THE DERIVE - NEWSLETTER #104

ISSN 1990-7079

# THE BULLETIN OF THE DERIVE USER GROUP + CAS-TI

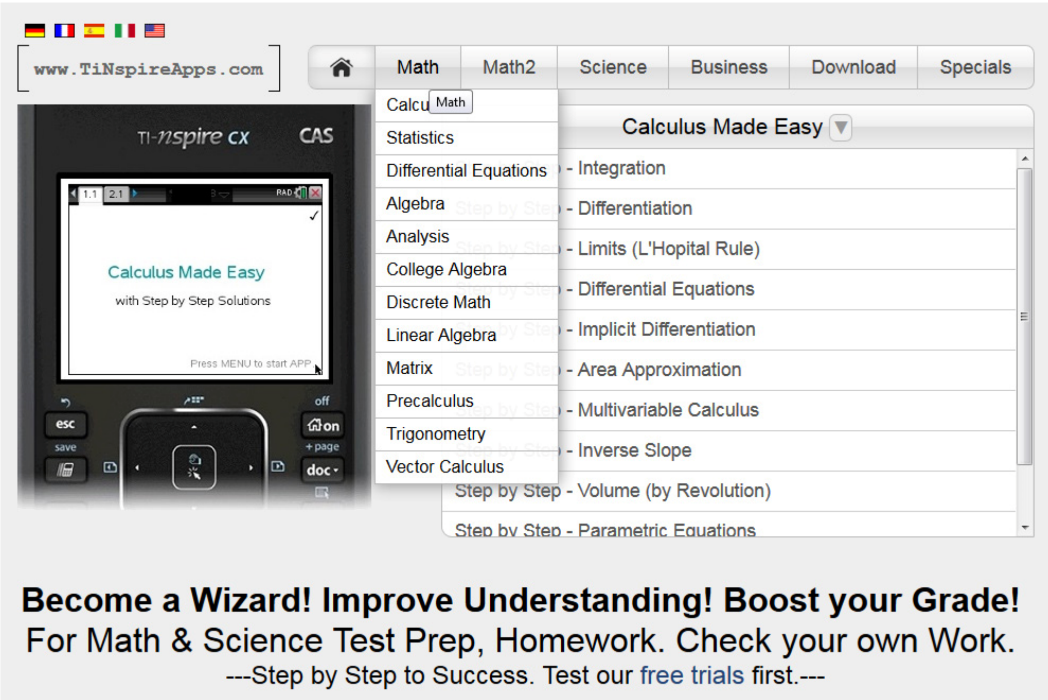
Contents:



December 2016

Information from DUG-Member Nils Hahnfeld:

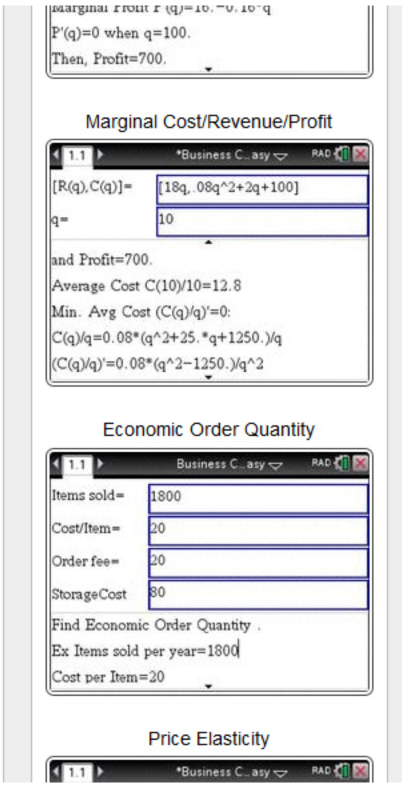
I have a lot of fun producing lua apps for TI-Nspire CX CAS. You can find a selection of them on the following website www.tinspireapps.com.




These are some features of the Business Calculus App:

**FUNCTIONALITY & MENU ITEMS OF APP**

- BUSINESS CALCULUS
- Marginal Profit
- Marginal Revenue
- Demand Analysis
- Break-Even Point
- Economic Order Qty
- Price Elasticity
- Gini Coefficient
- Simplex Algorithm
- Steady Income Valuation
- Find Equilibrium of Supply & Demand
- Consumer Surplus
- Producer Surplus
- Cobb-Douglas Prod.
- Future Value & Compound Interest
- Effective Interest Rate
- Differentials
- y=f'(x)* dx
- Logistic Differential Equation



Dear DUG Members,

Back home from an exciting travel through Ecuador including the Galapagos Islands (strolling on Charles Darwin's tracks, see some pictures) I can finish DNL#104.

Please notice the information about Nils Hahnfeld's rich collection of Apps for the TI-Nspire. He discovered the world of LUA for his programs. I like LUA, too and am busy to transfer a Flemish booklet on LUA-programming together with Steve Arnold's LUA-tutorials into German. This will become a more than 80 pages' script. I'll keep you informed.

I have the pleasure to welcome Jonny Griffiths as a new contributor for the DNL. I am happy that his article is on secondary school level, so that the teachers among you might benefit, many thanks Jonny. Roland Schröder's short article (page 3) could also serve for an interesting lesson in school.

Roland Schröder treats a very old and famous problem – the magic squares. The interesting part was programming the Magic Squares. Among the many Internet resources I found the 4k+2 MSq which inspired me trying to program these kind of MSqs, too.

Late Pat Leinbach and her husband, our old friend, Carl Leinbach demonstrate how GPS is working and show how to design problems for the class room.

I am a regular reader of the German weekly newspaper "*Die Zeit*", which provides every week a so called "*Logelei*" (Little Lie). You can try to solve the problems by thinking only or – as I am doing here – by using mathematical means – or as demonstrated in the third example by a combination of thinking and using mathematics & technology. This makes really great fun.

Finally, I forward a request which I received just recently. Do you know how to get a "History of DERIVE"? It would be great if somebody of our community could provide some details, many thanks in advance.

Best regards as ever

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

http://www.austromath.at/dug/

Enchanted woods and happy sea lions in Ecuador

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 3136365
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: March 2017

**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
Tools for 3D-Problems, P. Lüke-Rosendahl, GER
Simulating a Graphing Calculator in *DERIVE*, J. Böhm, AUT
Graphics World, Currency Change, P. Charland, CAN
Cubics, Quartics – Interesting features, T. Koller & J. Böhm, AUT
Logos of Companies as an Inspiration for Math Teaching
Exciting Surfaces in the FAZ / Pierre Charland´s Graphics Gallery
BooleanPlots.mth, P. Schofield, UK
Old traditional examples for a CAS – What´s new? J. Böhm, AUT
Mandelbrot and Newton with *DERIVE*, Roman Hašek, CZK
Tutorials for the NSpireCAS, G. Herweyers, BEL
Some Projects with Students, R. Schröder, GER
Dirac Algebra, Clifford Algebra, D. R. Lunsford, USA
A New Approach to Taylor Series, D. Oertel, GER
Henon & Co; Find your very own Strange Attractor, J. Böhm, AUT
Rational Hooks, J. Lechner, AUT
Statistics of Shuffling Cards, H. Ludwig, GER
Charge in a Magnetic Field, H. Ludwig, GER
Factoring Trinomials, D. McDougall, CAN
Selected Lectures from TIME 2016

and others

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

Roland Schröder sent this very nice short contribution:

## Four Functions with special attributes

Given are four functions $f_1$, $f_2$, $f_3$, $f_4$ with equations

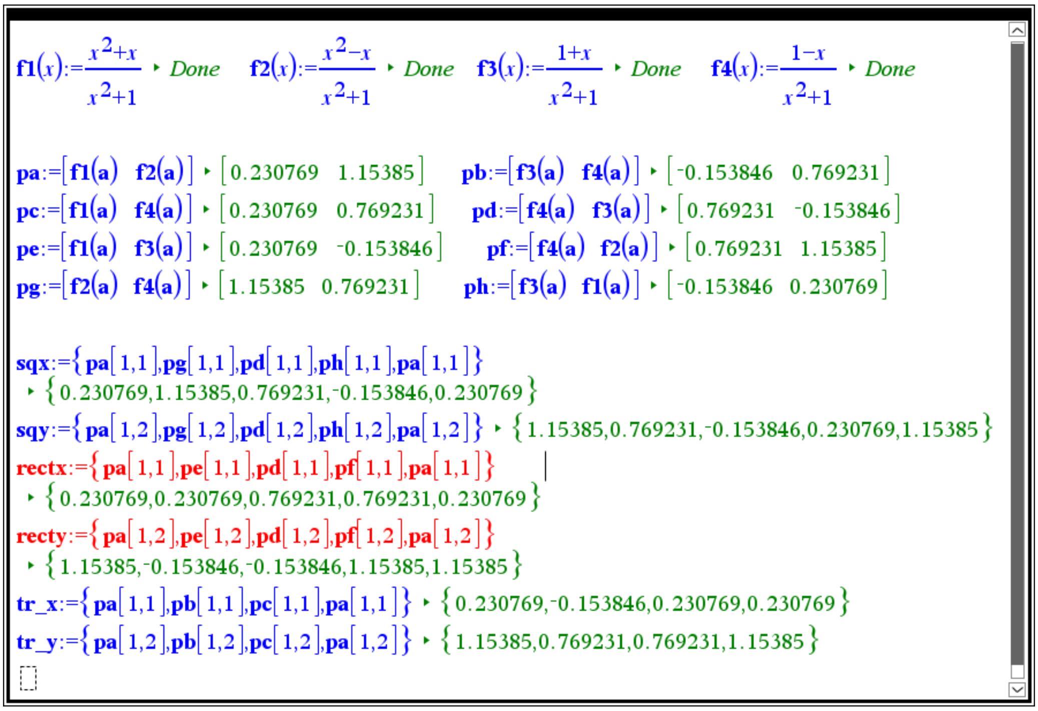
$$f_1(x)=\frac{x^2+x}{x^2+1},\ f_2(x)=\frac{x^2-x}{x^2+1},\ f_3(x)=\frac{1+x}{x^2+1},\ f_4(x)=\frac{1-x}{x^2+1}.$$

We define eight points depending on parameter $a$ as follows:

A($f_1(a)$, $f_2(a)$), B($f_3(a)$, $f_4(a)$), C($f_1(a)$, $f_4(a)$), D($f_4(a)$, $f_3(a)$), E($f_1(a)$, $f_3(a)$), F($f_4(a)$, $f_2(a)$), G($f_2(a)$, $f_4(a)$) and H($f_3(a)$,$f_1(a)$).

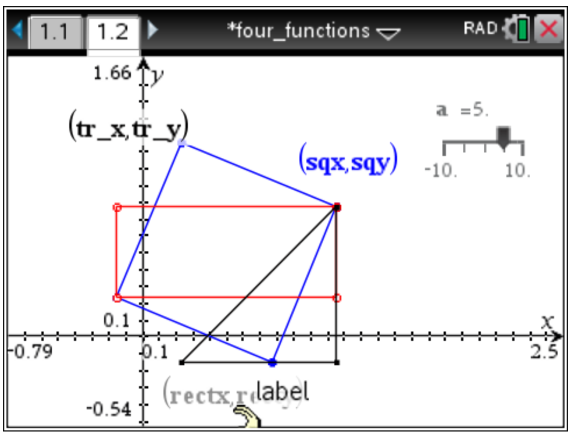
- Points AGDH form a unit square
- Points AEDF form a rectangle with sides parallel to the axes
- Points ABC form an isosceles right triangle

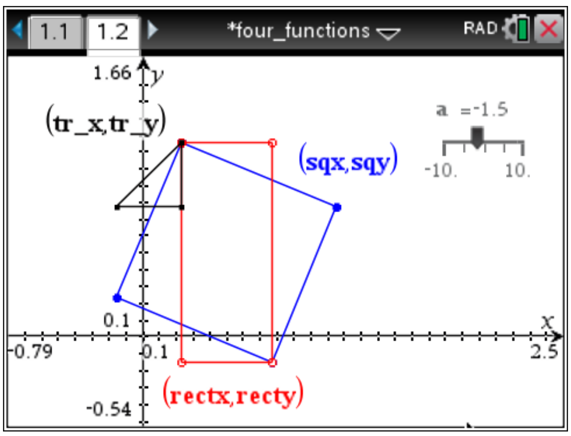
All definitions of the functions and the lists of coordinates (for scatter plots) are made in a notes page.



```
f1(x):=(x^2+x)/(x^2+1) ▸ Done   f2(x):=(x^2-x)/(x^2+1) ▸ Done   f3(x):=(1+x)/(x^2+1) ▸ Done   f4(x):=(1-x)/(x^2+1) ▸ Done

pa:=[f1(a) f2(a)] ▸ [0.230769 1.15385]      pb:=[f3(a) f4(a)] ▸ [-0.153846 0.769231]
pc:=[f1(a) f4(a)] ▸ [0.230769 0.769231]     pd:=[f4(a) f3(a)] ▸ [0.769231 -0.153846]
pe:=[f1(a) f3(a)] ▸ [0.230769 -0.153846]    pf:=[f4(a) f2(a)] ▸ [0.769231 1.15385]
pg:=[f2(a) f4(a)] ▸ [1.15385 0.769231]      ph:=[f3(a) f1(a)] ▸ [-0.153846 0.230769]

sqx:={pa[1,1],pg[1,1],pd[1,1],ph[1,1],pa[1,1]}
 ▸ {0.230769,1.15385,0.769231,-0.153846,0.230769}
sqy:={pa[1,2],pg[1,2],pd[1,2],ph[1,2],pa[1,2]} ▸ {1.15385,0.769231,-0.153846,0.230769,1.15385}
rectx:={pa[1,1],pe[1,1],pd[1,1],pf[1,1],pa[1,1]}
 ▸ {0.230769,0.230769,0.769231,0.769231,0.230769}
recty:={pa[1,2],pe[1,2],pd[1,2],pf[1,2],pa[1,2]}
 ▸ {1.15385,-0.153846,-0.153846,1.15385,1.15385}
tr_x:={pa[1,1],pb[1,1],pc[1,1],pa[1,1]} ▸ {0.230769,-0.153846,0.230769,0.230769}
tr_y:={pa[1,2],pb[1,2],pc[1,2],pa[1,2]} ▸ {1.15385,0.769231,0.769231,1.15385}
```

Three scatter plots give the figures. (I cannot use trx and try because try is a TI-Nspire command.)



