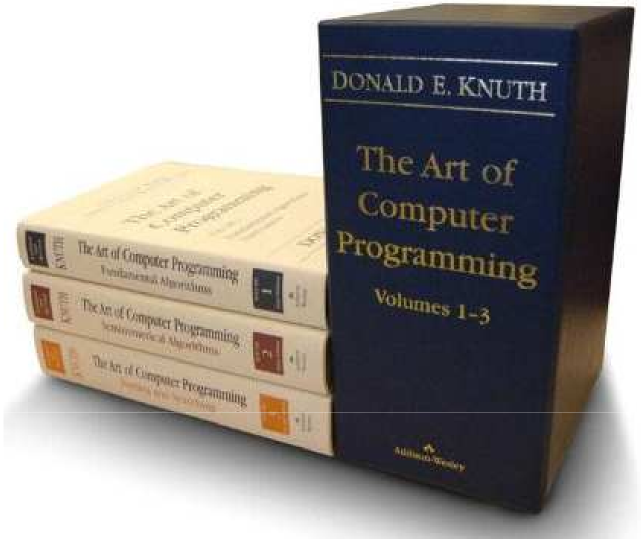
The Art of Computer Programming

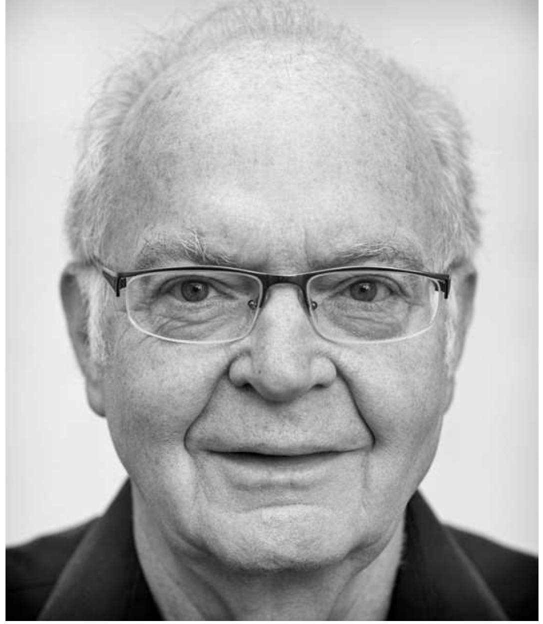
Dr Andrew French

"I'd like to welcome you to this course on Computer Science. Actually, that's a terrible way to start. Computer Science is a terrible name for this business. First of all, it's not a science. It might be engineering, or it might be art, or we'll actually see that computer so-called science actually has a lot in common with magic"

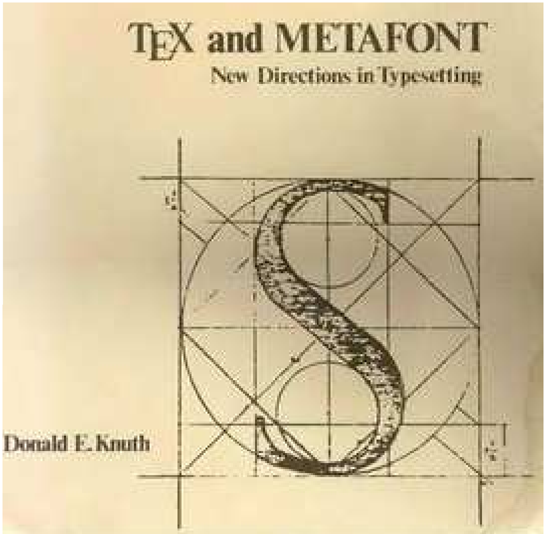
Harold Abelson, MIT (1986)

“Science is what we understand well enough to explain to a computer. Art is everything else we do.”





Donald Knuth 1938-
Stanford University



“Computer programming is an art, because it applies accumulated knowledge to the world, because it requires skill and ingenuity, and especially because it produces objects of beauty. A programmer who subconsciously views himself as an artist will enjoy what he does and will do it better.”

$\TeX$

$$\int_0^\infty \frac{2x \sin x}{1+x^2} dx = \frac{\pi}{e}$$

“I can’t go to a restaurant and order food because I keep looking at the fonts on the menu.”

Lecture map

Fast numerical calculation + display systems

Use of computer programming as an artistic tool

`functions`

`interfaces`

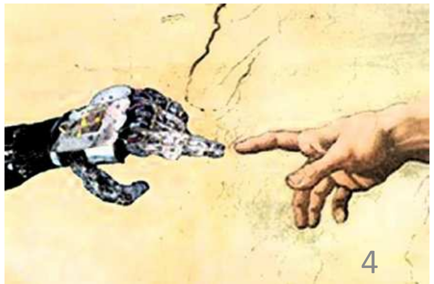
Where you can *design and refine* the tool

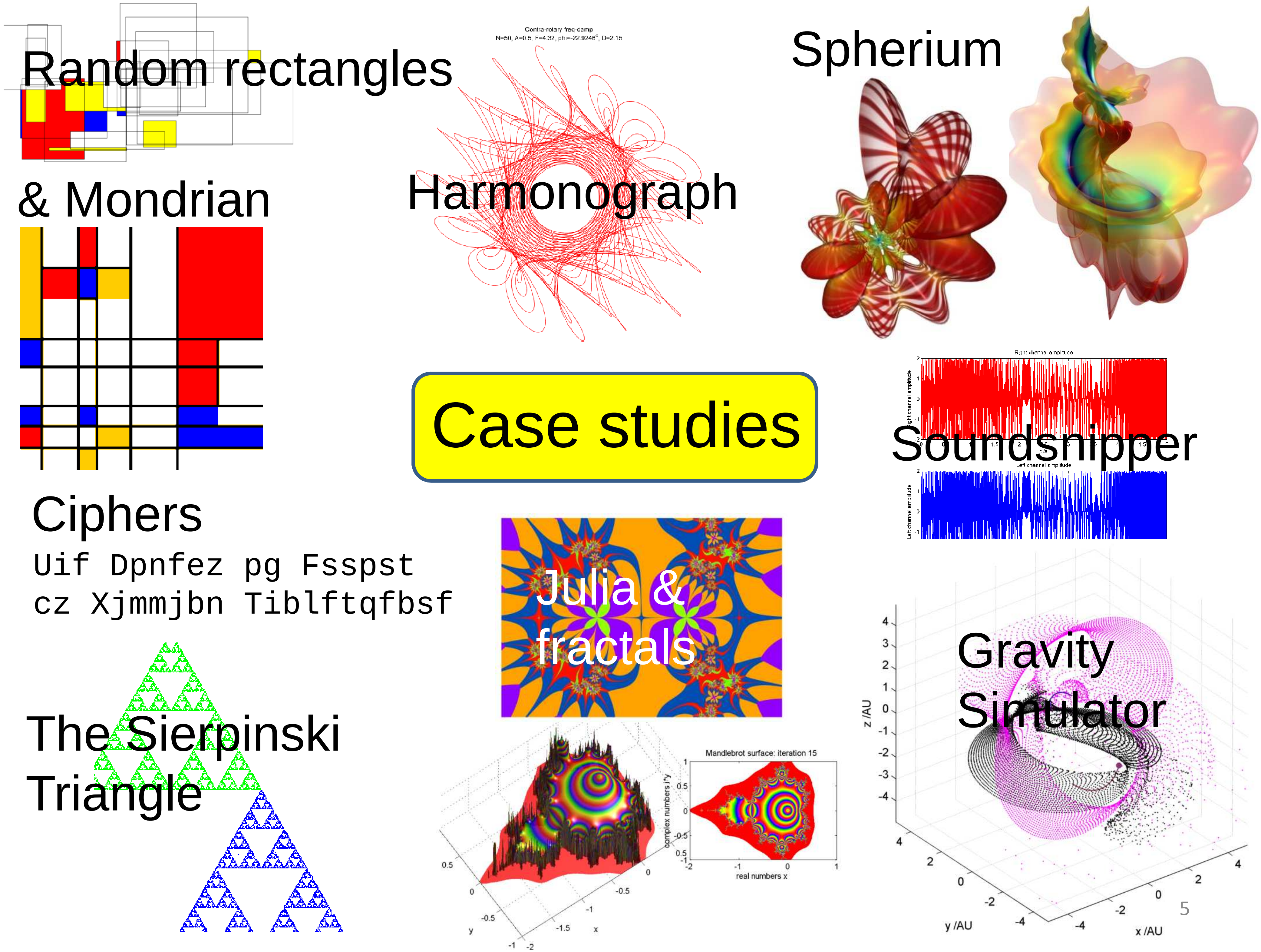
Complexity, and beauty, from simplicity (i.e. code)

Case studies

General thoughts on how *humans* best interact with information technology

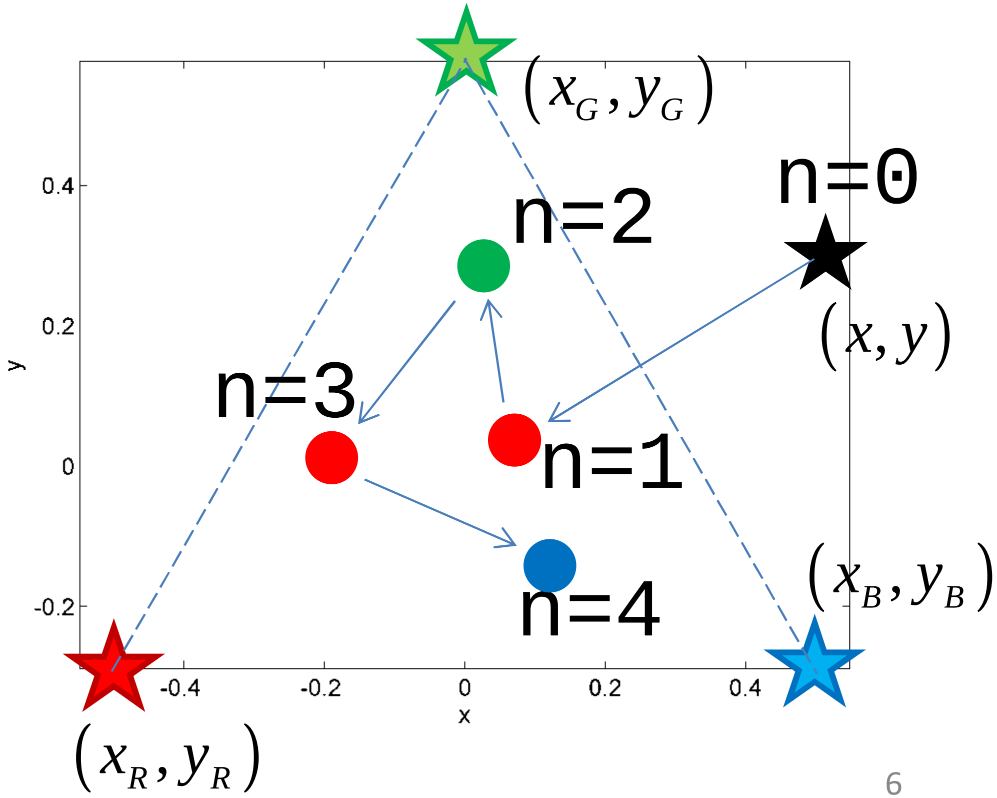
What You See Is What You Need!

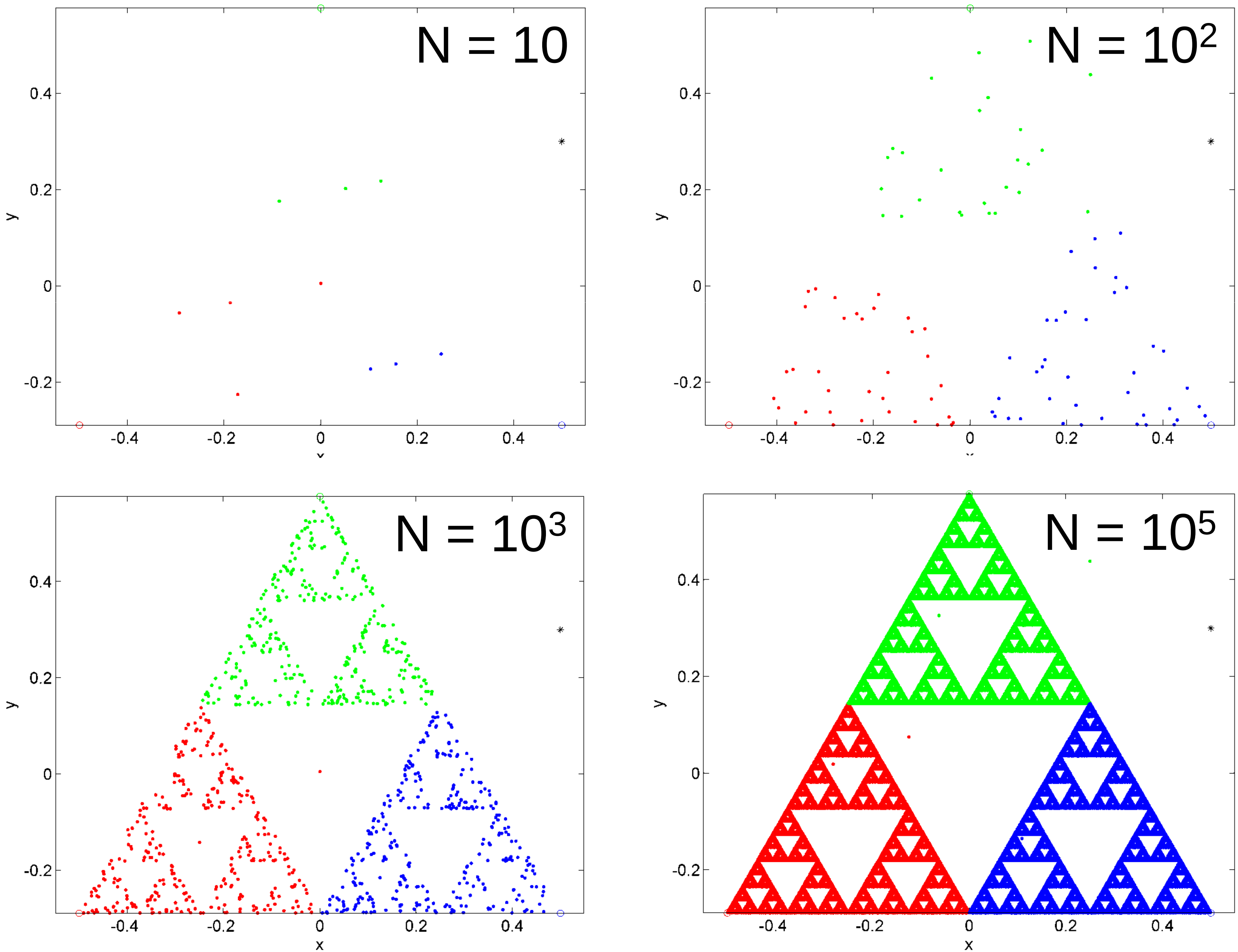



Random rectangles
& Mondrian
Contra-rotary freq-damp
Harmonograph
Spherium
Case studies
Right channel amplitude
Soundsnipper
Left channel amplitude
Ciphers
Uif Dpnfez pg Fsspst
cz Xjmmjbn Tiblftqfbsf
Julia &
fractals
Mandelbrot surface: iteration 15
complex numbers i*y
real numbers x
Gravity
Simulator
z /AU
y /AU
x /AU
The Sierpinski
Triangle

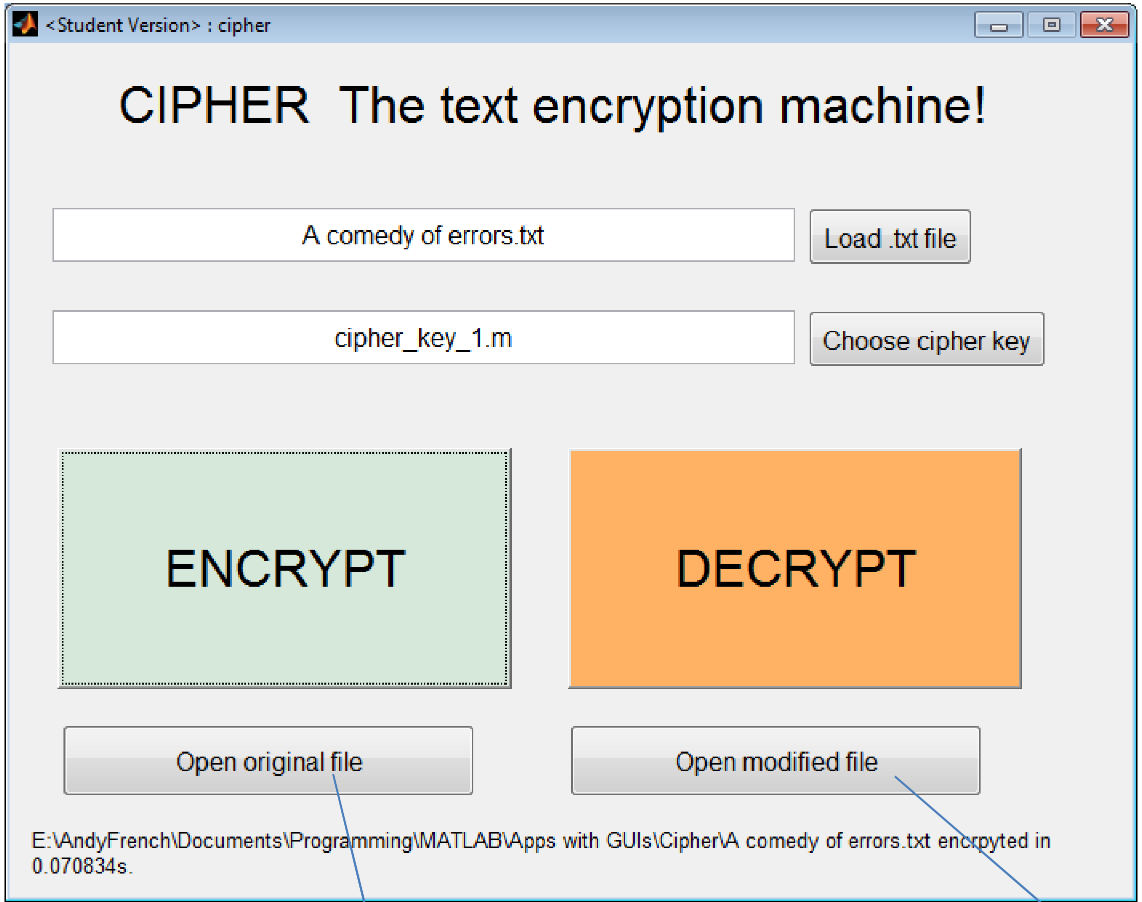
The Sierpinski Triangle

```
for n=1:N
   r = rand; %Generate a random number
   if ( r <= 1/3 )
      %Move half way towards red star
      x = 0.5*( xR + x );
      y = 0.5*( yR + y );
      %Plot a red dot
      plot( x,y, 'r.' );
   elseif ( r > 1/3 ) && ( r <=2/3 )
      %Move ... blue star
      x = 0.5*( xB + x );
      y = 0.5*( yB + y );
      %Plot a blue dot
      plot( x,y, 'b.' );
   else
      %Move ... green star
      x = 0.5*( xG + x );
      y = 0.5*( yG + y );
      %Plot a green dot
      plot( x,y, 'g.' );
   end
end
```





N = 10
N = 10^2
N = 10^3
N = 10^5
x
y
0.4
0.2
0
-0.2
-0.4



This simple *substitution cipher* is called a *Caesar shift*

```
cipher_key = {
    'A','B';...
    'a','b';...
    'B','C';...
    'b','c';...
    'C','D';...
    'c','d';...
    'D','E';...

  etc
                };
```

The Comedy of Errors by William Shakespeare

ACT I

SCENE I. A hall in DUKE SOLINUS'S palace.

Enter DUKE SOLINUS, AEGEON, Gaoler, Officers, and other Attendants

AEGEON

Proceed, Solinus, to procure my fall

And by the doom of death end woes and all.

Uif Dpnfez pg Fsspst cz Xjmmjbn Tiblftqfbsf

BDU J

TDFOF J£ B ibmm jo EVLF TPMJOVT'T qbmbdf£

Foufs EVLF TPMJOVT, BFHFPO, Hbpmfs, Pggjdfst, boe puifs Buufoebout

BFHFPO

Qspdffe, Tpmjovt, up qspdvsf nz gbmm

Boe cz uif eppn pg efbui foe xpft boe bmm£

```
fid = fopen( filename, 'r' ); %Open file filename (read only)

%Store filename text in a row vector A of characters, then close file
A = fscanf(fid, '%c'); fclose(fid);
```

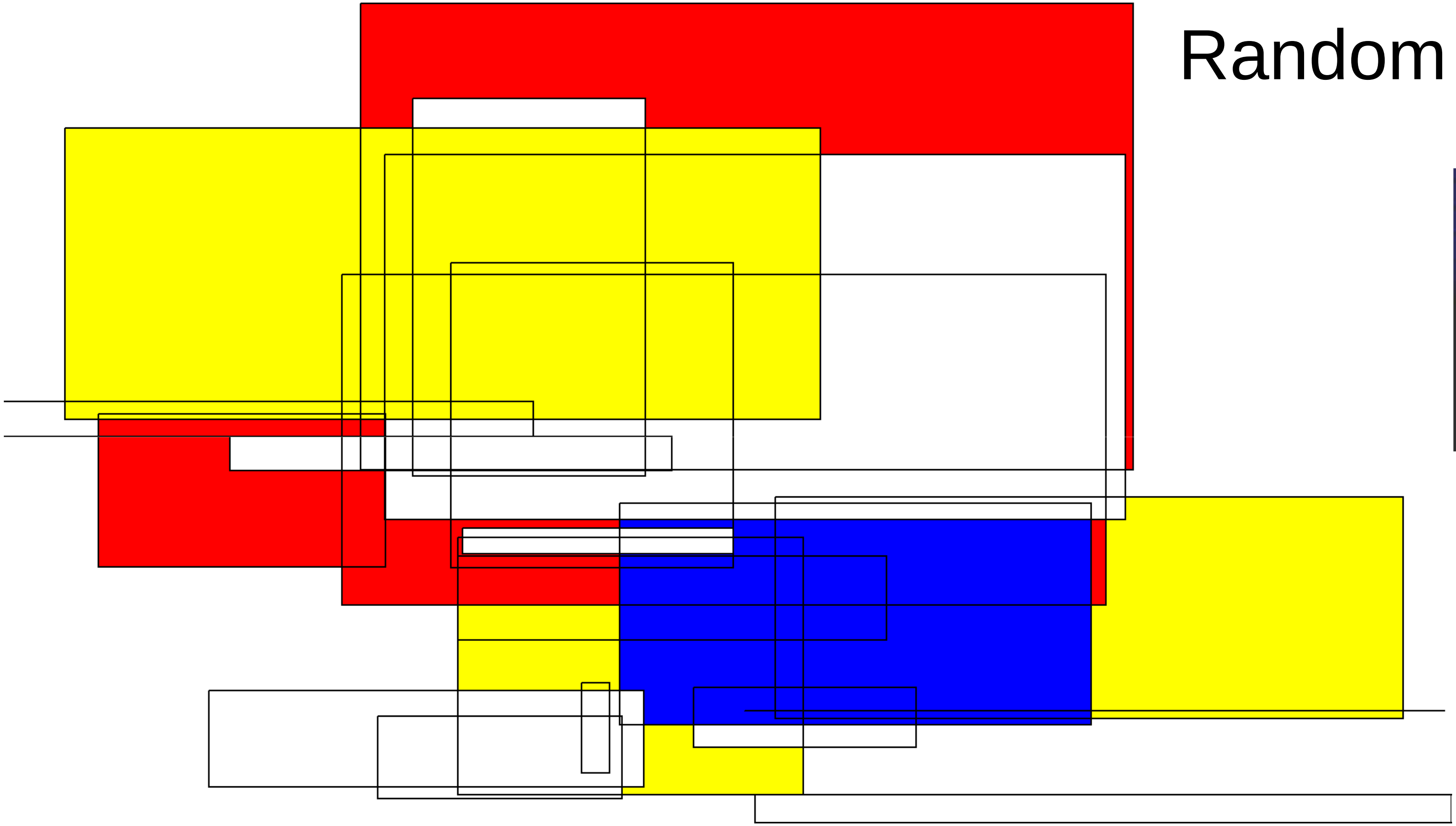
e.g A = 'The Comedy of Errors .....'

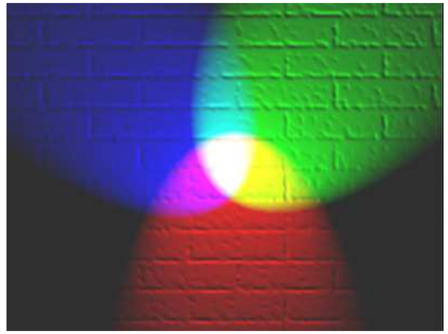
```
%Open file for writing
fid = fopen( strrep( filename,'.txt',['-',cipher_mode,'.txt'] ), 'w' );

%Step through cipher_key, replacing instances of the
%characters with their plaintext or enciphered equivalents
B = A; dim = size(cipher_key);
if strcmp(cipher_mode,'encrypt')==1
    %Encrypt file contents
    for n=1:dim(1)
        indices = strfind( A, cipher_key{n,1} );
        B(indices) = cipher_key{n,2};
    end
else
    %Decrypt file contents
    for n=1:dim(1)
        indices = strfind( A, cipher_key{n,2} );
        B(indices) = cipher_key{n,1};
    end
end

%Write encrypted character array B to a appended, then close file
fwrite(fid, B ); fclose(fid);
```

e.g. `plaintext.txt` would become `plaintext-encrypt.txt`

MATLAB code for `cipher.m`

Random rectangles



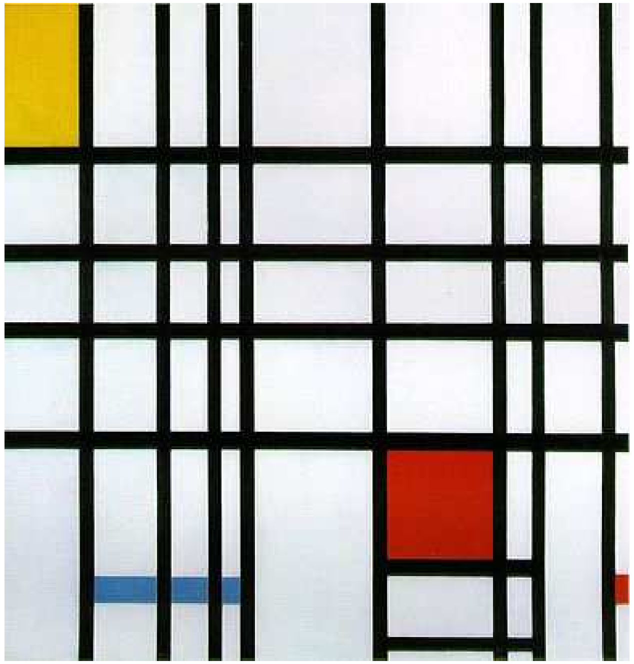


RGB colour model

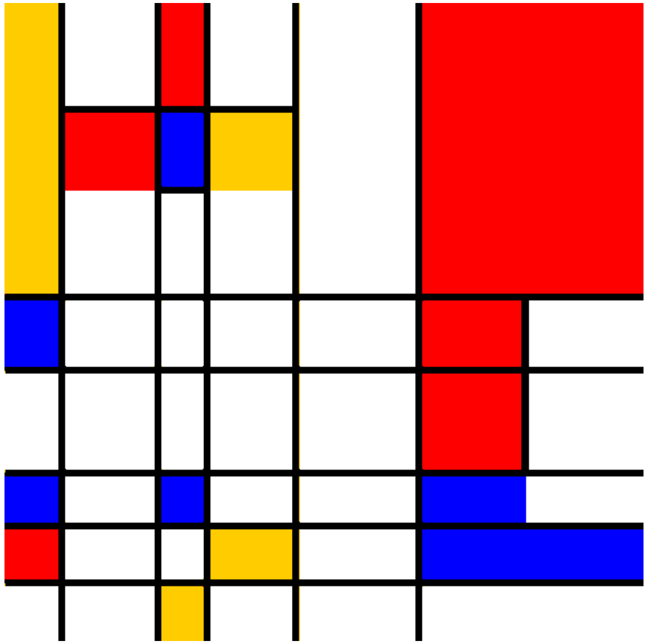
```
colours = { [1,0,0], [1,1,0], [0,0,1], [0,0,0] };
a = 2/( sqrt(5) + 1);
for n=1:20
    x0 = rand; x1 = x0 + rand; y0 = a*rand; y1 = y0 + a*rand;
    patch( 'xdata', [ x0, x1, x1, x0],...
           'ydata', [ y0, y0, y1, y1],...
           'FaceColor', colours{ randi(3,1) } );
end
```

[red, green, blue]

Composition with Yellow, Blue, and Red
1937–42, Piet Mondrian.
Oil on canvas; 72.5 x 69 cm.
London, Tate Gallery.



