\sqrt{2}\cdot e^{-x/2}\cdot x^{3/2}}{6\cdot\sqrt{\pi}}$

#2: Check that the area under the graph is 1

#3: $\int_0^{\infty} F(x)\ dx = 1$

#4: $M(t) := \int(F(x)\cdot e^{x\cdot t},\ 0,\ \infty)$

#5: $M(t) := \int\left(\dfrac{\sqrt{2}\cdot e^{x\cdot(t-1/2)}\cdot x^{3/2}}{6\cdot\sqrt{\pi}},\ 0,\ \infty\right)$

#6: t :∈ Real (−∞, 1/2)

#7: $M(t) = \int\left(\dfrac{\sqrt{2}\cdot e^{x\cdot(t-1/2)}\cdot x^{3/2}}{6\cdot\sqrt{\pi}},\ 0,\ \infty\right)$

#8: Failed! Calculate mean and variance directly!

#9: $\text{mean} := \int_0^\infty x\cdot F(x)\,dx$

#10: mean := 5

#11: $\text{variance} := \int_0^\infty x^2\cdot F(x)\,dx - \text{mean}^2$

#12: variance := 10

The situation with respect to parameters is not at all consistent. In the case of a Normal(0,1) random variable, with probability density function $N(x) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}$, *DERIVE* can immediately simplify the expression for the moment generating function, `M(t):=INT(N(x)*#e^(x*t),x,-inf,inf)` to obtain the result `#e^(t^2/2)` with no apparent need to restrict the range of t. This result is required for a proof of the Central Limit Theorem. *(See the next DNL, Josef).*

We have already pointed out (Mitic and Thomas 1994) that problems arise when the computer fails to calculate a result, and that consequently, problems presented as student exercises must be chosen so that they are 'do-able'. This is the case here. In order to teach moment generating functions, one has to choose suitable probability density functions. We have found that probability density functions of the form $g(x)\,x^n,\ x > 0$ with n a positive integer work provided that n is explicit. General results may be conjectured by constructing a VECTOR of particular examples. In the context of the class of probability density functions $f(x,n) = \frac{x^n\, e^{-x}}{n!},\ x \in R, n \in Z^+$, the expression

```
VECTOR(INT(x^n #e^(-x)/n!, x, 0, inf), n, 0, 10)
```

can be used to check that the areas under the graphs of the first 11 members of this class are all 1.

We would prefer to see the general result (obtained using *MATHEMATICA* 2.2, which does not simplify the result to 1)

$$\int_0^\infty \frac{x^n\, e^{-x}}{n!}\,dx = \frac{\Gamma(1+n)}{n!}$$

into which we can substitute n = 0, 1, 2, ...

The need to consider particular examples in this way is regrettable, but it is unavoidable with *DERIVE*.

CALCULATING PROBABILITIES

Calculating probabilities presents less of a problem technically, and also reinforces concepts in a way that a statistical package cannot.

A frequent task with discrete distributions is to use the probability generating function, $G(t)$, to calculate probabilities by isolating a coefficient or coefficients of t in a series expansion of $G(t)$. This often causes computational problems which overshadow the point of the exercise. We illustrate (#14 - #16 below) how this task is very simple using *DERIVE*. The geometric distribution Geom(1/6) has the probability generating function given by #14. The coefficient of t^r in $G(t)$ is obtained by computing

$$\left.\frac{d^r\, G(t)}{dt^r}\right]_{t=0}.$$

No formal expansion of $G(t)$ is needed, so the student can concentrate on the precise meaning of terms in a probability generating function. If the substitution $t = 0$ is programmed, as below, it is necessary to use the limit construct shown, despite its technical incorrectness.

#14: $G(t) := \dfrac{t}{6 - 5\cdot t}$

#15: For obtaining P(X = 3):

#16: $\lim_{t\to 0} \dfrac{\left(\frac{d}{dt}\right)^3 G(t)}{3!} = \dfrac{25}{216}$

In the case of continuous random variables it is best to use a probability density function directly. The use of a direct integration in this way stresses the concept that a probability is represented by the area under the graph of a probability density function, whether or not the function concerned is integrable analytically or not.

Expressions #18 – #20 below illustrate the calculation of P(0 < X < 2) where $X \sim \chi^2(5)$.

#18: $\int_0^2 F(x)\, dx = \dfrac{e^{-1}\cdot(3\cdot\sqrt{\pi}\cdot e\cdot \mathrm{ERF}(1) - 10)}{3\cdot\sqrt{\pi}}$

#19: simplified above and the right side approximated below:

#20: $\int_0^2 F(x)\, dx = 0.1508549639$

CONCLUSION

The point of using a computer algebra system should really be that *any* problem can be tackled. This context has highlighted a class of integrals and series which, unfortunately, cannot be tackled successfully with *DERIVE*. However, given certain general results, *DERIVE* can be used to compute means, variances, moments, probabilities etc without algebraic difficulties. Furthermore, all of these can involve some symbolic parameters, so that functional relations can be explored. This is not possible with a package which can only manipulate numbers. The shortcomings of *DERIVE* can be turned to advantage if the student is constrained to study relevant techniques and concepts, and hence to do some parts of the computation by hand.

REFERENCES

Etchells, T (1992), Investigating Probability Distributions with *DERIVE*. In Teaching Mathematics with *DERIVE*, Proceedings of the International School on the Didactics of Computer Algebra, Krems (Chartwell-Bratt)

Mitic, P and Thomas, P (1994); Limitations and Pitfalls of Computer Algebra. In Computers and Education 22, 4. Pergamon.

(Peter has forwarded two more papers of interest: Exploiting New Features in DERIVE3: Multiple Decisions and Whole Structure Programming and The Normal Distribution: Two Proofs and a Simulation. Josef)

Kurt Schmidt´s *MATHEMATICA-snail* from 1997

A Toast Celebrating the 25th Issue of the *DERIVE®* Newsletter

Josef and Noor, we in the *DERIVE* community are all so very grateful for your inspiration to initiate the independent DERIVE User Group and its bulletin; and for your talent and dedication in making it work so well.

Many of us are surely thinking "Can it really be 25 issues? How quickly time passes when we are having fun!" That is surely the secret of teaching mathematics – make it fun.

I have taken the occasion to review the previous issues, and I am awed by the total volume of great unique and still-relevant ideas in those issues. They are collector's treasures, and I cherish each and every one.

We all look forward to each issue with great anticipation. It is astounding to learn such ingenious and effective ways that others have used this technology.

So this toast of appreciation is not only to Josef and Noor, but also to all of the past and future authors. Your ideas are enriching mathematics for students, educators, and mathematical hobbyists all over the world.

Aloha and Mahalo,

David Stoutemyer
Chairman of the Board
Soft Warehouse, Inc.

DERIVE
25
TI-92

The 25th issue of the DERIVE Newsletter.

This is a wonderful moment: The 25th DNL. This means that I have reveived 24 issues before. There are 4 issues a year. 24 divided by 4 gives ...wait a moment ... I can do this without DERIVE ... just another second ... 6 ... I think this gives 6. What were we talking about? Oh yes. 6 years. Yes, it is true: The DERIVE User Group serves a growing community of DERIVE users and DERIVE enthusiasts for more than 6 years now.

The DNL is the major vehicle for communication among DUG members. And it serves its purpose very well. It is informative, challenging, and critical. The DNL is a valuable source of hints for the newcomers, it is a highly appreciated source of know-how for the experts, and it is a very serious and important means of feedback for the authors of DERIVE. I have seen both David Stoutemyer and Albert Rich sitting in their office in Honolulu studying the newest issue of the DNL very carefully - sometimes with deep wrinkles on their foreheads.

I cannot remember a single one of the 24 issues having been published late. Knowing Josef personally for more than 8 years now this is no surprise. He is one of the most diligent and reliable persons I ever met.

Josef and his wonderful wife Noor are absolutely charming people. If you haven't met them yet personally, come to one of the various international conferences (Josef lists them in the DNL) and you will make friends with them very quickly. There is no way not to become very friendly with them.

DERIVE and the DERIVE User Group. The first would not be the same without the second.

Josef and Noor, thank you!

Bernhard Kutzler
Managing Director of Soft Warehouse Europe

"A Day in the Life ....."

