THE DERIVE - NEWSLETTER #39

THE BULLETIN OF THE

DERIVE®

USER GROUP

+ TI 92

Contents:

revised version 2018 **September 2000**

[1] Mathematics in Action, Modelling the Real World Using Mathematics
Richard Beare, 531 pages + CD, Chartwell-Bratt, 1997, ISBN 91-44-00580-6.
With 91 Spreadsheet Models to accompany the Text.

I found this wonderful book in Philip Yorke´s booth at the Liverpool Conference. Philip told me that Richard Beare is working on a DERIVE-version of his book. 10 chapters present a rich variety of applications together with the mathematics behind: What is mathematical modelling; Getting started with graphs, data and simple algebra; Modelling Step by Step Processes; Modelling Continuous Processes; Flow Models and Compartment Models; Population Interactions; Case Study - Epidemics; Models in Mechanics; Modelling Random Processes; Spatial and Diffusion Models. Two examples, which I really enjoyed: Queues for showers on a campsite and Modelling the decline in numbers of loggerhead sea turtles using a Leslie matrix approach.

[2] Potenz-, Exponential- und Logarithmusfunktionen
Michael Roser, Sauerländer, 2000, ISBN 3-7941-4337-X.

"Der Funktionsbegriff nimmt eine zentrale Stellung in der Mathematik der Maturitätsschulen ein. Neue Schwerpunkte im Unterricht werden auch durch den sinnvollen Einsatz neuer Technologien, beispielsweise Computeralgebrasysteme (CAS), gesetzt. Daher ist auch Teil 4 der Reihe «Mathematik für Mittelschulen» für den Einsatz von grafikfähigen Taschenrechnern oder Computeralgebrasystemen konzipiert.

[3] Exploring Chaos With the TI-89, Reimund Albers,
bk teachware SL-11, ISBN 3-901769-27-7
Through the investigations presented in this booklet the reader becomes acquainted with some of the fundamental concepts of chaos theory: iteration, attractor and repeller, fixed point, cycles, Julia sets, Mandelbrot set.

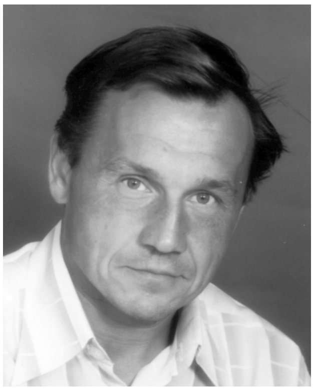
Interesting WEB sites http://......

`www.kutzler.com`	Bernhars Kutzler´s personal web site
`www.bk-teachware.com`	Bernhard Kutzler´s Online Shop
`www.chartwellyorke.com/dfwind.html`	Philip Yorke´s DfW web site
`www.acdca.ac.at`	Austrian Center of Didactics of Computer Algebra
`www.mailbase.ac.uk/lists/derive-news`	The *DERIVE*-News web site
`ham.nw.schule.de/mgh/ags/naturwissenschaft/physikag/physikversuche.html`	Physics projects done by students using the CBL and the TI-92
`www.bham.ac.uk/msor/came/events/weizmann/`	Web proceedings of the CAME workshop in Israel
`https://www.mi.sanu.ac.rs/vismath/`	Visual Mathematics home page
`http://www.mi.sanu.ac.rs/vismath/parker/gr.htm`	DPGraph Gallery
`http://www.nsta.org/publications/quantum.aspx`	National Science Teachers Association
`www.utexas.edu/world/lecture/math/`	The World Lecture Hall
`www.shu.edu/projects/reals/reals.html`	Interactive Real Analysis

Attention please: The links in red are not valid 2018

Dear DUG members,

Possibly this DNL will reach you some days later as usual. I beg your pardon for that. But this summer was very interesting and busy - I cannot say that I had much time for relaxing - and fall brought much more work in school as I have expected (more teaching hours and some new courses) -together with a lot of organizing in-service courses for Austrian teachers.



Two wonderful conferences brought again together the members of the DERIVE and TI-family: Portoroz and Liverpool. I´d like to thank the organizers of both events for their personal dedication to make them a success. Please accept our congratulations, you did a great job. Terence Etchells needs some training in weather making. However, the 10 days after the Liverpool conference which I spent in Wales were very sunny and warm, thanks Terence. It was a pity to miss some friends, at the other hand we met many of you and we made a lot of new friends.

It is a pleasure for me to announce that we plan to organize both conferences - ACDCA and DERIVE/TI-92 in 2002 at the same time on the same place to support exchanging ideas and meeting old and new friends from all over the world. And which place could be better than Vienna. So we are looking forward to welcoming you as guests in Austria. By the way, Michel Beaudin presented the idea to have the 2004 conference in Canada!!!

Preparing this DNL was difficult as ever: at first I am sitting in front of a big pile of papers, then I find some immediate contributions which I want to include (Adrian's, the 3D-plot, Rüdeger's), I add some comments and some messages for the User Forum .... and it has happened again, some candidates for DNL#39 must wait for appearing in DNL#40.

Best regards to you all

Josef

DERIVE for Insiders

Dieter Wickmann had very difficult question regarding DERIVE interna and Theresa Shelby answered:

Is there a DfW5 interface to Win32-API to change the mouse control and its attributes?

No, the mouse cursor position and attributes are handled by the Windows operating system and DfW5 doesn´t get involved except when performing certain operations (like selecting a plot range).

Is it possible to send instructions via Win32-API to DERIVE?

If you knew the DERIVE process ID and/or window it might be possible to construct a Window message and send it to DERIVE. However, there are no defined messages that will return what I believe you need to figure out the DERIVE x- and y-coordinates of the mouse cursor.

If no, is it possible to have access to DERIVE via OLE?

Making DERIVE an OLE server is on our wish list. However, it probably won´t happen until DERIVE 6.

Is there a possibility to send DDE-instructions to DERIVE?

I believe so. What kind of DDE instructions would you be sending?

Find all the DERIVE and TI-files on

www.austromath.at/dug/

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

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

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

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

Next issue: December 2000
Deadline 15 November 2000

Preview: Contributions for the next issues

Inverse Functions, Simultaneous Equations, Speck, NZL
A Utility file for complex dynamic systems, Lechner, AUT
Examples for Statistics, Roeloffs, NL
Quaternion Algebra, Sirota, RUS
Various Training Programs for the TI
Sand Dunes, Halprin, AUS
Type checking, Finite continued fractions, Welke, GER
Kaprekar´s "Self numbers", Schorn, GER
Some simulations of Random Experiments, Böhm, AUT
Flatterbandkurven, Rolfs, GER
Comparing statistics tools: a pie chart with DERIVE, a stem & leaf diagram on the TI, Some classroom experiments with DERIVE in a chemical engineering undergraduate course, Cardia-Lopes, POR

and
Setif, FRA; Vermeylen, BEL; Leinbach, USA; Aue, GER; Koller, AUT, ...

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

One year ago, - DfW5 existed only in α-versions - there was a nice problem in the DERIVE-news posted by Louis F Lowell:

Greetings from a very junior Derive(want-to-be) programmer.
I am having difficulty selecting the distinct elements from a list (vector) to form a new vector consisting of only the distinct elements.
As you can see below my most successful attempt is almost there, but I have '?' where the duplicates reside and I lose the last value. Any help will be appreciated.

```
b := [1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 5, 5, 5, 5, 6, 6, 7, 8, 9]

VECTOR(IF(b      ≠ b , b ), i, 1, DIMENSION(b) - 1)
           i + 1    i   i

[?, 1, ?, ?, 2, ?, ?, 3, ?, 4, ?, ?, ?, 5, ?, 6, 7, 8]
1, ?, ?, 2, ?, 3, ?, 4, ?, ?, ?, 5, ?, 6, 7, 8]
```

Lou Lowell, louislow@seacoast.com

This question was the start of a discussion among the programmers and several solutions were presented. I´ll show three of them, which use the features of DfW5, including the new programming tools.

Terence Etchells *presented a program:*

```
[?, 1, ?, ?, 2, ?, ?, 3, ?, 4, ?, ?, ?, 5, ?, 6, 7, 8]

vector_to_set(v, r, size, list) :=
  Prog
    r := 1
    size := DIM(v)
    list := {v↓r}
    Loop
      If r = size
         RETURN list
      r := r + 1
      list := ADJOIN(v↓r, list)

set_to_vector(set, tmp) := VECTOR(tmp, tmp, set)

vector_to_set(b) = {1, 2, 3, 4, 5, 6, 7, 8, 9}

set_to_vector(vector_to_set(b)) = [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

David Stenenga *from Honolulu had another idea:*

```
list_to_set(v) :=
  If v = []
     {}
     {v↓1} ∪ list_to_set(DELETE_ELEMENT(v, 1))

list_to_set(b) = {1, 2, 3, 4, 5, 6, 7, 8, 9}
```

which works in both versions and also works for the empty list which is a small bug in `vector_to_set`.

Johannes Wiesenbauer also played an important role in this discussion and he contributed a solution involving a great deal of calculus, but he also used “good old SOLVE”:

```
b := [9, 1, 2, 2, 4, 5, 9, 3, 3, 1, 10, -1]
l_t_s(list) := RHS(SOLVE(Π(x - b_, b_, list), x))
l_t_s(b) = [1, -1, 2, 3, 4, 5, 9, 10]
```

```
l_t_s(list) := SOLUTIONS(Π(x - b_, b_, list), x)
bb := [9, 1, 2, 2, 4, 5, 9, 3, 3, 1, 10, -1]
l_t_s(bb) = [1, -1, 2, 3, 4, 5, 9, 10]
```

__Gerhard Hagen, Germany__

.... I miss one feature in DfW5: I was happy to find in DNL#29 page 19 a possibility to avoid re-evaluation of RANDOM in an expression such as

```
x :== VECTOR(RANDOM(6)+1,i,1,10)
```

DfW5 seems not to accept :== anymore. Is there another method to achieve the same as earlier with :==?

I have a little problem but I can't find where the mistake is. I hope you can help me on this...

This is Albert Rich´s competent answer:

Hi,

In Derive 5 the := operator behaves identically to the :== operator in version 4 of Derive. So in Derive 5 when an assignment is simplified, the right operand is first simplified and then the result assigned to the left operand. In previous versions of Derive, the right operand was not simplified before being assigned to the left operand.

Therefore, in Derive 5 simplify the expression

```
x :=VECTOR(RANDOM(6)+1,i,1,10)
```

to assign x to be a fixed vector of random integers between 1 and 6, inclusive. However, if you just enter the above expression without simplifying it, x is assigned to the UNsimplified expression on the right.

Note that the :== operator was an undocumented feature in version 4 of Derive. However, for upward compatibility, Derive 5 automatically converts :== to := when parsing in expressions.

In response to your email, I have added the following paragraphs to the Derive 5 on-line help for the Declare > Variable Value command:

When an assignment is entered without simplifying it, the unsimplified right operand of the := operator is assigned to the left operand. However, when an assignment is simplified, the right operand is first simplified and then the result assigned to the left operand. (Note that in previous versions of DERIVE, the right operand was not simplified before being assigned to the left operand.) For example, if you enter the assignment

```
Digit := RANDOM(10)
```

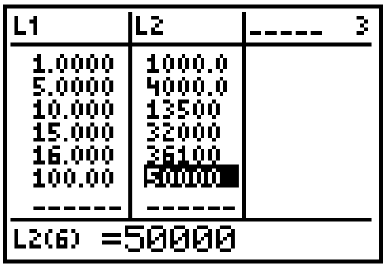
the variable Digit is assigned the value RANDOM(10). Thereafter, Digit will return a new random digit each time it is simplified. However, if you simplify the above assignment, RANDOM(10) is simplified and Digit is assigned the resulting digit. Thereafter, Digit will return this same digit each time it is simplified.

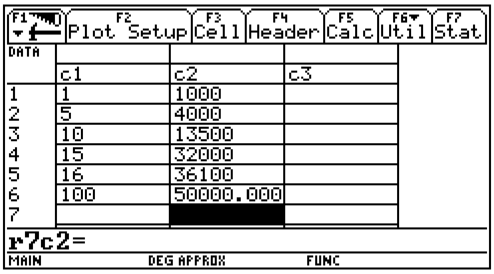
No matter whether an assignment is simplified or not, the left operand of the := operator is not simplified. Thus the left operand of the := operator must be the actual variable to be assigned a value. Use the ASSIGN function if you need to simplify an expression to determine the variable to be assigned a value (see Procedural Programming for details).

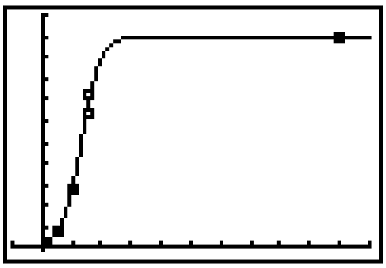
Hope this helps.
Aloha, Albert

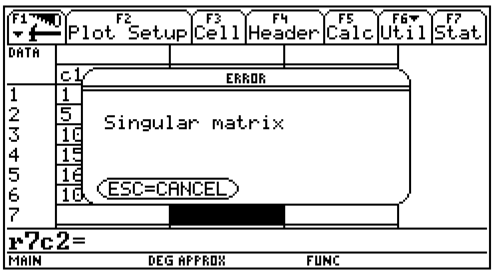
<u>Fritz Tinhof, Eisenstadt, Austria</u>

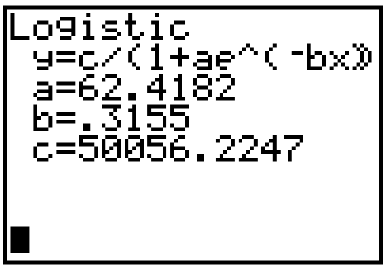
I´d like to perform a logistic regression on my TI-89/92 but unfortunately, I receive an error message (see below). The same fitting procedure does work without any problems on my TI-83?
Any idea, any help?
Regards Fritz

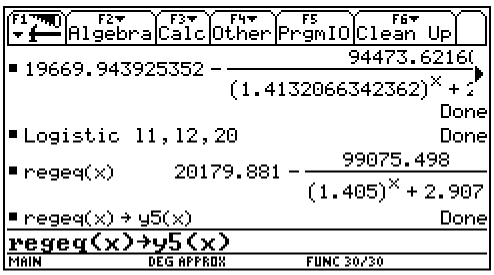






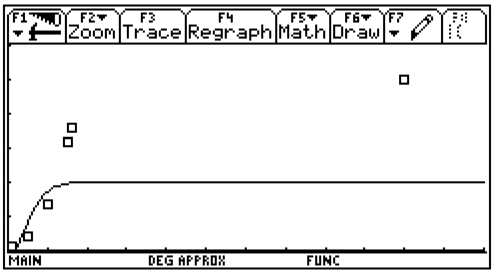






It is not surprising that the V200 performs as bad as the TI-92/89.

The reason is that TI-83 and the CAS-calculators apply different regression models.



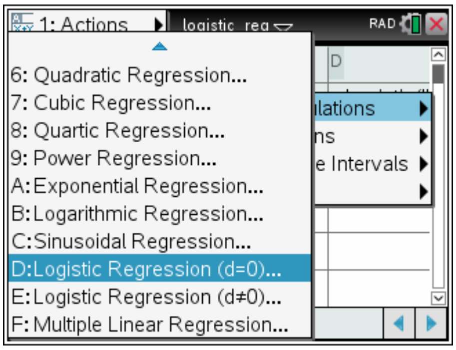
TI-Nspire offers two logistic models:

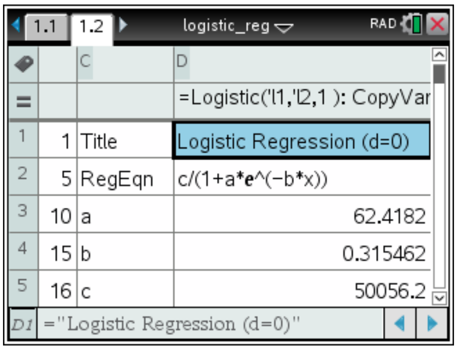
$$f(x) := \frac{c}{1 + a\,e^{-bx}} + d$$

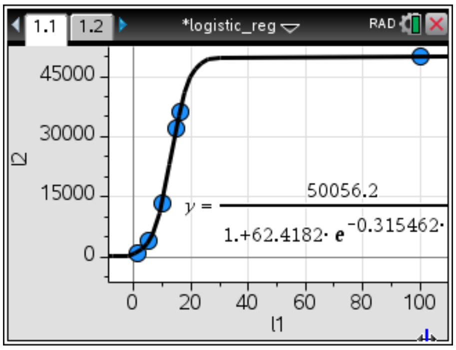
with $d = 0$ and $d \neq 0$.

(The TI-83 assumes $d = 0$.)

See below how it works:







See also Don Phillips' contribution: Nonlinear and other Regressions for TI-89 and DERIVE in DNL#79 from 2010:

$$\text{logisticreg}\begin{bmatrix} 1 & 1000 \\ 5 & 4000 \\ 10 & 13500 \\ 15 & 32000 \\ 16 & 36100 \\ 100 & 50000 \end{bmatrix} = \begin{bmatrix} \text{LogisticReg Line:} & y = \dfrac{4746381274.7\cdot e^{0.31546\cdot x}}{94821\cdot e^{0.31546\cdot x} + 5918554} \\ \text{SSE:} & 422121.9 \\ [a,b,c] & [50056.2,\ 62.418,\ -0.31546] \end{bmatrix}$$

Yves De Racker, Anwerpen, Belgium ydr@vt4.net

In the older version of DERIVE I could transpose a matrix of form `[[x=1,x=5,x=3],[x=0,x=9,x=6],[x=3,x=4,x=5]]` or with the new notation `[x=1,x=5,x=3;x=0,x=9,x=6;x=3,x=4,x=5]`.
But apparently that seems impossible in DfW5!

Albert Rich adr@flex.com

I have found and fixed the problems so it will be in the next version 5.03 of DERIVE. Thanks again for reporting such problems so they can be resolved.

In August we had the annual meeting of T³ (Teachers Teaching with Technology) coordinators in Chantilly, France. At the beginning of the conference Adrian Oldknow posed a problem for the delegates to deal with during the breaks. At the end of the conference he distributed the following paper containing his approach - together with some interesting very general ideas concerning problem solving strategies.

Modelling the Spread of Infection (and a SNOG with the STAR model)

Adrian Oldknow a_oldknow@compuserve.com August 2000

This is about attempts both to solve a problem, and about methodologies for solving problems. It starts with a "lesson-opener" suggested by Richard Taylor of Hockaday School, Dallas, during a recent TI workshop for science teachers in Dublin.

The idea is a simple simulation for the spread of infection. This could be used, say in biology or history, for the spread of infections such as the plague or influenza epidemics. It can also be used in politics, say, for the spread of a rumour - or in science or history for the spread of an invention such as printing. Here we will use a hypothetical model for the spread of good news about graphing calculators between mathematics teachers!

First, we need to define the size N of the population which may be susceptible to infection - i.e. the number of people in the room. Each person is allocated a number from 1 to N. Each step of the "infection game" corresponds to the passing of a day. On day one person 1 is deemed "infected", and stands up. The following day he or she may bump into someone and pass on the good news (or the bad germ). To decide who he or she meets, person 1 chooses a random integer between 1 and N. Of course, if it turns out to be the number 1 then this corresponds to self-infection! In the example below there are 25 people taking part. The first random number is 2, so person 2 is now infected and so stands up as well.

