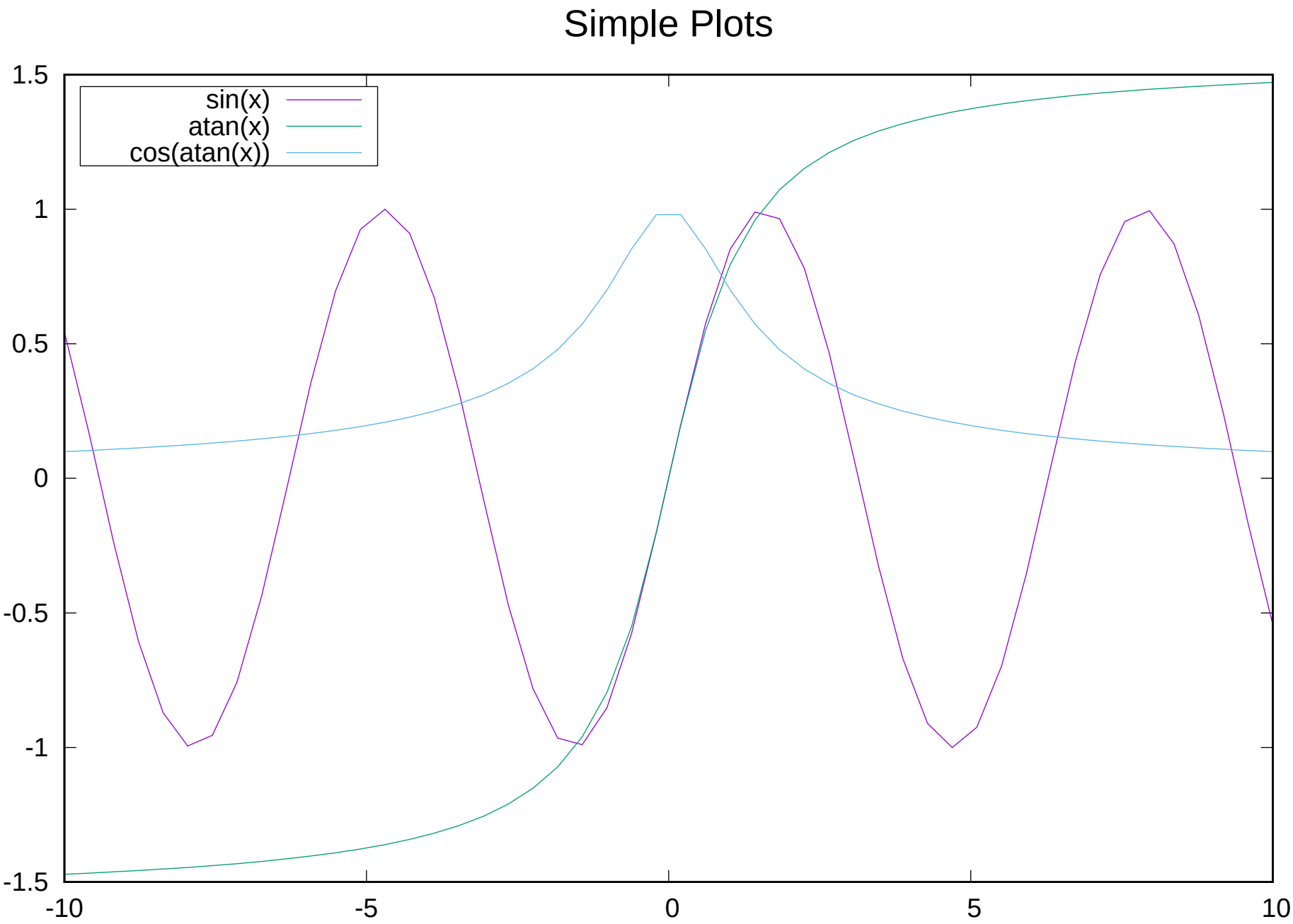
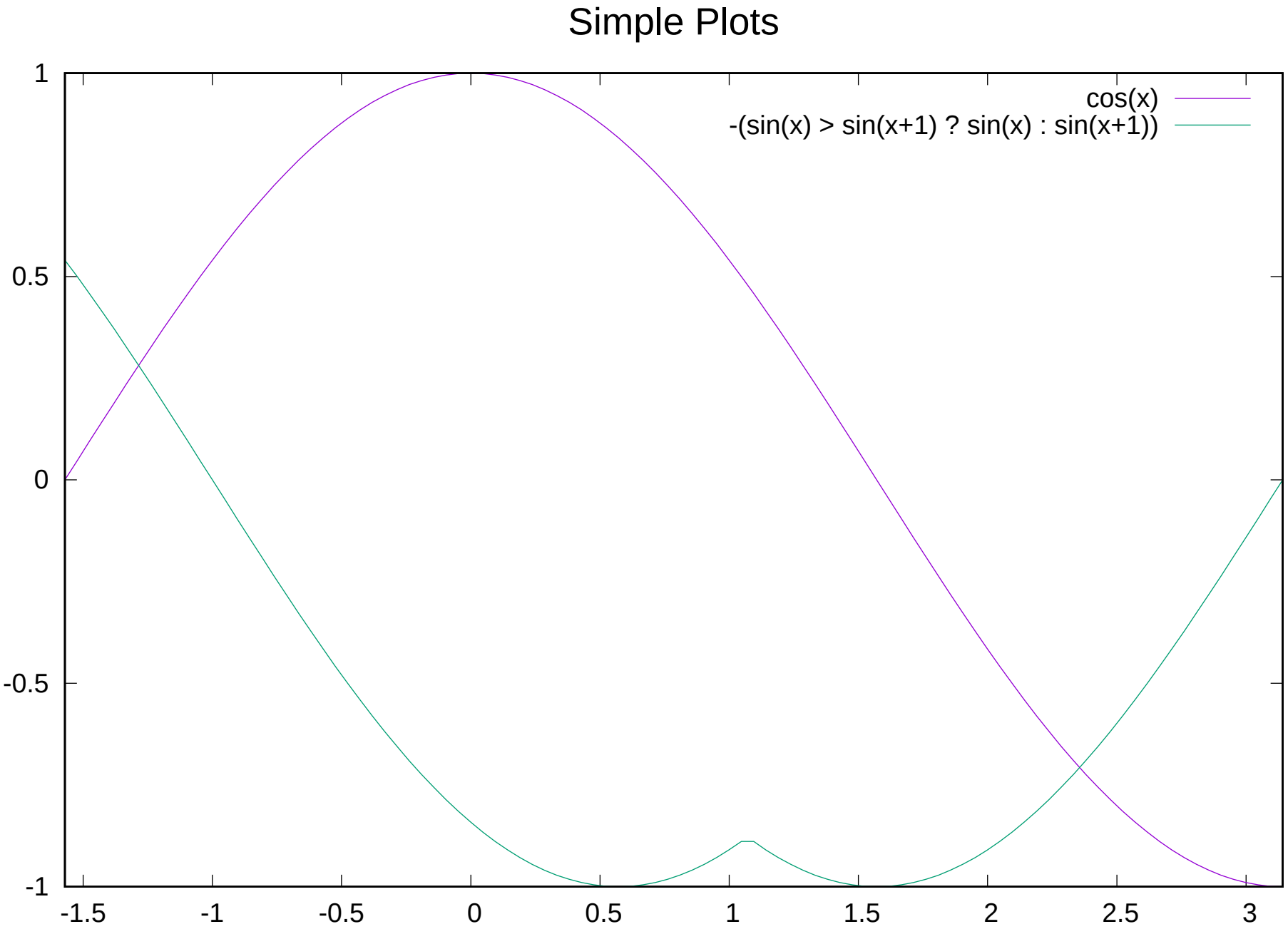
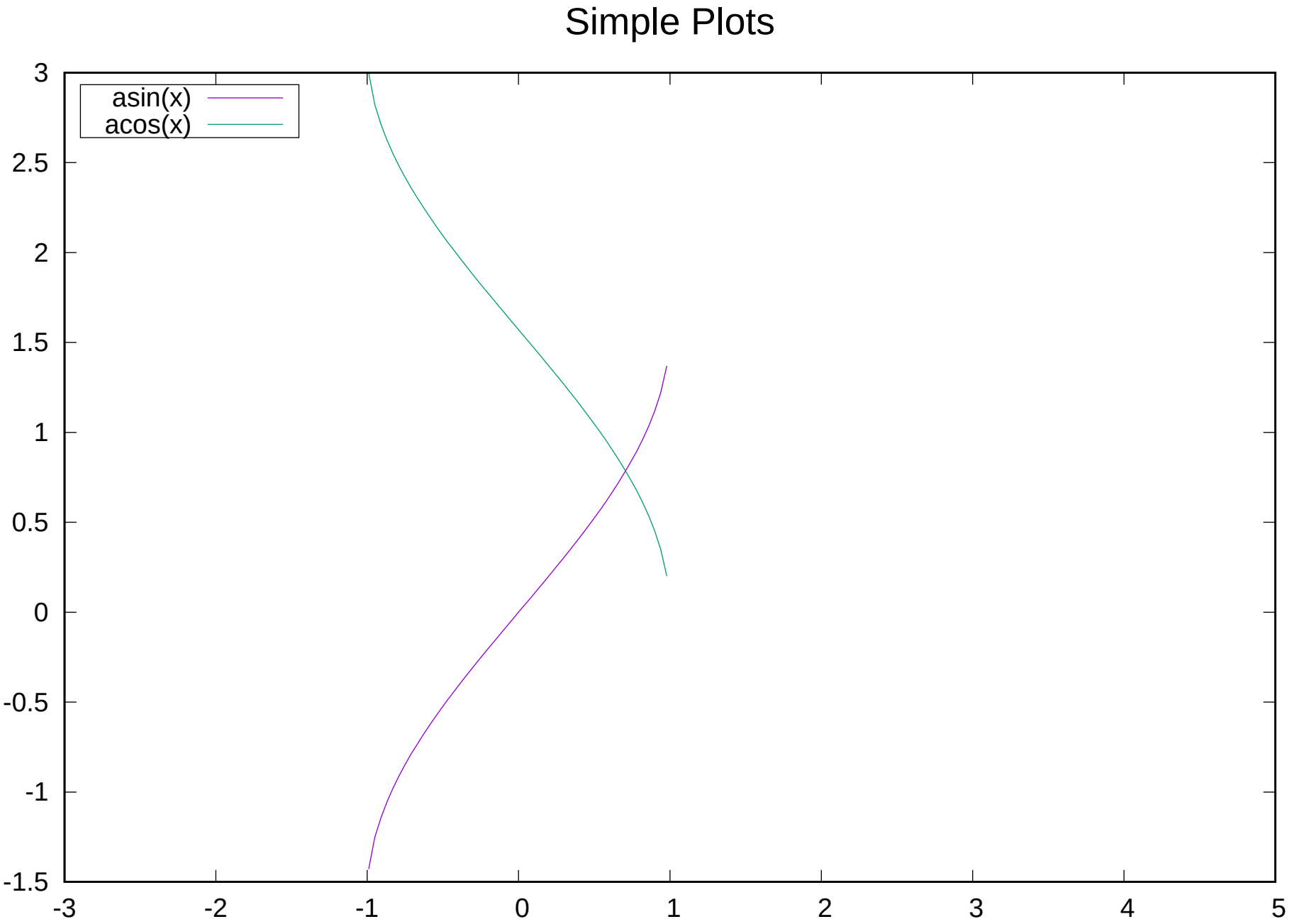
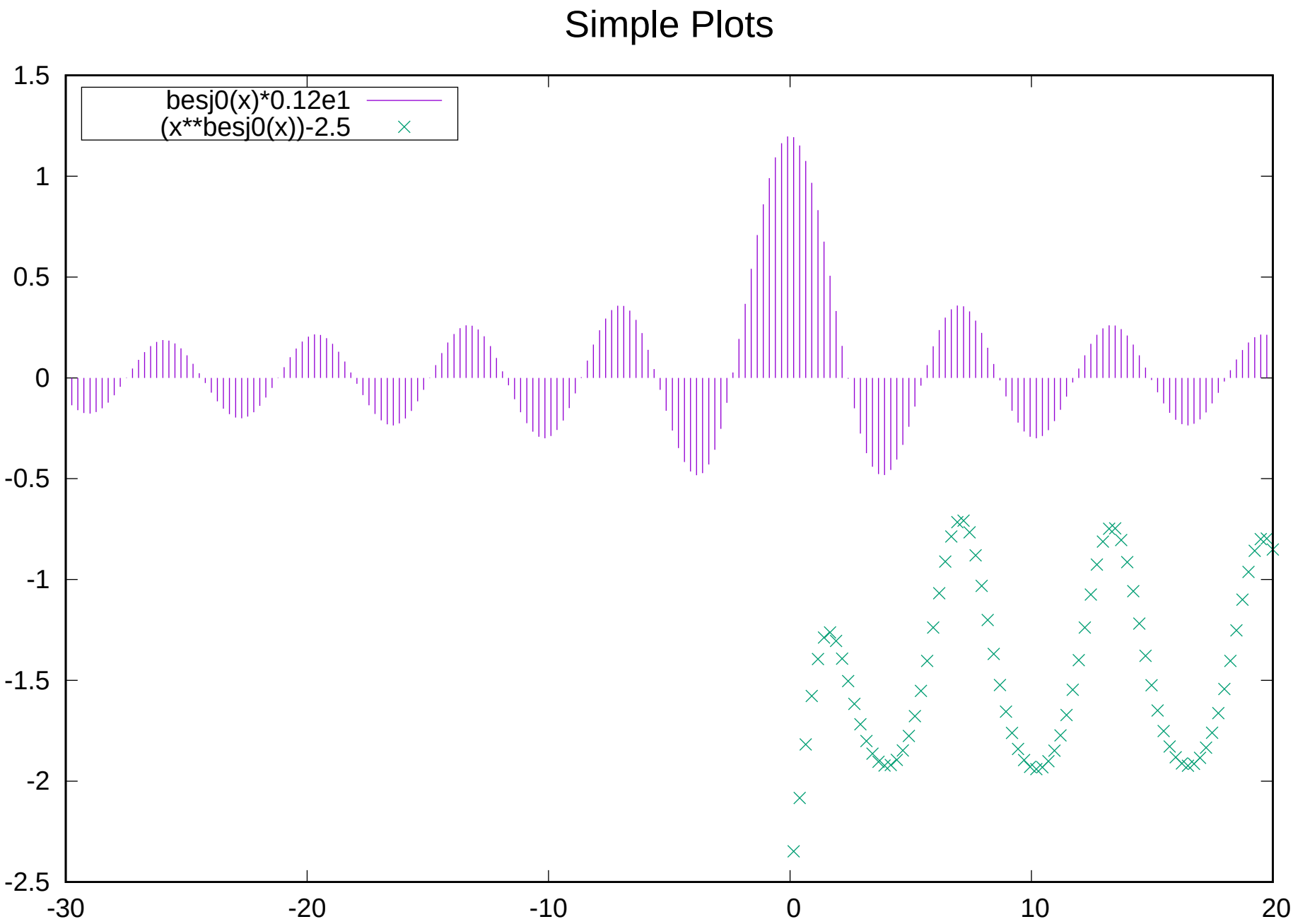
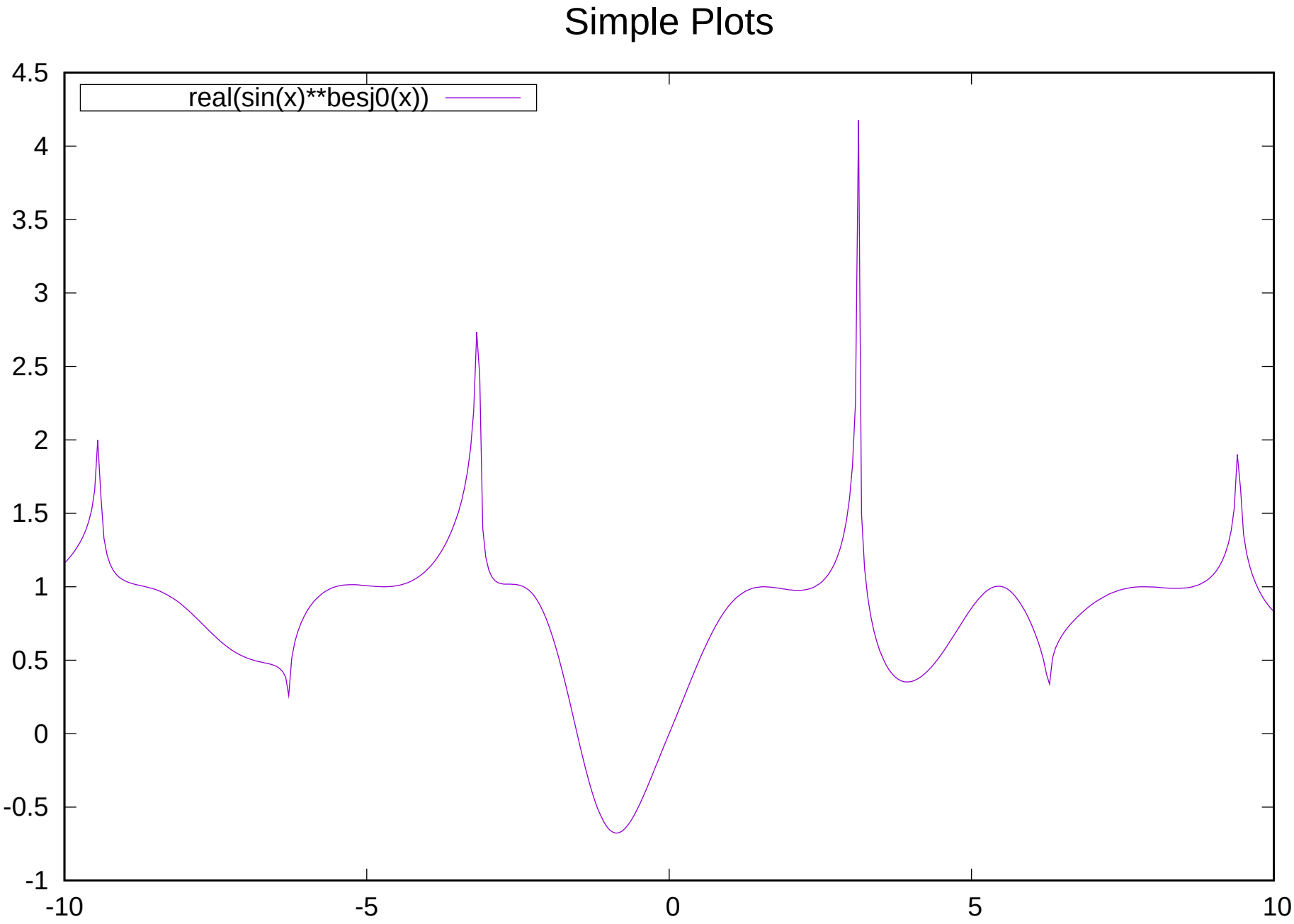
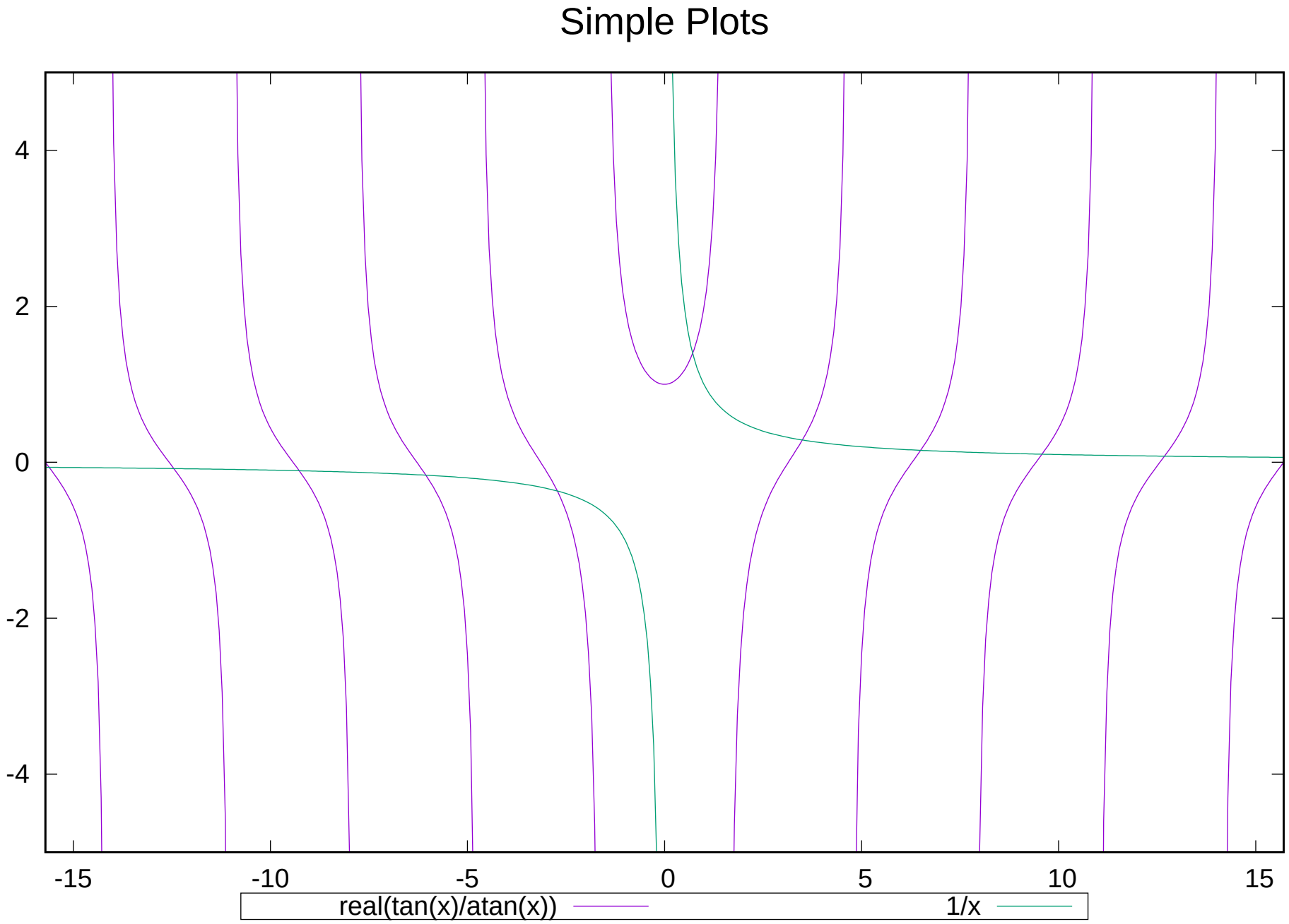

Simple Plots
sin(x)
atan(x)
cos(atan(x))
1.5
1
0.5
0
-0.5
-1
-1.5
-10
-5
0
5
10

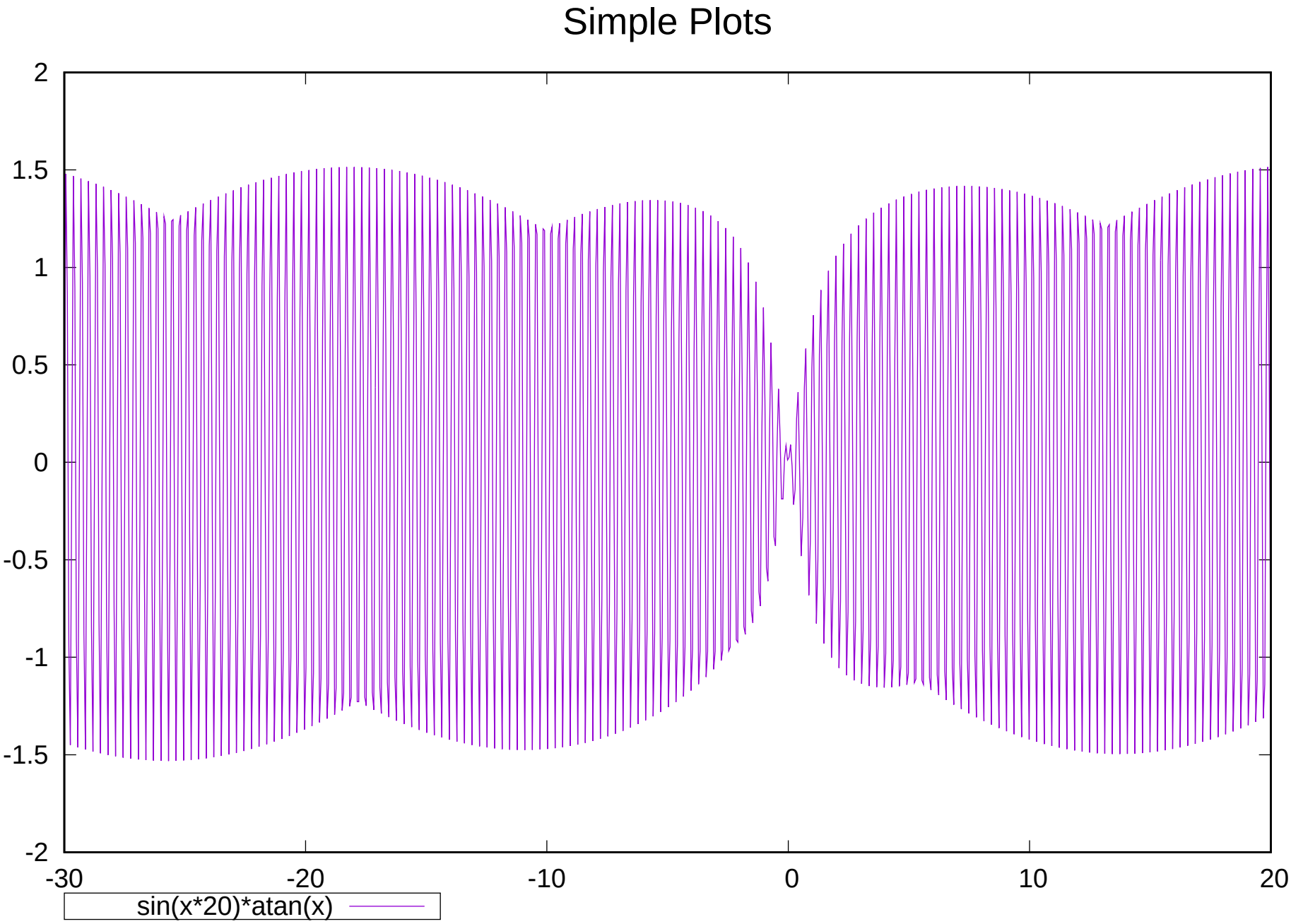

Simple Plots
cos(x)
-(sin(x) > sin(x+1) ? sin(x) : sin(x+1))
1
0.5
0
-0.5
-1
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3

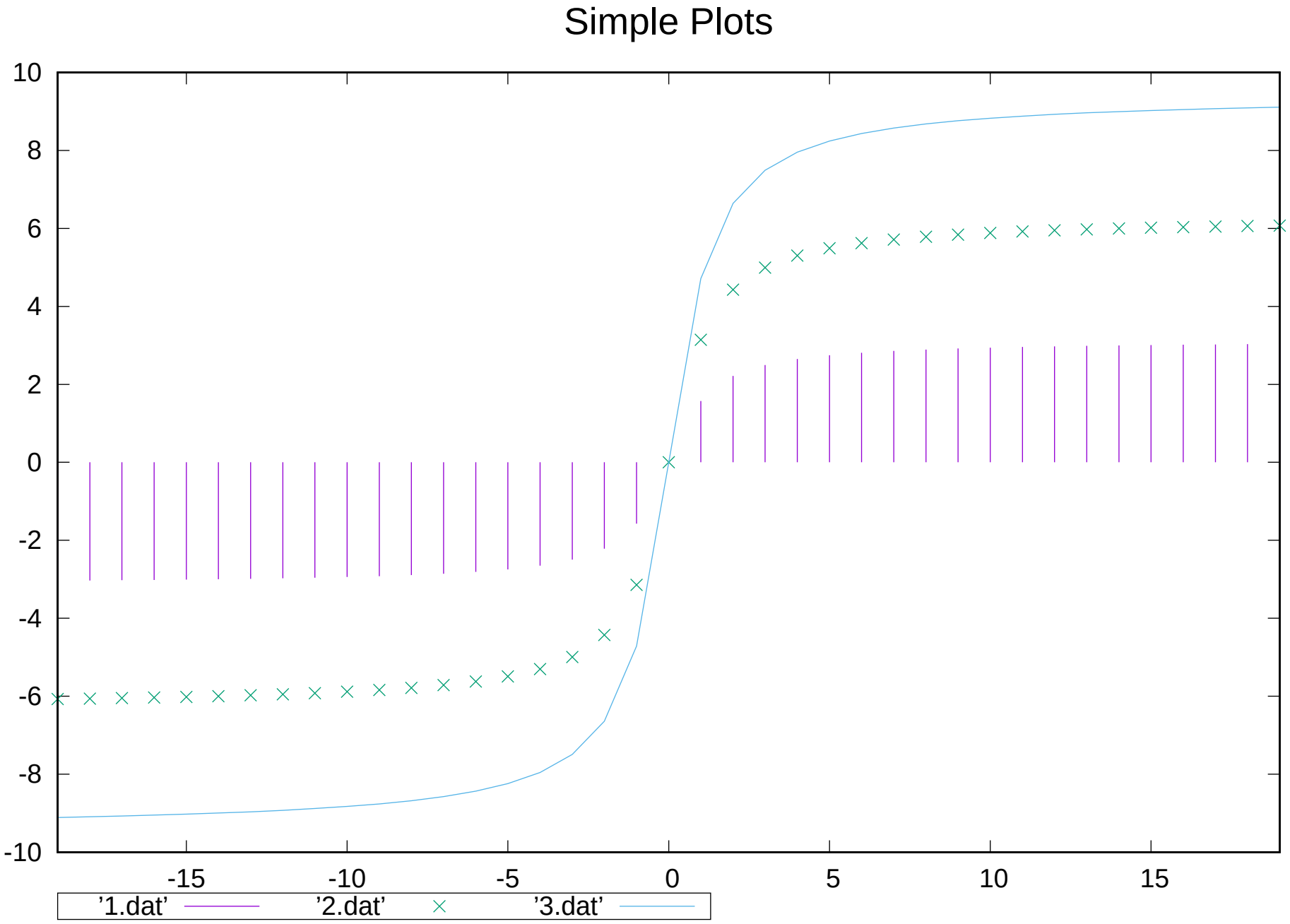

Simple Plots
asin(x)
acos(x)
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-3
-2
-1
0
1
2
3
4
5

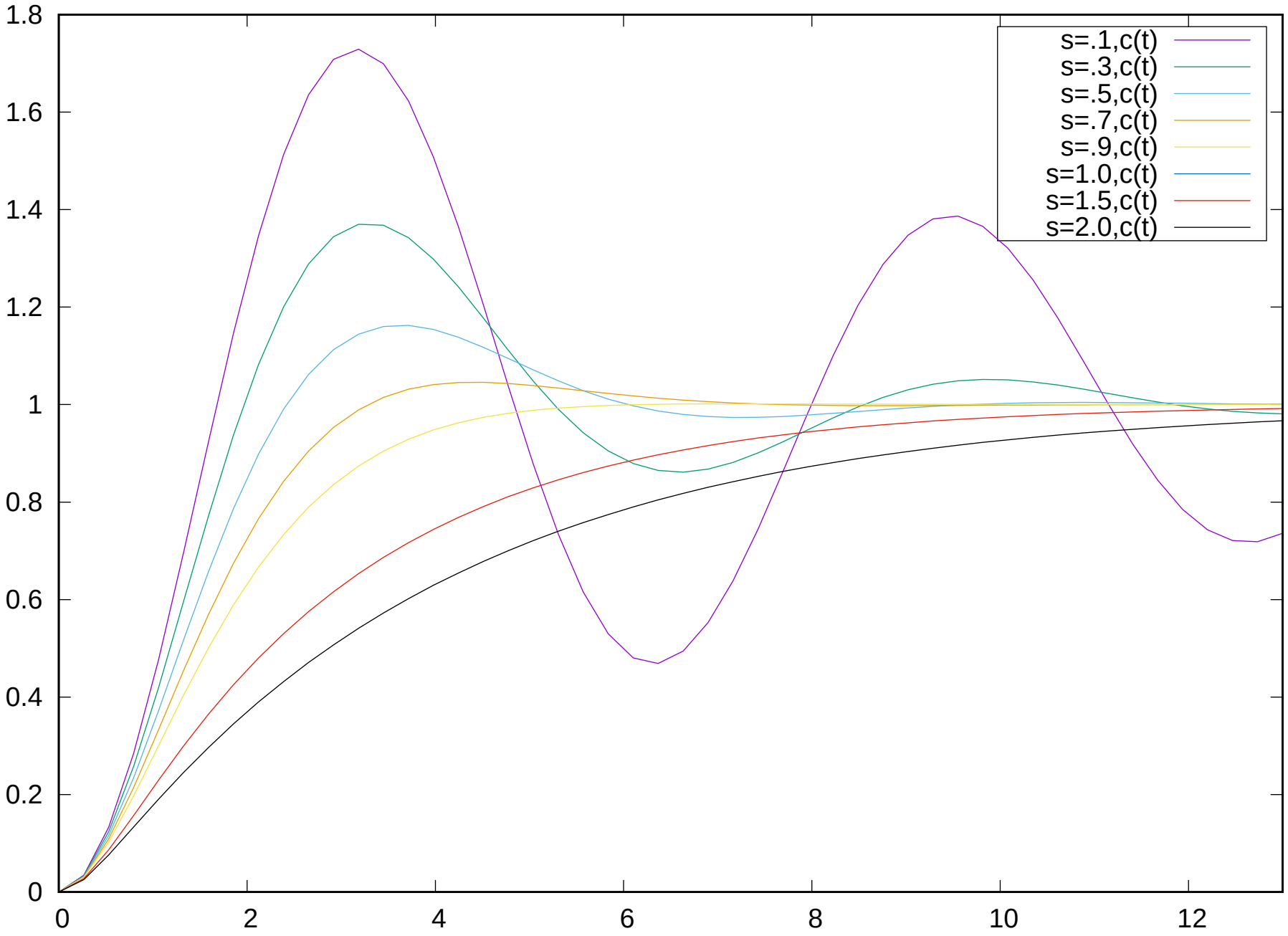

Simple Plots
besj0(x)*0.12e1
(x**besj0(x))-2.5
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-30
-20
-10
0
10
20

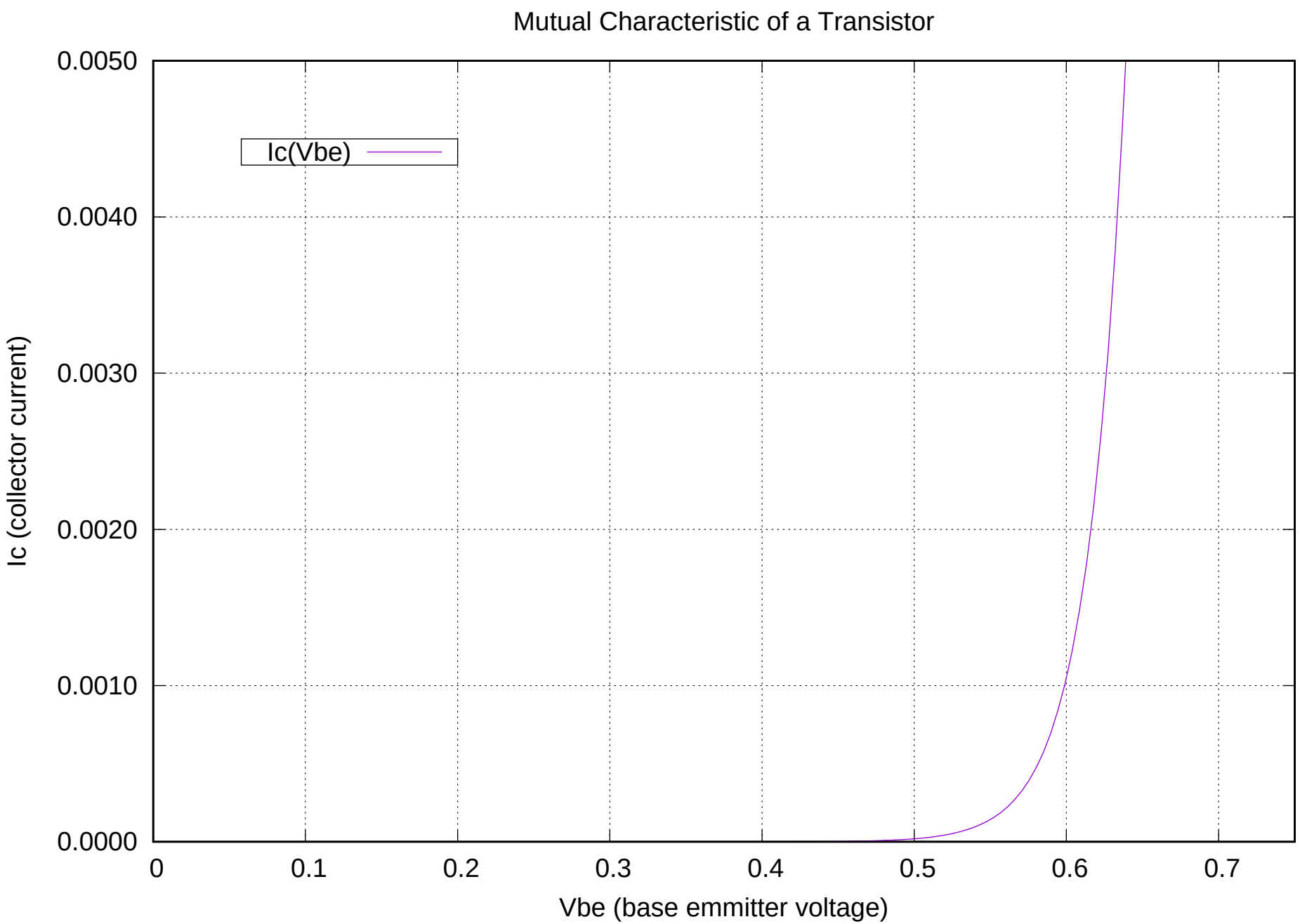

Simple Plots
real(sin(x)**besj0(x))
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-10
-5
0
5
10

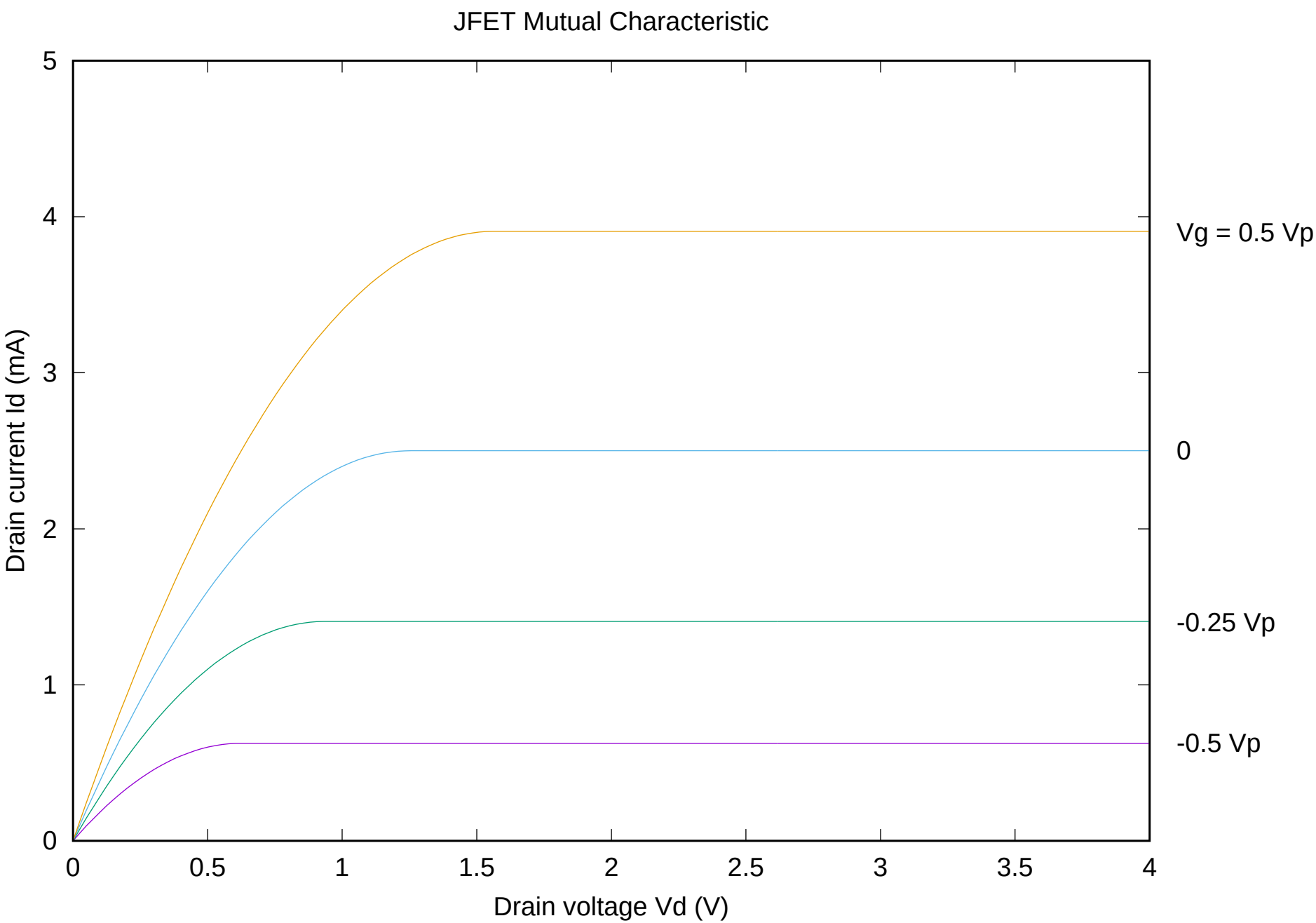

Simple Plots
4
2
0
-2
-4
-15
-10
-5
0
5
10
15
real(tan(x)/atan(x))
1/x

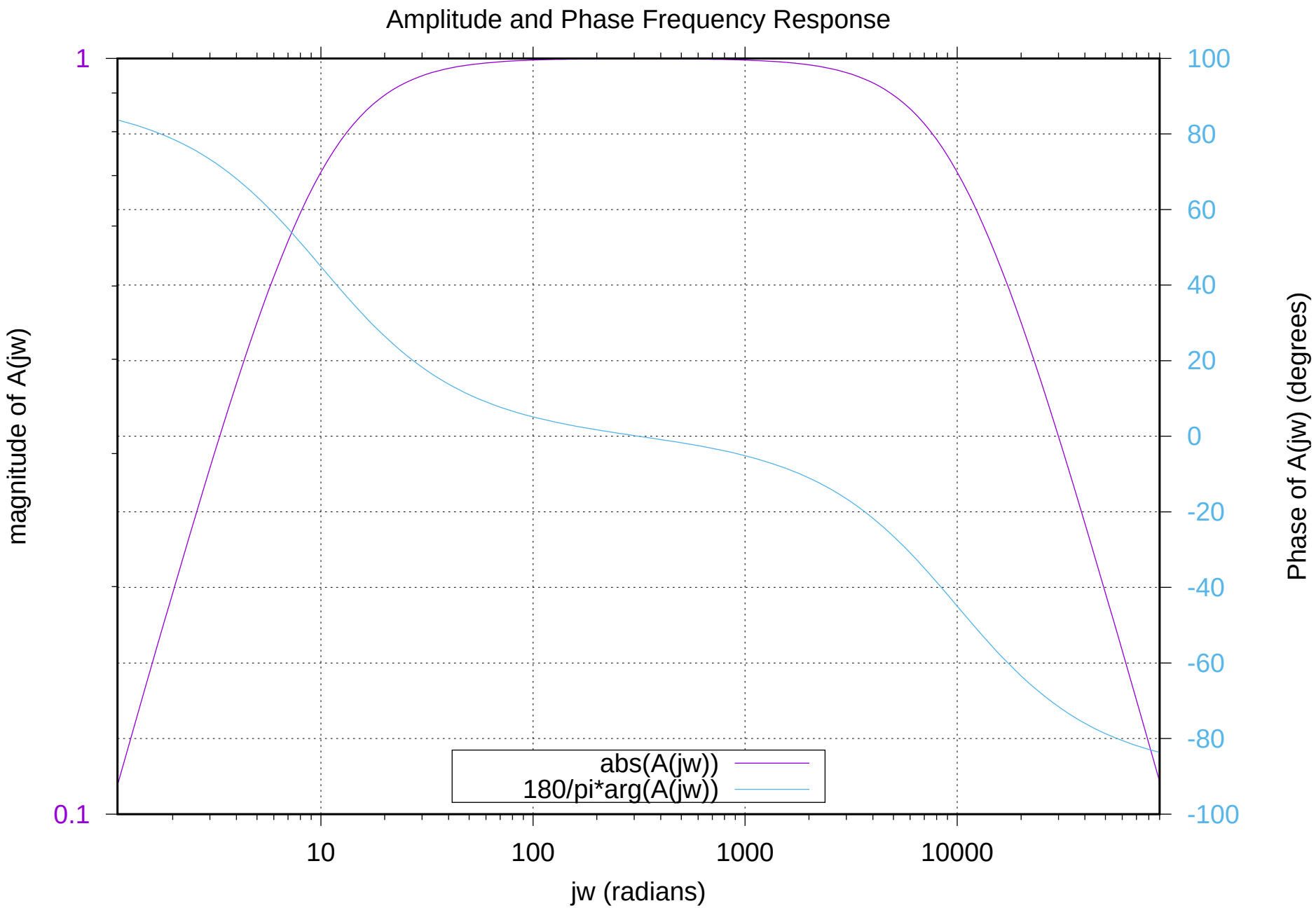

Simple Plots
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-30
-20
-10
0
10
20
sin(x*20)*atan(x)

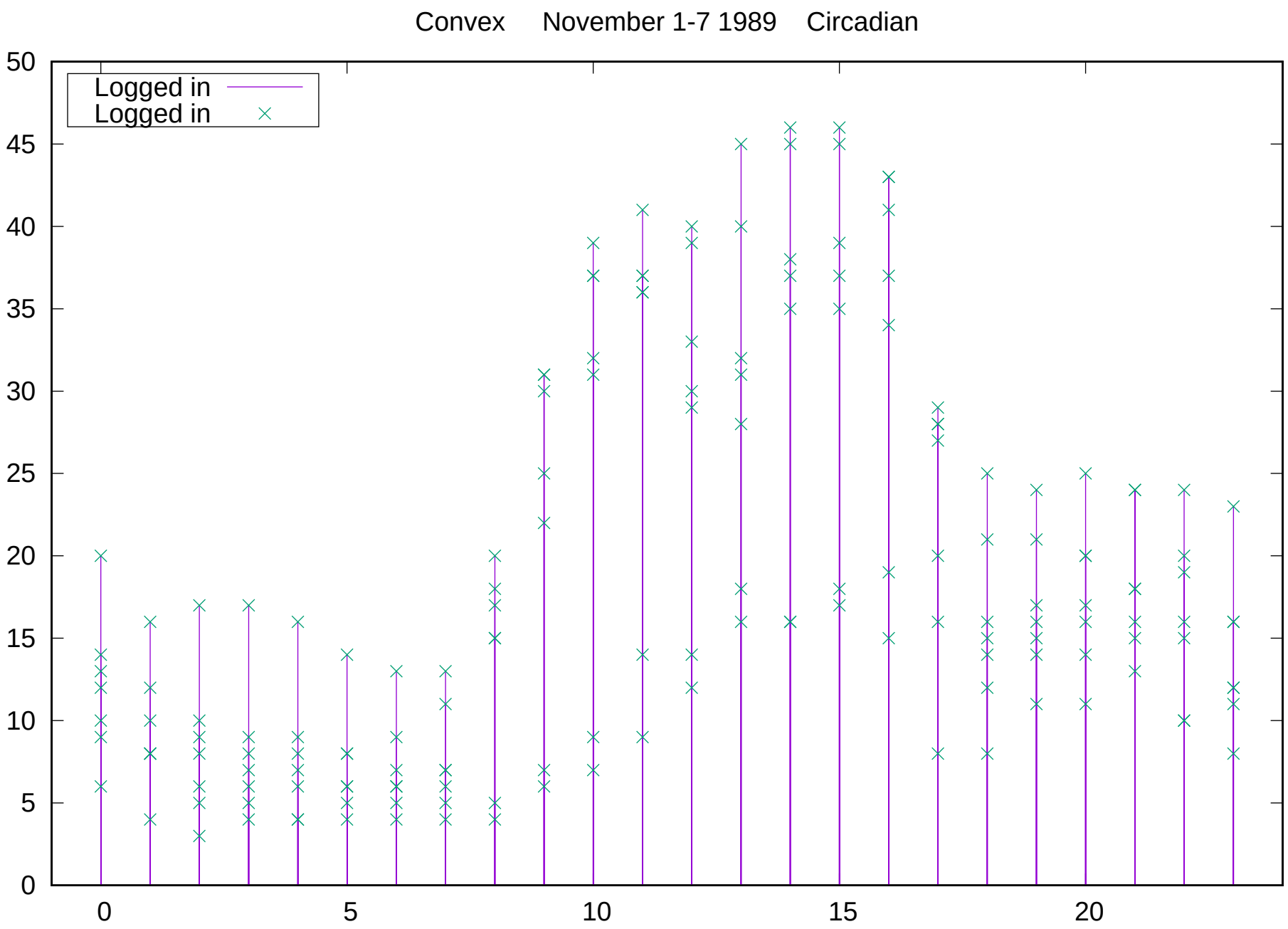

Simple Plots
10
8
6
4
2
0
-2
-4
-6
-8
-10
-15
-10
-5
0
5
10
15
'1.dat'
'2.dat'
'3.dat'

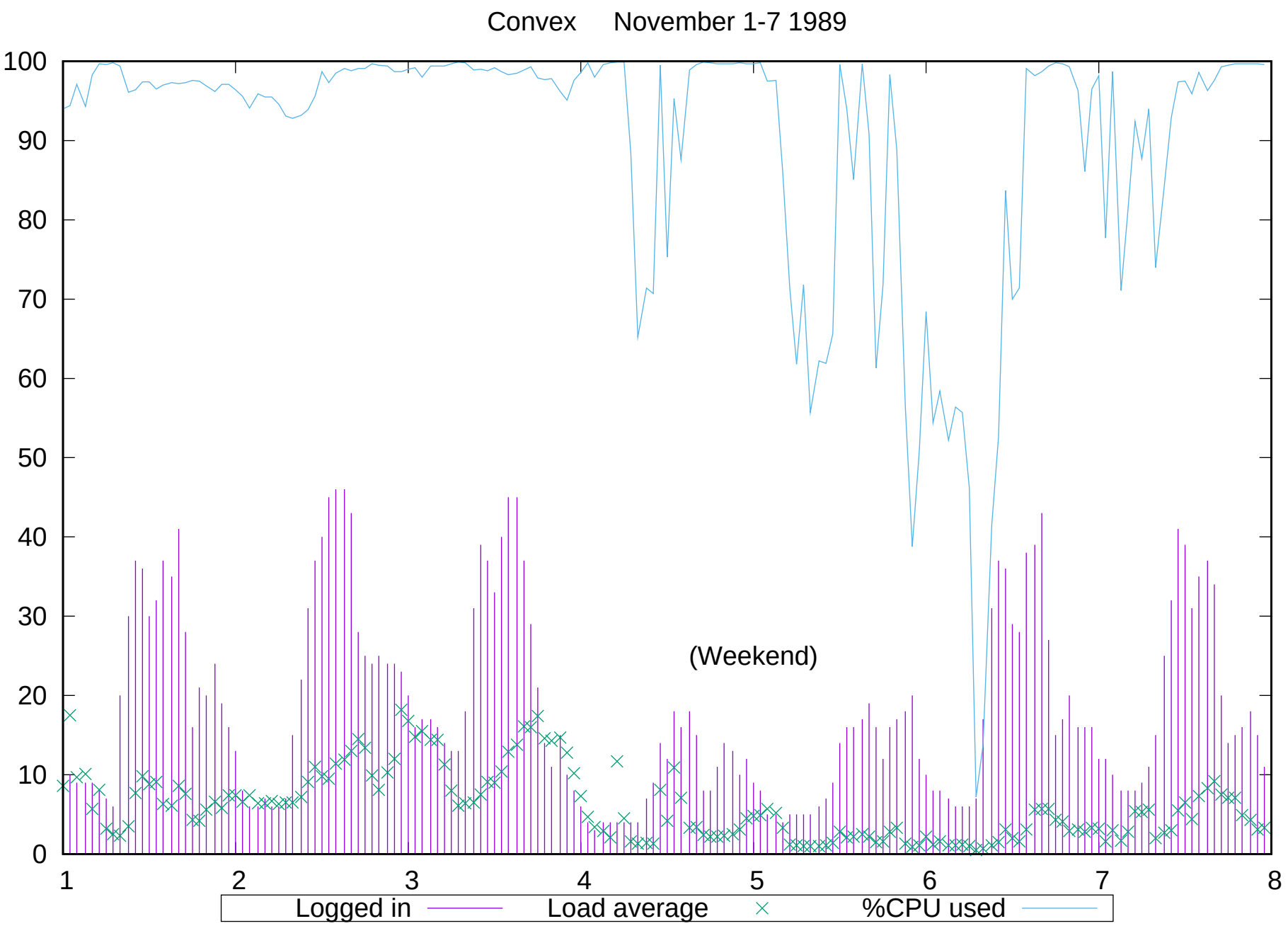

1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
12
s=.1,c(t)
s=.3,c(t)
s=.5,c(t)
s=.7,c(t)
s=.9,c(t)
s=1.0,c(t)
s=1.5,c(t)
s=2.0,c(t)


Mutual Characteristic of a Transistor
Ic(Vbe)
Ic (collector current)
Vbe (base emmitter voltage)
0.0050
0.0040
0.0030
0.0020
0.0010
0.0000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

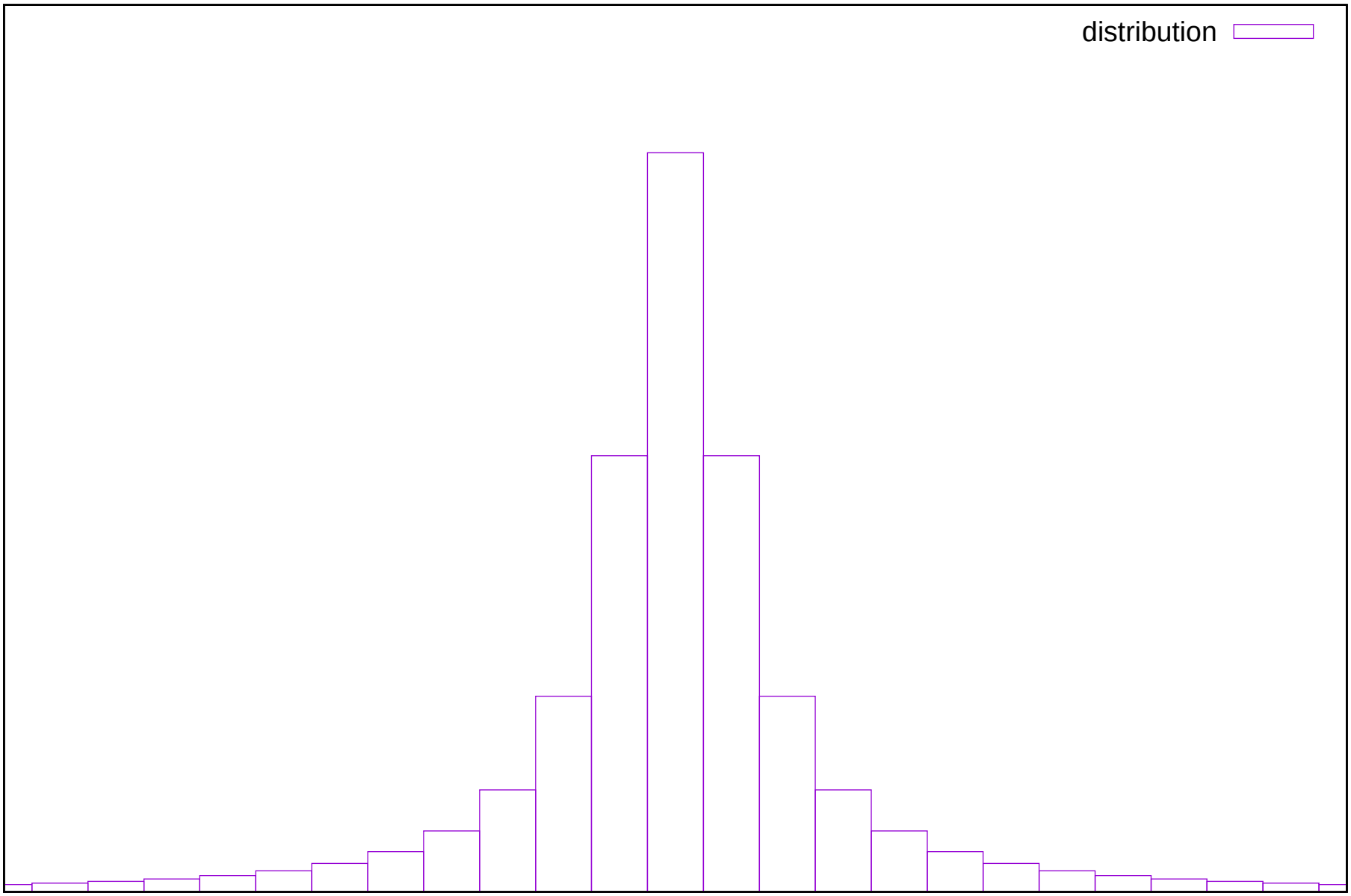

JFET Mutual Characteristic
Drain current Id (mA)
Drain voltage Vd (V)
0
1
2
3
4
5
0
0.5
1
1.5
2
2.5
3
3.5
4
Vg = 0.5 Vp
0
-0.25 Vp
-0.5 Vp


Amplitude and Phase Frequency Response
1
0.1
magnitude of A(jw)
100
80
60
40
20
0
-20
-40
-60
-80
-100
Phase of A(jw) (degrees)
10
100
1000
10000
jw (radians)
abs(A(jw))
180/pi*arg(A(jw))

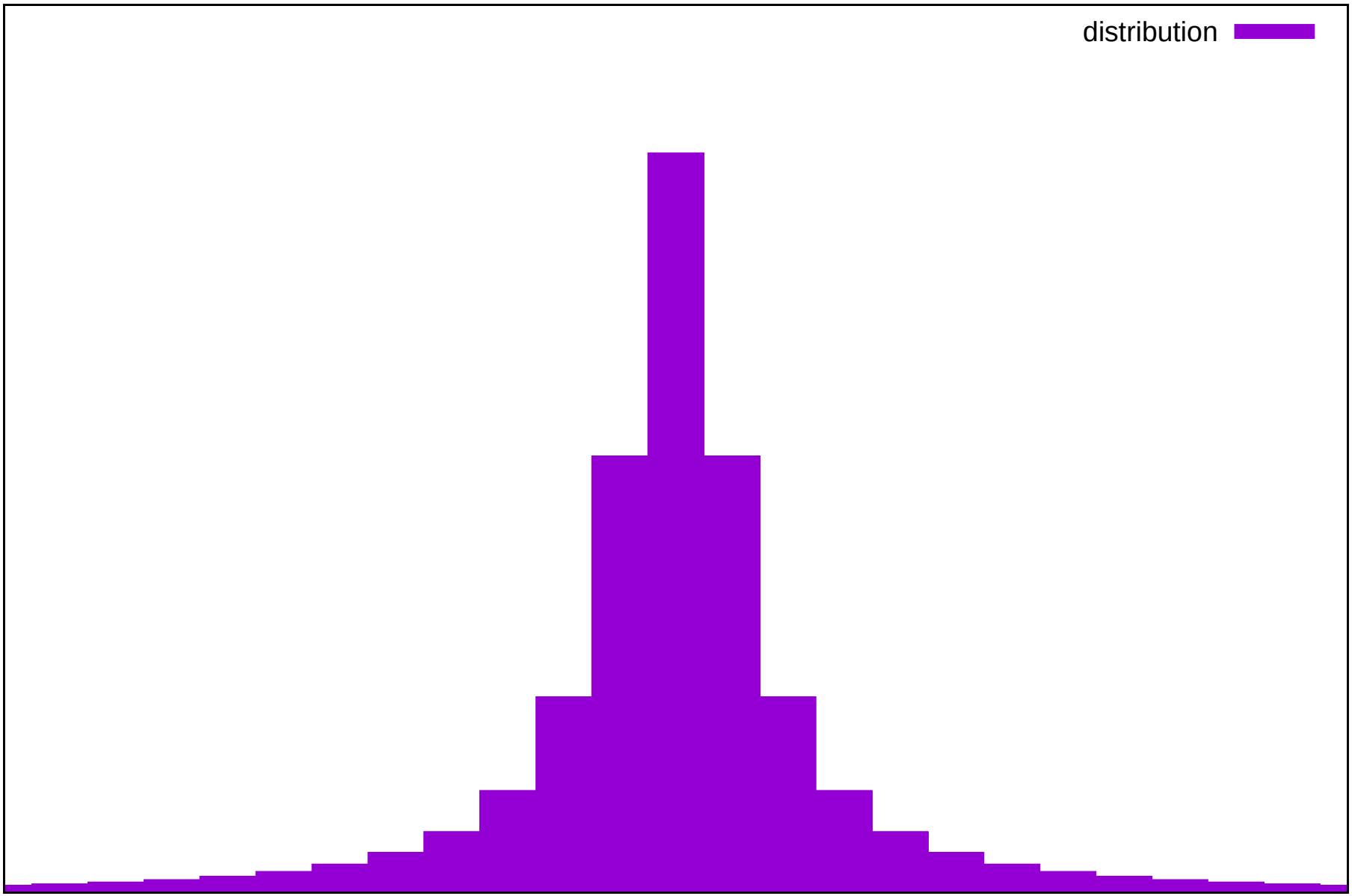

Convex November 1-7 1989 Circadian
Logged in
Logged in
50
45
40
35
30
25
20
15
10
5
0
0
5
10
15
20


Convex November 1-7 1989
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
(Weekend)
Logged in
Load average
%CPU used

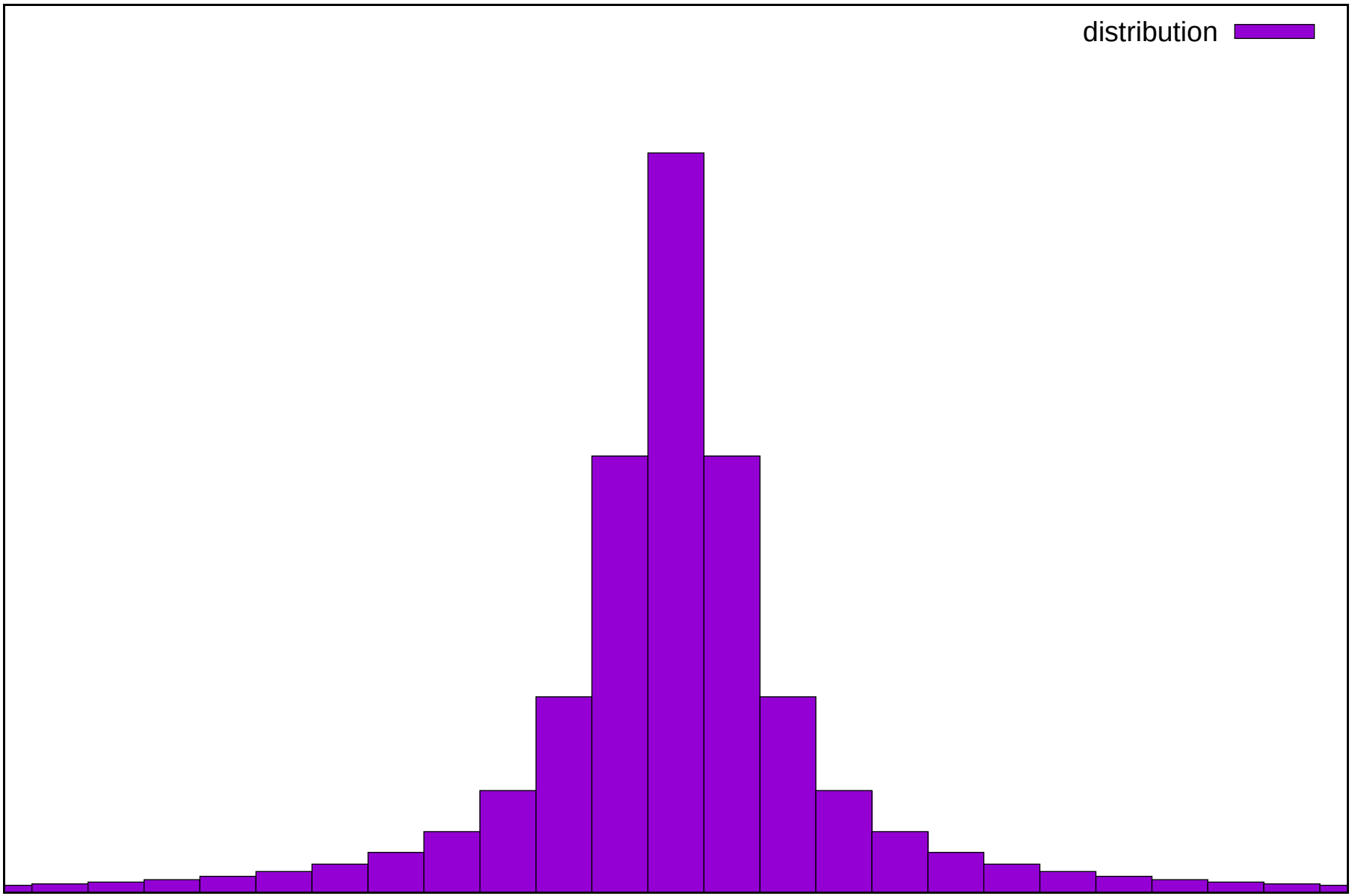
A demonstration of boxes with default properties



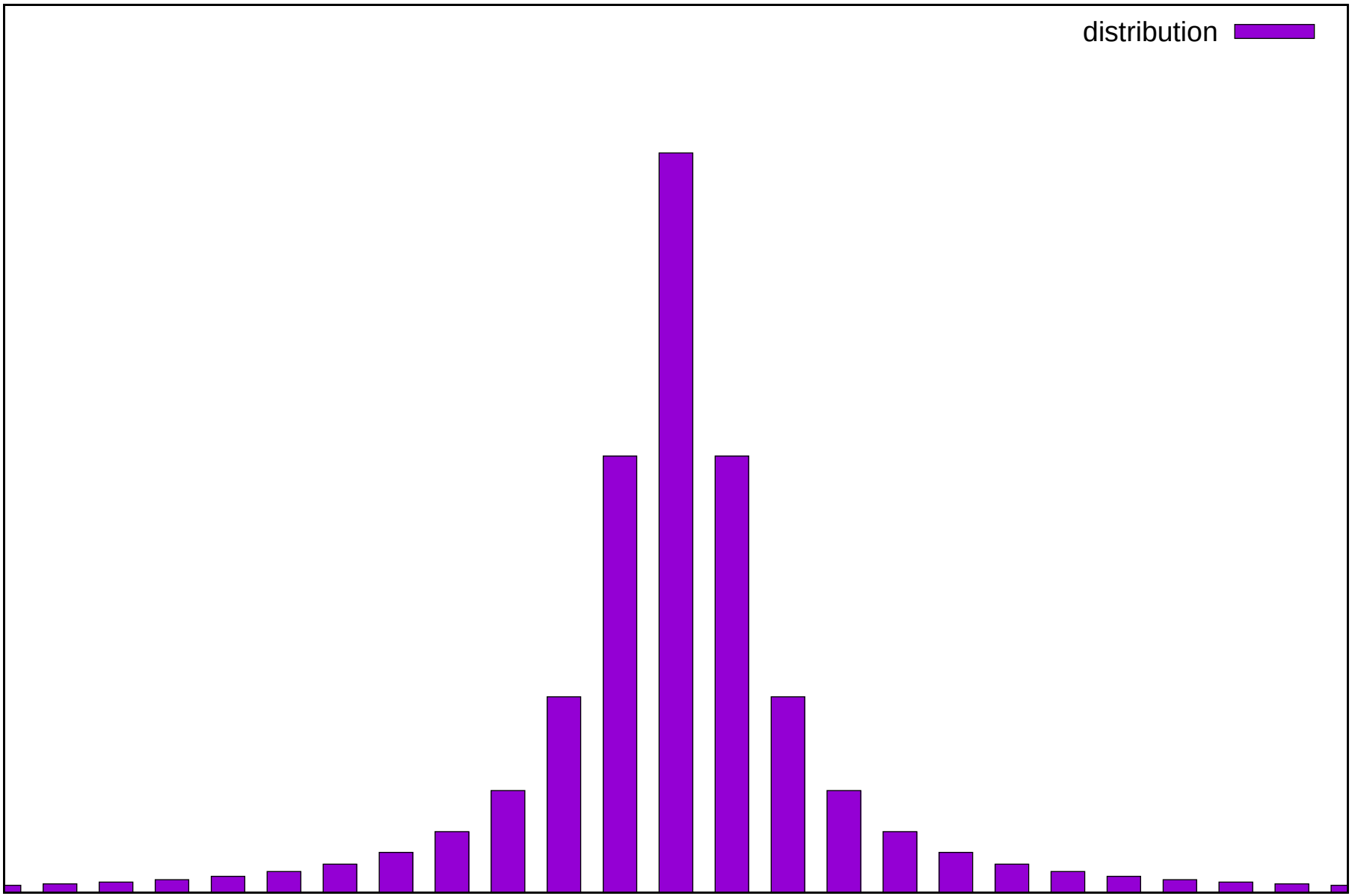
A demonstration of boxes with style fill solid 1.0



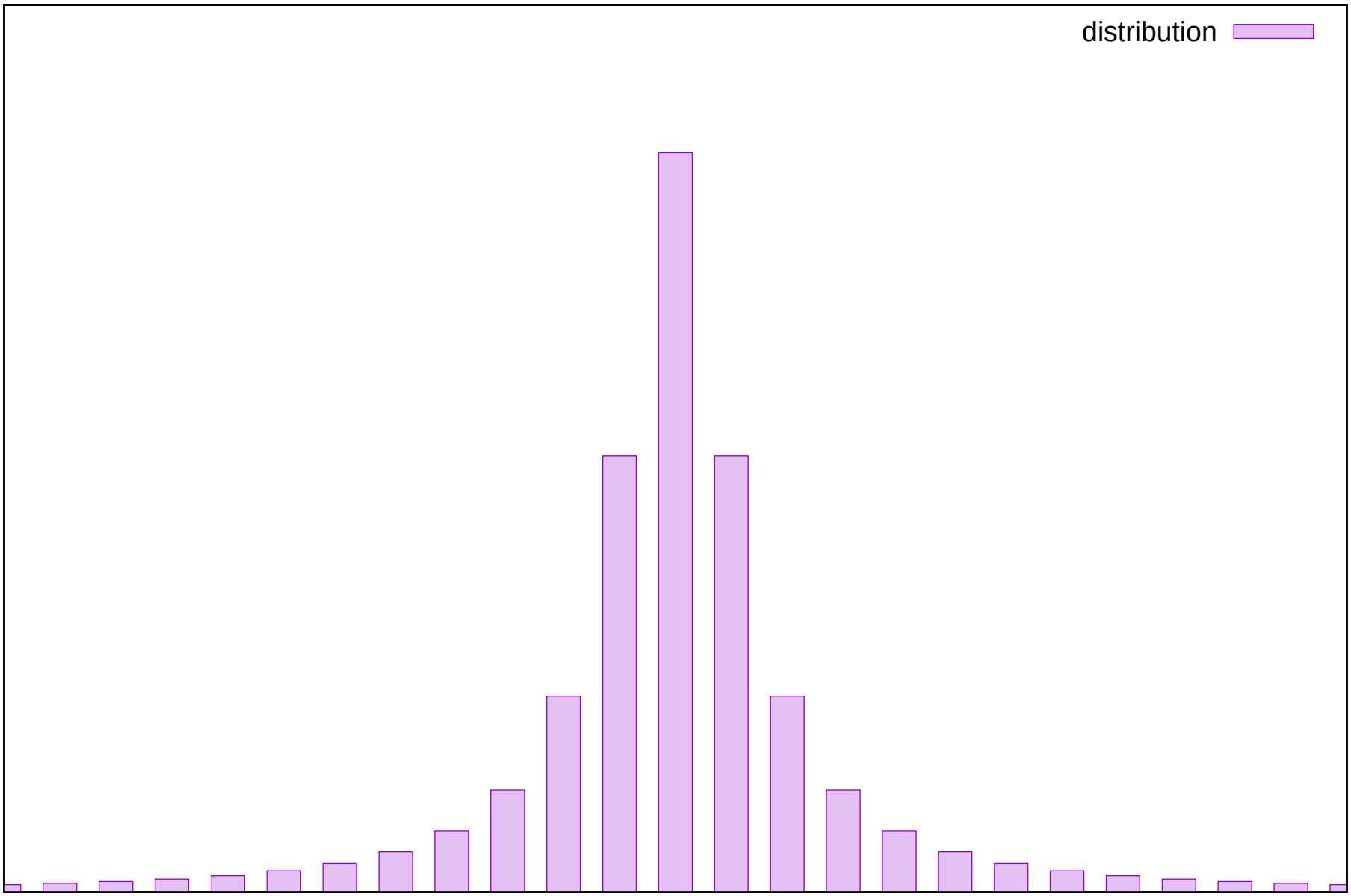
A demonstration of boxes with style fill solid border -1



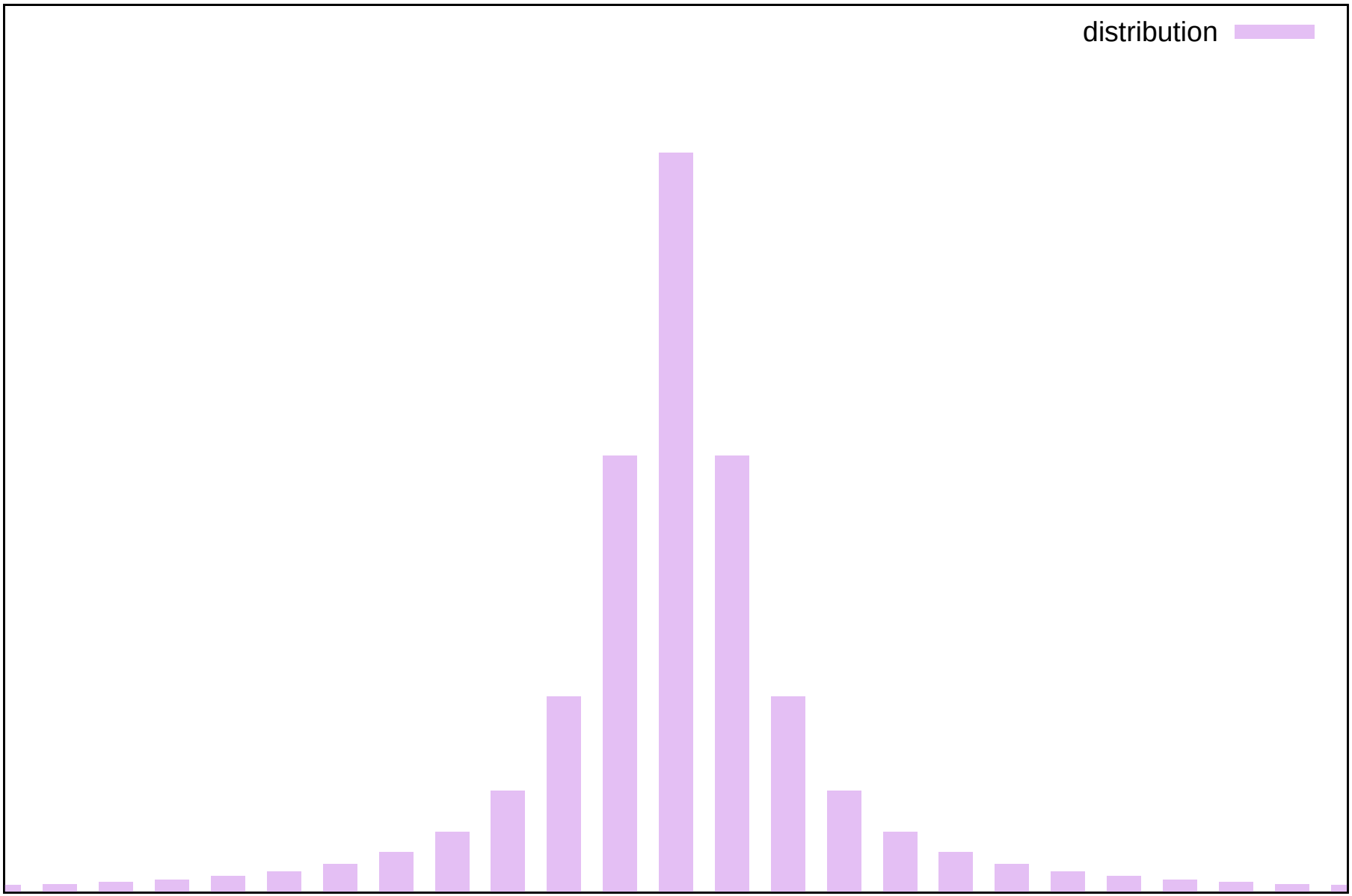
Filled boxes of reduced width



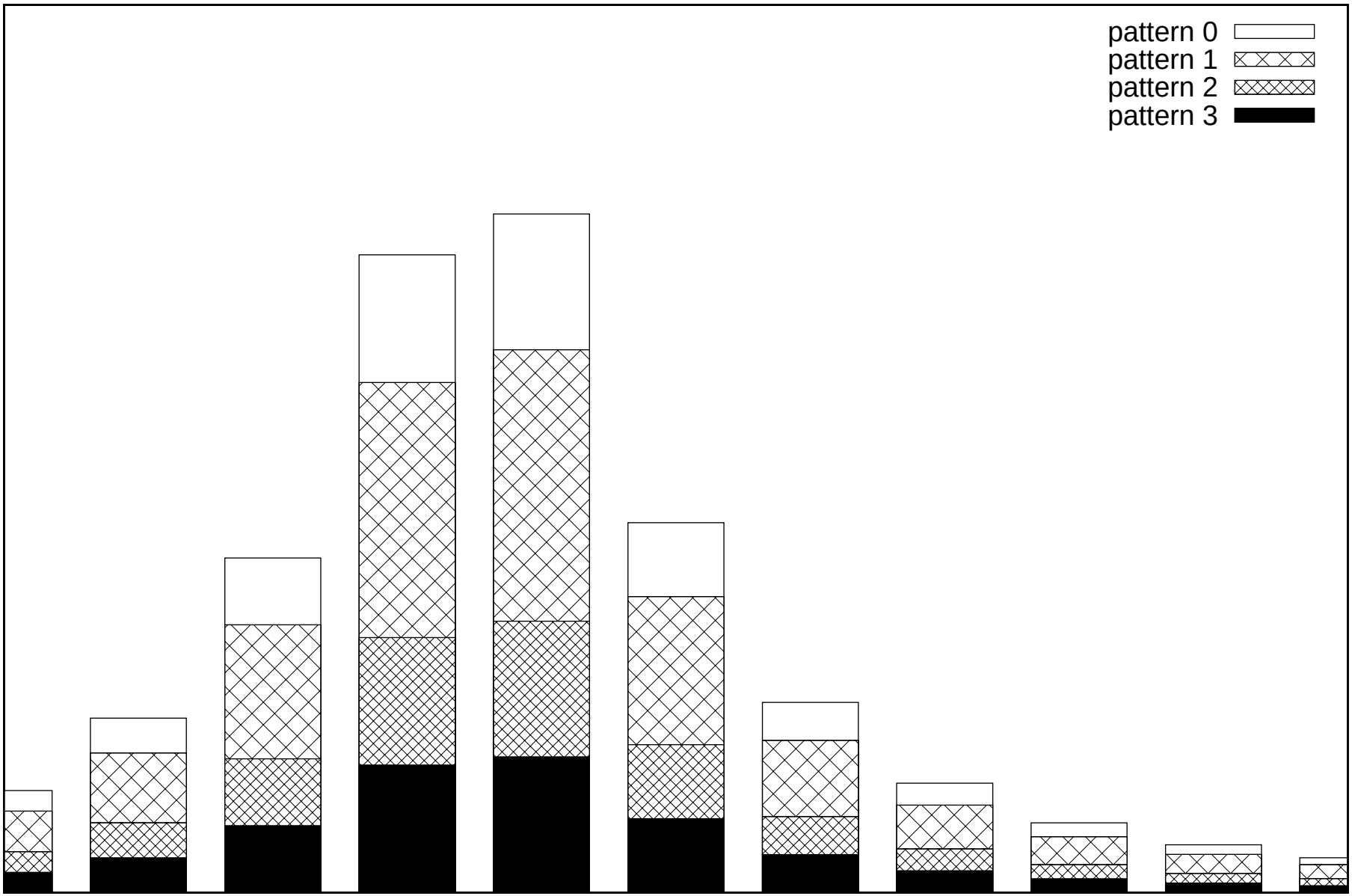
Filled boxes at 50% fill density

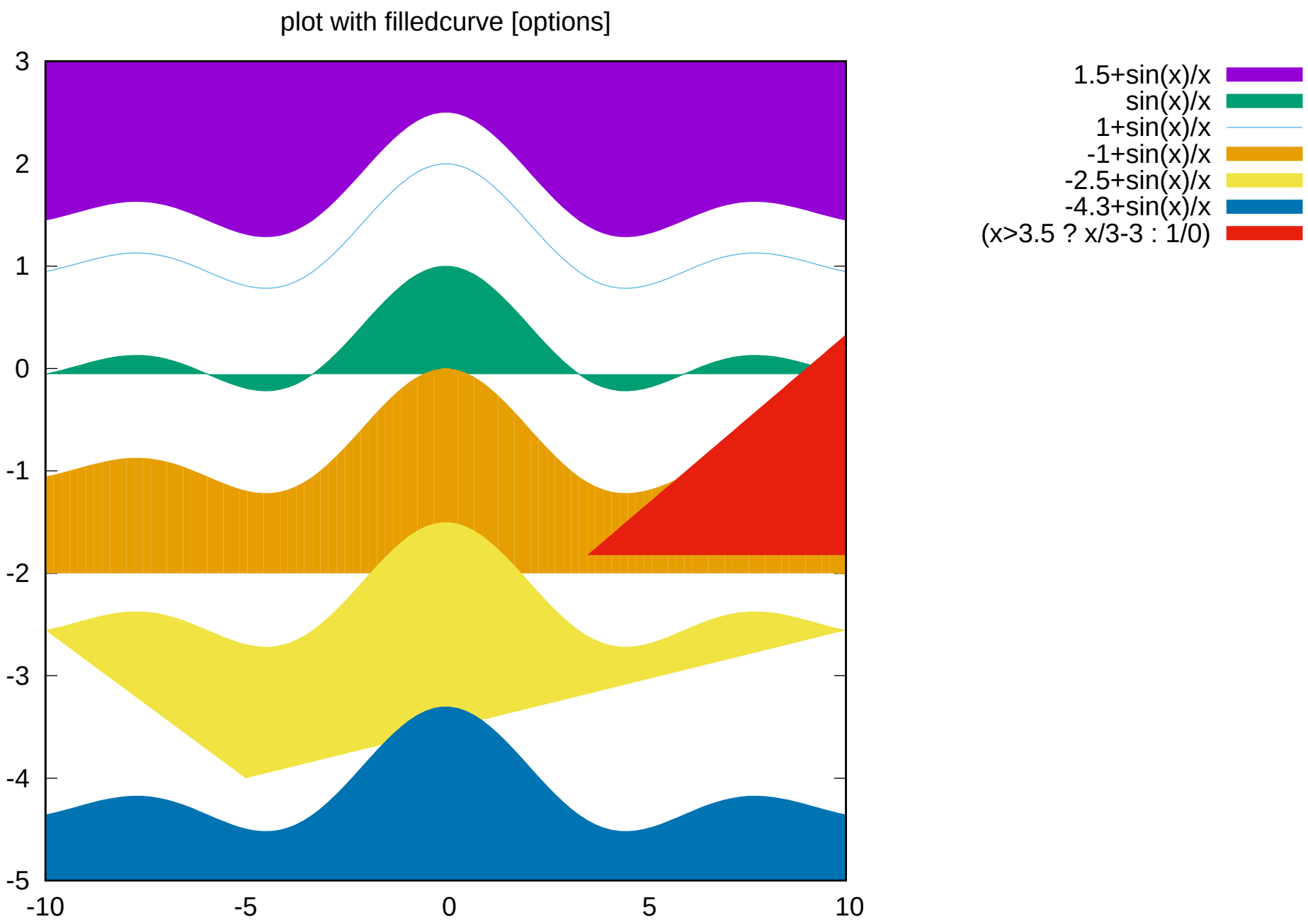


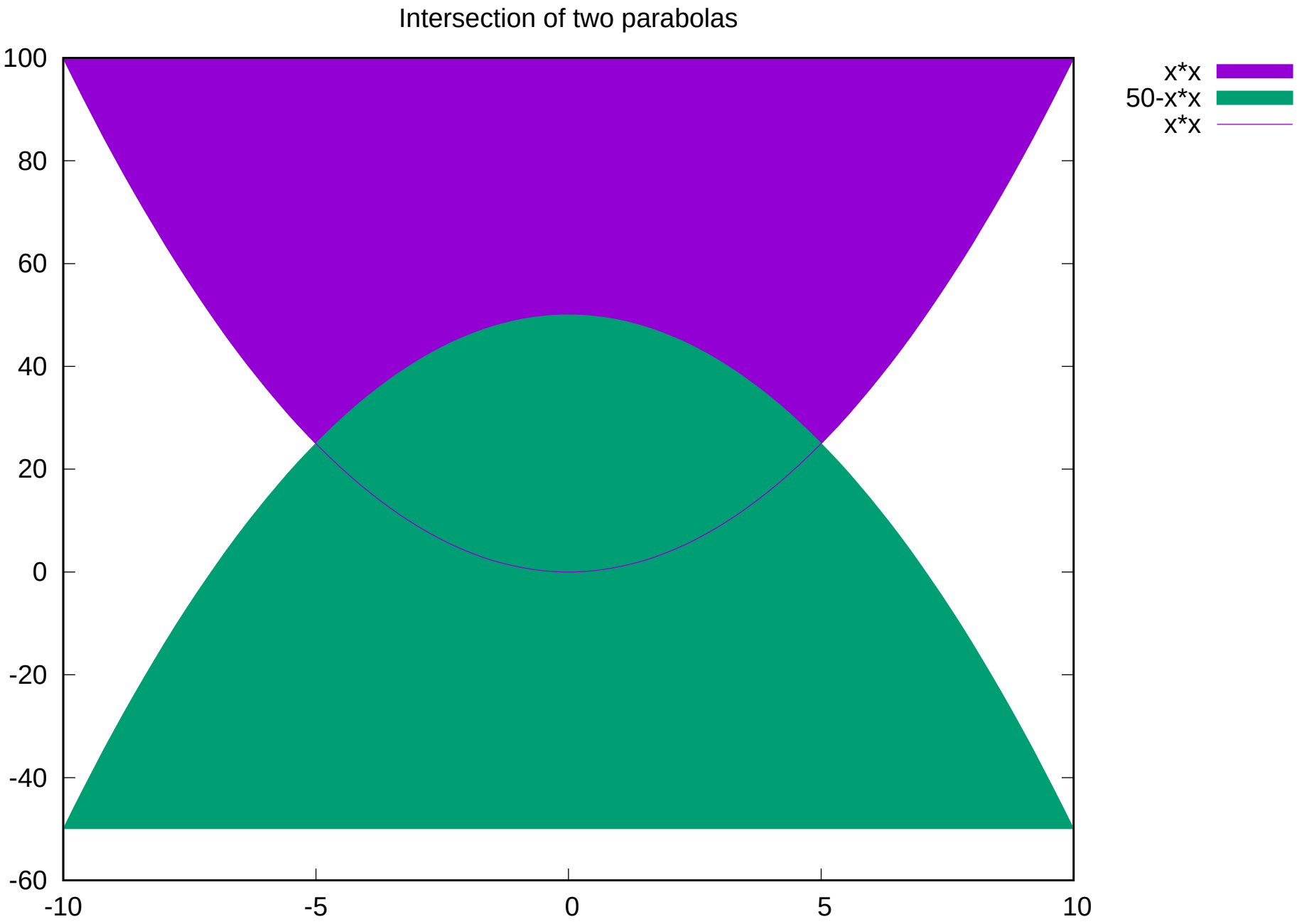
A demonstration of boxes with style fill solid 0.25 noborder

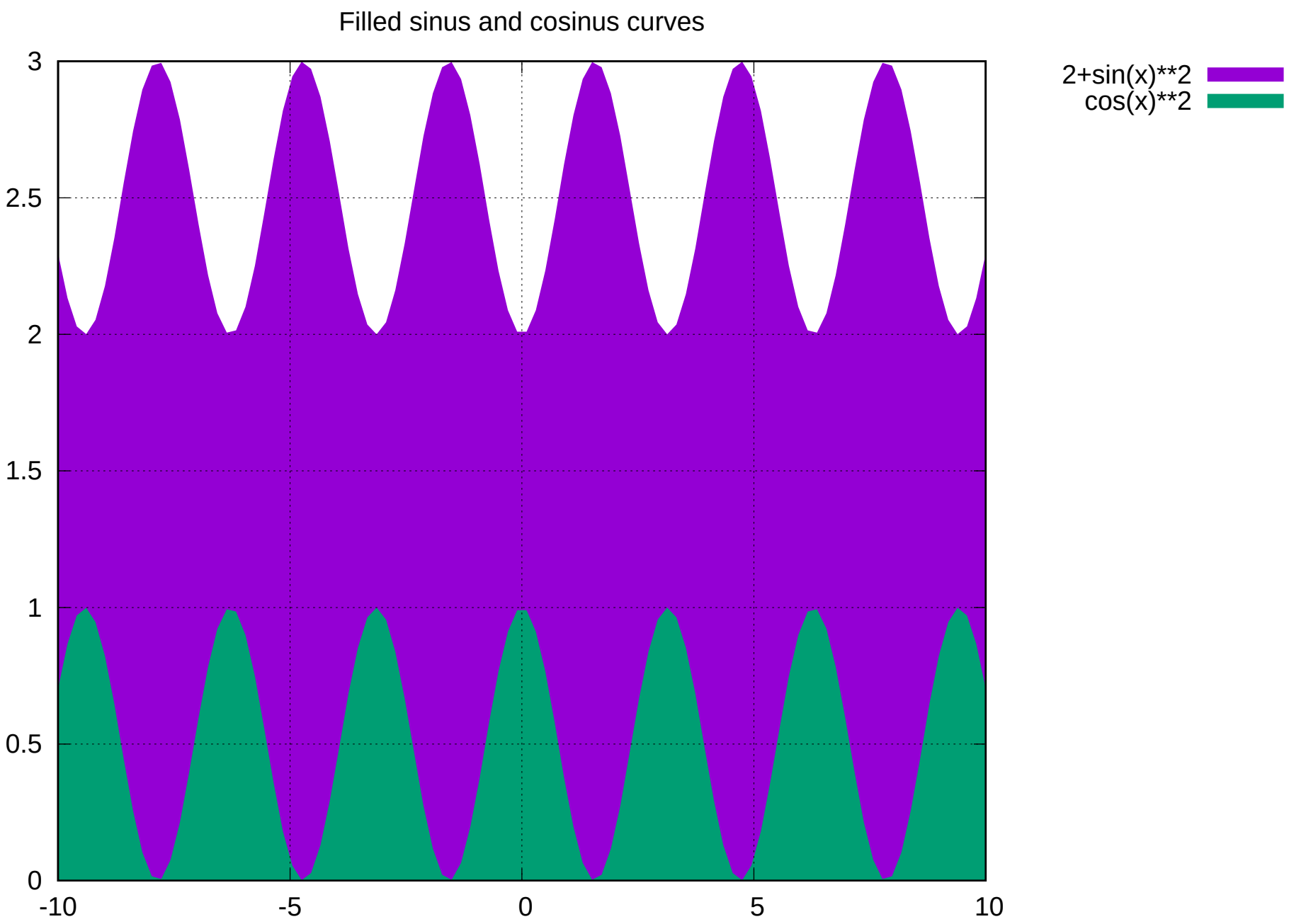


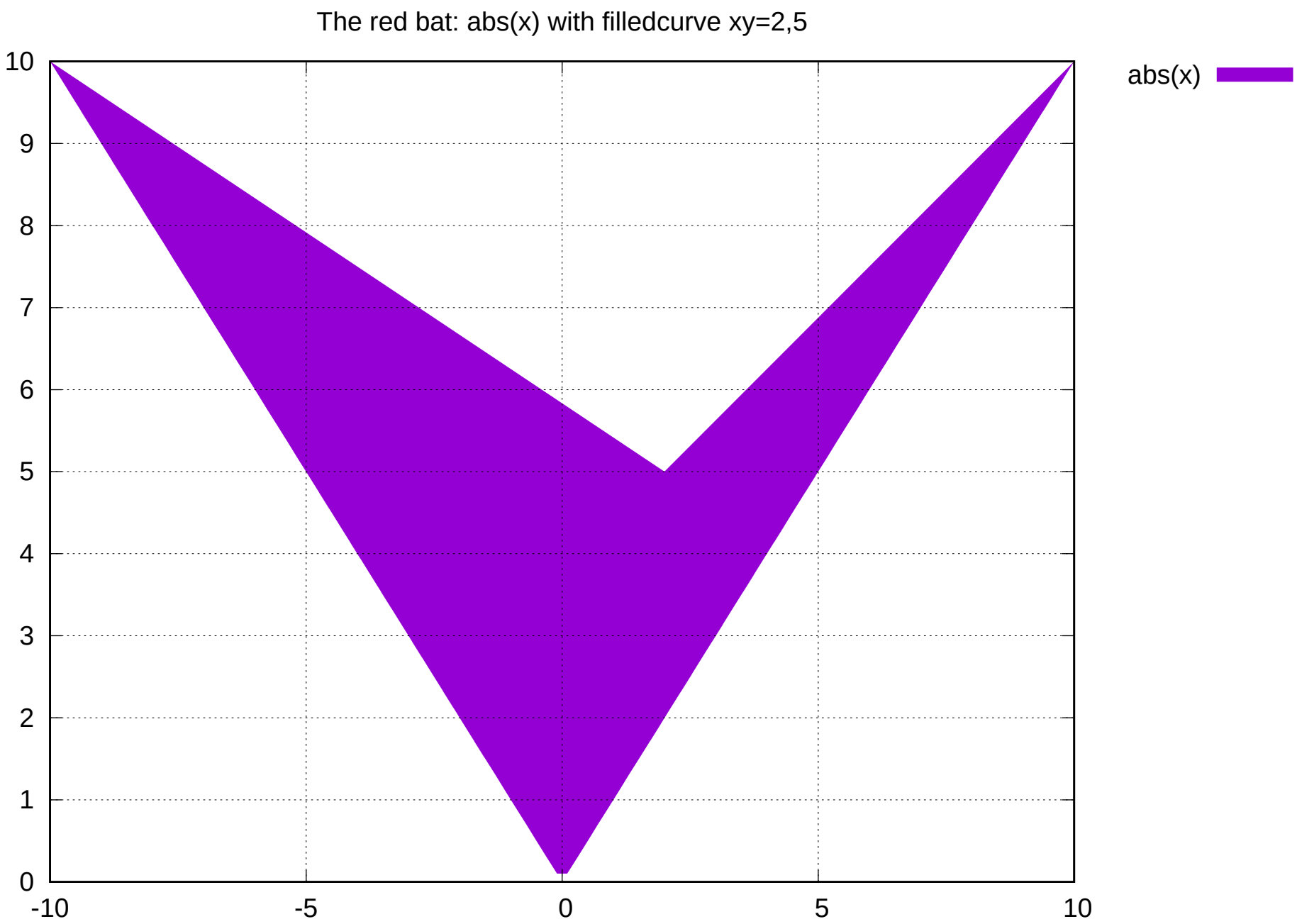
A demonstration of boxes in mono with style fill pattern

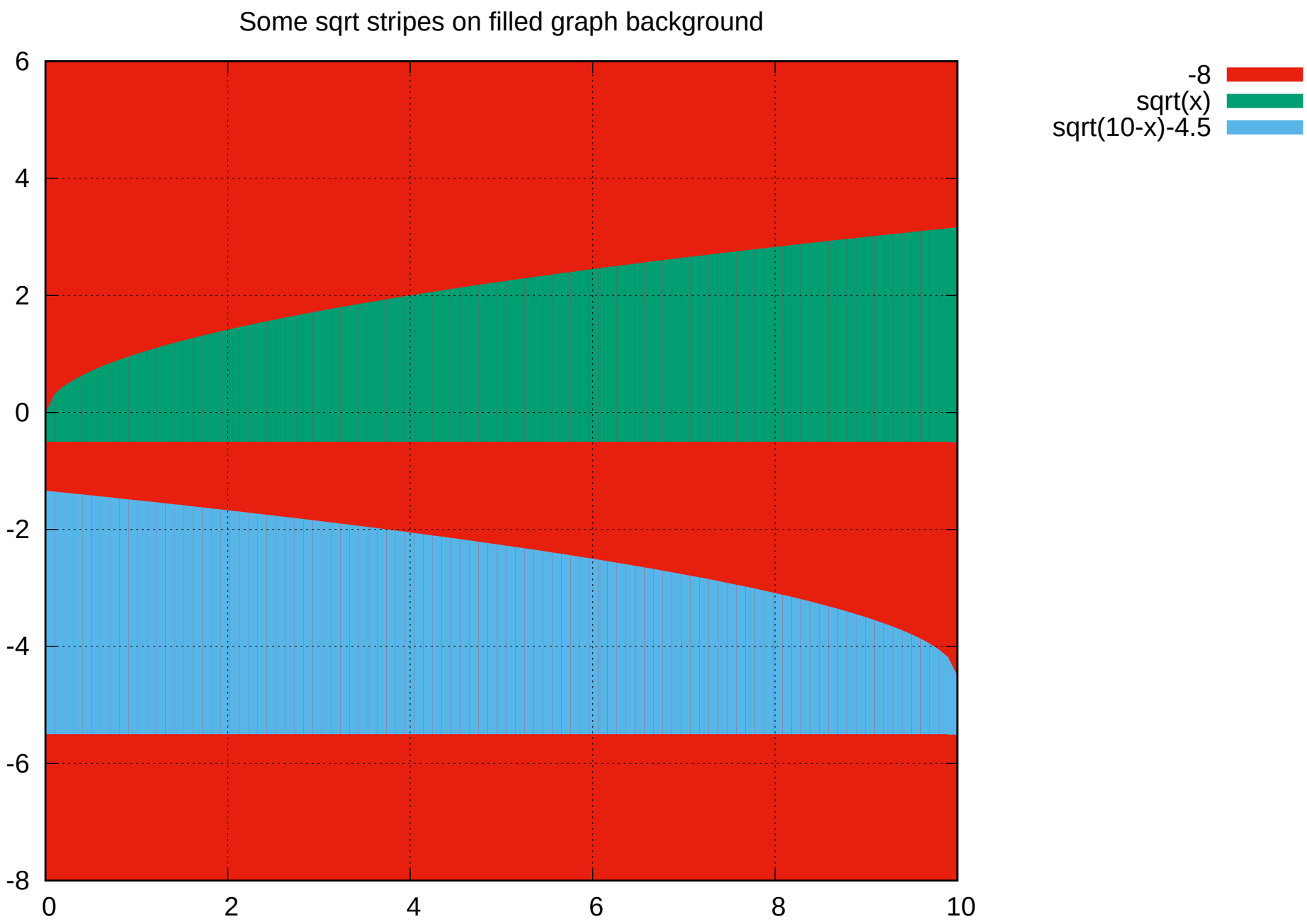



plot with filledcurve [options]
3
2
1
0
-1
-2
-3
-4
-5
-10
-5
0
5
10
1.5+sin(x)/x
sin(x)/x
1+sin(x)/x
-1+sin(x)/x
-2.5+sin(x)/x
-4.3+sin(x)/x
(x>3.5 ? x/3-3 : 1/0)

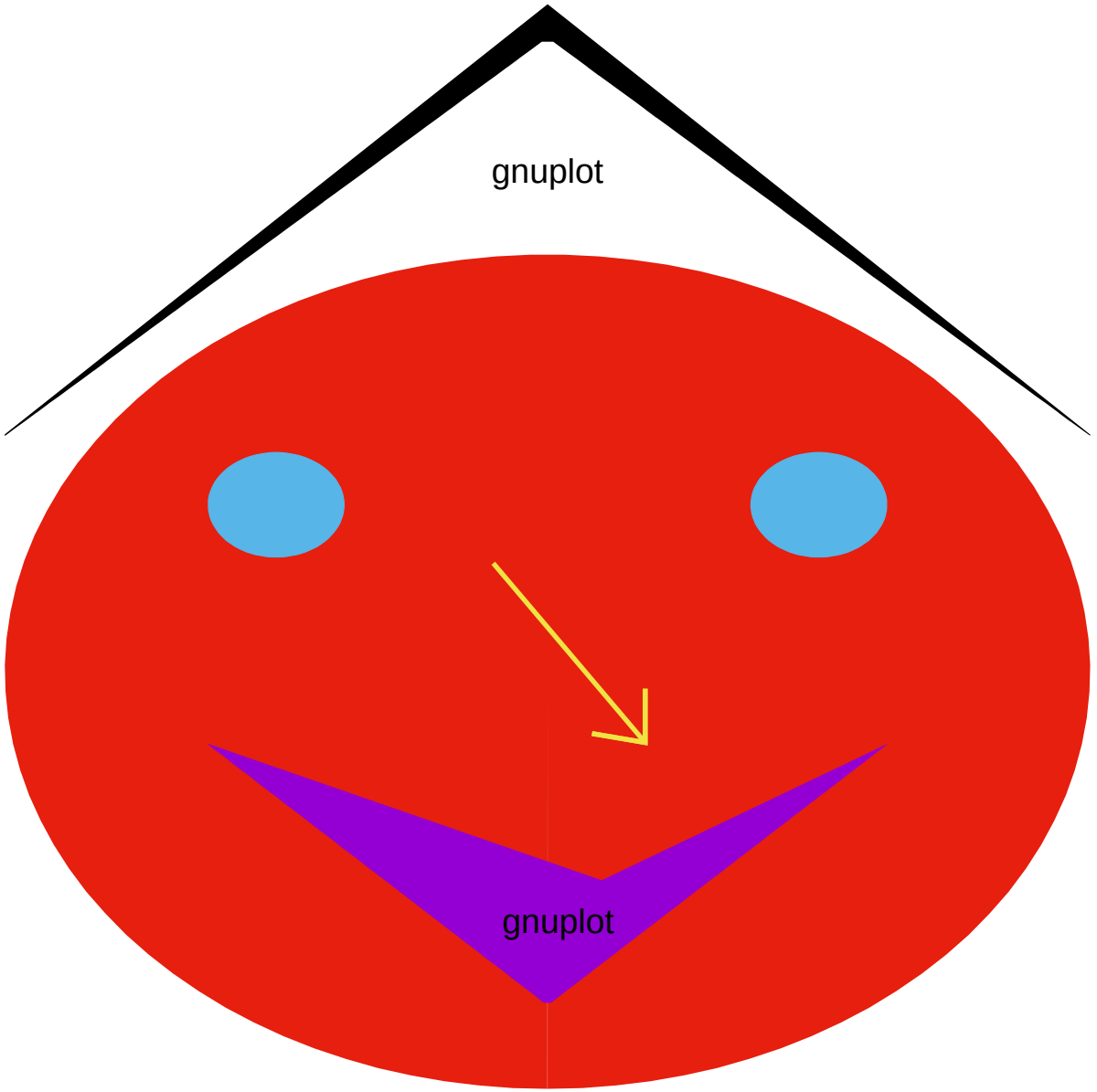

Intersection of two parabolas
100
80
60
40
20
0
-20
-40
-60
-10
-5
0
5
10
x*x
50-x*x
x*x


Filled sinus and cosinus curves
2+sin(x)**2
cos(x)**2
3
2.5
2
1.5
1
0.5
0
-10
-5
0
5
10

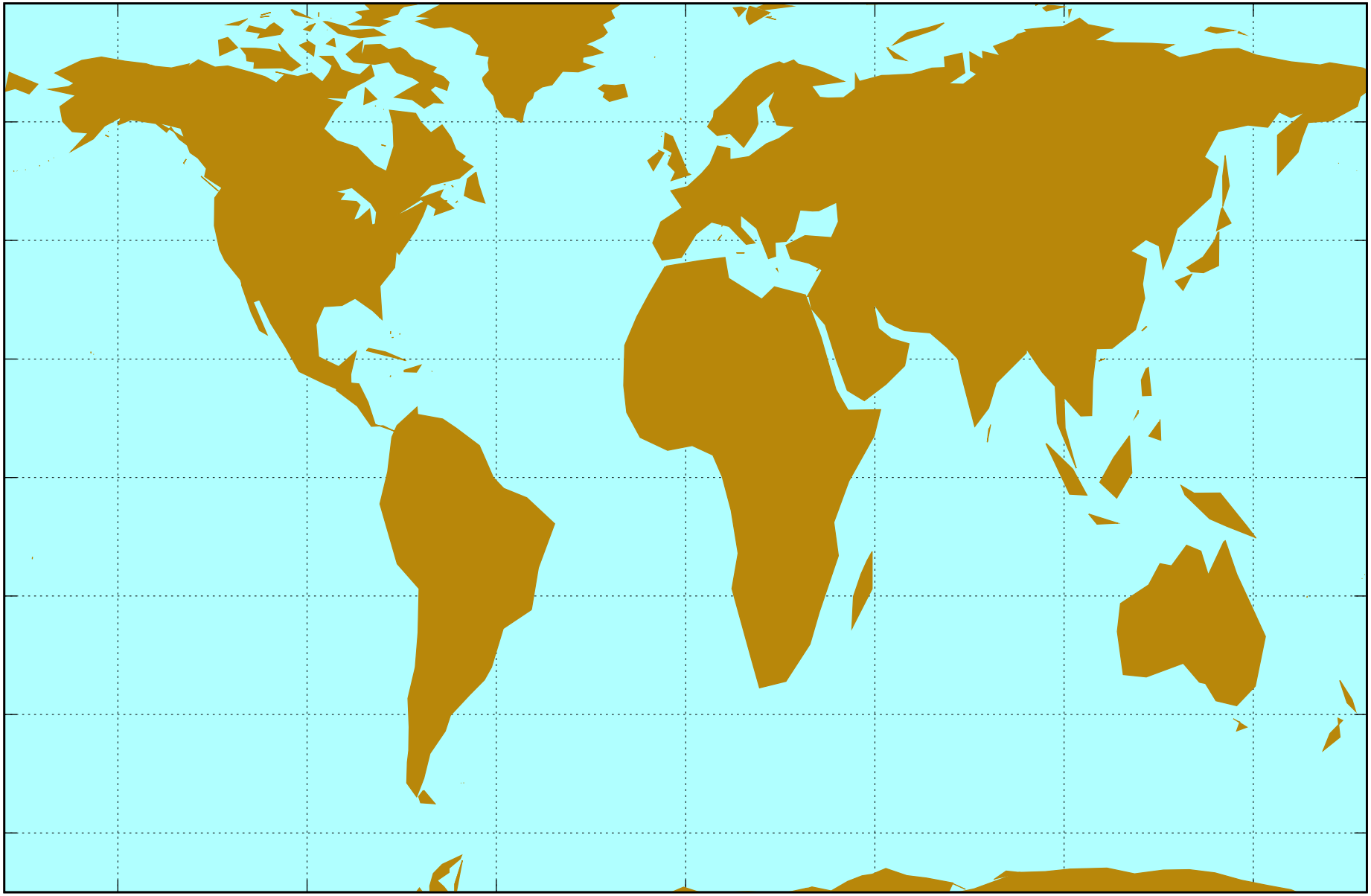

The red bat: abs(x) with filledcurve xy=2,5
abs(x)
10
9
8
7
6
5
4
3
2
1
0
-10
-5
0
5
10

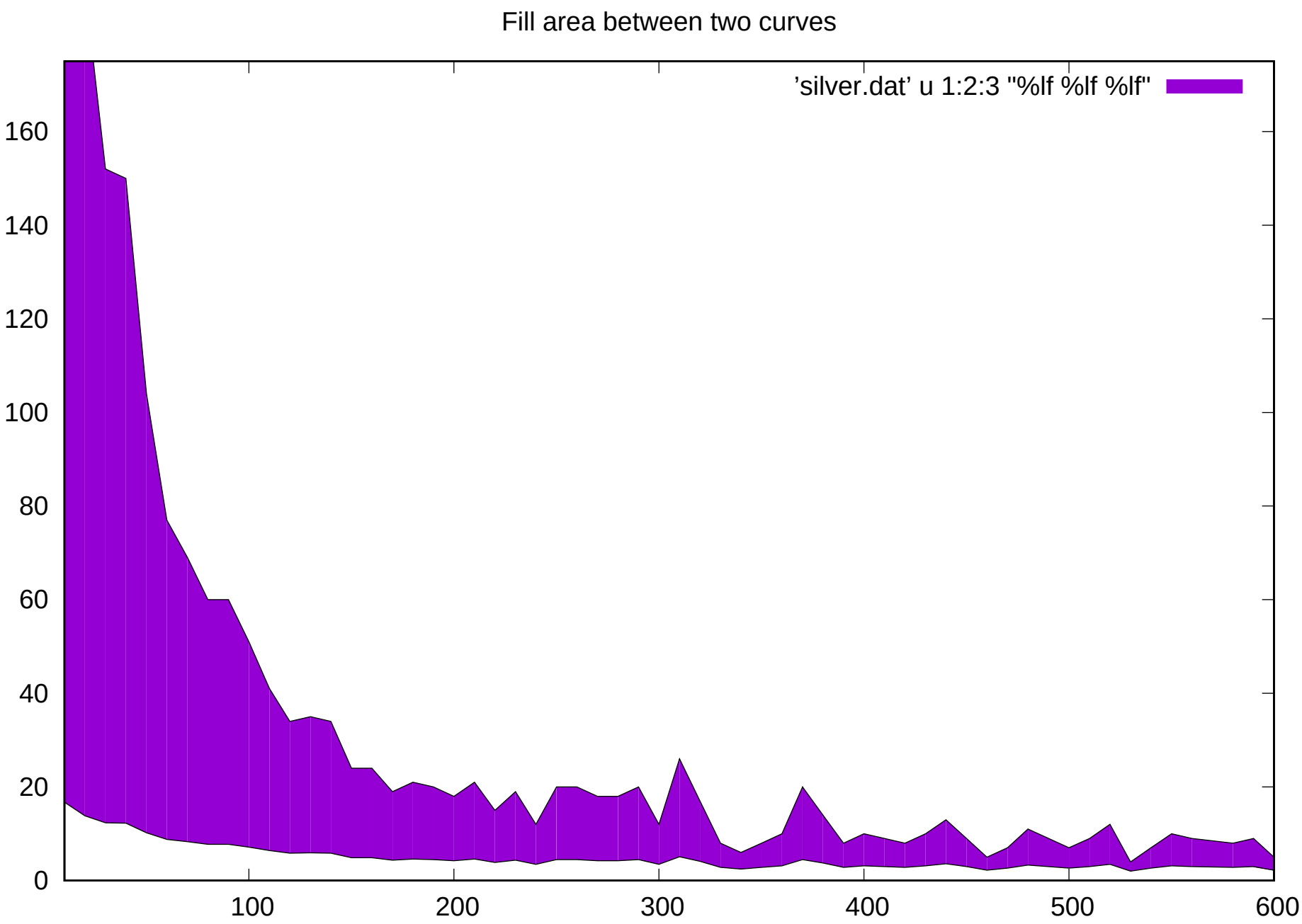

Some sqrt stripes on filled graph background
6
4
2
0
-2
-4
-6
-8
0
2
4
6
8
10
-8
sqrt(x)
sqrt(10-x)-4.5

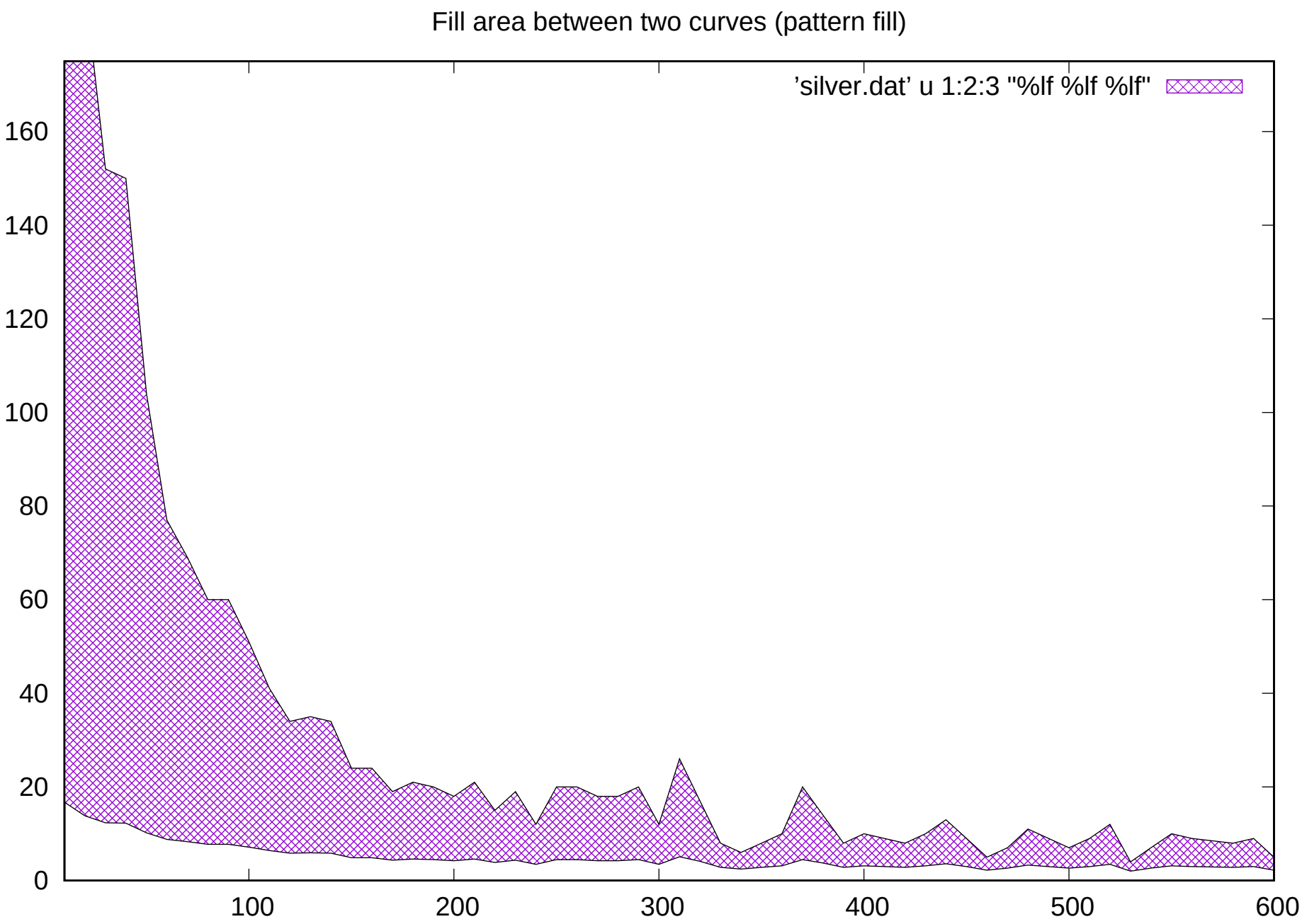
Let’s smile with parametric filled curves

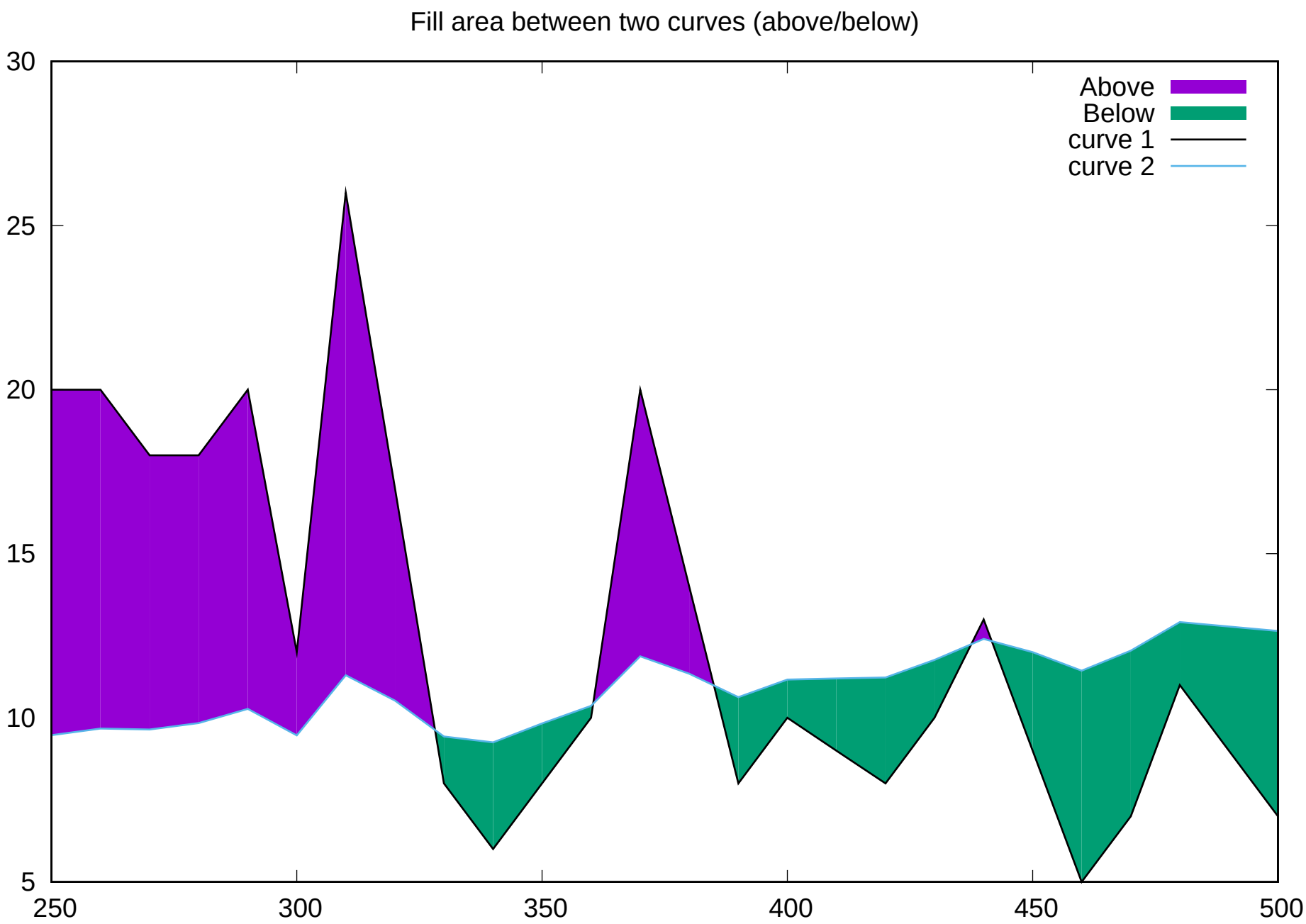


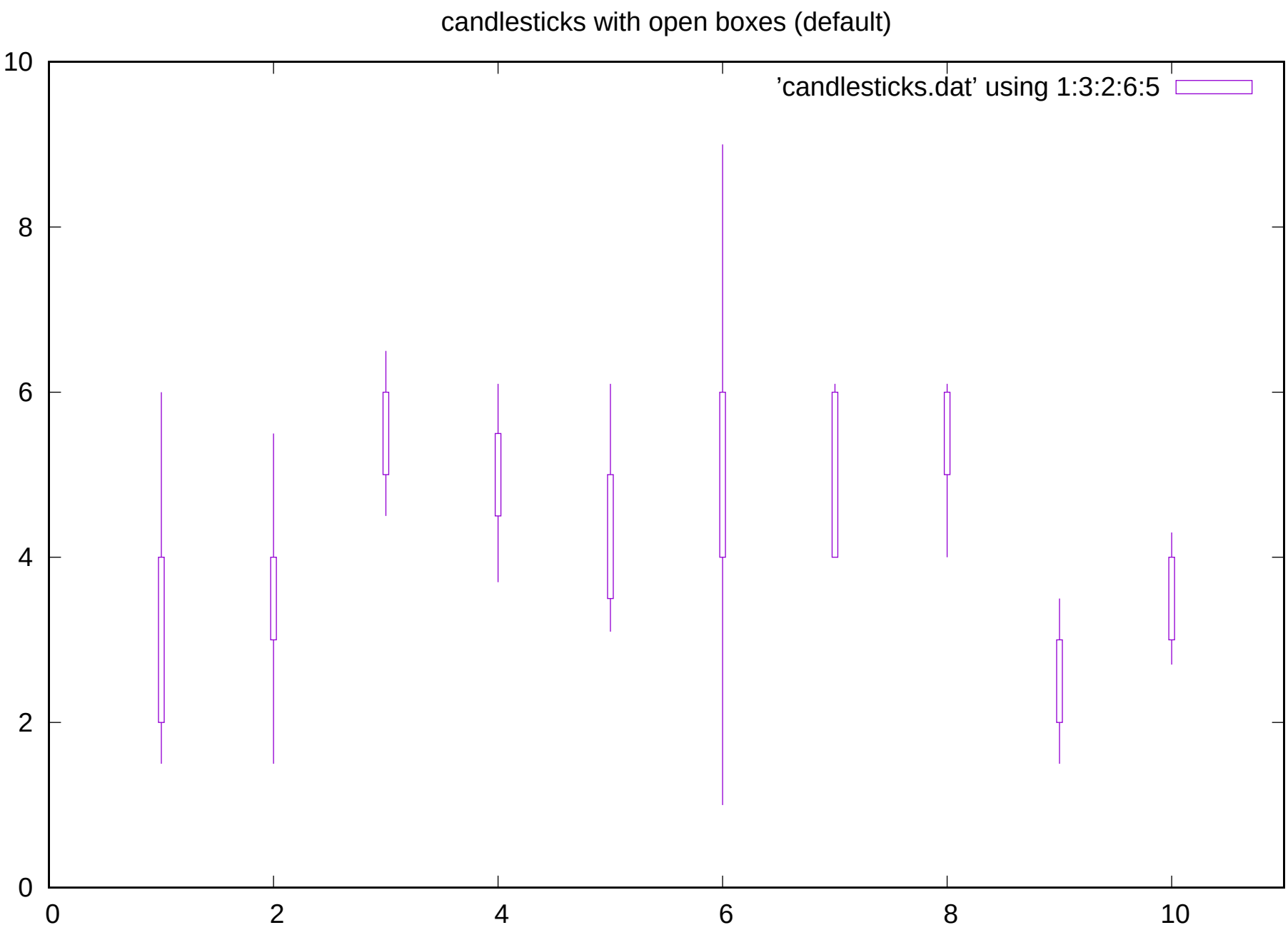
world.dat plotted with filledcurves

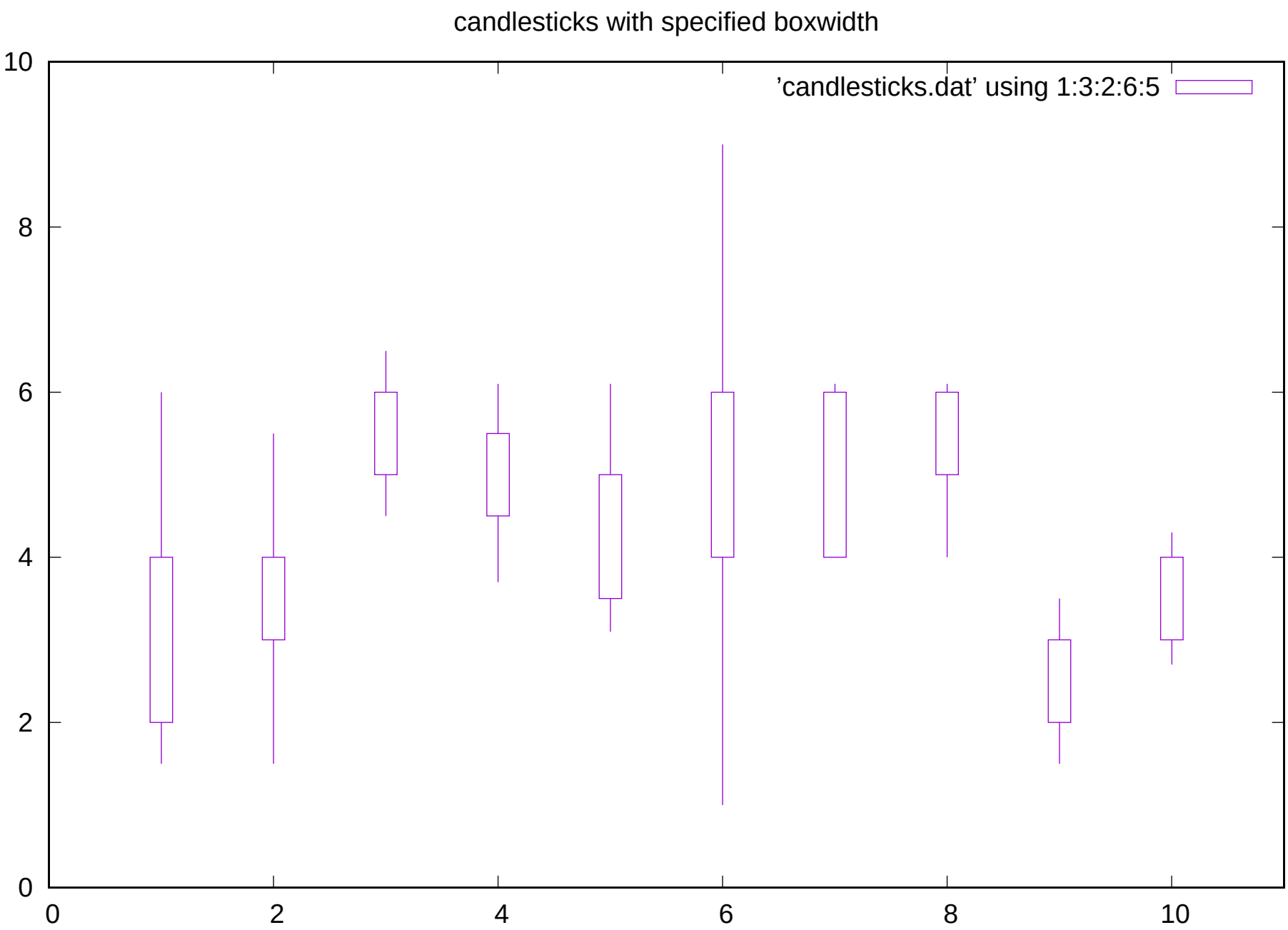


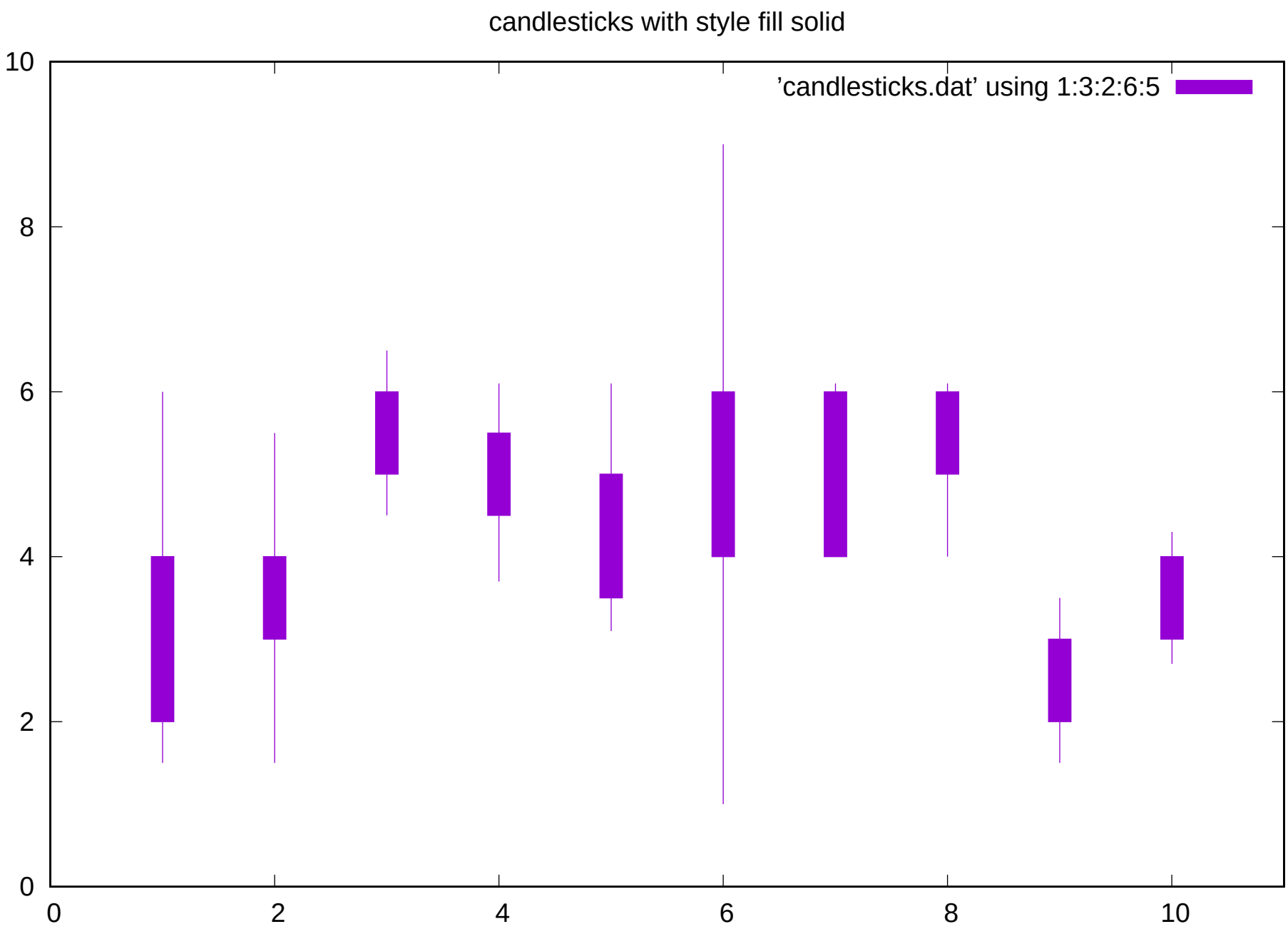

Fill area between two curves
'silver.dat' u 1:2:3 "%lf %lf %lf"
160
140
120
100
80
60
40
20
0
100
200
300
400
500
600

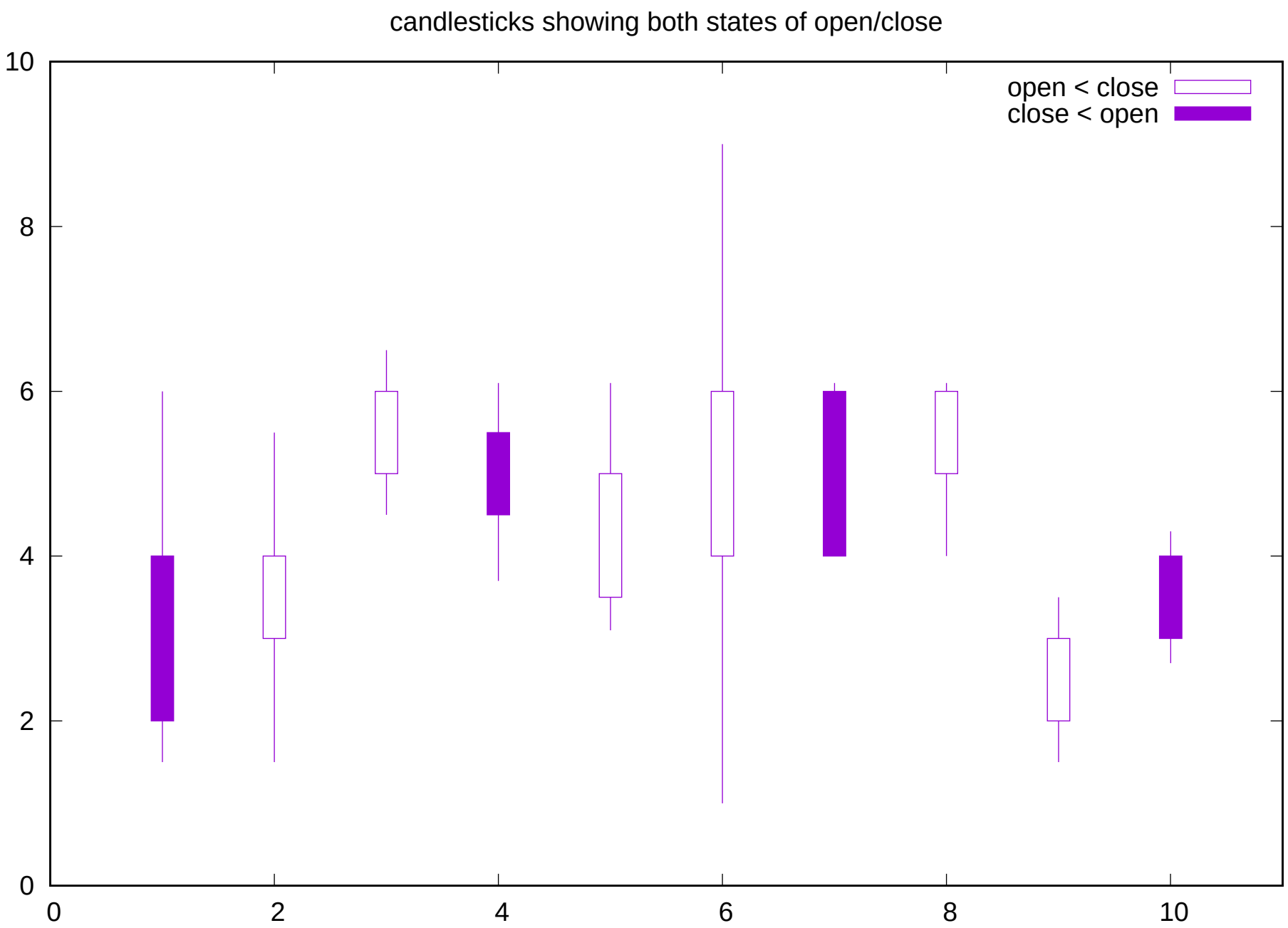

Fill area between two curves (pattern fill)
'silver.dat' u 1:2:3 "%lf %lf %lf"
160
140
120
100
80
60
40
20
0
100
200
300
400
500
600

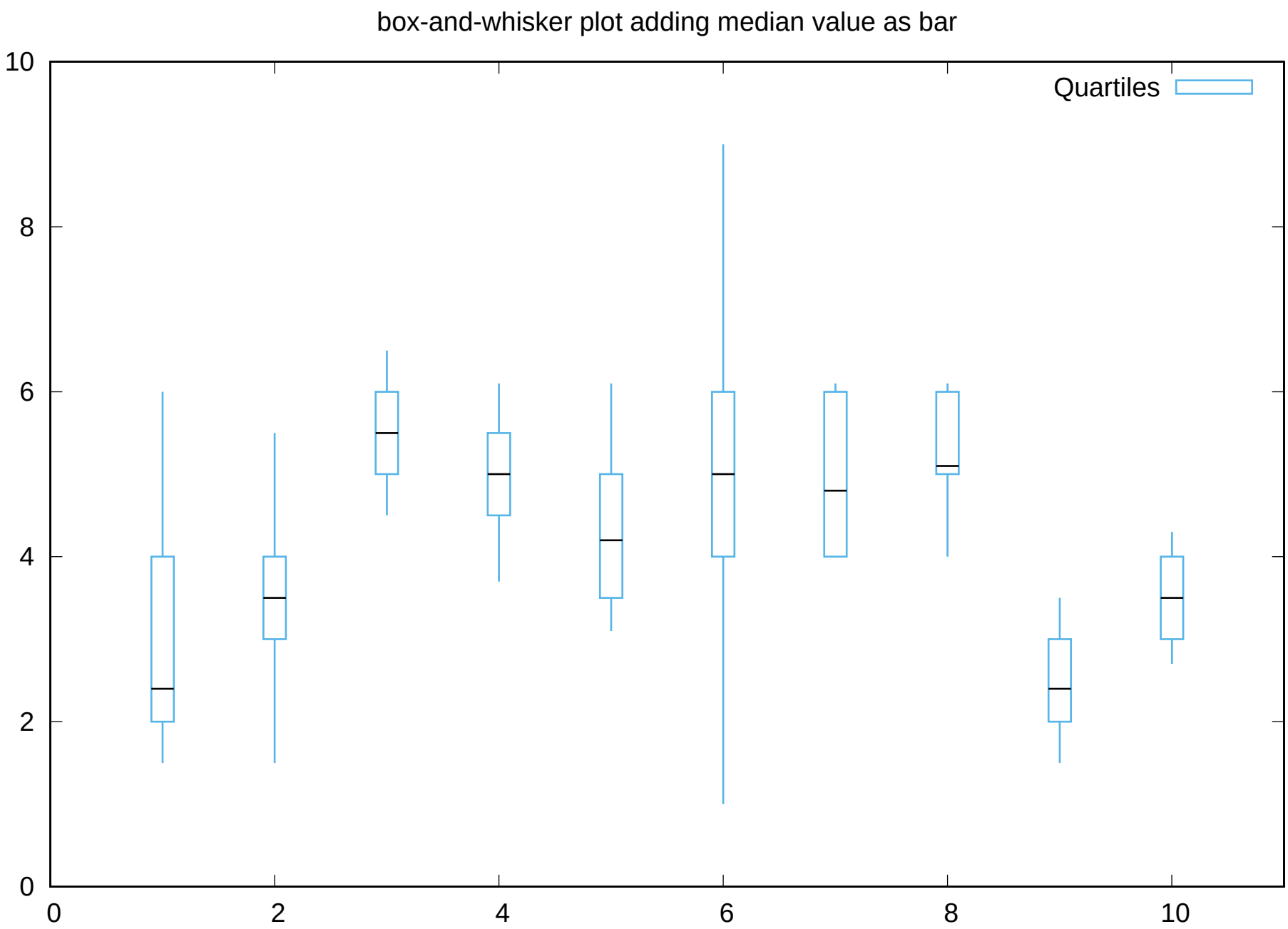

Fill area between two curves (above/below)
Above
Below
curve 1
curve 2
30
25
20
15
10
5
250
300
350
400
450
500

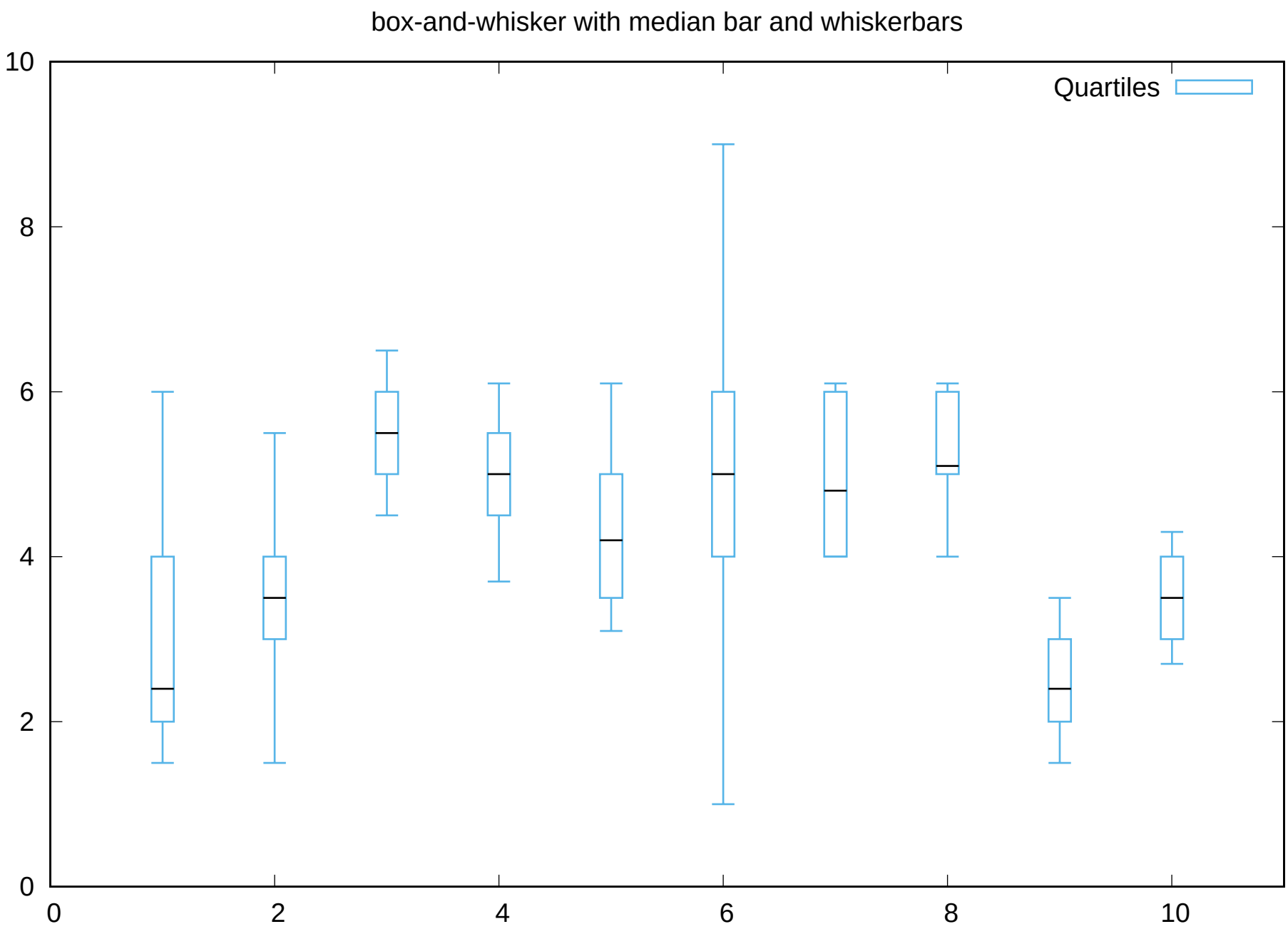

candlesticks with open boxes (default)
'candlesticks.dat' using 1:3:2:6:5
10
8
6
4
2
0
0
2
4
6
8
10

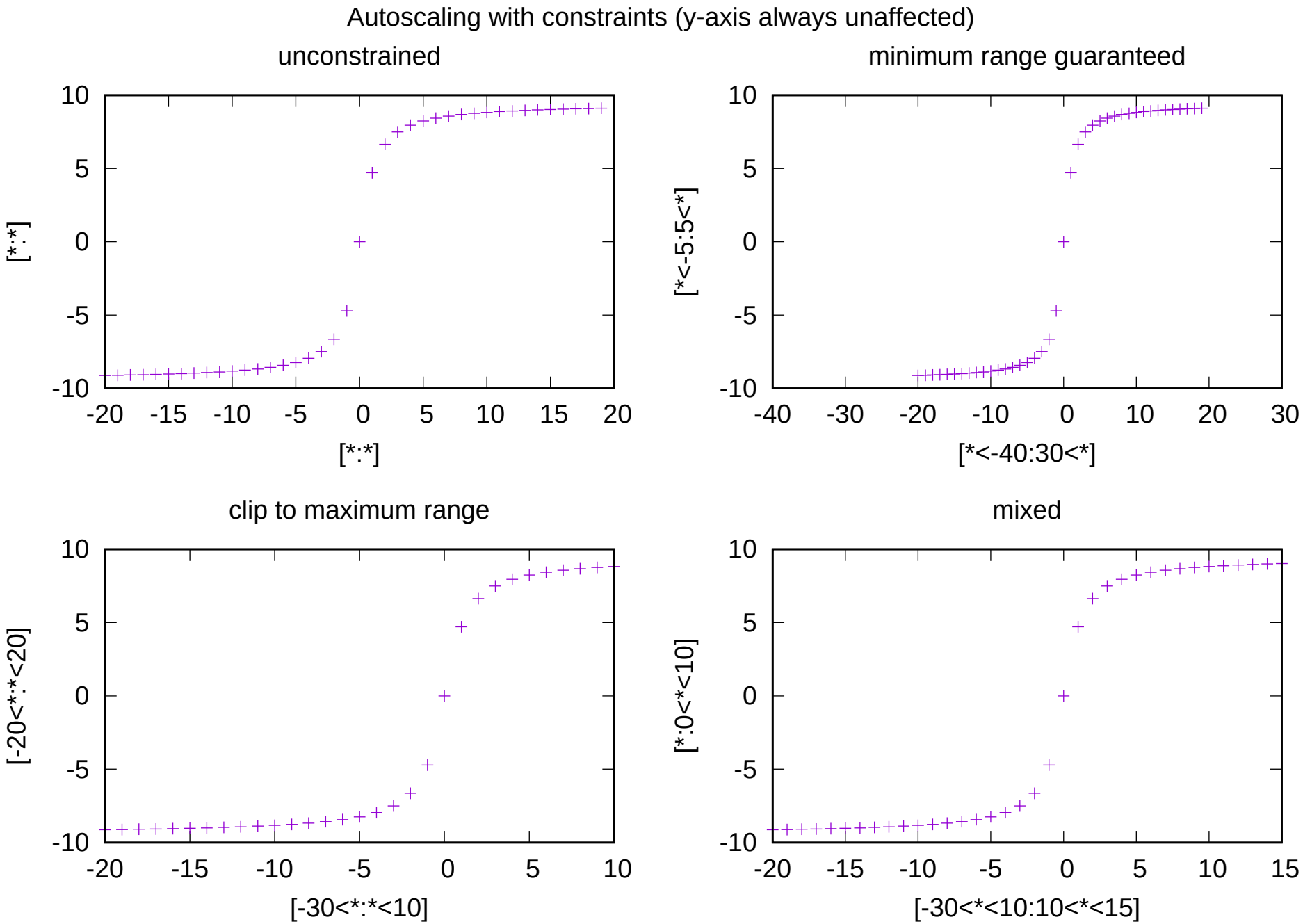

candlesticks with specified boxwidth
'candlesticks.dat' using 1:3:2:6:5
10
8
6
4
2
0
0
2
4
6
8
10

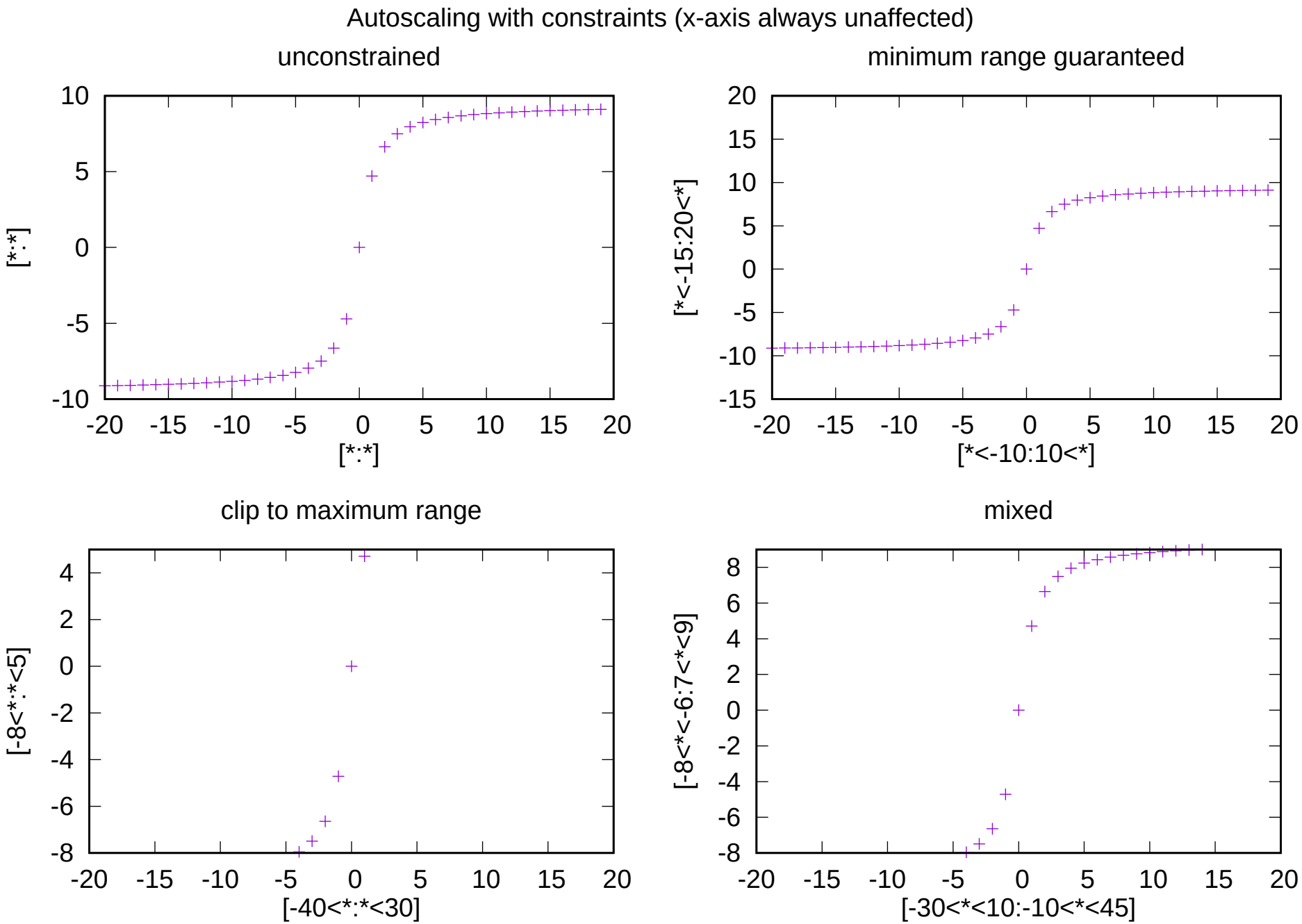

candlesticks with style fill solid
'candlesticks.dat' using 1:3:2:6:5
10
8
6
4
2
0
0
2
4
6
8
10

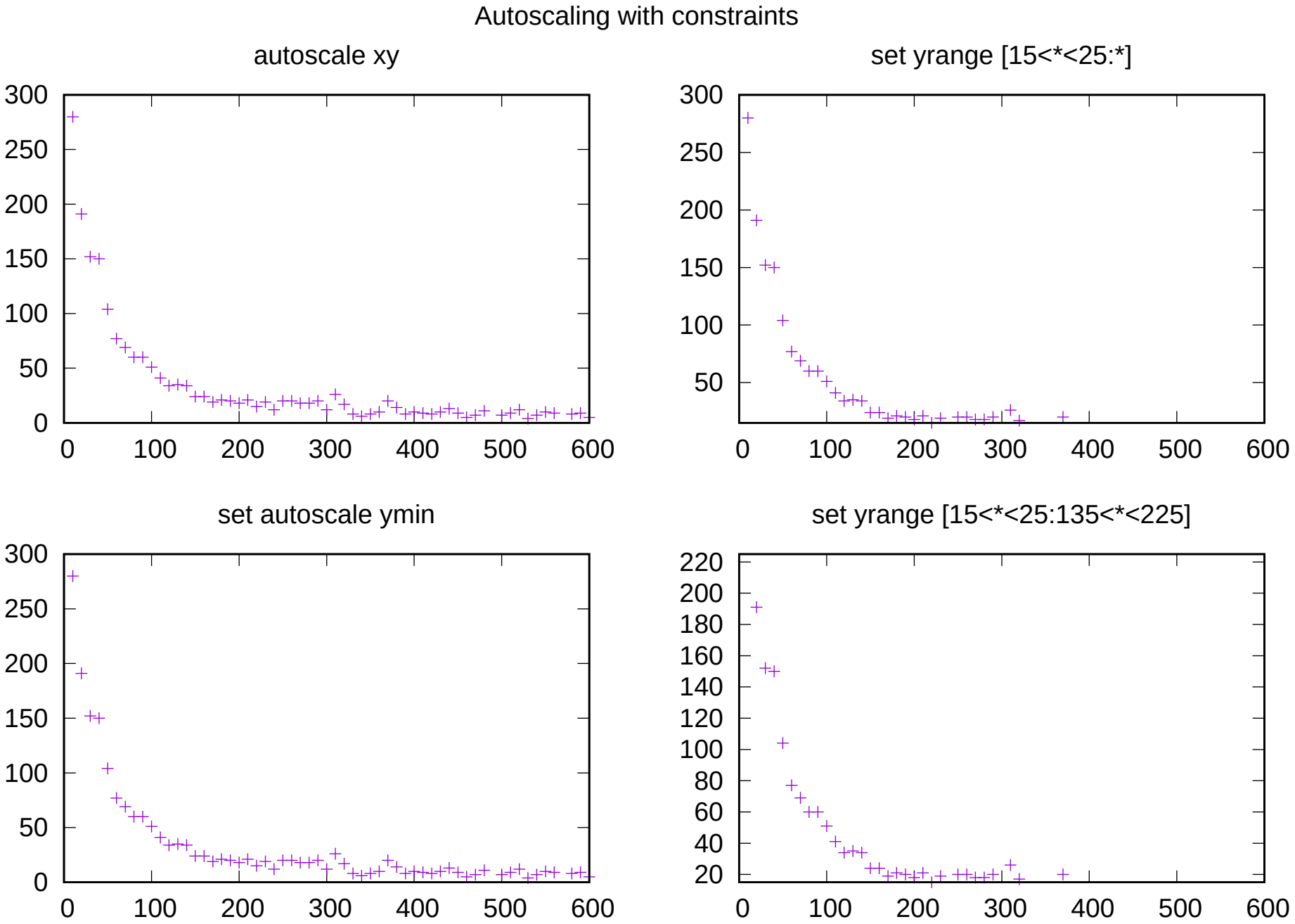

candlesticks showing both states of open/close
open < close
close < open
10
8
6
4
2
0
0
2
4
6
8
10

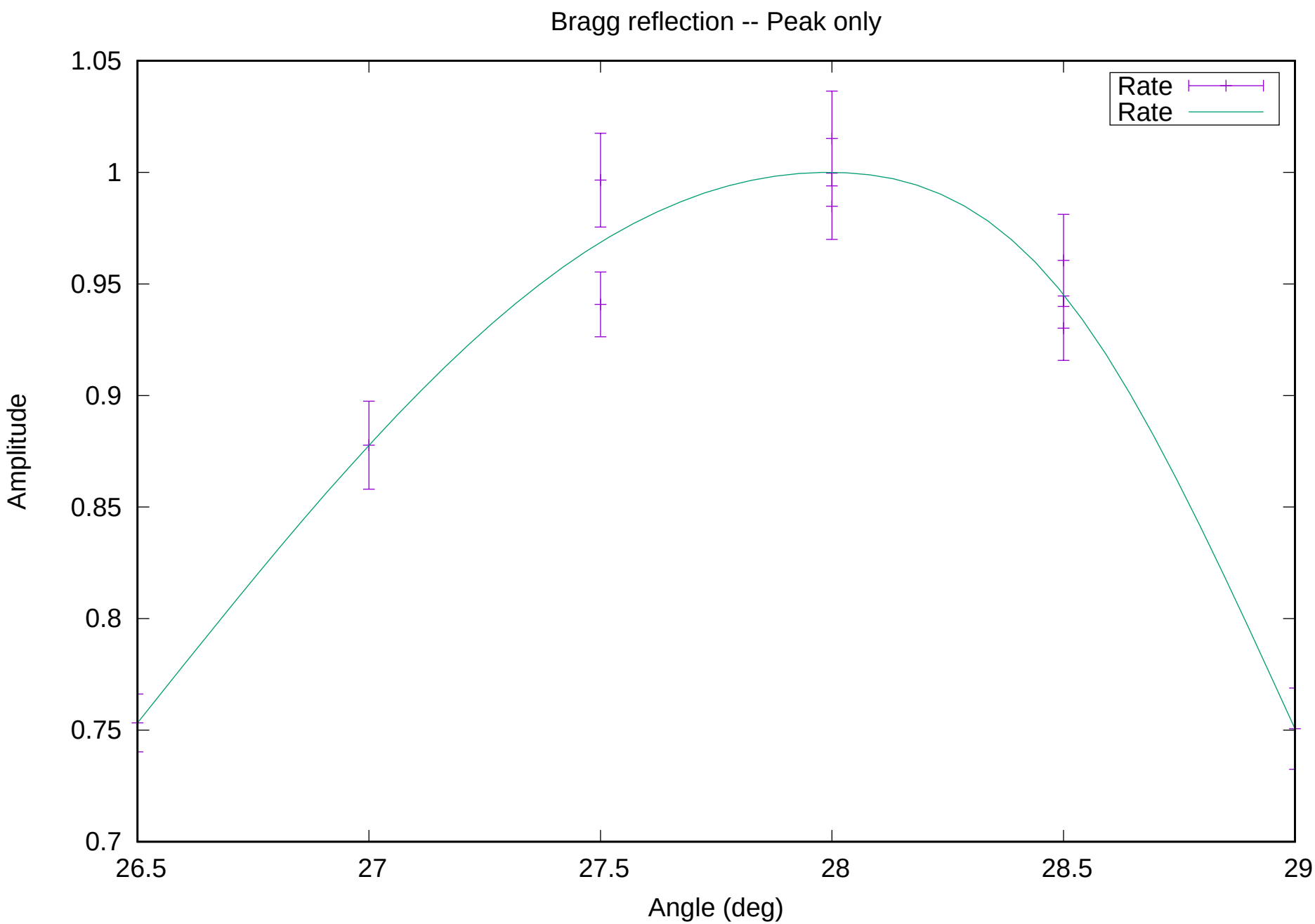

box-and-whisker plot adding median value as bar
Quartiles
10
8
6
4
2
0
0
2
4
6
8
10

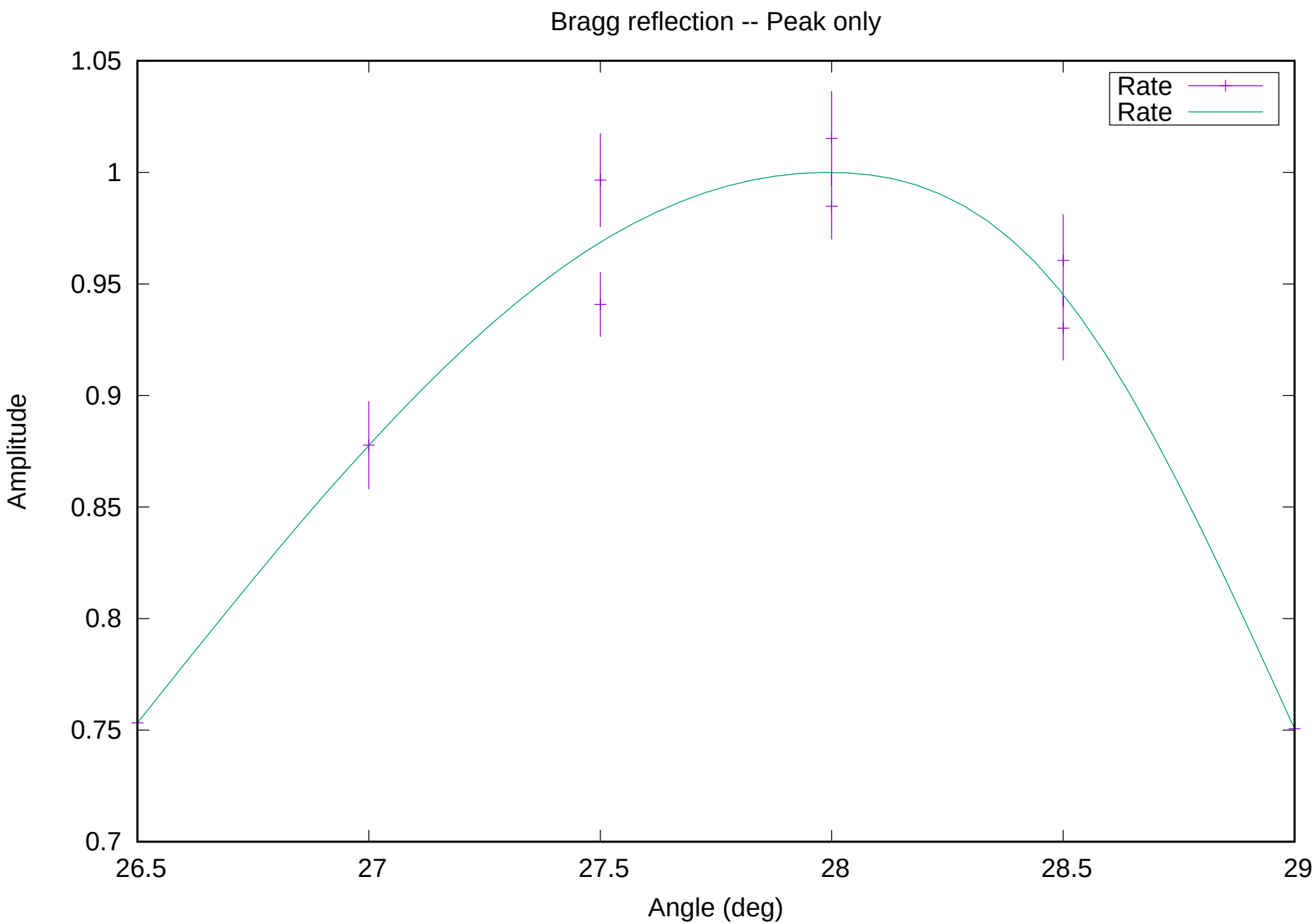

box-and-whisker with median bar and whiskerbars
Quartiles
10
8
6
4
2
0
0
2
4
6
8
10

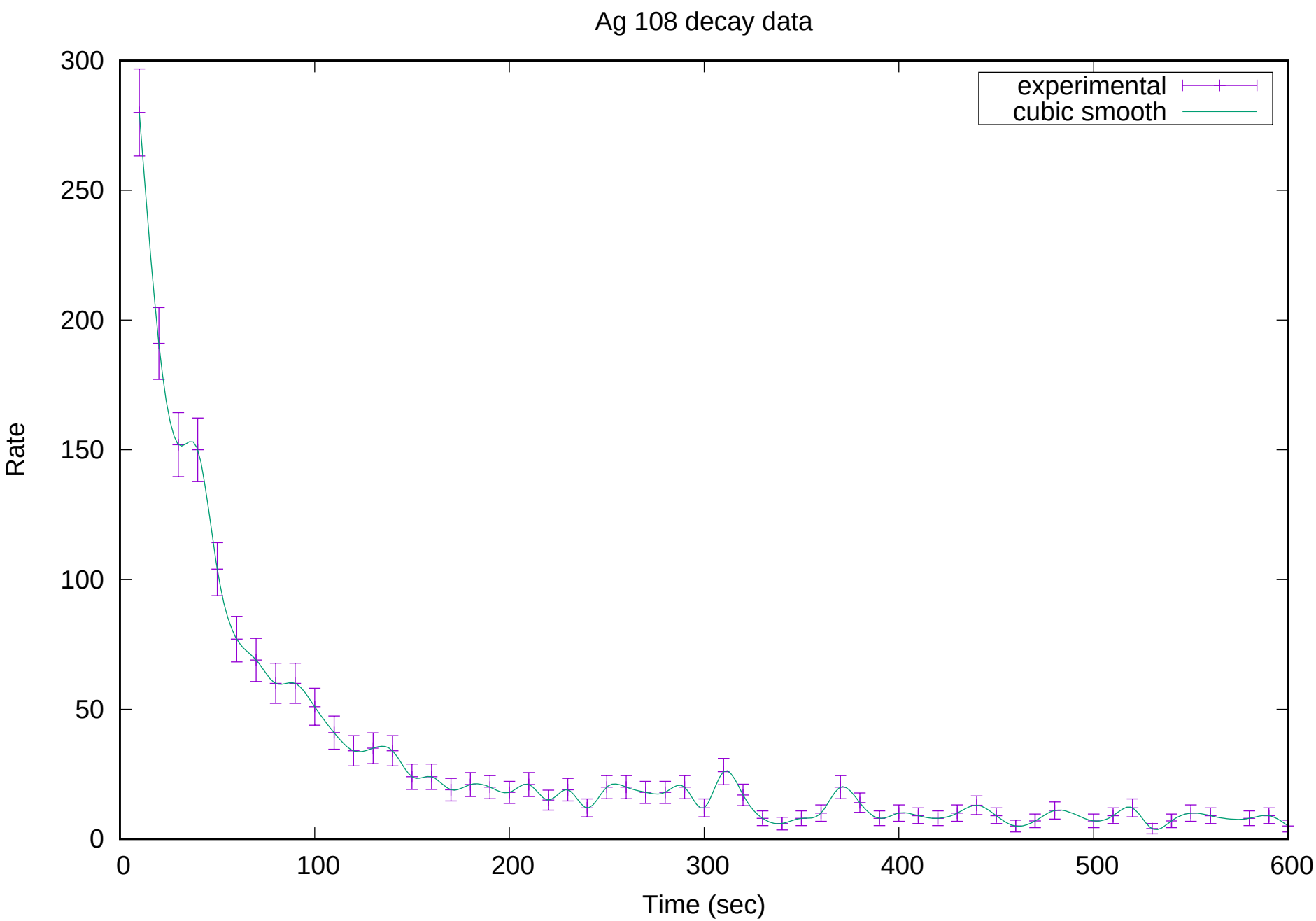

Autoscaling with constraints (y-axis always unaffected)
unconstrained
[*:*]
[*:*]
minimum range guaranteed
[*<-5:5<*]
[*<-40:30<*]
clip to maximum range
[-20<*:*<20]
[-30<*:*<10]
mixed
[*:0<*<10]
[-30<*<10:10<*<15]

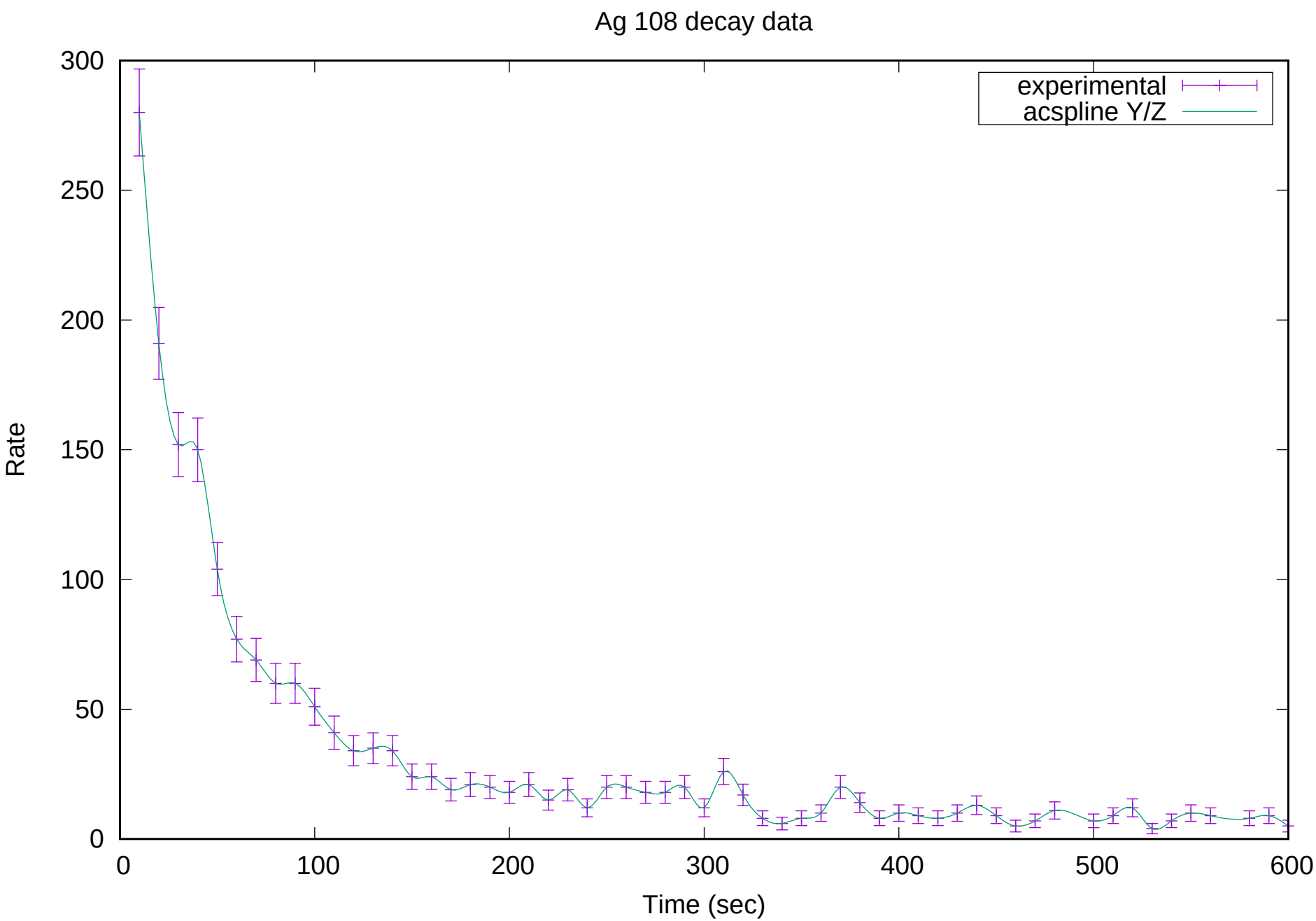

Autoscaling with constraints (x-axis always unaffected)
unconstrained
[*:*]
[*:*]
minimum range guaranteed
[*<-15:20<*]
[*<-10:10<*]
clip to maximum range
[-8<*:*<5]
[-40<*:*<30]
mixed
[-8<*<-6:7<*<9]
[-30<*<10:-10<*<45]

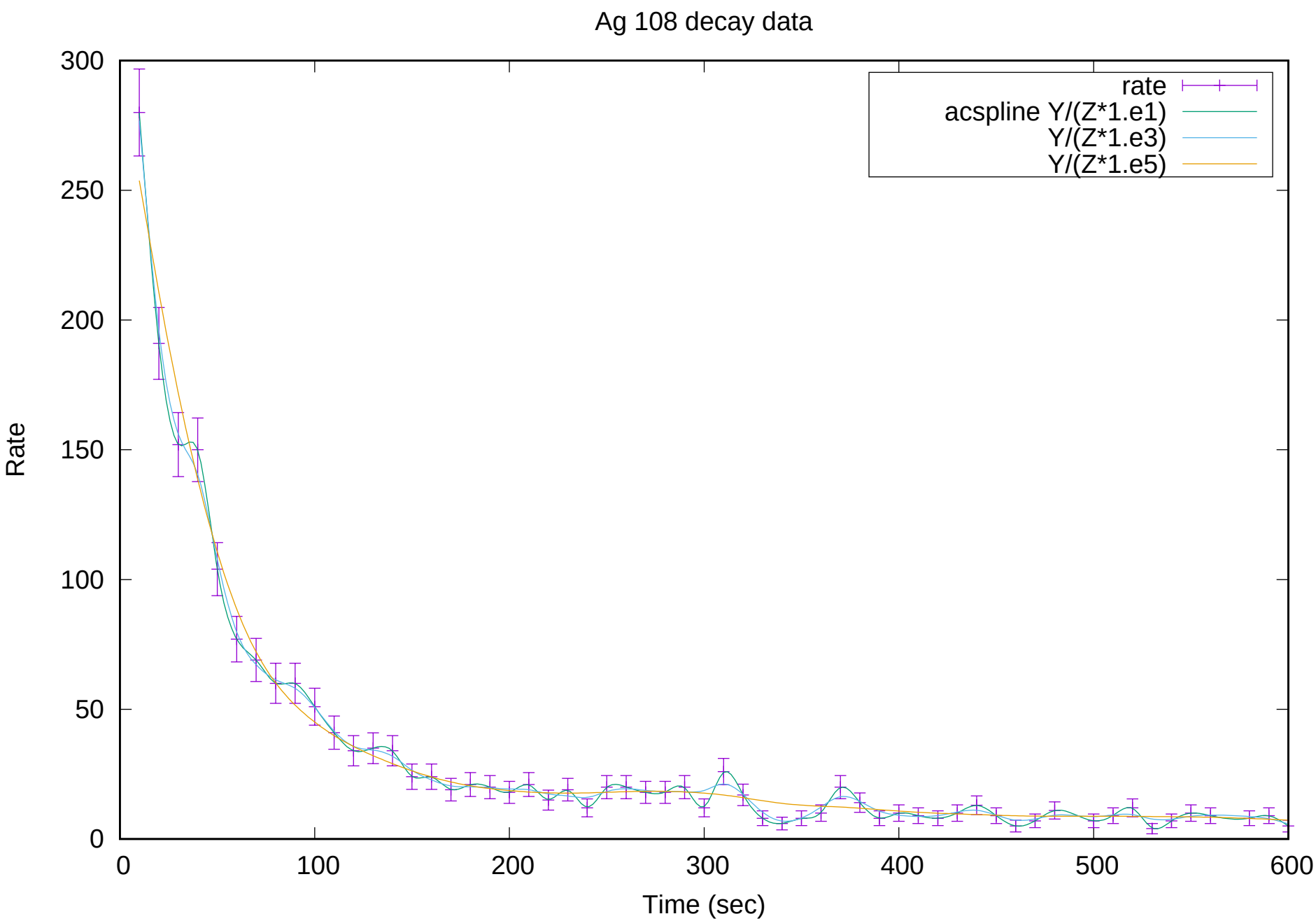

Autoscaling with constraints
autoscale xy
set yrange [15<*<25:*]
set autoscale ymin
set yrange [15<*<25:135<*<225]

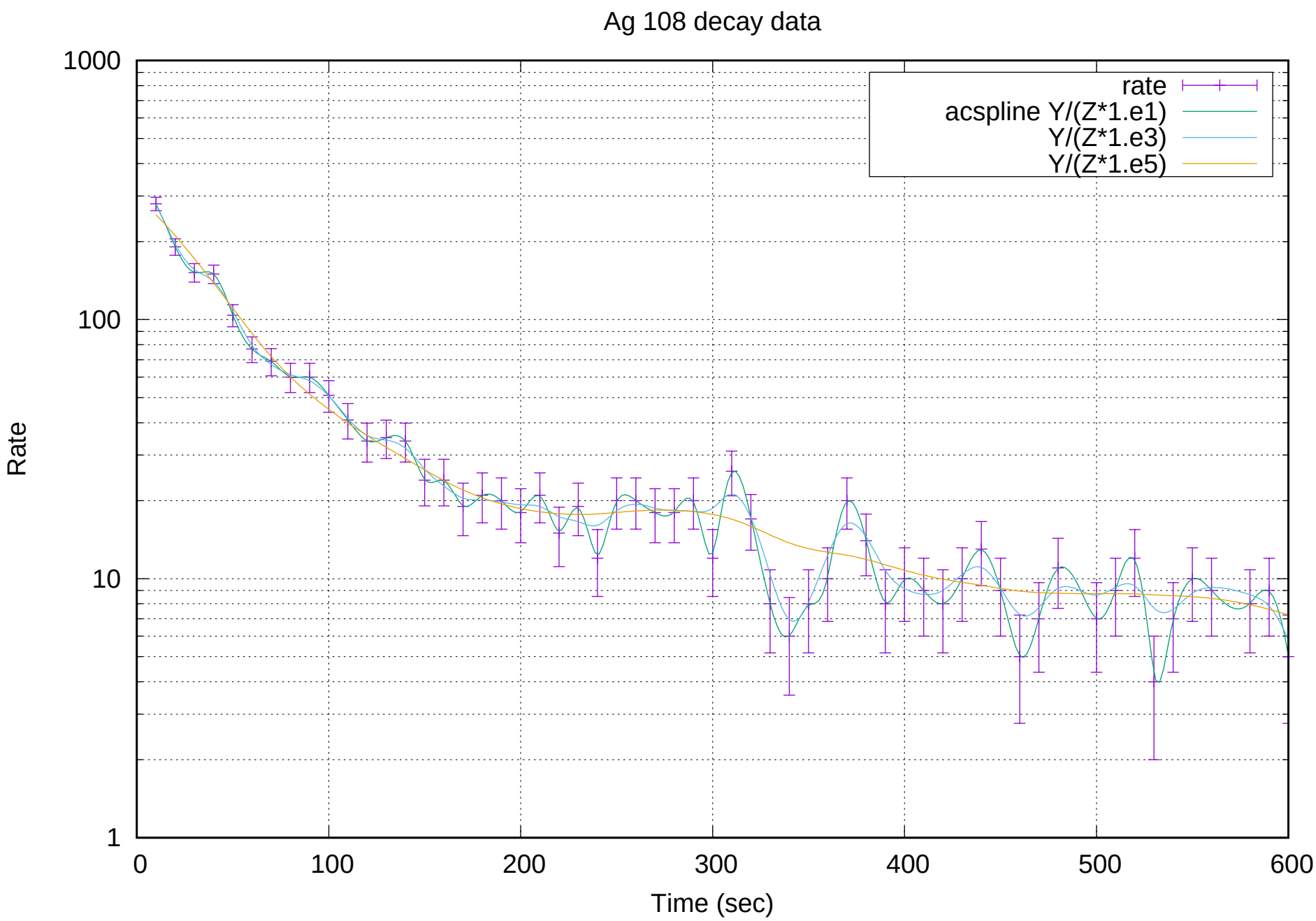

Bragg reflection -- Peak only
Rate
Rate
Amplitude
Angle (deg)
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
26.5
27
27.5
28
28.5
29

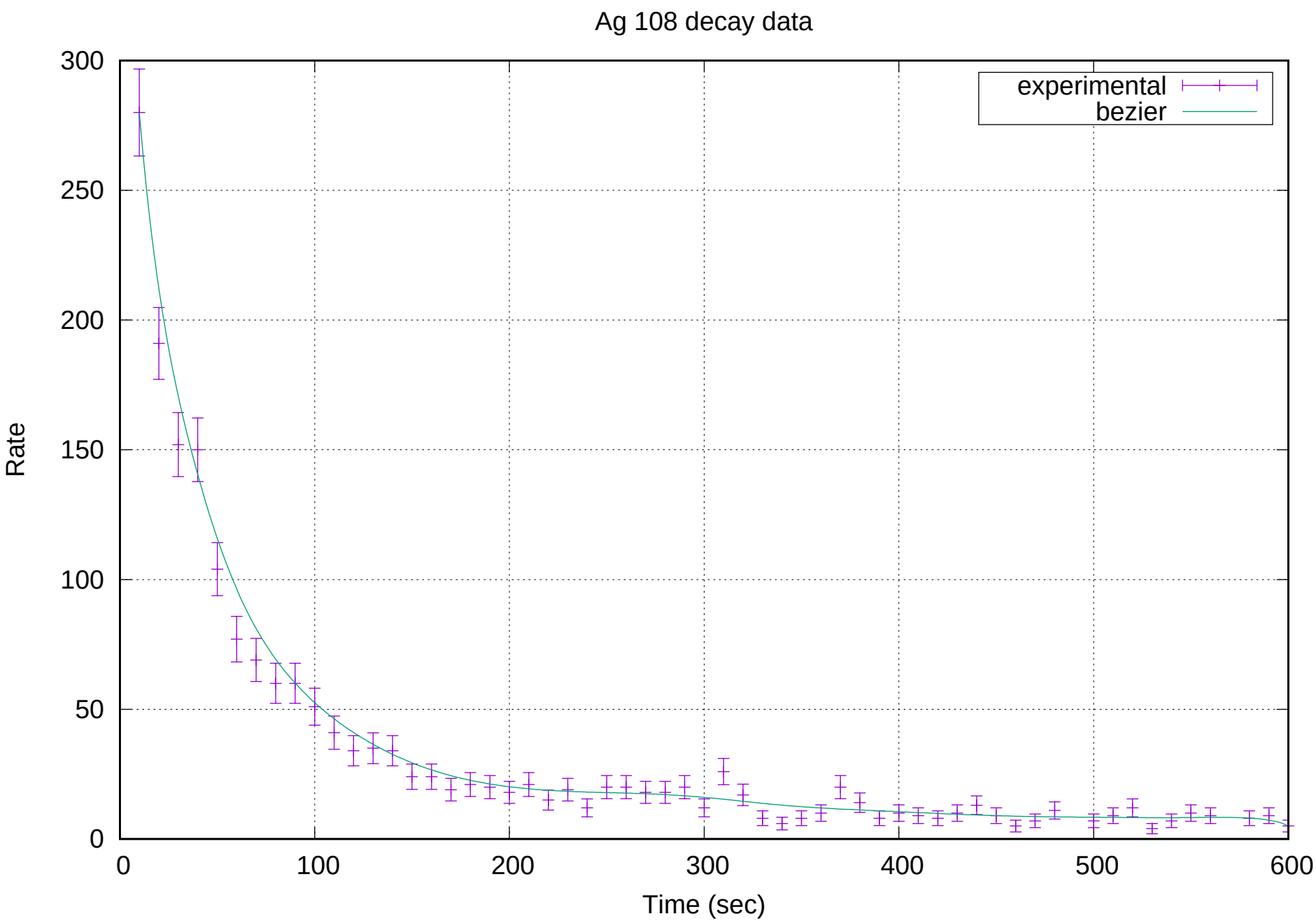

Bragg reflection -- Peak only
Rate
Rate
Amplitude
Angle (deg)
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
26.5
27
27.5
28
28.5
29

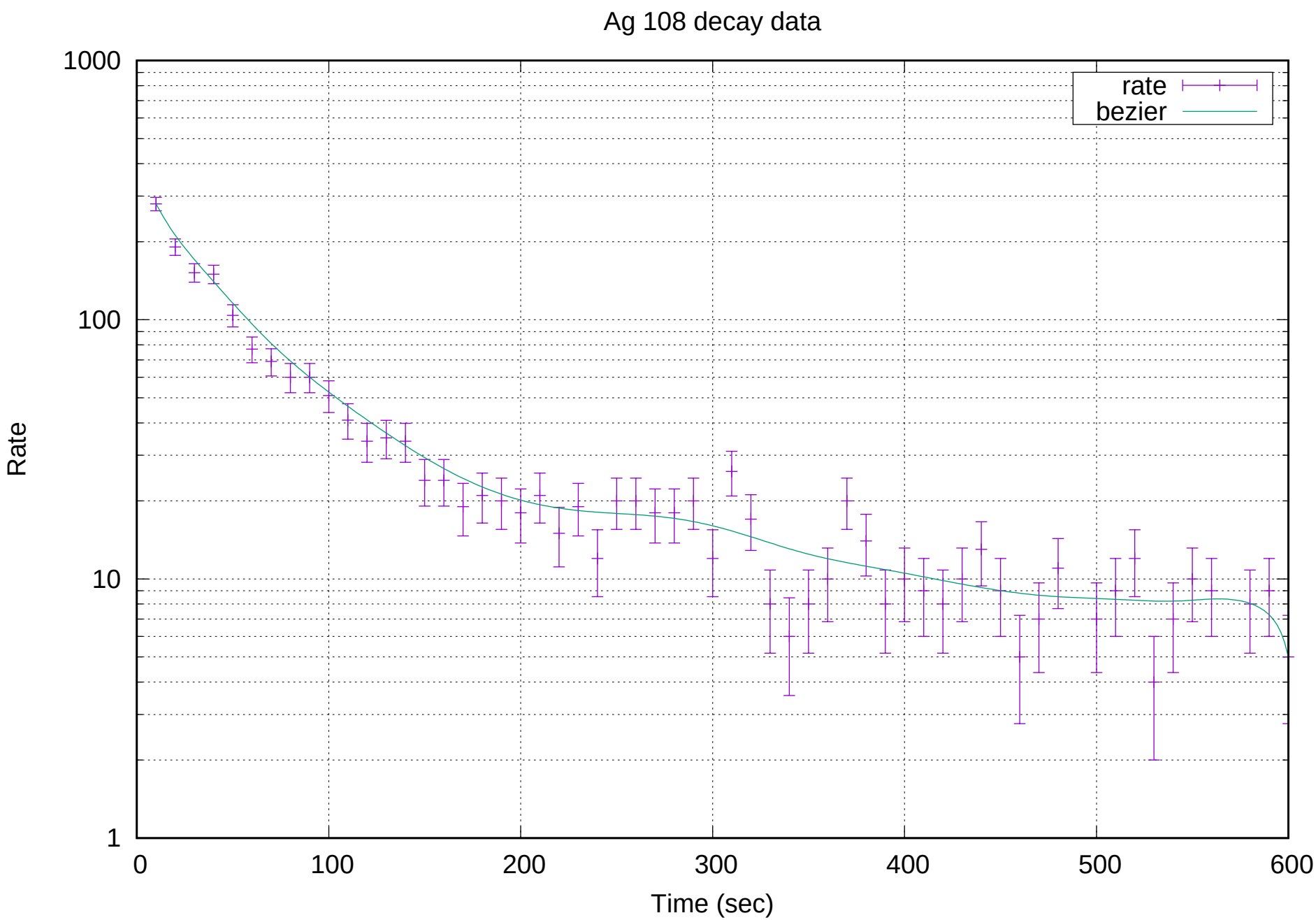

Ag 108 decay data
experimental
cubic smooth
Rate
Time (sec)
0
50
100
150
200
250
300
0
100
200
300
400
500
600