DERIVE as a Demonstration Tool in Upper Secondary Mathematics

Neil Bibby, Lancaster, UK

During recent years I have developed an integrated use of the *DERIVE* package into my work at all levels of upper secondary mathematics. This has been primarily as a demonstration tool, by which I mean that the students have not had direct hands-on use of the package. There are teo good reasons for this:

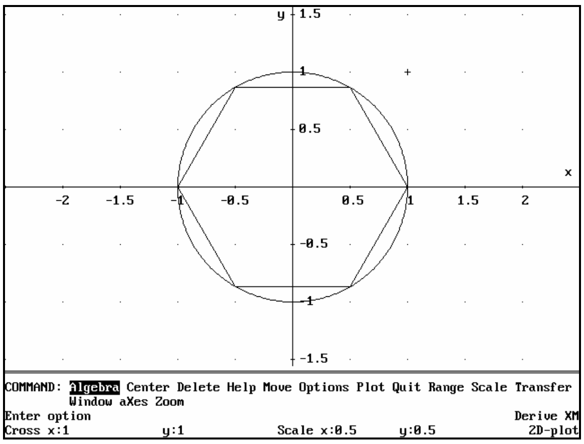
(1) *DERIVE* is a sophisticated piece of software which requires a thorough knowledge of mathematics on the part of the user, including a strong grasp of notation and notational variants. Although there is some evidence that it has been used successfully as a hands-on tool for students, equally other work has cast doubt on this. I think of *DERIVE* as a professional tool which in professional hands can provide a rich learning environment.

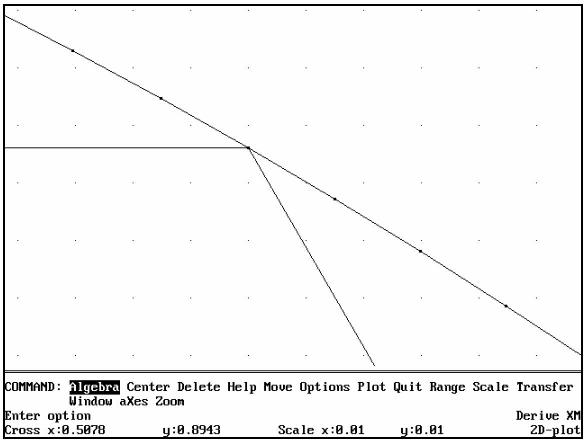
(2) The availability of graphic calulators enables students to have their own graphics system in the palm of their hand: so my preferred way of working was to integrate the use of the two where this was possible. The students can create their own version of the *DERIVE* screen display to reinforce what they have seen on the *DERIVE* display.

It will become apparent that my examples for the most part use the graphic capabilities of *DERIVE*. I do not claim that *DERIVE* is uniquely suited to this: no doubt much of what I present is equally implementable on other graphics systems. However the graphics system of *DERIVE* is flexible and easy to use once one has mastered a few basic techniques.

Example 1: π and Van Ceulen´s algorithm
(British Year 9 = Grade 8)

It is surprising that the treatment of the topic of π is so little changed by the availability of microtechnology. Apart from encounters with recurring decimals, it is the only place in elementary mathematics where the infinite process has to be really faced up to. Now that we have the computational tools there seems little to stop us from doing just that. The Archimedian approach of starting with a regular hexagon of unit radius and then doubling the number of sides is now eminently feasible on *DERIVE*. It´s easy to draw regular *n*-gons as the following diagram shows.

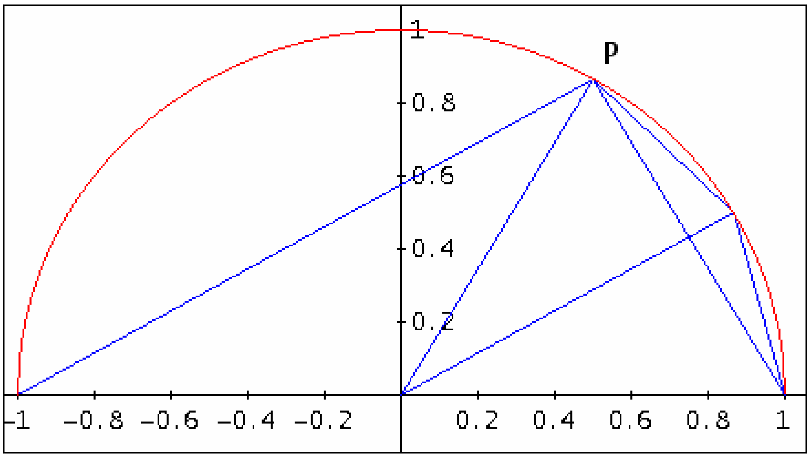




The screen shots above are from DERIVE for DOS from 1997.

In fact the left diagram shows a hexagon and a 360-gon!! The zoom facility of *DERIVE* allows us to easily see this in the right diagram.

But how can we calculate π? Ludolphus van Ceulen (born late 16th century in The Netherlands) devised a simple iterative method based on the Archimedian approach - to this day π continues to be known as the 'Ludolphine number - die Ludolph´sche Zahl' in German speaking Europe! By applying Pythagoras to two triangles in the following figure, we can simply relate the length s_{i+1} of the side of a regular 2*n*-gon to the length s_i of the side of a regular *n*-gon. The relation is that $s_{i+1} = \sqrt{2 - \sqrt{4 - s_i^2}}$. Starting with a hexagon of unit side, i.e. $n = 6$ and $s_0 = 1$, we can approximate π as the semi-perimeter of the polygon by $3 \cdot 2^i \cdot s_i$. The ITERATES and ITERATE functions of *DERIVE* make this very easy to do, and we can easily adjust the accuracy of the calculation to avoid cumulative rounding errors occuring.



#11: $\text{VECTOR}\left((\text{ITERATES}(\sqrt{(2 - \sqrt{(4 - s^2)})}, s, 1, 10))_k \cdot 3 \cdot 2^{k-1}, k, 1, 11\right)$

#12: [3, 3.1, 3.13, 3.13, 3.14, 3.14, 3.14, 3.14, 3.14, 3.14, 3.14]

#13: PrecisionDigits := 20

#14: NotationDigits := 20

#15: $\text{VECTOR}\left((\text{ITERATES}(\sqrt{(2 - \sqrt{(4 - s^2)})}, s, 1, 15))_k \cdot 3 \cdot 2^{k-1}, k, 1, 16\right)$

#16: [3, 3.1058285412302491481, 3.1326286132812381971, 3.1393502030468672071,
3.1410319508905096377, 3.1414524722854620752, 3.1415576079118576454, 3.141583892148318406,
3.1415904632280500 57, 3.1415921059992713602, 3.1415925166921566904, 3.1415926193653861406,
3.1415926450336881605, 3.1415926514508704393, 3.1415926530552116196, 3.1415926534558914124]

#17: π

#18: 3.1415926535897932384

Example 2: Population growth and exponential modelling
(British Year 11 = Grade 10)

This work formed the basis of some GCSE coursework. We started with the population data for the major Italian cities as given in the following table:

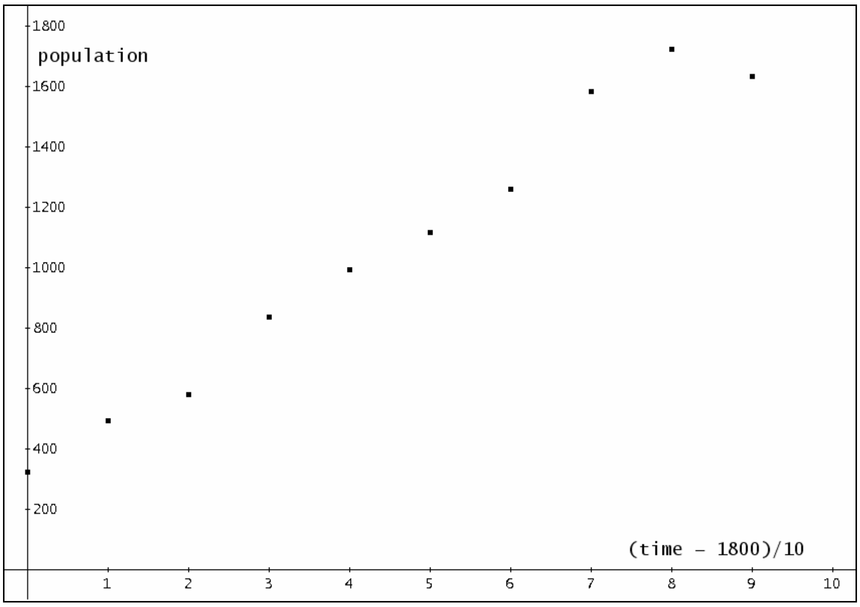
Table 6 *Population of the main Italian citi : (in thousands) 1800–1980*

	1800/1	1850/1	1860/1	1870/1	1880/1	1900/1	1910/1	920/1	1930/1	1940/1	1950/1	1960/1	1970/1	1980/1
Genoa	91	120	129	130	180	235	272	316	608	635	648	784	812	760
Turin	78	135	178	208	254	336	427	502	597	629	711	1026	1178	1104
Milan	135	242	242	262	322	493	579	836	992	1116	1260	1583	1724	1635
Rome	163	175	184	244	300	463	542	692	1008	1156	1652	2188	2800	2831
Naples	427	449	417	449	494	564	723	/22	839	866	1011	1183	1233	1211
Palermo	139	180	186	219	245	310	342	394	390	412	491	588	651	700

Note the continued growth of cities in the inter-war years (despite fascist attempts to encourage 'rurality') and the surge in the period 1950–70. Naples was Italy's largest city until the First World War.