```
25→N
                 25
randInt(1,N)
                  2
                 16
                  5
                 10
```

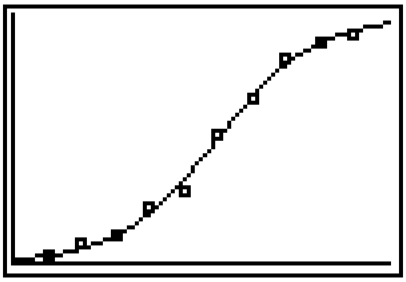
Thus, there are 2 infected people on day 2. Now both 1 and 2 chose a random number to see who they encounter on day 3. In this case person 1 infects person 16, and person 2 infects person 5 - and both the newly infected people now stand. So on day 3 there are 4 people infected. Obviously, we cannot keep on doubling each day, as the number of susceptibles starts to decrease and the chance of encountering already infected people grows. So the question which is at the heart of this problem is:

"*for a given N, how long do you think it will take for all N people to become infected*?"

In his example Richard Taylor collects the data on the numbers infected on each day as a table on the whiteboard. Here is a sample for N = 25 people.

Day	1	2	3	4	5	6	7	8	9	10
Cases	1	2	3	6	8	14	18	22	24	25

Entering this data in the STAT Editor you can produce a scatter diagram, and then fit a "Logistic" regression model to the data:



The form of the function used: $y = c/(1 + ae^{-bx})$ is called the logistic function and was derived by the Belgian Flemish demographer, Jean-François Verhulst in around 1840 as the solution to the differential equation:

$$dP/dt = k\,P\,(P_{max} - P) \quad \text{with initial condition } P(0) = P_0 .$$

This is a modification of the so-called Malthus model of exponential growth, where the rate of increase of the population declines as it approaches the maximum population which limited resources can sustain. For small values of x the function y behaves approximately exponentially, but for large x it approaches $y = c$ asymptotically. Hence the parameter c represents the maximum population P_{max}, which is why it is close to actual value of N.

To speed up the process it helps to have a program, INFECT, which automates the picking of the random numbers and returns the logistic regression model in Y1. The program can be found on the website (page 1). It produces a count of the days in list L1 and the corresponding numbers of infected in list L2. It performs the logistic regression and stores the resulting equation in function Y1. Its output is the scatter diagram of the data with the graph of the logistic fit superimposed on it, as above.

Well now we have an experimental model we can see if we can develop a theoretical one to match it. First we note that there only 3 parameters to determine: a, b, c. So we just 3 need constraints. We already now one of these i.e. $y(1) = 1$ - that is the infected population on day one. Now on day 2 there are just 2 possibilities. The probability that a second person is infected is (N – 1)/N , while that of a self-infection is 1/N. So on day 2 there will be 1 person infected, probability 1/N, or 2 people infected, probability (N – 1)/N. Hence the expected number P_2 of infected is:

$$P_2 = 1.1/N + 2.(N - 1)/N = 2 - 1/N . \qquad [1]$$

What are the possibilities on day 3?

Of course, it depends on day 2 - suppose there is just one person, called A, infected. Then the next possibilities are AA (self-infection) and AB (a new infection), with the same probabilities as above. Suppose though that there are two people infected: AB. Then each person selects another person - so the next possibilities are 2 people AB, 3 people ABC or 4 people ABCD. But these needs enumerating in greater detail: AB can be arrived at by either AAAB, AABB or ABBB through either self-infections or cross-infections! Actually, AABB can be arrived at by both - so there in fact 4 equiprobable ways in which 2 people at day 2 produce the same 2 again at day 3, each with probability $1/N^2$. A similar analysis is needed for ABC which can be arrived through AABC, ABBC or ABCC - with 2 ways for each of the first 2. So the probability of 3 infected at day 3 given 2 at day 2 is $5(N - 2)/N^2$. Finally ABCD can only be arrived at in one way - with probability $(N - 2)(N - 3)/N^2$.

As a check we can verify that: $4/N^2 + 5(N - 2)/N^2 + (N - 2)(N - 3)/N^2 = 1$.

So, multiplying to get the conditional probabilities we find that probabilities for the number of infected on day 3 are as follows:

1	$1/N^2$
2	$(N-1)(N+4)/N^3$
3	$5(N-1)(N-2)/N^3$
4	$(N-1)(N-2)\cdot(N-3)(N-4)/N^3$

and again, we can check that these also sum to 1. The expected number of infected P_3 on day 3 is then given by:

$$P_3 = 4 - (7N^2 - 6N + 2)/N^3 \qquad [2]$$

So we now know that $y(1) = 1$, $y(2) = P_2$ and $y(3) = P_3$ - can we solve the resulting equations for parameters a, b and c ?

If $y = c/(1 + ae^{-bx})$ then we can rearrange this in the form: $e^{bx} = ay/(c - y)$, and hence:
$bx = \ln a + \ln y - \ln(c - y)$. If we substitute in turn the pairs (1,1), $(2,P_2)$ and $(3,P_3)$ we will have a system of 3 non-linear simultaneous equations in a, b and c in terms of P_2 and P_3, and hence of N:

$$\begin{aligned} b &= \ln a && - \ln(c - 1) \\ 2b &= \ln a + \ln P_2 && - \ln(c - P_2) \\ 3b &= \ln a + \ln P_3 && - \ln(c - P_3) \end{aligned}$$

Eliminating b yields the 2 equations:

$$\begin{aligned} \ln a &= \ln P_2 - \ln(c - P_2) + 2\ln(c - 1) \\ 2\ln a &= \ln P_3 - \ln(c - P_3) + 3\ln(c - 1) \end{aligned}$$

Finally eliminating a yields the equation:

$$\ln(c - P_3) - 2\ln(c - P_2) + \ln(c - 1) = \ln P_3 - 2\ln P_2$$

and collecting terms and equating logarithms we get:

$$P_2^2 (c^2 - (1 + P_3)c + P_3) = P_3 (c^2 - 2P_2 c + P_2^2)$$

So we arrive at:

$$c = P_2 (P_2(1 + P_3) - 2P_3)/(P_2^2 - P_3) \qquad [3]$$

from which we can recapture a and b as:

$$a = P_2 (c - 1)^2 / (c - P_2) \qquad [4]$$

and

$$b = \ln a/(c - 1) = \ln P_2 (c - 1)/(c - P_2) \qquad [5]$$

From [3] we see that we cannot fit such a model to data where $P_2^2 = P_3$, for in that case we would have 3 data points (1,1), (2,P_2) and (3,P_2^2) which are in geometric progression, and hence are fitted by an exponential model.

We can now substitute for P_2 and P_3 in terms of N from [1] and [2]:

$c = (2N - 1)(2N^3 - 3N^2 + 4N - 2)/(N^2 (3N - 2))$ [6]

$a = (4N^3 - 7N^2 + 6N - 2)^2/(2N^3 (N - 1)(3N - 2))$ [7]

$b = \ln((4N^3 - 7N^2 + 6N - 2)/(2N (N^2 - 2N + 1))$ [8]

In particular we can see that, for large N, these approximate to:

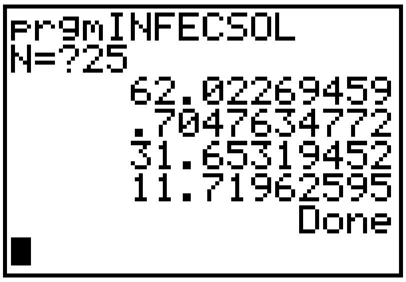
$c \approx 4N/3$ $\quad a \approx 8N/3$ $\quad b \approx \ln 2 \approx 0.7$

The time s taken to saturation will be given when the population reaches c - ½ , so we must solve:

$c/(1 + ae^{-bs}) = c - ½$ for s to give:
$s = 1/b \ln a(2c - 1) = (\ln a + \ln (2c - 1))/(\ln a - \ln (c - 1))$ [9]

and for large N this approximates to $s \approx 2.83 + 2.88 \ln N$ [10]

So we have now completed the analysis of a logistic model which agrees with the theoretical probability simulation at days 1, 2 and 3. Now we need to see how well it agrees with the data produced from the simulation. The program INFECSOL - downloadable - computes P_2, P_3, a, b, c and s for any given N.



Running this with N = 25 gives the following comparisons with 5 runs of INFECT:

	a	*b*	*c*	*s*
Theoretical	62.0	0.705	31.7	12
Run 1	64.6	0.737	27.4	9
Run 2	76.6	0.901	25.4	9
Run 3	113.2	0.876	27.6	8
Run 4	92.6	0.892	24.8	12
Run 5	53.8	0.673	25.2	13

So, while the value of s seems to match quite well, c seems to be an overestimate, and a seems to be an underestimate. Also the values of a and b from the runs of the simulation seem to vary considerably.

Well, what can we conclude? Our theoretical model was based on the premise that the logistic model would be a good fit for data generated from this simulation. We can only conclude that it is not as good a fit as we would have liked. Ideally we should know more about the probability distribution of the number of infected P(n) at day n with N people. But the calculation of even P(3) was hard and gives no suggestion that we can find either a simple form for P(n), or even a recurrence for P(n) in terms of P(n-1).

An alternative approach would be to start with the parameter c, which gives the maximum population. The approximation 4N/3 certainly is not realistic, since we only have N people. As the logistic function approaches, but never reaches, c we could conveniently take:

$$c = N + 1 \qquad [11]$$

So, to find a and b we only now need the data points (1,1) and $(2,P_2)$ to get:

$$a = P_2\,(c - 1)^2 / (c - P_2) = N^2\,(2N - 1)/(N^2 - N + 1) \qquad [12]$$

$$b = \ln P_2\,(c - 1)/(c - P_2) = \ln N\,(2N - 1)/(N^2 - N + 1) \qquad [13]$$

Testing these with N = 25 we have:

	a	*b*	*c*	*s*
New Theoretical	51.0	0.712	26	11.0

Also, for large N we now have:

$$a = N + 1 \qquad b \approx \ln 2\, c \approx 2N$$

and:

$$s \approx 2 + (2/\ln 2)\ln N \approx 2 + 2.88 \ln N \qquad [14]$$

Now it is up to you to decide whether this gives a reasonable fit, and, if not, to propose alternative models or methods of analysis.

So where did the **STAR** and the **SNOG** come in?

Well **STAR** stands for:

Strategist
Technician
Accountant
Reporter

And these seem to be the functions which are in common when we tackle mathematical problems, or do some modelling, or fit some data etc. To be a **Strategist** you need to be aware of a good range of possible representations, mathematical ideas etc. which may be helpful in the given situation. To be a **Technician** you need either to be competent at manipulating the symbols, numbers, representations etc. yourself, or to be competent at using technological aids (yes - I used Derive for [6], [7], [8]!). To be an **Accountant** you need to view your results with scepticism and ask whether (a) they are really accurate and reliable, and (b) whether they are sensible in terms of the original problem. To be a **Reporter** you need to be able to communicate your results clearly either to yourself, or to others, and be able to choose suitable notation and make use of graphs, diagrams, tables etc. - and again technological aids play an increasing part.

So, we should be happy if we can get enthusiastic teachers to infect each other about the capacity of graphing calculators to improve understanding and problem solving. We must get across the message that they raise the level of mathematics being tackled, and are not just mere substitutes for human thought. By supporting the Technician they allow more scope for the Strategist, Accountant and Reporter to develop their own important skills.

(You can find infect() on the web site, running only for the TI-92+/89 because of the logistic regression, which is not implemented on the ordinary TI-92. J.)

Well what does **SNOG** stand for? (The word "snog" in English is a slang word used by the young to mean: kissing and cuddling!). Here it is used for:

Symbolic
Numeric
Oral
Graphic

And these are all key forms of representing the problem and reporting attempts at its solution. Again, we emphasise that while the graphing calculator is excellent at being able to process problems in Symbolic, Numeric and Graphic forms - its power still rests in its symbiosis with its human master/mistress who in the end has to make explanations and decisions using natural language.

When Adrian had set the problem two of us - Jose Paolo Viano, the great magician from Portugal and I - joined facing the challenge to find a solution, but I didn't realize that it would be so difficult and time consuming. With the random variable

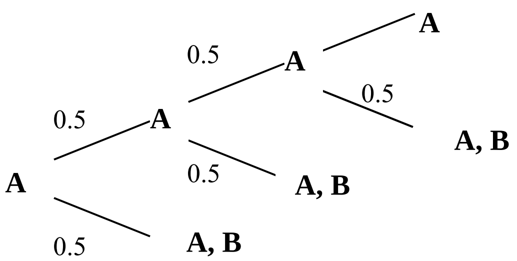
X = number of days until the whole population of N individuals is infected

we have the task to find its expected value $E(X)$ as a function of N.

For N = 1 it is obviously 1.

For N = 2 the expected value is 3. I use a probability tree. We have two people A and B and let us assume that on the first day person A is infected - otherwise change the names:

X = 1 X = 2 X = 3 X = 4 ..... and so on



The probabilities on all branches are 0.5. So, we can easily derive the expected value of X:

$$E(X) = 1\cdot 0 + 2\cdot\frac{1}{2} + 3.\frac{1}{4} + 4\cdot\frac{1}{8} + \ldots\ldots = \sum_{X=2}^{\infty} X\left(\frac{1}{2}\right)^{X-1} = 3$$

This was not so difficult, but now for N = 3:

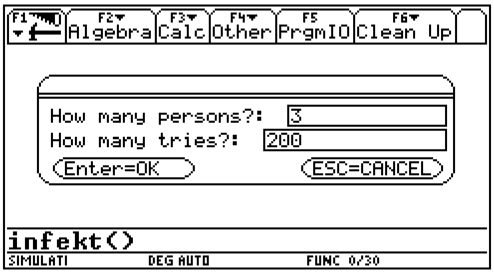
Again I built up my probability tree - and it turned out to be much more complicated than before, but at last I achieved a result in a closed formula:

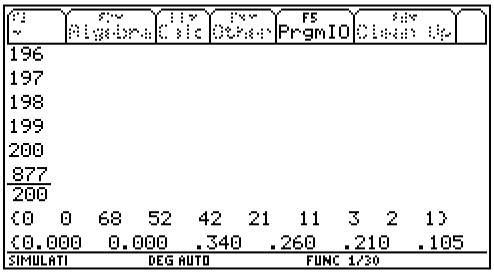
$$E(X) = \sum_{X=3}^{\infty} X\cdot\sum_{n=1}^{X-2} 2\cdot\frac{5}{9}\left(\frac{1}{3}\right)^{n}\left(\frac{4}{9}\right)^{X-2-i}$$

My TI-92 found the limit with $E(X) = \frac{43}{10}$. Fortunately, Jose Paolo had the same result. For

I was not able to find a formula for $N = 4$ like the above one, so I tried another approach.

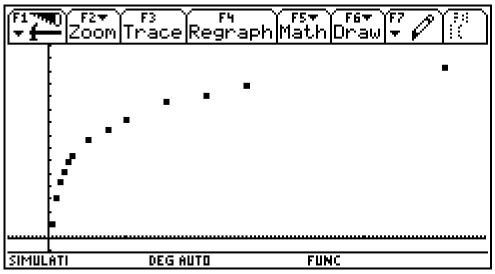
Having prepared recently some TI-92 simulations for random experiments I wrote the program infekt() for simulating the spread of this infection to confirm my results and to check the expected next ones. (It works also on the TI-92.)



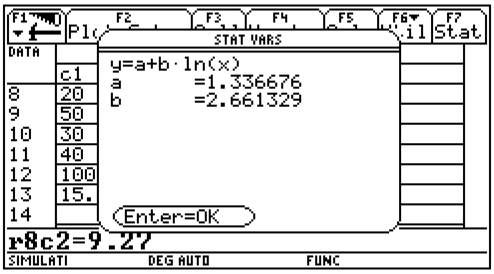


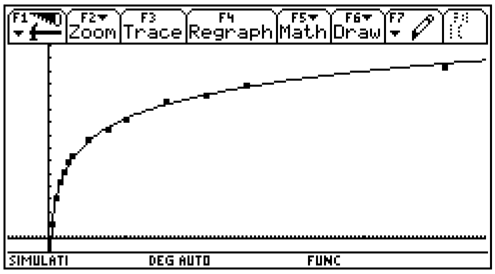
4.385 was very encouraging and so were the estimates for the various probabilities. I went on simulating for $N = 4, 5, 6, ...$ and set up the following table (using 100 experiments each time):

N	$E(X)$
1	1
2	3
3	4,3
4	5,04
5	5,92
6	6,35
10	7,70
15	8,43
20	9,27
30	10,56
40	11,12
50	11,82
100	13,22



The scatter diagram pointed very clear to a logarithmic function. I used the TI-92 to find a logarithmic regression line and superimposed its graph, and I was very much impressed!
You can download the respective program from our web site.

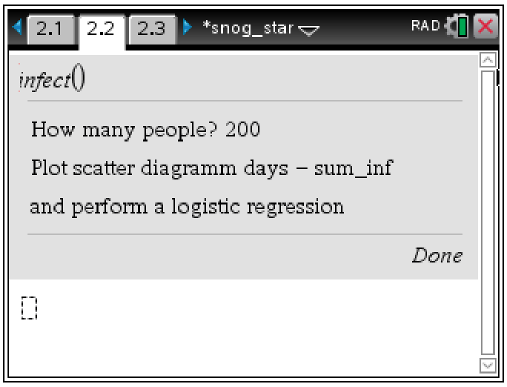


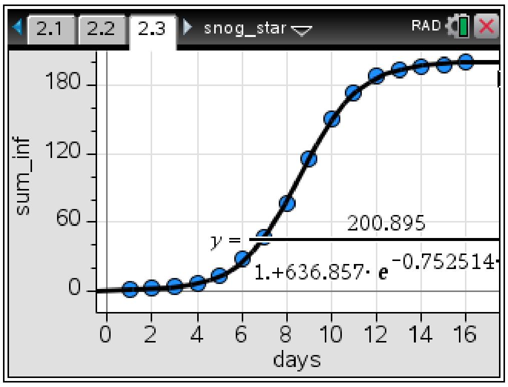


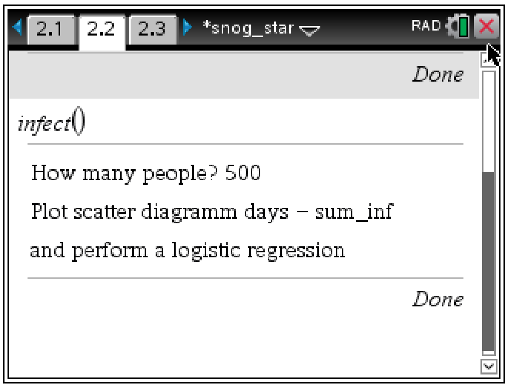
Finally, I had a simulation run of 200 experiments with 200 people. My empirical mean was 15.02. Adrian´s model gives 15.26 and mine 15.44.

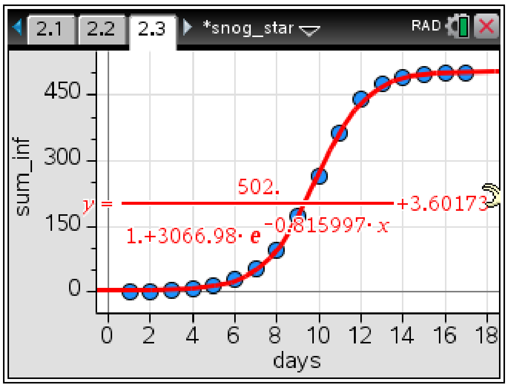
I’d like to invite you to join our efforts to find a solution for the problem - if there is one - and don’t blame us that we were unable to find a better one up to now. But as Adrian wrote very clear in his paper: there are so many ways for modelling.

How the infection spreads on the TI-Nspire

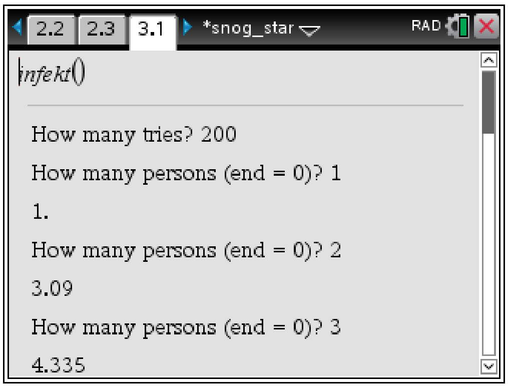


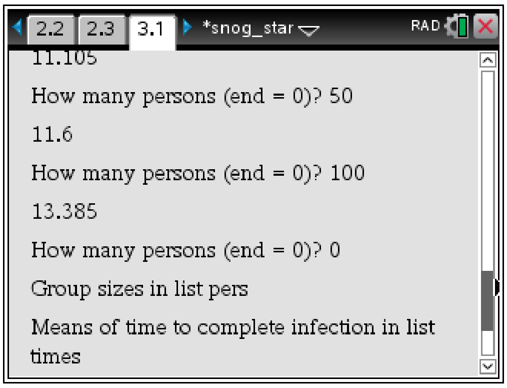


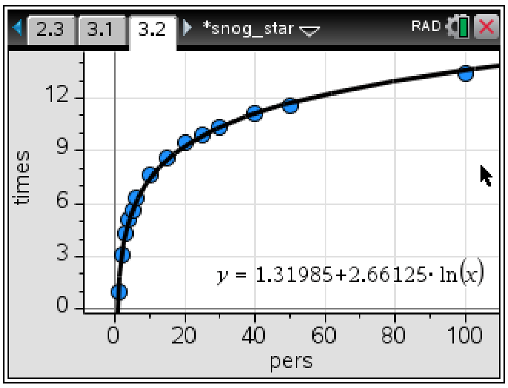




We produce data for the expected logarithmic regression line.



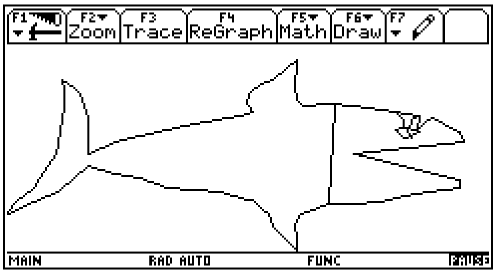


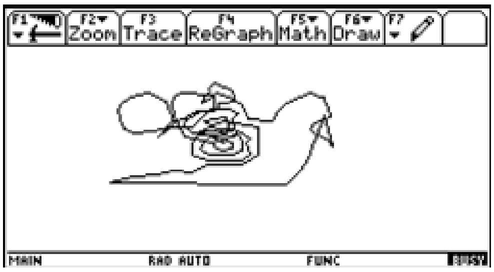


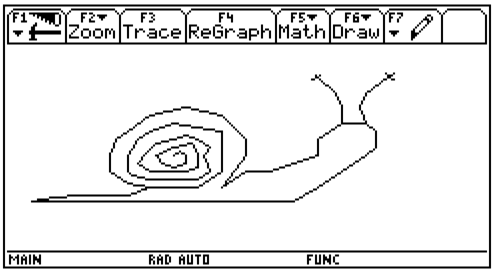
P hwdp ruskrvhv#rq#p |#WL0<5#

Magdalena Barthofer, Waidhofen a/d Ybbs

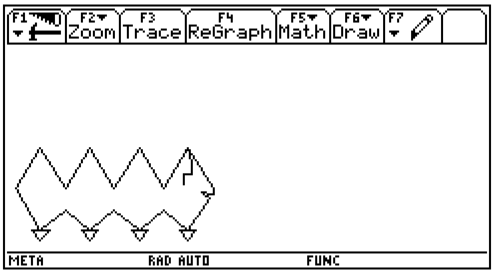
After we had got the new program called „meta“, I immediately started creating a metamorphose. At first, I choose an object which could be a plant, an animal or just a phrase. I took a bird and drew it as simple as possible into a coordinate system. To portray my picture on the TI 92 I had to key in all the necessary points. After this I had to choose another object, the bird should develop in. I decided to draw a bird again, but at this time the animal should fly. I replayed the actions with different points to get a flying bird. Finally, my first metamorphose was finished. After this success I decided to create other metamorphoses, so I drew a mosquito becoming an elephant and a fish that turns into a snail.

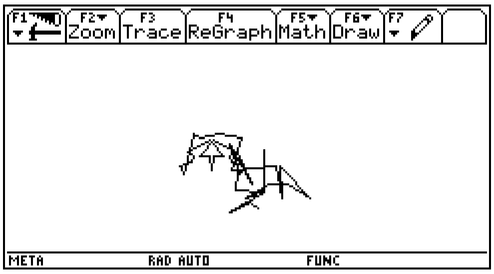


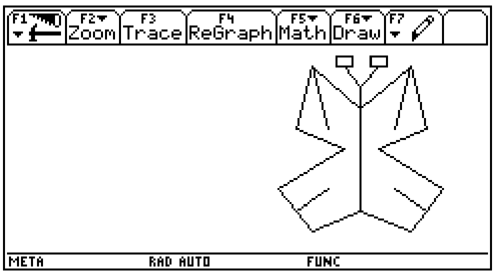




Kathrin Leitner, one of my students, performed a metamorphosis transforming a caterpillar to a butterfly. Josef







Request from Don Taylor:

With 2-d plots we have had option of plotting connected or discrete points.

With version 4 we didn't have the ability to spin the boxes so plotting discrete points in 3-d might not have been as easy to visualize or locate those points in space.

But now with the ability to spin the 3-d plot I have a dataset of 1000 points and the interconnecting lines obscure almost all of the features I am trying to see.

I suppose I CAN manually plot one point at a time but doing this even a few dozen times, let alone a thousand times, seems like more than I want to do.

Is there a trick that I haven't been able to find that can turn off the connecting lines in 3-d plots?

Or could we have yet one more option button?

Thanks
Don Taylor

The following excerpt from the Derive on-line help on the Insert > Plot Command explains how to plot points in the 3D-plot window:

For 3D data-point plots, the form of the vector being plotted determines how the data-points are displayed. If xi, yi, and zi are numerical constants, the following summarizes how vectors of these constants are plotted:

`[x, y, z]` and `[[x, y, z]]` each plot as a single isolated point.

`[[[x1, y1, z1]], [[x2, y2, z2]], ..., [[xn, yn, zn]]]` plots as n isolated points.

`[[x1, y1, z1], [x2, y2, z2]]` plots as a single line segment.

`[[x1, y1, z1], [x2, y2, z2], ..., [xn, yn, zn]]` plots as n-1 connected line segments.

`[[z11, z12, ..., z1n], [z21, z22, ..., z2n], ..., [zm1, zm2, ..., zmn]]`
plots as the functional surface defined by the n x m points `[i, j, zij]`.

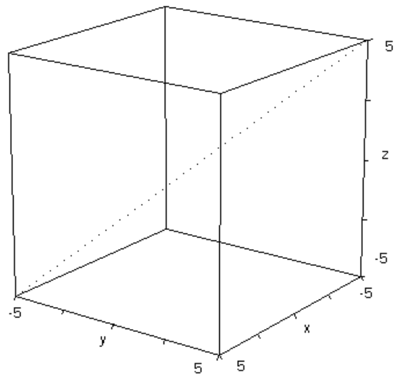
```
[[[x11, y11, z11], [x12, y12, z12], ..., [x1n, y1n, z1n]],
 [[x21, y21, z21], [x22, y22, z22], ..., [x2n, y2n, z2n]]
...
 [[xm1, ym1, zm1], [xm2, ym2, zm2], ..., [xmn, ymn, zmn]]]
```
plots as the parametric surface defined by the n x m points [xij, yij, zij].

Hope this helps,
Albert D. Rich
Co-author of Derive

I´ll try to illustrate Al´s answer showing the representation of some sets of points in various ways. Josef

Represent a cube's diagonal by a set of discrete points:

