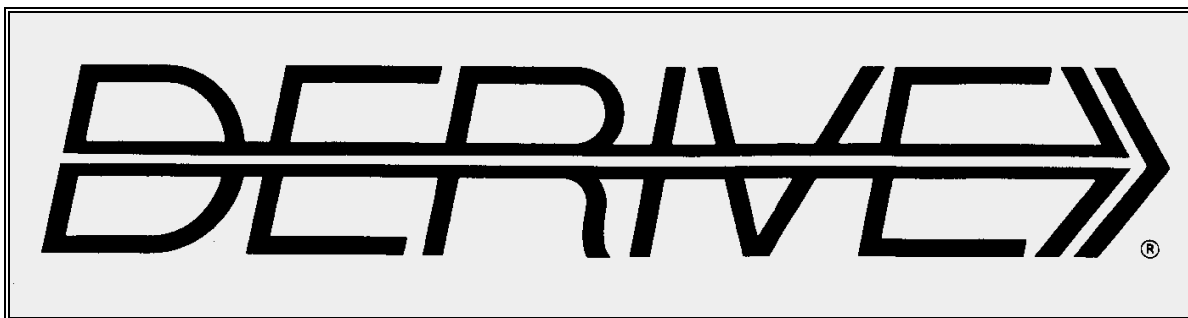


THE BULLETIN OF THE



USER GROUP



C o n t e n t s :

- | | |
|----|---|
| 1 | Letter of the Editor |
| 2 | Editorial - Preview |
| 3 | User Forum 1
DIMS-function, Repartition
Duncam E McDougall |
| 6 | Radical Radicals
Mc Donald R Phillips |
| 18 | Financial Mathematics III (NPV, IRR, mod. IRR, and EQV)
Lorenz Kopp |
| 27 | Four Graphic Simulations of Random Games |
| 31 | User Forum 2
Set Operations
Confusion with Complex Numbers
Two strange limits
InterConnectivity Problem with matrices
A Seashell |
| 43 | DES-TIME 2006 7 th International DERIVE & TI-CAS-Conference
Submitted and Accepted Lectures and Workshops |

There are new DERIVE Materials available:

[1] **Tipps & Tricks zu Derive**, *B Kutzler & V Kokol-Voljc*, bk teachware SR-51

[2] **Lehr- und Lernvideos zur Mathematik und DERIVE**, *H-J Kayser*, bk-teachware SR-52

Very interesting and useful:

[3] **Einführung in das "digitale Auge" ProScope**, *V Kokol-Voljc & B Kutzler*, bk-teachware Sr-49

Available: <http://shop.bk-teachware.com>

GREAT NEWS: Calculus Made Easy version 7.0 is now available.

MAJOR UPGRADES MAKE CME EVEN MORE POWERFUL THAN MATHEMATICA & MAPLE:

View Step by Step Solution to any Integration or Differentiation problem.

In Example, when performing U-Substitution to integrate $x \cdot \sin(3x^2)$ choose to select your own U and observe if it leads to the correct answer.

Or simply lean back and let CME find the correct U as well as the final solution using a detailed step by step by process.

Similarly, Step by Step Solutions to Integration problems involving:

Integration by parts,

Power rule,

Partial fractions,

Expanding/rewriting or

Arctan(x) and Arcsin(x) are available.

Even better, step by step solutions can also be viewed for differentiation using Chain rule, Power rule, Product rule, Quotient rule.

No other software currently offers this learner-friendly process making Calculus Made Easy the ideal Calculus learning tool.

To view some examples, check here: www.ti89.com/documentation.htm

ALSO NEW:

- Find Domain and Discontinuities of Functions

- SignCharts for $f(x)$, $f'(x)$ and $f''(x)$ -

ideal to find min/max/in- or decreasing and points of inflection/concavity.

- Automatic Sequence Convergence Tester.

- Recursive and Explicit Sequence Formulas.

- Limit Convergence Test for Series.

- Polar Coordinates: Convert to Cartesian Coordinates and back, Find Intersection Points, Find Angle Theta given coordinates and polar equation.

- Hessian and Jacobian .Find Differentials for Multivariable Calculus.

-Unit Circle shows angles in radian as well.

-Ultimate Solver: Solve any equation for any variable.

-Algebra: Simplify fractions, Expand polynomials, Find proper Fractions

Product information at: www.ti89.com/index_cme.htm .

To Upgrade: Simply click on <Upgrade> at www.ti89.com/store.htm .

Benefit from Referrals: Earn \$5.00 for every new referral to CME.

Contact: support@ti89.com

Have a wonderful day,

Nils Hahnfeld

SmartSoft

PS: Does $f(x)=|x^2-1|$ have inflection points???

Dear DUG-Members,

Three months passed so quickly and it is again time for the next newsletter. I hope that you downloaded revised DNL#9 in the meanwhile and that you are ready for DNL#61.

We had the honour and pleasure to welcome our vice president Bärbel Barzel at the occasion of a seminar in Amstetten, Lower Austria. Bärbel gave a plenary lecture followed by a workshop. We had also Benno Grabinger – the "Pringles Man" (DNL#59) and Bernhard Kutzler – the "Derive Europe Man" (DNL#60) as our very welcome lecturers. You can see that DUG members are highly appreciated to share their knowledge. The pictures show Bärbel and Josef Lechner (DUG-secretary), who organized this seminar in his school. Main organizers were Walter Klinger and Walter Wegscheider – again DUG-members. It was not a DUG-seminar but an official seminar of our school authorities ...



I'd like to give some comments on the contributions published in this DNL.

You might be surprised reading the first article *Radical Radicals* – nothing about using any technology – in contrary focussing on manipulating skills. Duncan sent this paper together with some others and I asked for his permission to publish it in the DNL hoping to raise – once again – the question about – more or less – "Indispensable skills". Find more information at the end of his article.

Starting compiling this issue I had the intention to make it a bit smaller – because of DNL#9 and a lot of other duties. But as you can see it is again very contentful – the User Forum turned out to be very interesting (Complex Number Problems, Limits,

InterConnectivity, ...). Jan Vermeyleen sent a challenge for graphic representation of point sequences. Just today I received an answer from Peter Schofield which seems to provide a very useful hint how to resolve this problem – see the next DNL.



Lorenz Kopp demonstrates how to create simulations of random experiments not only by calculation but also by visualization and finally our expert in financial mathematics, Don Phillips, offers the 3rd part of his series on this interesting math application.

On the last two pages you can find the list of submitted and accepted lectures and workshops for the Dresden Conference in July. In the meanwhile many of the DUG-members have registered and it will be great to meet you all in Saxony

The full program will be published on the website very soon and you are invited to browse the presentations.

Finally spring has arrived in Austria after a very long winter and I hope that you can enjoy this wonderful period of the year (and fall on the other hemisphere, of course).

If you download this DNL and you don't receive any information from me by email, then please send your address – it makes it easier to keep in touch. You can be sure that I don't forward your address to anybody else

Download all *DNL-DERIVE*- and TI-files from

<http://www.austromath.ac.at/dug/>

<http://www.derive-europe.com/support.asp?dug>

The *DERIVE-NEWSLETTER* is the Bulletin of the *DERIVE & CAS-TI User Group*. It is published at least four times a year with contents of 44 pages minimum. The goals of the *DNL* are to enable the exchange of experiences made with *DERIVE* and the *TI-89/92/Titanium/Voyage 200* 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 *CAS-TIs* the *DNL* tries to combine the applications of these modern technologies.

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

Contributions:

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

Next issue: June 2006
Deadline 31 May 2006

Preview: Contributions waiting to be published

Two Stage Least Squares, M. R. Phillips, USA
Some simulations of Random Experiments, J. Böhm, AUT
Wonderful World of Pedal Curves, J. Böhm
Another Task for End Examination, J. Lechner, AUT
Tools for 3D-Problems, P. Lüke-Rosendahl, GER
ANOVA with *DERIVE & TI*, M. R. Phillips, USA
Financial Mathematics 4, M. R. Phillips
Hill-Encryption, J. Böhm
Farey Sequences on the *TI*, M. Lesmes-Acosta, COL
Simulating a Graphing Calculator in *DERIVE*, J. Böhm
Hénon & Co, J. Böhm
Challenges from Fermat, Bj. Felsager, DEN
Are all Bodies falling equally fast, J. Lechner
Modelling Traffic Density, Th. Himmelbauer, AUT
Do you know this? Cabri & CAS on PC and Handheld, W. Wegscheider, AUT
An Interesting Problem with a Triangle, Steiner Point, P. Lüke-Rosendahl, GER
Mathematics and Design, Hubert Weller, GER
Diophantine Polynomials, Duncan E. McDougall, Canada
Contour Plots and Implicit Plots, Peter Schofield, UK

and Setif, FRA; Vermeylen, BEL; Leinbach, USA; Koller, AUT; Baumann, GER; Keunecke, GER,and others

Impressum:

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

DIMS-function**Dr. Lunsford**

16.2.06 08:40

Hello,

I've written a recursive function to find the dimensions of an array and return a vector (or 0 if the argument is a scalar);

```
DIMS(A) :=
  Prog
  If NOT (MATRIX?(A) OR VECTOR?(A))
    RETURN 0
  If A = APPEND(A)
    [DIM(A)]
  APPEND([DIM(A)], DIMS(A SUB 1))
```

Any objections? (Try it out with arrays of empty vectors.)

-drl

Mate Matica [mmatica@personal.ro]

17.2.06, 16:47

Hello,

(MATRIX?(A) OR VECTOR?(A)) is the same as VECTOR?(A).

What about this one?

```
dims(A,d:=[]):=IF(VECTOR?(A),dims(FIRST(A),APPEND(d,[DIM(A)])),d)
```

Mate

Johann Wiesenbauer

17.2.2006, 17:09

Hello,

What about this one?

```
dims(A) := [DIM(A), DIM(A`)]
```

Cheers,
Johann

Mate Matica [mmatica@personal.ro]

17.2.06, 19:16

Hello again,

Works only for 2 dimensions.

Mine is the best :-)

Mate

Repartition**Dr. Lunsford,**

17.2.06 20:12

OK make this one better! Acts like "reshape" in APL:

```

REPARTITION(A, D, B, W, N)
PROG(
  IF(NOT VECTOR?(A) OR A = [], RETURN A),
  IF(NOT VECTOR?(D), RETURN "err: second arg not a vector"),
  B := RAVEL(A),
  IF(PRODUCT(D) > DIM(B),
    W := APPEND(ITERATE(APPEND(B, x), x, B, FLOOR(PRODUCT(D), DIM(B))-1),
      B SUB [1, ..., MOD(PRODUCT(D), DIM(B))]),
    W := B SUB [1, ..., PRODUCT(D)]),
  N := DIM(D),
  LOOP(IF(N = 1, RETURN W, PROG(B:=W, W:=PARTITION(B, D SUB N), N:=N-1)))

```

-ross

Mate

17.2.2006, 23:47

APL? No thanks.

BTW, I think you have a typo: RAVEL instead of BEETHOVEN.

Mate

Dr. Lunsford

18.2.2006, 08:55

Sorry,

```

BEETHOVEN(A) :=
  Prog
  If A = APPEND(A)
  A
  BEETHOVEN(APPEND(A))

```

-drl

I've concluded that Derive is a very superior system for dealing with algebraic objects, at least as good as APL and likely better - I'm having no trouble at all reproducing the actions of APL operators on arrays. The Derive syntax is just as rich as APL's, and moreover Derive provides the "set" object - this may be somewhat like some APL's idea of enclosed array, but I'm not sure yet.

Is anything really known about this system other than application level solutions? I've forgotten about APL since stumbling across Derive. Any pointers to advanced programming topics? Data exchange interface? Algorithms? Idioms?

-drl

Johann Wiesenbauer

19.2.2006, 17:51

Hello,

What about the following version, which in particular doesn't use any auxiliary function like ravel (or beethoven or whatever)?

```
myrepart(a, d, d_) :=
  PROG(d_ :=product(d),
    IF(DIM(a) < d_,
      a := (ITERATE(APPEND(a, a_), a_, a, d_/DIM(a))) sub [1, ..., d_]),
      (ITERATE([PARTITION(b_, d sub k_), k_ - 1], [b_, k_], [a, DIM(d)],
        DIM(d) - 1)) sub 1)
```

Note that on my PC the PARTITION() imbedded in ITERATE on the last line didn't work unless I called a simple example of it before in order to load the corresponding library. A bug or a feature?

Cheers,
Johann

Johann Wiesenbauer

20.2.2006, 17:51

When rereading my email below, I found out that the assignment of a should be unconditionally, i.e.

```
myrepart(a, d, d_) :=
  PROG(
    d_ :=product(d),
    a := (ITERATE(APPEND(a, a_), a_, a, d_/DIM(a))) sub [1, ..., d_],
    (ITERATE([PARTITION(b_, d sub k_), k_ - 1], [b_, k_], [a, DIM(d)],
      DIM(d) - 1)) sub 1)
```

otherwise the program won't work correctly for all vectors a. Sorry!

Cheers,
Johann

Dr. Lunsford

21.2.2006

Thanks that is very clever Derive, but mine is faster :) 32.1 seconds for 5000 repetitive repartitions and assignments of [1,...,120] as [2,3,4,5], yours took 34.6. However, yours is conceptually superior so the speed difference is irrelevant.

Also, repartition of a scalar should return an D-sized array of these scalars - we both got that wrong - and I think repartition of EMPTY should not be allowed, and I got that one wrong, while repartition BY [] should return A sub 1 sub 1... sub 1 as a scalar. Finally repartition by a scalar s should be the same as repartition by [s].

We roll this in together with the DIMS function to make "SHAPE", which with 1 argument, returns the size of the array, and with 2, returns the redimensioned array:

Radical Radicals

Duncan E. McDougall, duncanemcdougall@hotmail.com

(1)

Today's math student does not like radicals any more than yesterday's student does. Although he or she is armed with a calculator, the guessing processes of how to break down radicals or when to rationalize the denominator still dominate the thinking and strategy processes. As well, students are not always sure that their final answer is in lowest terms, especially when dividing. There has to be a set approach to all operations involving radicals that the student can use to resolve some of these issues.

