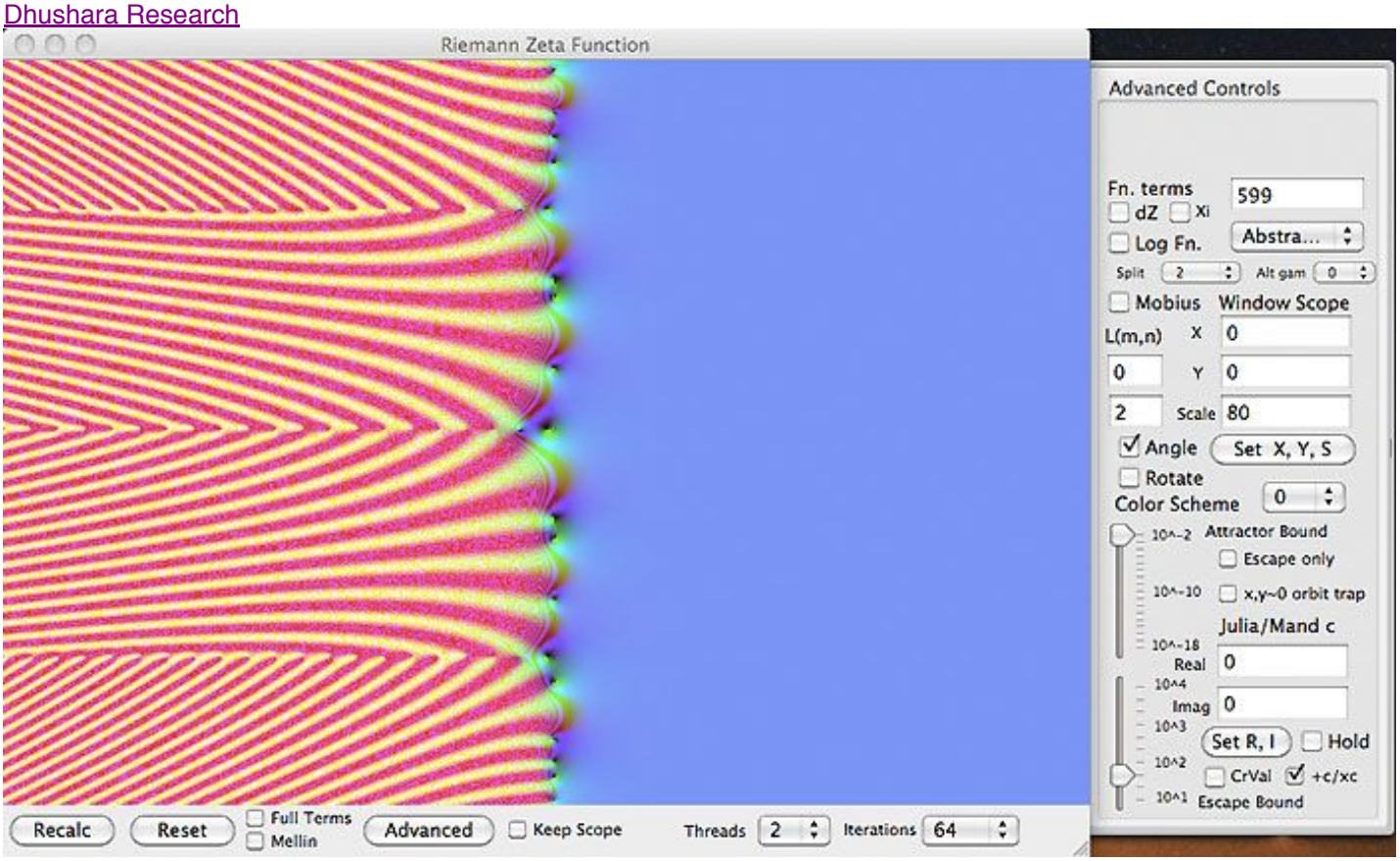
Dhushara Research



RZViewer showing a third degree transcendental *L*-function using the functional equation method in RZViewer

Riemann Zeta Function Viewer 2.0 for Mac Flight Manual

The viewer allows you to interactively explore the complex Riemann zeta function and its complex chaotic dynamics:

$$\zeta(z) = \sum_{n=1}^{\infty} \frac{1}{n^z} = \prod_{p\ \text{prime}} \frac{1}{\left(1 - p^{-z}\right)}$$

Current Issue 2.0.0 Mac OS Tiger to el Capitan

Dark Heart Package 2.0 for Mac Latest full overview of DH and RZ Viewer 2.0

Download all viewers with Riemann Zeta and Dark Heart viewer source codes

Uses Mellin transform integrals in Dirichlet, Hecke, elliptic curve and modular form L-functions to give a smooth representation of a wide variety of L-functions permitting investigation such as confirming the Birch Swinnerton-Dyer conjecture. Includes an experimental general Dokchitser type inverse Mellin transform algorithm for motivic *L*-functions, operating successfully on proof-of-concept test examples.

.

In addition to its scintillating Mandelbrot and Julia exploration facilities, version 1.7 supports a full spectrum of abstract zeta and *L*-functions, including those of elliptic curves and modular forms. It has direct onboard parametrization of Riemann and Hurwitz zeta functions, Dirichlet *L*-functions, Dedekind zeta, and Hecke *L*-functions as well as Davenport-Heilbronn counterexamples. It also contains a full suite of examples of *L*-functions of elliptic curves, modular forms and Maass forms, including the most recent third degree transcendental *L*-functions so far discovered.

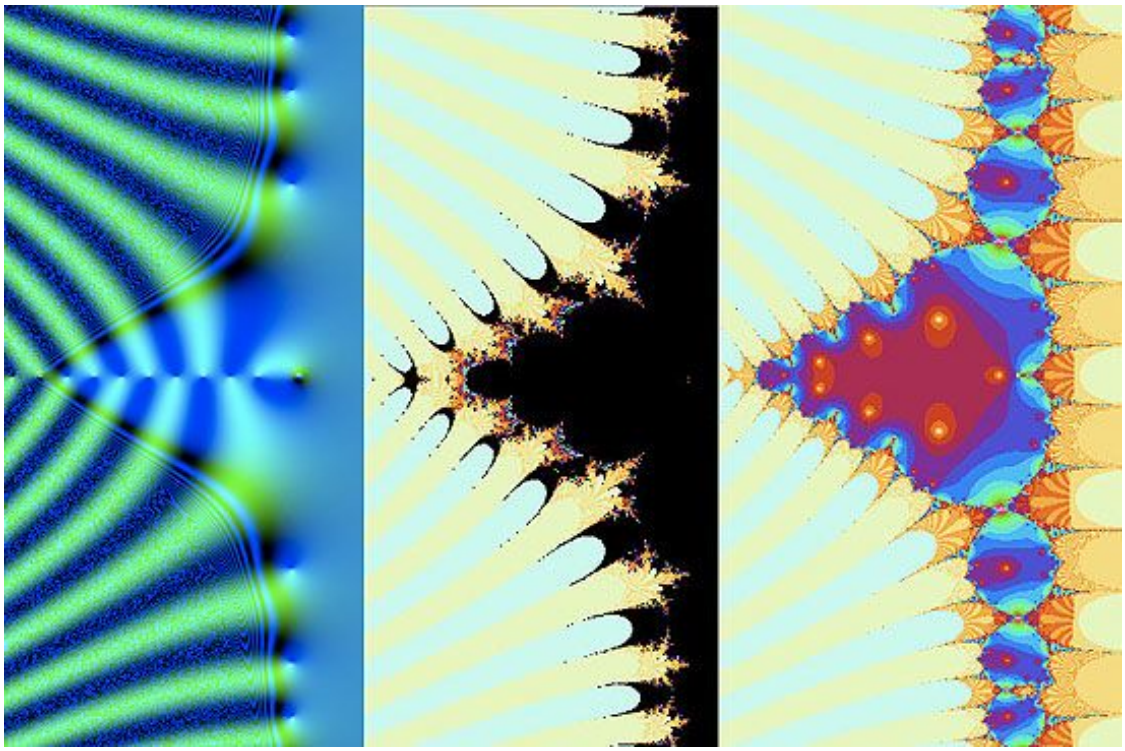
Includes template upload *C*-scripts running directly off Mac Terminal generating parameter files for RZViewer using coefficients generated from open source cross-platform Sage and PARI-GP scripts for the *L*-functions of elliptic curves and modular forms. The package also includes Tim Dokchitser's Computel algorithm and PARI-GP to generate further coefficient templates and bitmaps to run under RZViewer.

See http://dhushara.com/DarkHeart/RZV/ for research, updates and source code.

Research investigations using **RZViewer** its associated **Matlab toolbox** and its companion Dark Heart Viewer:

- **A Dynamical Key to the Riemann Hypothesis** (with movies)
- **Fractal Geography of the Riemann Zeta Function** (with movies)
- **Experimental Observations on the Riemann Hypothesis**
- **Exploding the Dark Heart of Chaos**

By clicking, you can also investigate the discrete dynamical parameter plane (Mandelbrot set) of the critical value 1 as real(z)>>+?, and the Julia sets of $z=f(z)+c$ (or $z=cf(z)$) for $(x,y)=c$.



Zeta, its additive Mandelbrot set and Julia set of c=0.

You can also investigate related functions eta, xi, gamma, and psi (digamma), the derivatives of zeta, xi and gamma and Newton's method on these as well (see below).

Basics:

Drag a rectangle to enlarge a portion of the current image.

Click a point to cycle:
Function > Mandelbrot > Julia(c)
at the clicked point c on the parameter plane (Mandelbrot set).

The controls all have ***floating help panes*** to guide you.

The window can be resized to suit in real time, resulting in a refresh and recalculation.

Standard settings:

Threads enables a multi-core machine to work faster using as many threads as coprocessors. ***Max Iterations*** gives how many iteration steps of a point on the Julia or Mandelbrot set before maximum iteration cutoff. ***Recalc*** refreshes and recalculates the existing screen. ***Reset*** returns to the RZ function and default scale but doesn't alter any other settings. ***Keep scope on click*** keeps the window centre and scale when clicking between Mandelbrot, Julia and function, so the other parameters can be changed without

changing the point of view.

To facilitate locating critical points for calculating Mandelbrot sets, tick ***dZ*** and explore the derivative function for a critical point (a zero of the derivative looking like a dimple). Use dragging to blow up to high resolution. Tick ***Keep scope*** and slowly click twice right on the zero without moving the mouse. The *c*-values will now appear in the Julia portrait. Then tick ***Hold***, untick ***dZ*** and click twice through the original function to the Mandelbrot set. The *c*-values are retained in the text field by ***Hold***. Now untick ***Hold*** and click the ***Set R,I*** button and you will get the Mandelbrot sets of the given *c*-value.

To facilitate fast fractal iteration, the default is 100 function series terms and a cutoff of 64 iterations for each point of the Mandelbrot and Julia sets, but these can be adjusted for greater accuracy.

Video of Julia set Explosions as we move *c* from ~ -19 to ~ -15. Notice the independent variations of the dynamics in the central bay and the right half-plane.

Menus:

About gives a complete instruction summary. ***Save*** saves the current window's image as a tif file and all the settings for a given calculation in a text file (see format below), enabling you to save all the parameters for future investigation and redrawing. It will overwrite any files of the same name. ***Load Parameters*** will load a previous calculation and redraw it. ***Load Movie Seq*** loads a sequence text file to generate a series of images to make a movie (see format below). ***Save Movie*** saves the sequence as a series of LZW compressed tifs. Images in the main window can also be captured and saved directly to png format using shift apple 4. ***Print*** enables an image to be printed or exported to pdf format. ***Page Setup*** needs to be landscape in 80% to fit the standard window on one A4 page. ***Help*** directs you to the ***About*** menu.

Video of fireworks in the zeros from $L(5,4)$, rotating each character by the factor implied by their position on the unit circle.

