THE DERIVE - NEWSLETTER #114

ISSN 1990-7079

THE BULLETIN OF THE DERIVE USER GROUP

+ CAS-TI

Contents:

June 2019

Steve Arnold is very busy with Lua. He sent two interesting links:

Hi Josef

Something you might find interesting...
Nice to run Lua scripts in your web page.
(Lua and JavaScript)

https://compasstech.com.au/luajs/index.html

See also

https://compasstech.com.au/cfrac/index.html

for an application. ("Fractured Fractions")

With best wishes,
Steve

Scientific American 6/19 (German Issue) presents an article about "*Tropical Geometry*".
If you are interested in this new field of mathematics, here are two links to obtain free papers on this topic:

https://www.math.uni-tuebingen.de/user/jora/downloads/FirstExpedition.pdf

https://www.math.uni-tuebingen.de/user/jora/downloads/main.pdf

---

Quantum was a great Magazine of Math and Science and was published from 1990 to 2001.
You can download all magazines as pdf-files from the NSTA-Publications-Archive:

https://www.nsta.org/publications/quantum.aspx#about

Exhibit: Milestones of Science

This is a nice collection of pictures of famous scientists and what is even more interesting pictures of title pages of their most famous books.

Just click on the various pictures on top of the website.

https://www.buffalolib.org/content/milestones-science/featured-authors-scientists

Dear DUG Members,

I am happy to be in time with DNL#114. It is very hot outside (33°) and I finish this issue writing my letter.

Regularly I browse in my collection of papers and e-mails. At this occasion I came across a thick stack of e-mails from 2001. Hank Schenker raised a probability theorem question and this problem incited a broad discussion within the then JISCmail-discussion group of Derivers.

In my opinion it is worth to be published not only because of the request and the respective answers but also because we now can visualize the results with DERIVE6 and TI-Nspire … and there is still one question remaining (page 16).

It gave me also the occasion to present the most important feature – at least from my point of view – of the latest Nspire version: programming graphics is back on TI!

While Bill Yancey is a master in traditional skills (e.g. integrating) am I referring on CAS and plots supported by sliders. It is nice to compare both approaches for one sample problem. See more on page 17.

I am glad that we have three contributions in this DNL which might be suitable for classroom use: *Factoring Trinomials*, which is a report without CAS use, *Throwing an Object …*, again traditional math vs – or better cooperating with – CAS, and the 2nd part of *Anageo*. The most interesting part is treating the case of three planes in space. I believe that this might be a challenging project for students.

Finally, I'd like to draw your attention to the links on the opposite page. They offer a rich resource on interesting materials (Quantum, Milestones, …)

Best regards and wishes

Josef

A good friend of mine was in Saint Petersburg this week. We were there some years ago, and unfortunately, I missed the occasion to visit a special tomb on the famous cemetery of the Alexander Nevsky Monastery.

Today I received a picture from my friend's mobile phone: *I am sure you know the engraved name!*

The *DERIVE-NEWSLETTER* is the Bulletin of the *DERIVE* & CAS-*TI User Group*. It is published at least four times a year with a content of 40 pages minimum. The goals of the *DNL* are to enable the exchange of experiences made with *DERIVE*, *TI*-CAS and other CAS as well to create a group to discuss the possibilities of new methodical and didactical manners in teaching mathematics.

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

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

Next issue: September 2019

Preview: Contributions waiting to be published

Some simulations of Random Experiments, J. Böhm, AUT, Lorenz Kopp, GER
Wonderful World of Pedal Curves, J. Böhm, AUT
Simulating a Graphing Calculator in *DERIVE*, J. Böhm, AUT
Cubics, Quartics – Interesting features, T. Koller & J. Böhm, AUT
Logos of Companies as an Inspiration for Math Teaching
Exciting Surfaces in the FAZ
BooleanPlots.mth, P. Schofield, UK
Old traditional examples for a CAS – What´s new? J. Böhm, AUT
Mandelbrot and Newton with *DERIVE*, Roman Hašek, CZ
Tutorials for the NSpireCAS, G. Herweyers, BEL
Dirac Algebra, Clifford Algebra, Vector-Matrix-Extension, D. R. Lunsford, USA
A New Approach to Taylor Series, D. Oertel, GER
Statistics of Shuffling Cards, Charge in a Magnetic Field, H. Ludwig, GER
Selected Lectures from TIME 2016
More Applications of TI-Innovator™ Hub and TI-Innovator™ Rover
Surfaces and their Duals, Cayley Symmetroid, J. Böhm, AUT
Affine Mappings –Treated Systematically, H. Nieder, GER
The Penney-Ante Game, MacDonald Phillips, USA
Hyper Operations for DERIVE, Julius Angres and others, Germany
Investigations of Lottery Game Outcomes, Wolfgang Pröpper, GER

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

E-mails from 2001

in order of coming in (DERVE-NEWS@JISCMAIL.AC.UK).

Hank Schenker (8 July 2001)

My DfD 4.11 locks on my computer (a slow Pentium with small memory) when attempting to approximate

$$\sum_{i=2}^{1000000} \frac{\mathrm{PERM}(999999,\ i-2)\cdot(i-1)}{1000000^{i-1}}$$

This expression comes from a probability experiment. Place 1,000.000 different objects in an urn. Select an object at random, note its value and return it to the urn. Repeat until the most recently drawn object matches one of those on the list of previously selected objects. Define a random variable as the number of draws necessary to obtain a match.

The expression above, I believe, gives the expected value of this random variable. If anyone cares to try approximate this using *Derive* on a powerful PC, I would be greatly appreciative.

Hank Schenker

S John (9 July 2001)

I tried to approximate the expression on my Pentium III, with no success. When I clicked the approximate button, a dialog box popped up showing how long the calculation was taking. I let it go for a very long time, and when it got to 3,000 seconds, I aborted it. I am not sure what the problem is, but I am curious to find it out. I think tomorrow I will try entering the expression on my TI-92 Plus and just see what happens.

Best regards
John

Wim de Jong (9 July 2001)

You may have to scale down the size of the problem. I defined `f_(n)` and asked *Derive* to approximate `f_(n)` for n = 100, 200, 300 and 1000. On each occasion it produced fairly promptly the answer 1.

$$f\_(n) := \sum_{k=2}^{n} \frac{\mathrm{PERM}(n-1,\ k-2)\cdot(k-1)}{n^{k-1}}$$

```
APPROX(f_(100)) = 1

APPROX(f_(200)) = 1

APPROX(f_(300)) = 1

APPROX(f_(1000)) = 1
```

It also had no difficulty with VECTOR(f_(n),n,2,100), giving an array which rapidly increases to 1. So, my guess is that the answer to your problem is 1 with a very high degree of accuracy.

Wim de Jong

Hank Schenker (9 July 2001)

I believe that there is some mistake in your expression, since the expected value couldn't be 1. Thanks for looking at this.

Hank

Johann Wiesenbauer (9 July 2001)

May I take the liberty to "scale down" the size of the problem another time? With a little high school mathematics your function f_(n) can be written in the simpler form

$$f\_(n) := 1 - \frac{n!}{n^n}$$

After all Hank's formula (unlike what he thought!) represents the probability that after *n* draws from the urn with *n* balls at least two of them are equal! Hence your guess is correct!

In fact, using some terms of Stirling's formula you could also consider the approximation

$$ff(n) := 1 - e^{-n} \cdot \sqrt{(2 \cdot \pi \cdot n)} \cdot \left(1 + \frac{1}{12 \cdot n} + \frac{1}{288 \cdot n^2}\right)$$

Which can be used in computing $f(10^6)$, i.e.

$$ff(10^6) = 1 - \frac{288000024000001 \cdot \sqrt{2} \cdot \sqrt{\pi} \cdot e^{-1000000}}{288000000000}$$

$$ff(10^6) = 1$$

Isn't it true that what sometimes appears to be a matter of programming (or computer power?) is simply a matter of sound mathematics (or should I say horse sense?) … At any rate, the case at issue is a nice example of this sort.

Cheers;
Johann

Don Taylor (9 July 2001)

I would certainly agree. But there is one thing that could be done.

I've suggested something to the authors, a couple of times, that might help users realize what is happening in cases like this. If *Derive* were to display the value of the index as it steps through the summation, and do likewise for other forms of iteration, then the user could have a tangible indication of how fast progress was being made. For nested levels of iteration there might possibly be a string of such values displayed.

Thus, in the original expression, when the little window popped up, showing that memory use seems to sit at 0% utilization for almost all problems I have tried, it could also include *i*=1, *i*=2, *i*=3, … And in this problem, he would have rapidly seen that it was going to be a very very long time before it got to *i*=1000000.

As more and more users, of widely differing backgrounds, tackle bigger and bigger problems with *Derive* it seems that providing some better indication of how things are proceeding might be a very useful feature. And it doesn't seem that this would be too difficult to implement.

Don

(Comment: In *DERIVE 6* we can use the "WRITE"-command to indicate the steps, Josef)

Ralph Freese (9 July 2001)

```
> f(n):=SUM(PERM(n-1, k-2)*(k-1)/n^(k-1), k, 2, n))
```

There are two changes needed for this. First the upper limit of the sum should be $n + 1$ (since it is possible that the number of draws until the first repeat is $n + 1$). After this change, $f(n) = 1$ for all n. $f(n)$ is the sum of all probabilities that the random variable X = number of draws is k. Since you are summing over all possible values of k, the sum is 1.

To get the expected value of X you need to sum the value of X times the probability it has that value. So, the expected value is

```
E(X) = SUM(k*PERM(n - 1, k - 2)*(k - 1)/n^(k - 1), k, 2, n+1)
```

(Unfortunately, I am at a machine without *Derive* so I cannot test this; maybe someone else can try for me.)

Efficiency: it is hardly surprising *Derive* has trouble summing a 1000000 terms, some of which have denominators on the order of 1000000^1000000. To make this easier note the probability $X = k$ is `PERM(n - 1, k - 2)*(k - 1)/n^(k - 1)` which written out is

$$\frac{n-1}{n}\,\frac{n-2}{n}\,\frac{n-3}{n}\cdots\frac{n-k+2}{n}\cdot\frac{k-1}{n}.$$

So, to write a program to evaluate the expected value you might have local variables (term holds the product above without the last factor)

$sum := 0$
$term := 1$

then you would iterate k from 2 to n + 1, doing

$term := term * (n - k + 2)/n$
$sum := sum + k * (k - 1)/n *$ term

when loop finished, the value of the sum should be the answer.

Aloha Ralph

Joe H Frisbee (10 July 2019)

I would like to offer my interpretation of a solution for the probability problem, assuming I have interpreted its statement correctly. First a comment about the results presented so far. It seems that a value near 1 is not reasonable. Perhaps the previously presented expression is just the sum over the index, of the probability of any particular value of the index being the count at which a repeated value is first obtained. Since the summation is 1 to N the final value should be equal to 1. In any case, here is my presentation of the pdf (probability density function) $f(x)$ associated with the first repeated value of a random draw, with replacement of course, from n unique values:

$$f(n,\ x) := \frac{(x-1)\cdot n!}{(n+1-x)!\cdot n^{x}}$$

$$\text{VECTOR}(f(4,\ k),\ k,\ 2,\ 5) = \left[\frac{1}{4},\ \frac{3}{8},\ \frac{9}{32},\ \frac{3}{32}\right]$$

$$\sum_{k=2}^{5} f(4,\ k) = 1$$

$$\sum_{k=2}^{101} f(100,\ k) = 1$$

$$\sum_{k=2}^{1001} f(1000,\ k) = 1$$

Using this function, the expected value of x with n values in an urn is given by:

$$e(n) := \sum_{x=1}^{n} x\cdot f(n,\ x)$$

$$e(50) = 9.543127039$$

Examples, some confirmed by experiment (up to 10000 trials) are

N	E(x)	Trial E(x)	DfW5 Time (sec)	DFW 6.1 Time
50	9.543	9.567	~ 0.0	~ 0.0
100	13.210	13.188	0.1	~ 0.0
1000	40.303	40.426	9.1	1.81
2000	56.719	-------	100.8	16.2

I estimate that trying to get an answer for n = 1000000 would take way too long to try. So, I offer no answer for that particular value of n.

Wim de Jong (10 July 2011)

Thank you, Johann, for putting (as usual) the icing on the cake. *Derive* makes one lazy. Without doing any high school maths myself I asked *Derive* to check your simplification for $n = 2$ to 100. It duly obliged by simplifying (not approximating) `vector(f(n)-1+n!/n^n,n,2,100)` to the 99-dimensional zero vector in about 5 seconds (0.16 sec with *Derive* 6.1, Josef). It would not simplify the general expression `f(n)-1+n!/n^n` to 0, which must have to do with its digestive system.

Cheers,
Wim

Ralph Freese (10 July 2001)

Here is a *Derive* program (I wrote it in *Derive 5* but it should also work in *Derive 4*) for my method of evaluating the expected value of number of draws (with replacement) until the first duplicate is found:

$$f(v) := \left[v_1 + \frac{v_2 \cdot v_3 \cdot (v_3 - 1)}{v_4}, \frac{v_2 \cdot (v_4 - v_3 + 1)}{v_4}, v_3 + 1, v_4\right]$$

$$g(n) := (\text{ITERATE}(f(v), v, [0, 1, 2, n], n))_1$$

$$g(1000) = 40.30321292$$

$$g(10000) = 125.9991218$$

$$g(100000) = 396.9997263$$

In the vector produced by `f(v)` the components are (in the notation of my last message):

1. sum
2. term (with the index shifted 1 from my last message)
3. *k*
4. *N*

It took about 8 seconds to approximate g(10000). (0.73 sec in *Derive* 6.10)
It took about 7 minutes to approximate g(100000). (37.5 sec in *Derive* 6.10)

I wonder what the asymptotic behavior of this sequence is? $c * \sqrt{n}$?

Johann Wiesenbauer (10 July 2001)

Yes, this is the proper algorithm for computing the sums Hank was actually aiming at, though I would slightly prefer the implementation

$$h(n) := \left(\text{ITERATE}\left(\left[s + t \cdot k \cdot (k - 1), \frac{t \cdot (n - k + 1)}{n}, k + 1\right], [s, t, k], \left[0, \frac{1}{n}, 2\right], n\right)\right)_1$$

$$h(10000) = 40.30321292$$

By the way, an amazingly good approximation is the function

$$hh(n) := \sqrt{\left(\frac{\pi \cdot n}{2}\right)}$$

hh(10000) = 125.3314137

So again, Ralph's guess was right.

Joe H Frisbee (10 July 2019)

In thinking over the several email notes again, I believe that the original form needs two corrections. First, the *N* needs to be replaced by *N*–1 within the summand. This will insure that the originally presented form (which is the CDF and not the expectation of *i*) sums to 1. As several later postings indicated, the original sum was almost equal to 1. This correction for *N* will correct the minor (for large *N*) discrepancy. Second, since the original form was actually the cumulative distribution function (CDF), the summand needs to be multiplied by *i*. This is assuming that the desired quantity is the expected value of *i* for which a repeat occurs.

With these two changes the result should be as desired. Note that my previous posting on this subject did not have the *N*–1 substitution included

John Frisbee

Hank Schenker (11 July 2001)

Many thanks to Ralph and Johann and the others who helped me with the expected value problem.
I realize that using a program or *Derive* iteration can be more efficient than the *Derive* SUM-function in certain situations.

Hank

Johann Wiesenbauer (12 July 2001)

There have been a lot of interesting contributions to Hank's probability problem. In particular, it has been pointed out by several people that he actually wanted to know the value of the function

```
f(n):=SUM(PERM(n–1,k–2)*(k–1)*k/n^(k–1),k,2,n+1)
```

at $n = 10^6$, which represents the expected number of draws (with replacement) from an urn with *n* balls assuming that you stop drawing if a drawn ball matches a previously selected one.

Interestingly enough, this question has not been answered yet, at least not for $n = 10^6$. In particular, all programs presented so far (including mine) seem to be too weak to settle this question in a reasonable time.

This will change dramatically with the following new versions (again essentially based on Ralph's ideas, but along with some important enhancements; note the second version is for *Derive 5* users only!) – and for *Derive 6* user, too, of course.

```
hank(n, k_ := 2, s_ := 0, t_ := 1, u_ := 1) :=
  Loop
    If s_ = u_
       RETURN s_/n
    u_ := s_
    s_ :+ t_·k_·(k_ - 1)
    t_ :* (n - k_ + 1)/n
    k_ :+ 1
```

For example, the values of `hank(n)` for n=10000, 100000, 1000000 are 125.9991218, 396.9997263, 1253.980907, respectively, and the computation times on my Pentium 450 MHz PC have been 0.27, 0.9, 2.48 s (yes, seconds!).