Consider an example like $\sqrt{75} + \sqrt{27}$. We do suggest to our students that they reduce the radicands before combining terms. The logical question, of course, is whether these terms can be combined to begin with, and if so, how to proceed with the problem. In this case, how is the innumerate student supposed to know that 3 is the number to start with, especially if we have taught our students to extract perfect squares from each of the given radicals? How is he or she to proceed with confidence and certainty to begin with?

We then state that we cannot add or subtract the terms if the g.c.f. of the radicands is 1. We also suggest not to multiply or divide until we have determined the g.c.f. of the given radicands. As well, we do not rationalize the denominator unless the g.c.f. of both numerator and denominator is 1.

The trained eye observes that g.c.g. (27,75) is not 1 but 3: thus, the question is now worth doing. As 3 is a common factor of both 75 and 27, we now express $\sqrt{75}$ and $\sqrt{27}$ in terms of $\sqrt{3}$. Instead of having to guess how to break down both 75 and 27, we already have one of the key factors. The student then has the simple task of dividing 3 into 27 and 75 before simplifying the terms.

Consider the conventional way of multiplying two radicals: e.g., $\sqrt{28} \cdot \sqrt{63}$. Many calculator-oriented students would multiply $28 \cdot 63 = 1764$ and then try to break down $\sqrt{1764}$ only to find 42. If we now tell the student to break them down first before multiplying, a guessing or guess/estimating process begins as to what numbers go into 28 and then into 63. It is discovered that $\sqrt{28} = 2\sqrt{7}$ and that $\sqrt{63} = 3\sqrt{7}$, but only after calculating the numbers one at a time.

The process of **Radical Radicals** involves considering both radicands at the same time by first finding their g.c.f. To do this, we use Euclid's Algorithm (without calculator).

(2)

e.g. Find g.c.f (28,63)

Steps

1. Divide the larger number by the smaller number, keeping track of the remainder.
2. Now divide the remainder into the previous divisor.
3. Continue step 2 until the remainder is exactly zero.
4. The divisor that yields a remainder of zero is our g.c.f.

$$\begin{array}{r} \underline{2} \\ 28) 63 \\ \underline{56} \\ 7 \end{array} \qquad \begin{array}{r} \underline{4} \\ 7) 28 \\ \underline{28} \\ 0 \end{array}$$

We know 28 and 63 have 7 as a greatest common factor and so $\sqrt{28}$ and $\sqrt{63}$ have $\sqrt{7}$ as a common factor. This brings about a different way of doing radicals because we work with two radicands at a time, not one. As well, when it does come time to break down the radicals, half the work is already done, as we already know one of the factors. Thus, guessing or guess/estimating is reduced dramatically.

Now consider $\sqrt{26} \cdot \sqrt{65}$. Instead of looking at $\sqrt{1690}$, we determine that $\text{g.c.f.}(26,65) = 13$ and express each factor in terms of $\sqrt{13}$.

This gives

$$\begin{aligned} & \sqrt{26} \cdot \sqrt{65} \\ &= \sqrt{2} \cdot \sqrt{13} \cdot \sqrt{5} \cdot \sqrt{13} \\ &= 13\sqrt{10} \end{aligned}$$

The numbers never become large, so simplifying is easier.

The disadvantage of this method is that students must find the g.c.f. of two numbers with little help from a calculator. The advantages are that once the g.c.f. has been found, half the work is already done because guessing is virtually eliminated and the numbers never increase in size. This reduces frustration and error made with large numbers.

In summary, the steps to this new approach are as follows:

1. Find the g.c.f. of the radicands
2. Express each radicand in terms of the square root of this radicand.
3. "Pair-off" like radicals.
4. Simplify remaining terms.

(3)

If you think multiplication done in this fashion is efficient, let us now look at division. this is where this method really shines.

Consider $\frac{\sqrt{175}}{\sqrt{112}}$.

Conventionally, if the student does not recognize that 7 is a common factor, he/she might multiply top and bottom by $\sqrt{112}$, giving horrendous numbers to work with.

Instead, use Euclid's Algorithm to find that g.c.f. (112,175)=7. Thus, we have

$$\frac{\sqrt{175}}{\sqrt{112}} = \frac{\sqrt{7} \cdot \sqrt{25}}{\sqrt{7} \cdot \sqrt{16}}$$

This leaves the answer $\frac{5}{4}$, with no other calculations to do.

What happens if the g.c.f. of the numerator and denominator is 1? We simply multiply top and bottom by the denominator knowing that we will not have to reduce the fraction after the multiplication of terms.

Now, what happens if the numerator is not a radical? Consider $\frac{2}{\sqrt{10}}$.

Here we do not multiply top and bottom by $\sqrt{10}$ because g.c.f. (2,10)=2 (not 1).

Instead we recall that $\sqrt{a} \cdot \sqrt{a} = a$ and in this case $\sqrt{2} \cdot \sqrt{2} = 2$. Since g.c.f. (1,10)=2, we express both numerator and denominator in terms of $\sqrt{2}$. This gives

$$\frac{2}{\sqrt{10}} = \frac{\sqrt{2} \cdot \sqrt{2}}{\sqrt{2} \cdot \sqrt{5}}$$

The $\frac{\sqrt{2}}{\sqrt{2}}$ "cancels," and we are now left with $\frac{\sqrt{2}}{\sqrt{5}}$. Since g.c.f. (2,5) = 1, we can now multiply top and bottom by $\sqrt{5}$ and not have to worry about reducing the new function.

Finally, we have

$$\frac{\sqrt{2}}{\sqrt{5}} \cdot \frac{\sqrt{5}}{\sqrt{5}} = \frac{\sqrt{10}}{5}$$

(4)

We do not have to go backwards or look over our shoulder to see if the fraction can be reduced as the g.c.f. is 1. Our rule then is to only multiply top and bottom by the denominator factor when g.c.f. is 1. In general, the steps for division are as follows.

Step 1 Find the g.c.f. of both numerator and denominator.

Step 2 Express each radicand in terms of the square root of this g.c.f.

Step 3 “Pair-off” like radicals and reduce.

Step 4 Simplify remaining terms.

Step 5 Multiply top and bottom by the denominator term only when the g.c.f. of numerator and denominator is 1.

For division then, we merely add step 5 to the method for multiplication.

This new process now simplifies addition and subtraction considerably because we can operate only if the g.c.f. is not 1. Hence, for $\sqrt{3} + \sqrt{7}$, we suggest that this cannot be simplified any further because g.c.f. (3,7)=1. By the same token, for $\sqrt{24} + \sqrt{54}$, we find that g.c.f. (24,54)=6. We then have

$$\begin{aligned} & \sqrt{24} + \sqrt{54} \\ &= \sqrt{4} \cdot \sqrt{6} + \sqrt{9} \cdot \sqrt{6} \\ &= \sqrt{6}(2+3) \\ &= 5\sqrt{6}. \end{aligned}$$

Again, we look at two at a time, not one at a time. We then ascertain that terms can indeed be combined because their g.c.f. is not 1. If indeed we have more than two terms, we can then look for two that have the same g.c.f.

For combined operations, we look at 2 g.c.f.'s as opposed to 1. Consider an example like $\sqrt{3} \cdot (\sqrt{15} + \sqrt{21})$. Since g.c.f. (3,15)=3 and g.c.f. (3,30)=3, we have $\sqrt{15} = \sqrt{3} \cdot \sqrt{5}$ and $\sqrt{21} = \sqrt{3} \cdot \sqrt{7}$. Thus, we can write

$$\begin{aligned} & \sqrt{3} \cdot (\sqrt{15} + \sqrt{21}) \\ &= \sqrt{3} \cdot (\sqrt{3} \cdot \sqrt{5} + \sqrt{3} \cdot \sqrt{7}) \\ &= 3 \cdot (\sqrt{5} + \sqrt{7}). \end{aligned}$$

(5)

For division, take an example like $\frac{5}{\sqrt{7}-\sqrt{2}}$. Now g.c.f. (5,7,2)=1, so we do multiply top and bottom by the conjugate $\sqrt{7} + \sqrt{2}$.

This gives

$$\begin{aligned} & \frac{5}{\sqrt{7}-\sqrt{2}} \cdot \frac{\sqrt{7}+\sqrt{2}}{\sqrt{7}+\sqrt{2}} \\ &= \frac{5 \cdot (\sqrt{7}+\sqrt{2})}{7-2} \\ &= \frac{5 \cdot (\sqrt{7}+\sqrt{2})}{5} \\ &= \sqrt{7} + \sqrt{2}. \end{aligned}$$

In summary, by finding the g.c.f. of the radicands we introduce the idea of working with two radicands at a time, not one at a time. And, once we do have the g.c.f., we cut the work by a least half because we have one of the factors of the radicand. We eliminate large numbers, errors caused by large numbers, and the frustration from guessing. In a proactive way, we streamline the process by keeping numbers simple and neat. In my experience, student feedback is always the same: "This is easy compared to what I used to do."

Other papers which reached me were about: *Factoring Trinomials, Reducing Fractions, Greatest Common Factor, Curve Sketching, Criss-Cross and Methods of Integration.*

Some time ago Wilfried Herget, Helmut Heugl, Eberhard Lehmann and Bernhard Kutzler published a paper about "Indispensable Skills ..." as a product of a two days discussion. This paper caused a very violent discussion among mathematicians from all over the world. You can download this paper from the sites below:

English version: http://b.kutzler.com/article/art_indi/indisp.htm

German version: <http://acdca.ac.at/>

This is part 3 of Don Phillips' very extended series on Financial Mathematics. As in his former contributions (Financial Mathematics I – Bond Price and Yield, DNL#52 and Financial Mathematics II – TVM and Amortization, DNL#55) I try to solve the problems using the CAS-TI. We can apply the Finance-Application together with the associated implemented functions and write own functions. Josef

Financial Mathematics III

NPV, IRR, Modified IRR, and EQV

MacDonald R. Phillips
don.phillips@gmail.com
July 2005

Financial Mathematics III has 6 main functions or programs:

```
CashFlowAdd(bv,amt,n,gs:=1,d:=0)
CashFlowPercent(bv,p,n,gs:=1,d:=0)
NPV(cf,im,ppy:=1,cpy:=1,d:=0)
IRR(cf,ppy:=1,cpy:=1)
MIRR(cf,fr,rr,ppy:=1,cpy:=1)
EQV(cf1_,cf2_,im,ppy:=1,cpy:=1)
```

Cash flows for **NPV**, **IRR**, **MIRR**, and **EQV** may be entered as either a vector, e.g. [-1000,150,300,450,600], or, if you have grouped cashflow, as a matrix, e.g. [[-1000,1],[150,2],[300,3],[450,1]]. **All cash flows must begin at time 0.**

CashFlowAdd and **CashFlowPercent** are useful in building cashflows.

As usual, the functions follow the usual cash flow sign convention: money coming in to you is entered as a positive number and money you pay out is entered as a negative number.

The functions are hidden, but may be seen by going to Author/Function Definition...

Since I believe in learning by example, here they are!

Part 1: Using CashFlowAdd and CashFlowPercent

The variables used in the two programs are:

```
bv – begining value
amt – constant amount
p – percent
n – number of cash flows
gs – number of cash flows in a group, defaults to 1.
d – the default value of 0 indicates that the cash flows begin at the end of
period 1; a value of 1 that the cash flows begin at the beginning of period 1.
```

For instance, assume you have a cash flow with a beginning value of 1000 that increases by 50 each period for 36 periods.

#1: CashFlowAdd(1000, 50, 36)

#2: [0, 1000, 1050, 1100, 1150, 1200, 1250, 1300, 1350, 1400, 1450, 1500, 1550, 1600, 1650, 1700, 1750, 1800, 1850, 1900, 1950, 2000, 2050, 2100, 2150, 2200, 2250, 2300, 2350, 2400, 2450, 2500, 2550, 2600, 2650, 2700, 2750]

If the cashflows began at time 0, then enter:

#3: CashFlowAdd(1000, 50, 36, 1, 1)

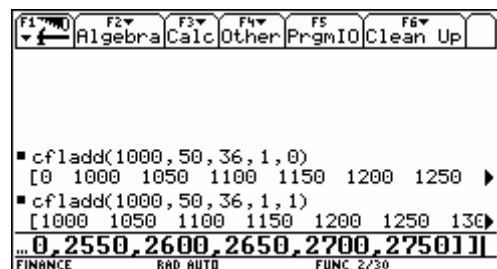
#4: [1000, 1050, 1100, 1150, 1200, 1250, 1300, 1350, 1400, 1450, 1500, 1550, 1600, 1650, 1700, 1750, 1800, 1850, 1900, 1950, 2000, 2050, 2100, 2150, 2200, 2250, 2300, 2350, 2400, 2450, 2500, 2550, 2600, 2650, 2700, 2750]

This is the DERIVE-function (program):

```
CashFlowAdd(bv, amt, n, gs := 1, d := 0, v, ng, fg, i) :=
  Prog
    fg := MOD(n, gs)
    ng := FLOOR(n, gs)
    v := VECTOR([bv + amt*i, gs], i, 0, ng - 1)
    If fg ≠ 0
      v := APPEND(v, [[v↓ng↓1 + amt, fg]])
    If d = 0
      v := APPEND([[0, 1]], v)
    If gs = 1
      v↓↓1
    v
```

And this is the TI-transcription (Josef):

```
cfl add(bv, amt, n, gs, d)
Func
Local fg, ng, l amt, l g, v
mod(n, gs) ÷ fg
fl oor(n/gs) ÷ ng
seq(gs, k, 1, ng) ÷ l g
seq(bv+amt*k, k, 0, ng-1) ÷ l amt
If fg 0 Then
  augment(l g, {fg}) » l g
  augment(l amt, {l amt[ng]+amt}) » l amt
Endl f
If d=0 Then
  augment({1}, l g) » l g
  augment({0}, l amt) » l amt
Endl f
l i st ÷ mat(augment(l amt, l g), di m(l g)) » v
If gs=1 Then
  v[1] » v
El se
  v™ » v
Endl f
EndFunc
```



In TI-functions it is not possible to have default values for variables (**gs** and **d**). They must be entered in all cases.

Now assume that the 36 periods are months and that the cash flows only increase once a year. If the cash flows begin at time 1, then enter:

#5: CashFlowAdd(1000, 50, 36, 12)

#6:

$$\begin{bmatrix} 0 & 1 \\ 1000 & 12 \\ 1050 & 12 \\ 1100 & 12 \end{bmatrix}$$

As you can see, grouped cash flows are entered in a matrix with the second column indicating the number of cash flows in each group.