Advanced settings:

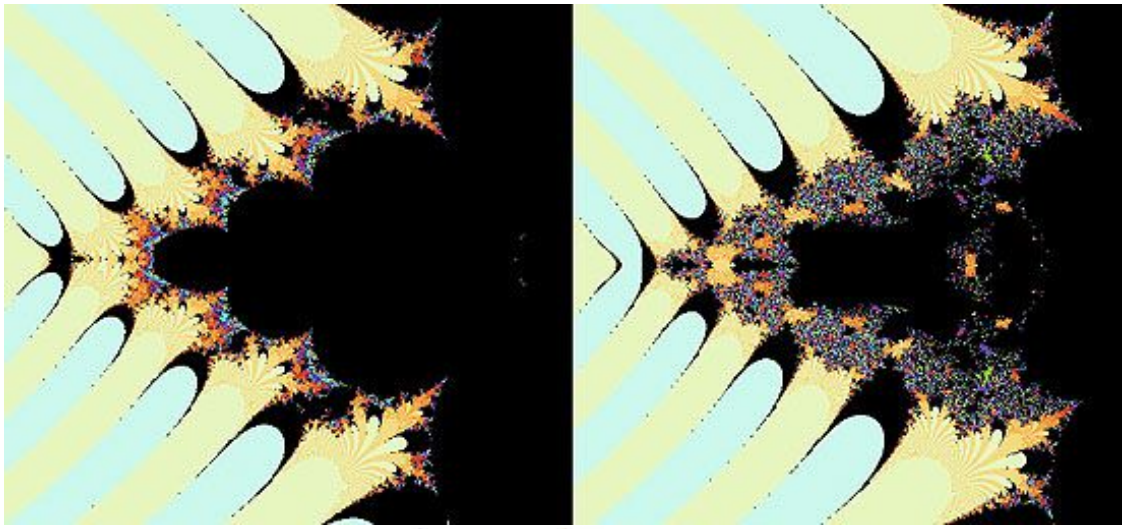
Pop-up Menus: ***Color schemes*** can be changed in real time using the color pop up menu. All the other controls require window recalculation.

Functions related to zeta, including eta, xi, gamma, psi and others discussed below can also be explored using the function pop-up menu. For completeness, an ***alternative Gamma*** product formula, can be used instead of the default Lancos approximation, by pulling down the pop-up menu to choose a power of 10 terms in the product, but is only slowly convergent, so is perilously slower and less accurate unless 10,000 iterations are used.

Buttons and Text Fields: A ***function terms*** text field is provided to set how many terms are calculated in the chosen sum or product function, with a default of 100 for speed of initial exploration. This can be adjusted to exact large values which have anomalous behavior (see end). Current ***window parameters*** and ***c values*** can be read out from and written into the text fields and set by pressing the appropriate button. ***L(m,n)*** text fields enable examining a wide variety of Dirichlet L-functions as well as Zeta and Eta. Numbers in the real text fields are double precision and may appear in floating format e.g. -7.000000186963007e-05 manifestly too large to see in their entirety in the text box without dragging or selecting all and copying. The Julia c values will be overwritten by the Mandelbrot critical point origin and current Julia click point each cycle but can be manually re-entered as desired.

Tick boxes: ***dZ*** portrays the derivative in function mode so one can find critical points for portraying Mandelbrot sets. ***Full Terms*** is located on the main page as a control and as an attribute. It forces the current function to use all the specified series terms. Otherwise it breaks off if the terms get smaller than 0.001 the size of the sum. Occasionally not having Full causes strip dislocations in a few regions. The control has immediate effect during drawing and is useful if a slow blow up of a Julia set needs to have full iterations just for the complex boundaries of the kernel. ***Mellin*** uses a Mellin transform in the central region if the function admits an analytic continuation. Like Full Terms, it is active during drawing. The ***logFn.*** tick box examines the log of the selected function. ***Split*** sets the functional equation split (-1> -100, 0>0, 1>0.25, 2>0.5). There is an x,y~0 orbit trap option highlighting points on the orbit which iterate close to the axes. ***Mobius*** transforms the complex plane, mapping the line x=1/2 onto the unit circle, so you can see the whole extent of the Dirichlet functions. For the exact mapping see below. ***Escape only*** tests only for escaping points, to avoid spurious periodic solutions from overwriting escaping orbits, but is slower because attracting orbits have to run to the Max interations for each point. ***crVal*** uses critical value instead of critical point for Mandelbrots. This is useful for generating parametric movies running between critical values. For finding critical values see blue skiews method. Tick box ***+c/xc*** changes the parametrization

from f(z)+c to cf(z) giving a second parameter plane and collection of Julia sets for each function. ***Angle*** turns on and off the green/yellow display of the angle (argument) of f(z) in function view for each mode. ***Rotate*** rotates the image by pi/2 to assist viewing the critical line.

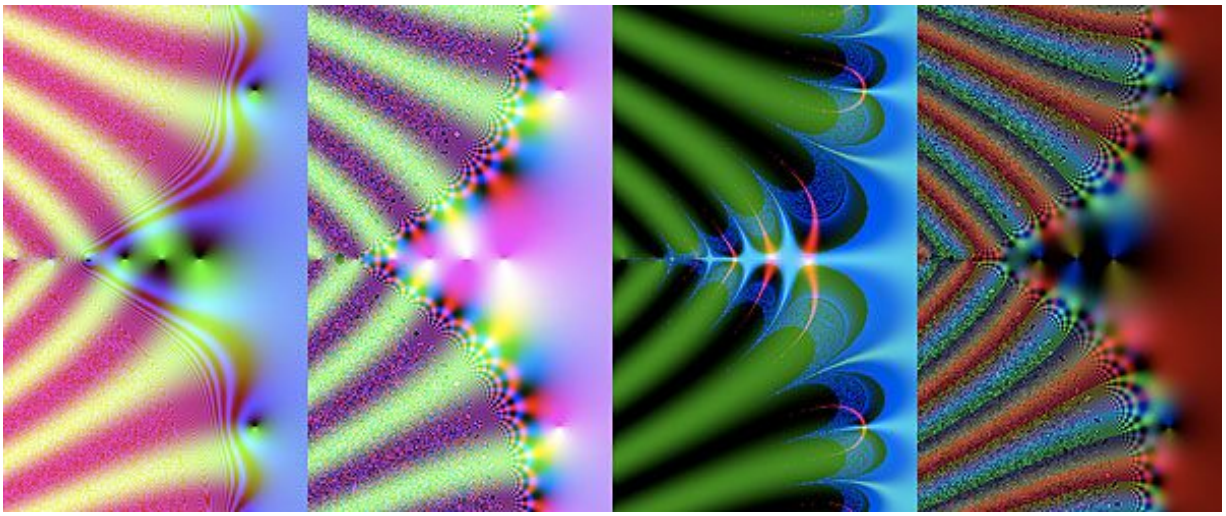


Parameter planes of $\zeta(z)+c$ and $c\zeta(z)$ showing extensive differences

Sliders: ***Attractor bound*** adjusts the size of the epsilon neighbourhood testing for fixed point or periodic attractors. ***Escape bound*** sets the real absolute bounds on points escaping to infinity.

There are several **colour schemes** for function, Mandelbrot and Julia:

Function: (0) ***Absolute-Angle mode*** rgb = red logarithmic abs(*z*), blue cosine abs(*z*), green angle(*z*). This is the most informative although not the most appealing (1) ***Real-Imaginary mode*** rgb = real, imaginary and angle, (2) ***X-ray mode*** with red, cyan real~0 and imag~0, by exp(-*x*^2), adjustable by the attractor bound, overlaying bg = abs(*z*) and angle(*z*) Highlights gram points with real=0 on *x*=1/2 and zeros at the intersections of real~0 and imag~0. (3) ***Positive-Negative mode***. Highlights the positive and negative real (r~c), imaginary (g~m) and angle by 1-cos (b~y). (4) ***Log-angle mode*** This method uses abs(log(1/(1-Z(z))) for the absolute value, adjustable using the escape bound slider, retaining the angle of Z(z). This still highlights the zeros since log(1/(1-0))=log(1)=0 but highlights variations in the function to the right of the critical line. It is good at highlighting the function in the right half-plane and in the critical strip in a way which evenly compares asymptotical large and near-unitary values. (5) Soft mode rgb = cos(angle) sin(angle), abs(z). Good for retrieving function data from tif file.



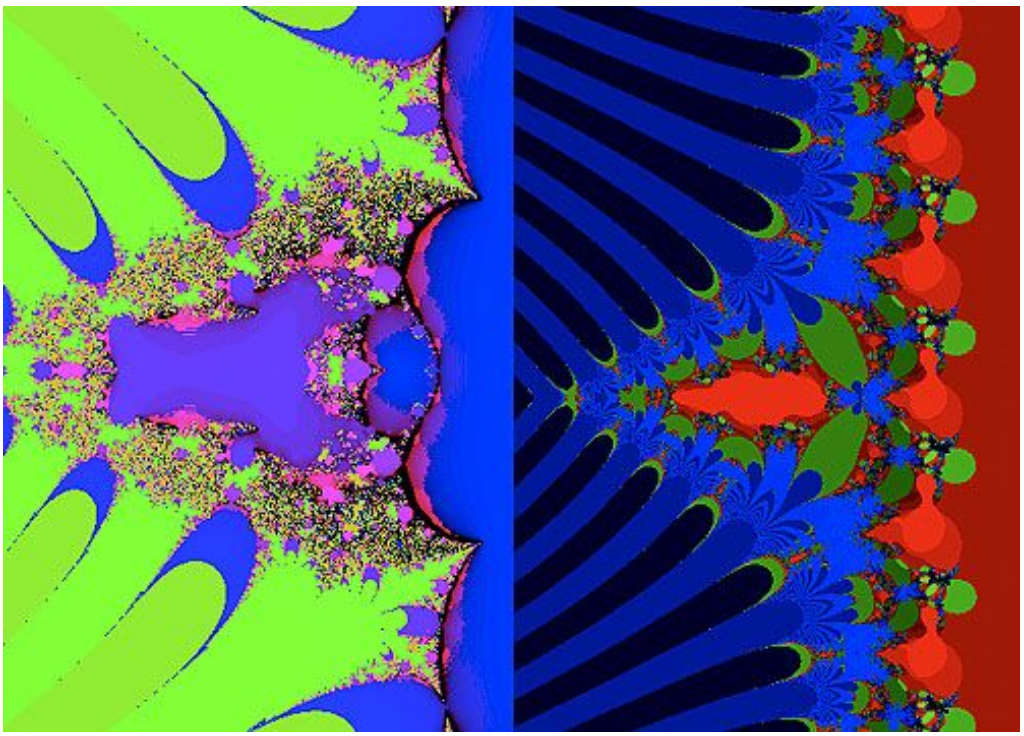
Function modes 0-3

Mandelbrot: (0) Sine wave colours, (1) attactor coded colours, (2) RGB ranked colours (3) Potential function rainbow. Attractor coding gives escaping points tending to real infinity green through orange and points remaining finite coloured by decreasing blue by iteration, combined with redness corresponding to the attractor period. (4) attractor coded inverse quadratic. (Highlights geometrical details lost in high

iteration numbers).

The 'critical' point for the Mandelbrot set is chosen to represent a prominent critical landmark. For zeta, eta, mu, and sigma it is chosen to be 1000 corresponding to the limit $f(z)\rightarrow 1$ as real(z)$\rightarrow$ +?. For gamma it is 1.465. For Xi it is 0. Other critical points can be chosen using the advanced settings and inputing CX CY values under Mandelbrot mode, just as c values can be inserted for any Julia set. A list of critical points for zeta eta and xi is provided below, or can be approximated using the values of observed zeros of dzeta etc. You can also input CX and CY in function mode to generate a transform of the function, whose zeros are at the fixed points of the critical values [f(cv)-v=0 or f(c+v)-v=0] and hence candidates for being in Mandelbrot kernels of the critical points.

You can flip easily between Mandelbrot mode and transformed function mode using 'keep scope on click'. To exit reset. This can potentially find a finite number, out of the infinite collection of kernels, but sometimes the one located is vanishingly small, leading to blowing the mantissa of the floationg point doubles before we reach it.
Not all the functions have meaningful Mandelbrot sets. The zeta attracting fixed point is at -0.2959050055752.



Colour scheme 1 uses attractor coded colours. At left the incipient and actual attracting periods of bays in the multiplicative Mandelbrot set are shaded with increasing redness. The periods can then be verified by inspecting local Julia sets. Right an additive Julia set of $\zeta(z)$ distinguishes escaping points (blue) from attracting periods in the negative reals (red) and positive (green) shaded by iterations to reach ε-periodicity, aiding in investigating their bifurcations.