The proofs are performed with DERIVE:

#1: CaseMode := Sensitive

#2: $\left[f1 := \dfrac{x^2 + x}{x^2 + 1},\ f2 := \dfrac{x^2 - x}{x^2 + 1}\right]$

#3: $\left[f3 := \dfrac{1 + x}{x^2 + 1},\ f4 := \dfrac{1 - x}{x^2 + 1}\right]$

#4: [A := [f1, f2], B := [f3, f4], C := [f1, f4], D := [f4, f3]]

#5: [E := [f1, f3], F := [f4, f2], G := [f2, f4], H := [f3, f1]]

#6: AGDH is a square!

#7: $[(G - A)\cdot(D - G),\ (D - G)\cdot(H - D)] = [0, 0]$

#8: $[(H - D)\cdot(A - H),\ (A - H)\cdot(G - A)] = [0, 0]$

#9: $[|G - A|,\ |D - G|,\ |H - D|,\ |A - H|] = [1, 1, 1, 1]$

#10: AEDF is a rectangle with edges parallel to the axes!

#11: $[(E - A)\cdot(D - E),\ (D - E)\cdot(F - D)] = [0, 0]$

#12: $[(F - D)\cdot(A - F),\ (A - F)\cdot(E - A)] = [0, 0]$

#13: $[|E - A|,\ |F - D|] = \left[\dfrac{|x^2 - 2\cdot x - 1|}{x^2 + 1},\ \dfrac{|x^2 - 2\cdot x - 1|}{x^2 + 1}\right]$

#14: $[|D - E|,\ |A - F|] = \left[\dfrac{|x^2 + 2\cdot x - 1|}{x^2 + 1},\ \dfrac{|x^2 + 2\cdot x - 1|}{x^2 + 1}\right]$

#15: $[A, E, F, D] = \begin{bmatrix} \dfrac{x\cdot(x + 1)}{x^2 + 1} & \dfrac{x\cdot(x - 1)}{x^2 + 1} \\ \dfrac{x\cdot(x + 1)}{x^2 + 1} & \dfrac{x + 1}{x^2 + 1} \\ \dfrac{1 - x}{x^2 + 1} & \dfrac{x\cdot(x - 1)}{x^2 + 1} \\ \dfrac{1 - x}{x^2 + 1} & \dfrac{x + 1}{x^2 + 1} \end{bmatrix}$

#16: ABC is a right triangle!

#17: $[|B - A|,\ |C - B|,\ |C - A|] = \left[\dfrac{\sqrt{2}\cdot|x^2 - 1|}{x^2 + 1},\ \dfrac{|(x + 1)\cdot(x - 1)|}{x^2 + 1},\ \dfrac{|(x + 1)\cdot(x - 1)|}{x^2 + 1}\right]$

#18: $(C - B)\cdot(C - A) = 0$

#19: $|B - A|^2 - 2\cdot|C - B|^2 = 0$

Mail from Jonny Griffiths:

Dear Josef,

I hope you are well; I am good this end, and enjoying Derive as always.

I attach a short article, that might fit into a Newsletter. The level is low, making the article accessible to everyone I hope.

With best wishes,

Jonny

Dear Jonny,

Many thanks for your article. It would be great if I would receive more "low level" contributions. Our members – and I, of course – appreciate these articles because they can be used in class room. … and the level is not so low …

Best regards and once more many thanks.

If you have more such "low level" papers, then …

Josef

Dear Josef,

Glad you like the article, and yes, if I have any similar ideas I'll happily send them through,

Best wishes,

Jonny

# Using Derive to extend an Exam Question

## Jonny Griffiths, Norwich, UK

This problem sparked my curiosity into action recently. Take the degree *n* polynomial equation

$$x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 = 0$$

This could have *n* real roots. Is it possible for those roots to be $a_{n-1}, \ldots a_1, a_0$?

Pencil and paper is fine for *n* = 1 ($x + a = 0$ cannot have the root $x = a$, unless $a = 0$). How about *n* = 2? We have

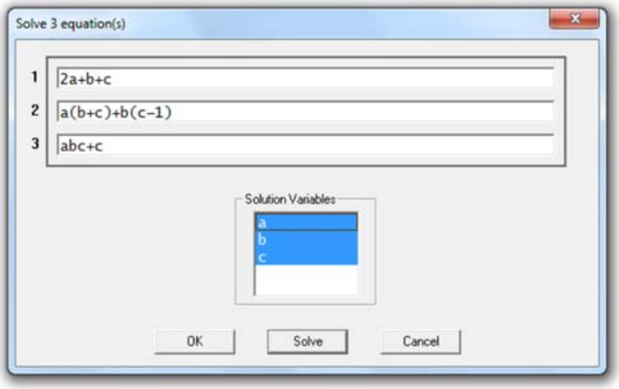
$$x^2 + a\,x + b = (x - a)(x - b) \Longrightarrow -a - b = a, b = b\,a,$$

which solve to give us the two solutions $(a,b) = (0,0)$ or $(1,-2)$. The second of these is clearly more interesting; when, we might ask, do we get solutions that are all non-zero?

When it comes to *n* = 3, I am already thinking Derive will help significantly. We need $x^3 + a\,x^2 + bx + c - (x - a)(x - b)(x - c)$ to be zero for all values of *x*, so the plan is to expand this into powers of *x*, and then equate each coefficient to zero. Not TOO demanding by hand, but we are thinking ahead. Derive gives us this:

$$x^2(2a+b+c)-x(a(b+c)+b(c-1))+abc+c$$

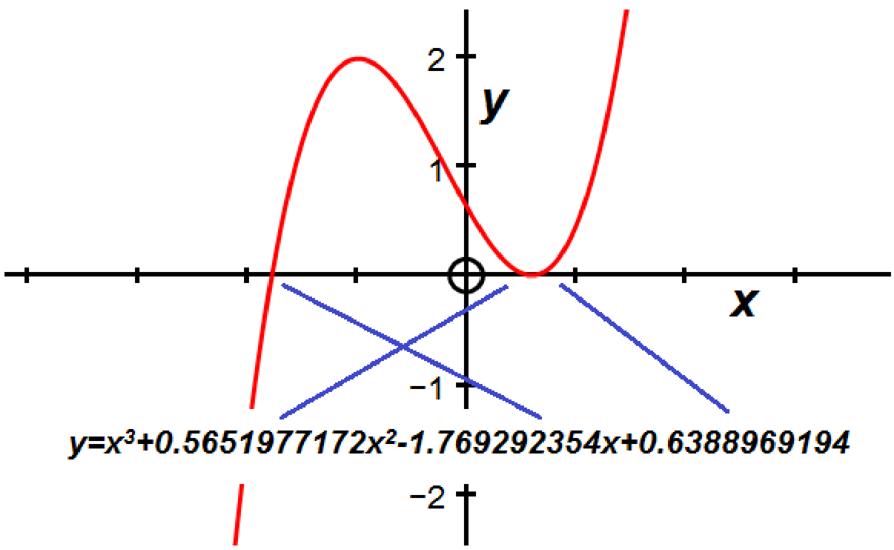
If we turn to the equation-solving facility in Derive, we see it will solve systems of equations for us.



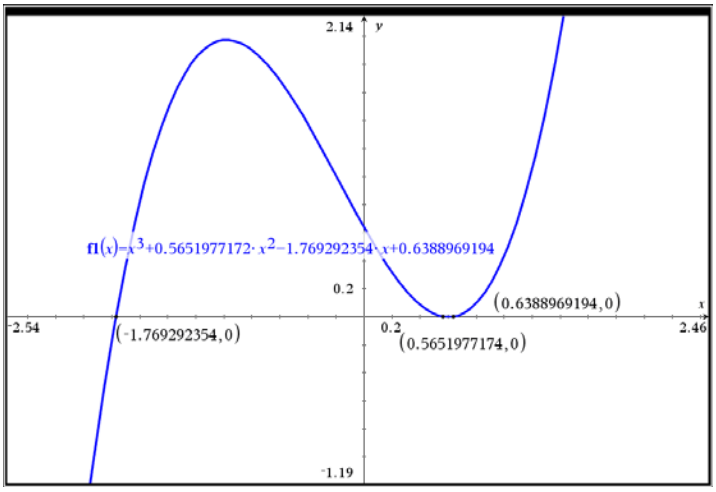

```
SOLUTIONS([2·a + b + c, a·(b + c) + b·(c - 1), a·b·c + c], [a, b, c], Real)
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & -1 \\ 1 & -2 & 0 \\ 0.5651977173 & -1.769292354 & 0.6388969194 \end{bmatrix}$$

Inserting our three equations and solving, we get six solutions for $(a,b,c)$:
(0,0,0), (1,–1,–1), (1,–2,0),(0.5651977173, –1.769292354, 0.6388969194),
together with two complex solutions. So this last possibility gives us something a little offbeat and unexpected that we can draw in a graph plotter (my such program of choice is Autograph).




When we plot the cubic in a TI-Nspire Graphs-Window we can easily verifiy the zeros by analyzing the graph (Josef):




We might note that the first and third solutions are those we had for $n = 2$ with a zero added, and a moment's thought will tell us this is as expected. If we have a solution for degree $n$, then multiplying this equation by $x$ will give us a solution for degree $n + 1$ that is our degree $n$ solution with a 0 added. So our solutions accumulate; there will always be at least as many degree $n + 1$ solutions as degree $n$ solutions.

This is where I tell myself to be careful with all this computing power. Might it sometimes encourage mental laziness? Leaping onto the computer is not always the best option, and there may be some simple reason why a conjecture is right or wrong that a little old-fashioned reflection will reveal immediately.

The above is feasible by hand, but the $n = 4$ case I don't think it is. Here we start to push Derive and the computer harder. On expanding

$$x^4 + ax^3 + bx^2 + cx + d - (x - a)(x - b)(x - c)(x - d),$$

we find four equations that we wish to solve together. I don't think that the Derive system-solver can take us the whole way now. We effectively wish for

$$[2a+b+c+d,\ a(b+c+d)+b(c+d-1)+cd,\ a(b(c+d)+cd)+bcd+c,\ -abcd+d) = [0,0,0,0].$$

The first element gives us $a = -\dfrac{b+c+d}{2}$ which on substituting back leads to

$$\left[0, -\frac{b^2+2b+c^2+d^2}{2}, -\frac{b^2(c+d)+b(c^2+cd+d^2)+c(cd+d^2-2)}{2}, \frac{d\left(b^2c+bc(c+d)+2\right)}{2}\right].$$

The last element gives us either $d = 0$ (which leads to the solution we already have) or $d = -\dfrac{b^2c+bc^2+2}{bc}$. Substituting this in gives

$$[0, -\frac{f(b,c)}{b^2c^2}, -\frac{g(b,c)}{b^2c^2}, 0]$$

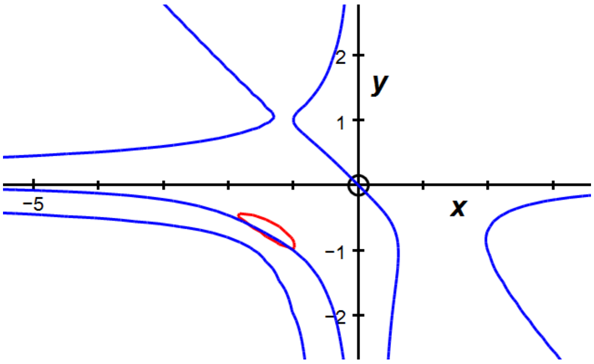
where

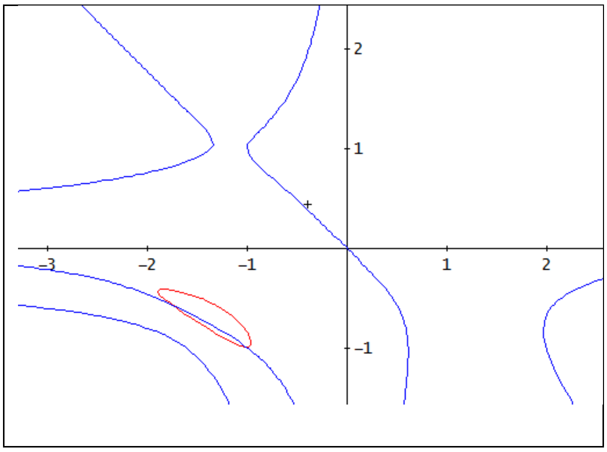
$$f(b,c) = b^4c^2 + b^3c^2(c+1) + b^2c(c^3+2) + 2bc^2 + 2$$

and

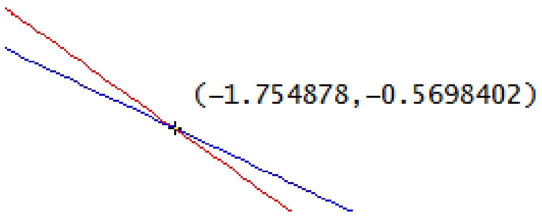
$$g(b,c) = b^4c^3 + b^3c(c^3+1) + b^2c^2(3-c) + b(c^3+2) + 2c.$$

Solving $f(b,c) = 0$ and $g(b,c) = 0$ together looks horrific, and certainly Derive refuses to do this. But, do we have to give up just yet? Autograph is excellent at plotting implicitly defined functions. What happens if we plot $f(x,y) = 0$ and $g(x,y) = 0$, and try to find where they meet?




Let's try with DERIVE and zoom in:

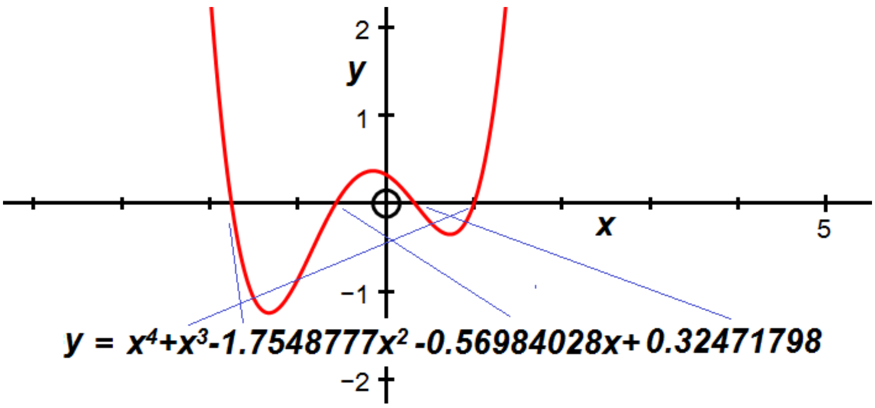



It is not possible – at the moment – to perform implicit plots with TI-NspireCAS (Josef).

These are the kind of offbeat curves that students love. Yes, they do intersect, at (–1,–1), which gives us a solution we have already, but also in the quirky point we seek, which we can find by zooming in;

(1, –1.7548777…, –0.56984028…, 0.32471798…)

All the roots/coefficients here are non-zero; Autograph gives us this (-and DERIVE does as well).



The attempt at $n = 5$ leads to three difficult polynomial equations to solve, each in three variables. We can treat these as surfaces in three dimensions, and try to find where the surfaces meet. Autograph draws even these strange animals, but reveals, to my eye at least, no intersection point. And so the investigation comes to an end, or at least a pause.

Those I've shared this problem with have liked it. It seems a natural one for a mathematician to ask. Yet $n = 3$ case is served by Derive, and the $n = 4$ and $n = 5$ cases are completely impossible without it. Sometimes using computing power is portrayed as 'cheating'; in what way? I would say here that to turn to Derive as a tool in your problem-solving bag is like turning to the mental screwdriver in my head that is the formula for the quadratic equation. Is that 'cheating'?

Maybe one reason I like using computing power is because it's so relatively recent. Derive was launched in 1988, that's 28 years ago, a minute time compared with how long humankind has been doing mathematics, It's possible I'm the first person to see the coefficients for the $n = 4$ case, in which case I could have just bought a tiny piece of immortality(!)

Jonny Griffiths, hello@jonny-griffiths.net, Sepember 2016

Dear Jonny,

as you can see in the attached DERIVE file, DERIVE is able to solve the 4 variables equation within an instant. Besides this I like the graphic solution using the two implicit plots.

I know Autograph – and I know Douglas Butler very well. I produced the German manual for Autograph.

Do you mind if I add the respective DERIVE and/or Nspire plots to your fine contribution?

Best regards
Josef