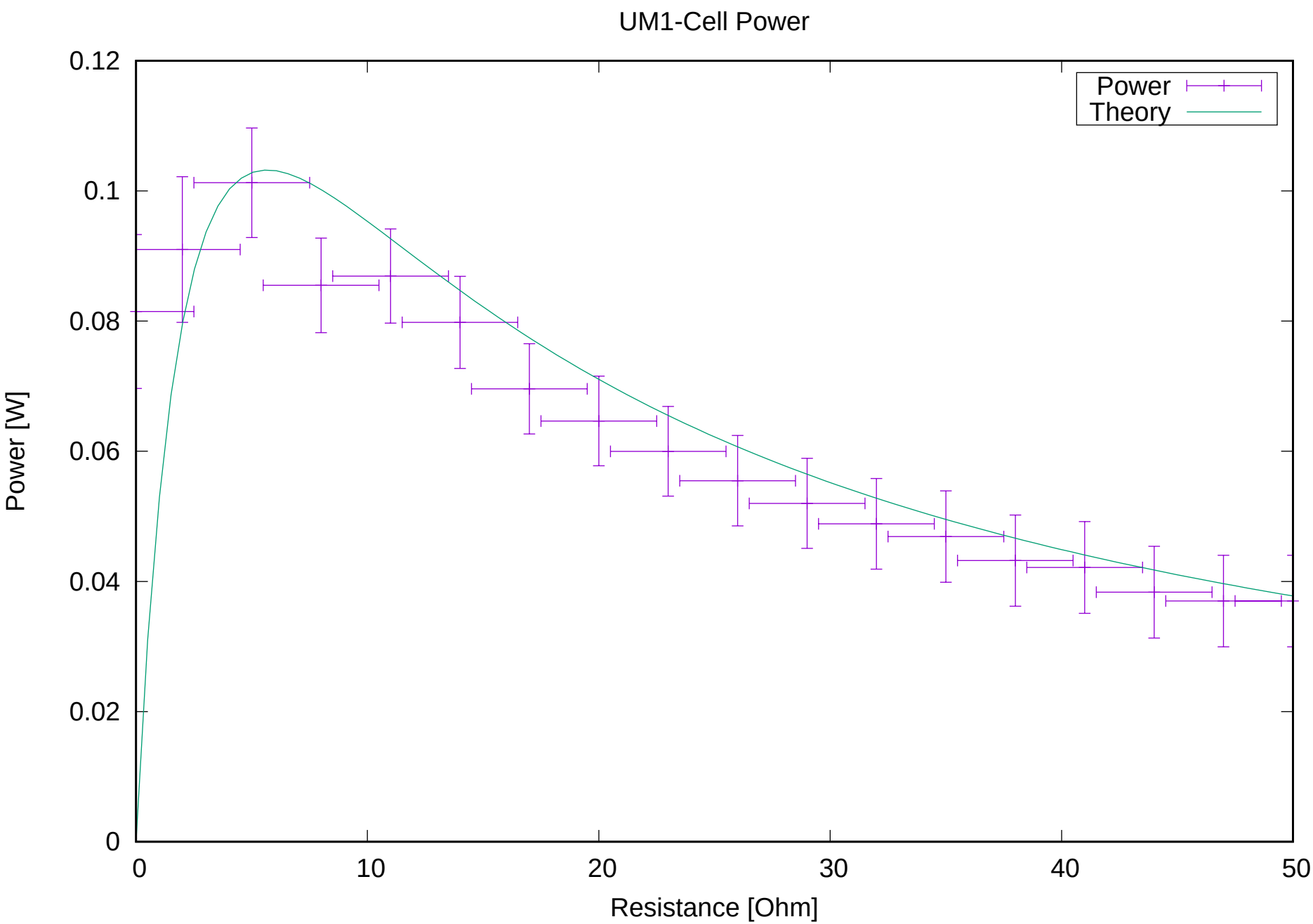

Ag 108 decay data
experimental
acspline Y/Z
Rate
Time (sec)
0
50
100
150
200
250
300
0
100
200
300
400
500
600

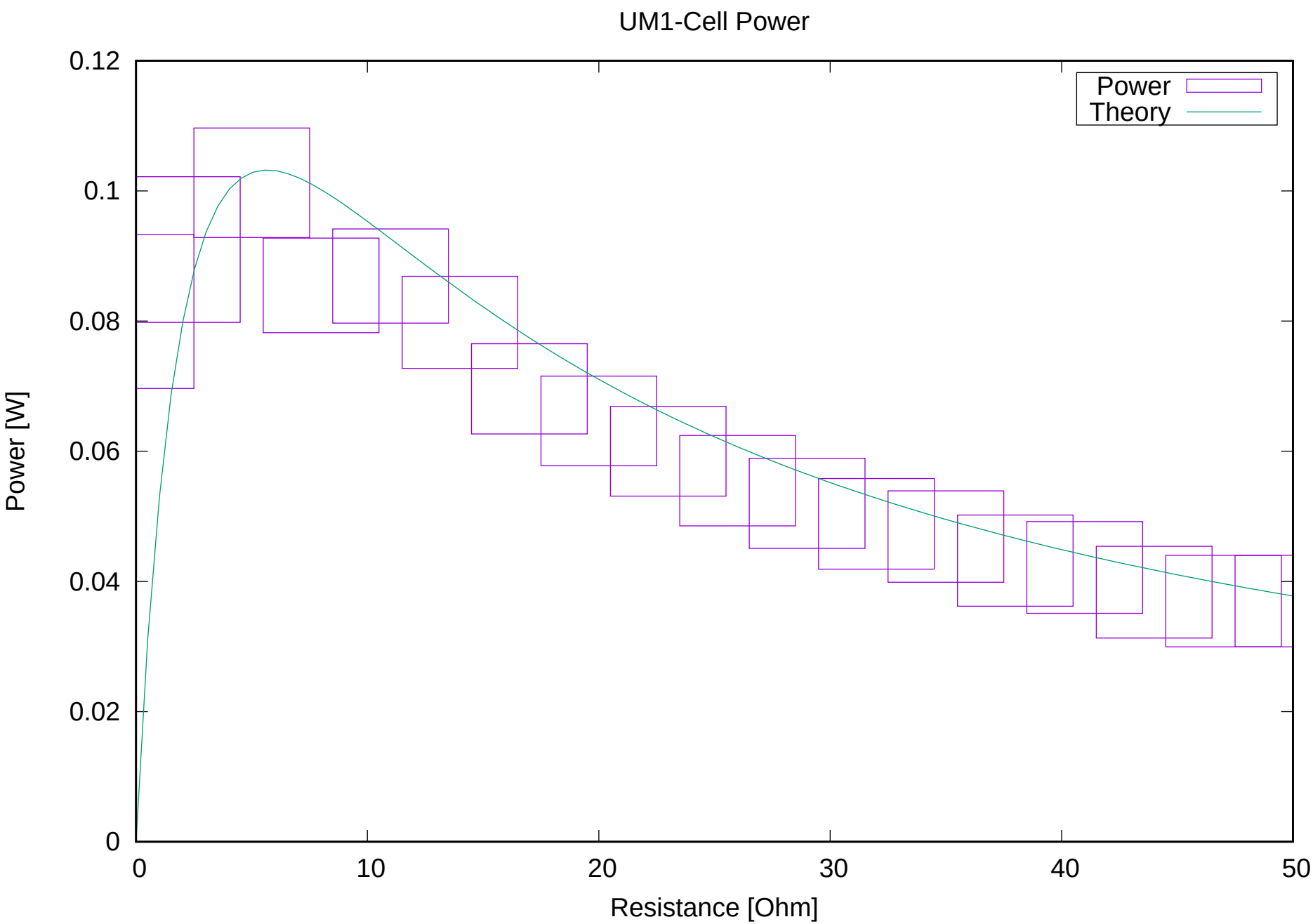

Ag 108 decay data
Rate
Time (sec)
0
50
100
150
200
250
300
0
100
200
300
400
500
600
rate
acspline Y/(Z*1.e1)
Y/(Z*1.e3)
Y/(Z*1.e5)

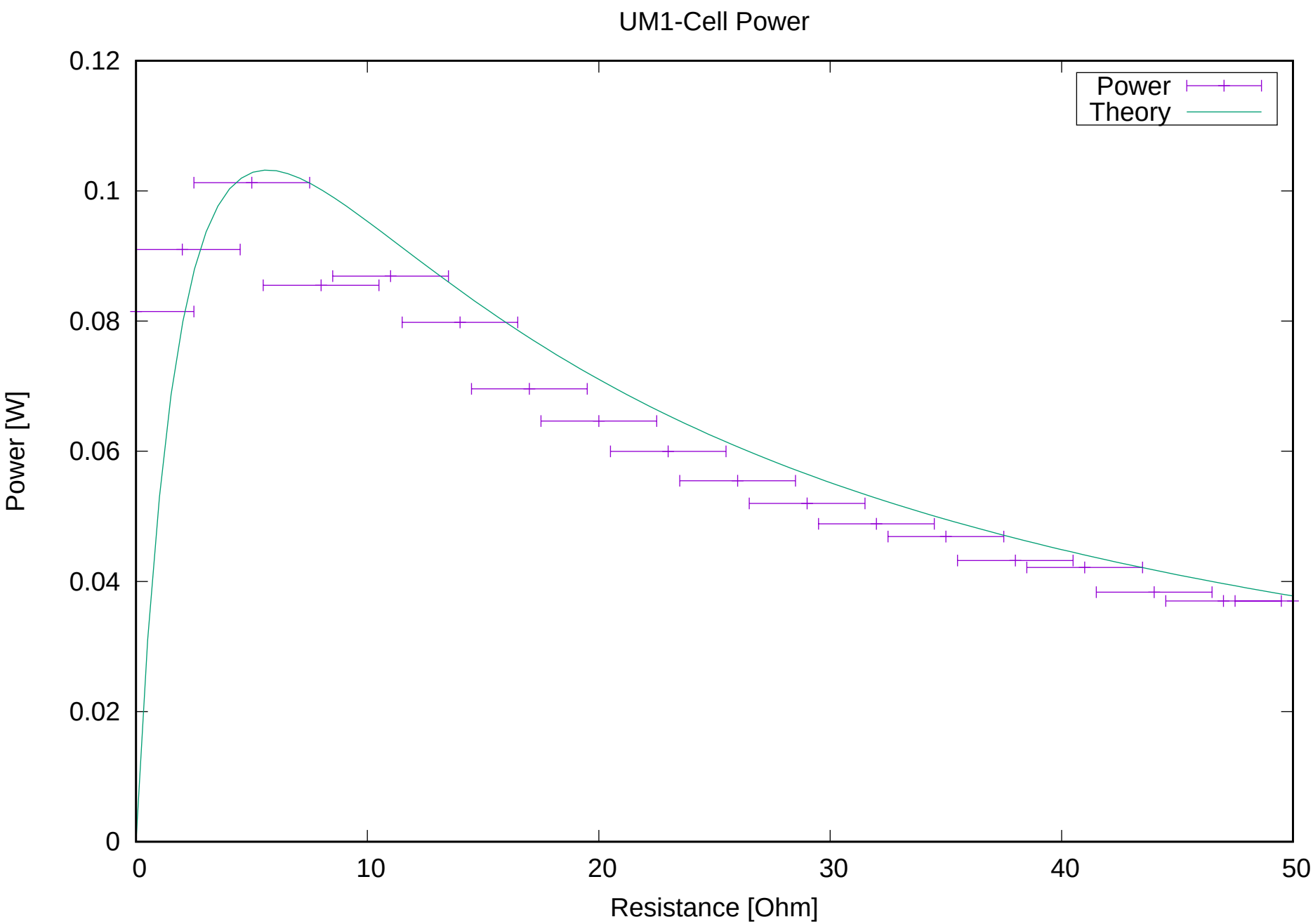

Ag 108 decay data
rate
acspline Y/(Z*1.e1)
Y/(Z*1.e3)
Y/(Z*1.e5)
Rate
Time (sec)
1000
100
10
1
0
100
200
300
400
500
600

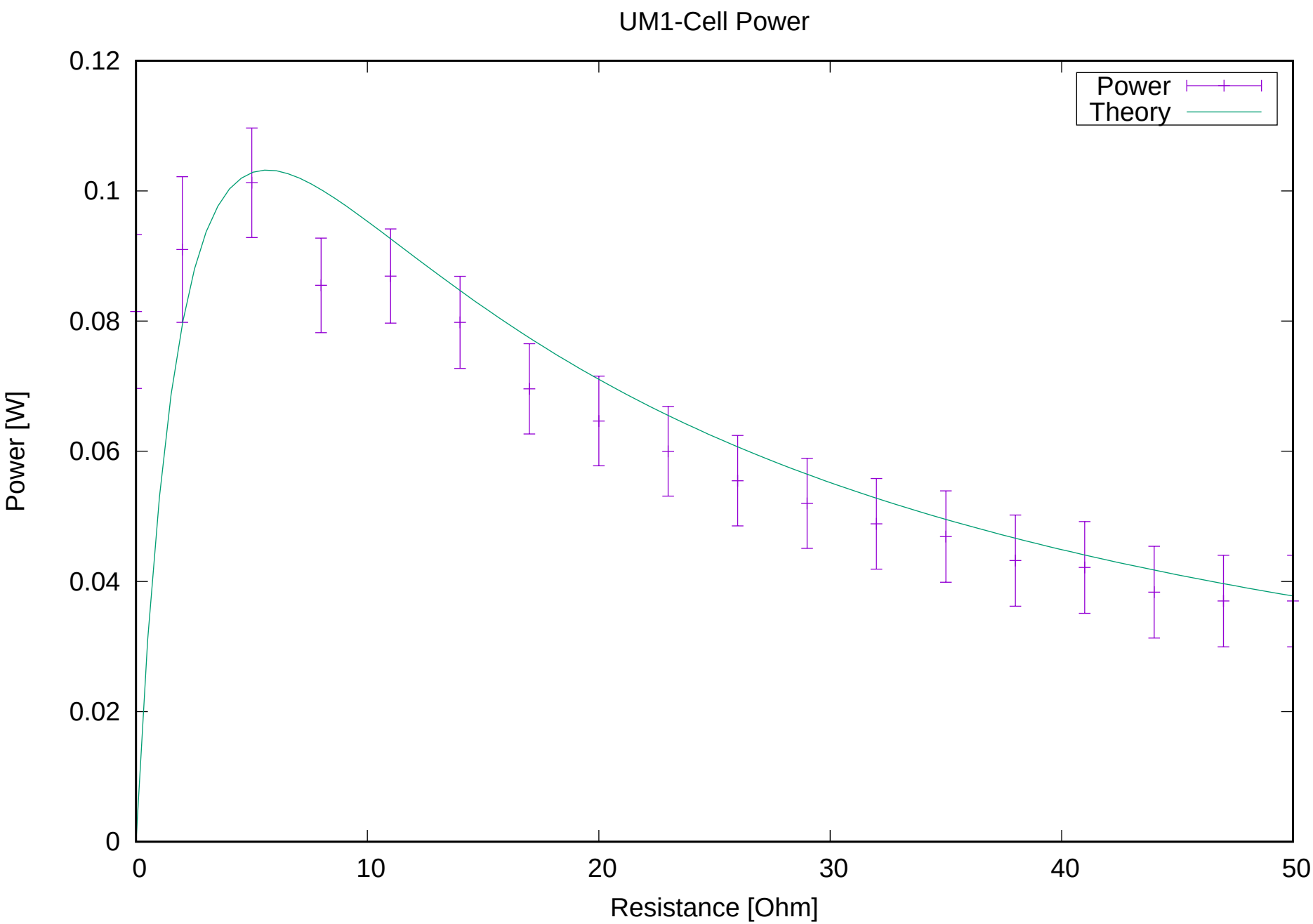

Ag 108 decay data
experimental
bezier
Rate
Time (sec)
0
50
100
150
200
250
300
0
100
200
300
400
500
600

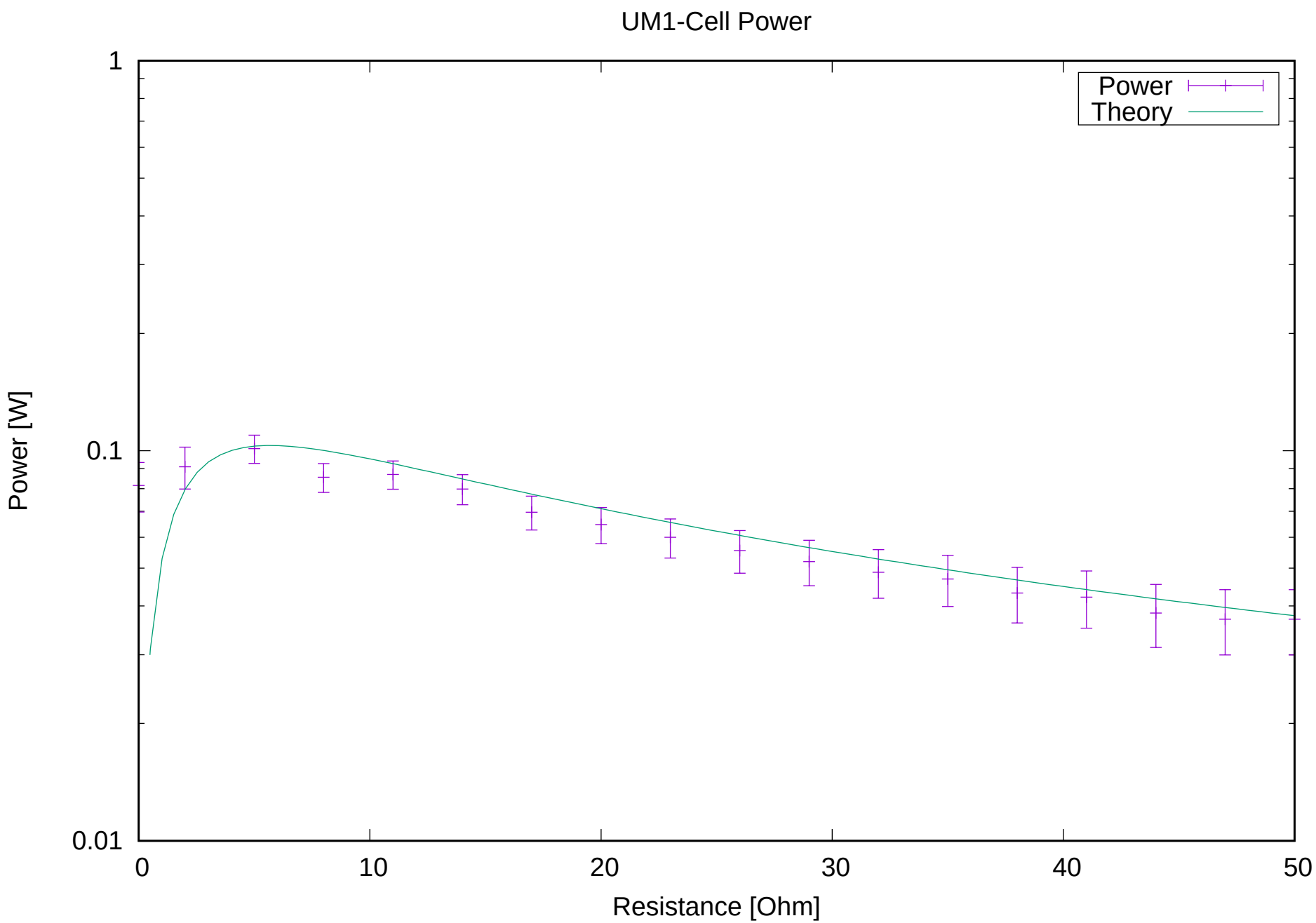

Ag 108 decay data
rate
bezier
1000
100
10
1
Rate
0
100
200
300
400
500
600
Time (sec)

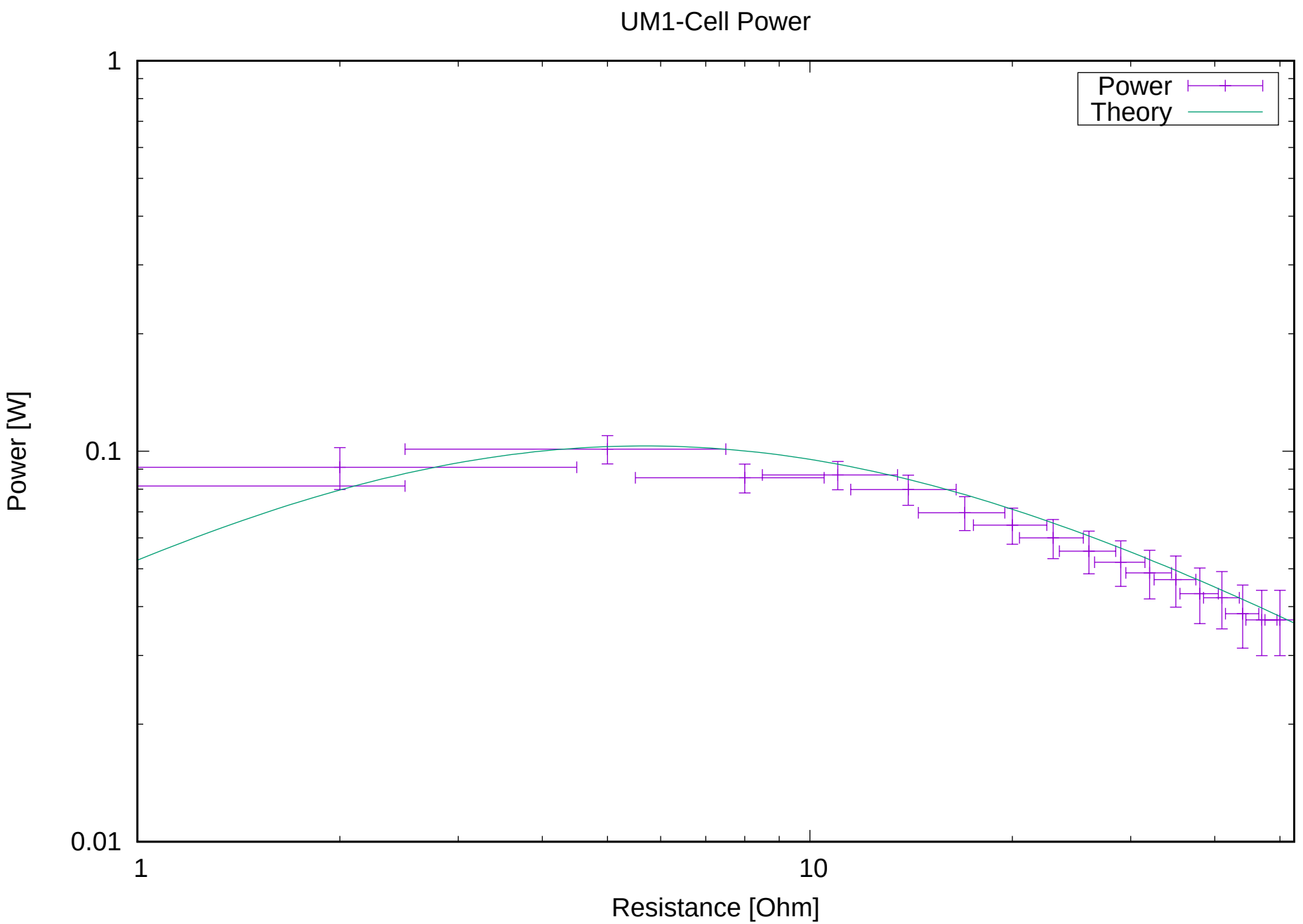

UM1-Cell Power
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

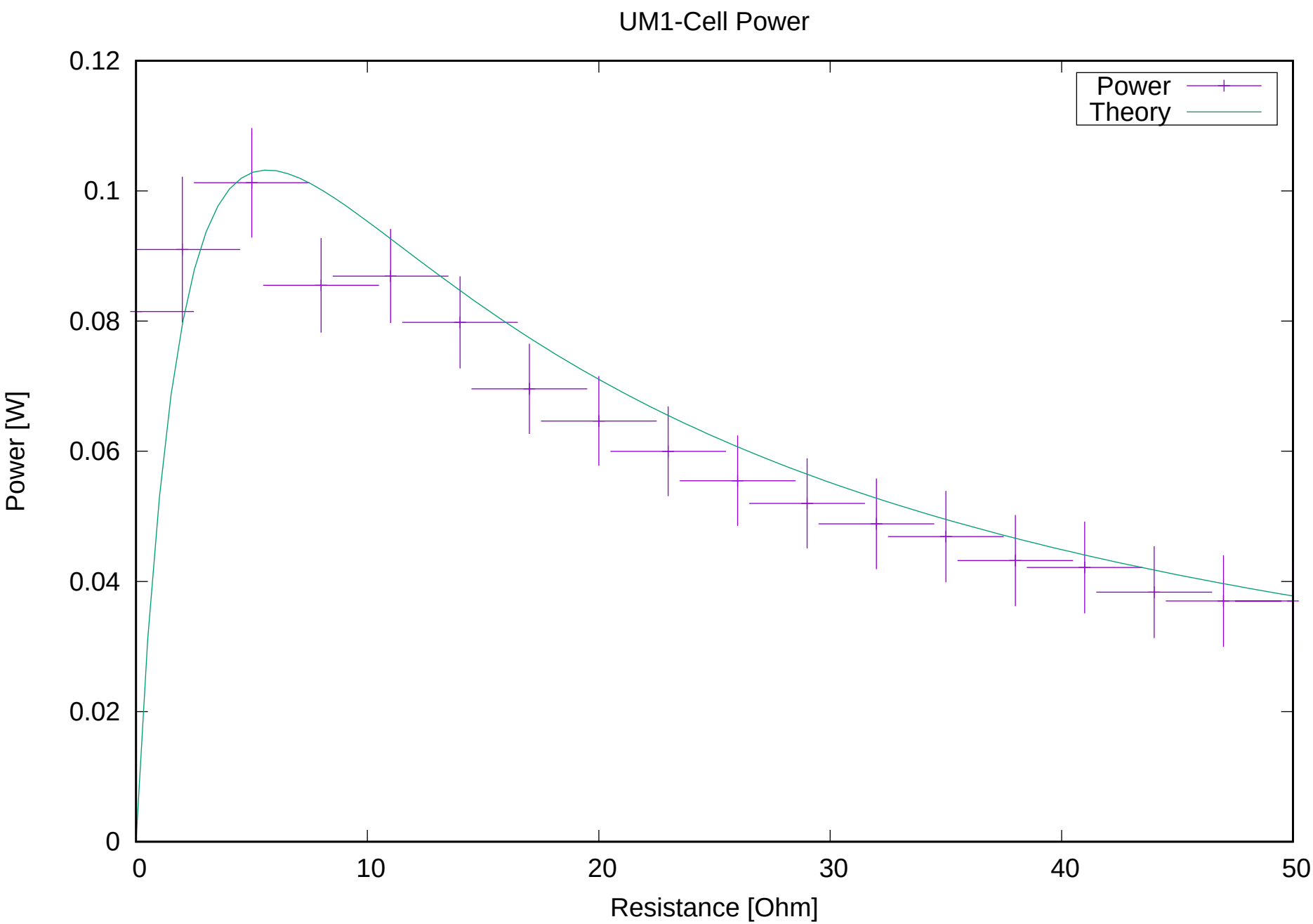

UM1-Cell Power
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

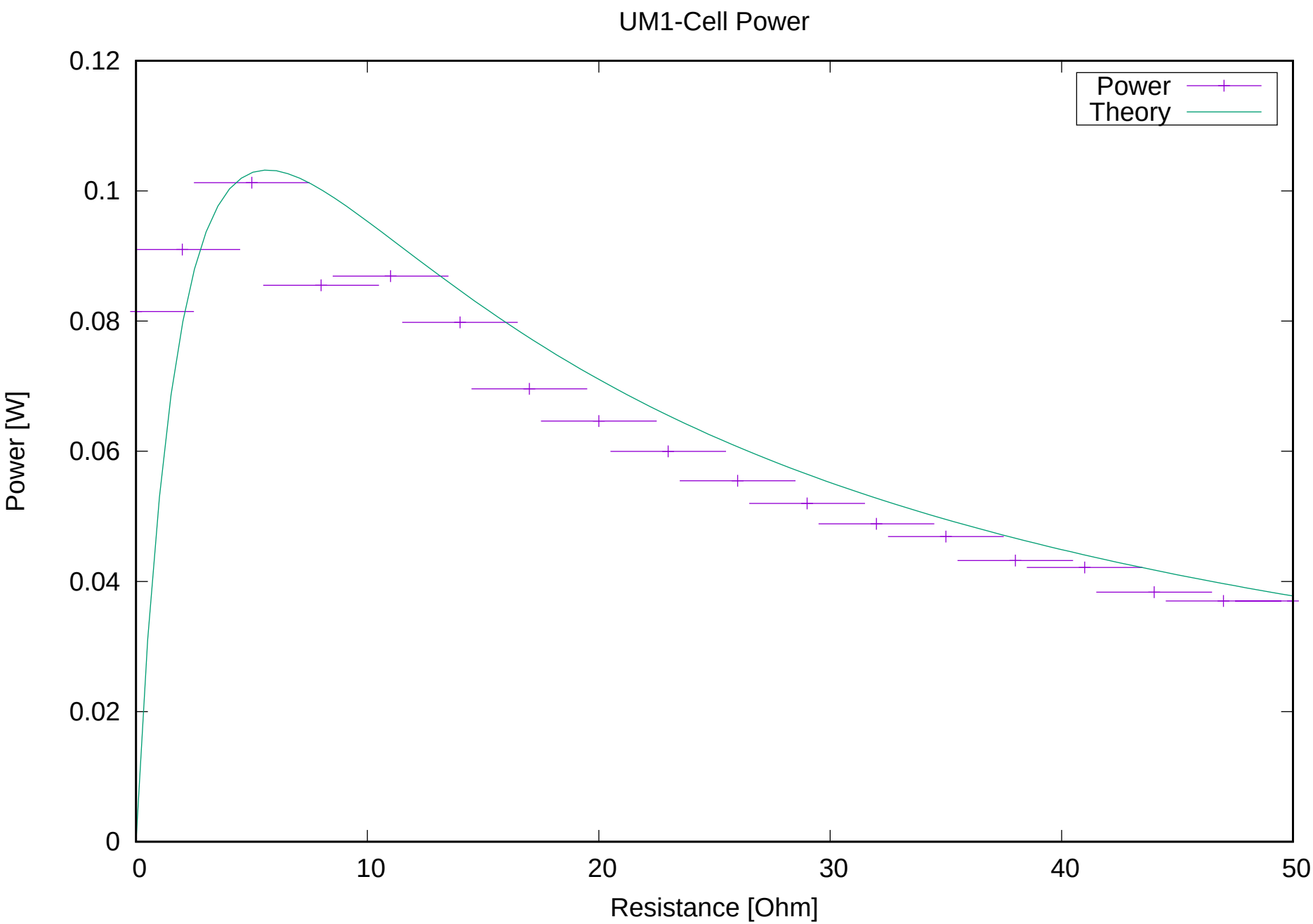

UM1-Cell Power
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

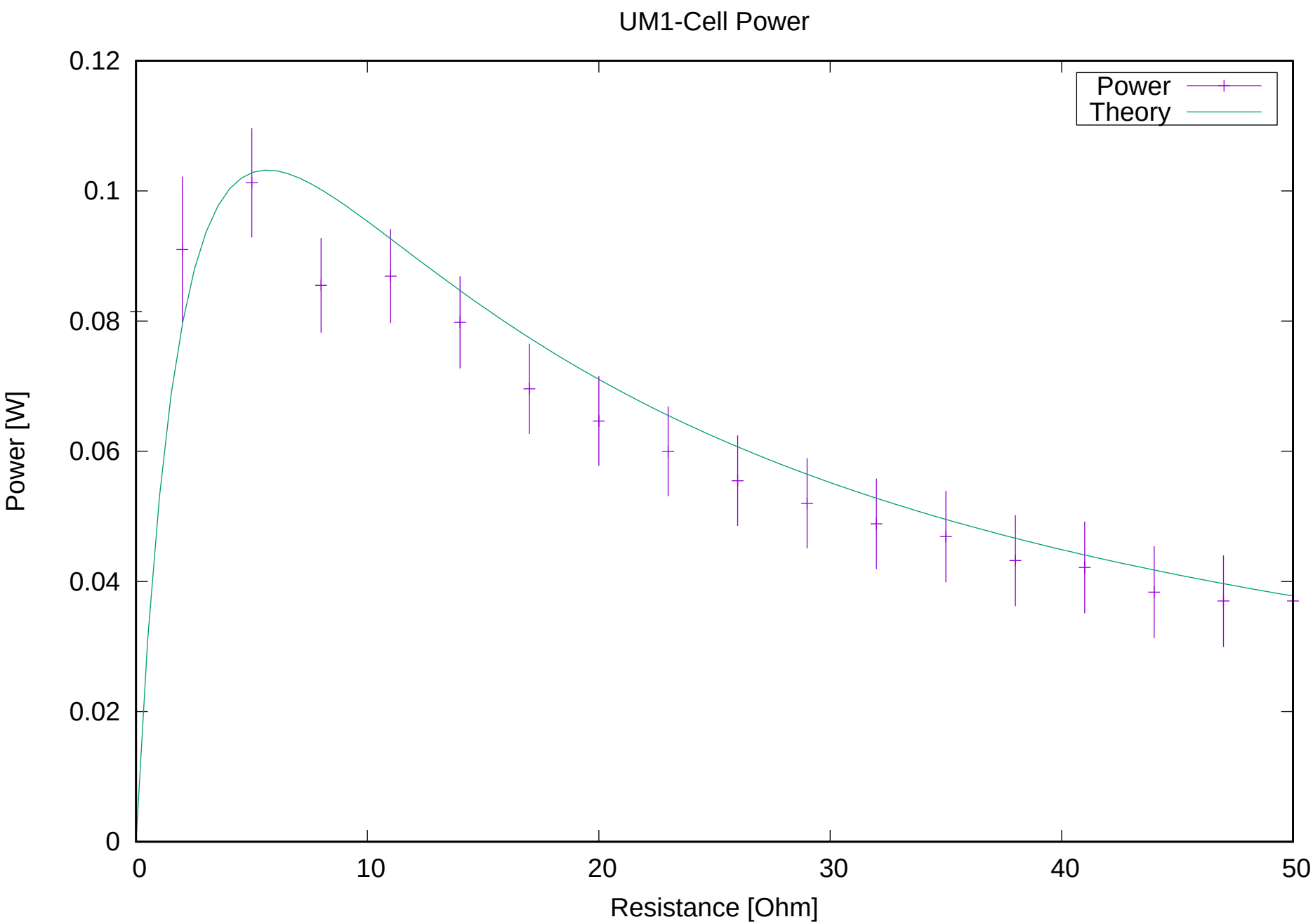

UM1-Cell Power
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

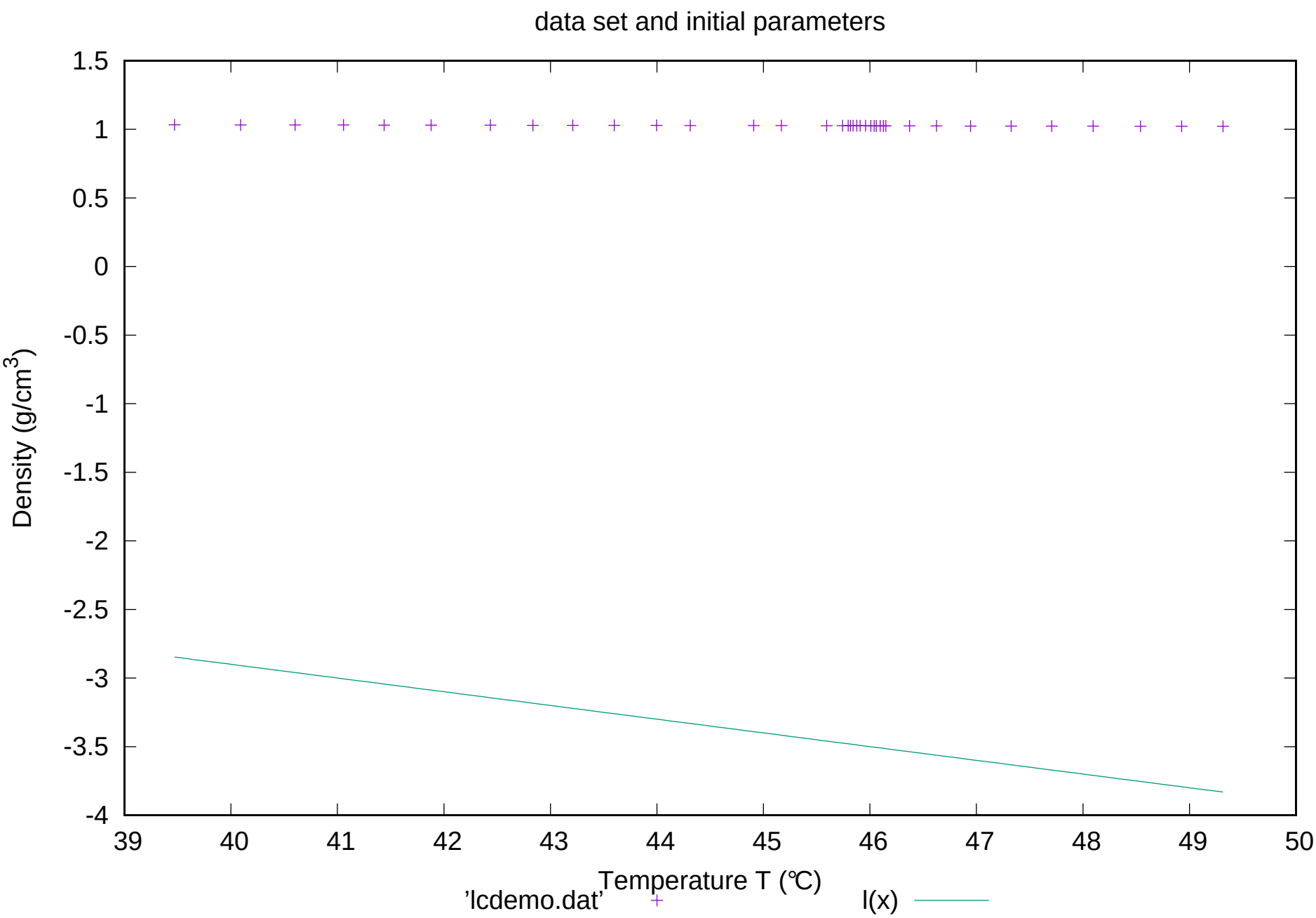

UM1-Cell Power
Power
Theory
Power [W]
1
0.1
0.01
0
10
20
30
40
50
Resistance [Ohm]

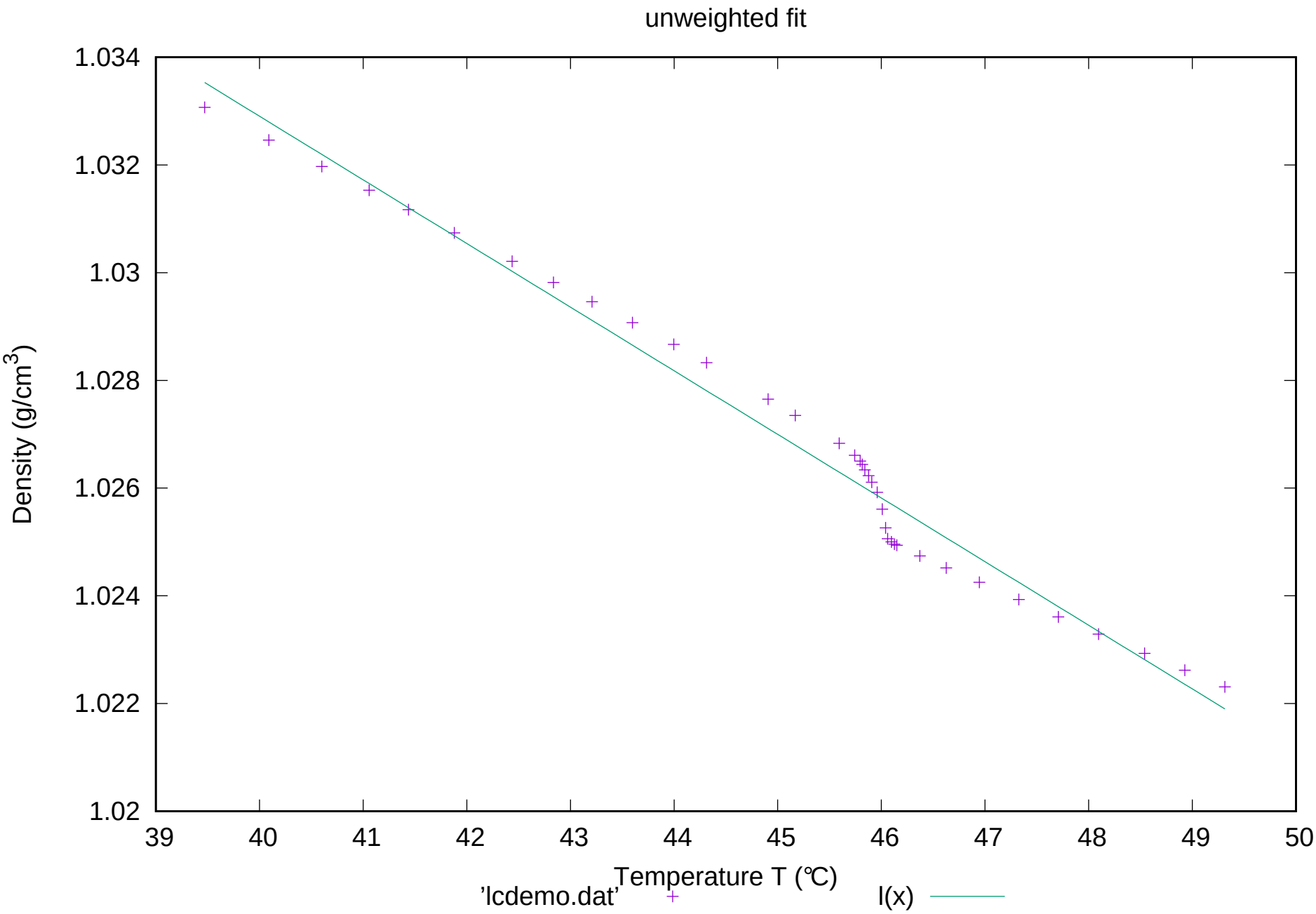

UM1-Cell Power
Power
Theory
1
0.1
0.01
Power [W]
1
10
Resistance [Ohm]

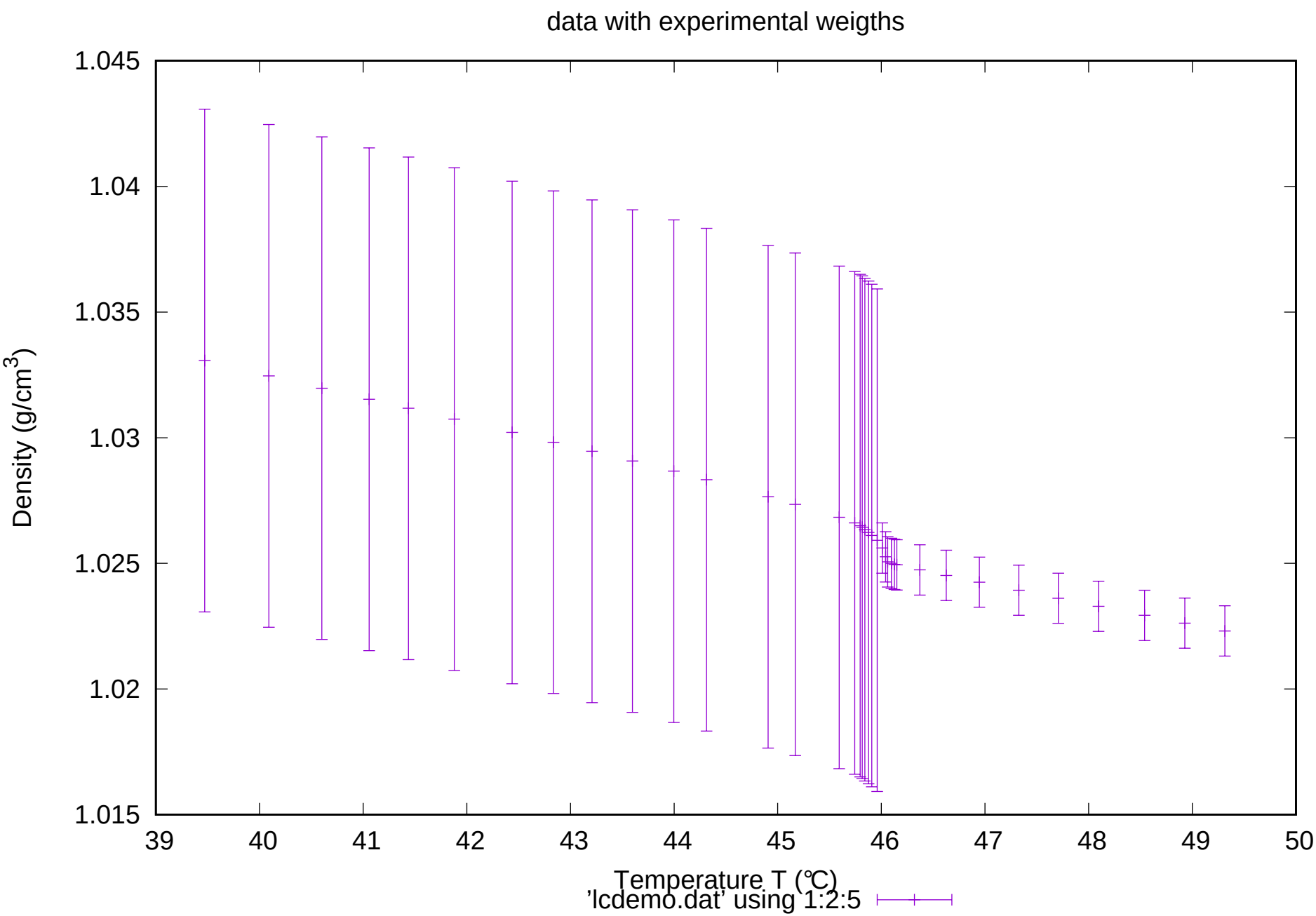

UM1-Cell Power
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

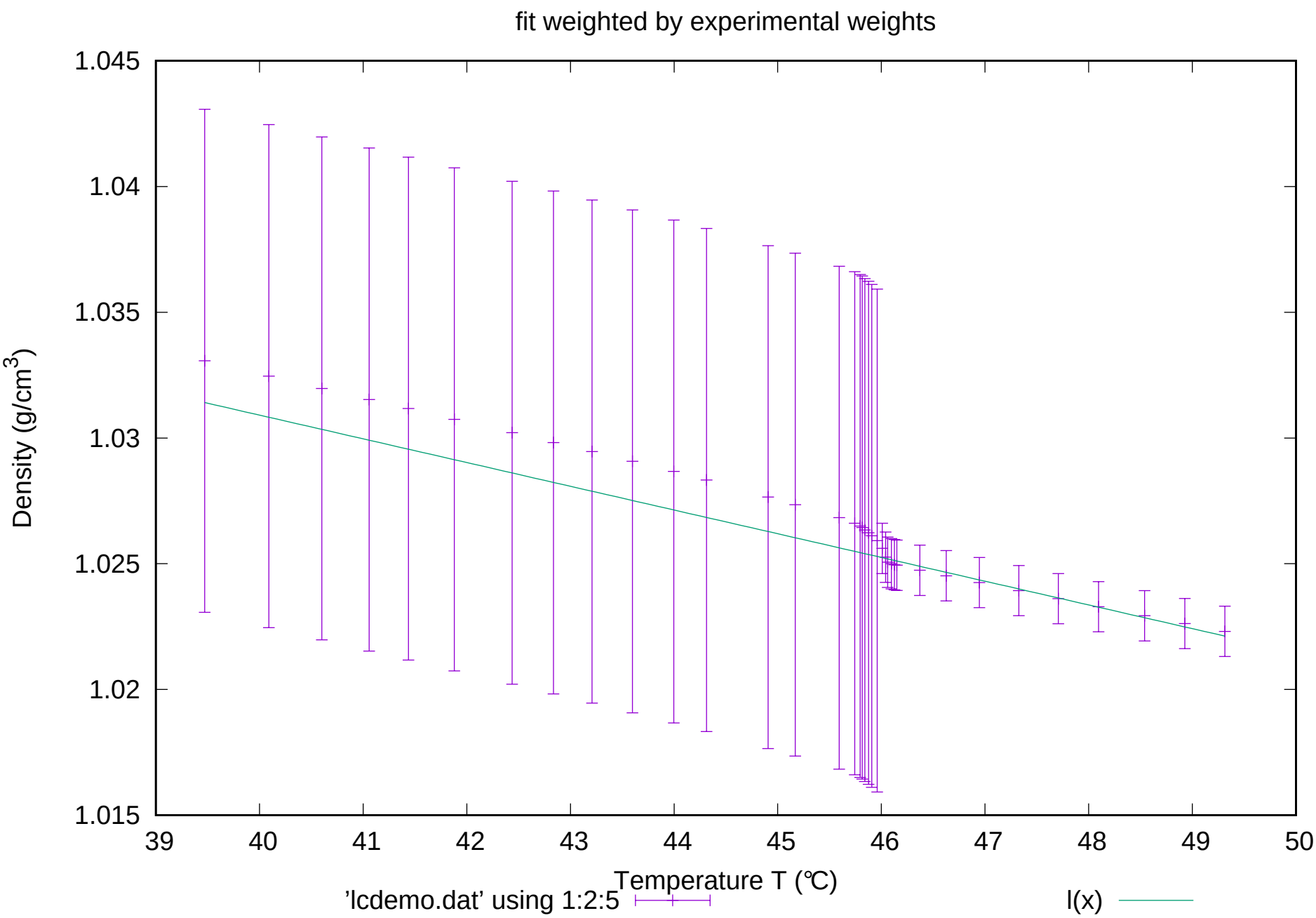

UM1-Cell Power
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

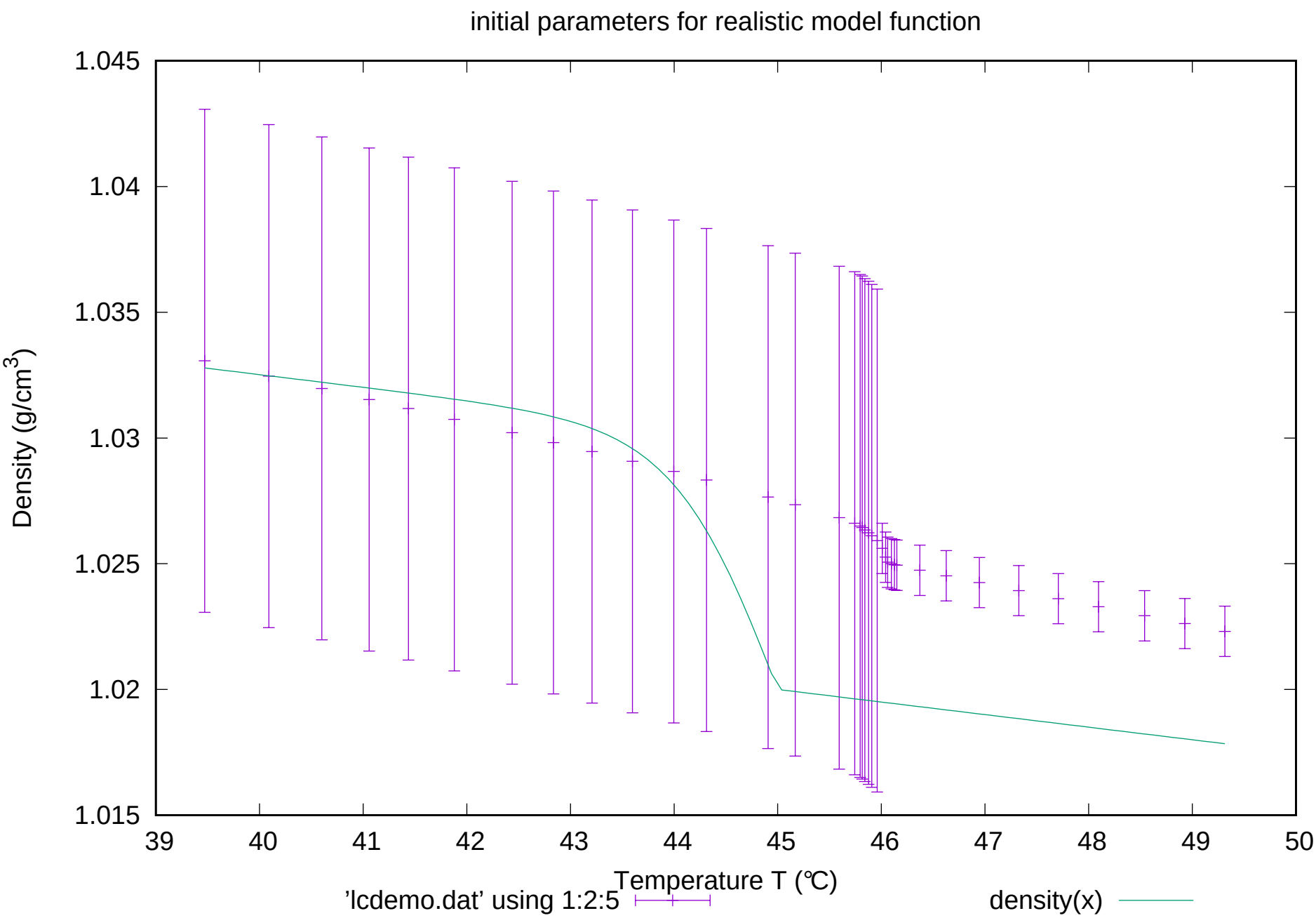

UM1-Cell Power
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

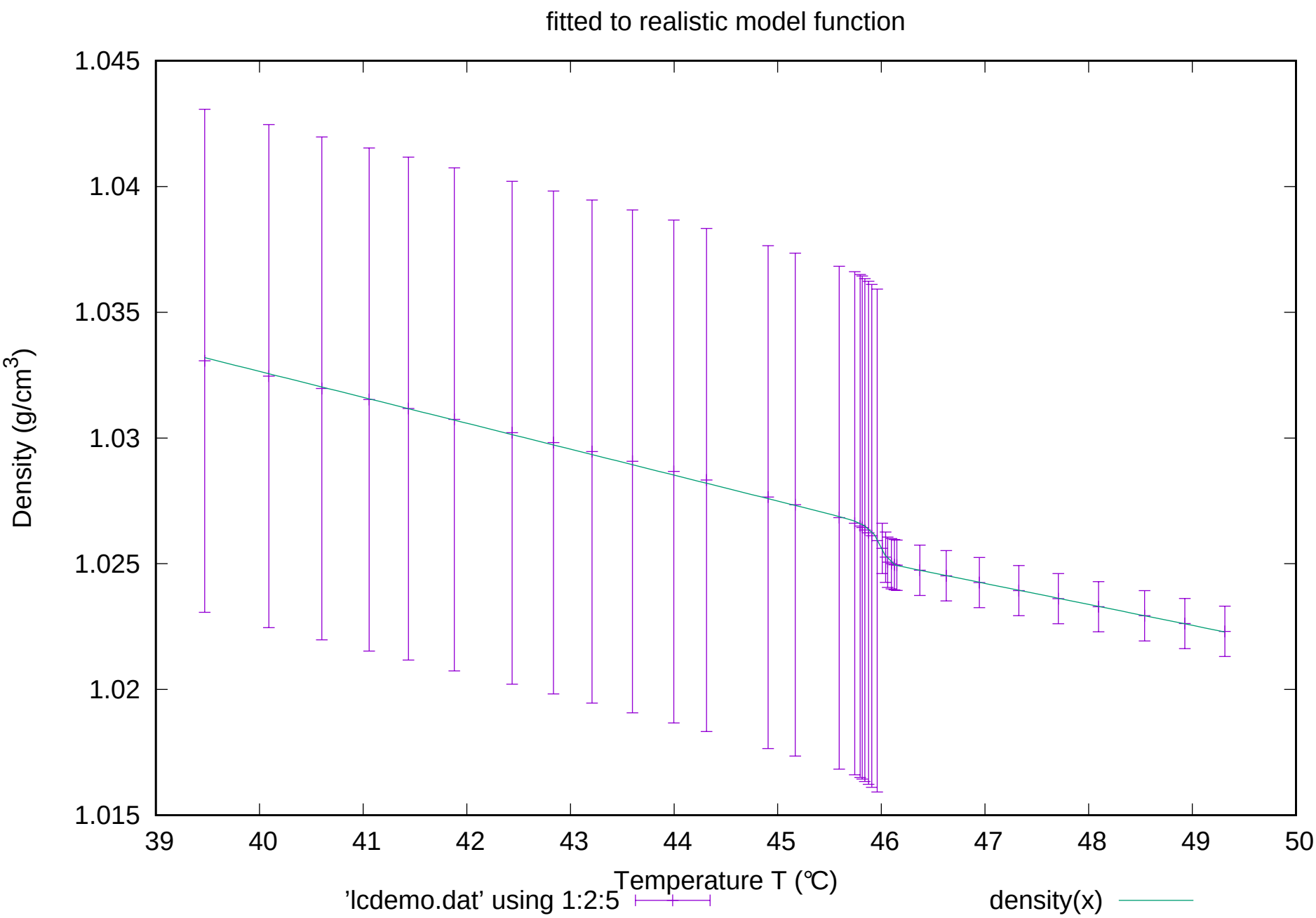

data set and initial parameters
Density (g/cm^3)
Temperature T (℃)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
39
40
41
42
43
44
45
46
47
48
49
50
'lcdemo.dat'
l(x)


unweighted fit
1.034
1.032
1.03
1.028
1.026
1.024
1.022
1.02
Density (g/cm3)
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T (℃)
'lcdemo.dat'
l(x)

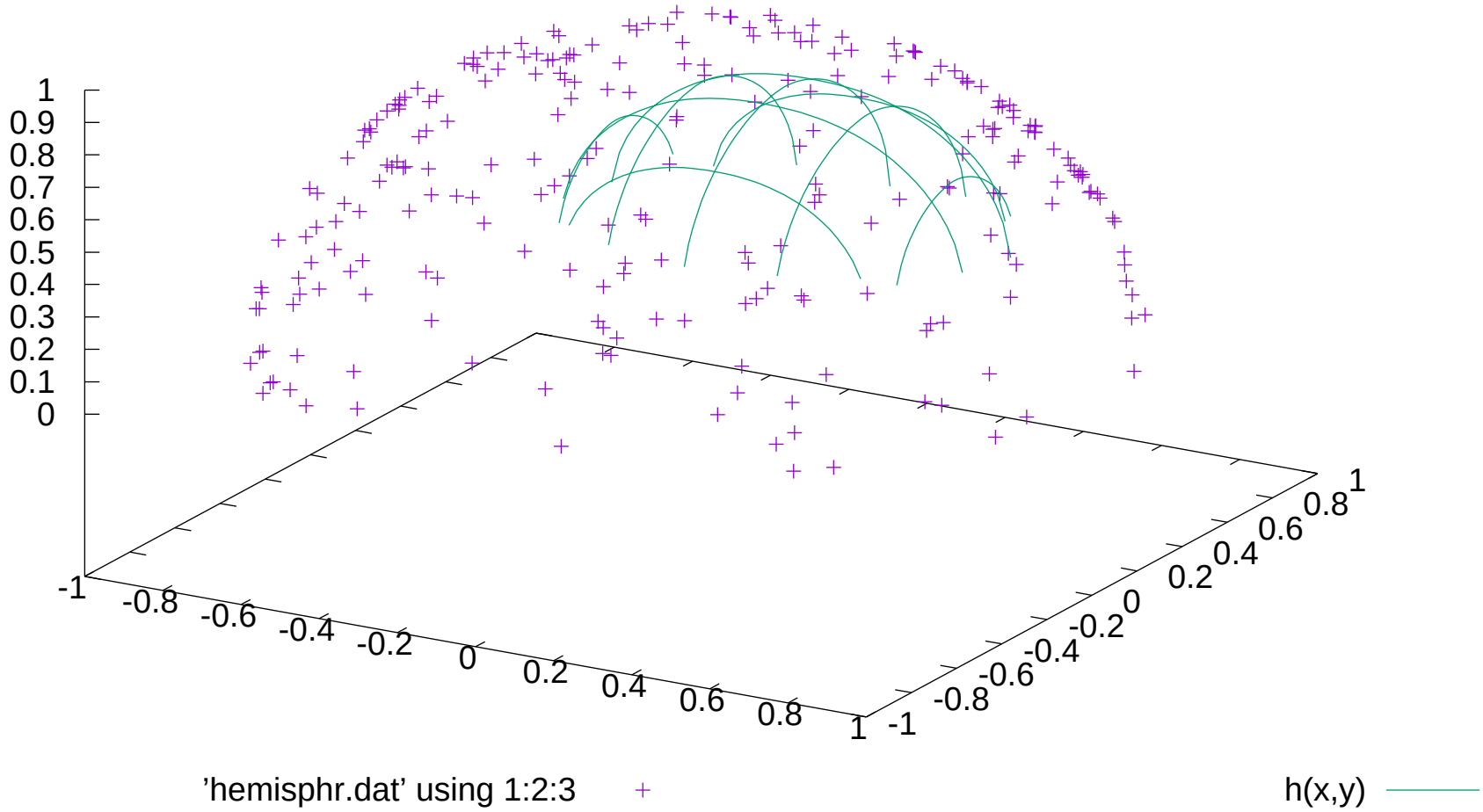

data with experimental weigths
1.045
1.04
1.035
1.03
1.025
1.02
1.015
Density (g/cm3)
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T (℃)
'lcdemo.dat' using 1:2:5

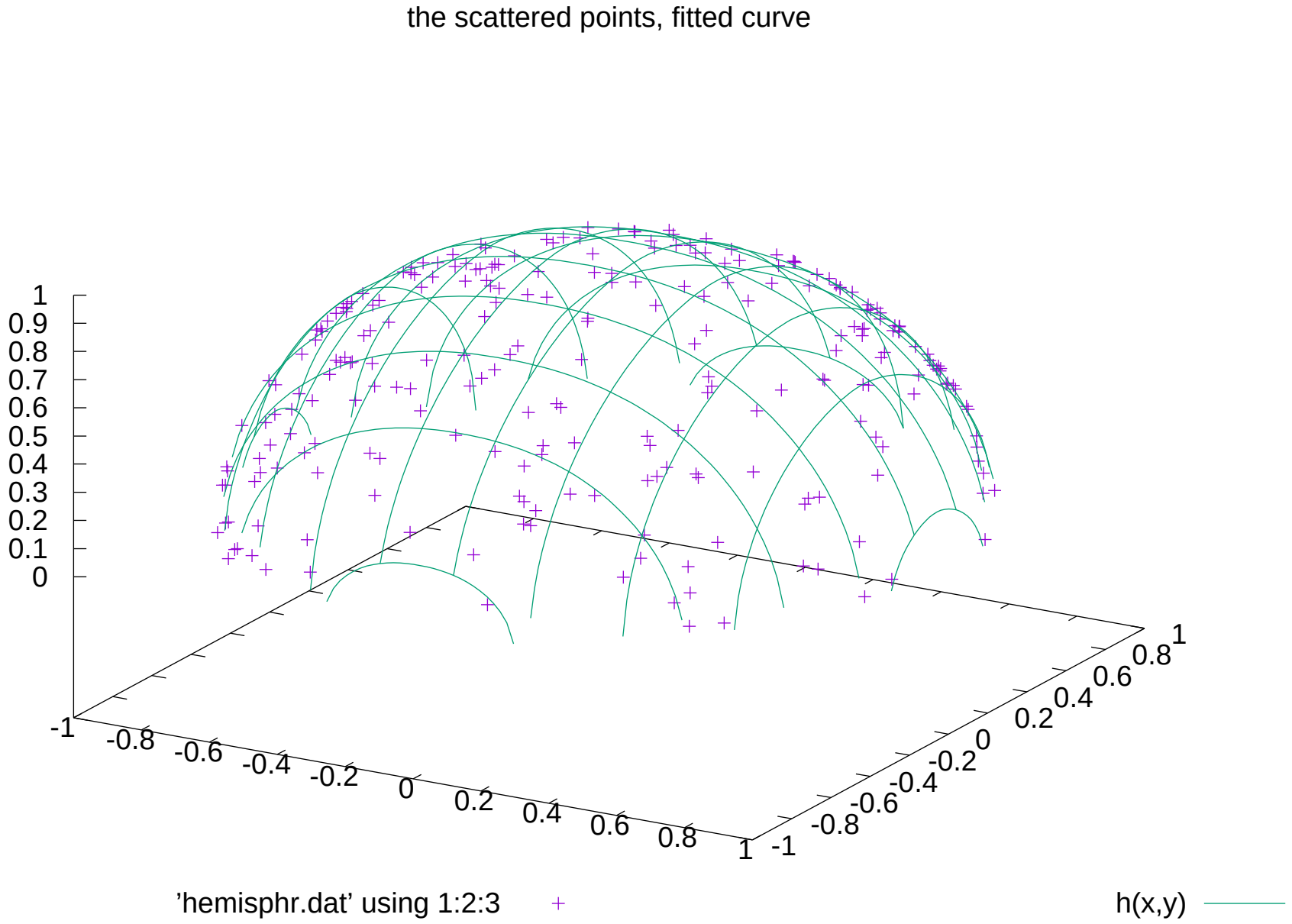

fit weighted by experimental weights
1.045
1.04
1.035
1.03
1.025
1.02
1.015
Density (g/cm3)
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T (℃)
'lcdemo.dat' using 1:2:5
I(x)

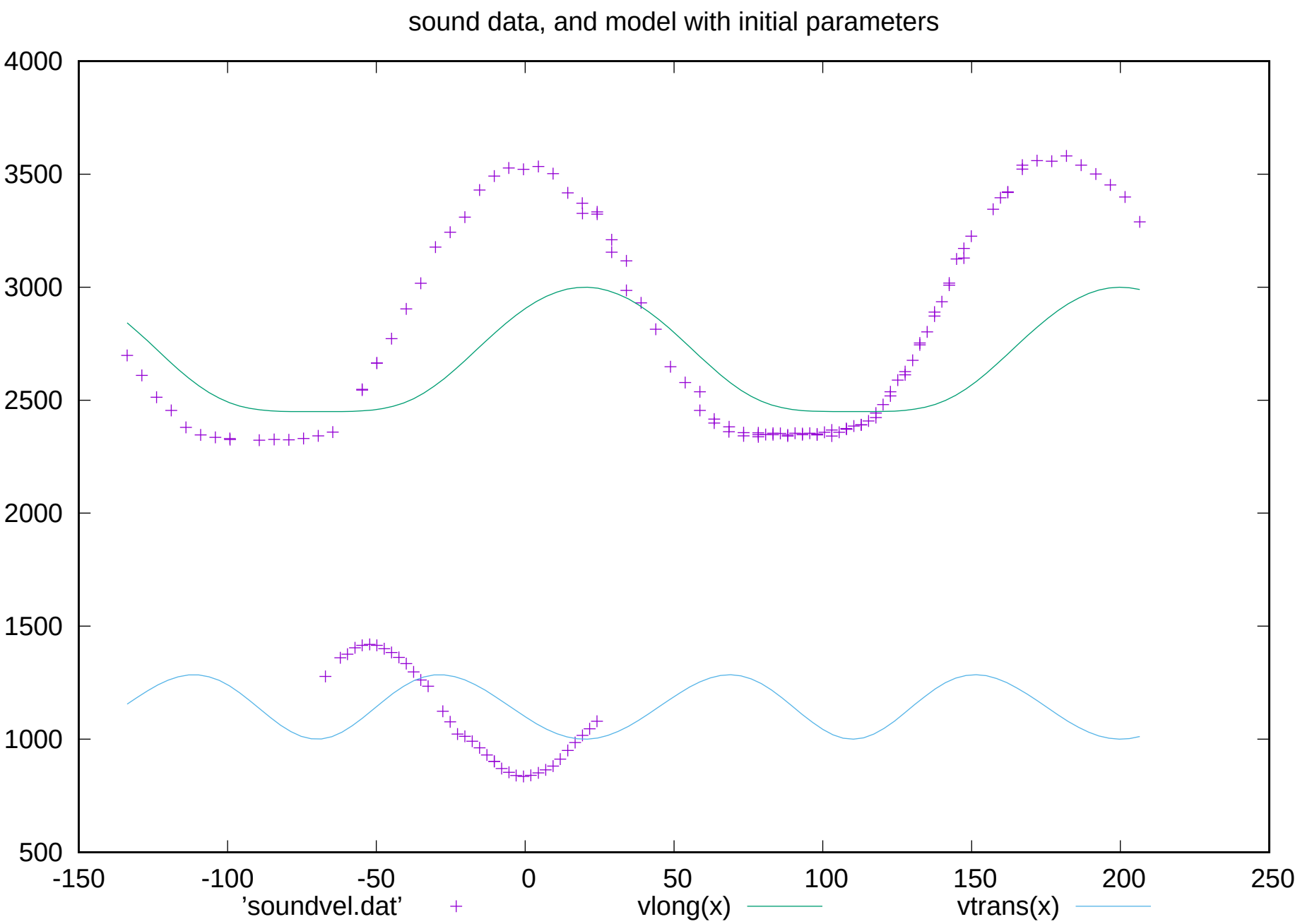

initial parameters for realistic model function
1.045
1.04
1.035
1.03
1.025
1.02
1.015
Density (g/cm3)
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T (℃)
'lcdemo.dat' using 1:2:5
density(x)

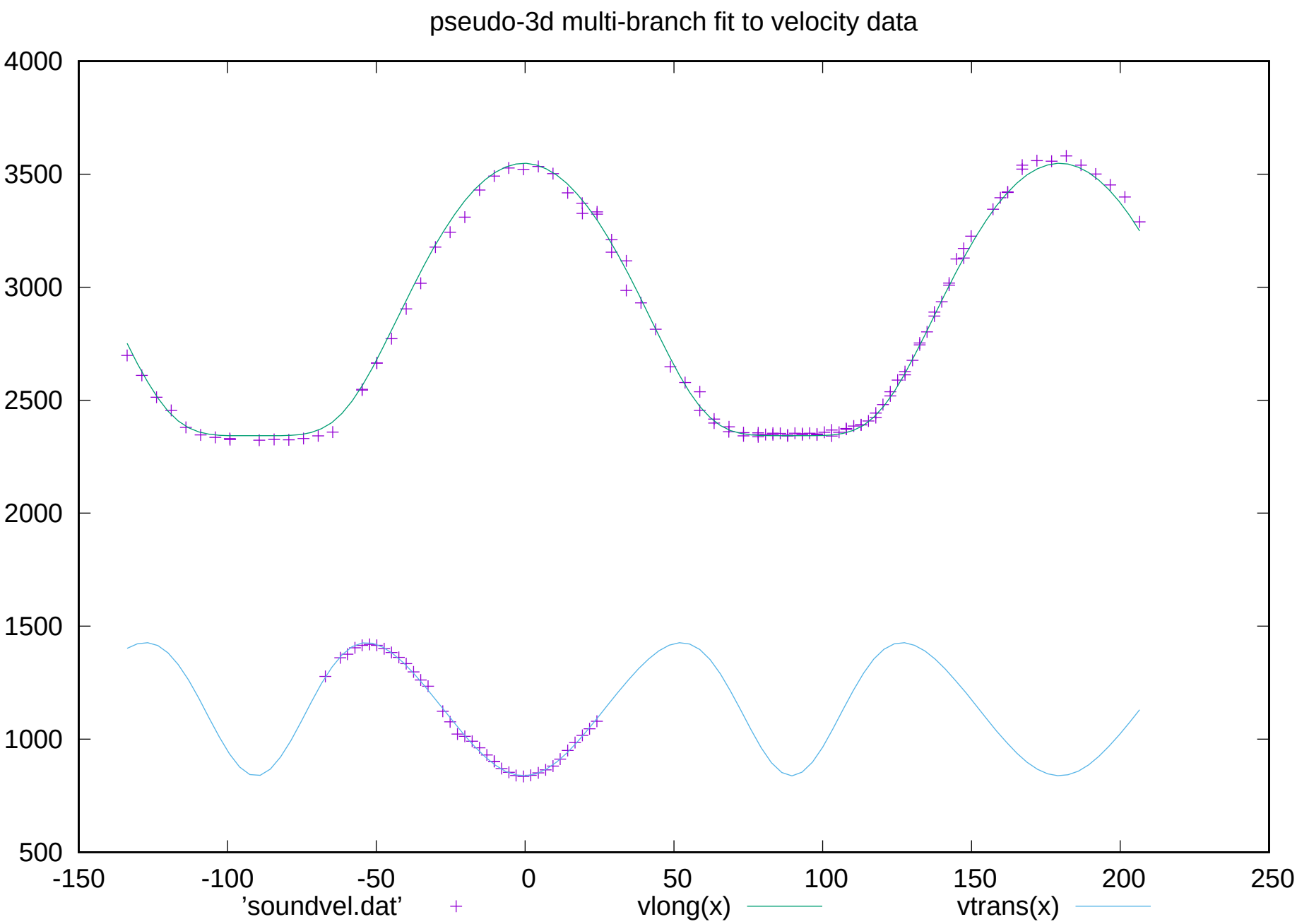

fitted to realistic model function
1.045
1.04
1.035
1.03
1.025
1.02
1.015
Density (g/cm3)
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T (℃)
'lcdemo.dat' using 1:2:5
density(x)

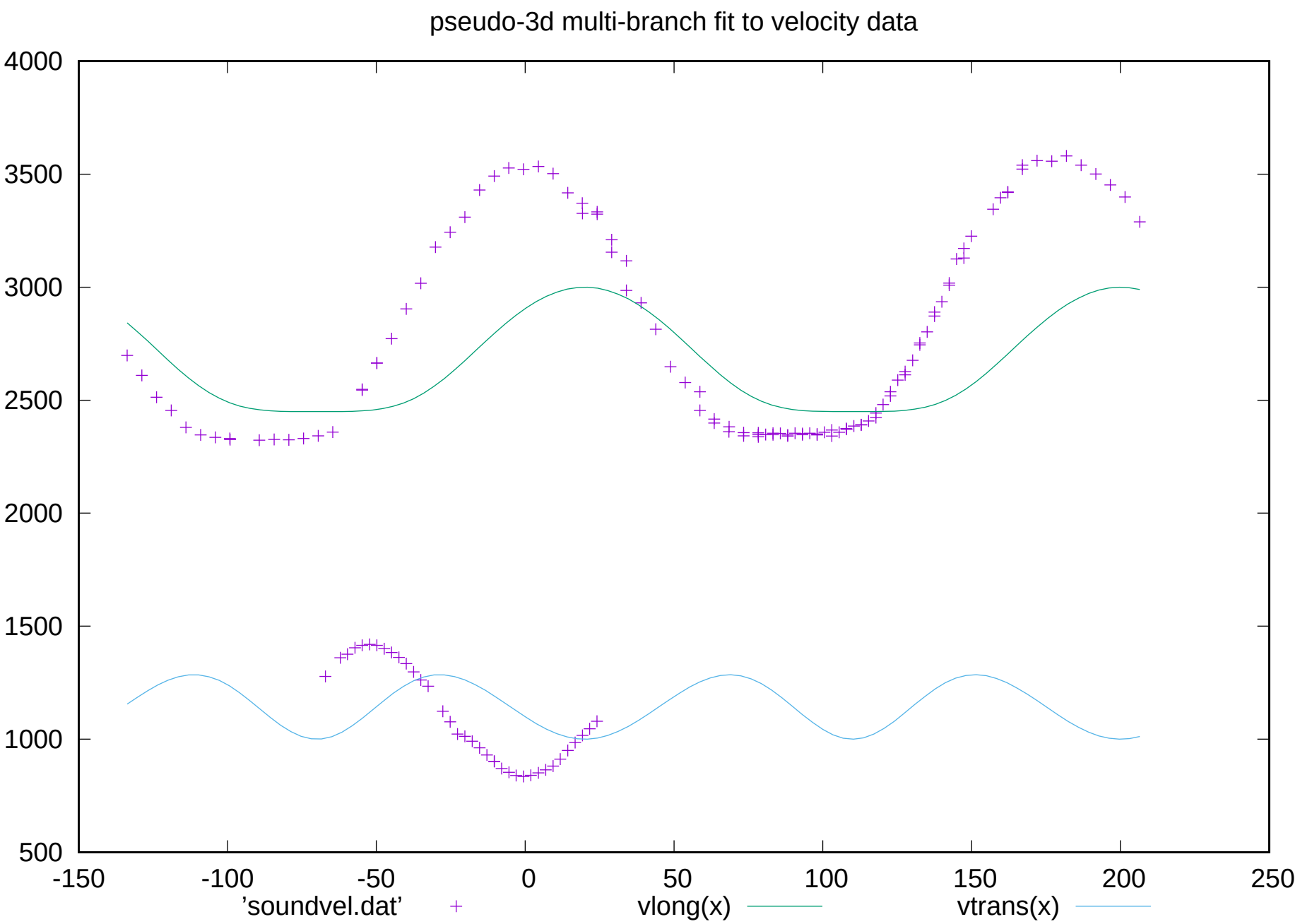
the scattered points, and the initial parameter

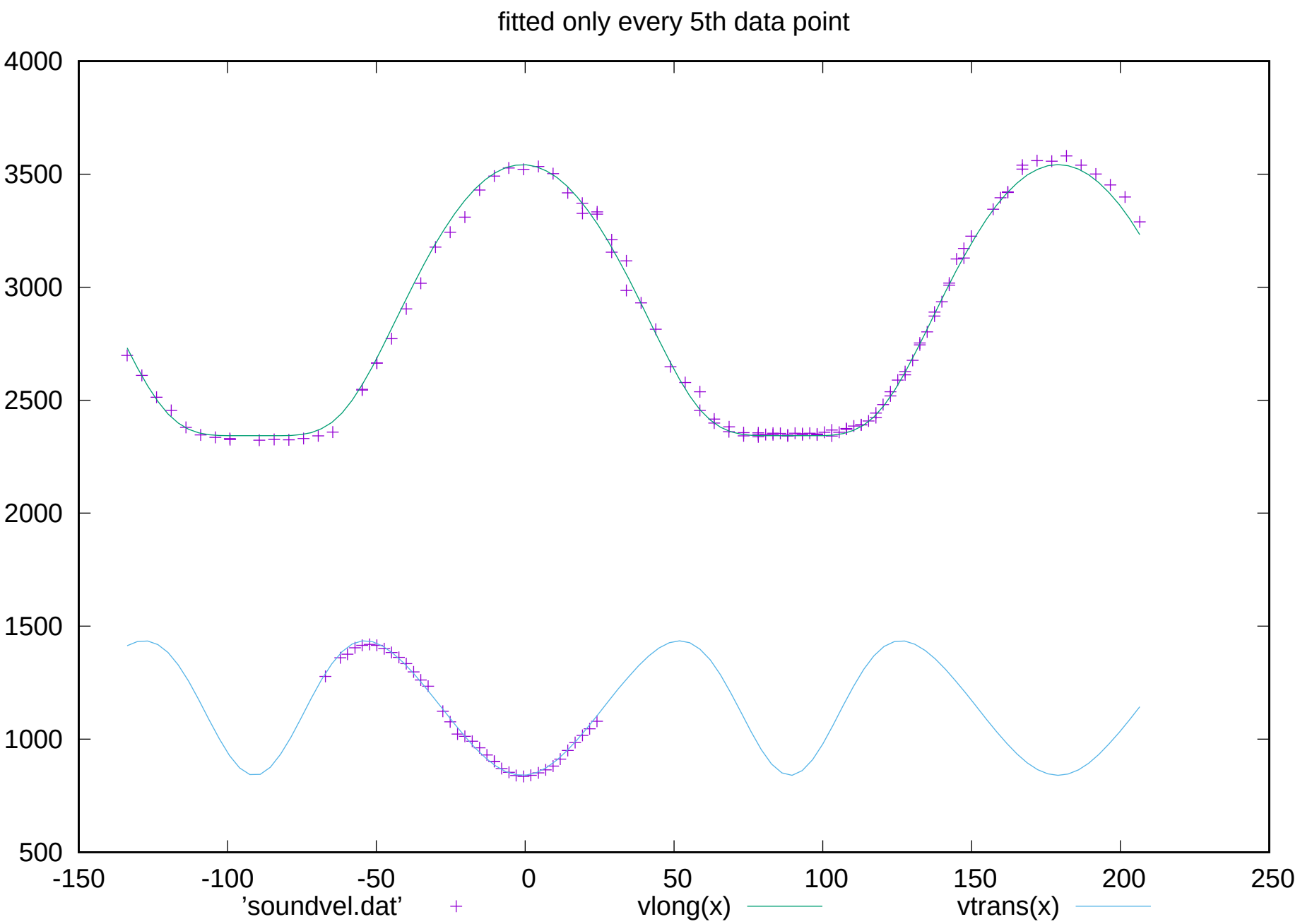


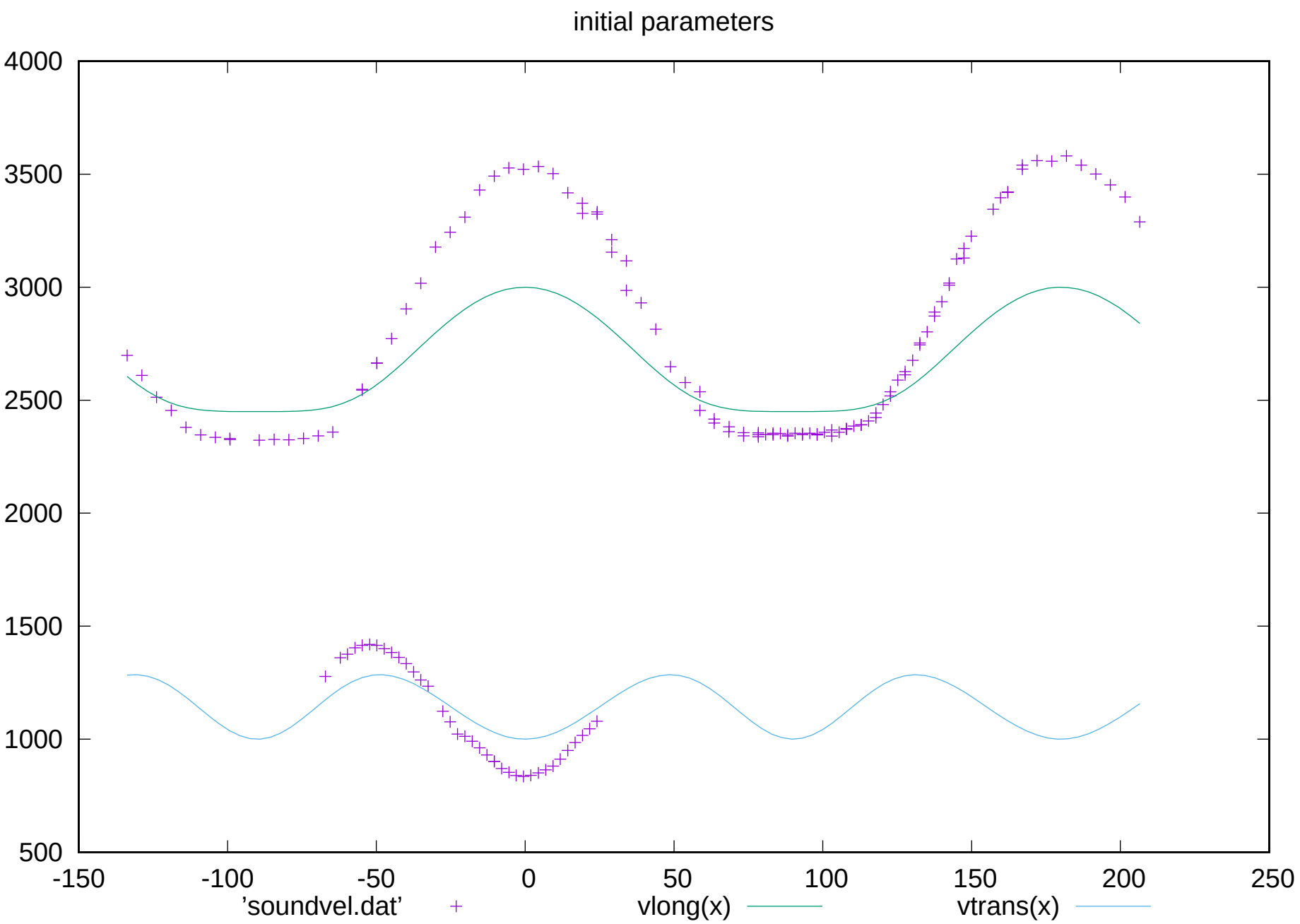

the scattered points, fitted curve
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
’hemisphr.dat’ using 1:2:3
h(x,y)

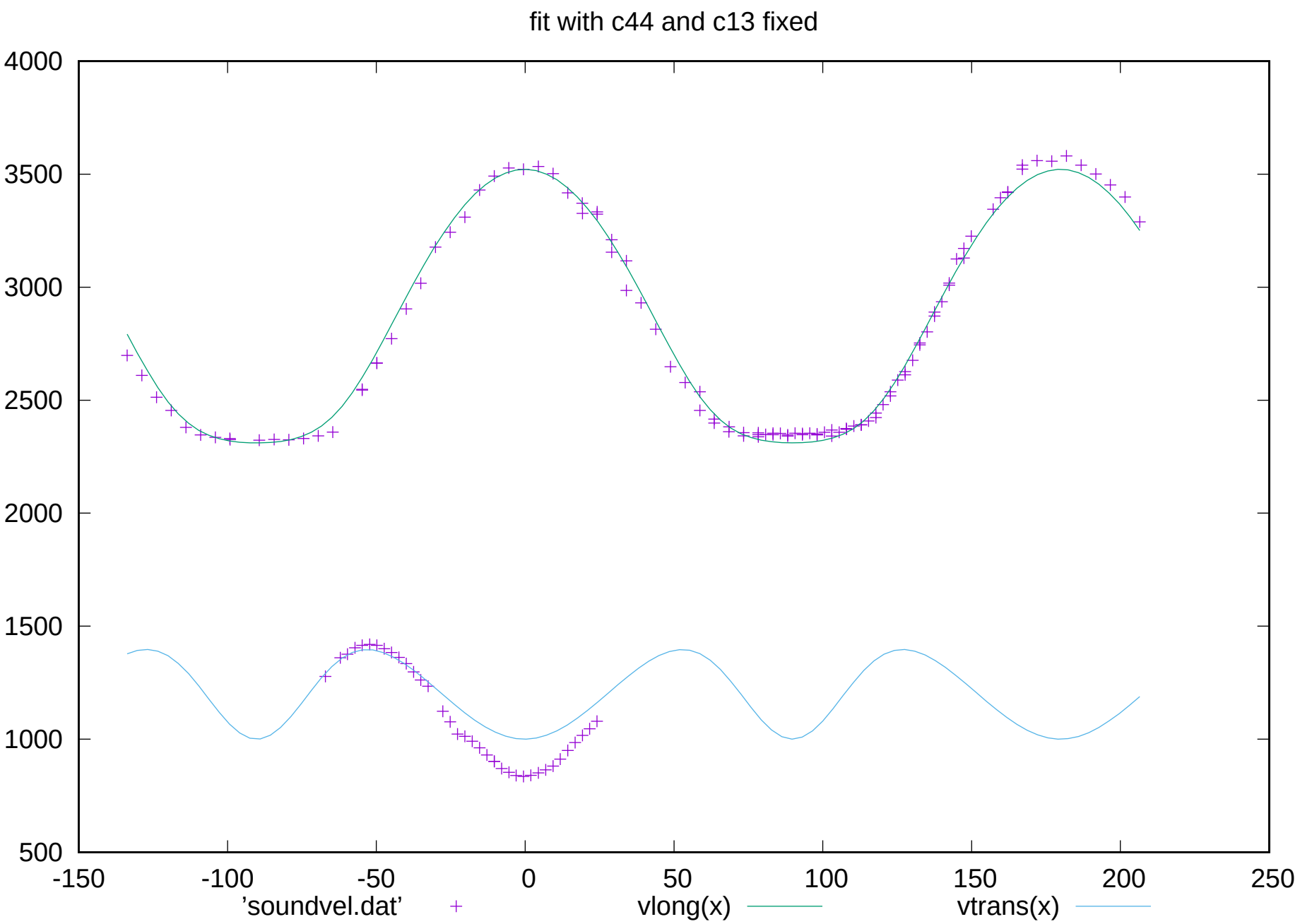

sound data, and model with initial parameters
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
'soundvel.dat'
vlong(x)
vtrans(x)

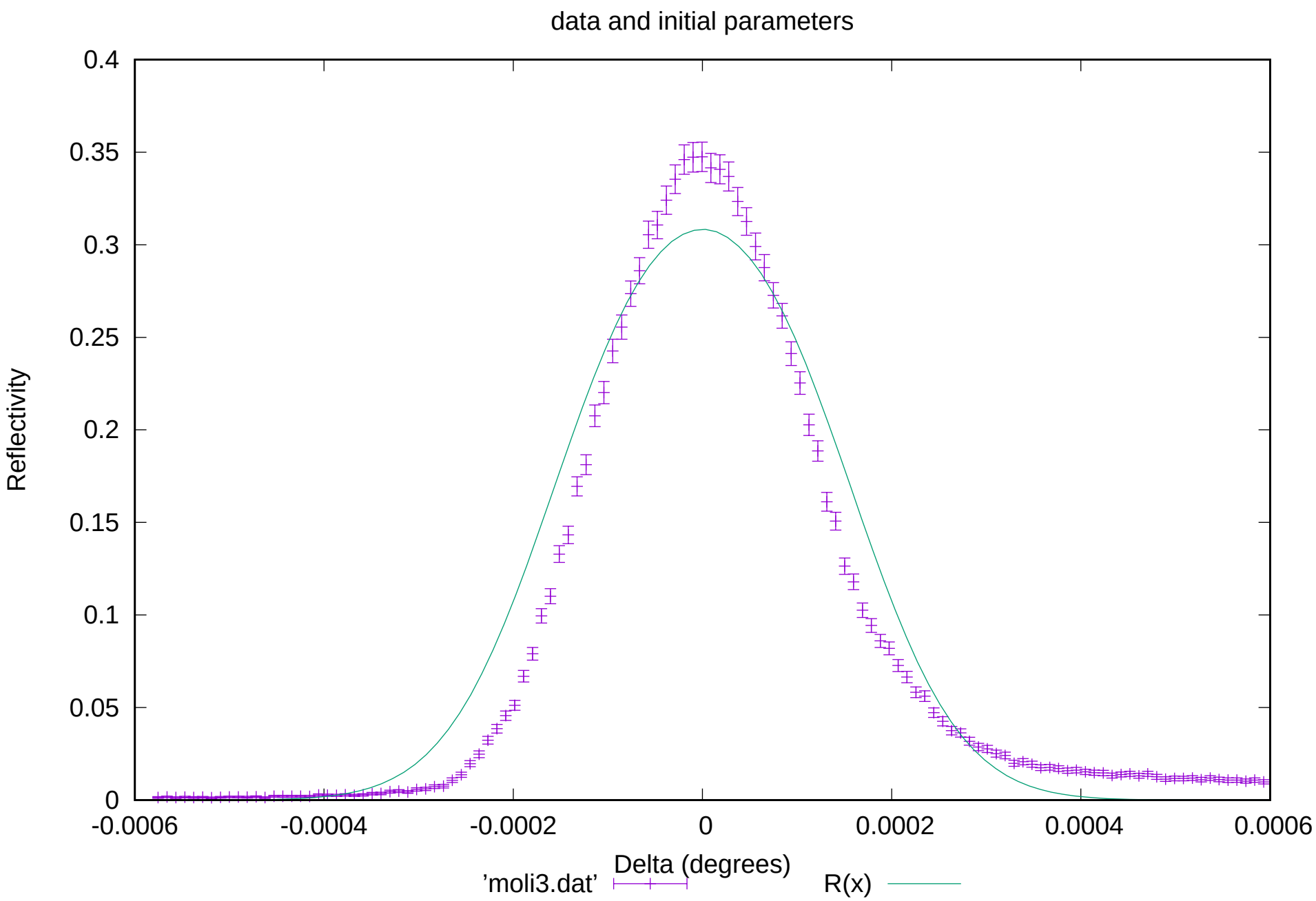

pseudo-3d multi-branch fit to velocity data
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
'soundvel.dat'
vlong(x)
vtrans(x)

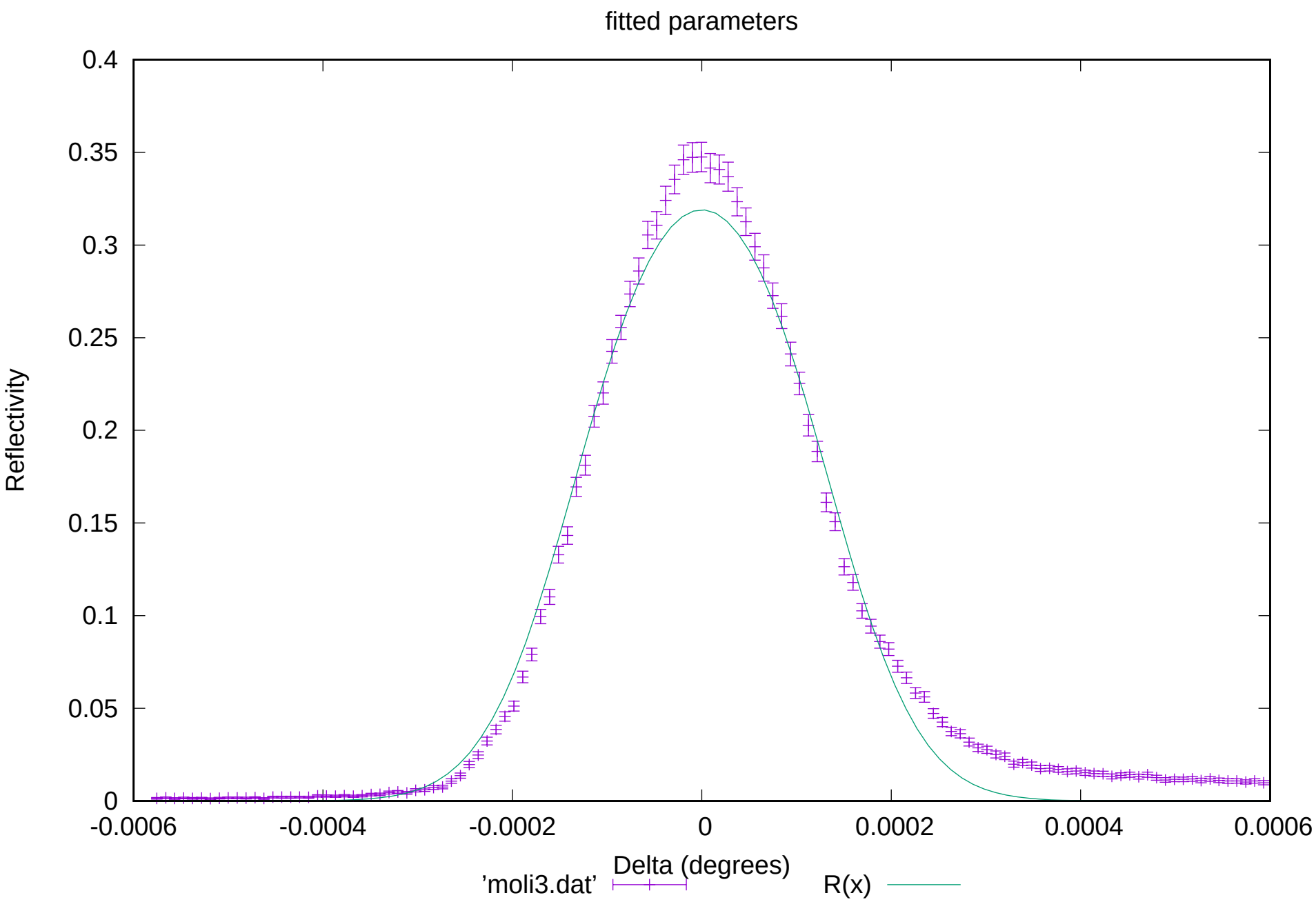

pseudo-3d multi-branch fit to velocity data
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
'soundvel.dat'
vlong(x)
vtrans(x)

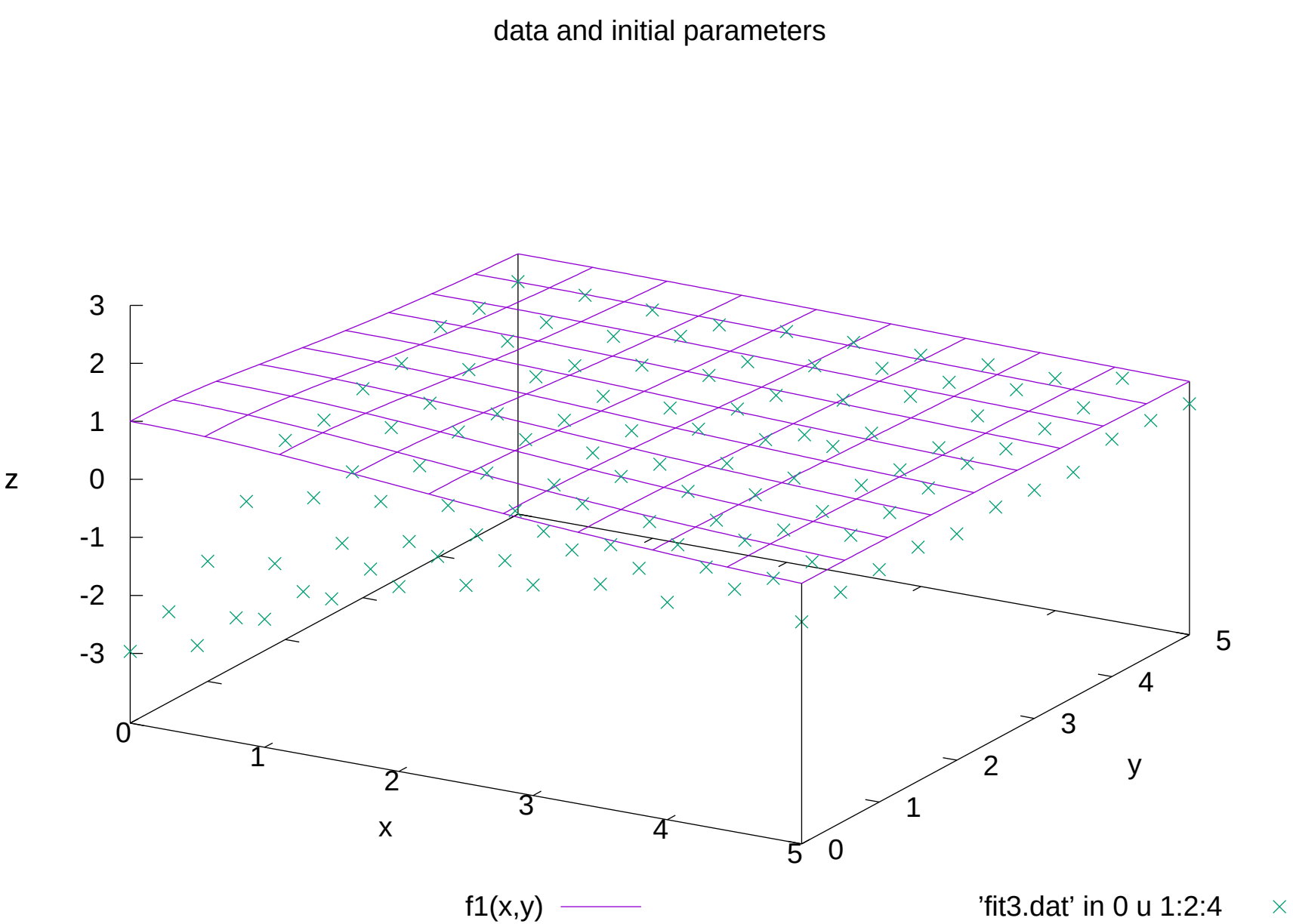

fitted only every 5th data point
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
'soundvel.dat'
vlong(x)
vtrans(x)

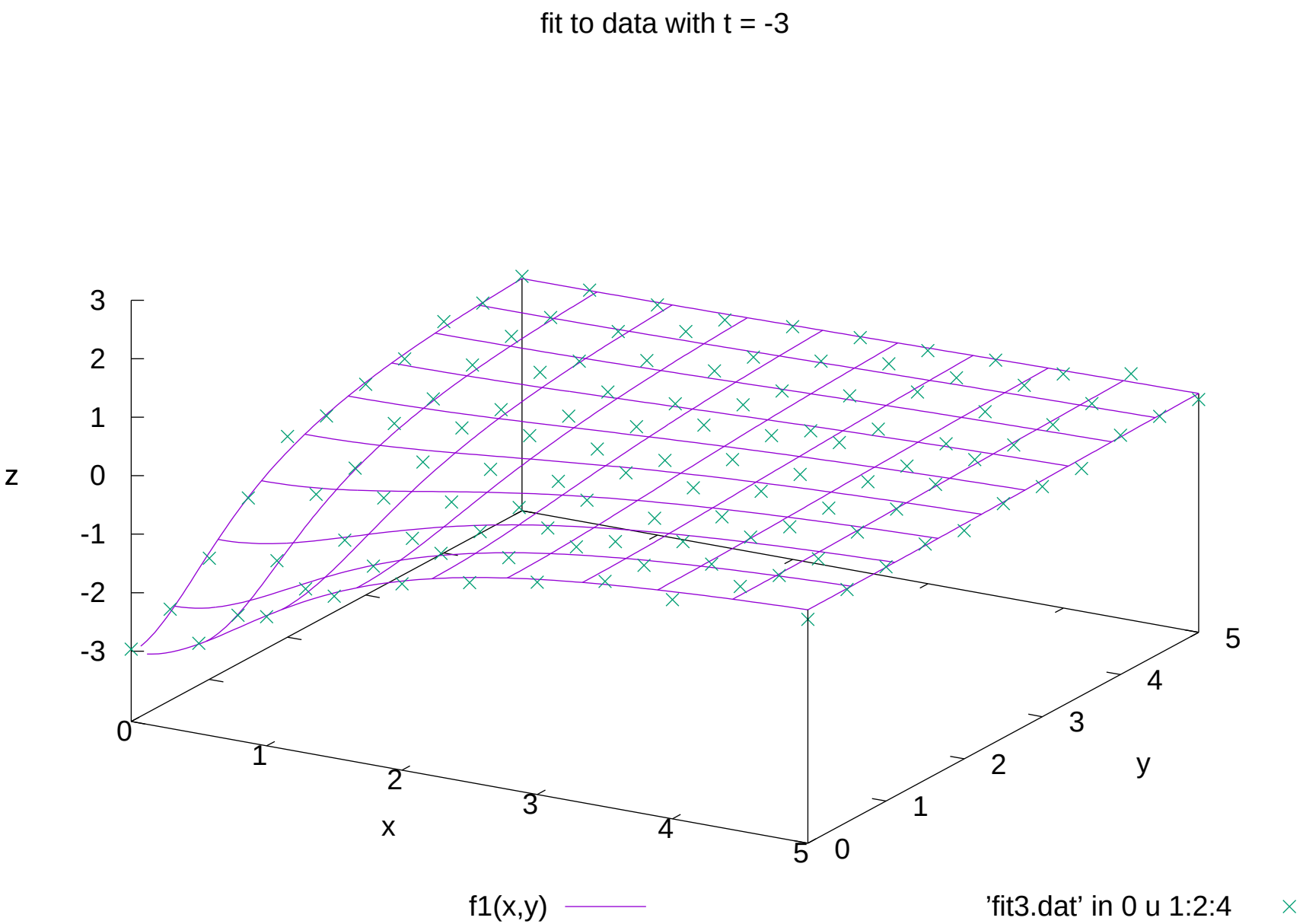

initial parameters
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
'soundvel.dat'
vlong(x)
vtrans(x)

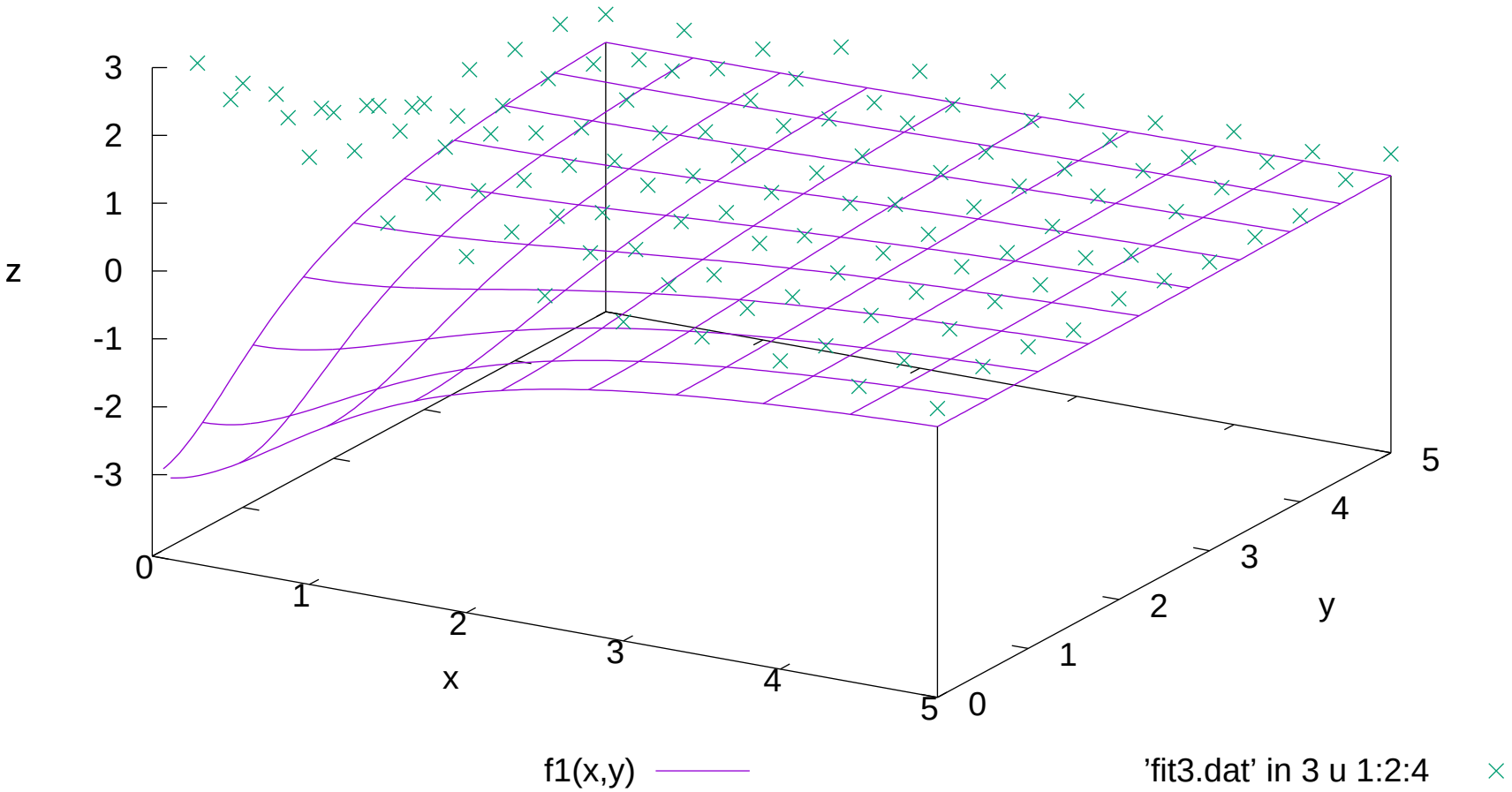

fit with c44 and c13 fixed
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
'soundvel.dat'
vlong(x)
vtrans(x)

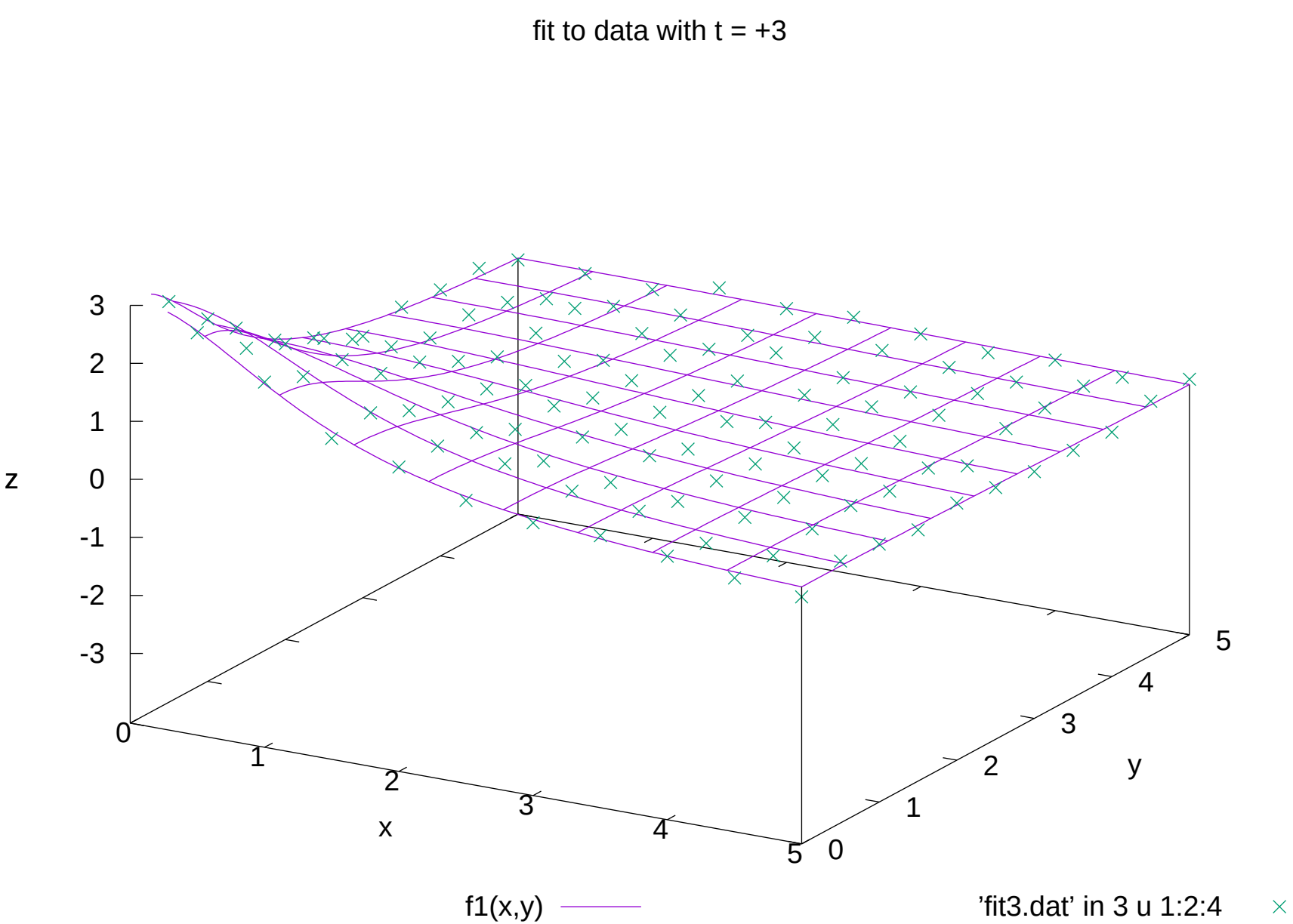

data and initial parameters
Reflectivity
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.0006
-0.0004
-0.0002
0
0.0002
0.0004
0.0006
Delta (degrees)
'moli3.dat'
R(x)

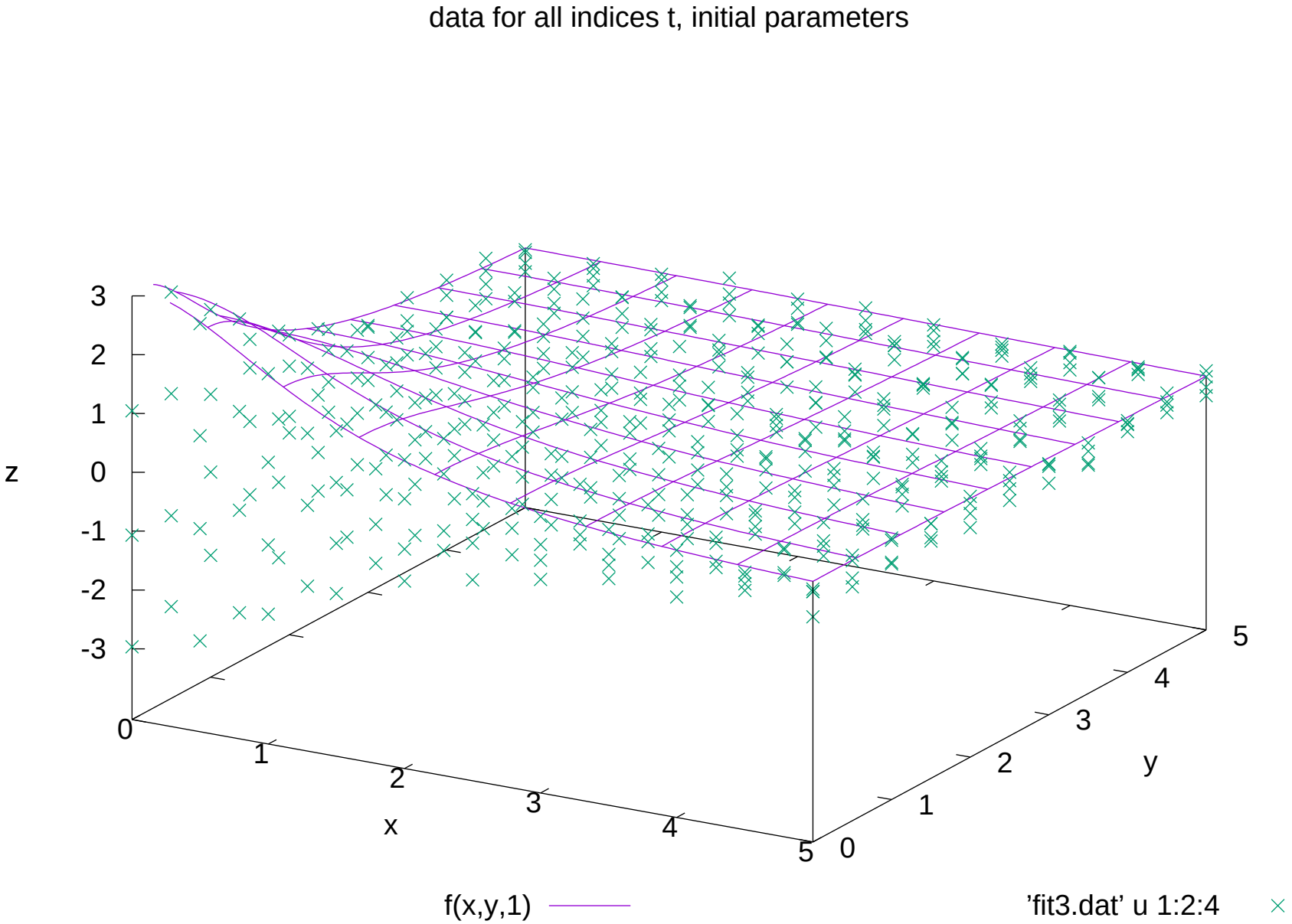

fitted parameters
Reflectivity
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.0006
-0.0004
-0.0002
0
0.0002
0.0004
0.0006
Delta (degrees)
'moli3.dat'
R(x)

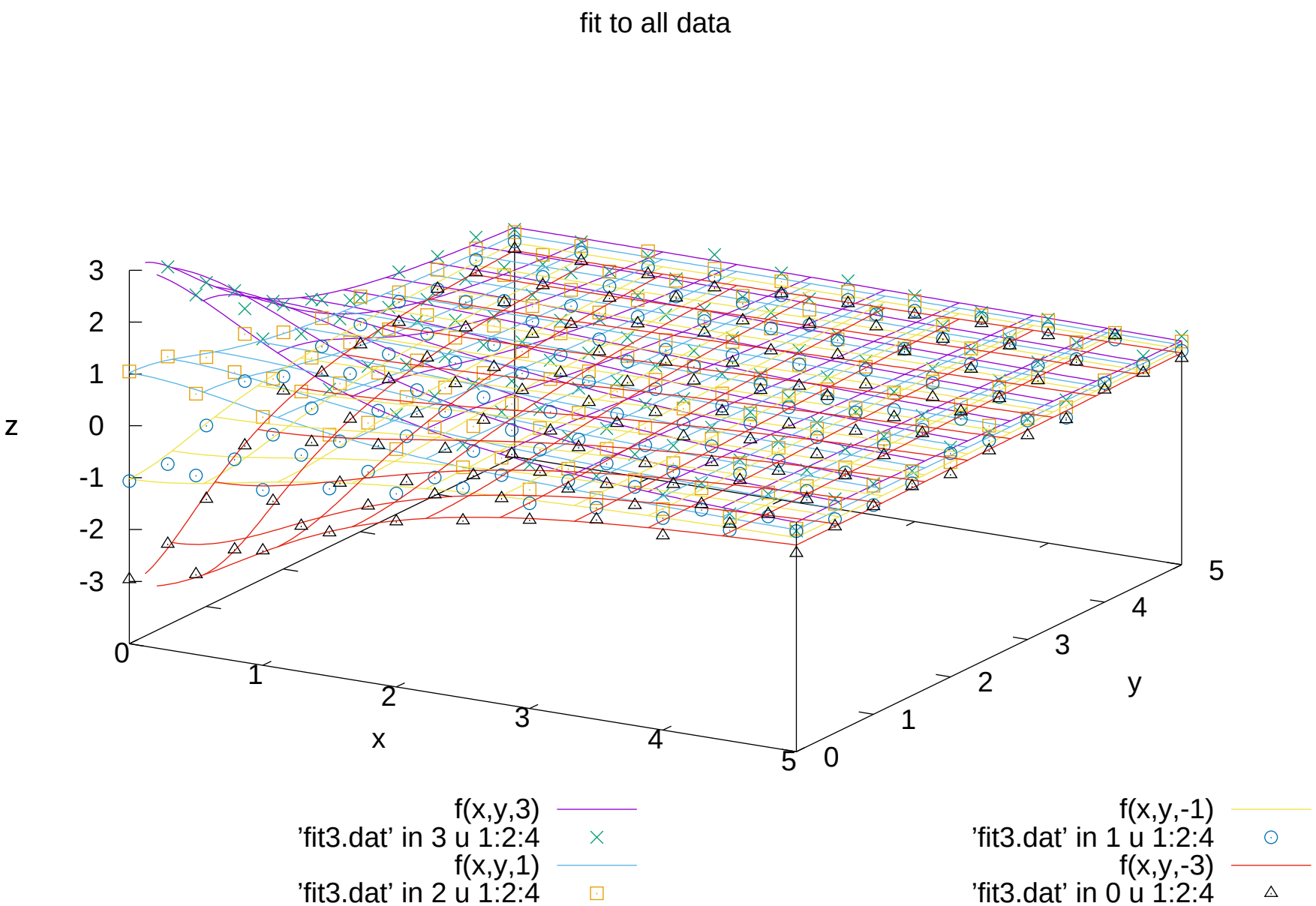

data and initial parameters
z
3
2
1
0
-1
-2
-3
0
1
2
3
4
5
x
0
1
2
3
4
5
y
f1(x,y)
'fit3.dat' in 0 u 1:2:4

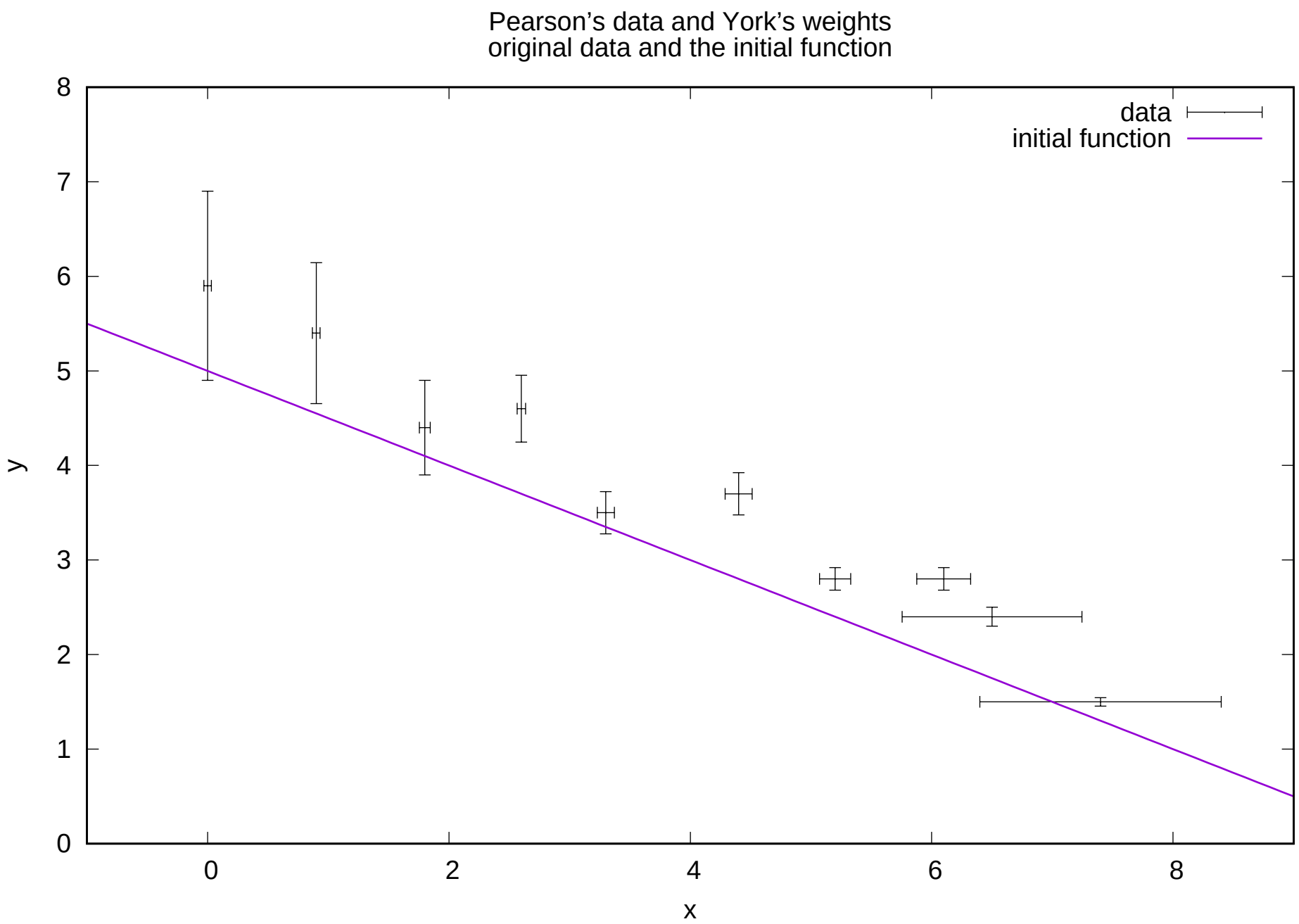

fit to data with t = -3
z
3
2
1
0
-1
-2
-3
0
1
2
3
4
5
x
0
1
2
3
4
5
y
f1(x,y)
'fit3.dat' in 0 u 1:2:4

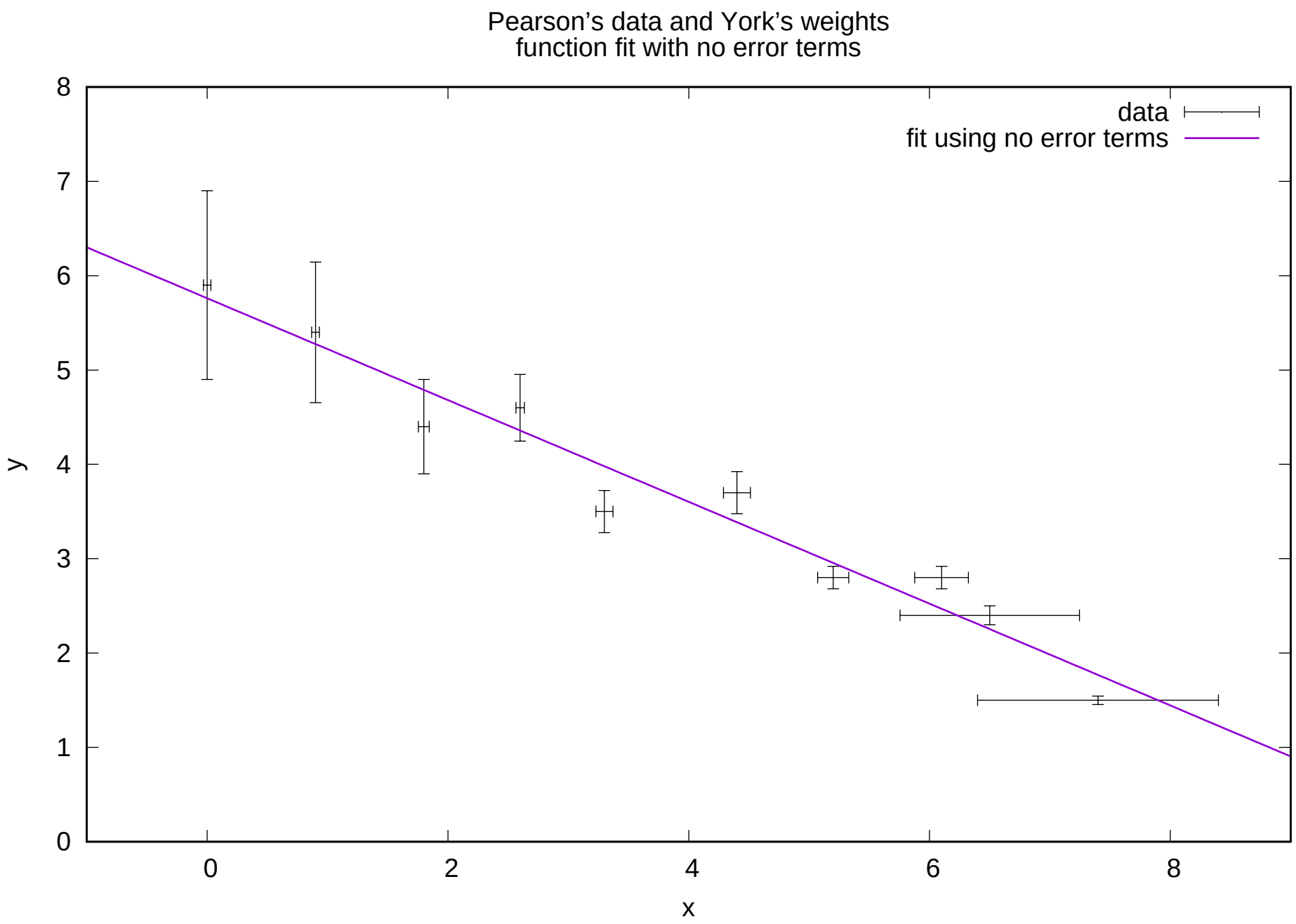

fit to data with t = +3, initial parameters
z
3
2
1
0
-1
-2
-3
0
1
2
3
4
5
x
0
1
2
3
4
5
y
f1(x,y)
'fit3.dat' in 3 u 1:2:4

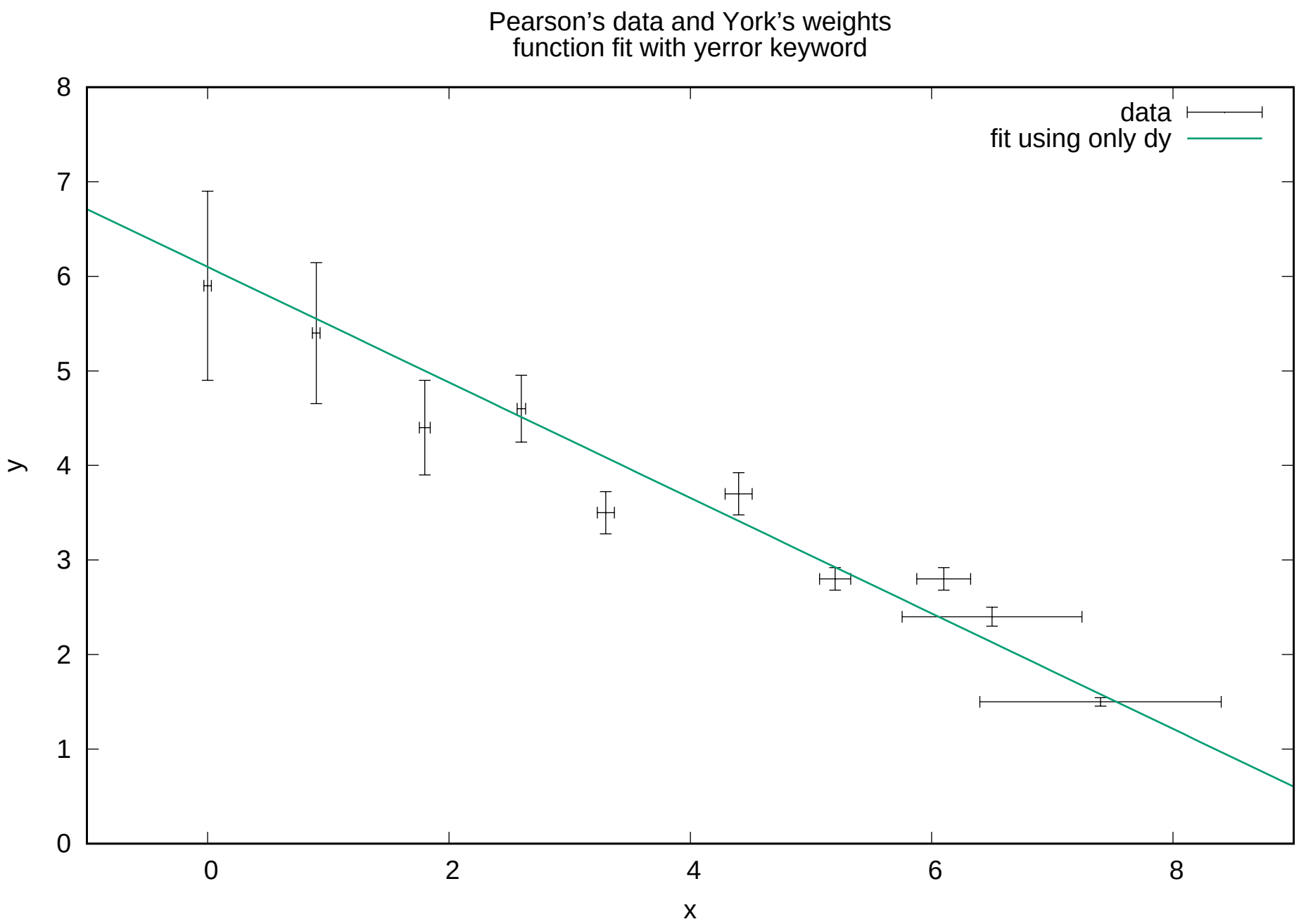

fit to data with t = +3
z
3
2
1
0
-1
-2
-3
0
1
2
3
4
5
x
0
1
2
3
4
5
y
f1(x,y)
'fit3.dat' in 3 u 1:2:4

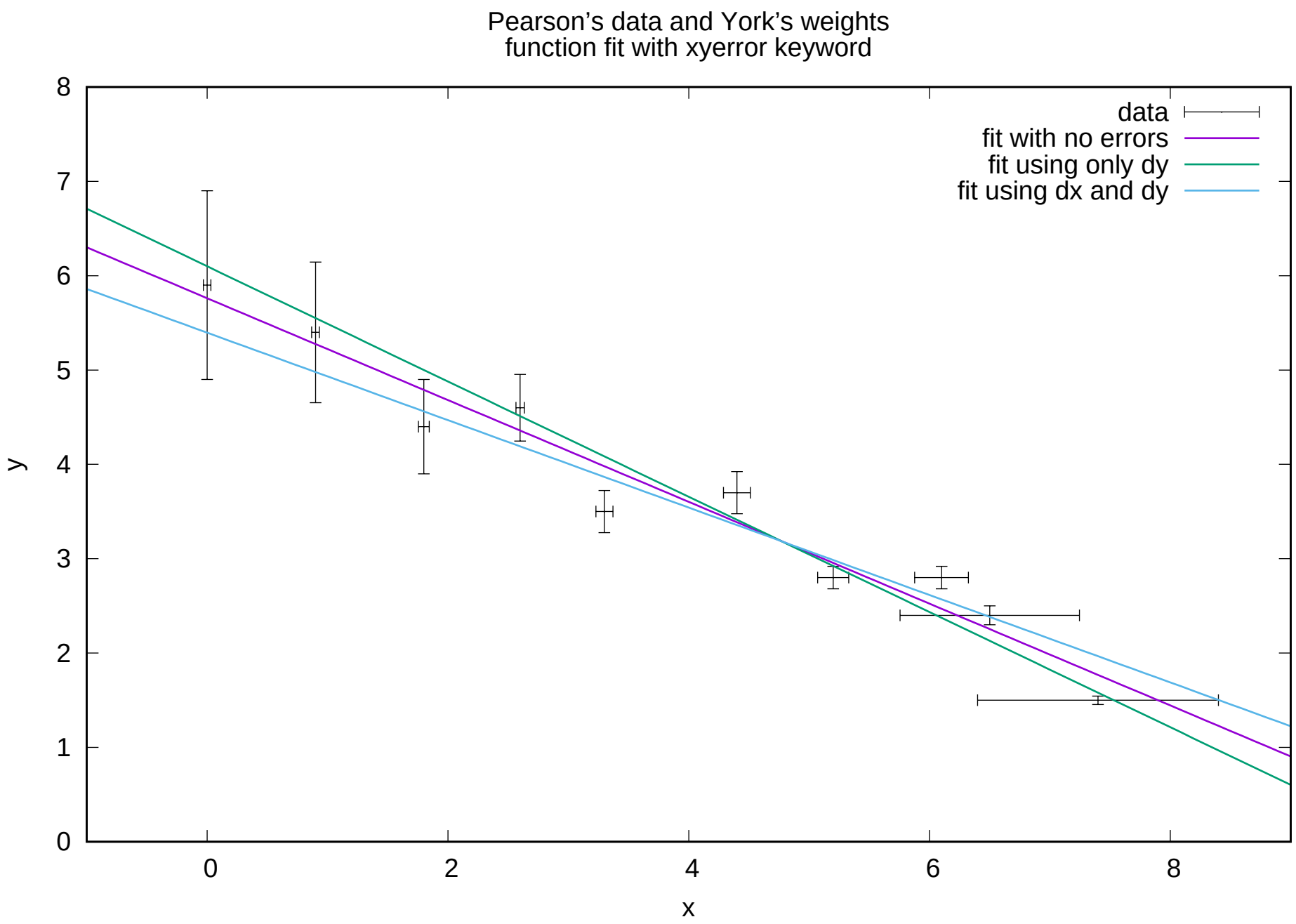

data for all indices t, initial parameters
z
3
2
1
0
-1
-2
-3
0
1
2
3
4
5
x
0
1
2
3
4
5
y
f(x,y,1)
'fit3.dat' u 1:2:4

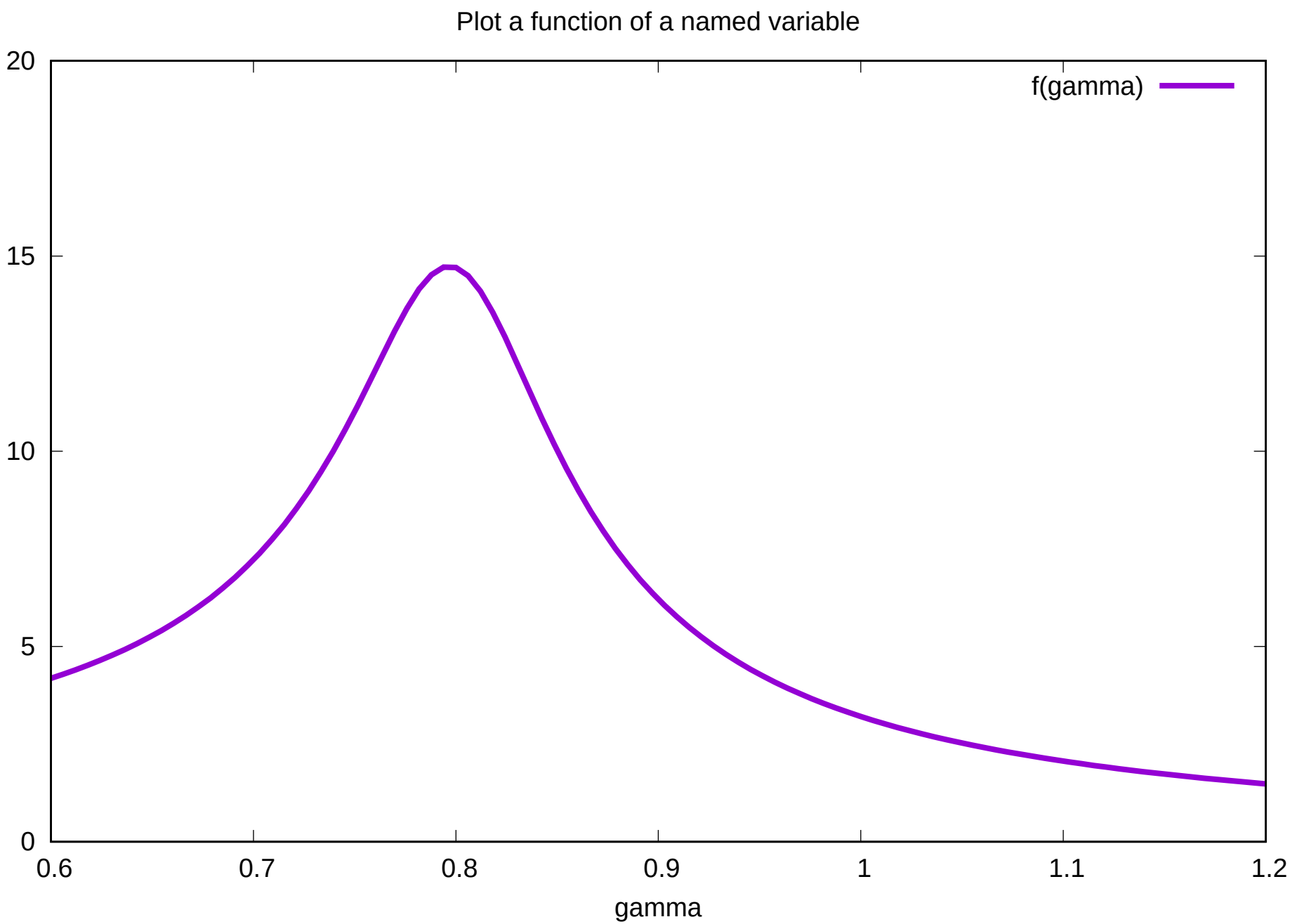

fit to all data
z
3
2
1
0
-1
-2
-3
0
1
2
3
4
5
x
y
f(x,y,3)
'fit3.dat' in 3 u 1:2:4
f(x,y,1)
'fit3.dat' in 2 u 1:2:4
f(x,y,-1)
'fit3.dat' in 1 u 1:2:4
f(x,y,-3)
'fit3.dat' in 0 u 1:2:4

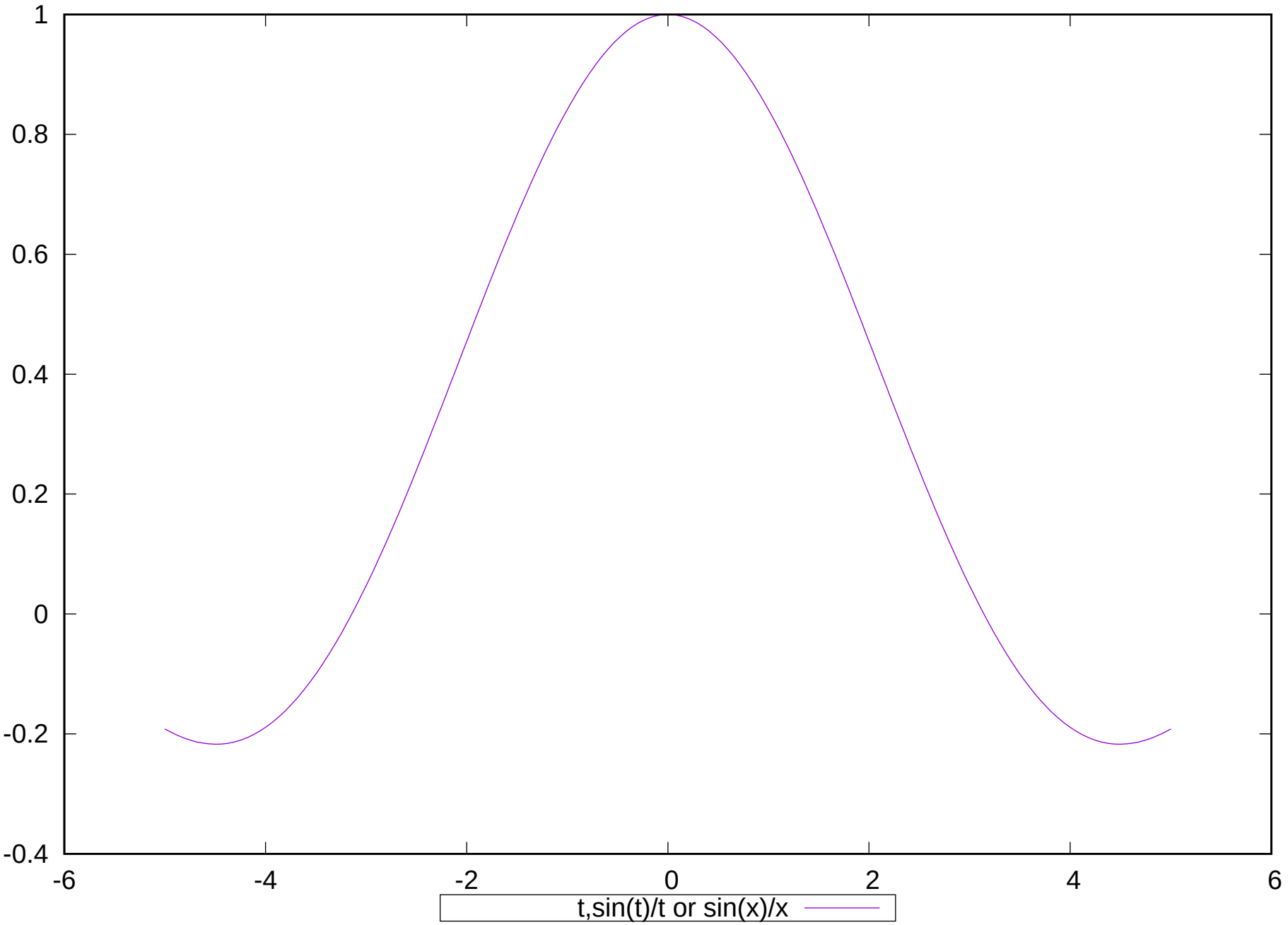

Pearson’s data and York’s weights
original data and the initial function
data
initial function
y
x
0
1
2
3
4
5
6
7
8
0
2
4
6
8

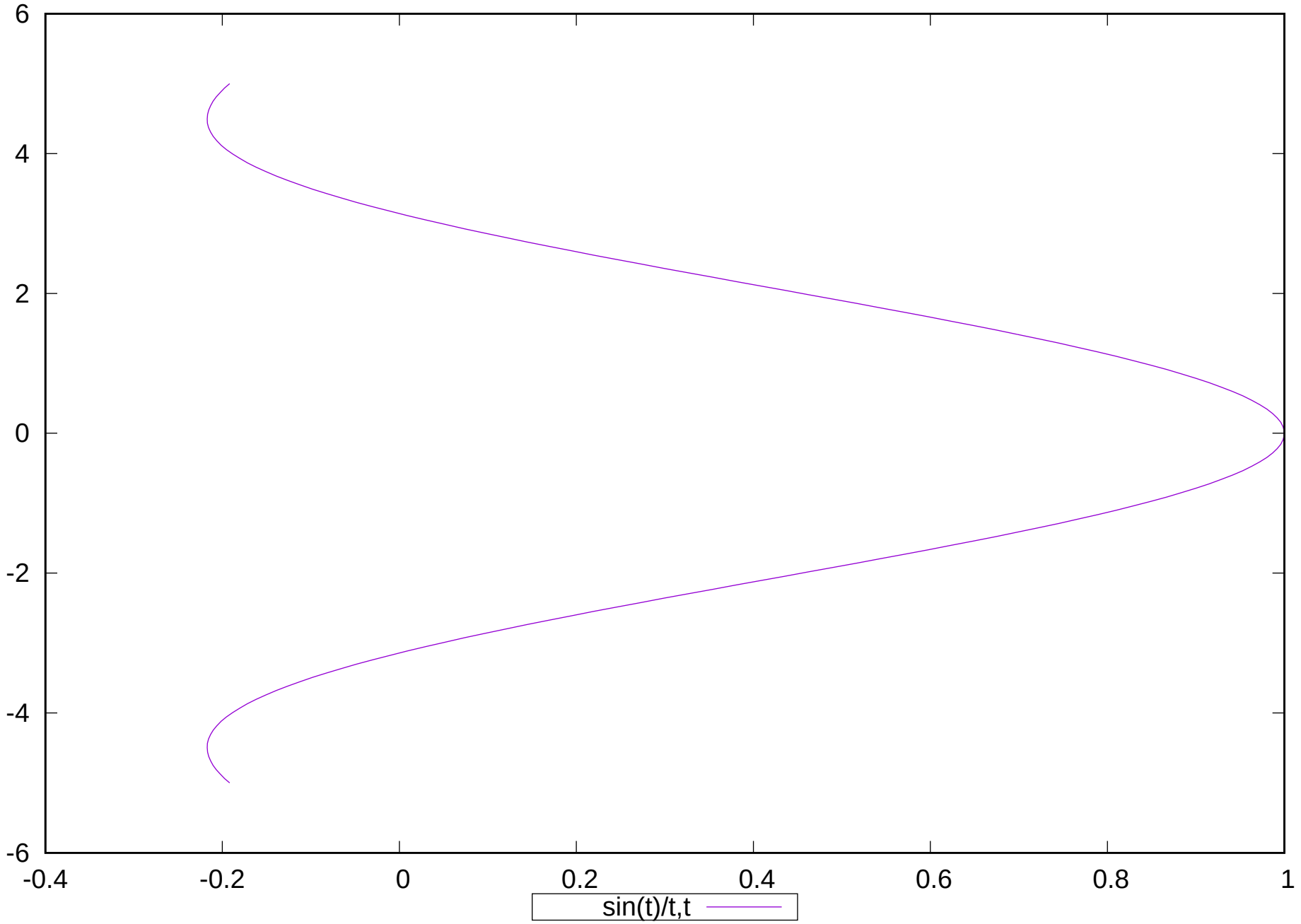

Pearson’s data and York’s weights
function fit with no error terms
data
fit using no error terms
y
x
0
1
2
3
4
5
6
7
8
0
2
4
6
8

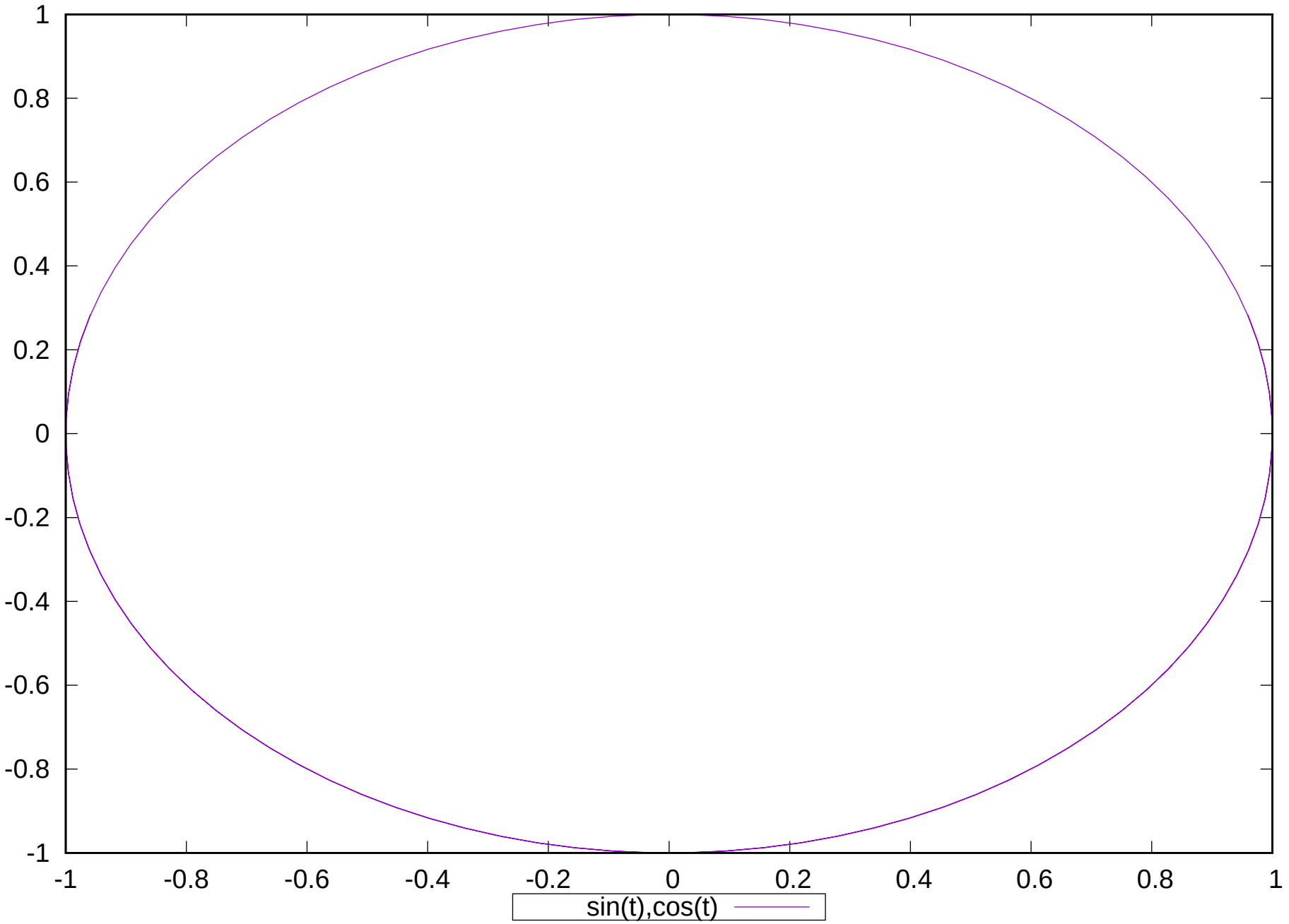

Pearson’s data and York’s weights
function fit with yerror keyword
data
fit using only dy
y
x
0
1
2
3
4
5
6
7
8
0
2
4
6
8

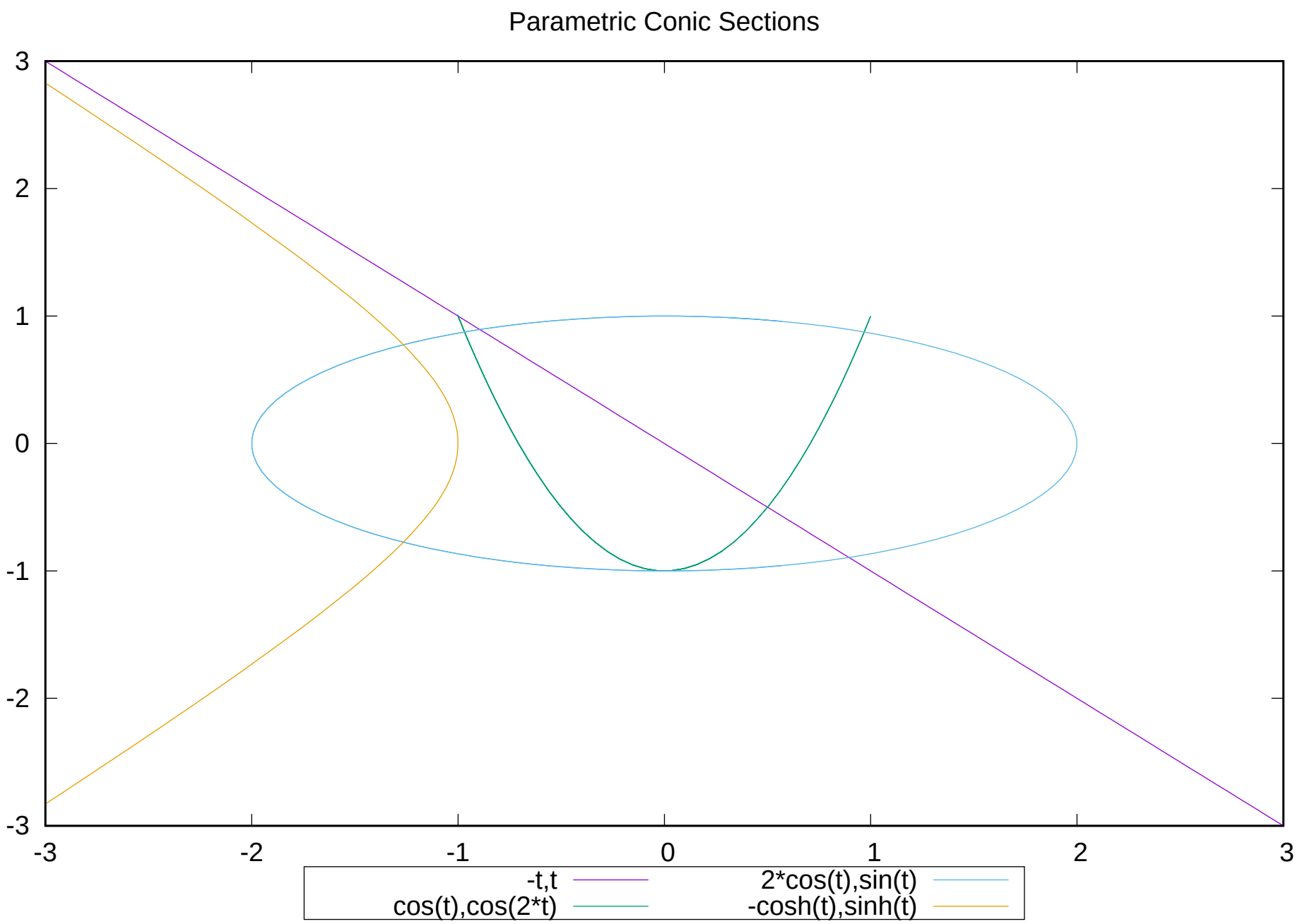

Pearson’s data and York’s weights
function fit with xyerror keyword
data
fit with no errors
fit using only dy
fit using dx and dy
y
x
0
1
2
3
4
5
6
7
8
0
2
4
6
8

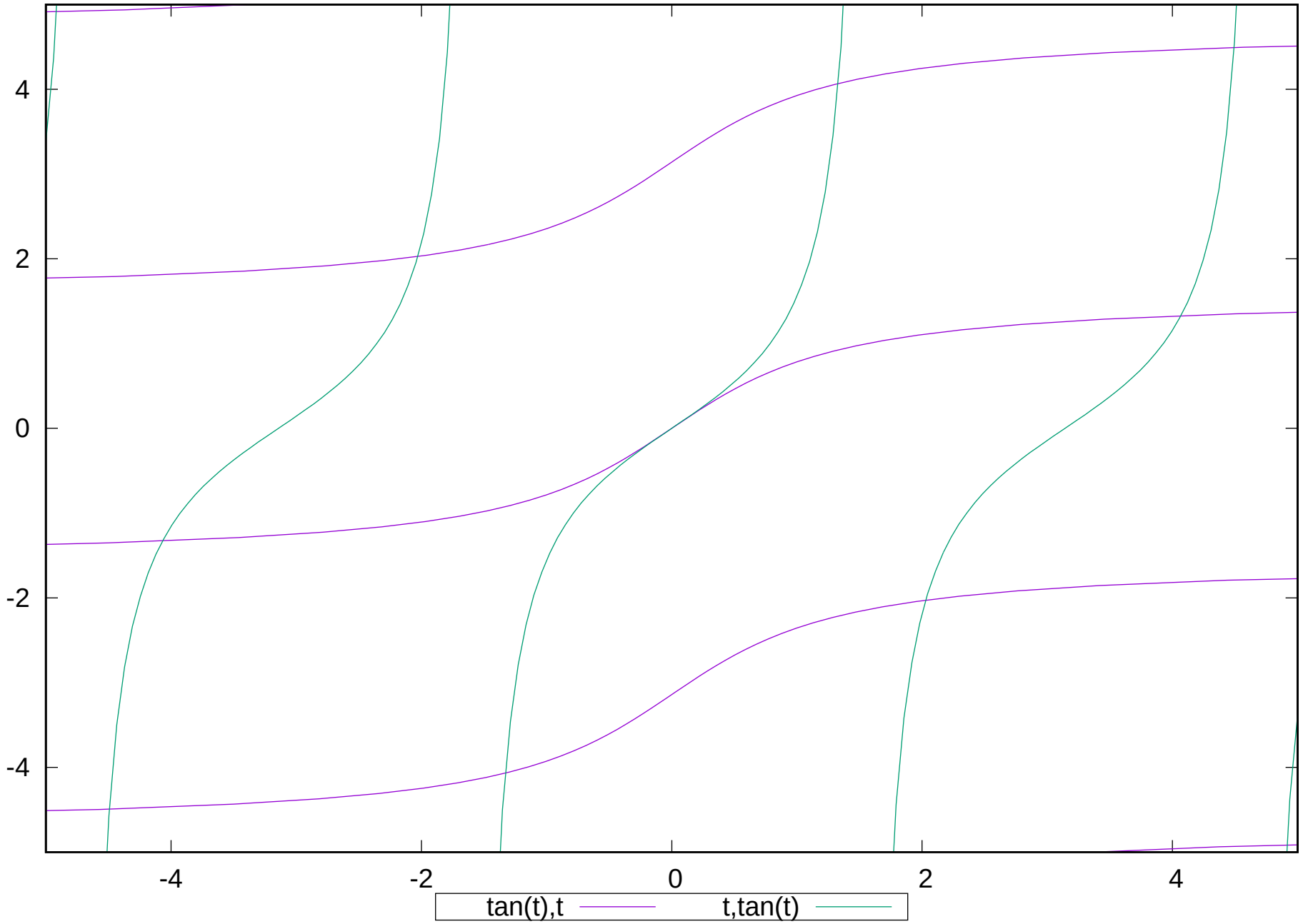

Plot a function of a named variable
f(gamma)
20
15
10
5
0
0.6
0.7
0.8
0.9
1
1.1
1.2
gamma

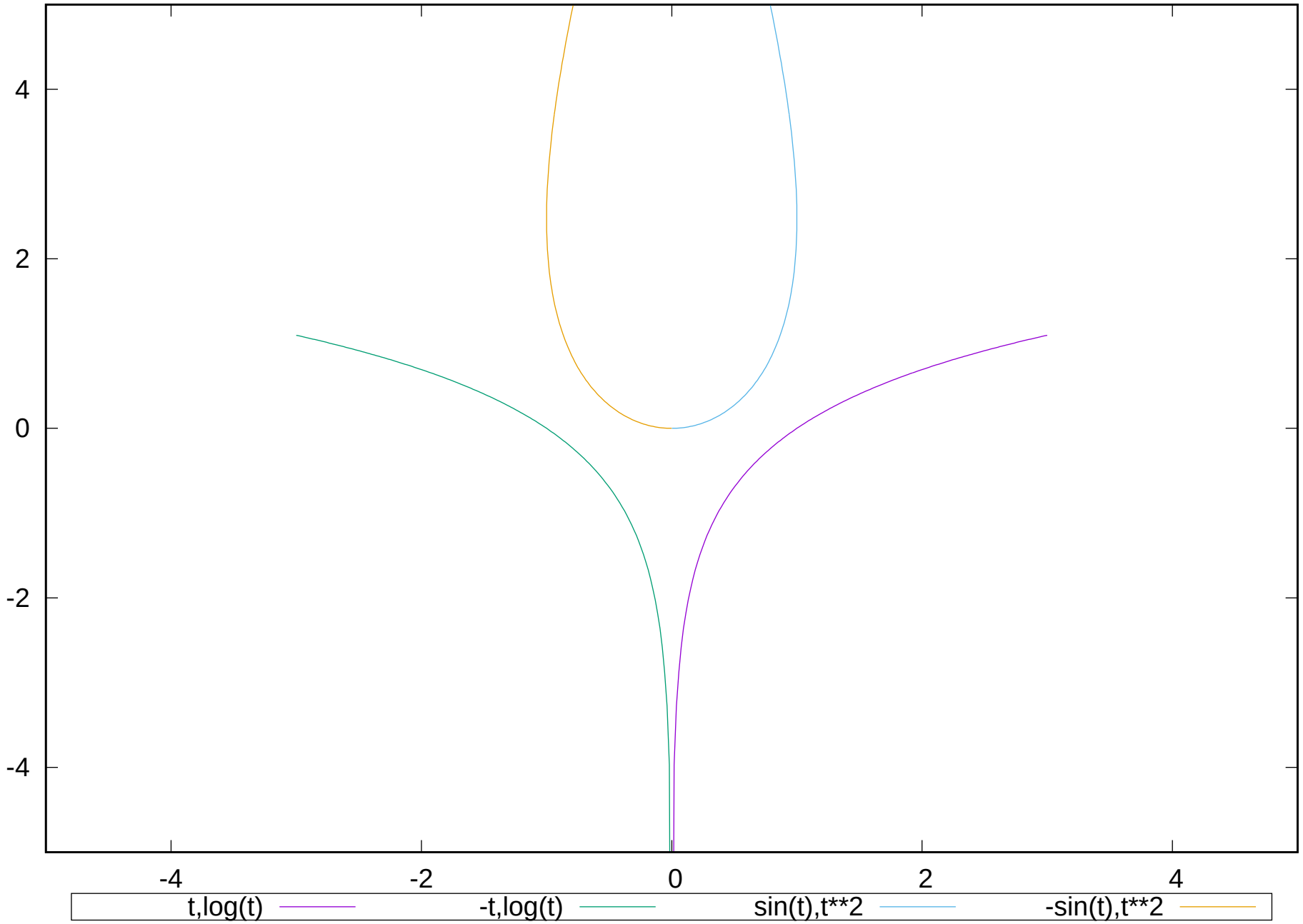

1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-6
-4
-2
0
2
4
6
t,sin(t)/t or sin(x)/x

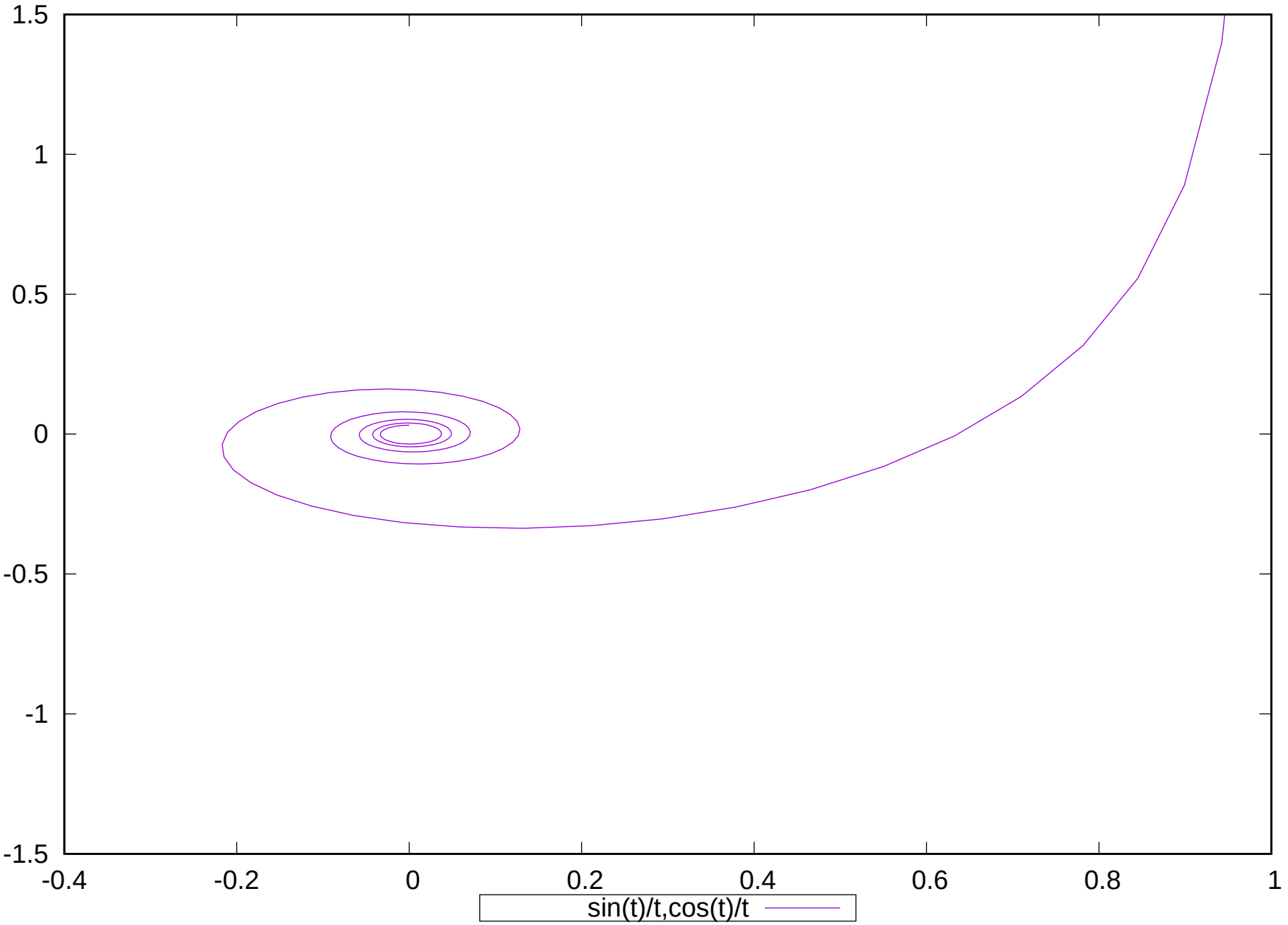

6
4
2
0
-2
-4
-6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
sin(t)/t,t


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
sin(t),cos(t)

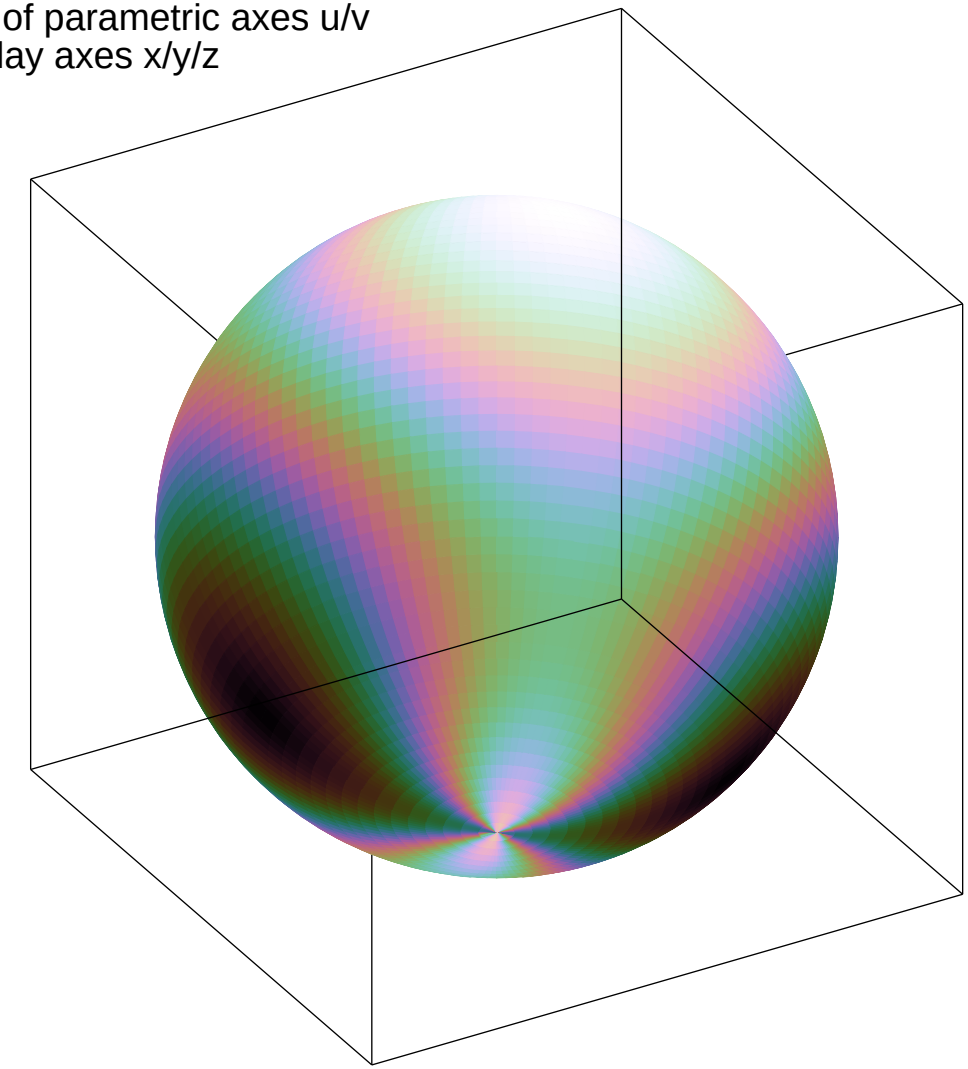

Parametric Conic Sections
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
-t,t
cos(t),cos(2*t)
2*cos(t),sin(t)
-cosh(t),sinh(t)

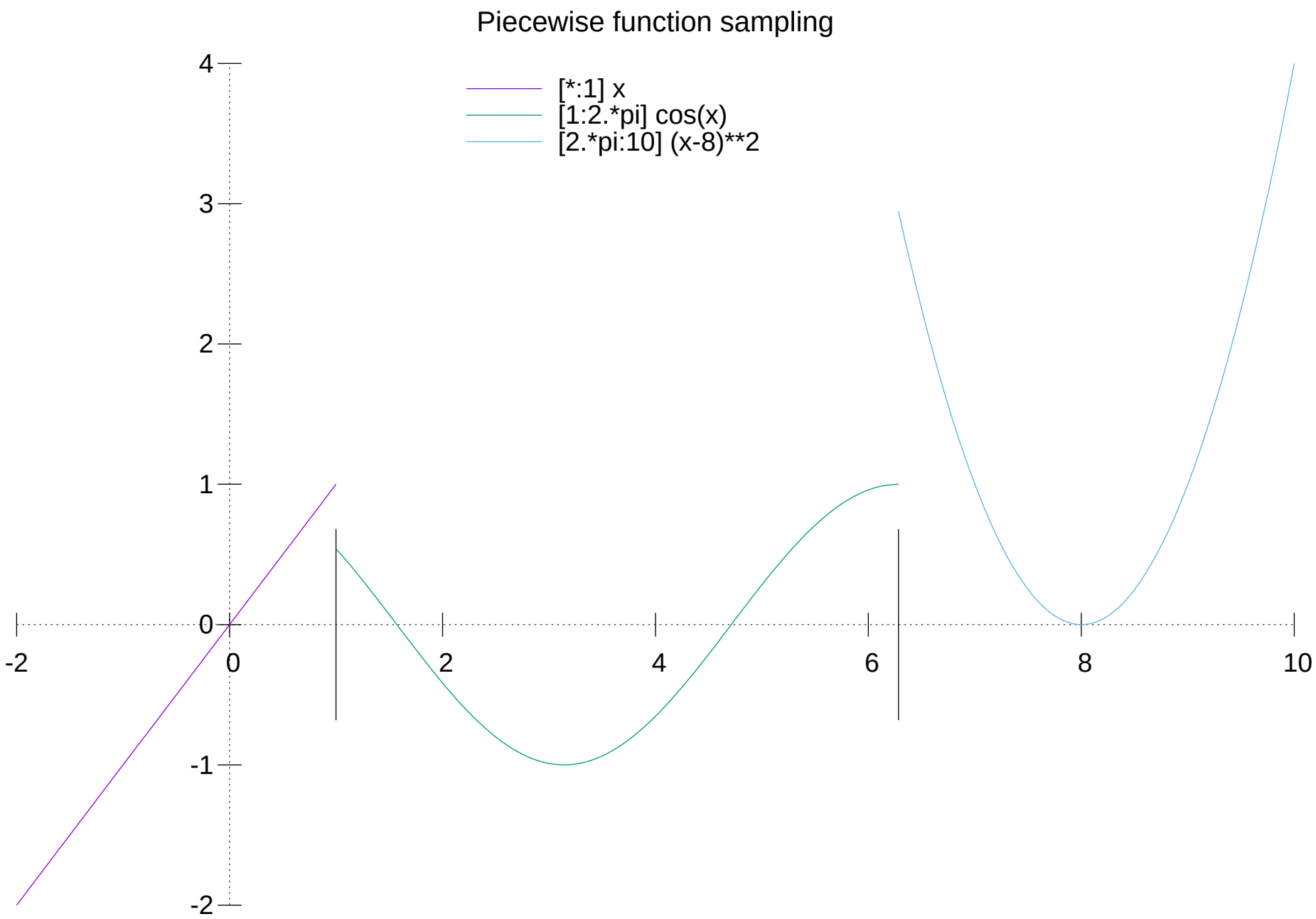

4
2
0
-2
-4
-4
-2
0
2
4
tan(t),t
t,tan(t)

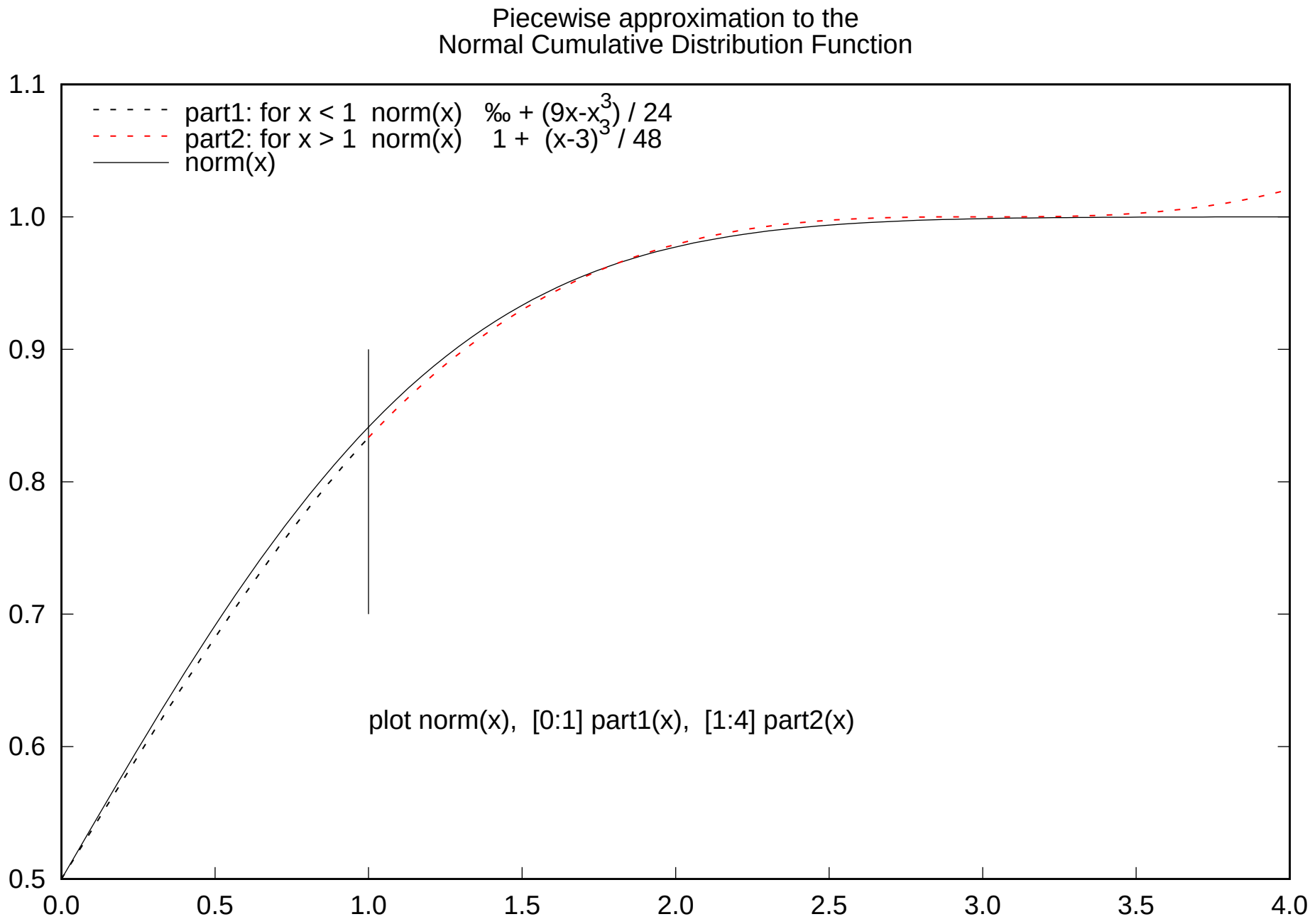

4
2
0
-2
-4
-4
-2
0
2
4
t,log(t)
-t,log(t)
sin(t),t**2
-sin(t),t**2


1.5
1
0.5
0
-0.5
-1
-1.5
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
sin(t)/t,cos(t)/t

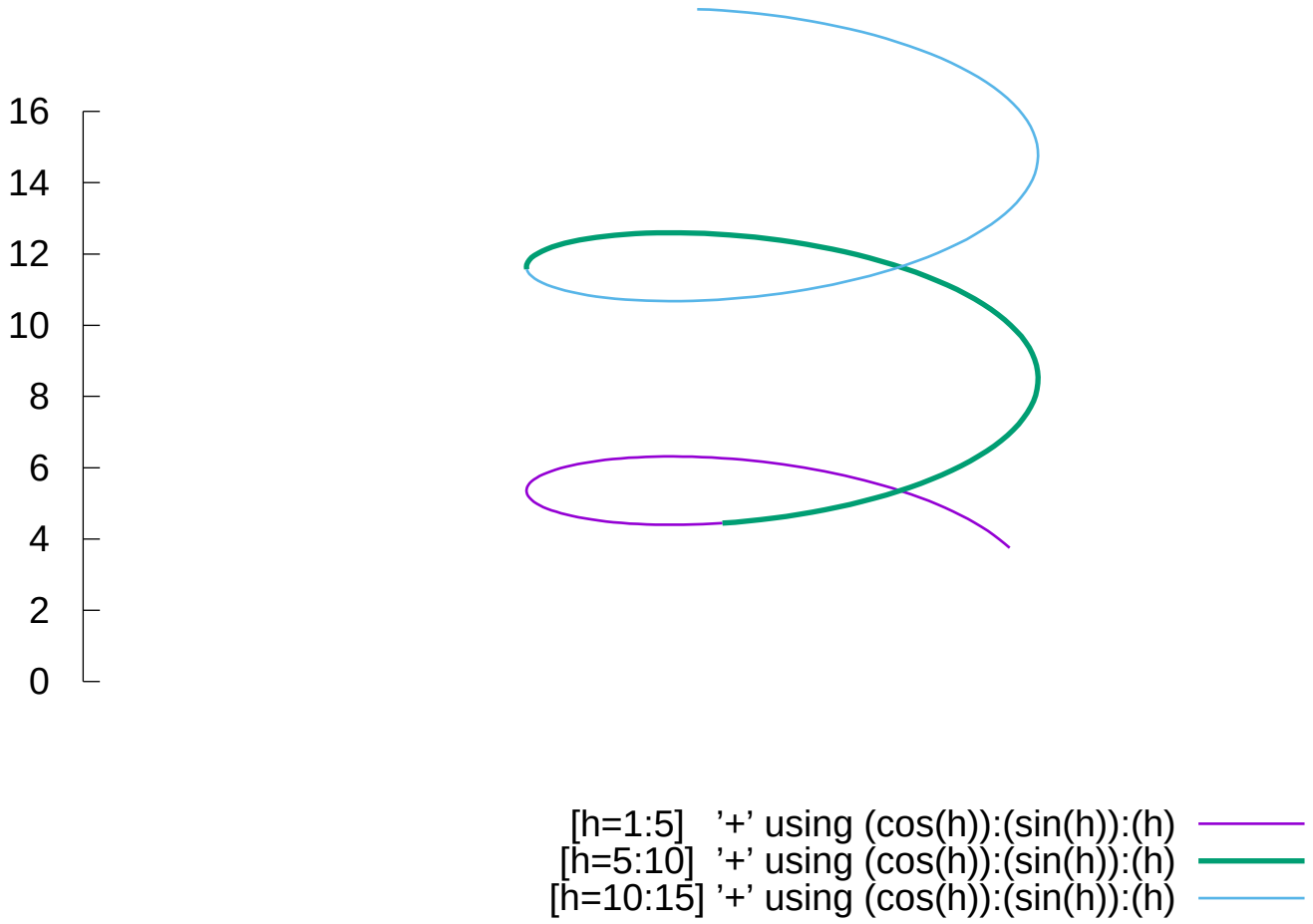
Decouple range of parametric axes u/v
from that of display axes x/y/z

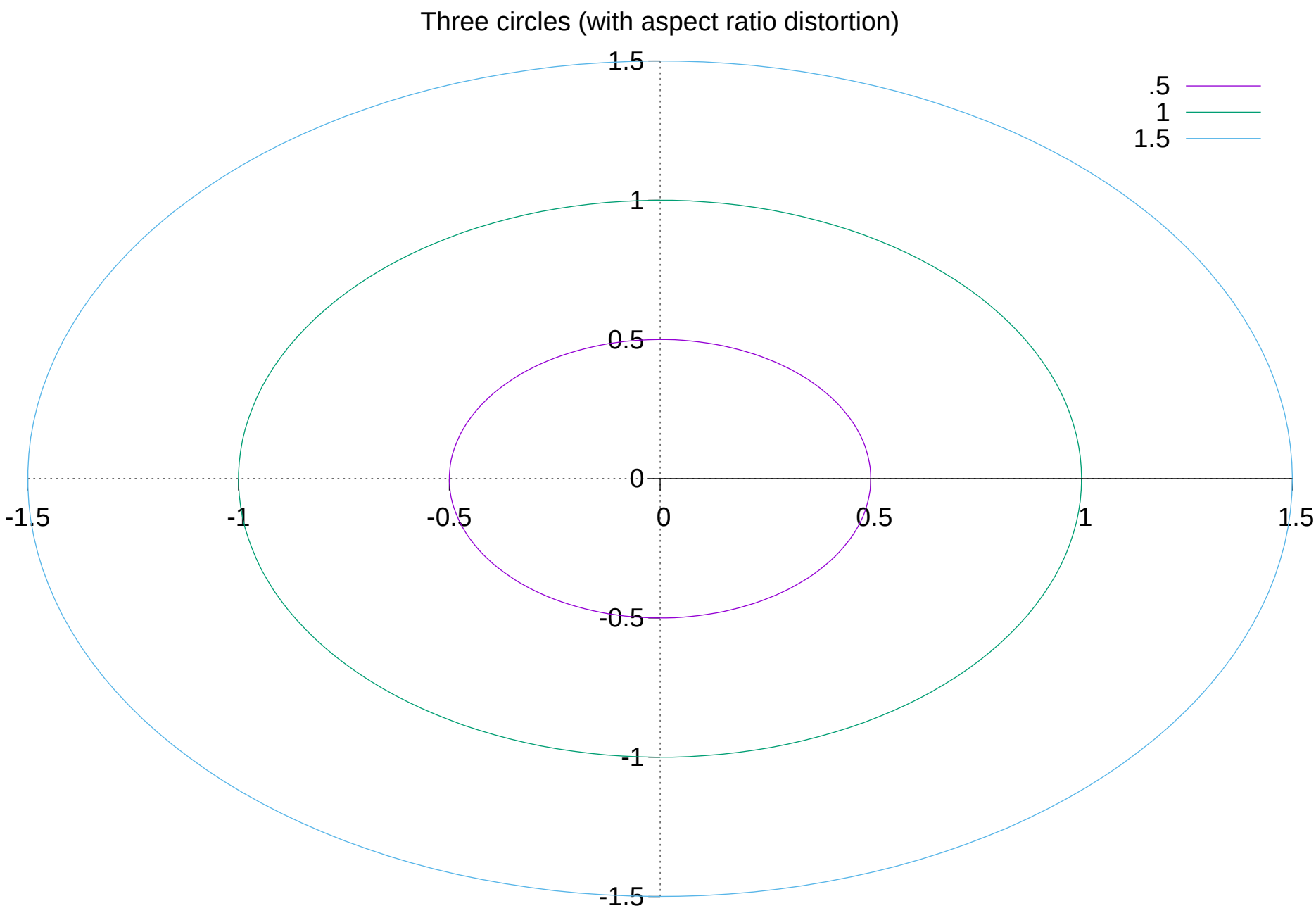


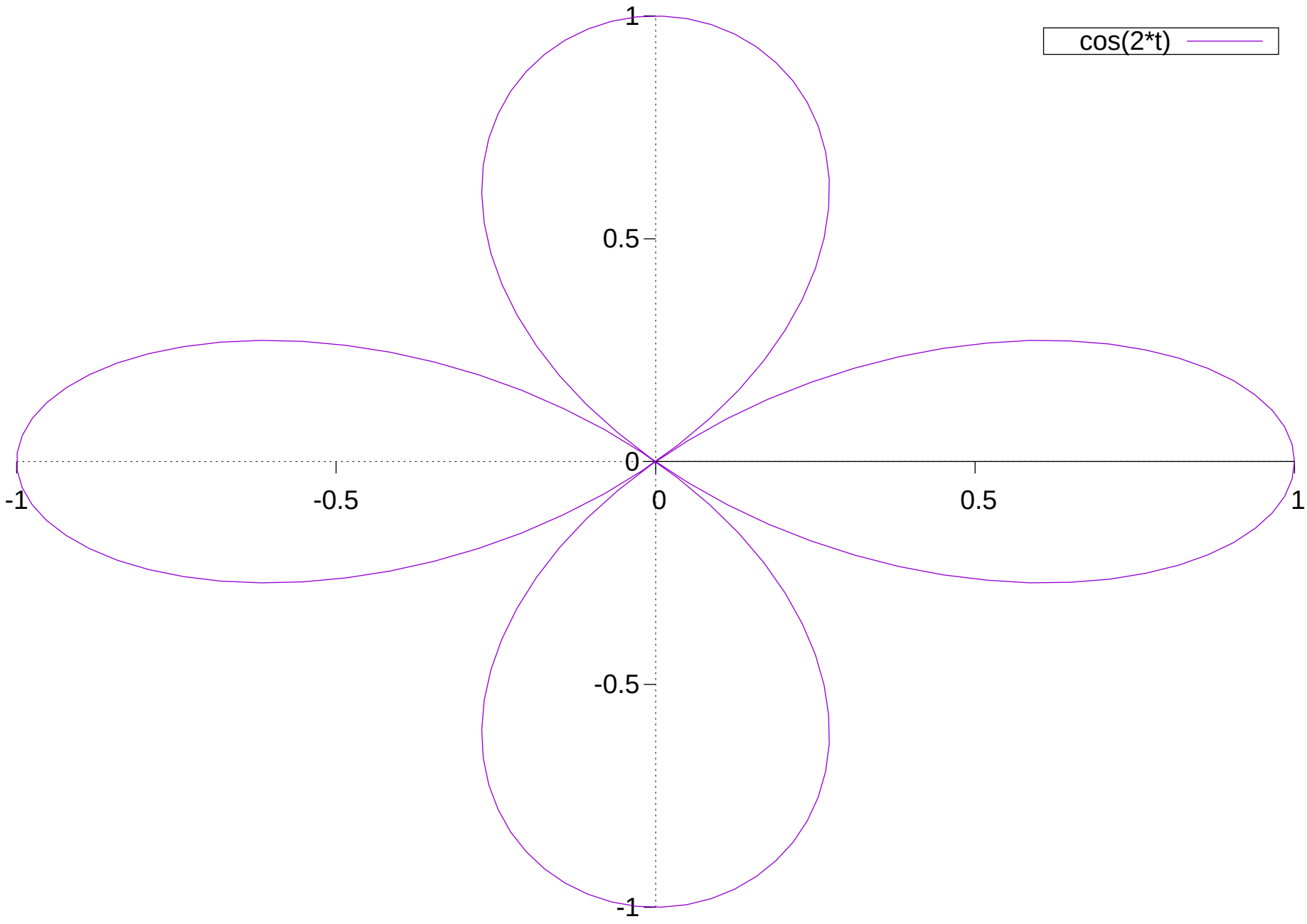

Piecewise function sampling
[*:1] x
[1:2.*pi] cos(x)
[2.*pi:10] (x-8)**2
4
3
2
1
0
-1
-2
-2
0
2
4
6
8
10