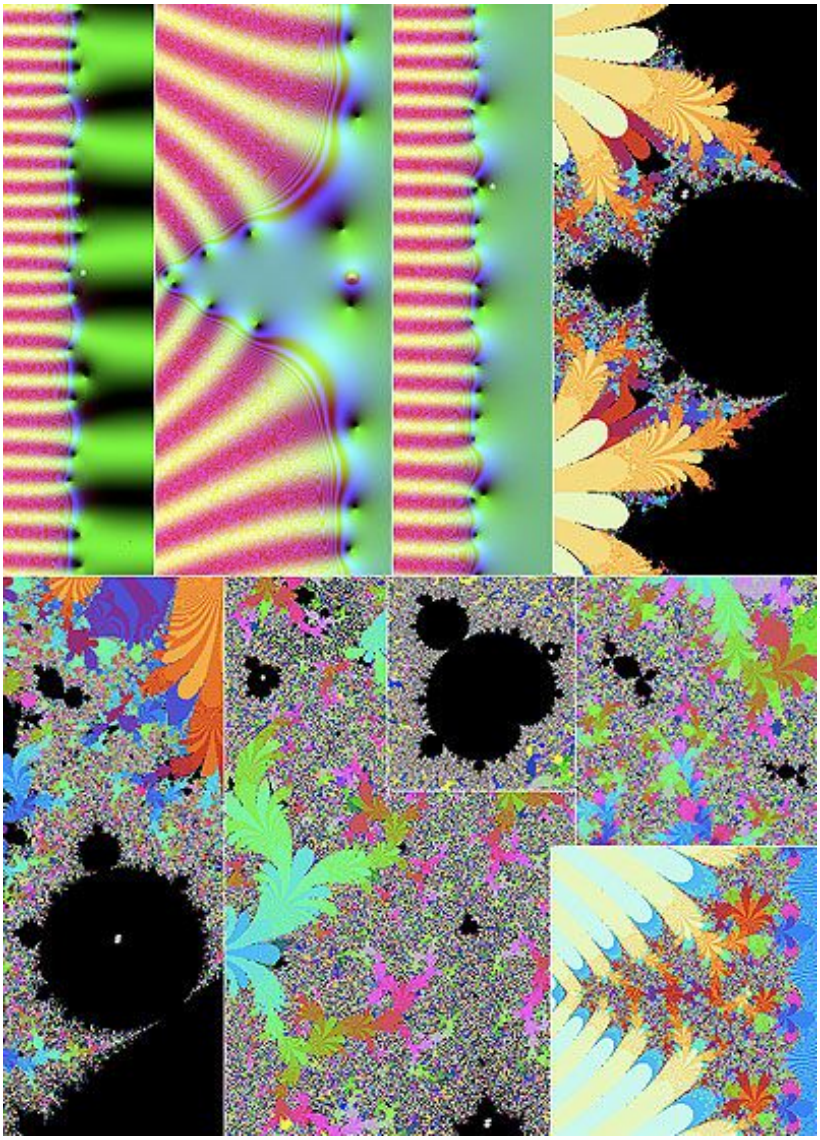
Julia: (0) Sine wave colours, (1) attactor coded colours, (2) RGB ranked colours, (3) rainbow with potential function on escaping points. Colour coded Julia attractors have blue shaded to escaping real, red shaded to periodic attractor, green non-negative periodic attractor, with grey indeterminate. The red and green are tinged with blue to indicate the period. (4) attractor coded inverse quadratic.

For exploring the higher zeros of zeta, a high number of series terms can be set using the slider. Values of the RZ function outside the critical strip [0,1] are set to 0 after y=?450i because the sin/cos component of the analytic continuation overflows to Inf. This leaves the zeros in the critical strip still well rendered up to values in the millions with the highest number of zeta iterations set in the slider.

Blue skies exploration of an additive Mandelbrot kernel near z95=0.78+95.29i. Top left to right. (1) The derivative function wih the chosen critical point starred. (2) The modified zeta function showing fixed values. (3) A further 'north' section of the same function showing the principal point starred close to the location of the original critical point. (4) The Mandelbrot kernel with bulbs. Bottom left to right: The hashed and starred locations point to a satellite Mandelbrot island whose period 3 bulb generates a Julia set (lower right) with period 3 kernels (mid right).

Blue-Skies Method for Finding Mandelbrot Kernels for Zeta Functions

This method enables exploration of a wide range of Mandelbrot sets of zeta functions intuitively taking advantage of the software algorithms without having to do complex arithmetic to look for the needle in the haystack.



Step 1: Explore the critical points of the function using the ***dz*** derivative option. Choose a critical point and scale the image by dragging rectangles to focus on a small region around the critical point. If the critical point has strip-like artefacts due to function term round-off, tick the ***full terms*** option while the process is drawing the immediate neighbourhood. If the imaginary value is large, you may need to increase the function terms to get an accurate critical point.

Step2: Tick the ***keep scope*** option and click the function twice right on the critical point to get Julia representation. This will insert the *c* value of the critical point into the ***Julia / Mand c*** text fields. Now tick ***Hold*** to lock these values in.

Step 3: You now need to decide whether you are exploring an additive or multiplicative Mandelbrot set. Untick ***+c/xc*** if you are using multiplicative.

Step 4: Untick ***dz*** so we are dealing with the original function (this didn't affect the Julia step because ***dz*** is active only in function mode) and do a ***Reset*** and then press ***Set R/I***. This causes the critical point values to modify the zeta function causing the fixed values and principal point of the critical point to be zeros of the modified zeta transfer function.

Step 5: Have a look initially at the principal point, which due to the maths is a double zero so has two yellow angular rays meeting in it. This is where one is most likely to easily find a principal Mandelbrot kernel. In the additive case the principal point will be quite close to the critical point because it only has an additive shift, but in the multiplicative case it may be rescaled and you will have to search for the double zero. You can insert a new Y value and click ***Set X,Y,S*** to move further up or down the imaginary axis.

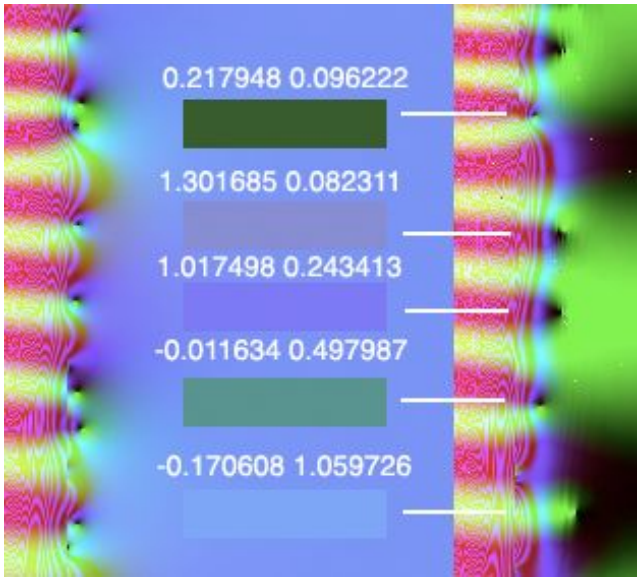
Step 6: With ***keep scope*** ticked, click the point once to go into Mandelbrot mode. Click ***Set R/I*** again to make sure the Mandelbrot is using the critical point you found. Drag a rectangle to zoom in on the neighbourhood of the point. This should give you a Mandelbrot kernel with reasonably well-formed bulbs, dendrites, and satellite black hearts if the number of function terms and iterations is high enough to be accurate at the values used.

Step 7: You can now explore the region of the Mandelbrot kernel or its islands and can check the corresponding Julia kernels by clicking once on a spot with ***Hold*** unticked to pick up the actual *c* value you are now investigating. The Julia set may have only very tiny connected periodic kernels representing the local dynamics near the Mandelbrot kernel and in the multiplicative case these may appear only in a region of the Julia set close to the principal point.

Non-trivial critical values of the Dedekind zeta function

Step 8: You can also find a critical value to use with ***crVal*** by locating the critical point as a zero of dz scaling down to a microscopic domain around it and unchecking dz and refreshing to give the value of the function, which is stationary at the critical point. This can then be found by saving a bitmap and viewing it as a text file where the stationary value will be at the midpoint (see bitmaps below). Use the smallest size

screen for a smaller bitmap. Selected critical values for zeta are included below.



The Functions:

The package is designed to facilitate research into the widest spectrum of zeta-related and L-functions possible. Each of zeta, eta and all the **L-function variants L(m,n)** for m up to 63 for all n and for m up to 1000 for n=1, are encoded as a generalized zeta function which can then be differentiated, or have any of the functions from xi, through those like lambda and mu whose coefficients are important arithmetic functions, to the Newtons method's and dfferentiation applied to them. This creates a widest a diversity of exploratory functions as possible.

Zeta

$$\zeta(z)=\sum_{n=1}^{\infty}\frac{1}{n^z}=\prod_{p\ \text{prime}}\frac{1}{\left(1-p^{-z}\right)}$$

Zeta is represented here in terms of Eta as:

$$\zeta(z)=\left(1-2^{1-z}\right)^{-1}\sum_{n=1}^{\infty}(-1)^{n+1}n^{-z}\quad \text{for real(z)?0}$$

and by analytic continuation using:

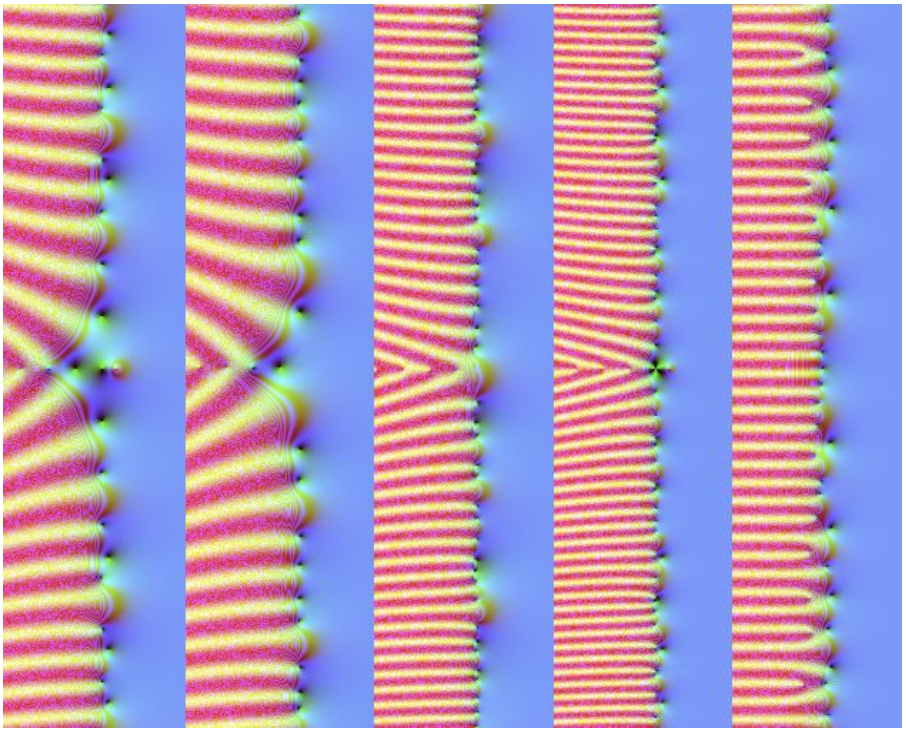
$$\zeta(z)=2^z\pi^{z-1}\cos\left(\frac{\pi(1-z)}{2}\right)\Gamma(1-z)\zeta(1-z)\ \text{for real}(z)<0.$$

http://en.wikipedia.org/wiki/Riemann_zeta_function

Eta

$$\eta(z)=\left(1-2^{1-z}\right)\zeta(z)=\sum_{n=1}^{\infty}(-1)^{n+1}n^{-z}$$

Used to extend Zeta from real(z)>=1 down to real(z)>=0.
http://en.wikipedia.org/wiki/Dirichlet_eta_function



Dirichlet *L*-functions: (Left to right) $L(5,1)$ with regular zeros on $x = 0$ as well as non-trivial zeros on $x = 1/2$. $L(5,2)$ with asymmetric non-trivial zeros on $x = 1/2$. $L(61,2)$ and $L(666,1)$ similar to $L(5,2)$ and $L(5,1)$. Right the period 10 non-*L*-function with $X=\{0,1,0,-1,0,0,0,1,0,-1\}$ has zeros manifestly varying wide of the critical line.

Dirichlet *L*-functions $L(z,\chi)=\sum_{n=1}^{\infty}\chi(n)n^{-z}=\prod_{p\ prime}\left(1-\chi(p)p^{-z}\right)^{-1}$ where $\chi(n), n=0,\cdots,k-1$ is a Dirichlet character of period *m*. Enter *m* and *n* into the *L*(*m*,*n*) text field. Each of the other variant zeta functions can be applied to any of the *L*-functions as well as zeta and eta. The viewer calculates the finite residue groups *Z*/*Zn* generating the characters, factors for sub-periodicities and for conductors, applies the correct functional equation and prime multiplicative factors to give the full representation on the complex plane. A Mellin transform is performed in the central basin to improve fidelity.

