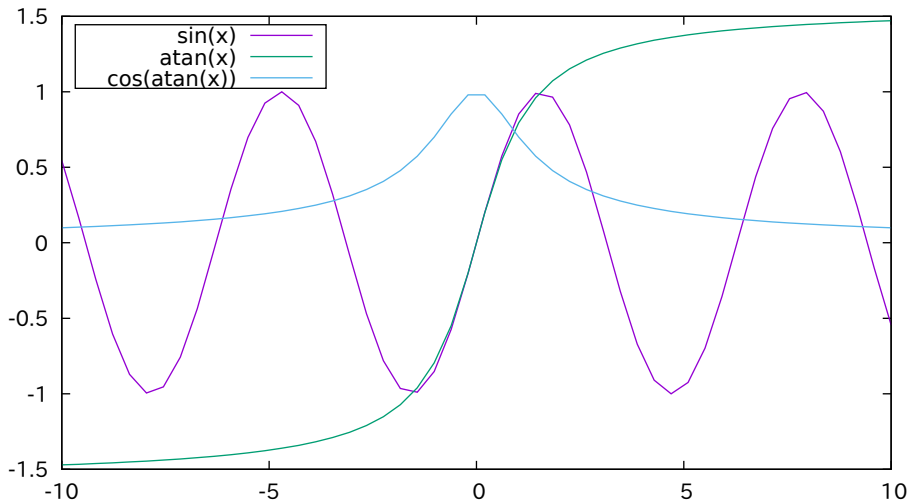
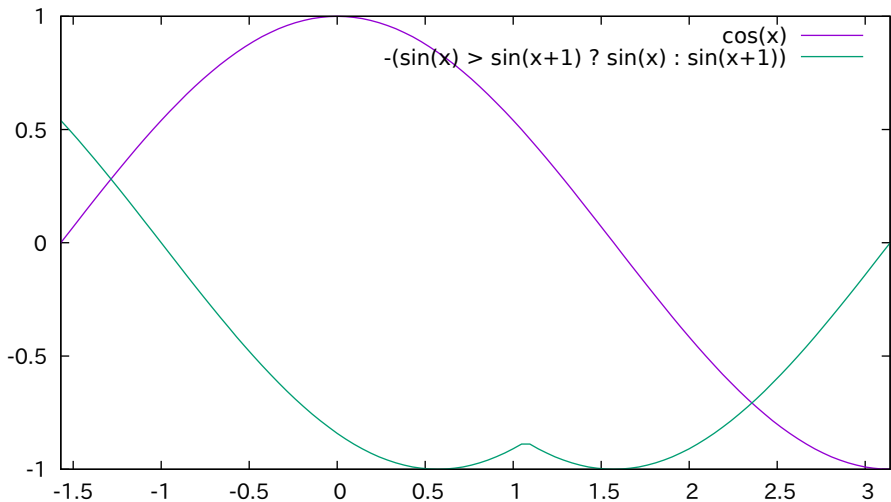


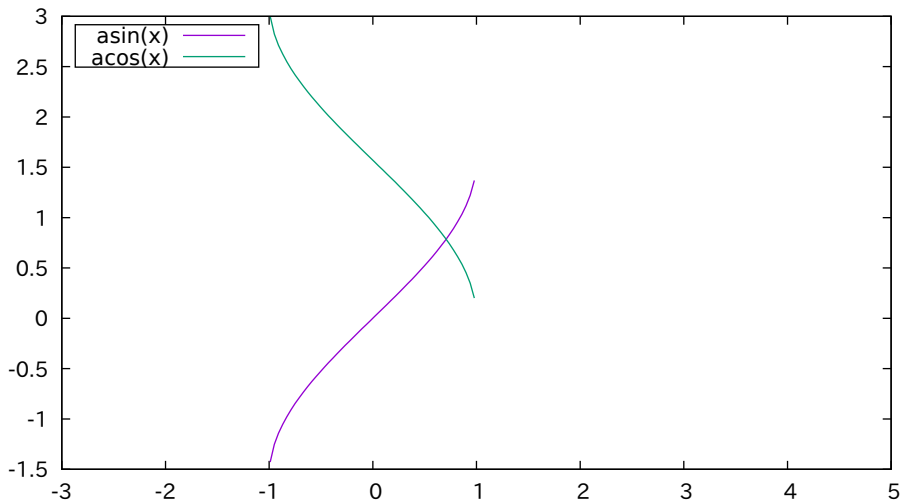
Simple Plots



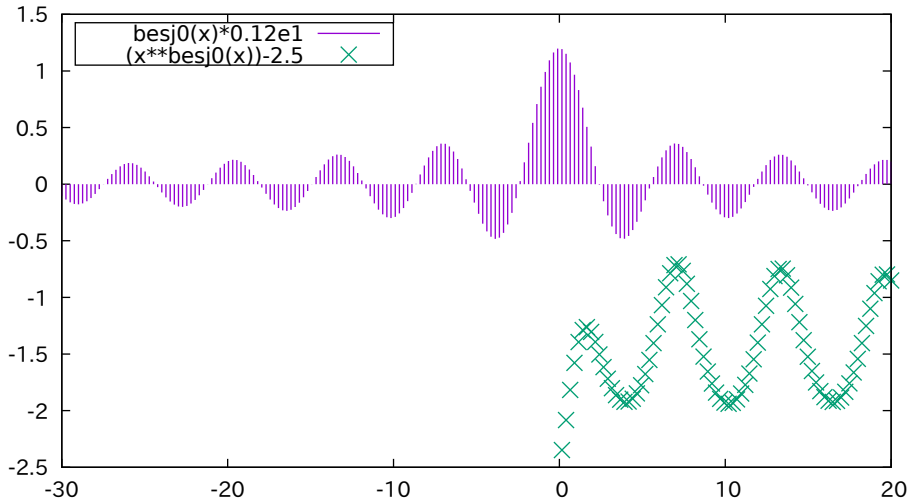
Simple Plots



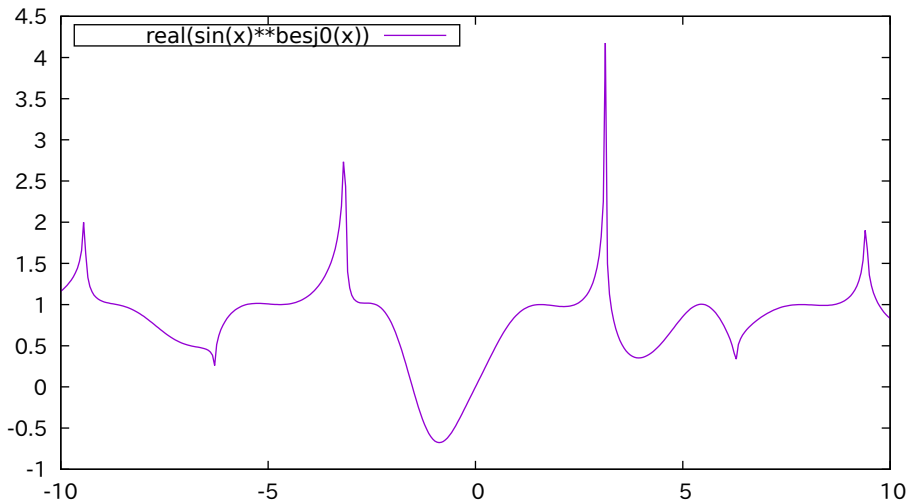
Simple Plots



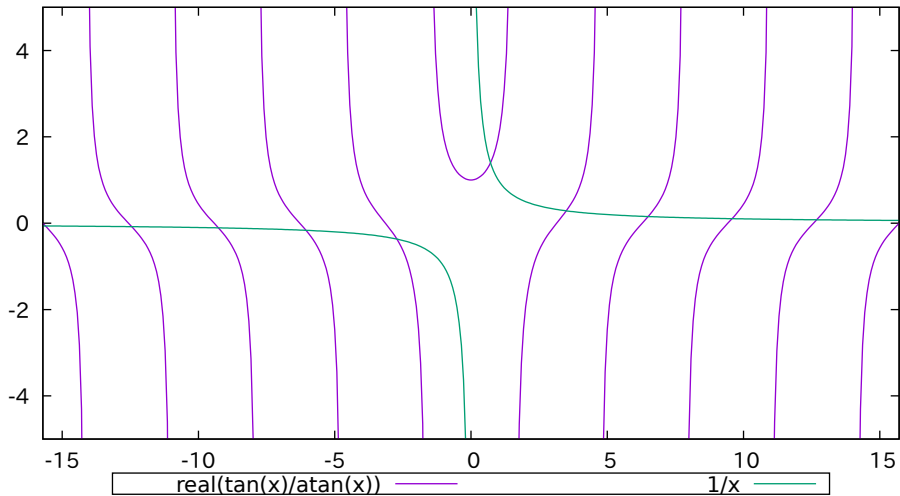
Simple Plots



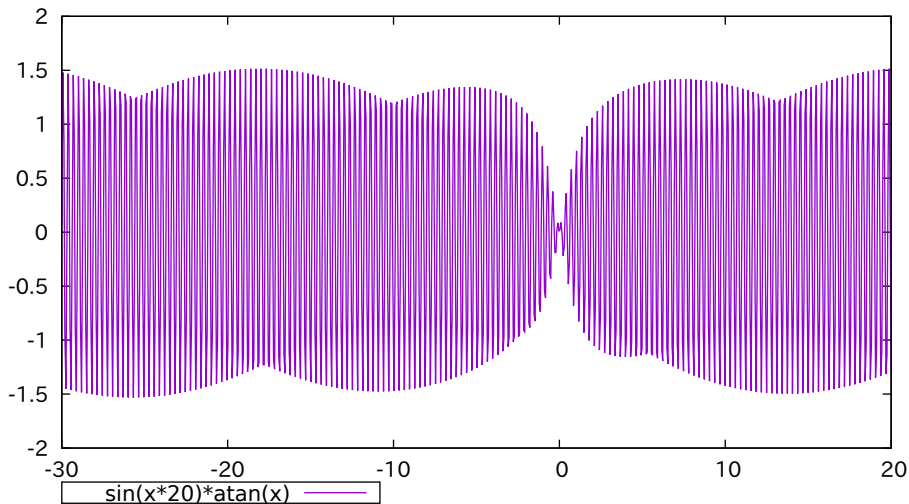
Simple Plots



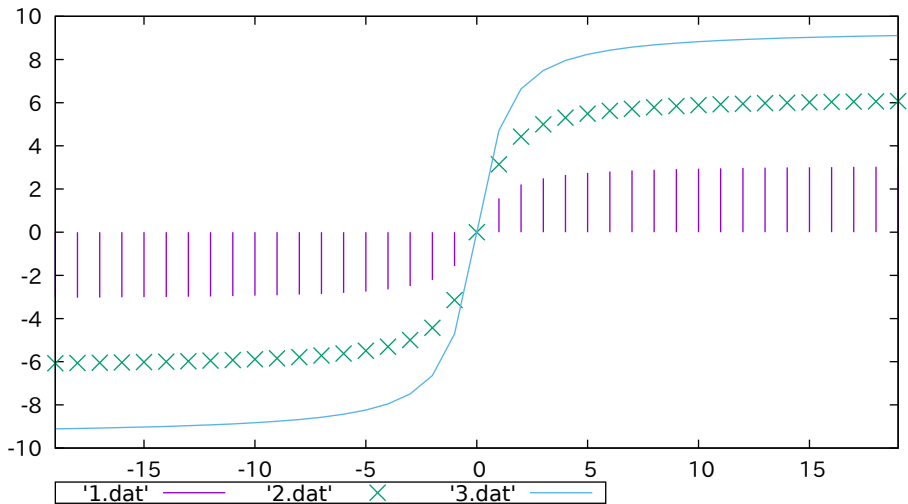
Simple Plots

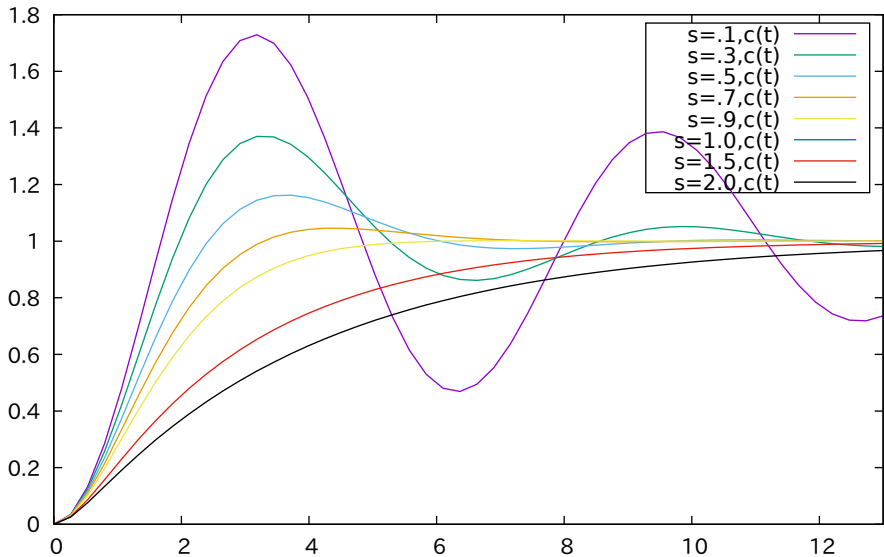


Simple Plots

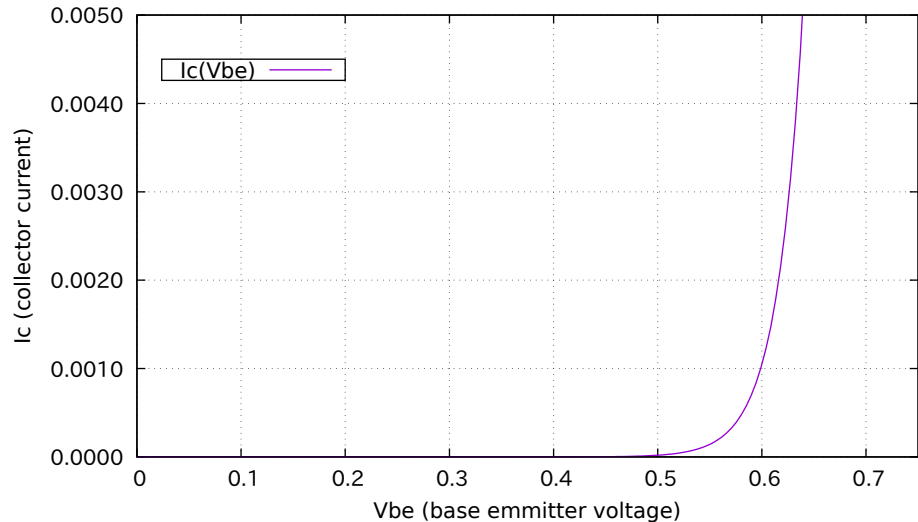


Simple Plots

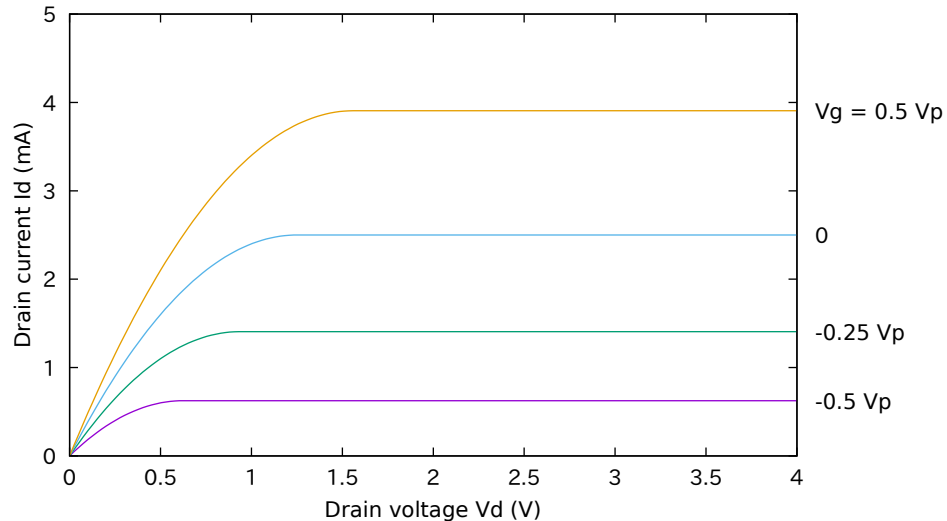




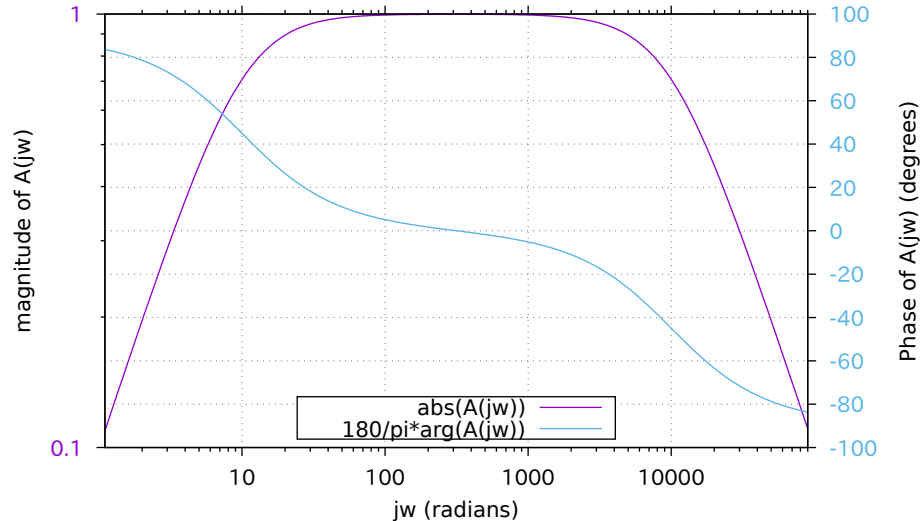
Mutual Characteristic of a Transistor



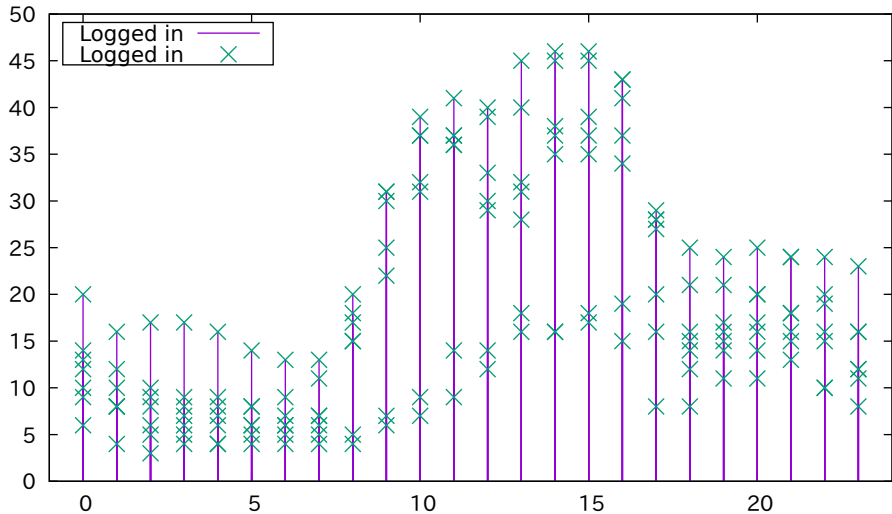
JFET Mutual Characteristic



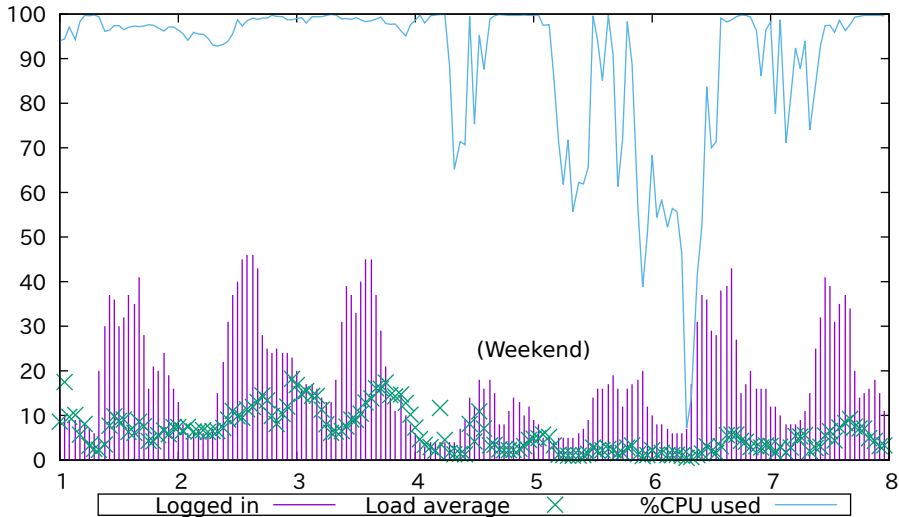
Amplitude and Phase Frequency Response



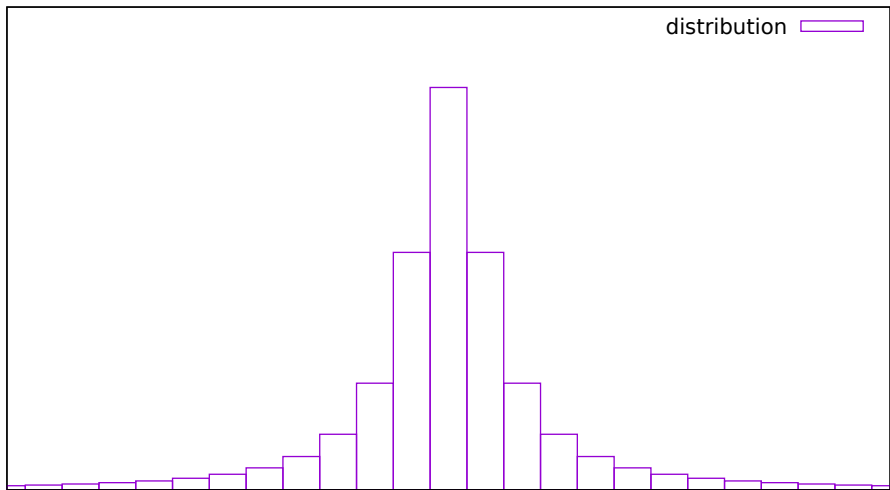
Convex November 1-7 1989 Circadian



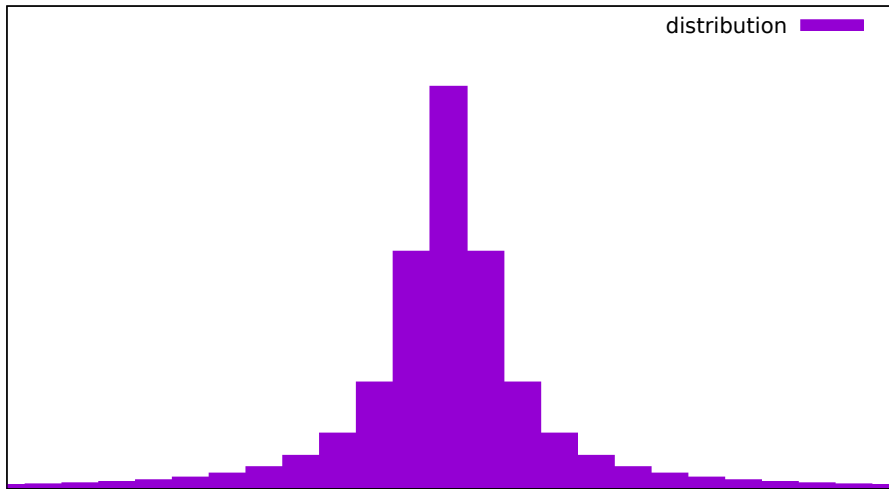
Convex November 1-7 1989



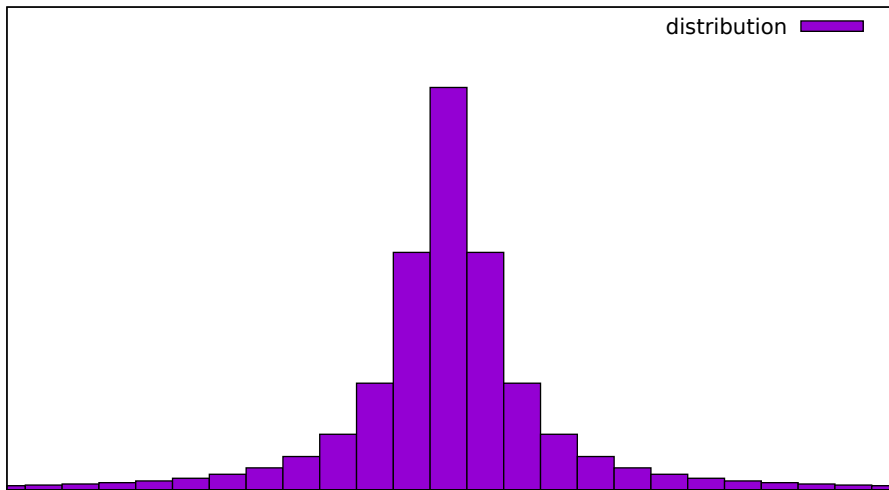
A demonstration of boxes with default properties



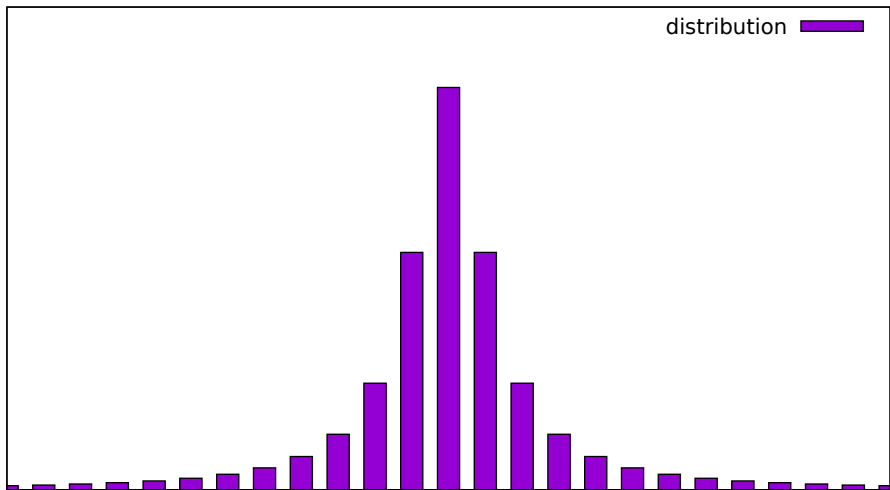
A demonstration of boxes with style fill solid 1.0



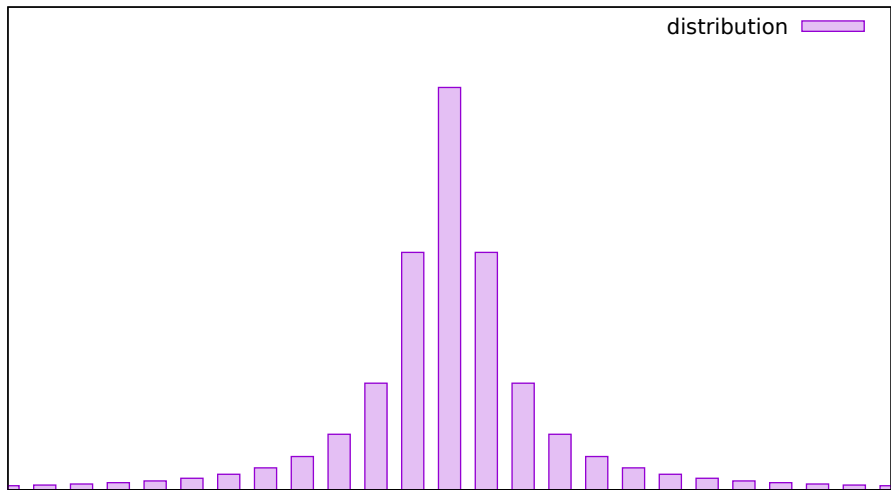
A demonstration of boxes with style fill solid border -1



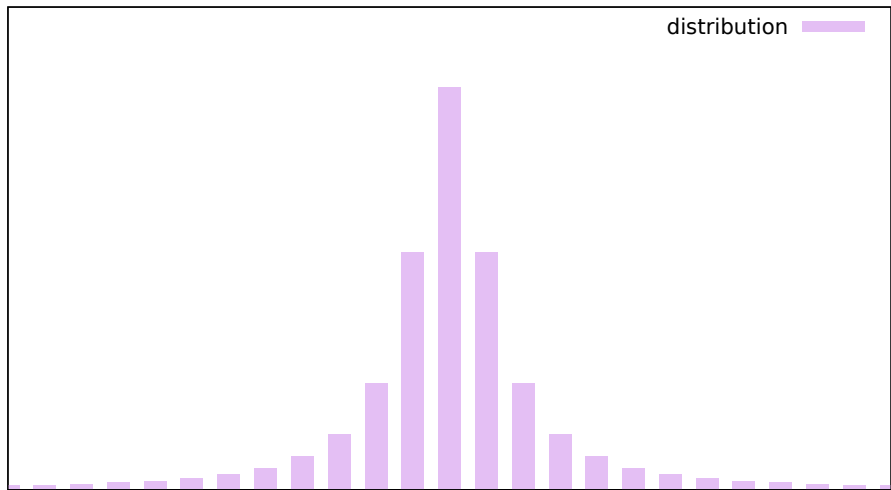
Filled boxes of reduced width



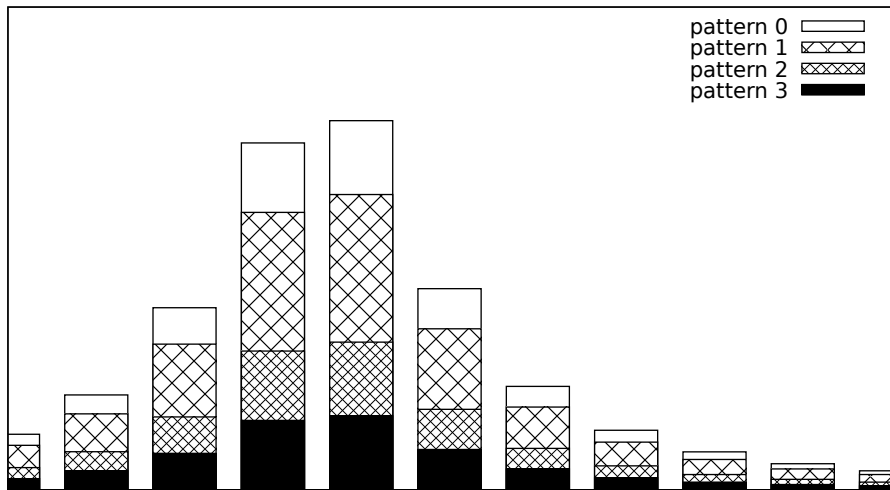
Filled boxes at 50% fill density



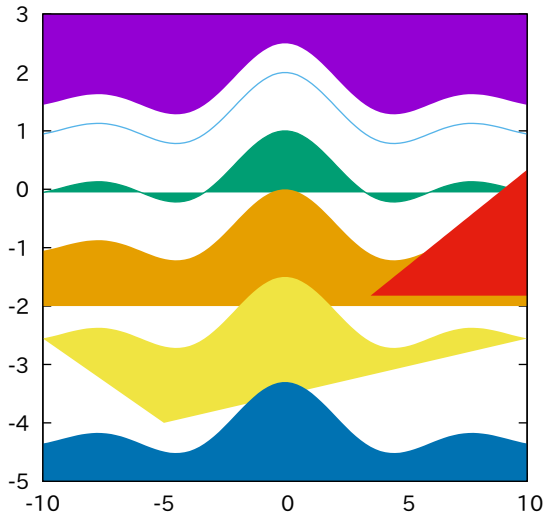
A demonstration of boxes with style fill solid 0.25 noborder



A demonstration of boxes in mono with style fill pattern

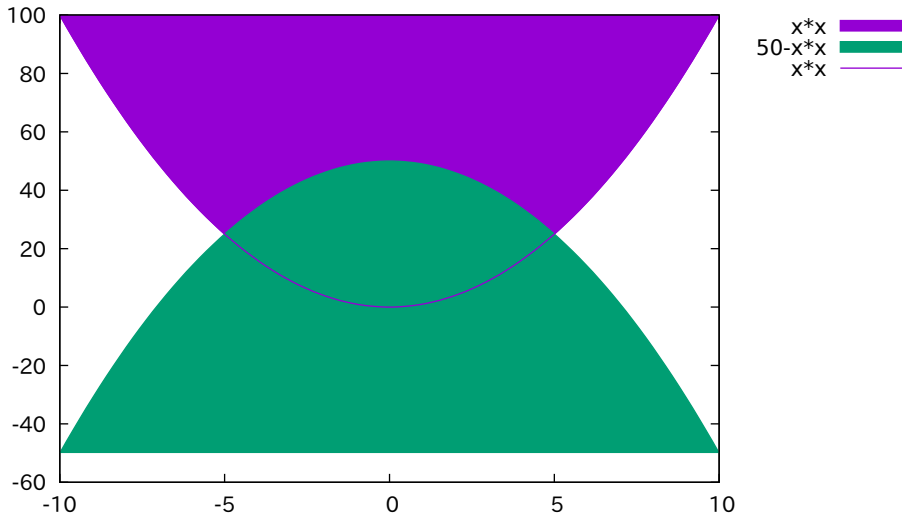


plot with filledcurve [options]

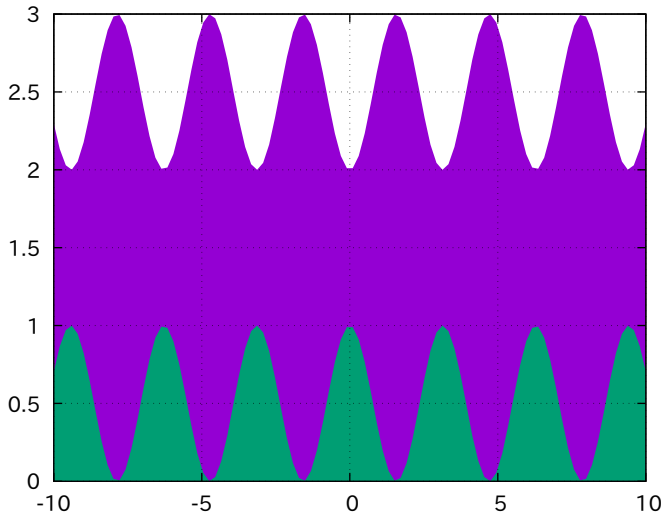


$1.5 + \sin(x)/x$ 
 $\sin(x)/x$ 
 $1 + \sin(x)/x$ 
 $-1 + \sin(x)/x$ 
 $-2.5 + \sin(x)/x$ 
 $-4.3 + \sin(x)/x$ 
 $(x > 3.5 ? x/3 - 3 : 1/0)$ 

Intersection of two parabolas

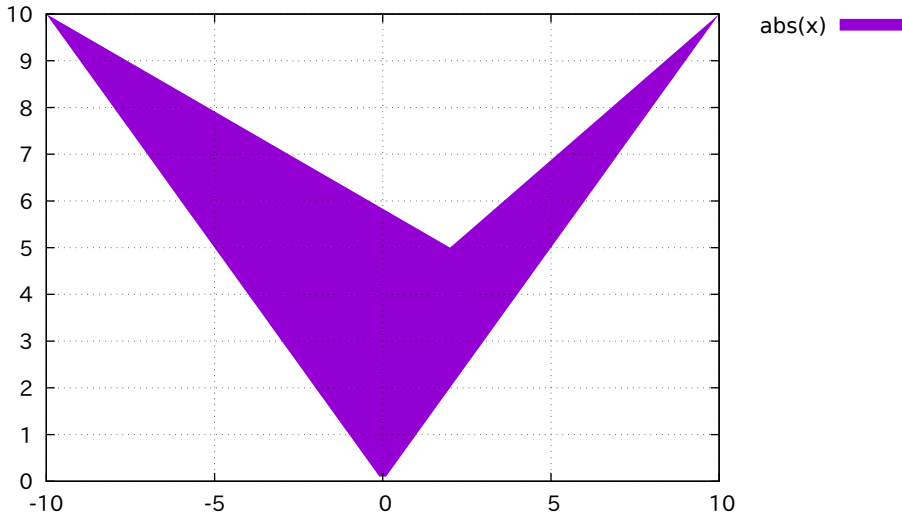


Filled sinus and cosinus curves

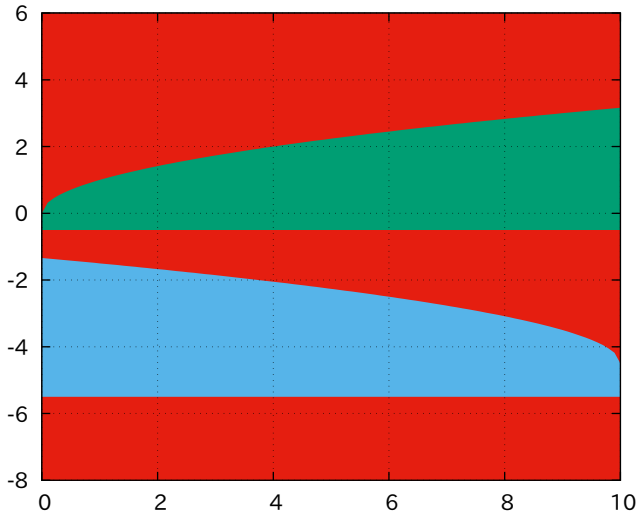


$2 + \sin(x)^2$ 
 $\cos(x)^2$ 

The red bat: $\text{abs}(x)$ with filledcurve $xy=2,5$

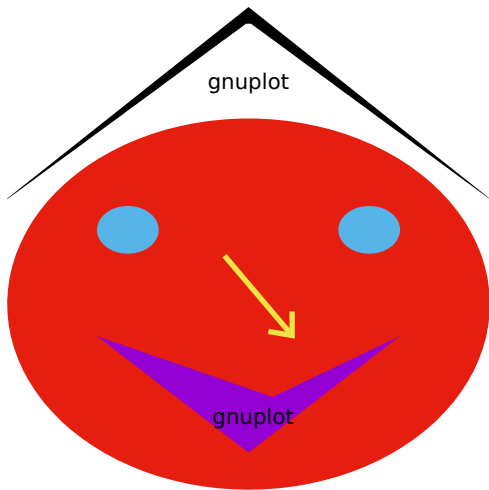


Some sqrt stripes on filled graph background

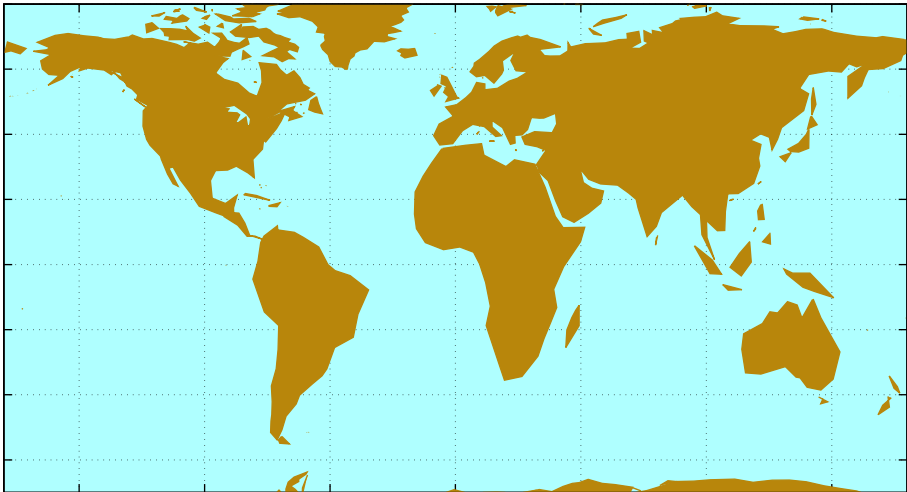


-8
 $\sqrt{x}$
 $\sqrt{10-x}-4.5$

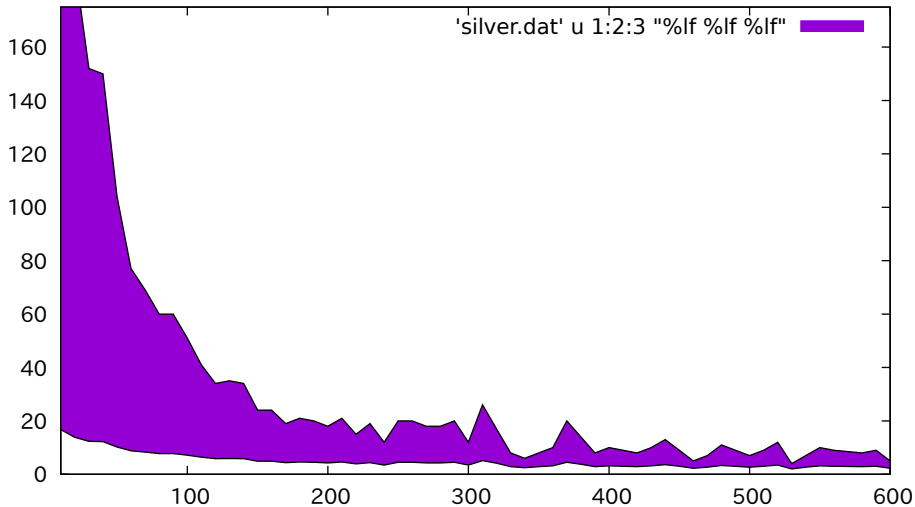
Let's smile with parametric filled curves



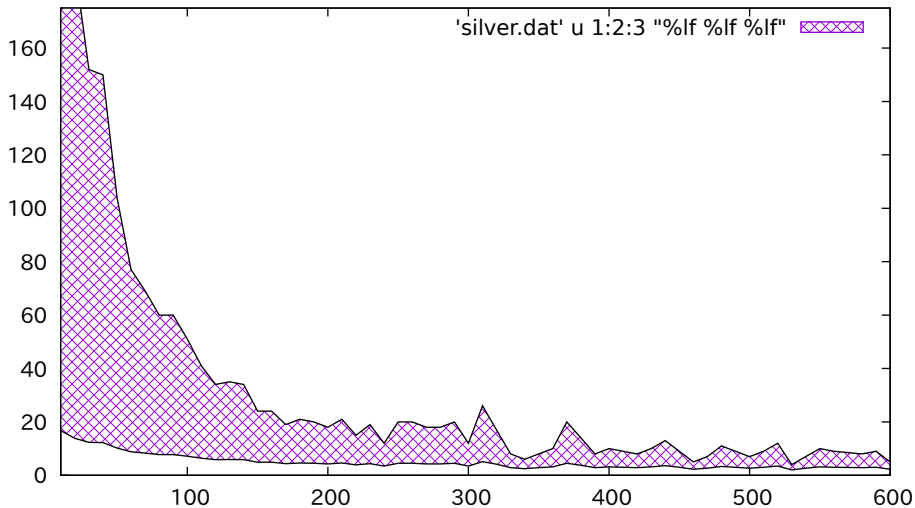
world.dat plotted with filledcurves



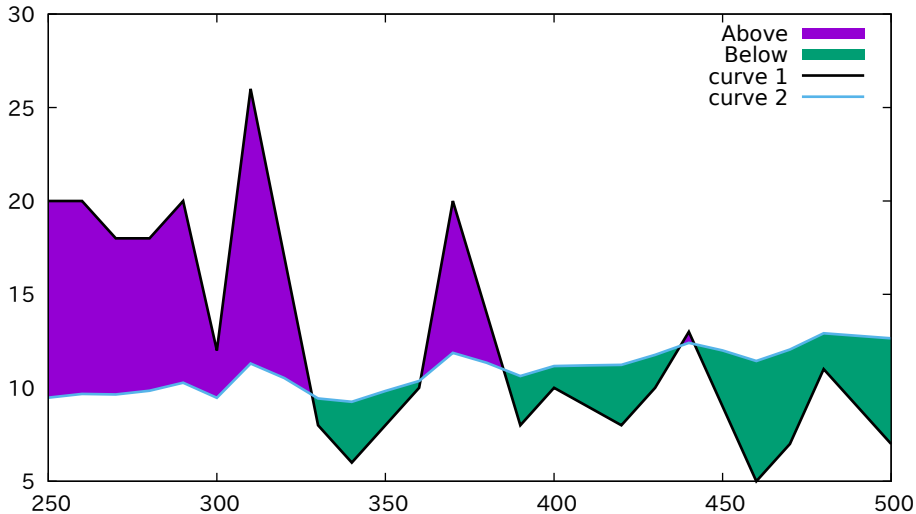
Fill area between two curves



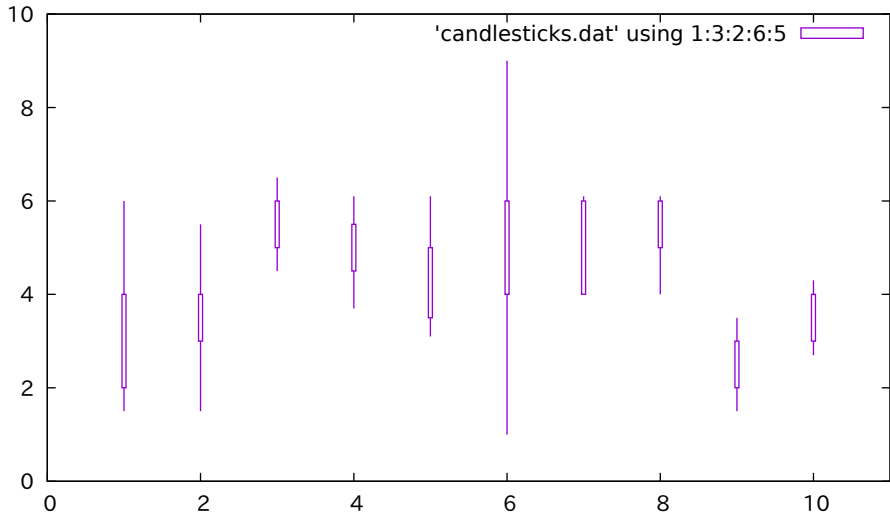
Fill area between two curves (pattern fill)



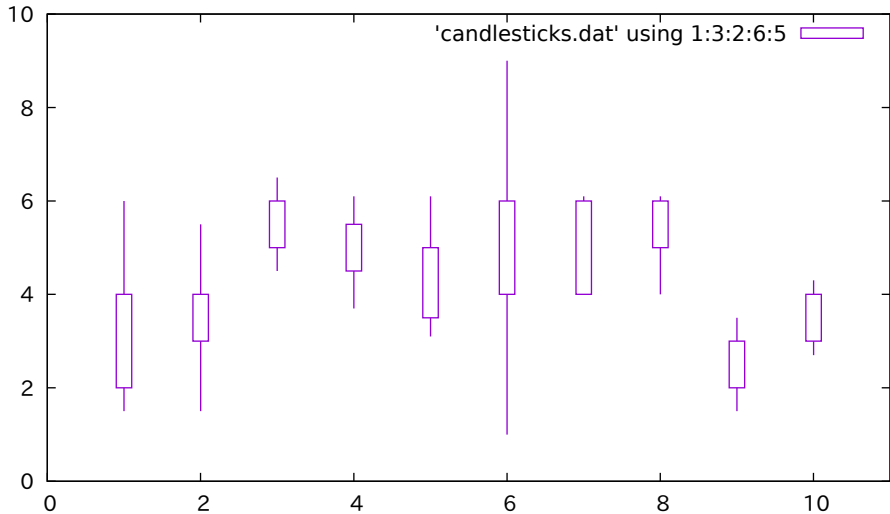
Fill area between two curves (above/below)



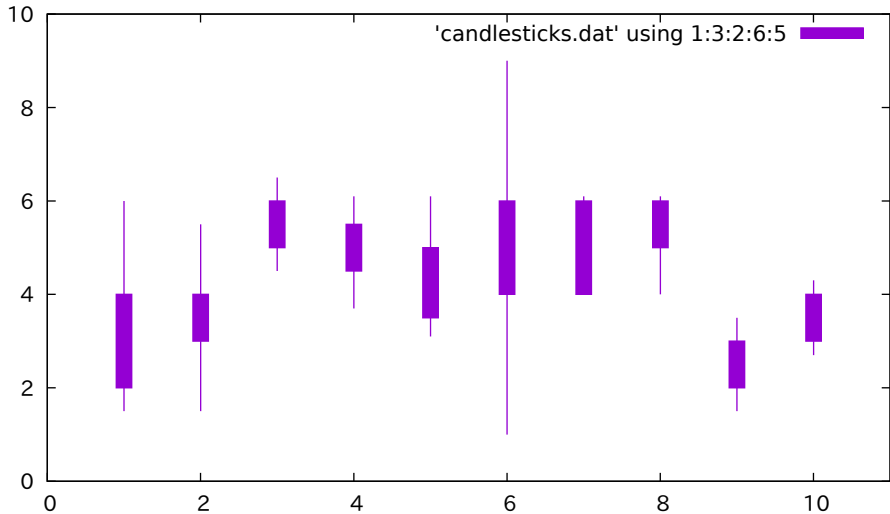
candlesticks with open boxes (default)



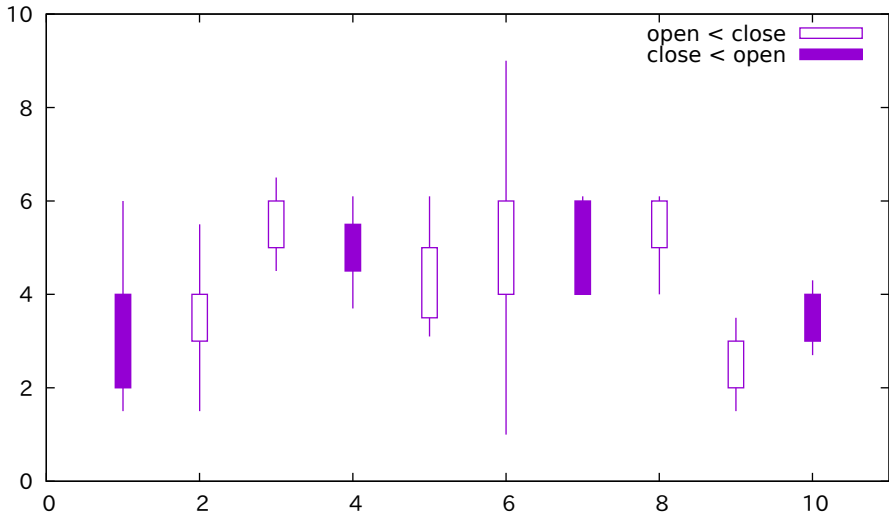
candlesticks with specified boxwidth



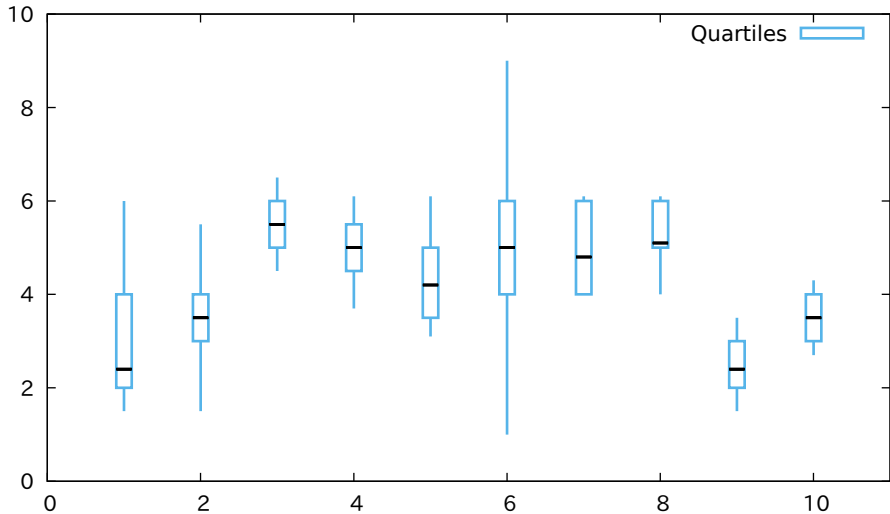
candlesticks with style fill solid



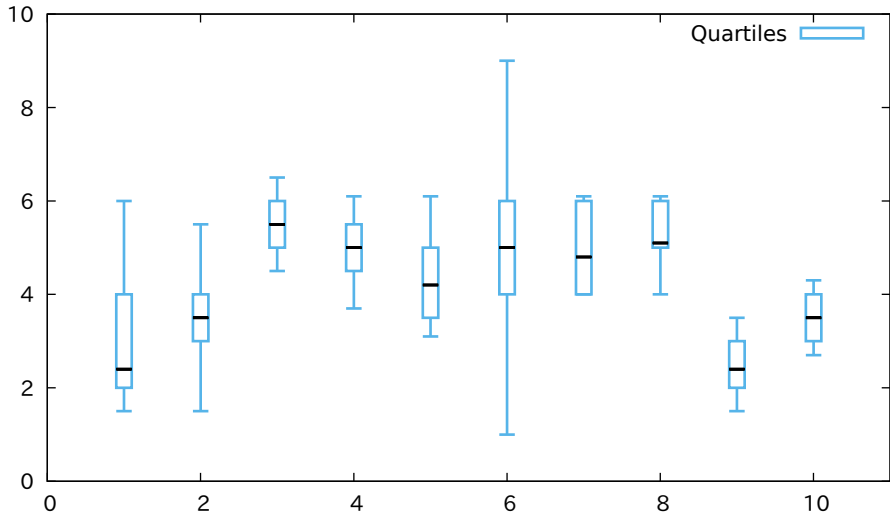
candlesticks showing both states of open/close



box-and-whisker plot adding median value as bar

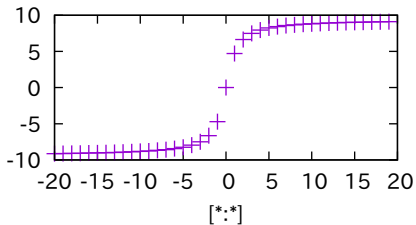


box-and-whisker with median bar and whiskerbars

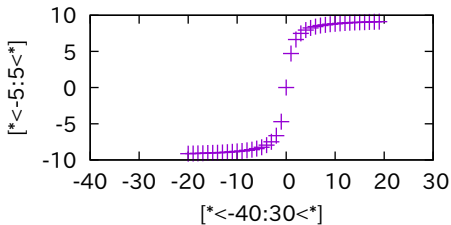


Autoscaling with constraints (y-axis always unaffected)

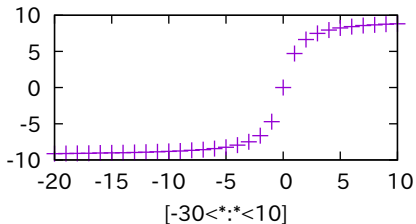
unconstrained



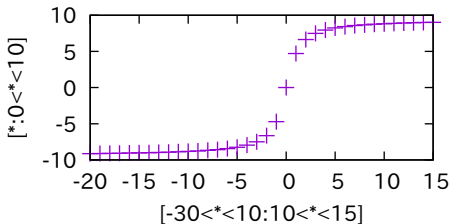
minimum range guaranteed



clip to maximum range

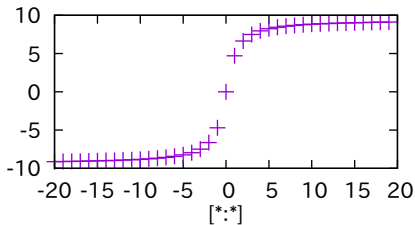


mixed

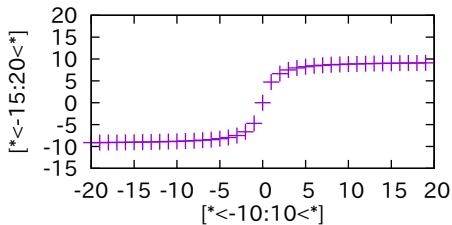


Autoscaling with constraints (x-axis always unaffected)

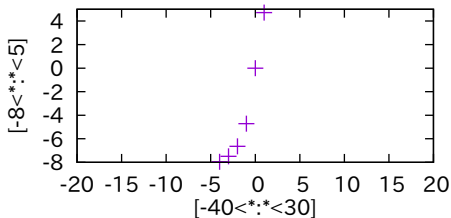
unconstrained



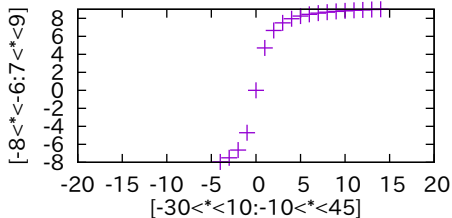
minimum range guaranteed



clip to maximum range

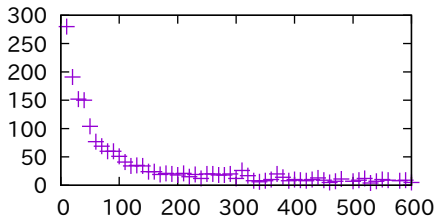


mixed

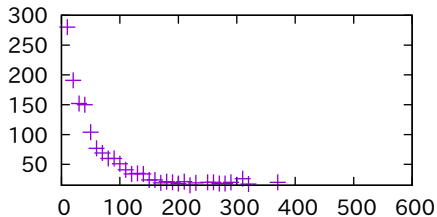


Autoscaling with constraints

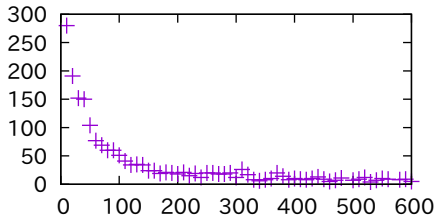
autoscale xy



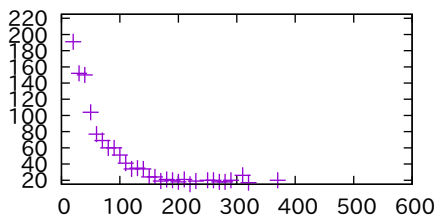
set yrange [15<*<25:*)]



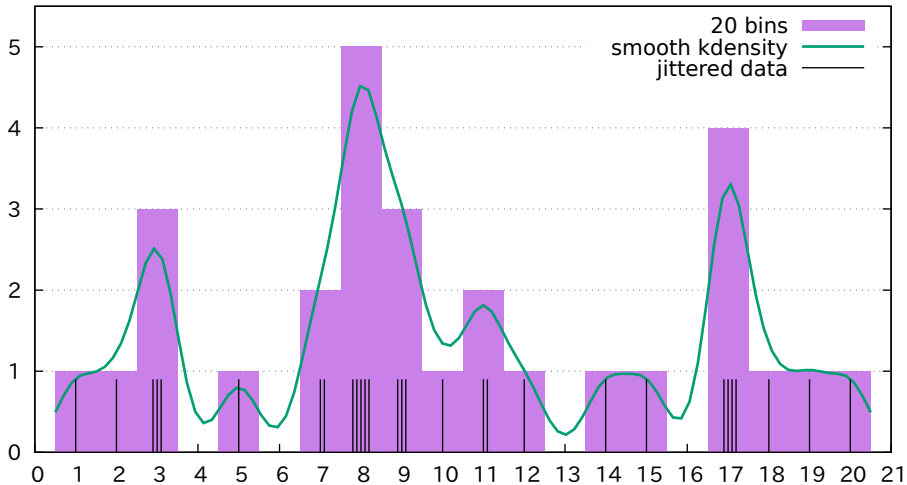
set autoscale ymin



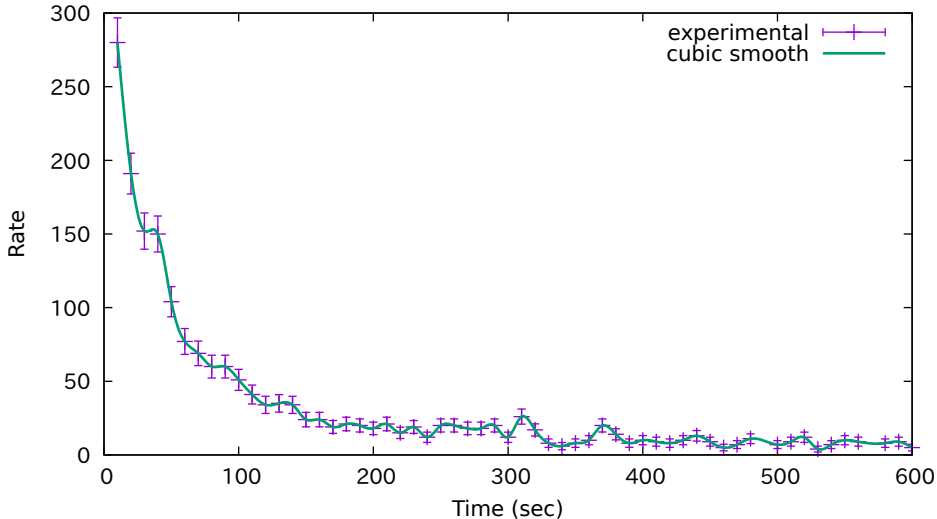
set yrange [15<*<25:135<*<225]



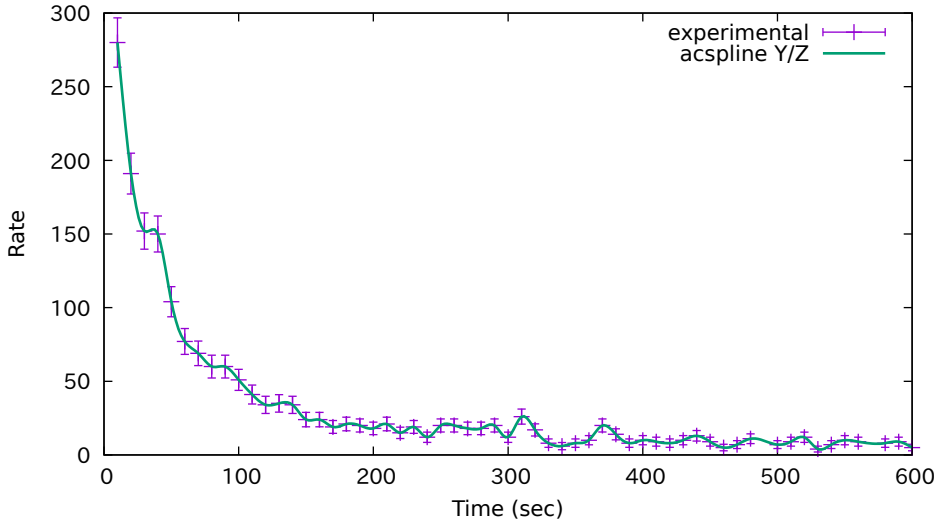
Comparison of a binned histogram and
a kernel density model of the same data



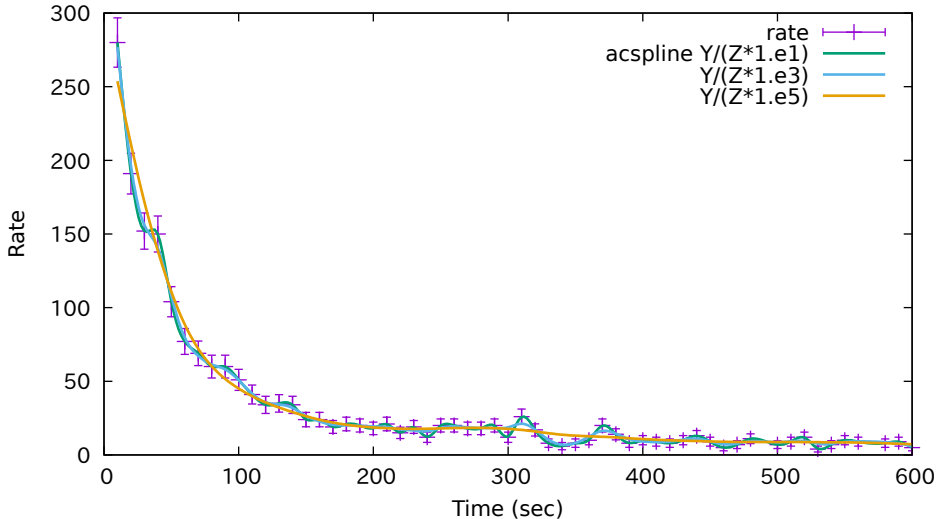
cubic spline fit to data (no weights)



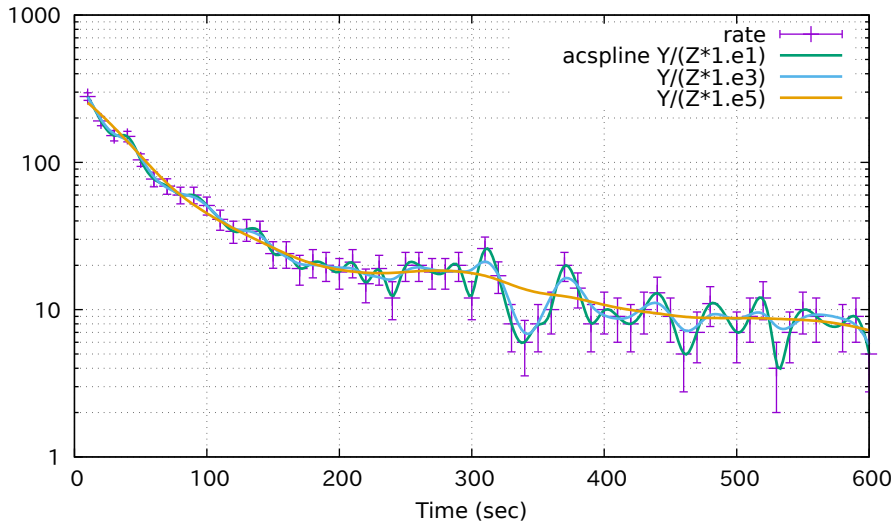
acsplines weighted by relative error



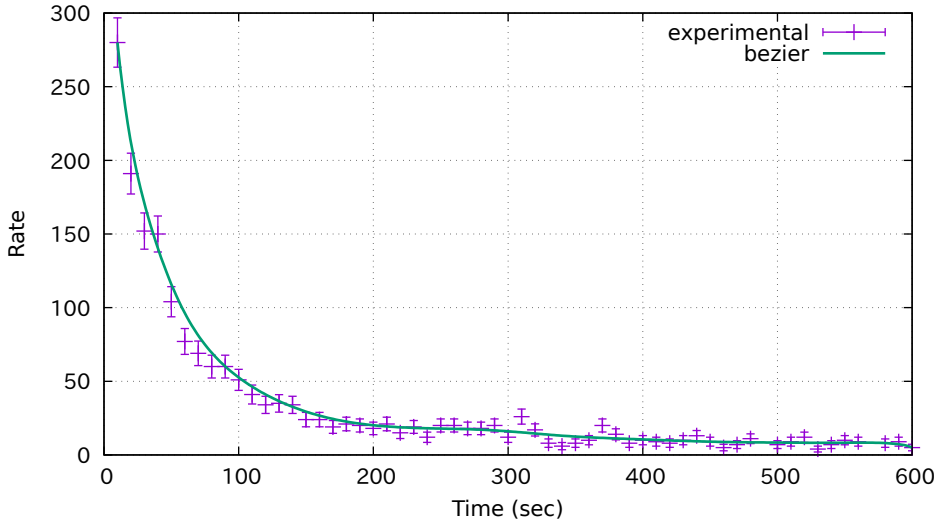
acsplines with increasing weight from error estimate



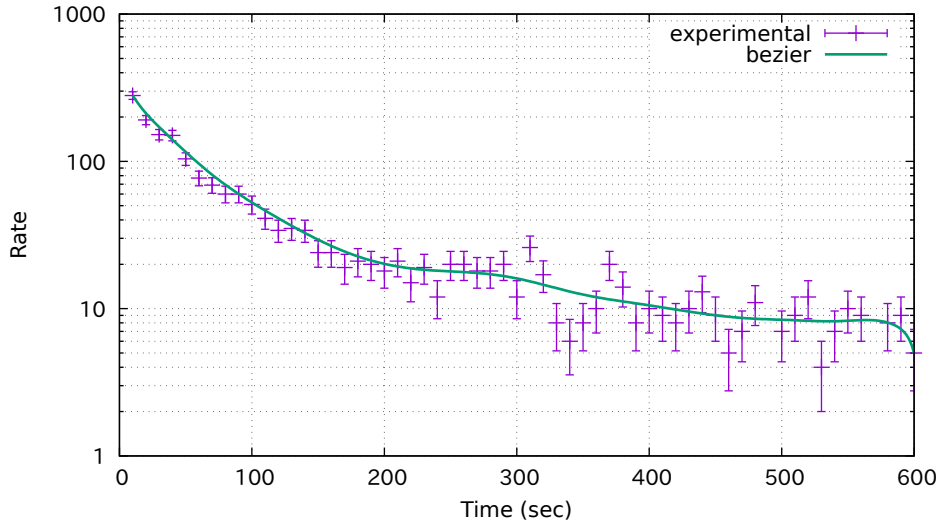
Same plot (various weighting) in log scale



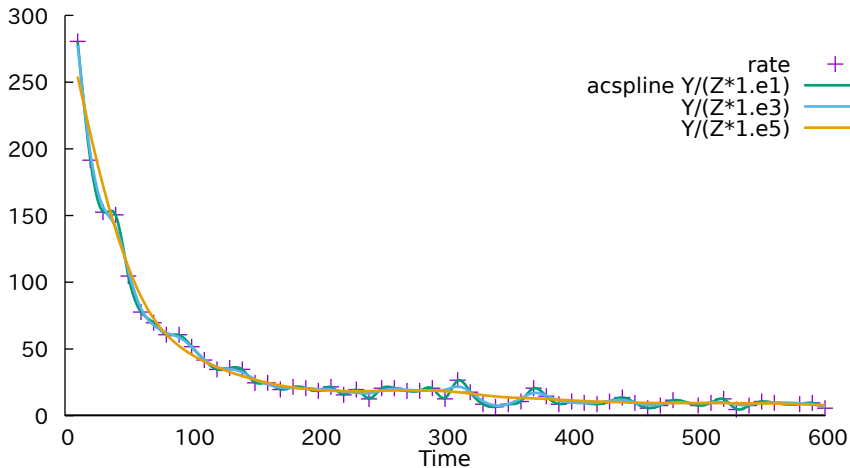
Bezier curve rather than cubic spline



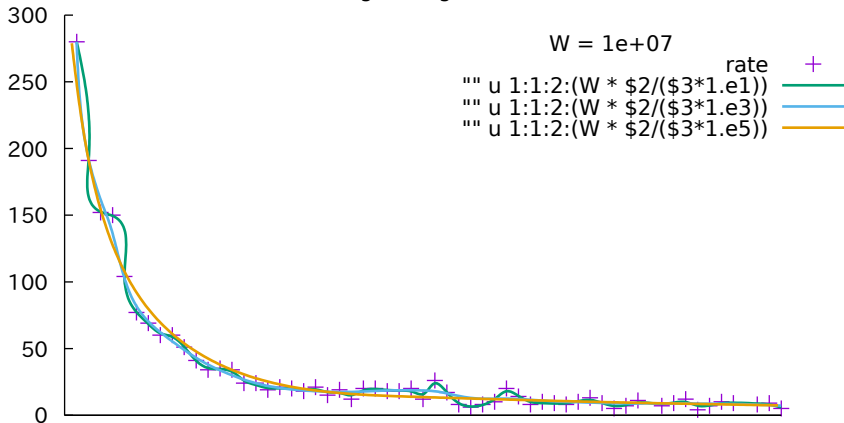
Bezier curve with log scale



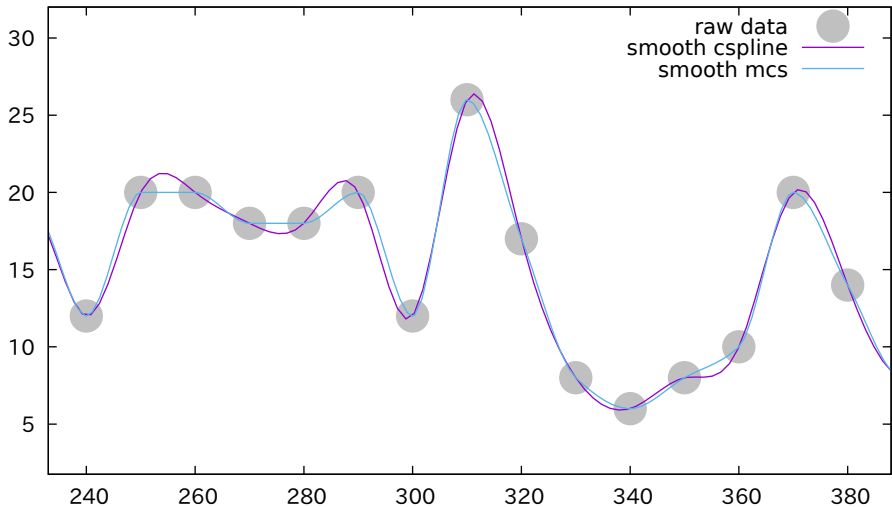
3D smooth acsplines (special case with curve in single plane)



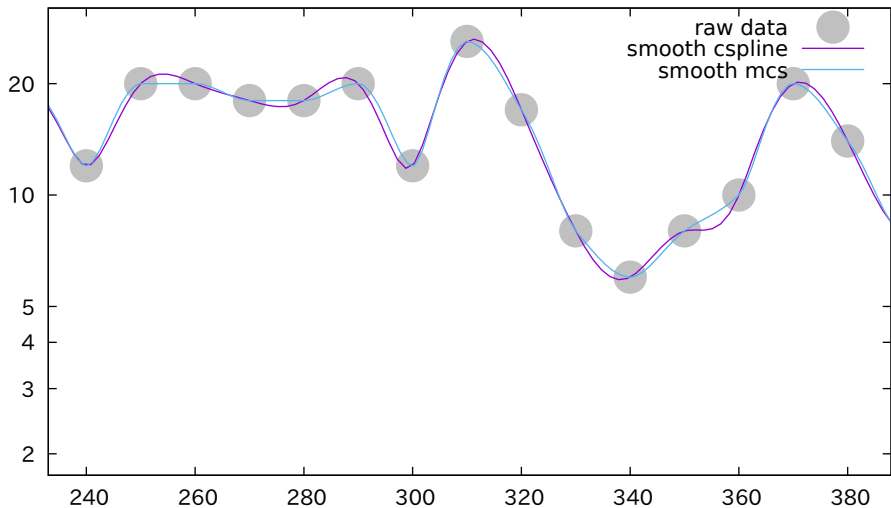
3D acsplines (general case)
Note much larger weight values needed



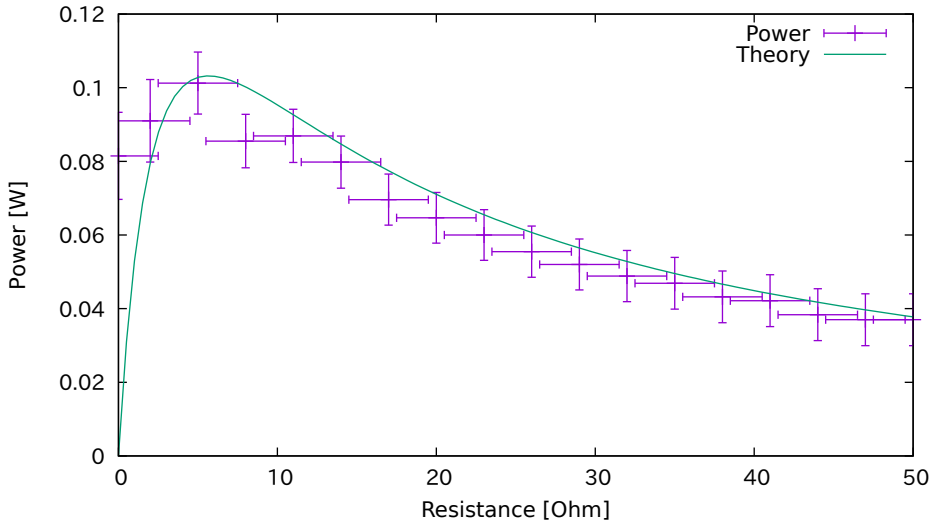
Monotonic cubic splines



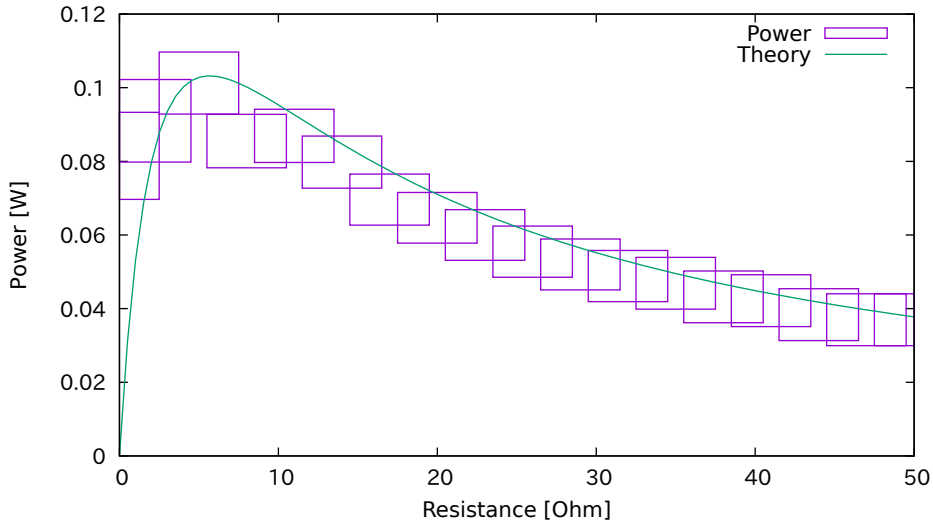
Monotonic cubic splines (log-scale data)



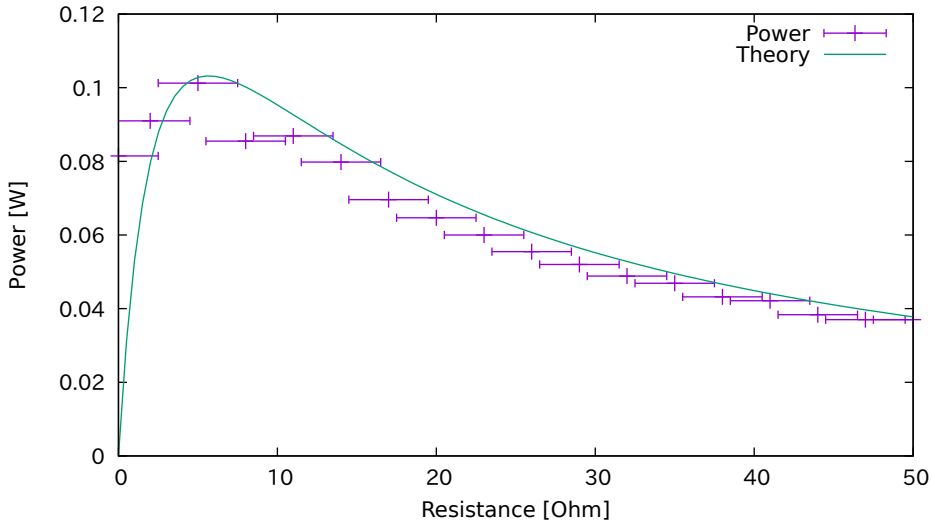
error represented by xyerrorbars



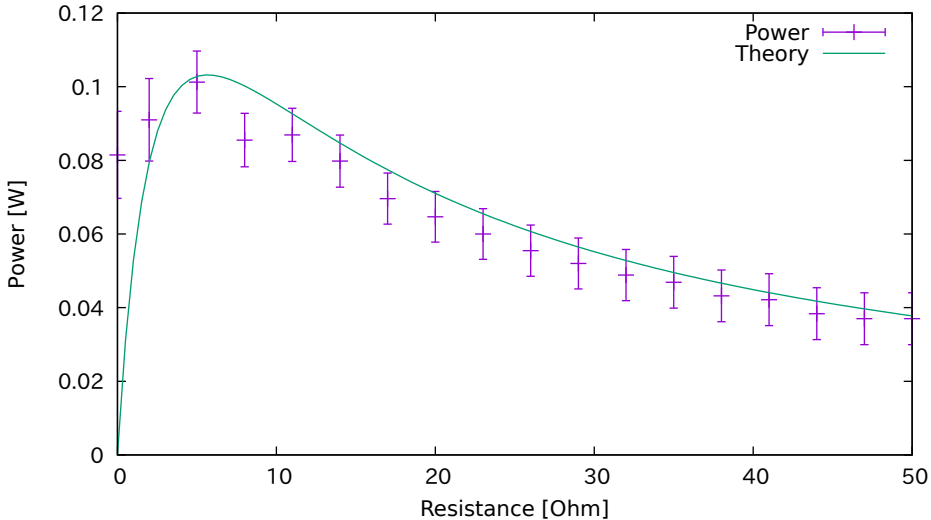
error represented by boxxyerror



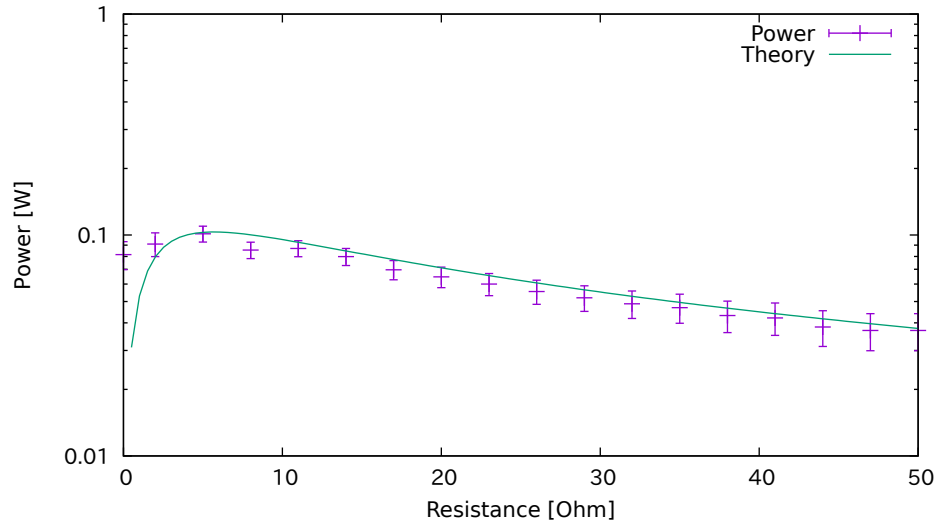
error represented by xerrorbars



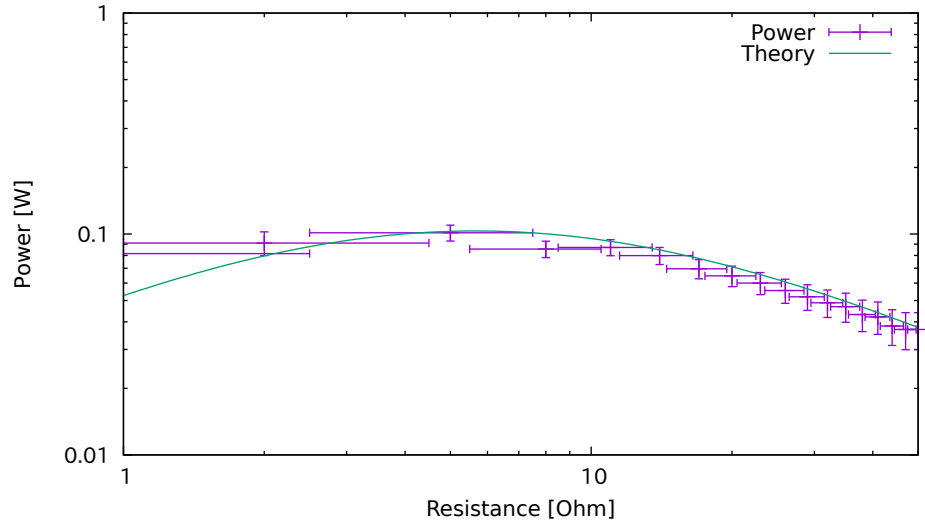
error represented by yerrorbars



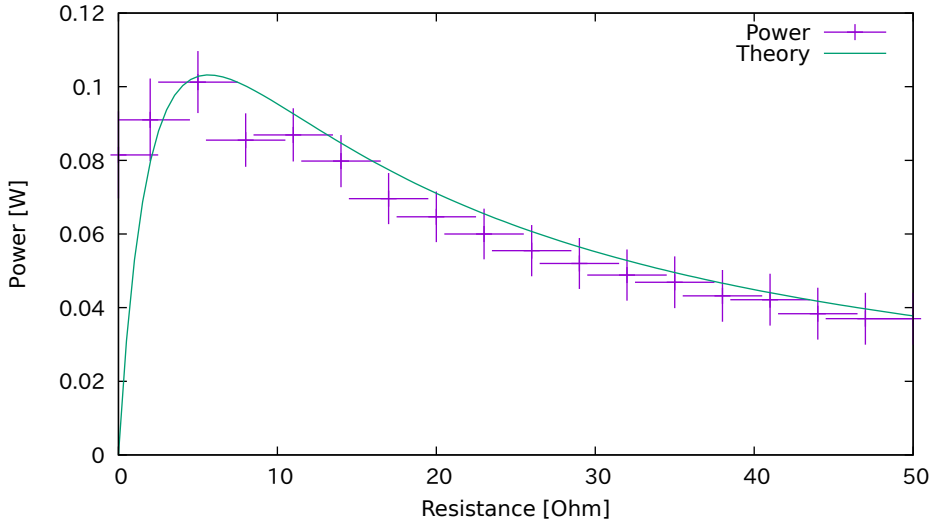
yerrorbars in log scale



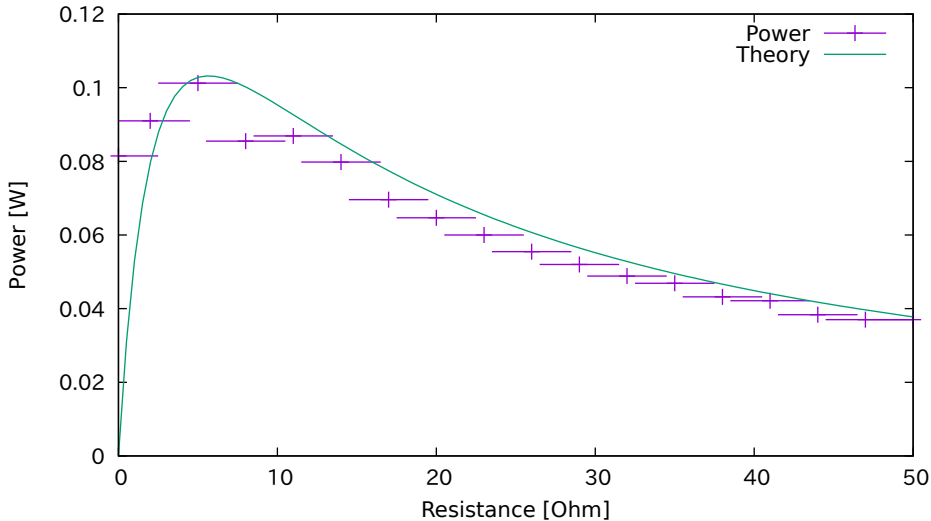
xyerrorbars in log scale



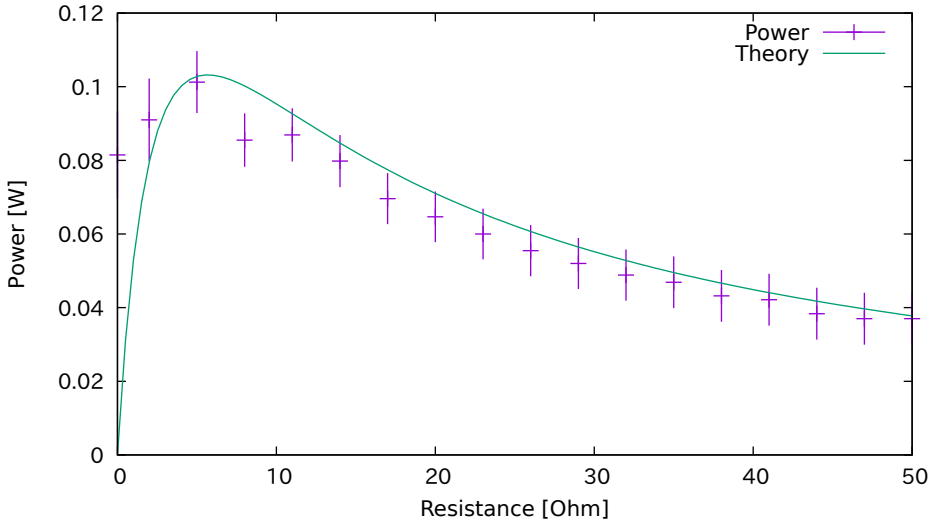
xyerrorbars with no crossbar



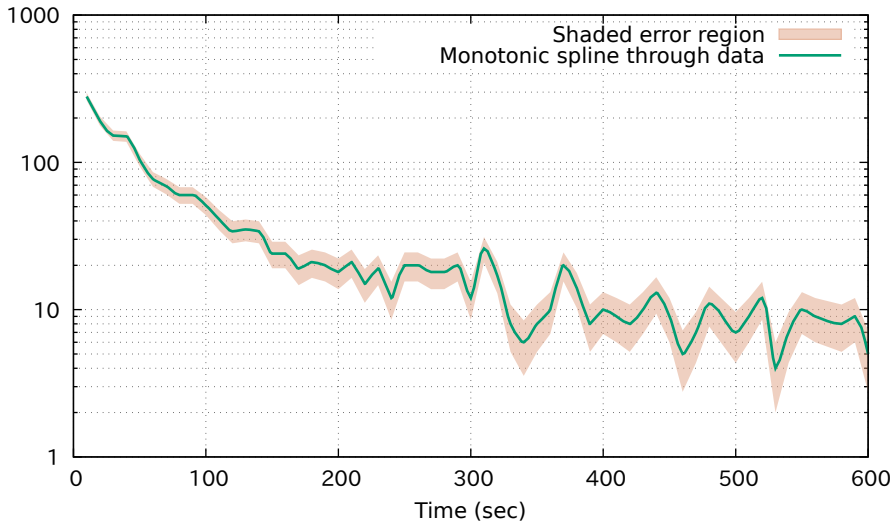
xerrorbars with no crossbar



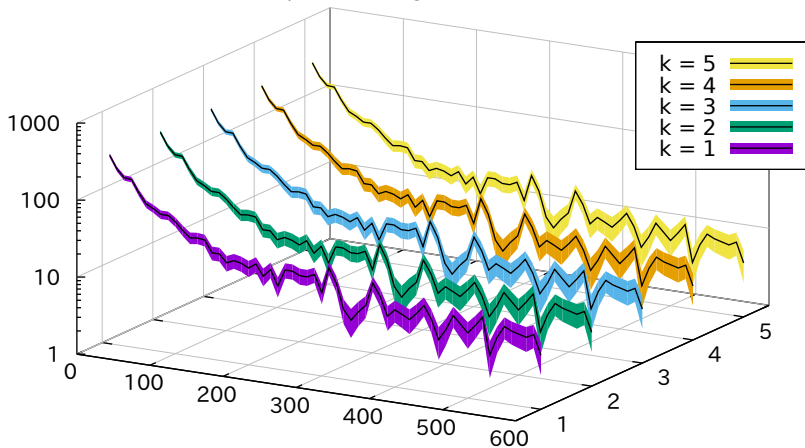
yerrorbars with no crossbar



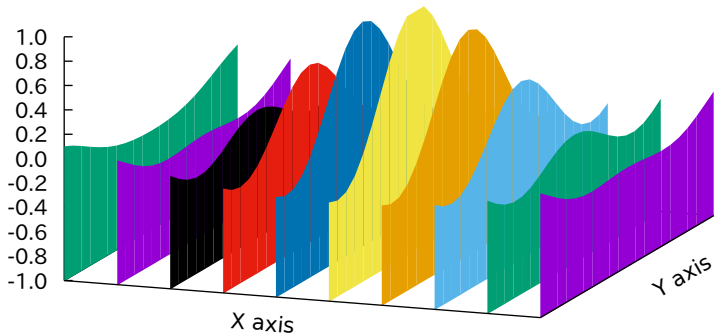
Error on y represented by filledcurve shaded region



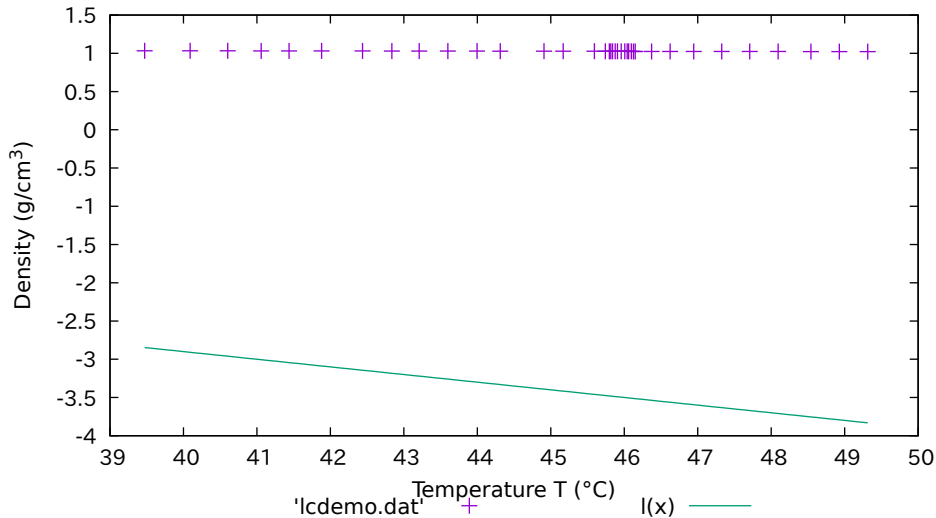
plot with zerrorfill
(note that plot ordering must be back-to-front)



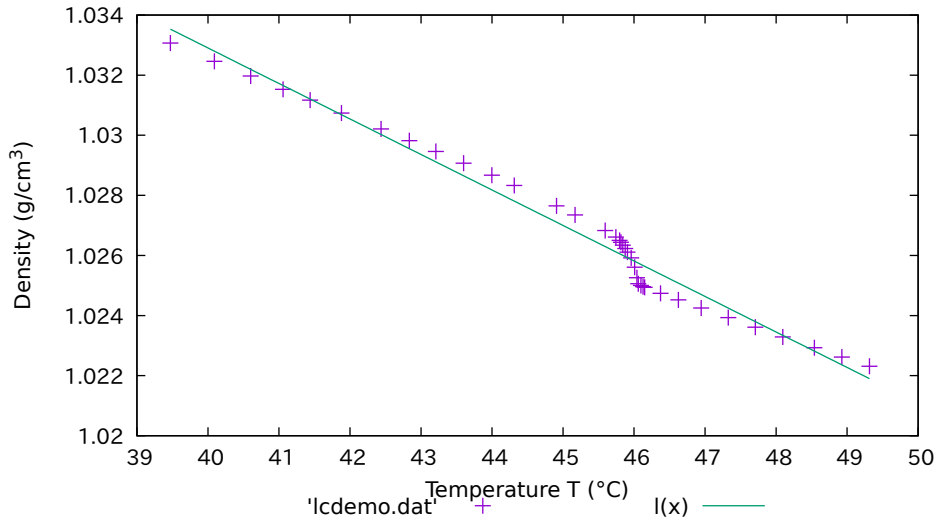
fence plot constructed with zerrorfill



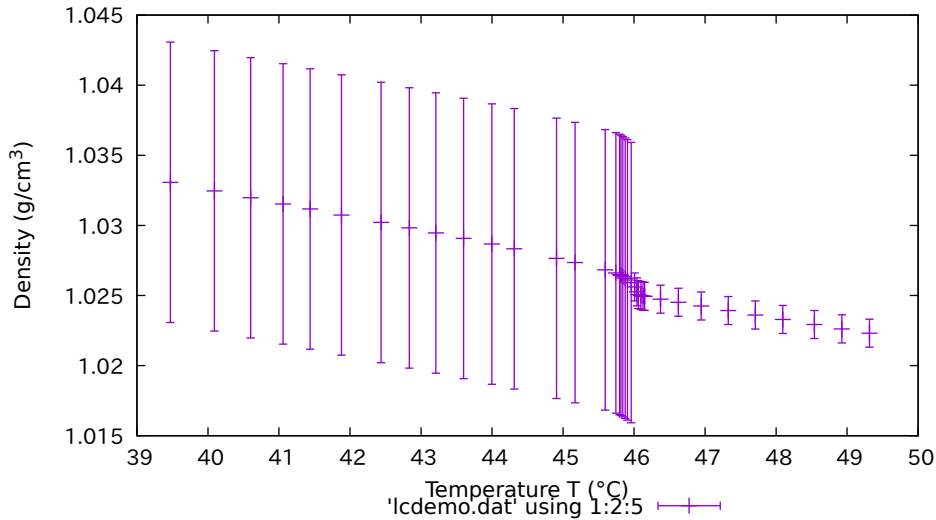
data set and initial parameters



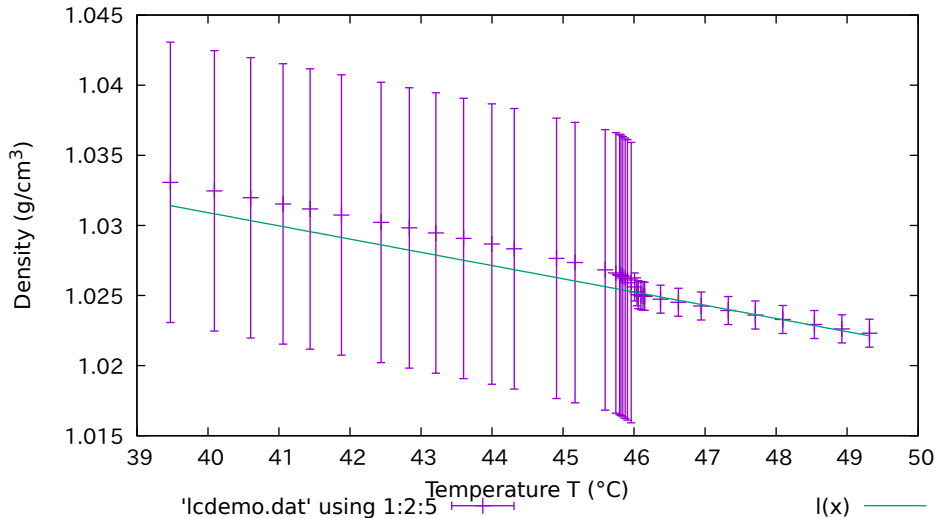
unweighted fit



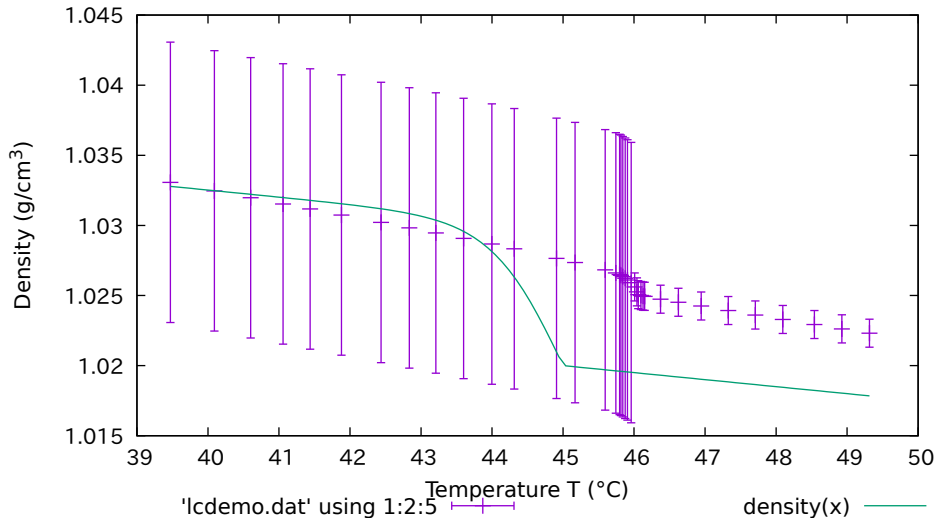
data with experimental weights



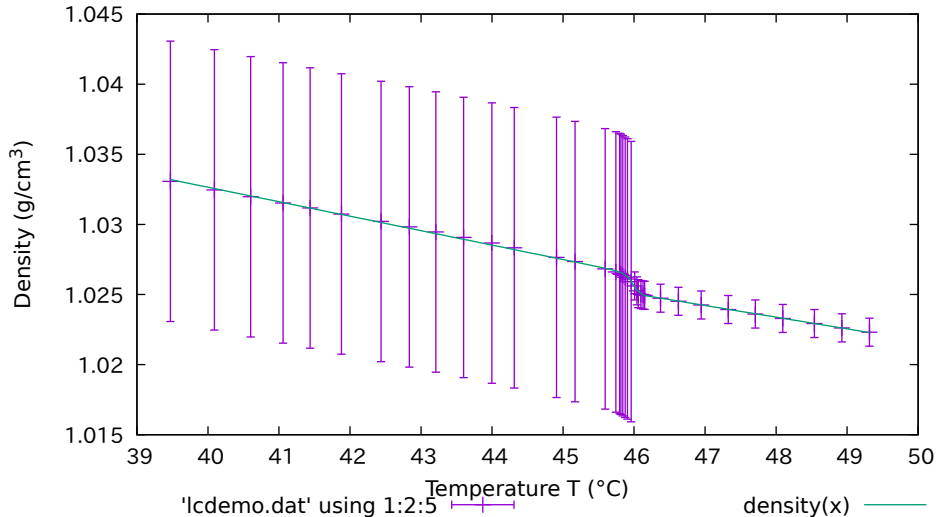
fit weighted by experimental weights



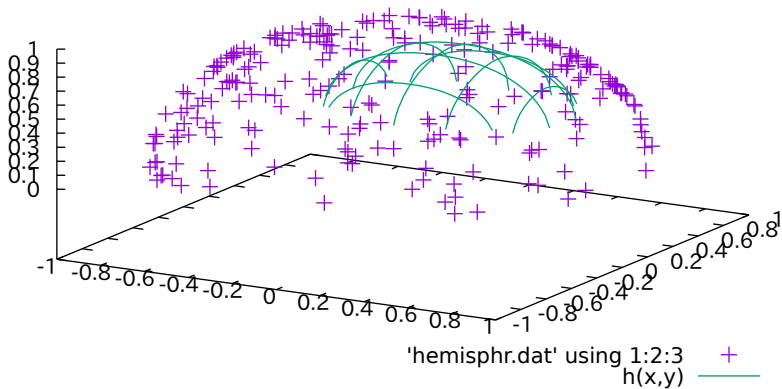
initial parameters for realistic model function



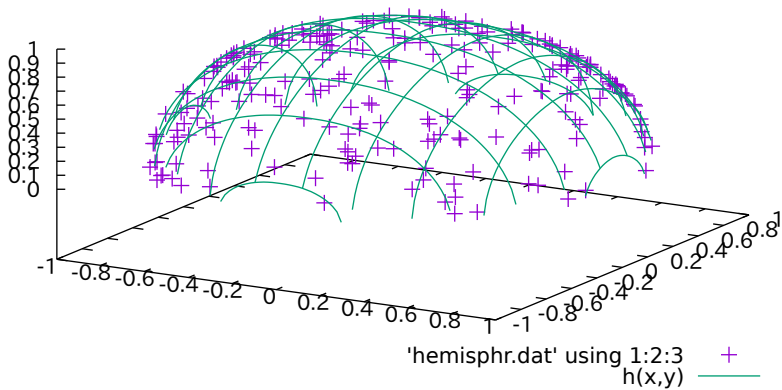
fitted to realistic model function



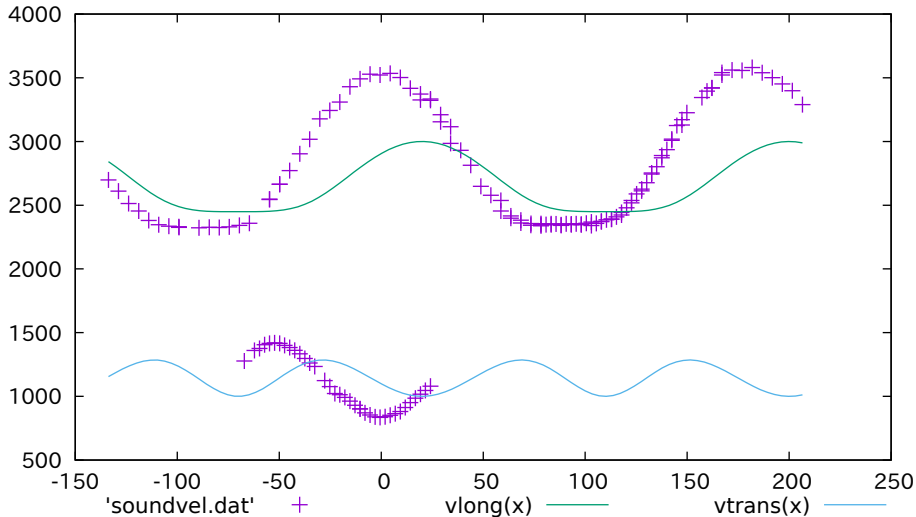
the scattered points, and the initial parameter



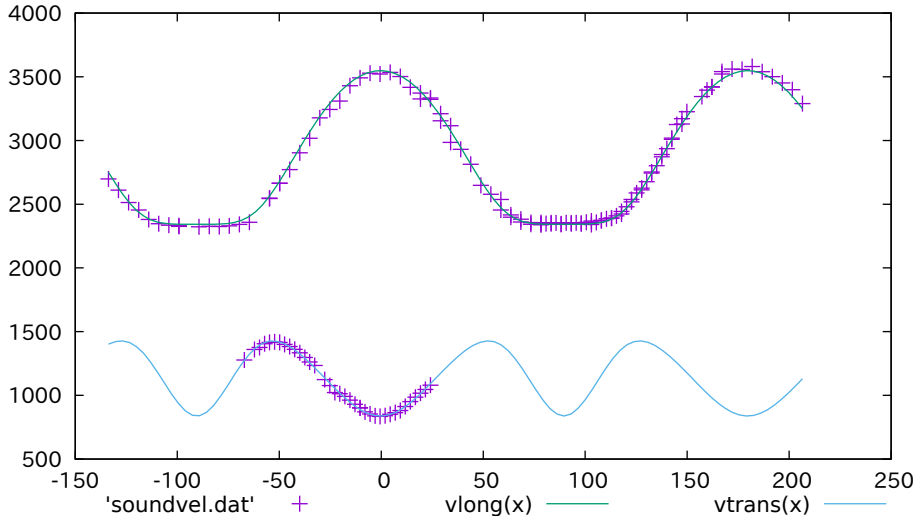
the scattered points, fitted curve



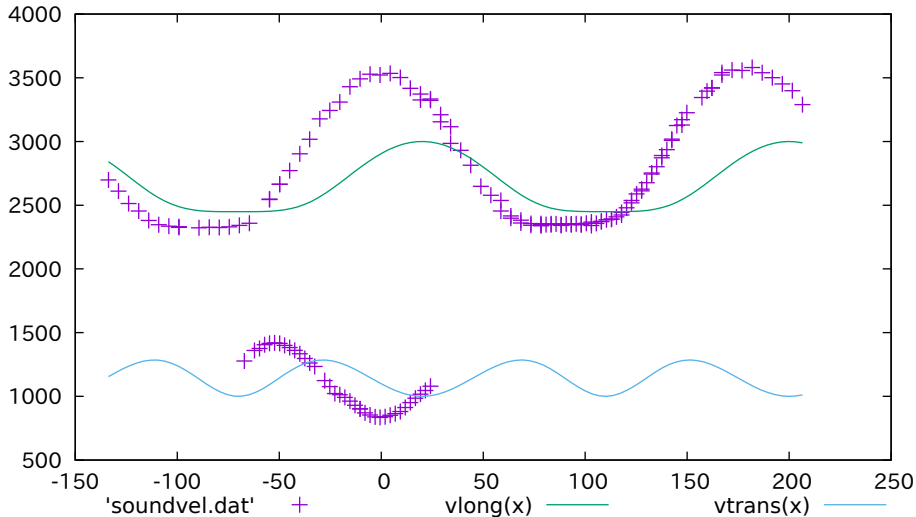
sound data, and model with initial parameters



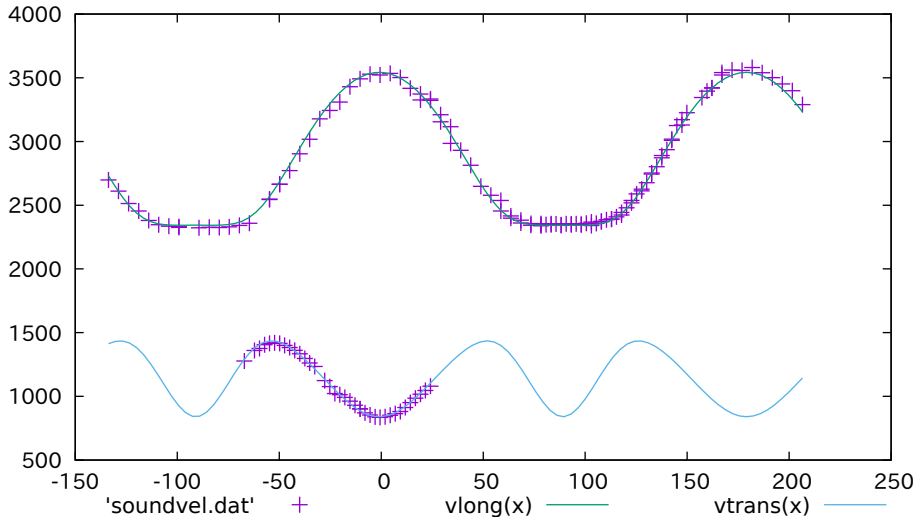
pseudo-3d multi-branch fit to velocity data



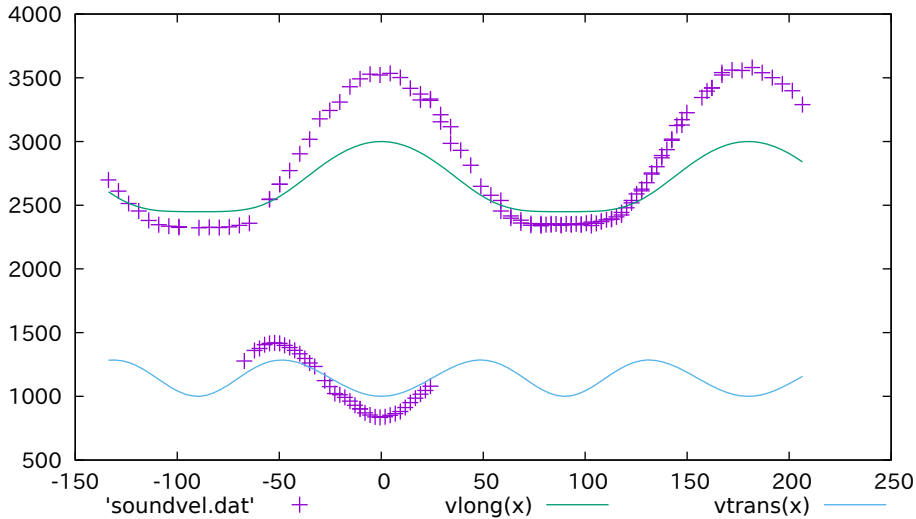
pseudo-3d multi-branch fit to velocity data



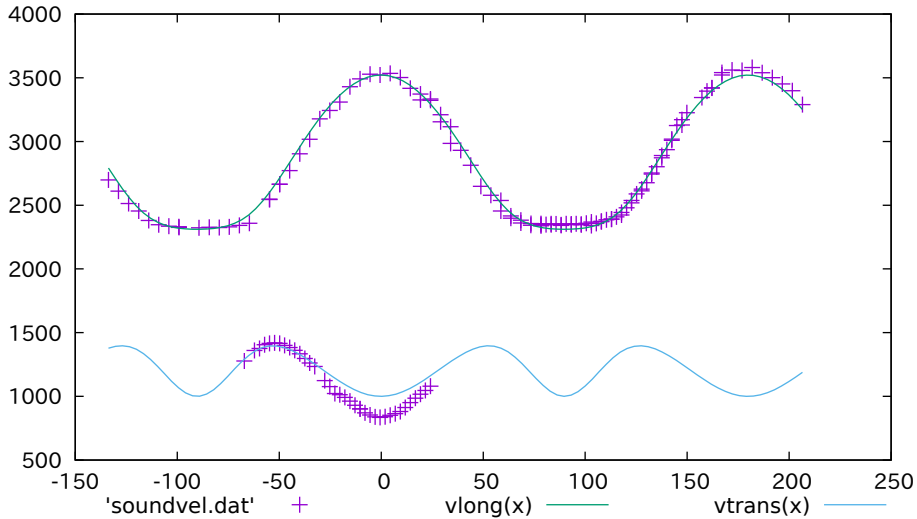
fitted only every 5th data point



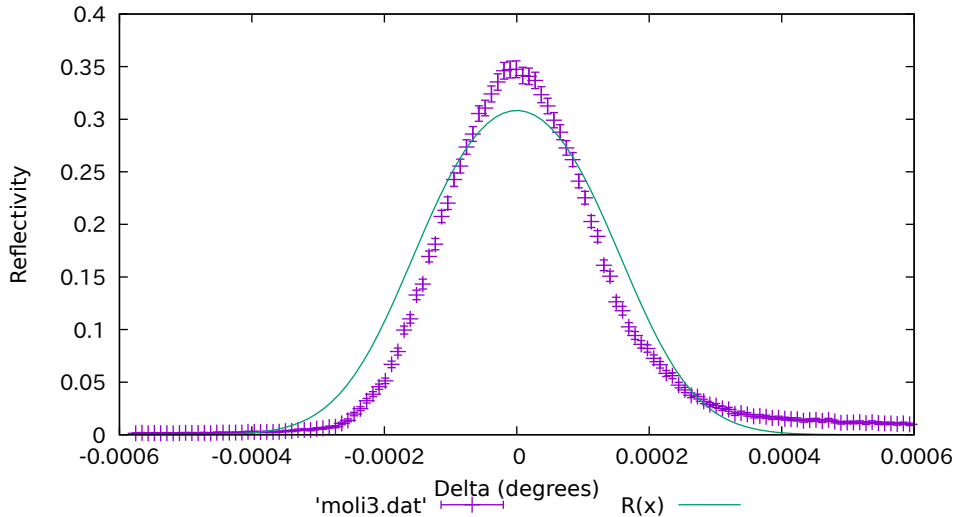
initial parameters



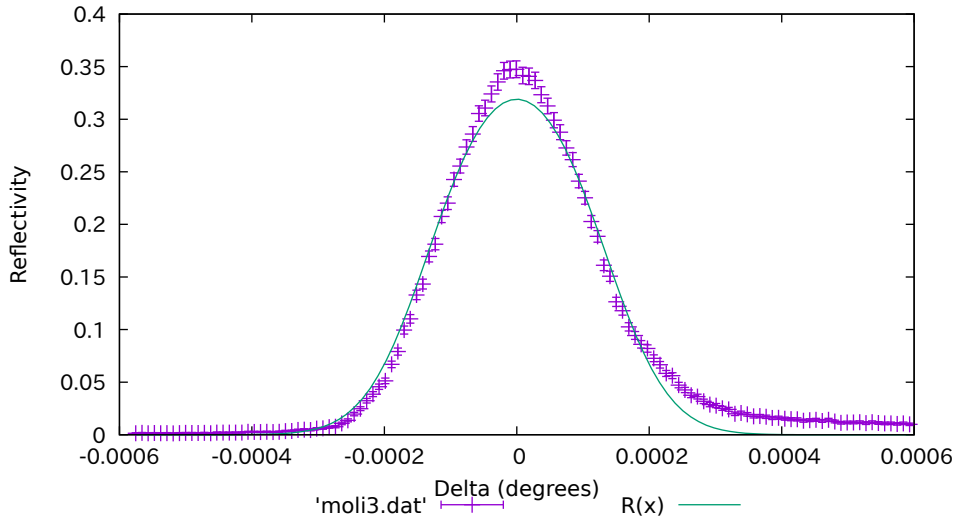
fit with c44 and c13 fixed



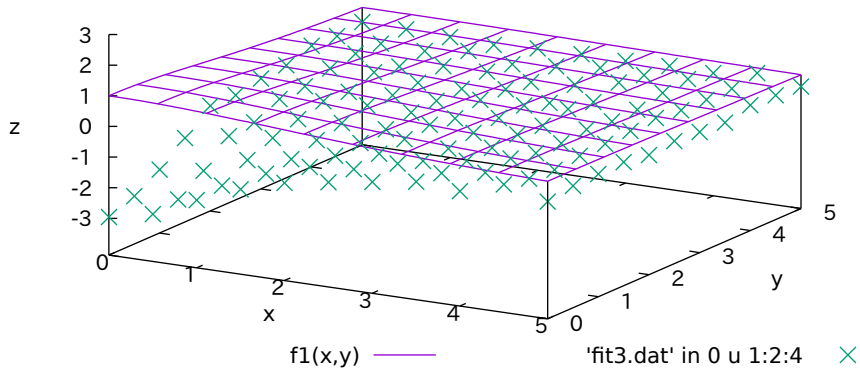
data and initial parameters



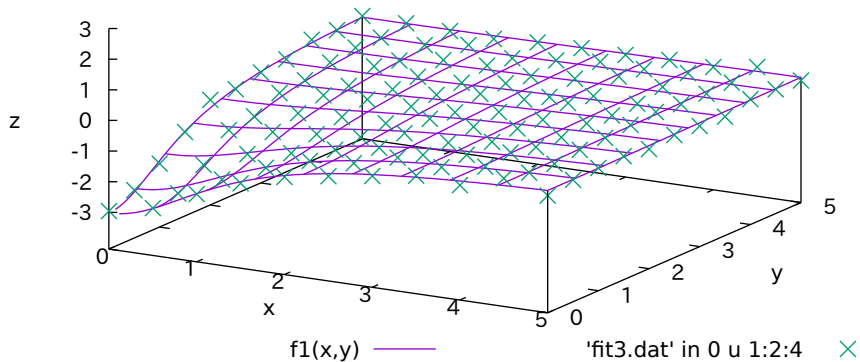
fitted parameters



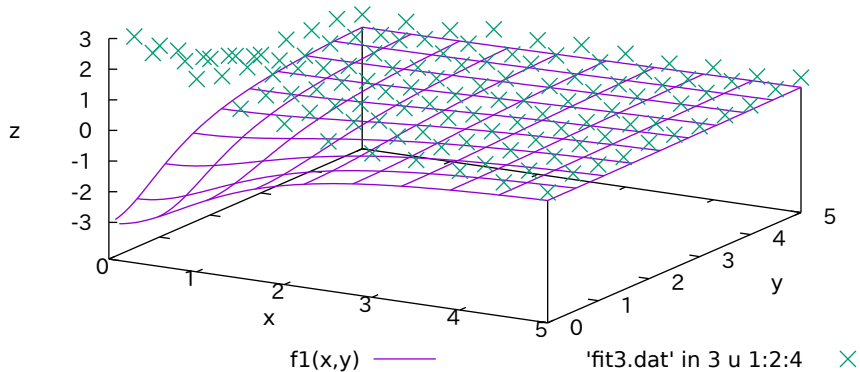
data and initial parameters



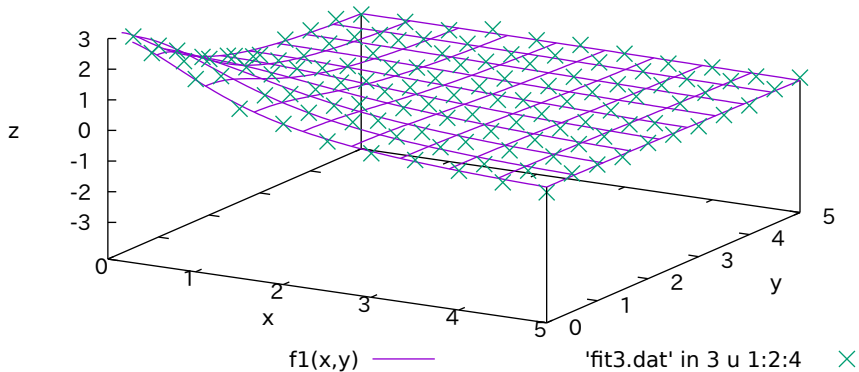
fit to data with $t = -3$



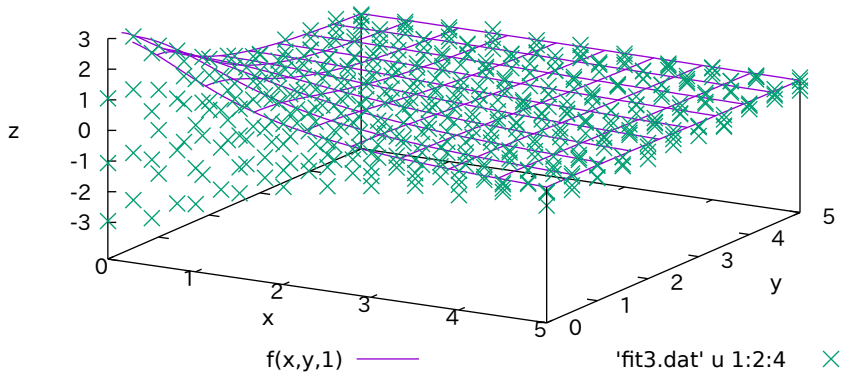
fit to data with $t = +3$, initial parameters