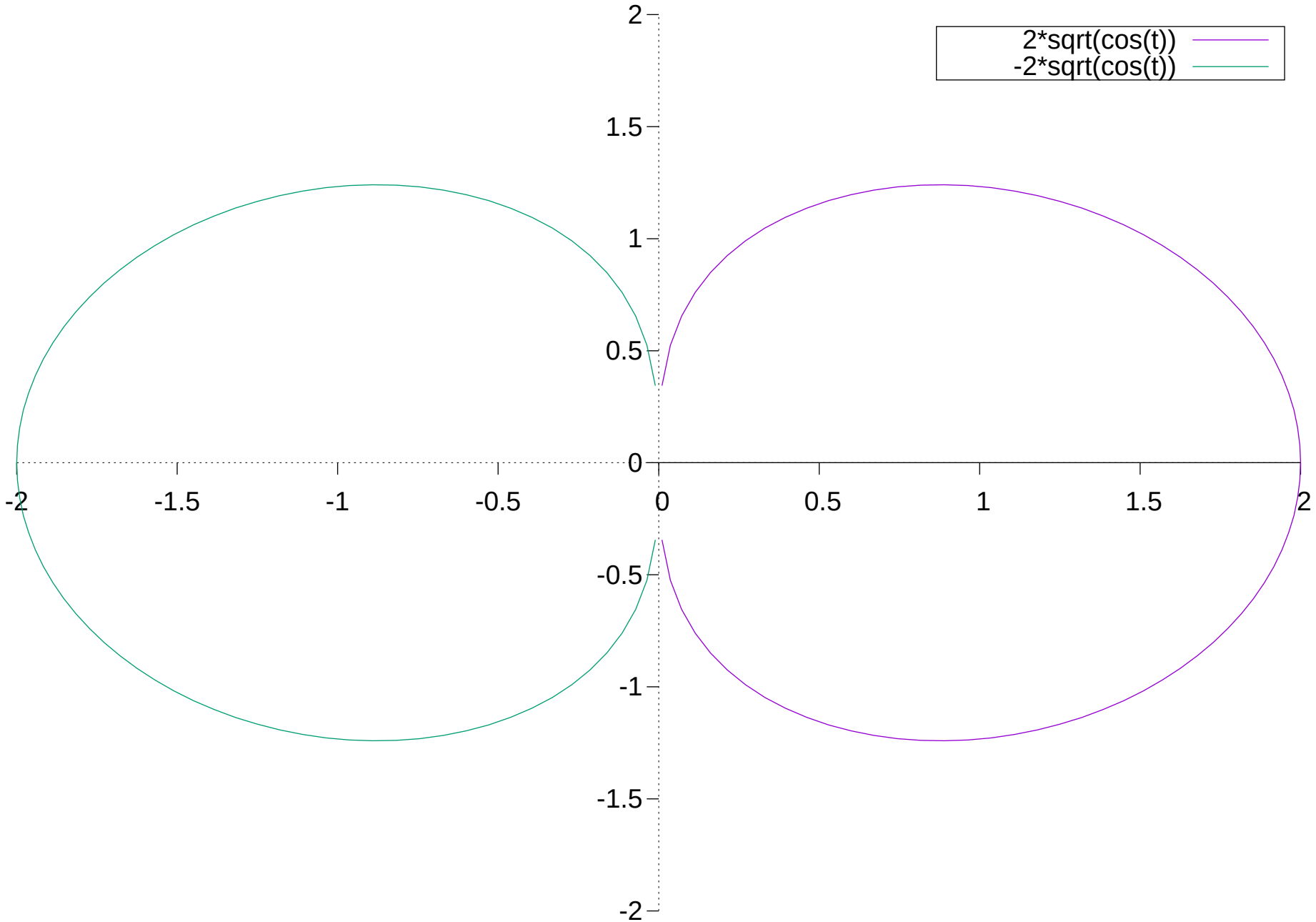

Piecewise approximation to the
Normal Cumulative Distribution Function
part1: for x < 1 norm(x) ‰ + (9x-x^3) / 24
part2: for x > 1 norm(x) 1 + (x-3)^3 / 48
norm(x)
plot norm(x), [0:1] part1(x), [1:4] part2(x)
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

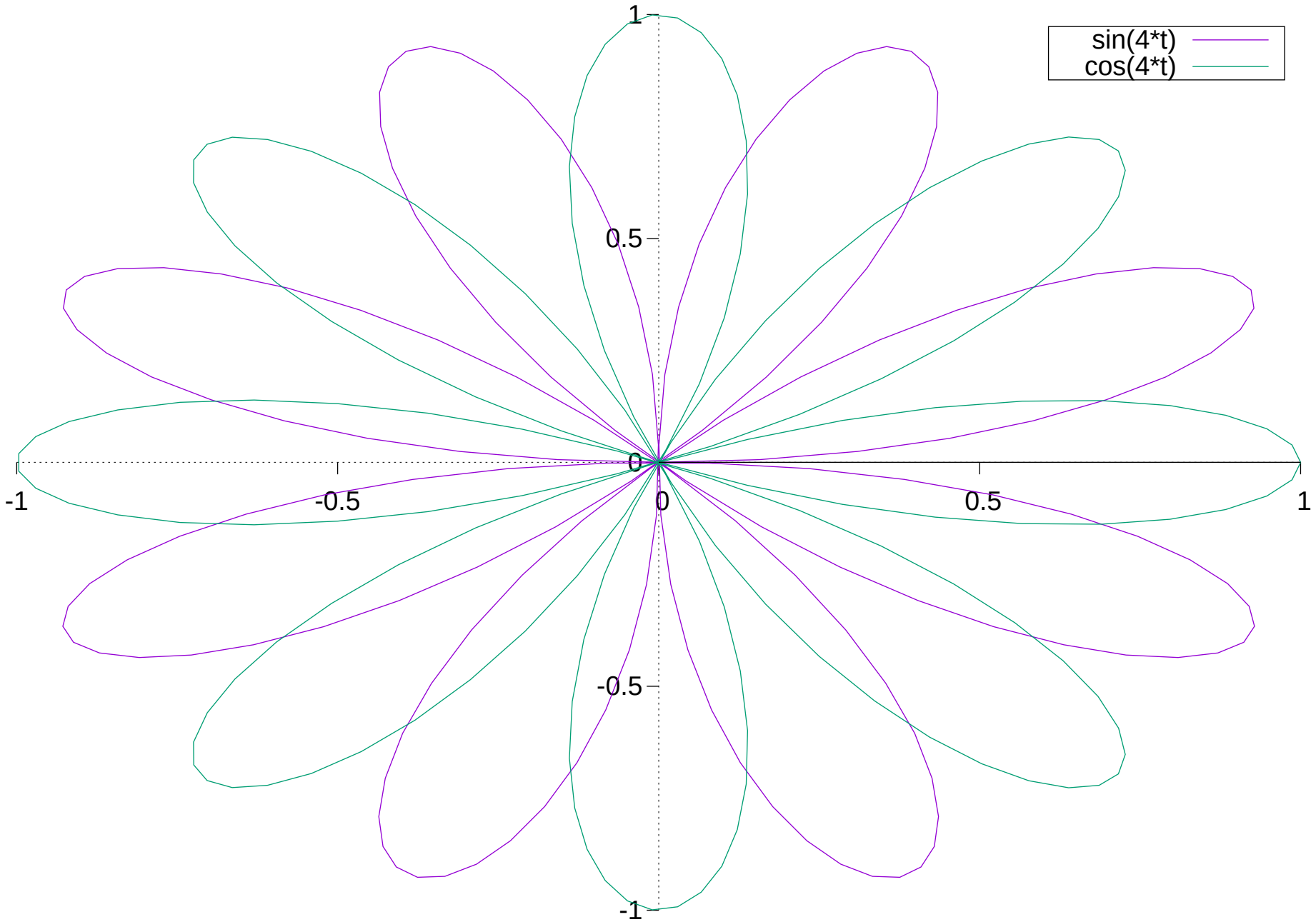
Piecewise function of one parameter in 3D

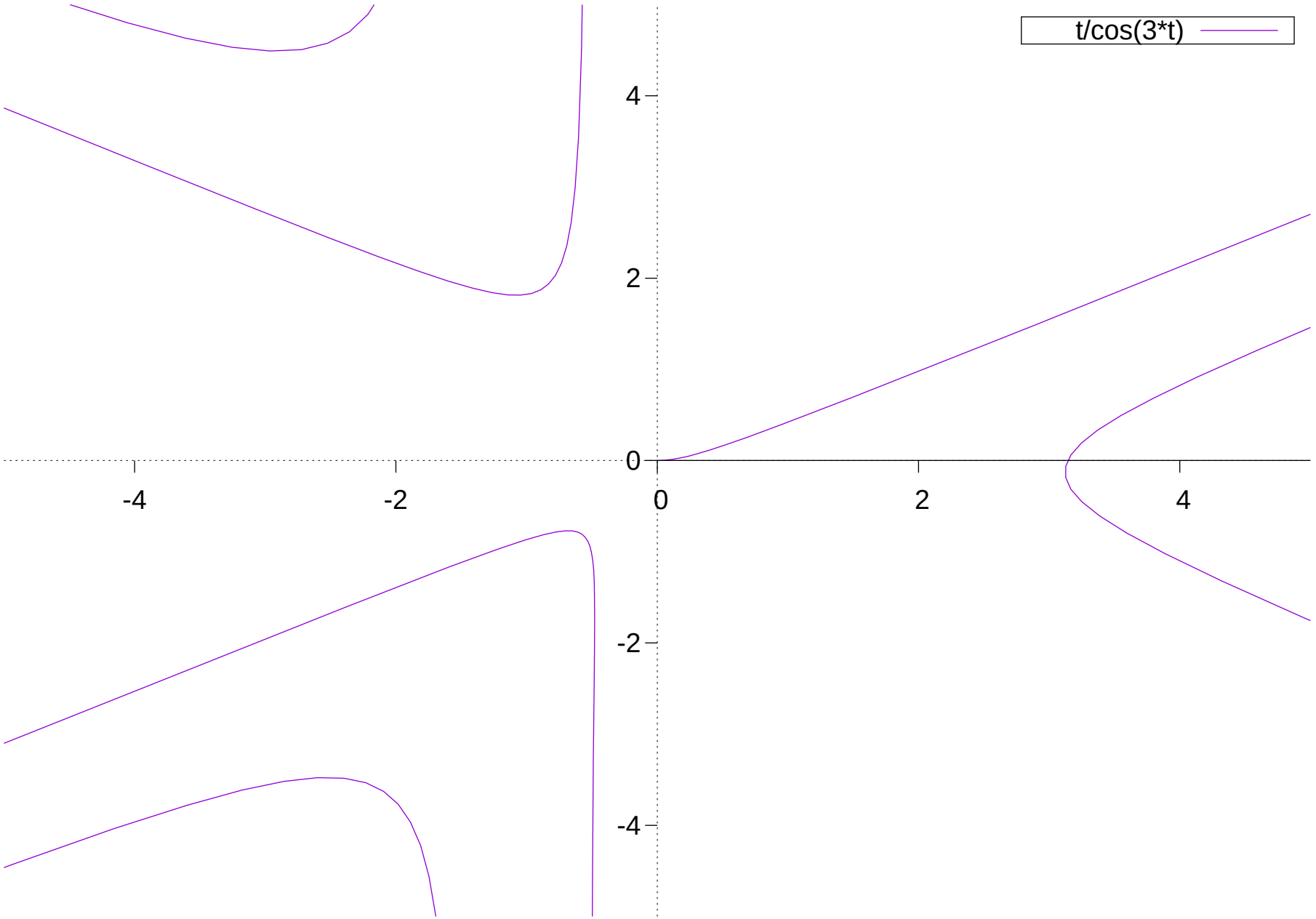


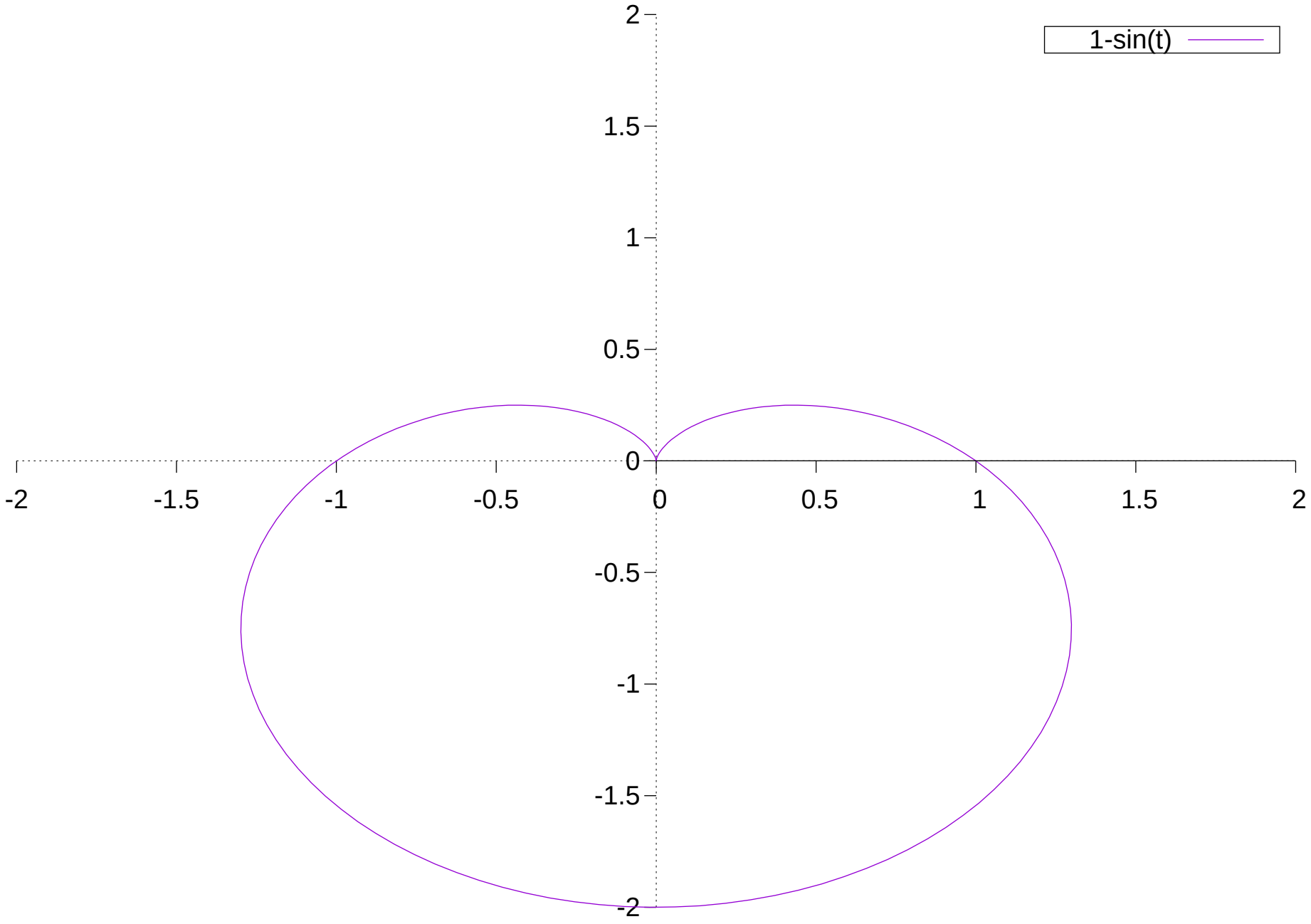

Three circles (with aspect ratio distortion)
.5
1
1.5
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5

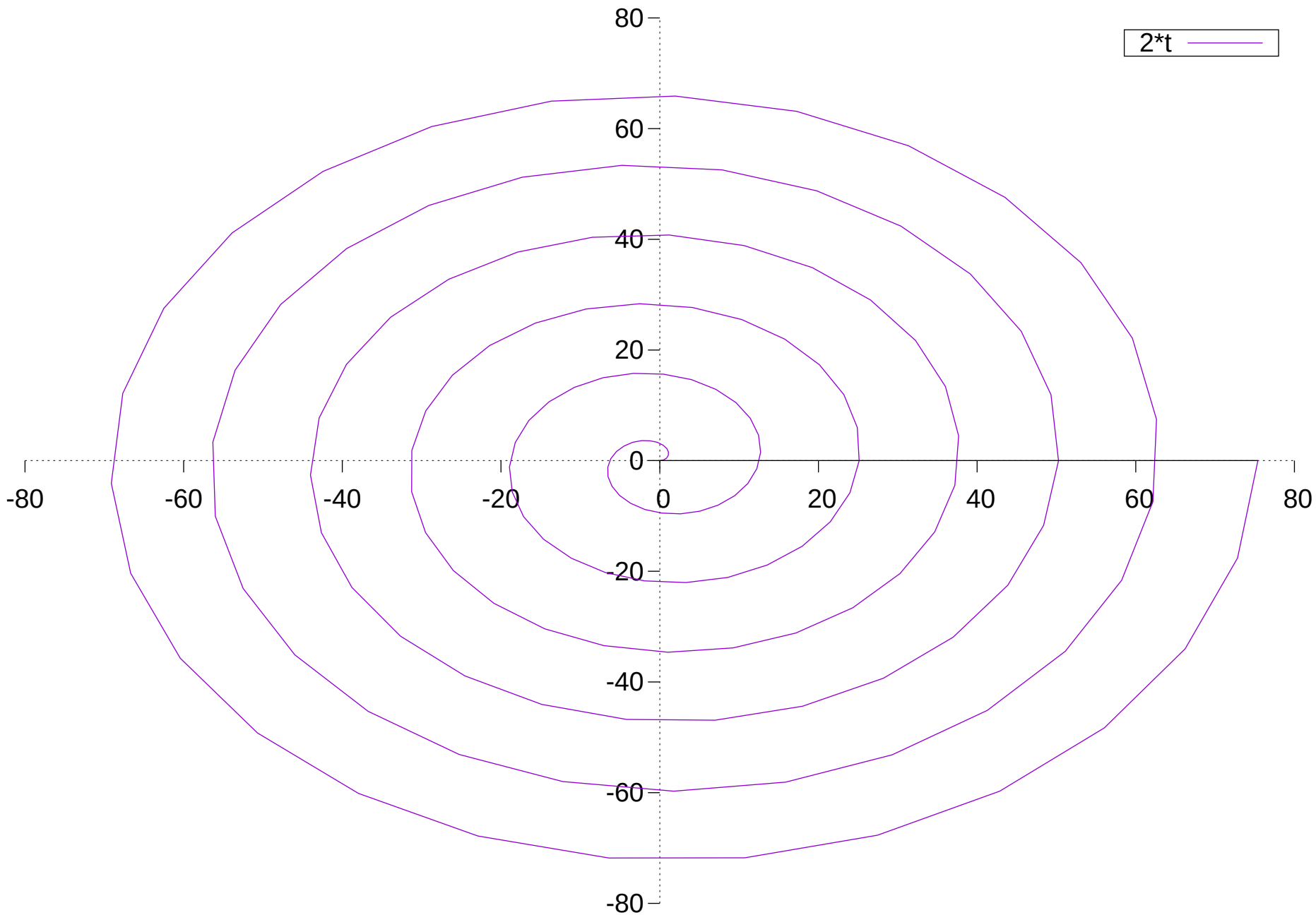

cos(2*t)
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

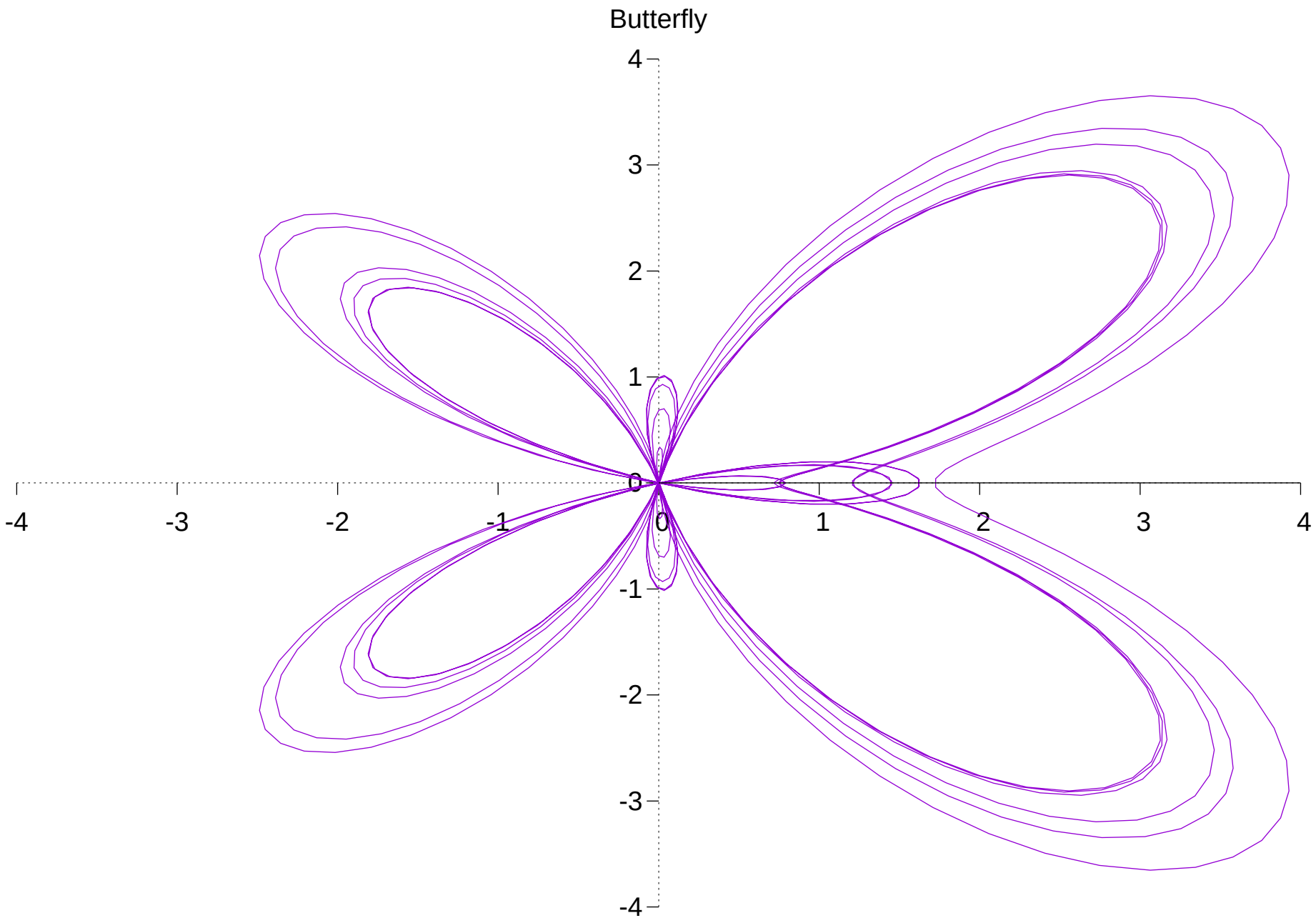

2*sqrt(cos(t))
-2*sqrt(cos(t))
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

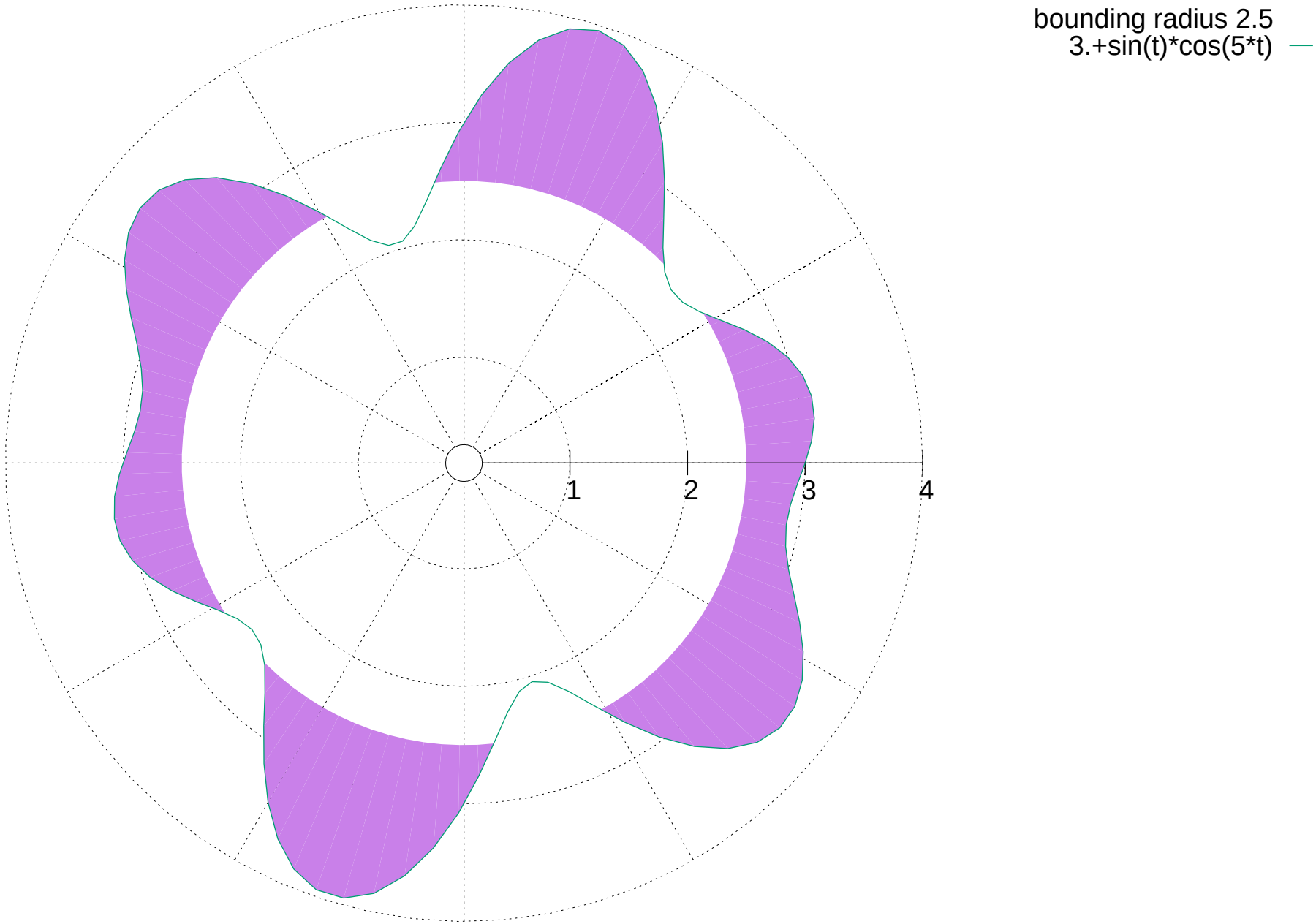

sin(4*t)
cos(4*t)
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

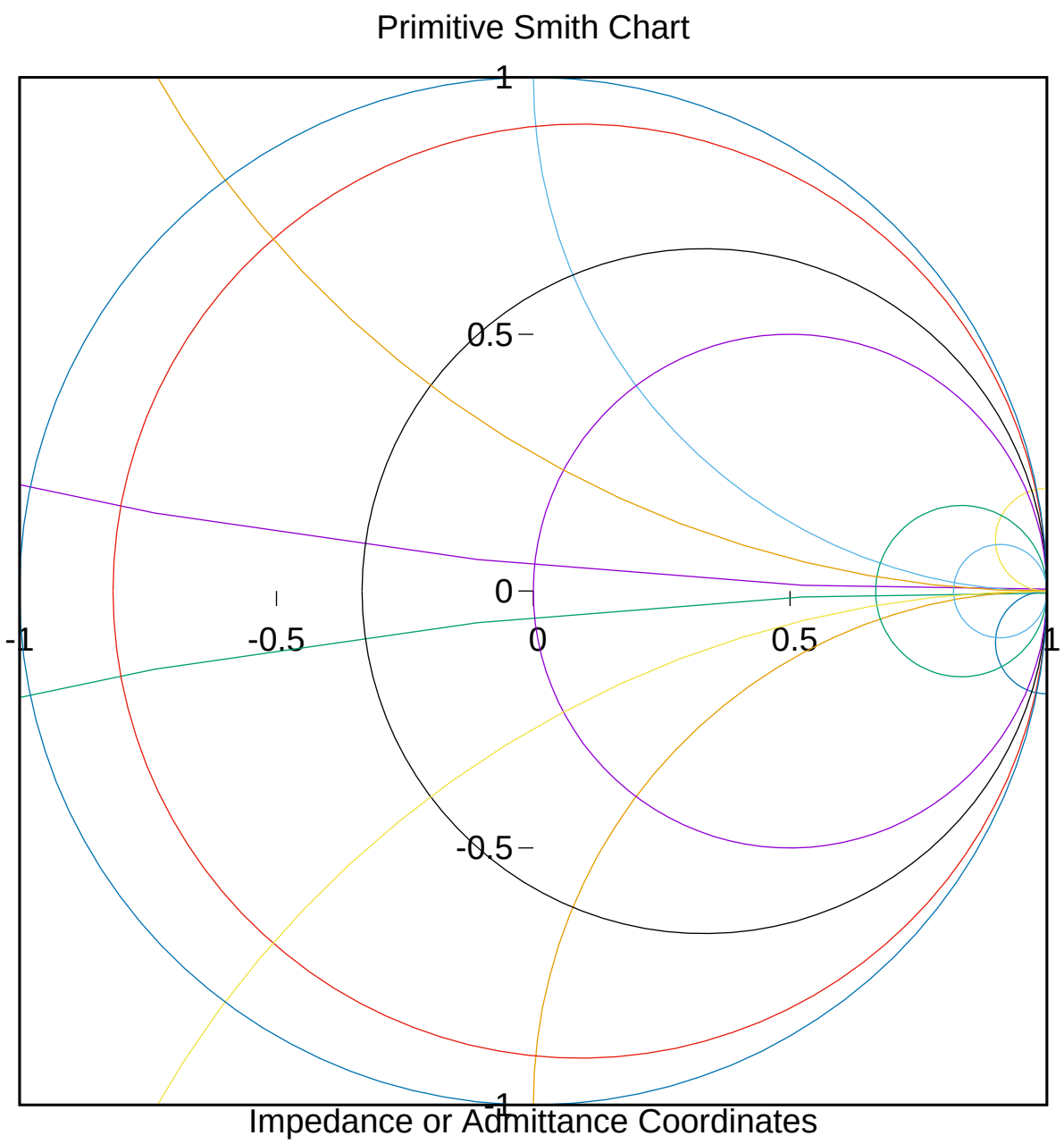

t/cos(3*t)
4
2
0
-2
-4
-4
-2
0
2
4

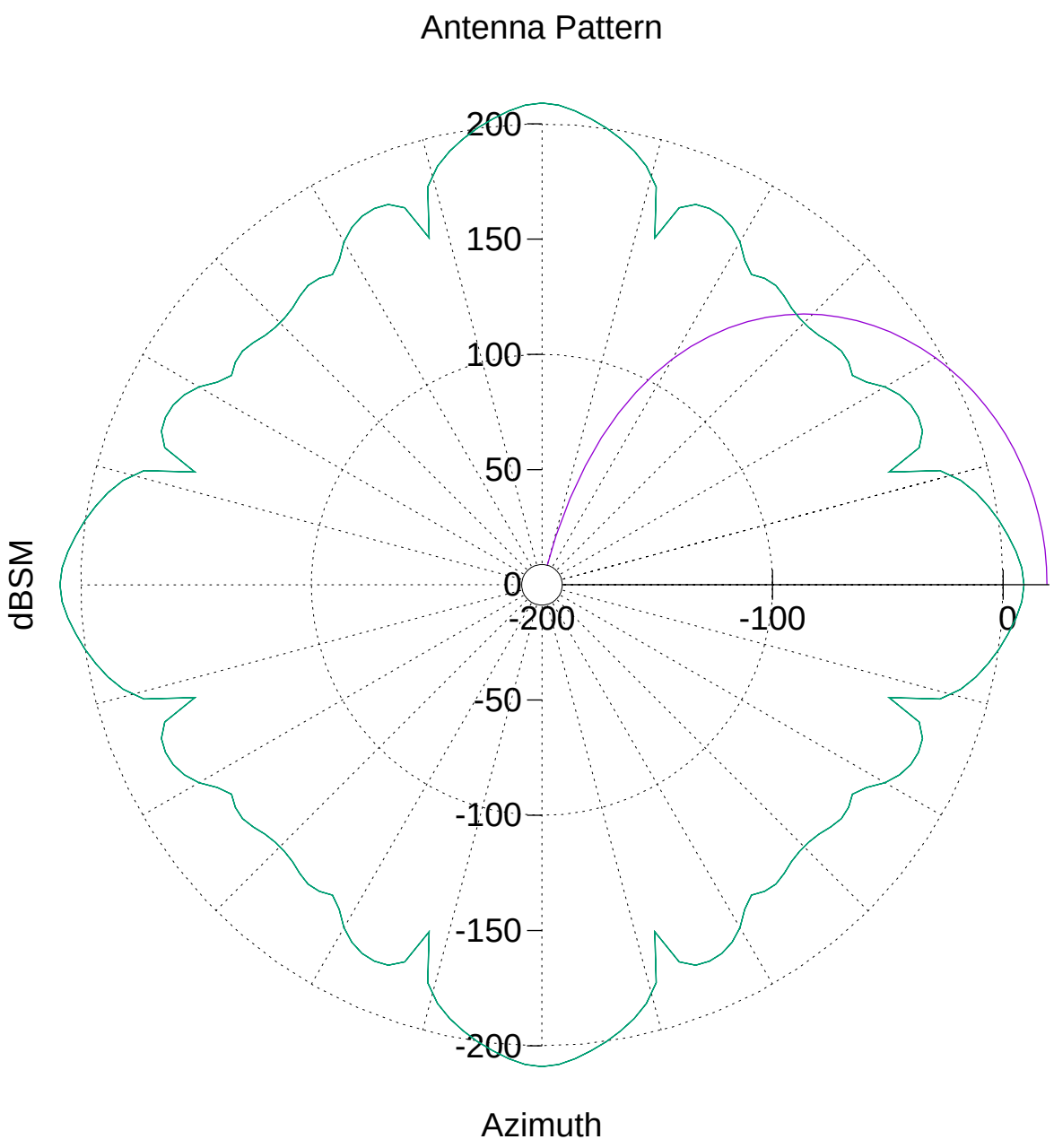

1-sin(t)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2


2*t
80
60
40
20
0
-20
-40
-60
-80
-80
-60
-40
-20
0
20
40
60
80

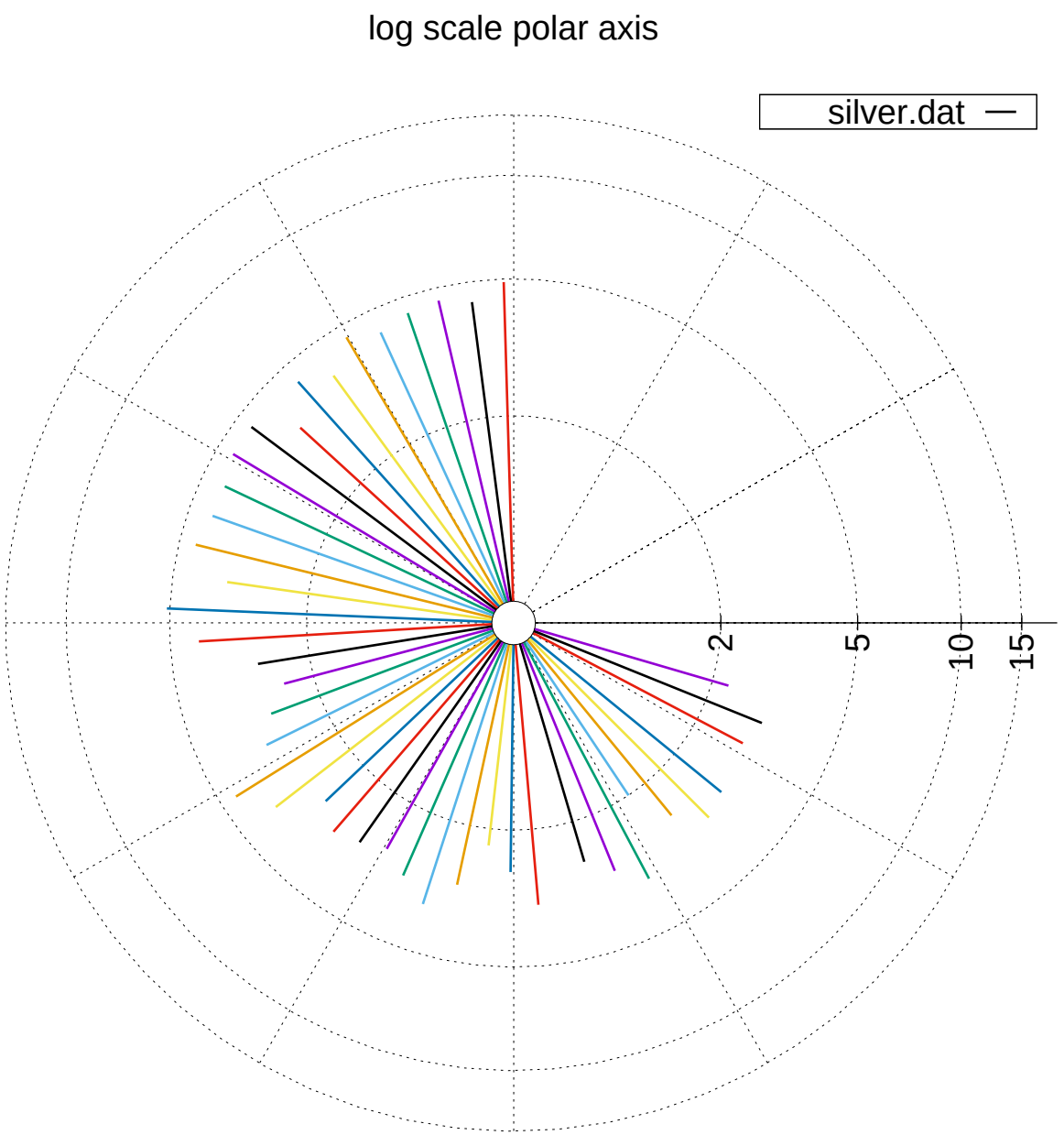

Butterfly
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4

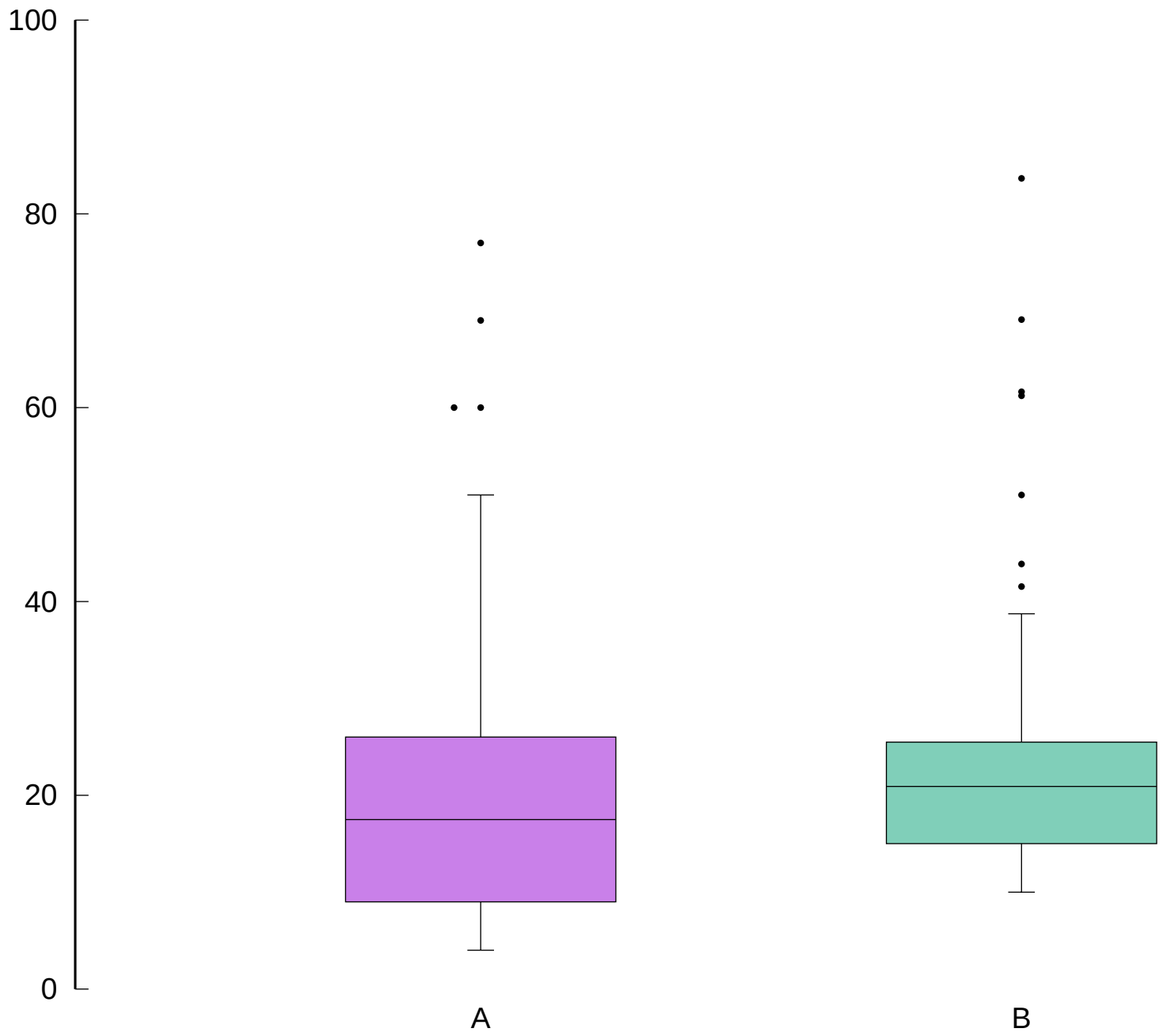

bounding radius 2.5
3.+sin(t)*cos(5*t)
1
2
3
4

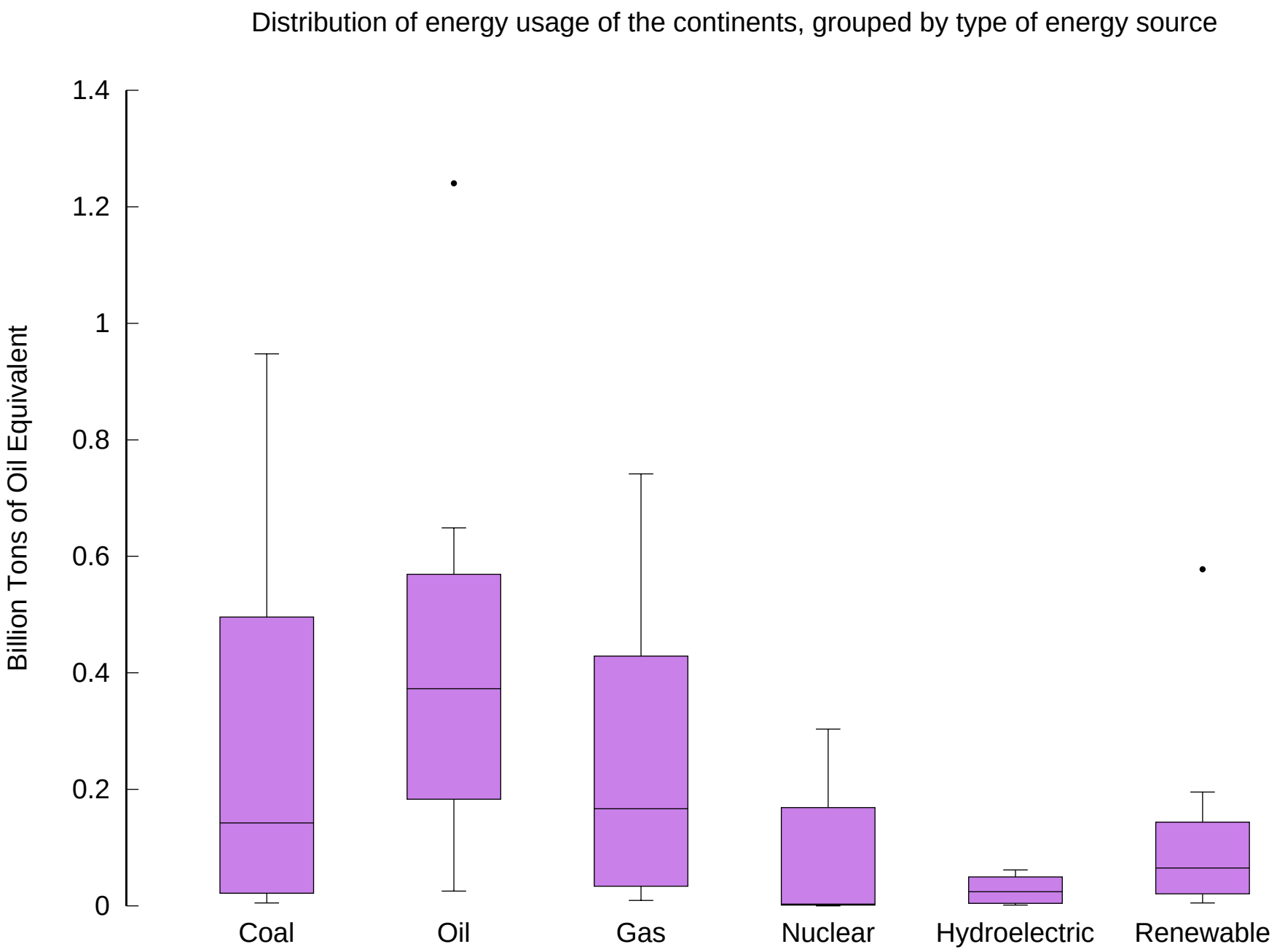

Primitive Smith Chart
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
Impedance or Admittance Coordinates
(¶â 10•î 07 14:39:49 2016

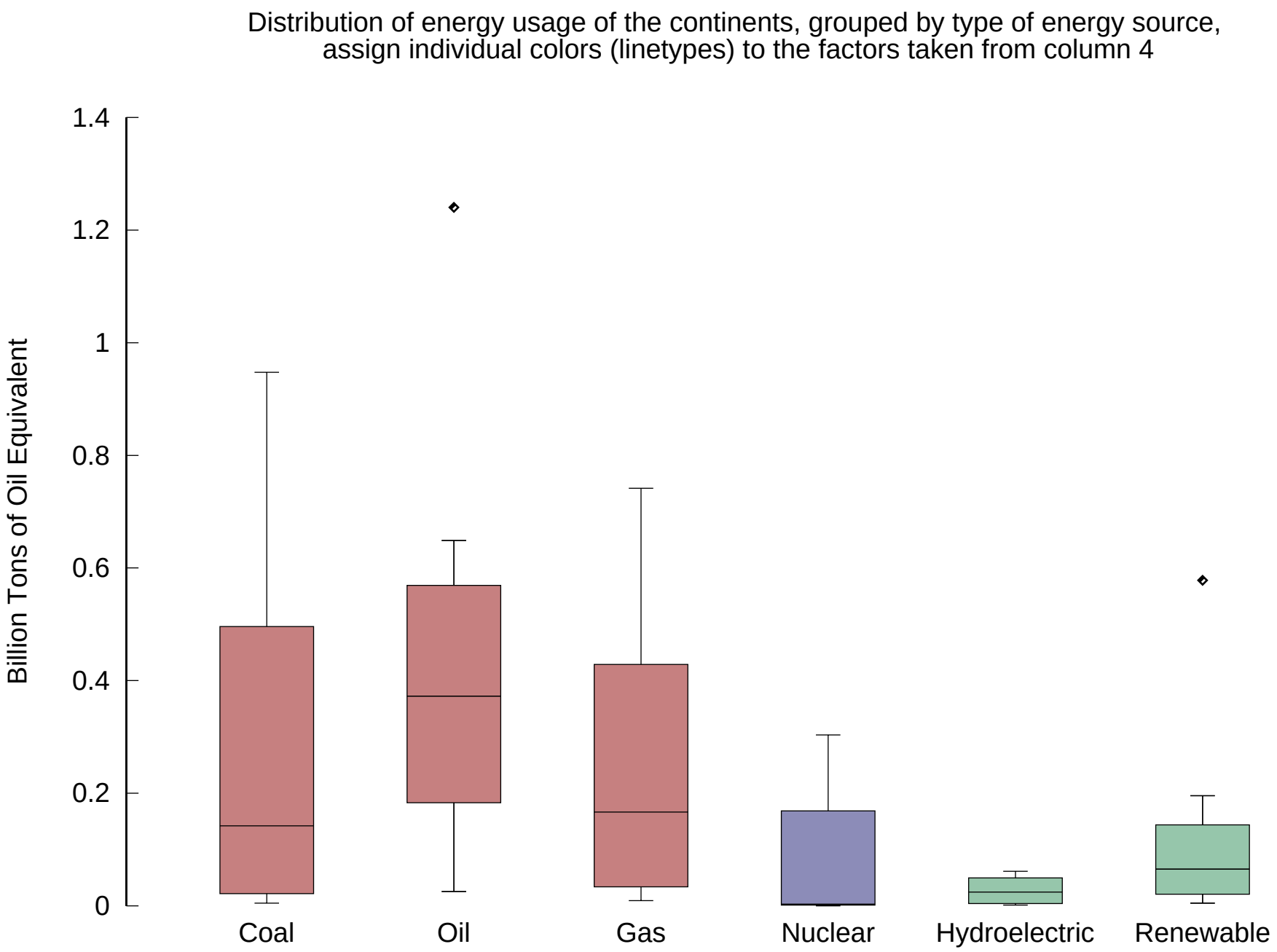


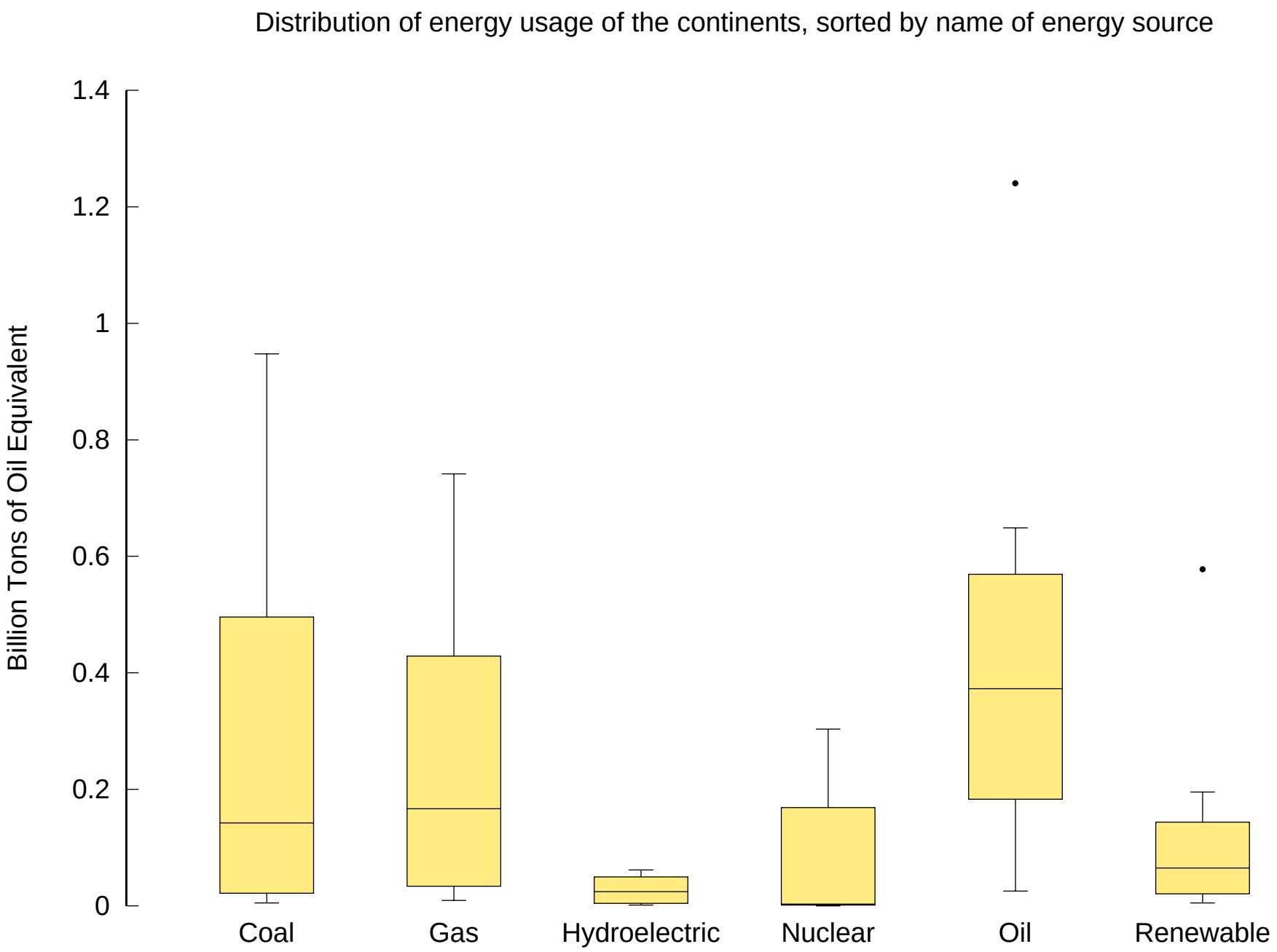
(¶â 10•î 07 14:39:49 2016

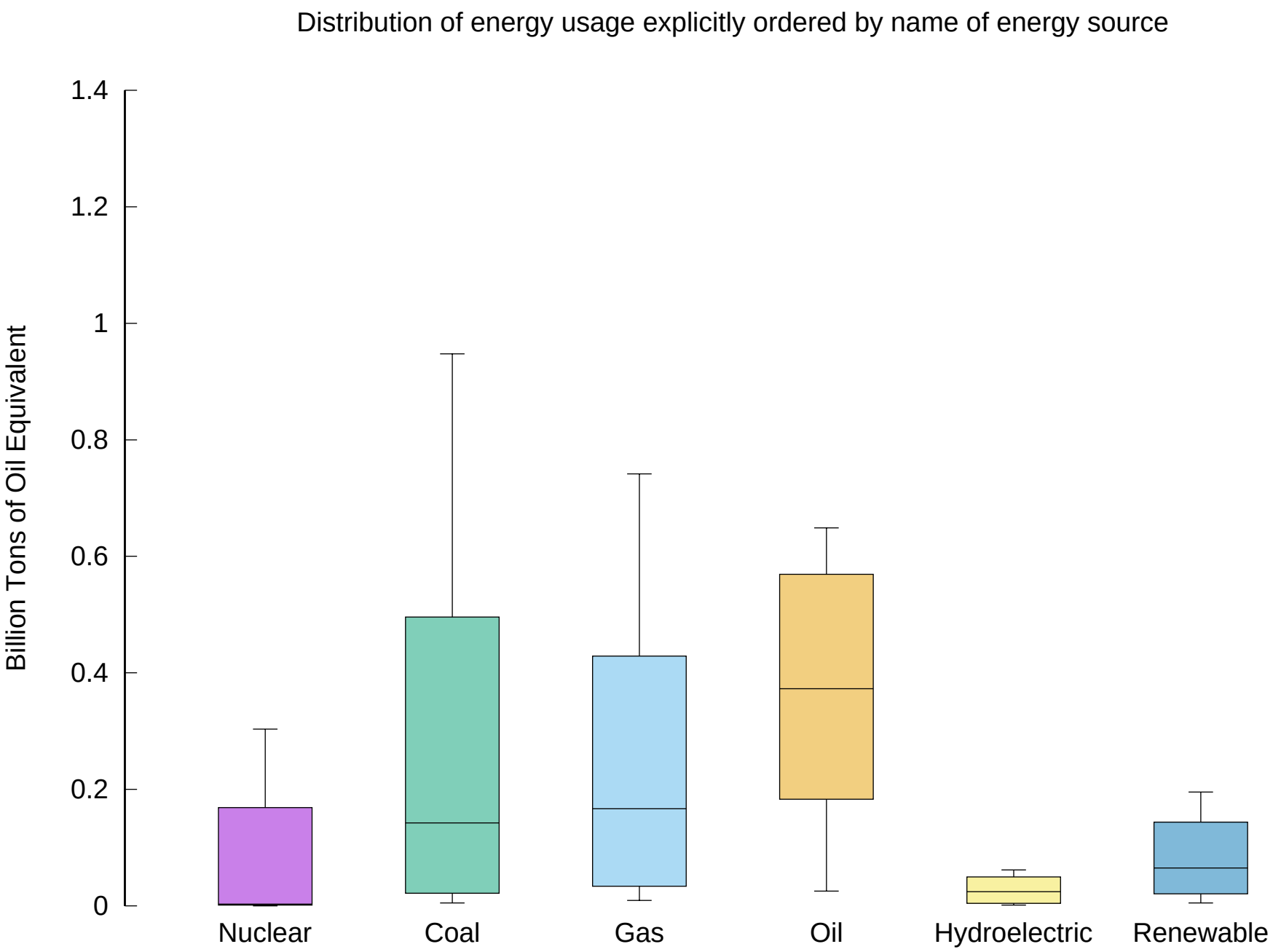

log scale polar axis
silver.dat
2
5
10
15

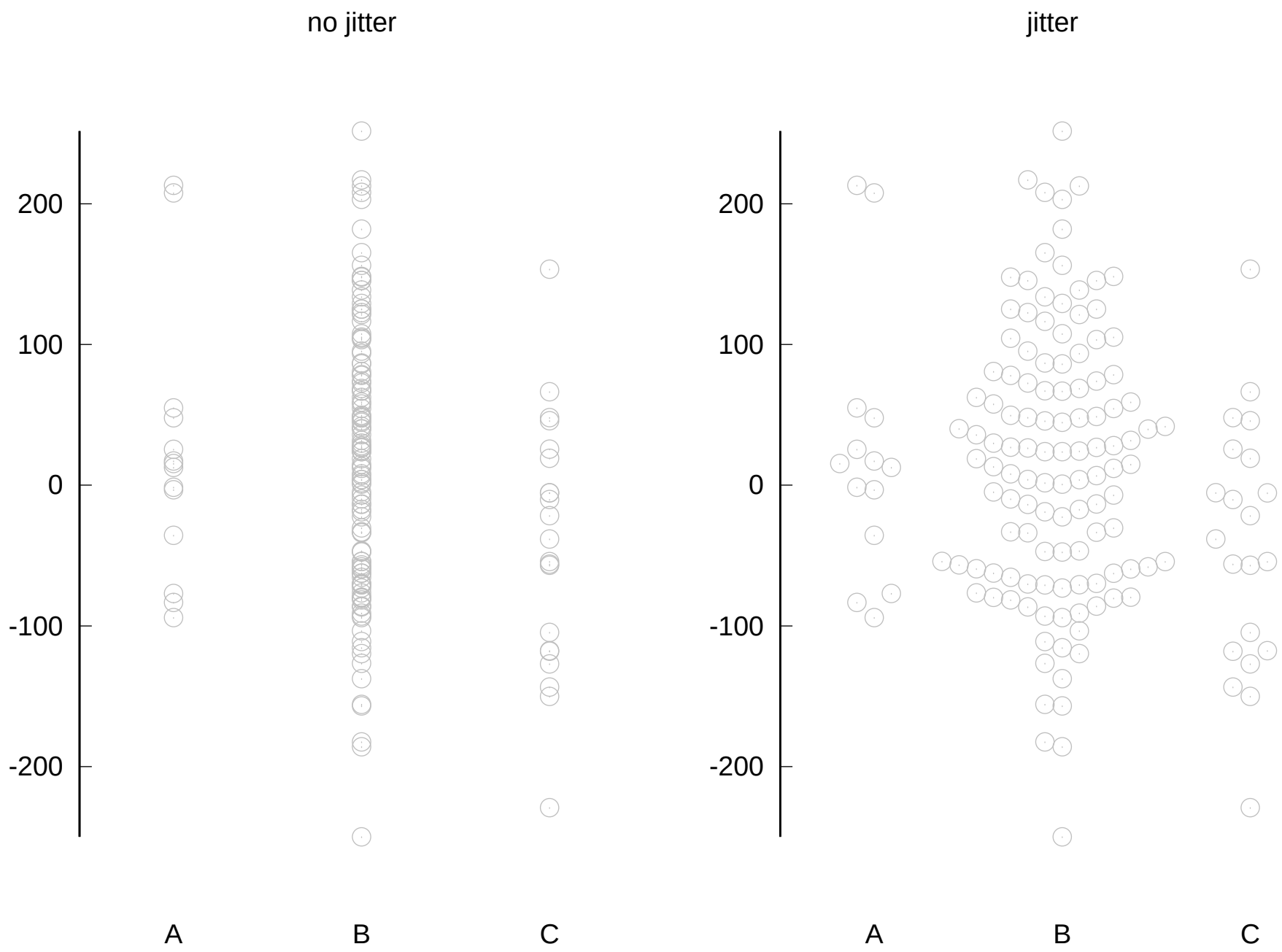

100
80
60
40
20
0
A
B

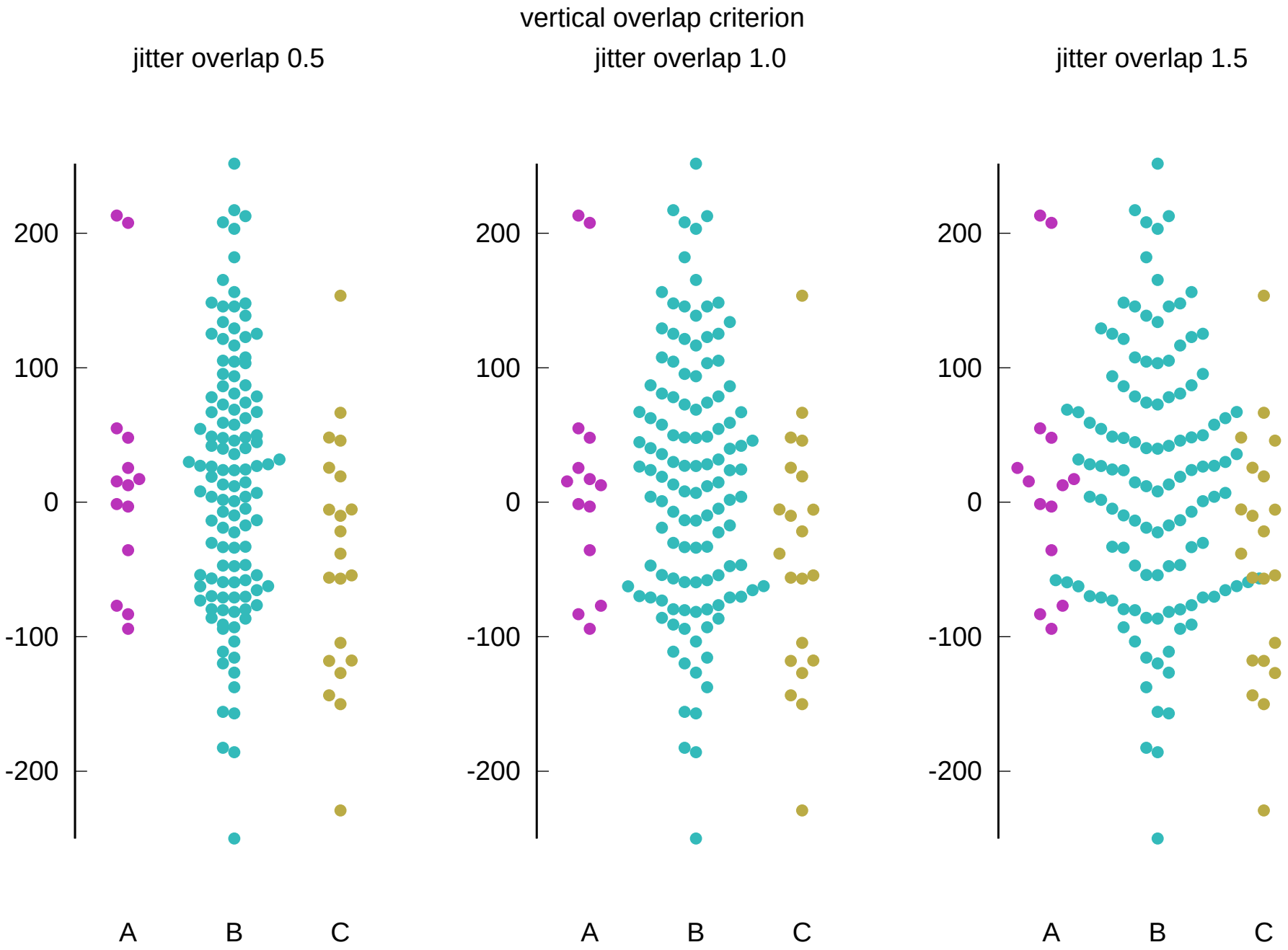

Distribution of energy usage of the continents, grouped by type of energy source
Billion Tons of Oil Equivalent
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Coal
Oil
Gas
Nuclear
Hydroelectric
Renewable

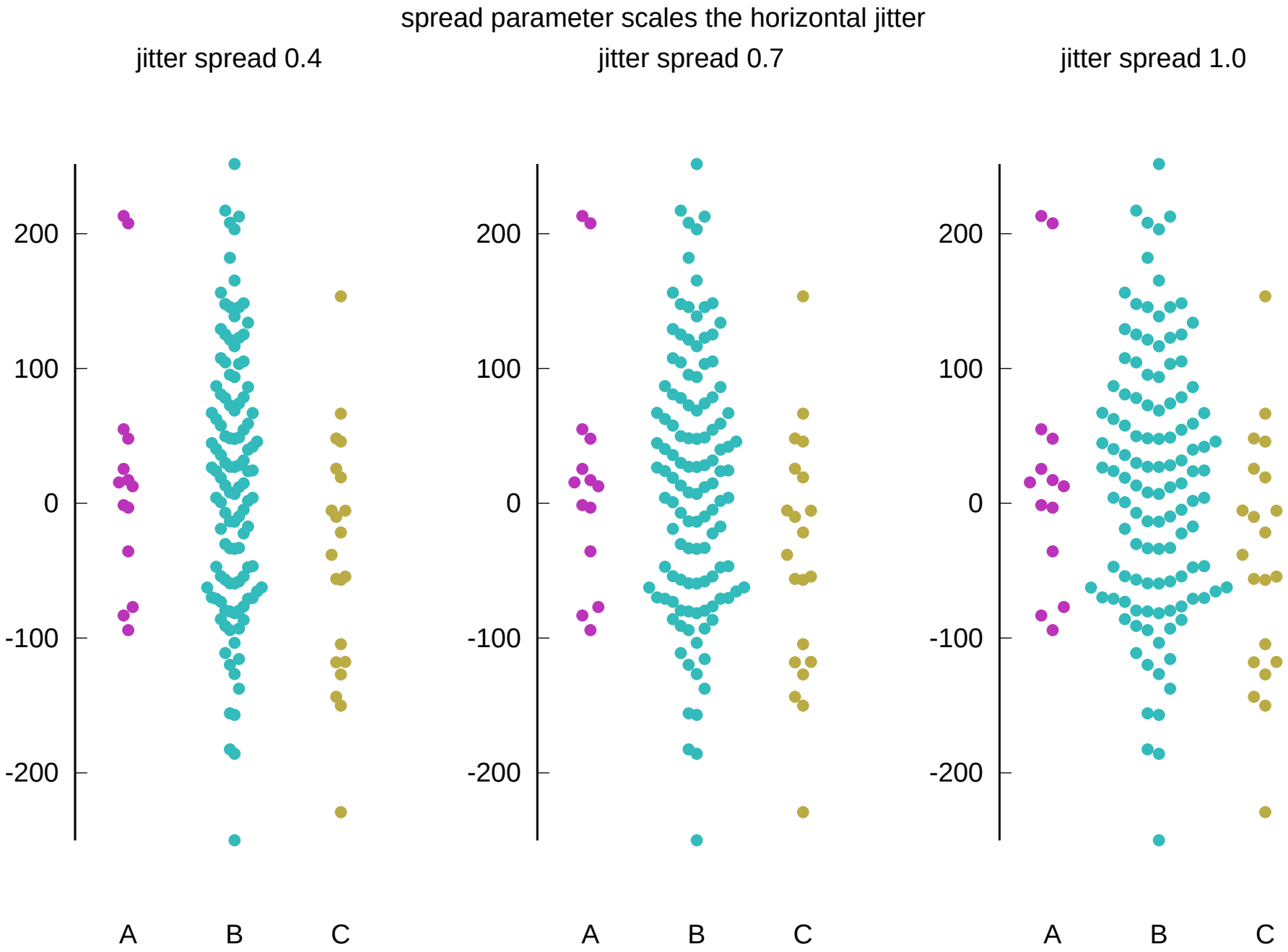

Distribution of energy usage of the continents, grouped by type of energy source,
assign individual colors (linetypes) to the factors taken from column 4
Billion Tons of Oil Equivalent
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Coal
Oil
Gas
Nuclear
Hydroelectric
Renewable

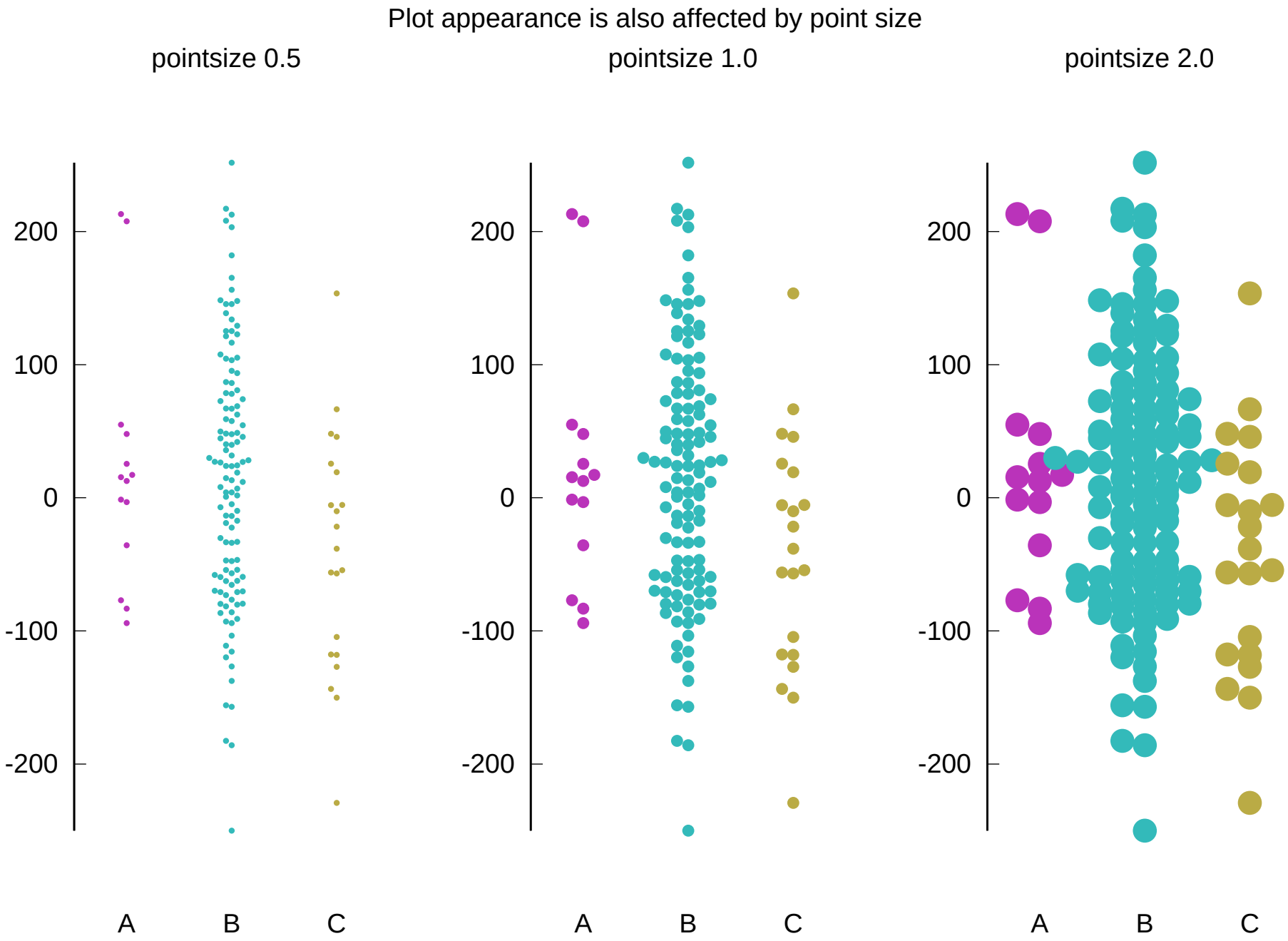

Distribution of energy usage of the continents, sorted by name of energy source
Billion Tons of Oil Equivalent
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Coal
Gas
Hydroelectric
Nuclear
Oil
Renewable

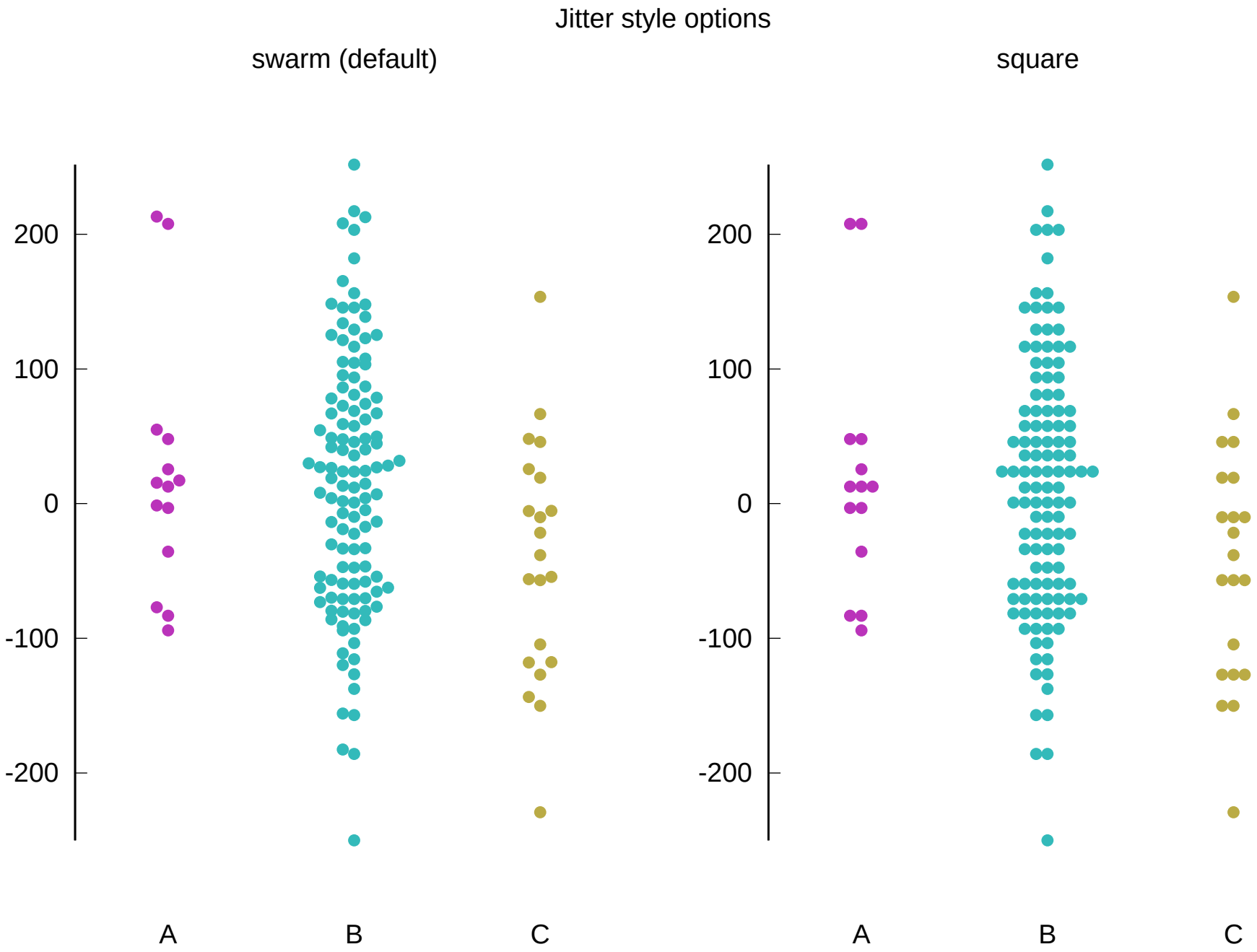

Distribution of energy usage explicitly ordered by name of energy source
Billion Tons of Oil Equivalent
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Nuclear
Coal
Gas
Oil
Hydroelectric
Renewable

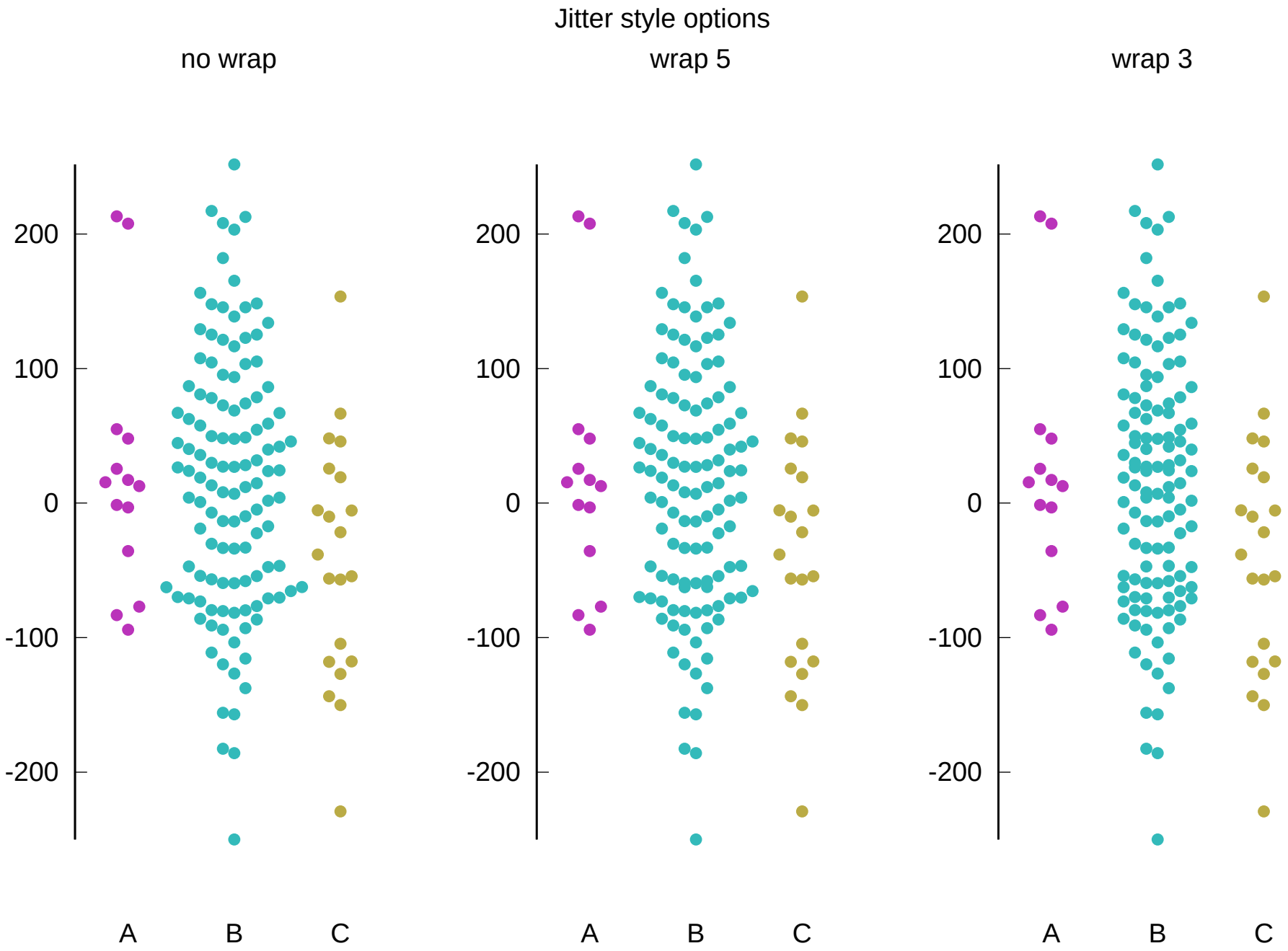

no jitter
jitter
200
100
0
-100
-200
A
B
C

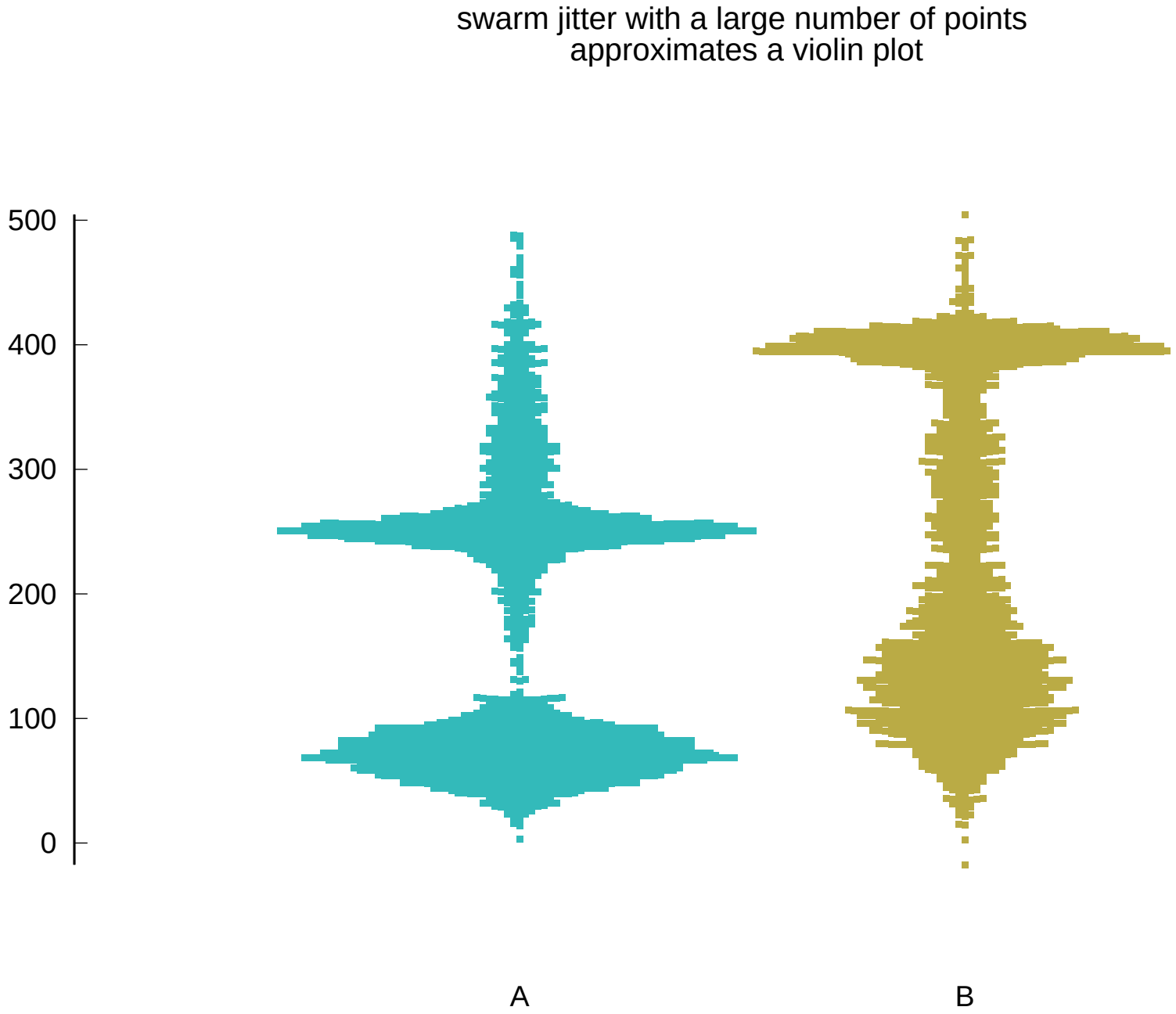

vertical overlap criterion
jitter overlap 0.5
jitter overlap 1.0
jitter overlap 1.5
200
100
0
-100
-200
A
B
C

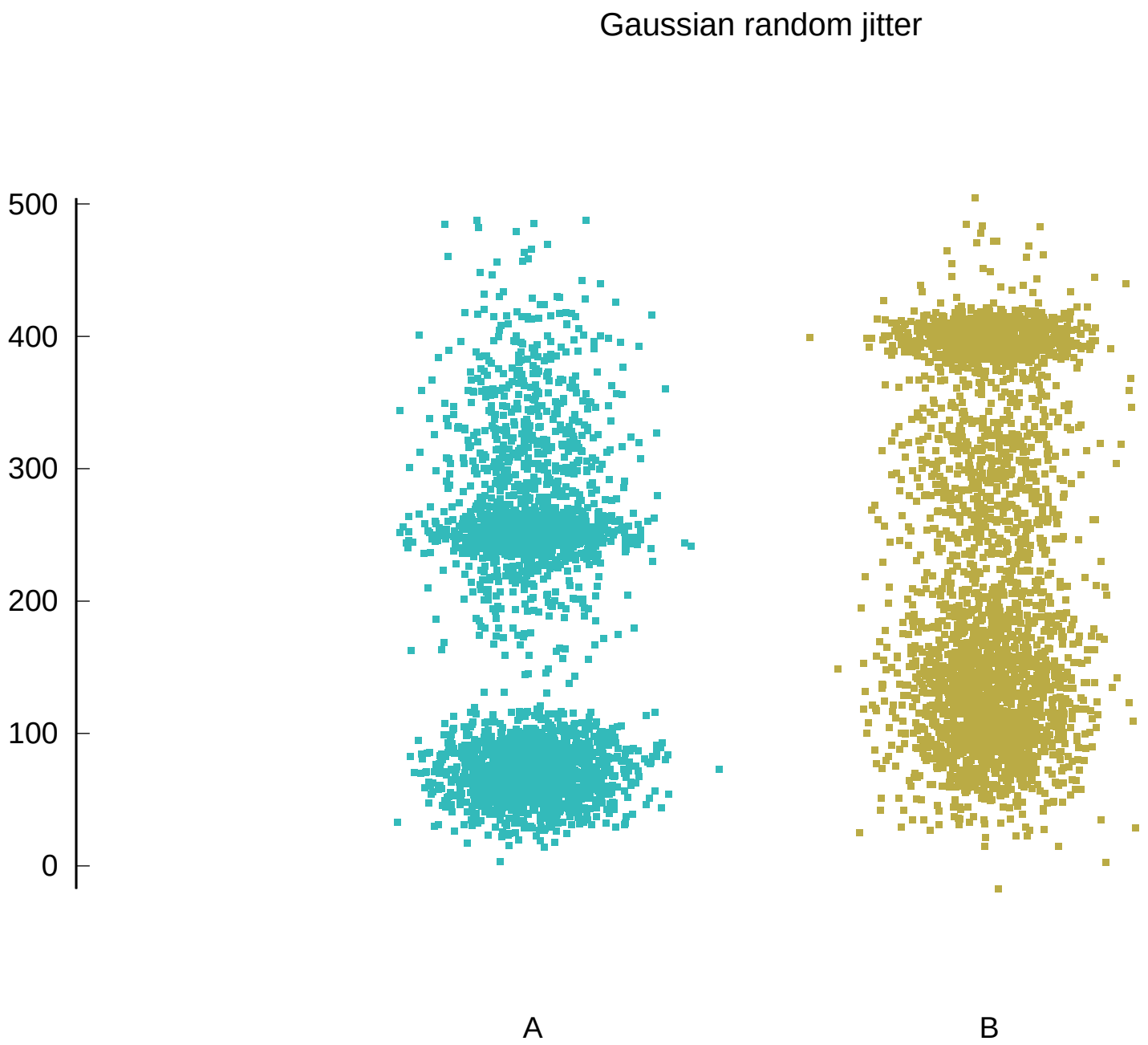

spread parameter scales the horizontal jitter
jitter spread 0.4
jitter spread 0.7
jitter spread 1.0
200
100
0
-100
-200
A
B
C

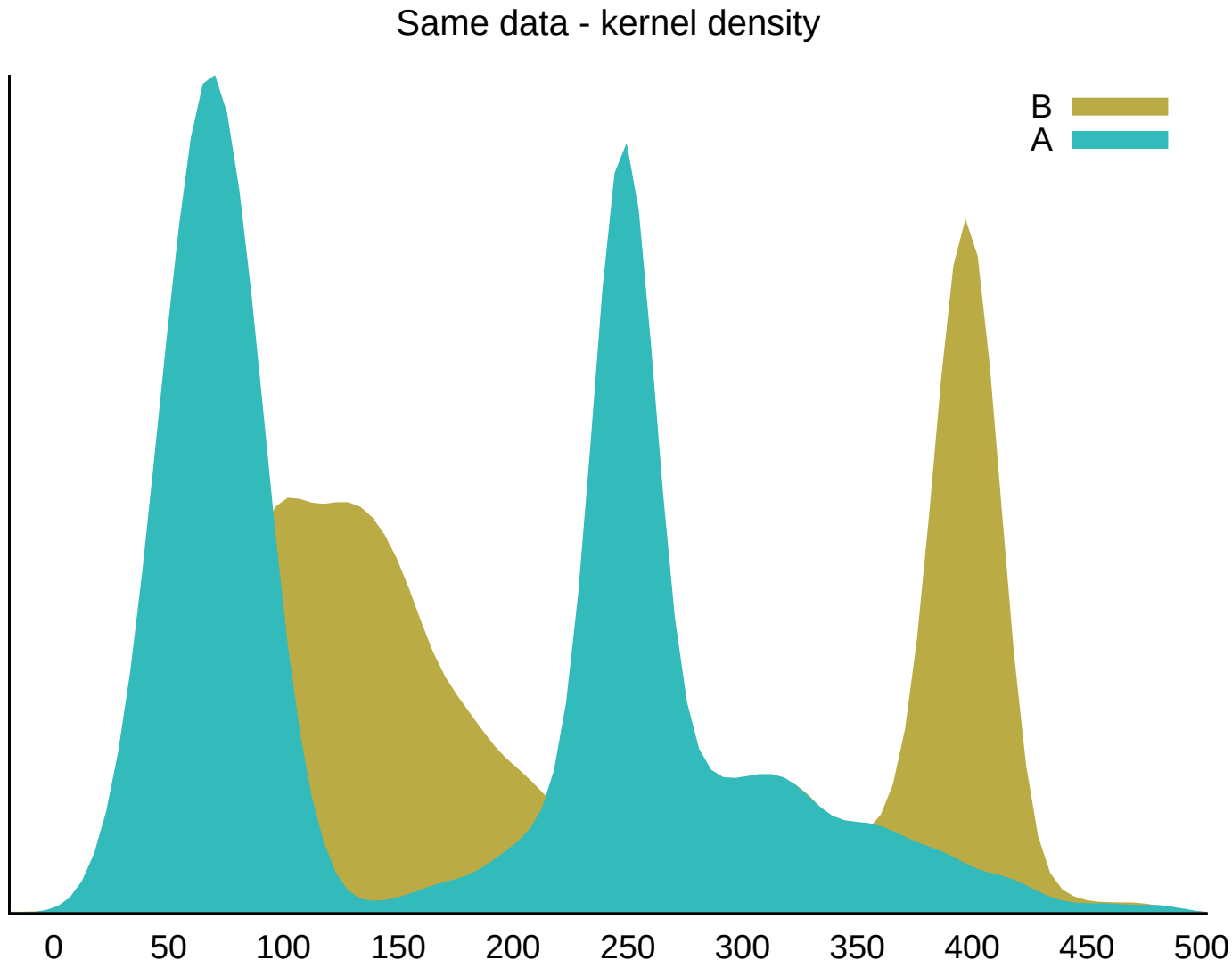

Plot appearance is also affected by point size
pointsize 0.5
pointsize 1.0
pointsize 2.0
200
100
0
-100
-200
A
B
C


Jitter style options
swarm (default)
square
200
100
0
-100
-200
A
B
C


Jitter style options
no wrap
wrap 5
wrap 3
200
100
0
-100
-200
A
B
C

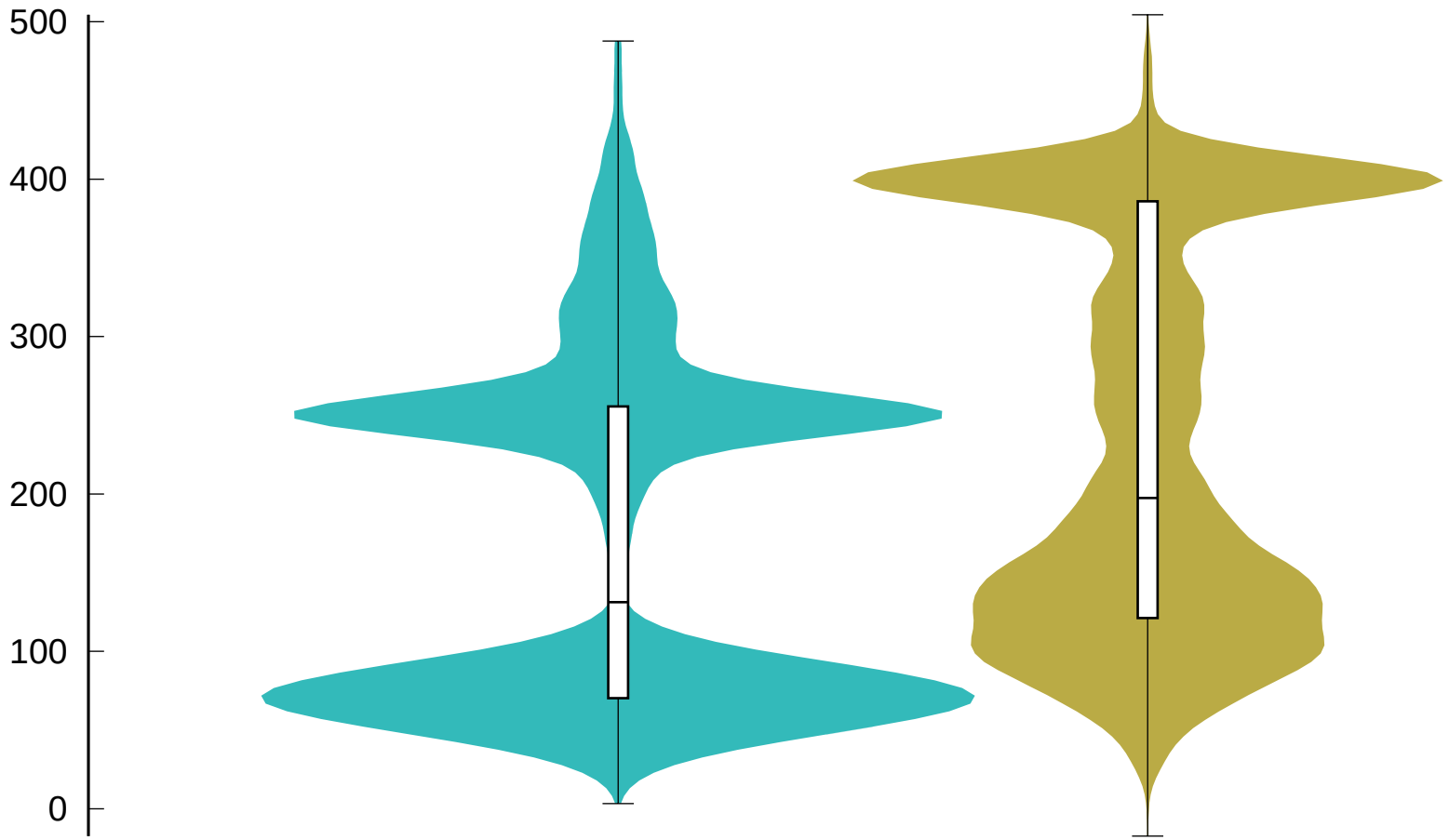

swarm jitter with a large number of points
approximates a violin plot
500
400
300
200
100
0
A
B

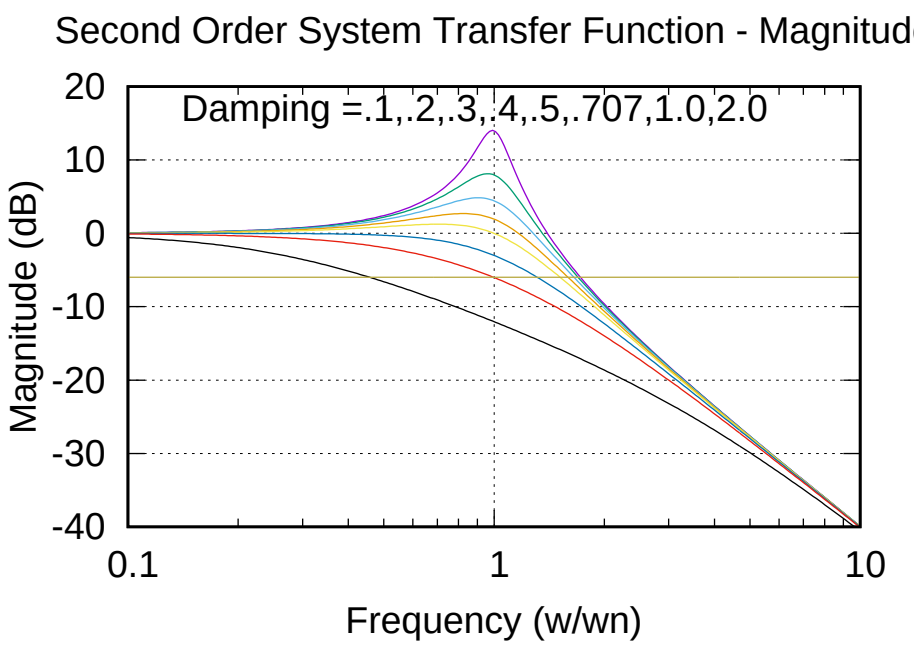

Gaussian random jitter
500
400
300
200
100
0
A
B

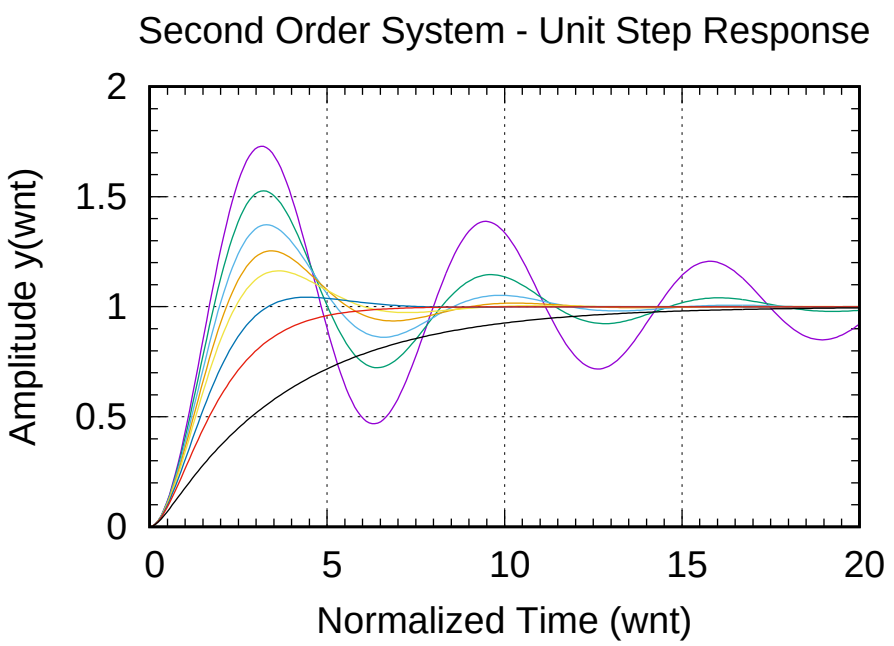

Same data - kernel density
B
A
0
50
100
150
200
250
300
350
400
450
500

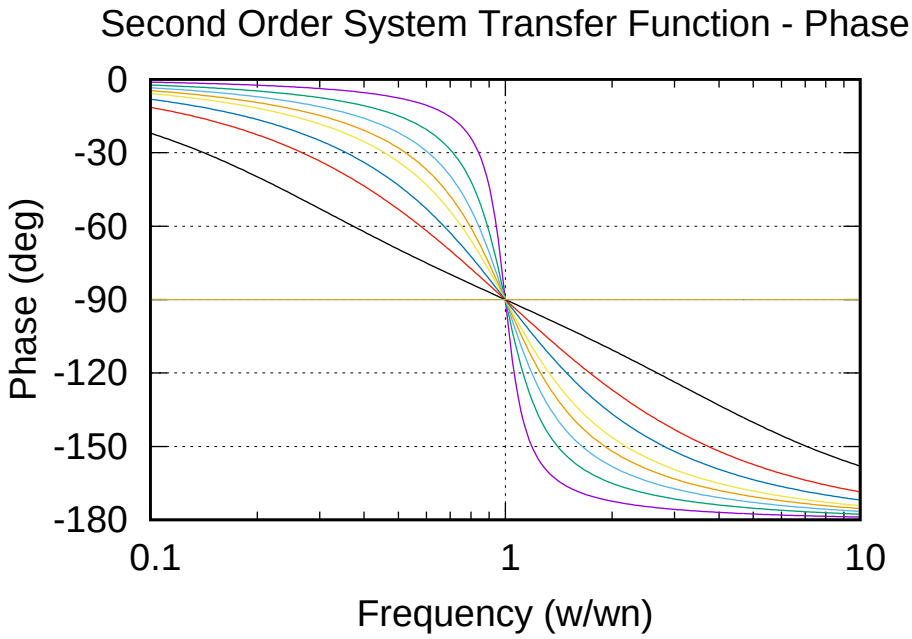
kdensity mirrored sideways to give a violin plot

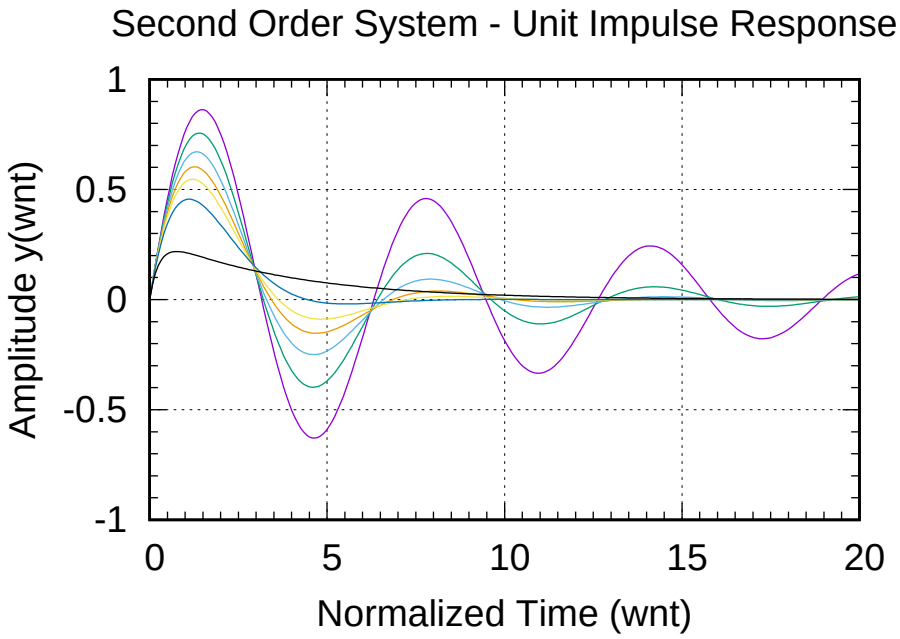
Superimposed violin plot and box plot

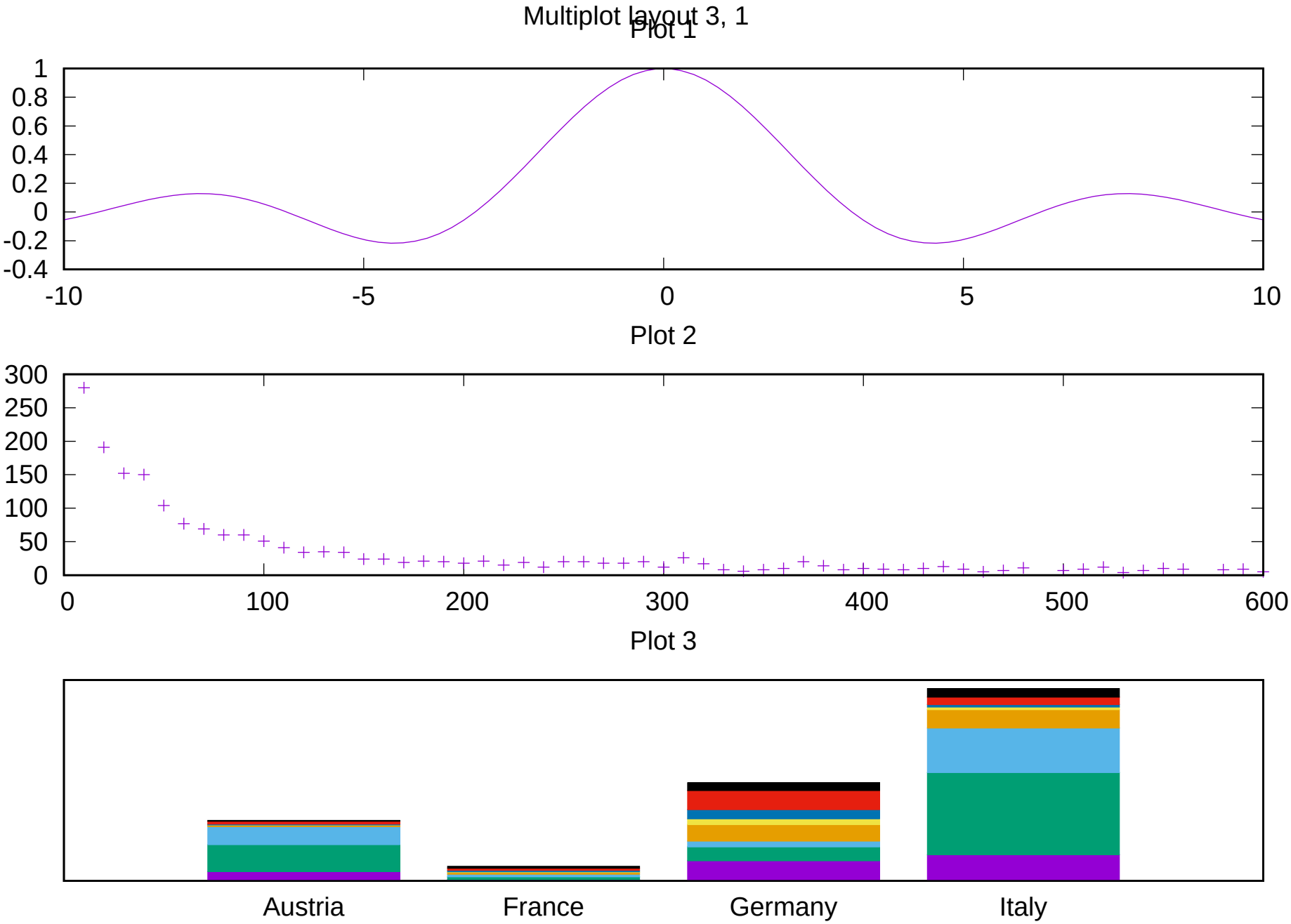


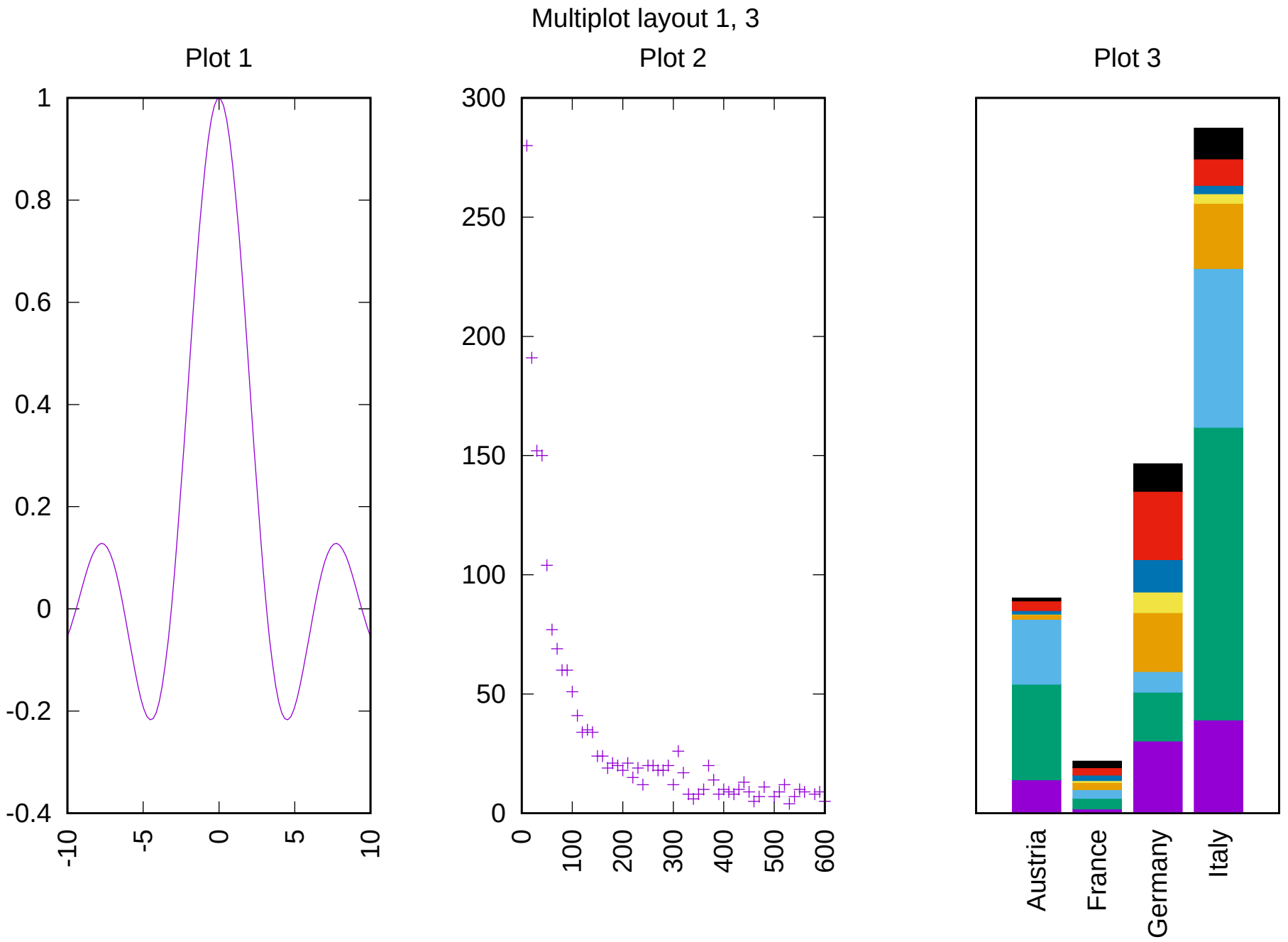

Second Order System Transfer Function - Magnitude
Damping =.1,.2,.3,.4,.5,.707,1.0,2.0
Magnitude (dB)
20
10
0
-10
-20
-30
-40
0.1
1
10
Frequency (w/wn)

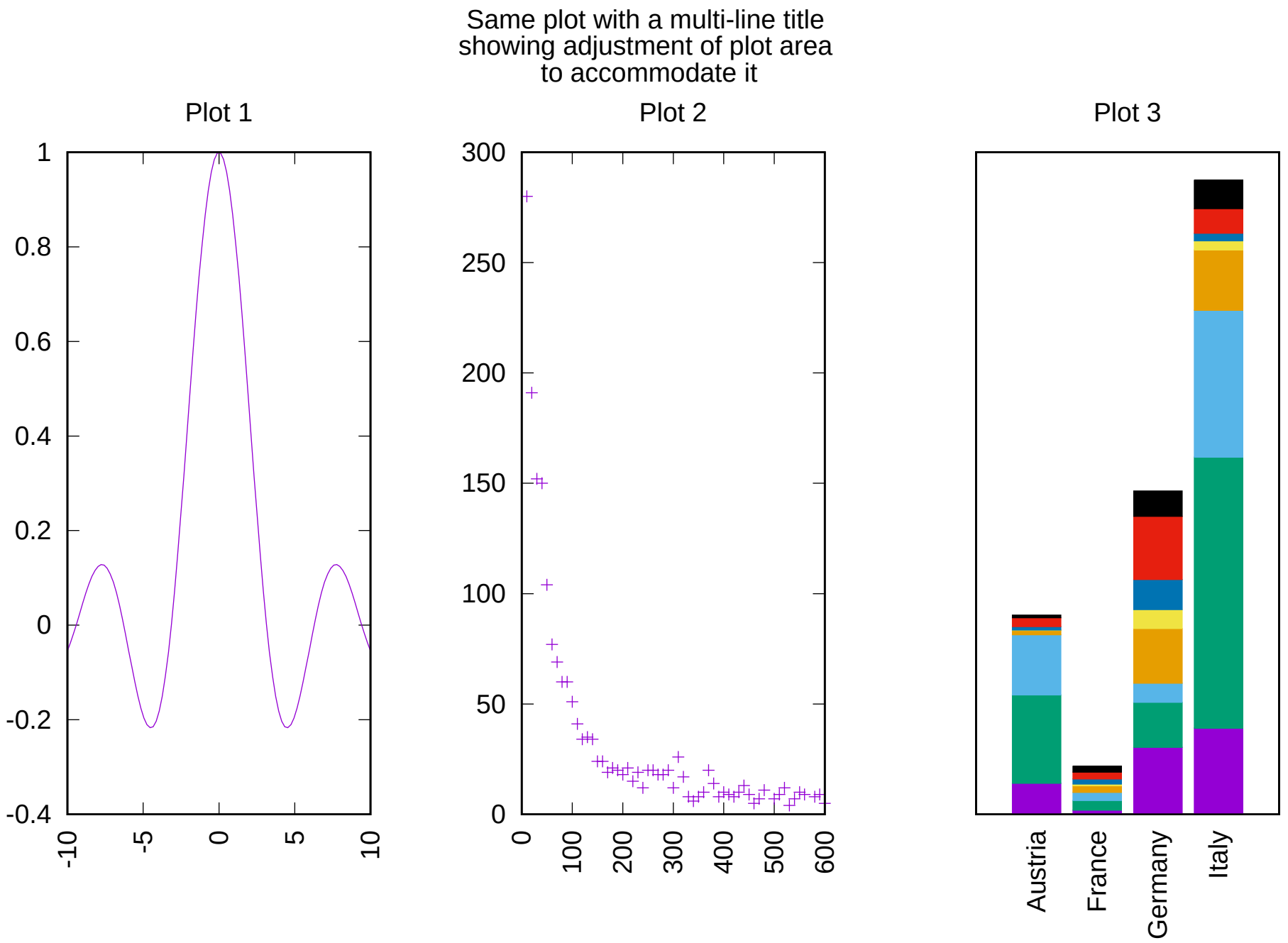

Second Order System - Unit Step Response
Amplitude y(wnt)
2
1.5
1
0.5
0
0
5
10
15
20
Normalized Time (wnt)

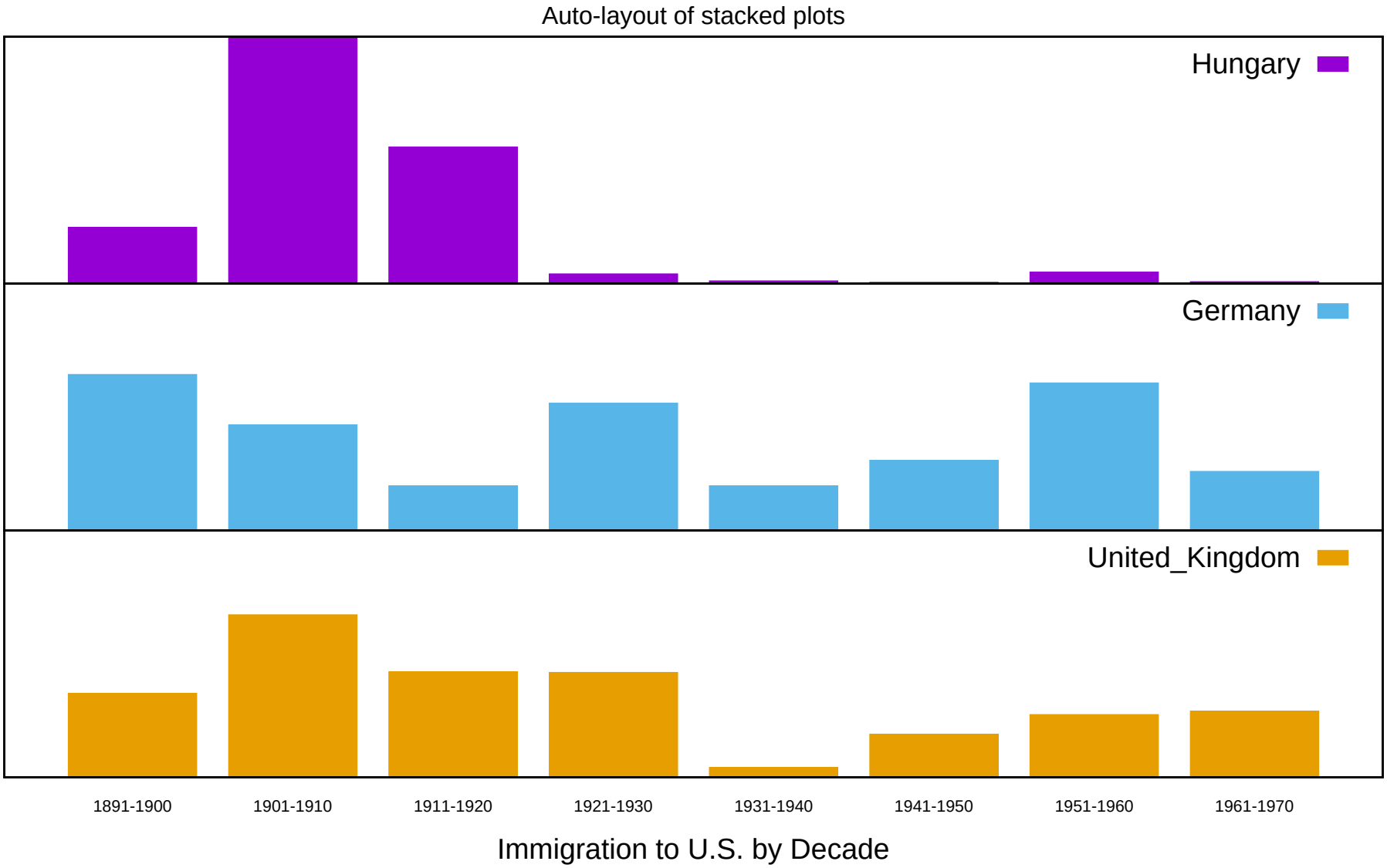

Second Order System Transfer Function - Phase
Phase (deg)
0
-30
-60
-90
-120
-150
-180
0.1
1
10
Frequency (w/wn)

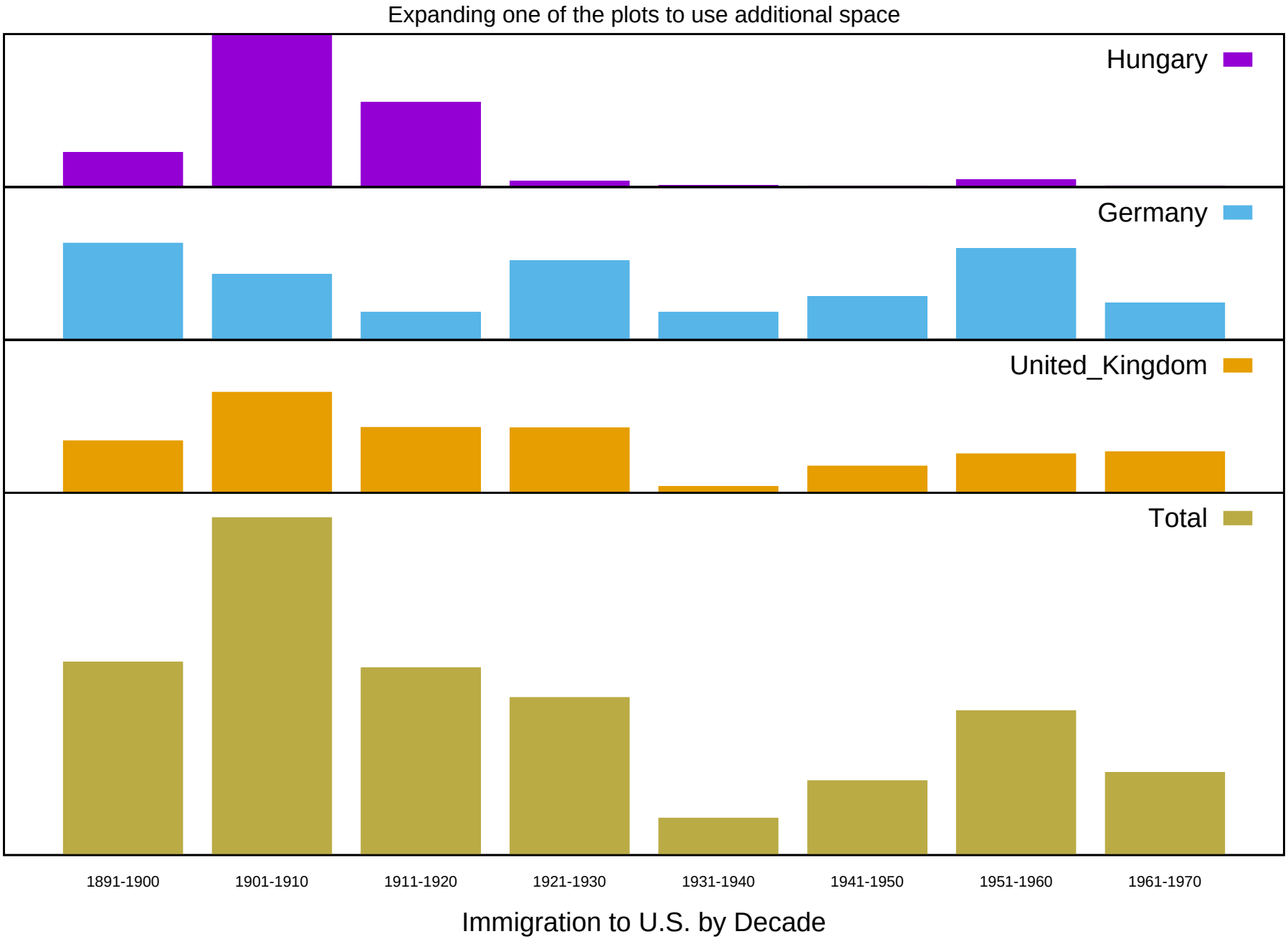

Second Order System - Unit Impulse Response
Amplitude y(wnt)
1
0.5
0
-0.5
-1
0
5
10
15
20
Normalized Time (wnt)

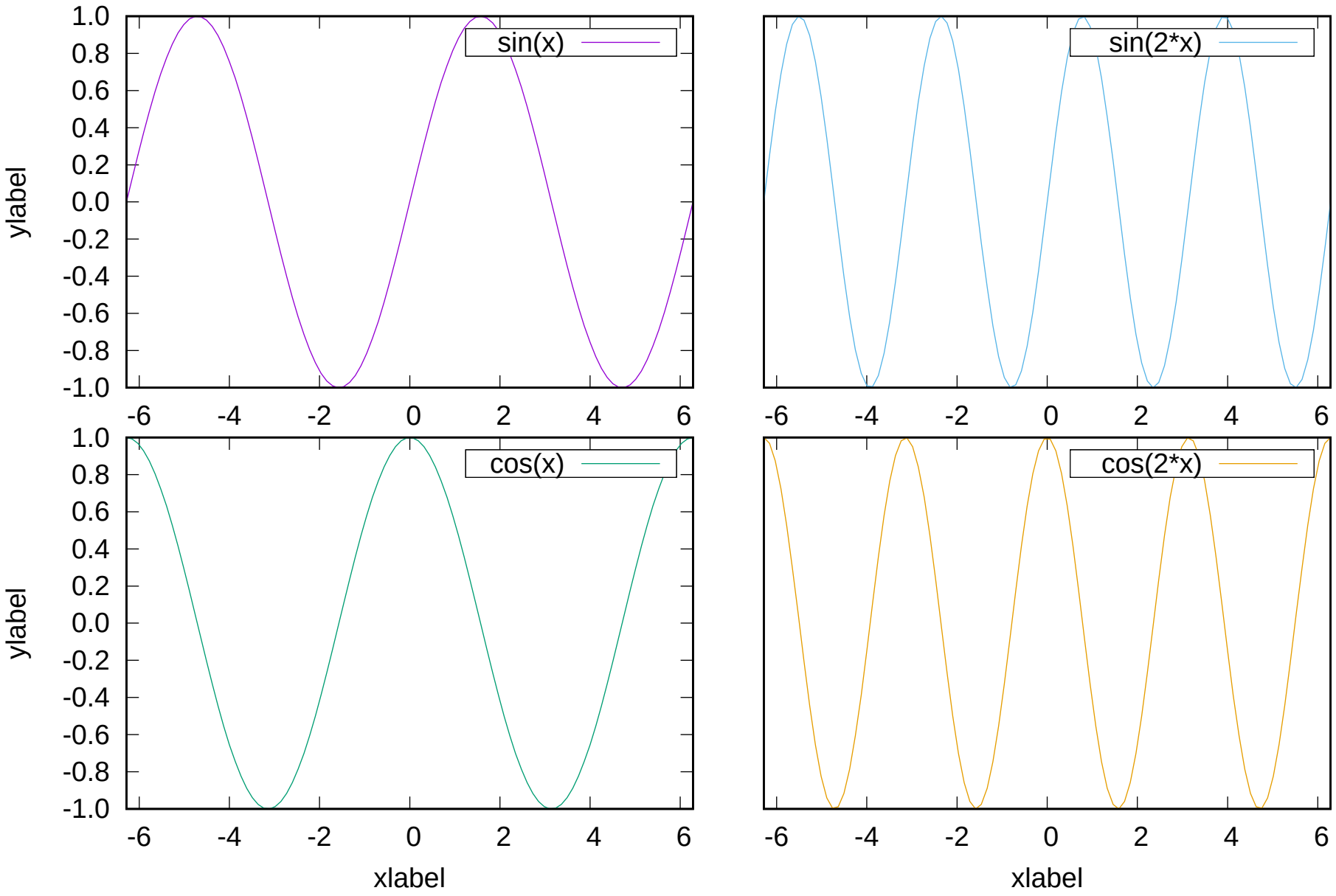

Multiplot layout 3, 1
Plot 1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10
Plot 2
300
250
200
150
100
50
0
0
100
200
300
400
500
600
Plot 3
Austria
France
Germany
Italy


Multiplot layout 1, 3
Plot 1
Plot 2
Plot 3
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10
300
250
200
150
100
50
0
0
100
200
300
400
500
600
Austria
France
Germany
Italy

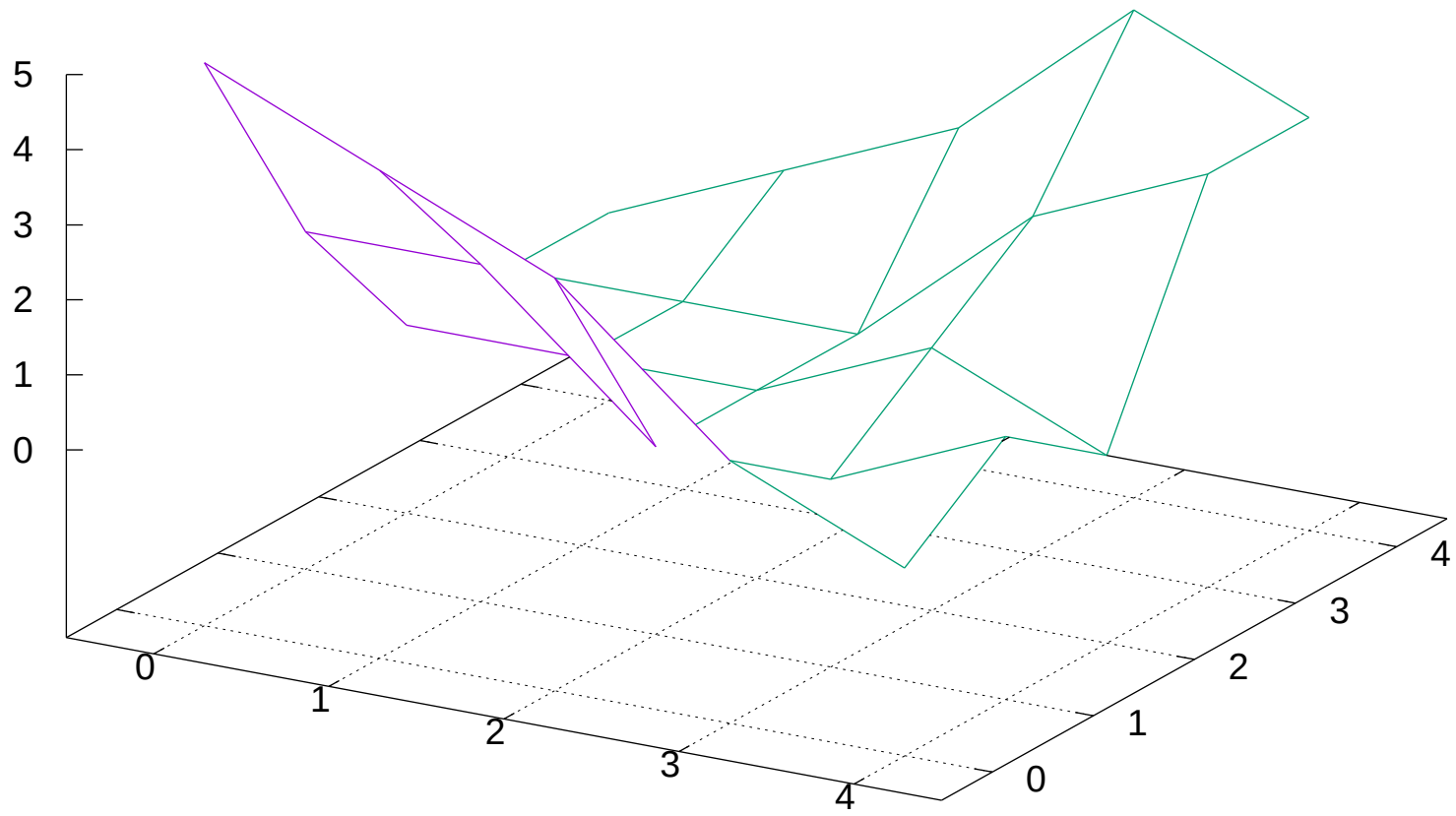

Same plot with a multi-line title
showing adjustment of plot area
to accommodate it
Plot 1
Plot 2
Plot 3
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10
300
250
200
150
100
50
0
0
100
200
300
400
500
600
Austria
France
Germany
Italy

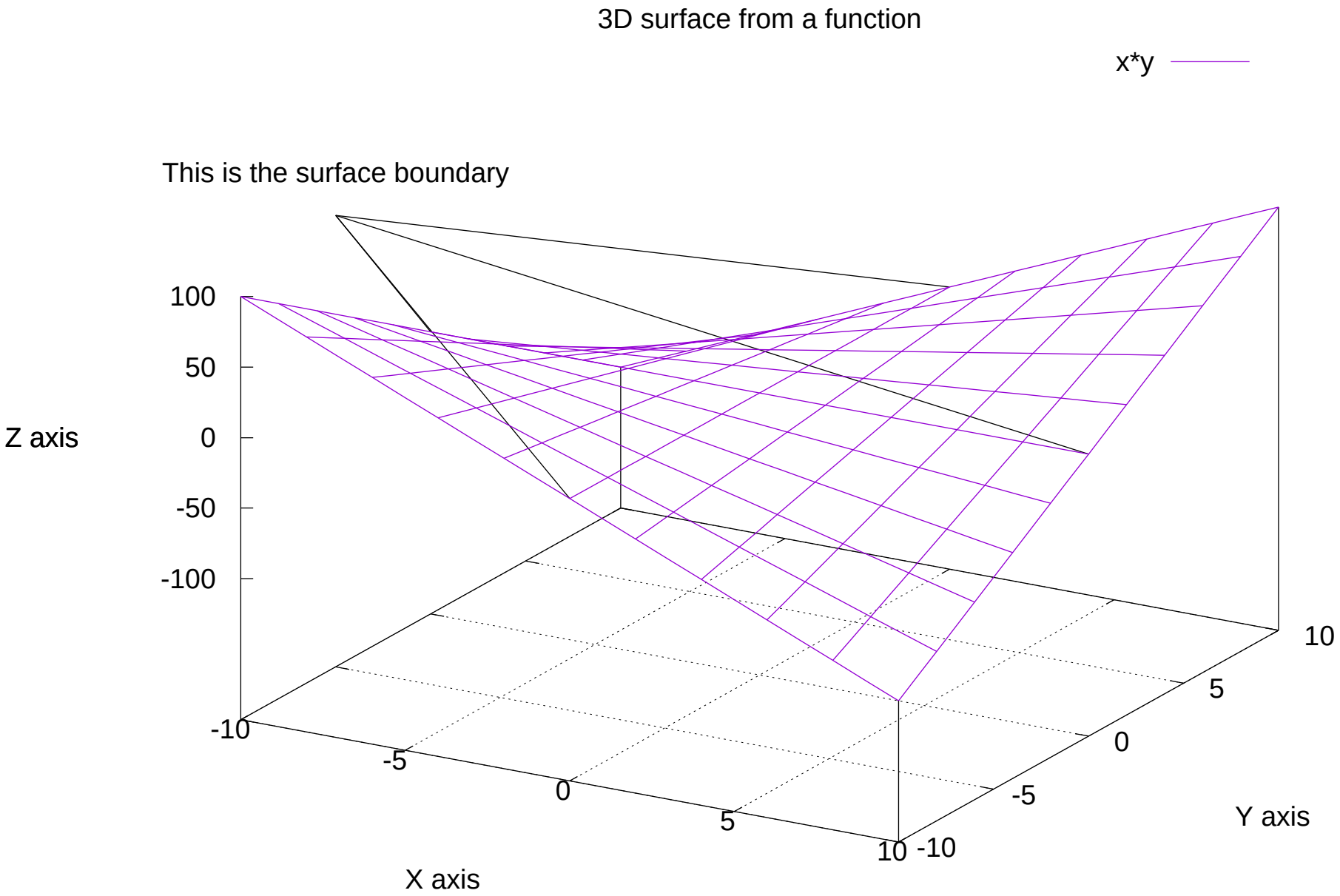

Auto-layout of stacked plots
Hungary
Germany
United_Kingdom
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Immigration to U.S. by Decade

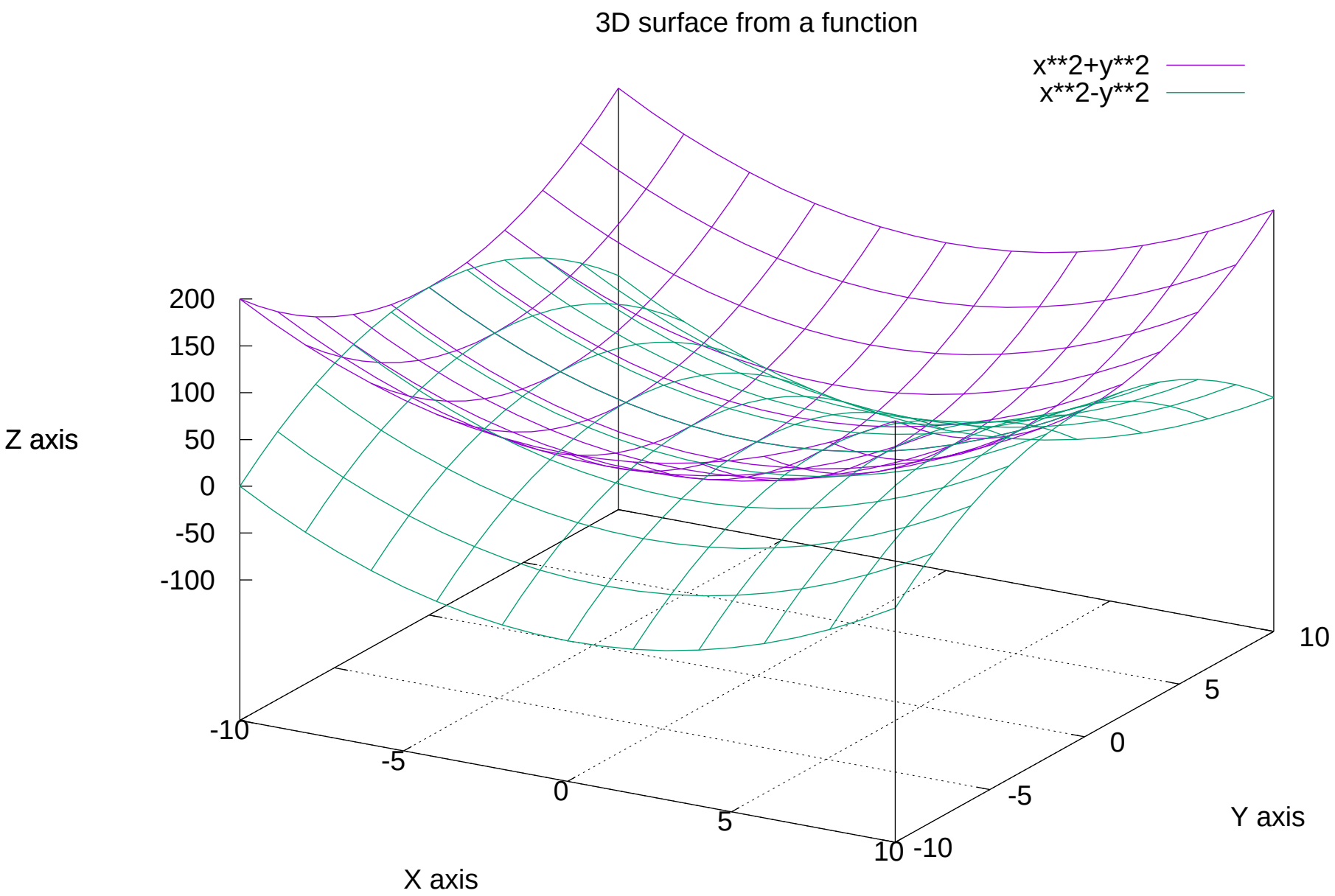

Expanding one of the plots to use additional space
Hungary
Germany
United_Kingdom
Total
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Immigration to U.S. by Decade

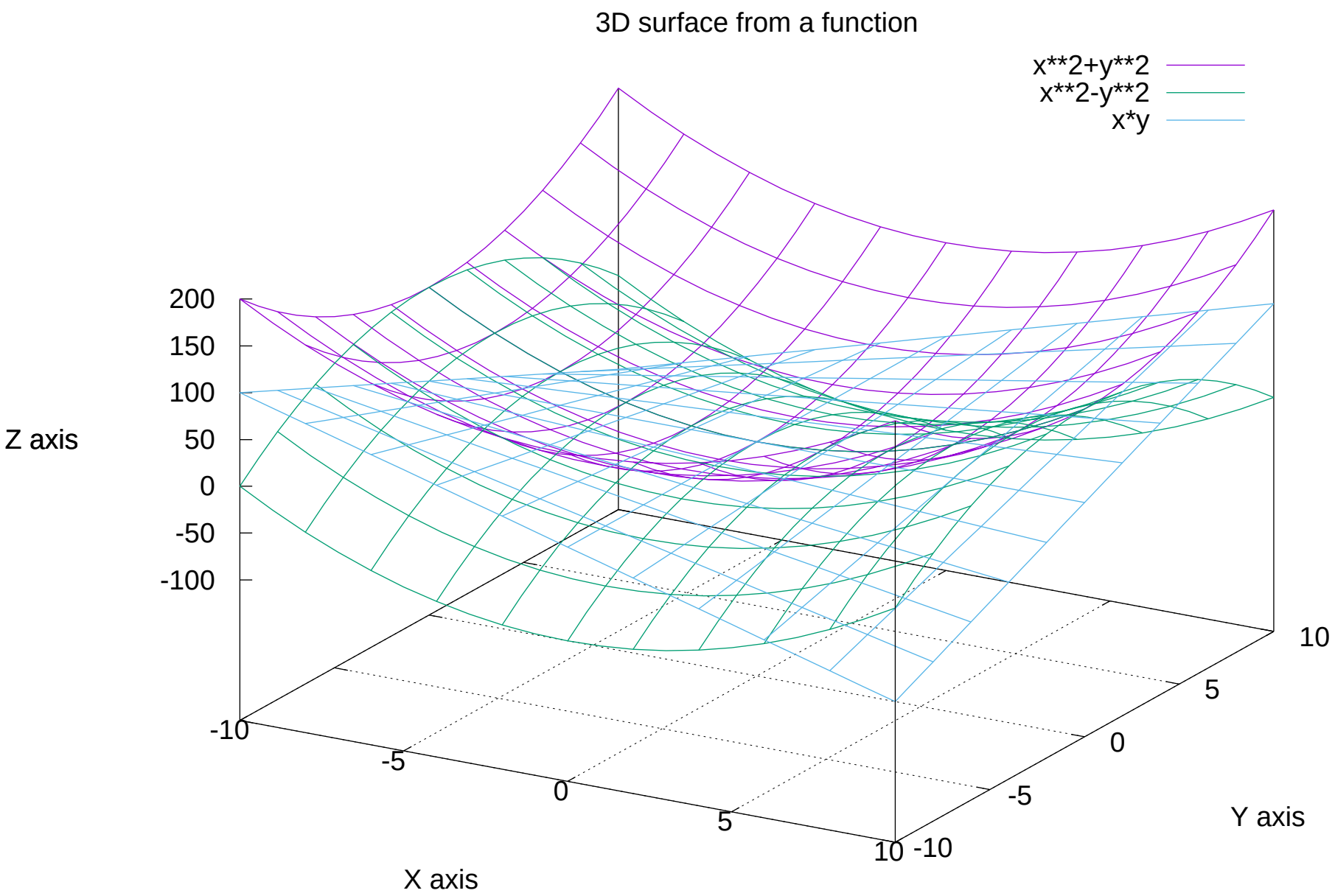

Multiplot with explicit page margins
sin(x)
sin(2*x)
cos(x)
cos(2*x)
ylabel
ylabel
xlabel
xlabel
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-6
-4
-2
0
2
4
6

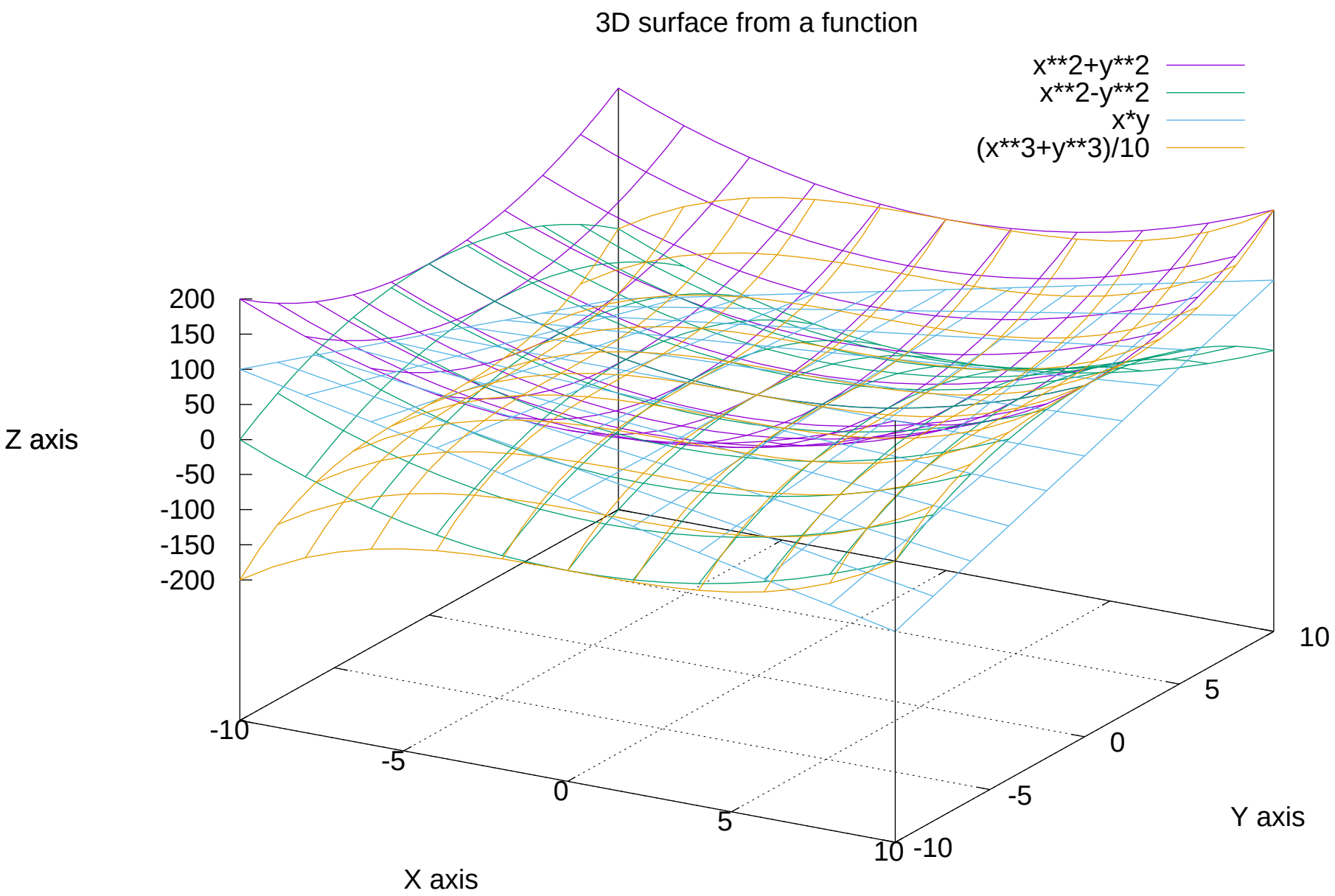
3D surface from a grid (matrix) of Z values

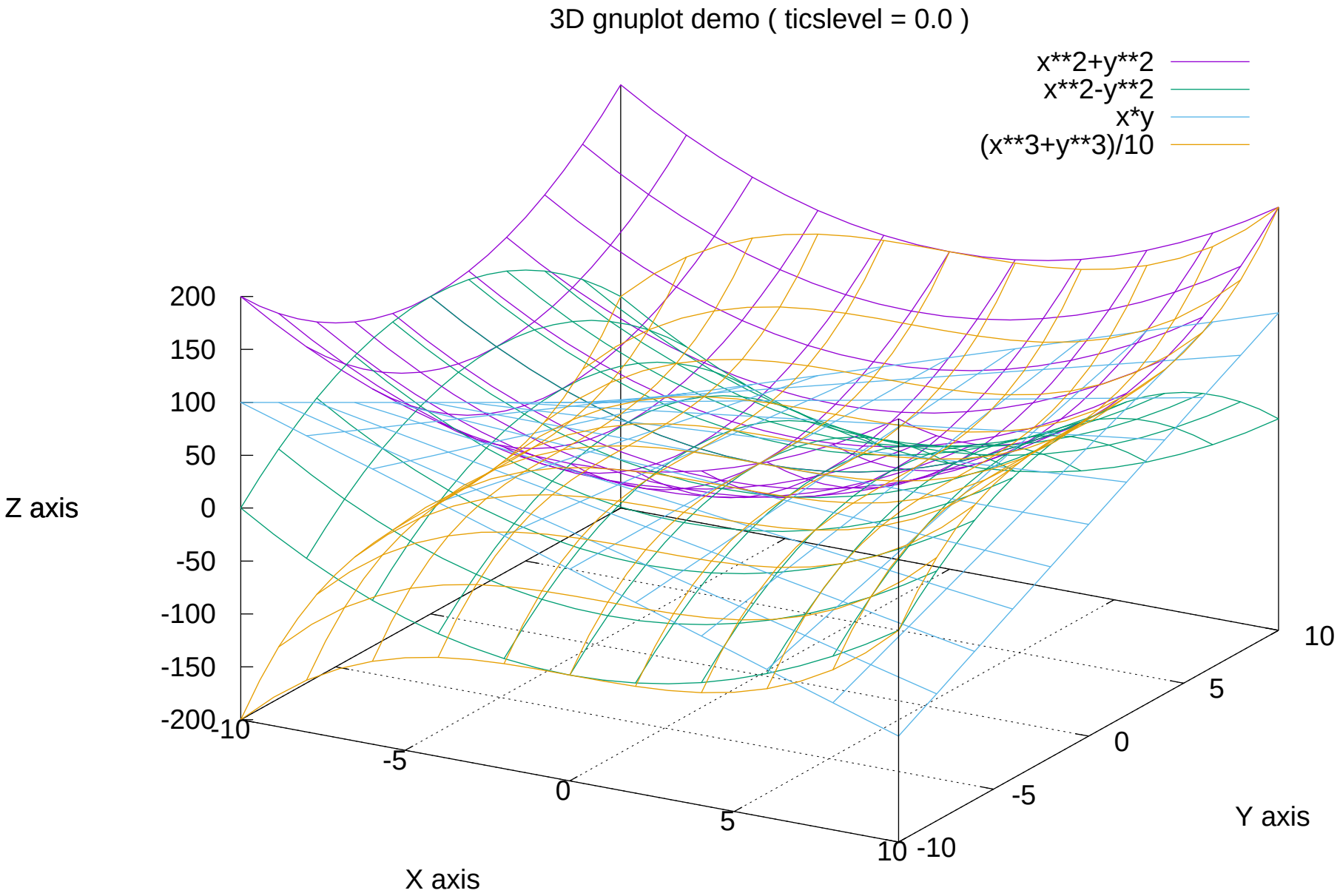


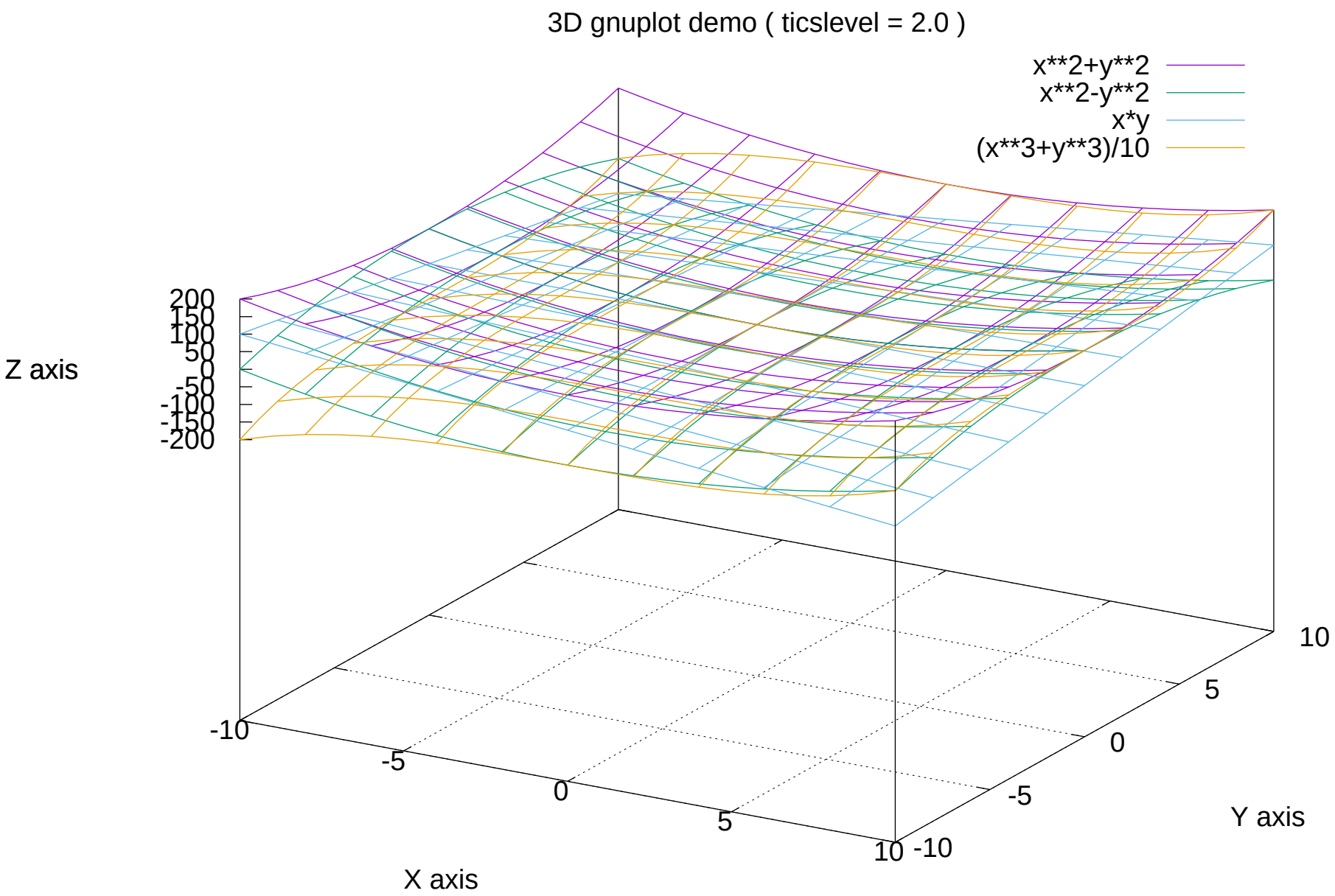

3D surface from a function
x*y
This is the surface boundary
Z axis
100
50
0
-50
-100
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

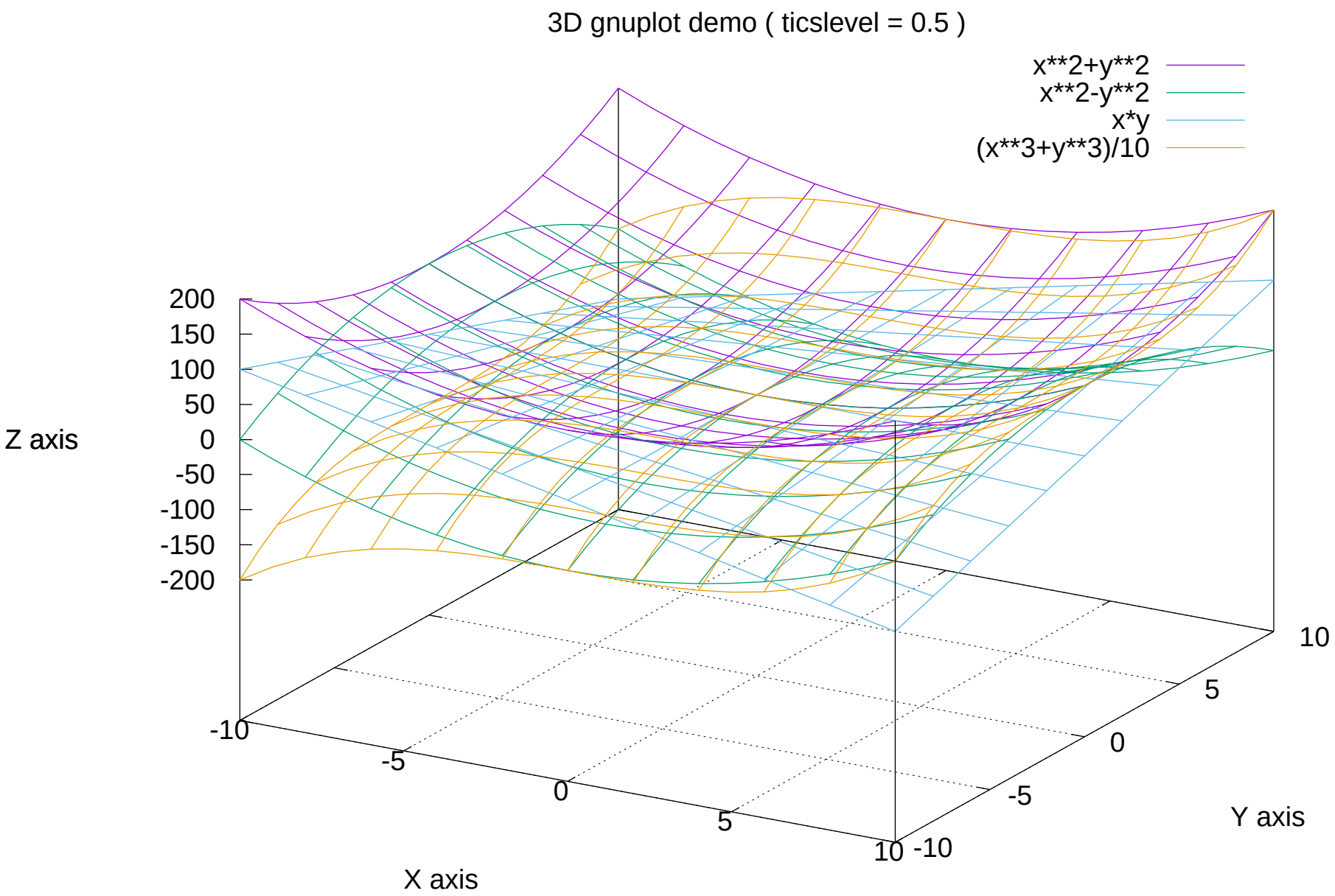

3D surface from a function
x**2+y**2
x**2-y**2
Z axis
200
150
100
50
0
-50
-100
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

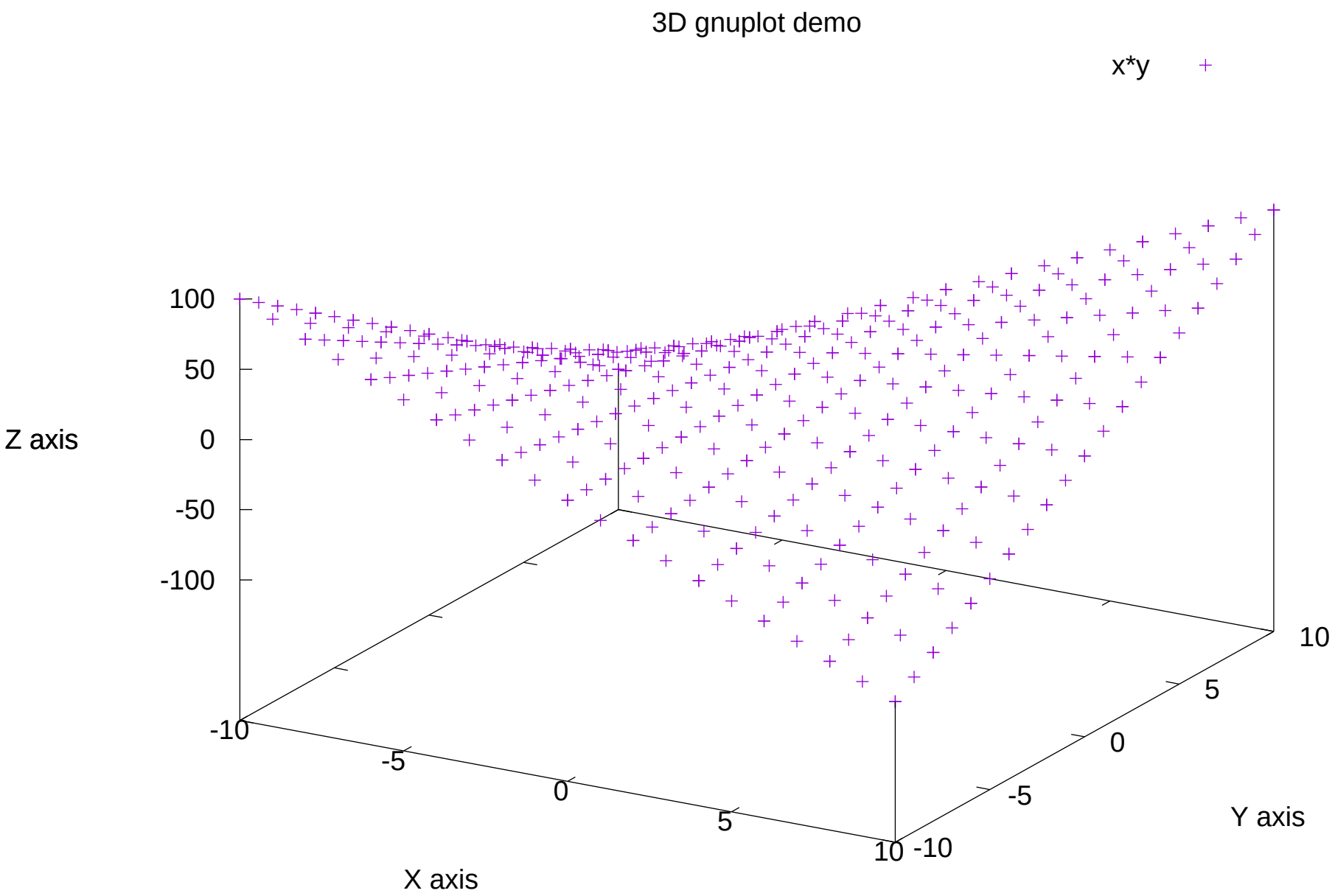

3D surface from a function
x**2+y**2
x**2-y**2
x*y
Z axis
200
150
100
50
0
-50
-100
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

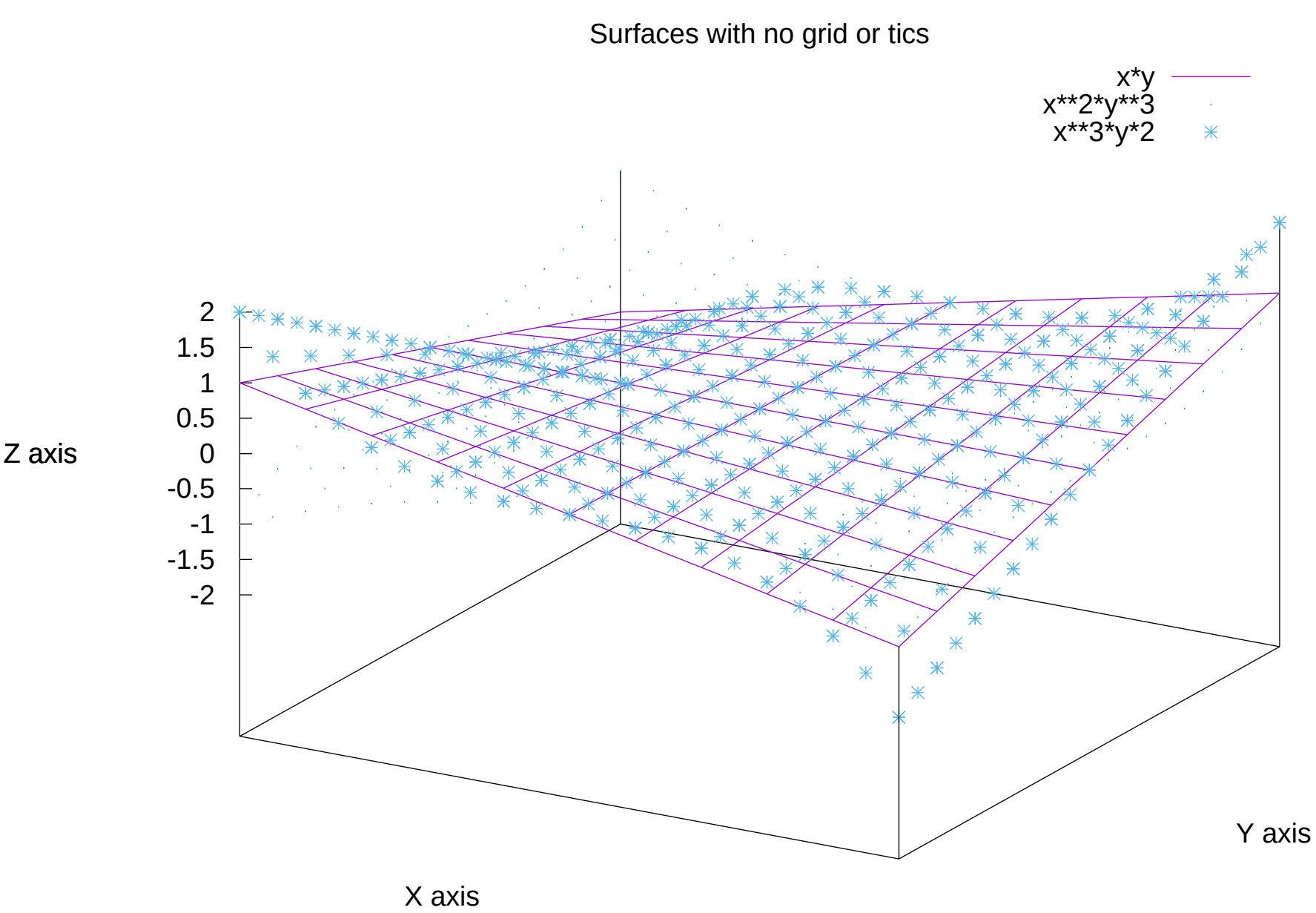

3D surface from a function
x**2+y**2
x**2-y**2
x*y
(x**3+y**3)/10
Z axis
200
150
100
50
0
-50
-100
-150
-200
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

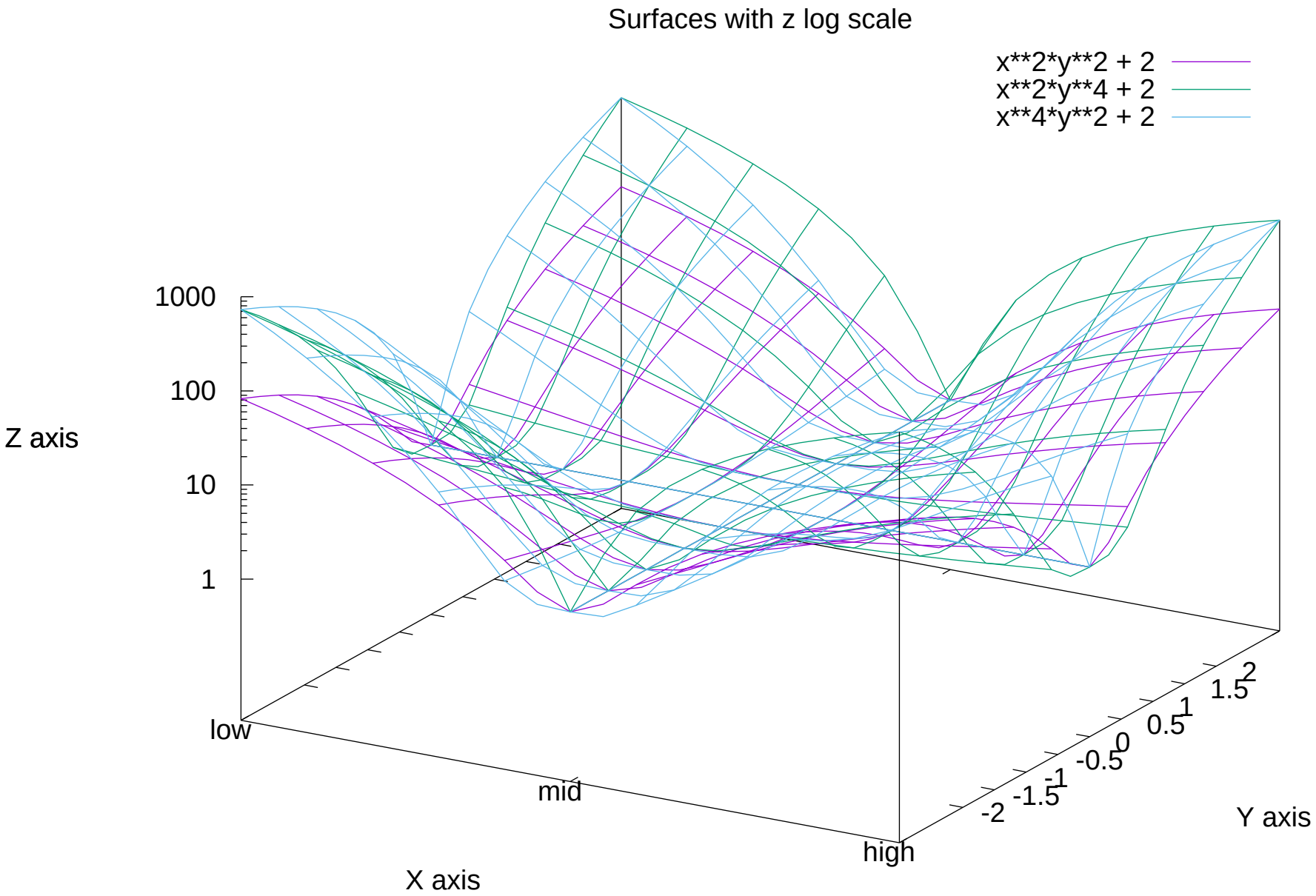

3D gnuplot demo (ticslevel = 0.0)
x**2+y**2
x**2-y**2
x*y
(x**3+y**3)/10
Z axis
200
150
100
50
0
-50
-100
-150
-200
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

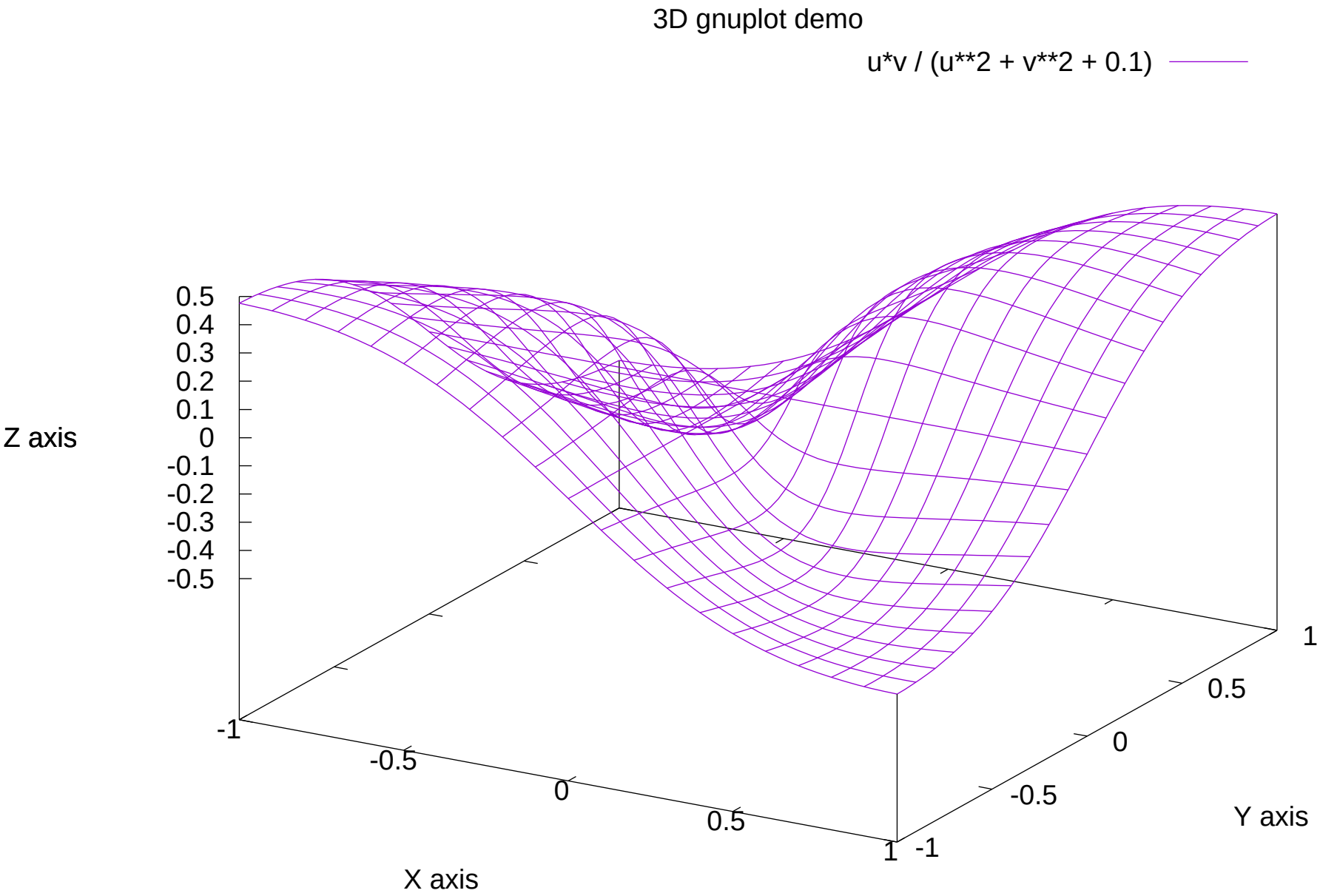

3D gnuplot demo (ticslevel = 2.0)
x**2+y**2
x**2-y**2
x*y
(x**3+y**3)/10
Z axis
200
150
100
50
0
-50
-100
-150
-200
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

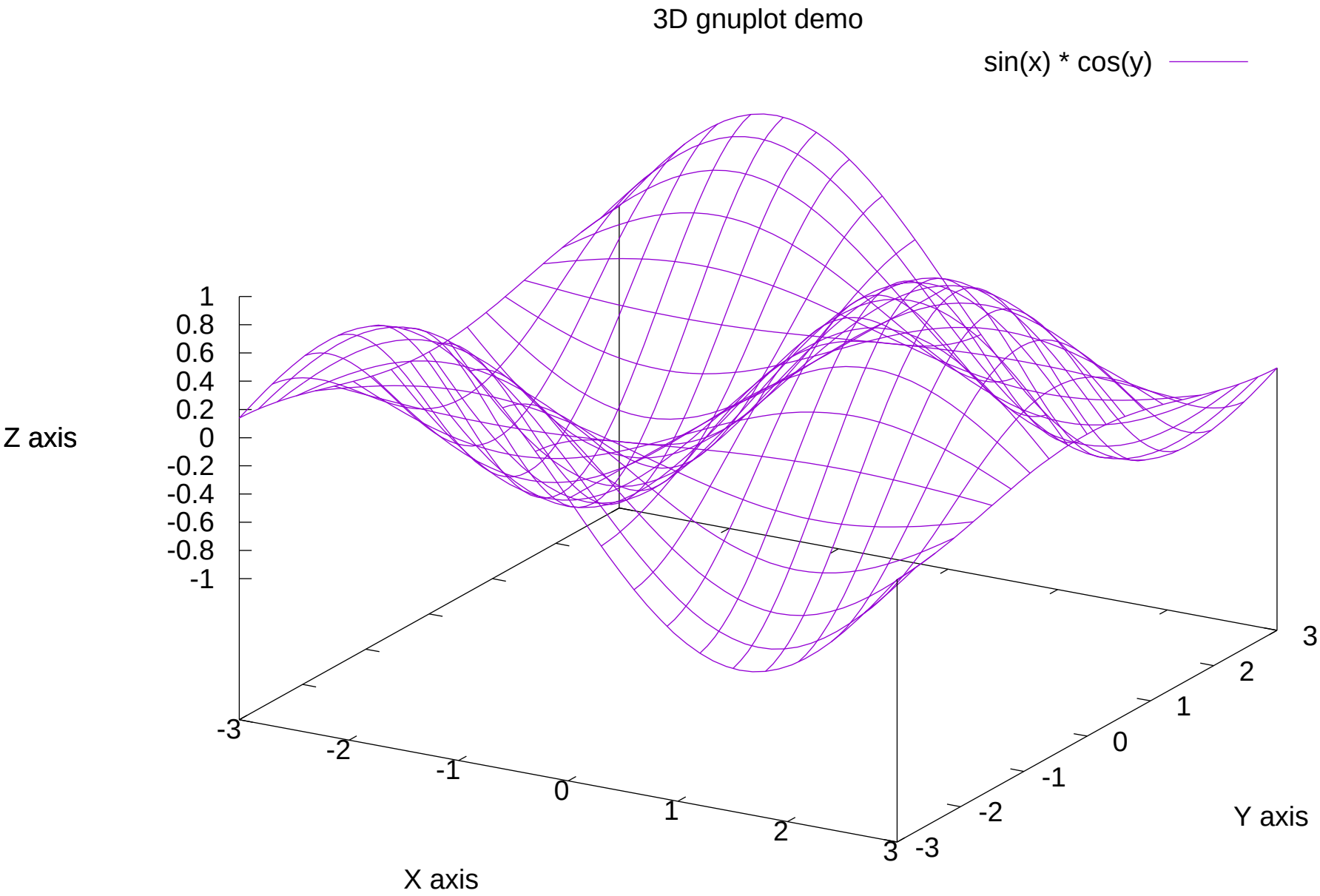

3D gnuplot demo (ticslevel = 0.5)
x**2+y**2
x**2-y**2
x*y
(x**3+y**3)/10
Z axis
200
150
100
50
0
-50
-100
-150
-200
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

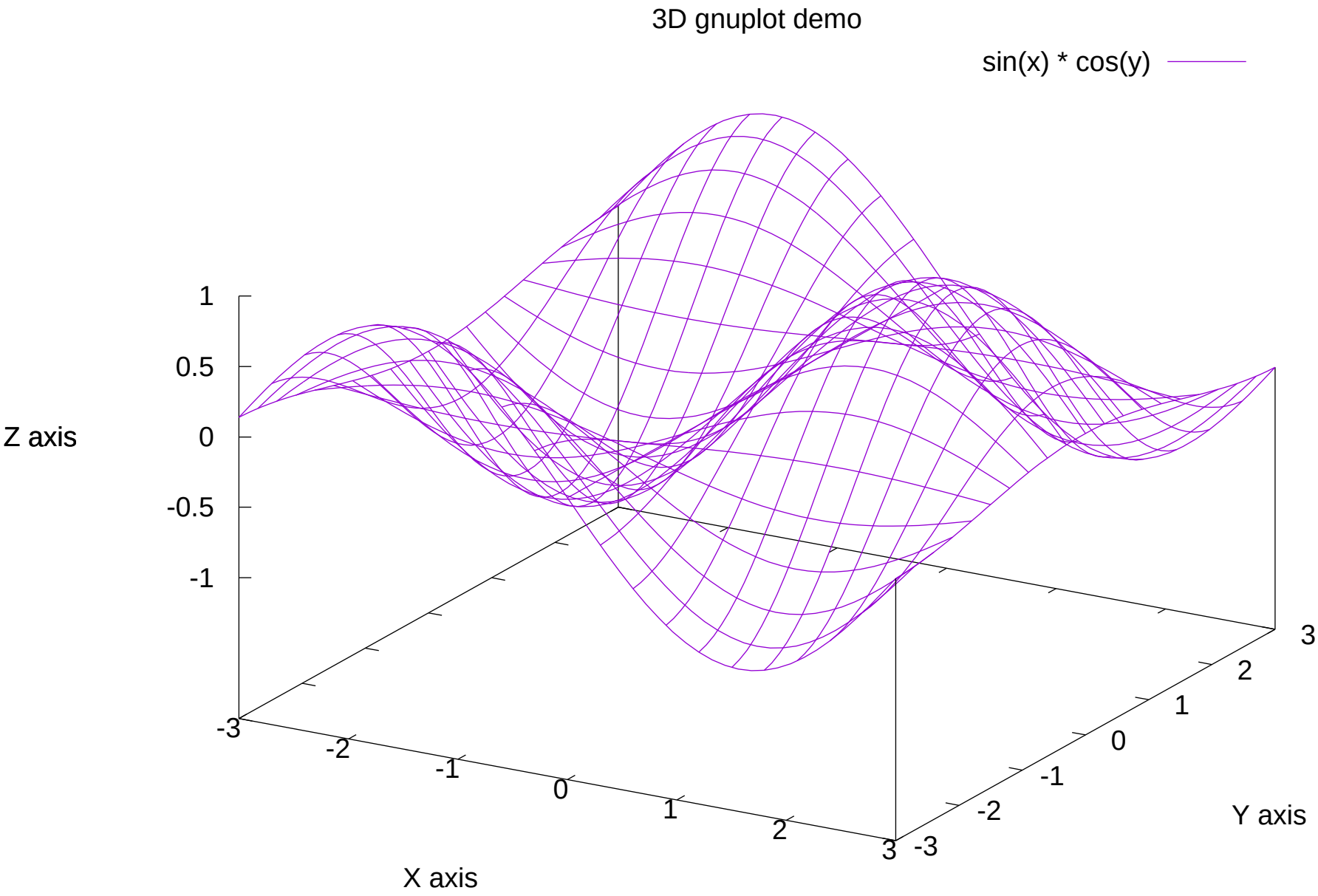

3D gnuplot demo
x*y
Z axis
100
50
0
-50
-100
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

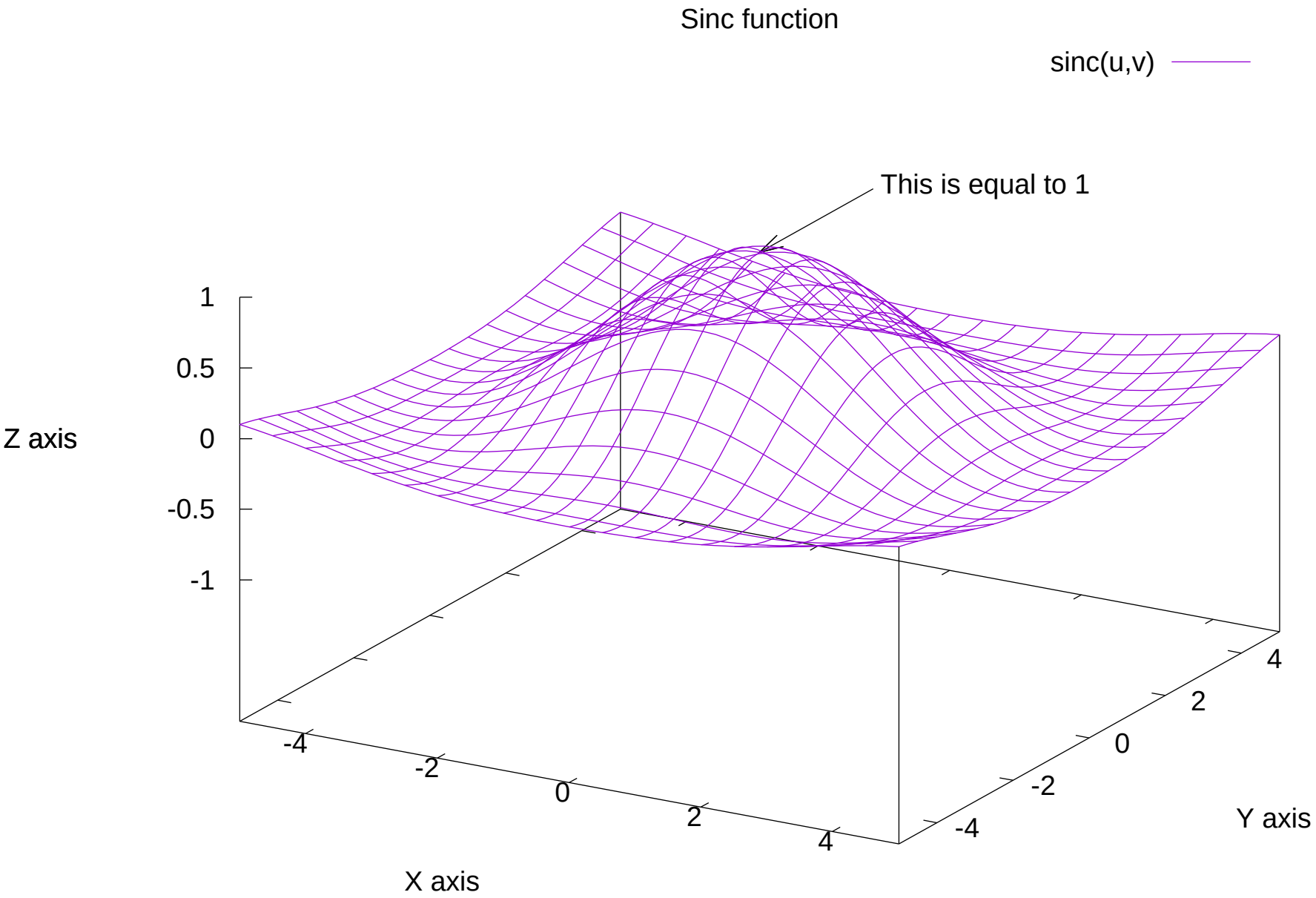

Surfaces with no grid or tics
x*y
x**2*y**3
x**3*y*2
Z axis
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
X axis
Y axis