The corresponding values of `hh(n)` are 125.3314, 396.3328, 1253.3141, i.e. they are pretty good approximations.

```
[hank(10000), hank(100000), hank(1000000)]
```

Approximate!

```
[125.9991218, 396.9997263, 1253.980907]
```

calculation time 0.312 sec (DERIVE 6.10 in 2019)

$$hh(n) := \sqrt{\left(\frac{\pi \cdot n}{2}\right)}$$

```
[hh(10000), hh(100000), hh(1000000)]

[125.3314137, 396.3327297, 1253.314137]
```

By the way, why the hell are the “exact”values almost integers in our examples? Frankly, I’d really like to know this …

Cheers, Johann

PS: While writing these lines I received an email of Michael Lasarev who wrote a paper on this urn problem. I still have to read it more thoroughly, but glancing over it, I saw he was referring to Ralph’s email (“It took about 8 seconds to approximate g(10000) to 125.9991218 and it took about 7 minutes to approximate g(100000) to 396.9997263 …”) claiming that MuPad does a much better job by computing g(100000) and f(100000) in less than 6 seconds.

Boy, I think he will have to rewrite his paper when reading this email!!! So, you want the value of Hank’s function for $n = 10^7$? No problem at all, just try it out … And, yes, MuPad failed to compute $f(n)$ for $n = 10^6$ … Oh well, we all know, it hasn’t been that easy for us Derivers either …

Ralph Freese (14 July 2001)

In case not all the readers saw, the idea is this. In each iteration of the loop *s_* is changed by adding *t_*k_*(k_* – 1) to it. But after some number of steps *t_*k_*(k_* – 1) gets so small that adding it to *s_* does not change *s_* in the current (approximate) precision, so there is no point in continuing. You can see how far into the loop it went by changing the `RETURN` statement to be `RETURN [s/n, k]`. When you do this with n = 1000000 you see that less than 7000 steps of the loop are executed. Also note in any mode (including exact) it will return the same answer as mine but much faster. [as a general technique the one thing people should be careful about it that the things being added to the partial sum (*s_*) should monotonically decrease to 0 (as is the case here) or it may give the wrong answer.]

Johann Wiesenbauer (16 July 2001)

I have been asked by some people how I arrived at the approximation SQRT($n*\pi/2$) for Hank's problem, which is asymptotically correct (i.e. the relative error goes to 0 as n goes to infinity). As this may be of general interest, I'll give the answer as a *Derive*-attachment to this email.

Just to make sure, let me state clearly that this is not a result of mine, but rather a classical result, which you will find in many textbooks on algorithms. Though it is my way of reasoning, it may be neither new or correct. (hence, check it carefully!)

As for me, this is definitely the end of this never-ending story. At its beginning there was what one might call "Hank's horrible howler" (sorry Hank, but this alliteration was simply too tempting …), which obviously aroused the imagination of a lot of people here. Some people (including me at the beginning) were interested in the erroneous formula given by him and its simplification, others dealt from the start with the underlying question … Anyway, many thanks to Hank for posing the nice problem!

Cheers,
Johann

As we know now, the upper bound of the sum is too small by one and what is more he "lost" the factor i somehow. The corrected sum can also be written in the form

#1: $$\sum_{i=2}^{n+1} \frac{\left(\prod_{j=1}^{i-2}\left(1-\frac{j}{n}\right)\right)\cdot(i-1)}{n}\cdot i$$

First we notice that this sum is equal to

#2: $$\sum_{i=0}^{\infty} \frac{\left(\prod_{j=1}^{i-2}\left(1-\frac{j}{n}\right)\right)\cdot(i-1)}{n}\cdot i$$

And now I make a series of replacements which are only asymptotically correct though. First I replace 1-j/n by exp(-j/n) getting thereby

#3: $$\sum_{i=0}^{\infty} \frac{\left(\prod_{j=1}^{i-2}\mathrm{EXP}\left(-\frac{j}{n}\right)\right)\cdot(i-1)}{n}\cdot i$$

and after simplifying the product term (not the whole sum !!)

#4: $$\sum_{i=0}^{\infty} \frac{e^{-i^2/(2\cdot n)+3\cdot i/(2\cdot n)-1/n}\cdot(i-1)}{n}\cdot i$$

Asymptotically this sum can also be written in the form

#5: $$\sum_{i=0}^{\infty} \frac{e^{-i^2/(2\cdot n)}\cdot i^2}{n}$$

Now I change to the integral

#6: n :∈ Integer (0, ∞)

#7: $$\int_0^{\infty} \frac{\text{EXP}\left(-\frac{x^2}{2\cdot n}\right)\cdot x^2}{n}\,dx$$

and let Derive compute it getting thereby

#8: $$\frac{\sqrt{2}\cdot\sqrt{\pi}\cdot\sqrt{n}}{2}$$

which can also be written as

#9: $$\sqrt{\left(\frac{\pi\cdot n}{2}\right)}$$

Johann Wiesenbauer (18 July 2001)

You may wonder why I'm writing a second response to Ralph's mail from 9 July. In retrospect, it has become clear that this lucid analysis of Ralph has lured me (and maybe also others) into thinking that this is definitely the most appropriate way of dealing with this problem. After reading the contributions to a parallel discussion (go to the mathematics section of http://go.compuserve.com/scimath and look there for Hank's thread "Infinite sum") I learned that it's only half of the truth, as it were.

(Comment Josef: The site mentioned above does not exist in 2019. But searching the Internet I came across another site which might be of interest for math and science educators: http://scimath.net/archive.asp)

Here comes the other half for those still interested in the subject.

Using the denotation

$$p(k) = \frac{n-1}{n}\,\frac{n-2}{n}\,\frac{n-3}{n}\ldots\frac{n-k+2}{n}\,\frac{k-1}{n}$$

Along with the definition $p(1) = 0$ it is clear that the expected value of draws is simply

$$p(1) + 2p(2) + 3p(3) + 4p(4) + 5p(5) + \ldots$$

So far so good! But all people involved in this discussion including me have overlooked the following simplification of the latter sum

$$p(1) + 2p(2) + 3p(3) + 4p(49 + 5p(5) + \ldots =$$

$$[p(1) + p(2) + p(3) + p(4) + p(5) + \ldots] +$$
$$[p(2) + p(3) + p(4) + p(5) + \ldots] +$$
$$[p(3) + p(4) + p(5) + \ldots] +$$
$$\ldots\ldots\ldots$$
$$= 1 + 1 + \frac{n-1}{n} + \frac{n-1}{n}\frac{n-2}{n} + \frac{n-1}{n}\frac{n-2}{n}\frac{n-3}{n} + \ldots$$

since $p(k)+p(k+1)+p(k+2)+...=\frac{n}{n}\frac{n-1}{n}\frac{n-2}{n}\frac{n-3}{n}...\frac{n-k+2}{n}$ (note that both sides of this equation represent the probability that there is no match in the first $k-1$ draws).

I for my part was flabbergasted when I saw this simple formula for the first time and it confirms what I said in a previous email about the central role of mathematics in these kinds of problems.

An implementation could like

```
hank2(n, k_ := 1, s_ := 1, t_ := 1, u_) :=
  Loop
    u_ := s_
    s_ :+ t_
    If s_ = u_
       RETURN s_
    t_ :* (n - k_)/n
    k_ :+ 1
```

```
[hank2(10000), hank2(100000), hank2(1000000)]
```

```
[125.9991218, 396.9997263, 1253.980908]
```

calculation time 0.218 sec (DERIVE 6.10 in 2019)

(following essentially a suggestion of Marijke van Gans on the SCI/MATH forum). It's also a little bit faster (taking now only 2.05s for $n = 10^6$ on my PC), but what is more important it has become a lot simpler.

Finally, I would like to mention that this "urn problem" (also called "birthday paradox" elsewhere) id of great practical importance., just in case you don't know this. If you stopped all computers on this globe checking what they are doing right now a surprisingly high percentage would simply make entries in hash tables. After how many tries should you expect "collisions", i.e. two records leading to the same hash value assuming that your database has n records? Well, I think you know the answer by now …

Cheers
Johann

Johann wrote about the "never-ending" story in 2001. Now we have 2019 and I hope that I can terminate this interesting and inspiring story adding some ideas, Josef.

When I was a teacher and it came to teach probability theory, I liked to introduce the subject performing random experiments in reality and by computer simulations. In the following I will offer how this could be done using Hank's problem – and, by the way I will take the welcome occasion to present the latest version of TI-NspireCAS which enables performing dynamic simulations of random experiments.

I will start with a question posed by Ralph Freese:

"I wonder what the asymptotic behavior of this sequence is? c ∗ √n ?"

Thanks Johann Wiesenbauer do we know the answer in the meanwhile. Let's assume that we – or our students – don't know, then we could try a regression.

```
tbl := TABLE(hank2(x), x, 500, 10000, 500)

Pwrreg(tbl)
```

$$\begin{bmatrix} \text{PowerReg Line:} & 1.32391\cdot x^{0.494503} \\ \text{SSE:} & 0.0978351 \\ \text{DetCoeff:} & 0.999993 \end{bmatrix}$$

These are the first three rows of the generated table:

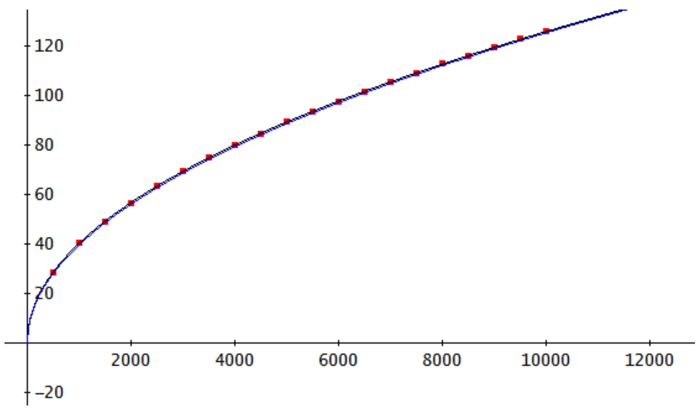
$$\begin{bmatrix} 500 & 28.6962 \\ 1000 & 40.3032 \\ 1500 & 49.2099 \end{bmatrix}$$

We can plot the scatter diagram of the points generated in tbl, then perform the regression (power function) using a tool presented in DNL#46 (from 2002) and plot the regression line (in red).

Referring to Johann's approximation formula I superimpose this function (in blue).

$$\sqrt{\left(\frac{\pi\cdot x}{2}\right)}$$

$$1.25331\cdot\sqrt{x}$$



Now, let's start the simulation – using statistics tools from DNL#45 (2002).

```
hanksim(n, u, i, urn, trs, dr, tr, dummy) :=
  Prog
    dummy := RANDOM(0)
    draws := []
    i := 1
    Loop
      If i > n exit
      trs := VECTOR(0, k, 1, u)
      tr := 1
      Loop
        dr := RANDOM(u) + 1
        If trs↓dr = 0
           Prog
             trs↓dr := 1
             tr :+ 1
           exit
      draws := APPEND(draws, [tr])
      i :+ 1
    AVERAGE(draws)
```

`hanksim(1000,100)` performs 1000 experiments with 100 objects in the urn.

```
hanksim(1000, 100)

13.487

FRETAB(draws)
```

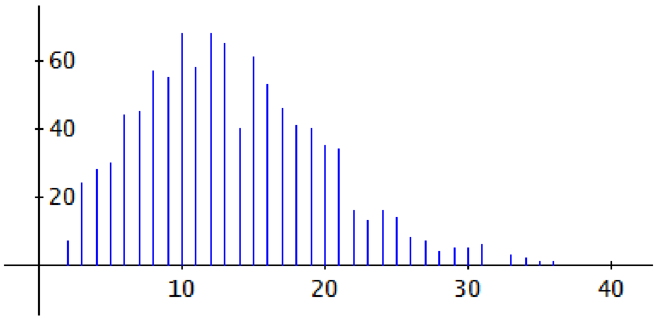
2	3	4	5	6	7	8	9	10
7	24	28	30	44	45	57	55	68

In average we need ~ 13.5 draws to come across a repetition.

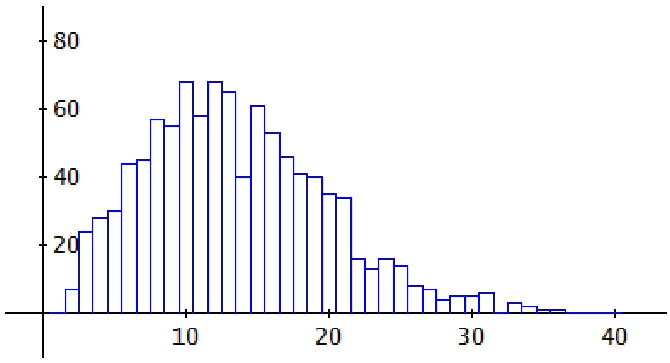
The expected value is 13.21 (see above).

We can plot the frequency-diagram in different forms:

```
#29:   FREQDIAG(draws)
```




```
#30:   HISTO(draws, 0.5, 40.5, 40)
```

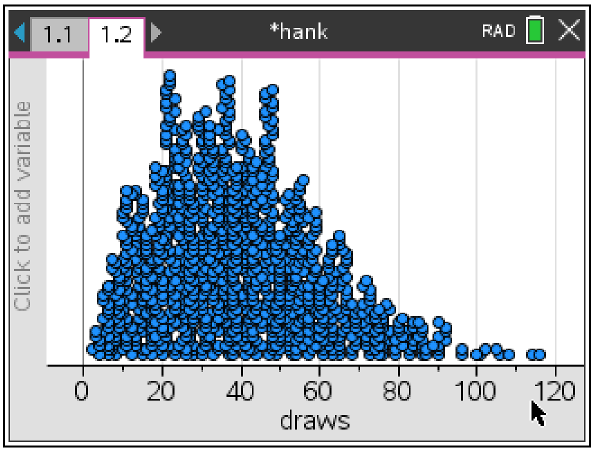


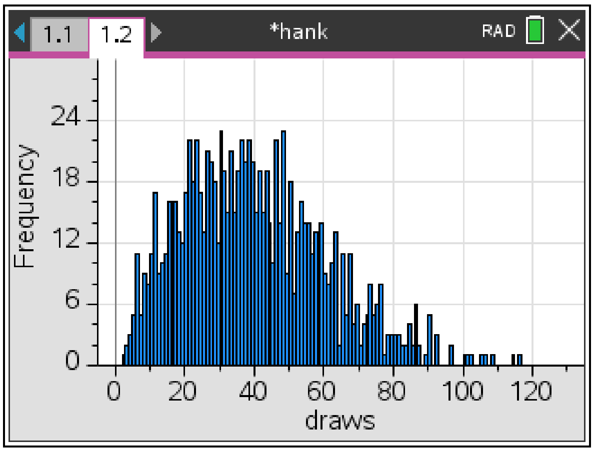
Let me draw your attention to TI-NspireCAS. **hank(n,u)** performs the same simulation as **`hanksim(n,u)`** with *DERIVE*. (Remember: expected value for 1000 objects is 40.30.)

```
hank(1000,1000)
              Number of necessary draws in list draws
              Avg number of nec. draws:  39.4965
                                                 Done
hank(1000,1000)
              Number of necessary draws in list draws
              Avg number of nec. draws:  39.0571
                                                 Done
hank(1000,1000)
              Number of necessary draws in list draws
              Avg number of nec. draws:  40.4725
                                                 Done
hank(1000,1000)
              Number of necessary draws in list draws
              Avg number of nec. draws:  39.1592
                                                 Done
```