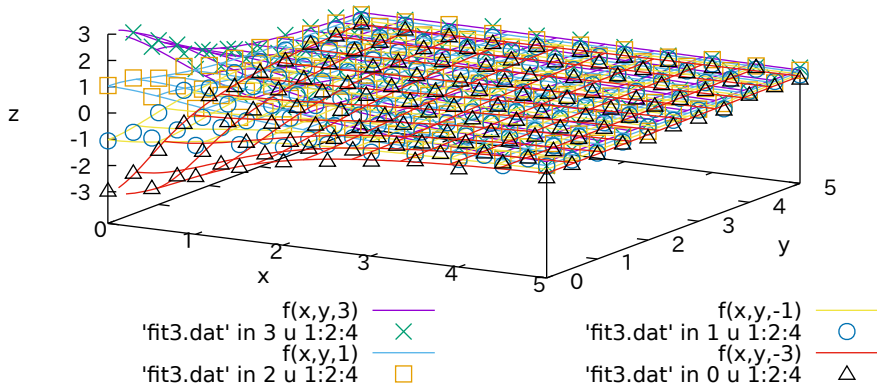
fit to data with $t = +3$



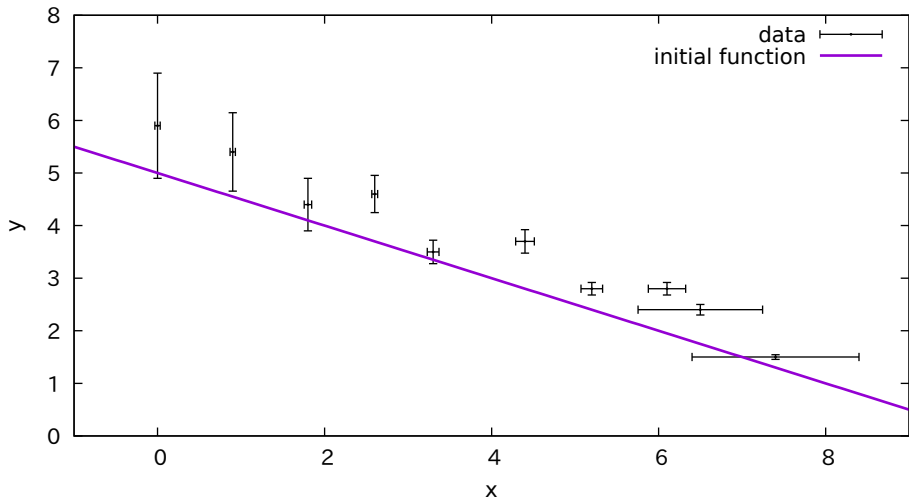
data for all indices t, initial parameters



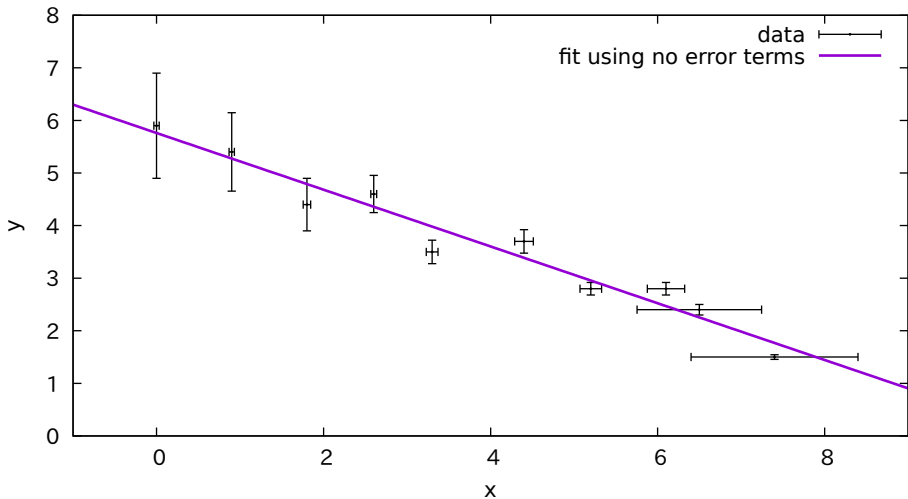
fit to all data



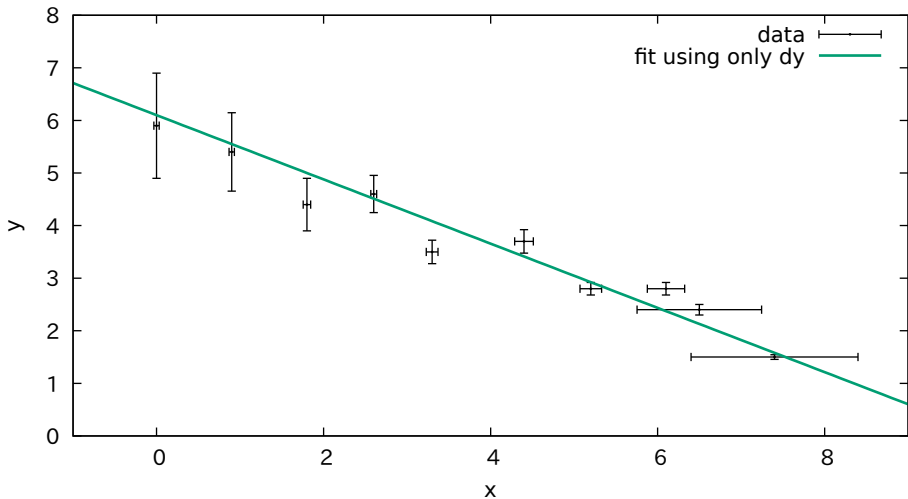
Pearson's data and York's weights
original data and the initial function



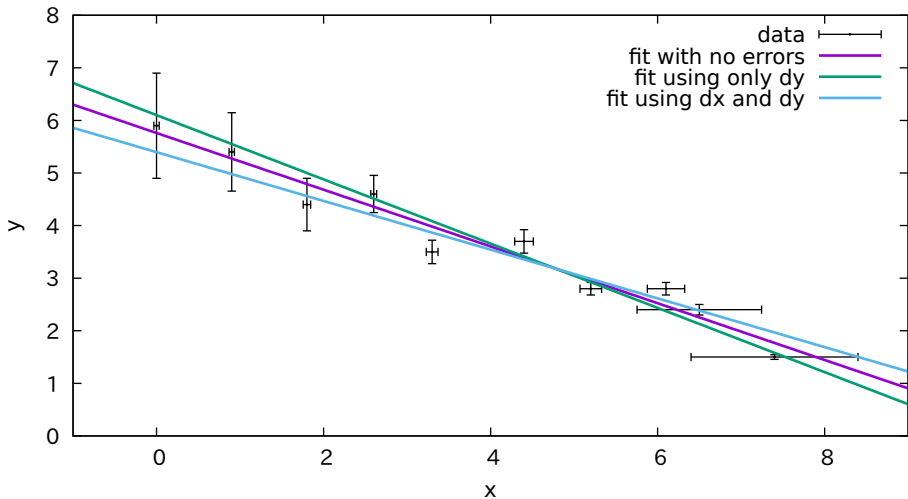
Pearson's data and York's weights
function fit with no error terms



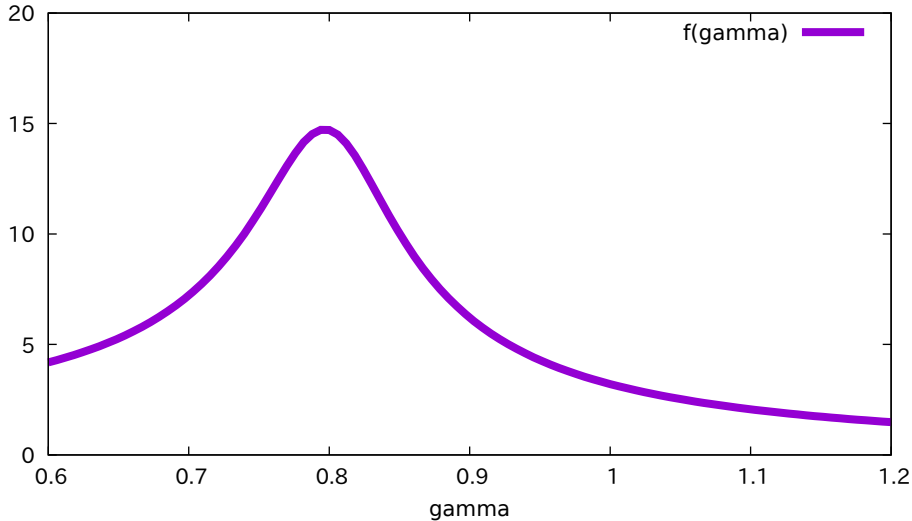
Pearson's data and York's weights
function fit with yerror keyword

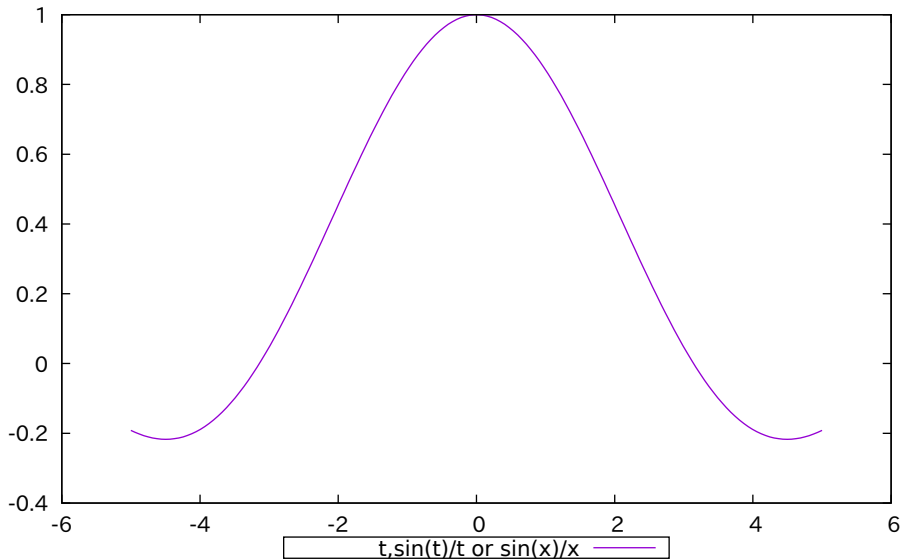


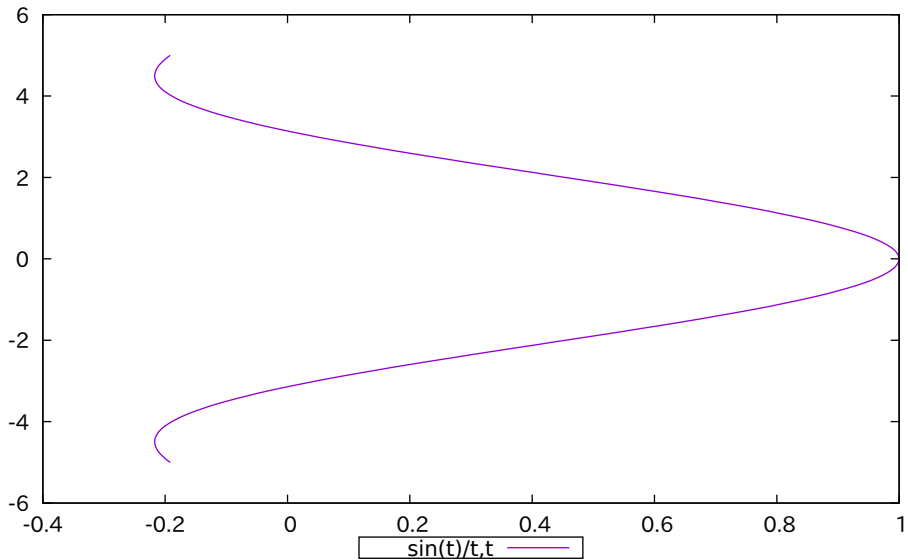
Pearson's data and York's weights
function fit with xyerror keyword

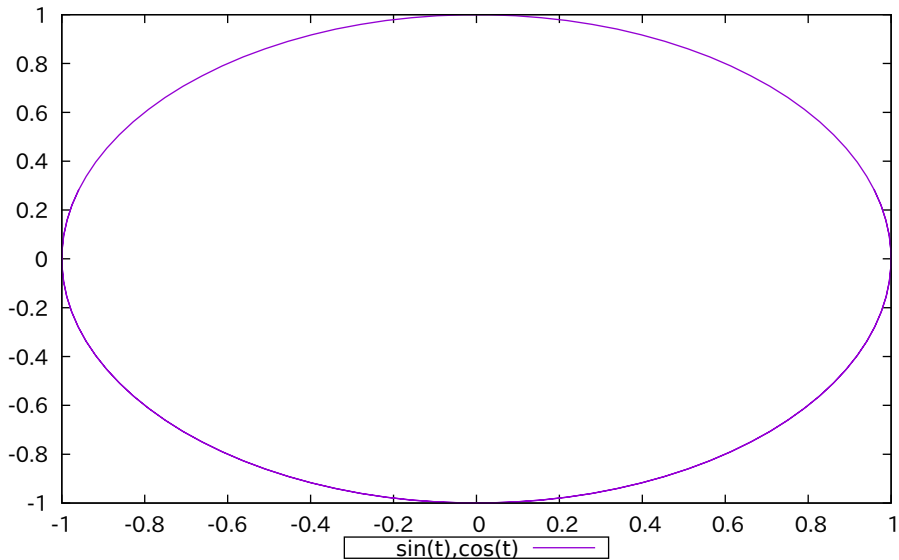


Plot a function of a named variable

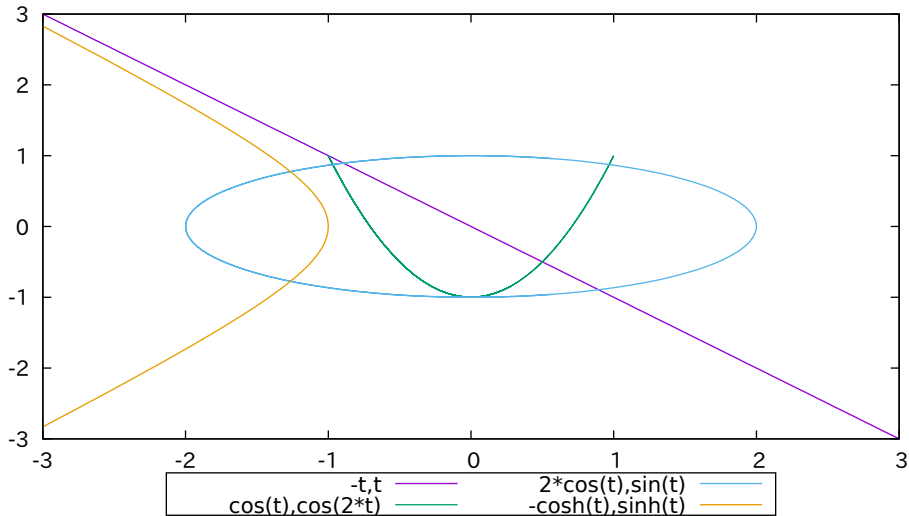


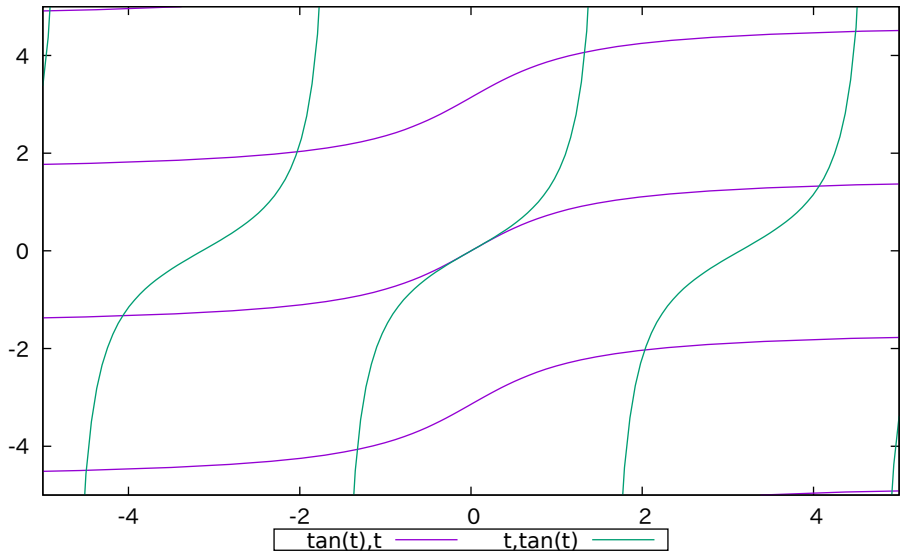


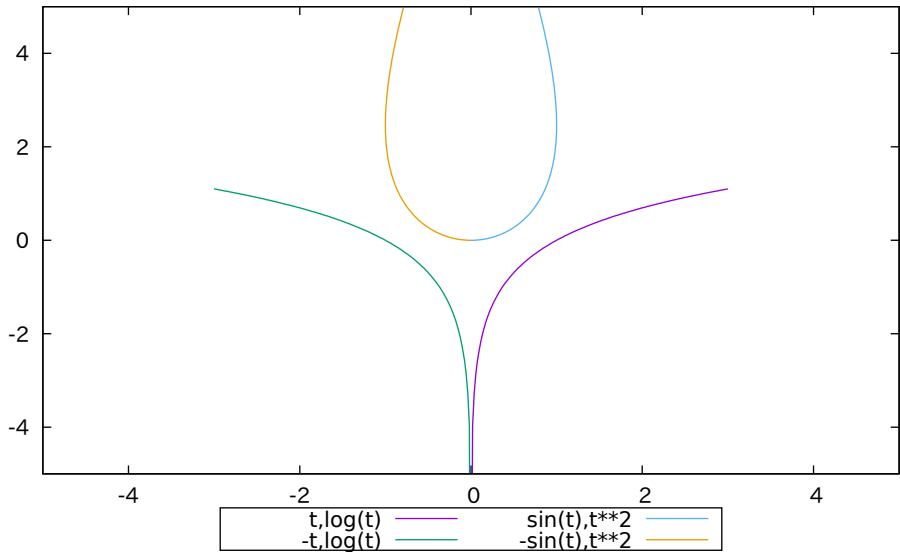


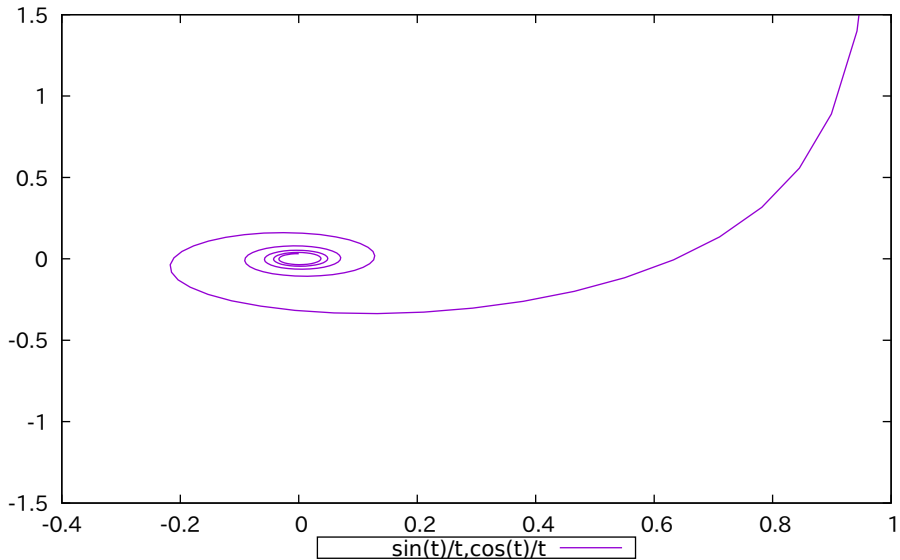


Parametric Conic Sections

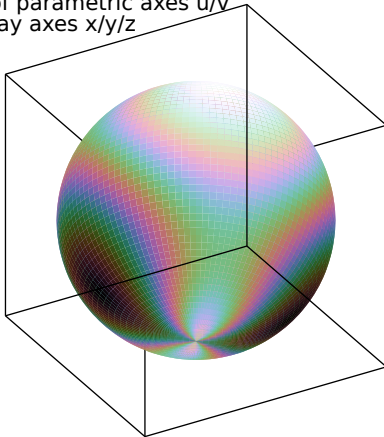




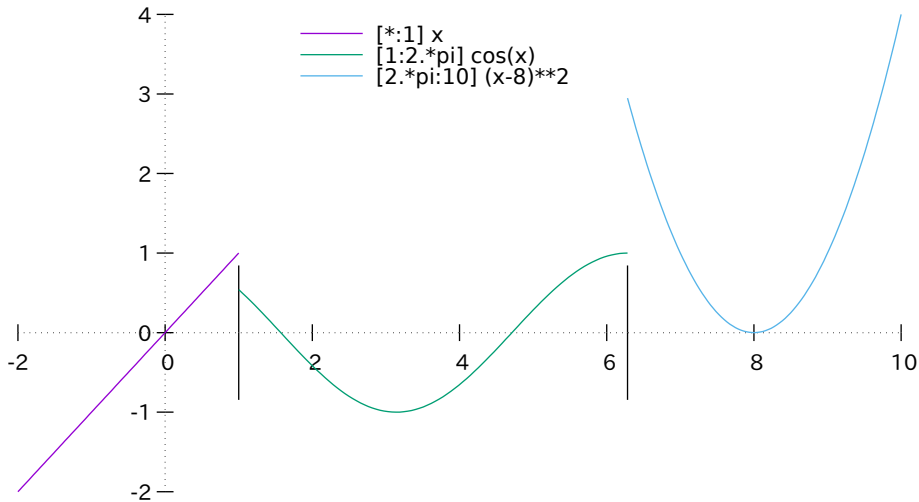




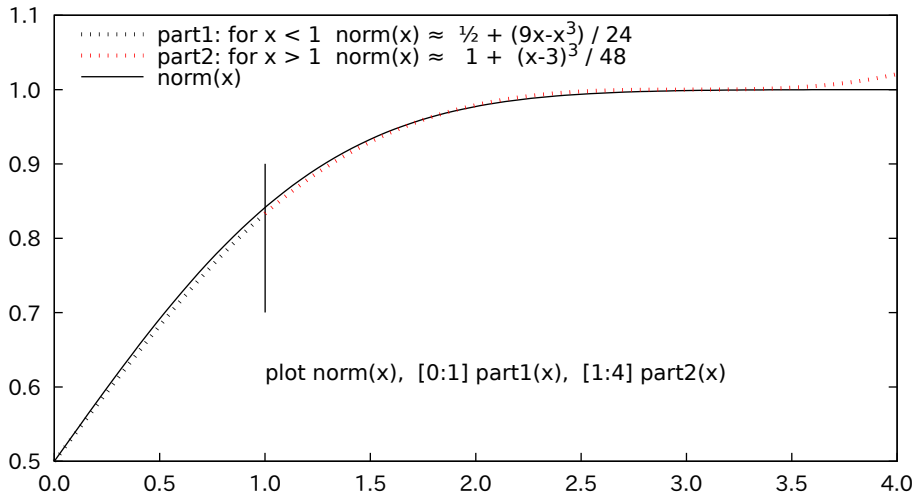
Decouple range of parametric axes u/v
from that of display axes $x/y/z$



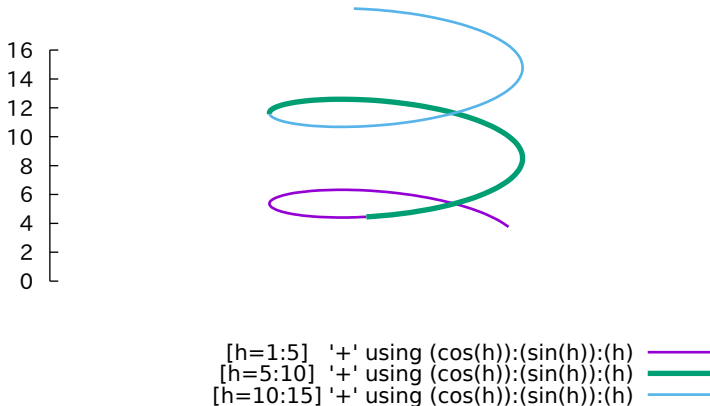
Piecewise function sampling



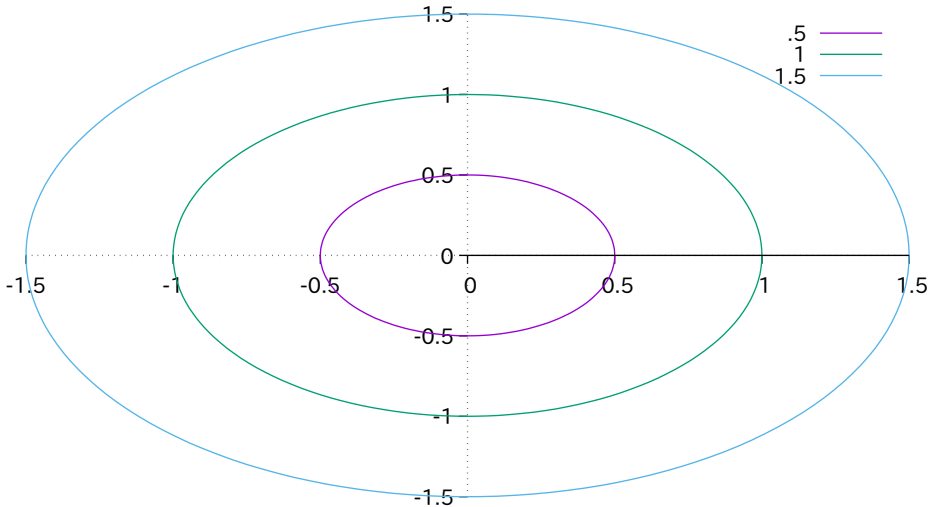
Piecewise approximation to the Normal Cumulative Distribution Function

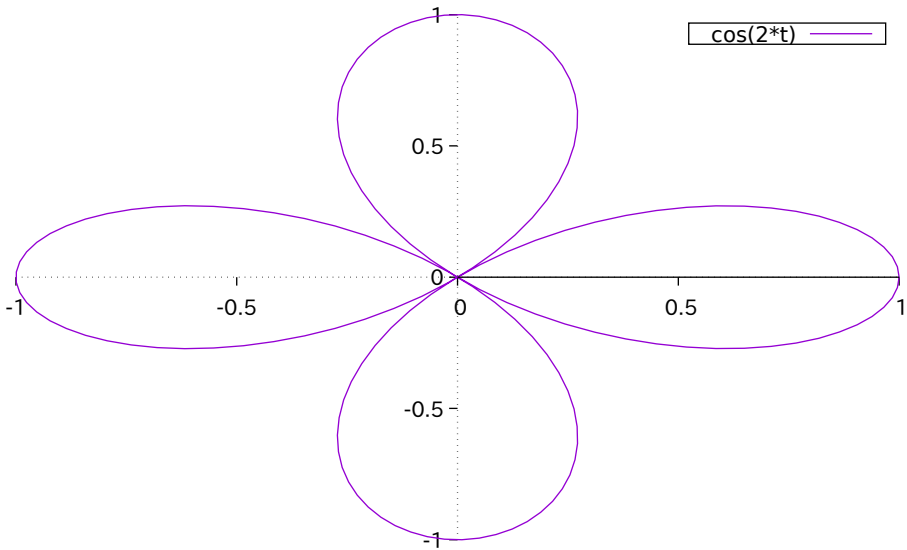


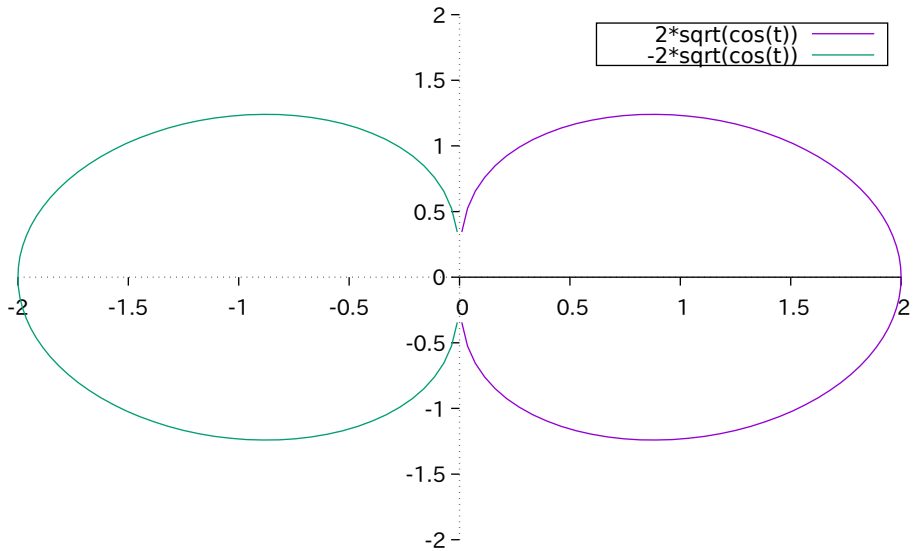
Piecewise function of one parameter in 3D

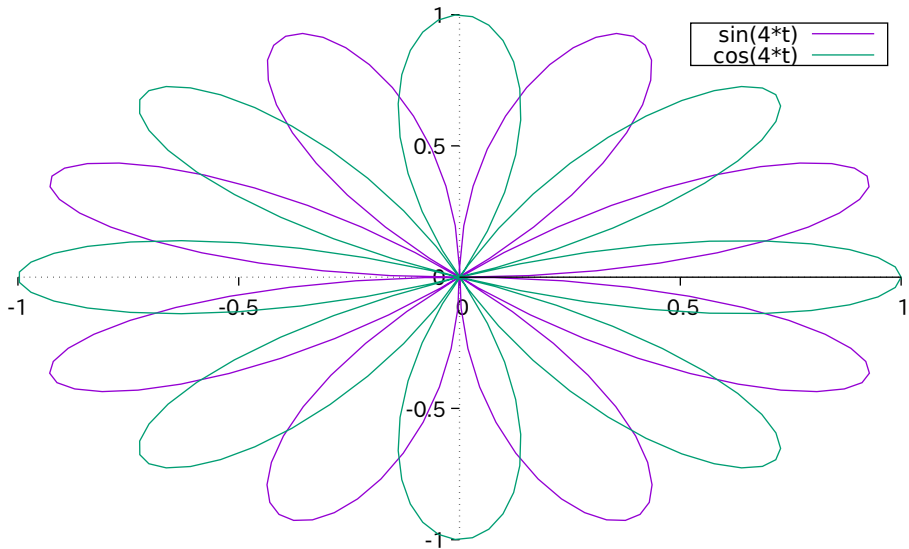


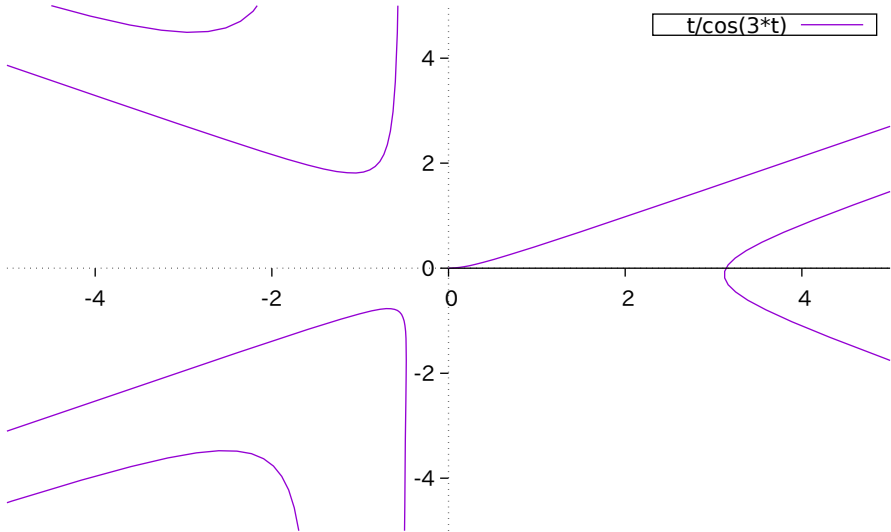
Three circles (with aspect ratio distortion)

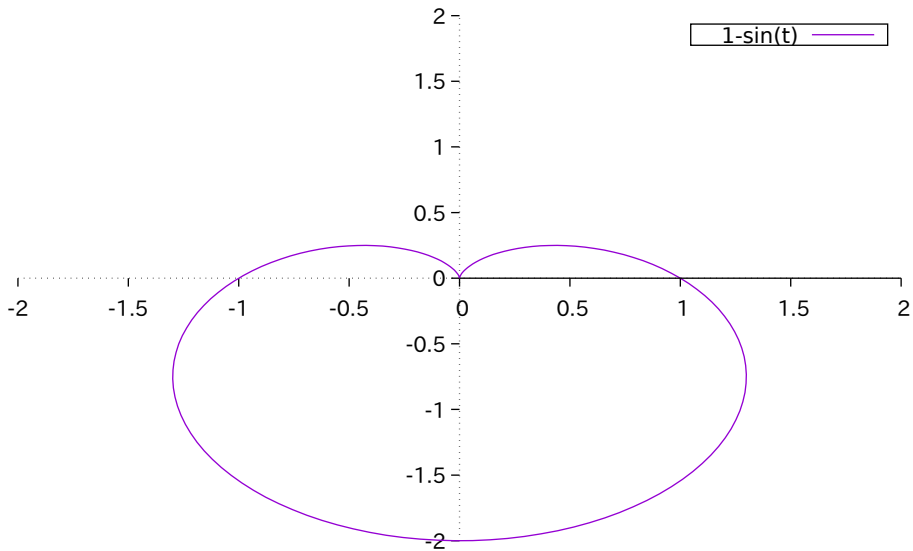


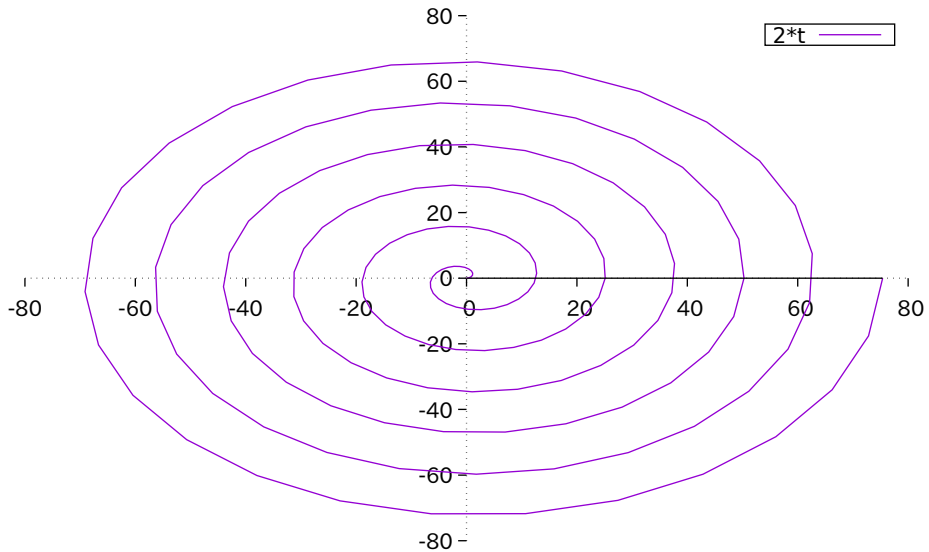




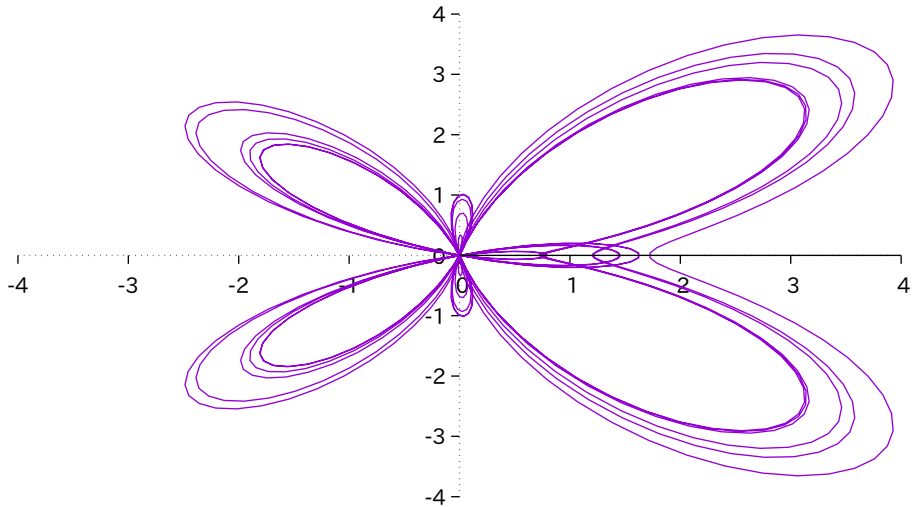




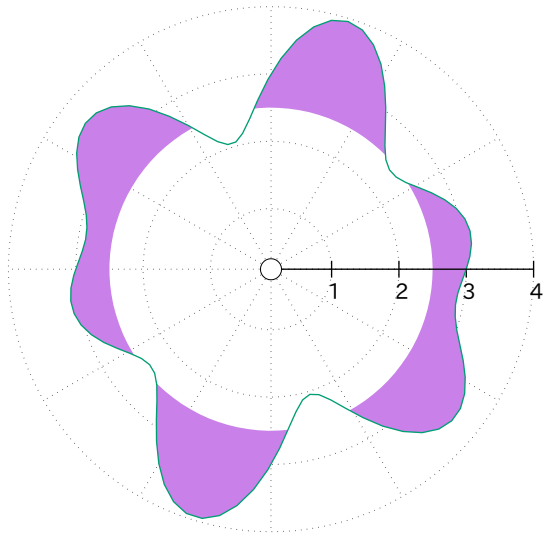




Butterfly

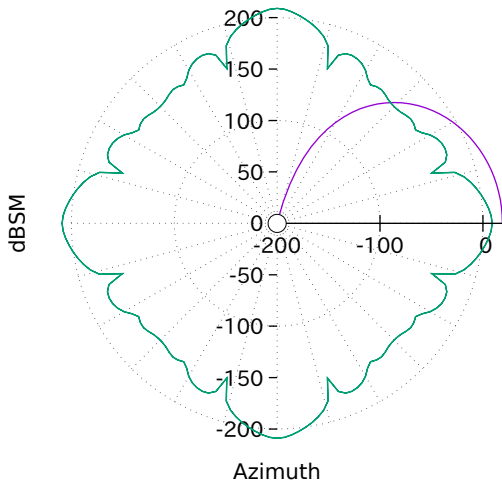


bounding radius 2.5
 $3. + \sin(t) * \cos(5 * t)$ —

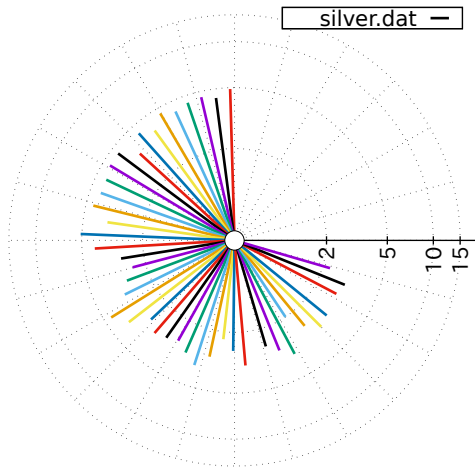


火 21 1月 2020

Antenna Pattern

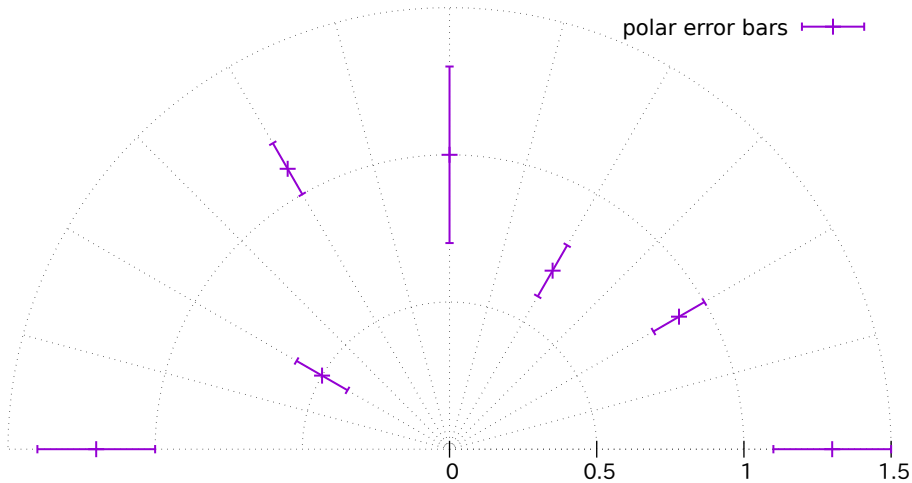


log scale polar axis, trange in degrees

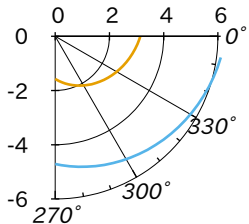
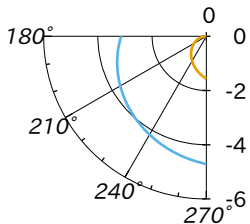
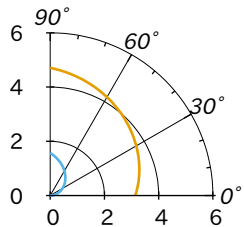
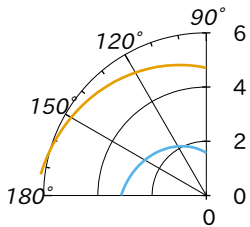


yerrors in polar mode

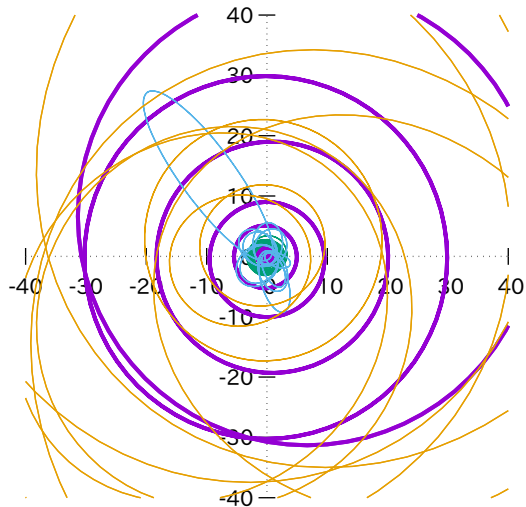
polar error bars 



Polar Quadrants



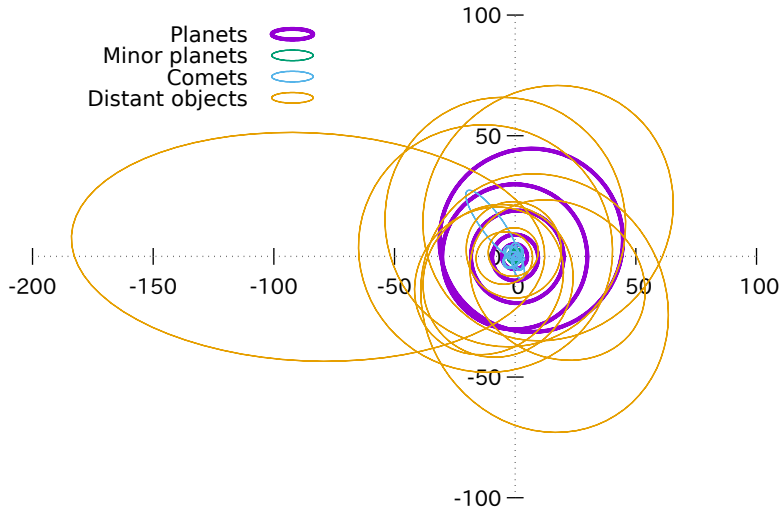
Orbits of selected Solar System objects



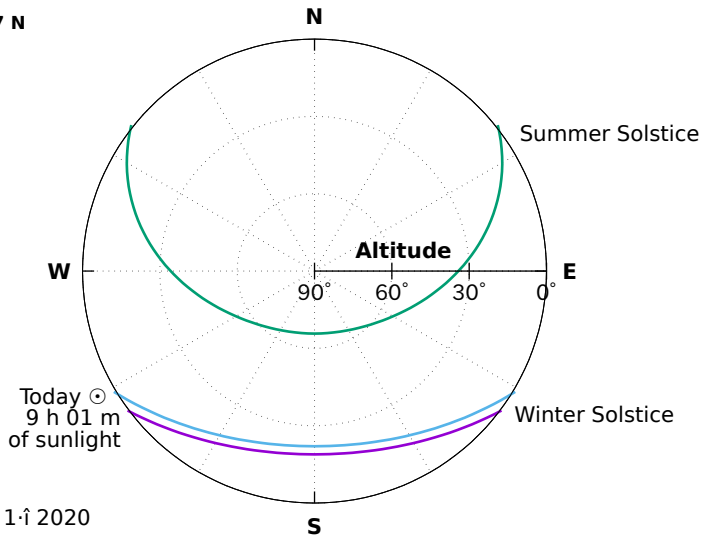
Planets
Minor planets
Comets
Distant objects

Orbits of selected Solar System objects

- Planets
- Minor planets
- Comets
- Distant objects

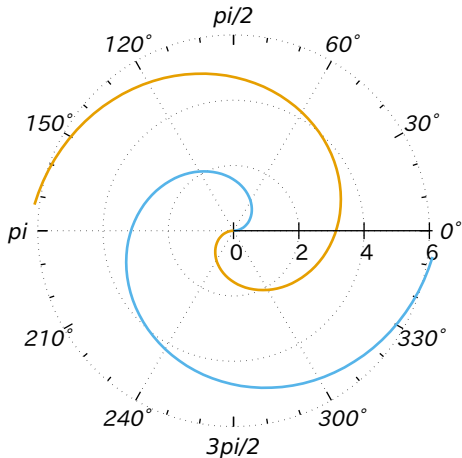


**Solar path at
Latitude 47.67 N**

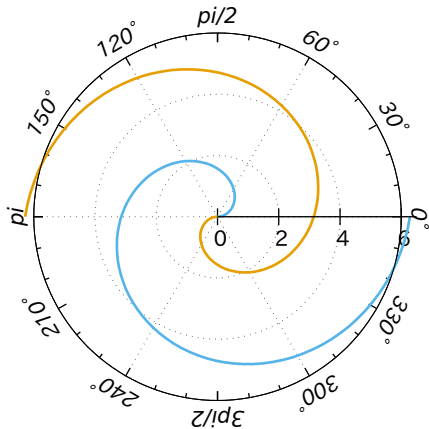


Seattle - 21 1·2020

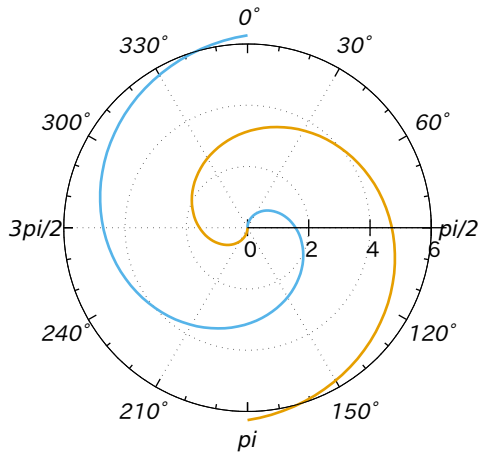
Angle labels (tticks) for polar plots

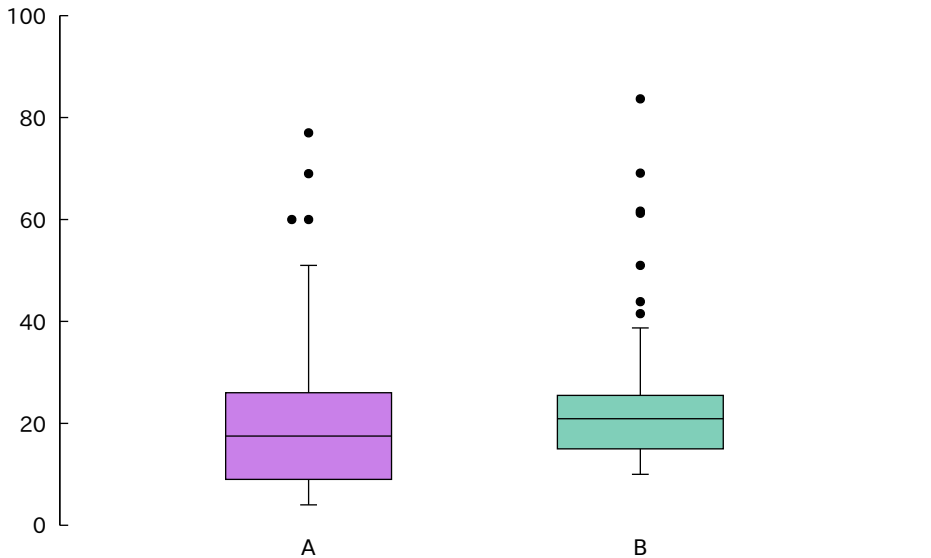


Polar plot with border and rotated labels for tticks

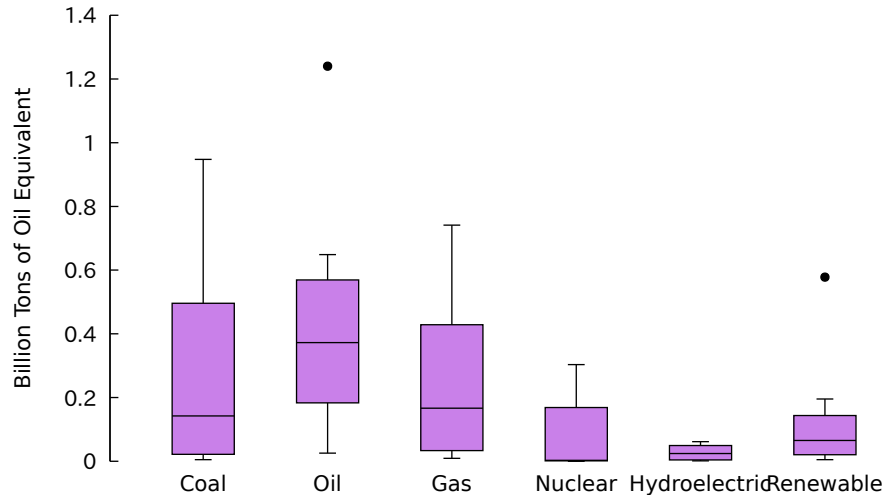


Theta origin at top, increasing clockwise

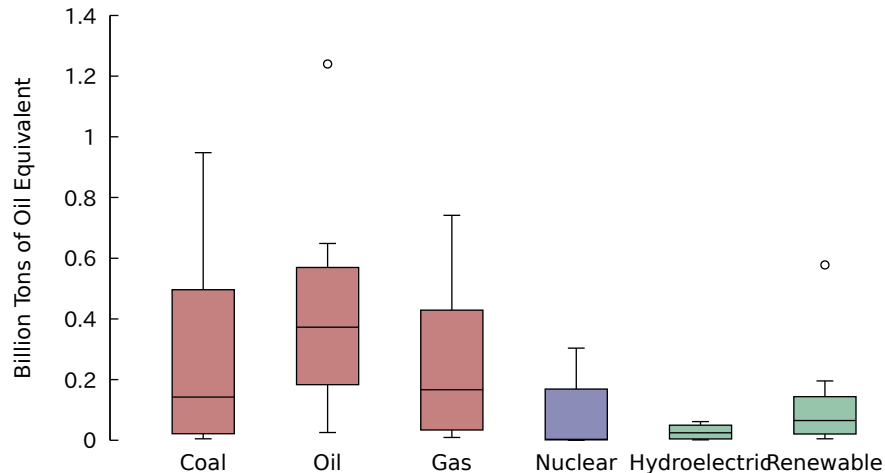




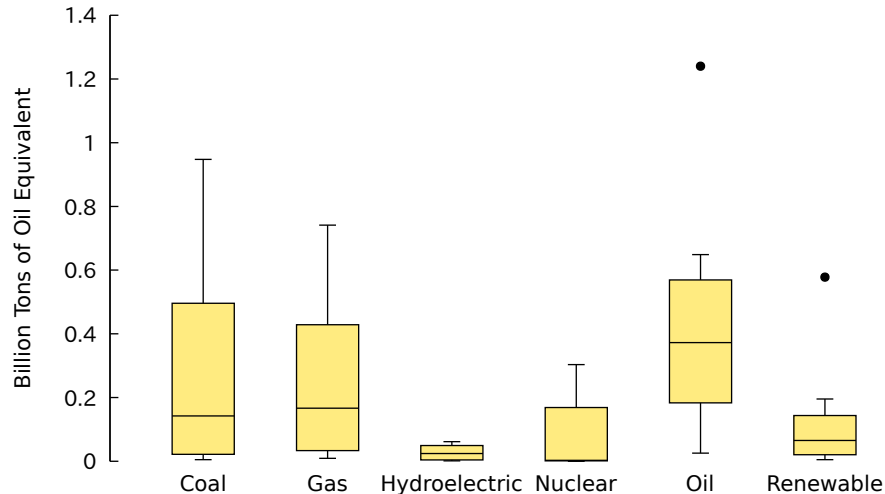
Distribution of energy usage of the continents, grouped by type of energy source



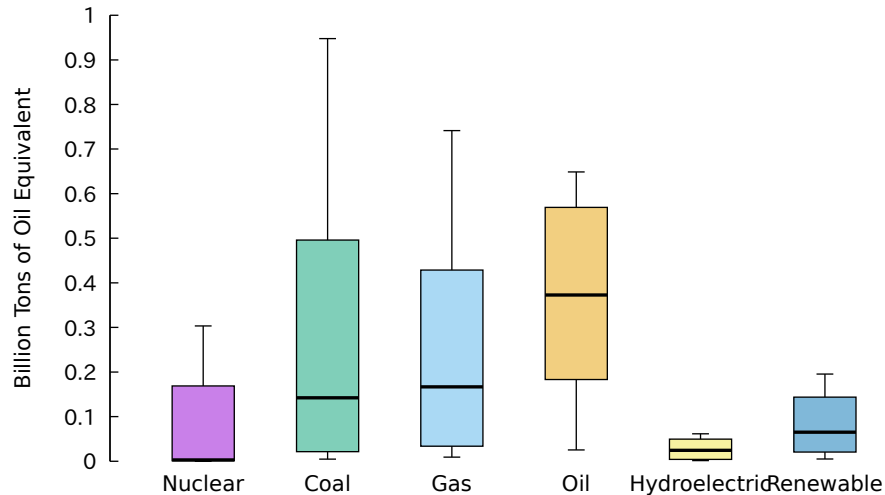
Distribution of energy usage of the continents, grouped by type of energy source, assign individual colors (linetypes) to the factors taken from column 4



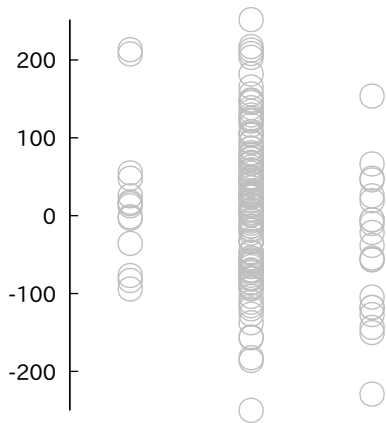
Distribution of energy usage of the continents, sorted by name of energy source



Distribution of energy usage explicitly ordered by name of energy source



no jitter

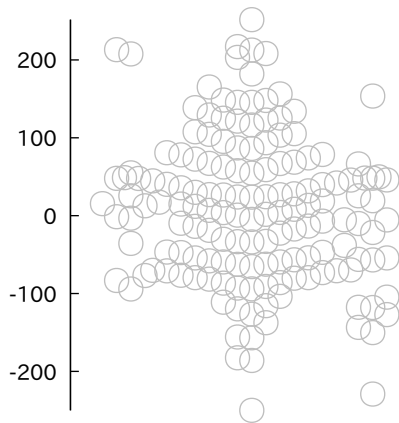


A

B

C

jitter



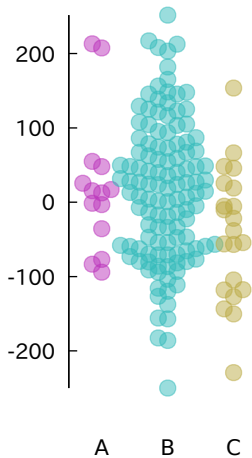
A

B

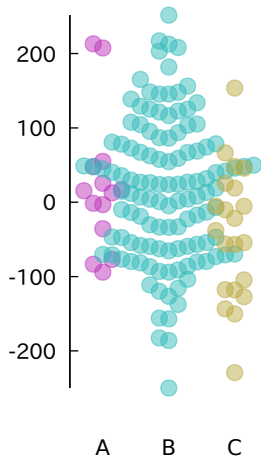
C

vertical overlap criterion

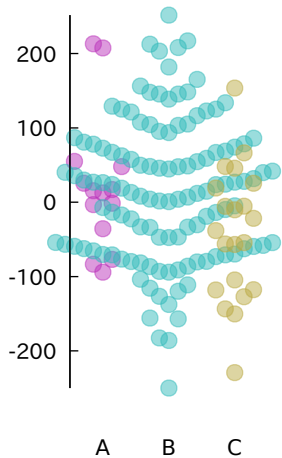
jitter overlap 0.5



jitter overlap 1.0

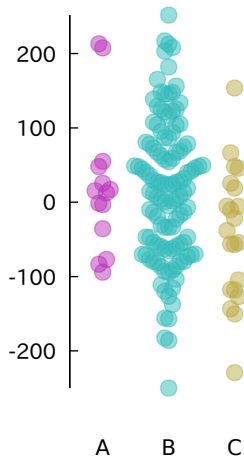


jitter overlap 1.5

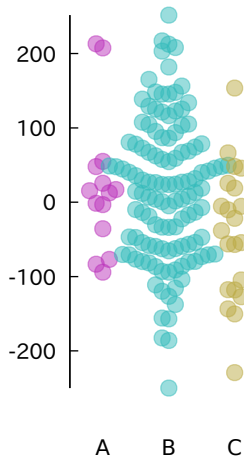


spread parameter scales the horizontal jitter

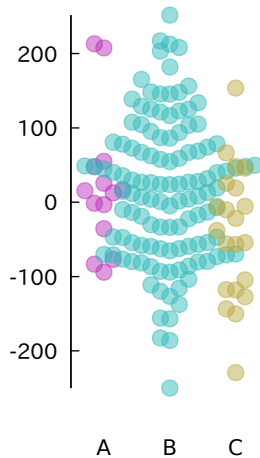
jitter spread 0.4



jitter spread 0.7

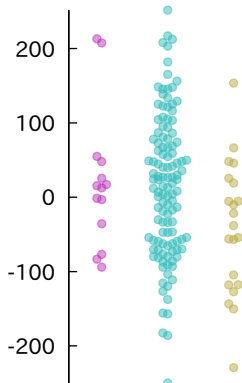


jitter spread 1.0



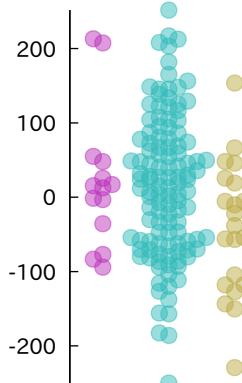
Plot appearance is also affected by point size

pointsize 0.5



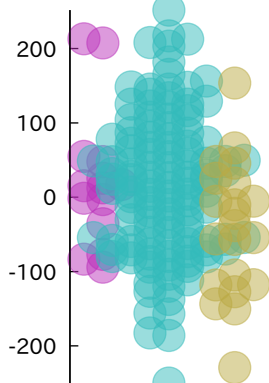
A B C

pointsize 1.0



A B C

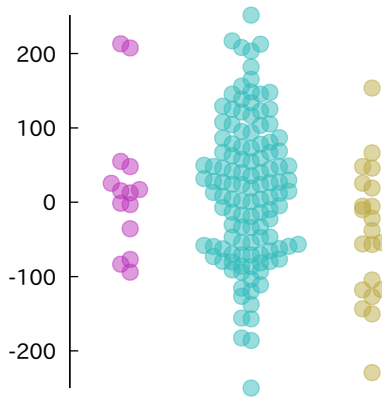
pointsize 2.0



A B C

Jitter style options

swarm (default)

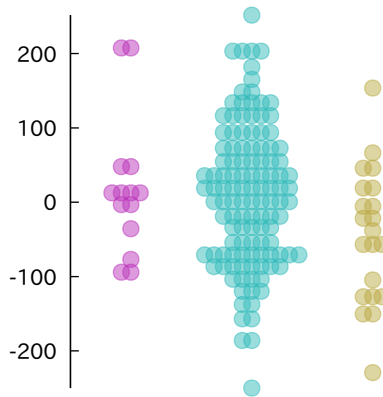


A

B

C

square



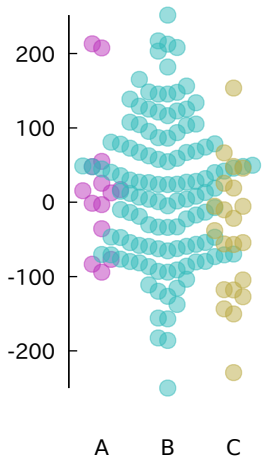
A

B

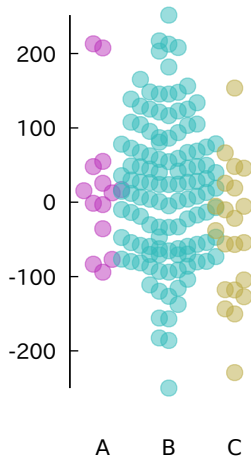
C

Jitter style options

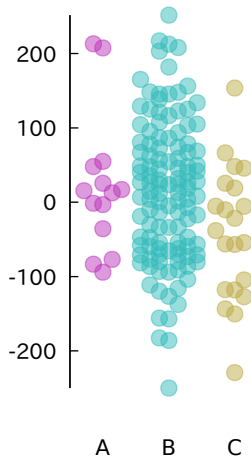
no wrap



wrap 5

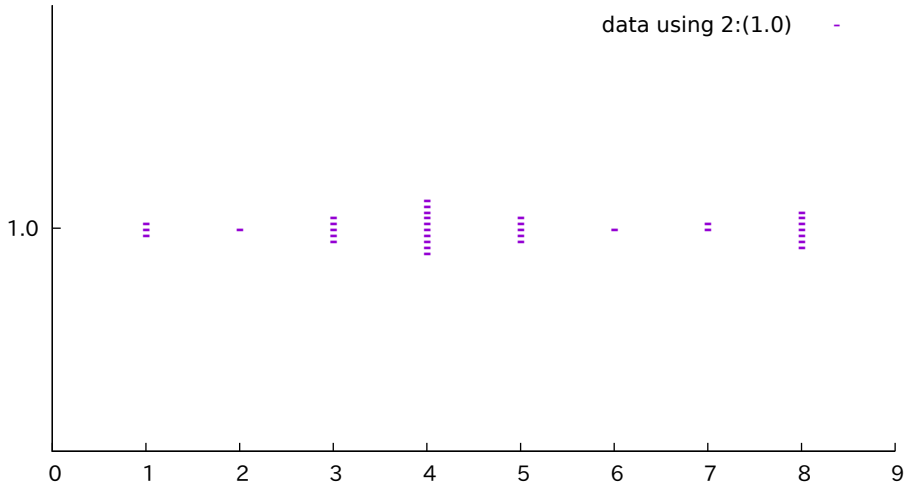


wrap 3

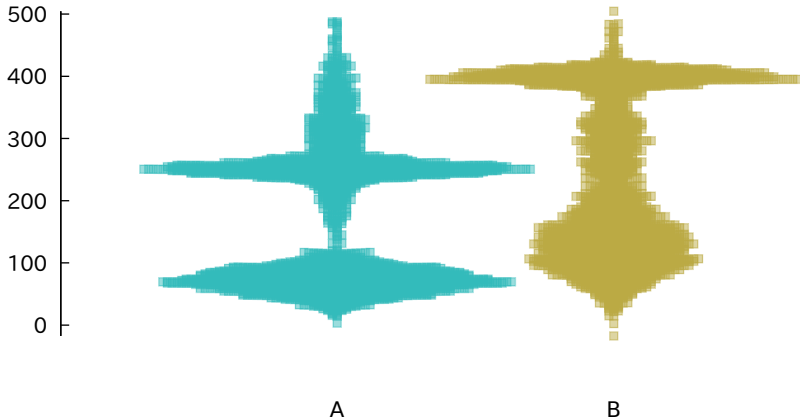


Jitter style option
vertical

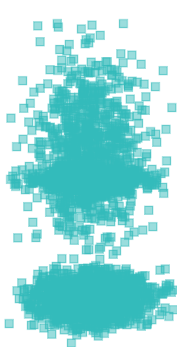
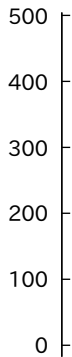
data using 2:(1.0)



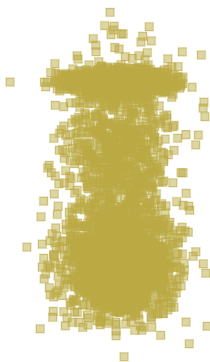
swarm jitter with a large number of points
approximates a violin plot



Gaussian random jitter

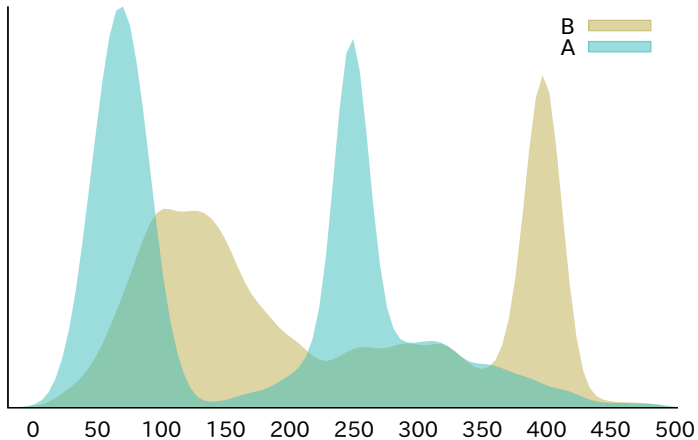


A

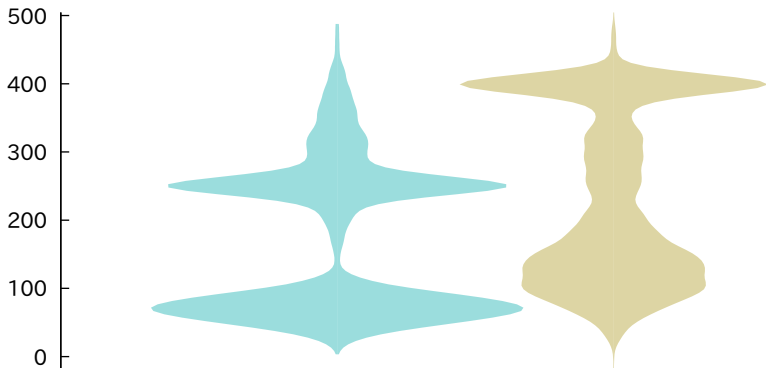


B

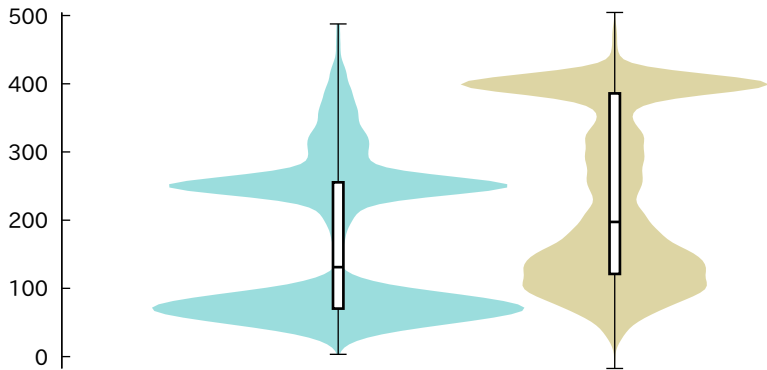
Same data - kernel density



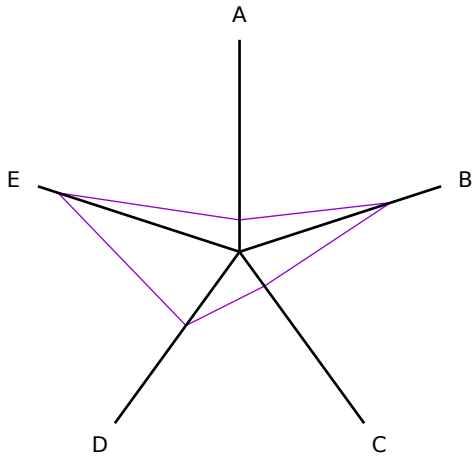
kdensity mirrored sideways to give a violin plot



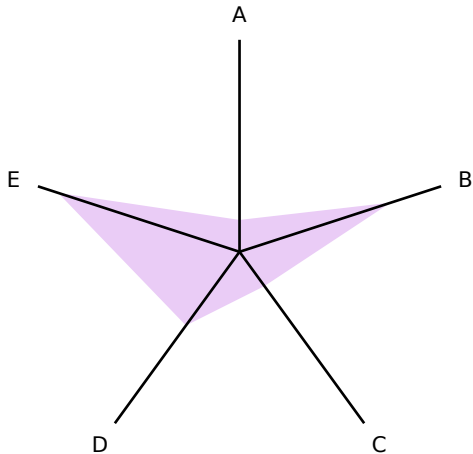
Superimposed violin plot and box plot



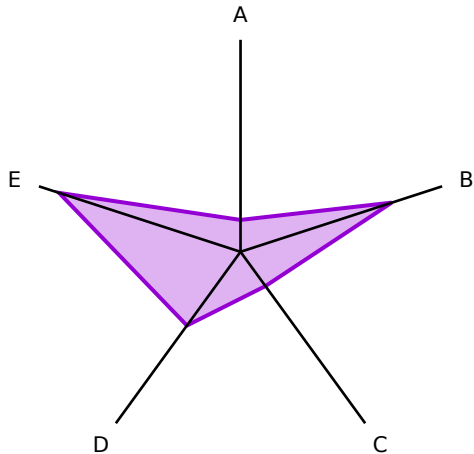
default spiderplot style



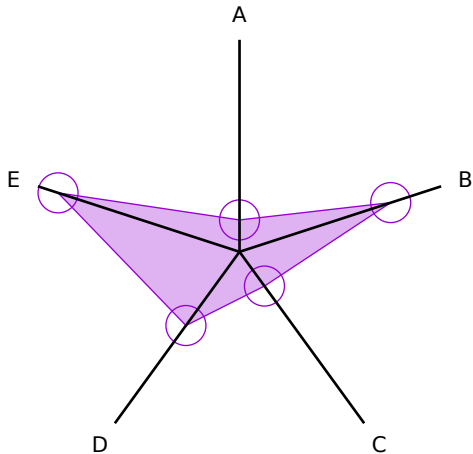
spiderplot fillstyle solid 0.2 noborder



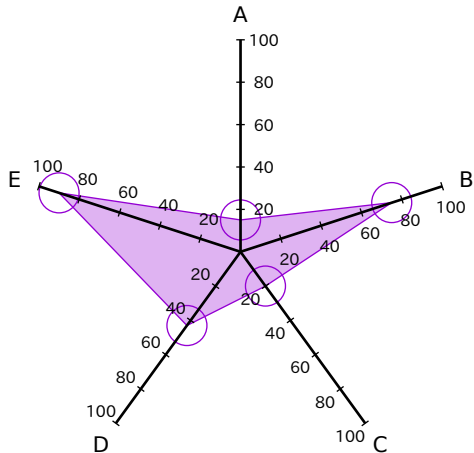
spiderplot fs solid 0.3 border lw 3.0



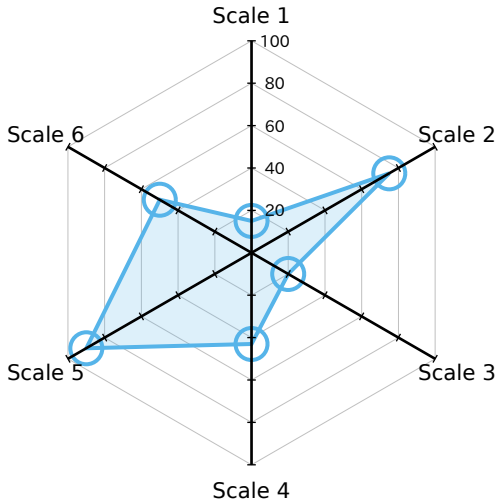
spiderplot fs solid 0.3 border lw 1 pt 6 ps 2.5



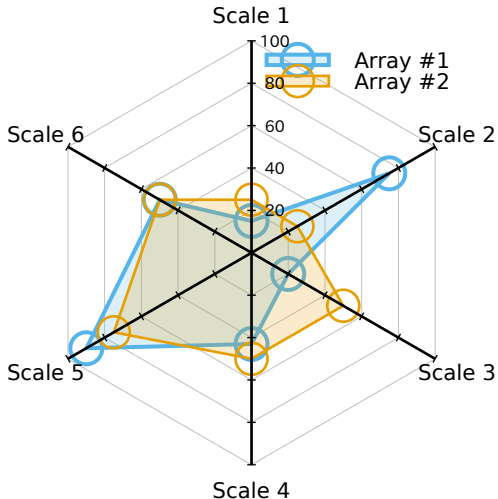
set for [p=1:5] paxis p tics font ',9'

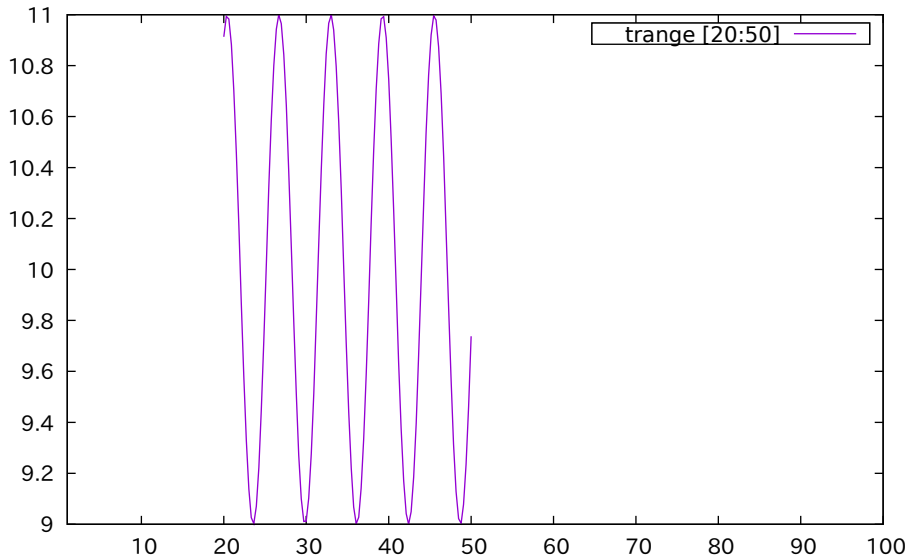


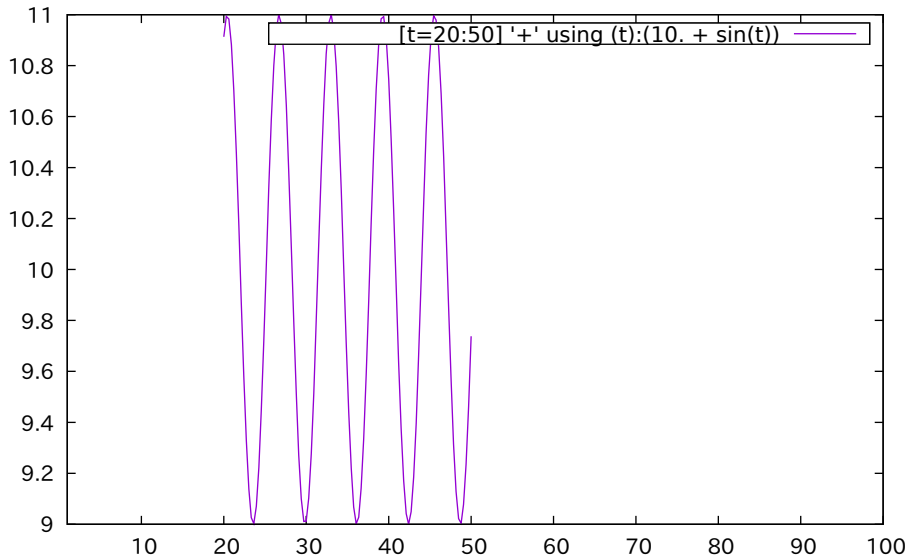
spiderplot from data array, show grid

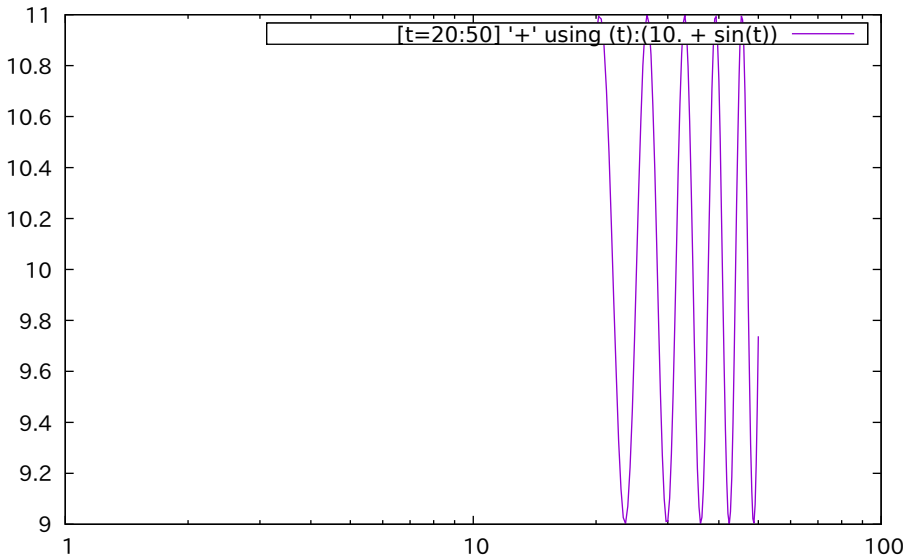


To plot from 2 different files or arrays, use 'newspiderplot'

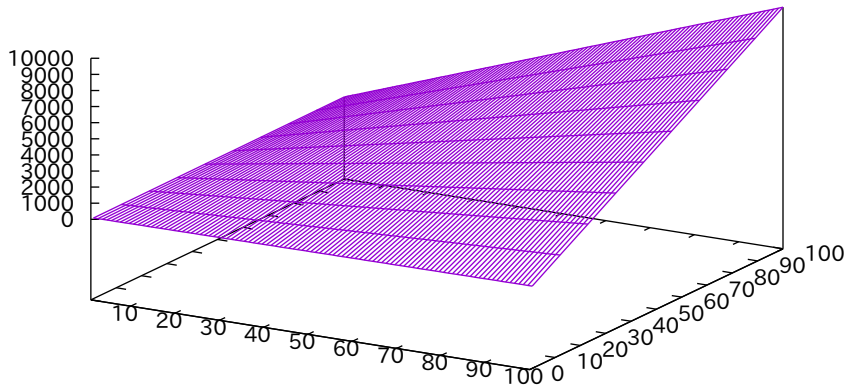




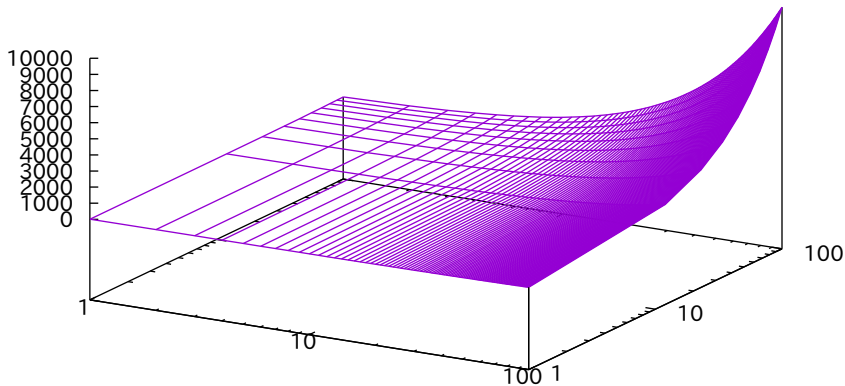


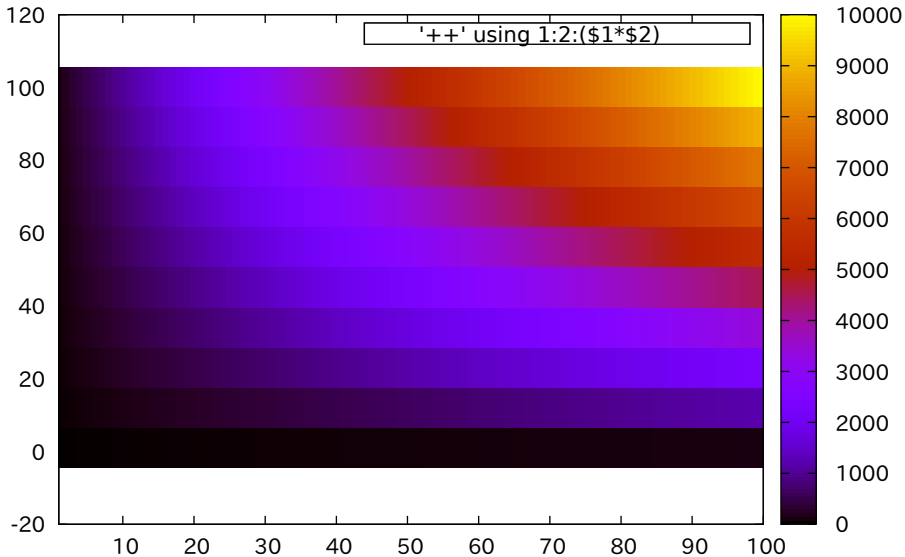


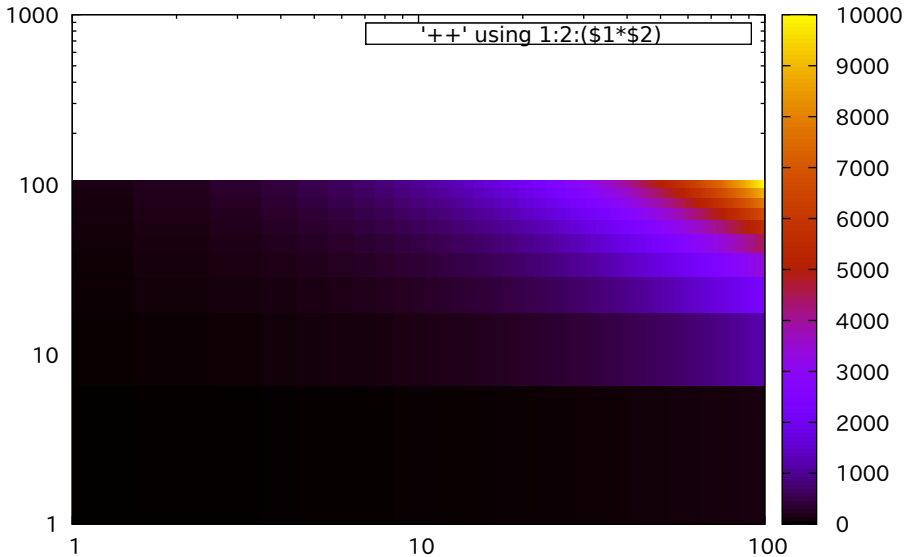
'++' using 1:2:(\$1*\$2)



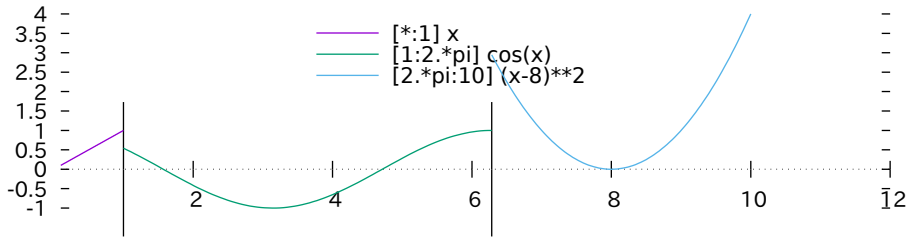
'++' using 1:2:(\$1*\$2)



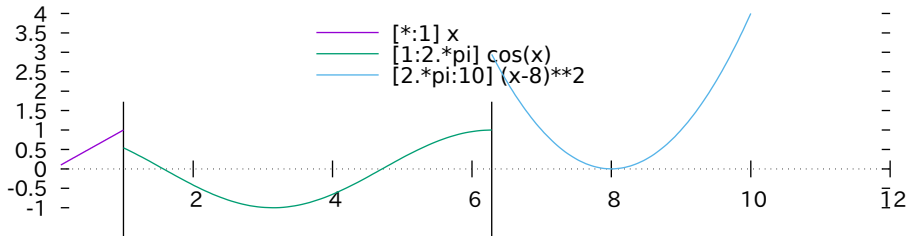




Piecewise function sampling along linear x

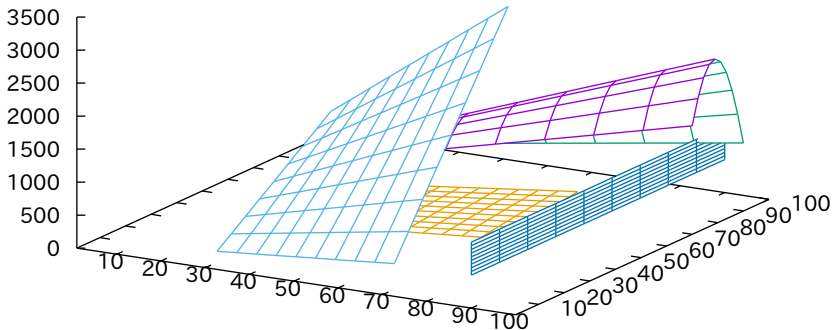


nonlinear (identity mapped) x

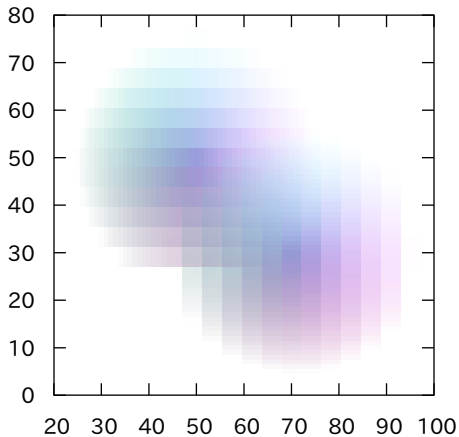


3D sampling range distinct from plot x/y range

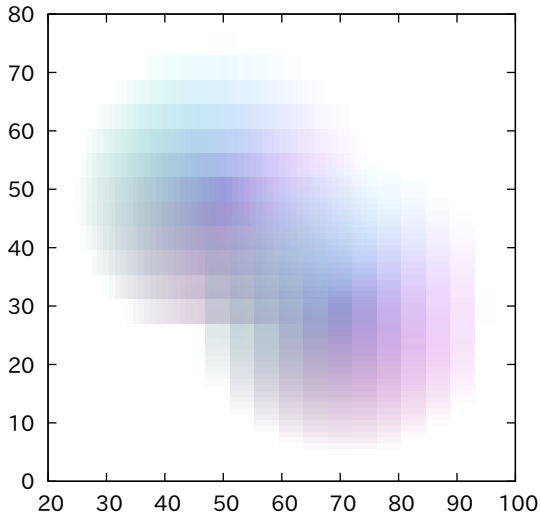
'++' using 1:2:(\$1*25.*sin(\$2/10))
[u=30:70][v=0:50] '++' using 1:2:(u*v)
[u=40:80][v=30:60] '++' using (u):(v):(u*sqrt(v))
[u=1:100][v=500:1000] '++' using (90):(u):(v)



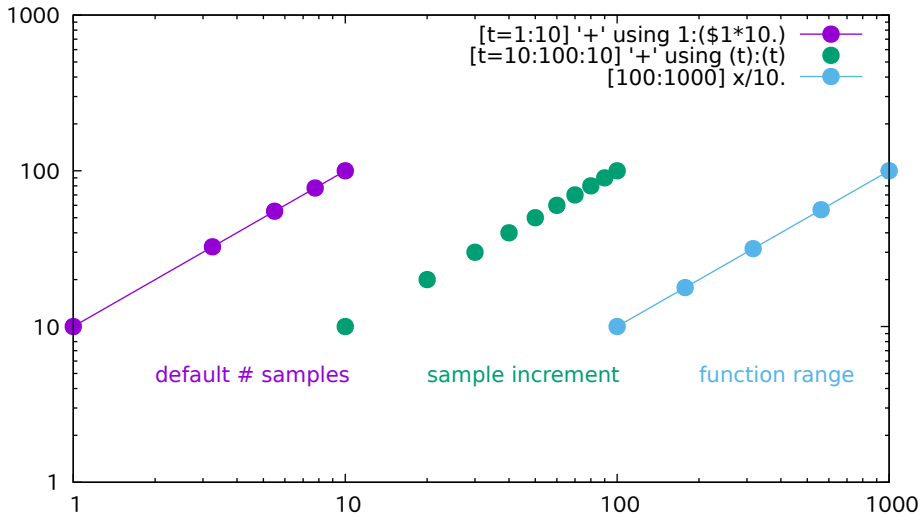
3D custom sampling on u and v using pseudofile '++'



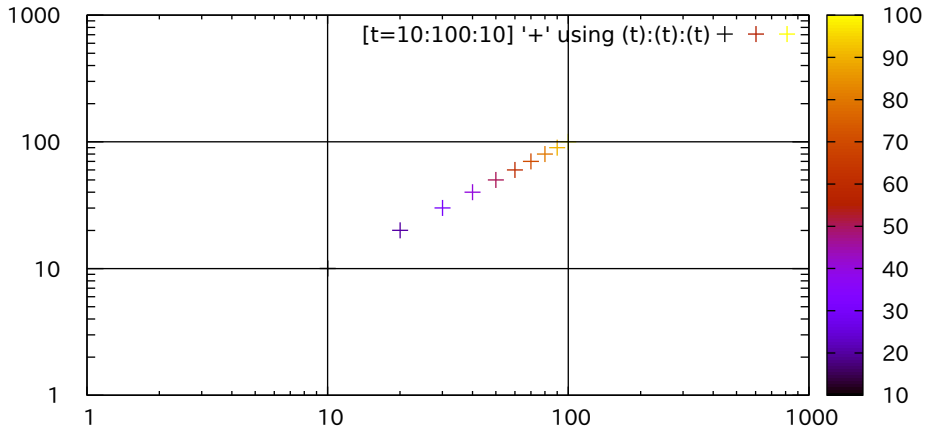
2D custom sampling on u and v using pseudofile '++'



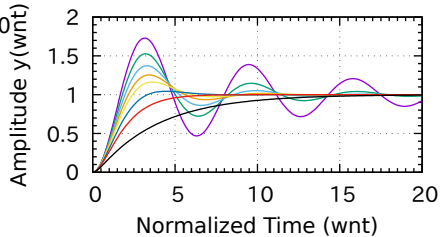
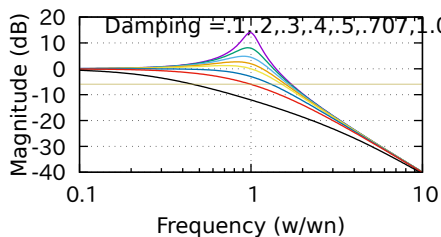
Sampling one dimension in 2D



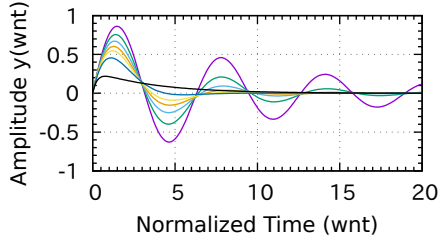
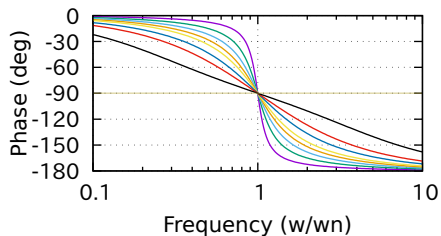
Sampling one dimension in 3D



Second Order System Transfer Function - Magnitude Second Order System - Unit Step Response

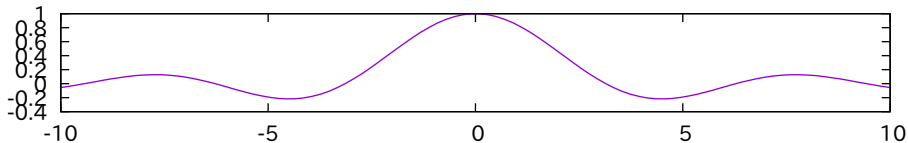


Second Order System Transfer Function - Phase Second Order System - Unit Impulse Response

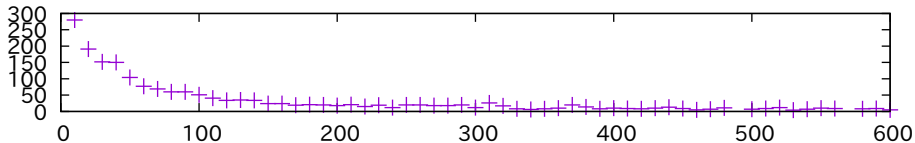


Multiplot layout 3, 1

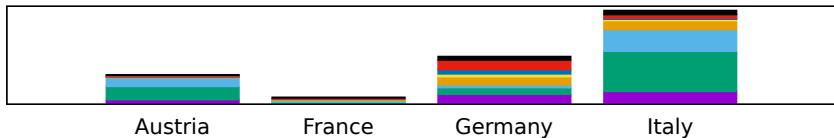
Plot 1



Plot 2

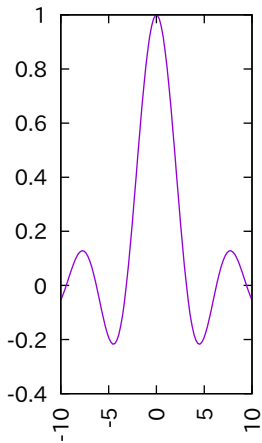


Plot 3

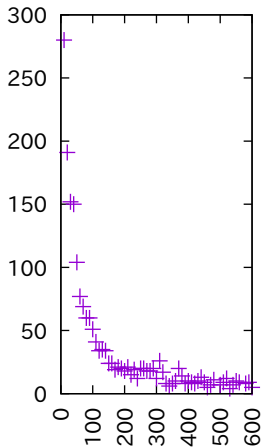


Multiplot layout 1, 3

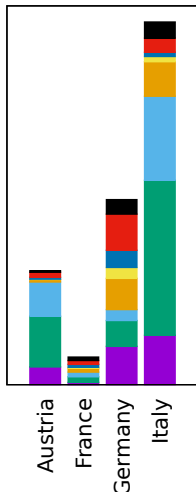
Plot 1



Plot 2

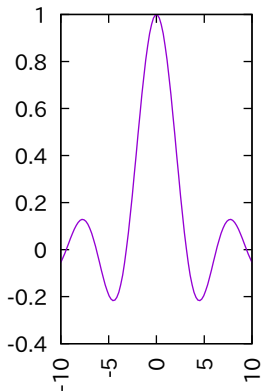


Plot 3

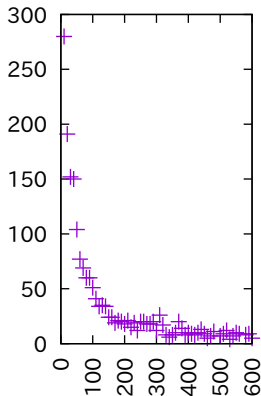


Same plot with a multi-line title
showing adjustment of plot area to accommodate it
Also note 'reset' command between plots 2 and 3

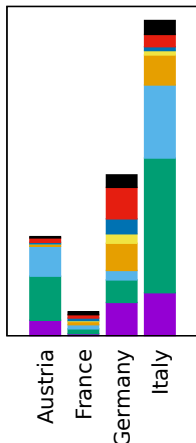
Plot 1



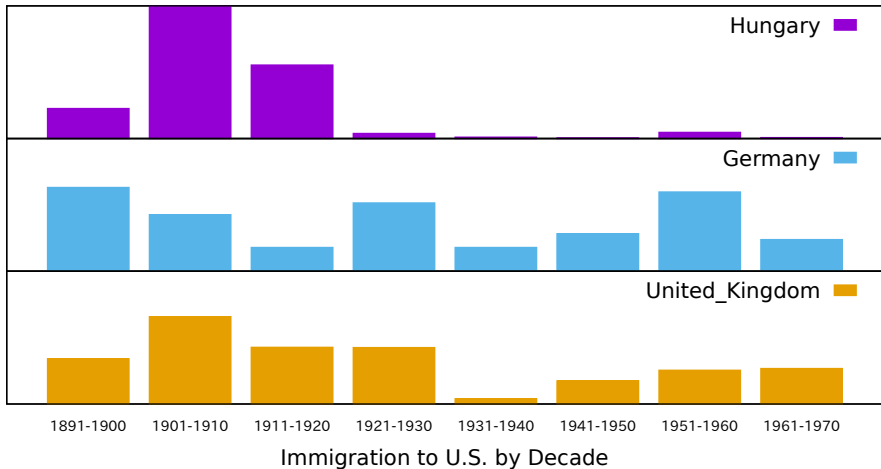
Plot 2



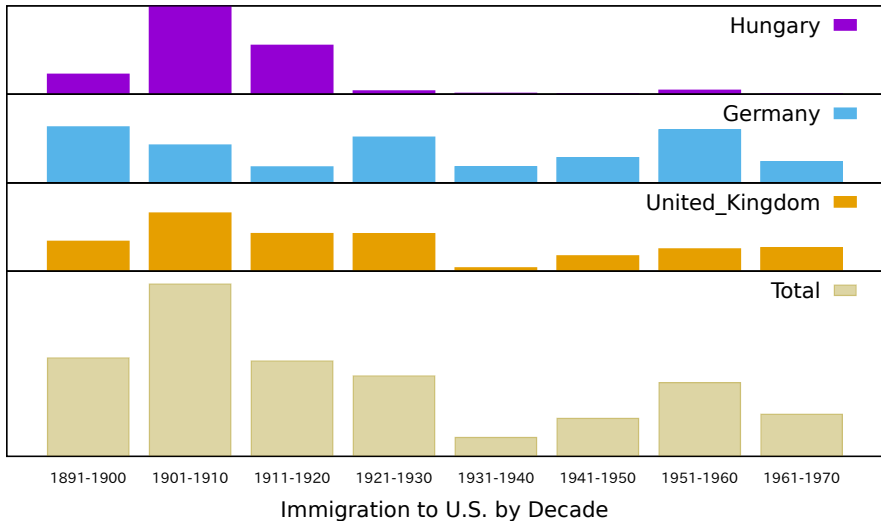
Plot 3



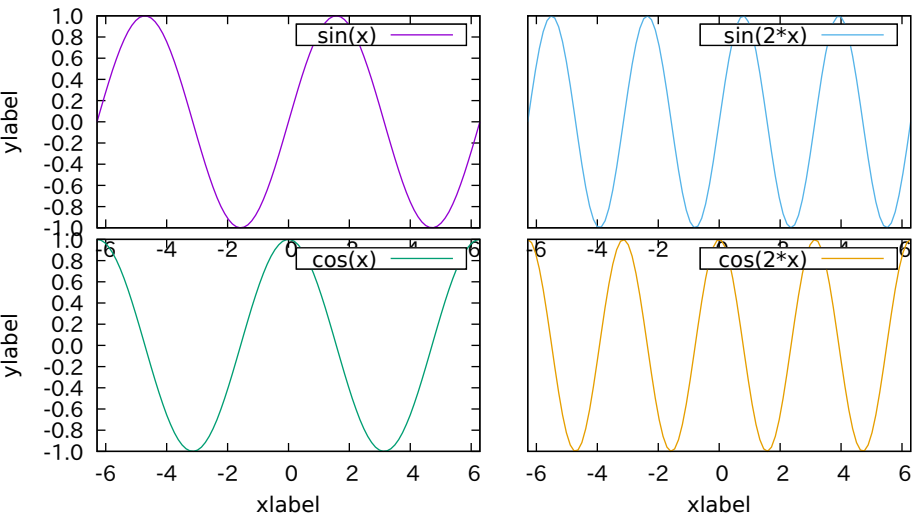
Auto-layout of stacked plots



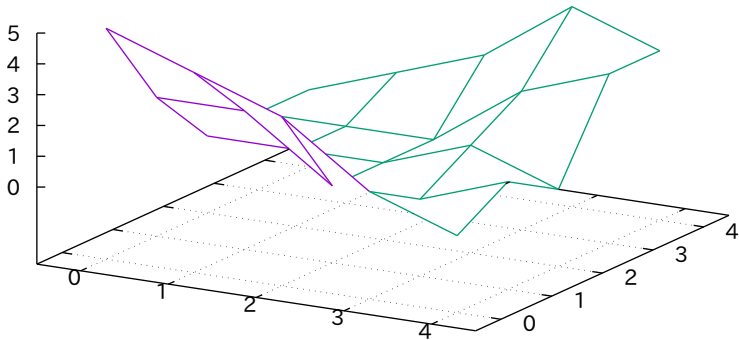
Expanding one of the plots to use additional space



Multiplot with explicit page margins



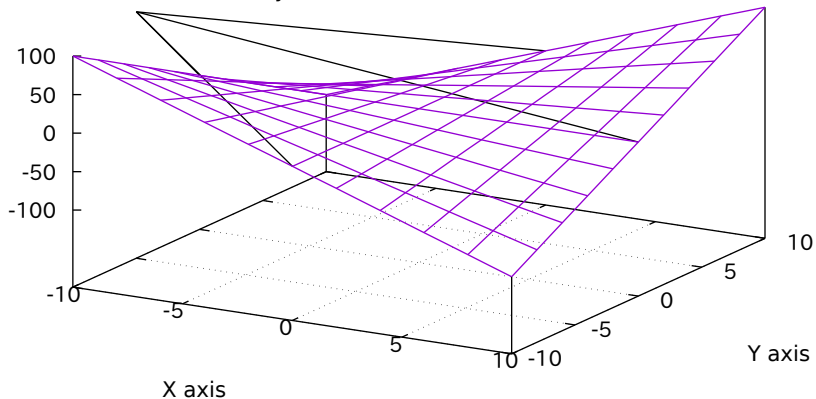
3D surface from a grid (matrix) of Z values



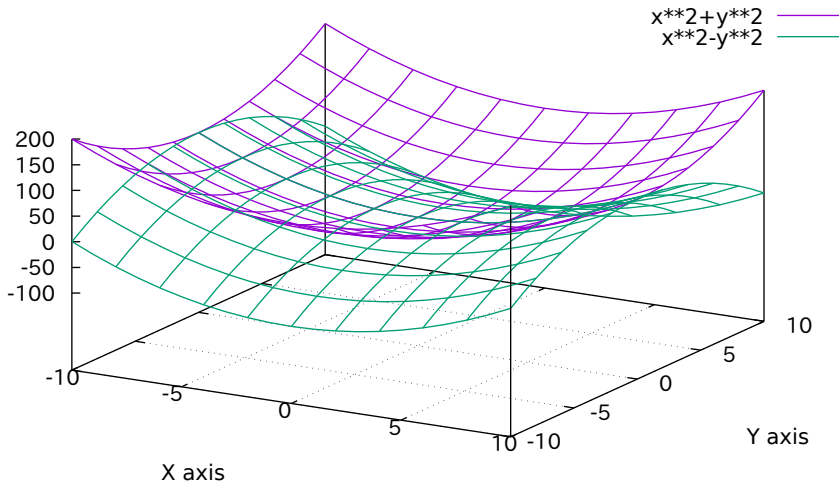
3D surface from a function

$x*y$ —

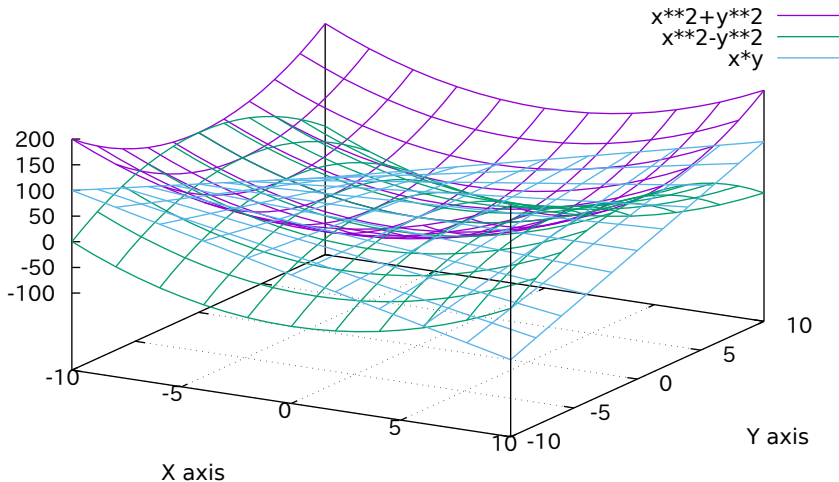
This is the surface boundary



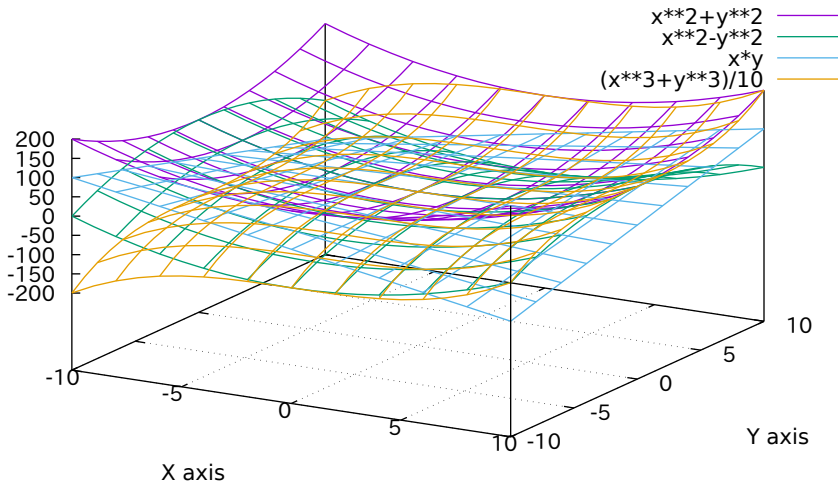
3D surface from a function



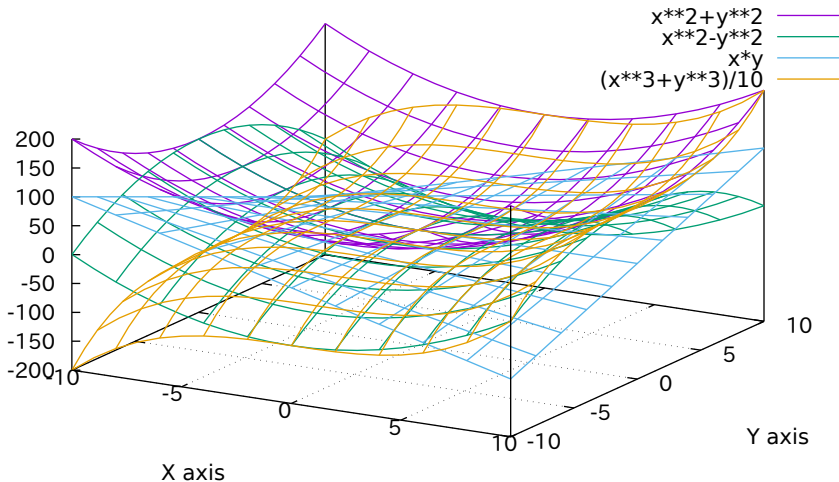
3D surface from a function



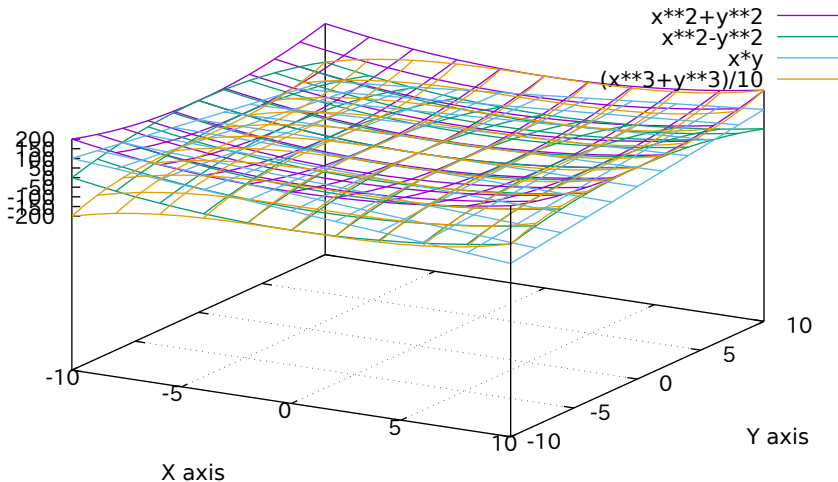
3D surface from a function



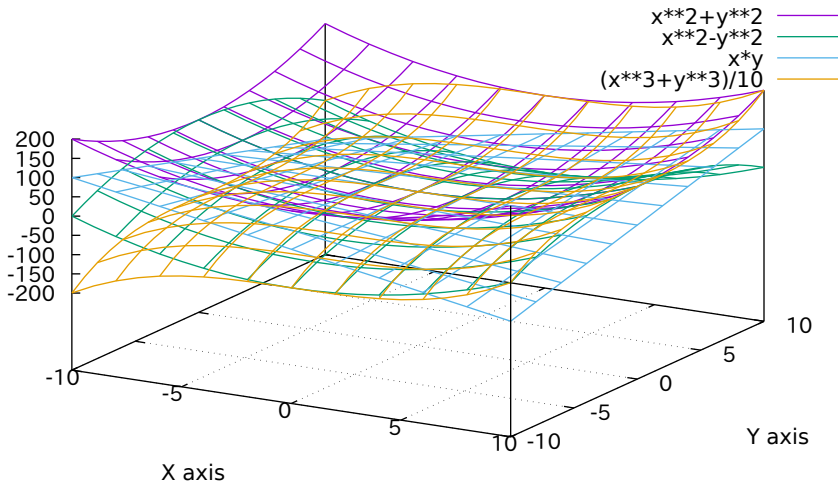
3D gnuplot demo (ticslevel = 0.0)



3D gnuplot demo (ticslevel = 2.0)

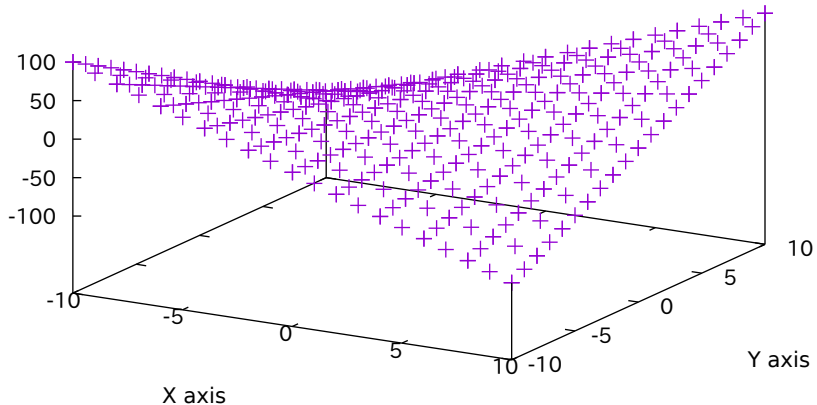


3D gnuplot demo (ticslevel = 0.5)

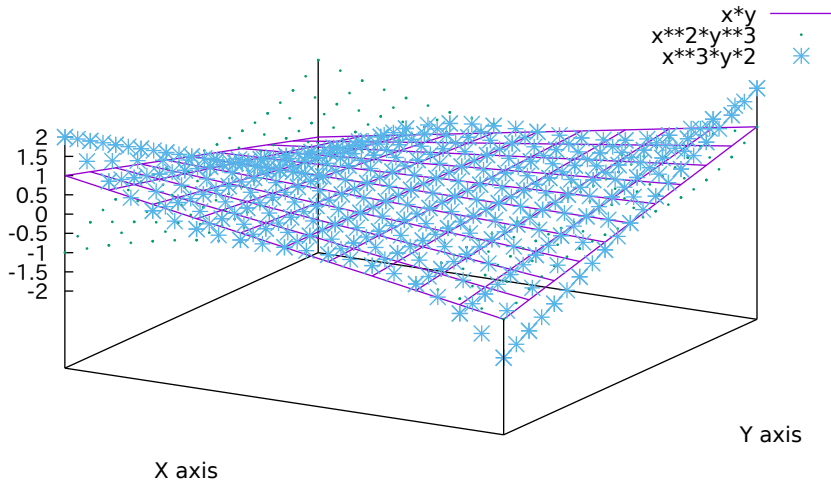


3D gnuplot demo

$x*y$ +

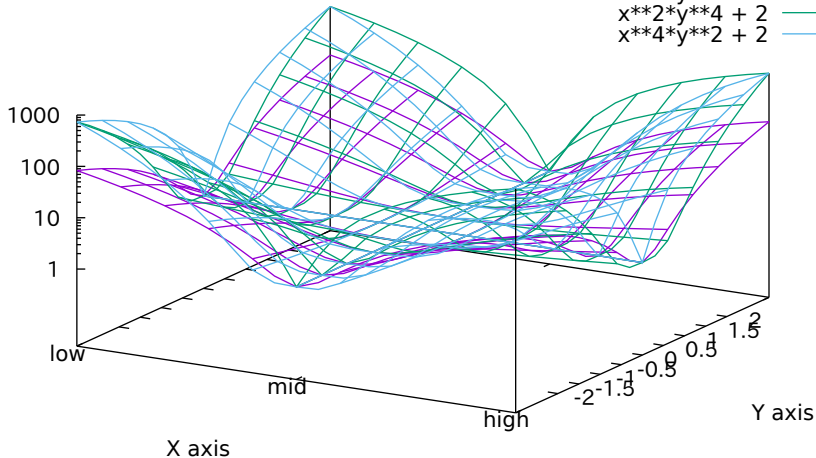


Surfaces with no grid or tics



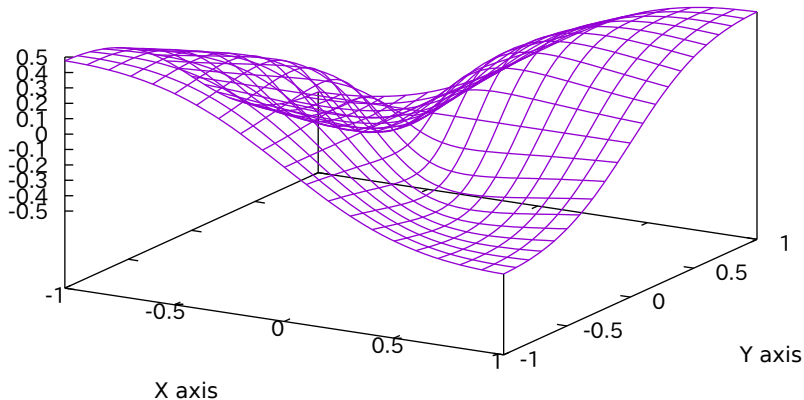
Surfaces with z log scale

$$\begin{aligned} x^{**2}y^{**2} + 2 & \text{ --- } \text{purple line} \\ x^{**2}y^{**4} + 2 & \text{ --- } \text{green line} \\ x^{**4}y^{**2} + 2 & \text{ --- } \text{blue line} \end{aligned}$$