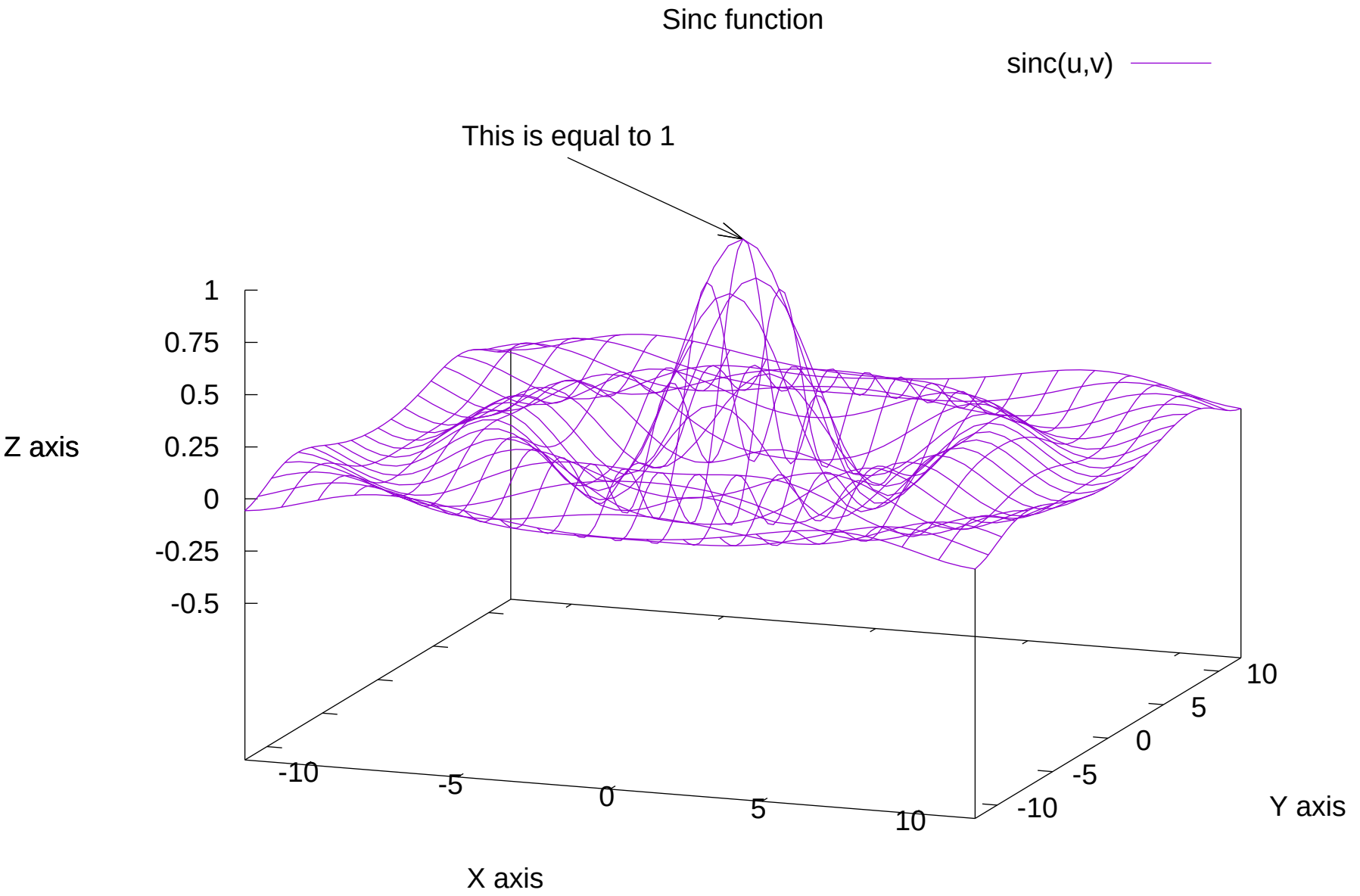

Surfaces with z log scale
x**2*y**2 + 2
x**2*y**4 + 2
x**4*y**2 + 2
Z axis
1000
100
10
1
low
mid
high
X axis
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
Y axis

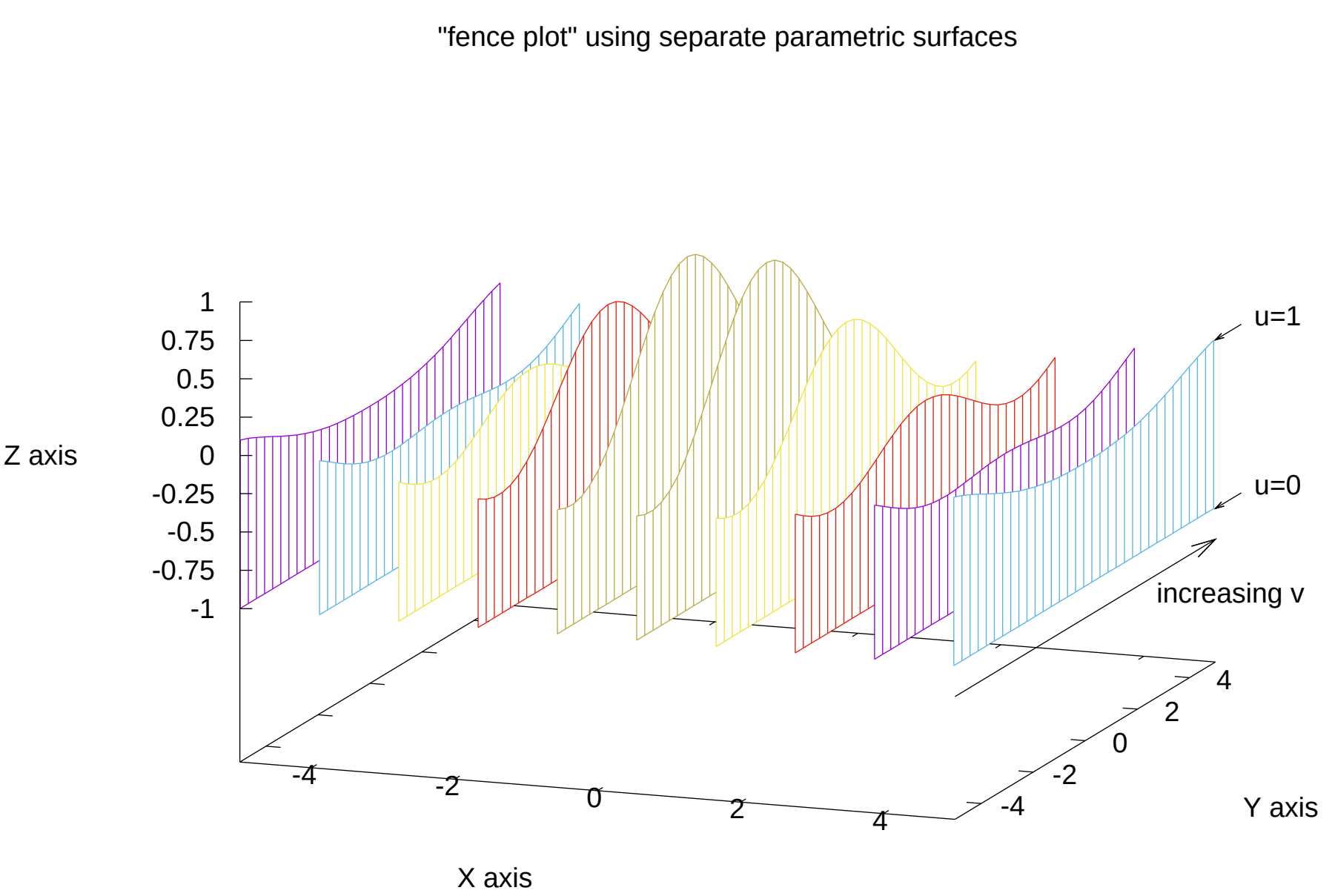

3D gnuplot demo
u*v / (u**2 + v**2 + 0.1)
Z axis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1
-0.5
0
0.5
1
X axis
-1
-0.5
0
0.5
1
Y axis


3D gnuplot demo
sin(x) * cos(y)
Z axis
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-3
-2
-1
0
1
2
3
X axis
-3
-2
-1
0
1
2
3
Y axis

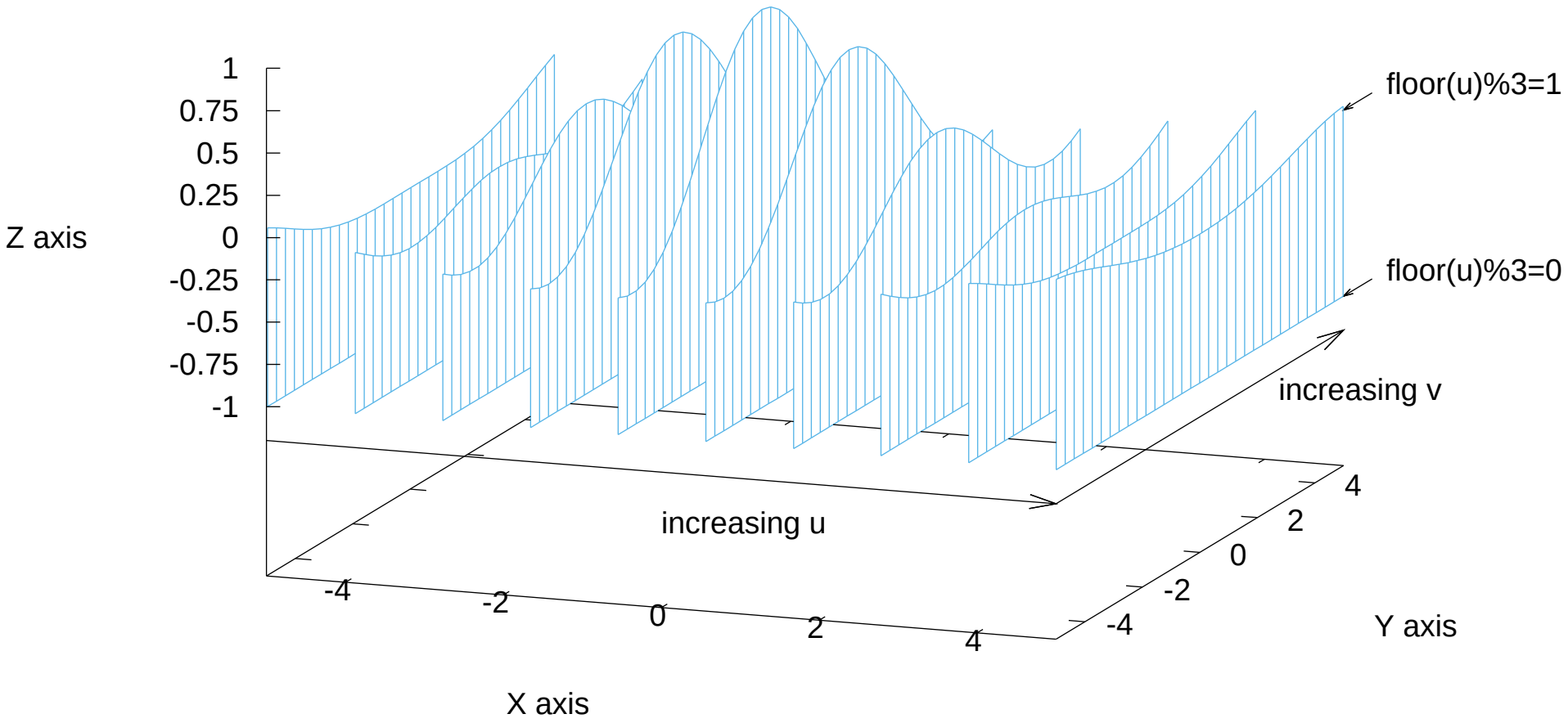

3D gnuplot demo
sin(x) * cos(y)
Z axis
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3
X axis
-3
-2
-1
0
1
2
3
Y axis

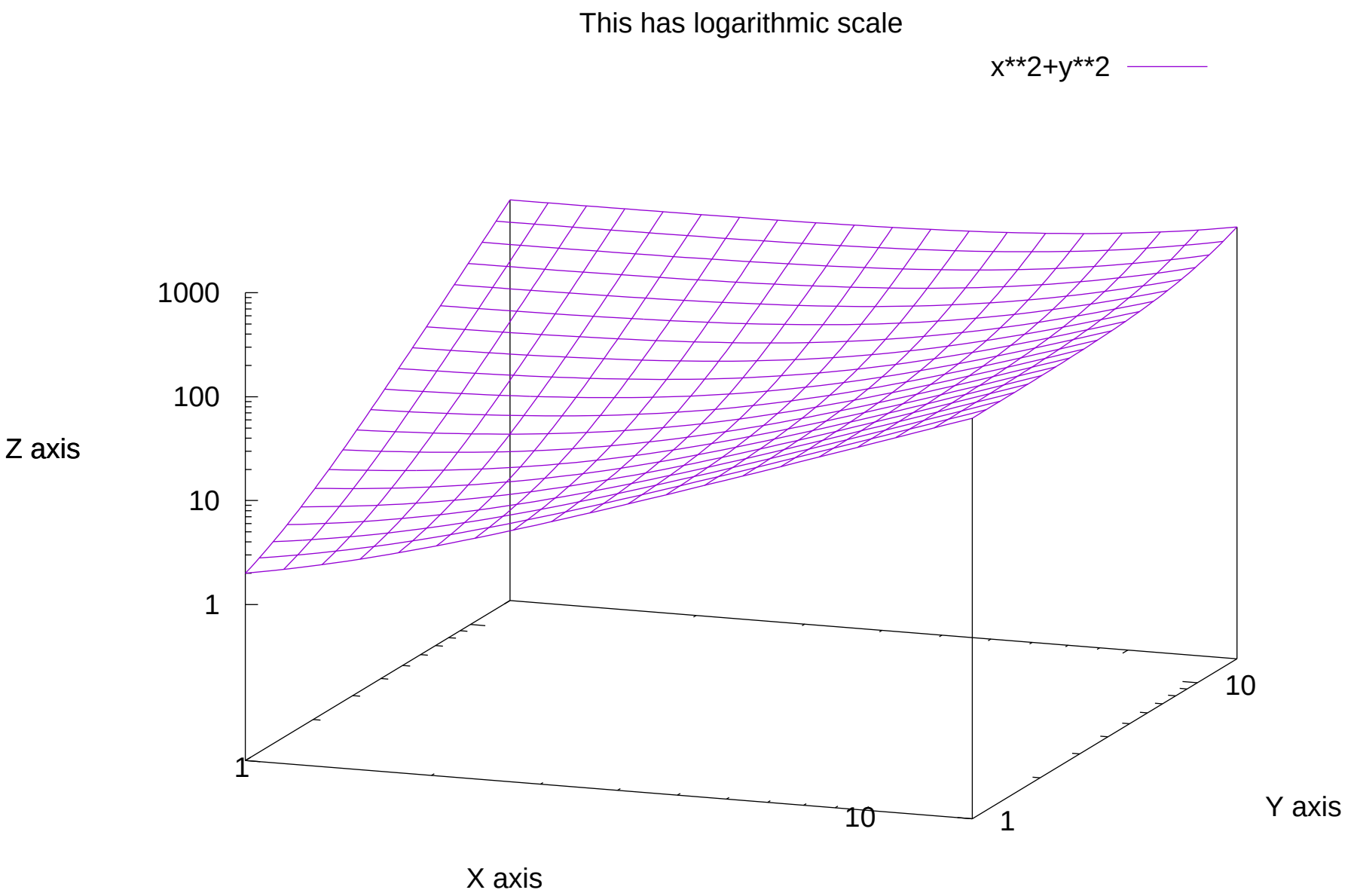

Sinc function
sinc(u,v)
This is equal to 1
Z axis
1
0.5
0
-0.5
-1
-4
-2
0
2
4
X axis
-4
-2
0
2
4
Y axis

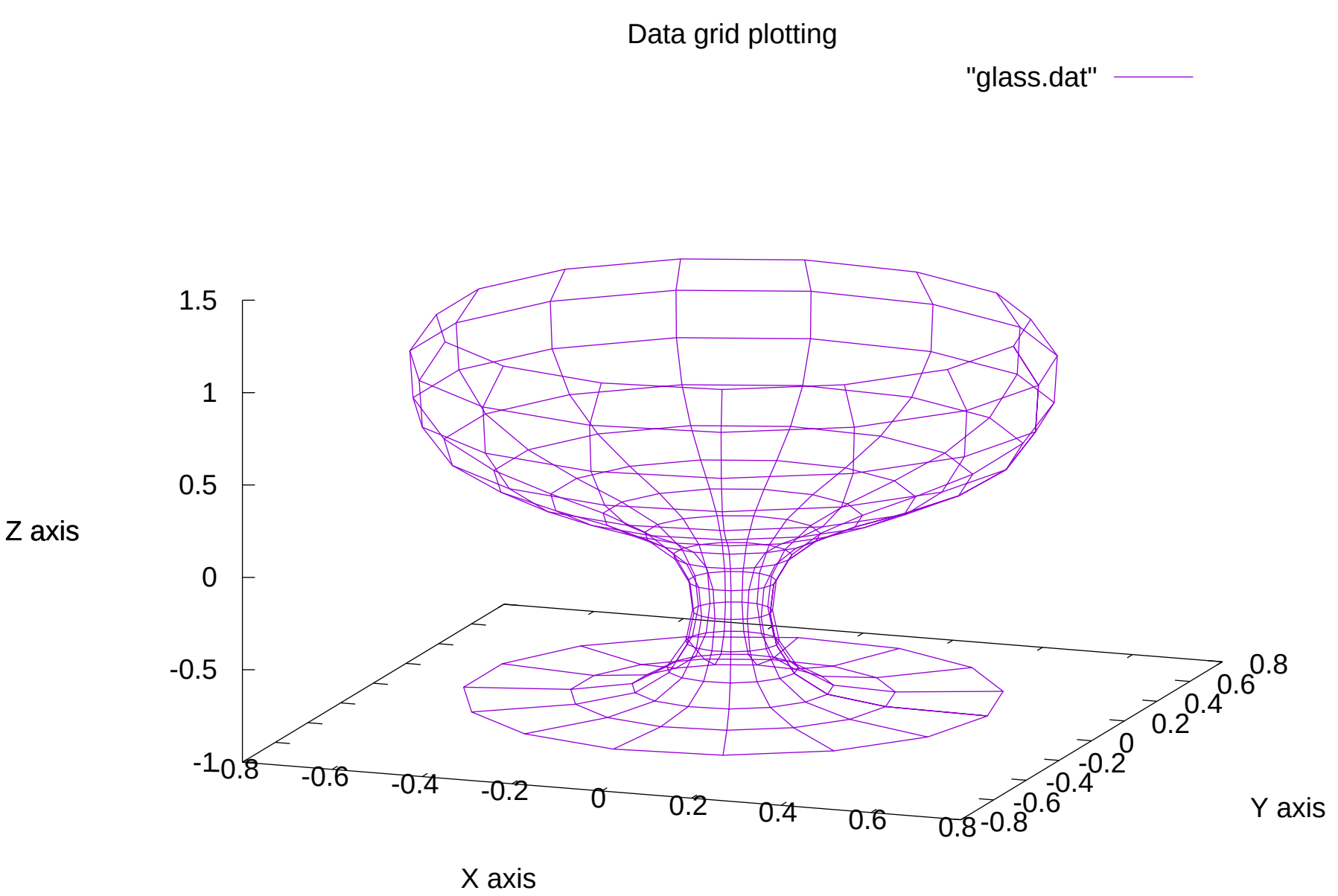

Sinc function
sinc(u,v)
This is equal to 1
1
0.75
0.5
0.25
0
-0.25
-0.5
Z axis
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

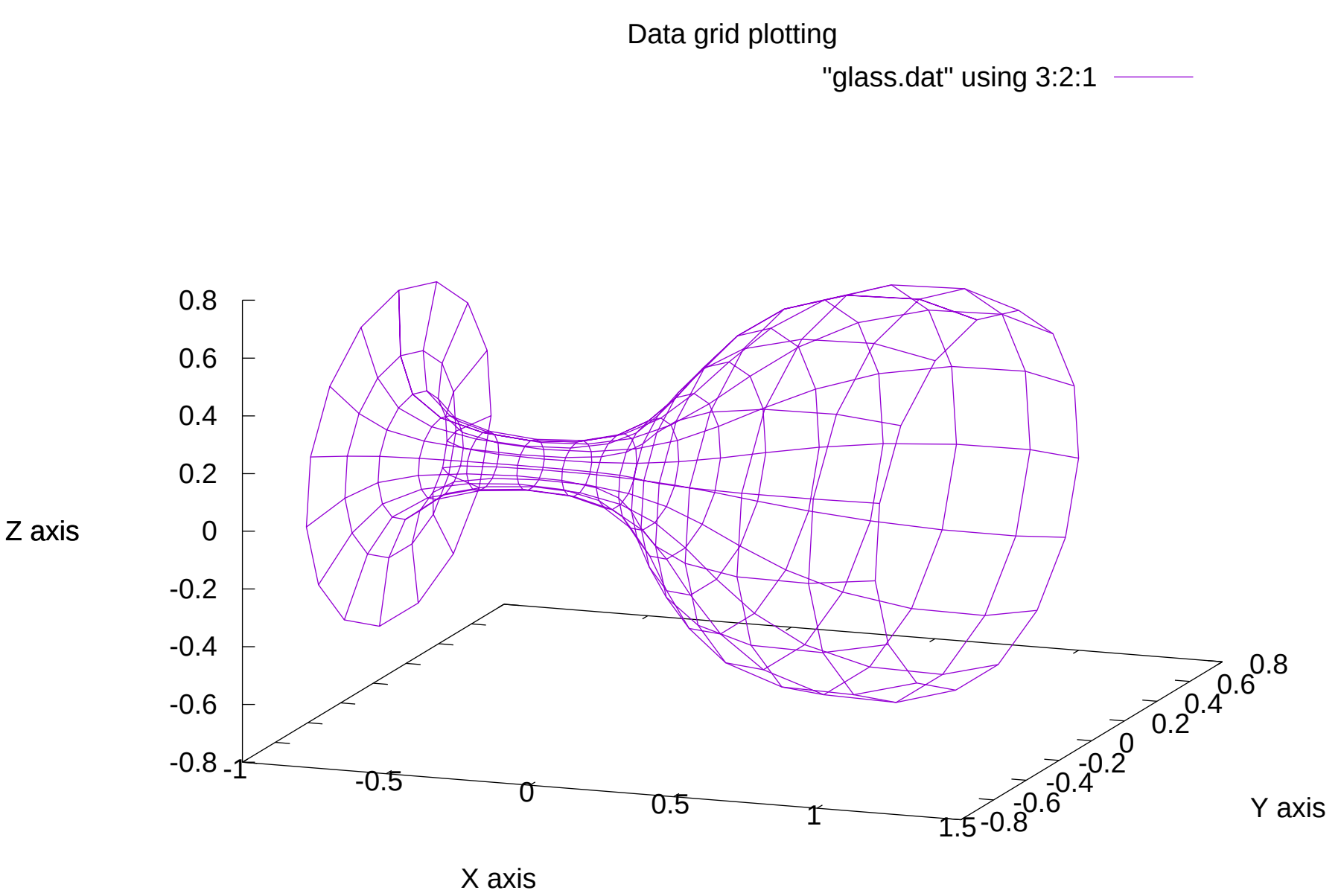

"fence plot" using separate parametric surfaces
Z axis
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1
u=1
u=0
increasing v
-4
-2
0
2
4
X axis
-4
-2
0
2
4
Y axis

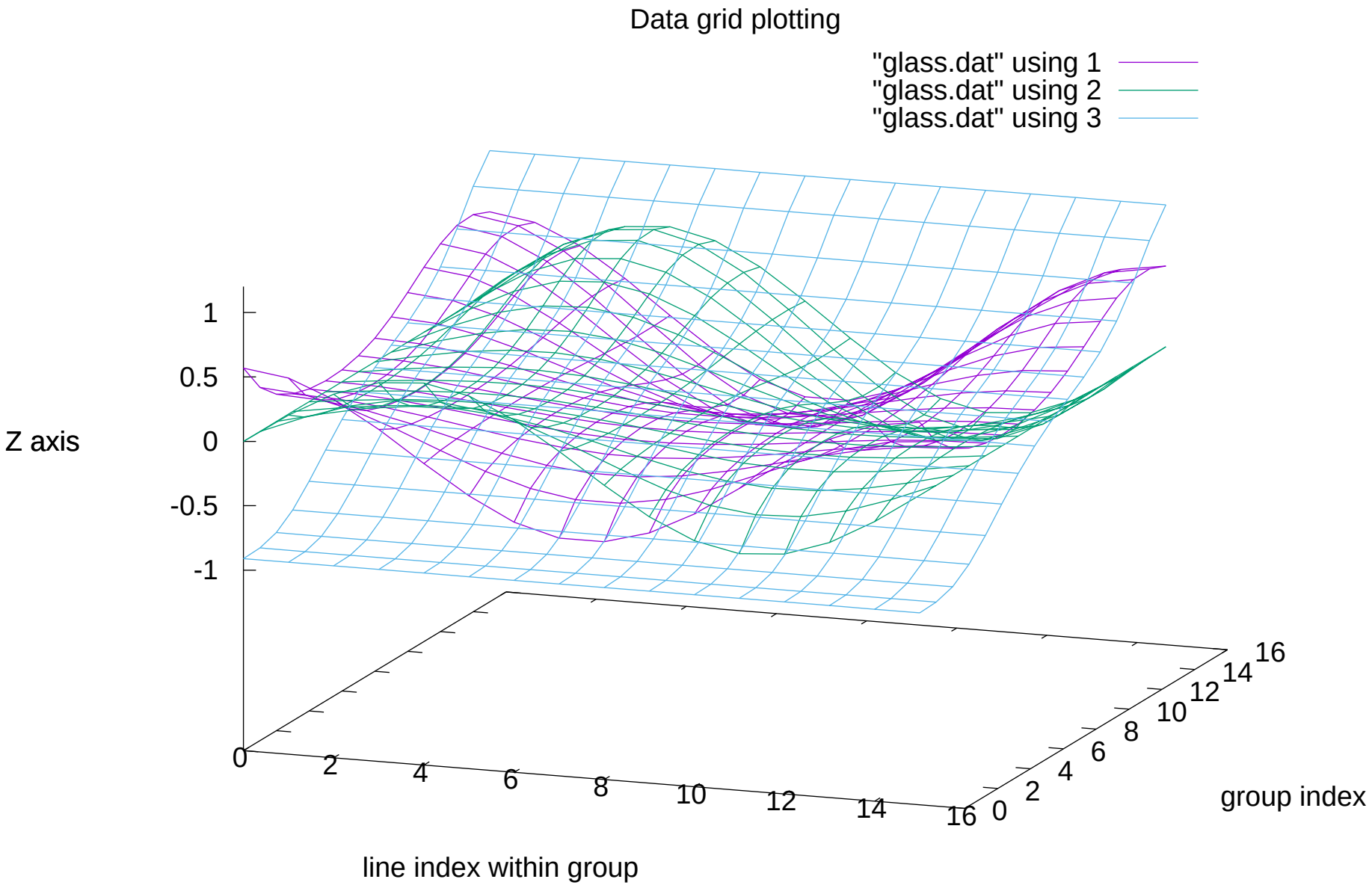
"fence plot" using single parametric surface with undefined points

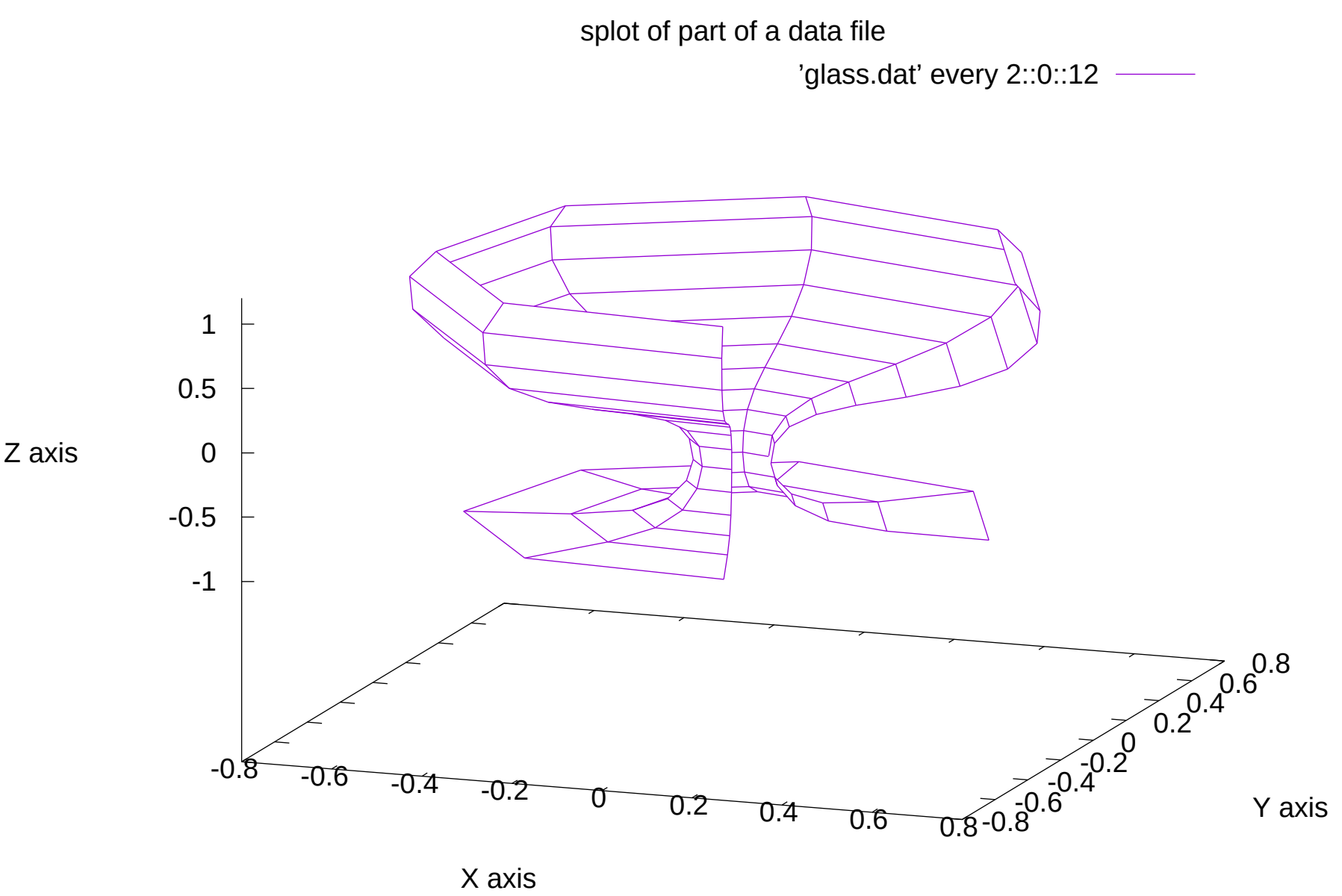


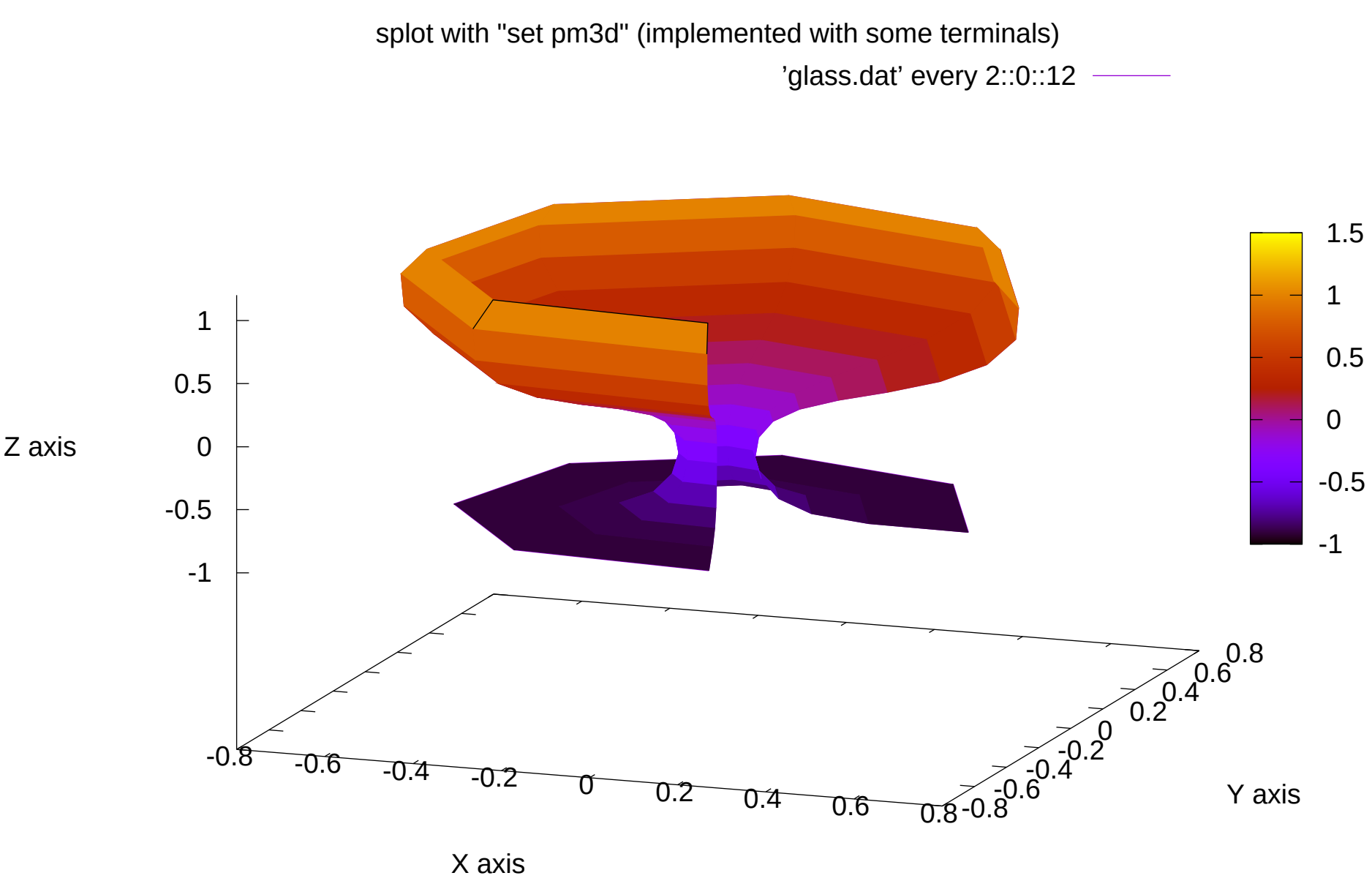

This has logarithmic scale
x**2+y**2
1000
100
10
1
Z axis
1
10
X axis
1
10
Y axis

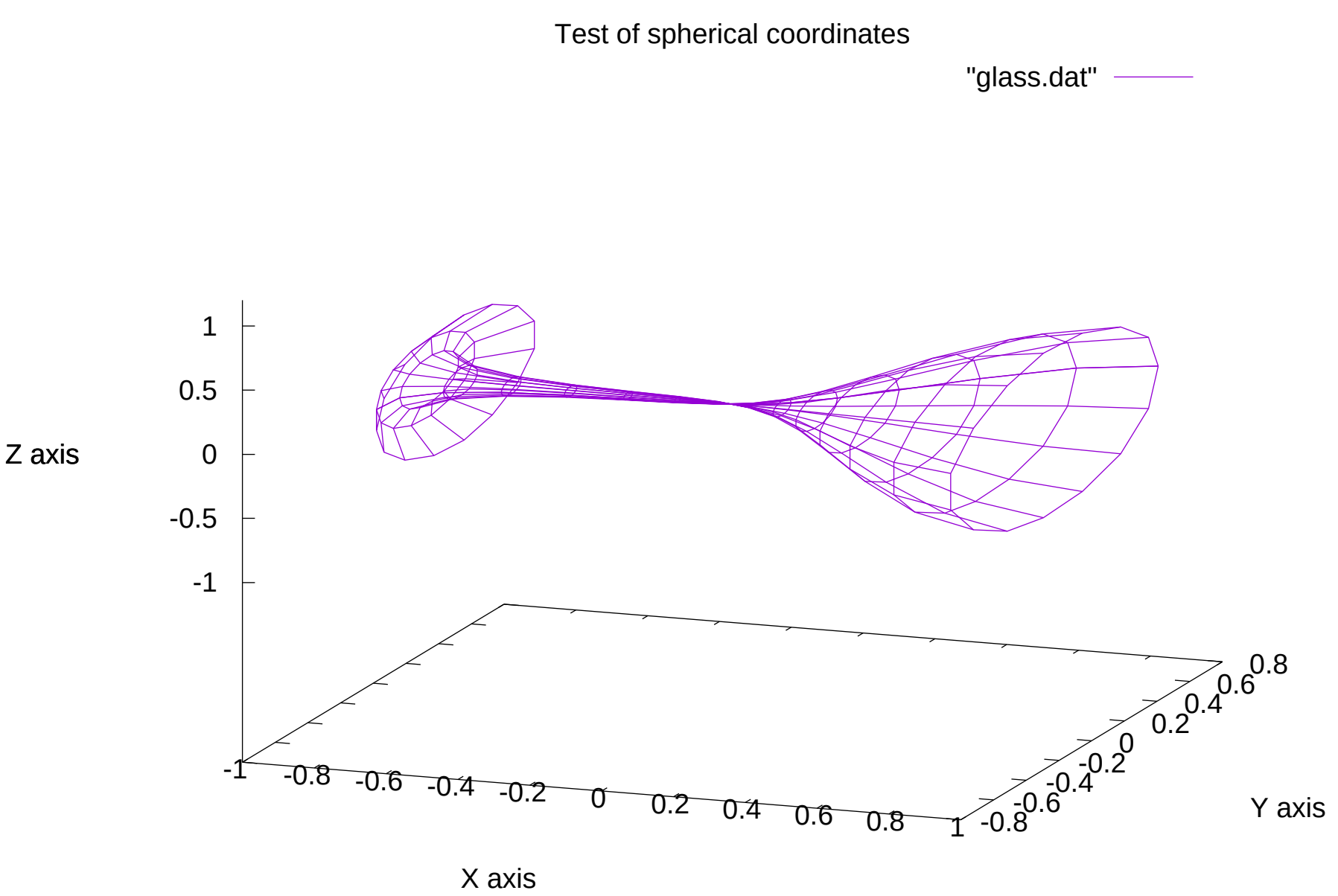

Data grid plotting
"glass.dat"
Z axis
1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
X axis
Y axis

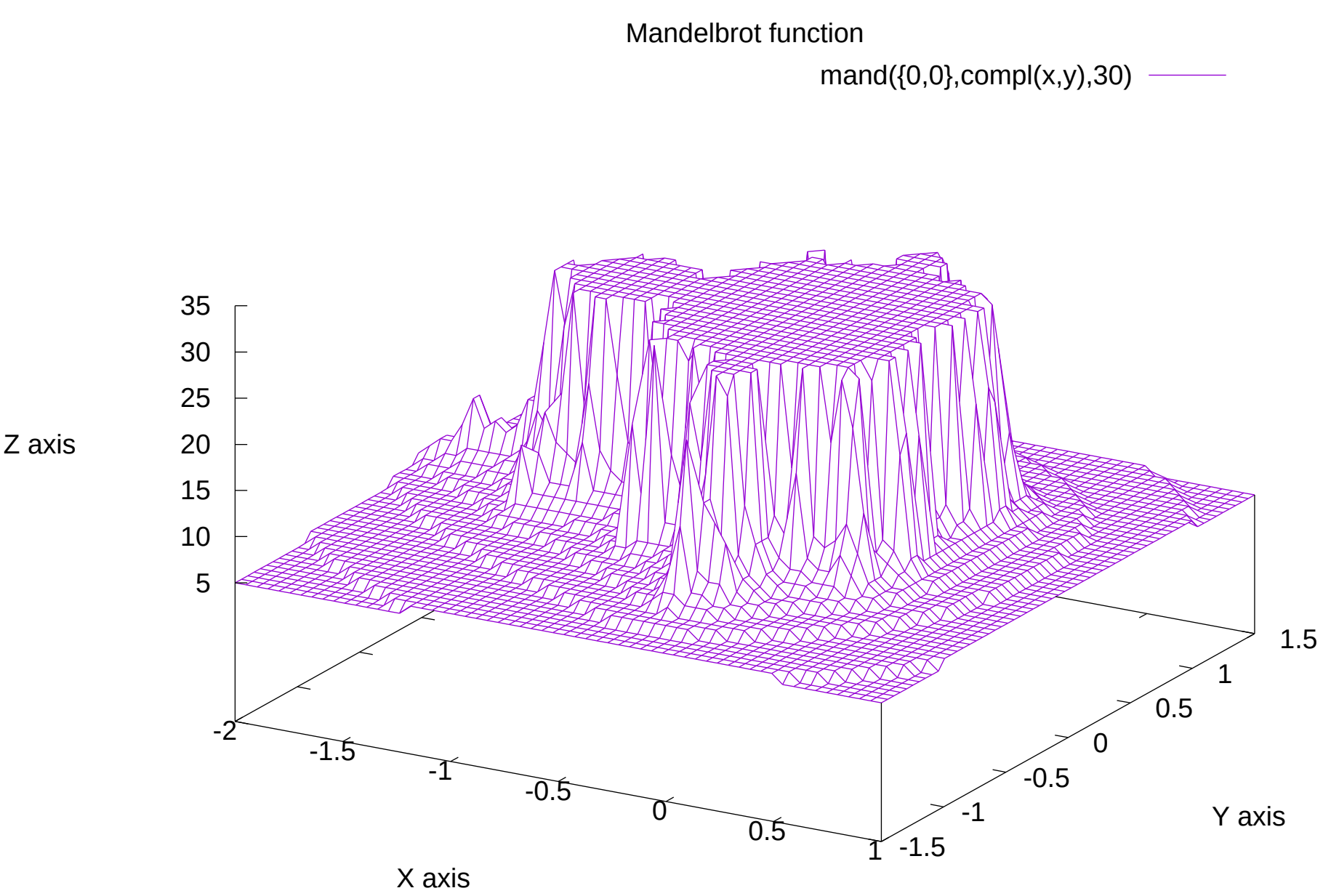

Data grid plotting
"glass.dat" using 3:2:1
Z axis
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0
0.5
1
1.5
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Y axis
X axis


Data grid plotting
"glass.dat" using 1
"glass.dat" using 2
"glass.dat" using 3
Z axis
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
16
line index within group
0
2
4
6
8
10
12
14
16
group index

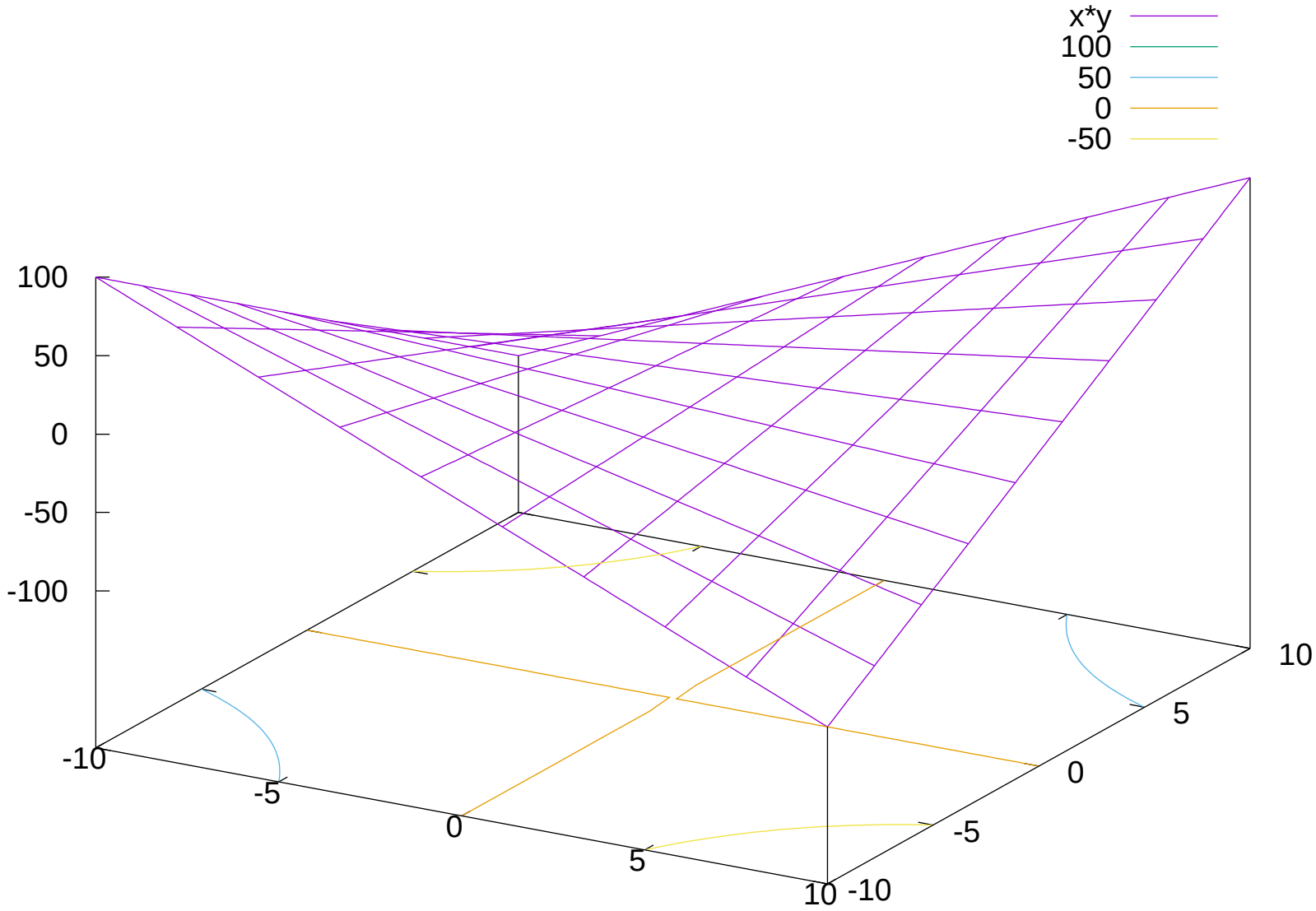

splot of part of a data file
'glass.dat' every 2::0::12
Z axis
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
X axis
Y axis

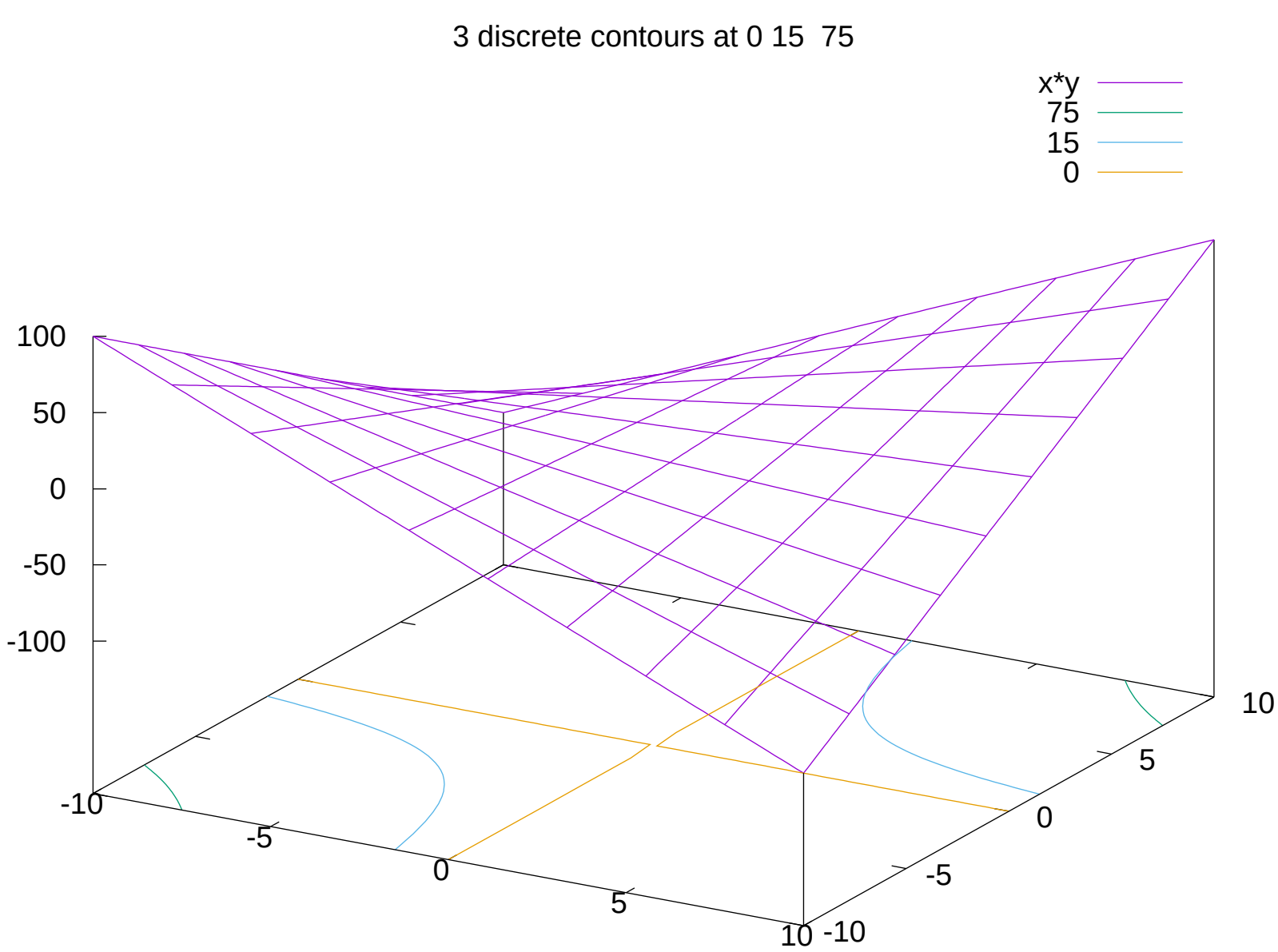

splot with "set pm3d" (implemented with some terminals)
'glass.dat' every 2::0::12
Z axis
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
X axis
Y axis
1.5
1
0.5
0
-0.5
-1

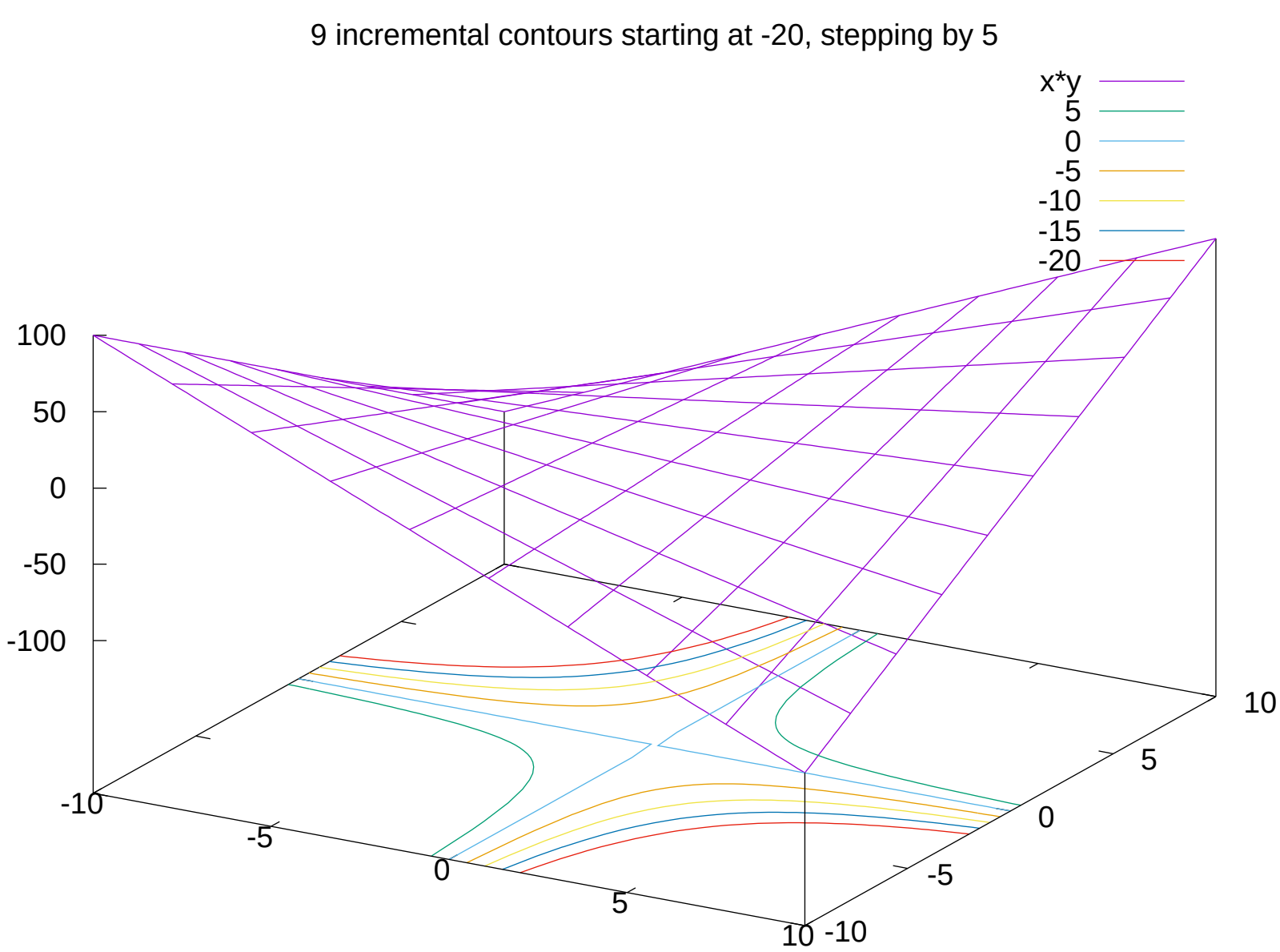

Test of spherical coordinates
"glass.dat"
Z axis
1
0.5
0
-0.5
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
X axis
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Y axis

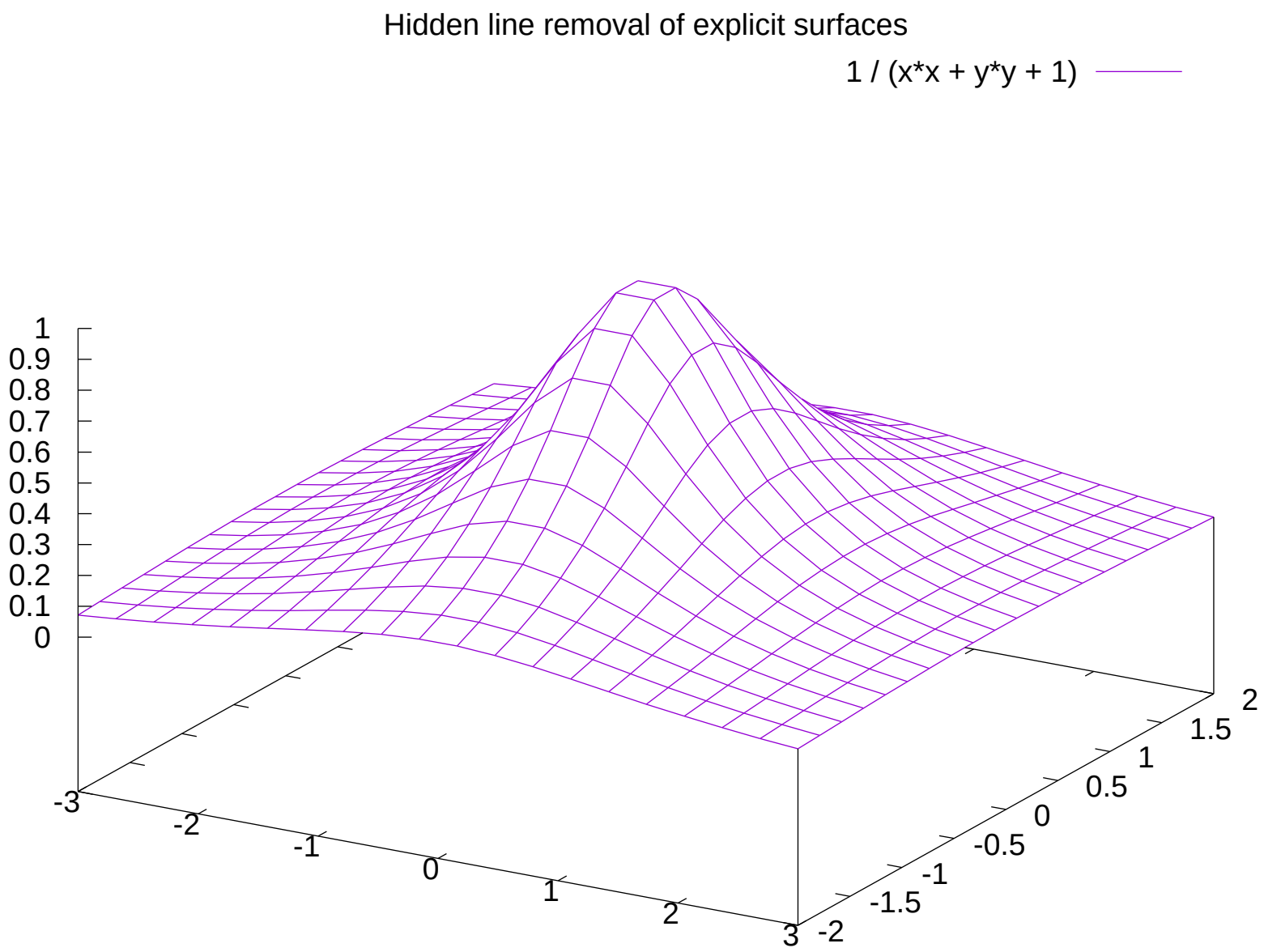

Mandelbrot function
mand({0,0},compl(x,y),30)
Z axis
35
30
25
20
15
10
5
-2
-1.5
-1
-0.5
0
0.5
1
X axis
-1.5
-1
-0.5
0
0.5
1
1.5
Y axis

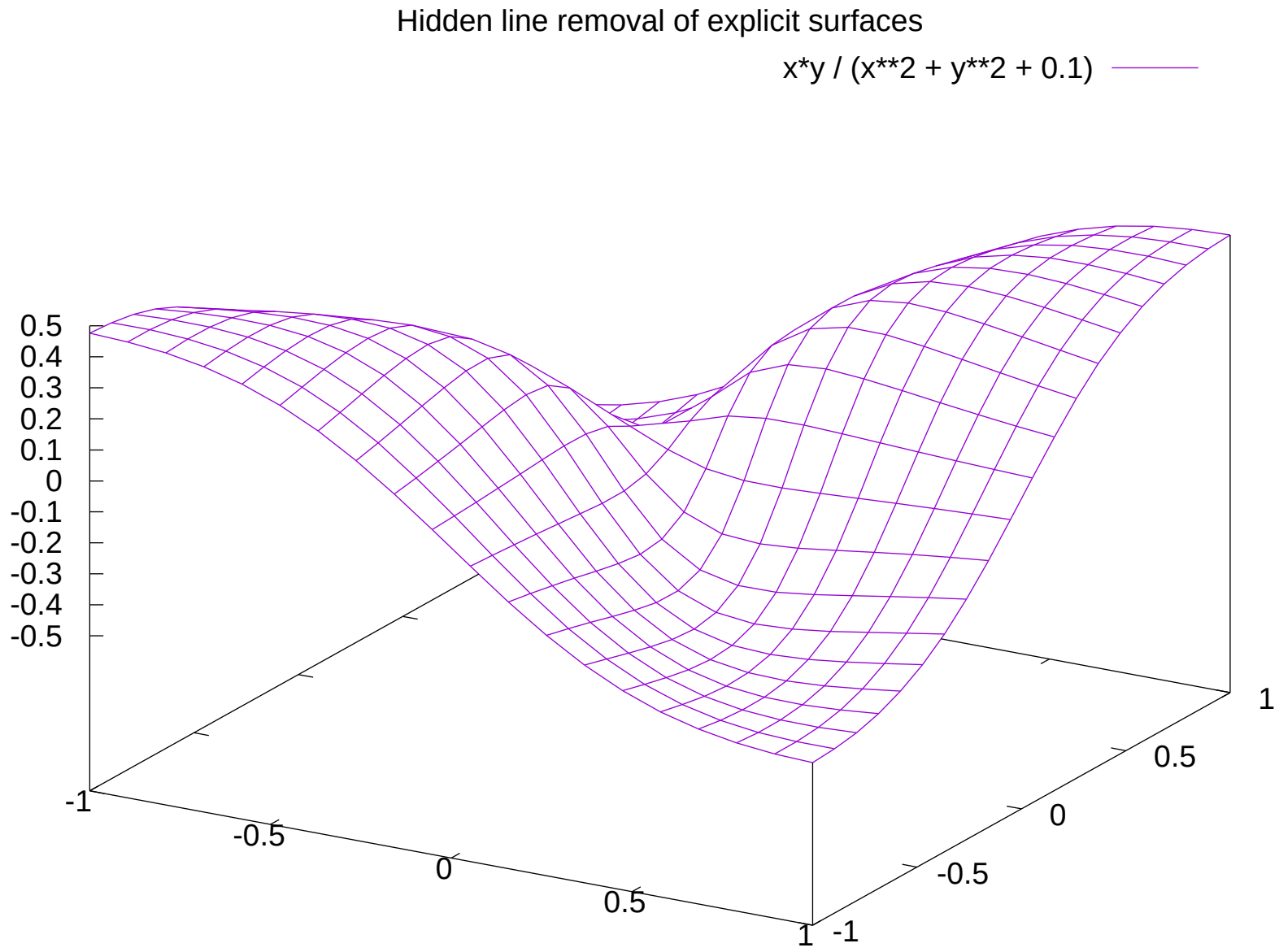
Demo of specifying discrete contour levels - default contours

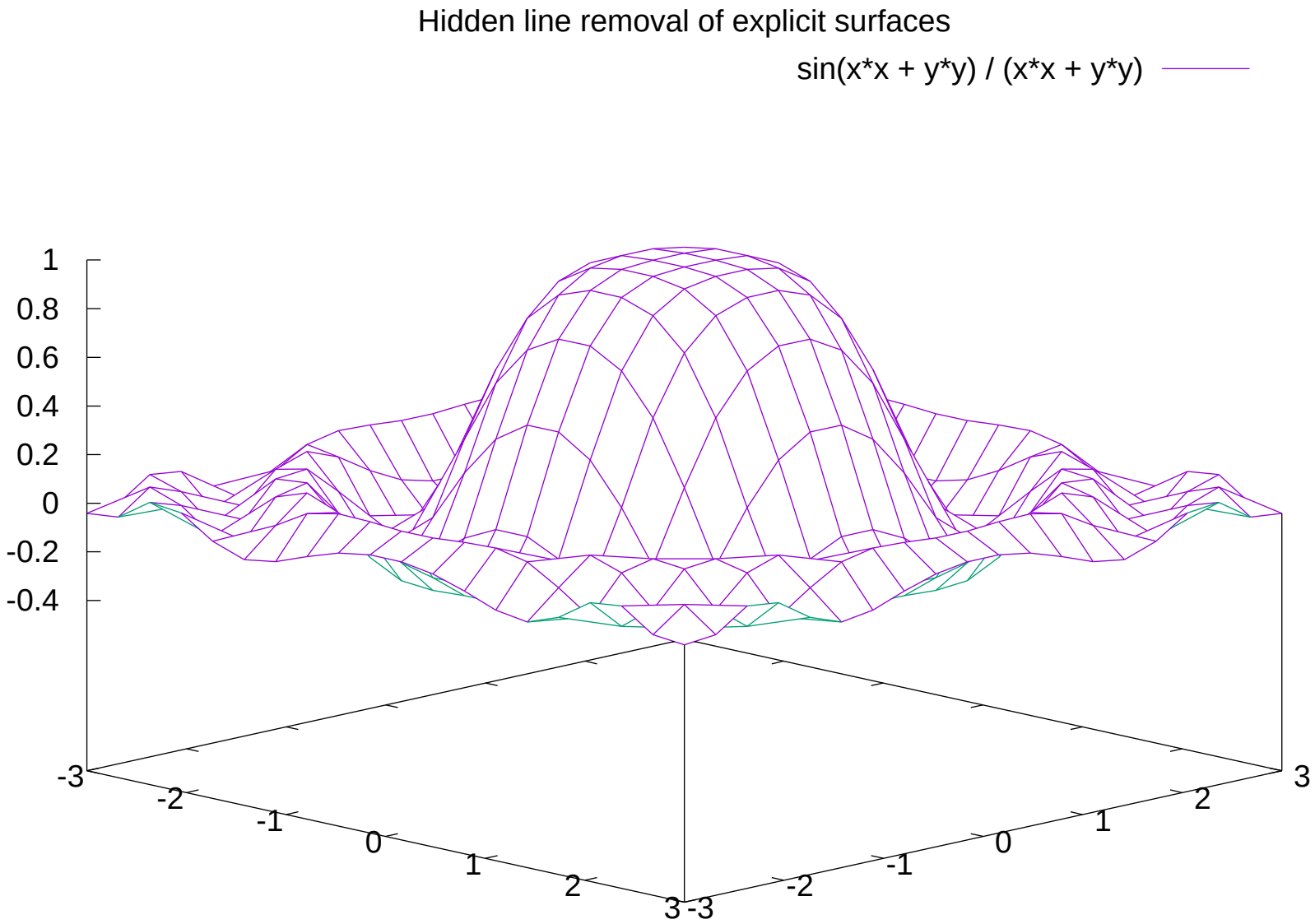


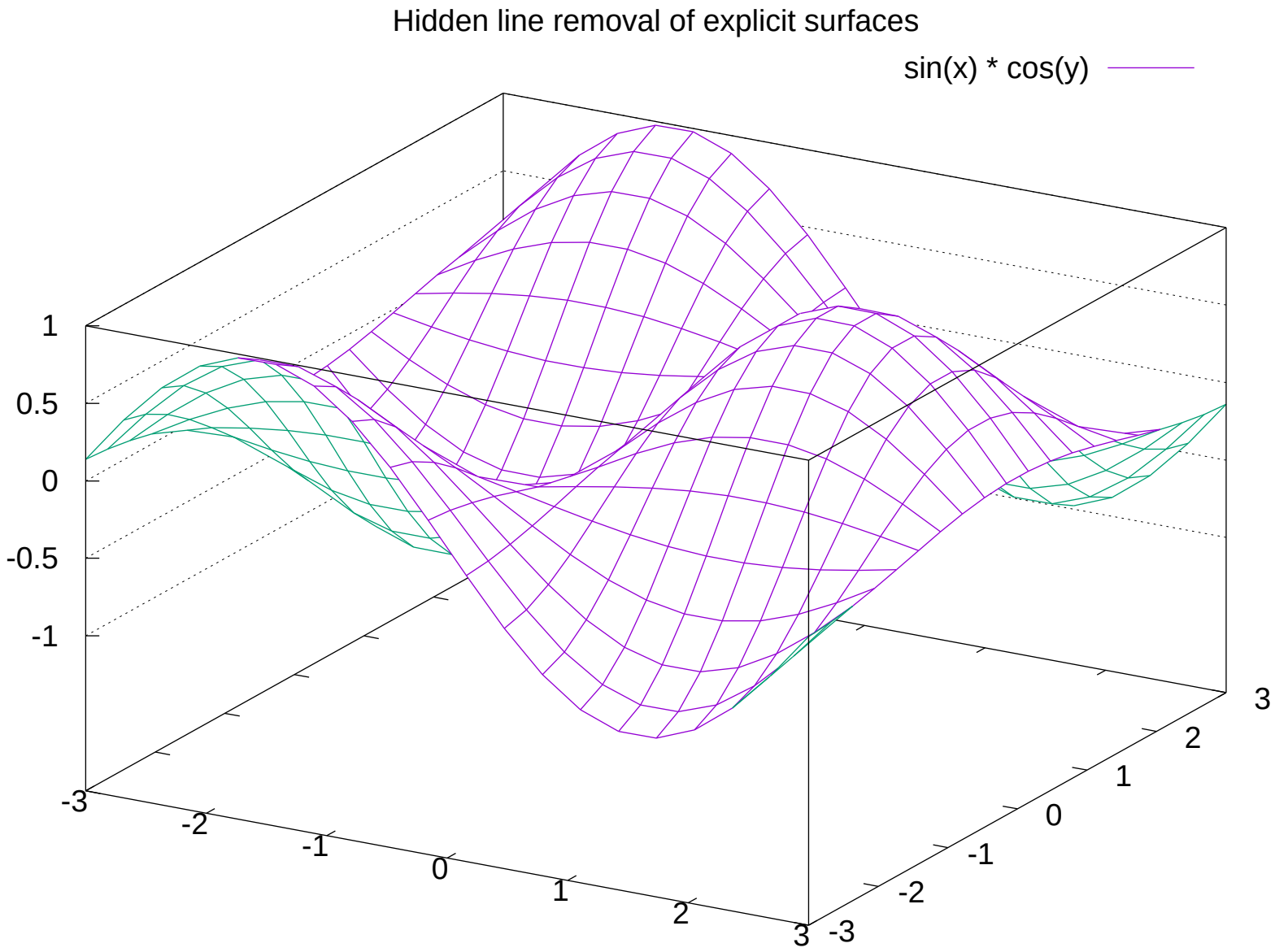

3 discrete contours at 0 15 75
x*y
75
15
0
100
50
0
-50
-100
-10
-5
0
5
10
-10
-5
0
5
10

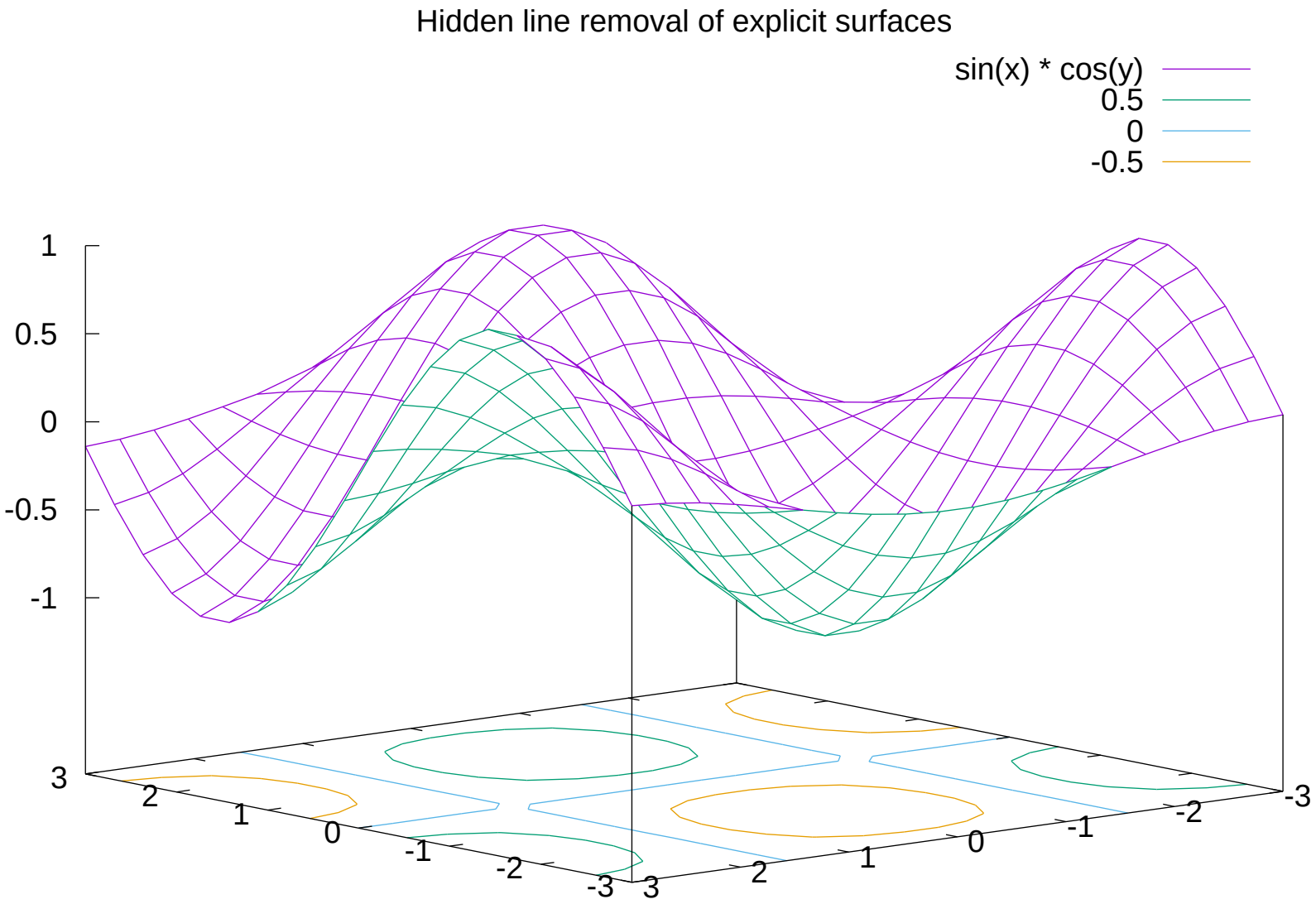

9 incremental contours starting at -20, stepping by 5
x*y
5
0
-5
-10
-15
-20
100
50
0
-50
-100
-10
-5
0
5
10
-10
-5
0
5
10

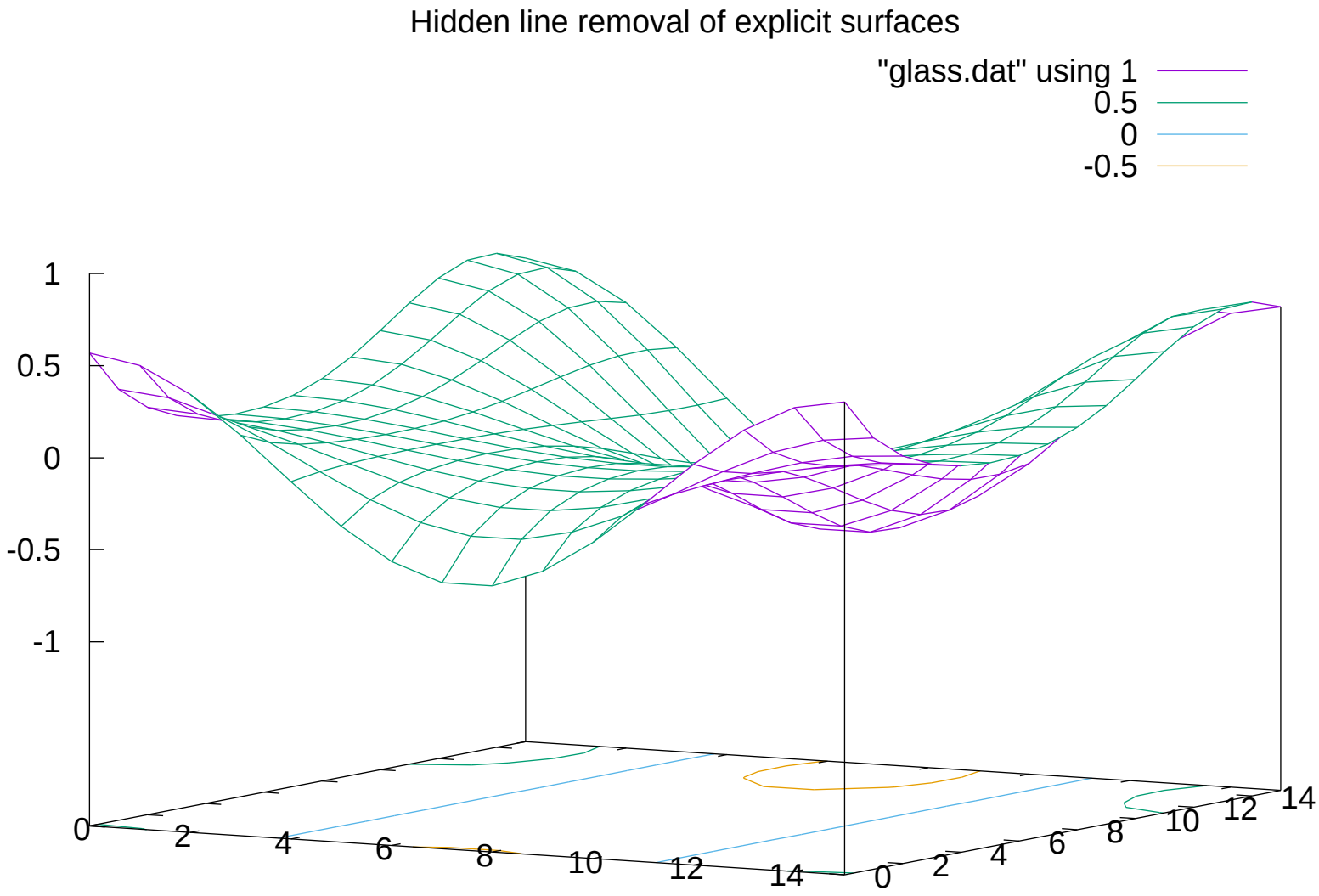

Hidden line removal of explicit surfaces
1 / (x*x + y*y + 1)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-3
-2
-1
0
1
2
3
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

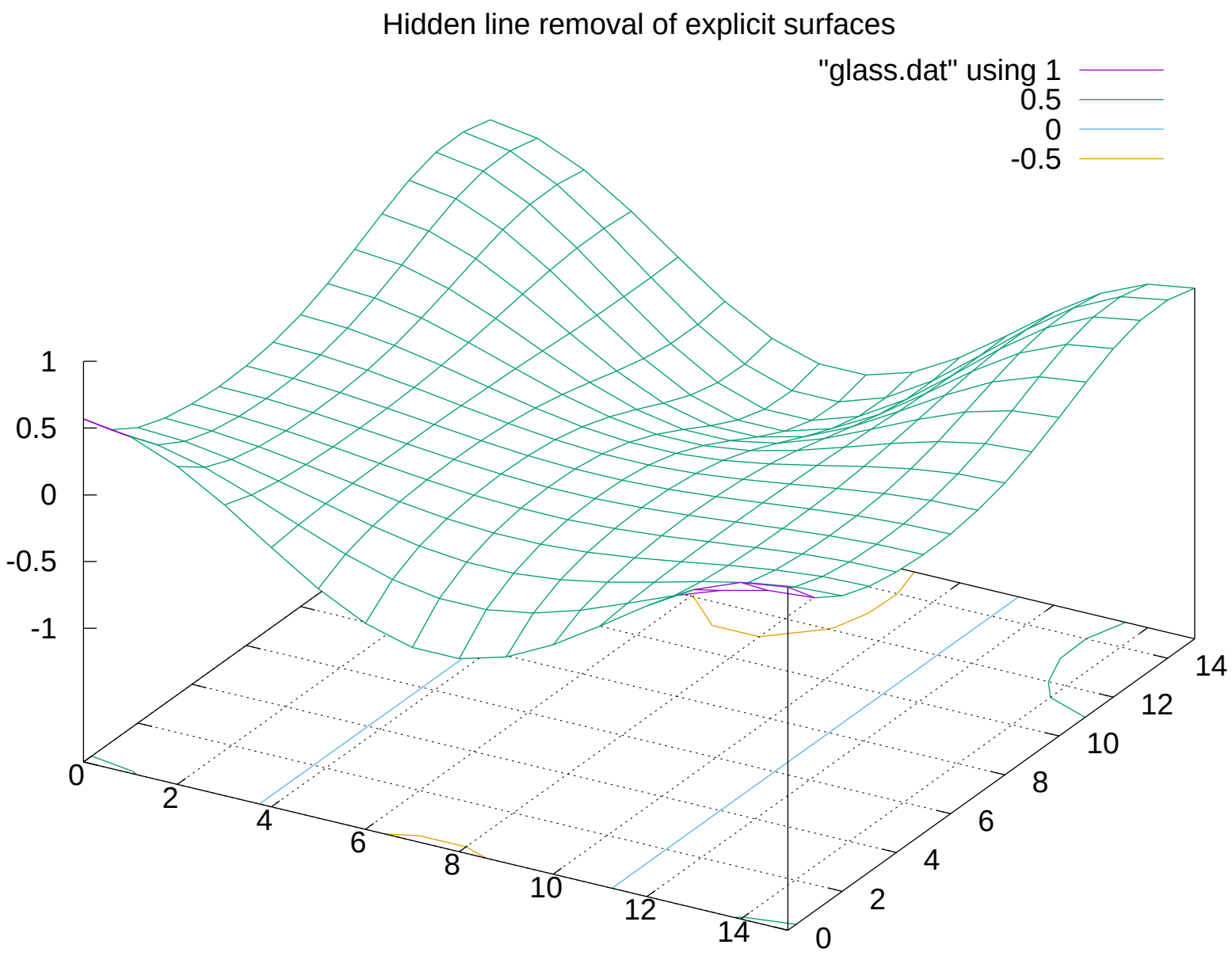

Hidden line removal of explicit surfaces
x*y / (x**2 + y**2 + 0.1)
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1

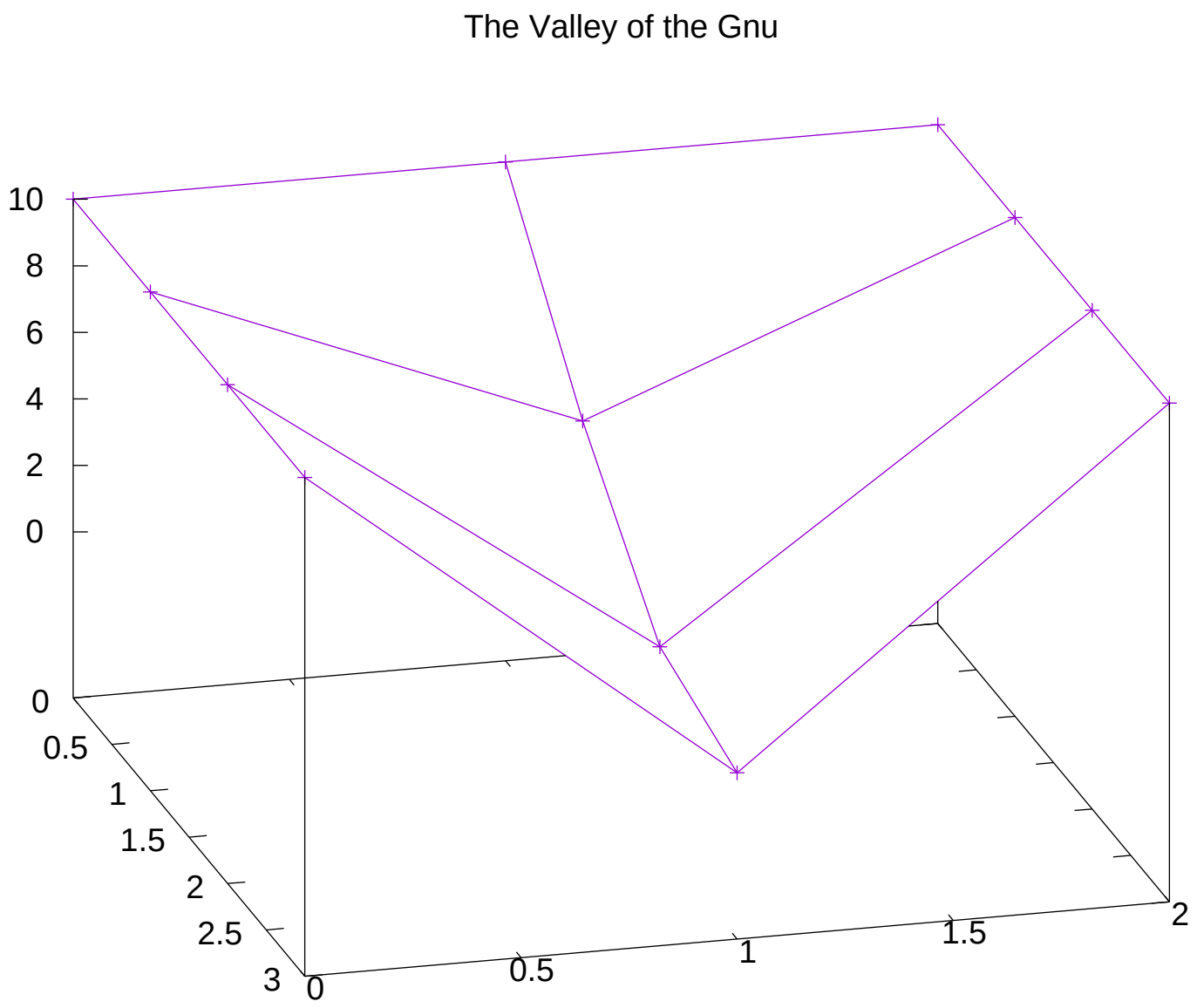

Hidden line removal of explicit surfaces
sin(x*x + y*y) / (x*x + y*y)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-3
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3

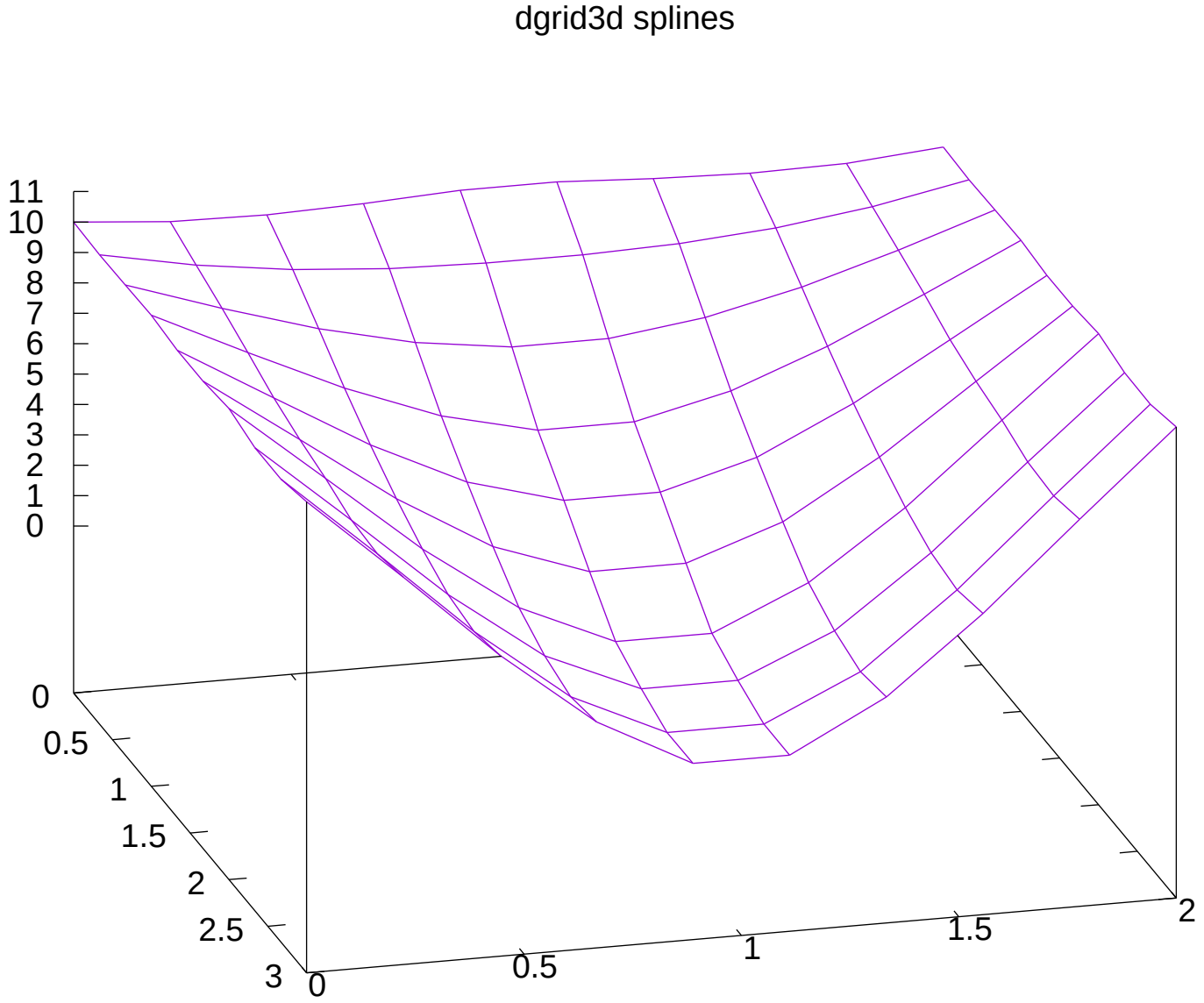

Hidden line removal of explicit surfaces
sin(x) * cos(y)
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3


Hidden line removal of explicit surfaces
sin(x) * cos(y)
0.5
0
-0.5
1
0.5
0
-0.5
-1
3
2
1
0
-1
-2
-3
3
2
1
0
-1
-2
-3

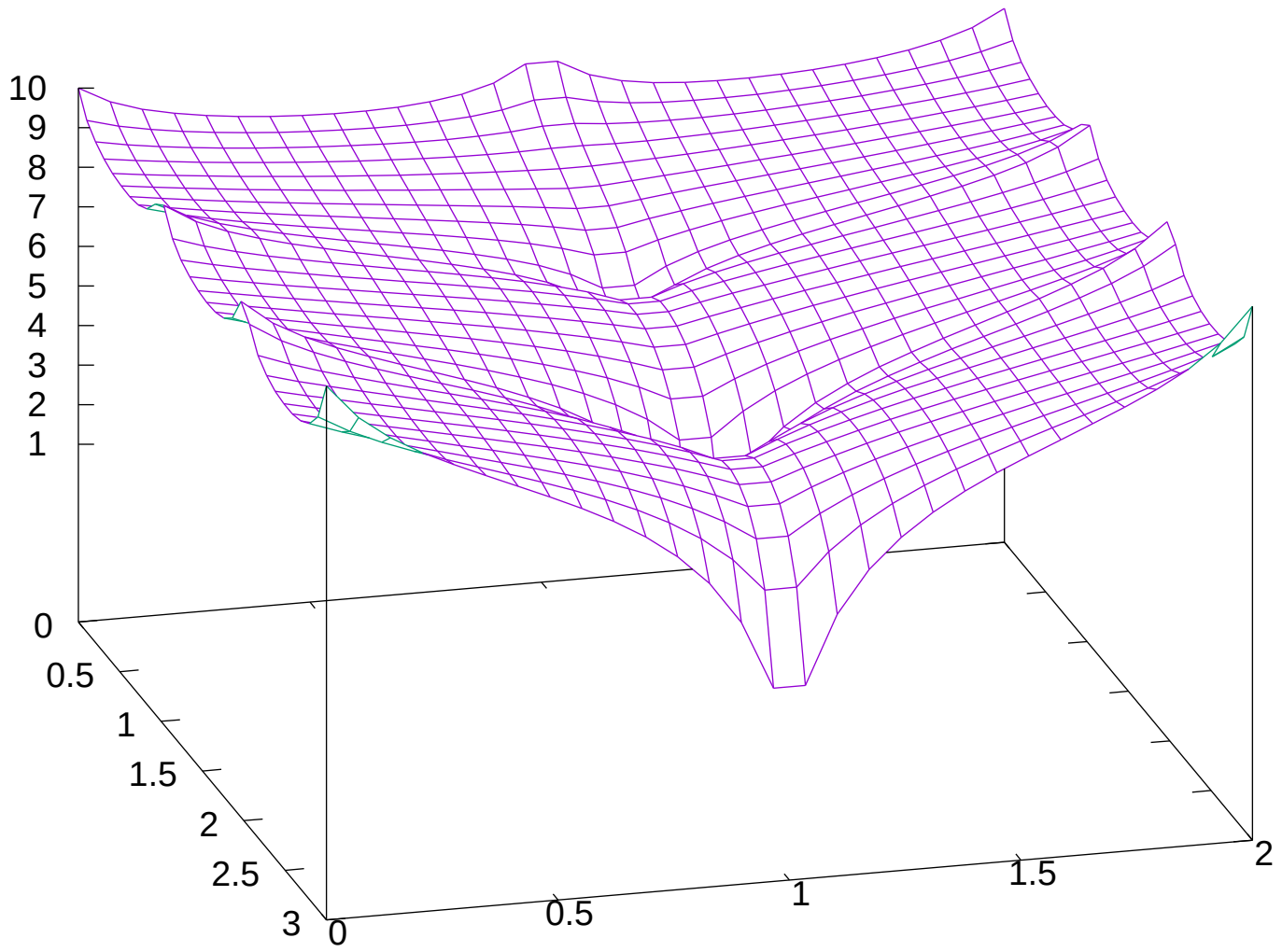

Hidden line removal of explicit surfaces
"glass.dat" using 1
0.5
0
-0.5
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

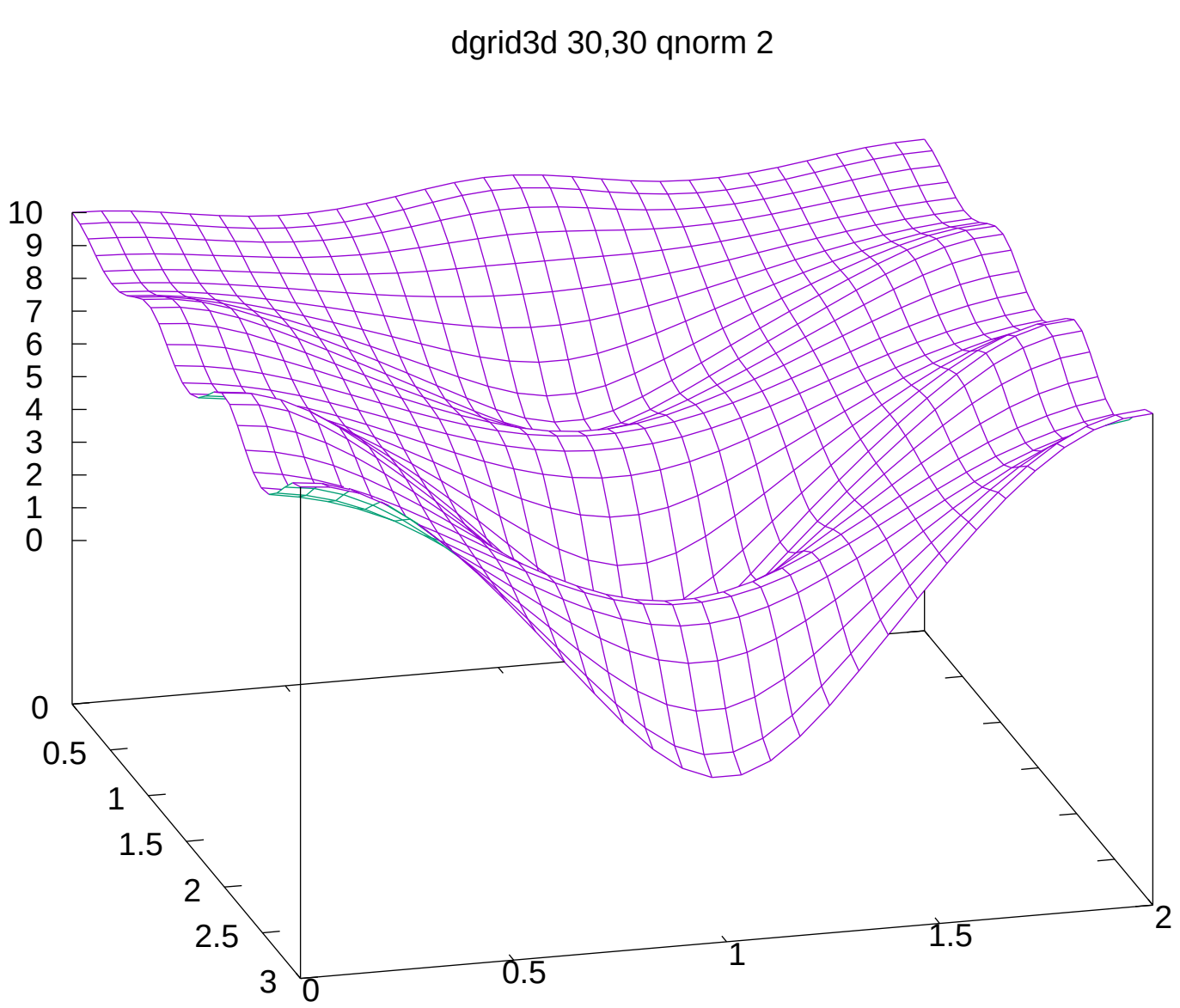

Hidden line removal of explicit surfaces
"glass.dat" using 1
0.5
0
-0.5
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

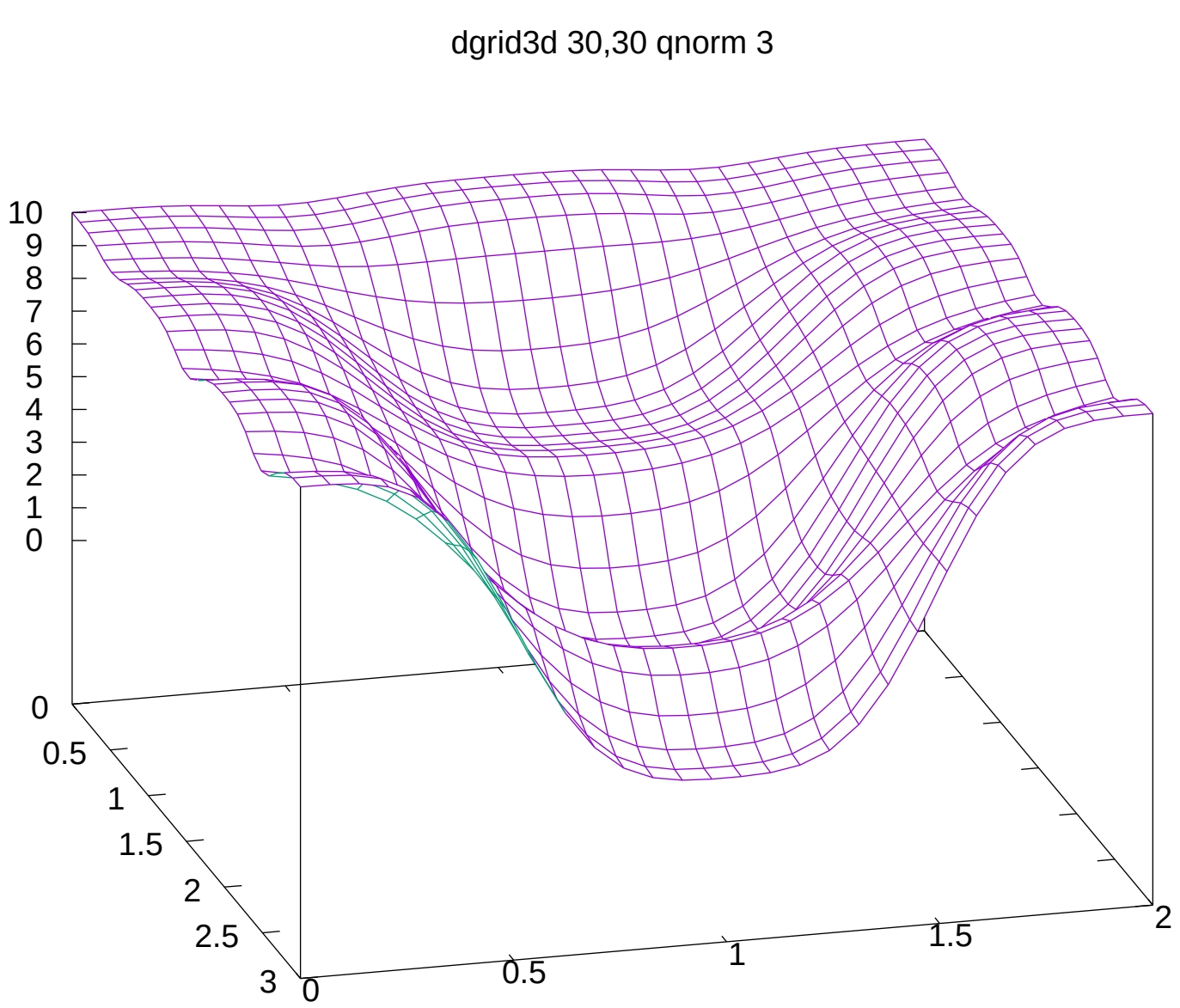

The Valley of the Gnu
10
8
6
4
2
0
0
0.5
1
1.5
2
2.5
3
0
0.5
1
1.5
2

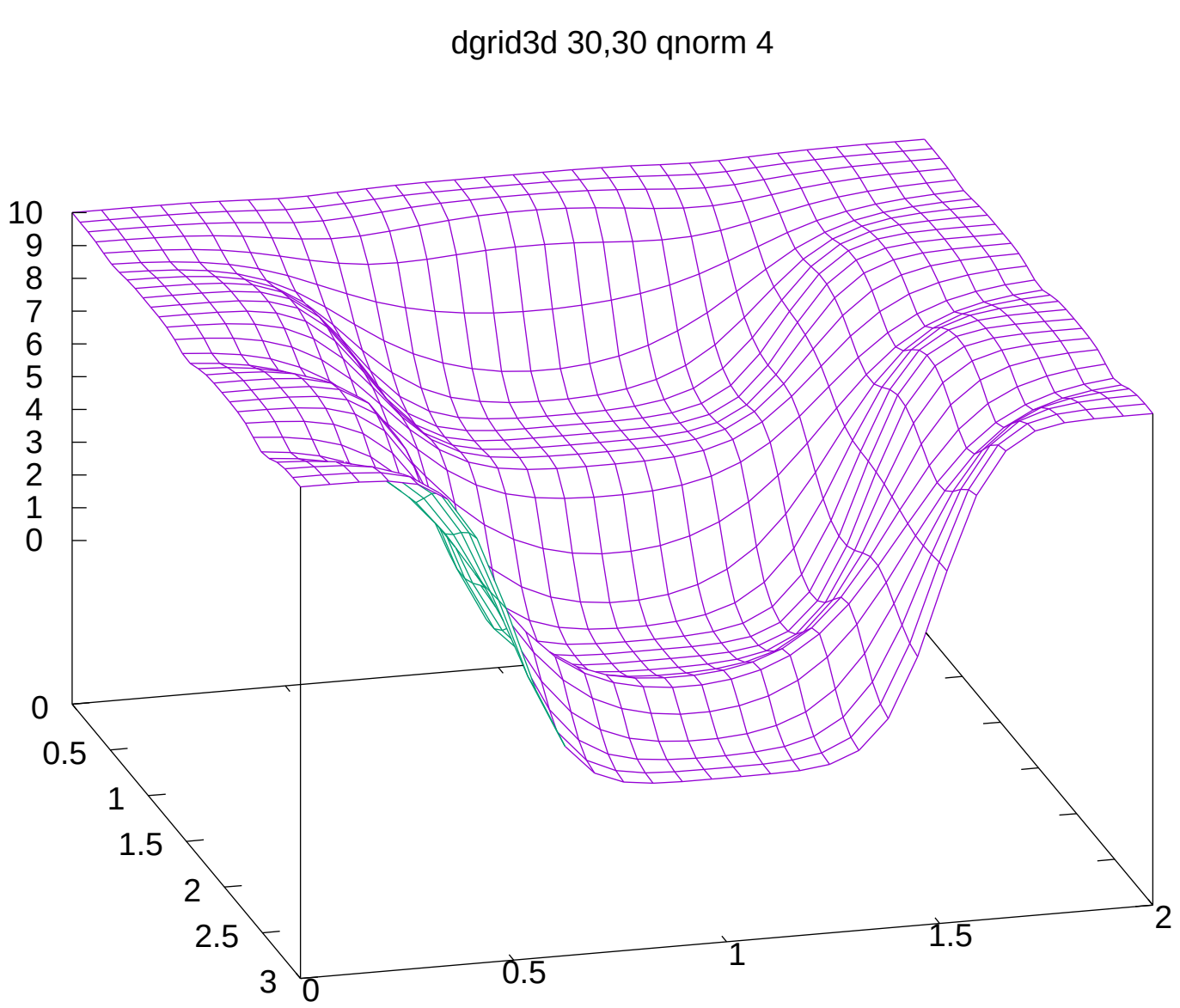

dgrid3d splines
11
10
9
8
7
6
5
4
3
2
1
0
0
0.5
1
1.5
2
2.5
3
0
0.5
1
1.5
2

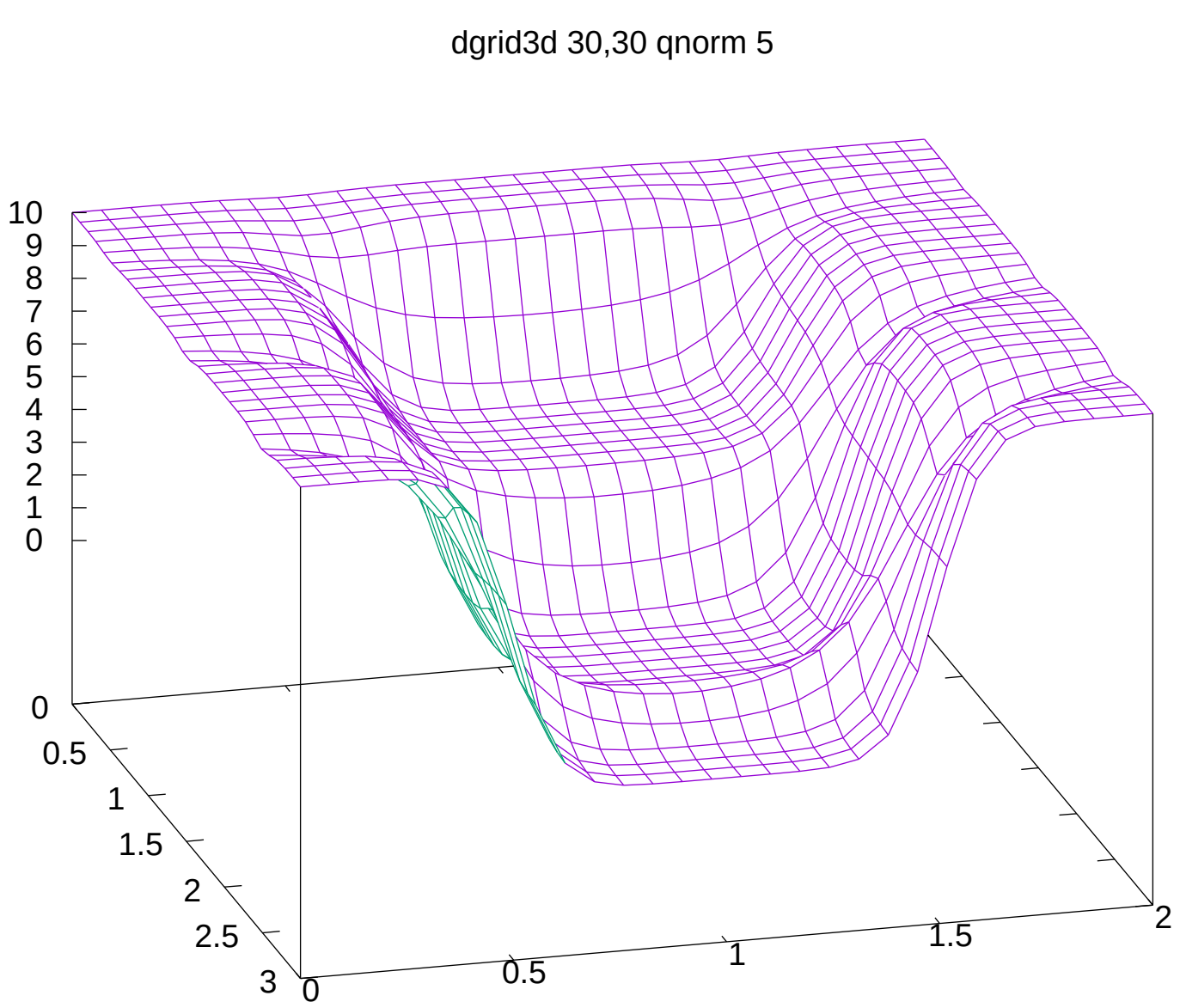
dgrid3d 30,30 qnorm 1



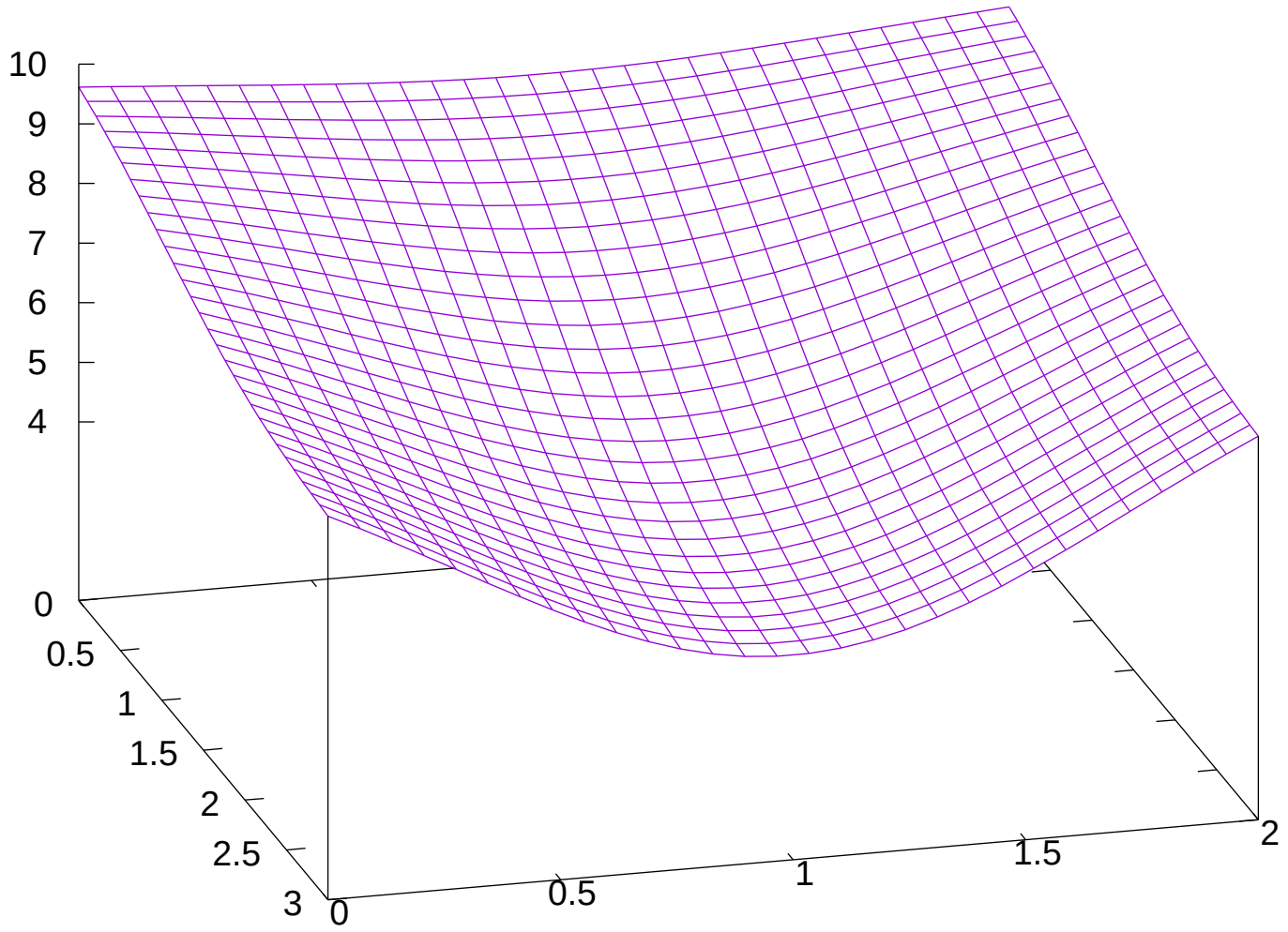

dgrid3d 30,30 qnorm 2
10
9
8
7
6
5
4
3
2
1
0
0
0.5
1
1.5
2
2.5
3
0
0.5
1
1.5
2


dgrid3d 30,30 qnorm 3
10
9
8
7
6
5
4
3
2
1
0
0
0.5
1
1.5
2
2.5
3
0
0.5
1
1.5
2

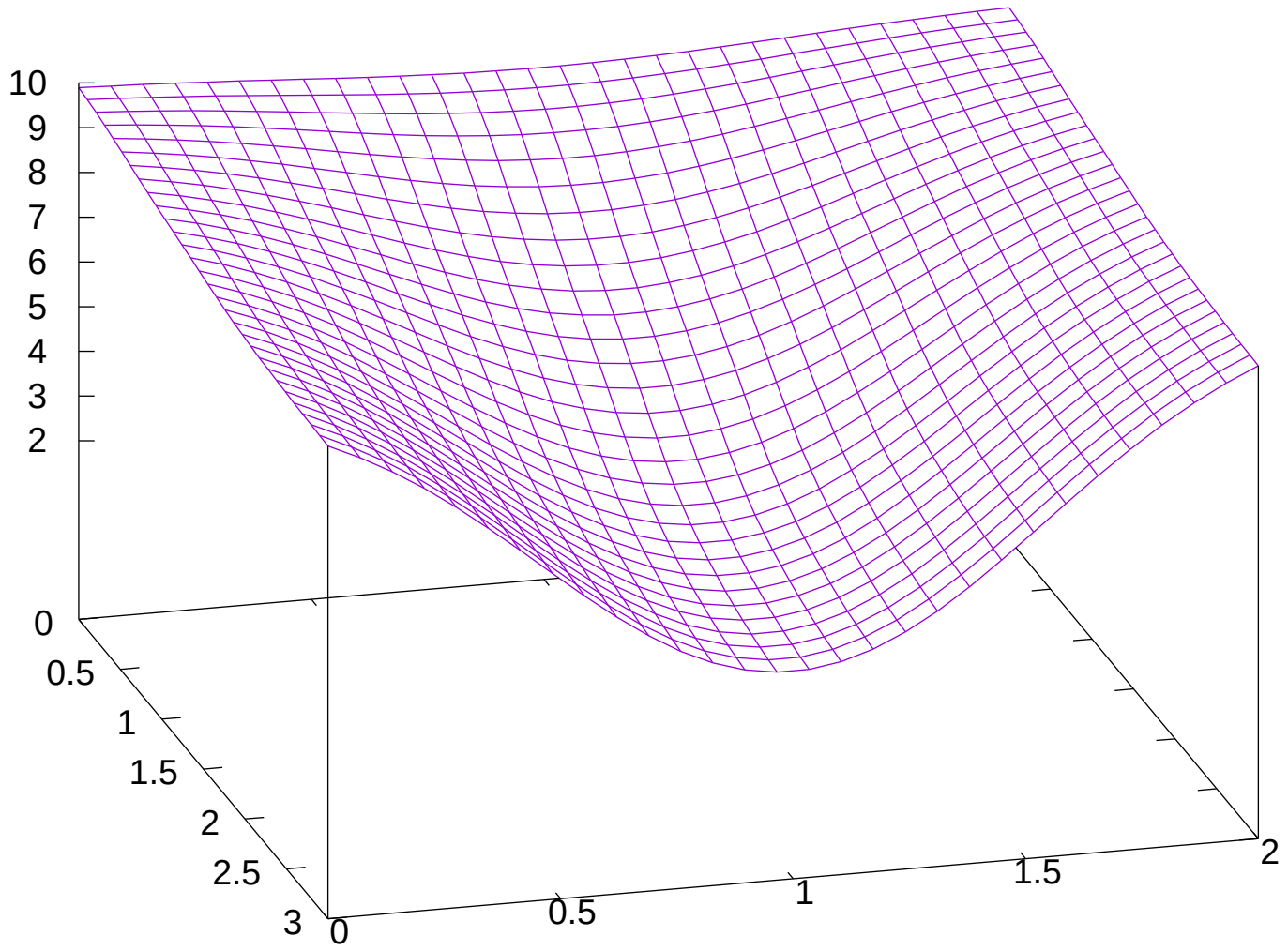

dgrid3d 30,30 qnorm 4
10
9
8
7
6
5
4
3
2
1
0
0
0.5
1
1.5
2
2.5
3
0
0.5
1
1.5
2


dgrid3d 30,30 qnorm 5
10
9
8
7
6
5
4
3
2
1
0
0
0.5
1
1.5
2
2.5
3
0
0.5
1
1.5
2

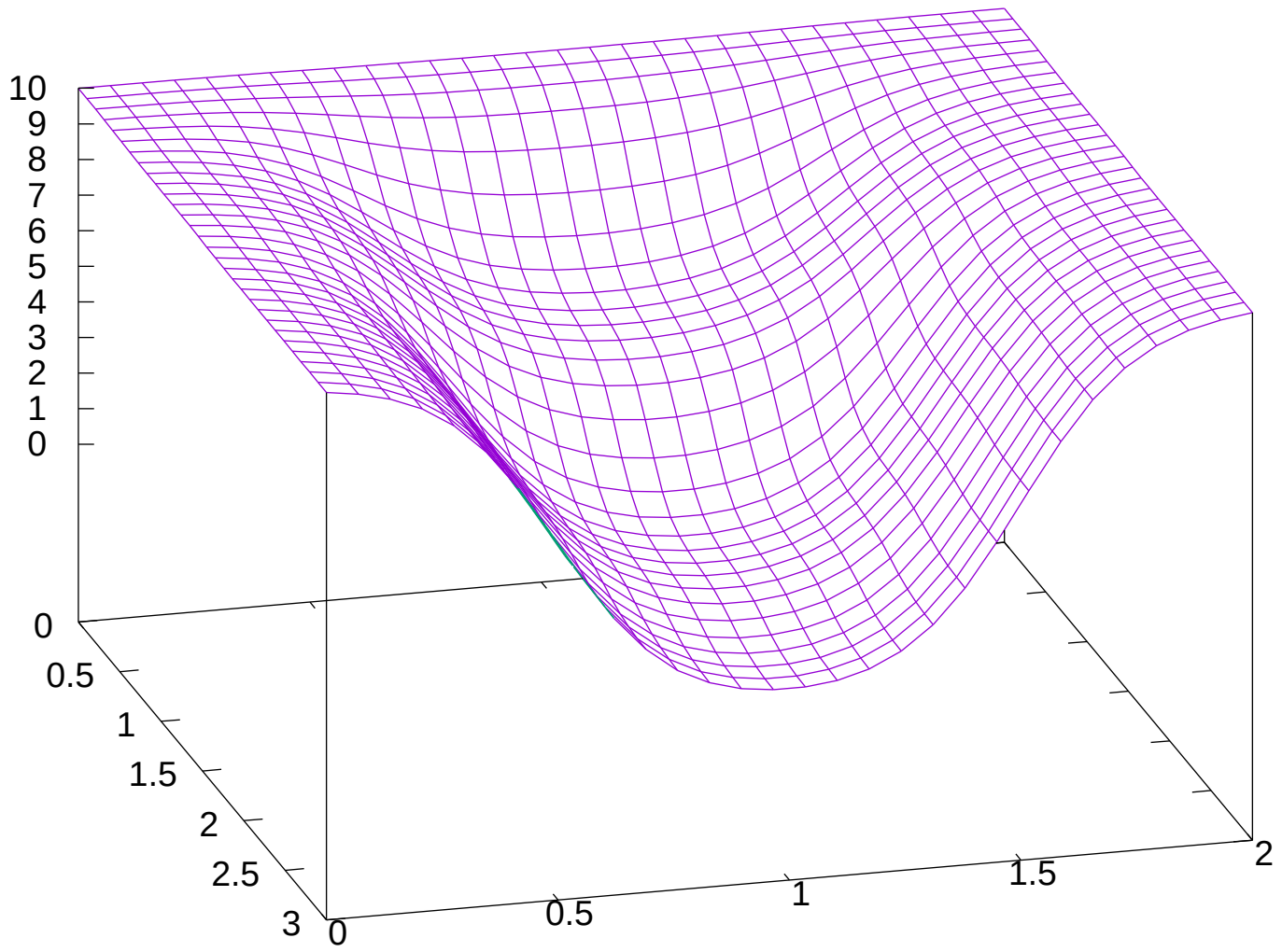
dgrid3d 30,30 gauss 1



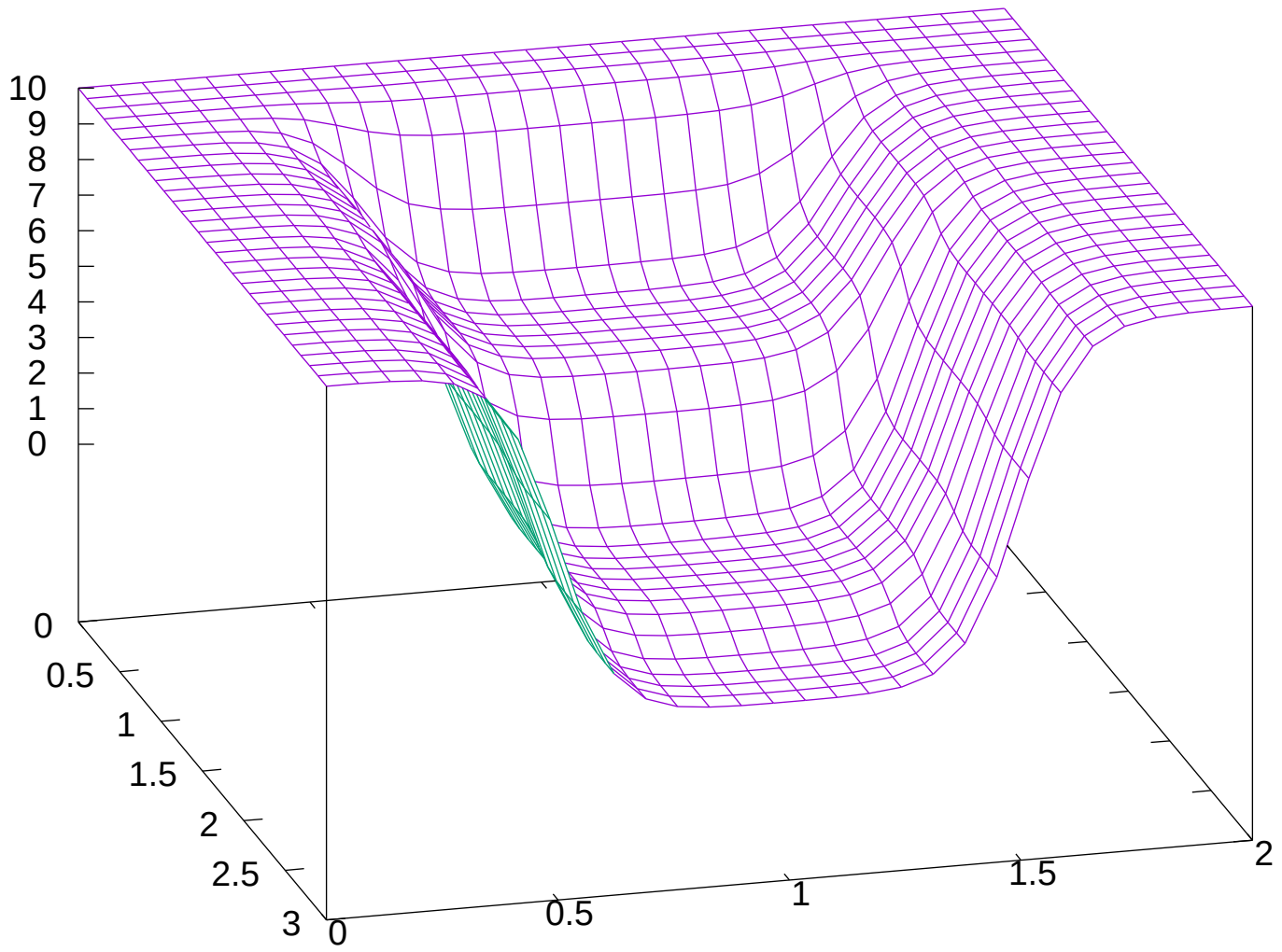
dgrid3d 30,30 gauss .75



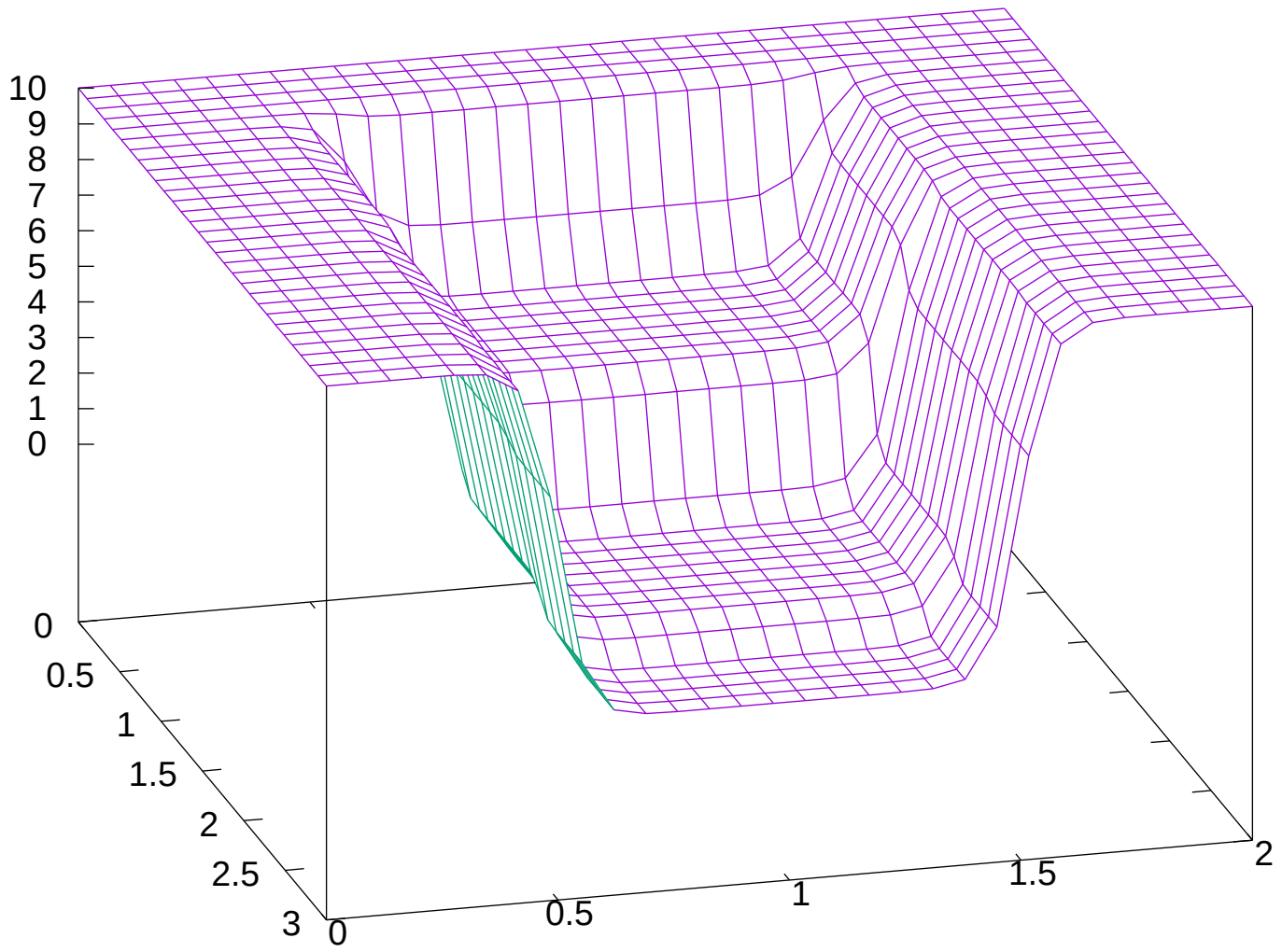
dgrid3d 30,30 gauss .5



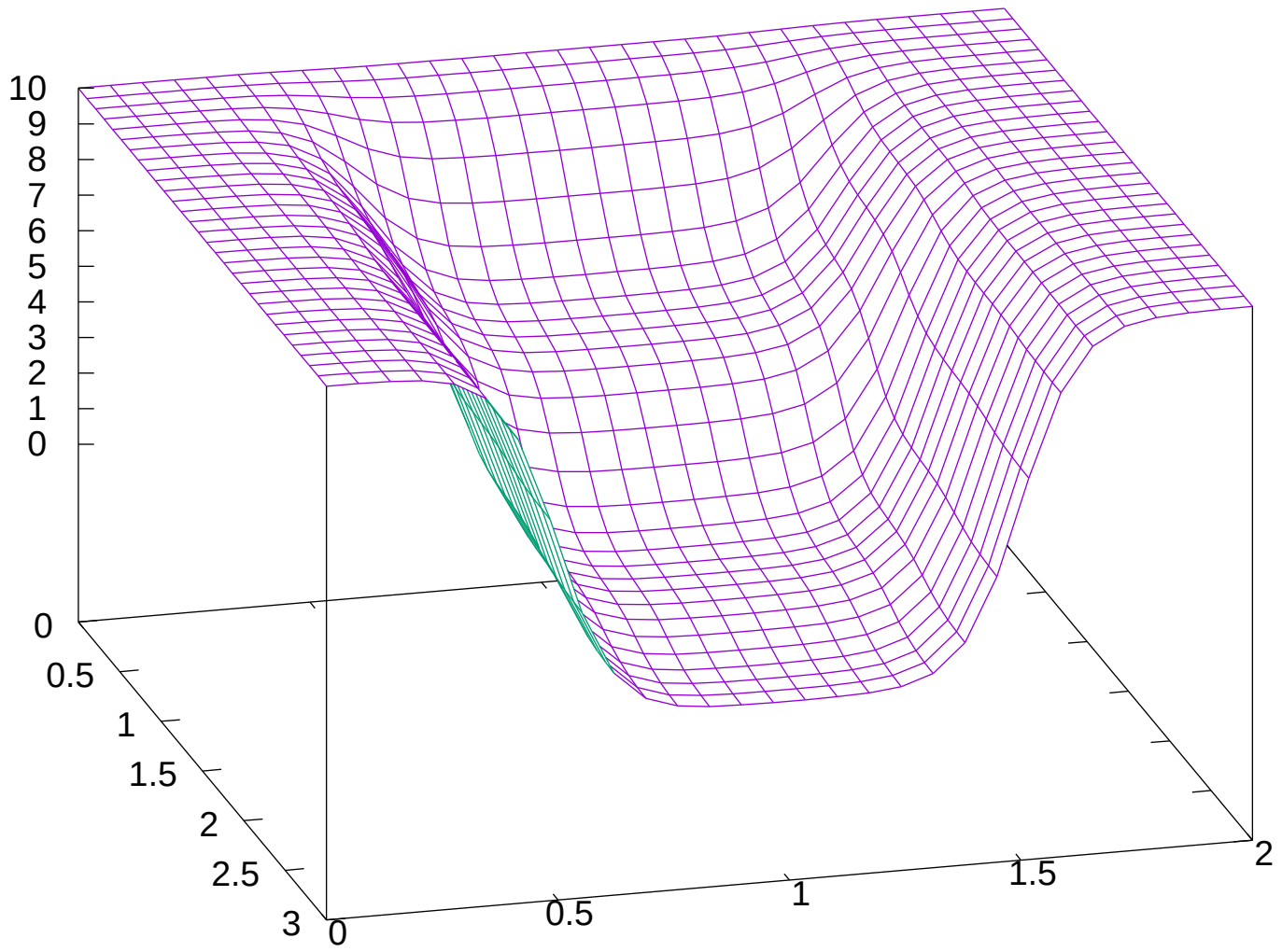
dgrid3d 30,30 gauss .35

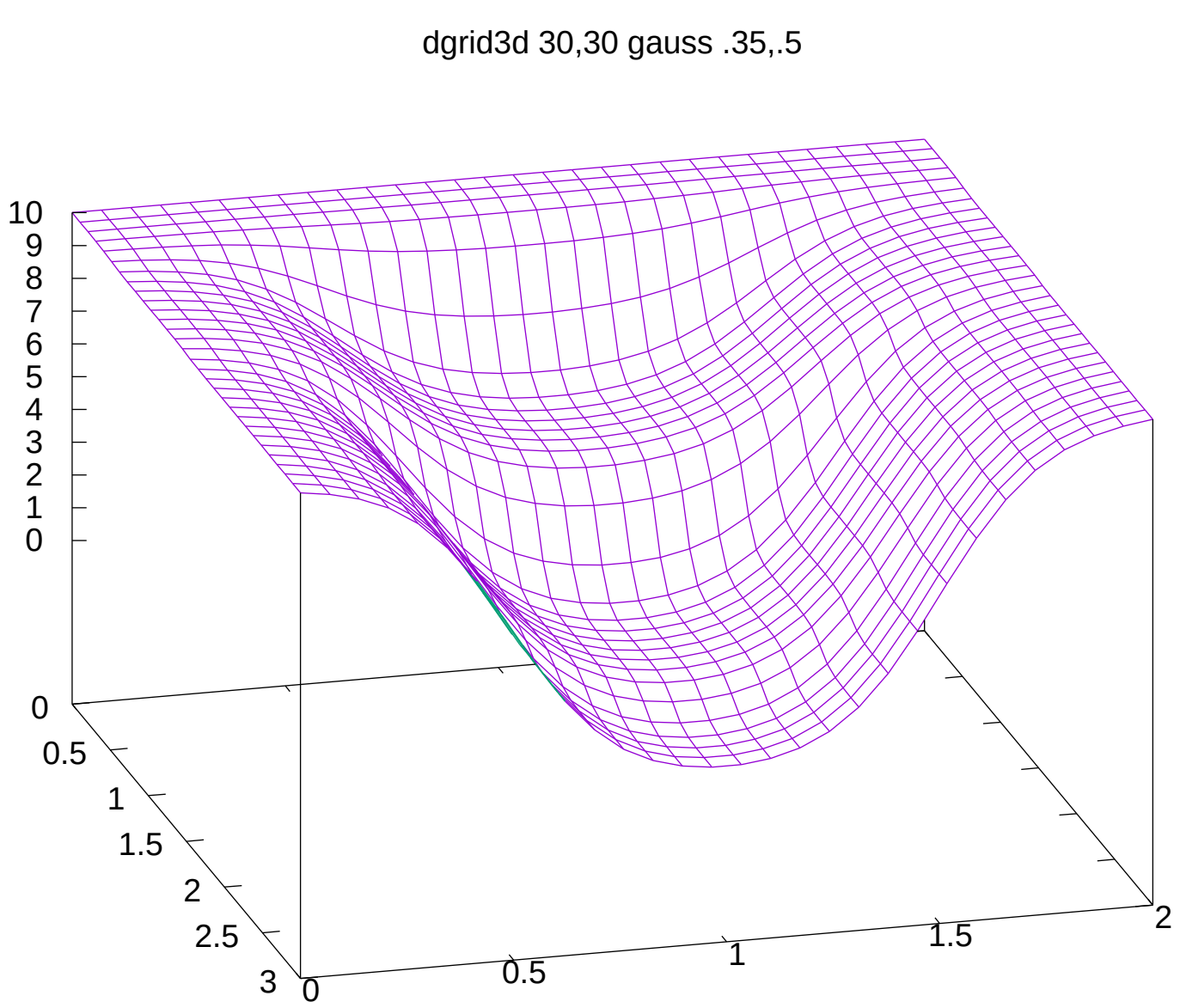


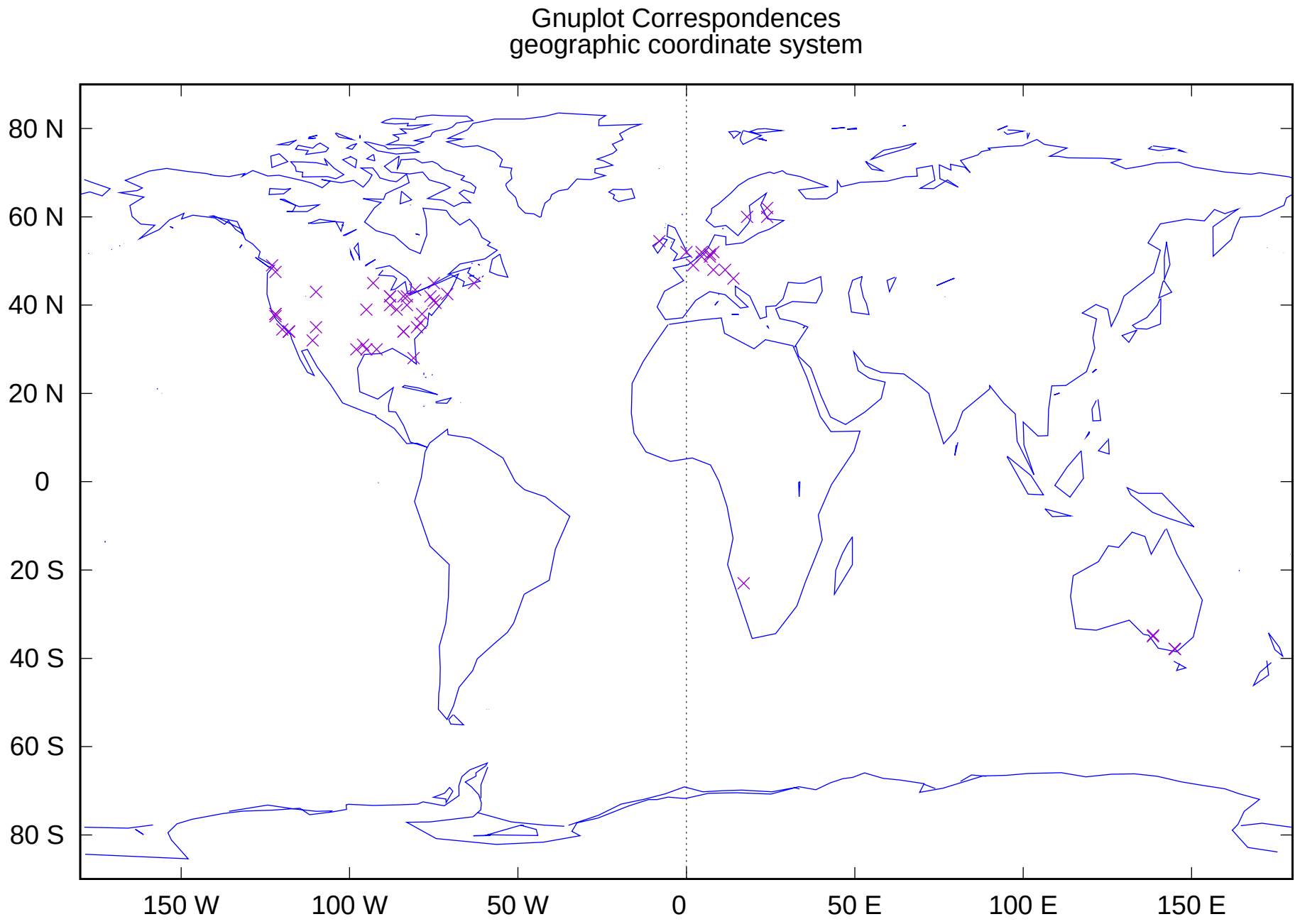
dgrid3d 30,30 gauss .25



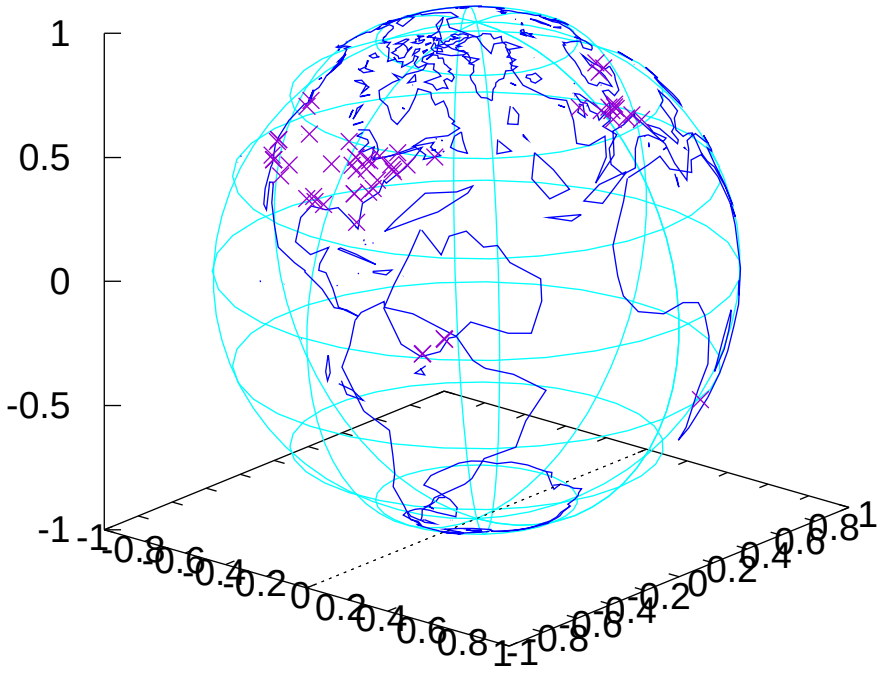
dgrid3d 30,30 gauss .5,.35



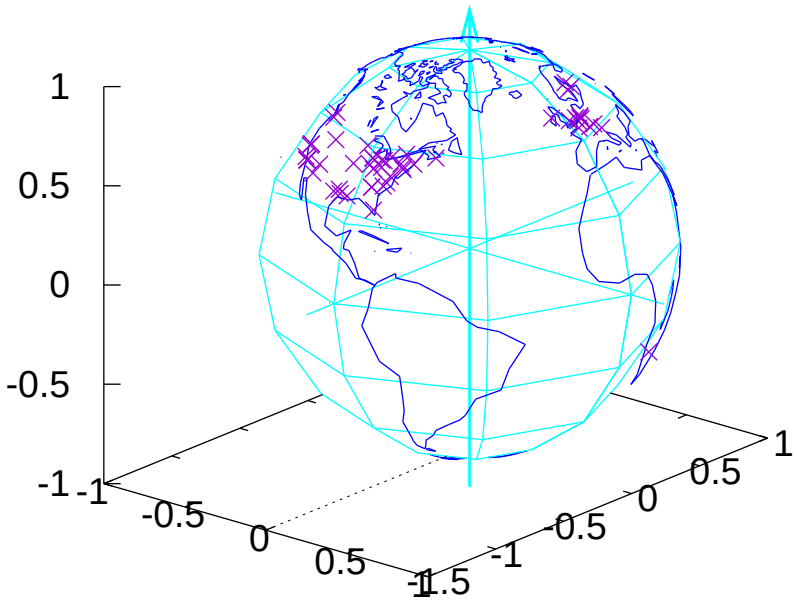

dgrid3d 30,30 gauss .35,.5
10
9
8
7
6
5
4
3
2
1
0
0
0.5
1
1.5
2
2.5
3
0
0.5
1
1.5
2


Gnuplot Correspondences
geographic coordinate system
80 N
60 N
40 N
20 N
0
20 S
40 S
60 S
80 S
150 W
100 W
50 W
0
50 E
100 E
150 E

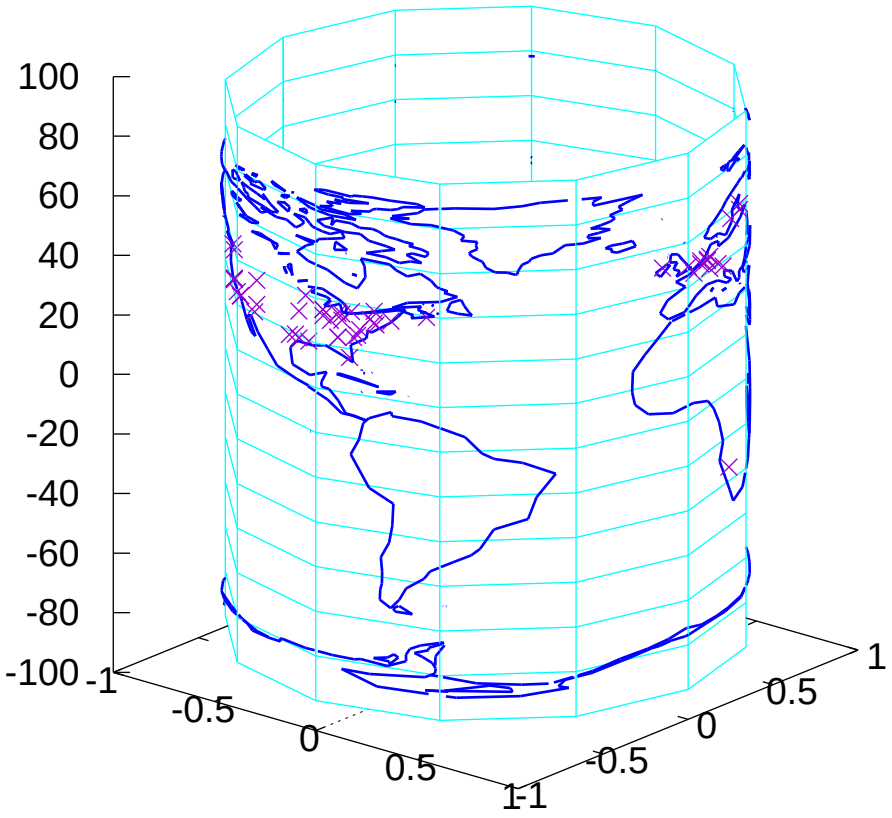
3D version using spherical coordinate system

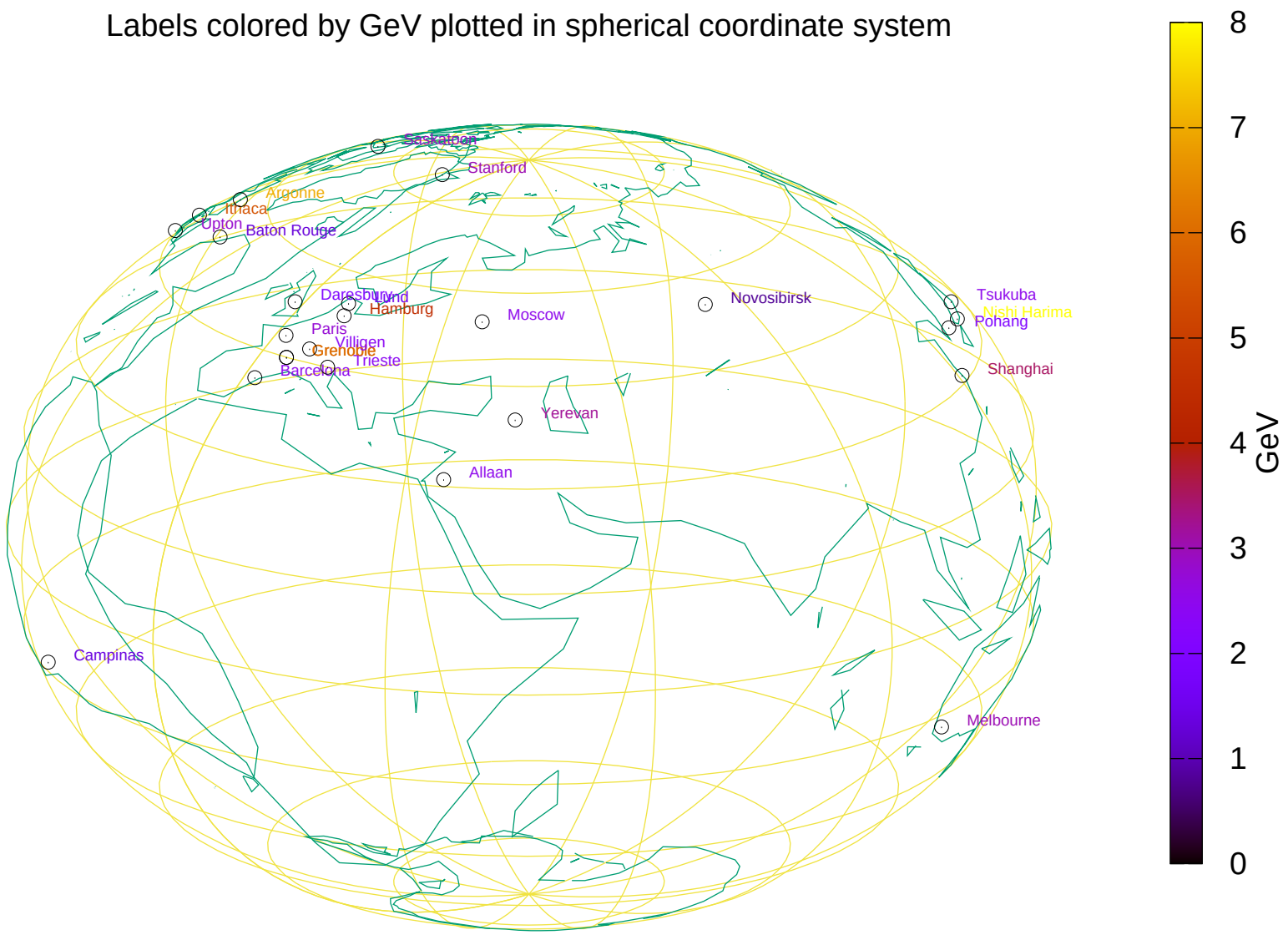


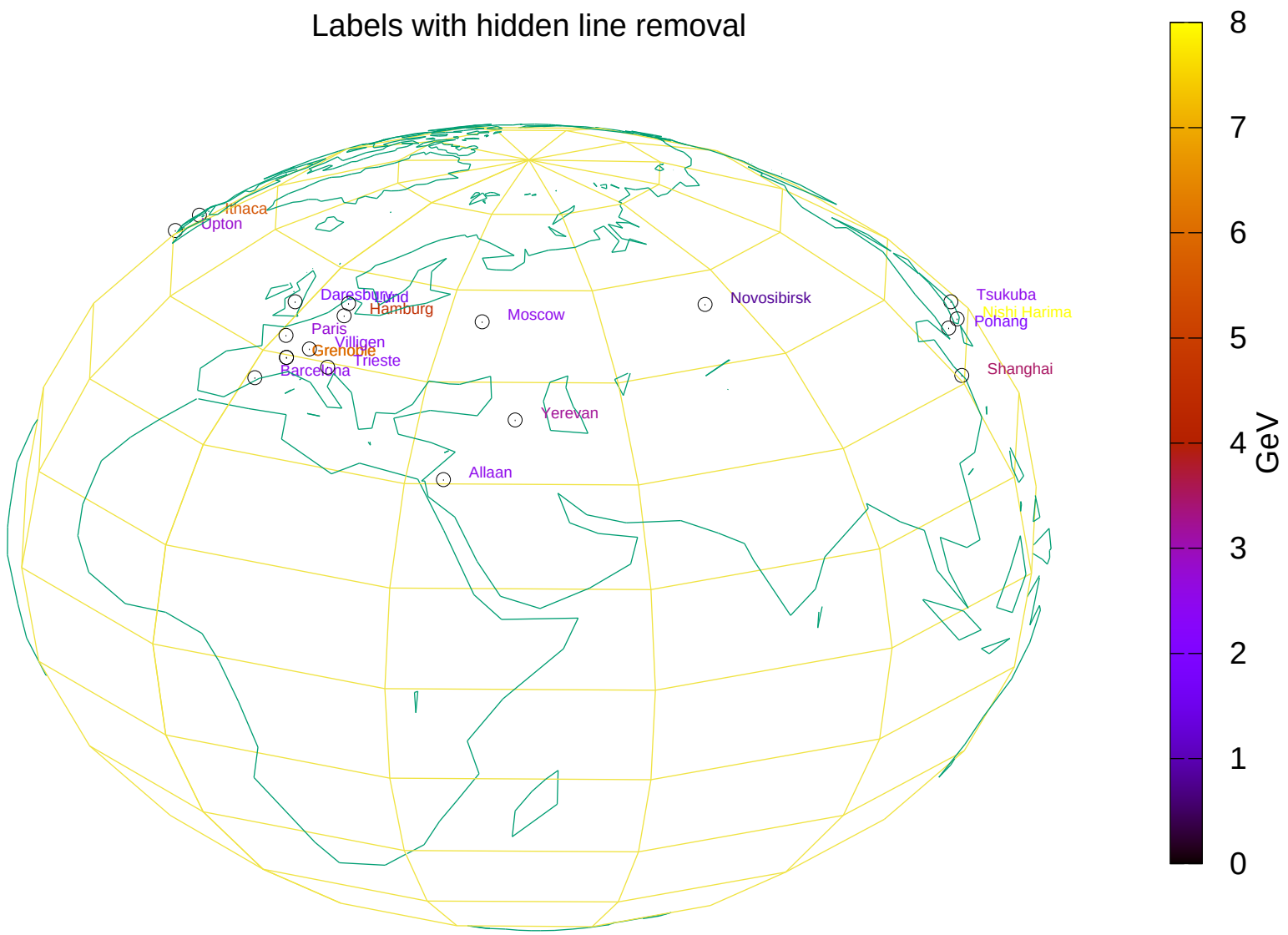
3D solid version with hidden line removal

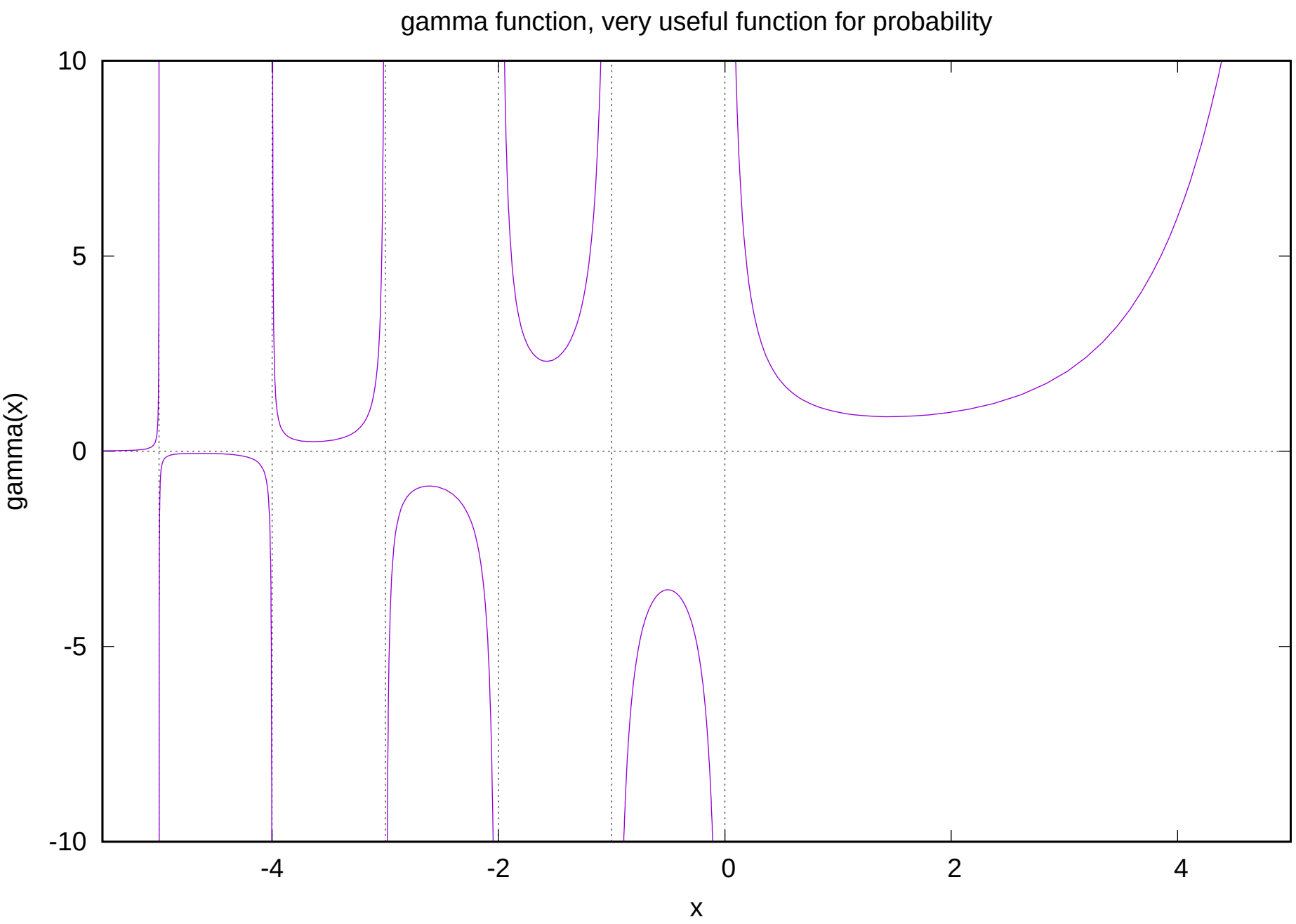


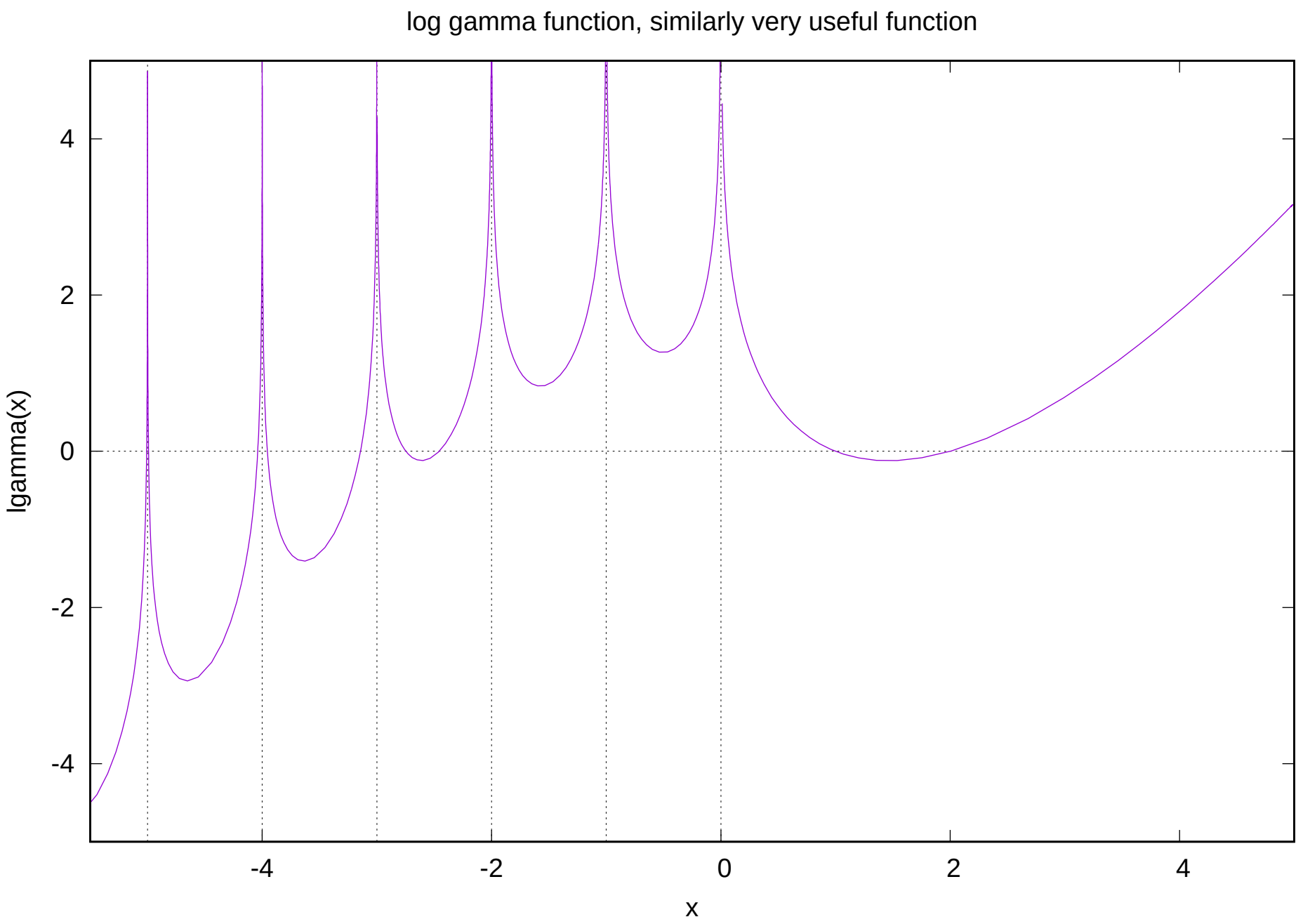
3D version using cylindrical coordinate system

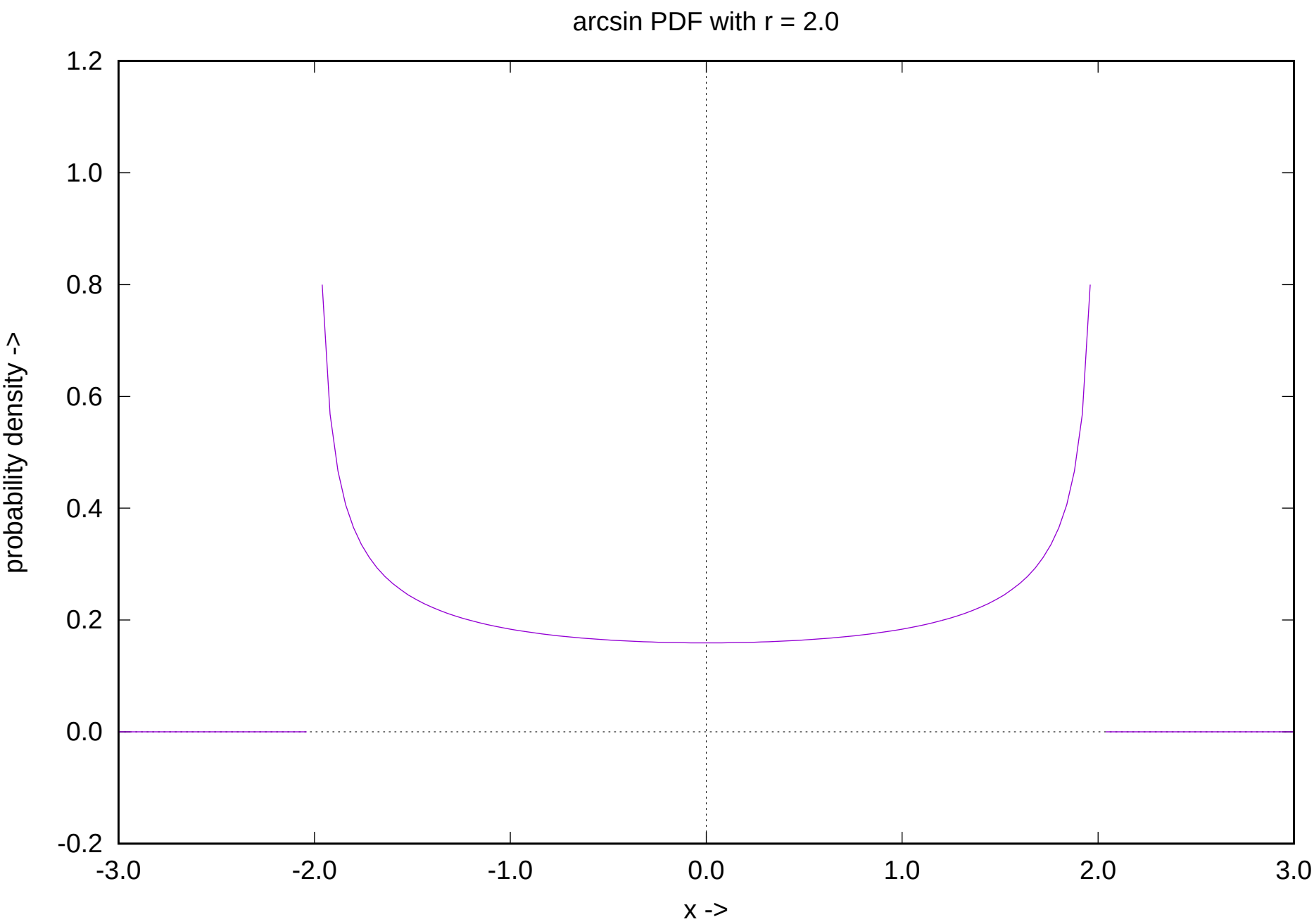


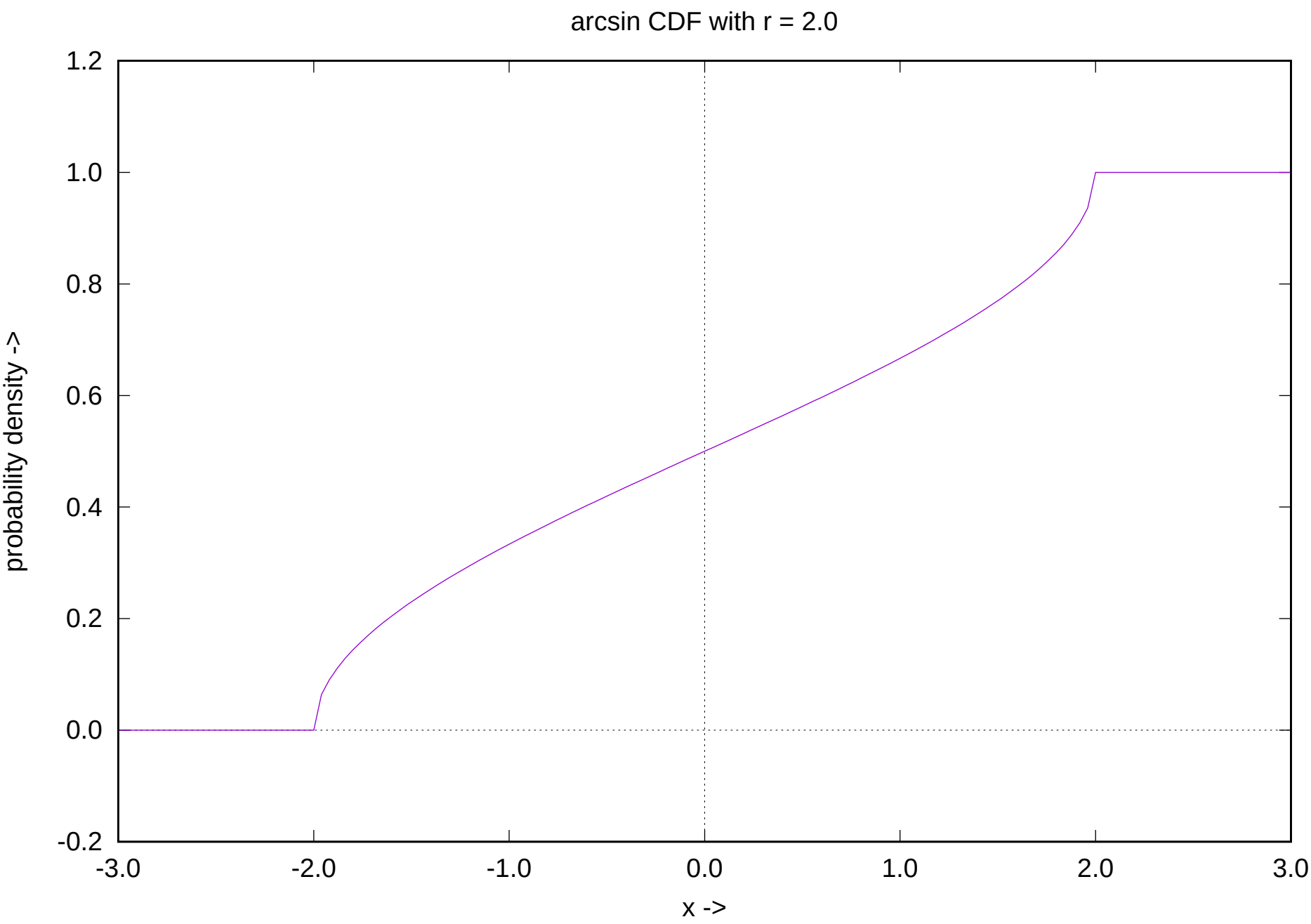

Labels colored by GeV plotted in spherical coordinate system
Saskatoon
Stanford
Argonne
Ithaca
Upton
Baton Rouge
Darsbury
Hamburg
Moscow
Novosibirsk
Tsukuba
Nishi Harima
Pohang
Paris
Villigen
Grenoble
Trieste
Barcelona
Shanghai
Yerevan
Allaan
Campinas
Melbourne
8
7
6
5
4
3
2
1
0
GeV

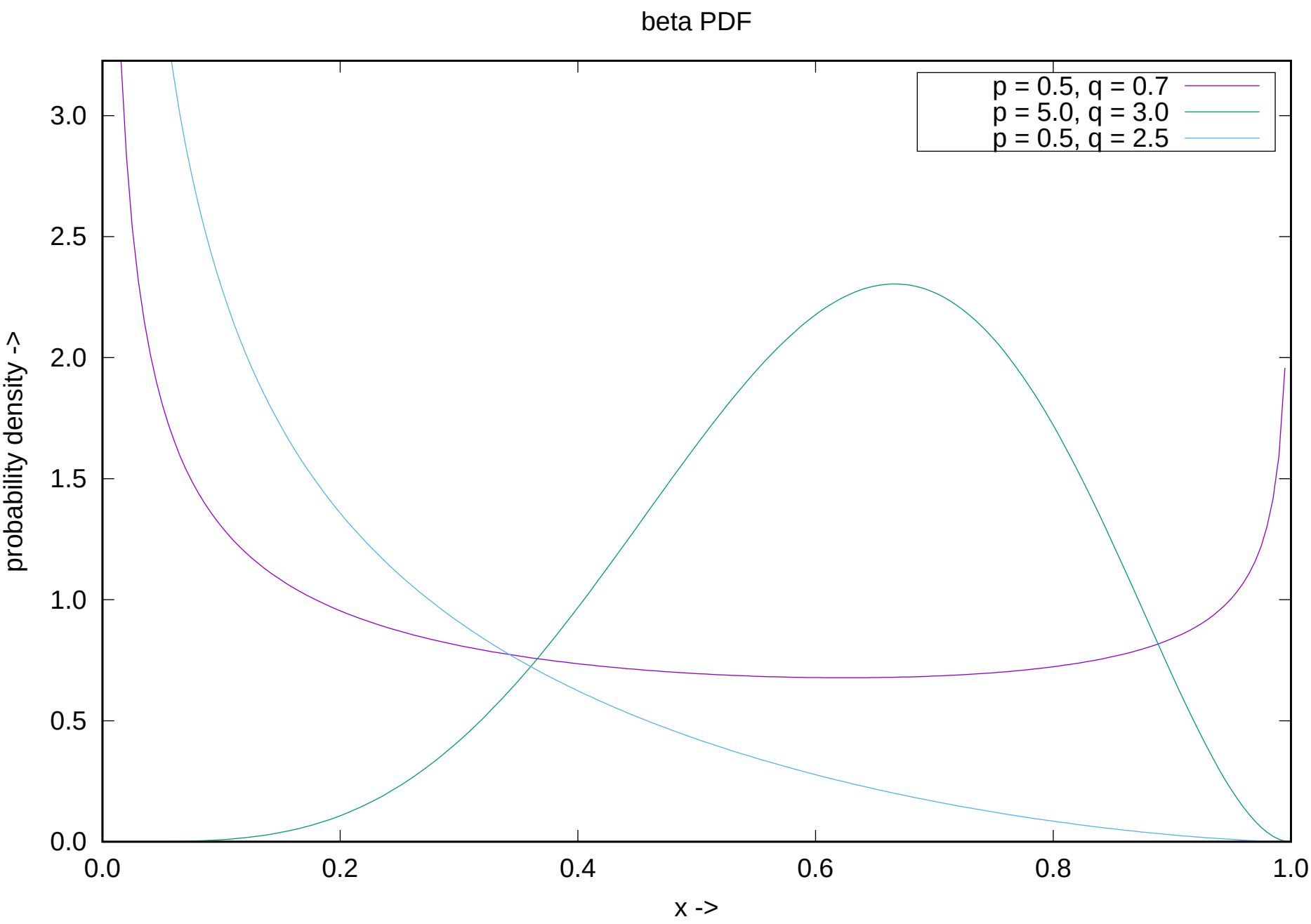

Labels with hidden line removal
Ithaca
Upton
Daresbury
Lund
Hamburg
Paris
Villigen
Grenoble
Trieste
Barcelona
Moscow
Novosibirsk
Tsukuba
Nishi Harima
Pohang
Shanghai
Yerevan
Allaan
8
7
6
5
4
3
2
1
0
GeV

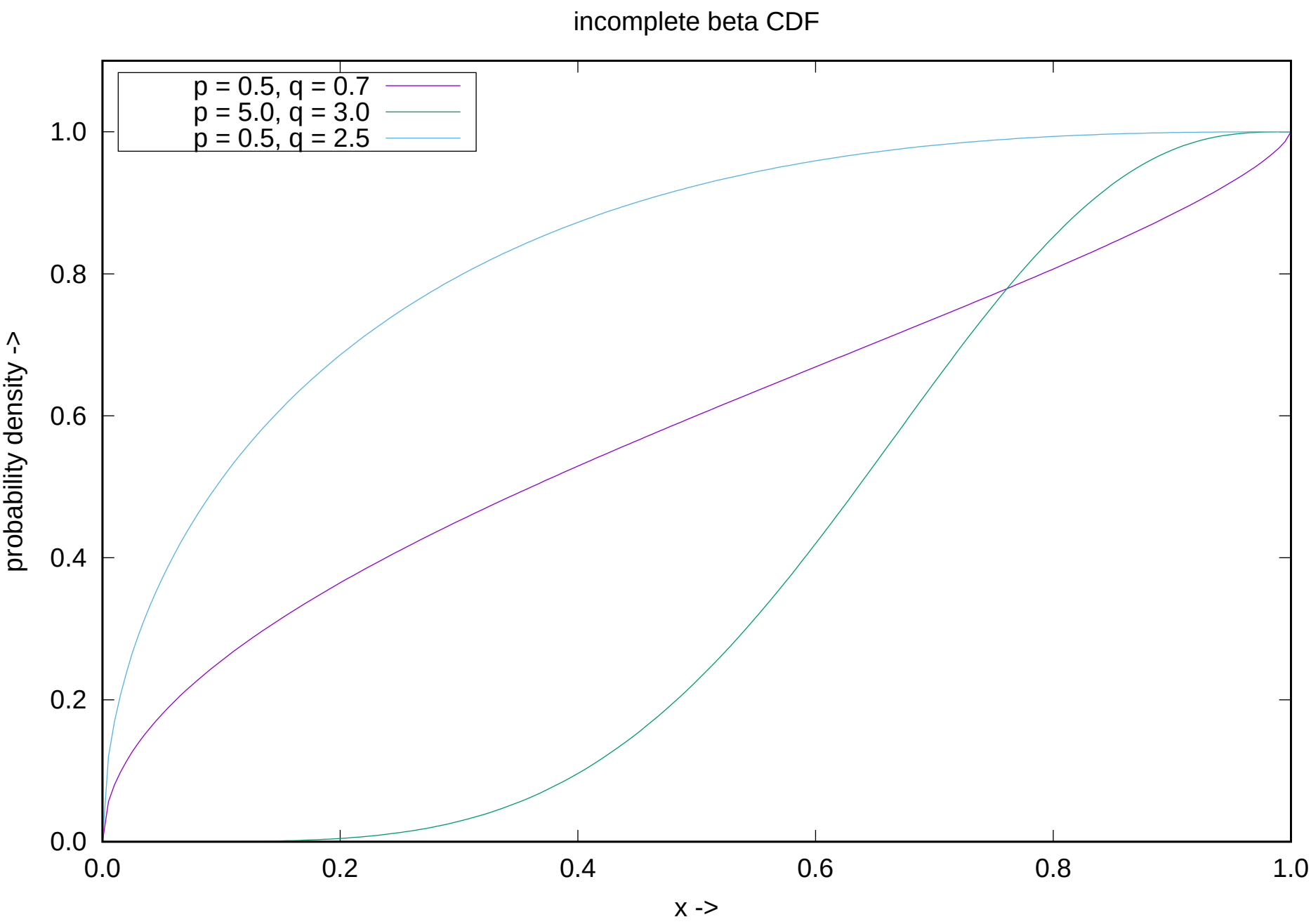

gamma function, very useful function for probability
gamma(x)
x
10
5
0
-5
-10
-4
-2
0
2
4

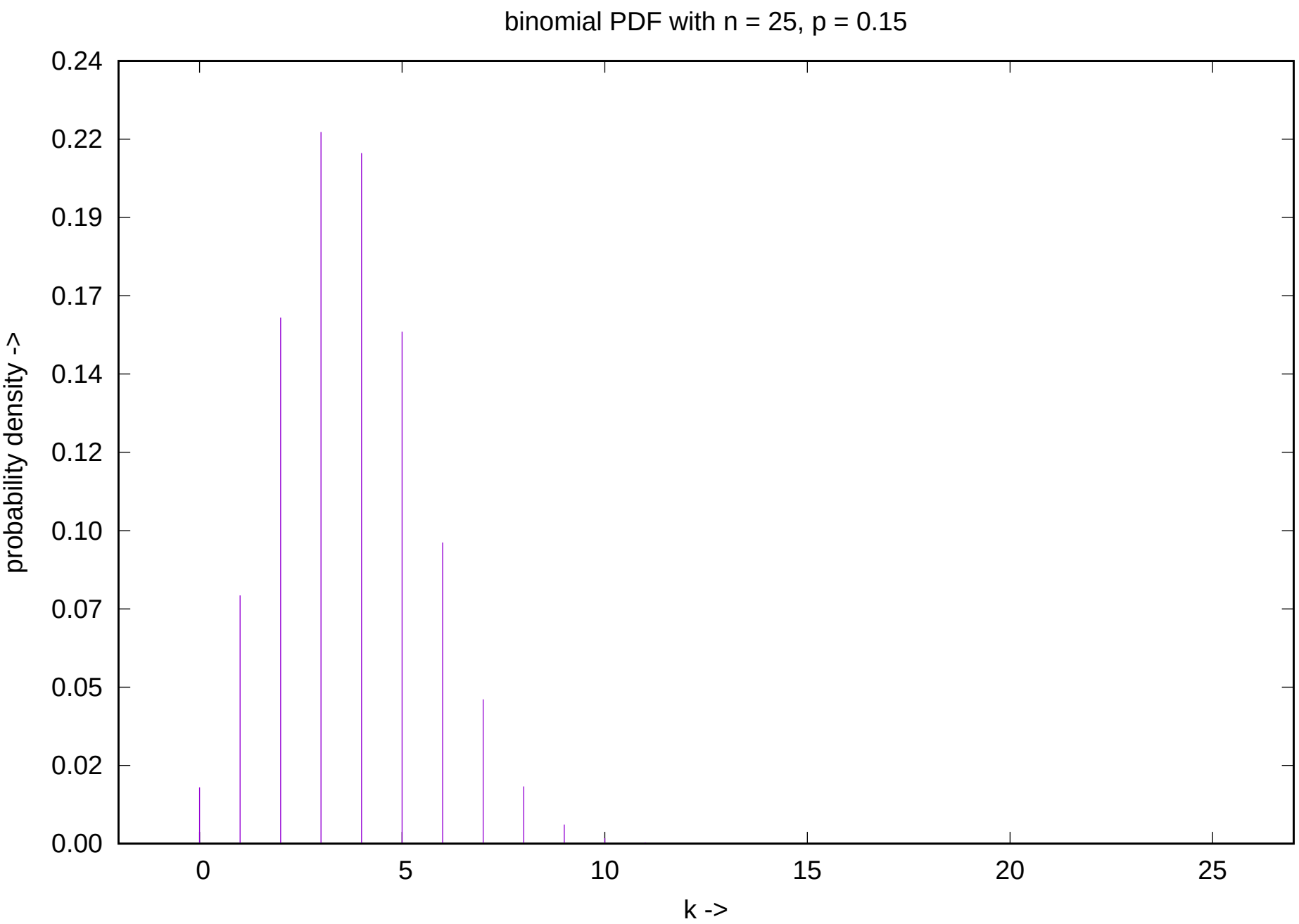

log gamma function, similarly very useful function
lgamma(x)
x
4
2
0
-2
-4
-4
-2
0
2
4

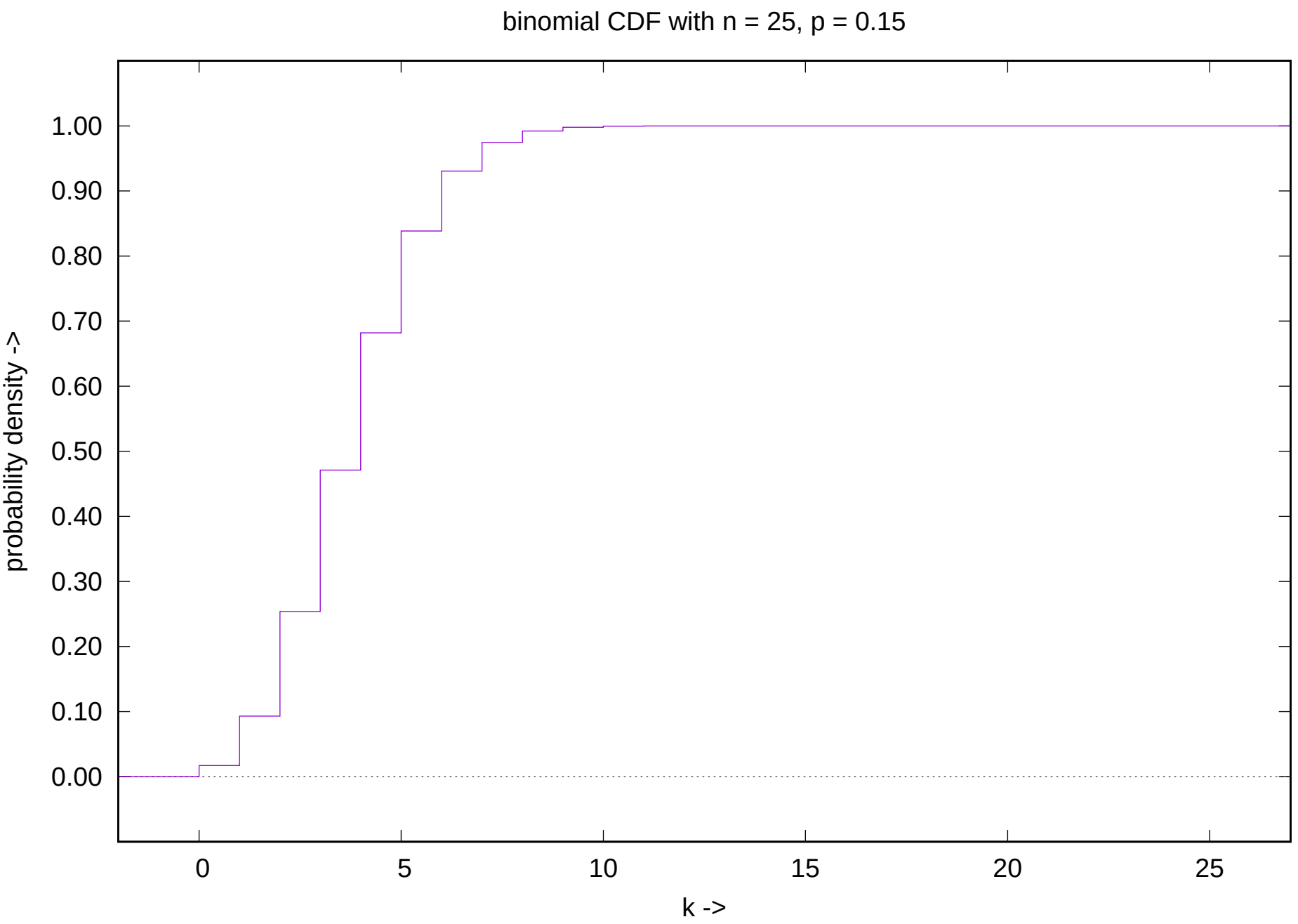

arcsin PDF with r = 2.0
probability density ->
x ->
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