If the cash flows begin at time 0, then enter:

#7: CashFlowAdd(1000, 50, 36, 12, 1)

#8:

$$\begin{bmatrix} 1000 & 12 \\ 1050 & 12 \\ 1100 & 12 \end{bmatrix}$$

Of course, the "addition" to subsequent cash flows can be negative.

#9: CashFlowAdd(1000, -50, 36, 12, 1)

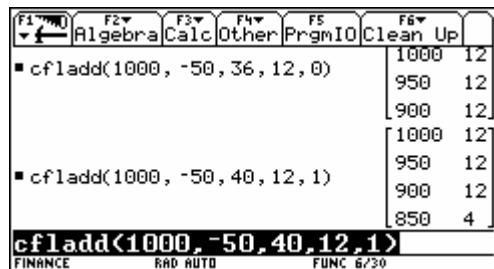
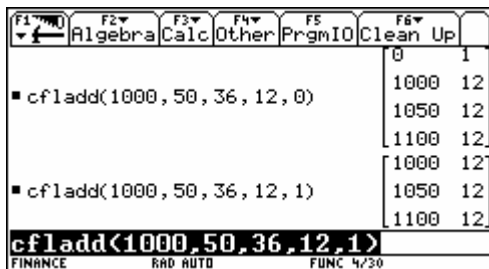
#10:

$$\begin{bmatrix} 1000 & 12 \\ 950 & 12 \\ 900 & 12 \end{bmatrix}$$

If the number of cash flows is not evenly divided by the number of cash flows in a group, the remaining number of cash flows simply occurs as the last group.

#11: CashFlowAdd(1000, -50, 40, 12, 1)

#12:

$$\begin{bmatrix} 1000 & 12 \\ 950 & 12 \\ 900 & 12 \\ 850 & 4 \end{bmatrix}$$


Same on the TI.

If the cash flows are more irregular, then the matrix or vector of cash flows simply has to be entered by hand.

The same principals apply to the function CashFlowPercent, except that the p refers to a percentage increase or decrease in the cash flows. For instance, a series of monthly cash flows beginning at time 1 and increasing by 4 percent each year for 5 years would be entered thus:

#13: CashFlowPercent(1000, 4, 60, 12)

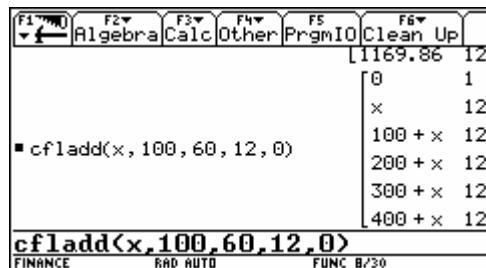
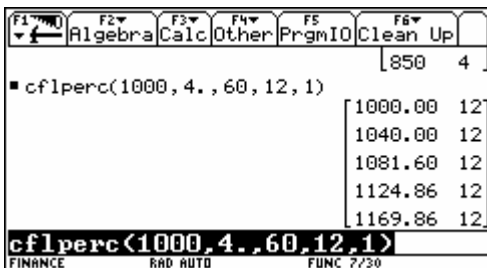
#14:	<table style="border-collapse: collapse; margin: auto;"> <tr><td style="padding: 5px 15px;">0</td><td style="padding: 5px 15px;">1</td></tr> <tr><td style="padding: 5px 15px;">1000</td><td style="padding: 5px 15px;">12</td></tr> <tr><td style="padding: 5px 15px;">1040</td><td style="padding: 5px 15px;">12</td></tr> <tr><td style="padding: 5px 15px;">1081.6</td><td style="padding: 5px 15px;">12</td></tr> <tr><td style="padding: 5px 15px;">1124.864</td><td style="padding: 5px 15px;">12</td></tr> <tr><td style="padding: 5px 15px;">1169.85856</td><td style="padding: 5px 15px;">12</td></tr> </table>	0	1	1000	12	1040	12	1081.6	12	1124.864	12	1169.85856	12
0	1												
1000	12												
1040	12												
1081.6	12												
1124.864	12												
1169.85856	12												

The beginning cash flow, does not have to be an actual dollar amount. In fact, the EQV function requires one of the cash flow series to have an unknown.

#15: CashFlowAdd(x, 100, 60, 12)

#16:	<table style="border-collapse: collapse; margin: auto;"> <tr><td style="padding: 5px 15px;">0</td><td style="padding: 5px 15px;">1</td></tr> <tr><td style="padding: 5px 15px;">x</td><td style="padding: 5px 15px;">12</td></tr> <tr><td style="padding: 5px 15px;">x + 100</td><td style="padding: 5px 15px;">12</td></tr> <tr><td style="padding: 5px 15px;">x + 200</td><td style="padding: 5px 15px;">12</td></tr> <tr><td style="padding: 5px 15px;">x + 300</td><td style="padding: 5px 15px;">12</td></tr> <tr><td style="padding: 5px 15px;">x + 400</td><td style="padding: 5px 15px;">12</td></tr> </table>	0	1	x	12	x + 100	12	x + 200	12	x + 300	12	x + 400	12
0	1												
x	12												
x + 100	12												
x + 200	12												
x + 300	12												
x + 400	12												

It works in the same way on the TI:



Part II: Net Present Value – NPV

The NPV function computes the Net Present Value, Equivalent Periodic Value (or Net Uniform Series), and the Net Future Value of a cash flow series. The Equivalent Periodic Value may be either an annuity immediate or an annuity due. The default (d:=0) is for an annuity immediate; if you want an annuity due, enter 1 for d.

A 36-month lease has the following payment schedule and beginning-of-period payments. If the interest rate is 10% compounded monthly, what is the present value of the lease and what even payment at the beginning of each month would result in the same present value?

The variables in NPV are:

cf – cash flows

im – interest rate compounded mthly

ppy – number of periodic cash flows per year, defaults to 1

cpy – number of compounding periods per year, defaults to 1

d – defaults to 0, specifying an equivalent periodic value beginning at the end of period 1. To specify an equivalent periodic value beginning at the beginning of period 1, enter 1 for d.

$$\#17: \text{NPV} \left(\begin{bmatrix} 0 & 4 \\ 5000 & 8 \\ 0 & 3 \\ 6000 & 9 \\ 0 & 2 \\ 7000 & 10 \\ 0 & 1 \end{bmatrix}, 10, 12, 12, 1 \right)$$

$$\#18: \begin{bmatrix} \text{Nominal Interest Rate:} & 10\% \\ \text{Net Present Value:} & 138088.4358 \\ \text{Equivalent Periodic Value Due:} & 4418.901231 \\ \text{Net Future Value:} & 186168.3218 \end{bmatrix}$$

The NPV is \$138,088.44 and the equivalent periodic value due is \$4,418.90. You may be wondering why there is a final cash flow of 0 at the end of the cash flow list. This is because the program assumes that the number of payments in the annuity does NOT include the cash flow at time 0. In addition, the FV of an annuity is normally calculated at the end of the period whether or not the final payment is at the beginning or end of the period.

On the TI I make use of the built in Finance-functions. I give a screenshot of the procedure saved as a text file (script). Then I run the script which results in the values from above (fortunately). Some programming (collecting the functions into one program and designing a nice output) could lead to a table like the above one.

```

F1 Command View Execute Find...
:1st Example
:
C:tifinance.npv(10/12,0,(0,5000,0,6000,0,
7000),(3,8,3,9,2,10))
C:tifinance.tvm_fv(3,10,-138088.43582216,
0,1,12)
C:pmt_bgn()
C:tifinance.tvm_pmt(36,10,138088.43582216
,0,12,12)
FINANCE RAD AUTO FUNC

```

```

F1 Algebra Calc Other PrgmIO Clean Up
tifinance.npv(10/12,0,(0 5000 0 6000)
138088.44
tifinance.tvm_fv(3,10,-138088.43582216)
186168.32
pmt_bgn() Done
tifinance.tvm_pmt(36,10,138088.43582216)
-4418.90
10,138088.43582216,0,12,12
FINANCE RAD AUTO FUNC 4/30

```

The two columns of the cash flow matrix must be entered as lists.

The TI-functions don't provide a parameter to define variable d (payment due at the beginning or at the end of the periods), so it is necessary to "tell" the TI how to do. `pmt_bgn()` and `pmt_end()` must be executed prior for the respective cases. The TVM-Solver offers one line `PMT: END BEGIN`. All functions are available via `CATALOG > F3 Flash Applications`.

The next example is a straight forward investment. An investor makes an \$80,000 investment and expects returns over the next 5 years as illustrated in the cash-flow vector. What is the NPV assuming an annual interest rate of 10.5%?

#19: $[-80000, 5000, 4500, 5500, 4000, 115000]$

#20: $\text{NPV}([-80000, 5000, 4500, 5500, 4000, 115000], 10.5)$

#21:

Nominal Interest Rate:	10.5%
Net Present Value:	4774.632754
Equivalent Periodic Value:	1275.664871
Net Future Value:	7865.95329

```

F1 Command View Execute Find...
:2nd Example
:
C:tifinance.npv(10.5,-80000,(5000,4500,55
00,4000,115000))
C:4774.62*(1.105)^5
C:Tifinance.Pmt_End()
C:tifinance.tvm_pmt(5,10.5,4774.632754753
,3,0,1,1)
FINANCE RAD AUTO FUNC

```

```

F1 Algebra Calc Other PrgmIO Clean Up
tifinance.npv(10.5,-80000,(5000 4500)
4774.63
4774.62*(1.105)^5
7865.93
tifinance.pmt_end() Done
tifinance.tvm_pmt(5,10.5,4774.63275475)
-1275.66
10.5,4774.6327547533,0,1,1
FINANCE RAD AUTO FUNC 4/30

```

Part III: Internal Rate of Return – IRR

The IRR function is set up so that it will try to find all IRRs for a series 24 or less cash flow. For a series of more than 24 cash flows it will attempt to find an IRR between a lower and upper bound. The default lower bound is 0% and the default upper bound is 100%.

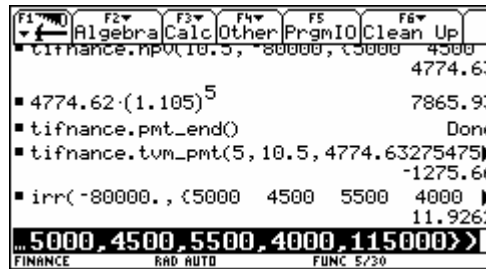
The variables used in IRR are:

- cf – cash flows
- ppy – number of periodic cash flows per year, defaults to 1
- cpy – number of compounding periods per year, defaults to 1
- l – lower bound for computing the IRR, defaults to 1%
- u – upper bound for computing the IRR, defaults to 100%

So, find the IRR for the investment in the last example.

#22: IRR([-80000, 5000, 4500, 5500, 4000, 115000])

#23: IRR = 11.92624987



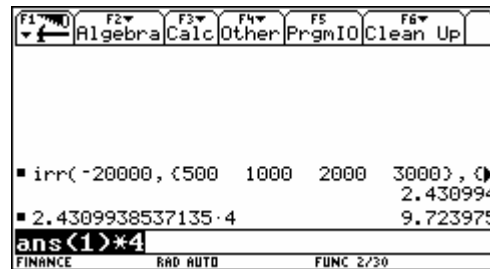
If your active folder is FINANCE then the preceding `tfinance.` is not necessary when entering the built in function names!

On an investment of \$20,000, you expect quarterly returns over the next four years of:

#24:
$$\begin{bmatrix} -20000 & 1 \\ 500 & 4 \\ 1000 & 4 \\ 2000 & 4 \\ 3000 & 4 \end{bmatrix}$$

Find the IRR.

#25:
$$\text{IRR} \left(\begin{bmatrix} -20000 & 1 \\ 500 & 4 \\ 1000 & 4 \\ 2000 & 4 \\ 3000 & 4 \end{bmatrix}, 4, 4 \right)$$



#26: IRR = 9.723975414

The periods are quarters, so we have to multiply the interest rate by four on the TI!

Since the payments and compounding periods were quarterly, 4 is entered for both ppy and cpy. The IRR is therefore 9.72% compounded quarterly. If you wanted to know the effective annual return, enter 1 for cpy.

#27:
$$\text{IRR} \left(\begin{matrix} -20000 & 1 \\ & 500 & 4 \\ & 1000 & 4 \\ & 2000 & 4 \\ & 3000 & 4 \end{matrix} \right), 4, 1$$

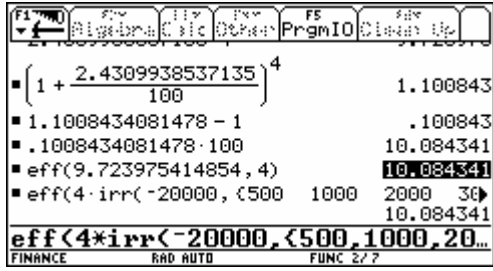
#28: IRR = 10.08434081

Or do this.

#29: ID(9.723975414, 4, 1)

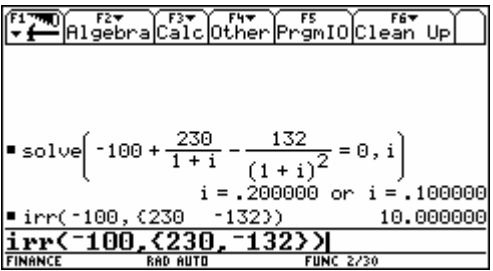
#30: 10.08434081

Or do this on the TI – you have several possibilities to perform the calculation:



A simple example of multiple rates.

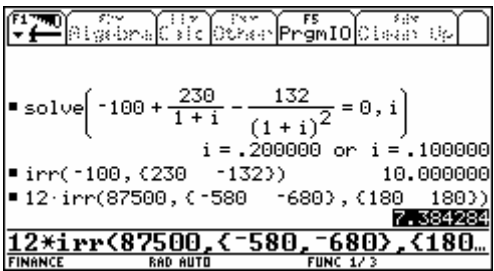
#31: IRR([-100, 230, -132]) = (IRR = [10, 20])



So we have two solutions for the internal interest rate. The TI-function irr delivers only one of them. Solving the respective – quadratic – equation results in both values for the irr.