```
VECTOR([[5, -5, -5] + t·[-10, 10, 10]], t, 0, 1, 0.02)
```



A pyramid as a sequence of line segments - a special "space curve":

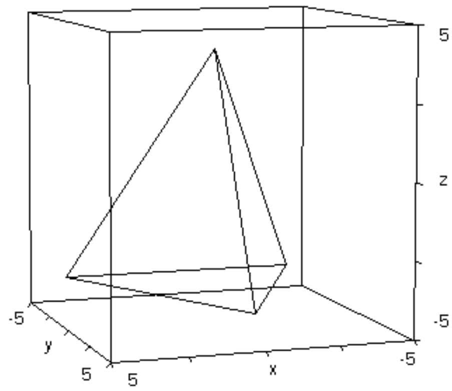
```
[a := [-4, -4, -4], b := [4, -4, -4], c := [0, 4, -4], d := [0, 0, 4]]

pyr := [a, b, c, a, d, c, b, d]
```

The same solid defined by its edges (results in the same plot):
[[a, b]; [b, c]; [c, a]; [a, d]; [b, d]; [c, d]] or

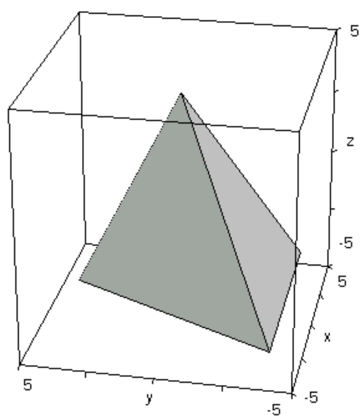
[[[a,b]],[[b,c]],.....]

$$\begin{bmatrix} [a, b] \\ [b, c] \\ [c, a] \\ [a, d] \\ [b, d] \\ [c, d] \end{bmatrix}$$



You can present it as a "parametric surface" editing   [a, b; b, c; c, a; a, d; b, d; c, d]

$$\begin{bmatrix} a & b \\ b & c \\ c & a \\ a & d \\ b & d \\ c & d \end{bmatrix}$$



A Note on the Evaluation of Fresnel Integrals in DERIVE

S. Abraham Ibrahim, P. Fernández de Córdoba

Departamento de Matemática Aplicada
Universidad Politécnica de Valencia
Valencia, Spain

Abstract

We present a set of procedures to evaluate Fresnel Integrals with DERIVE. In particular we use their definitions, their expansions in terms of Bessel functions and a continued fractions representation method. DERIVE proves to be a powerful academic tool because of its simplicity to program different procedures and allows to the students to face the same problem from different points of view; this feature converts DERIVE in a helpful assistant for teaching.

This paper presents a new code including continued fraction techniques which proves to be very efficient to evaluate Fresnel Integrals. Finally, due to the oscillatory behaviour of the integrands involved, this approach is more efficient than the *crude* calculations based on direct integration by quadrature.

1 Fresnel Integrals. Definitions.

The Fresnel Integrals (**FIs)** can be expressed as [1]

$$C(x) = \int_0^x \cos\frac{\pi t^2}{2}\,dt \qquad\qquad S(x) = \int_0^x \sin\frac{\pi t^2}{2}\,dt \tag{1}$$

Considering real arguments x, both $C(x)$ and $S(x)$ vanish for $x = 0$ and have an oscillatory character. Both $C(x)$ and $S(x)$ tend to ½ for $x \to \infty$.

The series expansion for **FIs** can be written as follows [1,2,3]:

$$\begin{aligned} C(x) &= \sum_{k=0}^{\infty}\frac{(-1)^k}{(2k)!}\left(\frac{\pi}{2}\right)^{2k}\frac{x^{4k+1}}{4k+1} \\ S(x) &= \sum_{k=0}^{\infty}\frac{(-1)^k}{(2k+1)!}\left(\frac{\pi}{2}\right)^{2k+1}\frac{x^{4k+3}}{4k+3} \end{aligned} \tag{2}$$

which converge for arbitrary x.

To evaluate the FIs for high values of $|\, x\,|$ it is convenient to use the following asymptotic representation [1]

$$\begin{aligned} C(x) &= \frac{1}{2} - \frac{1}{\pi x}\left[B(x)\cos\frac{\pi x^2}{2} - A(x)\sin\frac{\pi x^2}{2} \right] \\ S(x) &= \frac{1}{2} - \frac{1}{\pi x}\left[A(x)\cos\frac{\pi x^2}{2} + B(x)\sin\frac{\pi x^2}{2} \right] \end{aligned} \qquad (3)$$

where

$$\begin{aligned} A(x) &= \sum_{k=0}^{N} \frac{(-1)^k \alpha_{2k}}{\left(\pi x^2\right)^{2k}} + O\left(|x|^{-4N-4}\right) \\ B(x) &= \sum_{k=0}^{N} \frac{(-1)^k \alpha_{2k+1}}{\left(\pi x^2\right)^{2k+1}} + O\left(|x|^{-4N-6}\right) \\ \alpha_k &= 1\cdot 3 \cdot \ldots\ldots\ldots \cdot (2k-1), \quad \alpha_0 = 1 \end{aligned} \qquad (4)$$

If we now define the functions $C_2(x)$ and $S_2(x)$ through the relations $C(x) = C_2(\pi x^2/2)$, $S(x) = (\pi x^2/2)$, an expansion in terms of Bessel functions for **FIs** can be obtained[1]

$$\begin{aligned} C_2(x) &= J_{1/2}(x) + J_{5/2}(x) + J_{9/2}(x) + \ldots.. \\ S_2(x) &= J_{3/2}(x) + J_{7/2}(x) + J_{11/2}(x) + \ldots.. \end{aligned} \qquad (5)$$

This expansion, together with the symmetry relation $C(-x) = C(x)$, $S(-x) = -S(x)$ allows to compute $C(x)$ and $S(x)$ for positive and negative values of the argument.

The series expansion (2) is useful for the evaluation of $C(x)$ and $S(x)$ at low x values; for high x values it is convenient to use the asymptotic representation (3), while the expansion (5) proves to be adequate at intermediate values ([8]).

An efficient approach to evaluate FIs is presented in ref.[5]: there is *complex* continued fraction that yields both $C(x)$ and $S(x)$ simultaneously:

$$\begin{aligned} C(x) + i\,S(x) &= \frac{1 + i}{2}\, erf(z) \\ z &= \frac{\sqrt{\pi}}{2}(1 - i)\,x \end{aligned} \qquad (6)$$

where

$$\begin{aligned} e^{z^2}\, erf\,c(z) &= \frac{1}{\sqrt{\pi}}\left(\frac{1}{z+}\,\frac{1/2}{z+}\,\frac{1}{z+}\,\frac{3/2}{z+}\,\frac{2}{z+}\ldots \right) = \\ &= \frac{2z}{\sqrt{\pi}}\left(\frac{1}{2z^2+1-}\,\frac{1.2}{2z^2+5-}\,\frac{3.4}{2z^2+9-}\ldots \right) \end{aligned} \qquad (7)$$

In the last line we have converted the *standard* form ot the continued fraction to its *even* form[4], which converges twice faster. Notice that this method requires the use of complex variables.

2 Program specifications

(See file `INTFRE.MTH`)

In this section, we introduce the **DERIVE** program, using the continued fraction representation of the error function.

- With expression #1 we change the real *x-variable* to complex *z-variable.*
- With expressions #2 and #3 we define the *a´s* and *b´s* in expression (7) by comparing with the general expression of a continued fraction:

$$f(x) = b_0 + \frac{a_1}{b_1 +}\,\frac{a_2}{b_2 +}\,\frac{a_3}{b_3 +}\,\frac{a_4}{b_4 + \ldots}$$

- Evaluate the α´s and the β´s in expressions #4 and #5 that define the recurrence relations[6] that determine the different approximants of the continued fraction:

$$\alpha_k = b_k\,\alpha_{k-1} + a_k\,\alpha_{k-2}\,; \quad k = 1,2,\ldots n$$
$$\beta_k = b_k\,\beta_{k-1} + a_k\,\beta_{k-2}\,; \quad k = 1,2,\ldots n$$

with

$$\alpha_{-1} = 1, \quad \beta_{-1} = 0,$$
$$\alpha_0 = b_0, \quad \beta_0 = 1.$$

- We store the values of $\alpha(k,z)/\beta(k,z)$ (approximations to the continued fraction) in the H(k,z) vector in expression #6.
- The error function and complementary error function are defined in expressions #8 and #9 using the continued fraction.
- We calculate **FIs** in expressions #10 and #11 using equation (6).

3 Results and Conclusions.

We have analyzed different and efficient methods for the calculation of **FIs**.

One code evaluates the **FIs** though an expansion in terms of Bessel functions of fractional order, and another one using a continued fraction representation. In table 1 we present the results obtained with the continued fraction procedure and compare with those of the ref. [2]:

	Continued Fraction Method		Abramowitz and Stegun[2]	
x	*C*(10,*z*)	*S*(10,*z*)	*C*(10,*z*)	*S*(10,*z*)
2	0.4884556	0.3434215	0.4882534	0.3434157
3	0.6057153	0.4963178	0.6057208	0.4963130
4	0.4984262	0.4205162	0.4984260	0.4205158

Table 1. Calculated values of the **FIs** using continued fraction method, *x* and *z* are related by equation (6).

As an exercise, the students can evaluate **FIs** going up to higher values of *k* (for example *C(20,z)*, *S(20,z)* instead of *C(10,z)*, *S(10,z)*) to analyse the dependence of the results on this value).

Another important characteristic of this method is related with the CPU time and it is shown in Table 2 where the **FIs** are calculated using:

- the continued fraction method
- their expansion in terms of Bessel functions and
- direct integration by quadrature (using the FRESNEL.MTH **DERIVE** file).

If we compare the CPU times, the continued fraction method results faster than the others methods.

In the last column we present a summation up to $k = 20$. This number has been chosen as **an example** in order to guarantee enough precision in the results for our purposes, but for a more complete study we must analyse the dependence of the final results on this maximum value $k = 20$ and also for calculating the CPU time we have to check the use of different procedures to evaluate the Bessel functions $J_{2k+1/2}$ (for more details, see [6, 7, 8]).

x	Continued Fraction Method	time (sec)	FRESNEL_COS(x)	time (sec)	$\sum_{k=0}^{20} J_{2k+1/2}$	time (sec)
4	0.49842626802328	5.8	0.49842603303837	22.9	0.49842626802328	1315
3	0.60571534263134	5.7	0.60572078929793	17.1	0.60572078929789	976.5
2	0.48845565165378	5.9	0.48825340607553	10.1	0.48825340607534	738.9

Table 2. Using a Pentium MMx CPU at 200 Mhz we have calculated the **FIs** with the 3.01 **DERIVE** version with three different methods (see equations 7 and 5).

INTFRE.MTH

```
          √π·(1 - î)·x
#1:  z := ────────────
               2

#2:  A(k, z) := IF(k ≤ 0, 1, IF(k = 1, 1, (2·k - 3)·(2·k - 2)))
                                          2                        2
#3:  B(k, z) := IF(k ≤ 0, 0, IF(k = 1, 2·z  + 1, IF(k = 2, - 2·z  -
                5, B(k - 1, z) - 4)))

