

THE BULLETIN OF THE



USER GROUP

+ CAS-TI

C o n t e n t s :

- | | |
|----|--|
| 1 | Letter of the Editor |
| 2 | Editorial - Preview |
| | Josef Böhm |
| 3 | Attracted by (STRANGE) Attractors (3) |
| | Rolf Pütter |
| 23 | Ellipse Folding |
| 32 | DERIVE & CAS-TI User Forum
Leon Magiera's Improper Integral
Multiline Text Boxes for TI-Nspire |
| | Hubert Langlotz & Ralph Huspe |
| 37 | A Temperature Warner realized with TI-Innovator™ Hub |
| | Josef Böhm |
| 39 | Points and Dots in 3D Plots |

On 15 February I received a sad message sent by Peter Schorn:

“Please delete my father Richard Schorn from your mailing list. He passed away last October.”

Richard was a DUG member from the very first beginning of our activities. He was member #30 from 1991 and during his active membership he provided several contributions for our newsletter. All of us who had the privilege to meet Richard Schorn personally will never forget his wise humor and his rich mathematical and pedagogical expertise.



It's a great pleasure for me to announce a new (German) textbook for Statistics written by our DUG member Benno Grabinger.

Die didaktische Erfahrung des Autors beweist sich in einer geschickten Kombination von sauberer – aber nicht überbordenden – Theorie, anschaulichen Beispielen und motivierenden Aufgaben (95 inkl. ausführlichen Lösungen).

Schwerpunkte bilden Beschreibende Statistik und Wahrscheinlichkeitstheorie, während die Beurteilende Statistik nur so weit behandelt wird, dass man den fachspezifischen Veranstaltungen an Uni oder FH wird folgen können.

Das Buch ist an keine Technologie bzw. Software gebunden, es wird aber ein zugehöriges Paket an Excel-Dateien zum Download angeboten.

Das vorliegende Werk eignet sich ausgezeichnet zur An eignung oder Auffrischung des statistischen Grundwissens und bildet eine sichere Basis für eine weitere Beschäftigung mit statistischen Methoden.



Dear DUG Members,

It's wonderful spring time after a long winter with lots of snow in our mountains and I am very happy to have finished DNL#109.

Regrettably, our newsletter is starting with sad news. Our DUG member Richard Schorn passed away last fall. You can find his DNL-contributions in issues 5, 21, 33, 42, 45, 46, 50 and 53.

We have the third part of the "Strange Attractors" in this DNL. Working on this contribution I discovered a few possibilities to improve earlier parts. If you are interested you are invited to download the updated DNL#107 together with the respective zipped mth109.zip file.

Just two days ago I received a mail from our member David Dyer. He has a special request:

Do you have any idea where/how I can get my hands on a copy of the little red Users Manual that accompanied DERIVE? So far, I have found a copy on Amazon for \$250. I don't need one quite that desperately. I haven't been able to fish up an online copy either.

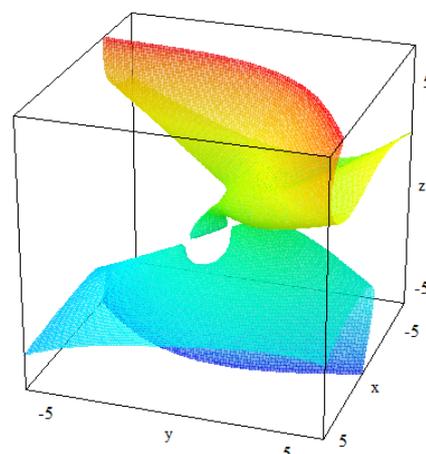
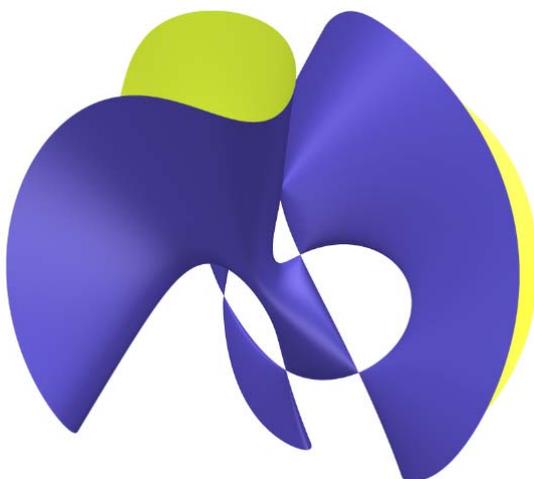
If anybody in our community could help David, then please contact me.

I am very glad that I could include a first application of the TI-Innovator Hub provided by Huber Langlotz and Ralph Huspe. Christian Zöpfl sent more examples which will be presented at next occasion. Many thanks to you.

The depicted surfaces below show two representations of a Cayley Symmetroid. They are fruits of a collaboration between CAS (DERIVE) and implicit 3D-plotting (Surfer & DPGraph).

Best regards and wishes

Josef



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:

June 2018

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
Rational Hooks, J. Lechner, AUT
Statistics of Shuffling Cards, Charge in a Magnetic Field, H. Ludwig, GER
Factoring Trinomials, D. McDougall, CAN
Selected Lectures from TIME 2016
More Applications of TI-Innovator™ Hub and TI-Innovator™ Rover
Surfaces and their Duals, Cayley Symmetroid
Affine Mappings – Treated Systematically, H. Nieder, GER
Fractal Dimension (Box Counting Method)
Three planes in space – how they can intersect
and others

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

Attracted by (STRANGE) Attractors (3)

An illustrated guided tour from well-known to unknown attractors

Your Tour Guide: Josef Böhm

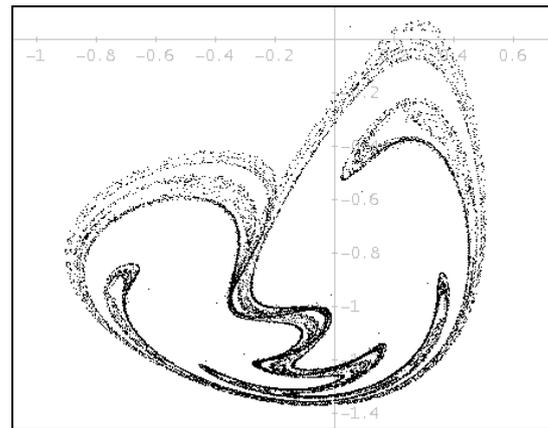
Special projections

Julien C. Sprott^[1] added a nice feature to his menu driven program: One can call various projections of the 2D-maps. He offers spherical, vertical and horizontal cylindrical and toroidal projections i.e. the graphs are projected onto the front side of the respective solids.

The challenge is to reproduce this with DERIVE!
See how it works.

We would like to project the given attractor
(20000 points) onto the solids mentioned above.

This is one of Sprott's cubic attractors:
Code: FIRPGVTFIDGCSXMFPKIDJ



Attractor to be projected

The DERIVE program is not too complicated. It is more or less the DERIVE-version of the respective part in Sprott's huge BASIC program:

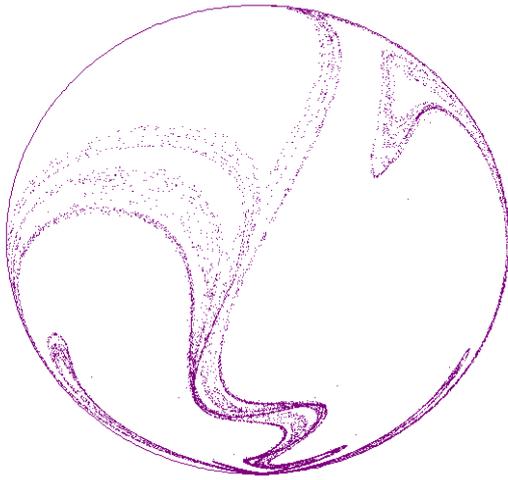
```
sph_proj(l1list, list, d, i, p1_list, pt, xmin, xmax, ymin, ymax, th, ph) :=
  Prog
  [l1list := l1list, p1_list := [x^2 + y^2 = 1]]
  d := DIM(list)
  [xmax := MAX(-1000, MAX(list↓1)), xmin := MIN(1000, MIN(list↓1))]
  [ymax := MAX(-1000, MAX(list↓2)), ymin := MIN(1000, MIN(list↓2))]
  i := 1
  Loop
  If i > d
    RETURN p1_list
  pt := list↓i
  th := π/(xmax - xmin)·(xmax - pt↓1)
  ph := π/(ymax - ymin)·(ymax - pt↓2)
  p1_list := APPEND(p1_list, [[COS(th)·SIN(ph), COS(ph)]])
  i := + 1
```

Exchanging the following lines leads to projection on a horizontal cylinder a vertical cylinder and on a torus:

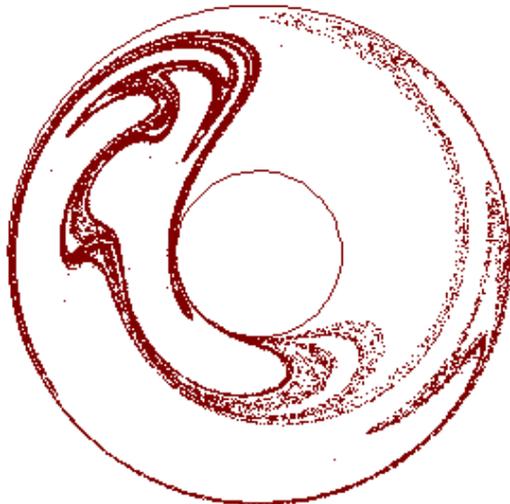
```
p1_list := [-1, 1]
p1_list := APPEND(p1_list, [(xmax - xmin)·(xmax - pt↓1), COS(ph)])

p1_list := [x = -1, x = 1]
p1_list := APPEND(p1_list, [-COS(th), -(ymax - ymin)·(ymax - pt↓2)])

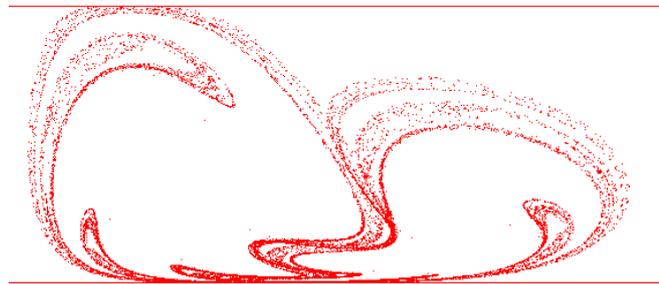
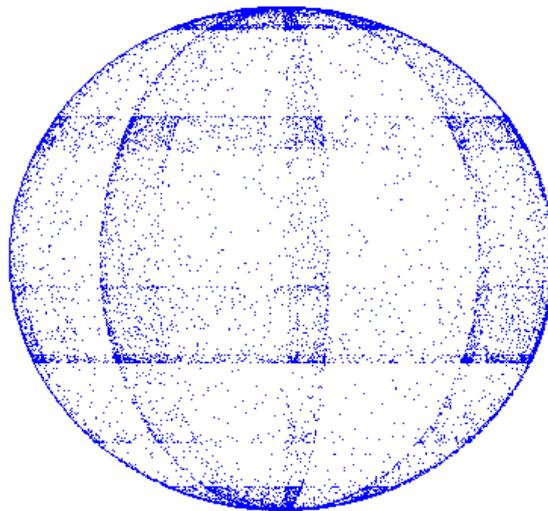
p1_list := [x^2+y^2 = 1.5^2, x^2+y^2 = 0.5^2]
p1_list := APPEND(p1_list, [(1+0.5·COS(th))·SIN(ph), (1+0.5·COS(th))·COS(ph)])
```



Projection on a sphere



Projection on a torus

Projection on a vertical cylinder (above)
and on a horizontal cylinder (below)

Spherical projection of a plaid

The plots are a little bit time consuming! But it is not too bad, just be patient.

Taking in account that we cannot plot 30000 or 40000 points and that we are satisfied with only some thousand points we can transfer the projection routines on the TI-NspireCAS without any problems:

```
sph_proj 12/12
Define sph_proj(x_l,y_l)=
Prgm
Local d,xmax,xmin,ymax,ymin,xd,yd,φ,θ,i
d:=dim(x_l)
xmax:=max(-1000,max(x_l)):xmin:=min(1000,min(x_l))
ymax:=max(-1000,max(y_l)):ymin:=min(1000,min(y_l))
xpro:={}:ypro:={}
xd:=xmax-xmin:yd:=ymax-ymin
For i,1,d
  θ:=π·(xmax-x_l[i])/xd:φ:=π·(ymax-y_l[i])/yd
  xpro:=augment(xpro,{cos(θ)·sin(φ)})
  ypro:=augment(ypro,{cos(φ)})
EndFor
Disp "Coordinates for scatter diagram in xpro and ypro"
EndPrgm
```

Start with producing the scatter diagram for the attractor and then use the created lists of coordinates for the intended projections (see the spherical projection above as an example).

```
map({0.04,1.95,-4.43,-1.46,1.81,-2.49,0.01,3.1,-3.06,-2.9,0.3,-0.21},4000)
```

Enter initial point {x0,y0} or ENTER:

Coordinates for scatter diagram in xlist and ylist

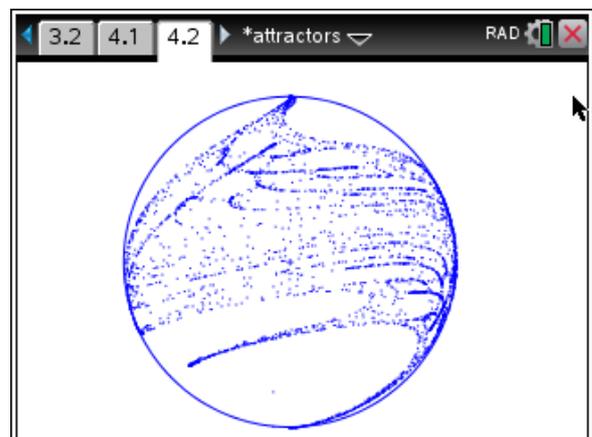
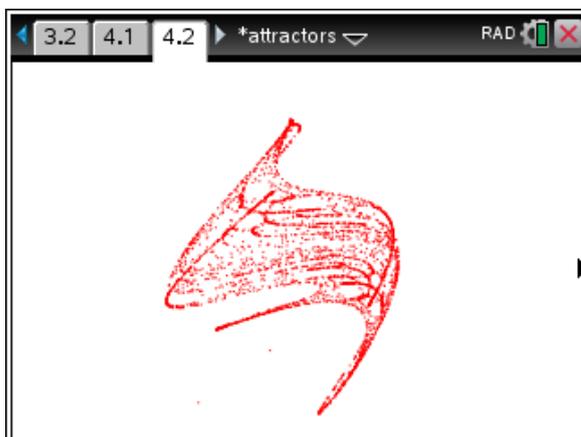
Done

```
sph_proj(xlist,ylist)
```

Coordinates for scatter diagram in xpro and ypro

Done

How the attractor and its projection on a sphere appear on the handheld screen:



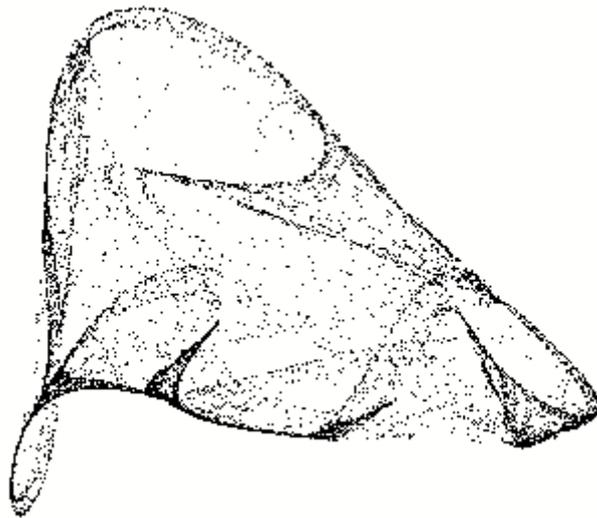
The circle must be plotted additionally in the Graph Application (right).