To show Derive's ability to solve polynomials, it can find the IRR of 361 cash flows in less than a minute on a computer running Windows XP Professional with a 1.7 GHz processor and 736 MB of RAM.

#33:
$$\text{IRR} \left(\begin{matrix} 87500 & 1 \\ -580 & 180 \\ -680 & 180 \end{matrix} \right), 12, 12$$



The TI does it in an instant!

#34: IRR = 7.384283528

Part IV: Modified Internal Rate Of Return

The following example is a prime candidate for using the MIRR routine. It yields three positive nominal internal rates of return. The MIRR routine uses reinvestment and financing rates that you specify. Thus, negative cash flows are discounted at the financing rate and positive cash flows are accumulated at the reinvestment rate. An interest rate for the discounted and accumulated cash flows is then computed; this is called the modified internal rate of return.

The variables used in MIRR are:

cf – cash flows

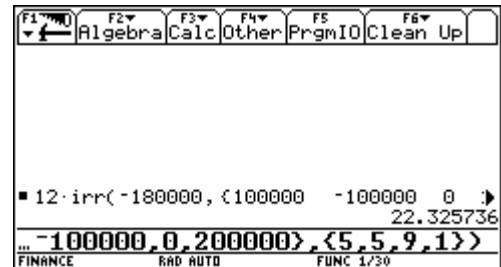
fr – financing rate of interest

rr – reinvestment rate of interest

ppy – number of periodic cash flows per year

cpy – number of compounding periods per year

$$\#35: \text{IRR} \left(\begin{bmatrix} -180000 & 1 \\ 100000 & 5 \\ -100000 & 5 \\ 0 & 9 \\ 200000 & 1 \end{bmatrix}, 12, 12 \right)$$



$$\#36: \text{IRR} = [22.32573567, 172.1765089, 345.7520213]$$

Calculate the MIRR with a financing rate of 8% and a reinvestment rate of 13%.

$$\#37: \text{MIRR} \left(\begin{bmatrix} -180000 & 1 \\ 100000 & 5 \\ -100000 & 5 \\ 0 & 9 \\ 200000 & 1 \end{bmatrix}, 8, 13, 12, 12 \right)$$

$$\#38: \text{MIRR} = 12.18283013$$

For those (few??) of you who are not so familiar with financial mathematics I'd like to make the Black Box MIRR a little bit grey by recalculating this modified internal rate of return in the traditional (conservative) way: We calculate the present value of all payments for the investment (negative amounts) by applying 8% (monthly compounded) and then the future value of all positive cash flows by applying the reinvestment rate of 13% due to the end of the investment period (end of month 20).

The present value of the negative cash flows turns out to be 654,136.81 and the future value of the positive cash flows is 800,582.75. Now we ask for the interest rate which makes from the present value the future value within the investment period. See the calculation:

TI-84 Plus calculator screen showing algebraic calculations. The display shows the equation $180000 + \frac{100000 \cdot v \cdot (1 - v^5) \cdot v^5}{1 - v} | v = \frac{1}{1 + \frac{.09}{12}}$ and the result 654136.808794 . Below it, the equation $\frac{100000 \cdot (r^5 - 1)}{r - 1} \cdot r^{15} + 200000 | r = 1 + \frac{.13}{12}$ is shown with the result 800582.754625 . The bottom of the screen shows the input $\rightarrow *r^{15} + 200000 | r = 1 + .13/12 \rightarrow fv1$.

TI-84 Plus calculator screen showing the zeros function. The display shows the equation $\frac{100000 \cdot (r^5 - 1)}{r - 1} \cdot r^{15} + 200000 | r = 1 + \frac{.13}{12}$ and the result 800582.754625 . Below it, the zeros function is used: $\text{zeros}(pv1 \cdot r^{20} - fv1, r)$ with the result $\{-1.010152, 1.010152\}$. The final answer is $\text{ans}(1) * 12 * 100$ resulting in 12.182830 .

One could use the TI-Finance-functions to calculate $pv1$ and $fv1$, but I wanted to apply “traditional” mathematics only – but not do without the V200 – what a contradiction!. Comparing this procedure with Don’s sophisticated function one really appreciates his product. Maybe that one or the other among you would like to write a TI-function for calculating the modified internal rate of return. Let us know!

Part V: The Equation of Value – EQV

The Equation of Value (EQV) routine computes an alternative series of cash flows that can replace a given cash flow series. The alternate series can have cash flows that increase or decrease by a given amount each period or group of periods and/or increase or decrease by a given percentage each period or group of periods.

The variables are:

- cf1 – first cash flow series
- cf2 – second cash flow series
- im – interest rate compounded mthly
- ppy – number of periodic cash flows per year
- cpy – number of compounding periods per year

If does not matter if cf1 is the original or alternate cash flow series; EQV handles either case.

You daughter is going to college in 12 years and you are starting a fund for her education. She will need \$15,000 at the beginning of each year for four years. The fund earns 9% compounded monthly, and you plan to make monthly deposits, starting at the end of the month and continuing until the last \$15,000 is paid out. How much do you need to deposit each month to meet her expenses?

The first \$15,000 is paid in 12 years or month 144. The last will be paid in 15 years or at month 180. This cash flow series can be entered as:

#39:

$$\begin{bmatrix} 0 & 144 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \end{bmatrix}$$

Notice that the first cash flow of 0 is repeated 144 times even though the first \$15,000 payment is made at month 144. This is because it also includes the time 0 payment of 0. It may be easier to keep things straight if you entered the cash flow as below where the time 0 cash flow is kept separate from the other cash flows.

$$\#40: \begin{bmatrix} 0 & 1 \\ 0 & 143 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \end{bmatrix}$$

The amount to be saved each month is entered as:

$$\#41: \begin{bmatrix} 0 & 1 \\ x & 180 \end{bmatrix}$$

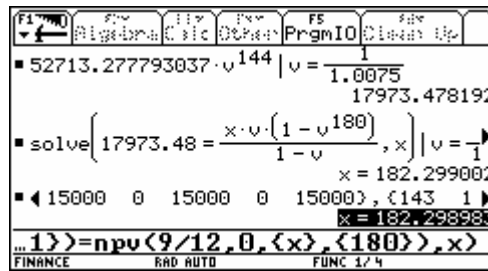
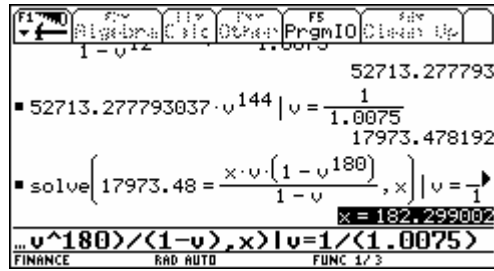
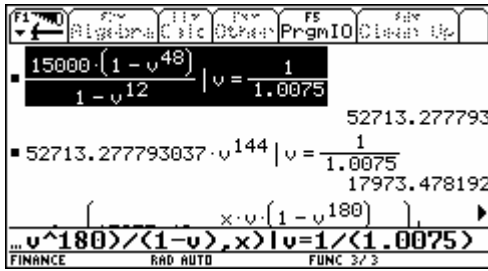
The unknown does not have to be x. It can be any letter.

Solving for x:

$$\#42: \text{EQV} \left(\begin{bmatrix} 0 & 144 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ x & 180 \end{bmatrix}, 9, 12, 12 \right)$$

$$\#43: \begin{bmatrix} \text{Group} & \text{Cash Flow} & \text{Value} & \text{\#CF's} \\ 1 & 0 & 0 & 1 \\ 2 & x & 182.298983 & 180 \end{bmatrix}$$

The required monthly deposit is \$182.30.



Syntax for solving the problem in one single line is:

$$\text{Solve}(\text{npv}(9/12, 0, \{0, 15000, 0, 15000, 0, 15000, 0, 15000\}, \{143, 1, 11, 1, 11, 1, 11, 1\}) = \text{npv}(9/12, 0, \{x\}, \{180\}), x)$$

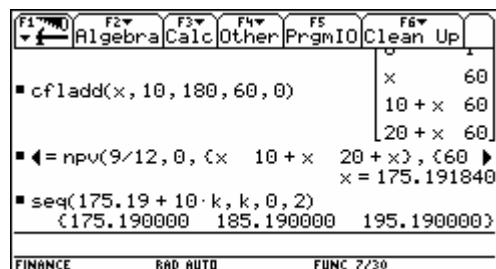
What if you wanted to increase your monthly deposit by \$10 every five years? This is where CashFlowAdd comes in handy.

#44: EQV $\left[\begin{array}{cc} 0 & 144 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \end{array} \right], \text{CashFlowAdd}(x, 10, 180, 60), 9, 12, 12$

#45:

Group	Cash Flow	Value	#CF's
1	0	0	1
2	x	175.1918397	60
3	x + 10	185.1918397	60
4	x + 20	195.1918397	60

This again is much more comfortable using Don's specialized function. But we can solve the problem on the TI without any serious difficulties, too.

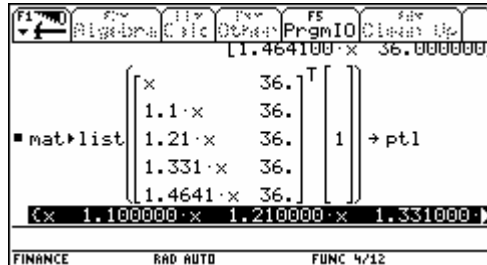
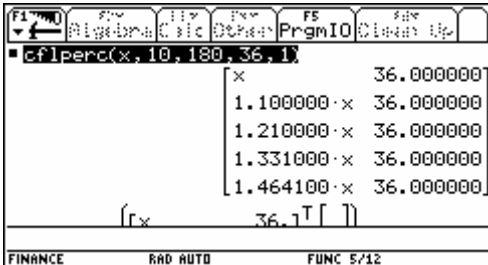


Or, suppose you wanted to increase your deposits by 10% every 3 years.

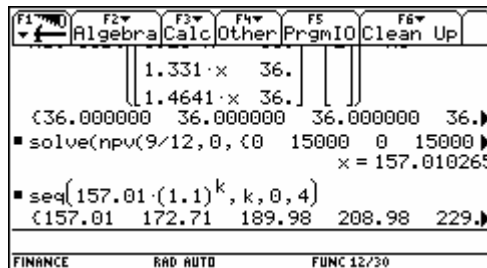
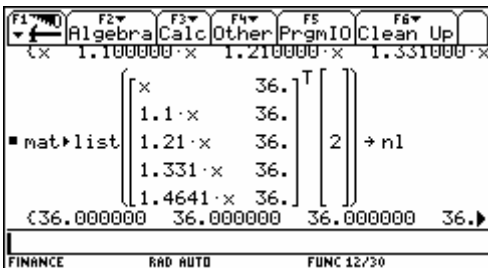
#46:
$$\text{EQV} \left(\begin{bmatrix} 0 & 144 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \\ 0 & 11 \\ 15000 & 1 \end{bmatrix}, \text{CashFlowPercent}(x, 10, 180, 36), 9, 12, 12 \right)$$

#47:

Group	Cash Flow	Value	#CF's
1	0	0	1
2	x	157.0102647	36
3	1.1·x	172.7112912	36
4	1.21·x	189.9824203	36
5	1.331·x	208.9806624	36
6	1.4641·x	229.8787286	36



I extract the two columns of the matrix and convert them to lists **pt1** and **n1** which can be used as function parameters in **npv()**.



The full line which gives the result $x = 157.01$, which is the first payment reads as follows – and is very similar to the expression given on page 22:

$$\text{solve}(\text{npv}(9/12, 0, \{0, 15000, 0, 15000, 0, 15000, 0, 15000\}, \{143, 1, 11, 1, 11, 1, 11, 1\}), x)$$

If, for any reason, you want to extract the results of EQV to use in other calculations, use the function ExtractValues.

	Group	Cash Flow	Value	#CF's
#48: ExtractValues	1	0	0	1
	2	x	157.0102647	36
	3	1.1·x	172.7112912	36
	4	1.21·x	189.9824203	36
	5	1.331·x	208.9806624	36
	6	1.4641·x	229.8787286	36

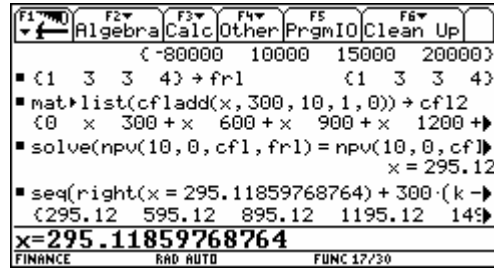
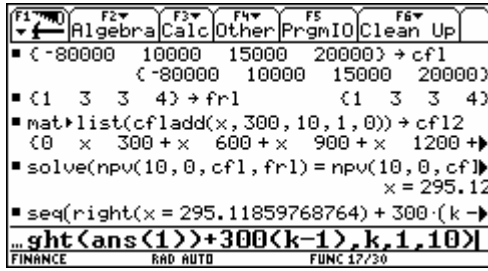
#49:	0	1
	157.0102647	36
	172.7112912	36
	189.9824203	36
	208.9806624	36
	229.8787286	36

There may be times when you want to compute an alternate cash flow series that increases (decreases) by a given amount each period, but you are unsure of the sign of the NPV of the original cash flow series, i.e., whether it is a net cash inflow or a net cash outflow. If it is a net cash inflow, you want your alternate series to be positive and increasing (decreasing) by the given amount. If it is a net cash outflow, you want your alternate series to be increasing (decreasing) also, but have a negative sign. The EQV routine handles such cases.

$$\#50: \text{EQV} \left(\begin{bmatrix} -80000 & 1 \\ 10000 & 3 \\ 15000 & 3 \\ 20000 & 4 \end{bmatrix}, \text{CashFlowAdd}(x, 300, 10), 10 \right)$$

	CF #	Cash Flow	Value
#51:	0	0	0
	1	x	295.1185976
	2	x + 300	595.1185976
	3	x + 600	895.1185976
	4	x + 900	1195.118597
	5	x + 1200	1495.118597
	6	x + 1500	1795.118597
	7	x + 1800	2095.118597
	8	x + 2100	2395.118597
	9	x + 2400	2695.118597
	10	x + 2700	2995.118597

Look how I am solving this problem on the TI:

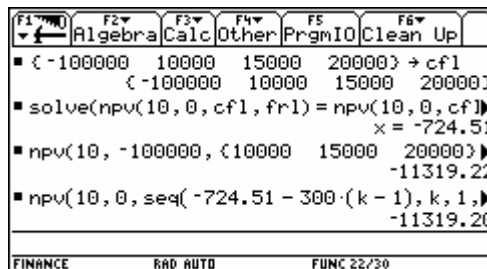
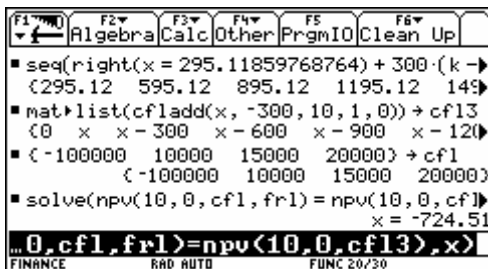


#52:
$$EQV \begin{pmatrix} -100000 & 1 \\ 10000 & 3 \\ 15000 & 3 \\ 20000 & 4 \end{pmatrix}, \text{CashFlowAdd}(x, 300, 10), 10$$

#53:

CF #	Cash Flow	Value
0	0	0
1	-x	-724.5129929
2	-x - 300	-1024.512992
3	-x - 600	-1324.512992
4	-x - 900	-1624.512992
5	-x - 1200	-1924.512992
6	-x - 1500	-2224.512992
7	-x - 1800	-2524.512992
8	-x - 2100	-2824.512992
9	-x - 2400	-3124.512992
10	-x - 2700	-3424.512992

Thus, if the NPV is negative, the alternate cash flows are negative indicating that they are cash outflows and that the outflows are in fact increasing each period by 300.