#4:  α(k, z) := IF(k = -1, 1, IF(k = 0, 0, IF(k ≥ 1,
                B(k, z)·α(k - 1, z) + A(k, z)·α(k - 2, z))))
                                                      2
#5:  β(k, z) := IF(k = -1, 0, IF(k = 0, 1, IF(k = 1, 2·z  + 1,
                IF(k > 1,B(k, z)·β(k - 1, z) + A(k, z)·β(k - 2, z)))))
```

#6: $H(k, z) := \mathrm{VECTOR}\left(\frac{\alpha(q-1, z)}{\beta(q-1, z)} + \frac{(-1)^{q-1} \cdot \prod_{m=1}^{q} A(m, z)}{\beta(q, z)\cdot\beta(q-1, z)}, q, 1, k\right)$

#7: FC(k, z) := ELEMENT(H(k, z), k)

#8: $\mathrm{ERORFC}(k, z) := \frac{2\cdot z}{\pi^{1/2}}\cdot \hat{e}^{-z^2}\cdot \mathrm{FC}(k, z)$

#9: ERORF(k, z) := 1 − ERORFC(k, z)

#10: $C(k, z) := \mathrm{RE}\left(\frac{1 + \hat{\imath}}{2}\cdot \mathrm{ERORF}(k, z)\right)$

#11: $S(k, z) := \mathrm{IM}\left(\frac{1 + \hat{\imath}}{2}\cdot \mathrm{ERORF}(k, z)\right)$

#12: x := 2

#13: [C(10, z), S(10, z)] = [0.48845568, 0.34342171]

#14: VECTOR([x, C(10, z), S(10, z)], x, [4, 3, 2])

#15: $\begin{bmatrix} 4 & 0.49842626802328 & 0.42051628696593 \\ 3 & 0.60571534263134 & 0.49631788994156 \\ 2 & 0.48845565165378 & 0.34342159705207 \end{bmatrix}$

#16: VECTOR([x, C(15, z)], x, [4, 3, 2])

#17: $\begin{bmatrix} 4 & 0.49842626802328 \\ 3 & 0.60571534263133 \\ 2 & 0.48845565167834 \end{bmatrix}$

References

[1] N.N. Lebedev, *Special Functions and their applications,* Dover Publications, Inc. (1972).

[2] M. Abramowitz and I.A. Stegun, (Eds.), *Handbook of Mathematical Functions,* Dover Publications, Inc. (1972).

[3] Y.L. Luke, *Mathematical Functions and their approximations,* Academic Press, 1975.

[4] H.S. Wall, *Analytic Theory of Continued Fractions,* Chelsea Publishing Company Bronx, N.Y. (1967).

[5] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes. The Art of Scientific Computing,* Cambridge University Press, 1986.

[6] S. Abraham Ibrahim and P. Fernández de Córdoba, *The modified Bessel Function in DERIVE,* Derive Newsletter 32 (1998) 16.

[7] Yu. L. Ratis and P. Fernández de Córdoba, *A code to evaluate (high-order) Bessel functions based on the Continued Fractions method,* Computer Physics Communications 76 (1993) 381.

[8] P. Fernández de Córdoba et al., *A procedure to evaluate Fresnel Integrals based on the Continued Fractions method,* Submitted to *Numerical Algorithms.*

RIVER MEANDER AND ELASTICA

David Halprin, North Balwyn, Australia

The planimetric geometry of River Meanders.

These are all serpentine-like curves, when observed 'in situ', but when extrapolated by mathematicians, one such researcher might write this type of paper.

Let ϕ be the tangential angle, (the direction at location 's'); it is the angle that the path, at a given point, makes with the mean down-path direction.

Let M be the channel length of a meander, so M = s for one wavelength of the meander curve.

Let ω be the maximum angle that the meander takes relative to the general direction of the river; at its maximum it can be as great as angle ϕ. ω is a unique function of the sinuosity, (tightness of bend), and is independent of M. The normal range of ω in degrees is from 0 (zero sinuosity) to 125 for 'gooseneck meanders' at a point of incipient crossing.

$$\omega = 2.2\sqrt{\frac{k-1}{k}} \quad or \quad k = \frac{1}{1-\left(\frac{\omega}{2.2}\right)^2}$$

The sinuosity, k, equals the average of the values of cos ϕ over the range from $\phi = 0$ to $\phi = \omega$. Thus a relationship can be defined between k and ω. An approximate algebraic expression is derived thus:
Relation of ω to sinuosity, k.

$k = \dfrac{M}{\int \cos\phi . ds}$, where $\phi = \omega . \sin\left(\dfrac{2\pi s}{M}\right)$. With an assumed value of ω, values of ϕ at 24 equally spaced intervals of $\dfrac{s}{M}$ were computed. The reciprocal of the average value of cos ϕ equals the sinuosity, k.

$$\omega(\text{radians}) = 2.2\sqrt{\frac{k-1}{k}} \quad \text{or} \quad \omega(\text{degrees}) = 125\sqrt{\frac{k-1}{k}}$$

`k' represents Sinuosity, ratio of path distance to down-valley distance.

Sinuosity, or tightness of bend, is expressed as the ratio of the length of the channel, (s), in a given curve to the wavelength of the curve, (λ).

The von Schelling solution to the River Meander equation demonstrated that the arclength `s' is defined by the following elliptic integral:

$$s = \frac{1}{\sigma} . \int \frac{d\,\phi}{\sqrt{2(\alpha - \cos\phi)}}$$, where α is a constant of integration.

Let $\alpha = \cos\omega$ in which ω becomes the maximum angle, that the path makes from the origin with the mean direction.

Von Schelling (1951) showed that a general condition for the most frequent path for a continuous curve of given length between two points, A & B is when $\int \frac{d\,s}{\rho^2}$ is a minimum, where ds is a unit distance along the path and ρ is the radius of curvature of the path in that unit distance.

Since $\rho = \frac{d\,s}{d\,\phi}$, where $d\,\phi$ is the angle, by which direction is changed in unit distance ds, therefore $\int \frac{d\,s}{\rho^2} = \int \frac{d\,\phi}{d\,s}.d\,\phi$

Basic Identities: $\rho = \frac{d\,s}{d\,\phi}$, $\frac{d\,x}{d\,s} = \cos\phi$, $\frac{d\,y}{d\,s} = \sin\phi$, $\frac{d\,x}{d\,\phi} = \rho.\cos\phi$, $\frac{d\,y}{d\,\phi} = \rho.\sin\phi$

$\phi = \omega.\sin\left(\frac{2\pi s}{M}\right)$	Whewell Type-1 Intrinsic Equation
$\frac{1}{\rho} = \frac{d\,\phi}{d\,s} = \frac{2\pi\omega}{M}\cos\left(\frac{2\pi\,s}{M}\right)$	
$\rho = \frac{d\,s}{d\,\phi} = \frac{M}{2\pi\omega}\sec\left(\frac{2\pi\,s}{M}\right)$	Cesaro Type-1 Intrinsic Equation
$\rho = \frac{M}{2\pi}.\frac{1}{\sqrt{\omega^2 - \phi^2}}$	Euler Type-1 Intrinsic Equation

This equation, which approximates the River Meander Curve, is also known as the Elastica or Lintearia. Analytically, it is a surprise, since we have an angle equated to the sine function of a displacement, a Whewell Intrinsic Equation Type-2. Further, this same equation represents the shape of a bent piece of spring steel, while under tension, since it is a curve of minimum total work in bending. Further still, this sine-generated curve minimises the sum of the squares of the changes in direction, $\sum_{i=1}^{n} \frac{(\Delta\phi_i)^2}{\Delta s_i}$, compared with other curves with the same length, wavelength and sinuosity.

In the case of the River Meander, the curve can be described as a path traced out by a random `walk' of successive moves on a surface of fixed step-lengths, where the direction of each step is random, but with the constraint, that the curve path begins at point A and ends at point B, (fixed end points), in a given number of steps, otherwise the path is `free'. It is called "A most probable random walk". There is a tendency for there to be a constant ratio between the wavelength and the radius of curvature, (approx. 4.7) and also there is a constant sinuosity of a given length of river, this being the ratio of the arclength to the wavelength, ranging from 1.3:1 to 4:1.

To the pure mathematician there is an added bonus or two to studying this curve, since:-

1) It is a study of the sine function in other than a Cartesian situation, where one sees evidence of wavelength and periodicity in a vastly different graphical display.

2) It has a close analogy to the Simple Pendulum, viz:-

$$\rho = \frac{1}{\kappa} = \frac{c}{y} \quad (\text{or}) \qquad \text{Elastica}$$

$$\kappa = \frac{1}{\rho} = \frac{d\,\phi}{d\,s} = \frac{y}{c}$$

Let c = - 1 then

$$\frac{d\,\kappa}{d\,s} = -\frac{d\,y}{d\,s} = \frac{d^2\,\phi}{d\,s^2} = -\sin\phi$$

Remember $\frac{d^2\,\theta}{d\,t^2} = -\sin\theta$ Simple Pendulum

3) The Cartesian representation of the Elastica is an Elliptic Function, an 'Incomplete Elliptic Integral of the First Kind'.

$$\frac{d^2\,\phi}{d\,s^2} = \frac{d\left[\frac{1}{2}\right]}{d\,\phi} = -\sin\phi$$

$$\frac{1}{2}.\left(\frac{d\,\phi}{d\,s}\right)^2 = \cos\phi - \cos\alpha = 2\left(\sin^2\frac{\alpha}{2} - \sin^2\frac{\phi}{2}\right) = 2\left(k^2 - \sin^2\frac{\phi}{2}\right)$$