```
#1:   qu := x^4 + a·x^3 + b·x^2 + c·x + d - (x - a)·(x - b)·(x - c)·(x - d)

#2:   coeff := VECTOR(POLY_COEFF(qu, x, n), n, 3, 0, -1)

#3:   coeff := [2·a + b + c + d, - a·(b + c + d) - b·(c + d - 1) - c·d,
        a·(b·(c + d) + c·d) + b·c·d + c, d - a·b·c·d]

#4:   SOLUTIONS(coeff = [0, 0, 0, 0], [a, b, c, d], Real)
```

#5:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 \\ 1 & -2 & 0 & 0 \\ 1 & -1.754877666 & -0.5698402909 & 0.3247179572 \\ 0.5651977173 & -1.769292354 & 0.6388969194 & 0 \end{bmatrix}$$

Hi Josef,

This made me laugh! Derive is always more powerful than you think!

# Magic Squares

Roland Schröder

A “*Magic Square*” of order $n$ is an arrangement of integers from 1 to $n^2$ in a matrix of $n$ rows and $n$ columns such that the sums of all rows, all columns and the main diagonals are equal. This sum is called “*Magic Number*” of the magic square.

It is easy to find the magic numbers of $n$ (with $n > 2$) order MSqs:

$$\frac{\sum_{k=1}^{k^2} k}{k} = \frac{k\cdot(k^2+1)}{2}$$

$$\text{VECTOR}\left(\frac{k\cdot(k^2+1)}{2}, k, 1, 10\right) = [1, 5, 15, 34, 65, 111, 175, 260, 369, 505]$$

## One Hand & Head Method

We start with a square of the following sums:

| | | | |
|---|---|---|---|
| t+k | r+l | s+j | u+i |
| u+j | s+i | r+k | t+l |
| r+i | t+j | u+l | s+k |
| s+l | u+k | t+i | r+j |

Then replace r, s, t, u by numbers 1, 2, 3, 4 in any order and i, j, k, l by 0, 4, 8, 12 again in any order. Then the 16 sums form the magic square. In this way, we can produce 4! × 4! = 576 different order 4 MSqs by permutating the numbers. But there are more MSqs of order 4 which can be generated in another way. MSqs are called “different” if they are created by rotation or reflection from each other.

## Let’s do it with DERIVE

We start defining eight matrices:

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 1 | 0 | | 0 | 1 | 0 | 0 | | 0 | 0 | 0 | 1 |
| A:= | 0 | 1 | 0 | 0 | B:= | 0 | 0 | 0 | 1 | C:= | 0 | 1 | 0 | 0 |
| | 0 | 0 | 0 | 1 | | 0 | 0 | 1 | 0 | | 1 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 0 | | 1 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 |
| | 0 | 1 | 0 | 0 | | 0 | 0 | 0 | 1 | | 0 | 0 | 1 | 0 |
| D:= | 0 | 0 | 1 | 0 | E:= | 1 | 0 | 0 | 0 | F:= | 1 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 | | 0 | 1 | 0 | 0 |
| | 0 | 0 | 0 | 1 | | 0 | 1 | 0 | 0 | | 0 | 0 | 0 | 1 |
| | 1 | 0 | 0 | 0 | | 1 | 0 | 0 | 0 | | | | | |
| G:= | 0 | 0 | 1 | 0 | H:= | 0 | 0 | 0 | 1 | | | | | |
| | 0 | 0 | 0 | 1 | | 0 | 1 | 0 | 0 | | | | | |
| | 0 | 1 | 0 | 0 | | 0 | 0 | 1 | 0 | | | | | |

Explanation: A is the matrix of appearances of s, B of r, C of i, etc. in the sums given above.

Forming the sum of products s·A + r·D + u·E + t·H + j·F + l·B + i·C + k·G generates exactly the square from above:

#1: CaseMode := Sensitive

#2: $\left[A := \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, B := \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, C := \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\right]$

#3: $\left[D := \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, E := \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, F := \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\right]$

#4: $\left[G := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, H := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\right]$

#5: mSq := s·A + r·D + u·E + t·H + j·F + l·B + i·C + k·G

#6: $mSq := \begin{bmatrix} k + t & l + r & j + s & i + u \\ j + u & i + s & k + r & l + t \\ i + r & j + t & l + u & k + s \\ l + s & k + u & i + t & j + r \end{bmatrix}$

We follow the method by replacing the variables as given in the instruction:

#7: MSQ(r, s, t, u, i, j, k, l) := s·A + r·D + u·E + t·H + j·F + l·B + i·C + k·G

#8: $MSQ(1, 2, 3, 4, 0, 4, 8, 12) = \begin{bmatrix} 11 & 13 & 6 & 4 \\ 8 & 2 & 9 & 15 \\ 1 & 7 & 16 & 10 \\ 14 & 12 & 3 & 5 \end{bmatrix}$

#9: $MMSQ(n) := n_2 \cdot A + n_1 \cdot D + n_4 \cdot E + n_3 \cdot H + n_6 \cdot F + n_8 \cdot B + n_5 \cdot C + n_7 \cdot G$

#10: $MMSQ([1, 2, 3, 4, 0, 4, 8, 12]) = \begin{bmatrix} 11 & 13 & 6 & 4 \\ 8 & 2 & 9 & 15 \\ 1 & 7 & 16 & 10 \\ 14 & 12 & 3 & 5 \end{bmatrix}$

The magic number of these MQs is 34, check it!

The second function (a vector/list as argument) is intended for further use, because we would like to produce randomly generated MSqs. For this purpose, we can refer to a program *perm* from an earlier DNL which generates all permutations of *n* elements and then picks out one of them randomly chosen. *MQ* does it all in one single step.

Program *perm* is not printed here – it works in the background. (Find *perm* on page 33.)

```
MQ := MMSQ(APPEND((perm([1, 2, 3, 4], 4))             , (perm([0, 4, 8, 12], 4))             ))
                                         RANDOM(24) + 1                            RANDOM(24) + 1
```

See three calls of MQ followed by a list of four MQs of order 4.

$$\#13: \quad MQ = \begin{bmatrix} 2 & 5 & 12 & 15 \\ 11 & 16 & 1 & 6 \\ 13 & 10 & 7 & 4 \\ 8 & 3 & 14 & 9 \end{bmatrix}$$

$$\#14: \quad MQ = \begin{bmatrix} 7 & 1 & 10 & 16 \\ 12 & 14 & 5 & 3 \\ 13 & 11 & 4 & 6 \\ 2 & 8 & 15 & 9 \end{bmatrix}$$

$$\#15: \quad MQ = \begin{bmatrix} 1 & 15 & 8 & 10 \\ 6 & 12 & 3 & 13 \\ 11 & 5 & 14 & 4 \\ 16 & 2 & 9 & 7 \end{bmatrix}$$

#16: VECTOR(MQ, k, 4)

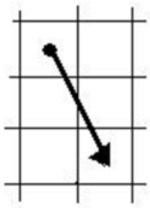
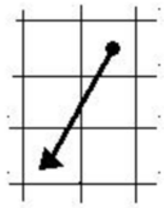
$$\#17: \quad \left[ \begin{bmatrix} 11 & 5 & 4 & 14 \\ 2 & 16 & 9 & 7 \\ 13 & 3 & 6 & 12 \\ 8 & 10 & 15 & 1 \end{bmatrix}, \begin{bmatrix} 6 & 4 & 9 & 15 \\ 11 & 13 & 8 & 2 \\ 16 & 10 & 3 & 5 \\ 1 & 7 & 14 & 12 \end{bmatrix}, \begin{bmatrix} 10 & 7 & 4 & 13 \\ 1 & 16 & 11 & 6 \\ 15 & 2 & 5 & 12 \\ 8 & 9 & 14 & 3 \end{bmatrix}, \begin{bmatrix} 13 & 3 & 12 & 6 \\ 10 & 8 & 15 & 1 \\ 7 & 9 & 2 & 16 \\ 4 & 14 & 5 & 11 \end{bmatrix} \right]$$

The eight matrices A – H are called *base of the magic square*. The art of generating MSqs is finding bases for various orders. I don't know a way which can be generally used for MSqs of even order.

The famous Dürer Square *Melencolia* cannot be produced in this way.

## Bases for Magic Squares of odd order

There exists a procedure to create the bases of odd order MSqs. All what we need are two knight's moves known from the chess game:

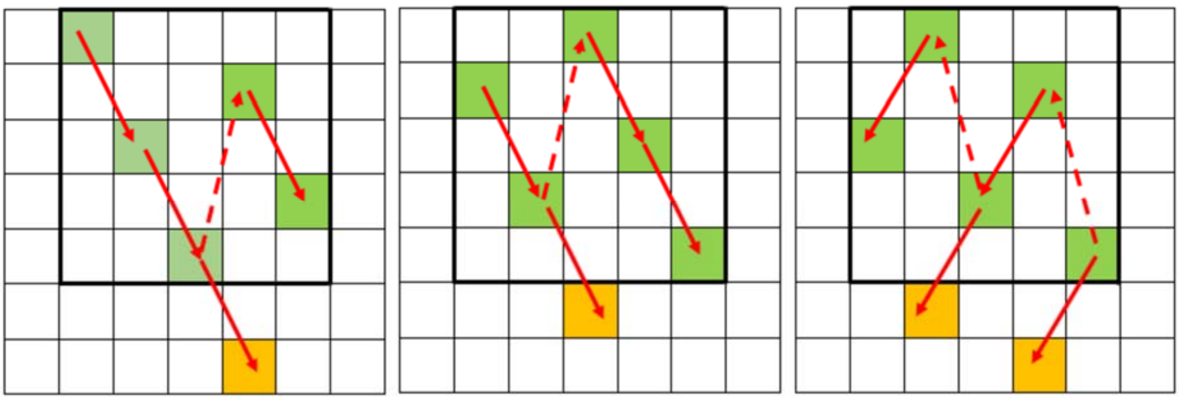
We start by entering a 1 in any empty cell and then proceed in knight's moves of same kind (say the left one) and again enters a 1 in all passed cells.

It can happen that the knight will jump into a cell outside of the square, then you will find a cell inside of the square carrying the same letter. A base matrix is ready when you come back to an occupied cell. (Each row and each column contains exact one cell "1").

For generating a system of base matrices (e.g. of order 5) start in cells a, f, k, l, and m and apply the knight's move left above. For the next five bases start in e, j, n, o, and p and apply the right moves.

| | | | | | | |
|---|---|---|---|---|---|---|
| e | a | b | c | d | e | |
| j | f | g | h | i | j | |
| n | k | | | q | n | k |
| o | l | | | r | o | l |
| p | m | | | s | p | m |
| e | a | b | c | d | e | a |
| j | f | g | h | i | j | f |

The pictures below may illustrate the bases which started in a, f and p.



So we get a set of 10 matrices (named according to the initial cells):

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 0 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 | 0 | | 0 | 0 | 0 | 0 | 1 |
| | 0 | 0 | 0 | 1 | 0 | | 1 | 0 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 | 0 |
| A:= | 0 | 1 | 0 | 0 | 0 | F:= | 0 | 0 | 0 | 1 | 0 | K:= | 1 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 1 | | 0 | 1 | 0 | 0 | 0 | | 0 | 0 | 0 | 1 | 0 |
| | 0 | 0 | 1 | 0 | 0 | | 0 | 0 | 0 | 0 | 1 | | 0 | 1 | 0 | 0 | 0 |
| | 0 | 1 | 0 | 0 | 0 | | 0 | 0 | 0 | 1 | 0 | | | | | | |
| | 0 | 0 | 0 | 0 | 1 | | 0 | 1 | 0 | 0 | 0 | | | | | | |
| L:= | 0 | 0 | 1 | 0 | 0 | M:= | 0 | 0 | 0 | 0 | 1 | | | | | | |
| | 1 | 0 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 | 0 | | | | | | |
| | 0 | 0 | 0 | 1 | 0 | | 1 | 0 | 0 | 0 | 0 | | | | | | |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 1 | | 0 | 0 | 1 | 0 | 0 | | 1 | 0 | 0 | 0 | 0 |
| | 0 | 1 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 | 1 | | 0 | 0 | 1 | 0 | 0 |
| E:= | 0 | 0 | 0 | 1 | 0 | J:= | 0 | 1 | 0 | 0 | 0 | N:= | 0 | 0 | 0 | 0 | 1 |
| | 1 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 1 | 0 | | 0 | 1 | 0 | 0 | 0 |
| | 0 | 0 | 1 | 0 | 0 | | 1 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 1 | 0 |
| | 0 | 0 | 0 | 1 | 0 | | 0 | 1 | 0 | 0 | 0 | | | | | | |
| | 1 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 1 | 0 | | | | | | |
| O:= | 0 | 0 | 1 | 0 | 0 | P:= | 1 | 0 | 0 | 0 | 0 | | | | | | |
| | 0 | 0 | 0 | 0 | 1 | | 0 | 0 | 1 | 0 | 0 | | | | | | |
| | 0 | 1 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 | 1 | | | | | | |

The sum A·q + F·r + K·s + L·t + M·u + E·v + J·w + N·x + O·y + P·z gives a MSq of order 5 when we replace q, r, s, t, u by the numbers 1, 2, 3, 4, 5 in any order and v, w, x, y, z by the numbers 0, 5, 10, 15, 20 also in any order.

This is Roland's paper. He didn't include the CAS-procedures for producing the final MSqs. So I took the chance to work with DERIVE again. See how I – Josef – did:

I didn't want to edit ten matrices, so I produced a 10-rows matrix with the column numbers of the ones in the five rows of each matrix. A short VECTOR command generates the whole system *sys* of the ten base matrices (which I don't reprint). Then we build the sums – and ready-made is our MSq.

#18: $$\text{mats} := \begin{bmatrix} 1 & 4 & 2 & 5 & 3 \\ 3 & 1 & 4 & 2 & 5 \\ 5 & 3 & 1 & 4 & 2 \\ 2 & 5 & 3 & 1 & 4 \\ 4 & 2 & 5 & 3 & 1 \\ 5 & 2 & 4 & 1 & 3 \\ 3 & 5 & 2 & 4 & 1 \\ 1 & 3 & 5 & 2 & 4 \\ 4 & 1 & 3 & 5 & 2 \\ 2 & 4 & 1 & 3 & 5 \end{bmatrix}$$

#19: $\text{mat}(v) := \text{VECTOR}(\text{VECTOR}(\text{IF}(v_k = i, 1, 0), i, 5), k, 5)$

#20: $\text{sys} := \text{VECTOR}(\text{mat}(v), v, \text{mats})$

#22: $\text{nums} := [1, 2, 3, 4, 5, 0, 5, 10, 15, 20]$

#23: $$\sum_{i=1}^{10} \text{sys}_i \cdot \text{nums}_i = \begin{bmatrix} 11 & 24 & 7 & 20 & 3 \\ 17 & 5 & 13 & 21 & 9 \\ 23 & 6 & 19 & 2 & 15 \\ 4 & 12 & 25 & 8 & 16 \\ 10 & 18 & 1 & 14 & 22 \end{bmatrix}$$

Now let's again use the random generated permutation of the numbers and find other MSqs:

```
       randperm := APPEND((perm([1, 2, 3, 4, 5], 5))                        ,
#24:                                                     RANDOM(120) + 1

        (perm([0, 5, 10, 15, 20], 5))                   )
                                          RANDOM(120) + 1

#25:   randperm = [5, 2, 4, 3, 1, 20, 15, 0, 10, 5]
```

#26: $\sum_{i=1}^{10} sys_i \cdot [5, 2, 4, 3, 1, 20, 15, 0, 10, 5]_i = \begin{bmatrix} 5 & 8 & 17 & 11 & 24 \\ 12 & 21 & 4 & 10 & 18 \\ 9 & 20 & 13 & 22 & 1 \\ 23 & 2 & 6 & 19 & 15 \\ 16 & 14 & 25 & 3 & 7 \end{bmatrix}$

I was not really happy with my random permutation because it first finds all permutations internally and then chooses one at random – too much work. (In the earlier DNL all permutations were needed). The next small function creates only one random permutation of the given elements. Then we can compose a function MQ5(sys) to generate a lot of MSqs.

```
Erzeugung einer einzelnen Zufallspermutation

       rndperm(v, p, k) :=
         Prog
           p := []
           Loop
#31:         If DIM(v) = 1
                RETURN APPEND(p, v)
             k := RANDOM(DIM(v)) + 1
             p := APPEND(p, [v↓k])
             v := DELETE(v, k)

#32:   rndperm(nums, 10) = [4, 5, 2, 0, 20, 15, 5, 10, 3, 1]
```

#33: $\sum_{i=1}^{10} sys_i \cdot [4, 5, 2, 0, 20, 15, 5, 10, 3, 1]_i = \begin{bmatrix} 14 & 1 & 10 & 23 & 17 \\ 8 & 35 & 12 & 5 & 5 \\ 3 & 9 & 3 & 20 & 30 \\ 15 & 15 & 21 & 7 & 7 \\ 25 & 5 & 19 & 10 & 6 \end{bmatrix}$

```
       MQ5(s_, v_) :=
         Prog
#34:       v_ := rndperm([1, 2, 3, 4, 5, 0, 5, 10, 15, 20], 10)
           Σ(s_↓i·v_↓i, i, 1, 10)

#35:   VECTOR(MQ5(sys), i, 3)
```

#36: $\left[\begin{bmatrix} 5 & 12 & 16 & 8 & 24 \\ 18 & 25 & 9 & 2 & 11 \\ 6 & 1 & 13 & 35 & 10 \\ 30 & 20 & 7 & 5 & 3 \\ 6 & 7 & 20 & 15 & 17 \end{bmatrix}, \begin{bmatrix} 7 & 7 & 6 & 15 & 30 \\ 1 & 35 & 14 & 8 & 7 \\ 15 & 8 & 2 & 21 & 19 \\ 22 & 5 & 20 & 15 & 3 \\ 20 & 10 & 23 & 6 & 6 \end{bmatrix}, \begin{bmatrix} 25 & 15 & 6 & 8 & 11 \\ 7 & 15 & 6 & 35 & 2 \\ 16 & 22 & 3 & 14 & 10 \\ 10 & 9 & 20 & 3 & 23 \\ 7 & 4 & 30 & 5 & 19 \end{bmatrix}\right]$

You can find numerous papers on Magic Squares on the web. I did some research and will present one nice example – especially from the point of view how to treat it with DERIVE.

On http://www.mathe.tu-freiberg.de/~hebisch/cafe/magisch.html a recipe to construct MSqs of order 4*k* + 2 is presented.

We divide the square in four subsquares of order 2*k* + 1and start with the square left above. We set *k* black marks in the first row followed by a green one and a red one. The remaining cells remain white (or empty). Then we fill this square such that in direction of the main diagonal same colors will appear as depicted.

The black cells are reflected in all directions, the red ones are reflected to the right and the green ones downwards. Now the whole square is filled with colors (see the 10 × 10 square).

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | | | 0 | | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | |
| 0 | | 0 | 0 | 0 | | 0 | 0 | | 0 |
| 0 | 0 | | 0 | 0 | 0 | 0 | | 0 | |
| 0 | 0 | 0 | | 0 | 0 | | 0 | | 0 |
| 0 | 0 | | | 0 | 0 | | | | 0 |
| 0 | | | 0 | 0 | 0 | 0 | | | |
| | | 0 | 0 | 0 | | 0 | 0 | | |
| | 0 | 0 | 0 | | | | 0 | 0 | |
| 0 | 0 | 0 | | | | | | 0 | 0 |

Proceed filling in numbers starting with the black cells left above. If the $x^{th}$ cell is black then we write number *x* into the cell, row after row which is here: 1, 2, -, -, -, -, -, -, 9, 10; -, -, 11, 12,…

Then we go on with the green cells starting right above counting from right to left. The first row reads now: 1, 2, 8, -, -, -, -, -, 9, 10.

For the red cells we start in last row at the left side and go from left to right and up the matrix row for row. The first red cells are in row 5, which reads now: 41, 49, 53, -, 45, 46, -, 58, -, 50.

The white (here empty) cells are remaining. We start right below and count from right to left. The last row should look like as 91, 92, 98, 7, 6, 5, 4, 3, 99, 100.

The next pages show my DERIVE procedure. I wanted to do it step by step as given in the directions above. It is very likely that there is a shorter program possible.

You can follow the three main steps. I printed the MSqs how they were generated.

```
evmq(n, sq1, sq, i) :=
  Prog
    If MOD(n, 4) ≠ 2
       RETURN "order must be 4k+2!"
    k := (n - 2)/4
    sq1 := VECTOR(VECTOR("w", j, n/2), i, n/2)
    sq := VECTOR(VECTOR("w", j, n), i, n)
    "Producing the black cells in the base square"
    i := 1
    Loop
      If i > n/2 exit
      j := 0
      Loop
        If j = k exit
        If i + j > n/2
           sq↓i↓(i + j - n/2) := "b"
           sq↓i↓(i + j) := "b"
        j :+ 1
      "DISPLAY(i)"
      "DISPLAY(sq↓i)"
      i :+ 1
    "DISPLAY(sq)"
    "Producing the green cells in the base square"
    i := 1
    Loop
      If i > n/2 exit
      "DISPLAY([i, i + k])"
      If i + k > n/2
         sq↓i↓(i + k - n/2) := "g"
         sq↓i↓(i + k) := "g"
      "DISPLAY(sq↓i)"
      i :+ 1
    "Producing the red cells in the base square"
    i := 1
    Loop
      If i > n/2 exit
      If i + k + 1 > n/2
         sq↓i↓(i + k + 1 - n/2) := "r"
         sq↓i↓(i + k + 1) := "r"
      i :+ 1
```

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| b | b | g | r | w | w | w | w | w | w |
| w | b | b | g | r | w | w | w | w | w |
| r | w | b | b | g | w | w | w | w | w |
| g | r | w | b | b | w | w | w | w | w |
| b | g | r | w | b | w | w | w | w | w |
| w | w | w | w | w | w | w | w | w | w |
| w | w | w | w | w | w | w | w | w | w |
| w | w | w | w | w | w | w | w | w | w |
| w | w | w | w | w | w | w | w | w | w |
| w | w | w | w | w | w | w | w | w | w |

The left above initial square

In the next loop we reflect the cells.

```
    "Reflecting the cells"
    i := 1
    Loop
      If i > n exit
      k := 1
      Loop
        If k > n exit
        If sq↓i↓k = "b"
           Prog
             sq↓i↓(n - k + 1) := "b"
             sq↓(n - i + 1)↓k := "b"
             sq↓(n - i + 1)↓(n - k + 1) := "b"
        If sq↓i↓k = "g"
           sq↓(n - i + 1)↓k := "g"
        If sq↓i↓k = "r"
           sq↓i↓(n - k + 1) := "r"
        k :+ 1
      i :+ 1
```

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| b | b | g | r | w | w | r | w | b | b |
| w | b | b | g | r | r | w | b | b | w |
| r | w | b | b | g | w | b | b | w | r |
| g | r | w | b | b | b | b | w | r | w |
| b | g | r | w | b | b | w | r | w | b |
| b | g | w | w | b | b | w | w | w | b |
| g | w | w | b | b | b | b | w | w | w |
| w | w | b | b | g | w | b | b | w | w |
| w | b | b | g | w | w | w | b | b | w |
| b | b | g | w | w | w | w | w | b | b |

The wholesquare with “colored“ cells

Last step: filling in the numbers according to the colors.