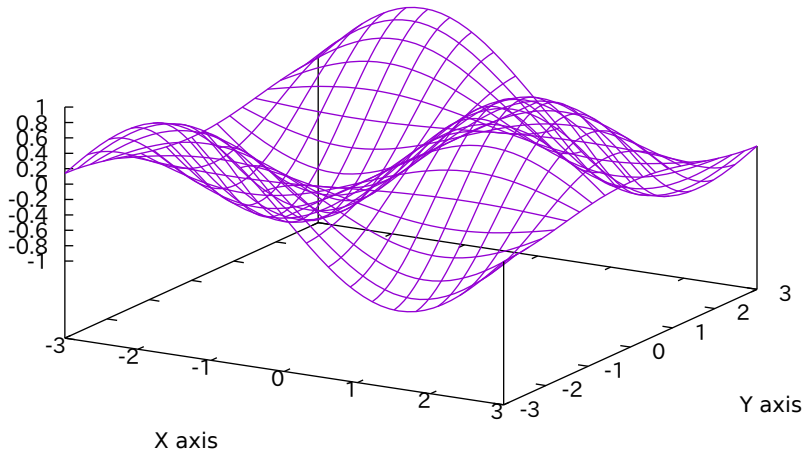
3D gnuplot demo

$$u*v / (u**2 + v**2 + 0.1)$$



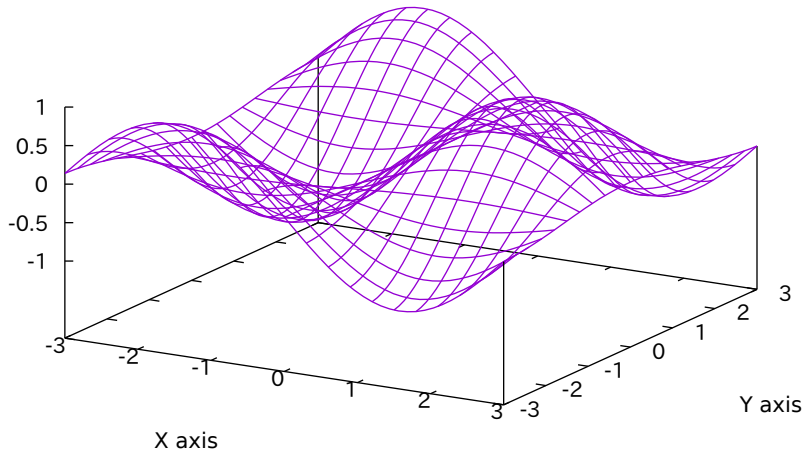
3D gnuplot demo

$\sin(x) * \cos(y)$ —



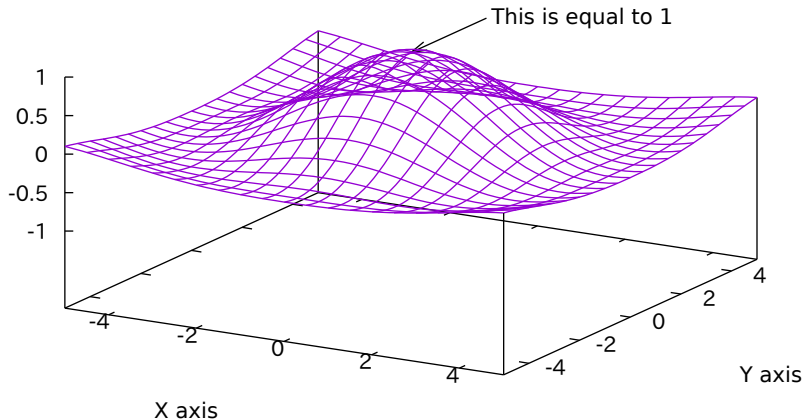
3D gnuplot demo

$\sin(x) * \cos(y)$ —



Sinc function

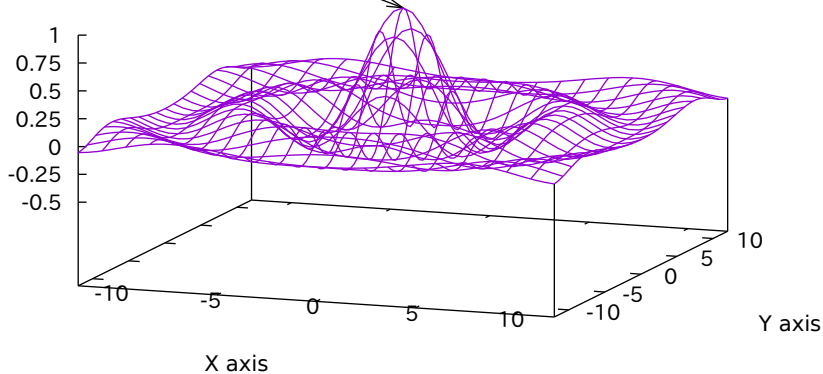
$\text{sinc}(u,v)$ —



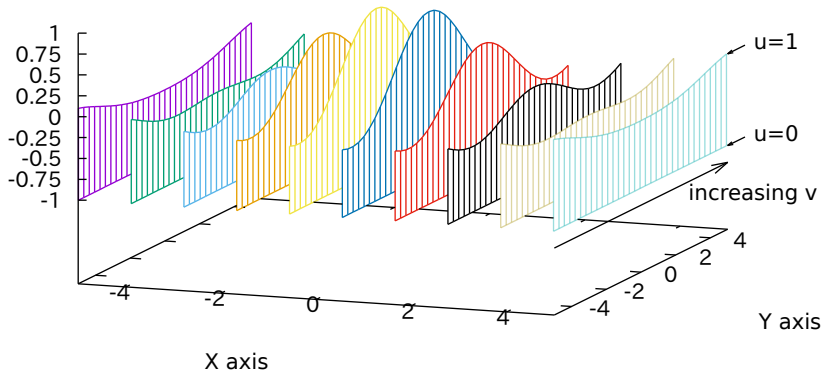
Sinc function

$\text{sinc}(u,v)$ —

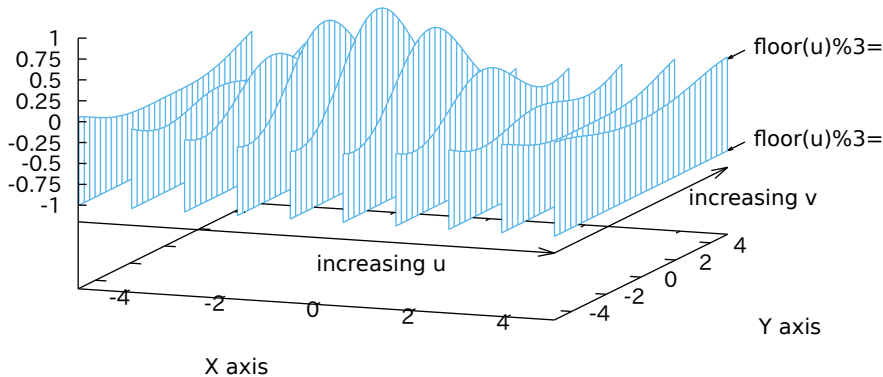
This is equal to 1



fence plot constructed with separate parametric surfaces

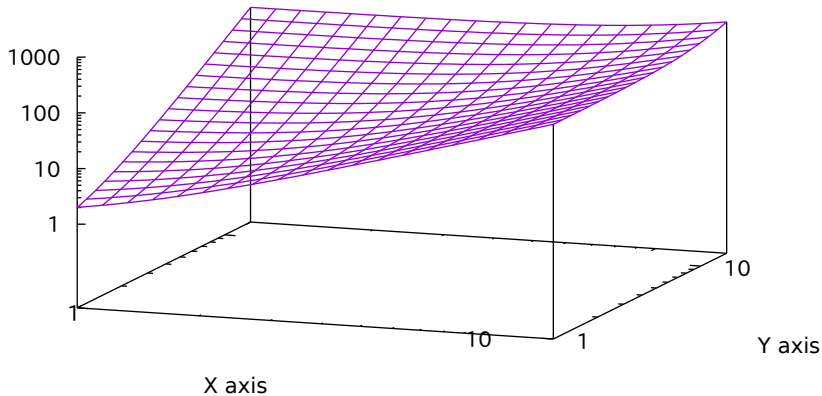


"fence plot" using single parametric surface with undefined points



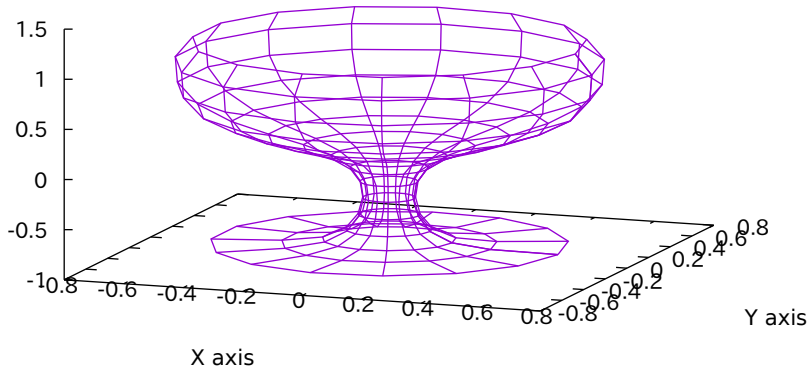
This has logarithmic scale

$$x^{**2}+y^{**2}$$



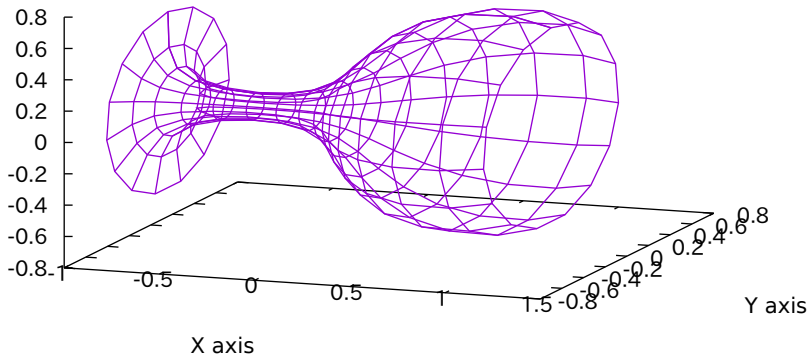
Data grid plotting

"glass.dat" —

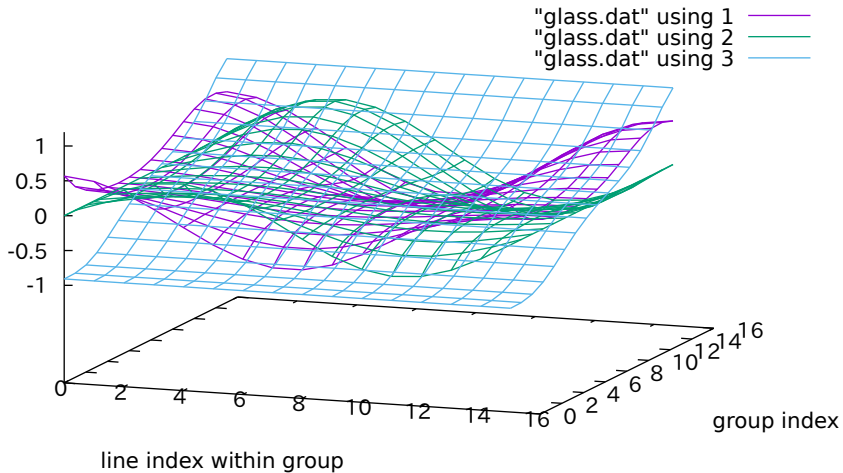


Data grid plotting

"glass.dat" using 3:2:1

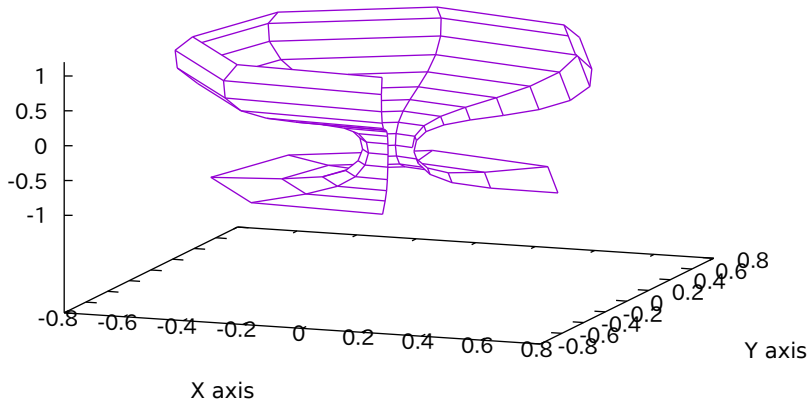


Data grid plotting



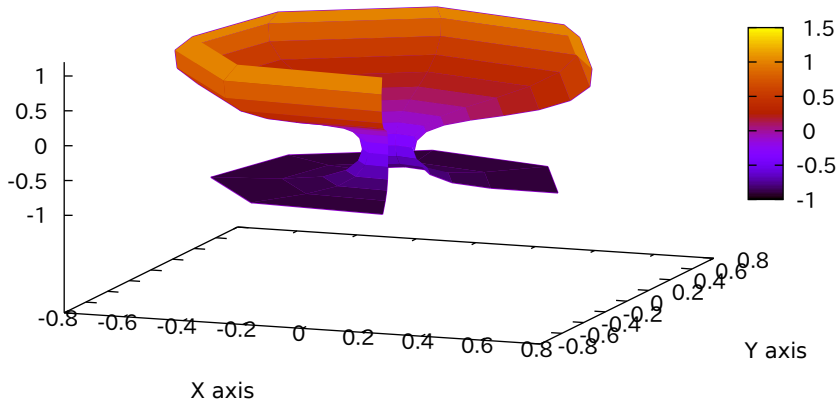
plot of part of a data file

'glass.dat' every 2::0::12



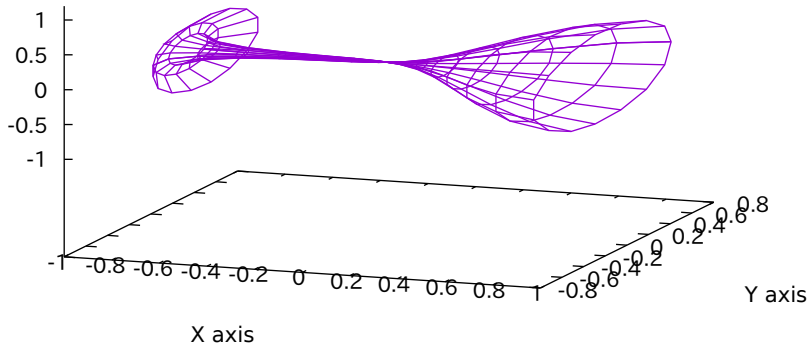
plot with "set pm3d" (implemented with some terminals)

'glass.dat' every 2::0::12



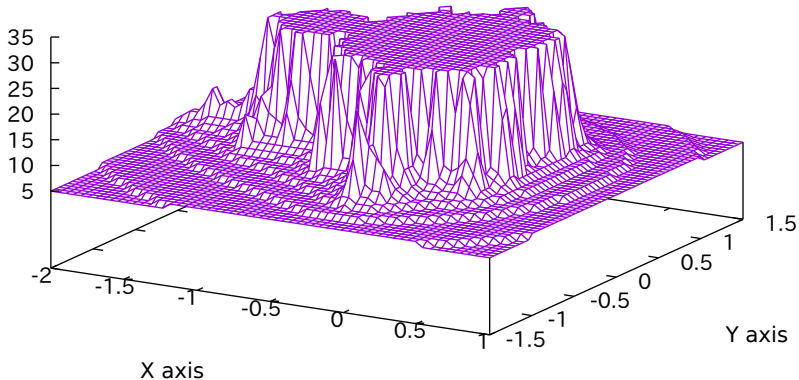
Test of spherical coordinates

"glass.dat" —

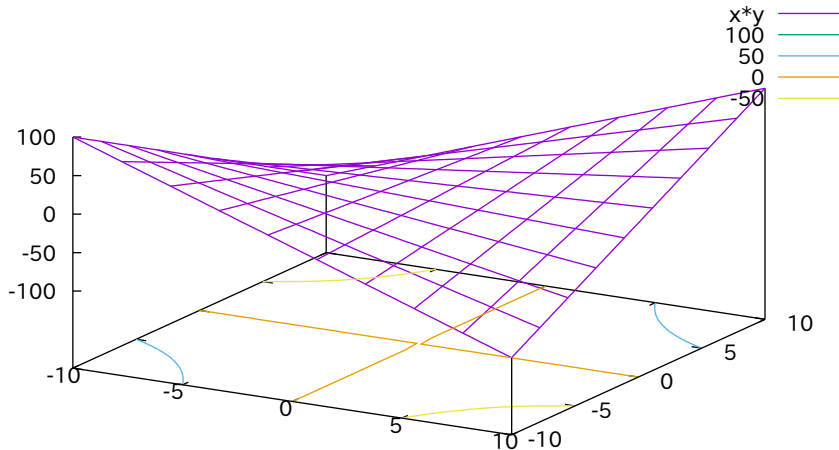


Mandelbrot function

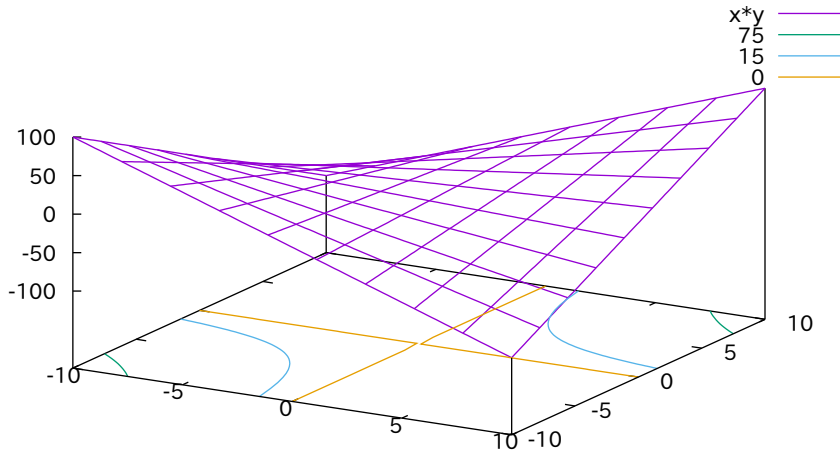
`mand({0,0},compl(x,y),30)` —



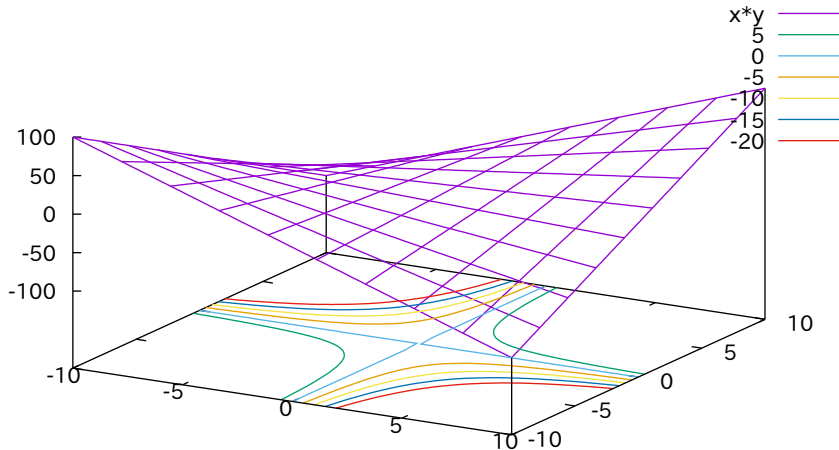
Demo of specifying discrete contour levels - default contours



3 discrete contours at 0 15 75

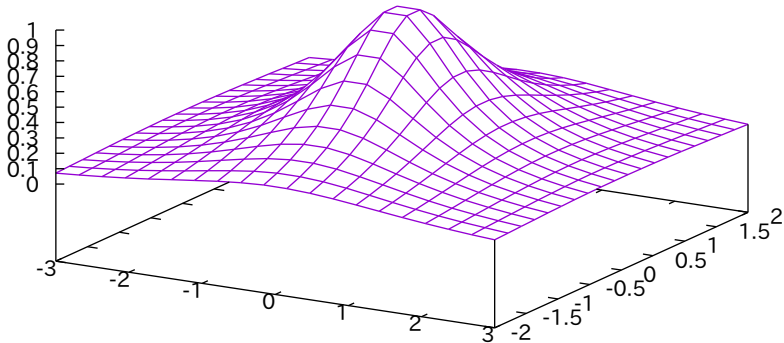


9 incremental contours starting at -20, stepping by 5



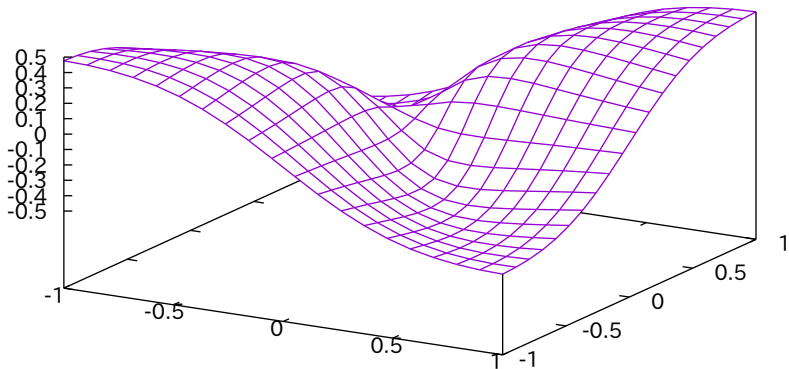
Hidden line removal of explicit surfaces

$$1 / (x^2 + y^2 + 1)$$



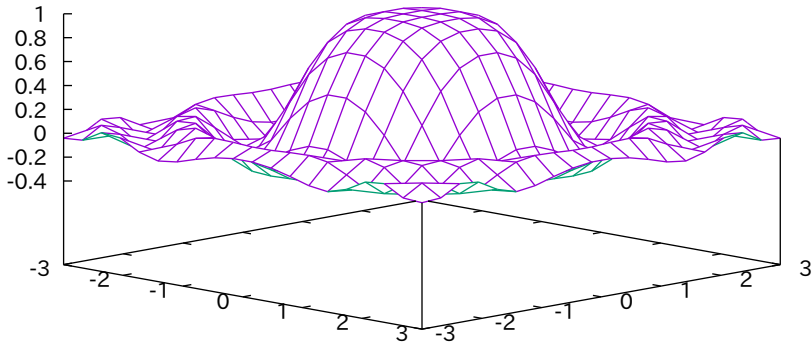
Hidden line removal of explicit surfaces

$$x*y / (x**2 + y**2 + 0.1) \quad \text{—}$$



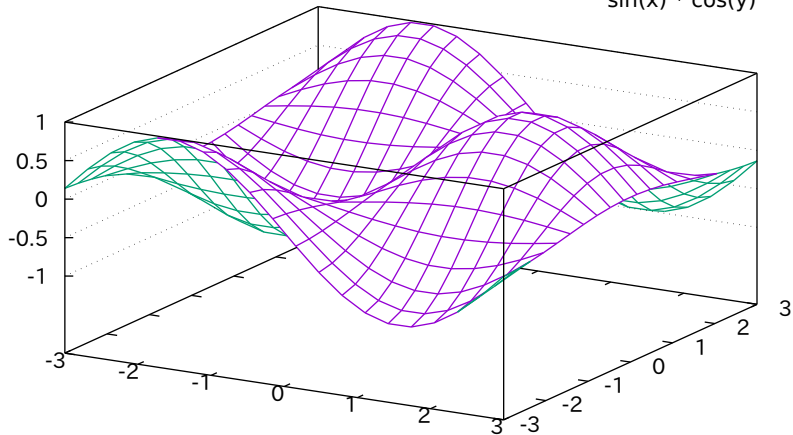
Hidden line removal of explicit surfaces

$$\sin(x^2 + y^2) / (x^2 + y^2) \quad \text{— purple line}$$

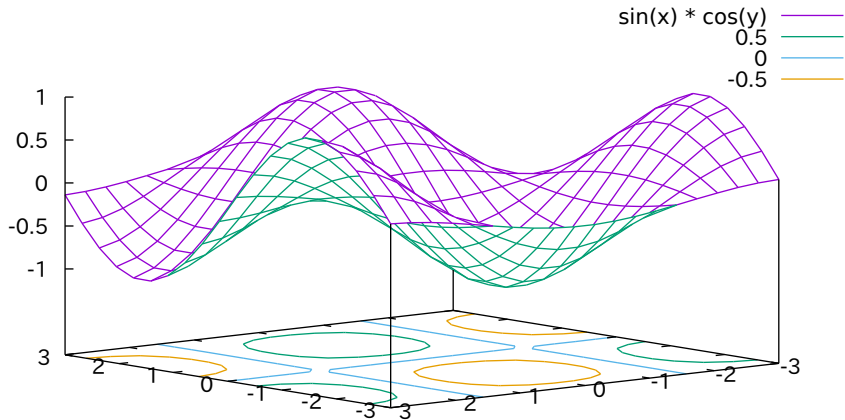


Hidden line removal of explicit surfaces

$\sin(x) * \cos(y)$ —

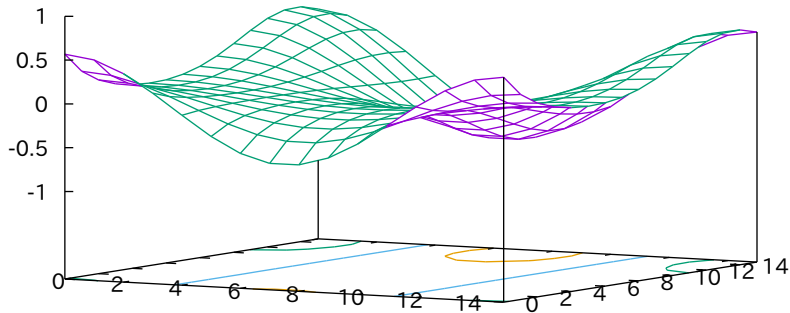


Hidden line removal of explicit surfaces

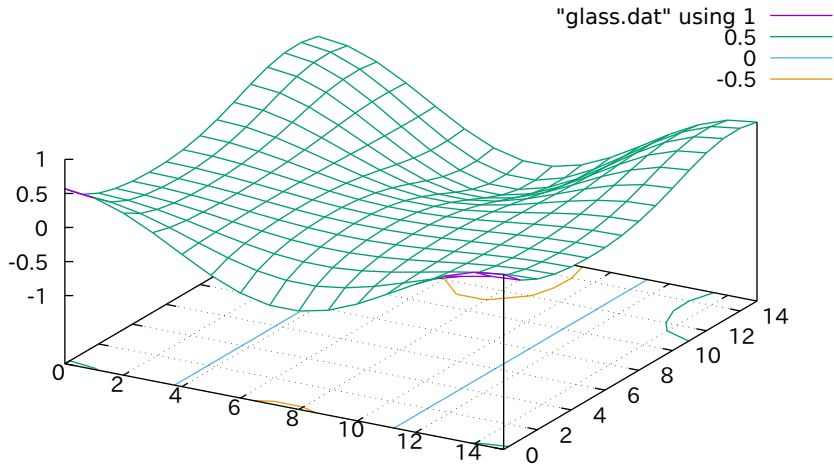


Hidden line removal of explicit surfaces

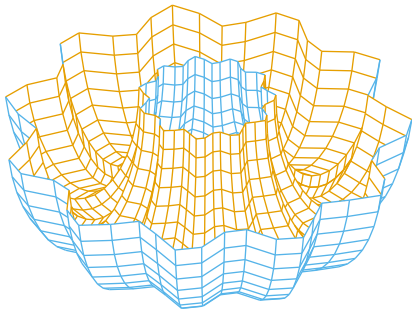
"glass.dat" using 1
0.5
0
-0.5



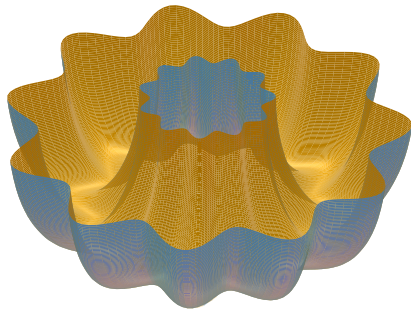
Hidden line removal of explicit surfaces



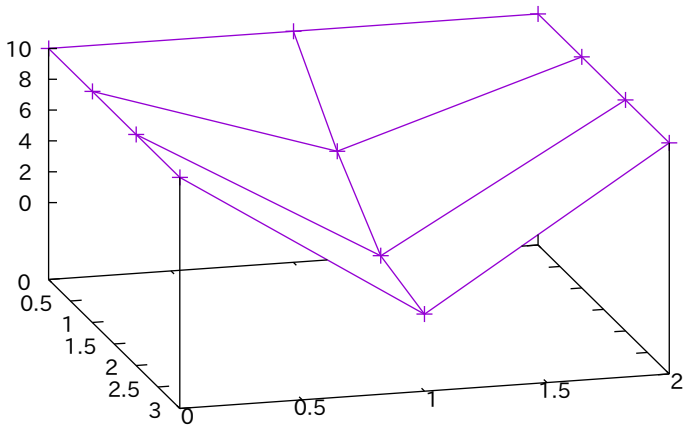
hidden3d 2-color surface



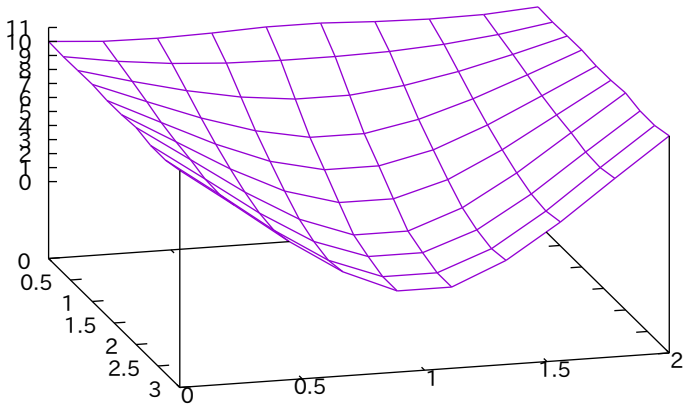
pm3d 2-color surface



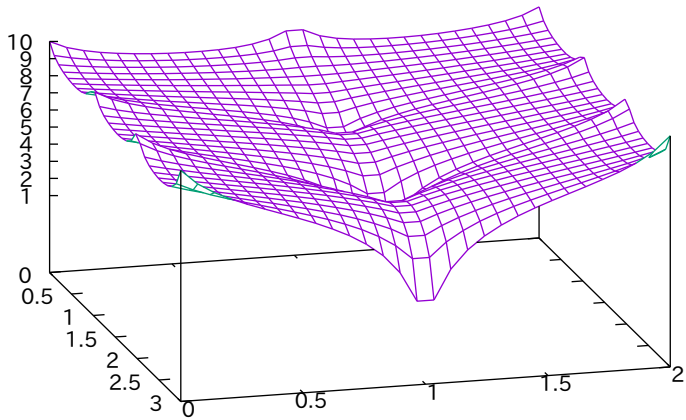
The Valley of the Gnu



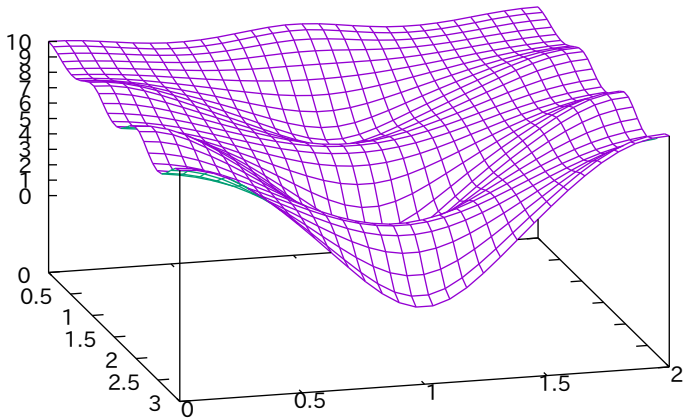
dgrid3d splines



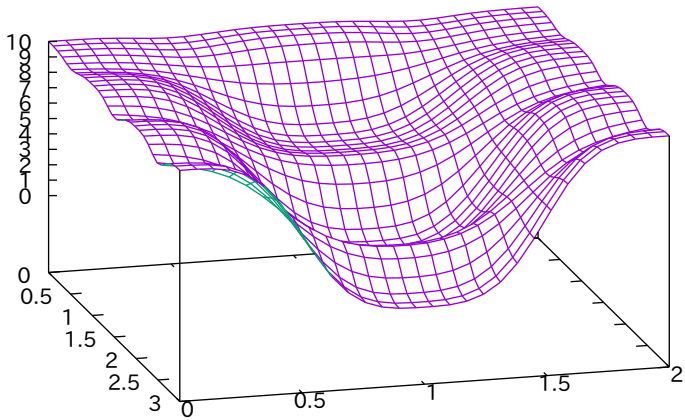
dgrid3d 30,30 qnorm 1



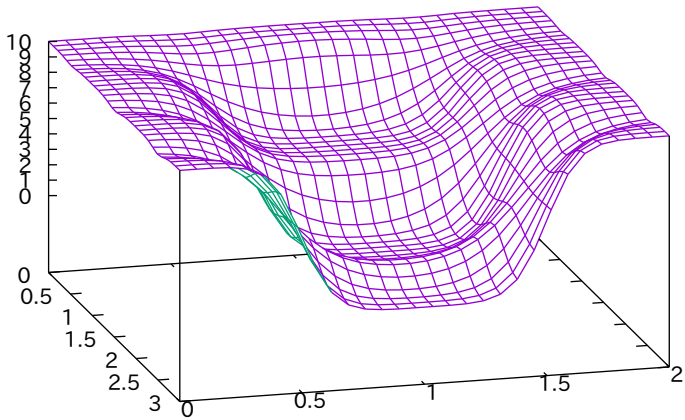
dgrid3d 30,30 qnorm 2



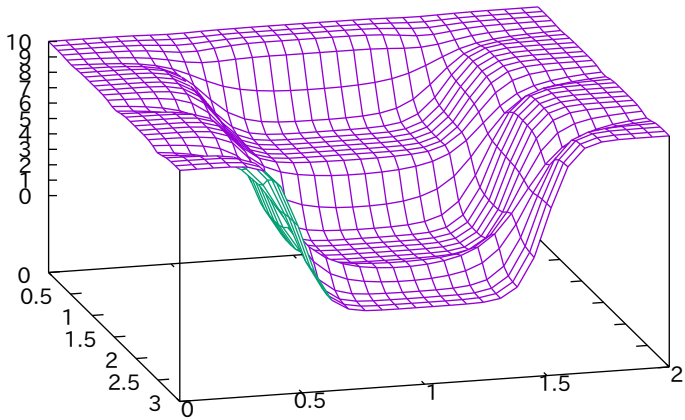
dgrid3d 30,30 qnorm 3



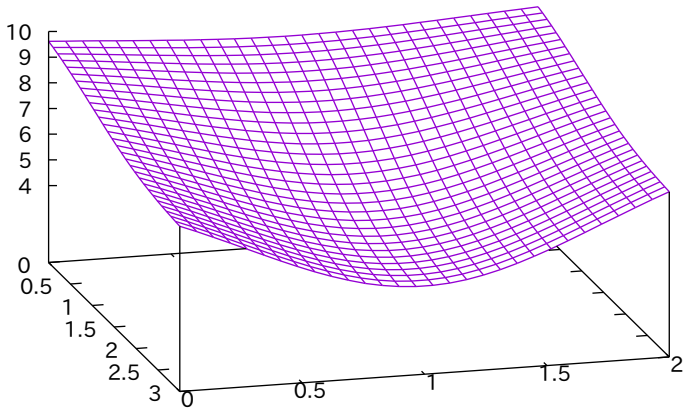
dgrid3d 30,30 qnorm 4



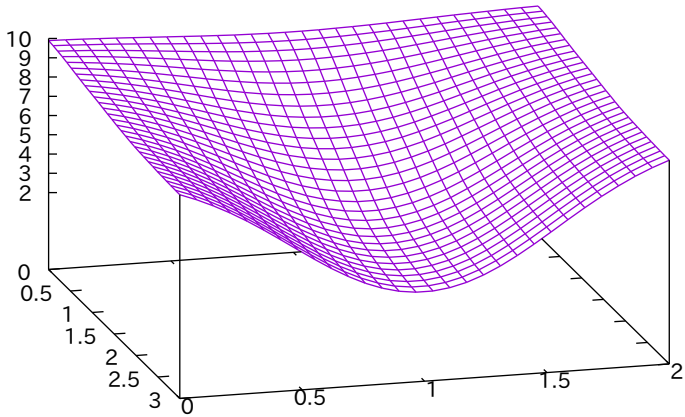
dgrid3d 30,30 qnorm 5



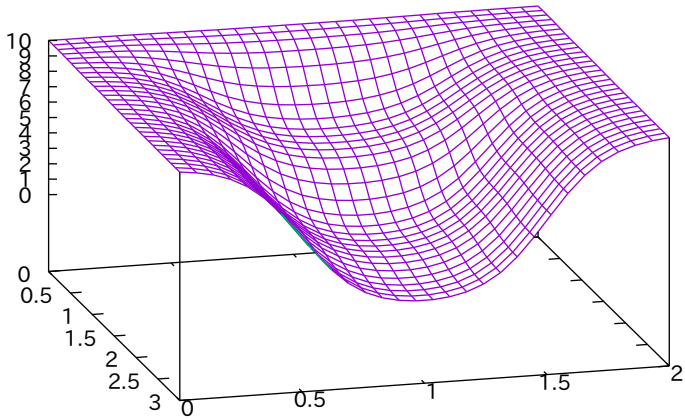
dgrid3d 30,30 gauss 1



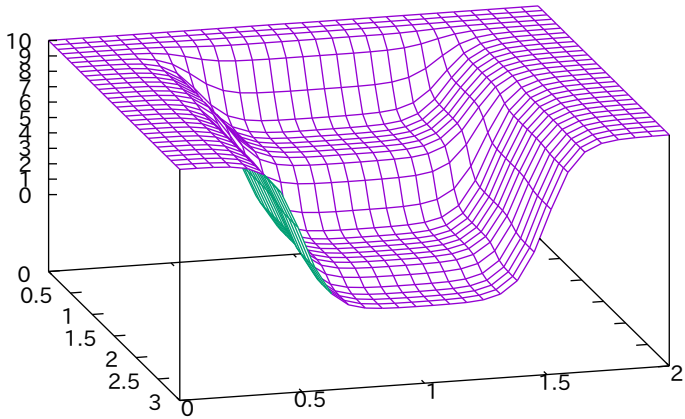
dgrid3d 30,30 gauss .75



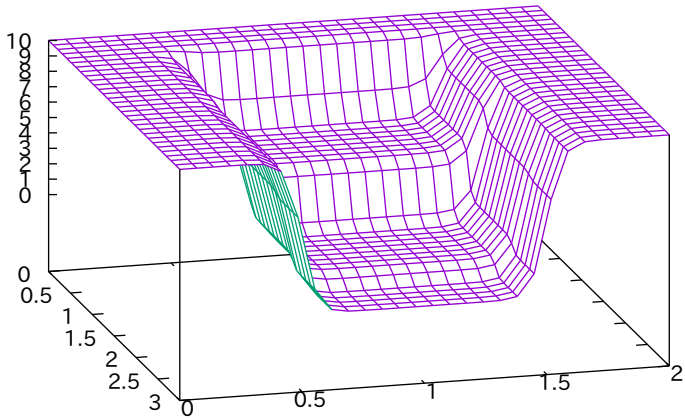
dgrid3d 30,30 gauss .5



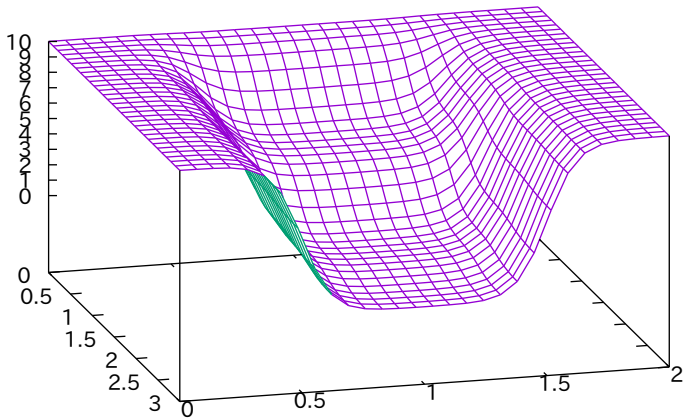
dgrid3d 30,30 gauss .35



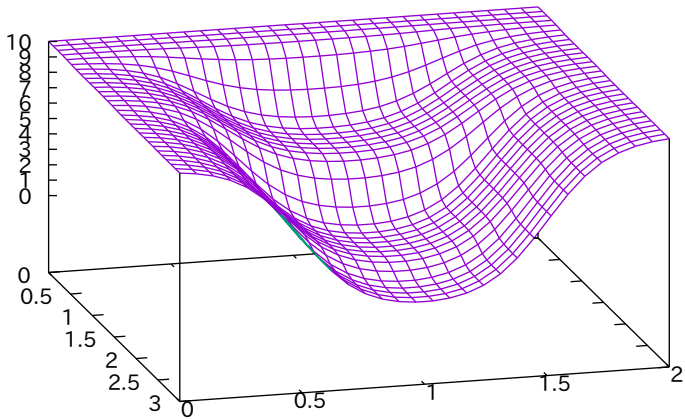
dgrid3d 30,30 gauss .25



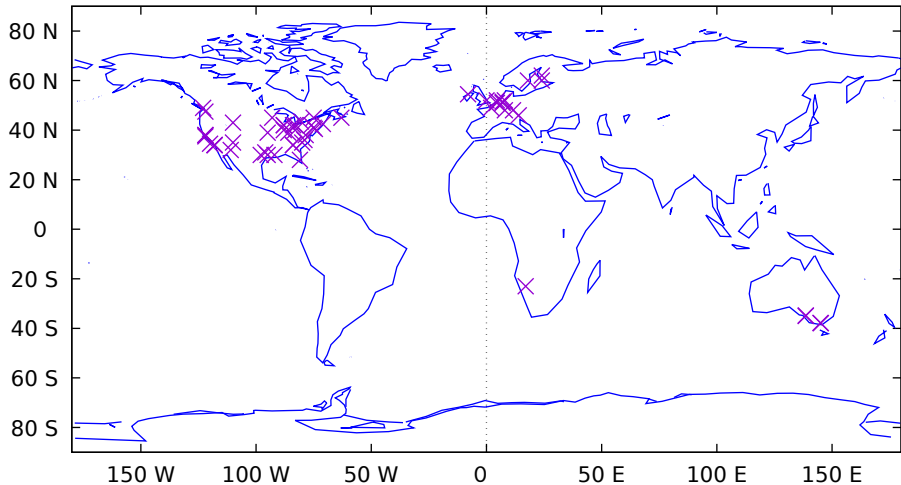
dgrid3d 30,30 gauss .5,.35



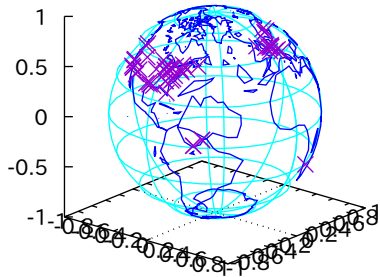
dgrid3d 30,30 gauss .35,.5



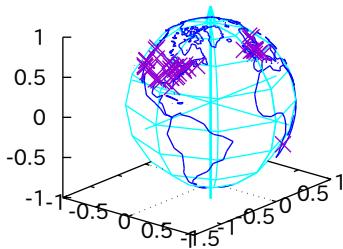
Gnuplot Correspondences geographic coordinate system



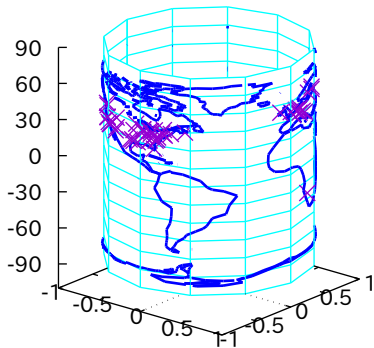
3D version using spherical coordinate system



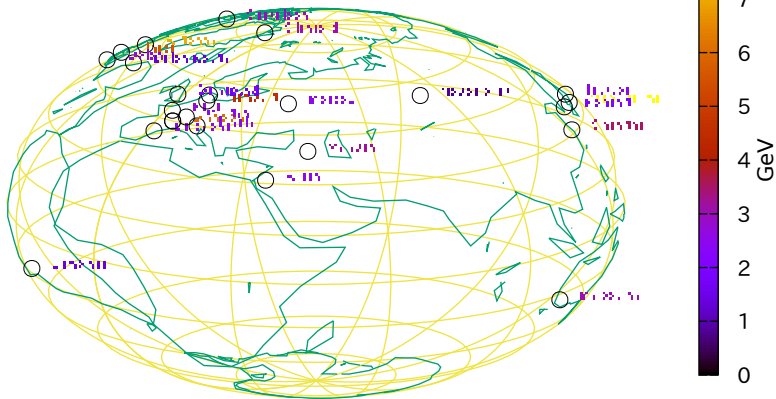
3D solid version with hidden line removal



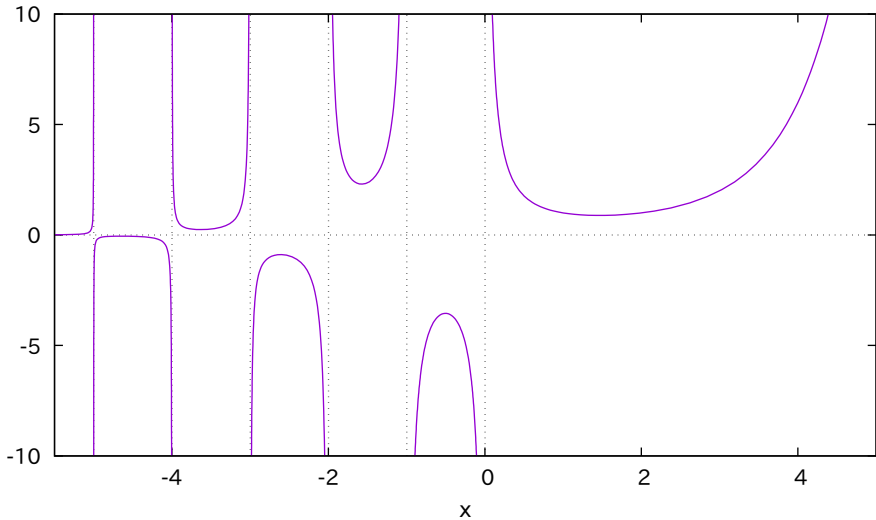
3D version using cylindrical coordinate system



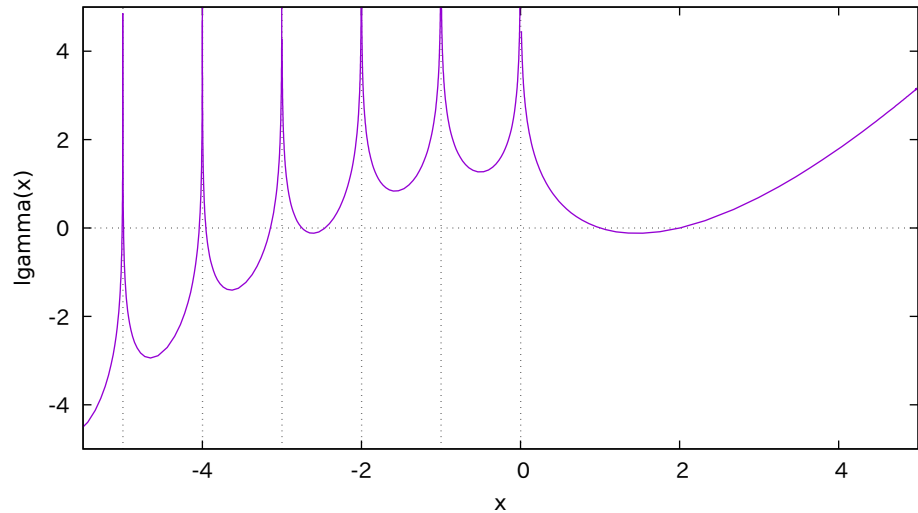
Labels colored by GeV plotted in spherical coordinate system



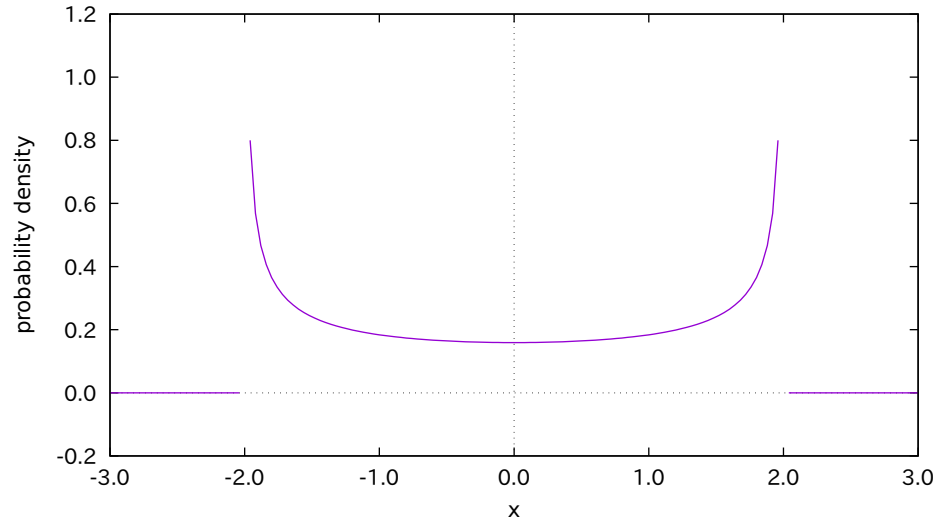
Gamma function Γ , very useful function for probability



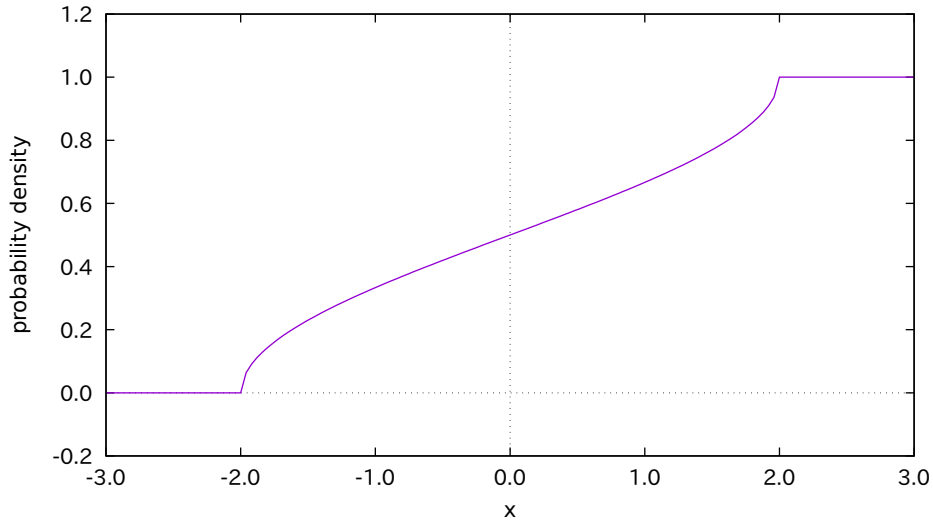
log gamma function, similarly very useful function



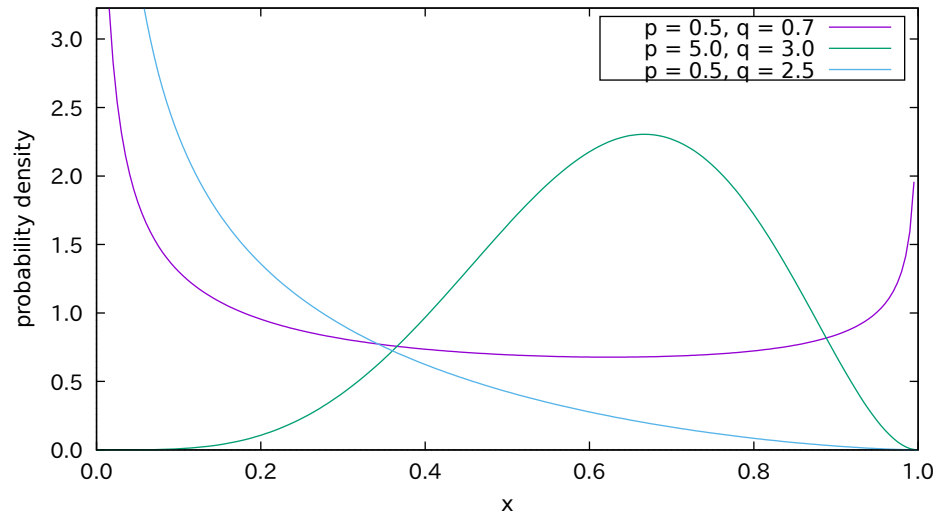
arcsin PDF with $r = 2.0$



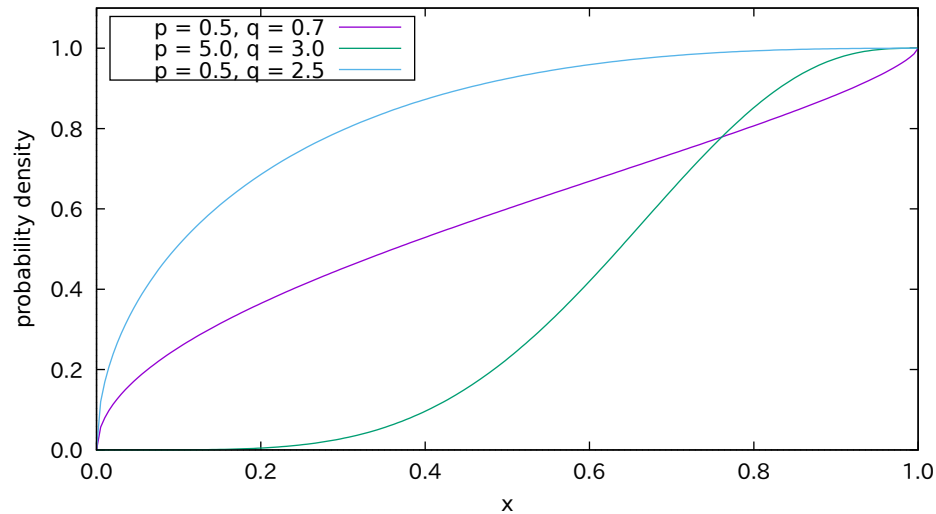
arcsin CDF with $r = 2.0$



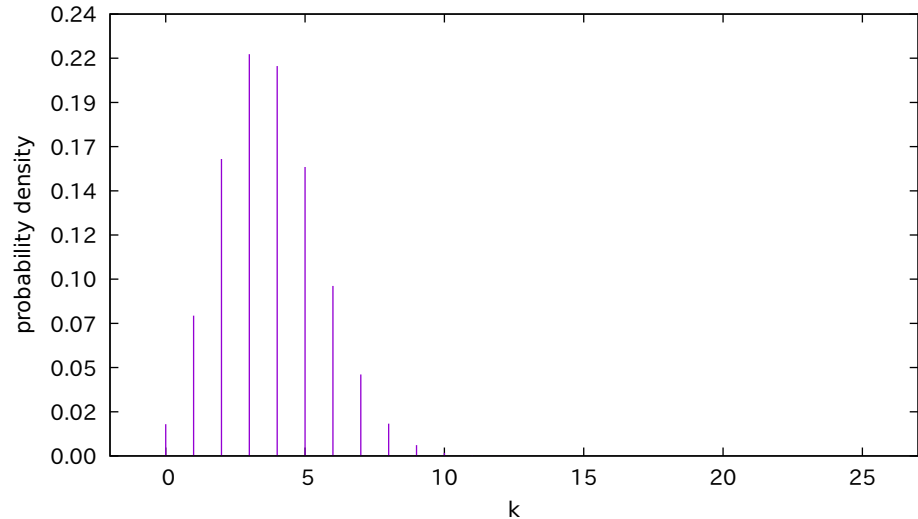
beta PDF



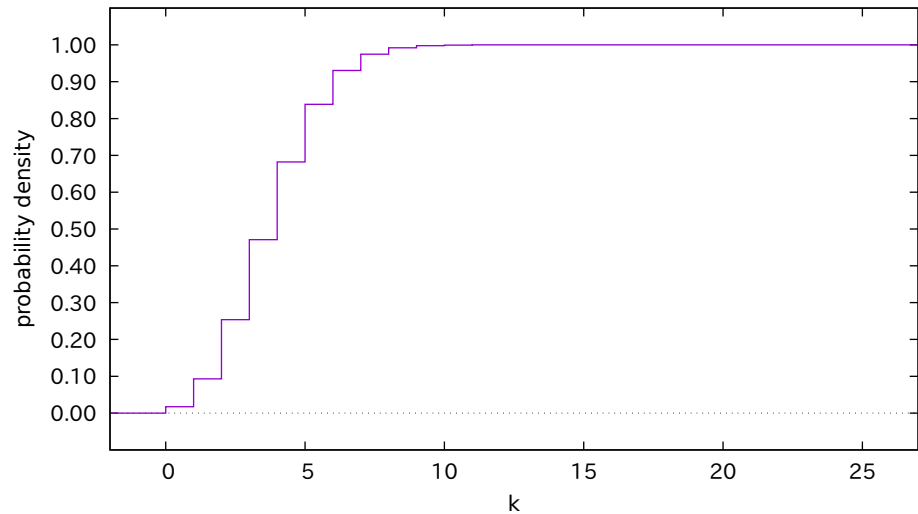
incomplete beta CDF



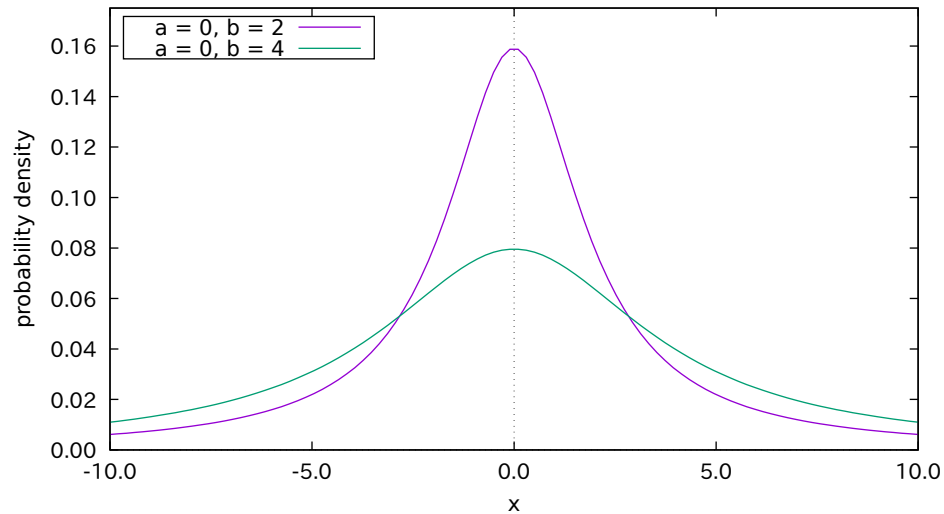
binomial PDF with $n = 25$, $p = 0.15$



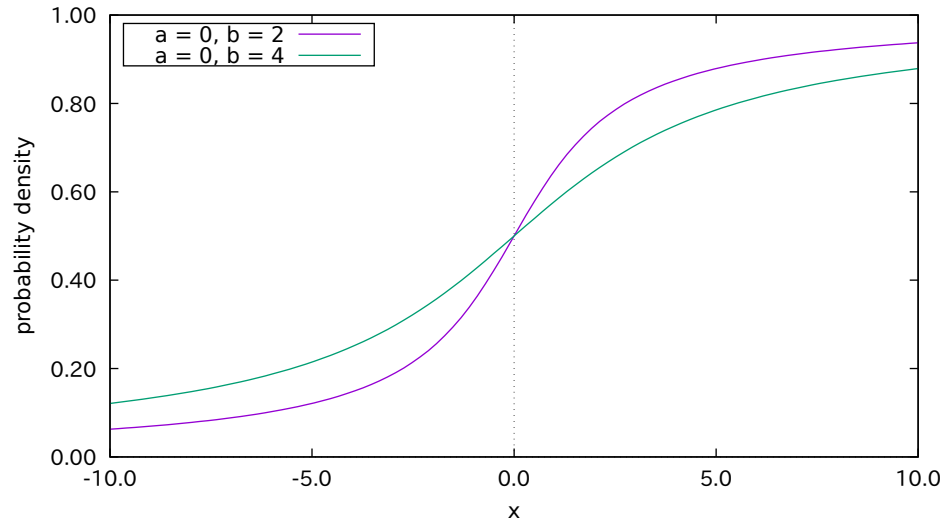
binomial CDF with $n = 25$, $p = 0.15$

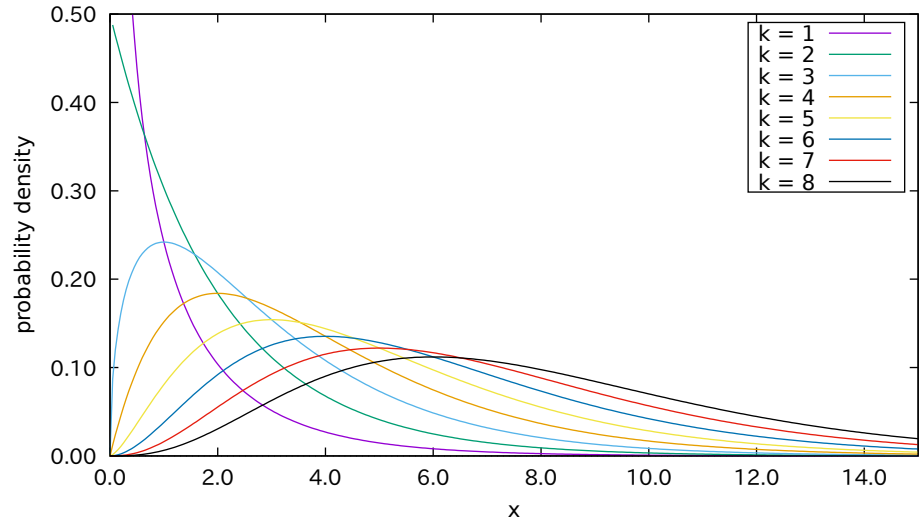


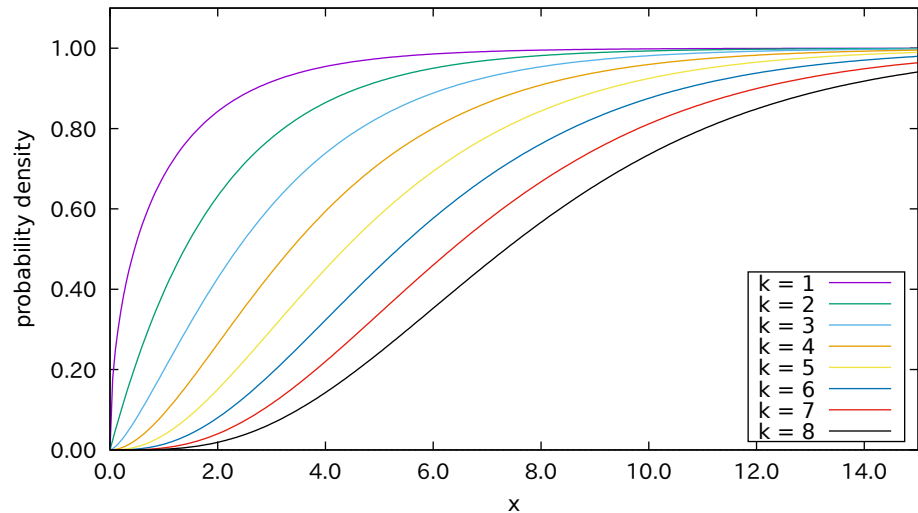
Cauchy PDF



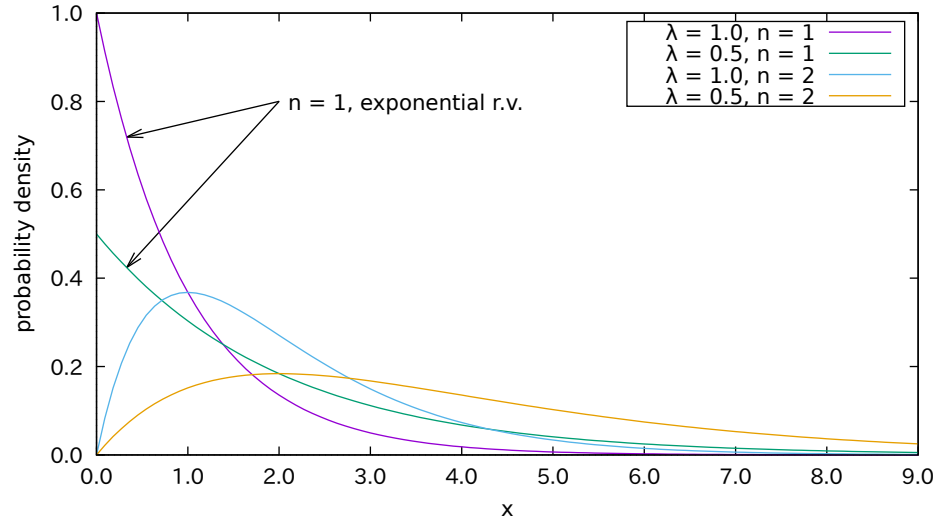
Cauchy CDF



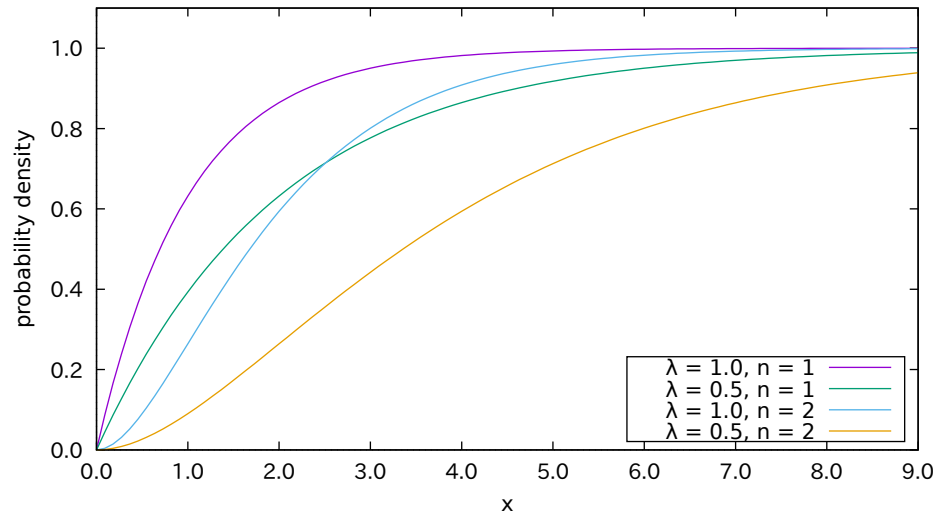
Chi-square χ^2 PDF

Chi-square χ^2 CDF

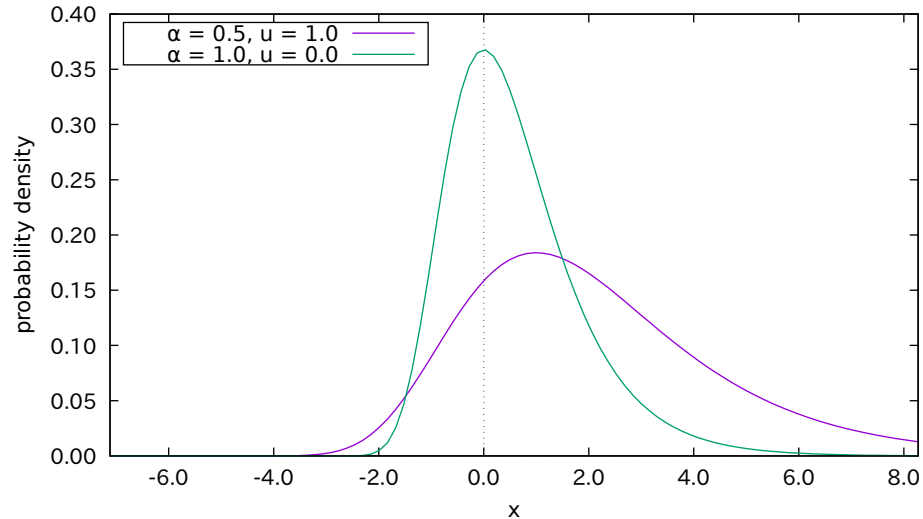
Erlang PDF



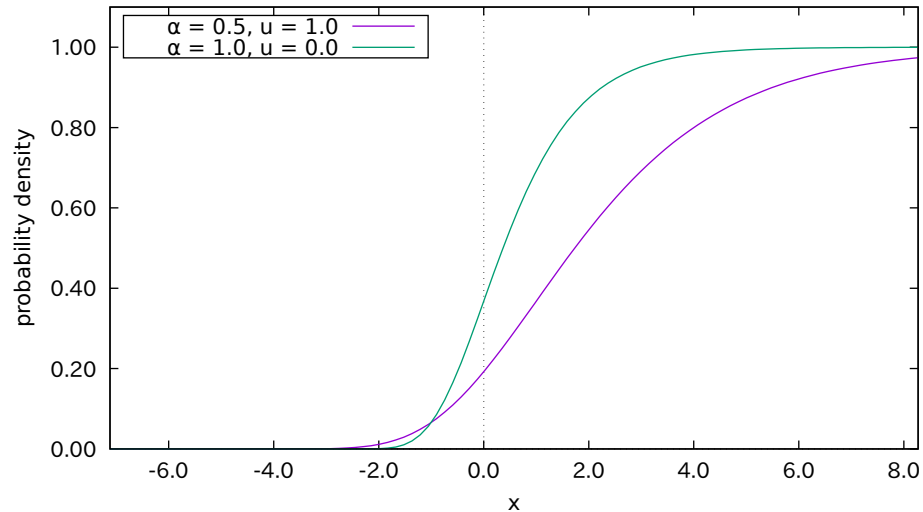
Erlang CDF



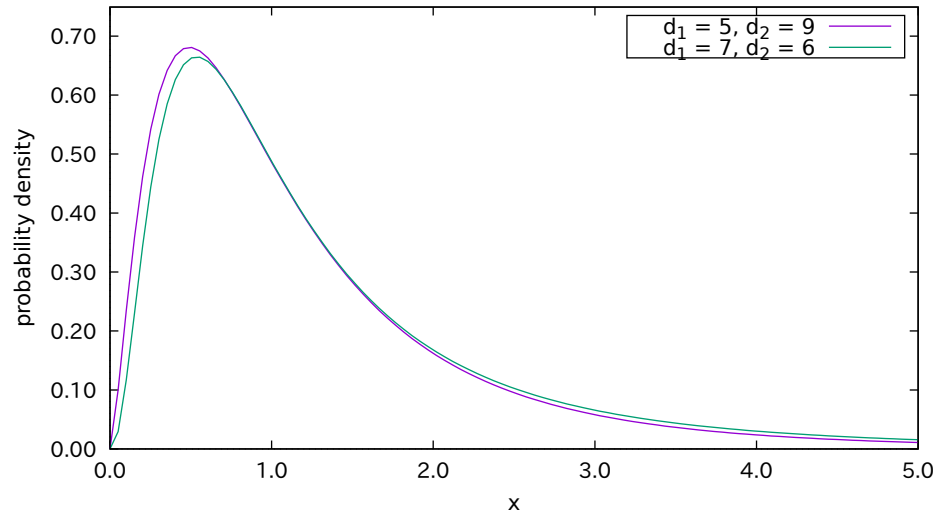
extreme PDF



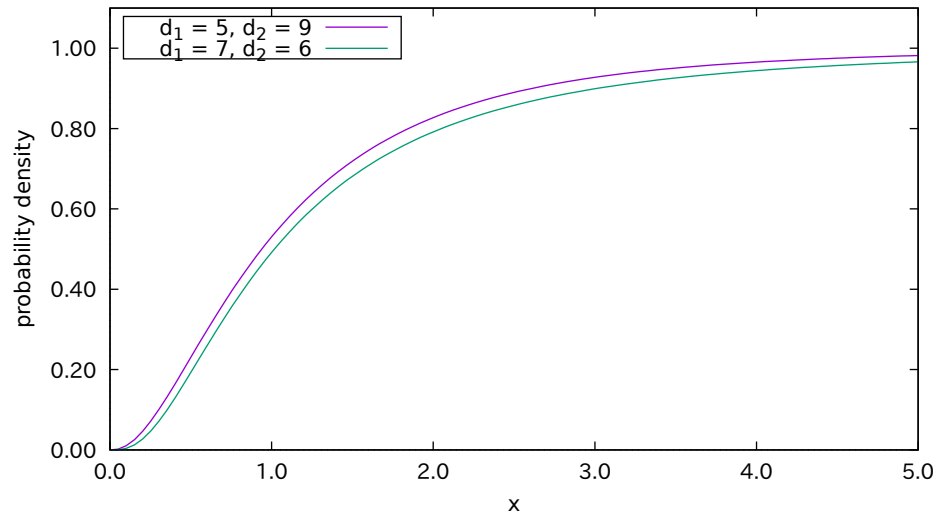
extreme CDF

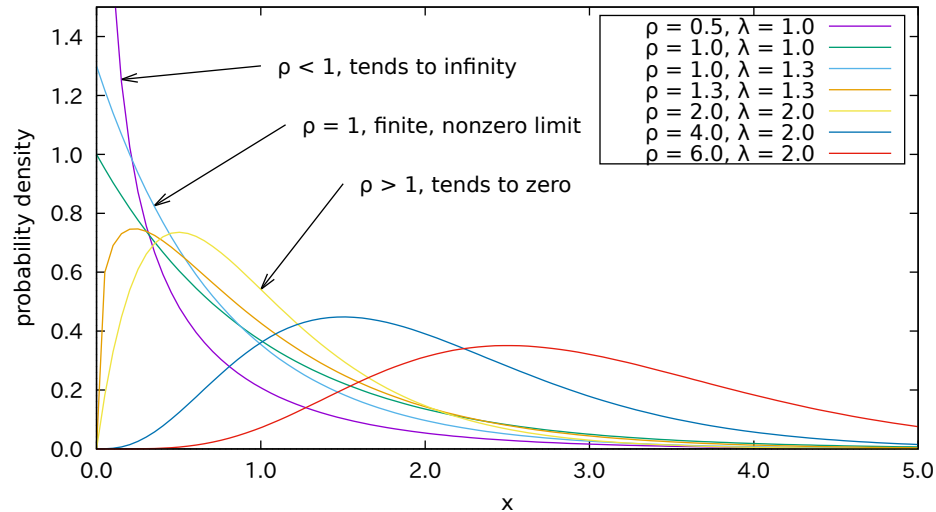


F PDF

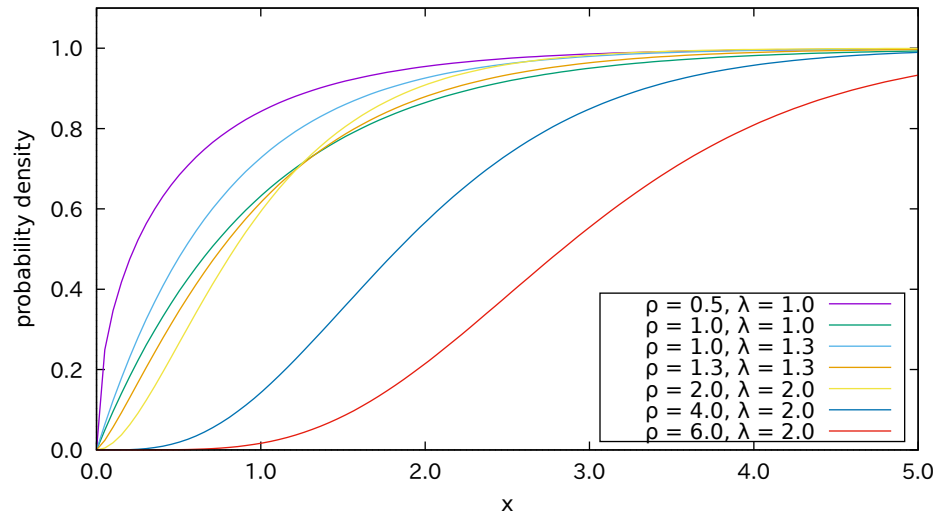


F CDF

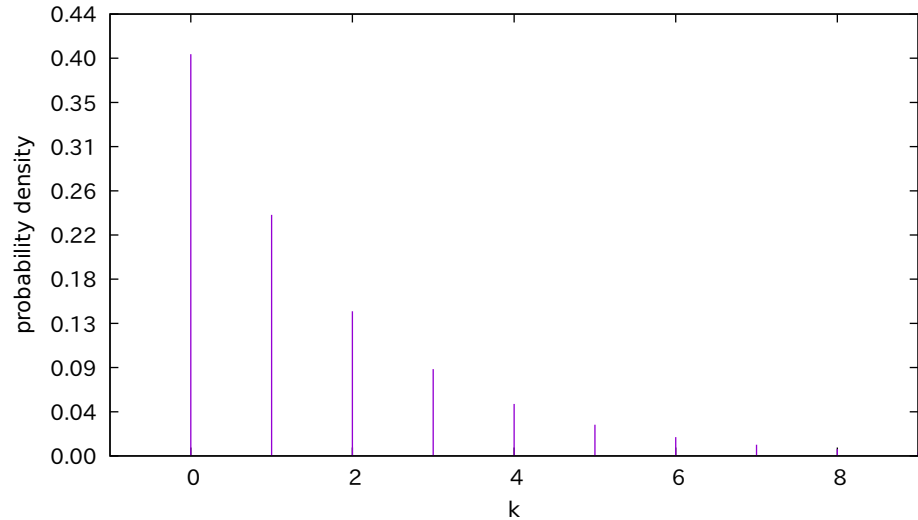


Gamma Γ PDF

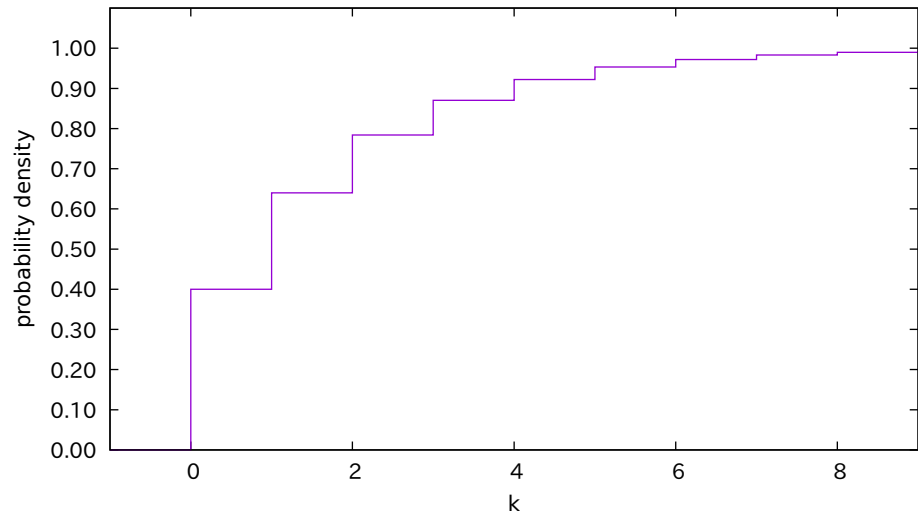
incomplete gamma CDF



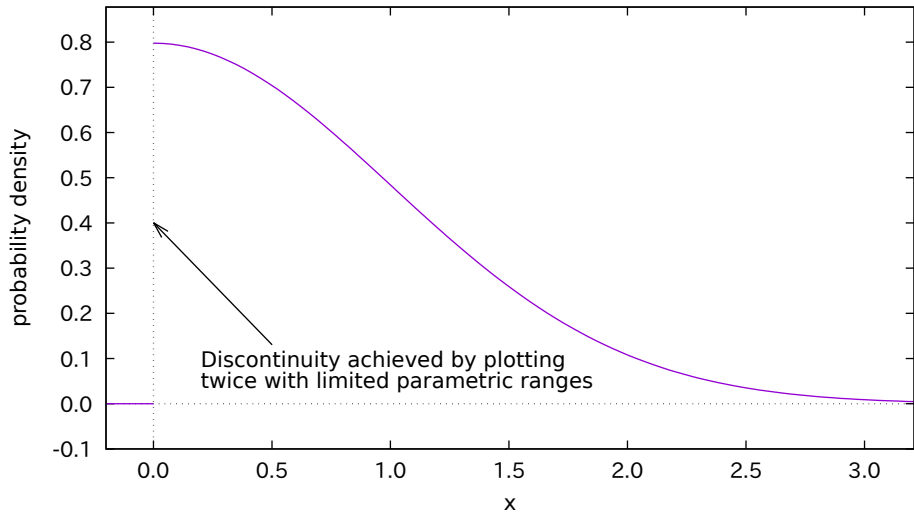
geometric PDF with $p = 0.4$



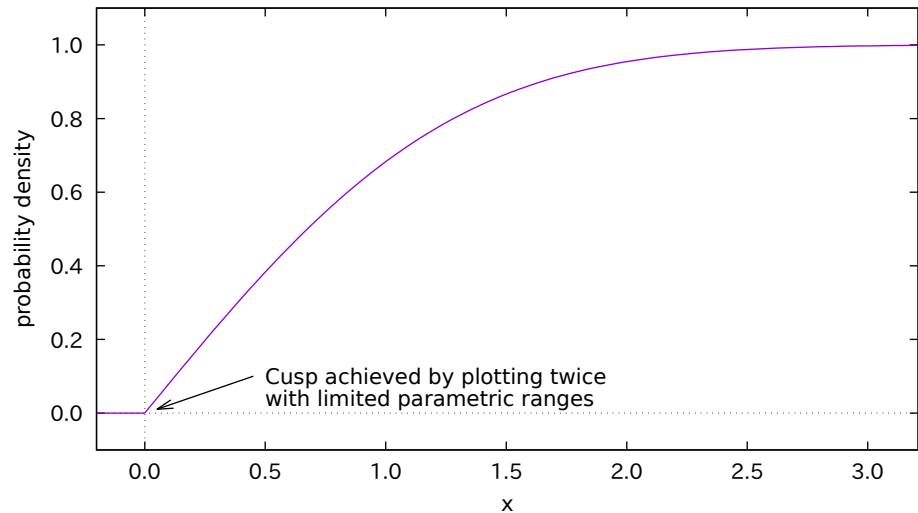
geometric CDF with $p = 0.4$



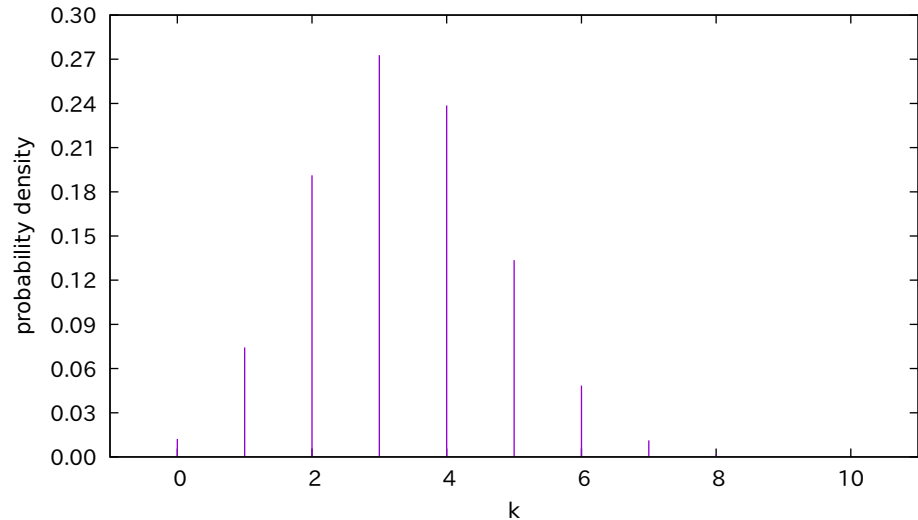
half normal PDF, $\sigma = 1.0$



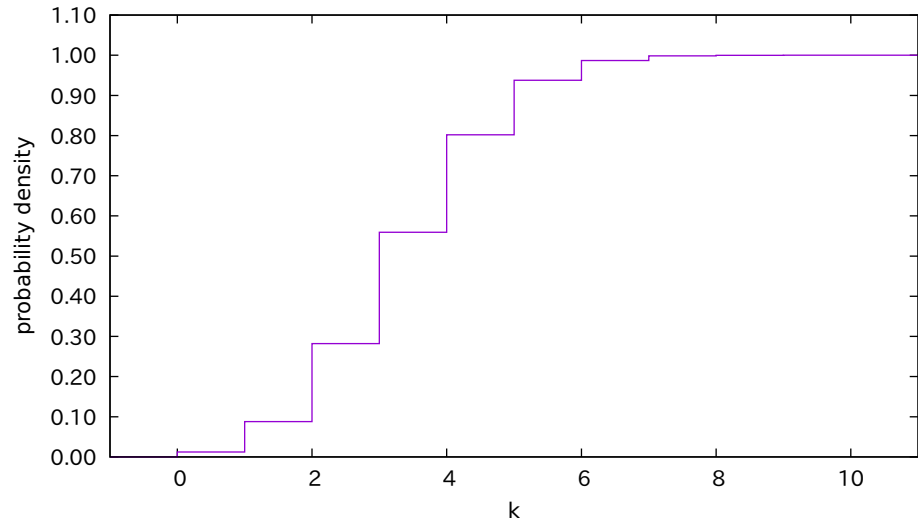
half normal CDF, $\sigma = 1.0$



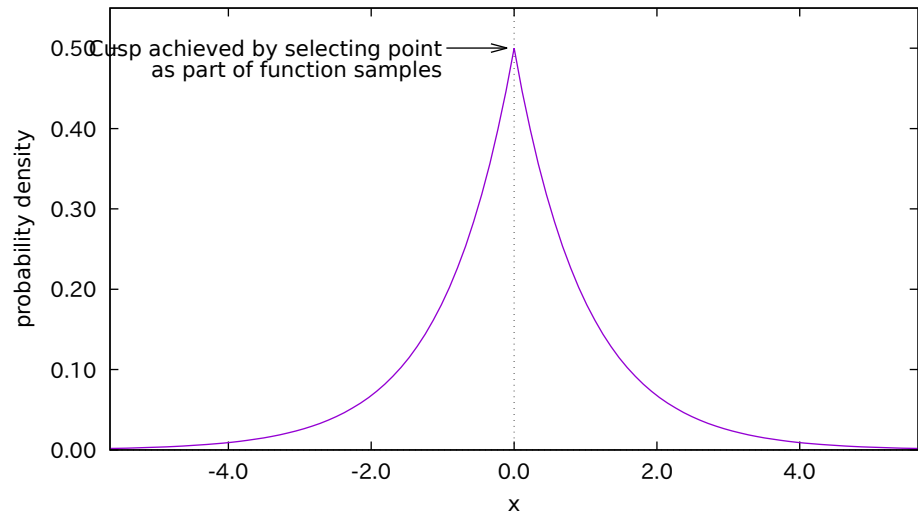
hypergeometric PDF with $N = 75$, $C = 25$, $d = 10$



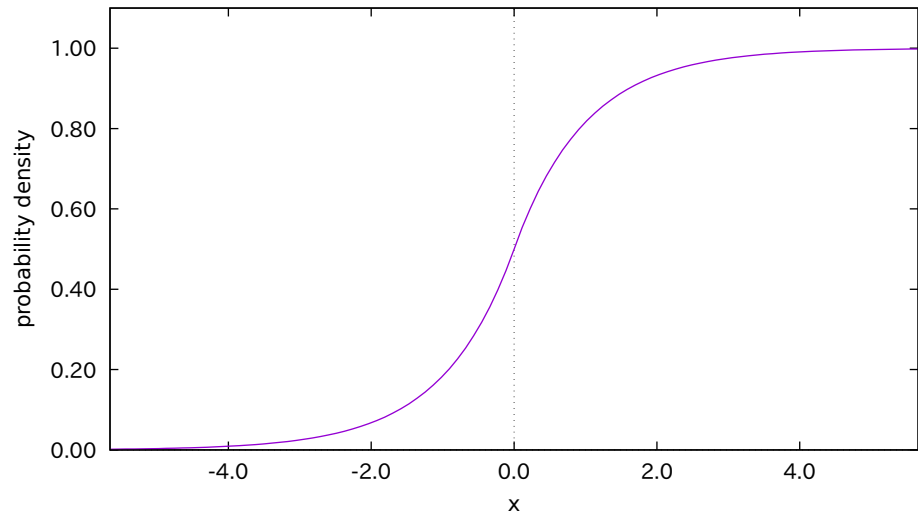
hypergeometric CDF with $N = 75$, $C = 25$, $d = 10$



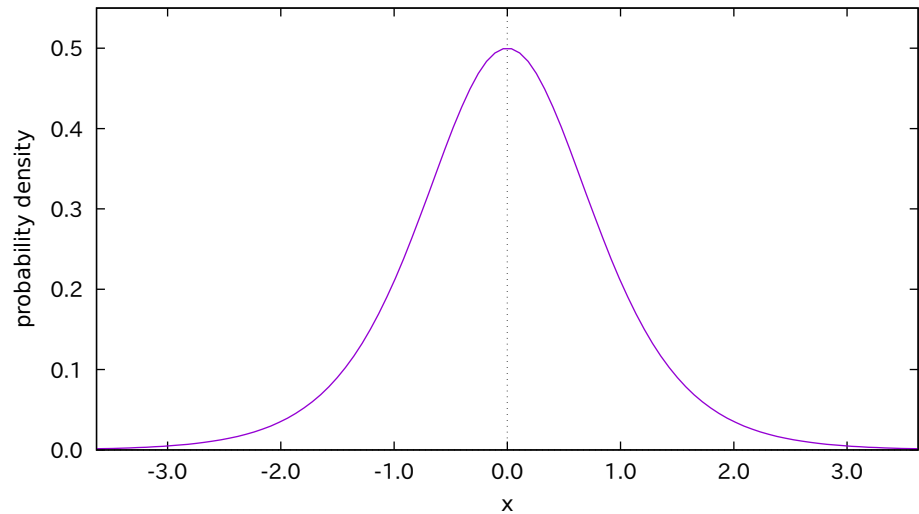
Laplace (or double exponential) PDF with $\mu = 0$, $b = 1$



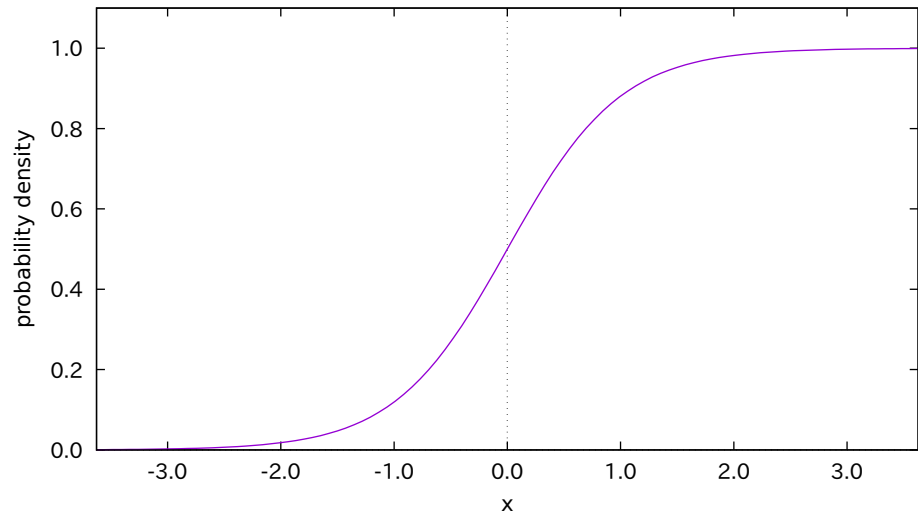
Laplace (or double exponential) CDF with $\mu = 0$, $b = 1$



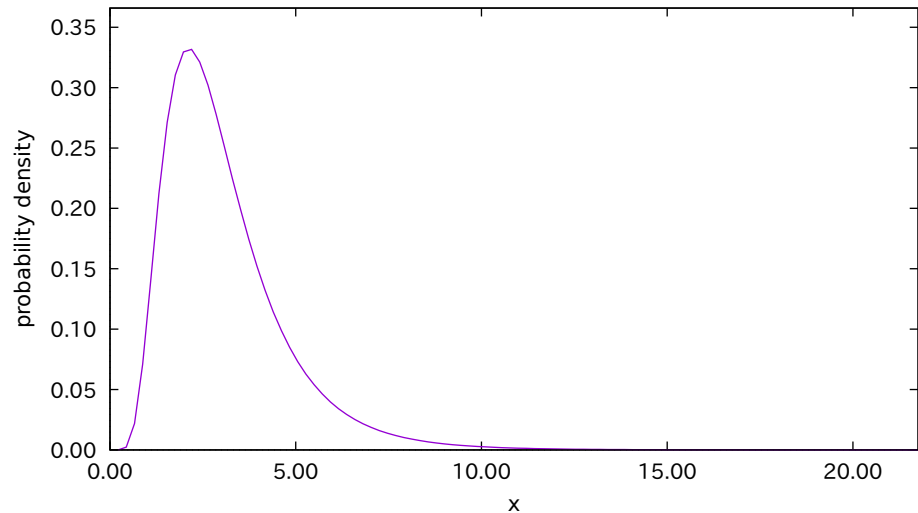
logistic PDF with $a = 0$, $\lambda = 2$



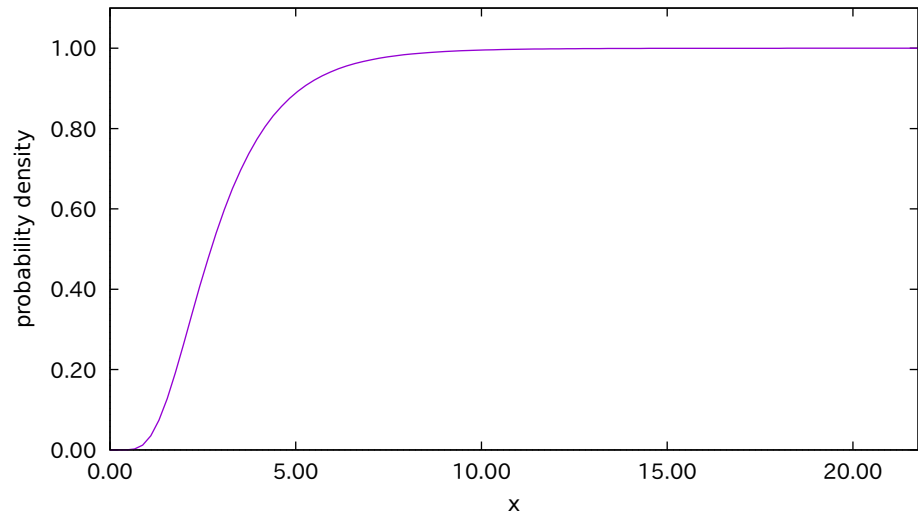
logistic CDF with $a = 0$, $\lambda = 2$



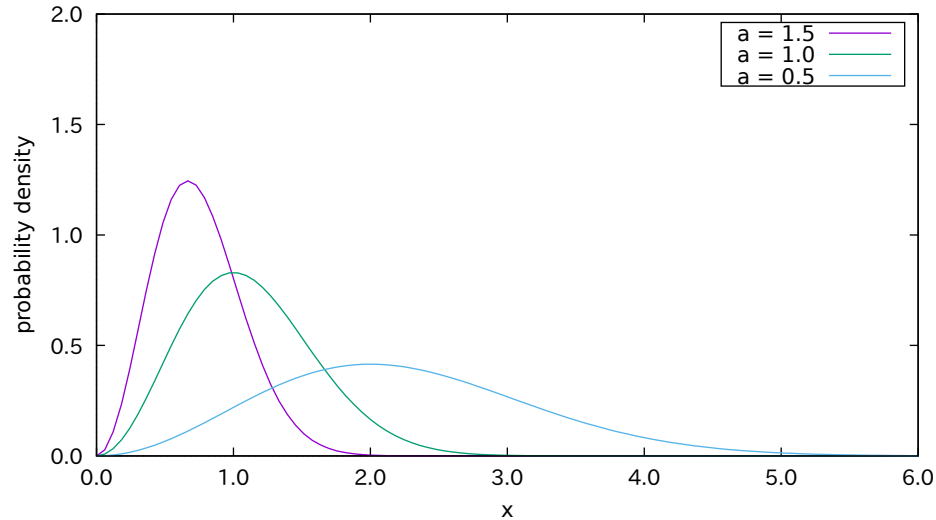
lognormal PDF with $\mu = 1.0$, $\sigma = 0.5$



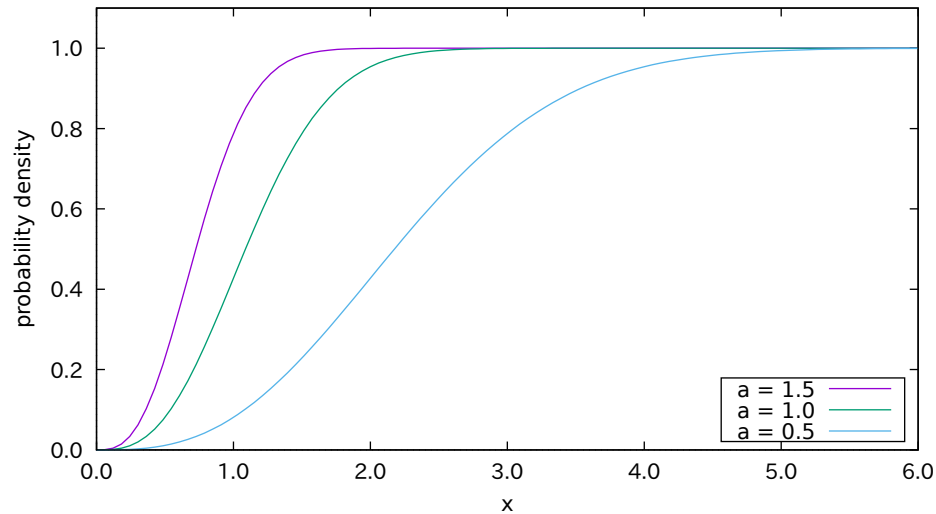
lognormal CDF with $\mu = 1.0$, $\sigma = 0.5$



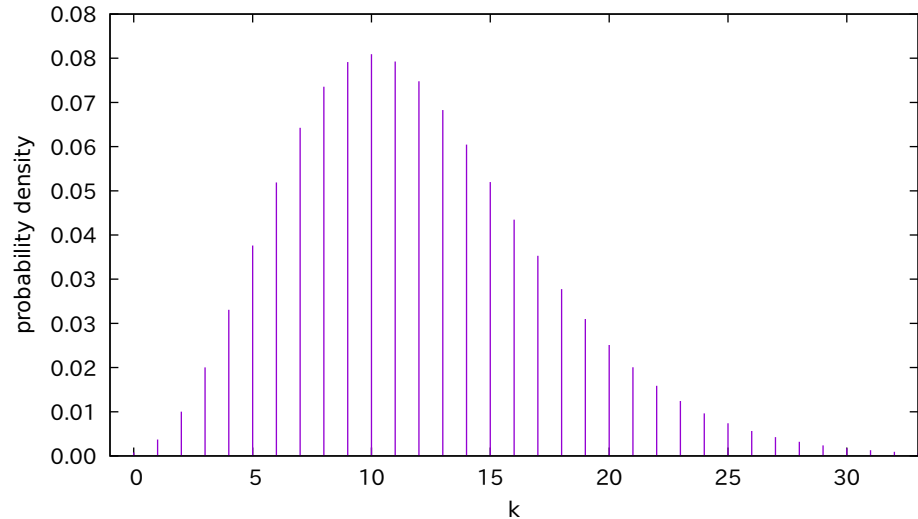
Maxwell PDF



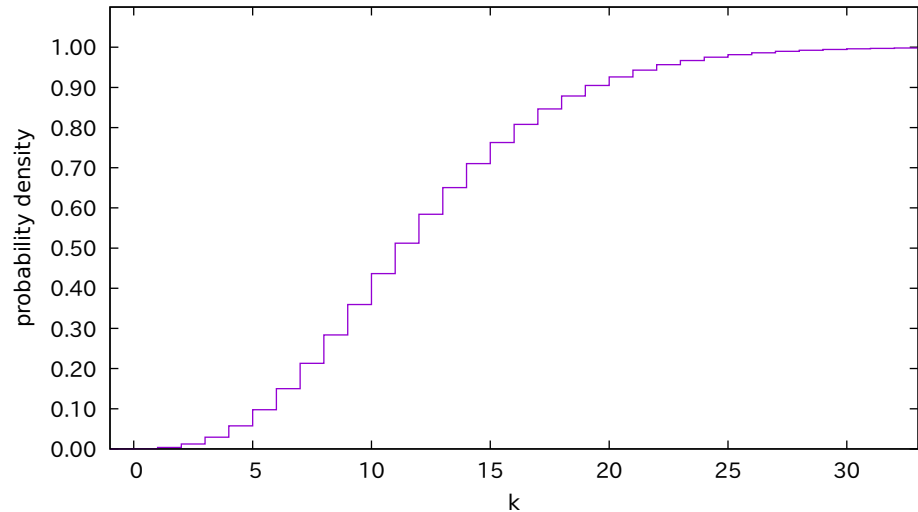
Maxwell CDF



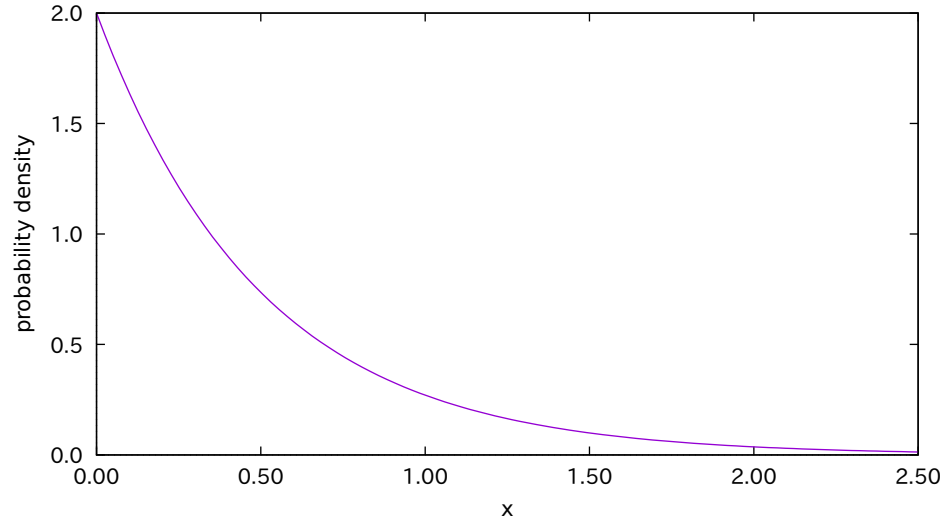
negative binomial (or Pascal or Polya) PDF with $r = 8$, $p = 0.4$



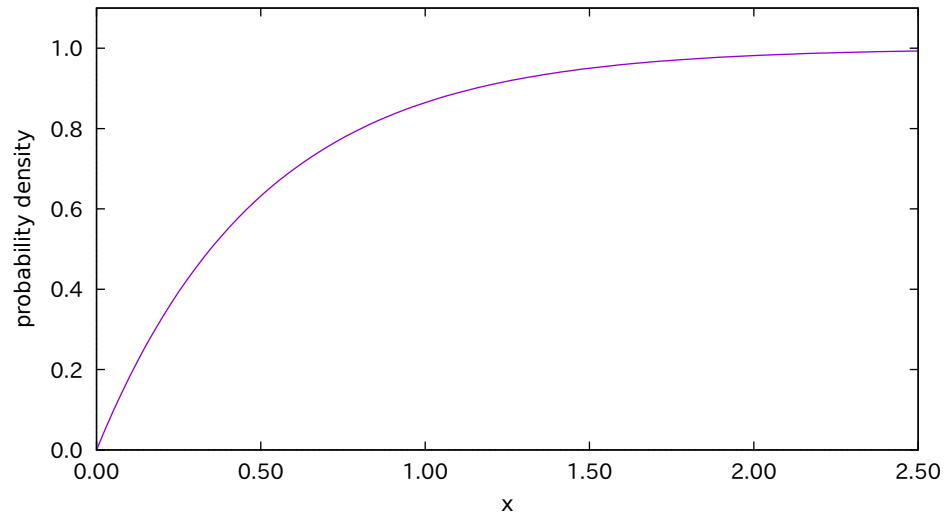
negative binomial (or Pascal or Polya) CDF with $r = 8$, $p = 0.4$



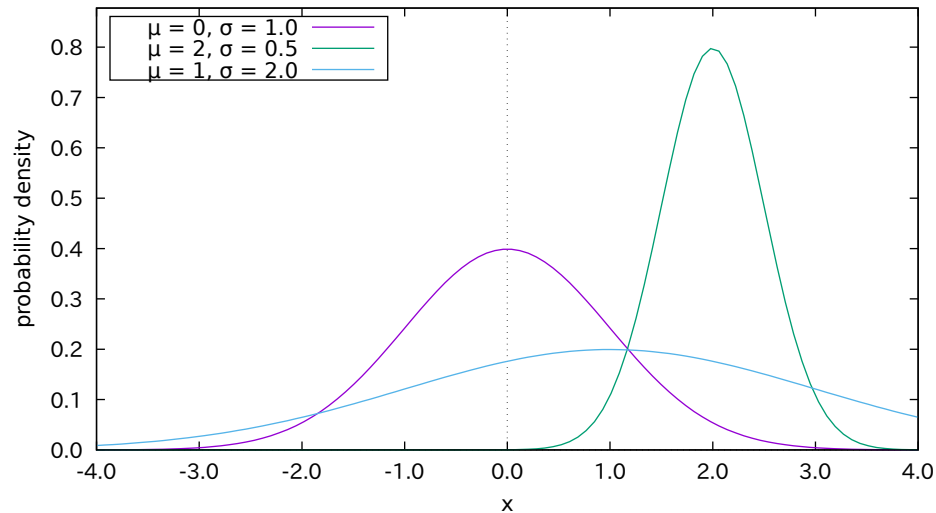
negative exponential (or exponential) PDF with $\lambda = 2.0$



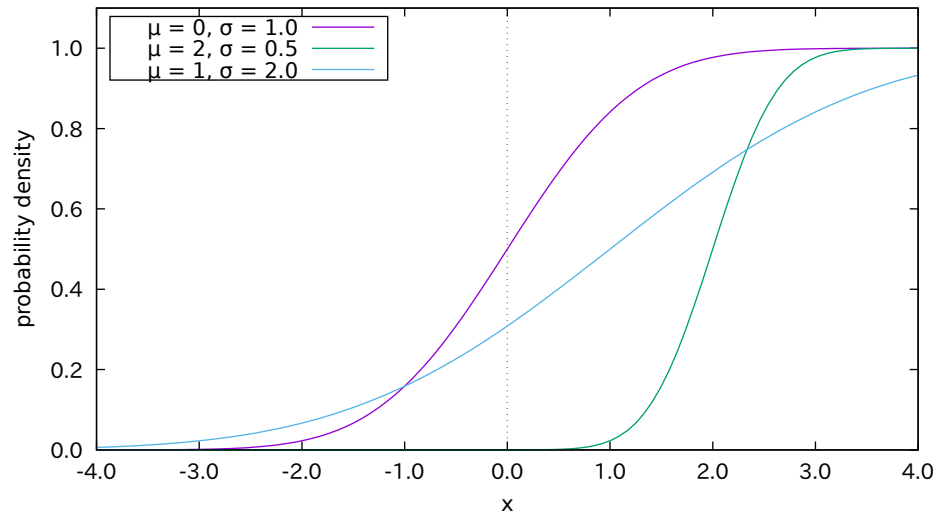
negative exponential (or exponential) CDF with $\lambda = 2.0$



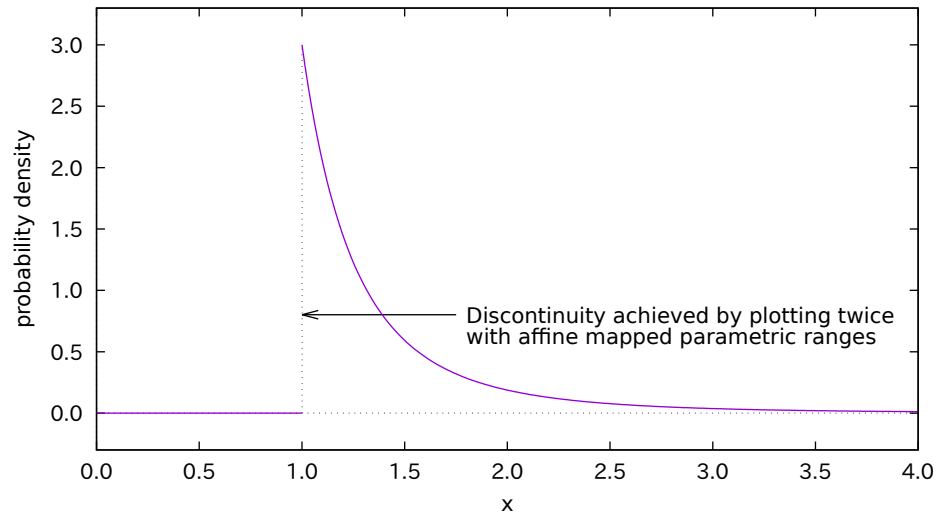
normal (also called Gauss or bell-curved) PDF



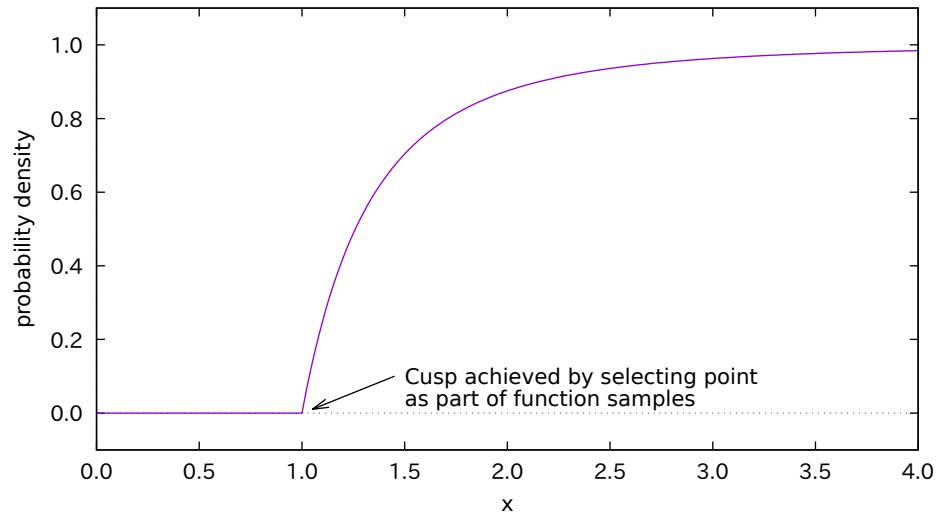
normal (also called Gauss or bell-curved) CDF



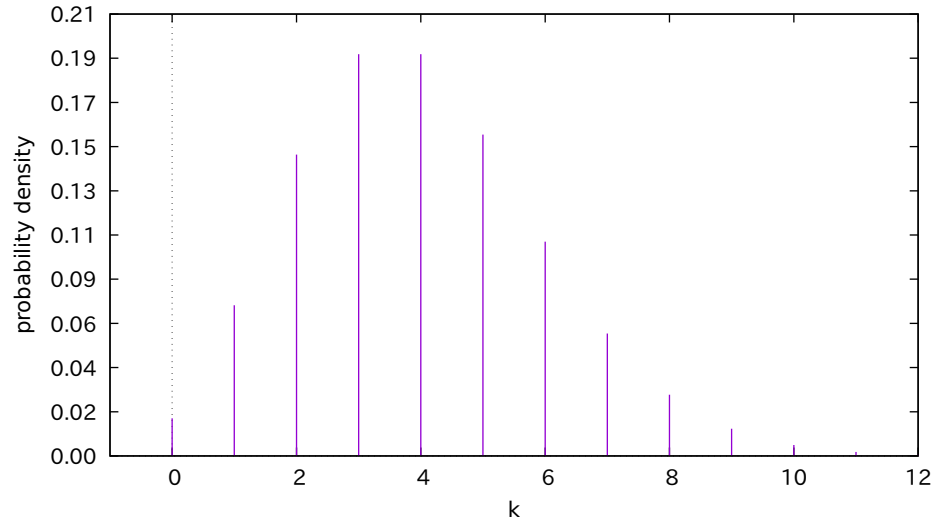
Pareto PDF with $a = 1$, $b = 3$



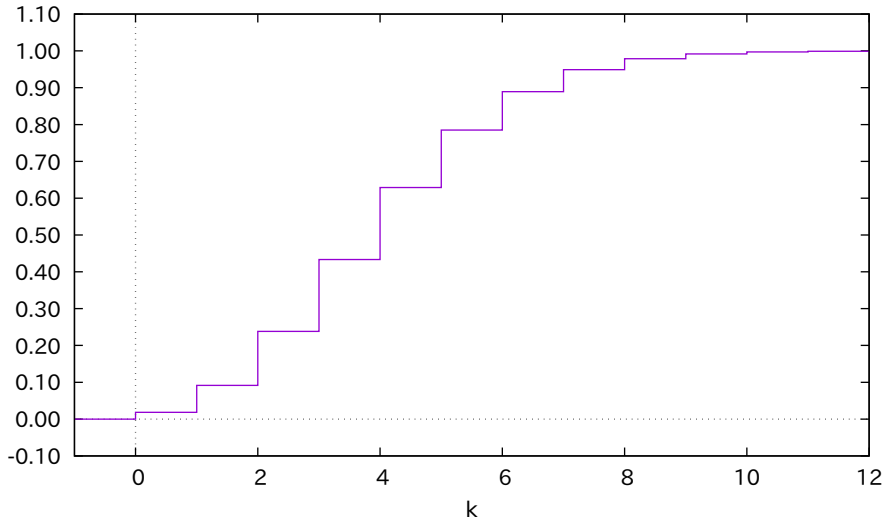
Pareto CDF with $a = 1$, $b = 3$



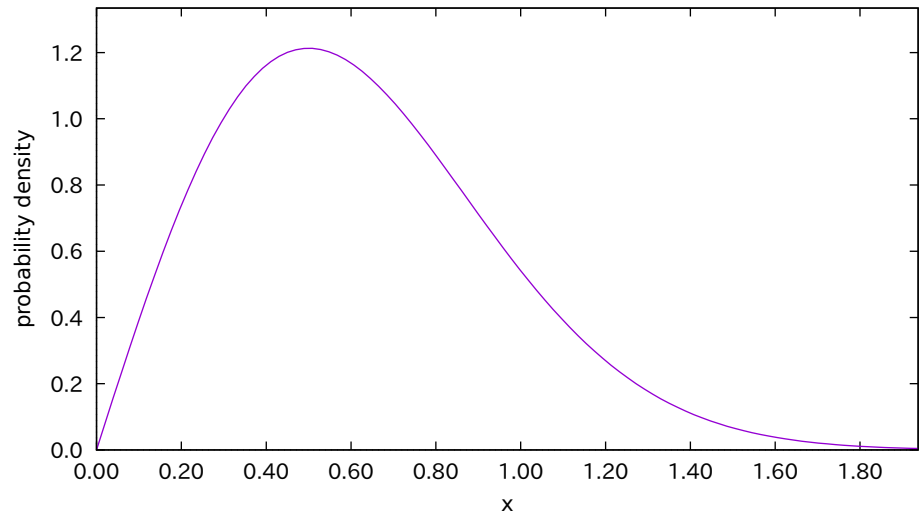
Poisson PDF with $\mu = 4.0$



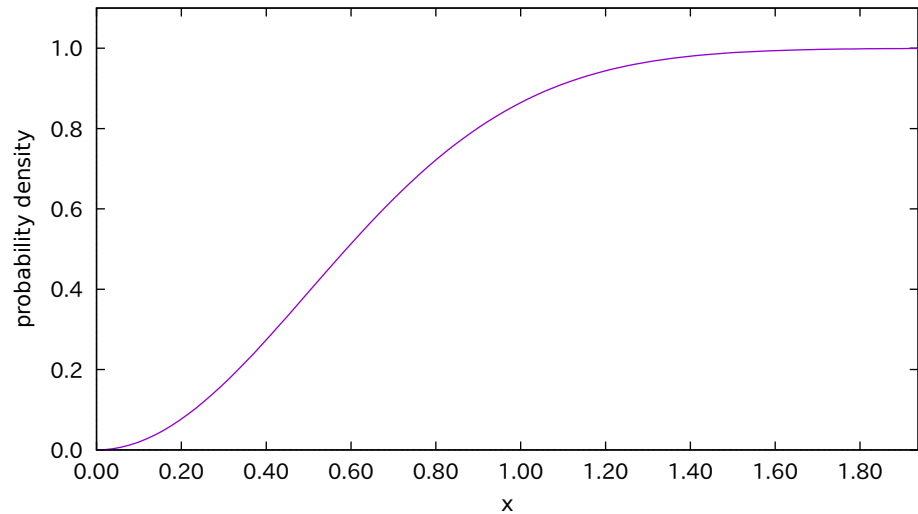
Poisson CDF with $\mu = 4.0$



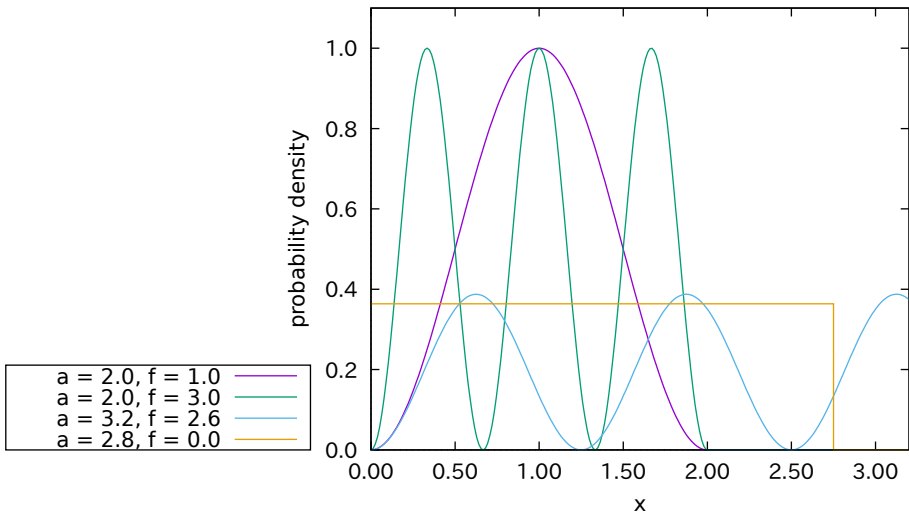
Rayleigh PDF with $\lambda = 2.0$



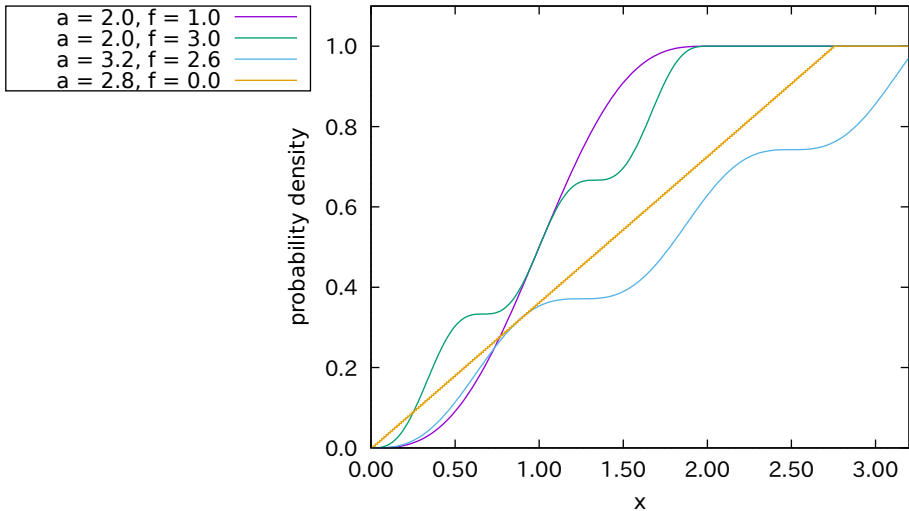
Rayleigh CDF with $\lambda = 2.0$



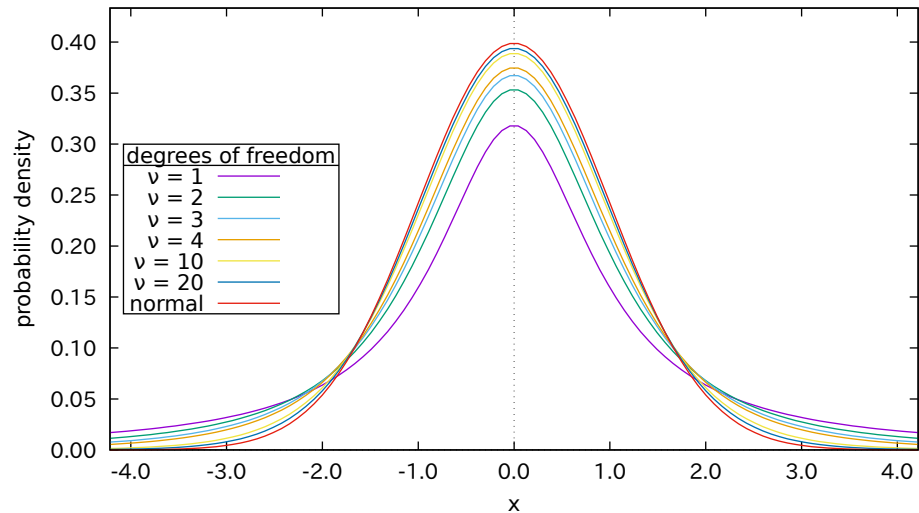
sine PDF



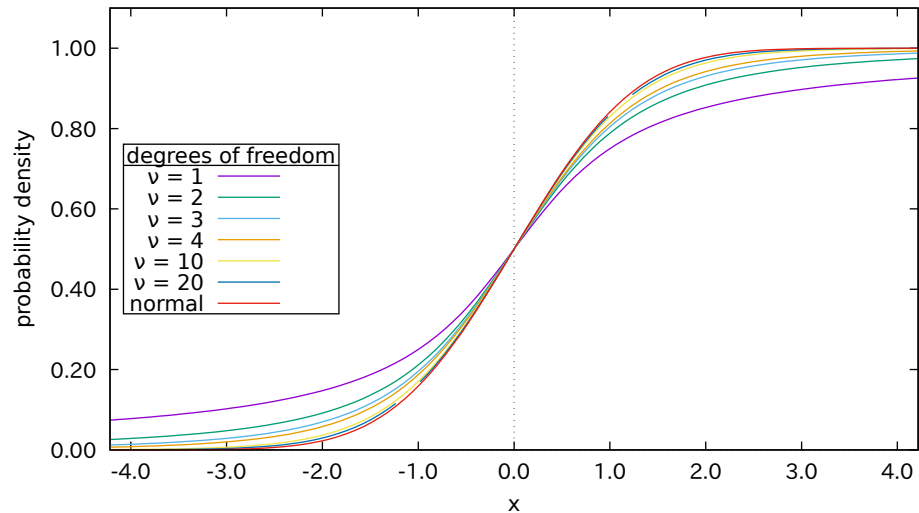
sine CDF



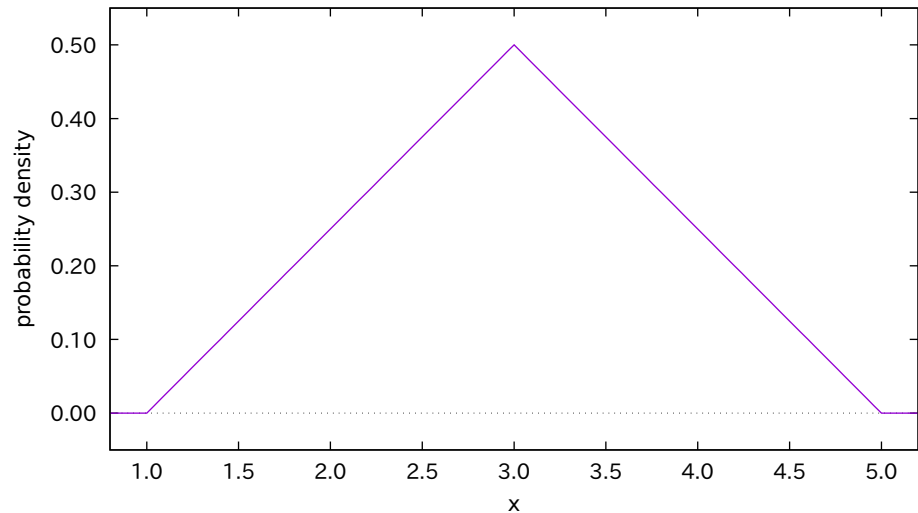
t PDF (and Gaussian limit)