$$\therefore \frac{d\,s}{d\,\phi} = \frac{1}{2\sqrt{k^2 - \sin^2\frac{\phi}{2}}} \quad \therefore \quad s = \int \frac{d\left(\frac{\phi}{2}\right)}{\sqrt{k^2 - \sin^2\frac{\phi}{2}}}$$

Thus, the inclination of Elastica curves varies with the path length, as one goes along them, as does the inclination of a Simple Pendulum to the vertical vary with the time in the same mathematical manner.

The Elastica can be expressed in other coordinate systems also: -

$$\frac{d\,y}{d\,x} = \frac{x^2}{\sqrt{a^4 - x^4}} \quad \text{or} \quad \frac{d\,y}{d\,x} = \frac{y^2}{\sqrt{b^4 - y^4}} \qquad \text{Cartesian}$$

$$2\,x = \frac{a}{\rho} = \frac{a\,.\,y''}{[1 + (y')^2]^{3/2}} \qquad \text{Hybrid Cartesian- Intrinsic}$$

$$\therefore\; x^2 + C = a\,.\int \frac{d\,y'}{[1 + (y')^2\;]^{3/2}} = \frac{-a\,.\,y'}{\sqrt{1 + (y')^2}}$$

$$\therefore \frac{d\,y}{d\,x} = \frac{x^2 + c}{\sqrt{a^2 - (x^2 + c^2)^2}} \quad \text{or} \quad \frac{d\,y}{d\,x} = \frac{y^2 + d}{\sqrt{b^2 - (y^2 + d^2)^2}} \qquad \text{Q.E.D.}$$

Some of the numerous delightful and varied shapes, generated by varying ω in the Whewell Type-2 Intrinsic Equation of the Elastica curve, are illustrated below. Also, the coefficient of `s' has some significance too.

I have let $\phi = m\cdot\sin s$ in the Derive XM file, which means that $M = 2\pi$ and $\omega = m$.

We have a number of various running curves as well as a stationary figure 8 curve, (m = 2.366) and other stationary ones at m = 5.513 and 8.65 resp. The morphology is fascinating as one sees the changing shapes due to increasing m-values. It starts off moving in a positive direction until it reaches an oscillatory phase then it travels backwards to the next oscillatory phase in all sorts of convolutions, the lobes becoming two rows of circles, then the upper row drops down to level of the lower row, then drops further etc., then back to positive direction etc.

Some variations of the Elastica curve may be seen in the memoirs of William Whewell, cited below, and illustrated with extras in a future paper ELASTIC2.WPF from this writer.

PLOTTING ALGORITHM

Due to the fact that the Elastica has no closed form in Cartesian Coordinates, then the points have to be determined by one of the numerical approximation methods. It appears that Derive for Windows will not attempt to do this numerical integration and plotting point by point but Derive XM will do so if it is 'tricked'. Since it will not do so without definite limits in place, one sets them from 'zero to s' and just to let you know that you are trying out a `no-no' it substitutes the ampersand '@' for s as the arbitrary variable in the expression under the integral sign but retains the 's' in the limit. Still not satisfied it throws out an error message "Dubious Accuracy" and stops plotting until I nominate the range to exclude a guessed-at point of discontinuity.

$$\phi = W(s)\ ,\ \frac{dx}{ds} = \cos\phi = \cos W(s)\ ,\ \frac{dy}{ds} = \sin\phi = \sin W(s)$$

$$x = \int \cos W(s).ds = \int \cos(m.\sin s).ds\ ,\ y = \int \sin W(s).ds = \int \sin(m.\sin s).ds$$

One can determine the points of inflection, being where the curvature is zero having changed from positive to negative at that point.

$$\frac{dy}{dx} = \tan\phi \ \therefore\ \frac{d^2y}{dx^2} = \sec^2\phi.\frac{d\phi}{dx} = \frac{\sec^2\phi}{\rho.\cos\phi} = \frac{\sec^3\phi}{\rho}$$

$$\phi = m.\sin s\ \therefore\ \frac{d\phi}{ds} = \frac{1}{\rho} = \kappa = m.\cos s = 0 \text{ iff } s = (2n+1).\frac{\pi}{2}$$

$$\text{or } \frac{d^2y}{dx^2} = \sec^3\phi.m.\cos s = 0 \text{ iff } \cos s = 0\ \therefore\ s = (2n+1).\frac{\pi}{2}$$

$$\text{where } n = 0,1,2,3,......$$

The gradient at the points of inflection is therefore: -

$$\phi = m.\sin\left((2n+1).\frac{\pi}{2}\right) = \pm m\ \therefore\ \frac{dy}{dx} = \tan m \text{ or } \cot m$$

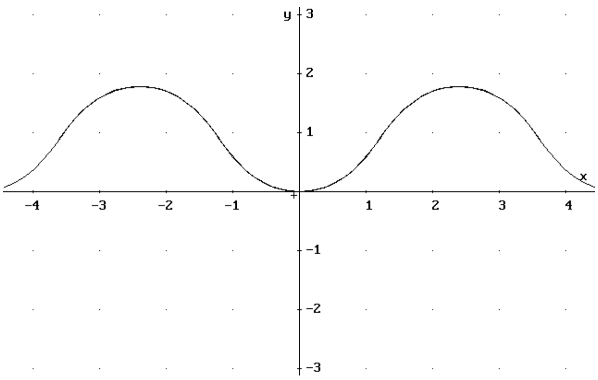
AUTHOR'S COMMENTS

Unfortunately my routine cannot plot in Derive for Windows [1], (it tells me so), yet Derive XM will accept it, albeit a near refusal, since it converts the integral variable 's' to the arbitrary variable '@' yet retains `s' in the limits of the definite integral.

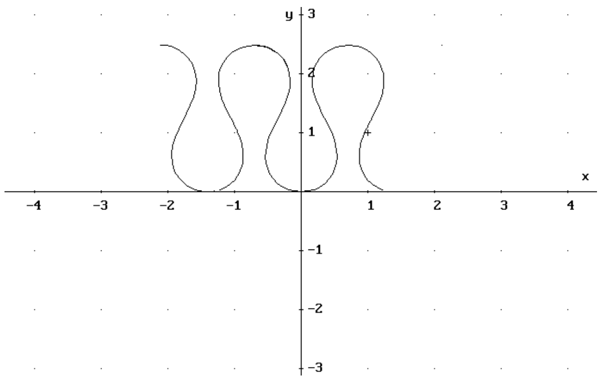
Unfortunately Derive appears not to offer the option of a bunch of particular error trapping routines in case one wants to plot a curve which has point(s) of discontinuity, since this means one has to make a patchwork graph with successive ranges, that do not sweep over those points. Far better for the ardent investigator to be able to opt for a graph-plot that will sweep over those points by having conditional clauses within the program to instruct the point to jump to the next step in those routines that say, for instance, "step by increments of s from 1 to n [as in `VECTOR(u,k,m,n,s)`] and if there is a division by zero (or some other mathematical no-no), then next s."

Another option that would be good to have is for the user to be able to nominate the step size (p = ?) in normal plotting mode; it could be a time saver in many instances.

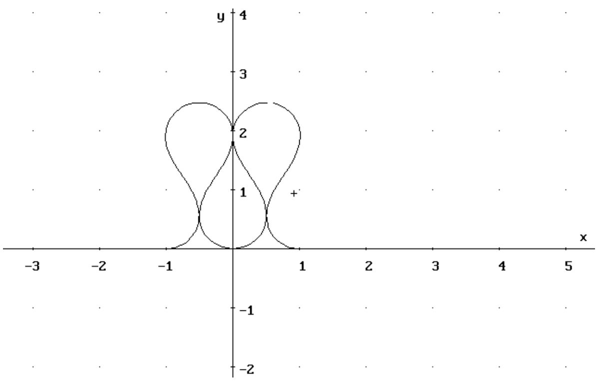
Some graphs:



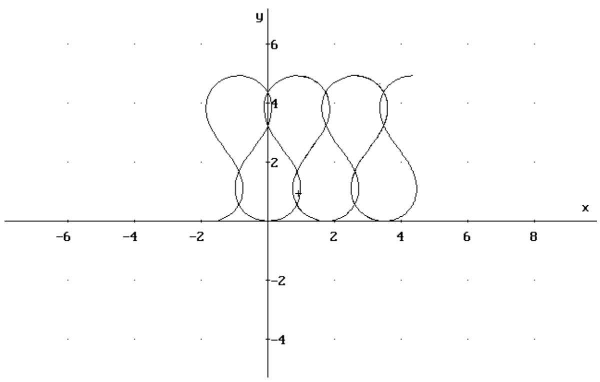
`m = 1`



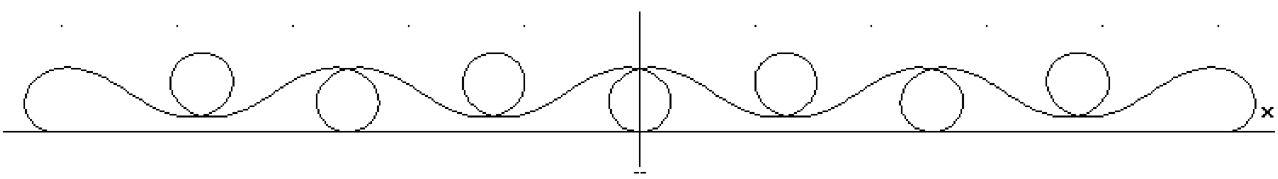
`m = 1.9`



`m = 2.11`

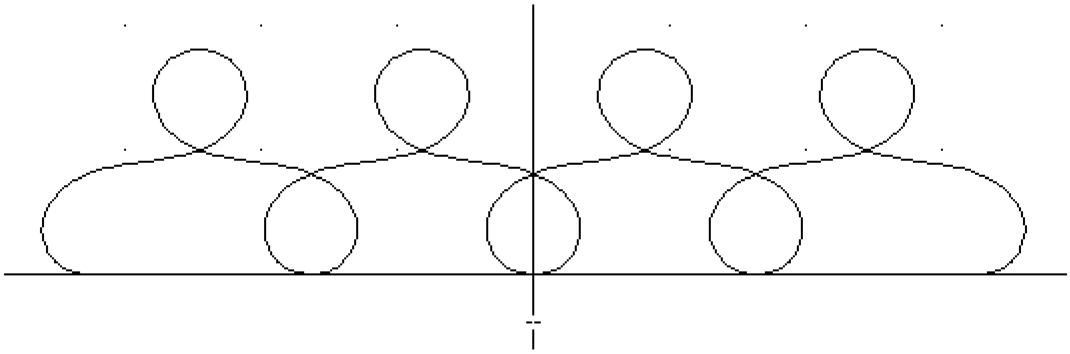


`m = 2.15`

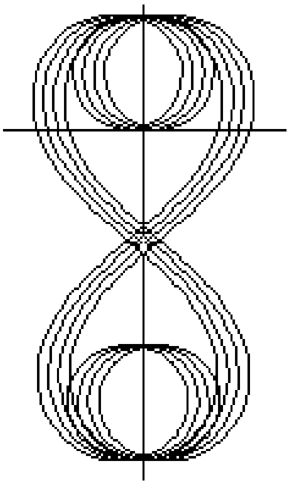


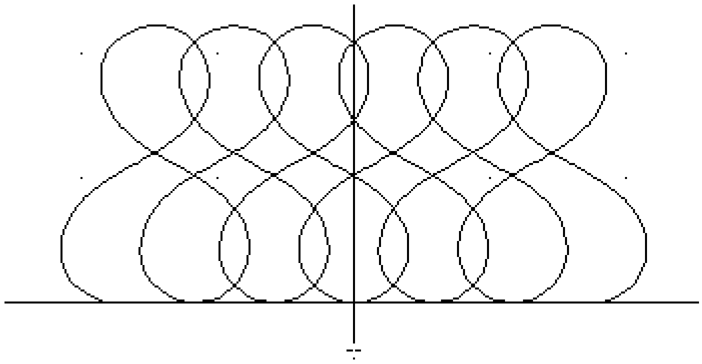
`m = 1.2`

[1] *I reproduced the plots in DERIVE for WINDOWS 4 and 5, and it works. I had much fun experimenting with Dave Halprin´s file MEANDER2.MTH. Josef*



m = 3





m = 5.535

See also the Appendix on page 50, Josef.

BIBLIOGRAPHY

There are disparate references to Sinuosity and Meanders amongst papers on Geomorphology, in particular.

Eagle,A., *The Elliptic Functions As They Should Be*, Galloway & Porter 1958

Langbein, W. & Leopold L., *River Meander - Theory of Minimum Variance*
Geological Survey Professional Paper 422-H 1966

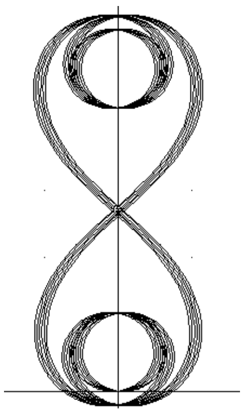
Langbein, W. & Leopold L., *River Meanders,* Scientific American June 1966:

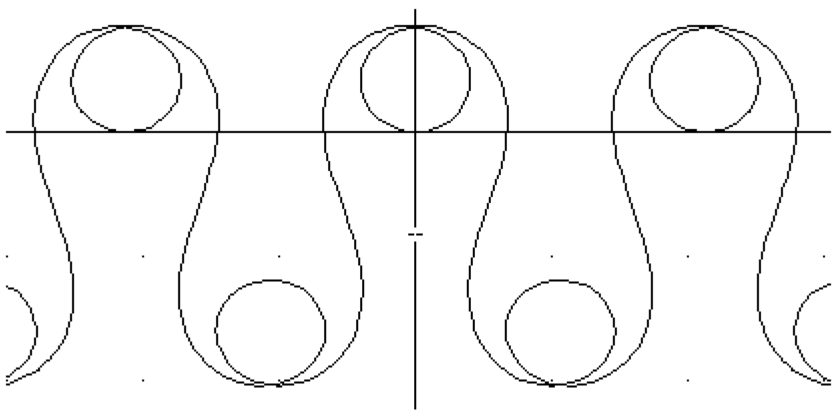
Muses.C.: *Destiny and Control in Human Systems* (Chronotopology), Kluwer Nijhoff Publishing 1983

Scheidegger, *Theoretical Geomorphology*

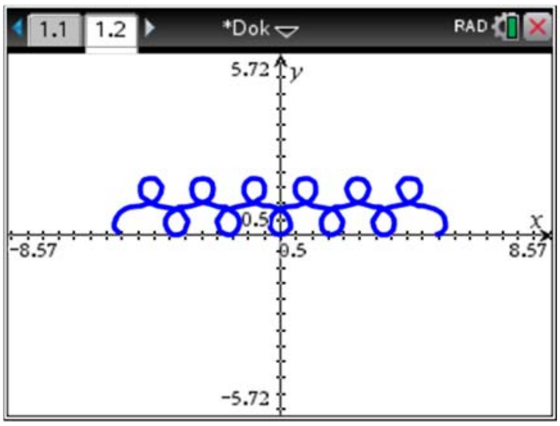
Schelling, H. von, *Most Frequent Random Walks*, General Electric Report 64GL92 (1964)

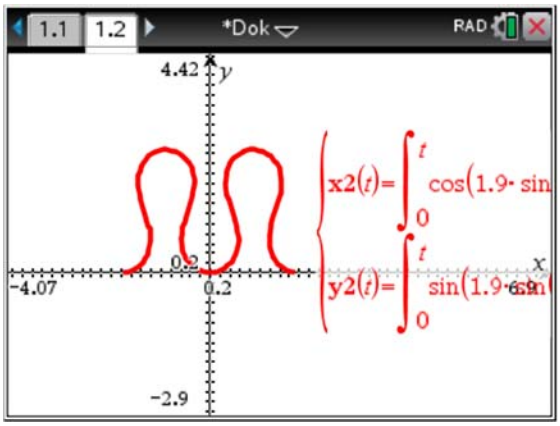
Whewell,W., *On the Intrinsic Equation of a Curve & its Application*, Cambridge Philosophical Transactions
Vol.VIII 1849



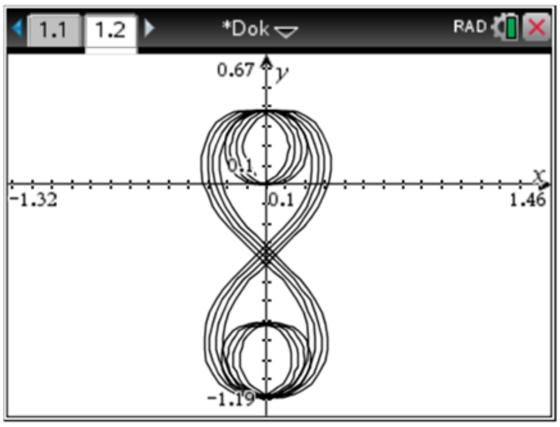


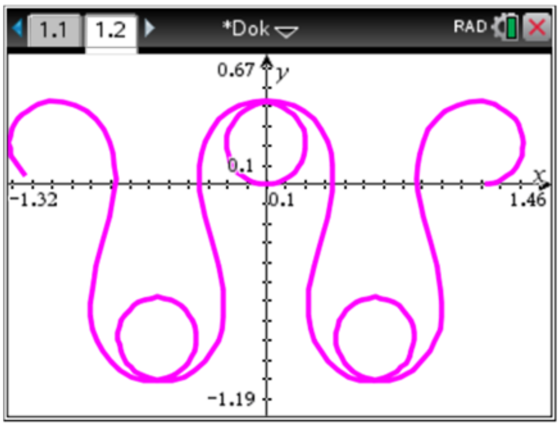
I add some TI-Nspire Meanders (Josef):

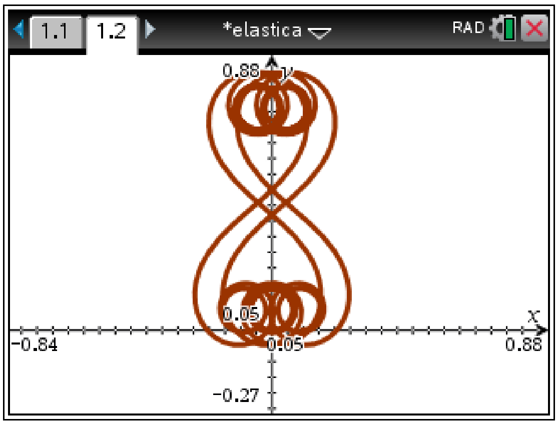


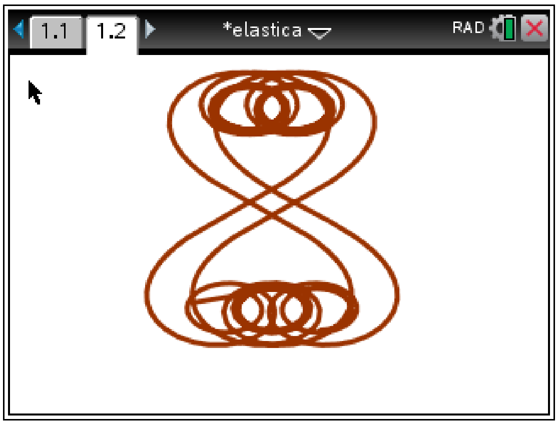


Just when I revised DNL#99 we undertook a hike in Upper Austria's mountains and we found wonderful true Meanders (produced by the Teichl on the Wurzeralm). The water disappears in a height of 1400 m above sea level and appears again after a "travel" through the mountain down in the valley.









Vigenére Cipher using DfW5

Rüdeger Baumann, Celle, Germany

The - nowadays - so called **Vigenére Cipher** was published by the French diplomat Blaise de Vigenére (1523-1596) in 1586. He had studied the several papers about cryptography (Trithemius and others) and met the pope´s leading cryptographers and cryptoanalysts in Rome. He developed various systems of coding by his own. In his famous book *Traicté des chiffres* (1586) he collected the cryptographic knowledge of his time.

The procedure works as follows. At first, we have to set up the **Vigenére Square** (this is Trithemius´ *tabula recta*). Then we choose a keyword, eg. the word JOSEF, we write down the plaintext (= message to be encoded) and put the keyword above it as often as necessary. See an example:

```
JOSEFJOSEFJOSEFJOSEFJOSEFJO
IWISHALLDUGMEMBERSAFINEFALL
```

Plain	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
2	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
3	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
12	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
14	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
15	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
16	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
17	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
26	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

The letter of the keyword which stands above the plaintext´s letter indicates the particular cipher alphabet within the Vigenére square, which is now used to encode the letter according to Caesar´s method. To encode the I we have to take the row starting with J (9^{th} row) and take the letter in column I, which gives R, thus I $\rightarrow$ R.

If we have a numerical representation of plaintext and keyword (A = 0, B = 1, .., Z = 25) we can use modulo arithmetic to perform encoding (and decoding, of course). We tie up the plaintext with the keyword sequence by an addition modulo 26. The keyword sequence consists of a repeated sequence of characters $c_1, c_2, c_3, ..., c_n$ and we find

$$y_i = \mathrm{mod}(c_i + x_i, 26) \quad \text{for } c_{n+i} = c_i \quad (i = 1,2,3,....)$$

*DfW*5 allows working with strings, so let´s use these new features:

VIGENERE.DFW

```
encodes(text) := VECTOR(k - 65, k, NAME_TO_CODES(text))

decodes(numbers) := CODES_TO_NAME(VECTOR(k + 65, k, numbers))
```

$$f(c, x) := \mathrm{VECTOR}(\mathrm{MOD}(c_{\mathrm{MOD}(i - 1, \mathrm{DIM}(c)) + 1} + x_i, 26), i, 1, \mathrm{DIM}(x))$$

```
encrypts(key, text) := decodes(f(encodes(key), encodes(text)))

kW := JOSEF

kT := IWISHALLDUGMEMBERSAFINEFALLANDANDIAMLOOKINGFORWARDTOMEETINGYOUATANYOC
CASION
```

Now we can hide our message calling `encrypts(keyword,plaintext)`:

```
message := encrypts(kW, kT)
RKAWMJZDHZPAWQGNFKEKRBWJFUZSRIJBVMFVZGSPRBYJTAKSVICCEIJCWFKDXISXFWMGGHJGASS
```

Calculating the letters' frequencies, we can easily see how the typical and characteristical frequency distribution of a language is obscured by the polyalphabetic encoding. We define:

$$h(list, number) := \sum_{i=1}^{\mathrm{DIM}(list)} \mathrm{IF}(list_i = number)$$

```
distrib(text) := VECTOR([CODES_TO_NAME(k + 65), h(NAME_TO_CODES(text),
                 k + 65)], k, 0, 25)`
```

and compare `distrib(kT)` with `distrib(message)`:

```
distrib(kT)
```

$$\begin{bmatrix} A & B & C & D & E & F & G & H & I & J & K & L & M & N & O & P & Q & R & S & T & U & V & W & X & Y & Z \\ 10 & 1 & 2 & 4 & 5 & 3 & 3 & 1 & 7 & 0 & 1 & 5 & 4 & 7 & 7 & 0 & 0 & 3 & 3 & 3 & 2 & 0 & 2 & 0 & 2 & 0 \end{bmatrix}$$

```
distrib(message)
```

$$\begin{bmatrix} A & B & C & D & E & F & G & H & I & J & K & L & M & N & O & P & Q & R & S & T & U & V & W & X & Y & Z \\ 4 & 3 & 3 & 2 & 2 & 5 & 5 & 2 & 4 & 6 & 5 & 0 & 3 & 1 & 0 & 2 & 1 & 4 & 6 & 1 & 1 & 3 & 5 & 2 & 1 & 4 \end{bmatrix}$$

10 times A and 7 times N change to 4 times A and 1 N.

I wanted to find a function to count specified letters, but unfortunately the following function does not work as intended:

$$\text{h1(text, letter)} := \sum_{i=1}^{\text{DIM(text)}} \text{IF(IDENTICAL?(text}_i\text{, letter))}$$

```
distrib1(text) := VECTOR([l, h1(text, l)], l, [A, E, I, O, U])`
```

$$\text{distrib1(kT)} = \begin{bmatrix} A & E & I & O & U \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

I add a little problem for deciphering (the language is - a bit old fashioned - English, the keyword has only 3 letters:

KQGLPRWLBOV-
NFLUSDRFWFZIZWKUWMXTGWGEYSVQCVRWJLCVRWEUSBPSZ-
RANBJI-
AQVNEMJZIGXWEGZQWZBJLRYZRWTRLURKKEWEGZMRMXNHMTDM
GEMYGWFWBUQBBFOHWEVLPJAVRUIYDMFL-
BUGCGZIGLPVFONDMTOPVUPVKBUAVAWAGLPVFMBJUVFM

References:

Singh, S.: *Geheime Botschaften. Die Kunst der Verschlüsselung von der Antike bis in die Zeiten des Internet.* München: Hanser, 2000

Singh, S.: *The Code Book. The Science of Secrecy from Ancient Egypt to Quantum Cryptography.* London: Fourth Estate, 1999

I changed Rüdeger´s `distrib1`*-function and put the list of letters to be counted into the parameter list - and it works. Don´t forget to write the letters under quotes:* `["A","E","I","O","U"]`*. Josef*

```
distrib2(text, list) := VECTOR([l, h1(text, l)], l, list)`

distrib2(kT, [A, E, I, O, U])
```

$$\begin{bmatrix} A & E & I & O & U \\ 10 & 5 & 7 & 7 & 2 \end{bmatrix}$$

$$\text{distrib2(message, [A, E, I, O, U])} = \begin{bmatrix} A & E & I & O & U \\ 4 & 2 & 4 & 0 & 1 \end{bmatrix}$$

And if you would like to check your skills as cryptanalyst:

$$\text{g(c, x)} := \text{VECTOR(MOD(x}_i - \text{c}_{\text{MOD(i − 1, DIM(c)) + 1}}\text{, 26), i, 1, DIM(x))}$$

```
decrypts(key, text) := decodes(g(encodes(key), encodes(text)))

decrypts(sW, message)

IWISHALLDUGMEMBERSAFINEFALLANDANDIAMLOOKINGFORWARDTOMEETINGYOUATANYOCCASION
```

It is not too difficult to implement the encoding and decoding routine on TI-Nspire:

```
Define vigenere(kw,kt,s)=
Prgm
Local d,i,n,vn,kw1,nw,vtxt
kw1:={ }
For i,1,dim(kw)
  nw:=ord(mid(kw,i,1))−65
  kw1:=augment(kw1,{nw})
EndFor
vtxt:=" "
d:=dim(kt)
For i,1,d
  n:=ord(mid(kt,i,1))−65
  n:=mod(n+s· kw1[mod(i−1,dim(kw))+1],26)+65
  vtxt:=vtxt&char(n)
EndFor
vtxt
EndPrgm
```

```
1.1   *vigenere   RAD
vigenere("NSPIRECAS","ITISNOTASECRE▸
VLXAESVAKRUGMKXJALGZXADIVHGQ
UPVSIFEUVHWMIIF
                                Done
vigenere("NSPIRECAS","VLXAESVAKRU(▸
ITISNOTASECRETTHATTHISMETHODCA
NBEDECIPHERED
```

The third argument of the function gives the direction: 1 for encoding and -1 for decoding, Josef

```
                                                              Done
vigenere("NSPIRECAS","VLXAESVAKRUGMKXJALGZXADIVHGQUPVSIFEUVHWMIIF",-1)
                          ITISNOTASECRETTHATTHISMETHODCANBEDECIPHERED
                                                              Done
vigenere("JOSEF","IWISHALLDUGMEMBERSAFINEFALLANDANDIAMLOOKINGFORWARDTOMEE▸
RKAWMJZDHZPAWQGNFKEKRBWJFUZSRIJBVMFVZGSPRBYJTAKSVICCEIJCWFKDXISXFWMGG
HJGASS
                                                              Done
distr("IWISHALLDUGMEMBERSAFINEFALLANDANDIAMLOOKINGFORWARDTOMEETINGYOUAT▸
```

"A"	"B"	"C"	"D"	"E"	"F"	"G"	"H"	"I"	"J"	"K"	"L"	"M"	"N"	"O"	"P"	"Q"	"R"	" ▸
10	1	2	4	5	3	3	1	7	0	1	5	4	7	7	0	0	3	

```
                                                              Done
distr("RKAWMJZDHZPAWQGNFKEKRBWJFUZSRIJBVMFVZGSPRBYJTAKSVICCEIJCWFKDXISXFW▸
```

"A"	"B"	"C"	"D"	"E"	"F"	"G"	"H"	"I"	"J"	"K"	"L"	"M"	"N"	"O"	"P"	"Q"	"R"	" ▸
4	3	3	2	2	5	5	2	4	6	5	0	3	1	0	2	1	4	