```
"Filling the cells with numbers"
""
i := 1
Loop
  If i > n exit
  k := 1
  Loop
    If k > n exit
    If sq↓i↓k = "b"
       sq↓i↓k := (i - 1)·n + k
    If sq↓i↓k = "g"
       sq↓i↓k := n·i - k + 1
    If sq↓i↓k = "r"
       sq↓i↓k := (n - i)·n + k
    If sq↓i↓k = "w"
       sq↓i↓k := (n - i)·n + n - k +
    k :+ 1
  i :+ 1
sq
```

MQ := evmq(10)

MQ :=

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 8 | 94 | 96 | 95 | 97 | 93 | 9 | 10 |
| 90 | 12 | 13 | 17 | 85 | 86 | 84 | 18 | 19 | 81 |
| 71 | 79 | 23 | 24 | 26 | 75 | 27 | 28 | 72 | 80 |
| 40 | 62 | 68 | 34 | 35 | 36 | 37 | 63 | 69 | 61 |
| 41 | 49 | 53 | 57 | 45 | 46 | 54 | 58 | 52 | 50 |
| 51 | 59 | 48 | 47 | 55 | 56 | 44 | 43 | 42 | 60 |
| 70 | 39 | 38 | 64 | 65 | 66 | 67 | 33 | 32 | 31 |
| 30 | 29 | 73 | 74 | 76 | 25 | 77 | 78 | 22 | 21 |
| 20 | 82 | 83 | 87 | 16 | 15 | 14 | 88 | 89 | 11 |
| 91 | 92 | 98 | 7 | 6 | 5 | 4 | 3 | 99 | 100 |

Let's check the *magic number*:

$\text{VECTOR}(\Sigma(MQ_i), i, 10) = [505, 505, 505, 505, 505, 505, 505, 505, 505, 505]$

VECTOR(Σ(MQ↓↓i), i, 10) = [505, 505, 505, 505, 505, 505, 505, 505, 505, 505]

TRACE(MQ) = 505

evmq(14) =

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 11 | 187 | 191 | 190 | 189 | 188 | 192 | 186 | 12 | 13 | 14 |
| 182 | 16 | 17 | 18 | 24 | 174 | 176 | 175 | 177 | 173 | 25 | 26 | 27 | 169 |
| 168 | 167 | 31 | 32 | 33 | 37 | 161 | 162 | 160 | 38 | 39 | 40 | 156 | 155 |
| 141 | 153 | 152 | 46 | 47 | 48 | 50 | 147 | 51 | 52 | 53 | 143 | 142 | 154 |
| 70 | 128 | 138 | 137 | 61 | 62 | 63 | 64 | 65 | 66 | 130 | 129 | 139 | 127 |
| 71 | 83 | 115 | 123 | 122 | 76 | 77 | 78 | 79 | 117 | 116 | 124 | 114 | 84 |
| 85 | 86 | 96 | 102 | 108 | 107 | 91 | 92 | 104 | 103 | 109 | 101 | 97 | 98 |
| 99 | 100 | 110 | 95 | 94 | 93 | 105 | 106 | 90 | 89 | 88 | 87 | 111 | 112 |
| 113 | 125 | 82 | 81 | 80 | 118 | 119 | 120 | 121 | 75 | 74 | 73 | 72 | 126 |
| 140 | 69 | 68 | 67 | 131 | 132 | 133 | 134 | 135 | 136 | 60 | 59 | 58 | 57 |
| 56 | 55 | 54 | 144 | 145 | 146 | 148 | 49 | 149 | 150 | 151 | 45 | 44 | 43 |
| 42 | 41 | 157 | 158 | 159 | 163 | 36 | 35 | 34 | 164 | 165 | 166 | 30 | 29 |
| 28 | 170 | 171 | 172 | 178 | 23 | 22 | 21 | 20 | 19 | 179 | 180 | 181 | 15 |
| 183 | 184 | 185 | 193 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 194 | 195 | 196 |

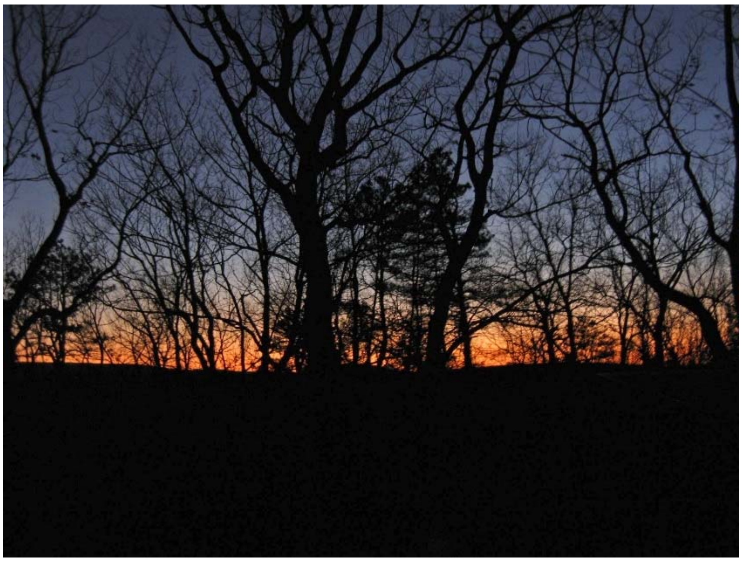
Check the Magic Number of this order 14 Magic Square ☺.!

http://www.mathe-online.at/materialien/maria.koth/files/Magische_Quadrate_Infotext.pdf
https://www.oemg.ac.at/DK/Didaktikhefte/2007%20Band%2040/VortragKoth.pdf

# Where In The World Is IT?

## Understanding How GPS Uses Mathematics

Pat and Carl Leinbach, Bigglerville, USA



**SCENARIO**

Two hikers have been on the trail for five days. Tomorrow they plan to reach their destination and return home from their wilderness vacation. It has been a wonderful week. The weather was great and their trail had a wonderful late fall appearance that is so different from that of the trail in summer. They make camp and cook their evening meal. On a nearby ridge the sunset is giving a spectacular display. The hikers walk over to see it. At this point their idyllic day ends. They discover some bones lying in the woods not far from their campsite. They appear to be human. The skeleton is incomplete. There are some leg bones and some ribs.

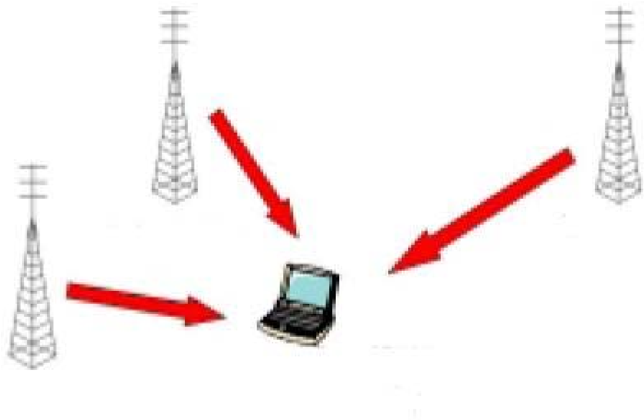
The terrified hikers decide that they must notify the emergency response organizations in the area. They have a pocket size GPS and a cell phone. They call 911 and report their finding and the GPS reading for the position of the bones. The emergency control center dispatcher consults the center's maps of the area and locates the campers' position. The police are notified. There is no need for medical personnel to respond to the scene. On the other hand you are notified, since there apparently is a deceased on the scene.

You call your deputy to accompany you to the scene, put on warm clothing and hiking boots, gather up a flashlight and your crime scene kit, and drive to the nearest access point to the hiking trail. After your deputy arrives, the two of you proceed to the scene. A brief examination indicates that the bones are not human. They appear to be those of a deer who died somewhere in the woods. At several locations on the bones teeth marks are apparent. Scavengers had feasted on the carcass and the bones were probably scattered as they were carried off for private dining.

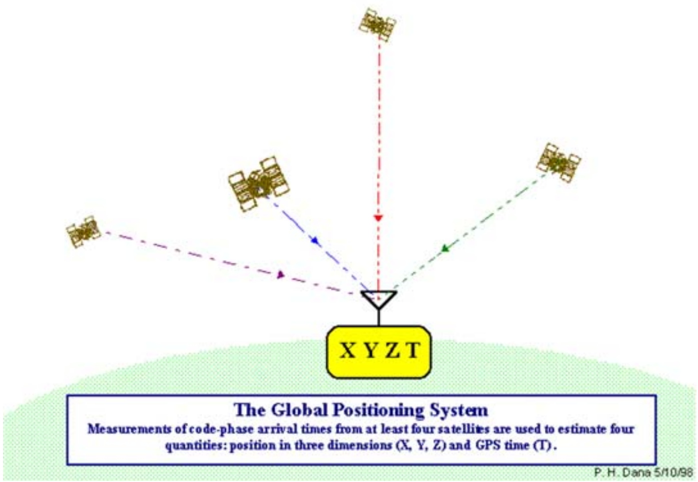
**BACKGROUND**

The use of a GPS in the above scenario was very important because it allowed the control center dispatchers to quickly locate the position on their maps and then alert and efficiently direct the personnel to the scene. This case was a bit of a wild goose chase, but if the emergency had been a real one, quick response could have made a huge difference. Even in this case it shortened a long, dark hike through the woods.

There are two general types of Global Positioning Systems (GPS) being used at present. The first is a 2–Dimensional system. This system uses the band width and time stamping of High Definition TV signals and can penetrate buildings and many areas that traditional GPS systems simply cannot be used. They can only give longitude and latitude coordinates. They do not give the altitude of the location. As a consequence, they are primarily used for local applications where the altitude is relatively constant.



The other is a 3–Dimensional system. It obtains distances from satellites whose position is known. It can return the longitude, latitude, and altitude of the receiving unit. As a result, this system is used for more wide area applications than those used by 2-D applications.




Both systems have their advantages and disadvantages. 2-D systems can penetrate buildings and obtain coordinates, but they cannot determine altitude. 3-D systems can give all three coordinates of position, but require that the receiving unit have a "clear view of the sky."

The basic components of any GPS are:

- A pre assigned coordinate system
- Devices (Television towers or satellites) whose exact positions within the coordinate system are known. The devices must be able to relay their positions by way of sending signals.
- A timing mechanism that is associated with the signal
- A unit that can receive signals from several devices and either quickly compute its distance from each of the devices or relay the information to a base station that can perform these calculations

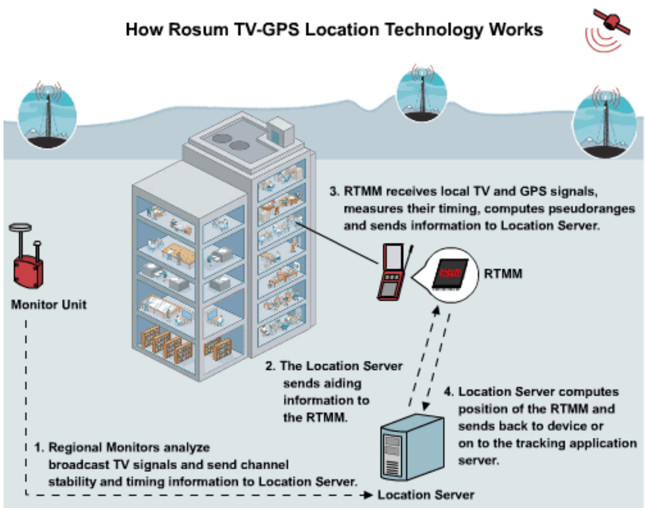
Please note that we have not included any requirements for maps or said anything about navigating based on the calculations. This is a discussion of a *Positioning System*. Getting from here to there is another problem that also involves some interesting mathematics.

## *2-Dimensional GPS*

In the United States and several other countries, 2-D GPS is used to keep track of the movements of prisoners who are placed on house arrest with strict limitations of their movements. It is also used as a means for keeping track of children wearing GPS bracelets or anklets. It has even been responsible for helping to locate lost children and, quite probably saving lives.

Basically, GPS is nothing more than triangulation. A receiving device receives signals from three locations whose exact position is known and that have accurate clocks that are synchronized precisely with each other. The signaling devices are constantly sending their position and the time. The receiving device records this information and stamps it with the time that it received it. The time stamp does not need to be synchronized with the sending devices. From the information received from three sending locations and the time stamp the receiver can either calculate the $x,y$-coordinates of its position, or relay the information to a central processor that will do the calculations. The latter case is shown in the figure below.

The proposed modern systems rely on the synchronization signals that exist within the standard Digital Television transmissions. Thus, the information required for this type of GPS relies on already existing devices and does not require the construction of new transmitting towers.



Using Digital Television Technology has some distinct advantages over using satellite transmissions in certain environments. Digital Television signals have:

- On average about 40dB more power
- A wider band width
- Lower frequencies
- Superior geometry

This means that the signals can reach locations, such as inside buildings, that are inaccessible to standard GPS signals

Thus, this technology has applications that expand the available GPS applications. Here are two applications:

1. Locating the positions of first responders using a module located inside their emergency radios and portable signal generators attached to their belts.
2. Finding lost children in indoor locations, such as shopping malls or amusement parks, or in densely wooded locations using a bracelet or anklet device worn by the child.

*Illustrating How The 2-Dimensional GPS Works*

A child, wearing a "locator bracelet" has gone missing in a shopping mall. Transmitters located in the mall at positions 100 meters East and 200 meters North; 300 meters West and 100 meters North; and 200 meters East and 200 meters South of a central control station record the following distances from the child's bracelet:

Transmitter 1: 223.60680 meters

Transmitter 2: 200.00000 meters

Transmitter 3: 424.26407 meters

The following illustration shows how we can locate the approximate position of the "lost" child using Derive[tm] 6.1 graphics with slider bars.

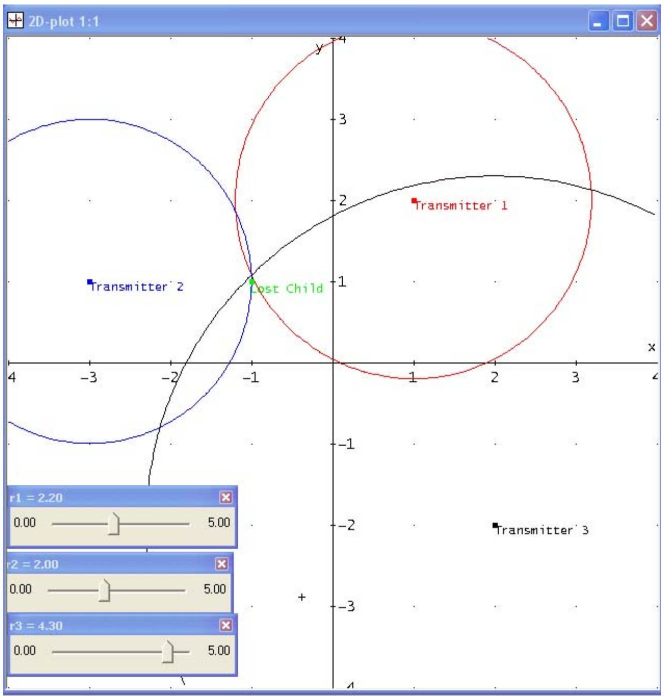
The process is relatively simple. Since the digital tv signal waves travel outward from each transmitter in a uniform circle, we simply construct a coordinate system that contains the points (1, 2), the location of the first transmitter in hundreds of meters, (-3, 1), the location of the second transmitter, and (2, -2), the location of the third transmitter.

Then we enter the three equations of the circles leaving the radius as an unknown in each equation:

$$(x-1)^2+(y-2)^2=r_1^2$$
$$(x+3)^2+(y-2)^2=r_2^2$$
$$(x-2)^2+(y+2)^2=r_3^2$$

As a penultimate step we move to the graphics screen and create a slider bar for each of r1,r2, and r3. Let the range of each slider bar go from 0 to a number large enough to include the distance from the child. Also, allow the increment to be 0.01 so we can slide the bar to the nearest foot of distance

Finally, move the first slider bar to 2.24, the second to 2.00, the third to 4.24.




While the distances are not absolutely accurate, the three circles appear to intersect at a unique point, (-1, -1). Thus, we locate the child at approximately 100 meters West and 100 meters North of the control station.

This solution is not necessarily an exact solution simply because we did not have enough pixels on the screen to give us better decimal accuracy, but the approximate solution got us close enough so that we can locate the child. Furthermore, the process of using slider bars gives us a feeling of how the GPS operates.

*Solving The Equations Exactly*

Radio and TV signals are waves traveling at the speed of light  If we let *c* denote the speed of light (*c* = 299,792.458 km/sec), then

$$\text{distance traveled} = \text{c} * (\text{time traveed}) = \text{c} * (\text{time traveled - discrepency})$$

or,

$$d = c * (t - \delta)$$

where δ = discrepancy between the true time and the observed time.

Let's consider the information that is resident in the receiving unit of the GPS.

Transmitter #1:

Location: $(a_{11}, a_{12})$

Observed time to receive signal: $t_1$

Transmitter #2:

Location: $(a_{21}, a_{22})$

Observed time to receive signal: $t_2$

Transmitter #3:

Location: $(a_{31}, a_{32})$

Observed time to receive signal: $t_3$

This information results in the following system of equations:

$$(x-a_{11})^2+(y-a_{12})^2=c^2(t_1-\delta)^2$$
$$(x-a_{21})^2+(y-a_{22})^2=c^2(t_2-\delta)^2$$
$$(x-a_{31})^2+(y-a_{32})^2=c^2(t_3-\delta)^2$$

NOTE: The geometric approximate solution is no longer available since we do not, at this stage, know the distance between the transmitter and receiver. All that we know is the time that the signal was received from the transmitter.

The system of equations that we have to solve is a system of three *non-linear* equations in three unknowns: $x$, $y$, and $\delta$. The problem is that we only know how to solve systems of *linear* equations. One possible solution to our dilemma is to introduce a fourth transmitter and use the data and resulting equation from that transmitter to reduce the above system of non-linear equations to a system of three linear equations in three unknowns.

The problem with this solution is that a fourth transmitter may not always be available and, even if one is, there is the problem of synchronization of this transmitter with the existing three transmitters, i.e. we are increasing the probability of error.

To help us out with the solution of our problem, we will use the non-linear equation solver that exists on our CAS. Yes, we are accepting a solution that uses a method that we do not know, and in mathematics, blind faith can be dangerous. On the other hand, it is the author's hope that you will pursue the study of such methods, as you learn more mathematics. Consider this tool as our means of whetting your appetite for more advanced study.

We will solve the same problem of the lost child that we considered above. Remember that our unknowns are $x$, $y$, and $\delta$. We will use the value of

$$c = 0.299792458 \text{ km/µsec}$$

for the speed of light. All of our measurements will be in kilometers.

The following is the Derive[tm] 6.1 screen for solving the system of three equations in 3 unknowns. The solve command uses the same format as for linear equations.