If you compute the NPVs of both cash flows in the last example, you will find that they are, in fact, equal.

#54: NPV $\left(\begin{bmatrix} -100000 & 1 \\ 10000 & 3 \\ 15000 & 3 \\ 20000 & 4 \end{bmatrix}, 10 \right)$

#55: $\left[\begin{array}{l} \text{Nominal Interest Rate:} \quad 10\% \\ \text{Net Present Value:} \quad -11319.22133 \\ \text{Equivalent Periodic Value:} \quad -1842.151146 \\ \text{Net Future Value:} \quad -29359.145 \end{array} \right]$

#56: NPV $\left(\begin{array}{l} \text{ExtractValues} \\ \left[\begin{array}{l} \text{CF \#} \quad \text{Cash Flow} \quad \text{Value} \\ 0 \quad 0 \quad 0 \\ 1 \quad -x \quad -724.5129929 \\ 2 \quad -x - 300 \quad -1024.512992 \\ 3 \quad -x - 600 \quad -1324.512992 \\ 4 \quad -x - 900 \quad -1624.512992 \\ 5 \quad -x - 1200 \quad -1924.512992 \\ 6 \quad -x - 1500 \quad -2224.512992 \\ 7 \quad -x - 1800 \quad -2524.512992 \\ 8 \quad -x - 2100 \quad -2824.512992 \\ 9 \quad -x - 2400 \quad -3124.512992 \\ 10 \quad -x - 2700 \quad -3424.512992 \end{array} \right] \end{array} \right), 10$

#57: $\left[\begin{array}{l} \text{Nominal Interest Rate:} \quad 10\% \\ \text{Net Present Value:} \quad -11319.22133 \\ \text{Equivalent Periodic Value:} \quad -1842.151146 \\ \text{Net Future Value:} \quad -29359.14499 \end{array} \right]$

If you have any questions or suggestions for improvement, please feel free to send email.

Financial Mathematics IV will cover the Black-Scholes or European Options Model and American Call Options using the bivariate normal distribution.

- [1] Financial Mathematics I (Bond, Price and Yield), DNL#52
- [2] Financial Mathematics II (TVM and Amortization), DNL#55

The following contribution enable simulating random games using DERIVE-functions. The very special parts are the graphic realisation of coins, dice, wheel of fortune and roulette. Studying Lorenz' files is very useful for those among you who want to learn how to produce graphic representations. More can be found on his very recommendable website

<http://lzkopp.gmxhome.de/index.htm>

We will present more examples in one of the future DNLs. Josef

Four Graphic Simulations of Random Games

Lorenz Kopp

#2 - #14: Utility-file to produce the numbers 0 to 99

(Prepared to save it as a mth-file. You find it saved as numbut .mth among the files)

1. "Throw Coins"

`rd(k_)`: Random-number 1 ... k_

`kd`: Constant with last value of `rd(k_)`

`coingraph(c_, r_, i_)`: Circle marked with "1" or "0"

Settings of 2-D-plot window for the following three examples:

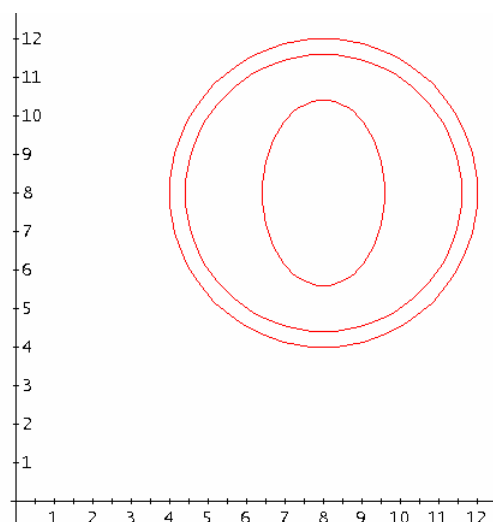
Tile vertically, reset plot range, zoom out once, reset aspect ratio, cross position (7 / 7), cross as center ($-1 \leq x, y \leq 15$), simplify before plotting, points connect, points small, grids off.

Next 3 expressions:

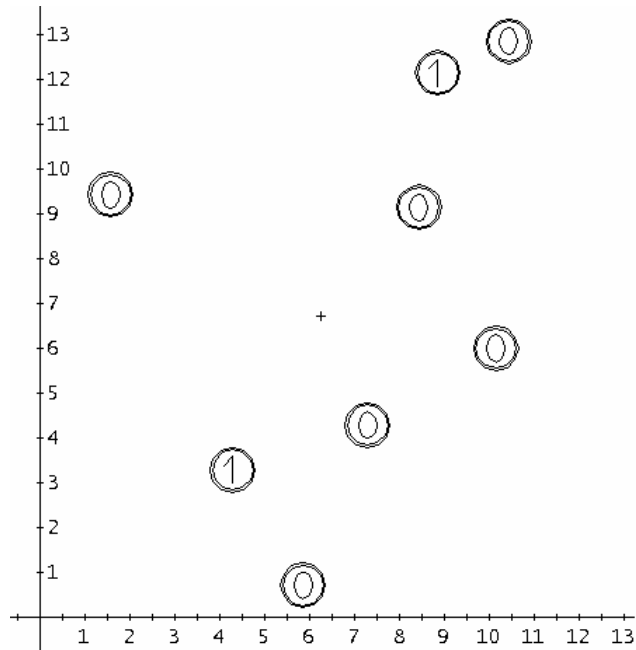
1. One "coin" center (8/8), $r = 4$,
2. One coin, center (by random function) $4/7 \leq x, y \leq 100/7$, $r = 1/2$ (repeat plotting),
3. 20 throws of a coin randomly distributed

Random(0)

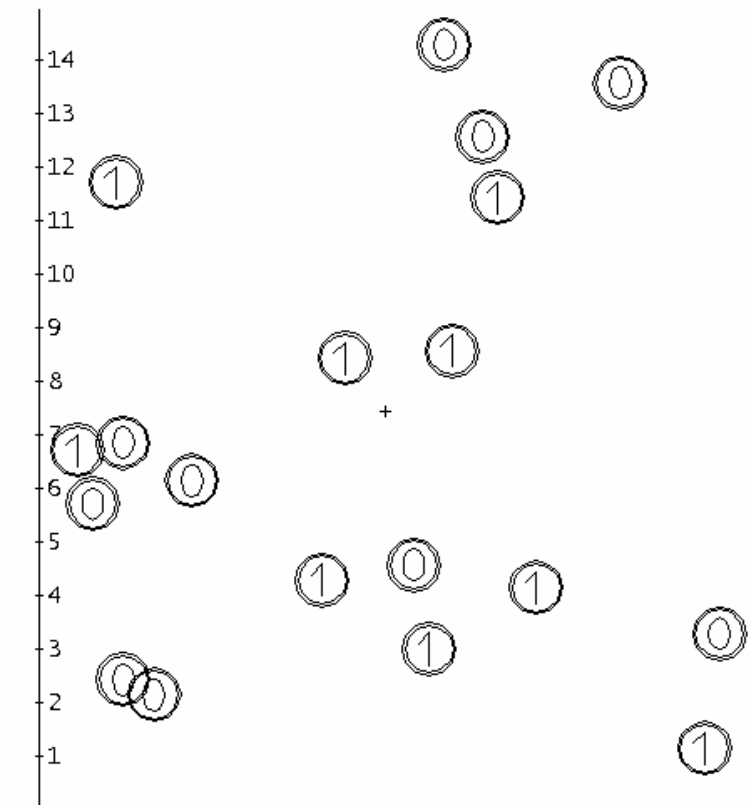
`coingraph([8,8],4,36)`



$$\text{coingraph}\left(\left[\frac{\text{ran}(4,100)}{7}, \frac{\text{ran}(4,100)}{7}\right], \frac{1}{2}, 6\right)$$



$$\text{VECTOR}\left(\text{coingraph}\left(\left[\frac{\text{ran}(4,100)}{7}, \frac{\text{ran}(4,100)}{7}\right], \frac{1}{2}, 6\right), i, 1, 20\right)$$



2. "Wheel of Fortune"

`rotfig(f_, d_)`: Rotates a vector `f_` of polygons, center (0/0), angle `d_`

`wheelgraph(k_, r_)`: Circle with `k_` sectors, radius `r_`, center (0/0), a marking triangle on the top, area of circles shaded from 0 to $0.7r_$, step $0.1r_$

(The third element `VECTOR(crcar()...`) only seems to imitate a rotation, it may be deleted.)

`rotnum(k_, r_)`: Plots the numbers 1 to `k_` equidistant on a circle, number `kd` on the highest point

`rotwheel(k_, r_)`: Executes `wheelgraph`, random-number `rd` and `rotnum`

2D-plot-settings for the following example:

($r_ < 1$ to avoid plotting the line $y = kd$ ($1 \leq kd \leq k_$) in the plot range)

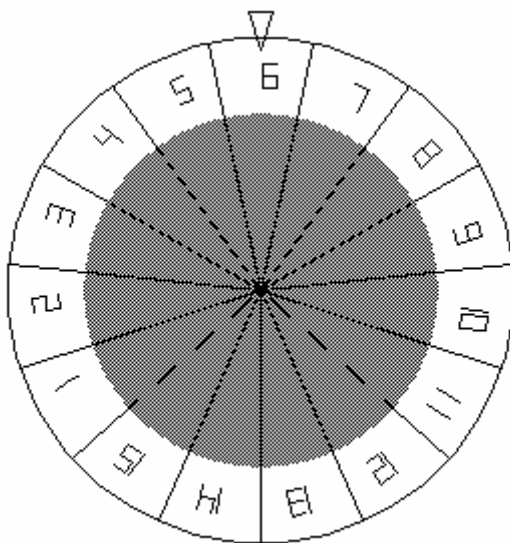
Axes: lines, labels, titles off, Grids: off, Points: connect, small, Approximate before plotting.

Windows: tile vertically, plot range reset, zoom in twice ($-0.8 \leq x, y \leq 0.8$), reset aspect ratio.

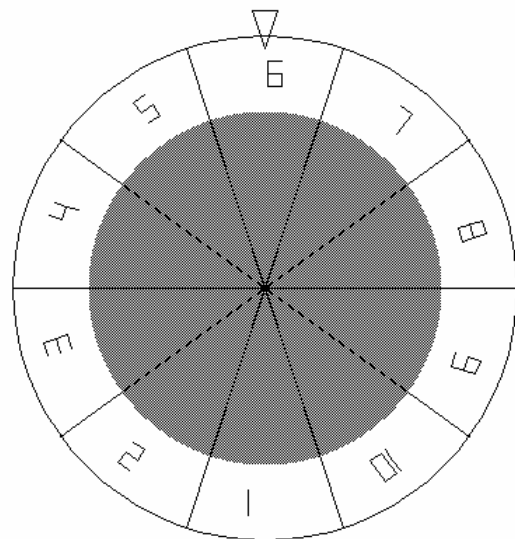
(The preset values in #6 can be used till $k_ = 30$ (e.g. 31 ... 45: `yu:=0.85, fh:=0.08`.)

`random(0)`

`rotwheel(15,0.7)`



`rotwheel(10,1)`



3. "Play Dice" with 1 to 4 Dice

`pt(c_, r_)`: #32 points plotted as 4 circles (faster) or #33 points shaded (slower)

`allpt(c_, a_)`: 7 points with order

4	3
6	1 7
2	5

`dice(c_, a_, k_)`: Square with center `c_`, length $2a_$ with `k_` points

`rn(n_)`: Simplify produces a vector of `n_` random numbers 1 ... 6,

`kn`: Constant with last value of `rn(n_)`

`cent(a_)`: Centers of 4 squares (1. - 4. quadrant)

`play(a_, n_)`: Plots `n_` dice

`playsum(a_, n_)`: Sum of points added

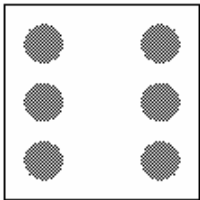
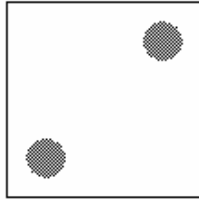
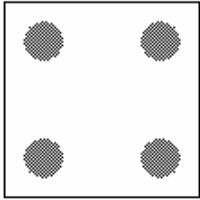
The same settings of 2-D-plot window as in nr. 2.

Mark one of the functions in the next two lines and plot.