```
                                                              Done
```

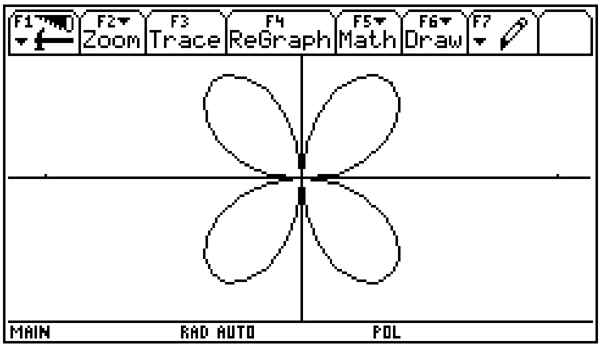
Algebraische und Transzendente Kurven (Folge 10)

Rhodoneen (Rhodoneas = Rosenkurven)

Thomas Weth, Germany

Ähnlich wie Bodo Habenicht (vgl. Folge 9) versuchte auch Guido Grandi, ein Zeitgenosse von Leibniz, die Gestalt von Blüten mit Hilfe mathematischer Kurven zu beschreiben. Ebene Kurven des Typs r = a sin (b ß) nannte er wegen ihrer Gestalt Rhodoneen, also Rosenkurven. "Die wunderschönen Eigenschaften dieser Kurven wurden von Grandi dem Leibniz in zwei Briefen mitgeteilt, die auf den Dezember 1713 zurückgehen, wurden aber Allgemeingut erst zehn Jahre später, infolge einer der Gesellschaft der Wissenschaften zu London eingereichten Abhandlung; ihre vollständige Theorie wurde dann von Grandi selbst im Jahre 1728 in einem besonderen Werkchen dargelegt" (Loria S. 298).

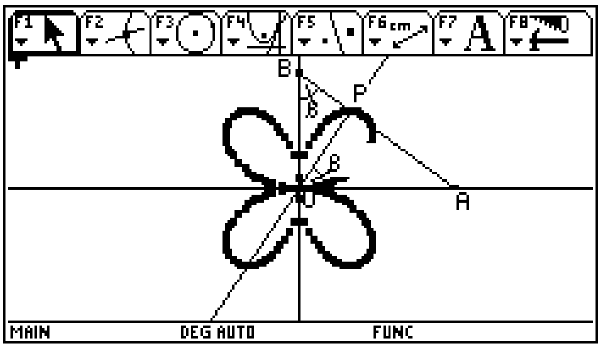
Im Folgenden beschränken wir uns auf die Diskussion eines *vierblättrigen Rosenblatts,* der *Rosette* (in obiger Polardarstellung erhält man sie für b=2). Die Formenvielfalt, die sich mit anderen b-Werten ergibt, lässt sich mit Derive einfach darstellen und bleibe dem Leser überlassen.



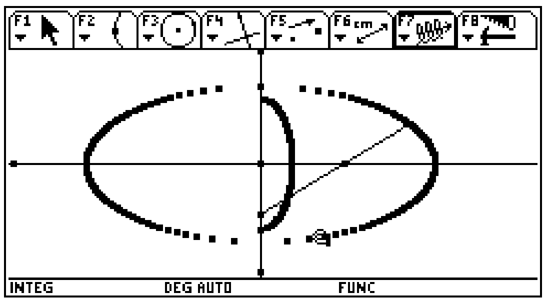
Similar to Bodo Habenicht Guido Grandi a contemporary of Leibniz tried to describe the form of flowers using mathematical curves. He called plane curves of form r = a sin (b ß) Rhodoneas = rose curves. In the following we restrict to a discussion of the Quadrifolium - with b = 2. Varying b gives a rich variety of forms, which can easily be investigated by the reader by his own.

Konstruktion und Herleitung der Kurvengleichung

Eine Möglichkeit, die Rosette punktweise zu konstruieren ergibt sich aus der Modifikation einer Ellipsenkonstruktion als Gleitkurve:
Eine Strecke AB = a gleite mit ihren Endpunkten auf den Achsen: Fällt man dann von O auf AB das Lot OP, so beschreibt P eine Rosette.



(Anm.: Eine Ellipse ergibt sich, wenn man einen festen Punkt P auf der Strecke AB betrachtet, während AB auf den Achsen gleitet.)

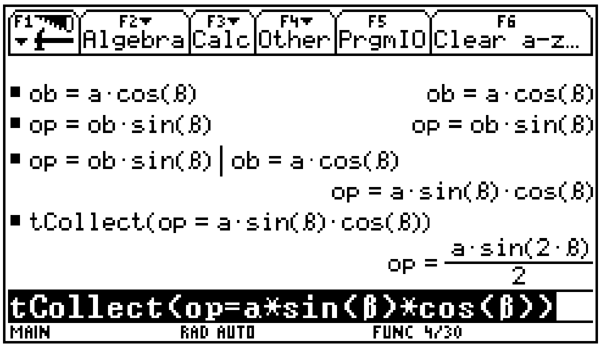


Among several ways of constructing the Quadrifolium one can apply a modification of constructing an ellipse as gliding curve:

Segment AB = a glides with its endpoints on the axes. The intersection point of the perpendicular line to AB from O gives one point of the curve.
Comment: the locus of a point on the segment results in an ellipse. (That is the construction I learnt in school under the name „Papierstreifenkonstruktion“. Josef)

In der nebenstehenden Abbildung ist die Herleitung der Kurvengleichung, die sich mit obiger Konstruktion ergibt, dargestellt. Man erhält also die Polardarstellung der Rosetten zu:

$$r = \frac{a}{2}\sin(2ß).$$



The screen shot shows the derivation of the curve´s equation, resulting from the construction above. One obtains the polarform of the Quadrifolium

$$r = \frac{a}{2}\sin(2ß)$$

Die algebraische Kurvengleichung der Rosette erhält man wie üblich durch Umwandeln der Polardarstellung in kartesische Koordinaten mit x=r cos ß und y=r sin ß, wobei $r^2=x^2+y^2$.

Damit folgt: $\sqrt{x^2+y^2} = r = a\cos ß \sin ß = a\frac{x}{\sqrt{x^2+y^2}}\frac{y}{\sqrt{x^2+y^2}}$ und daraus:

$(x^2+y^2)^3 = a^2x^2y^2$, die **algebraische Kurvengleichung der Rosette**.

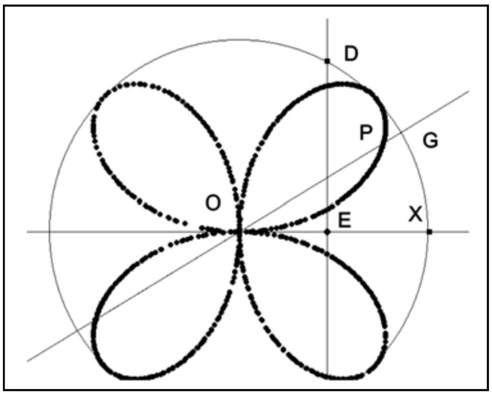
To find the algebraic equation of the curve follow the usual way changing from polar to Cartesian form: x=r cos ß *and* y=r sin ß *with* $r^2=x^2+y^2$, *which leads immediately to*

$\sqrt{x^2+y^2} = r = a\cos ß \sin ß = a\frac{x}{\sqrt{x^2+y^2}}\frac{y}{\sqrt{x^2+y^2}}$, *thus*

$(x^2+y^2)^3 = a^2x^2y^2$, *is the* ***algebraic equation of the Quadrifolium (=4-petalled rose).***

Weitere Konstruktion und Quadratur

Direkt aus der Polardarstellung $r = \frac{a}{2}\sin(2ß)$ ergibt sich eine weitere einfache Konstruktion:

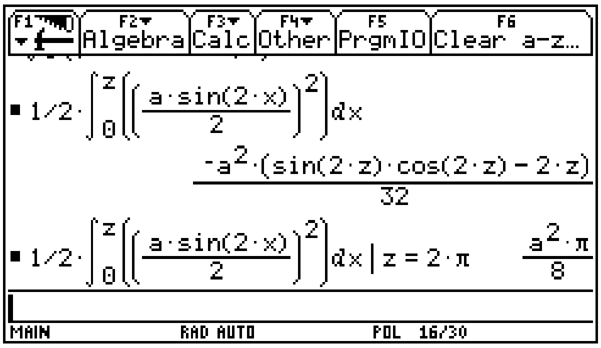


Gegeben sei ein Kreis mit Radius $\frac{a}{2}$. Zunächst halbiert man einen Mittelpunktswinkel ∠XOD und erhält die Halbgerade OG. Bezeichnet man das Maß von ∠XOG mit *ß*, so ist |∠XOD| = 2*ß* und die Lotstrecke DE von D auf die Gerade OX hat die Länge $\frac{a}{2}\sin(2ß)$. Trägt man also die Länge von DE auf der Halbgeraden OG ab, erhält man einen Kurvenpunkt P.

Directly from the polar form $r = \frac{a}{2}\sin(2ß)$ *one can obtain another easy construction of the points:*

Given is a circle with radius r = a/2. The angle bisector of ∠XOD gives the ray OG. If ∠XOG = ß then |∠XOD| = 2ß. E is pedal point of the vertical line through D and segment ED has length $\frac{a}{2}\sin(2ß)$*. Measurement transfer of DE on ray OG gives a point P, which is a point of the curve.*

Zu einem interessanten Ergebnis kommt man, wenn man den von der Rosette eingeschlossenen Flächeninhalt berechnet: Allgemein gilt für den eingeschlossenen Flächeninhalt $A = \frac{1}{2}\int_{=0}^{2\pi} r(ß)^2\, d$. Mit $r = \frac{a}{2}\sin(2ß)$ ergibt sich: $A = \frac{a^2\pi}{8}$. Damit ist der Inhalt der Rosette genau halb so groß wie der Inhalt ihres Umkreises: ein merkwürdiges Ergebnis.



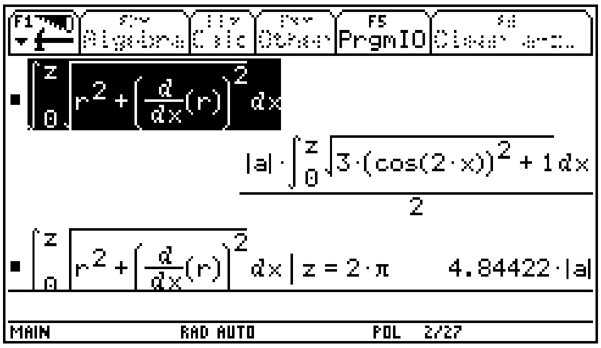
We achieve an interesting result calculating the area enclosed by the four leaves. Generally, the area is given by

A = $\frac{1}{2}\int_{=0}^{2\pi} r(ß)^2\, d$ *. With* $r = \frac{a}{2}\sin(2ß)$ *we receive A =* $\frac{a^2\pi}{8}$ *The area is half of the circumscribed circle: a remarkable result, indeed.*

Für die Länge der Rosette erhält man kein so schönes Ergebnis. Der TI-92 liefert gemäß

$l = \int\sqrt{r()^2 + r'()^2}\, d$ für die Bogenlänge einer Kurve in Polarkoordinaten:

$l \approx 4{,}84$ a.



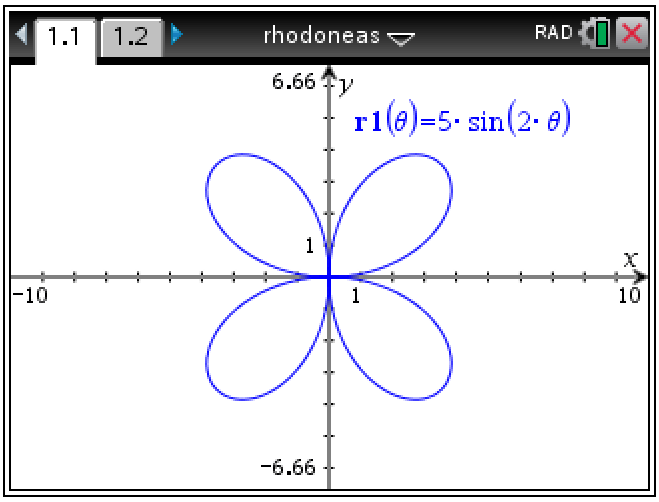
Abschließend sei darauf hingewiesen, dass sich die Rosette auch als Rollkurve (Hypotrochoide) und mit Hilfe von Gelenkmechanismen erzeugen lässt. Auf ihre enge Verwandtschaft zur Astroide (Sternkurve) wird in der nächsten Folge des Kurvenlexikons eingegangen.

The arc length of this "flower" is not such nice. The TI-92 returns according to l = $\int\sqrt{r()^2 + r'()^2}\, d$ *for the arc length of a curve given in parameter form:*

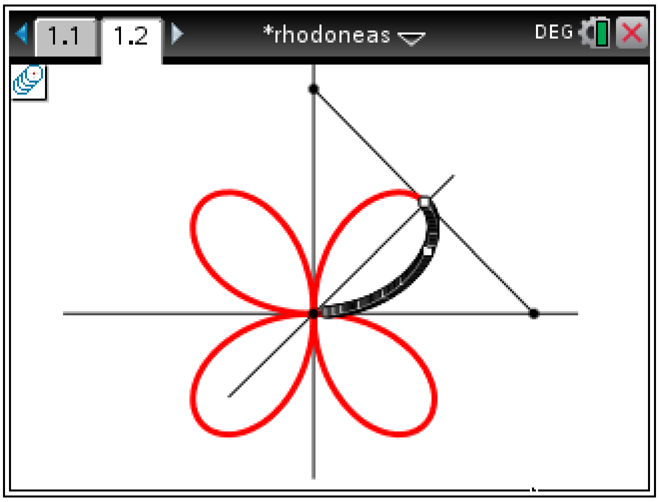
l ≈ 4,84 a.

At the end I´d like to point out that the quadrifolium can be produced as a hypotrochoid (a circle with radius 1 rolls in another one with radius 4) or using a linkage. It is very close related to the astroid - the star curve.

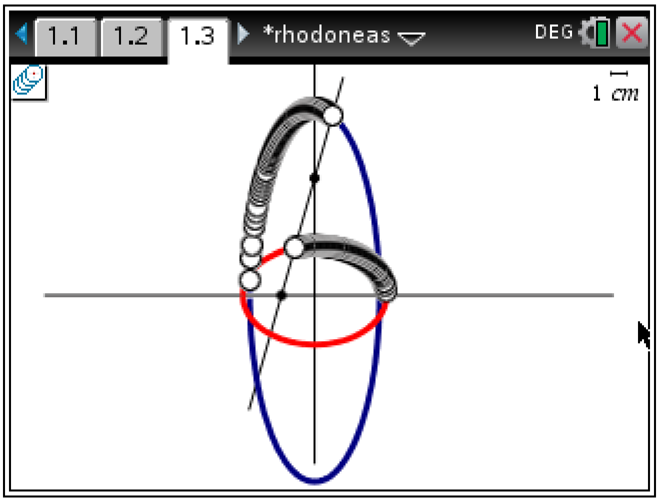
This is the TI-Nspire-Quadrifolium



Polar form representation



Tracing the pedal point and plotting its locus



Tracing and plotting the ellipse

Winding Numbers and Area of Nonconvex Polygons

Stefan Welke, Spwelke@aol.com
In der Wehrhecke 29, D-53125 Bonn, Germany

May 10, 1999

Abstract

The winding number for closed polygons in the Euclidean plane allows to decide if a point is in the interior or exterior of a closed polygon. Polygons need not be convex and self-intersections are allowed. A simple change in the definition of the winding number function allows to compute the area of arbitrary closed polygons. This area function generalizes to three dimensions and allows the computation of the volumina of arbitrary closed polyhdra.

1 Introduction

Let $A_1, A_2, ..., A_n$ be vertices of a closed polygon $\mathcal{P}$ in a plane. The interior problem for the polygon $\mathcal{P}$ is to decide, if a point P is in the interior of the closed polygon or not. The purpose of this paper is to settle the nonconvex case in the plane. A simple change of the resulting formula will provide us with a function that computes the area of an arbitrary convex or nonconvex polygon.

If $\mathcal{P}$ is convex and has anticlockwise orientation, then the interior problem can be solved in the following way: A point P is in the interior of the polygon if and only if P is lying to the left of each straight line containing an edge of the polygon. This condition leads to a system of n linear inequalities, where n is the number of edges.

We now remind the topological concept of the winding number of a closed continuous curve γ with respect to a point P in the plane. Since our point of interest are closed polygons we restrict ourselves to that case. Consider figure 1: The sum of the oriented angles $\measuredangle z_1pz_2 + \measuredangle z_2pz_3 + \measuredangle z_3pz_4 + \measuredangle z_4pz_5 + \measuredangle z_5pz_6 + \measuredangle z_6pz_1$ equals 2π if P is in the interior of the closed anticlockwise

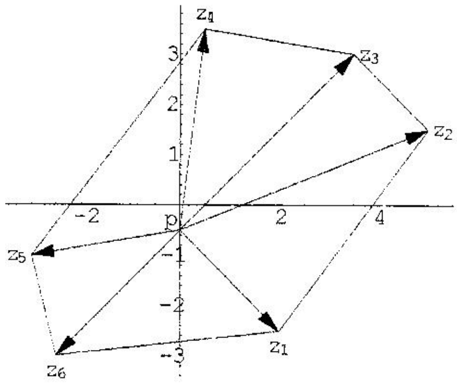


Figure 1:

oriented polygon and 0 if P is in the exterior. If the polygon had clockwise orientation and P is in the interior, then this sum were -2π and 0 otherwise. This sum equals $\pm 2\pi$ for any interior point even if the polygon is not convex, because backward turns of the vector $\overrightarrow{pz_k}$ are counted with opposite sign. If a closed polygon surrounds a point P twice, then the sum of the oriented angles is $\pm 4\pi$, according to the orientation. The use of complex numbers allows a very elegant

Definition 1 *Let $\mathcal{P} := \{z_0, z_1, ..., z_n\}$ be a closed polygon in the complex plane with $z_0 = z_n$, and let p be a point not lying on any edge of the polygon, then the number*

$$w(p, \mathcal{P}) := \frac{1}{2\pi} \sum_{k=1}^{n} Arg(\frac{z_k - p}{z_{k-1} - p}) = \frac{1}{2\pi} \sum_{k=1}^{n} Arg((z_k - p)\overline{(z_{k-1} - p)}) \qquad (1)$$

is called the winding number of p with respect to the polygon $\mathcal{P}$.

$w(p, \mathcal{P})$ is always an integer and counts how many times the closed polygon $\mathcal{P}$ surrounds the point p in the positive or negative sense. Note that the winding number is not defined for points on the boundary of the polygon.

The winding number settles the two-dimensional interior problem: A closed polygon divides the complex plane into a finite set $\{U_1, U_2, ..., U_m\}$ of bounded simply connected regions and one unbounded region U_∞. We call the union $\bigcup_{k=1}^{m} U_k$ the interior and U_∞ the exterior of the polygon. Then a point is in the interior of a polygon $\mathcal{P}$ if and only if $w(p, \mathcal{P}) \neq 0$.

We call a polygon$\mathcal{P}$ *simply closed* if and only if $w(p, \mathcal{P}) = 1$ for all interior points $p \in \mathcal{P}$. We next replace the expression $Arg((z_k - p)\overline{(z_{k-1} - p)})$ in the definition of winding number by $\text{Im}((z_k - p)\overline{(z_{k-1} - p)})$ and drop the factor

π in the denominator. The resulting formula computes modulo the sign the area of a simply closed polygon, no matter which point p is chosen. Thus we can define the following area function with $p = 0$:

$$AREA(\mathcal{P}) := \left| \frac{1}{2} \sum_{k=1}^{n} \operatorname{Im}(z_k \cdot \overline{z_{k-1}}) \right|$$

The explanation is simple: $\frac{1}{2}\operatorname{Im}(z_k \cdot \overline{z_{k-1}})$ computes modulo the sign the area of the triangle spanned by the vectors z_{k-1} and z_k. The sign depends on the orientation of the pair (z_{k-1}, z_k). The following figure demonstrates the way this formula works:

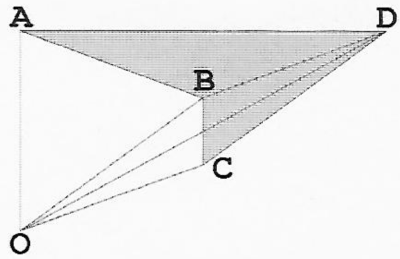


Figure 2

The areas of the triangles OCD and ODA have the same sign opposite to the signs of the areas of OCB and OBA. Thus the area of the shaded polygon $ABCD$ remains.

2 Computation of winding number and area with DERIVE

```
#1: WN(p,poly) :=
    SUM(PHASE((poly SUB k - p)*[1, #i]*CONJ((poly SUB(k-1) - p)*[1,#i])),
                 k, 2, DIMENSION(poly))/2/pi
#2: AR1(poly) :=
    ABS(SUM(IM(poly SUB k *[1,#i]*CONJ(poly SUB(k-1)*[1,#i])),
                 k, 2, DIMENSION(poly)))/2
```

This is the raw form of the two functions. In the present form the first and last element of the vector that represents the polygons must coincide, e.g. [a,b,c,d,a]. An equivalent form of the area function that uses determinants instead of the imaginary part is:

```
#3: AR2(poly) :=
    ABS((DET([poly SUB k, poly SUB(k - 1)]), k, 2, DIMENSION(poly)))/2
```

Now follows the final version of the area function, and it is no longer required that the first and last element of the polygon vector coincide:

```
#4: AREA(poly):=AR2(APPEND(poly, [poly SUB 1]))
#5: AREA([[1, 1], [-1, 1], [-1, -1], [1, -1]])        ( Simplify )
#6: 4
```

Note that the winding number is not defined for points lying on the boundary of the polygon. Thus we must check the point p in consideration for its position relative to the boundary. The next two functions provide this check.

```
#7: PTINLINE(p,v):=
    IF([0]<=RHS((SOLVE(t*v SUB 1+(1 - t)*v SUB 2 = p,t))SUB 1)<=[1],0,1,1)
#8: PTONBOUNDARY(p,v):=
    PROD(PTINLINE(p,[v SUB k,v SUB(k+1)]),k,1,DIMENSION(v)-1)
```

Now we are prepared for the definition of the winding number function:

```
#9:  CLOSE(v) := APPEND(v,[v SUB 1])
#10: WINDING_NUMBER(p, w):=
     IF(PTONBOUNDARY(p,CLOSE(w))=1,WN(p,CLOSE(w)),
             ''notdefined'',''notdefined'')
```

Let us consider an example:

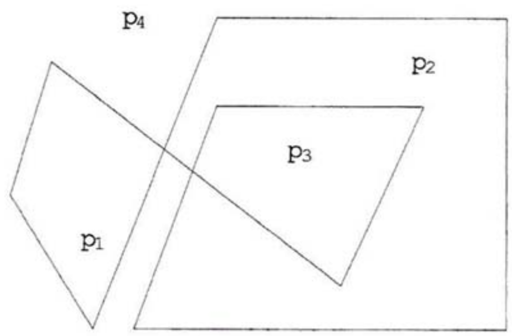


Figure 3

```
#11: poly:=
     [[0,4],[7,4],[7,-3],[-2,-3],[0,2],[5,2],[3,-2],[-4,3],[-5,0],[-3,-3]]
#12: [p1:=[-3,-1],p2:=[5,3],p3:=[2,1],p4:=[-2,4]]
```

```
#13: WINDING_NUMBER(p1,poly)      (Simplify)
#14: 1
#15: WINDING_NUMBER(p2,poly)      (Simplify)
#16: -1
#17: WINDING_NUMBER(p3,poly)      (Simplify)
#18: -2
#19: WINDING_NUMBER(p4,poly)      (Simplify)
#20: 0
#21: WINDING_NUMBER(0.3*[0,4]+0.7*[7,4],poly)      (Simplify)
#22: notdefined
```

3 Generalization to three dimensions

The winding number as a topological quantity can be generalized to higher dimensions. The special case of winding number of a point p relative to a polyhedron $\mathcal{P}$ in $\mathbb{R}^3$ has been treated with elementary geometric means by the author using *Mathematica*. The main point here is to find a substitute for the Arg resp. PHASE function, which measures the spatial angle of the faces of a polyhedron. Although all considerations are elementary the details are numerous and the definition of the corresponding winding number function cannot be done within two or three lines. But it is fairly easy to generalize the AREA function to compute the volume of arbitrary polyhedra in $\mathbb{R}^3$.

Consider a convex polyhedron $\mathcal{P}$ and a point P in the interior of $\mathcal{P}$. Then the volume of the polyhedron is the sum of the volumina of the pyramids with common vertex P and bases which are the faces of the polyhedron. We dissect each face into triangles to compute the volume of each single pyramid as the sum of volumina of three-sided pyramids. Since the volume of a three-sided pyramid is six times the determinant of three positively oriented vectors that span such a pyramid, we arrive at the following volume function for polyhedra with $P = (0, 0, 0)$ for simplicity:

```
#23: PYRAMID(v):=
     SUM(DET([v SUB 1,v SUB(k - 1),v SUB k]),k,3,DIMENSION(v))
#24: VOLUME(poly):=ABS(SUM(PYRAMID(poly SUB k),k,1,DIMENSION(poly)))/6
```

It is necessary for the volume function to give all faces a *coherent orientation*, i.e.: Two adjacent faces that share a common edge must induce opposite orientations on that common edge as for example [A, B, C, D] and [F, D, C, G, H] with common edge [C,D] resp. [D,C]. In this case a thorough geometric discussion shows that this VOLUME function works with convex and nonconvex polyhedra as well as with points P not in the interior of the polyhedron. This is because the sign of the determinant function resembles the orientations of the pyramids. Thus we can always choose $P = (0, 0, 0)$. We close with an example:

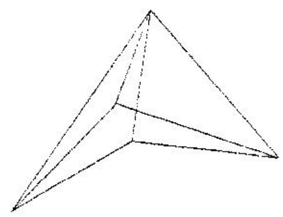


Figure 4