```
#1:   c := 0.299792458

The time stamps on the receiving unit:

#2:   t := [6.377871991, 6.29912819, 7.047192602]

                      2          2    2         2
#3:   k1 := (x - 0.1)  + (y - 0.2)  = c ·(t  - δ)
                                         1

                      2          2    2         2
#4:   k2 := (x + 0.3)  + (y - 0.1)  = c ·(t  - δ)
                                         2

                      2          2    2         2
#5:   k3 := (x - 0.2)  + (y + 0.2)  = c ·(t  - δ)
                                         3

#6:   [Precision := Mixed, Notation := Mixed]

#7:   [PrecisionDigits := 25, NotationDigits := 10]
```

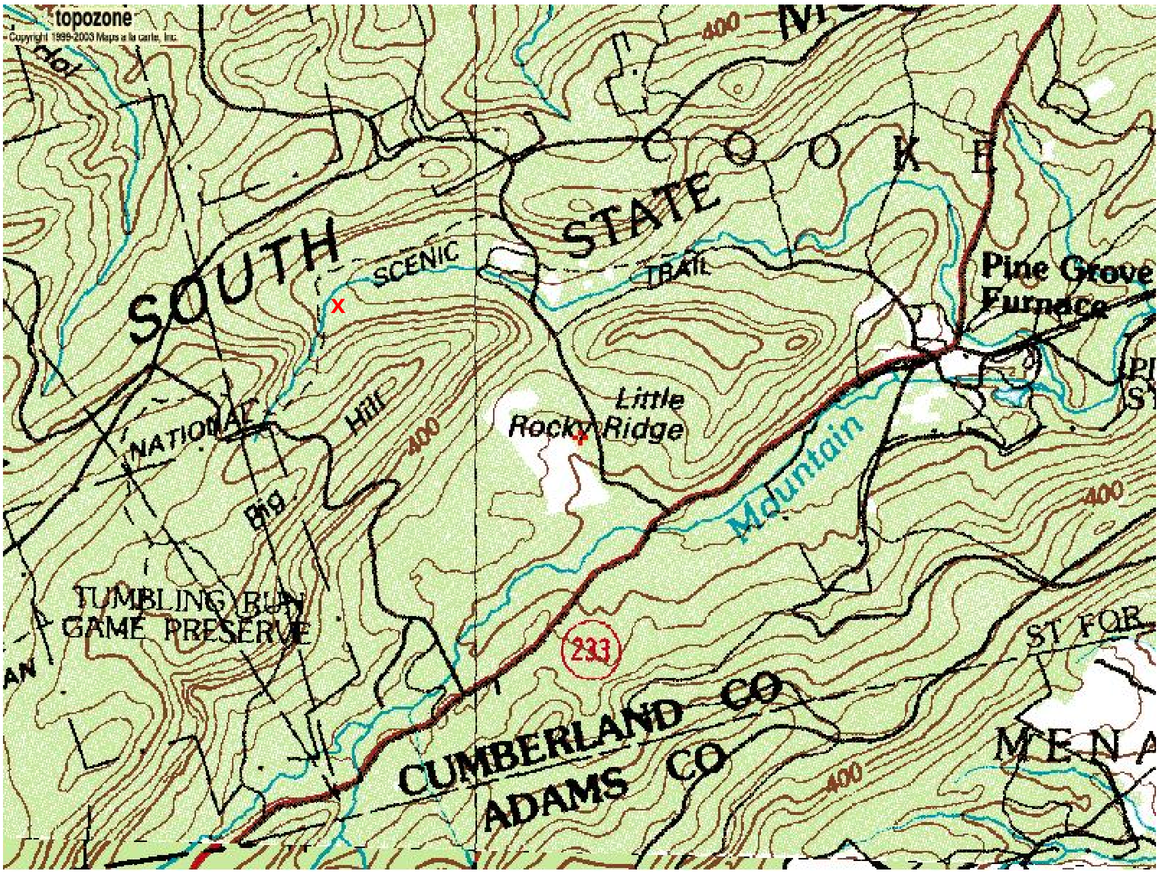
```
#6:   [Precision := Mixed, Notation := Mixed]

#7:   [PrecisionDigits := 25, NotationDigits := 10]

#8:   SOLVE(k1 ∧ k2 ∧ k3, [x, y, δ])

#9:   (x = 0.007232271666 ∧ y = -0.1830929454 ∧ δ = 7.692664945) ∨ (x =

#10:  SOLUTIONS(k1 ∧ k2 ∧ k3, [x, y, δ])