Dirichlet, Hecke, elliptic curve and modular form *L*-functions are generated in five phases, illustrated here for Dirichlet:

1: Principal functions and those with similar factorizations are expressed in terms of zeta or a lower character *L*-function: $L(z,\chi)=\prod_{p|N}\left(1-p^{-z}\right)\zeta(z)$

2: Non-primitive functions are expressed in terms of primitives: $L(z,\chi_N)=L(z,\chi_M), M\mid N$

3: For primitives, in the right half-plane $s \geq 1/2$, we calculate the series: $L(z,\chi)=\sum_{n=1}^{\infty}\frac{\chi(n)}{n^z}$

4: In the left half plane we use the analytic continuation where $a=\{\chi(-1)\equiv -1\}$

$$L(z,\chi)=i^{-a}N^{-1/2}\sum_{n=1}^{N}\chi(n)e^{2\pi ni/N}(\pi/q)^{-\frac{1-z+a}{2}}(\pi/q)^{\frac{z+a}{2}}\Gamma\left(\frac{1-z+a}{2}\right)/\Gamma\left(\frac{z+a}{2}\right)L(1-z,\chi)$$

5: In the central critical strip, for -0.15<*x*<1.2, -7<y<7, we use the Mellin transform:

$$L(z,\chi)=\pi^{-(z+a)/2}/\Gamma((z+a)/2)\left(\int_{1/N}^{\infty}y^{z/2}\theta_{\chi}(iy)\frac{dy}{2y}+\frac{-i^{a}N^{1-z}}{\sum_{n=1}^{N}\overline{\chi}(n)e^{2\pi ni/N}}\int_{1/N}^{\infty}y^{(1-z)/2}\theta_{\overline{\chi}}(iy)\frac{dy}{2y}\right)\ \text{where}\ \theta_{\chi}(iy)=\begin{cases}\sum_{n=1}^{\infty}\chi(n)e^{-\pi n^2y}, & a=0\\ \sum_{n=1}^{\infty}\chi(n)ny^{1/2}e^{-\pi n^2y}, & a=1\end{cases}$$

Gamma the complex number extension of integer factorial *n*!
By default, the Lanczos approximation is used:

$$\Gamma(z+1)=\sqrt{2\pi}\left(z+g+\frac{1}{2}\right)^{z+\frac{1}{2}}e^{-\left(z+g+\frac{1}{2}\right)}A_g(z),\quad \Gamma(1-z)\Gamma(z)=\frac{\pi}{Sin\pi z},\ \mathrm{Re}(z)<0$$

$$A_g(z)=\frac{1}{2}p_0(g)+p_1(g)\frac{z}{z+1}+p_2(g)\frac{z(z-1)}{(z+1)(z+2)}+\ldots\ g:\mathrm{Re}\left(z+g+\frac{1}{2}\right)>0$$

with $g=9$ and $p_i(g)$ calculated by Paul Godfrey as constants from the relation:

$$p_k(g)=\sum_{a=0}^{k}C(2k+1,2a+1)\frac{\sqrt{2}}{\pi}\left(a-\frac{1}{2}\right)!\left(a+g+\frac{1}{2}\right)^{-(a+\frac{1}{2})}e^{a+g+\frac{1}{2}}$$

where
$$C(1,1)=C(2,2)=1,\ C(i,1)=-C(i-2,1)$$
$$C(i,i)=2C(i-1,i-1),\ C(i,j)=2C(i-1,j-1)-C(i-2,j),\ i>j$$

are coefficients of the Chebyshev polynomial matrix. See Matlab toolbox for details.

http://en.wikipedia.org/wiki/Lanczos_approximation
or alternatively the product formula

$$\Gamma(z)=\frac{1}{z}\prod_{k=1}^{\infty}\frac{(1+1/k)^z}{1+z/k}$$

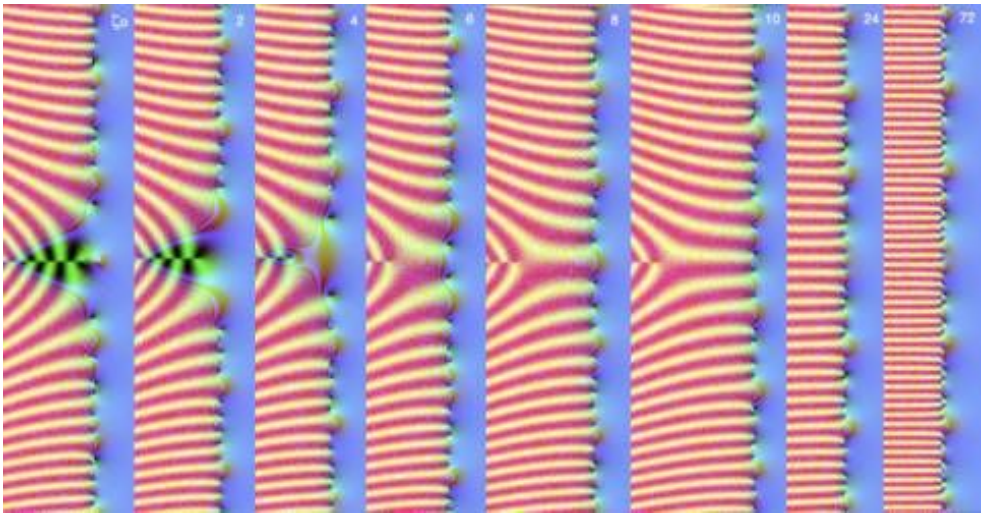
with $n=10^k$ terms if **alt gamma** is set to a non zero value.
http://en.wikipedia.org/wiki/Gamma_function

$\xi(z)$ and its additive and multiplicative parameter planes show subtle symmetry-breaking.

Xi

$$\xi(z) = \Gamma\left(\frac{z}{2}+1\right)(z-1)\pi^{-\frac{z}{2}}\zeta(z)$$

Provides a symmetrical presentation of the zeros on x=1/2. It is now inclded as a sepatre check box which works for all functions possessing a functional equation.
http://en.wikipedia.org/wiki/Xi_function
Note: Xi is rotated so the zeros are on the critical line, rather than the real axis.



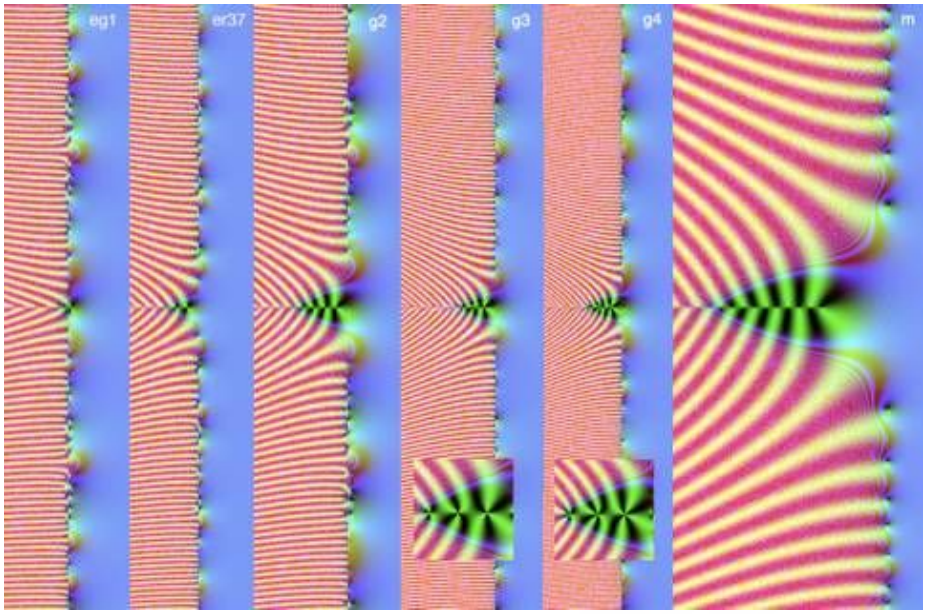
Profiles of the Dedekind zeta and Hecke *L*-functions for $\boldsymbol{Z}[i]$, the extension to the Gaussian integers.

DedekindHekke Dedekind zeta and Hecke *L*-functions of field extensions of the rationals $\boldsymbol{Q}$. We look at those of the Gaussian integers $\boldsymbol{Z}[i]$, defined by appending *i* to the integers, resulting in the lattice of complex numbers with integer real and imaginary parts.

Here we have $N\alpha = \alpha\bar{\alpha} = |\alpha|^2$, so $\zeta_o = \sum_{0\neq\alpha\in o} \frac{1}{(N\alpha)^z} = \frac{1}{4}\sum_{m,n \text{ not both } 0} \frac{1}{(m^2+n^2)^z}$. This has a functional equation $\pi^{-z}\Gamma(z)\zeta_o(z) = \pi^{-(1-z)}\Gamma(1-z)\zeta_o(1-z)$.

Correspondingly we have Hecke *L*-functions defined as follows. Consider the multiplicative group $\chi : Z[i] \to S^1,\ \chi(\alpha) \to (\alpha/\bar{\alpha})^l,\ l \in Z$. To give the same value on every generator this requires *l* to be trivial on units, hence $1 = \chi(i) = \left(\frac{i}{-i}\right)^l = (-1)^l,\ \text{so } l \in 2Z$. We then have for each such *l* a Hecke *L*-function:

$L(z,\chi) = \sum_{0\neq\alpha\in o} \frac{\chi(\alpha)}{(N\alpha)^z} = \frac{1}{4}\sum_{m,n \text{ not both } 0} \frac{(\alpha/\bar{\alpha})^l}{(m^2+n^2)^z} = \prod_{\omega \text{ prime}} \frac{1}{1-\chi(\omega)(N\omega)^{-z}}$ where the primes are now those of Gaussian integers, units ±1 or ±*i* times one of 3 types: 1+ *i* or a real prime which isn't a sum of squares (*p* mod 4 = 3), or has coefficients squaring to a prime (*p* mod 4 = 1). Again we have a functional equation $\pi^{-(z+|l|)/2}\Gamma(z+|l|/2)L(z,\chi) = (-1)^l\pi^{-(1-z+|l|)/2}\Gamma(1-z+|l|/2)L(1-z,\chi)$. The profiles of these functions with their analytic continuations are shown above. The profiles are again generated using a Dirichlet series, analytic continuation and a Mellin transform in the centre.



From left to right, L-functions of the genus-1 elliptic curve $y^2+y=x^3-7x+6$, the elliptic curve $y^2+y=x^3+2x^2+(19+8\omega)x+(28+11\omega),\ \omega=(1+\sqrt{37})/2$ over $K=Q(\sqrt{37})$, the genus-2 curve $y^2+(x^3+x+1)y=x^5+x^4$, the genus-3 curve $y^2+(x^3+x^2+x+1)y=x^7+2x^6+2x^5+x^4$, the genus-4 curve $y^2+(x^5+x+1)y=x^7-x^6+x^4$, and the modular cusp form $\Delta(z)=\sum_{n\geq 1}\tau(n)e^{2\pi inz}$, of weight 12, the 'modular discriminant;, using Ramanujan's Tau function

$$\tau(n)=(5\sigma(n,3)+7\sigma(n,5))\frac{n}{12}-35\sum_{k=1}^{n-1}(6k-4(n-k))\sigma(k,3)\sigma(n-k,5)\ ,\ \text{where}\ \sigma(n,k)=\sum_{d\mid n}d^k$$

Riemann zeta

$$\zeta(s)=\pi^{s/2}/\Gamma\left(\frac{s}{2}\right)\int_0^\infty y^{s/2}(\theta(iy)-1)\frac{dy}{2y},\ \phi(t)=e^{-\pi t^2},\ \theta(iy)=\sum_{n\in Z}\phi(ny)=\sum_{n\in Z}e^{-\pi n^2 y},\ \theta(iy)=\frac{1}{\sqrt{y}}\theta\left(\frac{i}{y}\right)$$

$$\pi^{-s/2}\Gamma\left(\frac{s}{2}\right)\zeta(s)=\int_1^\infty\left(y^{s/2}-y^{(1-s)/2}\right)(\theta(iy)-1)\frac{dy}{2y}+\frac{1}{2}\left(\frac{1}{s-1}-\frac{1}{s}\right)=\pi^{-(1-s)/2}\Gamma\left(\frac{1-s}{2}\right)\zeta(1-s)$$