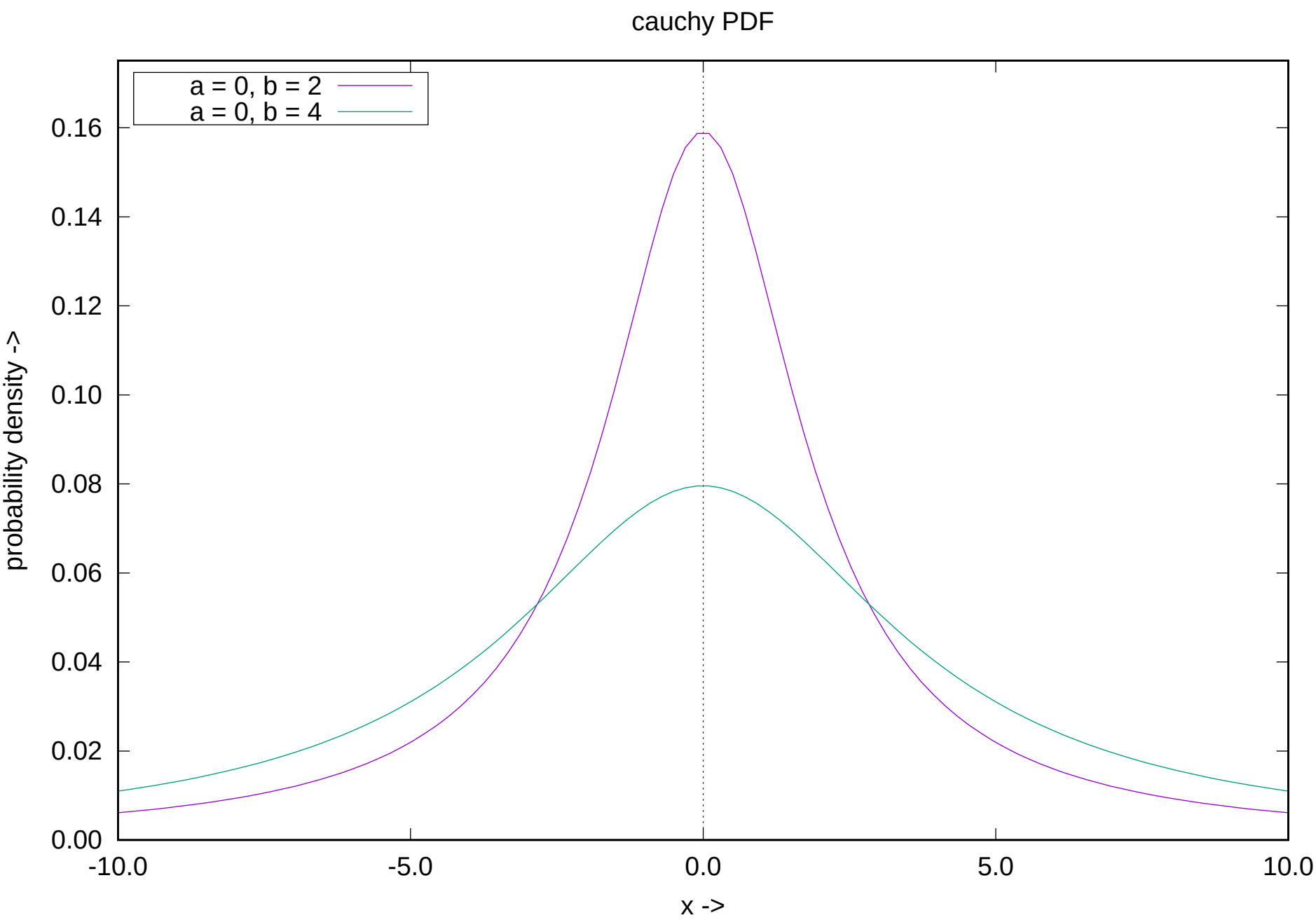

arcsin CDF with r = 2.0
probability density ->
x ->
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

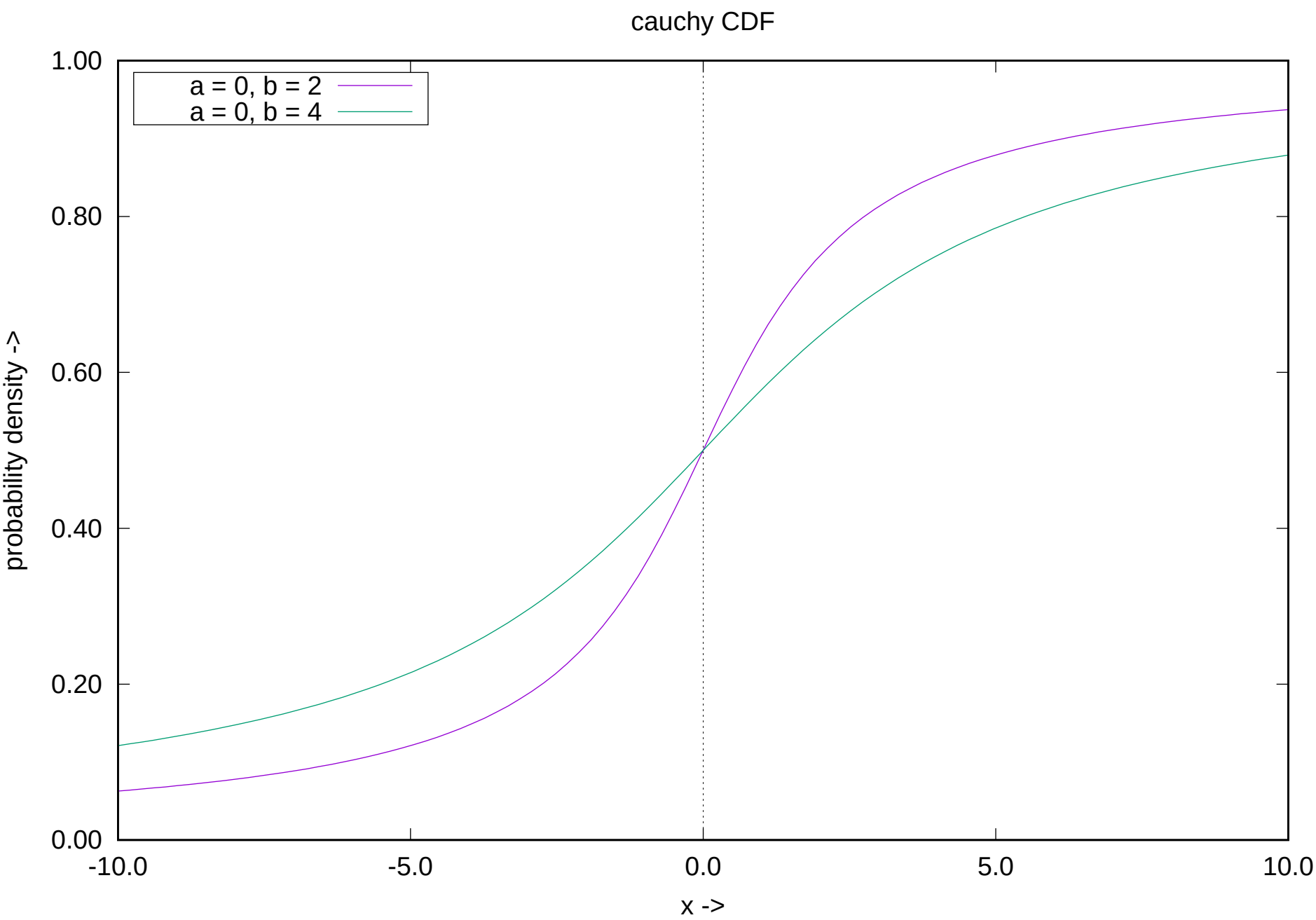

beta PDF
p = 0.5, q = 0.7
p = 5.0, q = 3.0
p = 0.5, q = 2.5
probability density ->
x ->
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
1.0

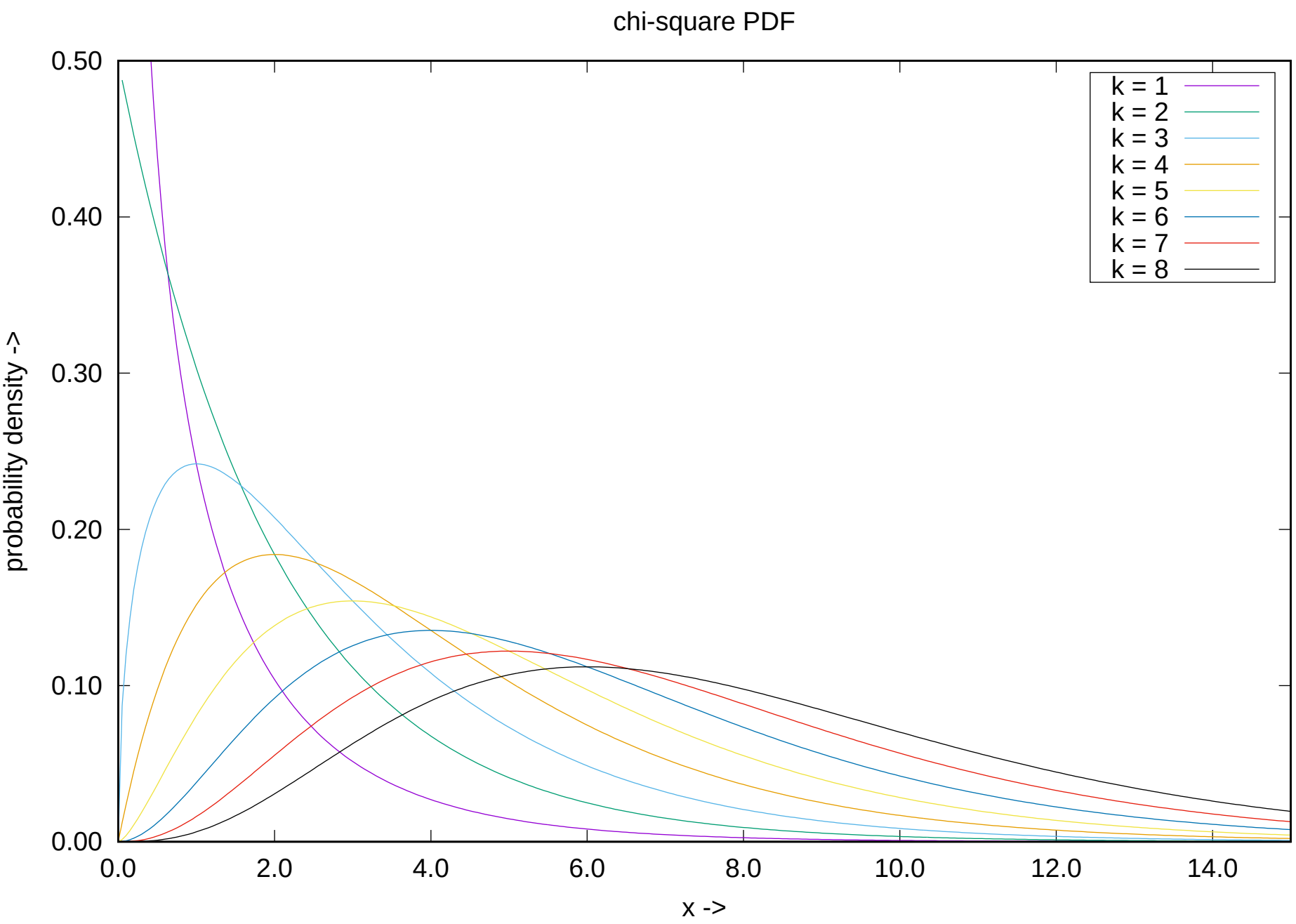

incomplete beta CDF
p = 0.5, q = 0.7
p = 5.0, q = 3.0
p = 0.5, q = 2.5
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0

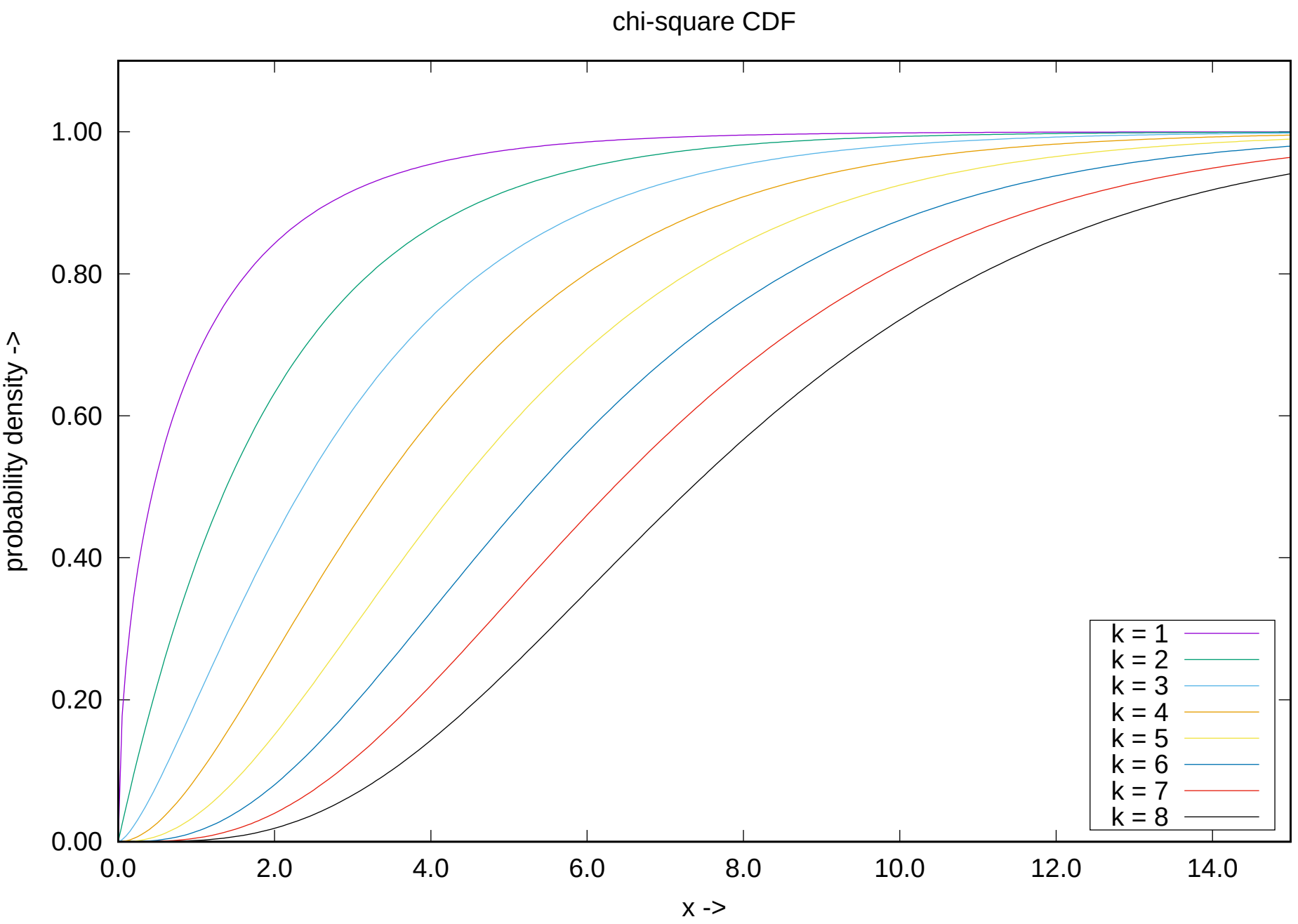

binomial PDF with n = 25, p = 0.15
probability density ->
0.24
0.22
0.19
0.17
0.14
0.12
0.10
0.07
0.05
0.02
0.00
0
5
10
15
20
25
k ->

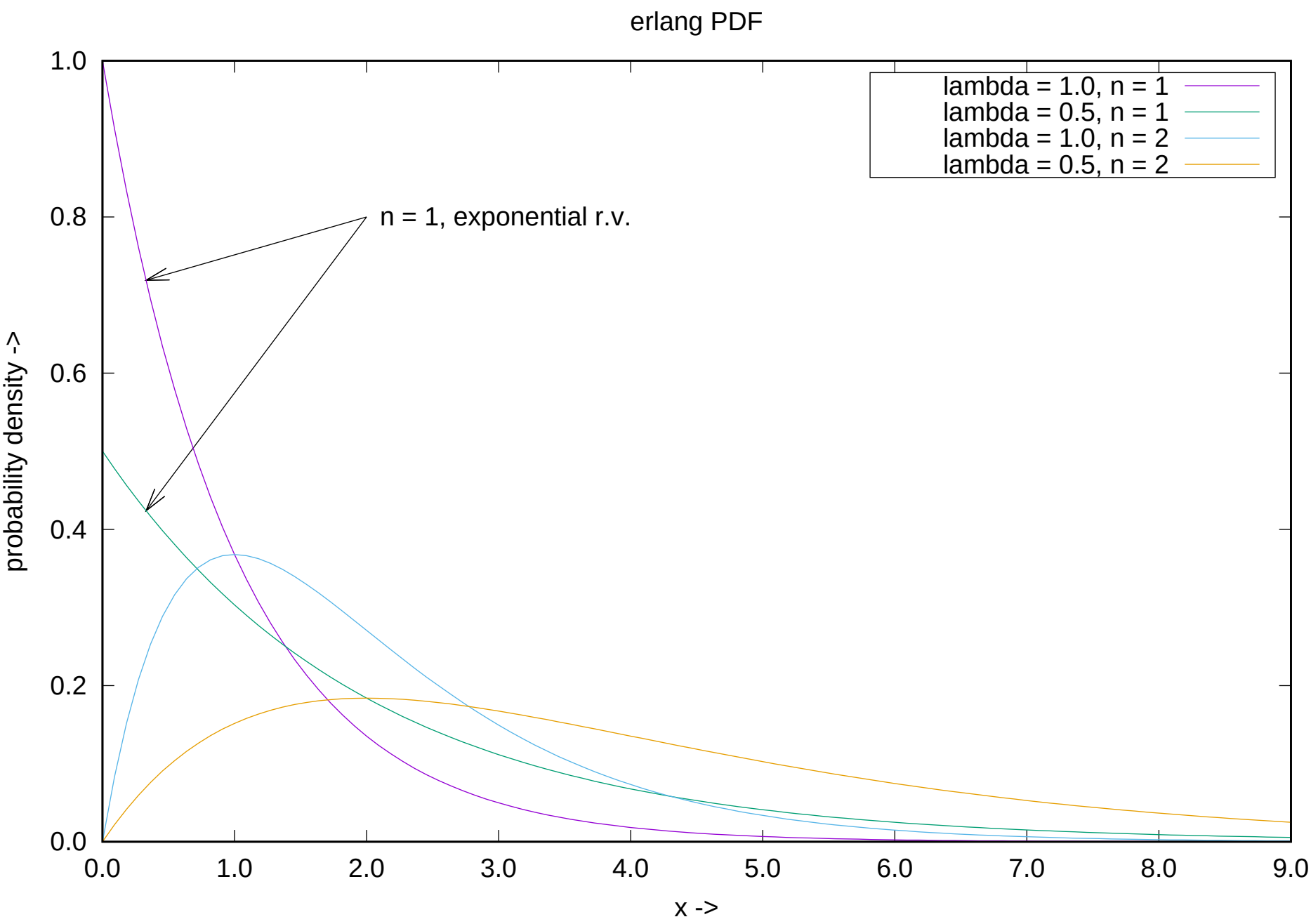

binomial CDF with n = 25, p = 0.15
probability density ->
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0
5
10
15
20
25
k ->

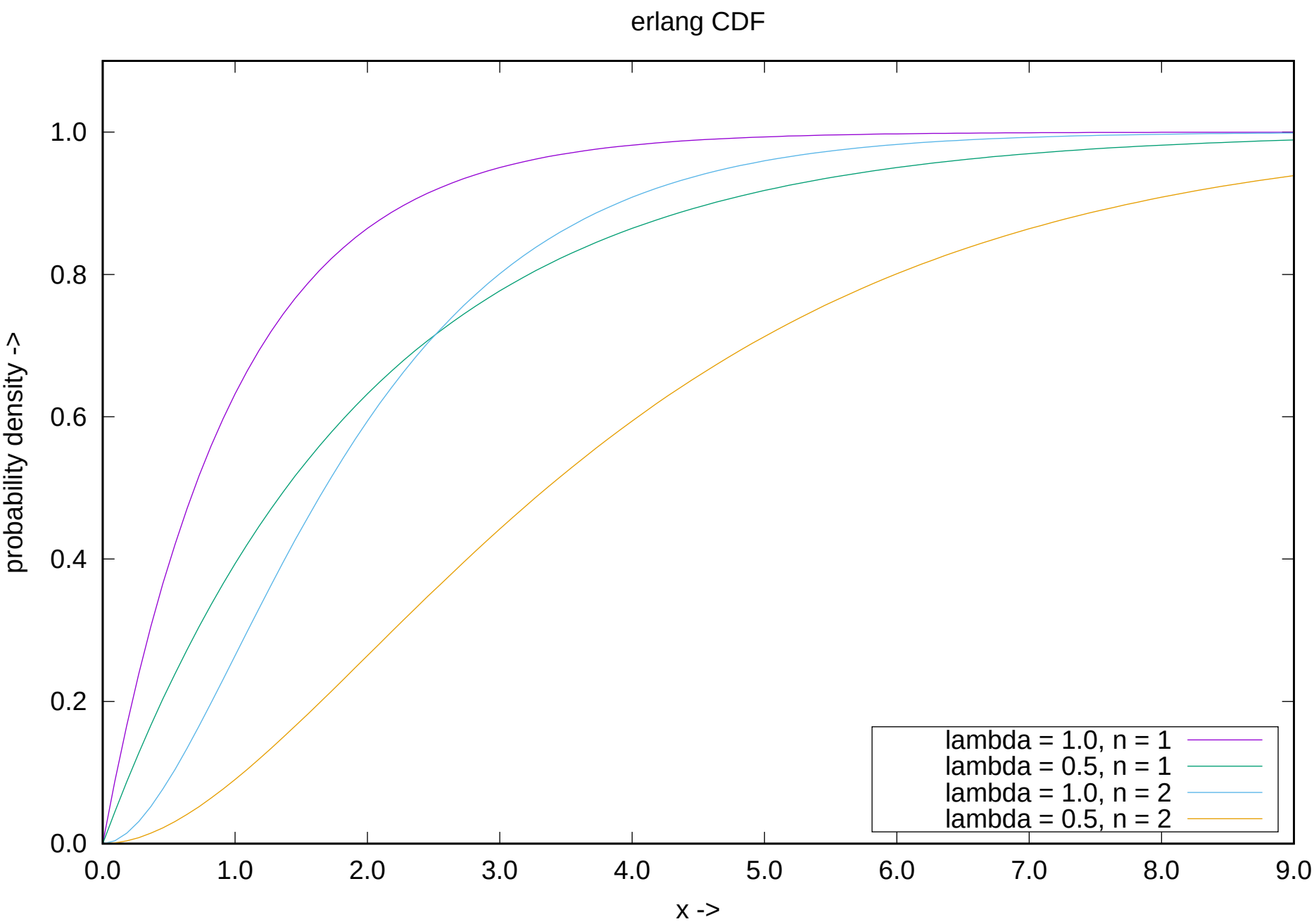

cauchy PDF
a = 0, b = 2
a = 0, b = 4
probability density ->
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
-10.0
-5.0
0.0
5.0
10.0
x ->

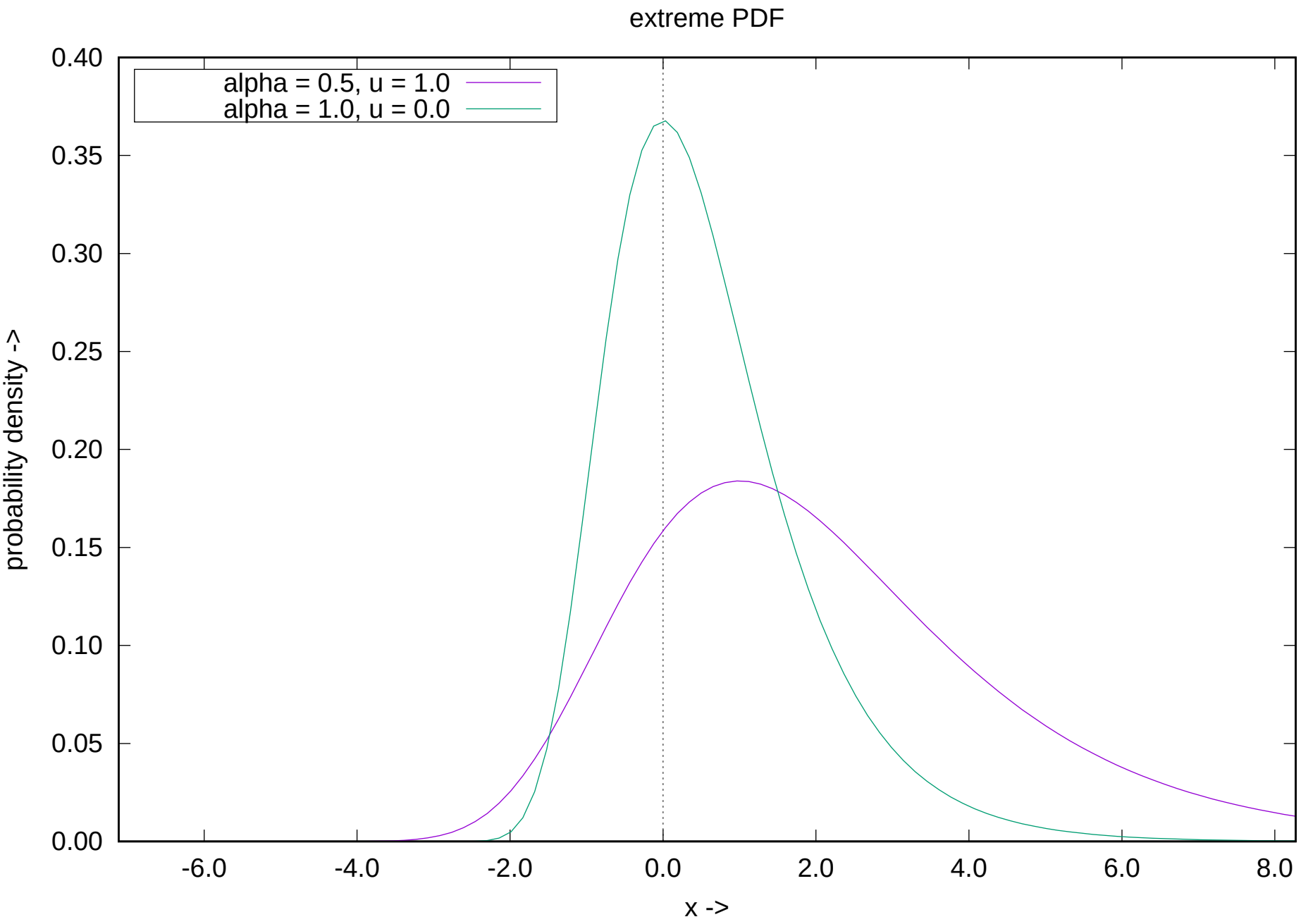

cauchy CDF
a = 0, b = 2
a = 0, b = 4
1.00
0.80
0.60
0.40
0.20
0.00
probability density ->
-10.0
-5.0
0.0
5.0
10.0
x ->

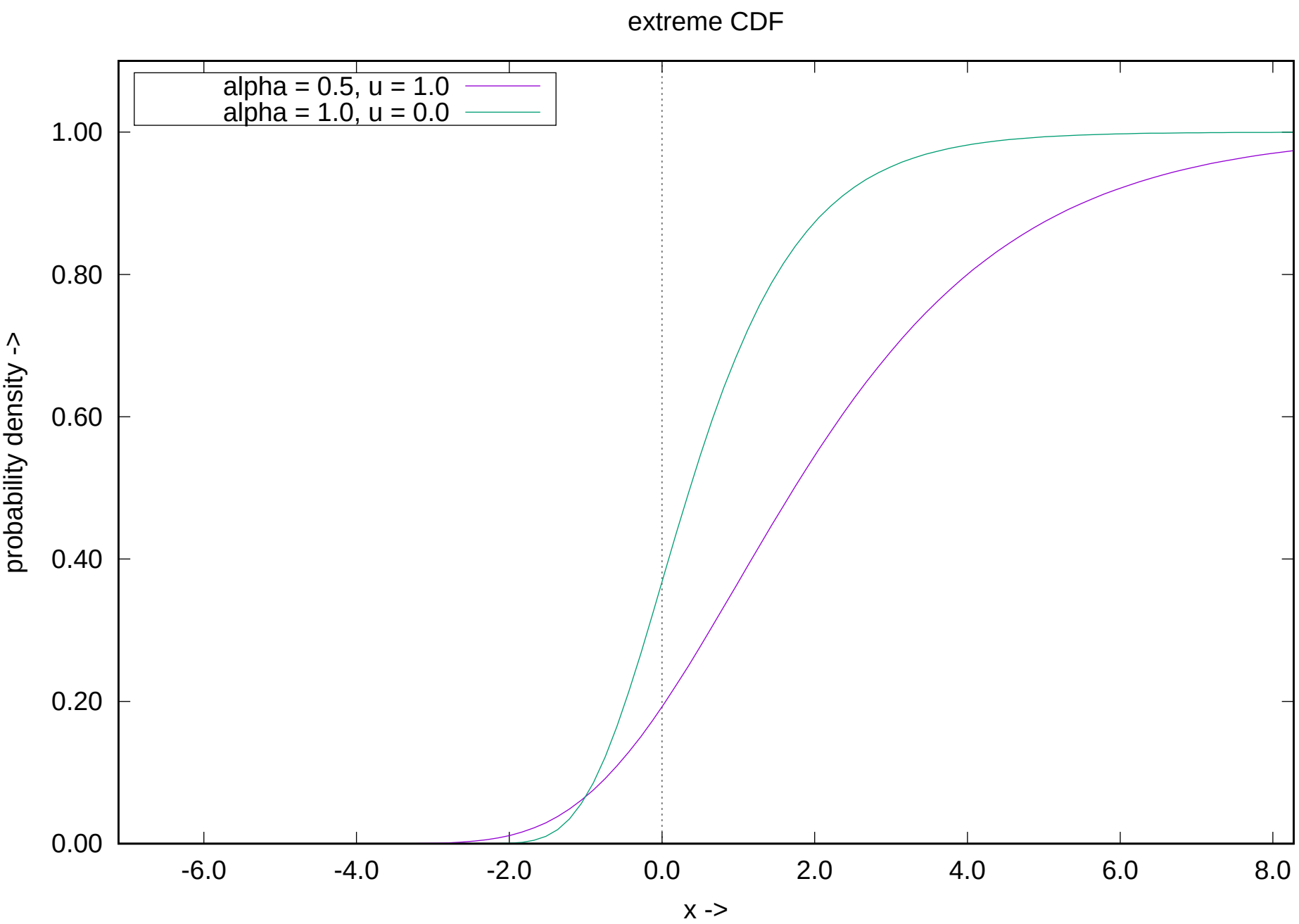

chi-square PDF
probability density ->
x ->
0.00
0.10
0.20
0.30
0.40
0.50
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
k = 1
k = 2
k = 3
k = 4
k = 5
k = 6
k = 7
k = 8

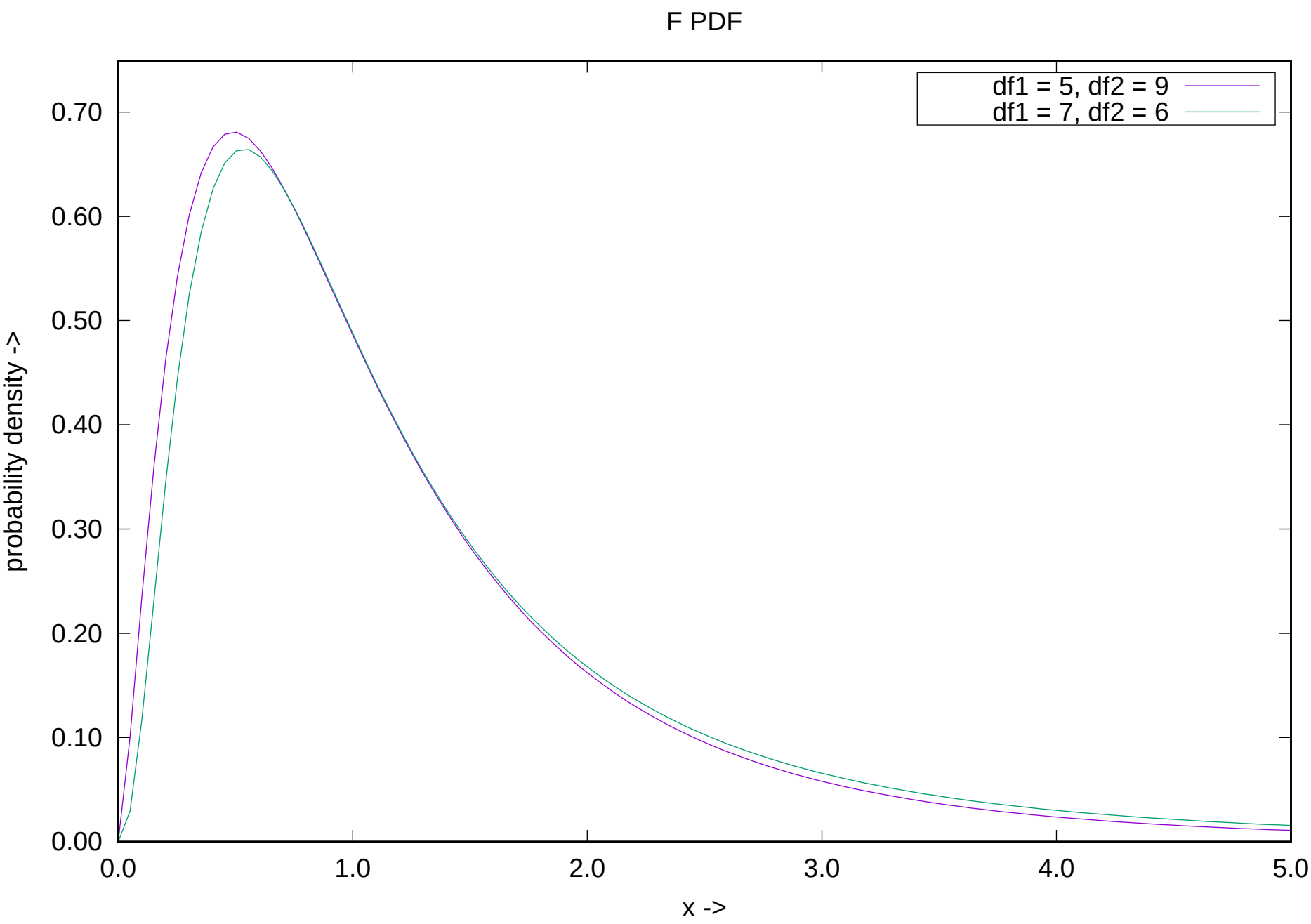

chi-square CDF
probability density ->
x ->
0.00
0.20
0.40
0.60
0.80
1.00
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
k = 1
k = 2
k = 3
k = 4
k = 5
k = 6
k = 7
k = 8

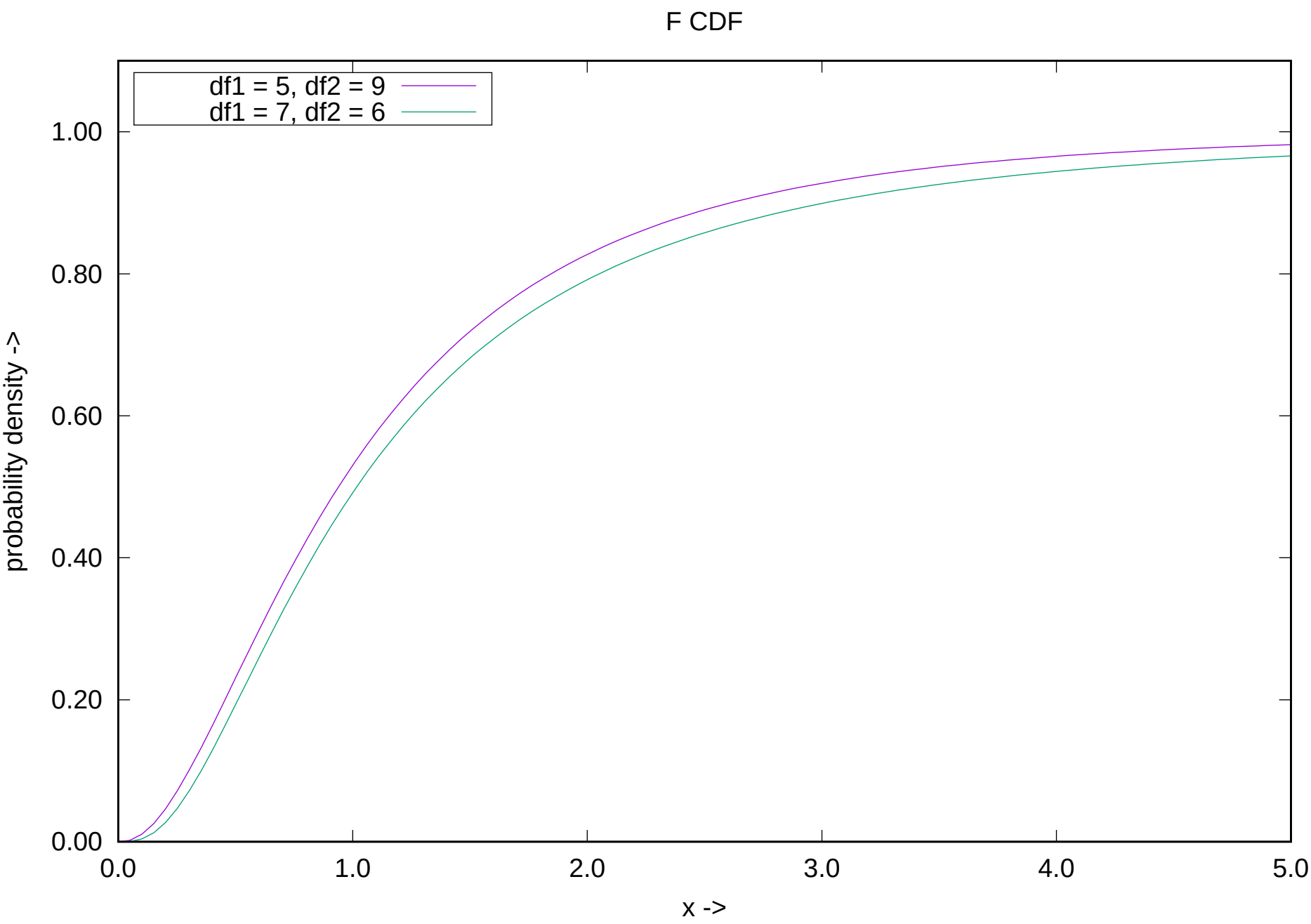

erlang PDF
lambda = 1.0, n = 1
lambda = 0.5, n = 1
lambda = 1.0, n = 2
lambda = 0.5, n = 2
n = 1, exponential r.v.
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0

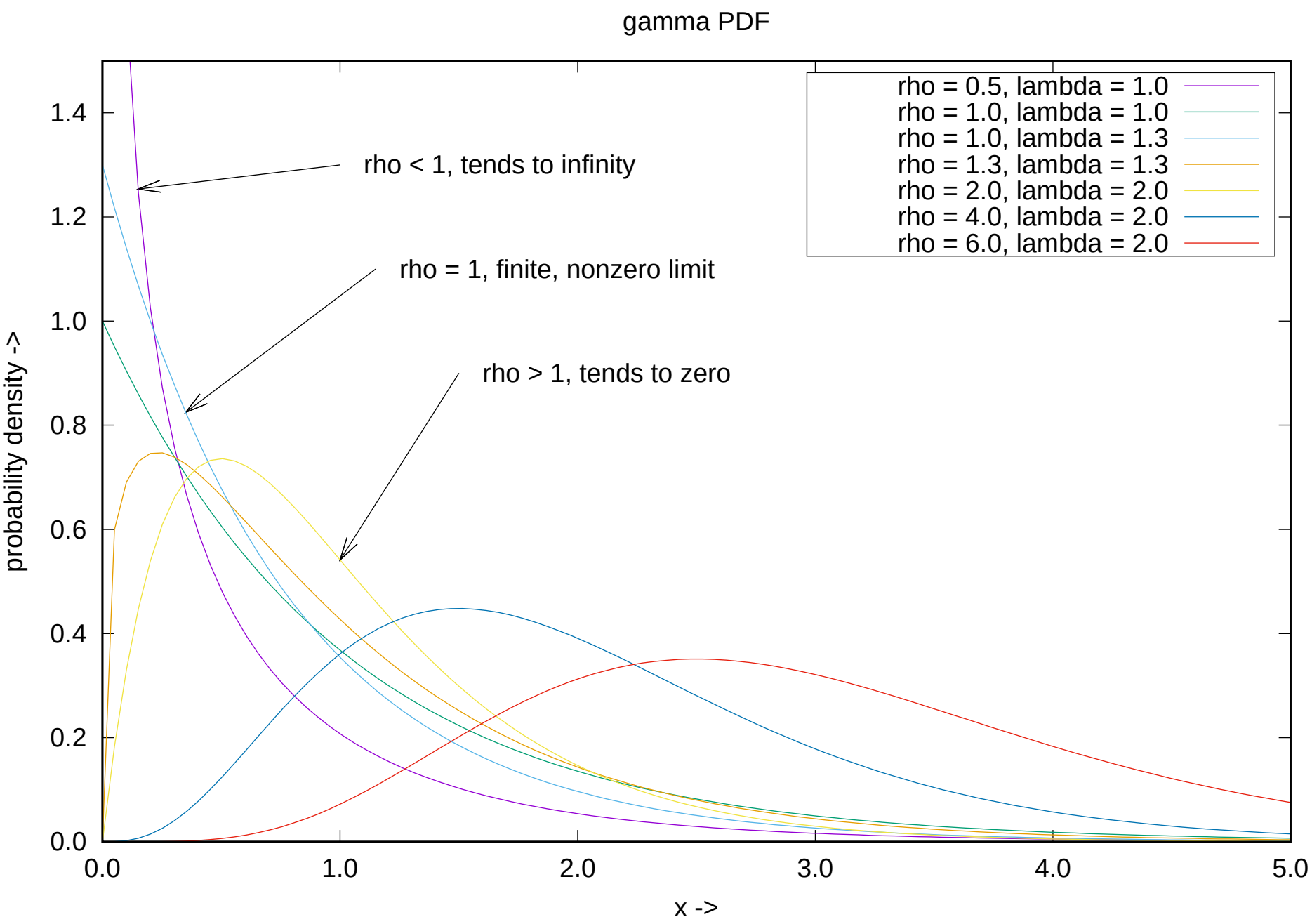

erlang CDF
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
lambda = 1.0, n = 1
lambda = 0.5, n = 1
lambda = 1.0, n = 2
lambda = 0.5, n = 2

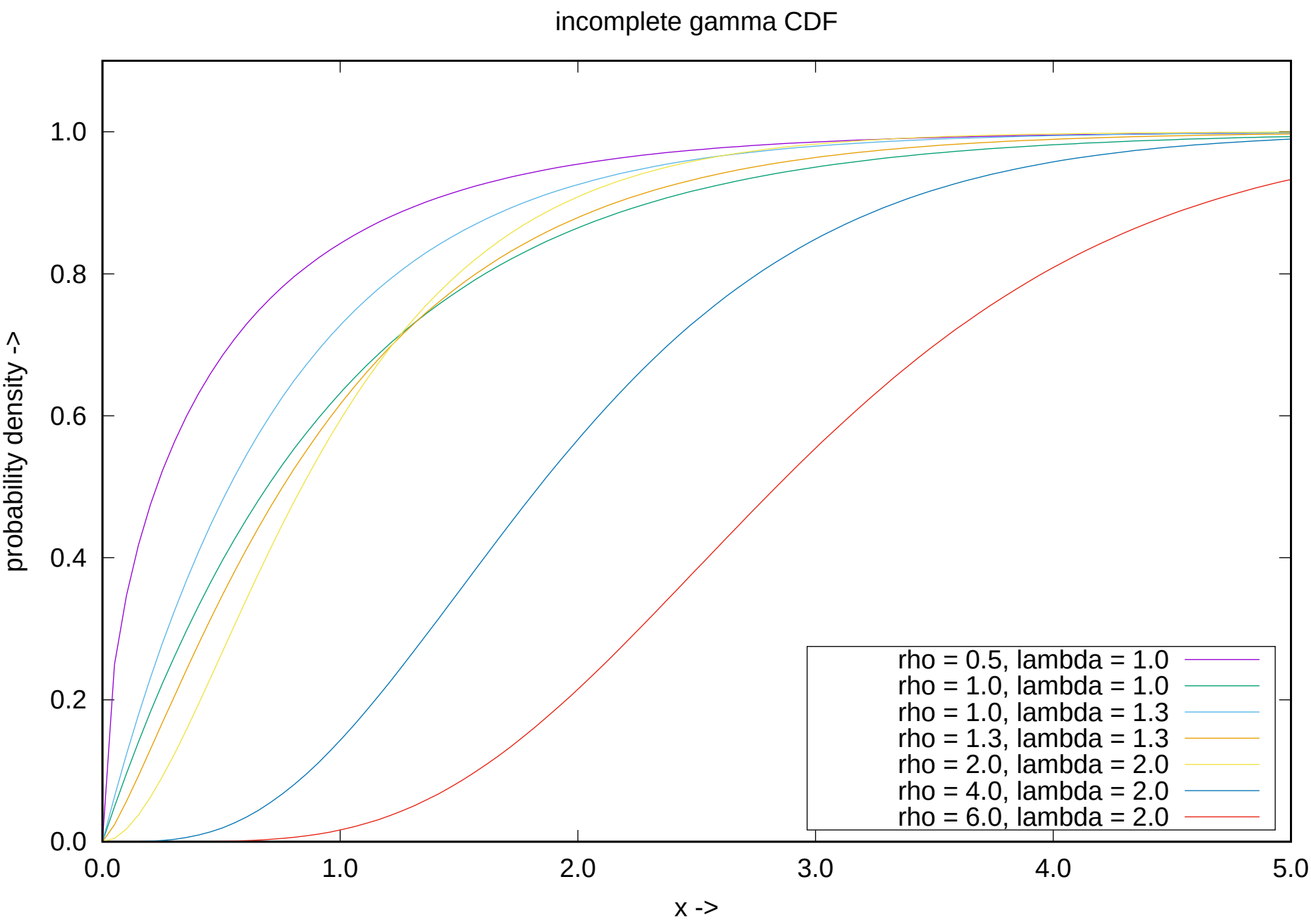

extreme PDF
alpha = 0.5, u = 1.0
alpha = 1.0, u = 0.0
probability density ->
x ->
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-6.0
-4.0
-2.0
0.0
2.0
4.0
6.0
8.0

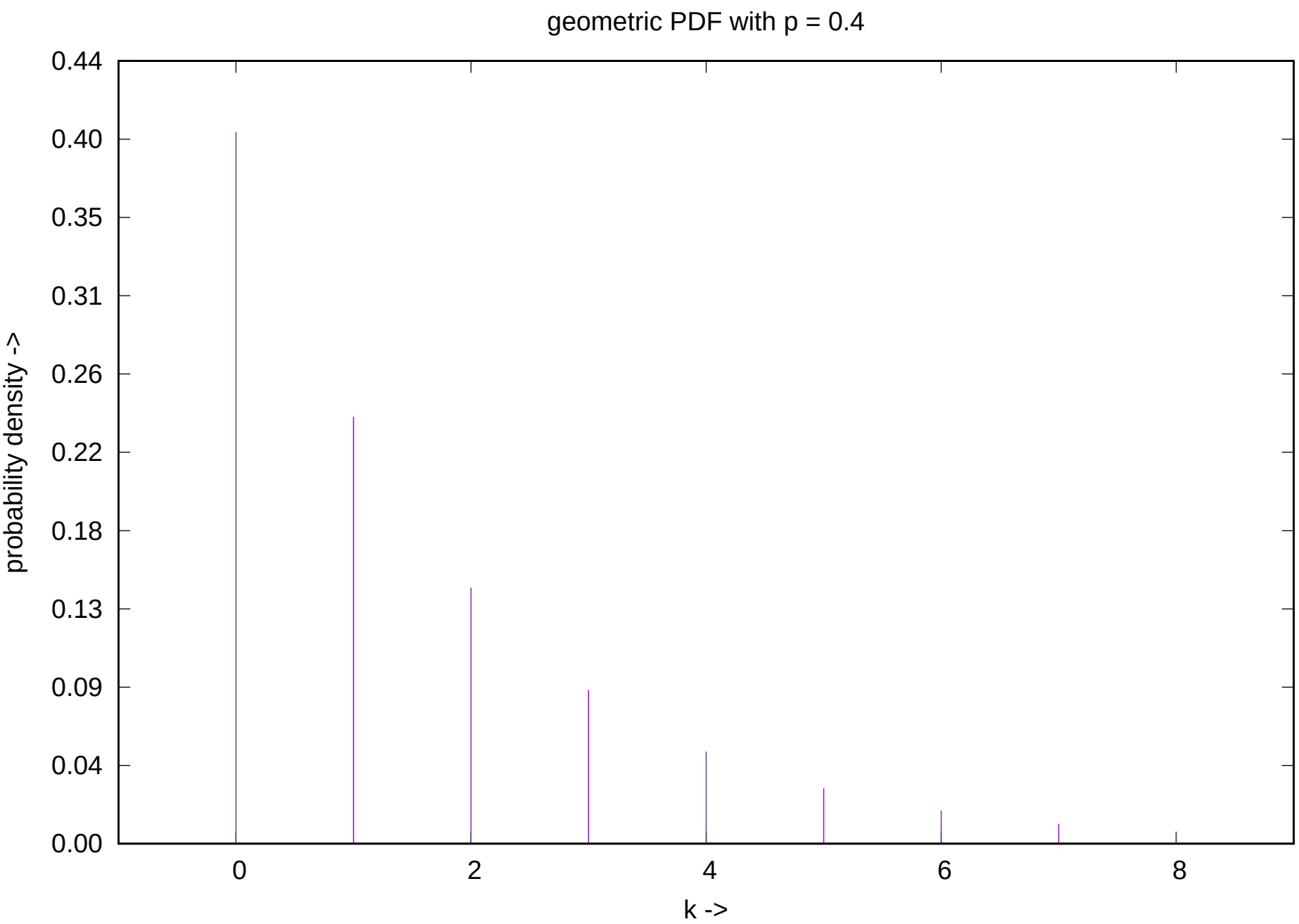

extreme CDF
alpha = 0.5, u = 1.0
alpha = 1.0, u = 0.0
probability density ->
1.00
0.80
0.60
0.40
0.20
0.00
-6.0
-4.0
-2.0
0.0
2.0
4.0
6.0
8.0
x ->

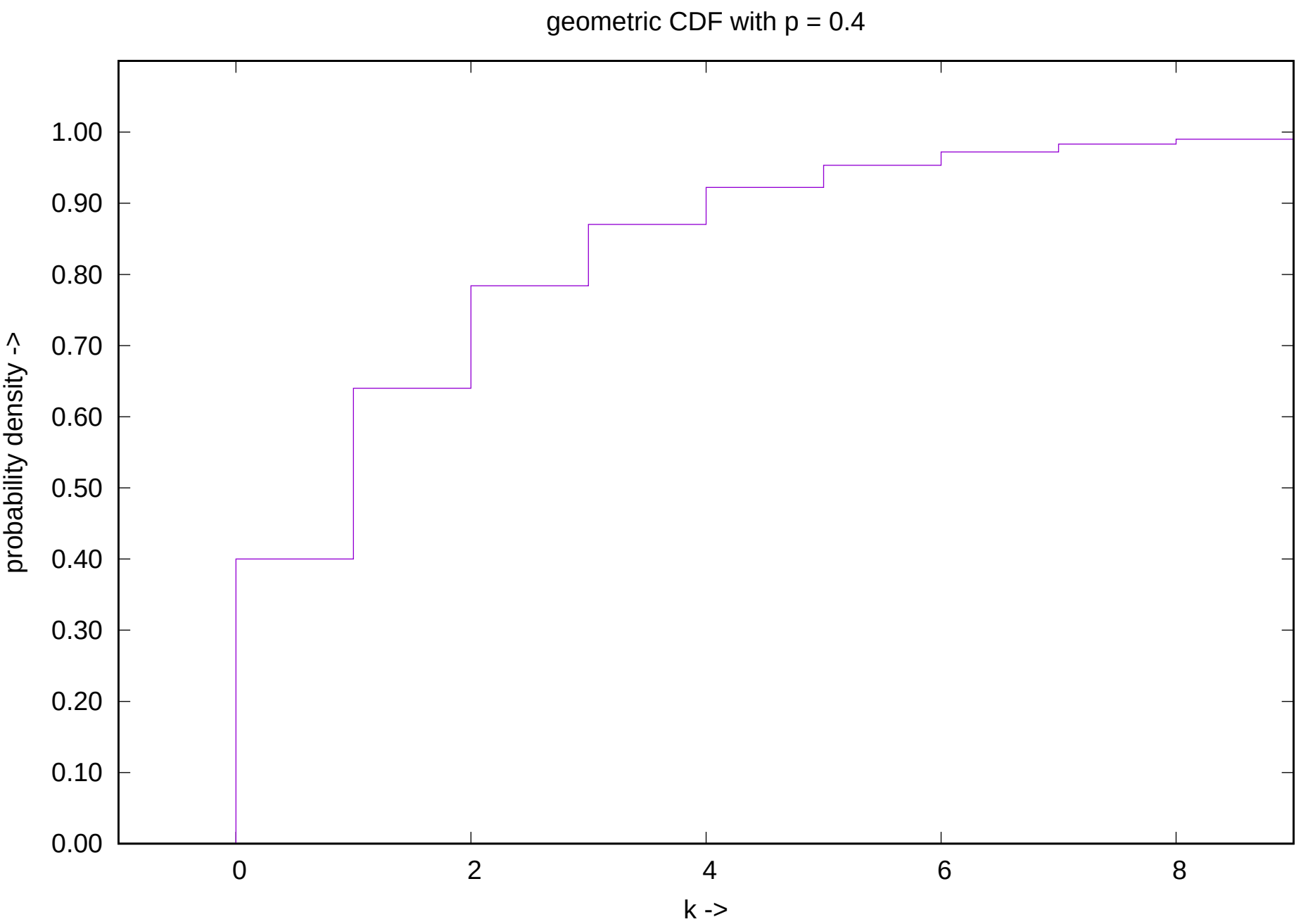

F PDF
df1 = 5, df2 = 9
df1 = 7, df2 = 6
probability density ->
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.0
1.0
2.0
3.0
4.0
5.0
x ->

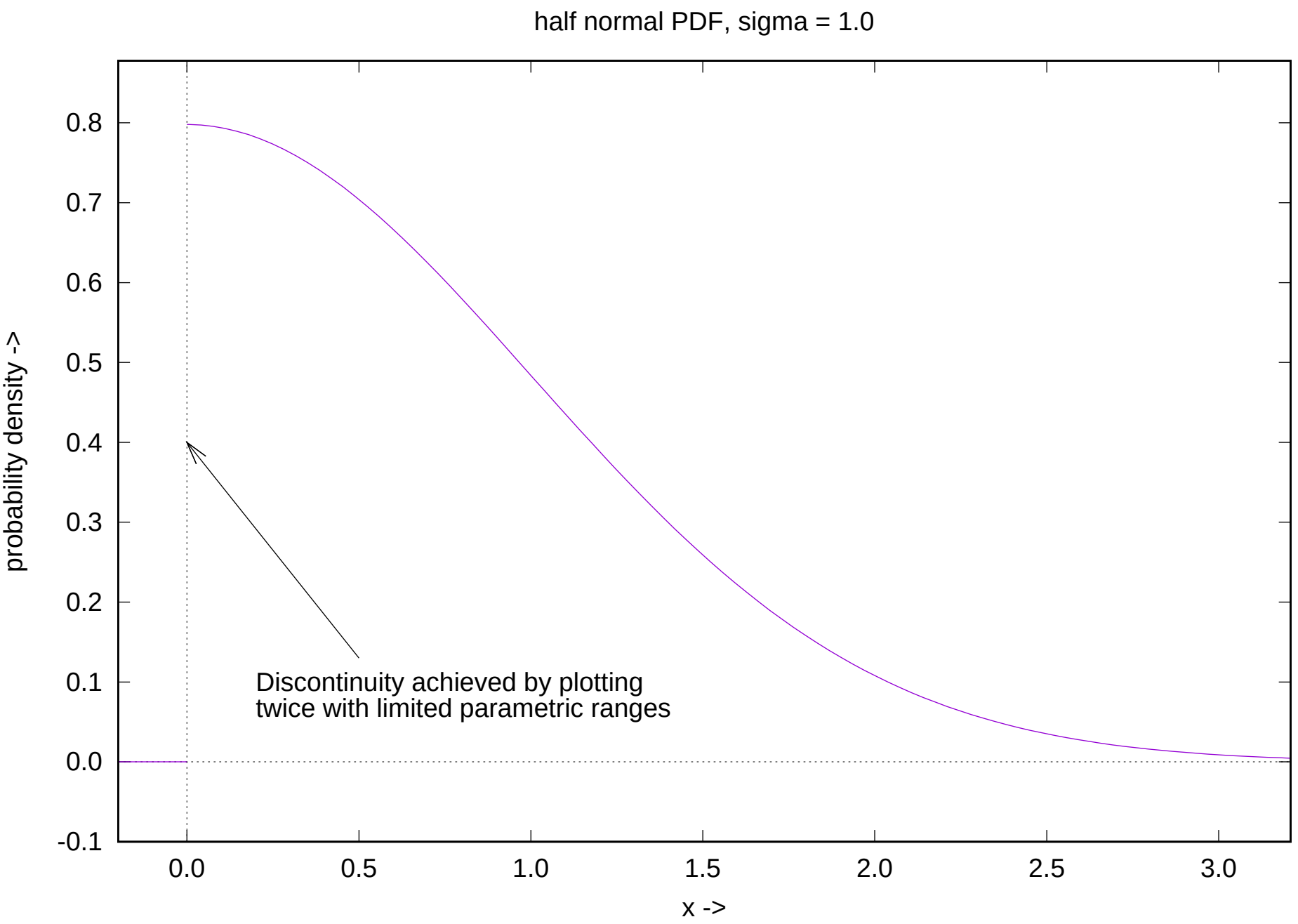

F CDF
df1 = 5, df2 = 9
df1 = 7, df2 = 6
probability density ->
x ->
0.00
0.20
0.40
0.60
0.80
1.00
0.0
1.0
2.0
3.0
4.0
5.0

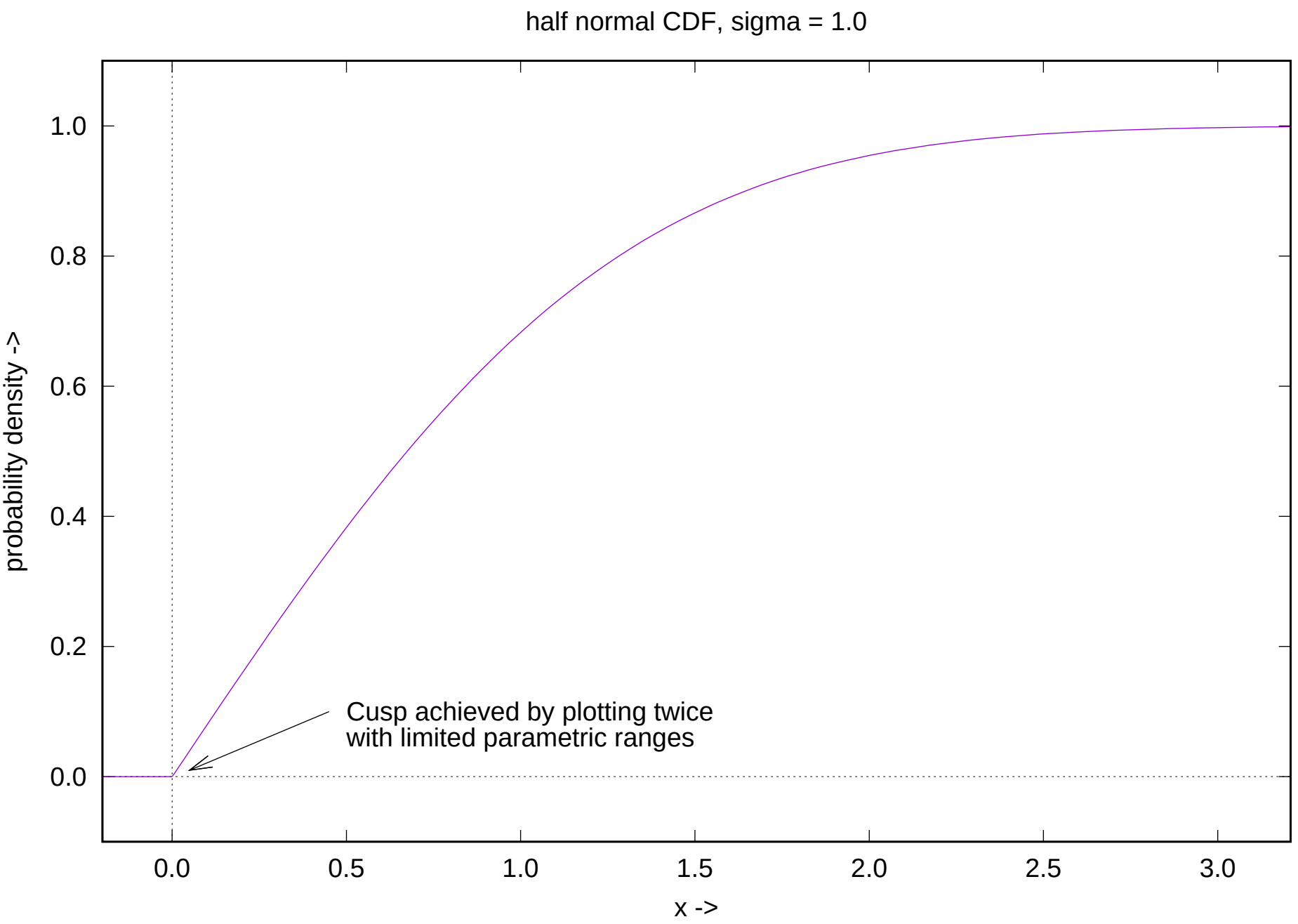

gamma PDF
rho = 0.5, lambda = 1.0
rho = 1.0, lambda = 1.0
rho = 1.0, lambda = 1.3
rho = 1.3, lambda = 1.3
rho = 2.0, lambda = 2.0
rho = 4.0, lambda = 2.0
rho = 6.0, lambda = 2.0
rho < 1, tends to infinity
rho = 1, finite, nonzero limit
rho > 1, tends to zero
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0.0
1.0
2.0
3.0
4.0
5.0

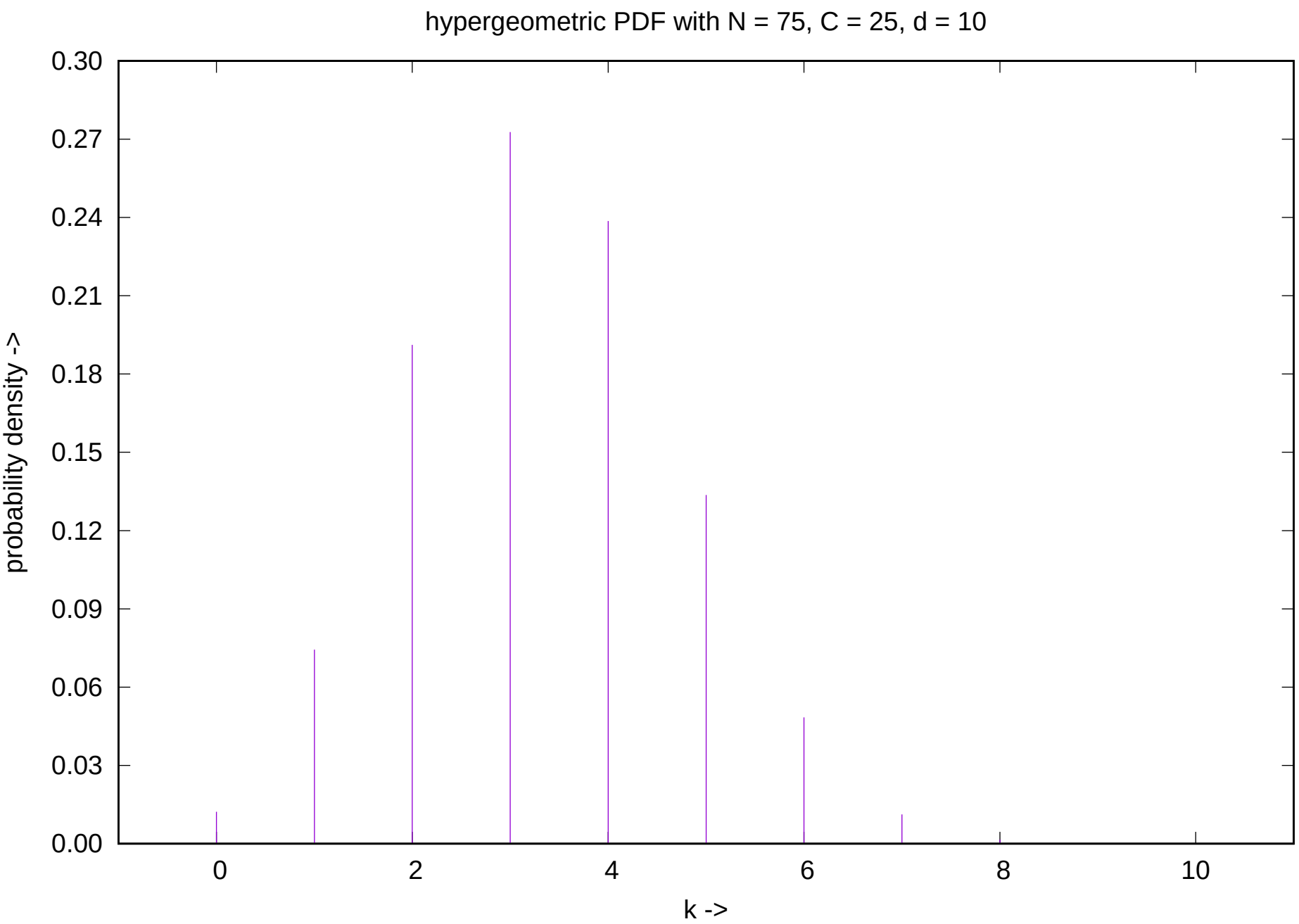

incomplete gamma CDF
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.0
1.0
2.0
3.0
4.0
5.0
rho = 0.5, lambda = 1.0
rho = 1.0, lambda = 1.0
rho = 1.0, lambda = 1.3
rho = 1.3, lambda = 1.3
rho = 2.0, lambda = 2.0
rho = 4.0, lambda = 2.0
rho = 6.0, lambda = 2.0

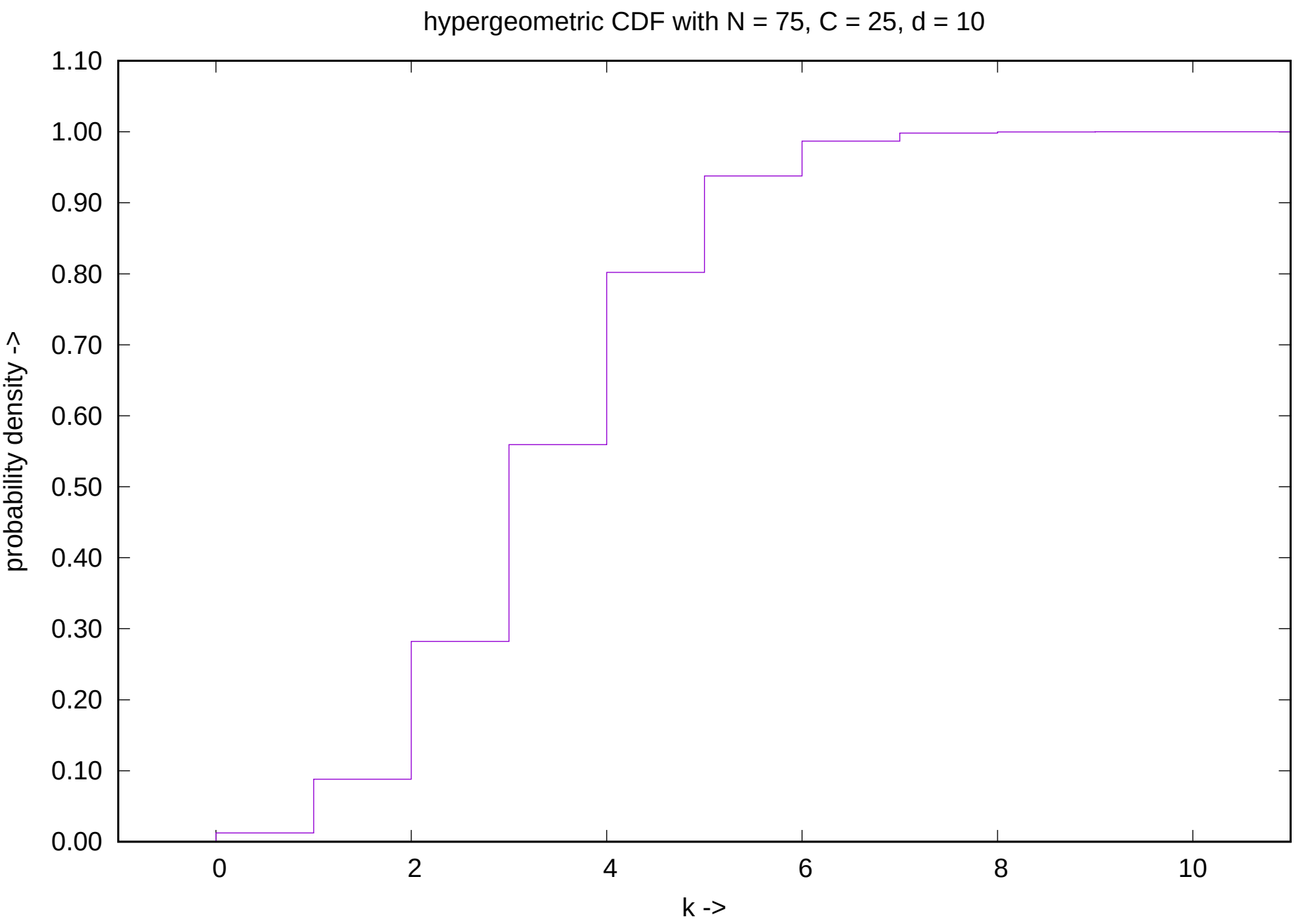

geometric PDF with p = 0.4
probability density ->
0.44
0.40
0.35
0.31
0.26
0.22
0.18
0.13
0.09
0.04
0.00
0
2
4
6
8
k ->

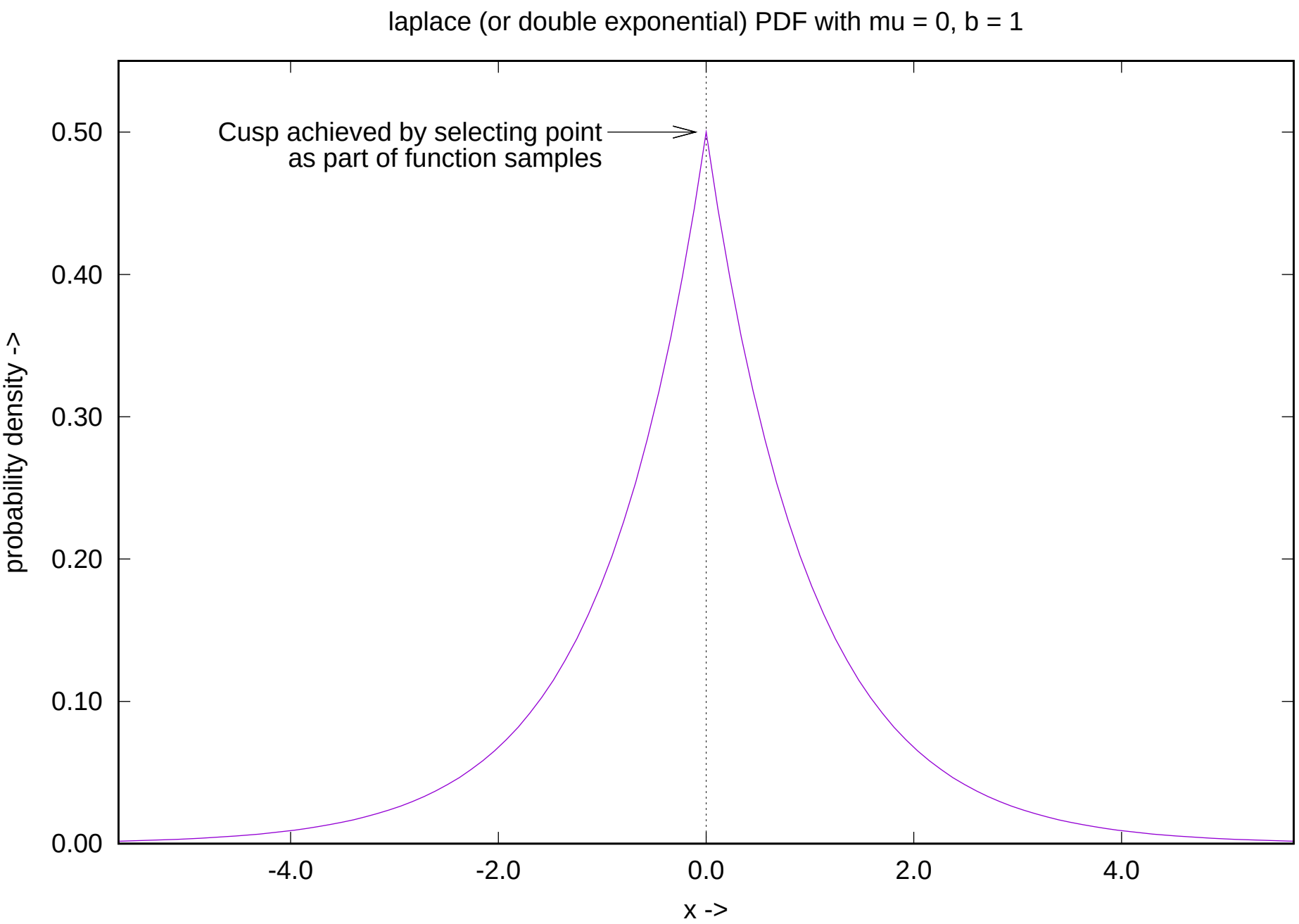

geometric CDF with p = 0.4
probability density ->
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0
2
4
6
8
k ->

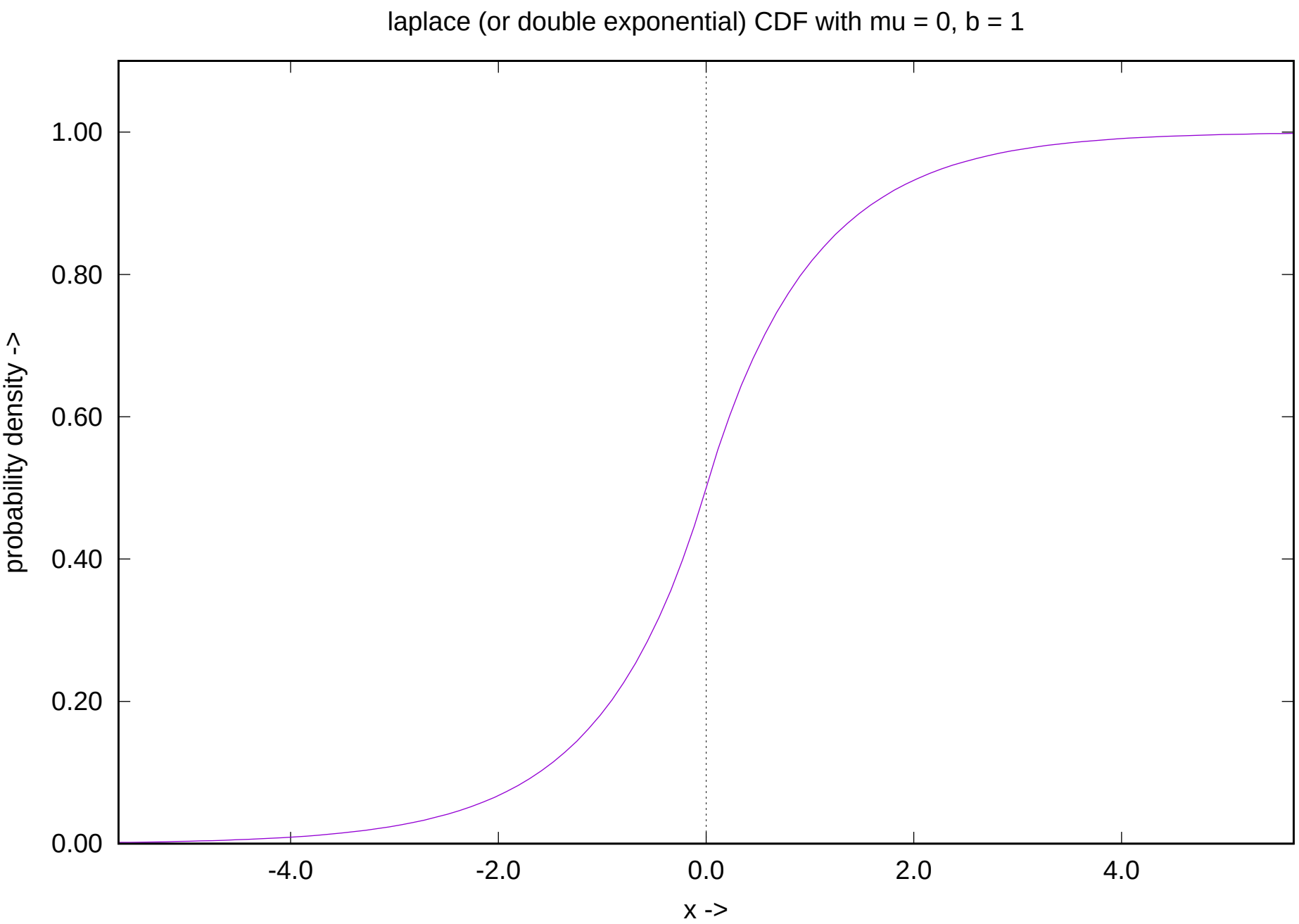

half normal PDF, sigma = 1.0
probability density ->
x ->
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Discontinuity achieved by plotting
twice with limited parametric ranges

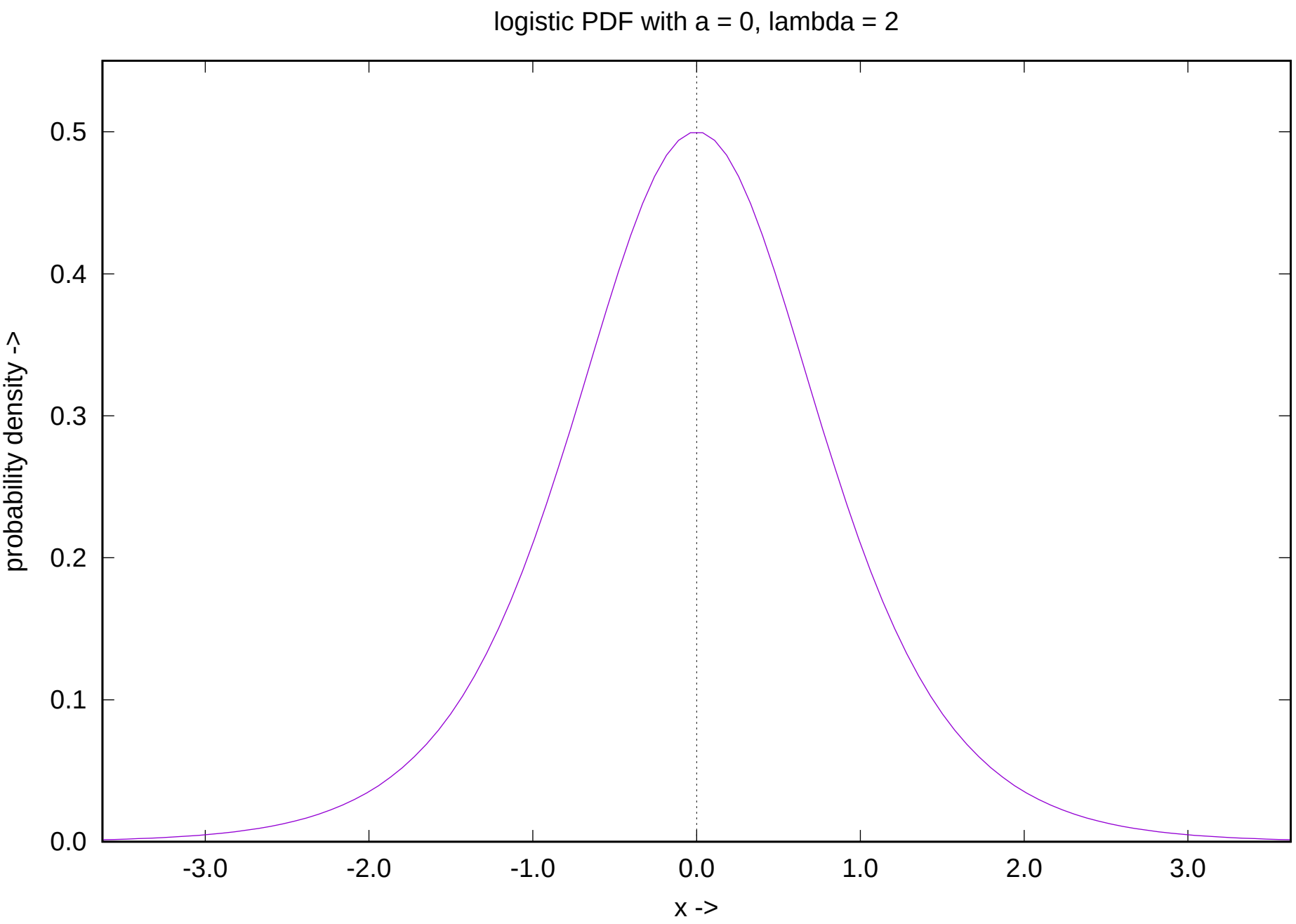

half normal CDF, sigma = 1.0
probability density ->
x ->
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Cusp achieved by plotting twice
with limited parametric ranges

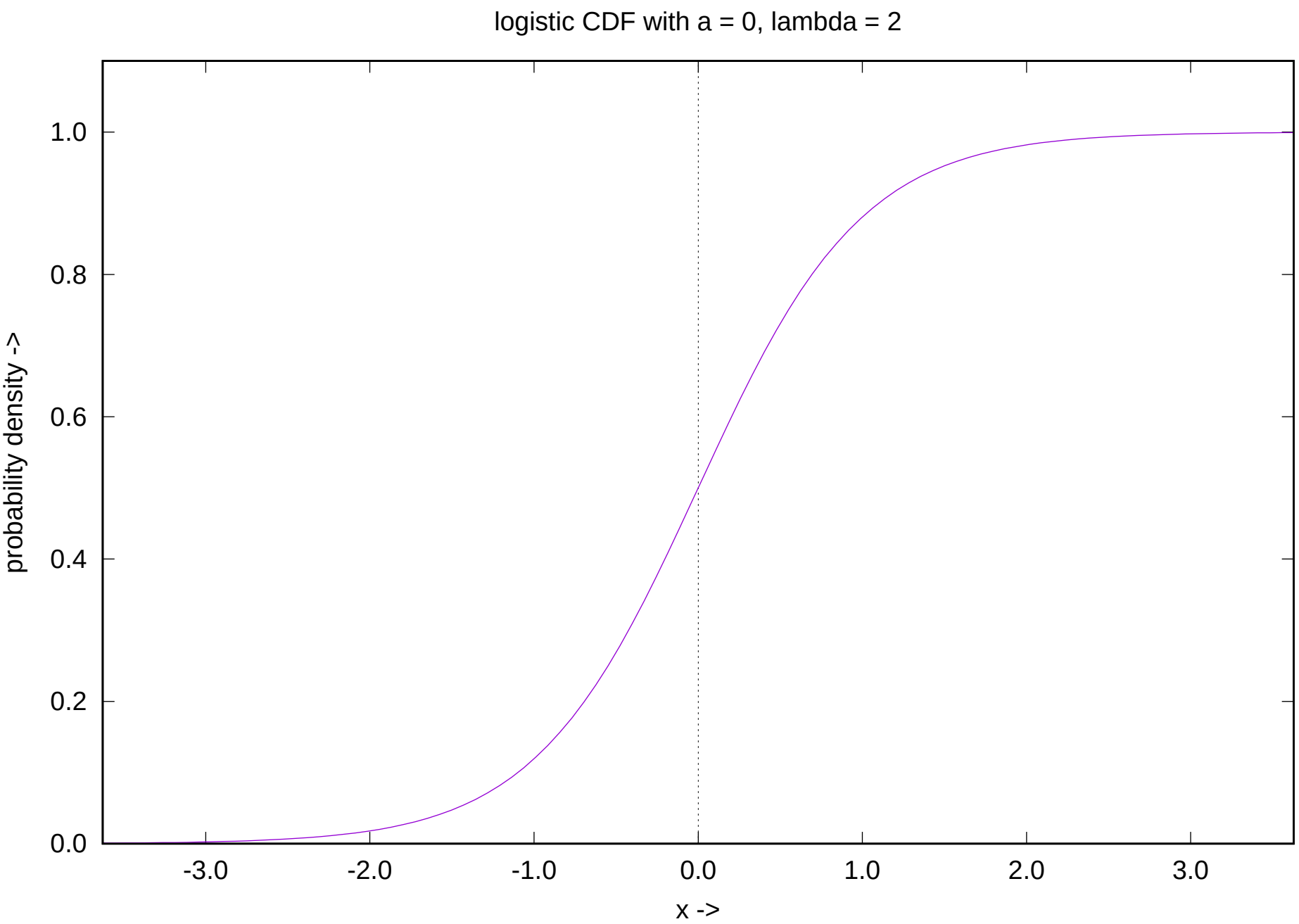

hypergeometric PDF with N = 75, C = 25, d = 10
probability density ->
0.30
0.27
0.24
0.21
0.18
0.15
0.12
0.09
0.06
0.03
0.00
0
2
4
6
8
10
k ->

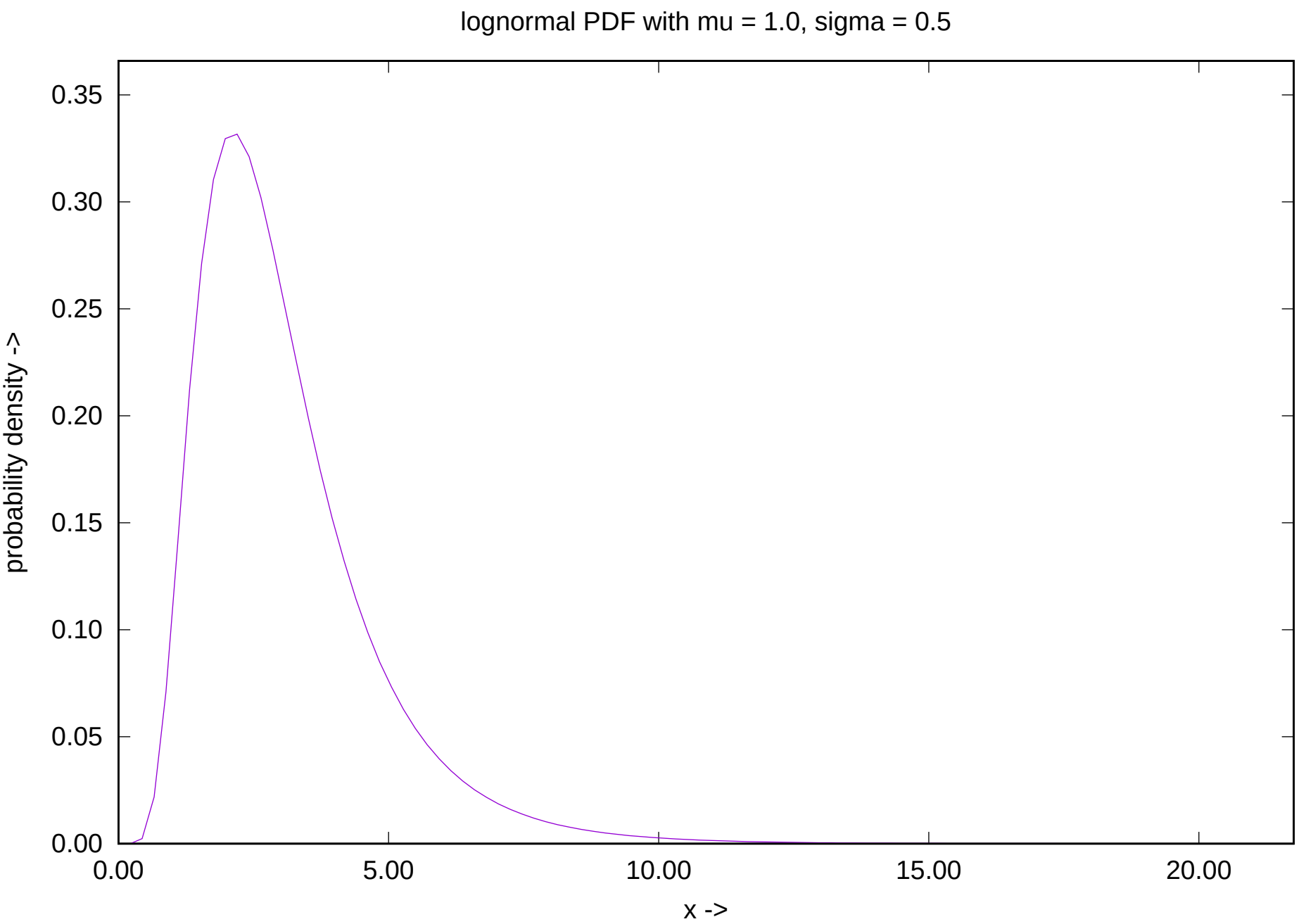

hypergeometric CDF with N = 75, C = 25, d = 10
probability density ->
k ->
1.10
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0
2
4
6
8
10

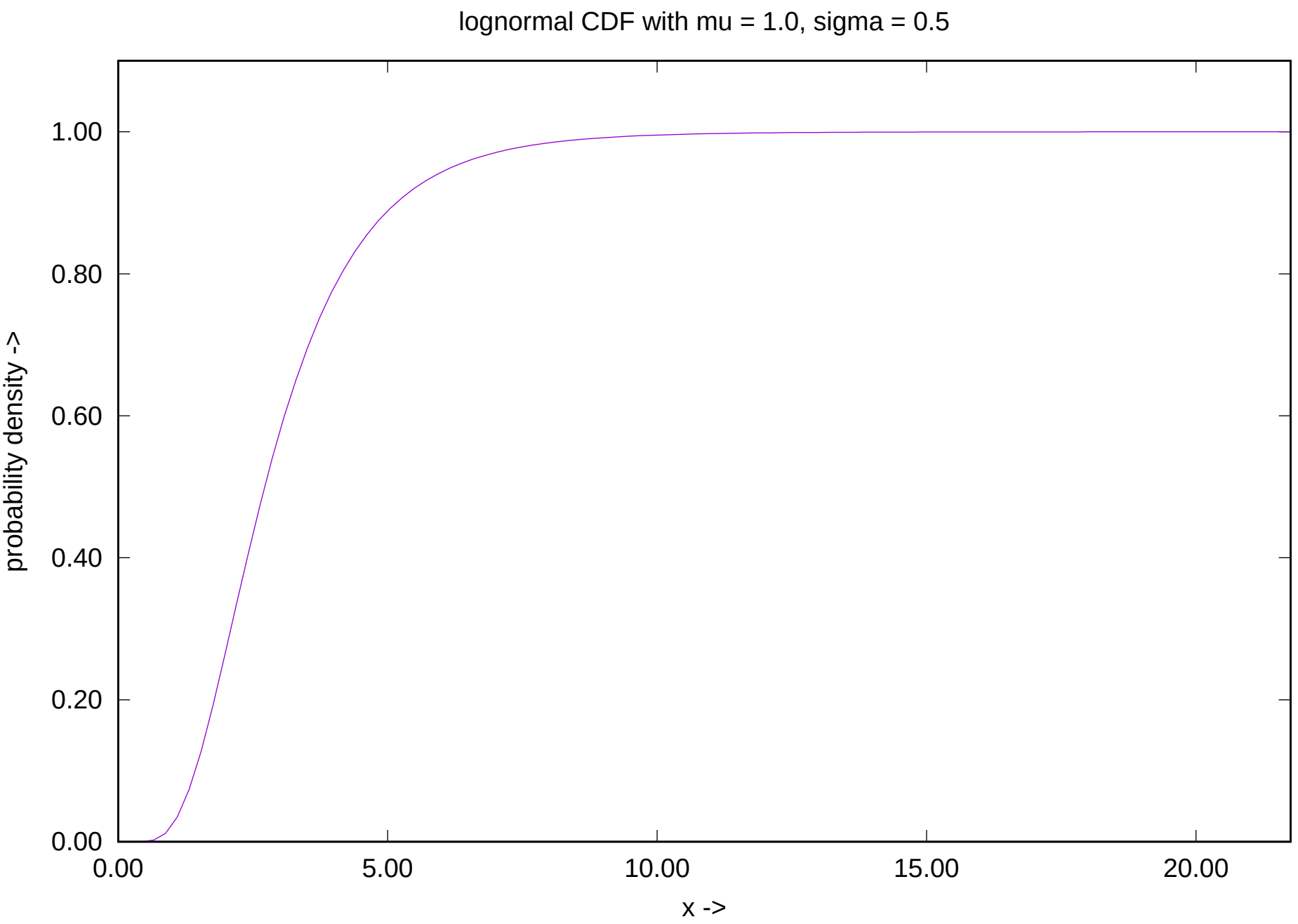

laplace (or double exponential) PDF with mu = 0, b = 1
Cusp achieved by selecting point
as part of function samples
probability density ->
0.50
0.40
0.30
0.20
0.10
0.00
-4.0
-2.0
0.0
2.0
4.0
x ->

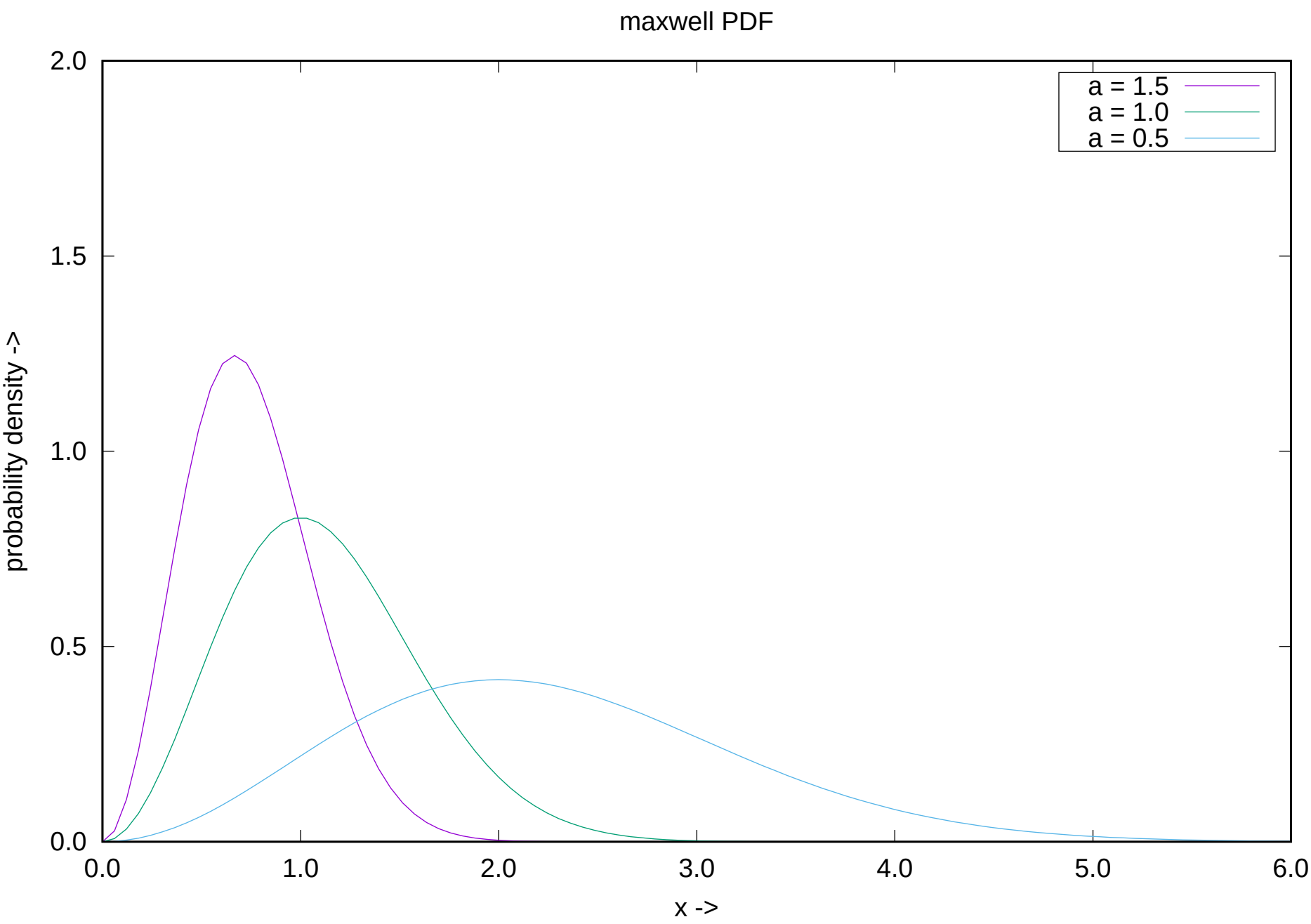

laplace (or double exponential) CDF with mu = 0, b = 1
probability density ->
1.00
0.80
0.60
0.40
0.20
0.00
-4.0
-2.0
0.0
2.0
4.0
x ->

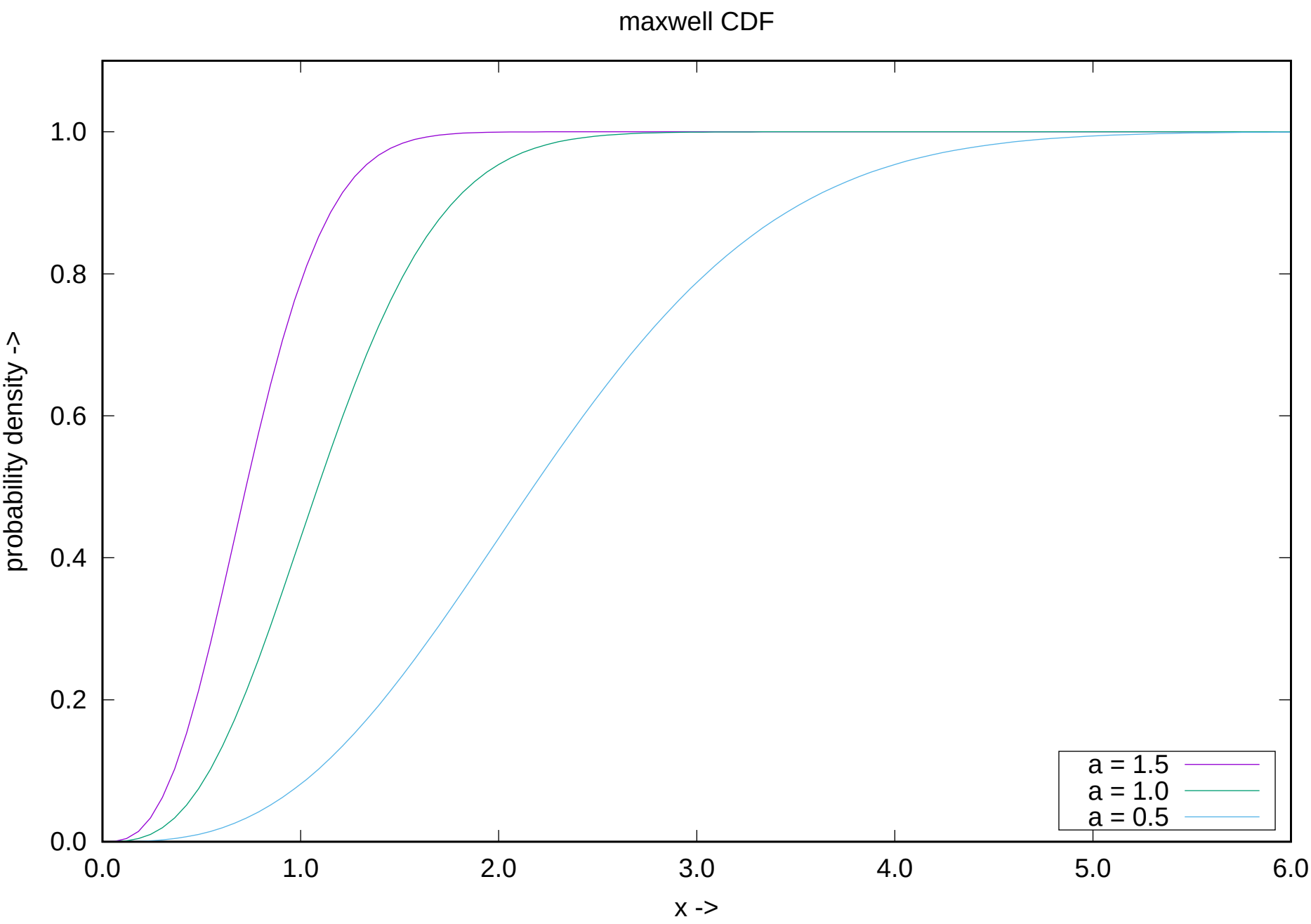

logistic PDF with a = 0, lambda = 2
probability density ->
0.5
0.4
0.3
0.2
0.1
0.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
x ->

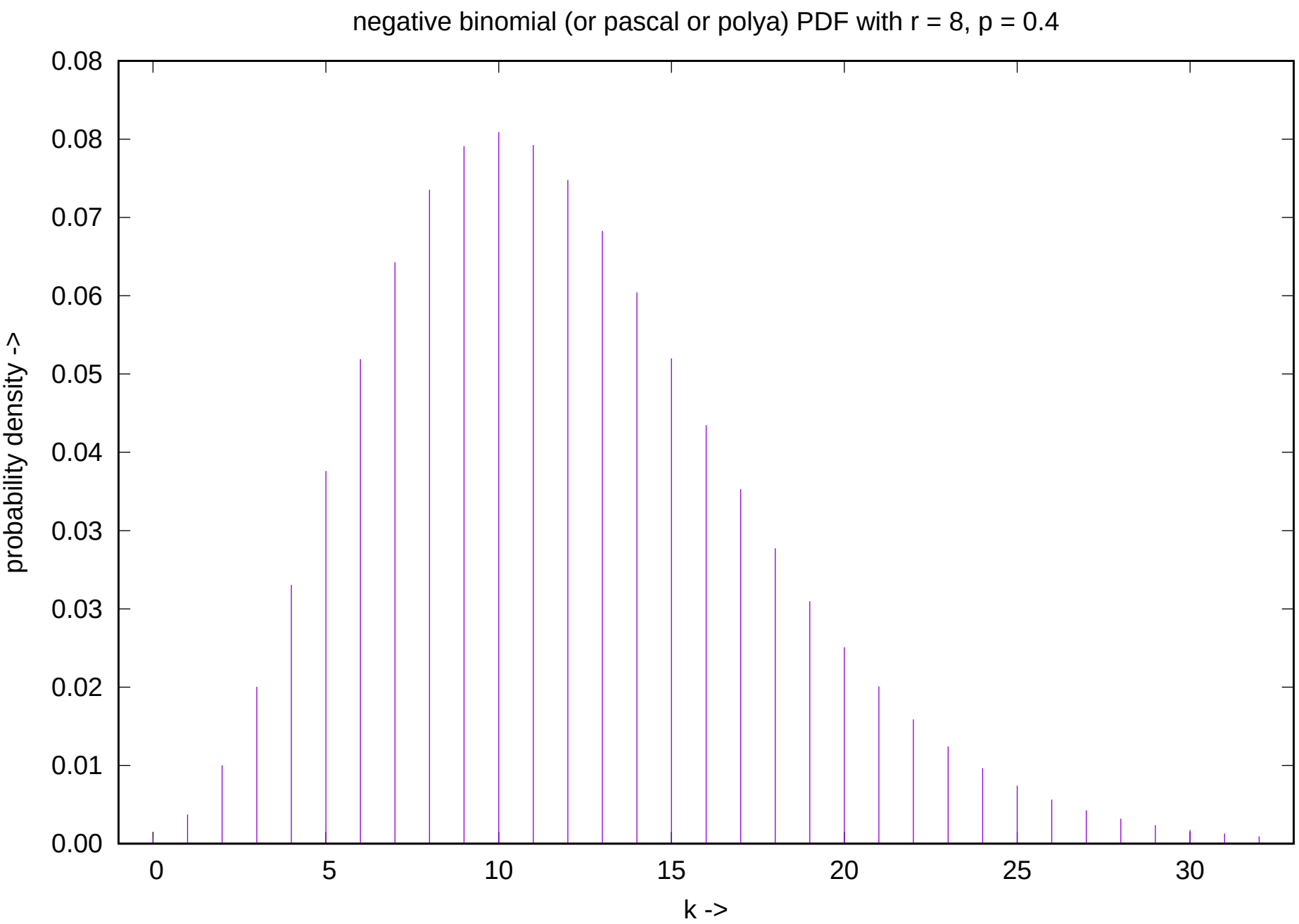

logistic CDF with a = 0, lambda = 2
probability density ->
1.0
0.8
0.6
0.4
0.2
0.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
x ->

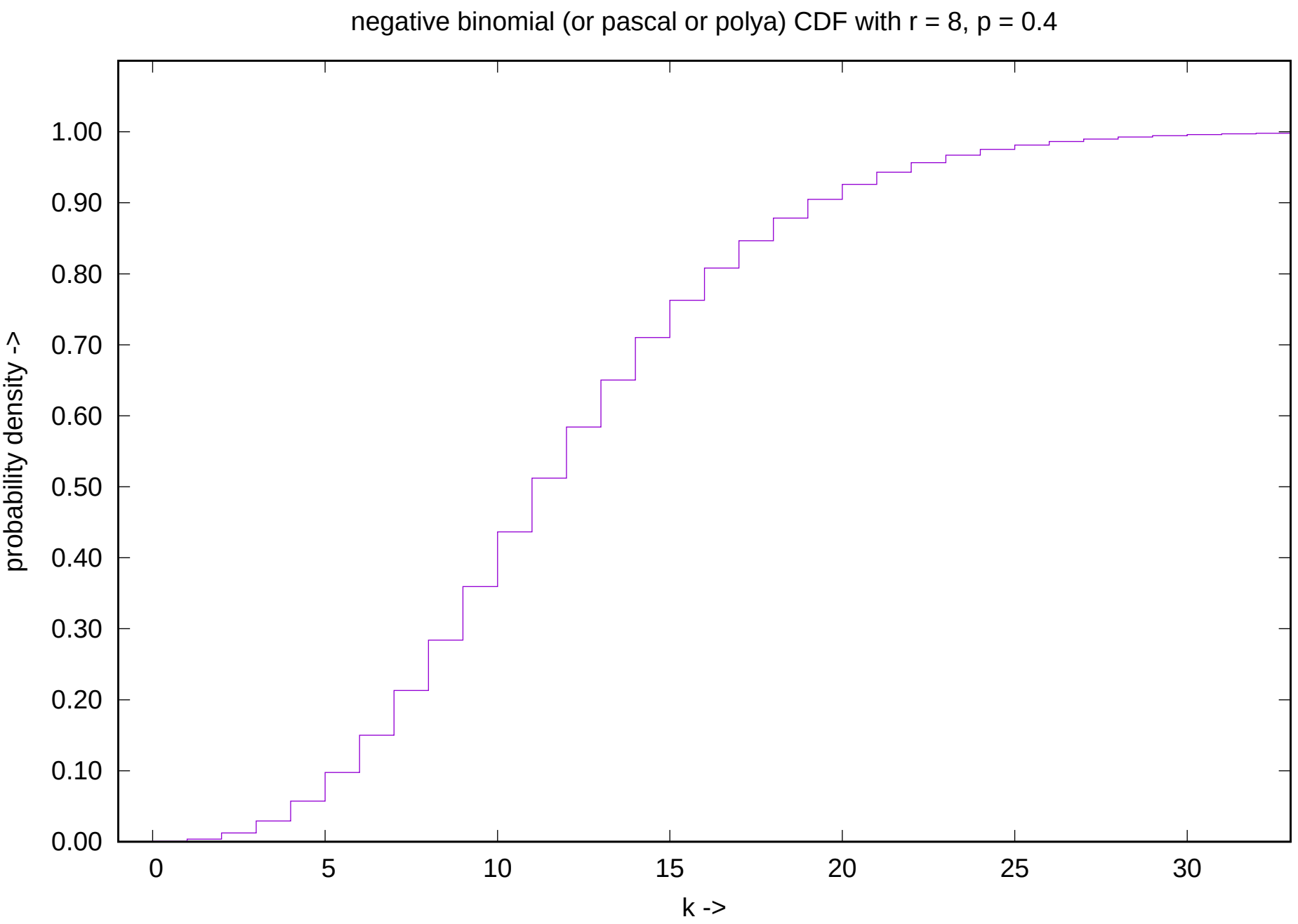

lognormal PDF with mu = 1.0, sigma = 0.5
probability density ->
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.00
5.00
10.00
15.00
20.00
x ->

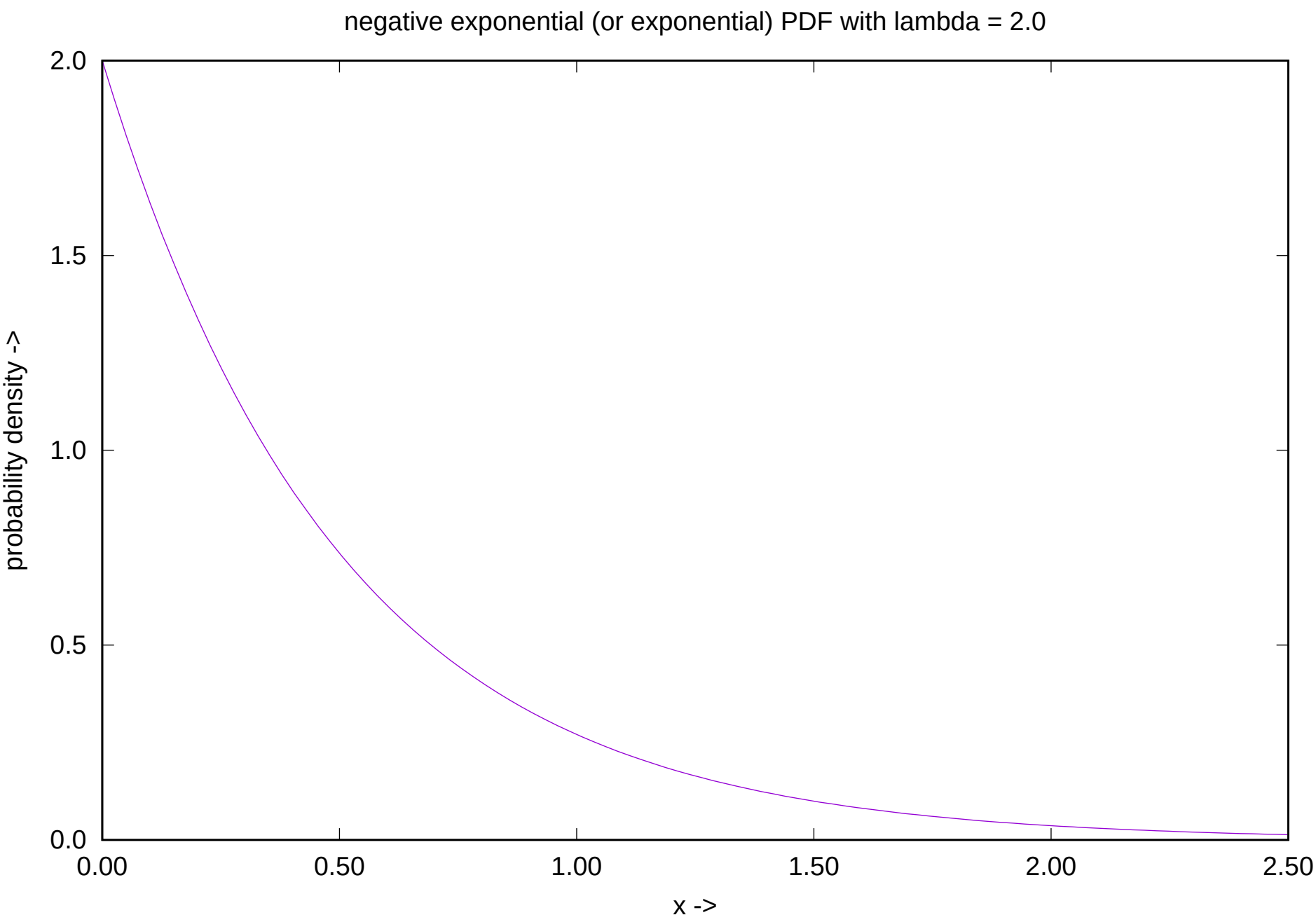

lognormal CDF with mu = 1.0, sigma = 0.5
probability density ->
1.00
0.80
0.60
0.40
0.20
0.00
0.00
5.00
10.00
15.00
20.00
x ->

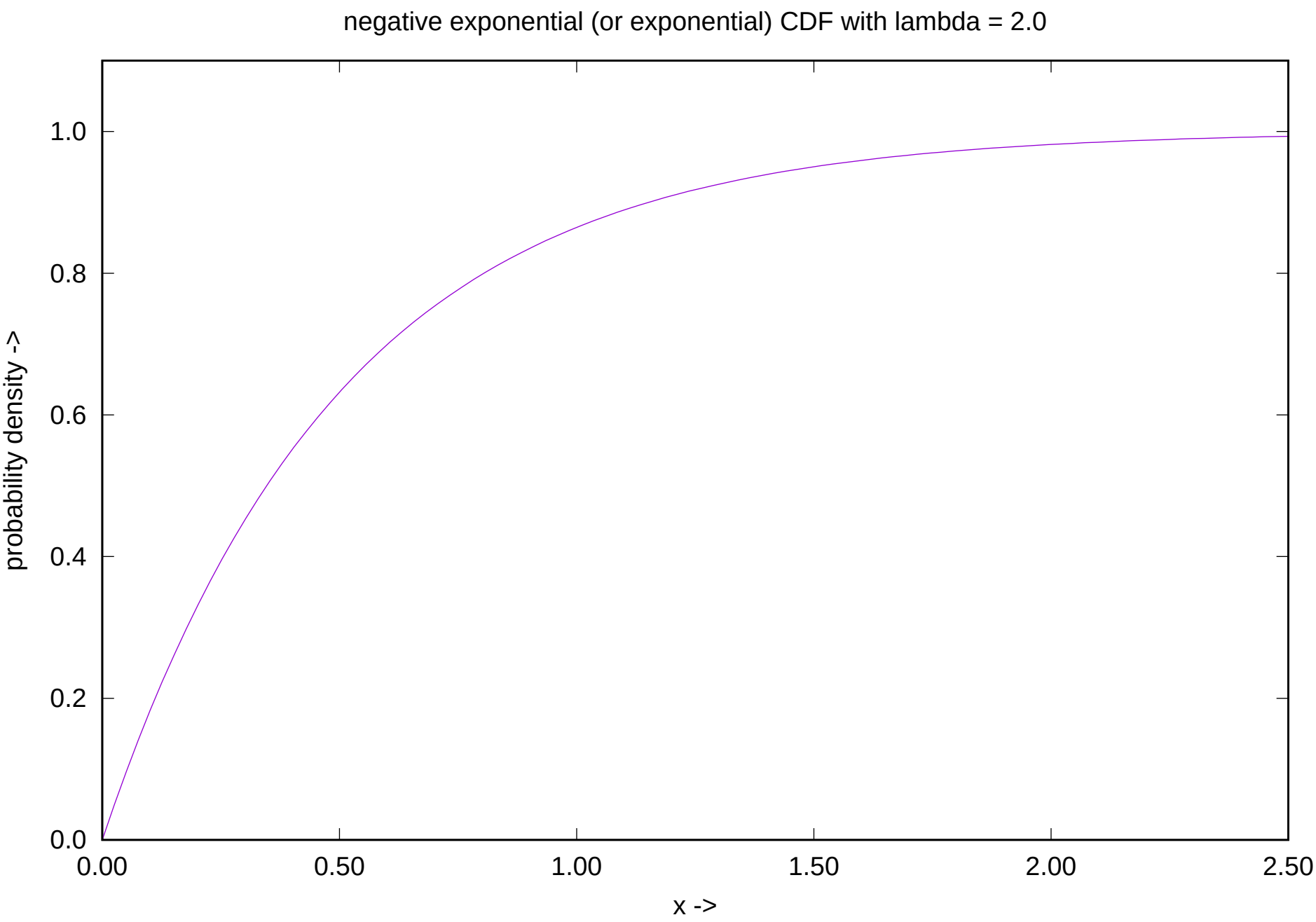

maxwell PDF
a = 1.5
a = 1.0
a = 0.5
probability density ->
x ->
0.0
0.5
1.0
1.5
2.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0

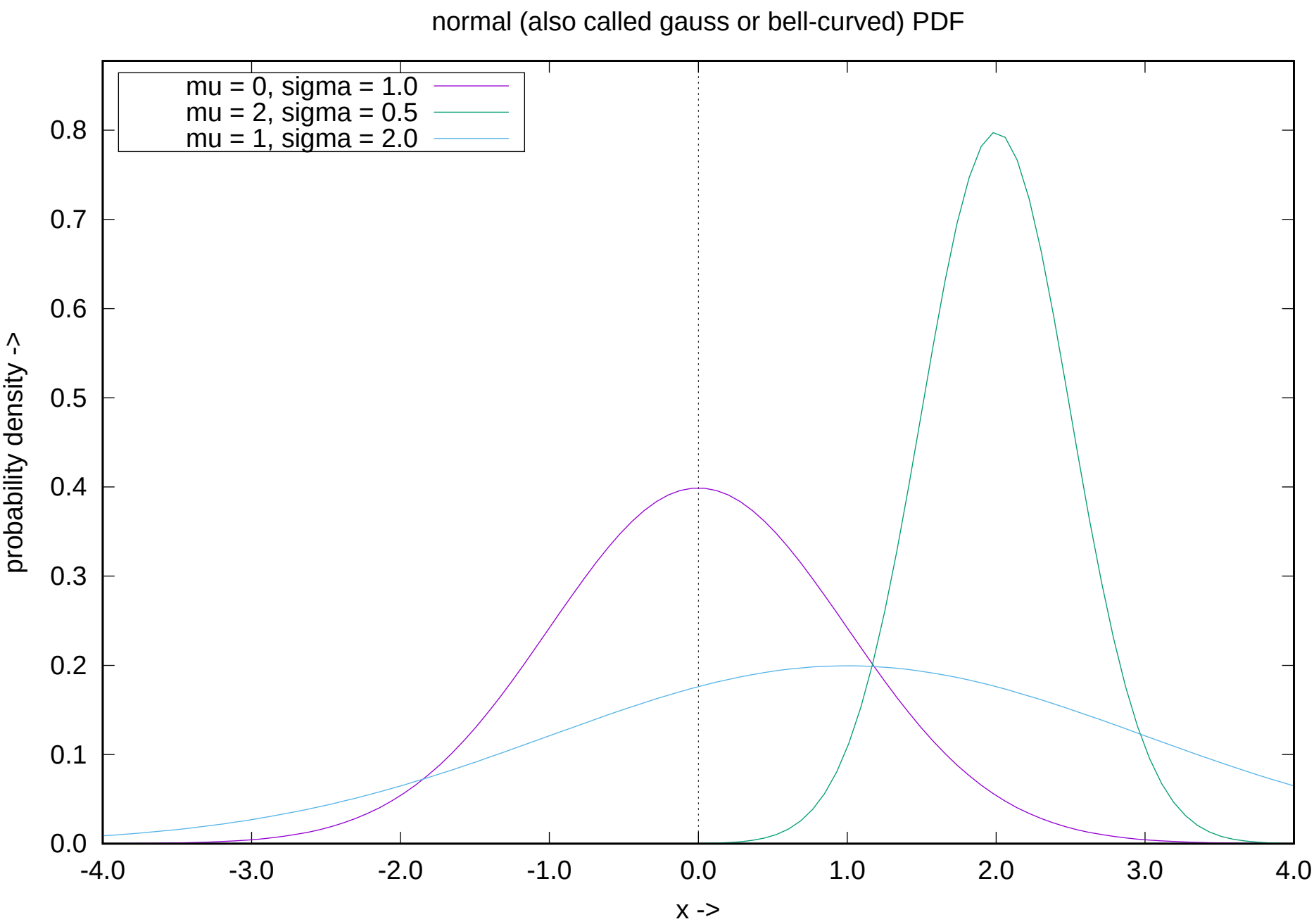

maxwell CDF
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
a = 1.5
a = 1.0
a = 0.5

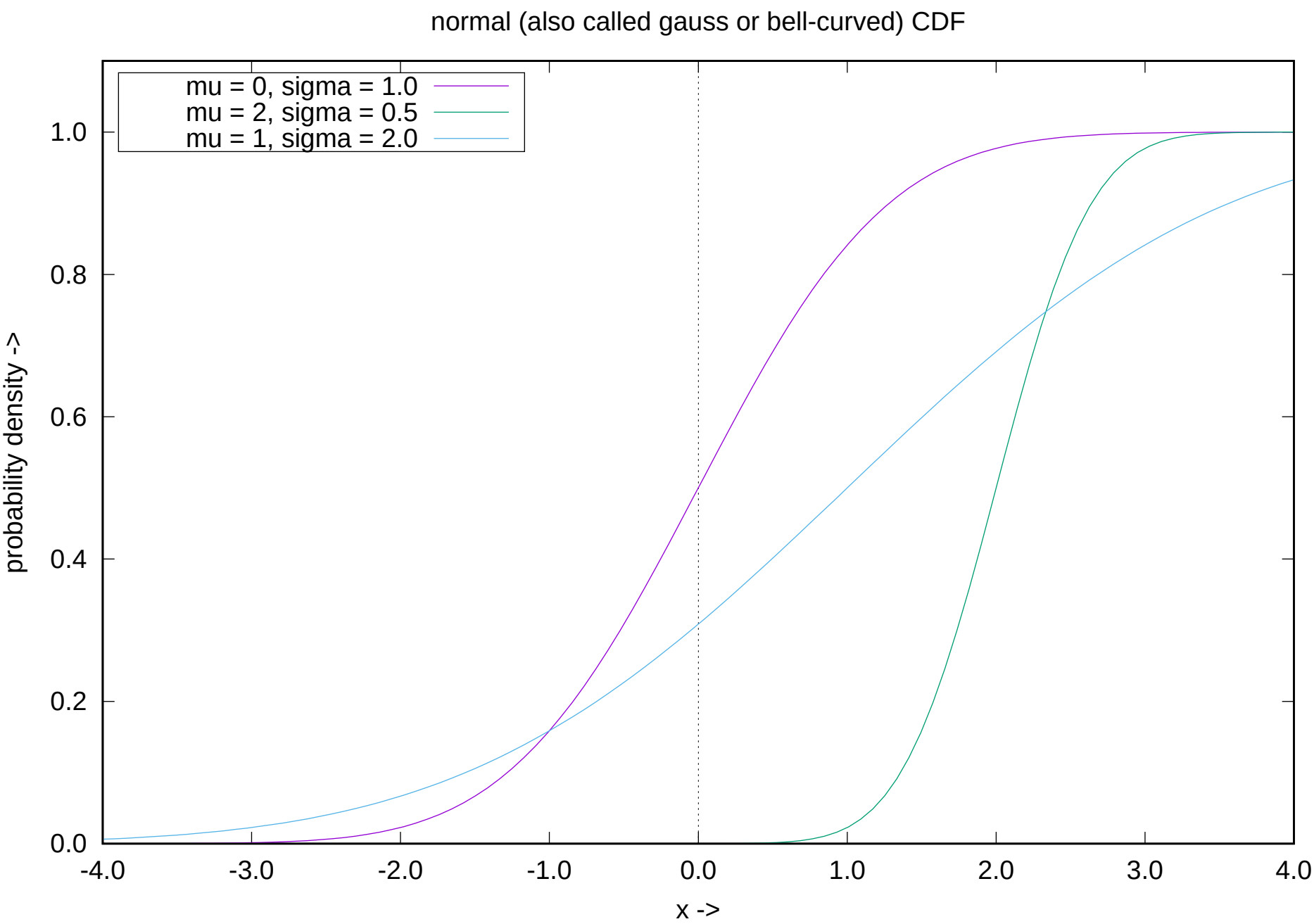

negative binomial (or pascal or polya) PDF with r = 8, p = 0.4
probability density ->
k ->
0.00
0.01
0.02
0.03
0.03
0.04
0.05
0.06
0.07
0.08
0.08
0
5
10
15
20
25
30

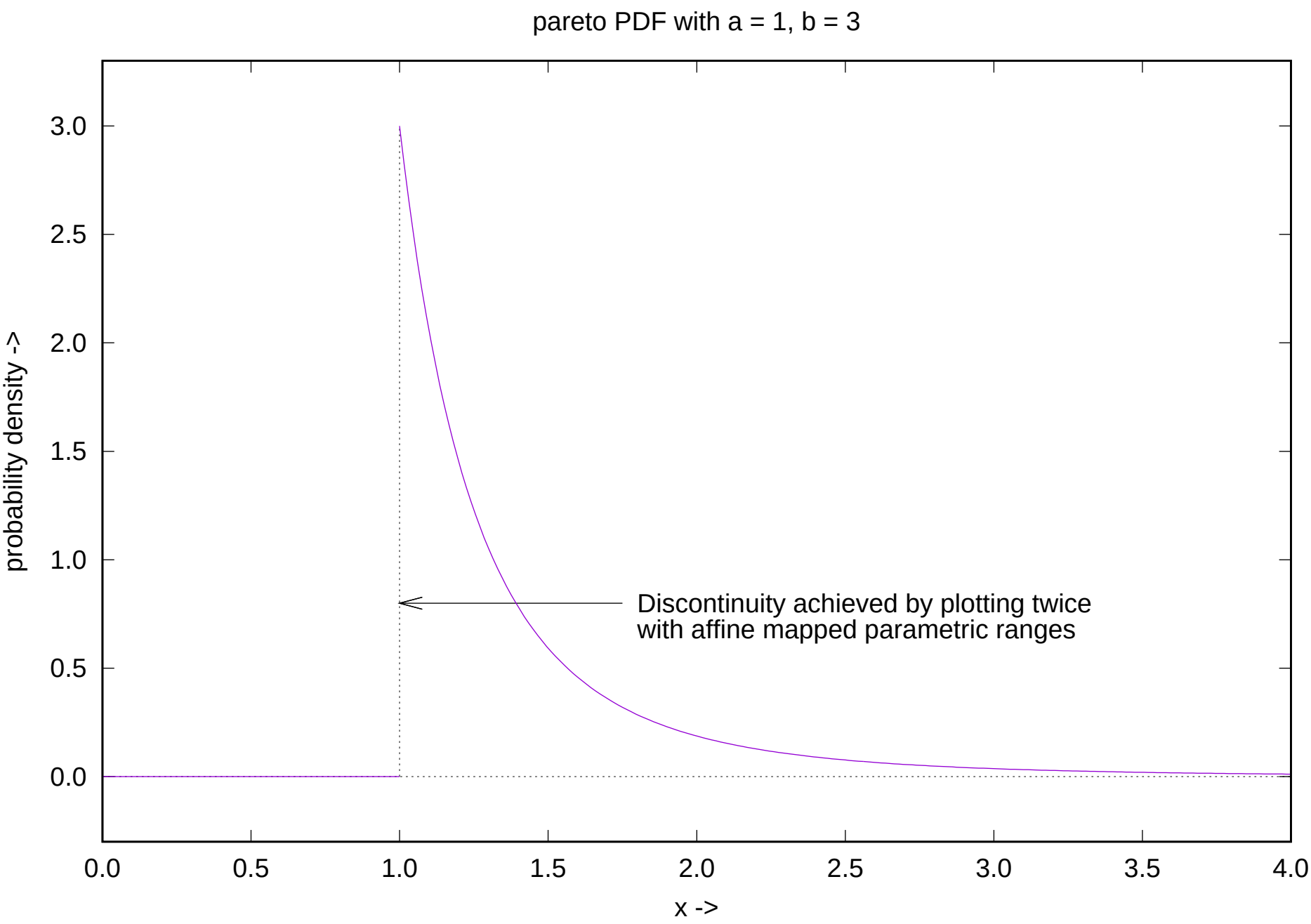

negative binomial (or pascal or polya) CDF with r = 8, p = 0.4
probability density ->
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0
5
10
15
20
25
30
k ->

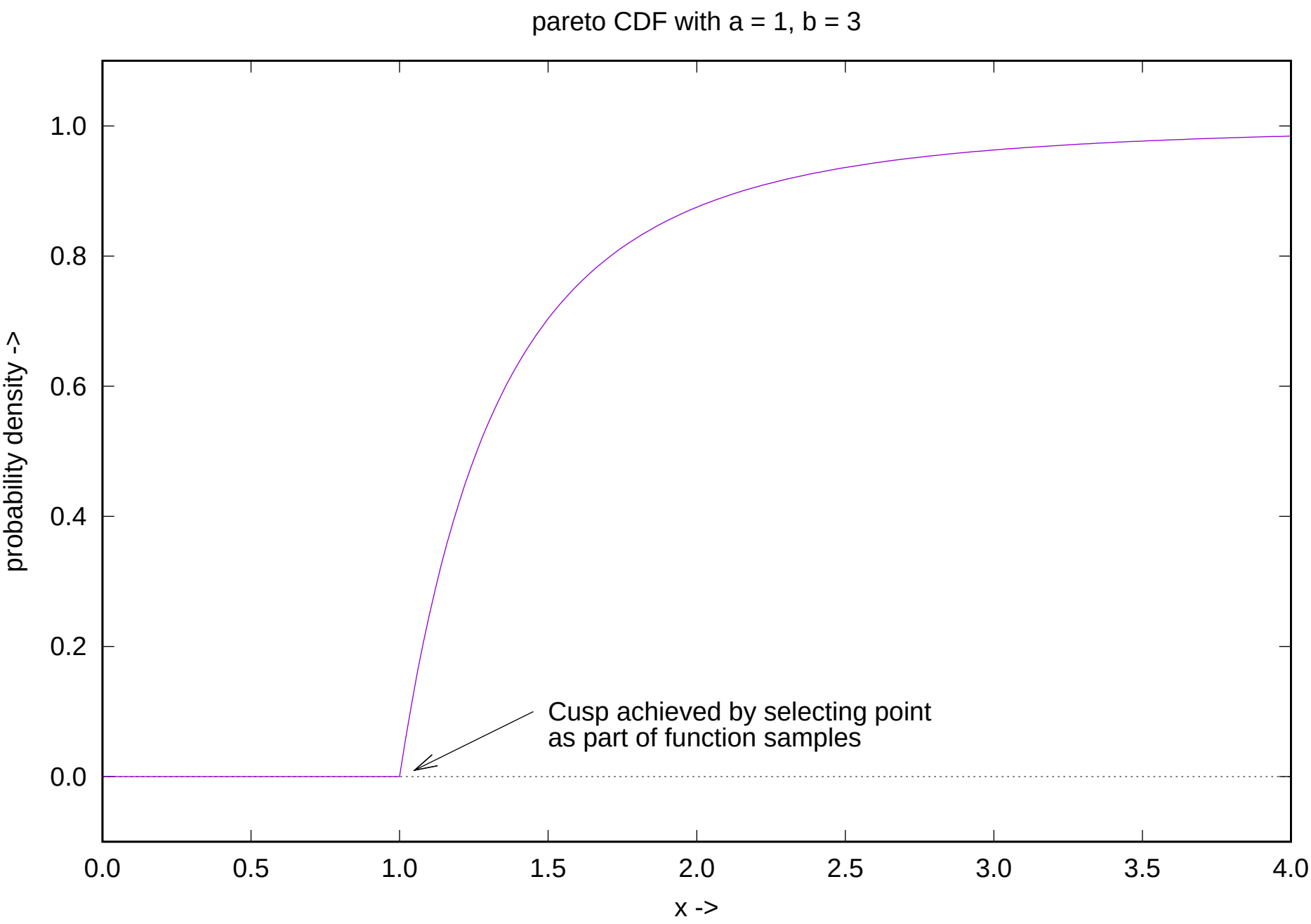

negative exponential (or exponential) PDF with lambda = 2.0
probability density ->
x ->
2.0
1.5
1.0
0.5
0.0
0.00
0.50
1.00
1.50
2.00
2.50

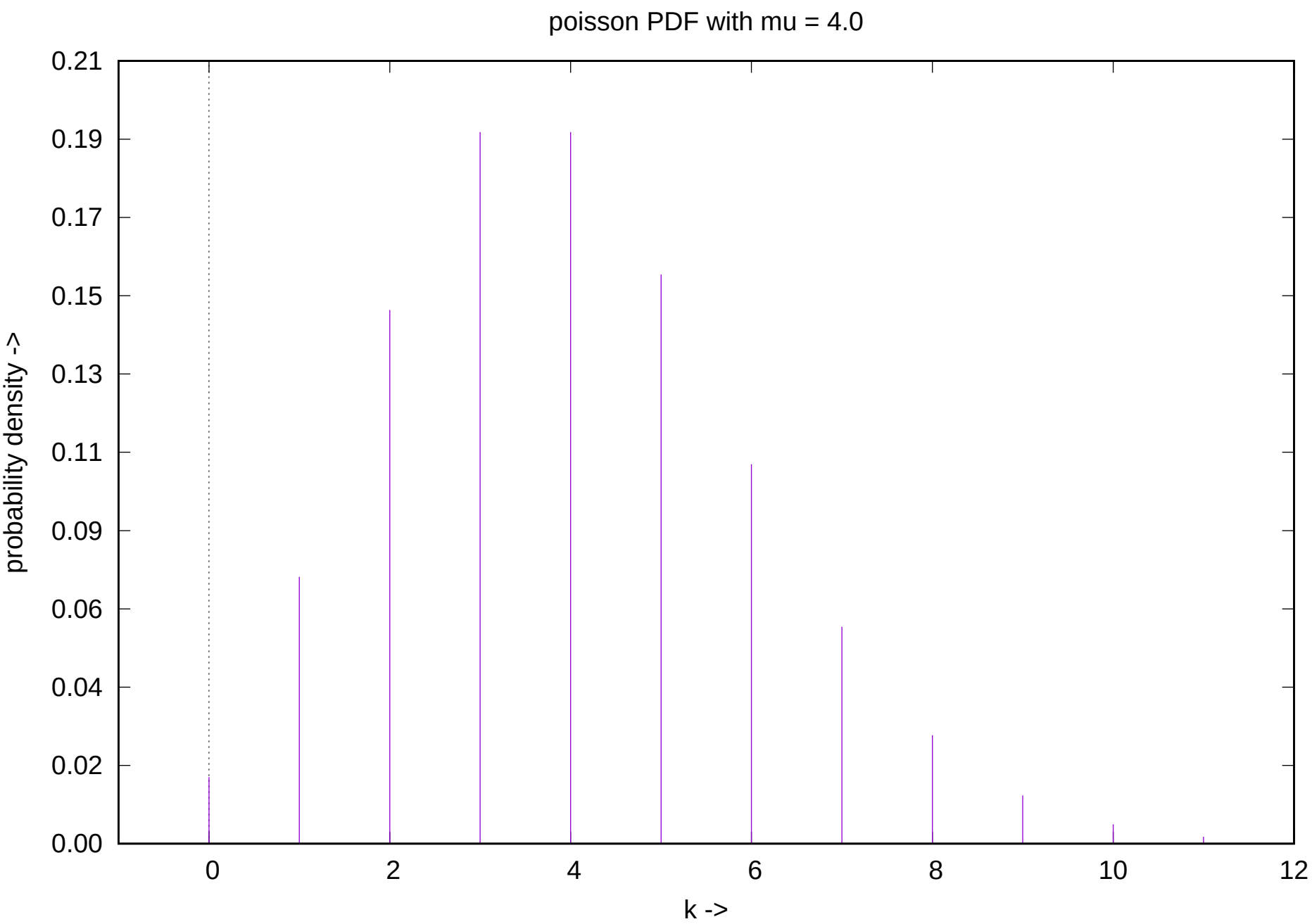

negative exponential (or exponential) CDF with lambda = 2.0
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.00
0.50
1.00
1.50
2.00
2.50

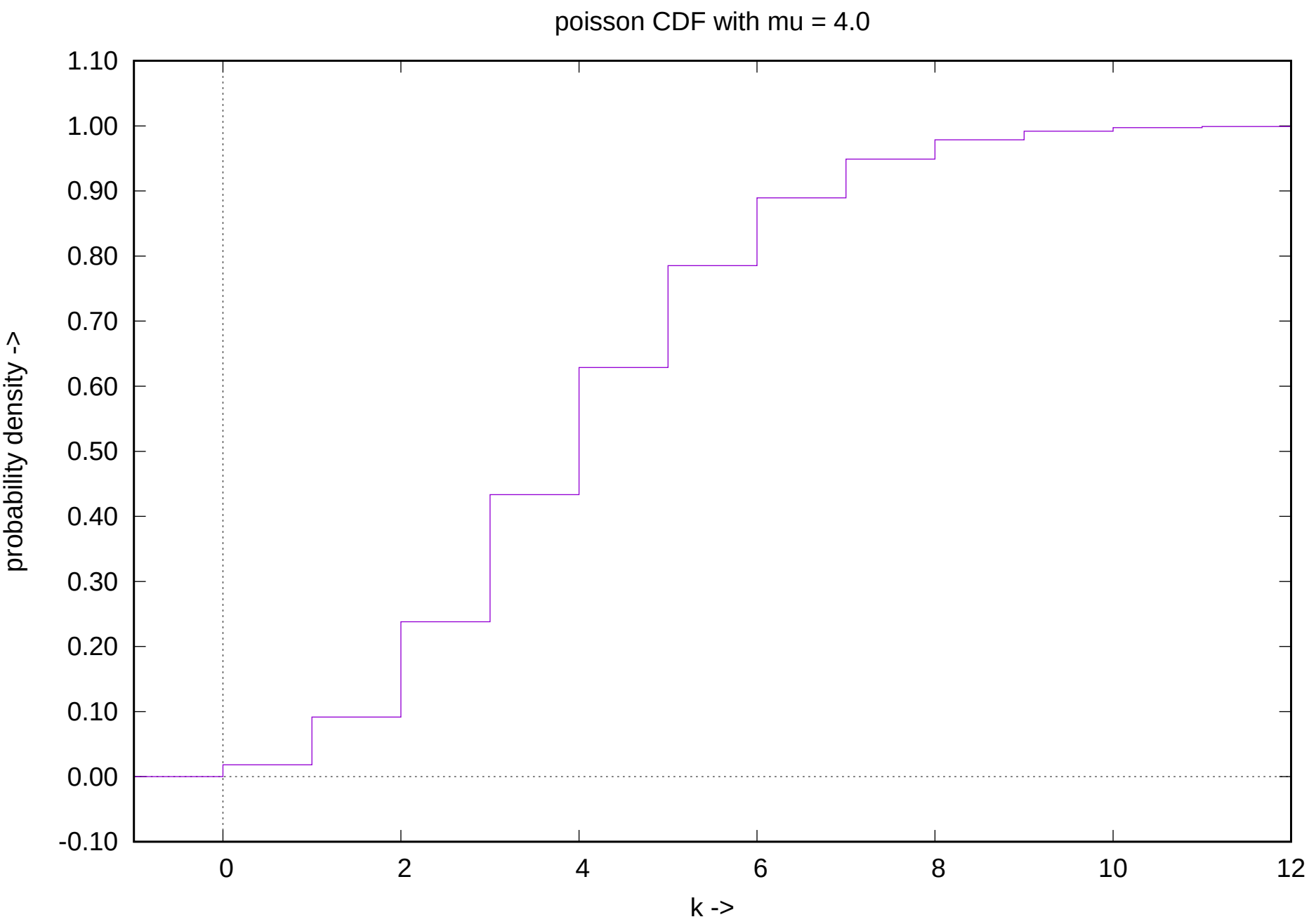

normal (also called gauss or bell-curved) PDF
mu = 0, sigma = 1.0
mu = 2, sigma = 0.5
mu = 1, sigma = 2.0
probability density ->
x ->
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
-4.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0

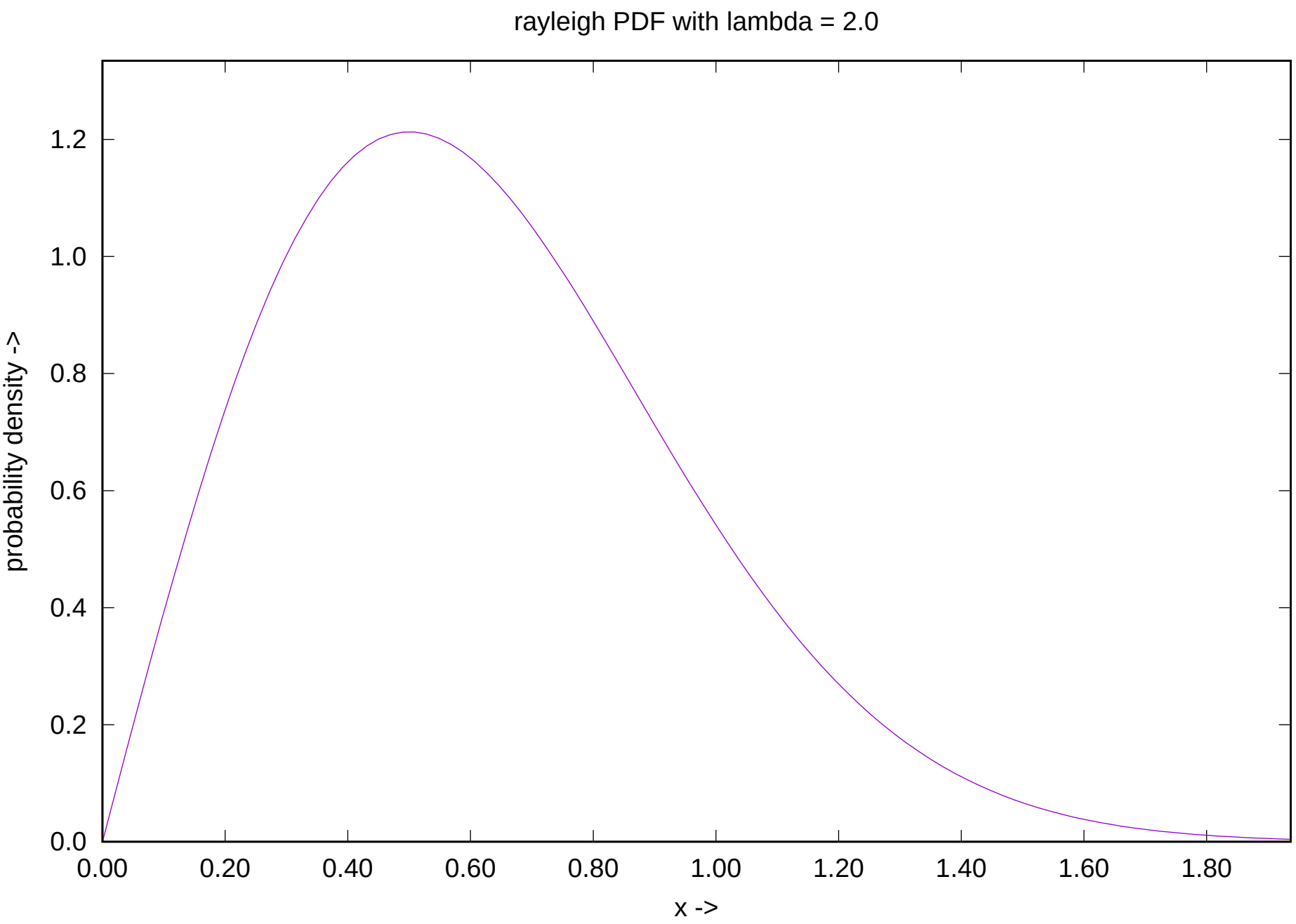

normal (also called gauss or bell-curved) CDF
mu = 0, sigma = 1.0
mu = 2, sigma = 0.5
mu = 1, sigma = 2.0
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
-4.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0

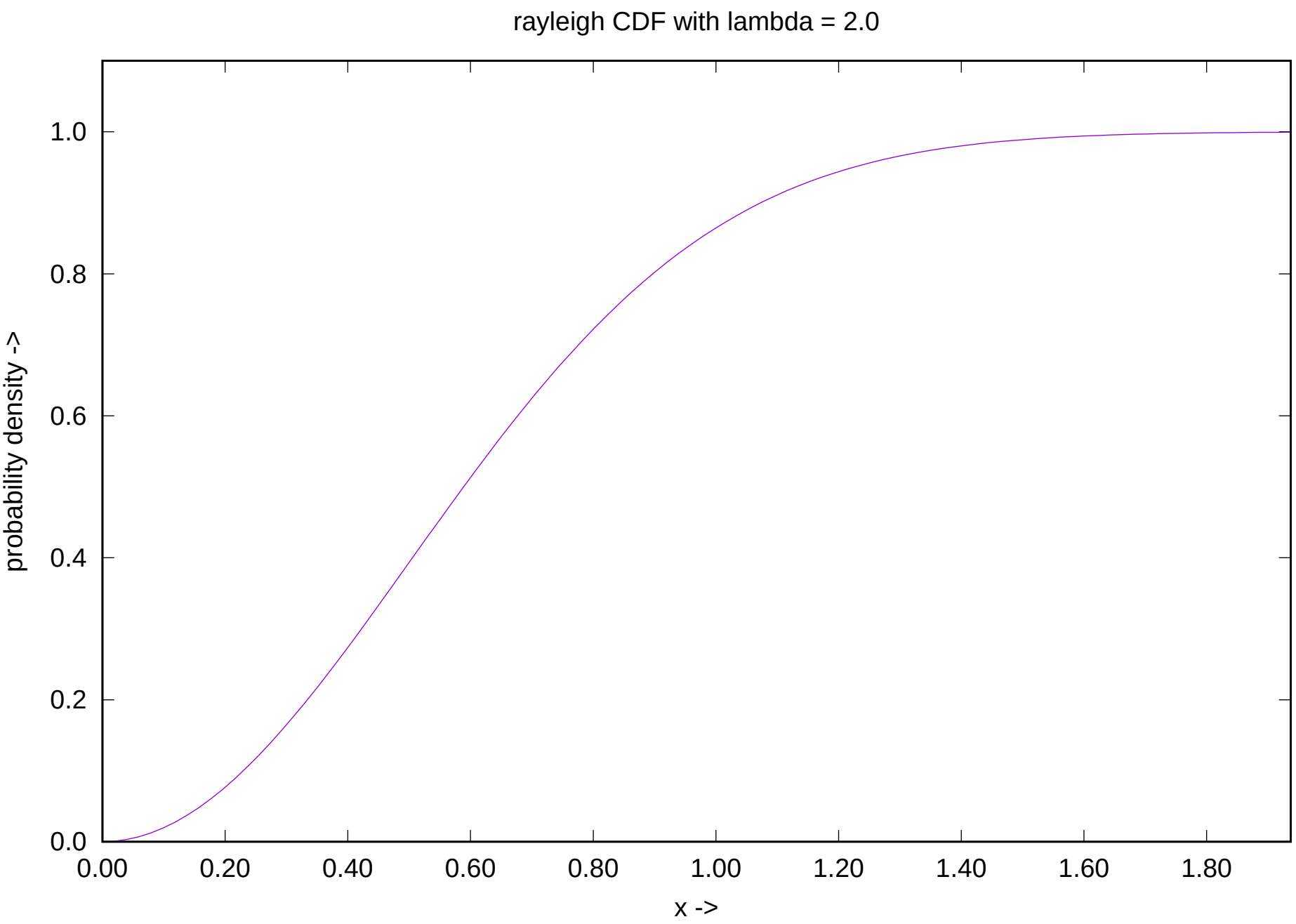

pareto PDF with a = 1, b = 3
probability density ->
x ->
Discontinuity achieved by plotting twice
with affine mapped parametric ranges
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

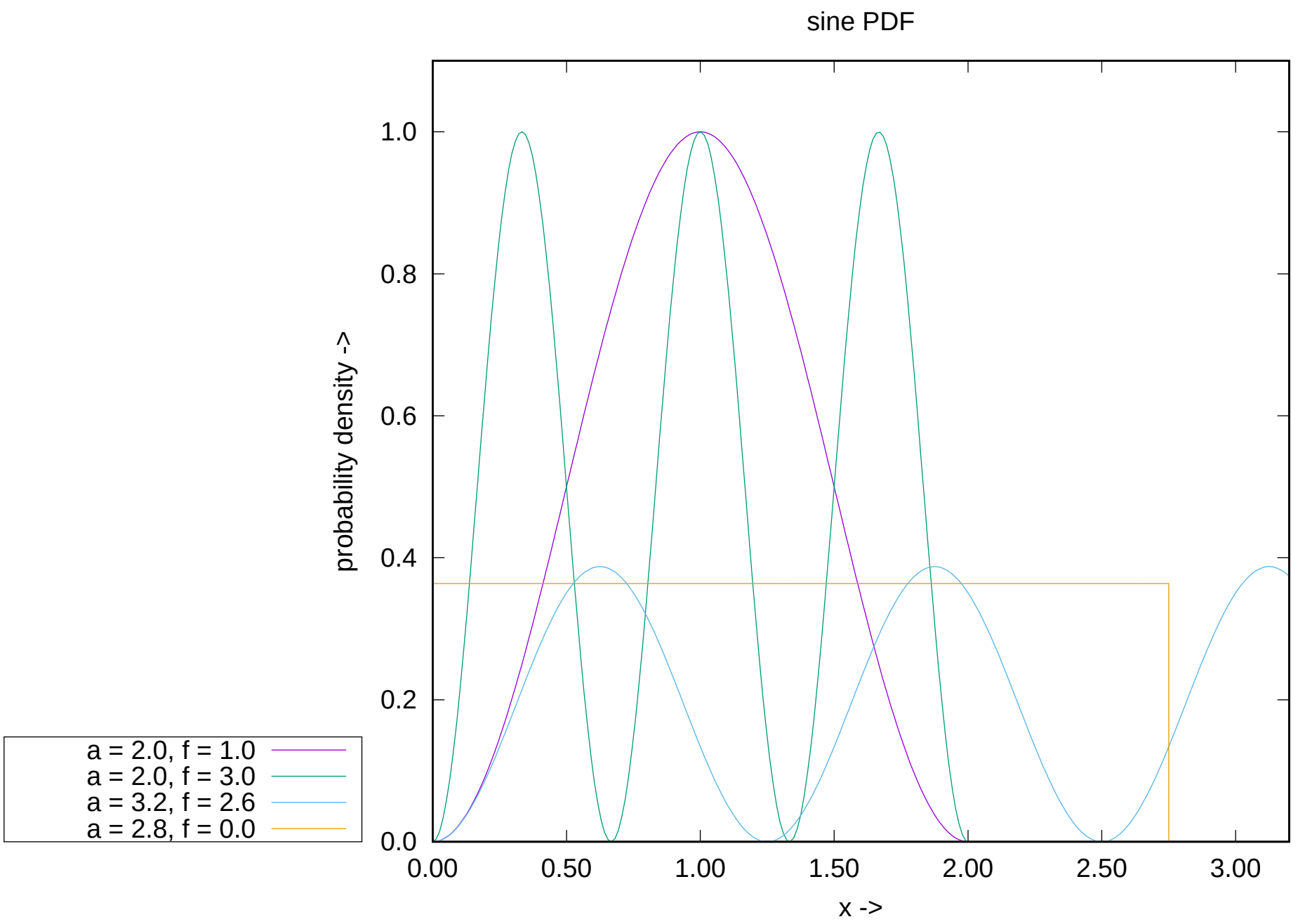

pareto CDF with a = 1, b = 3
probability density ->
x ->
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Cusp achieved by selecting point
as part of function samples

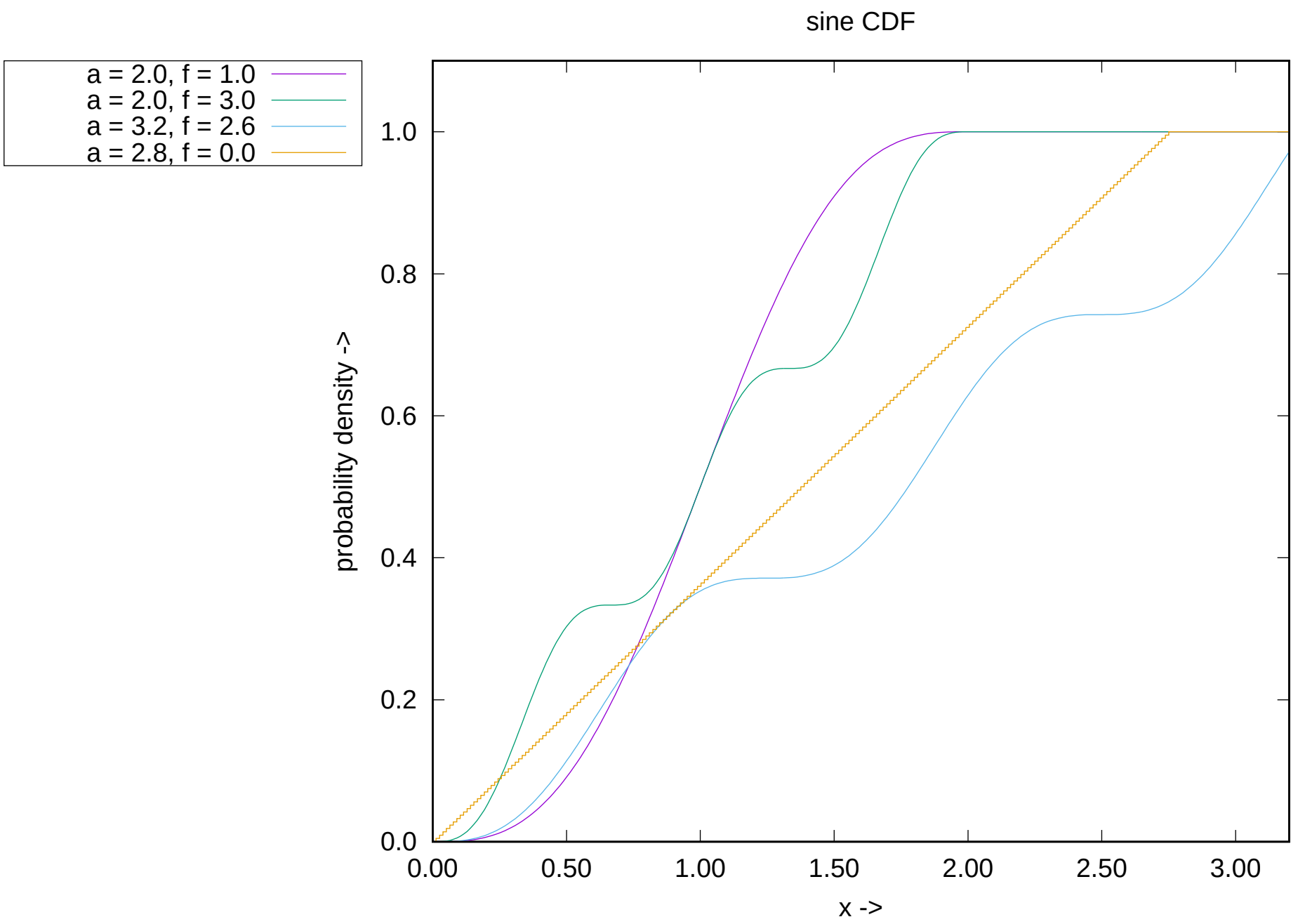

poisson PDF with mu = 4.0
probability density ->
k ->
0.21
0.19
0.17
0.15
0.13
0.11
0.09
0.06
0.04
0.02
0.00
0
2
4
6
8
10
12

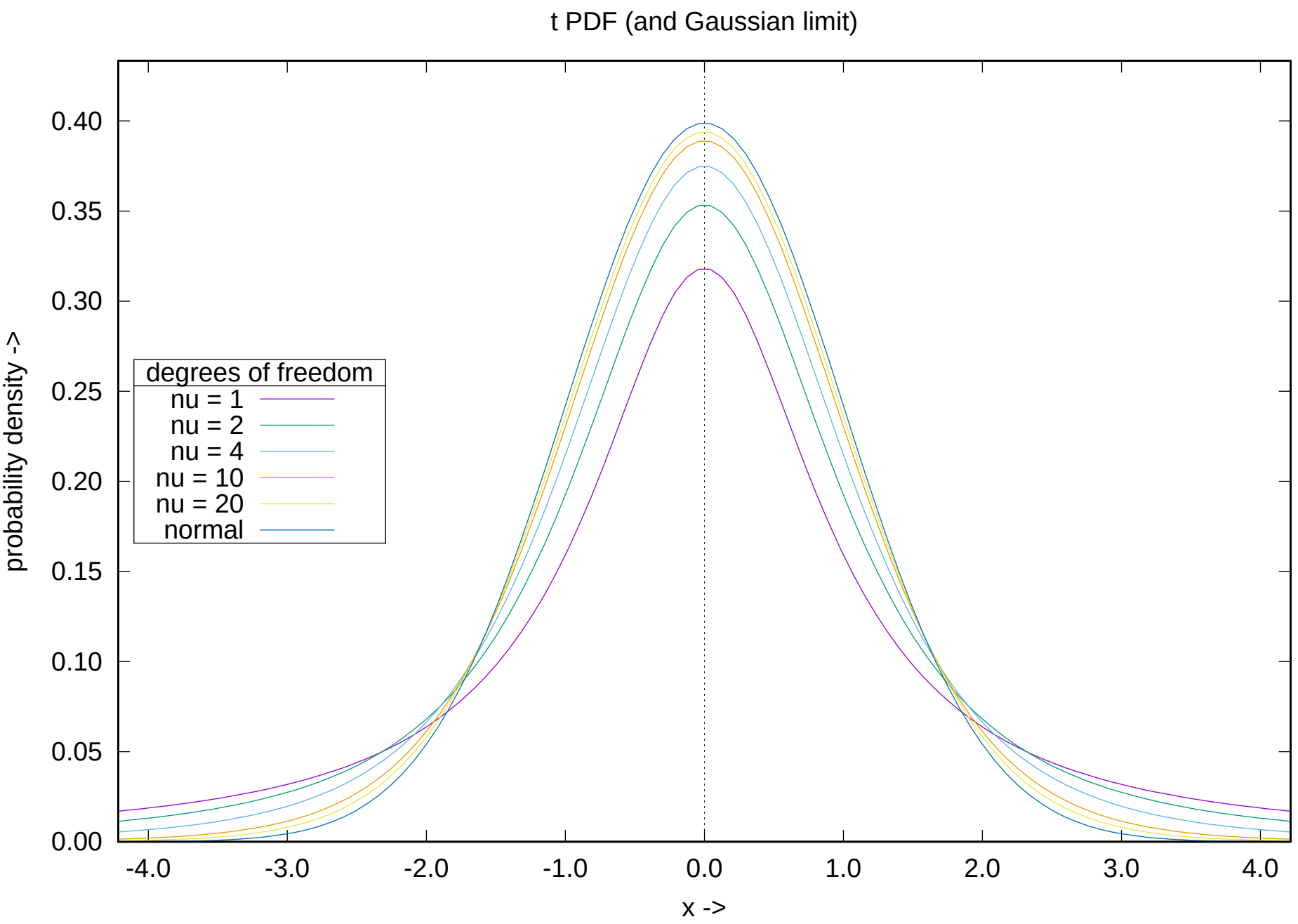

poisson CDF with mu = 4.0
probability density ->
k ->
1.10
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
-0.10
0
2
4
6
8
10
12

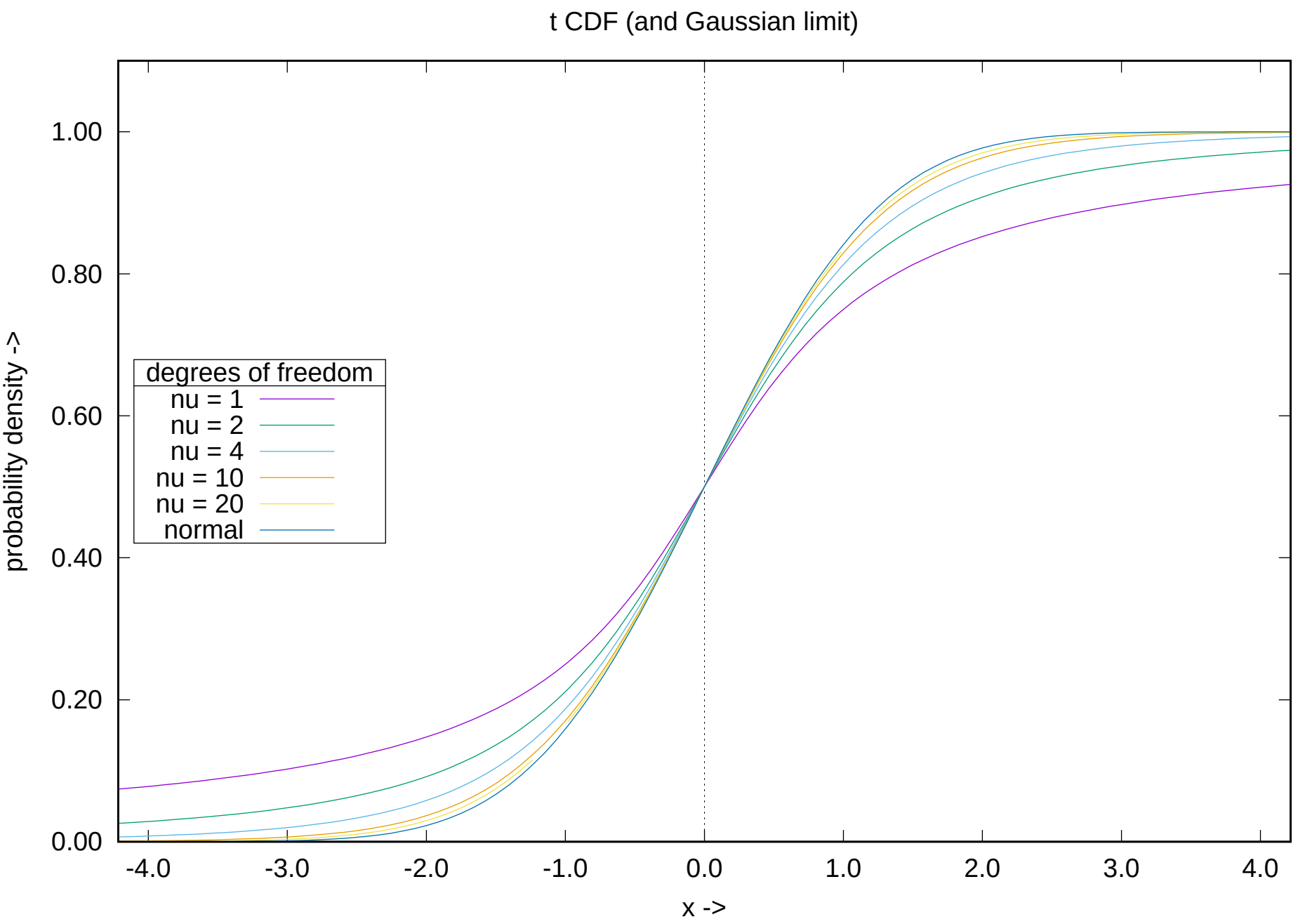

rayleigh PDF with lambda = 2.0
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0.00
0.20
0.40
0.60
0.80
1.00
1.20
1.40
1.60
1.80

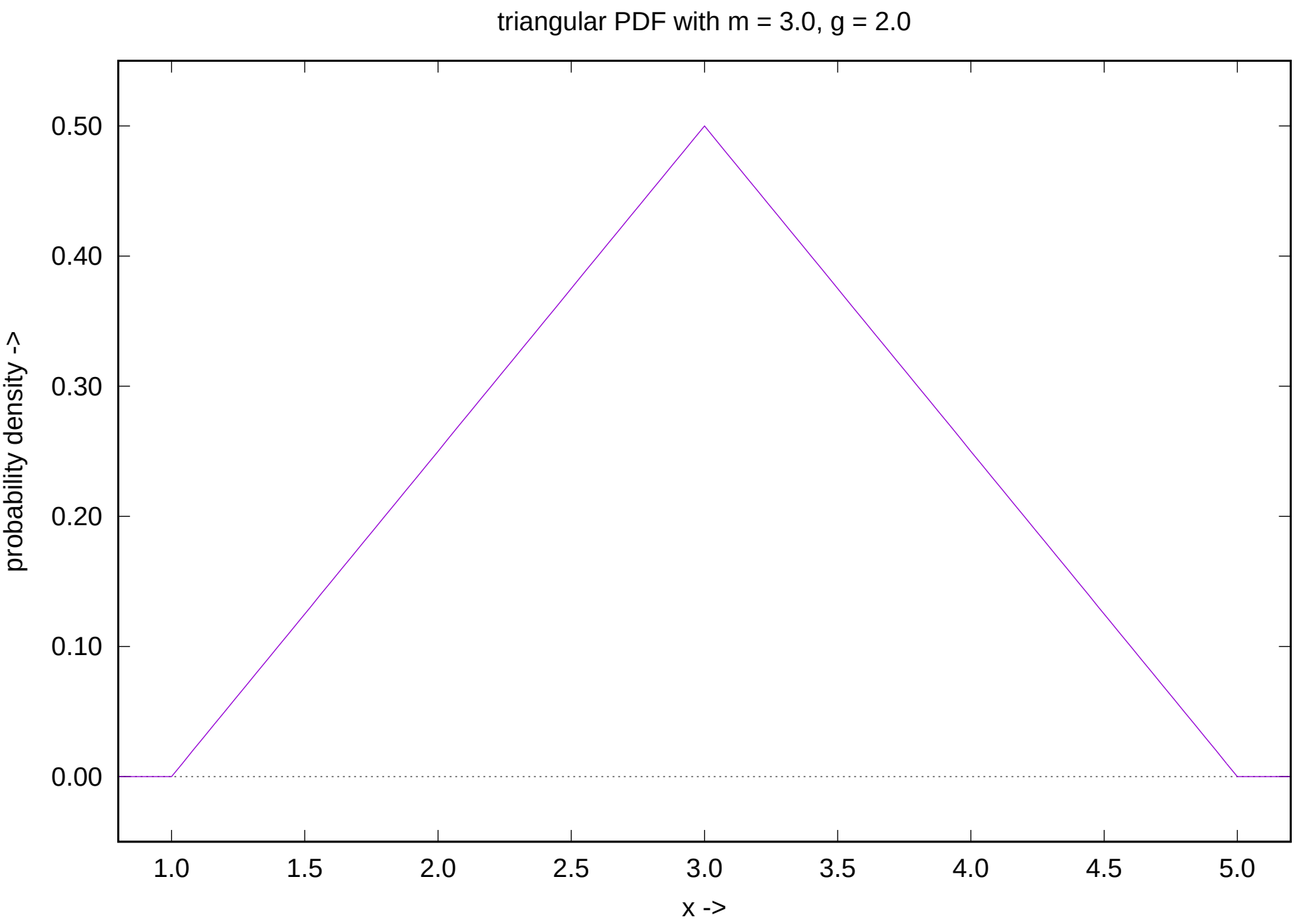

rayleigh CDF with lambda = 2.0
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.00
0.20
0.40
0.60
0.80
1.00
1.20
1.40
1.60
1.80

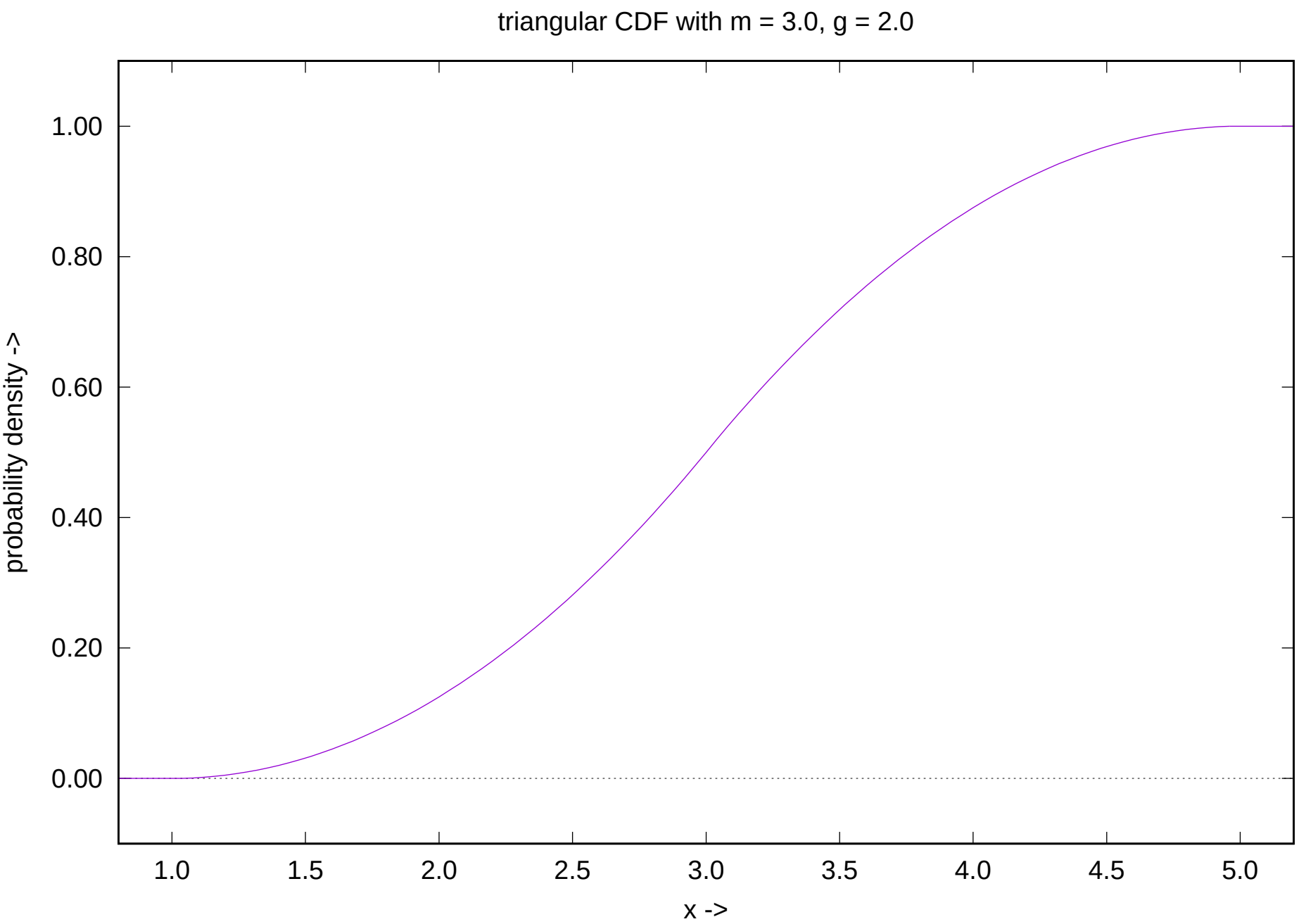

sine PDF
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.00
0.50
1.00
1.50
2.00
2.50
3.00
a = 2.0, f = 1.0
a = 2.0, f = 3.0
a = 3.2, f = 2.6
a = 2.8, f = 0.0

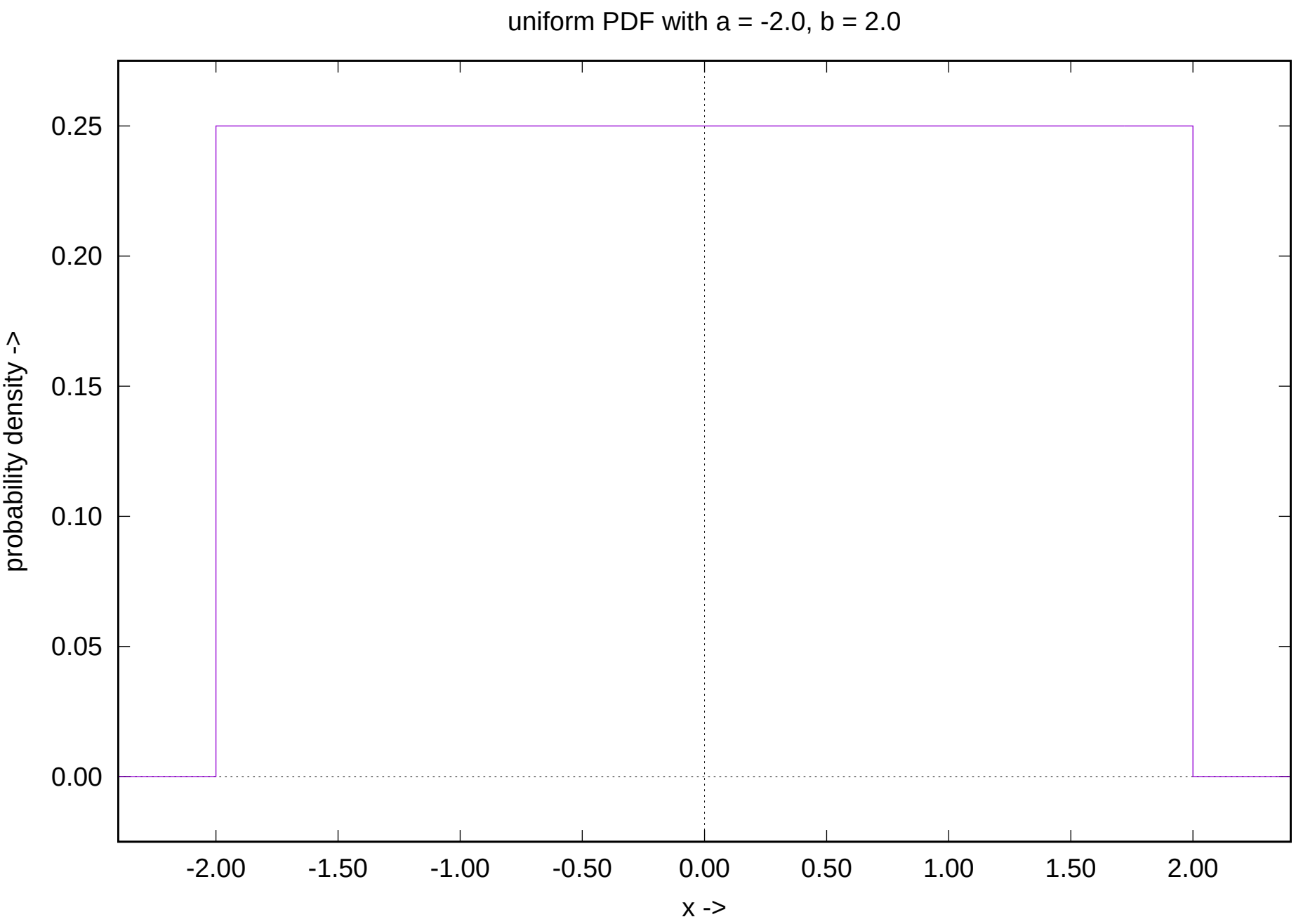

sine CDF
a = 2.0, f = 1.0
a = 2.0, f = 3.0
a = 3.2, f = 2.6
a = 2.8, f = 0.0
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.00
0.50
1.00
1.50
2.00
2.50
3.00

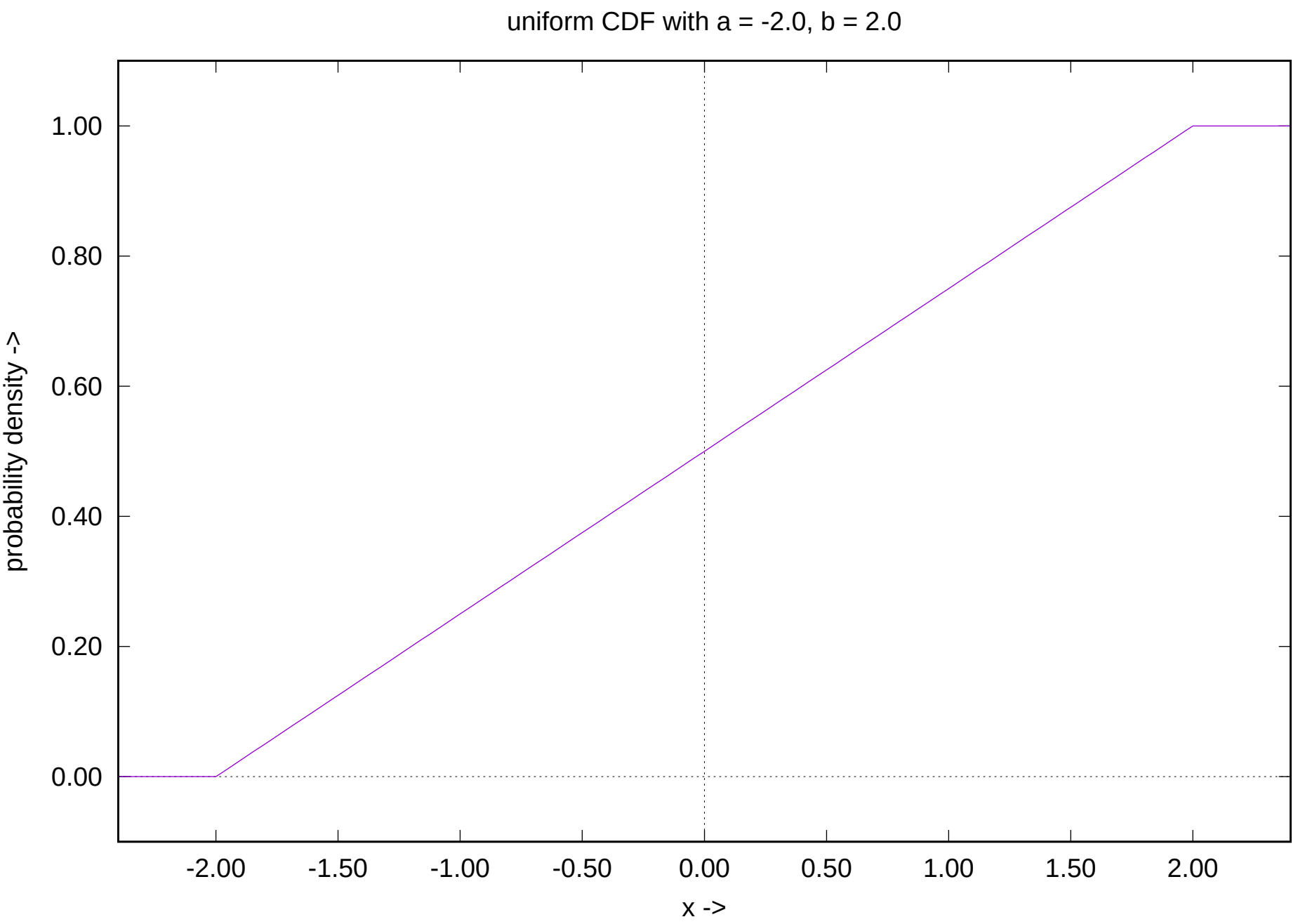

t PDF (and Gaussian limit)
probability density ->
x ->
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-4.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0
degrees of freedom
nu = 1
nu = 2
nu = 4
nu = 10
nu = 20
normal

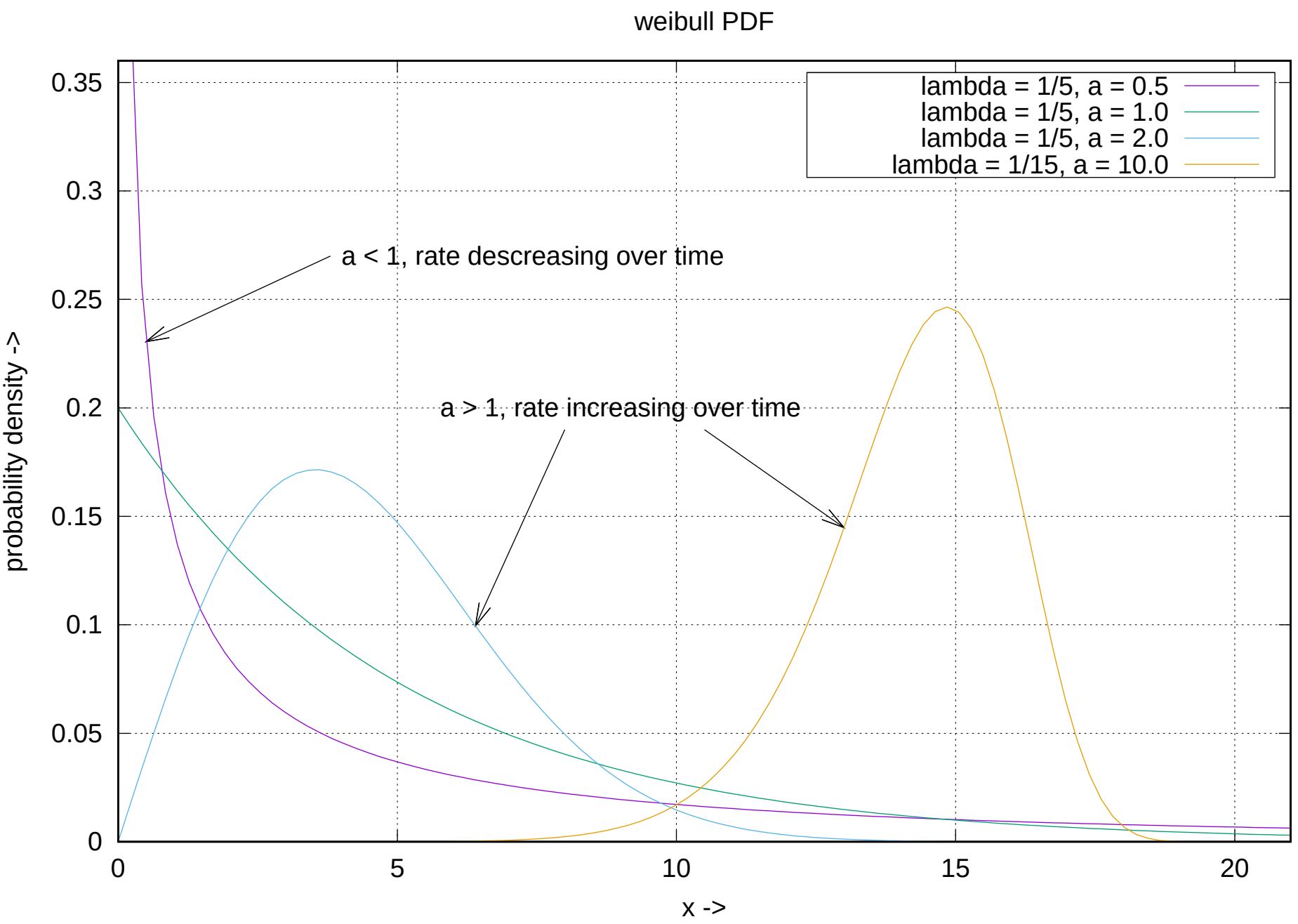

t CDF (and Gaussian limit)
probability density ->
x ->
1.00
0.80
0.60
0.40
0.20
0.00
-4.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0
degrees of freedom
nu = 1
nu = 2
nu = 4
nu = 10
nu = 20
normal

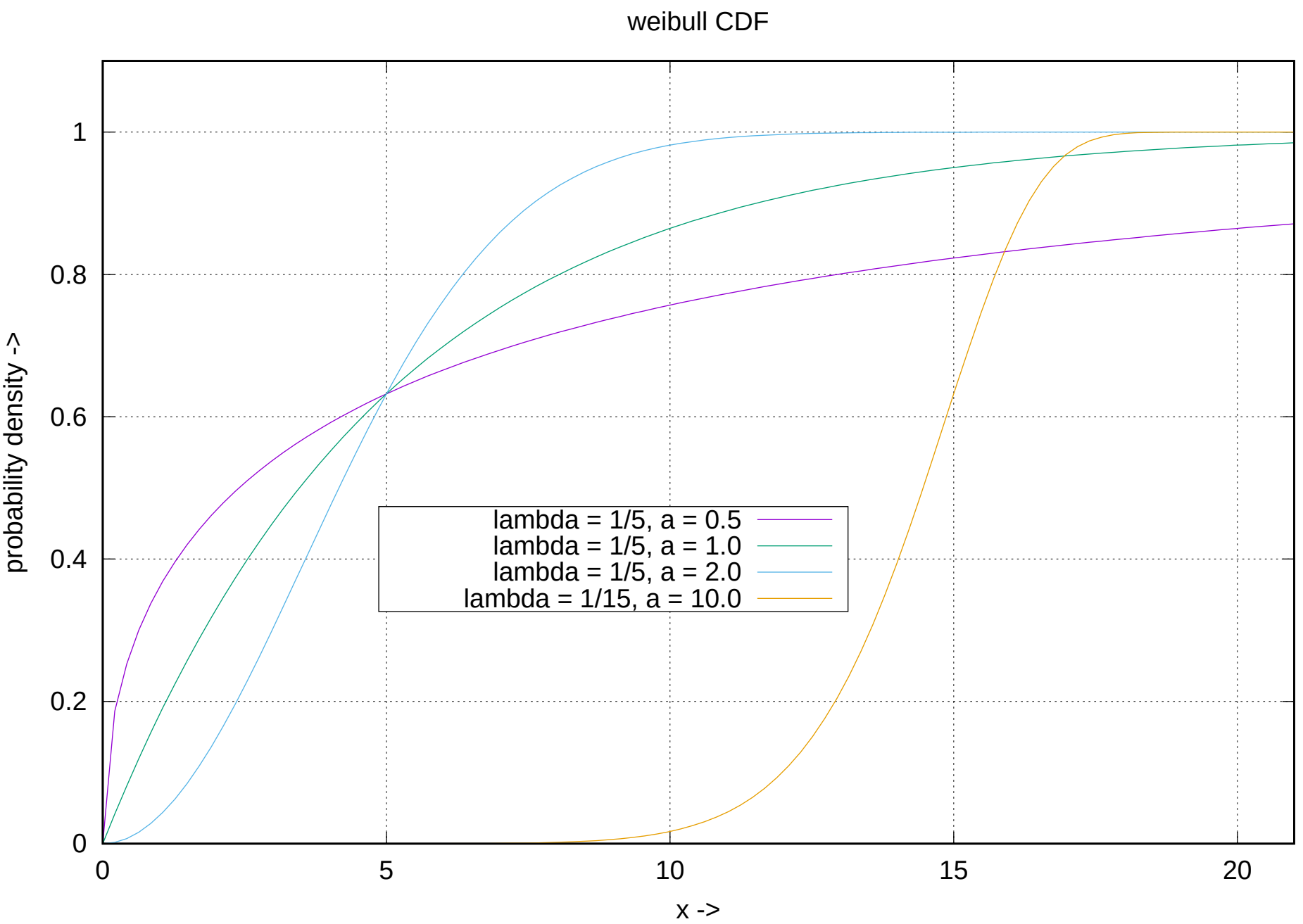

triangular PDF with m = 3.0, g = 2.0
probability density ->
0.50
0.40
0.30
0.20
0.10
0.00
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
x ->