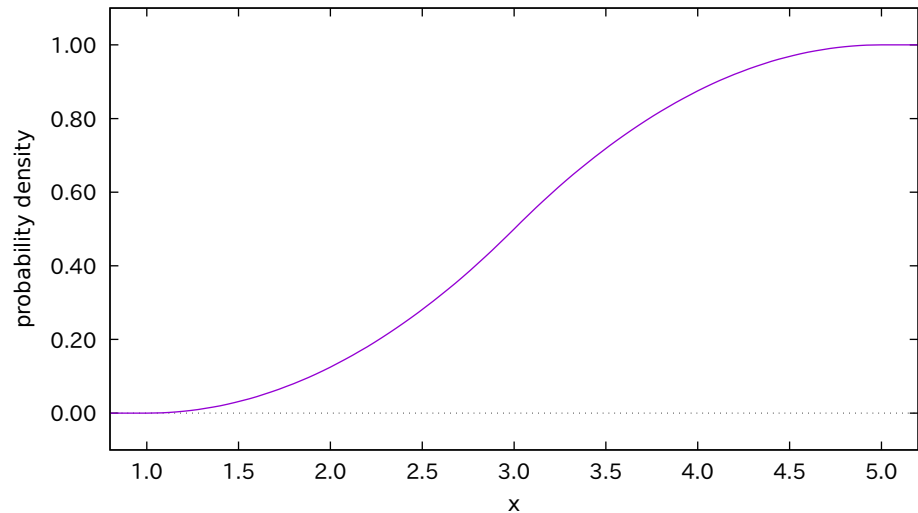
t CDF (and Gaussian limit)



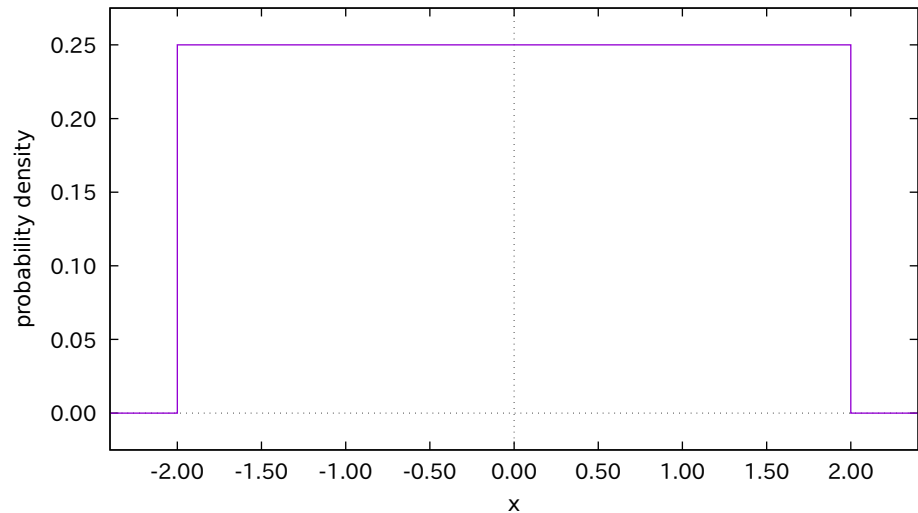
triangular PDF with $m = 3.0$, $g = 2.0$



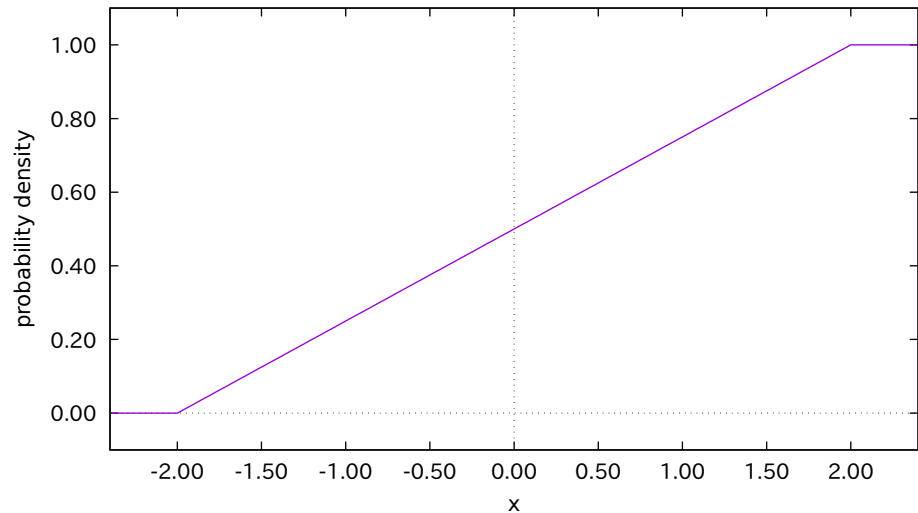
triangular CDF with $m = 3.0$, $g = 2.0$



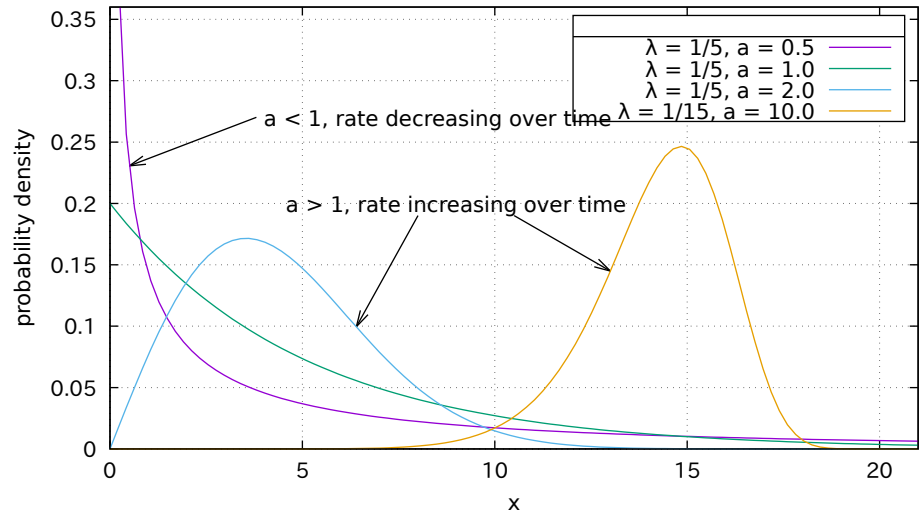
uniform PDF with $a = -2.0$, $b = 2.0$



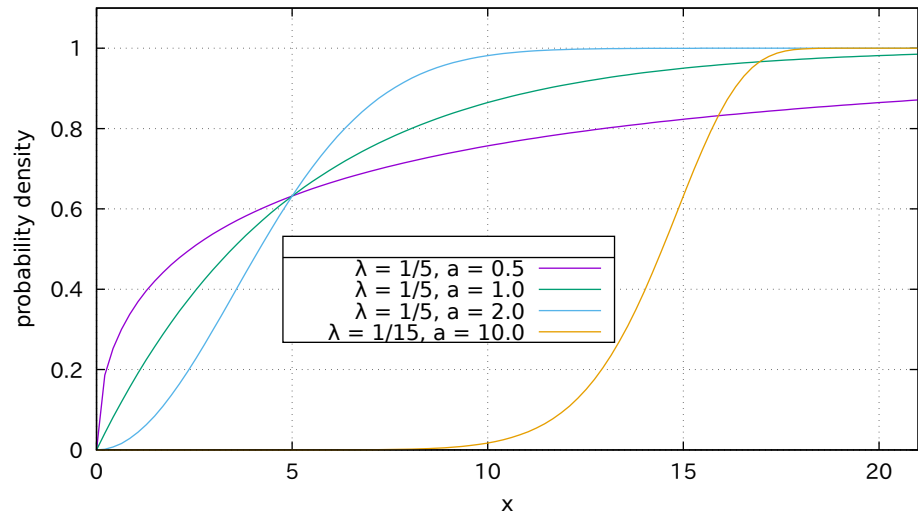
uniform CDF with $a = -2.0$, $b = 2.0$



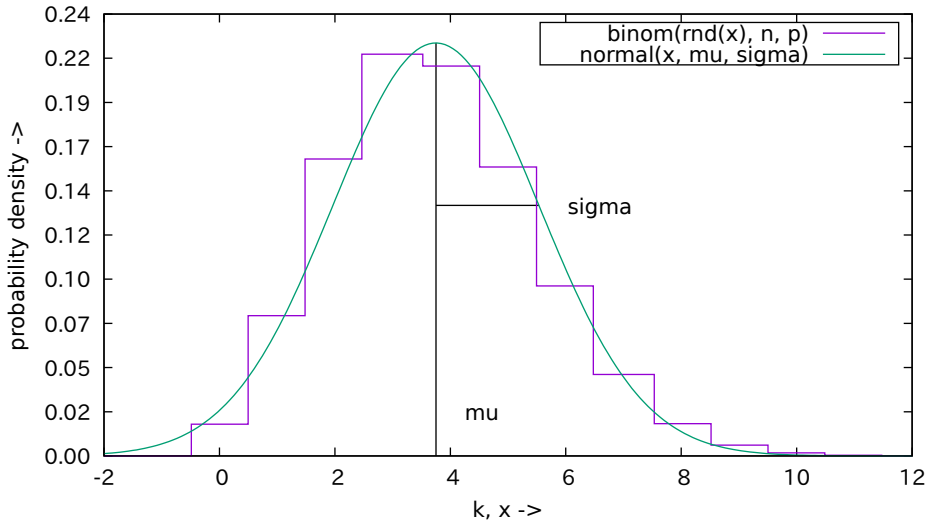
Weibull PDF



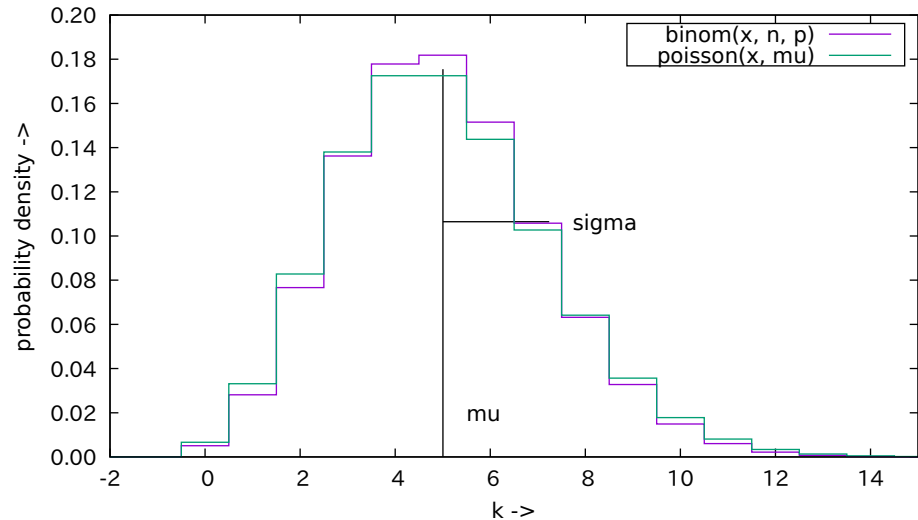
Weibull CDF



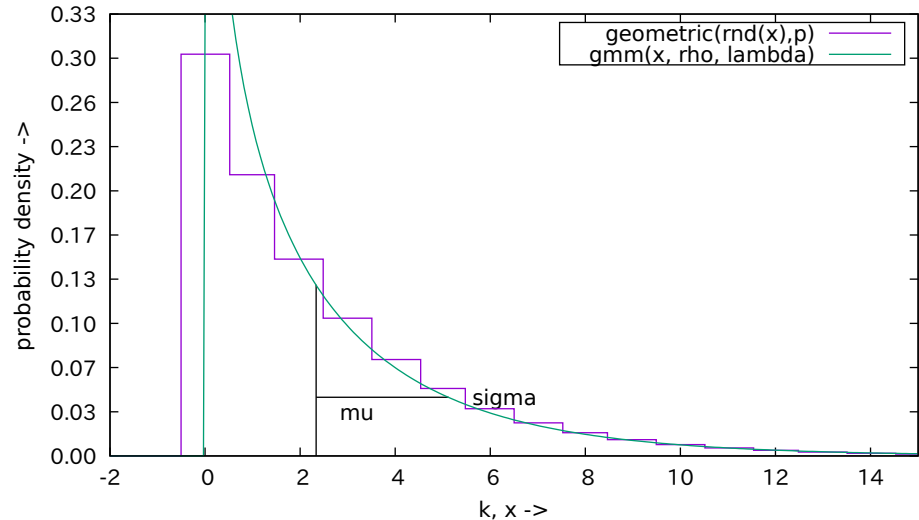
binomial PDF using normal approximation



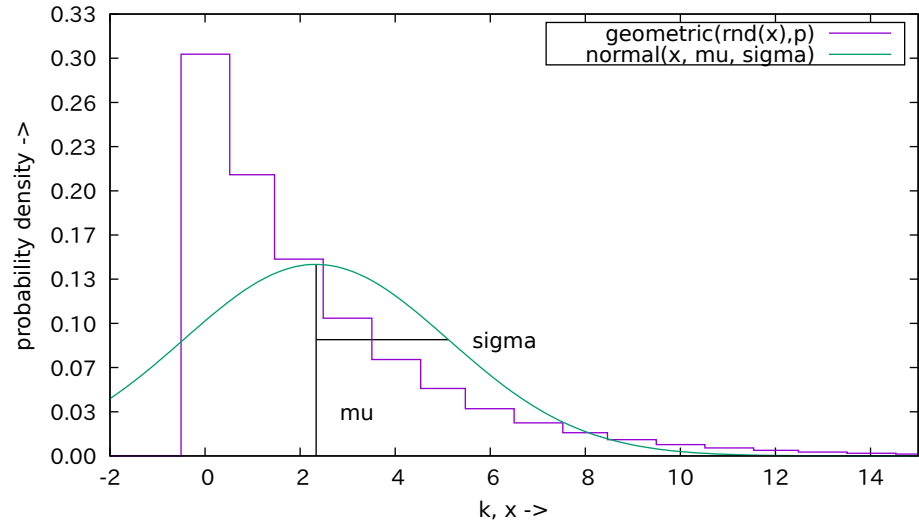
binomial PDF using poisson approximation



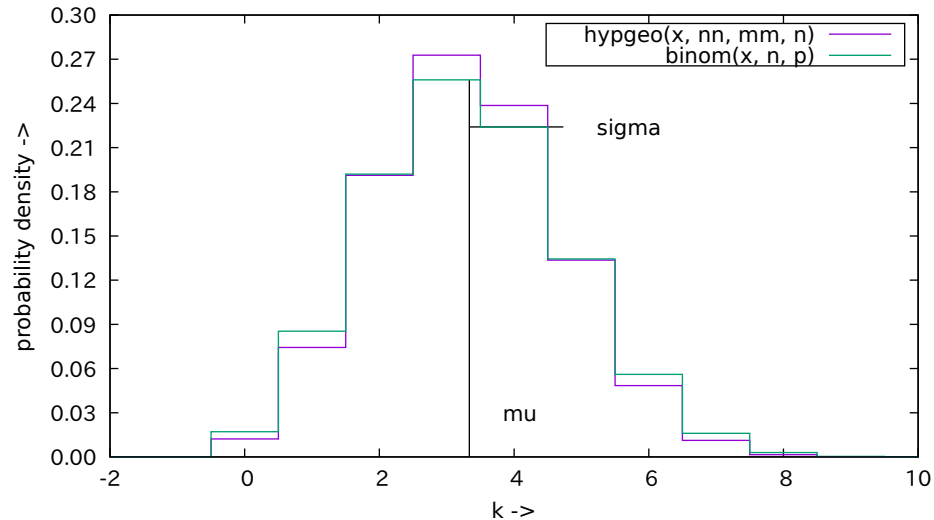
geometric PDF using gamma approximation



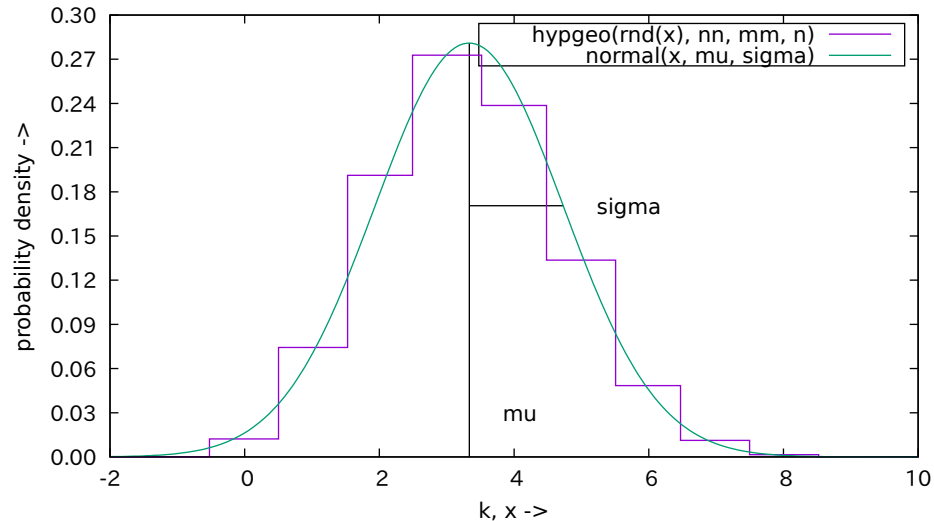
geometric PDF using normal approximation



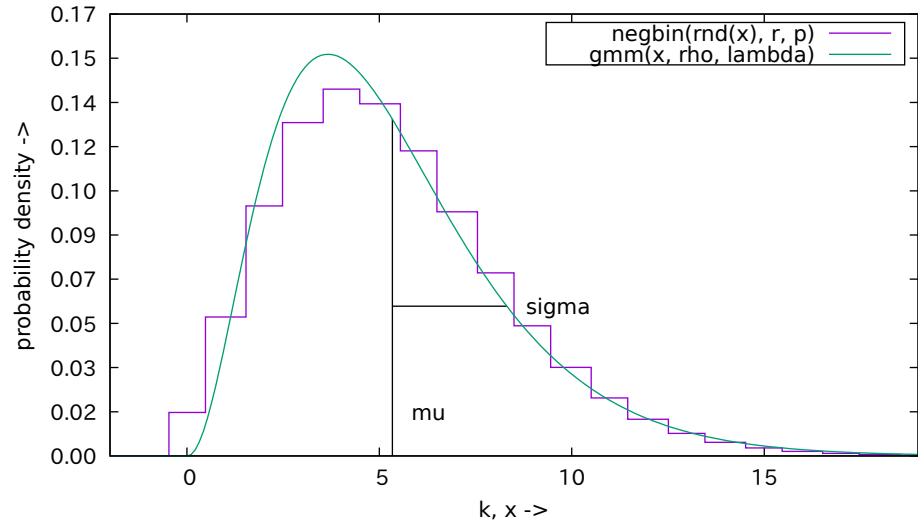
hypergeometric PDF using binomial approximation



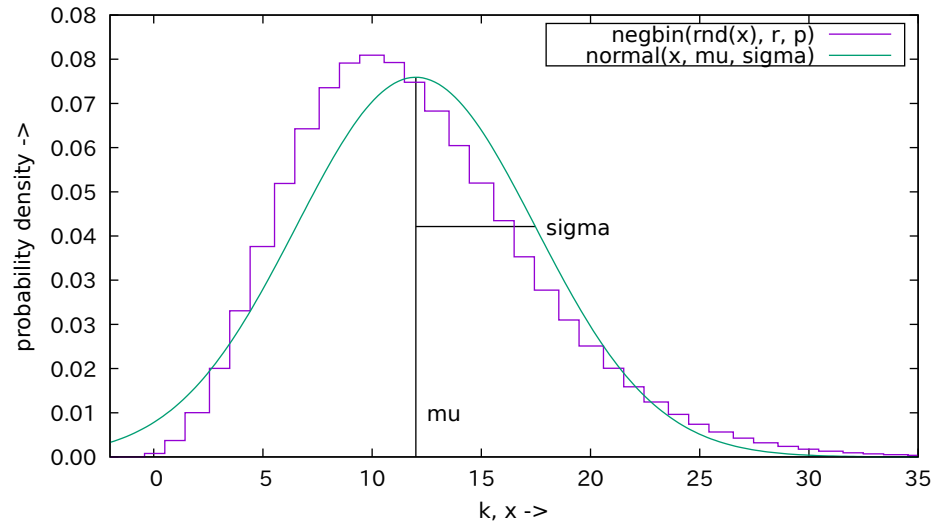
hypergeometric PDF using normal approximation



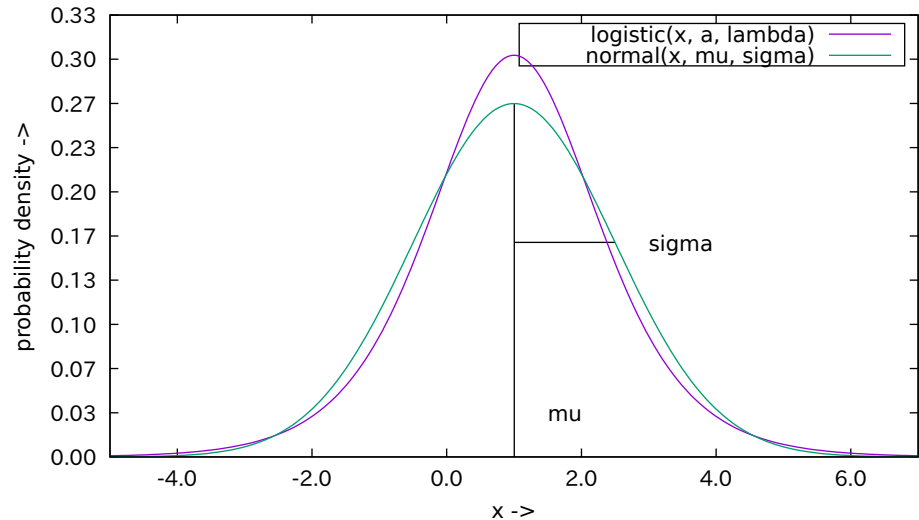
negative binomial PDF using gamma approximation



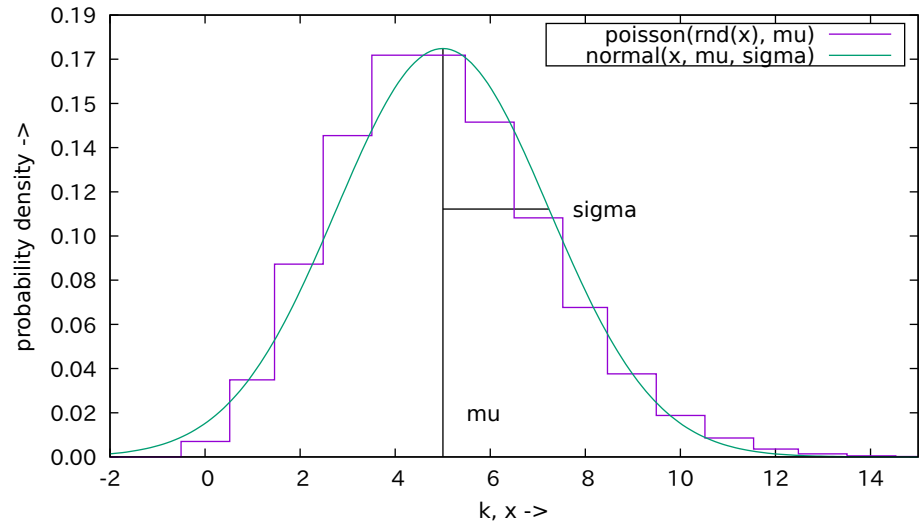
negative binomial PDF using normal approximation



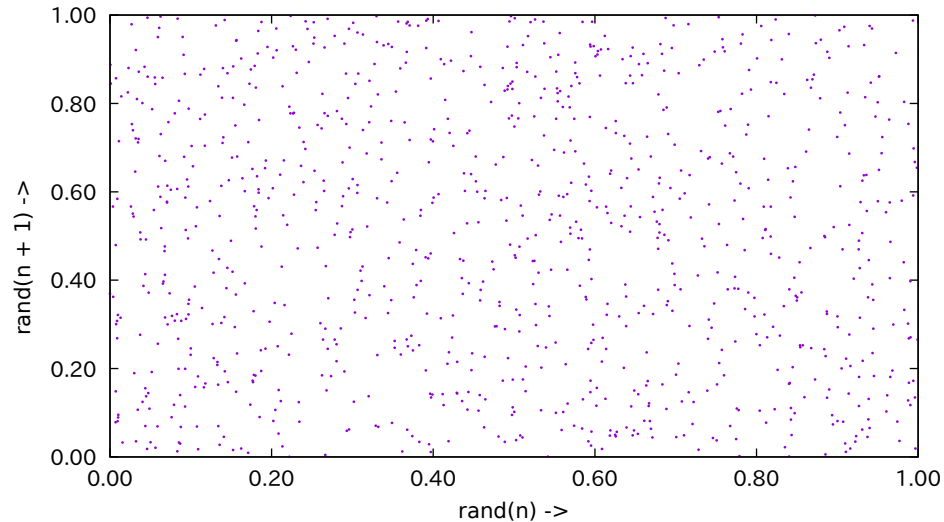
normal PDF using logistic approximation



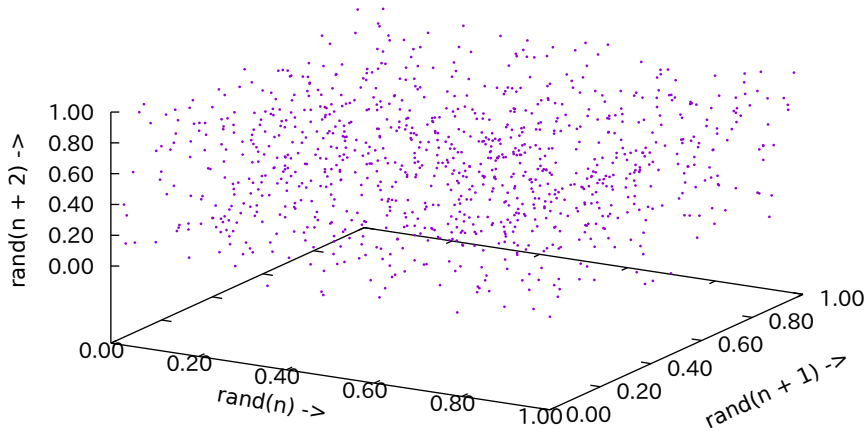
poisson PDF using normal approximation



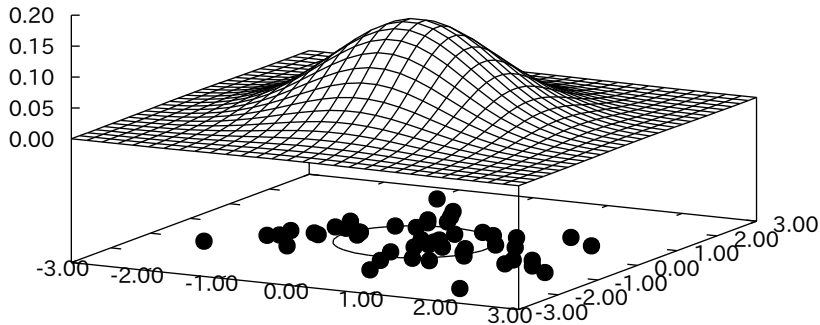
Lattice test for random numbers



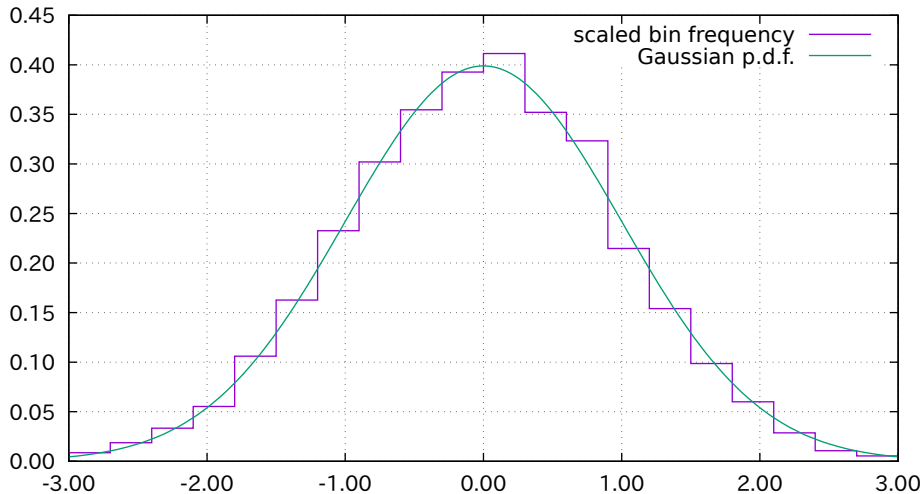
Lattice test for random numbers



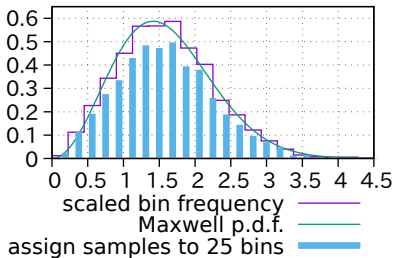
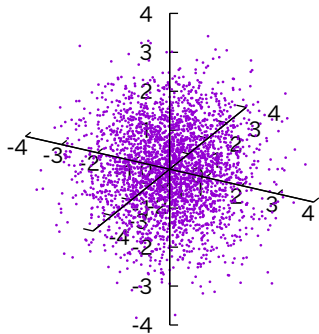
50 random samples from a 2D Gaussian PDF with
unit variance, zero mean and no dependence



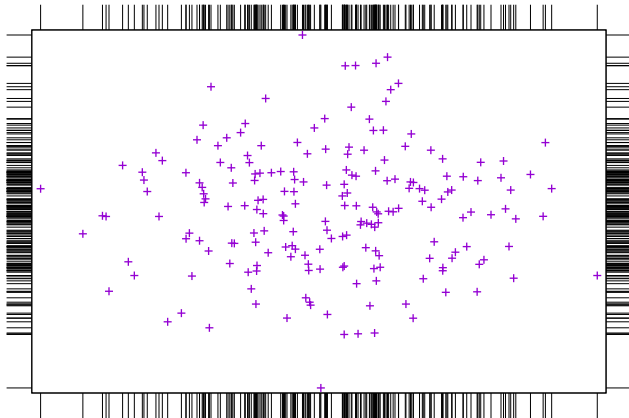
Histogram of 5000 random samples from a univariate
Gaussian PDF with unit variance and zero mean



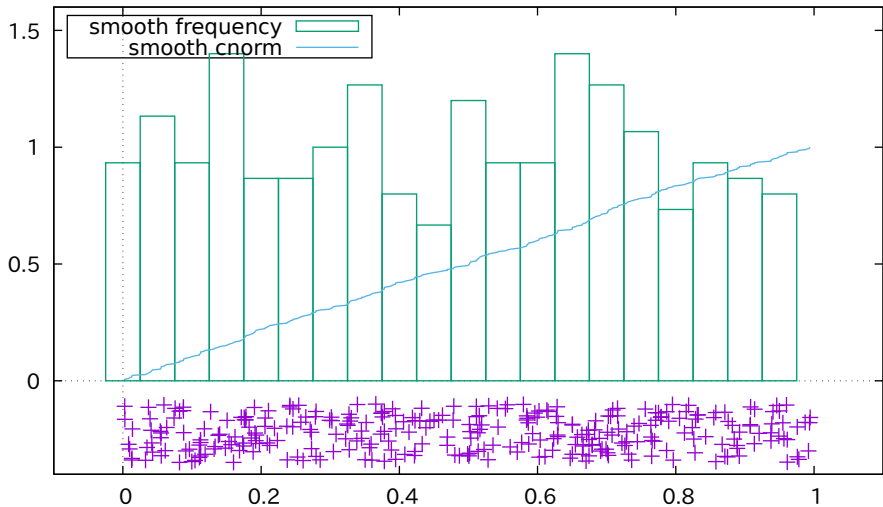
Gaussian 3D cloud of 3000 random samples Histogram of distance from origin of
3000 multivariate unit variance samples



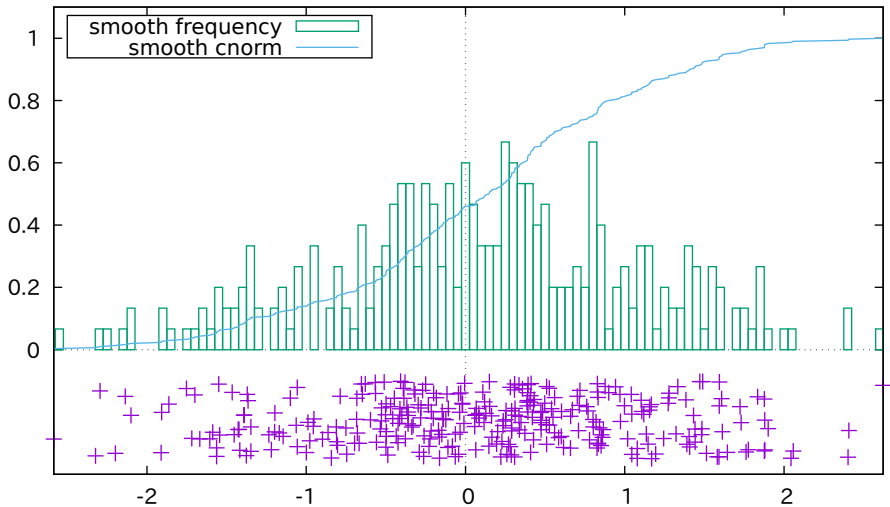
Rug Plot



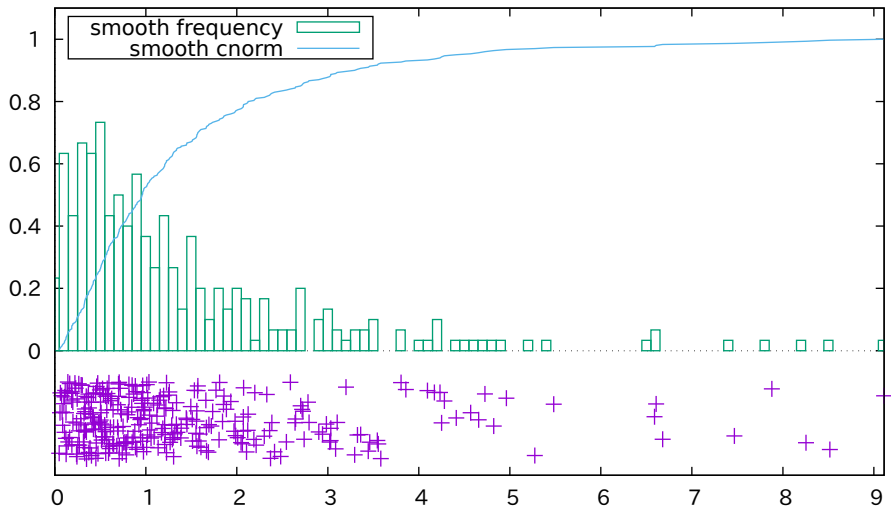
Uniform Distribution



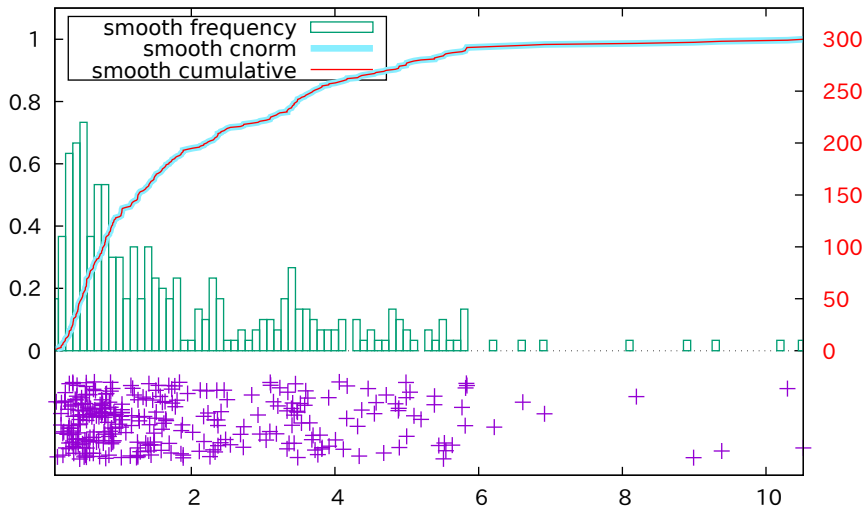
Normal Distribution



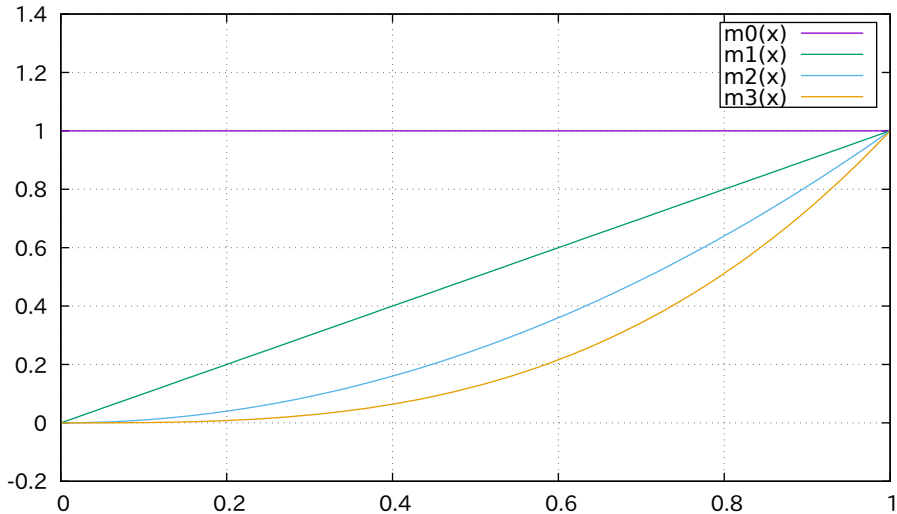
Lognormal Distribution



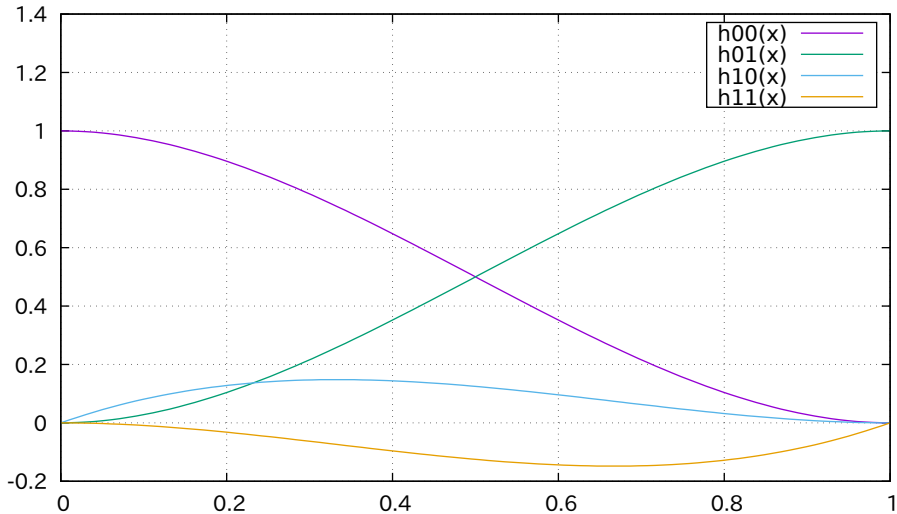
Mixed Distribution (Lognormal with shifted Gaussian)



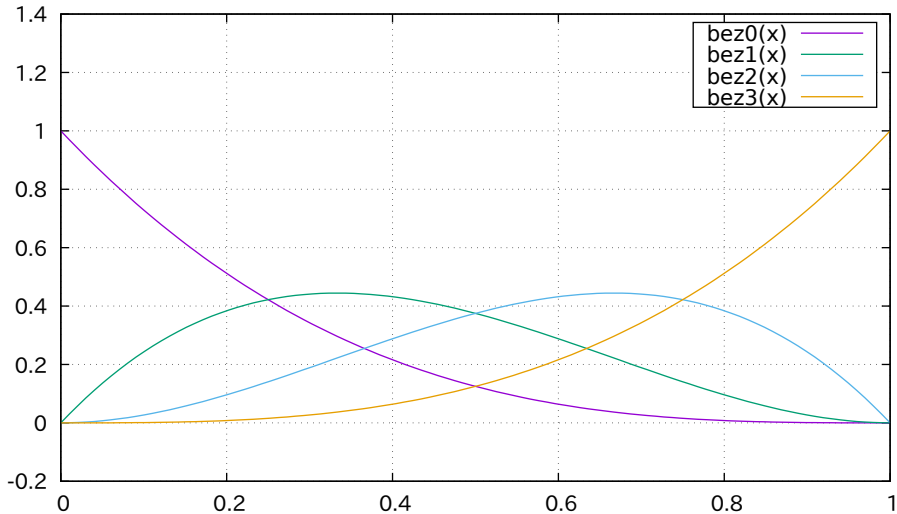
The cubic Monomial basis functions



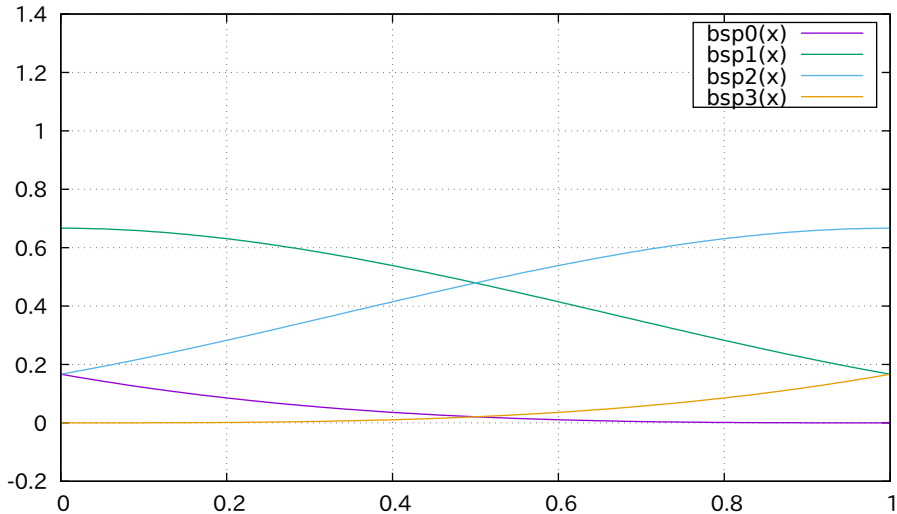
The cubic Hermite basis functions



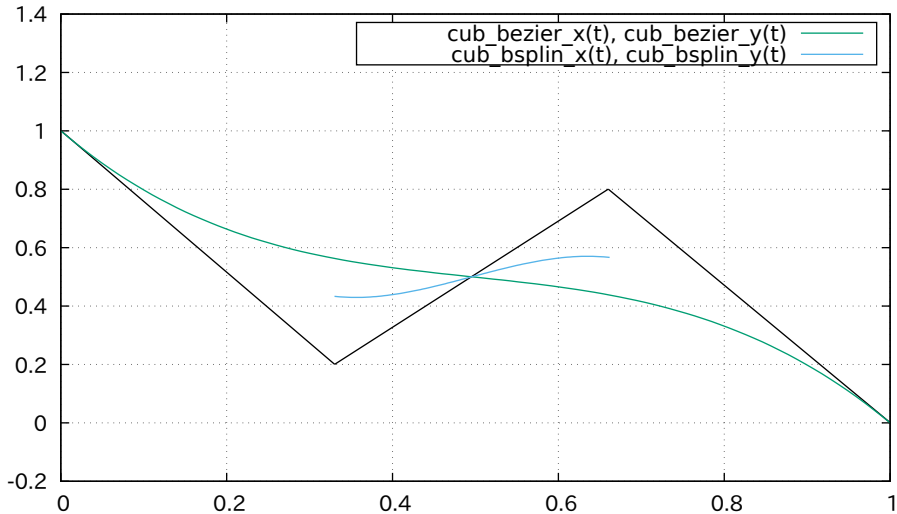
The cubic Bezier basis functions



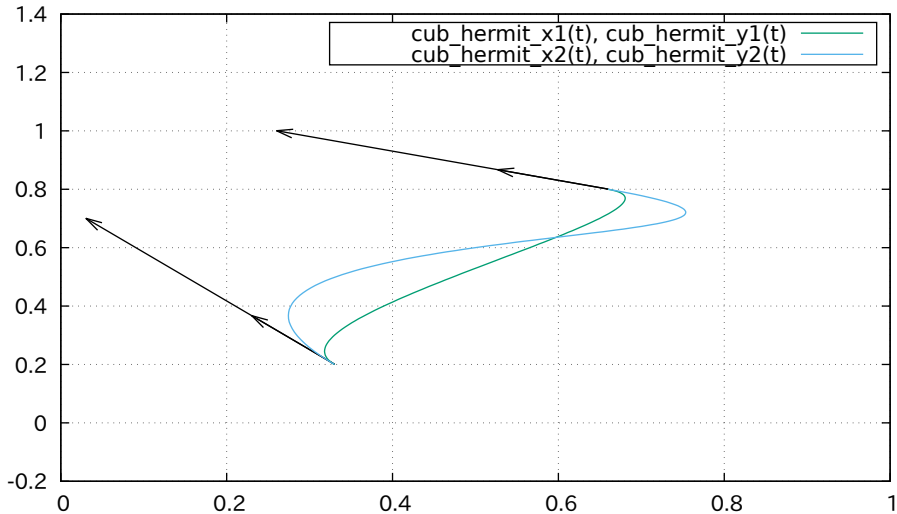
The cubic uniform Bspline basis functions



The cubic Bezier/Bspline basis functions in use

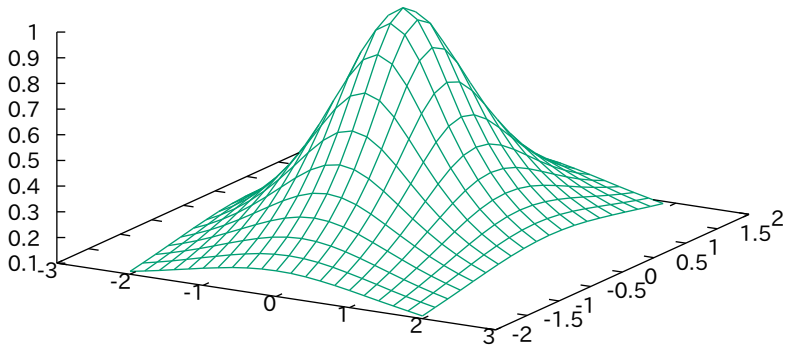


The cubic Hermite basis functions in use



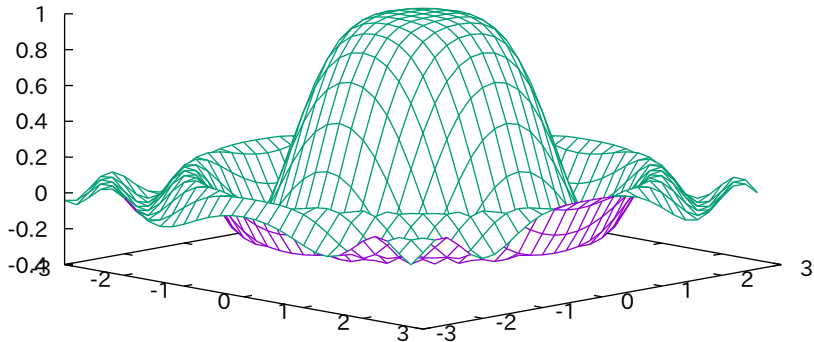
Hidden line removal of explicit binary surfaces

"binary1" binary



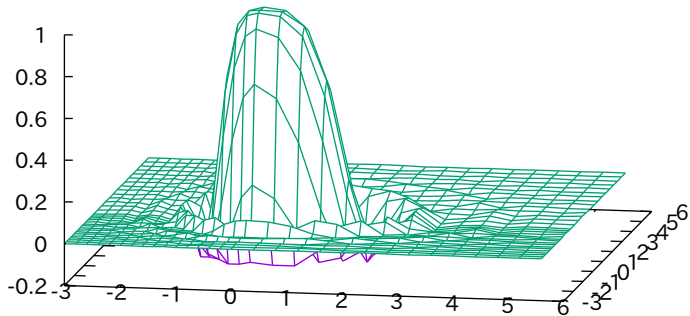
Hidden line removal of explicit binary surfaces

"binary2" binary

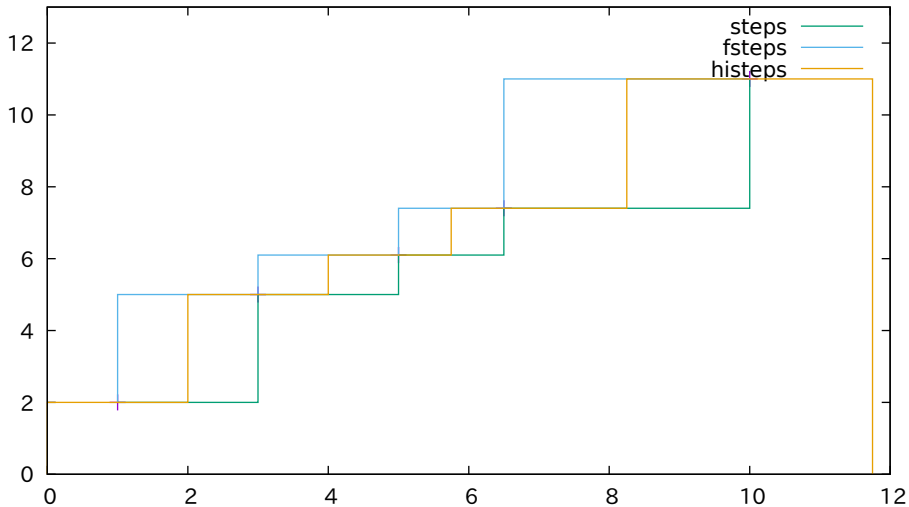


Notice that sampling rate can change

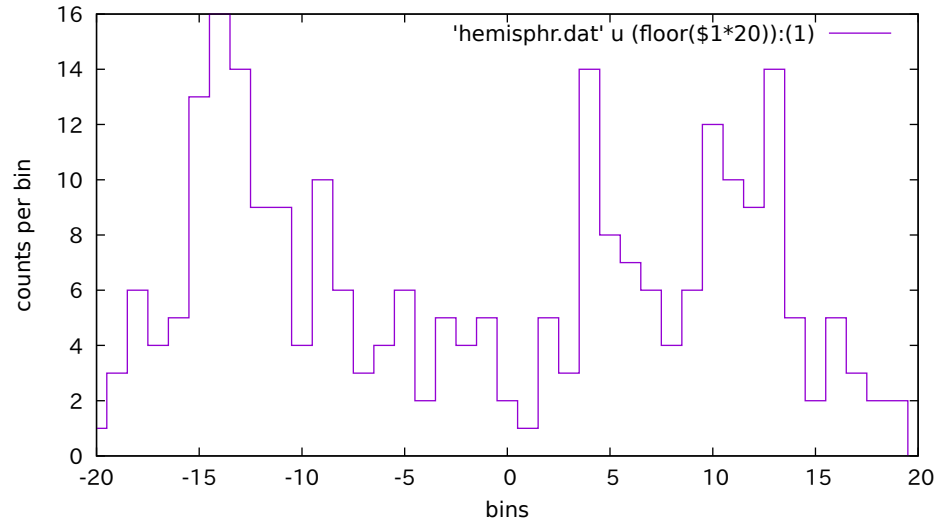
"binary3" binary



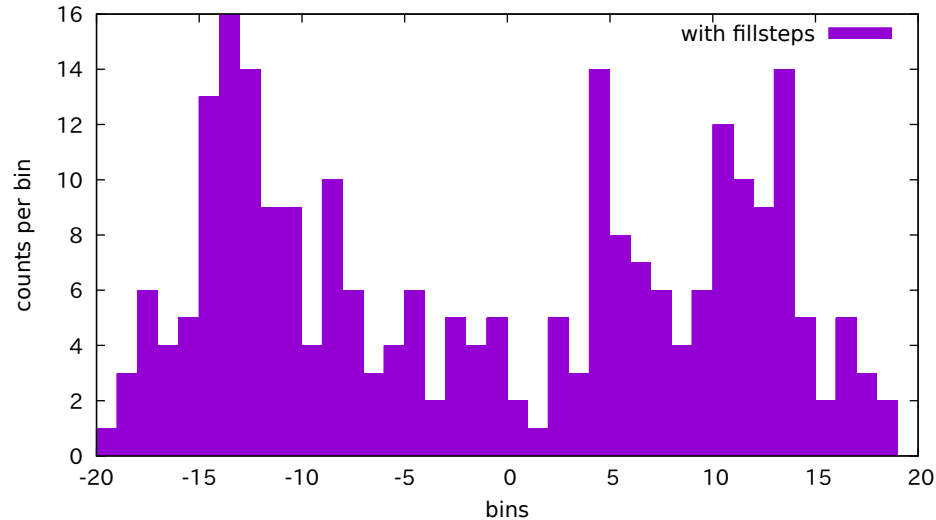
Compare steps, fsteps and histeps



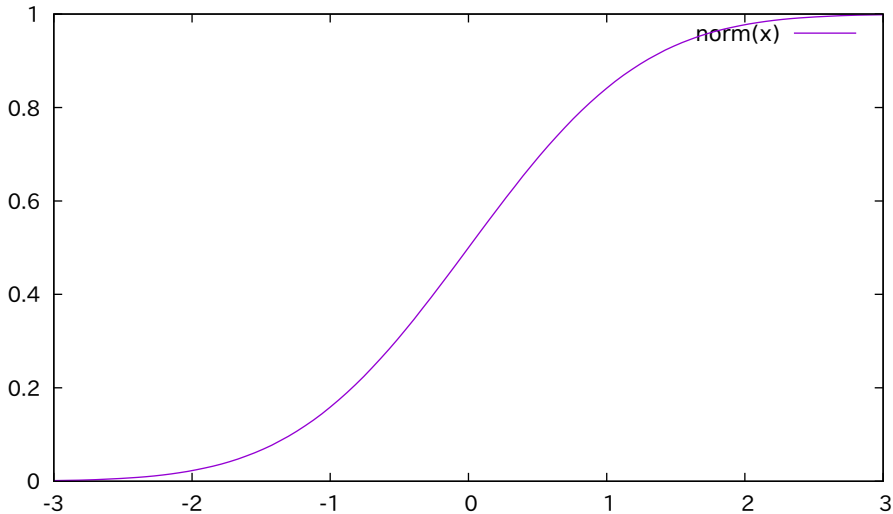
Histogram built from unsorted data by 'smooth frequency'



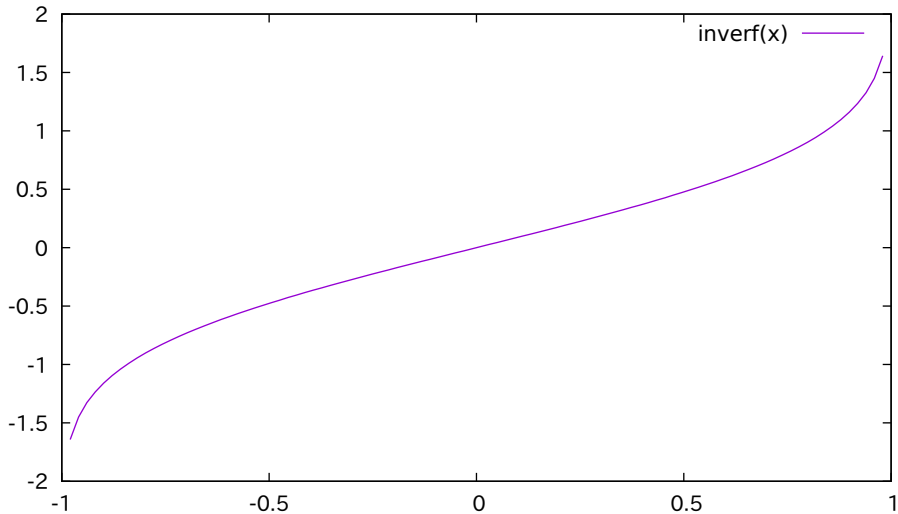
Histogram built from unsorted data by 'smooth frequency'



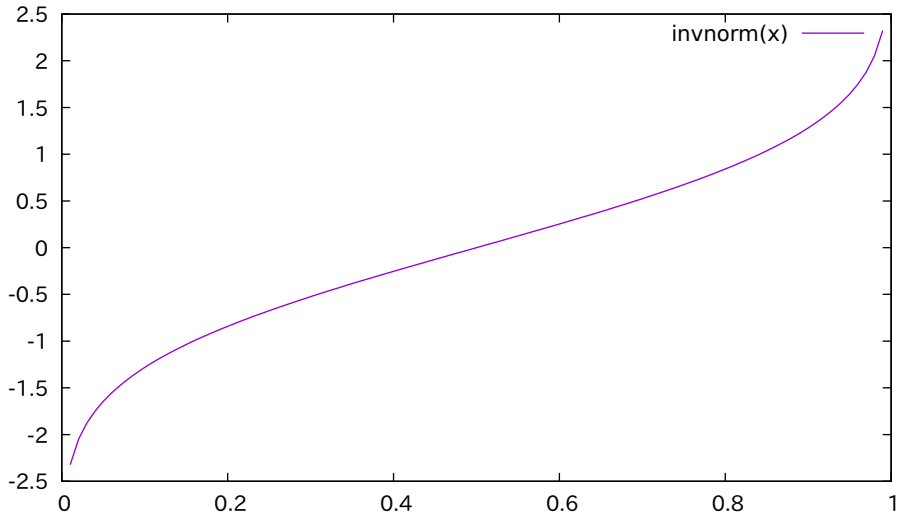
Normal Distribution Function



Inverse Error Function



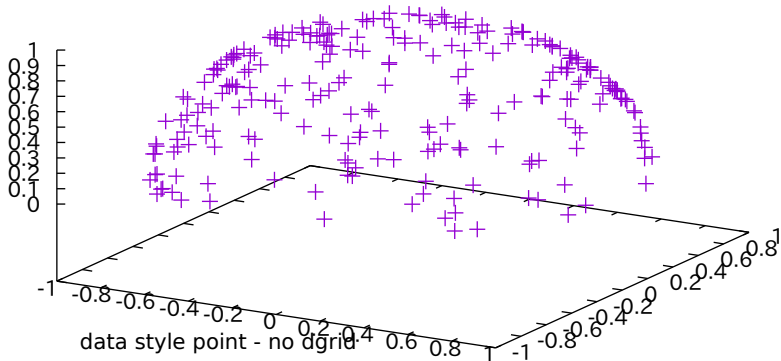
Inverse Normal Distribution Function



Simple demo of scatter data conversion to grid data

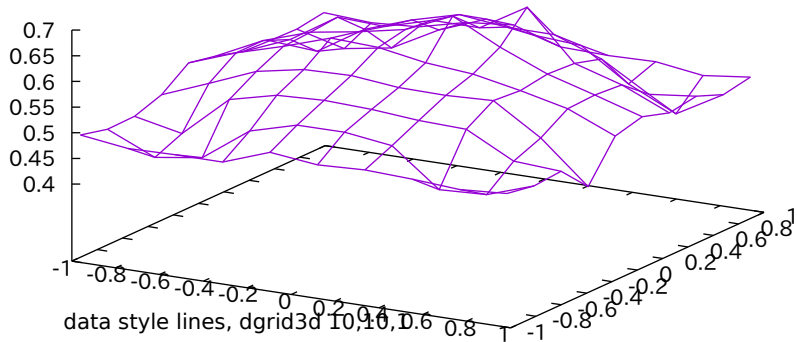
"hemisphr.dat"

+



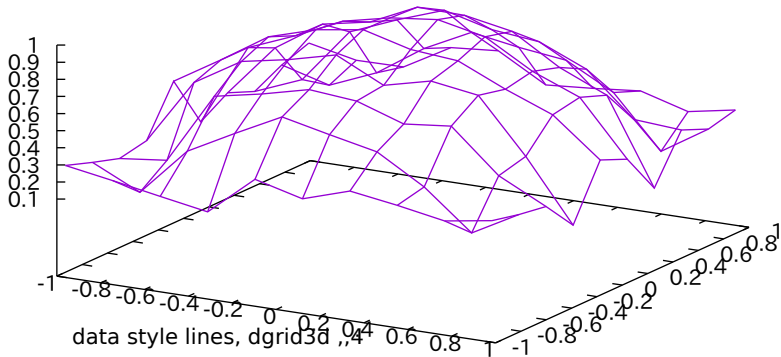
Simple demo of scatter data conversion to grid data

"hemisphr.dat" —



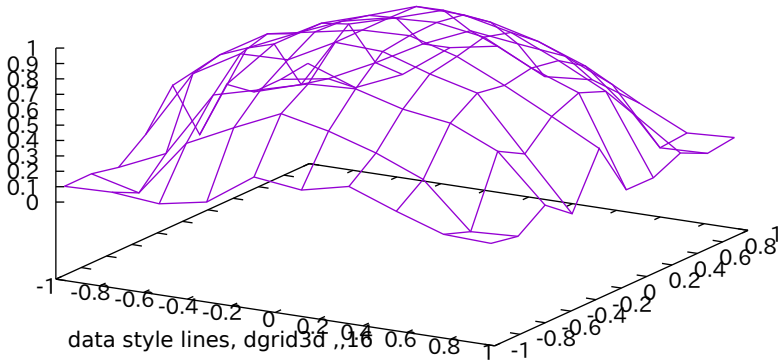
Simple demo of scatter data conversion to grid data

"hemisphr.dat"

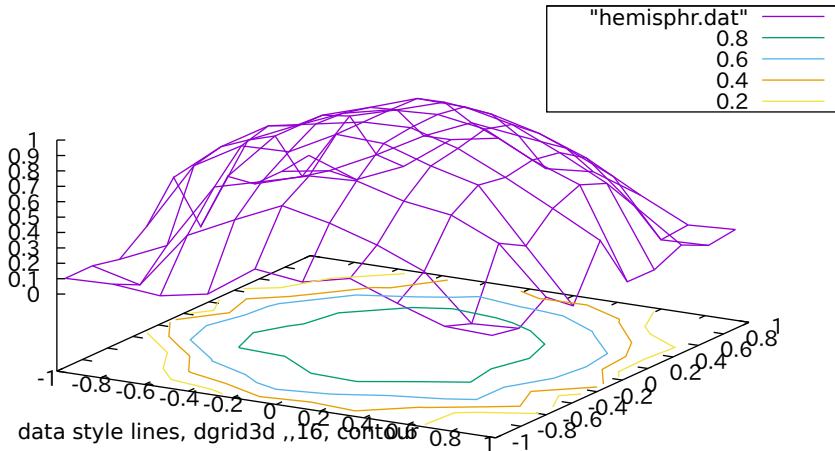


Simple demo of scatter data conversion to grid data

"hemisphr.dat"

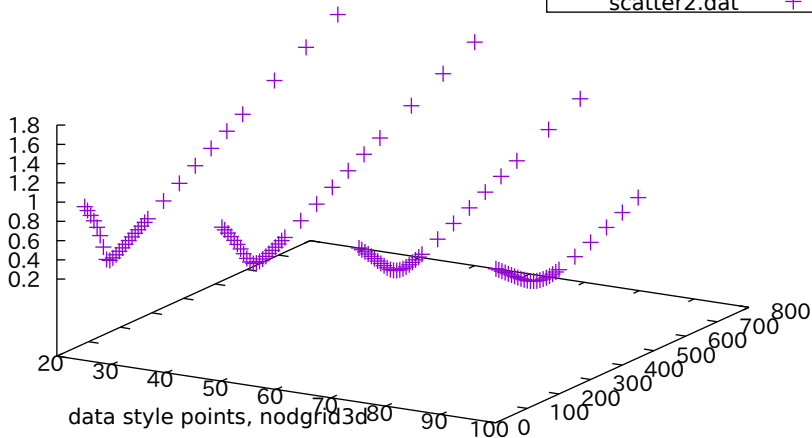


Simple demo of scatter data conversion to grid data

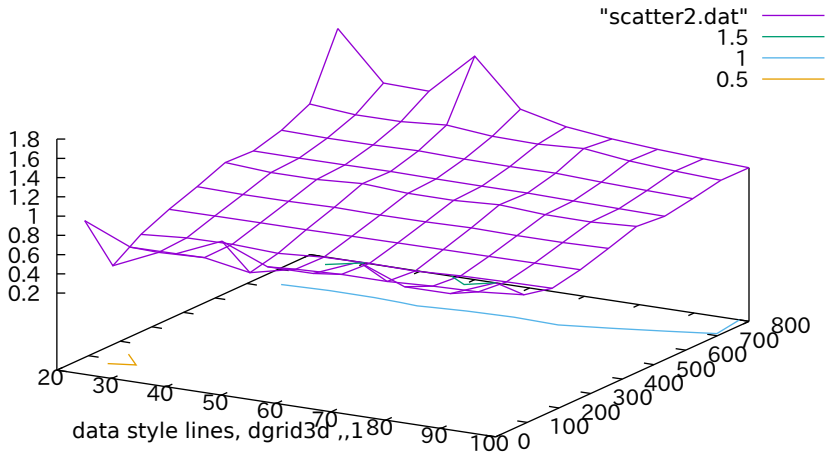


Simple demo of scatter data conversion to grid data

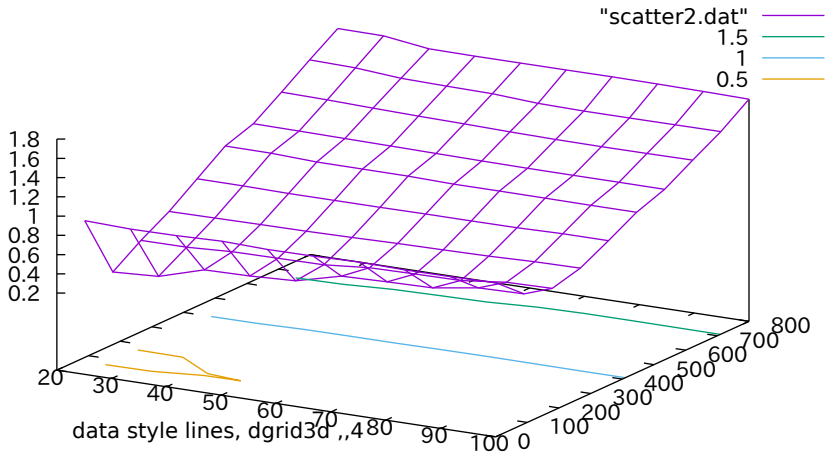
"scatter2.dat" +



Simple demo of scatter data conversion to grid data

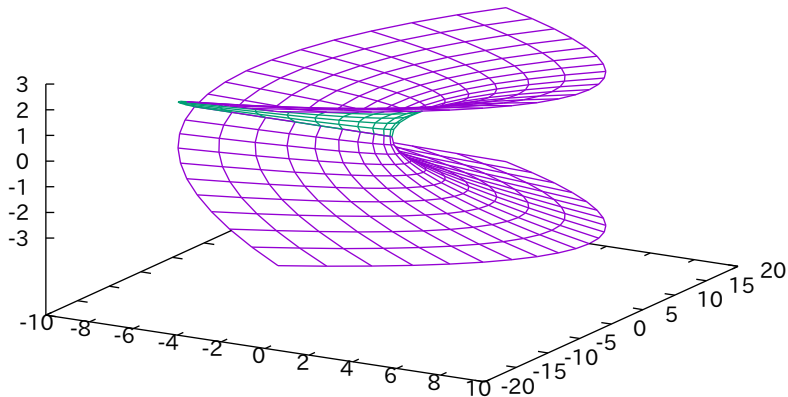


Simple demo of scatter data conversion to grid data



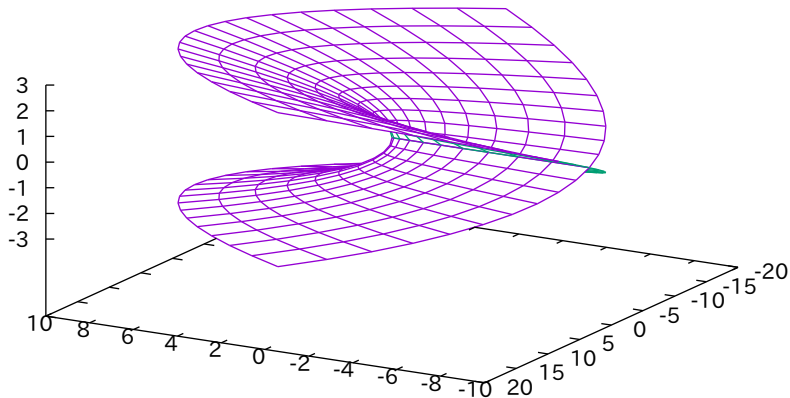
Real part of complex square root function

$$u^2 - v^2, 2uv, u$$



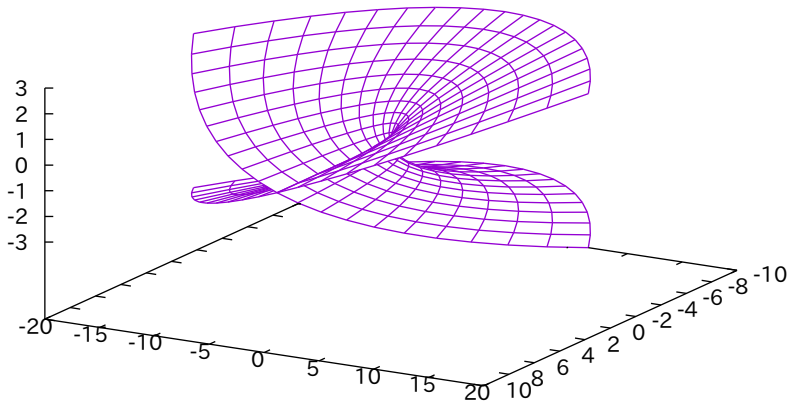
Real part of complex square root function (different view)

$$u^2 - v^2, 2uv, u$$



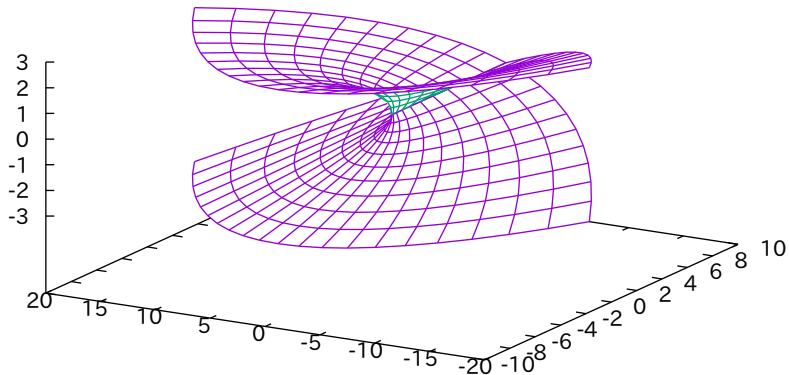
Imaginary part of complex square root function

$$u^2 - v^2, 2uv, v$$



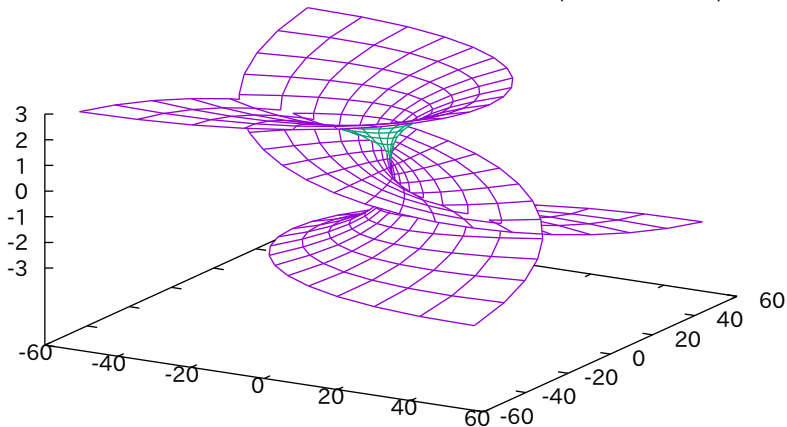
Imaginary part of complex square root function (different view)

$$u^2 - v^2, 2uv, v$$



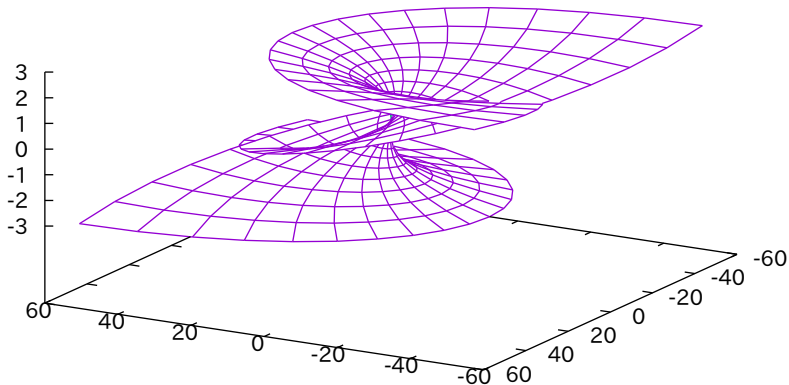
Real part of complex cube root function

$$u^3 - 3uv^2, 3u^2v - v^3, u$$



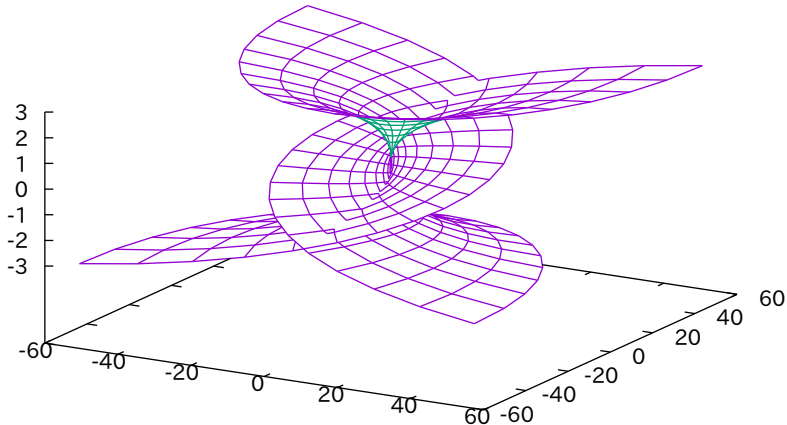
Real part of complex cube root function (different view)

$$u^3 - 3uv^2, 3u^2v - v^3, u$$



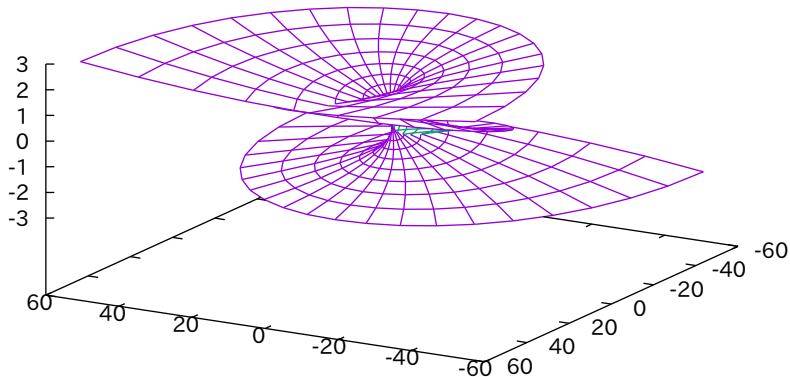
Imaginary part of complex cube root function

$$u^3 - 3uv^2, 3u^2v - v^3, v$$



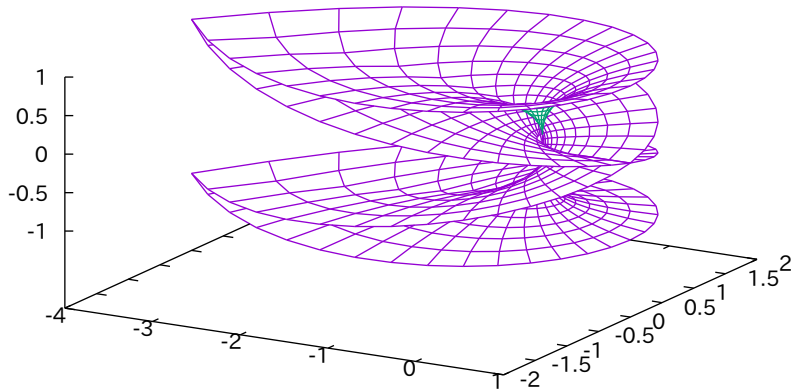
Imaginary part of complex cube root function (different view)

$$u^3 - 3uv^2, 3u^2v - v^3, v$$



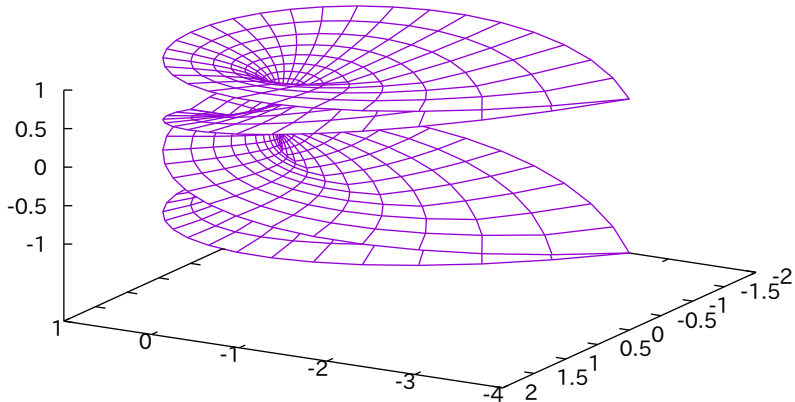
Real part of complex 4th root function

$$u^4 - 6u^2v^2 + v^4, 4u^3v - 4uv^3, u$$



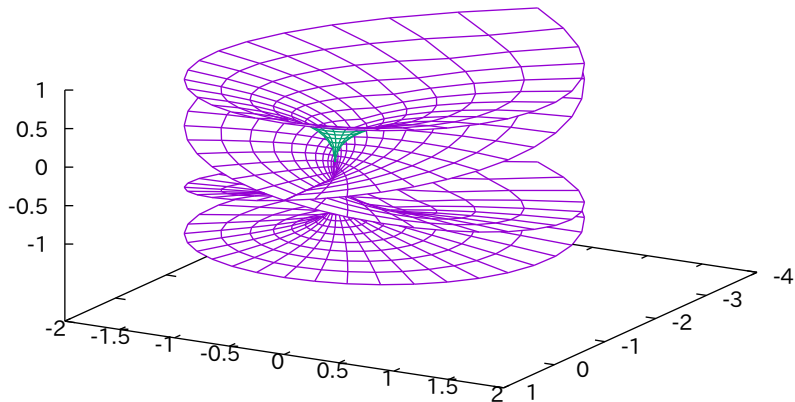
Real part of complex 4th root function (different view)

$$u^{**4}-6*u^{**2}*v^{**2}+v^{**4}, 4*u^{**3}*v-4*u*v^{**3}, u$$



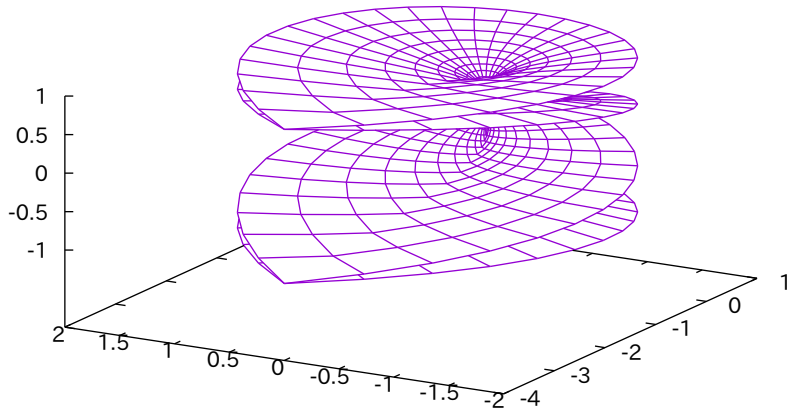
Imaginary part of complex 4th root function

$$u^{**4}-6*u^{**2}*v^{**2}+v^{**4}, 4*u^{**3}*v-4*u*v^{**3}, v$$



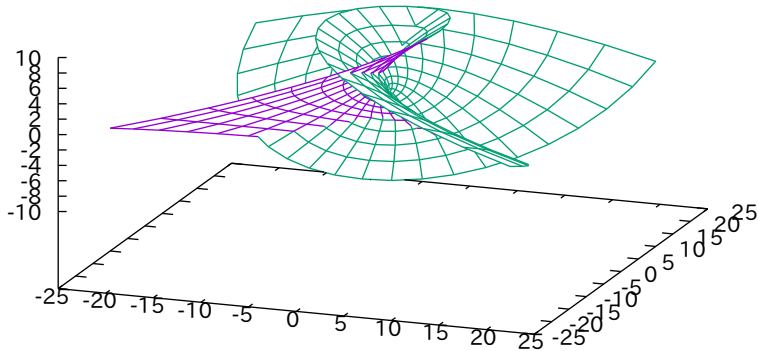
Imaginary part of complex 4th root function (different view)

$$u^{**4}-6*u^{**2}*v^{**2}+v^{**4}, 4*u^{**3}*v-4*u*v^{**3}, v$$



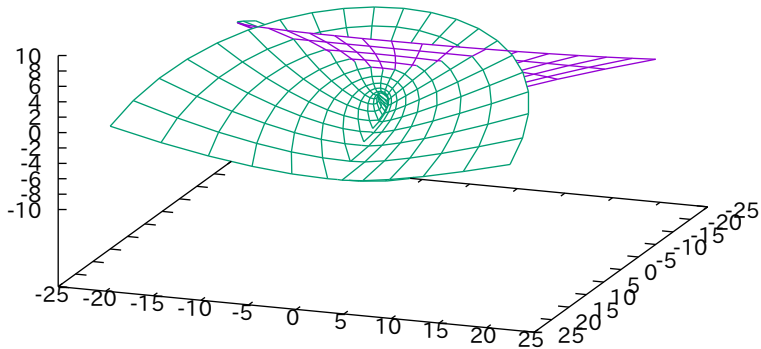
Enneper's surface

$$u - \frac{u^3}{3} + u v^2, v - \frac{v^3}{3} + v u^2, u^2 - v^2$$



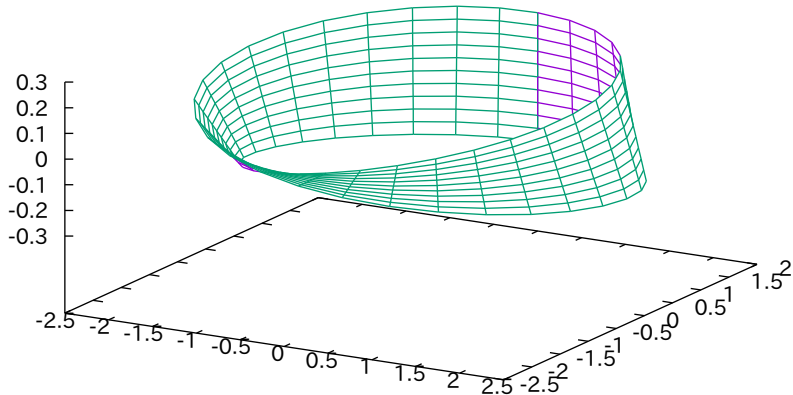
Enneper's surface (different view)

$$u - \frac{u^3}{3} + u v^2, v - \frac{v^3}{3} + v u^2, u^2 - v^2$$



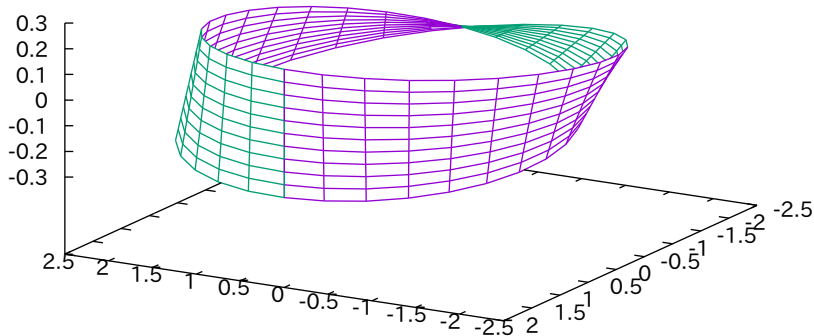
Moebius strip

$$(2-v*\sin(u/2))*\sin(u), (2-v*\sin(u/2))*\cos(u), v*\cos(u/2)$$

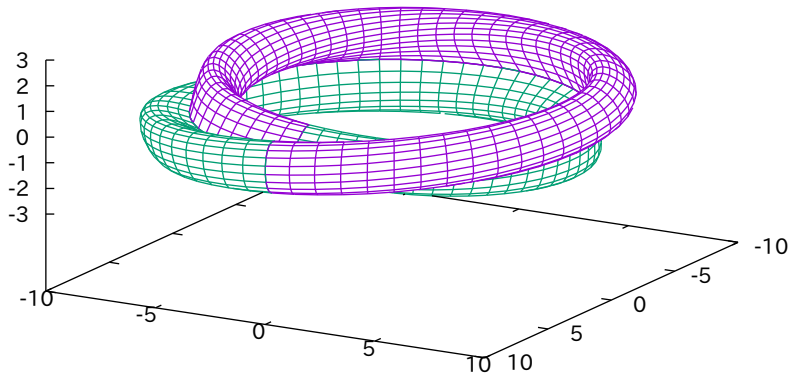


Moebius strip (view from opposite side)

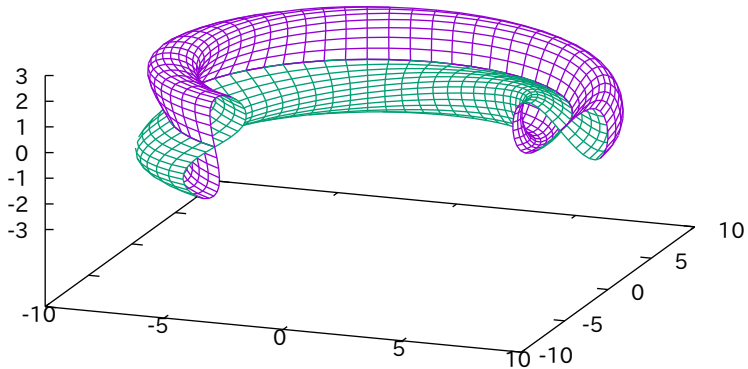
$$(2-v*\sin(u/2))*\sin(u), (2-v*\sin(u/2))*\cos(u), v*\cos(u/2)$$



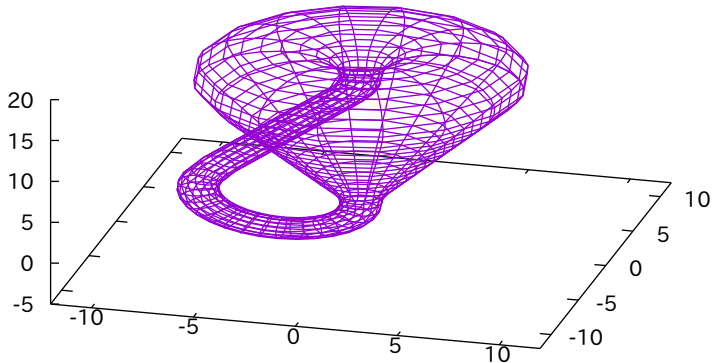
Klein bottle



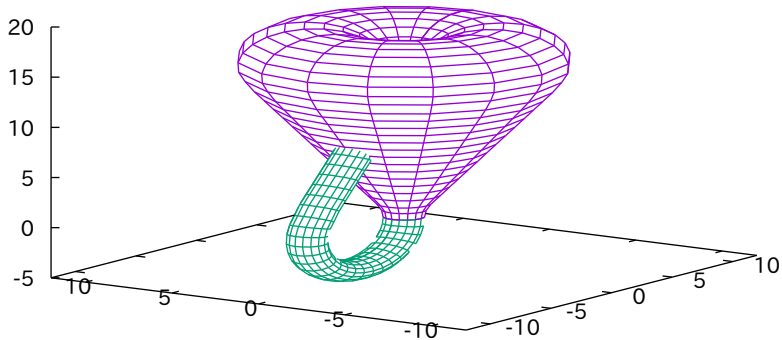
Klein bottle with look at the 'inside'



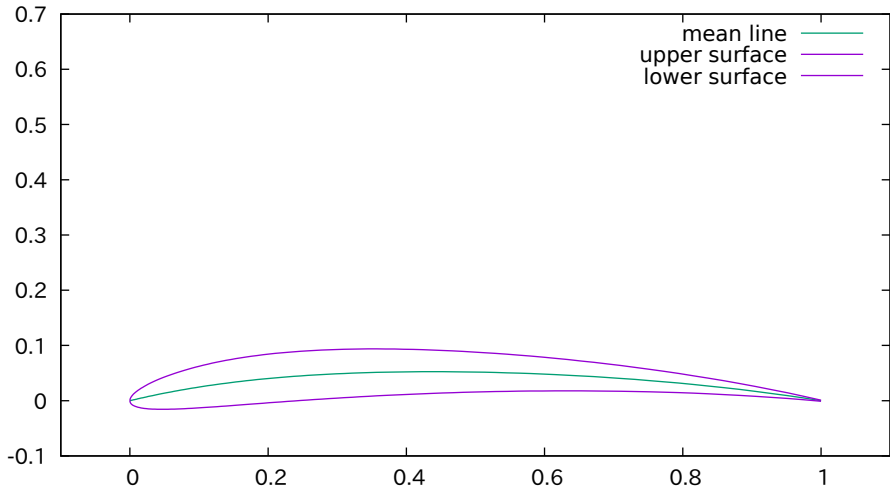
Klein bottle, glassblowers' version (look-through)



Klein bottle, glassblowers' version (solid)

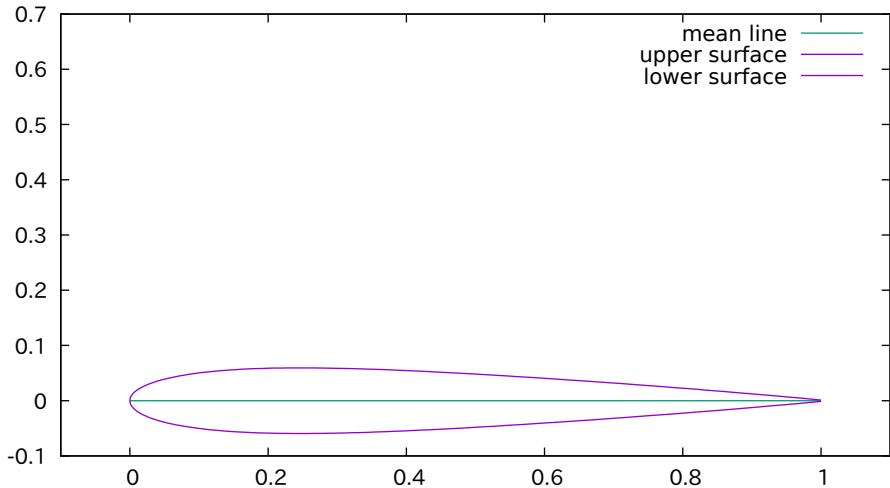


NACA6409 Airfoil



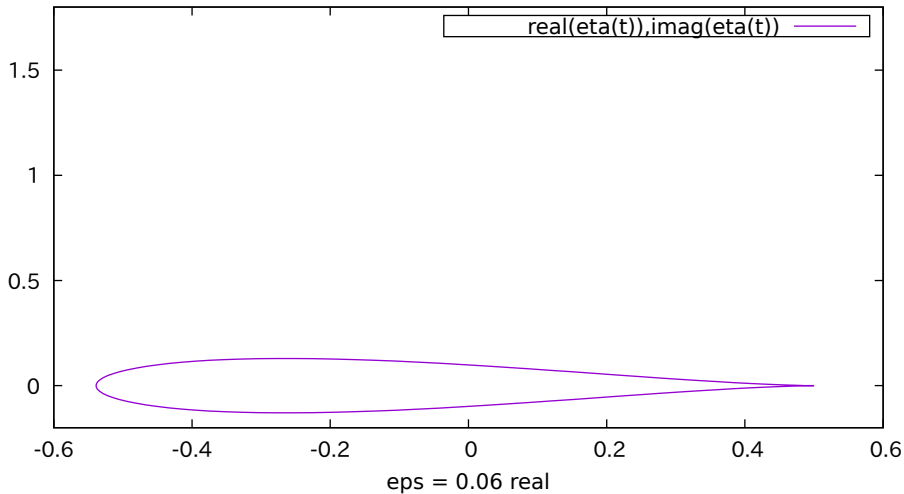
NACA6409 -- 9% thick, 40% max camber, 6% camber

NACA0012 Airfoil

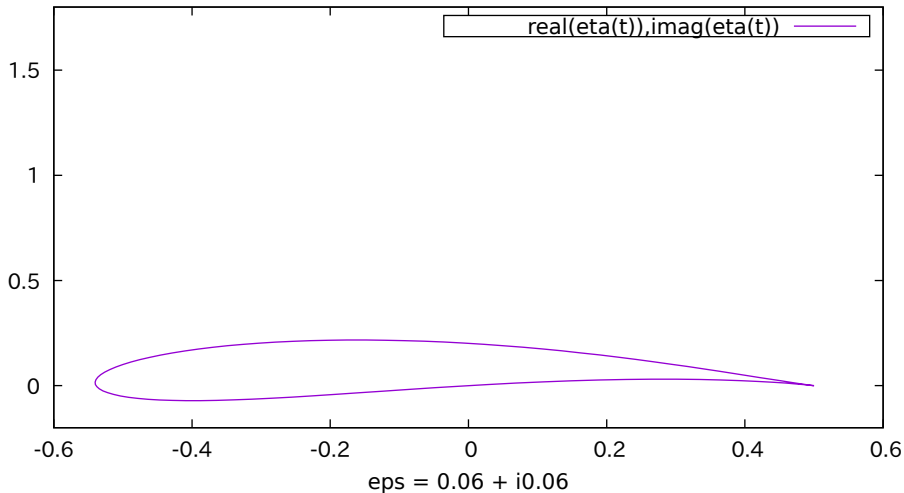


12% thick, no camber -- classical test case

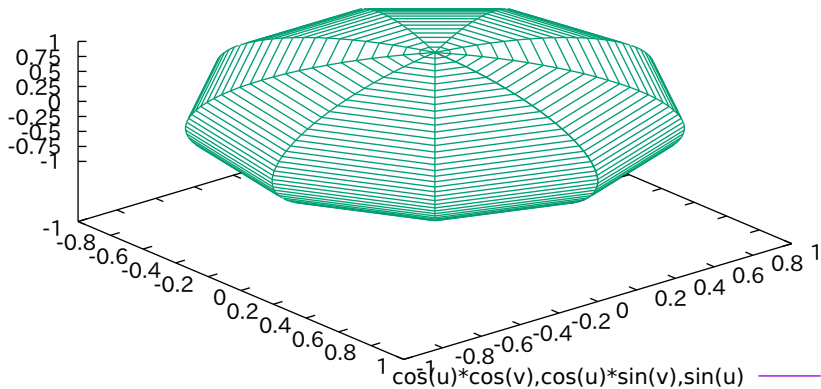
Joukowski Airfoil using Complex Variables



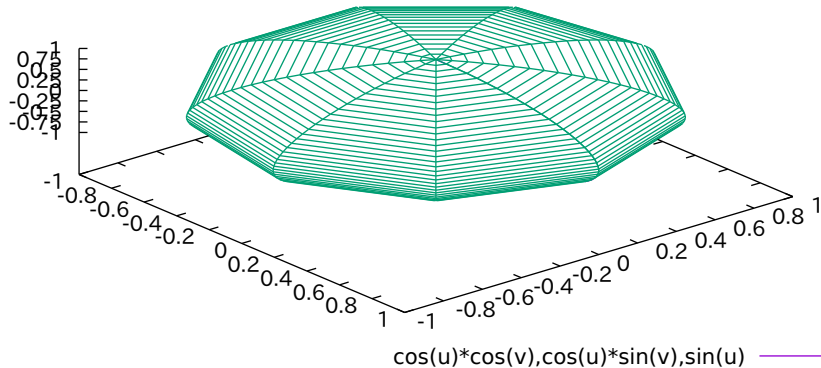
Joukowski Airfoil using Complex Variables



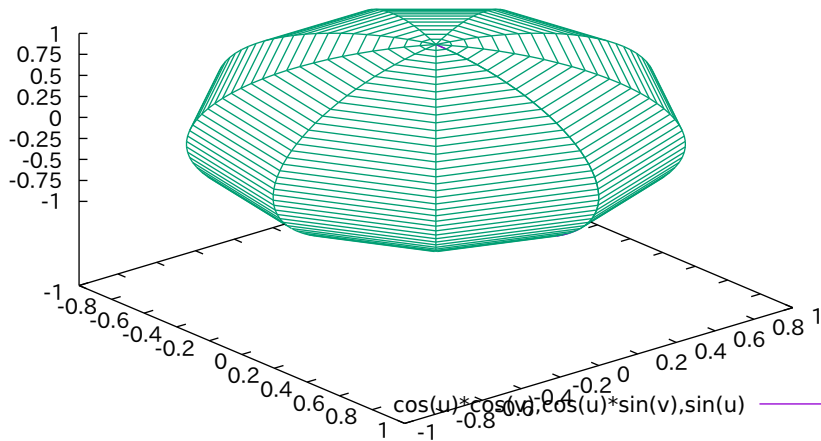
Parametric Sphere



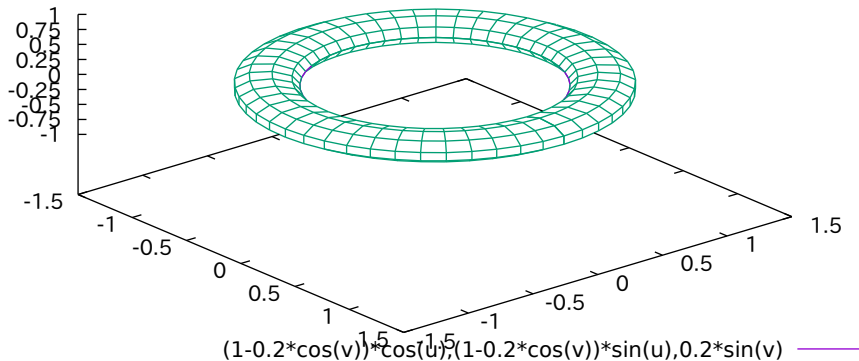
Parametric Sphere, crunched z axis



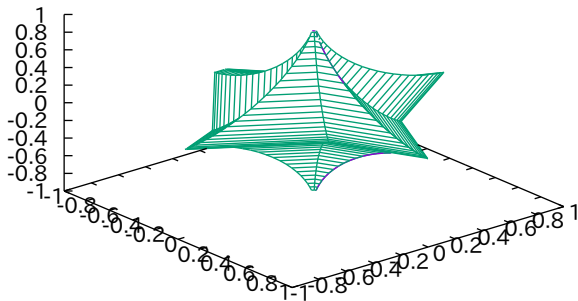
Parametric Sphere, enlarged z axis



Parametric Torus



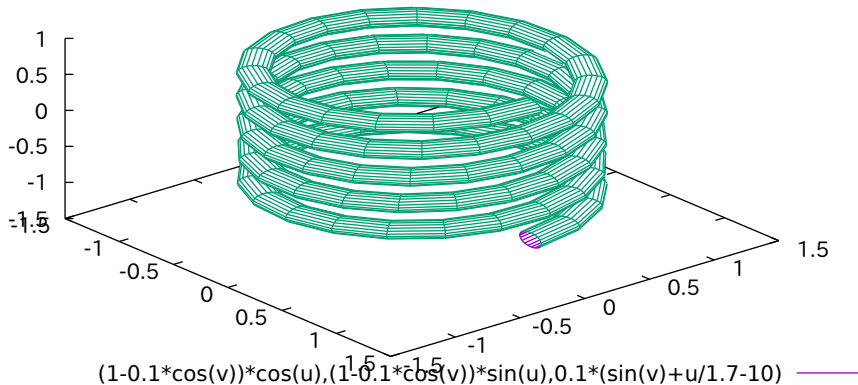
Parametric Hexagon



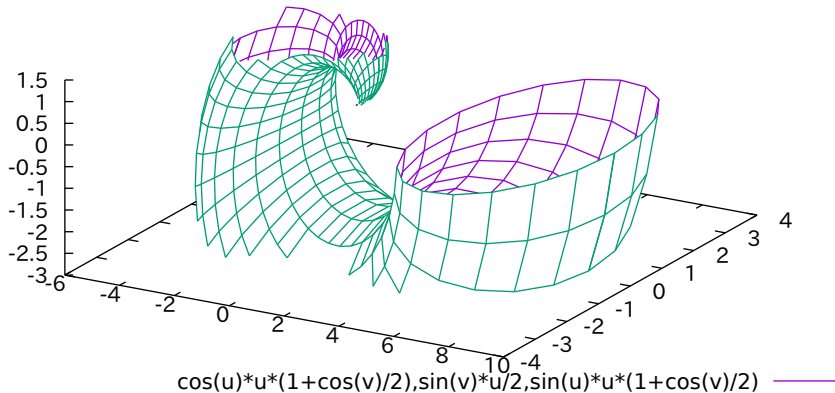
$$\cos(v)^3 \cos(u)^3, \sin(v)^3 \cos(u)^3, \sin(u)^3$$



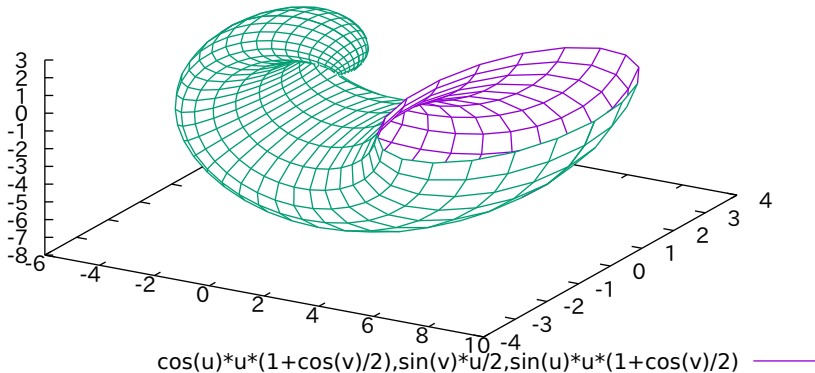
Parametric Helix



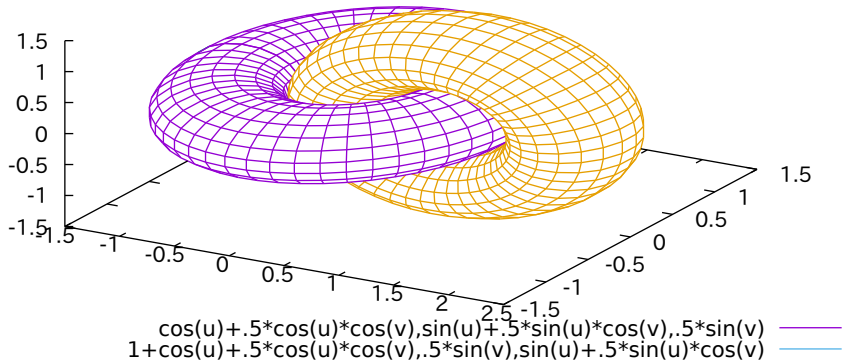
Parametric Shell (clipped to limited z range)



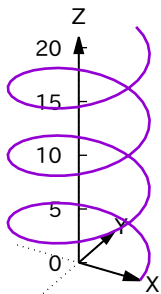
Parametric Shell (automatic z range)



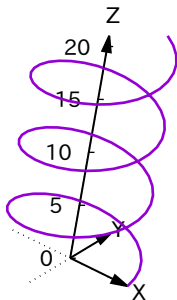
Interlocking Tori



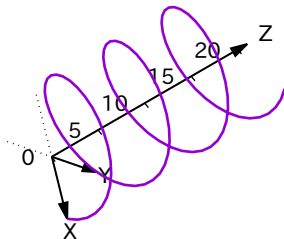
azimuth 0



azimuth 10

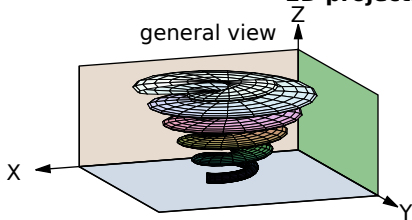


azimuth 60

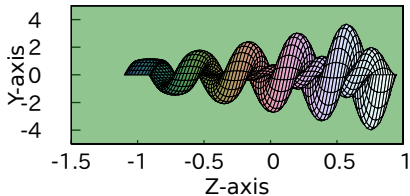


2D projections of a 3D surface

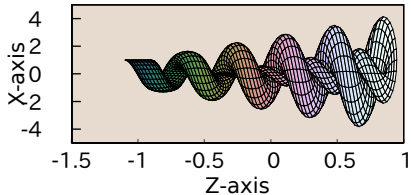
general view



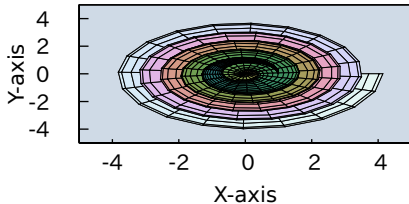
set view projection yz



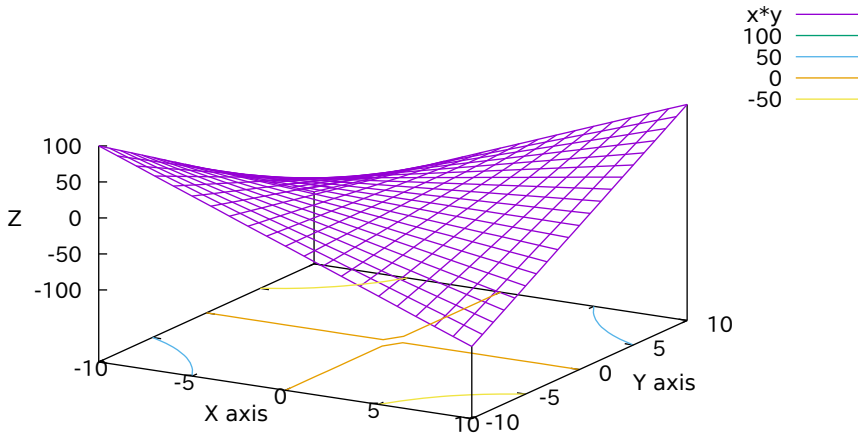
set view projection xz



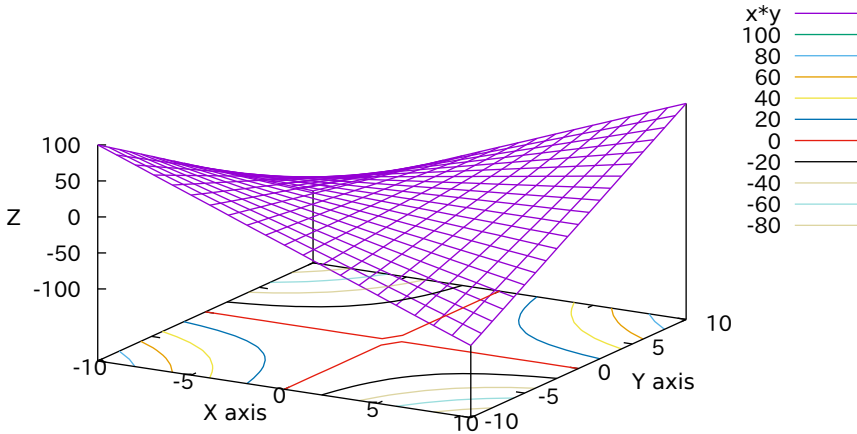
set view map



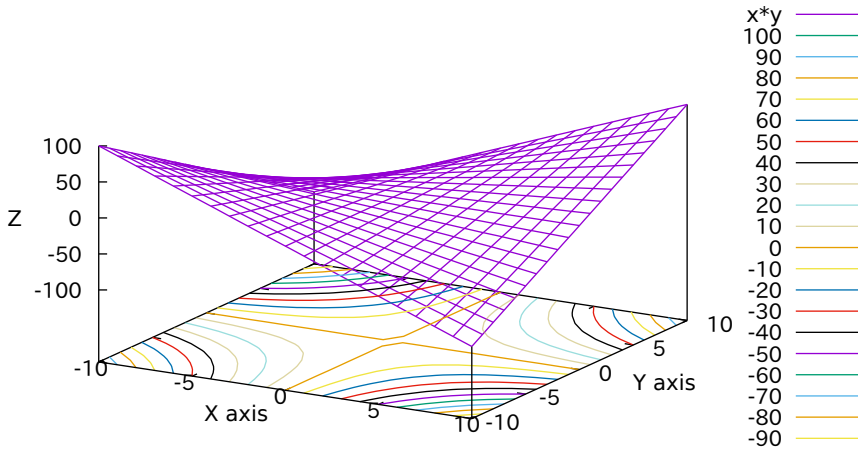
contour plot



more contours (15 levels)

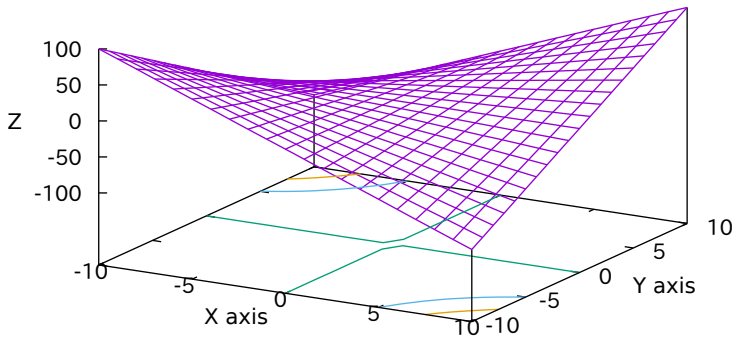


contour by increments (every 10, starting at -100)

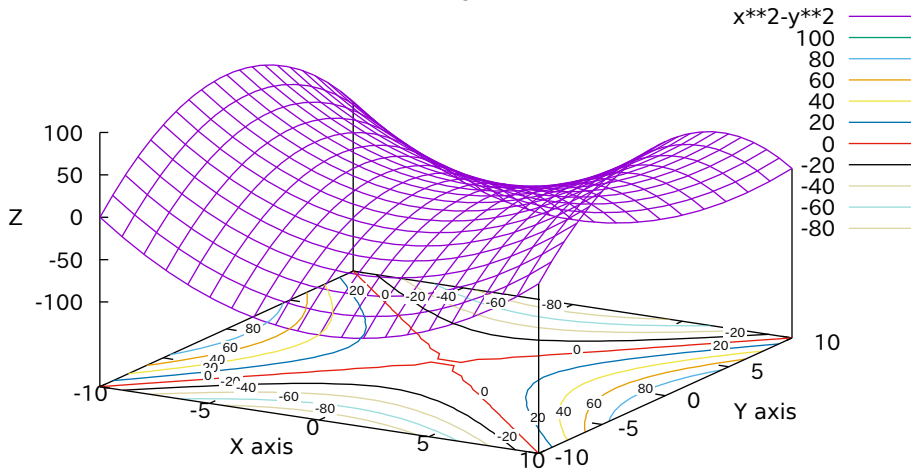


discrete set of contours (at -75, -50, 0)

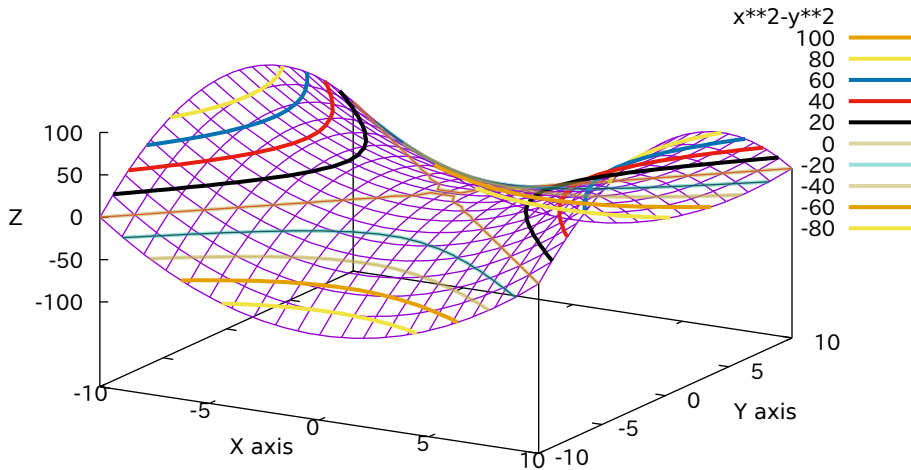
$x*y$ ———
0 ———
-50 ———
-75 ———



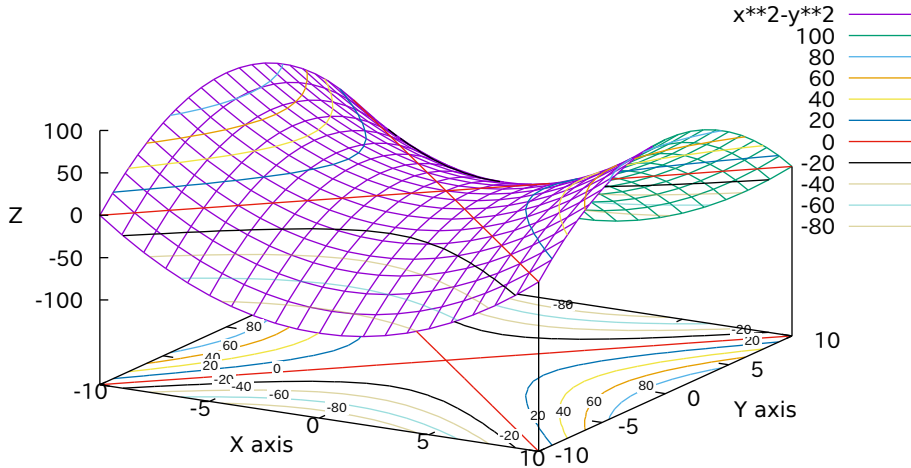
contours on base grid with labels



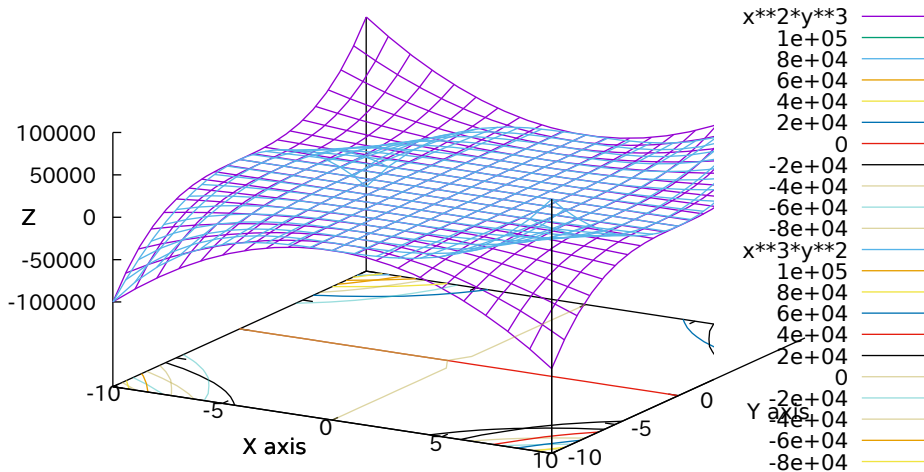
contours drawn on surface



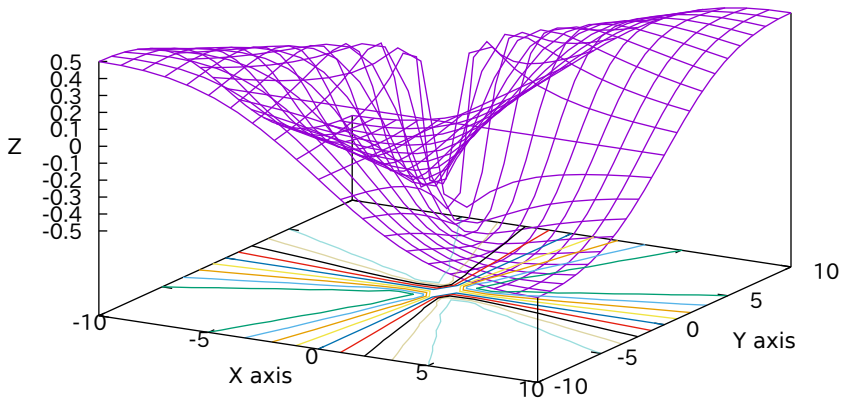
contours on both base and surface



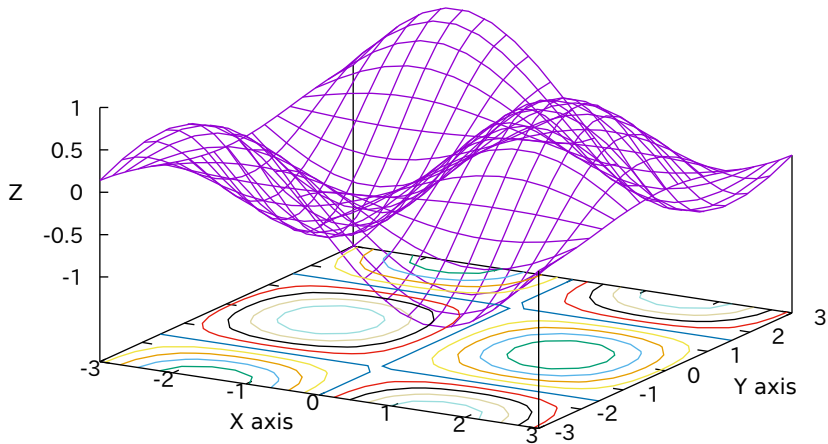
2 surfaces



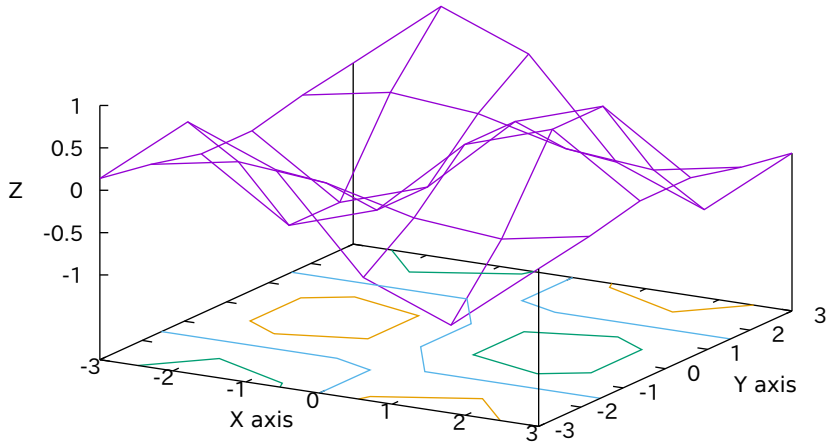
some more interesting contours



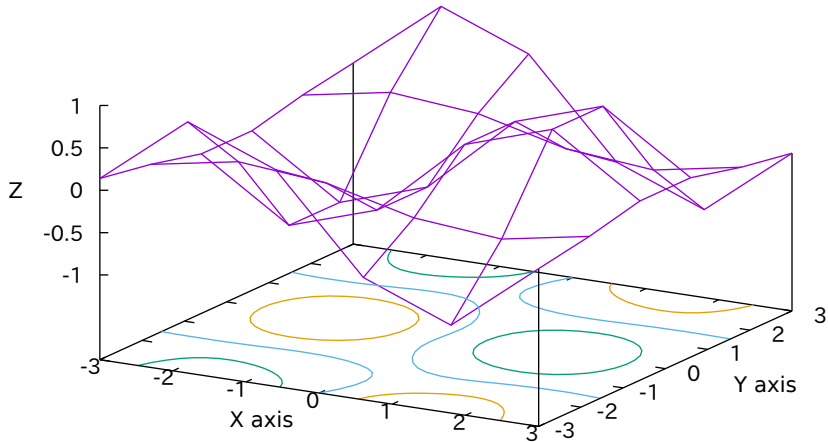
some more interesting contours



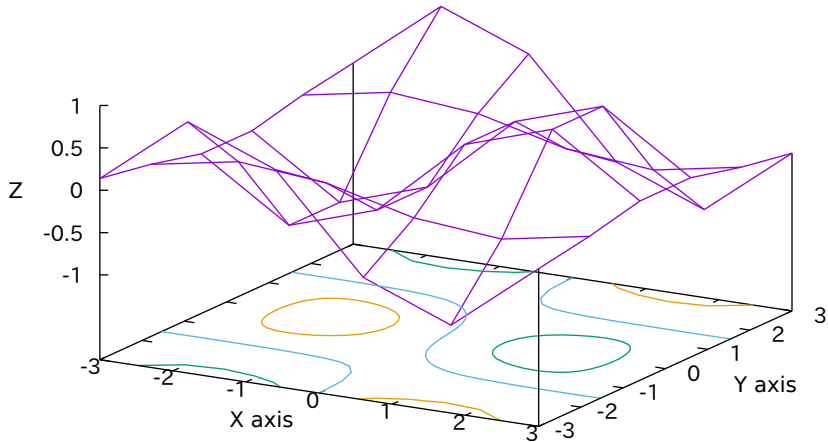
low resolution (6x6)



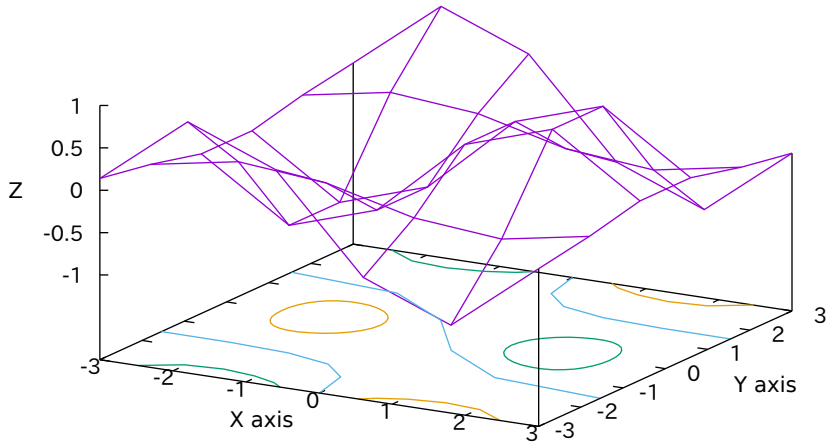
low resolution (6x6) using cubic splines



low resolution (6x6) using bspline approx.



low resolution (6x6) raise bspline order.