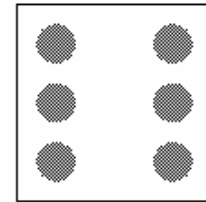
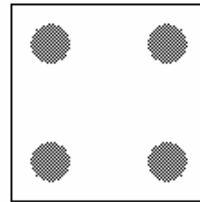
```
random(0)
```

```
[play(0.3,1),play(0.3,2),play(0.3,3),play(0.3,4)]
```

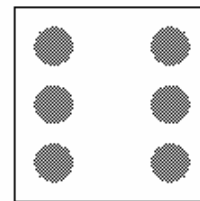
```
[playsum(0.3,1),playsum(0.3,2),playsum(0.3,3),playsum(0.3,4)]
```



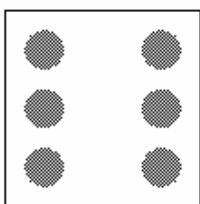
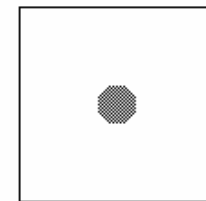
play(0.3,3)



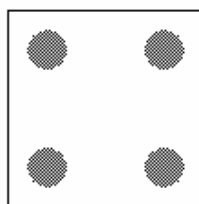
play(0.3,2)



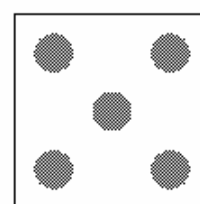
H



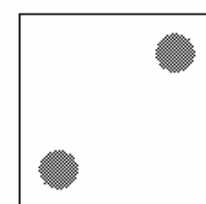
□



playsum(0.3,2)



playsum(0.3,4)



4. "Roulette"

`list`: List of roulette-numbers

`rotlist(r_)`: Numbers plotted equidistant on a circle

`noir(r_)`: 37 radial lines, black sectors marked with 8 shorter lines

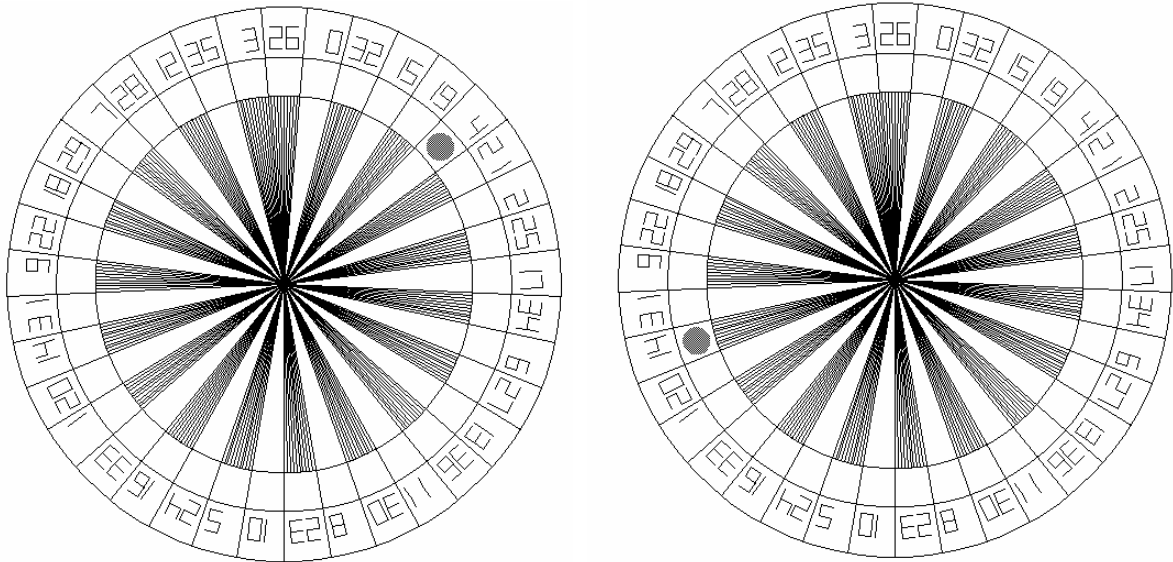
`roulette(r_)` plots 3 circles, radial lines and numbers (roulette wheel)

`pellet(r_)` plots dependent on random number a shaded circle under one number

Plot first **roulette** (slow), then **pellet** (fast). Delete last plot (only pellet) and repeat plotting pellet.

Set $yu:=0.85$, $fh:=0.08$.

[roulette(0.7),pellet(0.7)]



DERIVE- and CAS-TI-User Forum 2

continued from page 5

```
SHAPE(a, d := {}, n, m, l) :=
  Prog
  If d = {}
    RETURN DIMS(a)
  If a = []
    RETURN "REPARTITION: err: Can't repartiton EMPTY"
  If ~ VECTOR?(a)
    RETURN REPARTITION([a], d)
  If ~ VECTOR?(d)
    RETURN REPARTITION(a, [d])
  If d = []
    RETURN (RAVEL(a))&#8595;1
  n := &#8719;(d)
  m := DIM(a)
  l := DIM(d)
  a := (ITERATE(APPEND(a, x), x, a, n/m))&#8595;[1, ..., n]
  (ITERATE([PARTITION(x, d&#8595;y), y - 1], [x, y], [a, l], l - 1))
  &#8595;1
```

This now works like the "reshape" (rho) function in APL, monadically and dyadically.

-drl

Lunsford 21.2.2006

One more refinement to make it recognize strings:

```
SHAPE(a, d := {}, n, m, l) :=
  Prog
  If d = {}
    If STRING?(a)
      RETURN [DIM(a)]
      RETURN DIMS(a)
  If a = []
    RETURN "SHAPE: err: Can't reshape EMPTY"
  If NOT VECTOR?(a)
    RETURN SHAPE([a], d)
  If NOT VECTOR?(d)
    RETURN SHAPE(a, [d])
  If d = []
    RETURN (RAVEL(a)) SUB 1
  n := PRODUCT(d)
  m := DIM(a)
  l := DIM(d)
  a := (ITERATE(APPEND(a, x), x, a, n/m)) SUB [1, ..., n]
  (ITERATE([PARTITION(x, d SUB y), y - 1], [x, y], [a, l], l - 1))
  SUB 1
```

Set Operations

Dr. Lunsford, 17.2.06 08:39

Derive 6.1 gives $\{m\}^{\{n\}}$ as $\{[e^N, M]\}$. You can extract the member with FIRST

FIRST($\{m\}^{\{n\}}$) is $[e^N, M]$

So FIRST($\{m\}^{\{n\}}$) SUB 1 is e^n etc.

This behavior seems consistent, if you add elements you get for example

$\{M,N,P,Q\}^{\{A,B,C,D\}}$ returns

```
{[e^c, N], [e^c, m], [e^c, p], [e^c, q], [e^D, N], [e^D, m], [e^D, p],
[e^D, q], [e^B, N], [e^B, m], [e^B, p], [e^B, q], [e^A, N], [e^A, m], [e^A,
p], [e^A, q]}
```

What on earth is this about? Note the odd order of symbols, and their capitalization. This behavior is not documented with the product.

If you think of $A^B = e^{(B \log A)}$ we can write in Derive

$\{e^N, M\}$ as $e^{\{N, \log M\}}$

so this in a sense is the explanation of $\{M\}^{\{N\}}$.

I assume all this is well known?

Any pointers to online resources would be appreciated. I notice this list is not very active. Is there a compiled bug list? Idiom list? What a frustrating product.

-drl

Valeriu Anisiu

17.2.06, 15:37

If A,B are sets, A^B is not defined in Derive, so it should be returned unevaluated. (only A^n makes sense, where n is a positive integer). The fact that A^B returns some garbage is a bug but not a very serious one.

V. Anisiu

Confusion with Complex Numbers – imaginary part zero!!

Heinrich Ludwig

28.2.2006

Sehr geehrter Herr Böhm,

ich bin auf einen sehr subtilen Fehler in DERIVE 6.1 gestoßen, der wegen seines gehässig versteckten Auftretens vielleicht noch nicht bekannt ist.

Bemerkenswerterweise tritt er in DERIVE 5 nicht auf. DERIVE 6.1 läuft bei mir auf Windows 98, Built 4.10.2222A.

Bitte betrachten Sie folgenden DERIVE-Text. Da mth-Dateien dankenswerterweise noch ASCII-Text sind, habe ich der Einfachheit halber den Code hier eingefügt. Ich bitte Sie, ihn für das Weitere in eine mth-Datei zu kopieren.

```
F(x,m,p):=PROG(m:=x/(-3+0*#i),p:=m^2,RETURN([m,p]))

F([-3,6],[-6,0])

;Simp(#2)
[[[1,-2],[2,0]],[[-3,-2],[2,?]]]
```

Wenn Sie die Funktion F mit der angegebenen Matrix, also #2, auswerten, erhalten Sie zwei Matrizen. Die erste ist, wie man dem Code entnimmt, der (-3)-te Teil des Funktions-Arguments. Soweit ist alles in Ordnung. Quadriert man diese Matrix wiederum, so stellt sich der Fehler ein. Er besteht darin, dass $-3+0\cdot i$ intern nicht gleich -3 zu sein scheint.

Allerdings kann man durch keine der üblichen Maßnahmen (DISPLAY, Vergleich mit -3 ergibt true, usw.) den Unterschied erkennen. Auch die Matrix m schaut korrekt aus. Erst beim Quadrieren zeigt sich, daß m doch nicht gleich $[1, -2; 2, 0]$ ist, obwohl es nach außen den Anschein hat und Tests der Koeffizienten mit INTEGER?(·) true ergeben. Auch INTEGER?($3+0\cdot i$) wird vernünftigerweise zu true vereinfacht.

Ich bin auf den Fehler gestoßen, als ich die von Herrn Prof. Wiesenbauer veröffentlichte Funktion EXTENDED_GCD verwendet habe. Sie enthält die Zeile

```
q_ := ROUND(RE(q_)) + ROUND(IM(q_))*#i
```

Dabei kommt es durchaus vor, dass der Imaginärteil von $q_$ Null ist.

Verwendet man den Output der Funktion weiter, dann bekommt man das oben geschilderte Problem, jedenfalls mit DERIVE 6.1. Natürlich hat Prof. Wiesenbauer vollkommen korrekt programmiert und zu DERIVE-5-Zeiten hat sich der Code auch noch richtig verhalten.

Heinrich Ludwig came across a strange fact, that working with a complex number having the imaginary part 0 (i.e. $a + 0i$) can lead to serious problems. Internally a and $a + 0i$ seem to be different as one can see following the matrix-operation given above (matrix m divided by $-3+0i$ and then squared). Heinrich found this bug? first when using Johann Wiesenbauer's EXTENDED_GCD which can show 0 as imaginary part. Using the output of this function leads immediately to the problem. In DERIVE 5 one cannot find this strange behaviour.

As Mr Ludwig refers to Johann's function I sent this mail to him and as usual he immediately resent an answer:

Johann Wiesenbauer

2.3.2006

Lieber Josef!

Leider muss ich Herrn Ludwig recht geben, es liegt hier ein offensichtlicher Bug in der aktuellen Version von Derive vor. In seiner einfachsten Form kann man ihn erkennen an der Funktion $h(x)$, die folgendermaßen definiert ist:

$$h(x) := \text{PROG}(x := x \cdot (1 + 0 \cdot \#i), [x, x \cdot x])$$

Damit ist z.B. $h([1, 2]) = [[1, 2], 5]$

was durchaus korrekt ist, aber eben $h([1, 0]) = [[1, 0], ?]$

d.h. das Problem ist, dass hier zwei Nullen aufeinanderstossen, die intern offenbar als verschieden behandelt werden.

Ich kann daher nur empfehlen, diesen Bug in der Derive 6.1 umgehend an Albert Rich weiterzuleiten, damit er im nächsten Patch behoben wird.

Johann Wiesenbauer

2.3.2006, 5 hours later

Hier noch ein einfacheres Beispiel für den Bug, welches zeigt, dass das Ganze mit Vektoren und Matrizen gar nichts zu tun hat.

$$g(x) := \text{PROG}(x := 1 + 0 \cdot \#i, [x, x \cdot x])$$

Es ist $g(5) = [5, 25]$

Aber $g(0) = [0, ?]$

Bis zum nächsten Patch, in dem das hoffentlich behoben sein wird, fällt mir auch keine bessere vorläufige "Reparatur" als

$$g(x) := \text{PROG}(x := 1 + 0 \cdot \#i, x := \text{if}(x=0, 0, x), [x, x \cdot x])$$

ein.

Johann sent two mails within 5 hours confirming Heinrich Ludwig's findings. This is a bug and it should be fixed in the next update of DERIVE. The problem is that two kinds of zeros are appearing. The bug has nothing to do with vectors and matrices. This is the content of Johann's 2nd mail giving a possibility to "repair" this deficiency for his example of a multiplication by $1+0i$ (see above).

Heinrich Ludwig

3.3.2006, 12:15

Sehr geehrter Herr Böhm,

vielen Dank, dass Sie so schnell auf meine jüngste Fehlermeldung reagiert haben. Das ermutigt mich, Ihnen von einem weiteren Bugs in DERIVE zu berichten. Ich werde Sie noch über einen weiteren Fehler unterrichten, brauche aber noch etwas Zeit zur genauen Analyse.

Der neue Fehler hängt mit dem vor kurzem geschilderten zusammen, ist aber nicht mit ihm identisch, da er sowohl in DERIVE 5.06 als auch in DERIVE 6.1 auftritt. Er hat offenbar damit zu tun, dass DERIVE Schwierigkeiten hat, komplexe Zahlen, deren Imaginärteil gleich Null ist mit reellen Zahlen zu identifizieren. Ich möchte nicht sagen, dass DERIVE das nicht könne oder tue, sondern nur, dass diese Identifikation verzögert geschieht, was beim Programmieren zu Schwierigkeiten führt, die zu umgehen unnötige "Workarounds" erfordert.

DERIVE erkennt nicht von Haus aus, dass $z \cdot \text{CONJ}(z)$ stets reell ist, sondern führt die Konjugation und anschließend die Multiplikation aus. Dabei ergibt sich (aus Sicht von DERIVE zufällig), dass der Imaginärteil Null ist. Übergibt man das Ergebnis einer Funktion, die prinzipiell nur reellwertige Argumente akzeptiert, wie MAX, MIN, GCD, LCM etc., so sollte dies für DERIVE Anlass sein, das übergebene Argument versuchsweise zu einer reellen Zahl zu vereinfachen. Das tut DERIVE merkwürdigerweise

nicht konsequent. Sie finden das dargestellt in einer mth-Datei im Anhang dieses E-Mails. Interessant ist, dass die von den Funktionen a und b erzeugten Matrizen bei weiterer Vereinfachung (CTRL-B) richtig sind, d.h. dass nicht zu Ende vereinfacht wurde. Hierin unterscheidet sich der heute gemeldete Bug von dem vorgestern gemeldeten wesentlich.