We make a step into the 3rd dimension

DERIVE programs make possible to “read” Sprott’s code, to plot the 3D attractor ..., and to perform own searches and discoveries.

```
sa3d(2, 500)
```

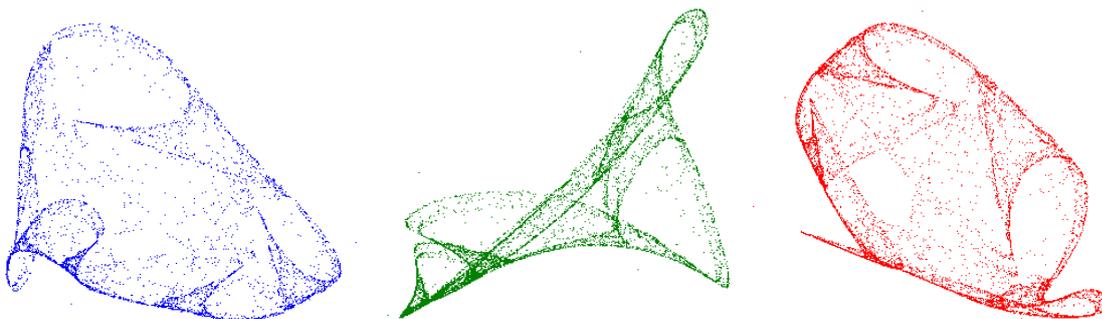
```
[ 0.056812 [0.598, 0.148, -0.494, -0.582, -0.834, -0.172, 0.806, -0.718, -0.378, 0.604,
[ 0.14289 [0.188, 0.7, -0.486, -0.56, 0.698, 0.104, 0.618, -0.42, 0.492, -0.334,
-0.594, 0.752, 0.592, -0.266, 0.782, 0.012, -0.794, -0.034, 0.408, -0.16, -0.042, -0.3
0.114, 0.442, 0.02, 0.02, 0.112, -0.244, 0.284, -0.772, 0.056, 0.796, 0.9, -0.546, 0.2
map_3d([0.188, 0.7, -0.486, -0.56, 0.698, 0.104, 0.618, -0.42, 0.492, -0.334, -0.302, -0
-0.772, 0.056, 0.796, 0.9, -0.546, 0.256, 0.882, -0.48, -0.58, -0.832], 10000)
```



A 3-dimensional attractor ($\lambda = 0.14289$)

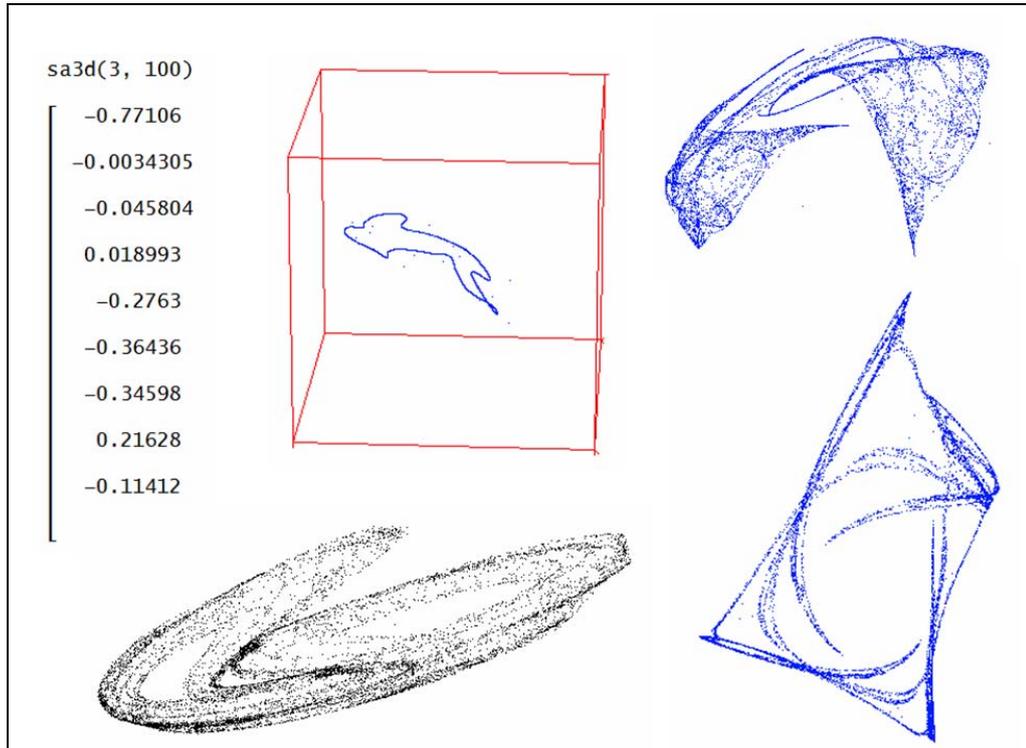
What we can do with DERIVE - and what Sprott’s awesome program is not able to do – is plotting a real 3D-graph which can be rotated. (But his program has many other features which - at least I - am unable to reproduce with DERIVE and TI-Nspire as well!!).

See below the projections of this mysterious veil from above in x -, y -, and z -direction.

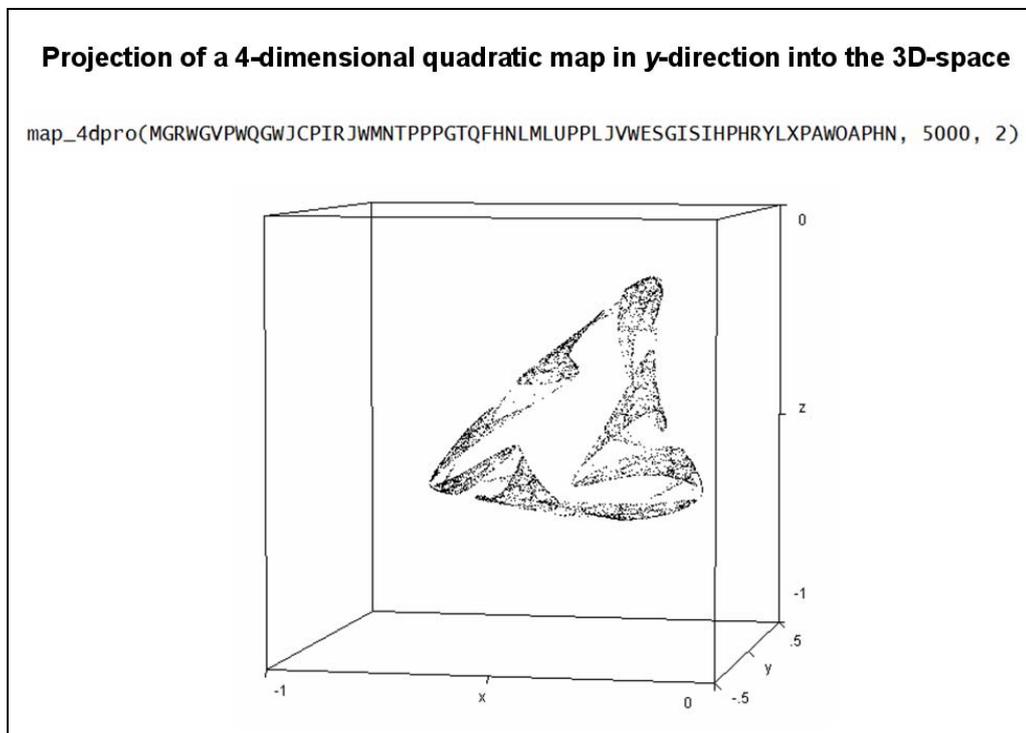


Front-, top-, and side-view of the 3D-attractor

I include two slides from my ppt-presentation:

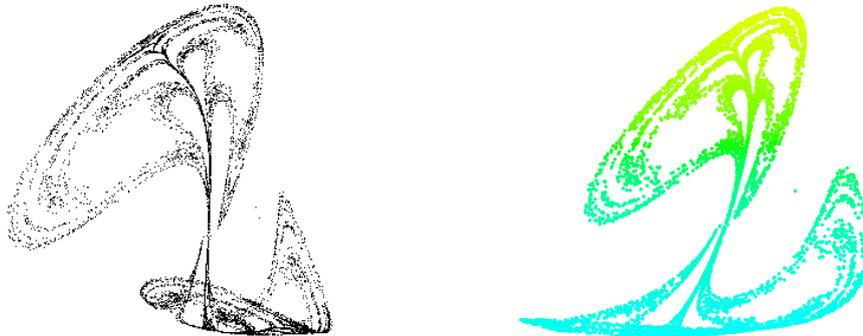


Some 3-dimensional attractors



One of Sprott's 4-dimensional attractors in a 3D projection

You can plot tiny points (recommended for a large set of them – more than 10 000) or dots (recommended for a smaller amount of data points). I will describe the difference in programming both ways in a short appendix “Points and Dots in 3D-plots” (page 39).



```
map_3d(ILURCEGOHOIQFJKBSNYGSNRUKKIKIHW, 10000)
map_3d_dots(ILURCEGOHOIQFJKBSNYGSNRUKKIKIHW, 10000)
```

I take it always as a challenge to transfer DERIVE programs to TI-NspireCAS. You can find the programs in files SA_DERIVE_3D (for DERIVE) and attractors.tns (for TI-Nspire).

```
map3d("ILURCEGOHOIQFJKBSNYGSNRUKKIKIHW",3000)
```

Enter initial point {x0,y0,z0} or ENTER:

Coordinates for scatter diagram in xlist, ylist and zlist

Done

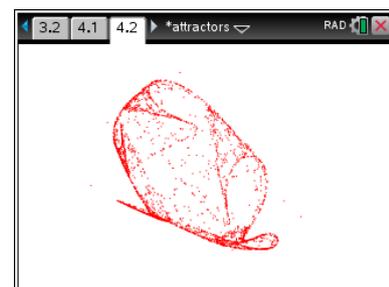
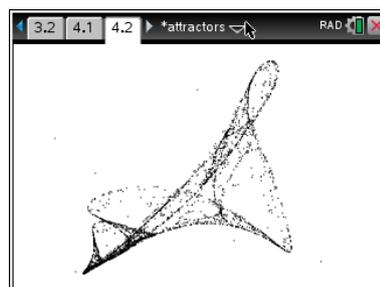
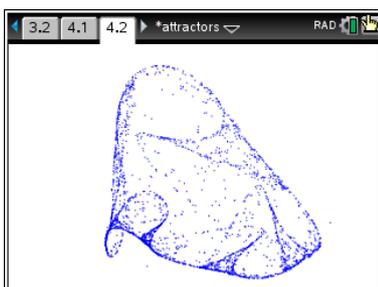
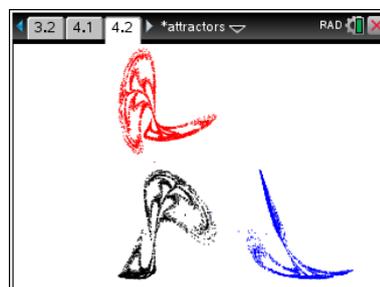
```
map3d({0.188,0.7,-0.486,-0.56,0.698,0.104,0.618,-0.42,0.492,-0.334,-0.302,-0.31,0.932,0.114,0.442,0.}
```

Enter initial point {x0,y0,z0} or ENTER:

Coordinates for scatter diagram in xlist, ylist and zlist

Done

Unfortunately, we cannot plot a 3D scatter diagram with TI-Nspire. So, we must restrict our presentation on projections in direction of the axes. See first all projections on one screen (code: "ILUR...").



Compare the projections on the handheld with the DERIVE plots on page 6.

How to search in 3rd dimension with TI-NspireCAS?

I am quite sure, that you will know, what I was doing – translating the DERIVE Code into TI-Nspire language – and it works (The coefficients in list cf are chosen between -1.3 and +1.3.)

This is the TI-NspireCAS program. Some details will be given.

Define sa3d(o,n)=

Prgm

:Local r,lsum,st0,cf,cfs,fs,i,f,g,h,xs,ys,zs,x,y,z,xe,ye,ze,xee,yee,zee,j,rs,dl3,vals,xn,yn,zn, λ

:Try

:RequestStr "Enter any pos. integer or press ENTER:",r

: If r \neq "" Then

: RandSeed expr(r)

: EndIf

: Else

: Goto next

:EndTry

:Lbl next

:If o=2 Then

: fs:={1,x,x²,x*y,x*z,y,y²,y*z,z,z²}

:EndIf

:If o=3 Then

: fs:={1,x,x²,x³,x²*y,x²*z,x*y,x*y²,x*y*z,x*z,x*z²,y,y²,y³,y²*z,
y*z,y*z²,z,z²,z³}

:EndIf

:If o=4 Then

: fs:={1,x,x²,x³,x⁴,x³*y,x³*z,x²*y,x²*y²,x²*y*z,x²*z,x²*z²,
x*y,x*y²,x*y³,x*y²*z,x*y*z,x*y*z²,x*z,x*z²,x*z³,y,y²,y³,y⁴,
y³*z,y²*z,y²*z²,y*z,y*z²,y*z³,z,z²,z³,z⁴}

:EndIf

:cfs:=dim(fs) : vals:=0

:For i,1,n

:cf:=((floor(100*(2.6*rand(3*cfs)-1.3)))/(100))

:f:=sum(left(cf,cfs)*fs):g:=sum(mid(cf,cfs+1,cfs)*fs):h:=sum(right(cf,cfs)*fs)

:lsum:=0

:st0:=iterate_3({f,g,h},{x,y,z},{0.05,0.05,0.05},5)

:xs:=st0[1]:ys:=st0[2]:zs:=st0[3]:xe:=xs+1.E-6: ye:=ys: ze:=zs

:For j,1,2000

: If abs(xs)>10 or abs(ys)>10 or abs(zs)>10 Then

: Goto nexttry

: EndIf

: xn:=f|x=x_s and y=y_s and z=z_s: yn:=g|x=x_s and y=y_s and z=z_s

: zn:=h|x=x_s and y=y_s and z=z_s: xee:=f|x=x_e and y=y_e and z=z_e

: yee:=g|x=x_e and y=y_e and z=z_e: zee:=h|x=x_e and y=y_e and z=z_e

: xe:=xee: ye:=yee: ze:=zee

```

: dl3:=(xn-xe)^(2)+(yn-ye)^(2)+(zn-ze)^(2):rs:=((1)/(10^(6)*√(dl3)))
: xs:=xn:ys:=yn:zs:=zn:xe:=xs+rs*(xe-xs):ye:=ys+rs*(ye-ys):ze:=zs+rs*(ze-zs)
: lsum:=lsum+ln(10^(12)*dl3)
:EndFor
:λ:=((((1)/(2*ln(2)))*lsum)/(j))
:If λ>0.05 Then
: vals:=1:Disp "λ="&string(λ):Disp cf
:EndIf
:Lbl nexttry
:EndFor
:If vals=0 Then
: Disp "no attractor found"
:EndIf
:EndPrgm