Dirichlet L-functions

$$L(s,\chi)=\pi^{(s+a)/2}/\Gamma\left(\frac{s+a}{2}\right)\int_0^\infty y^{s/2}\theta_\chi(iy)\frac{dy}{2y}$$

$$=\pi^{(s+a)/2}/\Gamma\left(\frac{s+a}{2}\right)\left(\int_{1/N}^\infty y^{s/2}\theta_\chi(iy)\frac{dy}{2y}+\frac{-i^a N^{1-s}}{\langle\bar{\chi},\varphi\rangle}\int_{1/N}^\infty y^{(1-s)/2}\theta_{\bar{\chi}}(iy)\frac{dy}{2y}\right),\ \langle\chi,\varphi\rangle=\sum_{n=1}^N\chi(n)e^{2\pi ni/N}$$

$$\text{where}\ \theta_\chi(iy)=\begin{cases}\sum_{n=1}^\infty\chi(n)e^{-\pi n^2 y}, & a=0\\ \sum_{n=1}^\infty\chi(n)ny^{1/2}e^{-\pi n^2 y}, & a=1\end{cases},\ \varepsilon(\chi)=\frac{\sqrt{N}}{\sum_{n=1}^N\bar{\chi}(n)e^{2\pi ni/N}},\ \theta_\chi(iy)=\frac{-i\langle\chi,\varphi\rangle}{N\sqrt{y}}\theta_{\bar{\chi}}\left(\frac{i}{N^2y}\right)$$

$$N^{s/2}\pi^{-(s+a)/2}\Gamma\left(\frac{s+a}{2}\right)L(s,\chi)=\int_{1/N}^\infty(Ny)^{s/2}\theta_\chi(iy)\frac{dy}{2y}+\varepsilon(\chi)\int_{1/N}^\infty(Ny)^{(1-s)/2}\theta_{\bar{\chi}}(iy)\frac{dy}{2y}$$

$$=\varepsilon(\chi)\int_{1/N}^\infty\varepsilon(\bar{\chi})(Ny)^{s/2}\theta_\chi(iy)\frac{dy}{2y}+\int_{1/N}^\infty(Ny)^{(1-s)/2}\theta_{\bar{\chi}}(iy)\frac{dy}{2y}=N^{(1-s)/2}\pi^{-((1-s)+a)/2}\Gamma\left(\frac{(1-s)+a}{2}\right)L(1-s,\bar{\chi})$$

Dedekind zeta

$$\zeta_*(s)=\frac{1}{4}\sum_{m,n\ \text{not both}\ 0}\frac{1}{(m^2+n^2)^s}=\pi^s/\Gamma(s)\int_0^\infty y^s(\theta(iy)-1)\frac{dy}{4y},\ \theta(iy)=\sum_{m,n\in Z}e^{-\pi(m^2+n^2)y},\ \theta(iy)=\frac{1}{y}\theta\left(\frac{i}{y}\right)$$

$$\pi^{-s}\Gamma(s)\zeta(s)=\int_1^\infty\left(y^s+y^{1-s}\right)(\theta(iy)-1)\frac{dy}{4y}+\frac{1}{4}\left(\frac{1}{s-1}-\frac{1}{s}\right)=\pi^{-(1-s)}\Gamma(1-s)\zeta(1-s)$$

Hecke L-functions

$$L(s,k)=\pi^{(s+|k|)}/\Gamma(s+|k|)\int_1^\infty\left(y^s+y^{1-s}\right)\theta_\chi(iy)\frac{dy}{4y},\ \theta_\chi(iy)=\frac{(-1)^{k1}}{y}\theta_\chi\left(\frac{i}{y}\right)$$

$$\theta_\chi(iy)=\sum_{m,n\in Z}(m+ni)^{2k}y^{|k|}e^{-\pi(m^2+n^2)y}=y^{k1}\sum_{m=0}^\infty\sum_{n=0}^\infty\text{Re}((m+ni)^{2k})e^{-\pi(m^2+n^2)y}$$

$$\pi^{-(s+|k|)}\Gamma(s+|k|)L(s,k)=\int_1^\infty\left(y^s+y^{1-s}\right)\theta_\chi(iy)\frac{dy}{4y}=(-1)^{|k|}\pi^{-((1-s)+|k|)}\Gamma((1-s)+|k|)L(1-s,k)$$

Abstract/Raw L-functions such as those of elliptic curves and modular forms can be uploaded using the parameter files and Terminal C-scripts in the files included with the package, which are also able to specify varying functional equations conductors and gamma factors. See the section on uploading parameter files and the installation file set for examples of each. If full terms is selected, the algorithm will also use a Mellin transform for elliptic curves and modular forms to give a fully smoothed L-function profile in the central region. The analytic continuation and Mellin integral transform formulas are shown on the right. Raw portraits without a functional equation can also be generated.

With 1.7.3, an experimental general inverse Mellin tranform has been implemented which enables Mellin tranform portraits of geneneral motivic L-functions in the central region.

The analytic continuation of each class of *L*-function is achieved by applying a product of gamma functions $\gamma(s)=\Gamma\left(\frac{s+\lambda_1}{2}\right),\cdots,\Gamma\left(\frac{s+\lambda_d}{2}\right)$ derived from the symmetries in the θ functions, each of which is generated as $\theta(t)=\sum_{n=1}^{\infty}a_n\phi\left(\frac{nt}{A}\right)$, where $L(s)=\sum_{n=1}^{\infty}a_n n^{-s}$. For each of the cases outlined above, ϕ has a known exponential form and other cases include Bessel functions and more general harmonic exponentials, as shown in fig 37b. Because we can express the complete gamma factor as a Mellin transform: $\gamma(s)=\int_0^{\infty}\phi(t)t^s\frac{dt}{t}$, and the *L*-function is a Mellin transform of theta:

$$\int_0^{\infty}\theta(t)t^s\frac{dt}{t}=\sum_{n=1}^{\infty}a_n\int_0^{\infty}\phi\left(\frac{nt}{A}\right)t^s\frac{dt}{t}=\sum_{n=1}^{\infty}a_n\int_0^{\infty}\phi(t)\left(\frac{At}{n}\right)^s\frac{dt}{t}=A^s\gamma(s)\sum_{n=1}^{\infty}a_n n^{-s}=A^s\gamma(s)L(s),\ A=N^{1/2}\pi^{-d/2}$$

we can thus reverse the process of defining the gamma factors through the Mellin transforms and derive the Mellin transform of an *L*-function by specifying its gamma factors and generating $\phi(t)$ as the inverse Mellin transform of $\gamma(s)$: $\phi(t)=\int_{c-i\infty}^{c+i\infty}\gamma(s)t^{-s}\,ds$ to get θ . This integral can easily be performed numerically on an interval up the critical strip giving good correspondence with known $\phi(t)$, as shown in fig 37b. Note that this behaves in the manner of a Fourier transform on $\gamma(s)$, as t^{-s} is sinusoidal on the imaginary dimension. The functional equations are, in turn, derived from θ symmetries such as $\theta(t)=\varepsilon t^w\theta(1/t)-\sum_j r_j t^{p_j}$ where p_j are any pole singularities. For example, for $\zeta(s)$ by Poisson summation equating the function sum to its summed Fourier transforms: $\sum_{n\in\mathbb{Z}}f(n)=\sum_{n\in\mathbb{Z}}\hat{f}(k),\ \hat{f}=F(f)$, we have $\theta(1/t)=\theta(t)+t+1$, since the Fourier transform of $e^{-\pi x^2}$ is $\sqrt{\frac{\pi}{n}}e^{-\pi^2k^2/n}$ and $\pi^{-s/2}\Gamma\left(\frac{s}{2}\right)\zeta(s)$ has residues 1, -1 at 0, 1. A similar result holds for modular forms by modularity. We can then evaluate the *L*-function using θ :

Modular forms and elliptic curves

$$f(q)=\sum_{n=1}^{\infty}a_n q^n=\sum_{n=1}^{\infty}a_n e^{2\pi inz}\in S_k(\Gamma_1(N)),\ \text{By modularity}\ f\left(-\frac{1}{Nz}\right)=CN^{-k/2}(-Nz)^k f(z)$$

$$\int_0^{\infty}f(z)z^s\frac{dz}{z}=\int_0^{\infty}\sum_{n=1}^{\infty}a_n e^{2\pi inz}z^s\frac{dz}{z}=\int_0^{\infty}\theta(z)z^s\frac{dz}{z}=\sum_{n=1}^{\infty}a_n\int_0^{\infty}z^s e^{2\pi inz}\frac{dz}{z}$$

[where $t=-2\pi inz$, $\phi(t)=e^{-2\pi t}$, $\theta(z)=\sum_{n=1}^{\infty}a_n\phi(iz)$]

$$=\sum_{n=1}^{\infty}a_n\left(\frac{-1}{2\pi n}\right)^s\int_0^{\infty}e^{-t}t^s\frac{dt}{t}=(-2\pi i)^{-s}\Gamma(s)\sum_{n=1}^{\infty}\frac{a_n}{n^s}=(-2\pi i)^{-s}\Gamma(s)L(f,s)=(2\pi)^{-s}\Gamma(s)L(f,s)$$

$$\int_0^{\infty}f(z)z^s\frac{dz}{z}=\int_0^{1/\sqrt{N}}f(z)z^s\frac{dz}{z}+\int_{1/\sqrt{N}}^{\infty}f(z)z^s\frac{dz}{z}=\int_{1/\sqrt{N}}^{\infty}\left(f(z)z^s+i^kCN^{-k/2}f(z)\left(-\frac{1}{Nz}\right)^{s-k}\right)\frac{dz}{z}$$

$$=\int_{1/\sqrt{N}}^{\infty}\left(i^s f(iy)y^s+i^kCN^{-k/2}f(iy)\left(-\frac{1}{Niy}\right)^{s-k}\right)\frac{dy}{y}=\int_{1/\sqrt{N}}^{\infty}\left(i^s y^s+i^kCN^{-k/2}i^{s-k}N^{k-s}y^{k-s}\right)f(iy)\frac{dy}{y}$$

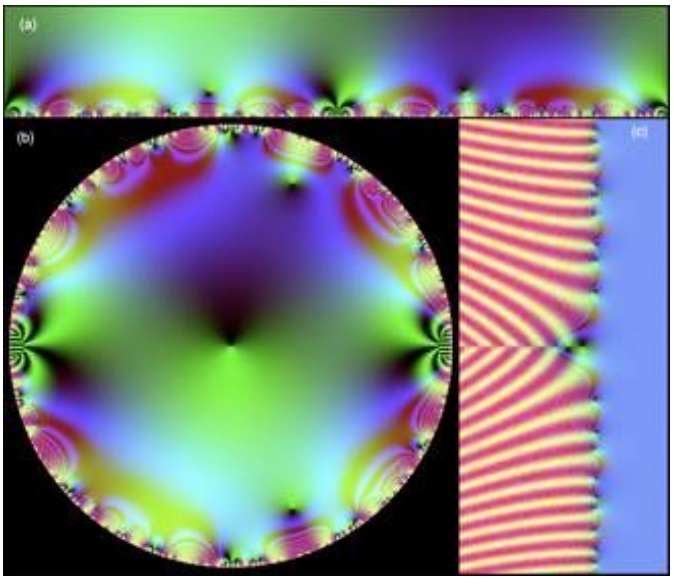
$$=i^s\int_{1/\sqrt{N}}^{\infty}\left(y^s+CN^{k/2-s}y^{k-s}\right)f(iy)\frac{dy}{y}=i^s\int_{1/\sqrt{N}}^{\infty}\left(y^s+CN^{k/2-s}y^{k-s}\right)\sum_{n=1}^{\infty}a_n e^{-2\pi ny}\frac{dy}{y}$$

$$L(f,s)=(2\pi)^s/\Gamma(s)\int_{1/\sqrt{N}}^{\infty}\left(y^s+CN^{k/2-s}y^{k-s}\right)\sum_{n=1}^{\infty}a_n e^{-2\pi ny}\frac{dy}{y}$$

$$\Lambda(f,s)=N^{s/2}(2\pi)^{-s}\Gamma(s)L(f,s),\ \Lambda(f,s)=i^k w\Lambda(f,k-s),\ w=\pm1$$

$L^*(s)=A^s\gamma(s)L(s)=\int_1^{\infty}\theta(t)\left(t^s+\varepsilon t^{w-s}\right)\frac{dt}{t}+\sum_j\frac{r_j}{p_j-s}$ hence $L^*(s)=A^s\gamma(s)L(s)=\int_1^{\infty}\theta(t)\left(t^s+\varepsilon t^{w-s}\right)\frac{dt}{t}+\sum_j\frac{r_j}{p_j-s}$. Hence we have the functional equation: $L^*(s)=\varepsilon L^*(w-s)$.

See ***parameter files*** for details.