```
#25: [a:=[1,1,-1],b:=[-0.5,0.5,-1],c:=[-1,-1,-1],d:=[-1,1,-1],
      s:=[0,0,1]]
#26: polyhedron:=[[a,b,c,d],[d,c,s],[c,b,s],[b,a,s],[a,d,s]]
#27: VOLUME(polyhedron)       (Simplify)
#28: 2/3
```

Experiments Using CBL/CBR and the TI-92 – New Conceptions for Teaching Natural Science

Heinz-Dieter Hinkelmann, Korneuburg, Austria

(continued from DNL#38)

3. 3 Newton´s law of cooling

Newton stated that the rate at which a warm body cools is approximately proportional to the temperature difference between the temperature of the warm object and the temperature of its surroundings. From this simple assumption he showed that the temperature change is exponential in time and can be predicted by

$$T - C = (T - T_o)\, e^{-kt}$$

(T ... body´s temperature at some time, C ... surrounding temperature,

T_o ... body´s temperature when t = 0, t ... time)

In this exercise you attempt to verify the mathematical model developed by Newton.

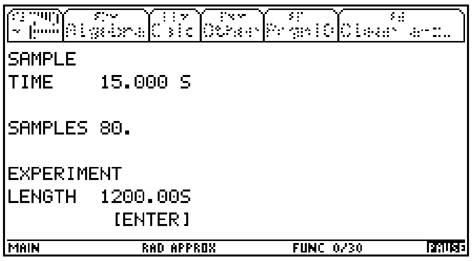
Equipment required

- CBL unit, TI-92 calculator with unit to unit link cable
- TI temperature probe
- 35-mm film canister and hot water

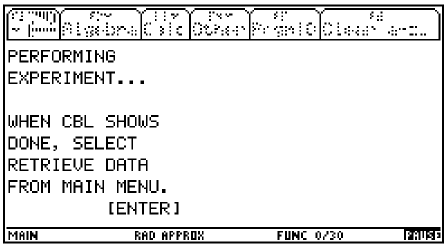
Instructions

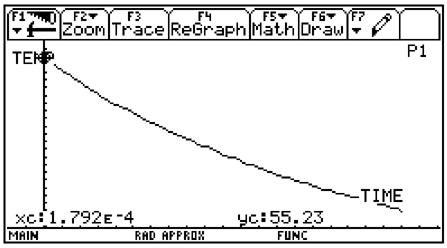
- ❑ Connect CBL and the TI-92 and connect the temperature probe to the CH1 input.
- ❑ Run the PHYSICS program.
- ❑ Select “Set up probes” from the “Main menu”; select “One” as the number of probes; select “Temperature” from the “Select probe menu”.

- ❑ Confirm that the temperature probe is connected and press <Enter>.
- ❑ Determine room temperature. To do this select “Collect Data” from the “Main menu” and “Monitor input” from the “Data collection menu”.
- ❑ Observe the temperature reading on the calculator. When it is stable, record the value as the room temperature. Press <+> to leave the monitor mode.
- ❑ Obtain some water at about 55°C. Carefully fill the film canister about three-fourths full with the hot water. Place the cap containing the probe onto the canister and press until it is sealed with a click.
- ❑ Select “Collect Data” from the “Main menu” and “Time graph” from the “Data collection menu”. Sample time should be 15 s and number of samples 80. This makes 20 minutes length of the experiment.



- ❑ Press <Enter> and select “Use time setup” to continue.
- ❑ Select “Non-live displ” from the “Time graph menu”.
- ❑ Press <Enter> to begin data collection. After the CBL shows “Done” on its screen, the calculator will have turned itself off. Turn it back on and press <Enter>.
- ❑ Select “Retrieve data” from the “Main menu” and press <Enter>.
- ❑ Press <Enter> to see your graph.

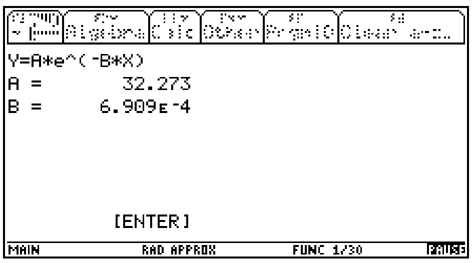


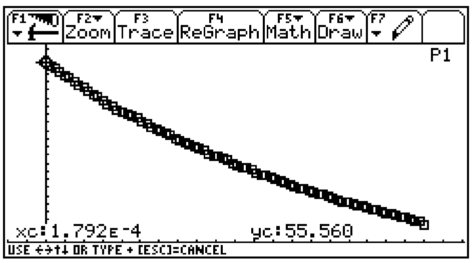


- ❑ Press <Enter> and select “No” to return to the “Main menu”.
- ❑ Select “Quit” from the “Main menu” and press <F5> to return to the home screen.
- ❑ Press „L2 - (room temperature, here 22.53) <STO ▸ > L2“. This replaces the measured water temperatures with the temperature above room temperature.
- ❑ Restart the PHYSICS program and proceed to the “Main menu”.

Next fit the exponential function to your data.

- ❑ Select “Analyze” from the “Main menu”; select “Curve fit” from the “Analyze menu”; select “Exponent L1, L2” from the “Curve fit menu”. Press <Enter> to view the fitted equation along with your data.





Titbits from Algebra and Number Theory (18)

by Johann Wiesenbauer, Vienna

Embarrassingly enough, I have to start this column by pointing out a mistake in my last one. When talking about the new options of string manipulation I gave the example

```
ODD_PART(n) := REVERSE(REVERSE(n))
```

as a short way of computing the odd part of a nonzero integer. Unfortunately, I forgot to mention that Derive must be in the binary mode for this, otherwise it won't work. Look at the following example with all the steps written out in detail

```
n := 52

OutputBase := Binary

                                        n = 110100

                         (n := REVERSE(n)) = 1011

                         (n := REVERSE(n)) = 1101

OutputBase := Decimal

                                          n = 13
```

We are relying here on the simple fact that after the first n:=REVERSE(n), which reverses the order of digits in the binary representation of n, trailing zeros become leading zeros and these are left out by Derive. Hence, the complete program should look thus

```
ODD_PART(n) :=
  Prog
    OutputBase := Binary
    n := REVERSE(REVERSE(n))
    OutputBase := Decimal
    n
```

which is no longer shorter or more elegant than the original version, namely

```
ODD_PART(n) :=
  Loop
    If ODD?(n)
       RETURN n
    n := n/2
```

Surprisingly, this small though regrettable mistake of mine (which was by the way corrected in the accompanying file TITBIT17.DFW available on the Internet) really got the goat of one of my readers who wrote a furious letter to the editor complaining about my "sinister tricks" *("… leider ist unter der Ägide des Hl. Johann das Programmieren in Derive zu einer sinistren Trickserei geworden.")*.

On this occasion, it also turned out that he didn't understand the way local variables are handled in DfW 5. *("Beispielsweise ist es absolut unvernünftig, lokale Variablen in der Parameterliste aufzuführen, wie es in DNL #38 vorgeführt wird. Wer hat diesen Unsinn sich eigentlich ausgedacht? Keine vernünftige Programmiersprache lässt so etwas zu… ["For example, there is absolutely no point in including local variables in the list of parameters of a function, as this was practised in DNL #38. Who is actually to blame for making up this rubbish? No reasonable programming language would allow such things…])*#

Well, obviously he doesn't know one of the oldest programming languages, namely LISP, which uses exactly the same method of introducing local variables ….

But let's deal with this point in more detail. In the course of the subsequent discussion he suggested changing my function PRIME_POWER?( ) from NUMBER.MTH, i.e.

```
PRIME_POWER?(n, k_ := 1, t_ := 2) :=
  Loop
    If t_ > n
       RETURN false
    If PRIME(n^(1/k_)) exit
    k_ :+ 1
    t_ :* 2
```

in the following way

```
prime_power?(n) :=
  Prog
    k := 1
    t := 2
    Loop
      If t > n
         RETURN false
      If PRIME?(n^(1/k)) exit
      k := k + 1
      t := 2·t
```

(I have only changed the name of his function making use of the fact that DfW5 is smart enough to distinguish between names written in upper case and lower case whenever there is an ambiguity.)

Can you see at first glance what's wrong with his version of PRIME_POWER?( ) ? If not, the following might be quite instructive…

Suppose somebody is using it as an auxiliary function within his own function

```
count(l) :=
  Prog
    k := 0
    Loop
      If l = []
         RETURN k
      If prime_power?(FIRST(l))
         k :+ 1
      l := REST(l)
```

which is supposed to count the number of prime powers in a given list L. Then he would be very surprised by the following result

```
count([1, ..., 10]) = 4
```

which is obviously wrong. How come? Well, it is because the variable k, which clearly has the status of a global variable, is used both in prime_power?( ) and in count( ) thereby causing problems. (Looks familiar to people who did a lot of programming in Derive before version 5, doesn't it?)

Now what about the function prime_power?( ) itself, does it return correct results? Yes, it does, but if there happens to be another global variable k somewhere in the program, its value will be changed inevitably by calling prime_power?( ). Have a look at the following example

```
k := 5
                                   prime_power?(9) = true
                                          k = 2
```
#

(This is the nightmare of every programmer, who simply calls a library function without5thinking about the consequences for his own variables, isn't it?)

How is it done properly? Well, since Derive is based on LISP, it comes as no surprise that both languages use the same method to introduce real local variables by including them in the list of parameters of the function, like it or lump it. What follows is the correct version of COUNT( ) (based on my PRIME_POWER?( ) above):

#

```
COUNT(l, k_ := 0) :=
  Loop
    If l = []
       RETURN k_
    If PRIME_POWER?(FIRST(l))
       k_ :+ 1
    l := REST(l)
```
#

As you would expect it, my functions pass both these tests with flying colours:

```
k_ := 5
                                   COUNT([1, ..., 10]) = 7
                                          k_ = 5
```
#

You may have noticed that I underscored variables in the parameter list of the function which technically speaking aren't placeholders but auxiliary variables. This is supposed to be a broad hint to the user that he shouldn't worry about these variables and simply leave them out when calling the function. I highly recommend this practice to all Derive-programmers in order to make their programs more easily readable …

By the way, I have also sent all the examples and explanations above to the guy who complained about the inclusion of local variables in the parameter list of a function and, believe me, I really would like to have been able to tell you that there was a happy ending after all. But this is what he wrote back:

"Im Unterricht würde ich trotzdem meine Version von 'Primzahlpotenz' vorziehen (weil sie verständlicher ist) und dann darauf hinweisen, dass man beim Test nicht die gleichen Variablen verwenden darf…" *[In the classroom I would prefer my version of prime_power( ) all the same (because it is more understandable) and then point out that different variables should be used when testing…]*

Well, it's nice if a function becomes more "understandable" (I don't want to argue about whether this is really the case here or not), but above all it should work properly, shouldn't it? Anyway, this convinced me that giving more examples and explanations would have been a complete waste of time…

Let's turn to a totally different topic now. Taking as an example NTH_PRIME( ) from NUMBER.MTH, I'll show how you can endow functions in Derive with a kind of memory (a feature you may know from some other CAS, e.g. Maple). What we want is that every time you call NTH_PRIME(n) for some n, then the pair (n,NTH_PRIME(n)) should be stored in a kind of table and further calls should make use of the entries in this table.

The original function NTH_PRIME(n) looks like this

```
NTH_PRIME(n) := ITERATE(NEXT_PRIME(k_), k_, 1, n)
```

The following two calls

```
NTH_PRIME(10000) = 104729
NTH_PRIME(10100) = 105943
```

took about the same amount of time, namely 0.93s on my PC. First let's modify NTH_PRIME( ) in the following way

```
NTH_PRIME(n, s_ := 1, t_ := 0) := ITERATE(NEXT_PRIME(k_), k_, s_, n - t_)
```

in order to exploit the knowledge of an equation of the form NTH_PRIME(t_)=s_. For example, the computation

```
NTH_PRIME(10100, 104729, 10000) = 105943
```

takes only 0.01s now. As you can see by having a closer look at it, the time saved results from the fact that we didn't count the first 10000 primes once more, but only the additional 100 primes making clever use of the first function call NTH_PRIME(10000)=104729.

As for the table mentioned above, we could use a polynomial BRAIN with the initial value 2x whose summands are of the form sx^t meaning that NTH_PRIME(t)=s. For example, after making the two function calls above this variable should look like this

$$BRAIN = 105943 \cdot x^{10100} + 104729 \cdot x^{10000} + 2 \cdot x$$

#

The further steps should have become fairly clear by now. Each time we want to compute NTH_PRIME(n) for some positive integer n we do the following:

- Look up in the variable BRAIN the biggest t ≤ n such that the nonzero term sx^t is a summand of it.
- Compute s:=NTH_PRIME(n,s,t) as in the example above.
- Add the term sx^n to BRAIN, unless t=n.

This leads to the following implementation:

```
BRAIN := 2·x

NTH_PRIME(n, s_, t_) :=
  Prog
    t_ := FIRST(TERMS(REMAINDER(BRAIN, x^(n + 1))))
    s_ := LIM(t_, x, 1)
    t_ := x·DIF(LN(t_), x)
    s_ := ITERATE(NEXT_PRIME(k_), k_, s_, n - t_)
    If t_ < n
       BRAIN :+ s_·x^n
    s_

                                NTH_PRIME(10000) = 104729

                                NTH_PRIME(10100) = 105943  #
```

Again the computation of NTH_PRIME(10100) is done in no time at all (0.02s). Note that both x and BRAIN are global variables which must not be assigned any values by the user other than the ones above!

Another interesting function closely related to NTH_PRIME(n) is PRIMEPI(x), which counts the number of primes p ≤ x for any positive real number x. As a matter of fact, the restriction of PRIMEPI(x) to the set **P** of primes is the inverse of NTH_PRIME(n). There is already a nice implementation of PRIMEPI(x) in the utility-file NUMBER.MTH, namely

```
PRIMEPI(x, d, a) :=
  If d = 0
     0
     Σ(IF(PRIME(d·n_ + a)), n_, 0, (x - a)/d)
     PRIMEPI(x, 6, 1) + PRIMEPI(x, 6, 5) + IF(x ≥ 3, 2, CHI(2, x, 3, 1)) #
```

which computes more generally the number of primes p ≤ x of the special form kd+a, where d and a are fixed coprime natural numbers. If d and a are left out of the function call, i.e. by evoking PRIMEPI(x) you get the number of all primes p ≤ x. (This raises the interesting question: Are d and a auxiliary variables in this case or "genuine" variables, i.e. placeholders? As you can see here, a clear distinction is not always easily made!) For example, we have

```
PRIMEPI(104729) = 10000
```

as it should be. The implementation above is reasonably fast, if x is not too large, say less than 10^6. For bigger values of x one should use more sophisticated implementations (cf. [1]). There also exist some nice approximations for PRIMEPI(x) and even an exact formula that uses the zeros of the so-called Riemann zeta-function. But this is another story which I'd better save for another time (j.wiesenbauer@tuwien.ac.at).

[1] J. Wiesenbauer, Abzählen von Primzahlen mit DERIVE, ÖMG Schriftenreihe zur Didaktik, Heft 27, 196-206 (1997)

Find all the DERIVE and TI-files on the following web site

www.austromath.at/dug/

Joseph

Further to my paper in DNL#39 and our subsequent correspondence, referring to the values of 'm', for which the Meander/Elastica family of equations; $\varphi = m \sin(s)$ is represented by a Jordan (closed) curve, meaning it stops its progress in the positive or negative direction and has an oscillatory phase, staring with the 'figure eight' curve at $m_1 = 2.366$ (approximately) and then at approximately $m_2 = 5.513$ and $m_3 = 8.648$ etc...

Since I found these out by trial and error using a routine, that espoused 'dubious accuracy', I wondered if there was a direct mathematical method. Since, unsurprisingly, they increase by, π (as near as damn it), then the first value, m_1 is the fundamental one to seek with great accuracy.

$3\pi/4 = 2.35619$, which is a little less than my value of $m_1 = 2.366$, unless I was exercising the 'dubious accuracy' option too freely. Nevertheless, I cannot justify $\varphi = (3\pi/4) \cdot \sin(s)$.

"There's the rub", as Shakespeare would assert.

We know that the arclength is infinite in all cases, except those instances, when at the oscillatory phase. But "to use that or not use that, that is the question", is the bard's sure retort.

When one solves a known equation such as $\sin(s) = 1$, then $s = \text{Arcsin}(1) = \pi\,(2n + 1/2)$.

But if the sign of 1 were optionally positive or negative, then $s = \text{Arcsin}(1) = \pi\,(n + 1/2)$.

So, solving the Meander/Elastica for s where φ may be positive or negative we obtain:

$$s = \text{Arcsin}(\varphi/m = n\,\pi + \text{Arcsin}(\varphi/m))$$

This only explains the incremental advancements from M_1. Perhaps one can use L'Hopital's Rule for the limit of $m = \varphi(\sin(s)$ as s and φ approach some value(s), if one knows what that value is; then again maybe I'm on the wrong track, and then some.

"A solution, a rigorous solution, my kingdom for a solution", Will' saith.

Was denkst du, Herr Magister, oder andere?

Herzlichst

David Halprin

Lady Slippers from the „Meander region" (page 29), Josef