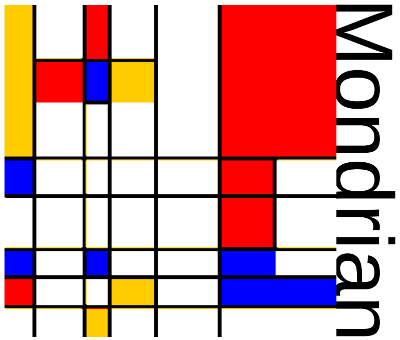
Piet Mondrian
(1872-1944)

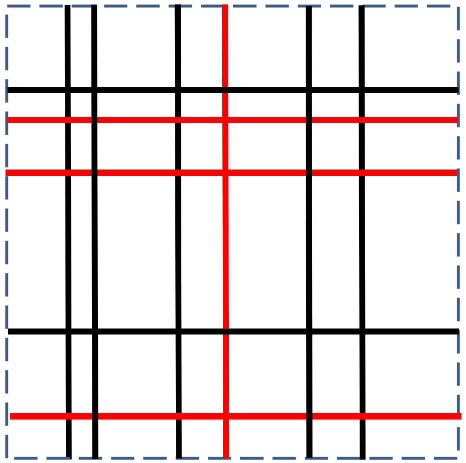


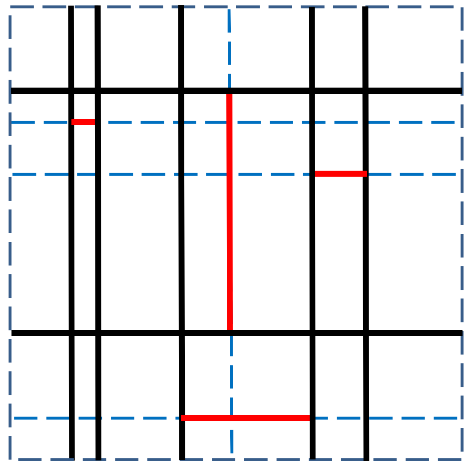
Randomly generated
from `mondrian.m`

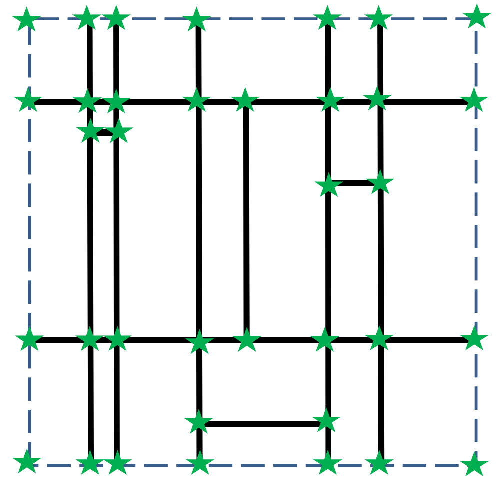
"a post or support"

De Stijl movement
(Amsterdam, 1917-1931)
"Neoplasticism"
"Ultimate simplicity and abstraction"







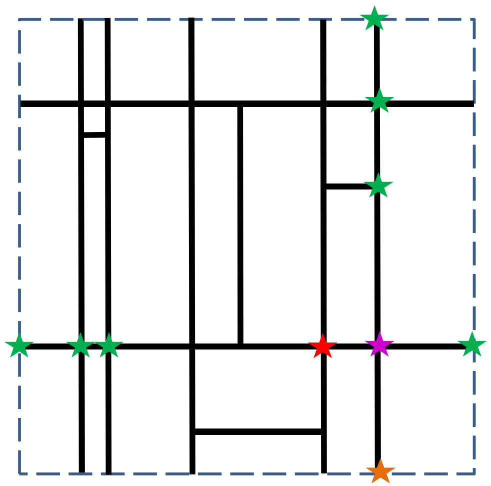


1 Cut a rectangle randomly in horizontal and vertical directions. Randomly divide into two types

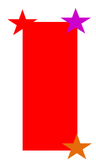
2 Shrink the ‘red’ type to the black lines

3 Find coordinates of all line segment intersections

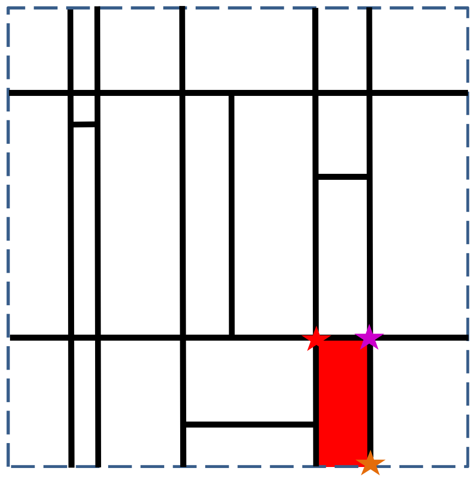
4



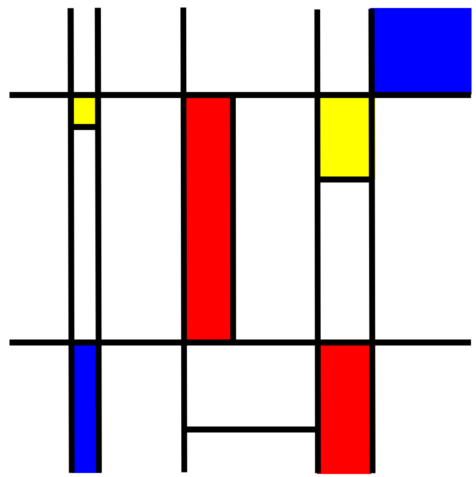
1. Choose an intersection at random ★
2. Find nearest ★ intersection which has the *same y coordinate*
3. Find the *nearest* ★ *intersection from this* which has the *same x coordinate*
4. Construct a rectangle



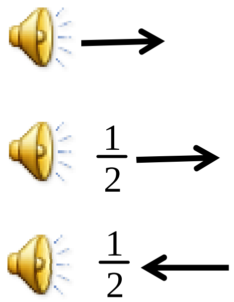
5



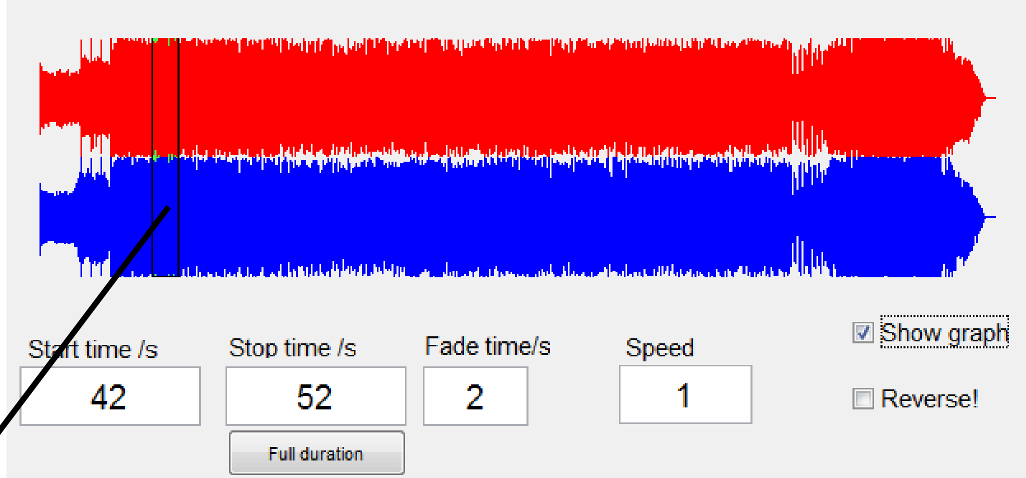
Repeat from **4** cycling through red, blue and yellow colours

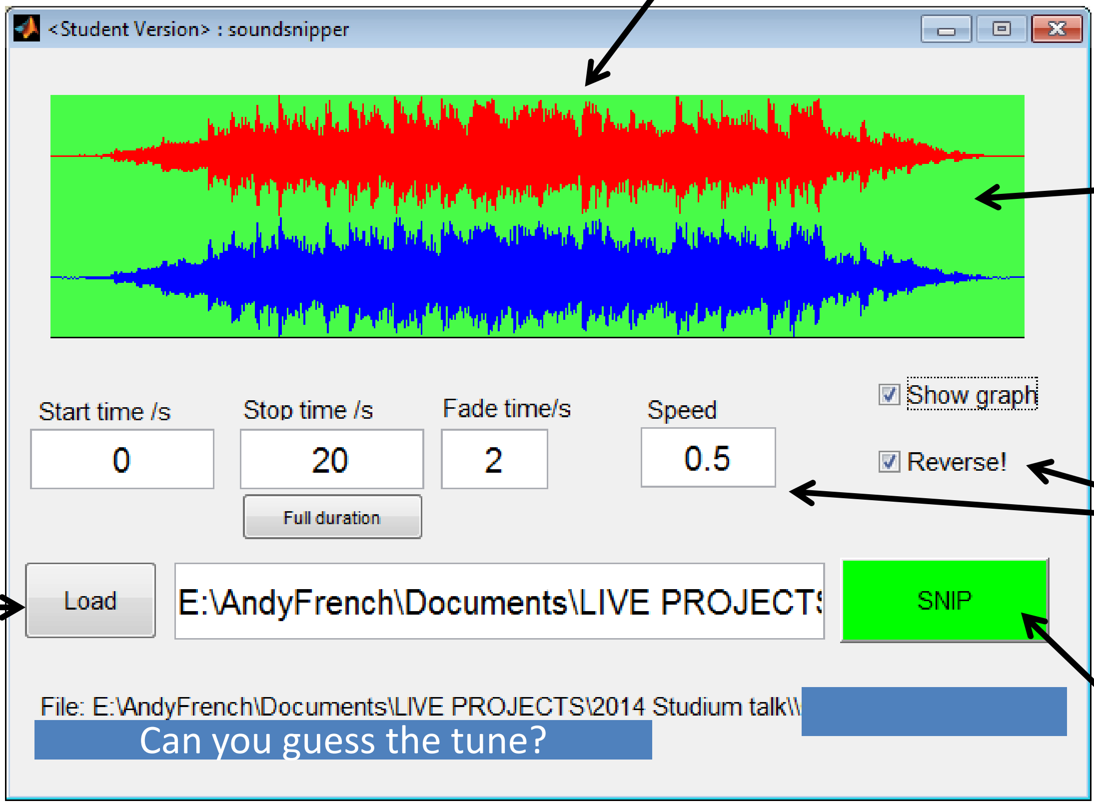


Soundsnipper GUI



Snip a time segment of a `.wav` or `.mp3` audio file





Amplitude vs time of Right and Left stereo channels

Brings up a 'choose file' window

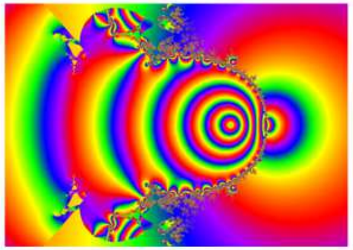
Playback speed or reverse options

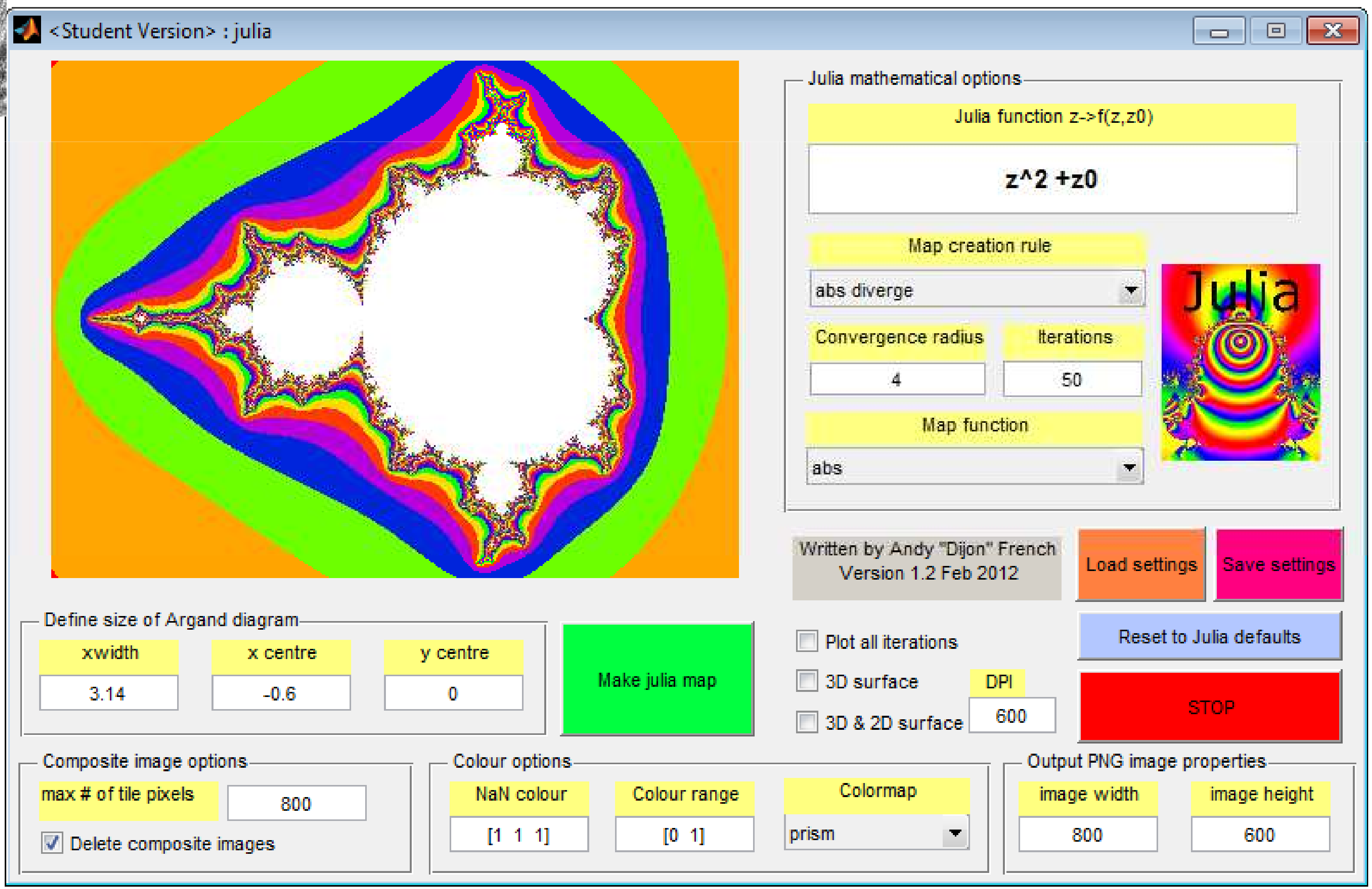
Performs the snip and writes out a new audio file

What You See Is What You Need

Gaston Julia
(1893-1978)

julia





Mandlebrot transformations of **complex numbers**

$$i^2 = -1$$

$$z = x + iy$$

$$x = \mathrm{Re}(z)$$

$$y = \mathrm{Im}(z)$$

$$|z| = \sqrt{x^2 + y^2}$$

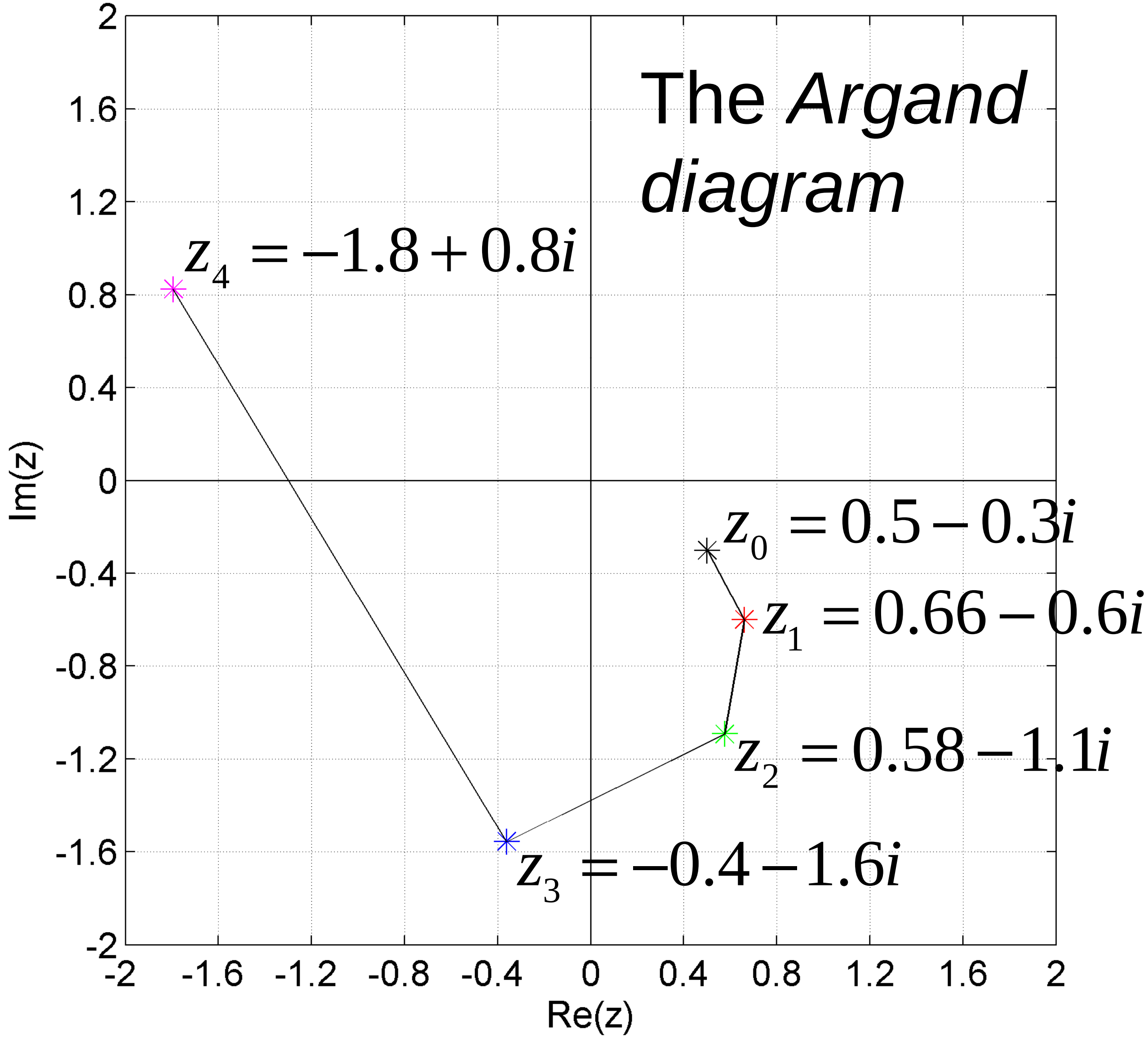
$$(1+i)(1+i)$$

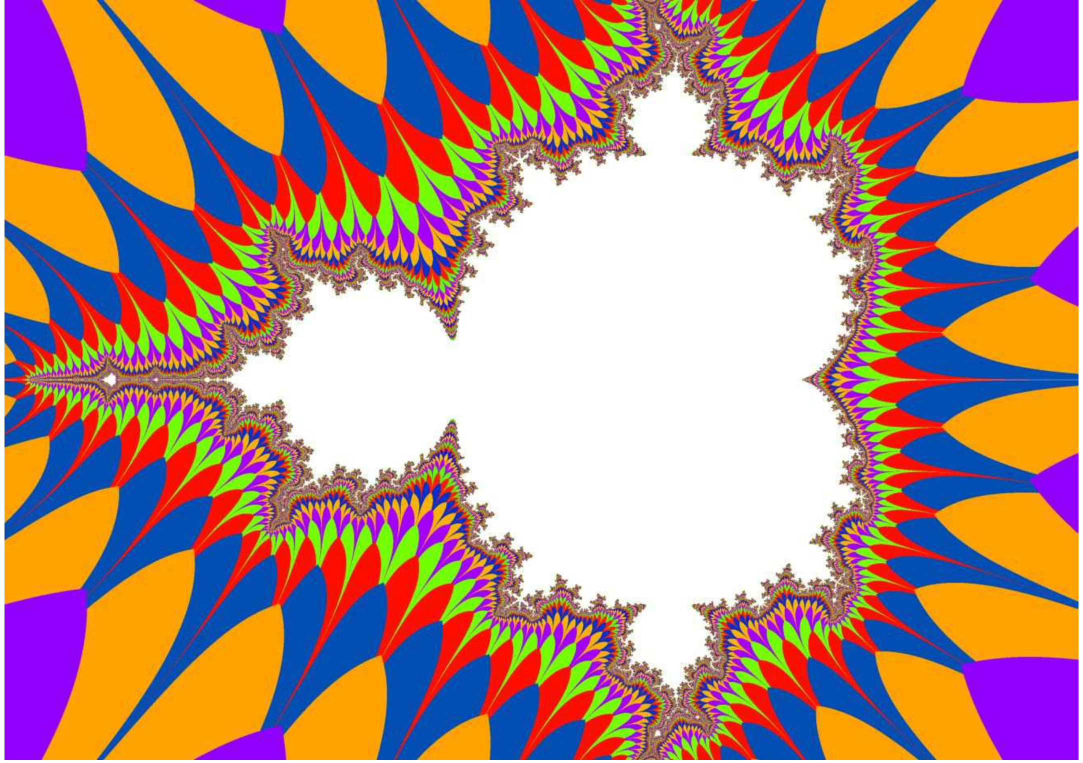
$$= 1 + 2i + i^2$$

$$= 1 + 2i - 1$$

$$= 2i$$

$$z_{n+1} = z_n^2 + z_0$$

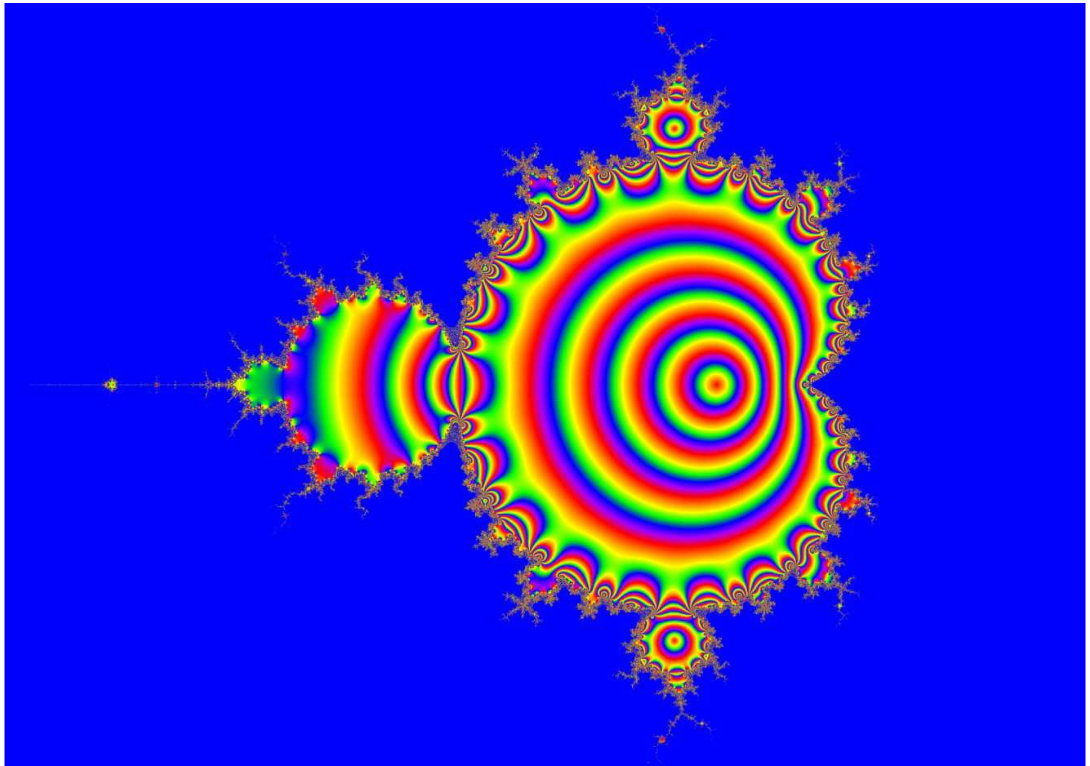




`julia.m` plot option `abs diverge`

Plot a surface with height *h(x,y).* This is the *iteration number* when |z| exceeds a certain value e.g. 4

In this case *colours* indicate height *h(x,y).* It is a 'colour-map'.



`julia.m` plot option `plot z`

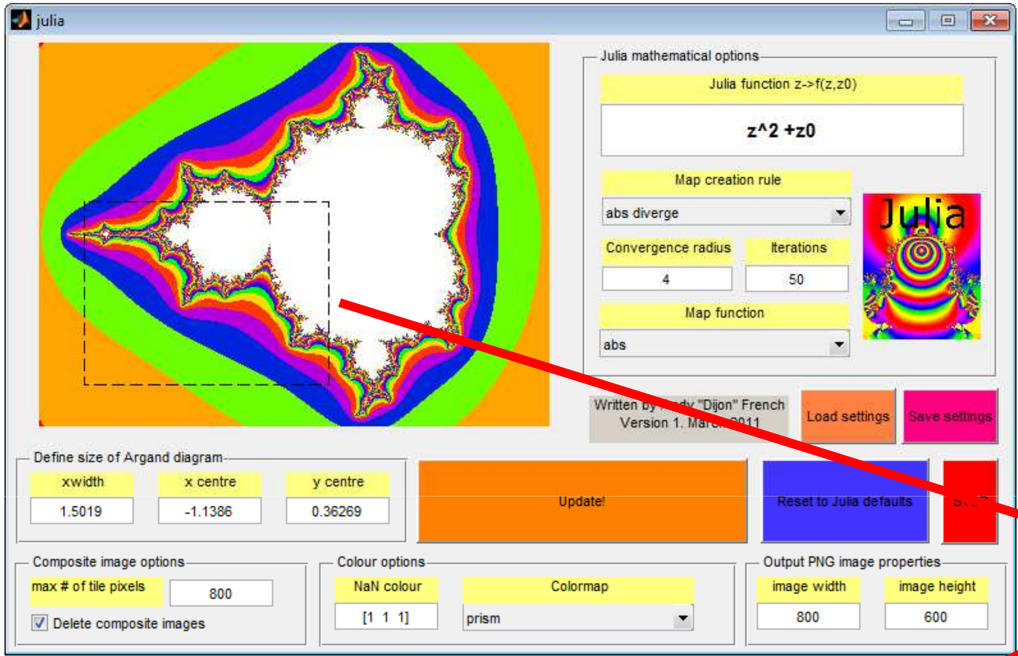
Plot a surface with height *h(x,y)*

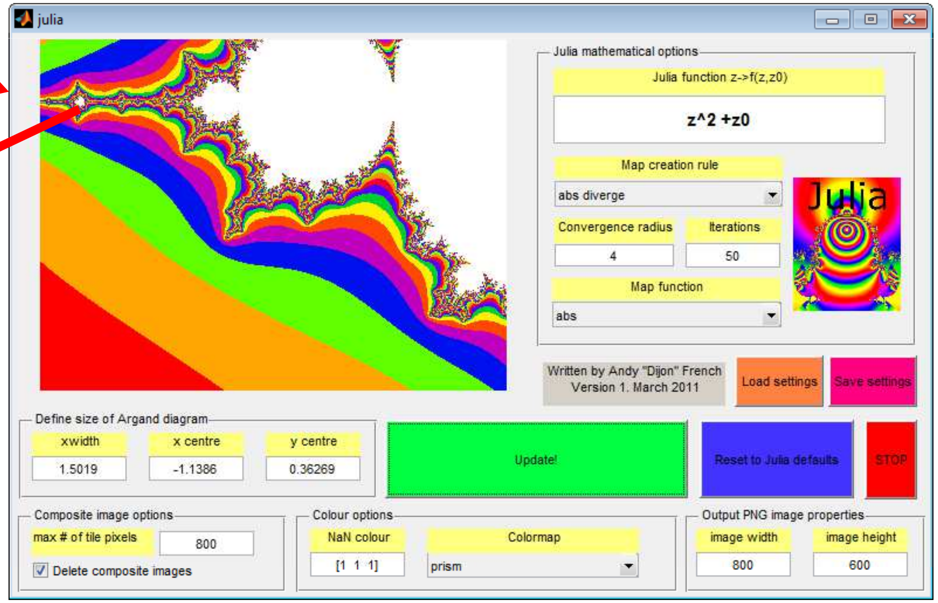
$$x = \mathrm{Re}(z), \quad y = \mathrm{Im}(z)$$

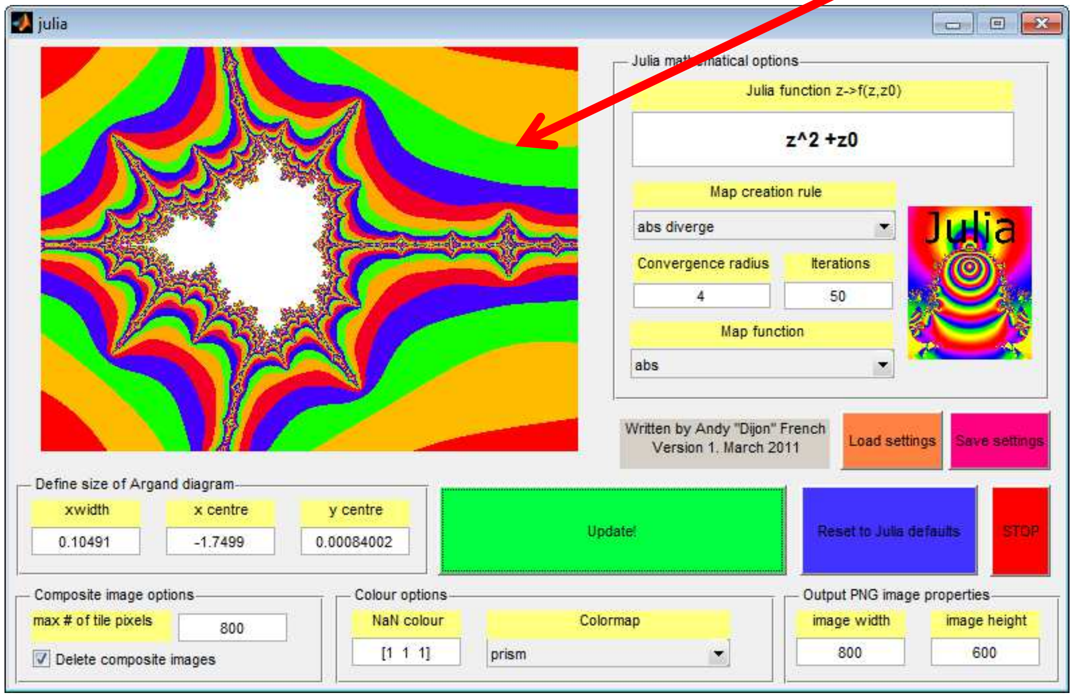
$$h(x,y) = e^{-\sqrt{x^2+y^2}}$$

[Mandlebrot deep zoom (YouTube)]

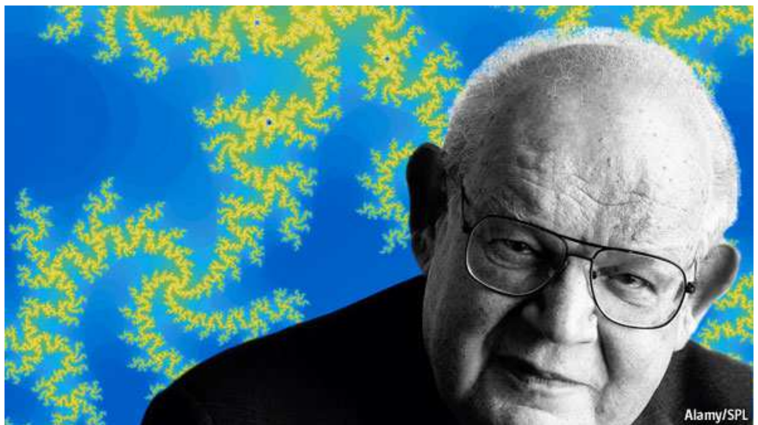
The *Mandlebrot Set* has infinite complexity!
... But a recursive *fractal* geometry