```

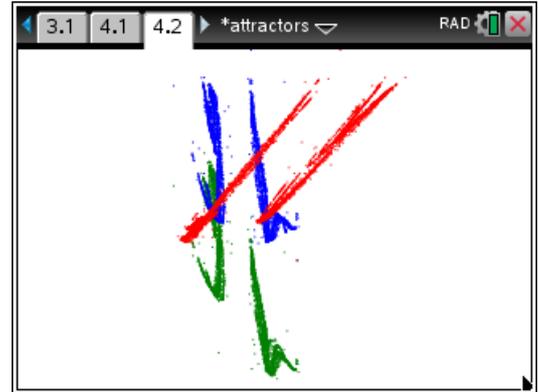
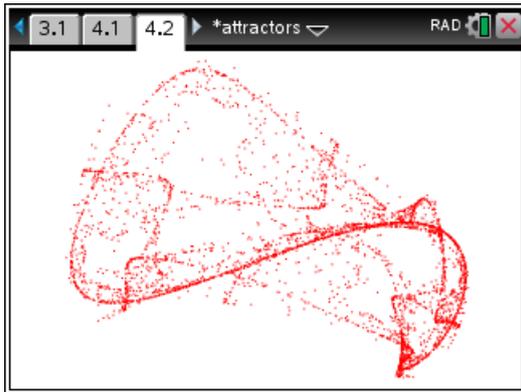
You can use the built-in initial number for the random number generator or you can enter any integer for generating the set of coefficients cf. The important part is calculating the Lyapunov Exponent λ . All sets of coefficients giving $\lambda > 0.05$ promise a – more or less – attractive attractor.

```

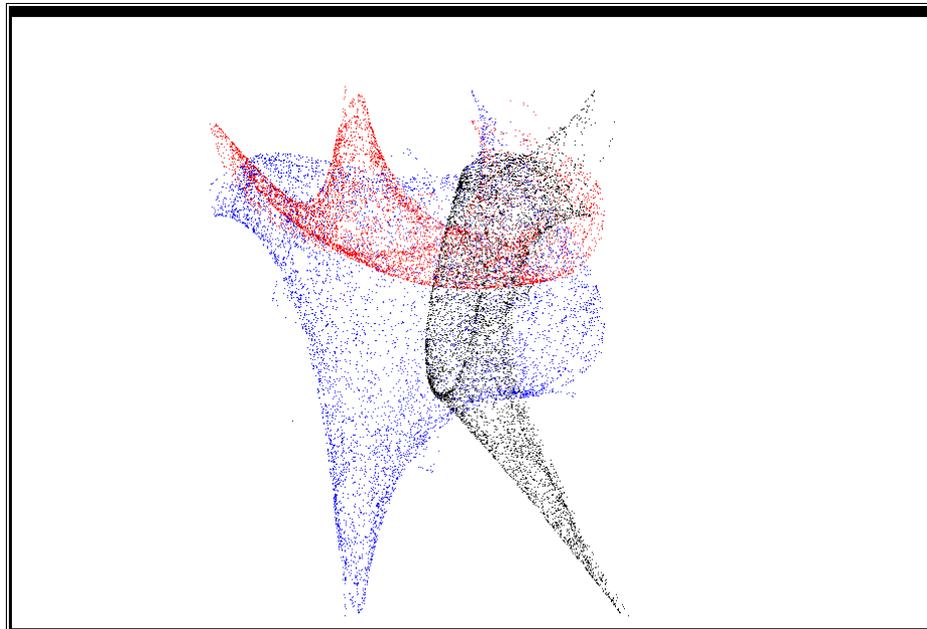
sa3d(3,1000)
Enter any pos. integer or press ENTER:
λ=0.186
{0.532,0.218,-0.238,0.583,-0.125,0.313,-0.165,0.483,0.017,0.292,0.203,-0.03,-0.527,0.803,0.763,-0.412}
λ=0.112
{-0.23,0.627,0.263,-0.648,-0.147,0.208,-0.495,0.718,-0.073,0.28,-0.732,0.207,0.608,0.343,-0.495,-0.58,0.763,-0.412}
λ=0.629
{-0.78,-0.39,-0.437,-0.348,-0.578,0.422,-0.478,-0.12,-0.072,0.265,-0.608,0.478,-0.14,-0.215,-0.352,-0.7}
λ=0.095
{0.6,-0.38,-0.658,-0.153,-0.228,-0.768,0.67,0.355,0.792,0.073,0.318,0.545,-0.39,-0.633,0.437,-0.032,0.763,-0.412}
Done
map3d({0.532,0.218,-0.238,0.583,-0.125,0.313,-0.165,0.483,0.017,0.292,0.203,-0.03,-0.527,0.803,0.763,0.412})
Enter initial point {x0,y0,z0} or ENTER:
Coordinates for scatter diagram in xlist, ylist and zlist
Done
sa3d(2,500)
Enter any pos. integer or press ENTER:

```

My very first 1000 tries gave four positive L.Es. Then I applied *map3d* in order to receive the lists of coordinates. Thus, I was able to plot a projection on a plane as a scatter diagram:



The right plot shows all possible projections of a quadratic 3D-attractor. Another attractive attractor is presented below.



Just for fun I copied the list of coefficients to DERIVE and tried to plot the attractor in form of 30000 dots (left, heat wave) and 40000 points (right). Now I could perform a true 3D-plot and move the cloud of data points around.



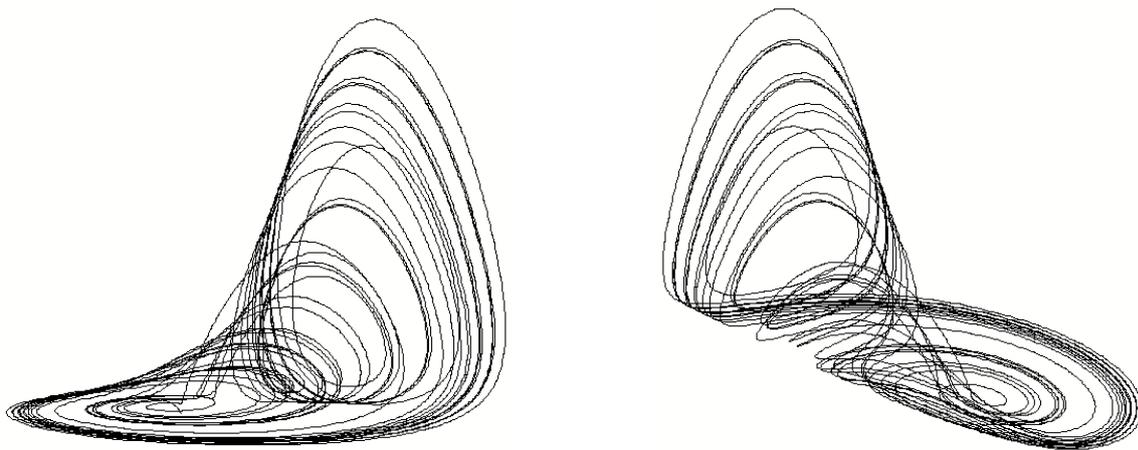
I skip presenting the procedure to create 3D-projections of 4D attractors. The DERIVE program is included in SA_DERIVE_3D.dfw.

The Rössler Attractor

Two very famous and well known 3-dimensional attractors are the Lorenz attractor and the Rössler attractor. We will skip the first one and focus on the Rössler attractor.

We receive it when solving the following system of ODEs numerically.

$$\begin{aligned}\dot{x} &= -y - z \\ \dot{y} &= x + ay \\ \dot{z} &= b + z(x - c)\end{aligned}$$



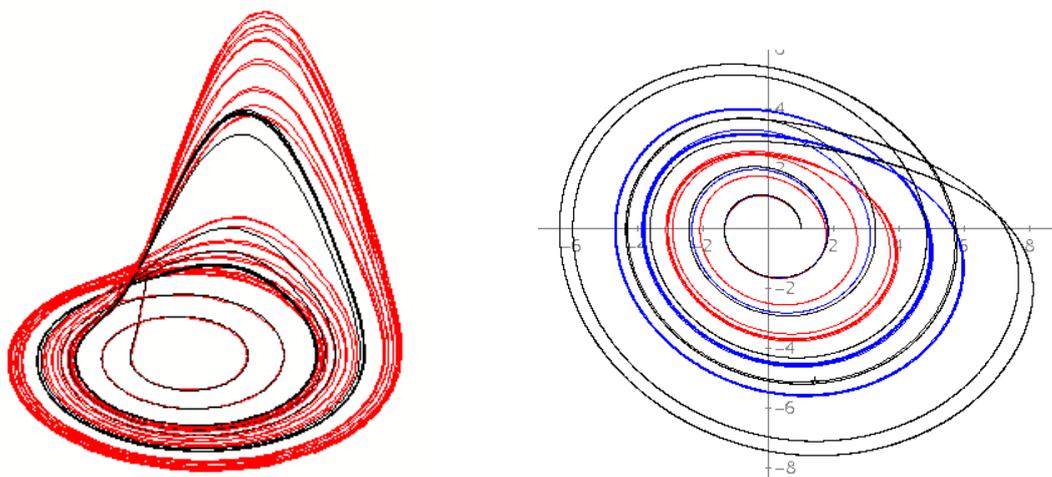
3D graph of the Rössler Attractor (DERIVE made)

I can do this using various software tools.

With DERIVE I will apply the implemented Runge-Kutta-Algorithm:

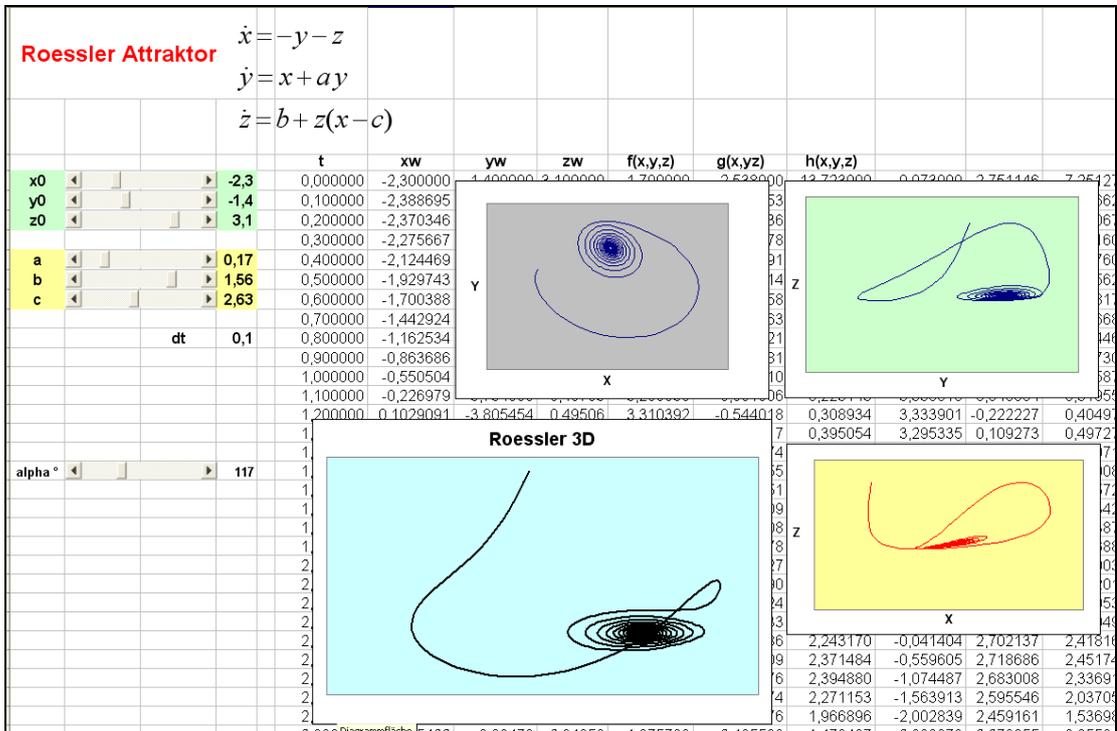
Taking the parameters as follows we have limit cycles for $c = 2, 3$ and 4 , which let us recognize the period doubling.

```
VECTOR((RK([-y - z, x + 0.2*y, 0.2 + z*(x - c)], [t, x, y, z], [0, 1, 0, 0],
0.01, 10000))↓[2, 3], c, 2, 4)
```

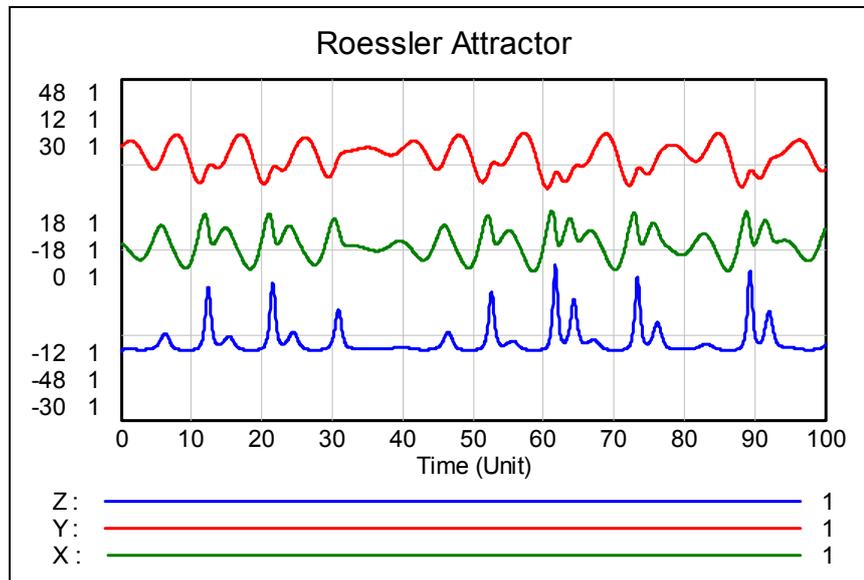


Three Rössler attractors for $c = 2, 3, 4$ (3D and top view)

This the representation of the attractor with MS Excel. As you can see I introduced sliders for the parameters a , b , and c and the initial values as well. The 3D representation is the result of a transformation into the xy -plane.



Finally I'd like to show the oscillations of x , y , and z created with the dynamic systems software VENSIM PLE.

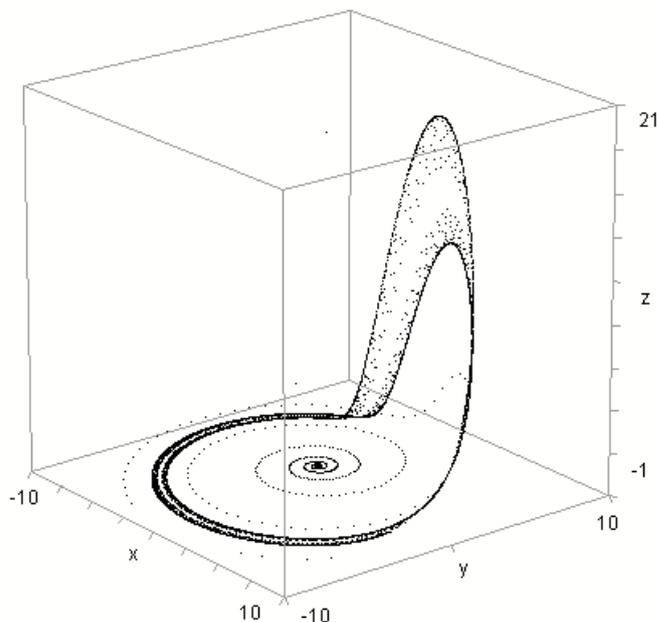


Rössler attractor with VENSIM PLE

“Attractive” systems of ODEs

Sprott mentions the Rössler attractor, too and offers it as one of an infinite number of systems of ODEs generating unexpected graphs. I transferred his program to DERIVE and plotted the attractor without Runge-Kutta as a set of 10000 not connected points.

```
map_ode([0, 0, 0, 0, 0, -1, 0, 0, -1, 0, 0, 1, 0, 0, 0, 0.15, 0, 0, 0, 0,
0.2, 0, 0, 0, 1, 0, 0, 0, -5, 0], 10000)
```



Rössler attractor as a strange attractor

Maybe that you now can imagine that I got an appetite for my own homemade ODE attractors.

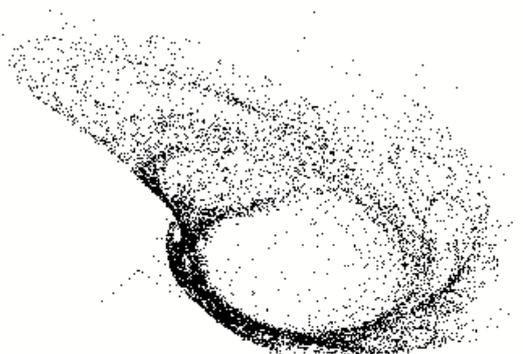
```

sa_ode(o, n, tr, t_, xs, ys, zs, xn, yn, zn, xe, ye, ze, xee, yee, zee, f, fs, g,
  Prog
  dummy := RANDOM(0)
  tr := 1
  vals := []
  cfs := (o + 1) * (o + 2) * (o + 3) / 6
  fs := [0, [1, x, x^2, x*y, x*z, y, y^2, y*z, z, z^2], [1, x, x^2, x^3, x^2*y, >
  Loop
  If tr > n exit
  WRITE([tr, DIM(vals)])
  cf := VECTOR(FLOOR(100 * (4 * RANDOM(1) - 2)) / 100, k, 3 * cfs)
  f := x + 0.1 * cf_1[1, ..., cfs] * fs_1o
  g := y + 0.1 * cf_1[cfs + 1, ..., 2 * cfs] * fs_1o
  h := z + 0.1 * cf_1[2 * cfs + 1, ..., 3 * cfs] * fs_1o
  st0 := [0.05, 0.05, 0.05]
  lsum := 0, l := 0
  st0 := ITERATE([f, g, h], [x, y, z], st0, 5)
  [xs := st0_1, ys := st0_2, zs := st0_3]
  [xe := xs + 0.000001, ye := ys, ze := zs]
  i := 1
  Loop
  If ABS(xs) > 10 v ABS(ys) > 10 v ABS(zs) > 10 exit
  If i > 300 ^ l > 0
    vals := APPEND(vals, [[1, cf]])
  If i > 300 exit
  xn := SUBST(f, [x, y, z], [xs, ys, zs])
  yn := SUBST(g, [x, y, z], [xs, ys, zs])
  zn := SUBST(h, [x, y, z], [xs, ys, zs])
  xee := SUBST(f, [x, y, z], [xe, ye, ze])
  yee := SUBST(g, [x, y, z], [xe, ye, ze])
  zee := SUBST(h, [x, y, z], [xe, ye, ze])
  [xe := xee, ye := yee, ze := zee]
  d13 := (xn - xe)^2 + (yn - ye)^2 + (zn - ze)^2
  rs := 1 / (10^6 * sqrt(d13))
  [xs := xn, ys := yn, zs := zn]
  [xe := xs + rs * (xe - xs), ye := ys + rs * (ye - ys), ze := zs + rs * (ze - zs)]
  lsum := lsum + LOG(10^12 * d13)
  l := 0.72134 * lsum / (i * 0.1)
  i := i + 1
  tr := tr + 1
  RETURN vals

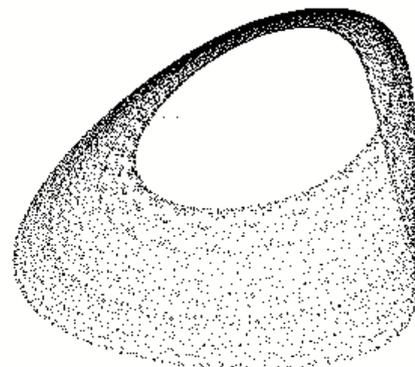
```

Above is a part of my “Search Attractive ODEs”-program with order $2 \leq o \leq 5$ testing n sets of randomly generated coefficients between -2 and +2.