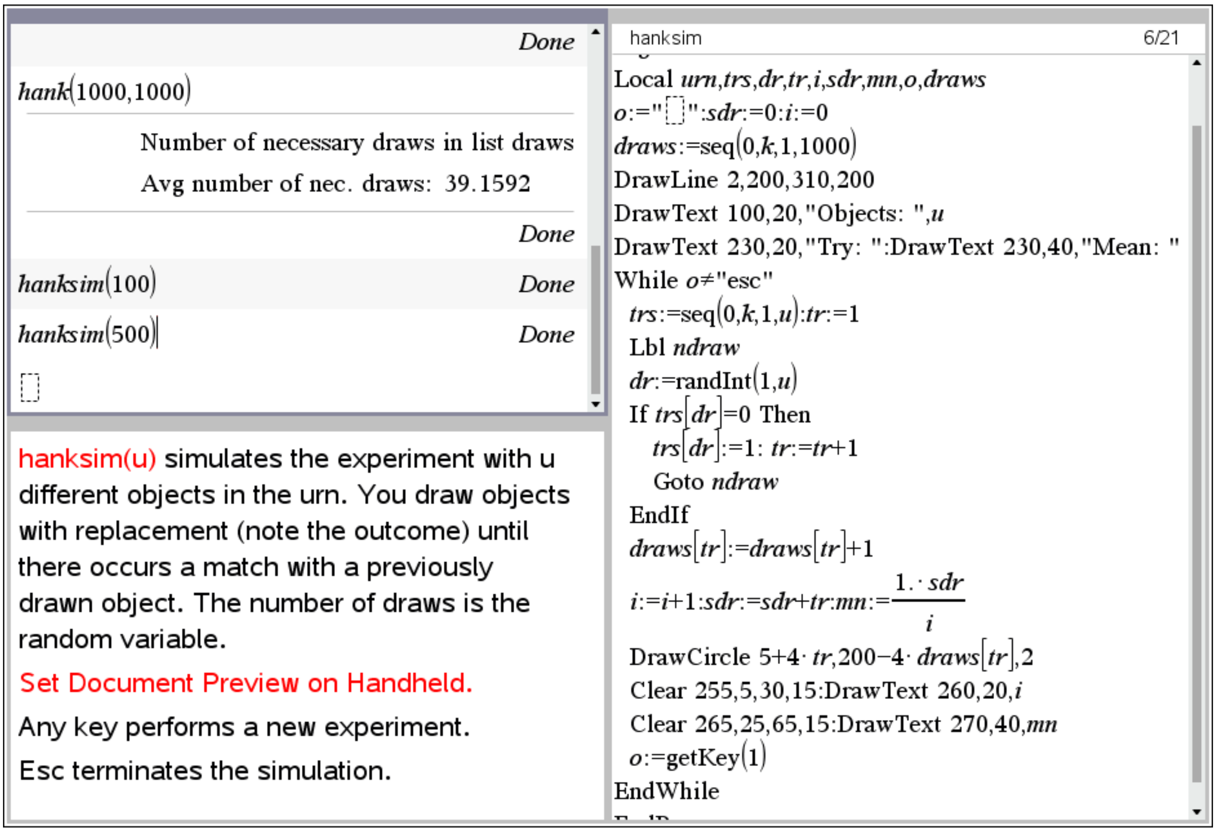
```
hank                                          1/16
Define hank(n,u)=
Prgm
Local trs,dr,tr,i
draws:={ }
i:=1
While i<n
trs:=seq(0,k,1,u):tr:=1
Lbl ndraw
dr:=randInt(1,u)
If trs[dr]=0 Then
  trs[dr]:=1:   tr:=tr+1
  Goto ndraw
EndIf
draws:=augment(draws,{tr})
i:=i+1
EndWhile
Disp "Number of necessary draws in list draws"
Disp "Avg number of nec. draws: ",1.· mean(draws)
EndPrgm
```

Here we don't need special statistics tools, because they are provided in the system:



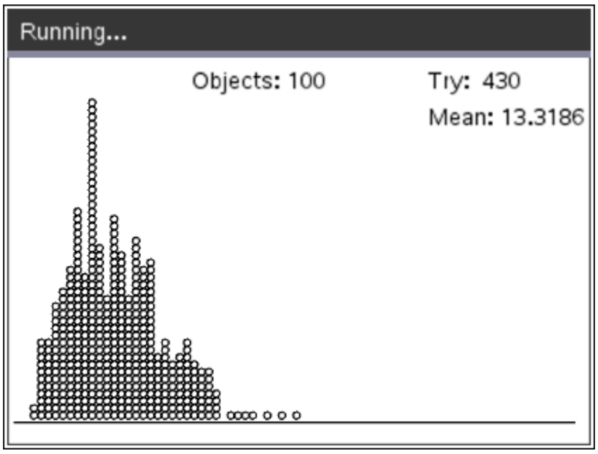


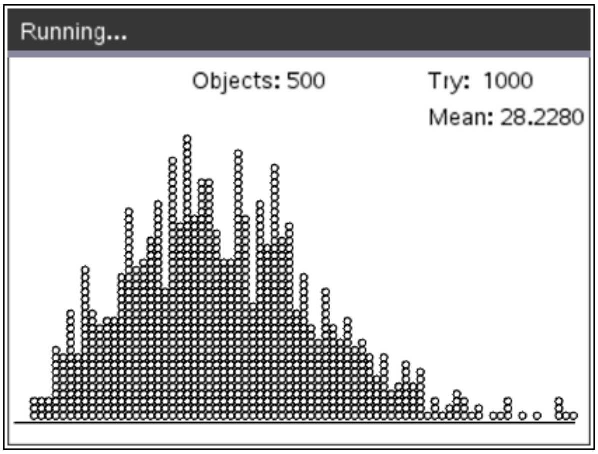
This is quite nice but we have not yet reached the end of the story. The latest version of TI-Nspire allows programming graphics objects. So, we can program a dynamic simulation and perform one experiment after the other, simultaneously watching the developing frequency diagram and the average.



All commands Draw… address the new feature. hanksim(u) does not run under TI-Nspire version 4 and below.

The left screen shows the simulation with 500 objects in the urn and we interrupt after 430 experiments for taking a screen shot. The right screen takes 500 objects and I stopped after 1000 experiments.





A possible extension would be to save the intermediate values of the mean and then present its development watching its convergence towards the theoretical expected value.

The definitive end of the “never-ending story” is my “translation” of Johann’s ultimate function for calculating the expected value. It is a one by one translation including the loop-loop which is typically for *DERIVE* (because there we don’t have a for-endfor-loop at our disposal).

```
Done
hank(1000,1000)
        Number of necessary draws in list draws
        Avg number of nec. draws:  39.0571
                                         Done
hank(1000,1000)
        Number of necessary draws in list draws
        Avg number of nec. draws:  40.4725
                                         Done
hank(1000,1000)
        Number of necessary draws in list draws
        Avg number of nec. draws:  39.1592
                                         Done
{hank2(10000),hank2(100000),hank2(1000000)}
               {125.9991,396.9997,1253.9809}
```

```
hank2                                    6/9
Define hank2(n)=
Func
Local s_,t_,u_,k_
s_:=1:t_:=1:k_:=1
Loop
  u_:=s_
  s_:=s_+t_
  If s_=u_:Return 1.·s_
  t_:=t_·(n-k_)/n
    k_:=k_+1
  EndLoop
EndFunc
```

I will close with one comment and one question.

Comment: I am very glad that I reanimated the collection of e-mails from 2001. I hope that the readers share my pleasure in the discussion and the results of the investigations.

Question: I would like to find an approximation or formula for a function describing the frequency diagrams (absolute or relative). It must be any skew density function?

Dave Dyer and Bill Yancey invited me to join their private "Math Group". Our idea is to work through Strogatz' book *Nonlinear Dynamics and Chaos*. Each chapter is followed by a couple of exercises and we try to solve selected ones. Then we exchange our results. While Bill is a master in traditional skills, I am trying to apply CAS and graphic features of DERIVE, TI-Nspire and other tools. Here is one of our results:

Manipulation Skills vs CAS

Bill Yancey, USA and Josef Böhm, Austria

2.2.6 Given is $\dot{x} = 1 - 2\cos x$.

a) Find the fixed points and discuss their stability. Sketch the respective vector field.

b) Sketch the graph of $x(t)$ for different initial conditions.

c) Find – if possible - the analytical solution of the differential equation.

BILL YANCEY

The fixed points are the zeros of the given differential equation.

For

$$\dot{x} = 1 - 2\cos x$$

We have fixed points with

$$1 - 2\cos x = 0$$

for

$$x = \pm\frac{\pi}{3} + 2k\pi.$$

The derivative changes from negative to positive for fixed points

$$x = \frac{\pi}{3} + 2k\pi$$

So, these fixed points are stable. Also

$$\ddot{x} = 2\sin x$$

So, the graphs inflect for $x = k\pi$. So for

$$\frac{\pi}{3} + 2k\pi < x_0 < -\frac{\pi}{3} + 2(k+1)\pi$$

The solutions increase asymptotically to $-\frac{\pi}{3} + 2(k+1)\pi$ and for

$$-\frac{\pi}{3} + 2k\pi < x_0 < \frac{\pi}{3} + 2k\pi$$

The solutions decrease asymptotically to $-\frac{\pi}{3} + 2k\pi.$

For an analytic solution using separation of variables,

$$\int \frac{1}{1 - 2\cos x}dx = \int dt$$

Using the substitution $z = \tan\left(\frac{x}{2}\right)$ which results in

$$\cos x = \frac{1-z^2}{1+z^2};\ dx = \frac{2}{1+z^2}dz$$

so that

$$\begin{aligned}\int \frac{1}{1-2\cos x}dx &= \int \frac{1}{1-2\frac{1-z^2}{1+z^2}}\frac{2}{1+z^2}dz\\ &= \int \frac{2}{3z^2-1}dz\\ &= \int\left(\frac{1}{\sqrt{3}z-1}-\frac{1}{\sqrt{3}z+1}\right)dz\\ &= \frac{1}{\sqrt{3}}\left(\log(\sqrt{3}z-1)-\log(\sqrt{3}z+1)\right)\\ &= \frac{1}{\sqrt{3}}\log\left(\frac{\sqrt{3}z-1}{\sqrt{3}z+1}\right)\end{aligned}$$

so in terms of z,

$$\begin{aligned}\frac{1}{\sqrt{3}}\log\left(\frac{\sqrt{3}z-1}{\sqrt{3}z+1}\right) &= t+C\\ \frac{\sqrt{3}z-1}{\sqrt{3}z+1} &= Ae^{\sqrt{3}}\\ \sqrt{3}z\left(1-Ae^{\sqrt{3}t}\right) &= 1+Ae^{\sqrt{3}t}\\ z &= \frac{1}{\sqrt{3}}\frac{1+Ae^{\sqrt{3}t}}{1-Ae^{\sqrt{3}t}}\end{aligned}$$

And since $z = \tan\frac{x}{2}$,

$$x = 2\arctan\left(\frac{1}{\sqrt{3}}\frac{1+Ae^{\sqrt{3}t}}{1-Ae^{\sqrt{3}t}}\right)$$

where

$$A = \frac{\sqrt{3}\tan\left(\frac{x_0}{2}\right)-1}{\sqrt{3}\tan\left(\frac{x_0}{2}\right)+1}.$$

JOSEF

I tried to answer the questions supported by a CAS, I must admit that it's a long time ago when I applied the integration rules (and sometimes tricks). So, I relied on *Derive* and other tools for calculating and plotting as well.

I start finding the zeros and plotting them:

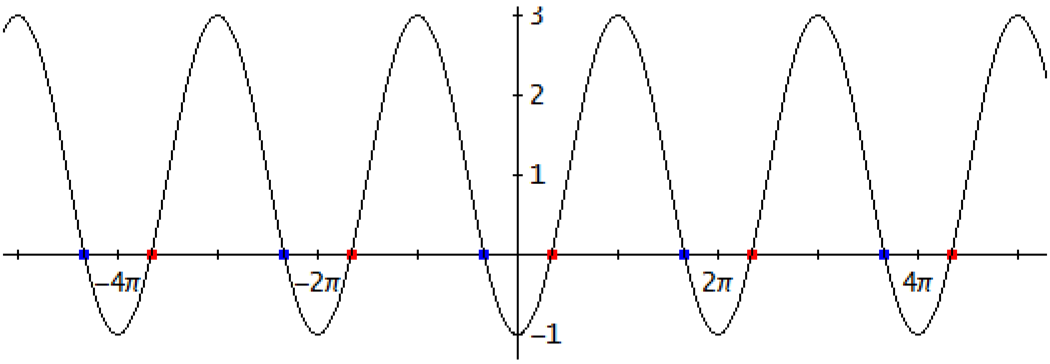
$$1 - 2\cdot COS(x)$$

$$SOLVE(1 - 2\cdot COS(x) = 0,\ x)$$

$$x = \frac{5\cdot\pi}{3} \vee x = -\frac{\pi}{3} \vee x = \frac{\pi}{3}$$

$$VECTOR\left(\left[\left[\frac{\pi}{3} + 2\cdot k\cdot\pi,\ 0\right]\right],\ k,\ -2,\ 2\right)$$

$$VECTOR\left(\left[\left[-\frac{\pi}{3} + 2\cdot k\cdot\pi,\ 0\right]\right],\ k,\ -2,\ 2\right)$$



Fixed points are the zeros; zeros with increasing function are unstable (red) , with decreasing function are stable (blue).

Derive is able to solve the respective differential equation but is not able to find an explicit form for $x(t)$.

$$DSOLVE1\_GEN(2\cdot COS(x) - 1,\ 1,\ t,\ x)$$

$$\frac{\sqrt{3}\cdot LN(COS(x) - \sqrt{3}\cdot SIN(x) + 1)}{3} - \frac{\sqrt{3}\cdot LN(COS(x) + \sqrt{3}\cdot SIN(x) + 1)}{3} - t = -c$$

$$\frac{\sqrt{3}\cdot LN(COS(x) - \sqrt{3}\cdot SIN(x) + 1)}{3} - \frac{\sqrt{3}\cdot LN(COS(x) + \sqrt{3}\cdot SIN(x) + 1)}{3} = -c + t$$

Logarithm := Collect

$$\sqrt{3}\cdot LN((COS(x) - \sqrt{3}\cdot SIN(x) + 1)^{1/3}) - \sqrt{3}\cdot LN((COS(x) + \sqrt{3}\cdot SIN(x) + 1)^{1/3}) = t - c$$

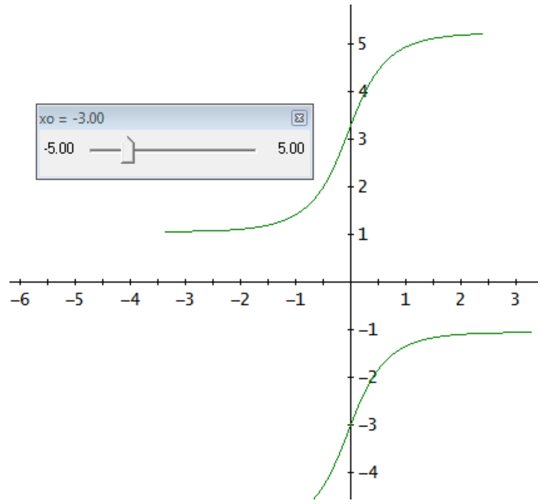
$$\frac{\sqrt{3}\cdot (LN(COS(x) - \sqrt{3}\cdot SIN(x) + 1) - LN(COS(x) + \sqrt{3}\cdot SIN(x) + 1))}{3} = t - c$$

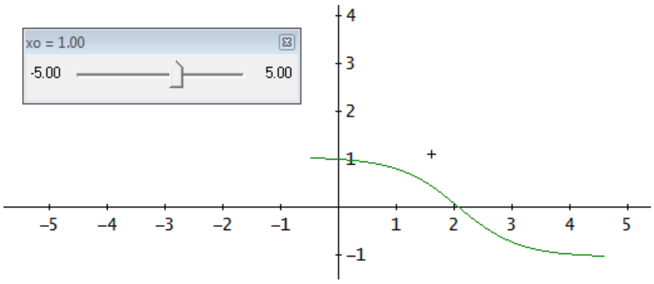
$$\frac{\sqrt{3}}{3}\cdot LN\left(\frac{COS(x) - \sqrt{3}\cdot SIN(x) + 1}{COS(x) + \sqrt{3}\cdot SIN(x) + 1}\right) = t - c$$

I set $x(t=0) = x_0$, solve for c and obtain the general solution (but only in implicit form):

$$\sqrt{3}\cdot \mathrm{LN}\left(\left(\frac{\mathrm{COS}(x) - \sqrt{3}\cdot \mathrm{SIN}(x) + 1}{\mathrm{COS}(x) + \sqrt{3}\cdot \mathrm{SIN}(x) + 1}\right)^{1/3}\right) - \sqrt{3}\cdot \mathrm{LN}\left(\left(\frac{\mathrm{COS}(x0) - \sqrt{3}\cdot \mathrm{SIN}(x0) + 1}{\mathrm{COS}(x0) + \sqrt{3}\cdot \mathrm{SIN}(x0) + 1}\right)^{1/3}\right) = t$$

Before comparing my solution with Bill's one – and trying other tools – I plot the solution for two values of x_o. In *Derive* I can introduce a slider for x_0.