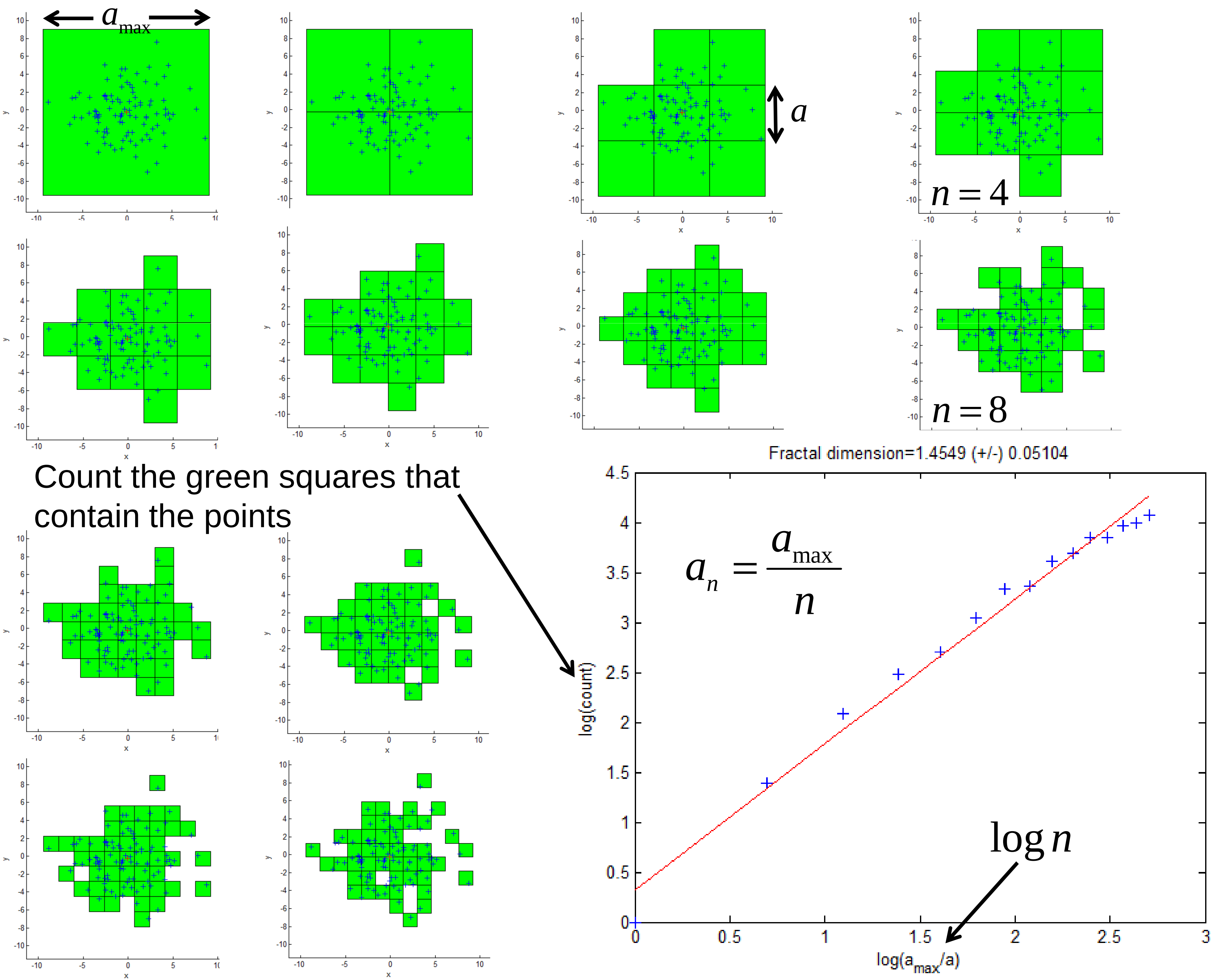






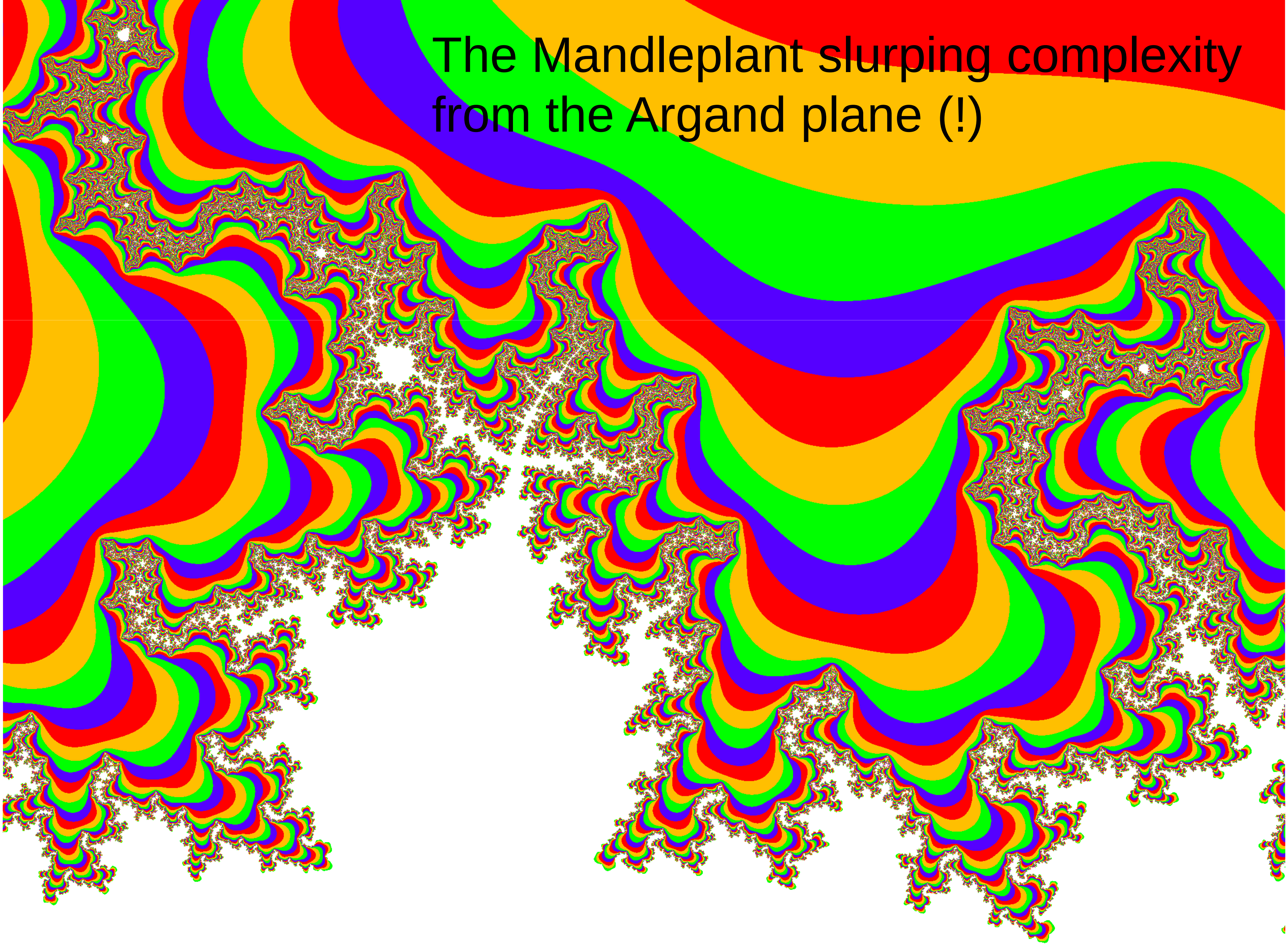
Benoit Mandlebrot (1924-2010)

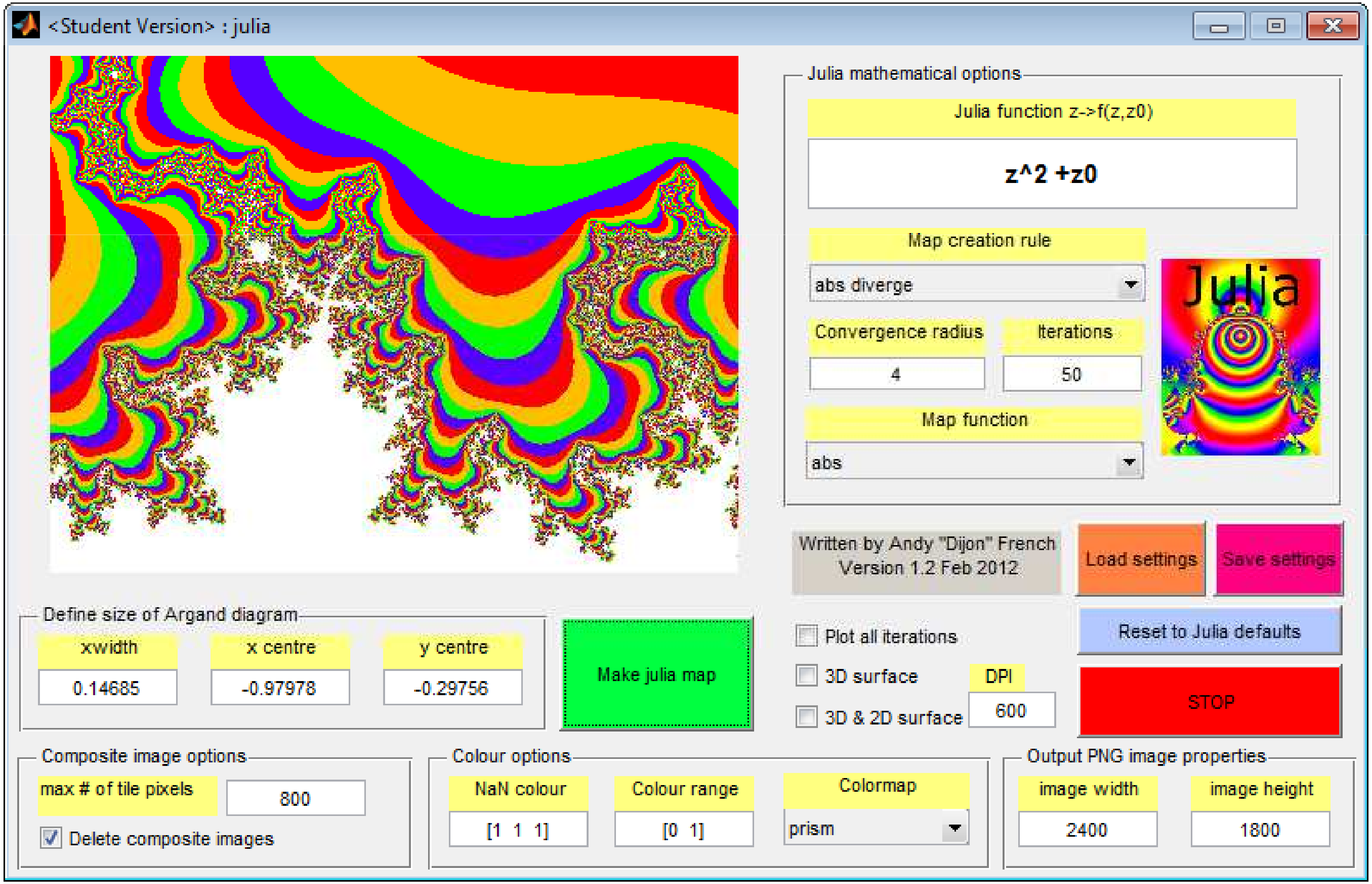




a_{max}
a
$n = 4$
$n = 8$
Count the green squares that contain the points
Fractal dimension=1.4549 (+/-) 0.05104
$a_n = \frac{a_{max}}{n}$
$\log n$
log(count)
log(a_{max}/a)

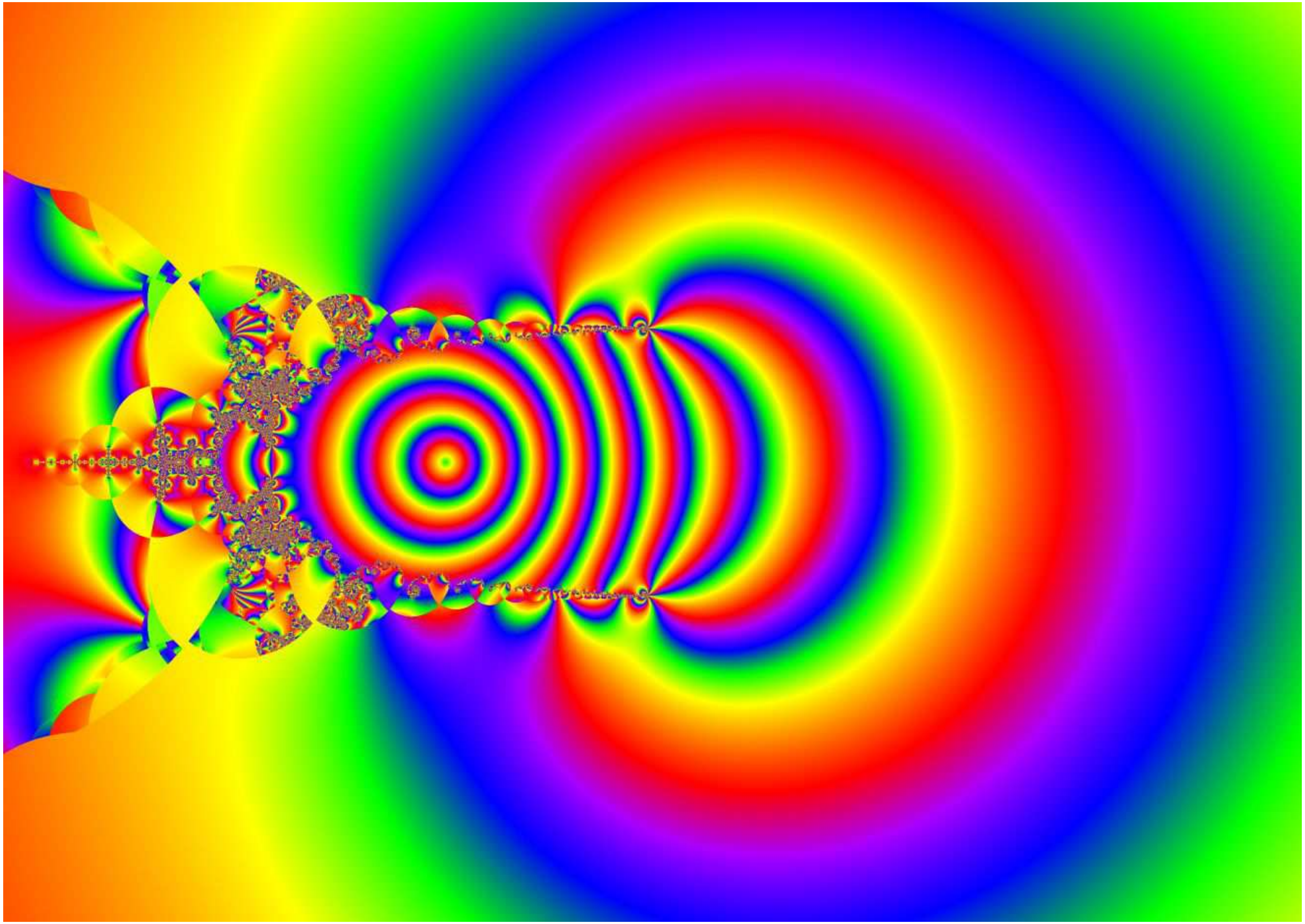
The Mandleplant slurping complexity from the Argand plane (!)





<Student Version> : julia
Julia mathematical options
Julia function z->f(z,z0)
z^2 +z0
Map creation rule
abs diverge
Convergence radius
Iterations
4
50
Map function
abs
Julia
Written by Andy "Dijon" French
Version 1.2 Feb 2012
Load settings
Save settings
Define size of Argand diagram
xwidth
x centre
y centre
0.14685
-0.97978
-0.29756
Make julia map
Plot all iterations
Reset to Julia defaults
3D surface
DPI
600
3D & 2D surface
STOP
Composite image options
max # of tile pixels
800
Delete composite images
Colour options
NaN colour
Colour range
Colormap
[1 1 1]
[0 1]
prism
Output PNG image properties
image width
image height
2400
1800

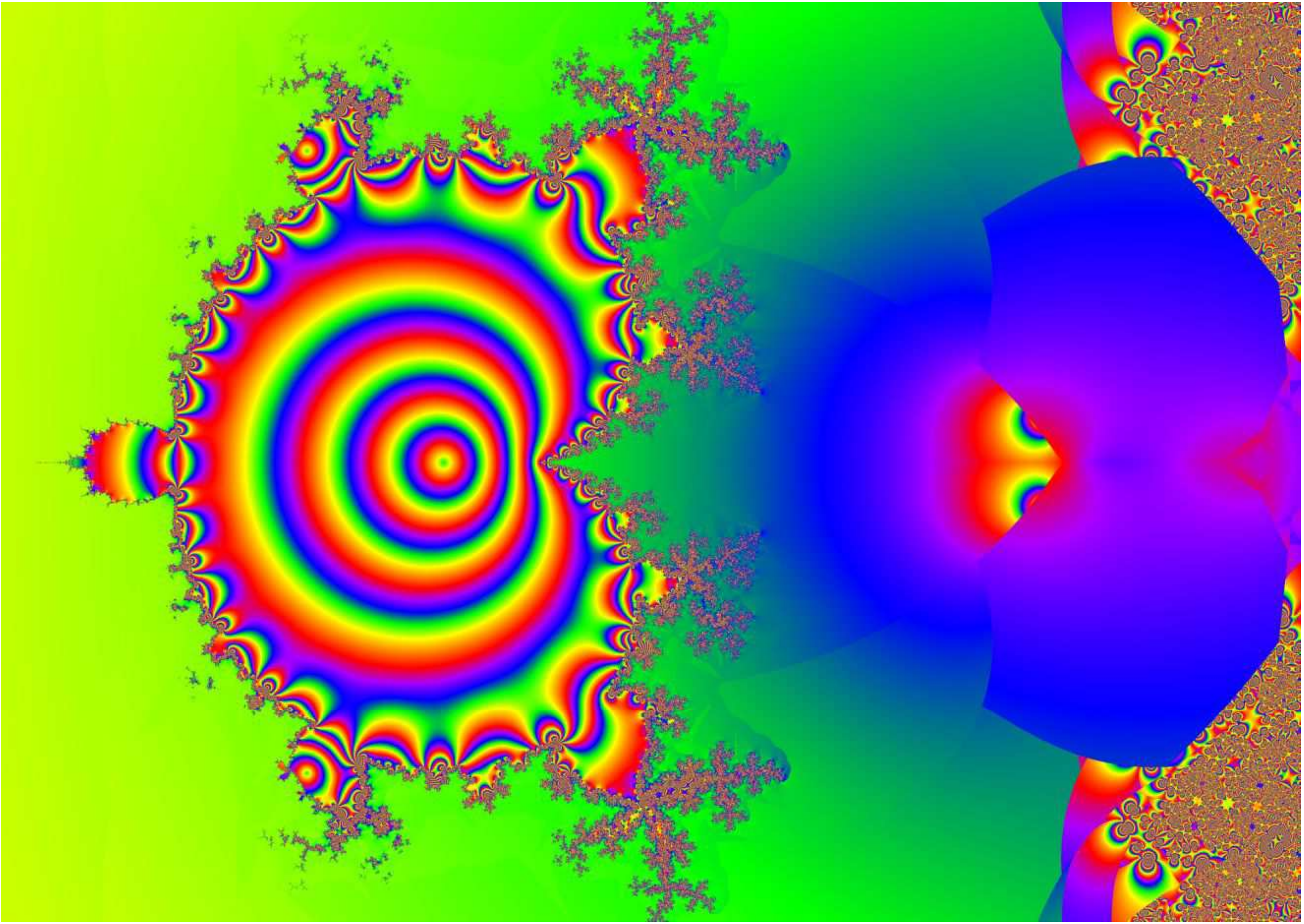

The
Mandlebrot
Variations



The light bulb

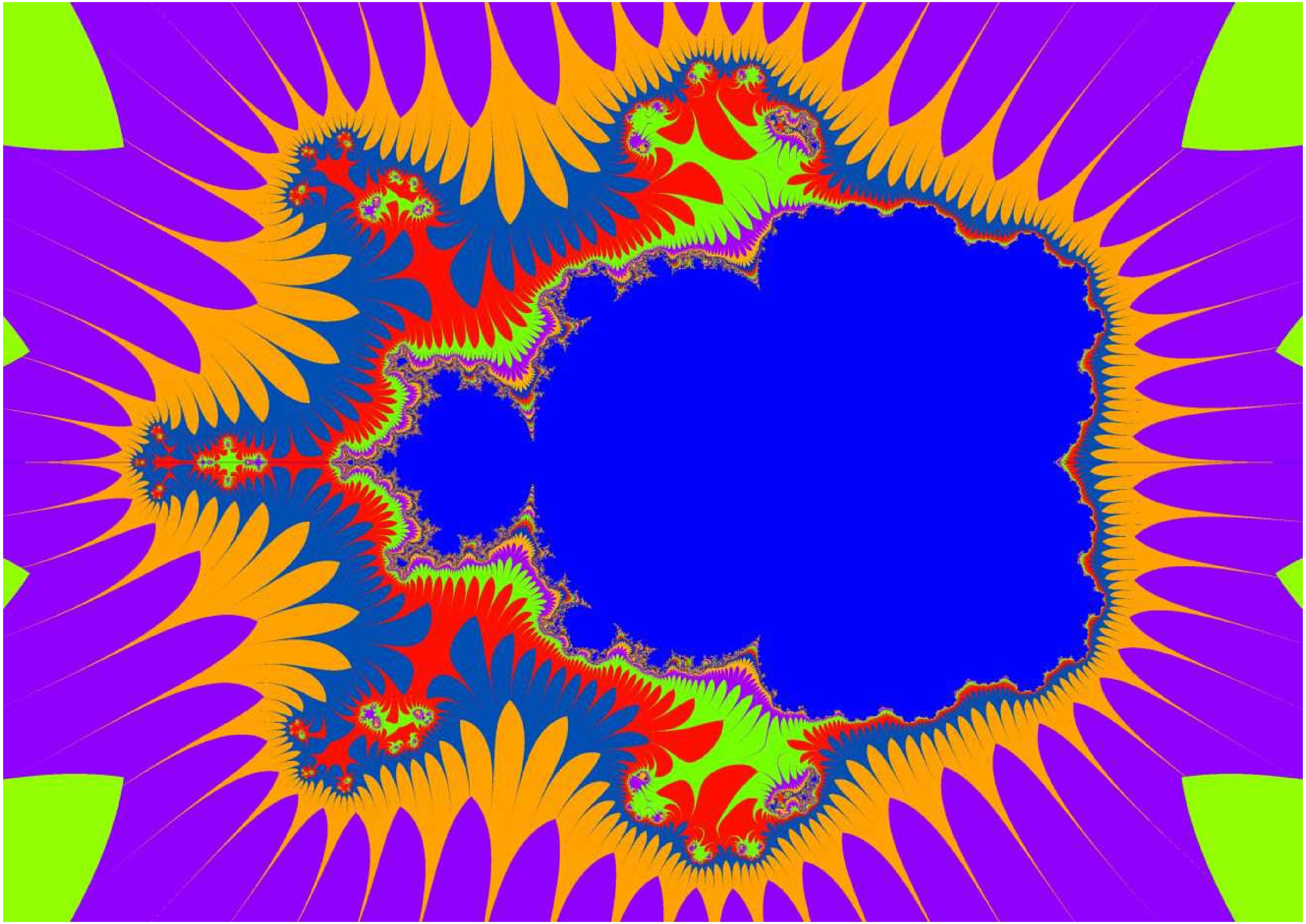
$$z_{n+1} = \log\left(z_n^2 + z_0\right)$$

7 steps to enlightenment $z_{n+1} = \tan^{-1}\left(z_n^2 + z_0\right)$



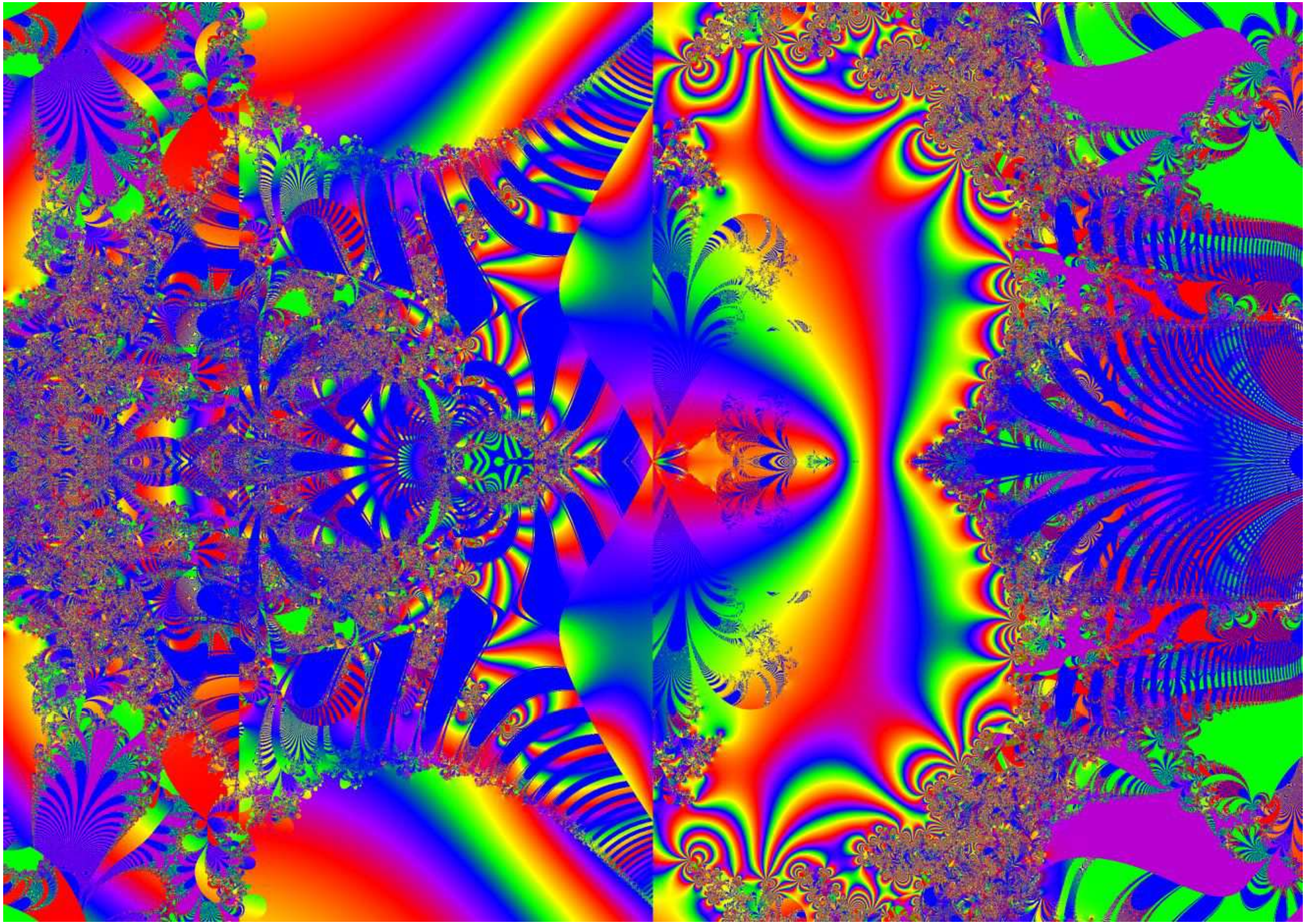
The Mandlerocket!