      ⎡ -0.09999999998   0.1000000001    5.631999999 ⎤
#11:  ⎢                                               ⎥
      ⎣ 0.007232271666  -0.1830929454    7.692664945 ⎦
```

Note that there are two possible solutions to this set of equations. This is a result that there could be two values for δ. We note that the second is the solution we found using the geometric method. Recall the position is given in kilometers, δ in μseconds.

One last caution for this exercise: use the simplify button (=), not the approximate button (≈) after entering the solve command.

*Creating An Exercise For Others to Solve*

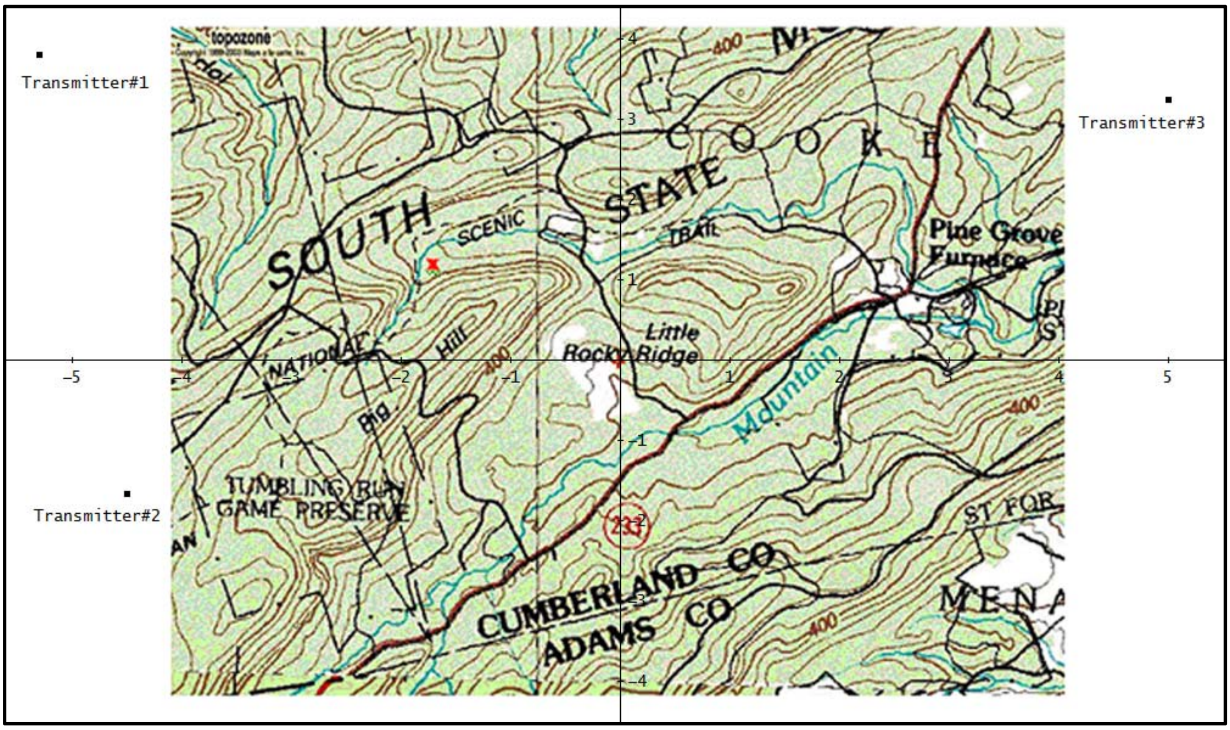
For this section we return to the situation described in the scenario. The hikers were on a trail in the deep woods. Below is a topographical map of the area where they were hiking. The map was obtained from **topozone**tm (http://www.topozone.com ). The center of the map is located at GPS coordinates 40° 2.77'N 77° 22.28'W. The scale is: 1:50 000 (1 km = 2 cm).




A red '×' just below the 'H' in 'SOUTH' locates the spot where the bones were found along the hiking trail.

We are going to use Derive[tm]-6.1 to create a problem based on the scenario. Our first task is to create a Derive graphing screen that has the map in the background with the center of the map at the origin of the coordinate system. Follow these steps:

1. Open a Derive full screen graphing window
2. Using a metric ruler measure the length of each axis and choose a range for each axis (Set > Plot Range > Length/Center) so that 1unit = scale for 1 km. (in our case 1 unit = 2 cm)
3. If the map image is not a Bit Map image, create a Bit Map image of the map.
4. Go to Options > Display > Color > Picture. Click on the button to the right of right of the empty text box and browse for the Bit Map image of your map. Place its name in the text box. Choose the 'Center' option and click OK.
5. Using the pointer check the location of where the bones were found relative to the center of the coordinated system. In our case it is at (-1.734, 1.237)
6. Arbitrarily, choose three positions for the transmitter towers. They need not appear on the screen, but for this illustrative problem, the points chosen were (5, 3.25); (-5.3, 4); and (-4.5, -1.67). These plots were plotted and labeled as shown below.




NOTE: The map size has been reduced to fit the Derive screen onto this page. However, the scale has been preserved.

We now outline the procedure for constructing a problem to be solved by someone else.

1. Decide on a time discrepancy, δ, between the synchronized transmitter clocks and the GPS receiver. Let's choose δ = 23.57 μsec. Don't publish this value, but it is needed to compute the time stamp to be used in the equations to be solved.

2. Calculate the times in µsec needed for the signals to reach the GPS receiver.
    a. Compute the distance in km from each transmitter to the (unpublished) position of the GPS receiver.
    b. Divide these distances by the speed of light in km/µsec.
3. Add δ to each of these times, these are the time stamps to be used in the equations.
4. Create the equations to be solved and check the result.

$$\#12: \quad \text{transm} := \begin{bmatrix} 5 & 3.25 \\ -5.3 & 3.8 \\ -4.5 & -1.67 \end{bmatrix}$$

$$\#13: \quad \text{loc} := [-1.72,\ 1.18]$$

A vector of time stamps to be used in the equations

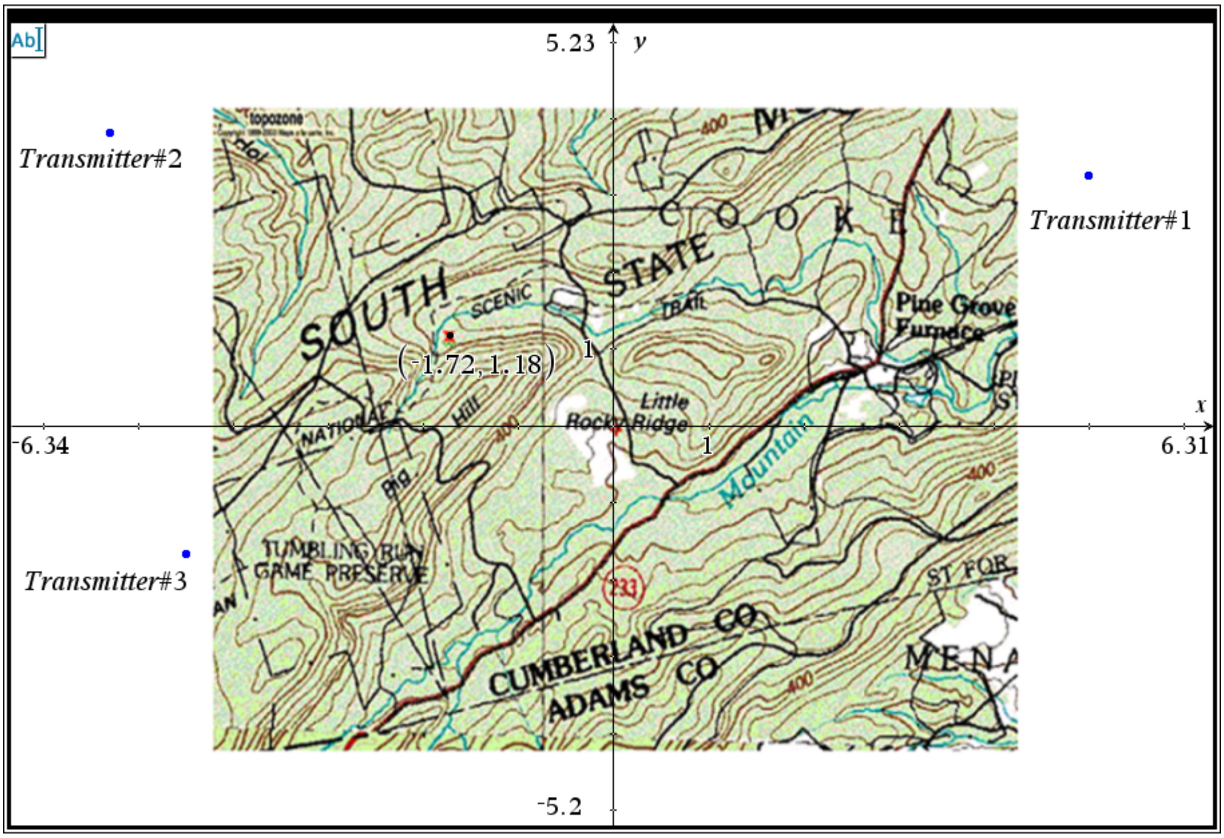
$$\#14: \quad t := \text{VECTOR}\left(\frac{|v - \text{loc}|}{c} + 23.57,\ v,\ \text{transm}\right)$$

$$\#15: \quad t := [47.02486954,\ 38.36791986,\ 36.85025028]$$

$$\#16: \quad \text{SOLUTIONS}\left(\text{VECTOR}\left((x - \text{transm}_{i,1})^2 + (y - \text{transm}_{i,2})^2 = (c\cdot(t_i - \delta))^2,\ i,\ 3\right),\ [x,\ y,\ \delta]\right)$$

$$\#17: \quad \begin{bmatrix} 1.118416427 & 2.50870069 & 60.20644125 \\ -1.72 & 1.18 & 23.56999999 \end{bmatrix}$$

You can see the necessary DERIVE commands above. The TI-NspireCAS treatment is given below.



$transm:=\begin{bmatrix}5 & 3.25\\ ^-5.3 & 3.8\\ ^-4.5 & ^-1.67\end{bmatrix}$ $\quad\begin{bmatrix}5 & 3.25\\ ^-5.3 & 3.8\\ ^-4.5 & ^-1.67\end{bmatrix}$

$loc:=[^-1.72\ \ 1.18]$ $\quad[^-1.72\ \ 1.18]$

$c:=0.299792458$ $\quad 0.299792458$

$t:=\mathrm{seq}\left(\frac{\mathrm{norm}(transm[i]-loc)}{c}+23.57,i,1,3\right)$ $\quad\{47.0248695406,38.3679198616,36.8502502889\}$

$eqs:=\mathrm{seq}\left((x-transm[i,1])^2+(y-transm[i,2])^2=(c\cdot(t[i]-\delta))^2,i,1,3\right)$

$\{x^2-10\cdot x+y^2-6.5\cdot y+35.5625=0.089875517874\cdot(\delta-47.0248695406)^2,x^2+10.6\cdot x+y^2-7.6\cdot y+42.53=0$▸

$\mathrm{solve}(eqs[1] \text{ and } eqs[2] \text{ and } eqs[3],\{x,y,\delta\})$

$x=^-1.72$ and $y=1.18$ and $\delta=23.57$ or $x=1.11841642705$ and $y=2.5087006906$ and $\delta=60.2064412503$

Once again there are two answers for the problem. The second is the one that you are searching for. You are now ready to publish the map and your problem giving the positions of the transmitters and the time stamps shown in the receiver.

As you can see above, we can treat the problem with TI-NspireCAS, too. We have the background picture available and we can also solve nonlinear systems of equations if they are not too complicated. The seq-function takes the role of Derives powerful VECTOR-command.

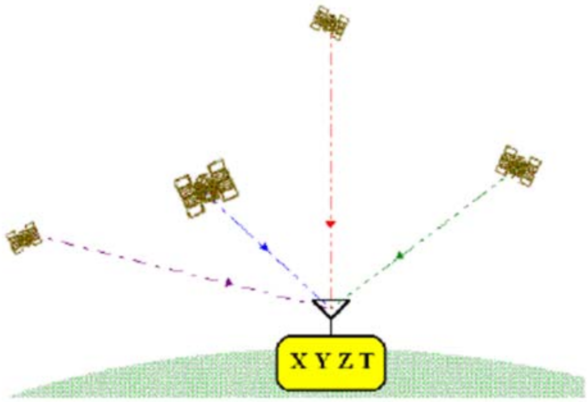
*3-Dimensional GPS*

Much of the material for this section was taken from the web site of Peter H. Dana of the Department of Geography of the University of Texas at Austin. The URL for this information is located at the University of Colorado.

http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html

Having carefully worked through the two-dimensional problem, the basic workings of the three-dimensional GPS should be clear to us. Instead of being able to determine the *x* and *y*-coordinates of an object, we can determine the *x*, *y*, and *z*-coordinates, i.e. latitude, longitude, and altitude. Of course, our GPS receiver will have an unknown time discrepancy with the atomic clocks in the sending satellites, so we end up solving a system of four non-linear equations with four unknowns.

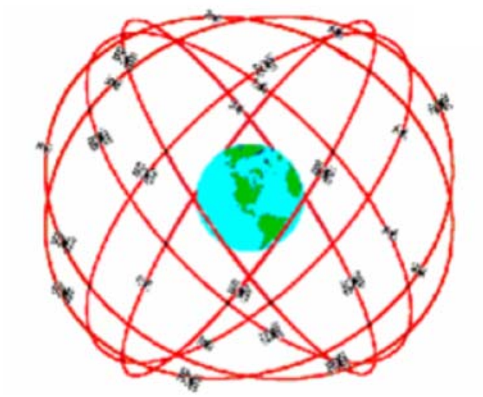
Contrary to what some may think, a GPS satellite knows nothing about the position of your GPS receiver. It knows only two facts: its own position in space and the exact time as synchronized by an atomic clock with the times of all of the other GPS satellites. Your GPS receiver receives this information from four satellites and calculates your distance from each of the satellites and the discrepancy between its clock and the satellite clocks. From this information, it can determine to within a certain numerical accuracy your position on the Earth. (See illustration below.)




The GPS

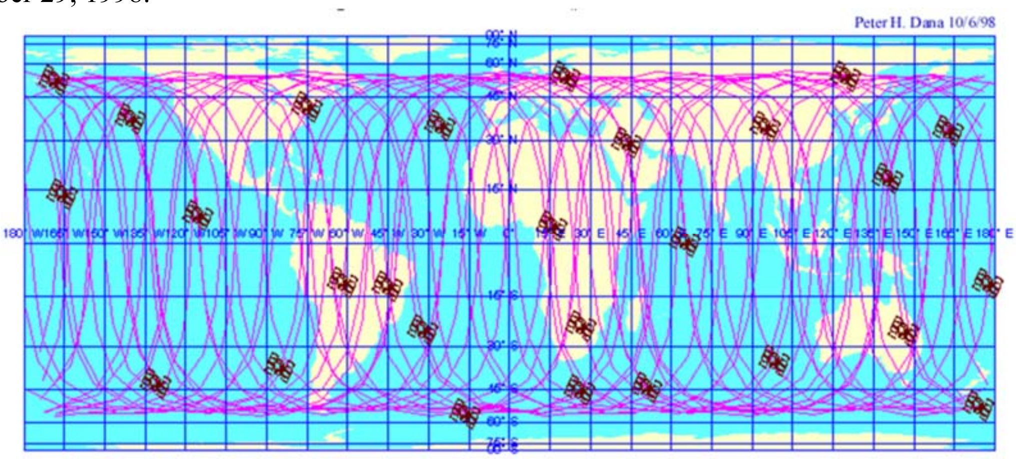
Measurements of code-phase arrival times from at least four satellites are used to estimate four values: the local position in three dimensions (X, Y, Z) and the GPS-time T.

There are 27 GPS satellites positioned in orbits that are 22,000 kilometers (13670.1662292 miles) above mean sea level, MSL. The satellites orbits are at an angle of 55° with the Earth's equator. We will assume that at MSL the Earth is a sphere of radius 3982.998 miles. This positioning is shown in the following illustration



Measurements of code-phase arrival times from at least four satellites are used to estimate four values: the local position in three dimensions (X, Y, Z) and the GPS-time T.

The satellites are positioned so that at any moment, every spot on earth is "visible" by at least four of the satellites. Here is an illustration that depicts the satellites positions above the earth at one moment on September 29, 1998.



**FORENSIC OBJECTIVE**

1. To understand the use of a tool that can be of use for more completely describing the scene of death or discovery of a body, especially when it is located in a remote or not easily accessible location.

**MATEMATICAL/SCIENTIFIC OBJECTIVE**

1. To understand the role of mathematics in the workings of a tool that is becoming widely used in all aspects of our daily lives.
2. To learn how to use a Computer Algebra System for solving some systems of non-linear equations.
3. To inspire you to learn more mathematics so that you can understand the mathematics behind the solutions found by the Computer Algebra System.

**MATERIALS NEEDED**

1. A Computer Algebra System with the capability of placing pictures on the coordinate system of the graphing tool, the ability to graph functions and equations having an undefined parameter, a slider bar, and the ability to give highly accurate approximate decimal solutions to systems of nonlinear equations.
2. A detailed map of a restricted area that, if possible, contains your school's location and other public buildings, monuments, streets, etc. in that location. An alternative is a topographical map of an area which is near your location.
3. A metric ruler or measuring tape.
4. 

We repeat and summarize the directions how to set up problems for students:

**ACTIVITY**

*Geometrically Solving the Problem*

1. Using the topographical map given above that is not on the Derive screen, but has the scale of 2cm = 1km, arbitrarily pick a new location for the GPS receiver.
2. Place a coordinate system on the map with its origin at the center of the map. Place tick marks along the axes at distances of one km.
3. Place the three transmitters at positions having integer *xy*-coordinates. They need not be on the detailed portion of the map.
4. Measure the distance from each transmitter to the location of the receiver on the map.
5. Write the equations for circles with their centers at each of the transmitters and with the radius of each circle as a separate parameter.
6. Open the graphics screen and set the aspect ratio to 1:1 and set up slider bars for each of the radii of the circles. Be sure to include a range of values that will allow the circles to intersect.
7. Find approximate values for the position of the GPS receiver.

*Creating A Positioning Problem To Be Solved*

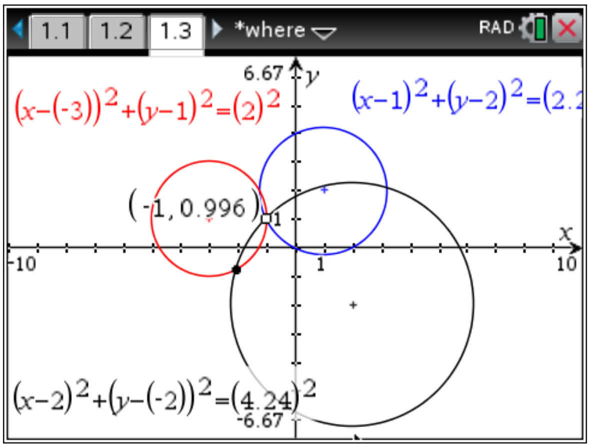
1. Using the map of your location, choose a prominent land mark (public building, monument, or street intersection) as the location for your receiver.
2. Place coordinate axes on your map and use the map scale to establish units for the coordinate system.
3. Choose three locations for your transmitters and a value in μseconds for δ, the time discrepancy between the receiver and the transmitters.
4. Follow the steps given above and construct your problem.
5. Before giving the problem to the class, check it by solving it yourself.

I will add two possible extensions (Josef):

### Extension #1

We can solve the problem either applying Dynamic Geometry plotting the circles (Voyage 200) or even more comfortable with TI-Nspire in the Graph-Window entering the equations of the circles and finding the intersection point.

This works even on the small screen of the handheld version.




### Extension #2

We can reduce the nonlinear of the three equations of circles with two unknowns to a linear system with three unknowns (without regarding the discrepancy δ).

$$(x-1)^2+(y-2)^2=r_1^2 \qquad x^2+y^2-2x-4y=2.24^2-1-4=0.0176$$
$$(x+3)^2+(y-2)^2=r_2^2 \quad \text{We expand} \quad x^2+y^2=2^2-9-1=-6$$
$$(x-2)^2+(y+2)^2=r_3^2 \qquad x^2+y^2=4.24^2-4-4=9.9776$$

And then set $u = x^2 + y^2$. Solve the resulting simultaneous equations and compare the results.

**CORONER'S COMMENTS**

Every coroner has had the experience of being called to supposed coroner's scenes only to find that the unearthed bones belong to an animal. The most common is the burial scene for a family pet. I had the experience of being called to a 'scene' by the State Police only to find that the bones were those of a beef cow that was butchered and the bones dumped off in a particularly bushy forested area. Because all of the meat had not been cleaned off of the bones, the smell was most unpleasant. Another time, I was called to the scene of a home remodeling. There was the fear that a disposal place for a mass murderer had been unearthed. It turned out that the former owner had had a small butcher shop on the premises and the bones were discarded pork bones. As noted, the experience of having to identify non-human skeleton bones is common.

It is always important for the coroner to make a detailed description of the scene of a coroner's case. The GPS is a tool that can be of help in making such a description. This is more commonly the case when the scene is in a remote location and not near any prominent land marks. At present, the GPS is used more commonly in crime prevention, and location of individuals wearing GPS locators. A recent use is for individuals suffering from Alzheimer's Disease who may be prone to wondering off alone.

**Duties of a Coroner (in State Pennsylvania)**

Pennsylvania has mixed Coroner/medical investigation system. The function can be occupied according to the regulations of the county by election or by appointment. Coroners need not to be doctors and need not to have a medical past but they must inscribe a course and pass the respective final exam.

Typical tasks of a coroner are:

Identifying the victim
Detect the manner and cause of death
Finding the time of death
Informing the next of kin

Pat Leinbach was coroner in Adams County for many years. You can find more information about Coroner's duties and about Coroners in Adamns County in [6] and [7].

**REFERENCES**

[1] Proceedings CD of the Intl DERIVE & TI-Conference 2000, Liverpool

[2] Proceedings DES-TIME 2006, Dresden

[3] P. und C. Leinbach, J. Böhm; *Forensische Mathematik für den Unterricht*, bk-teachware 2007

[4] www.colorado.edu/geography/gcraft/notes/gps/gps_f.html

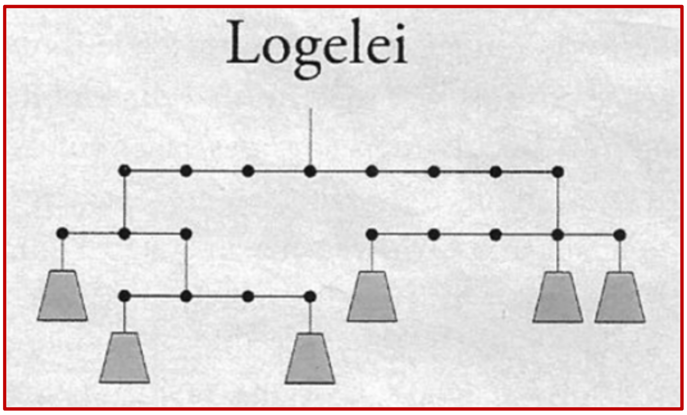
[5] www.gps.gov/support/website/

[6] en.wikipedia.org/wiki/Coroner

[7] www.adamscounty.us/Dept/Coroner/Pages/default.aspx

# New Brain Twisters from "*Die Zeit*" CAS-treated

Josef Böhm, Würmla, Austria




Mrs Smith demonstrates proudly her new mobile to her husband. "I fixed all figures such that they are in equilibrium." He asked: Great, how could you manage this?". "Oh, it was so easy, I weighed them and then it was no problem to calculate the lengths of the wires. This is an extra light but very stiff material, so its weight can be neglected."

The weights are 67 g, 134 g, 201 g, 268 g, 335 g and 402 g.

What is the order of the weights from left to right?