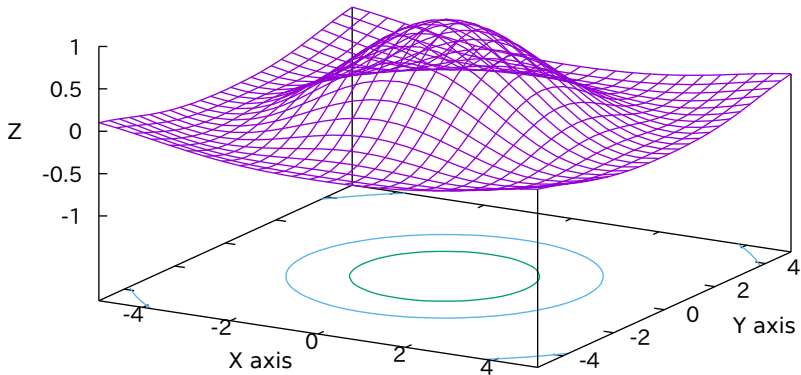


contour of Sinc function

$$\frac{\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$$

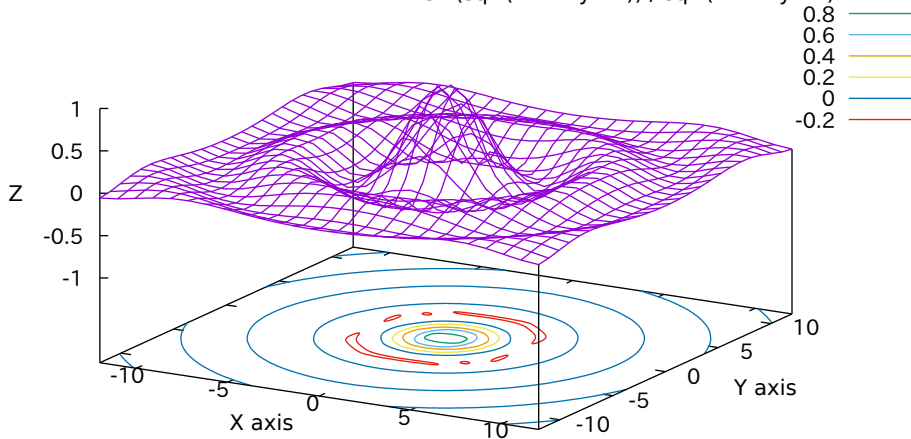
0.5

0

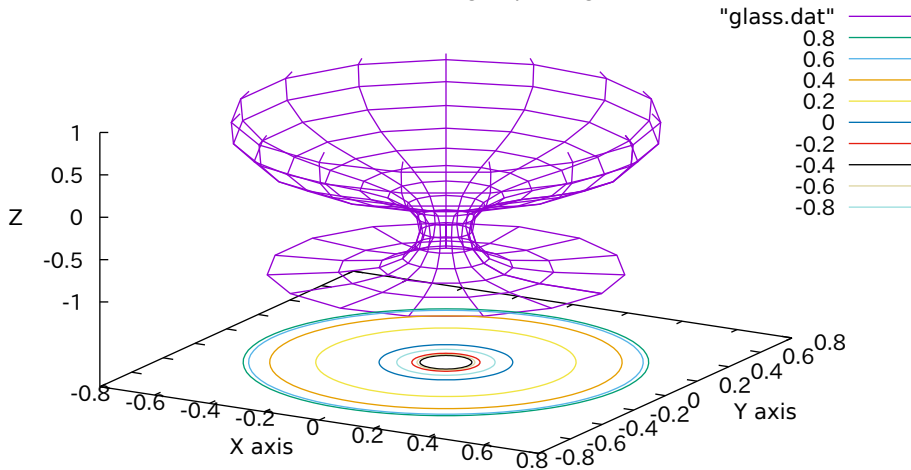


contour of Sinc function

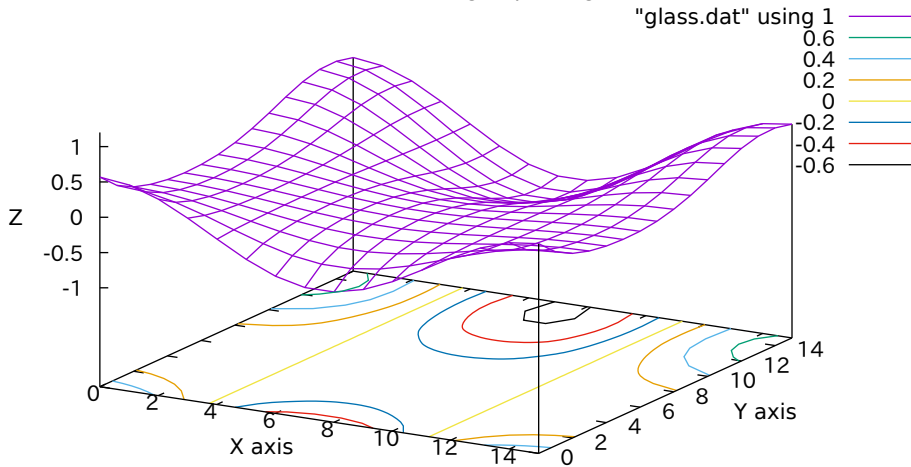
$$\sin(\sqrt{x^2+y^2}) / \sqrt{x^2+y^2}$$



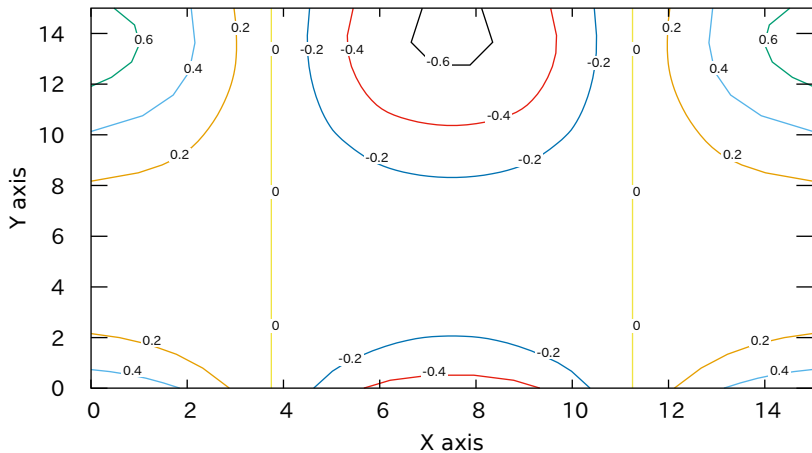
contour of data grid plotting



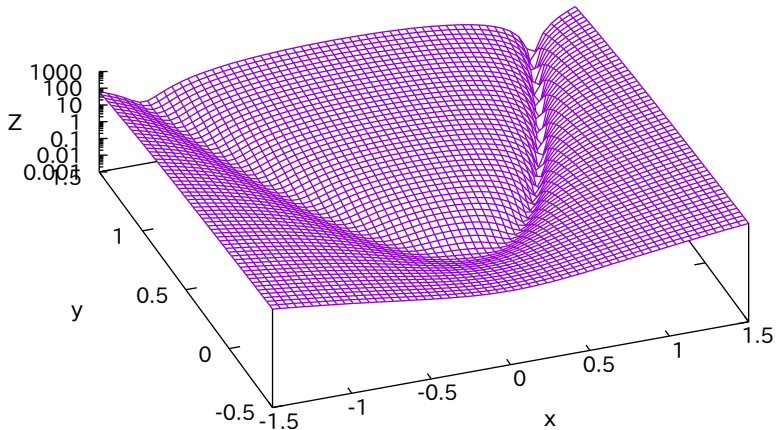
contour of data grid plotting



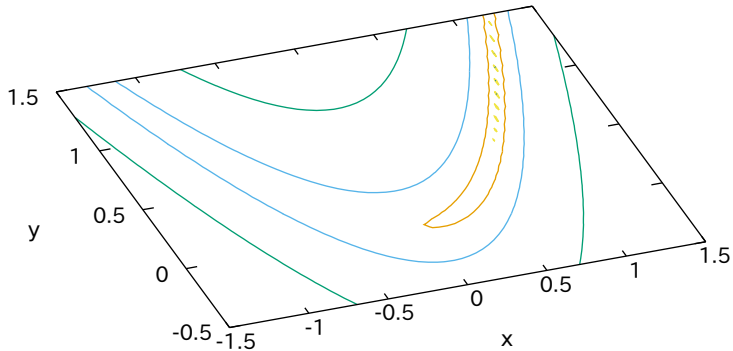
2D contour projection of previous plot



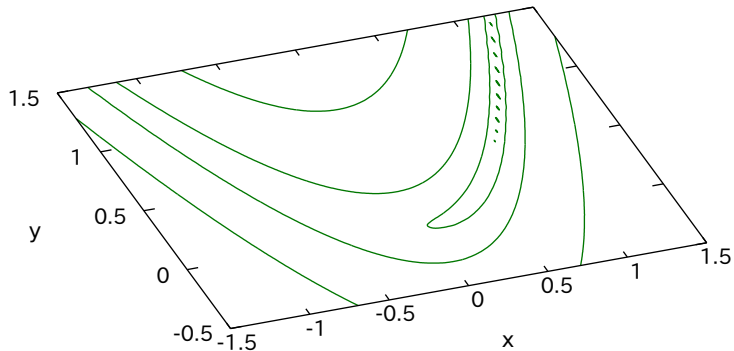
Rosenbrock Function



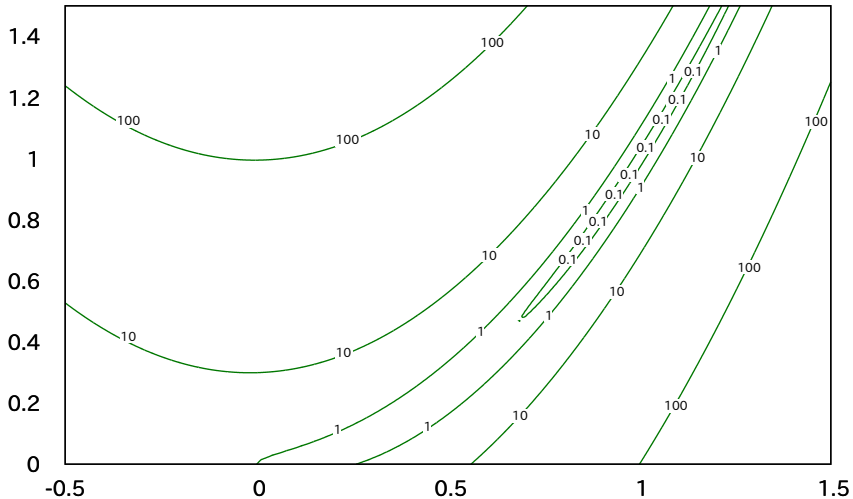
Rosenbrock Function

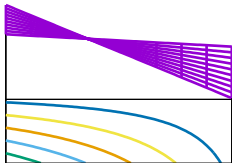


All contours drawn in a single color



Sometimes it helps to use multiplot





default

$x*y$

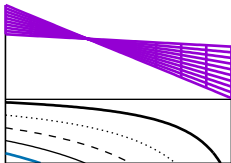
50

40

30

20

10



firstline sort

$x*y$

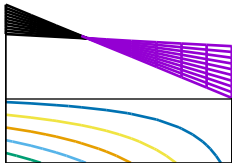
10

20

30

40

50



hidden3d

$x*y$

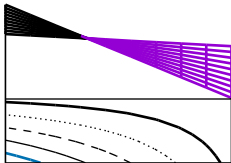
50

40

30

20

10



$x*y$

10

20

30

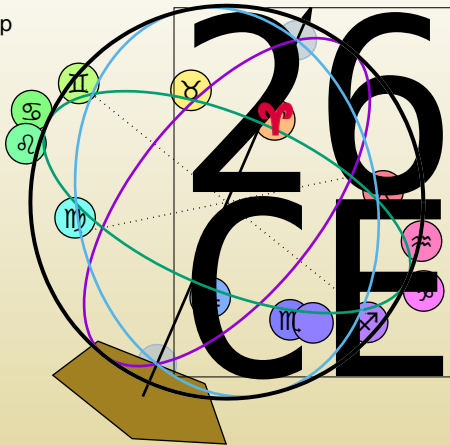
40

50

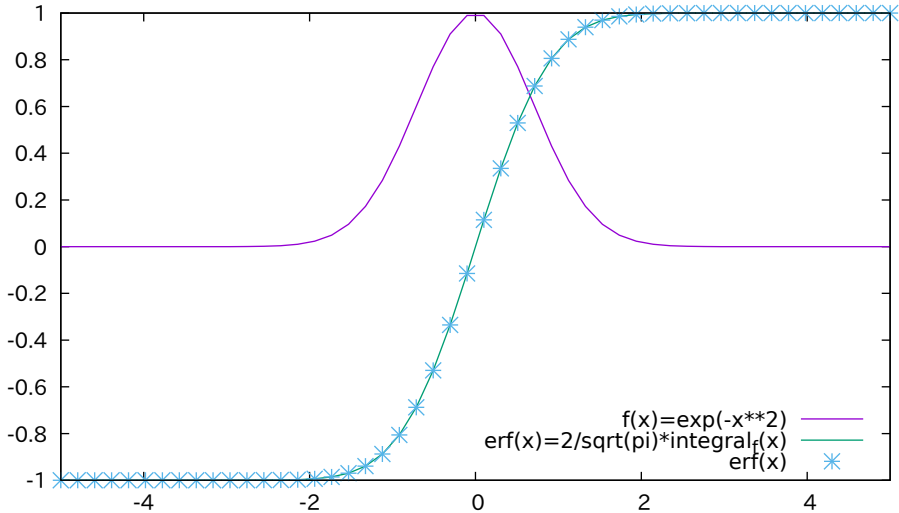
Circle and polygon objects in 3D

Pixmap use

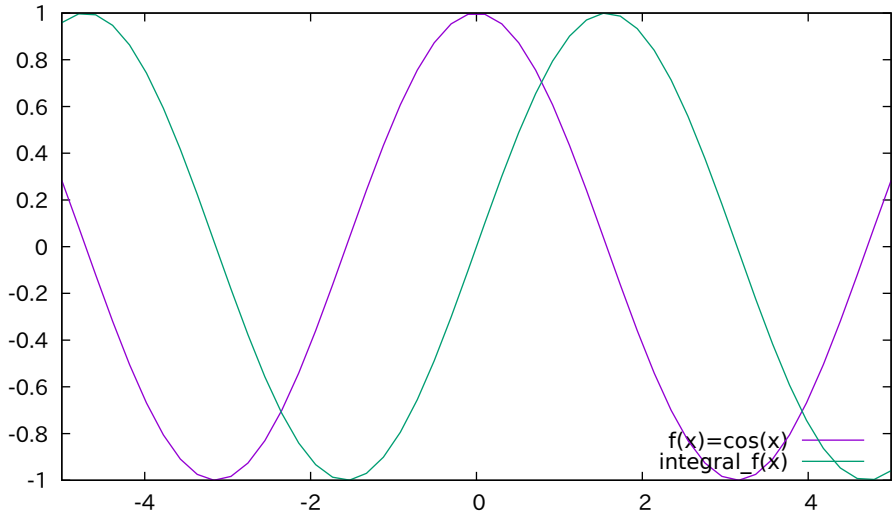
- Gradient used as a backdrop
- Project logo bottom left
- Icon (Aries) as plot element



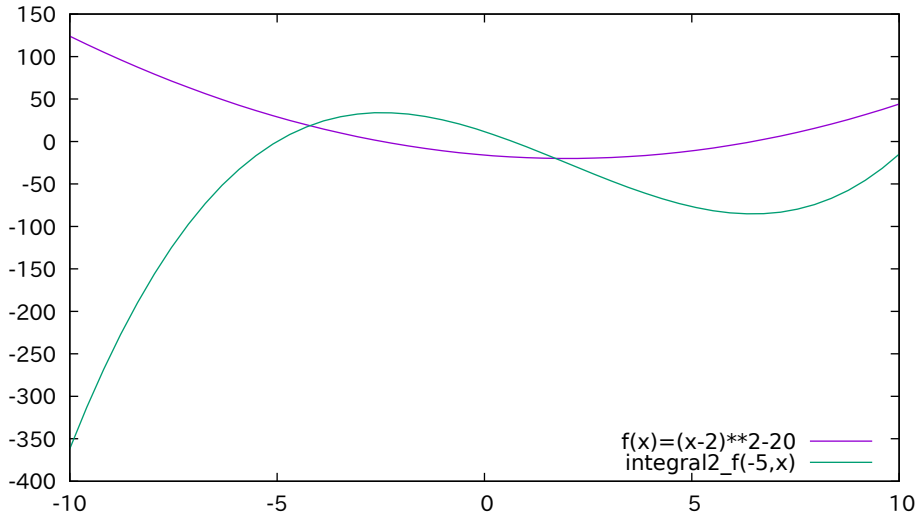
approximate the integral of functions



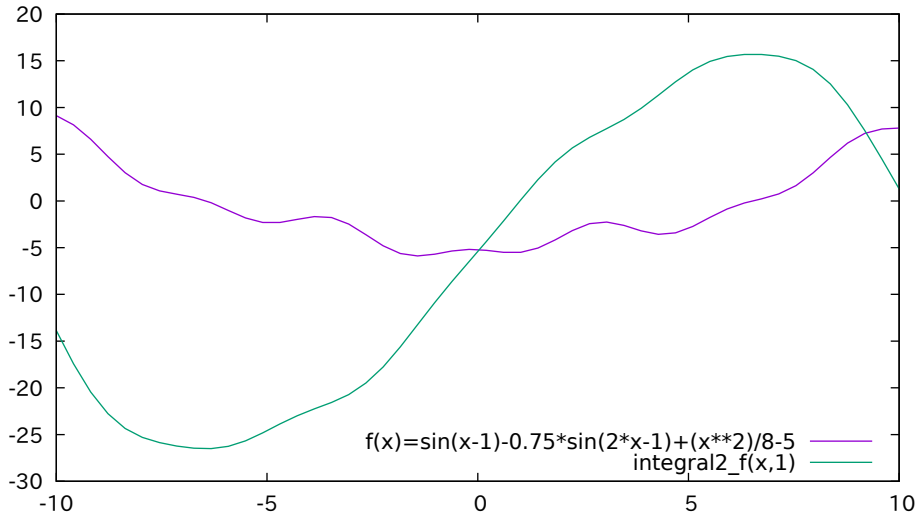
approximate the integral of functions



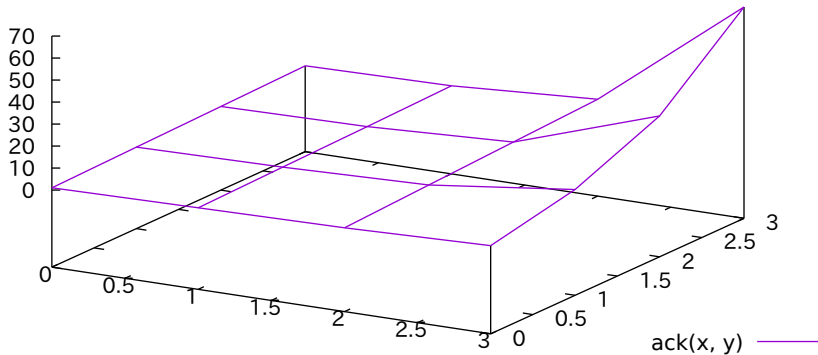
approximate the integral of functions (upper and lower limits)



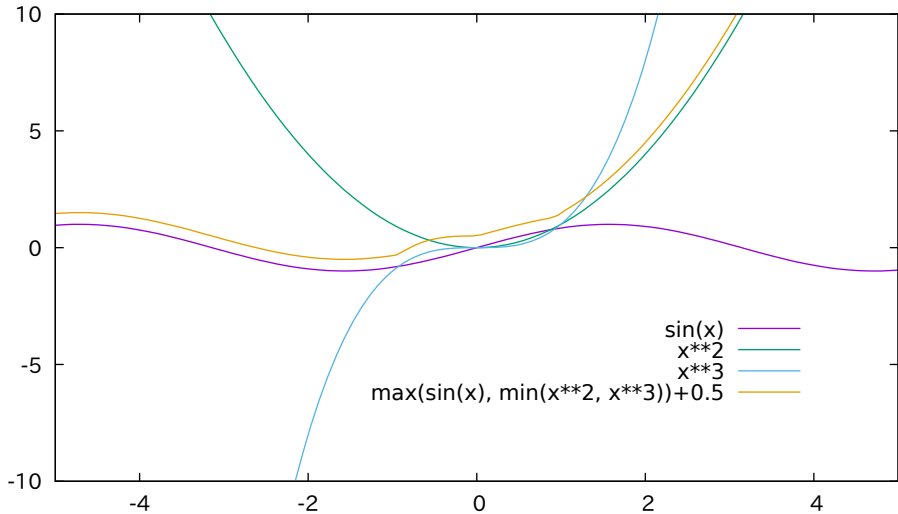
approximate the integral of functions (upper and lower limits)



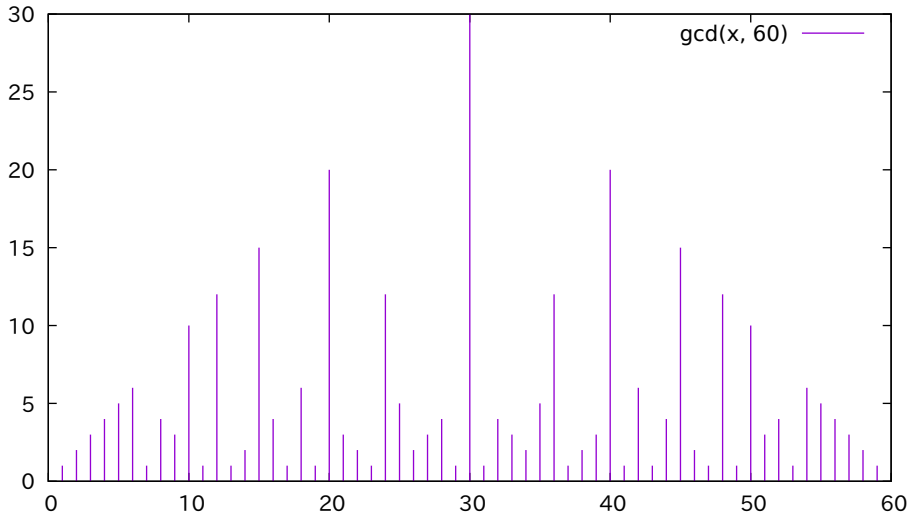
Plot of the ackermann function



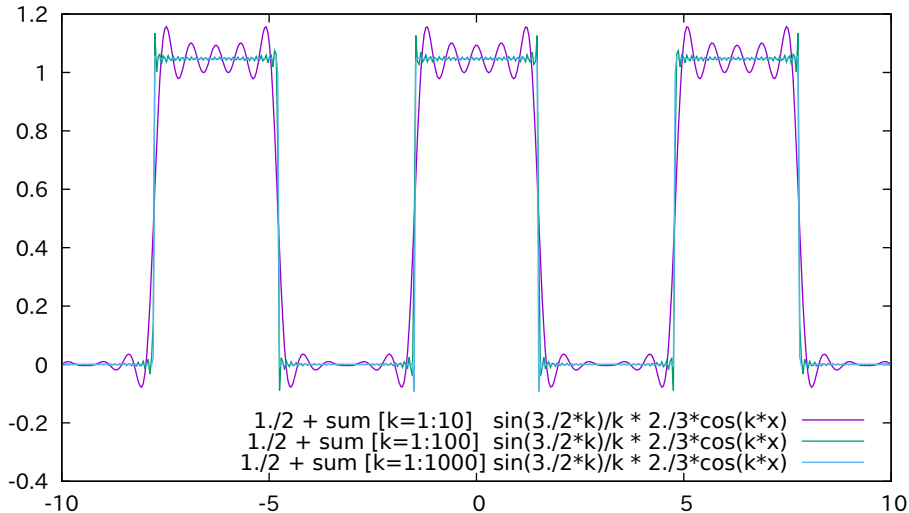
Min(x,y) and Max(x,y)



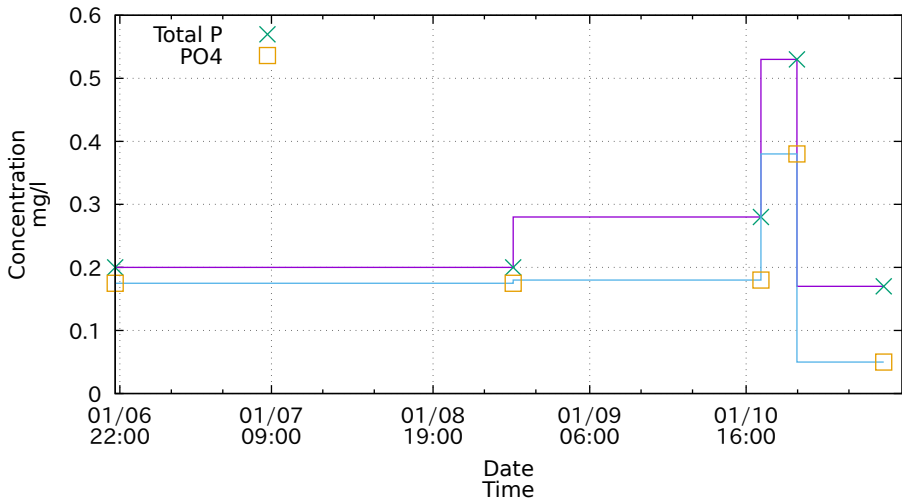
Greatest Common Divisor (for integers only)



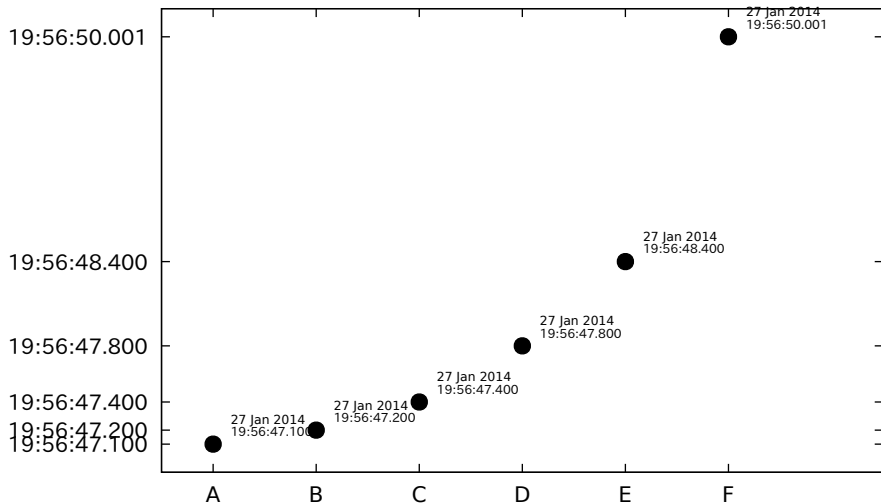
Finite summation of 10, 100, 1000 fourier coefficients



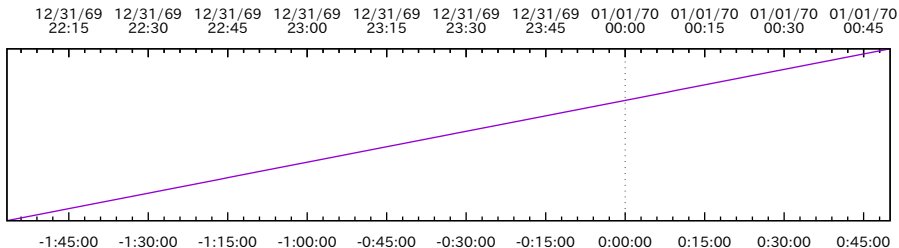
Fsteps plot
with date and time as x-values



Time data on Y, millisecond precision

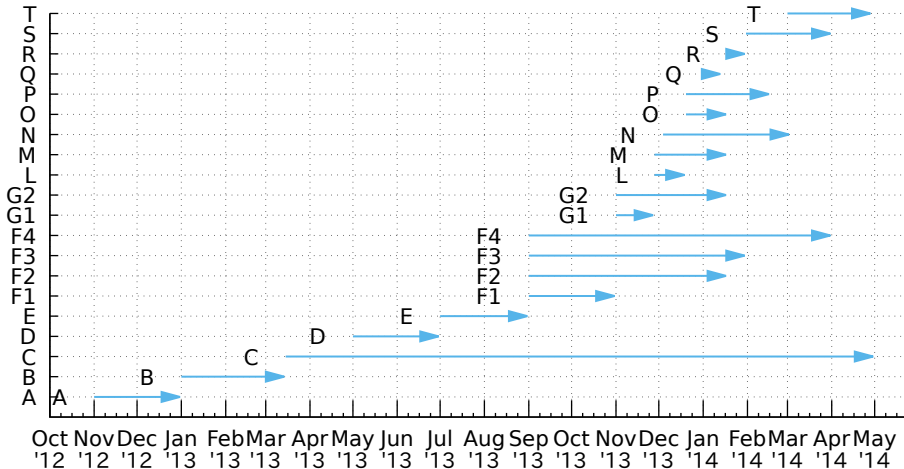


Date format (top) vs Time format (bottom)



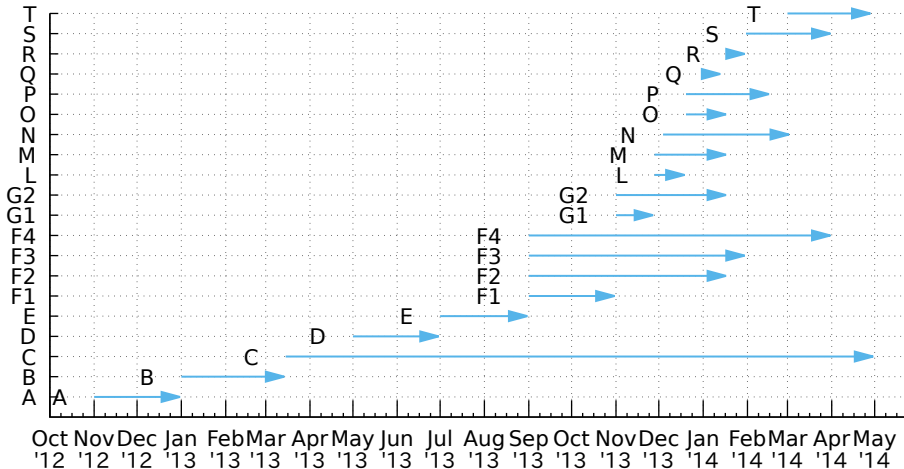
Simple Gantt Chart

Task start and end times in columns 2 and 3

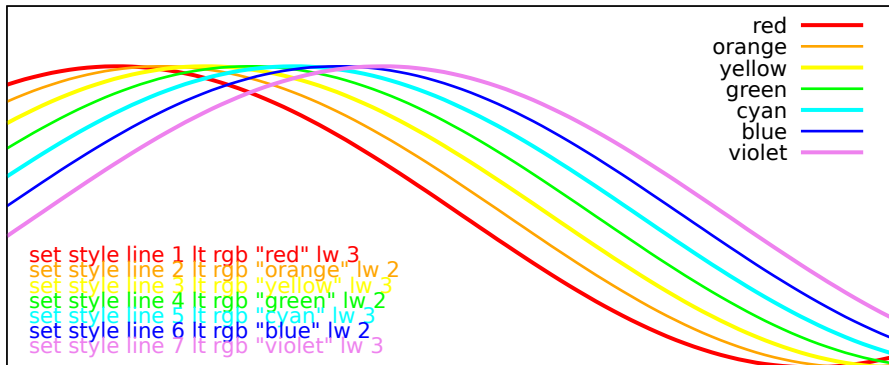


Simple Gantt Chart

Task start and end times in columns 2 and 3

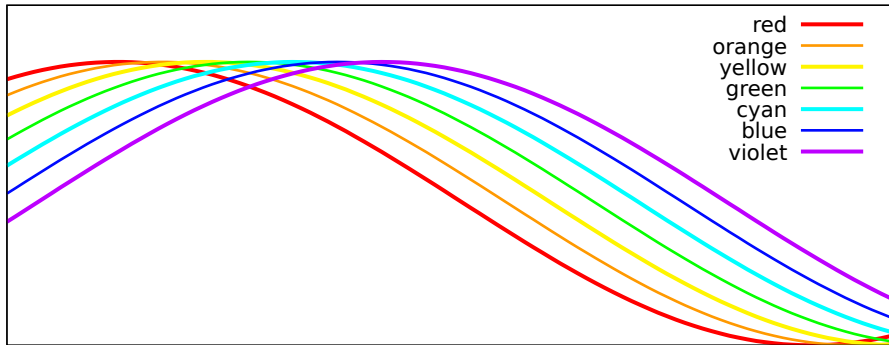


Terminal-independent RGB colors in 2D

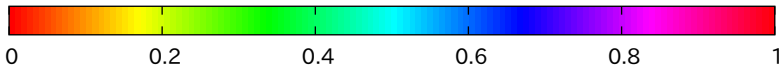


Implemented using built-in rgb color names
(only works for terminals that can do full rgb color)

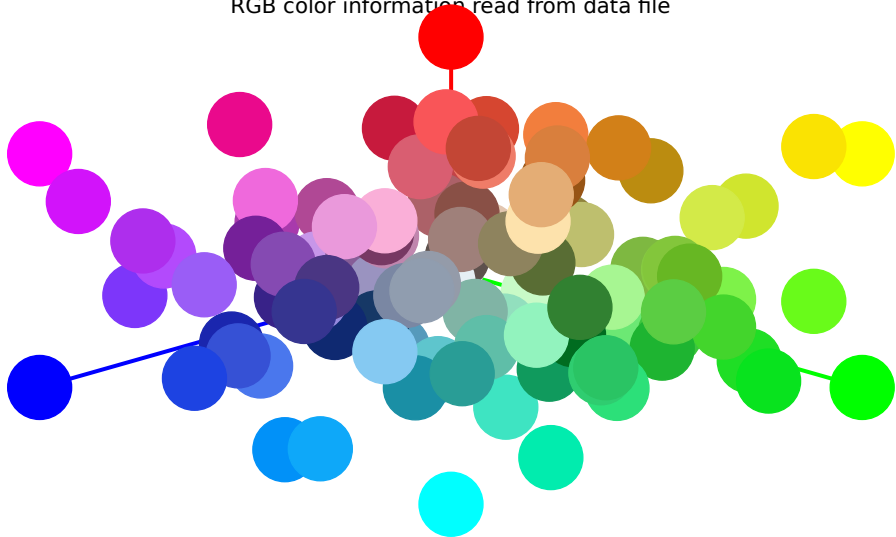
Terminal-independent palette colors in 2D
Implemented using command line macros referring to a fixed HSV palette



HSV color wheel



RGB color information read from data file



Both RGB color information
and point size controlled by input



Both RGB color information
and point size controlled by input

rgb.dat using 1:2:3:(rgb(\$1,\$2,\$3

0xff



Blue

25

200

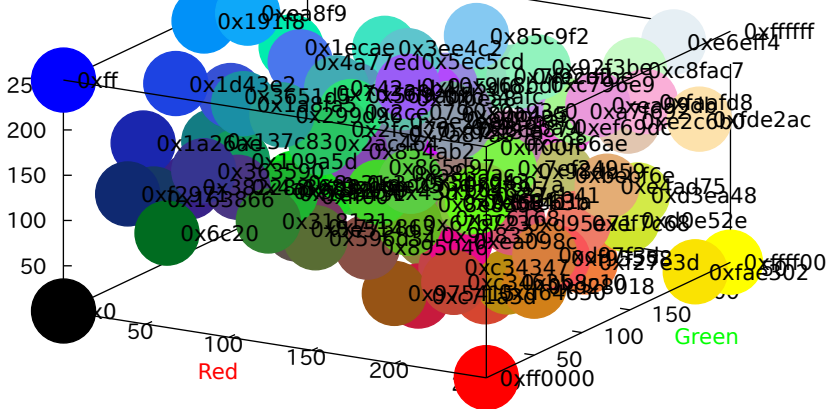
150

100

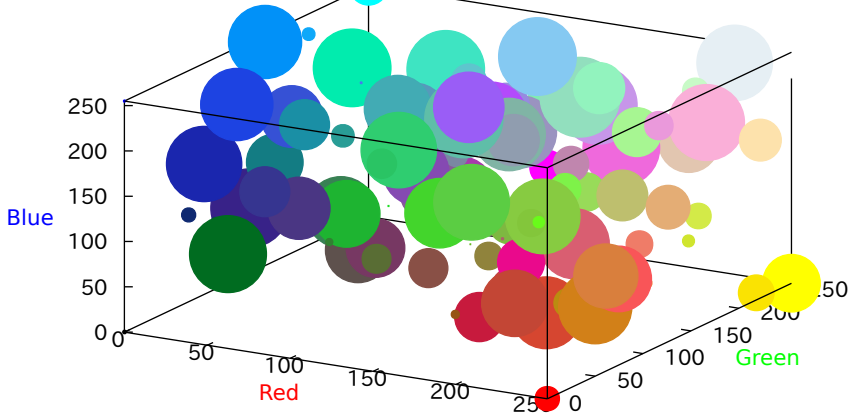
50

Red

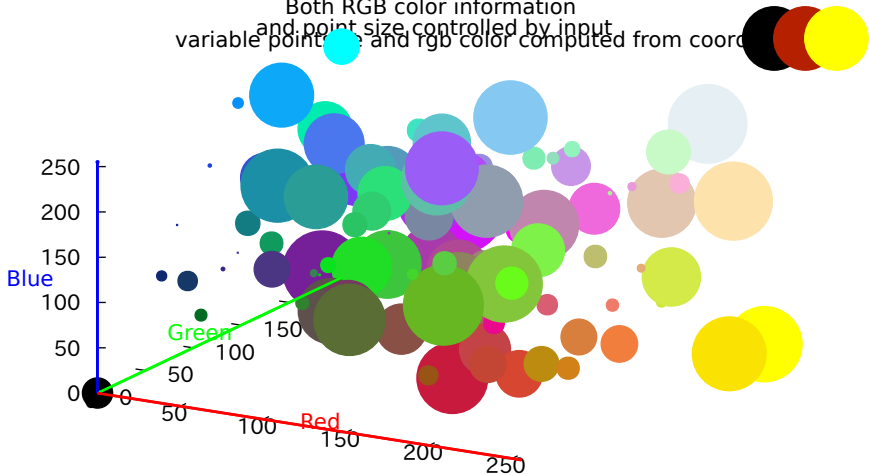
Green



Both RGB color information
and point size controlled by input
variable point size and rgb color read as hexadecimal

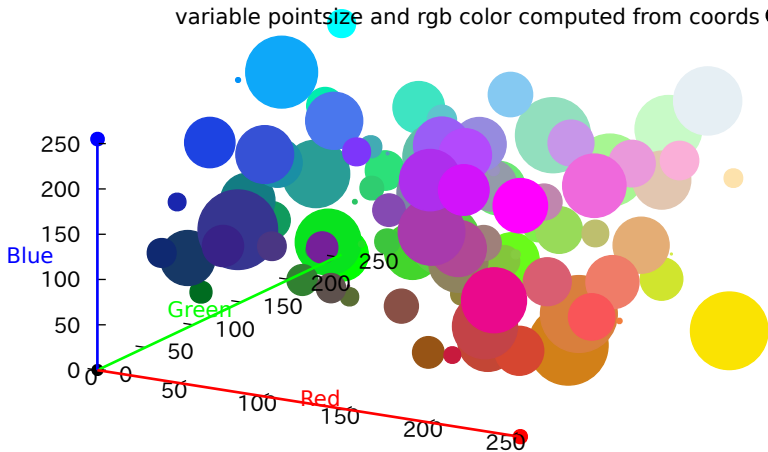


Both RGB color information
and point size controlled by input
variable point size and rgb color computed from coord

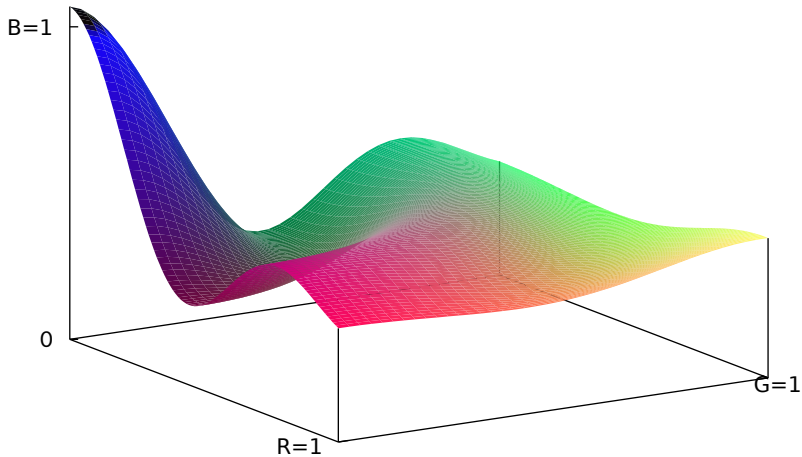


Demo of hidden3d with points only (no surface)

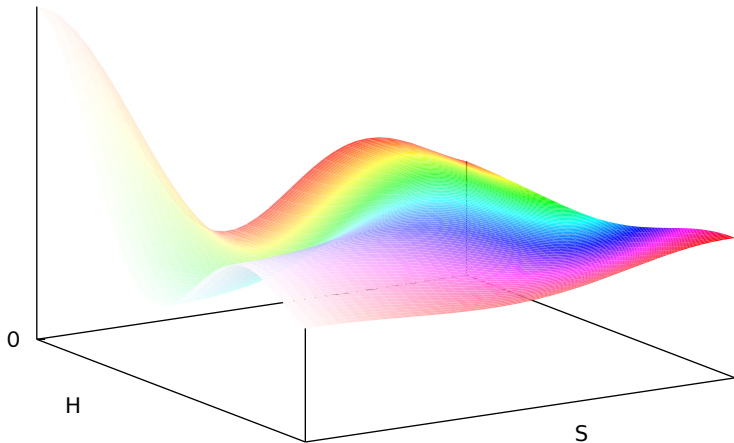
variable pointsize and rgb color computed from coords ● ● ●



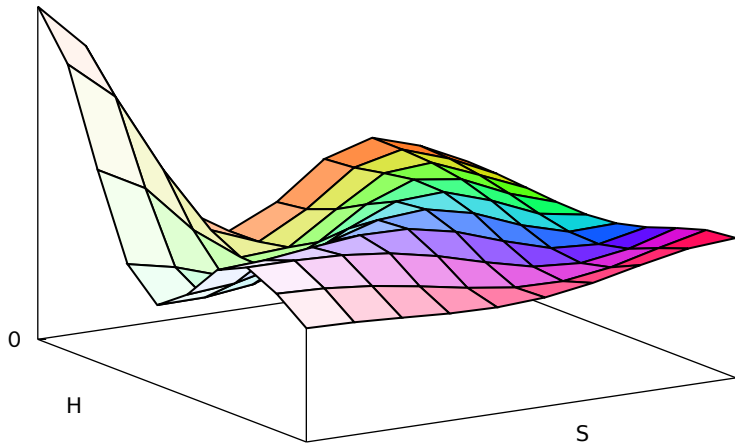
RGB coloring of pm3d surface

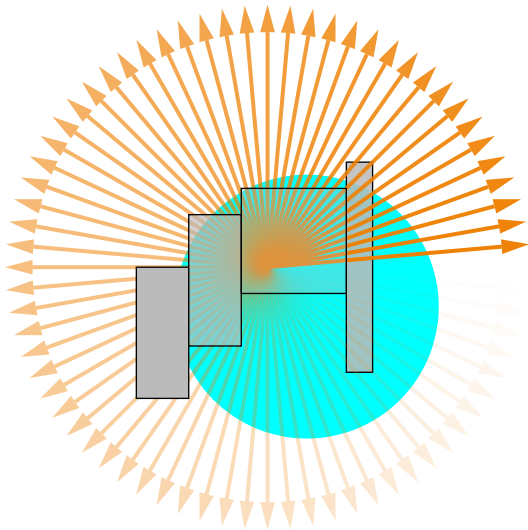


HSV coloring of pm3d surface
($V=1$)

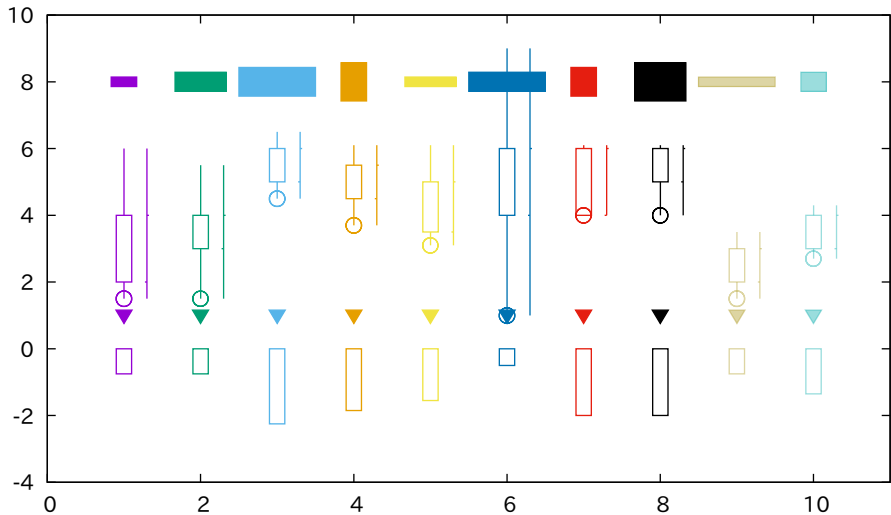


Explicit borders for pm3d tiling

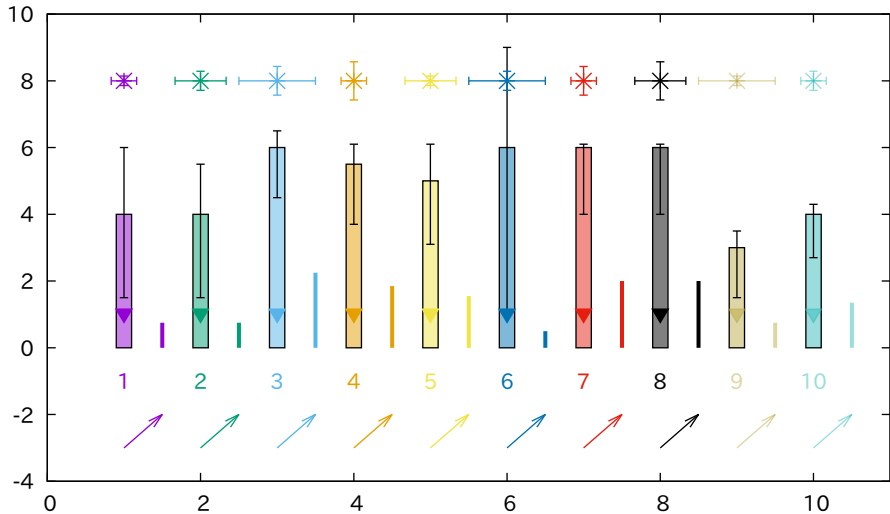




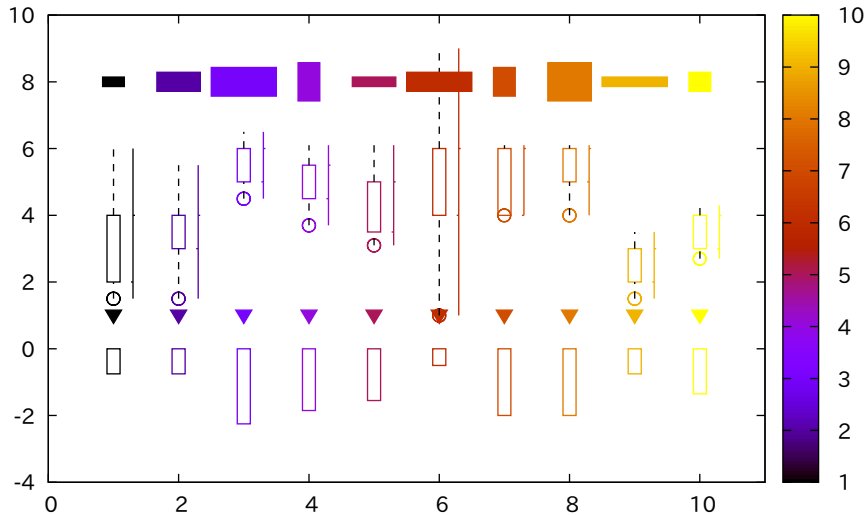
variable color points, circles, candlesticks, boxes, and boxxyerror



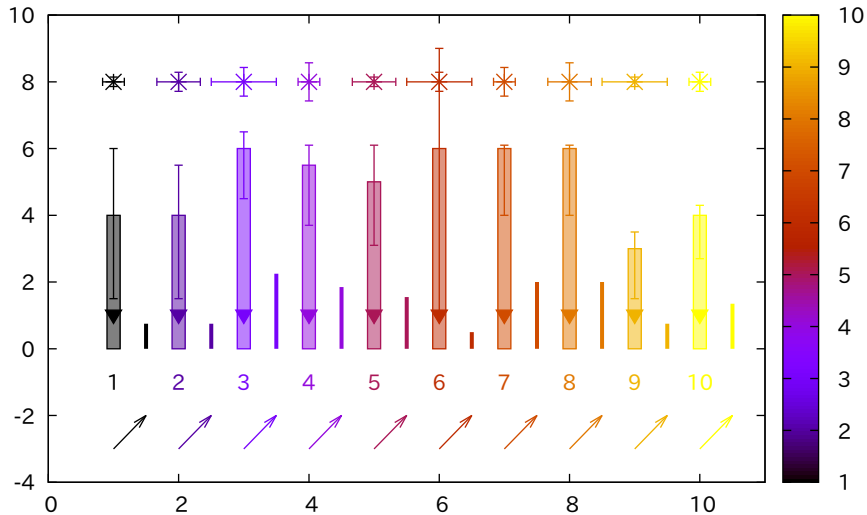
variable color boxerror, xyerrorbars, impulses, vectors, and labels



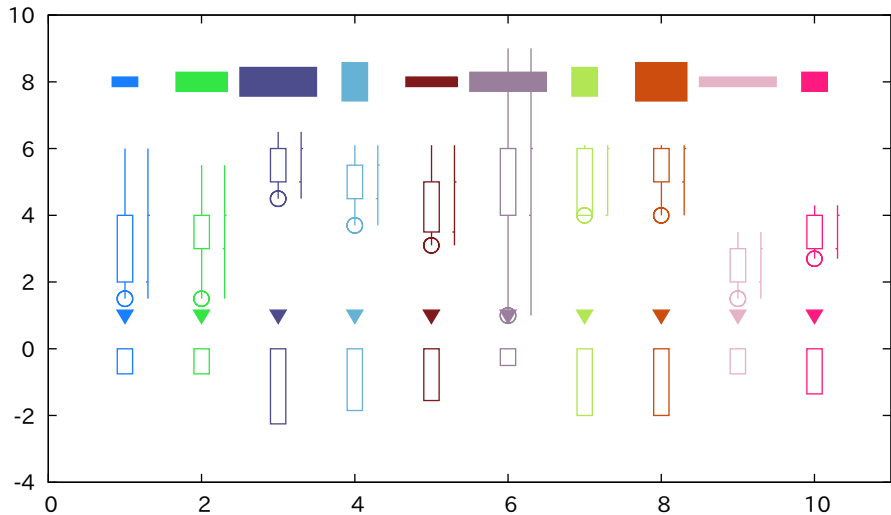
variable color using 'lc palette z'



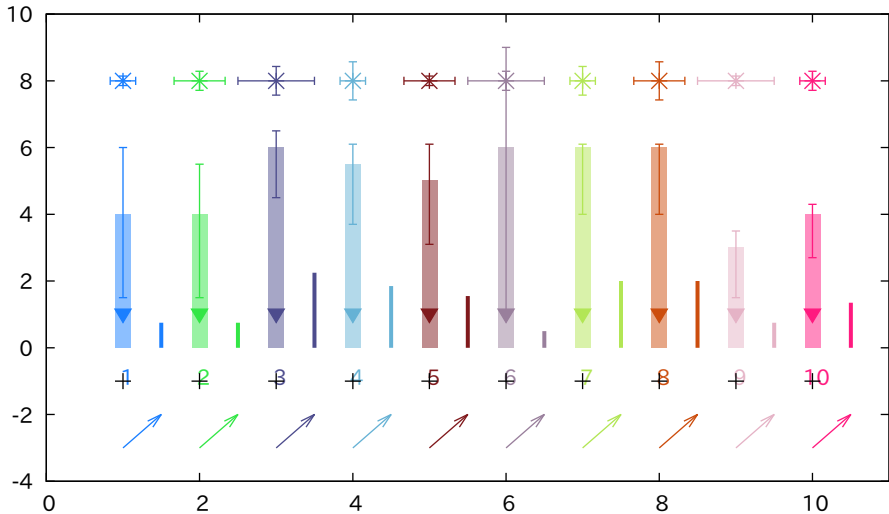
variable color using 'lc palette z'



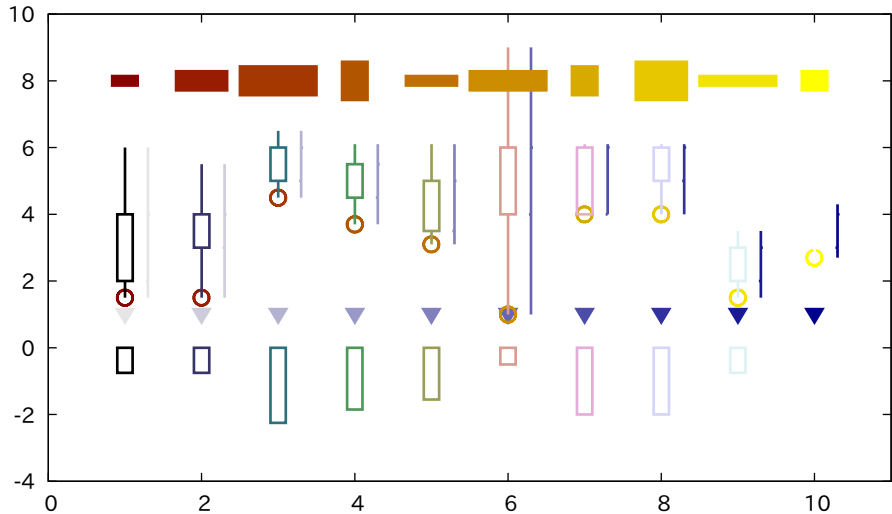
variable color using 'lc rgb variable'



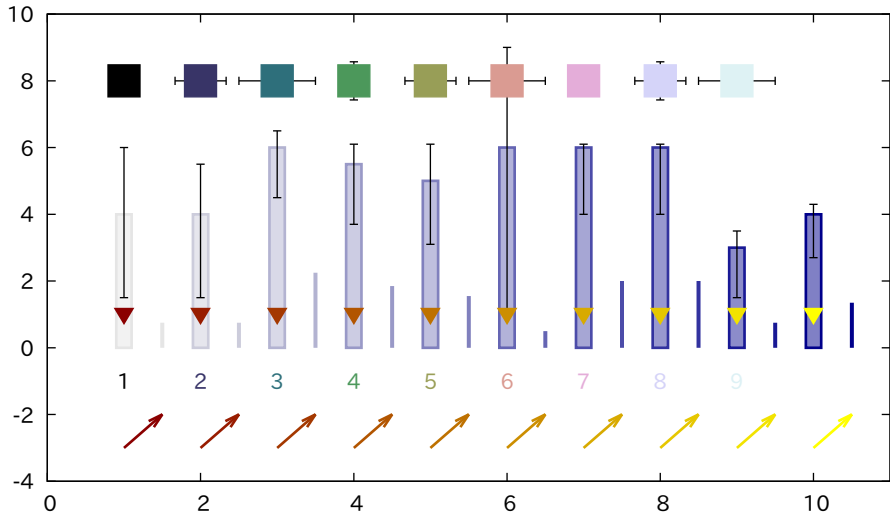
variable color using 'lc rgb variable'



variable color using multiple named palettes

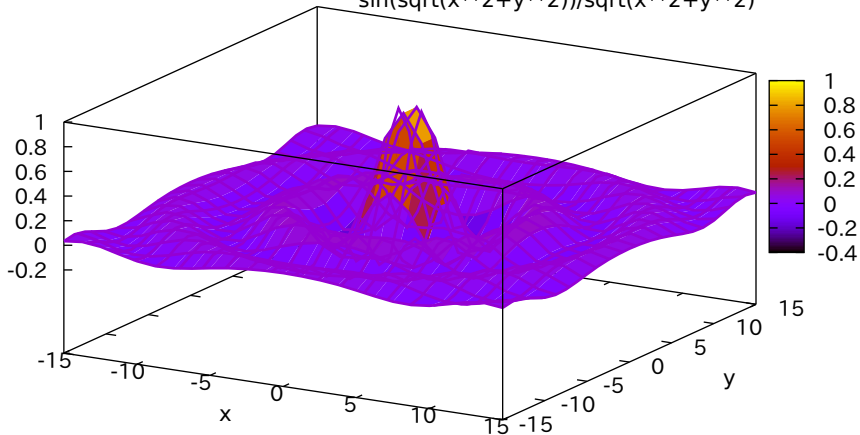


variable color using multiple named palettes



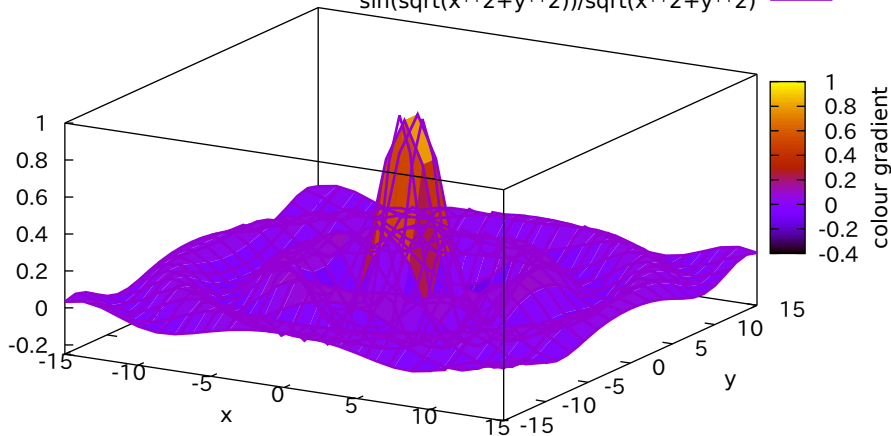
pm3d demo. Radial sinc function. Default options.

$$\frac{\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$$



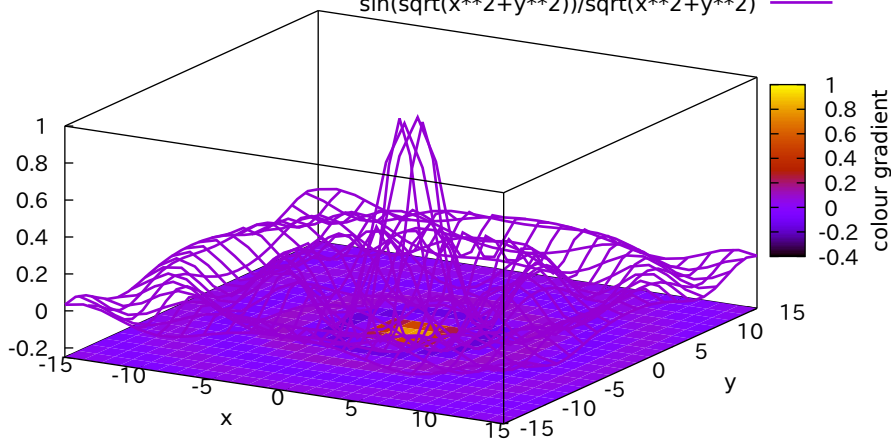
pm3d at s (surface) / ticslevel 0

$\sin(\sqrt{x^2+y^2})/\sqrt{x^2+y^2}$



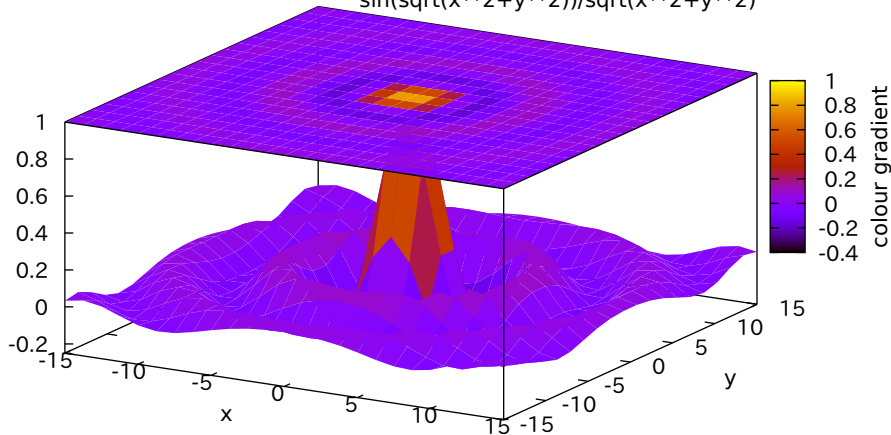
pm3d at b (bottom)

$\sin(\sqrt{x^2+y^2})/\sqrt{x^2+y^2}$



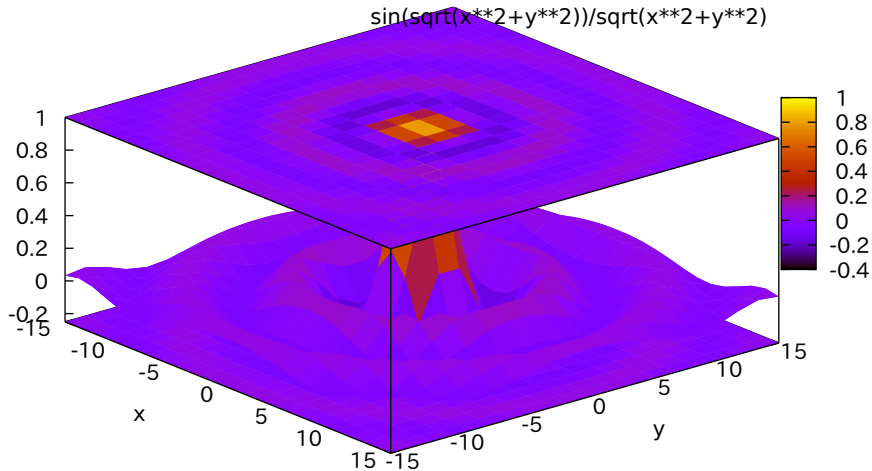
unset surface; set pm3d at st (surface and top)

$$\sin(\sqrt{x^2+y^2})/\sqrt{x^2+y^2}$$

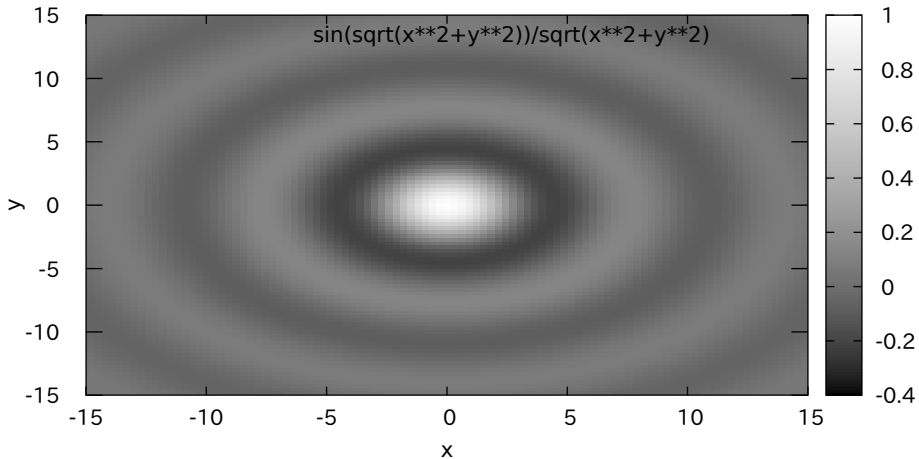


set pm3d at bstbst (funny combination, only for screen or postscript)

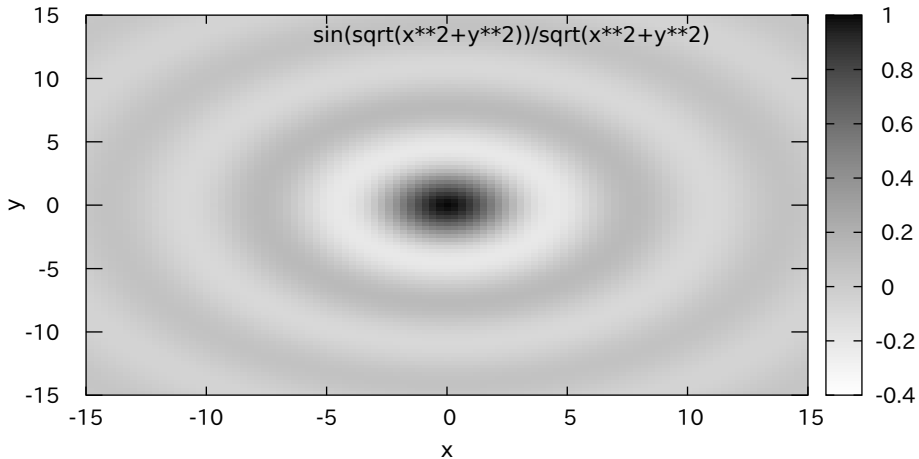
$$\sin(\sqrt{x^2+y^2})/\sqrt{x^2+y^2}$$



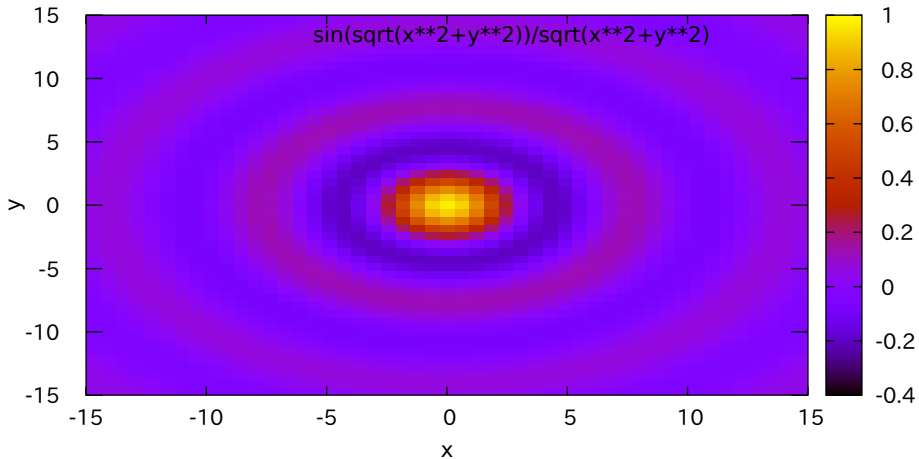
gray map



gray map, negative

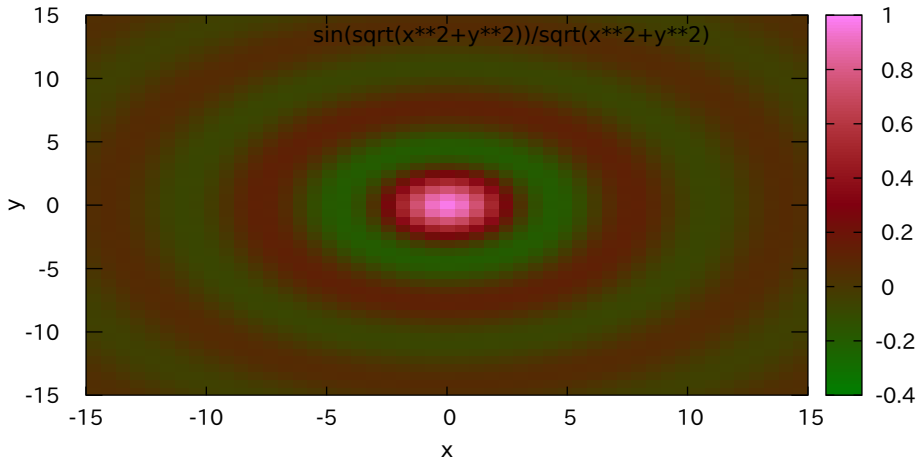


colour map, using default rgbformulae 7,5,15 ... traditional pm3d (black-blue-red-yellow)

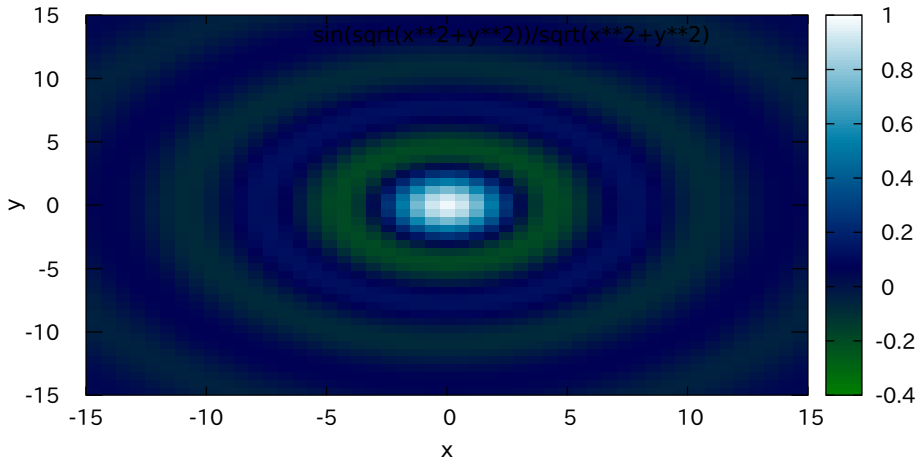


colour, rgbformulae 3,11,6 ... green-red-violet

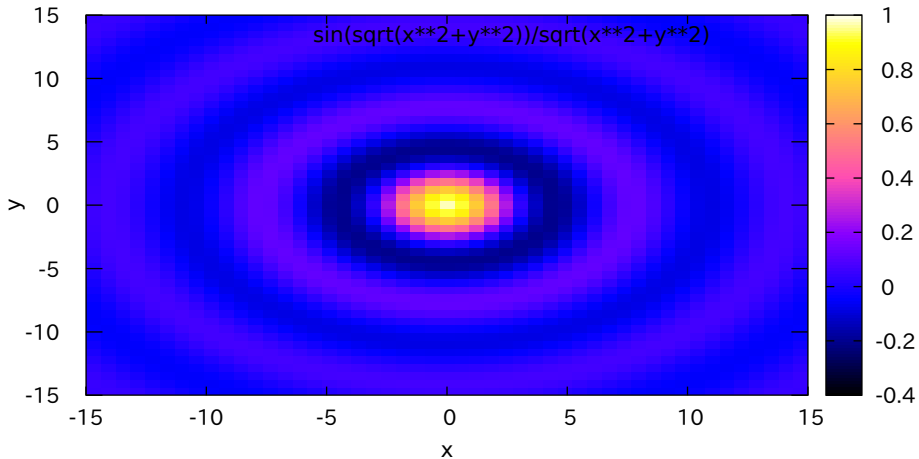
$$\sin(\sqrt{x^2+y^2})/\sqrt{x^2+y^2}$$



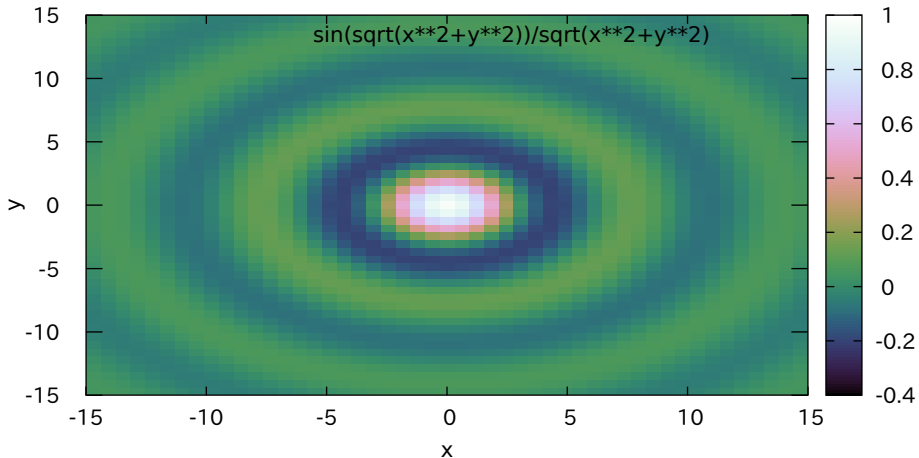
colour, rgbformulae 23,28,3 ... ocean (green-blue-white); OK are also all other permutation



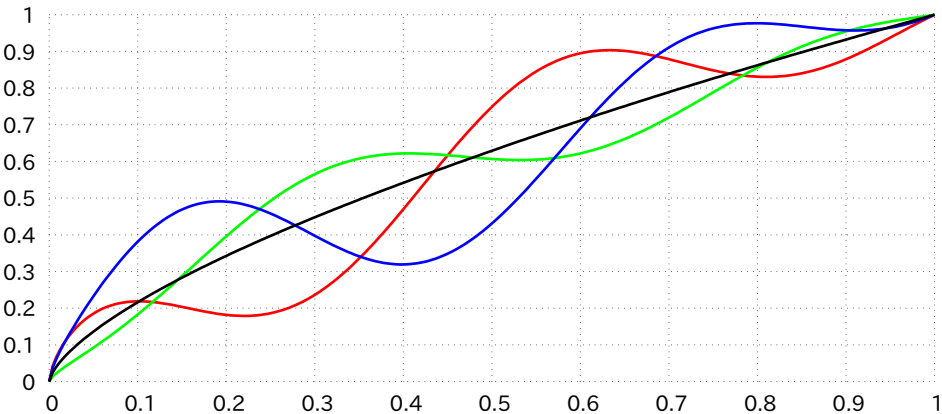
colour, rgbformulae 30,31,32 ... color printable on gray (black-blue-violet-yellow-white)



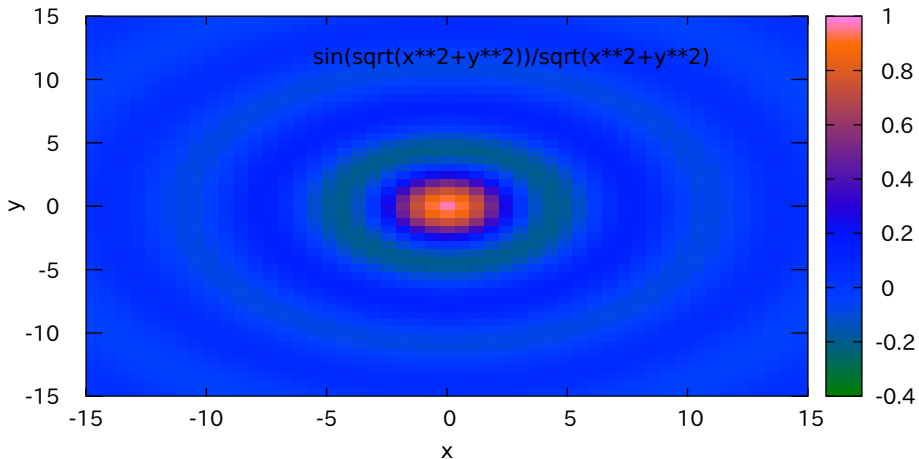
cubehelix color scheme with monotonic intensity
D A Green (2011) <http://arxiv.org/abs/1108.5083>



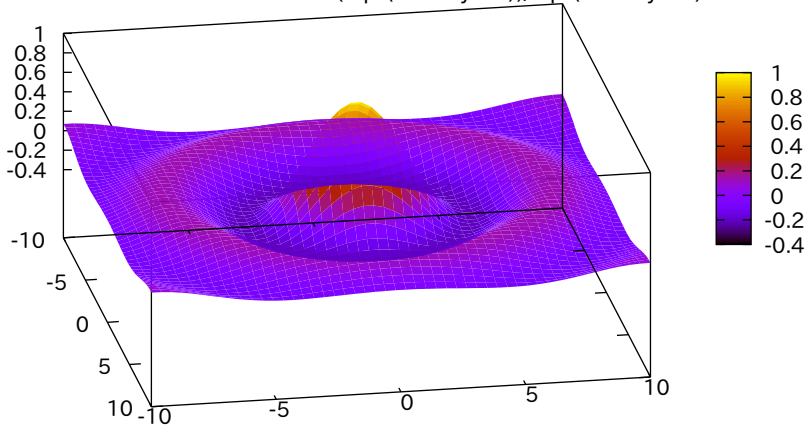
R,G,B profiles of the current color palette
red — green — blue — NTSC —



rgbformulae 31,-11,32: negative formula number=inverted color

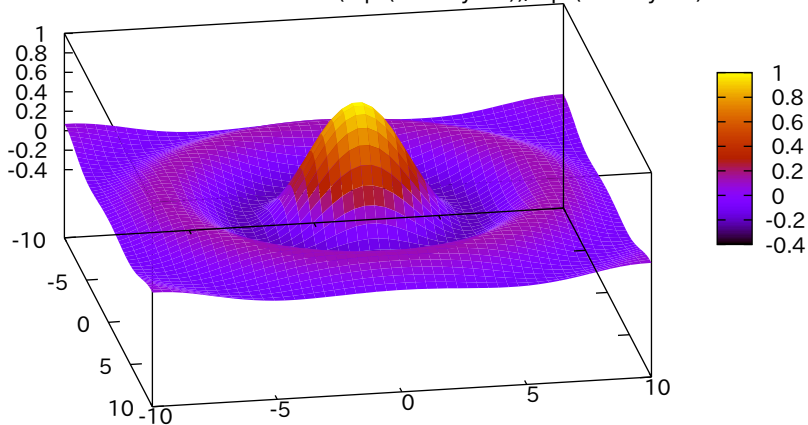


set pm3d scansforward: wrong, because back overwrites front
 $\sin(\sqrt{x^2+y^2})/\sqrt{x^2+y^2}$



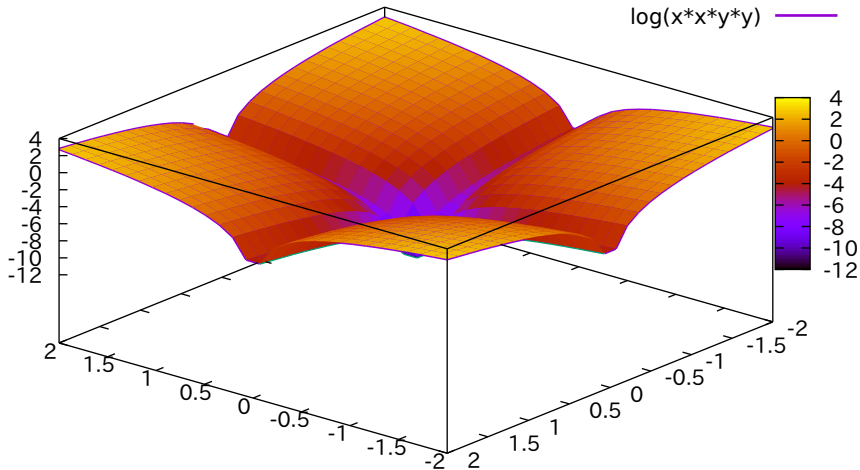
set pm3d scansbackward: correctly looking surface

$$\sin(\sqrt{x^2+y^2})/\sqrt{x^2+y^2}$$



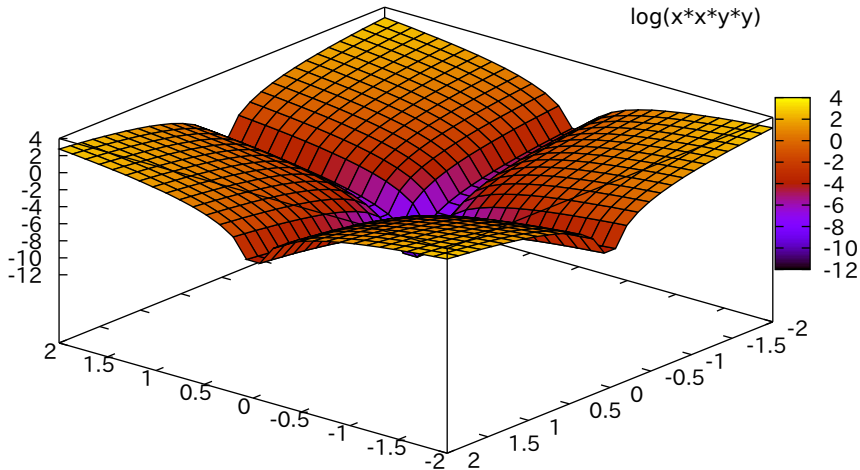
set hidden3d

$\log(x*x*y*y)$



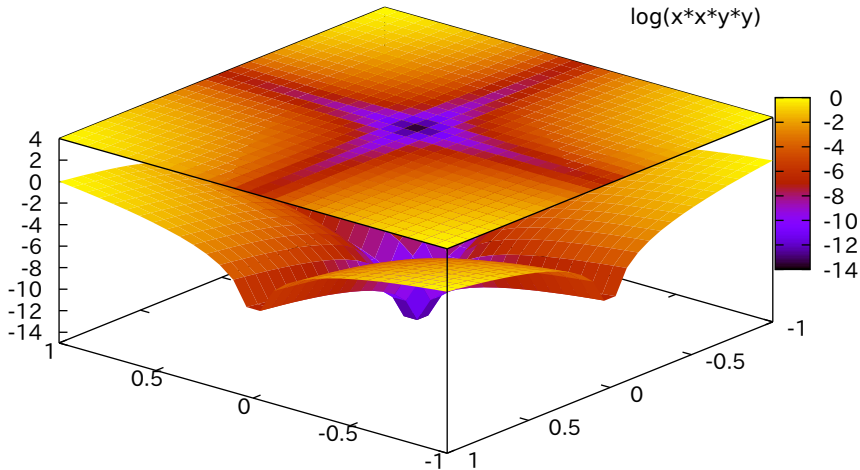
set pm3d depthorder border lc 'black' lw 1

$\log(x*x*y*y)$



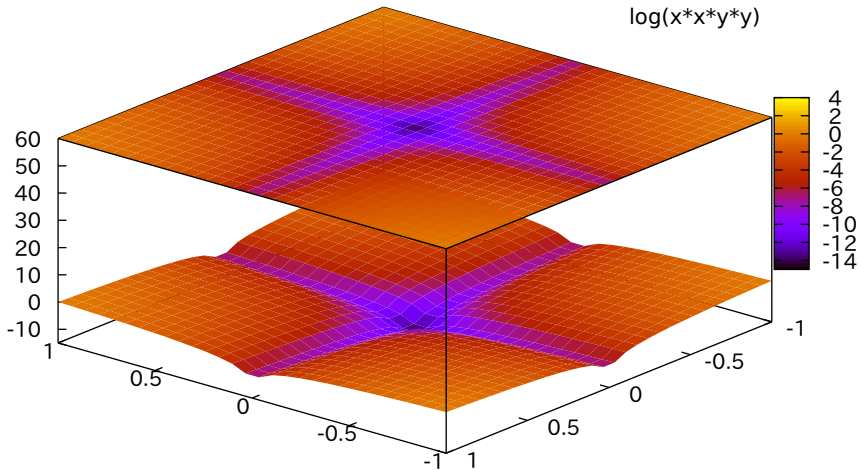
bad: surface and top are too close together

$\log(x*x*y*y)$

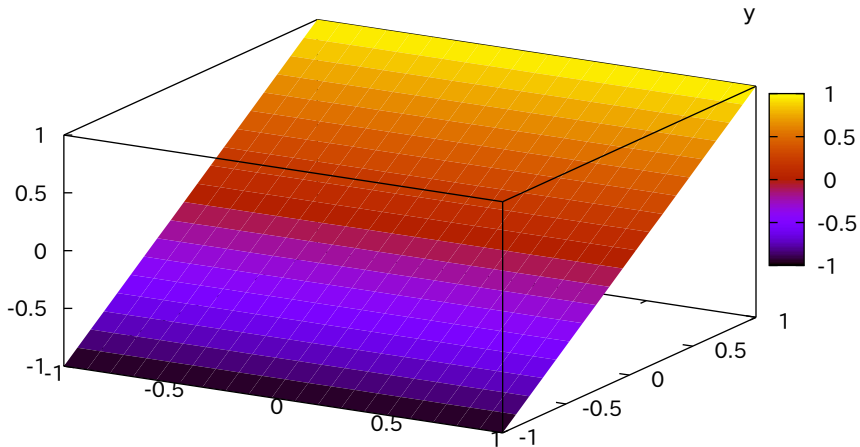


solution: use independent 'set zrange' and 'set cbrange'

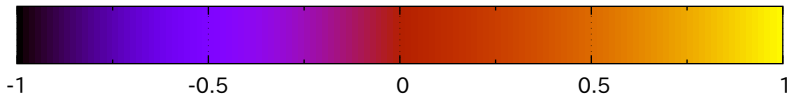
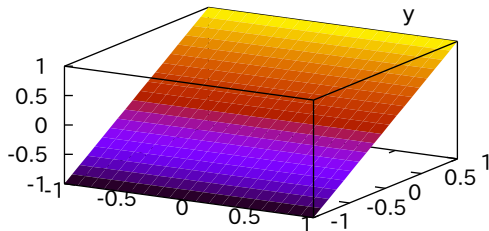
$\log(x*x*y*y)$



color box is on by default at a certain position

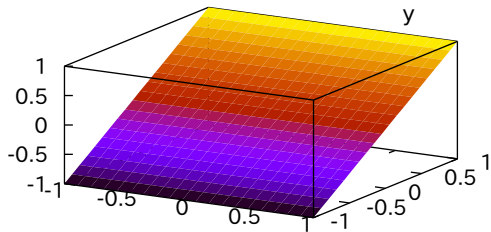


color box is on again, now with horizontal gradient

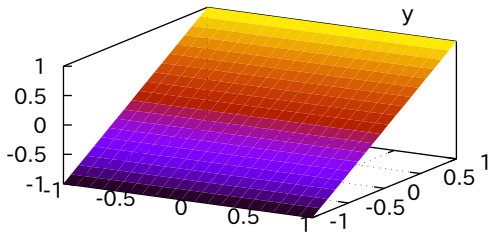


see cblabel, grid cb, mcbticks, ...

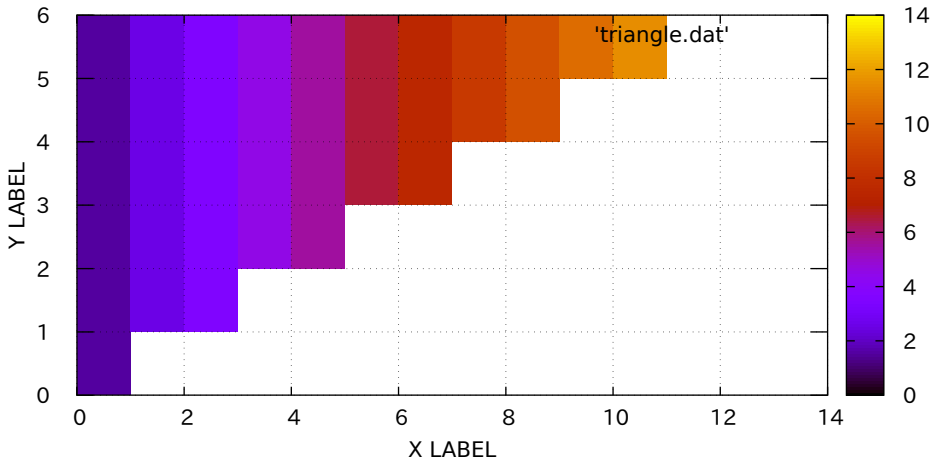
color box is switched off



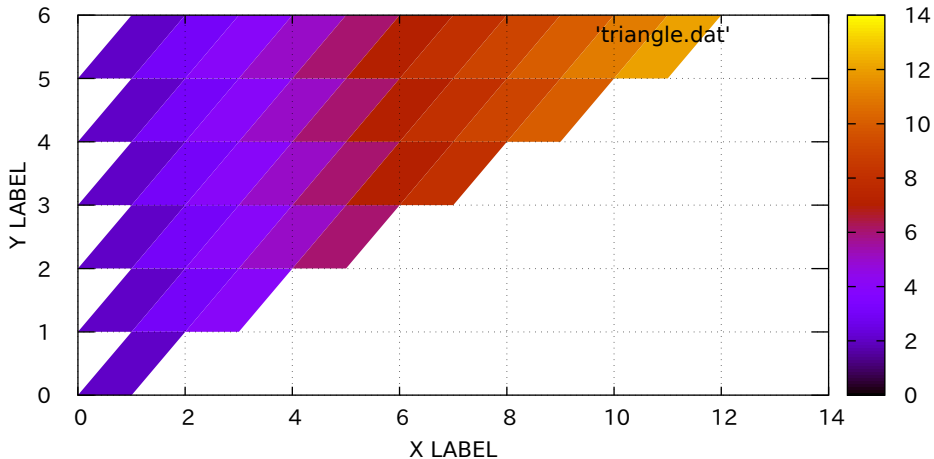
using now "set grid back; unset colorbox"



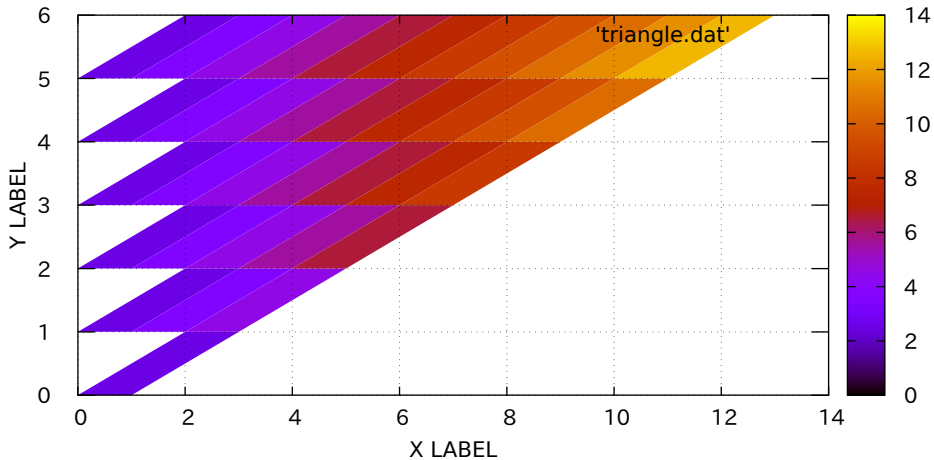
Datafile with different nb of points in scans; pm3d flush begin



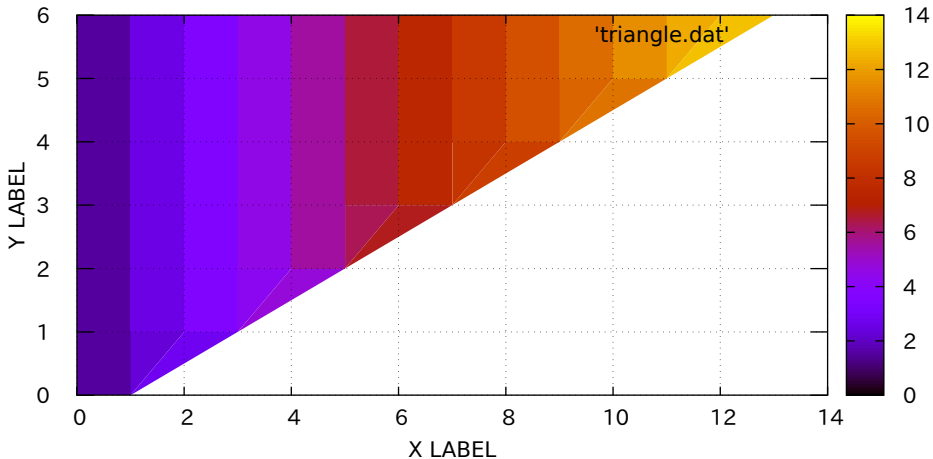
Datafile with different nb of points in scans; pm3d flush center



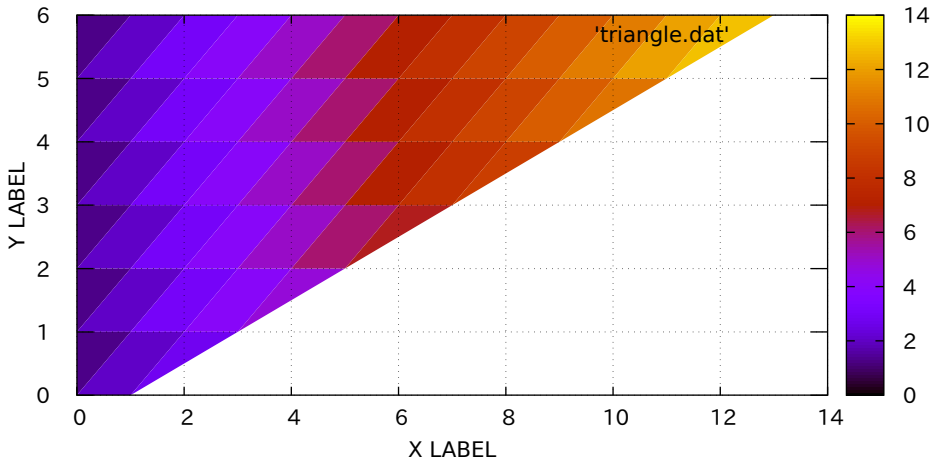
Datafile with different nb of points in scans; pm3d flush end



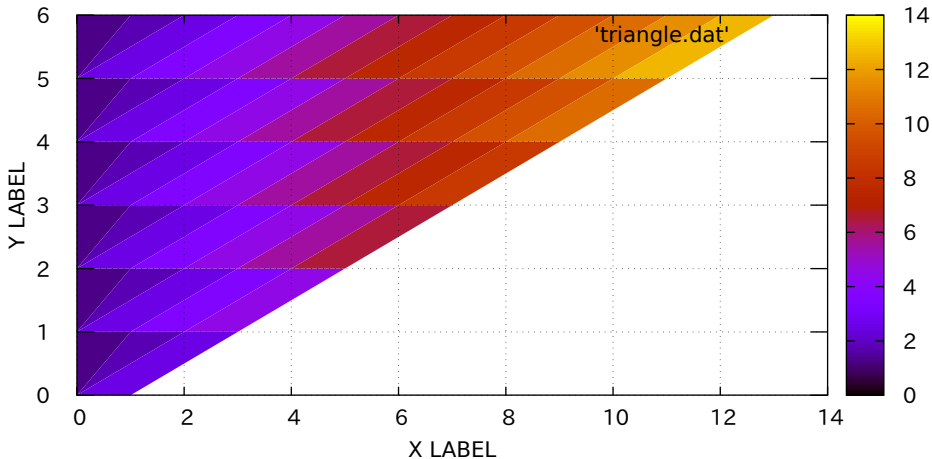
Data with different nb of points in scans; pm3d ftriangles flush begin



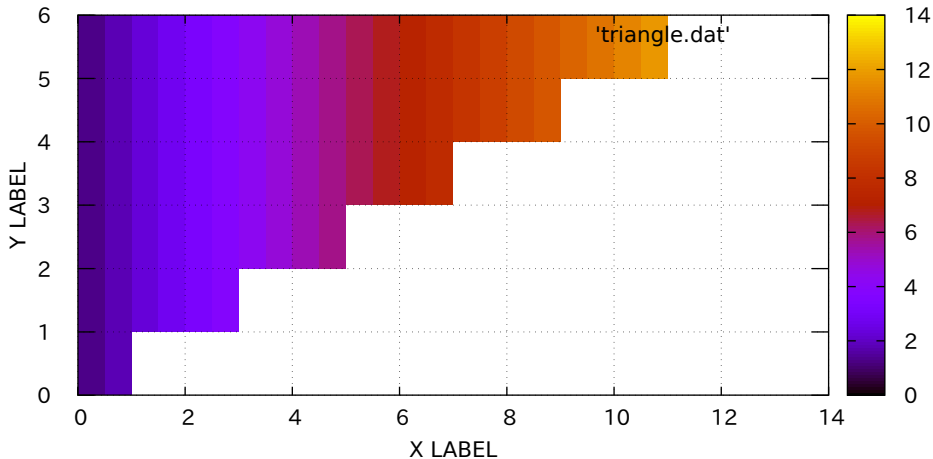
Data with different nb of points in scans; pm3d ftriangles flush center



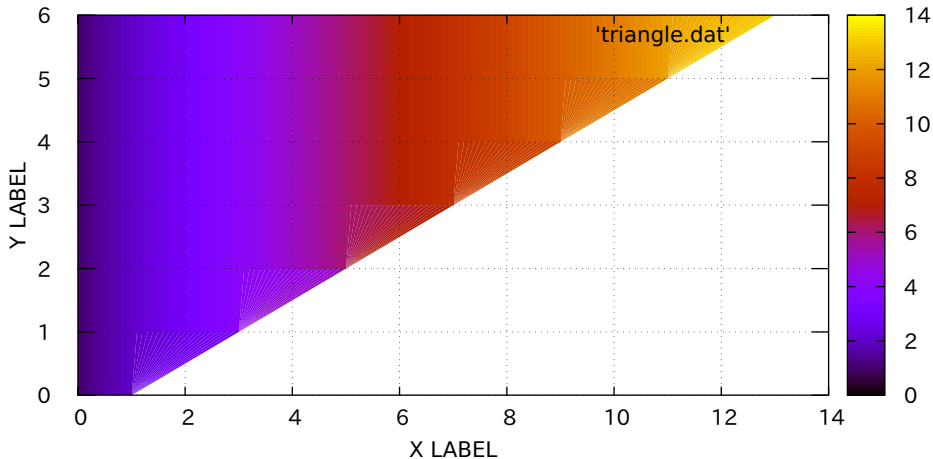
Data with different nb of points in scans; pm3d ftriangles flush end



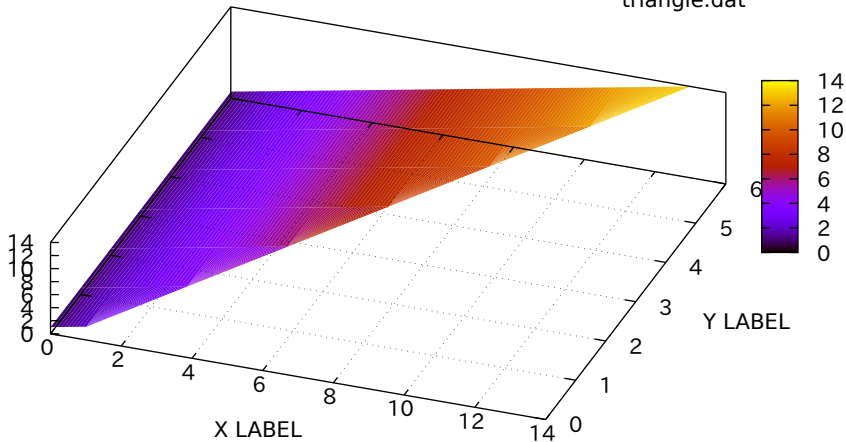
Using interpolation with datafile; pm3d interpolate 2,1



Using interpolation with datafile; pm3d ftriangles interpolate 10,1

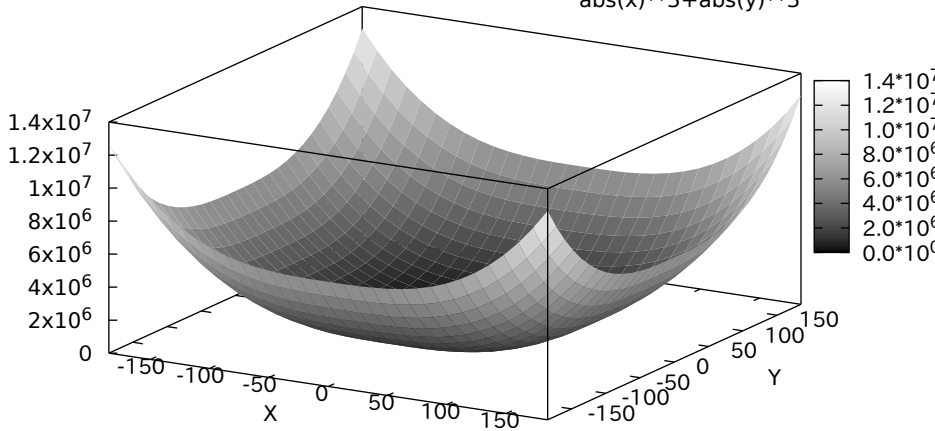


Using interpolation with datafile; pm3d at s ftriangles interpolate 10,1
'triangle.dat'

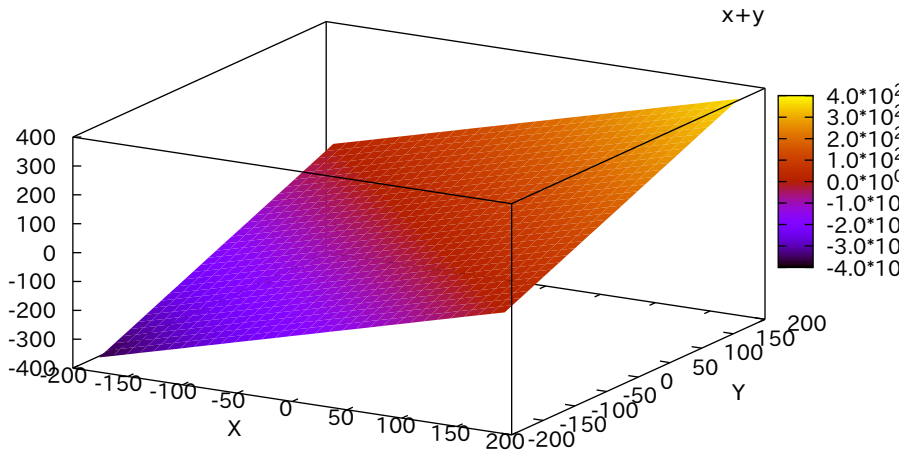


only for enhanced terminals: 'set format cb ...'