$$z_{n+1} = \sin^{-1}\left(z_n^2 + z_0\right)$$



Micro mandlebeast

$$z_{n+1} = \left(z_n^2 + z_0 \right)^2$$

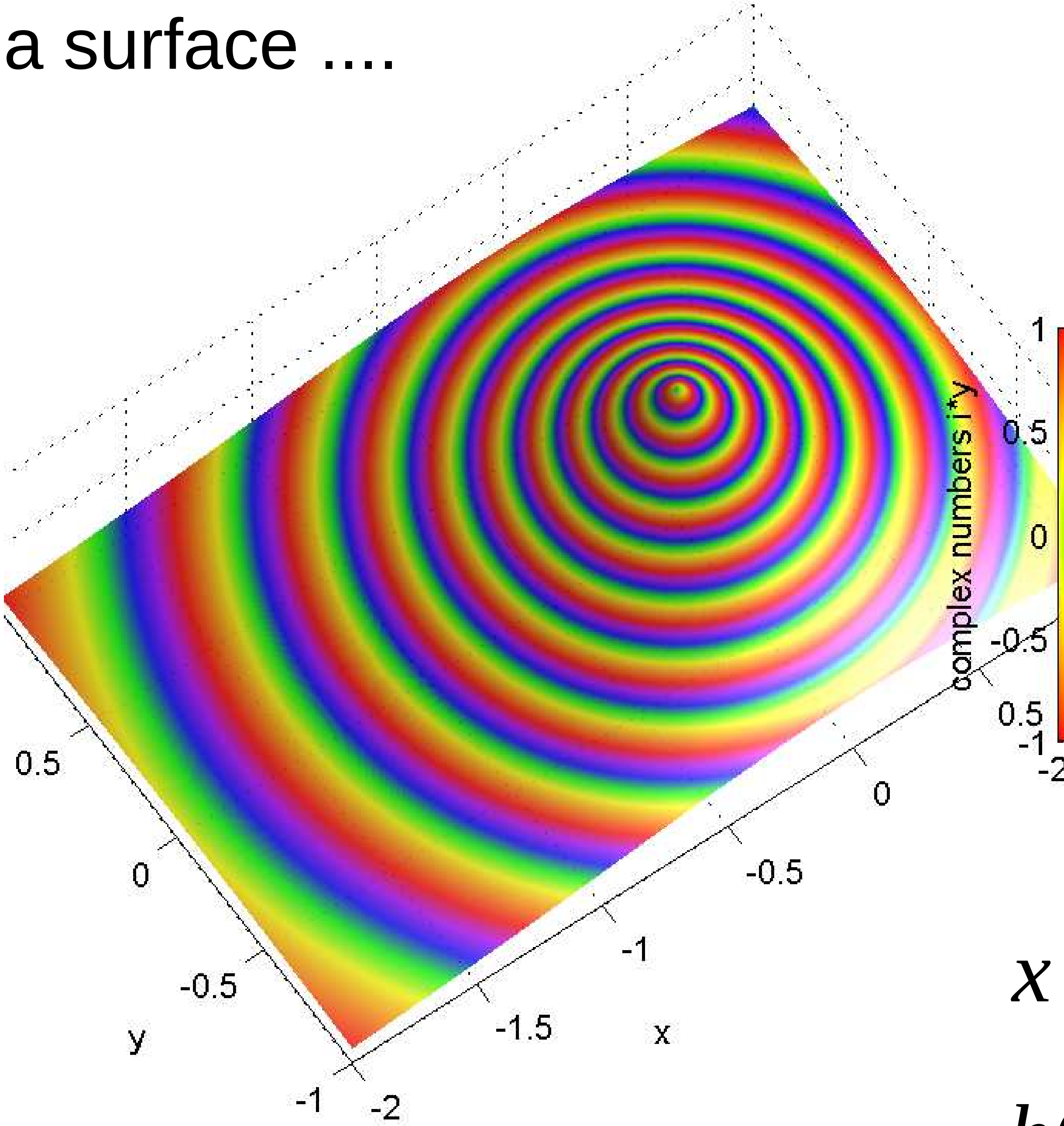


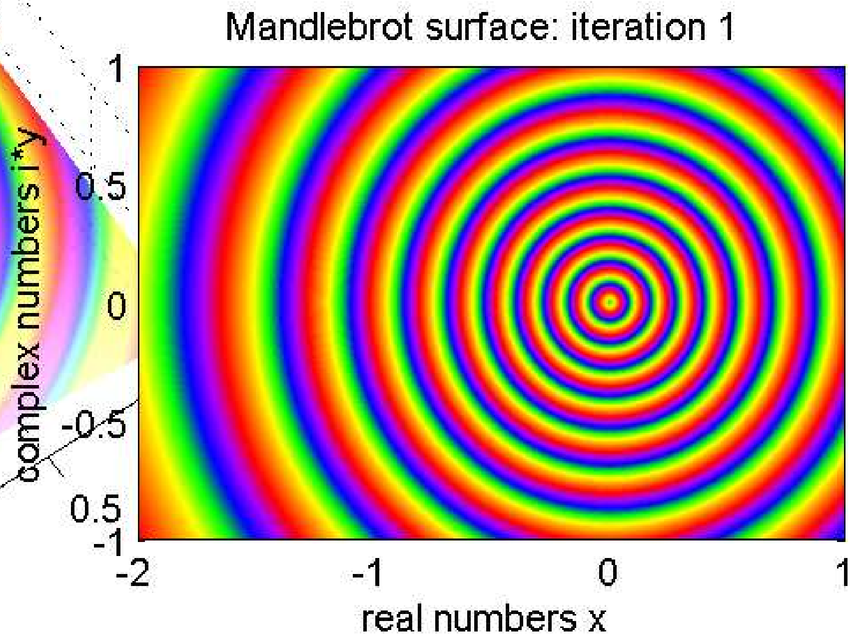
The profusion of power

$$z_{n+1} = \left(z_n^2 + z_0 \right)^{z_n}$$

Remember *h(x,y)* is a surface ....

$$z_{n+1} = z_n^2 + z_0$$

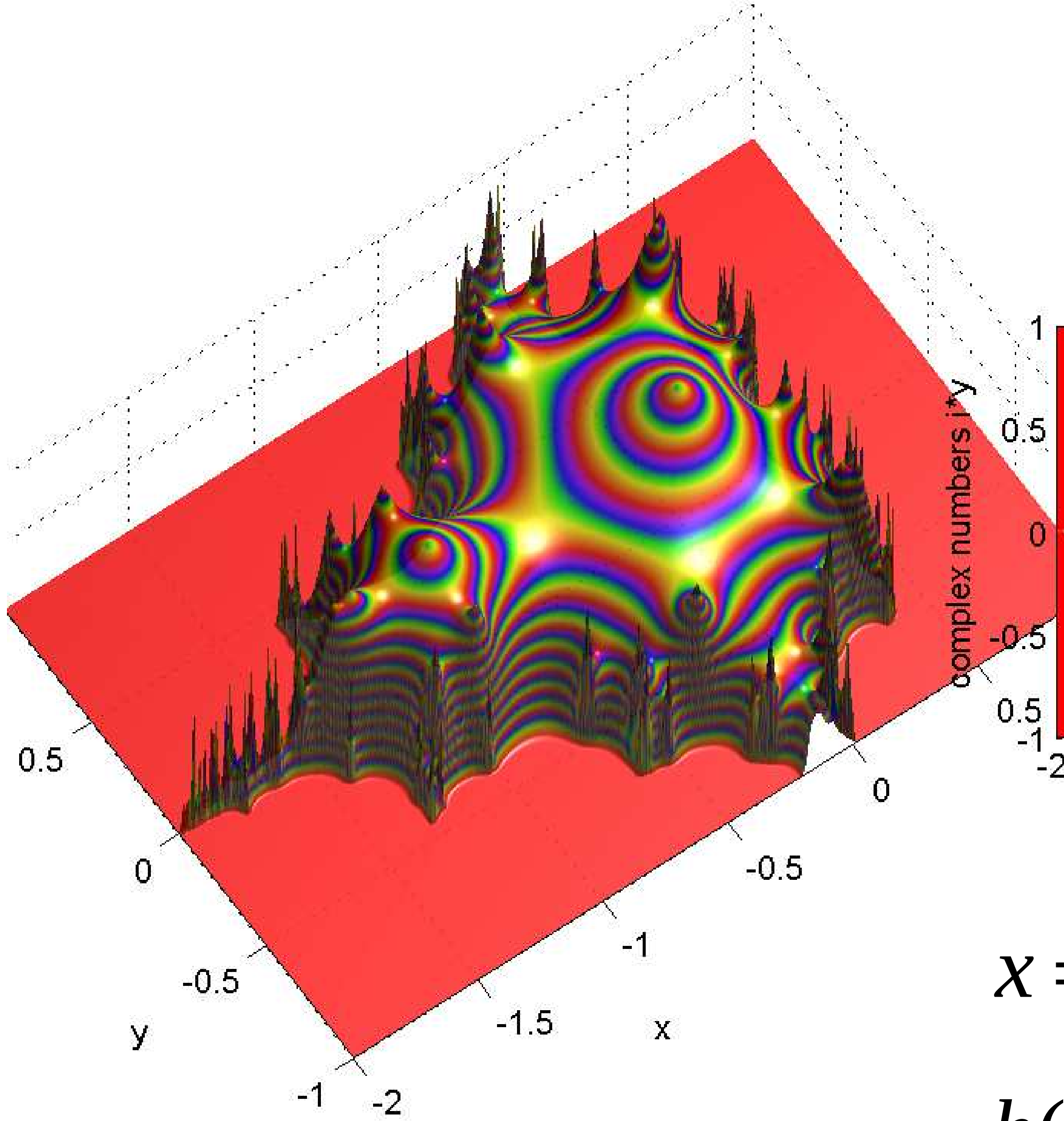


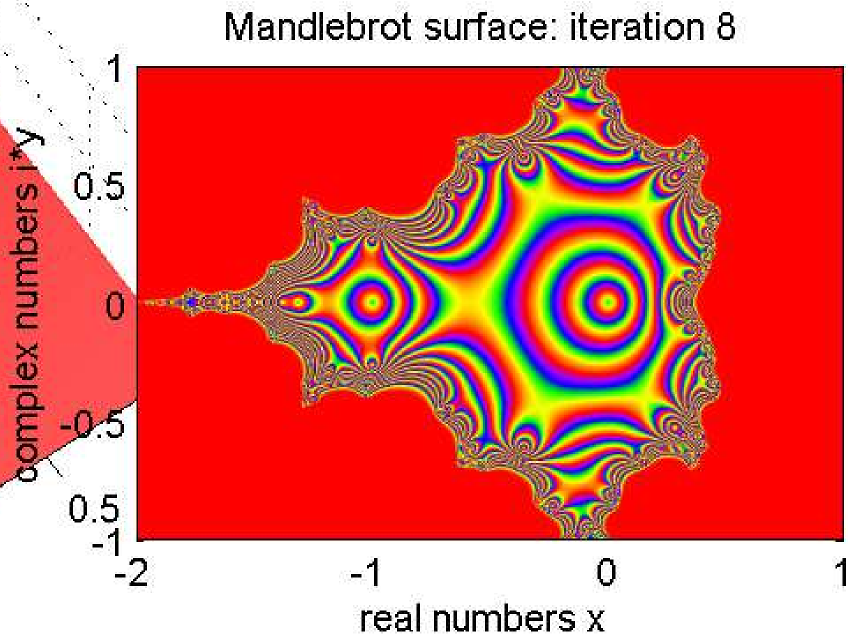


$$x = \mathrm{Re}(z), \quad y = \mathrm{Im}(z)$$

$$h(x, y) = e^{-\sqrt{x^2 + y^2}}$$

$$z_{n+1} = z_n^2 + z_0$$

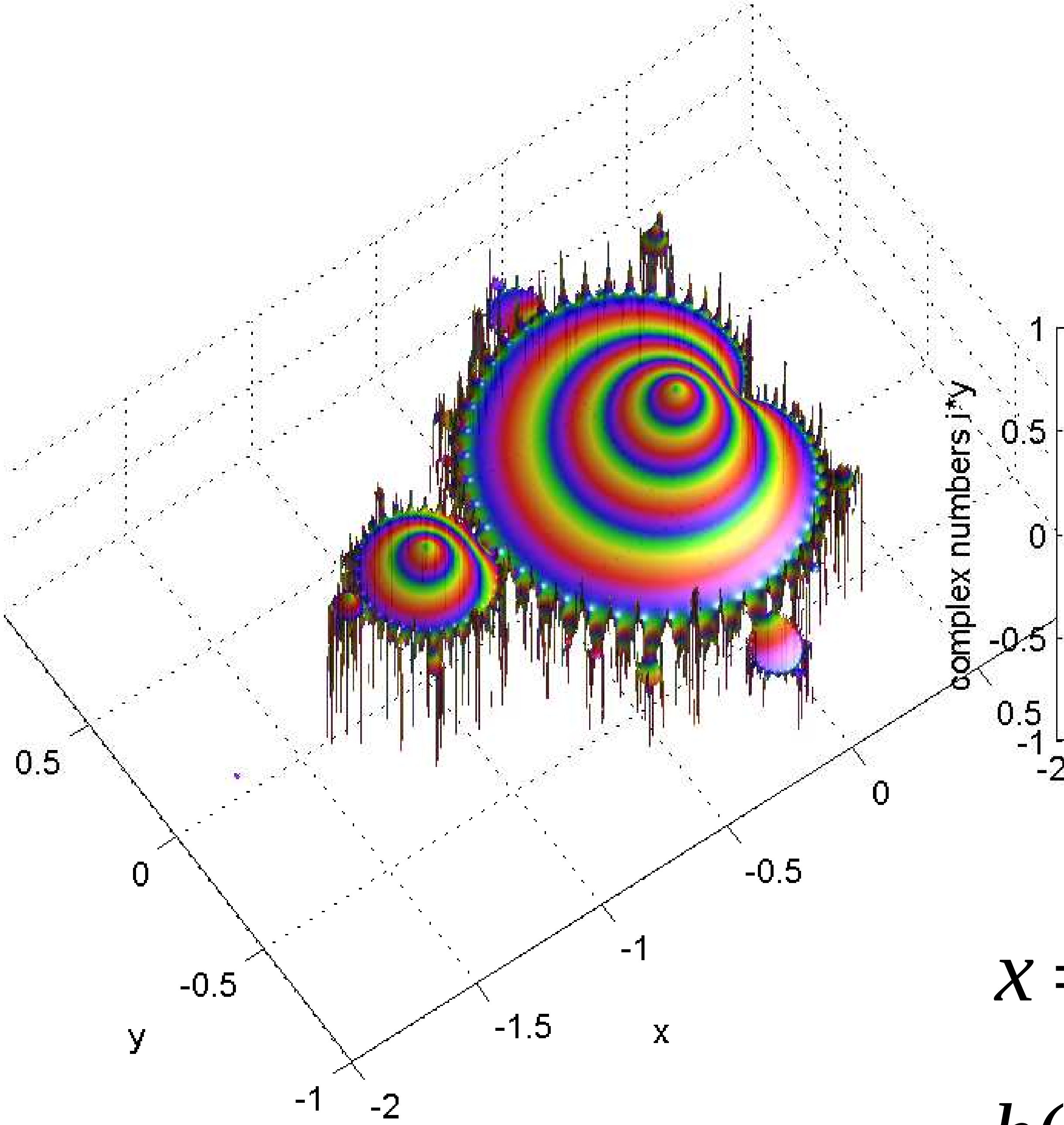


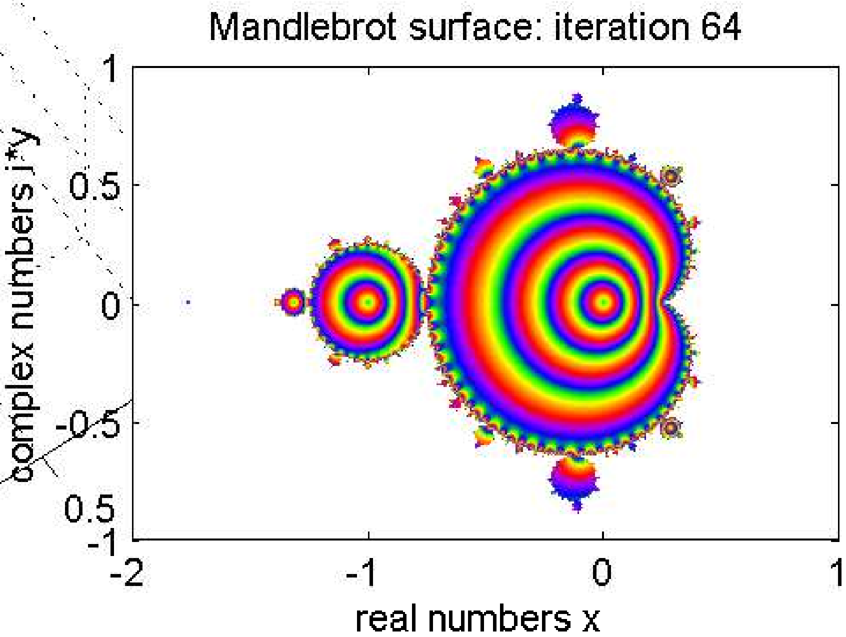


$$x = \mathrm{Re}(z), \quad y = \mathrm{Im}(z)$$

$$h(x,y) = e^{-\sqrt{x^2+y^2}}$$

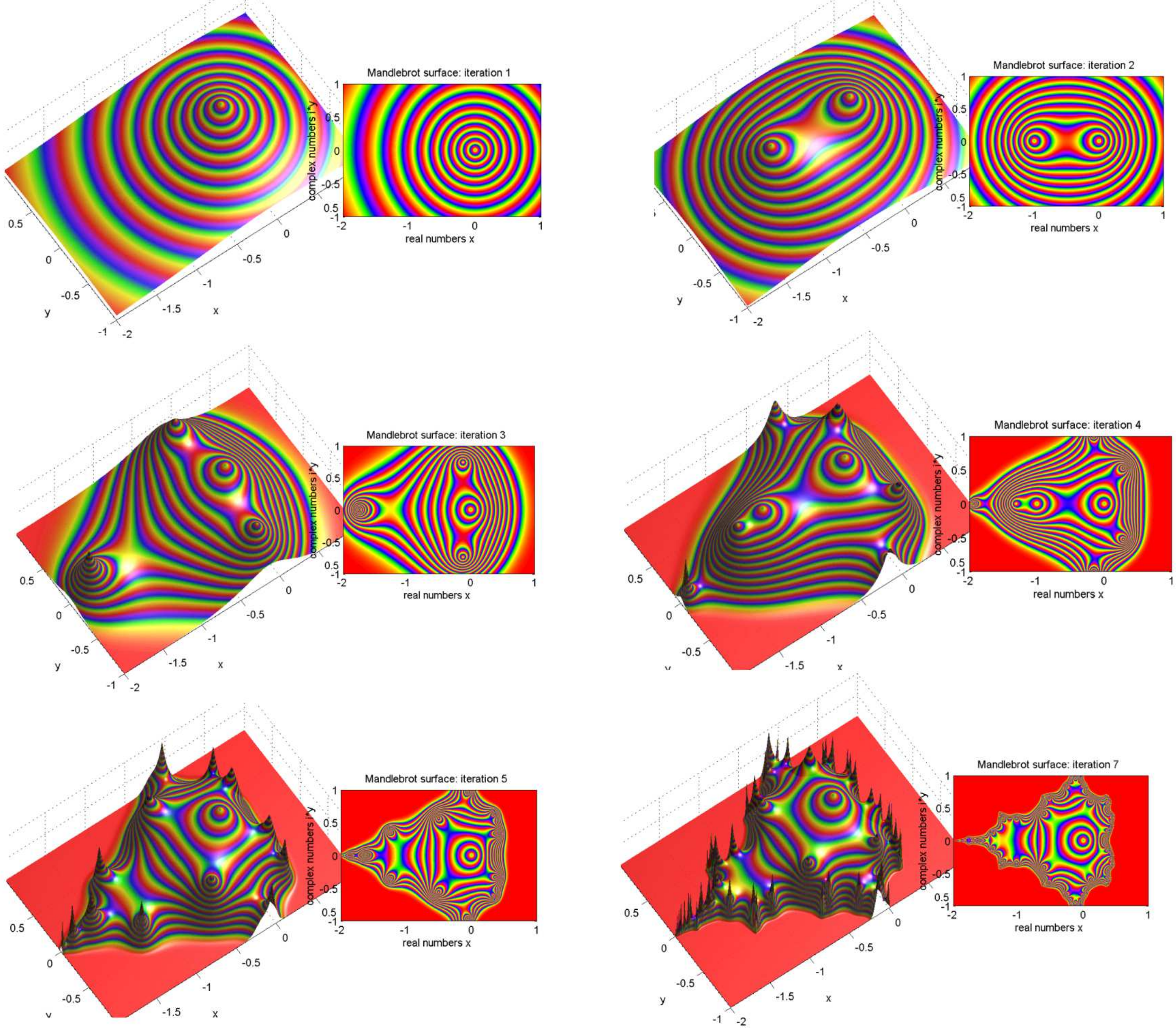
$$z_{n+1} = z_n^2 + z_0$$



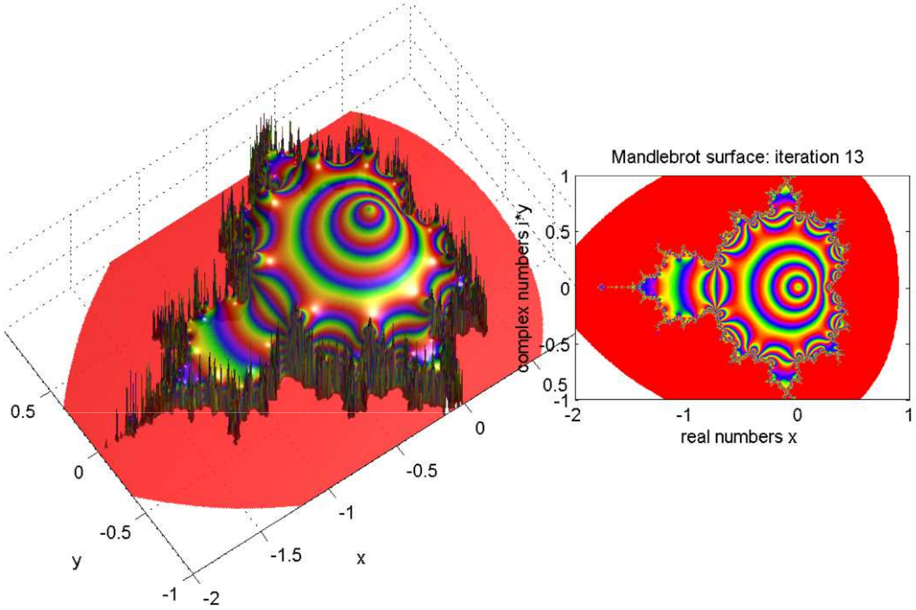


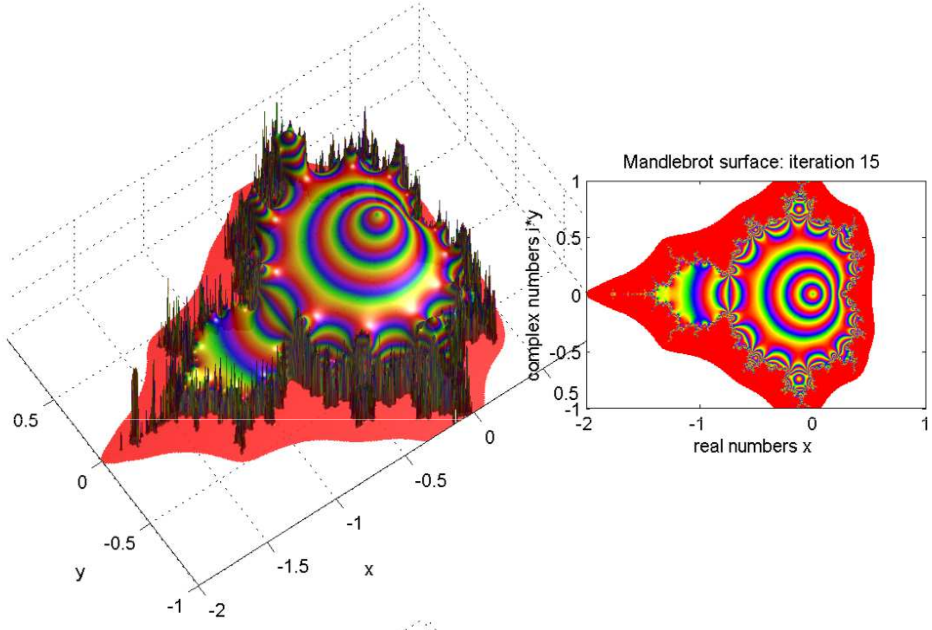
$$x = \mathrm{Re}(z), \quad y = \mathrm{Im}(z)$$

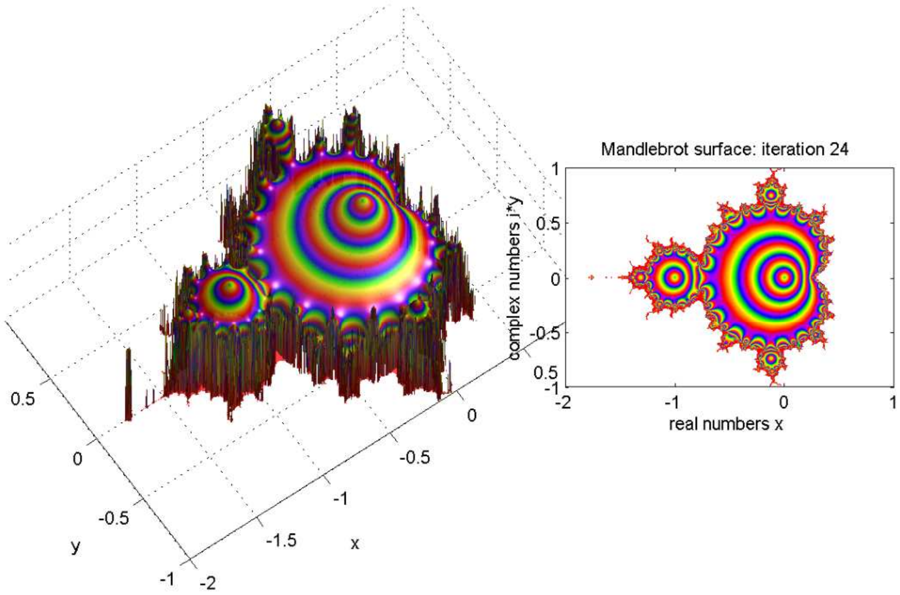
$$h(x,y) = e^{-\sqrt{x^2+y^2}}$$

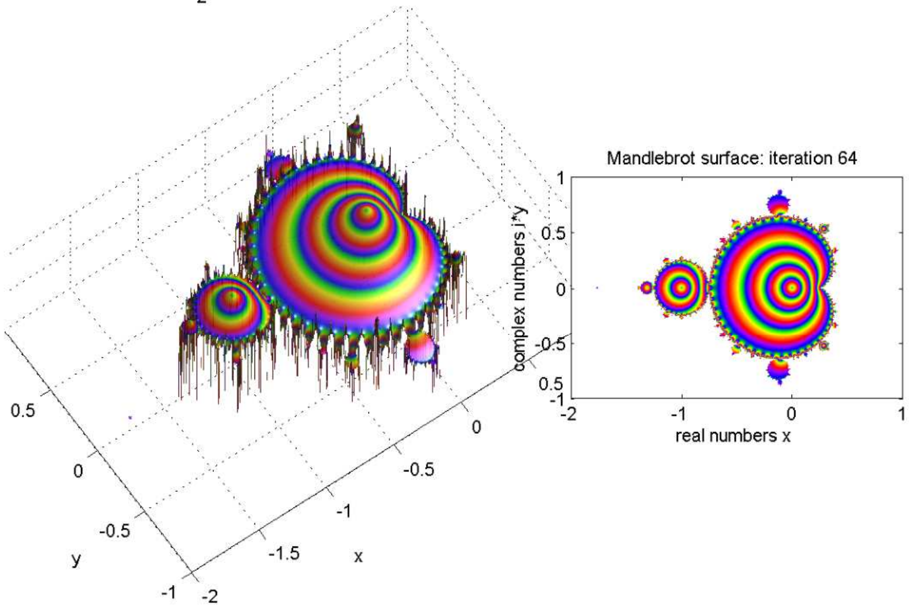


Mandelbrot surface: iteration 1
Mandelbrot surface: iteration 2
Mandelbrot surface: iteration 3
Mandelbrot surface: iteration 4
Mandelbrot surface: iteration 5
Mandelbrot surface: iteration 7
complex numbers i*y
real numbers x
x
y