Natürlich lässt sich das Fehlverhalten leicht vermeiden, wenn man das Betragsquadrat der komplexen Zahl mit $\text{RE}(\cdot)$ kapselt oder gleich stattdessen $\text{RE}(z)^2 + \text{IM}(z)^2$ schreibt. Dennoch bleibt ein Fehler ein Fehler.

Ich möchte auch gerne "Positives" beitragen. Mir gefallen Ihre Tipps und Tricks recht gut, die Sie veröffentlichen, und hätte da ein bisschen was, was vielleicht dazupassen würde. Das möchte ich Ihnen aber in getrennter Post zuschicken, sonst wird diese Nachricht zu lang.

Mit freundlichem Gruß
Heinrich Ludwig

FEHLER.MTH: Fehlerdemonstration.

$f(z) := \text{MAX}(1, z \cdot \text{CONJ}(z))$

$g(z) := \text{GCD}(1, z \cdot \text{CONJ}(z))$

$h(z) := \text{GCD}(0, z \cdot \text{CONJ}(z))$

$a := \text{VECTOR}(\text{VECTOR}(f(x + i \cdot y), x, -2, 2), y, -2, 2)$

$b := \text{VECTOR}(\text{VECTOR}(g(x + i \cdot y), x, -2, 2), y, -2, 2)$

$c := \text{VECTOR}(\text{VECTOR}(h(x + i \cdot y), x, -2, 2), y, -2, 2)$

a

$$\begin{bmatrix} 8 & 5 & \text{MAX}(1, 4) & 5 & 8 \\ 5 & 2 & \text{MAX}(1, 1) & 2 & 5 \\ \text{MAX}(1, 4) & \text{MAX}(1, 1) & \text{MAX}(1, 0) & \text{MAX}(1, 1) & \text{MAX}(1, 4) \\ 5 & 2 & \text{MAX}(1, 1) & 2 & 5 \\ 8 & 5 & \text{MAX}(1, 4) & 5 & 8 \end{bmatrix}$$

b

$$\begin{bmatrix} 1 & 1 & \text{GCD}(1, 4) & 1 & 1 \\ 1 & 1 & \text{GCD}(1, 1) & 1 & 1 \\ \text{GCD}(1, 4) & \text{GCD}(1, 1) & \text{GCD}(1, 0) & \text{GCD}(1, 1) & \text{GCD}(1, 4) \\ 1 & 1 & \text{GCD}(1, 1) & 1 & 1 \\ 1 & 1 & \text{GCD}(1, 4) & 1 & 1 \end{bmatrix}$$

c

$$\begin{bmatrix} 8 & 5 & 4 & 5 & 8 \\ 5 & 2 & 1 & 2 & 5 \\ 4 & 1 & 0 & 1 & 4 \\ 5 & 2 & 1 & 2 & 5 \\ 8 & 5 & 4 & 5 & 8 \end{bmatrix}$$

In his second mail Heinrich Ludwig reports about another bug? which occurs not only in DERIVE 6.1 but also in DERIVE 5.06. DERIVE seems to have difficulties to identify complex numbers with imaginary part 0 with real numbers. It is not that DERIVE cannot do this, but it needs an additional simplifying step which can lead to problems within a program and requires some unnecessary "workarounds".

*So DERIVE does obviously not recognize that $z * \text{CONJ}(z)$ always gives a real result, but first finds the conjugate and then performs the multiplication. This leads – by chance from DERIVE's point of view – to imaginary part 0. Using the product in a function which accepts only real arguments like MAX; MIN; GCD, LCM etc, DERIVE should try to simplify it to a real number, but DERIVE does not. You can find a documentation of this behaviour in the attached MTH-file (see above). It is very interesting that the matrices produced by a and b will become correct after an additional simplify (CTRL+B).*

Of course one can avoid this incorrect behaviour by encapsulating the squared absolute value with RE() or writes $RE(z)^2 + IM(z)^2$. But in my opinion a bug remains a bug.

Regards

Heinrich Ludwig

3.3.2006, 14:00

Sehr geehrter Herr Böhm,

nachdem ich Ihnen in kurzer Folge geschrieben habe, entschuldigen Sie bitte, dass ich's nochmal tue. Ich habe als Lehrer in Bayern gerade Ferien und das ist die einzige Zeit, in der man endlich einmal Zeit hat, Dinge nicht nur oberflächlich abzutun. Ich möchte Sie keineswegs in irgendeiner Weise drängen, weil ich so kurz aufeinanderfolgend schreibe.

Wussten Sie schon, dass $5374978561 - 2403763488 \cdot \sqrt{5} = 0$? DERIVE 6.1 ist gelegentlich der Meinung, auch im Vereinfachungsmodus "Exact". Das Verhalten trat in der Version 5 noch nicht auf, wie ich unten darlege, deshalb möchte ich es als Fehlverhalten einstufen.

Wenn Sie in der anhängenden Datei Fehler3.mth nachsehen, erkennen Sie in der Funktion GGT den Euklidischen Algorithmus. Zu Analyse Zwecken ist die Display-Anweisung eingefügt. Wendet man den Algorithmus auf zwei nicht kommensurable Zahlen an, z.B. 1 und $\sqrt{5}$, dann sollte die Abbruchbedingung $b=0$ nie erfüllt sein. Mit DERIVE 6.1 terminiert GGT aber nach endlich vielen Schritten, weil aus unbekanntem Gründen die o.g.

merkwürdige Gleichheit eintritt. Die Erscheinung ist nicht singulär, man kann leicht auch andere Wurzeln nehmen. Sicherlich ist die o.g. Differenz eine recht gute Approximation an Null ($9.3E-11$), aber DERIVE sollte doch "exact" rechnen! Im Modus "mixed" ist an DERIVEs Verhalten nichts auszusetzen. Interessant ist: Wenn man die o.g. Differenz alleine, also ohne den Kontext der Funktion GGT, eingibt und dann vereinfacht (Strg-B), wird sie nicht zu null, genau wie es sein soll.

Da ich schon einmal die Erfahrung gemacht habe, dass sich von Version 5 auf 6 allerhand Internes verändert hat, habe ich auch Version 5 das Problem vorgelegt. Zunächst erkennt man, dass die alte Version im Gegensatz zur neuen $\text{ROUND}(\sqrt{5})$ im Modus "exact" nicht weiter vereinfacht. Insofern ist die neue Version eine Verbesserung. Um Version 5 die Funktion GGT sinngemäß ausführen zu lassen, muss man zum Runden $\text{ROUND}(\text{APPROX}(a/b, n))$ setzen. (Dies ist in GGT2 ausgeführt.) Nun zeigt sich

Erstaunliches: Beide Versionen verhalten sich 17 Schleifendurchläufe lang gleich. Während Version 6.1 nach der 17. abbricht, rechnet Version 5 weiter und kommt nur zum Absturz, weil irgendwann a/b bei der eingestellten Stellenzahl zu "+-inf" approximiert wird. Also enthält Version 5 den genannten Fehler nicht.

Darf ich ehrlich sein? Ich neige jetzt immer mehr dazu, mich wieder an Version 5 zu halten. Sollte ich einmal das nützliche Degug-Feature DISPLAY brauchen, so werde ich für zwischendrin ab und zu die Sechs-Einser nehmen.

Schade.

Heinrich Ludwig's 3rd report: In his mail he points out that DERIVE sometimes has the opinion that $5374978561 - 2403763488 \cdot \sqrt{5} = 0$ even in Exact Mode. In the attached file FEHLER3.MTH you will recognize the Euklid algorithm in function GGT. For analysing purpose I inserted the DISPLAY instruction. Applying the GGT on two incommensurable numbers as 1 and $\sqrt{5}$ then the process should never end. DERIVE 6.1 terminates GGT after some steps. DERIVE 5 produces an infinite loop which is correct.

Surely $9.3E-11$ is a fine approximation for zero but I expect DERIVE to work "EXACT" in Exact Mode. In Mixed Mode everything is ok. Interesting detail simplifying the above difference (CTRL+B) does not result in 0.

Version 5 does not simplify $\text{ROUND}(\sqrt{5})$ in Exact Mode – this is an improvement in version 6. For performing GGT in version 5 one has to set $\text{ROUND}(\text{APPROX}(a/b, n))$ – see in GGT2. We can observe the following:

Both versions show the same behaviour for the first 17 loops. Version 6.1 stops then but version 5 goes on calculating and crashes at a certain moment because a/b will be simplified to "+-inf". So Derive 5 works exact.

```
Precision := Exact
PrecisionDigits := 10
Notation := Decimal
GGT(a, b, q, r) :=
  Loop
  If b = 0
    RETURN a
  q := ROUND(a/b)
  r := a - q*b
  a := b
  b := r
  DISPLAY([b, APPROX(b, 27)])

GGT2(a, b, q, r, i := 0) :=
  Loop
  i :=+ 1
  If b = 0
    RETURN a
  q := ROUND(APPROX(a/b, 27))
  r := a - q*b
  a := b
  b := r
  If i = 17
    RETURN [b, a]
```

GGT(1, $\sqrt{5}$)

[31622993. $\sqrt{5}$ - 70711162, 0.000000007071019424]

[299537289 - 133957148. $\sqrt{5}$, 0.000000001669241254]

[567451585. $\sqrt{5}$ - 1268860318, 0.0000000003940544068]

[5374978561 - 2403763488. $\sqrt{5}$, 0.00000000009302362684]

#7: GGT(1, $\sqrt{5}$) = 567451585. $\sqrt{5}$ - 1268860318

#8: GGT2(1, $\sqrt{5}$) = [5374978561 - 2403763488. $\sqrt{5}$, 567451585. $\sqrt{5}$ - 1268860318]

Heinrich Ludwig

27.3.2006

Ich habe wieder einen Derive-Bug zu melden. Er steht möglicherweise in Zusammenhang mit denen, über die ich jüngst berichtet habe, aber es muss nicht so sein. Leider bin ich jetzt zu spät dran, dass im DNL#61 noch darüber berichten werden könnte. Ich bin erst gestern von meinem Kollegen Wolfgang Jäger - einem langjährigen treuen Anhänger von Derive übrigens, auf ihn geht auch die Verwendung von Derive an unserer Schule zurück - darauf hingewiesen worden.

$\text{COS}(\text{ACOS}(x))$ wird zu x vereinfacht, wenn man den Definitionsbereich nicht einschränkt. Das ist in komplexer Rechnung richtig und Derive macht gute Arbeit. Wenn man aber mit Derive 5 den Term $\text{COS}(\text{ACOS}(x))$ in einem 2D-Fenster sich graphisch darstellen lässt, dann wird der Term nur für x aus $[-1; +\infty[$ gezeichnet. Derive 6 zeichnet dagegen richtig.

Nun wird es kein Patch mehr für Derive 5 geben, nachdem man für Derive 6 schon so lange auf Patches wartet. Aber meine Fehlermeldung mag die Programmierer von Derive darauf hinweisen, noch einmal genau nachzuschauen, ob in Derive 6 der Fehler wirklich behoben ist, oder ob nur ein Sonderfall vorliegt, der diesmal das Fehlverhalten kaschiert. Wenn Sie sich möglicherweise erinnern, so haben die Bugs, die ich gemeldet habe, alle damit zu tun, dass $a + 0*i$ von Derive nicht zuverlässig als reelle Zahl erkannt und weiterverarbeitet wird. Genau darum scheint es sich auch hier zu handeln: Schreibt man $\text{RE}(\text{COS}(\text{ACOS}(x)))$, so tritt der Fehler nicht mehr auf und Derive 5 zeichnet korrekt.

Heinrich Ludwig points out that there is another problem which might be connected with the bugs mentioned earlier:

*$\text{COS}(\text{ACOS}(x))$ is simplified to x if the range is not restricted. This is ok for calculations with complex numbers and DERIVE does a good job. Plotting the graph with DERIVE 5 the graph only appears for $[-1, +\infty[$, the DERIVE 6 plot is correct. The developers of DERIVE should investigate (for a hopefully coming Update) if the bug has really been fixed or if there is only a special case which hides the bug. I believe that this behaviour again is based on the fact that DERIVE does not recognize in many cases that $a + 0*i$ is a real number. Writing $\text{RE}(\text{COS}(\text{ACOS}(x)))$ solves the problem in DERIVE 5 and leads to a correct plot, too.*

Manuel Garcia Mateos

22.3.2006

Hello Derivians,

I tried to solve the following complex equation of third degree $2z^3 - (3+15i)z^2 + (-45+9i)z + 54i = 0$

One solution should be $z=3i$ (the others are $3+3i$ and $3/2+3/2i$).

I don't get these solutions.

After a substitution of z by $3i$ ($\text{subst}(\text{eq}, z, 3i)$) Derive says "false".

Where is the error?

Greetings,

Manuel

Aleksey Tetyorko

Hi!

I do not understand your troubles. My Derive 4.11 solves:

$$2z^3 - (3+15i)z^2 + (-45+9i)z + 54i = 0$$

```
;Solve(#1)
```

```
z=3*i
```

```
;Solve(#1)
```

```
z=3+3*i
```

```
;Solve(#1)
```

```
z=-3/2+3*i/2
```

Aleksey

Johann Wiesenbauer

23.3.2006

Hi Manuel,

It is true, I had also troubles when trying to solve your equation algebraically in exact mode. To be more precise, I got a screen full of nested square roots, but Derive doesn't see how to simplify them. It can see though that $3i$ is a solution if you check it like this

$$w := 2 \cdot z^3 - (3 + 15 \cdot i) \cdot z^2 + (-45 + 9 \cdot i) \cdot z + 54 \cdot i$$

$$\text{SUBST}(w, z, 3 \cdot i) = 0$$

With "a little help" it can also compute the exact solutions using the following small program

$$\text{mysolve}(w) := [1, i] \cdot \text{SOLUTIONS}([\text{RE}(\text{SUBST}(w, z, u + v \cdot i)), \text{IM}(\text{SUBST}(w, z, u + v \cdot i))], [u, v], \text{Real})'$$

$$\text{mysolve}(w) = \left[3 \cdot i, 3 + 3 \cdot i, -\frac{3}{2} + \frac{3 \cdot i}{2} \right]$$

Hope this helps.