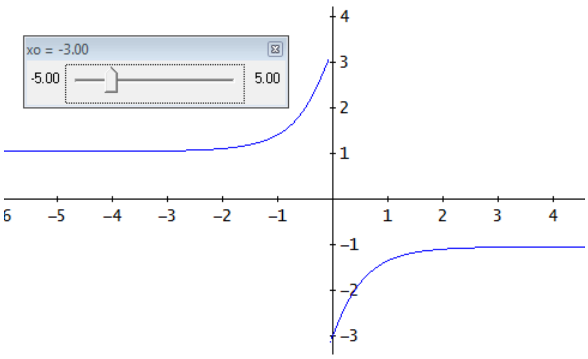


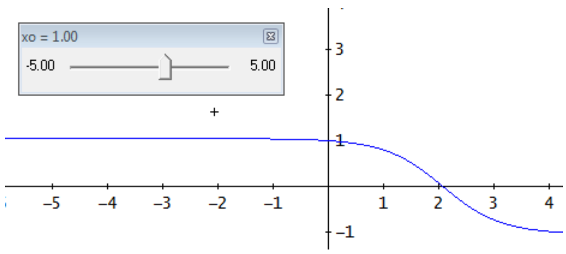
Now, let's plot Bill's solution (explicit function $x(t)$):

$$2\cdot \mathrm{ATAN}\left(\frac{\frac{1}{\sqrt{3}}\cdot\left(1 + a\cdot e^{\sqrt{3}\cdot x}\right)}{1 - a\cdot e^{\sqrt{3}\cdot x}}\right)$$

$$a := \frac{\sqrt{3}\cdot \mathrm{TAN}\left(\frac{xo}{2}\right) - 1}{\sqrt{3}\cdot \mathrm{TAN}\left(\frac{xo}{2}\right) + 1}$$

$$-\ 2\cdot \mathrm{ATAN}\left(\frac{\sqrt{3}\cdot\left(e^{\sqrt{3}\cdot x}\cdot(\mathrm{COS}(xo) - \sqrt{3}\cdot \mathrm{SIN}(xo) + 1) - \mathrm{COS}(xo) - \sqrt{3}\cdot \mathrm{SIN}(xo) - 1\right)}{3\cdot\left(e^{\sqrt{3}\cdot x}\cdot(\mathrm{COS}(xo) - \sqrt{3}\cdot \mathrm{SIN}(xo) + 1) + \mathrm{COS}(xo) + \sqrt{3}\cdot \mathrm{SIN}(xo) + 1\right)}\right)$$





Maxima fails giving the implicit form of the solution:

```
(%i6)    ode2('diff(x,t)=1-2*cos(x), x, t);
```

(%o6)
$$\frac{\log\left(\frac{3\sin(x)-\sqrt{3}\cos(x)-\sqrt{3}}{3\sin(x)+\sqrt{3}\cos(x)+\sqrt{3}}\right)}{\sqrt{3}}=t+\%c$$

TI-NspireCAS gives an explicit result for $x(t)$, which is a very bulky expression:

Solution obtained by TI-NspireCAS

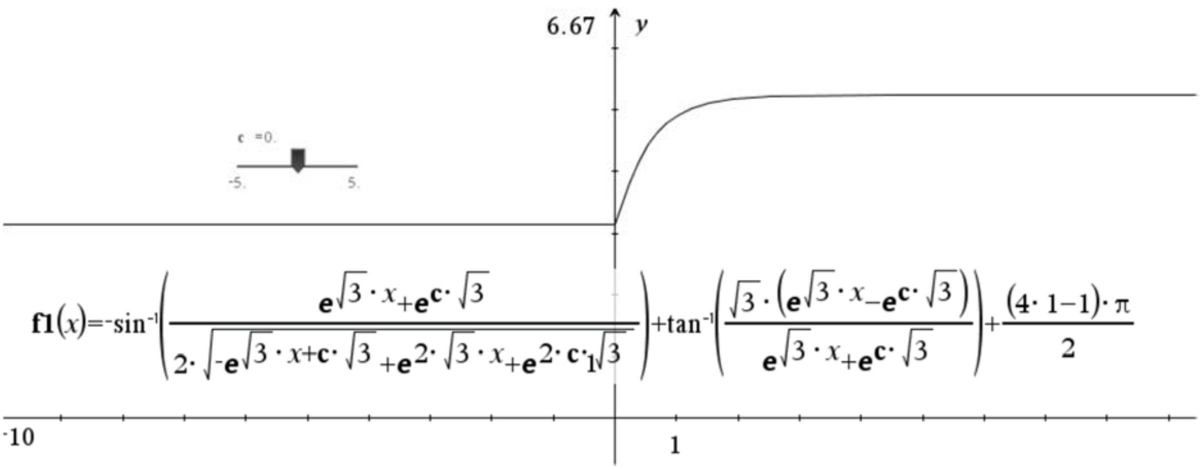
deSolve$(x'=1-2\cdot\cos(x),t,x)$

$$\frac{-\sqrt{3}\cdot\ln\left(\frac{-\cos(x)+\sqrt{3}\cdot\sin(x)-1}{\cos(x)+\sqrt{3}\cdot\sin(x)+1}\right)}{3}=c3-t$$

$$\text{solve}\left(\frac{-\sqrt{3}\cdot\ln\left(\frac{-\cos(x)+\sqrt{3}\cdot\sin(x)-1}{\cos(x)+\sqrt{3}\cdot\sin(x)+1}\right)}{3}=c3-t,x\right)$$

$$x=-\sin^{-1}\left(\frac{e^{\sqrt{3}\cdot t}+e^{c3\cdot\sqrt{3}}}{2\cdot\sqrt{-e^{\sqrt{3}\cdot t+c3\cdot\sqrt{3}}+e^{2\cdot\sqrt{3}\cdot t}+e^{2\cdot c3\cdot\sqrt{3}}}}\right)+\tan^{-1}\left(\frac{\sqrt{3}\cdot\left(e^{\sqrt{3}\cdot t}-e^{c3\cdot\sqrt{3}}\right)}{e^{\sqrt{3}\cdot t}+e^{c3\cdot\sqrt{3}}}\right)+\frac{(4\cdot n1-1)\cdot\pi}{2}\text{ and }-2\leq\frac{\quad}{\sqrt{-e^{\sqrt{3}}}}$$

I was not able to solve for the parameter ***c3***. So, I introduced a slider for ***c3*** (= c) and set ***n1*** = 1. The resulting function graph is not very satisfying. (You must imagine a *t*-*x*- system instead of *x*-*y*-axes.)



but MATHEMATICA is successful - and presents a very short and compact general solution:

$$x(t)=2\tan^{-1}\left(\frac{\tanh\left(\frac{1}{2}\left(\sqrt{3}\,c_1-\sqrt{3}\,t\right)\right)}{\sqrt{3}}\right)$$

FACTORING TRINOMIALS
A STUDENT'S PERSPECTIVE

DUNCAN E. McDOUGALL and PEGA ALERASOOL

duncanemcdougall@hotmail.com

(I)

It is when we least expect it that a student will make an observation about a conventional method that we virtually take for granted. In this case, it was factoring

The Trinomial form $ax^2 + bx + c$ by decomposition.

In particular, I presented $3x^2 - 2x - 8$ for consideration. The question posed was: "How can we factor this if the 3 is not a perfect square?"

Pega Alerasool, an industrious student with a different view of this procedure, did not understand the conventional algorithm as it had been presented to her. It was clear that my perception of this method and hers were very different. So, after explaining that the 3 or any other coefficient of the x^2 term didn't have to be a perfect square, she asked, "Why can't it be a perfect square?" That is, why couldn't the coefficient of x^2 always be a perfect square? Now I was curious and asked her to explain her approach. In essence, let $A = 3x^2 - 2x - 8$ then multiply both sides by 3 in order to make the coefficient of x^2 a perfect square. The advantage here is that when we apply decomposition, we do not have to worry about the positions of positioning of the constant terms. The illustrated format looks like:

	Factor $3x^2 - 2x - 8$	Given
Let	$A = 3x^2 - 2x - 8$	Substitution
then	$3A = 9x^2 - 2(3)x - 24$	Multiplication
	$3A = (3x - 6)(3x + 4)$	$9x^2 = 3x \bullet 3x$ and the factors of –24 whose sum is –2 are –6 and 4

(II)

As with previous methods of factoring, the position of the variable and constant terms is crucial. Here, however, we no longer have this worry. Since the numerical coefficient of the ***x*** term is the same in both brackets, it does not matter where to place the constant factors, in the case –6 and 4. Continuing this process we have

$3A = (3x - 6)(3x + 4)$	from above
$3A = 3(x - 2)(3x + 4)$	factor out the g.c.f.
$A = (x - 2)(3x + 4)$	Division

In general, the sequence looks like this:

Factor	$ax^2 + bx + c$	{where $b = a + c$ to be factorable
Let	$A = ax^2 + bx + c$	{ by substitution
Then	$Aa = a^2x^2 + abx + ac$	{by multiplication to make the coefficient
		{ of x^2 a perfect square
Now	$Aa = (ax + a)(ax + c)$	{factors of *ac* are *a* and *c* regardless of order
	$Aa = a(x + 1)(ax + c)$	{factor out the g.c.f.
	$A = (x + 1)(ax + c)$	{division by *a*

The only real question remaining is what to do when the coefficient of x^2 in a given question is already a perfect square. As tempting as it is to factor as is, it doesn't work as shown below (diagram 1). We still multiply the coefficient of x^2 by itself so that we can get the constant factors to work properly, as in (diagram 2).

(III)

<u>Diagram (1)</u>

$4x^2 - 3x - 1$
$= (2x - 1)(2x + 1)$
does not work

<u>Diagram (2)</u>

Let	$A = 4x^2 - 3x - 1$	substitution
	$4A = 16x^2 - 3(4)x - 4$	multiplication by 4
	$4A = (4x - 4)(4x + 1)$	$16x^2 = 4x \bullet 4x$ and $-4 = -4 - 1$ and $-3 = -4 + 1$
	$4A = 4(x - 1)(4x + 1)$	factor out the g.c.f.
	$A = (x - 1)(4x + 1)$	division

The above was Pega's take on how factoring by decomposition. This is how it made sense to her. If it appeals to other students like Pega, then it has become an alternative method to factoring. Try it, you'll like it.

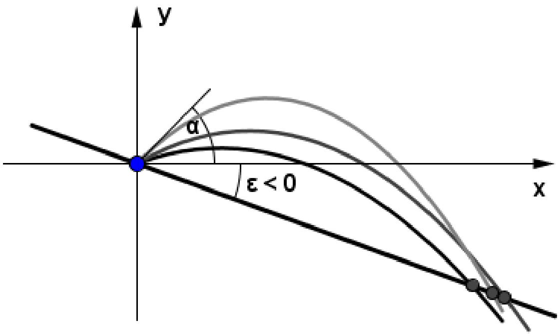
Wurfweite am schrägen Hang

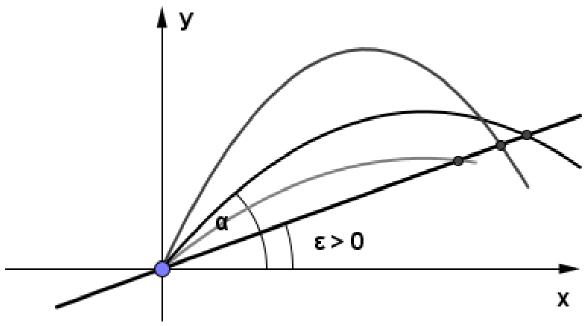
Throwing an Object on an inclined Hillside

Wolfgang Alvermann, Germany

Das folgende Problem, gefunden in *Georg Gläser: Der mathematische Werkzeugkasten*, bietet aus der Fragestellung - Unter welchem Winkel α muss ein Gegenstand mit der Geschwindigkeit v_0 abgeschossen werden, um auf einem schrägen Hang mit dem Neigungswinkel ε möglichst weit zu fliegen? – einige mathematische Aufgaben.

The following problem – found in *Georg Gläser: Der mathematische Werkzeugkasten* – offers starting with the question – which launching angle α and given velocity v_0 is necessary to reach the maximum flight distance on a hillside with inclination ε? – some mathematical tasks.





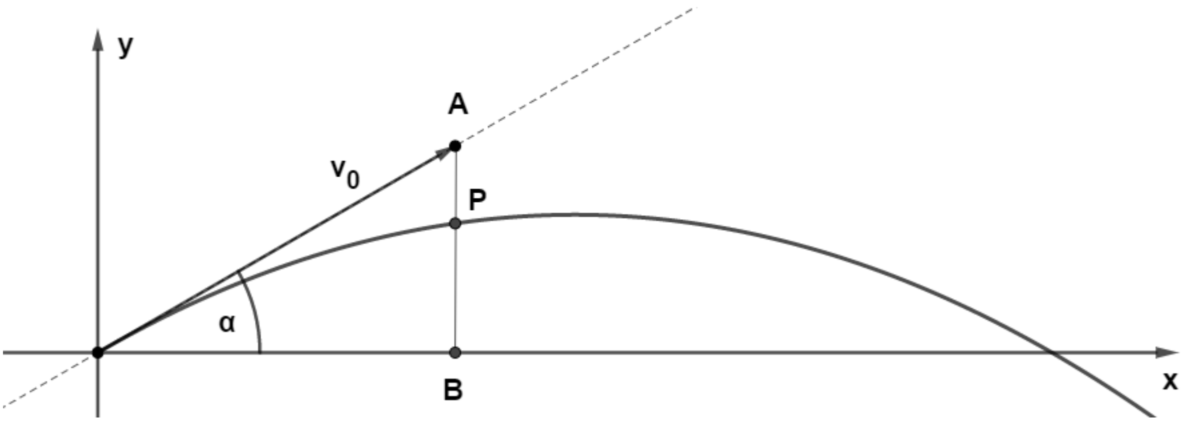
Grundsätzlich handelt es sich um das Problem *Schnittpunkt Gerade – Parabel.*

Die Gerade hat die Gleichung $g(x) = \tan(\varepsilon) \cdot x = m \cdot x$

Die Parabel (Wurfparabel) *p*(x) wird aus der folgenden Grafik hergeleitet.

The basic problem is finding the intersection line – parabola.

The equation of the line is $g(x) = \tan(\varepsilon) \cdot x = m \cdot x$. The equation of the parabola *p*(*x*) can be derived from the following graph.



We can read off the parameter form (1), (2) and obtain the equation of the trajectory parabola *p*(*x*) by eliminating parameter *t* (time).

Der beliebige Punkt P der Wurfparabel kann in Parameterform beschrieben werden:

$$(1)\ \ x = v_0 \cdot t \cdot \cos(\alpha)$$

$$(2)\ \ y = v_0 \cdot t \cdot \sin(\alpha) - \frac{g}{2} \cdot t^2$$

Gleichung (1) wird umgeformt nach *t*:

$$(3)\ \ t = \frac{x}{v_0 \cdot \cos(\alpha)}$$

Dieser Term in (2) eingesetzt ergibt

$$\boxed{p(x) = x \cdot \tan(\alpha) - \frac{g}{2 \cdot v_0^2 \cdot \cos^2(\alpha)} \cdot x^2}$$

Then we calculate the intersection points of *p*(*x*) and *g*(*x*) and receive the *x*-coordinates of the two solution points.

Zu berechnen sind nun die Schnittpunkte von *p*(*x*) und *g*(*x*); durch Gleichsetzen ergibt sich:

$$\frac{g}{2 \cdot v_0^2 \cdot \cos^2(\alpha)} \cdot x^2 + x \cdot [m - \tan(\alpha)] = 0$$

mit den Lösungen

$$x_1 = 0$$

$$x_2 = \frac{2 \cdot v_0^2}{g} \cdot [\tan(\alpha) - m] \cdot \cos^2(\alpha)$$

The maximum distance is reached if $x_2 = f(\alpha)$ becomes a maximum.

$x_2 = f(\alpha)$ soll ein Maximum werden; es genügt,

$$f(\alpha) = \frac{1}{2} \cdot \sin(2\alpha) - m \cdot \cos^2(\alpha)$$

zu betrachten.

$$f'(\alpha) = \cos(2\alpha) - m \cdot \sin(2\alpha) \Rightarrow \tan(2\alpha) = -\frac{1}{m}$$

Für α erhält man also zunächst folgende Lösungsformel:

The formula for the requested angle *α* is given by:

$$\boxed{\alpha = f(\varepsilon) = \frac{1}{2} \cdot \arctan\left(-\frac{1}{\tan(\varepsilon)}\right)}$$

Beispiele / Examples

$$\varepsilon = -45° \Rightarrow \alpha = \frac{1}{2} \cdot \arctan\left(-\frac{1}{\tan(-45°)}\right) \qquad \underline{\alpha = 22.5°}$$

$$\varepsilon = -25° \Rightarrow \alpha = \frac{1}{2} \cdot \arctan\left(-\frac{1}{\tan(-25°)}\right) \qquad \underline{\alpha = 32.5°}$$

$$\varepsilon = 35° \Rightarrow \alpha = \frac{1}{2} \cdot \arctan\left(-\frac{1}{\tan(35°)}\right) \qquad \alpha = -27.5° \ \ ???$$

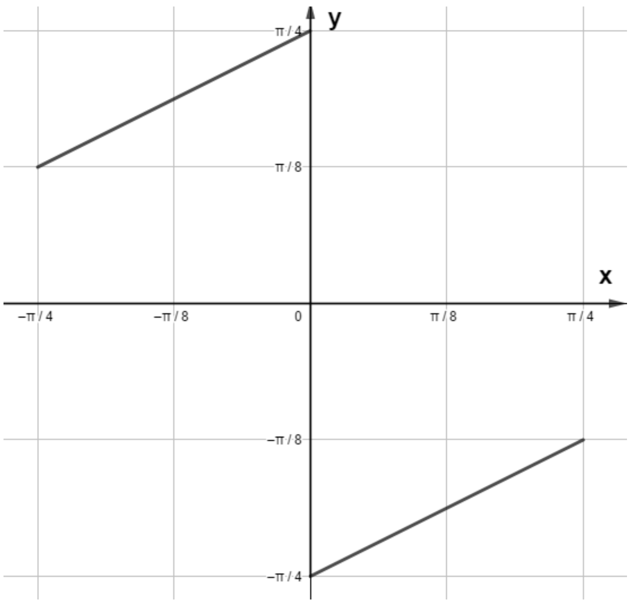
Das letzte Ergebnis gibt Anlass zu weiteren Überlegungen; für negative ε ergibt sich der korrekte Wert für *α*, für positive ε-Werte jedoch negative *α*-Werte.