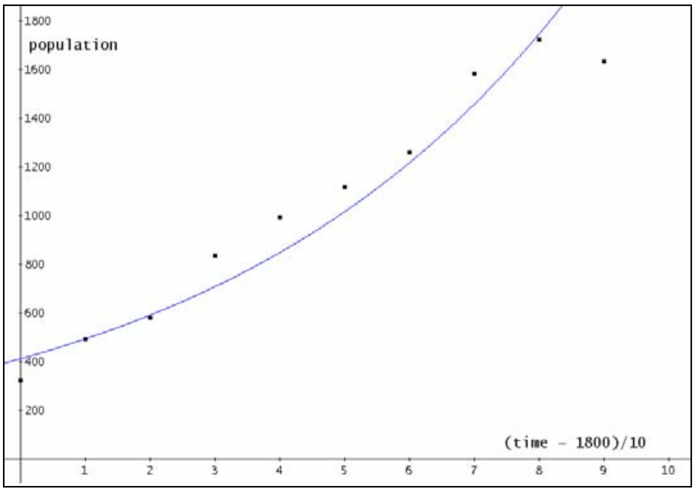
Source: B. R. Mitchell, *International Historical Statistics, Europe 1750–1988* (N.Y., 1992).

We decided to concentrate on Milan from 1880 to 1980 (not necessarily the easiest choice - why?) the data can be plotted as follows:



The problem now was to try to fit a curve of the form $y = a \cdot b^x$ to this data, with y as the population in thousands and x as the number of decades from 1880. Initial suggestions usually were a = 322 and for the choice of b to give the visual best fit to the other points. Since the doubling-time appeared from the table to be about 4 decades, by the rule-of-70 this gave a growth-rate of about 18% (since $70/4 \approx 18$), and hence $b \approx 1.18$.

Further discussion led to the method of visual-best fit being questioned, and the TI-82´s statistical **ExpReg** function being used to provide a least squares fit. I hasten to add that the intricacies of the least-squares method were not discussed: the TI-82 function was therefore treated as reliable and trusted 'blackbox' which would give an appropriate result. (*DERIVE´s* FIT function could also have been used for this purpose). The values obtained were a = 412.5, b = 1.198 (4 s.f.) and plotting $y = a \cdot b^x$ with these values gave the following result:



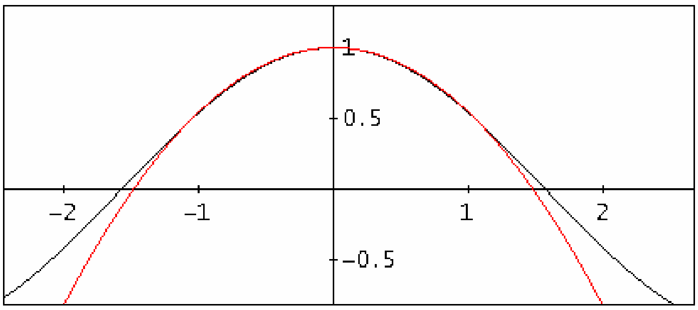

```
                          ⎡ 0    LN(322) ⎤
                          ⎢ 1    LN(493) ⎥
                          ⎢ 2    LN(579) ⎥
                          ⎢ 3    LN(836) ⎥
#25:  FIT ⎛[x, a·x + b],  ⎢ 4    LN(992) ⎥⎞
                          ⎢ 5   LN(1116) ⎥
                          ⎢ 6   LN(1260) ⎥
                          ⎢ 7   LN(1583) ⎥
                          ⎢ 8   LN(1724) ⎥
                          ⎣ 9   LN(1635) ⎦

#26:  0.180389·x + 6.02224

       0.180389·x + 6.02224
#27:  e
```

The visual immediacy of this treatment, combined with the use of TI-82 (albeit in a 'blackbox' mode) proved pedagogically.

Example 3: Simpson´s Rule
(British Year 12 = Grade 11)

My preference with this algorithm is to emphasise the fitting of a quadratic function by three points of the graph of the function whose integral is required. A nice example to start with is INT(cos(x), x, -π/3, π/3). The interpolating quadratic is easily seen to be $y = 1 - \frac{9x^2}{2\pi^2}$.

The graph of this gives the following, together with that of cosine; omitting here but also worth including, is the graph of the second degree Taylor approximation of cosine at 0, to emphasise that this approximates better at 0, but doesn´t interpolate at all.

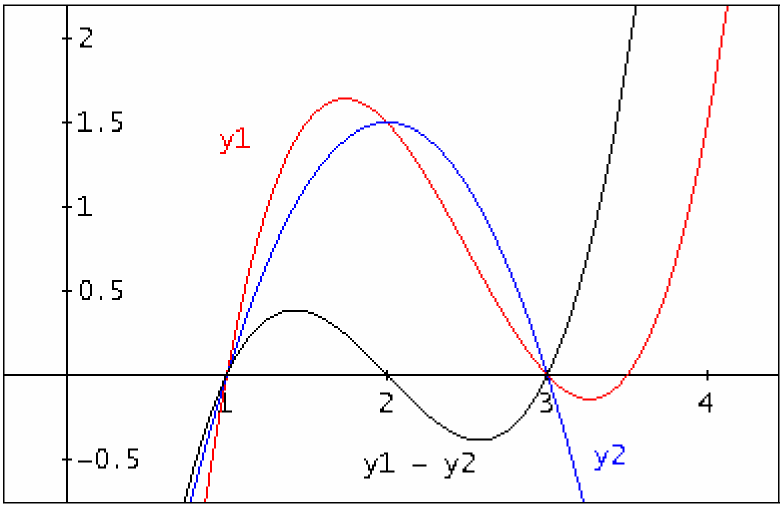


A comparison of the corresponding integrals is then illuminating: the exact value is √3 (1.732...), the Simpson value is 1.745.... and the 'Taylor' value is 1.711....

Students should appreciate that Simpson´s Rule (actually due to Newton originally - who else?) is exact for quadratics: this seems obvious, but a few corroborative calculations using *DERIVE* push home the point. However one of the most intriguing features of Simpson´s Rule is that it is exact for cubics!! This 'two-for-the-price-of-one' aspect needs some explanation. A formal proof is possible of course, but an informal graphical approach might be appropriate and convincing initially.

The diagram shows the graphs of a cubic, its quadratic interpolator, and the difference of the two. This difference is a new cubic, whose zeros must be a, $a + h$, and b (= $a + 2h$).

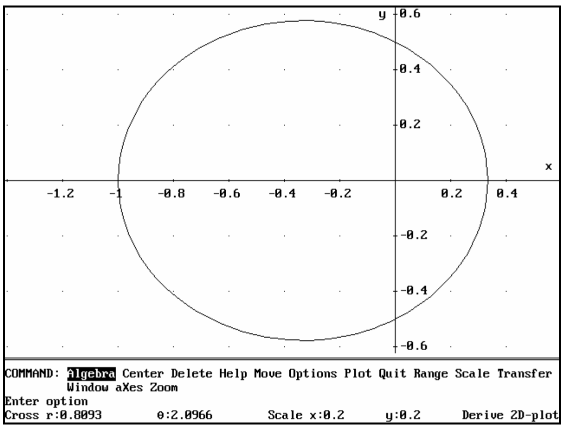
So the point where $x = a + h$ must be a point of inflexion, i.e. the symmetry point of the graph. Hence the two 'lobes' of this cubic are of equal area, and thus Simpson´s Rule is exact!! The visual plausibility of this is compelling, but should not detract from the details of the argument; once again the two aspects corroborate each other.



Example 4: Polar and Cartesian Coordinates
(British Year 13 = Grade 12)

DERIVE version 3 has the facility to plot implicitly. This is naturally slower than the explicit plotting, but can be used to good effect, especially when explicit plotting is simply not possible (e.g. $y^5 + y = x$). It is especially effective when the equivalence of polar forms and cartesian (rectangular) forms of an equation are being considered.