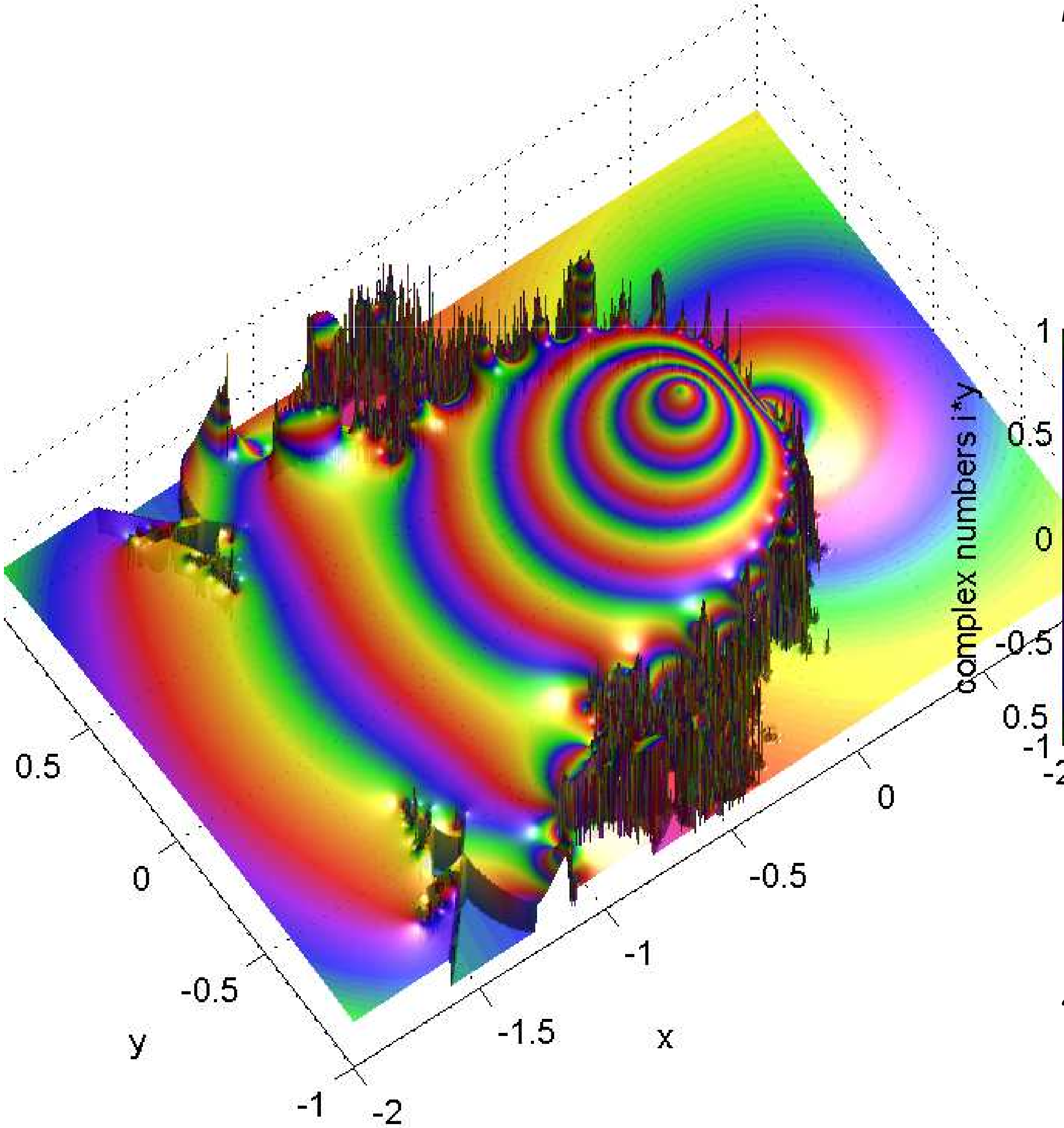


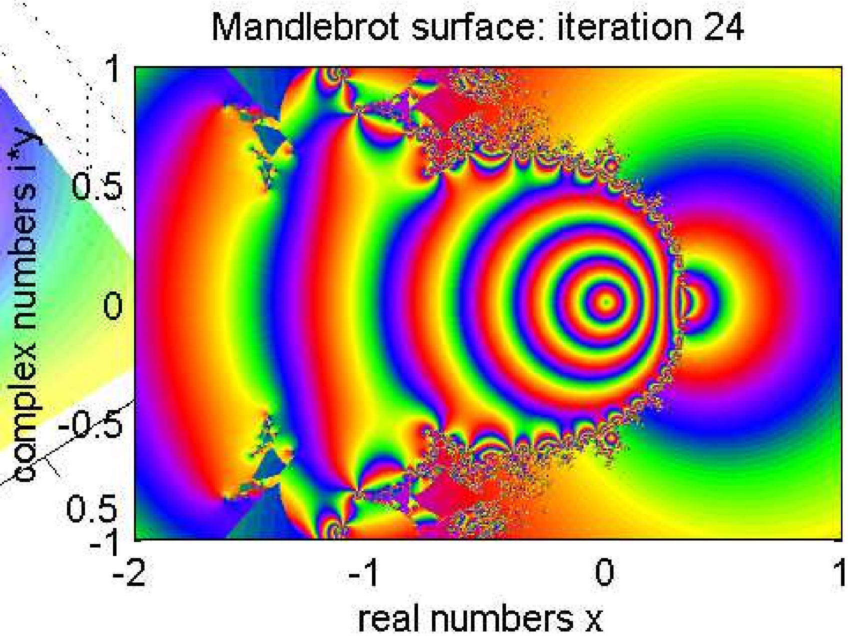


Selection from *Day of Julia.* Mathematicon Exhibition, 2014

7 steps to enlightenment

$$z_{n+1} = \tan^{-1}\left(z_n^2 + z_0\right)$$



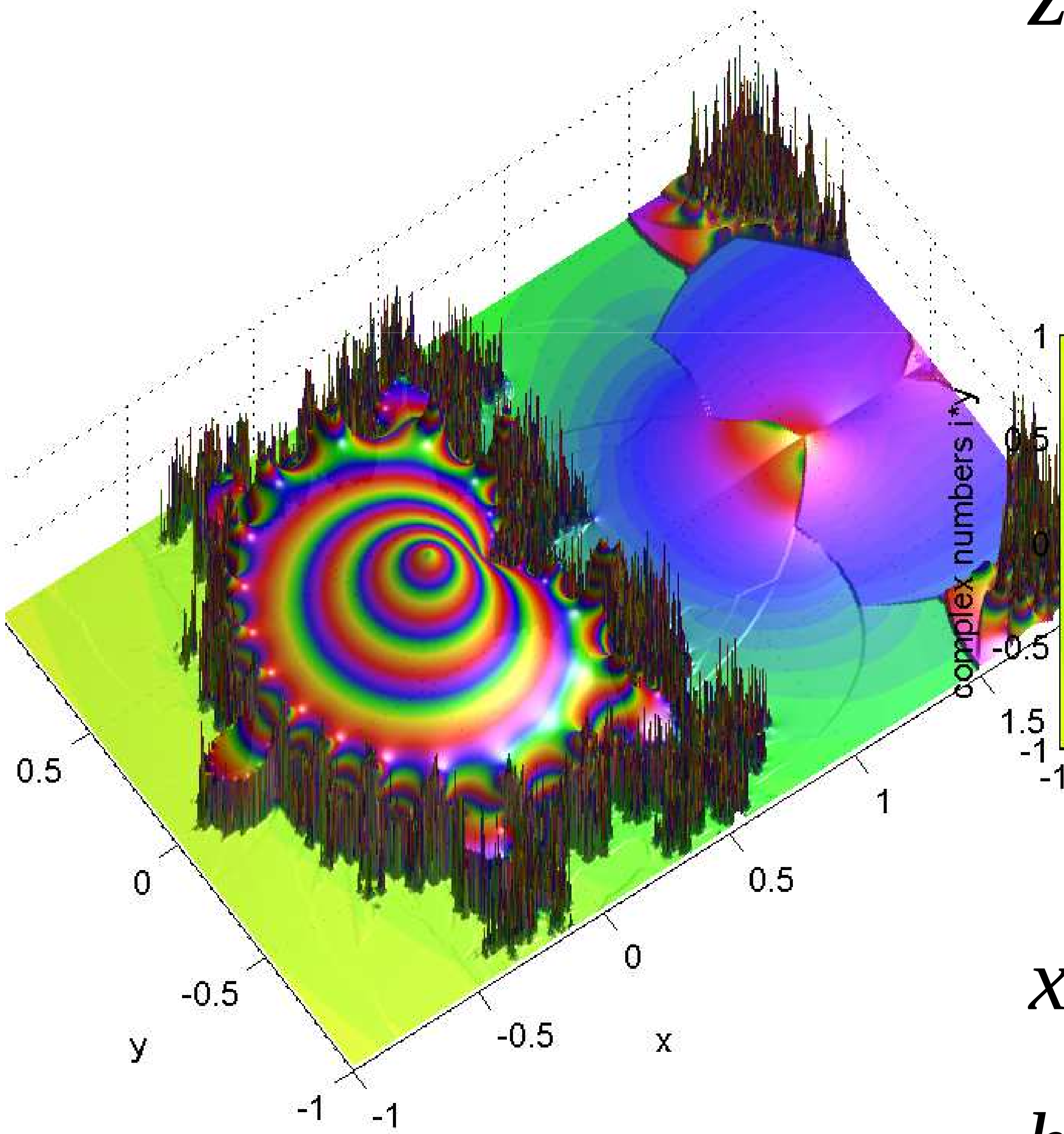


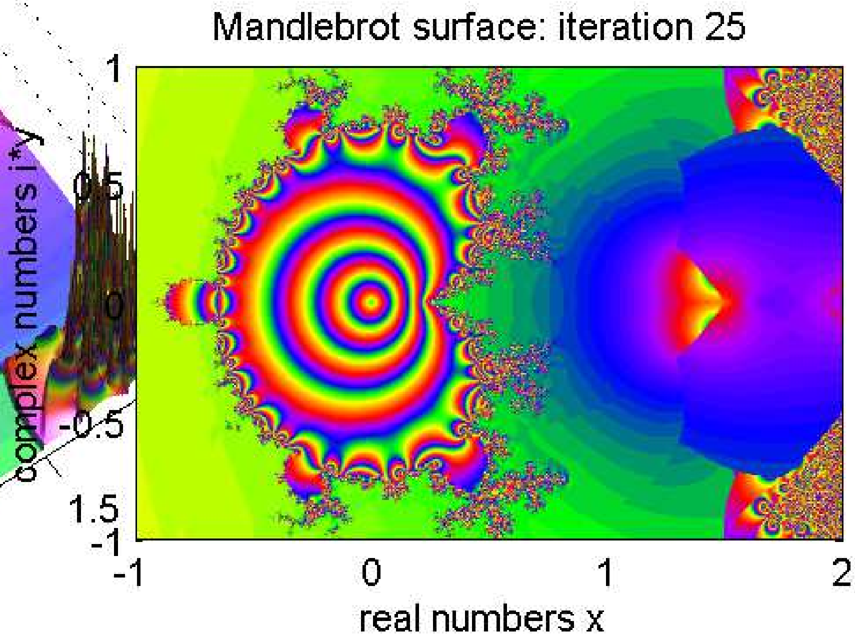
$$x = \mathrm{Re}(z), \quad y = \mathrm{Im}(z)$$

$$h(x, y) = e^{-\sqrt{x^2 + y^2}}$$

The Mandlerocket

$$z_{n+1} = \sin^{-1}\left(z_n^2 + z_0\right)$$

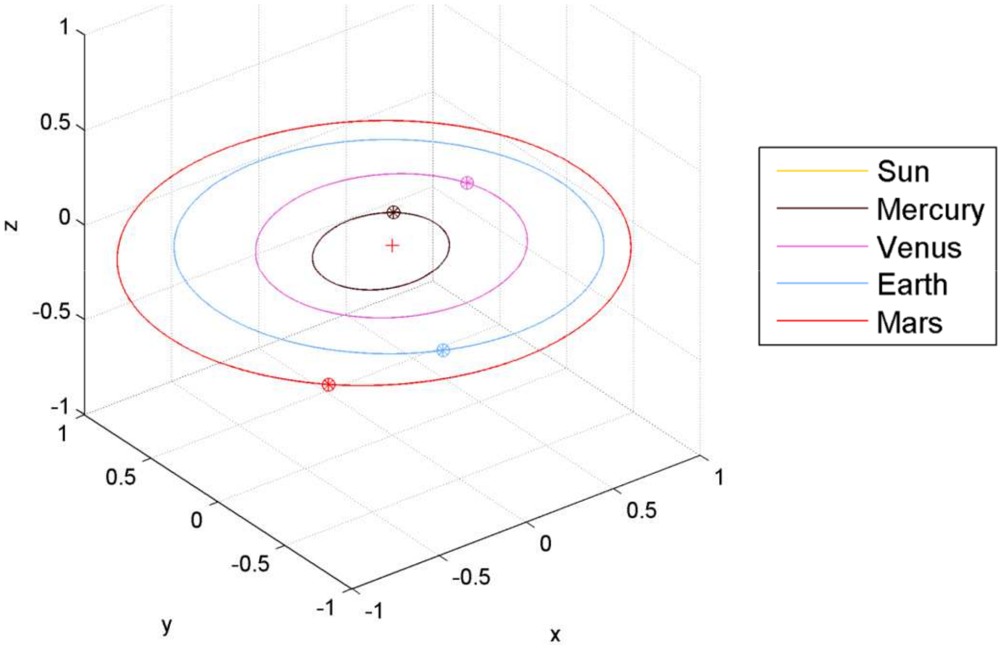


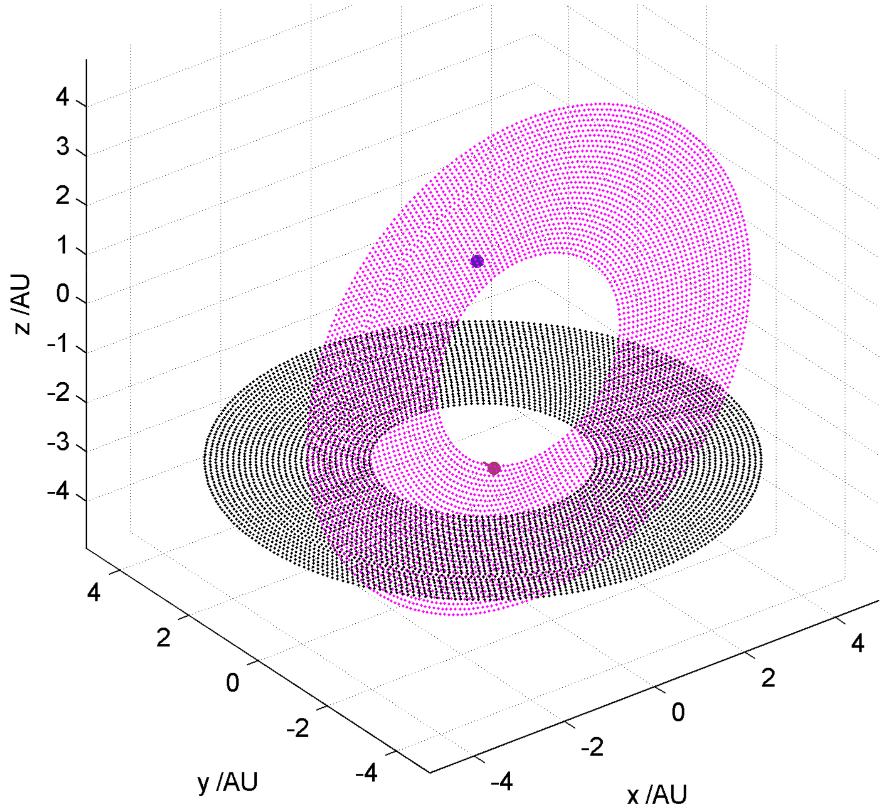


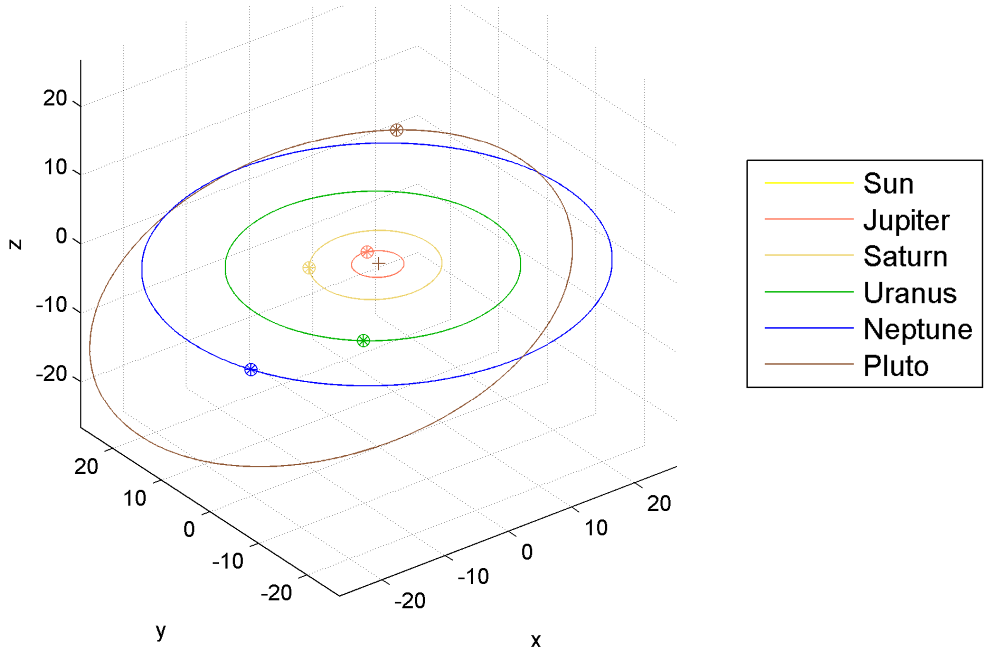
$$x = \mathrm{Re}(z), \quad y = \mathrm{Im}(z)$$

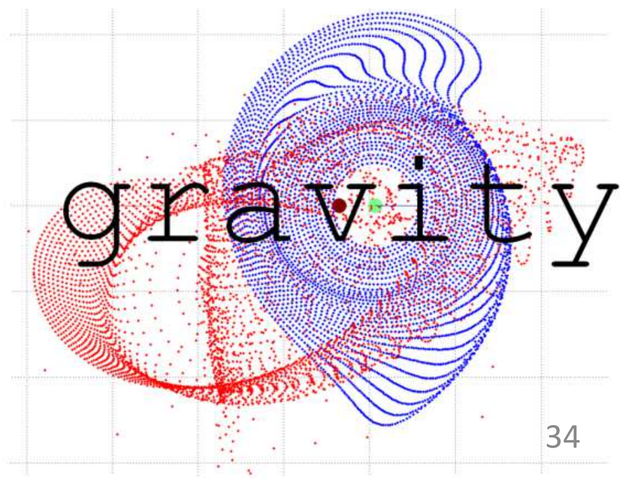
$$h(x,y) = e^{-\sqrt{x^2+y^2}}$$

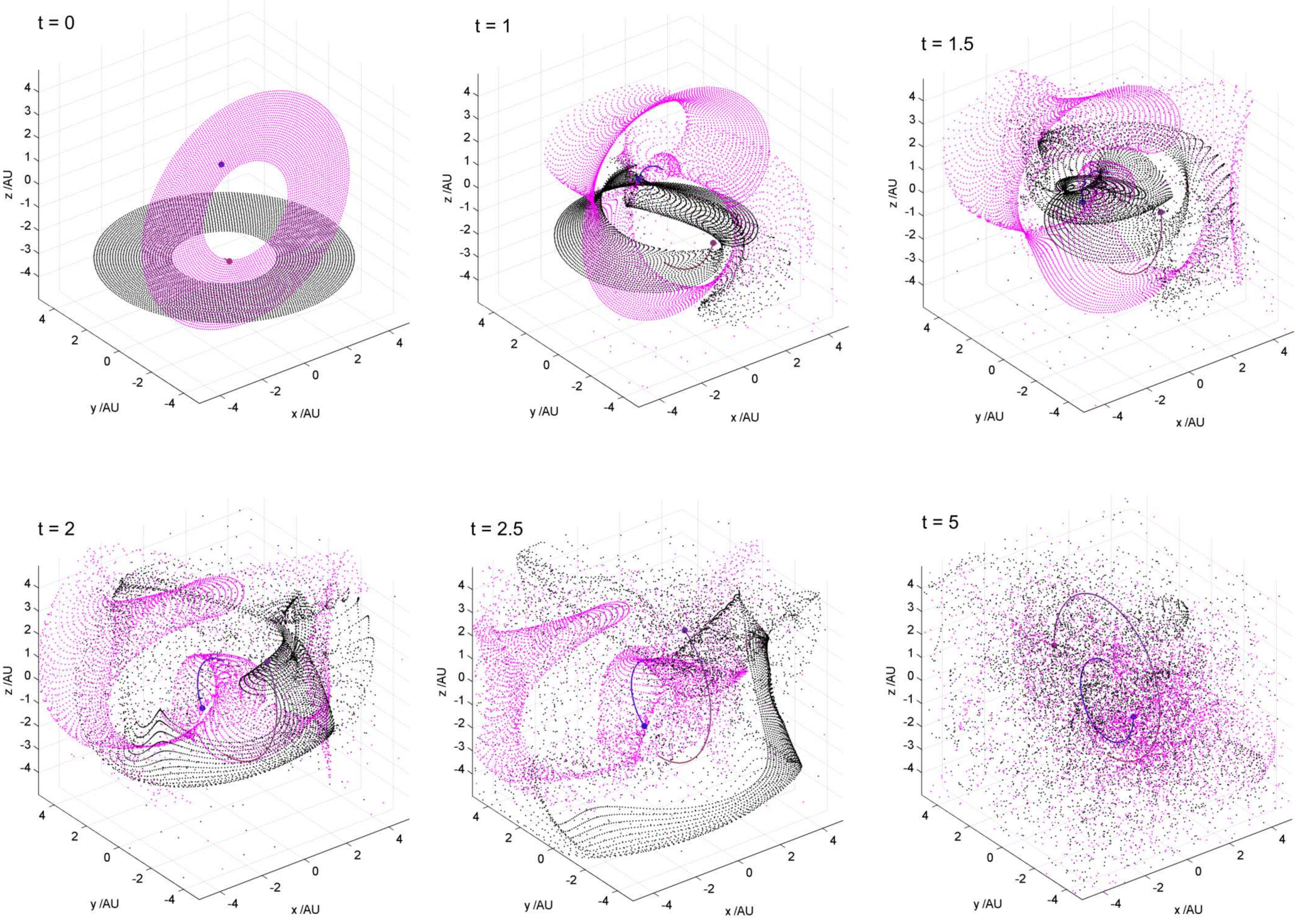
$$m_i \frac{d\mathbf{v}_i}{dt} = A m_i \sum_{j \neq i} \frac{m_j \left(\mathbf{r}_j - \mathbf{r}_i\right)}{\left|\mathbf{r}_j - \mathbf{r}_i\right|^{P+1}} - B m_i \sum_{j \neq i} R_j \frac{m_j \left(\mathbf{r}_j - \mathbf{r}_i\right)}{\left|\mathbf{r}_j - \mathbf{r}_i\right|^{Q+1}}$$