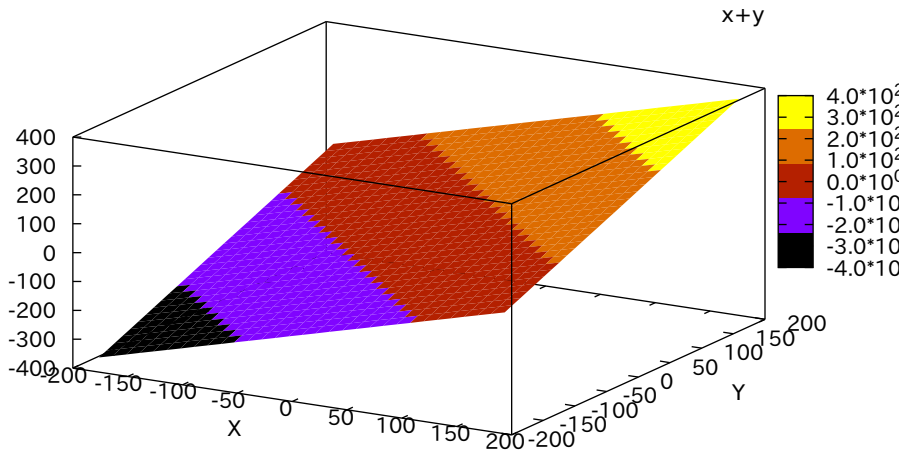
$$\text{abs}(x)^3 + \text{abs}(y)^3$$



function 'x+y' using all colors available, 'set pal maxcolors 0'

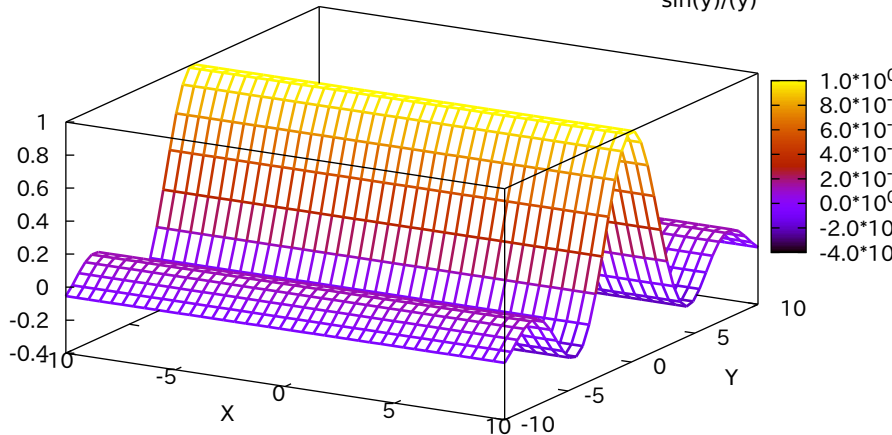


function 'x+y' using only 5 colors, 'set pal maxcolors 5'



color lines: 'splot sin(y)/(y) with lines palette'

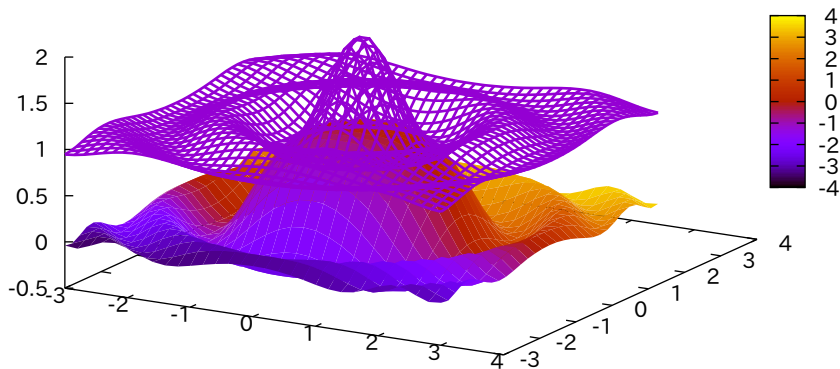
$\sin(y)/y$ —



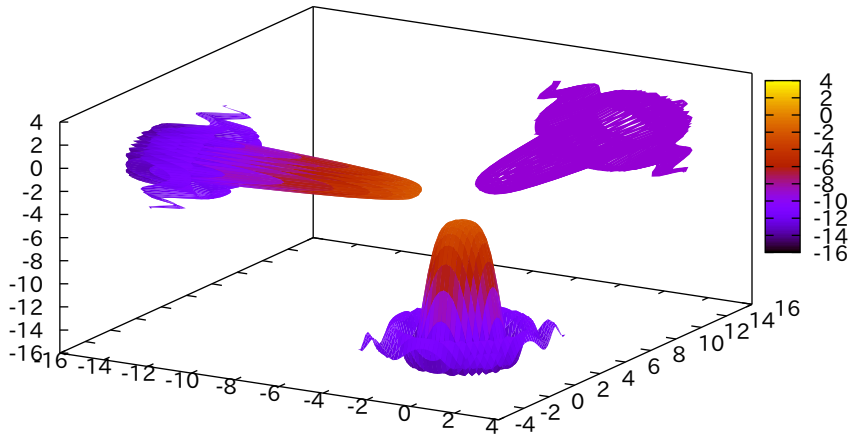
pm3d explicit mode --- coloring according to the 4th parameter of 'using'

'binary2' binary u 1:2:3:(\$2+(\$1+\$2)/10)

1+sinc(x*4, y*4) ———



coloring according to the 3rd 'using' parameter (left) and to the z-value (bottom)



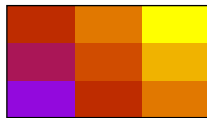
c3



set pm3d corners2color mode mean



c4



harmean



Original grid points



geomean



c1



median

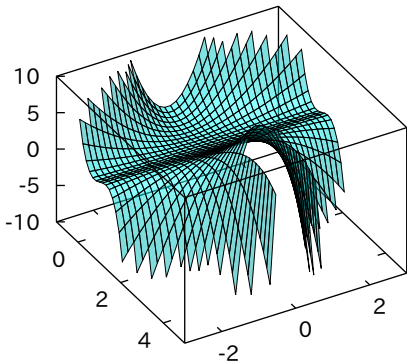


c2

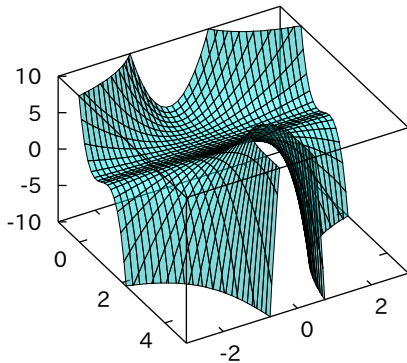


new pm3d option for smooth clipping against zrange

set pm3d clip4in (old default)

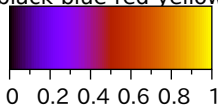


set pm3d clip z (new default)

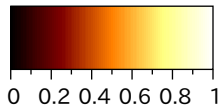
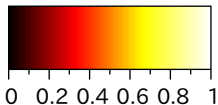


Palettes according to 'help palette rgbformulae'

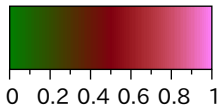
traditional pm3d
(black-blue-red-yellow)



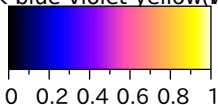
hot (black-red-yellow-white) AFM hot (black-red-yellow-white)



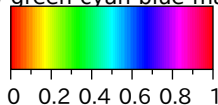
green-red-violet



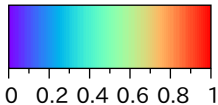
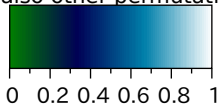
color printable on gray
(black-blue-violet-yellow-white)



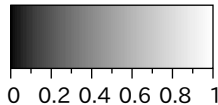
HSV model



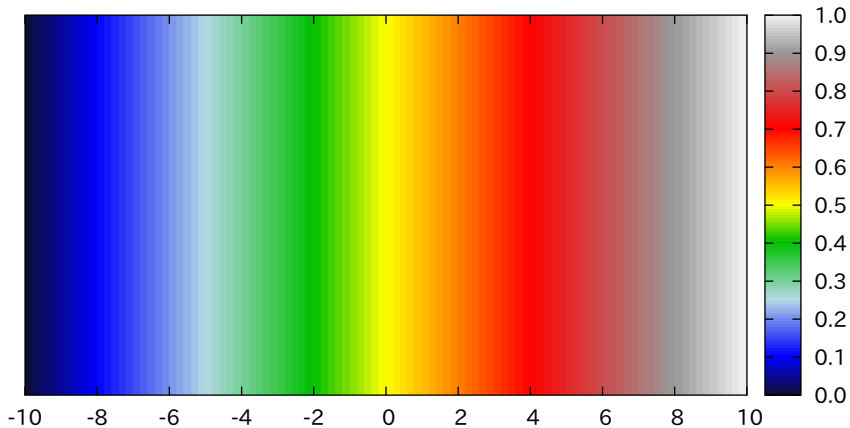
ocean (green-blue-white) rainbow (blue-green-yellow-red)
try also other permutations



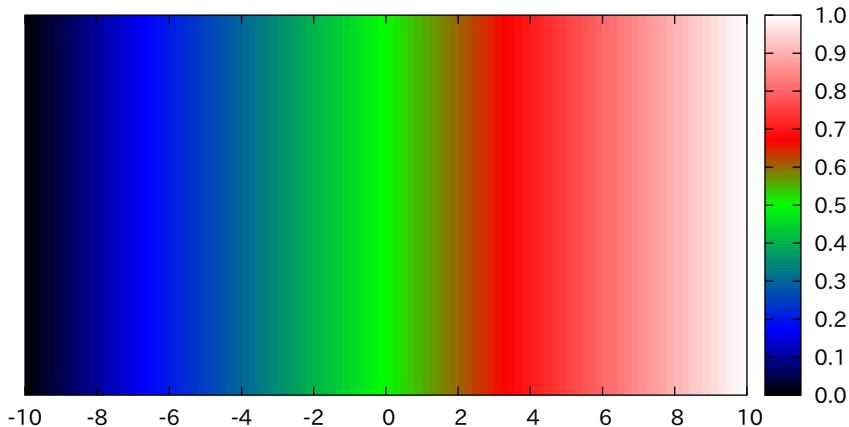
gray palette



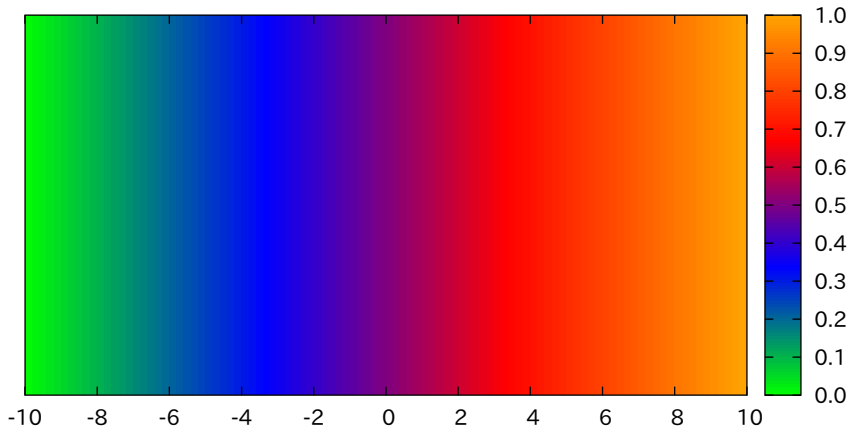
set palette defined



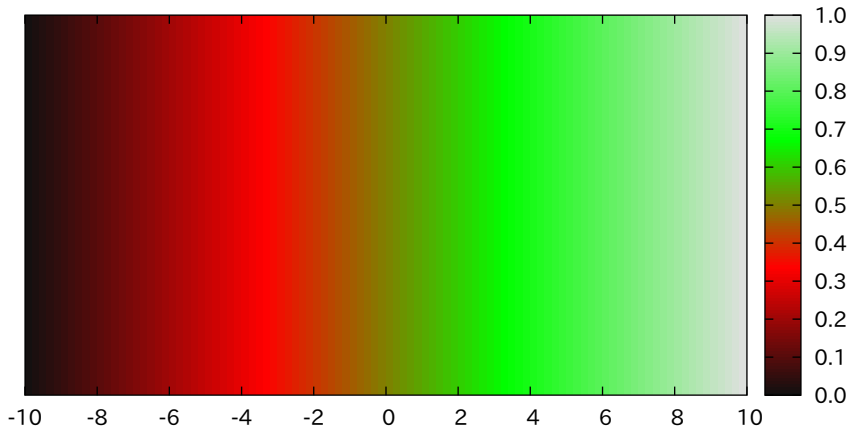
set palette defined (0 0 0 0, 1 0 0 1, 3 0 1 0, 4 1 0 0, 6 1 1 1)



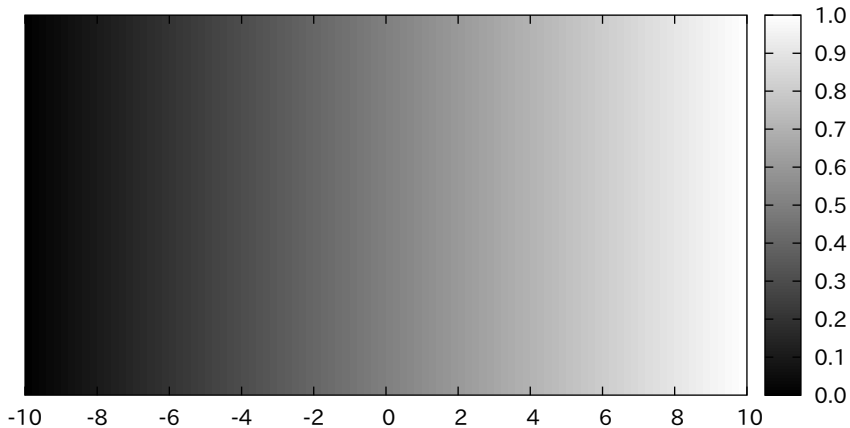
set palette defined (0 "green", 1 "blue", 2 "red", 3 "orange")



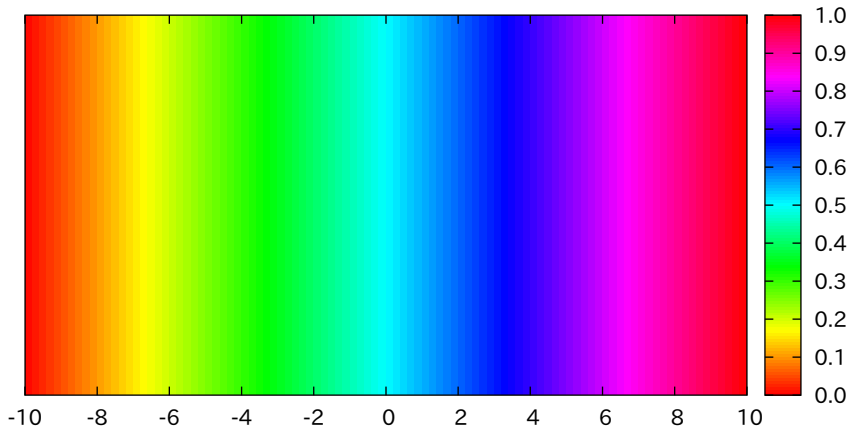
set palette defined (20 "#101010", 30 "#ff0000", 40 "#00ff00", 50 "#e0e0e0")

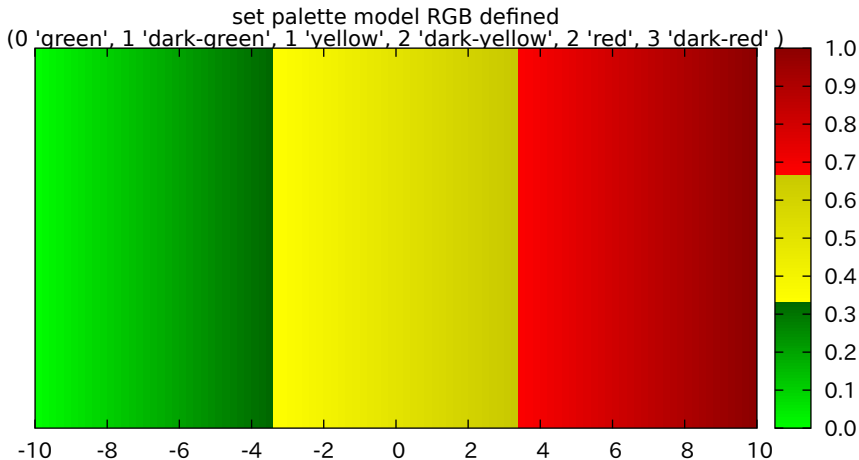


set palette defined (0 0 0 0, 1 1 1 1)

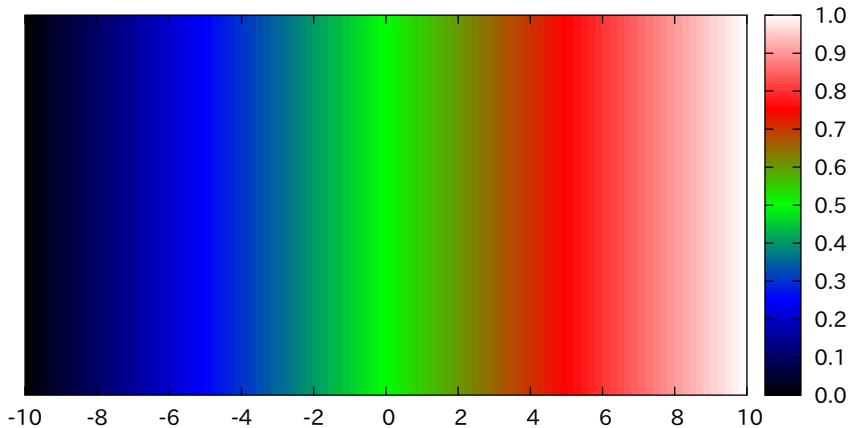


set palette model HSV defined (0 0 1 1, 1 1 1 1)

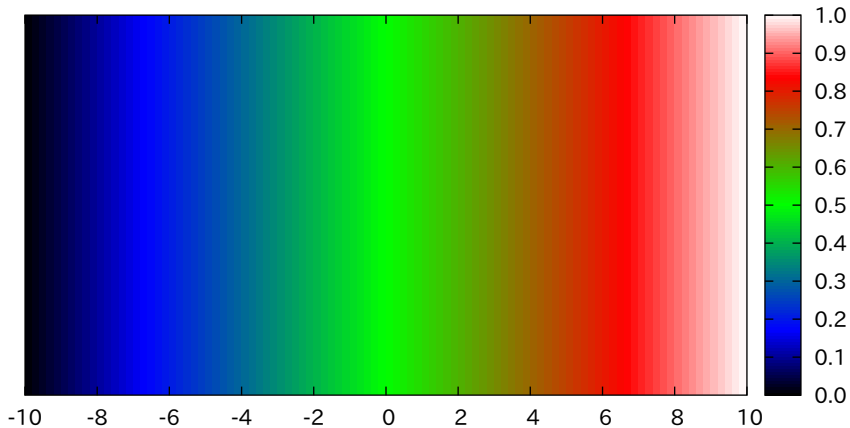




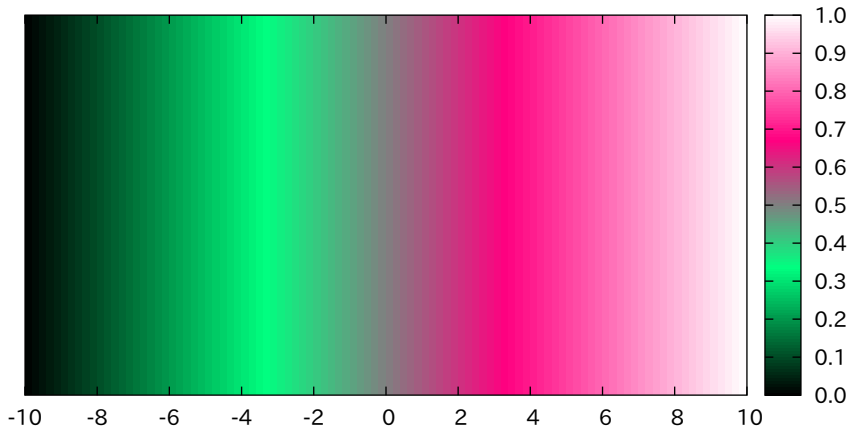
set palette file "-" (file with 3 columns)



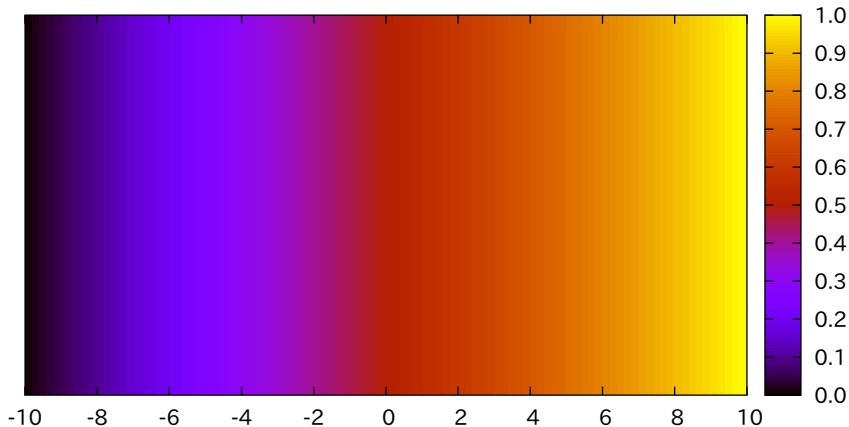
set palette file "-" (file with 4 columns)



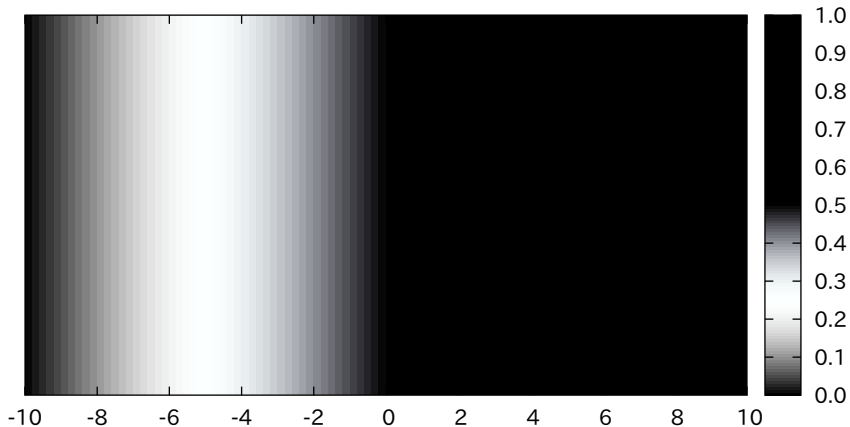
set palette file "-" using 1:2:(\$1+\$2)/2



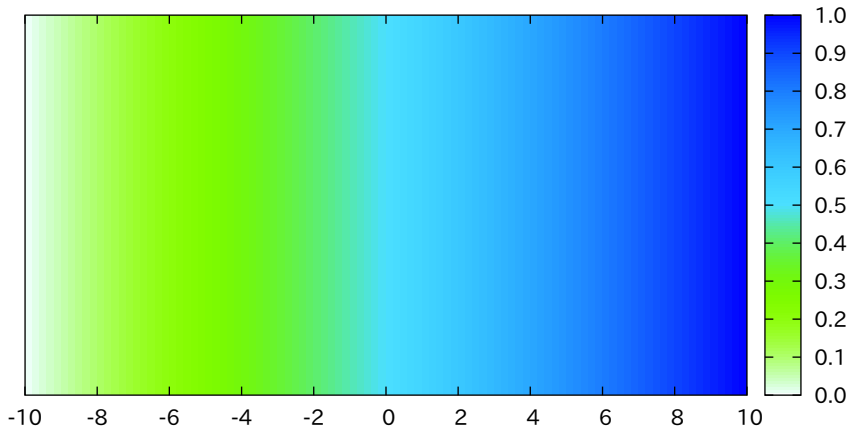
set palette model RGB rgbformulae 7,5,15



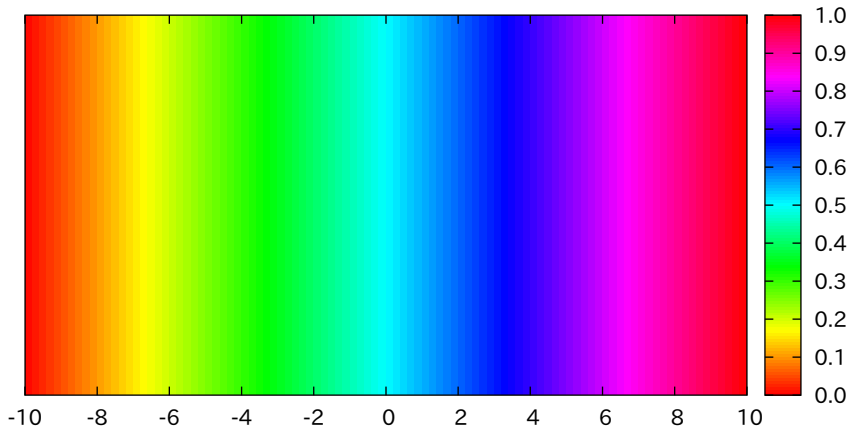
set palette model HSV rgbformulae 7,5,15



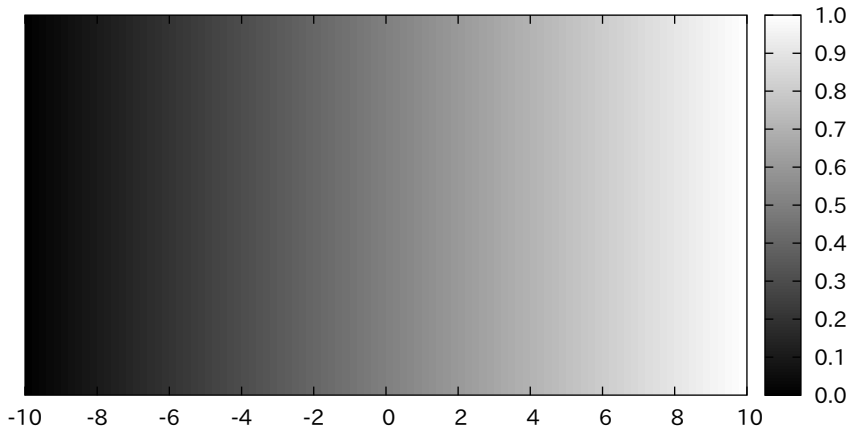
set palette model CMY rgbformulae 7,5,15



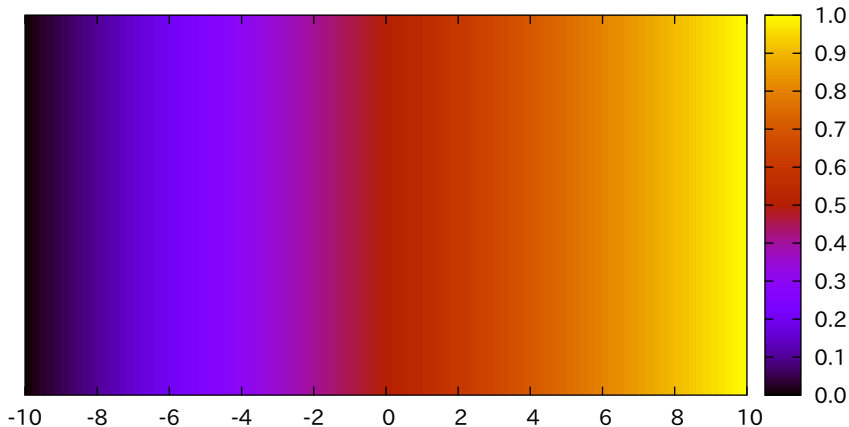
set palette model HSV rrgbformulae 3,2,2



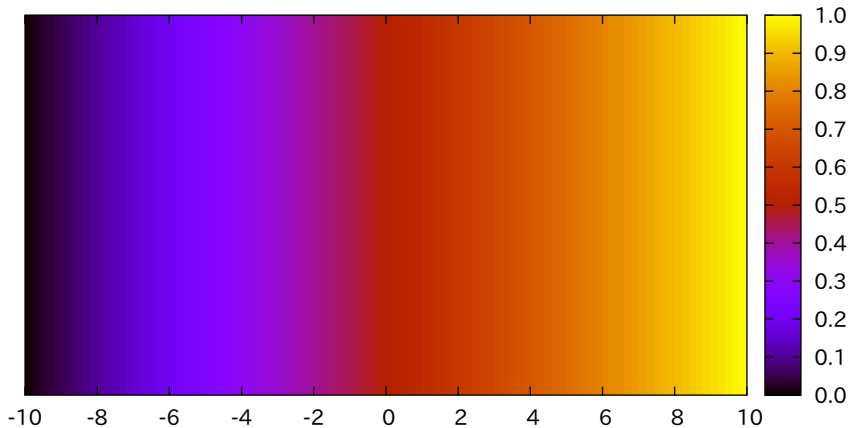
set palette functions gray, gray, gray

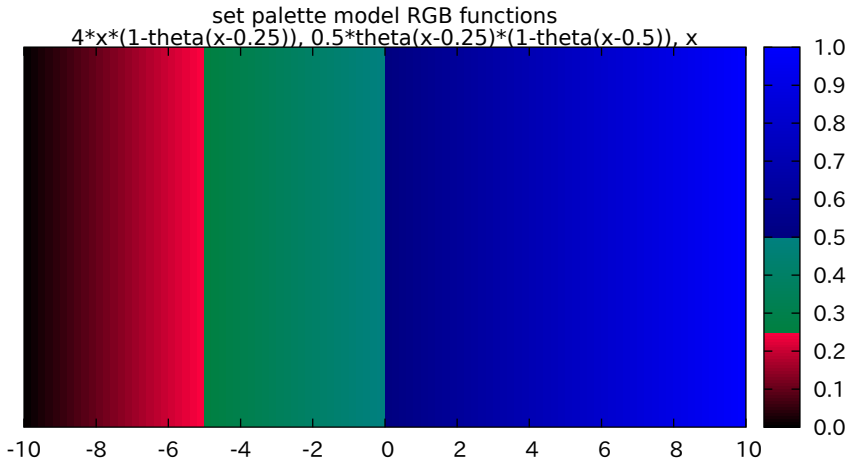


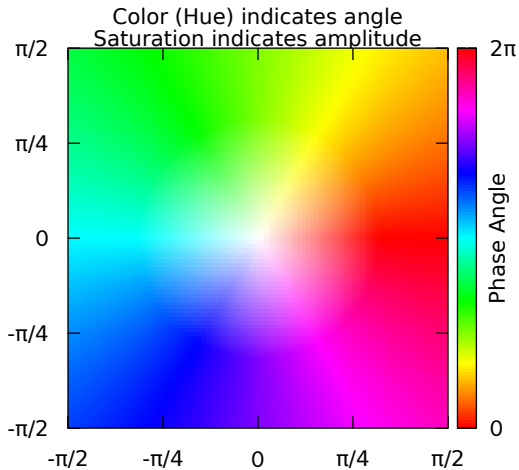
set palette functions sqrt(gray), gray**3, sin(gray*2*pi) <--> 7,5,15



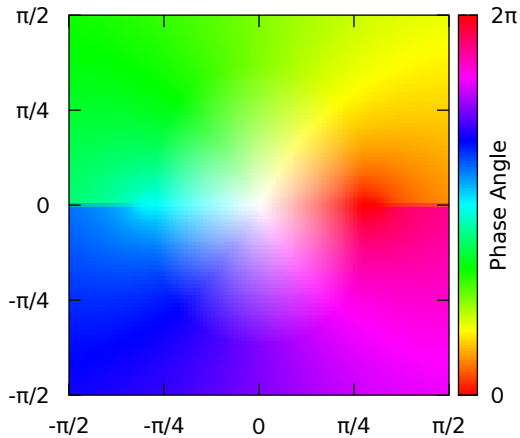
set palette rgbformulae 7,5,15



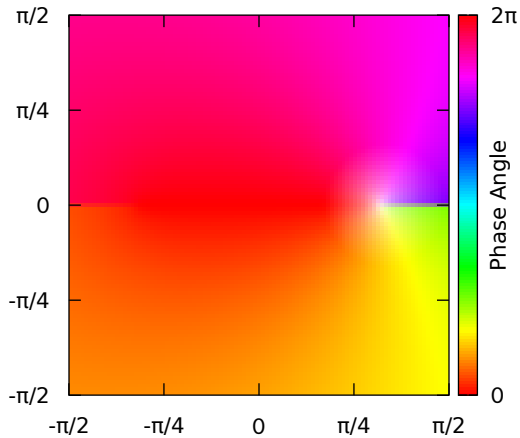




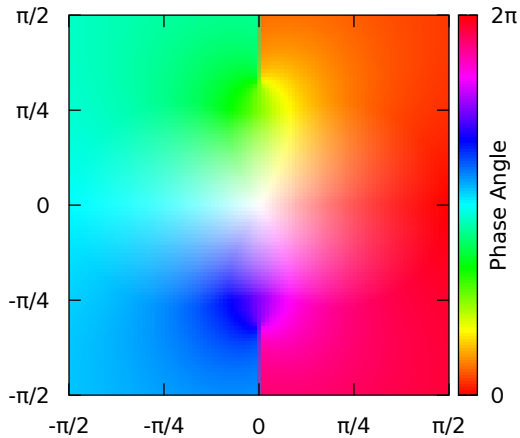
$\text{asin}(x + iy)$



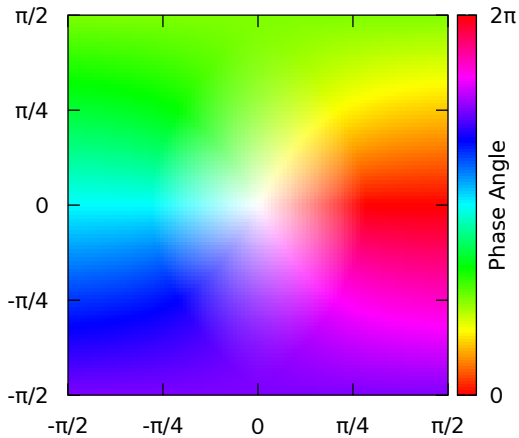
$\text{acos}(x + iy)$



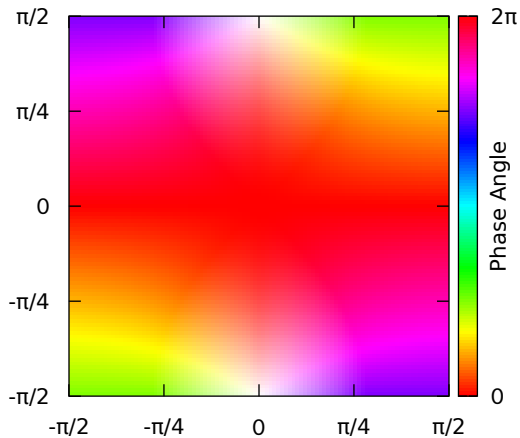
$\text{atan}(x + iy)$



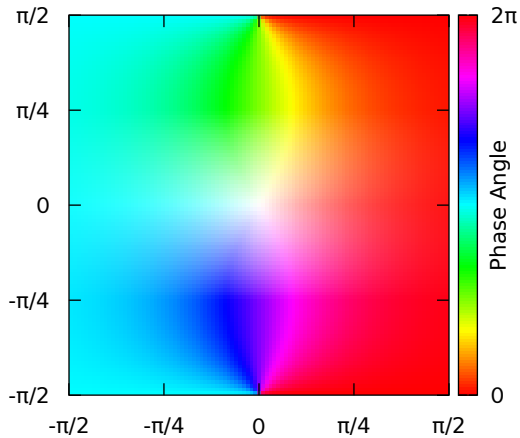
$\sinh(x + iy)$

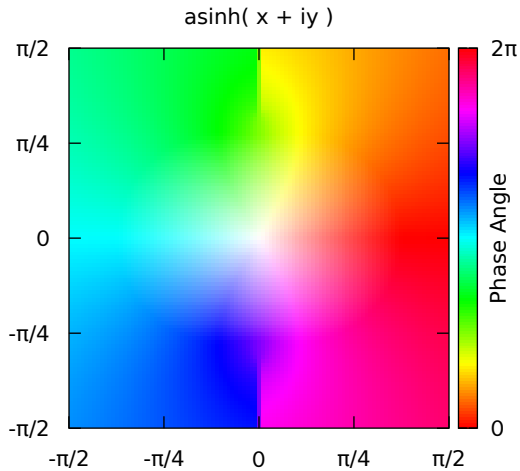


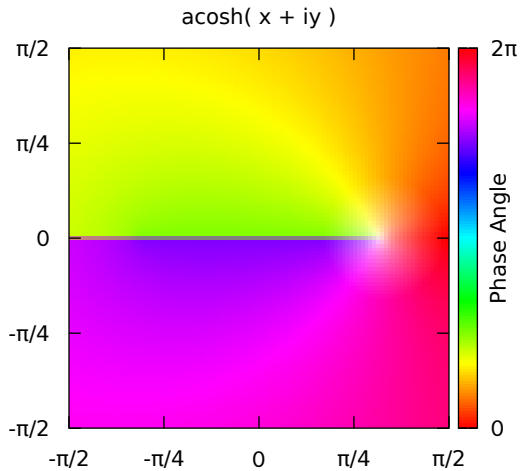
$\cosh(x + iy)$

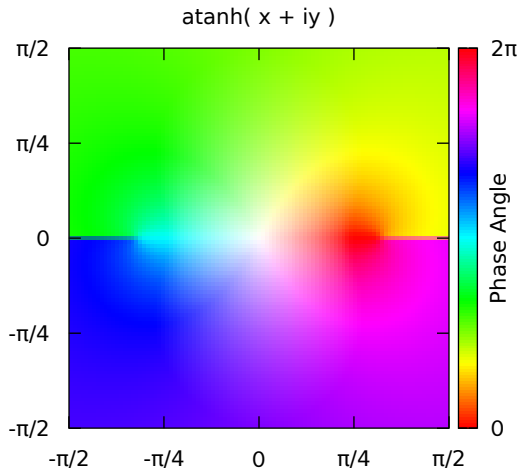


$\tanh(x + iy)$

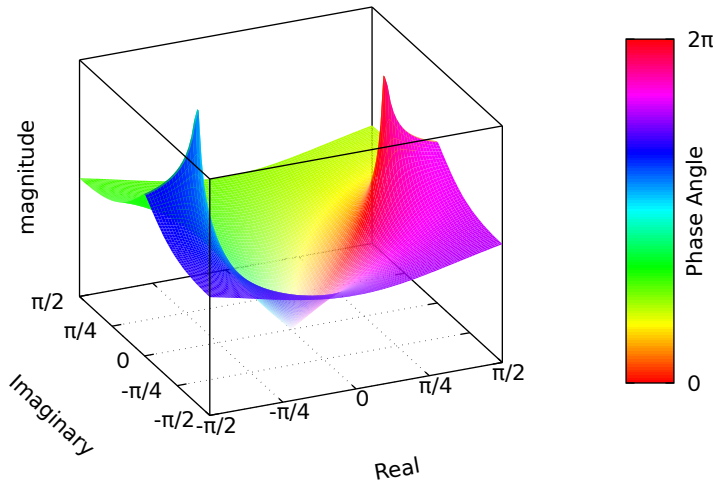




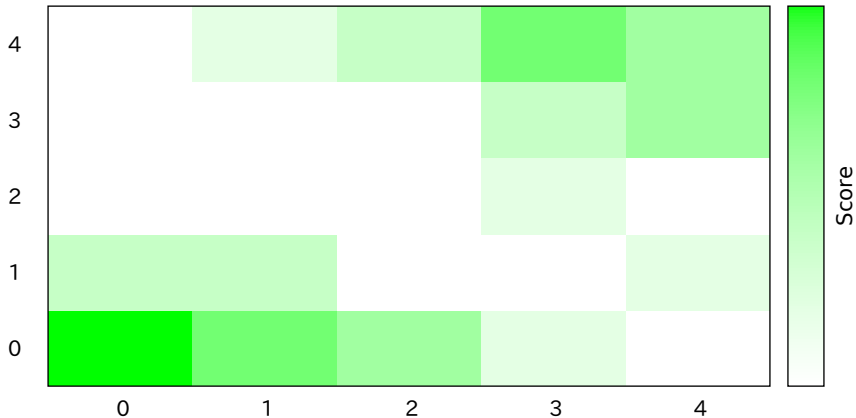




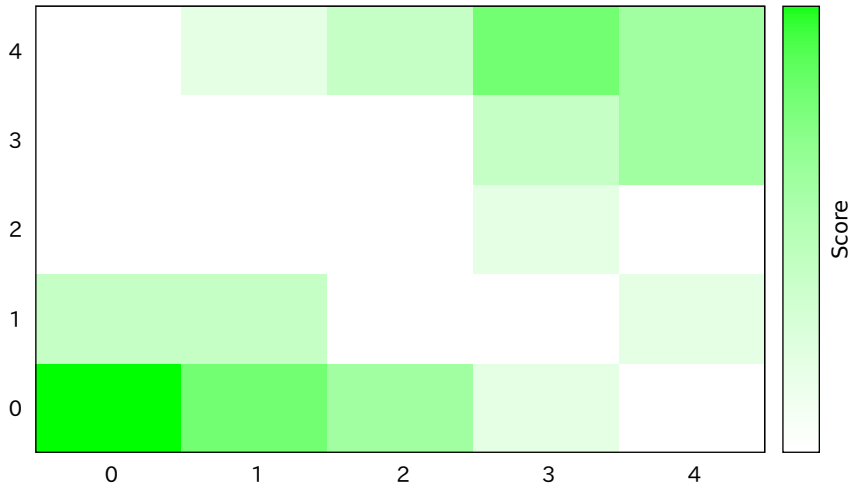
$$\operatorname{atanh}(x + iy)$$



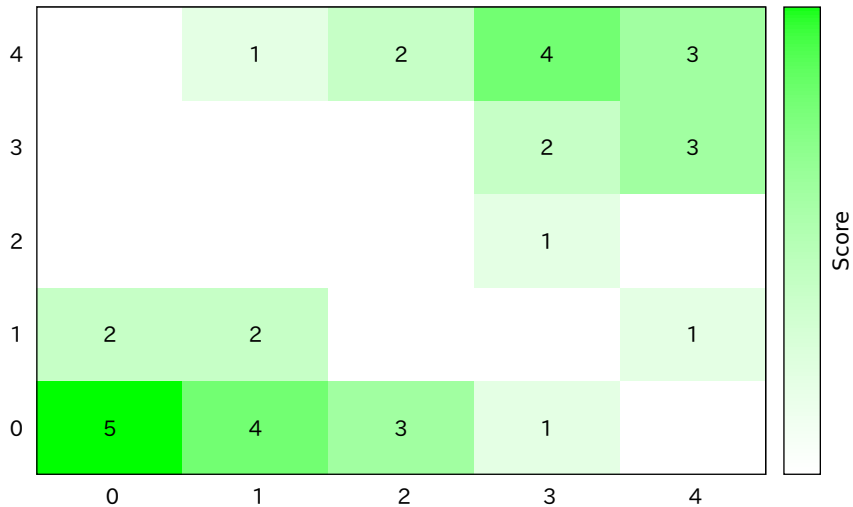
Heat Map generated from a file containing Z values only



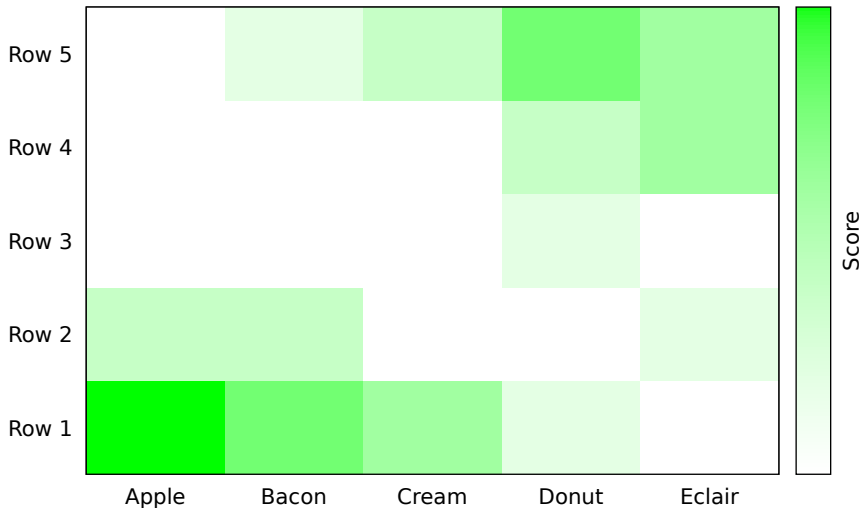
Heat Map generated by 'plot' from a stream of XYZ values
NB: Rows must be separated by blank lines!



Heat map with non-zero pixel values written as labels



Heat map from csv data with column and row labels

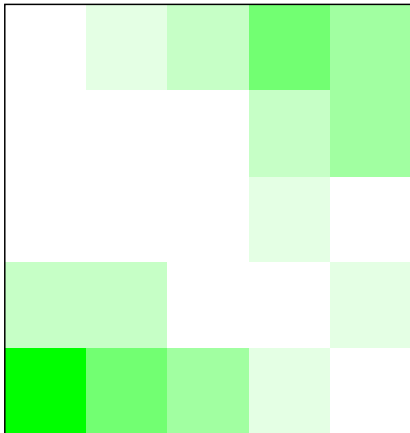


Compare 'image' and 'image pixels' modes

plot with image

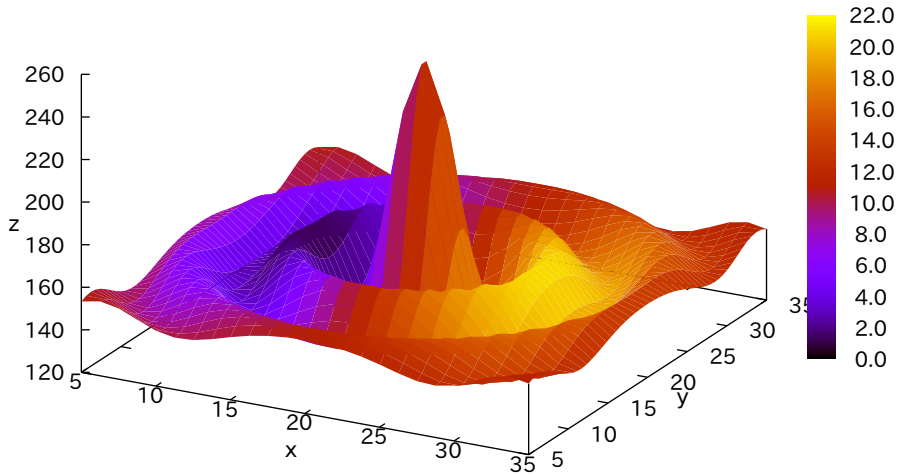


plot with image pixels

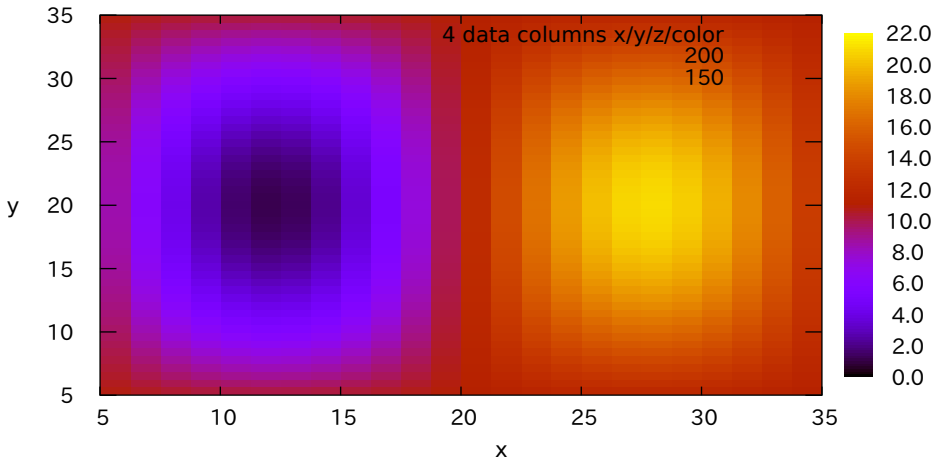


4D data (3D Heat Map)
Independent value color-mapped onto 3D surface

4 data columns x/y/z/color

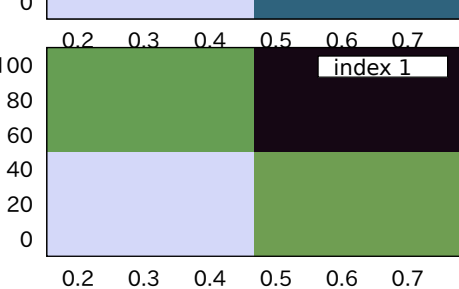
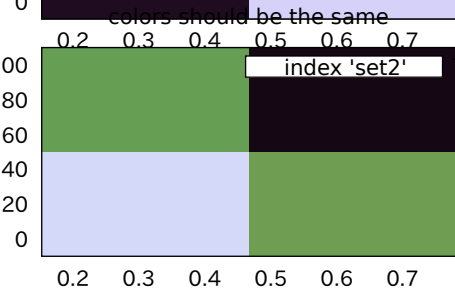
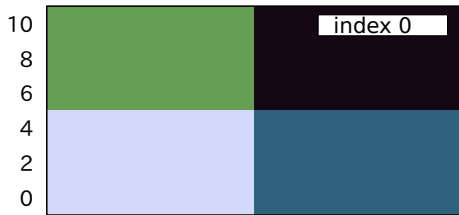
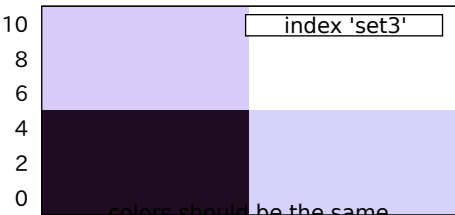


4D data (3D Heat Map)
Z is contoured. Independent value is color-mapped

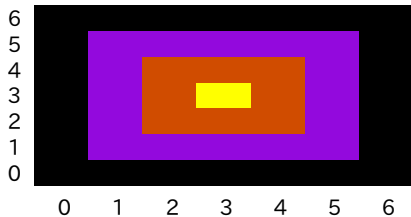


Data file contains labeled ascii matrices

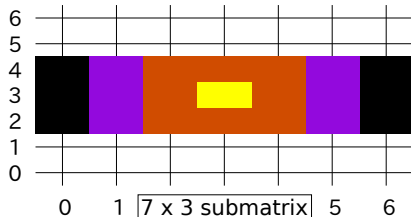
Y range should be the same



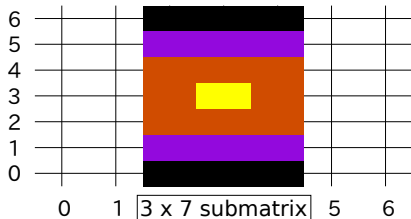
Full 7x7 matrix



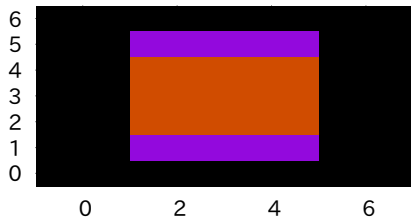
Subsample rows by every $::2::4$



Subsample columns by every $::2::4$



Sample alternate columns by every 2



$\gamma = 0.75$



$\gamma = 1.0$



$\gamma = 1.5$ (default)

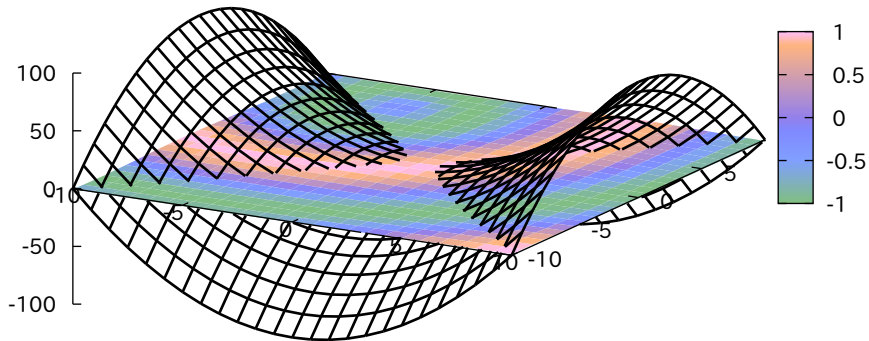


$\gamma = 2.0$

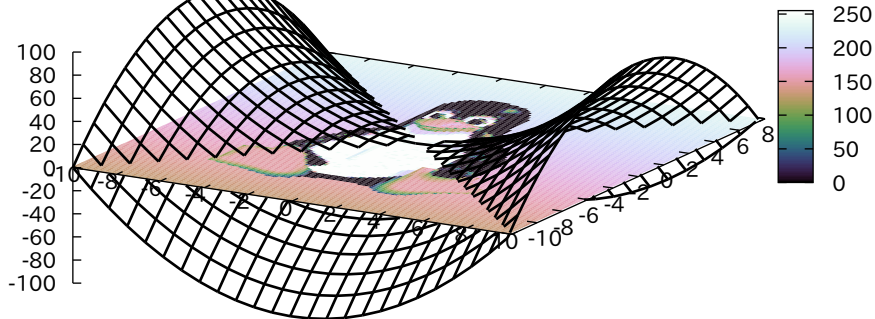


-10 -8 -6 -4 -2 0 2 4 6 8 10

Mixing pm3d surfaces with hidden-line plots

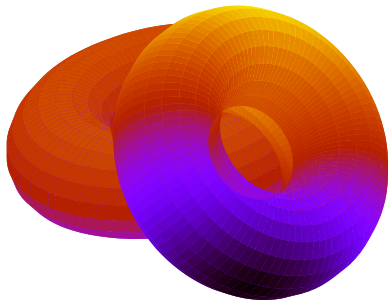


Mixing image surface with hidden-line plots

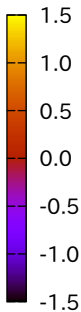
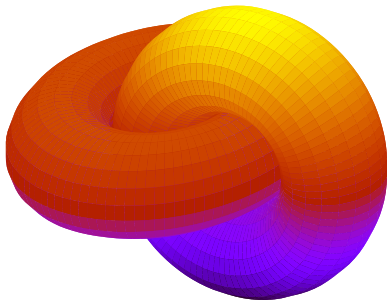


Interlocking Tori

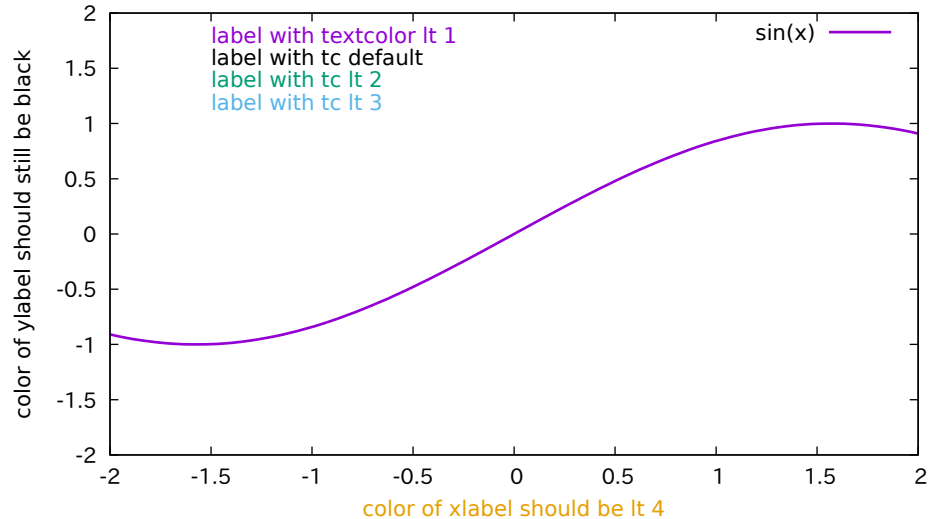
PM3D surface
no depth sorting



PM3D surface
depth sorting



Textcolor options in 2D plot (notice this title in color)



Textcolor options in plot (notice this title in color)

textcolor lt 1

tc It 2

tc lt 3

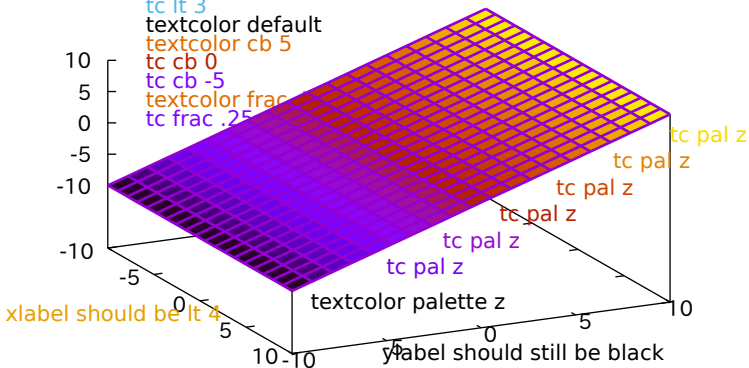
textcolor default

```
textcolor cb 5
```

tc cb 0

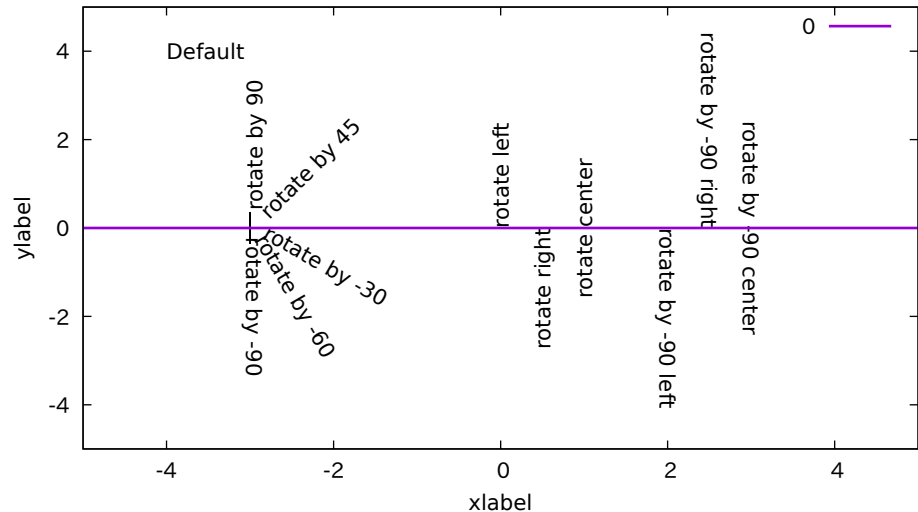
tc cb -5

textcolor frac

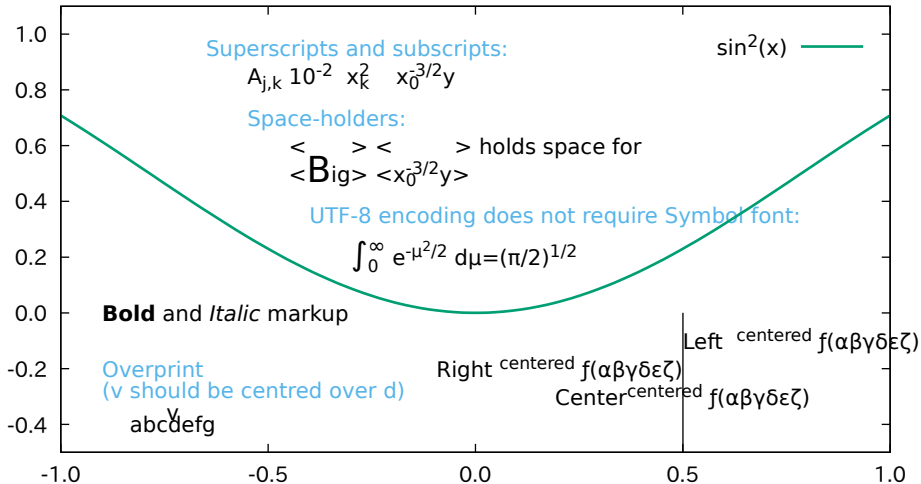
 $\text{tc frac} .25$ 

color cblabel

Rotation of label text

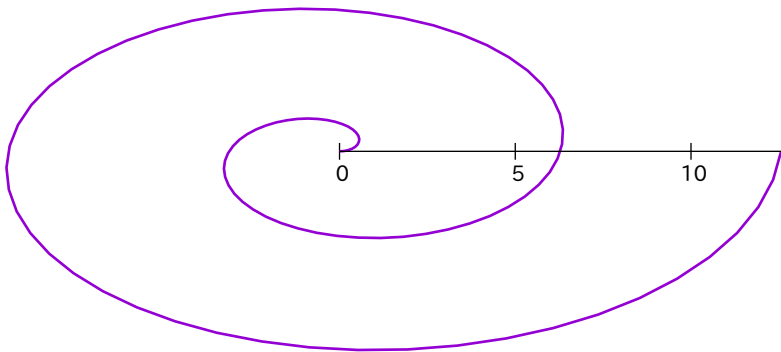


Demo of enhanced text mode using a single UTF-8 encoded font
 There is another demo that shows how to use a separate Symbol font



Enhanced text style markup

Default **Bold** *Italic* Default



Default *Italic* **Bold** Normal Default

Illustrate use of unicode escape sequences

unicode \U+221E : \U+221E Infinity

unicode \U+210F : \U+210F Planck constant h-bar

unicode \U+222C : \U+222C Double integral

unicode \U+03F5 : \U+03F5 Greek lunate epsilon

unicode \U+7403 : \U+7403 CJK unified ideograph 'sphere'

{a\U+0361}b : a\U+0361b Ligature tie (combining)

v\U+20D7 : v\U+20D7 Combining right arrow above

Terminal's native dashtypes

The diagram illustrates the mapping of 10 data types (dt 1 to dt 10) to five distinct line styles. The mapping is as follows:

Data Type	Line Style
dt 1	Solid
dt 2	Dashed
dt 3	Dotted
dt 4	Dash-dot
dt 5	Dash-dot
dt 6	Solid
dt 7	Dashed
dt 8	Dotted
dt 9	Dash-dot
dt 10	Dash-dot

Custom dashtypes

dt "."

dt "-"

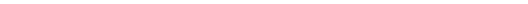
dt "._"

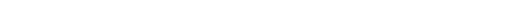
dt "..-"

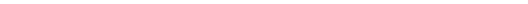
dt (50,6,2,6)

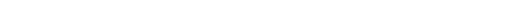
Terminal's native dashtypes

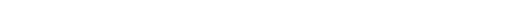
dt 1 

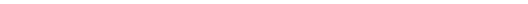
dt 2 

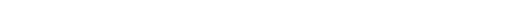
dt 3 

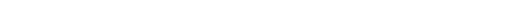
dt 4 

dt 5 

dt 6 

dt 7 

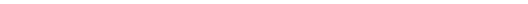
dt 8 

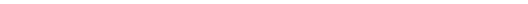
dt 9 

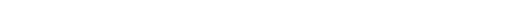
dt 10 

Custom dashtypes

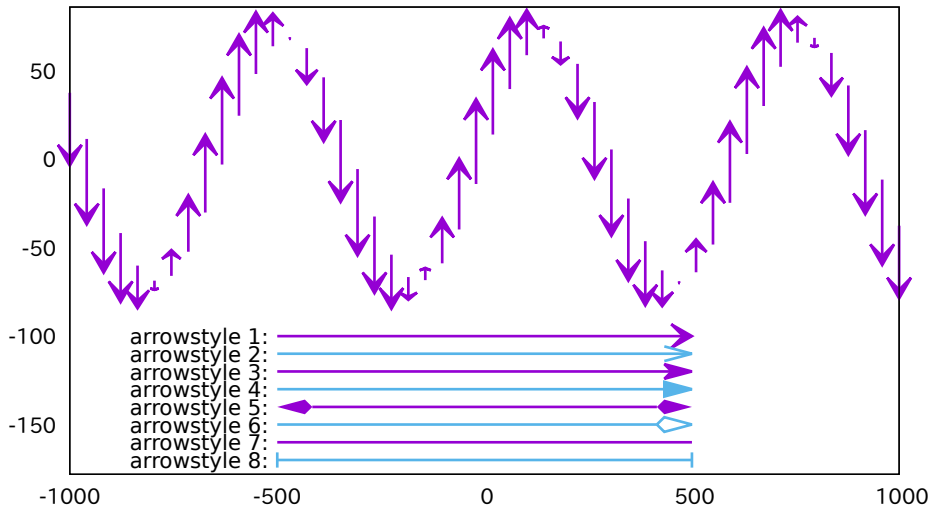
dt "." 

dt "-" 

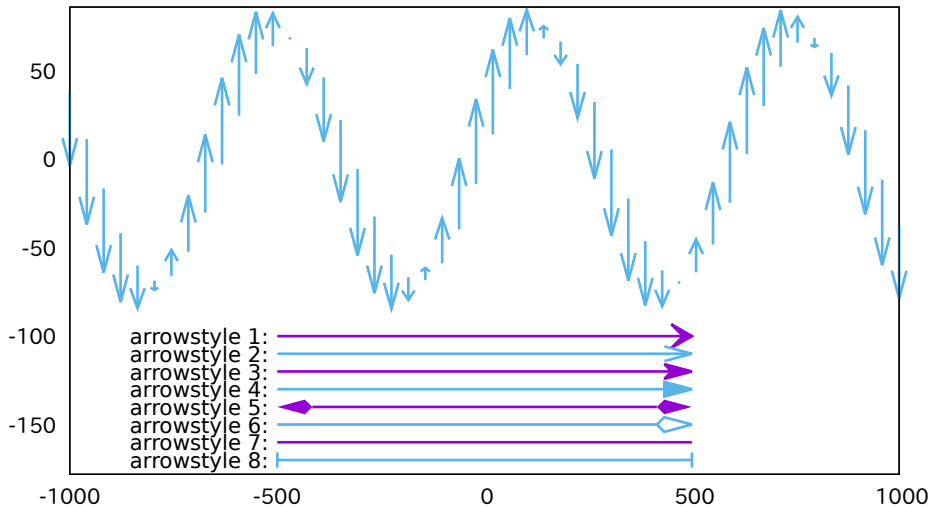
dt "._" 

dt "._-" 

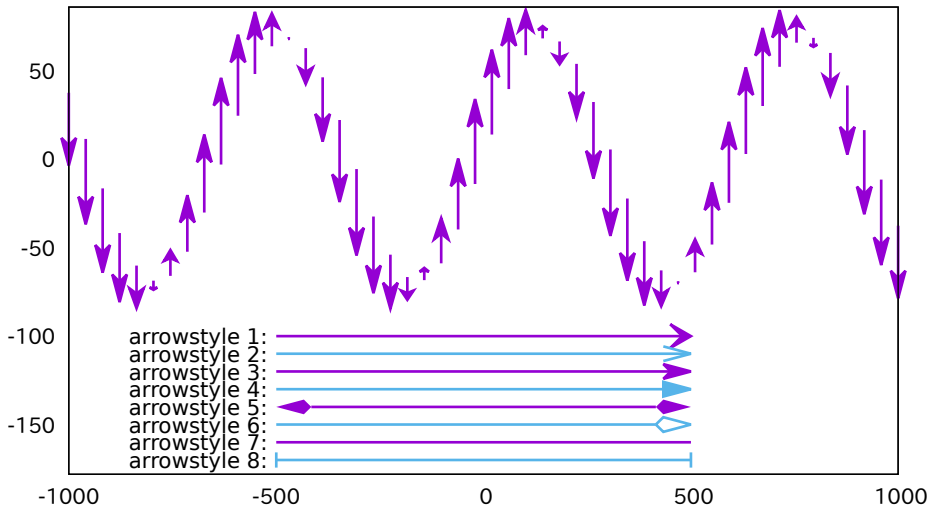
Top: plot with vectors arrowstyle 1, Bottom: explicit arrows



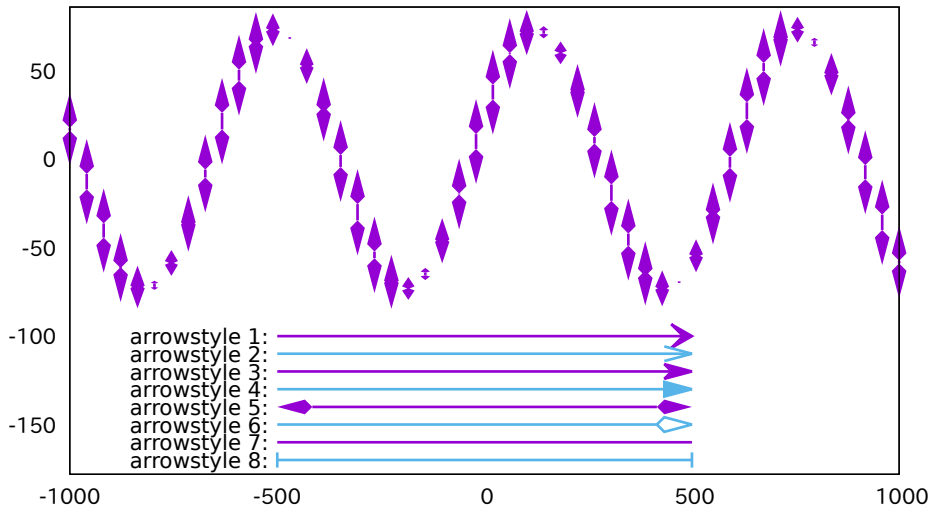
Top: plot with vectors arrowstyle 2, Bottom: explicit arrows



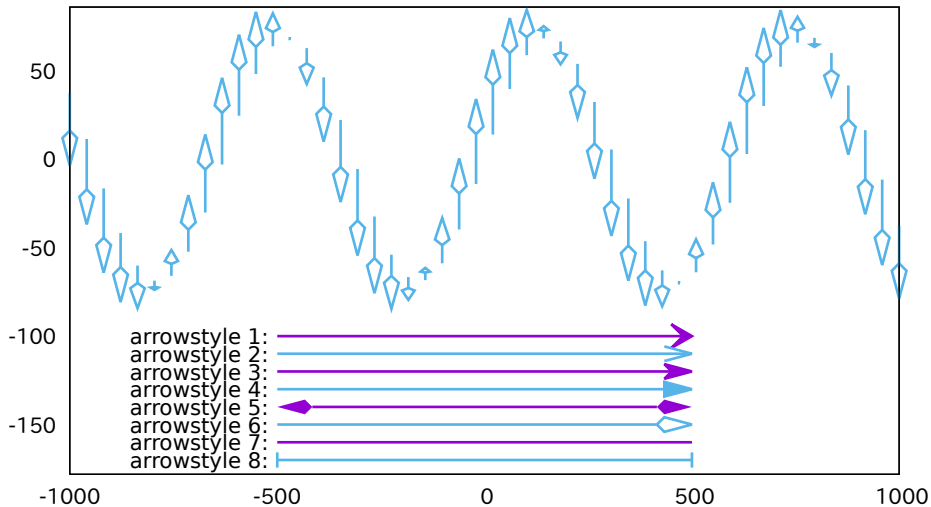
Top: plot with vectors arrowstyle 3, Bottom: explicit arrows



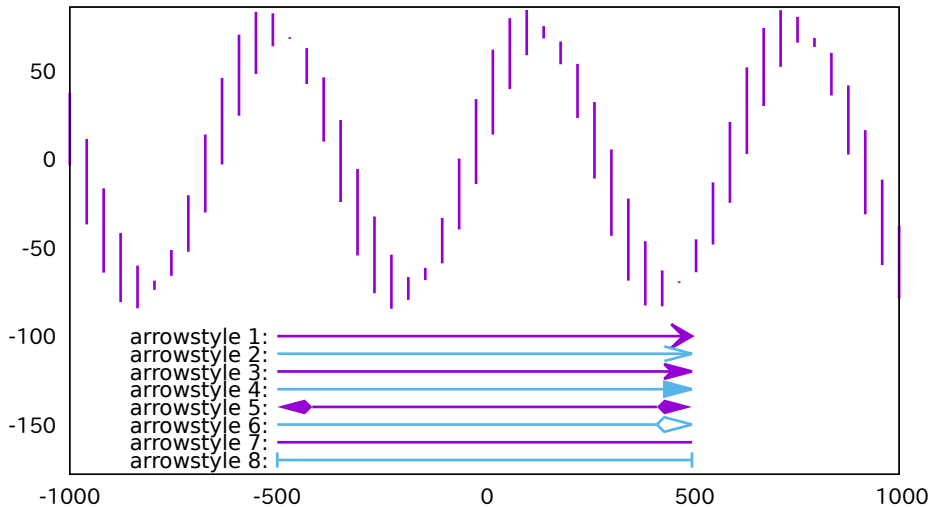
Top: plot with vectors arrowstyle 5, Bottom: explicit arrows



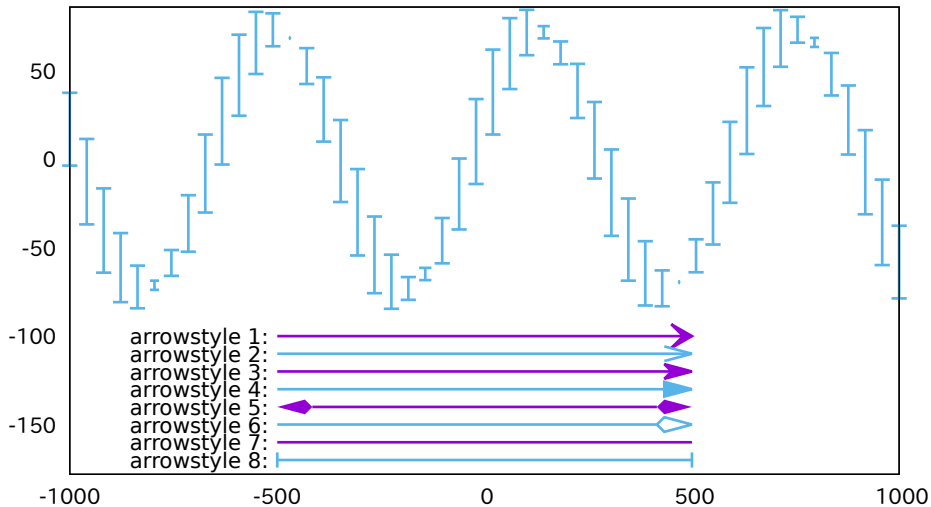
Top: plot with vectors arrowstyle 6, Bottom: explicit arrows



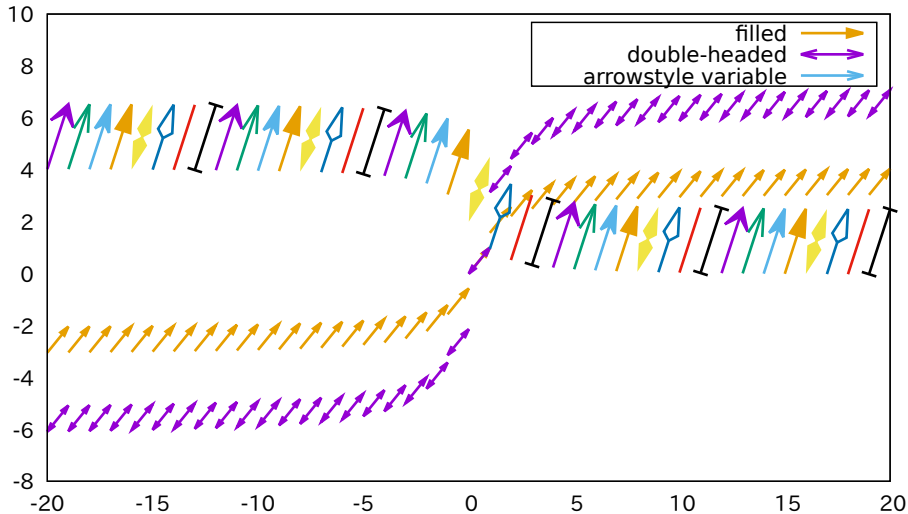
Top: plot with vectors arrowstyle 7, Bottom: explicit arrows

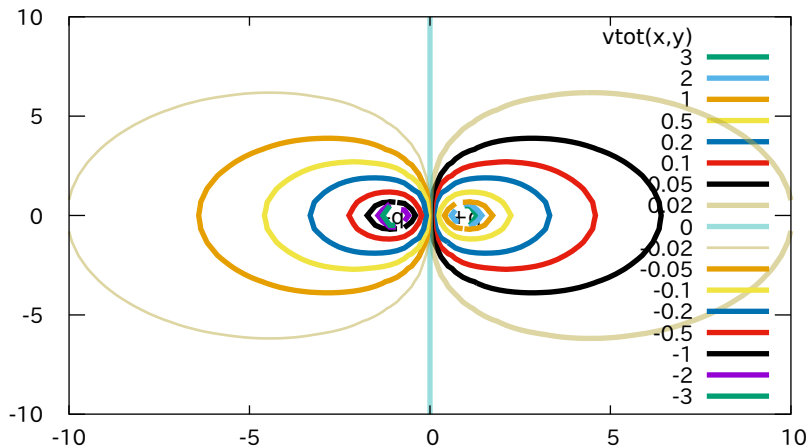


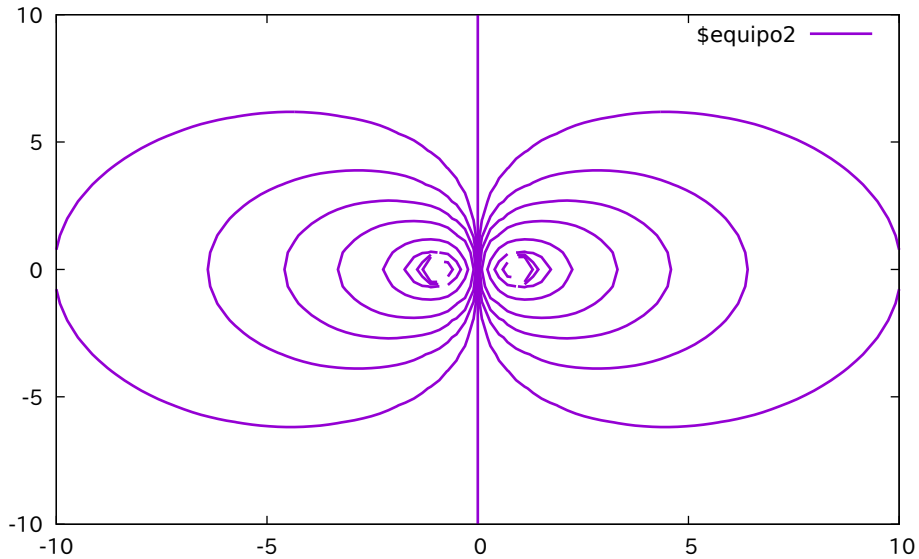
Top: plot with vectors arrowstyle 8, Bottom: explicit arrows

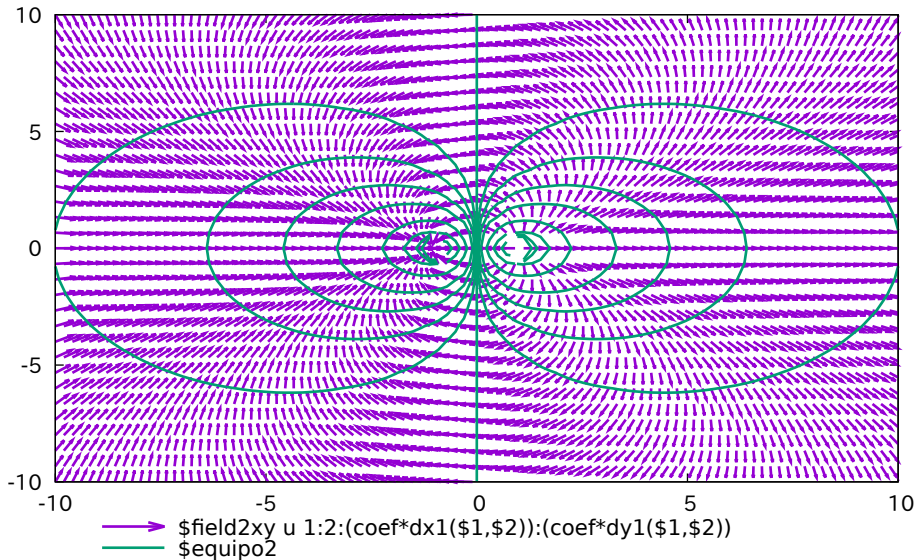


Plot 'file' with vectors <arrowstyle>

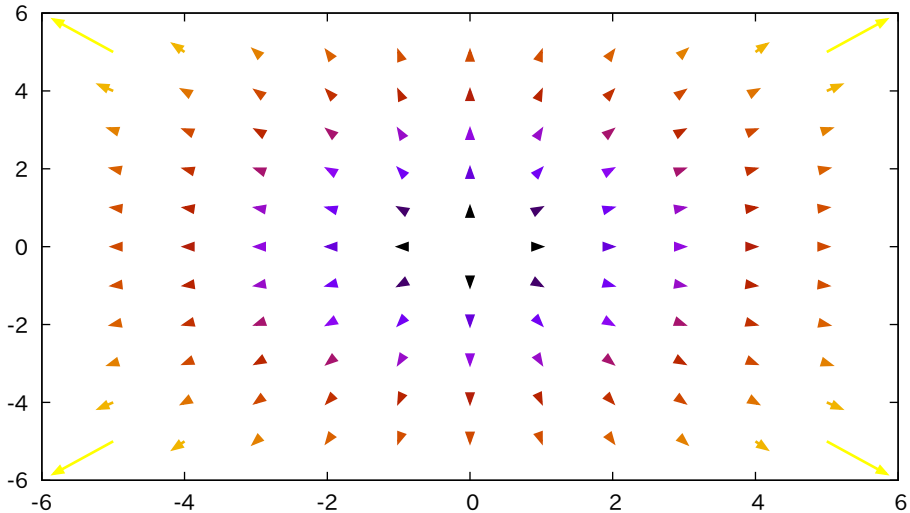




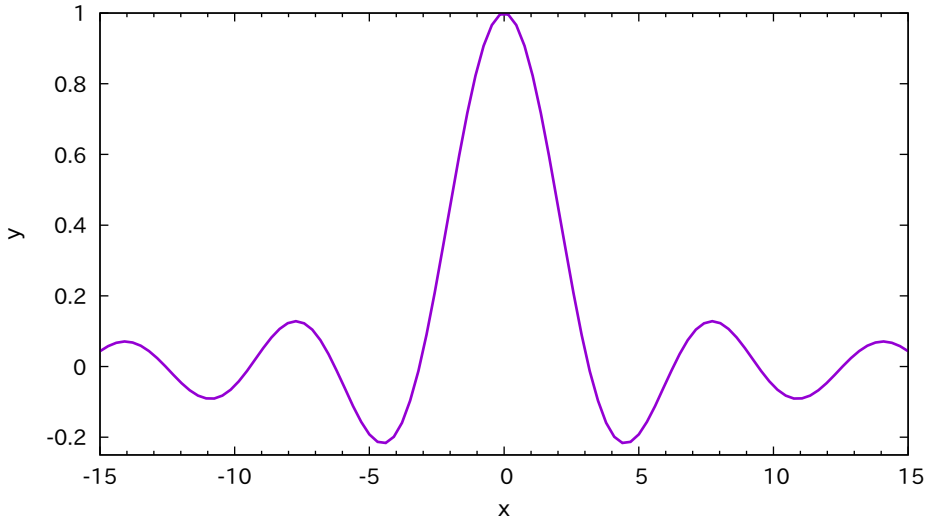




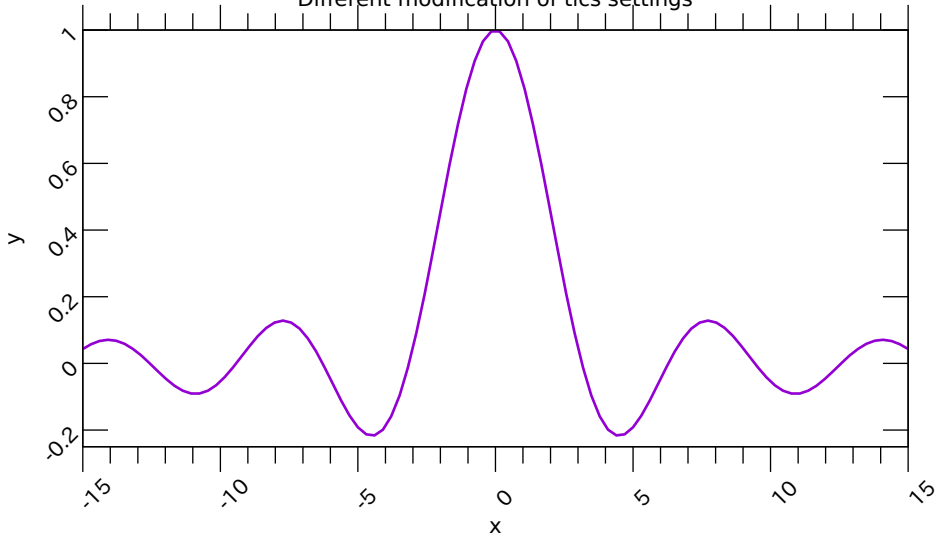
fixed size arrowheads for very short vectors



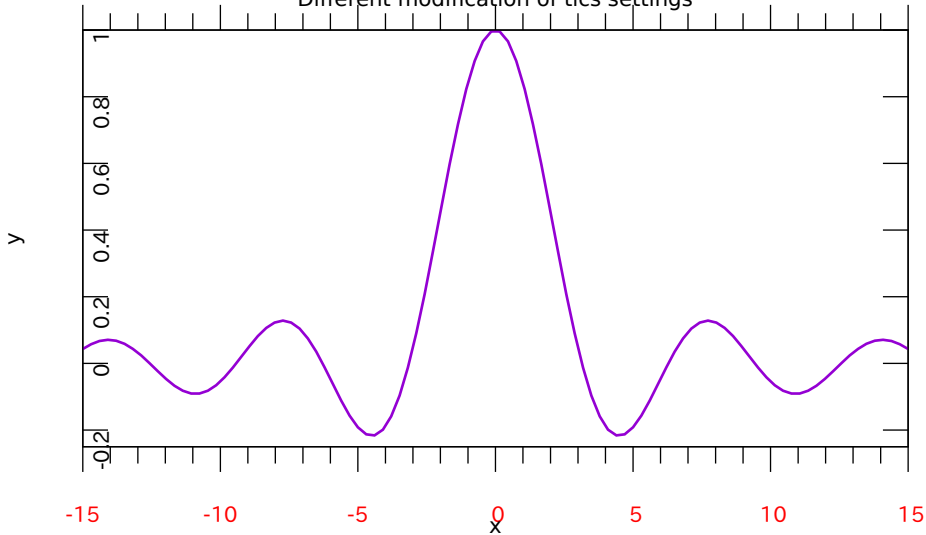
Default tics settings



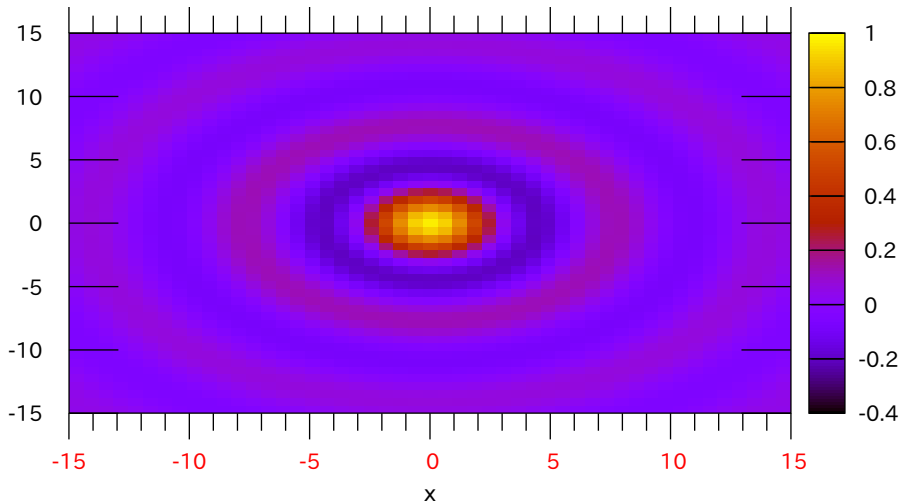
Different modification of tics settings



Different modification of tics settings

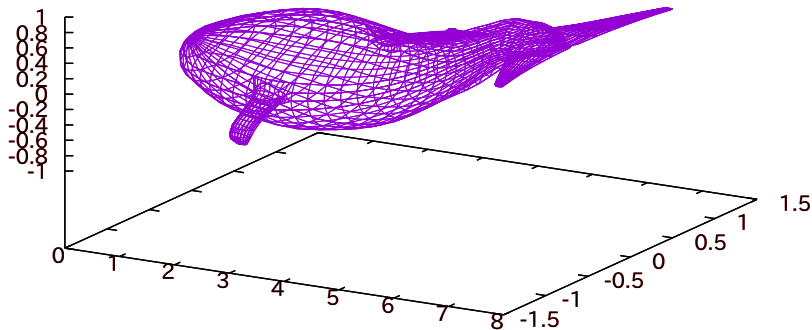


Modified tics settings (pm3d palette with colorbar)

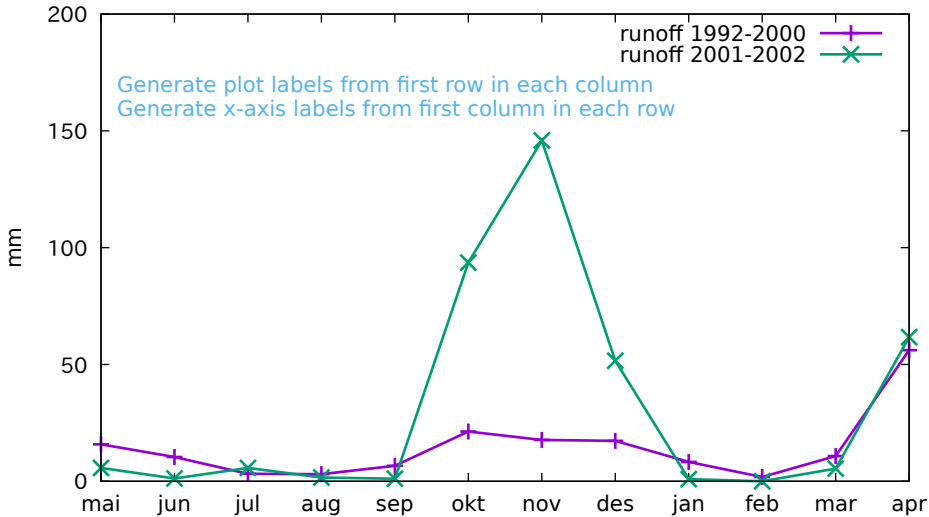


Nothing interesting here, just a unit test for volatile, skip, and refresh

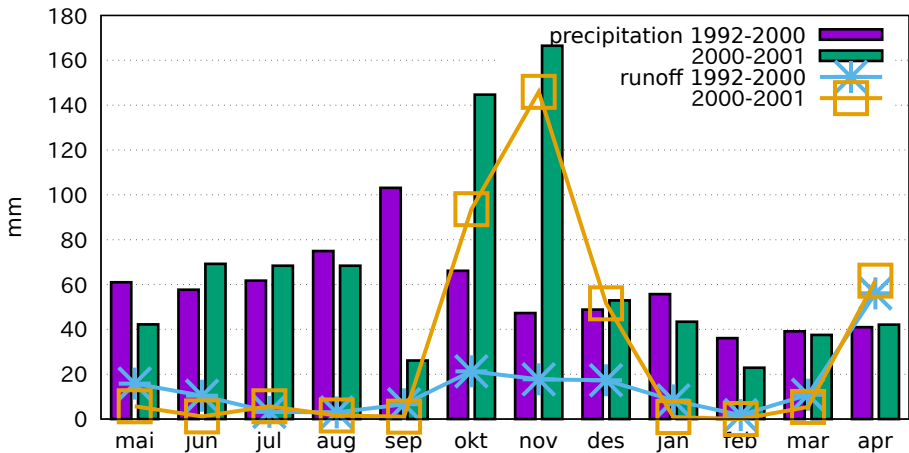
"whale.dat" skip 5 volatile —



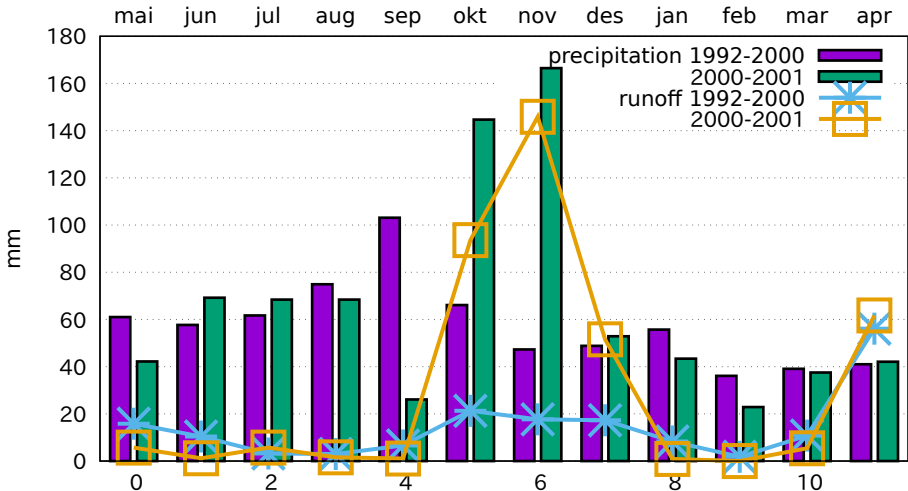
Auto-labeling plots from text fields in datafile



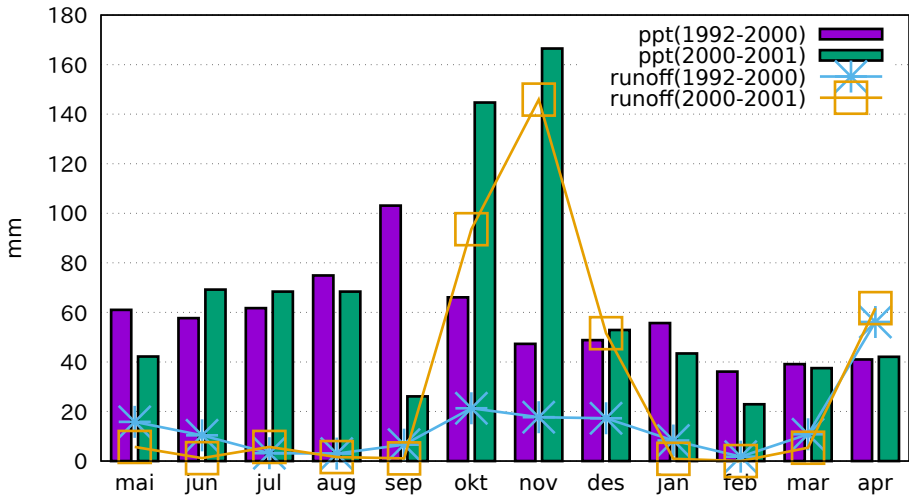
Read tic labels from a datafile column
An approximation of Hans Olav Eggstad's categoric plot patch
using 'using (\$0):2:xticlabels(1)' and 'set style fill solid border -1'



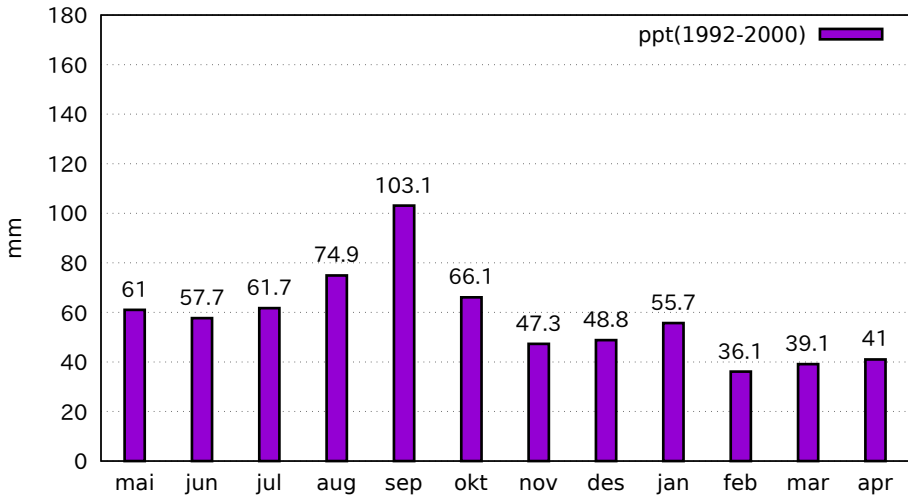
Same plot using x2ticlabels also



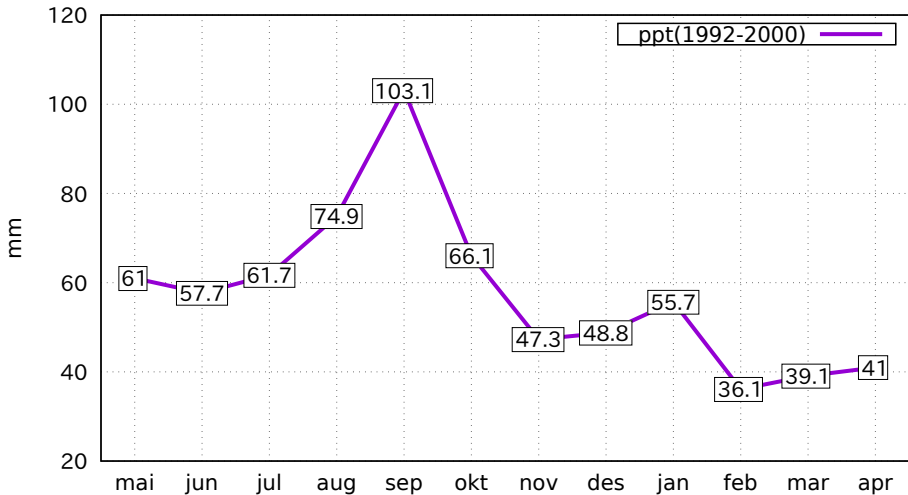
Plot from table format (titles taken from column headers)



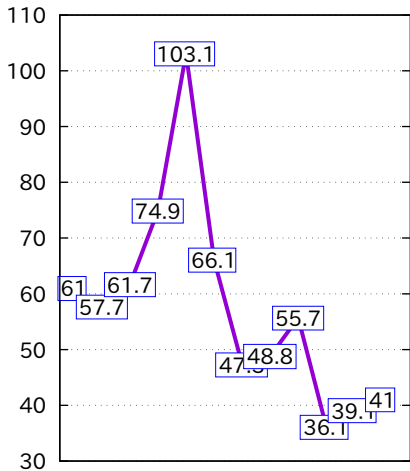
Plot actual y-value as a label



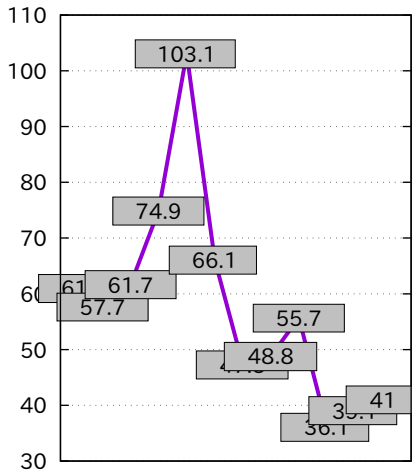
Plot using boxed labels



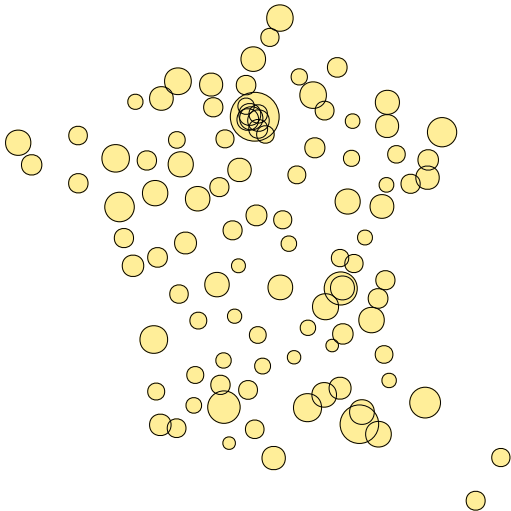
textboxes with blue border



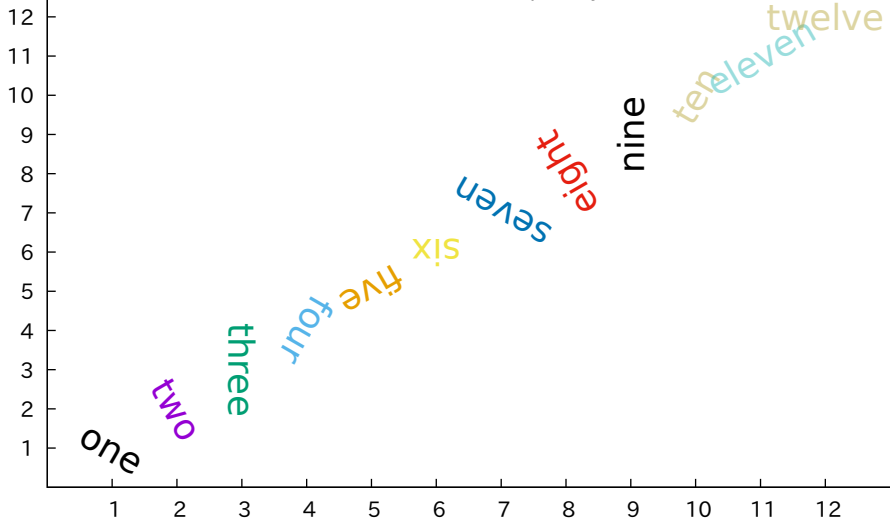
larger textboxes with grey fill



Hypertext is shown when the mouse is over a point

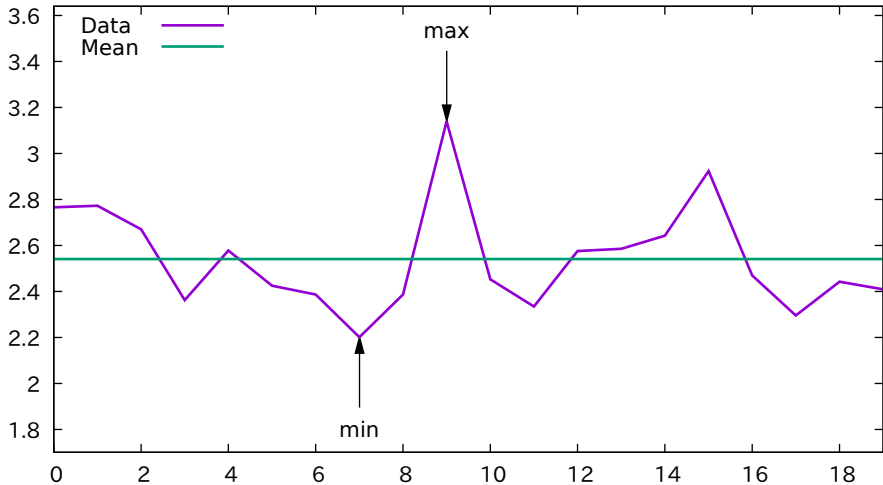


variable color and orientation in plotstyle 'with labels'

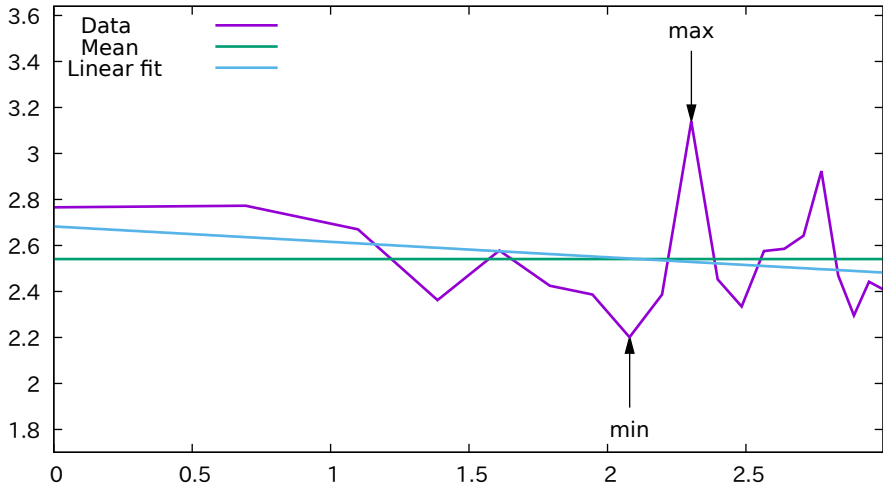


Use of stats command to find min/max/mean before plotting

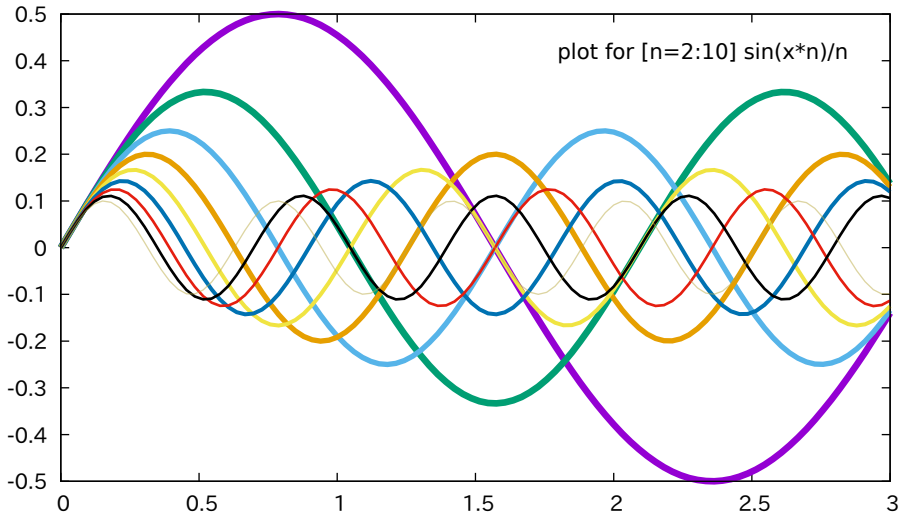
One data column



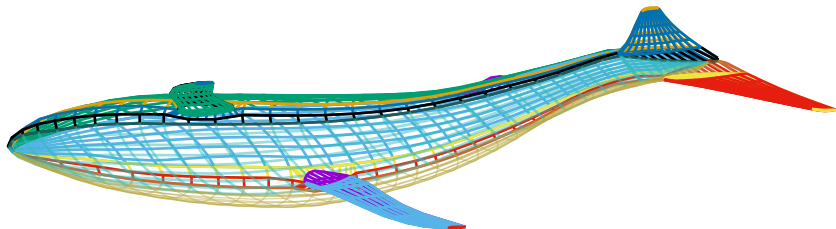
Use of stats command to find min/max/mean before plotting
Two data columns



Iteration within plot command



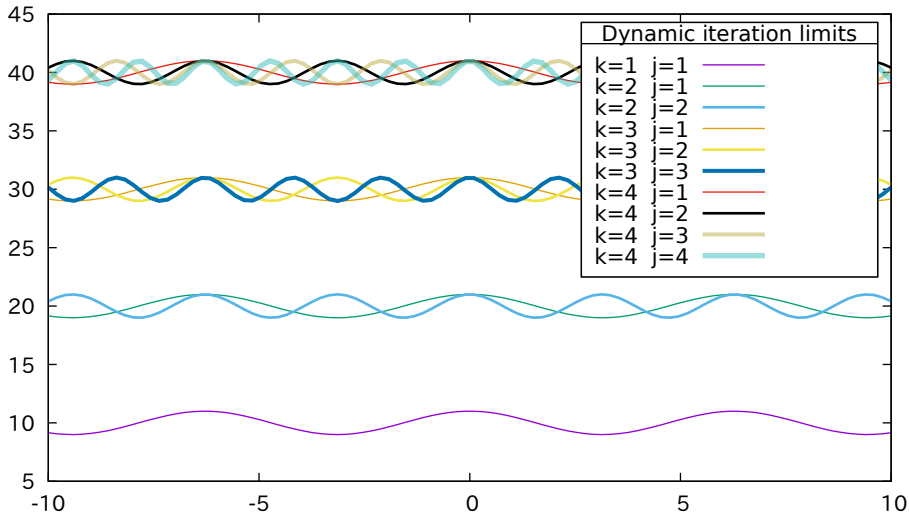
Iteration over all available data in a file



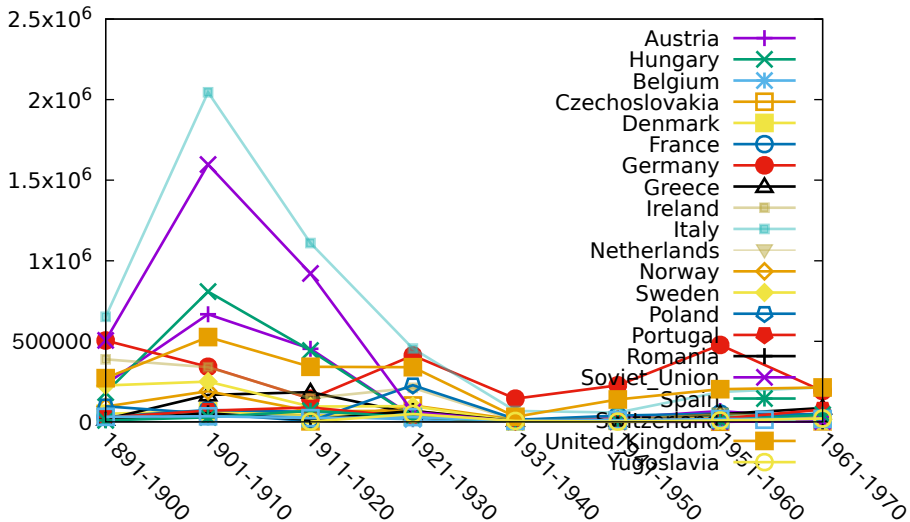
plot for [scan=1:~] 'whale.dat' index scan

scan 1	—	scan 7	—	scan 13	—	scan 19	—
scan 2	—	scan 8	—	scan 14	—	scan 20	—
scan 3	—	scan 9	—	scan 15	—	scan 21	—
scan 4	—	scan 10	—	scan 16	—	scan 22	—
scan 5	—	scan 11	—	scan 17	—	scan 23	—
scan 6	—	scan 12	—	scan 18	—		

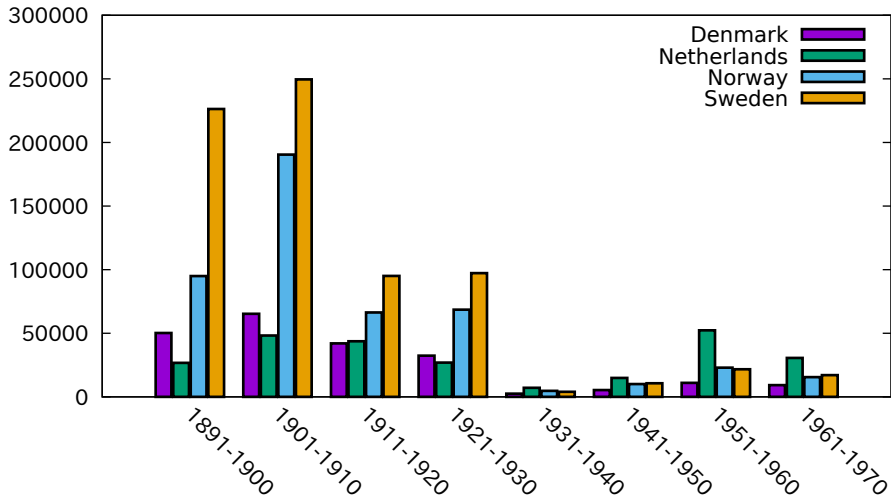
plot for $[i=1:4]$ for $[k=i:i]$ for $[j=1:k]$ $10*k + \cos(j*x)$



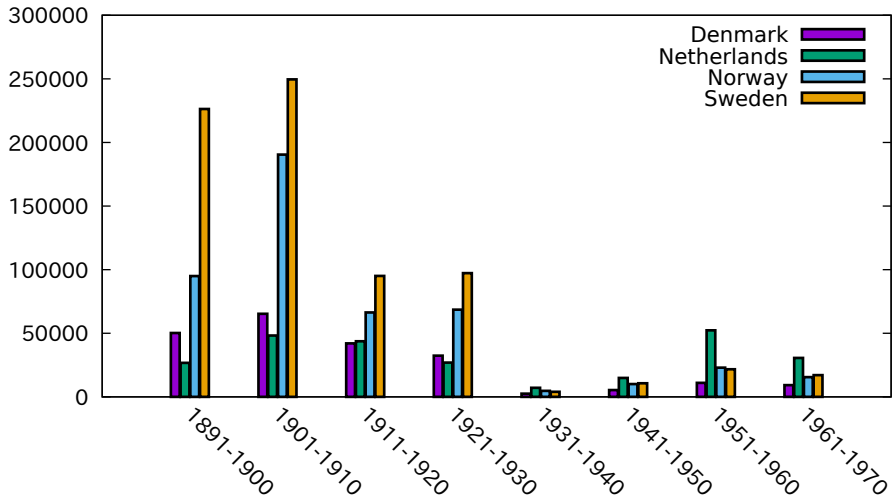
US immigration from Europe by decade



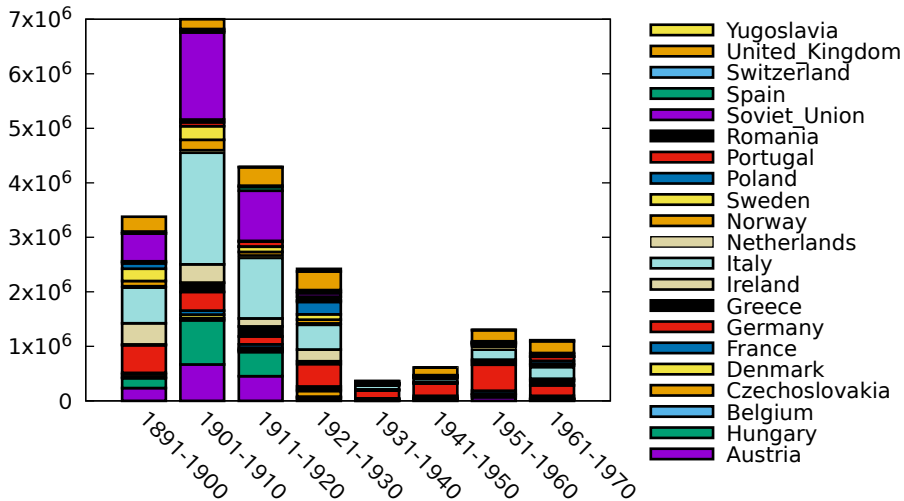
US immigration from Northern Europe
Plot selected data columns as histogram of clustered boxes



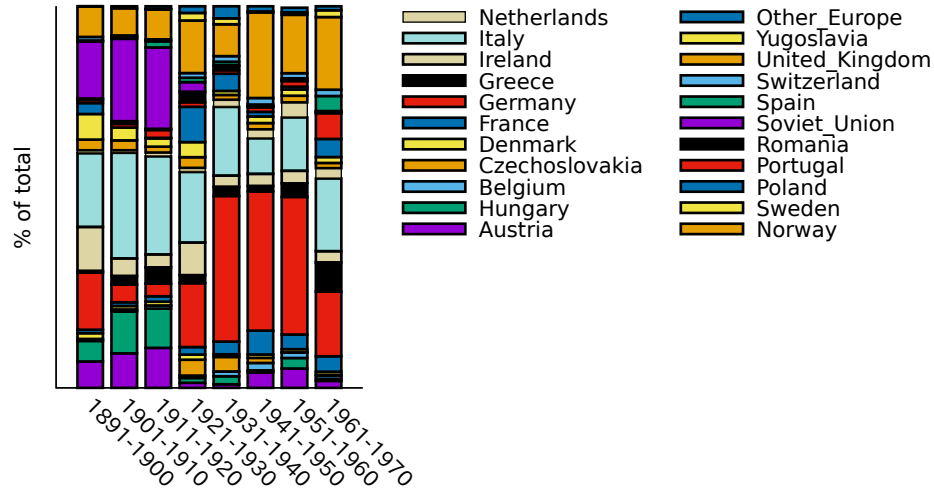
US immigration from Northern Europe
(same plot with larger gap between clusters)



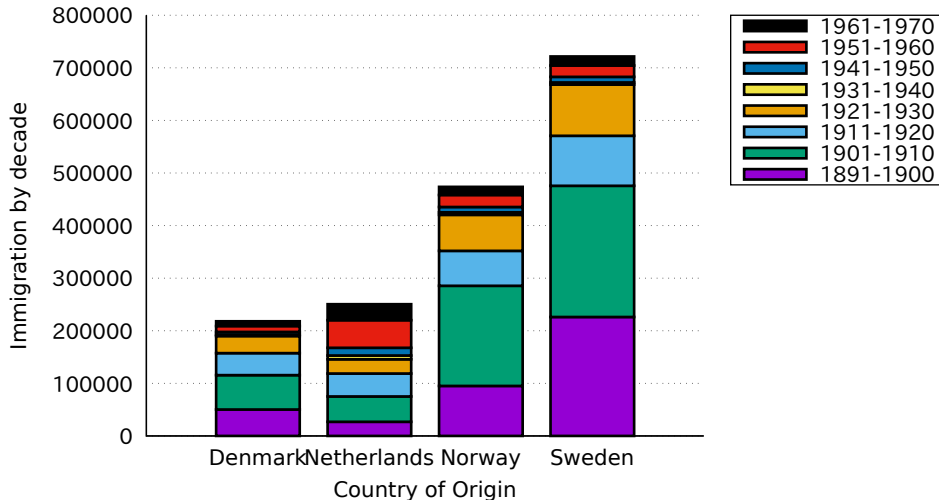
US immigration from Europe by decade
Plot as stacked histogram



US immigration from Europe by decade
 portion of total plotted as stacked histogram

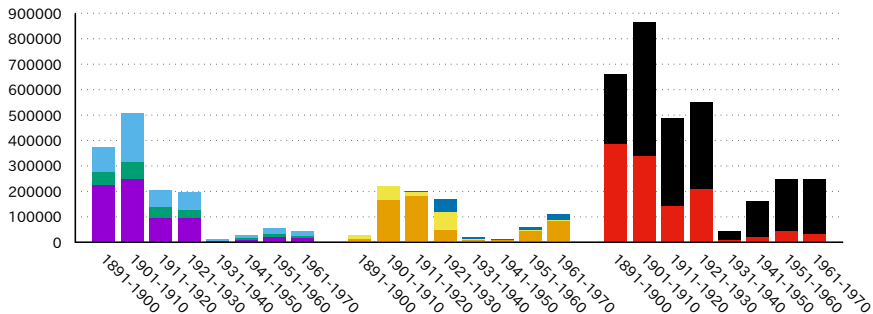


Immigration from Northern Europe
(columnstacked histogram)



Immigration from different regions
(give each histogram a separate title)

Immigration by decade



Northern Europe
(Same plot using rowstacked rather than clustered histogram)

Southern Europe

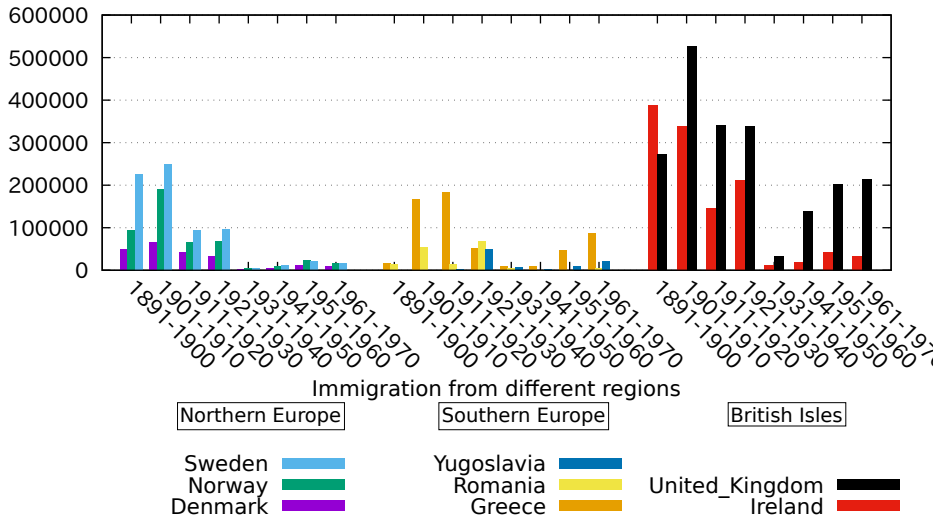
British Isles

Sweden
Denmark
Norway

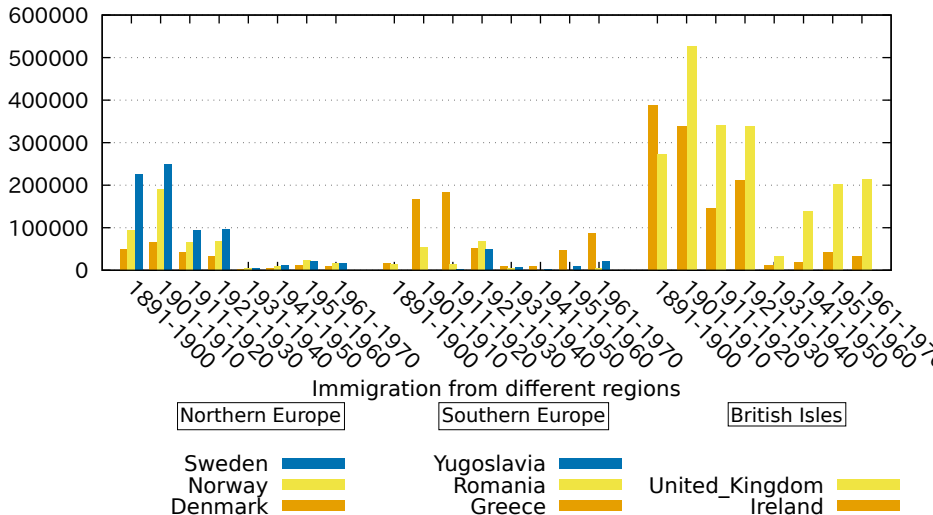
Greece
Romania
Yugoslavia

Ireland
United_Kingdom

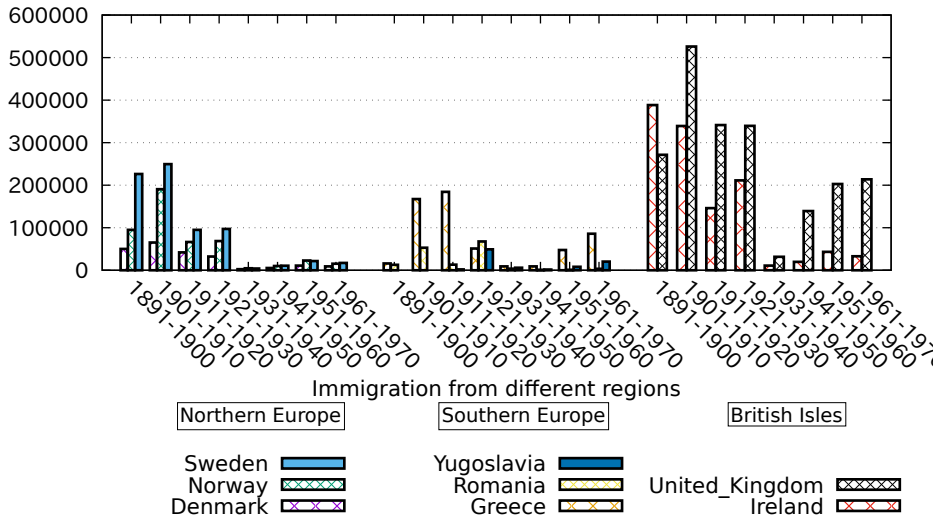
Default Histogram Colouring



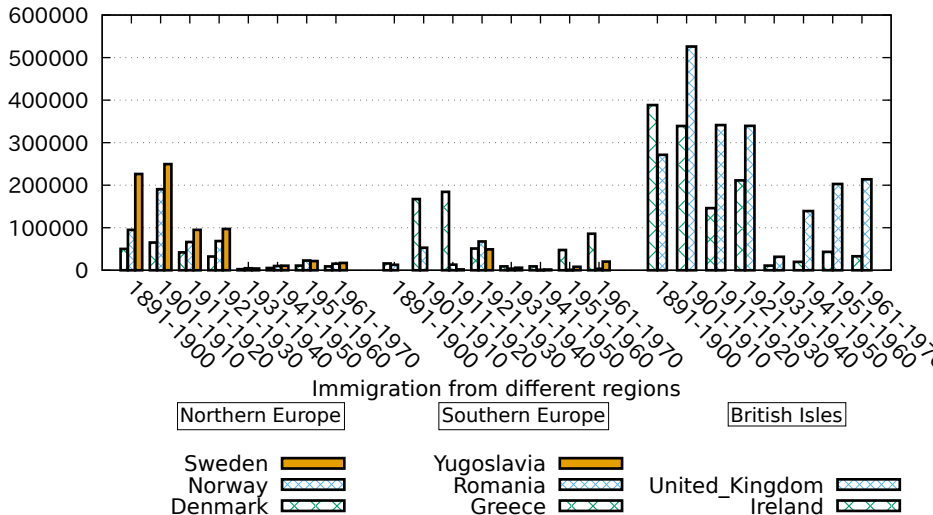
Explicit start color in 'newhistogram' command



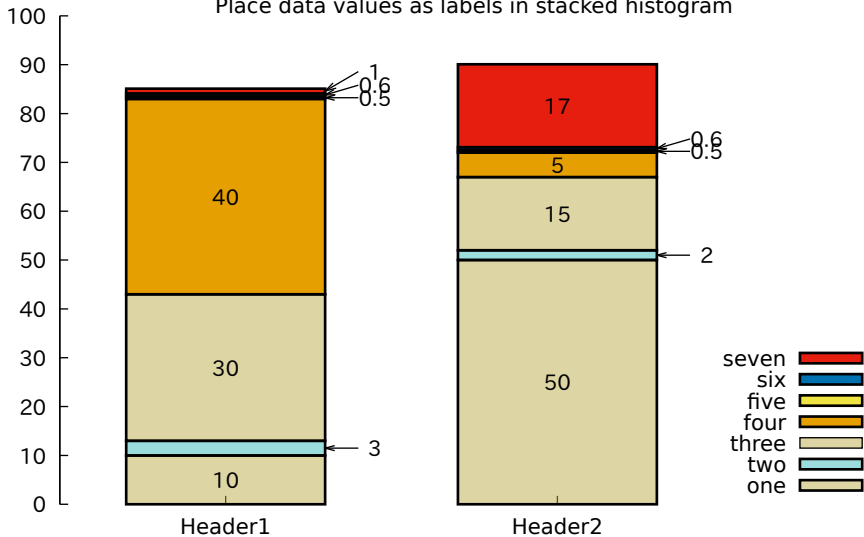
Explicit start pattern in 'newhistogram' command



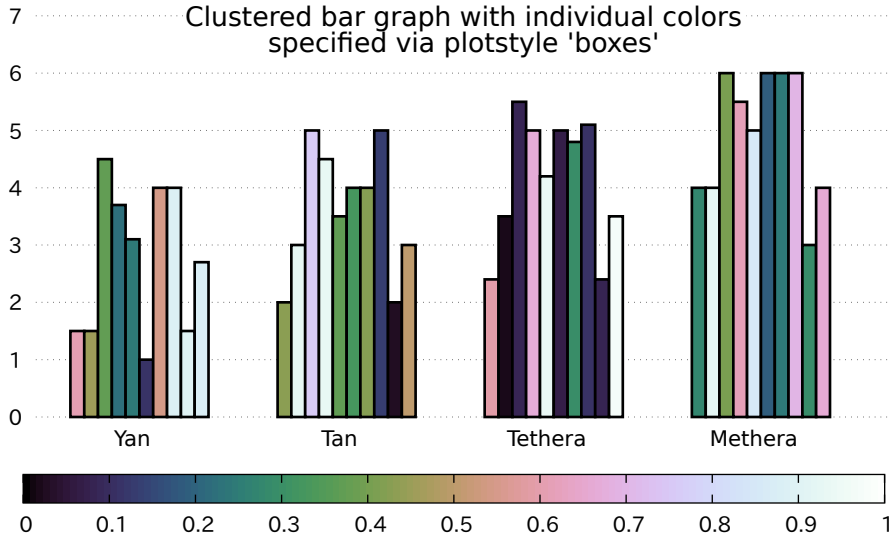
Explicit start pattern and linetype



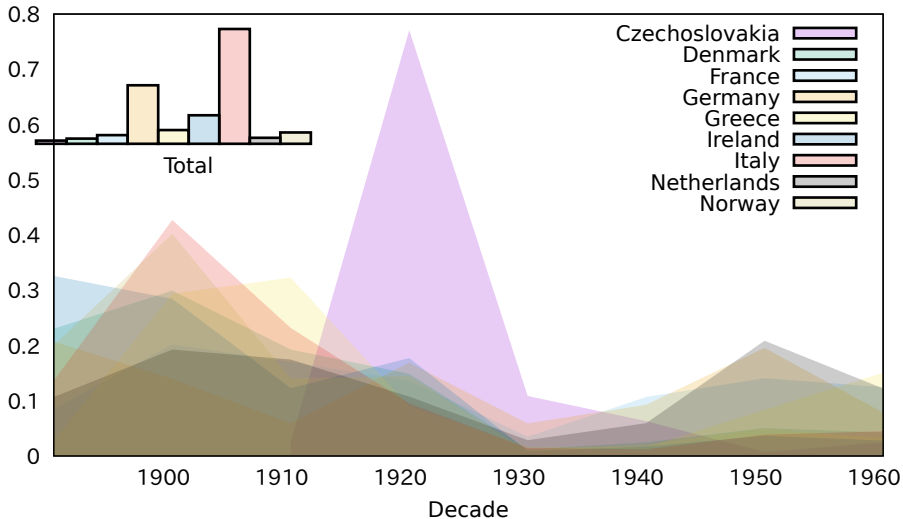
Place data values as labels in stacked histogram



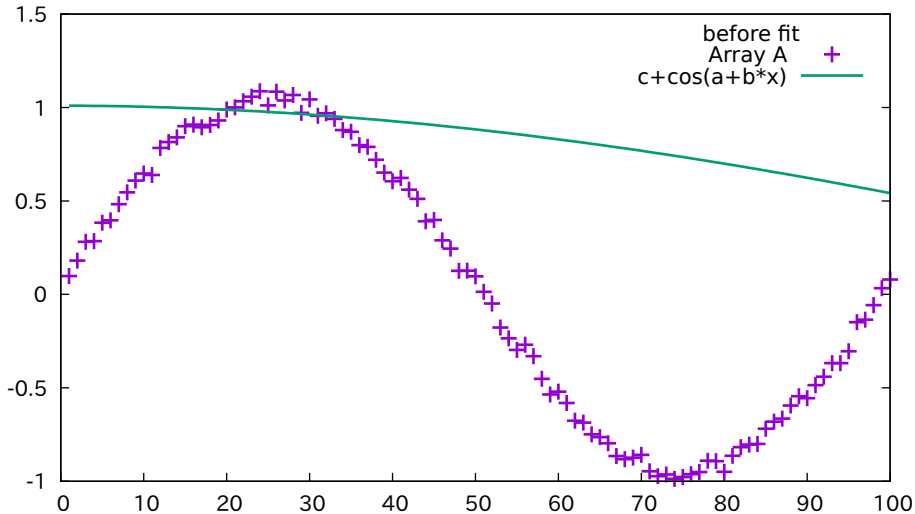
Clustered bar graph with individual colors
specified via plotstyle 'boxes'



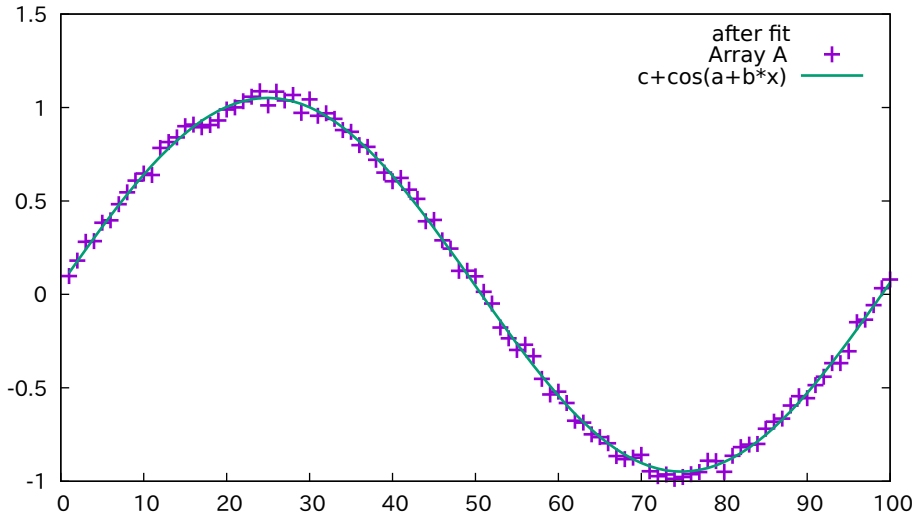
Use of an array to aid normalization and to plot summed values



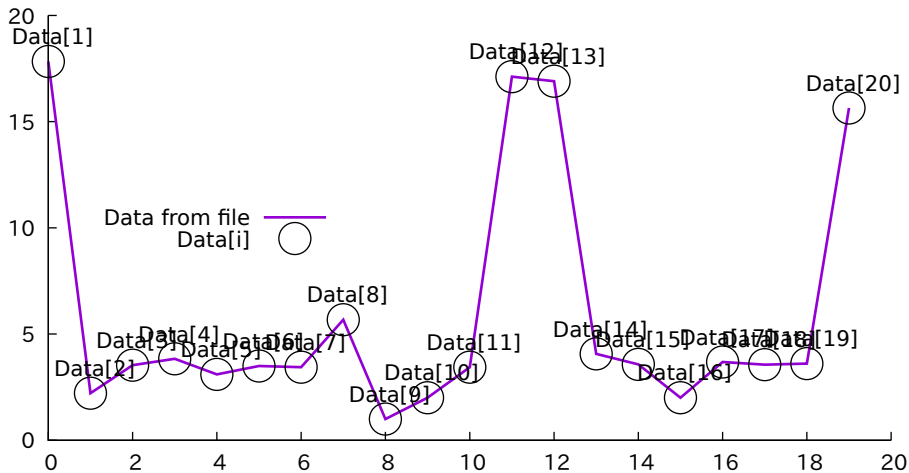
Fit function to values stored in an array



Fit function to values stored in an array



Illustrate loading an array from a column in a data file
Note that first data point in the file is 'line 0'
but it goes into array element Data[1]



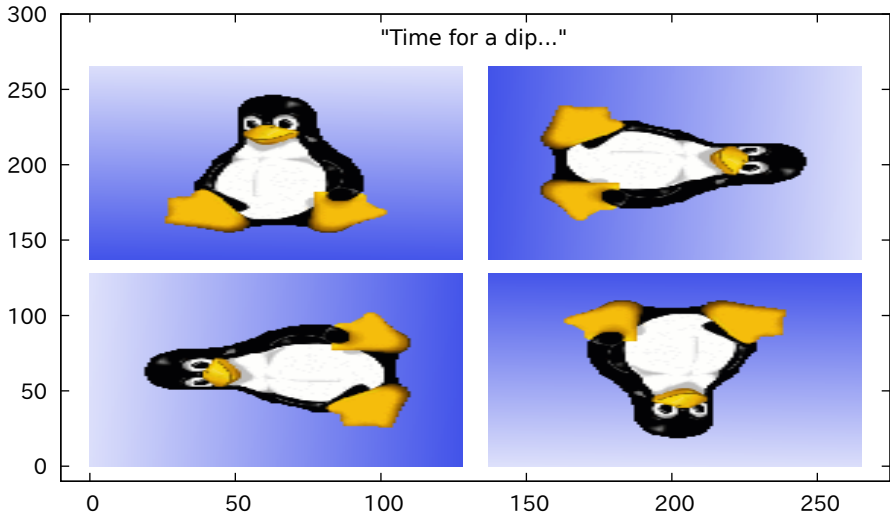
Larry Ewing's GIMP penguin on vacation basking in
the balmy waters off the coast of Murmansk

'blutux.rgb' binary array=(128,128) flipy format='%uchar'

"I flew here... by plane. Why? For the halibut."