Cheers, Johann

Stefan Welke

23.3.2006

Hello Manuel and Johann,

in contradiction to Johann's observation my DERIVE 6 solves this equation:

Just try

$$\text{SOLVE}(2 \cdot z^3 - (3 + 15 \cdot i) \cdot z^2 + (-45 + 9 \cdot i) \cdot z + 54 \cdot i, z)$$

The result is very clumsy and contains many roots etc. Then apply

Simplify->Basic from the menu bar and you will immediately get:

$$z = -\frac{3}{2} + \frac{3 \cdot i}{2} \vee z = 3 + 3 \cdot i \vee z = 3 \cdot i$$

Greetings,

Stefan Welke

Johann Wiesenbauer

23.3.2006

Hi Stefan,

While it is true that Derive 6.1 can surprisingly simplify its first result when manually forced to do so, I'm not sure if this a "contradiction" to what I said. At any rate, this "2-step simplification" is a little bit unsatisfactory. In a way Derive behaves like a pupil, who is unable to produce the expected result in the first try, but can do so after being given the hint that a further simplification is possible.

Cheers, Johann

Wim de Jong

24.3.2006

Hi all,

It is indeed surprising that the complicated algebraic solution is not directly simplified. However if you Simplify (Basic) it you get the required answer. More directly, Approximation of the expression Solve(2z^3-(3+15i)z^2+(-45+9i)z+54i=0,z) does the trick.

As to the substitution of $3i$ for z , Simplify>Subexpression Substitution yields $0=0$.

Cheers,

Wim de Jong

Two strange limits

Jim FitzSimons

This limit is wrong: $\lim_{x \rightarrow \infty} \frac{1 + \cos(x)}{1 - \cos(x)} = 1$

David Sjöstrand

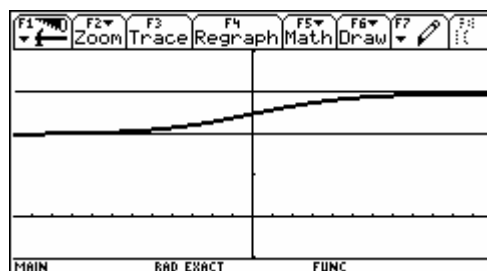
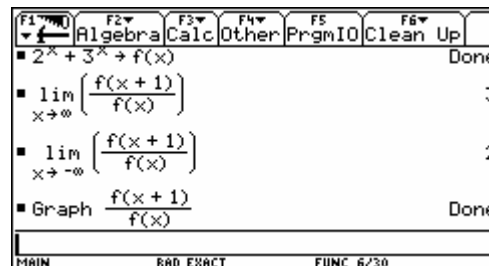
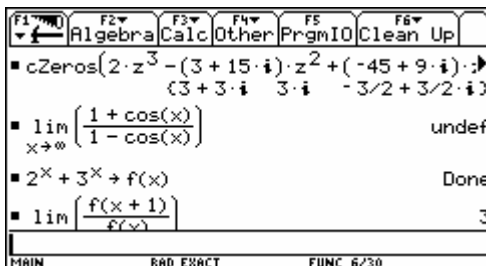
Derive doesn't seem to be able to compute the limit $\text{LIM}(f(x + 1)/f(x), x, \text{inf})$, with $f(x) := 2^x + 3^x$

$$f(x) := 2^x + 3^x$$

$$\lim_{x \rightarrow \infty} \frac{f(x+1)}{f(x)} = \lim_{x \rightarrow 0^+} \frac{3^{(x+1)/x} \cdot \text{LN}(3)^{10} + 2^{(x+1)/x} \cdot \text{LN}(2)^{10}}{3^{1/x} \cdot \text{LN}(3)^{10} + 2^{1/x} \cdot \text{LN}(2)^{10}}$$

$$\lim_{x \rightarrow \infty} \frac{f(x+1)}{f(x)} = \lim_{x \rightarrow 0^+} \frac{2 \cdot 3^{1/x} \cdot \text{LN}(3)^{10} + 3 \cdot 2^{1/x} \cdot \text{LN}(2)^{10}}{3^{1/x} \cdot \text{LN}(3)^{10} + 2^{1/x} \cdot \text{LN}(2)^{10}}$$

Let's look how the TI deals with the problems from above:



Jan Vermeylen

11.3.2006

Hello Josef,

I have a question of one of my science colleagues.

He has a set of some thousand quadruples (t, x, y, z) which show the position (x, y, z) of a particle at a certain time t .

Is it possible in Derive to produce a 3D graph of the path of this particle with an animation of the moving particle? Or should I use DPGraph?

Thank you for giving me advice.

Best regards,

Jan

DNL: Good question, Jan. I tried to introduce a slider bar for the subscript of a point matrix to address one point after the other for being plotted but I failed. Obviously the slider bar cannot be applied on discrete variables?

It would be great if any of our graphic sorceres had any advice. Josef

Last News:

Peter Schofield sent a mail containing very valuable advice – see next DNL

An InterConnectivity-Communication problem between DERIVE and the TI

Rainer Gerdes from Germany asked for advice because he failed trying to transfer a matrix equation from the TI to DERIVE. This might be of interest for some of you. See a summary of his report, and how to overcome this deficiency:

I enter the equation $[1, 2; 4, 5] * [x; y] = [1, 2]$ on the Voyage200, save it as a textfile, connect the device with my PC, launch DERIVE and import the file. After some seconds the error message "Syntax Error" appears. What am I doing wrong?

Regards

Rainer Gerdes

DNL: Dear Mr. Gerdes,

You are doing right, InterConnectivita does wrong – that's the bad news, but your matrix calculation can be transferred – but with some tricks, that's the good news. There is something to be improved by the software smithies.

One way is to edit the matrices in the text editor of the TI as lists. Your equation should then read as follows (first attempt):

$\{\{1, 2\}, \{4, 5\}\} * \{\{x\}, \{y\}\} = \{1, 2\}$ (which delivers the result false because of the non suiting dimensions, the next line will do better:)

$\{\{1, 2\}, \{4, 5\}\} * \{\{x\}, \{y\}\} = \{\{1\}, \{2\}\}$ will result in simultaneous equations in DERIVE-environment.

Interestingly enough one can leave the matrices (and the product but not the whole equation) with brackets and it will be transferred correctly. So you can also transfer without any problem the following forms of the equation:

$[1, 2; 4, 5] * [x; y] = [1, 2]$ and $[1, 2; 4, 5] * [x; y] = \{\{1\}, \{2\}\}$.

And there is another very "perfidious" trick:

In the text editor remove the leading "C"s (which declare the lines as executable commands \Rightarrow leading to commands in DERIVE) converting the lines to simple comments. This will be very welcome by DERIVE appearing as a comment in a textbox. In DERIVE copy and paste this "fictitious comment" into the edit line, ENTER, and it is done.

Best regards and much luck

Josef

Susan Jackman from England asked how to produce the DERIVE-Snail (Cover of the manual). As you might remember we had this snail in the last DNL. I sent the file to Susan, but she had problems to bring the snail alive with DERIVE 5 (no slider bars). Finally it lucked but in the meanwhile one of her students had made his/her own tries. Susan wrote:

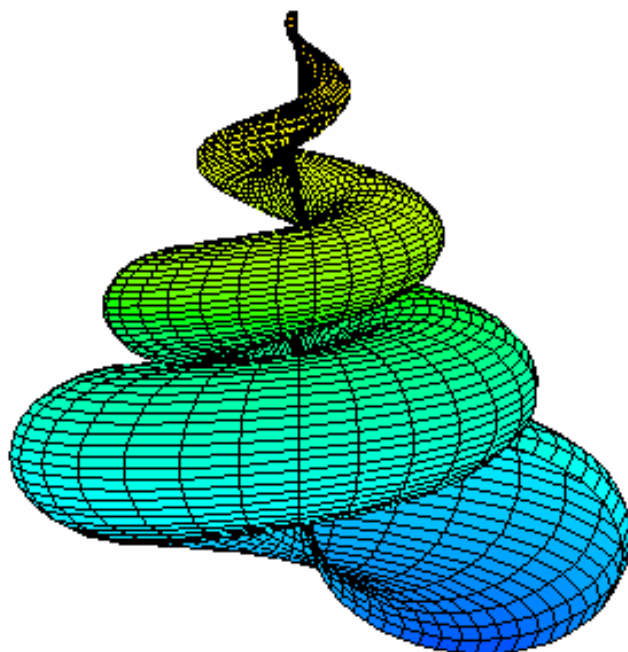
Josef,

Thank you. I now have opened your snail and I'm very impressed. I am attaching my seashell which is based on a much simpler function. The idea came from a student who found something like it (done on Mathematica) on www.wolfram.com However I spent some time changing values of the parameters and changing the scale.

Best wishes,

Susan

$$\left[\left(\left(1 - \frac{v}{2 \cdot \pi} \right) \cdot ((1 + \cos(u)) + 0.1) \cdot 10 \cdot \cos(2 \cdot v), \left(1 - \frac{v}{2 \cdot \pi} \right) \cdot ((1 + \cos(u)) + 0.1) \cdot 10 \cdot \sin(2 \cdot v), \frac{v}{2 \cdot \pi} + 0.2 \cdot \sin(u) \cdot \left(1 - \frac{v}{2 \cdot \pi} \right) \right] \right]$$



If anybody of you has other animals for our DERIVE – zoo, please send your pictures, Josef

On the Titbits:

We have no Titbits in this issue, because the journal is full and Johann has a lot to do on the University. He promised the next Titbits for DNL#62. Come and meet Johann in Dresden. He will enjoy any talk about DERIVE and his column. Josef

Submitted and Accepted Lectures and Workshops for the 7th International DERIVE and TI-CAS-Conference in the Frame of DES –TIME 2006 Dresden
(in order of submitting the papers)

The Lectures:

Karsten Schmidt, D,	Working with Random Variables and Random Numbers in DERIVE
Alex Lobregt, NL,	Signal Processing using a CAS
Dennis Pence, USA	Activities with Implicit Functions and Implicit Differentiation the TI-89/V200
Michel Beaudin, CAN	Theory versus Technology: don't look for a competition, look for collaboration
Jose Luis Galan, SPA	LINE_INTEGRALS.MTH: Solving Problems of Line Integrals using Derive
Jose Luis Galan, SPA	MULTIPLE_INTEGRALS.MTH: Solving Problems of Multiple Integrals
Josef Böhm, A	Background Pictures as a Stimulating Means for Math Teaching
Giora Mann, ISR	Changing the Viewing Angle on a Conic Section
Lisa Townsley, USA	DERIVE, a Mathematical Assistant
Gabriel Aguilera, SPA	GRAPH_ALGORITHMS.MTH: Graph algorithms using display step in D6
Ludwig Paditz, D	The Rank of a Matrix with Parameters and the Solution of a Linear System
Gilles Picard, CAN	Using the Voyage 200 in the Classroom: Surprising Results
David Jeffrey, CAN	Flying with DERIVE
Steven Schonefeld, USA	Amorous Bugs and Pursuit Problems
Jean-Jacques Dahan,	FR Vizualizing Solutions of Des and Systems of 2nd order with Cabri2+
Max Schröfel, D	Cubic Spline Functions, Matrices and DERIVE
Günter Alfan, A	Der Einsatz von CBL/CBR im fächerübergreifenden, naturw. Unterricht
Eugenio Roanes, SPA	Eight Wishes about Computer Algebra Systems
Renee Gossez, BEL	Using DERIVE 6 to find equation and to visualize a locus in 3D space
Gilles Picard, CAN	Numerical Methods with the V200: To Function or to Program, that's the quest
Stephan Joubert, RSA	The Mathematics and Industrial Applications associated with the singing wine glass using a CAS
Josef Lechner, A	See it complex - make it easy, Experimental working using complex numbers
David Sjöstrand, SWE	Interactive Investigations with Slider Bars in DERIVE 6
Roman Hasek, CZ	Geometry with DERIVE
Walter Wegcheider, A	Solving Problems of Spherical Trigonometry with the Help of Computer Algebra

Johann Wiesenbauer, A	Elliptic Curve Cryptography with DERIVE
Pavel Pech, CZ	Discovering and Proving Geometric Inequalities by CAS
Tania Koller, A	Motivierender Mathematikunterricht mit CAS - ein Erfahrungsbericht
Temple Fay, RSA	A Quadratic Spring Model
Wolfgang Schlichthorn, D	Das Rechnen mit Größen in Anwendungsaufgaben
Quay van der Hoff, RSA	Real-Life Examples of Limit Cycles
Anna Salvadori, I	Mathematics & Reality
Leon Magiera, POL	Interactive Work with CAS when Solving Problems on Physics
Peter van der Velden, NL	Didactical Principles of Integrated Learning Math with CAS

The Workshops:

René Hugelshofer, CH	Dynamic Algebra
Jose Luis Galan, SPA	DERIVE as a Pedagogical CAS: Programming using the DISPLAY-function
Josef Böhm, A	Slider Bars narrow the Gap between Computeralgebra and Dynamic Geometry
Lisa Townsley, USA	Calculus Lab Transformation from DERIVE to DERIVE and Biology
Jean-Jacques Dahan, FR	Vizualizing Solutions of DEs and Systems of 1st and 2nd order with Cabri2+
Guido Herweyers, BEL	Applied Linear Algebra
Matija Lokar, SLO	Selected Themes from Teaching Calculus with CAS
Günter Alfanz, A E	Experimente für den Einsatz von CBL/CBR im fü, naturw. Unterricht
Carl Leinbach, USA	Using DERIVE to Teach Bioinformatics Algorithms
Eoghan MacAogáin, IRL	Student Projects in using DERIVE to help teach topics in 2nd level maths
Johann Wiesenbauer, A	Elliptic Curve Cryptography with DERIVE
Wade Ellis, USA	Integrating Graphics, Tabular, and Symbolic Calculation

There will be another approximately exciting 60 lectures and workshop presented at the
9th ACDCA Summer Academy.

You are kindly invited to visit the Conference website for browsing the scientific program, the venue and the social program

www.des-time-2006.de