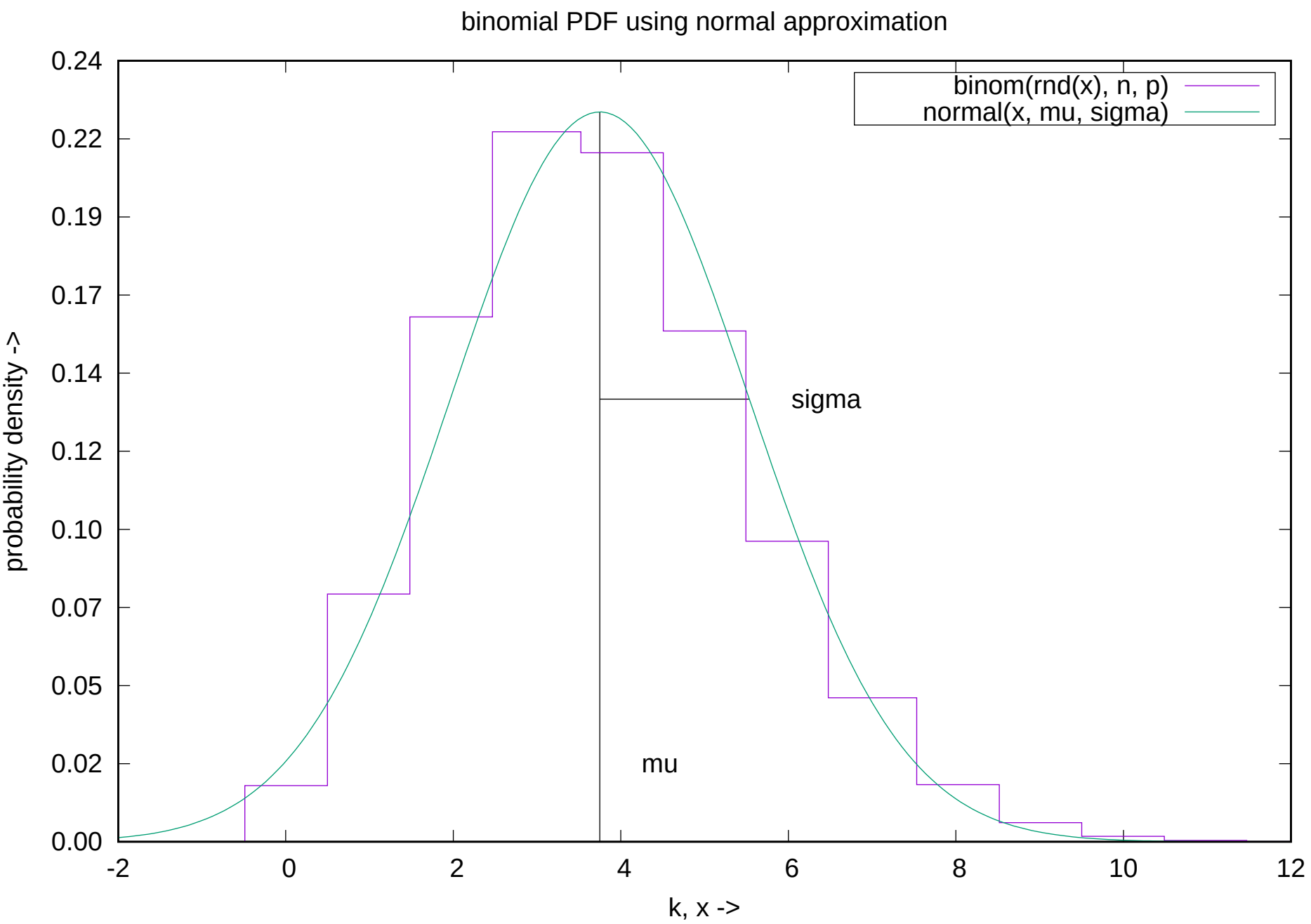

triangular CDF with m = 3.0, g = 2.0
probability density ->
1.00
0.80
0.60
0.40
0.20
0.00
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
x ->

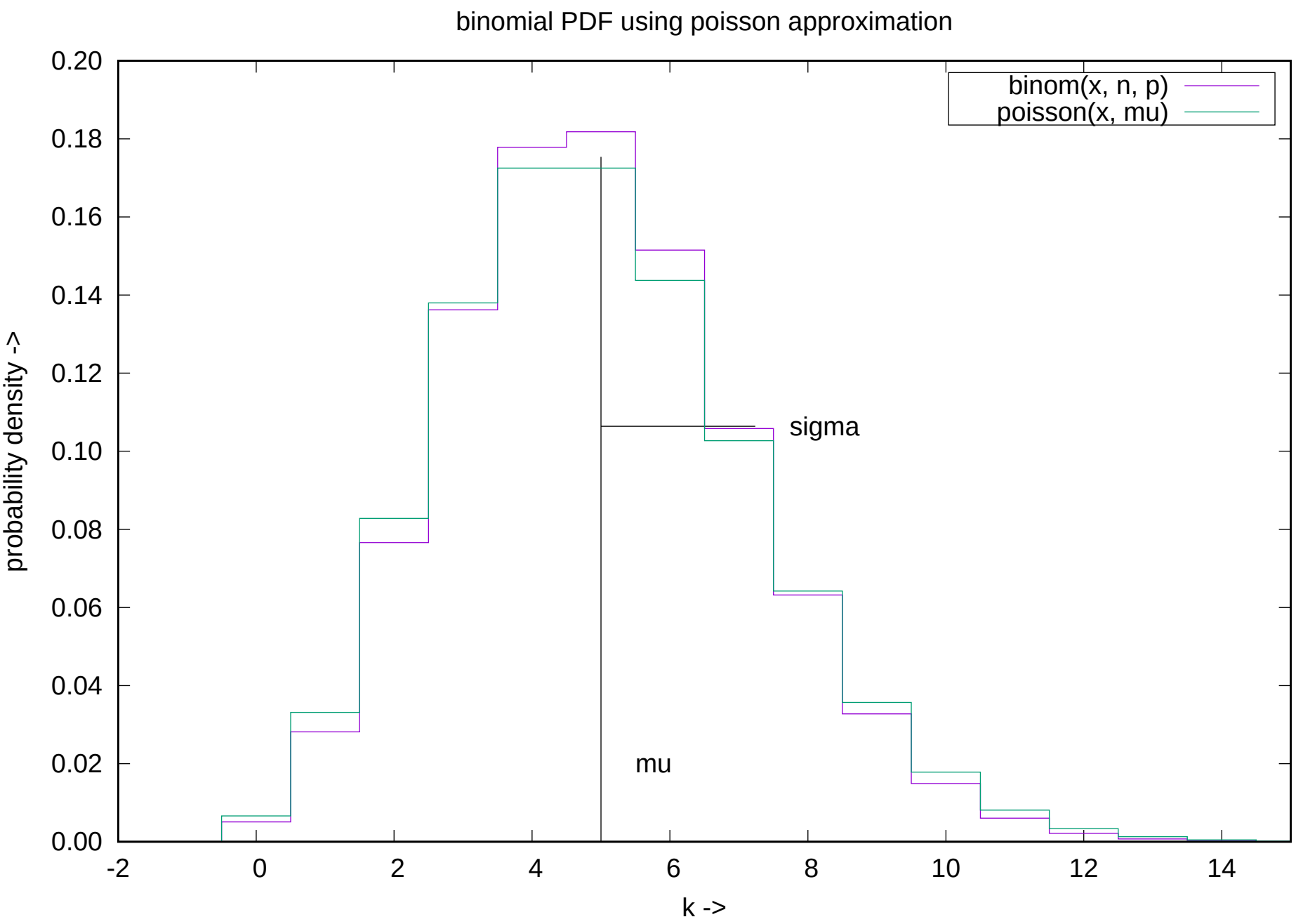

uniform PDF with a = -2.0, b = 2.0
probability density ->
0.25
0.20
0.15
0.10
0.05
0.00
-2.00
-1.50
-1.00
-0.50
0.00
0.50
1.00
1.50
2.00
x ->

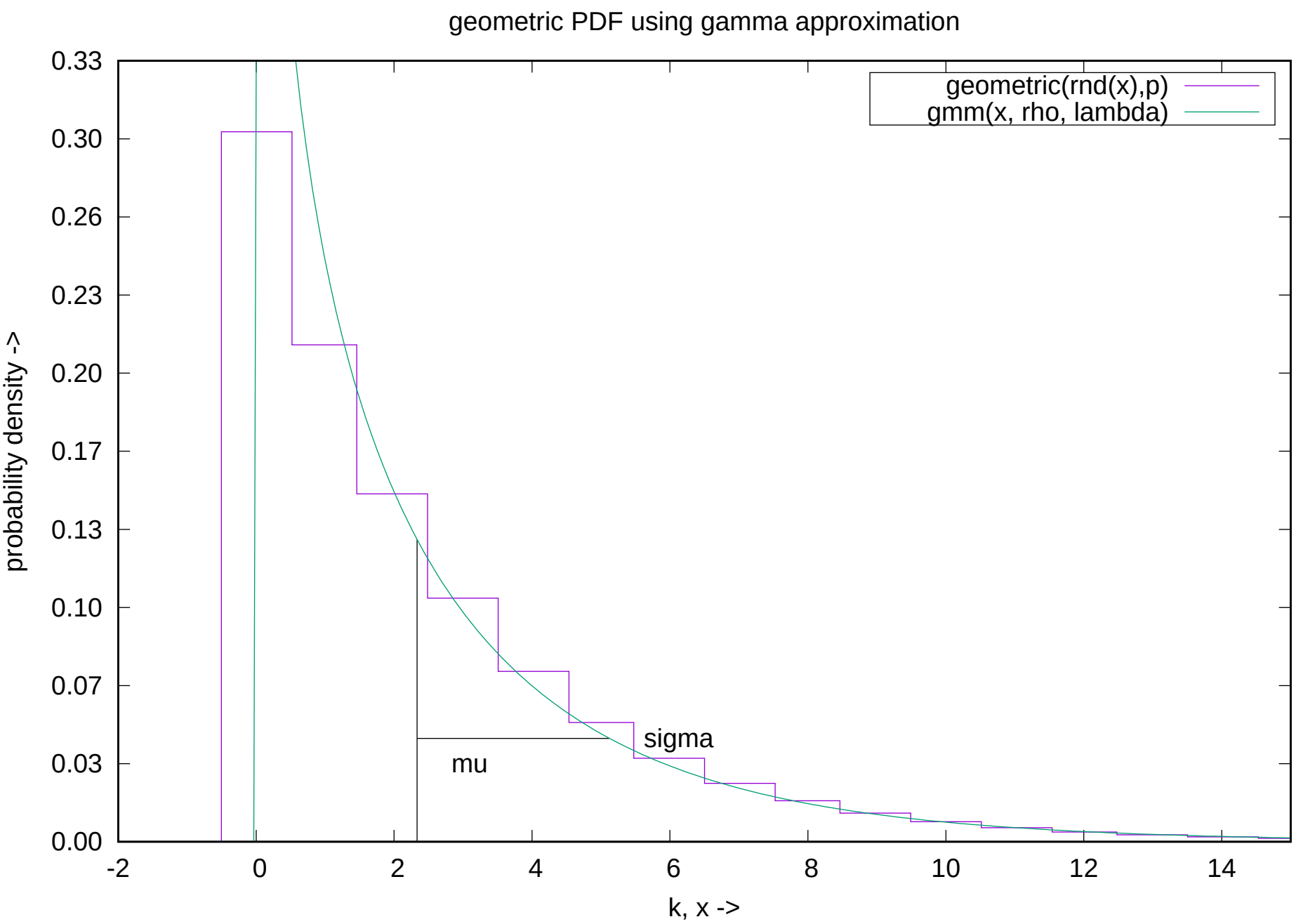

uniform CDF with a = -2.0, b = 2.0
probability density ->
1.00
0.80
0.60
0.40
0.20
0.00
-2.00
-1.50
-1.00
-0.50
0.00
0.50
1.00
1.50
2.00
x ->

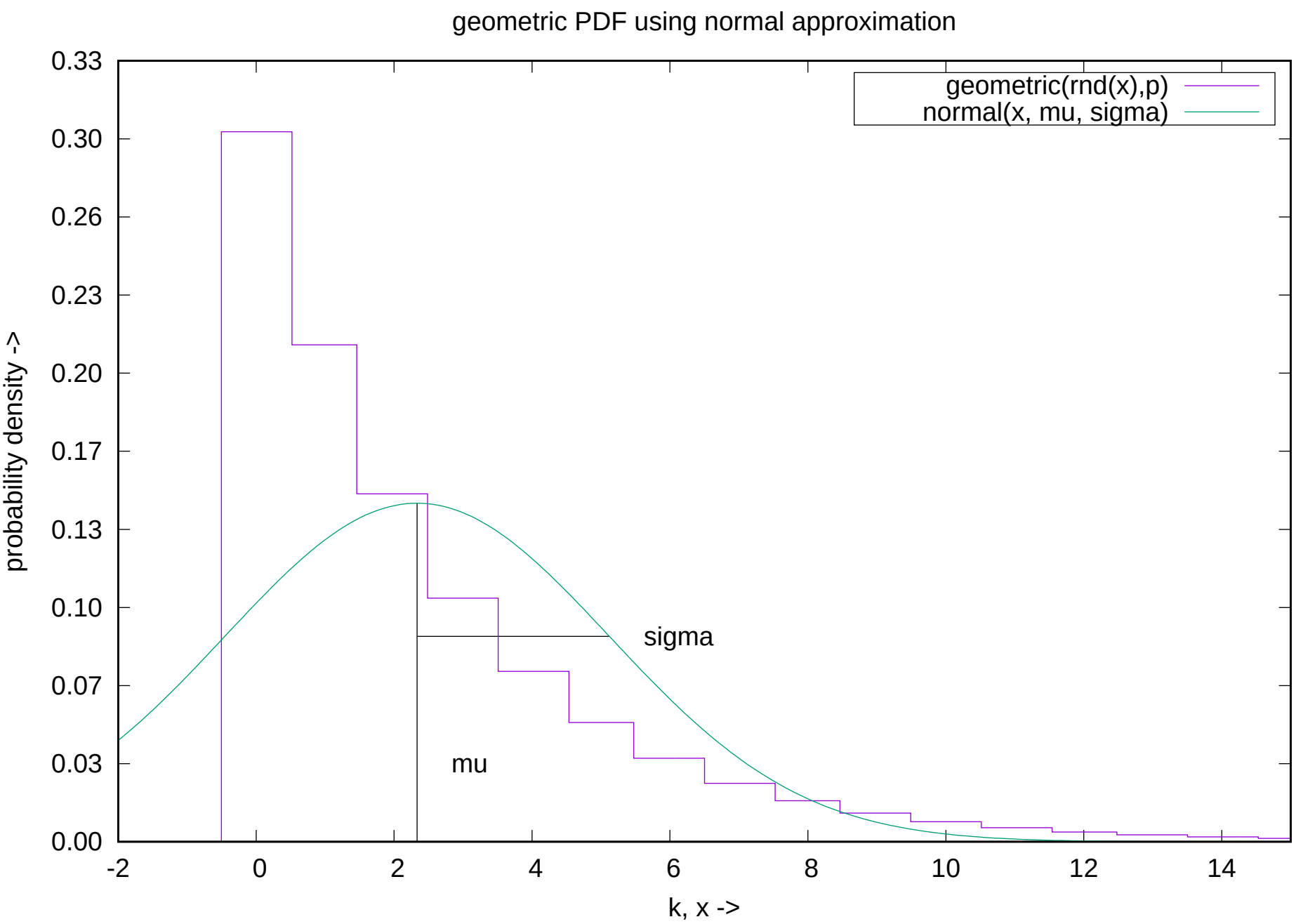

weibull PDF
lambda = 1/5, a = 0.5
lambda = 1/5, a = 1.0
lambda = 1/5, a = 2.0
lambda = 1/15, a = 10.0
a < 1, rate descreasing over time
a > 1, rate increasing over time
probability density ->
x ->
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0
5
10
15
20

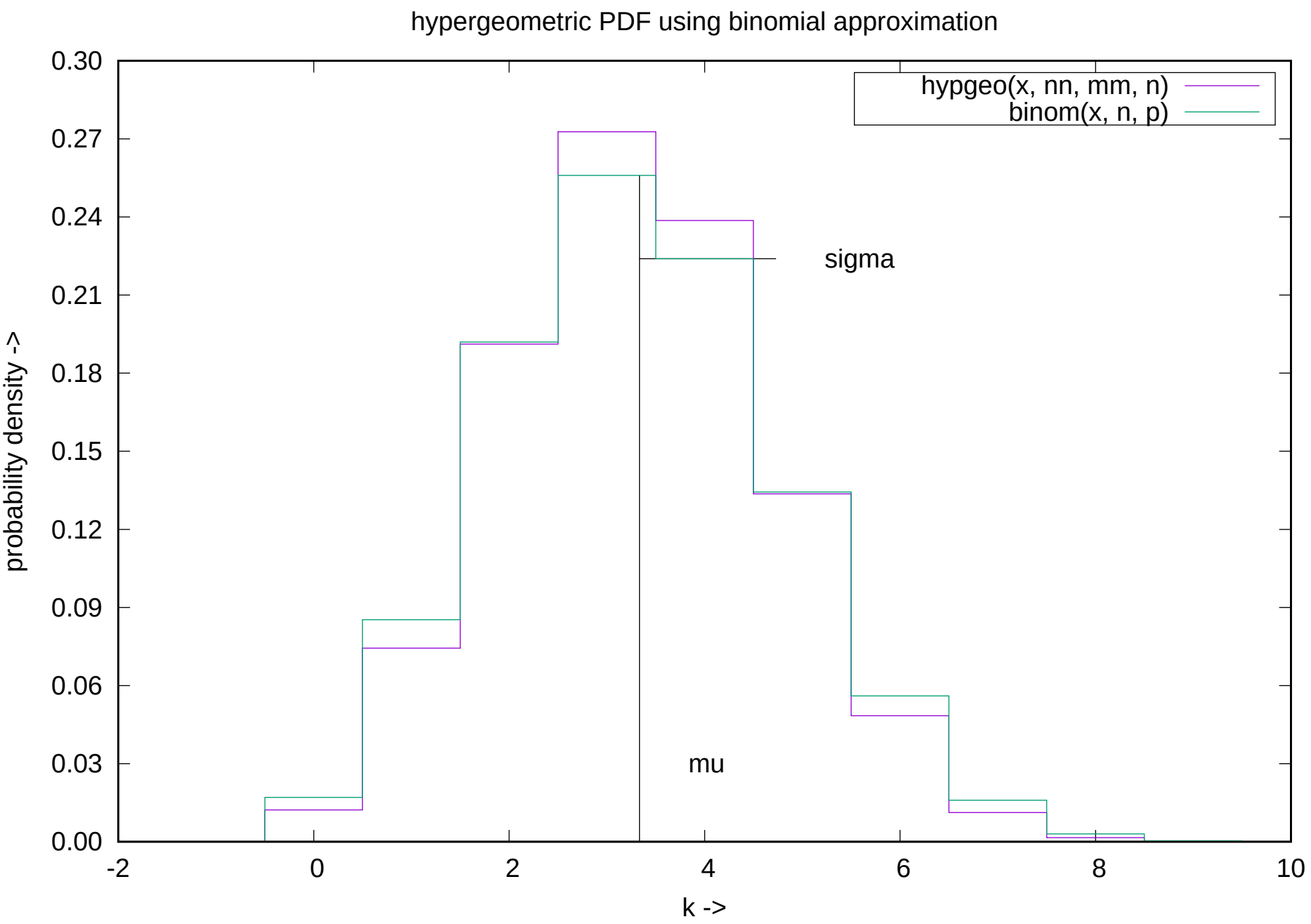

weibull CDF
probability density ->
x ->
0
0.2
0.4
0.6
0.8
1
0
5
10
15
20
lambda = 1/5, a = 0.5
lambda = 1/5, a = 1.0
lambda = 1/5, a = 2.0
lambda = 1/15, a = 10.0

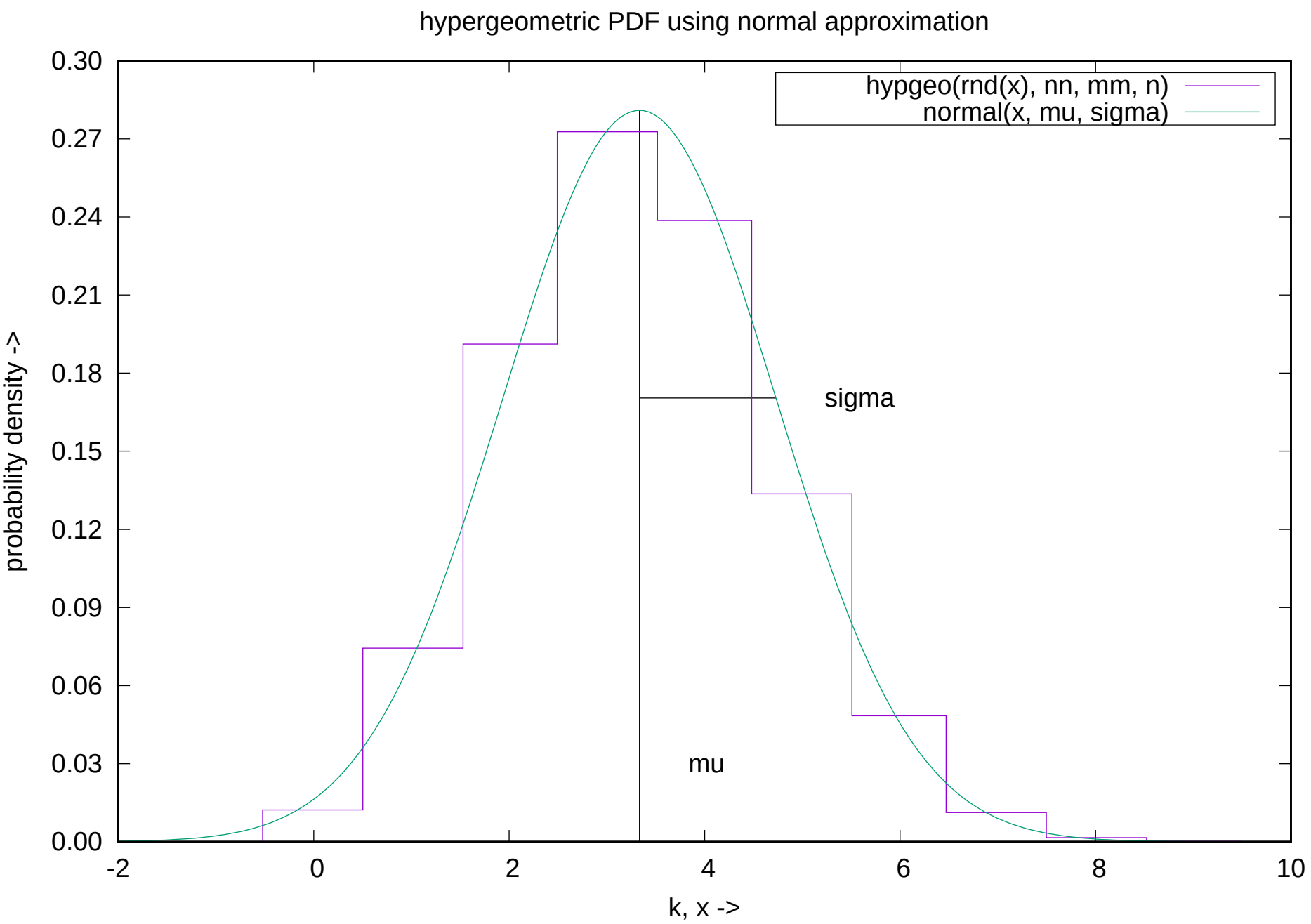

binomial PDF using normal approximation
binom(rnd(x), n, p)
normal(x, mu, sigma)
sigma
mu
probability density ->
k, x ->
0.24
0.22
0.19
0.17
0.14
0.12
0.10
0.07
0.05
0.02
0.00
-2
0
2
4
6
8
10
12

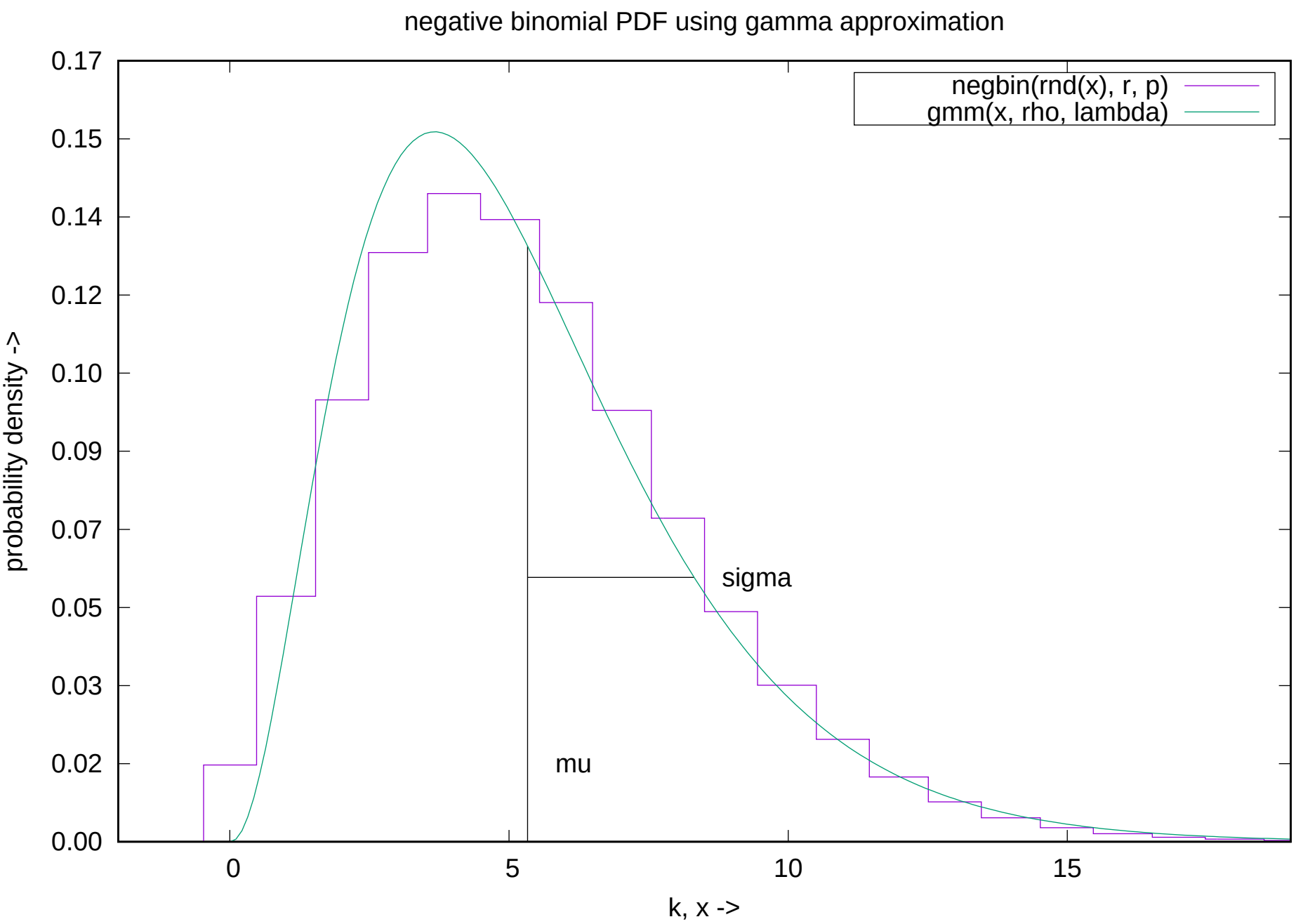

binomial PDF using poisson approximation
binom(x, n, p)
poisson(x, mu)
sigma
mu
probability density ->
k ->
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
-2
0
2
4
6
8
10
12
14

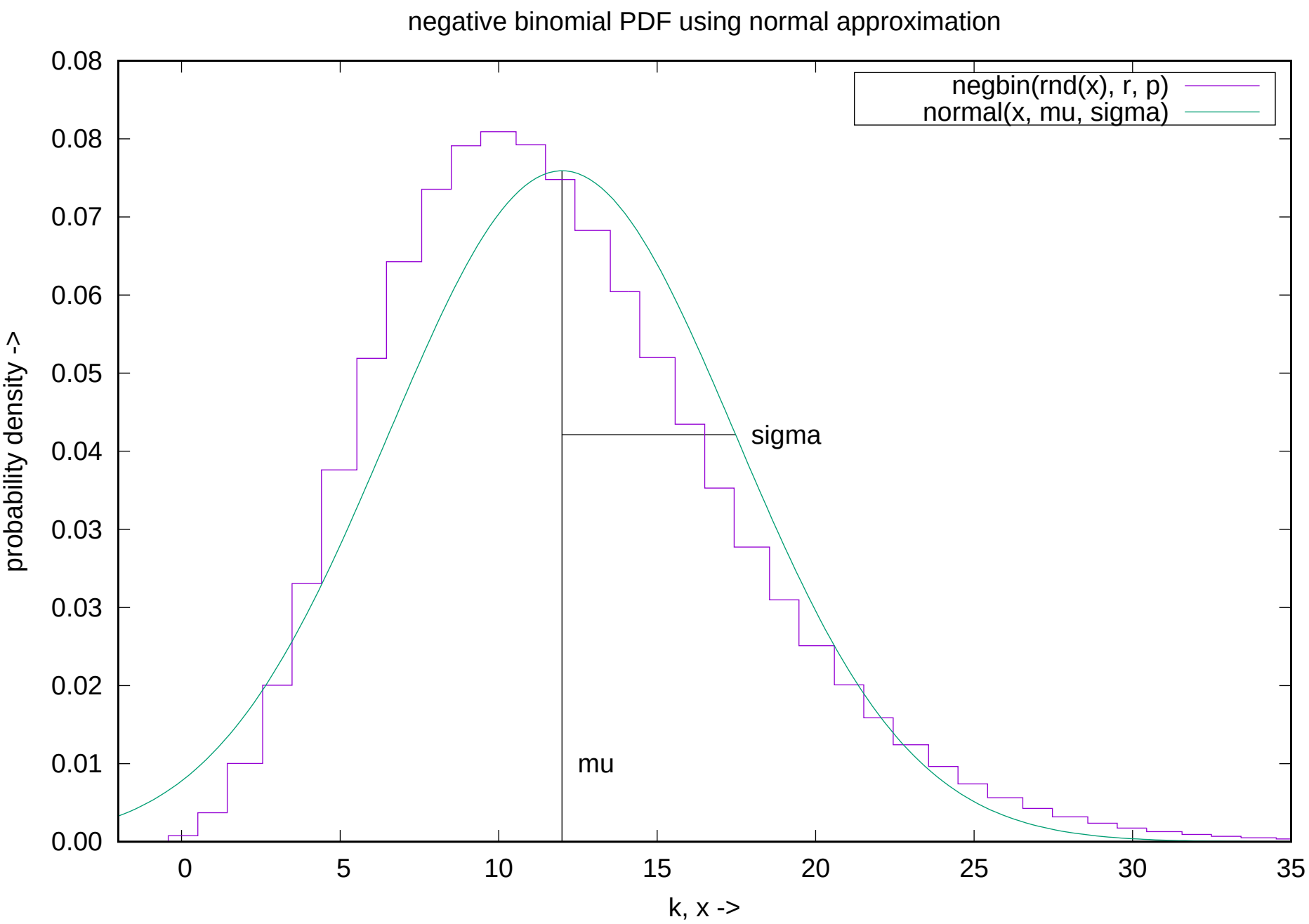

geometric PDF using gamma approximation
geometric(rnd(x),p)
gmm(x, rho, lambda)
probability density ->
k, x ->
sigma
mu
0.33
0.30
0.26
0.23
0.20
0.17
0.13
0.10
0.07
0.03
0.00
-2
0
2
4
6
8
10
12
14

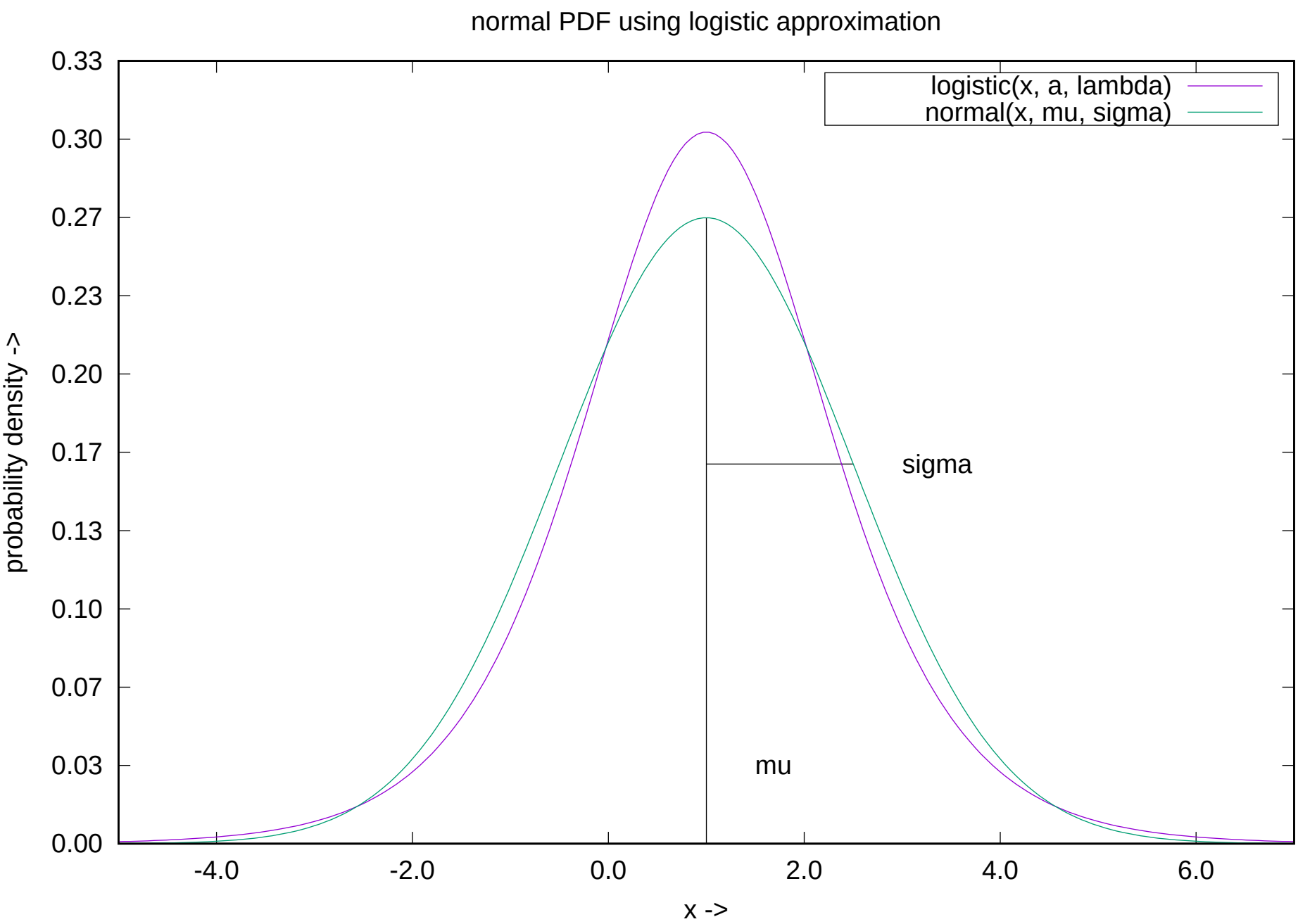

geometric PDF using normal approximation
geometric(rnd(x),p)
normal(x, mu, sigma)
probability density ->
k, x ->
sigma
mu
0.33
0.30
0.26
0.23
0.20
0.17
0.13
0.10
0.07
0.03
0.00
-2
0
2
4
6
8
10
12
14

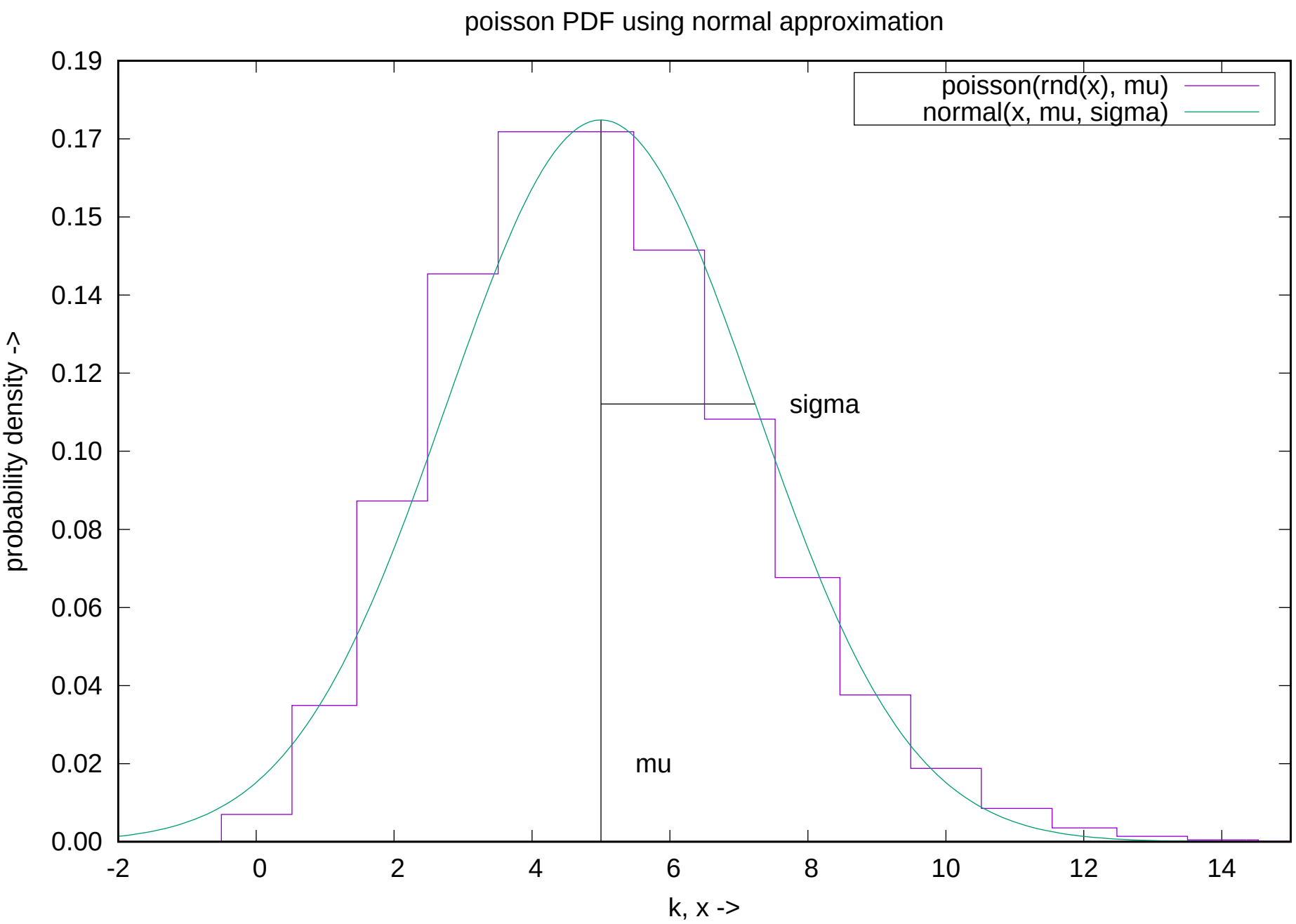

hypergeometric PDF using binomial approximation
hypgeo(x, nn, mm, n)
binom(x, n, p)
sigma
mu
probability density ->
k ->
0.30
0.27
0.24
0.21
0.18
0.15
0.12
0.09
0.06
0.03
0.00
-2
0
2
4
6
8
10

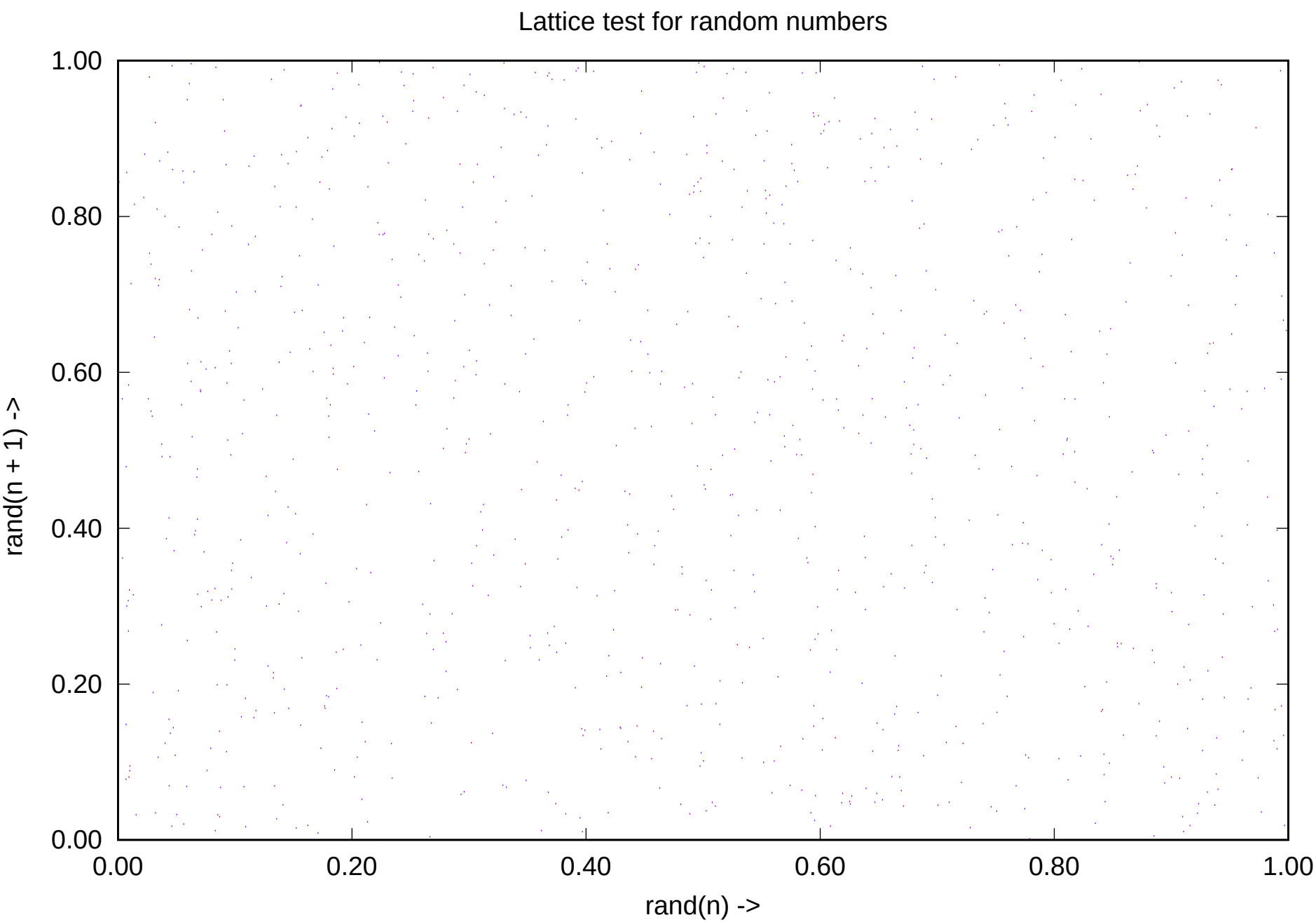

hypergeometric PDF using normal approximation
hypgeo(rnd(x), nn, mm, n)
normal(x, mu, sigma)
sigma
mu
probability density ->
k, x ->
0.30
0.27
0.24
0.21
0.18
0.15
0.12
0.09
0.06
0.03
0.00
-2
0
2
4
6
8
10

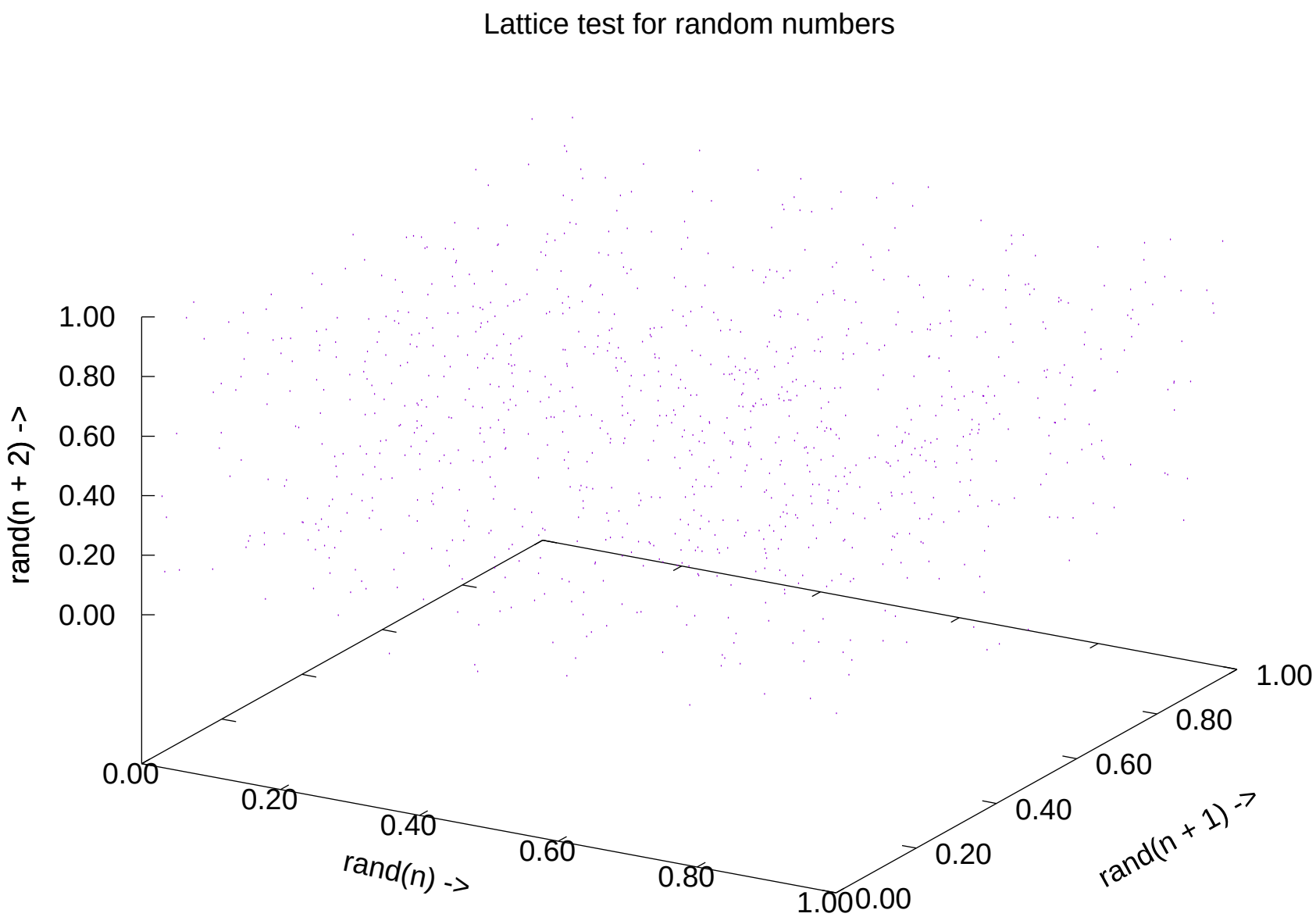

negative binomial PDF using gamma approximation
negbin(rnd(x), r, p)
gmm(x, rho, lambda)
sigma
mu
probability density ->
k, x ->
0.17
0.15
0.14
0.12
0.10
0.09
0.07
0.05
0.03
0.02
0.00
0
5
10
15


negative binomial PDF using normal approximation
negbin(rnd(x), r, p)
normal(x, mu, sigma)
probability density ->
k, x ->
sigma
mu
0.00
0.01
0.02
0.03
0.03
0.04
0.05
0.06
0.07
0.08
0.08
0
5
10
15
20
25
30
35

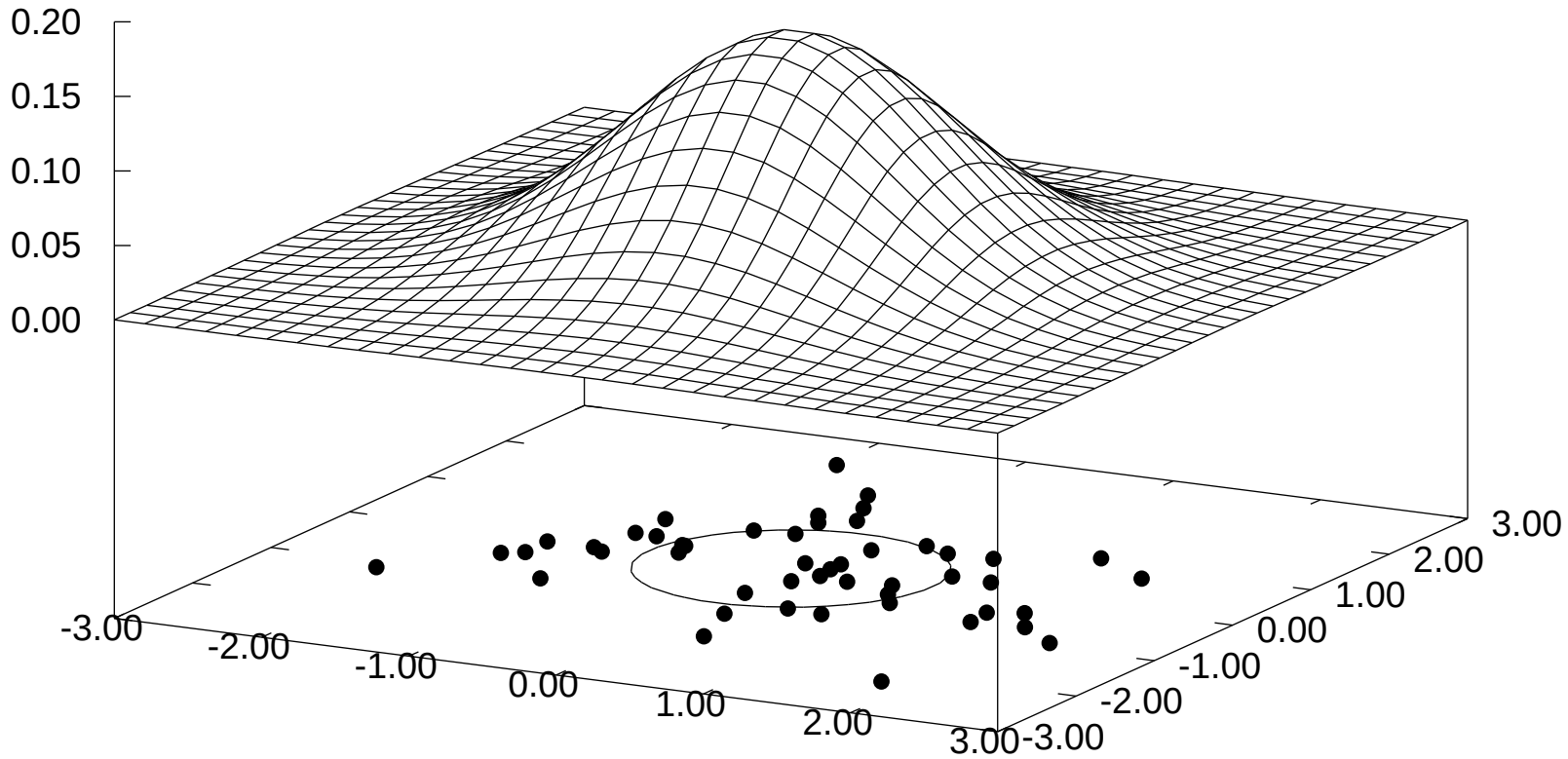

normal PDF using logistic approximation
logistic(x, a, lambda)
normal(x, mu, sigma)
sigma
mu
probability density ->
x ->
0.33
0.30
0.27
0.23
0.20
0.17
0.13
0.10
0.07
0.03
0.00
-4.0
-2.0
0.0
2.0
4.0
6.0

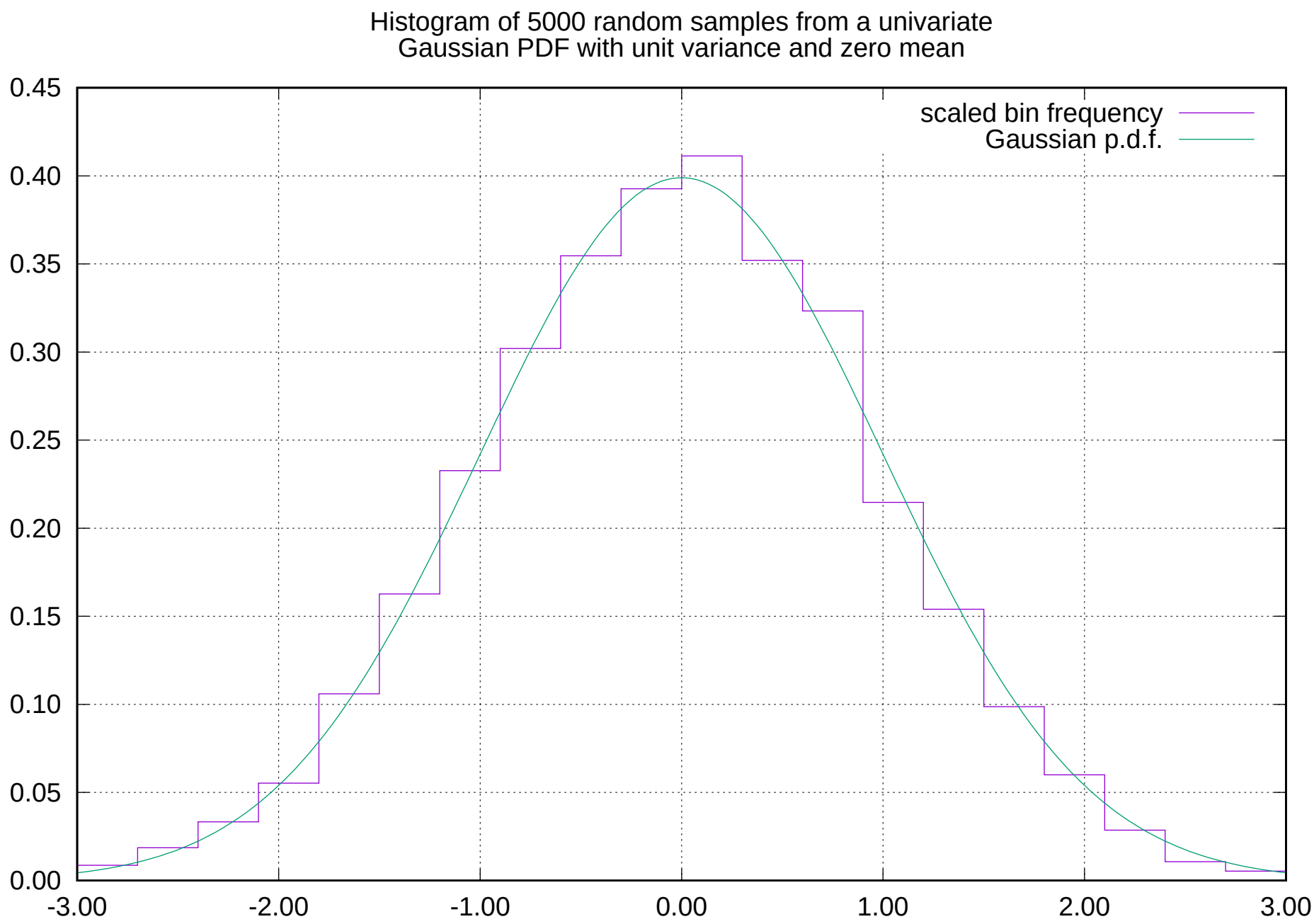

poisson PDF using normal approximation
poisson(rnd(x), mu)
normal(x, mu, sigma)
sigma
mu
probability density ->
k, x ->
0.19
0.17
0.15
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
-2
0
2
4
6
8
10
12
14


Lattice test for random numbers
1.00
0.80
0.60
0.40
0.20
0.00
rand(n + 1) ->
0.00
0.20
0.40
0.60
0.80
1.00
rand(n) ->

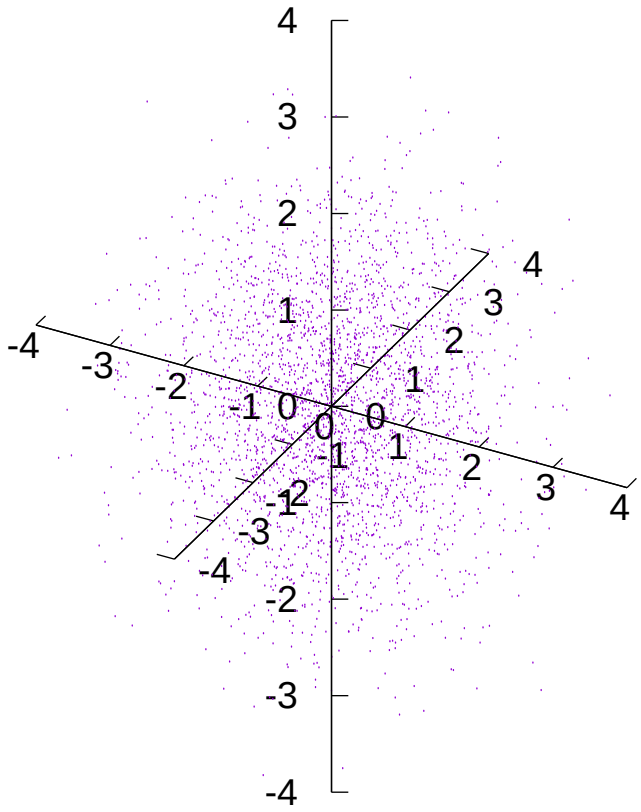

Lattice test for random numbers
1.00
0.80
0.60
0.40
0.20
0.00
rand(n + 2) ->
0.00
0.20
0.40
0.60
0.80
1.00
rand(n) ->
0.00
0.20
0.40
0.60
0.80
1.00
rand(n + 1) ->

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

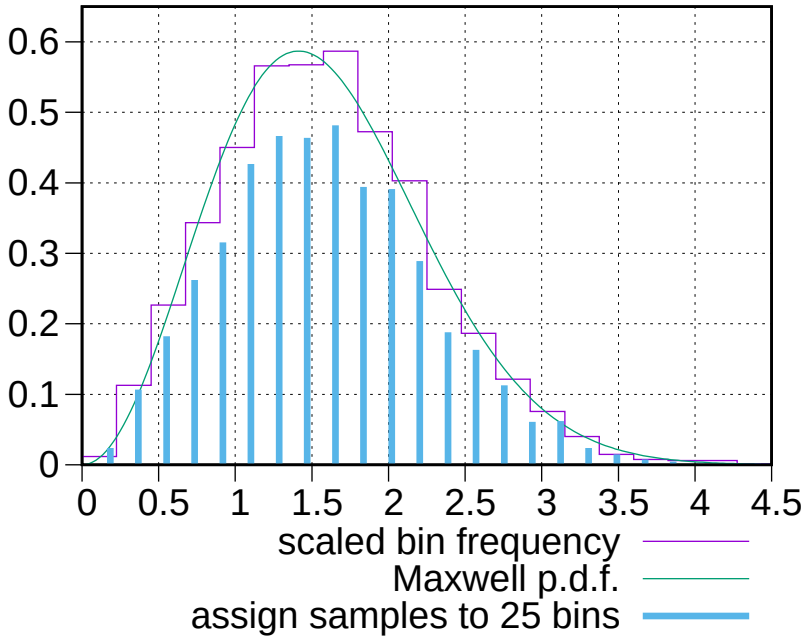



Histogram of 5000 random samples from a univariate
Gaussian PDF with unit variance and zero mean
scaled bin frequency
Gaussian p.d.f.
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-3.00
-2.00
-1.00
0.00
1.00
2.00
3.00

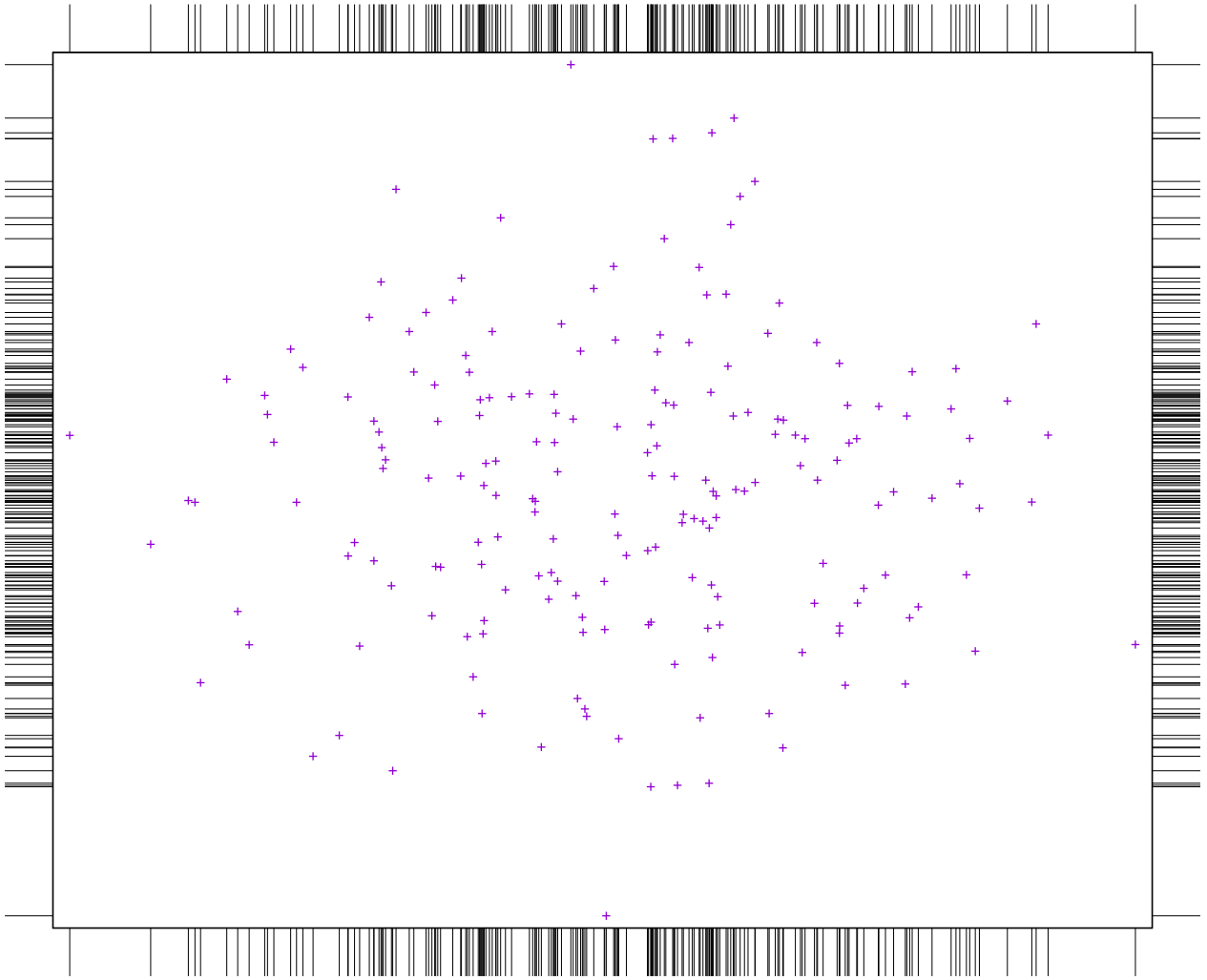
Gaussian 3D cloud of 3000 random samples

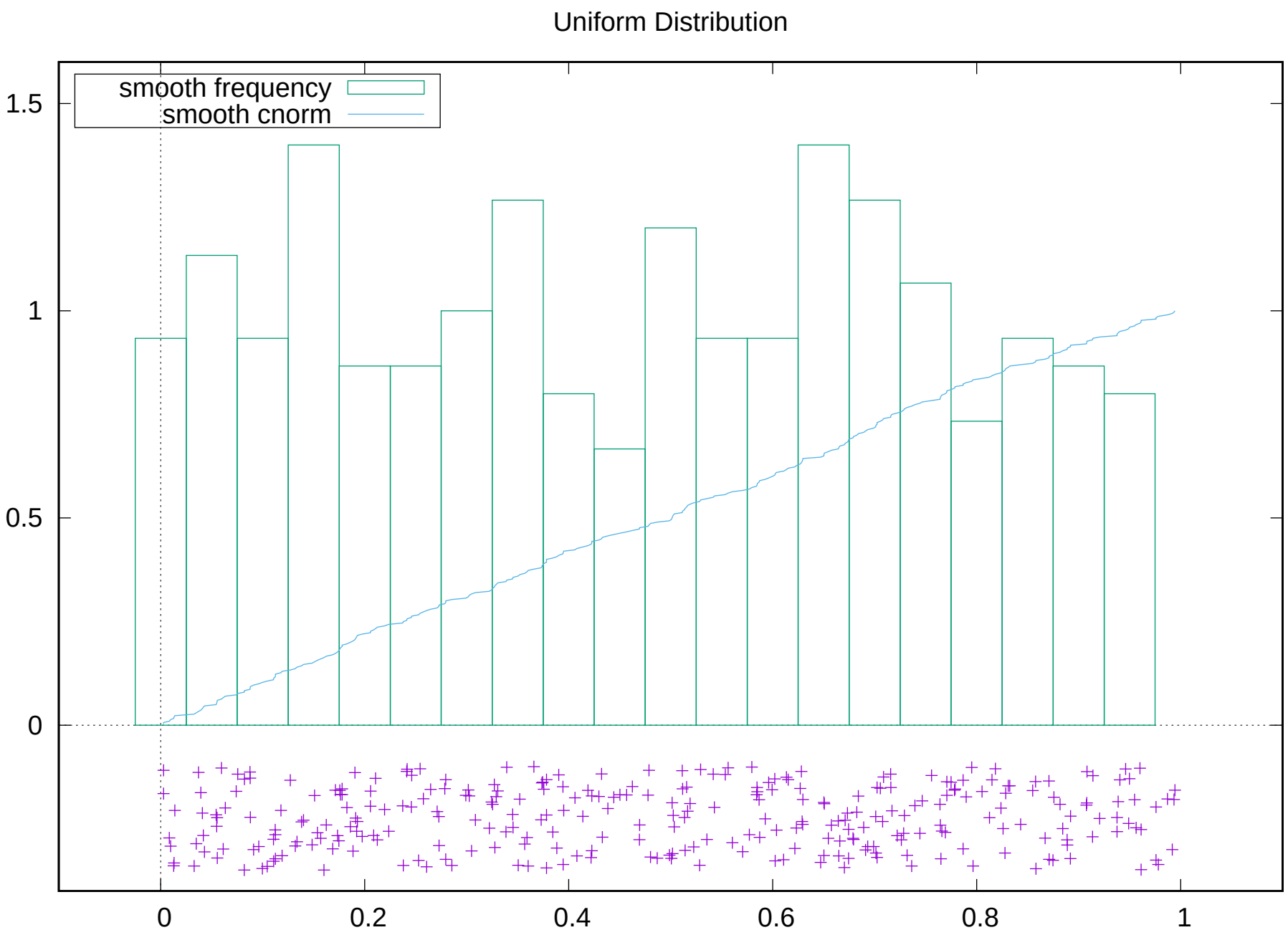


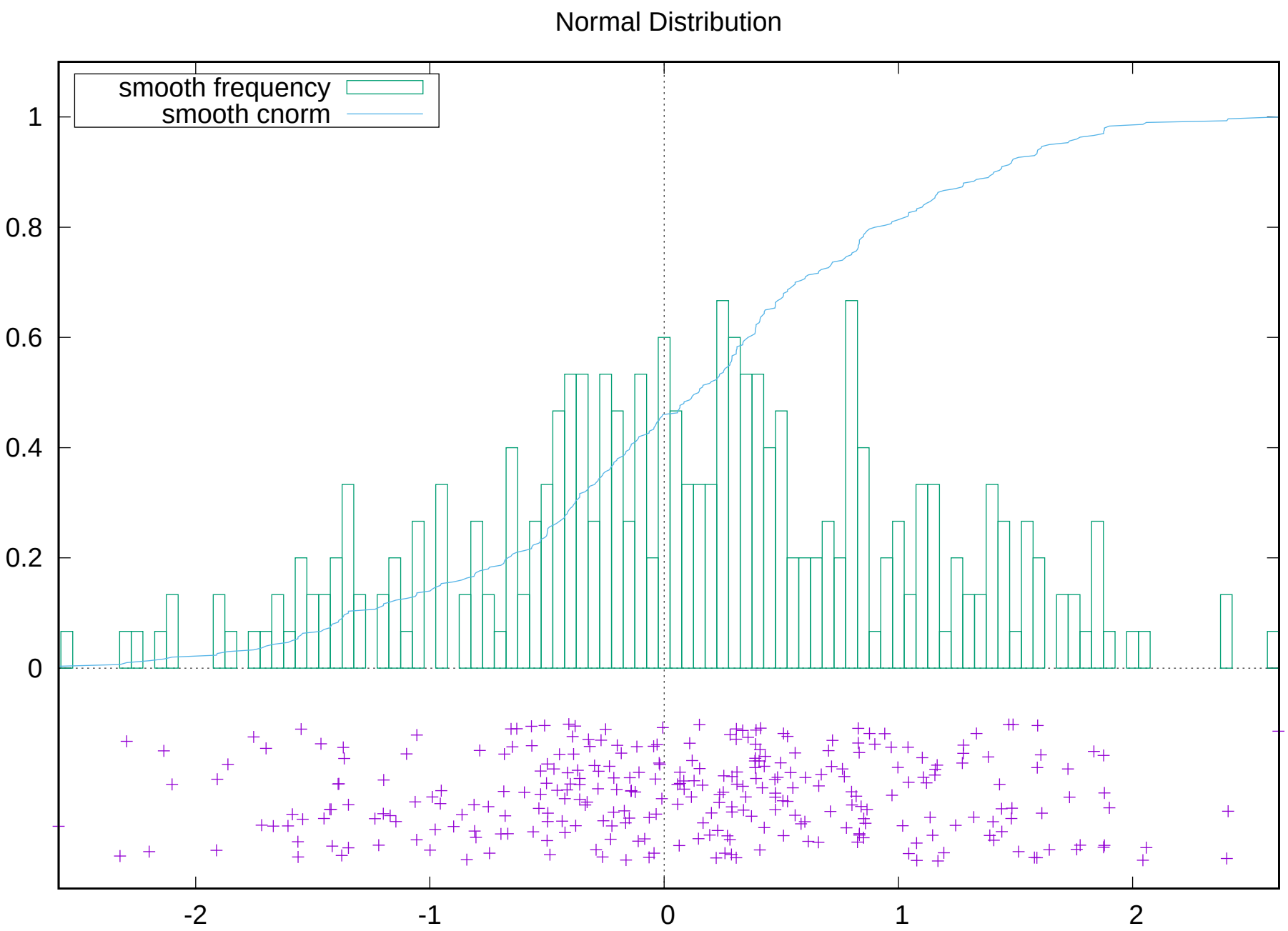
Histogram of distance from origin of
3000 multivariate unit variance samples

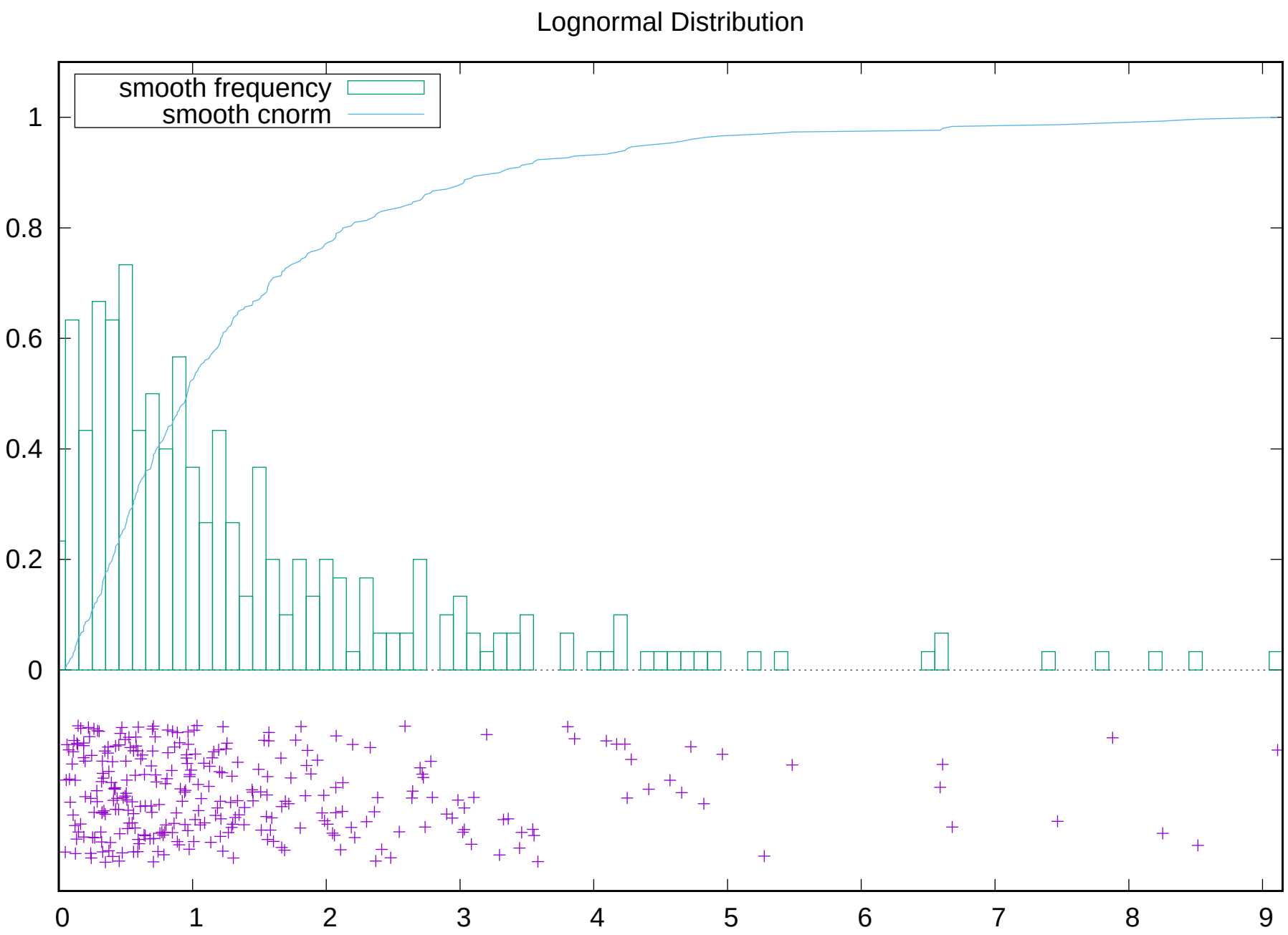


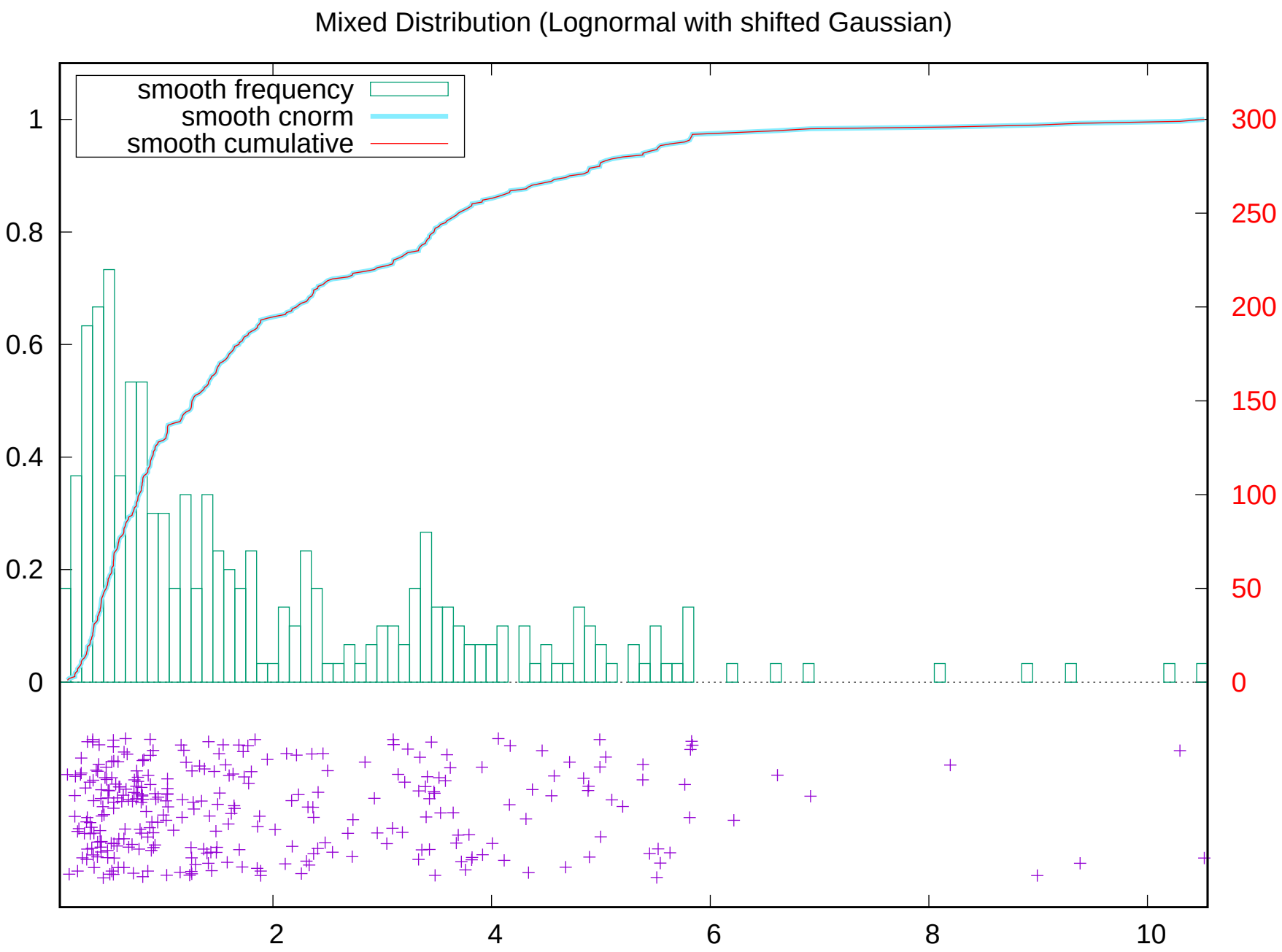
Rug Plot

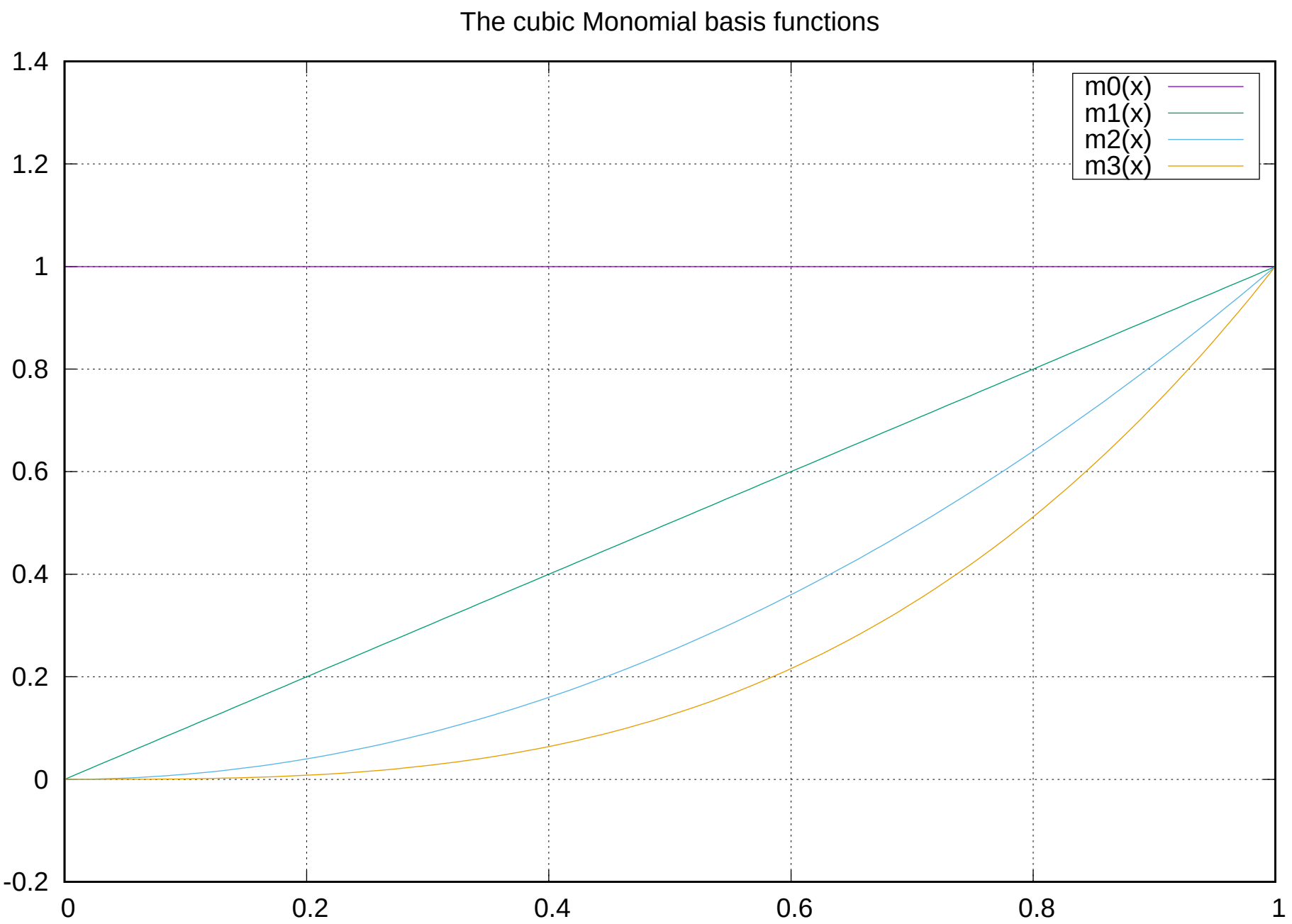


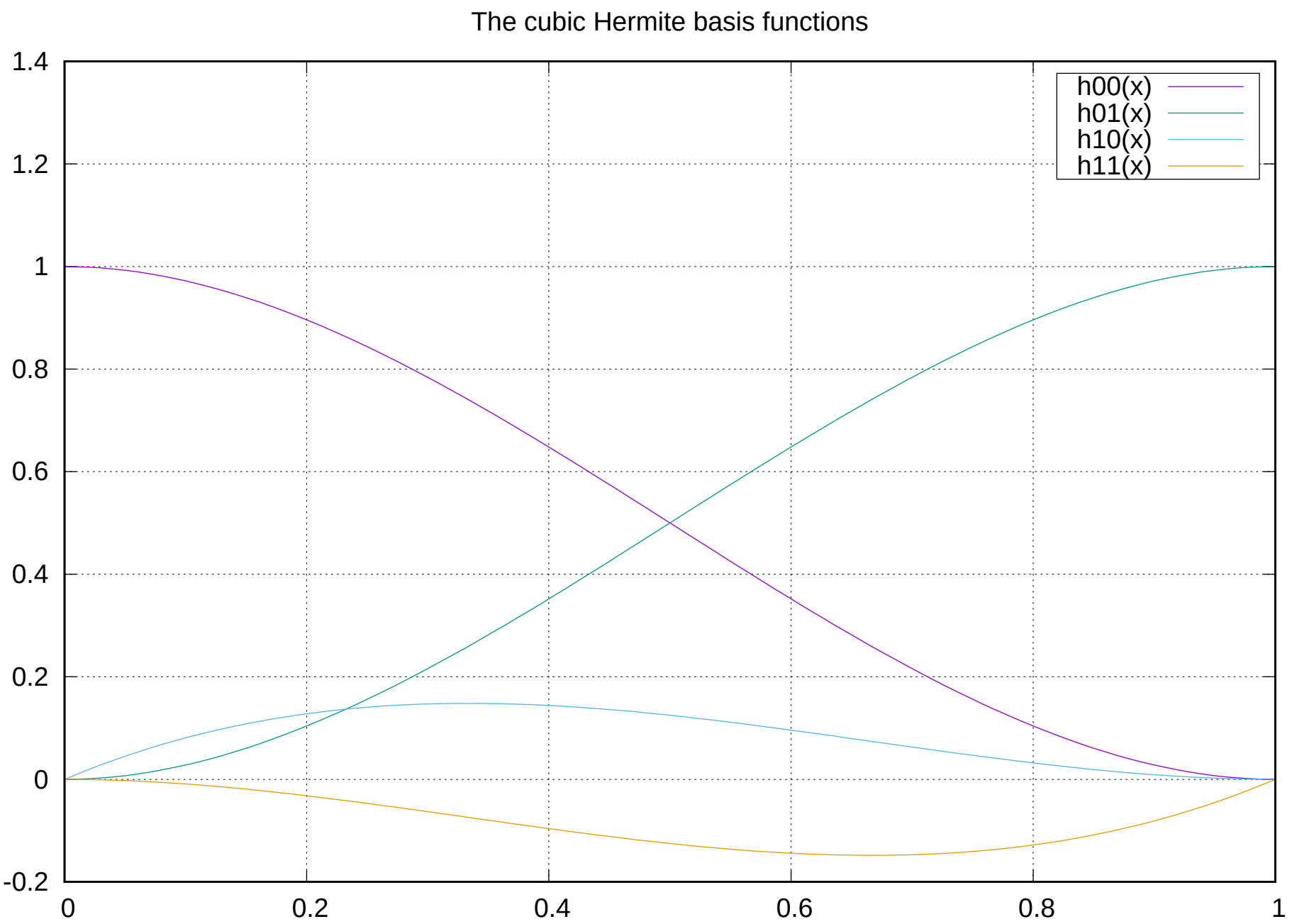

Uniform Distribution
smooth frequency
smooth cnorm
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1

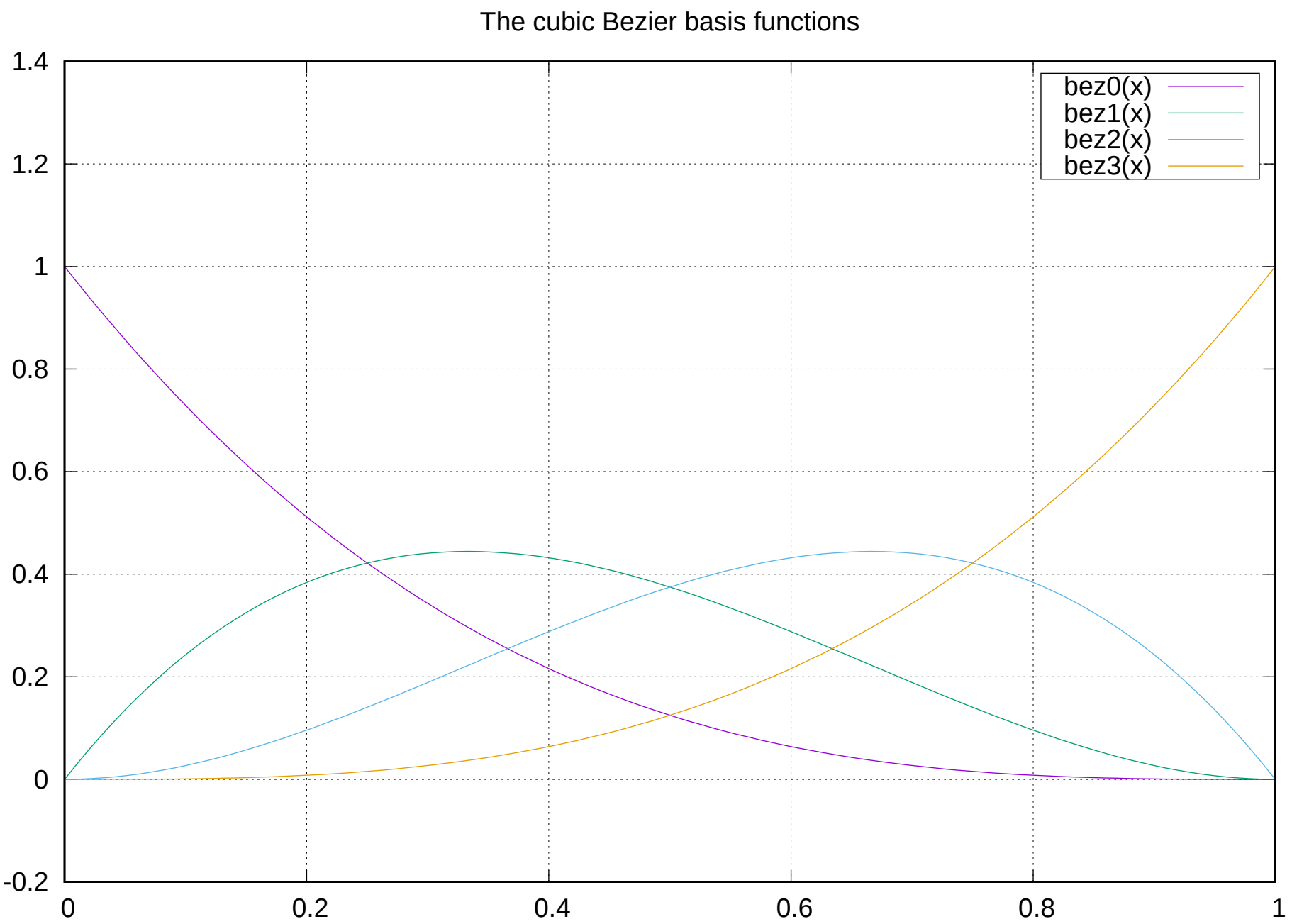

Normal Distribution
smooth frequency
smooth cnorm
1
0.8
0.6
0.4
0.2
0
-2
-1
0
1
2

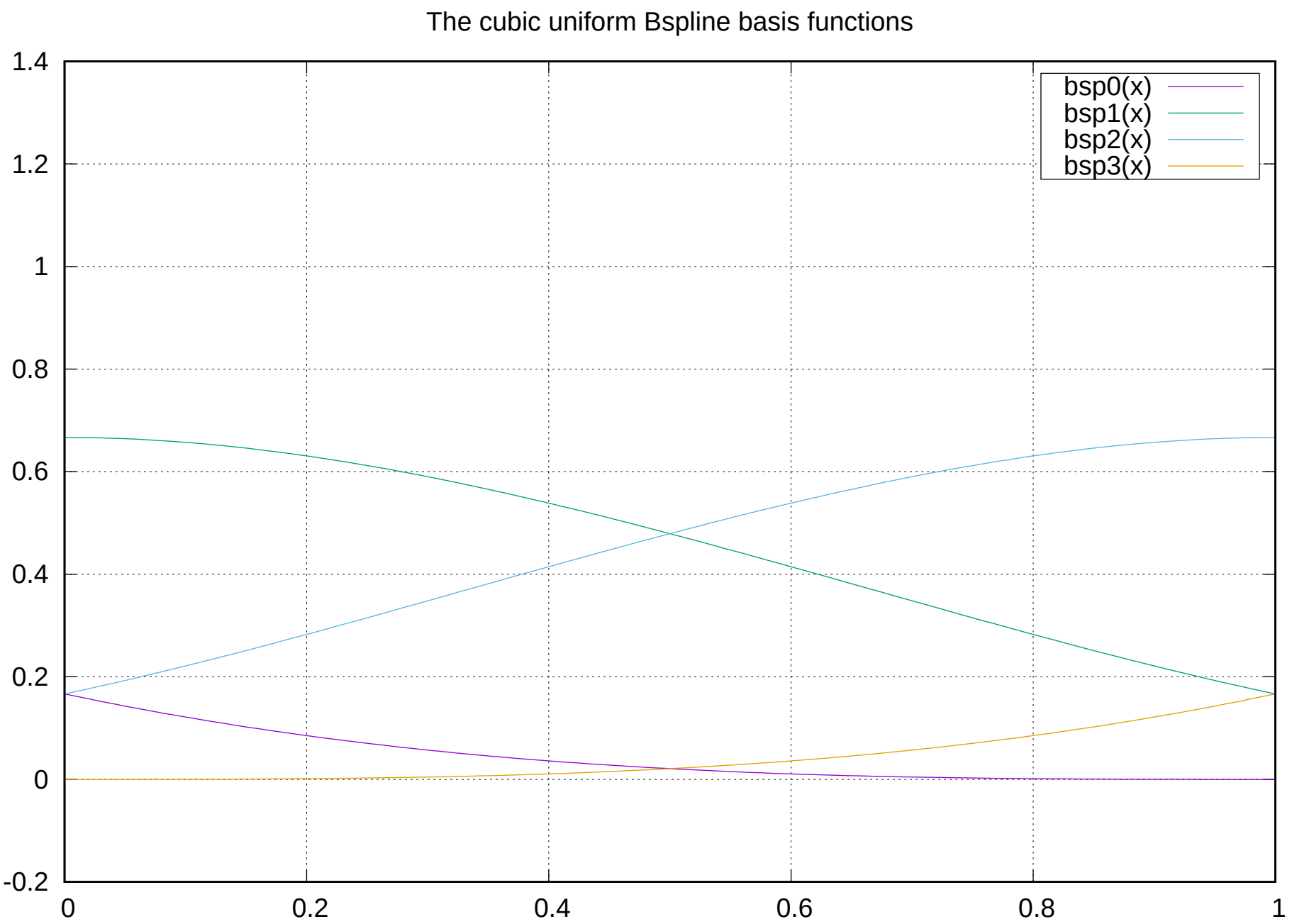

Lognormal Distribution
smooth frequency
smooth cnorm
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
7
8
9

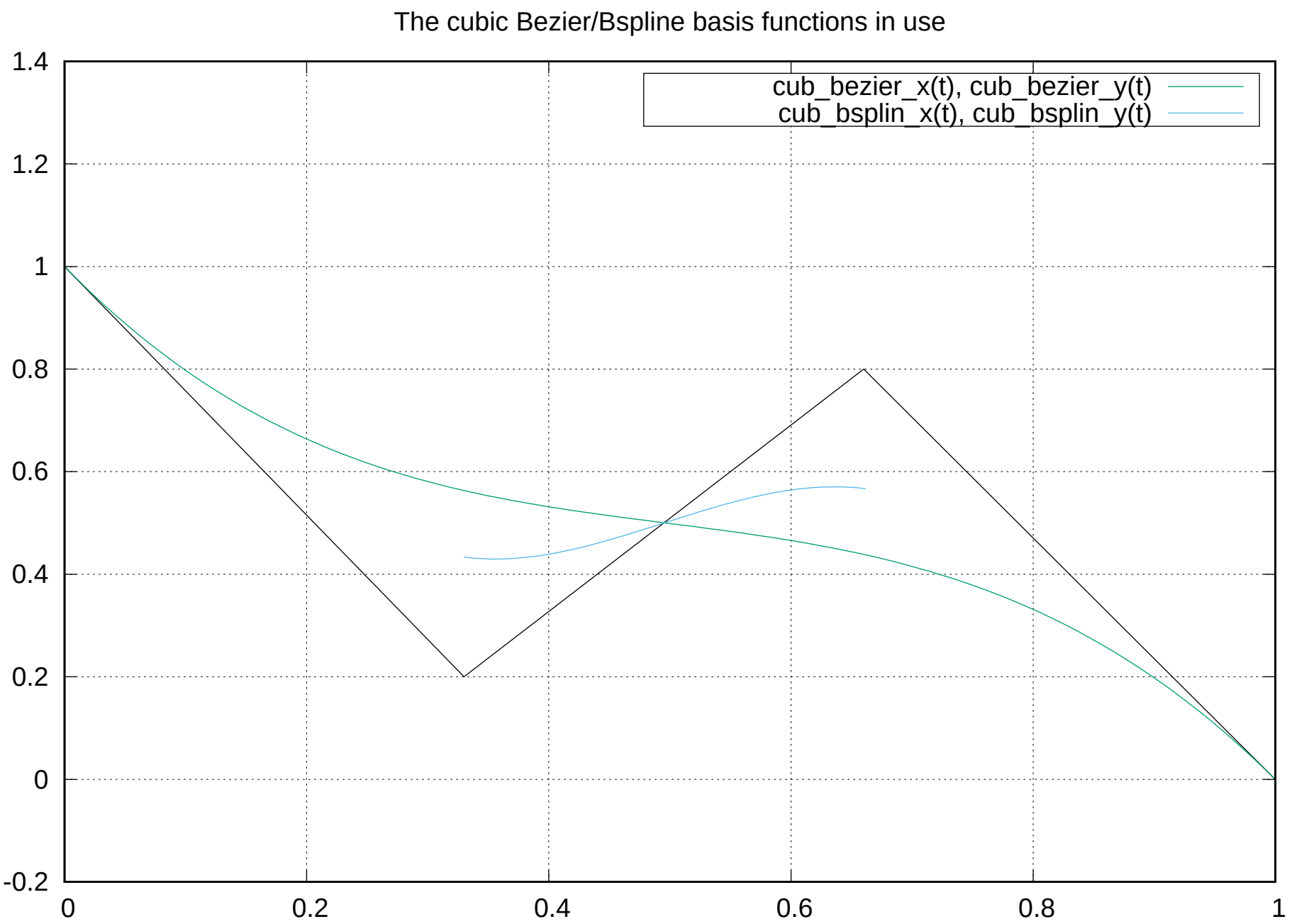

Mixed Distribution (Lognormal with shifted Gaussian)
smooth frequency
smooth cnorm
smooth cumulative
1
0.8
0.6
0.4
0.2
0
300
250
200
150
100
50
0
2
4
6
8
10

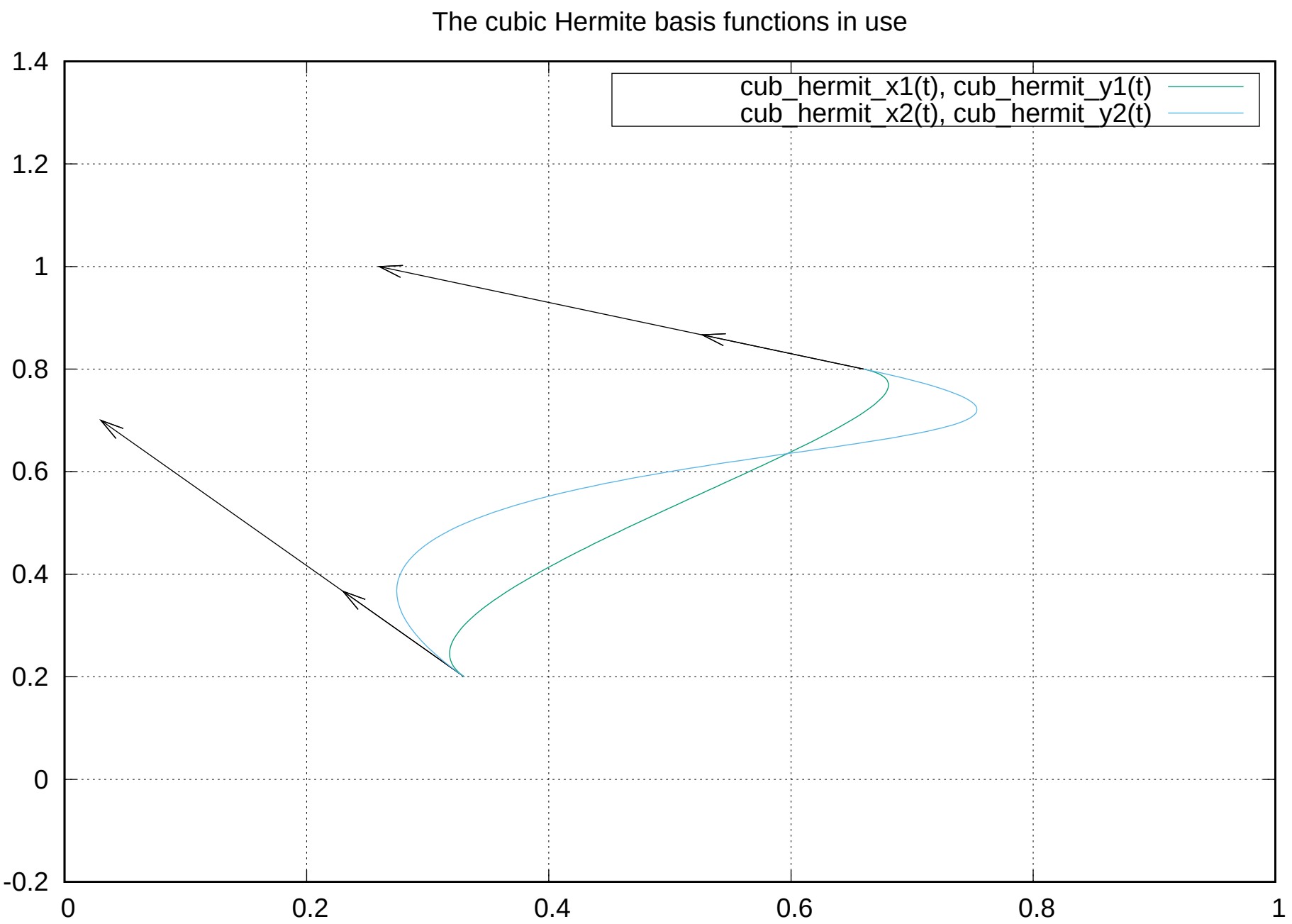

The cubic Monomial basis functions
m0(x)
m1(x)
m2(x)
m3(x)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1

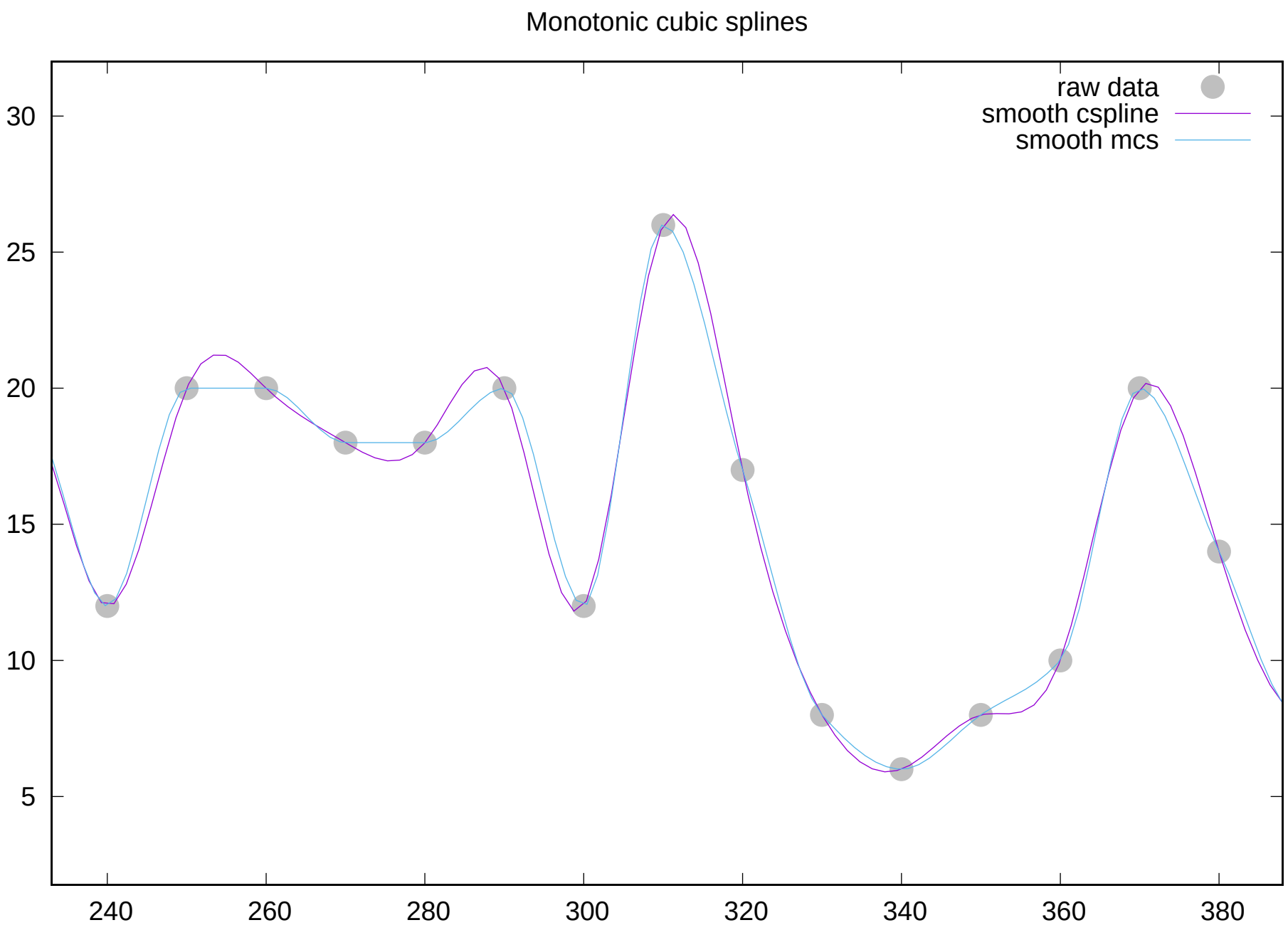

The cubic Hermite basis functions
h00(x)
h01(x)
h10(x)
h11(x)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1

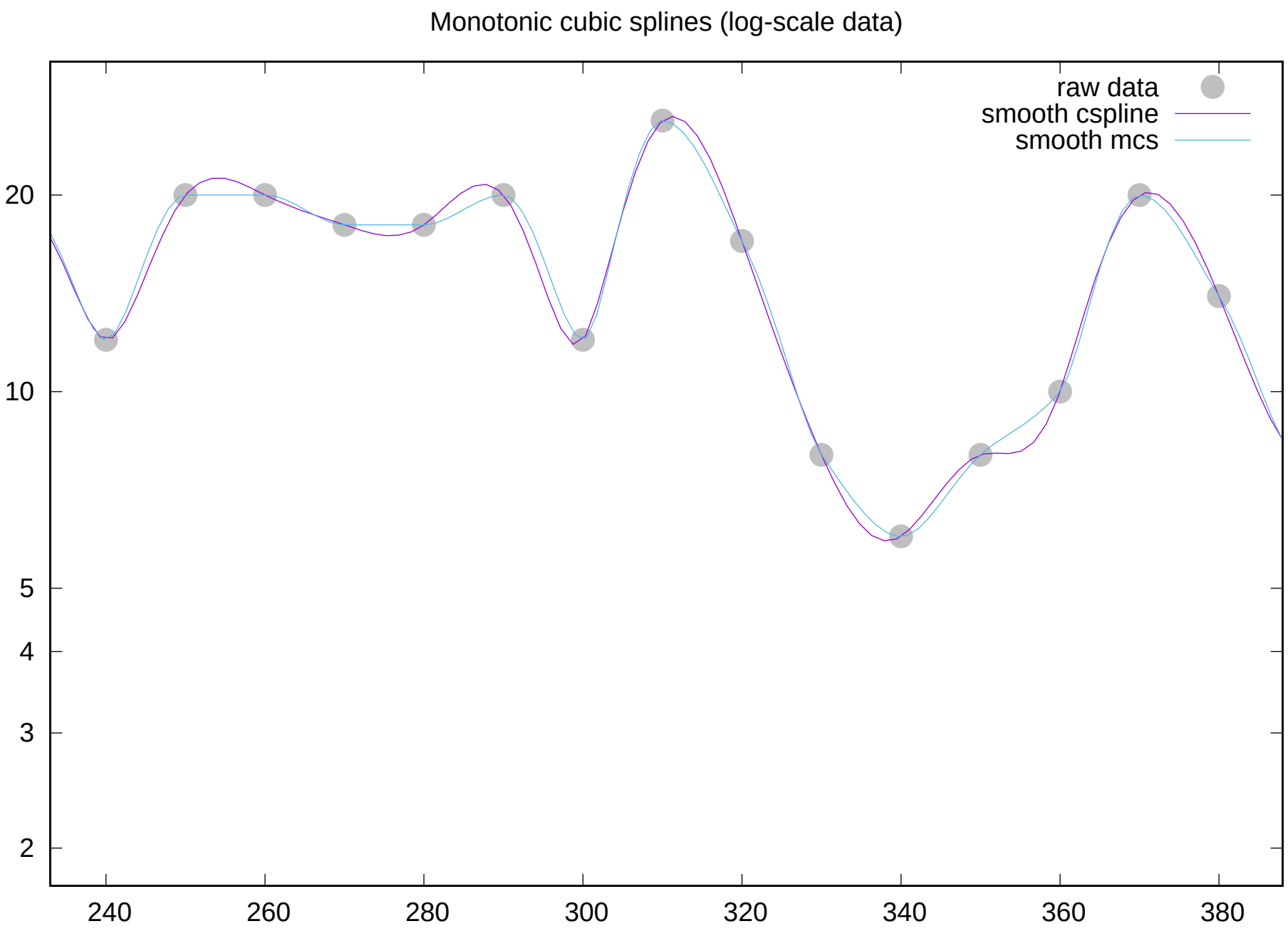

The cubic Bezier basis functions
bez0(x)
bez1(x)
bez2(x)
bez3(x)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1

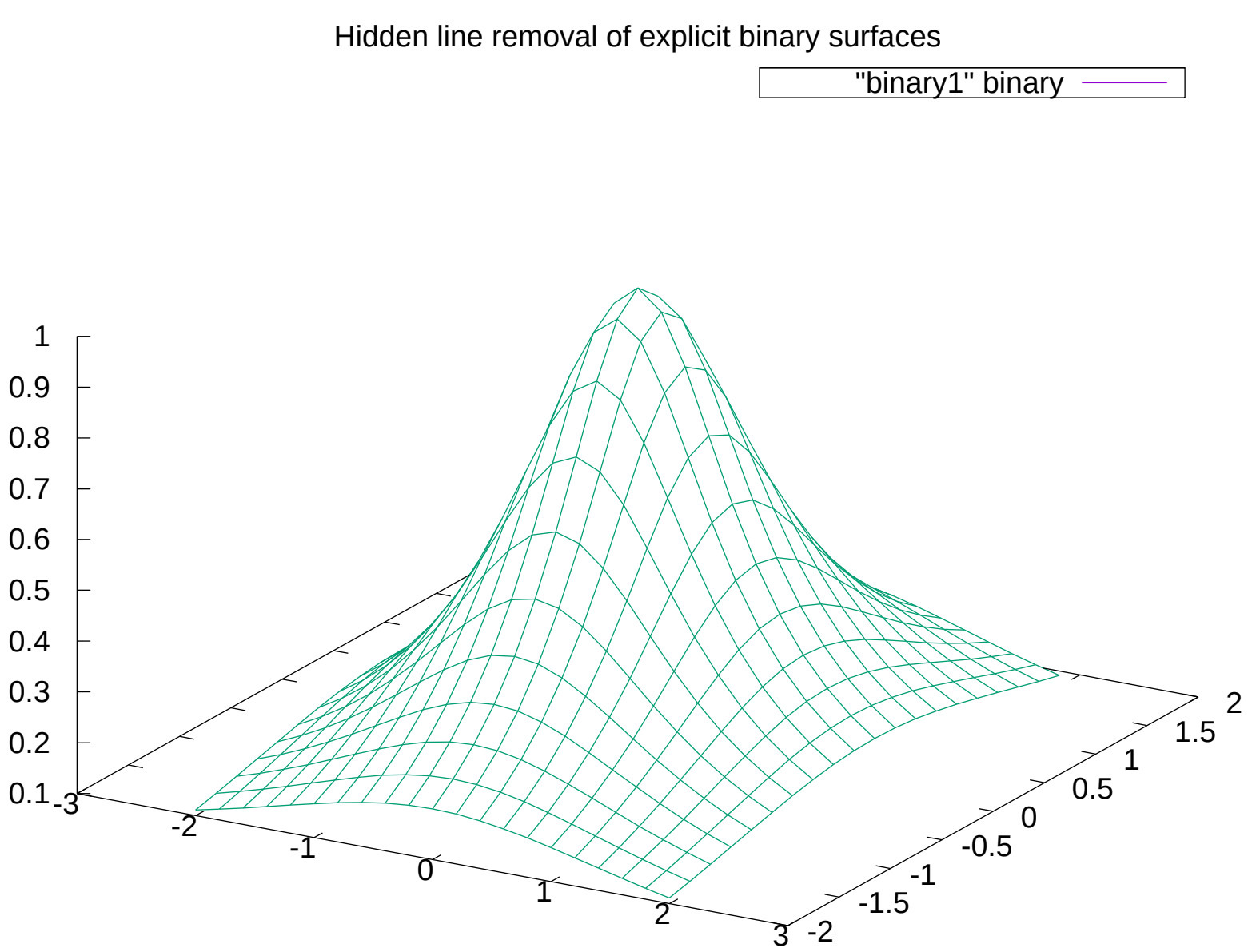

The cubic uniform Bspline basis functions
bsp0(x)
bsp1(x)
bsp2(x)
bsp3(x)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1

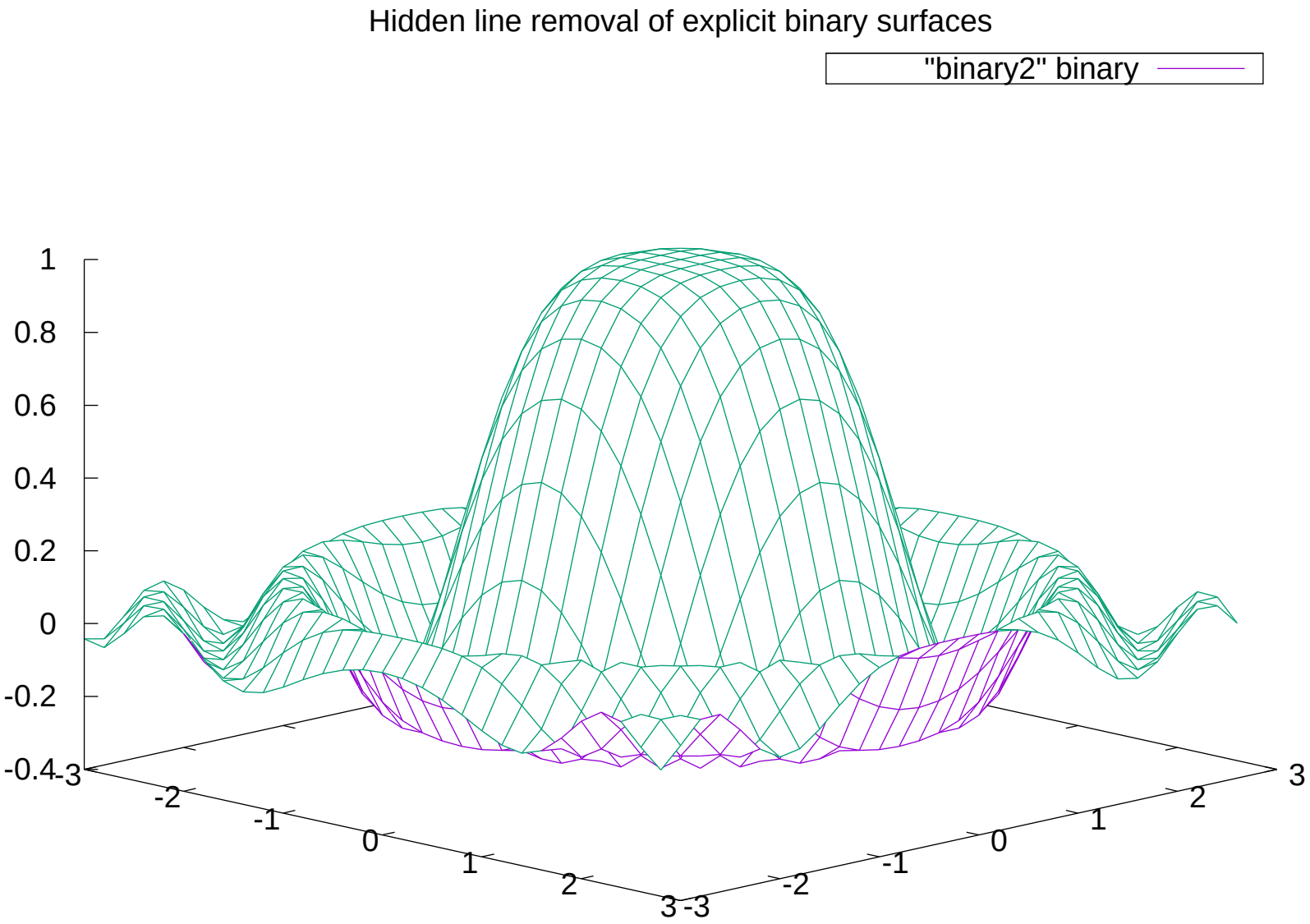

The cubic Bezier/Bspline basis functions in use
cub_bezier_x(t), cub_bezier_y(t)
cub_bsplin_x(t), cub_bsplin_y(t)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1


The cubic Hermite basis functions in use
cub_hermit_x1(t), cub_hermit_y1(t)
cub_hermit_x2(t), cub_hermit_y2(t)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1

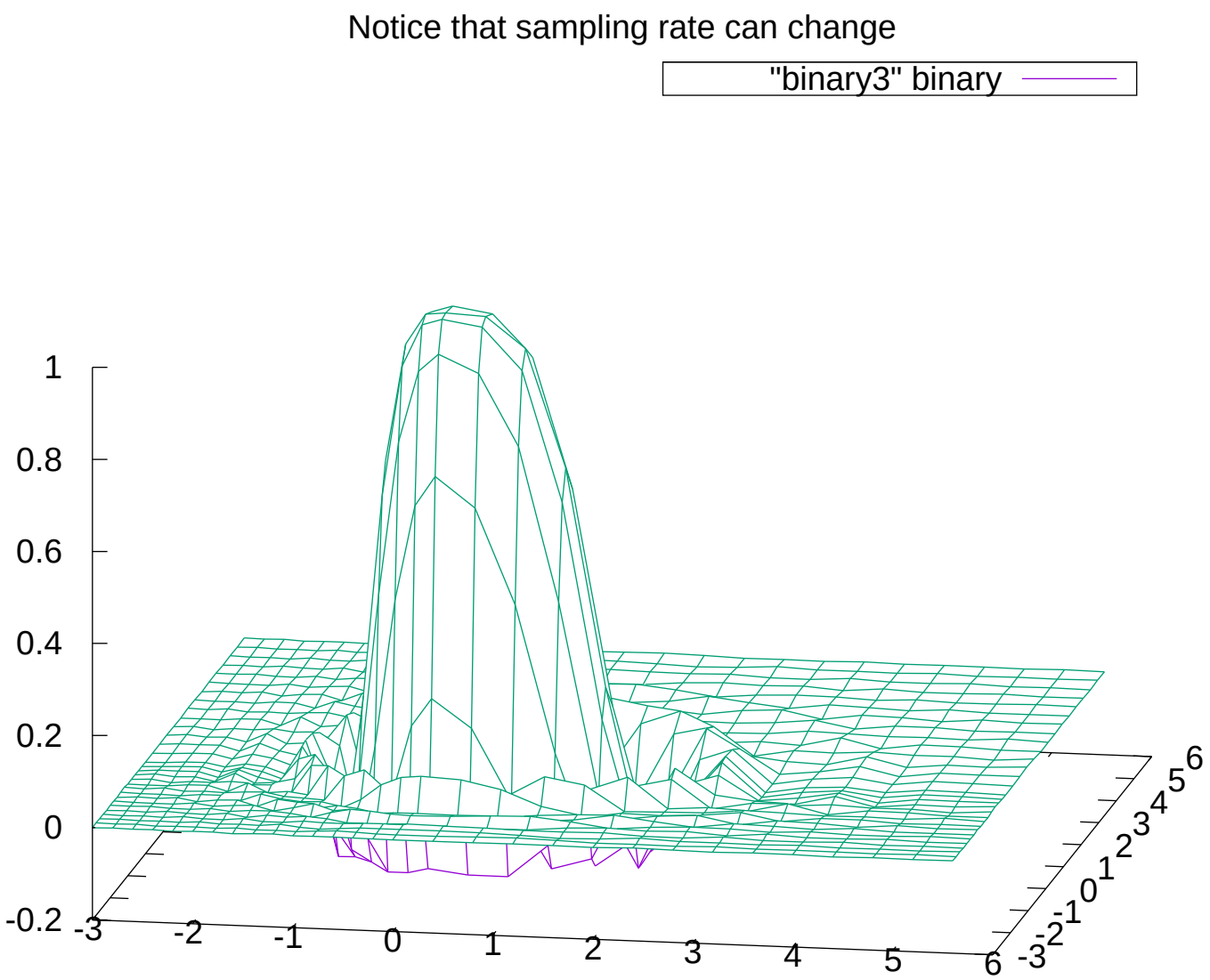

Monotonic cubic splines
raw data
smooth cspline
smooth mcs
30
25
20
15
10
5
240
260
280
300
320
340
360
380

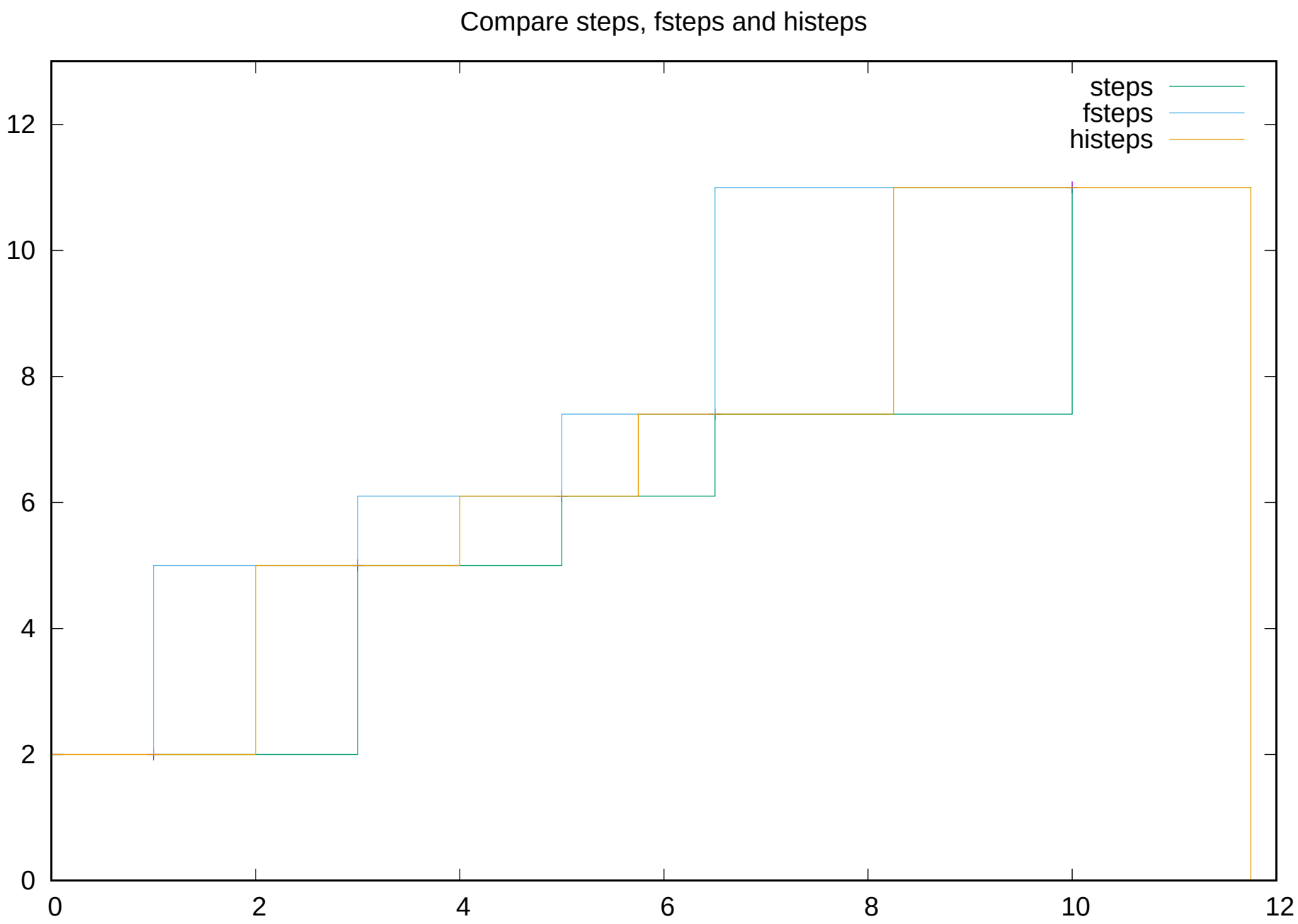

Monotonic cubic splines (log-scale data)
raw data
smooth cspline
smooth mcs
20
10
5
4
3
2
240
260
280
300
320
340
360
380

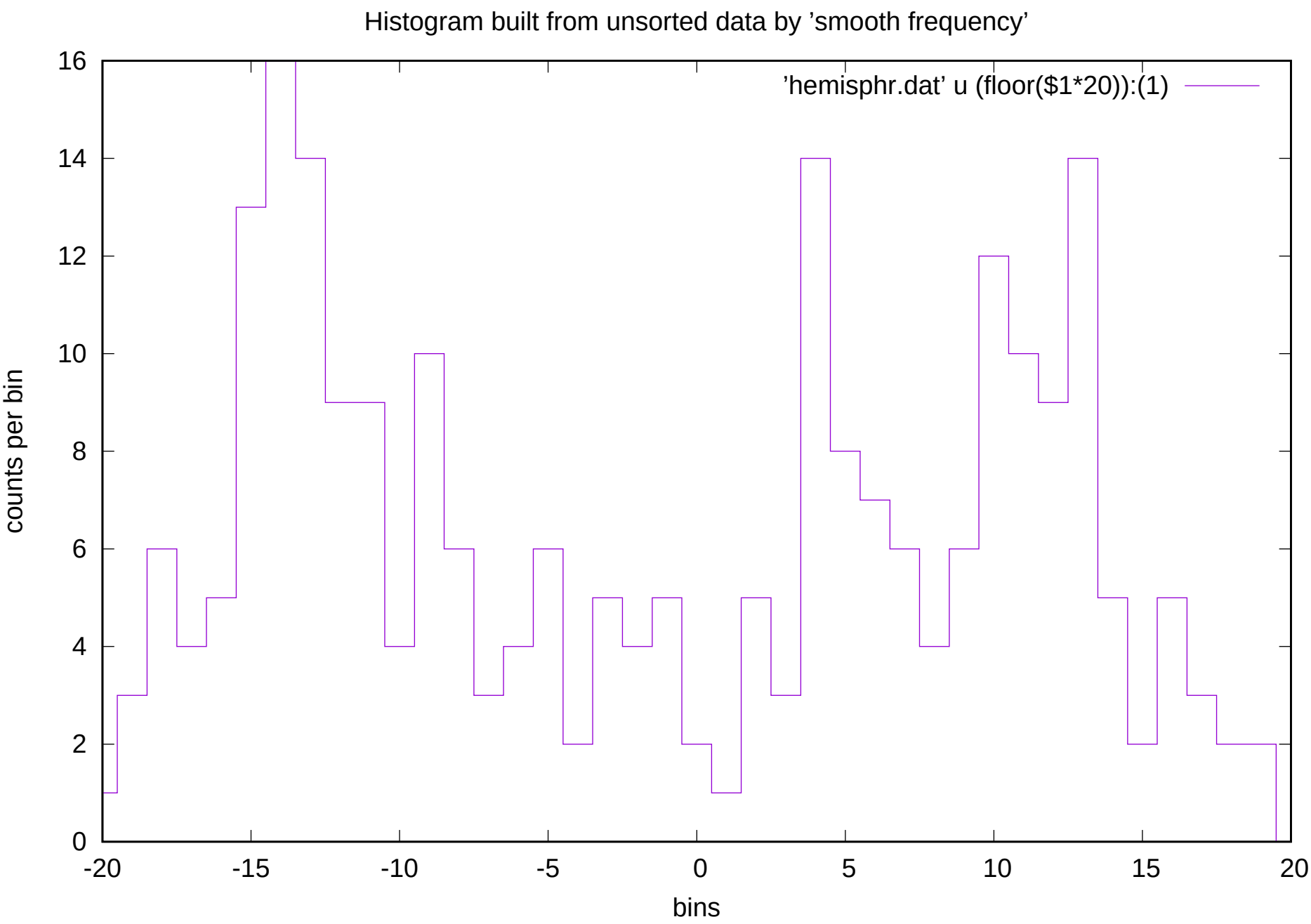

Hidden line removal of explicit binary surfaces
"binary1" binary
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
-3
-2
-1
0
1
2
3
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

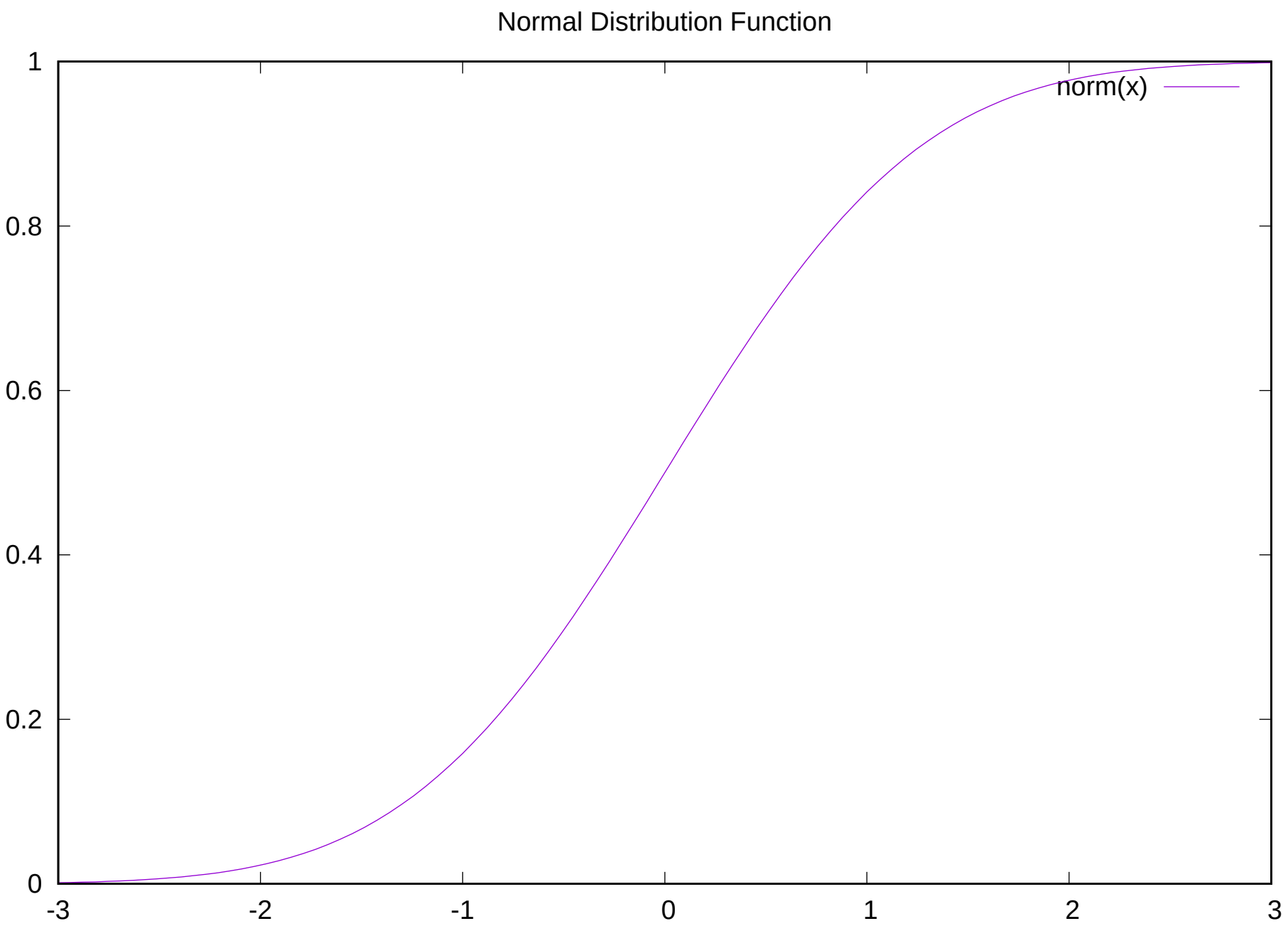

Hidden line removal of explicit binary surfaces
"binary2" binary
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-3
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3

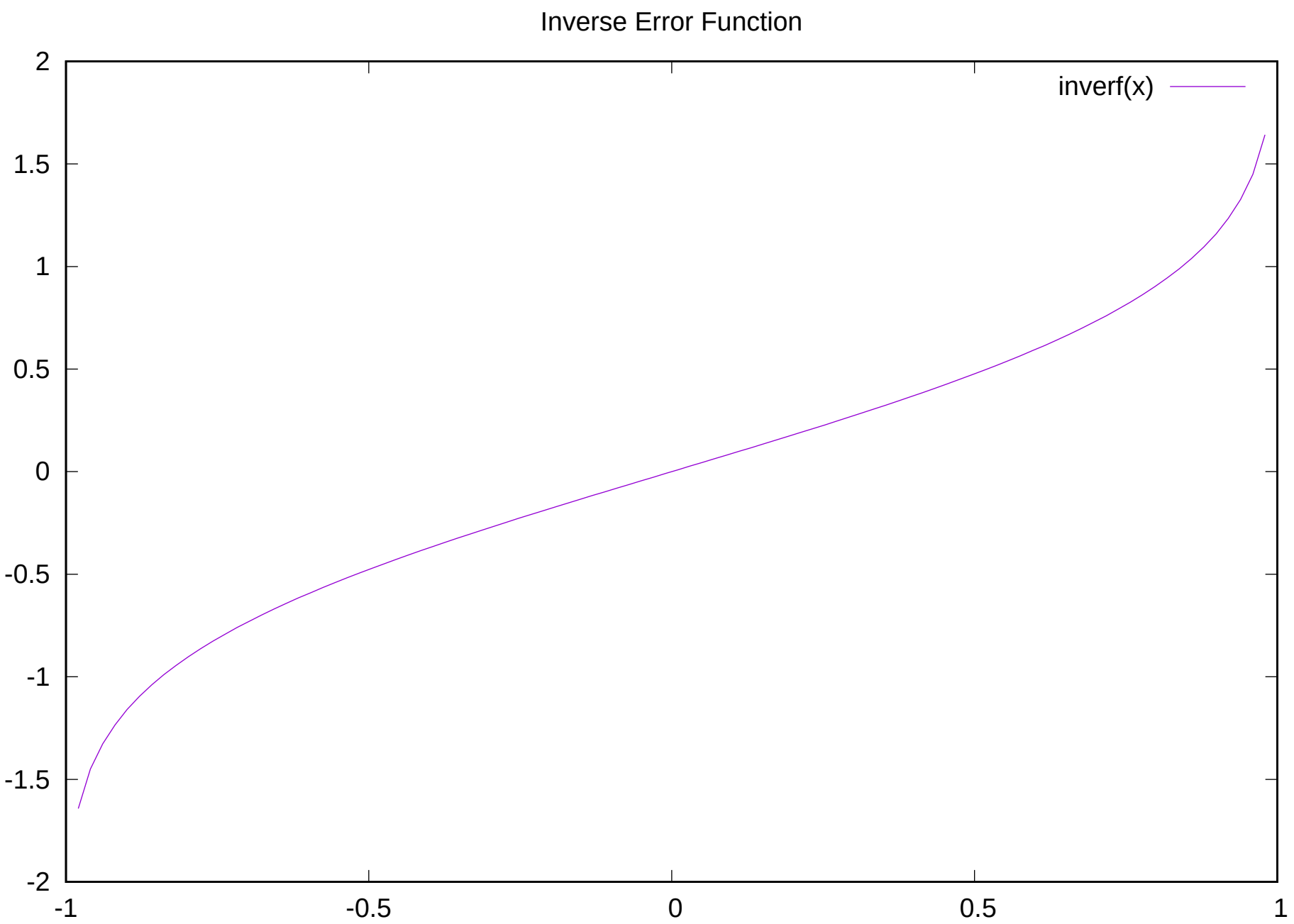
Notice that sampling rate can change

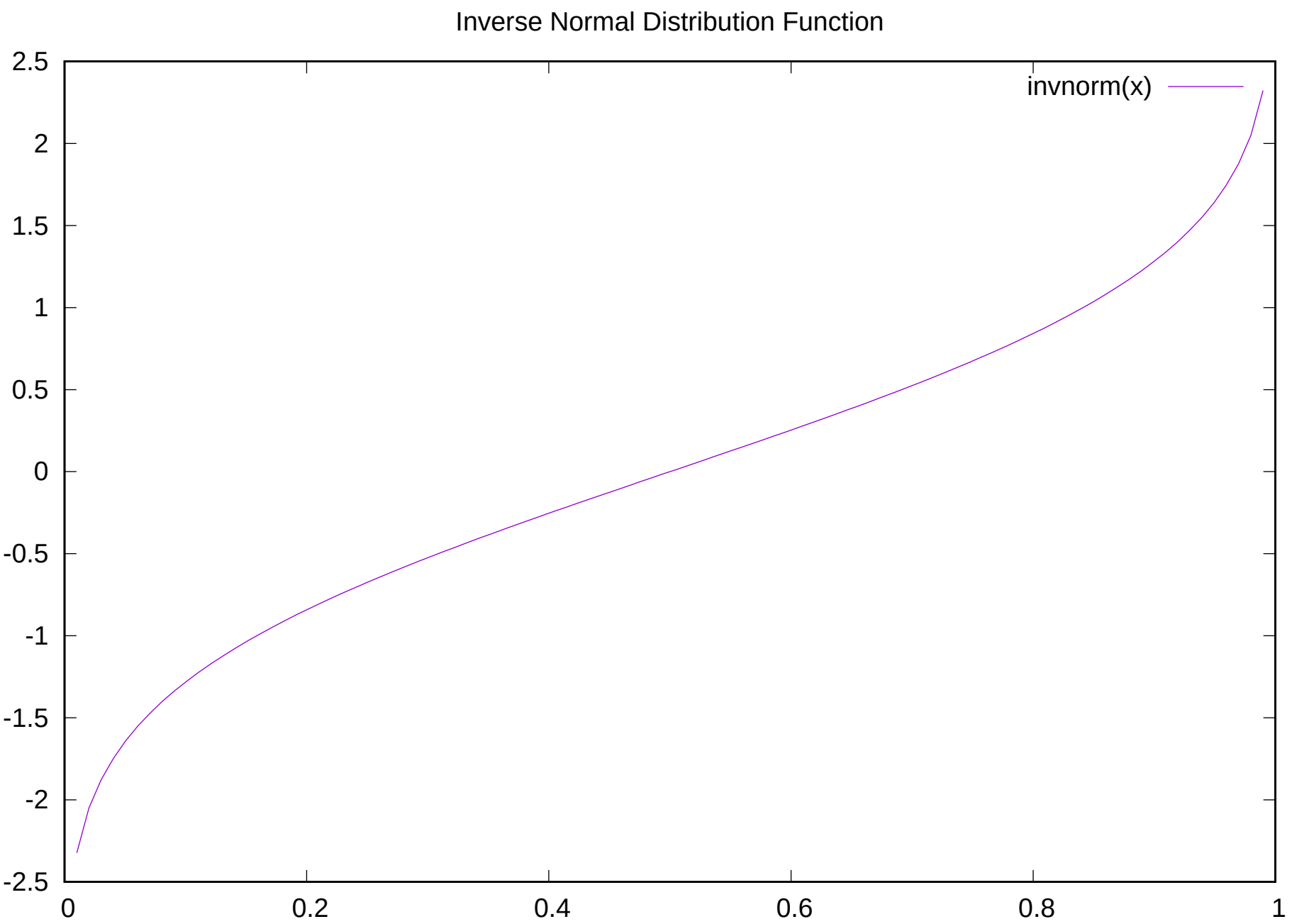


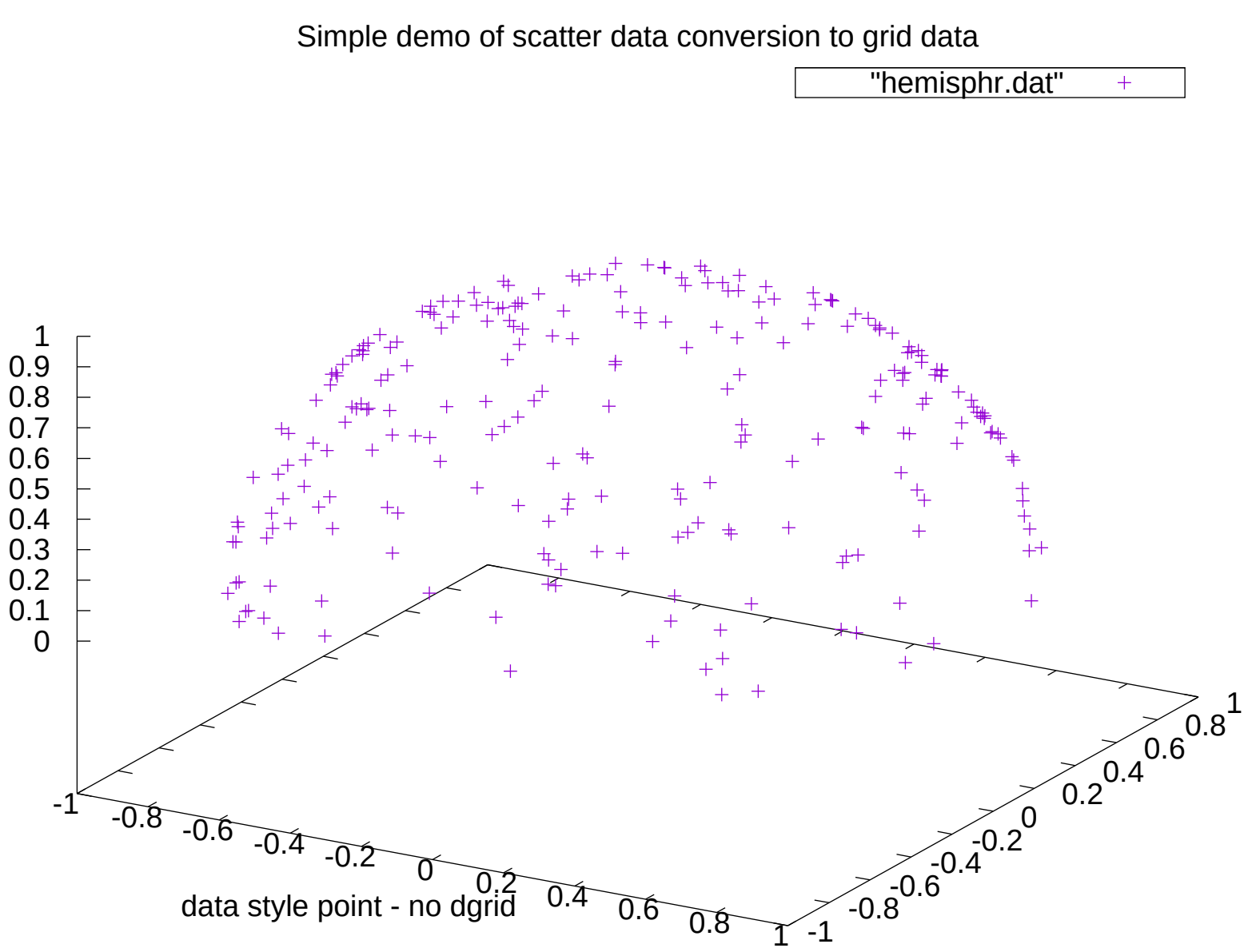

Compare steps, fsteps and histeps
steps
fsteps
histeps
0
2
4
6
8
10
12
0
2
4
6
8
10
12

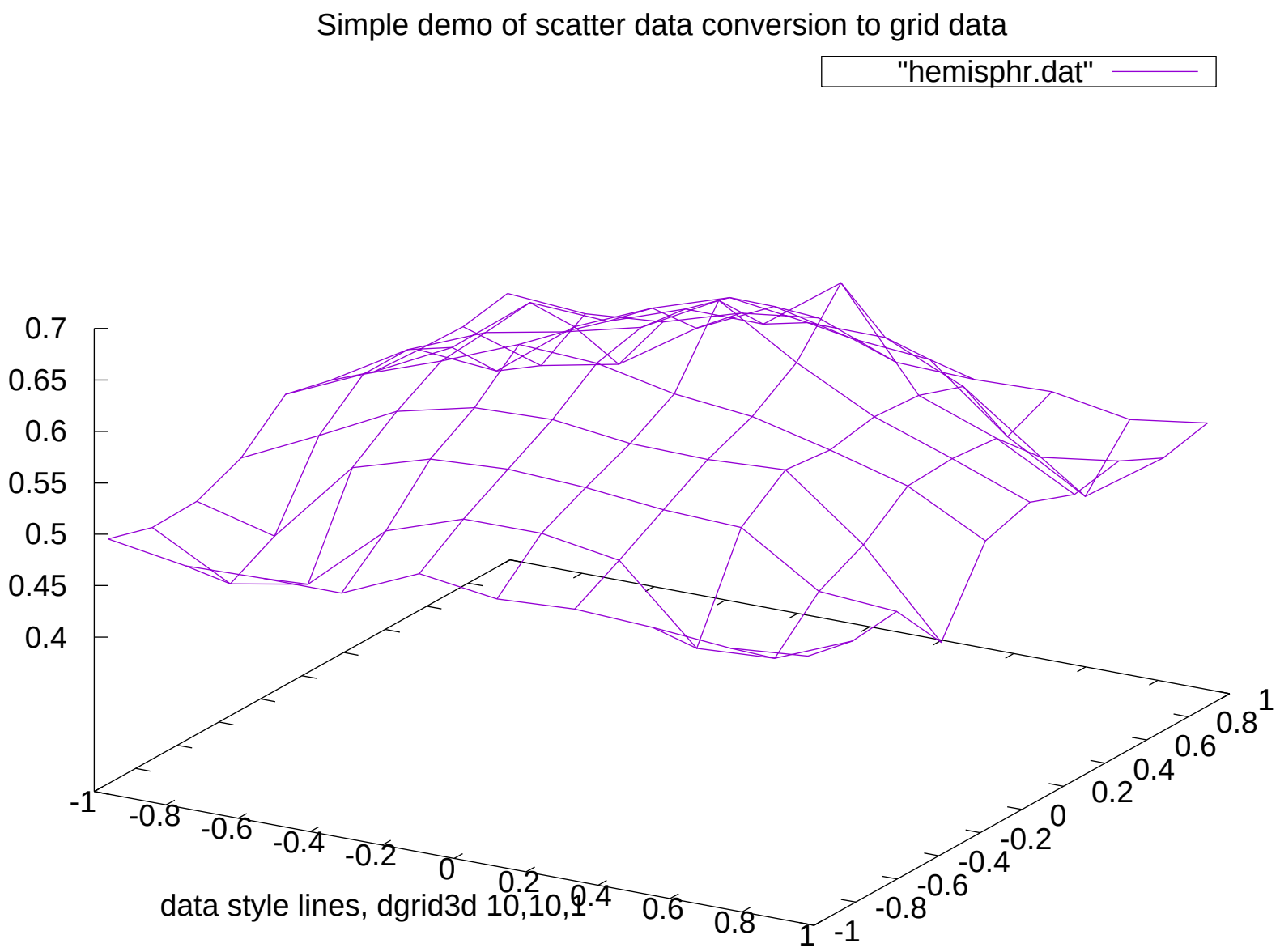

Histogram built from unsorted data by 'smooth frequency'
'hemisphr.dat' u (floor($1*20)):(1)
counts per bin
bins
0
2
4
6
8
10
12
14
16
-20
-15
-10
-5
0
5
10
15
20

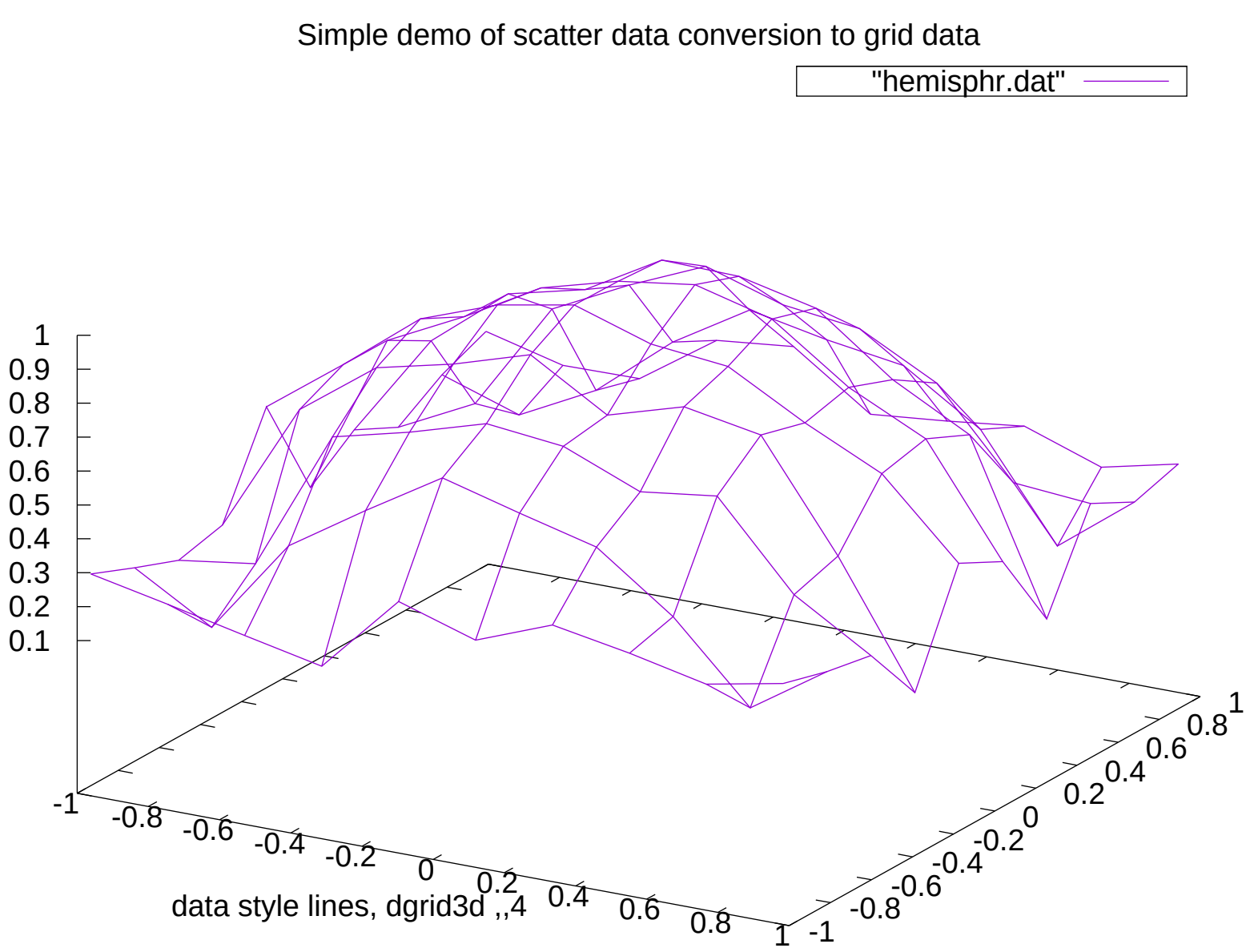

Normal Distribution Function
norm(x)
1
0.8
0.6
0.4
0.2
0
-3
-2
-1
0
1
2
3

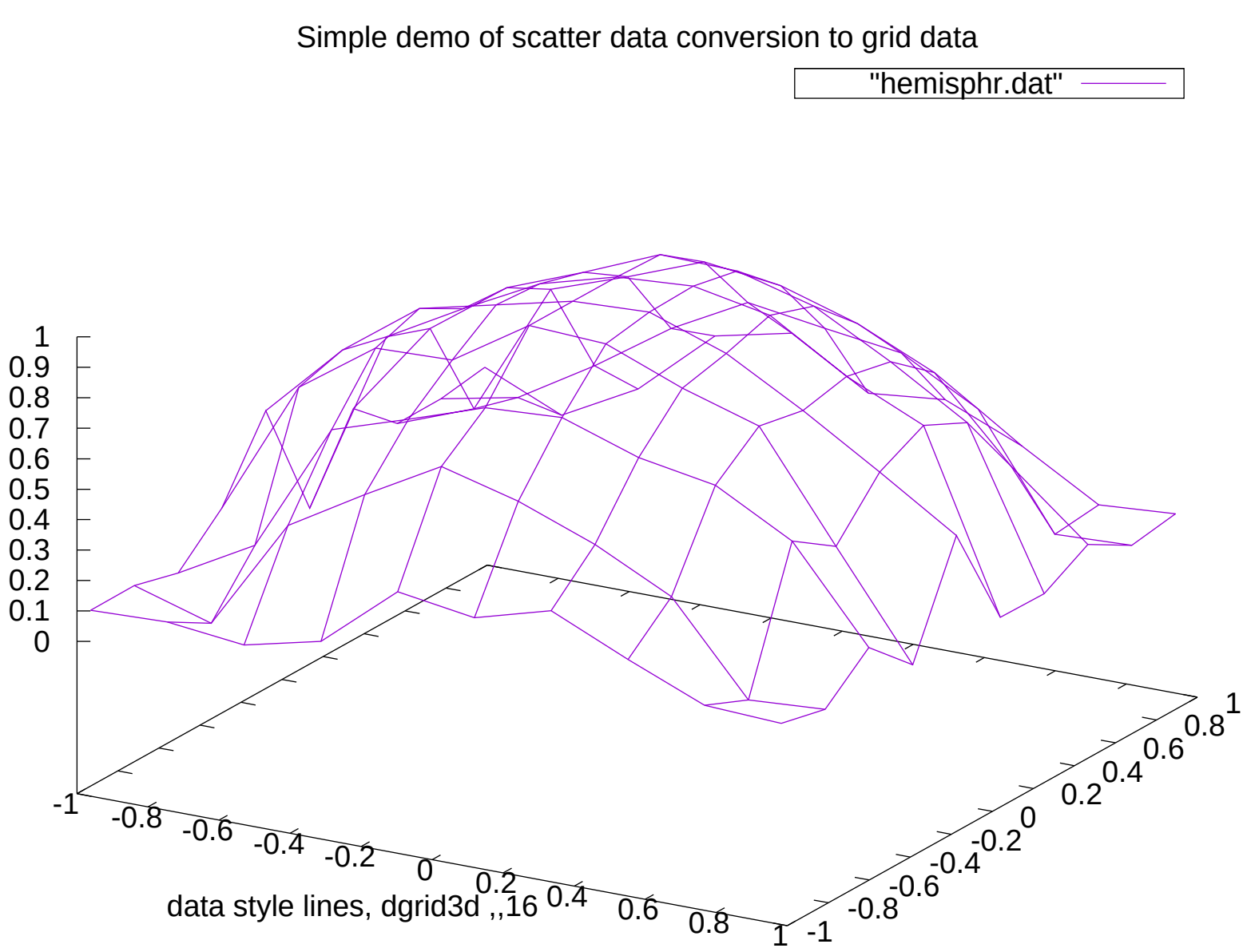

Inverse Error Function
inverf(x)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1
-0.5
0
0.5
1

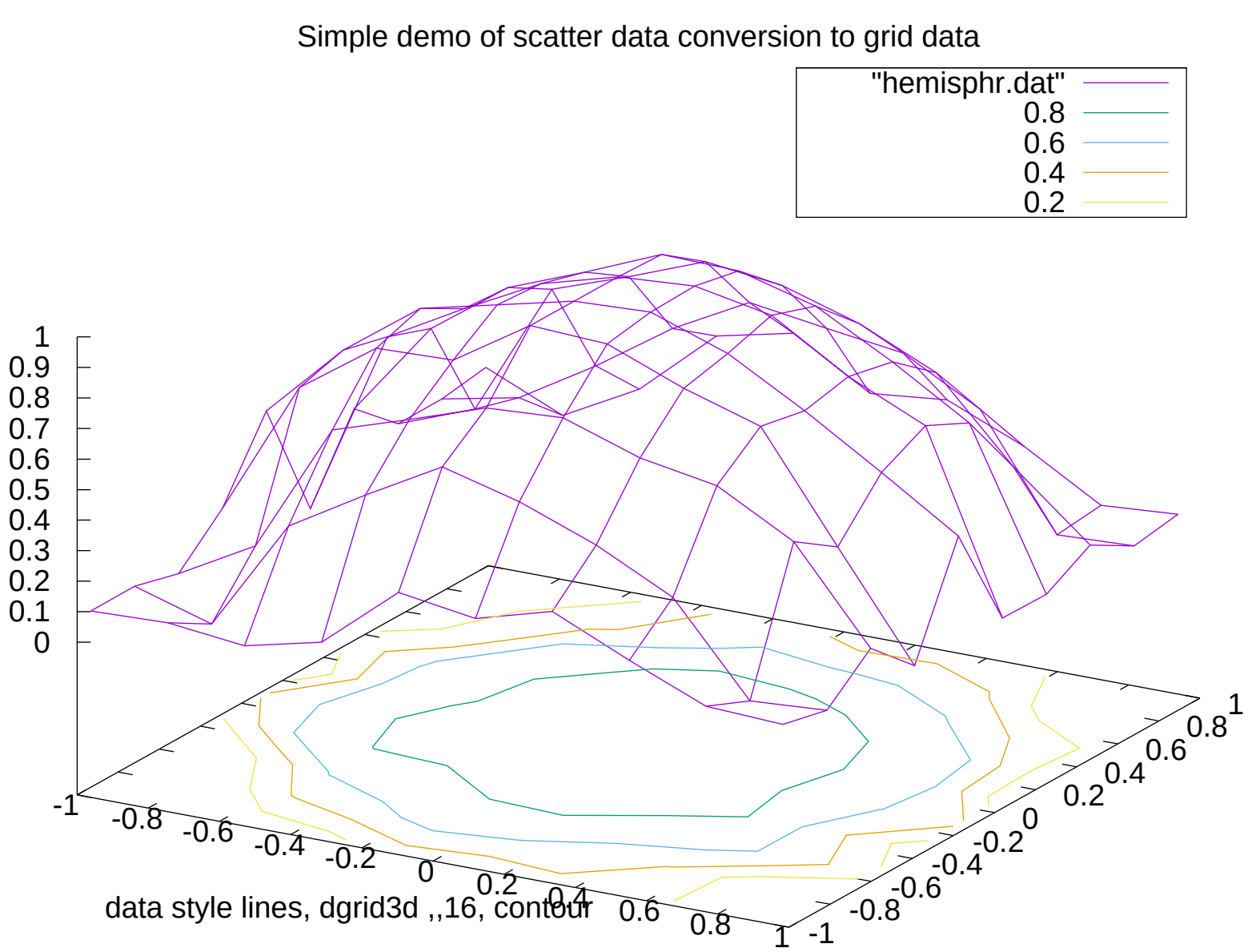

Inverse Normal Distribution Function
invnorm(x)
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
0.2
0.4
0.6
0.8
1

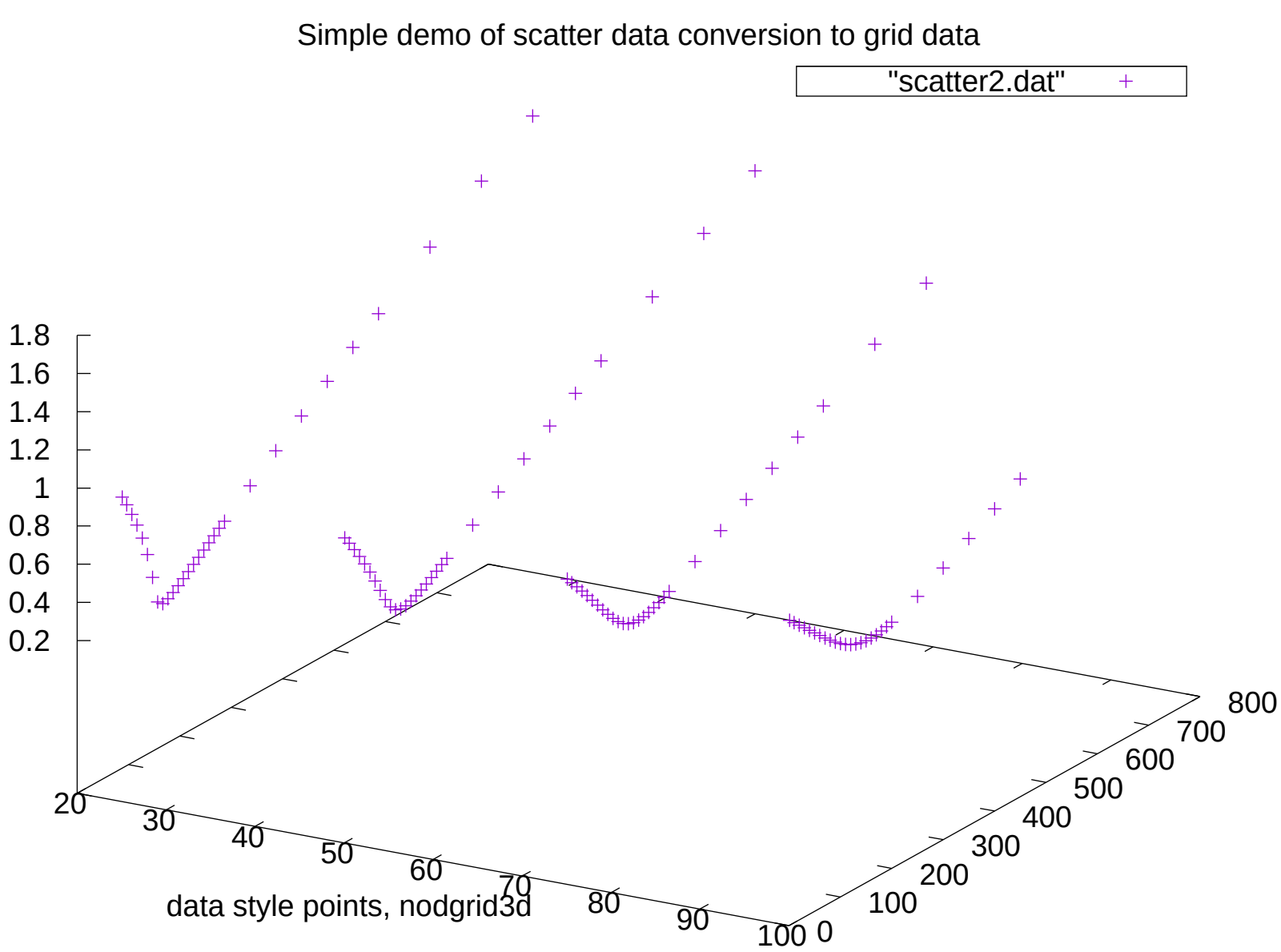

Simple demo of scatter data conversion to grid data
"hemisphr.dat"
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
data style point - no dgrid

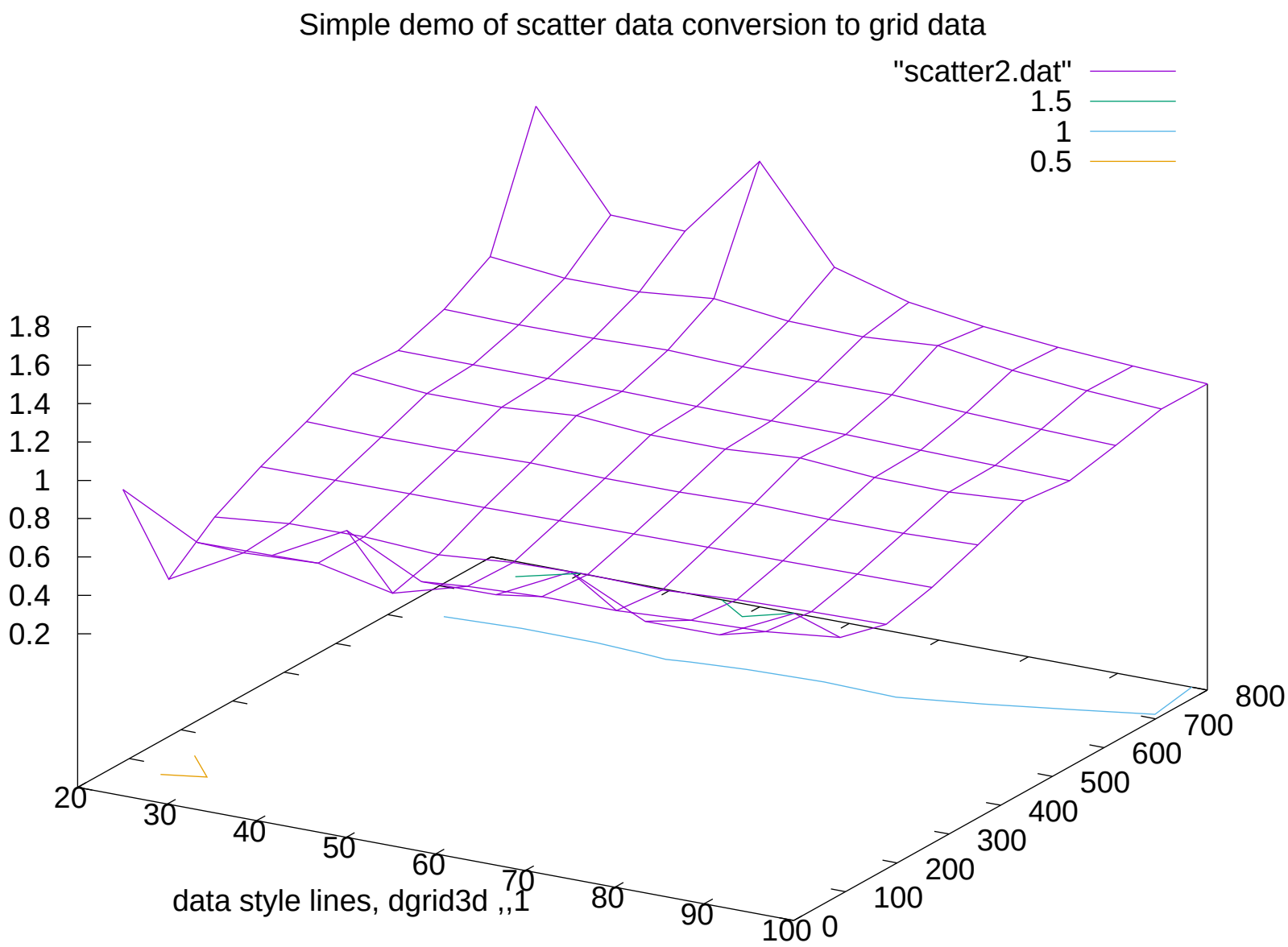

Simple demo of scatter data conversion to grid data
"hemisphr.dat"
0.7
0.65
0.6
0.55
0.5
0.45
0.4
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
data style lines, dgrid3d 10,10,1

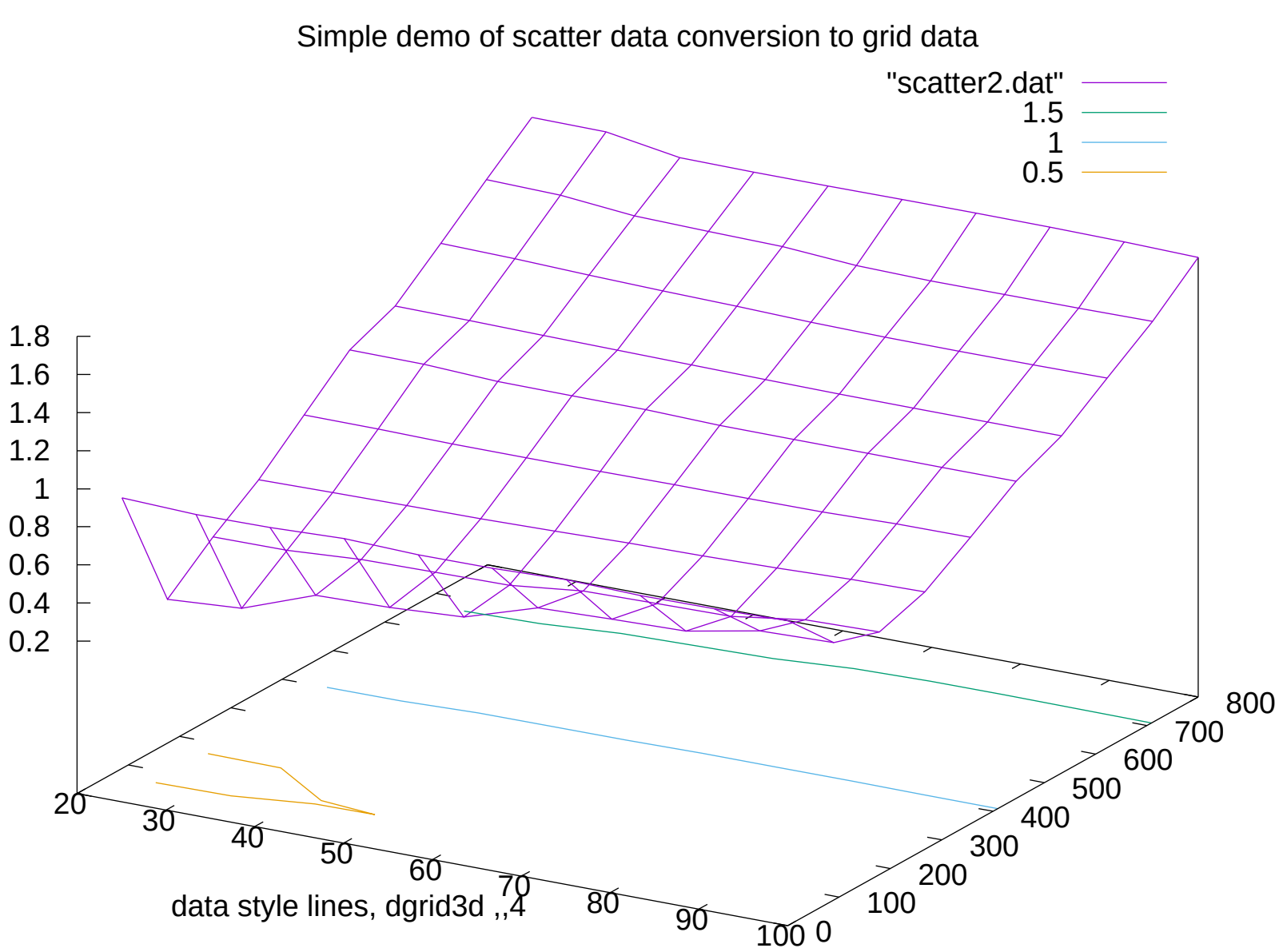

Simple demo of scatter data conversion to grid data
"hemisphr.dat"
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
data style lines, dgrid3d ,,4

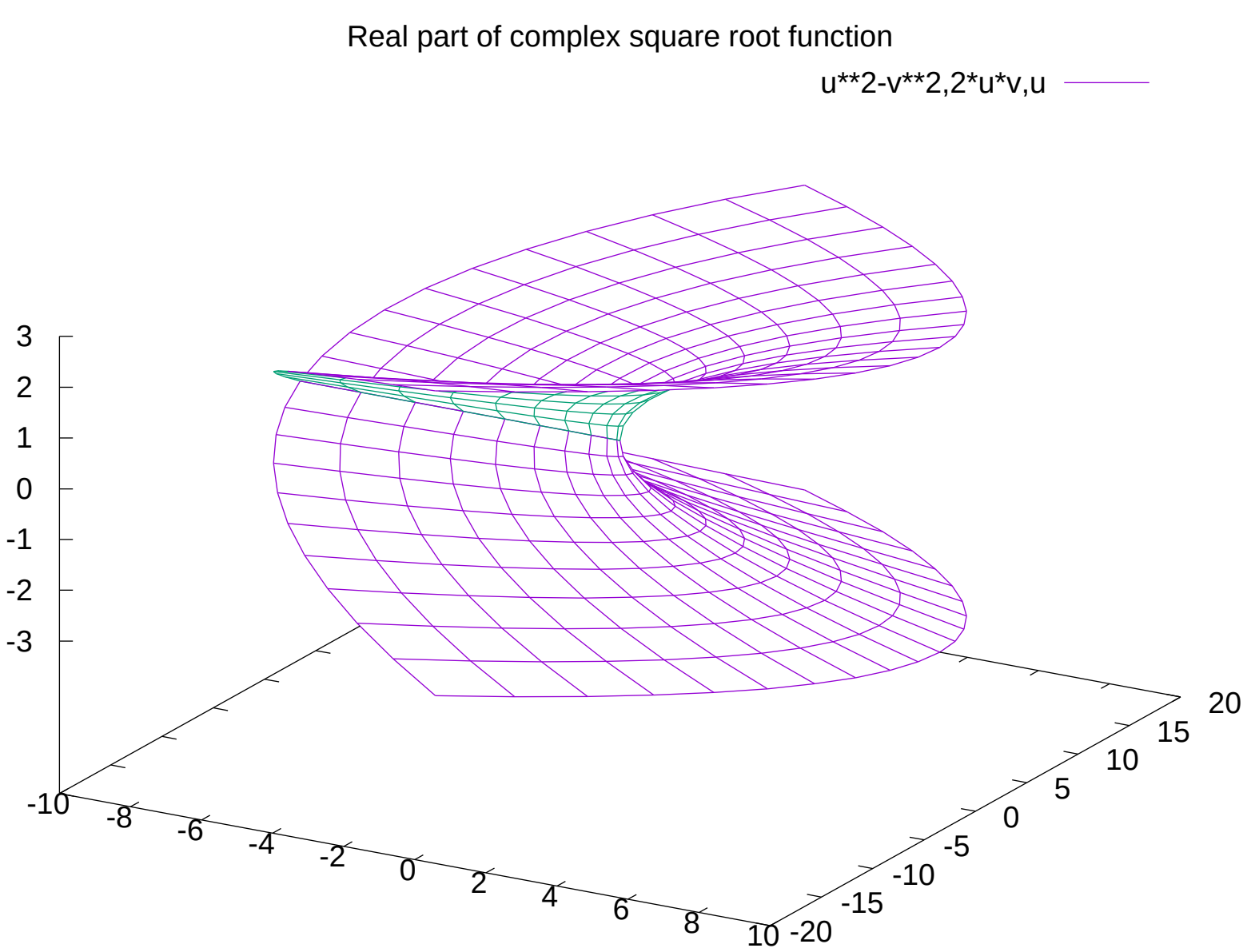

Simple demo of scatter data conversion to grid data
"hemisphr.dat"
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
data style lines, dgrid3d ,,16

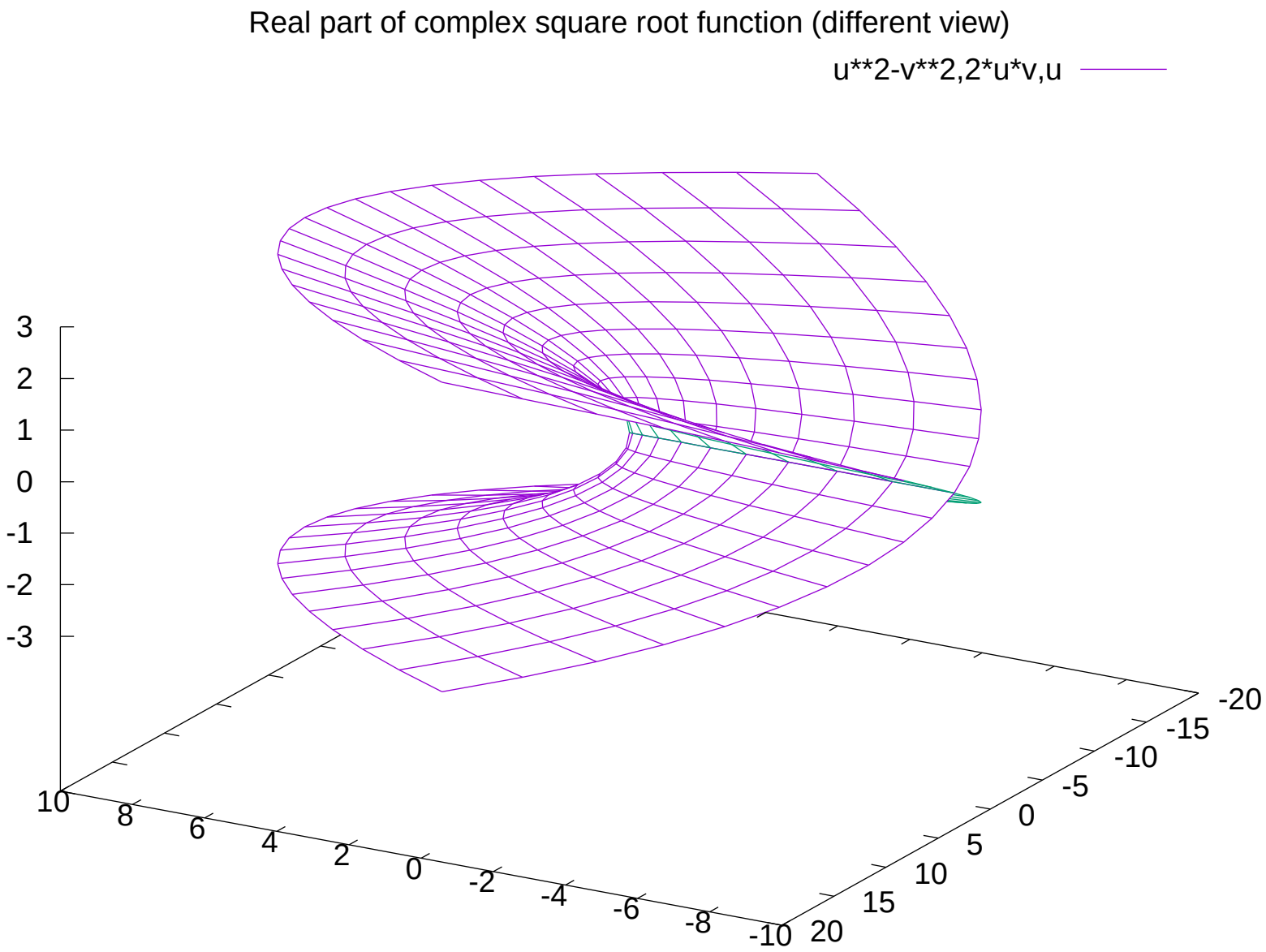

Simple demo of scatter data conversion to grid data
"hemisphr.dat"
0.8
0.6
0.4
0.2
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
data style lines, dgrid3d ,,16, contour

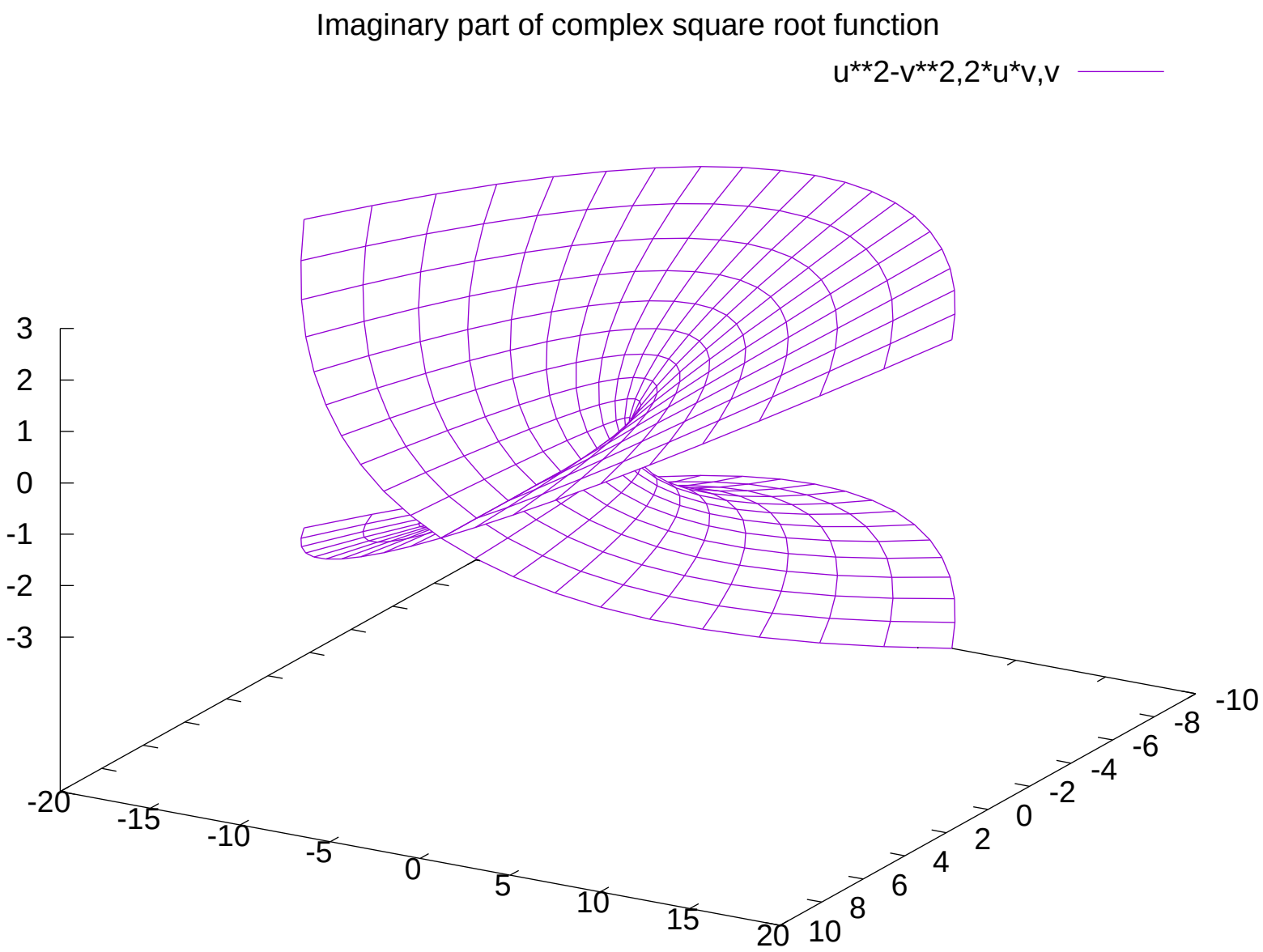

Simple demo of scatter data conversion to grid data
"scatter2.dat"
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500
600
700
800
data style points, nodgrid3d


Simple demo of scatter data conversion to grid data
"scatter2.dat"
1.5
1
0.5
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500
600
700
800
data style lines, dgrid3d ,,1

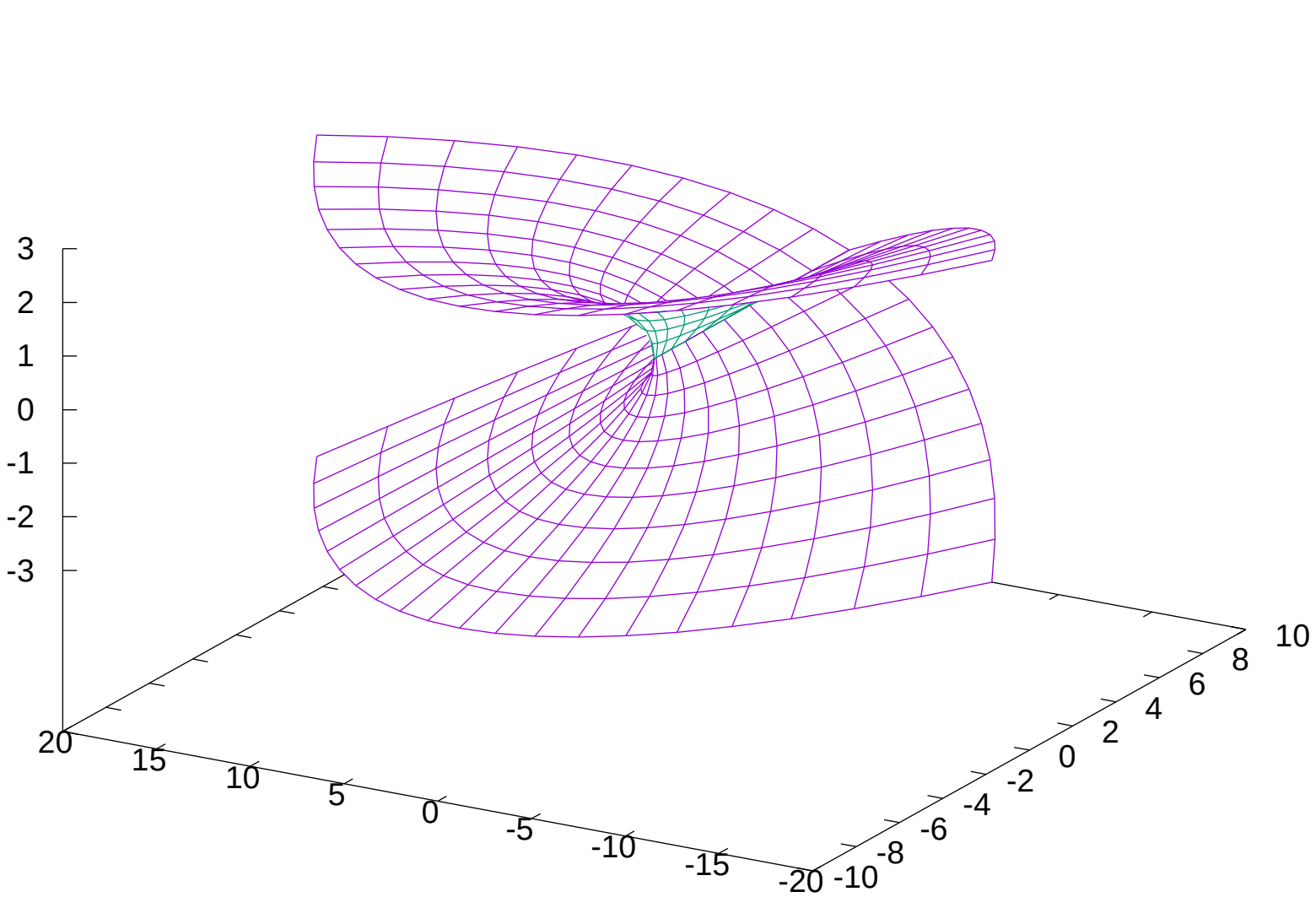

Simple demo of scatter data conversion to grid data
"scatter2.dat"
1.5
1
0.5
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500
600
700
800
data style lines, dgrid3d ,,4