One approach to the conic sections is to use the focus-directrix definitions to arrive at $r = \dfrac{a}{1 + e\cdot\cos\theta}$. (It is instructive to consider the limaçons $r = a(1 + e\cdot\cos\theta)$ first. As an example we consider $r = \dfrac{1}{2 + \cos\theta}$, where $a = e = 0.5$. This gives an ellipse (or hypObola, as I sometimes call them!), and is easily plotted using *DERIVE:*



It is now instructive to ask "what is the *cartesian* equation of this curve?" We use the substitutions $r=\sqrt{x^2+y^2}$ and $\cos\theta=\frac{x}{\sqrt{x^2+y^2}}$ to obtain the cartesian equivalent $\frac{9}{4}(x+\frac{1}{3})^2+3y^2=1$. The static graph above cannot do justice to the dynamic effect of seeing this plotted (remember to switch back to rectangulars!) in a new colour exactly on top of the polar verson. The monochrome calculator displays cannot compete here, even if they could plot implicitly! The cartesian form provides us with new information: we can see from the equation that $x=-\frac{1}{3}$ is a line of symmetry, and that the lines $y=\pm\frac{1}{\sqrt{3}}$ are tangent lines: in the follwing diagram we confirm these:

#60: $\sqrt{x^2+y^2}=\dfrac{1}{2+\dfrac{x}{\sqrt{x^2+y^2}}}$

#61: $\sqrt{x^2+y^2}=\dfrac{\sqrt{x^2+y^2}}{2\cdot\sqrt{x^2+y^2}+x}$

#62: $\left(\sqrt{x^2+y^2}=\dfrac{\sqrt{x^2+y^2}}{2\cdot\sqrt{x^2+y^2}+x}\right)\cdot(2\cdot\sqrt{x^2+y^2}+x)$

#63: $x\cdot\sqrt{x^2+y^2}+2\cdot x^2+2\cdot y^2=\sqrt{x^2+y^2}$

#64: $(x\cdot\sqrt{x^2+y^2}+2\cdot x^2+2\cdot y^2=\sqrt{x^2+y^2})-x\cdot\sqrt{x^2+y^2}$

#65: $2\cdot x^2+2\cdot y^2=(1-x)\cdot\sqrt{x^2+y^2}$

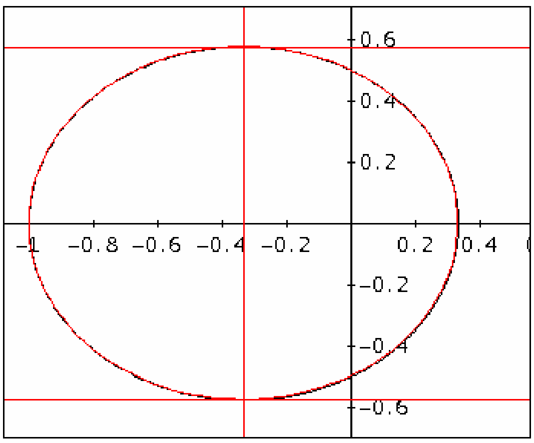
#66: $(2\cdot x^2+2\cdot y^2=(1-x)\cdot\sqrt{x^2+y^2})^2$

#67: $4\cdot(x^2+y^2)^2=(x-1)^2\cdot(x^2+y^2)$

#68: $\dfrac{4\cdot(x^2+y^2)^2=(x-1)^2\cdot(x^2+y^2)}{x^2+y^2}$

#69: $4\cdot(x^2+y^2)=(x-1)^2$

#70: $\dfrac{9}{4}\cdot\left(x+\dfrac{1}{3}\right)^2+3\cdot y^2=1$



I hope that these examples will convey the favour of this approach. I now find *DERIVE* an indispensible tool in my day-to-day teaching.

Please don´t hesitate to contact me if you need further advice or assistance.

Neil Bibby
Department of Mathematics
University College of St Martin
Lancaster LA1 3JD, UK

DERITANS

TANGRAMS with DERIVE

Alfonso J. Población, Valladolid, Spain

INTRODUCTION

One of the oldest parts of Recreational Mathematics concerns Dissection problems. Among them, the ones derived from seven pieces (called tans) known as tangrams are very popular. (see figure 1). Arranging these tans it is possible to form a wide variety of figures. The rules are simple: the seven tans must be used and it is not allowed to overlap any of them, although we can let holes among the pieces. Martin Gardener in [1] classifies in three types the questions we can deal with tangrams:

1. to find how to build a given tangram or to prove its impossibility
2. to find how to form several real figures such as animals, objects or persons in the most artistic or amazing way
3. to solve problems about Combinatorial Geometry that the seven tans set forth.

In the mentioned reference we can find more about these categories of problems and the history of tangrams.

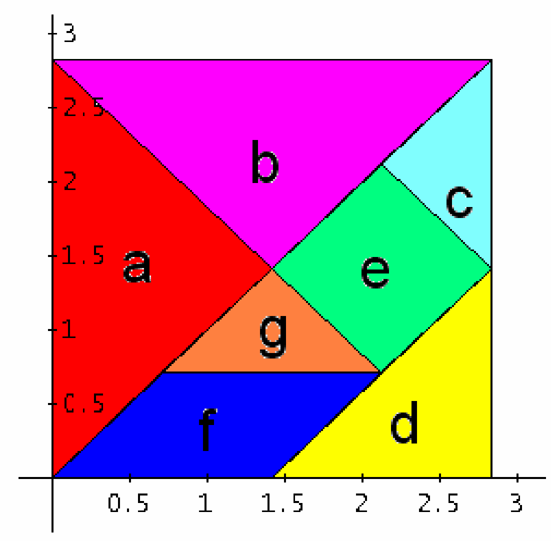
HOW TO USE *DERIVE*

We will try to play with the tans using *DERIVE* and our imagination. First of all, we will define the seven pieces (lines#3 and #4) from its original place in the initial square.