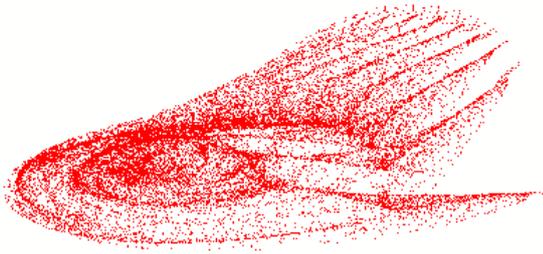
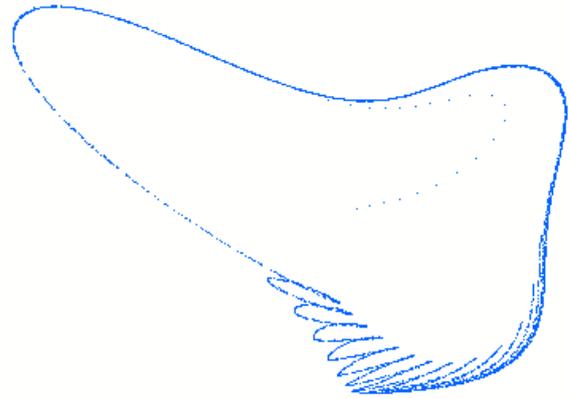
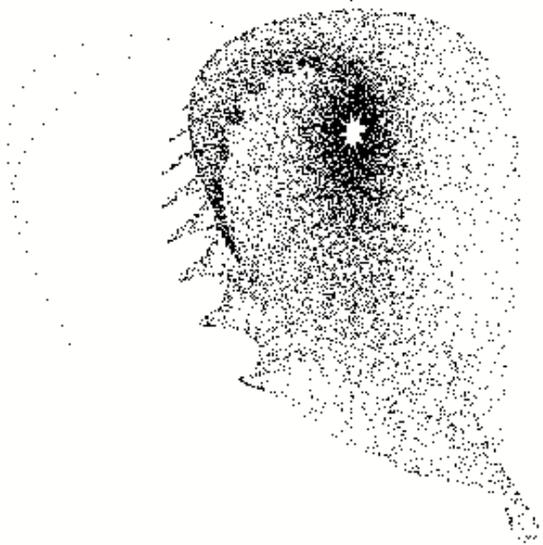
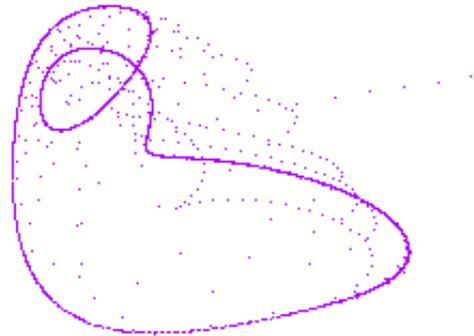
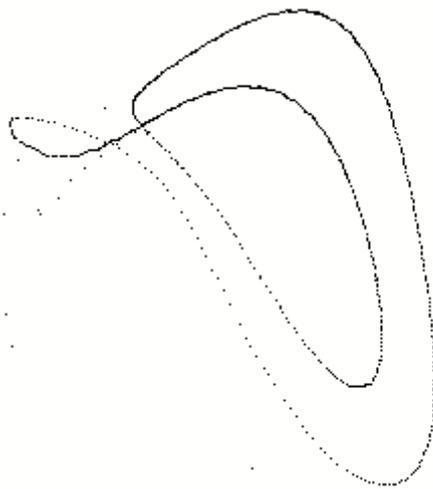
In the following will present some of my findings:



Order 2, $\lambda = 1.59$, 10000 points

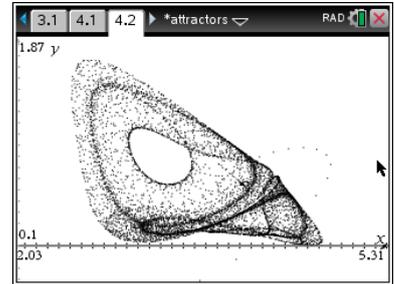
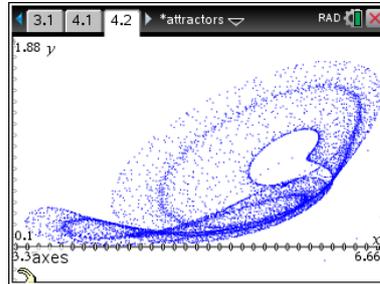
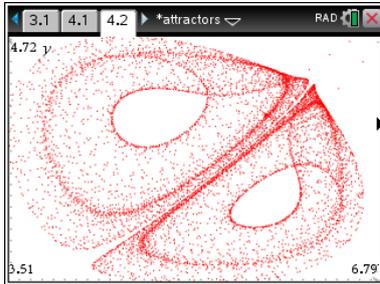


Order 2, $\lambda = 0.09$, 10000 points

Order 3, $\lambda = 1.30$, 20000 pointsOrder 3, $\lambda = 1.15$, 10000 pointsOrder 4, $\lambda = 0.886$, 10000 pointsOrder 4, $\lambda = 0.26$, 10000 pointsOrder 5, $\lambda = 0.09$, 1000 pointsOrder 5, $\lambda = 1.43$, 10000 points

TI-NspireCAS proudly presents on of Sprott's ODE-attractors:

`map_ode("QRREQDTWELEQMTMLAAPRGDJJKLPYAFO",7000)`



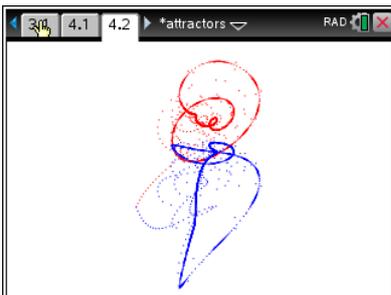
What's about our own ODE-attractors? No problem – see below:

```
map_ode({0.698,-0.715,0.598,0.733,1.165,0.993,-1.043,1.,1.228,-0.433,0.603,-1.028,-0.575,-0.025,-0.88}
Enter initial point {x0,y0,z0} or ENTER:
Coordinates for scatter diagram in xlist, ylist and zlist
Done

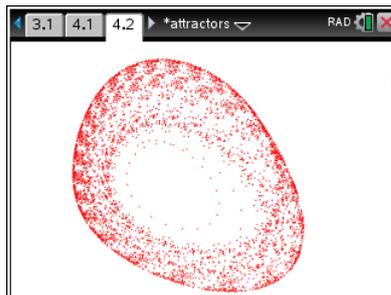
sa_ode(3,300)
Enter any pos. integer or press ENTER: 567
0.271022
{0.033000,-0.017000,-0.187000,0.108000,0.400000,0.620000,-0.503000,0.717000,0.432000,0.122000,-0
3.447196
{-0.290000,-0.623000,0.575000,-0.600000,0.800000,0.185000,0.523000,-0.283000,0.548000,0.067000,0
0.050896
{0.242000,-0.137000,0.062000,-0.115000,-0.113000,0.588000,0.310000,-0.428000,0.483000,0.422000,0
Done

map_ode({-0.29,-0.623,0.575,-0.6,0.8,0.185,0.523,-0.283,0.548,0.067,0.648,-0.643,-0.475,-0.287,-0.278}
Enter initial point {x0,y0,z0} or ENTER: {0.1,0.05,0.02}
Coordinates for scatter diagram in xlist, ylist and zlist
Done
```

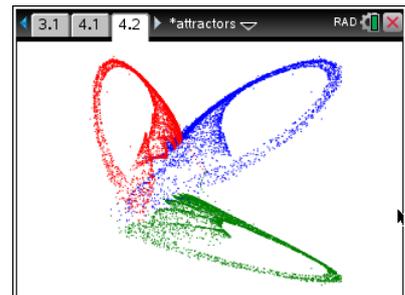
We enter `sa_ode(2≤order≤4)`, number of tries) and receive sets of coefficients together with the corresponding Lyapunov-Exponent in two separated lines. Then it is comfortable to copy the list into the `map_ode()`-program call.



Order 4, $\lambda = 0.16$ (top and side view)



Order 2, $\lambda = 0.24$



Order 3, $\lambda = 3.45$ (all views)

Gumowski-Mira-Attractors

You can find many other features offered in Sprott's book and in his SA.EXE program: attractors with shadows, 4-dimensional attractors with the 4th dimension interpreted as colour or as sound, sliced attractors, attractors as analoglyphs, and, and, and, ... Now we will leave Sprott's wonderful source of inspiration.

I will take the occasion to present some other types of recursive functions which also result in surprising graphs.

Two physicists, I. Gumowski and C. Mira found in the frame of their work at CERN in Geneva in 1980 a new type of attractors^[12, 13].

The original model (type 1) is described by

$$\begin{aligned} x_{n+1} &= b \cdot y_n + f(x_n) \\ y_{n+1} &= -x_n + f(x_{n+1}) \end{aligned} \quad \text{with } f(x) = a \cdot x + \frac{2(1-a)x^2}{1+x^2}; \quad a, b \text{ are constants.}$$

You can find many variations of the original form (see the references). Here is one of them (type 2):

$$\begin{aligned} x_{n+1} &= y_n + a \cdot (1 - b \cdot y_n^2) \cdot y_n + f(x_n) \\ y_{n+1} &= -x_n + f(x_{n+1}) \end{aligned} \quad \text{with } f(x) = \mu \cdot x + \frac{2(1-\mu)x^2}{1+x^2}; \quad a, b, \mu \text{ are constants.}$$

You can also vary function $f(x)$.

I tried to write my DERIVE-function gm(a, b, f, n, type, x0, y0) in a form to cope with both types. $type = 1$, $x0 = y0 = 0.1$ are the default values.

gm(aa, bb, f, n, type := 1, x0 := 0.1, y0 := 0.1, i, pts, xn, yn) :=

Prog

pts := [[x0, y0]]

i := 1

Loop

If i > n

RETURN pts

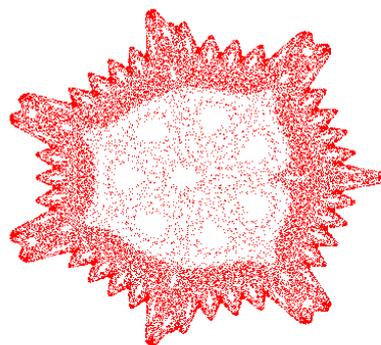
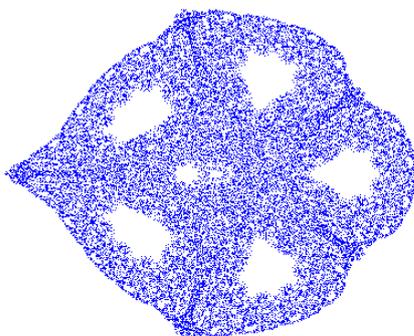
xn := ((1 - bb) · (type - 2) + 1) · y0 + (type - 1) · aa · (1 - bb · y0²) · y0 +
LIM(LIM(LIM(f, x, x0), a, aa), b, bb)

yn := -x0 + LIM(LIM(LIM(f, x, xn), a, aa), b, bb)

pts := APPEND(pts, [[xn, yn]])

[x0 := xn, y0 := yn]

i :=+ 1

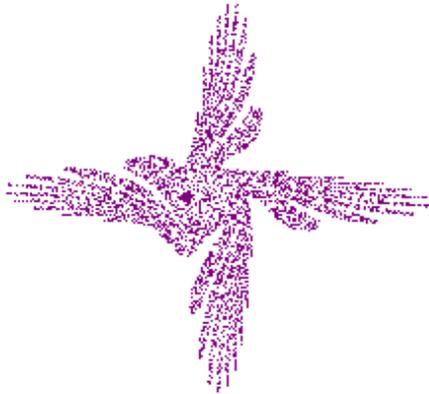


$$\text{gm} \left(0.245, 1, a \cdot x + \frac{2 \cdot (1 - a) \cdot x^2}{1 + x^2}, 30000, 1, 1, 1 \right)$$

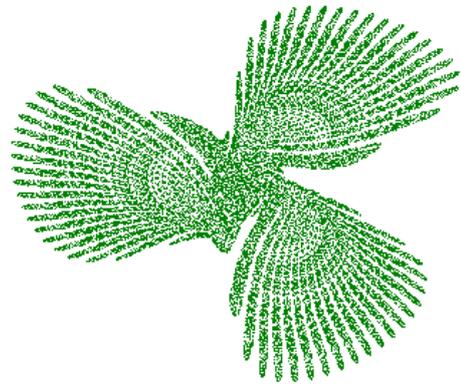
$$\text{gm} \left(-0.245, 1, a \cdot x + \frac{2 \cdot (1 - a) \cdot x^2}{1 + x^2}, 30000, 1, 1, 1 \right)$$

My first Gumowski-Mira attractors

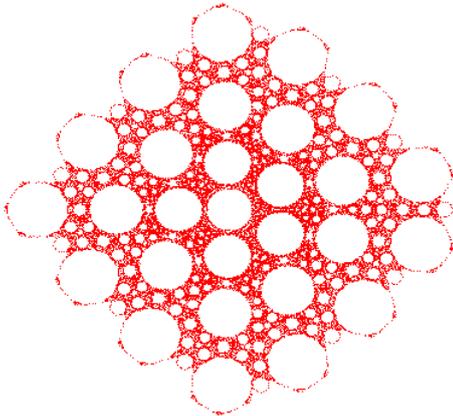
A visit in my Gumowski-Mira Gallery



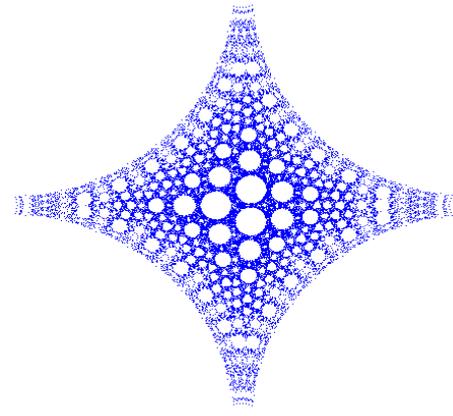
$$\text{gm} \left(0.01, 0.978, a \cdot x + \frac{2 \cdot (1 - a) \cdot x^2}{1 + x^2}, 30000, 1, 1, 1 \right)$$



$$\text{gm} \left(-0.48, 0.9924, a \cdot x + \frac{2 \cdot (1 - a) \cdot x^2}{1 + x^2}, 30000, 1, 1, 1 \right)$$



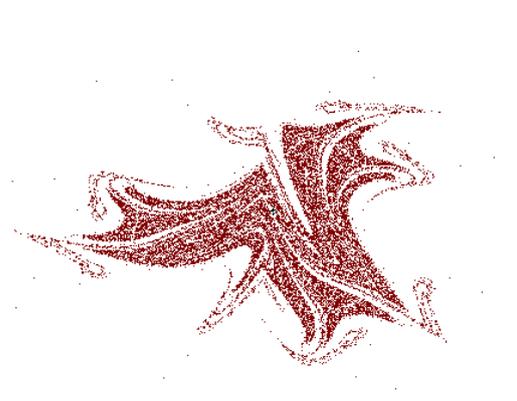
$$\text{gm} \left(0.04, 1, a \cdot x - \frac{3 - a}{b \cdot x}, 30000, 1, -2.8, 17 \right)$$



$$\text{gm} \left(0.04, 1, a \cdot x + \frac{3 - a}{b \cdot x}, 30000, 1, -2.8, 17 \right)$$



$$\text{gm} \left(0.01, 0.08, -0.105 \cdot x + \frac{2.4 \cdot 1.7 \cdot x^2}{1 + x^2}, 20000, 2 \right)$$



$$\text{gm} \left(-0.96, 0.96, a \cdot x - \left| \text{ATAN}(a - x) + \frac{b \cdot x^2}{1 + x^2} \right|, 20000, 1, 12, 9 \right)$$