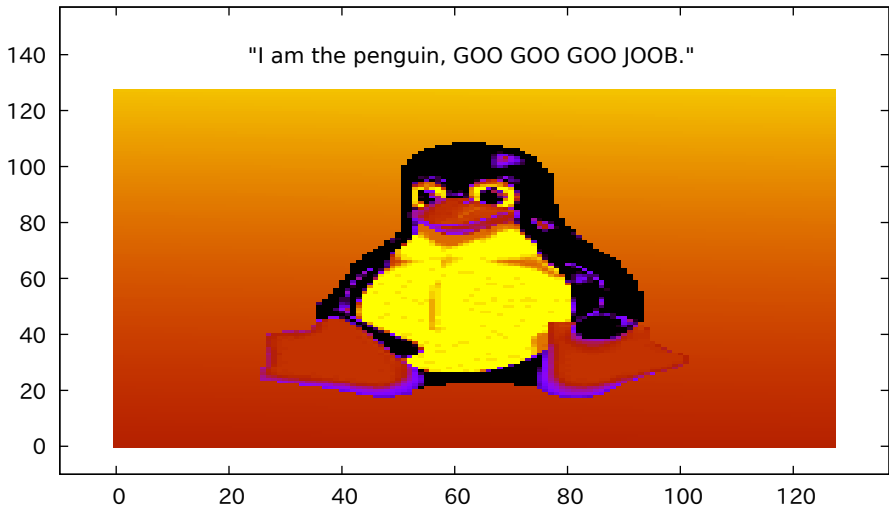


Translations of position variables via 'using'



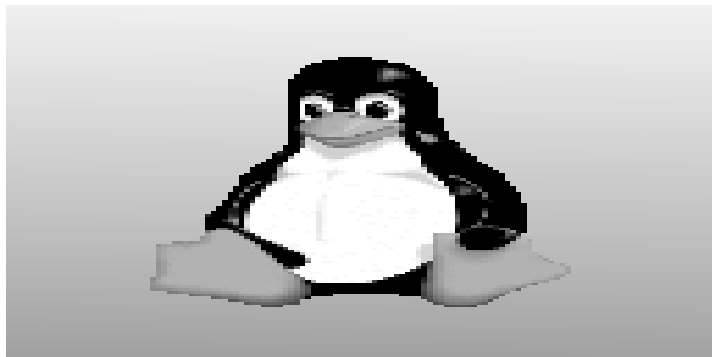
Palette mode 'image' used to produce psychedelic bird

"I am the penguin, GOO GOO GOO JOOB."

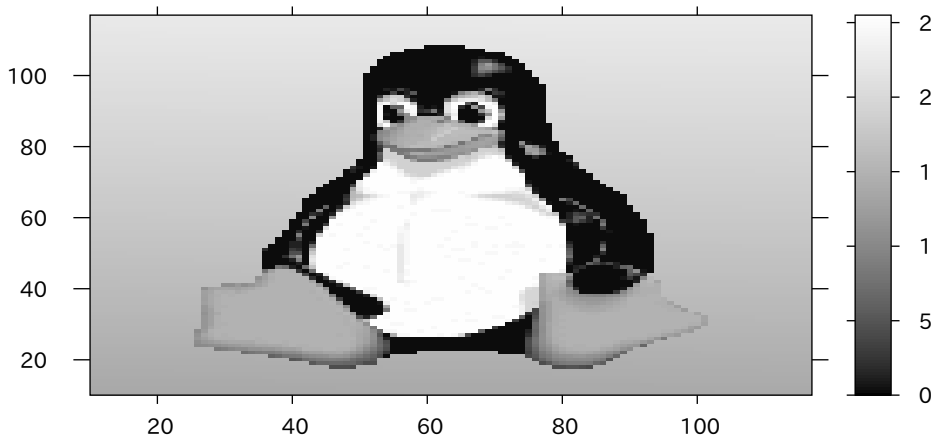


The palette can be changed from color to gray scale

"This picture was taken by my friend Ansel Adams."

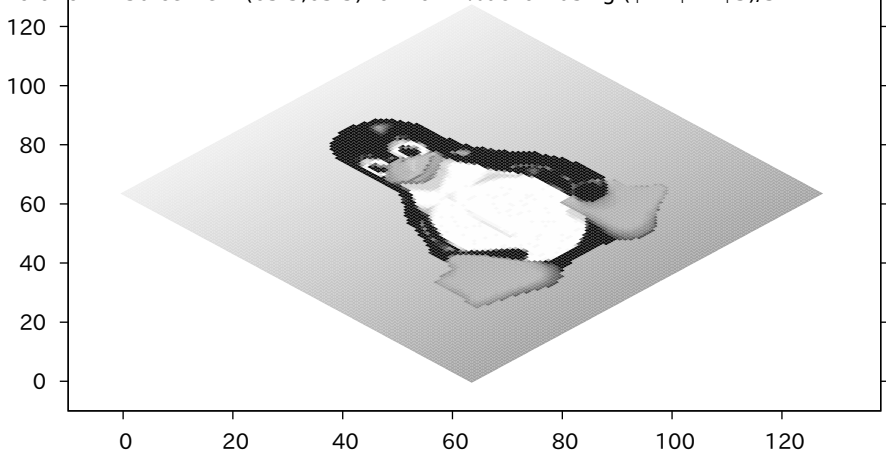


As with 3d color surfaces, a color box may be added to the plot

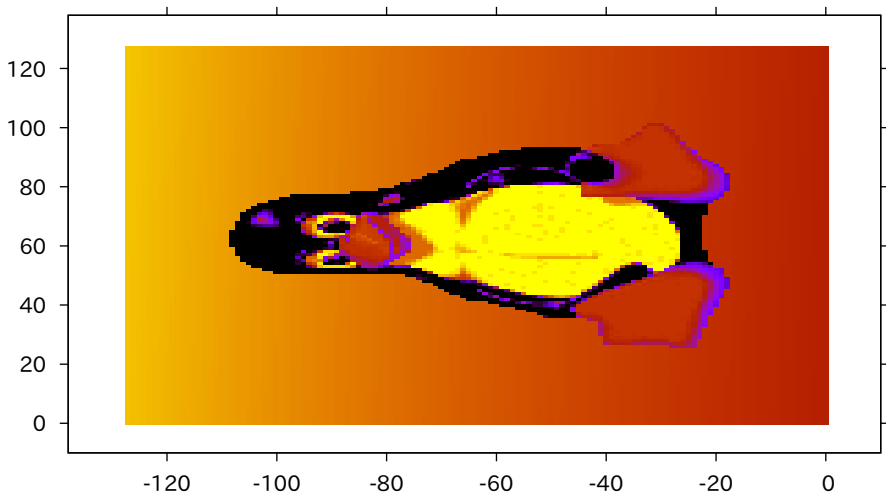


Polygons used to draw pixels for rotated images
Notice the slower refresh rate than for the next plot

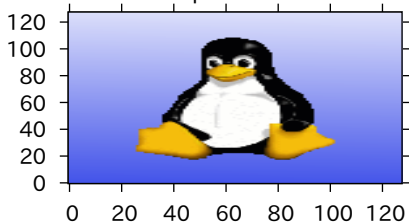
ipy rotation=45d center=(63.5,63.5) format='%uchar' using (\$1+\$2+\$3)/3



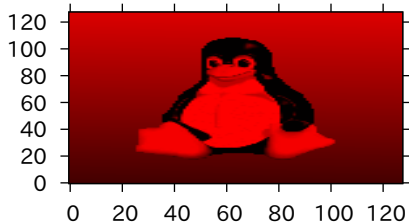
Terminal image routine used to draw plot rotated about origin
Notice the faster refresh rate than for the previous plot



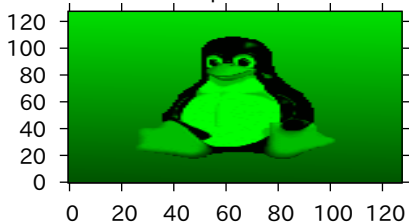
Selection of the input channels via `using`
"I do impersonations..."



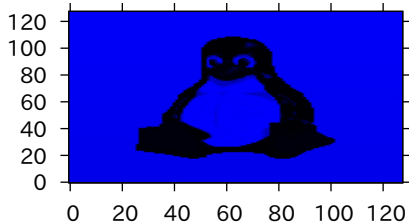
"A cardinal."



"A parrot."

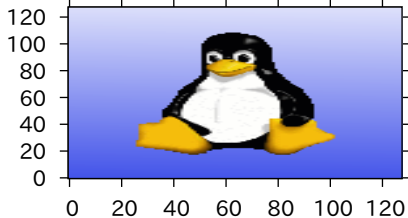


"A bluebird."

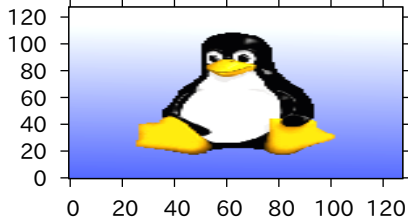


Adjust color balance in the using spec

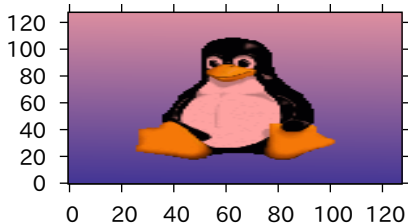
Lake Mendota, "or Wonk-sheck-ho-mik-la!"



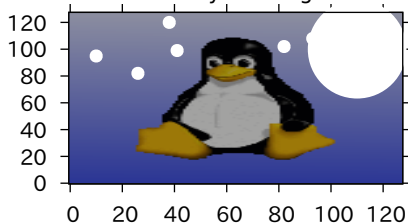
"Lucky I brought sunscreen."



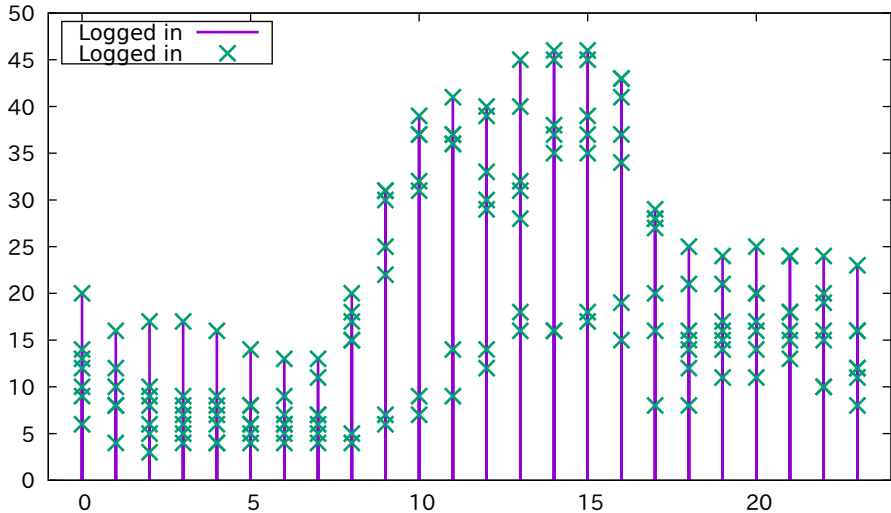
Sunset on the Terrace



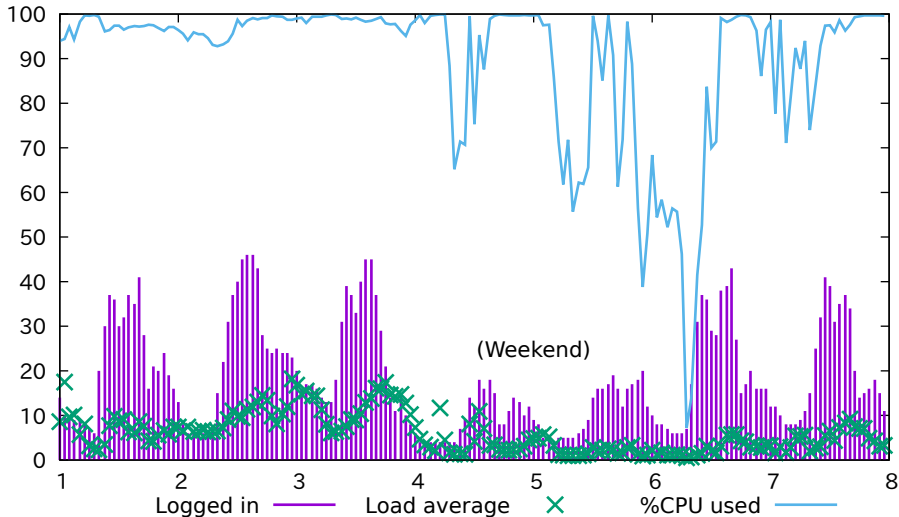
Sultry evening



Convex November 1-7 1989 Circadian

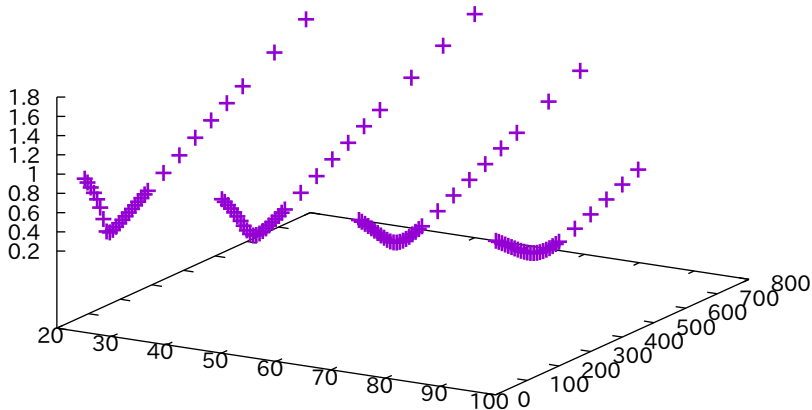


Convex November 1-7 1989



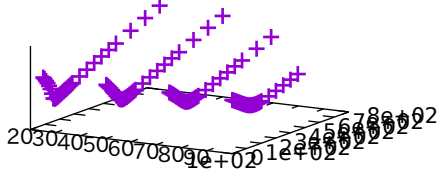
2d binary data example where record length is part of command
'scatter2.bin' binary endian=little record=30:30:29:26 using 1:2:3

+

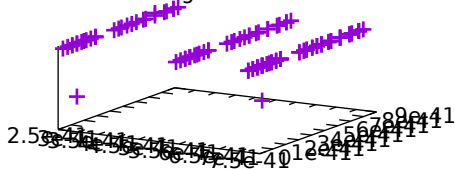


If plots in columns match, your compiler is little endian

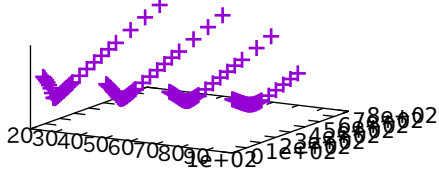
Little endian



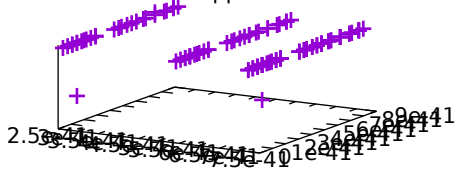
Big endian



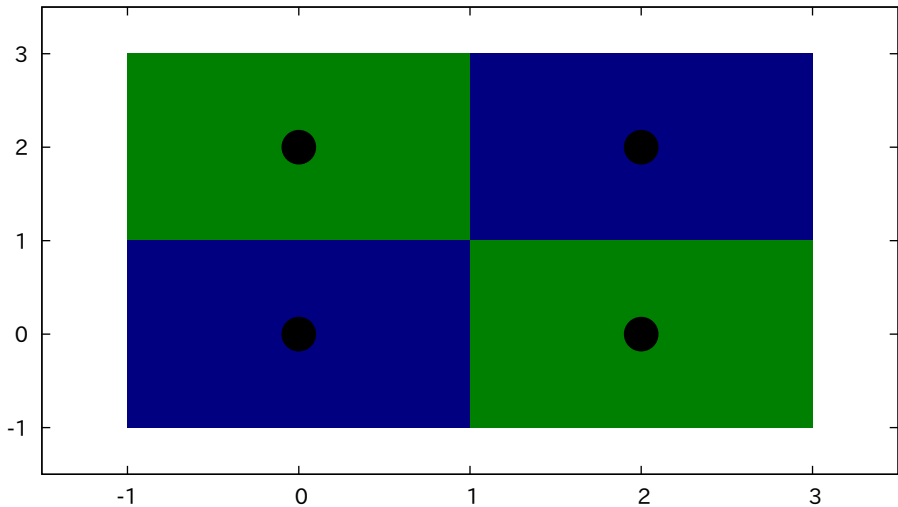
Default



Swapped

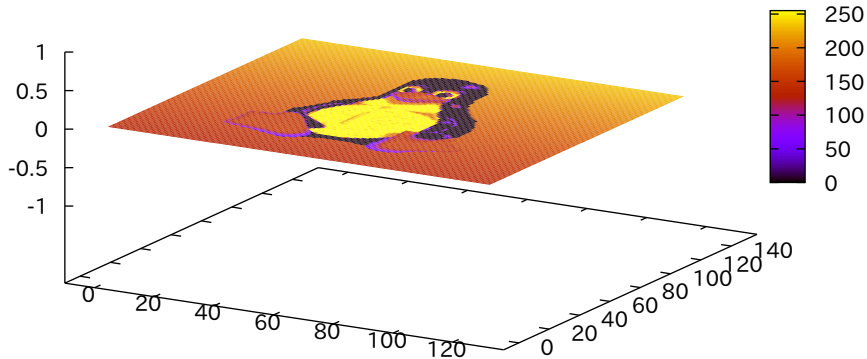


Close up of pixels having grid points $(0,0)$, $(0,2)$, $(2,0)$ and $(2,2)$



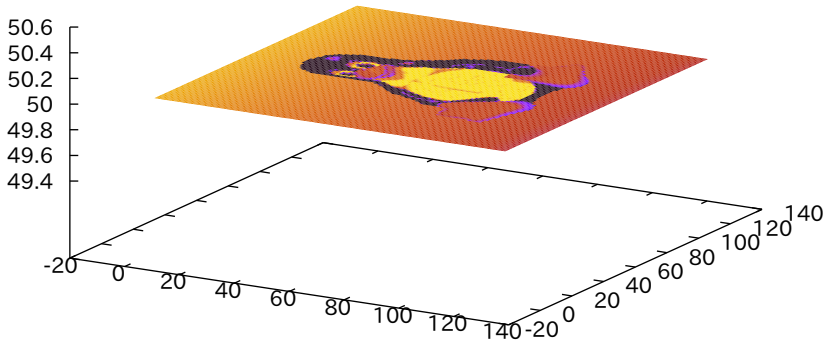
Simple extension of a two dimensional image into three dimensions

array=(128,128) flip=y format='%uchar%uchar%uchar' using (\$1+\$2+\$3)/3



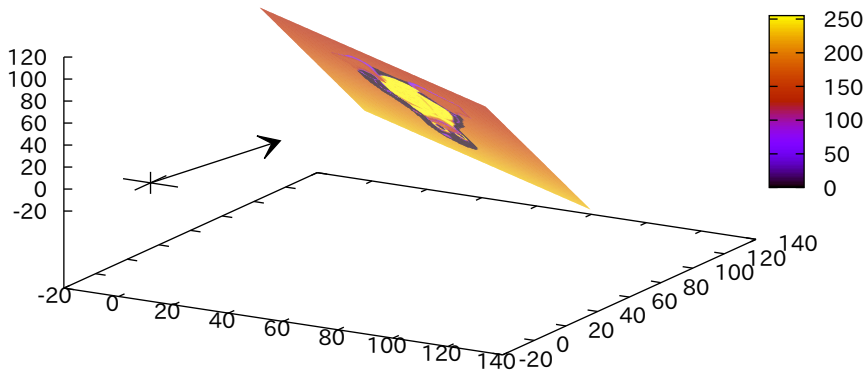
Orientation operations from 'plot' also apply to 'splot'

d center = (63.5,63.5,50) format='%uchar%uchar%uchar' using (\$1+\$2+\$3)



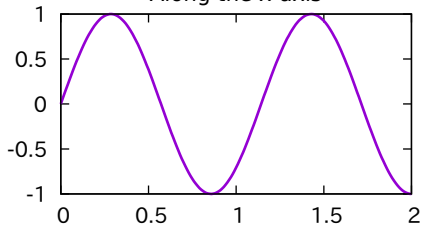
The key word 'perpendicular' applies only to 'splot'

5,63.5,50) perp=(1,1,1) format='%uchar%uchar%uchar' using (\$1+\$2+\$3)/3

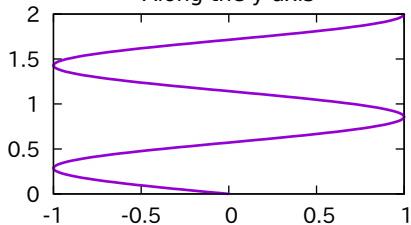


Temporal data having one generated coordinate

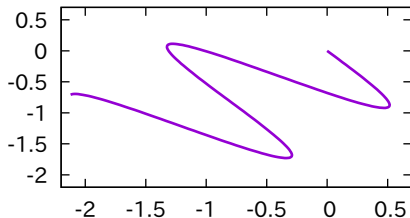
Along the x-axis



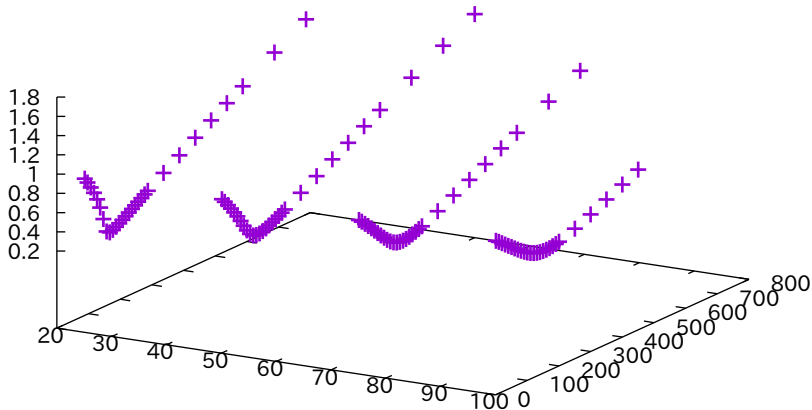
Along the y-axis



Along a 225 degree projection

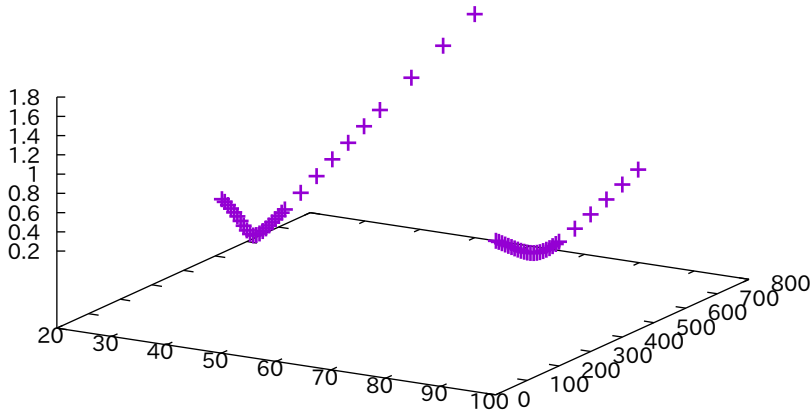


2d binary data example where x coordinate is ignored then generated
29:26 origin=(25,0,0):(50,0,0):(75,0,0):(100,0,0) format='%f%f' using (0):2:3 +



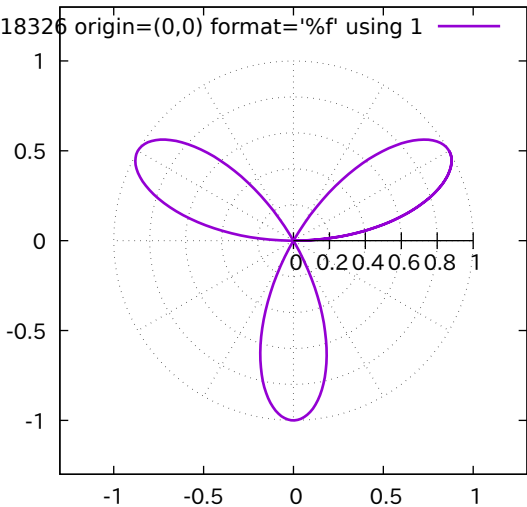
The key word 'skip' used to ignore some data

d=30:26 skip=360:348 origin=(50,0,0):(100,0,0) format='%f%f' using (0):2:3 +



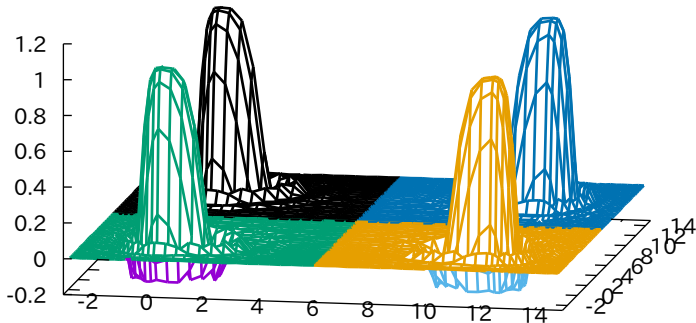
Uniform sampling in the polar coordinate system

e array=201 dt=0.018326 origin=(0,0) format='%f' using 1



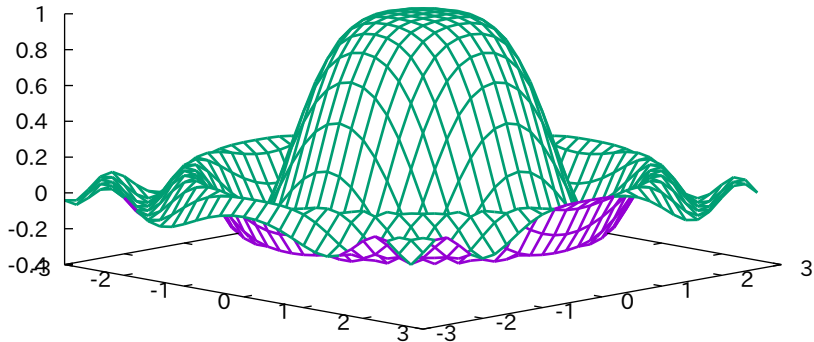
Matrix binary data (gnuplot binary) translated

"binary3" binary center=(1.5,1.5,0) — purple
"binary3" binary center=(10.5,1.5,0) rotate=0.5pi u 1:2:3 — light blue
"binary3" binary center=(10.5,10.5,0) rotate=1.0pi u 1:2:3 — yellow
"binary3" binary center=(1.5,10.5,0) rotate=1.5pi u 1:2:3 — red



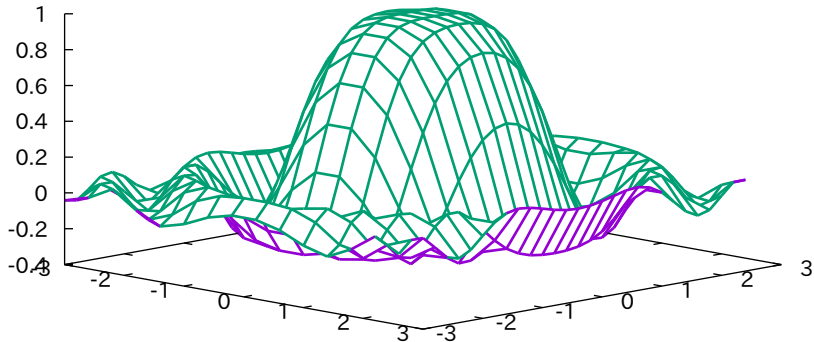
Non-decimated matrix data file

"binary2" binary



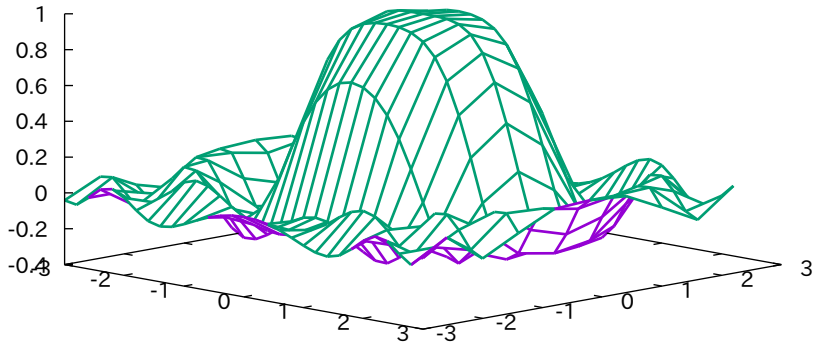
Decimate by two in first dimension

"binary2" binary every 2



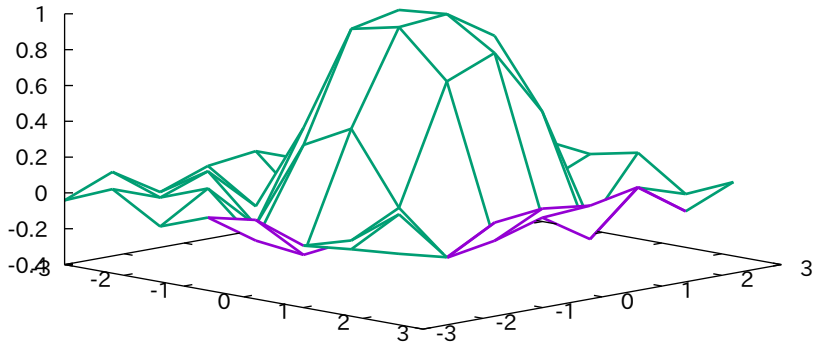
Decimate by three in second dimension

"binary2" binary every :3



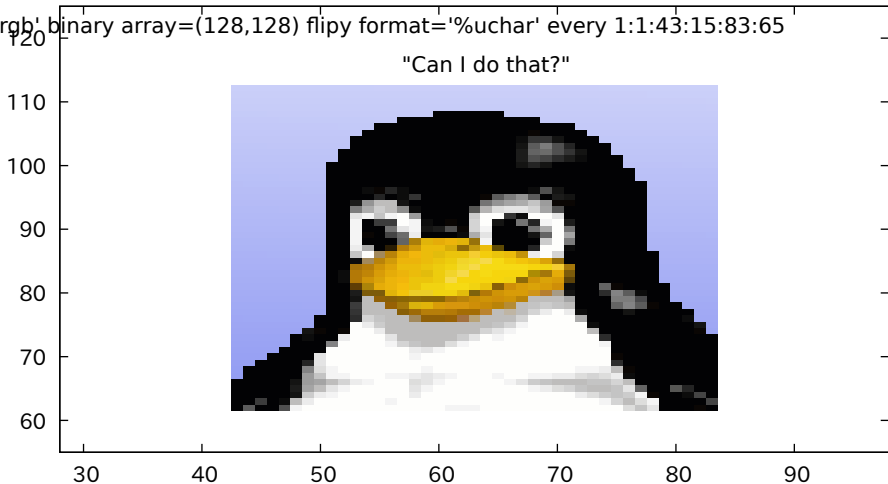
Decimate by four in both dimensions

"binary2" binary every 4:4

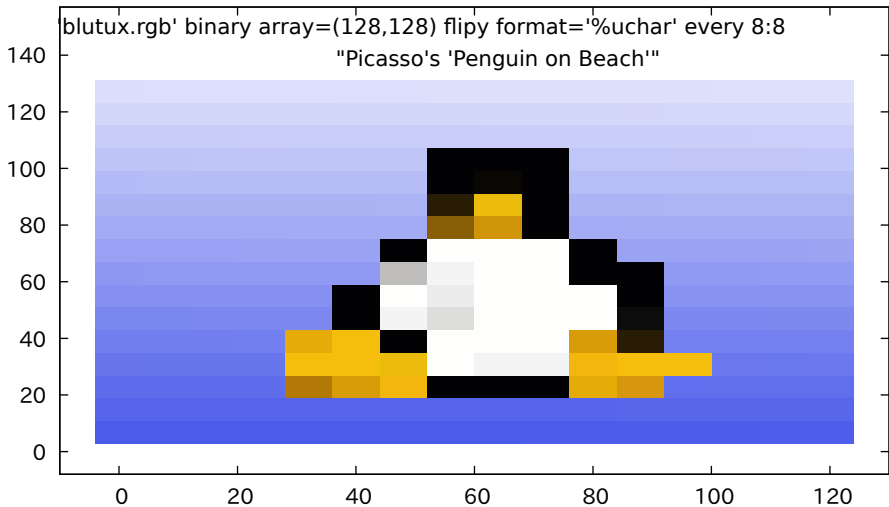


Decimation works on general binary data files as well.
Let Tux have his fun...

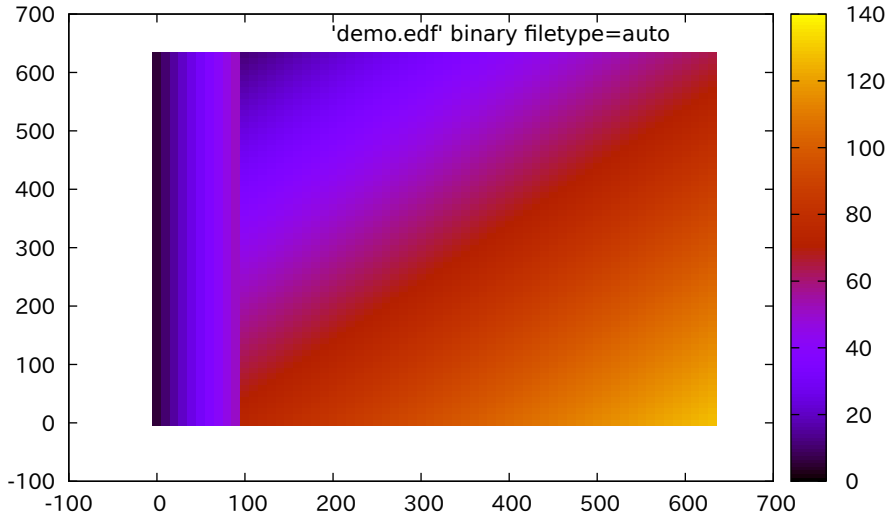
ux.rgb' binary array=(128,128) flipy format='%uchar' every 1:1:43:15:83:65



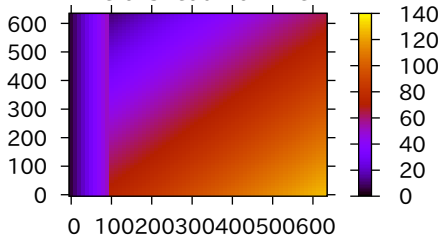
... Sure, go ahead.



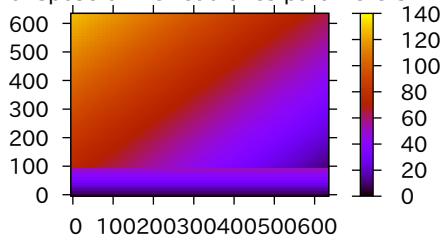
Automatically recognizing file type and extracting file information



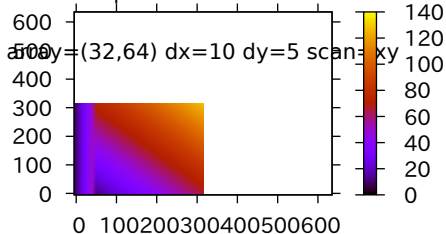
Details read from file



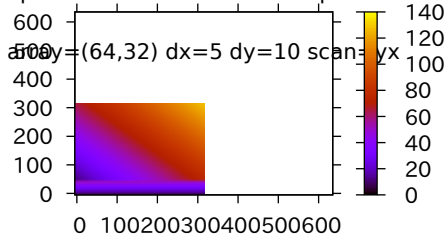
Transpose of file-read axes parameters



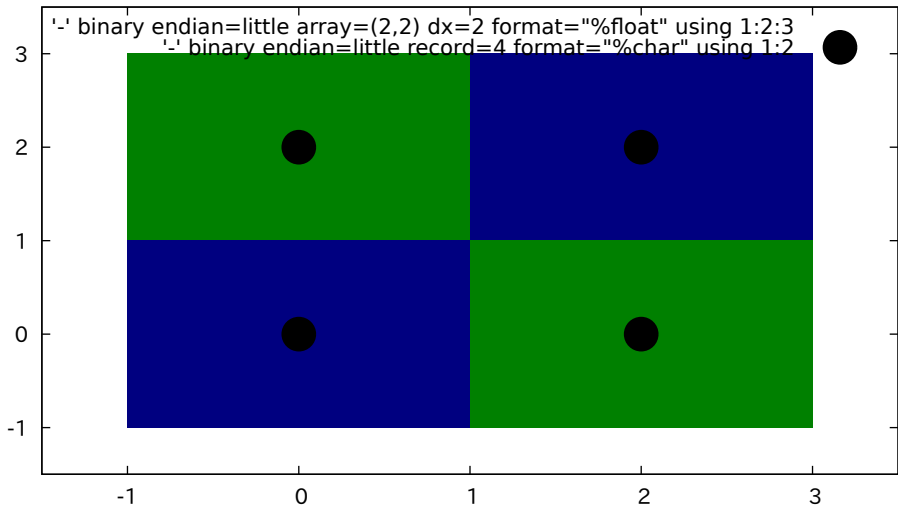
Details specified at command line



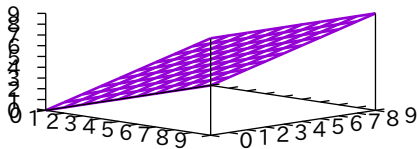
Transpose of command line axes parameters



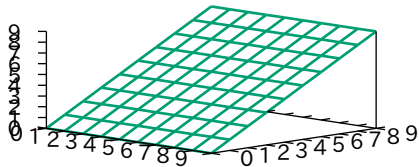
Binary data specified at the command line, intended for use through pipe



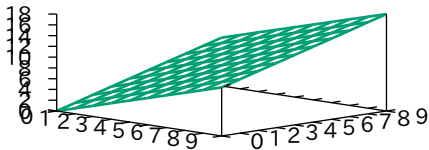
'asciimat.dat' matrix index 0



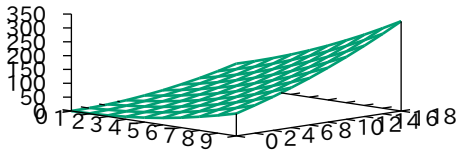
'asciimat.dat' matrix index 1



'asciimat.dat' matrix index 2

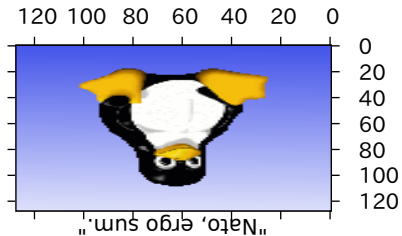
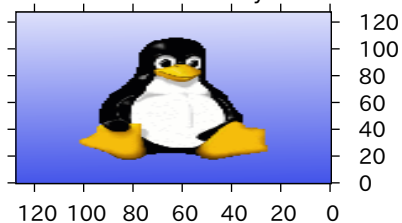


'asciimat.dat' matrix index 2 using $1:(2*\$2):(\$3*$

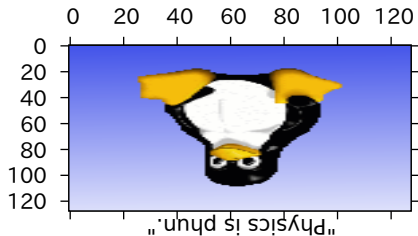
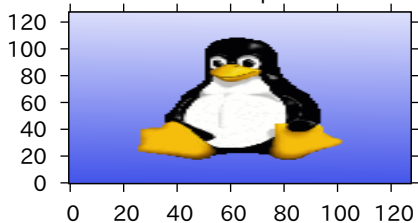


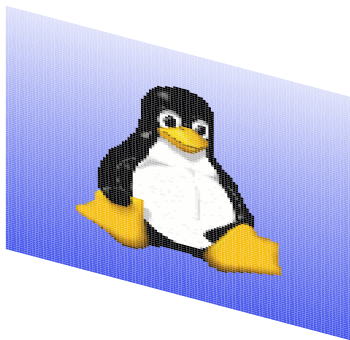
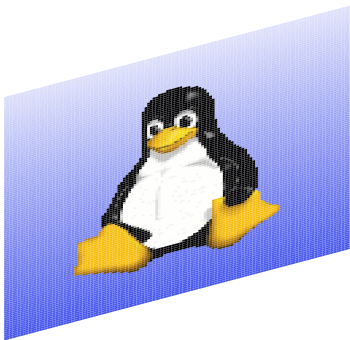
Images reverse according to axis orientation

"Eccentric coordinate systems"

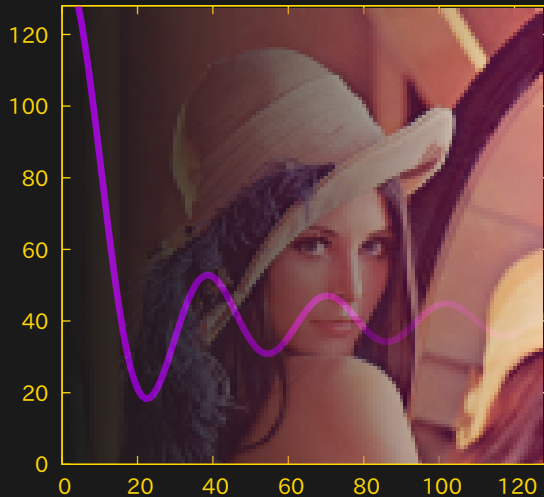


"Cartesian plane!"

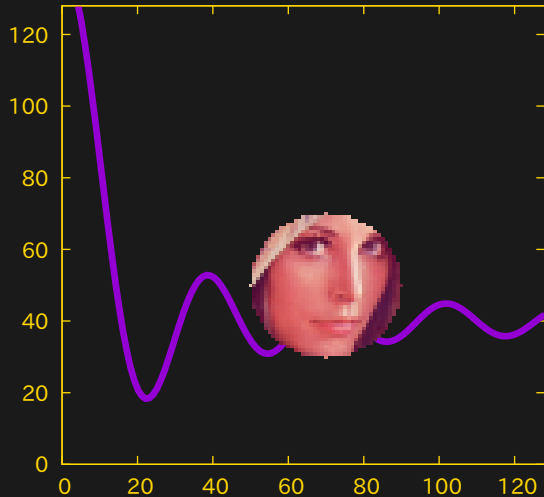




Tux in a reflective mood

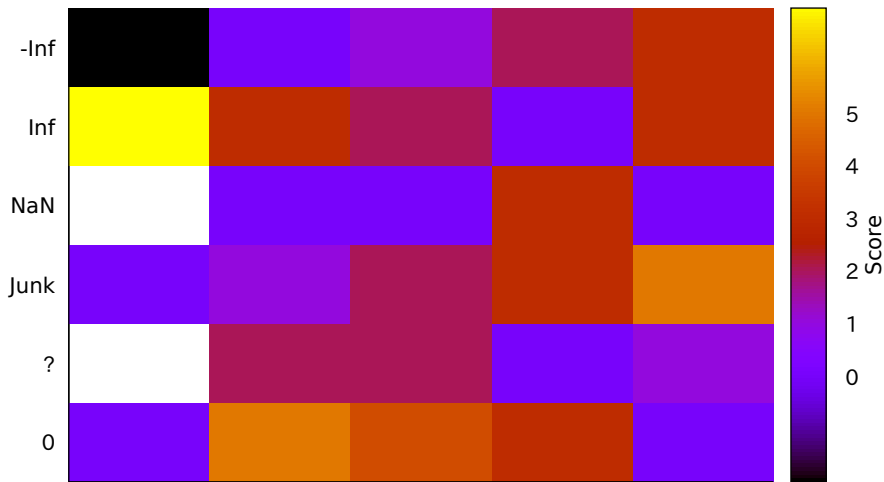


Plot style rgba
solid line
Lena with linear
alpha gradient



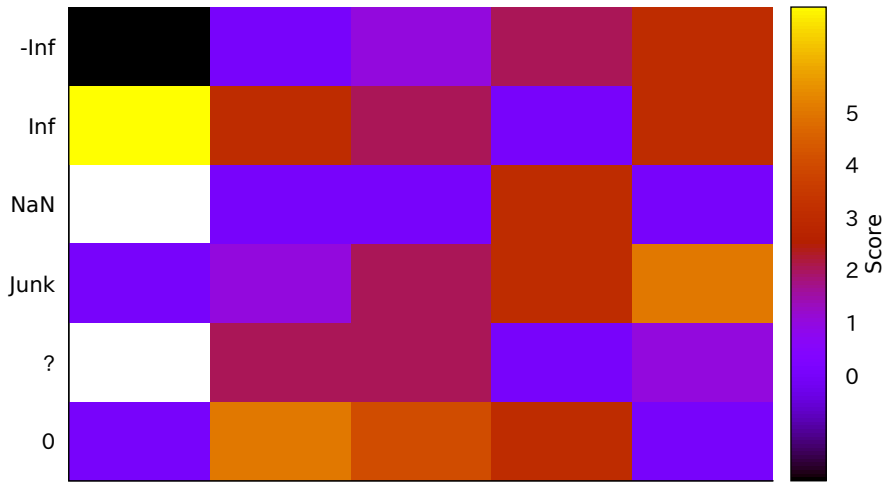
Plot style rgba
solid line
Lena with circular mas

Treatment of missing/undefined/NaN/Inf data



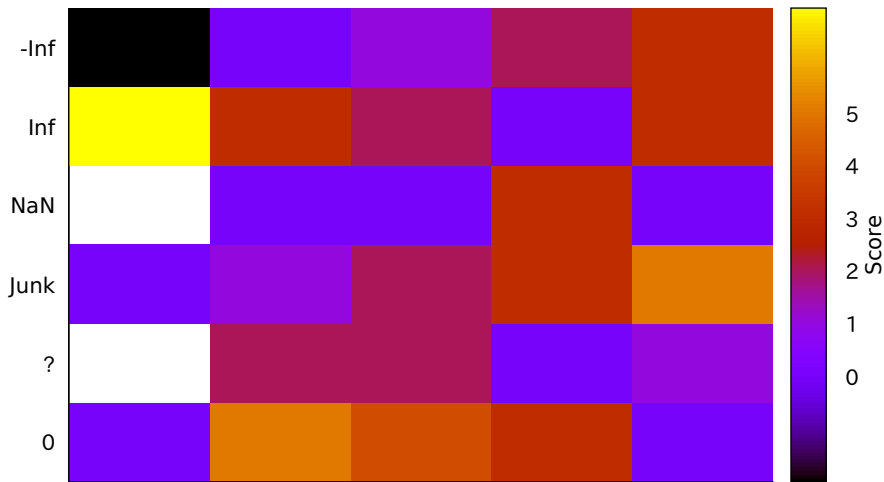
First column contains various odd values

Same thing in 'pixels' mode (2D)



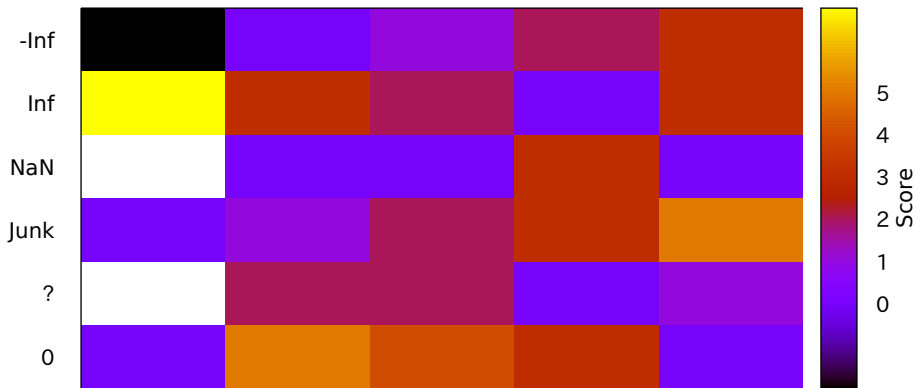
First column contains various odd values

Same thing passing data value through 'using 1:2:(\$3)'



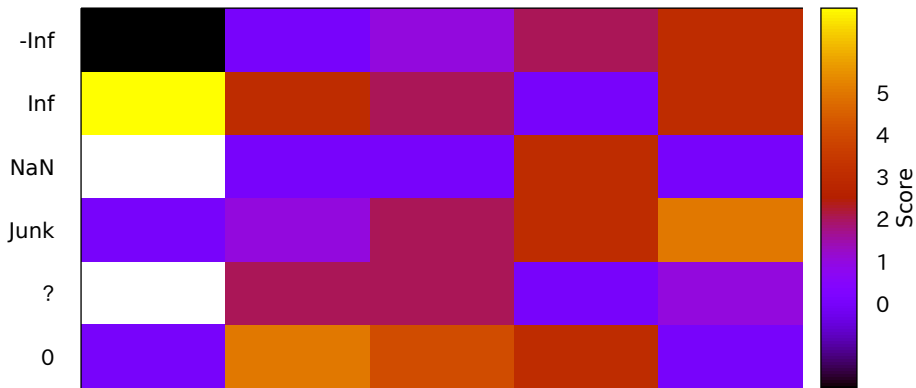
First column contains various odd values

Same thing in 3D mode



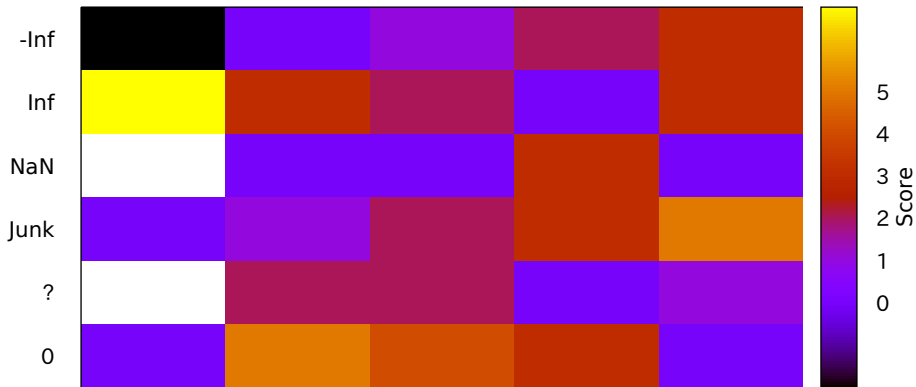
First column contains various odd values

Same thing in 'pixels' mode (3D)



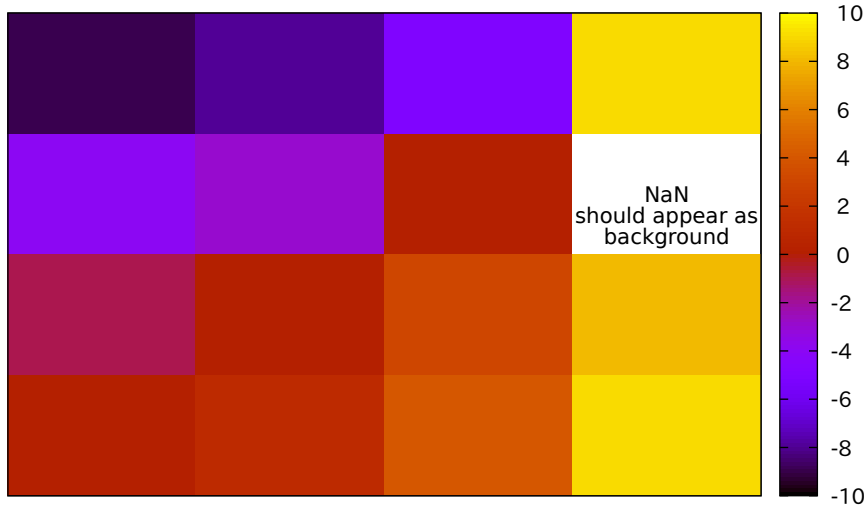
First column contains various odd values

3D image with pixel value in 4th column

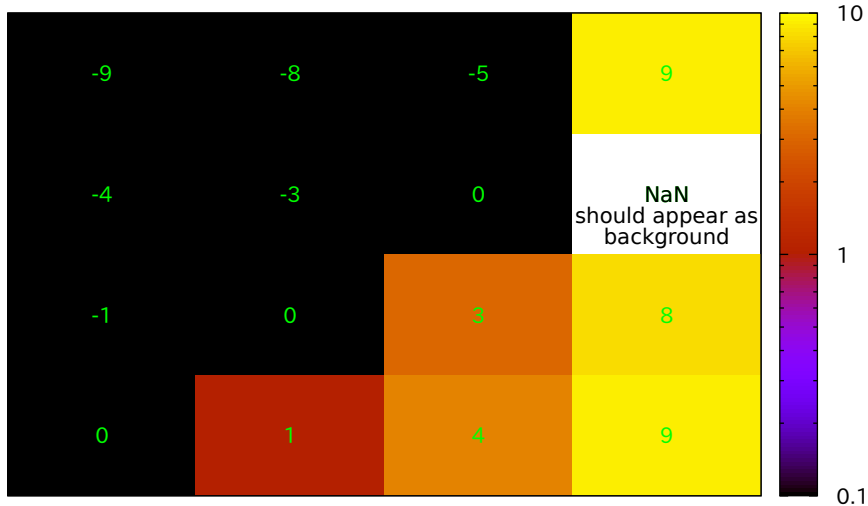


First column contains various odd values

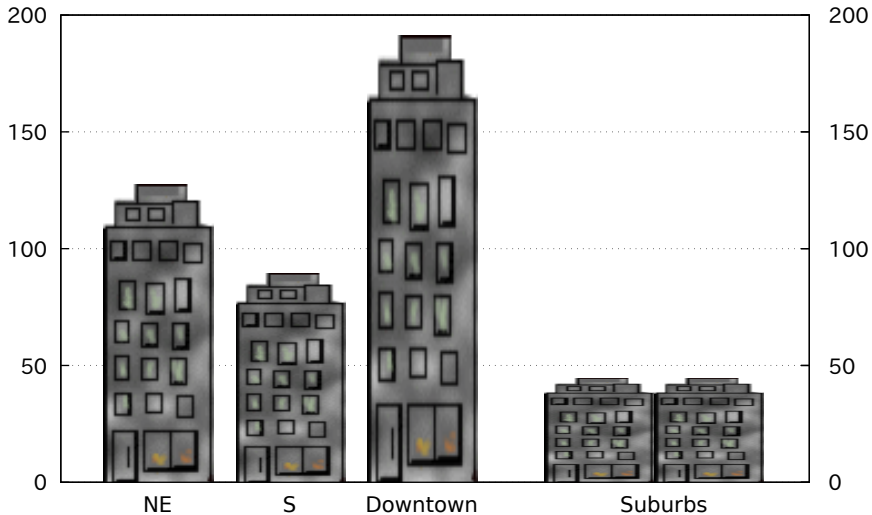
image from non-matrix data



negative values mapped to log-scale colorbar



Building Code Height Limits



Exercise substring handling

```
beg = 2 end = 4
foo      = ABCDEF
foo[3:5]  = CDE
foo[1:1]  = A
foo[5:3]  =
foo[beg:end] = BCD
foo[end:beg] =
foo[5:]   = EF
foo[5:*]  = EF
foo[:]    = ABCDEF
foo[*:*]  = ABCDEF
foo.foo[2:2] = ABCDEFB
(foo.foo)[2:2] = B
```

```
foo[1:1] eq 'A' foo[2:2] ne 'X' = true
```

Exercise string handling functions

```
foo      = ABCDEF  
strlen(foo) = 6  
substr(foo,3,4) = CD
```

```
haystack  = `date`  
haystack  = 2020年 1月21日 火曜日 16時10分58秒 JST  
needle    = :  
S = strstr(haystack,needle) = 0  
haystack[S-2:S+2] = 20  
It is now 20
```

```
words(haystack) = 5  
word(haystack,5) = JST
```

sprintf output of long strings works OK

Exercise word and words functions

foo = word and words can handle 'quoted string'

words(foo) = 6

word(foo, 6) = quoted string

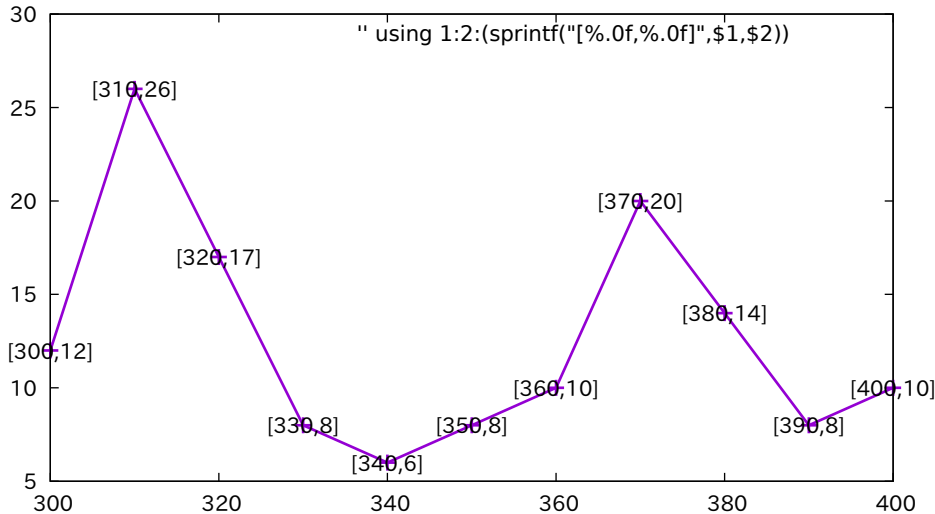
foo = "double quotes" or 'single quotes'

words(foo) = 3

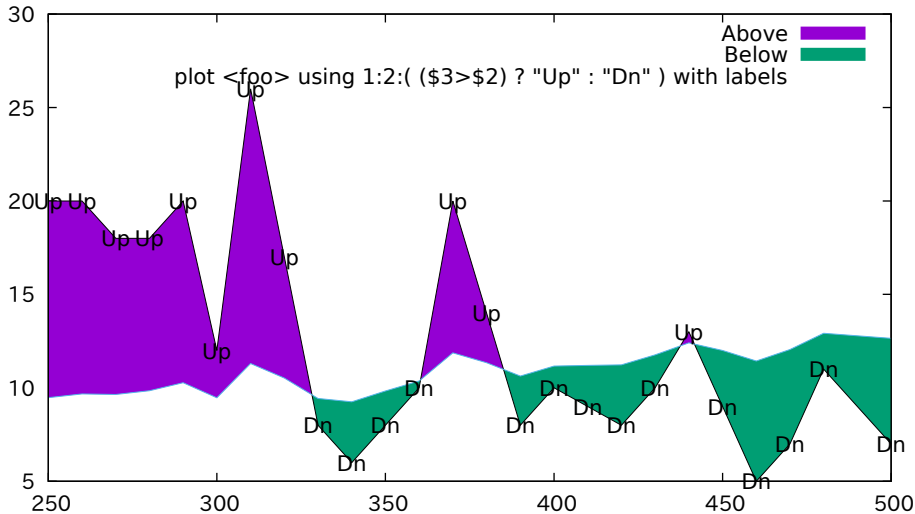
foo = Apostrophes inside words don't matter

word(foo, 4) = don't

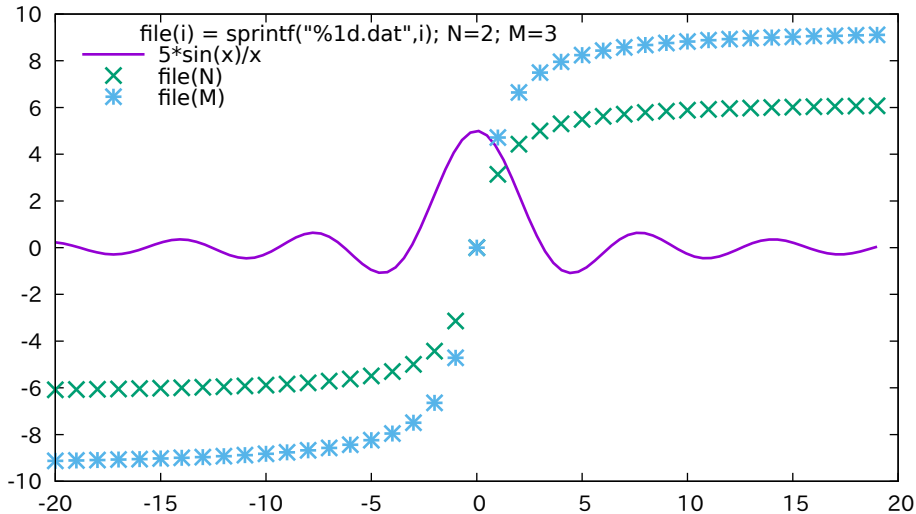
String-valued expression in using spec

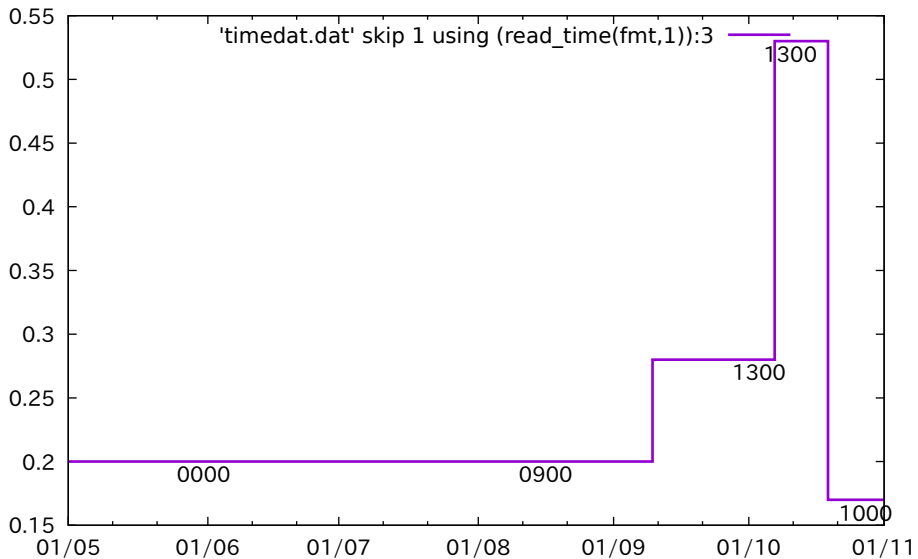


Constant string expressions as plot symbols

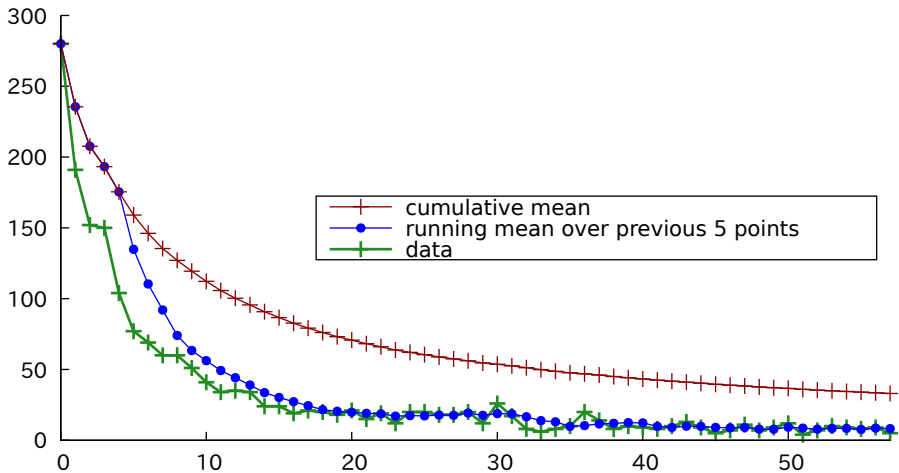


String-valued functions to generate datafile names

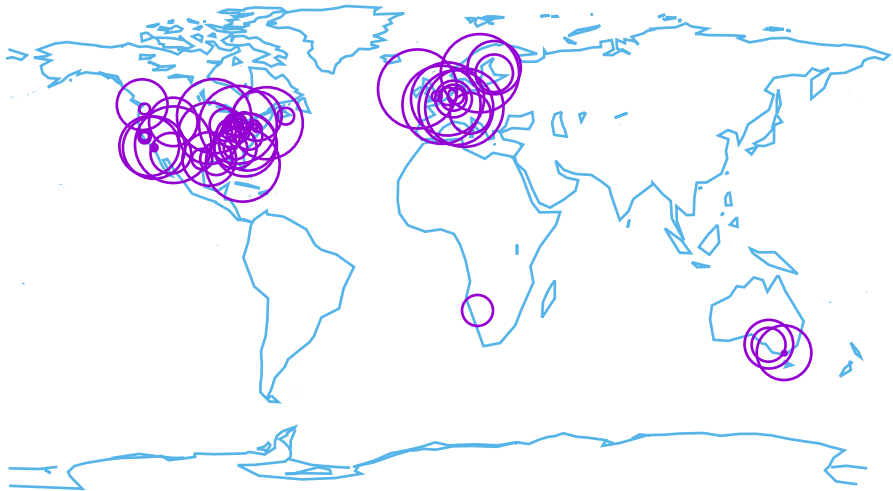




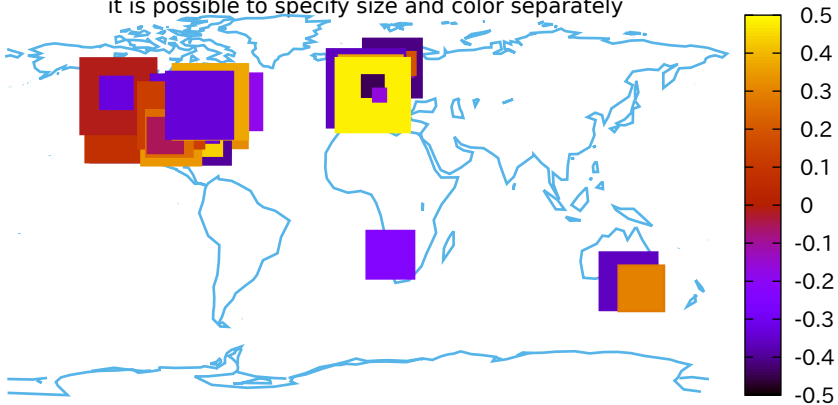
Demonstrate use of assignment and serial evaluation operators to accumulate statistics as successive data lines are read in



plot with variable size points



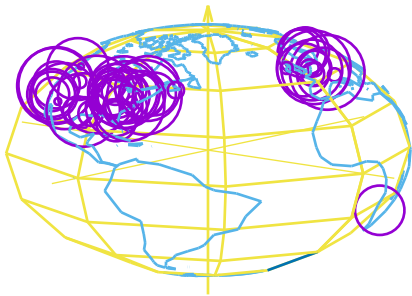
plot with variable size points
it is possible to specify size and color separately



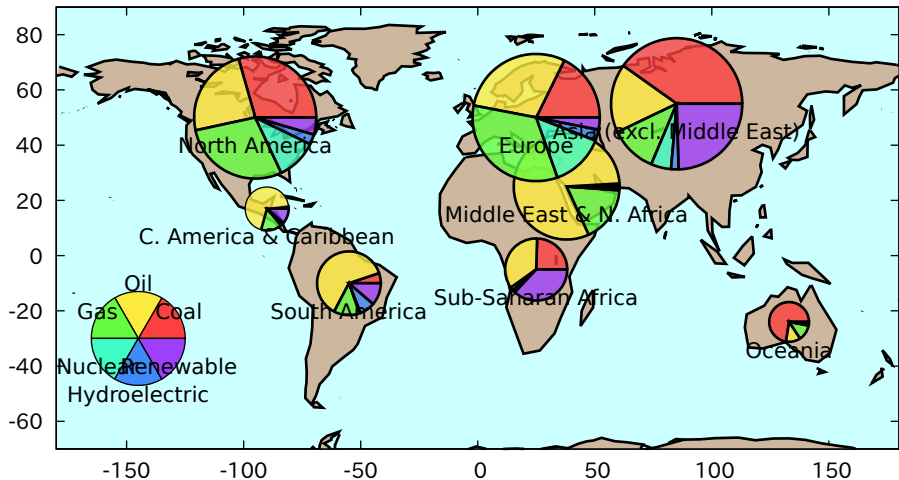
3D version using spherical coordinate system



3D solid version through hiddenlining

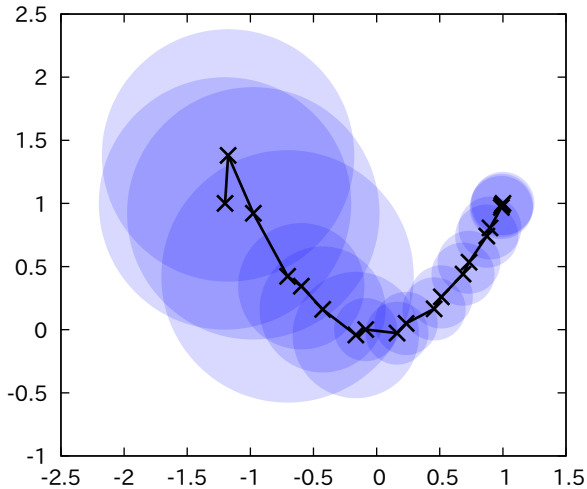


Sources of energy production, plotted for each continent



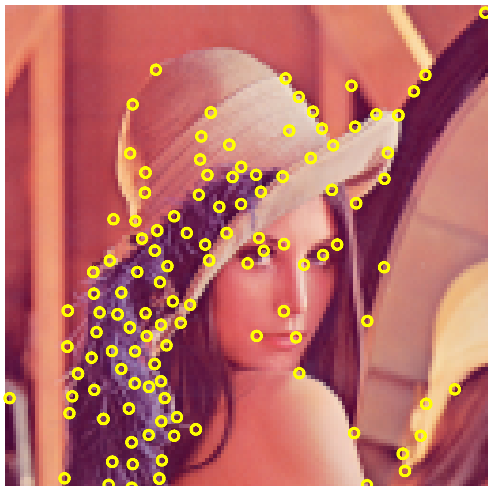
Dynamically generated pie charts

Trace of unconstrained optimization with trust-region method

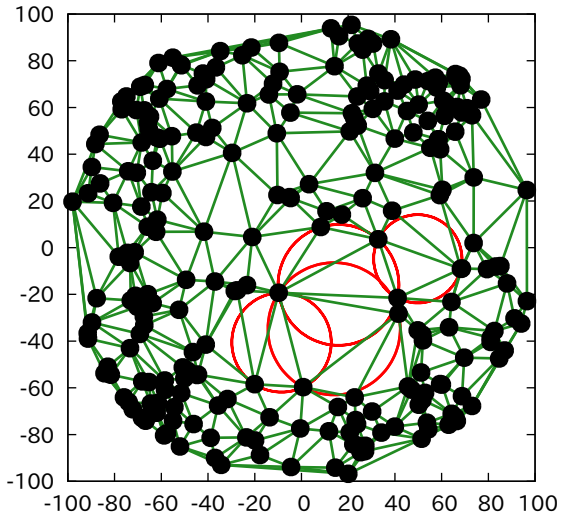


Note that overlapping transparent circles produce a darker area

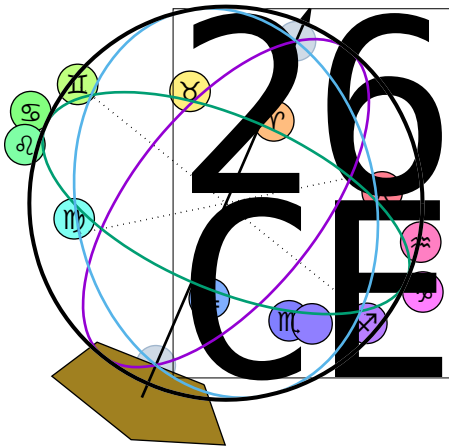
Lena's key points



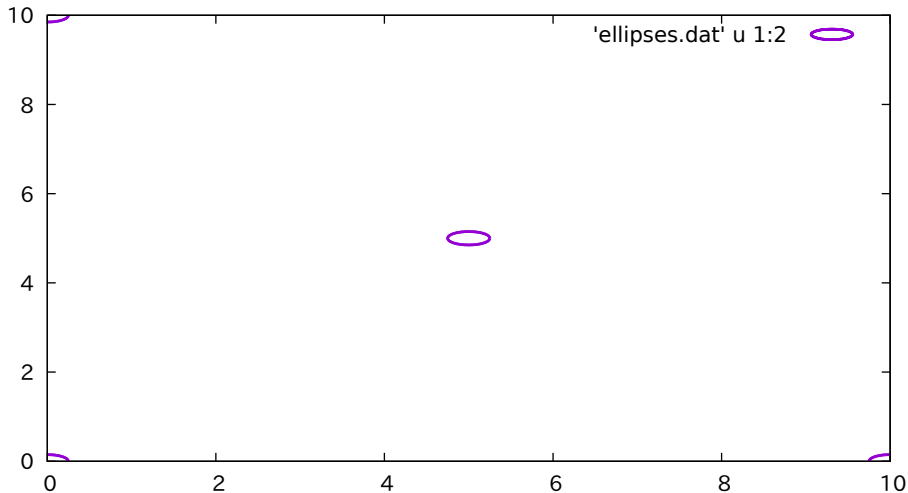
Delaunay triangulation of Hemisphere points, some empty circles in red



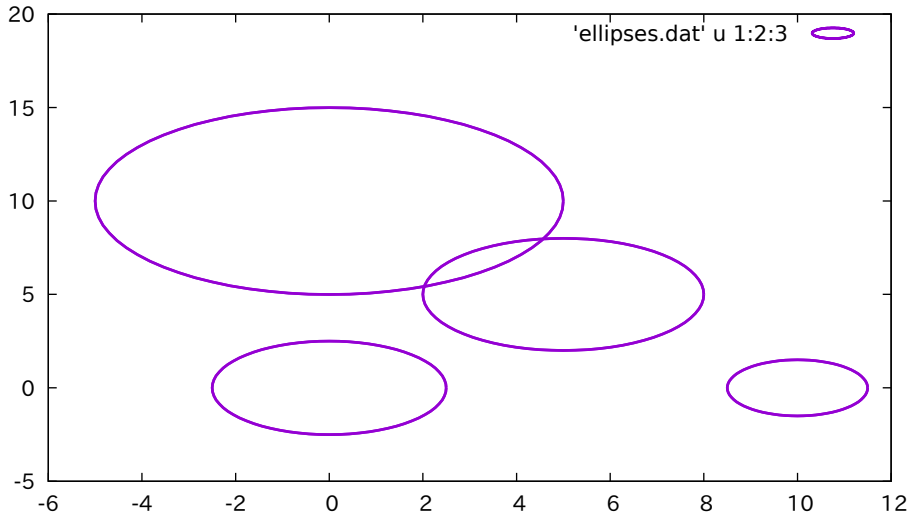
Circle and polygon objects in 3D



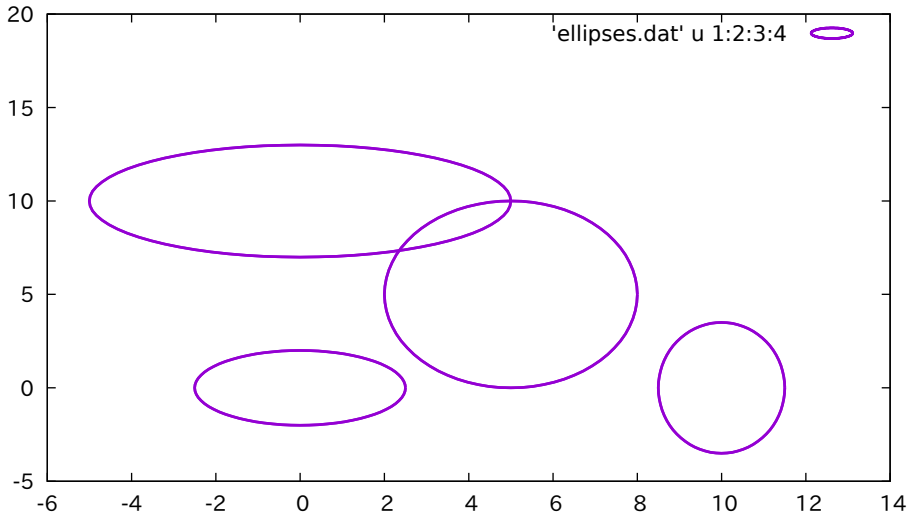
Demonstration of the 'ellipses' plotting style
Two-column form: x y (default size)



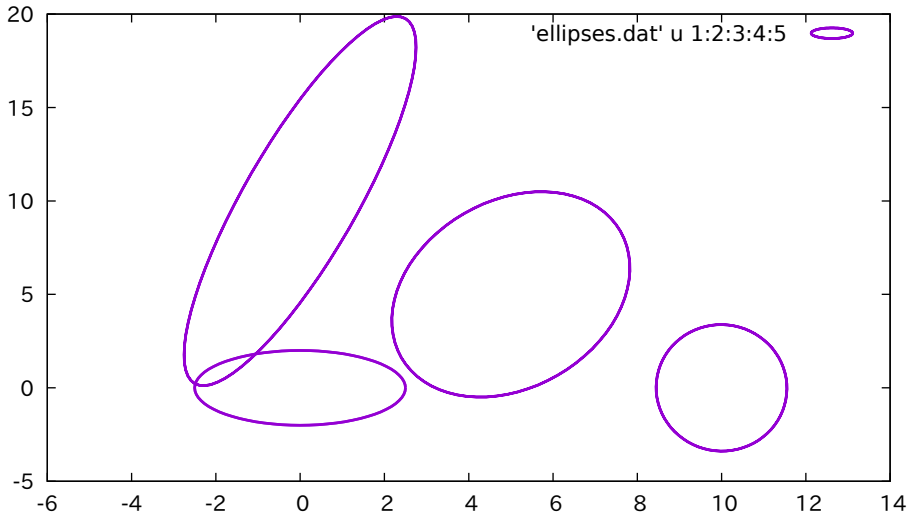
Three-column form: x y major_diameter (minor diameter is the same)



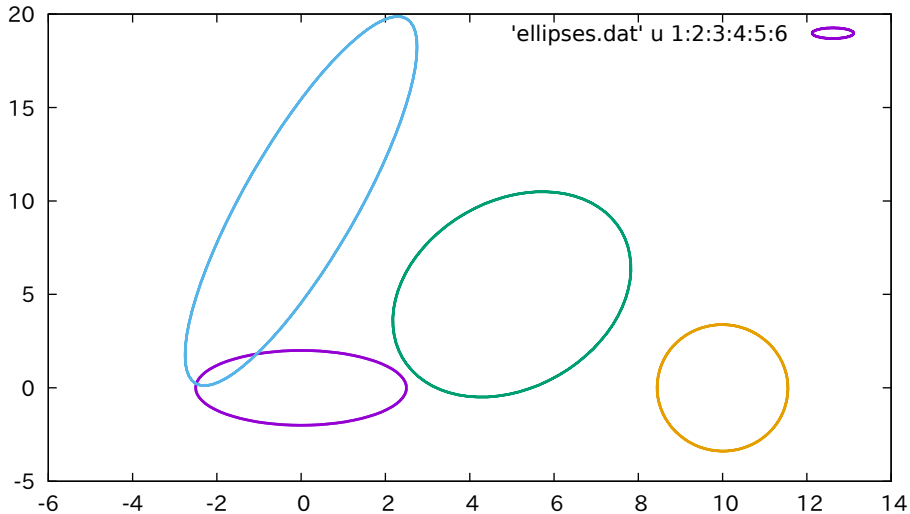
Four-column form: x y major_diameter minor_diameter



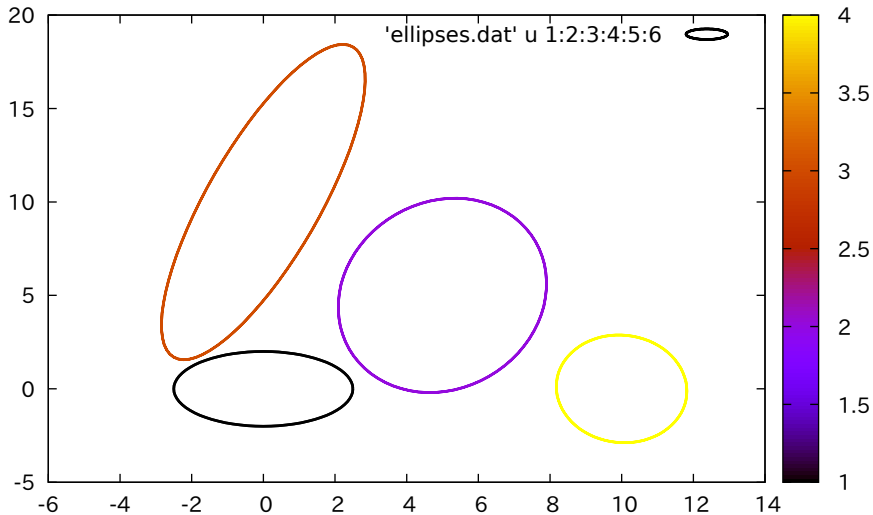
Five-column form: x y major_diameter minor_diameter angle



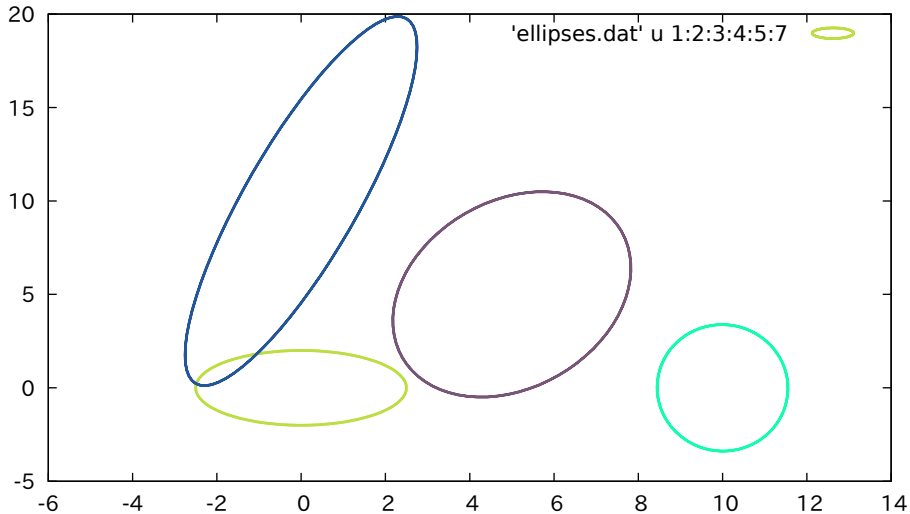
Six-column form: 6th column variable color (lc variable)



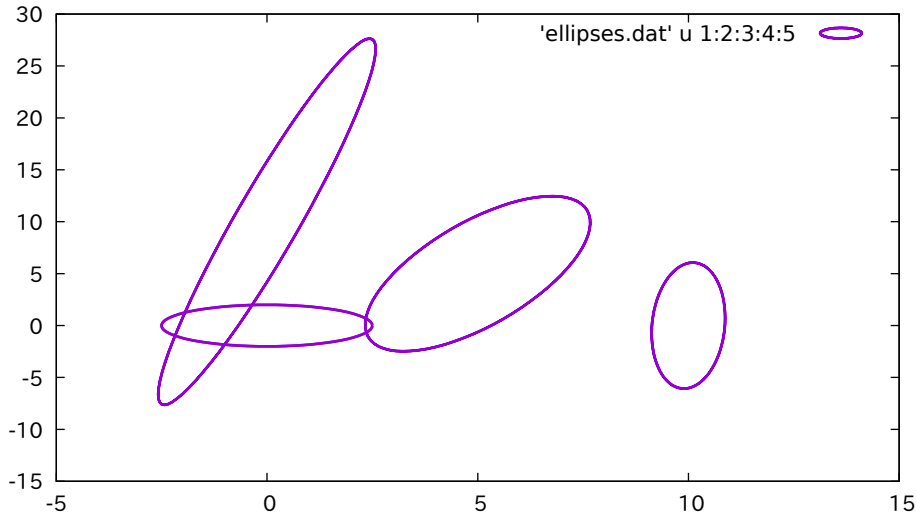
Six-column form: 6th column variable color (lc palette)



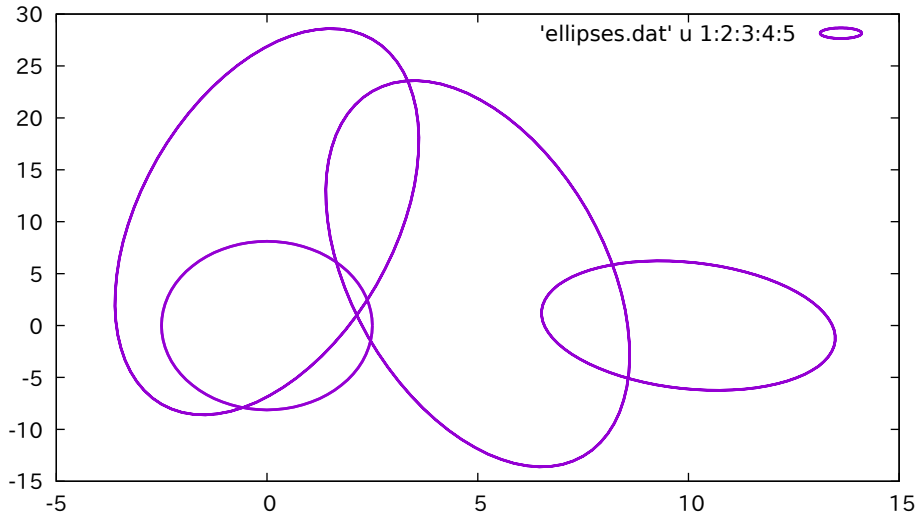
Six-column form: 6th column variable color (lc rgb variable)



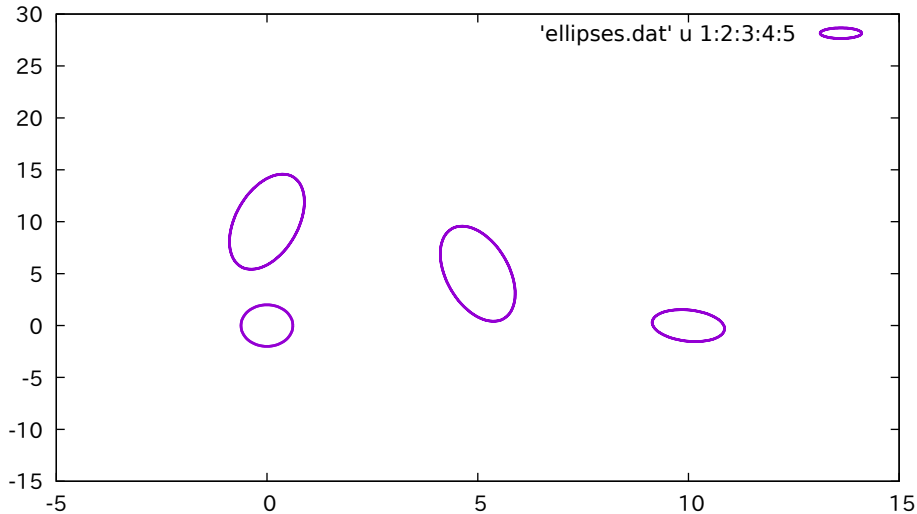
Scaling of axes: units xy



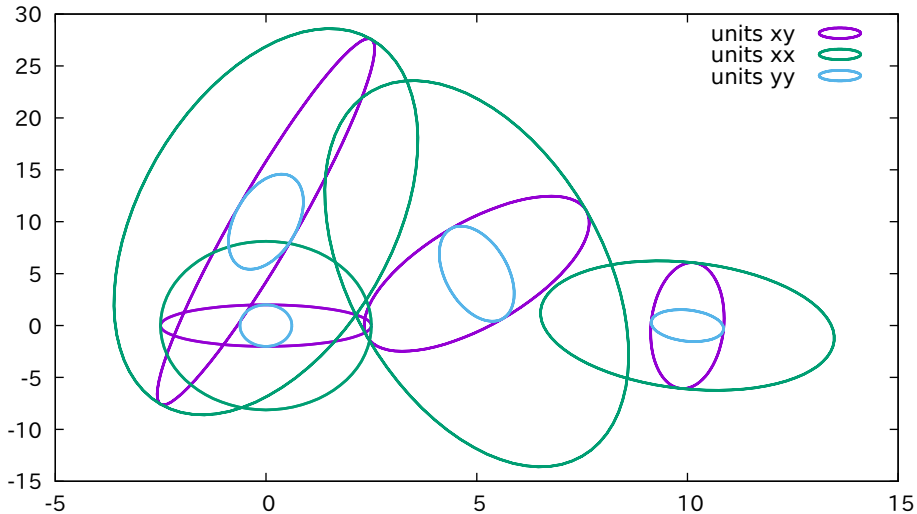
Scaling of axes: units xx



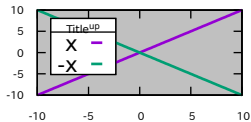
Scaling of axes: units yy



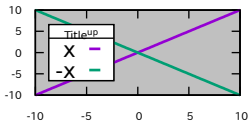
Now see all three together



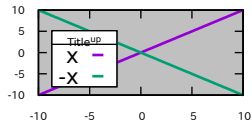
Key (ins vert left top)



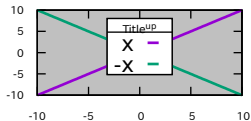
Key (ins vert center left)



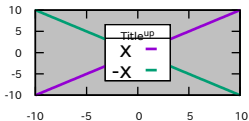
Key (ins vert bot left)



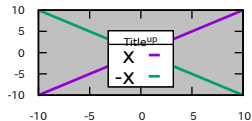
Key (ins vert center top)



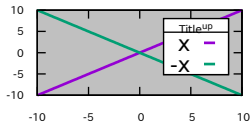
Key (inside vertical center)



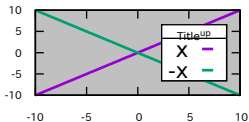
Key (ins vert bot center)



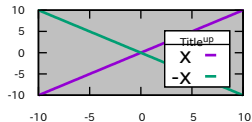
Key (ins vert right top)



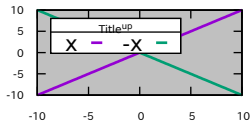
Key (ins vert cent right)



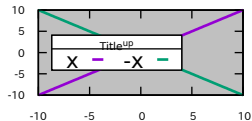
Key (ins vert bot right)



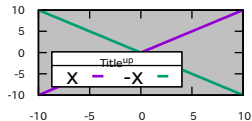
Key (ins horiz left top)



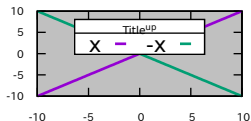
Key (ins horiz center left)



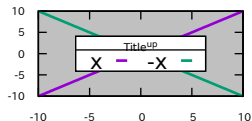
Key (ins horiz bot left)



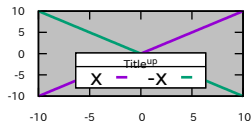
Key (ins horiz center top)



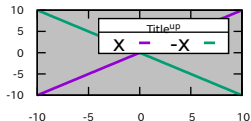
Key (inside horizontal center)



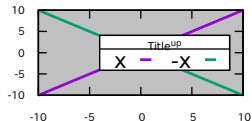
Key (ins horiz bot center)



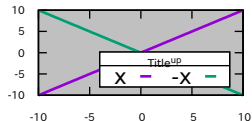
Key (ins horiz right top)

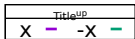


Key (ins horiz cent right)

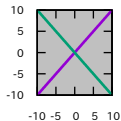
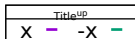
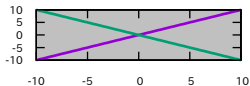


Key (ins horiz bot right)

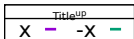
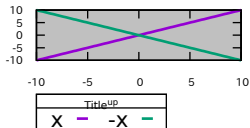




Key (out horiz left top)

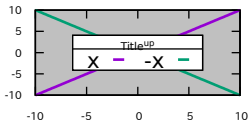
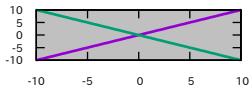


Key (out horiz center left)Key (out horiz bot left)

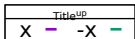
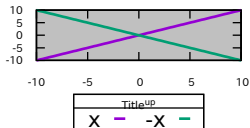


Key (outside horizontal center)

Key (out horiz center top)

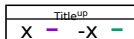
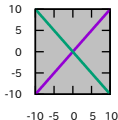
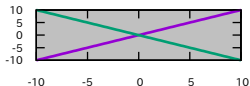


Key (out horiz bot center)

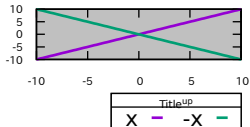


Key (out horiz cent right)

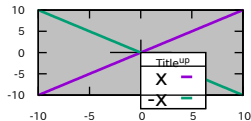
Key (out horiz right top)



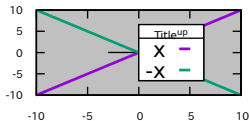
Key (out horiz bot right)



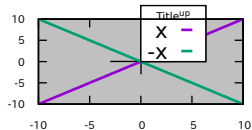
Key (<manual> vert left top)



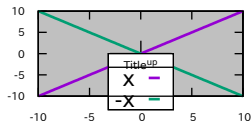
Key (<manual> vert center left)



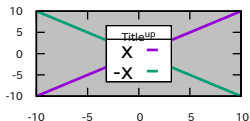
Key (<manual> vert bot left)



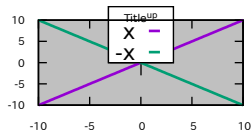
Key (<manual> vert center top)



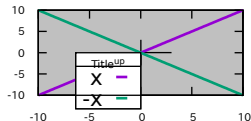
Key (<manual> vertical center)



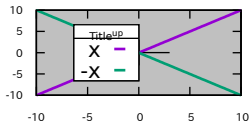
Key (<manual> vert bot center)



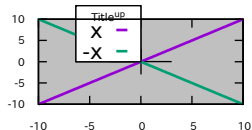
Key (<manual> vert right top)



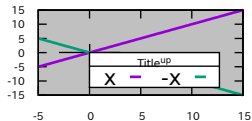
Key (<manual> vert cent right)



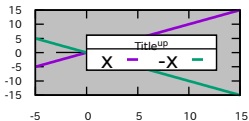
Key (<manual> vert bot right)



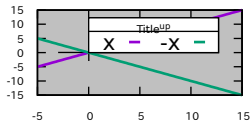
Key (<manual> horiz left top)



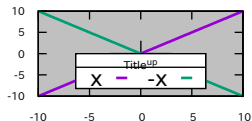
Key (<manual> horiz center left)



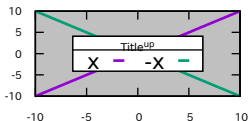
Key (<manual> horiz bot left)



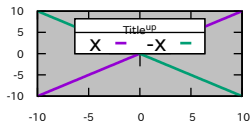
Key (<manual> horiz center top)



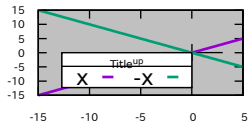
Key (<manual> horizontal center)



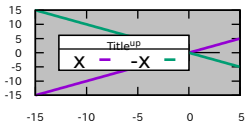
Key (<manual> horiz bot center)



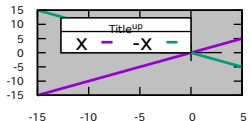
Key (<manual> horiz right top)



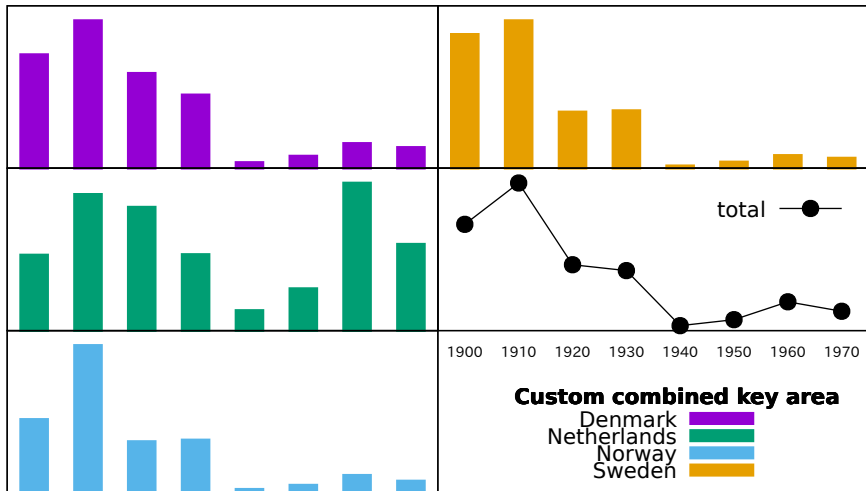
Key (<manual> horiz cent right)



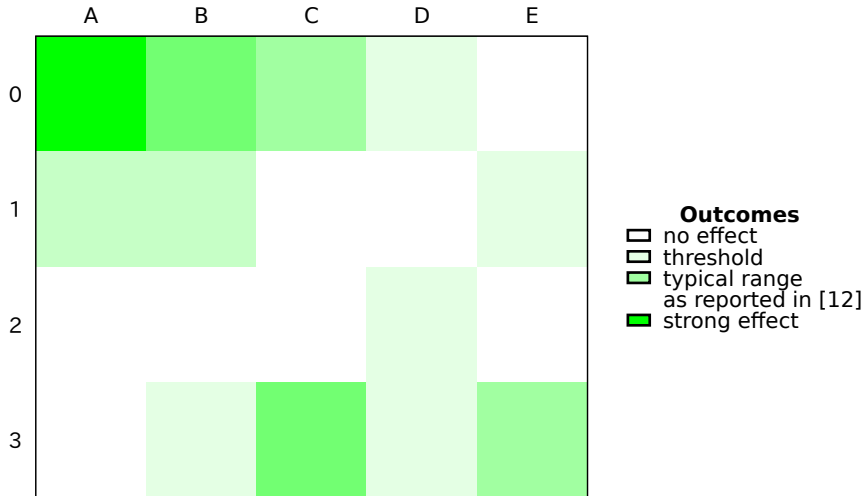
Key (<manual> horiz bot right)



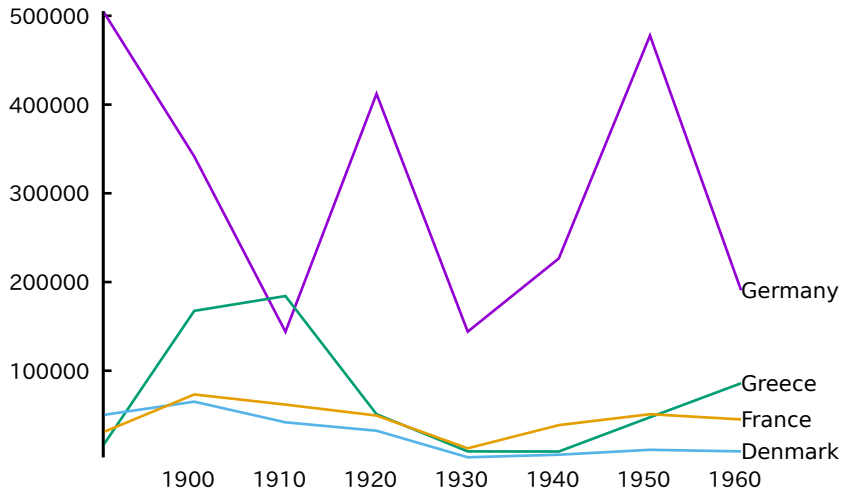
Illustrate use of a custom key area

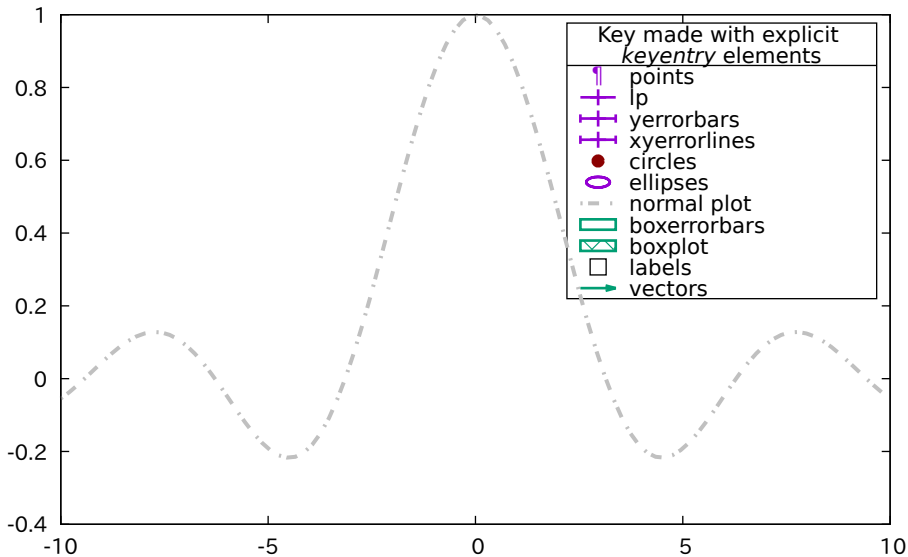


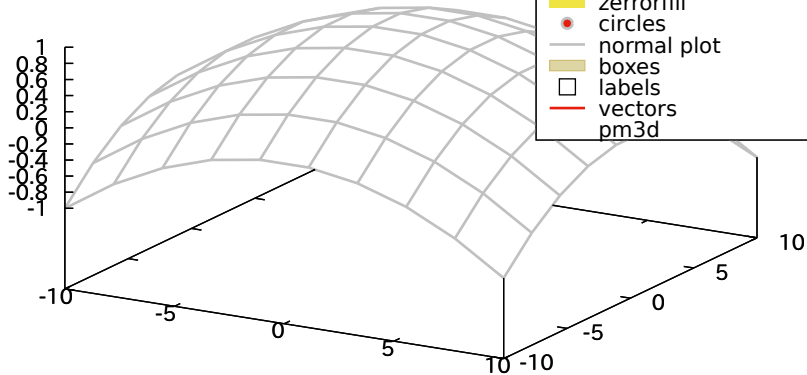
Construct key from custom entries



Position plot titles at the end of the corresponding curve
rather than in a separate key



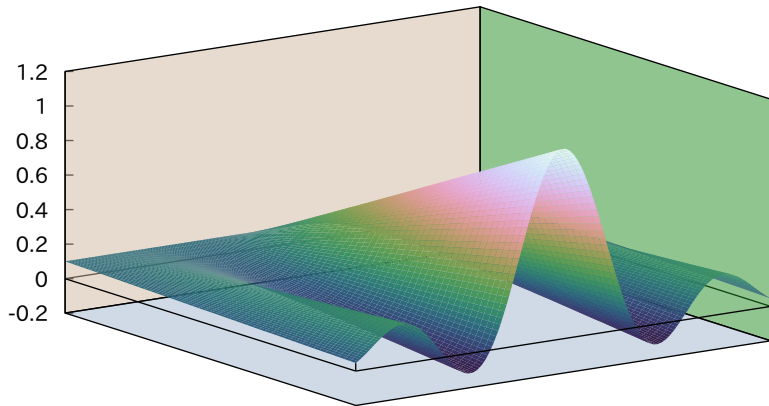




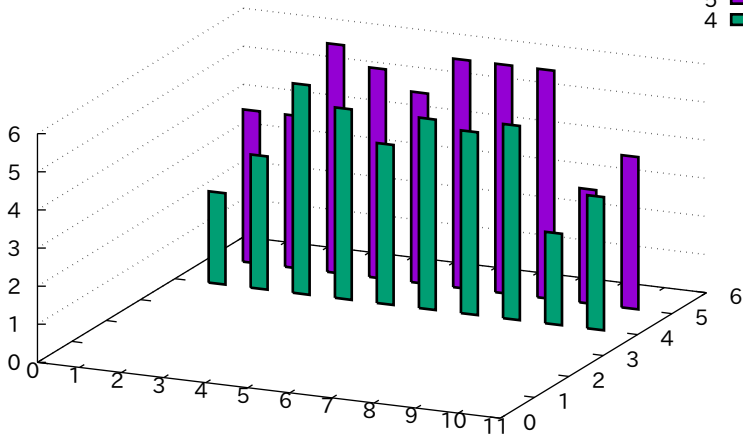
Key made with explicit
keyentry elements

-  points
-  lp
-  zerrorfill
-  circles
-  normal plot
-  boxes
-  labels
-  vectors
- pm3d

Test/demo of new feature 'grid walls'

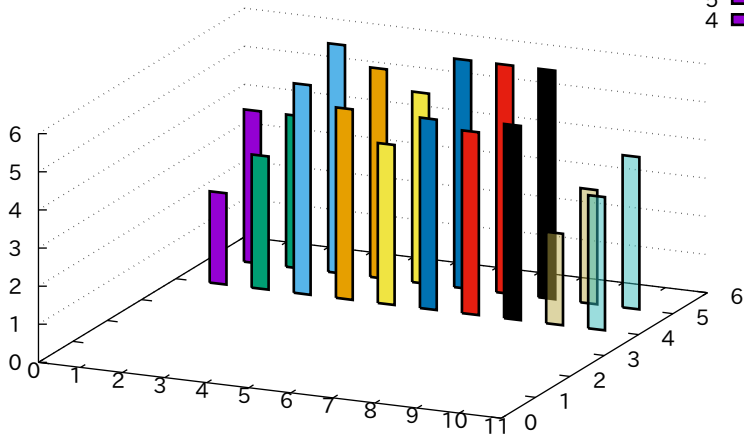


3D Boxes



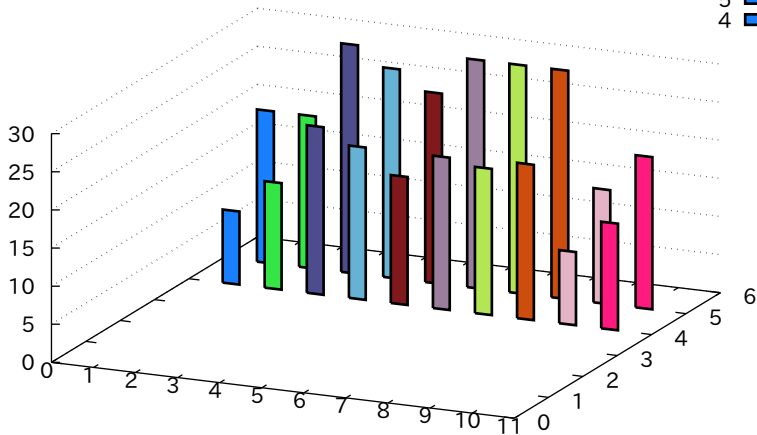
lc variable (from column 1)

5 
4 

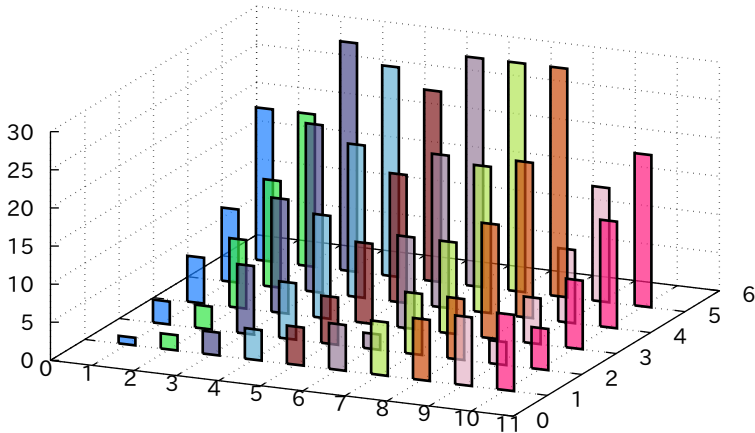


lc rgb variable

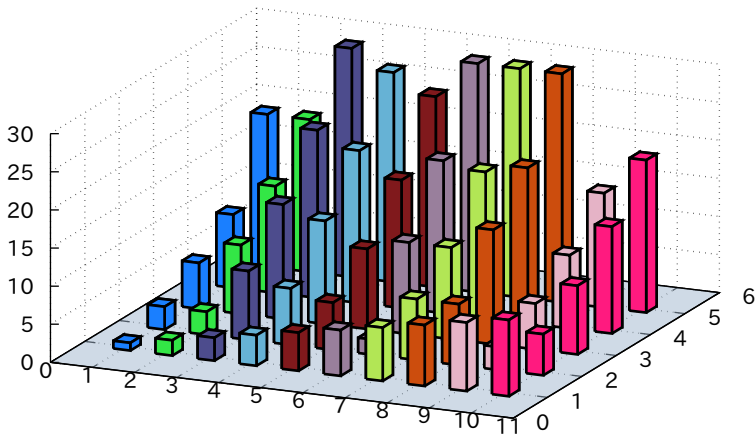
5 
4 



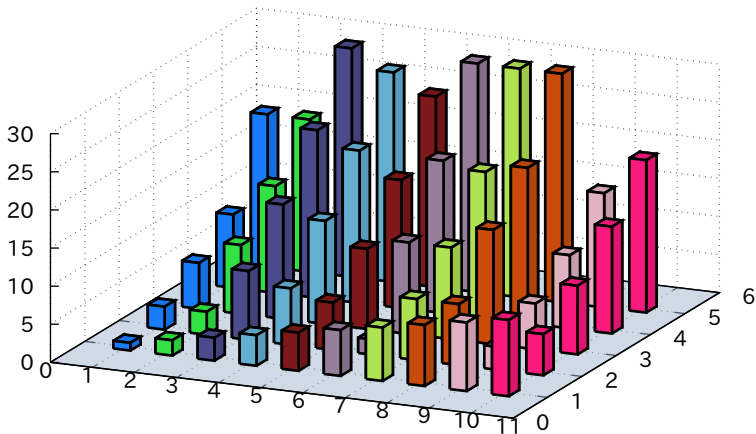
transparent boxes with imperfect depth sorting



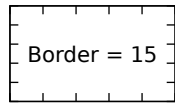
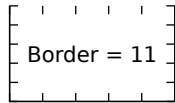
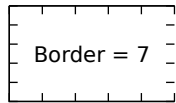
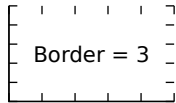
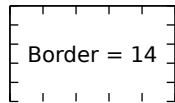
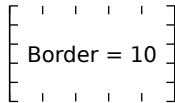
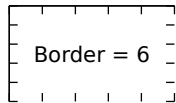
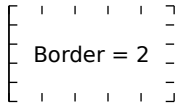
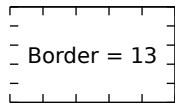
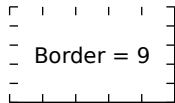
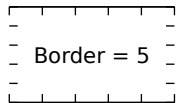
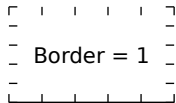
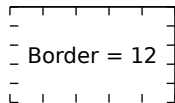
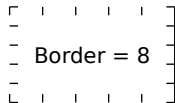
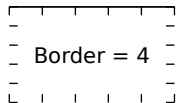
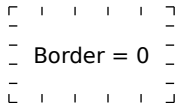
give the boxes a 3D depth and correct depth sorting



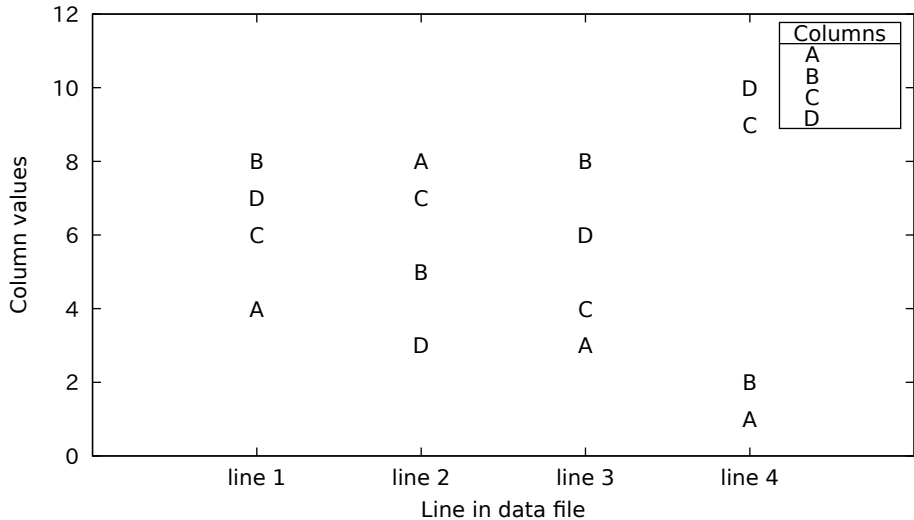
Full treatment: 3D boxes with pm3d depth sorting and lighting



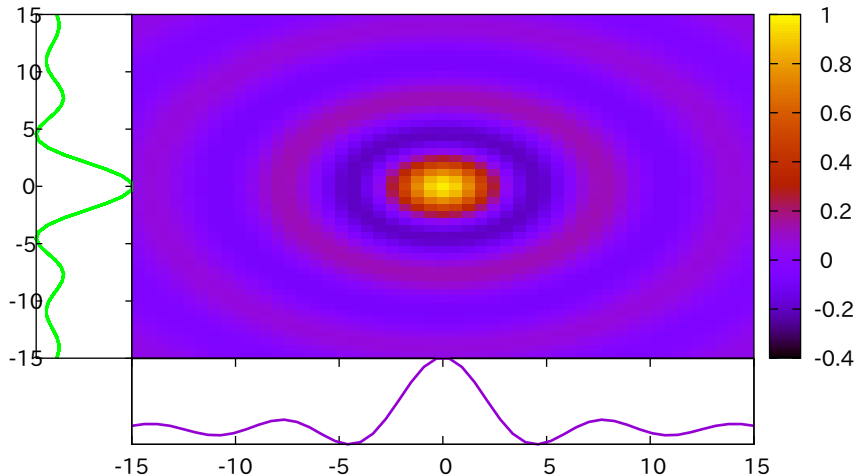
Demonstration of different border settings

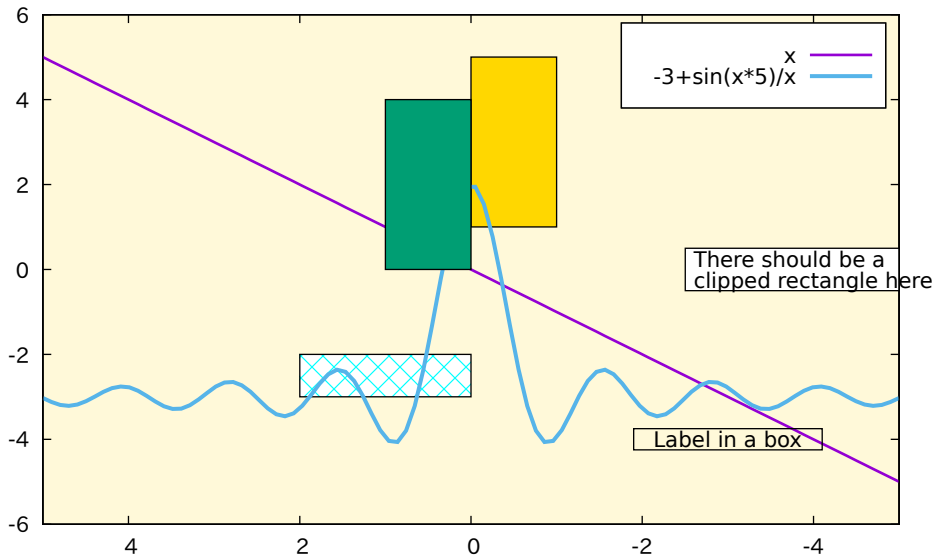


Point labels show which column they came from

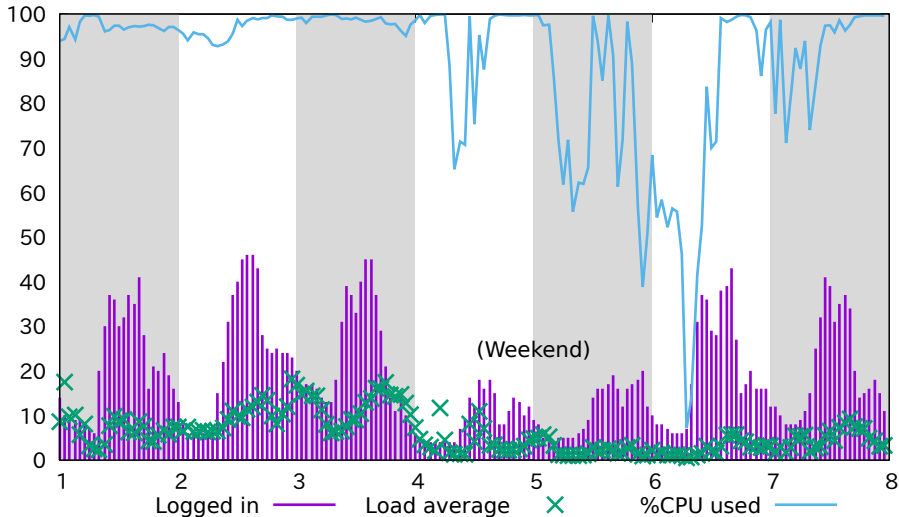


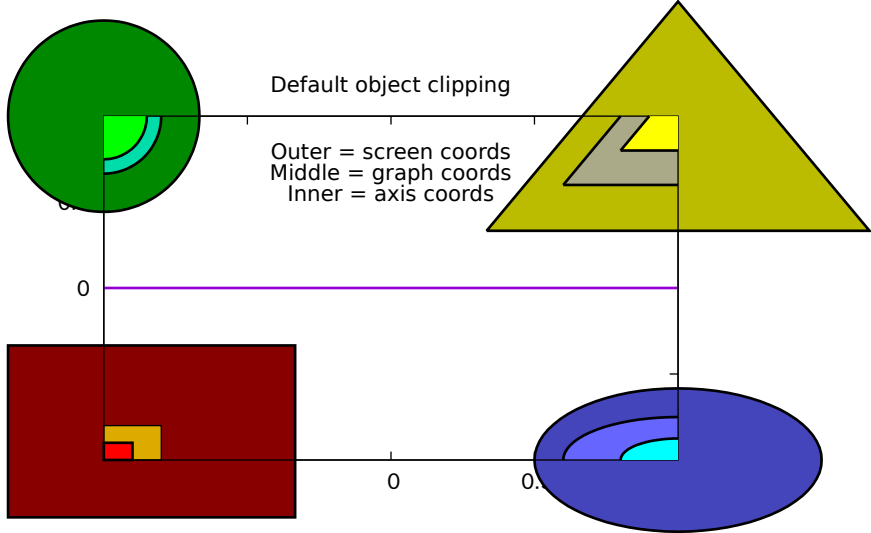
Demo of placing multiple plots (2D and 3D)
with explicit alignment of plot borders