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

#2:  The seven tans
```

#3: $$\left[a := \begin{bmatrix} 0 & 0 \\ \sqrt{2} & \sqrt{2} \\ 0 & 2\cdot\sqrt{2} \\ 0 & 0 \end{bmatrix},\ b := \begin{bmatrix} 0 & 2\cdot\sqrt{2} \\ \sqrt{2} & \sqrt{2} \\ 2\cdot\sqrt{2} & 2\cdot\sqrt{2} \\ 0 & 2\cdot\sqrt{2} \end{bmatrix},\ c := \begin{bmatrix} 2\cdot\sqrt{2} & \sqrt{2} \\ 2\cdot\sqrt{2} & 2\cdot\sqrt{2} \\ \frac{3\cdot\sqrt{2}}{2} & \frac{3\cdot\sqrt{2}}{2} \\ 2\cdot\sqrt{2} & \sqrt{2} \end{bmatrix}\right]$$

#4: $$\left[d := \begin{bmatrix} \sqrt{2} & 0 \\ 2\cdot\sqrt{2} & 0 \\ 2\cdot\sqrt{2} & \sqrt{2} \\ \sqrt{2} & 0 \end{bmatrix},\ e := \begin{bmatrix} \frac{3\cdot\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 2\cdot\sqrt{2} & \sqrt{2} \\ \frac{3\cdot\sqrt{2}}{2} & \frac{3\cdot\sqrt{2}}{2} \\ \sqrt{2} & \sqrt{2} \\ \frac{3\cdot\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix},\ f := \begin{bmatrix} 0 & 0 \\ \sqrt{2} & 0 \\ \frac{3\cdot\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 0 & 0 \end{bmatrix},\ g := \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{3\cdot\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \sqrt{2} & \sqrt{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}\right]$$



You might wonder about the colours with the *DERIVE* plot. This is easy to perform with *DERIVE 6*.

Transfer by applying Mark and Copy the figure into the Algebra Window. Mark the graph and activate "Convert Picture Object" in the Edit-Menu. Then you can switch to Paintbrush and treat the graph as you like.

(See also Tania Koller´s note in DNL#63, page 14).

#5: Two basic functions to play

#6: $T(x, y, \alpha, u, v) := [\cos(\alpha)\cdot x - \sin(\alpha)\cdot y + u, \sin(\alpha)\cdot x + \cos(\alpha)\cdot y + v]$

#7: $\text{MOVEMENT}(w, \alpha, u, v) := \text{VECTOR}(T(w_{i,1}, w_{i,2}, \alpha, u, v), i, \text{DIMENSION}(w))$

The first one moves a point (x.y) by a rotation of angle α followed by a translation of vector (u,v). The last applies a movement T to a whole *tan* w. As I said before we can add REFLECTIONS or any other affine motions we desire.

Now the goal is to find the movements needed to bring the tans from its initial coordinates to the new ones required to form a figure. Let me explain it with an example. We desire to sketch the Yacht in figure 2. After discovering where the tans must be put (this can also be tried using *DERIVE*), but it can make us spend a lot of time), we can find the movements by testing or calculate them exactly. I got the Yacht with these. (Simplify and plot the result in Connected Mode):

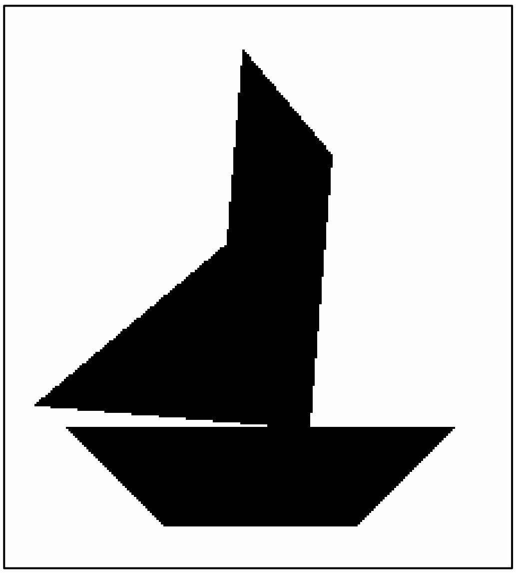
#8: $\text{MOVEMENT}\left(e, -\frac{\pi}{4}, 0, 0\right)$

#9: $\text{MOVEMENT}\left(c, -\frac{\pi}{4}, 0, 0\right)$

#10: $\text{MOVEMENT}\left(g, -\frac{\pi}{4}, 0, 0\right)$

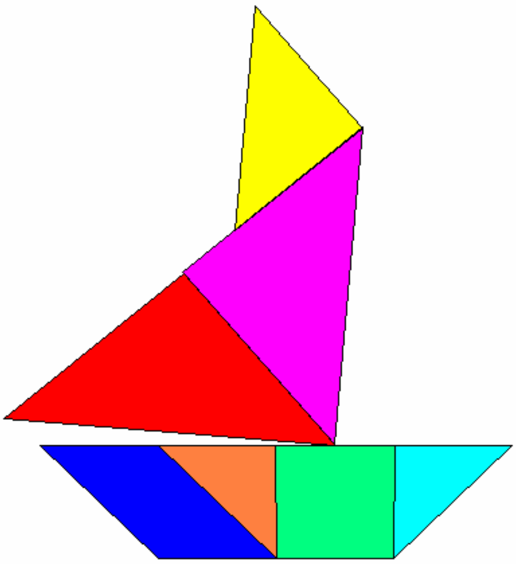
#11: $\text{MOVEMENT}\left(f, -\frac{\pi}{4}, 0, 0\right)$

#12: $\text{MOVEMENT}\left(a, \frac{\pi}{2} - \frac{1}{12}, 2.5, 0\right)$



$$\#13:\ \text{MOVEMENT}\left(b,\ \frac{3\cdot\pi}{2} - \frac{1}{12},\ -2\cdot\sqrt{2}\cdot\text{COS}\left(\frac{1}{12}\right) + 2\cdot\sqrt{2}\cdot\text{SIN}\left(\frac{1}{12}\right) + \frac{5}{2},\ 2\cdot\sqrt{2}\cdot\text{COS}\left(\frac{1}{12}\right) + 2\cdot\sqrt{2}\cdot\text{SIN}\left(\frac{1}{12}\right)\right)$$

$$\#14:\ \text{MOVEMENT}\left(d,\ \frac{\pi}{2} - \text{ATAN}\left(\frac{\sqrt{\left(2\cdot\left(\text{COS}\left(\frac{1}{12}\right) + \text{SIN}\left(\frac{1}{12}\right)\right)^2 + \left(\text{SIN}\left(\frac{1}{6}\right) + 1\right)\cdot\left(\text{SIN}\left(\frac{1}{6}\right) - 1\right)\right)}}{\sqrt{\left(\text{SIN}\left(\frac{1}{6}\right) + 1\right)\cdot\left(\text{COS}\left(\frac{1}{12}\right) - \text{SIN}\left(\frac{1}{12}\right)\right)}}\right),\ 2\cdot\sqrt{2}\cdot\text{SIN}\left(\frac{1}{12}\right) - \frac{\sqrt{2}\cdot\sqrt{\left(4\cdot\text{SIN}\left(\frac{1}{6}\right) - \text{COS}\left(\frac{1}{3}\right) + 3\right)}}{\sqrt{\left(\text{SIN}\left(\frac{1}{6}\right) + 1\right)}} + \frac{5}{2},\ (2\cdot\sqrt{2} - 2)\cdot\text{COS}\left(\frac{1}{12}\right) + 2\cdot\text{SIN}\left(\frac{1}{12}\right)\right)$$



Of course the last ones cannot be obtained at first sight. (The complete procedure can be found in file DERITANS.MTH.)

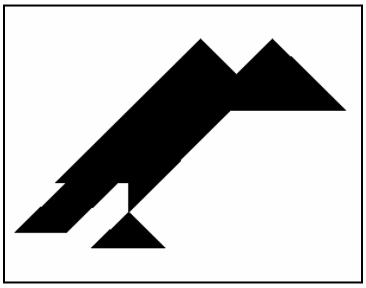
The next two functions can be helpful to find angles and distances:

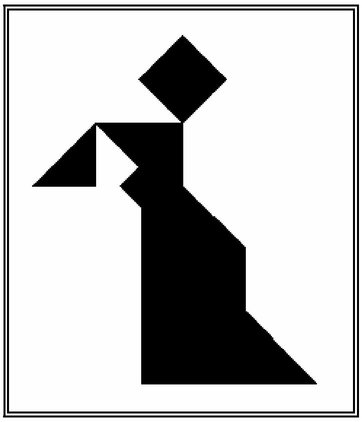
$$\text{LINE}(x\_,\ y\_) := \frac{x\_{}_2\cdot(y\_{}_1 - x) - y\_{}_2\cdot(x\_{}_1 - x)}{y\_{}_1 - x\_{}_1}$$

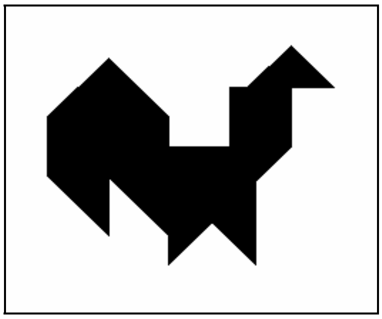
$$\text{DIST}(x\_,\ y\_) := \sqrt{\left((y\_{}_1 - x\_{}_1)^2 + (y\_{}_2 - x\_{}_2)^2\right)}$$

You can find much more about tans in the mentioned reference. To practise I propose you two different problems:

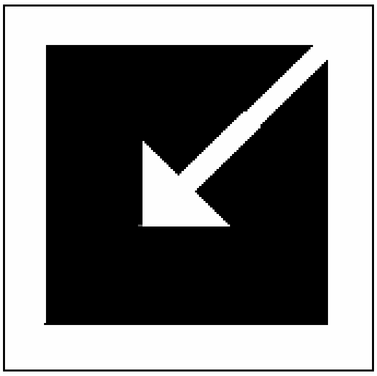
(1) Try to find the movements to compose these figures:







(2) What is the biggest area that a tangram can contain? (In [1], it is given a solution but as far as I know it is not still proved that this is the biggest one. This is an open problem).

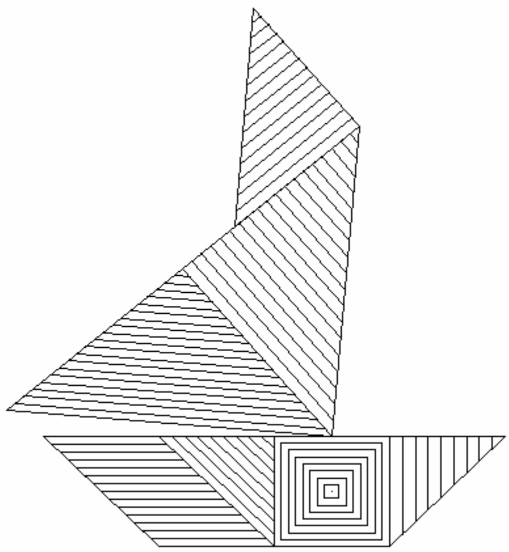


Finally, two crazy suggestions. It could be funny to play with *DERIVE* and the tans among several players trying to find the movements before the opponents. Or perhaps anyone could think about setting out exams in which the students are given a figure to find the movements needed to be formed. So our students would love and enjoy mathematics much more or perhaps they hate us even more. If someone dares to put it in practice, tell us how it goes.

REFERENCES

[1] Gardner, M. Viajes por el tiempo otras perplejidades matemáticas. Capitulos 3 y 4. Ed. Labor. S.A., 1988 (English version: Time travel and other mathematical bewilderments).

I was really fascinated by Alfonso´s article. It combines fun and creative phantasy with mathematics in a wonderful way. I imagined wooden tans to play with. So I tried to produce shaded tans - just for fun - to present the solutions or the problems in a nice form. That is the "wooden" Yacht. If you want to work with the shaded tans then use the **_new** *tans.*
(The complete solution for the Yacht will be found on the diskette of the year. I hope to be able to present the actual files to be downloaded on the homepage of my school in the near future.)
Josef



Al Rich´s interesting answer to Nigel Backhouse´s comparison of calculation times:

The following is in response to Nigel Backhouse comments dated 27 January 1997 to the DERIVE News mailing list concerning integer factoring of 2^101-1:

To factor integers, DERIVE uses trial division, followed by a Monte-Carlo Primality test, followed by the Pollard Rho algorithm as detailed in Knuth, The Art of Computer Programming, Volume 2. For the most part, these algorithms are implemented using loops rather than recursion. This combination of factoring methods is usually the most efficient until the size of the second largest prime factor exceeds about 12 digits.

Glossing over details, the more complicated elliptic curve method is then usually the most efficient until the size of the second largest prime factor exceeds about 25 digits. Quadratic residue methods are usually the most efficient beyond that.

The times Backhouse reports measure the relative speed of the algorithms used by the various computer algebra systems to factor 2^101-1, more than they measure the speed of the underlying arithmetic. To try to determine the underlying combination of algorithms being used by the various systems, construct a family of integers having n digits in its second largest prime, and then compare the times required to factor the family of integers by the various systems.

The relative rankings on different CPUs are likely to be the same, and more or less proportional to sequential memory fetch and store, so testing on additional CPUs is not likely to yield much additional information.

The bignum arithmetic in DERIVE uses the same nonrecursive algorithms (also described by Knuth) as your other systems, but DERIVE's is implemented in Intel 32-bit assembly language. Therefore, the DERIVE bignum arithmetic should be faster than bignum arithmetic compiled from languages such as C. To test this, compare times for multiplying some non-special numbers having 100, 200, 400, ... digits, and time the multiplication. Times for the smallest examples are likely to masked by the timer resolution unless done using ITERATE.

Here they are – at the occasion of Josef's and Josef's daughter's birthday (same day!!):
1st row: Constantin, Dominic, Yvonne, **Kim (the baby from page 1)**
2nd row: Moritz, Naomi and Maxime

Parametric Plots

In previous sections we saw that *DERIVE* will plot any vector of functions, say

$$[f_1, f_2, \ldots, f_n]$$

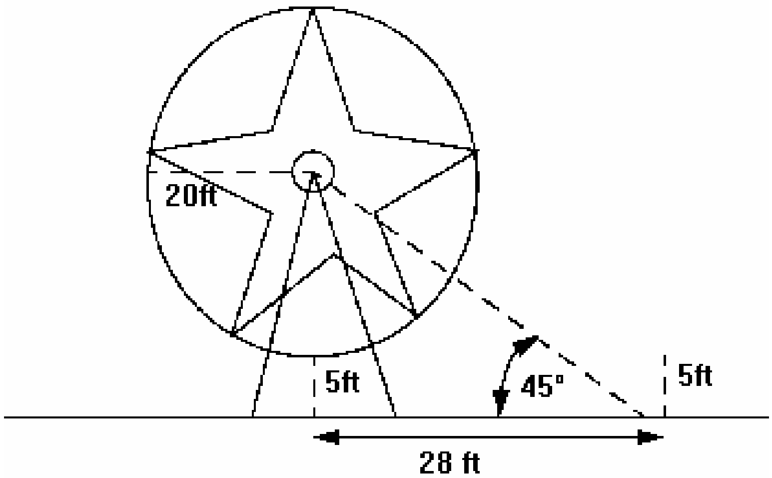
as *n* different plots on the two dimensional Cartesian Plane. There is an exception to this rule. It is when $n = 2$. In this case *DERIVE* regards the pair as the parametric desription and assumes:

$$x(t) = f_1(t)$$
$$y(t) = f_2(t)$$

for some parameter, *t*. We will use this feature to solve the following problem.

You and a fried decide to play cath with as baseball. To make things a little more interesting, your friend decides to ride a ferris wheel during the game. The seat of the ferris wheel is situated so that when a passenger boards it, the seat is five feet above the ground and it is 20 feet from the center of the wheel. The angular velocity of the wheel is 0.5 radians per second. You are standing 28 feet from the ground center of the wheel. You can throw the ball with a velocity of 64 feet per second and release it from 5 feet above the ground. At what angle and time after your friend is directly above the ground center of the wheel must you throw the ball so that it can be caught when your friend´s angle of elevation with the ground is 45°? You may assume that the horizontal component of the velocity of the ball remains constant.

Let´s draw a picture.



Newton´s Laws tell us that the height of the ball above the ground at time, *t*, is given by

$$Y(t) = Y_0 + v_{0Y}\, t - 16t^2$$

where

$Y_0 = 5$ ft. ; v_{0Y} = initial velocity in the *y* direction.

The vertical component of the velocity of the ball is: $v_Y = v_{0Y} - 32t$.

First we look at the path followed by your friend on the ferris wheel. Place a coordinate system at the ground center of the wheel. It is easy to figure that for your friend, the position *t* seconds after the wheel starts is:

$$x(t) = 20 \sin\left(\frac{t}{2}\right)$$

$$y(t) = 25 - 20 \cos\left(\frac{t}{2}\right)$$

From the picture we can see that your friend will be in position to catch the ball when $t = \pi/2$. We will draw your friend´s path from the time the ferris wheel starts until the ball is supposed to arrive.

In *DERIVE* author: `[20 sin(t/2), 25 – 20 cos(t/2)]`

To plot this enter the Plot Window and choose the Plot option. *DERIVE* will prompt you for the range of the parameter, *t* in this case, enter 0 for Min and π/2 for Max. An eighth of a circle will be drawn.

What about the ball? Let *P(t) and Y(t)* be the horizontal and vertical components of the position of the ball. Let θ be the angle that the ball is pitched (angle with the negative *x*-axis). Then

$$P(t) = 28 - 64\cos(\theta)\,t$$

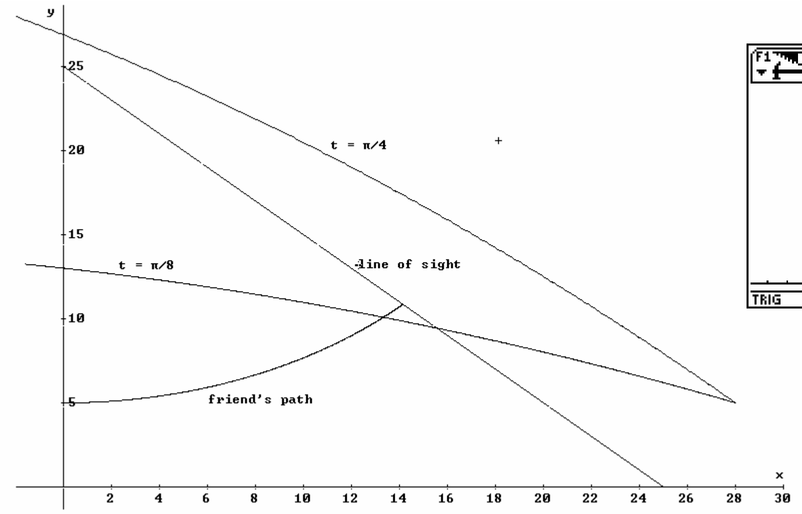
$$Y(t) = 5 + 64\sin(\theta)\,t \; - \; 16t^2$$

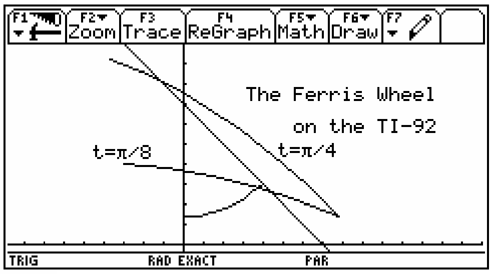
(A nice question for the non-English or non-American students: Where does the number 16 come from?? Josef)

We need to determine the choice of θ that will cause the curve determined by [*P(t), Y(t)*] to intersect the path of our friend when t = π/2. We will determine the time delay later. **A**uthor