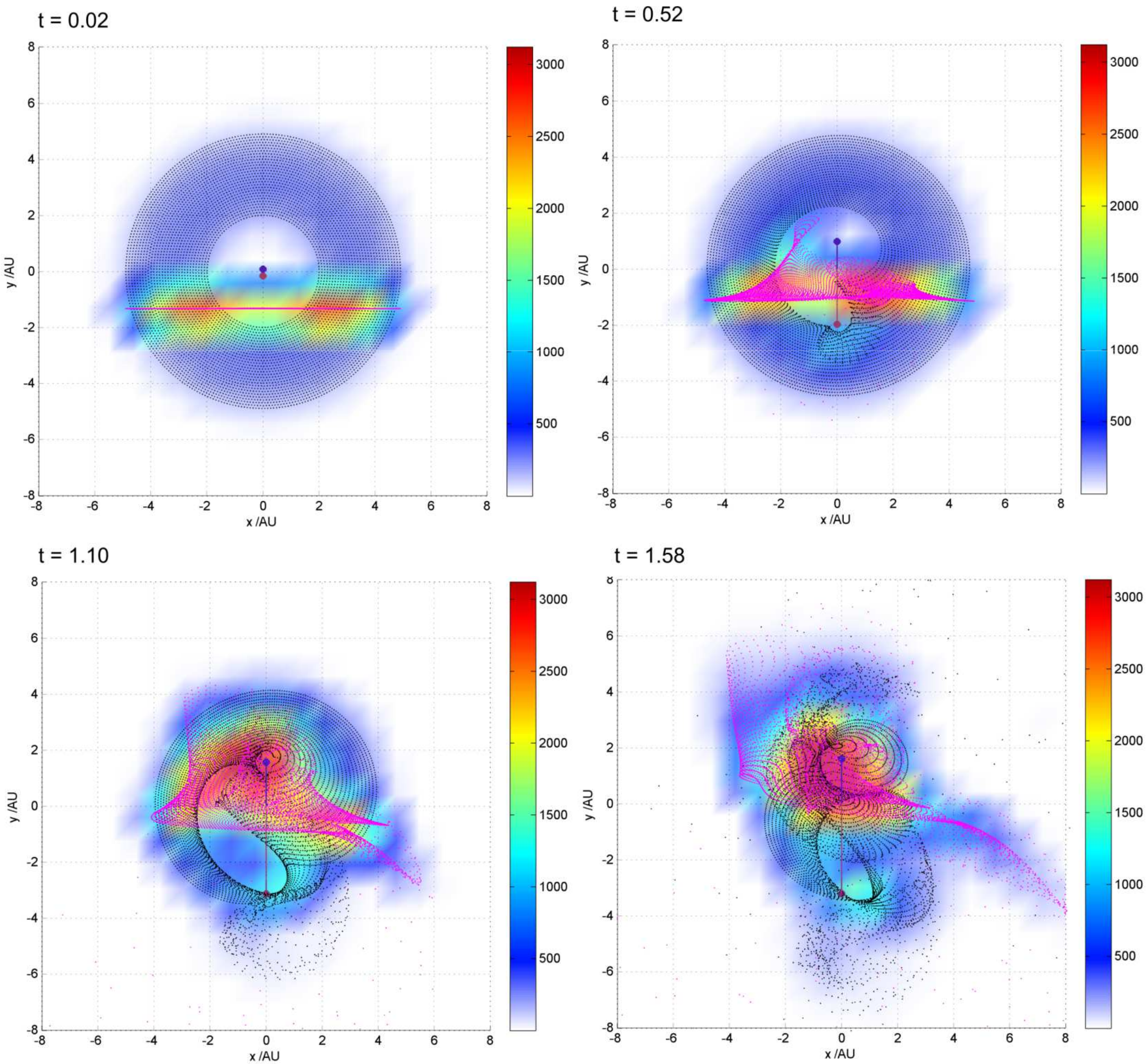









t = 0
t = 1
t = 1.5
t = 2
t = 2.5
t = 5
z /AU
y /AU
x /AU
4
3
2
1
0
-1
-2
-3
-4

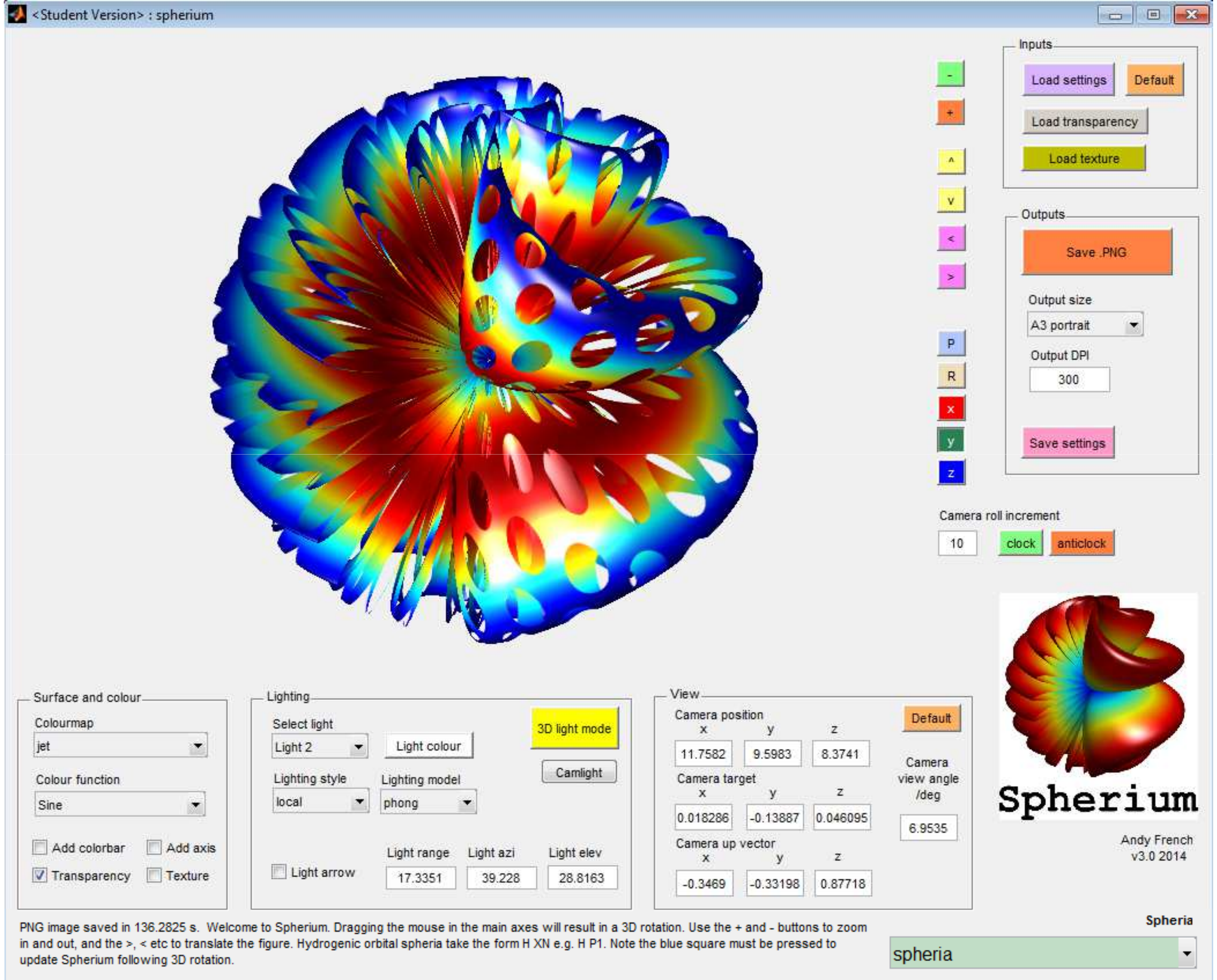


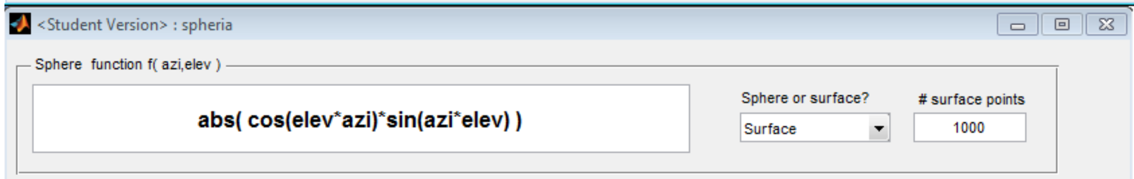
Movie

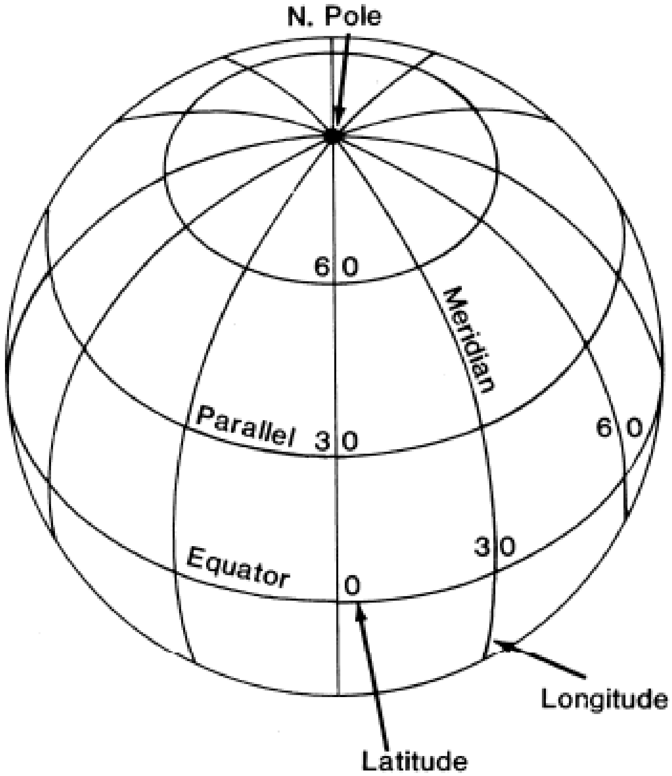
Movie
HD

spherium

$$R = f(\text{azi}, \text{elev})$$



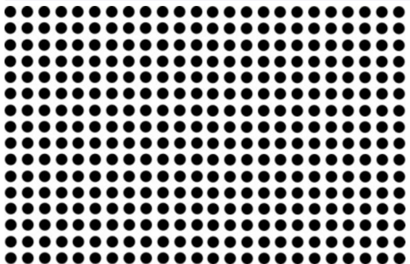




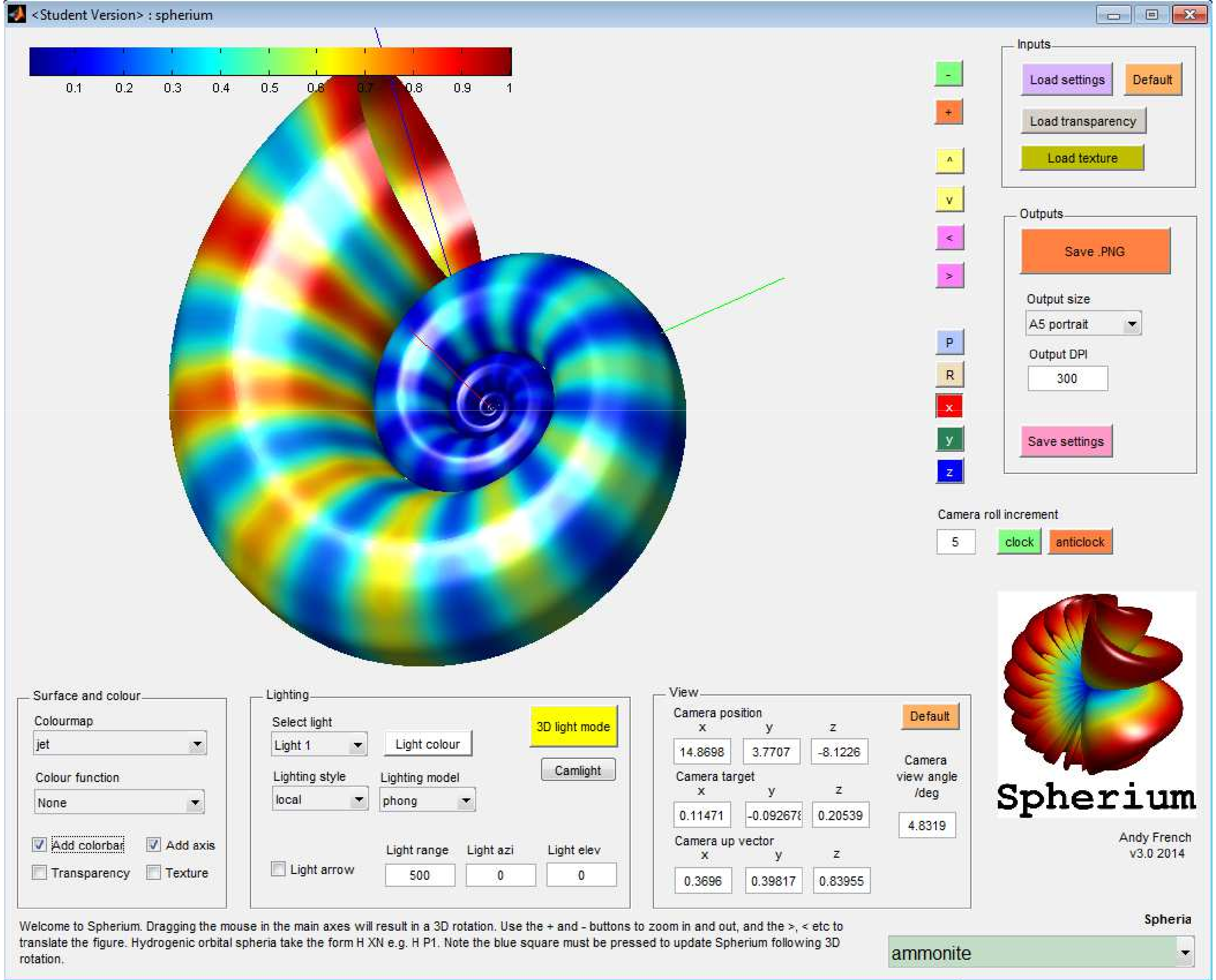
R = radius

azi = longitude

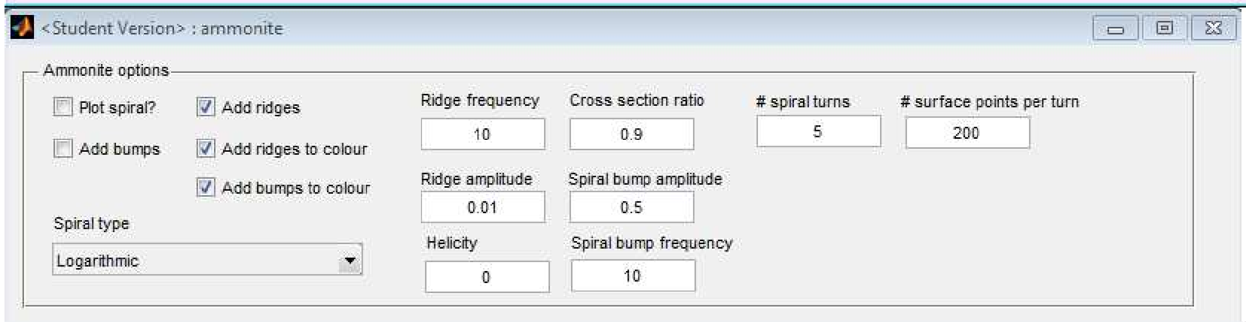
elev = latitude

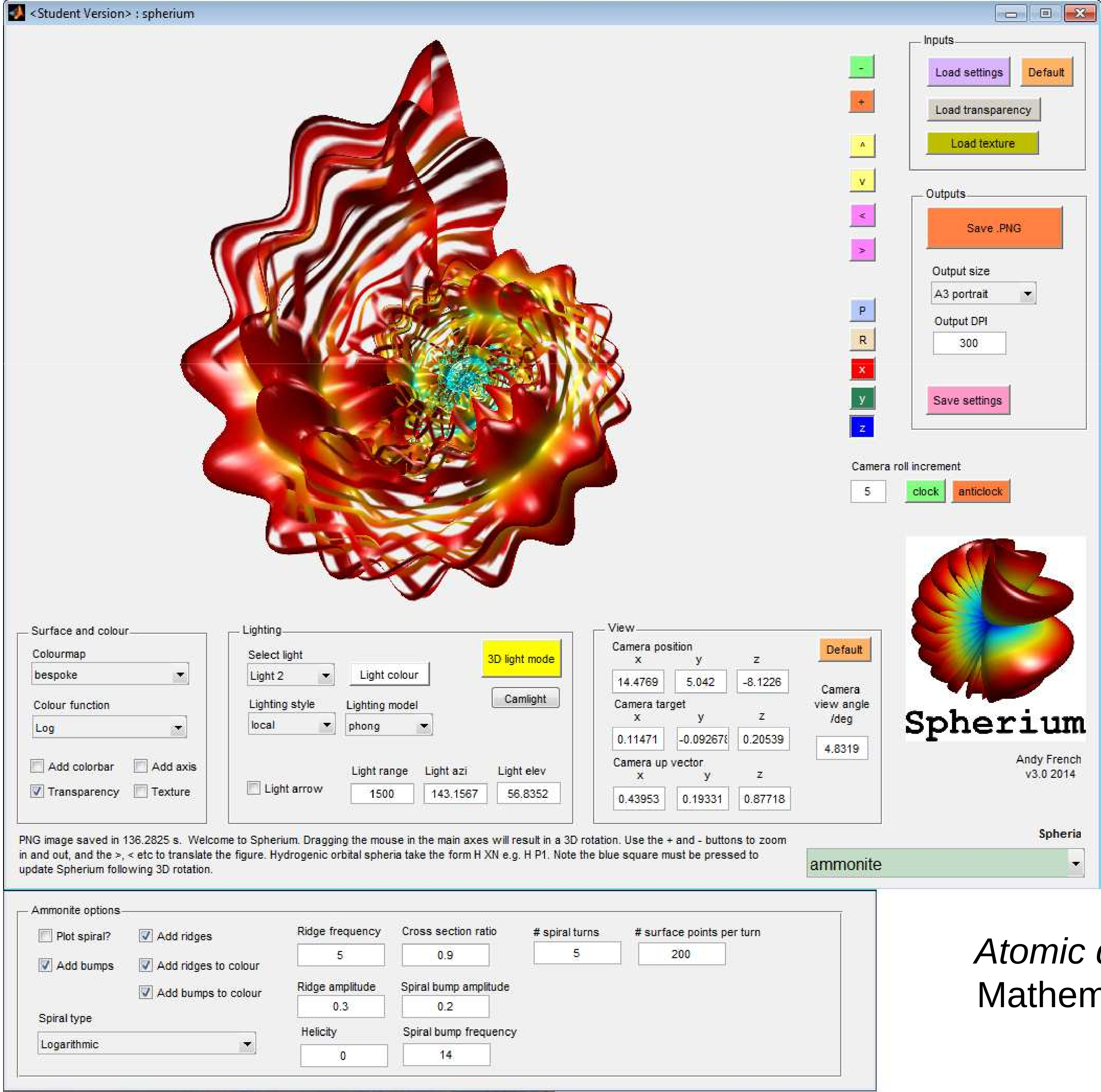


Transparency



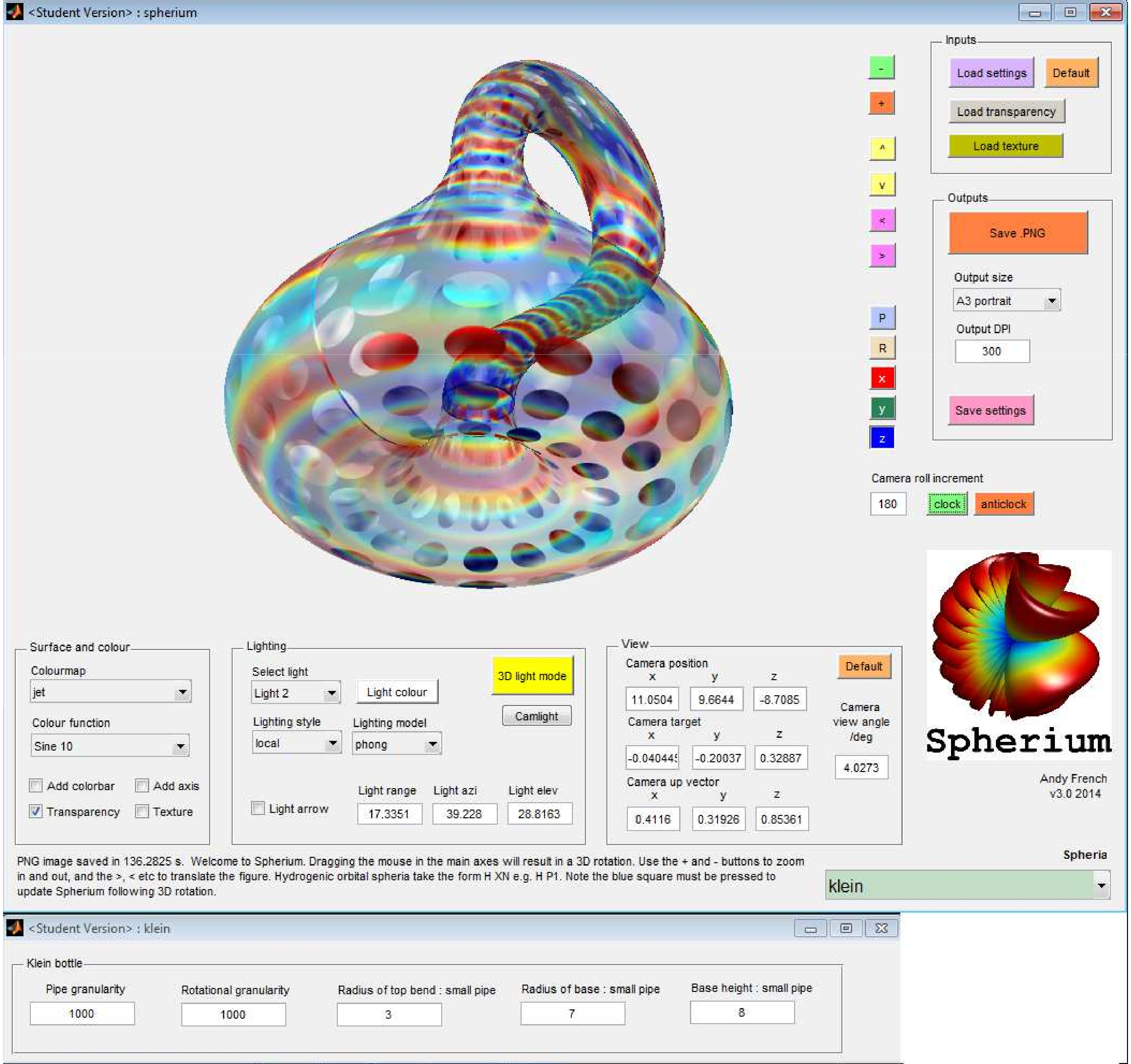
JURASSIC COAST Trust

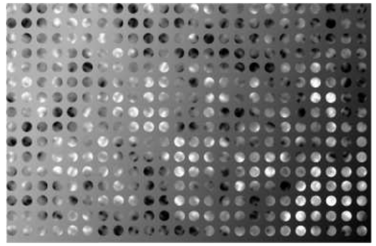




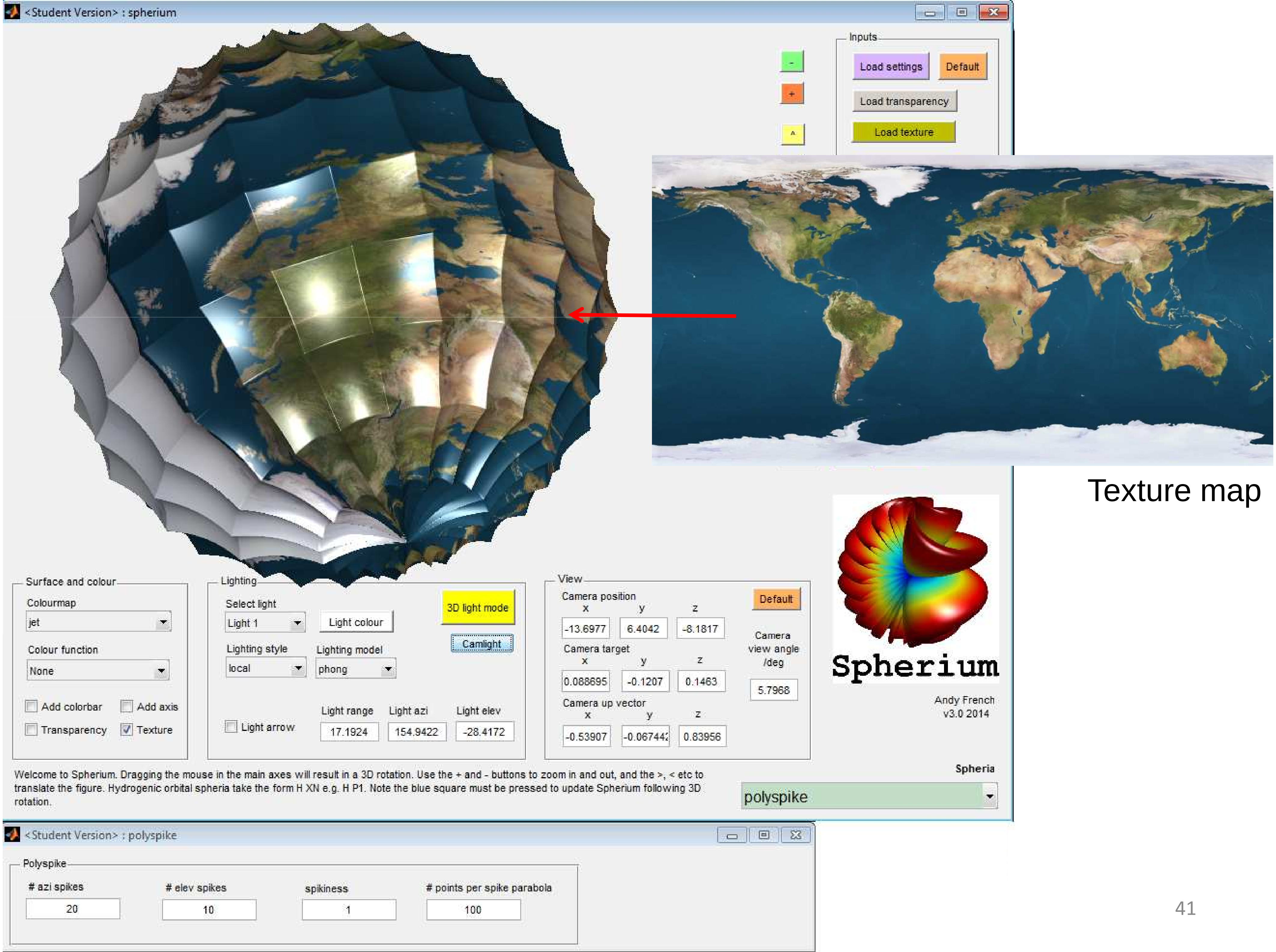
Atomic dragon spiral

Mathematicon, 2014

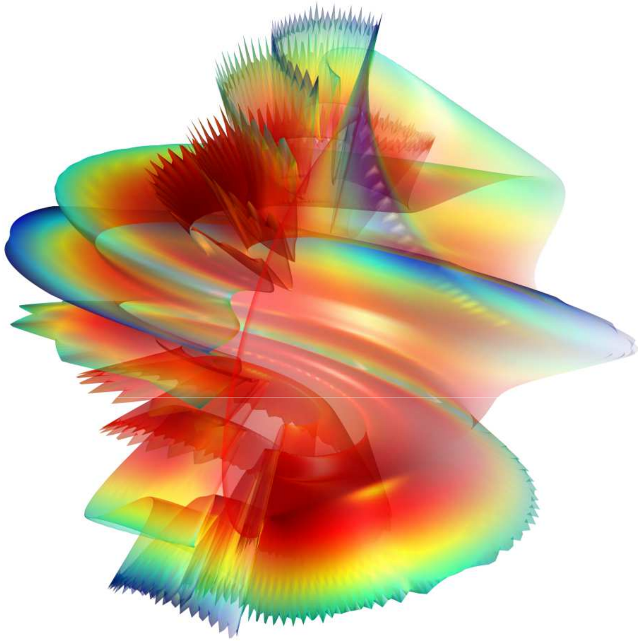


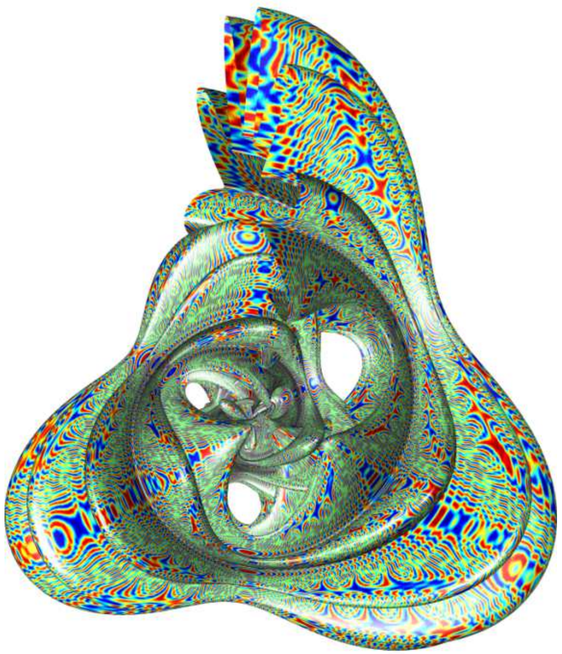


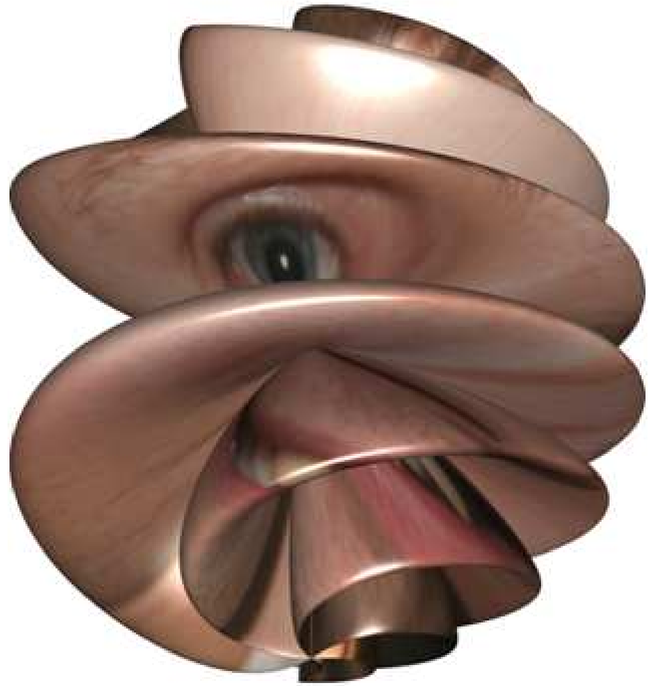
Klein bottle with cloudy holes transparency map


<Student Version> : spherium
Inputs
Load settings
Default
Load transparency
Load texture
Spherium
Andy French
v3.0 2014
Surface and colour
Colourmap
jet
Colour function
None
Add colorbar
Add axis
Transparency
Texture
Lighting
Select light
Light 1
Light colour
3D light mode
Camlight
Lighting style
local
Lighting model
phong
Light arrow
Light range
Light azi
Light elev
17.1924
154.9422
-28.4172
View
Camera position
x
y
z
-13.6977
6.4042
-8.1817
Default
Camera target
0.088695
-0.1207
0.1463
Camera view angle /deg
5.7968
Camera up vector
-0.53907
-0.067442
0.83956
Welcome to Spherium. Dragging the mouse in the main axes will result in a 3D rotation. Use the + and - buttons to zoom in and out, and the >, < etc to translate the figure. Hydrogenic orbital spheria take the form H XN e.g. H P1. Note the blue square must be pressed to update Spherium following 3D rotation.
Spheria
polyspike
<Student Version> : polyspike
Polyspike
azi spikes
20
elev spikes
10
spikiness
1
points per spike parabola
100

Texture map



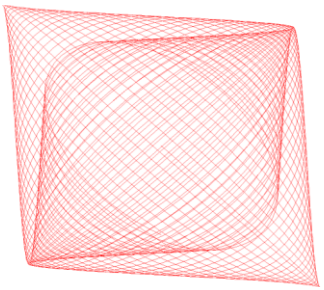



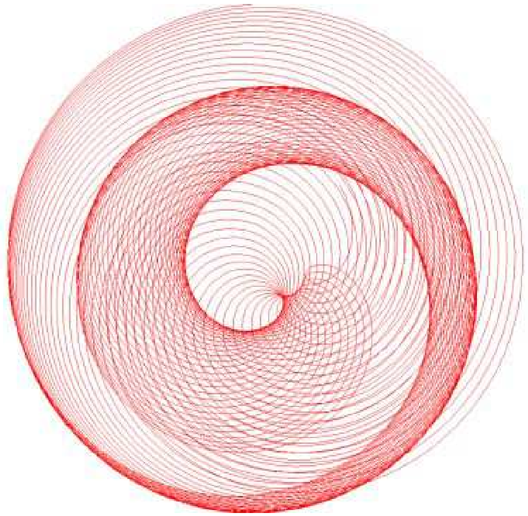


harmonograph

- The Harmonograph was a Victorian curiosity attributed to Professor Blackburn in 1844

- Use two or three pendulums to create strange and beautiful patterns





Example of a *lateral* harmonograph

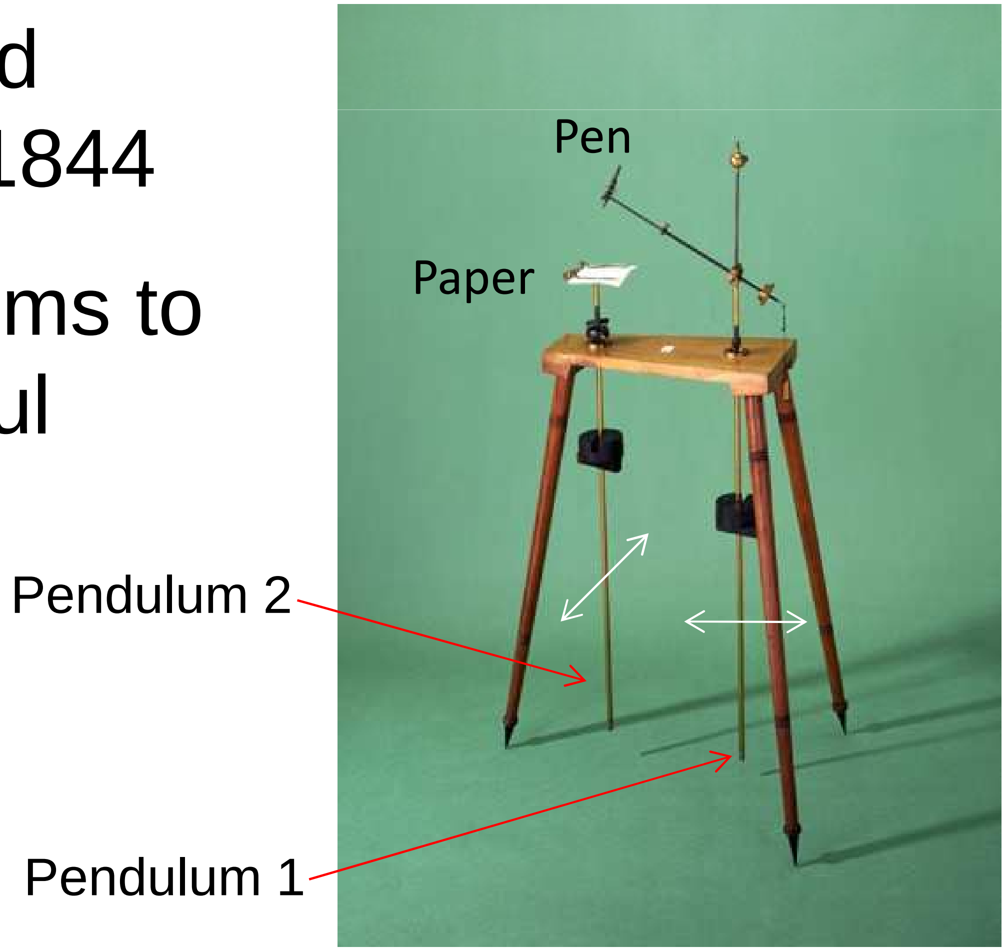
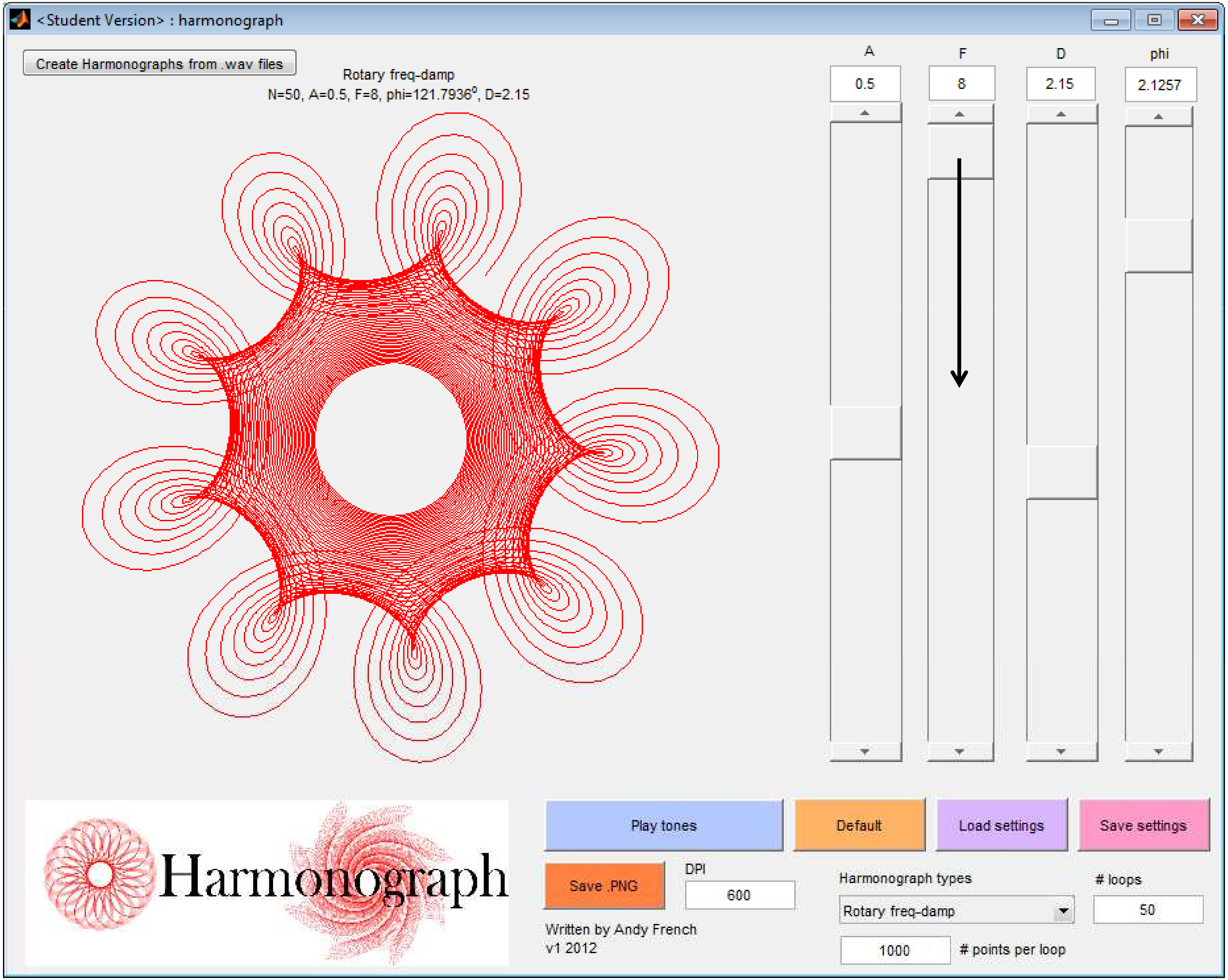
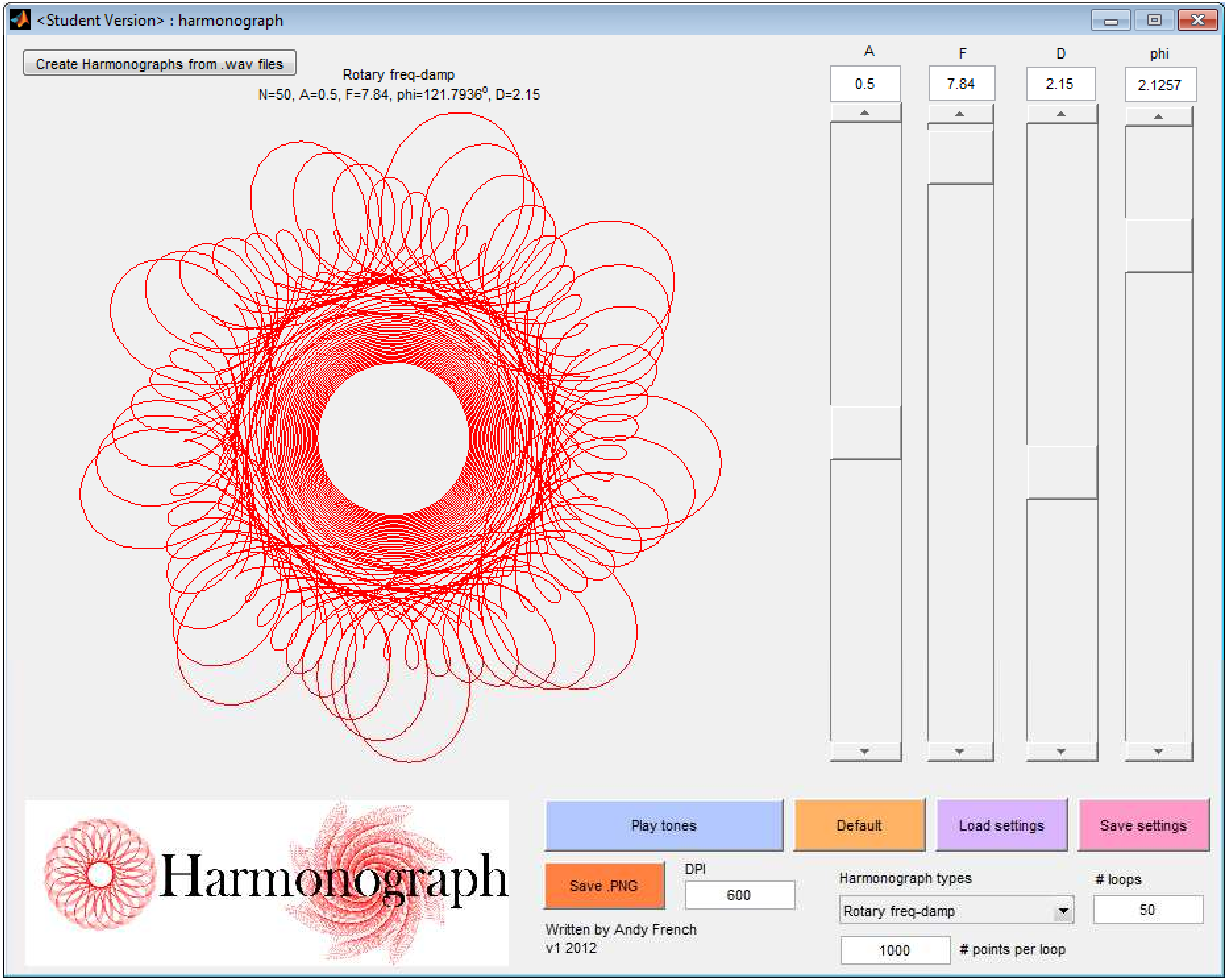