Dies ist aber – siehe Zeichnung – nicht möglich.

Daher soll die Funktion $f(\varepsilon)=\frac{1}{2}\cdot\arctan\left(-\frac{1}{\tan(\varepsilon)}\right)$ grafisch dargestellt werden.

The last result gives reason for further considerations; for $\varepsilon < 0$ we receive correct values for *α*, but for $\varepsilon > 0$ we have negative values for *α*. This is according to the sketch impossible! Therefor we will give a graphic representation of function $f(\varepsilon)=\frac{1}{2}\cdot\arctan\left(-\frac{1}{\tan(\varepsilon)}\right)$.

Inspecting the graph, we notice that the lower branch is shifted by π/2. We can read off the slope *m* = 1/2. Attaching the lower branch to the upper one in (0; π/2) leads to the correct relationship between *α* and ε.



Bei genauer Betrachtung fällt auf, dass der untere Ast der Funktion um π/2 versetzt ist.

Aus der Grafik lässt sich auch die Steigung *m* ablesen: $m=\frac{1}{2}$. Setzt man den unteren Ast oben in $\left(0;\frac{\pi}{4}\right)$ an, erhält man die korrekte Beziehung zwischen *α* und ε.

$$\boxed{\alpha = f(\varepsilon)=\frac{1}{2}\cdot\varepsilon+\frac{\pi}{4} \quad \text{für} \quad -\frac{\pi}{4}<\varepsilon<\frac{\pi}{4}}$$

So liefert $f(35°)=62.5°$

Now we have $f(35°)=62.5°$

The intersection point given above with $\alpha=\frac{\varepsilon}{2}+\frac{\pi}{4}$ leads to the general formula for the *x*-coordinate – dependent on ε and v_0 of the point of impact.

Der oben angegebene Schnittpunkt *x* ergibt zusammen mit $\alpha=\frac{\varepsilon}{2}+\frac{\pi}{4}$ die allgemeine Formel für die *x*-Koordinate des Treffpunkts (abhängig von ε und v_0):

$$\boxed{x = w(v_0,\varepsilon)=\frac{2\cdot v_0^2}{g}\cdot\left[\frac{1}{2}\cdot\cos(\varepsilon)-\tan(\varepsilon)\cdot\cos^2\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)\right]}$$

Die Wurfweite w am schrägen Hang erhält man dann über $\cos(\varepsilon) = \frac{x}{w}$, also $w = \frac{x}{\cos(\varepsilon)}$:

Calculating the distance along the hillside applying $\cos(\varepsilon) = \frac{x}{w} \rightarrow w = \frac{x}{\cos(\varepsilon)}$:

$$w(v_0, \varepsilon) = \frac{2 \cdot v_0^2}{g} \cdot \left[\frac{1}{2} \cdot \cos(\varepsilon) - \tan(\varepsilon) \cdot \cos^2\left(\frac{\varepsilon}{2} + \frac{\pi}{4}\right)\right] \cdot \frac{1}{\cos(\varepsilon)}, \text{ g = 9.81 m/s}^2.$$

Mit einem CAS vereinfacht erhält man die einfache Formel:

Simplifying this expression using a CAS we obtain the following short formula:

$$\boxed{w(v_0, \varepsilon) = \frac{v_0^2}{g \cdot \cos^2(\varepsilon)} \cdot [1 - \sin(\varepsilon)]}; \text{ g = 9.81 m/s}^2$$

(Beachten Sie bitte die folgende CAS-Bearbeitung, bzw. auch die händische Durchführung der notwendigen „altmodischen" Umformungen, die ich leider schon ziemlich vergessen habe, Josef!)

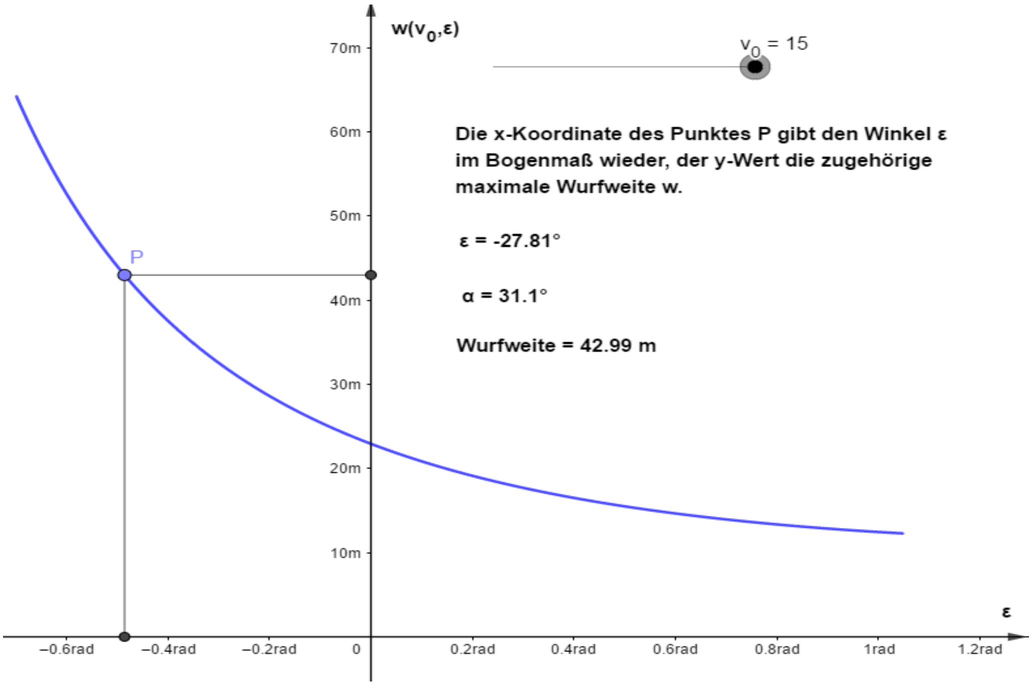
(Please notice the following CAS-treatment and how to perform the necessary trig-manipulations using "old fashioned" techniques – which I hardly can remember, Josef!)

Dieser Zusammenhang ist in der folgenden Grafik dargestellt:

- In der GeoGebra-Datei kann der Punkt P auf der o. g. Funktion verschoben werden, dadurch ändern sich die Winkelwerte sowie die Wurfweite.
- Verändert man am Schieberegler v_0, wird die Lage der Kurve flacher und die Wurfweite geringer; ε und α bleiben unverändert.

This relationship is presented in the following graph:

- Point P can be moved on the graph which results in changing values for the angles and the throw distance.
- Moving the slider for v_0 gives a flatter curve and a shorter distance; ε and α remain unchanged.



Comment of the Editor: I liked Wolfgang's contribution because in my opinion it is very suitable for the classroom. So, I transferred the calculation and the plots to DERIVE and TI-Nspire. See the TI-Nspire version on a Notes page:

The inclined hillside

$\mathbf{g}(x):=\tan(\varepsilon)\cdot x$ ▸ *Done* calculation

$\mathbf{f1}(x):=\tan(\mathbf{eps})\cdot x$ ▸ *Done* plot

The parabola (from parameter form to explicit equation):

$x=v0\cdot t\cdot \cos(\alpha)$ ▸ $x=v0\cdot \cos(\alpha)\cdot t$

$y=v0\cdot t\cdot \sin(\alpha)-\frac{g\_}{2}\cdot t^2$ ▸ $y=v0\cdot \sin(\alpha)\cdot t-\frac{g\_\cdot t^2}{2}$

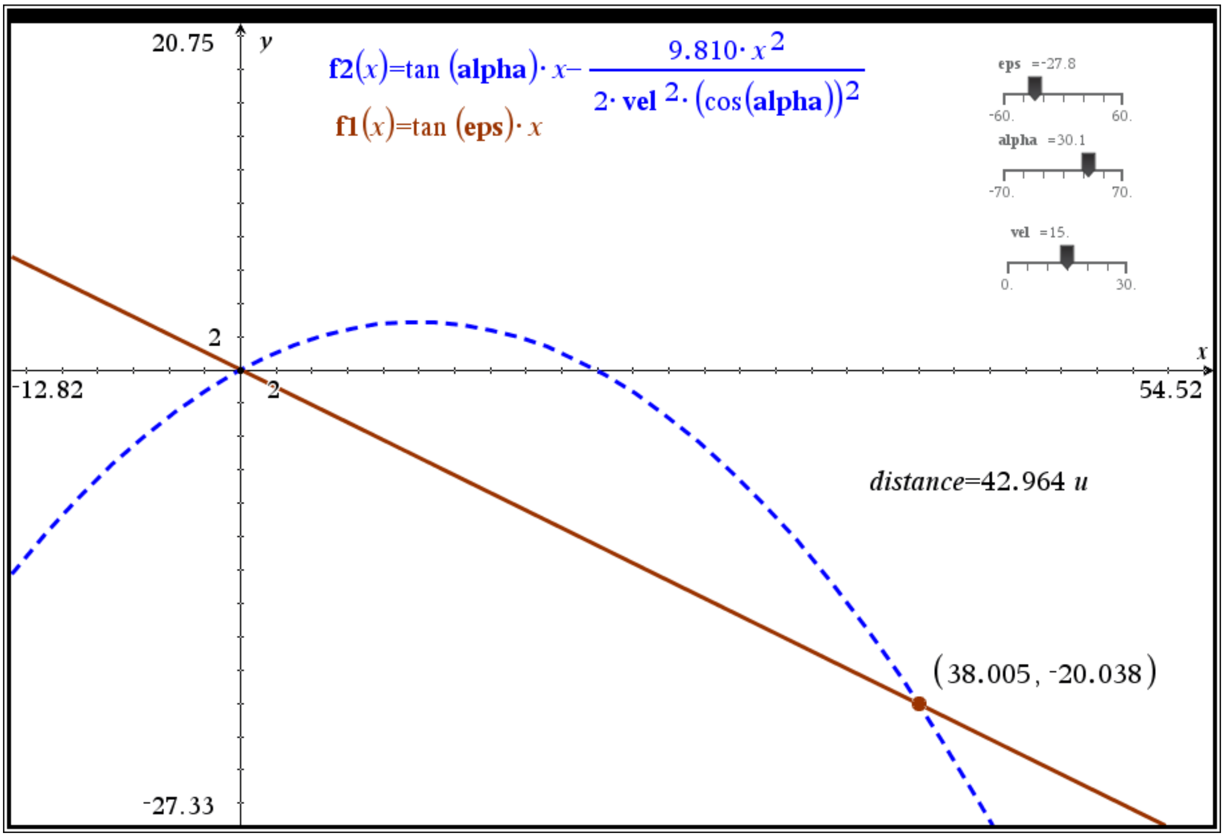
$\text{solve}(x=v0\cdot t\cdot \cos(\alpha),t)$ ▸ $t=\frac{x}{v0\cdot \cos(\alpha)}$

$\mathbf{p}(x):=v0\cdot t\cdot \sin(\alpha)-\frac{g\_}{2}\cdot t^2|t=\frac{x}{v0\cdot \cos(\alpha)}$ ▸ *Done* calculation

$\mathbf{p}(x)$ ▸ $\tan(\alpha)\cdot x-\frac{g\_\cdot x^2}{2\cdot v0^2\cdot (\cos(\alpha))^2}$ ⚠

$\mathbf{f2}(x):=\tan(\mathbf{alpha})\cdot x-\frac{9.81\cdot x^2}{2\cdot \mathbf{vel}^2\cdot (\cos(\mathbf{alpha}))^2}$ ▸ *Done* plot

I differ between function to plot (f1, f2) and to calculate (in order to avoid problems with the variables used for the sliders)



I illustrate the problem using three sliders for all involved parameters. Then I proceed performing the necessary calculations:

Intersection point between line and parabola (dependent on α):

$\text{solve}(\mathbf{g}(x)=\mathbf{p}(x),x)$

$\blacktriangleright\ x=\frac{-2\cdot v0^2\cdot\cos(\alpha)\cdot(\cos(\alpha)\cdot\sin(\varepsilon)-\sin(\alpha)\cdot\cos(\varepsilon))}{g\_\cdot\cos(\varepsilon)}$ or $x=0$ or $\frac{1}{v0^2\cdot(\cos(\alpha))^2\cdot\cos(\varepsilon)}=0$ ⚠

Finding the extremal value for x−coordinate of intersection:

$\frac{d}{d\alpha}(\cos(\alpha)\cdot(\cos(\alpha)\cdot\sin(\varepsilon)-\sin(\alpha)\cdot\cos(\varepsilon)))$

$\blacktriangleright\ -(\cos(\alpha))^2\cdot\cos(\varepsilon)-2\cdot\sin(\alpha)\cdot\cos(\alpha)\cdot\sin(\varepsilon)+(\sin(\alpha))^2\cdot\cos(\varepsilon)$

$\text{solve}((\cos(\alpha))^2\cdot\cos(\varepsilon)+2\cdot\sin(\alpha)\cdot\cos(\alpha)\cdot\sin(\varepsilon)-(\sin(\alpha))^2\cdot\cos(\varepsilon)=0,\alpha)$

$\blacktriangleright\ \sin(2\cdot\alpha)\cdot\sin(\varepsilon)+(2\cdot(\cos(\alpha))^2-1)\cdot\cos(\varepsilon)=0$ ⚠

$\text{tCollect}(\sin(2\cdot\alpha)\cdot\sin(\varepsilon)+(2\cdot(\cos(\alpha))^2-1)\cdot\cos(\varepsilon))\ \blacktriangleright\ \cos(2\cdot\alpha-\varepsilon)$ ⚠

α is the angle to throw for given ε of the slope

$\text{solve}(\cos(2\cdot\alpha-\varepsilon)=0,\alpha)\ \blacktriangleright\ \alpha=\frac{2\cdot\varepsilon+(2\cdot n1-1)\cdot\pi}{4}$

ε is the angle of the slope for a given throwing angle

$\text{solve}(\cos(2\cdot\alpha-\varepsilon)=0,\varepsilon)\ \blacktriangleright\ \varepsilon=2\cdot\alpha+\frac{(2\cdot n2-1)\cdot\pi}{2}$

I don't change Nspire's standard angle-setting (radian) but I want to enter and receive the angles in degrees!

$\mathbf{ang}(slope):=\frac{\left(\frac{slope\cdot\pi}{2\cdot 180}+\frac{\pi}{4}\right)\cdot 180}{\pi}\ \blacktriangleright\ Done$

compare with results from above!