```
[28 – 64•COS(θ)•t, 5 + 64•SIN(θ)•t – 16•t^2]
```

In *DERIVE* (for DOS) θ ist typed as Alt-h. We can not draw any graphs at this point. We must make a choice for either θ or *t*. This is the beauty of having a tool like *DERIVE*! We can experiment. Let´s make a naive choice, $\theta = \pi/4$. Use **M**anage/**S**ubstitute to draw, replace θ and then **P**lot the resulting expression for *t* going from 0 to 1. As you see on the graph below, this overshoots the mark and is too long a time. Now go back to the Algebra Window and use **M**anage/**S**ubstitute to replace θ in our original expression with π/8. This time when you **P**lot let *t* run from 0 to 0.7. As we see this undershoots the mark and is still too long a time.





You continue the investigation and find out the value for θ that will work for this problem. Hint: π/8 isn´t off by much. Once you have the correct angle, you can use the trace to determine how long it takes the ball to reach your friend´s path. Now you can decide how long to wait. If you want *DERIVE* to solve for *t* analytically, you can also do that once you have the value for θ. Go back to the **A**lgebra Window and substitute π/2 into the parametric equations for your friend´s position and then set *P(t)* with θ given as the value you found from your graphical exploration equal to the x-coordinate of your friend´s position.

This example shows that it is possible for students with only a knowledge of trigonometric relations to solve a very sophisticated problem using that knowledge and some good graphics. We grant that some analytical steps were bypassed and the answer may be an approximation to the "correct" answer, but was the student any less involved with the mathematics? We leave that question for your decide.

THE TI-92 CORNER

(EDITED BY B. WAITS, F. DEMANA, B. KUTZLER & J. BÖHM)

DERIVE and TI-92 do not always yield the complete set of solutions to systems of linear equations containing parameters

Karl-Heinz Keunecke, Kiel, Germany (kh@KeuKiel.NetzService.de)

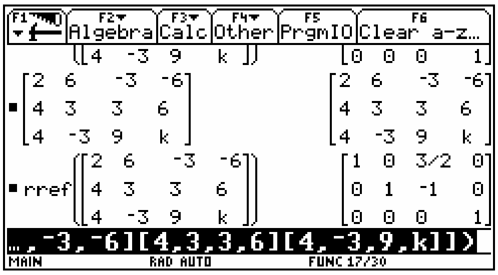
At school in the field of linear algebra it is often helpful to make use of CAS to solve systems of linear equations.

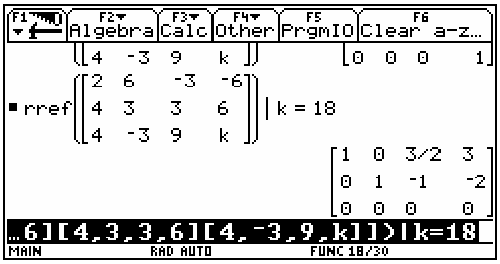
After the input of the augmented matrix students can calculate the row echelon form of the matrix by means of the DERIVE command Row_Reduce or with the command rref() of the TI-92 from which the solution of the linear equation can be read directly. This is shown in the following example.

$$\begin{array}{l}2x+y-3z=4\\ x+y+z=0\\ 3x-y-z=2\end{array} \Rightarrow \begin{pmatrix}2&1&-3&4\\1&-1&1&0\\3&-1&-1&2\end{pmatrix} \Rightarrow \begin{pmatrix}1&0&0&2\\0&1&0&3\\0&0&1&1\end{pmatrix} \Rightarrow \begin{array}{l}x=2\\ y=3\\ z=1\end{array}$$

This method can be applied to equations with parameters and is shown in the example below in which t is an arbitrary real number.

$$\begin{array}{l}3x-2y+z=2t\\ 5x-4y-z=2\\ x-3y-2z=2t+6\end{array} \Rightarrow \begin{pmatrix}3&-2&1&2t\\5&-4&-1&2\\1&3&-2&2t+6\end{pmatrix} \Rightarrow \begin{pmatrix}1&0&0&t+1\\0&1&0&t+1\\0&0&1&t-1\end{pmatrix} \Rightarrow \begin{array}{l}x=t+1\\ y=t+1\\ z=t-1\end{array}$$





However, if one selects the following equations with

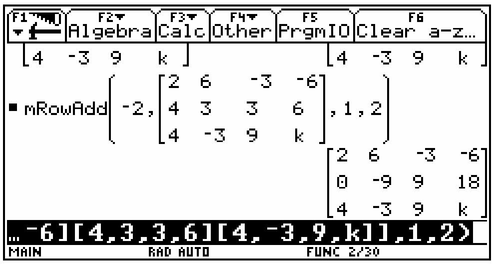
$$(1)\qquad \begin{array}{l}2x+6y-3z=-6\\ 4x+3y+3z=6\\ 4x-3y+9z=k\end{array}\ ,$$

then neither DERIVE nor the TI-92 give the correct (complete) solution. In the display the command rref() was applied to the augmented matrix of the system of equations. From the result it can be concluded that for all $k \in \mathrm{Re}$ the system of equations has no solution. The bottom row may be interpreted as $0x+0y+0z=1$ and gives accordingly a false statement.

This result is obviously incorrect. Because for when $k = 18$ you get a different solution as can be seen from the adjacent display of the screen. The solution is read as,

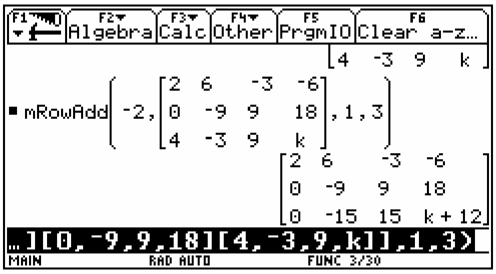
$k=18$: $x=-\dfrac{3}{2}z+3 \;\wedge\; y=z-2,\;\; z\in \mathrm{Re}$, where z is an arbitrary parameter.

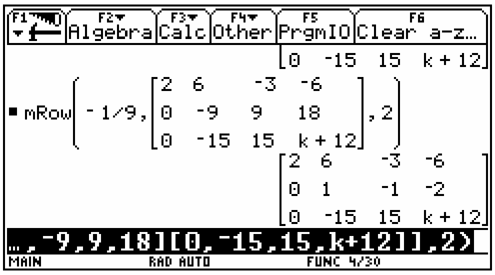
The question is then, why the solution for $k = 18$ is not recognized by the CAS. The cause is, of course, the algorithm of the respective CAS. But even without knowing this, one can understand why the above solution is not recognized. For this one merely needs to determine the solution step by step by means of elementary row manipulations. This can be carried out quite easily with the commands of the TI-92. By means of the commands mRow(*factor, matrix, n*) one can multiply the row *n* of *matrix* by *factor*. The command mRowAdd (*factor, matrix,* n_1,n_2) allows the row n_1 of *matrix* to be multiplied by *factor* and to be added to the row n_2 .

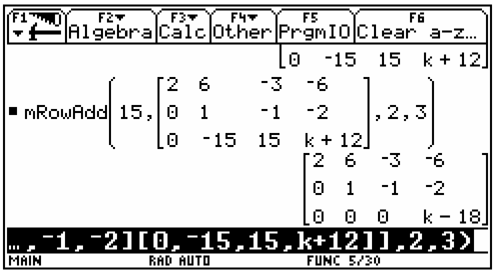


Firstly row 1 is multiplied by -2 and is then added to row #2.

Then row #1 is multiplied again by -2 and the added to row #3 (below left).
Now row #2 is di-vided by -9 (below right).

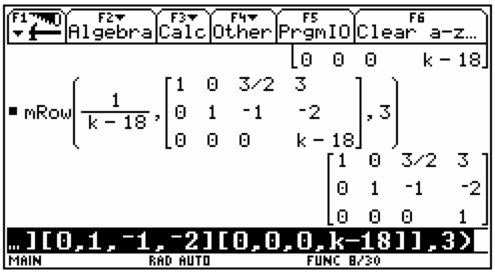






If row #2 is multiplied by 15 and then added to row #3 the result shown on the left is obtained. The last element of the bottom row is k - 18. Therefore it is necessary to differentiate between the two cases $k = 18$ or $k \neq 18$.

In the first case the set (1) of equations is soluble but not in the second. The solution for the case $k = 18$ can, after further transformations, be read directly from the display:



$$k = 18: \quad x = -\frac{3}{2}z + 3 \quad \wedge \quad y = z - 2, \quad z \in \mathrm{Re}$$

The CAS have not made the distinction between the cases, but rather have erroneously divided the last row by k - 18. Even with the TI commands this is consistently possible as can be seen in the display on the left. This result is identical to the result of the command rref() at the beginning of this report.

The lack of distinction in cases and illegal division respectively can occur only, if in the row echelon form of the augmented coefficient matrix, in any single row only zeroes and, to the far right, an expression appear. The set of solutions is then either empty or it has a dimension greater than or equal to 1, as if shown in example (1).

Remarks:

1. The reported problem arose in the course of vector algebra. The students should find out how the intersection of the planes

   E_1: $2x + 6y - 3z = -6$
   E2: $4x + 3y + 3z = 6$ depends on the real number k.
   E_3: $4x - 3y + 9z = k$

2. The commands mRow() und mRowAdd() are very well suited to helping students master the method of solving systems of equations. They can concentrate fully upon the method, since the calculator relieves them of the numerical work.

As a devoted DERIVIAN I tried to reproduce Karl-Heinz´s ideas using DERIVE (Josef): I found two ways to do so.

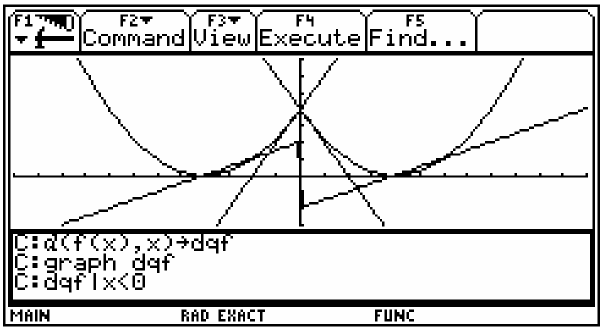
Using the elimination (addition-) method by defining the singular equations, or using a special function from the Utility file VECTOR.MTH.

I´m sure that this file doesn´t need any additional explanation.

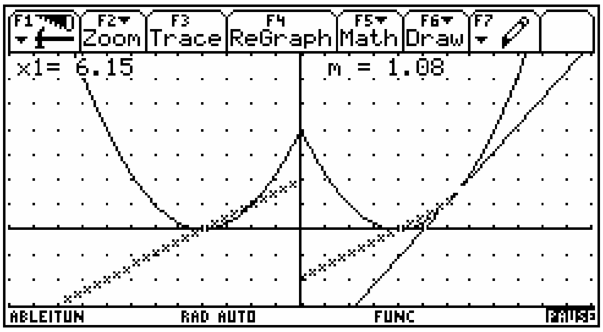
```
#1:  "The system:     "                                                  User
#2:  SOLVE([2·x+6·y-3·z=-6,4·x+3·y+3·z=6,4·x-3·y+9·z=k],[x,y,z])        User
#3:  []                                                               Simp(#2)
#4:  [g1:=2·x+6·y-3·z=-6,g2:=4·x+3·y+3·z=6,g3:=4·x-3·y+9·z=k]           User
#5:  "Define two equations h1 and h2:"                                   User
#6:  [h1:=g2-2·g1,h2:=g3-2·g1]                                           User
                  ┌ 2·x+6·y-3·z=-6 ┐
#7:  [ g1  h1  h2 ]`=│   9·z-9·y=18   │                          Expd(User')
                  └ 15·z-15·y=k+12 ┘
#8:  l1:=3·h2-5·h1                                                       User
                  ┌ 2·x+6·y-3·z=-6 ┐
#9:  [ g1  h1  l1 ]`=│   9·(z-y)=18   │                      User=Simp(User)
                  └   0=3·(k-18)   ┘
#10: "File VECTOR.MTH, copyright (c) 1990-1994 by Soft Warehouse, Inc."User
#11: default1:=1                                                         User
#12: SUBTRACT_ELEMENTS(v,i,j,default1):=VECTOR(IF(m_=i,v -default1·v ,ELEMENT(v,~
                                                       i            j
       m_)),m_,DIMENSION(v))
#13: SUBTRACT_ELEMENTS(m,i,j,s)                                          User
          ┌ 2   6  -3  -6 ┐
#14: gls:=│ 4   3   3   6 │                                              User
          └ 4  -3   9   k ┘
                        ┌┌ 2   6  -3  -6 ┐       ┐ ┌ 2   6  -3  -6 ┐
#15: SUBTRACT_ELEMENTS ││ 4   3   3   6 │,2,1,2│=│ 0  -9   9  18 │
                        └└ 4  -3   9   k ┘       ┘ └ 4  -3   9   k ┘
                        ┌┌ 2   6  -3  -6 ┐       ┐ ┌ 2   6   -3   -6 ┐
#23: SUBTRACT_ELEMENTS ││ 0  -9   9  18 │,3,1,2│=│ 0  -9    9   18 │
                        └└ 4  -3   9   k ┘       ┘ └ 0  -15  15  k+12 ┘
                        ┌┌ 2   6   -3   -6  ┐      15┐ ┌ 2   6  -3   -6 ┐
#24: SUBTRACT_ELEMENTS ││ 0  -9    9   18  │,3,2,──│=│ 0  -9   9   18 │
                        └└ 0  -15  15  k+12 ┘      9 ┘ └ 0   0   0 k-18 ┘
#20: "or with a TI-like MROWADD:"                                        User
#21: MROWADD(faktor,matrix,n1,n2):=SUBTRACT_ELEMENTS(matrix,n1,n2,faktor)