```
      perm(v, n, vv, n, k_, n_ := 2, s_ := [[1]], t_, ii) :=
        Prog
          vv := v
          Loop
            If n_ > n exit
            k_ := n_
            t_ := []
            Loop
              If k_ = 0 exit
#1:           t_ := APPEND(t_, VECTOR(INSERT(n_, v_, k_), v_, s_))
              k_ :- 1
            s_ := t_
            n_ :+ 1
          If n = DIM(v)
             RETURN VECTOR(vv↓ii, ii, s_)
          v := POWER_SET(MAP_LIST(v↓j_, j_, {1, ..., DIM(v)}), n)
          v := VECTOR(SORT(v_), v_, v)
          APPEND(VECTOR(VECTOR(v_↓u_, u_, s_), v_, v))

#2:   all := perm([67, 134, 201, 268, 335, 402], 6)

#3:   DIM(all) = 720

#4:   [a := v↓1, b := v↓2, c := v↓3, d := v↓4, e := v↓5, f := v↓6]

#5:   [eq1 := 2·c = b, eq2 := a = b + c]

#6:   [eq3 := f = 3·d, eq4 := 3·(a + b + c) = 4·(d + e + f)]

#7:   (SELECT(eq1 ∧ eq2 ∧ eq3 ∧ eq4, v, all))↓1

#8:   [402, 268, 134, 67, 335, 201]
```

I apply a program from an earlier DNL generating all permutations of order *k* of *n* elements. I assign variables *a* to *f* to the weights from left to right and then set up the equations according to the lever rule: load × load arm = effort × effort arm. This is the important part for the students to do. Then we try to find a solution, which finally is given by: $a = 402$, $b = 268$, $c = 134$, $d = 67$, $e = 335$ and $f = 201$.

## Oscar's Zoo

"I don't want to go to the zoo because I don't like watching these boring penguins!", Oscar moans. "When I will be grown up, then I will open my own zoo where the visitors have to pay for each single animal. Then they can do without the penguins."

Oscar is so fascinated by his idea that he starts making a pricelist. Finally his list is ready and reads as follows:

AAL (eel): 9 EURO, AFFE (monkey): 18 EURO, ELEFANT: 40 EURO, ENTE (duck): 29 EURO, RABE (raven): 22 EURO, RATTE (rat): 31 EURO, TARANTEL: 45 EURO and ZEBRA: 23 EURO.

"Oh, that's pretty expensive. How did you get these prices?", asks Oscar's mother. "That was easy done", he answers,"I assigned each of the appearing letters a fixed number between 1 and 9 and then I summed up the numbers."

How does Oscar's assigment look like?

(I have to use the German words for the animals, otherwise it does not work.)

```
#2:   [eq1 := 2·a + l = 9, eq2 := a + 2·f + e = 18, eq3 := 2·e + l + f + n + t + a = 40]

#3:   [eq4 := 2·e + n + t = 29, eq5 := r + a + b + e = 22, eq6 := r + a + 2·t + e = 31]

#4:   [eq7 := 2·t + 2·a + r + n + l + e = 45, eq8 := z + e + b + r + a = 23]

#5:   all := perm([1, 2, 3, 4, 5, 6, 7, 8, 9], 9)

#6:   DIM(all) = 362880
```

#7: $[a := w_1, b := w_2, e := w_3, f := w_4, l := w_5]$

#8: $[n := w_6, r := w_7, t := w_8, z := w_9]$

```
#9:   cond := eq1 ∧ eq2 ∧ eq3 ∧ eq4 ∧ eq5 ∧ eq6 ∧ eq7 ∧ eq8
```

#10: $(\text{SELECT}(\text{cond}, w, \text{all}))_1 = [2, 3, 8, 4, 5, 7, 9, 6, 1]$

```
#11:  28 sec
```

The second way to find the solution is more interesting:

```
#12:  SOLUTIONS([eq1, eq2, eq3, eq4, eq5, eq6, eq7, eq8], [a, b, e, f, l,
        n, r, t, z])

#13:  [[@1, 10·@1 - 17, 14 - 3·@1, @1 + 2, 9 - 2·@1, @1 + 5, 25 - 8·@1,
        5·@1 - 4, 1]]

#14:  VECTOR([@1, 10·@1 - 17, 14 - 3·@1, @1 + 2, 9 - 2·@1, @1 + 5, 25 -
        8·@1, 5·@1 - 4, 1], @1, 1, 10)
```

We solve the system and receive a solution containing one parameter @1. Then we generate a list of possible solution varying the parameter from 1 to 9 (intergers only).

#15:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | −7 | 11 | 3 | 7 | 6 | 17 | 1 | 1 |
| 2 | 3 | 8 | 4 | 5 | 7 | 9 | 6 | 1 |
| 3 | 13 | 5 | 5 | 3 | 8 | 1 | 11 | 1 |
| 4 | 23 | 2 | 6 | 1 | 9 | −7 | 16 | 1 |
| 5 | 33 | −1 | 7 | −1 | 10 | −15 | 21 | 1 |
| 6 | 43 | −4 | 8 | −3 | 11 | −23 | 26 | 1 |
| 7 | 53 | −7 | 9 | −5 | 12 | −31 | 31 | 1 |
| 8 | 63 | −10 | 10 | −7 | 13 | −39 | 36 | 1 |
| 9 | 73 | −13 | 11 | −9 | 14 | −47 | 41 | 1 |

Solution:

[a, b, e, f. l, n, r, t,,z] = [2, 3, 8, 4, 5, 7, 9, 6, 1]. Check it!

## Tanja's PhD celebration

Tanja made her PhD in mathematics. Her family had a special idea honoring her success. They fixed a note on her doctor's degree: "*In the requested number every digit either appears multiple or not all. No two same numbers follow each other*." While Tanja is studying this strange message all friends are hurrying away. Tanja wants to enter the room with the celebrating guests but the room is locked and there is another note fixed on the combination lock: "*The number is divisible by 55 but not by 25 and its sum of digits is 23.*" Tanja investigates the lock: there are 8 number rings containing numbers from 1 to 9 (no zero).

She does not need much time, then she smiles and opens the door. All guests are cheering when she enters the room. Which number was necessary to open the lock?

First of all we need a procedure to create all permutations with replacement where *k* elements are selected from *n* elements. The *n* elements are given in vector (list) *v* below.

```
       vars(v, k, b, k_, m_ := 0, n_, s_ := [], t_) :=
         Loop
           b := DIM(v)
           If m_ = b^k
              RETURN REVERSE(s_)
           k_ := k
           n_ := m_
#1:        t_ := []
           Loop
             t_ := ADJOIN(v↓(MOD(n_, b) + 1), t_)
             n_ := FLOOR(n_, b)
             k_ :- 1
             If k_ = 0 exit
           s_ := ADJOIN(t_, s_)
           m_ :+ 1

#2:    all := vars([1, 2, 3, 4, 5], 7)

#3:    DIM(all) = 78125
```

All possible permutations (in German textbooks often called *Variationen* to distinguish arrangements regarding the order from the others which are the Combinations) with replacement of the nine digits in groups to eight digits – from 11111111 to 99999999 (without 0) are $9^8$ = 43 046 721 arrangements. Testing all of them would work even too hard for DERIVE.

So we will support the "artificial brain" with the "human brain". The last digit must be 5 (divisible by 55), i.e. we need a 7 digit number only. The sum is given with 23 which is now 18 for the remaining 7 digits. The numbers must appear multiple, so 9, 8, 7 and 6 cannot appear among them.

Our problem is reduced: we have to consider the permutations of [1,2,3,4,5] in groups of seven with replacement. Their total number is 78 125. Too many 7-digit numbers to print them out but not too much to have them in DERIVE's memory.

The first condition is thet the sum must be 18, the second that no to fives can follow each other:

#4: $\text{cond1} := \Sigma(v) = 18 \wedge v_7 \neq 5$

#5: $\text{cond2} := \text{MOD}(v \cdot [10^7, 10^6, 10^5, 10^4, 10^3, 10^2, 10] + 5, 55) = 0$

#6: sel := SELECT(cond1 ∧ cond2, v, all)

$$\begin{bmatrix} 1 & 2 & 1 & 5 & 1 & 5 & 3 \\ 1 & 2 & 1 & 5 & 2 & 5 & 2 \\ 1 & 2 & 1 & 5 & 3 & 5 & 1 \end{bmatrix}$$

We see the first possible solutions – reduced to a number of 100.

```
#8:   DIM(sel) = 100
```

#9: $sel1 := VECTOR(v\cdot\left[10^7,\ 10^6,\ 10^5,\ 10^4,\ 10^3,\ 10^2,\ 10\right] + 5,\ v,\ sel)$

```
#10:  sel1 = [12151535, 12152525, 12153515, 12251525, 12252515, 12351515,
        13251425, 13252415, 13341515, 13351415, 14131535, 14132525, 14133
        14232515, 14241425, 14242415, 14251325, 14252315, 14331515, 14341
```

I transform the resulting permutations to the respective numbers – including the last digit 5. In the next step I sort out all numbers divisible by 25. 70 numbers will remain.

```
#11:  sel2 := SELECT(MOD(v, 25) ≠ 0, v, sel1)
#12:  DIM(sel2) = 70
```

The next step is to eliminate all numbers with two subsequent same numbers (11, 22, 33 and 44).

```
#13:  sel3 := SELECT(POSITION(11, v) = false, v, sel2)
#14:  DIM(sel3) = 70
#15:  sel4 := SELECT(POSITION(22, v) = false, v, sel3)
#16:  DIM(sel4) = 62
#17:  sel5 := SELECT(POSITION(33, v) = false, v, sel4)
#18:  DIM(sel5) = 50
#19:  sel6 := SELECT(POSITION(44, v) = false, v, sel5)
#20:  DIM(sel6) = 50
```

Now only 50 numbers are remaining.

Both a single 4 and single 3 are not permitted, hense:

```
#21:  sel7 := SELECT(POSITION(4, v) ∨ POSITION(3, v) = false, v, sel6)
#22:  sel7 := [14242415, 24142415, 24241415, 25251215]
#23:  25251215
```

By inspection we can easily find the right code among the last four numbers.

Solution:

Only the last number shows more than one fives. The number to unlock the door is 25251215.

**References:**

Logelei, *Die Zeit*, Nr. 48, 2016
Tanja's PhD Celebration, *Die Zeit*, Nr. 50, 2016
Oscar's Zoo, *Die Zeit*, Nr. 38, 2016

We had a great time in Ecuador. I will share some impressions with you, Josef.

Mitad del Mundo, Right: North – Left: South

Strolling through Colonial Quito - La Compania

hiking to the Cotopaxi – Glacier and

around the Quilotoa Crater Lake,

meeting a Spectacled or Andean Bear,

encountering Giant Turtles,

Sea Birds and

Iguanas on the Galapagos Islands