L-function of modular form S_2(Gamma0,26) (right) compared with
the power series on |*z*|<1 (below)
and the Fourier series in the positive half-plane (above)

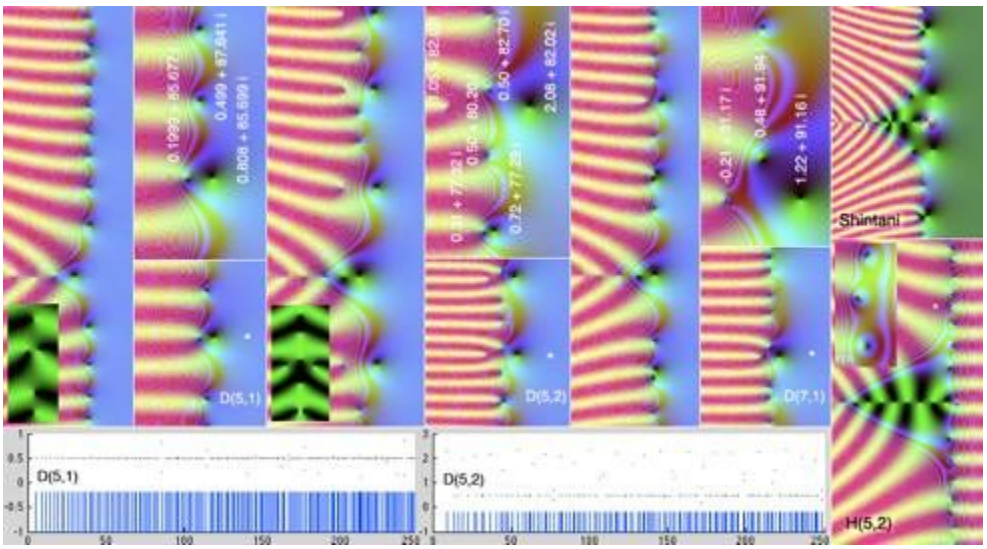
Power Series gives the power series *P*(*z*)=Sum[1,?∞](*a*(*n*)*z*^{n}) corresponding to the Dirichlet series *L*(*z*)=Sum[1,?∞](*a*(*n*)/*n*^{z}) either of character Xm, Xn or uploaded as a coefficient set (see below).

Fourier does the Fourier series viz F(z)=Sum[1,?∞](a(n)exp(2π*inz*)) either of character Xm, Xn or uploaded as a coefficient set (see below).

Both the two above functions are useful when examining **modular forms** which are expressed as Fourier series in the upper half-plane or as power series in the unit circle. By selectng **Escape Only**, you

can set the areas where the functions are formally undefined to black. By also selecting **Orbit Trap** you add 1 to the result which enables loading an Eisenstein series leaving out its leading 1 in the same way as an L-function so that one can do a Fourier or Taylor portrait of all the components of a modular form space.

Weierstrass plots a Weierstrass modular function using the first two complex coefficients of any Dirichlet or loaded abstract L-function as periodic vectors w1 and w2 on a parallelogrammatic grid extending for 2*sqrt(fnterms) to illustrate what modular functions associated with elliptic curves look like.



Davenport-Heilbronn zeta functions (5,1), (5,2), and (7,1) possess functional equations demonstrating they are meromorphic on the complex plane but lack an Euler product and have a rich array of non-trivial zeros off the critical line demonstrating a functional equation is insufficient for RH. Significant is the large number of quasi-critical zeros (below) with a sparse spread of off-critical ones, which shows why many critical zeros don't necessarily imply all. Intriguingly the quasi-critical zeros fall at a minimum divergence from criticality for the parameter *xi*, although all values have a functional equation expressed as the sum of rotated *DL*-functions (see appendix 3), so this non-superimposed form of the functional equation looks pivotal. Hurwitz zeta (5/2) likewise shows sums of *L*-functions have off-critical zeros. The symmetrical placing of the zeros about the critical line in (5,1) and (5,2) is confirmed in the local xi portraits inset.

Davenport-Heilbronn also shows examples of functions with a functional equation but no Euler product which have some zeros off the critical line despite having a symmetric Xi function expression

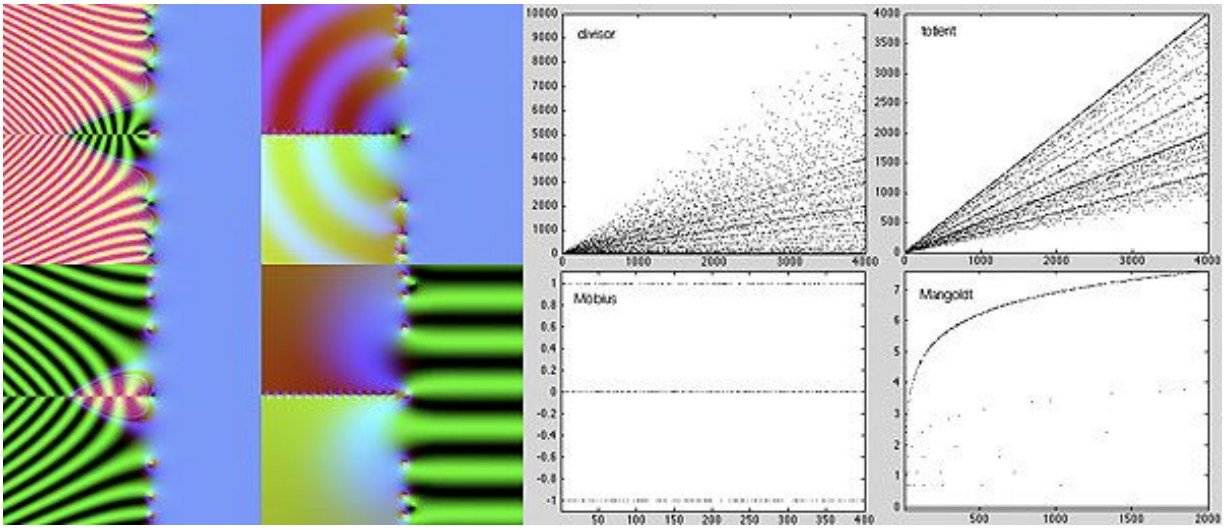
Psi (Digamma)

$$\psi_0(z) = \frac{\Gamma'(z)}{\Gamma(z)}$$ the logarithmic derivative of gamma

$$\psi_0(z) = -\gamma + \sum_{n=1}^{\infty} \frac{z-1}{n(n+z-1)}, \gamma = 0.57721\ 56649\ 01532\ 86060\ 65120\ 90082\ 40243$$

http://en.wikipedia.org/wiki/Digamma_function

The following derived zeta functions will also work with any L-function



Divisor function $\sigma_i(n)$, totient function $\varphi(n)$, M?bius function $\mu(n)$ and Mangoldt function $\Lambda(n)$

The ***Mobius*** function $\frac{1}{\zeta(z)} = \sum_{n=1}^{\infty} \frac{\mu(n)}{n^z}, \mu(n) = \begin{cases} (-1)^k, & n \text{ has } k \text{ distinct prime factors of multiplicity 1} \\ 0 & \text{otherwise} \end{cases}$

http://en.wikipedia.org/wiki/M?bius_function

The ***totient*** function $\frac{\zeta(z-1)}{\zeta(z)} = \sum_{n=1}^{\infty} \frac{\varphi(n)}{n^z}$

$\varphi(n) =$ the number of +ve integers $k \le n$ coprime to n

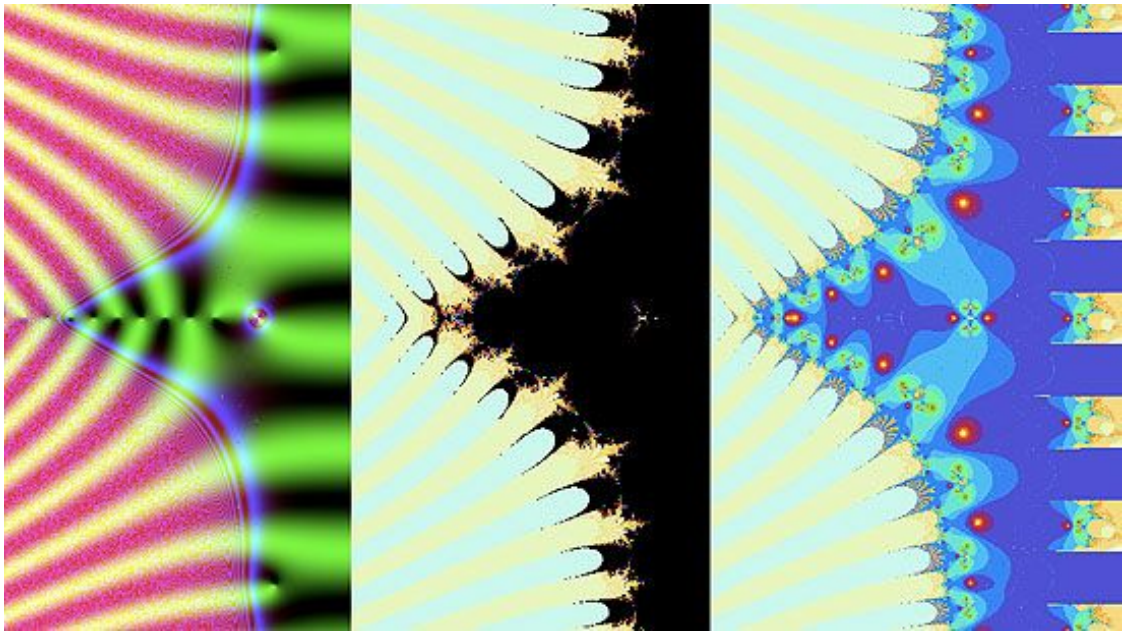
http://en.wikipedia.org/wiki/Totient_function

The ***Mangoldt*** function $\frac{\zeta'(z)}{\zeta(z)} = -\sum_{n=1}^{\infty} \frac{\Lambda(n)}{n^z}, \Lambda(n) = \begin{cases} \log p, n = p^k \\ 0 \text{ otherwise} \end{cases}$

http://en.wikipedia.org/wiki/Mangoldt_function

The ***divisor*** function $\zeta(z)\zeta(z-k) = \sum_{n=1}^{\infty} \frac{\sigma_k(n)}{n^z}, \sigma_z(n) = \sum_{d|n} d^z$

http://en.wikipedia.org/wiki/Divisor_function



$\zeta'(z)$, its additive parameter plane and Julia set of 0, showing the seamless nature of the transition across the *y*-axis.

dGamma, **dXi** and **dZeta** are the derivatives $\Gamma'(z), \xi'(z), \zeta'(z)$.

$$\Gamma'(z) = \psi_0(z)\Gamma(z)$$

The dZ tick-box uses the differential derivative $\zeta'(z) \sim \frac{\zeta(z+\Delta z) - \zeta(z)}{\Delta z}, |\Delta z| << 1$ on the left-half plane for zeta and the theoretical formula for real(z)>0 and the differential derivative throughout for the other functions. The ***dZ*** tick box replicates this for function mode only to locate critical points of any function combination for locating layered Mandelbrot sets, however you can get a second-derivative function portrait by combining the two.

$$\left(1-2^{1-z}\right)\zeta(z)=\sum_{n=1}^{\infty}(-1)^{n+1}n^{-z}$$

$$\left(1-2^{1-z}\right)\zeta'(z)+(-1)^2 2^{1-z}\log(2)\zeta(z)=\sum_{n=1}^{\infty}(-1)^{n+1}\log(n)(-1)n^{-z}$$

$$\zeta'(z)=\left(1-2^{1-z}\right)^{-1}\left[\sum_{n=1}^{\infty}(-1)^{n}\log(n)n^{-z}-2^{1-z}\log(2)\zeta(z)\right]$$