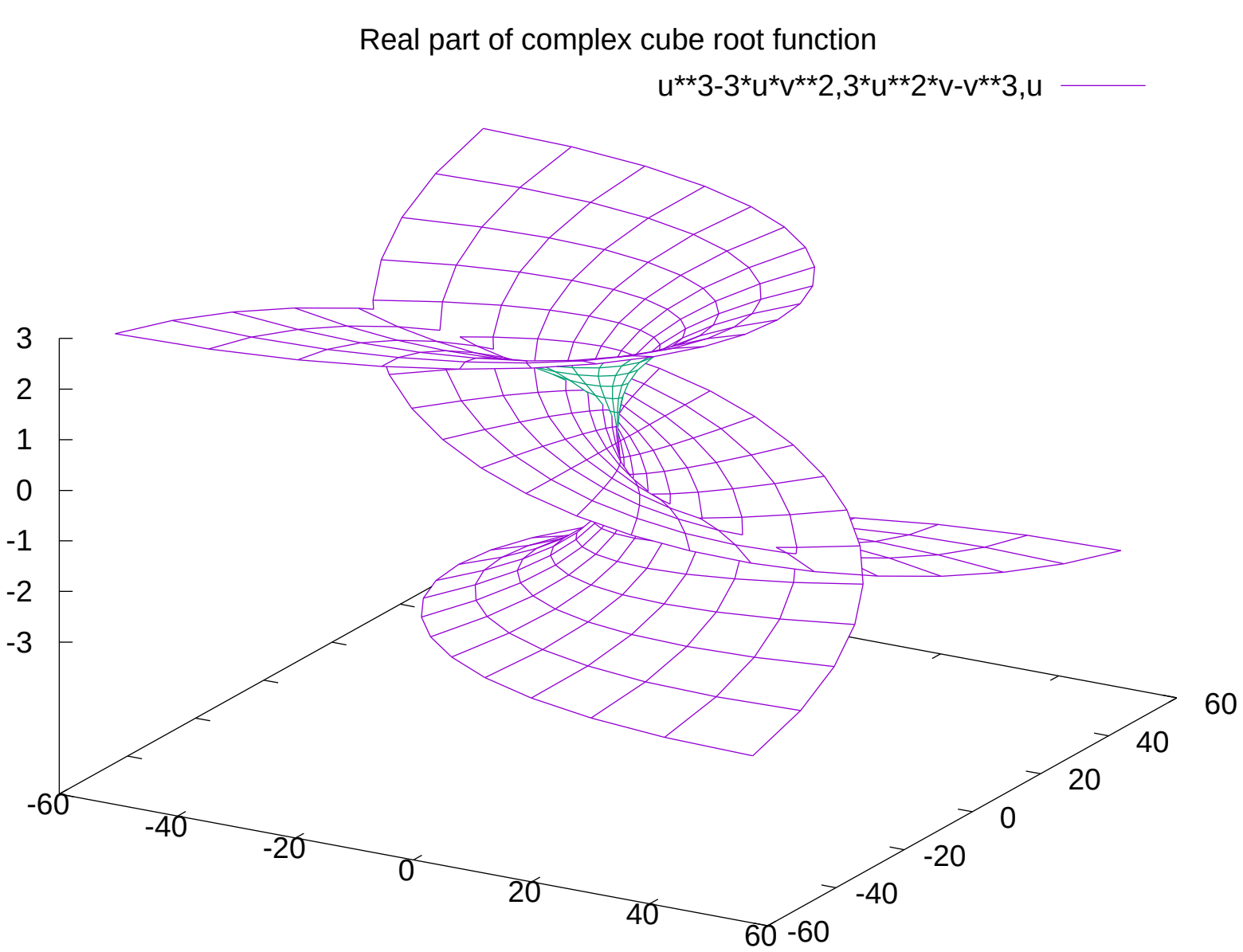

Real part of complex square root function
u**2-v**2,2*u*v,u
3
2
1
0
-1
-2
-3
-10
-8
-6
-4
-2
0
2
4
6
8
10
-20
-15
-10
-5
0
5
10
15
20

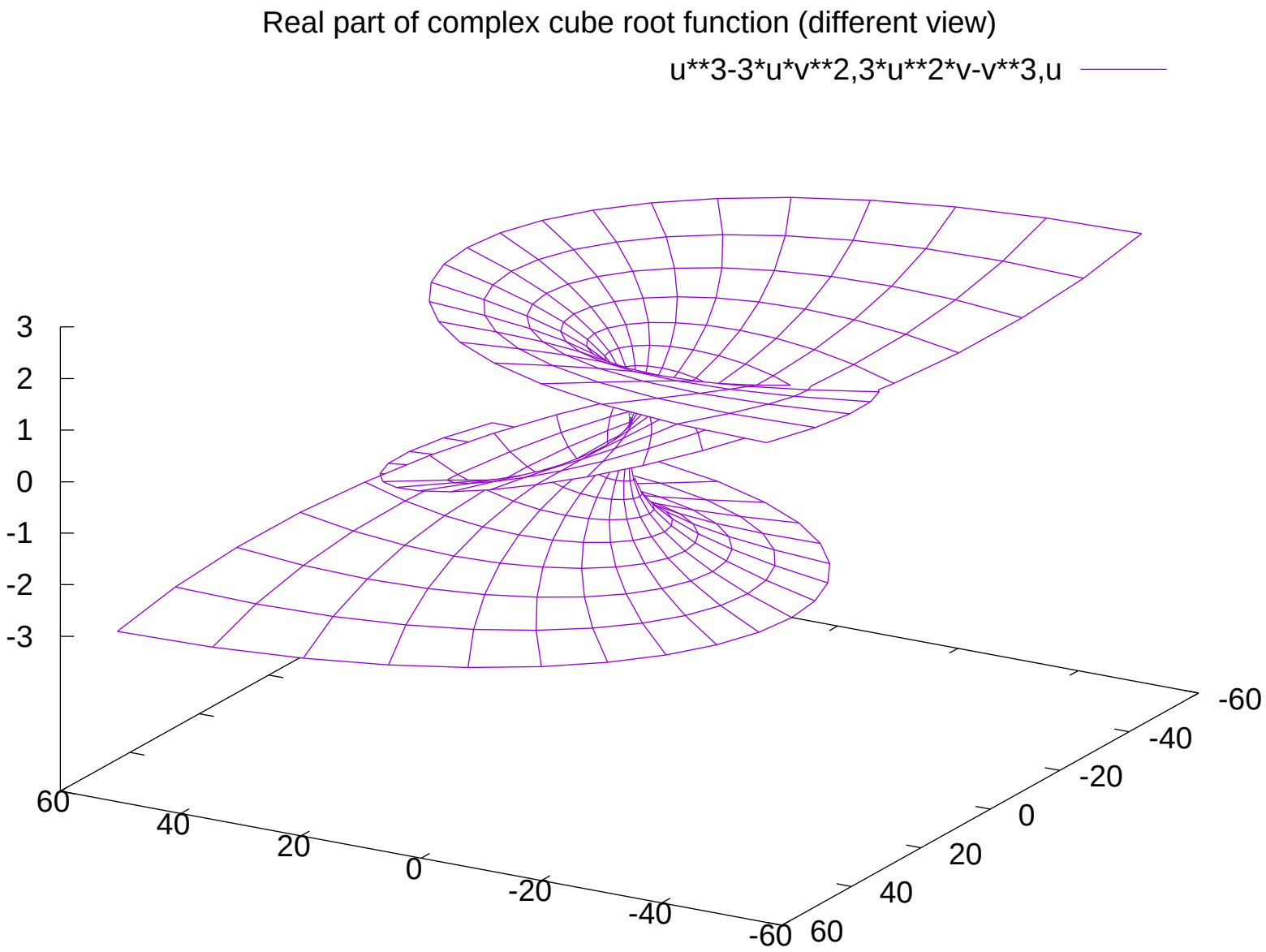

Real part of complex square root function (different view)
u**2-v**2,2*u*v,u
3
2
1
0
-1
-2
-3
10
8
6
4
2
0
-2
-4
-6
-8
-10
-20
-15
-10
-5
0
5
10
15
20

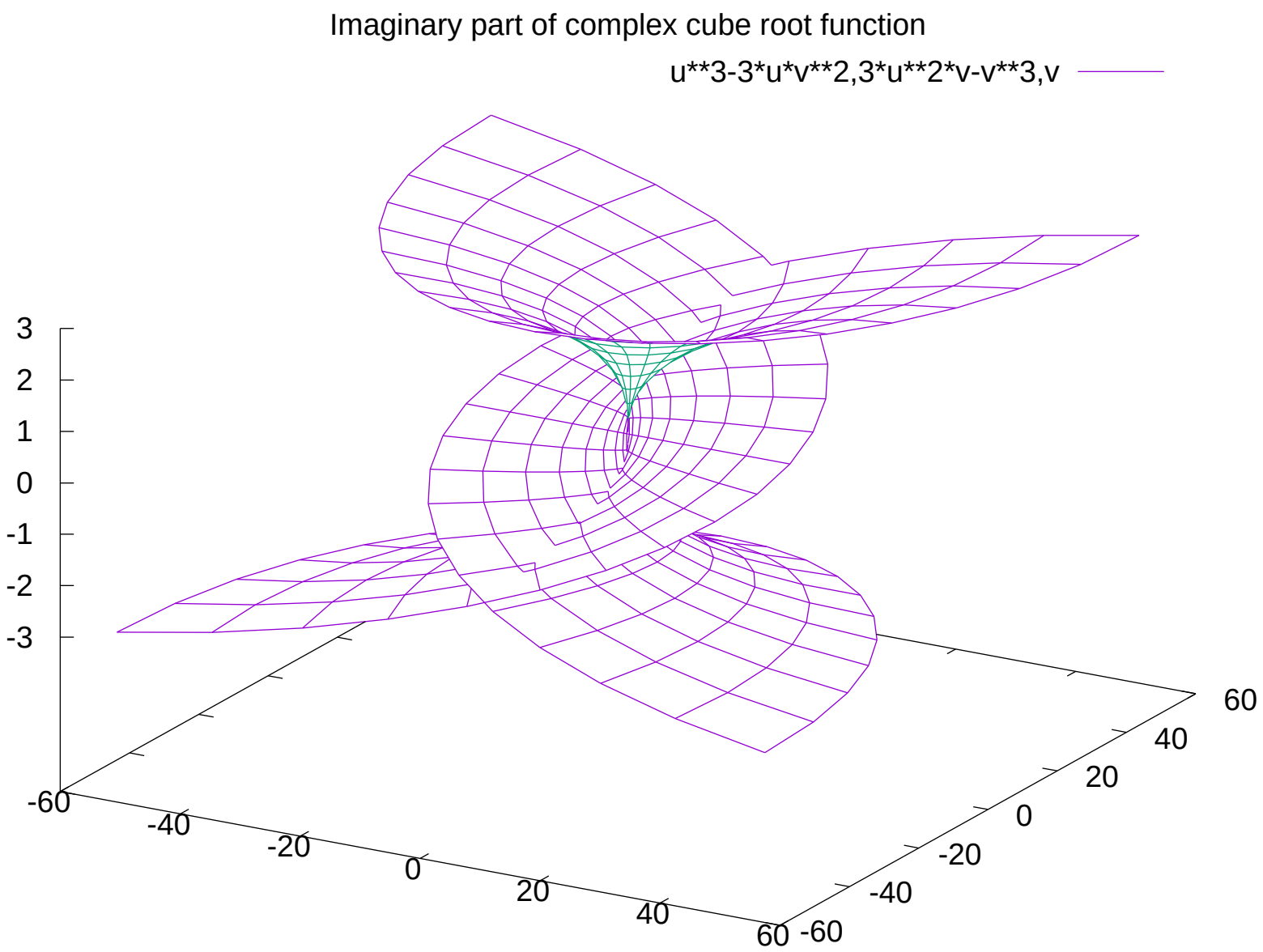

Imaginary part of complex square root function
u**2-v**2,2*u*v,v
3
2
1
0
-1
-2
-3
-20
-15
-10
-5
0
5
10
15
20
10
8
6
4
2
0
-2
-4
-6
-8
-10

Imaginary part of complex square root function (different view)



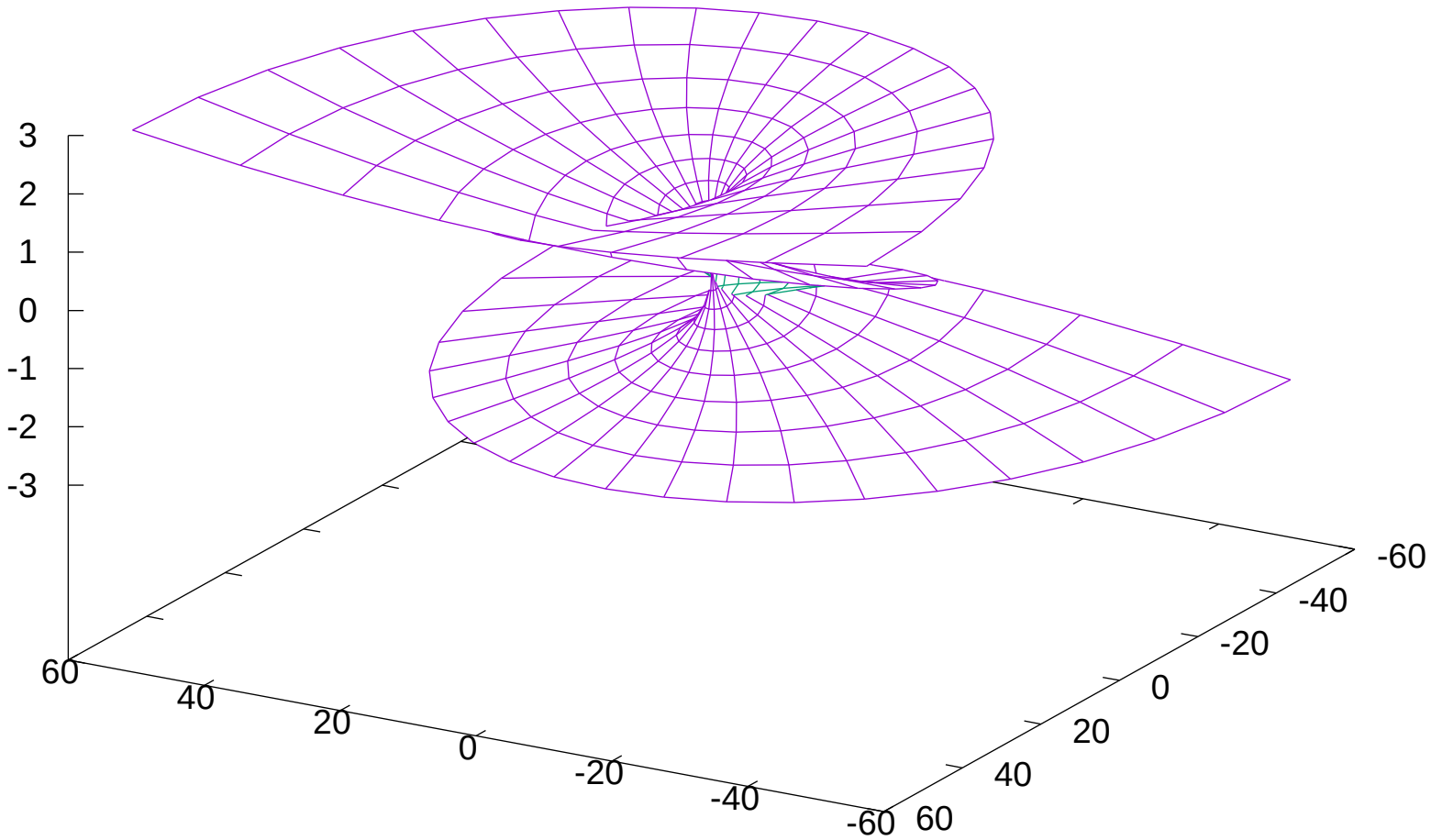

Real part of complex cube root function
u**3-3*u*v**2,3*u**2*v-v**3,u
3
2
1
0
-1
-2
-3
-60
-40
-20
0
20
40
60
-60
-40
-20
0
20
40
60


Real part of complex cube root function (different view)
u**3-3*u*v**2,3*u**2*v-v**3,u
3
2
1
0
-1
-2
-3
60
40
20
0
-20
-40
-60
-60
-40
-20
0
20
40
60


Imaginary part of complex cube root function
u**3-3*u*v**2,3*u**2*v-v**3,v
3
2
1
0
-1
-2
-3
-60
-40
-20
0
20
40
60
-60
-40
-20
0
20
40
60

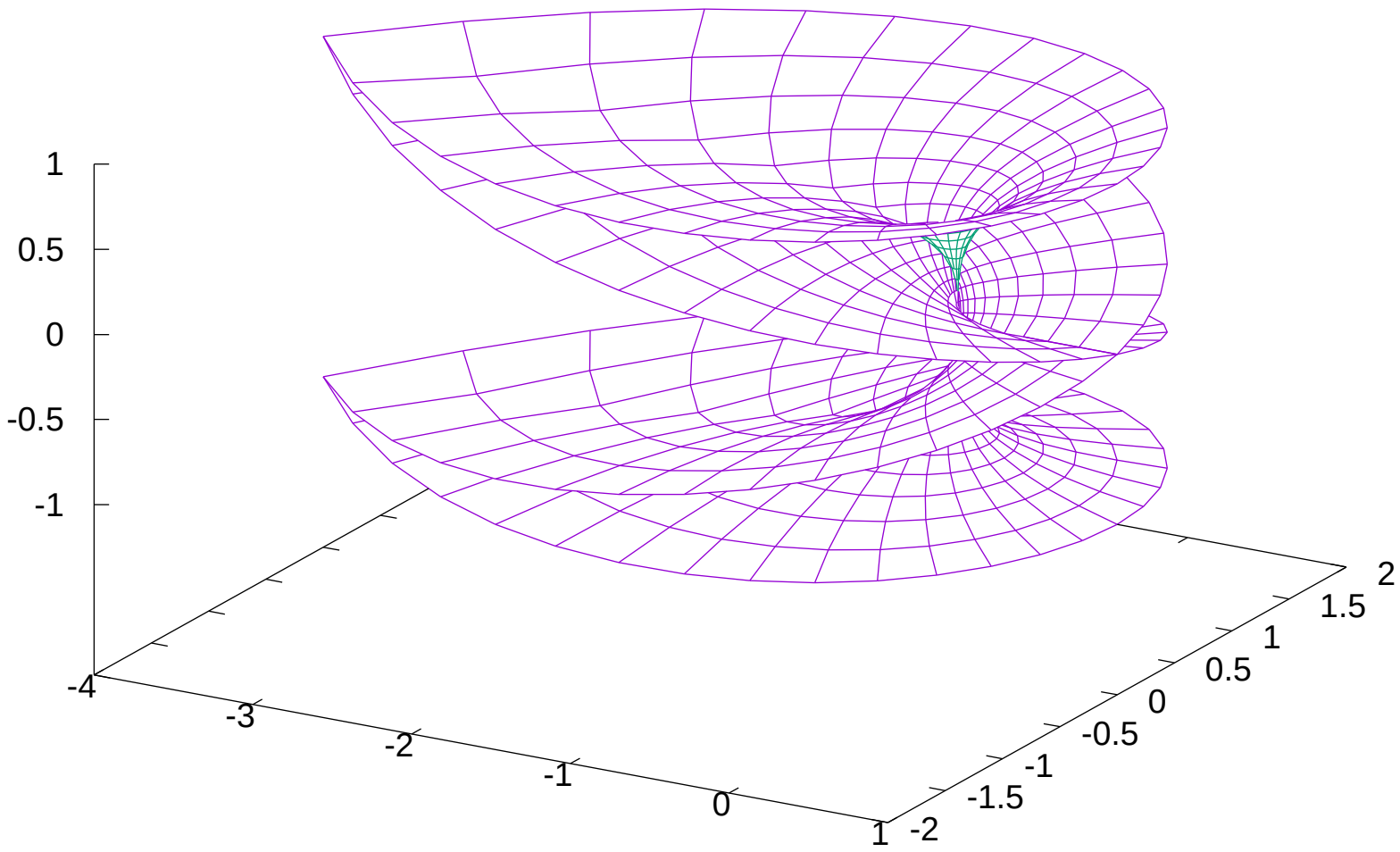
Imaginary part of complex cube root function (different view)

u**3-3*u*v**2,3*u**2*v-v**3,v



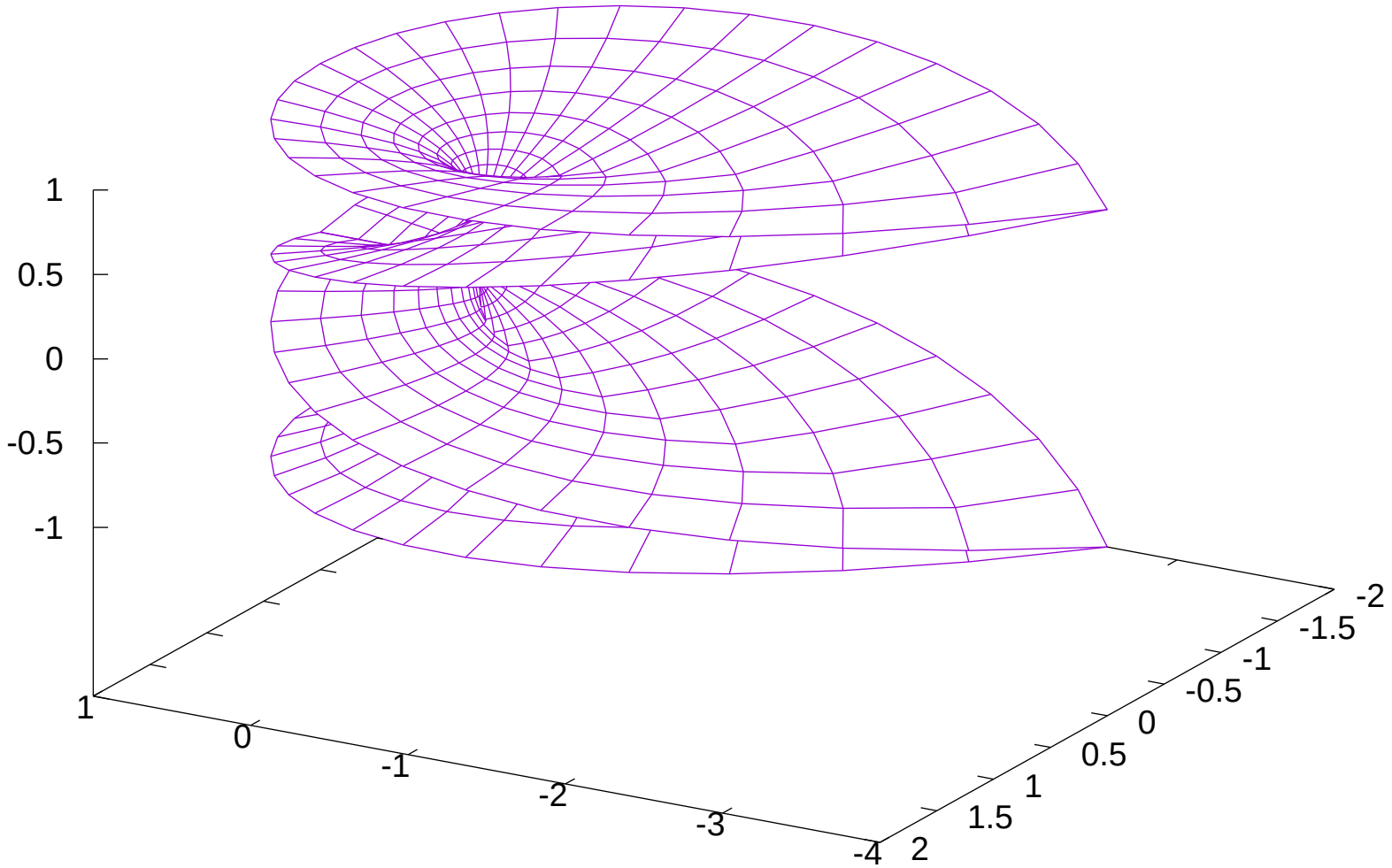
Real part of complex 4th root function

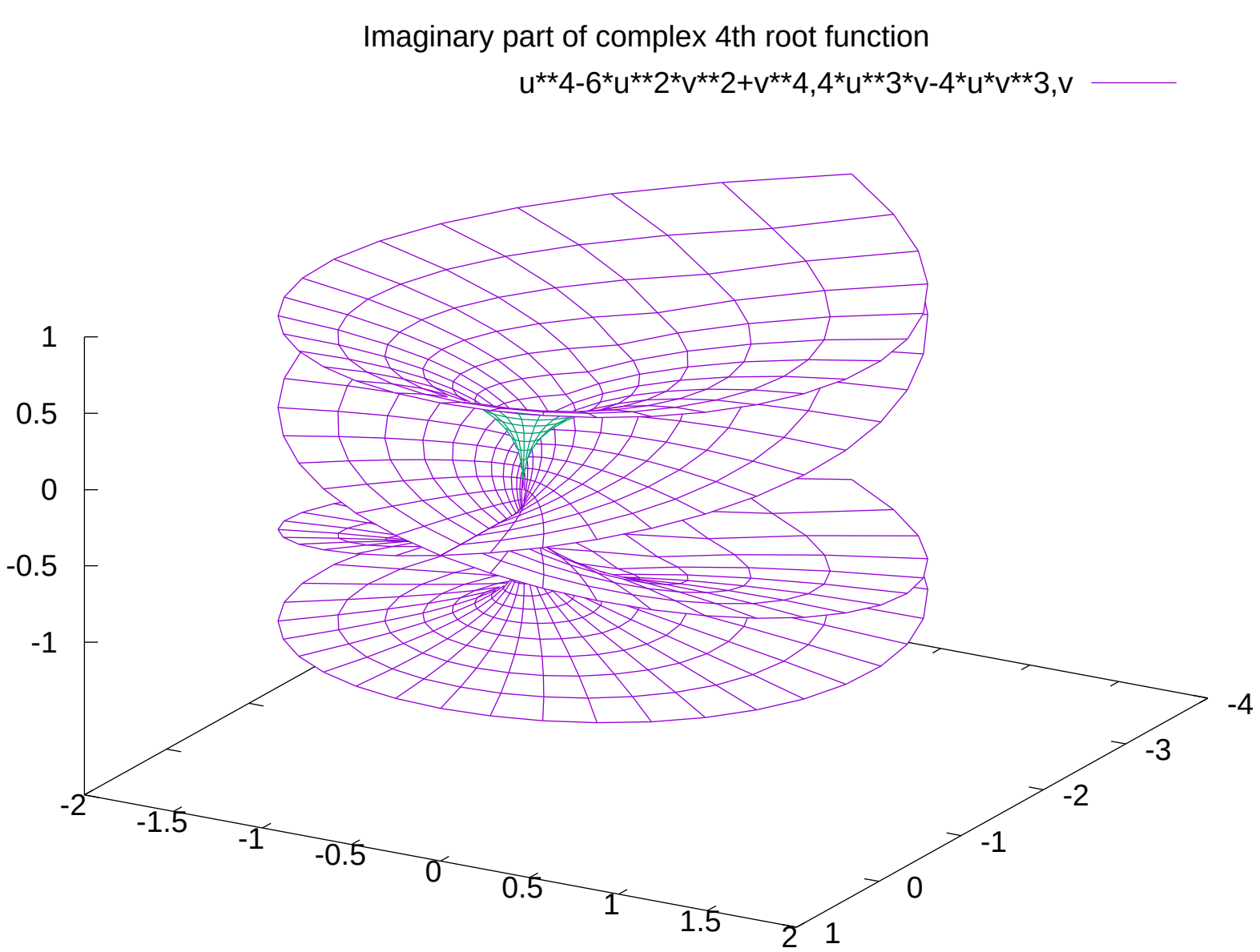
u**4-6*u**2*v**2+v**4,4*u**3*v-4*u*v**3,u



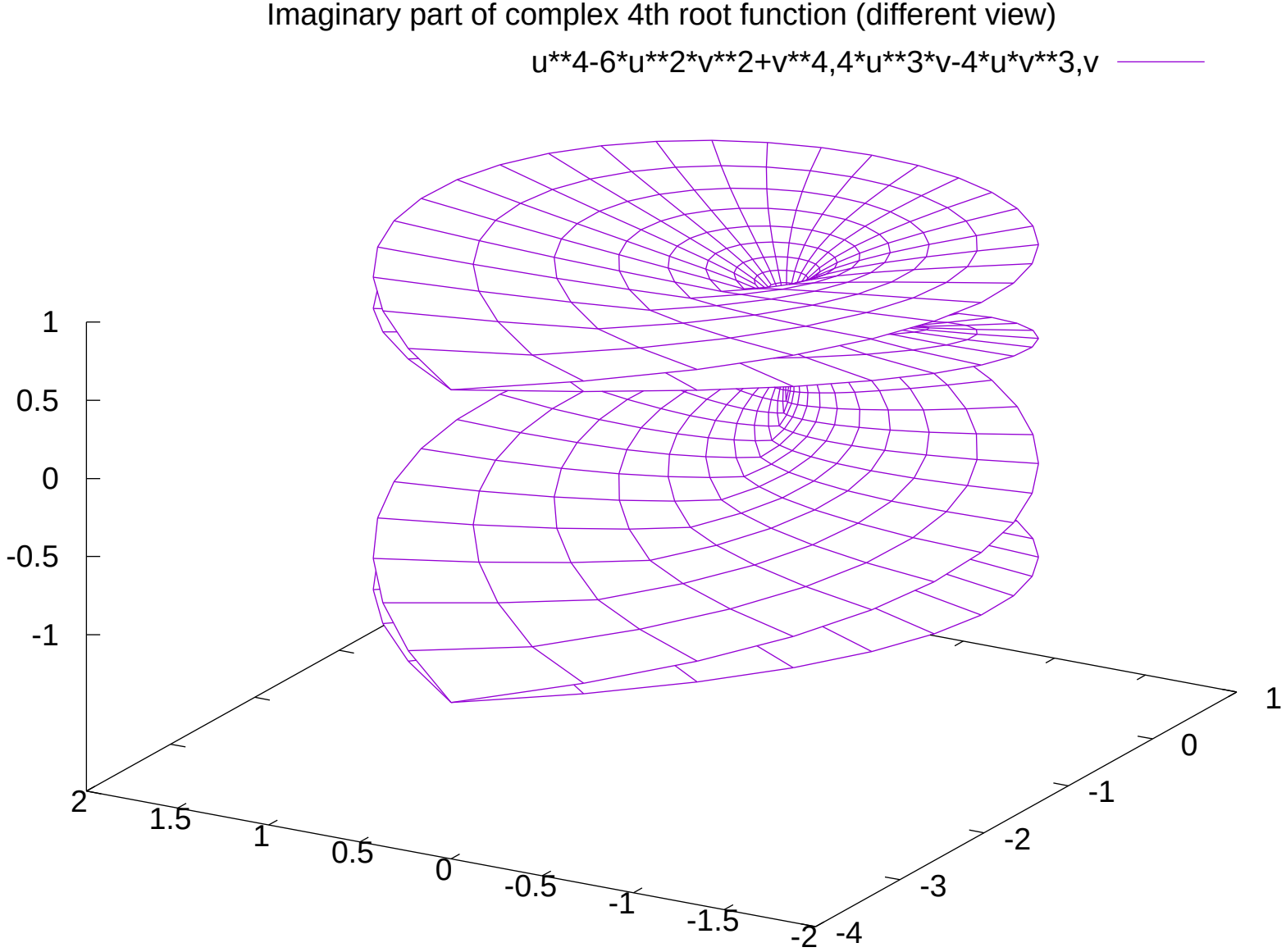
Real part of complex 4th root function (different view)

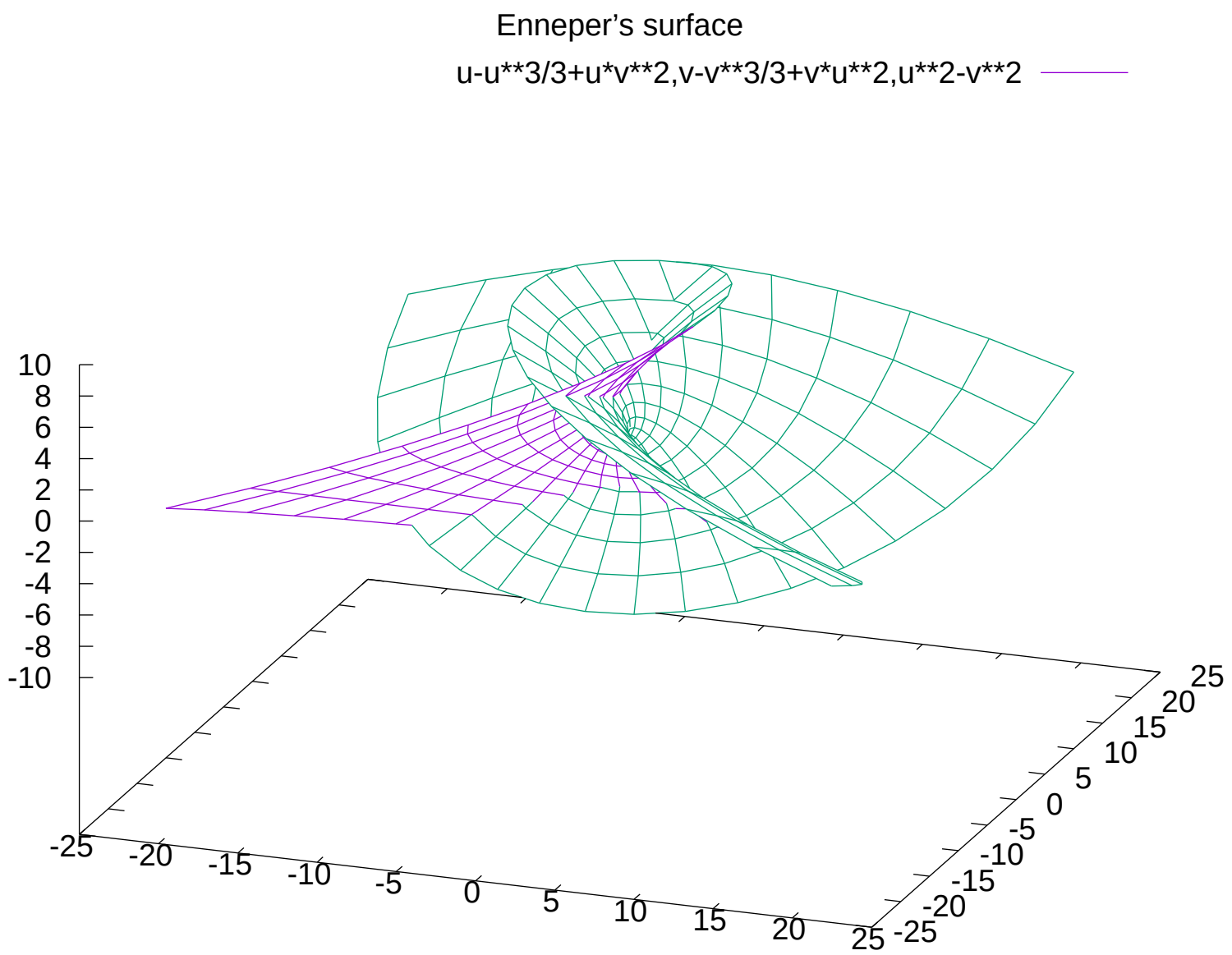
u**4-6*u**2*v**2+v**4,4*u**3*v-4*u*v**3,u




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
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
1
0
-1
-2
-3
-4

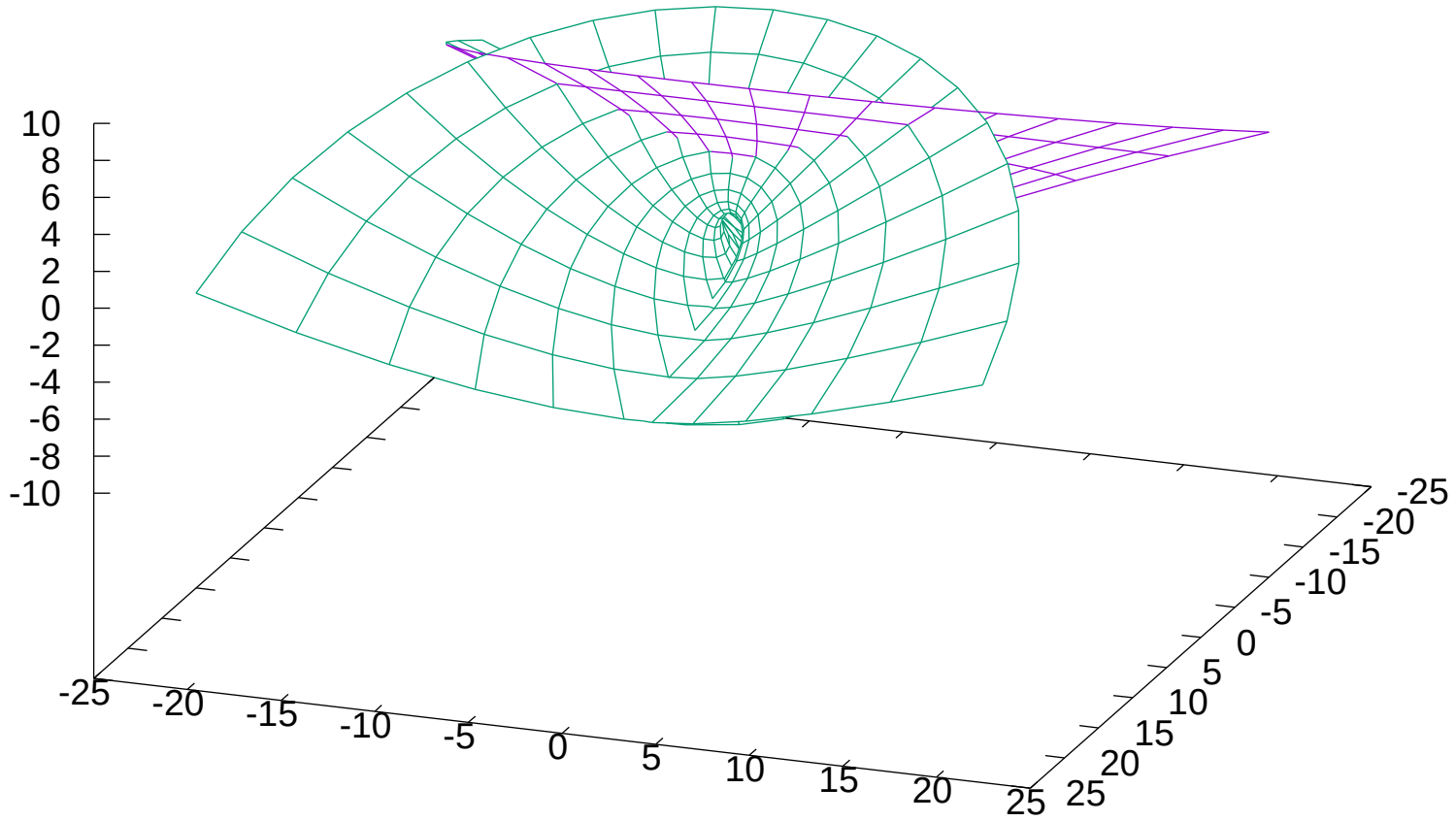
Imaginary part of complex 4th root function (different view)

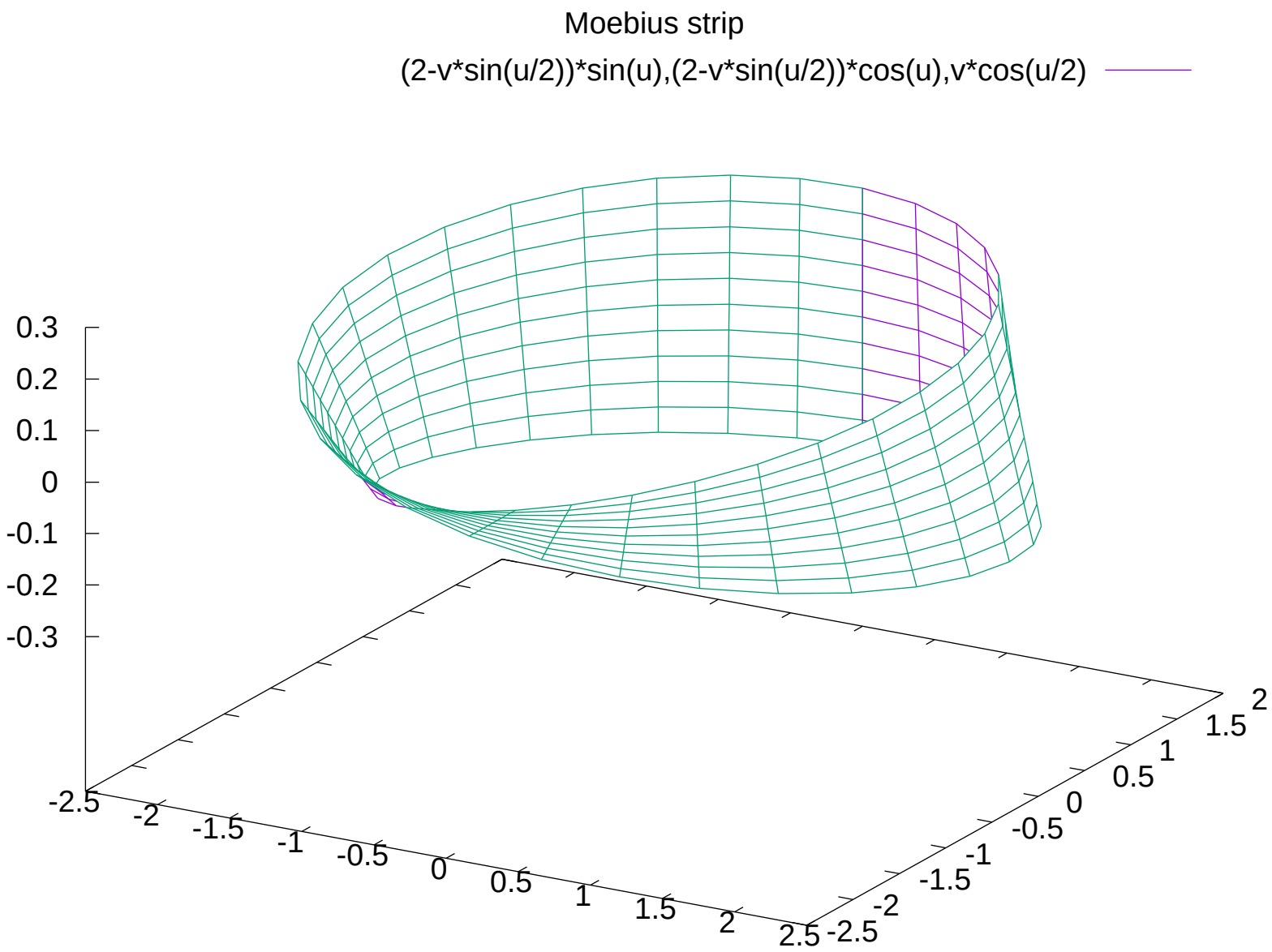


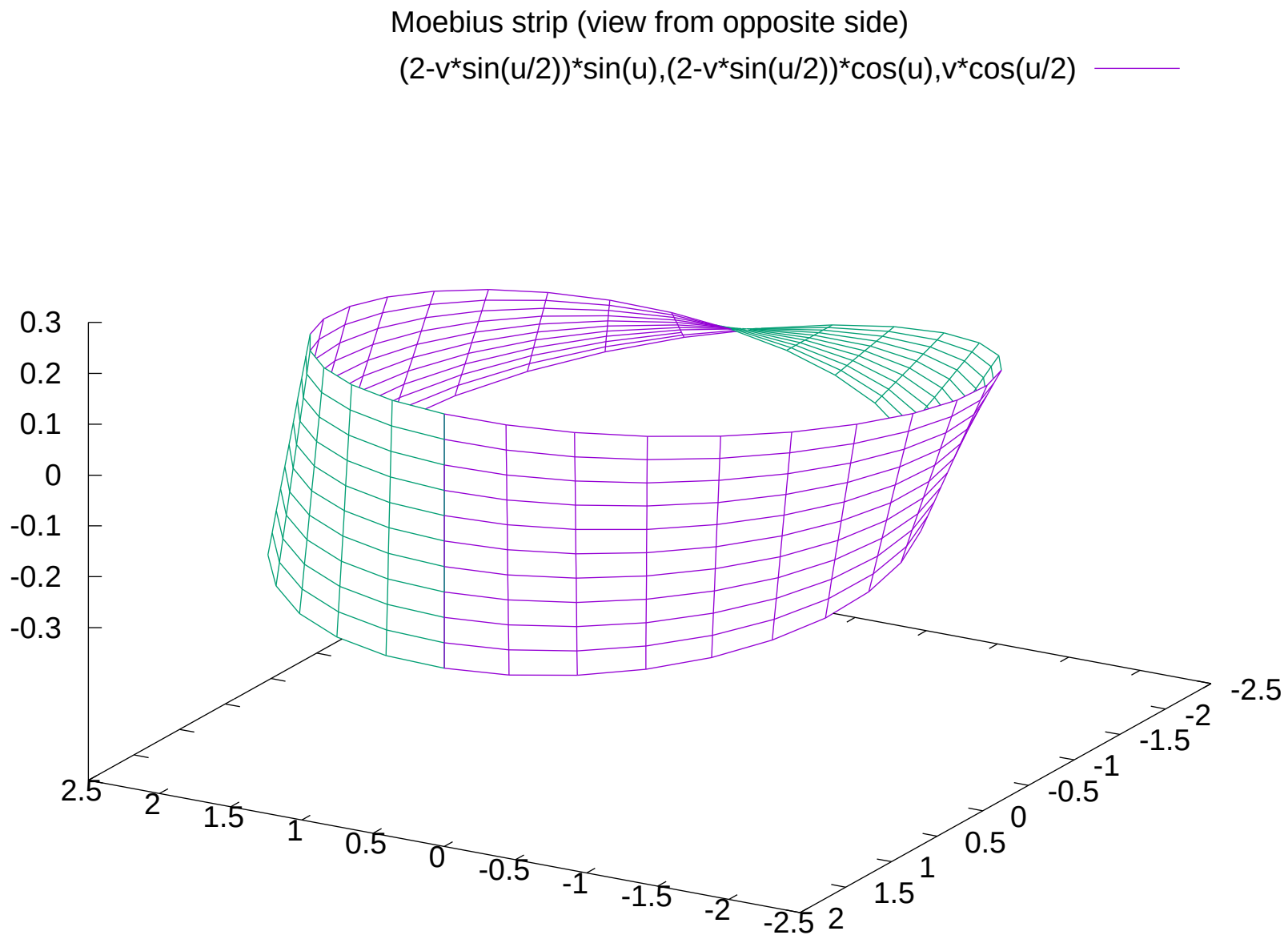

Enneper's surface
u-u**3/3+u*v**2,v-v**3/3+v*u**2,u**2-v**2
10
8
6
4
2
0
-2
-4
-6
-8
-10
-25
-20
-15
-10
-5
0
5
10
15
20
25
25
20
15
10
5
0
-5
-10
-15
-20
-25

Enneper’s surface (different view)

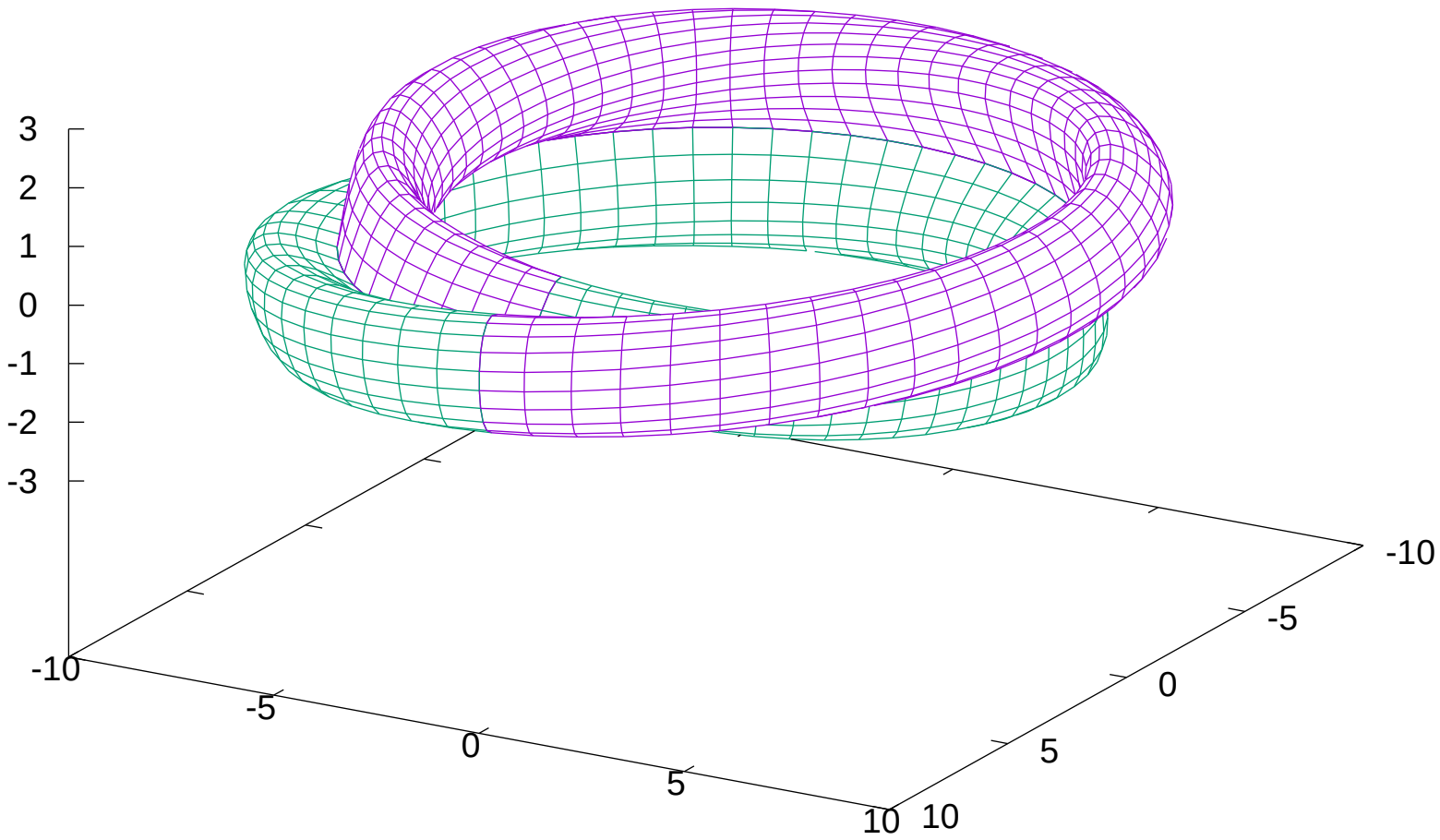
u-u**3/3+u*v**2,v-v**3/3+v*u**2,u**2-v**2



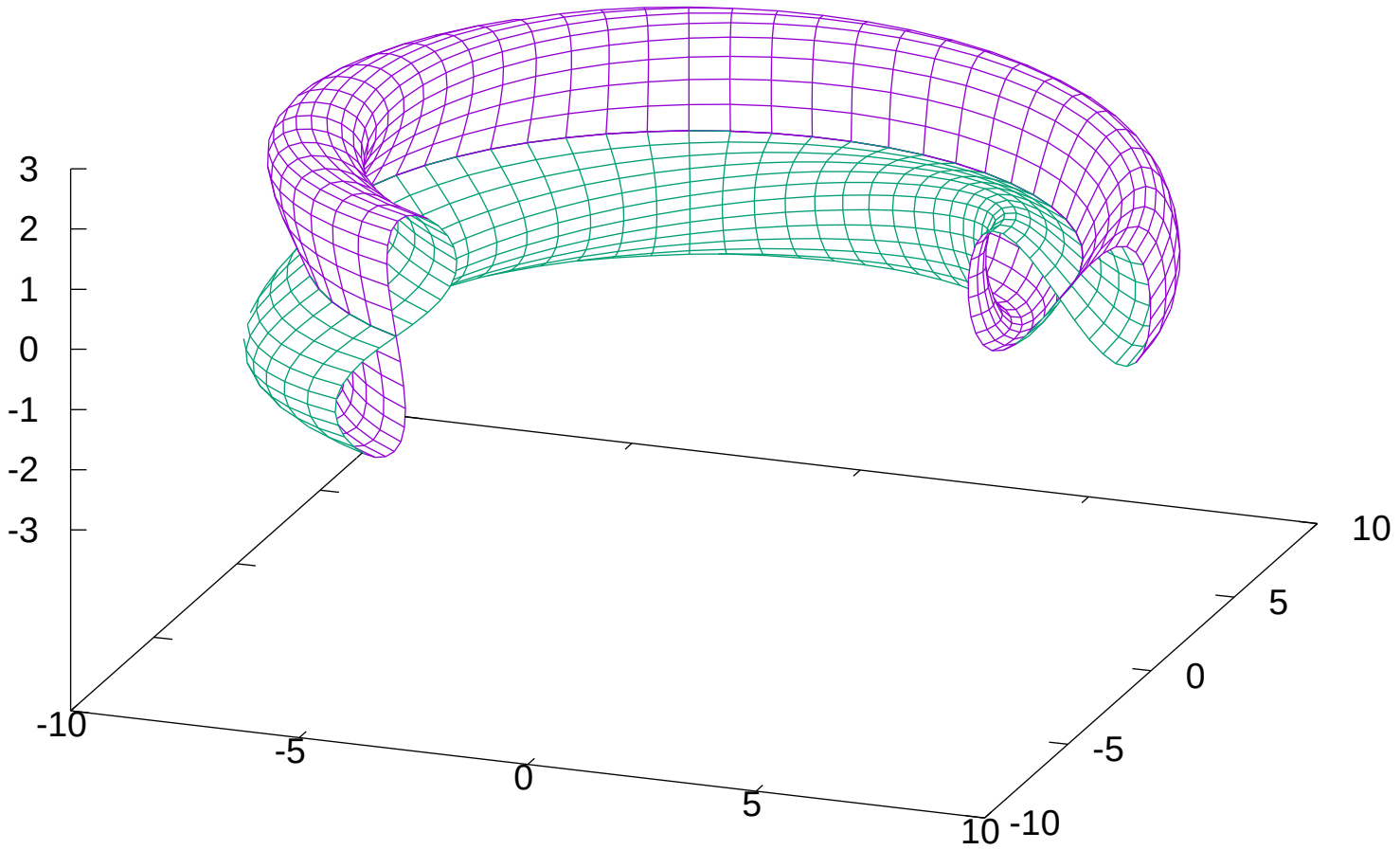

Moebius strip
(2-v*sin(u/2))*sin(u),(2-v*sin(u/2))*cos(u),v*cos(u/2)
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2


Moebius strip (view from opposite side)
(2-v*sin(u/2))*sin(u),(2-v*sin(u/2))*cos(u),v*cos(u/2)
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5

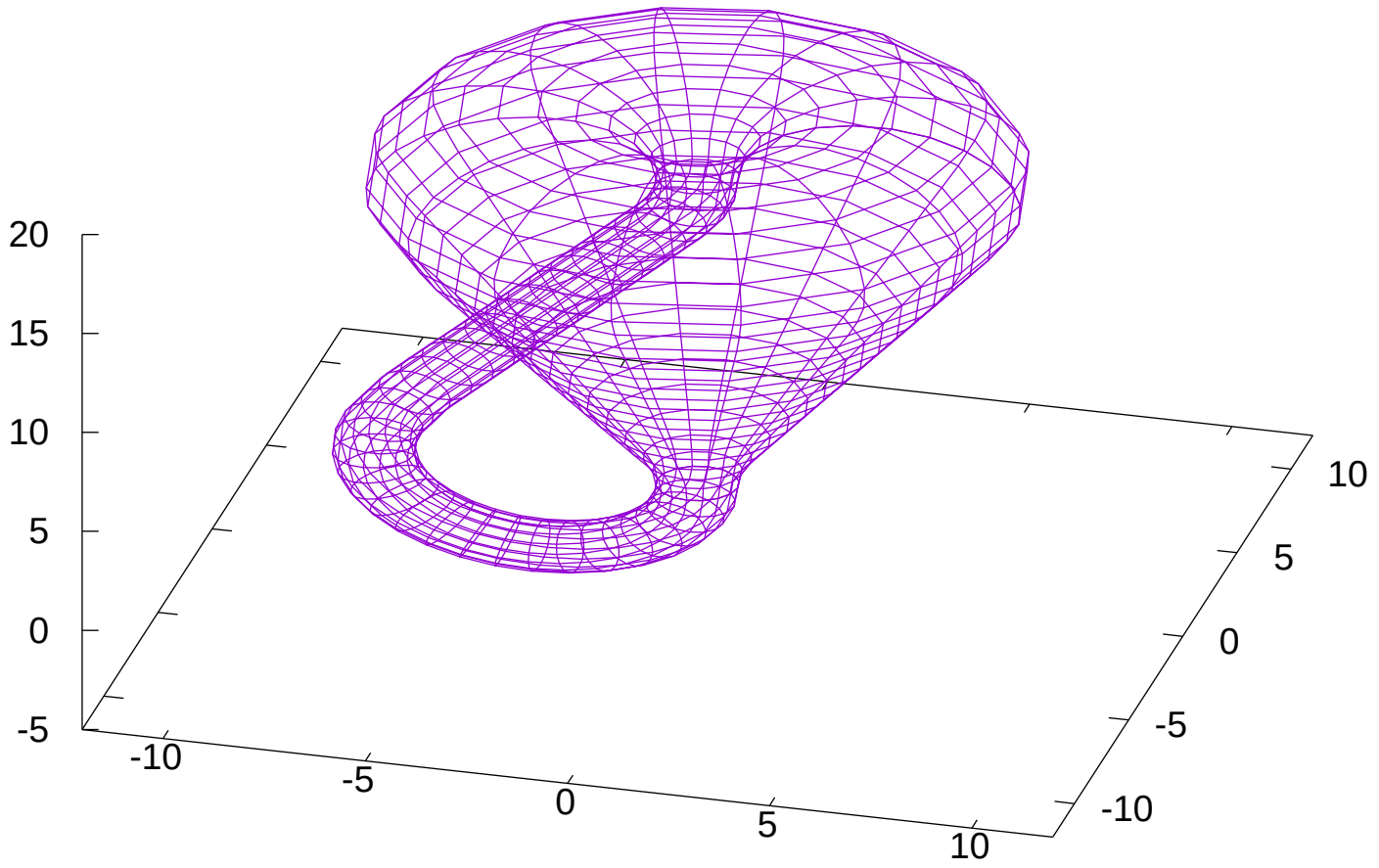
Klein bottle



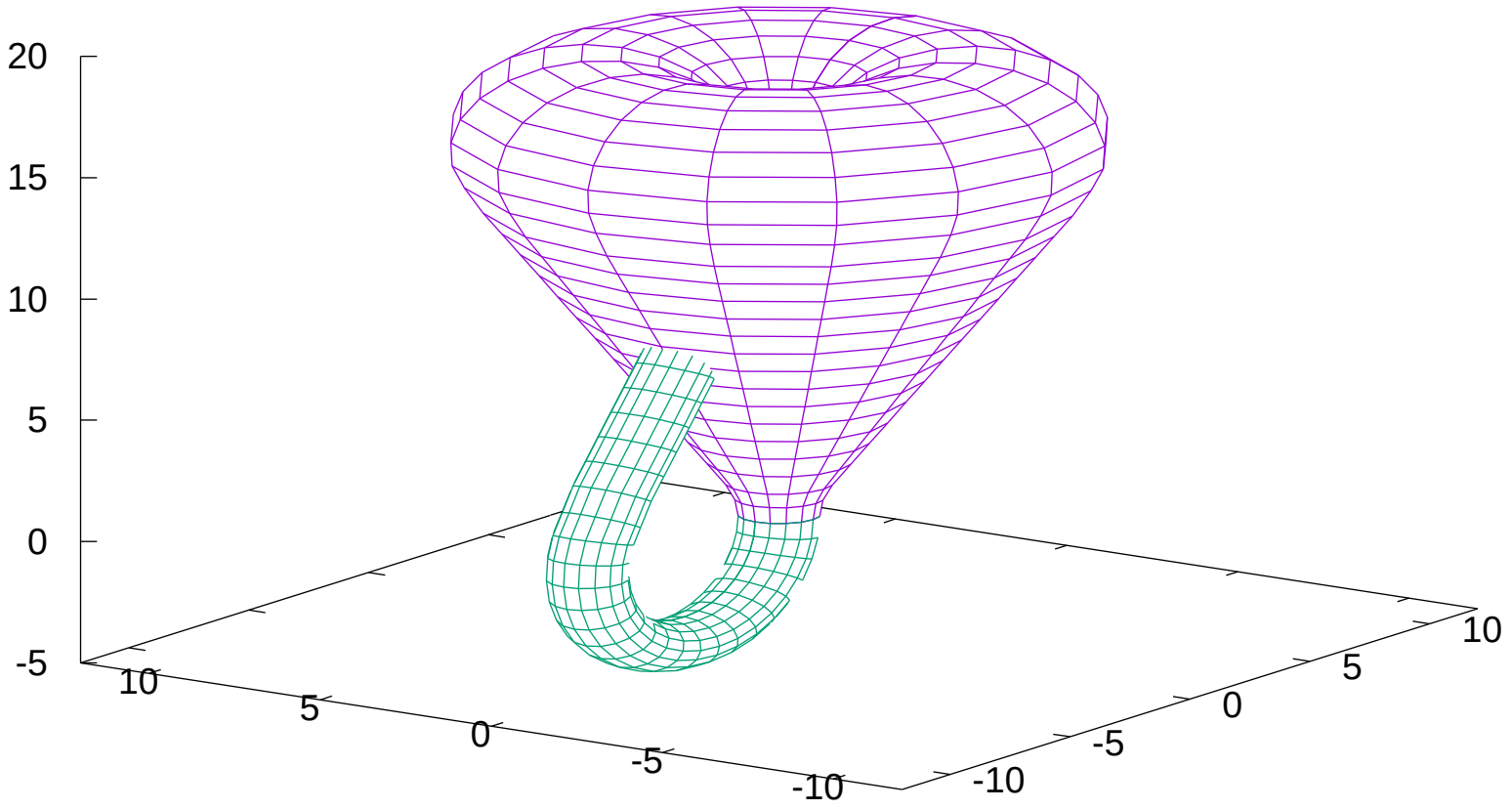
Klein bottle with look at the ’inside’

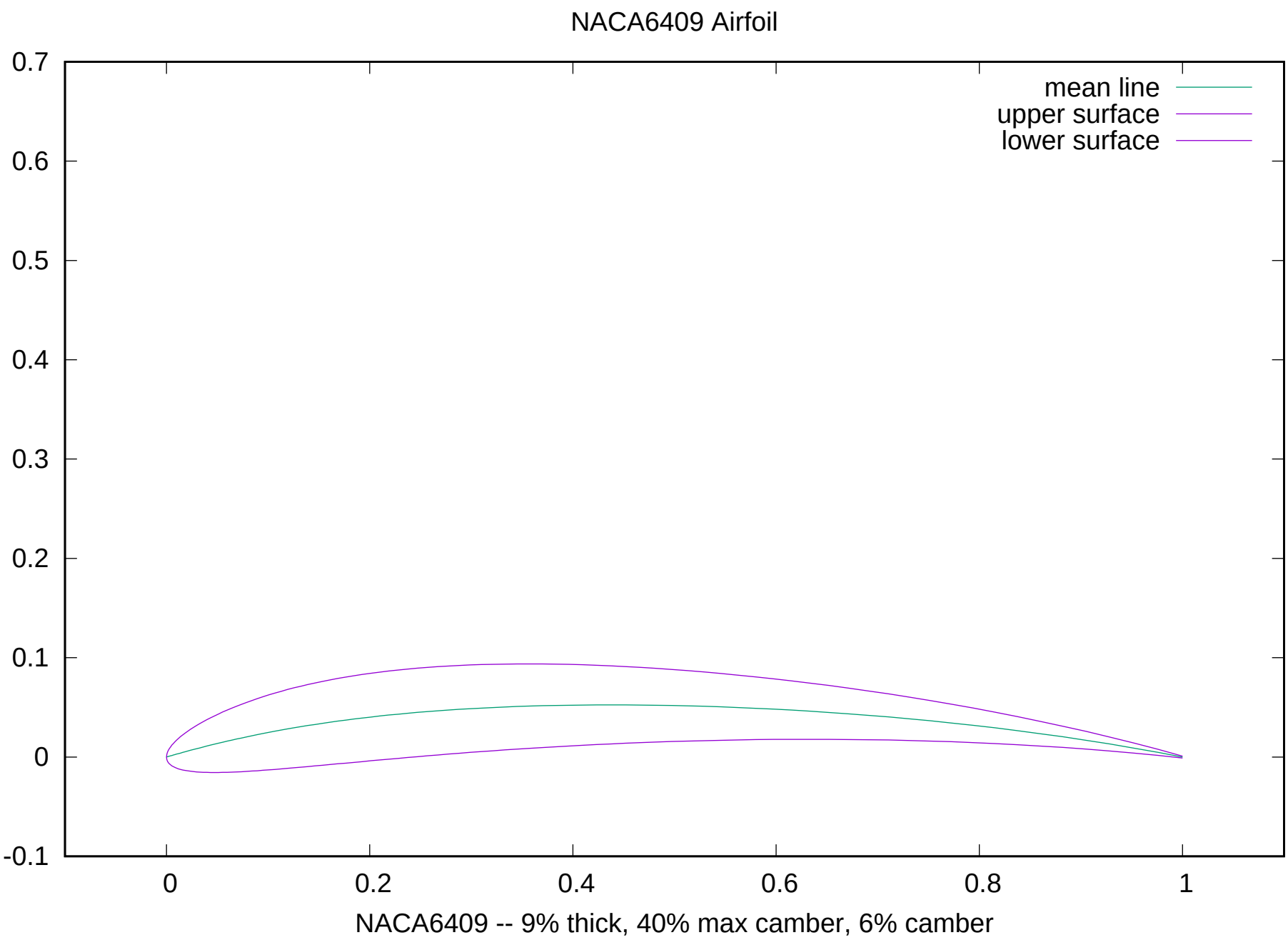


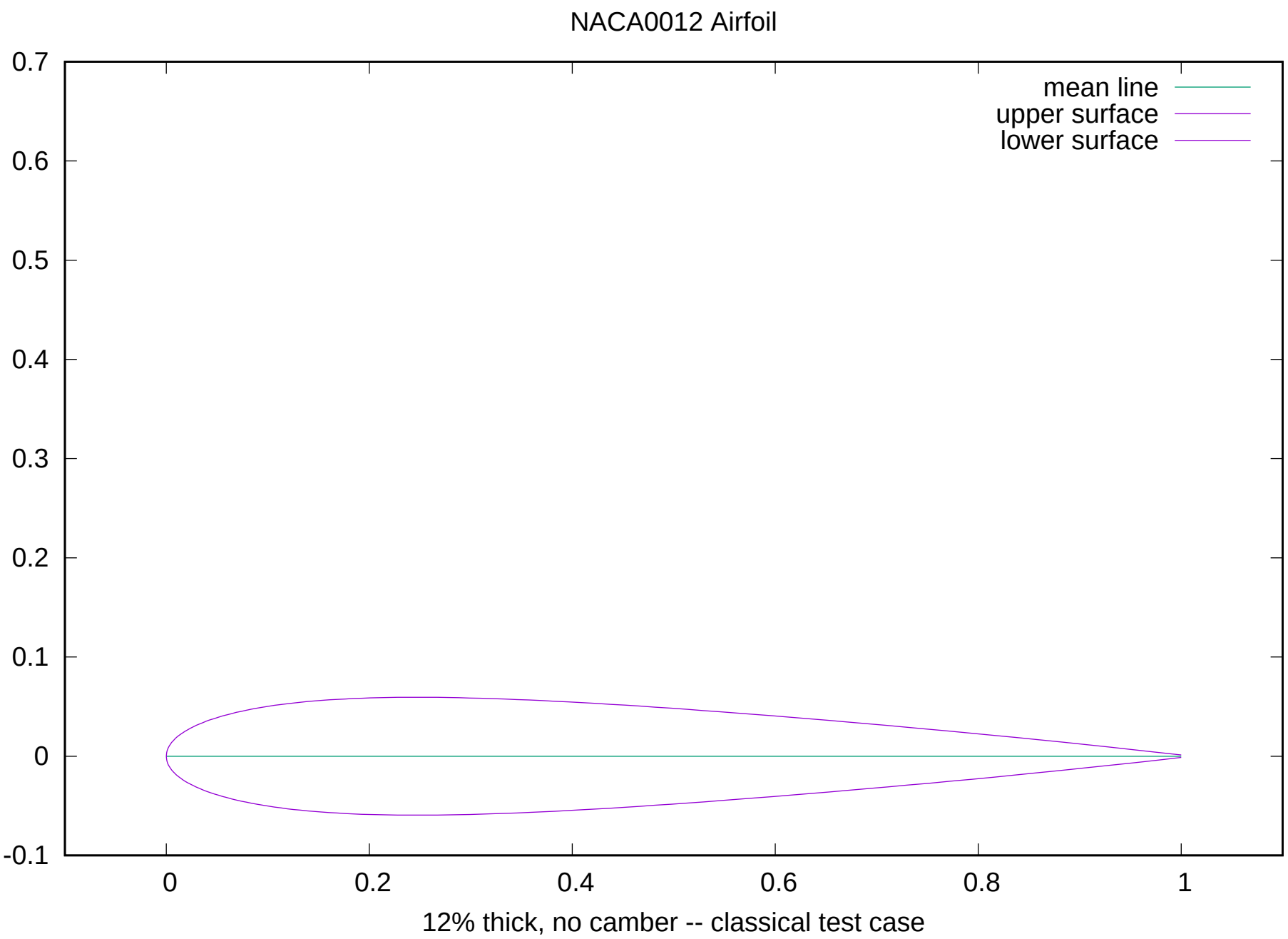
Klein bottle, glassblowers' version (look-through)

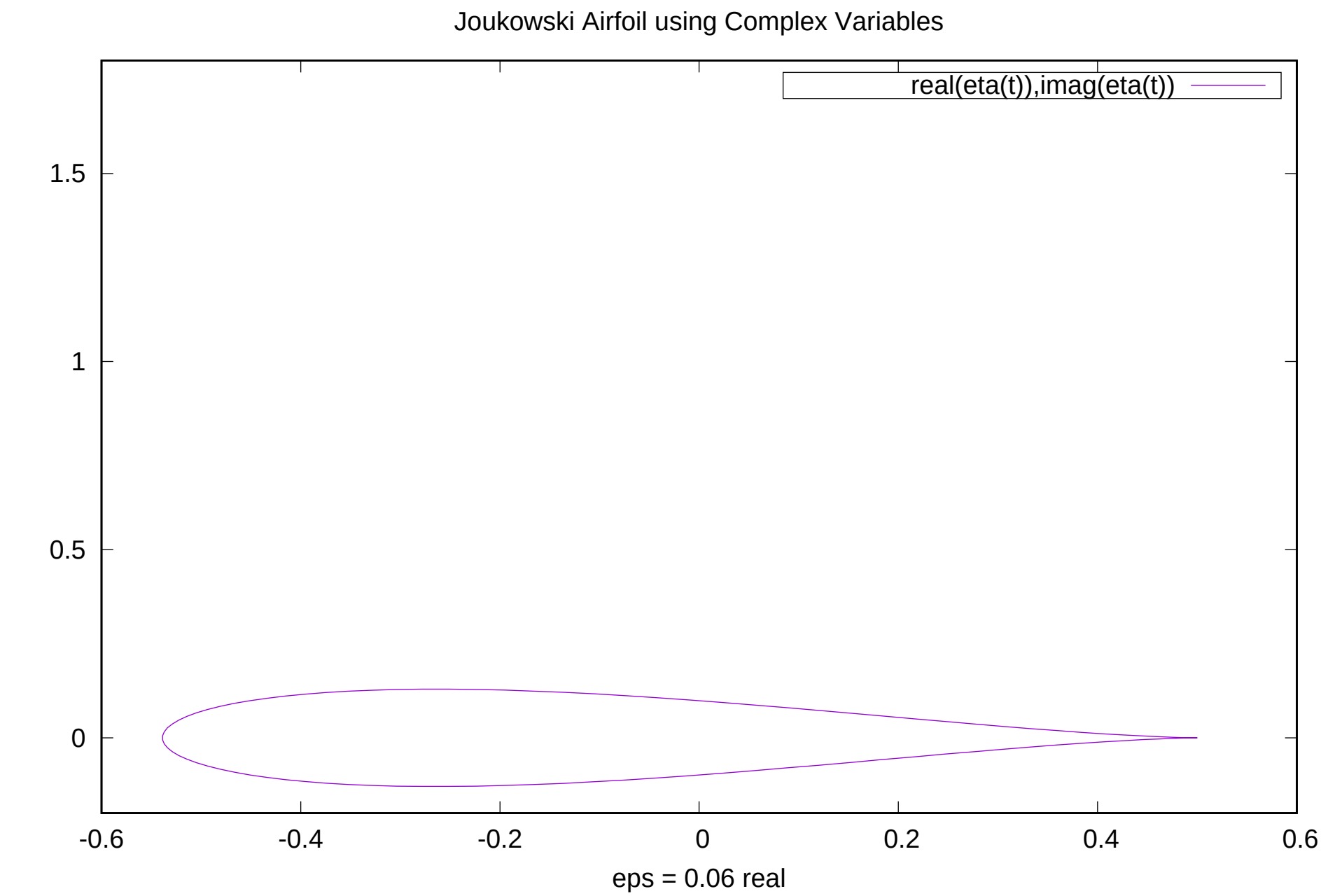


Klein bottle, glassblowers' version (solid)

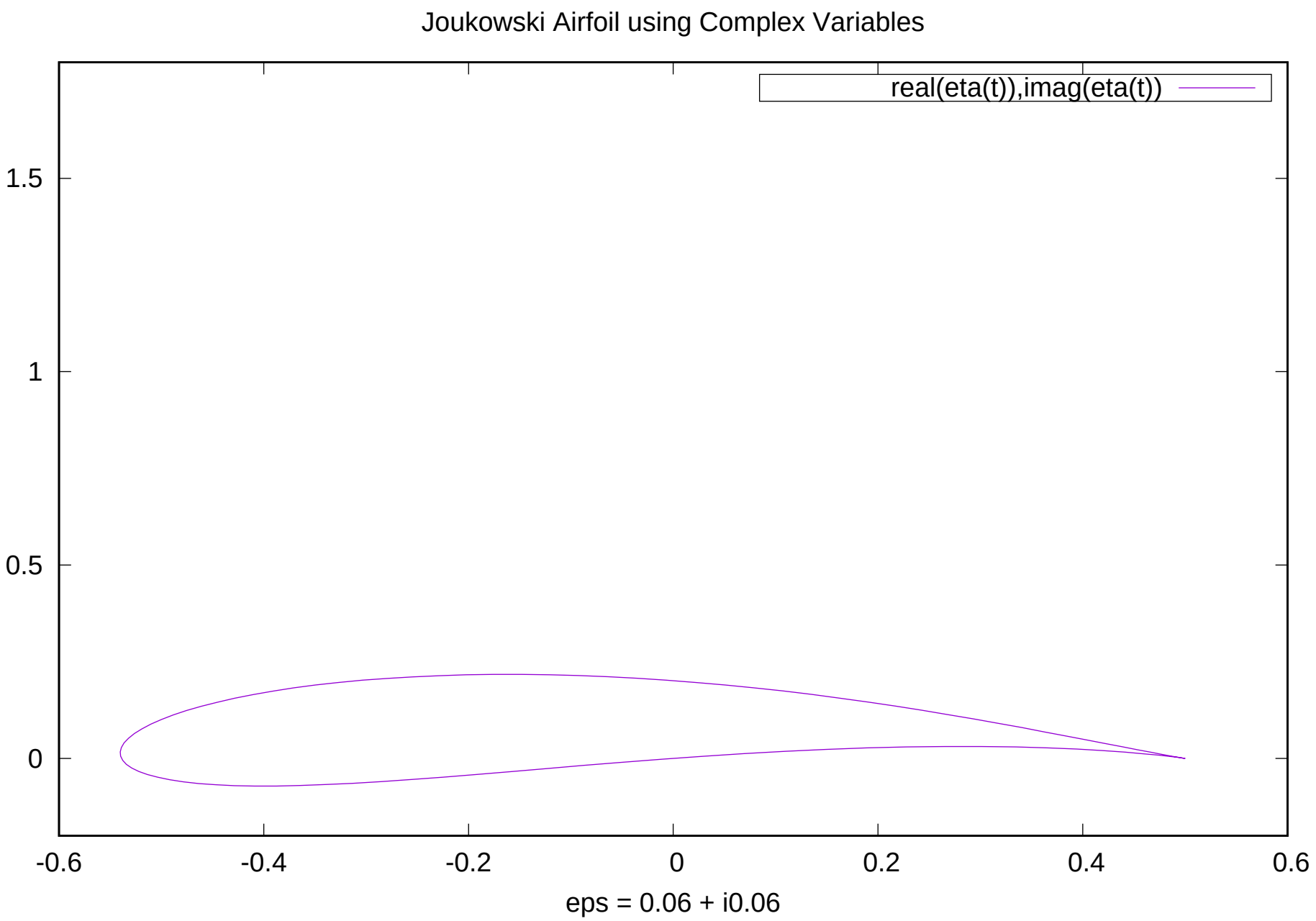



NACA6409 Airfoil
mean line
upper surface
lower surface
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
0
0.2
0.4
0.6
0.8
1
NACA6409 -- 9% thick, 40% max camber, 6% camber

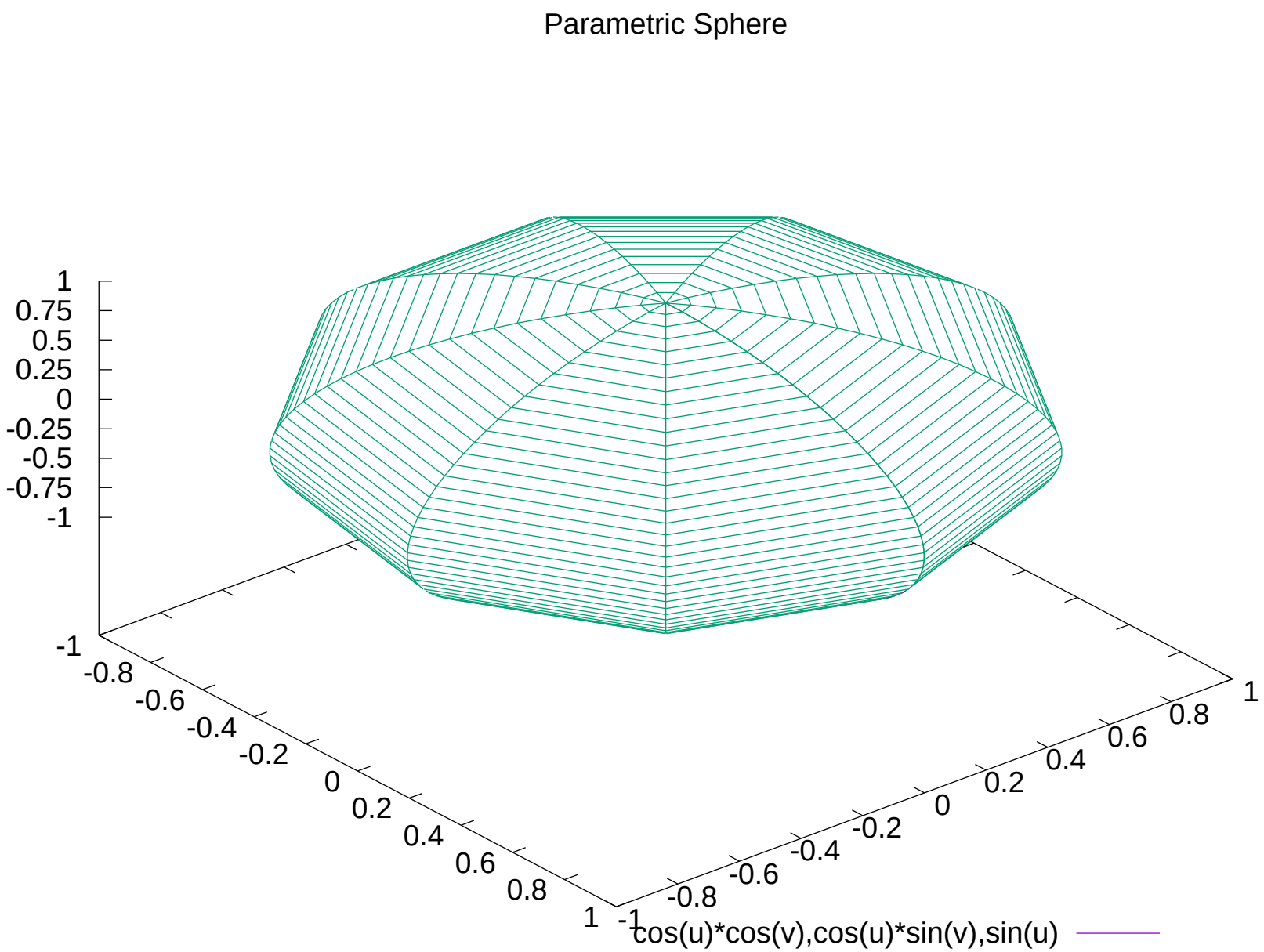

NACA0012 Airfoil
mean line
upper surface
lower surface
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
0
0.2
0.4
0.6
0.8
1
12% thick, no camber -- classical test case



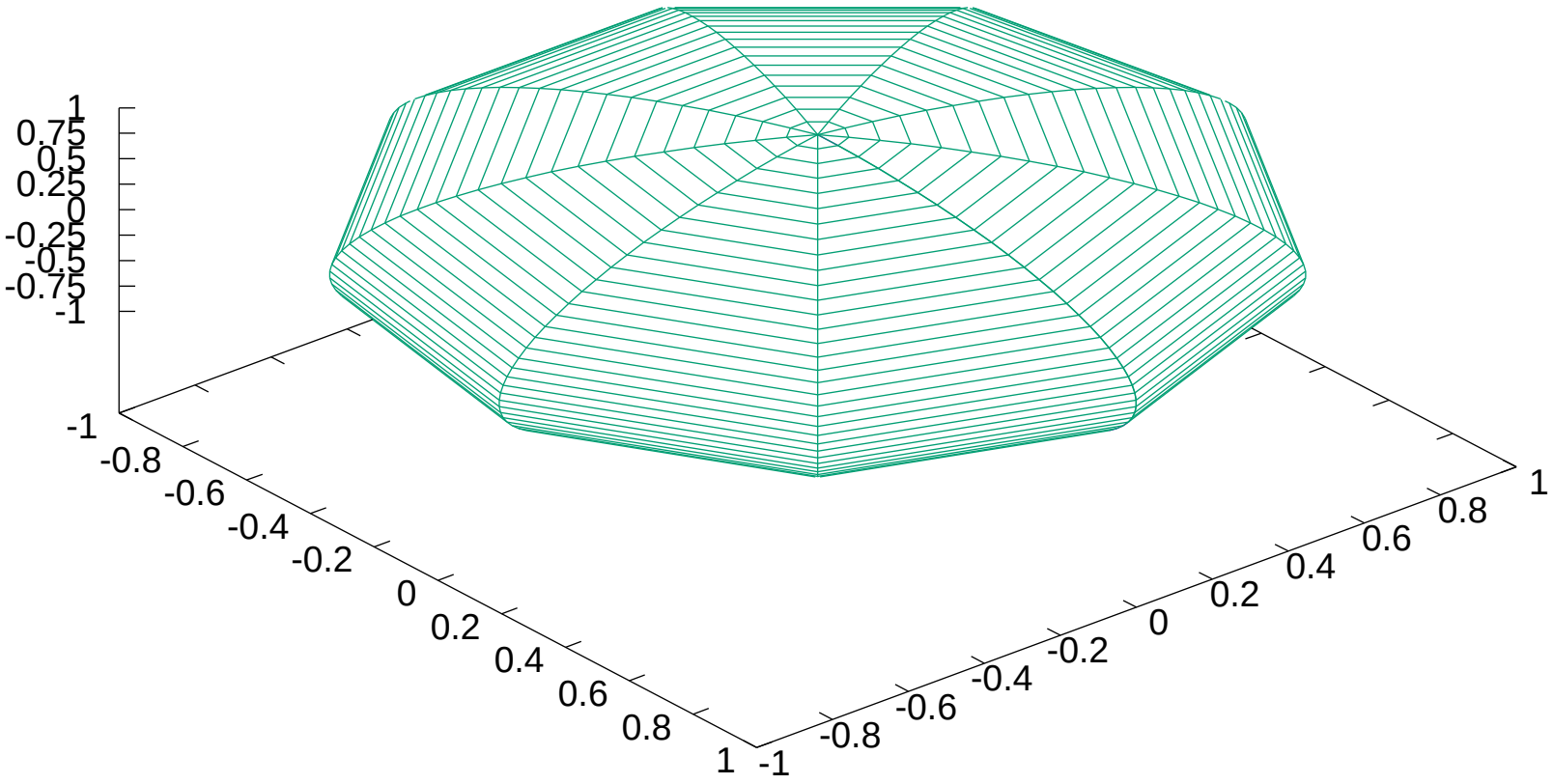
(¶â 10•î 07 14:40:09 2016

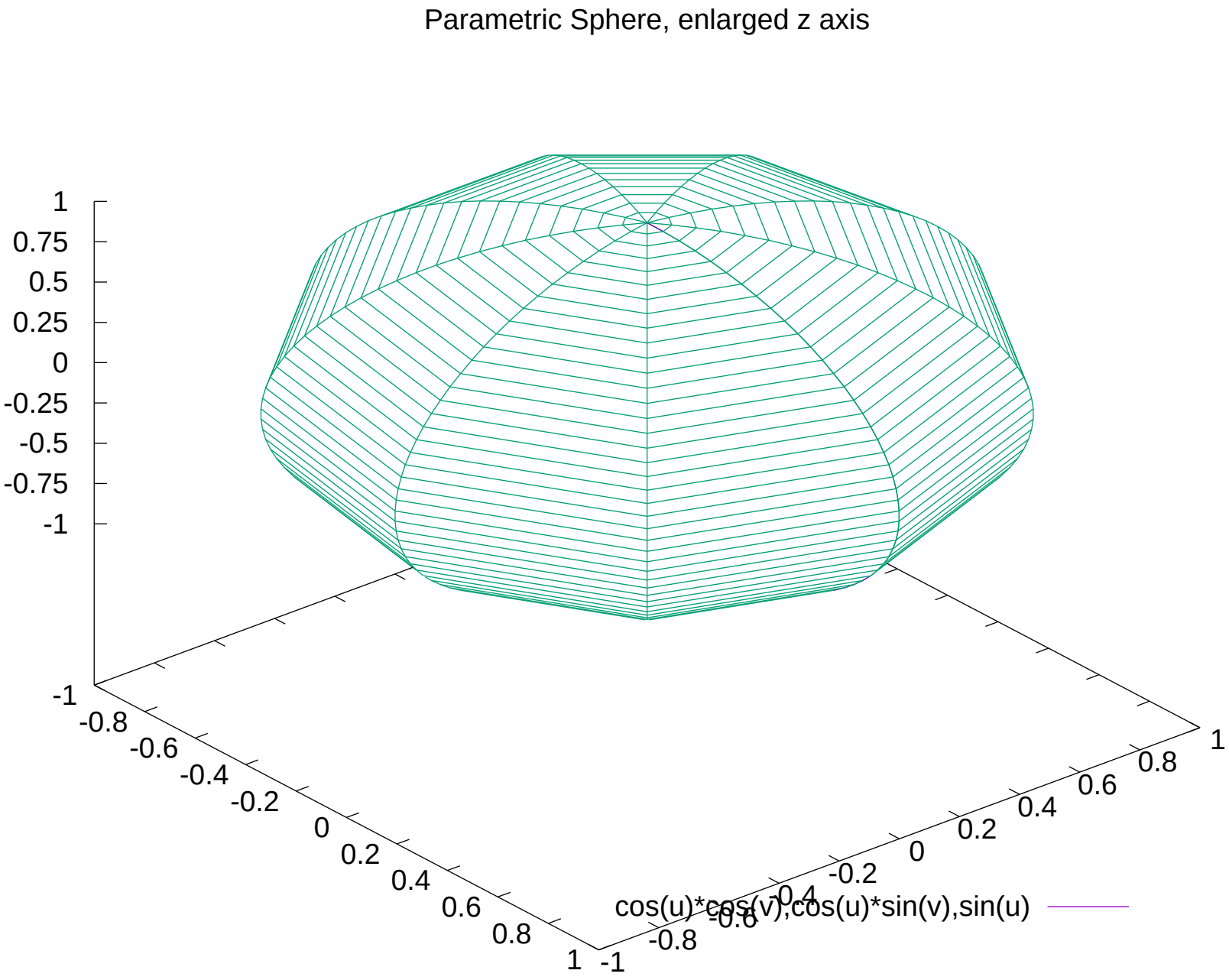


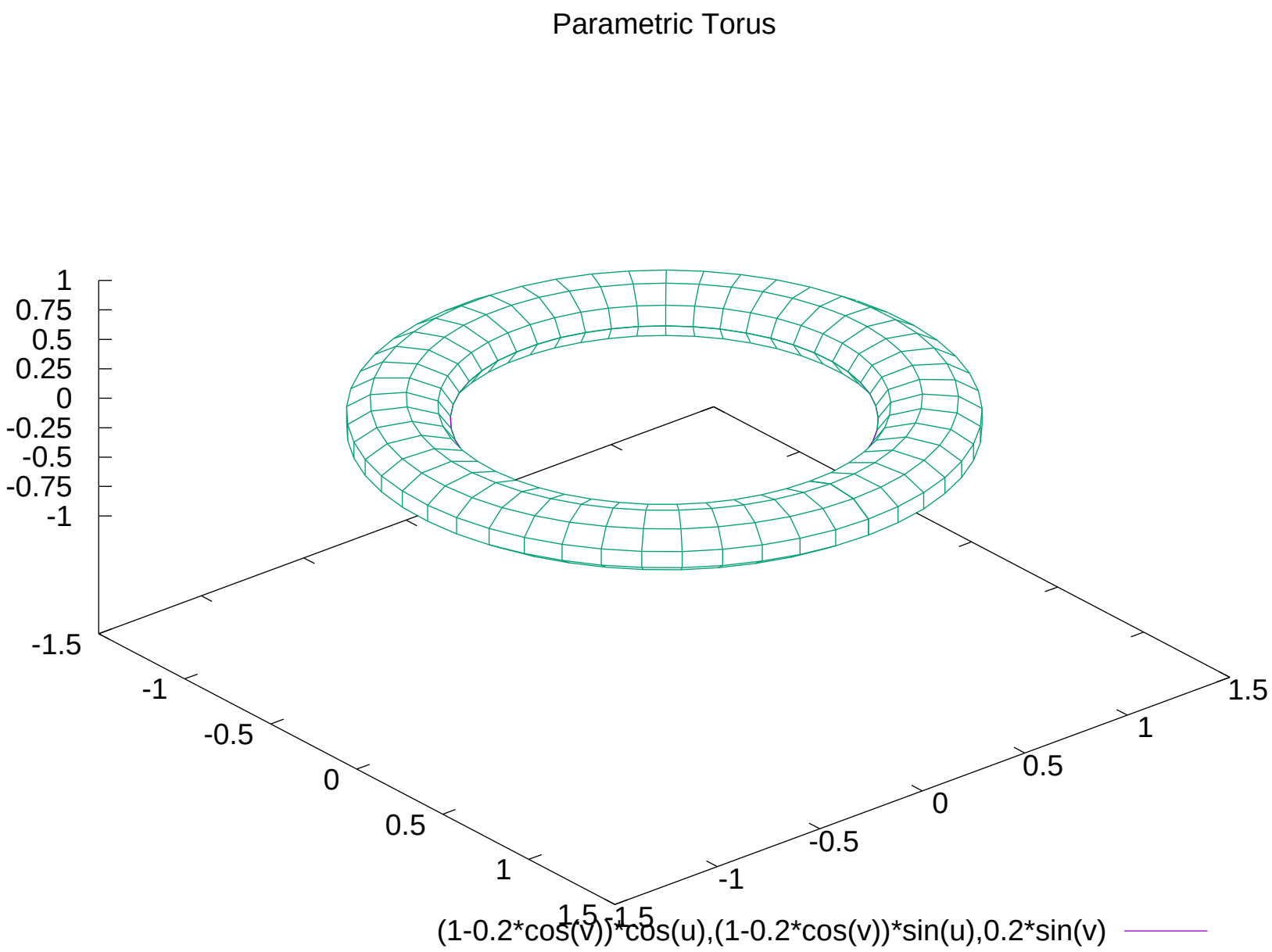
(¶â 10•î 07 14:40:09 2016


Parametric Sphere
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
cos(u)*cos(v),cos(u)*sin(v),sin(u)

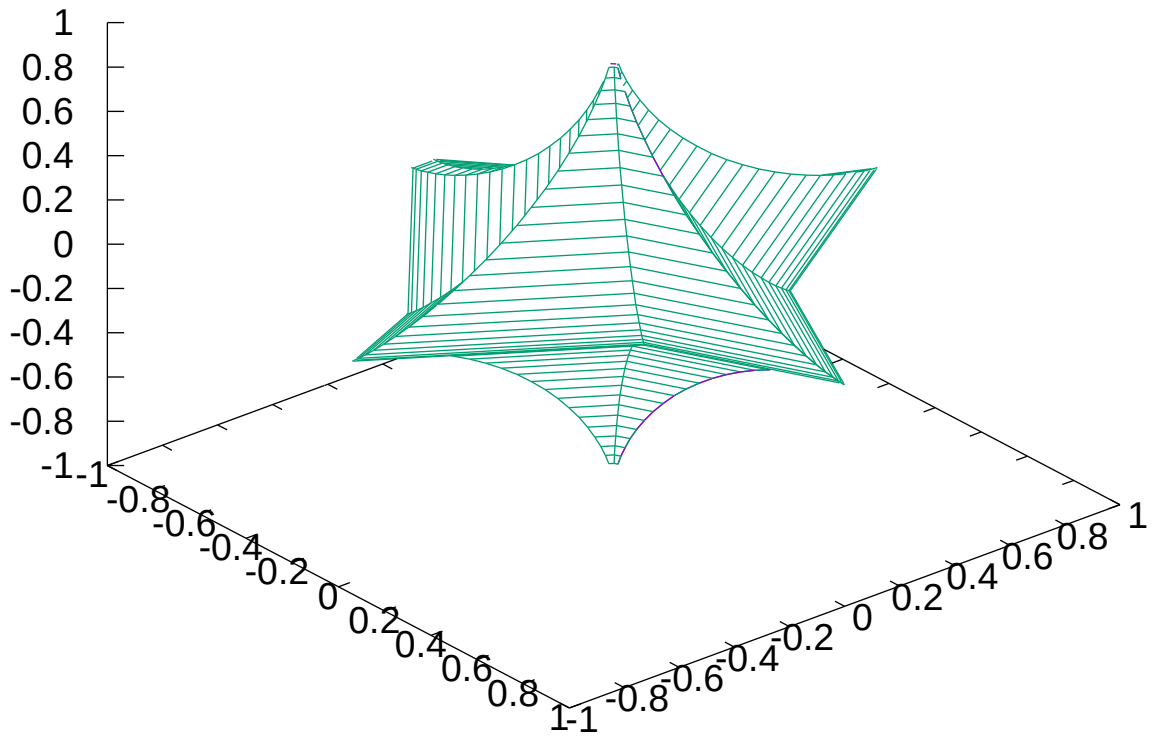
Parametric Sphere, crunched z axis



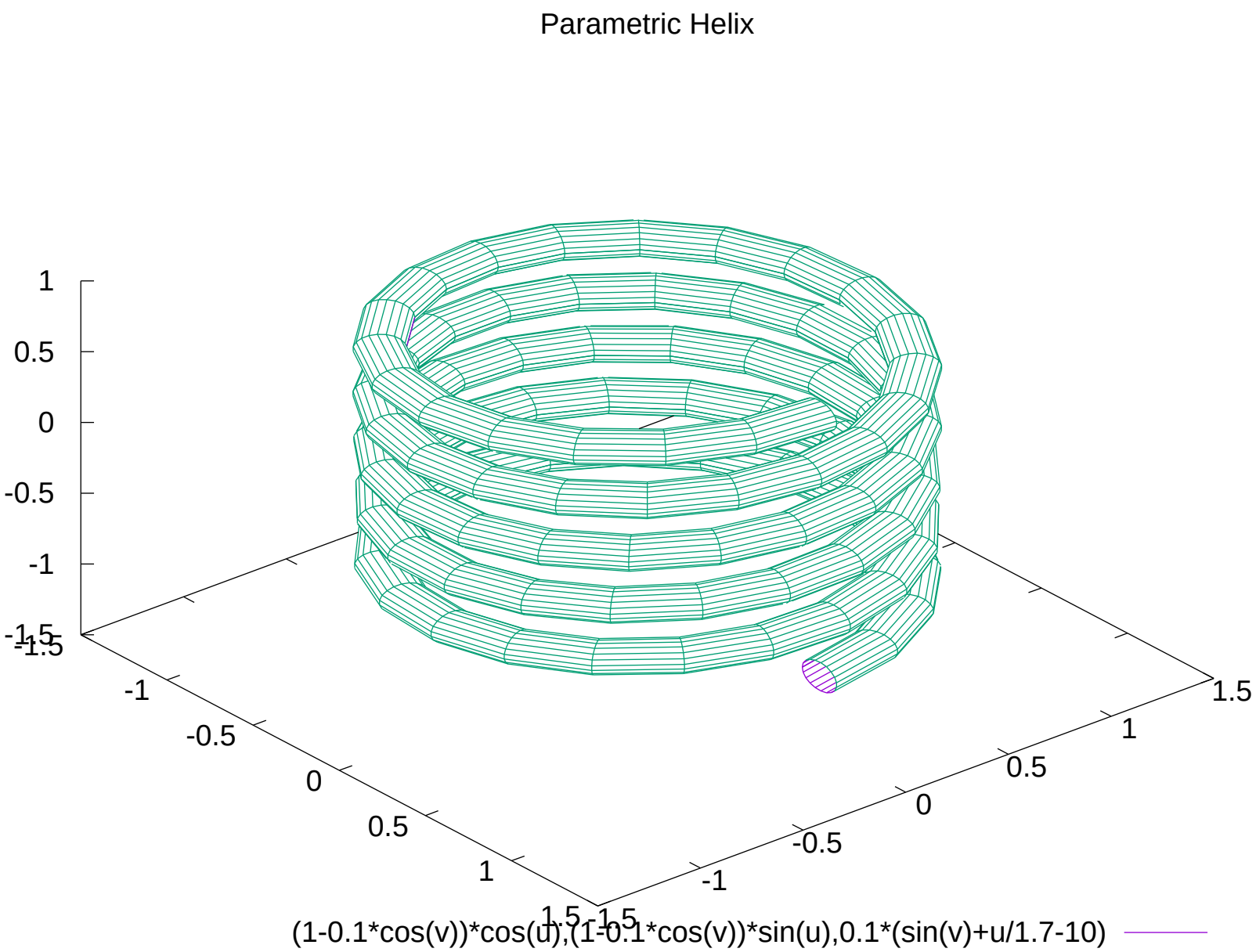

Parametric Sphere, enlarged z axis
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
cos(u)*cos(v),cos(u)*sin(v),sin(u)


Parametric Torus
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1
-1.5
-1
-0.5
0
0.5
1
1.5
1.5
1
0.5
0
-0.5
-1
-1.5
(1-0.2*cos(v))*cos(u),(1-0.2*cos(v))*sin(u),0.2*sin(v)

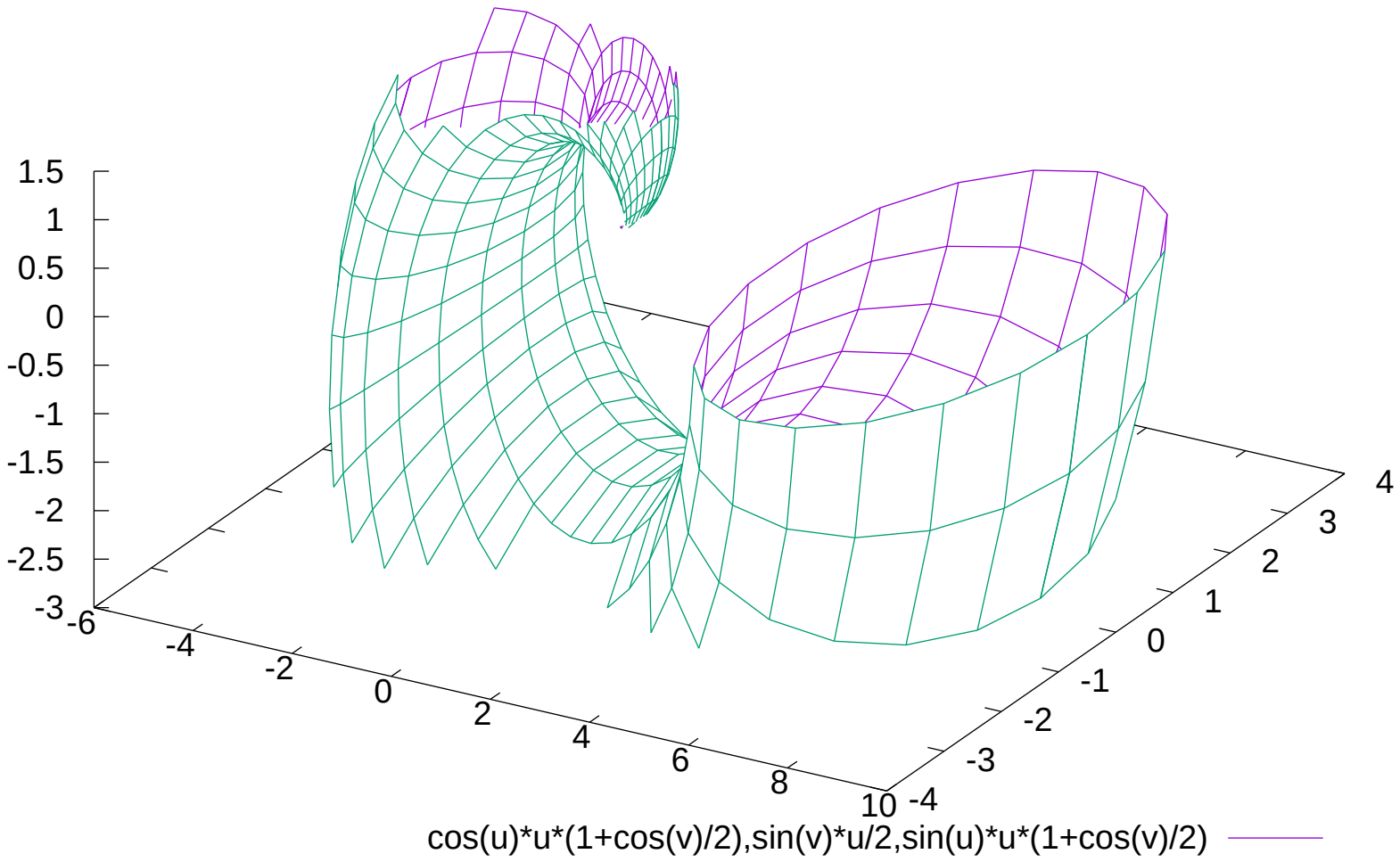
Parametric Hexagon

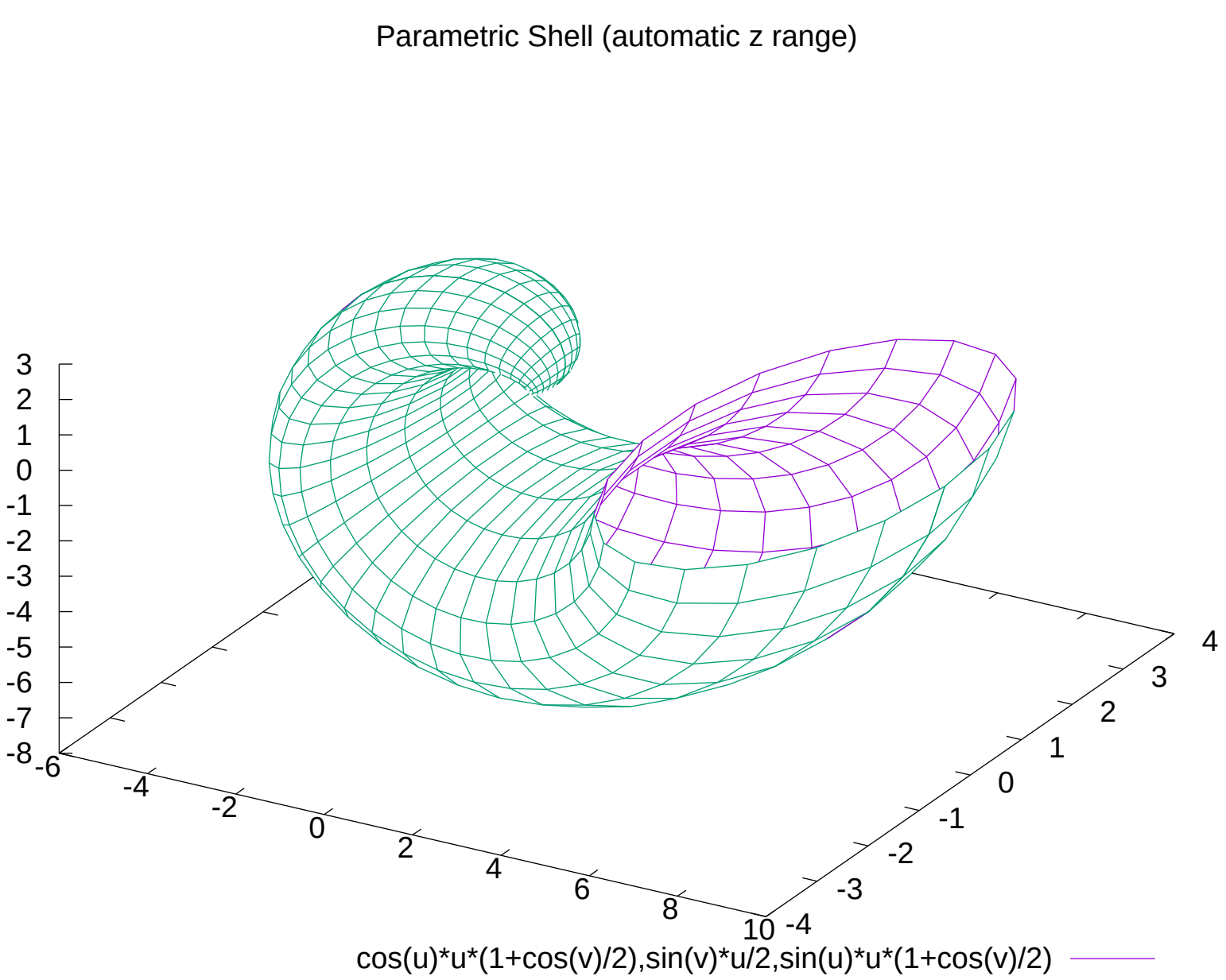


cos(v)**3*cos(u)**3,sin(v)**3*cos(u)**3,sin(u)**3

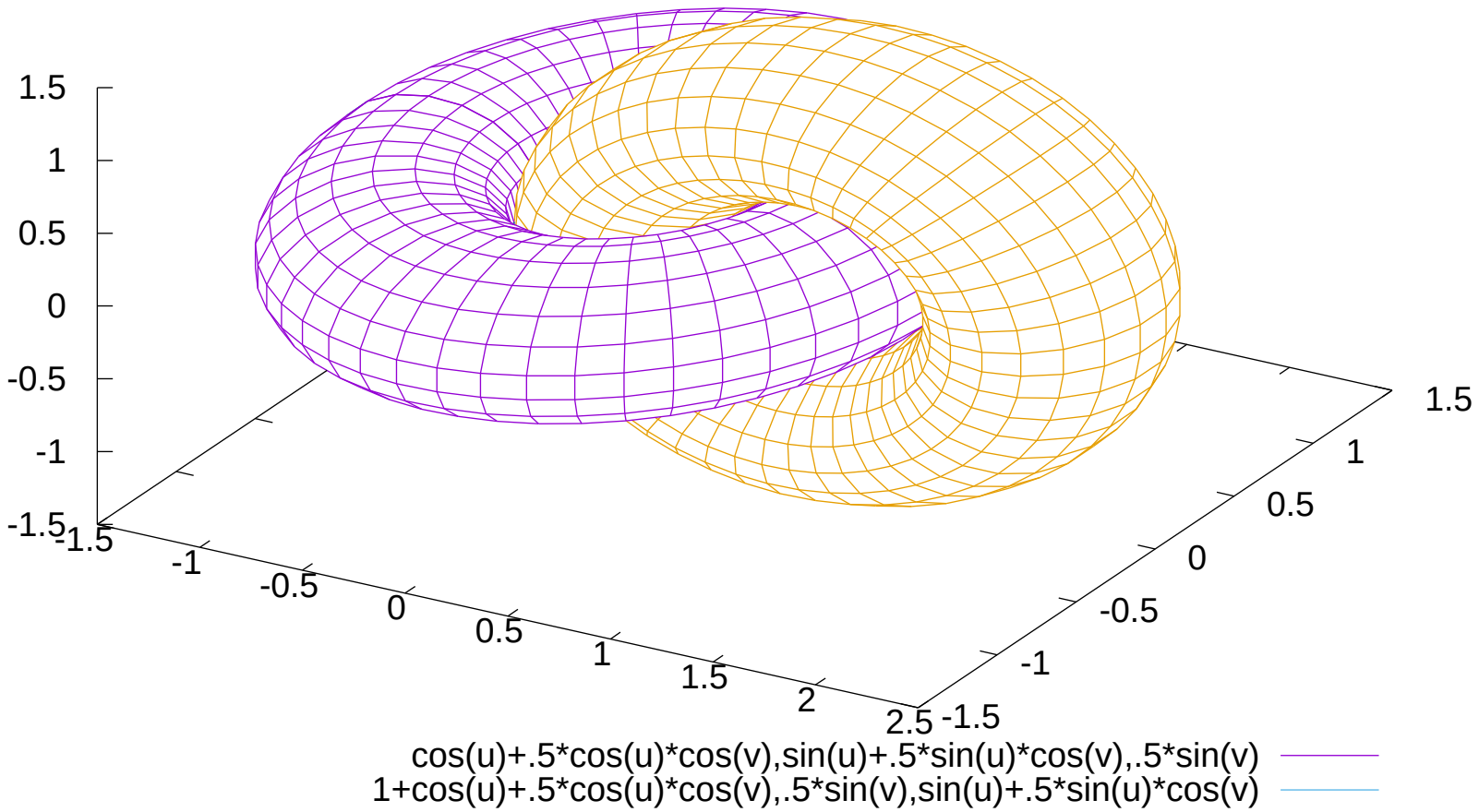

Parametric Helix
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5
1.5
1
0.5
0
-0.5
-1
-1.5
(1-0.1*cos(v))*cos(u),(1-0.1*cos(v))*sin(u),0.1*(sin(v)+u/1.7-10)

Parametric Shell (clipped to limited z range)

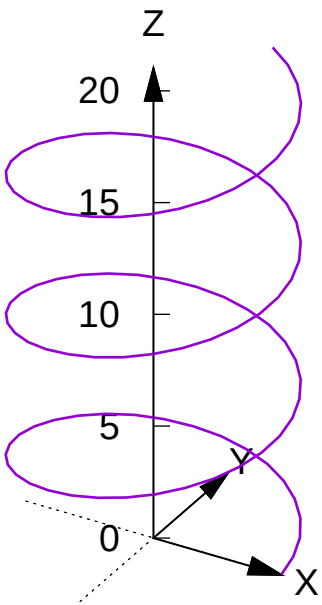



Parametric Shell (automatic z range)
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-6
-4
-2
0
2
4
6
8
10
-4
-3
-2
-1
0
1
2
3
4
cos(u)*u*(1+cos(v)/2),sin(v)*u/2,sin(u)*u*(1+cos(v)/2)

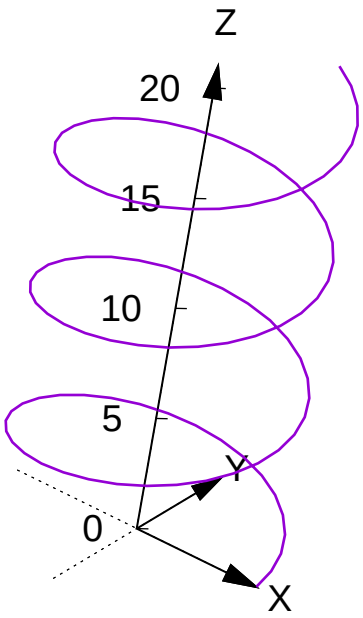
Interlocking Tori



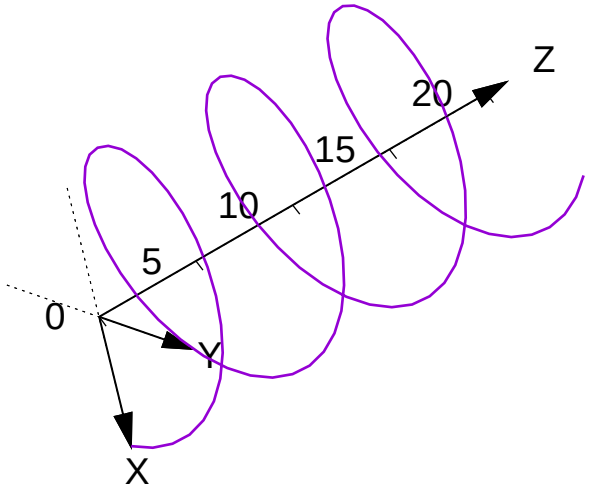
azimuth 0

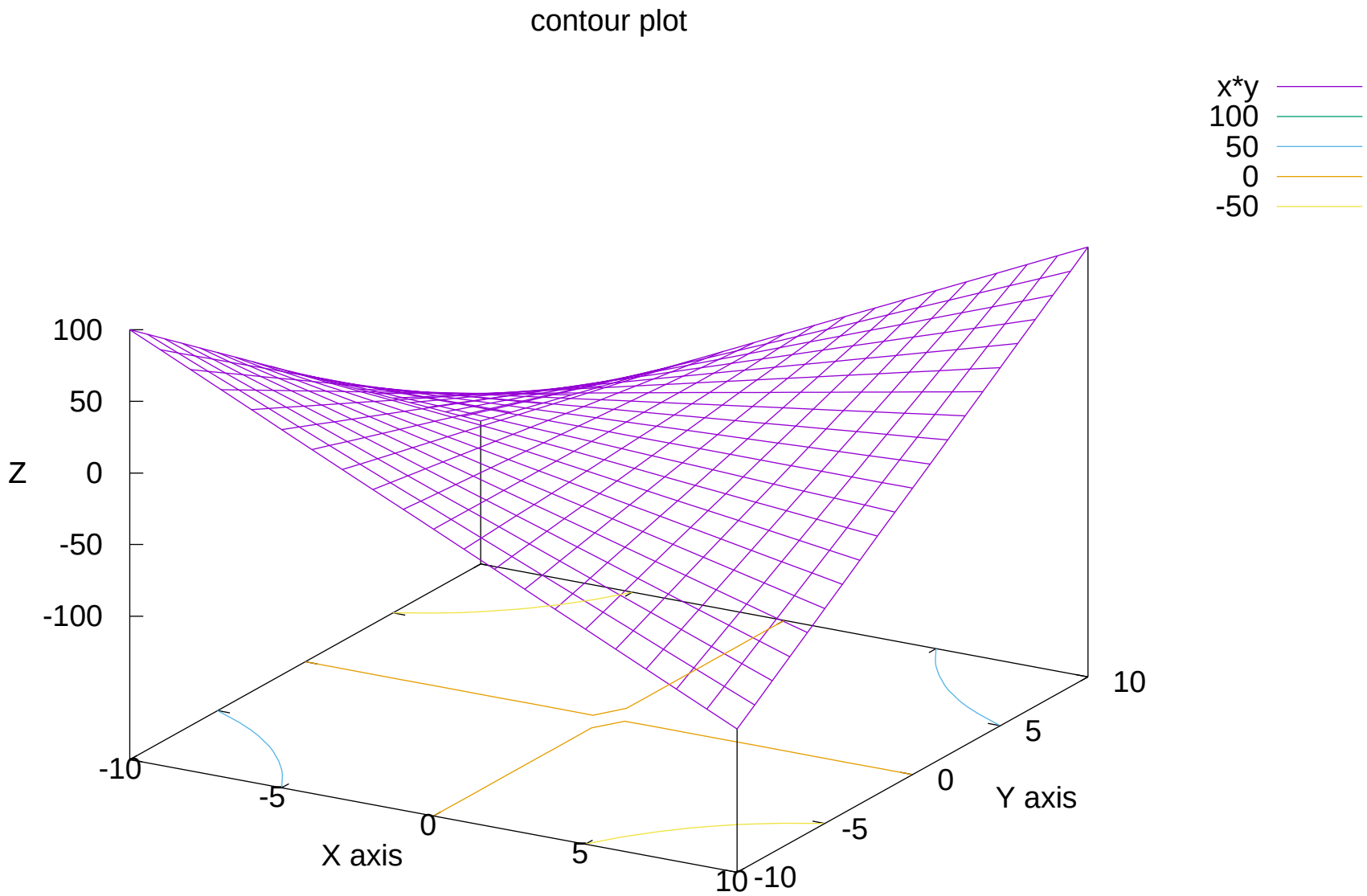


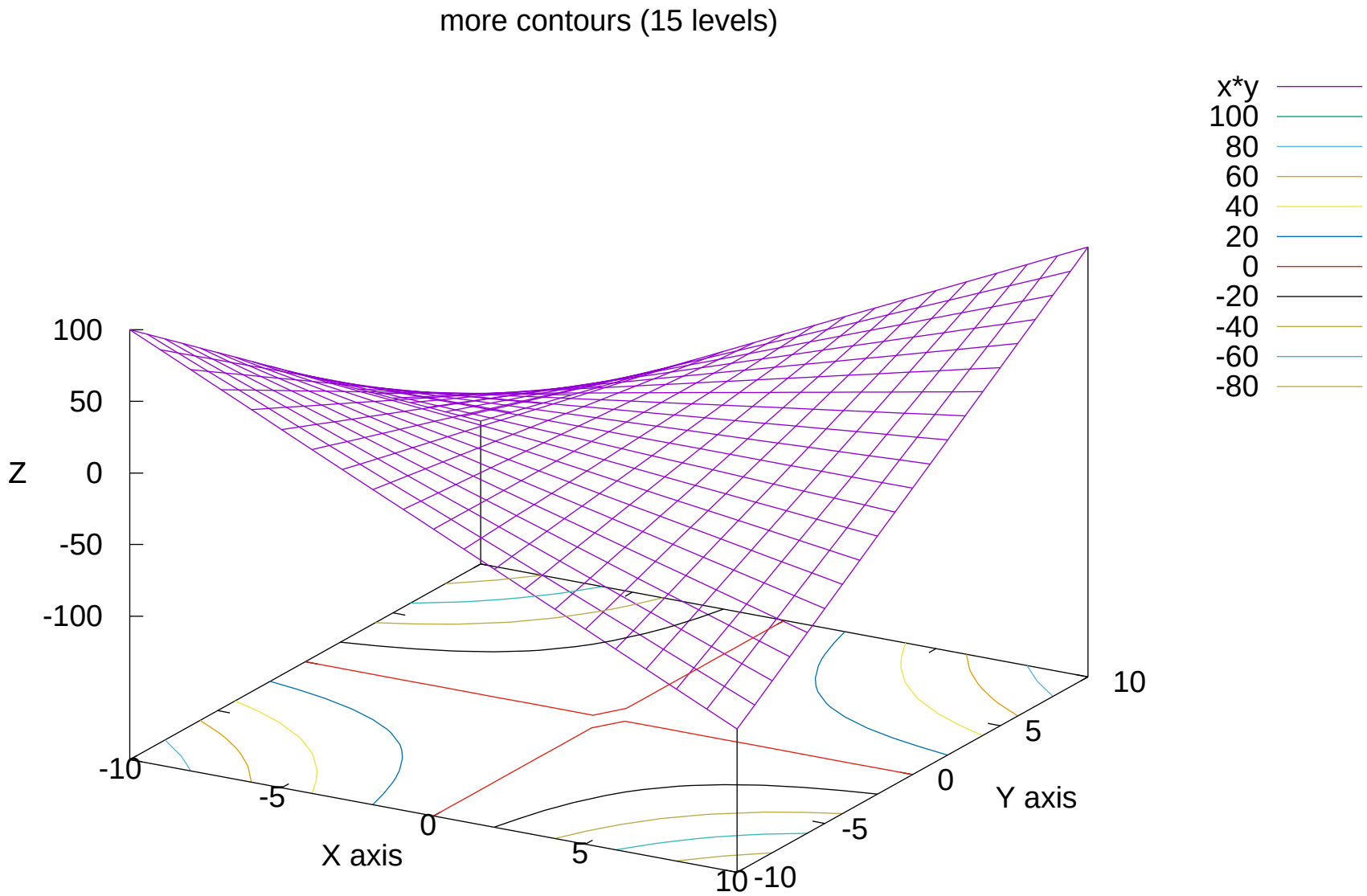
azimuth 10

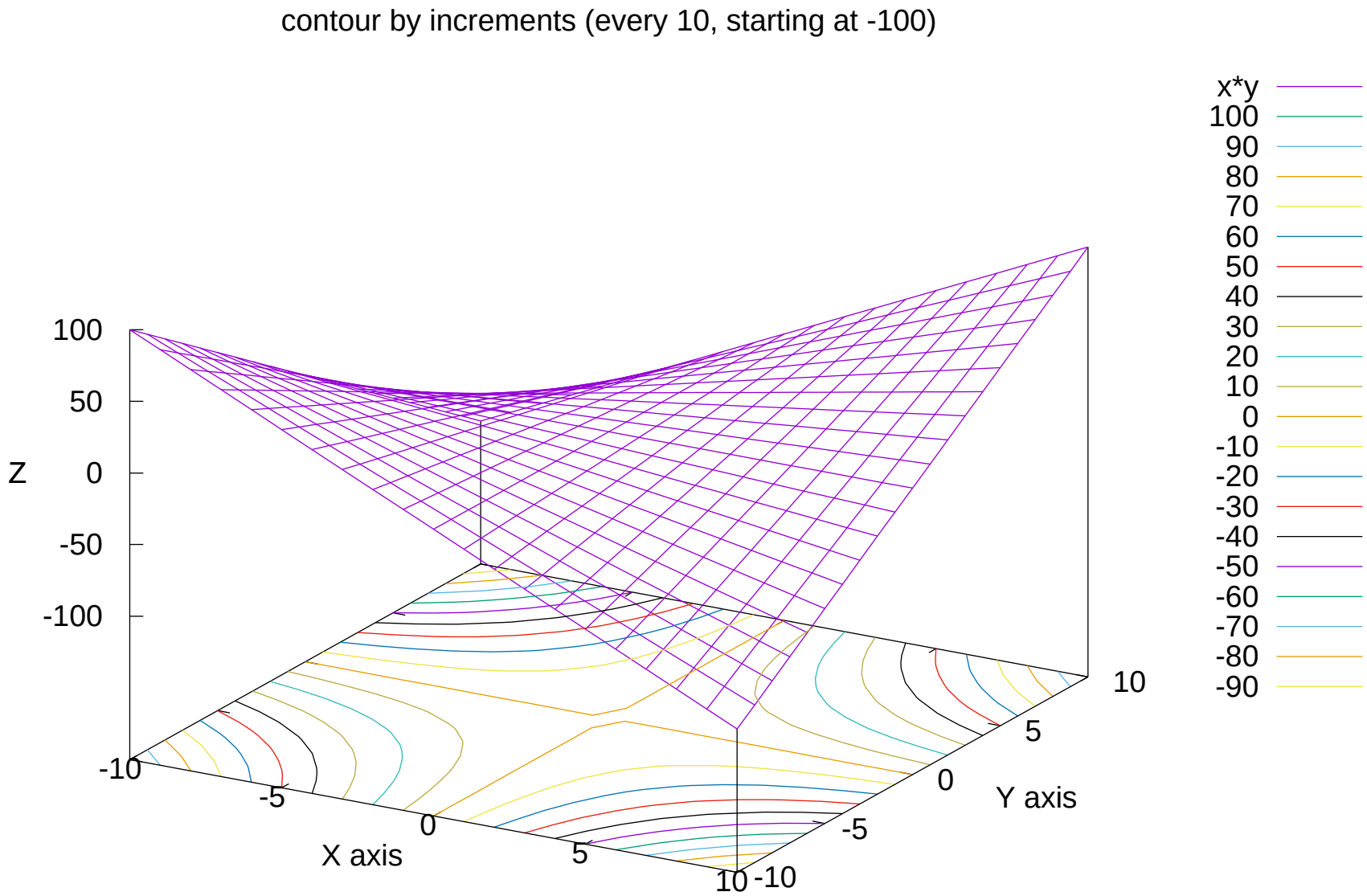


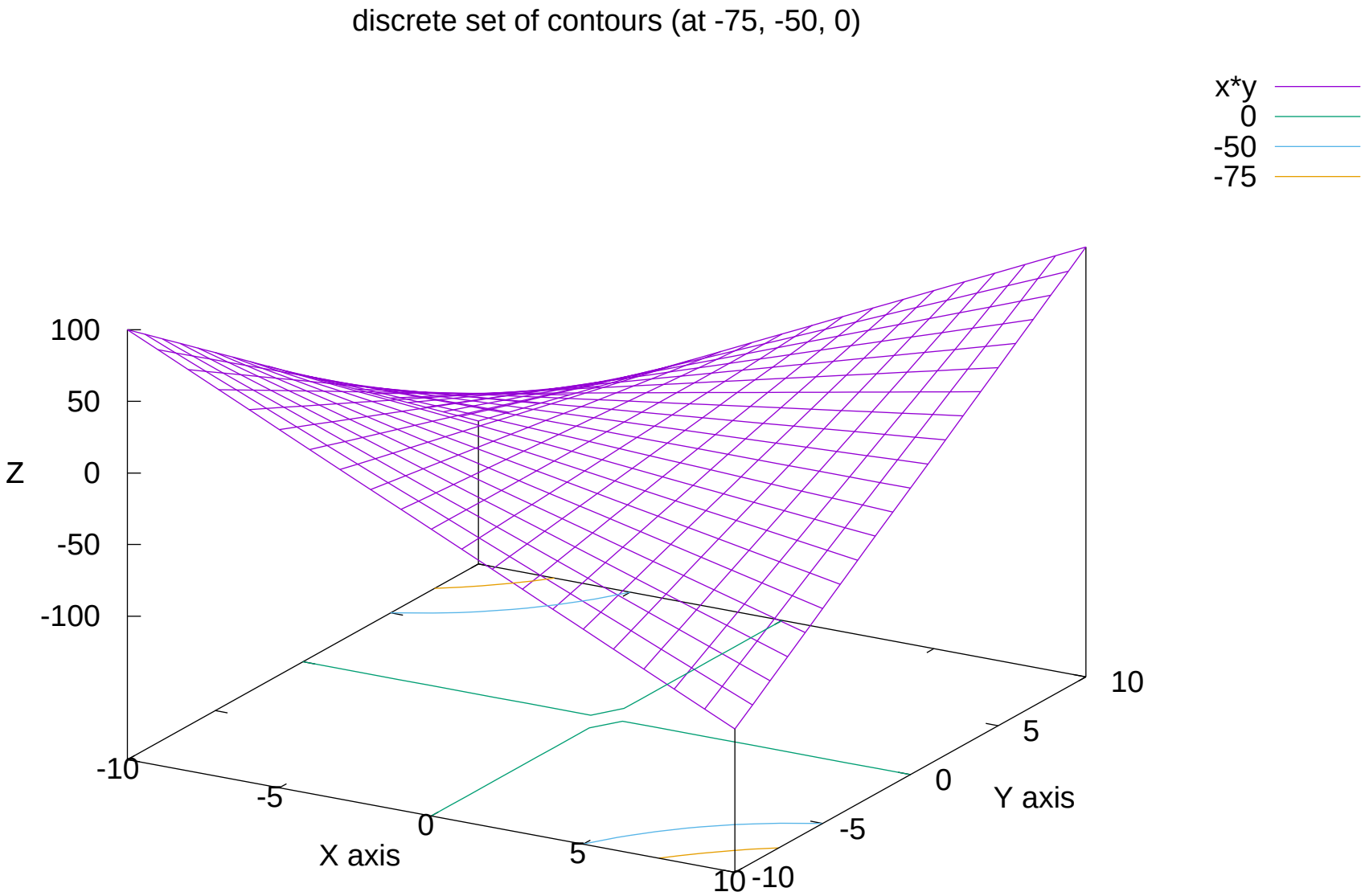
azimuth 60

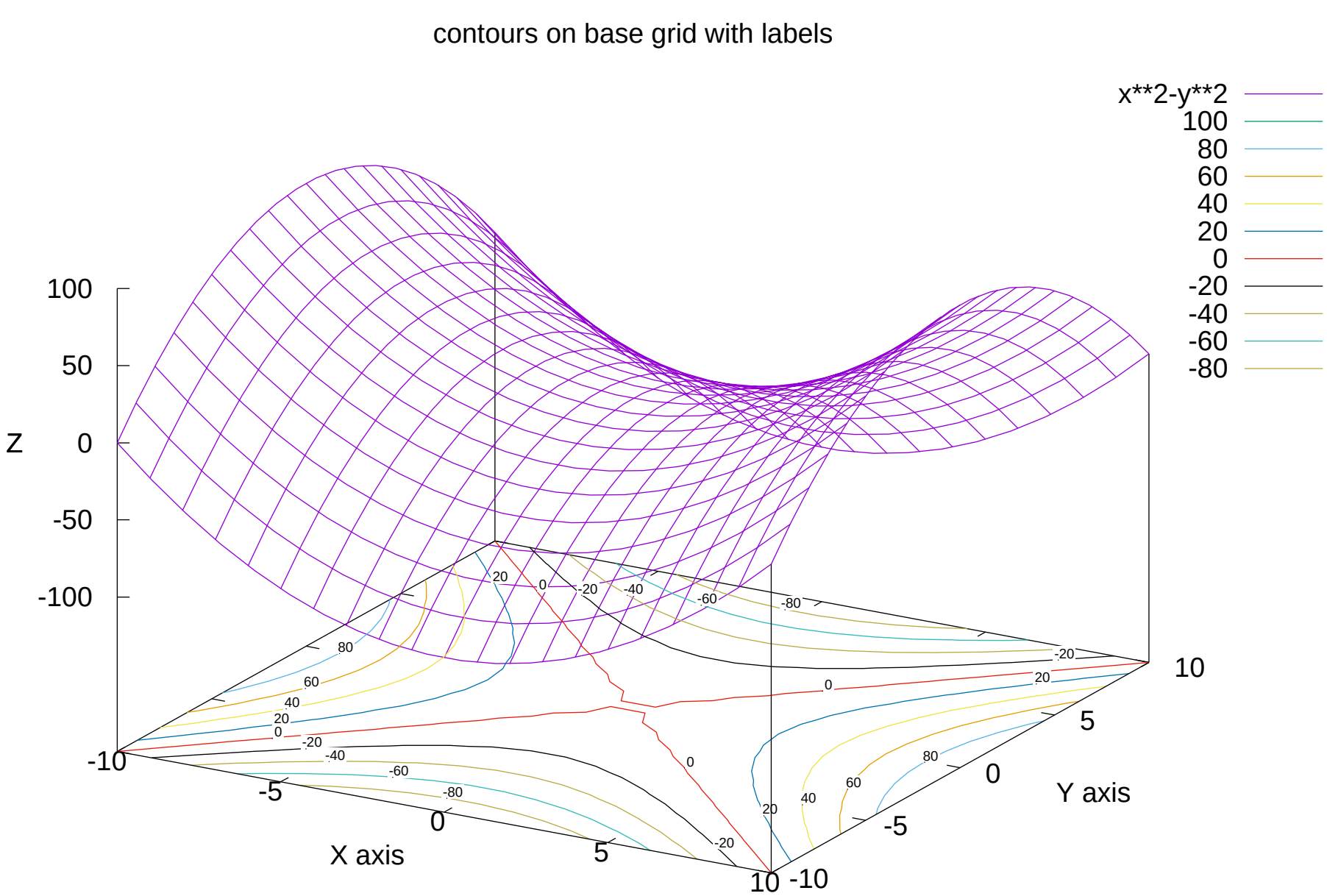


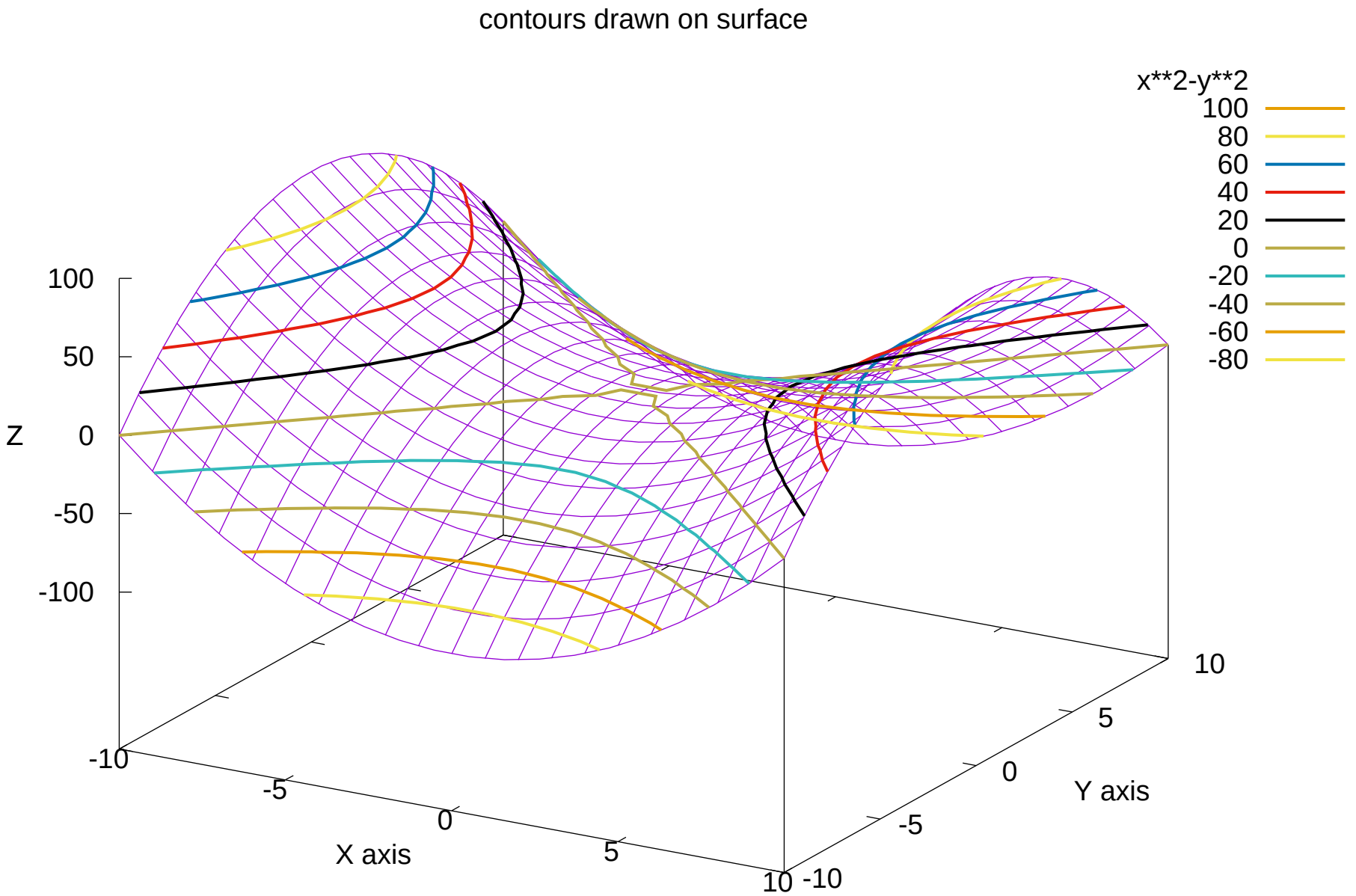

contour plot
x*y
100
50
0
-50
Z
100
50
0
-50
-100
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

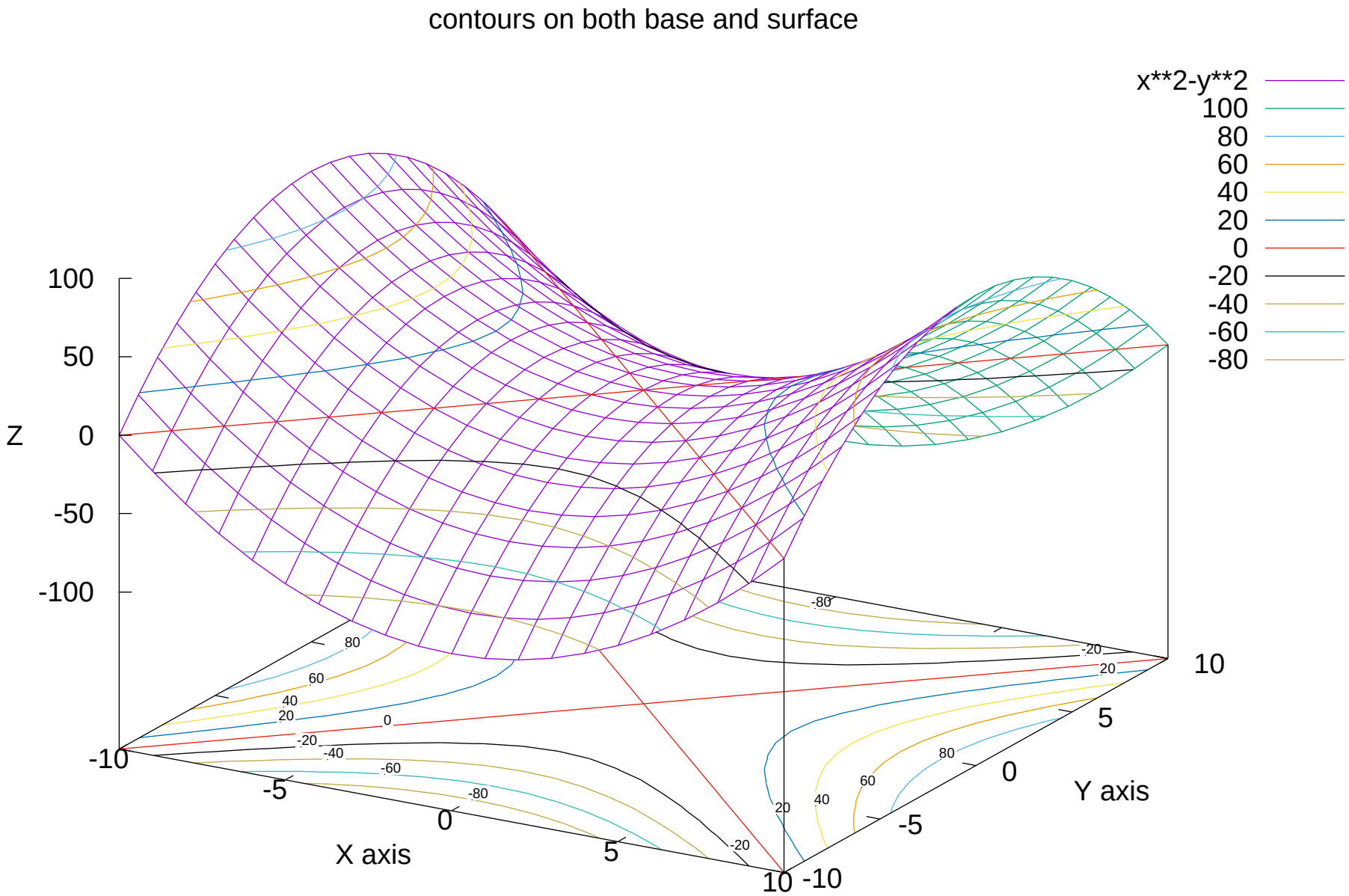

more contours (15 levels)
x*y
100
80
60
40
20
0
-20
-40
-60
-80
Z
100
50
0
-50
-100
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

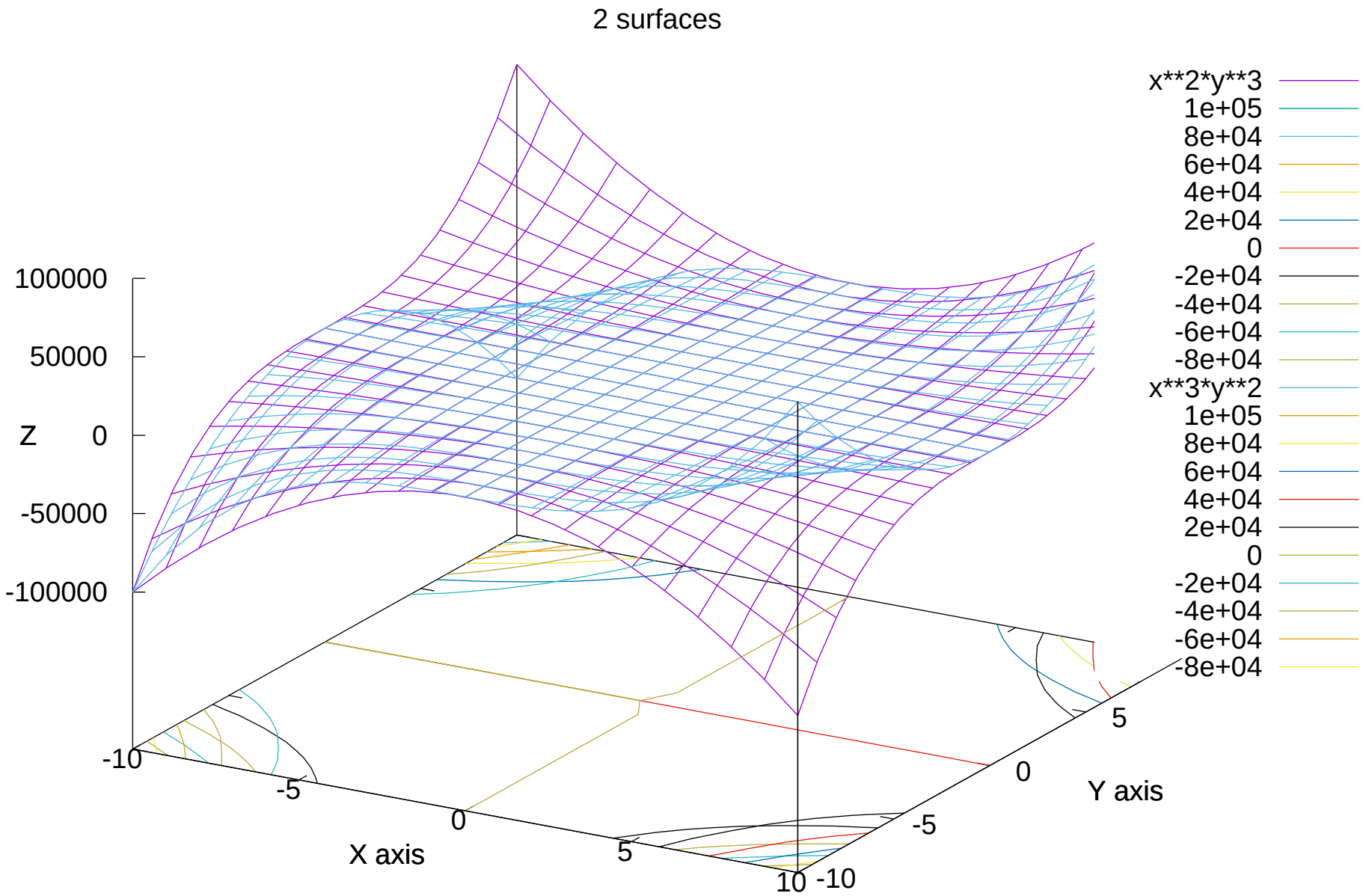

contour by increments (every 10, starting at -100)
x*y
100
90
80
70
60
50
40
30
20
10
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
Z
100
50
0
-50
-100
X axis
-10
-5
0
5
10
Y axis
-10
-5
0
5
10


discrete set of contours (at -75, -50, 0)
x*y
0
-50
-75
Z
100
50
0
-50
-100
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

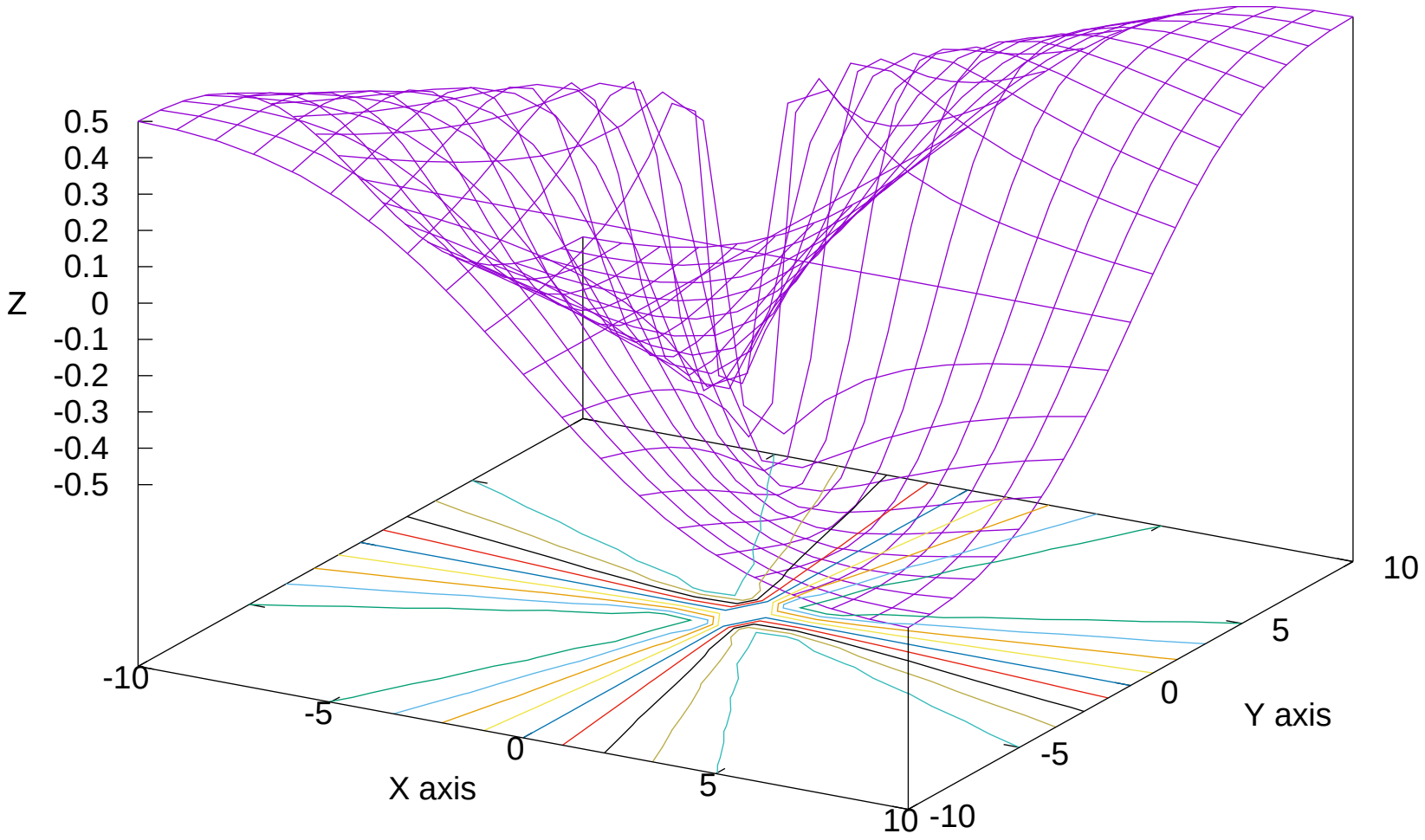

contours on base grid with labels
x**2-y**2
100
80
60
40
20
0
-20
-40
-60
-80
Z
100
50
0
-50
-100
X axis
-10
-5
0
5
10
Y axis
-10
-5
0
5
10

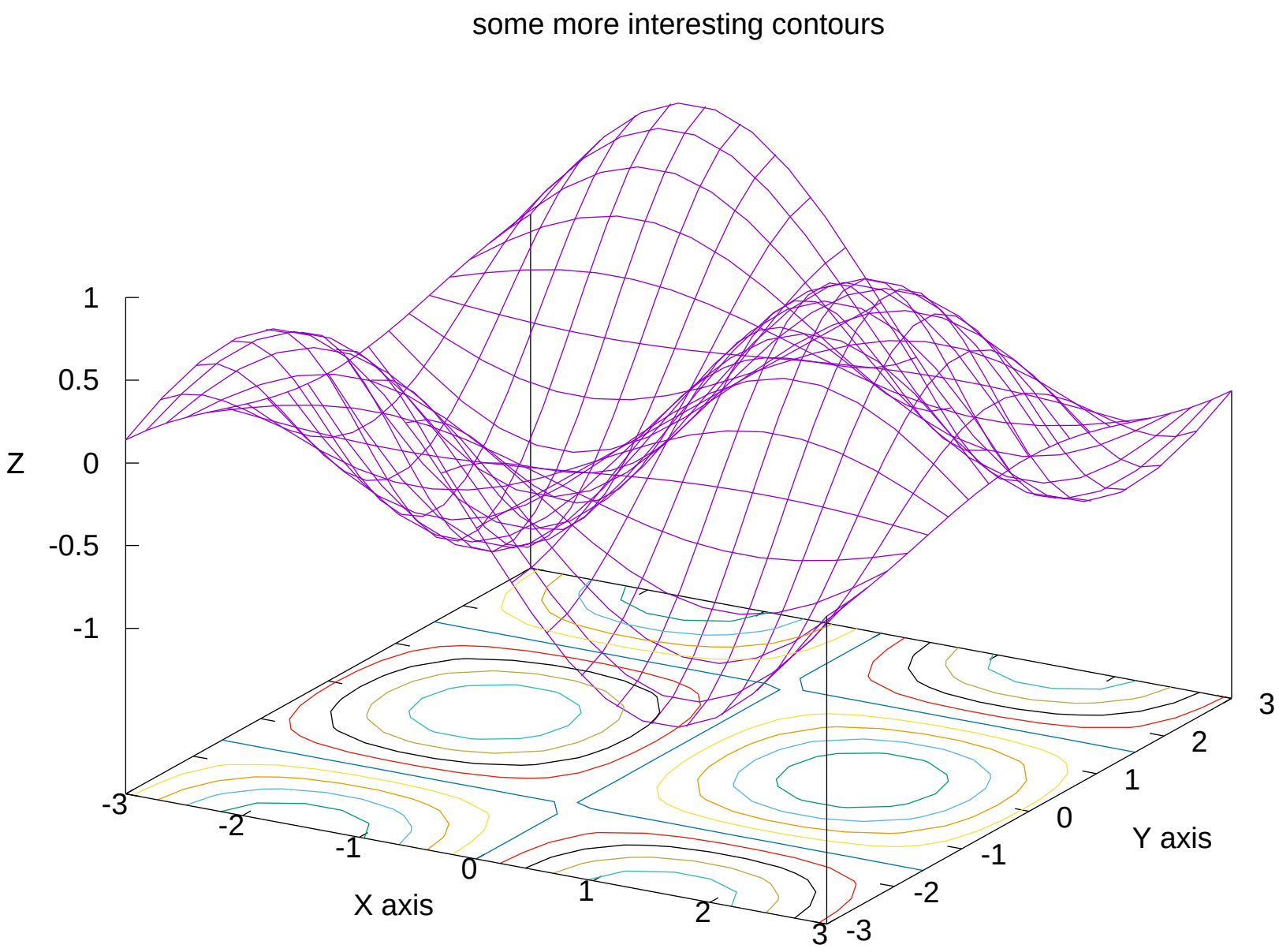

contours drawn on surface
x**2-y**2
100
80
60
40
20
0
-20
-40
-60
-80
Z
100
50
0
-50
-100
X axis
-10
-5
0
5
10
Y axis
-10
-5
0
5
10

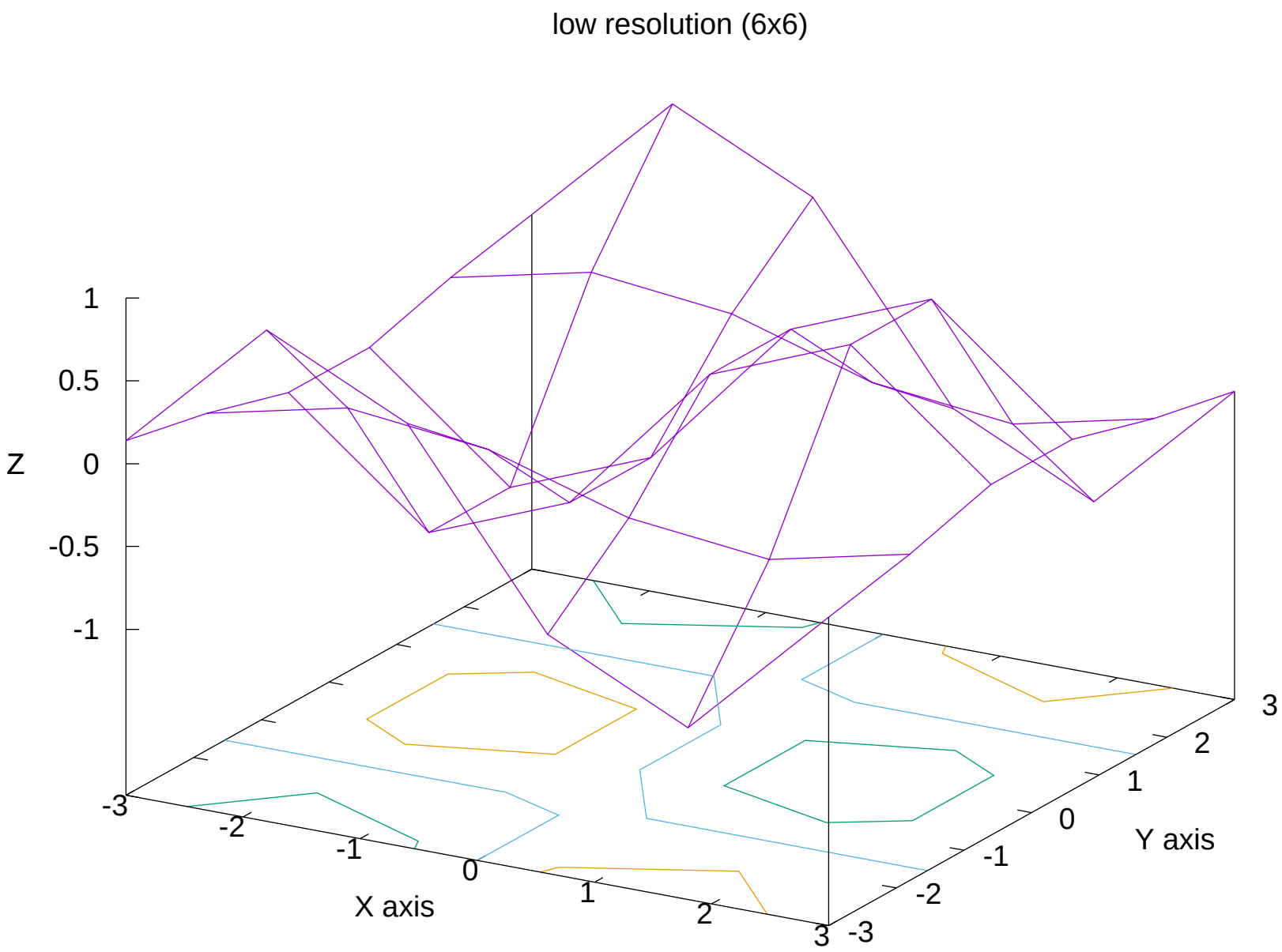

contours on both base and surface
x**2-y**2
100
80
60
40
20
0
-20
-40
-60
-80
Z
100
50
0
-50
-100
X axis
-10
-5
0
5
10
Y axis
-10
-5
0
5
10

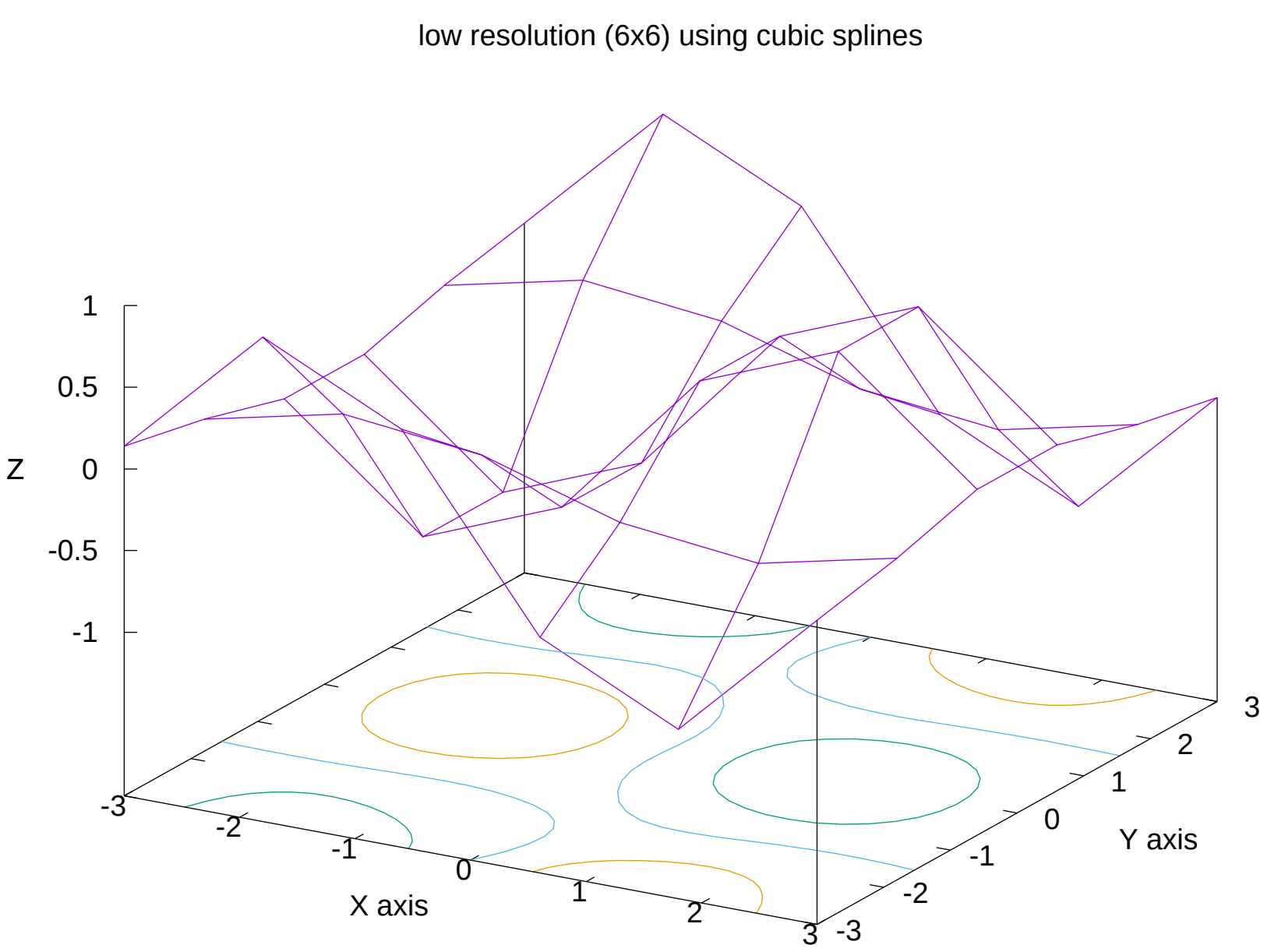

2 surfaces
100000
50000
Z 0
-50000
-100000
-10
-5
0
5
10
X axis
-10
-5
0
5
Y axis
x**2*y**3
1e+05
8e+04
6e+04
4e+04
2e+04
0
-2e+04
-4e+04
-6e+04
-8e+04
x**3*y**2
1e+05
8e+04
6e+04
4e+04
2e+04
0
-2e+04
-4e+04
-6e+04
-8e+04

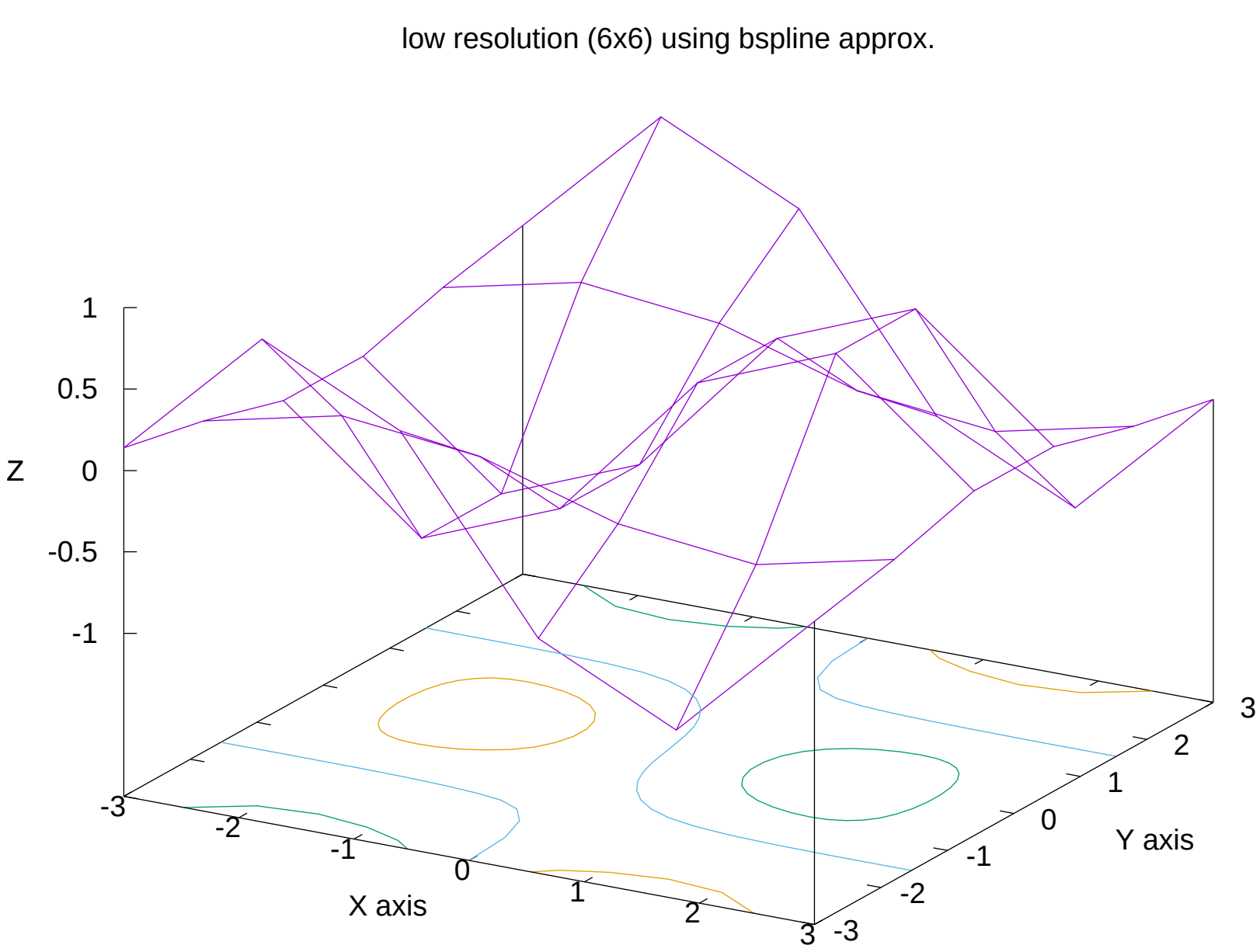
some more interesting contours



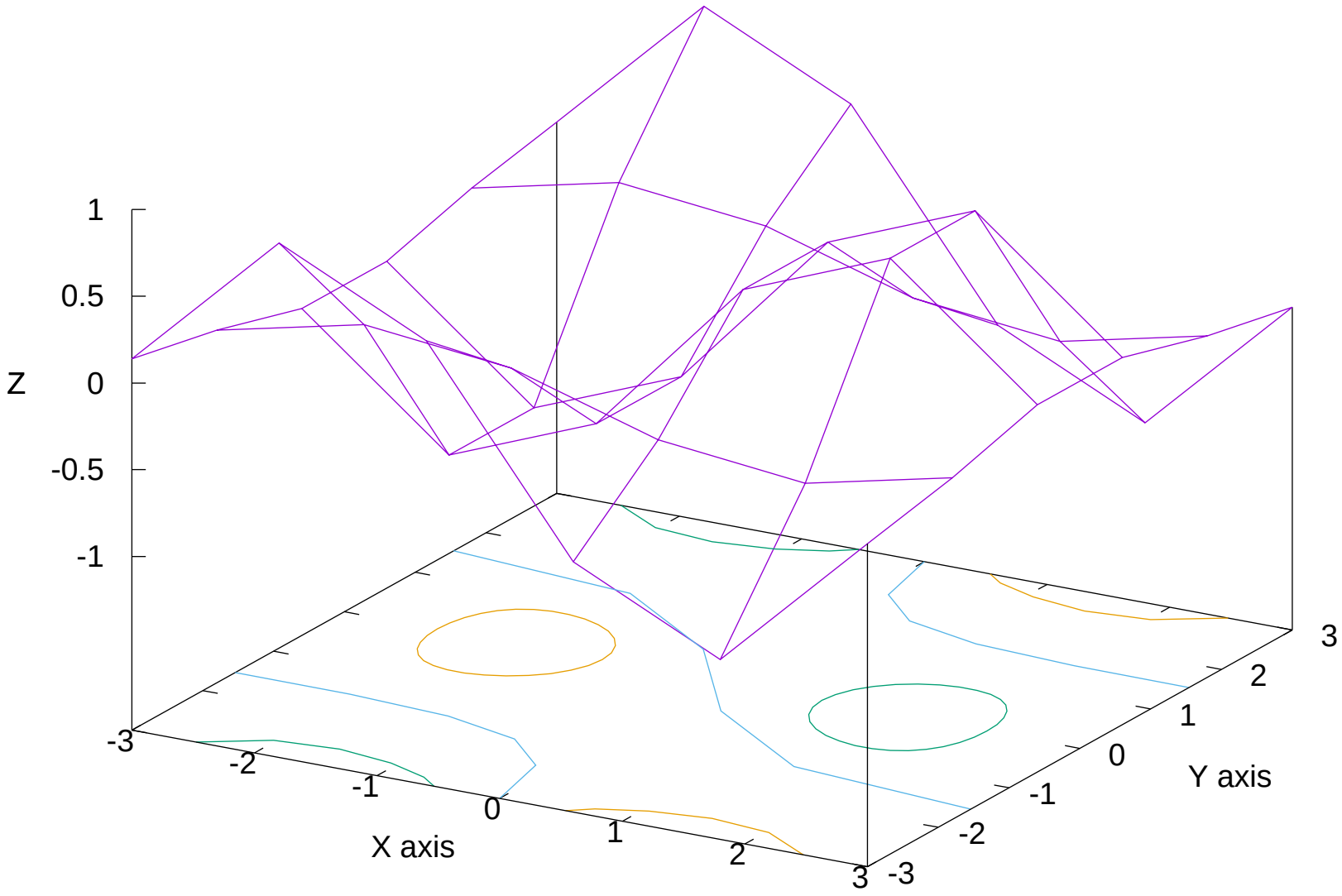

some more interesting contours
Z
1
0.5
0
-0.5
-1
X axis
-3
-2
-1
0
1
2
3
Y axis
-3
-2
-1
0
1
2
3

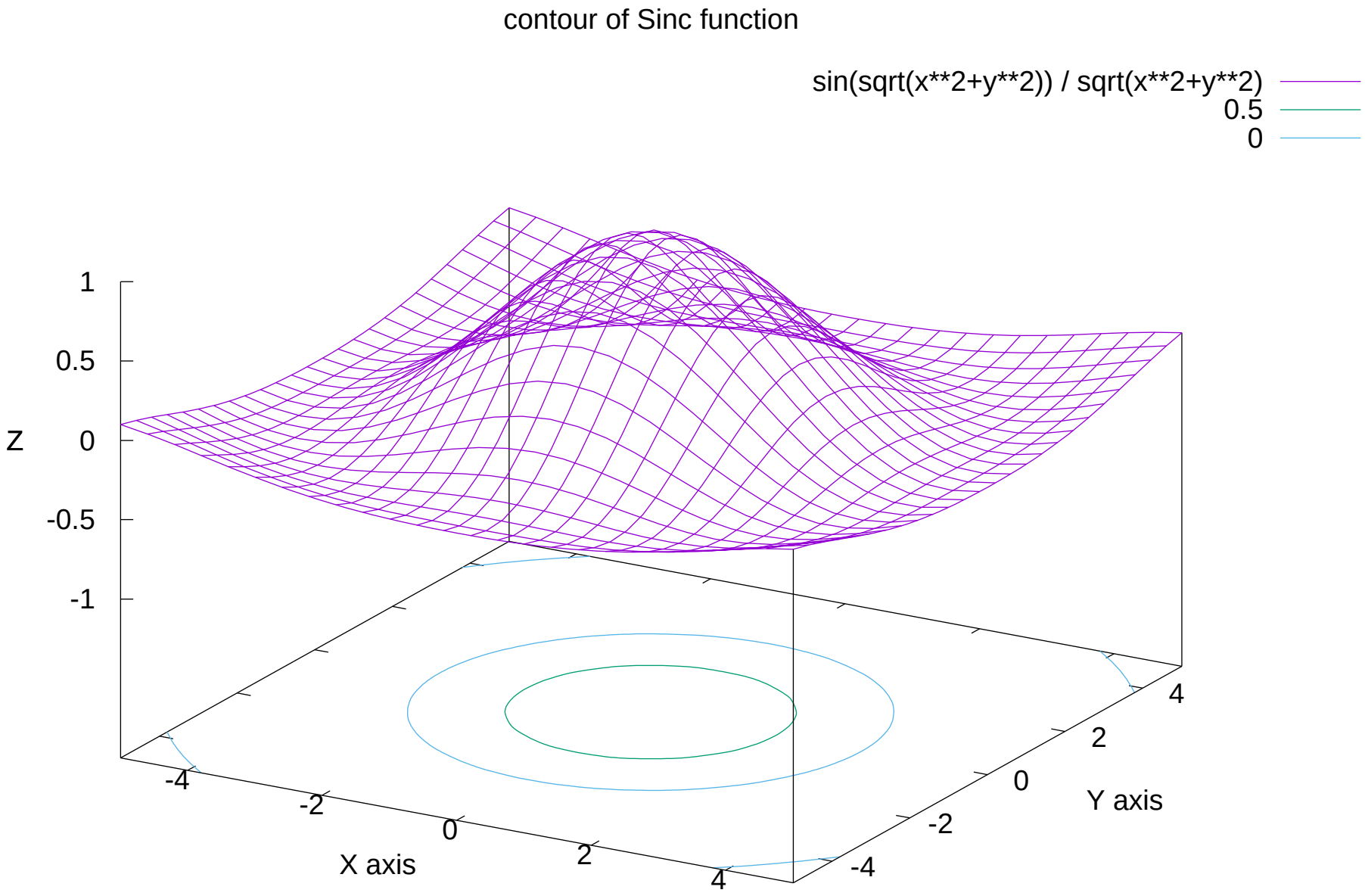

low resolution (6x6)
Z
1
0.5
0
-0.5
-1
X axis
-3
-2
-1
0
1
2
3
Y axis
-3
-2
-1
0
1
2
3

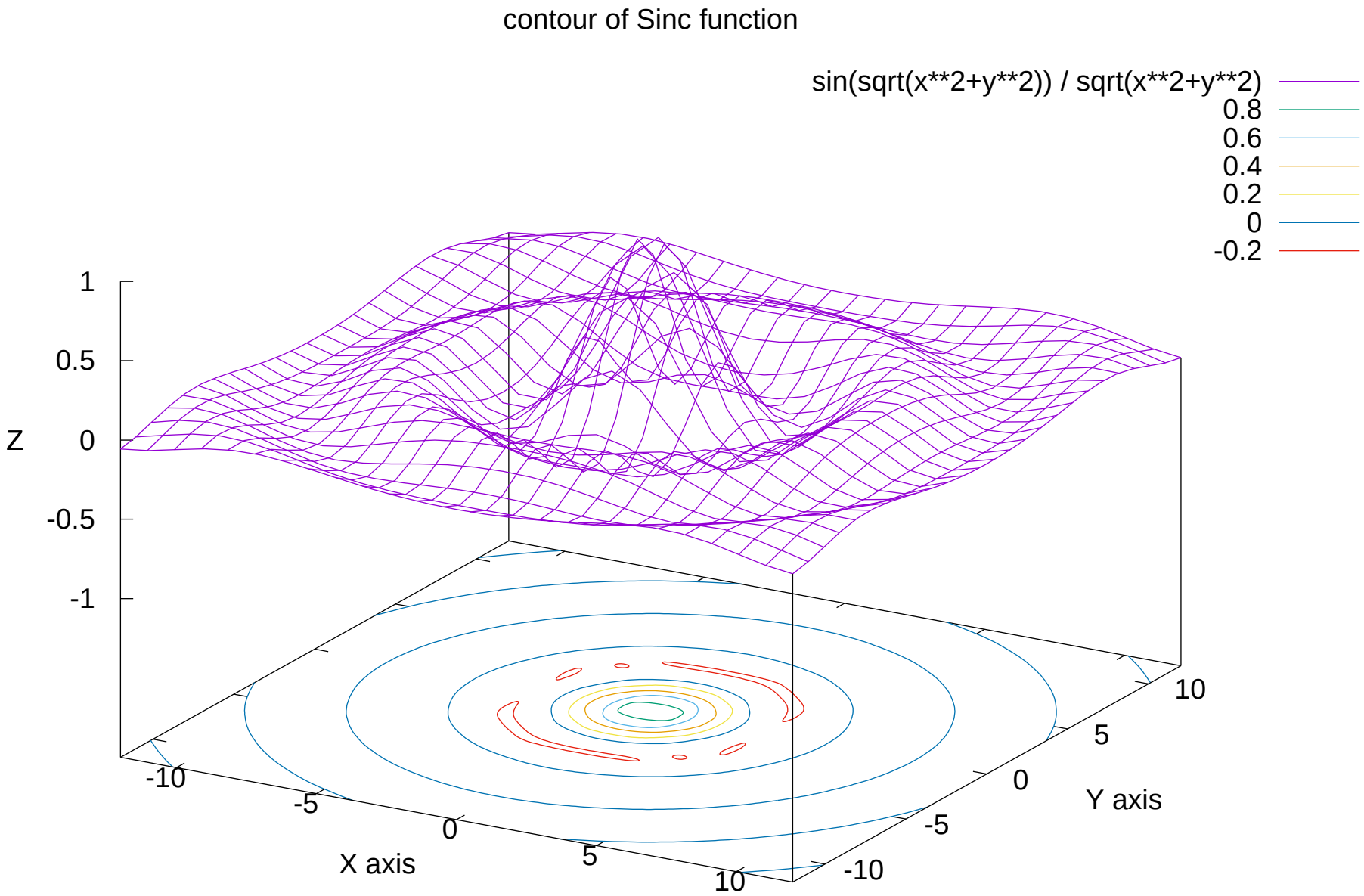

low resolution (6x6) using cubic splines
Z
1
0.5
0
-0.5
-1
X axis
-3
-2
-1
0
1
2
3
Y axis
-3
-2
-1
0
1
2
3

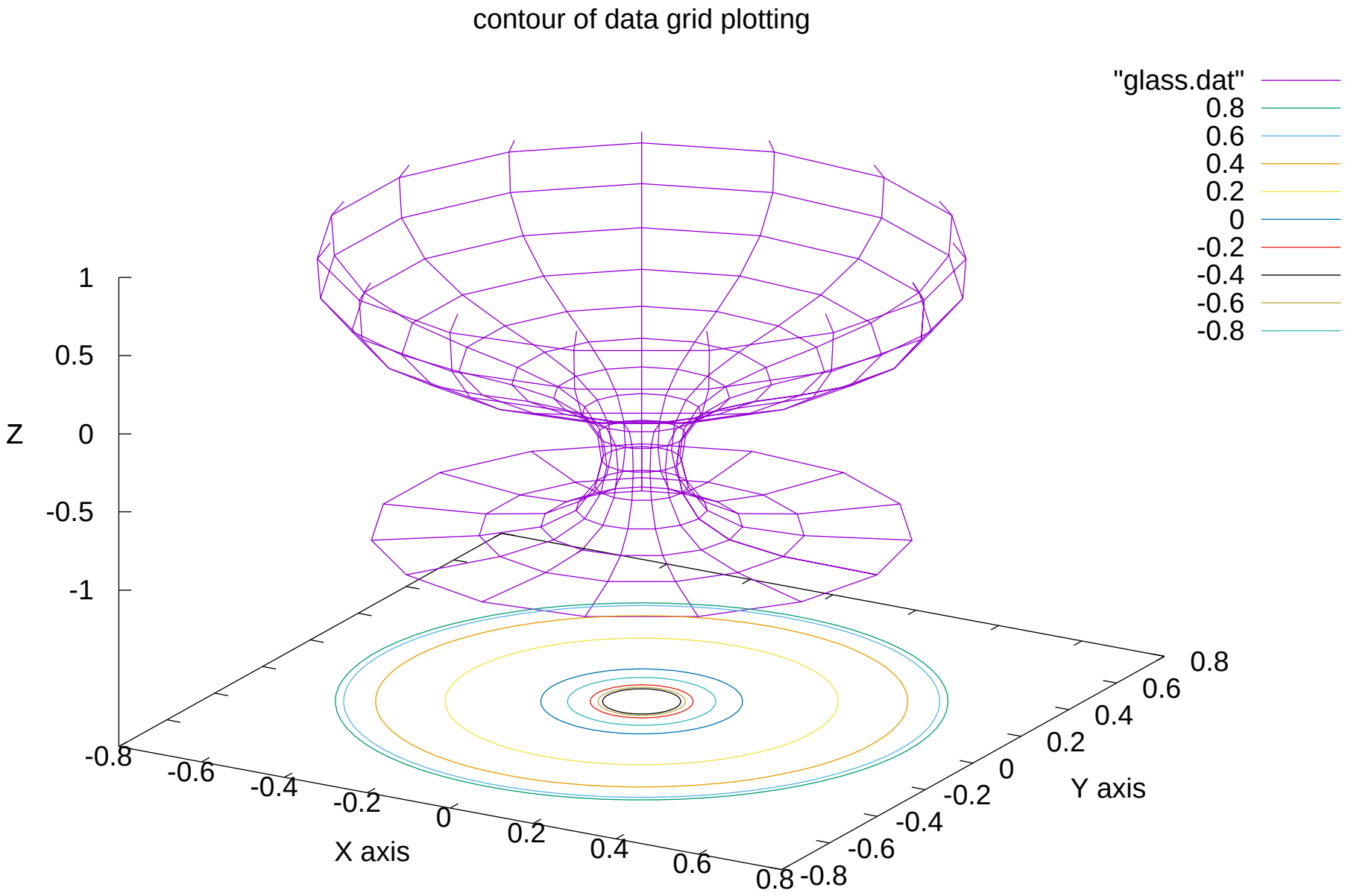

low resolution (6x6) using bspline approx.
Z
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3
X axis
-3
-2
-1
0
1
2
3
Y axis

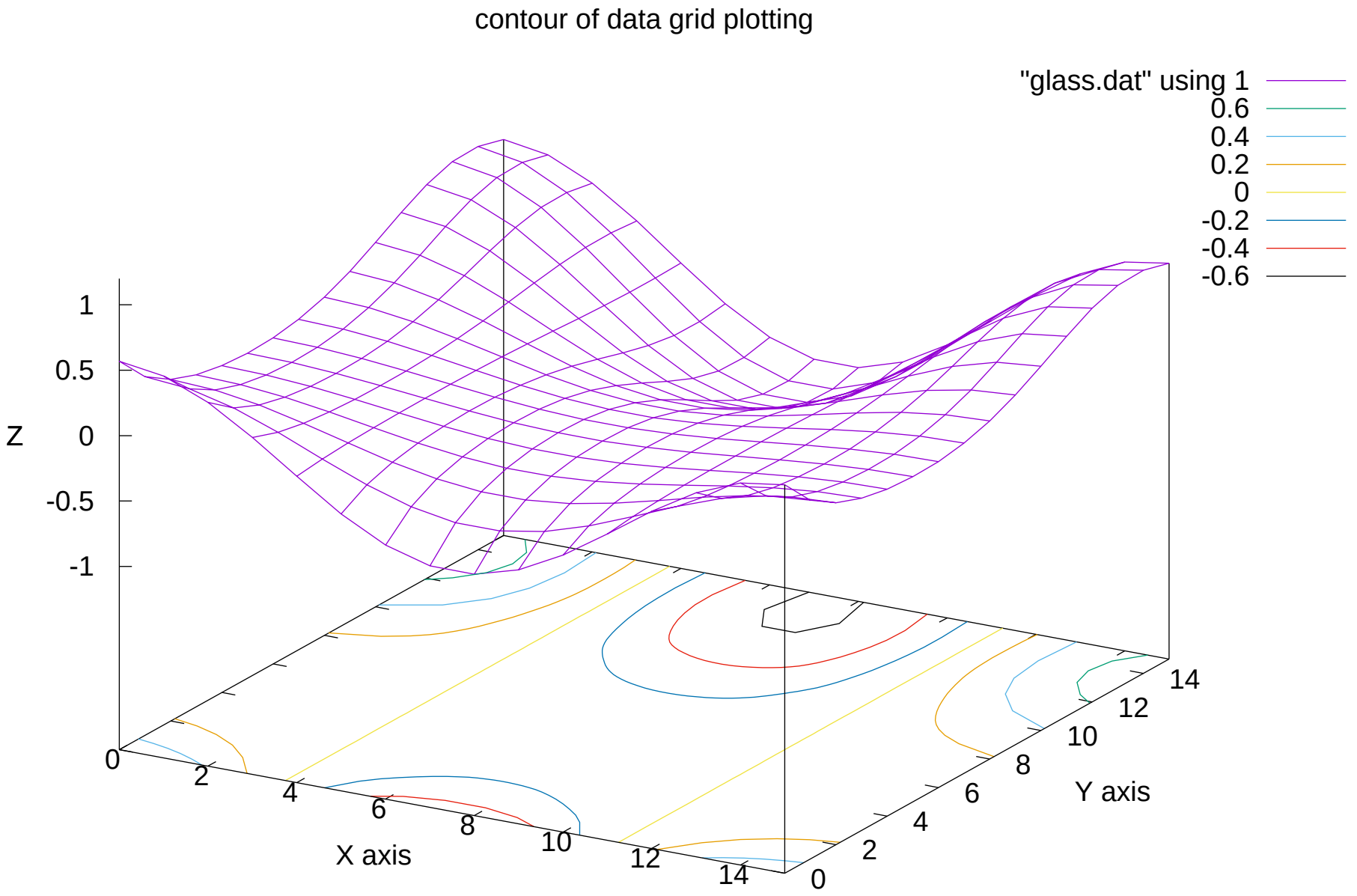
low resolution (6x6) raise bspline order.