This is the TI-NspireCAS version:

```

Define gm(aa,bb,f,n)=
Prgm
:Local type,start,xo,yo,xn,yn,i
:Lbl typ_
:Request "Enter type (1/2):",type
:If type≠1 and type≠2 Then
:  Goto typ_
:EndIf
:Try
:RequestStr "Enter initial point {x0,y0} or ENTER:",start
:If start≠"" Then
:  start:=expr(start)
:EndIf
:Else
:  start:={0.1,0.1}
:  Goto next
:EndTry
:Lbl next
:xo:=start[1]:yo:=start[2]:xlist:={xo}:ylist:={yo}
:For i,1,n
:If abs(xo)>100 or abs(yo)>100 Then
:  Goto end
:EndIf
:xn:=((1-bb)*(type-2)+1)*yo+(type-1)*aa*(1-bb*yo^(2))*yo+f|x=xo and a=aa and b=bb
:yn:=-xo+f|x=xn and a=aa and b=bb
:xlist:=augment(xlist,{xn}):ylist:=augment(ylist,{yn})
:xo:=xn:yo:=yn
:EndFor
:Lbl end
:Disp "Coordinates for scatter diagram in xlist and ylist"
:EndPrgm

```

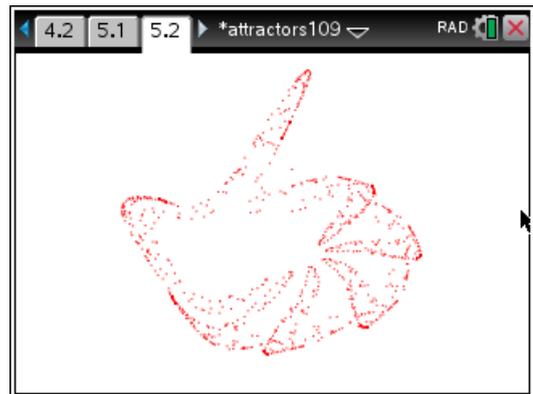
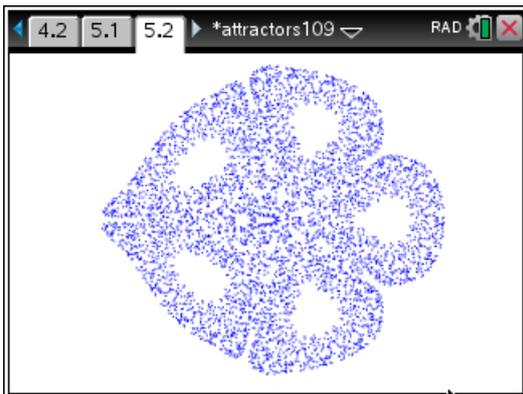
You can choose between G-M-attractors of type 1 and type 2. In case of type 2 you have to enter function $f(x)$ as third argument. See first one input example for type 1:

$$gm\left(0.245, 1, a \cdot x + \frac{2 \cdot (1-a) \cdot x^2}{1+x^2}, 5000\right)$$

Enter type (1/2): 1

Enter initial point {x0,y0} or ENTER: {1,1}

Coordinates for scatter diagram in xlist and ylist



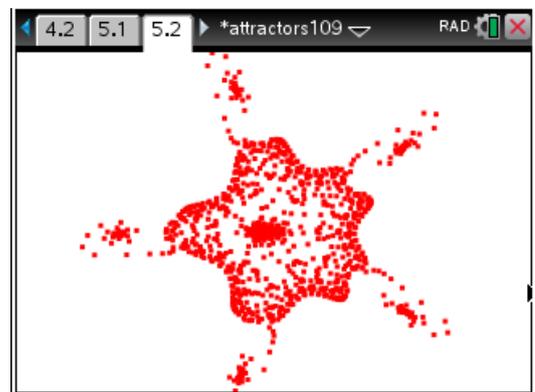
One example for GM-attractor of type 2

$$gm\left(0.008, 0.05, 0.2909 \cdot x + \frac{2 \cdot (1 - 0.2909) \cdot x^2}{1 + x^2}, 6000\right)$$

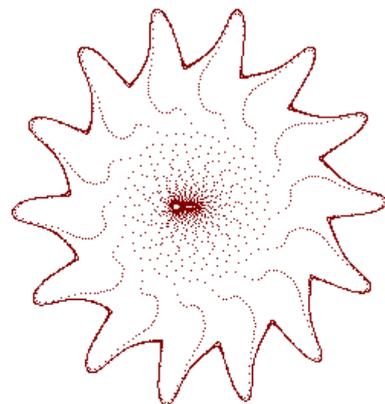
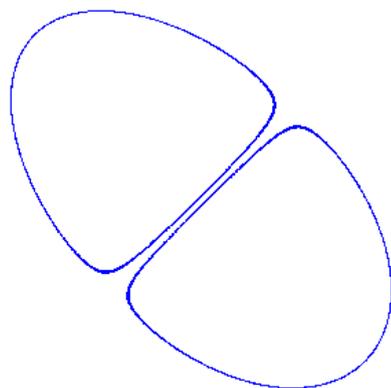
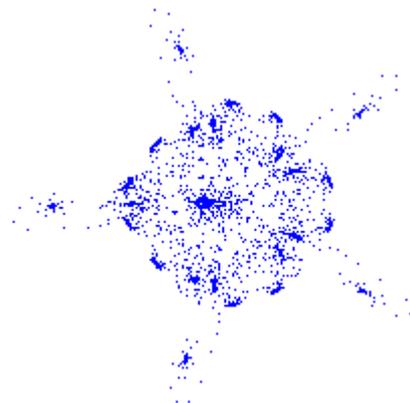
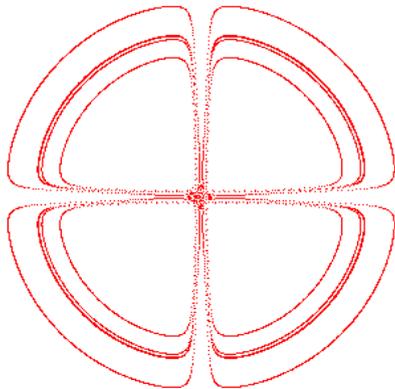
Enter type (1/2): 2

Enter initial point {x0,y0} or ENTER: {0.1,0.1}

Coordinates for scatter diagram in xlist and ylist



Another four attractors (gumowski2.dfw):



References

- [1] **Julien C. Sprott**, *Strange Attractors: Creating Patterns in Chaos*
<http://sprott.physics.wisc.edu/fractals/booktext/>
- [2],[3] DERIVE Newsletter 10 from 1993 and 13 from 1994, www.austromath.at/dug/
- [4] VENSIM PLE, Simulation software for educational purposes, download free of charge
<http://www.vensim.com/download.html>
- [5] Frank Piefke, *Simulationen mit dem Personalcomputer*, Hüttig 1991
- [6] Herbert Voß, *Chaos und Fraktale selbst programmieren*, Franzis' 1994
- [7] K.-H. Becker & M. Dörfler, *Dynamische Systeme und Fraktale*, Vieweg 1989
- [8] E. D. Schmitter, *Fraktale Geometrie*, Hofacker 1989
- [9] Clifford Pickover, *Mit den Augen des Computers*, Markt und Technik 1992
- [10] Clifford Pickover, *Computers and the Imagination*, St. Martin's Press Inc. 1991
- [11] Clifford Pickover, *Pattern, Chaos and Beauty*, Dover Publications 1989
- [12] Josef Böhm, *Dynamic Systems/Dynamische Systeme*,
<http://rfdz.ph-noe.ac.at/acdca/materialien.html>
- [13] Hans Lauwierier, *Fraktale verstehen und selbst programmieren*, Wittig Fachbuch 1989
- [14] T. Wegener & M. Peterson, *Fraktale Welten*, te-wi 1992

Recommended websites

www.math.kit.edu/iana1/~melcher/media/lyapunov-final.pdf

en.wikipedia.org/Lyapunov_exponent

mathworld.wolfram.com/LyapunovCharacteristicExponent.html

math.cmaisonneuve.qc.ca/alevesque/chaos_fract/Attracteurs/Attracteurs.html

blog.nihilogic.dk/2009/10/strange-attractors-beautiful-chaos-and.html

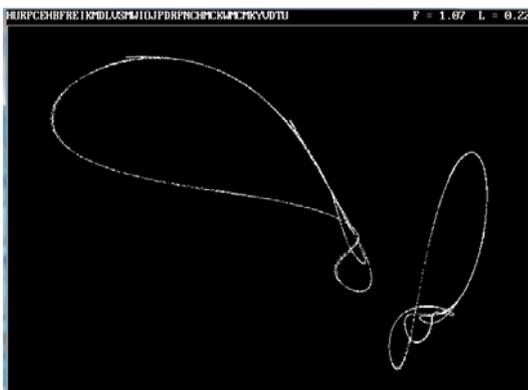
www.robert-doerner.de/Henon-System/henon-system.html

www.cc.gatech.edu/~phlosoft/attractors/

wonderfl.net/tag/Chaos

The screenshots below are two results produced by Sprott's SA.EXE which is a compiled BASIC program. It does not run under WIN 7/10. But it does work using a "DOS Box"-program.

Download Dos Box: <https://sourceforge.net/projects/dosbox/>



I recommend downloading Sprott's book text. It contains much more wonderful inspirations than I cannot present in the frame of my contribution.

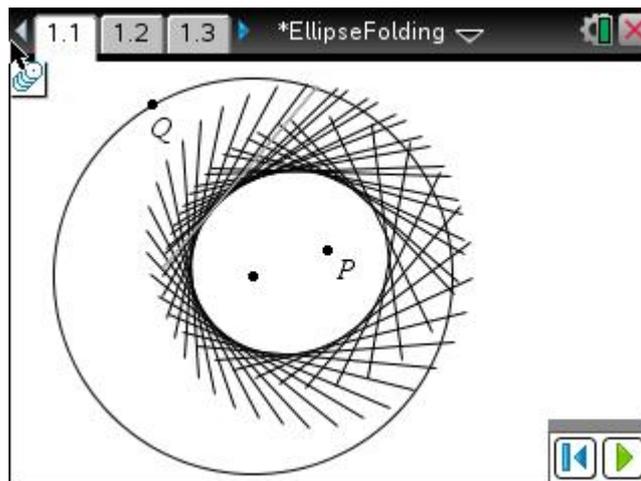
Ellipse Folding

Rolf Pütter, Germany

Martin Gardner, in „Sphere Packing, Lewis Carroll, and Reversi“, the third volume of his collected “Mathematical Games” columns, describes without proof the following method to obtain an ellipse: Cut out a circular disc from a piece of paper, mark a point P in the interior of the disc but not at its center, and then fold the disc so that its boundary falls onto P . Unfold and repeat many times with different points on the boundary. “The creases form a set of tangents that outline an ellipse.” ([G], p.193)

Let’s see if we can simulate the experiment on a TI-Nspire and provide a proof for the mathematical statement.

Construct a circle in a Geometry page of your Nspire and add a point P not at the center. Select another point Q on the circumference. The folding process can be expressed mathematically as a reflection across the perpendicular bisector of P and Q . We draw the perpendicular bisector via the “Construction” menu option. We animate Q to move on the circle via “Attributes”. We put a “Geometry Trace” on the perpendicular bisector. If we now start the animation, we get a picture like the following:



The perpendicular bisectors indeed surround a shape that looks like an ellipse. But is it actually one?

To fix ideas, we will assume that Q runs through the unit circle centered at the origin, hence $Q_a(\cos(\alpha), \sin(\alpha))$, $\alpha \in [0, 2\pi]$ and that P lies on the positive x -axis, $P(a, 0)$, $0 < a < 1$ for some fixed a .

The normal to the perpendicular bisector of P and Q_a is then given by the vector $\begin{pmatrix} \cos(\alpha) - a \\ \sin(\alpha) \end{pmatrix}$. The midpoint of the line segment PQ_a is $M(0.5 \cdot (\cos(\alpha) + a), 0.5 \cdot \sin(\alpha))$.

The equation for the perpendicular bisector will be

$$x \cdot (\cos(\alpha) - a) + y \cdot \sin(\alpha) = d$$

for some $d \in \mathbb{R}$. Plugging in the coordinates of M for x and y we get

$$d = 0.5 \cdot (\cos^2(\alpha) - a^2 + \sin^2(\alpha)) = 0.5 \cdot (1 - a^2).$$

Hence, we get for the perpendicular bisector g_α :

$$(1) \quad g_\alpha: \quad x \cdot (\cos(\alpha) - a) + y \cdot \sin(\alpha) - 0.5 \cdot (1 - a^2) = 0$$

If we set

$$f(x, y, \alpha) = x \cdot (\cos(\alpha) - a) + y \cdot \sin(\alpha) - 0.5 \cdot (1 - a^2),$$

the family of perpendicular bisectors will be described by the equation

$$(2) \quad f(x, y, \alpha) = 0, \quad x \in \mathbb{R}, y \in \mathbb{R}, \alpha \in [0, 2\pi]$$

We define an *envelope* of the family of straight lines (1) to be a curve E that touches a member of the family at each of its points. Our goal is now to prove that the envelope of (1) is an ellipse. If we parameterize E as

$$x = x(\alpha), \quad y = y(\alpha),$$

insertion into (2) yields

$$f(x(\alpha), y(\alpha), \alpha) = 0.$$

Differentiating with respect to α , we obtain

$$f_x(x(\alpha), y(\alpha), \alpha) \cdot \frac{dx}{d\alpha} + f_y(x(\alpha), y(\alpha), \alpha) \cdot \frac{dy}{d\alpha} + f_\alpha(x(\alpha), y(\alpha), \alpha) = 0$$

The condition that E be tangent to the family of straight lines is

$$f_x(x, y, \alpha) \cdot \frac{dx}{d\alpha} + f_y(x, y, \alpha) \cdot \frac{dy}{d\alpha} = 0$$

for $\begin{pmatrix} f_x(x, y, \alpha) \\ f_y(x, y, \alpha) \end{pmatrix}$ is normal to the level line $f(x, y, \alpha) = 0$.

Taken together, the last two equations yield

$$(3) \quad f_\alpha(x, y, \alpha) = 0.$$

On our way from (2) to (3), we have followed the lines of thought presented in vol. 2 of Courant's "*Introduction to Calculus and Analysis*", see [C], p.290 ff. His approach is more general insofar as $f(x, y, \alpha)$ may describe any differentiable family of curves, not necessarily straight lines.