$\mathbf{ang}(-25)\ \blacktriangleright\ 32.5$ $\quad\mathbf{ang}(-27.77)\ \blacktriangleright\ 31.115$

$\mathbf{ang}(27.77)\ \blacktriangleright\ 58.885$ $\quad\mathbf{ang}(35)\ \blacktriangleright\ 62.5$

$\mathbf{x\_max}(\varepsilon,v0):=\frac{-2\cdot v0^2\cdot\cos(\alpha)\cdot(\cos(\alpha)\cdot\sin(\varepsilon)-\sin(\alpha)\cdot\cos(\varepsilon))}{g\_\cdot\cos(\varepsilon)}|\alpha=\frac{2\cdot\varepsilon+(2-1)\cdot\pi}{4}\ \blacktriangleright\ Done$

$\mathbf{x\_max}(\varepsilon,v0)\ \blacktriangleright\ \frac{-2\cdot v0^2\cdot\cos\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)\cdot\left(\sin(\varepsilon)\cdot\cos\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)-\cos(\varepsilon)\cdot\sin\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)\right)}{g\_\cdot\cos(\varepsilon)}$

Let's call the "miracle worker" CAS!

$$\text{tCollect}\left(\frac{-2\cdot v0^2\cdot \cos\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)\cdot\left(\sin(\varepsilon)\cdot\cos\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)-\cos(\varepsilon)\cdot\sin\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)\right)}{g\_\cdot\cos(\varepsilon)}\right) ▸ \frac{v0^2\cdot(1-\sin(\varepsilon))}{g\_\cdot\cos(\varepsilon)}$$

Gives a very compact expression for the x-coordinate of the farthest reachable point depending on v_0 and ε.

$$\mathbf{x\_m\_\varepsilon}(\varepsilon,v0):=\frac{v0^2\cdot(1-\sin(\varepsilon))}{g\_\cdot\cos(\varepsilon)} ▸ \textit{Done}$$

$$\mathbf{x\_m\_\varepsilon}\left(\frac{-27.77\cdot\pi}{180},15\right)|g\_=9.81 ▸ 37.9986 \qquad \mathbf{x\_m\_\varepsilon}\left(\frac{27.77\cdot\pi}{180},15\right)|g\_=9.81 ▸ 13.8439$$

$$\mathbf{x\_max\alpha}(\alpha,v0):=\frac{-2\cdot v0^2\cdot\cos(\alpha)\cdot(\cos(\alpha)\cdot\sin(\varepsilon)-\sin(\alpha)\cdot\cos(\varepsilon))}{g\_\cdot\cos(\varepsilon)}|\varepsilon=2\cdot\alpha+\frac{(2-1)\cdot\pi}{2} ▸ \textit{Done}$$

$$\mathbf{x\_max\alpha}(\alpha,v0) ▸ \frac{v0^2\cdot(\cos(\alpha)\cdot\cos(2\cdot\alpha)+\sin(\alpha)\cdot\sin(2\cdot\alpha))}{g\_\cdot\sin(\alpha)}$$

$$\text{tCollect}\left(\frac{v0^2\cdot(\cos(\alpha)\cdot\cos(2\cdot\alpha)+\sin(\alpha)\cdot\sin(2\cdot\alpha))}{g\_\cdot\sin(\alpha)}\right) ▸ \frac{v0^2}{g\_\cdot\tan(\alpha)}$$

Gives a very compact expression for the x-coordinate of the farthest reachable point depending on v_0 and α.

And this was the pleasant surprise: a wonderful compact result for the requested maximum distance appeared, which provoked Wolfgang to reproduce it applying his trig-manipulating skills.

$$\mathbf{x\_m\_\alpha}(\alpha,v0):=\frac{v0^2}{g\_\cdot\tan(\alpha)} ▸ \textit{Done}$$

$$\mathbf{x\_m\_\alpha}\left(\frac{31.12\cdot\pi}{180},15\right)|g\_=9.81 ▸ 37.9911 \qquad \mathbf{x\_m\_\alpha}\left(\frac{58.885\cdot\pi}{180},15\right)|g\_=9.81 ▸ 13.8439$$

This is the maximum distance measured on the ground (along the hillside):

$$\mathbf{m\_dist}(\alpha,v0):=\frac{\mathbf{x\_m\_\alpha}(\alpha,v0)}{\left|\cos\left(2\cdot\alpha+\frac{\pi}{2}\right)\right|}|g\_=9.81 ▸ \textit{Done} \qquad \mathbf{m\_dist}\left(\frac{31.12\cdot\pi}{180},15\right) ▸ 42.9323$$

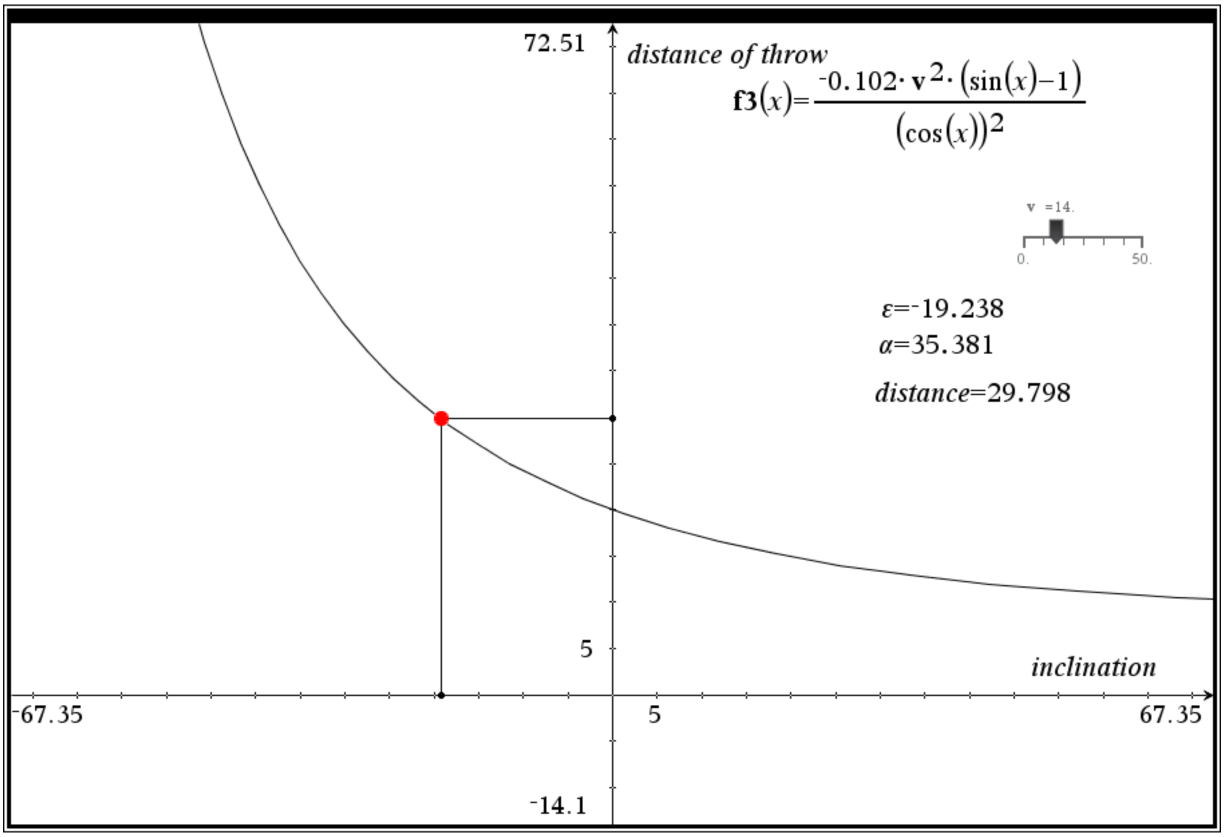
$$\mathbf{m\_dist\_\varepsilon}(x,\mathrm{v}):=\frac{\mathbf{x\_m\_\varepsilon}(x,\mathrm{v})\cdot g\_}{\cos(x)\cdot 9.81} ▸ \textit{Done}$$

$$\mathbf{m\_dist\_\varepsilon}(x,\mathrm{v}) ▸ \frac{-19.9796\cdot(\sin(x)-1)}{(\cos(x))^2}$$

$$\mathrm{f3}(x):=\frac{-0.101937\cdot\mathbf{v}^2\cdot(\sin(x)-1)}{(\cos(x))^2} ▸ \textit{Done} \qquad \text{plot}$$

$$\frac{-0.101937\cdot 15^2\cdot(\sin(x)-1)}{(\cos(x))^2}|x=\frac{-27.77\cdot\pi}{180} ▸ 42.9448$$

See finally Wolfgang's GeoGebra-animation reproduced on the TI-Nspire-screen. Unfortunaly, we cannot move the point on the graph so easily in *DERIVE*, but what we can do is tracing the graph and this is a good alternative.



It should be no problem to analyze the throw using DERIVE, too.

Trigonometrie macht Spaß – Trigonometry is Fun

Auf der Seite 29 ist dargelegt, dass das CAS die Gleichung

On page 29 we wrote that a CAS simplifies the equation

$$w(v_0,\varepsilon)=\frac{2\cdot v_0^2}{g}\cdot\left[\frac{1}{2}\cdot\cos(\varepsilon)-\tan(\varepsilon)\cdot\cos^2\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)\right]\cdot\frac{1}{\cos(\varepsilon)}$$

with little effort to / mit wenig Aufwand vereinfacht wird zu

$$w(v_0,\varepsilon)=\frac{v_0^2}{g\cdot\cos^2(\varepsilon)}\cdot\left[1-\sin(\varepsilon)\right]$$

We show that this can be achieved (in an easy way??) by manual calculating skills.

$$w(v_0,\varepsilon)=\frac{2\cdot v_0^2}{g}\cdot\left[\frac{1}{2}\cdot\cos(\varepsilon)-\tan(\varepsilon)\cdot\cos^2\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)\right]\cdot\frac{1}{\cos(\varepsilon)}$$

$$w(v_0,\varepsilon)=\frac{2\cdot v_0^2}{g}\cdot\left[\frac{1}{2}-\frac{\sin(\varepsilon)}{\cos^2(\varepsilon)}\cdot\cos^2\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)\right]$$ and further to

$$w(v_0,\varepsilon)=\frac{v_0^2}{g\cdot\cos^2(\varepsilon)}\cdot\left[\cos^2(\varepsilon)-2\cdot\sin(\varepsilon)\cdot\cos^2\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)\right]$$

Formula $\cos(\alpha+\beta)=\cos(\alpha)\cdot\cos(\beta)-\sin(\alpha)\cdot\sin(\beta)$ applies on $2\cdot\cos^2\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)$:

$$2\cdot\cos^2\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)=2\cdot\left[\cos\left(\frac{\varepsilon}{2}\right)\cdot\frac{\sqrt{2}}{2}-\sin\left(\frac{\varepsilon}{2}\right)\cdot\frac{\sqrt{2}}{2}\right]^2$$

$$2\cdot\cos^2\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)=\left[\underbrace{\cos^2\left(\frac{\varepsilon}{2}\right)+\sin^2\left(\frac{\varepsilon}{2}\right)}_{1}-\underbrace{2\cdot\sin\left(\frac{\varepsilon}{2}\right)\cdot\cos\left(\frac{\varepsilon}{2}\right)}_{\sin(\varepsilon)}\right]$$

$$2\cdot\cos^2\left(\frac{\varepsilon}{2}+\frac{\pi}{4}\right)=1-\sin(\varepsilon)$$

So, we obtain:

$$w(v_0,\varepsilon)=\frac{v_0^2}{g\cdot\cos^2(\varepsilon)}\cdot\left[\cos^2(\varepsilon)-\sin(\varepsilon)\cdot(1-\sin(\varepsilon))\right]$$

$$w(v_0,\varepsilon)=\frac{v_0^2}{g\cdot\cos^2(\varepsilon)}\cdot\left[\cos^2(\varepsilon)-\sin(\varepsilon)+\sin^2(\varepsilon)\right]$$

and finally, the CAS-result:

$$\boxed{w(v_0,\varepsilon)=\frac{v_0^2}{g\cdot\cos^2(\varepsilon)}\cdot\left[1-\sin(\varepsilon)\right]}$$

Wer braucht schon ein CAS?

Who does really need a CAS?

Request from Guiseppe Ornaghi, Italy

Dear Josef,

I'm taking advantage of your kindness to ask you a question.

I have found that Derive, when possible, is able to transform (a + sqrt(b))^(1/3) into x + sqrt(y).

Do you know where to find the algorithm that Derive uses?

Regards,

Giuseppe

I must admit that I don't have any idea. I constructed three examples (hoping that they correspond with Guiseppe's question),

Josef

Examples?

#1: $(77\cdot\sqrt{2}+155)^{1/3}=\sqrt{2}+5$

#2: $(232\cdot\sqrt{2}+360)^{1/3}=2\cdot\sqrt{2}+6$

#3: $(153\cdot\sqrt{6}+469)^{1/3}=\sqrt{6}+7$

Functional Derivative

Francisco M. Fernández, Argentina

framfer@gmail.com

In classical mechanics the Lagrangian $L(q,\dot{q})$ is a function of the generalized coordinates $q = (q_1, q_2, \ldots, q_n)$ and their derivatives with respect to time $\dot{q} = (\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n)$, $\dot{q}_i = \frac{dq_i}{dt}$ (or generalized velocities). The n second-order differential equations that describe the motion of the physical system in the case of consecutive forces are given by

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \left(\frac{\partial L}{q_i}\right) = 0,\ i = 1, 2, \ldots, n \tag{1}$$

The Lagrangian is given by $L = T - V$, where T and V are the kinetic and potential energies.

The Hamiltonian $H(q,p)$ is a function of the generalized coordinates q and momenta $p = (p_1, p_2, \ldots, p_n)$ given by

$$p_i = \frac{\partial L}{\partial \dot{q}_i},\ i = 1, 2, \ldots, n \tag{2}$$

It is related to the Lagrangian by

$$H = \sum_{i=1}^{n} p_i \dot{q}_i - L \tag{3}$$

And enables to describe the motion of the system by means of $2n$ differential equations of first order

$$\dot{q}_i = \frac{\partial H}{\partial p_i},\ \dot{p}_i = -\frac{\partial H}{\partial q_i}. \tag{4}$$

In order to carry out these calculations we need a derivative of a function with respect to another function, which is not built in in Derive. For this reason, I propose the following functional derivative

$$FD(f, a, \xi) := LIM(DIF(SUBST(f, a, \xi), \xi), \xi, a). \tag{5}$$

For some unknown reason (at least to me) the use of SUBST or LIM twice does not work.

As an example, I choose the textbook problem of a particle of mass m moving in a two-dimensional space under the effect of a potential $V(r)$ that depends on the distance $r = \sqrt{x^2 + y^2}$ to the origin. It is convenient to resort to polar coordinates $x = r\cos\theta$, $y = r\sin\theta$ as generalized coordinates $q = (r,\theta)$

In this case

$$T = \frac{m}{2}\left(\dot{x}^2 + \dot{y}^2\right). \tag{6}$$

The accompanying commented file shows how to carry out the calculations in Derive.

Definition of the functional derivative

#1: $FD(f, a, \xi) := \lim_{\xi \to a} \frac{d}{d\xi} SUBST(f, a, \xi)$

Generalized coordinates

#2: [r(t) :=, θ(t) :=]

Potential-energy function

#3: V(r) :=

Coordinate transformation

#4: $[x(t) := r(t) \cdot COS(\theta(t)),\ y(t) := r(t) \cdot SIN(\theta(t))]$

Cartesian momenta

#5: $px(t) := m \cdot \frac{d}{dt} x(t)$

#6: $px(t) := m \cdot r'(t) \cdot COS(\theta(t)) - m \cdot r(t) \cdot \theta'(t) \cdot SIN(\theta(t))$

#7: $py(t) := m \cdot \frac{d}{dt} y(t)$

#8: $py(t) := m \cdot r(t) \cdot \theta'(t) \cdot COS(\theta(t)) + m \cdot r'(t) \cdot SIN(\theta(t))$

Lagrangian in generalized coordinates

#9: $L := \frac{m}{2} \cdot \left(\left(\frac{d}{dt} x(t) \right)^2 + \left(\frac{d}{dt} y(t) \right)^2 \right) - V(r(t))$

#10: $L := \frac{m \cdot r'(t)^2 + m \cdot r(t)^2 \cdot \theta'(t)^2 - 2 \cdot V(r(t))}{2}$

Lagrangian equations of motion

#11: $\frac{d}{dt} FD(L, r'(t)) - FD(L, r(t)) = 0$

#12: $m \cdot r''(t) - m \cdot r(t) \cdot \theta'(t)^2 + V'(r(t)) = 0$

#13: $\frac{d}{dt} FD(L, \theta'(t)) - FD(L, \theta(t)) = 0$

#14: $m \cdot r(t) \cdot (2 \cdot \theta'(t) \cdot r'(t) + r(t) \cdot \theta''(t)) = 0$

The z-component of the angular momentum Lz = x py - y px

#15: $x(t) \cdot py(t) - y(t) \cdot px(t) = m \cdot \theta'(t) \cdot r(t)^2$

is a constant of the motion (see #13)

#16: $\frac{d}{dt}(x(t)\cdot py(t) - y(t)\cdot px(t)) = m\cdot r(t)\cdot(2\cdot\theta'(t)\cdot r'(t) + r(t)\cdot\theta''(t))$

Generalized momenta

#17: [pr(t) :=, pθ(t) :=]

#18: [pr(t) = FD(L, r'(t)), pθ(t) = FD(L, θ'(t))]

#19: $\left[pr(t) = m\cdot r'(t),\ p\theta(t) = m\cdot\theta'(t)\cdot r(t)^2\right]$

pθ = Lz is a constant of the motion

#20: $\frac{d}{dt}(p\theta(t) = m\cdot\theta'(t)\cdot r(t)^2)$

#21: $p\theta'(t) = m\cdot r(t)\cdot(2\cdot\theta'(t)\cdot r'(t) + r(t)\cdot\theta''(t))$

The Hamiltonian

#22: $r'(t)\cdot m\cdot r'(t) + \theta'(t)\cdot m\cdot\theta'(t)\cdot r(t)^2 - L$

#23: $\frac{m\cdot r'(t)^2 + m\cdot r(t)^2\cdot\theta'(t)^2 + 2\cdot V(r(t))}{2}$

should be written in terms of the generalized momenta

#24: $\mathrm{SUBST}\left(\frac{m\cdot r'(t)^2 + m\cdot r(t)^2\cdot\theta'(t)^2 + 2\cdot V(r(t))}{2},\ [r'(t),\ \theta'(t)],\ \left[\frac{pr(t)}{m},\ \frac{p\theta(t)}{m\cdot r(t)^2}\right]\right)$

#25: $V(r(t)) + \frac{pr(t)^2}{2\cdot m} + \frac{p\theta(t)^2}{2\cdot m\cdot r(t)^2}$

Hamiltonian equations of motion

#26: $\frac{d}{dt}\theta(t) = \mathrm{FD}\left(V(r(t)) + \frac{pr(t)^2}{2\cdot m} + \frac{p\theta(t)^2}{2\cdot m\cdot r(t)^2},\ p\theta(t)\right)$

#27: $\theta'(t) = \frac{p\theta(t)}{m\cdot r(t)^2}$

#28: $\frac{d}{dt}p\theta(t) = -\mathrm{FD}\left(V(r(t)) + \frac{pr(t)^2}{2\cdot m} + \frac{p\theta(t)^2}{2\cdot m\cdot r(t)^2},\ \theta(t)\right)$

#29: $p\theta'(t) = 0$

$$\#30: \quad \frac{d}{dt} r(t) = FD\left(V(r(t)) + \frac{pr(t)^2}{2 \cdot m} + \frac{p\theta(t)^2}{2 \cdot m \cdot r(t)^2}, pr(t)\right)$$

$$\#31: \quad r'(t) = \frac{pr(t)}{m}$$

$$\#32: \quad \frac{d}{dt} pr(t) = -FD\left(V(r(t)) + \frac{pr(t)^2}{2 \cdot m} + \frac{p\theta(t)^2}{2 \cdot m \cdot r(t)^2}, r(t)\right)$$

$$\#33: \quad pr'(t) = \frac{p\theta(t)^2}{m \cdot r(t)^3} - V'(r(t))$$

Short information(from the web):

Definition of *Lagrangian*

: a function that describes the state of a dynamic system in terms of position coordinates and their time derivatives and that is equal to the difference between the potential energy and kinetic energy.