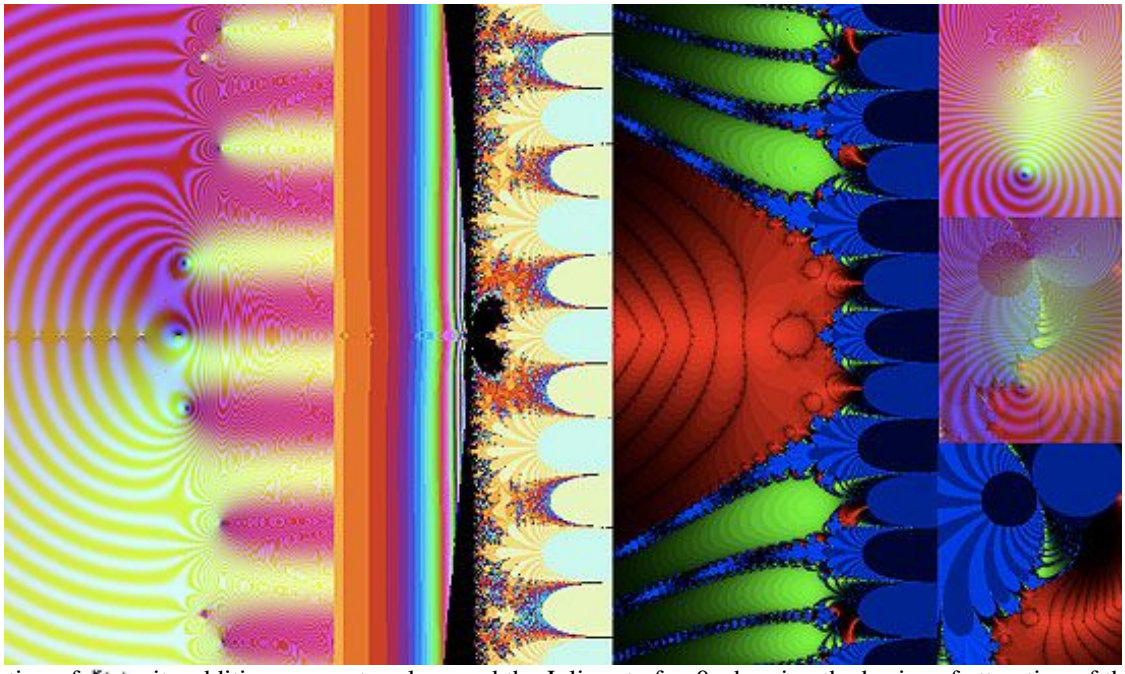
$$\zeta(z)=\pi^{-1}(2\pi)^{z}\sin\left(\frac{\pi z}{2}\right)\Gamma(1-z)\zeta(1-z)$$

$$\zeta'(z)=\pi^{-1}\log(2\pi)(2\pi)^{z}\sin\left(\frac{\pi z}{2}\right)\Gamma(1-z)\zeta(1-z)+\pi^{-1}(2\pi)^{z}\left(\frac{\pi}{2}\right)\cos\left(\frac{\pi z}{2}\right)\Gamma(1-z)\zeta(1-z)$$
$$+\pi^{-1}(2\pi)^{z}\sin\left(\frac{\pi z}{2}\right)(-1)\Gamma'(1-z)\zeta(1-z)-\pi^{-1}(2\pi)^{z}\sin\left(\frac{\pi z}{2}\right)\Gamma(1-z)(-1)\zeta'(1-z)$$

$$\zeta'(z)=\pi^{-1}(2\pi)^{z}\left(\left(\log(2\pi)\sin\left(\frac{\pi z}{2}\right)+\left(\frac{\pi}{2}\right)\cos\left(\frac{\pi z}{2}\right)\right)\Gamma(1-z)\zeta(1-z)\right.$$
$$\left.-\sin\left(\frac{\pi z}{2}\right)\Gamma(1-z)\left(\psi_0(1-z)\zeta(1-z)+\zeta'(1-z)\right)\right)$$

The derivative of Xi can then be derived as follows:

$$\xi'(z)=\frac{1}{2}\Gamma'\left(\frac{z}{2}+1\right)(z-1)\pi^{-\frac{z}{2}}\zeta(z)+\Gamma\left(\frac{z}{2}+1\right)\pi^{-\frac{z}{2}}\zeta(z)$$
$$+\Gamma\left(\frac{z}{2}+1\right)(z-1)\frac{-\log(\pi)}{2}\pi^{-\frac{z}{2}}\zeta(z)+\Gamma\left(\frac{z}{2}+1\right)(z-1)\pi^{-\frac{z}{2}}\zeta'(z)$$

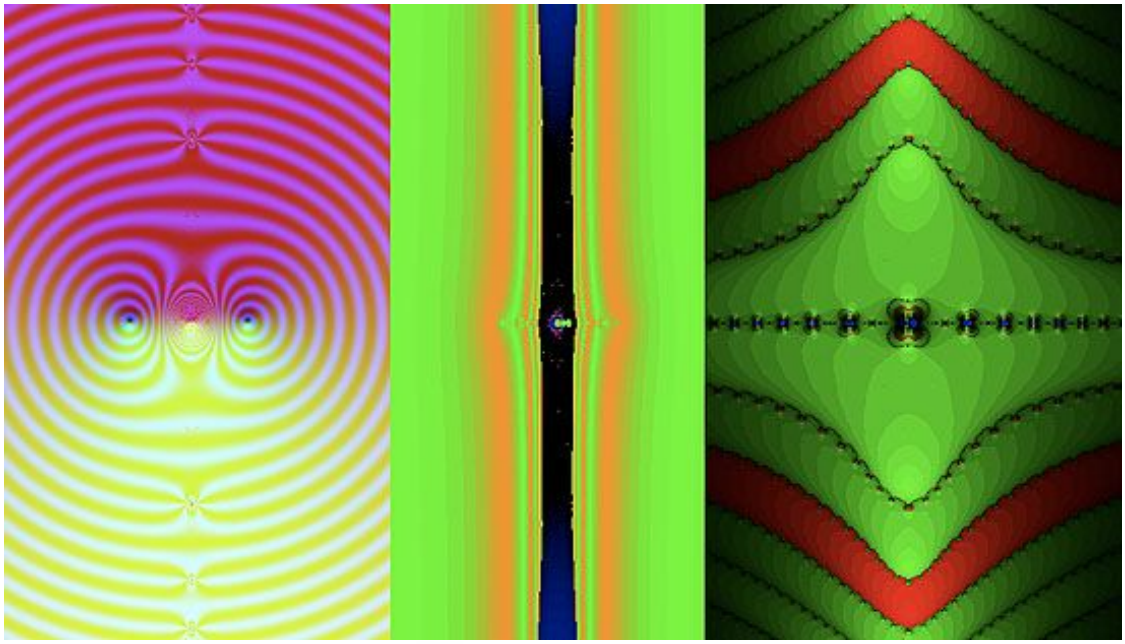


Newton function of $\zeta(z)$, its additive parameter plane and the Julia set of c=0, showing the basins of attraction of the non-trivial zeros (green) and trivial zeros (red). The inset (right) shows the poles caused by the zeros of $\zeta'(z)$ result in a ?dipole? corresponding to a fractal basin of attraction of 0 and a Julia escape singularity.

NewtonZ, **RepellingNZ** and **NewtonX** are the Newton?s method functions for zeta, and xi.

$$N\zeta(z)=z-\frac{\zeta(z)}{\zeta'(z)}$$

They are provided because they naturally relate to the zeros. Xi, dXi and NewtonX are symmetrized through the relations $\xi(z)=\xi(1-z),\ \xi'(z)=-\xi'(1-z)$ to avoid computational errors around the imaginary axis. This may result in some double zeros and double singularities along the critical line at the default values of series terms. The third zeros above and below have been coloured red to highlight the fractal interaction of the zeta zero basins.



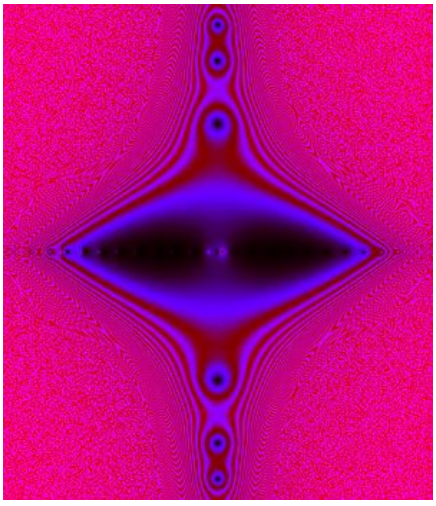
Corresponding Newton?s method images for $\xi(z)$ result in a vertical parameter plane and the Julia set of c=0 forming symmetrical basins of attraction of the zeta zeros.Proving the Riemann hypothesis my require linking the positions of the zeros to discrete criteria such as the separatrices of their basins of attraction.

***z*Zeta(*z*)** is included to explore its relationship with the zeta multiplicative parameter plane.

zetaprod is the product version of zeta which is technically defined only for real(z)>1, included for comparison with the sum formula analytic continuation version. Zetaterms adjusts the number of product terms in the same way as with zeta sum terms.

primeZeta is the function $p\zeta(z) = \sum_{p \text{ prime}} \frac{1}{p^z} = \sum_{k=1}^{\infty} \frac{\mu(k)}{k} \log(\zeta(kz))$ where *mu* is the ***M?bius*** function (see above).
It is included for completeness. It is defined only for real(z)>0.

Warning: Use of the last two functions invokes generating a list of primes as long as function terms, which can cause errors on the first refresh, as it is sill being compiled during run time, if a large value is inserted for function terms.



theta - the Riemann-Siegel theta function

$$\theta(t) = \arg\left(\Gamma\left(\frac{2it+1}{4}\right)\right) - t\frac{\log(\pi)}{2} = \frac{\log\left(\Gamma\left(\frac{2it+1}{4}\right)\right) - \log\left(\Gamma\left(\frac{2it+1}{4}\right)\right)}{2i} - t\frac{\log(\pi)}{2}$$

http://en.wikipedia.org/wiki/Riemann-Siegel_theta_function

RSZ - the Riemann-Siegel *Z* function $Z(t) = e^{i\theta}\zeta\left(\frac{1}{2}+it\right)$
Both functions are rotated by the inverse transformations $t = -i\left(z-\frac{1}{2}\right), z = \frac{1}{2}+it$ so that the zeta zeros are aligned with the critical line *x* = 1/2 (right).
http://en.wikipedia.org/wiki/Z_function

Warning: The last two functions are still experimental. The chequerboard pattern in some views results from ambiguities in the log's principal branch due to the gamma function's circuits around 0, with increasing angle (argument), as the imaginary value of z is increased. However, when ***angle*** is turned off in the advanced controls, these artifacts disappear (right) and the Z function clearly displays the zeta zeros.

Parameters Text file format: [0] Mandel (=2 function =1 Mandelbrot =0 Julia), [1-5] Scale, X, Y, cr, ci, [6] altgamma (0 or 2-4), [7] the current function, [8] split (0, 1, -1 2), [9] orbit trap epsilon (negative exponent 10n), [10] addtive or multiplicative c [11] escape only (0 or 1), [12] color scheme (0 to 3), [13] escape

bound (10-100), [14] number of zeta terms, [15] attractor radius epsilon bound, [16] maximum iterations, [17] do log of fn [18] angle included in colour scheme [19] do mobius transformation [20] cval (holds entered critical point), [21] old fn. [22] real k [23] imag k [24] crval+2*xi [25] full terms+2*dZ+4*Mellin [26] xm [27] xn [28] flip.

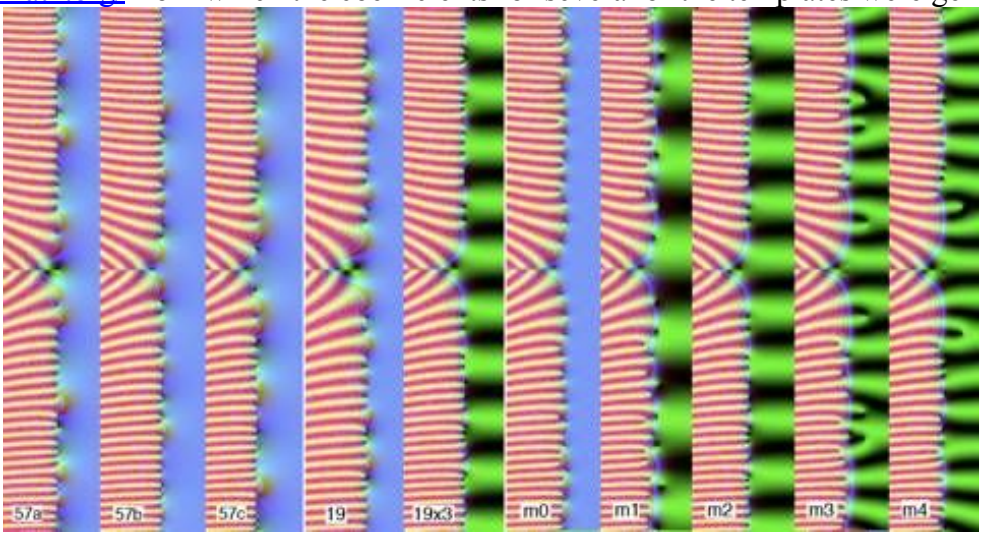
Arbitrary Dirichlet series: You can also add additional parameters to a saved text file or scripted movie file to define an arbitrary generalized Dirichlet series Sum[*an*]*[*n*+*bn*]^-*z*. These are not saved as they cannot be adjusted in the advanced settings. [29] coefficient period length *k* [30] type h (2 or -999 -998). Type 2 has *k* real and imaginary coefficients which can be varied in a movie script. Type -999 has both *k* coefficients and an additional *k* adjustments bn to the integers *n* forming the powers *n*^-*z*. In this way we can generate continuous movies which pass between *L*-functions, even using non integer exponent bases. Type -998 is used to instead multiply the resulting function with integer coefficients by (1-2^(1-z))^-1 to convert an eta type series into its zeta form.