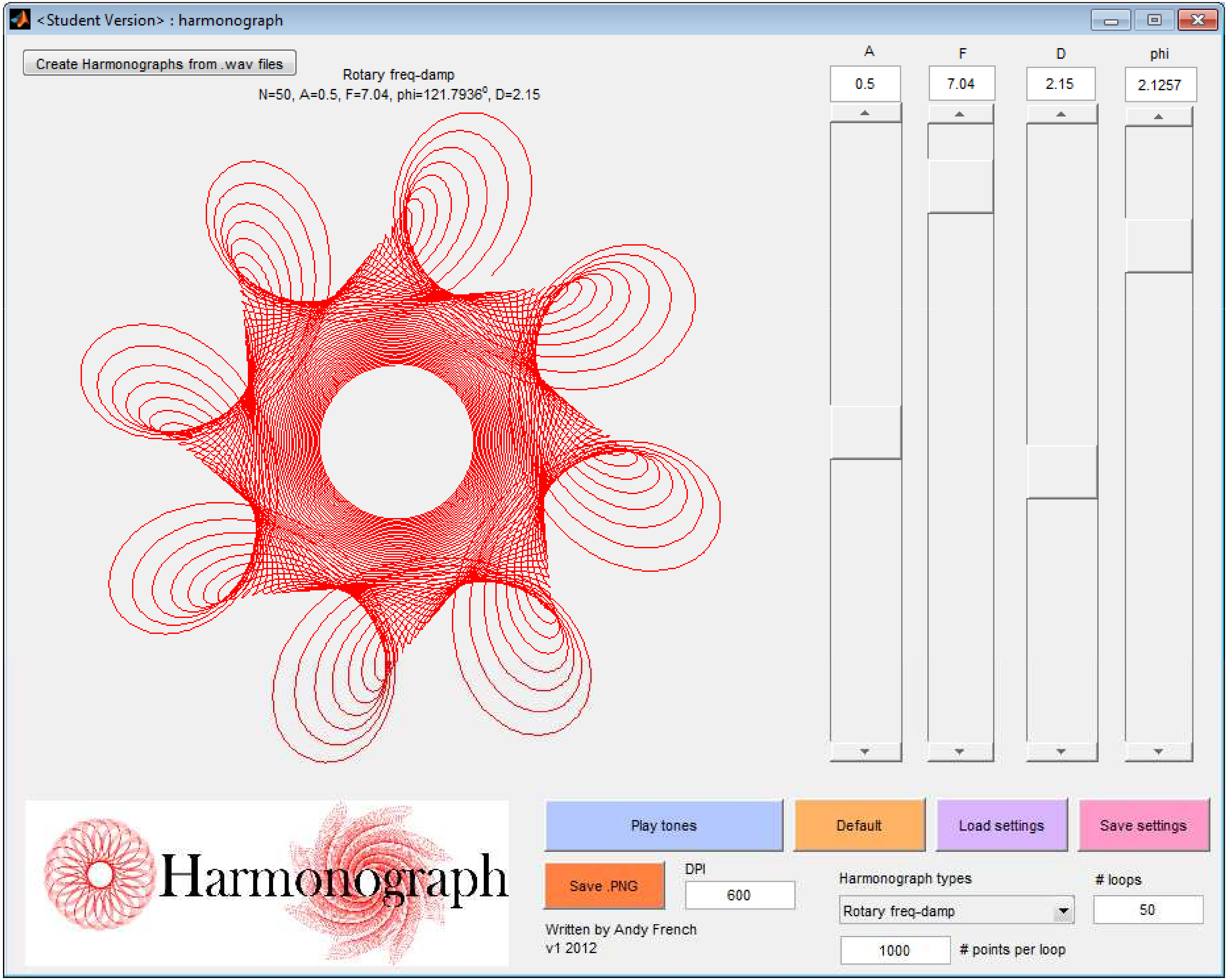
Photo from The Science Museum



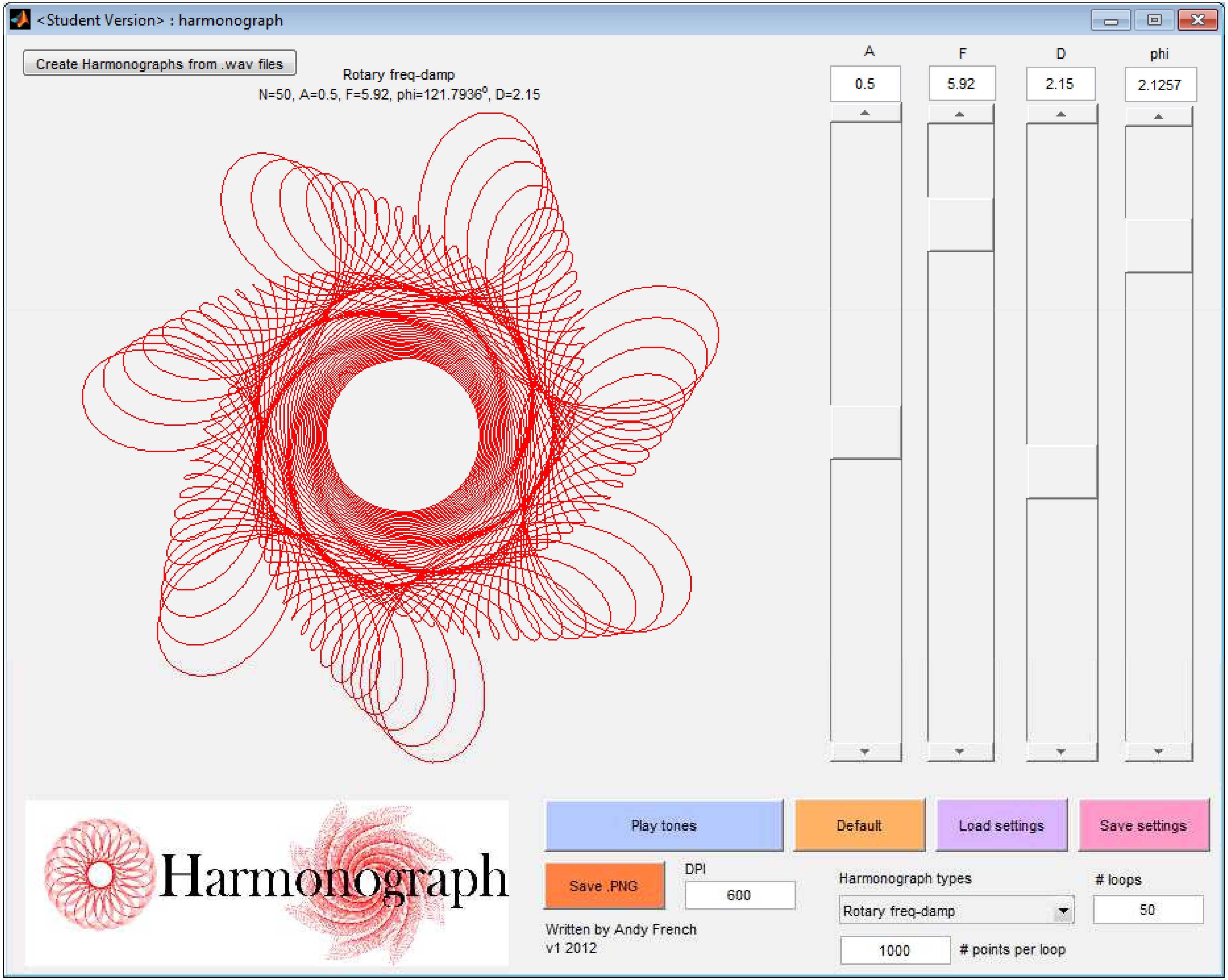
<Student Version> : harmonograph
Create Harmonographs from .wav files
Rotary freq-damp
N=50, A=0.5, F=8, phi=121.7936°, D=2.15
A
0.5
F
8
D
2.15
phi
2.1257
Harmonograph
Play tones
Default
Load settings
Save settings
Save .PNG
DPI
600
Harmonograph types
Rotary freq-damp
loops
50
Written by Andy French
v1 2012
1000
points per loop

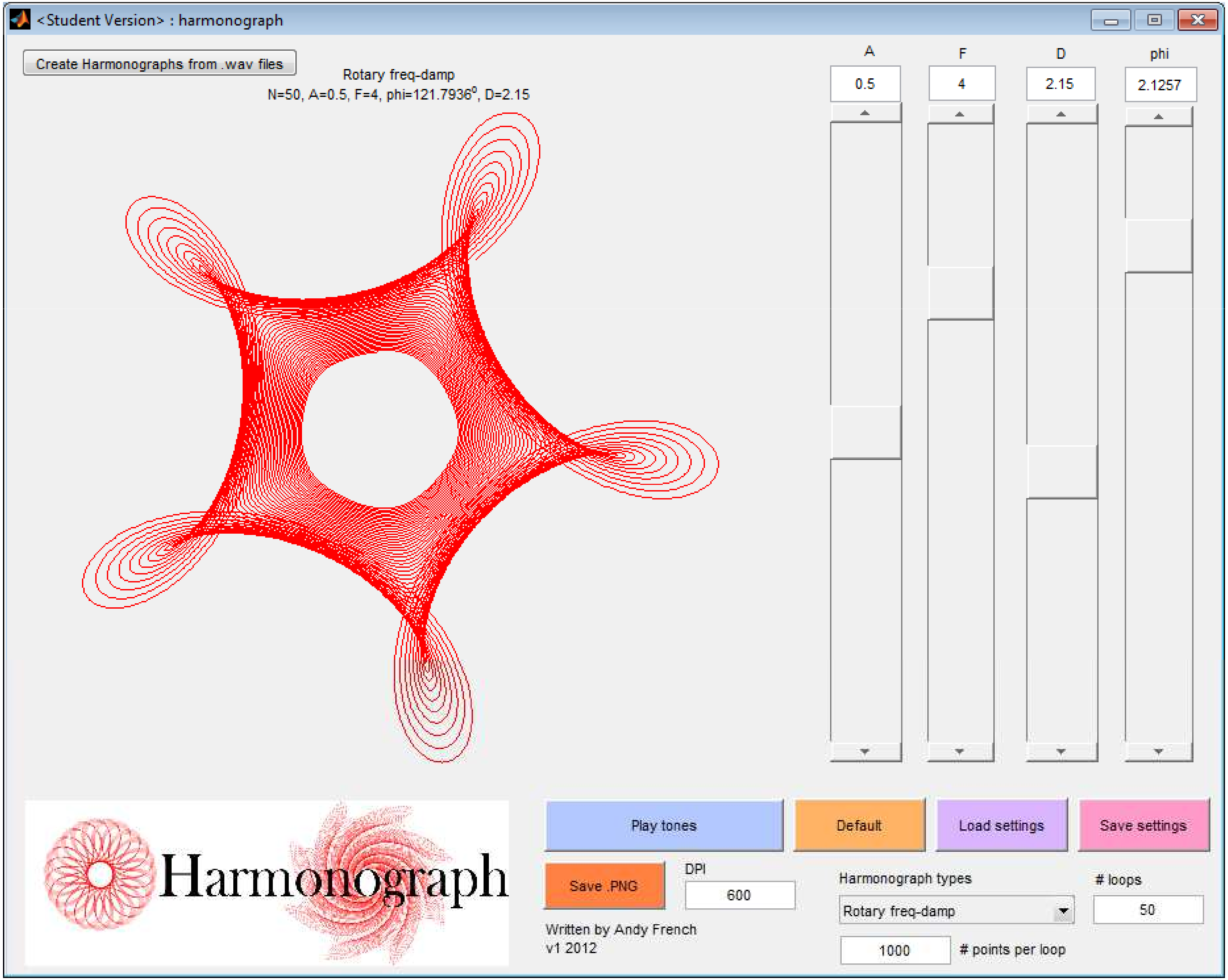


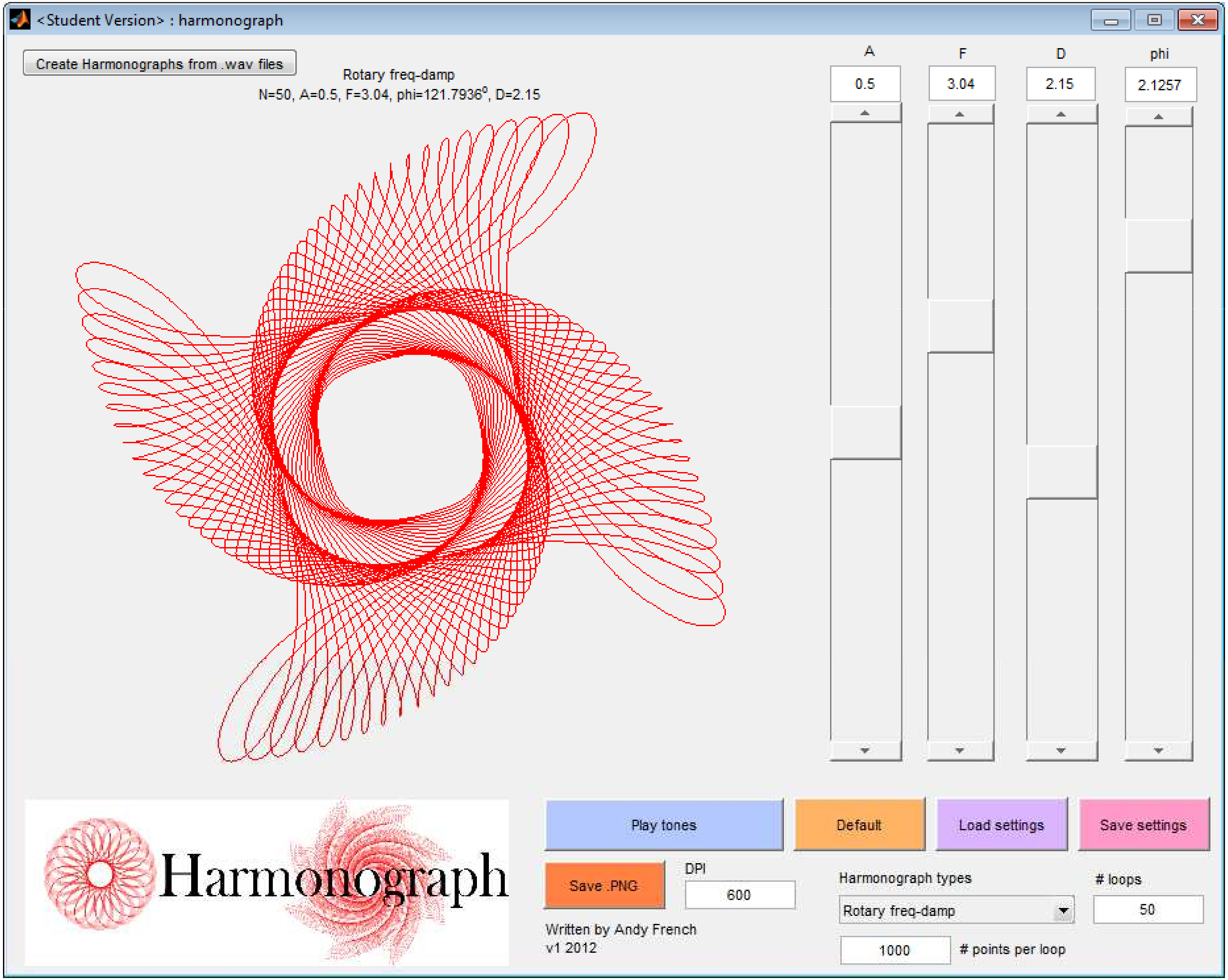
<Student Version> : harmonograph
Create Harmonographs from .wav files
Rotary freq-damp
N=50, A=0.5, F=7.84, phi=121.7936⁰, D=2.15
A
0.5
F
7.84
D
2.15
phi
2.1257
Harmonograph
Play tones
Default
Load settings
Save settings
Save .PNG
DPI
600
Harmonograph types
Rotary freq-damp
loops
50
1000
points per loop
Written by Andy French
v1 2012

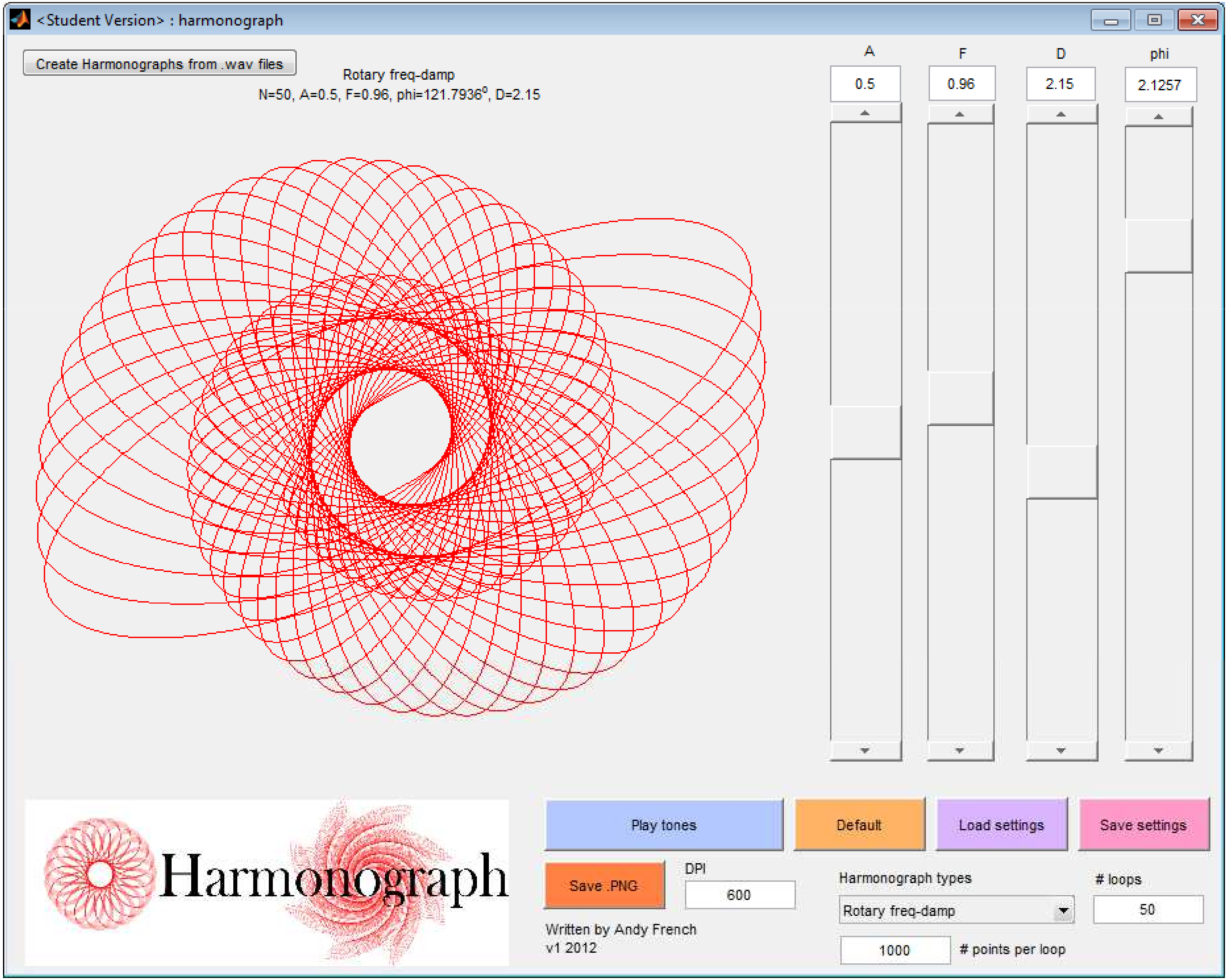


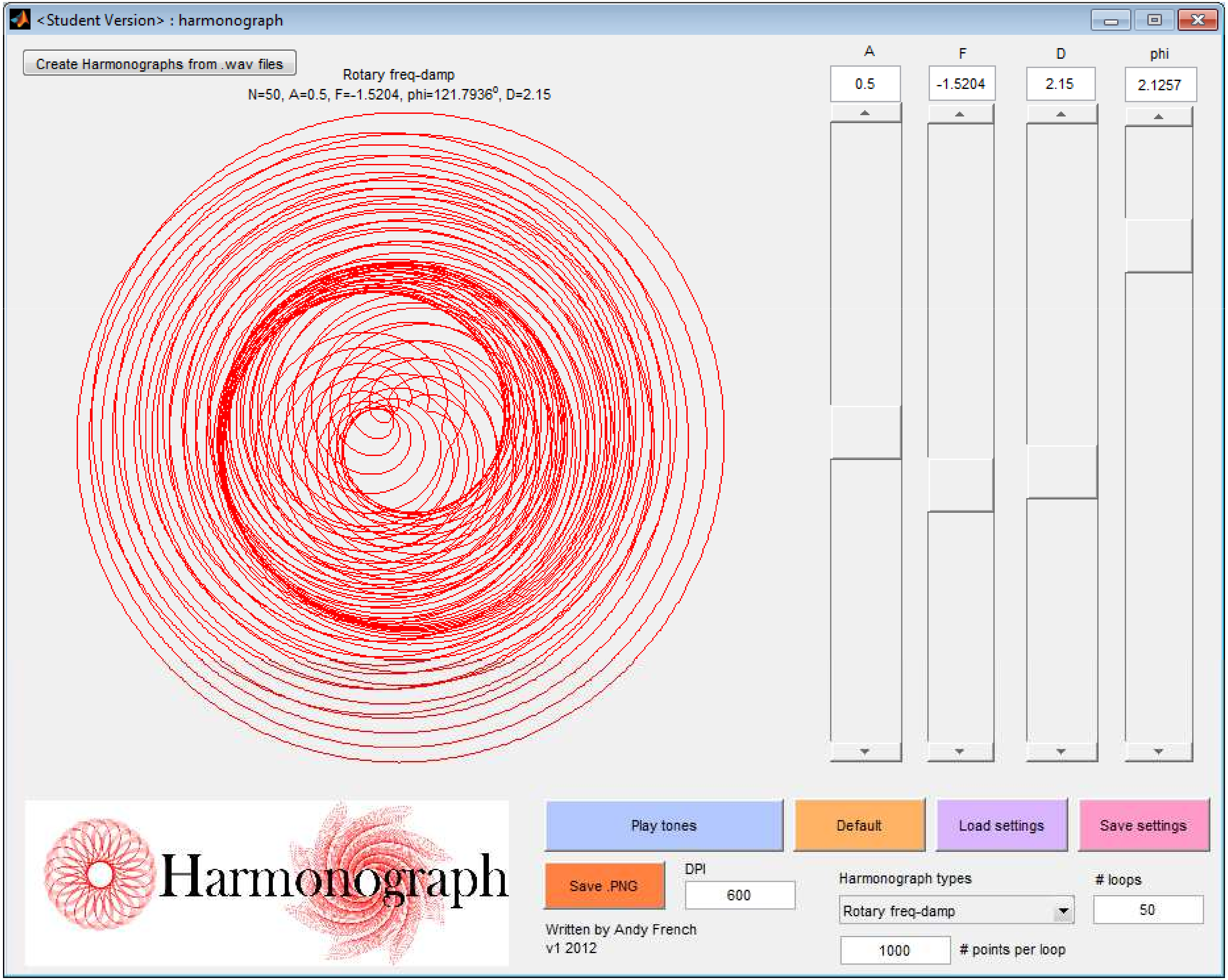
<Student Version> : harmonograph
Create Harmonographs from .wav files
Rotary freq-damp
N=50, A=0.5, F=7.04, phi=121.7936°, D=2.15
A
0.5
F
7.04
D
2.15
phi
2.1257
Harmonograph
Play tones
Default
Load settings
Save settings
Save .PNG
DPI
600
Harmonograph types
Rotary freq-damp
loops
50
Written by Andy French
v1 2012
1000
points per loop


<Student Version> : harmonograph
Create Harmonographs from .wav files
Rotary freq-damp
N=50, A=0.5, F=5.92, phi=121.7936°, D=2.15
A
0.5
F
5.92
D
2.15
phi
2.1257
Harmonograph
Play tones
Default
Load settings
Save settings
Save .PNG
DPI
600
Harmonograph types
Rotary freq-damp
loops
50
Written by Andy French
v1 2012
1000
points per loop

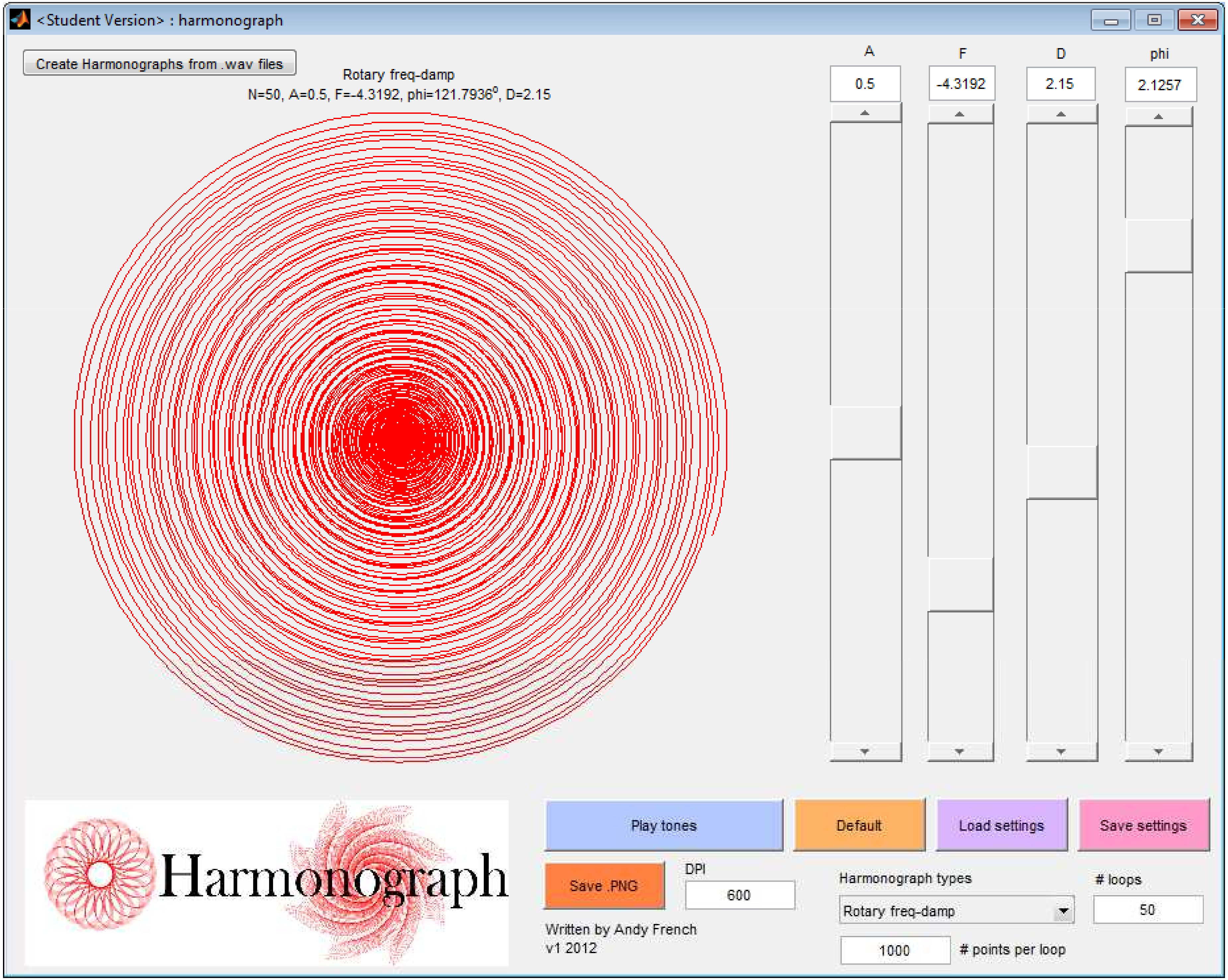

<Student Version> : harmonograph
Create Harmonographs from .wav files
Rotary freq-damp
N=50, A=0.5, F=4, phi=121.7936°, D=2.15
A
0.5
F
4
D
2.15
phi
2.1257
Harmonograph
Play tones
Default
Load settings
Save settings
Save .PNG
DPI
600
Harmonograph types
Rotary freq-damp
loops
50
Written by Andy French
v1 2012
1000
points per loop


<Student Version> : harmonograph
Create Harmonographs from .wav files
Rotary freq-damp
N=50, A=0.5, F=3.04, phi=121.7936^0, D=2.15
A
0.5
F
3.04
D
2.15
phi
2.1257
Harmonograph
Play tones
Default
Load settings
Save settings
Save .PNG
DPI
600
Harmonograph types
Rotary freq-damp
loops
50
Written by Andy French
v1 2012
1000
points per loop