Definition of *Hamiltonian*

: a function that is used to describe a dynamic system (such as the motion of a particle) in terms of components of momentum and coordinates of space and time and that is equal to the total energy of the system when time is not explicitly part of the function.

Links to Lagrangian and Hamiltonian:

http://www.physicsinsights.org/lagrange_1.html

https://brilliant.org/wiki/lagrangian-formulation-of-mechanics/

http://www.people.fas.harvard.edu/~djmorin/chap6.pdf

https://www.quora.com/What-is-the-difference-between-a-Lagrangian-and-a-Hamiltonian

Anageo (2)

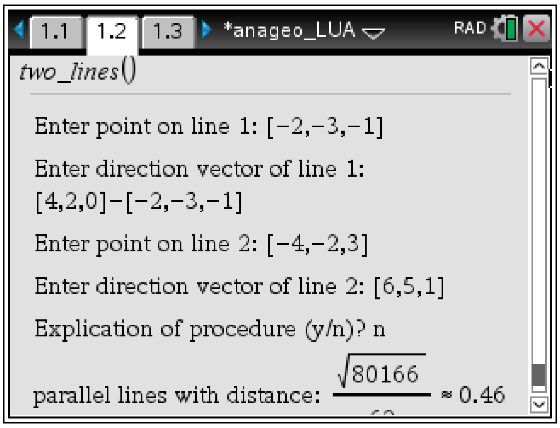
Jens Staacke, Borna, Germany and Josef Böhm (supported by
Steve Arnold, Australia and Geneviève Savard, Canada))

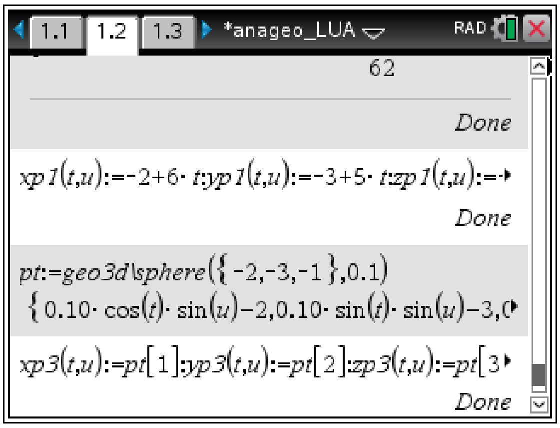
Next problem(s): What is the relationship between

a) line *g*: [(-2,-3,-1), (4,2,0)] and line *k*: $x = \begin{pmatrix} -4 \\ -2 \\ 3 \end{pmatrix} + t \cdot \begin{pmatrix} 6 \\ 5 \\ 1 \end{pmatrix}$

b) line *g* and line *k*: [(0,0,0), (4,-2,-2)]

c) line *g* and line *k*: [(-3,4,2), (1,0,4)]



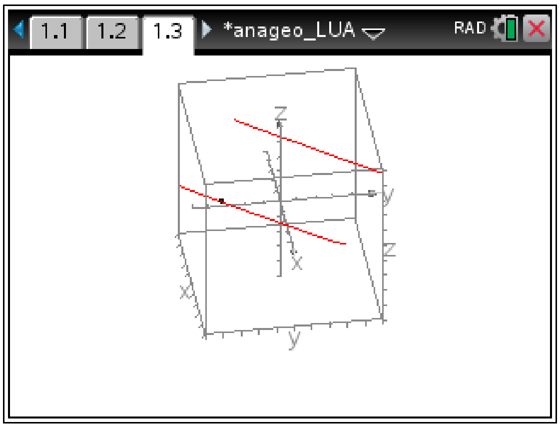


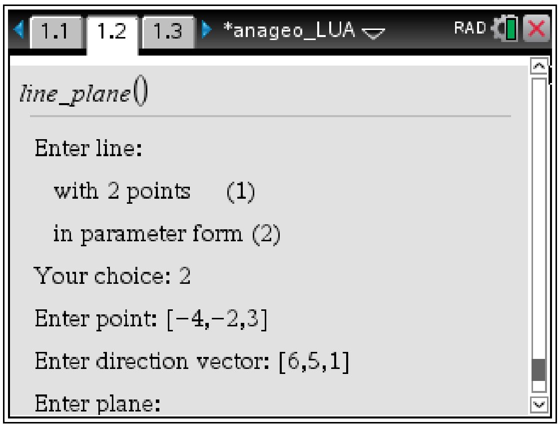
Case a)

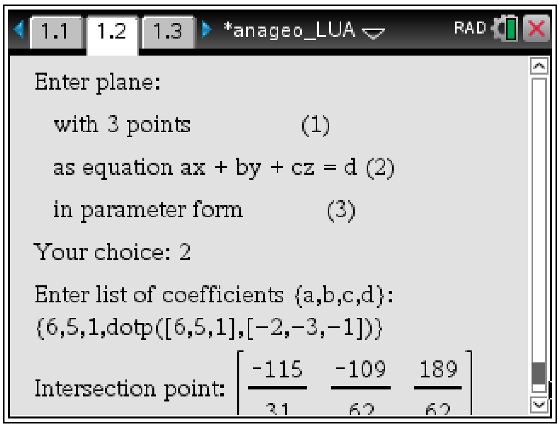
The lines are parallel and their distance is given in the program output.

I represent both lines and one point first because I'd like to visualize their distance: I intersect line 2 with a plane normal to line 1 passing the first given point.

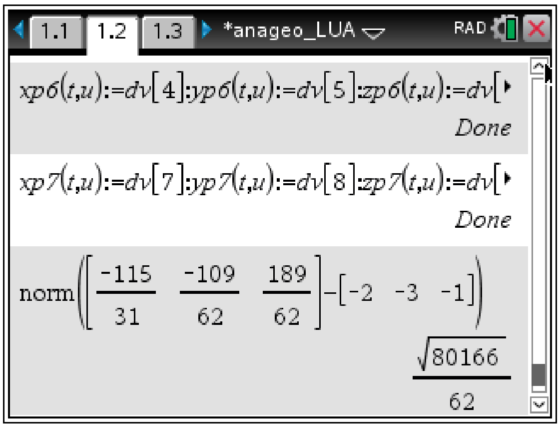
The distance between the two points is the distance between the lines.

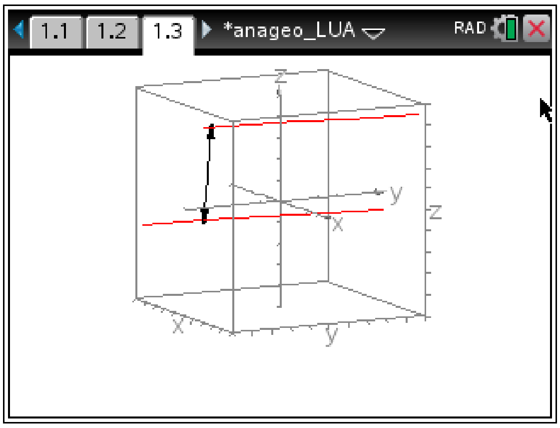






(Without Lua: Load anageo.tns; *anageo_menu*() calls the German package, *ana_menu*() the English one)

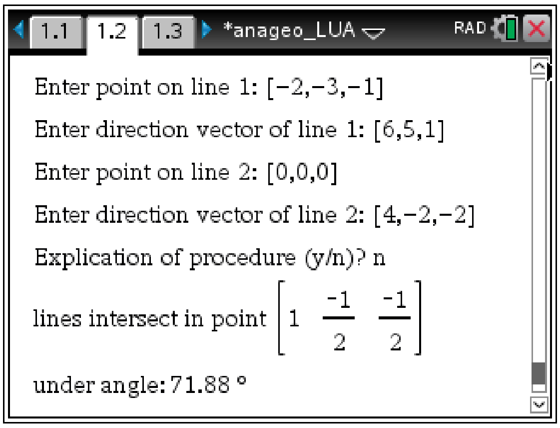


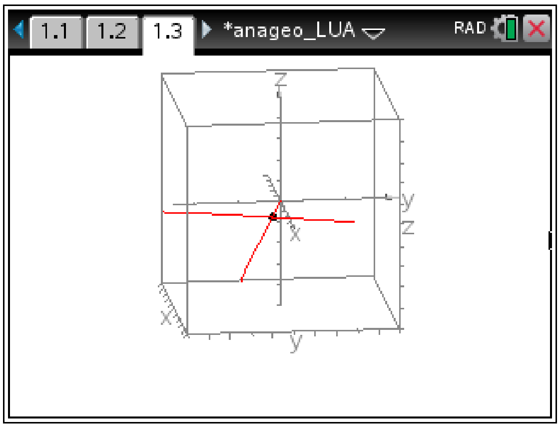


As an extra the distance is represented by a double arrow. (My function *d_vec*() is an extension of Geneviève Savard's geo3d.tns library, geo3d_ext.tns – same syntax as *vec*().)

Case b)

The lines have a common point.

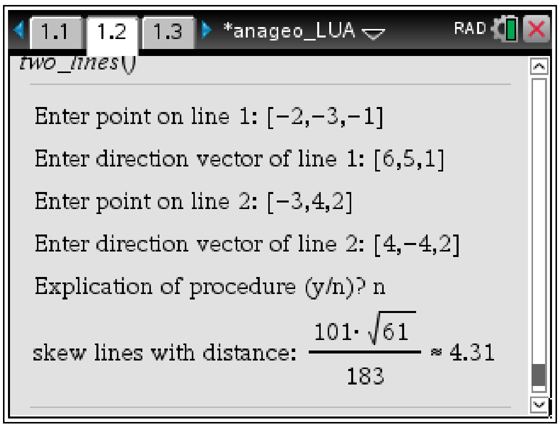


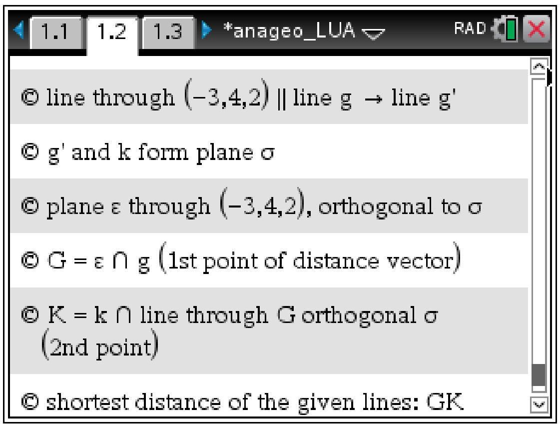


Case c)

This is the demanding case (not for the program – maybe for the programmer, but for students to calculate or to construct).

The right screen below describes the procedure:

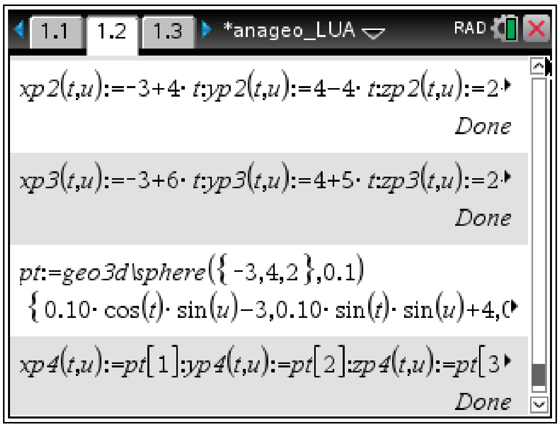


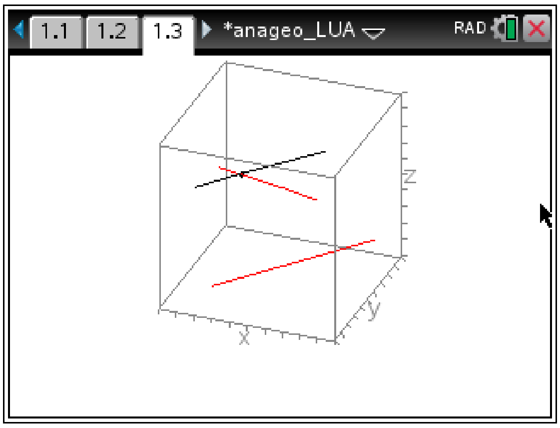


I start plotting the two lines in TI-Nspire's 3D-Graphing View. The given lines (red) and line *g'* (black) in parameter form. I enter the parameter equations in the calculator.

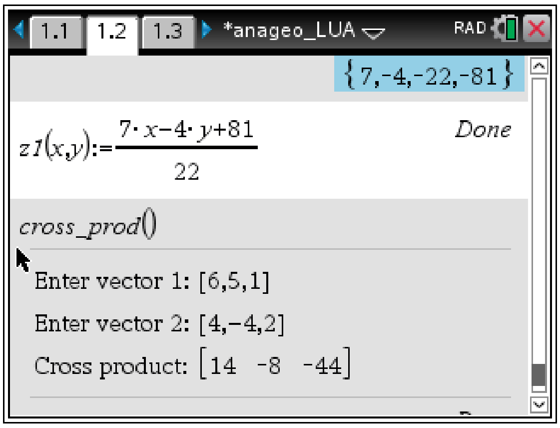
The first point of line *k* is added as a small sphere.

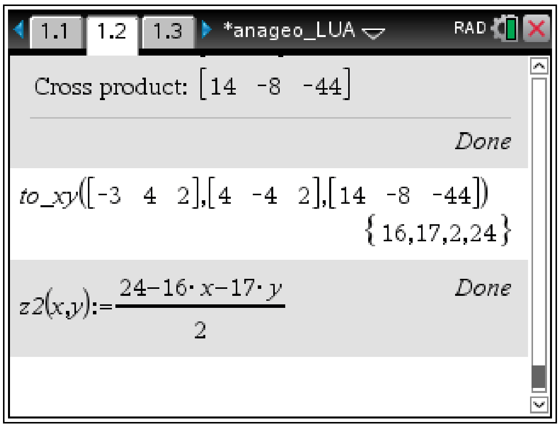
The axes are hidden to not overload the small screen.