             ┌    ┌ 2   6   -3   -6  ┐     ┐ ┌ 2   6  -3   -6 ┐
             │ 15 │                  │     │ │                │
#22: MROWADD │──, │ 0  -9    9   18  │,3,2│=│ 0  -9   9   18 │
             │ 9  │                  │     │ │                │
             └    └ 0  -15  15  k+12 ┘     ┘ └ 0   0   0 k-18 ┘
```

Example of a script (in combination with W. Pröpper´s program ABLEIT() - find it among the files!)

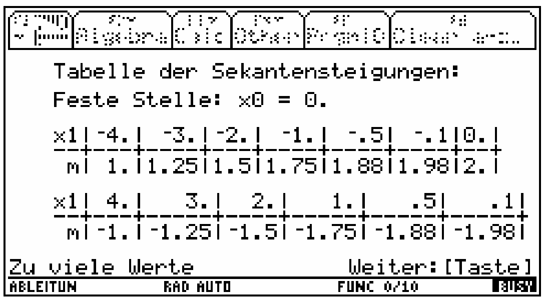
```
 :We investigate the differential quotient of a special function for x=0
 :We set up the screen:
C:setmode ({"Split Screen","Top-Bottom","Split 1 App","Home","Split 2 App","Text Editor","Split Screen Ratio","2:1"}):clrhome
C:(abs(x)/2-2)^2»f(x)
 :f(x=0) = ?
C:f(0)
 :we define the rate of change approaching from right hand side: fromr(x)
C:(f(x+h)-f(x))/h»fromr(x)
 :Now special for  x=0:
C:fromr(0)
 :From right: limes mit h»0
C:limit(fromr(0),h,0)
 :Why that?
 :h mus be positive! Hence:
C:fromr(0)|h>0
 :Failed! Help the TI and try Expand:
C:expand(fromr(0))|h>0
 :the limit is?
C:limit(ans(1),h,0)
```