In our case, equations (2) and (3) take the form

$$\begin{aligned} x \cdot (\cos(\alpha) - a) + y \cdot \sin(\alpha) - 0.5 \cdot (1 - a^2) &= 0, \\ -x \cdot \sin(\alpha) + y \cdot \cos(\alpha) &= 0. \end{aligned}$$

The second equation can be converted to $\frac{x}{\cos(\alpha)} = \frac{y}{\sin(\alpha)}$, hence there exists an $r(\alpha) \in \mathbb{R}$ such that

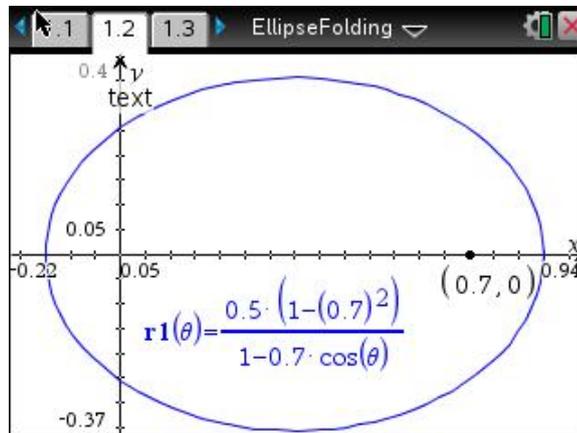
$$(4) \quad x = r(\alpha) \cdot \cos(\alpha), \quad y = r(\alpha) \cdot \sin(\alpha)$$

Insert (4) into (1) to get

$$\begin{aligned} r(\alpha) \cdot (\cos^2(\alpha) - a \cdot \cos(\alpha) + \sin^2(\alpha)) - 0.5 \cdot (1 - a^2) &= 0, \\ r(\alpha) \cdot (1 - a \cdot \cos(\alpha)) &= 0.5 \cdot (1 - a^2) \end{aligned}$$

$$(5) \quad r(\alpha) = \frac{0.5 \cdot (1 - a^2)}{1 - a \cdot \cos(\alpha)}$$

This is the representation of an ellipse E in polar coordinates centered at its left focal point with its major axis lying on the x -axis. See the Wikipedia article on the ellipse and the snapshot of page 1.2; a has been chosen as 0.7:



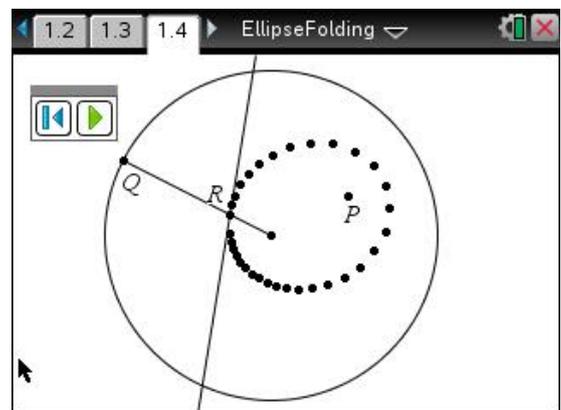
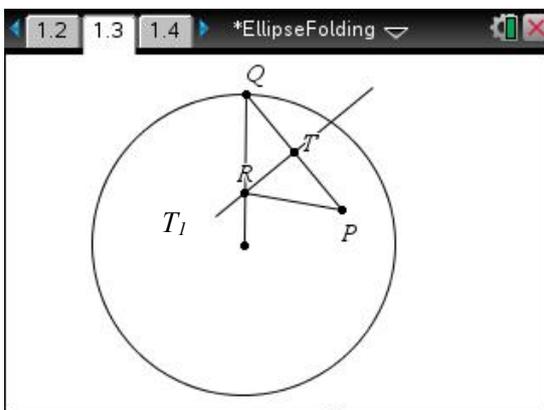
The two vertices of this ellipse are $V_1(0.5 \cdot (-1 + a), 0)$, $V_2(0.5 \cdot (1 + a), 0)$, so that their distance is 1 and the major semi-axis of E has length $\frac{1}{2}$. The tangent to E is horizontal if Q_α is positioned perpendicularly above $P(a, 0)$, that is, for $Q_\alpha(a, \sqrt{1 - a^2})$. Hence the minor semi-axis measures $\frac{1}{2}\sqrt{1 - a^2}$.

That E is an ellipse can also be seen by elementary geometric means. Remember g_α to be the perpendicular bisector of Q_α and P . By equations (1) and (4) we see that each point on E is the intersection of a radius OQ_α and the associated g_α :

$$(6) \quad \forall R \text{ on } E \exists \alpha \in [0, 2\pi] \text{ such that } R = OQ_\alpha \cap g_\alpha.$$

Let T be the point of intersection of g_α and PQ_α . Triangles $\Delta(RTQ_\alpha)$ and $\Delta(RPT)$ are congruent, as Q_α , under reflection across g_α , goes to P , whereas R and T remain fixed. Hence $\overline{RQ_\alpha} = \overline{PR}$, $\overline{OR} + \overline{PR} = 1$. The sum of the lengths of the line segments from a point on E to O and P is constant, hence E is an ellipse with foci O and P .

If we animate Q again and put a trace on R , we get



The dots sweep out a curve which we now know to be an ellipse.

So, the folding process evolved into an example for the more general concept of the envelope of a family of curves, leading to equations (2) and (3). The two representations (5) and (6) for the ellipse then followed readily. The TI-Nspire Geometry application yielded the corresponding pictures with a minimum of effort.

Bibliography:

[C] Courant, Richard & John, Fritz. *Introduction to Calculus and Analysis, vol.2*. Wiley, New York 1974.

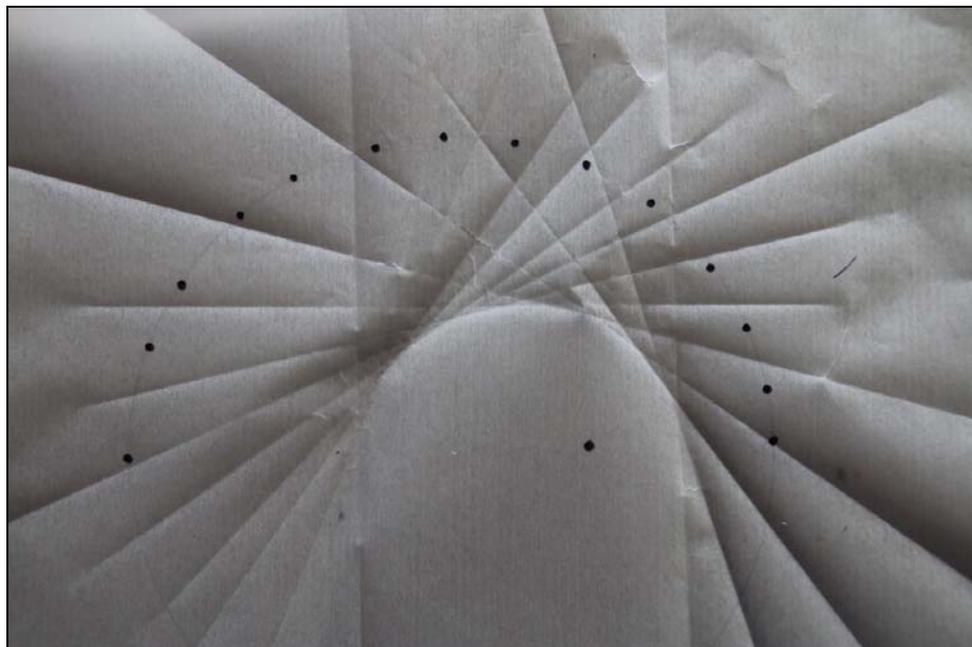
[G] Gardner, Martin. *Sphere Packing, Lewis Carroll, and Reversi*. Cambridge University Press, New York 2009.

Josef's Appendix

I took the headline of Rolf Pütter's contribution literally and wanted to create an ellipse by folding a paper according to Gardener's instructions. I draw a circle on a sheet of trace paper and marked some points on its circumference and marked one other point inside of the circle. Then I folded the paper that one points on the circumference after the other came to lie on the interior point. By and by the folding lines showed the contour of an ellipse.

I believe that this could be a nice approach for students. Take different distances of the interior point to the circumference and discuss its influence on the form of the ellipse.

Later on, I will perform another folding experiment – maybe that the students will have this idea on their own ...



First of all I had the idea to reproduce the dynamics geometry with DERIVE using appropriate slider bars for all parameters of importance. Then I combined this intention with another approach to the ellipse –applying hidden calculus (However, R. Pütter's explanation using the classical definition of an ellipse is the easiest one). But let us do some computer algebra ...

#1: CaseMode := Sensitive

Slider bars for r, a, x_ with $0 \leq a, r \leq 5$, $-5 \leq x_ \leq 5$

$$\#2: x^2 + y^2 = r^2$$

$$\#3: [a, 0]$$

$$\#4: [x_, \sqrt{(r^2 - x_^2)}]$$

$$\#5: \begin{bmatrix} a & 0 \\ x_ & \sqrt{(r^2 - x_^2)} \end{bmatrix}$$

$$\#6: \begin{bmatrix} 0 & 0 \\ x_ & \sqrt{(r^2 - x_^2)} \end{bmatrix}$$

Plot the circle, point P (#3), PQ (#5) and OQ (#6).
Perpendicular bisector of P(a,0) and Q(x_,y_)
(Q = point on circumference of circle (0,0;r)).

$$\#7: m := y - \frac{y_-}{2} = \frac{a - x_-}{y_-} \cdot \left(x - \frac{a + x_-}{2} \right)$$

Ray [O(0,0),Q(x_,y_)]

$$\#8: s := y = \frac{y_-}{x_-} \cdot x$$

Substitute $y_- = \sqrt{(r^2 - x_-^2)}$ in m and plot the perpendicular bisector.

$$\#9: pb := \text{SUBST}(m, y_-, \sqrt{(r^2 - x_-^2)})$$

$$\#10: pb := y - \frac{\sqrt{(r^2 - x_-^2)}}{2} = \frac{(a - x_-) \cdot (2 \cdot x - a - x_-)}{2 \cdot \sqrt{(r^2 - x_-^2)}}$$

Intersection point $m \cap s \rightarrow$ point R;

Find the locus of R when Q moves on the circumference of the circle.

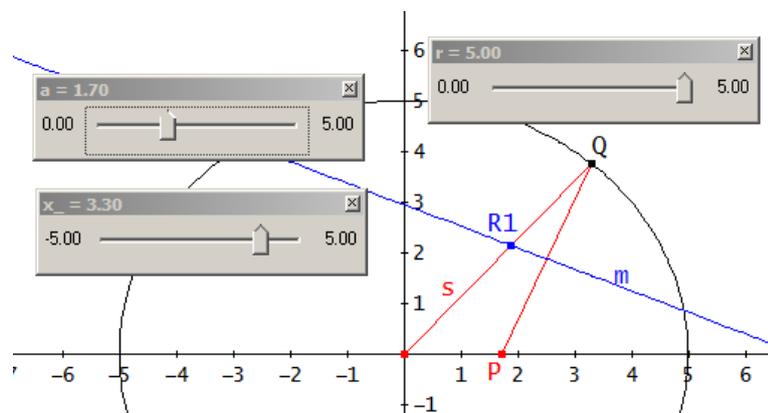
$$\#11: R := (\text{SOLUTIONS}(m \wedge s, [x, y]))_1$$

$$\#12: R := \left[\frac{x_- \cdot (a^2 - x_-^2 - y_-^2)}{2 \cdot (a \cdot x_- - x_-^2 - y_-^2)}, \frac{y_- \cdot (a^2 - x_-^2 - y_-^2)}{2 \cdot (a \cdot x_- - x_-^2 - y_-^2)} \right]$$

$$\#13: R1 := \left[\frac{x_{\perp} \cdot (a^2 - x_{\perp}^2 - \sqrt{(r^2 - x_{\perp}^2)})}{2 \cdot (a \cdot x_{\perp} - x_{\perp}^2 - \sqrt{(r^2 - x_{\perp}^2)})}, \frac{\sqrt{(r^2 - x_{\perp}^2)} \cdot (a^2 - x_{\perp}^2 - \sqrt{(r^2 - x_{\perp}^2)})}{2 \cdot (a \cdot x_{\perp} - x_{\perp}^2 - \sqrt{(r^2 - x_{\perp}^2)})} \right]$$

$$\#14: R1 := \left[\frac{x_{\perp} \cdot (a^2 - r^2)}{2 \cdot (a \cdot x_{\perp} - r^2)}, \frac{(a^2 - r^2) \cdot \sqrt{(r^2 - x_{\perp}^2)}}{2 \cdot (a \cdot x_{\perp} - r^2)} \right]$$

Now plot point R1, too and vary the parameters.
Observe the trace of point R1 and the positions of the bisector.
(Only in the upper semiplane.)

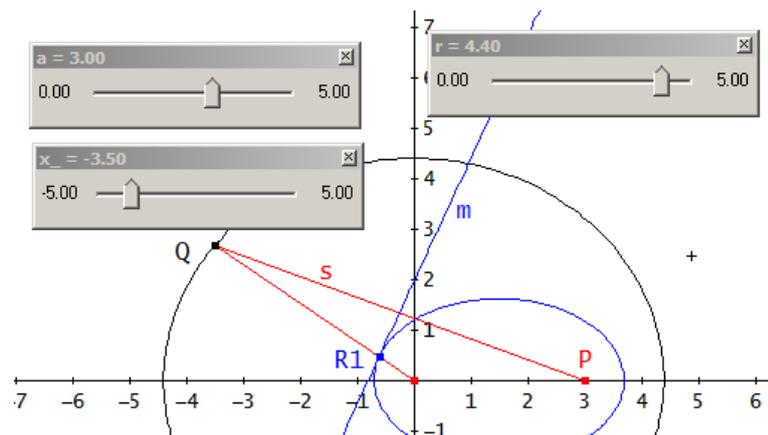


We will eliminate parameter x_{\perp} in order to obtain the requested locus of R:

$$\#15: \text{SOLUTIONS} \left(x = \frac{x_{\perp} \cdot (a^2 - r^2)}{2 \cdot (a \cdot x_{\perp} - r^2)}, x_{\perp} = \frac{2 \cdot r^2 \cdot x}{2 \cdot a \cdot x - a^2 + r^2} \right)$$

$$\#16: \text{SUBST} \left(y = \frac{(a^2 - r^2) \cdot \sqrt{(r^2 - x_{\perp}^2)}}{2 \cdot (a \cdot x_{\perp} - r^2)}, x_{\perp} = \frac{2 \cdot r^2 \cdot x}{2 \cdot a \cdot x - a^2 + r^2} \right)$$

$$\#17: y = \frac{(a^2 - r^2) \cdot (4 \cdot x^2 - 4 \cdot a \cdot x + a^2 - r^2)}{4 \cdot r^2}$$



Next attempt (using trig functions):
 substitute $x = r \cos(t)$ and $y = r \sin(t)$ in #11,
 call it R2 and simplify

$$\#18: R2 := \left[\frac{(r \cdot \cos(t)) \cdot (a^2 - (r \cdot \cos(t))^2 - (r \cdot \sin(t))^2)}{2 \cdot (a \cdot (r \cdot \cos(t)) - (r \cdot \cos(t))^2 - (r \cdot \sin(t))^2)}, \right. \\ \left. \frac{(r \cdot \sin(t)) \cdot (a^2 - (r \cdot \cos(t))^2 - (r \cdot \sin(t))^2)}{2 \cdot (a \cdot (r \cdot \cos(t)) - (r \cdot \cos(t))^2 - (r \cdot \sin(t))^2)} \right]$$

$$\#19: R2 := \left[\frac{(a^2 - r^2) \cdot \cos(t)}{2 \cdot (a \cdot \cos(t) - r)}, \frac{(a^2 - r^2) \cdot \sin(t)}{2 \cdot (a \cdot \cos(t) - r)} \right]$$

Now we have to eliminate parameter t .
 (This is not so comfortable as eliminating x above.)

$$\#20: \text{SOLUTIONS} \left(x = \frac{(a^2 - r^2) \cdot \cos(t)}{2 \cdot (a \cdot \cos(t) - r)}, t \right)$$

$$\#21: \left[-\text{ACOS} \left(\frac{2 \cdot r \cdot x}{2 \cdot a \cdot x - a^2 + r^2} \right), \text{ASIN} \left(\frac{2 \cdot r \cdot x}{2 \cdot a \cdot x - a^2 + r^2} \right) + \frac{3 \cdot \pi}{2}, \text{ACOS} \left(\frac{2 \cdot r \cdot x}{2 \cdot a \cdot x - a^2 + r^2} \right) \right]$$

We take the first solution and proceed in order to obtain the locus

$$\#22: \text{SUBST} \left(y = R2, t, \text{ACOS} \left(\frac{2 \cdot r \cdot x}{2 \cdot a \cdot x - a^2 + r^2} \right) \right)$$

$$\#23: y = \frac{\sqrt{((a^2 - r^2) \cdot (4 \cdot x^2 - 4 \cdot a \cdot x + a^2 - r^2)) \cdot \text{SIGN}(2 \cdot a \cdot x - a^2 + r^2)}}{2 \cdot r}$$

We can set $\text{SIGN}(2ax - a^2 + r^2) = 1 \dots$

$$\#24: (2 \cdot r \cdot y = \sqrt{((a^2 - r^2) \cdot (4 \cdot x^2 - 4 \cdot a \cdot x + a^2 - r^2))})$$

$$\#25: 4 \cdot r \cdot y^2 = (a^2 - r^2) \cdot (4 \cdot x^2 - 4 \cdot a \cdot x + a^2 - r^2)$$

... and we identify the expression as a conic.

$$\#26: 4 \cdot r \cdot y^2 = -4 \cdot r \cdot x^2 + 4 \cdot a \cdot x^2 + 4 \cdot a \cdot r \cdot x - 4 \cdot a^2 \cdot x + r^3 - 2 \cdot a \cdot r^2 + a^4$$

$$\#27: 4 \cdot r \cdot y^2 + 4 \cdot r \cdot x^2 - 4 \cdot a \cdot x^2 - 4 \cdot a \cdot r \cdot x + 4 \cdot a^2 \cdot x = r^3 - 2 \cdot a \cdot r^2 + a^4$$

$$\#28: \frac{4 \cdot r \cdot y^2 + 4 \cdot r \cdot x^2 - 4 \cdot a \cdot x^2 - 4 \cdot a \cdot r \cdot x + 4 \cdot a^2 \cdot x = r^3 - 2 \cdot a \cdot r^2 + a^4}{4}$$

Showing that the perpendicular bisector is a tangent is easy done – if the students remember the attribute of an ellipse, that is that the tangent is the angle bisector of the focal rays OR and PR , which can be seen in the construction because TR is the altitude in the isosceles triangle $\Delta(PRQ)$.

$$\#29: x^2 \cdot (r^2 - a^2) + a \cdot x \cdot (a^2 - r^2) + r^2 \cdot y^2 = \frac{a^4 - 2 \cdot a^2 \cdot r^2 + r^4}{4}$$

$$\#30: \frac{r^2 - a^2}{4} \cdot \left(x - \frac{a}{2}\right)^2 + \frac{r^2}{4} \cdot y^2 = \frac{r^2 \cdot (r^2 - a^2)}{16}$$

Ellipse Center $(a/2, 0)$, horizontal (major) axis = $r/2$,
vertical (minor) axis = $\sqrt{(r^2 - a^2)}/2$

You are invited to compare with expression #16!!