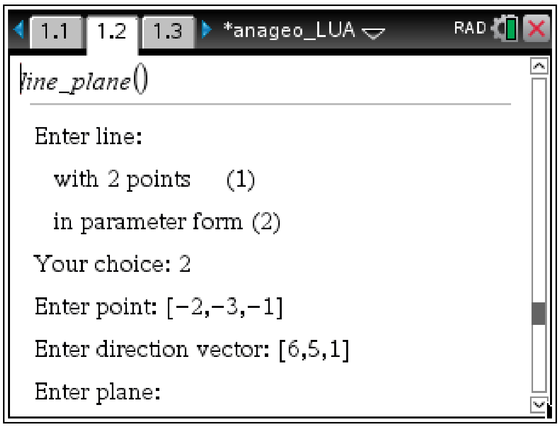


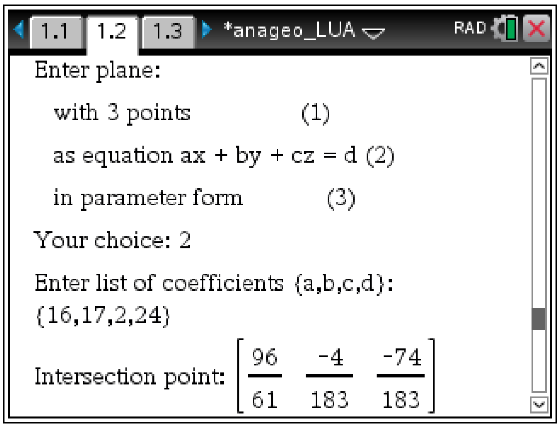
I follow the procedure described above step-by-step. *to_xy(a,u,v)* gives the coefficients of the plane defined by point *a* and direction vectors *u* and *v*. Plane *σ* is plotted as *z1(x,y)*. In a similar way we get *z2(x,y)* is which is plane *ε*.



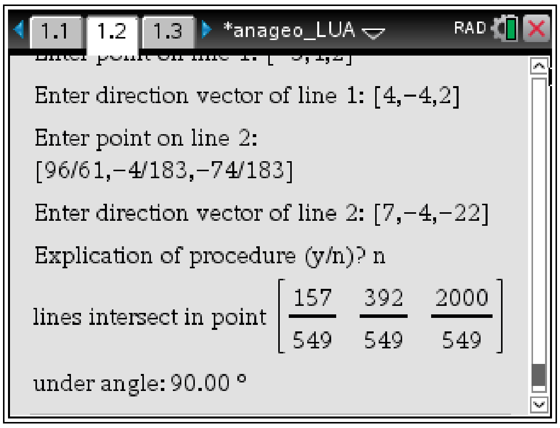


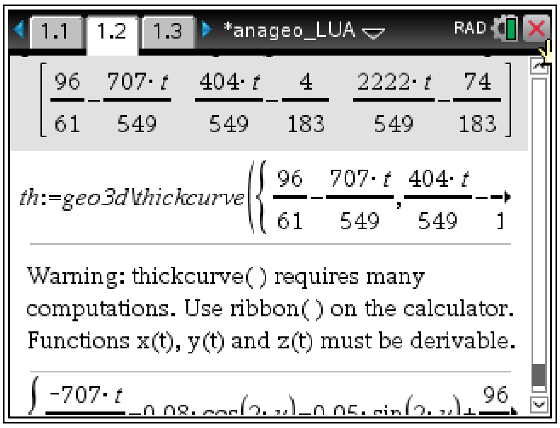
Point *G* is the intersection point of plane *ε* and line *g*.





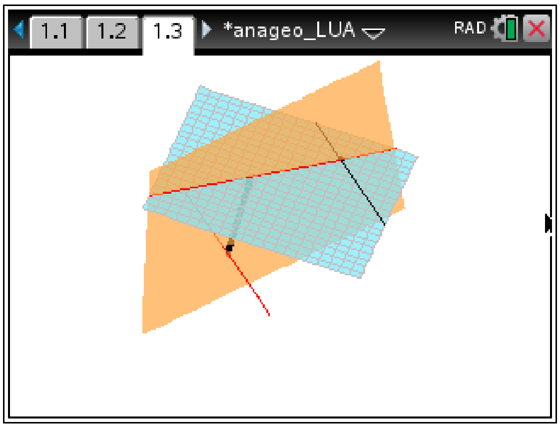
This is point *G*.





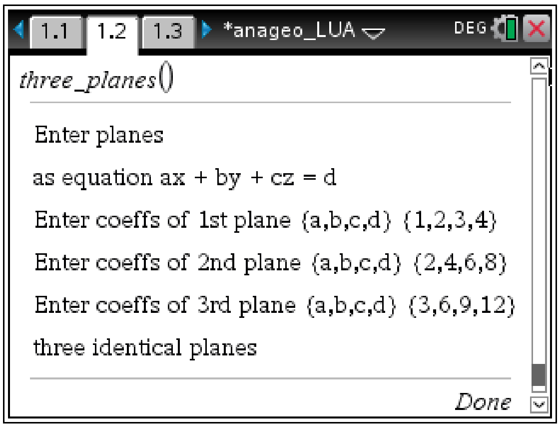
The shortest distance of the two lines is visualized as a *thickcurve* (provided in the geo3d library).

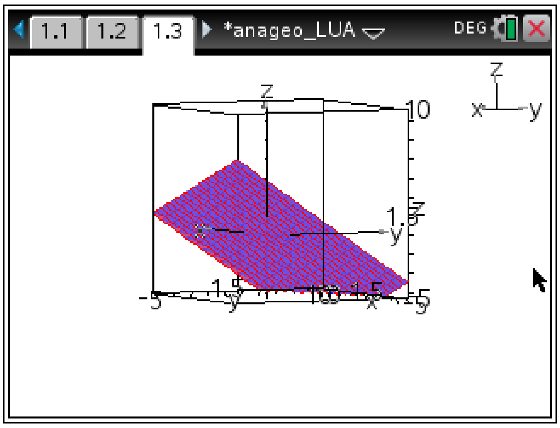
I will skip the *two_planes* and treat the more interesting case of possible relationships between three planes using the program *three_planes*.



Last task: Three planes in space – how can they behave?

Given planes are: $x + 2y + 3z = 4$, $2x + 4y + 3z = 8$, $3x + 6y + 8z = 12$





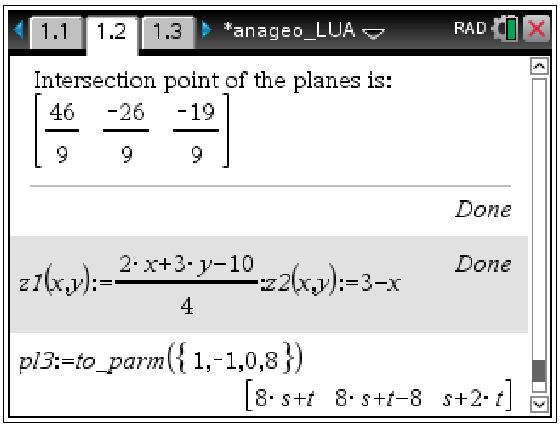
This is the easiest case, of course – doesn't need any program but I'd like to consider really all possible cases.

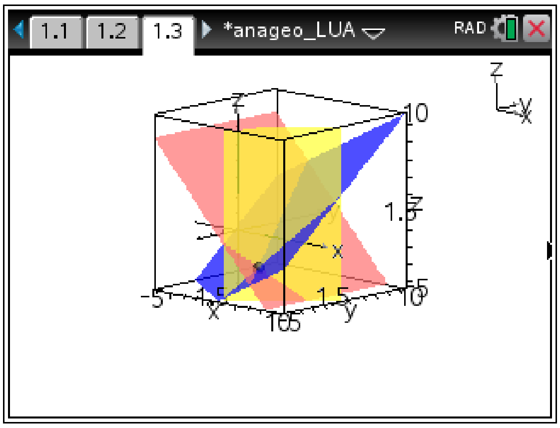
And I must admit that this was not so easy. Many special cases must be treated. Some calculations become difficult if there are vertical or horizontal planes …

Given planes are: $2x + 3y - 4z = 10$, $x - y = 8$, $x + z = 3$

This is the most common case:

For showing the intersection point I use again Genevieve Savard's great library geo3d.tns. The point is represented as a small sphere.

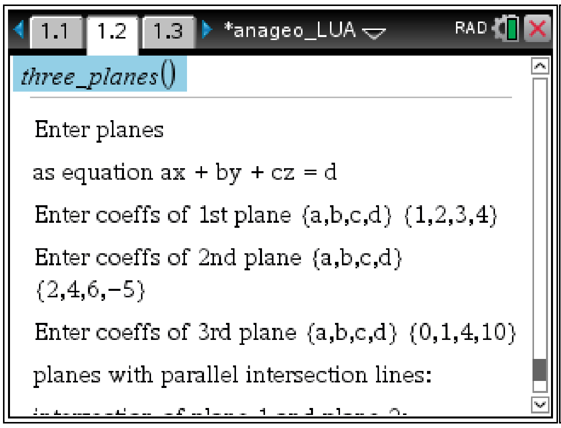


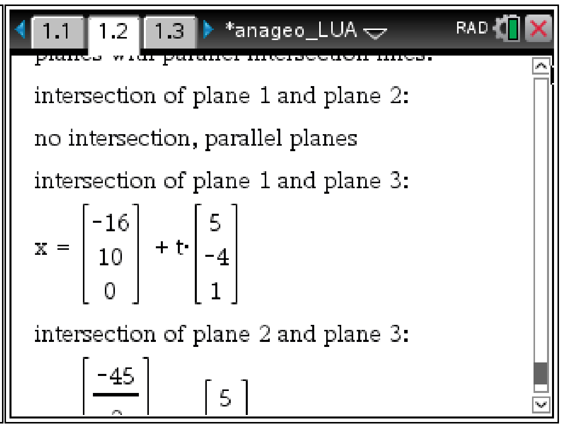


Just enter the coefficients of the planes and you will obtain the intersection point. As the equation of the second plane cannot be entered as a function of *z*, we find the respective parameter form – using *to_parm*(*coeffs*) and enter this using *u* and *t* as parameters.

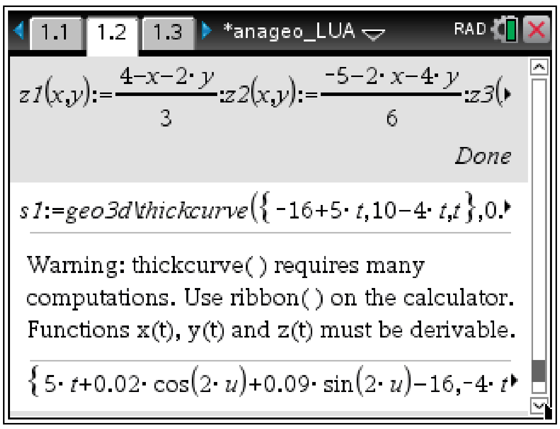
Next case: Here we have two parallel planes which are intersected by the third one.

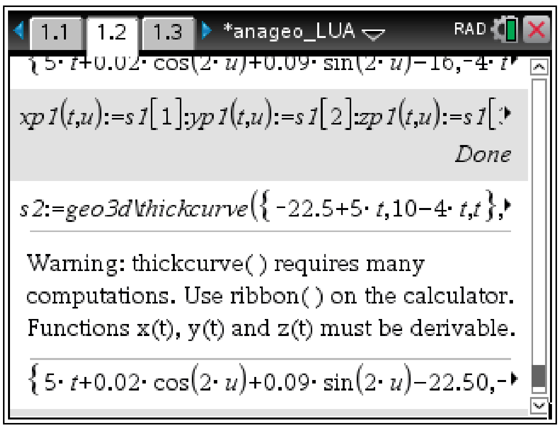
Given planes are: $x + 2y + 3z = 4$, $2x + 4y + 6z = -5$, $y + 4z = 10$





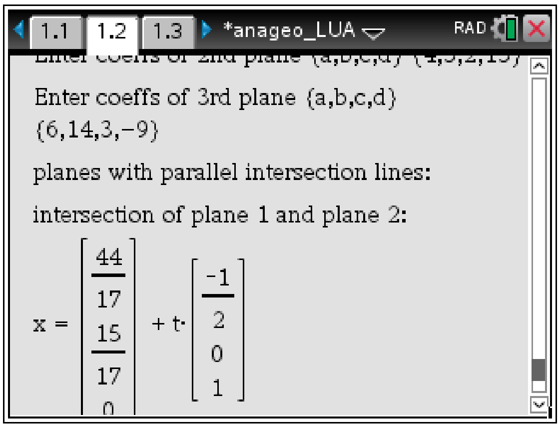
We can plot the planes together with the intersection lines ("*thickcurves*")

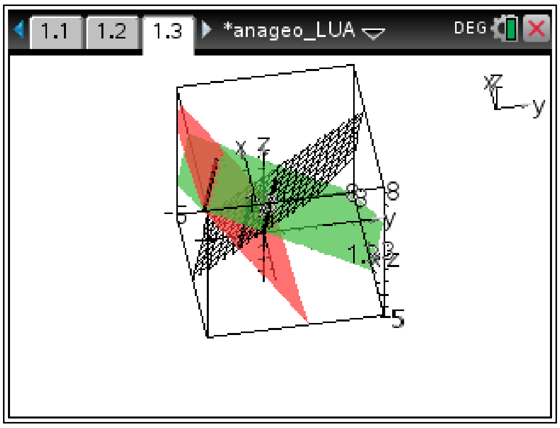




Next case will result in three parallel intersection lines:

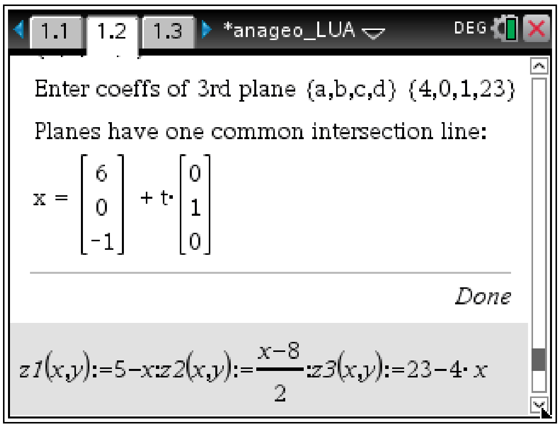
Planes: $6x - 4y + 3z = 12$, $4x + 3y + 2z = 13$, $6x + 14y + 3z = -9$

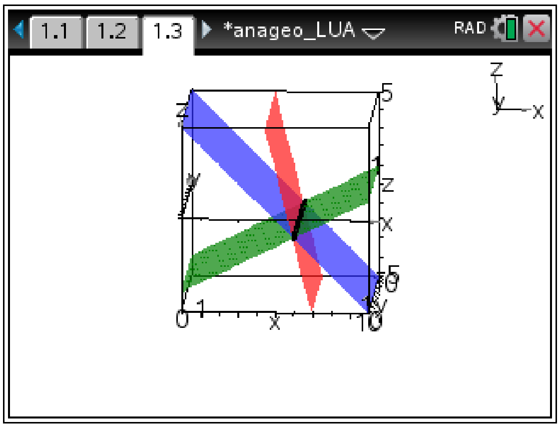




We can plot the planes and the intersection lines (again as *thickcurves*).

Planes: $x + z = 5$, $x - 2z = 8$, $4x + z = 23$

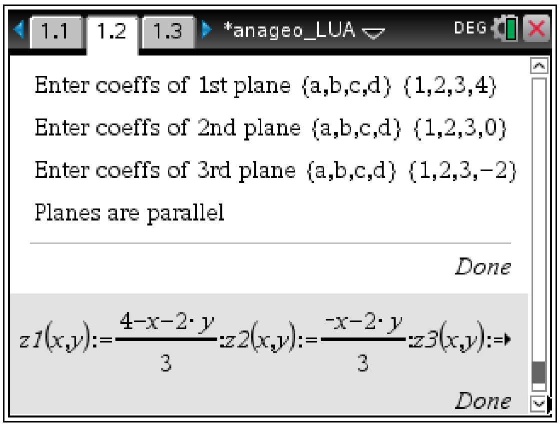


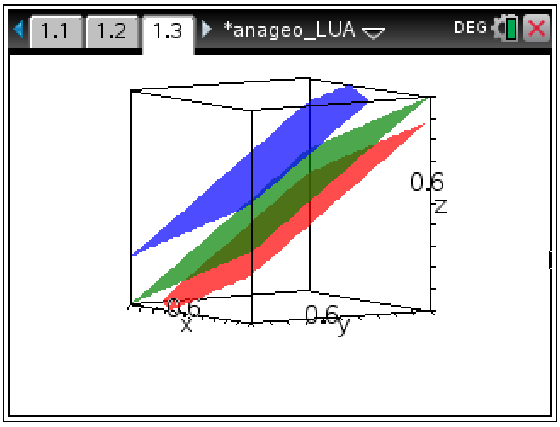


The planes have one common line.

One last case is remaining: three parallel planes.

Given planes are: $x + 2y + 3z = 4$, $x + 2y + 3z = 0$, $x + 2y + 3z = -2$





Many thanks to Geneviève Savard for 3dgeo.tns which helped illustrating the problems – and their solutions as well.