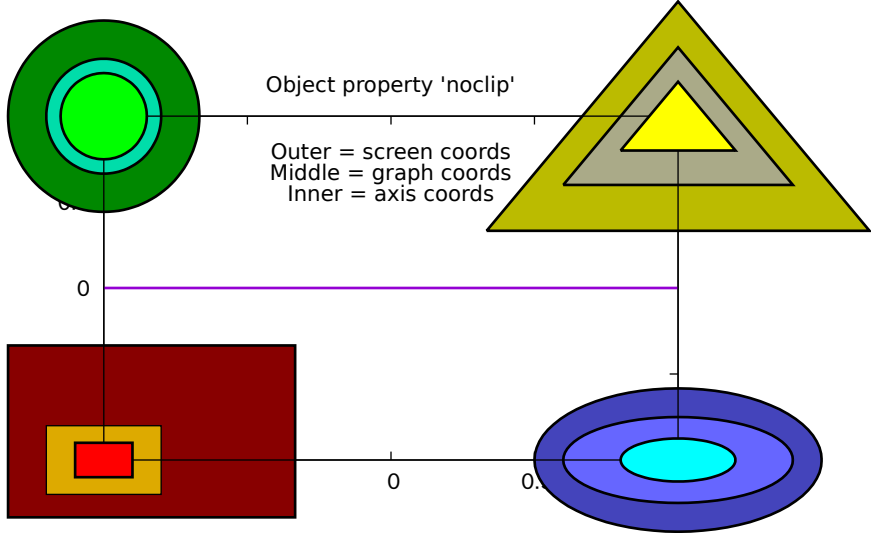




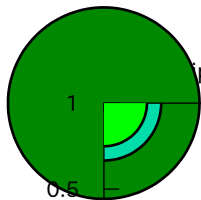
Convex November 1-7 1989



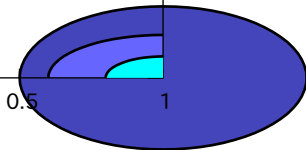
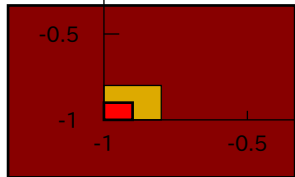
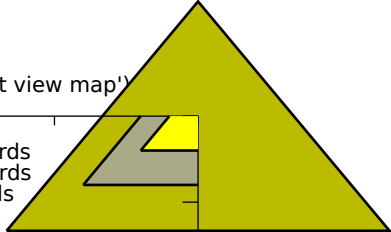


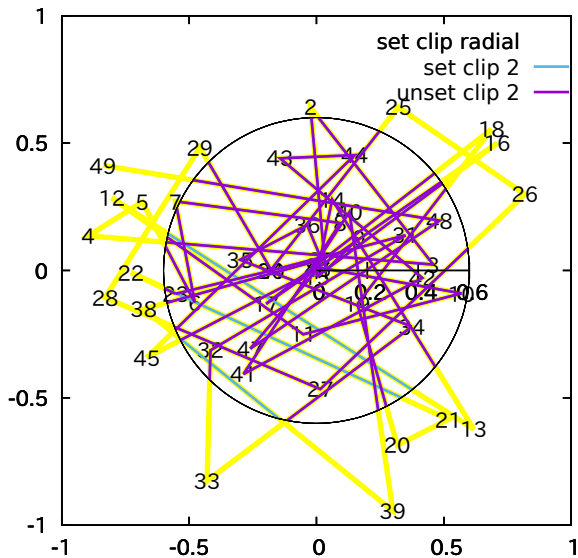


Clipping in 3D projection ('set view map')

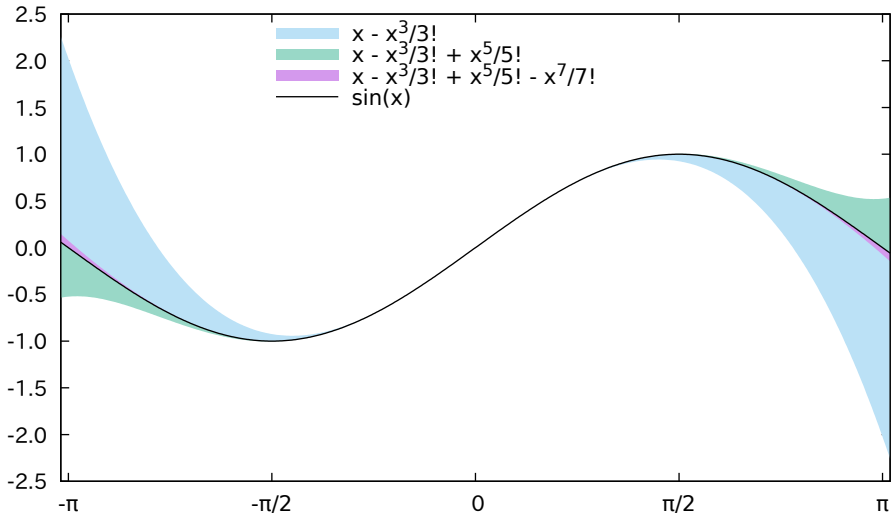


Outer = screen coords
Middle = graph coords
Inner = axis coords

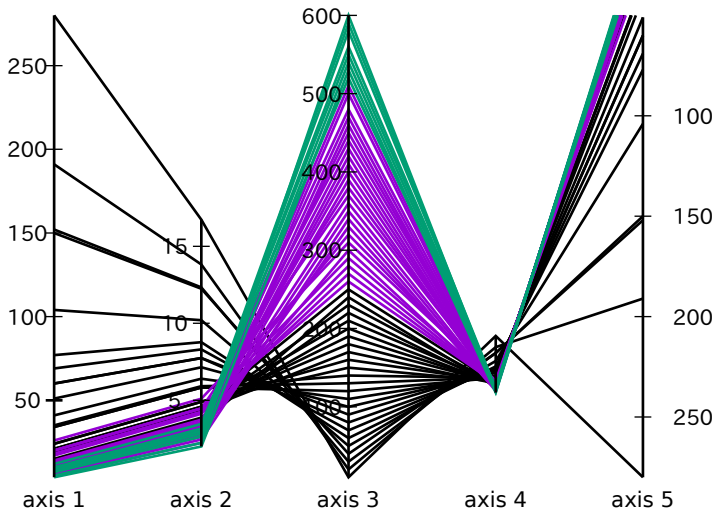




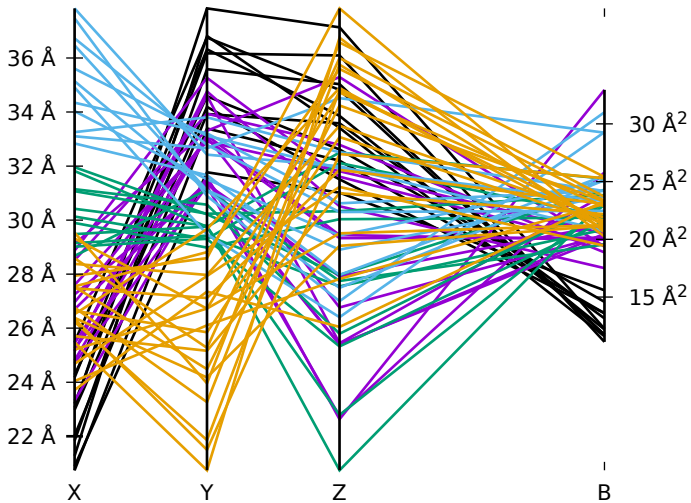
Polynomial approximation of $\sin(x)$



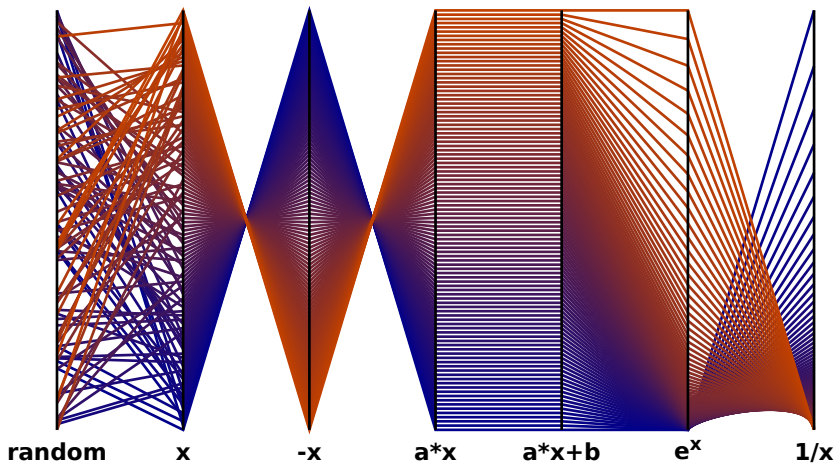
Parallel Axis Plot



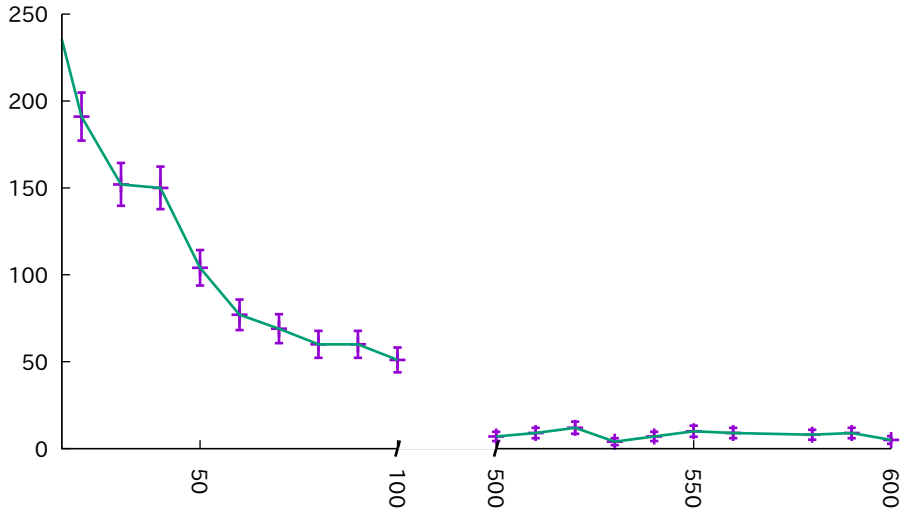
Parallel Axis Plot



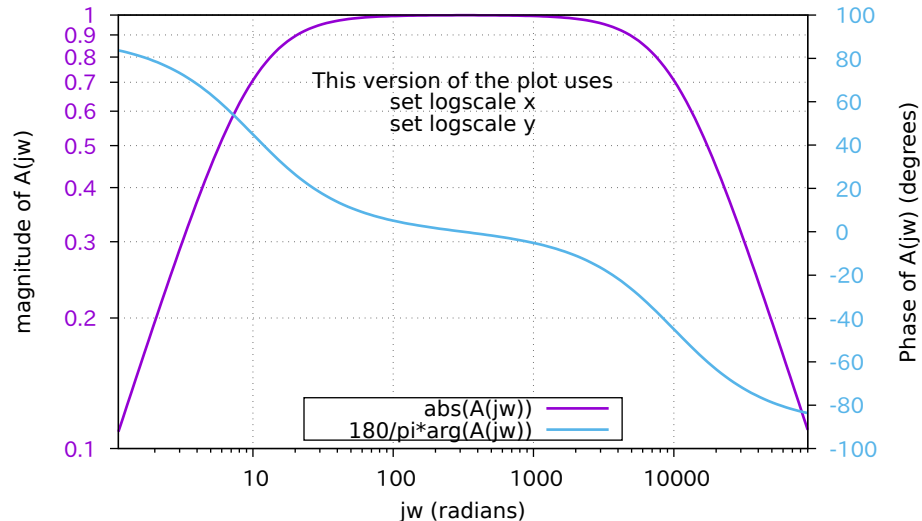
Parallel Axis Plot



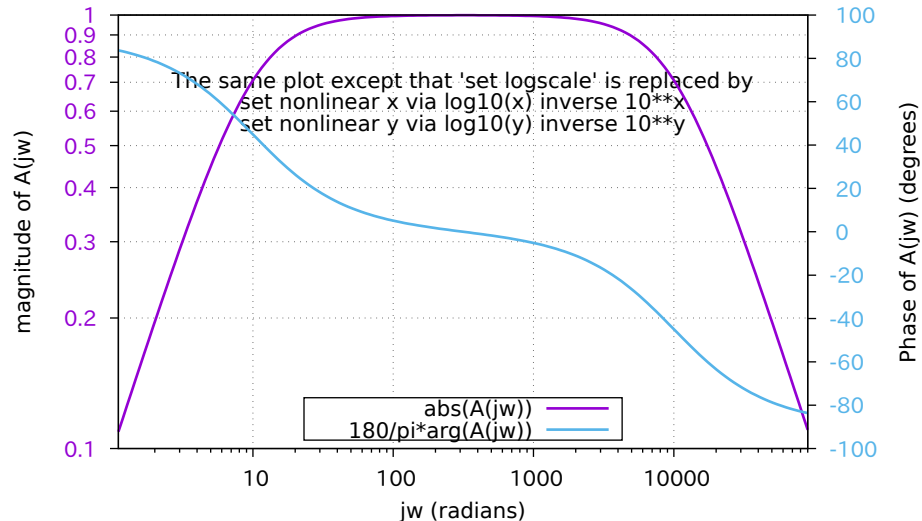
A 'broken' x axis can be defined using 'set nonlinear x'



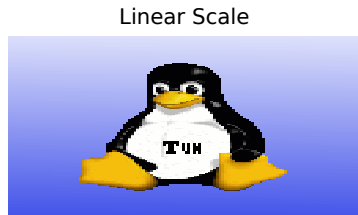
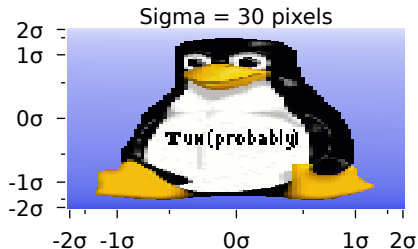
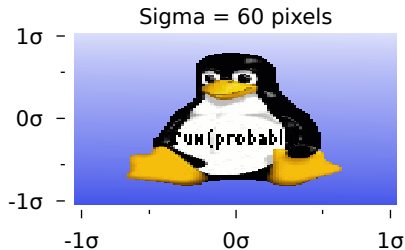
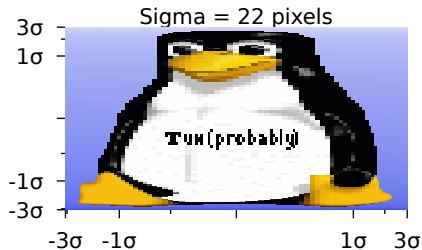
Log-scaled axes defined using 'set log'



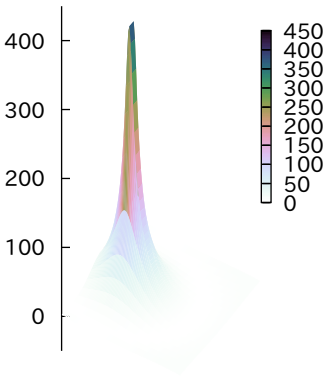
Log-scaled axes defined using 'set nonlinear'



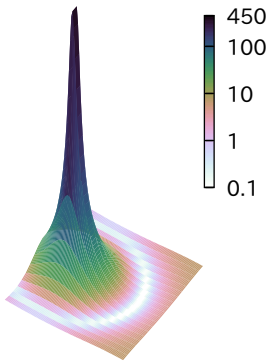
Probability axes: Scale image pixels by distance from center treated as a Z-score



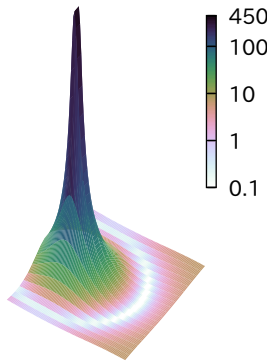
Linear cb axis



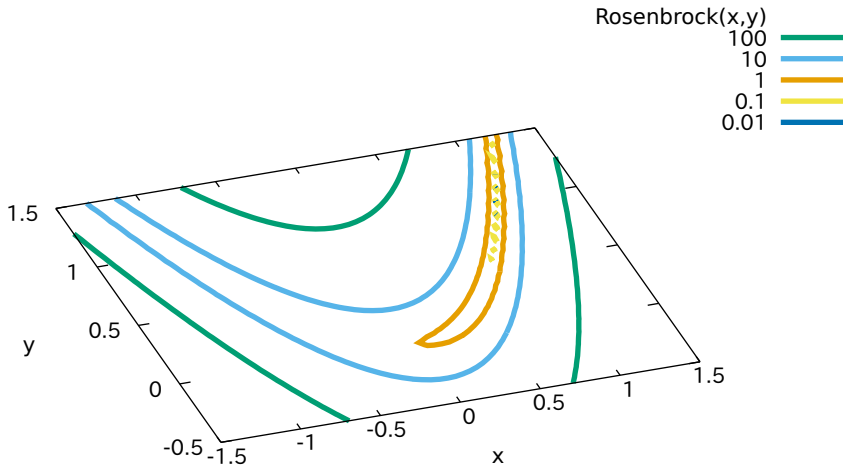
set log cb

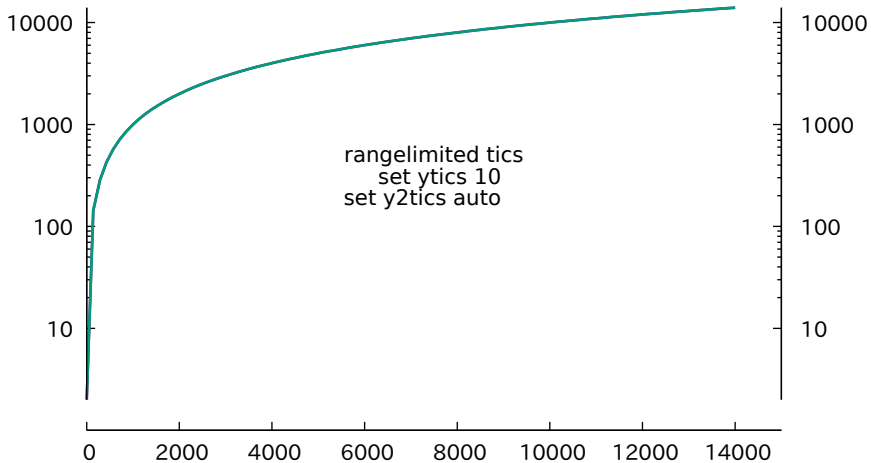


set nonlinear cb
via $\log_{10}(z)$ inv 10^{**z}

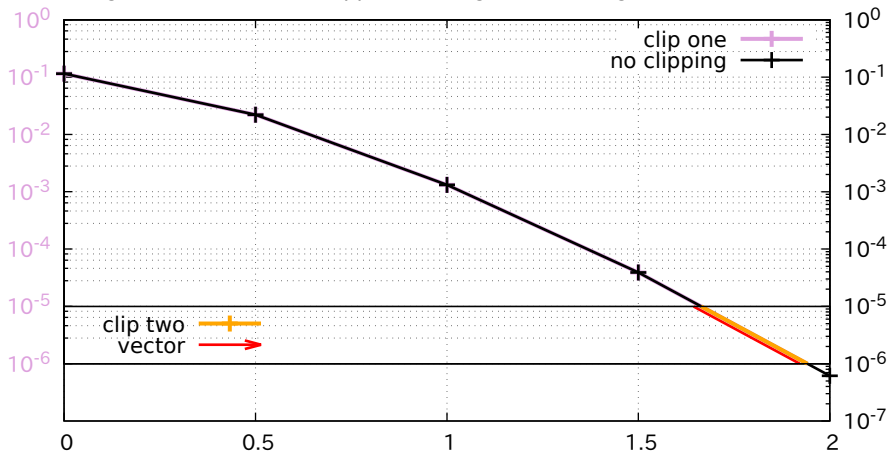


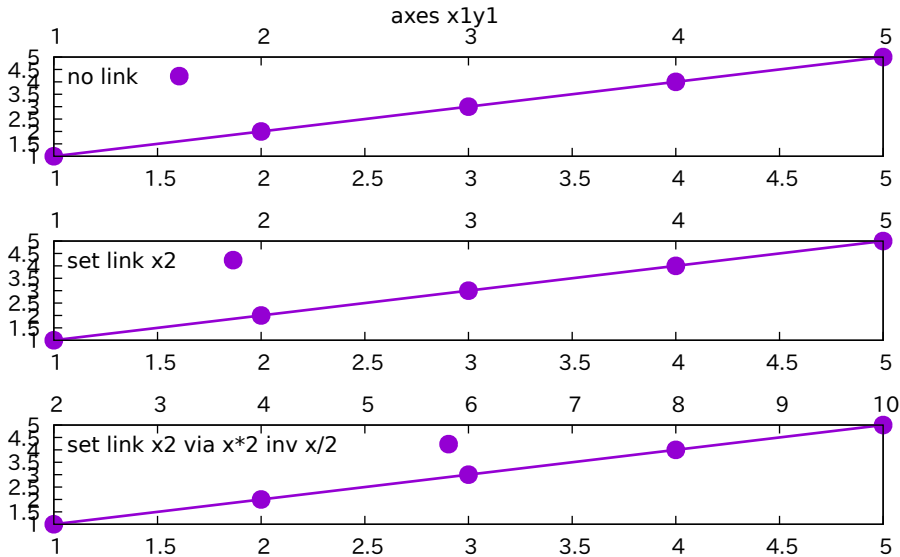
Rosenbrock Function

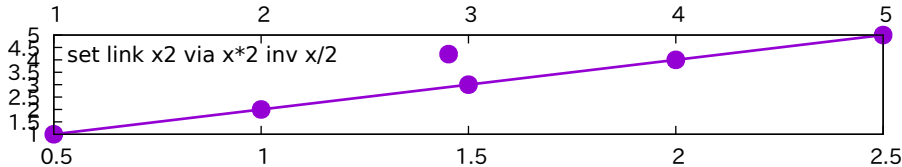
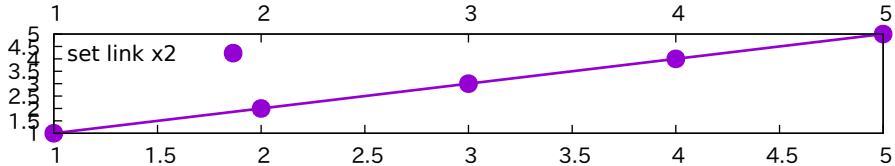
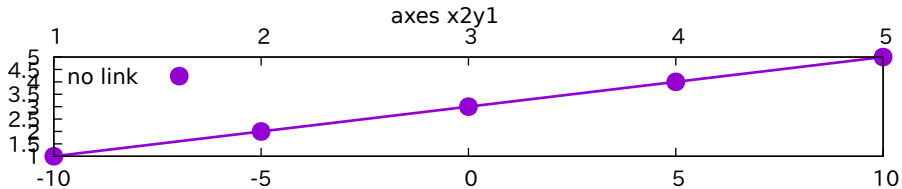




Bug #2046 - incorrect clipped line segments for logscale coordinates

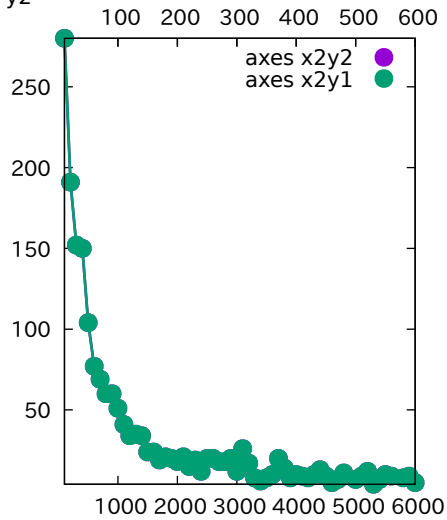
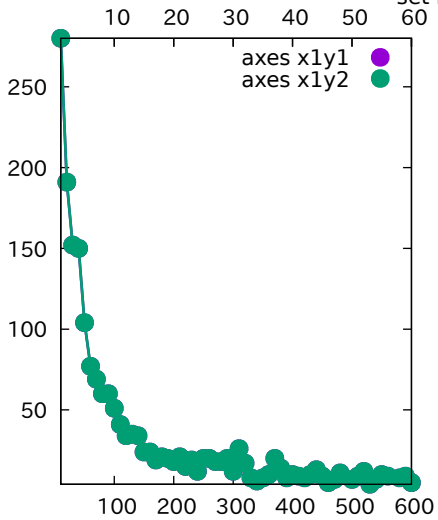




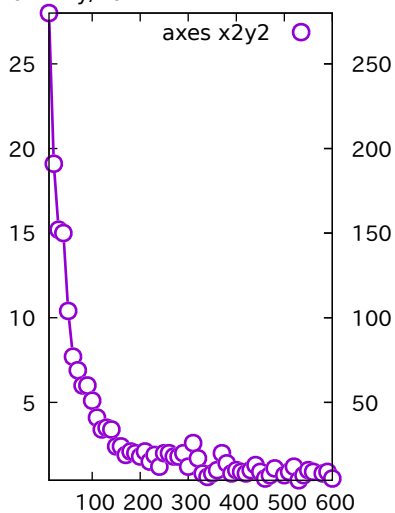
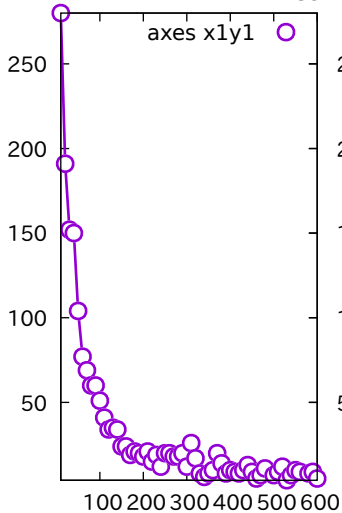


set link x2 via x/10. inv x*10

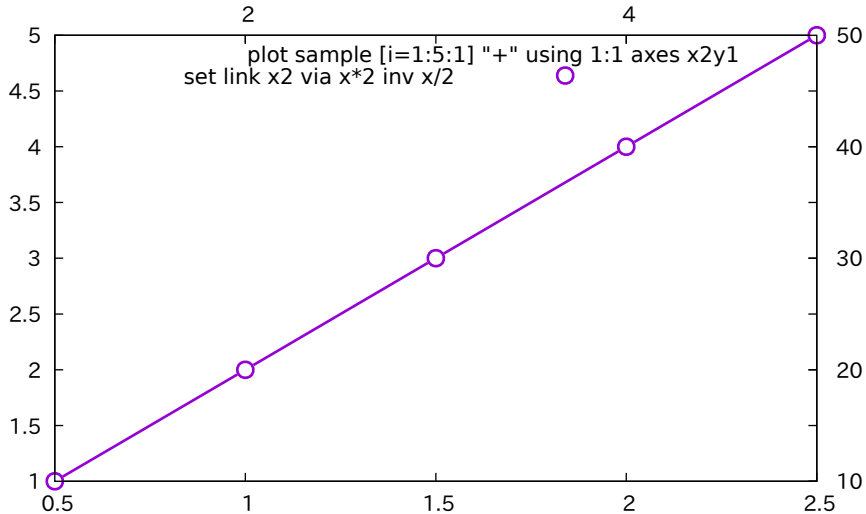
set link y2



set link x2
set link y2 via $y \cdot 10$. inv $y/10$.

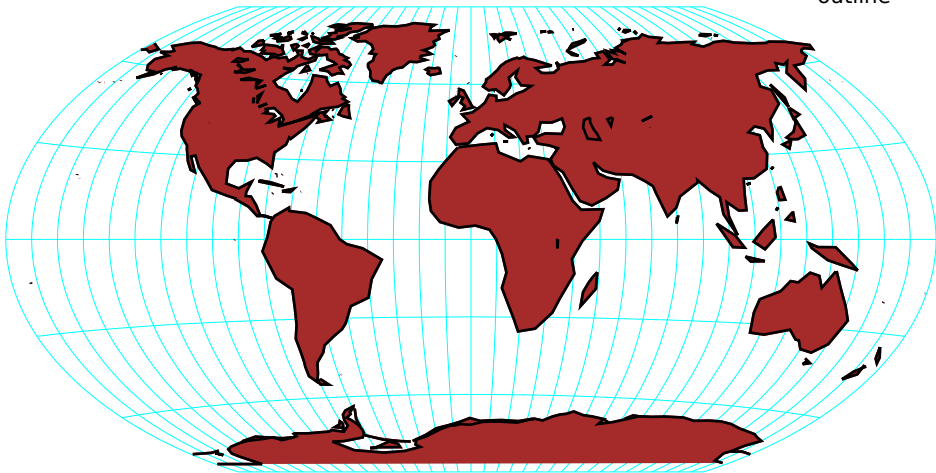


Should be 5 samples but bug may give only 3



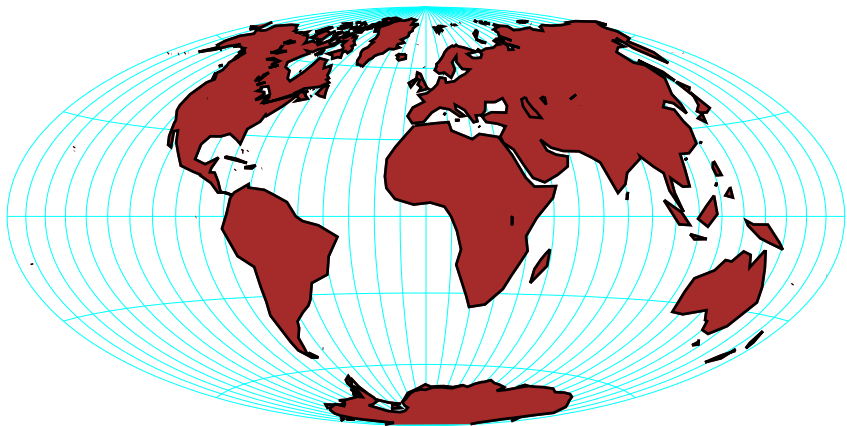
'Winkel tripel' map projection

fill ■
outline —



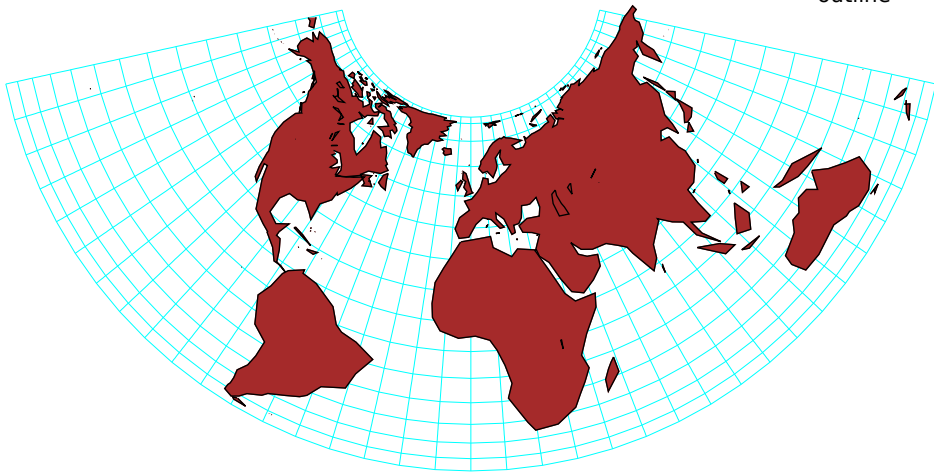
Hammer equal-area map projection

fill ■
outline —

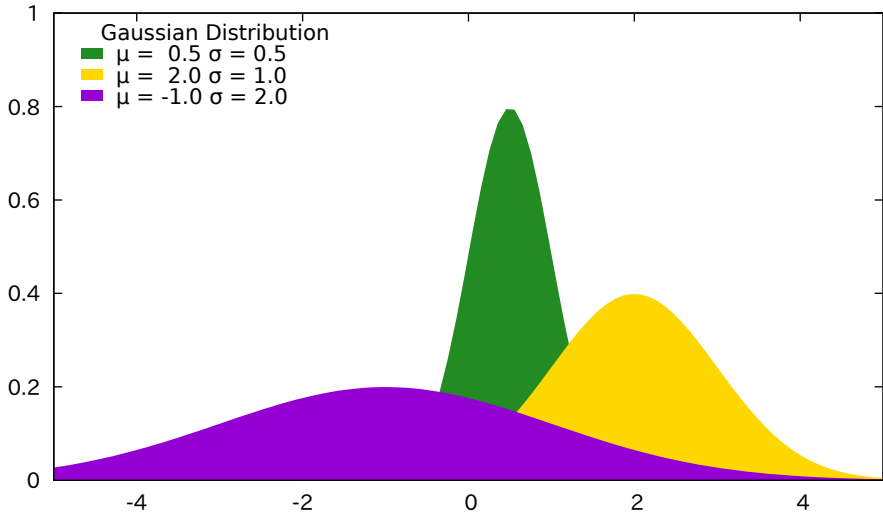


Albers equal-area conic projection

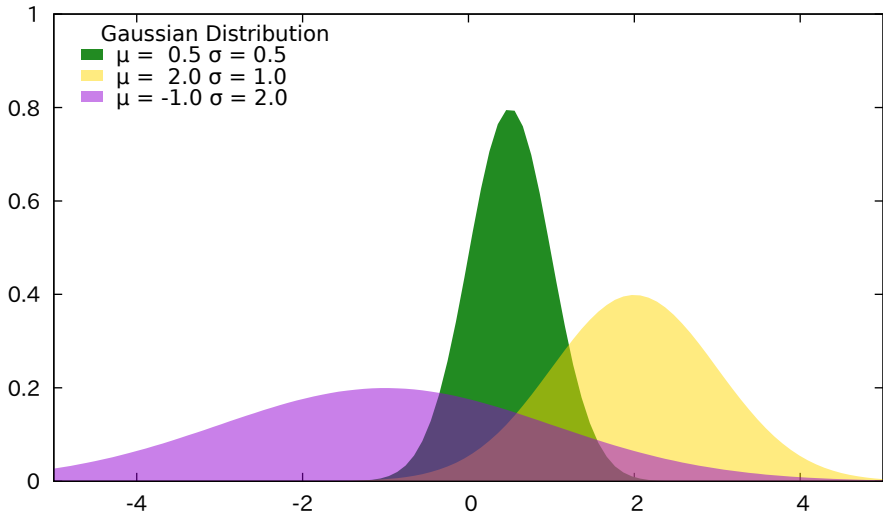
fill ■
outline —



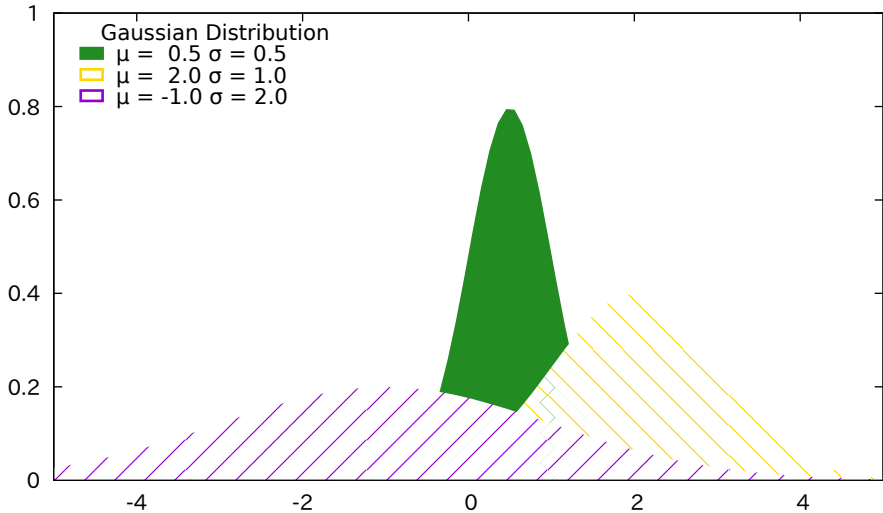
Solid filled curves



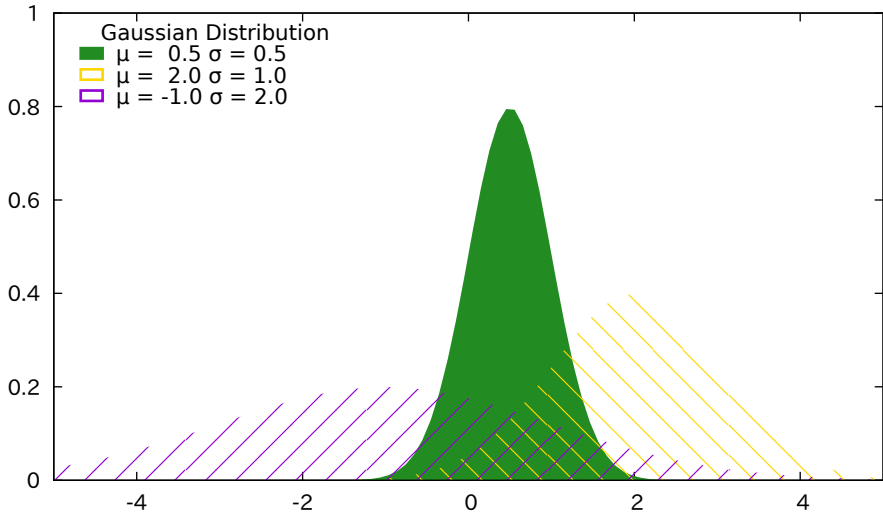
Transparent filled curves



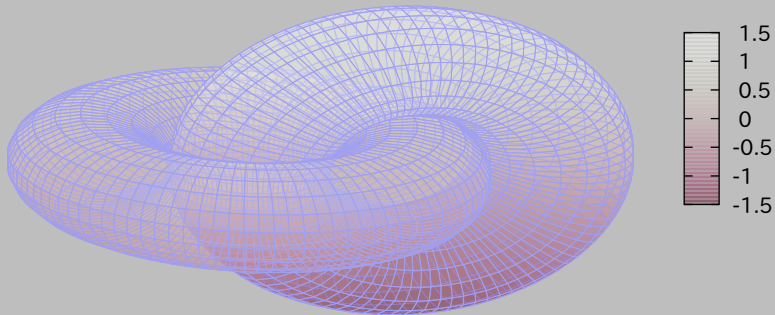
Pattern-filled curves



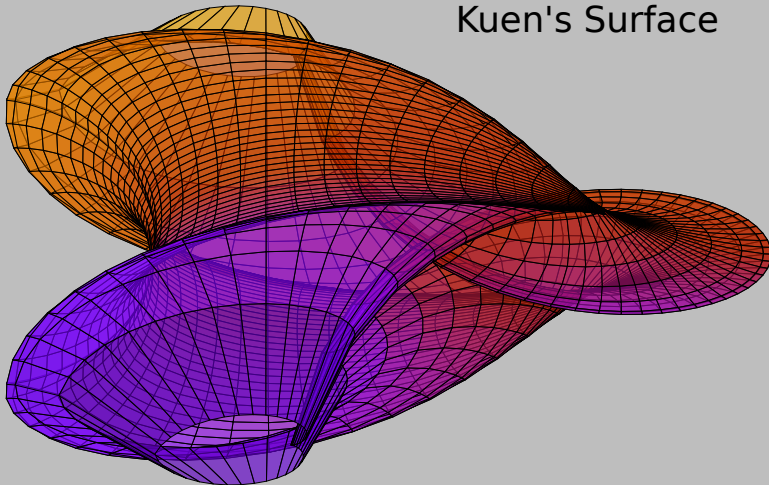
Transparent pattern-filled curves



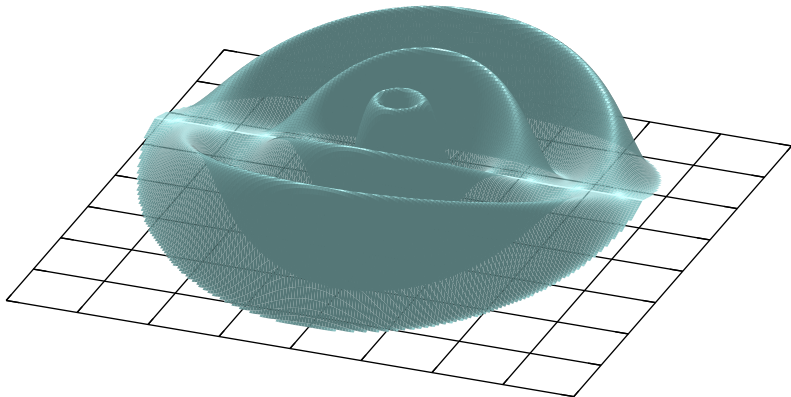
Interlocking Tori - PM3D surface with depth sorting and transparency



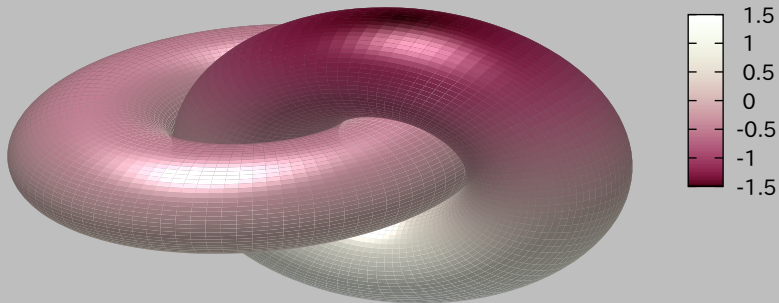
Kuen's Surface



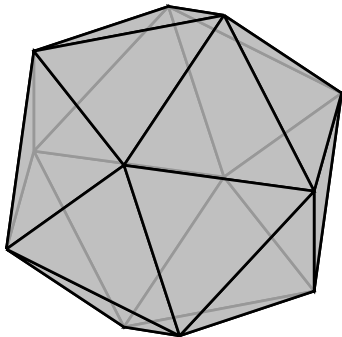
pm3d lighting model with specular highlighting



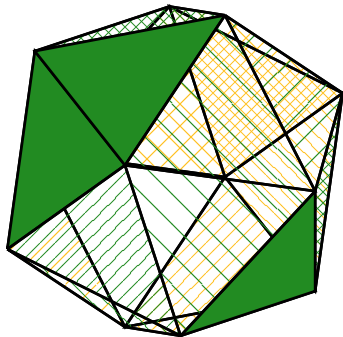
PM3D surfaces with specular highlighting



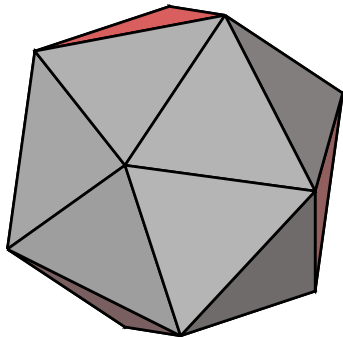
Faces of an icosahedron drawn as 20 individual objects



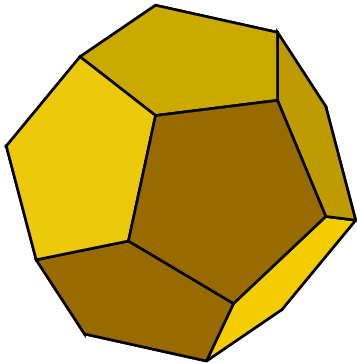
2-sided coloring
green outside, yellow inside



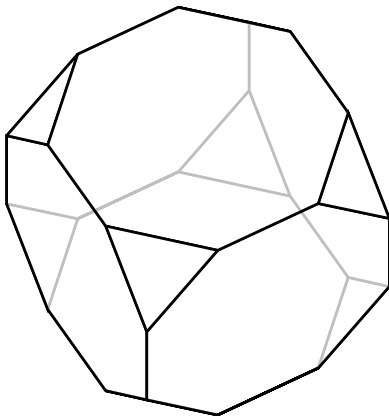
plot icosahedron.dat with polygons



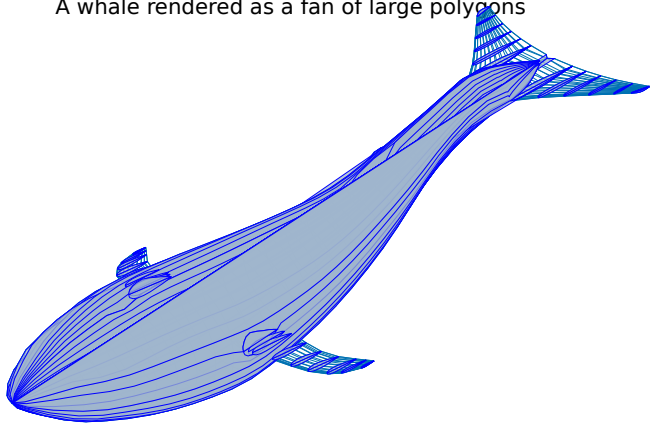
plot dodecahedron.dat with polygons



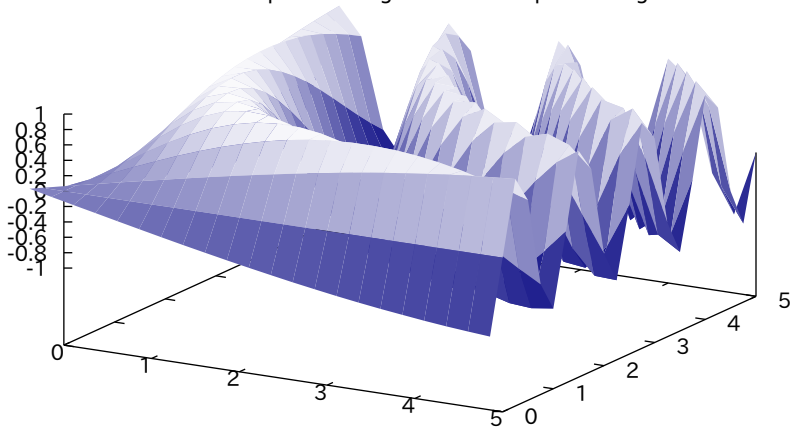
plot truncated_cube with polygons



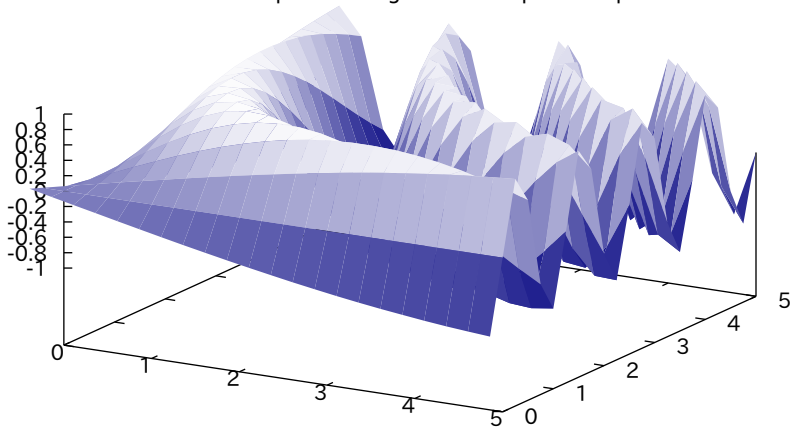
A whale rendered as a fan of large polygons



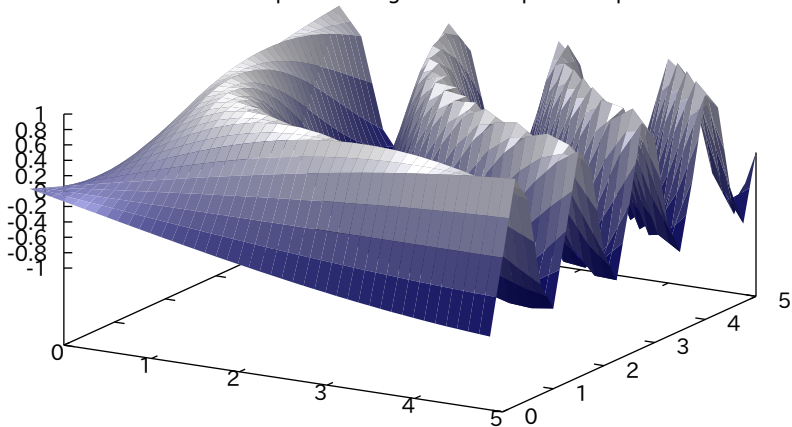
Use hand-constructed 'blues' palette via rgb variable
pm3d using 1:2:3:4 with pm3d lc rgb variable



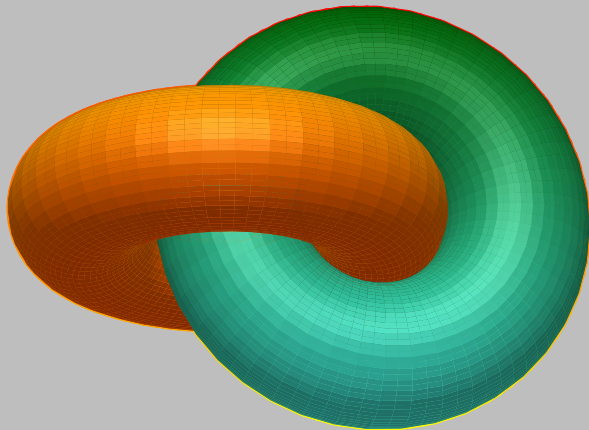
Version 5.5 offers a new keyword to access this palette
pm3d using 1:2:3 with pm3d fc palette blues



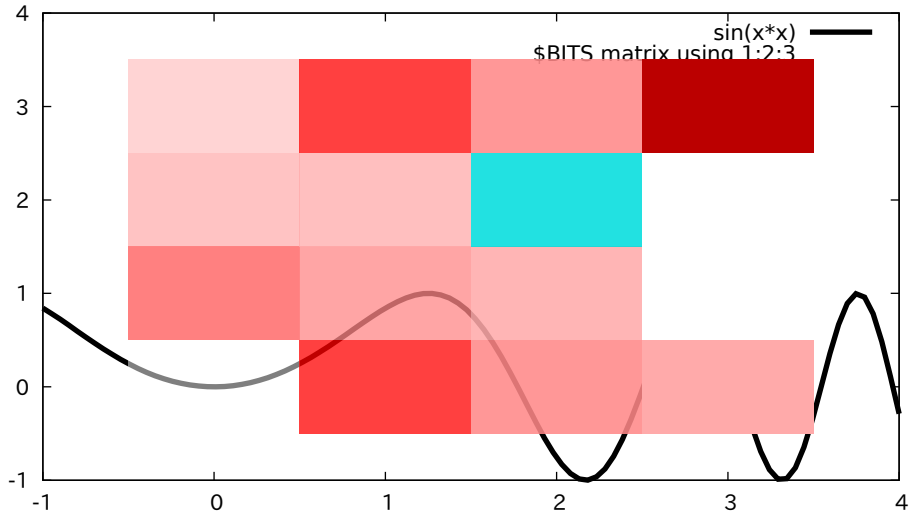
This method allow pm3d interpolation and lighting
pm3d using 1:2:3 with pm3d fc palette blues



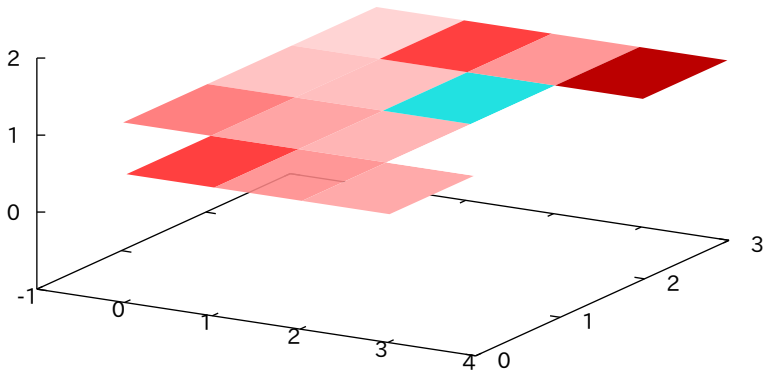
pm3d coloring using two named colormap palettes



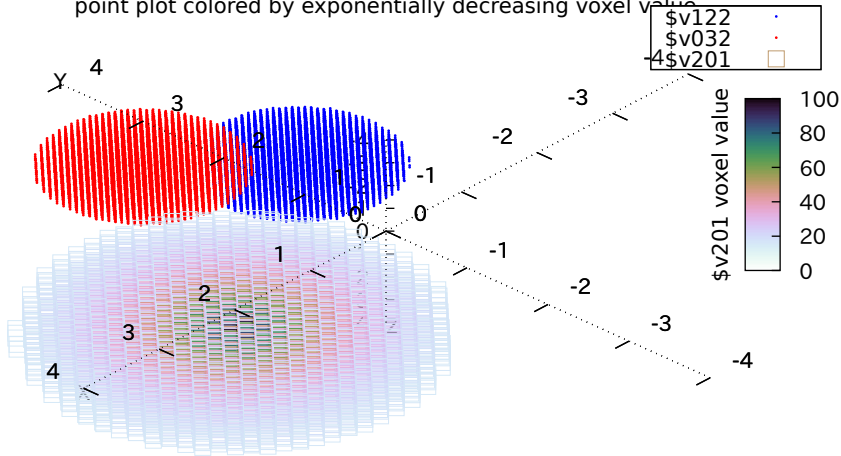
Alpha+RGB image data loaded from a hexadecimal matrix



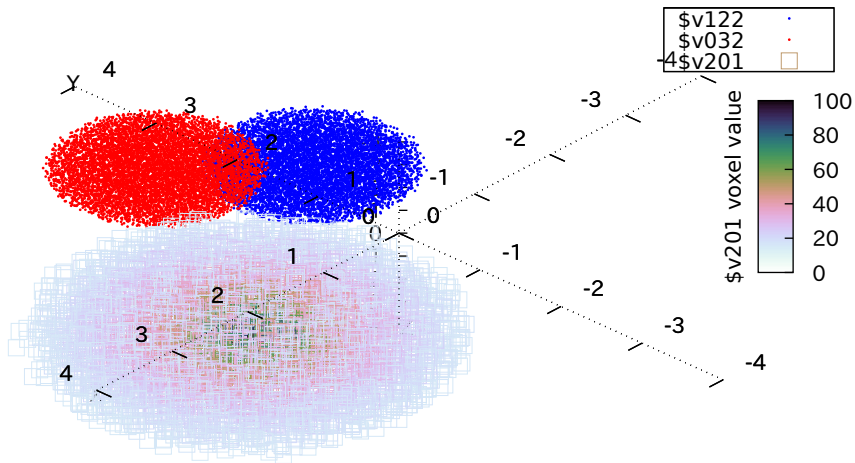
Alpha+RGB image data loaded from a hexadecimal matrix
\$BITS matrix using 1:2:(1.0):3



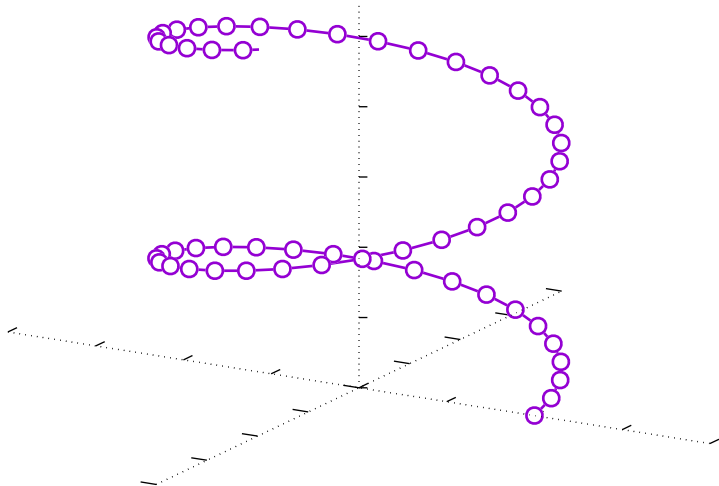
overlapping dot plots with constant color
point plot colored by exponentially decreasing voxel value



Same voxel plot with jitter

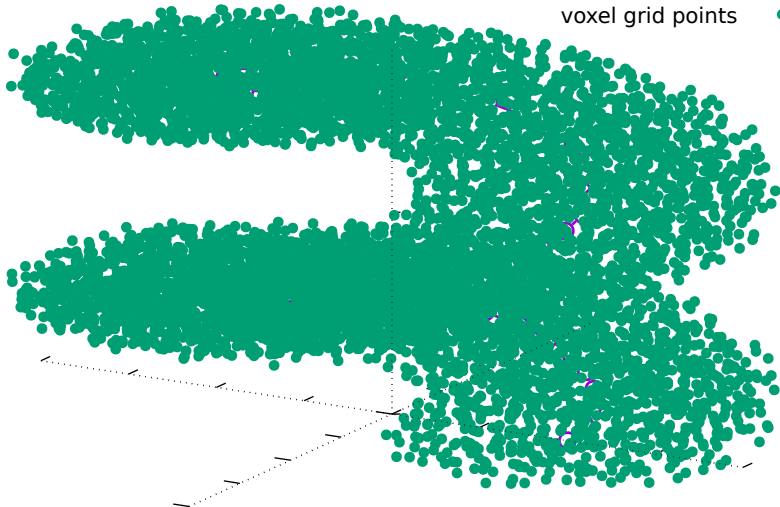


[t=0:20] '+' using (cos(\$1)):(sin(\$1)):(1) —○—



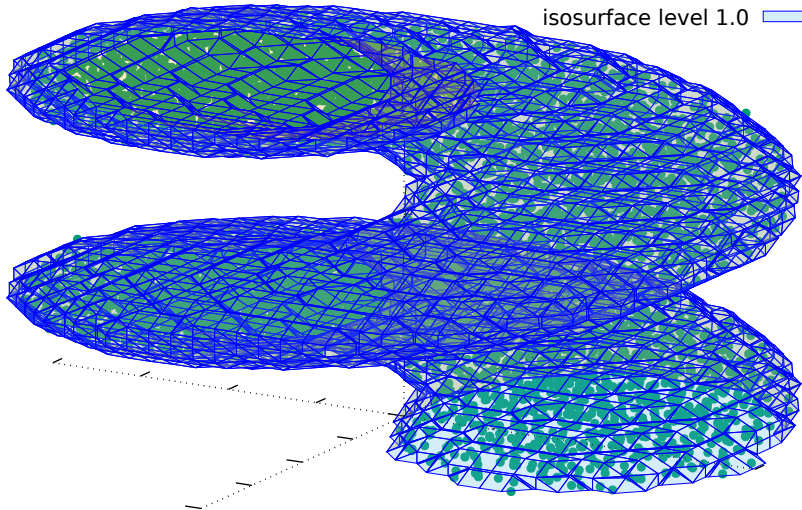
Fill voxel grid around the points shown

voxel grid points



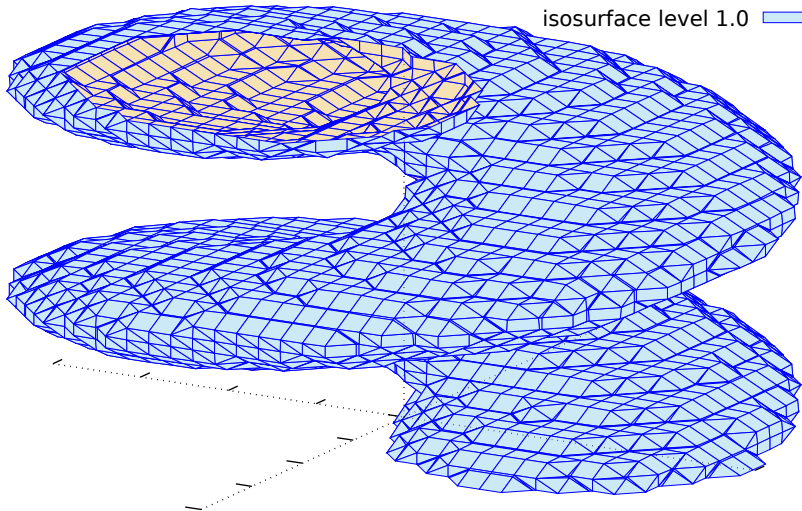
Draw isosurface enclosing all points

isosurface level 1.0 

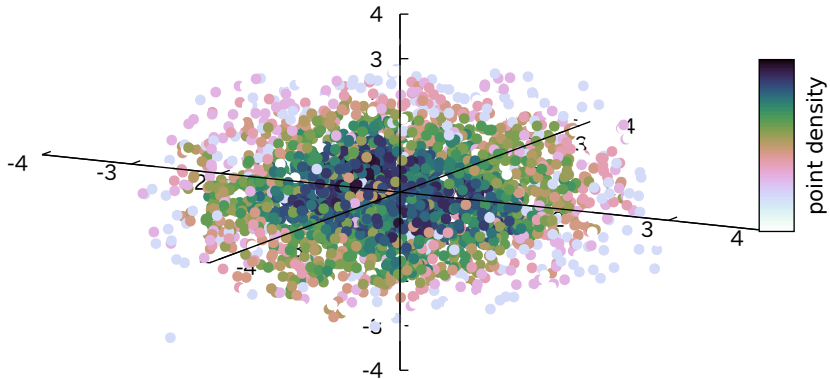


isosurface alone

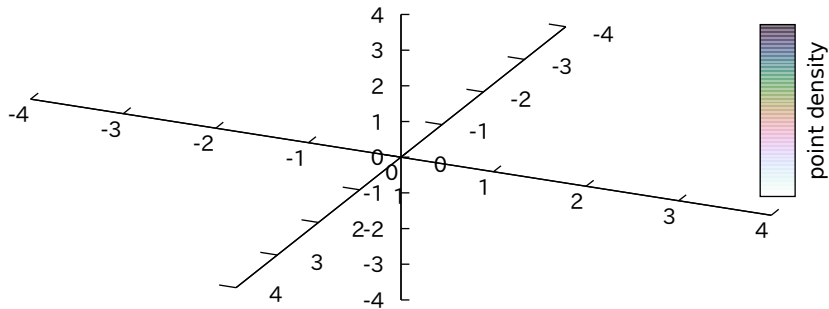
isosurface level 1.0 



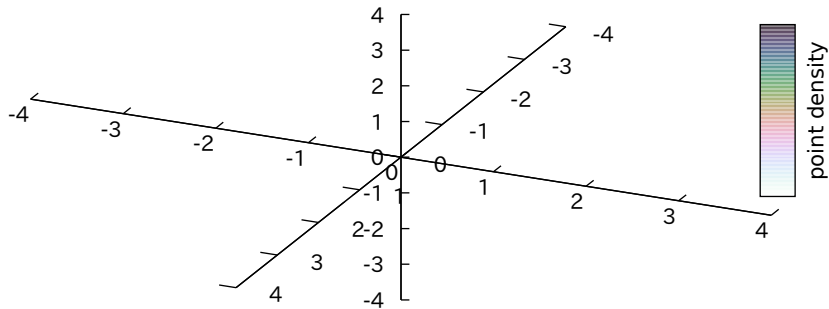
Gaussian 3D cloud of 3000 random samples
colored by local point density



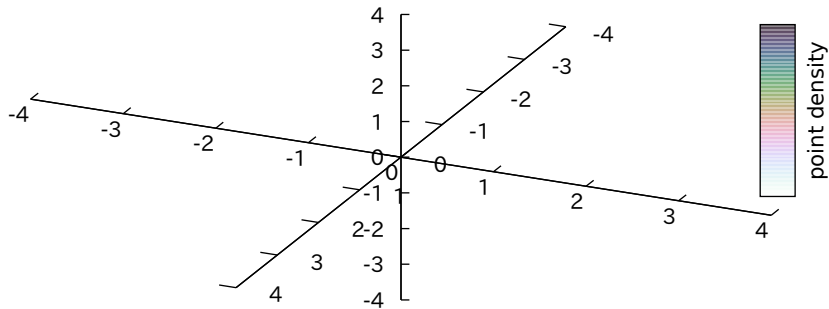
step through volume using the density values to color a surface
splot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



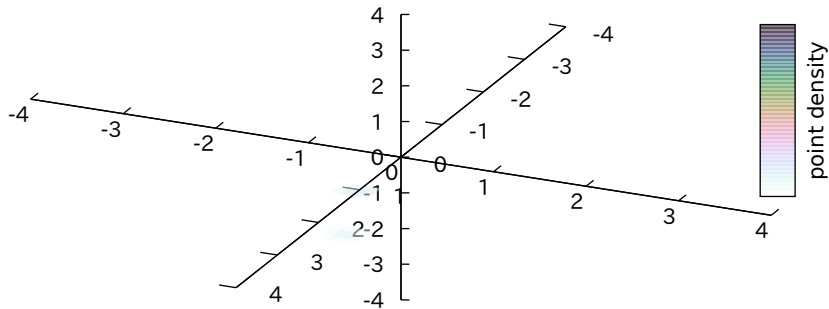
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



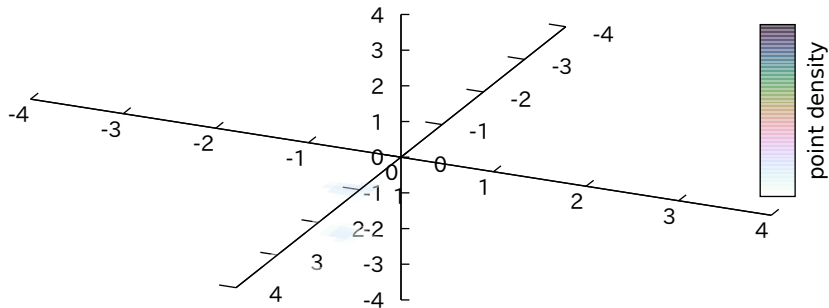
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



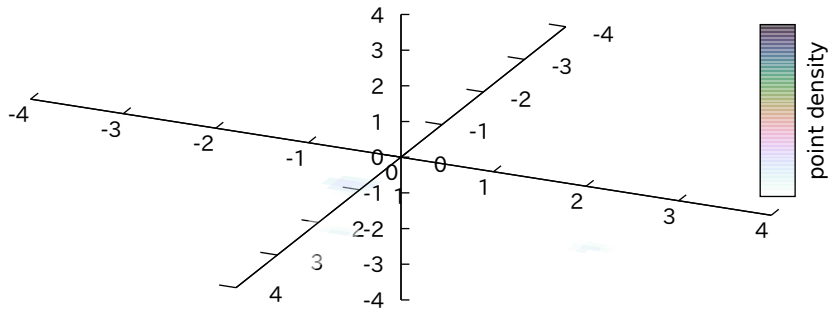
step through volume using the density values to color a surface
splot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



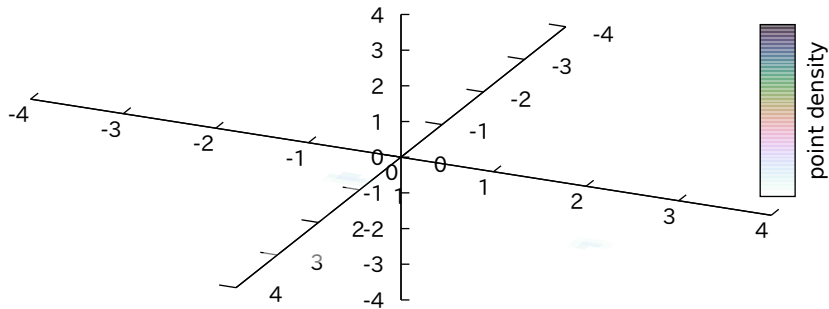
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



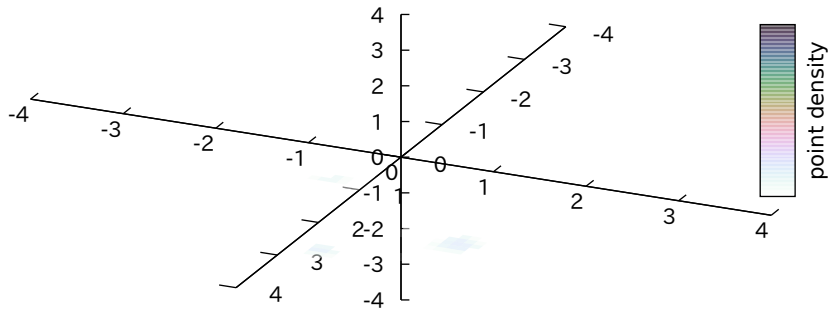
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



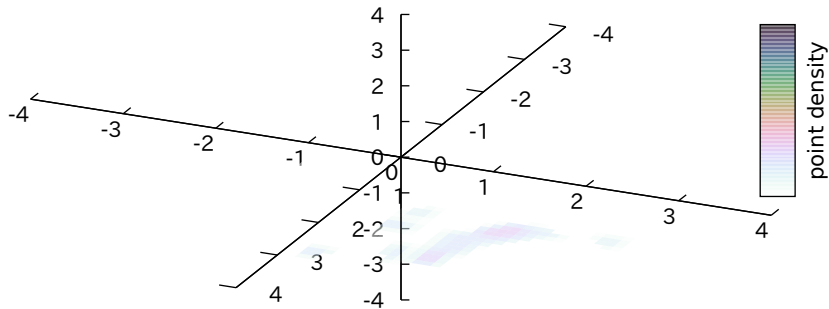
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



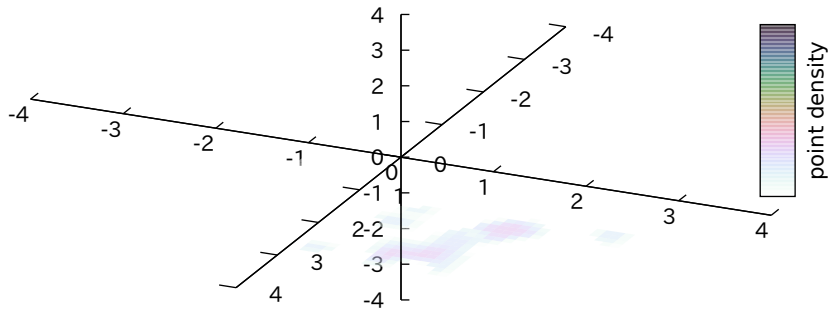
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



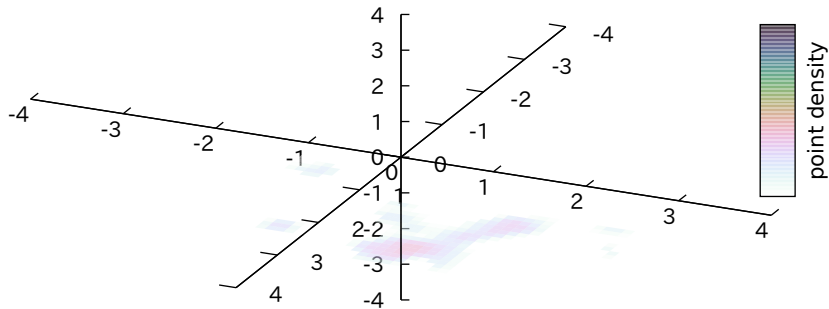
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



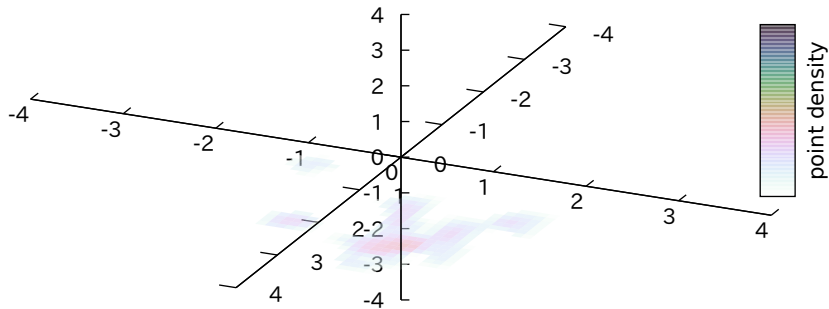
step through volume using the density values to color a surface
splot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



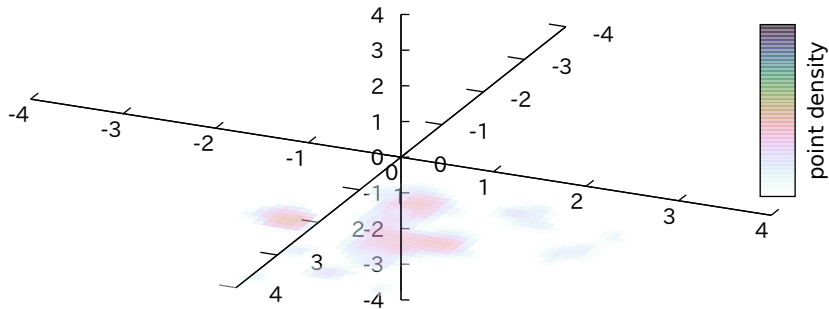
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



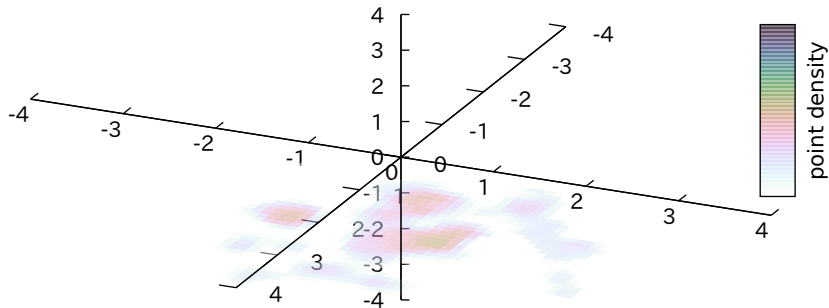
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



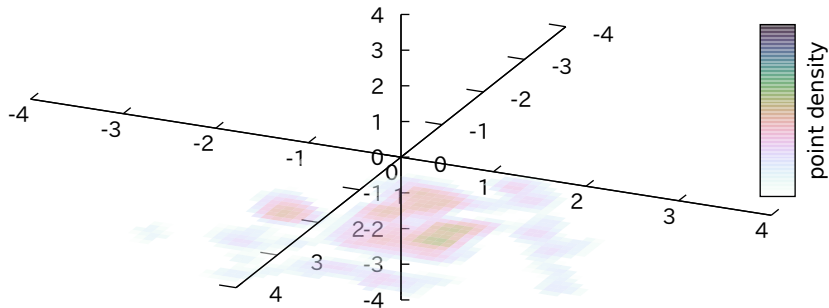
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



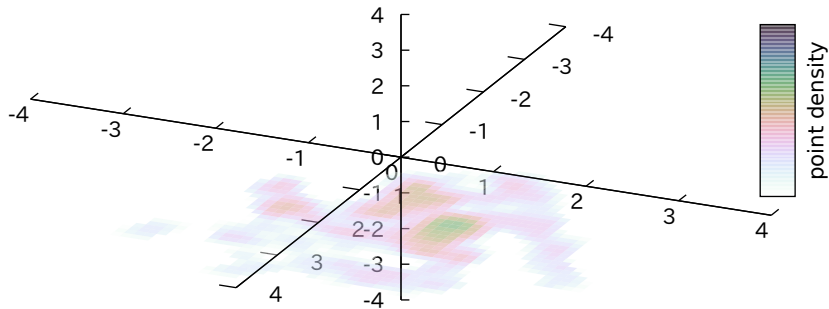
step through volume using the density values to color a surface
plot '+' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



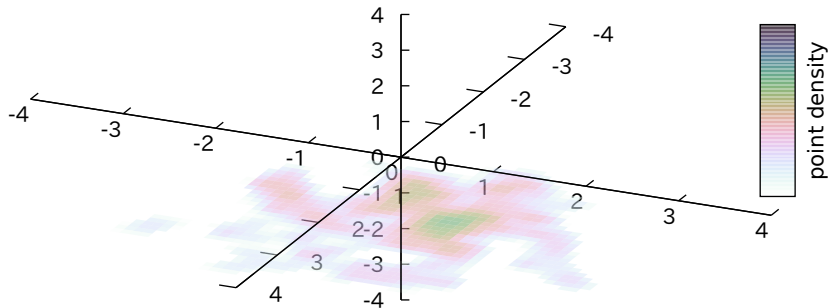
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



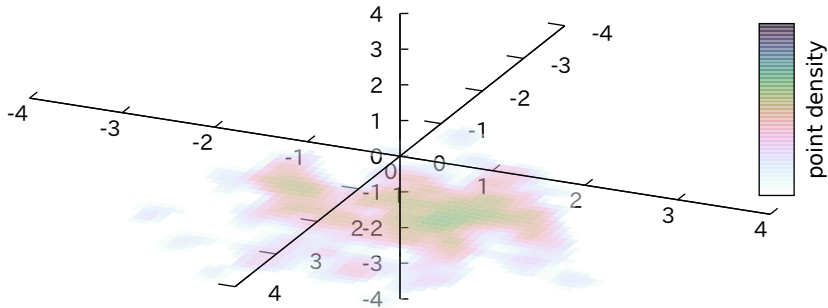
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



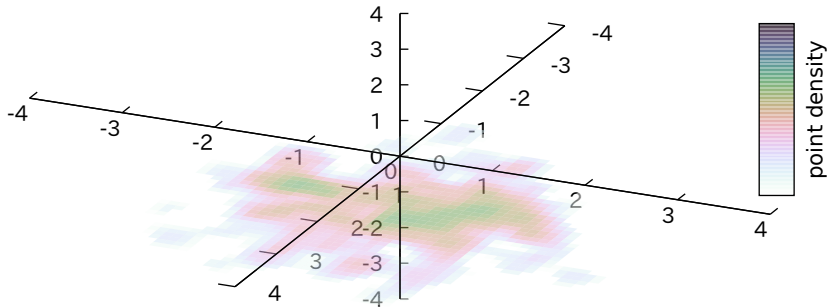
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



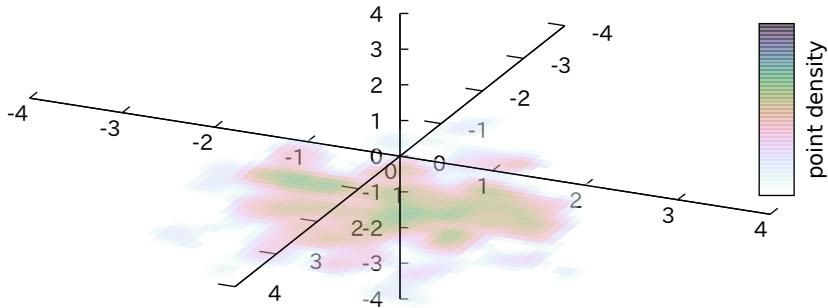
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



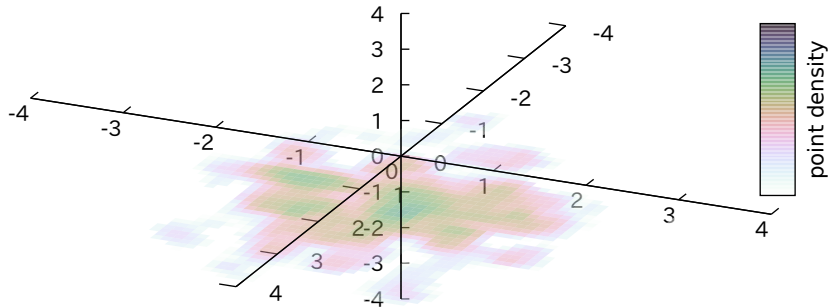
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



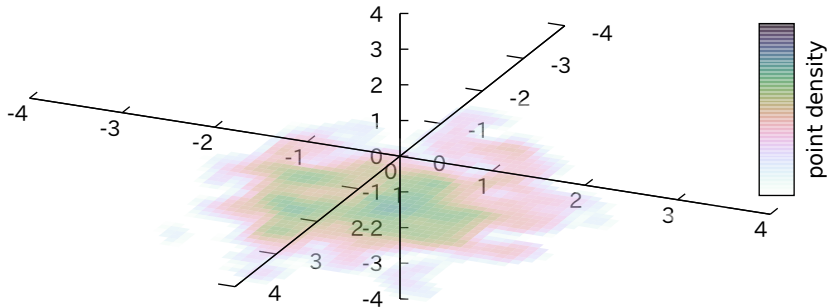
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



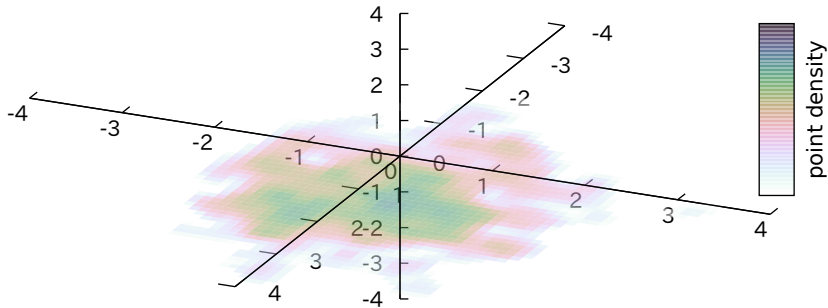
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



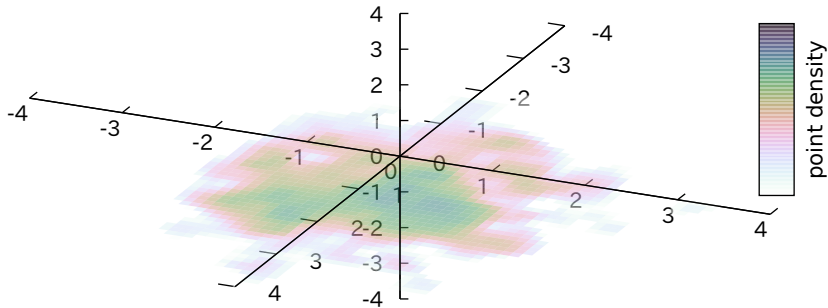
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



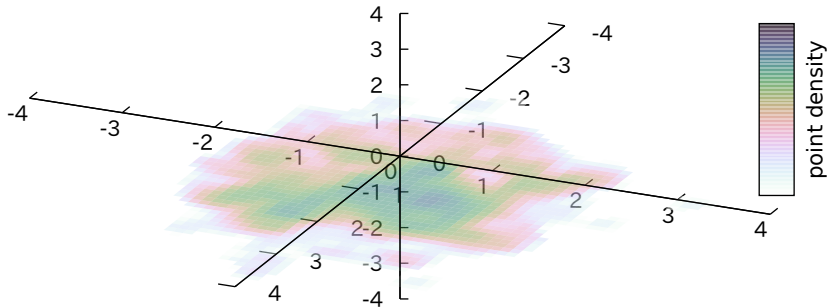
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



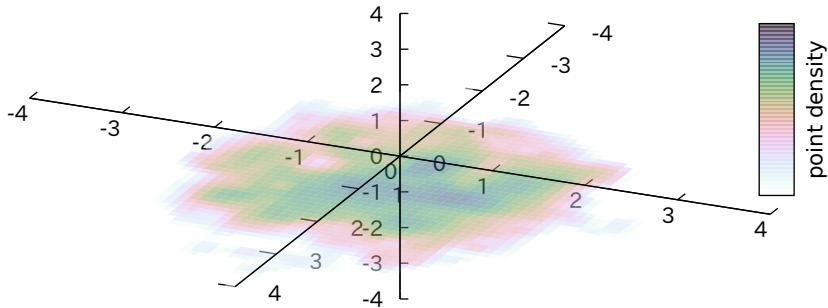
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



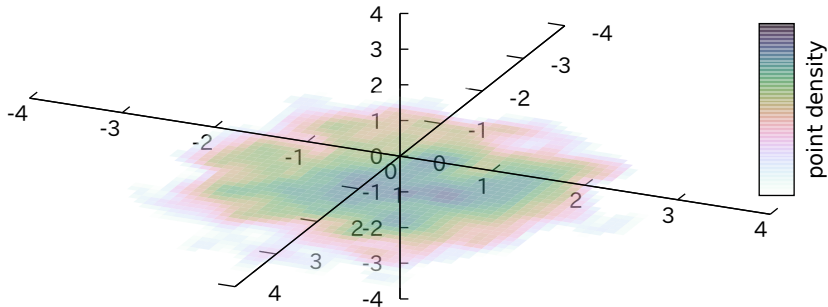
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



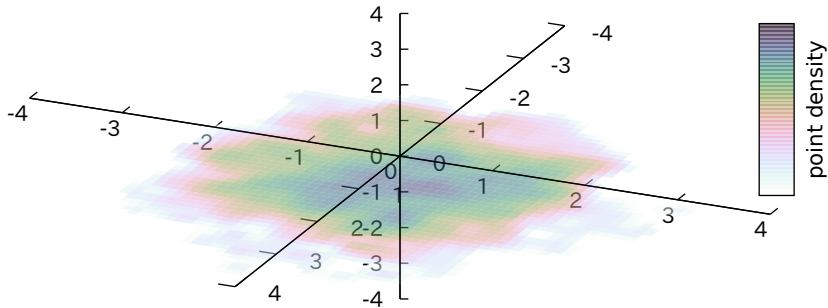
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



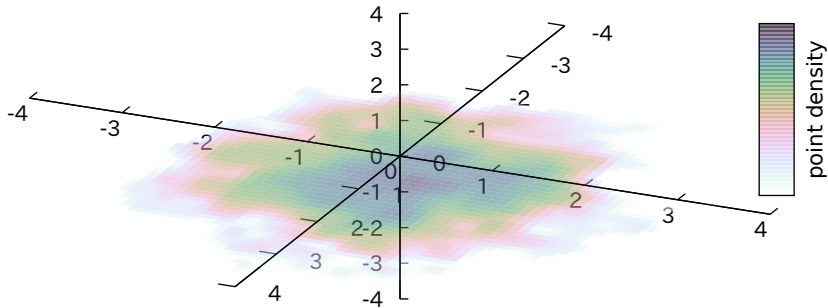
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



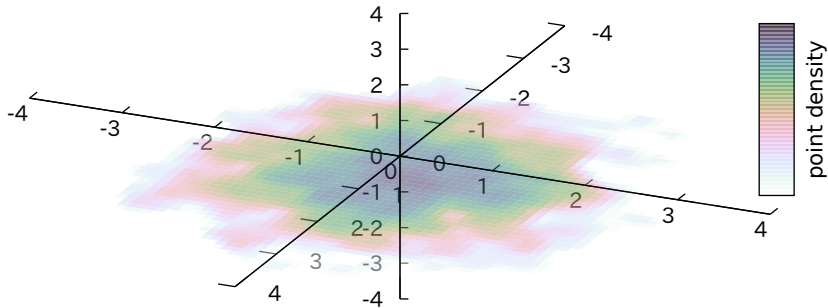
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



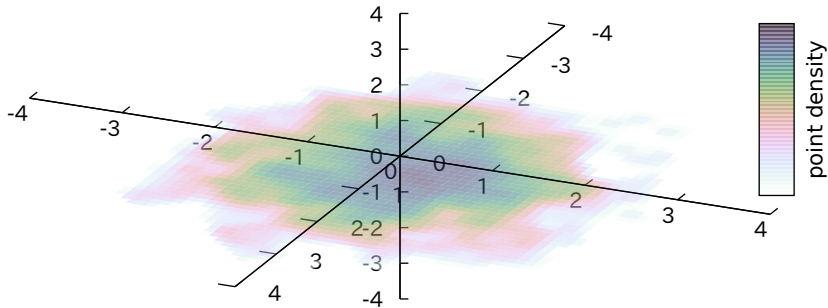
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



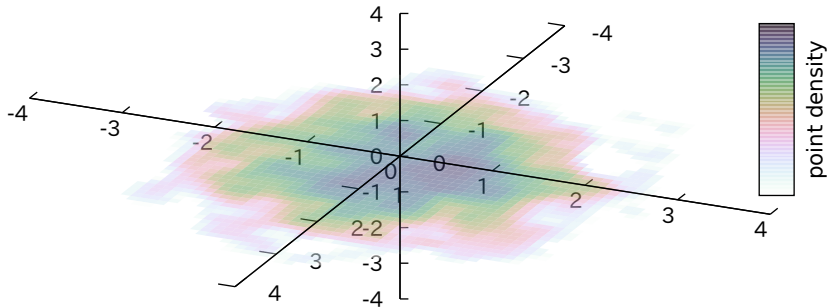
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



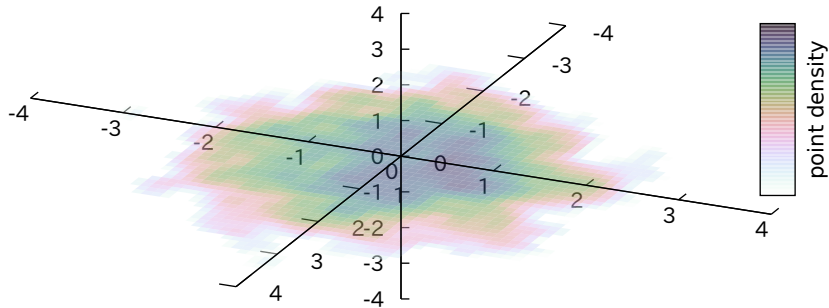
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



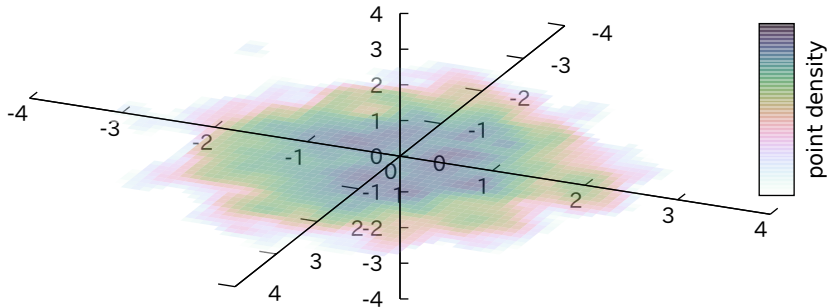
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



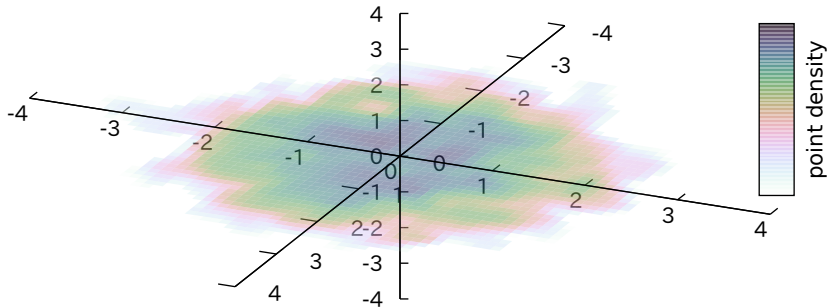
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



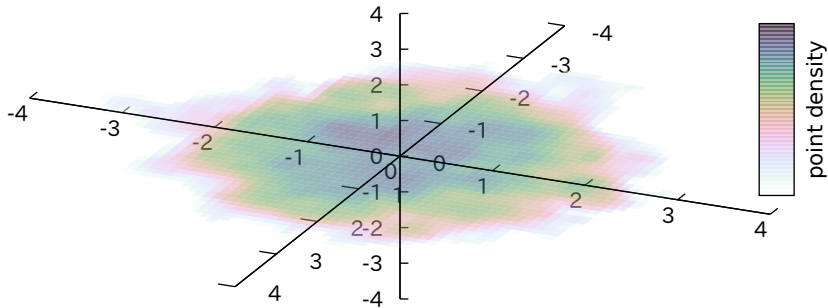
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



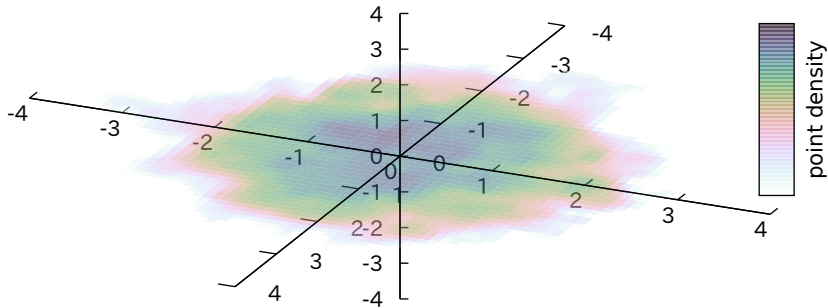
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



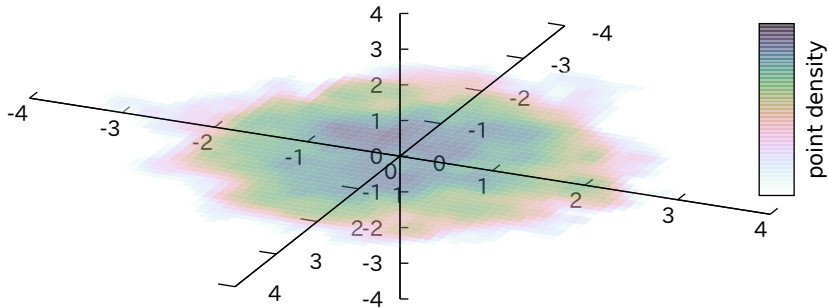
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



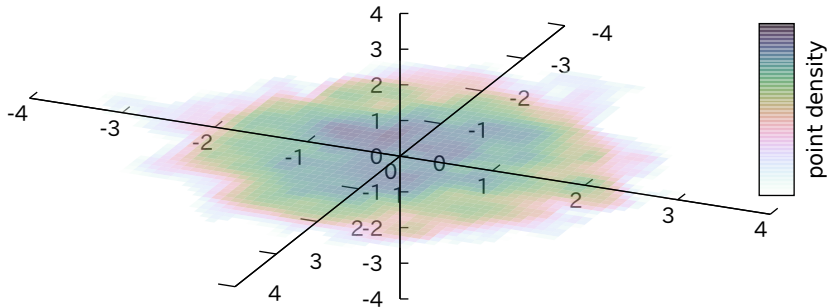
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



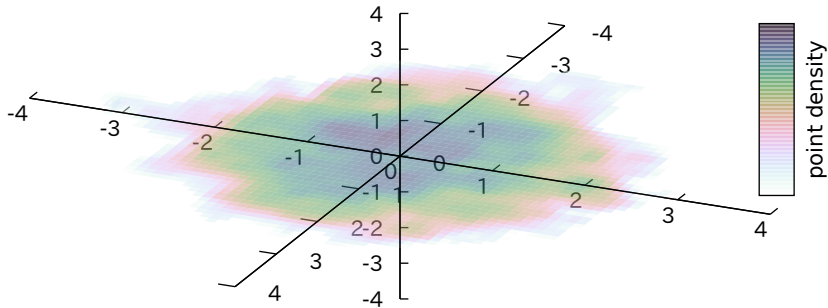
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



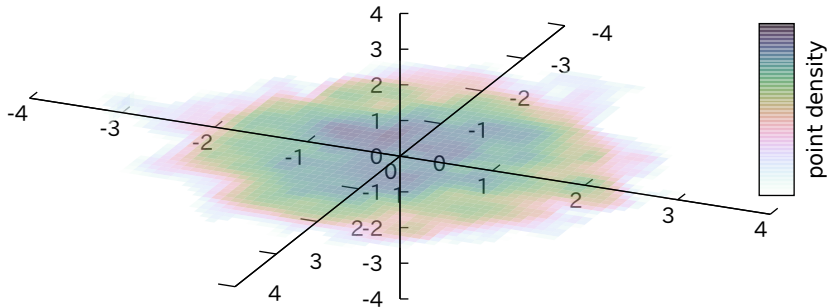
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



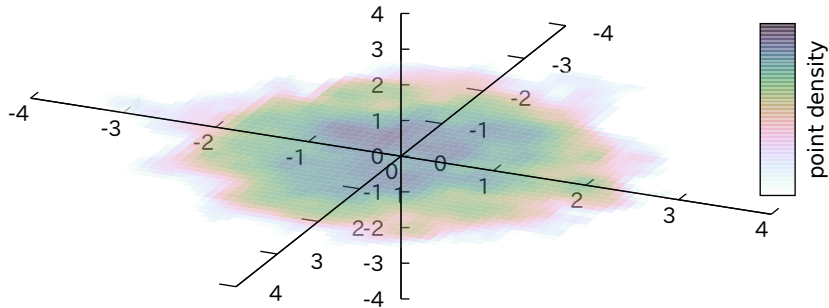
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



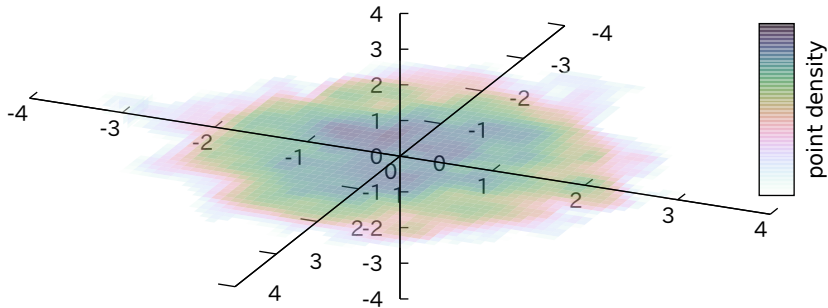
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



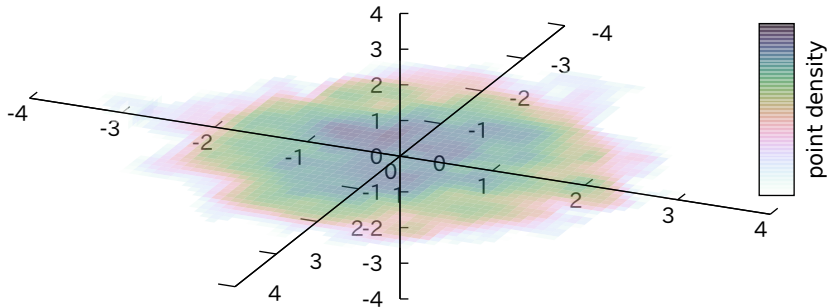
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



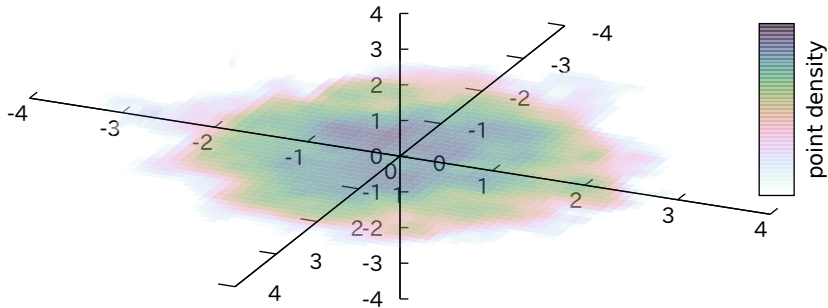
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



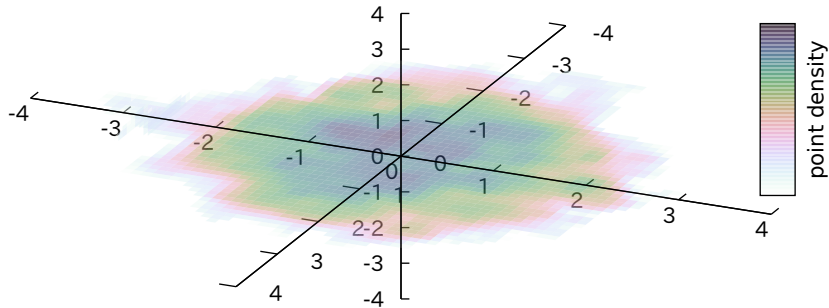
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



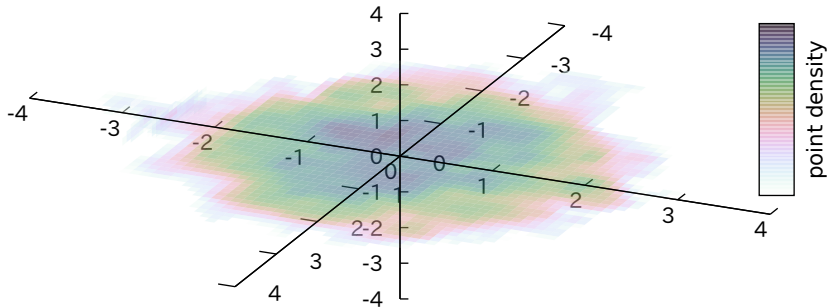
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



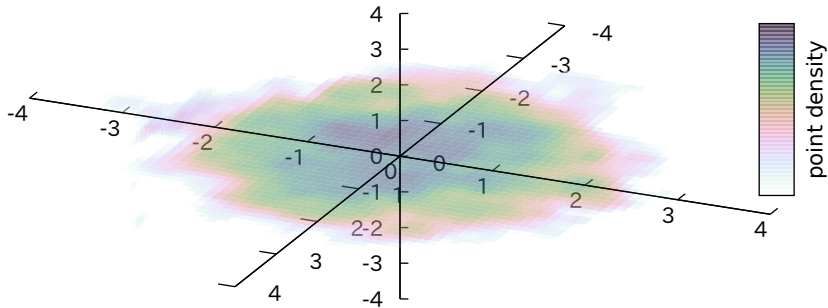
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



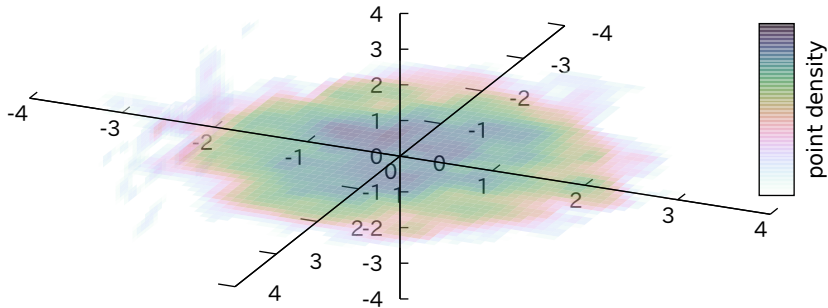
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



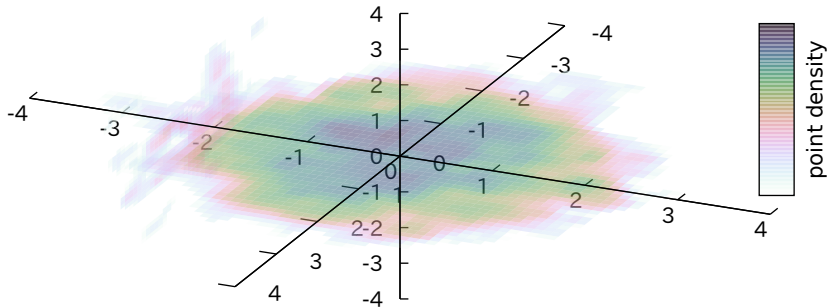
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



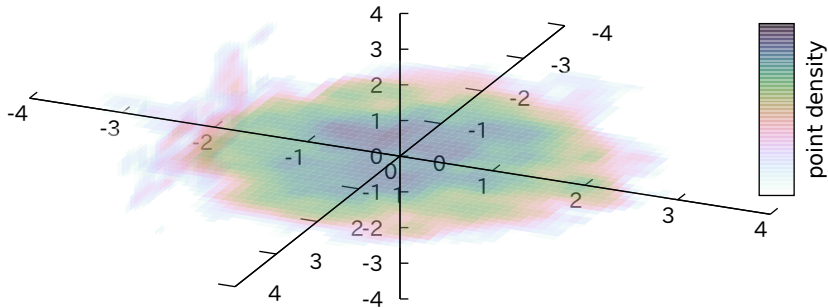
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



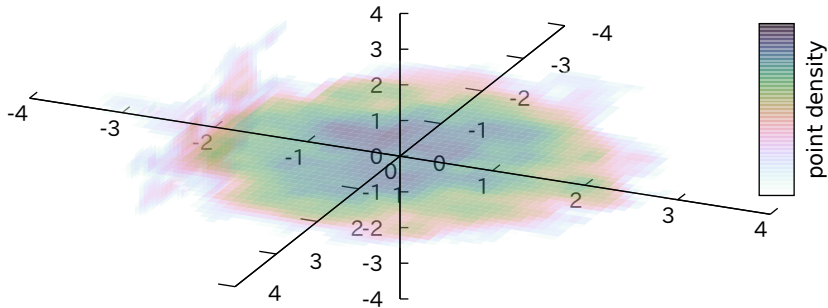
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



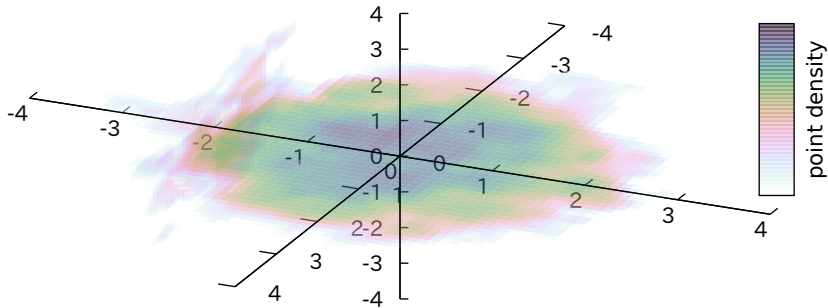
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



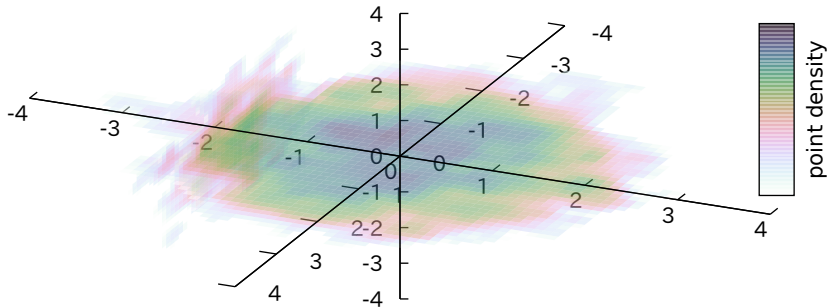
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



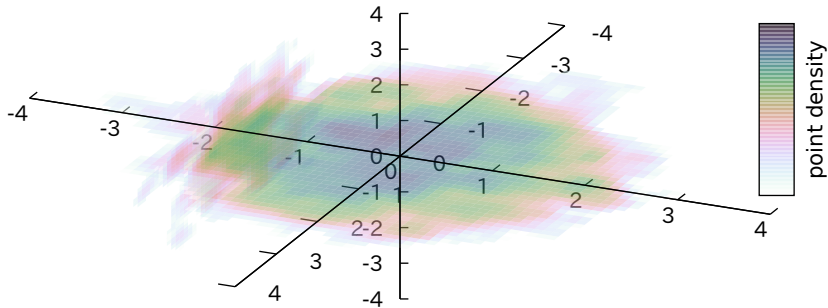
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



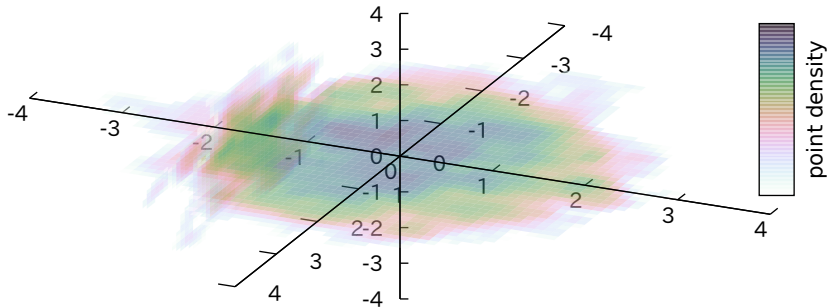
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



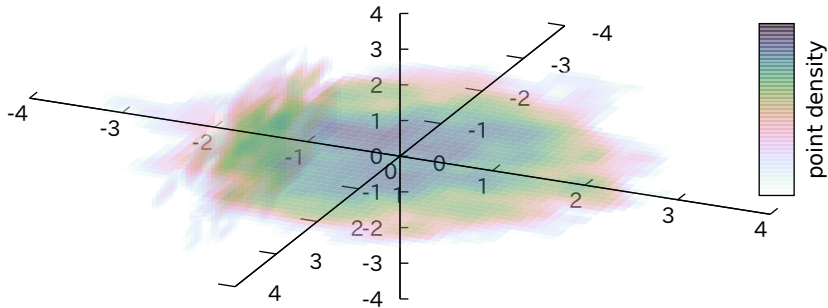
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



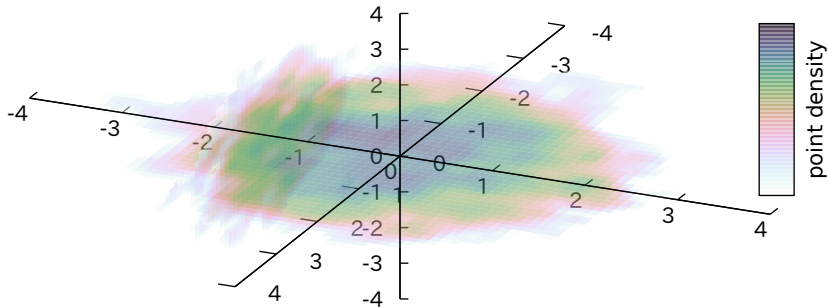
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



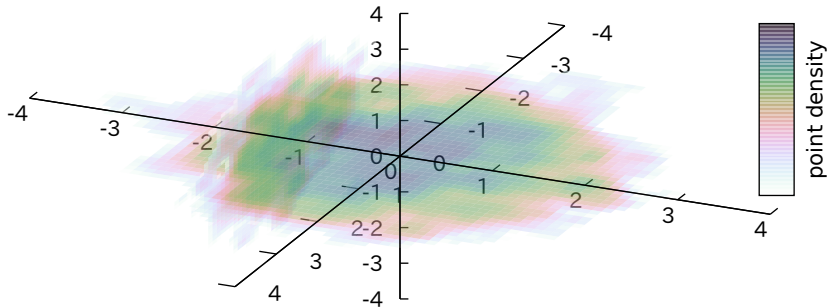
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



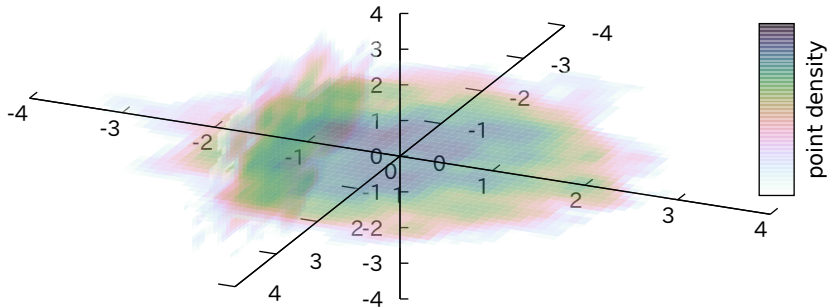
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



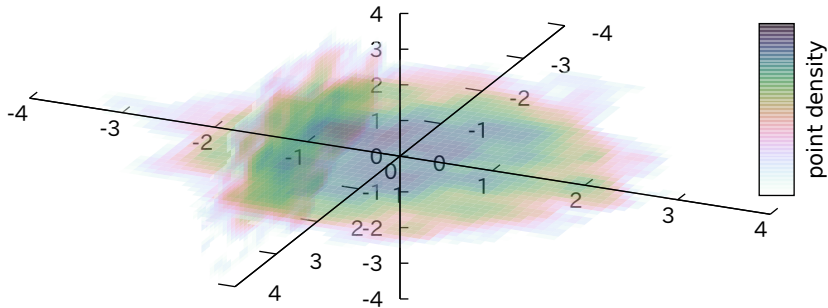
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



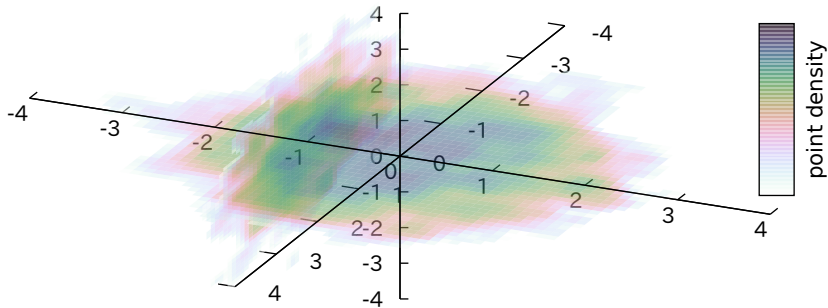
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



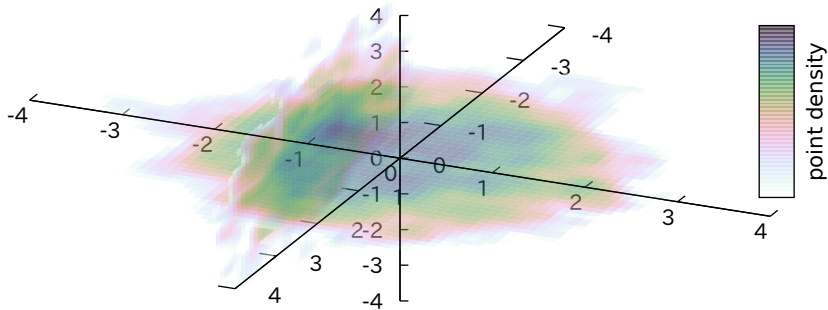
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



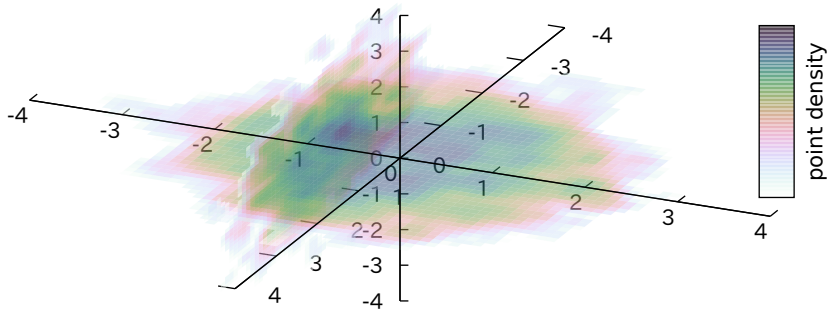
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



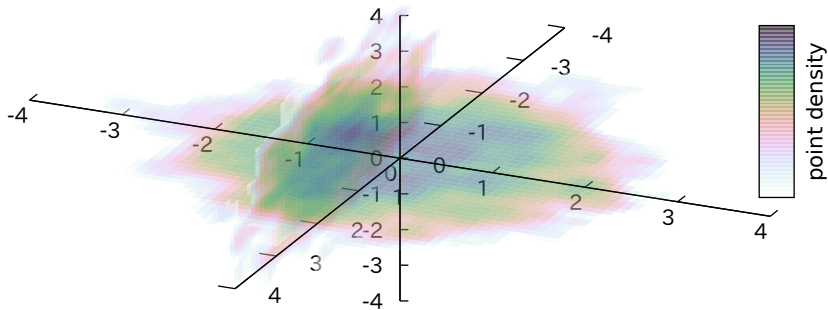
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



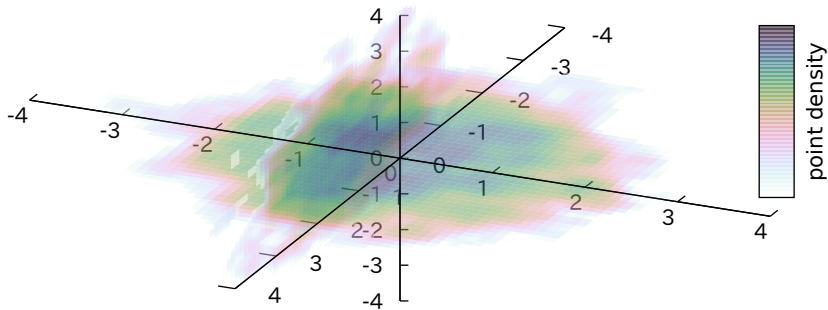
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



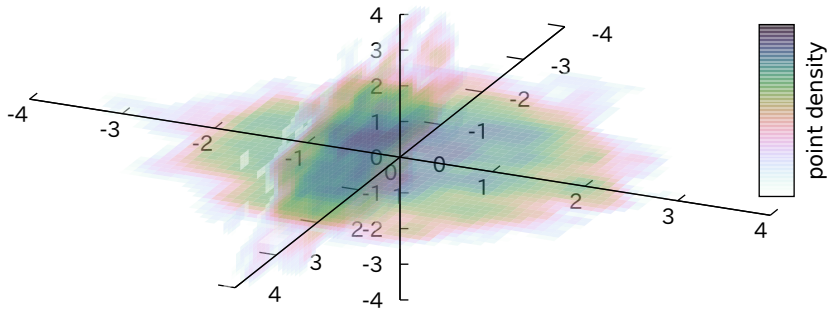
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



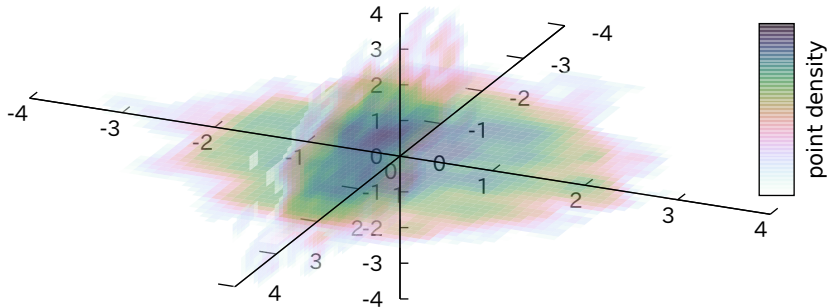
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



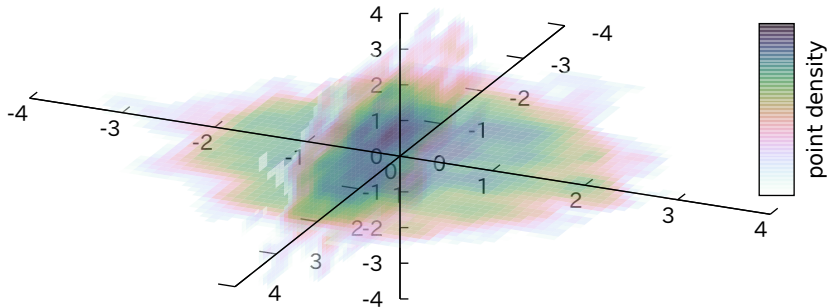
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



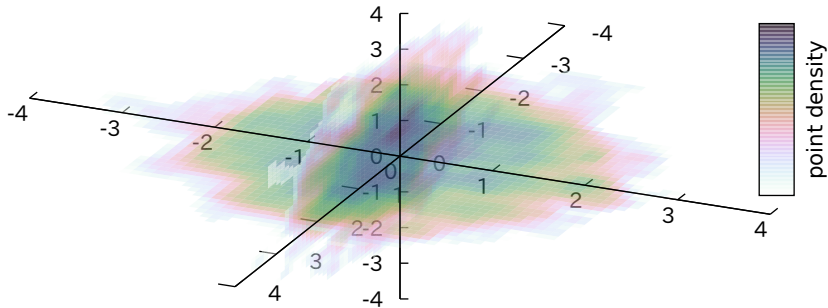
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



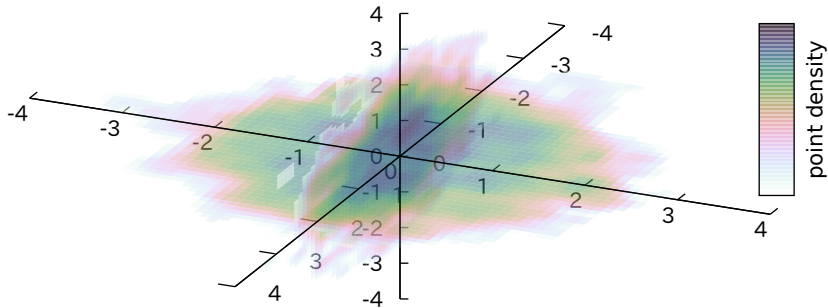
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



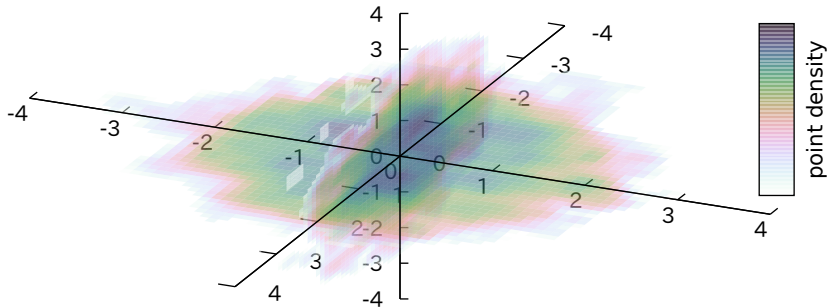
step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



step through volume using the density values to color a surface
plot '++' using 1:2:(z):(voxel(\$1,\$2,z)) with pm3d



orthogonal slices through volume
using the density values to color the surfaces