We show that the perpendicular bisector m of PQ is the tangent on the ellipse in R ($pb = tr$):

The explicit form of $m = pb$:

$$\#31: y = (\text{SOLUTIONS}(pb, y))_1 = \left(y = \frac{2 \cdot x \cdot (a - x) - a^2 + r^2}{2 \cdot \sqrt{(r^2 - x^2)}} \right)$$

Let's assume that the students are fixed in their thoughts in analytic geometry and they are happy to remember the formula of the ellipse tangent (hidden calculus!!):

The tangent of the ellipse in point R :

$$\#32: tr := (r^2 - a^2) \cdot \left(x - \frac{a}{2}\right) \cdot \left(R1_1 - \frac{a}{2}\right) + r^2 \cdot y \cdot R1_2 = \frac{r^2 \cdot (r^2 - a^2)}{4}$$

$$\#33: tr := \frac{r^2 \cdot (a^2 - r^2) \cdot (2 \cdot x \cdot (a - x) - 2 \cdot y \cdot \sqrt{(r^2 - x^2)} - a \cdot (a - x))}{4 \cdot (r^2 - a \cdot x)} = \frac{r^2 \cdot (r^2 - a^2)}{4}$$

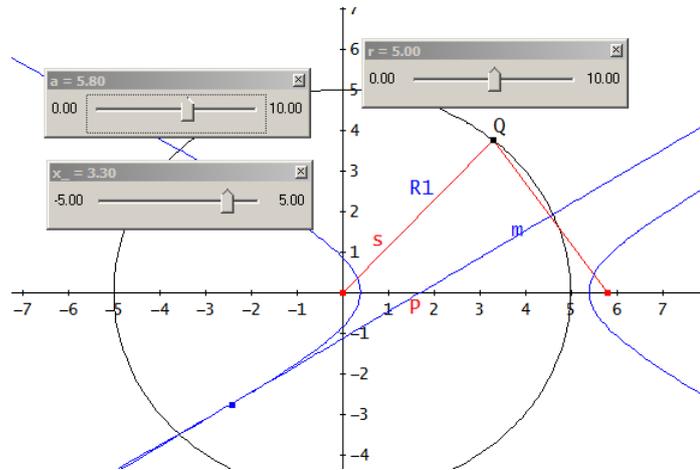
in its explicit form:

$$\#34: y = (\text{SOLUTIONS}(tr, y))_1 = \left(y = \frac{2 \cdot x \cdot (a - x) - a^2 + r^2}{2 \cdot \sqrt{(r^2 - x^2)}} \right)$$

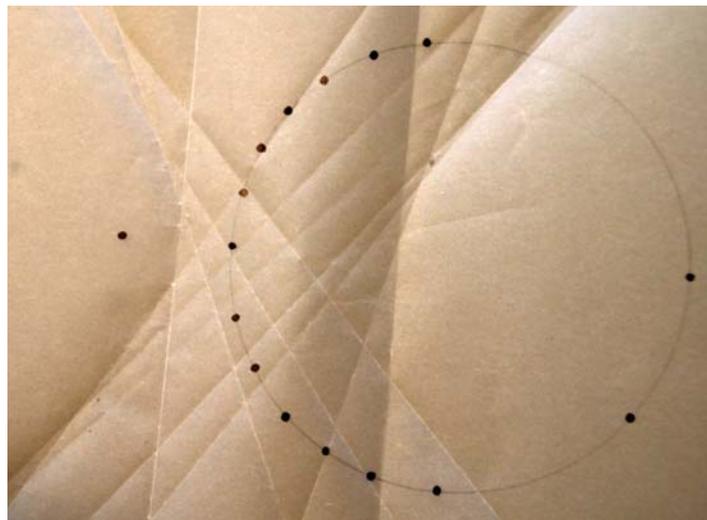
Compare #31 and #34

Then I was ready with folding an ellipse and played around with my DERIVE and Nspire dynamic geometry models:

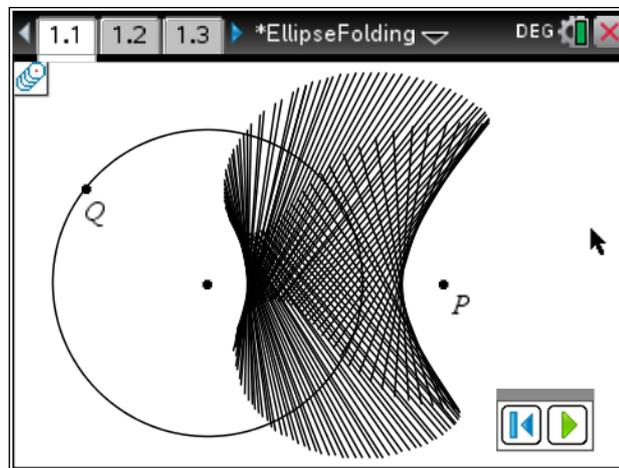
What will happen when point be is lying outside of the circle? Gussed right: a hyperbola is appearing.



Why not folding a hyperbola?



And finally, we will return to the start activity and produce a family of perpendicular bisectors:



Rolf Pütter wrote that “Folding a Hyperbola” was very surprising for him. Many thanks to him for his motivation to investigate further, Josef

Leon Magiera, Poland, posed a question in DNL#108:

Hello Josef,

I have the following question: How to calculate the following integral (with $a > 0$)?

$$\int_0^{\infty} \frac{x^2}{(\cosh(x) + 1) \cdot (\cosh(x) + a)} dx$$

Leon

Stefan Welke, Germany, sent a solution:

Dear Mr. Böhm,

You can find the calculation of pan Magiera's integral in the attachment (using Mathematica).

With best regards

Stefan Welke

```
In[1]> Integrate[ $\frac{x^2}{(\cosh[x] + 1) * (\cosh[x] + a)}$ , x, Assumptions -> a > 0]
```

$$\text{Out[1]} = \frac{1}{(-1 + a) (1 + \cosh[x])} \left(\cosh\left[\frac{x}{2}\right] \left(-2 \cosh\left[\frac{x}{2}\right] \left(x \left(x + 4 \log[1 + e^{-x}] \right) - 4 \text{PolyLog}[2, -e^{-x}] \right) + \frac{1}{\sqrt{-1 + a^2}} \right. \right. \\ \left. \cosh\left[\frac{x}{2}\right] \left(-2 x^2 \log\left[1 + \frac{e^x}{a - \sqrt{-1 + a^2}}\right] + 2 x^2 \log\left[1 + \frac{e^x}{a + \sqrt{-1 + a^2}}\right] - \right. \right. \\ \left. \left. 4 x \text{PolyLog}\left[2, \frac{e^x}{-a + \sqrt{-1 + a^2}}\right] + 4 x \text{PolyLog}\left[2, -\frac{e^x}{a + \sqrt{-1 + a^2}}\right] + \right. \right. \\ \left. \left. 4 \text{PolyLog}\left[3, \frac{e^x}{-a + \sqrt{-1 + a^2}}\right] - 4 \text{PolyLog}\left[3, -\frac{e^x}{a + \sqrt{-1 + a^2}}\right] \right) + 2 x^2 \sinh\left[\frac{x}{2}\right] \right)$$

```
In[2]> Integrate[ $\frac{x^2}{(\cosh[x] + 1) * (\cosh[x] + a)}$ , {x, 0, Infinity}, Assumptions -> a > 0]
```

$$\text{Out[2]} = \frac{\sqrt{-1 + a^2} \pi^2 + 6 \text{PolyLog}\left[3, \frac{1}{-a + \sqrt{-1 + a^2}}\right] - 6 \text{PolyLog}\left[3, -\frac{1}{a + \sqrt{-1 + a^2}}\right]}{3 (-1 + a) \sqrt{-1 + a^2}}$$

I passed this to Leon and he wrote back:

Many thanks to Mr. Welke. My colleague and his wife found out the following result for the integral:

$$\left(\pi^2 - \frac{(b^3 + b \cdot \pi^2)}{\sqrt{a^2 - 1}} \right) \cdot \frac{1}{3(a-1)} \quad \text{with } b = \log\left(a + \sqrt{a^2 - 1}\right).$$

I must admit that I am not able to confirm or to disprove the identity of both expressions. The second one is doubtless much more comfortable than the Mathematica result. All what I could do was trying to verify the identity by substituting any value for $a > 1$ and comparing the numerical results:

I started with Mathematica using its Online Calculator <http://www.wolframalpha.com/>



(Sqrt[3]*pi^2+6*PolyLog[3,1/(-2+Sqr[3])]-6*PolyLog[3,-1/(2+Sqr[3])])/(3*Sqr[3])

Input:

$$\frac{\sqrt{3} \pi^2 + 6 \operatorname{Li}_3\left(\frac{1}{-2+\sqrt{3}}\right) - 6 \operatorname{Li}_3\left(-\frac{1}{2+\sqrt{3}}\right)}{3 \sqrt{3}}$$

Open code

$\operatorname{Li}_n(x)$ is the polylogarithm function

Decimal approximation: More digits

0.348854254503944402088768921563491814881414741593155232964...

Alternate forms: More

$$\frac{\pi^2 (\sqrt{3} - \cosh^{-1}(2)) - \cosh^{-1}(2)^3}{3 \sqrt{3}}$$

$$\frac{6 \operatorname{Li}_3(-2 - \sqrt{3}) - 6 \operatorname{Li}_3(-2 + \sqrt{3}) + \sqrt{3} \pi^2}{3 \sqrt{3}}$$

$$\frac{6 \operatorname{Li}_3\left(\frac{1}{-2+\sqrt{3}}\right) - 6 \operatorname{Li}_3(-2 + \sqrt{3}) + \sqrt{3} \pi^2}{3 \sqrt{3}}$$

There appeared another “comfortable” result, but not this one given by Leon’s colleagues. (Li stands for logarithmic integral).

Then I evaluated the “short” result for a = 2:

$$\frac{\pi^2 - \frac{3}{b} + b \cdot \pi^2}{\sqrt{(a-1)^2}} \cdot \frac{1}{a-1}$$

$$\frac{\operatorname{LN}(\sqrt{(a-1)^2 + a})^3 + \pi^2 \cdot \operatorname{LN}(\sqrt{(a-1)^2 + a}) - \pi^2 \cdot \sqrt{(a-1)^2}}{3 \cdot (1-a) \cdot \sqrt{(a-1)^2}}$$

$$\frac{\operatorname{LN}(\sqrt{(2-1)^2 + 2})^3 + \pi^2 \cdot \operatorname{LN}(\sqrt{(2-1)^2 + 2}) - \pi^2 \cdot \sqrt{(2-1)^2}}{3 \cdot (1-2) \cdot \sqrt{(2-1)^2}}$$

[0.3488542545](#)

Seems to be correct!!

Text boxes for the latest version of TI-NspireCAS (version 4.5)

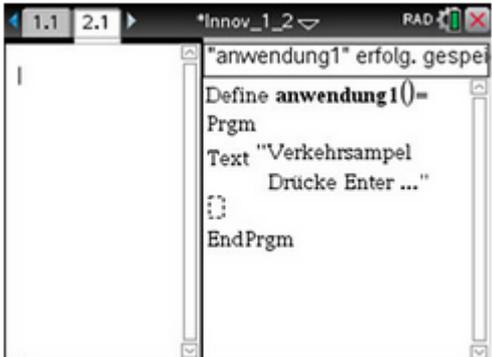
Mail from Wolfgang Pröpper, Germany

(I translate because the problem might be of interest for all Nspire users, Josef.)

Hello Josef,

in *10 Minutes of Code* for TI-Innovator I found in Unit 1, Application, Step 3 (Setting up the Title Screen - Einrichten des Eröffnungsbildschirms) for programming a text box:

Type between quotes the message "*Verkehrsampel* [enter] *Drücke Enter...*".

 <pre> "anwendung1" erfolg. gespeichert Define anwendung1()= Prgm Text "Verkehrsampel Drücke Enter ..." EndPrgm </pre>	<p>The text box should look as shown below.</p> 
--	--

My problem is:

When I press after word *Verkehrsampel* the [Enter]-key then the cursor leaves the text-command and I am faced with a new empty program line. I tried several key combinations, but nothing works. Same happens on the handheld.

Neither the CX-manual nor the Reference Guide provide any advice.

Do you have any idea?

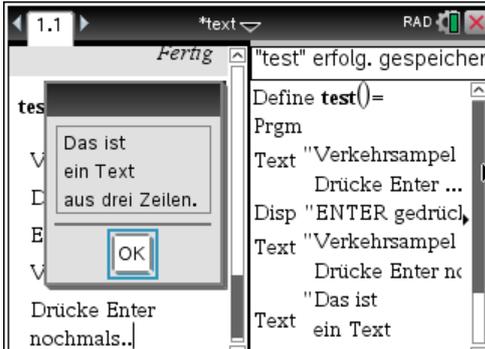
Best regards

Wolfgang

I was bewildered and immediately tried to produce a two-lined textbox and I failed like Wolfgang. But I knew that I did it when I translated the US-original. I loaded the respective tns-file and the two-line box appeared. Then I checked the US-website: "... press the [enter] key and proceed typing ...". I had no idea ...

It was just the night when I watched Marcel Hirscher running for the Olympic Gold Medal – and then I had the "golden idea": At night I took my laptop with TI-NspireCX Version 4.4 installed and tried again the two-line text box ... and it worked.

So, the mystery was solved. But I still did not know how to create a multi-line message in a text box in version 4.5. (We had one – not very comfortable idea, which worked: write the text with any Editor and then copy and paste the text between quotes into the Nspire program ...).



```

"test" erfolg. gespeichert
Define test()=
Prgm
Text "Verkehrsampel
      Drücke Enter ..."
Disp "ENTER gedrückt"
Text "Verkehrsampel
      Drücke Enter nochmals.."
Text "Das ist ein Text
      aus drei Zeilen."

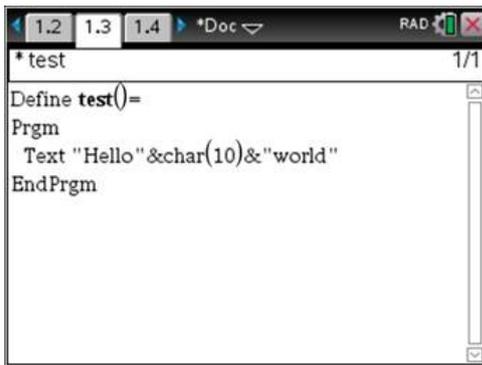
```

Wolfgang wrote to ticares about the problem and we received an answer from an expert, John Hanna

Von: "John Hanna" <jehanna@optonline.net>

CX+Innovator and CE+Innovator lessons have been updated, but not the original 10 minutes stuff.

There are two methods for multi-line Text statements:



```
* test 1/1
Define test(=
Prgm
Text "Hello"&char(10)&"world"
EndPrgm
```

10 is the ASCII-code for LF = line feed.

In a Question app, write the Text command in a math box:



Then copy/paste into a program. This would be challenging on a handheld. You'd have to provide a doc with the question app.

The Program Editor and the Notes app were modified to prevent an [enter] within a string to make Send statements easier to write and process, as embedding an [enter] messes up the command.

Regards,
John

Mail from Herbert Nieder, Germany

Sehr geehrter Herr Mag. Josef Böhm,

bei dem Versuch, affine Abbildungen im \mathbb{R}^2 systematisch zu erfassen, bin ich auf das Problem gestoßen, dass so gut wie alle Artikel irgendwo stecken bleiben, auch Wikipedia oder zum Beispiel

<https://www.cg.tuwien.ac.at/courses/CG1/textblaetter/02%20Geometrische%20Transformationen.pdf>

Ich wollte eine konkrete Darstellung in Form von Matrizen und keinen Baukasten. Homogene Koordinaten werden fast überall verwendet, aber die letzte Konkretisierung fehlt, z.B. habe ich nirgendwo eine Matrix gefunden, die eine Scherung an der Geraden $y=mx+b$ darstellt, ebensowenig eine Spiegelung an derselben, auch keine Scherungen oder Spiegelungen an Parallelen zur y -Achse.