```
 :the slope of the tangent is?
 :the intersection of the tangent and the y-axis is?
 :the equation of the Tangent?
C:ª2x+4»t1
 :let us plot it!
C:clrgraph
 :the settings for the Graph screen:
C:ª12»xmin:12»xmax:ª3»ymin:7»ymax
C:graph t1
 :Now let's approach from the left:
C:(f(x)-f(x-h))/h»froml(x)
C:expand(froml(0))|h>0
C:limit(ans(1),h,0)
```




```
 :the derivatives are different -> not differentiable!!
 :we obtain another tangent
C:2x+4»t2:graph t2
 :Could you imagine the curve's shape?
C:graph f(x)
 :It has a "Cusp"
 :Do you have an idea how the derivative looks like?
 :Zeros of the derivative?
 :What's the value of the derivative for x=0?
 :"Let" calculate the derivative and store under dqf(x):
C:¶(f(x),x) »dqf(x)
C:graph dqf(x)
C:dqf(x)|x<0
C:dqf(x)|x>0
 :Can you explain the last results?
 :Dont forget house keeping!!
C:delvar dqf,fromr,froml,dqf,t1,t2
C:setmode ("Split Screen","Full")
 :Back with Diamond HOME!
```



I used this teaching unit to repeat the differentiability of a function using a View Screen. The two products - the script and the program – supported each another in an excellent way. Josef