Abstract L-functions: you can upload an abstract zeta/L function using the character period length *k* [29] Dirichlet series coefficients as above with and chr. type *h* = -997, appended after the *k* complex coefficients by adding the following values also as complex coefficients: d=number of gamma factors *g*((*z*+*k*1)/2), *g*((*z*+*kd*)/2), k followed by *k*1, ... *kd*, *N* the conductor, *e* the sign of the functional equation, and finally w the weight. *h* = -995 has the same parameters entered as complex numbers to allow for Dirichlet and Maass *L*-series with complex gamma factors and *e* terms. For *L*-functions with very rapidly growing coefficients which would cause numerical overflow (large weights) enter chr. no -996, scale down the coefficients of *n* by a factor *n*^-*sh* and enter *sh* as a shift factor after the weight.

You can explore abstract *L*-function fractal sets, save an image and parameter file. You can then load the parameters for the image later after making sue you have first loaded the correct *L*-function and the existing abstract function will continue to operate.

If the abstract *L*-function has 2 gamma factors of 0 and 1 respectively and Mellin is set, RZViewer will also use the Mellin transform for a modular form (or elliptic curve) in the region 0,â§*x*,â§*w*, -8,â§*y*,â§8 to improve fidelity in the central basin. This is useful for considering the Birch and Swinnerton-Dyer conjecture. Finally with version 1.7.3 a general Dokchitser type gamma function inverse Mellin transform routine has been instituted for arbitary *L*-functions which is currently experimental, but works successfully on the genus 2 and 3 examples. To activate this ***fullterms*** needs to be set to 4 in the parameter file. In both the modular and general cases, if the number of gamma factors is given a negative sign, the three terms after weight (or the shift factor with -996) will be the height and width of a Mellin transform window and the residue of any pole singularity at *z*=1.

Mac Terminal C-script examples are provided in the attached files to generate parameter files for uploading. For details of these parameters and the original examples see: http://www.dpmms.cam.ac.uk/~td278/computel/. Note that PARI-GP is now incorporated into Sage http://www.sagemath.org/ from which the coefficients for several of the templates were generated.



N=57 modular forms have a five-dimensional echelon basis (right) . There are three *newforms* (left) corresponding to elliptic curve *L*-functions *e*57*a*, *e*57*b*, and *e*57*c*. We also have the Hecke eigenforms *e*19*a* and its 'harmonic' $3^{-z}e19a$, which is equivalent to the

third echelon basis function 38*m*2.

Bitmaps: You can both save and load bitmaps as well as parameter files and tiff images. You can save whole bitmaps as text files by using color option 5 and selecting the Save Bitmap menu item. This makes it possible to generate *L*-functions which couldn't otherwise be accurately portrayed such as the echelon basis functions of modular forms. A terminal c-script is included to enable merging of saved bitmaps as shown in the illustration above for N=57. One can also use computel to generate bitmaps which can be loaded into RZViewer. The image will initially show in its exact size. Since it's a bitmap blowing it up will reduce the resolution of individual pixels. The package includes the PARI-Computel for generating RZViewer bitmaps, based on Tim Dokchitser's generalized Mellin transform algorithm which complements the functional equation algorithms and standard Mellin transform integrals of RZViewer for difficult *L*-functions.

The Sage file in the 57 folder shows the information to form the matrix expressing 57a, 57b, 57c, 19a and 19ax3 in terms of 57m0-4 and its inverse, thus enabling us to express the basis elements in terms of the elliptic curve eigenfunctions.

To regenerate the bitmaps:
1: In RZViewer, load one of the above parameter files for the elliptic curves in the parameter file folder in 57, turn on full terms and open the advanced drawer.
2: Slide the window to downsize it to the smallest possible to avoid overflowing bitmap memory on reload.
3: Set color mode to 5 and set scale e.g. to 25 and click **Set *X,Y,S***.
4: When the window has refreshed with the *L*-function profile, save the bitmap under the same name as the parameter file - e.g. "57a.txt".
5: Do the same for each of the 5 files.
6: Put all the o files in a folder along with merge57e.c open terminal and cd to the folder, then execute "gcc -o rzo merge57e.c -lm && ./rzo". The merge script is already set to use the input filenames you have defined.
7: Rename the output file to something like 57m0.txt
8: Run the same c script again setting opt to 0 - 4 to generate 57m0.txt to 57m4.txt.
9: Upload each of these into RZViewer one at a time using Load Bitmap to view the results and save as required to get tiffs.

Movie Sequence Formats: The text file consists of s eries of floating point numbers. (1) File format 1-3 (2) N<=1000 number of frames (3) 1/0 Mandelbrot/Julia followed by a sequence of frame parameters screen X, Y, Scale CX, CY. Format 1: only the start and end parameters are specified and DHViewer will generate N frames in a linear sequence between these values. Format 2: the file supplies a series of N parameters on any trajectory, for instance computed values running around the cardioid of a Mandelbrot set. Format 1 and 2 files can either generate blowups of Mandelbrot sets or movies of Julia sets as CX and CY vary. The sequence file has no parameters except mandel, so you can set all the other settings manually before saving the movie. If you want to determine in advance the complete parameters for a movie run, save your parameters and load them before loading the movie sequence file. Format 3: The format file contains only parameters 1 & 2 and is accompanied by a series of full parameter files, generated by another application such as Matlab. If the format file is myfile.txt, the parameter files need to be numbered myfile0.txt - myfileN,txt. This format enables batch programming of any sequence of images automatically.

First Zeta Critical Points
x-axis
z = -2.7172 z(z) = 0.0092
z = -4.9368 z(z) = -0.0040
z = -7.0746 z(z) = 0.0042
z = -9.1705 z(z) = -0.0079
z = -11.2412 z(z) = 0.0227
z = -13.2956 z(z) = -0.0937
z = -15.3387 z(z) = 0.5206
z = -17.3739 z(z) = -3.7436
z = -19.4031 z(z) = 33.8083

z = -21.4279 z(z) = -374.4187
z = -23.4492 z(z) = 4988

Critical line (dzeta approximation plus Newton)
z = 2.463356 + 23.297678i, z(z) = 0.9289 + 0.0308i
z = 1.286578 + 31.708124i, z(z) = 0.7073 + 0.0110i
z = 2.306454 + 38.489796i, z(z) = 0.9919 - 0.0931i
z = 1.382872 + 42.2909775i, z(z) = 0.7930 + 0.1273i
z = 0.9646998675 + 48.8471714103i
z = 2.1016662620 + 52.4314144061i
z = 1.8960801201 + 57.1342532310i
z = 0.8487341239 + 60.1408464582i z(z) = 0.4881 + 0.1283i
z = 1.2073178358 + 65.9199666872i z(z) = 0.7776 - 0.2061i
z = 0.7806300097 + 95.2929688274i z(z) = 0.4295 + 0.0779i

Eta Critical Points
z = -3.0430 e(z) = -0.1252
z = -5.2330 e(z) = 0.2673
z = -7.3360 e(z) = -1.2390
z = -9.4020 e(z) = 9.8440
z = 2.758314+14.451968i, e(z) = 1.0548 - 0.0412i
z = 1.776063+20.348036i, e(z) = 0.9585 + 0.1994i
z = 1.386667+25.644800i, e(z) = 0.7875 - 0.2597i

Xi Critical Points
z = 0.5, x(z)= 0.5
z = 0.5 + 15.444i, x(z)= -0.0009
z = 0.5 + 21.918i, x(z)= 0

Zeta Zeros A short list of some zeta zeros mentioned in the dark heart research, which can be copied and pasted into the text fields are as follows:

1-12: 14.13472514, 21.02203964, 25.01085758, 30.42487613, 32.93506159, 37.58617816, 40.91871901, 43.32707328, 48.00515088, 49.77383248, 52.97032148, 56.4462477

125-130: 278.2507435, 279.2292509, 282.4651148, 283.2111857, 284.835964, 286.6674454

287-293: 523.9605309, 525.0773857, 527.9036416, 528.4062139, 529.8062263, 530.8669179, 532.688183

171382-171390: 121412.139210209, 121412.990421458, 121414.488895067, 121414.739043607, 121415.047364581, 121415.640550747, 121416.302522095, 121416.823543637, 121417.618749154

For further zeros go to Odlyzko's comprehensive super-computer listing:
http://www.dtc.umn.edu/~odlyzko/zeta_tables/index.html

The number of function terms can also be adjusted to large exact values to show up anomalous behavior, for example the 84270 term zeta product up to the prime 1079999, has a value at the first zeta zero of 6.25, well above 0. That for the 1518898th prime 24199999 is a stunning 20067241.5306, anything but 0!

```
//zerobounce10.c Generates a movie parameter file set of zeta zeros of
//{1,0,c,0,0,0,-c,0,-1,0} c=exp(2*pi*I*param)
//to be processed with RZViewer
//To run this file from terminal put it in a folder, cd to the folder and input
//gcc -o rzmovieo zerobounce10.c -lm && ./rzmovieo
//To run successive compiled versions type ./rzmovieo
//See: http://www.dhushara.com/DarkHeart/RZV/RZViewer.htm
#include <stdio.h>
```

```
#include<complex.h>
const double pi=3.1415926535;
const int frames = 5; // use n+1 frames for circular parameter n-fold symmetry
const int spcno = 29+2+2*10;
const char base[] = "Movie";
void WriteParamFile(int frameno, double param);
int main(void)
{
FILE *mf;
char filename [ FILENAME_MAX ];
int filetyp = 3;
int k;
double xstt = 0.25;
double xend = 1.25;
double xpos;
sprintf(filename, "%s.txt", base);
mf=fopen(filename,"w");
fprintf(mf,"%f %f \n",(float)filetyp, (float)frames);
fclose(mf);
xpos=xstt;
for(k=0;k<frames;k++)
{
if(k==0)
xpos=xstt;
else
xpos=xstt+(xend-xstt)*k/(frames-1);
WriteParamFile(k, (double)xpos);
}
return 0;
}
void WriteParamFile(int frameno, double param)
{
FILE *mf;
char filename [ FILENAME_MAX ];
int j;
double specs[spcno];
double mandel=2;
double scale=15;
double screenX=24;
double screenY=0;
double cx=0;
double cy=0;
double altgamma=0;
double thisfn=18;
double split=1;
double orbitTrapE=0;
double addc=1;
double escapeonly=0;
double colsch=0;
double zetaterms=100;
double escapeboundExp=1.7;
double attractorEscBd=-2;
double maxIterations=64;
double dolog=0;
double sang=1;
double domoib=0;
double cval=0;
double oldfn=thisfn;
double realk=0;
double imagk=0;
double crval=0;
double fullt=0;
double xm=0;
double xn=2;
```

```
double flip=1;
specs[0]=mandel;
specs[1]=scale;
specs[2]=screenX;
specs[3]=screenY;
specs[4]=cx;
specs[5]=cy;
specs[6]=altgamma;
specs[7]=thisfn;
specs[8]=split;
specs[9]=orbitTrapE; //orbit trap epsilon
specs[10]=addc; //additive c/ mult c
specs[11]=escapeonly;
specs[12]=colsch;
specs[13]=escapeboundExp; //escape bound exponent
specs[14]=zetaterms;
specs[15]=attractorEscBd; //attractor escape bound 10^-k
specs[16]=maxIterations; //iterations
specs[17]=dolog; //take log of the function
specs[18]=sang; //angle ploting
specs[19]=domoib; //mobius
specs[20]=cval;
specs[21]=oldfn;
specs[22]=realk;
specs[23]=imagk;
specs[24]=crval;
specs[25]=fullt;
specs[26]=xm;
specs[27]=xn;
specs[28]=flip;
specs[29]=10;
specs[30]=0;
specs[31]=1; //complex characters
specs[32]=0;
specs[33]=0;
specs[34]=0;
specs[35]=creal(cexp(2*pi*I*param));
specs[36]=cimag(cexp(2*pi*I*param));
specs[37]=0;
specs[38]=0;
specs[39]=0;
specs[40]=0;
specs[41]=0;
specs[42]=0;
specs[43]=-creal(cexp(2*pi*I*param));
specs[44]=-cimag(cexp(2*pi*I*param));
specs[45]=0;
specs[46]=0;
specs[47]=-1;
specs[48]=0;
specs[49]=0;
specs[50]=0;

sprintf(filename, "%s%d.txt", base, frameno);
//printf("filename = \"%s\"\n", filename);
mf=fopen(filename,"w");
for (j=0;j<spcno;j++)
fprintf(mf,"%f ",specs[j]);
fclose(mf);
}
```