<Student Version> : harmonograph
Create Harmonographs from .wav files
Rotary freq-damp
N=50, A=0.5, F=0.96, phi=121.7936°, D=2.15
A
0.5
F
0.96
D
2.15
phi
2.1257
Harmonograph
Play tones
Default
Load settings
Save settings
Save .PNG
DPI
600
Harmonograph types
Rotary freq-damp
loops
50
Written by Andy French
v1 2012
1000
points per loop



<Student Version> : harmonograph
Create Harmonographs from .wav files
Rotary freq-damp
N=50, A=0.5, F=-1.5204, phi=121.7936°, D=2.15
A
0.5
F
-1.5204
D
2.15
phi
2.1257
Harmonograph
Play tones
Default
Load settings
Save settings
Save .PNG
DPI
600
Harmonograph types
Rotary freq-damp
loops
50
Written by Andy French
v1 2012
1000
points per loop

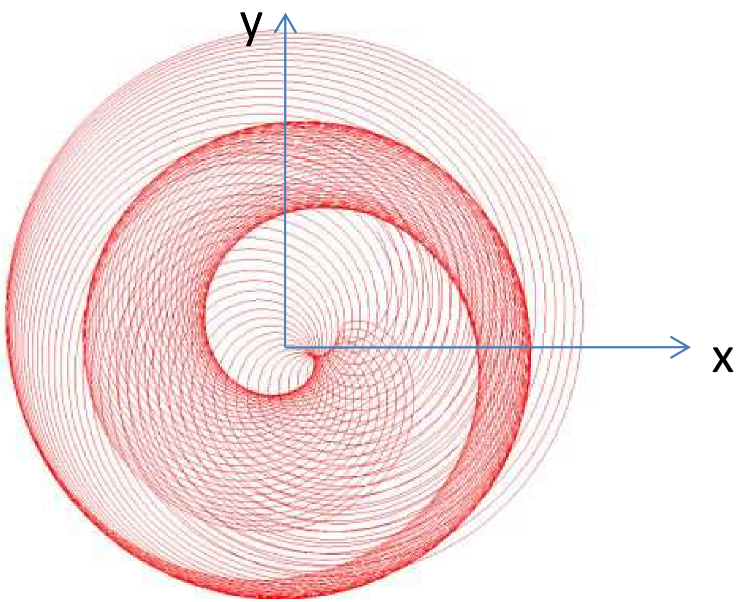


<Student Version> : harmonograph
Create Harmonographs from .wav files
Rotary freq-damp
N=50, A=0.5, F=-4.3192, phi=121.7936°, D=2.15
A
0.5
F
-4.3192
D
2.15
phi
2.1257
Harmonograph
Play tones
Default
Load settings
Save settings
Save .PNG
DPI
600
Harmonograph types
Rotary freq-damp
loops
50
Written by Andy French
v1 2012
1000
points per loop

$$x = A_1 e^{-\frac{t}{T_1}} \sin(tW_1 + P_1) + A_2 e^{-\frac{t}{T_2}} \sin(tW_2 + P_2)$$

$$y = A_3 e^{-\frac{t}{T_3}} \sin(tW_3 + P_3) + A_4 e^{-\frac{t}{T_4}} \sin(tW_4 + P_4)$$

Rotary harmonograph with frequency damping



$$T = \frac{2\pi}{\omega \log\left(\frac{100}{100-D}\right)} \left[1, \frac{1}{F}, 1, \frac{1}{F}\right]$$

$$A = [1, a, 1, a]$$

$$W = [\omega, -F\omega, \omega, -F\omega]$$

$$P = \left[0, \phi, \frac{\pi}{2}, \frac{\pi}{2} + \phi\right]$$

Parameters

t is time /seconds

ω is 2π times the first pendulum swing frequency /Hz

a is the amplitude ratio

F is the frequency ratio

D is the damping factor (typically between 0 and 5)

ϕ is the phase difference /radians between the pendula

Musical harmony

- The mathematics of music has been known since the time of Pythagoras, 2500 years ago
- Frequency intervals of simple fractions e.g. 3:2 (a fifth) yield ‘harmonious’ music
- An **octave** means a **frequency ratio of 2**. An octave above concert A (440Hz) is therefore 880Hz. An octave below is 220Hz.
- The modern ‘equal-tempered scale’ divides an octave (the frequency ratio 2) into twelve parts such that

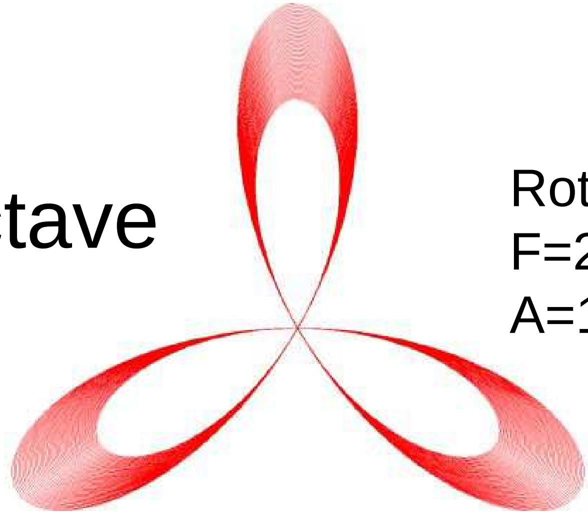
$$F_n = 2^{n/12} = \sqrt[\frac{n}{12}]{2}$$

Musical harmony

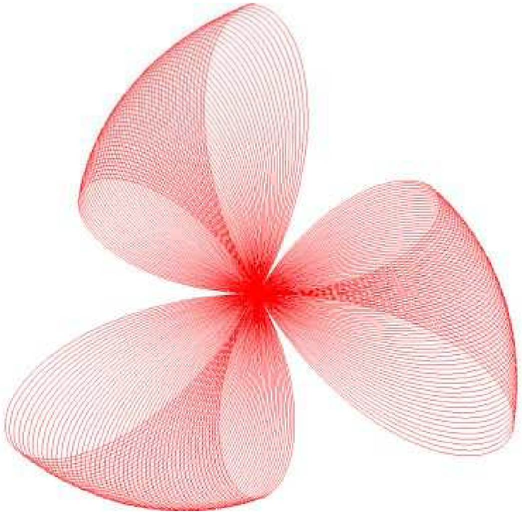
Name	Exact value in 12-TET	Decimal value in 12-TET	Cents	Just intonation interval
Unison (C)	$2^{0/12} = 1$	1.000000	0	$\frac{1}{1}$ = 1.000000
Minor second (C♯/D♭)	$2^{1/12} = \sqrt[12]{2}$	1.059463	100	$\frac{16}{15}$ = 1.066667
Major second (D)	$2^{2/12} = \sqrt[6]{2}$	1.122462	200	$\frac{9}{8}$ = 1.125000
Minor third (D♯/E♭)	$2^{3/12} = \sqrt[4]{2}$	1.189207	300	$\frac{6}{5}$ = 1.200000
Major third (E)	$2^{4/12} = \sqrt[3]{2}$	1.259921	400	$\frac{5}{4}$ = 1.250000
Perfect fourth (F)	$2^{5/12} = \sqrt[12]{32}$	1.334840	500	$\frac{4}{3}$ = 1.333333
Augmented fourth (F♯/G♭)	$2^{6/12} = \sqrt{2}$	1.414214	600	$\frac{7}{5}$ = 1.400000
Perfect fifth (G)	$2^{7/12} = \sqrt[12]{128}$	1.498307	700	$\frac{3}{2}$ = 1.500000
Minor sixth (G♯/A♭)	$2^{8/12} = \sqrt[3]{4}$	1.587401	800	$\frac{8}{5}$ = 1.600000
Major sixth (A)	$2^{9/12} = \sqrt[4]{8}$	1.681793	900	$\frac{5}{3}$ = 1.666667
Minor seventh (A♯/B♭)	$2^{10/12} = \sqrt[6]{32}$	1.781797	1000	$\frac{7}{4}$ = 1.750000
Major seventh (B)	$2^{11/12} = \sqrt[12]{2048}$	1.887749	1100	$\frac{15}{8}$ = 1.875000
Octave (C)	$2^{12/12} = 2$	2.000000	1200	$\frac{2}{1}$ = 2.000000

Represent musical harmonies visually with the harmonograph!

octave



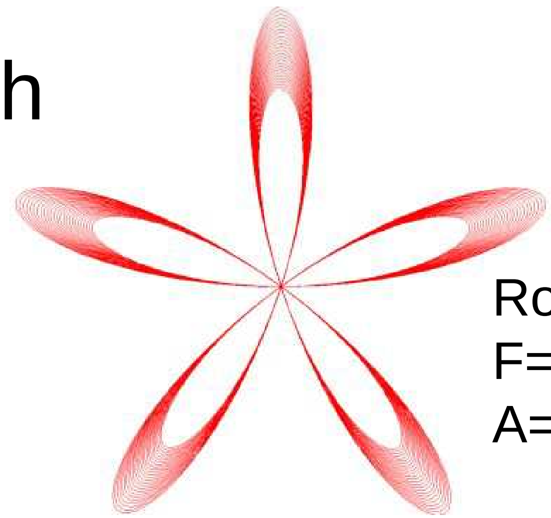
Rotary
F=2, D=0.7,
A=1, phi=0



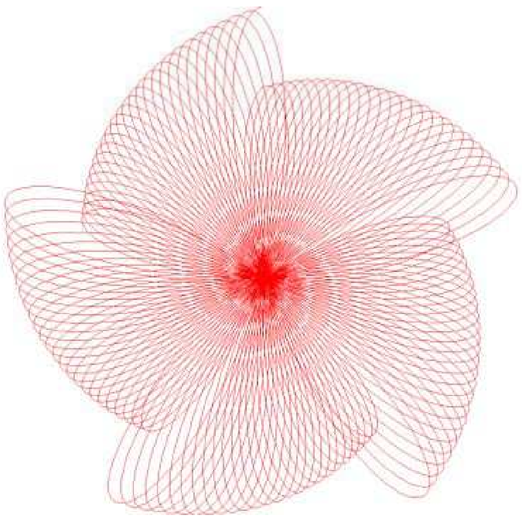
Rotary
F=2.01, D=0.7,
A=1, phi=0

Note the difference a small change in F makes....

fifth

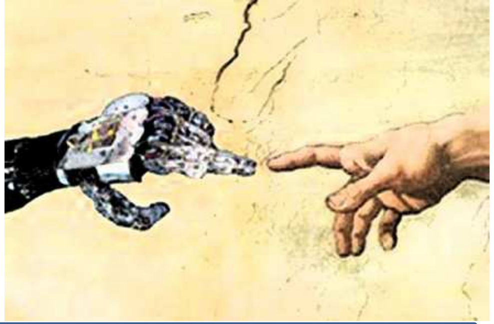


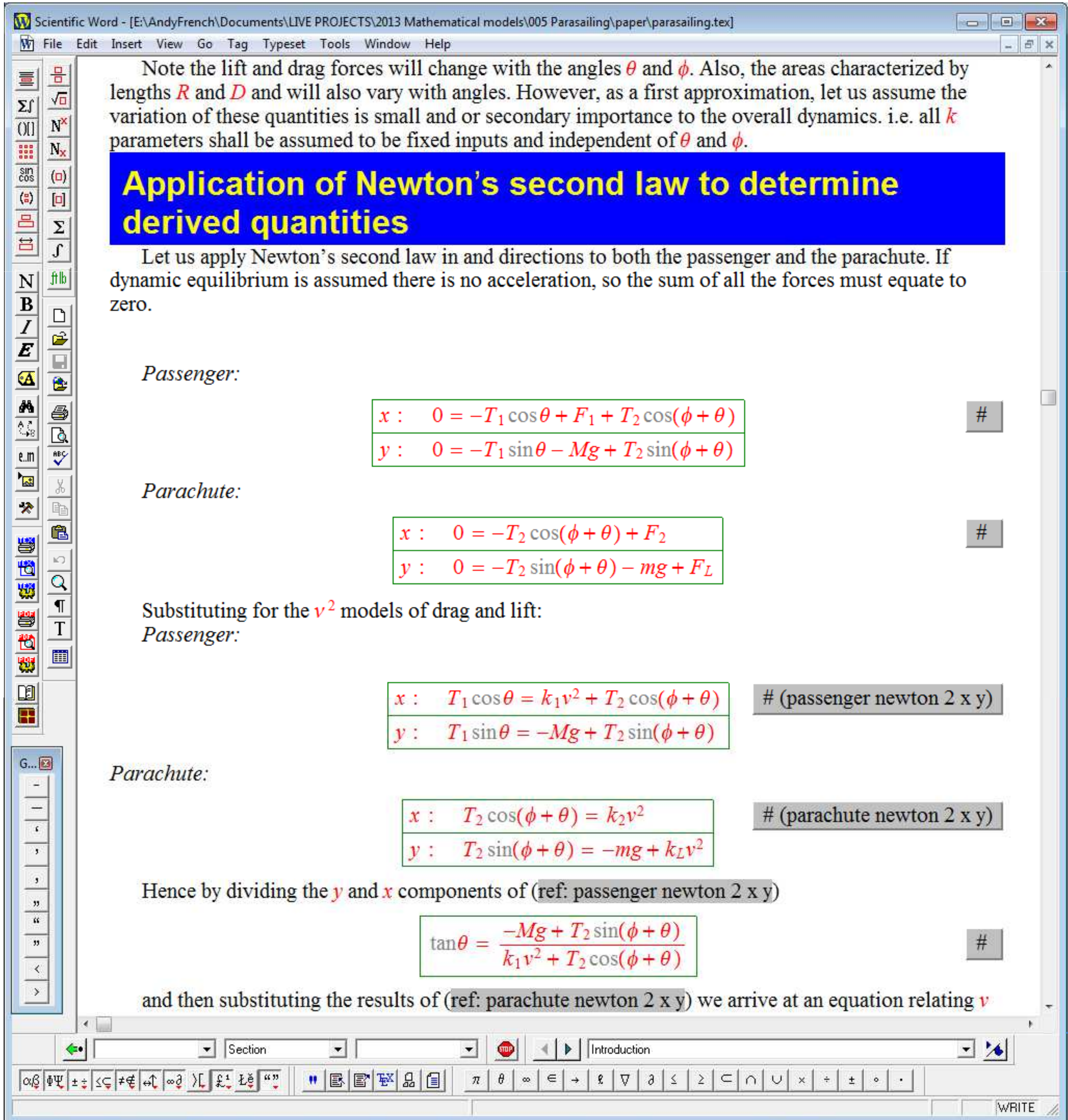
Rotary
F=3/2, D=0.7,
A=1, phi=0



Rotary
F=1.51, D=0.7,
A=1, phi=0

What You See Is What You Need





Note the lift and drag forces will change with the angles θ and ϕ. Also, the areas characterized by lengths R and D and will also vary with angles. However, as a first approximation, let us assume the variation of these quantities is small and or secondary importance to the overall dynamics. i.e. all k parameters shall be assumed to be fixed inputs and independent of θ and ϕ.

Application of Newton's second law to determine derived quantities

Let us apply Newton's second law in and directions to both the passenger and the parachute. If dynamic equilibrium is assumed there is no acceleration, so the sum of all the forces must equate to zero.

Passenger:

$$x: \quad 0 = -T_1 \cos\theta + F_1 + T_2 \cos(\phi + \theta)$$
$$y: \quad 0 = -T_1 \sin\theta - Mg + T_2 \sin(\phi + \theta)$$
#

Parachute:

$$x: \quad 0 = -T_2 \cos(\phi + \theta) + F_2$$
$$y: \quad 0 = -T_2 \sin(\phi + \theta) - mg + F_L$$
#

Substituting for the v^2 models of drag and lift:
Passenger:

$$x: \quad T_1 \cos\theta = k_1 v^2 + T_2 \cos(\phi + \theta)$$
$$y: \quad T_1 \sin\theta = -Mg + T_2 \sin(\phi + \theta)$$
(passenger newton 2 x y)

Parachute:

$$x: \quad T_2 \cos(\phi + \theta) = k_2 v^2$$
$$y: \quad T_2 \sin(\phi + \theta) = -mg + k_L v^2$$
(parachute newton 2 x y)

Hence by dividing the y and x components of (ref: passenger newton 2 x y)

$$\tan\theta = \frac{-Mg + T_2 \sin(\phi + \theta)}{k_1 v^2 + T_2 \cos(\phi + \theta)}$$
#

and then substituting the results of (ref: parachute newton 2 x y) we arrive at an equation relating v

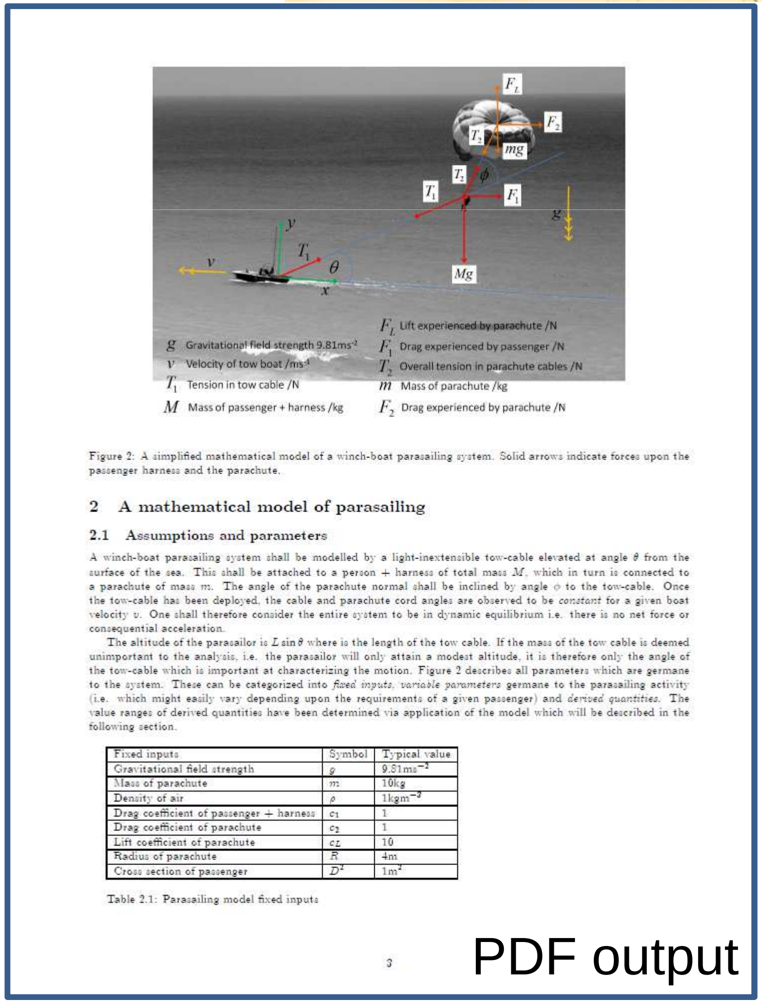


Figure 2: A simplified mathematical model of a winch-boat parasailing system. Solid arrows indicate forces upon the passenger harness and the parachute.

2 A mathematical model of parasailing

2.1 Assumptions and parameters

A winch-boat parasailing system shall be modelled by a light-inextensible tow-cable elevated at angle θ from the surface of the sea. This shall be attached to a person + harness of total mass M, which in turn is connected to a parachute of mass m. The angle of the parachute normal shall be inclined by angle ϕ to the tow-cable. Once the tow-cable has been deployed, the cable and parachute cord angles are observed to be *constant* for a given boat velocity v. One shall therefore consider the entire system to be in dynamic equilibrium i.e. there is no net force or consequential acceleration.

The altitude of the parasailor is $L \sin\theta$ where is the length of the tow cable. If the mass of the tow cable is deemed unimportant to the analysis, i.e. the parasailor will only attain a modest altitude, it is therefore only the angle of the tow-cable which is important at characterizing the motion. Figure 2 describes all parameters which are germane to the system. These can be categorized into *fixed inputs*, *variable parameters* germane to the parasailing activity (i.e. which might easily vary depending upon the requirements of a given passenger) and *derived quantities*. The value ranges of derived quantities have been determined via application of the model which will be described in the following section.

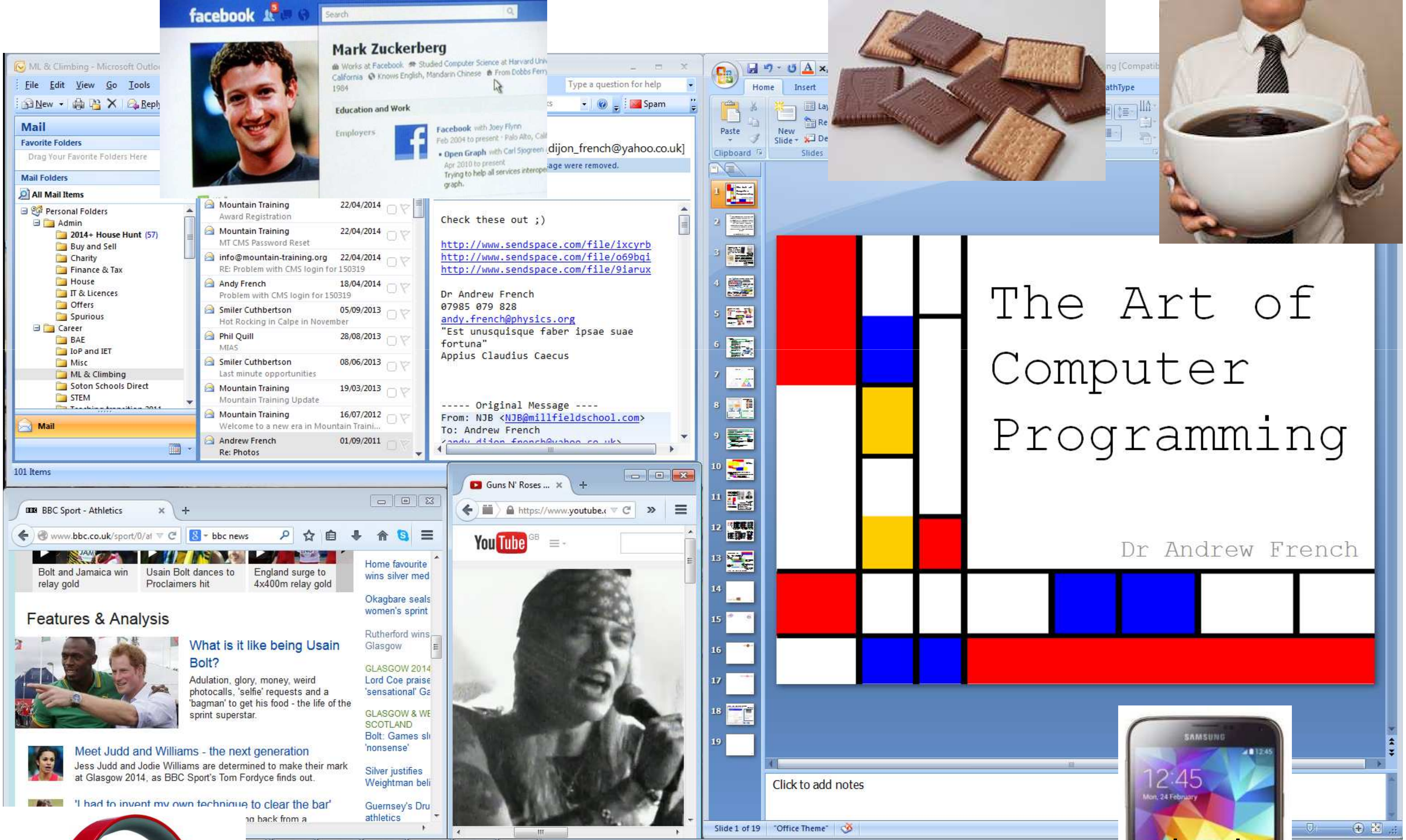
Fixed inputs	Symbol	Typical value
Gravitational field strength	g	9.81ms^{-2}
Mass of parachute	m	10kg
Density of air	ρ	1kgm^{-3}
Drag coefficient of passenger + harness	c_1	1
Drag coefficient of parachute	c_2	1
Lift coefficient of parachute	c_L	10
Radius of parachute	R	4m
Cross section of passenger	D^2	1m^2

Table 2.1: Parasailing model fixed inputs

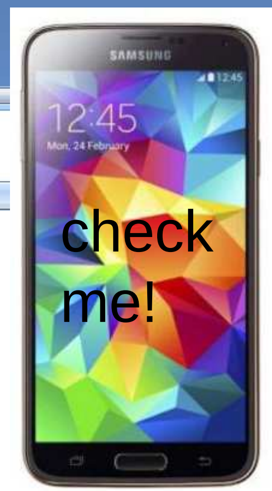
3

PDF output

LaTeX



‘High productivity multi-tasking,’
or are you just being distracted?



What You See Is What You Need

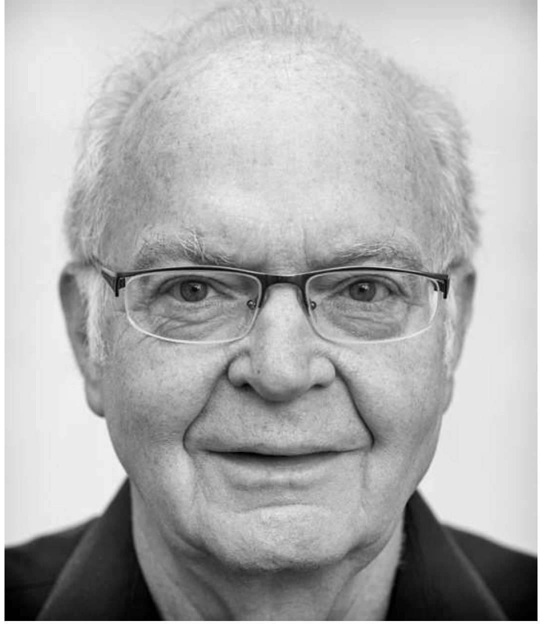
The rise of *Apps* for Smartphones....Typically software designed for a *very specific* purpose

Google play

Available on the App Store

→ ... and a market for *your* designs

Some final wisdom from Donald Knuth



"The psychological profiling of a programmer is mostly the ability to shift levels of abstraction, from low level to high level. To see something in the small and to see something in the large."

"Email is a wonderful thing for people whose role in life is to be on top of things. But not for me; my role is to be on the bottom of things. What I do takes long hours of studying and uninterruptible concentration."

Fast numerical calculation + display systems

Use of computer programming as an artistic tool

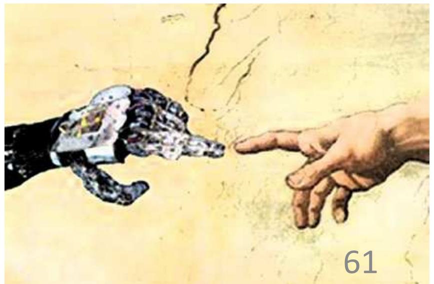
`functions`
`interfaces`

Where you can *design and refine* the tool

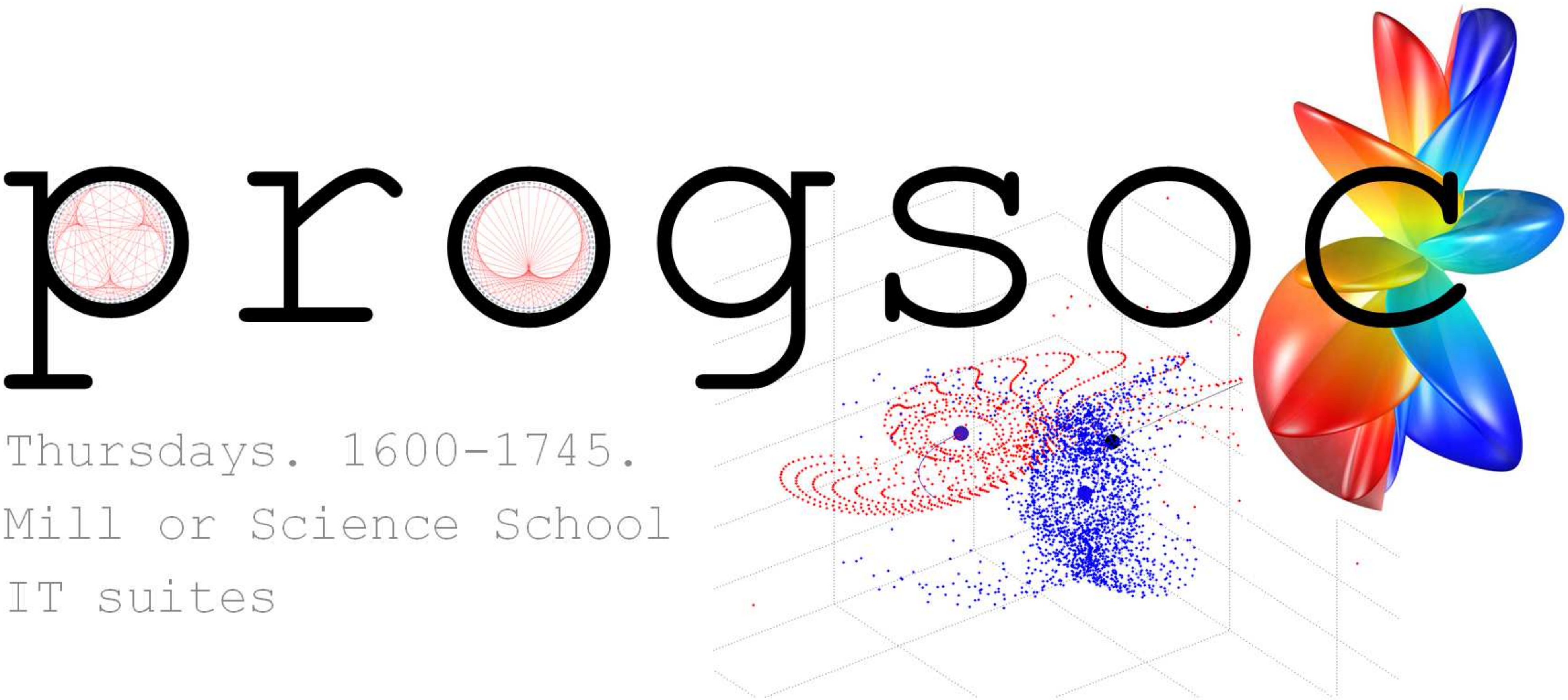
Complexity, and beauty, from simplicity (i.e. code)

Case studies

General thoughts on how *humans* best interact with information technology
What You See Is What You Need!



Inspired?



All welcome, regardless of prior experience