Der Schlüssel zum Verständnis der Ergebnisse der Matrixmultiplikationen der TU Wien im Vergleich mit der Darstellung der Geraden $y=mx+b$ als Fixgerade der Abbildungen war für mich die Gleichsetzung $m=\tan(\alpha)$ und der Gleichung $1/\cos^2=\tan^2+1$. Als ich das zusammengebracht hatte, war der Rest "nur" noch eine Fleißarbeit.

Falls Sie mit dem Ergebnis etwas anfangen können, freut es mich, falls nicht, wäre ich für eine kurze Mitteilung dankbar.

Im Anhang finden Sie eine Fassung als .odt und eine Kurzfassung als .dfw sowie -mache ich immer - eine .pdf-Datei

Falls Sie die Ergebnisse brauchbar finden, aber eine andere Form wünschen, könnte ich diese liefern, seit 1.2.2018 bin ich Pensionär und kann mich kümmern, aber sonst hätte ich mich wohl auch nicht damit beschäftigt.

Beste Grüße

Herbert Nieder

Herbert Nieder has been DUG member since 1992. I will give a short summary of his mail:

Dear Mr. Böhm,

When I tried to collect systematically affine mappings in \mathbb{R}^2 I came across the problem that almost all respective articles are not really complete (Wikipedia or as another example

<https://www.cg.tuwien.ac.at/courses/CG1/textblaetter/02%20Geometrische%20Transformationen.pdf>.)

I could not find a shear matrix wrt to a straight line $y = m x + b$ and also not a matrix for a reflection wrt to the line ...

I attach an article (odt, pdf and dfw file) and would be happy if you would find it useful. I am retired since February 2018 and could modify the paper if necessary.

Best regards

Herbert Nieder

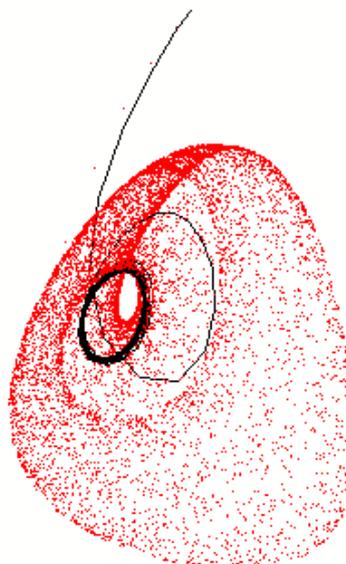
Herbert Nieder's paper was the start of an intense exchange of mails and files. Finally, we both were very satisfied with the result of our collaboration and I will publish its final version in the next DNL. Thanks to Herbert for his patience and cooperation, Josef

The red points are a homemade 3D-plot of an ODE-attractor.

Just for fun I took the coefficients and set up the respective ODE:

$$\frac{dx}{dt} = f(x, y, z), \frac{dy}{dt} = g(x, y, z), \frac{dz}{dt} = h(x, y, z)$$

and solved the system numerically using DERIVE's built-in RK-function (black curve). (You can find it in SA_DERIVE_3D.dfw.)



Ein Temperaturwarner realisiert mit dem TI-Innovator™

A Temperature Warner realized with TI-Innovator™

Ralph Huste, Hubert Langlotz

Der TI-Innovator™ Hub ist ein mit industriellen Komponenten aufgebauter Controller, der die Signale von Sensoren aufnehmen und Aktoren ansteuern kann. Dazu gibt es viele fertig Sensoren, aber man kann auch eigene Schaltungen auf Steckplatinen (Breadboard) entwerfen und anschließen. Der TI-Innovator™ Hub muss mit einem mit einem TI-Nspire™CX / CAS oder einem TI-84 Plus CE-T (nicht TI-84 plus) bzw. der entsprechenden Computersoftware betrieben werden, da er auf die Stromversorgung dieser Geräte angewiesen ist. Auf diesen Geräten werden auch die Programme geschrieben, die für den Betrieb des TI-Innovators™ Hub notwendig sind. Die möglichen Programmiersprachen sind TI Basic oder LUA. Im Folgenden ist ein kleines Beispielprogramm dargestellt, welches Eingangsdaten (hier Temperatur) erfasst und hieraus bestimmte Aktionen (verschiedenfarbiges Leuchten der LEDs bzw. Soundausgabe) erzeugt. Viele weitere solcher kleinen Programme zum Einstieg in die Arbeit mit dem Innovator finden sie ab Sommer im neuen T3-Heft „Sensorik mit und ohne TI-Innovator“ herausgegeben durch Frank Liebner und Hubert Langlotz

The TI-Innovator™ Hub is a controller assembled by industrial components designed to accept signals from sensors and to control actuators. Many readymade sensors are available but one can also design own circuits on breadboards and connect them with the Hub. The TI-Innovator™ Hub must be run via a TI-Nspire™CX / CAS or a TI-84 Plus CE-T (not TI-84 plus) or the respective computer software because it needs the electricity supply of these devices. On these devices you can write the programs for running the TI-Innovator™ Hub. Possible programming languages are TI Basic and LUA. In the following we will present a short sample program which accepts input data (here temperature) and reacts in form of lighting the LEDs in different colors and sounds, respectively. Many similar short introductory programs for working with the Innovator can be found in the T3-booklet „Sensorik mit und ohne TI-Innovator“ (published by Frank Liebner and Hubert Langlotz in summer 2018).

Aufgabe:

Das Programm soll mithilfe eines externen Sensors die Umgebungstemperatur erfassen und auswerten:

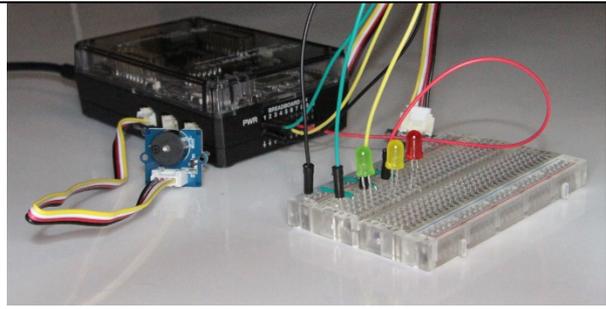
- ist die Temperatur im Sollbereich, leuchtet die grüne LED,
- nähert sich die Temperatur der Grenze, erlischt die grüne LED und eine gelbe LED leuchtet auf,
- ist eine kritische Temperatur erreicht, erlischt die gelbe LED, eine rote LED leuchtet auf und ein Signalton ertönt.

Task:

The program is intended to record and evaluate the environment temperature:

- when temperature is within target range then the green LED will be on,
- when temperature approaches the critical boundary then the green LED will be off and the yellow one will be on,
- when the critical temperature is reached then the red LED will be on and a warning buzzer will sound.

Hardware
 Innovator
 Temperatursensor / [temperature sensor](#)
 Breadboard
 LED: grün, gelb, rot / [green, yellow, red](#)
 Buzzer



mögliche Umsetzung / realisation:

Zuordnung des Sensors, der Leuchtdioden und des Buzzers zu den Eingängen bzw. Ausgängen des Innovators

[Assign the sensor, the LEDs and the buzzer to input and output of the Innovator](#)

Abbruch, wenn Esc-Taste gedrückt wurde

[Cancellation if Esc-key is pressed](#)

Temperatur erfassen und ausgeben

[Record and display temperature](#)

Fallunterscheidung für die gewählten Temperaturbereiche mit den gewählten Anzeigen und Ton

[Case decision for the chosen temperature ranges and the respective light signals and sound](#)

```

Define tempsensor()=
Prgm
:Send "CONNECT TEMPERATURE 1 IN 1 AS
NTC"
:Send "CONNECT LED 1 BB 1"
:Send "CONNECT LED 2 BB 2"
:Send "CONNECT LED 3 BB 3"
:Send "CONNECT BUZZER 1 TO OUT 1"
:
:While getkey()!="esc"
:
:Send "READ TEMPERATURE 1"
:Get t
:Disp "Temp ",t,"°"
:If t<28 Then
:  Send "SET LED 2 OFF"
:  Send "SET LED 1 OFF"
:  Send "SET LED 3 ON"
:Else
:  If t<30 Then
:    Send "SET LED 3 OFF"
:    Send "SET LED 1 OFF"
:    Send "SET LED 2 ON"
:  Else
:    Send "SET LED 3 OFF"
:    Send "SET LED 2 OFF"
:    Send "SET LED 1 ON"
:    Send "SET BUZZER 1 ON"
:  EndIf
:EndIf
:EndWhile
:EndPrgm
  
```

There are several possibilities to define and plot – tiny – points and – small/medium/large points (dots).

Insert > Plot > give details

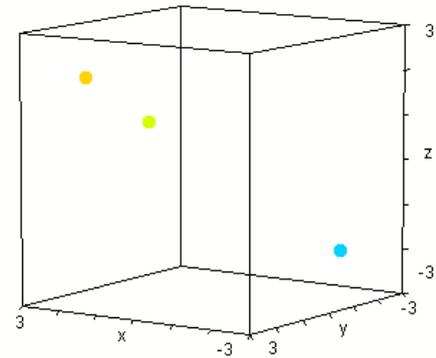
$$\text{list} := \begin{bmatrix} [-2, -2, -2] \\ [1, 1, 1] \end{bmatrix}$$

$$\text{APPEND}(\text{list}, [[2, 2, 2]])$$

If you want to start with an empty list in a program:

$$\text{list1} := []$$

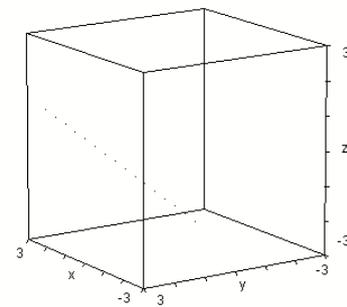
$$\text{APPEND}(\text{list1}, \text{list}) = \begin{bmatrix} [-2, -2, -2] \\ [1, 1, 1] \end{bmatrix}$$



Points must be defined as segments of length 0:

$$\text{VECTOR}\left(\left[\begin{bmatrix} a & a+1 & a-1 \\ a & a+1 & a-1 \end{bmatrix}\right], a, -1, 2, 0.2\right)$$

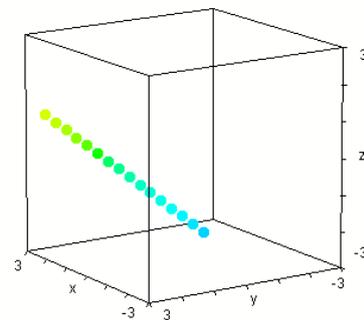
$$\begin{bmatrix} \begin{bmatrix} -1 & 0 & -2 \\ -1 & 0 & -2 \end{bmatrix} \\ \begin{bmatrix} -0.8 & 0.2 & -1.8 \\ -0.8 & 0.2 & -1.8 \end{bmatrix} \end{bmatrix}$$



Plot points separated (again via Insert > Plot > ...)

$$\text{VECTOR}([a, a+1, a-1], a, -1, 2, 0.2)$$

$$\begin{bmatrix} [-1, 0, -2] \\ [-0.8, 0.2, -1.8] \\ [-0.6, 0.4, -1.6] \\ [-0.4, 0.6, -1.4] \end{bmatrix}$$



How to produce lists of points and of dots (see the plots on next page):

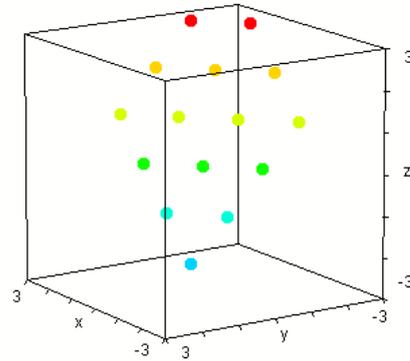
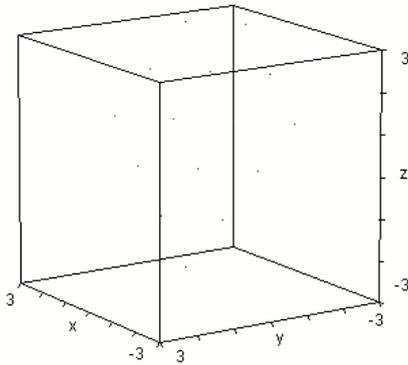
$$\text{VECTOR}\left(\text{VECTOR}\left(\left[\begin{bmatrix} a & b & a-b \\ a & b & a-b \end{bmatrix}\right], a, -1, 2\right), b, -2, 1\right)$$

$$\left[\begin{bmatrix} [-1 & -2 & 1] \\ [-1 & -2 & 1] \\ [0 & -2 & 2] \\ [0 & -2 & 2] \end{bmatrix}, \begin{bmatrix} [-1 & -1 & 0] \\ [-1 & -1 & 0] \\ [0 & -1 & 1] \\ [0 & -1 & 1] \end{bmatrix}, \begin{bmatrix} [-1 & 0 & -1] \\ [-1 & 0 & -1] \\ [0 & 0 & 0] \\ [0 & 0 & 0] \end{bmatrix}, \begin{bmatrix} [-1 & 1 & -2] \\ [-1 & 1 & -2] \\ [0 & 1 & -1] \\ [0 & 1 & -1] \end{bmatrix}\right]$$

$$\text{VECTOR}(\text{VECTOR}([a, b, a-b], a, -1, 2), b, -2, 1)$$

$$\left[\begin{bmatrix} [-1, -2, 1] \\ [0, -2, 2] \\ [1, -2, 3] \\ [2, -2, 4] \end{bmatrix}, \begin{bmatrix} [-1, -1, 0] \\ [0, -1, 1] \\ [1, -1, 2] \\ [2, -1, 3] \end{bmatrix}, \begin{bmatrix} [-1, 0, -1] \\ [0, 0, 0] \\ [1, 0, 1] \\ [2, 0, 2] \end{bmatrix}, \begin{bmatrix} [-1, 1, -2] \\ [0, 1, -1] \\ [1, 1, 0] \\ [2, 1, 1] \end{bmatrix}\right]$$

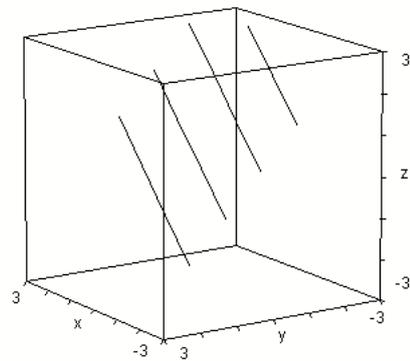
Separated points (left) and separated dots (right):



Points can be connected (in groups) ...

```
VECTOR([VECTOR([a, b, a - b], a, -1, 2), b, -2, 1)
```

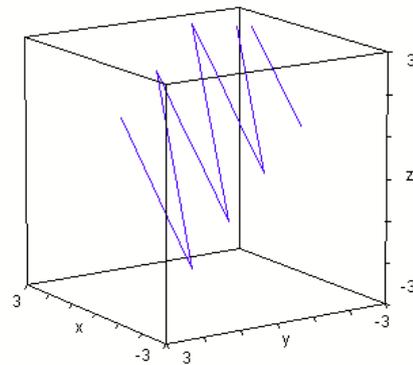
$$\begin{bmatrix} \begin{bmatrix} -1 & -2 & 1 \\ 0 & -2 & 2 \\ 1 & -2 & 3 \\ 2 & -2 & 4 \end{bmatrix} \\ \begin{bmatrix} -1 & -1 & 0 \end{bmatrix} \end{bmatrix}$$



... or as a broken line in space.

```
APPEND(VECTOR(VECTOR([a, b, a - b], a, -1, 2), b, -2, 1))
```

$$\begin{bmatrix} -1 & -2 & 1 \\ 0 & -2 & 2 \\ 1 & -2 & 3 \\ 2 & -2 & 4 \end{bmatrix}$$



And this collection of points results in a plane:

The blue large dots are the points given in the first column.

```
VECTOR(VECTOR([a, b, a - b], a, -1, 2), b, -2, 1)
```

$$\left[\begin{bmatrix} -1 & -2 & 1 \\ 0 & -2 & 2 \\ 1 & -2 & 3 \\ 2 & -2 & 4 \end{bmatrix}, \begin{bmatrix} -1 & -1 & 0 \\ 0 & -1 & 1 \\ 1 & -1 & 2 \\ 2 & -1 & 3 \end{bmatrix}, \begin{bmatrix} -1 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \\ 2 & 0 & 2 \end{bmatrix}, \begin{bmatrix} -1 & 1 & -2 \\ 0 & 1 & -1 \\ 1 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix} \right]$$