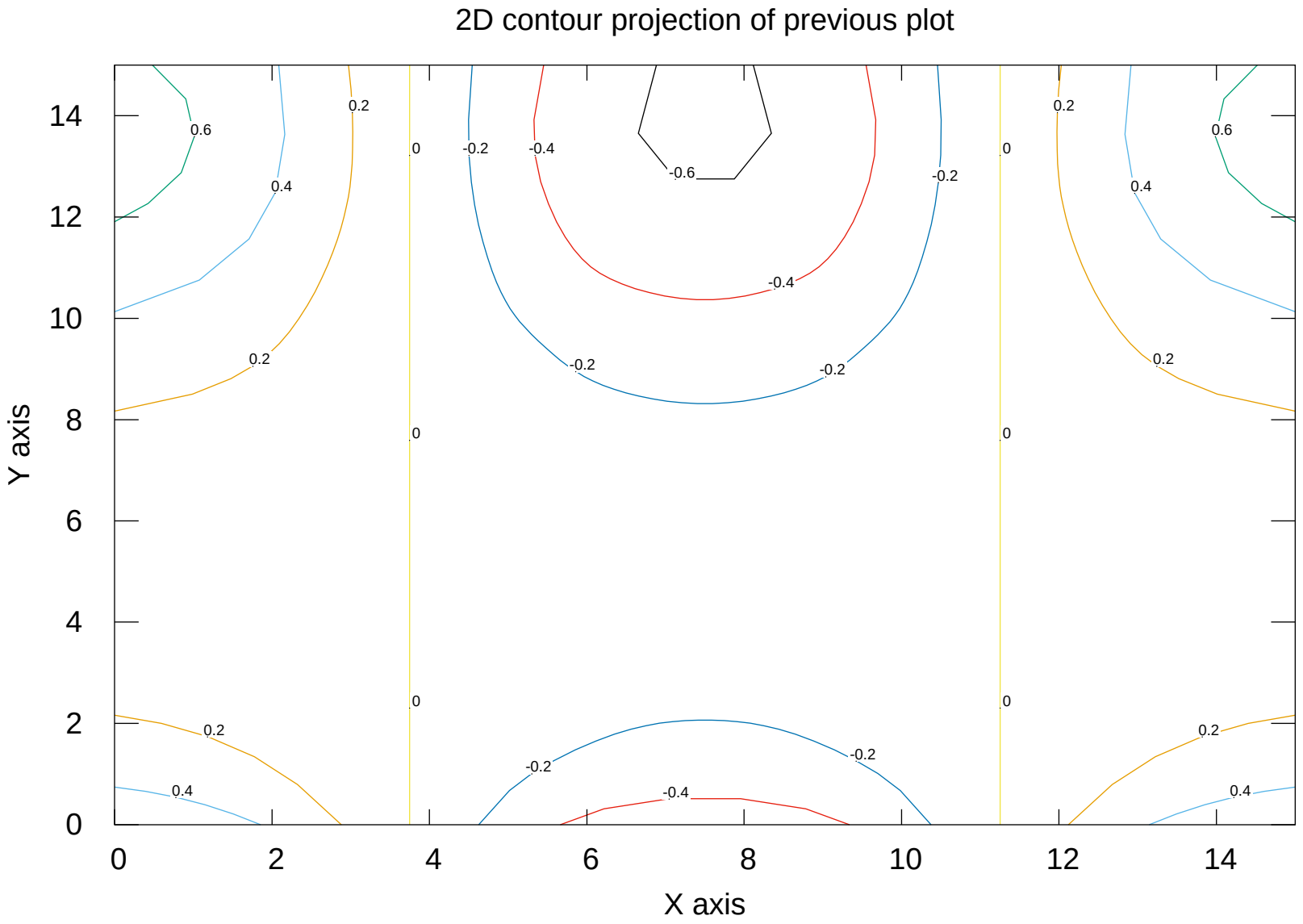


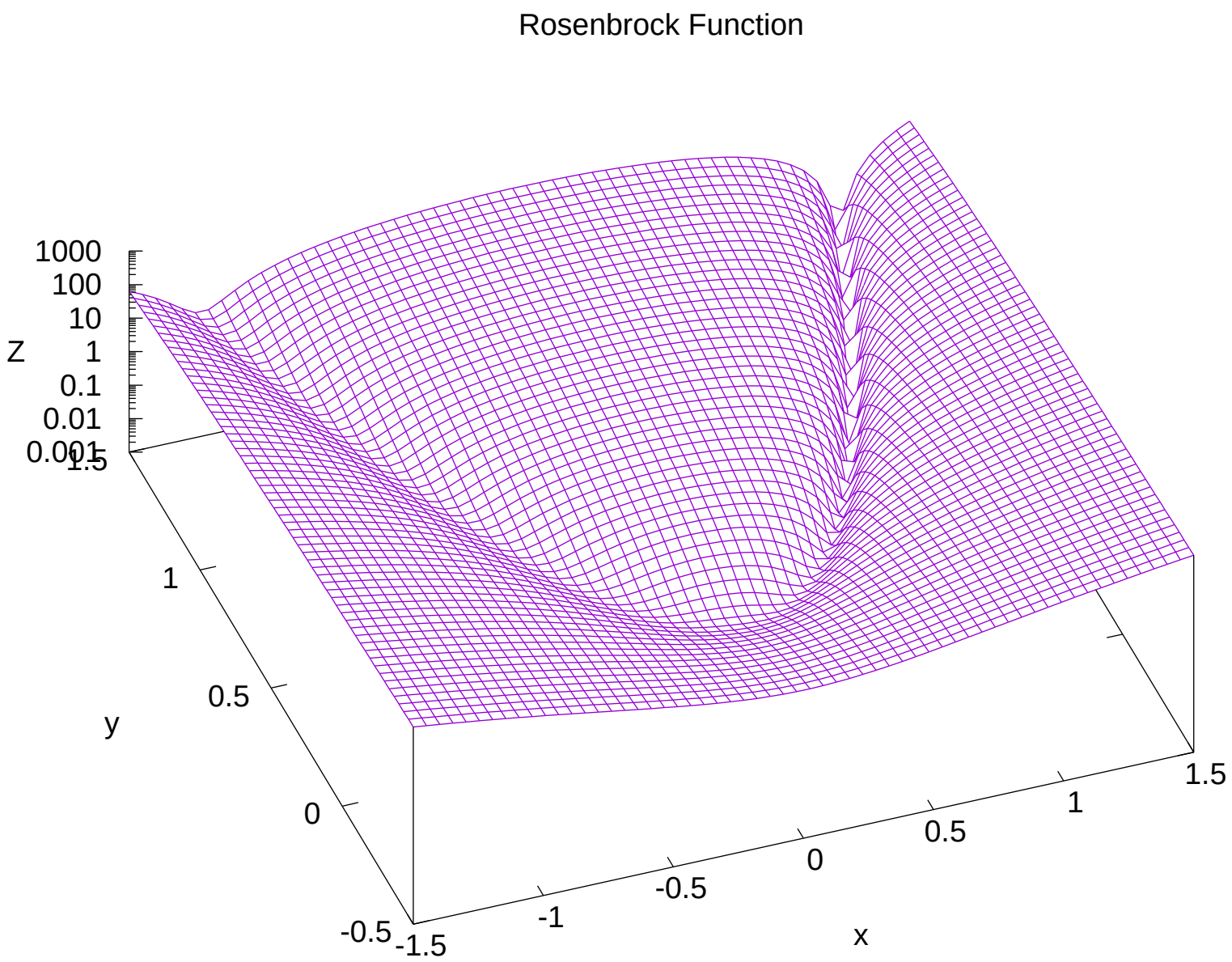

contour of Sinc function
sin(sqrt(x**2+y**2)) / sqrt(x**2+y**2)
0.5
0
Z
1
0.5
0
-0.5
-1
X axis
-4
-2
0
2
4
Y axis
-4
-2
0
2
4


contour of Sinc function
sin(sqrt(x**2+y**2)) / sqrt(x**2+y**2)
0.8
0.6
0.4
0.2
0
-0.2
Z
1
0.5
0
-0.5
-1
X axis
-10
-5
0
5
10
Y axis
-10
-5
0
5
10

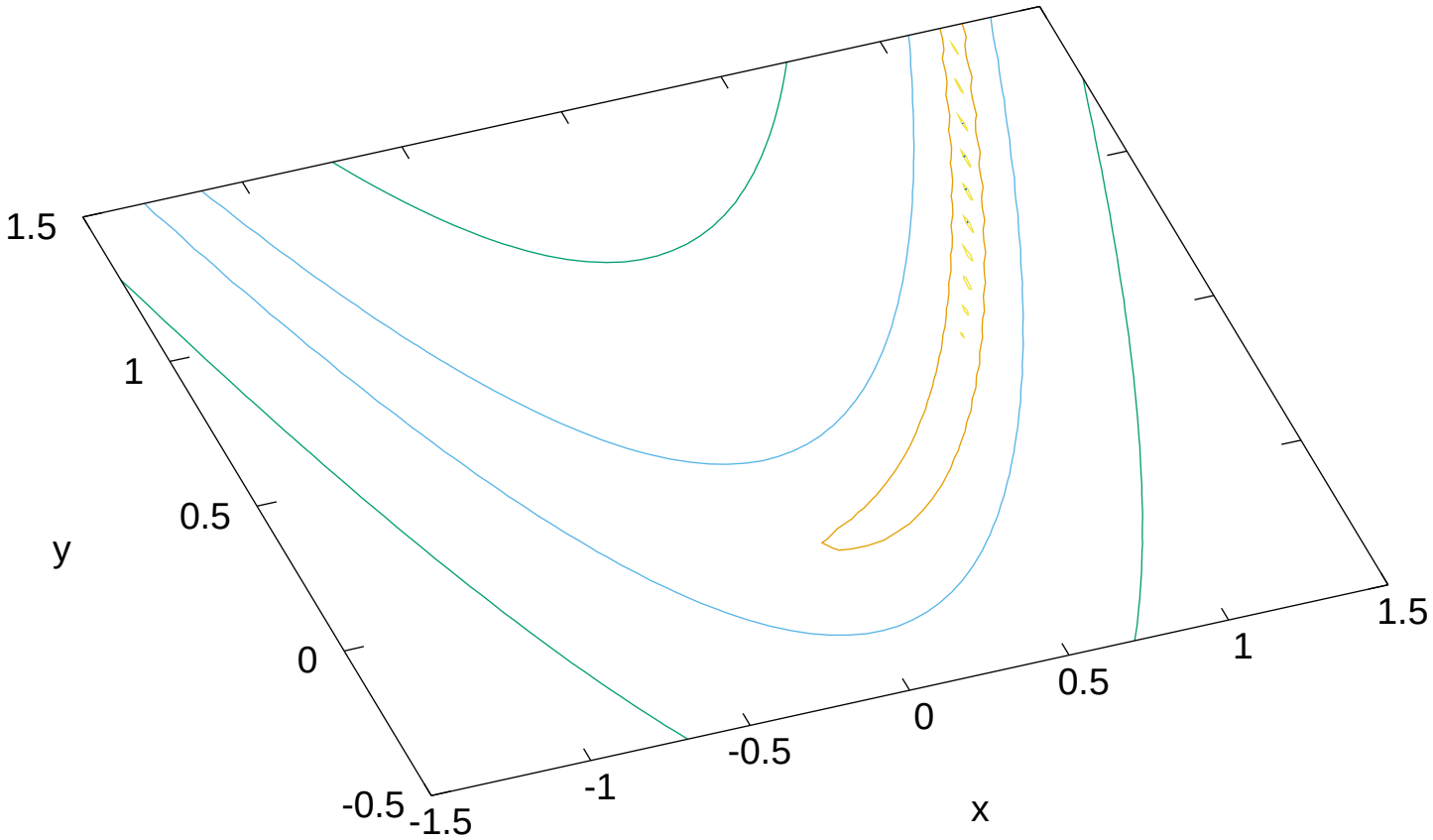

contour of data grid plotting
"glass.dat"
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
Z
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
X axis
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Y axis


contour of data grid plotting
"glass.dat" using 1
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
Z
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
X axis
0
2
4
6
8
10
12
14
Y axis

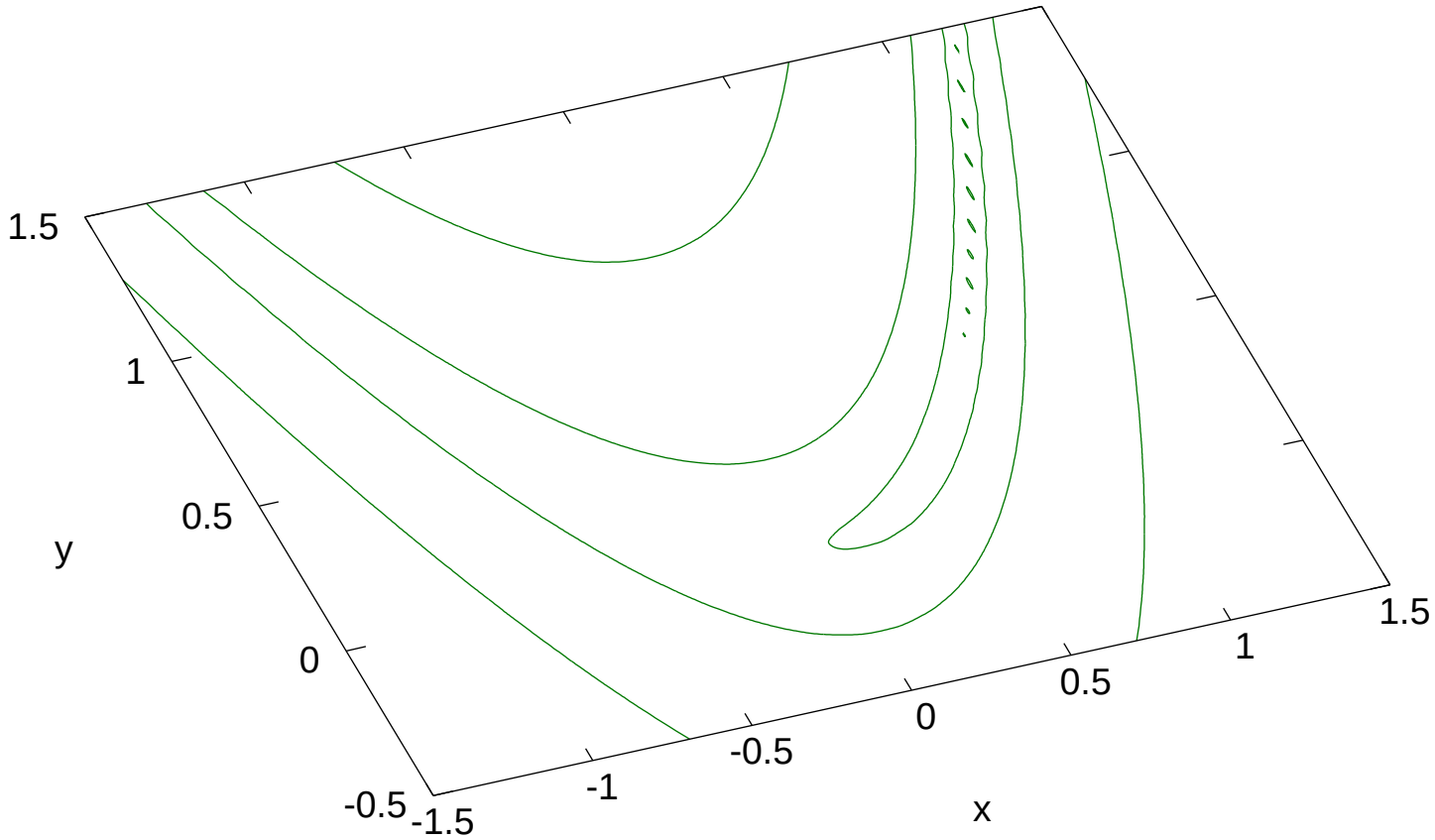

2D contour projection of previous plot
Y axis
X axis
0
2
4
6
8
10
12
14
0.6
0.4
0.2
0
-0.2
-0.4
-0.6

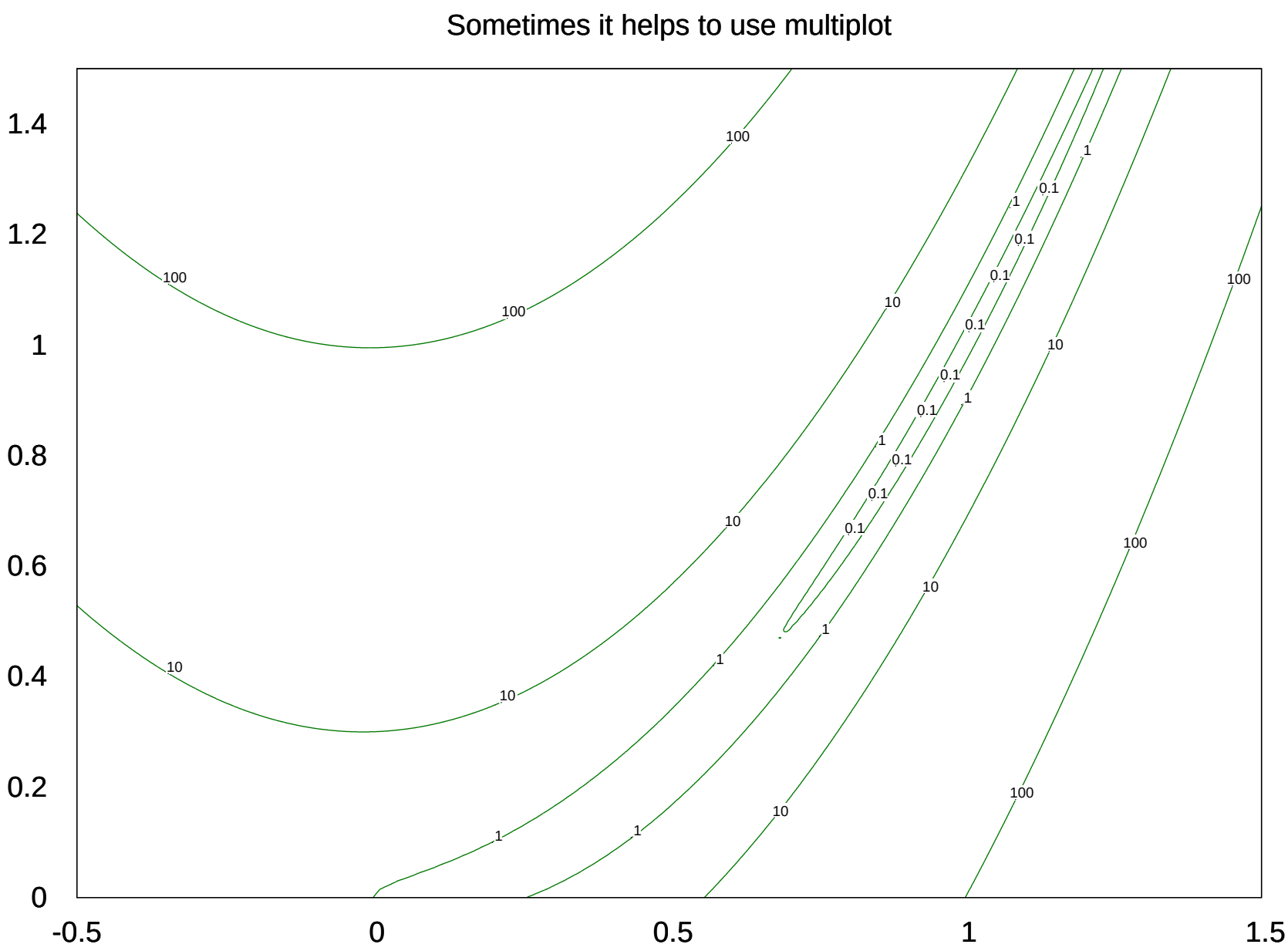

Rosenbrock Function
1000
100
10
Z
1
0.1
0.01
0.001
1.5
1
0.5
y
0
-0.5
-1.5
-1
-0.5
0
0.5
1
1.5
x

Rosenbrock Function

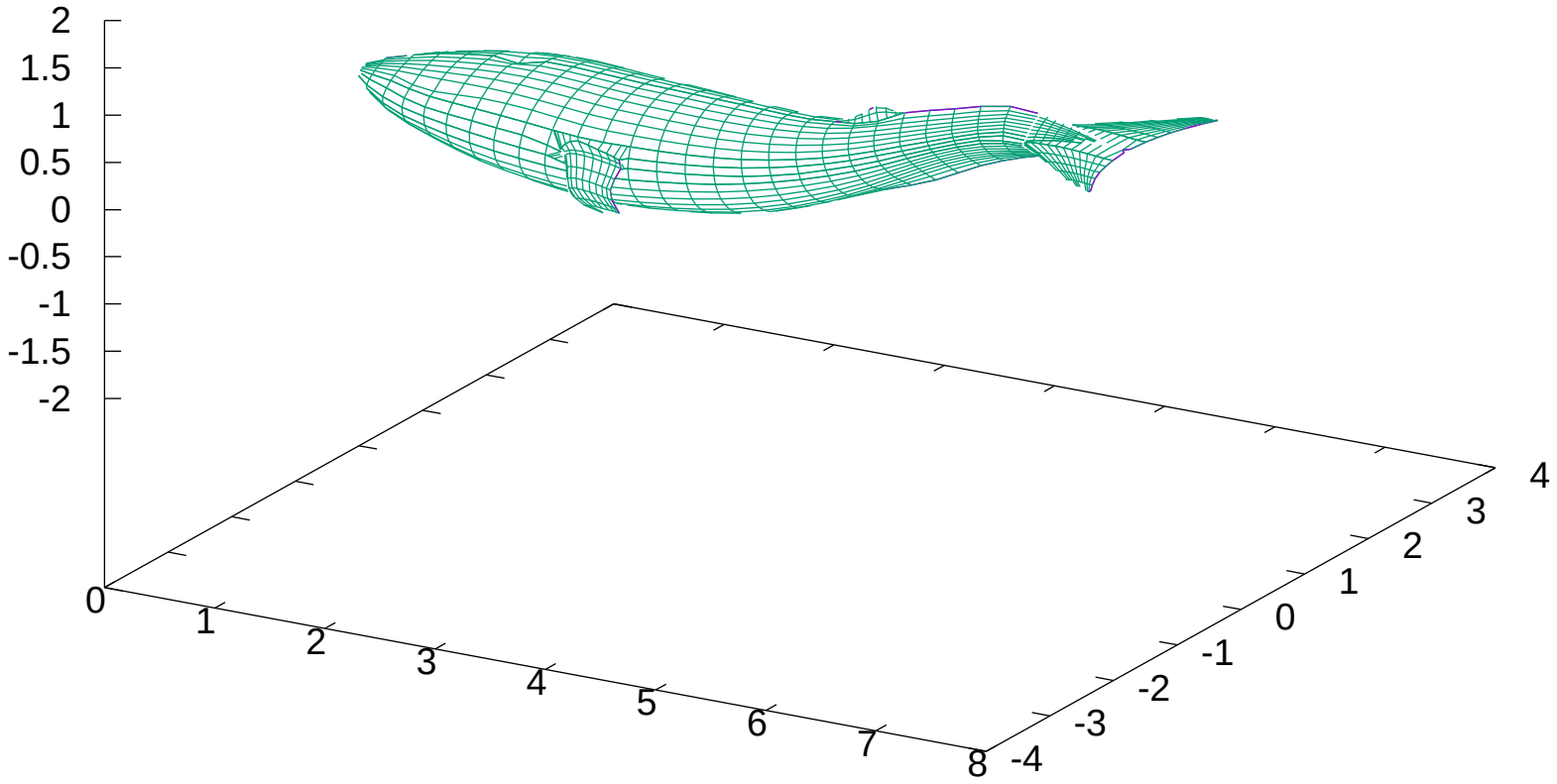


All contours drawn in a single color

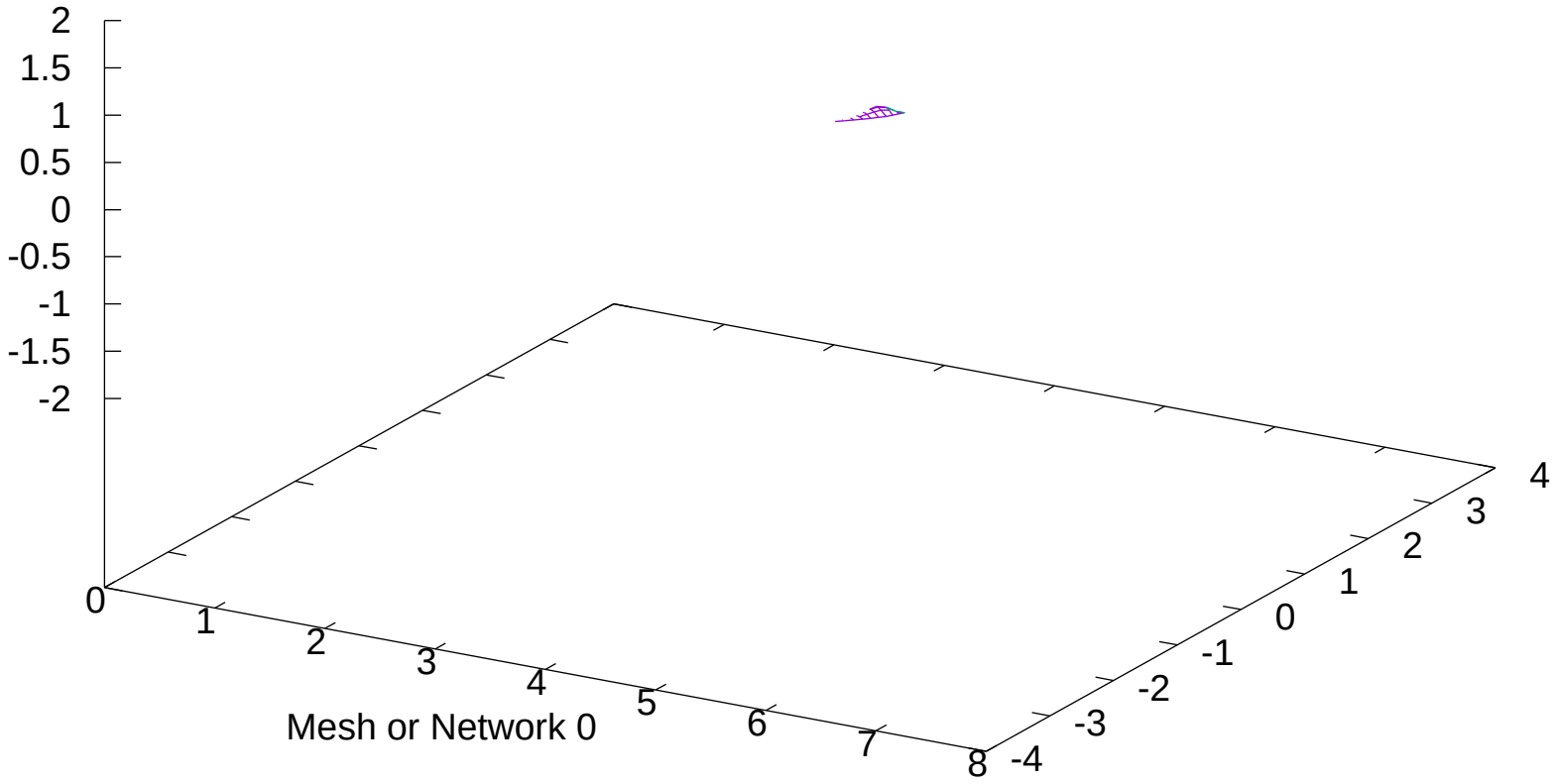



Sometimes it helps to use multiplot
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.5
0
0.5
1
1.5
100
10
1
0.1

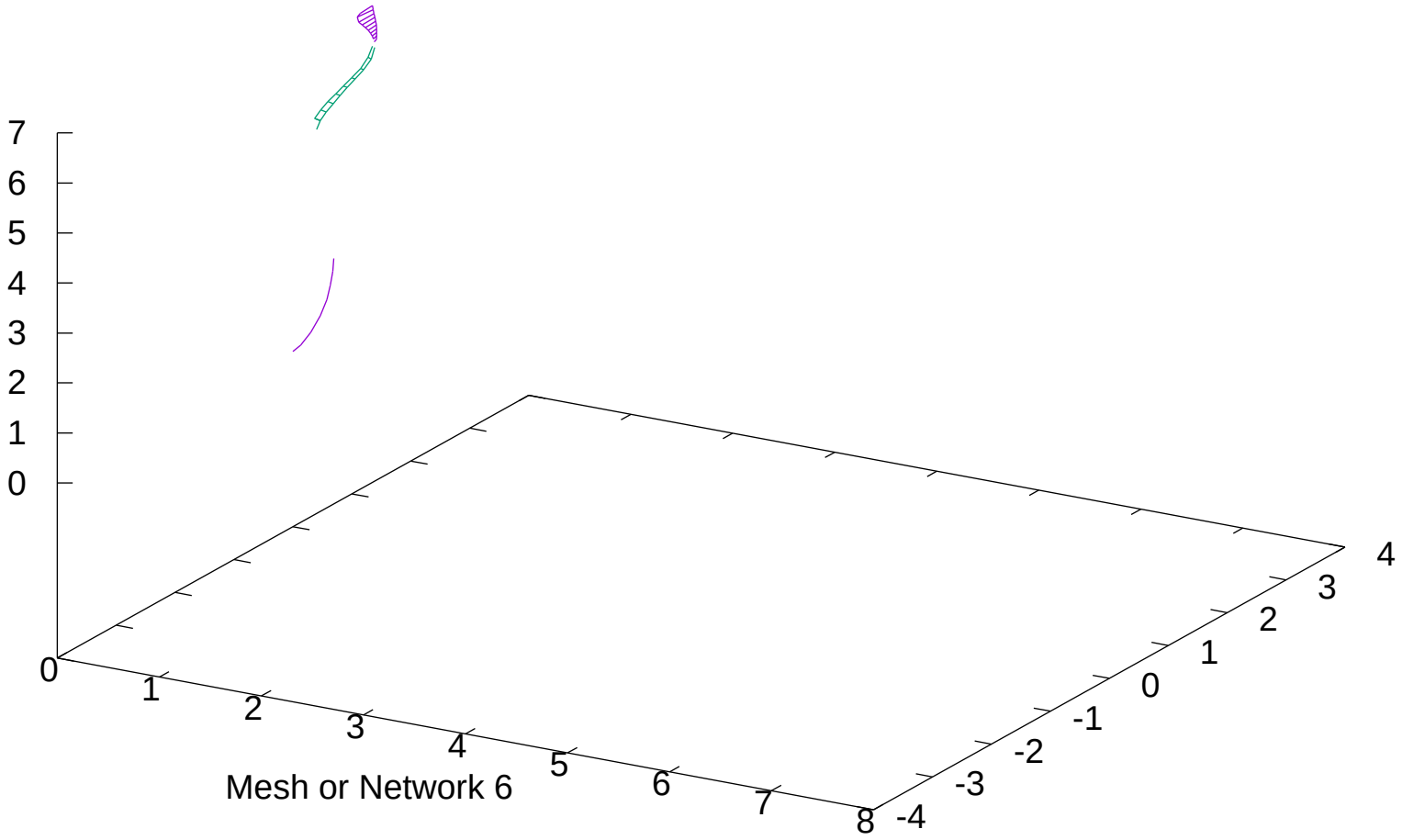
Demo of multiple mesh per file capability - Digitized Blue Whale



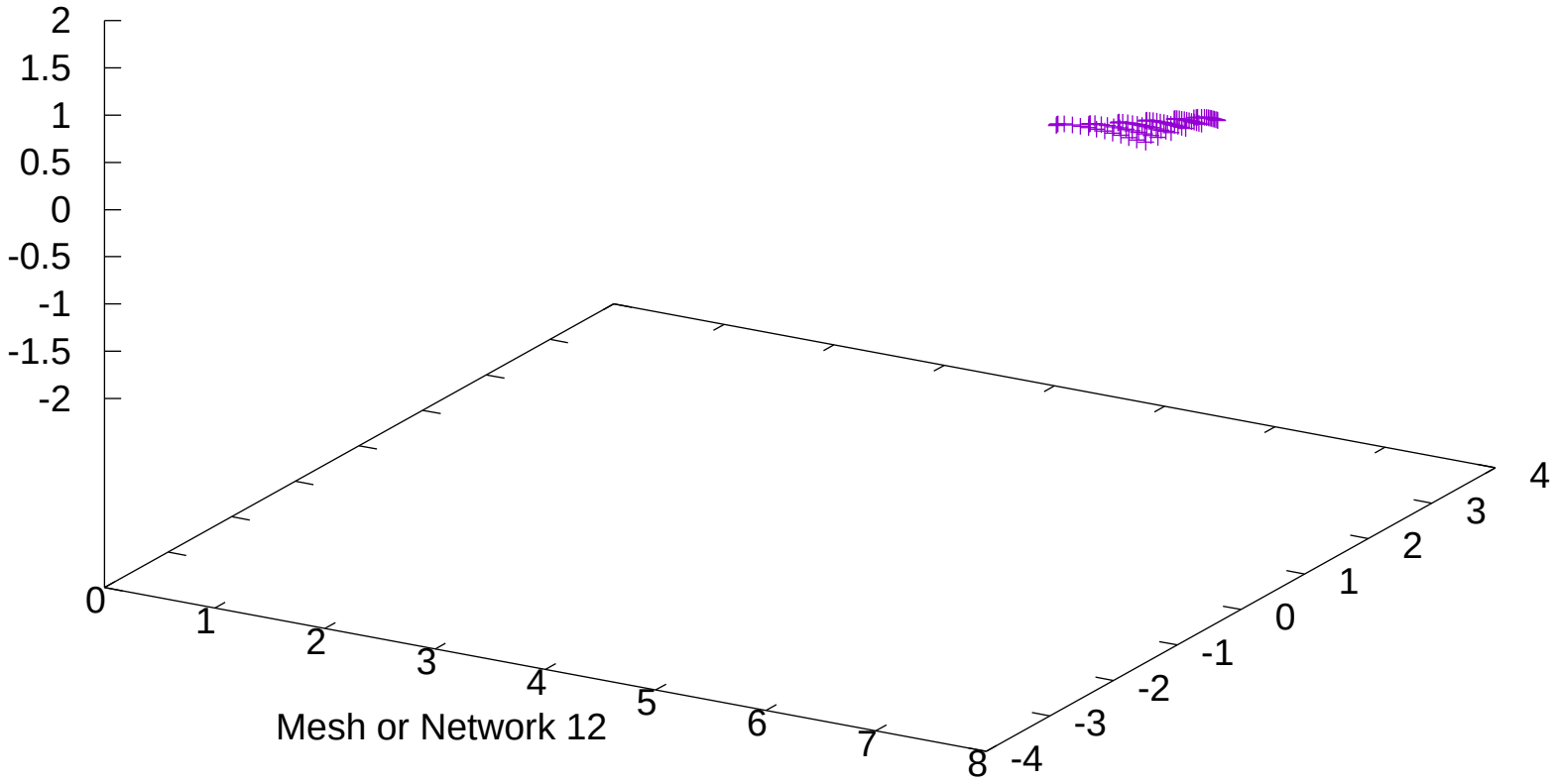
Demo of multiple mesh per file capability - Digitized Blue Whale



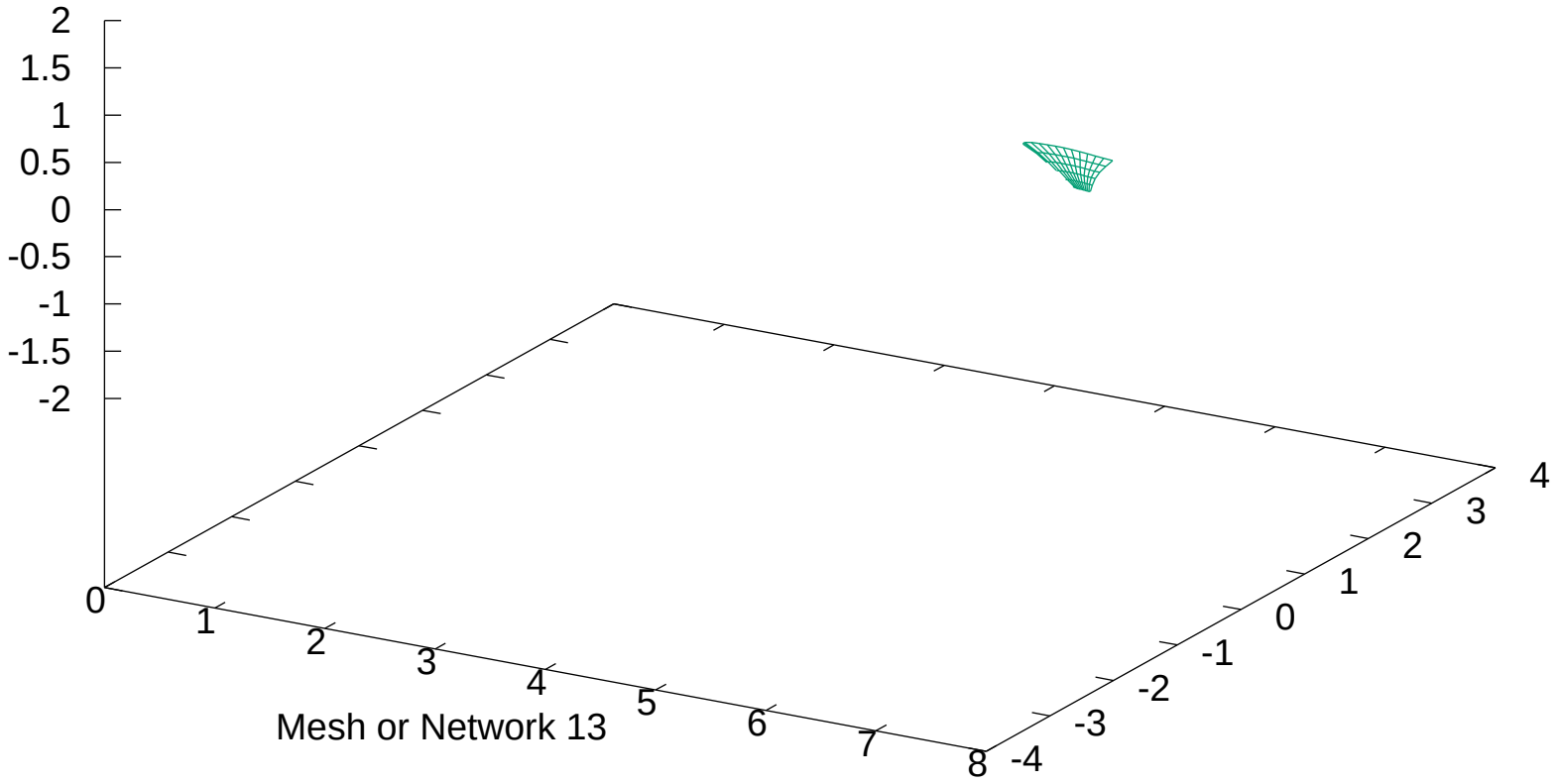
Demo of multiple mesh per file capability - Digitized Blue Whale



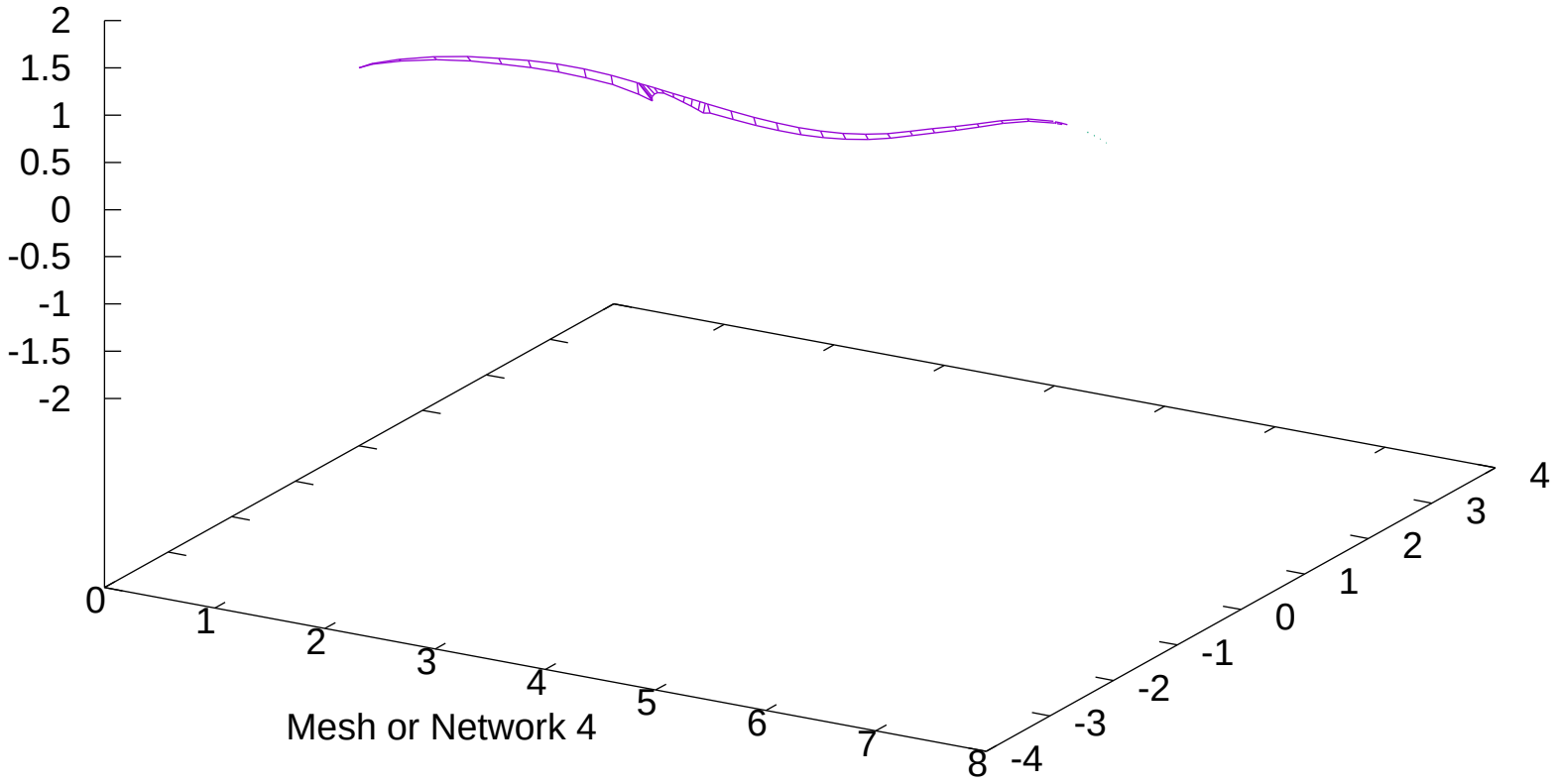
Demo of multiple mesh per file capability - Digitized Blue Whale



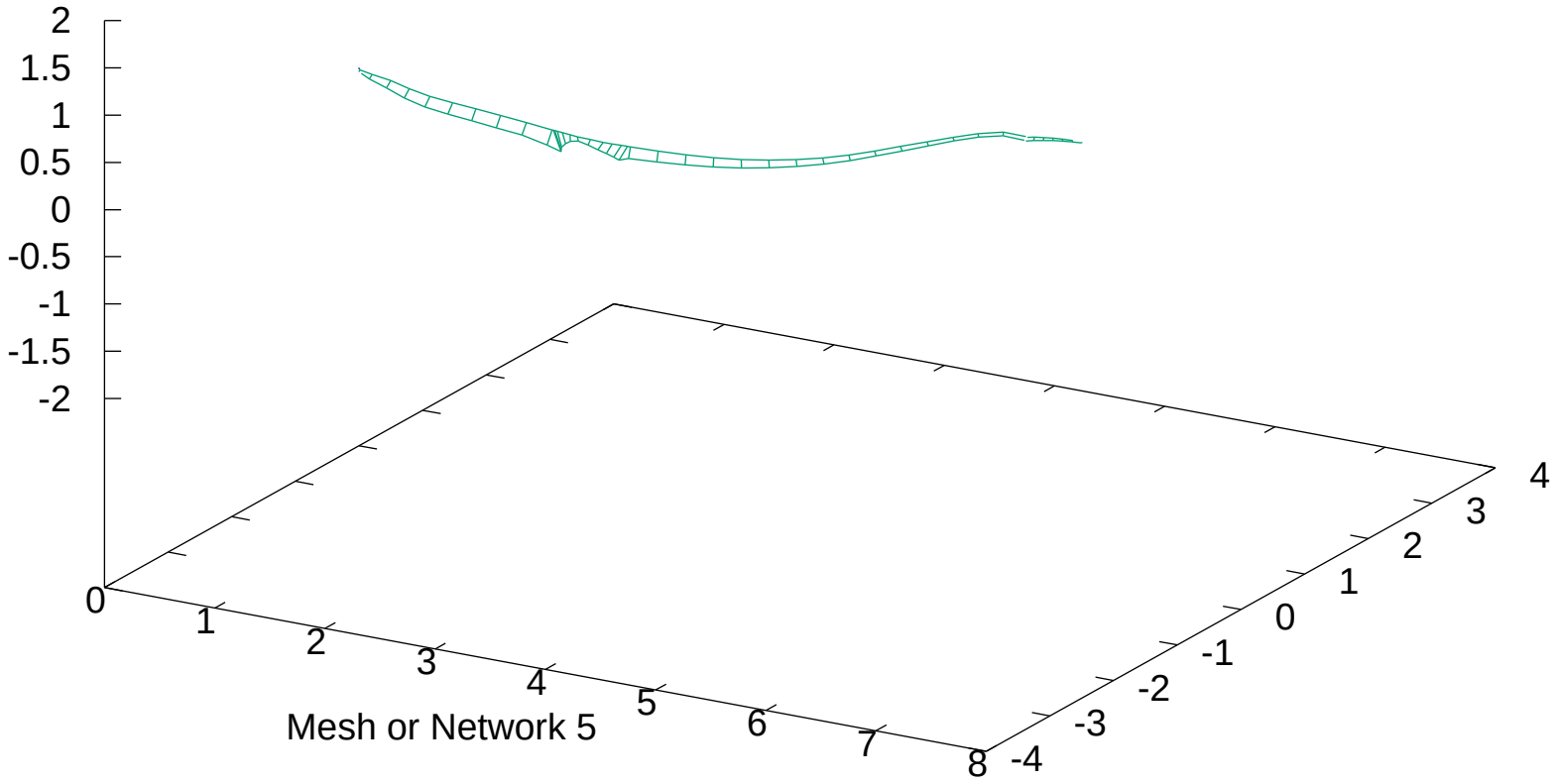
Demo of multiple mesh per file capability - Digitized Blue Whale

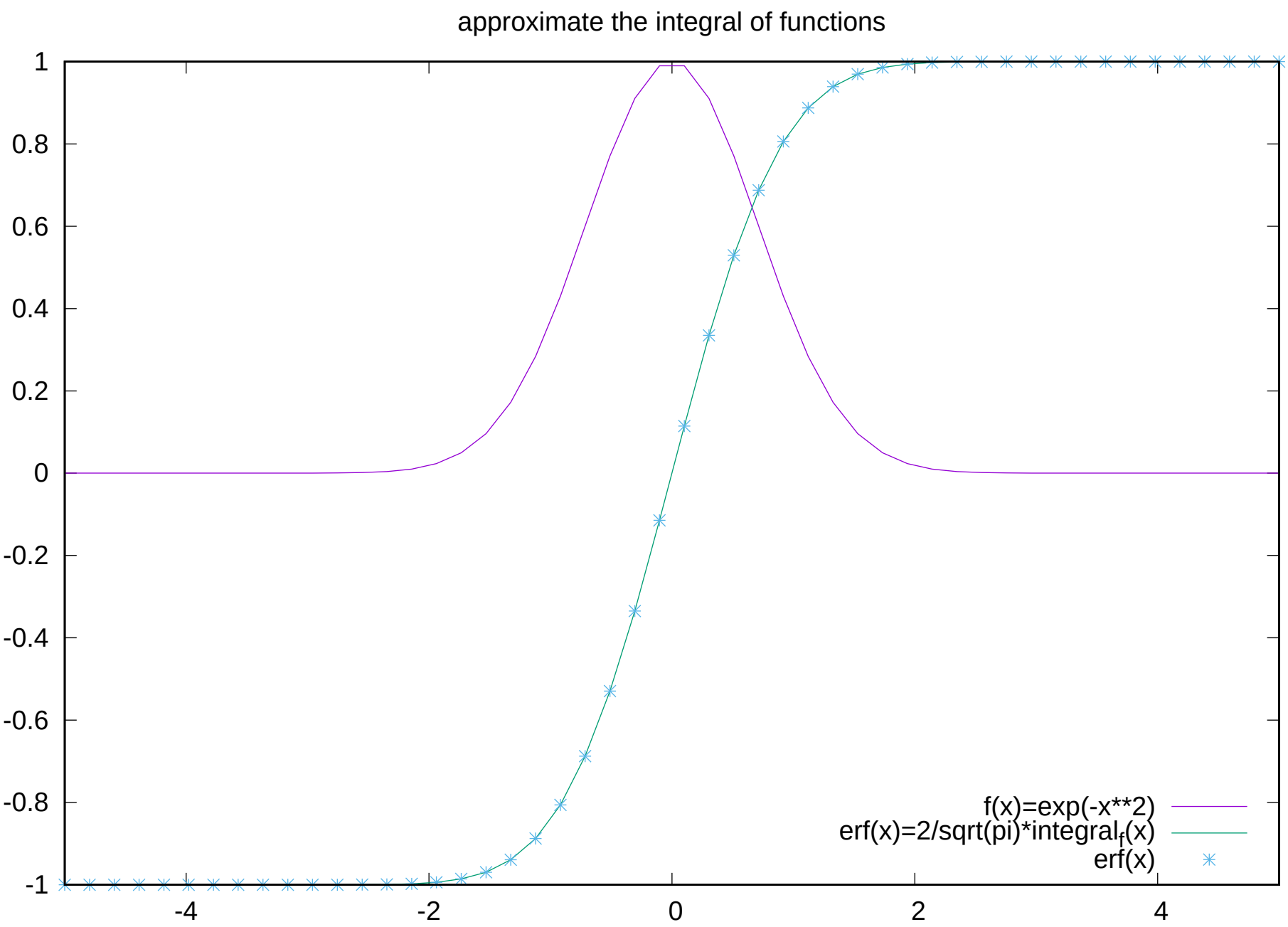


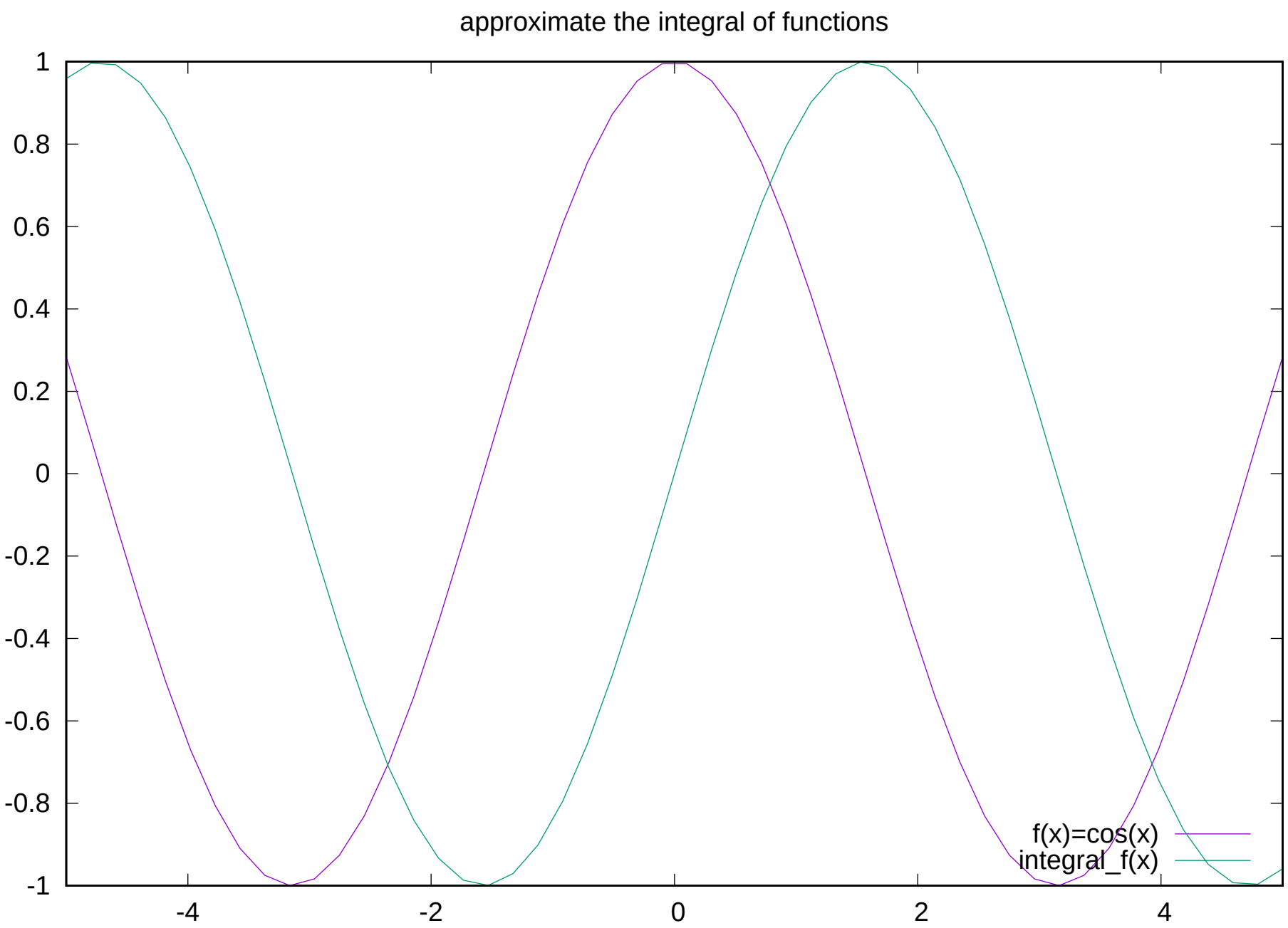
Demo of multiple mesh per file capability - Digitized Blue Whale

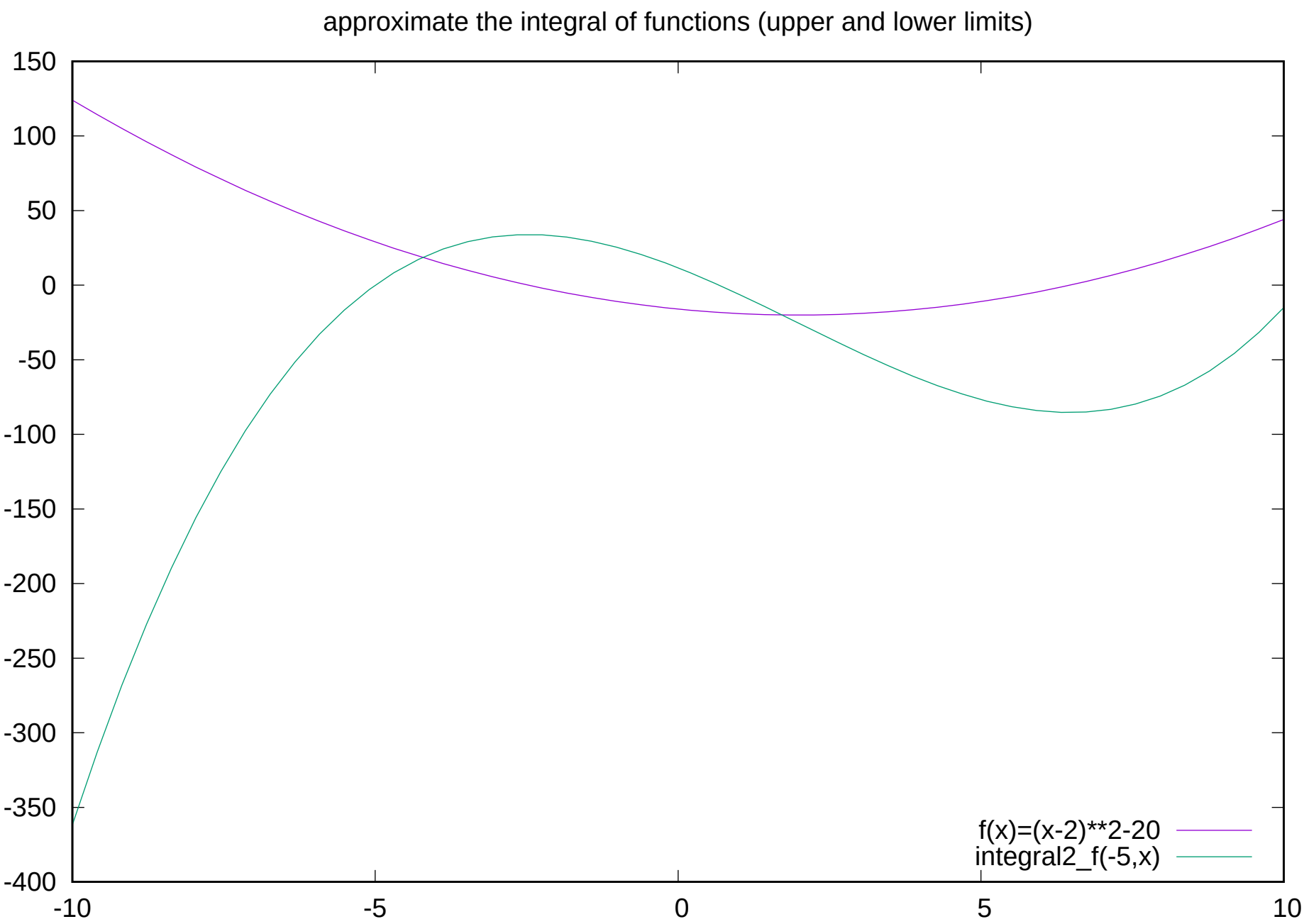


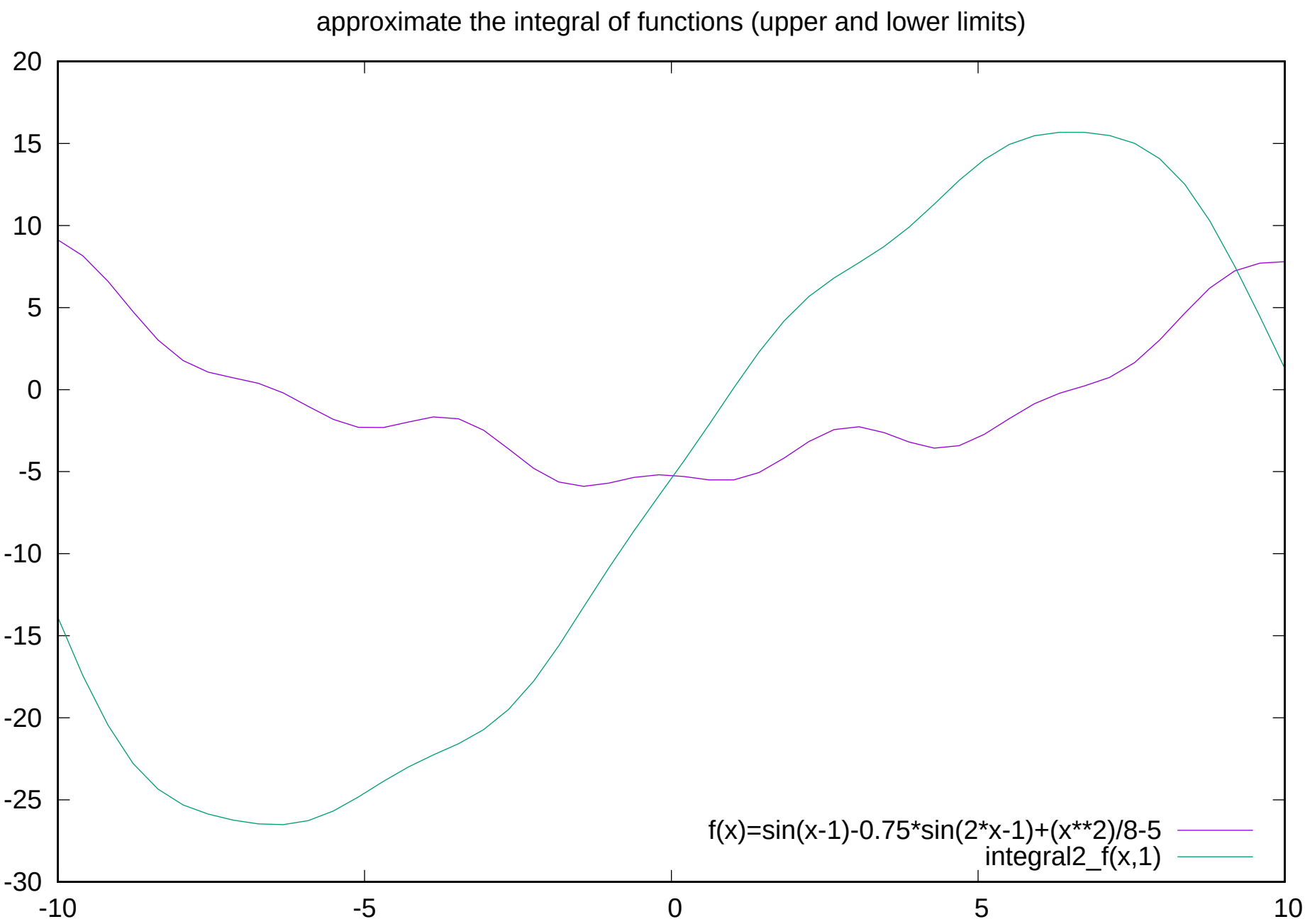
Demo of multiple mesh per file capability - Digitized Blue Whale



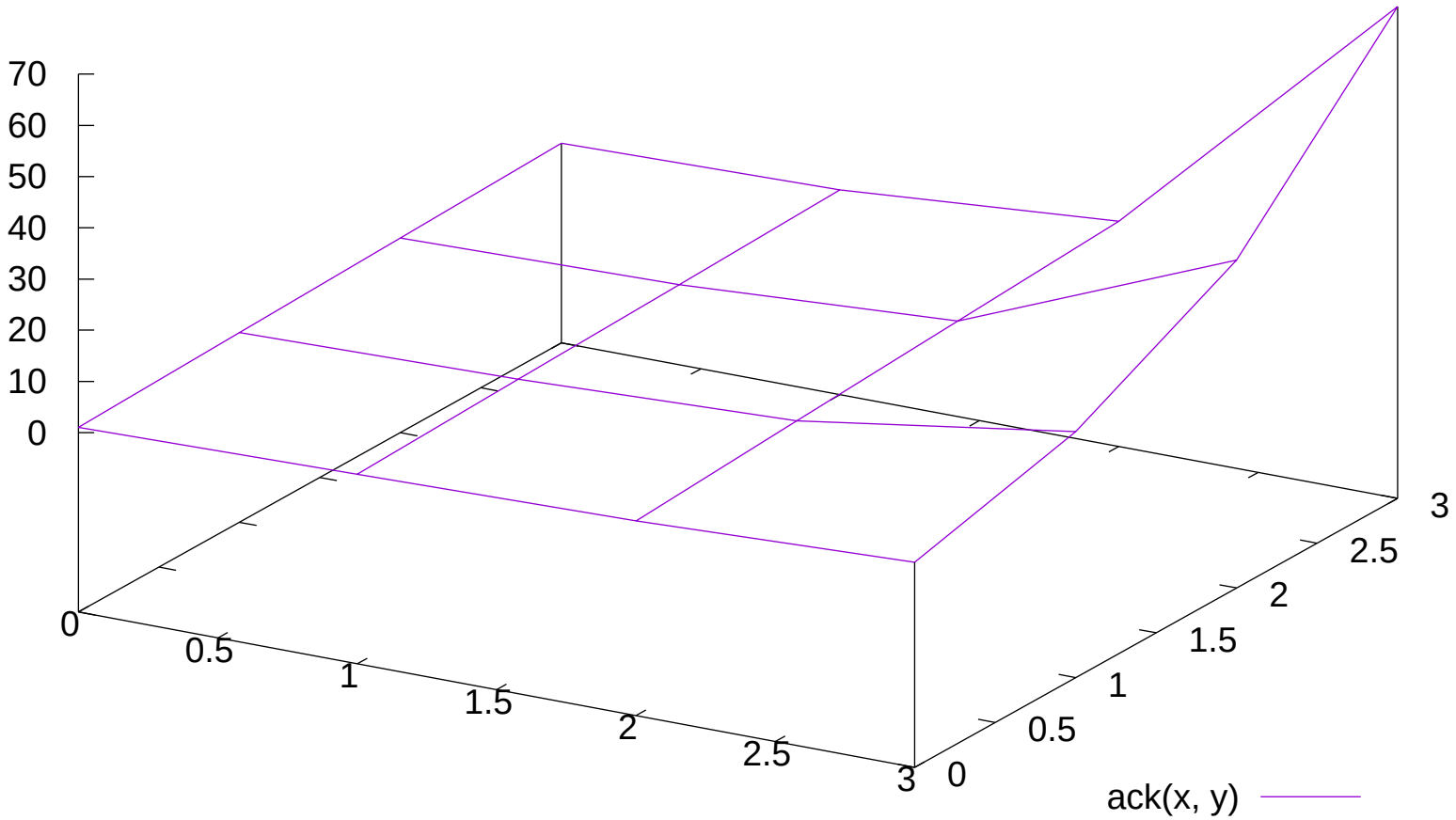

approximate the integral of functions
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-4
-2
0
2
4
f(x)=exp(-x**2)
erf(x)=2/sqrt(pi)*integral_f(x)
erf(x)

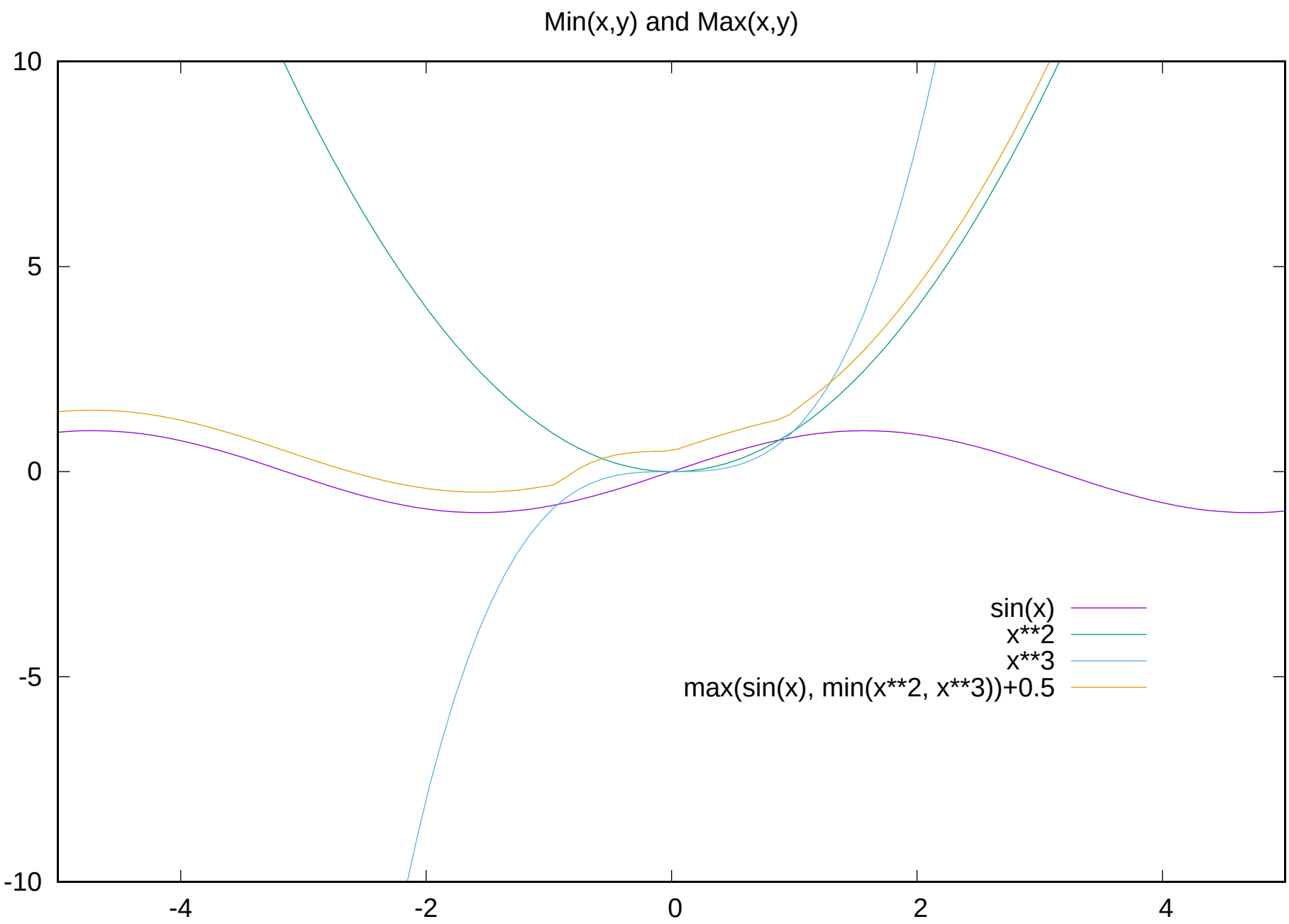

approximate the integral of functions
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-4
-2
0
2
4
f(x)=cos(x)
integral_f(x)

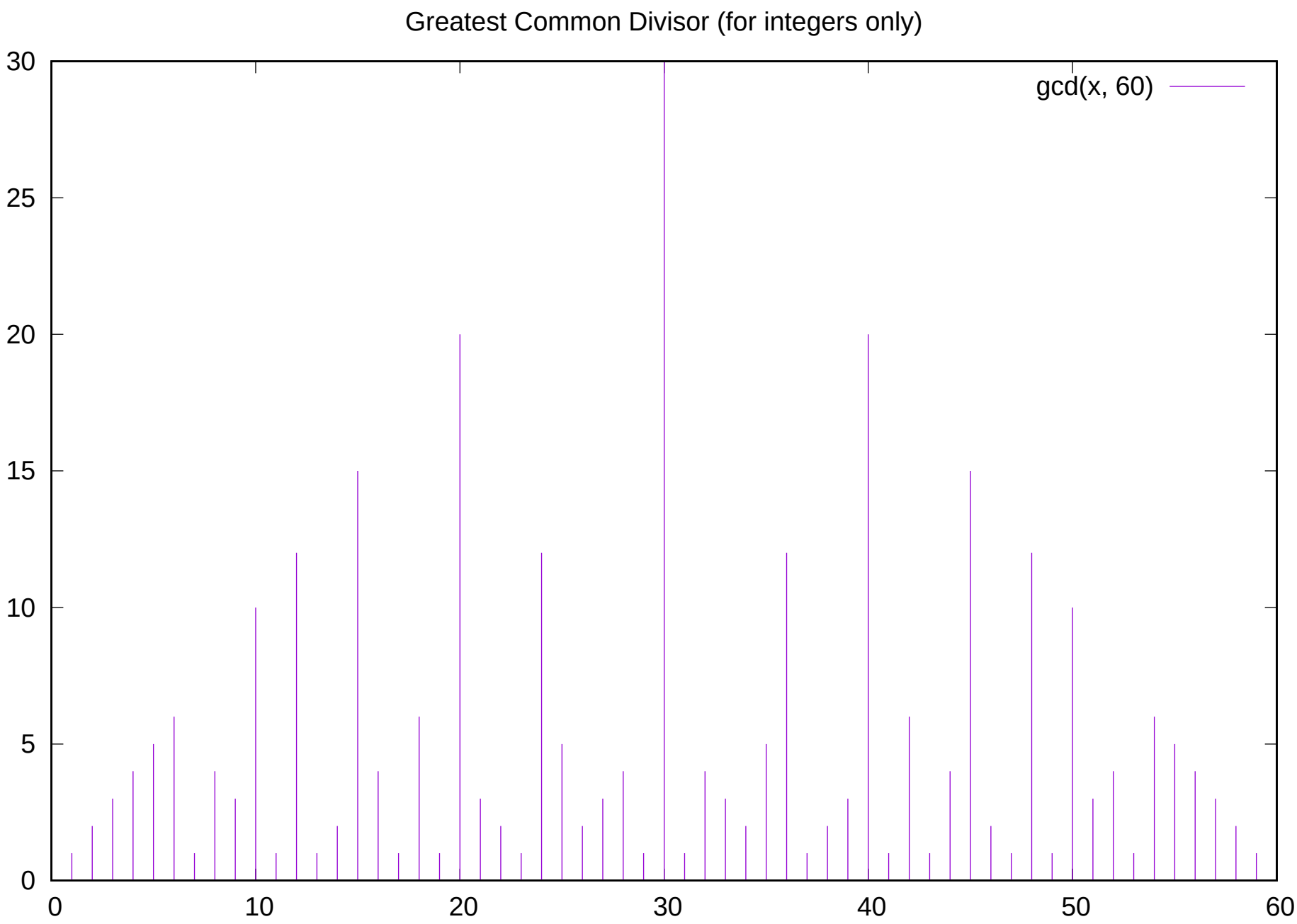

approximate the integral of functions (upper and lower limits)
150
100
50
0
-50
-100
-150
-200
-250
-300
-350
-400
-10
-5
0
5
10
f(x)=(x-2)**2-20
integral2_f(-5,x)

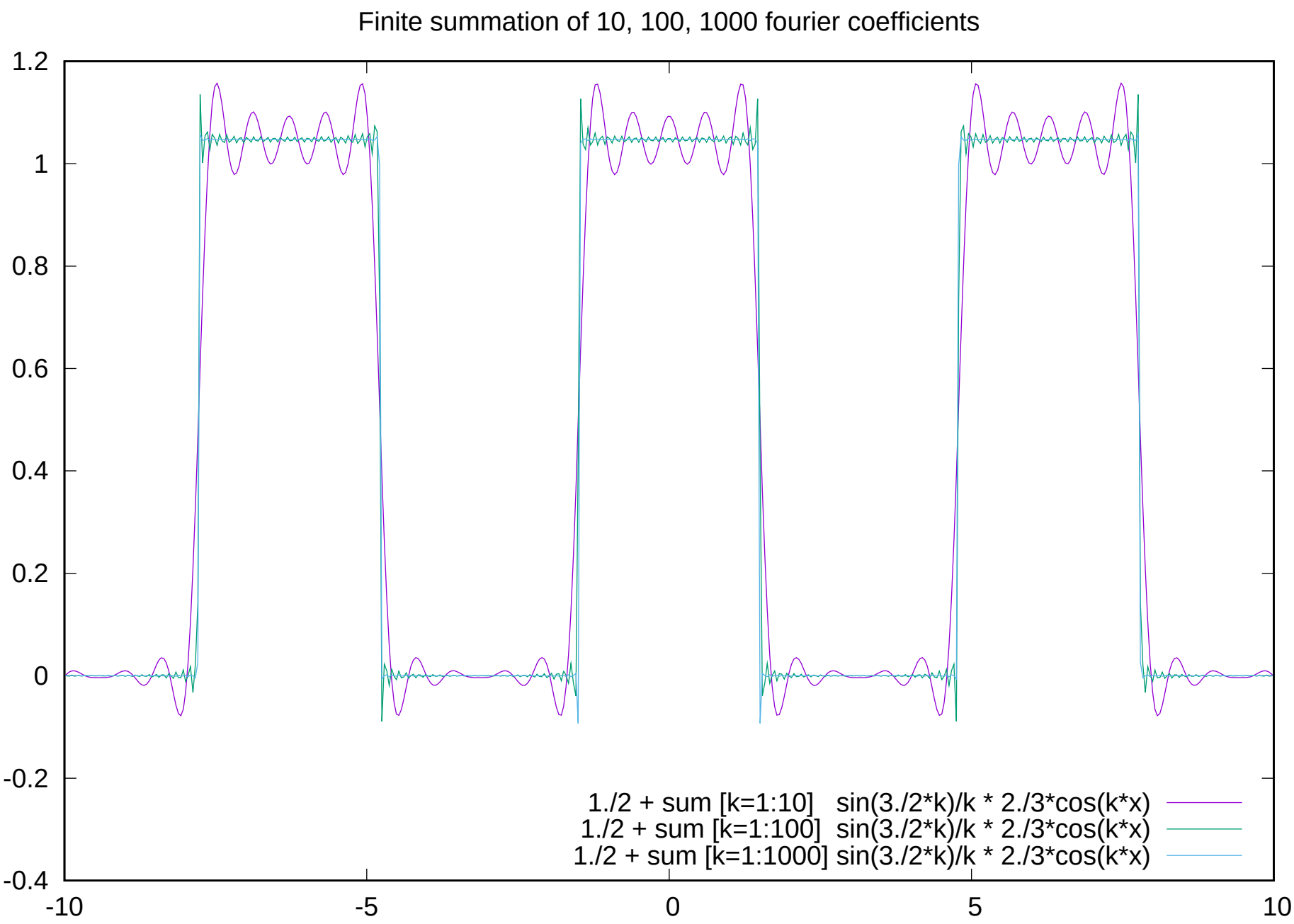

approximate the integral of functions (upper and lower limits)
20
15
10
5
0
-5
-10
-15
-20
-25
-30
-10
-5
0
5
10
f(x)=sin(x-1)-0.75*sin(2*x-1)+(x**2)/8-5
integral2_f(x,1)

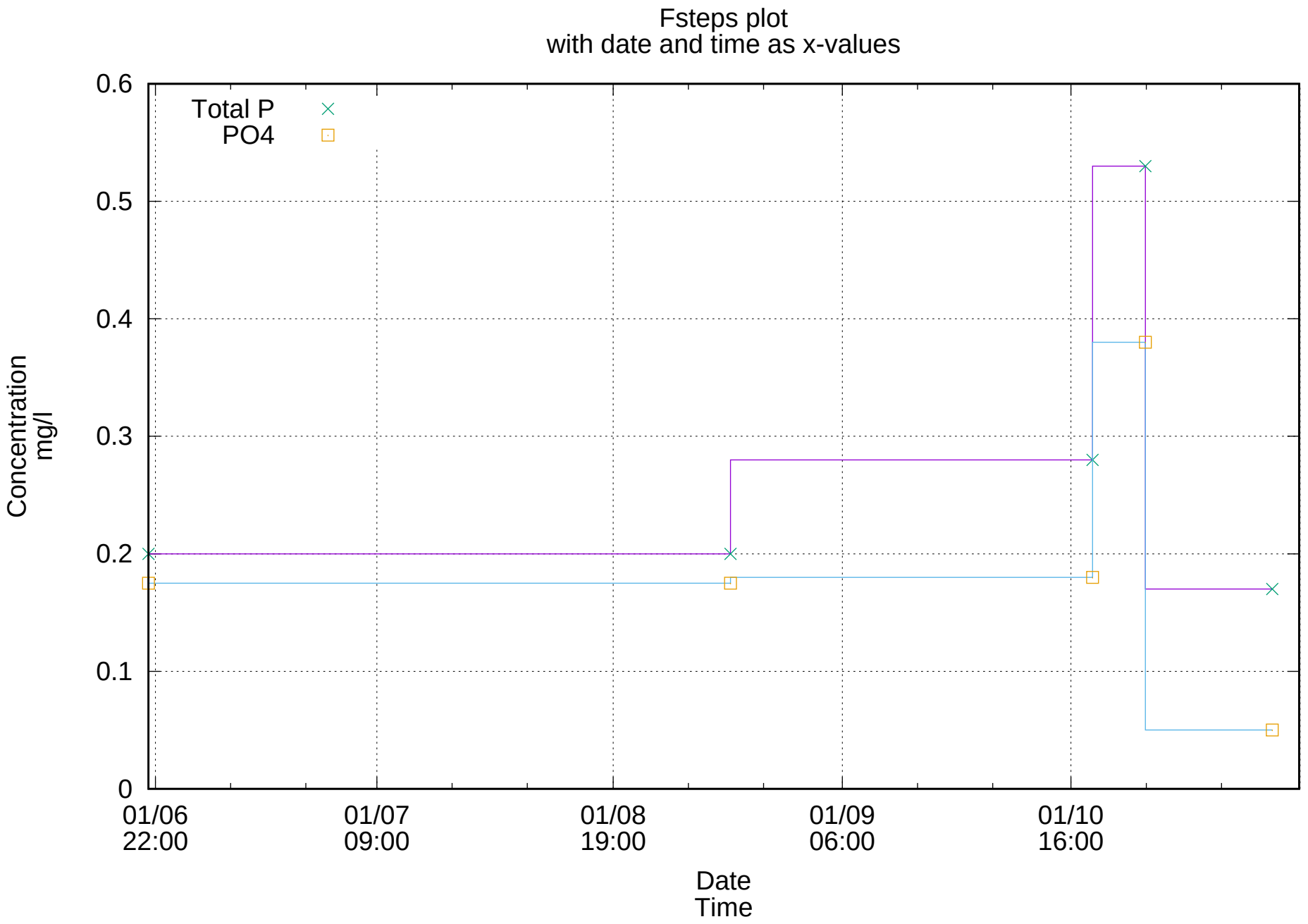
Plot of the ackermann function

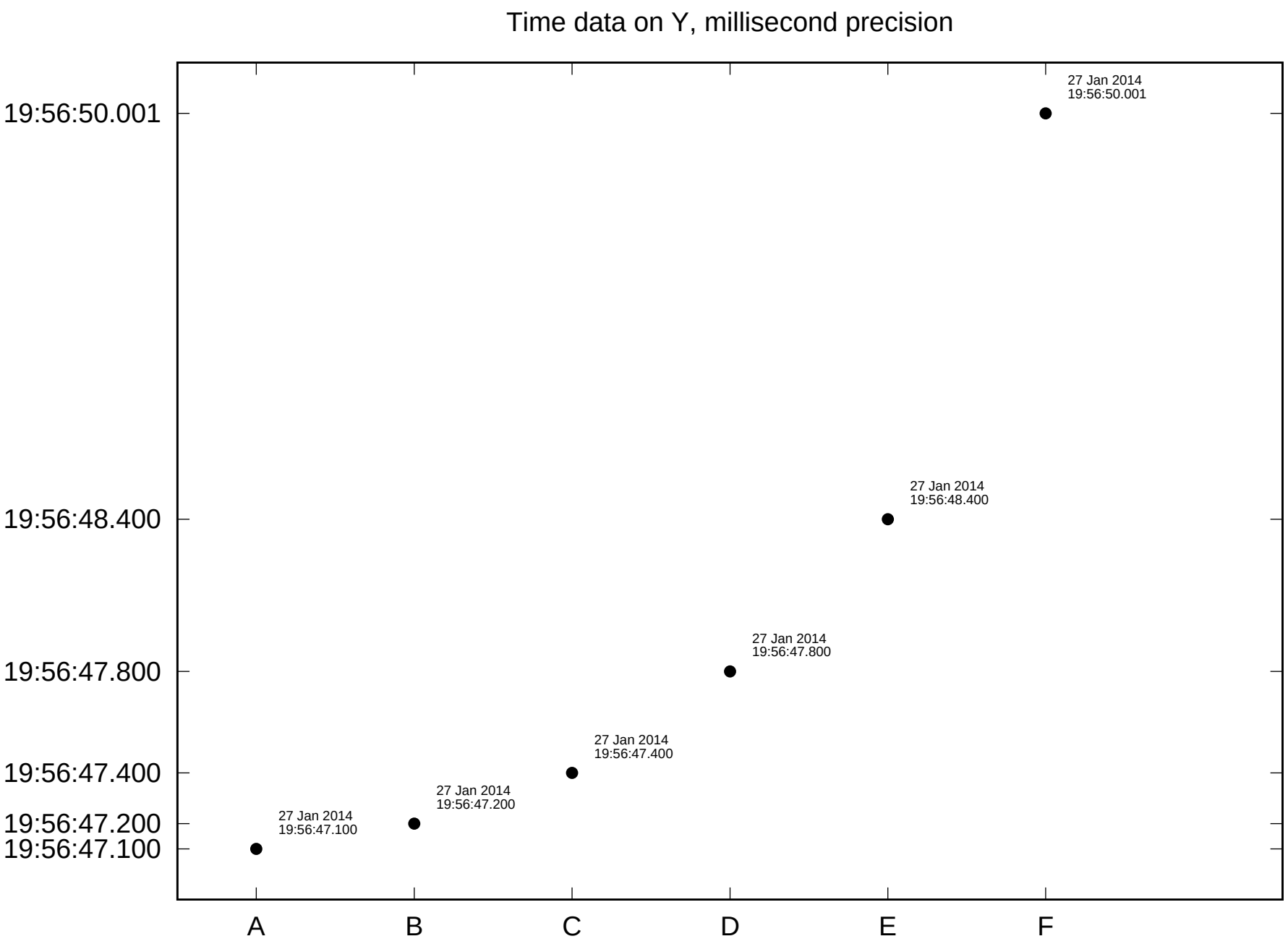


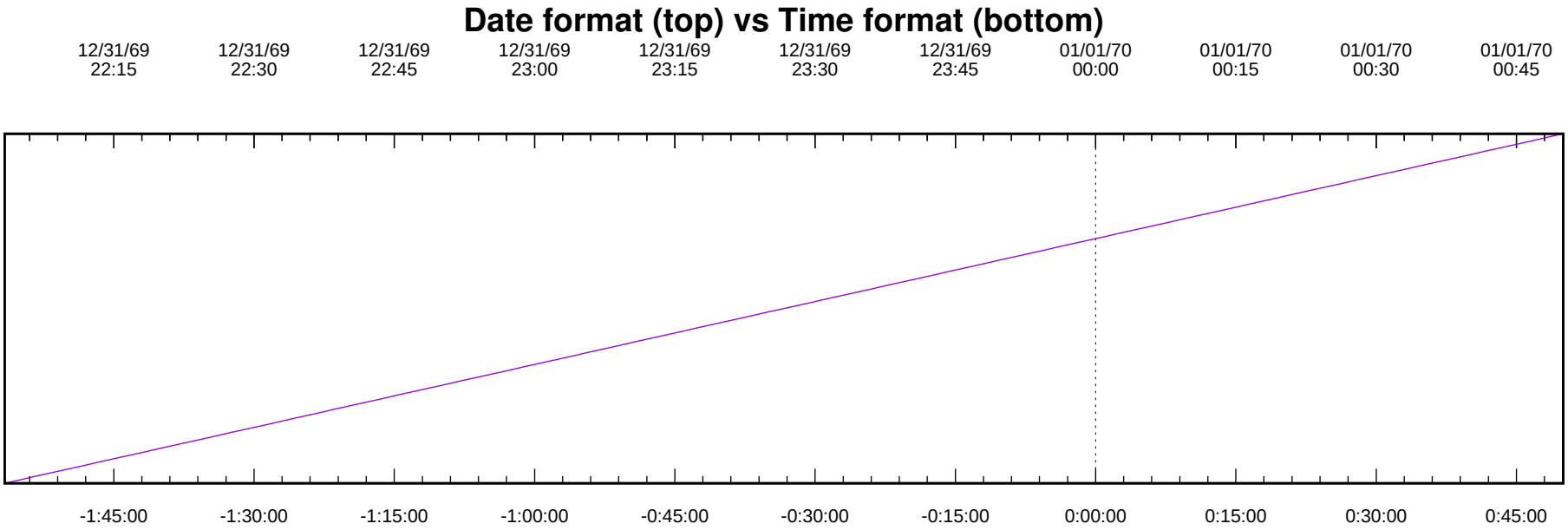

Min(x,y) and Max(x,y)
10
5
0
-5
-10
-4
-2
0
2
4
sin(x)
x**2
x**3
max(sin(x), min(x**2, x**3))+0.5

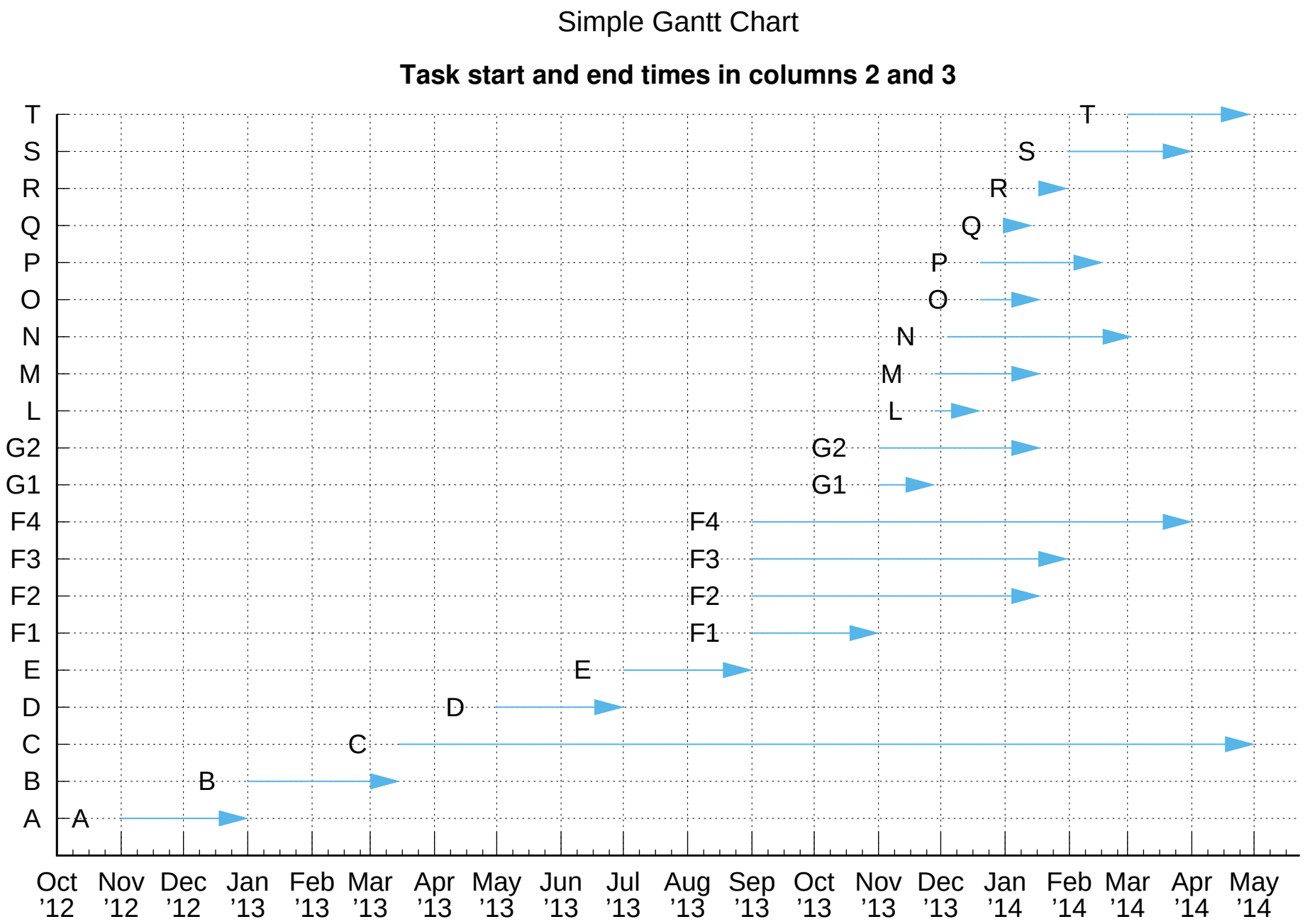

Greatest Common Divisor (for integers only)
gcd(x, 60)
30
25
20
15
10
5
0
0
10
20
30
40
50
60

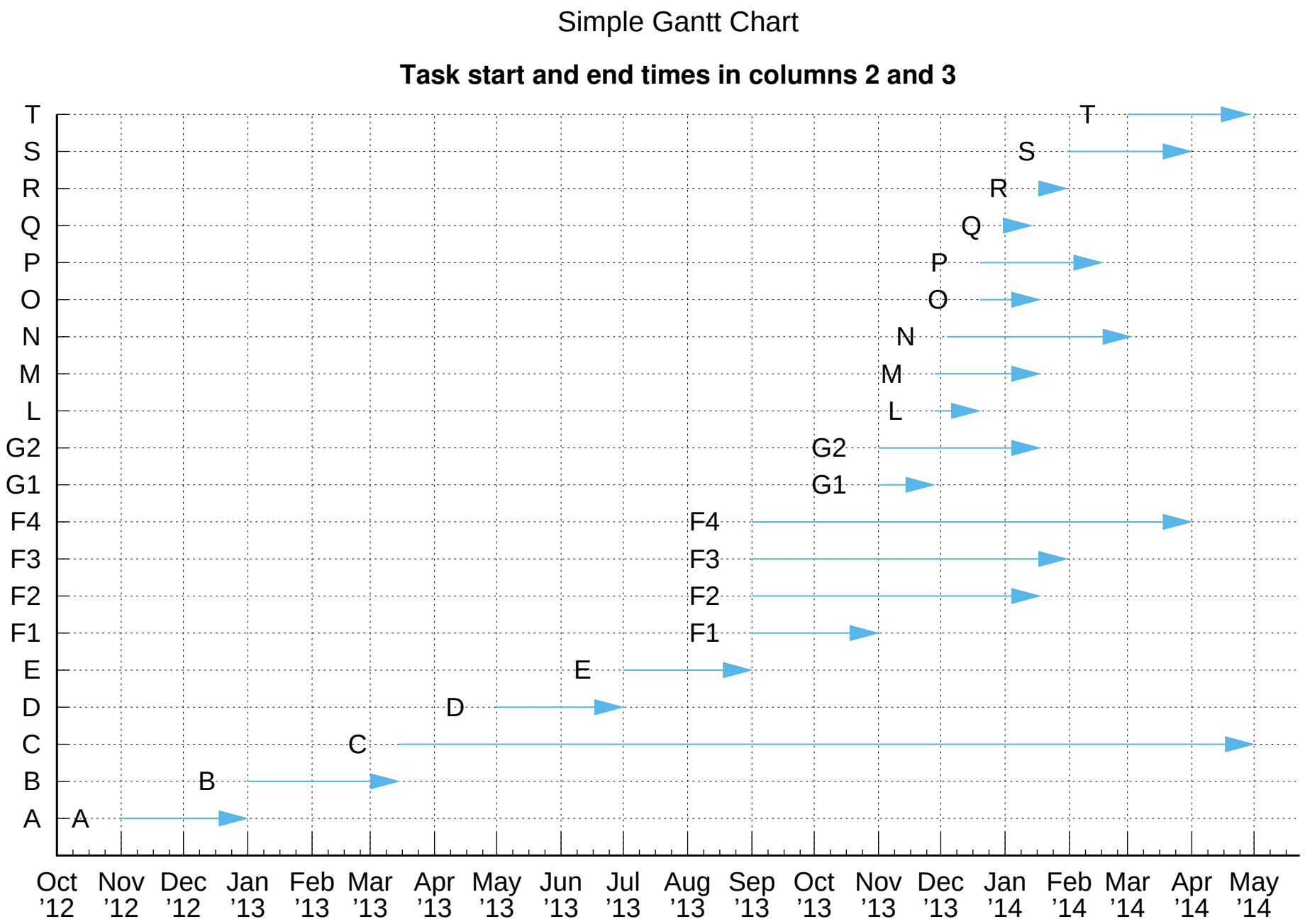

Finite summation of 10, 100, 1000 fourier coefficients
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10
1./2 + sum [k=1:10] sin(3./2*k)/k * 2./3*cos(k*x)
1./2 + sum [k=1:100] sin(3./2*k)/k * 2./3*cos(k*x)
1./2 + sum [k=1:1000] sin(3./2*k)/k * 2./3*cos(k*x)

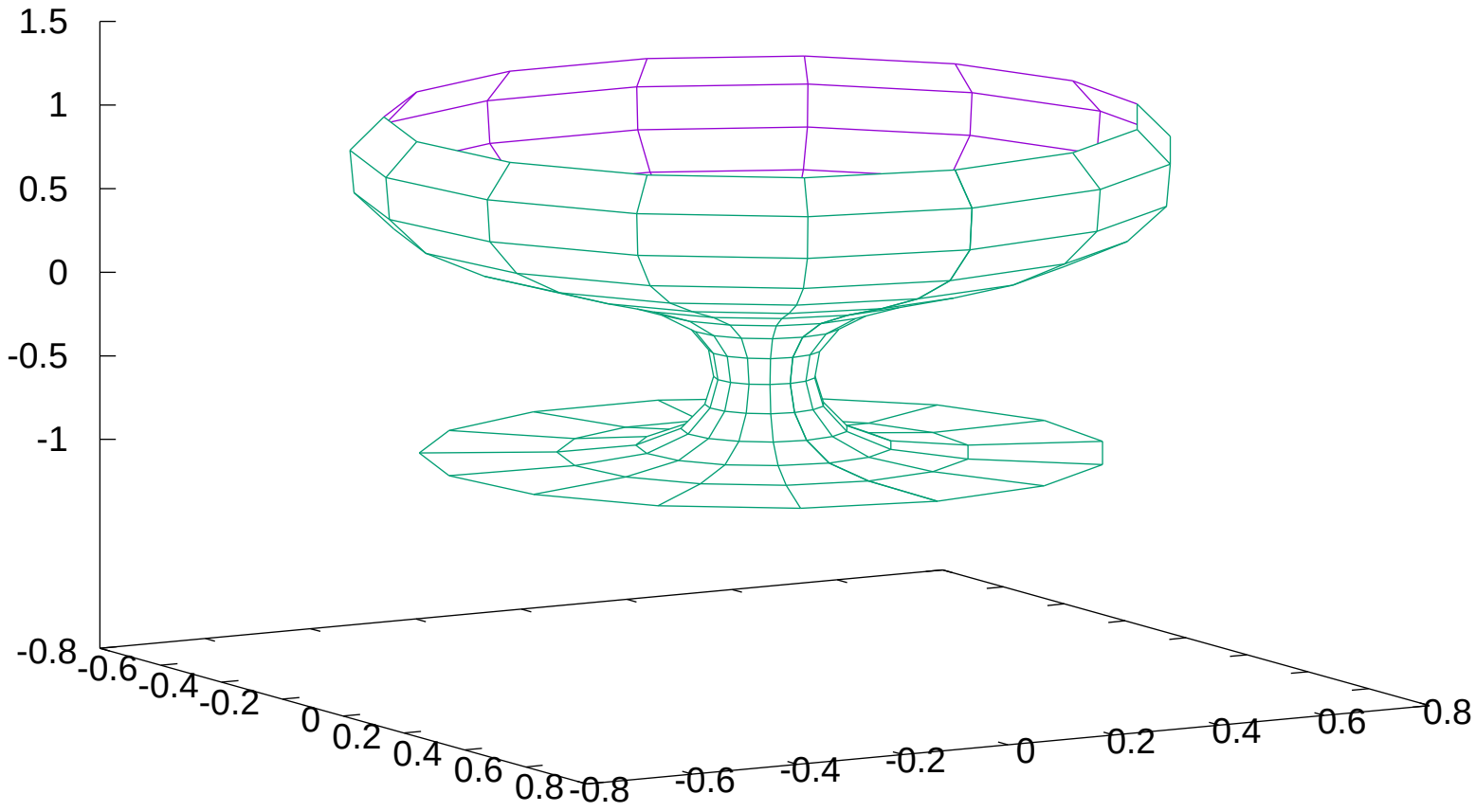

Fsteps plot
with date and time as x-values
Total P
PO4
Concentration
mg/l
0.6
0.5
0.4
0.3
0.2
0.1
0
01/06
22:00
01/07
09:00
01/08
19:00
01/09
06:00
01/10
16:00
Date
Time

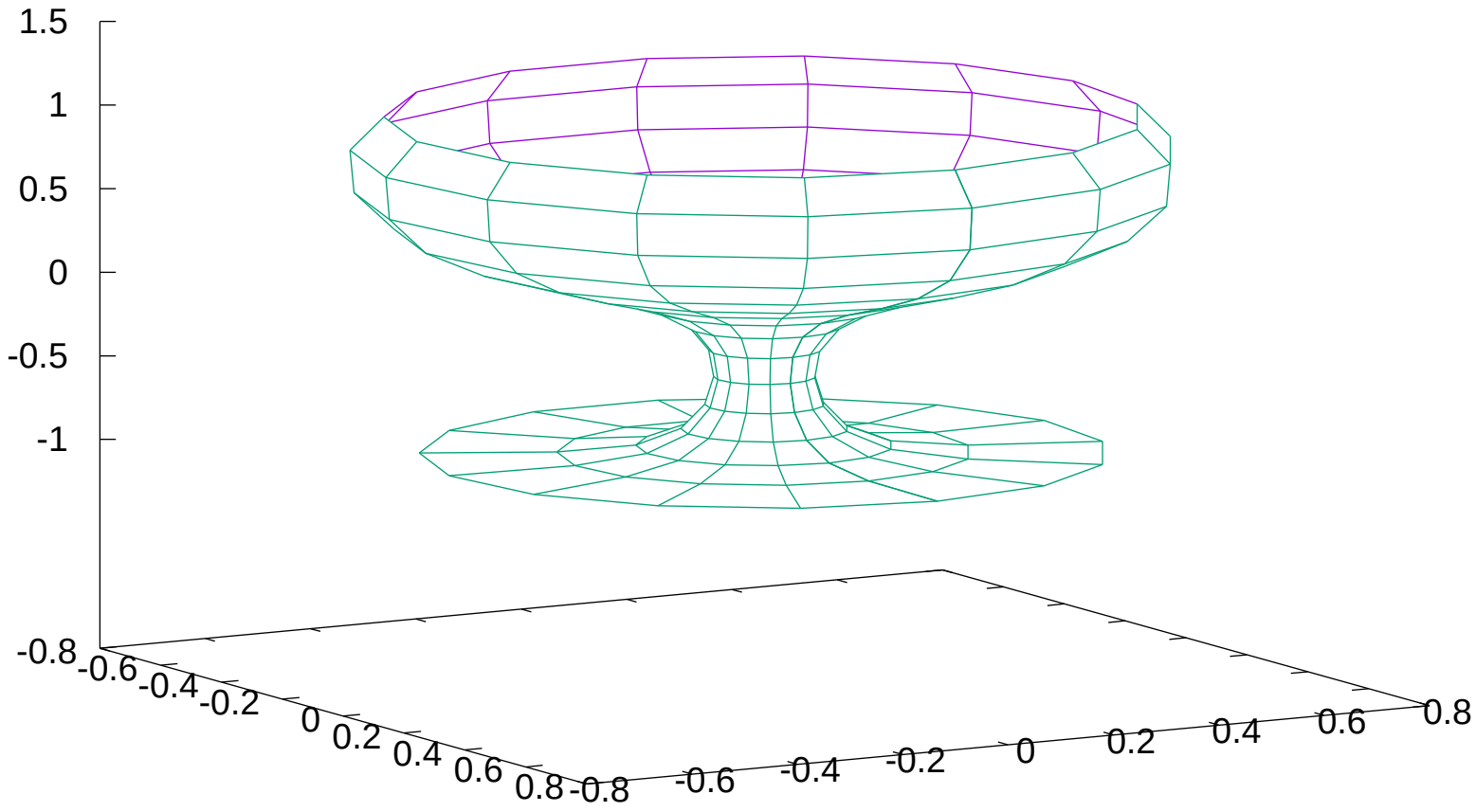

Time data on Y, millisecond precision
19:56:50.001
19:56:48.400
19:56:47.800
19:56:47.400
19:56:47.200
19:56:47.100
A
B
C
D
E
F
27 Jan 2014
19:56:50.001
27 Jan 2014
19:56:48.400
27 Jan 2014
19:56:47.800
27 Jan 2014
19:56:47.400
27 Jan 2014
19:56:47.200
27 Jan 2014
19:56:47.100

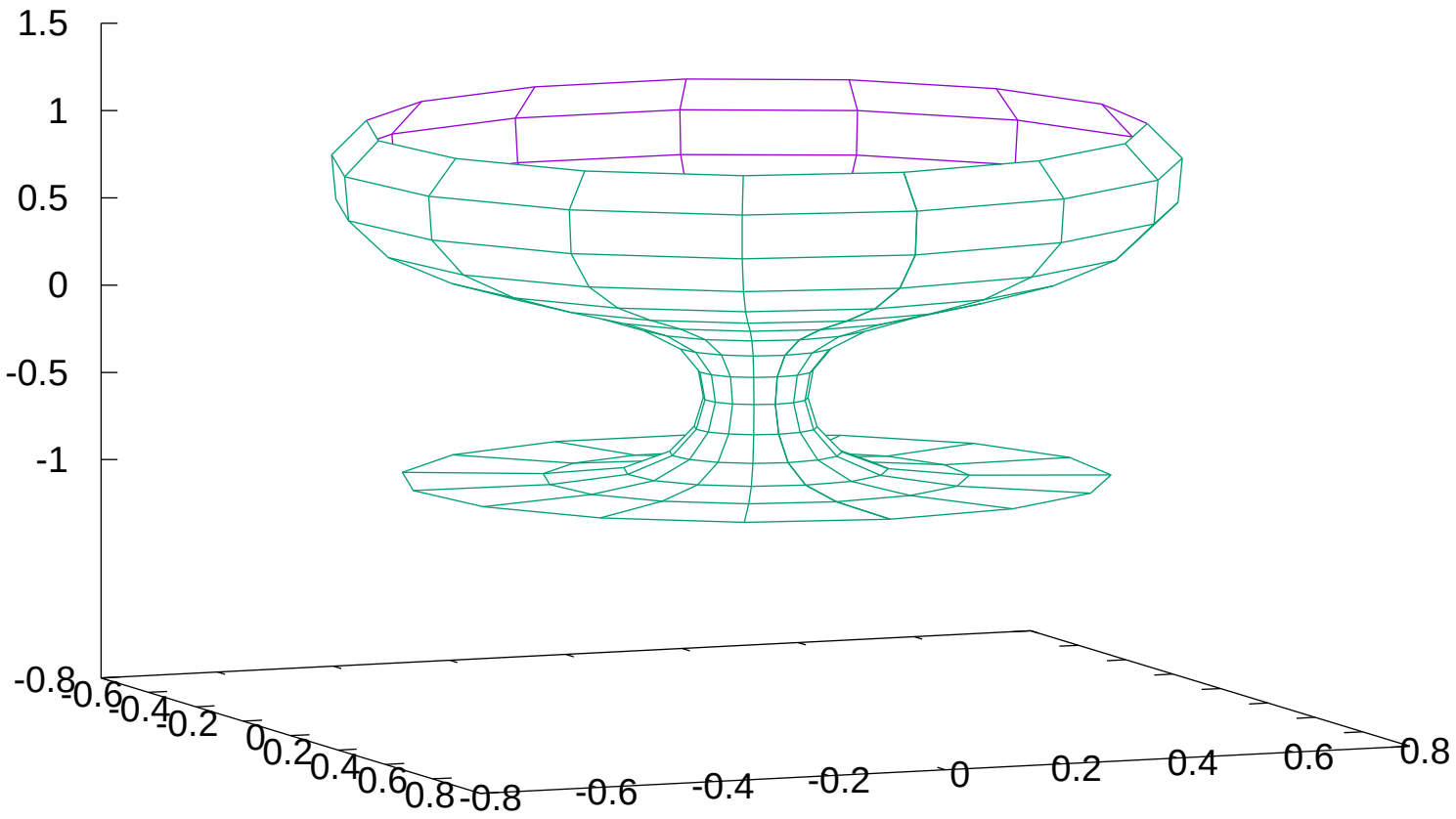

Date format (top) vs Time format (bottom)
12/31/69 22:15
12/31/69 22:30
12/31/69 22:45
12/31/69 23:00
12/31/69 23:15
12/31/69 23:30
12/31/69 23:45
01/01/70 00:00
01/01/70 00:15
01/01/70 00:30
01/01/70 00:45
-1:45:00
-1:30:00
-1:15:00
-1:00:00
-0:45:00
-0:30:00
-0:15:00
0:00:00
0:15:00
0:30:00
0:45:00

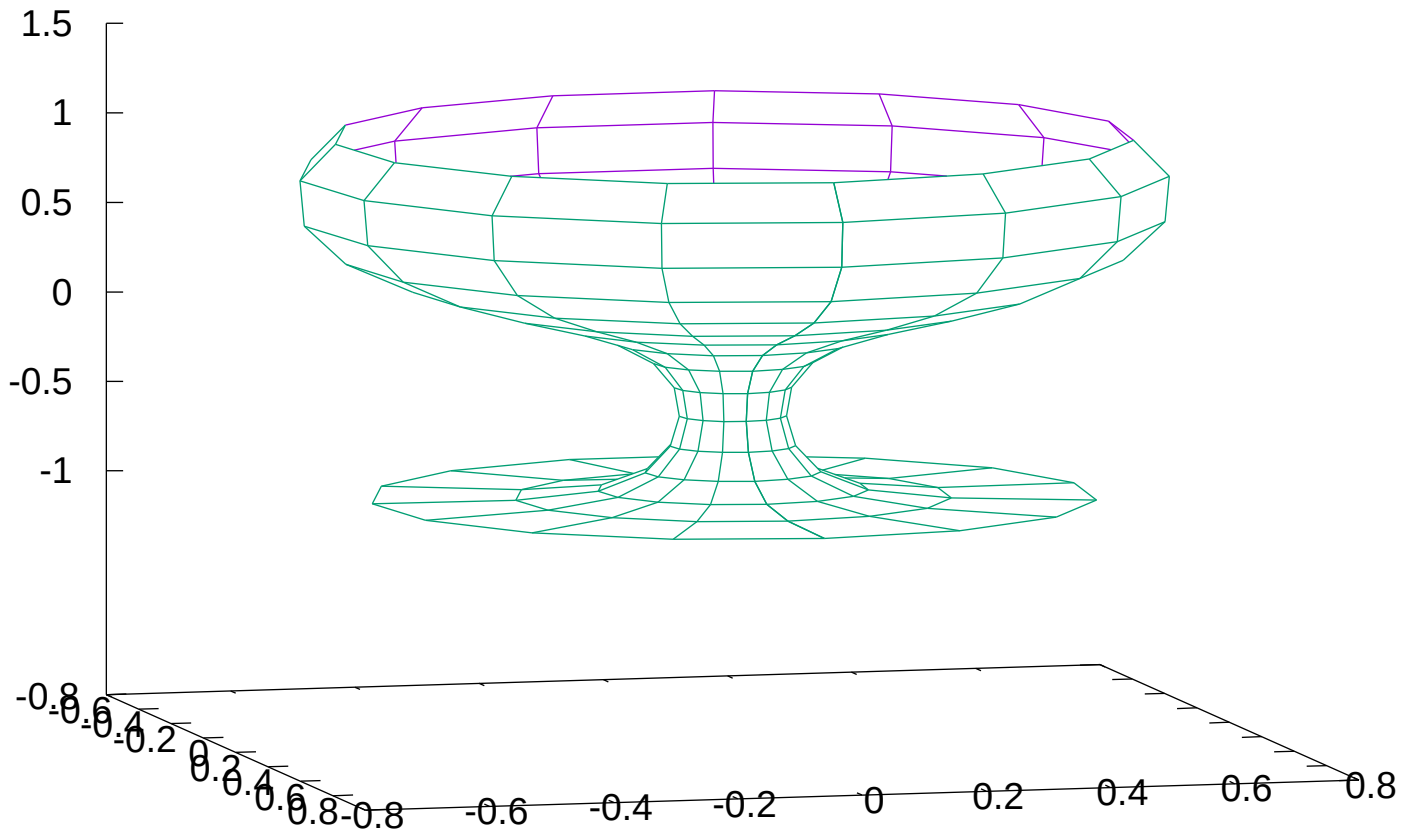

Simple Gantt Chart
Task start and end times in columns 2 and 3
T
S
R
Q
P
O
N
M
L
G2
G1
F4
F3
F2
F1
E
D
C
B
A
Oct '12
Nov '12
Dec '12
Jan '13
Feb '13
Mar '13
Apr '13
May '13
Jun '13
Jul '13
Aug '13
Sep '13
Oct '13
Nov '13
Dec '13
Jan '14
Feb '14
Mar '14
Apr '14
May '14

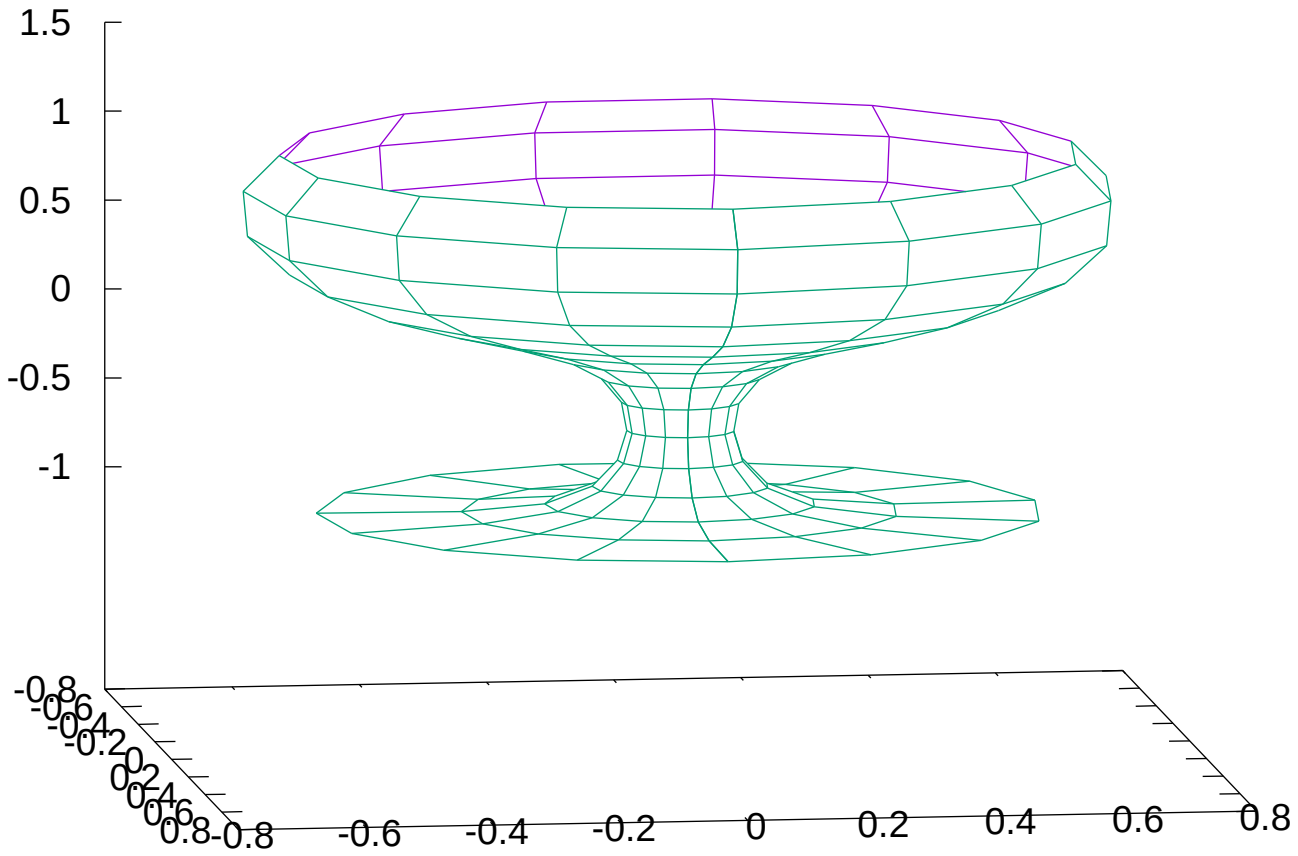

Simple Gantt Chart
Task start and end times in columns 2 and 3
T
S
R
Q
P
O
N
M
L
G2
G1
F4
F3
F2
F1
E
D
C
B
A
Oct '12
Nov '12
Dec '12
Jan '13
Feb '13
Mar '13
Apr '13
May '13
Jun '13
Jul '13
Aug '13
Sep '13
Oct '13
Nov '13
Dec '13
Jan '14
Feb '14
Mar '14
Apr '14
May '14

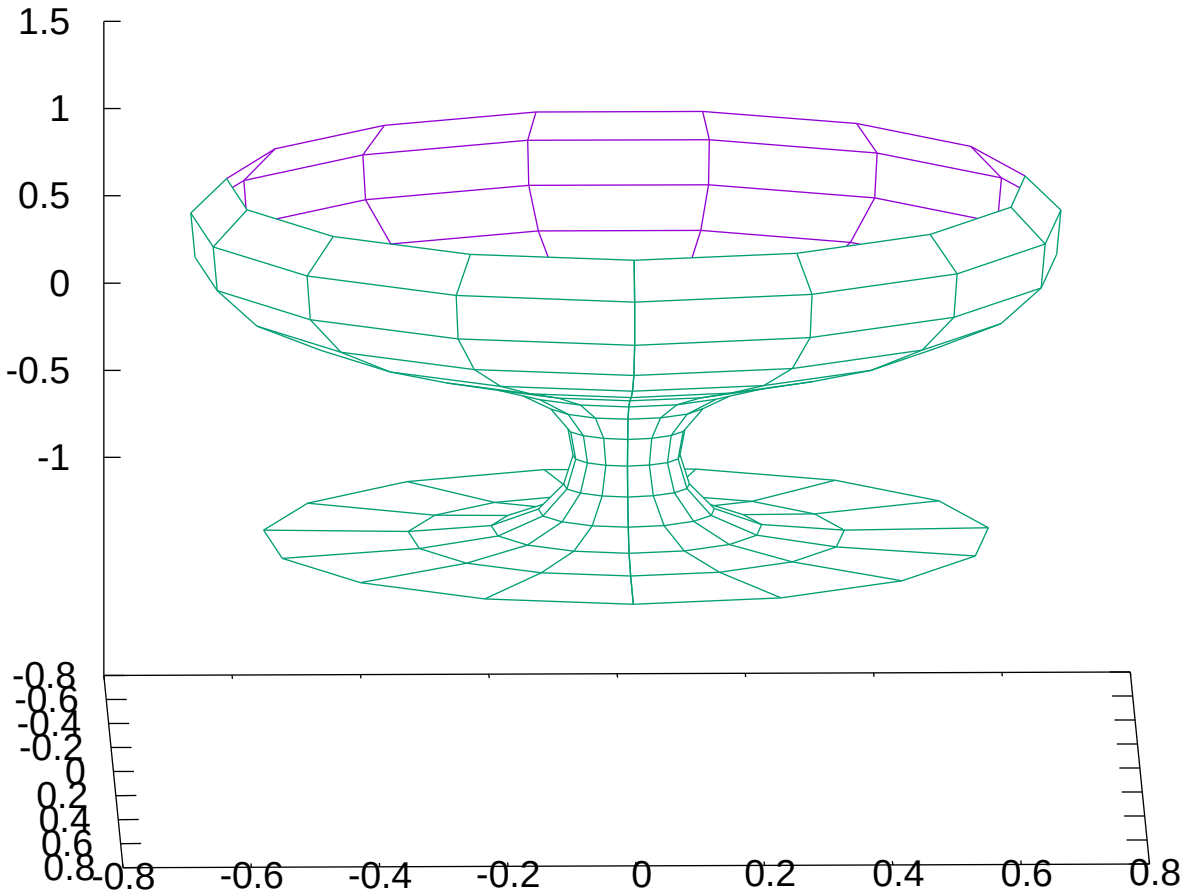

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

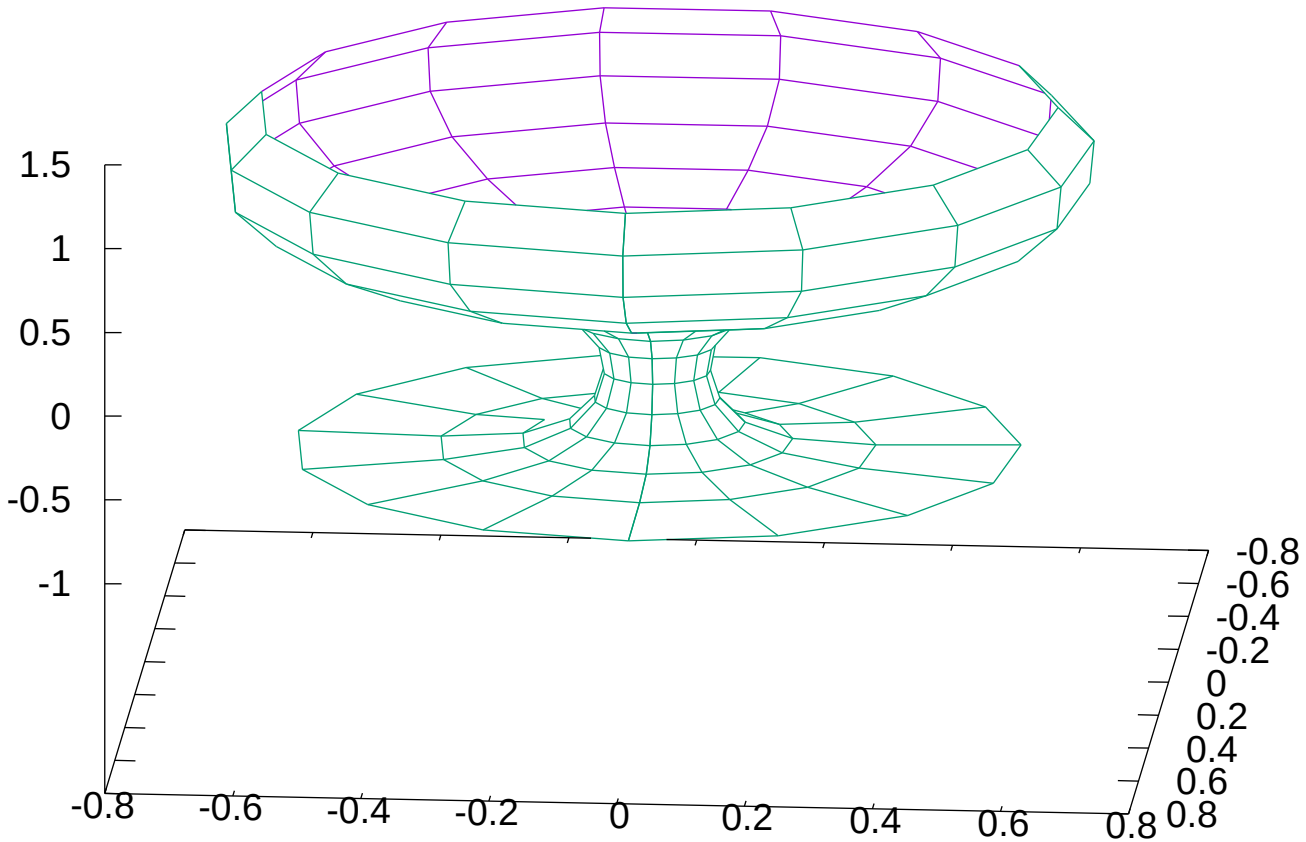

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

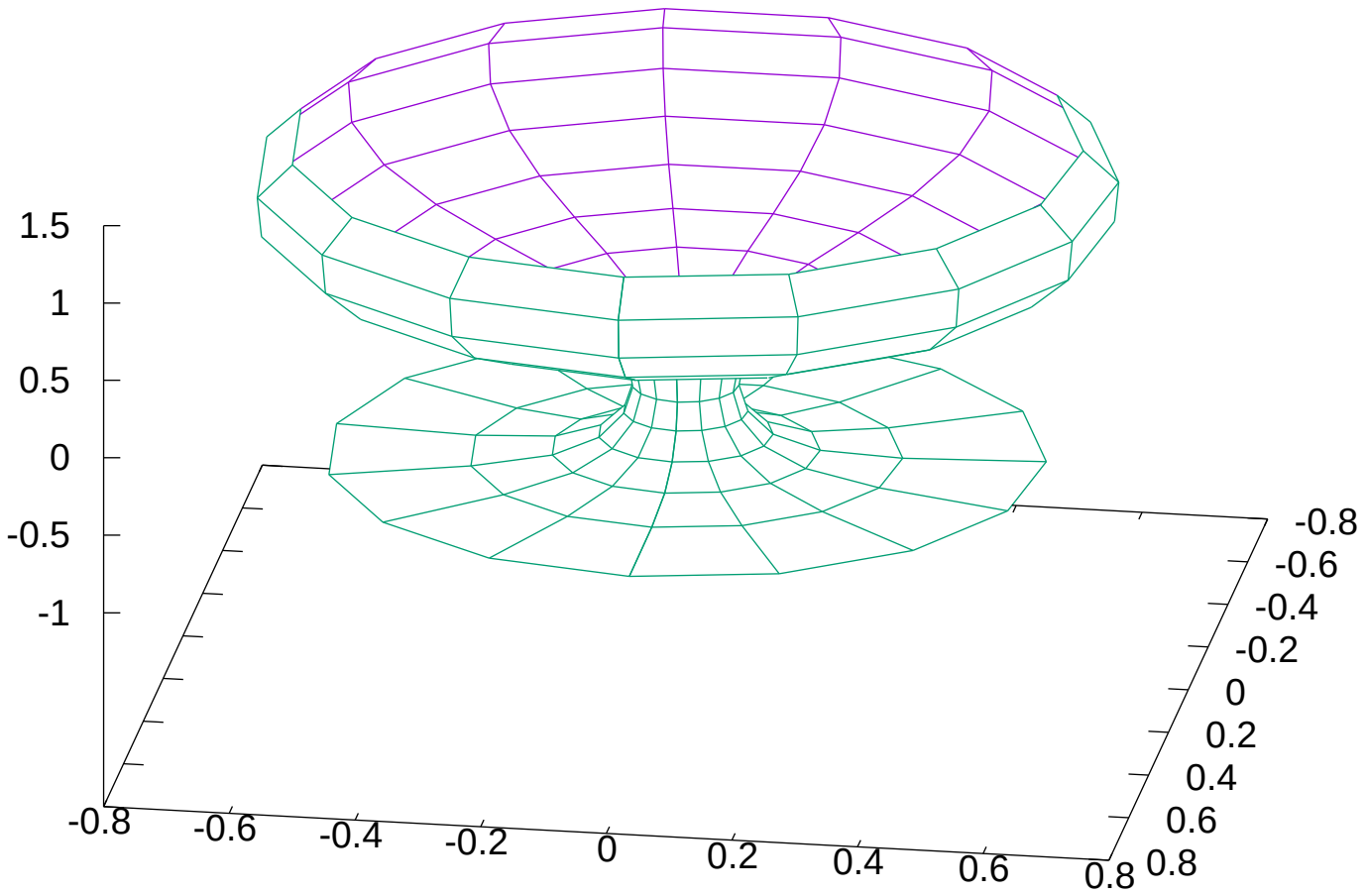

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

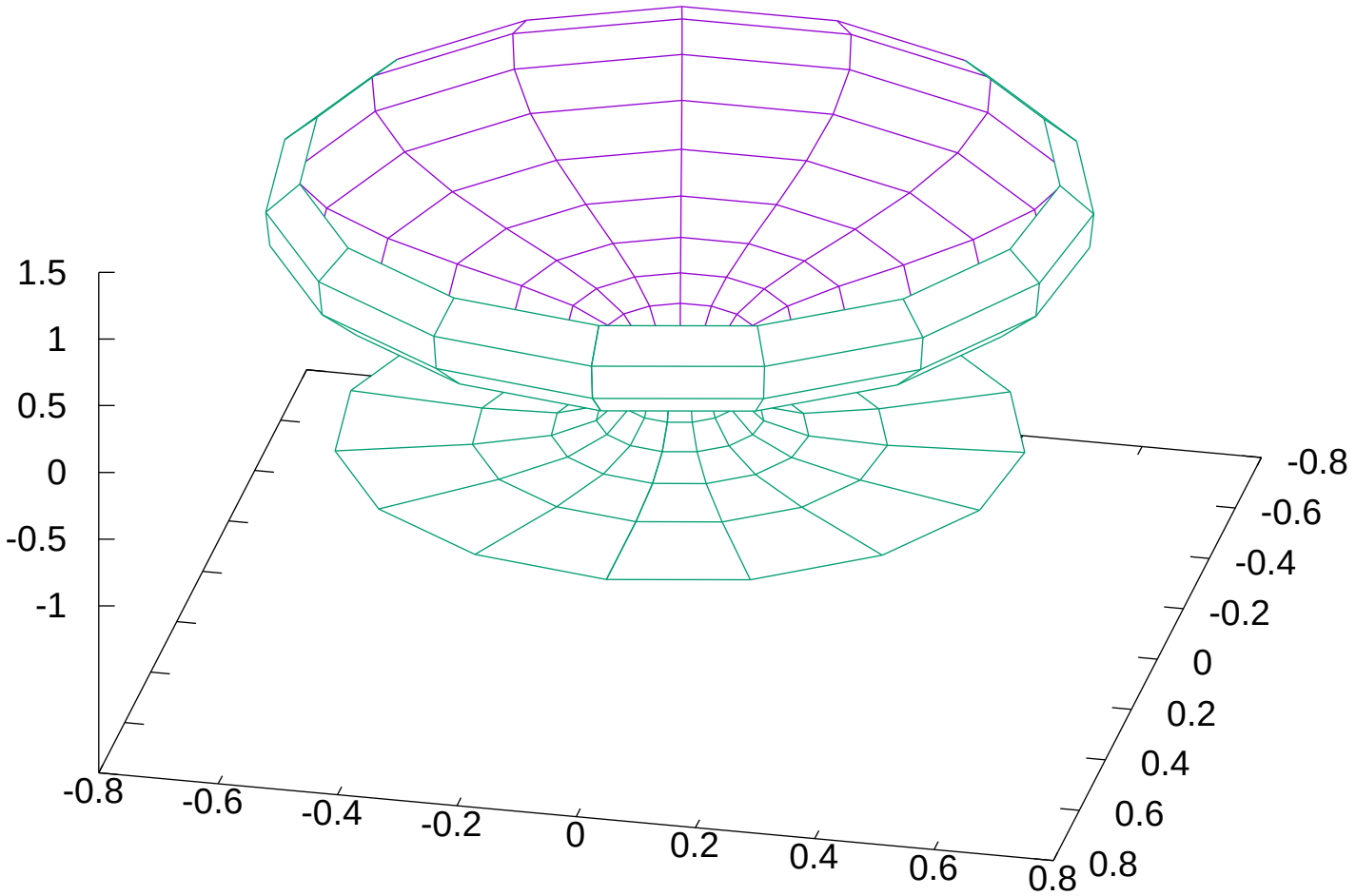

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

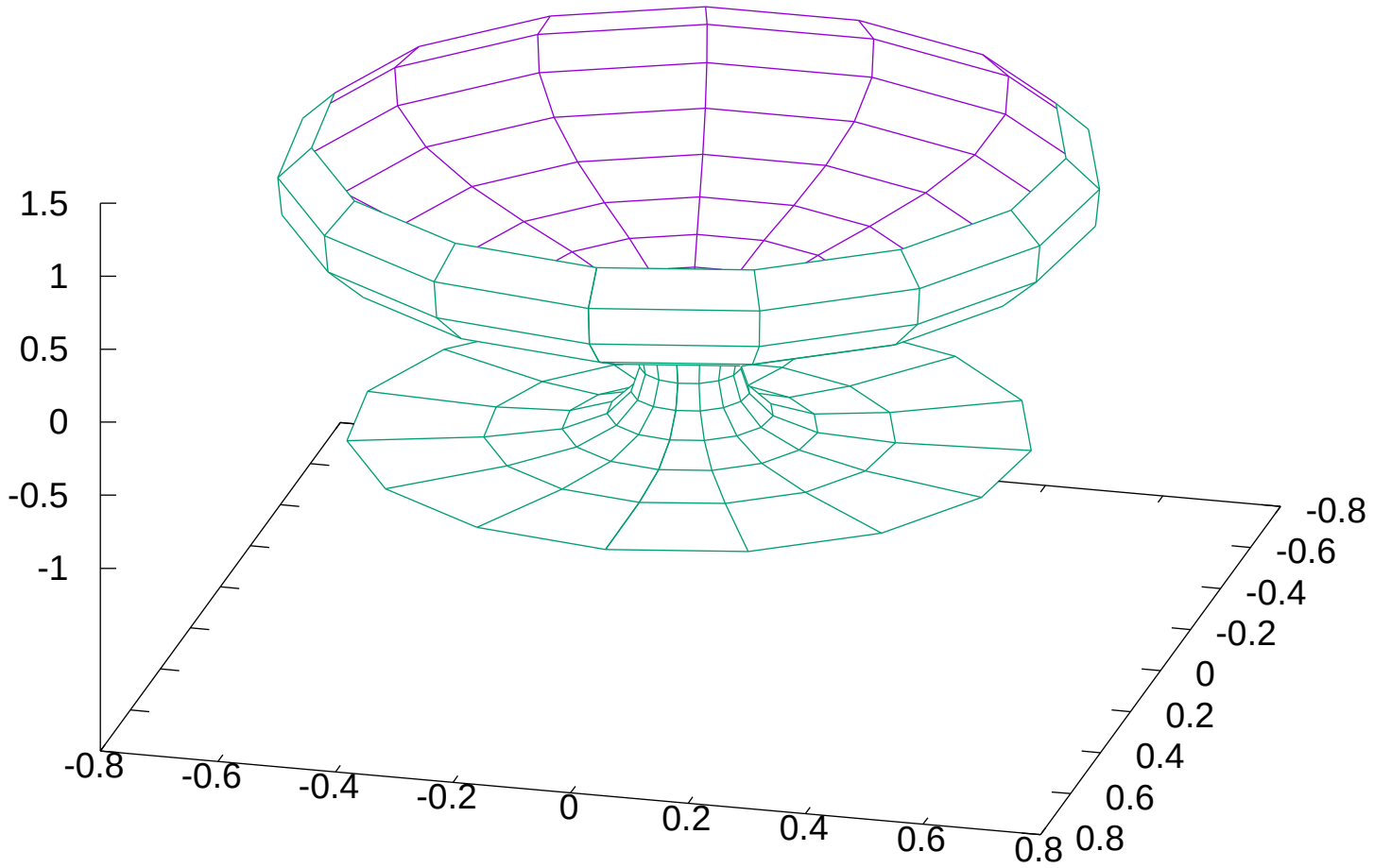

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

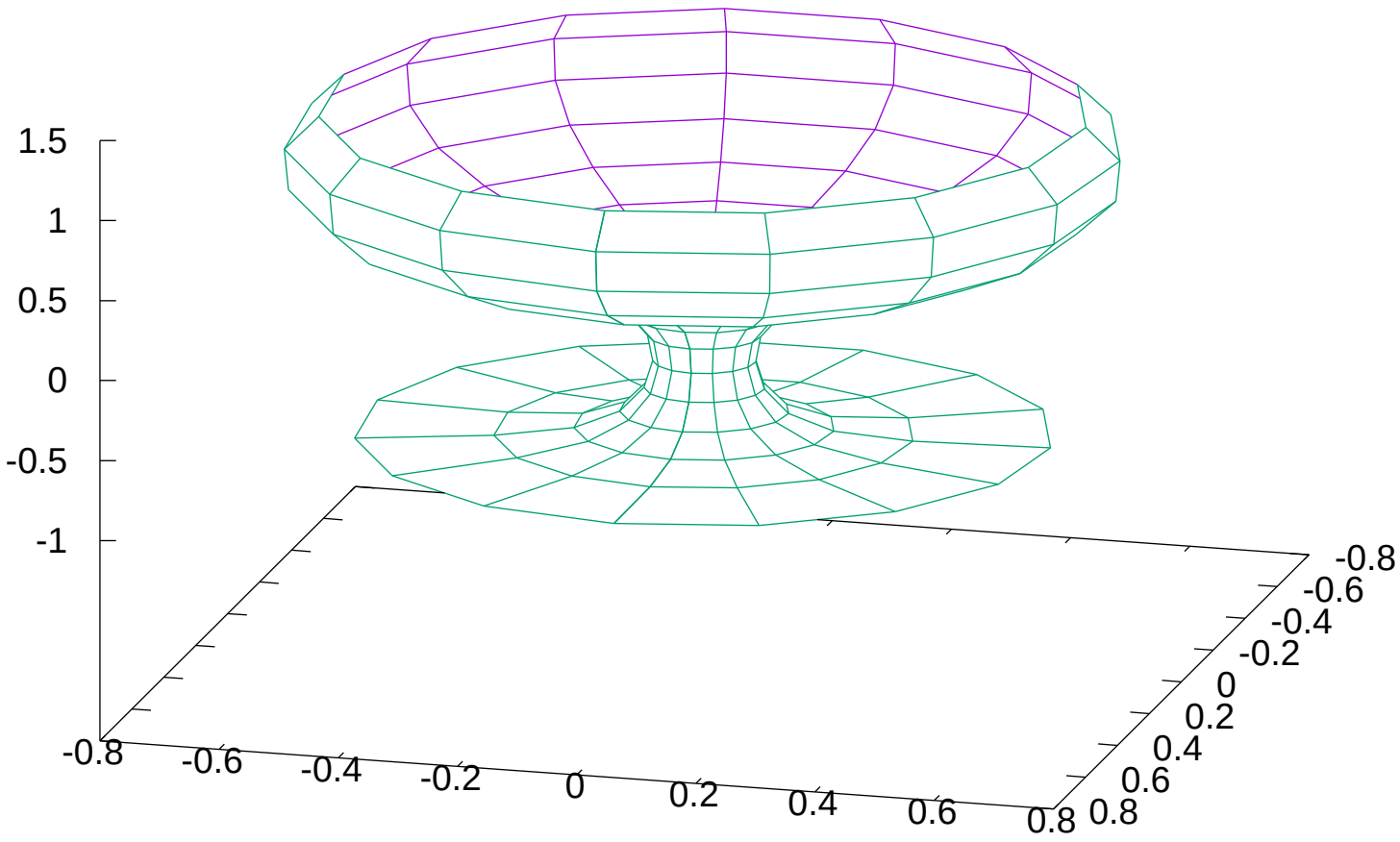

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

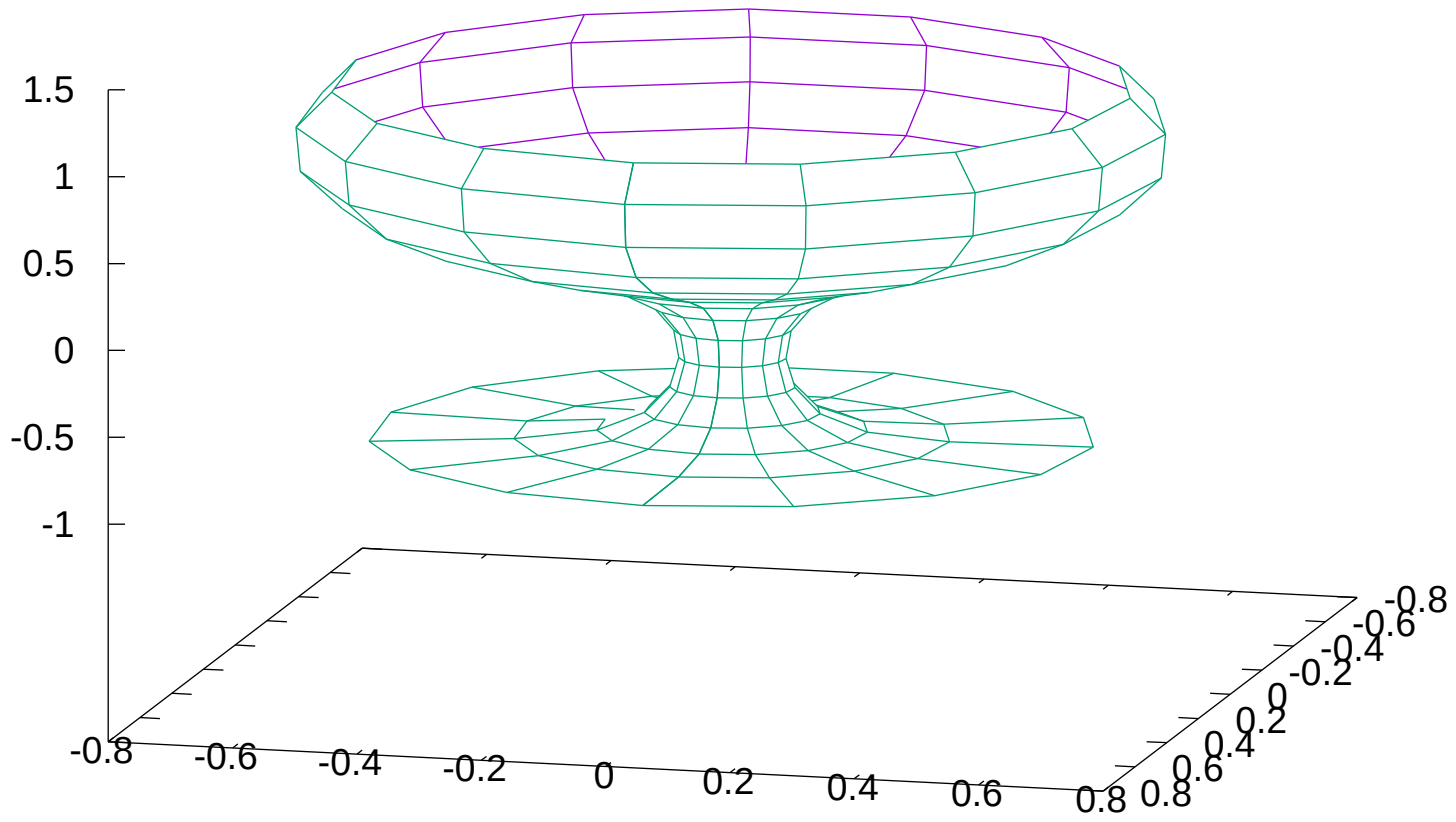

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

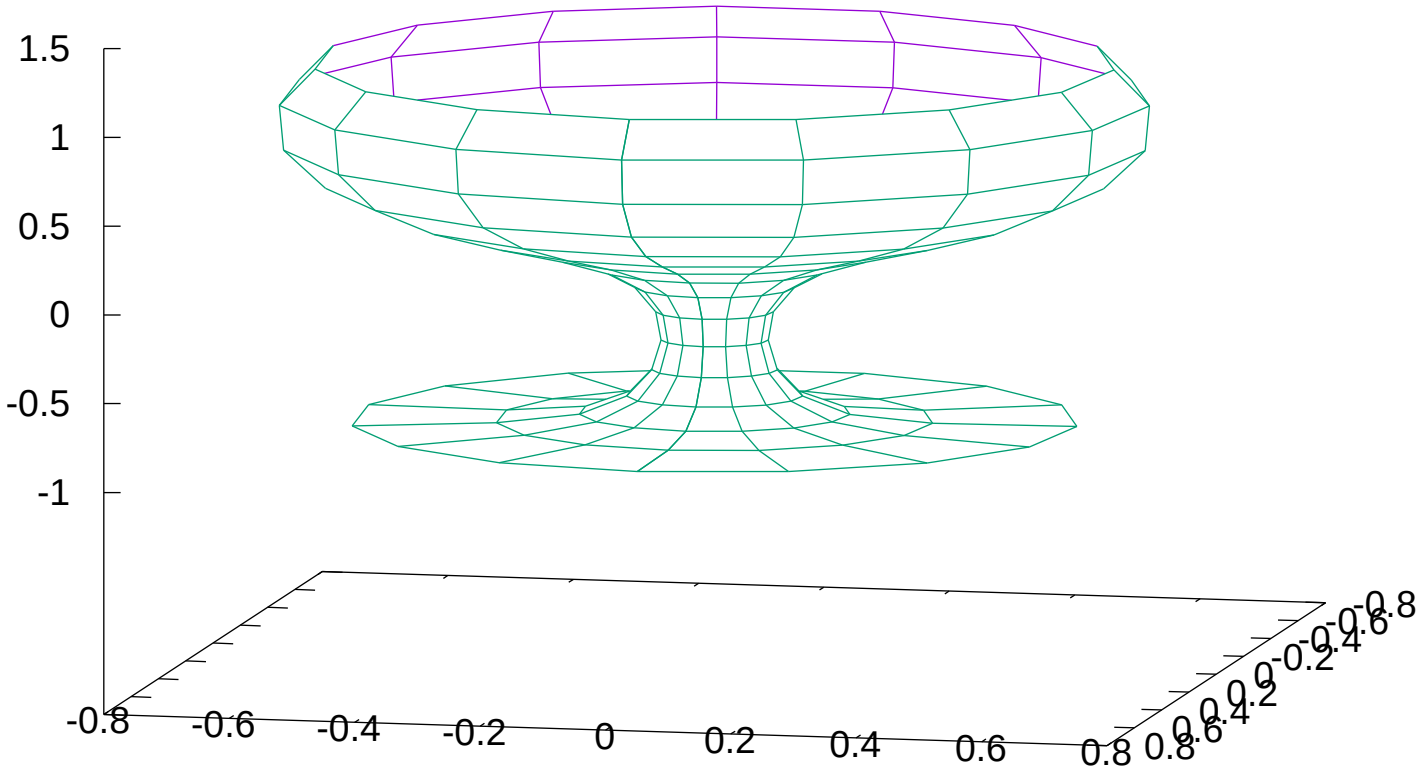

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

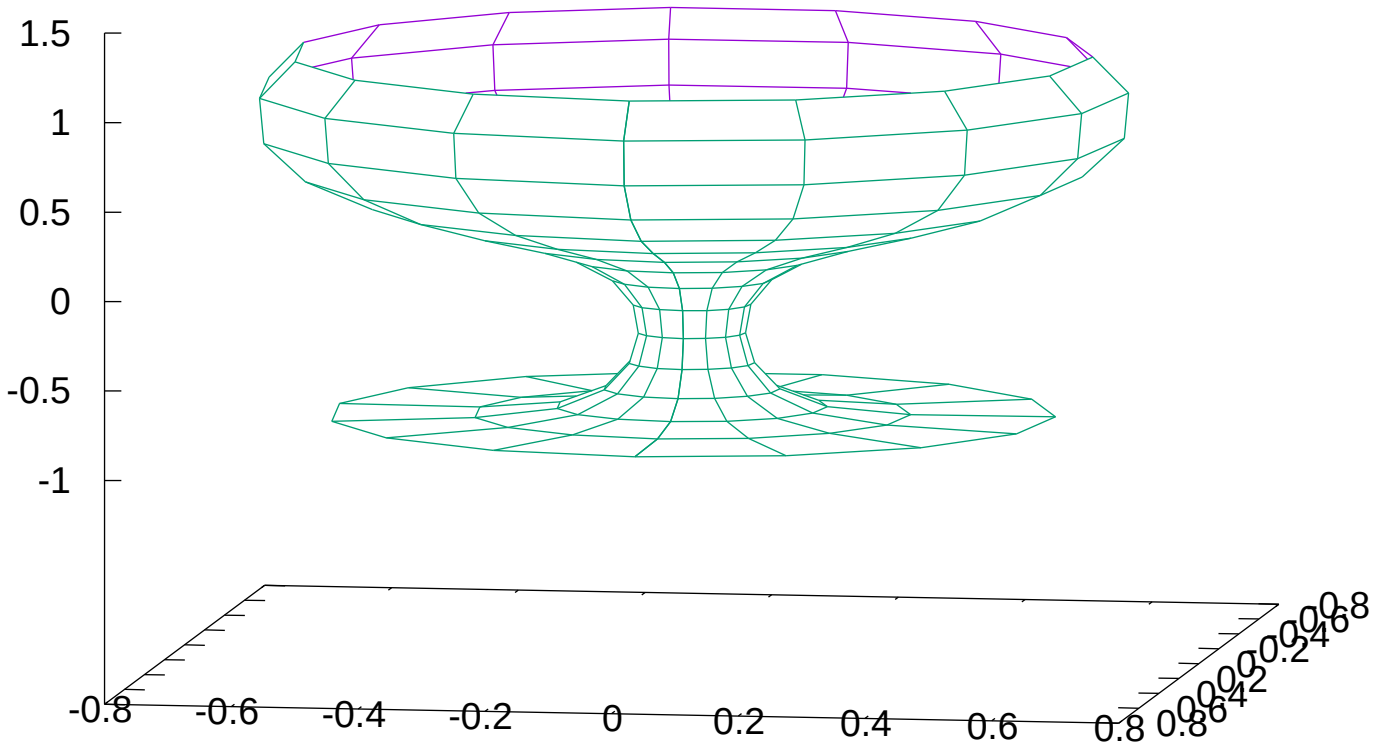

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

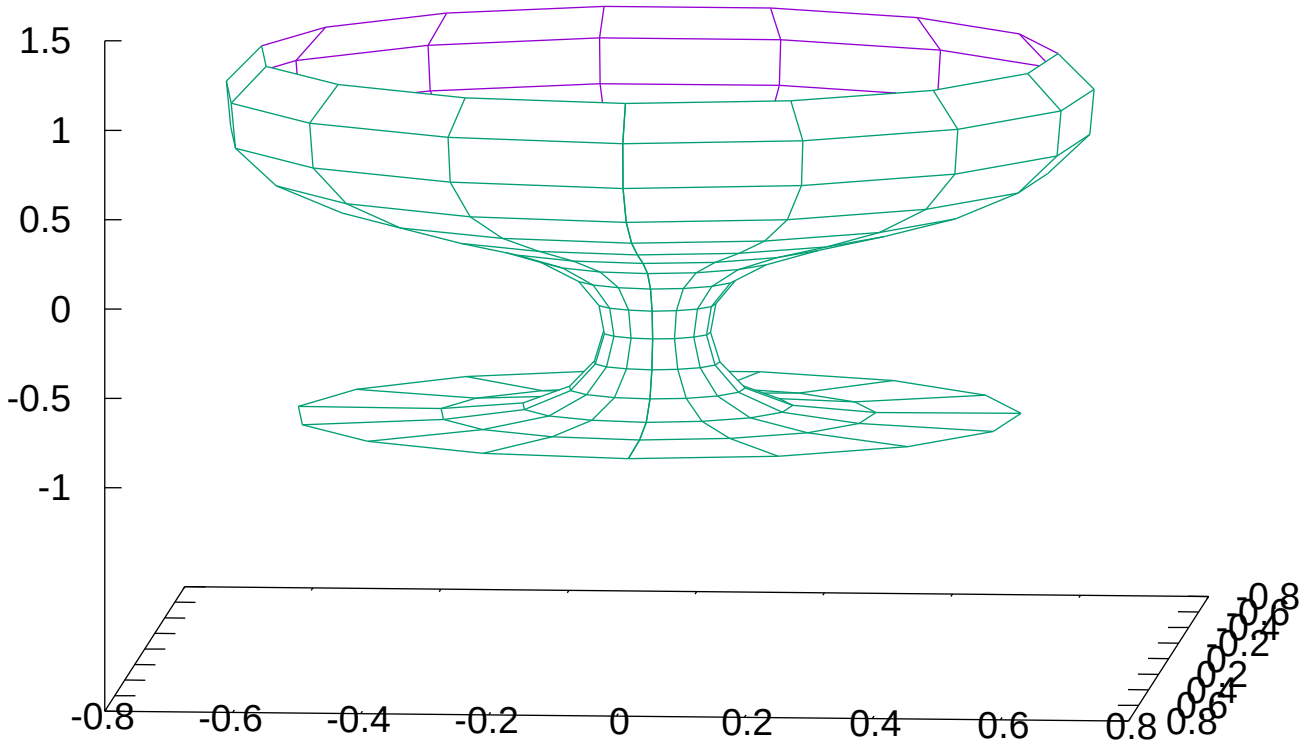

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

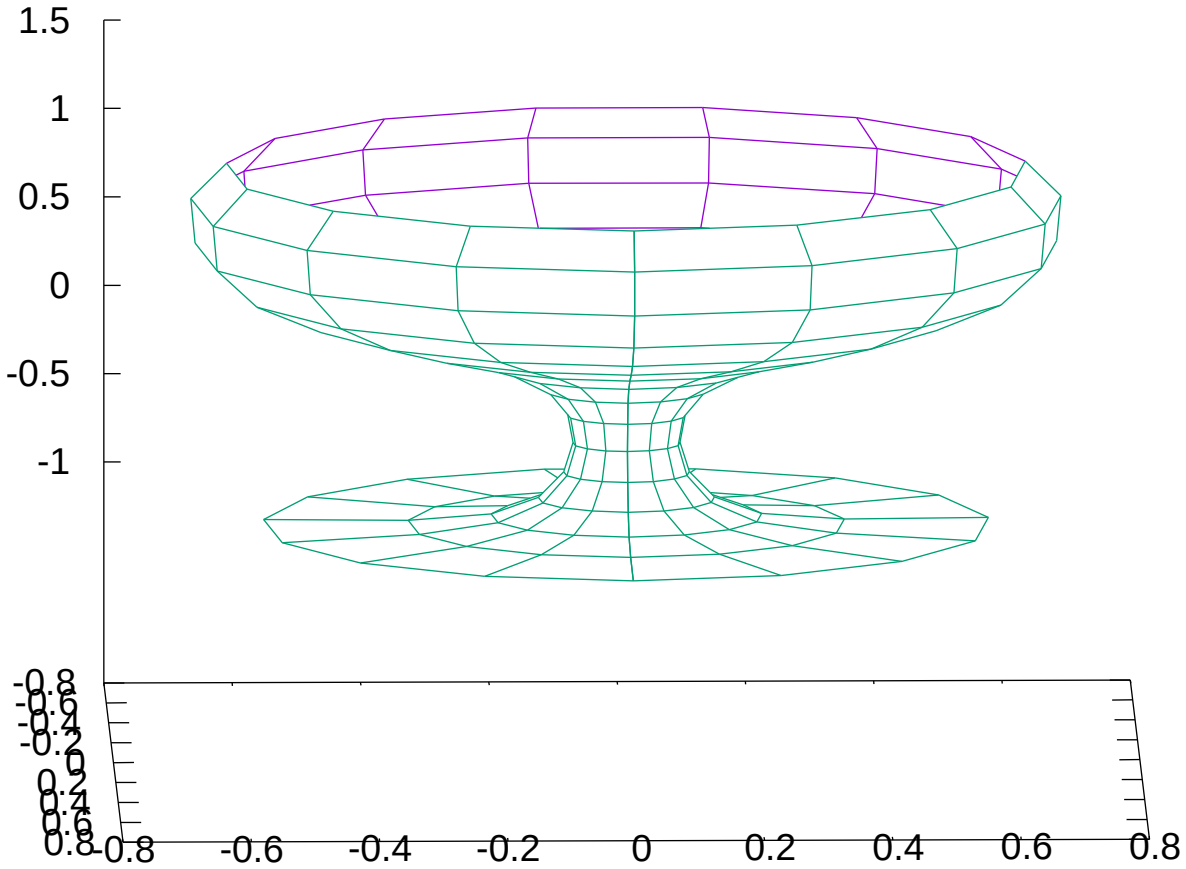

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

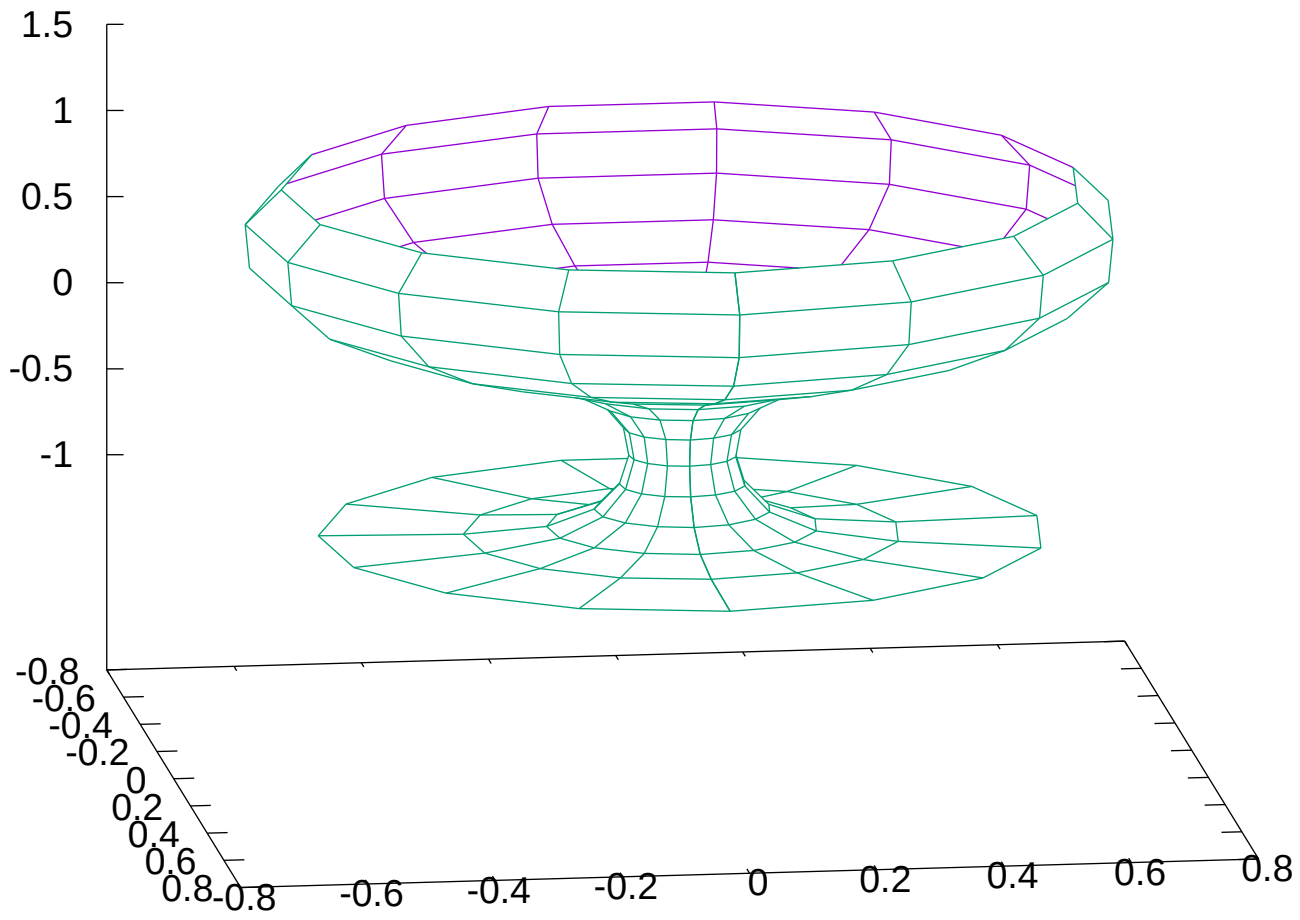

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

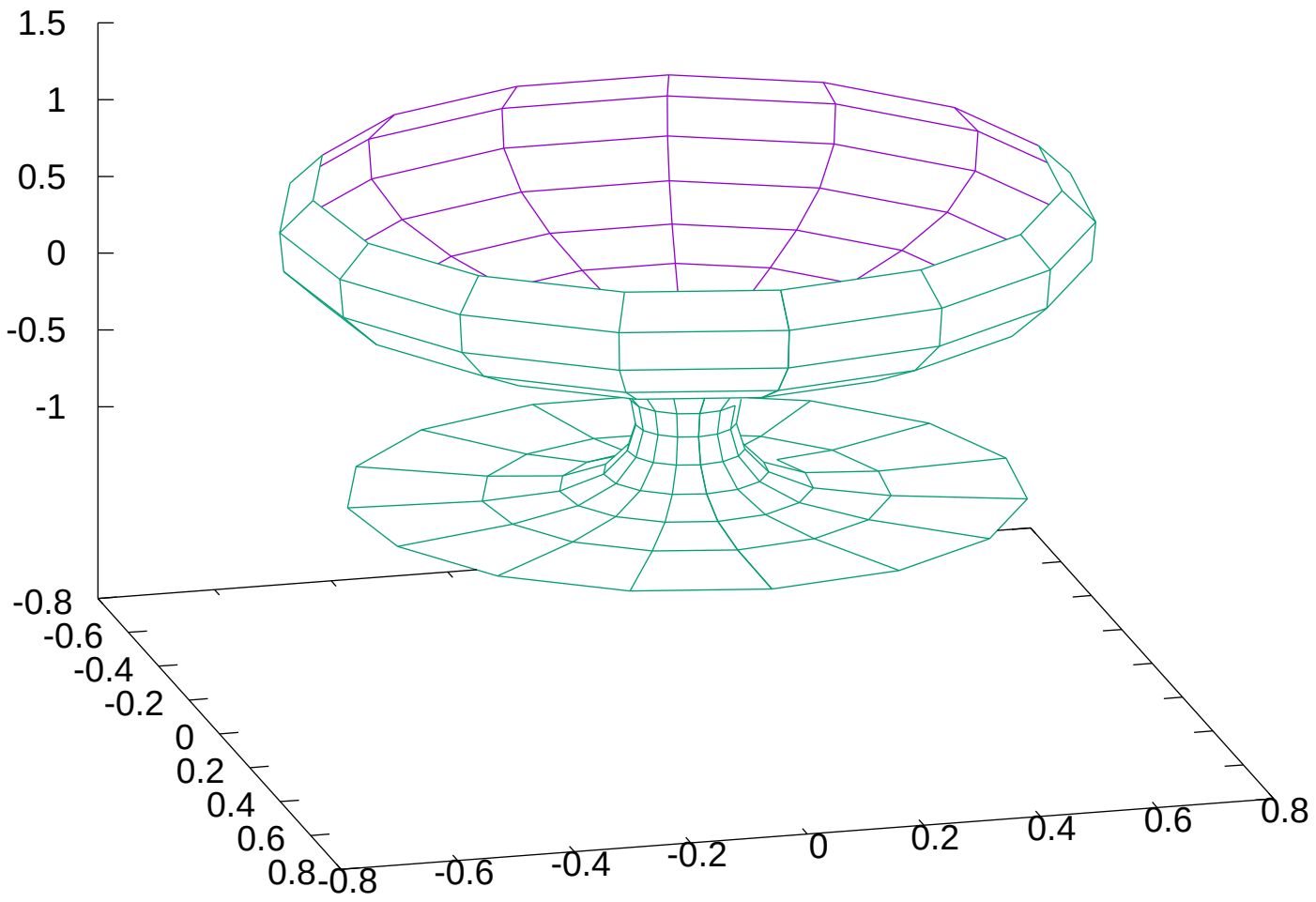

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

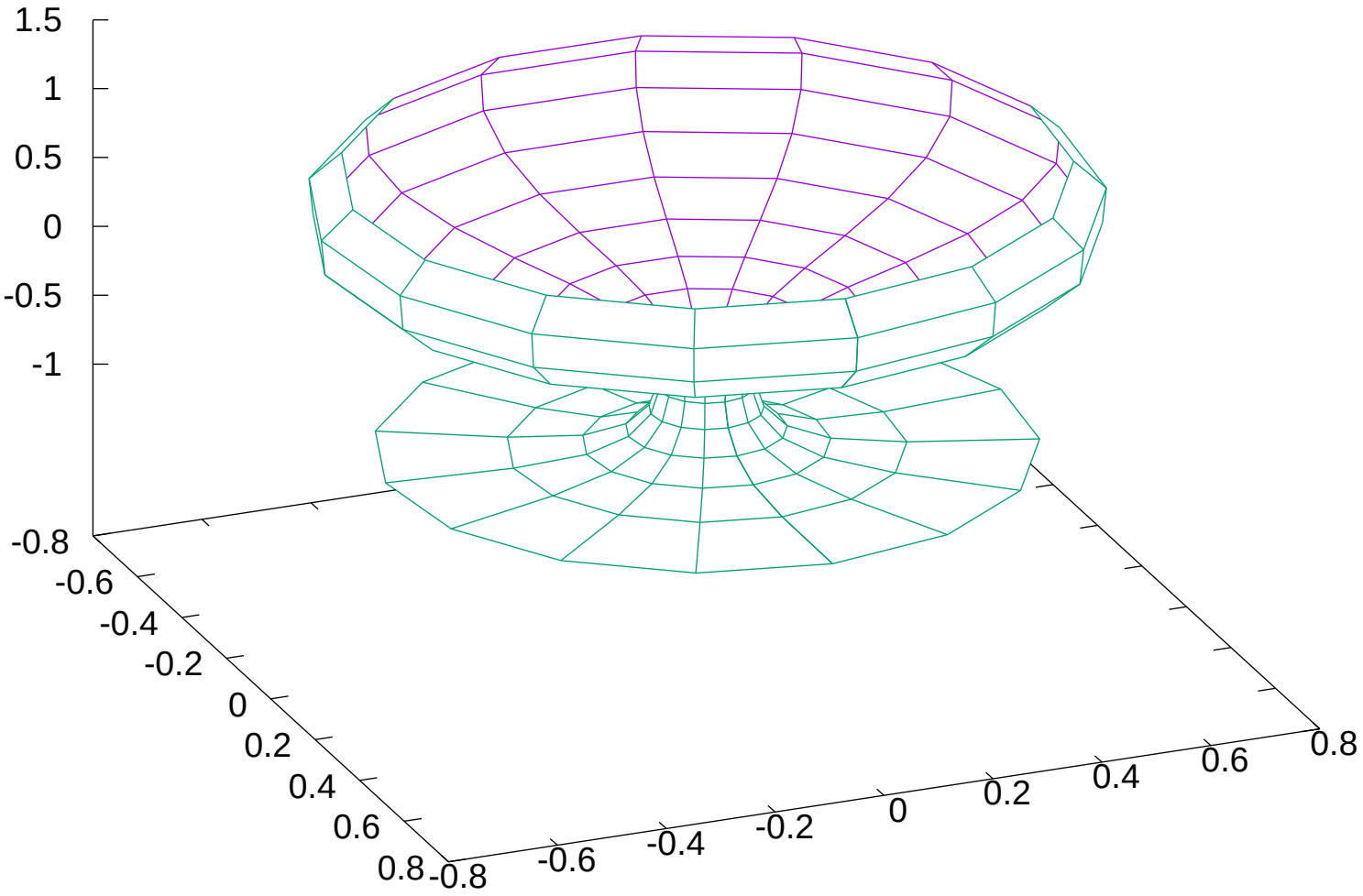

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8

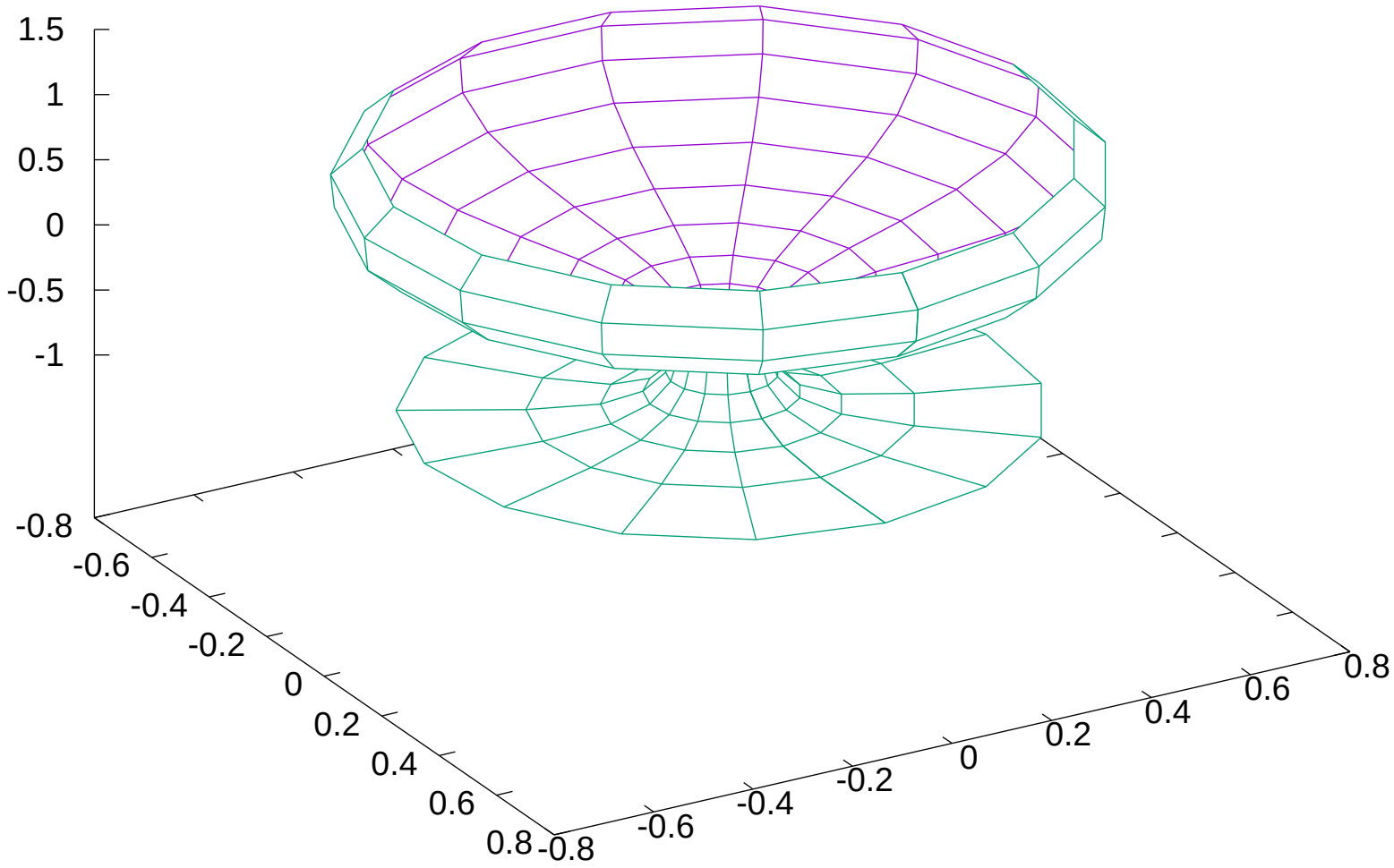

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0.8

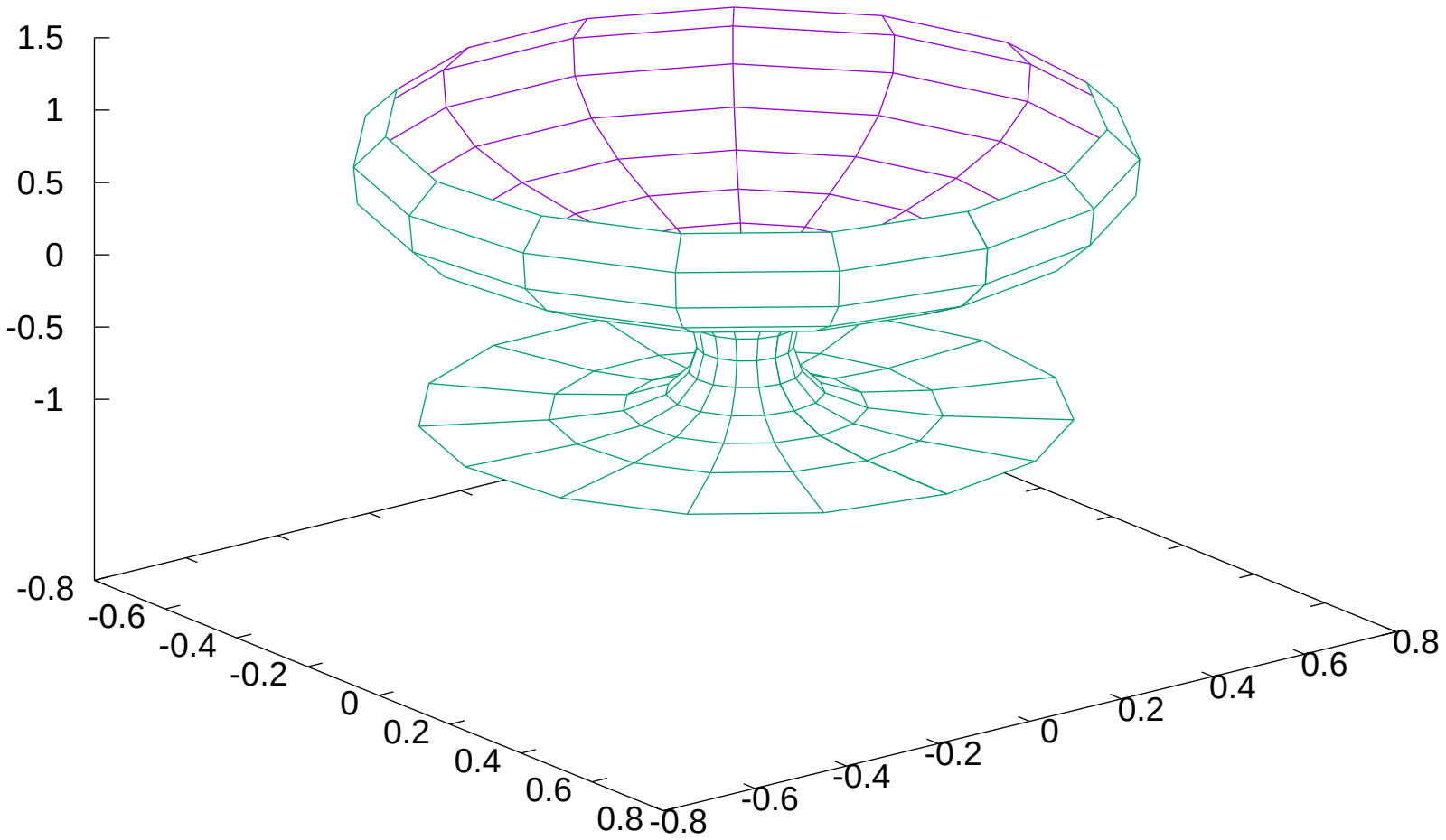

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

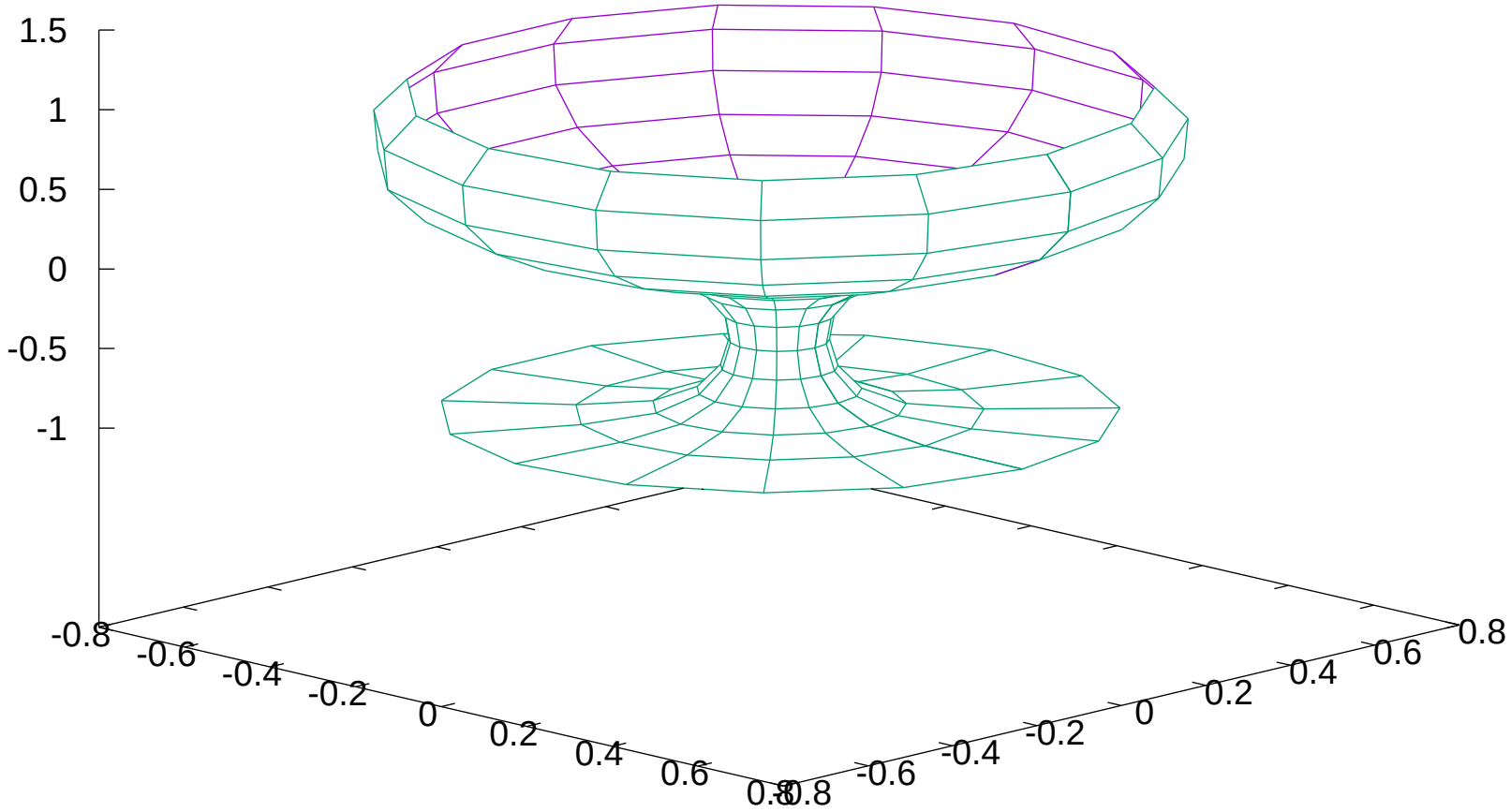

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

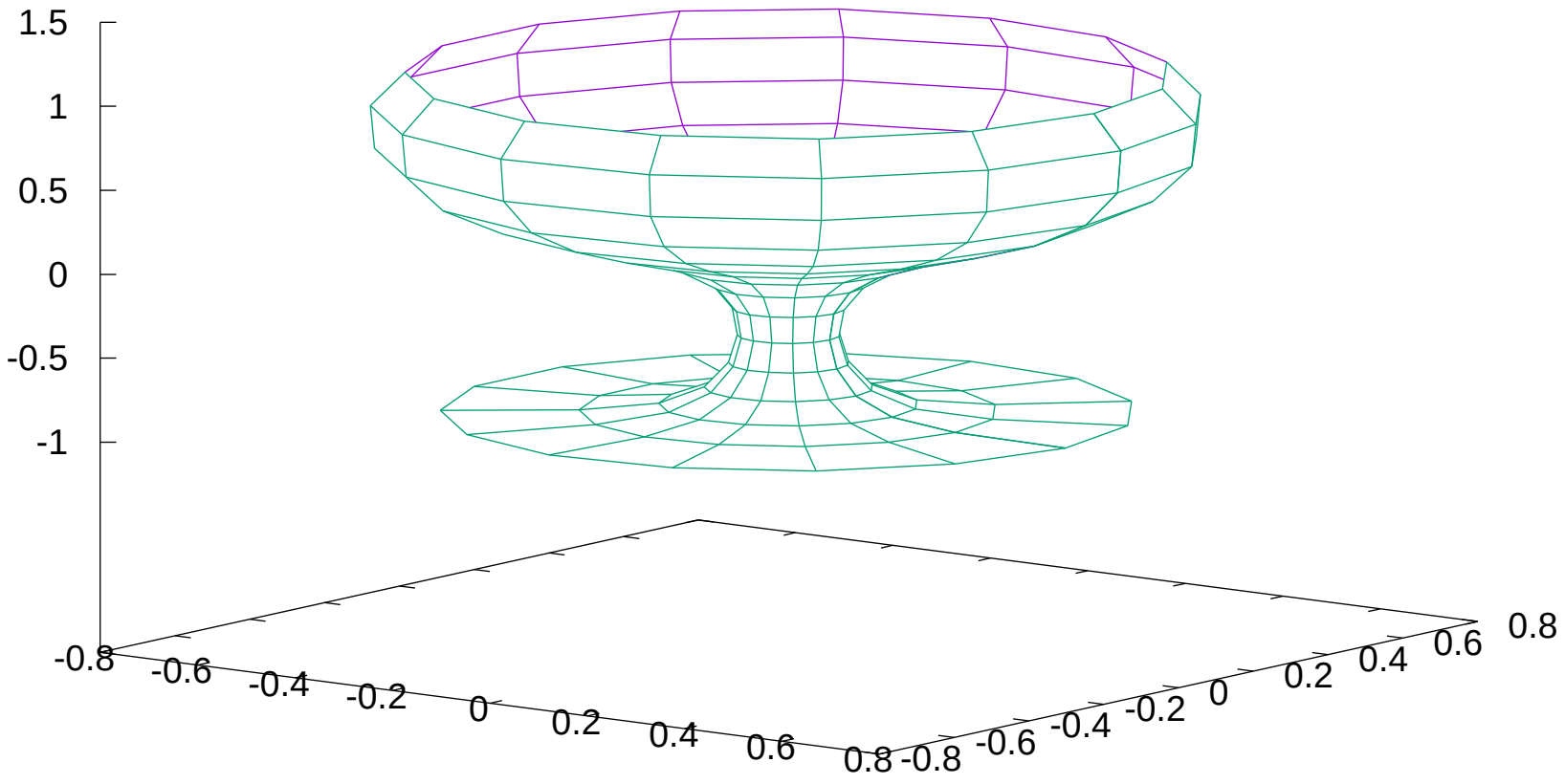

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

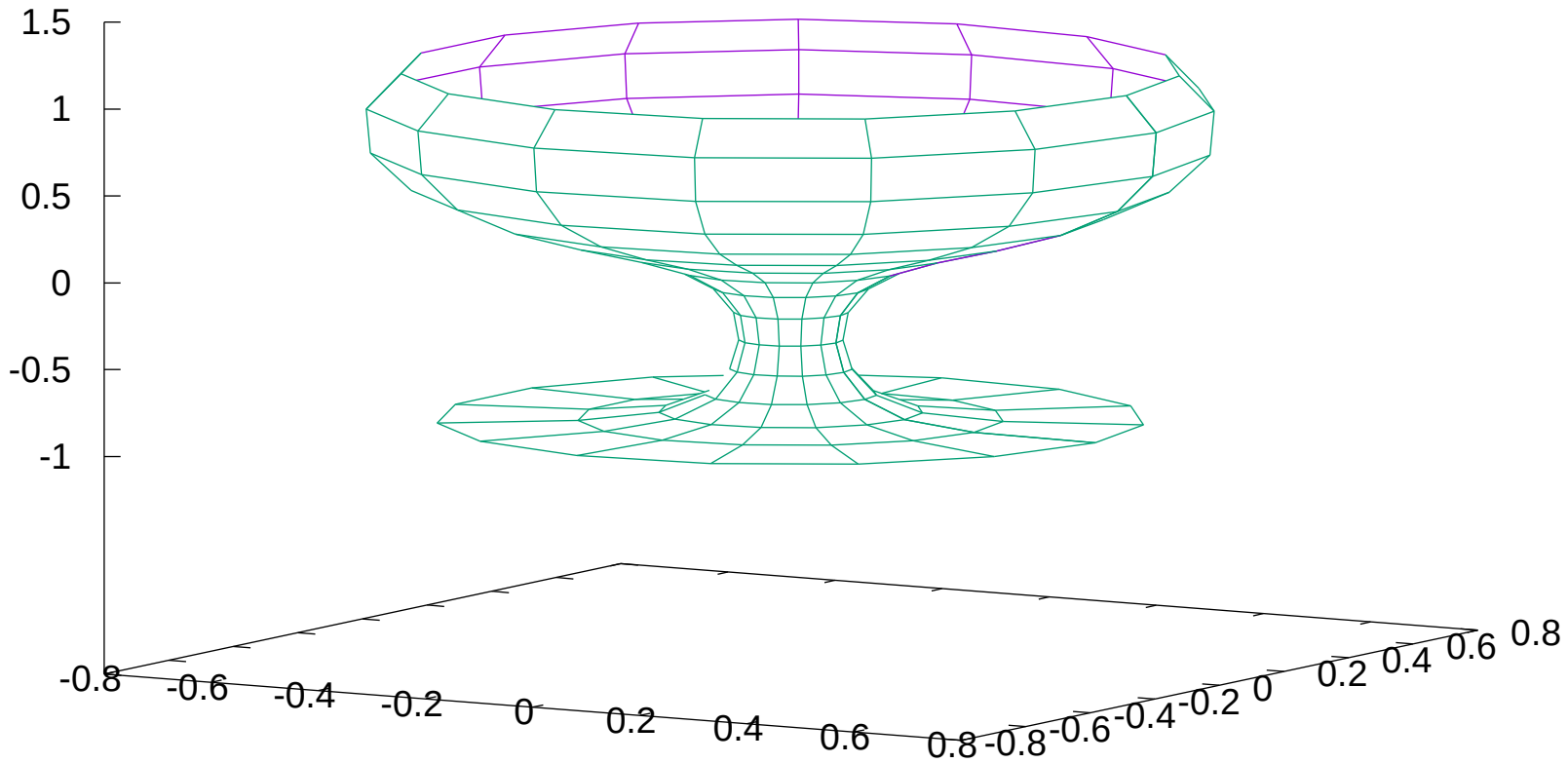

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

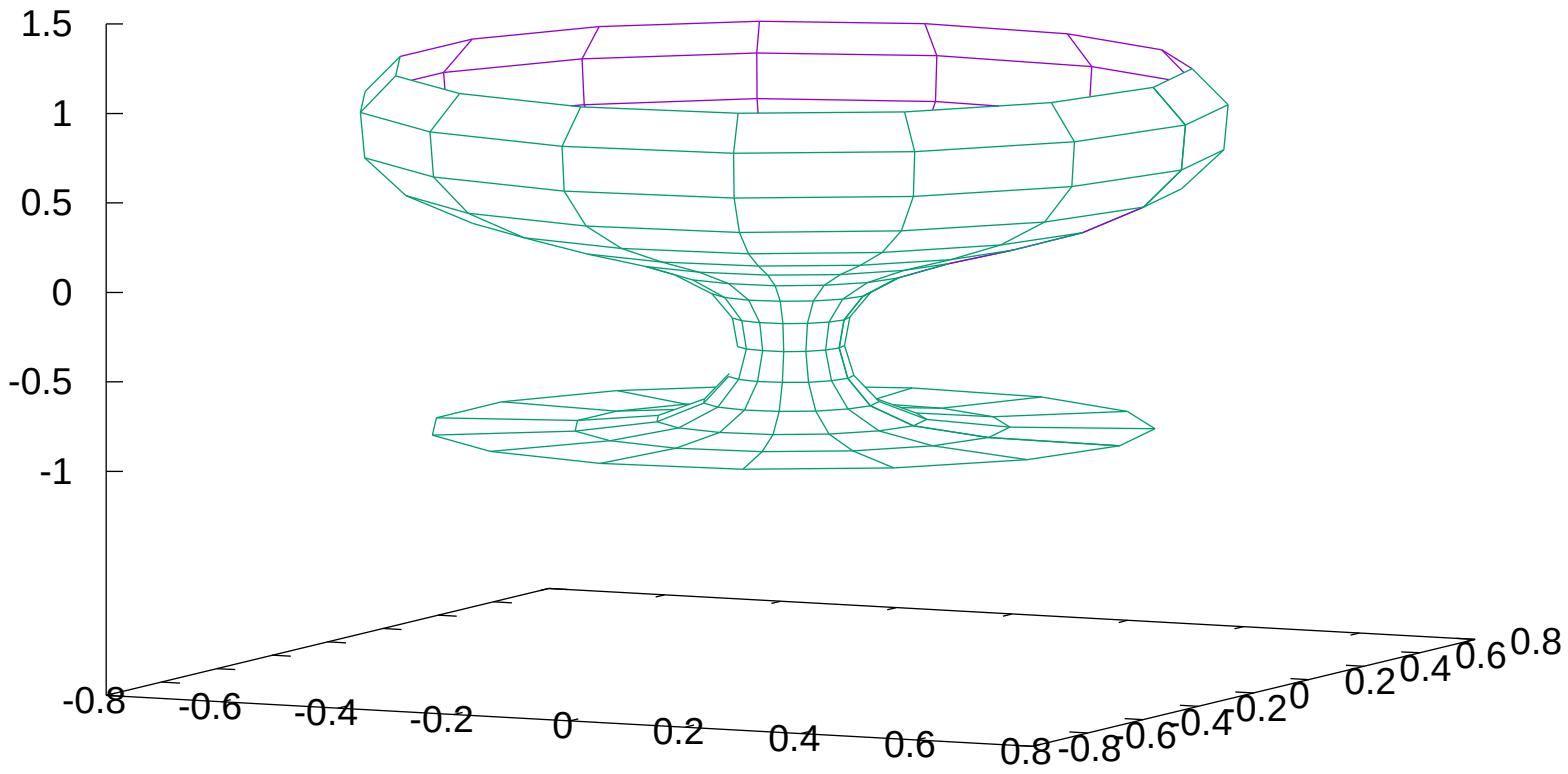

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

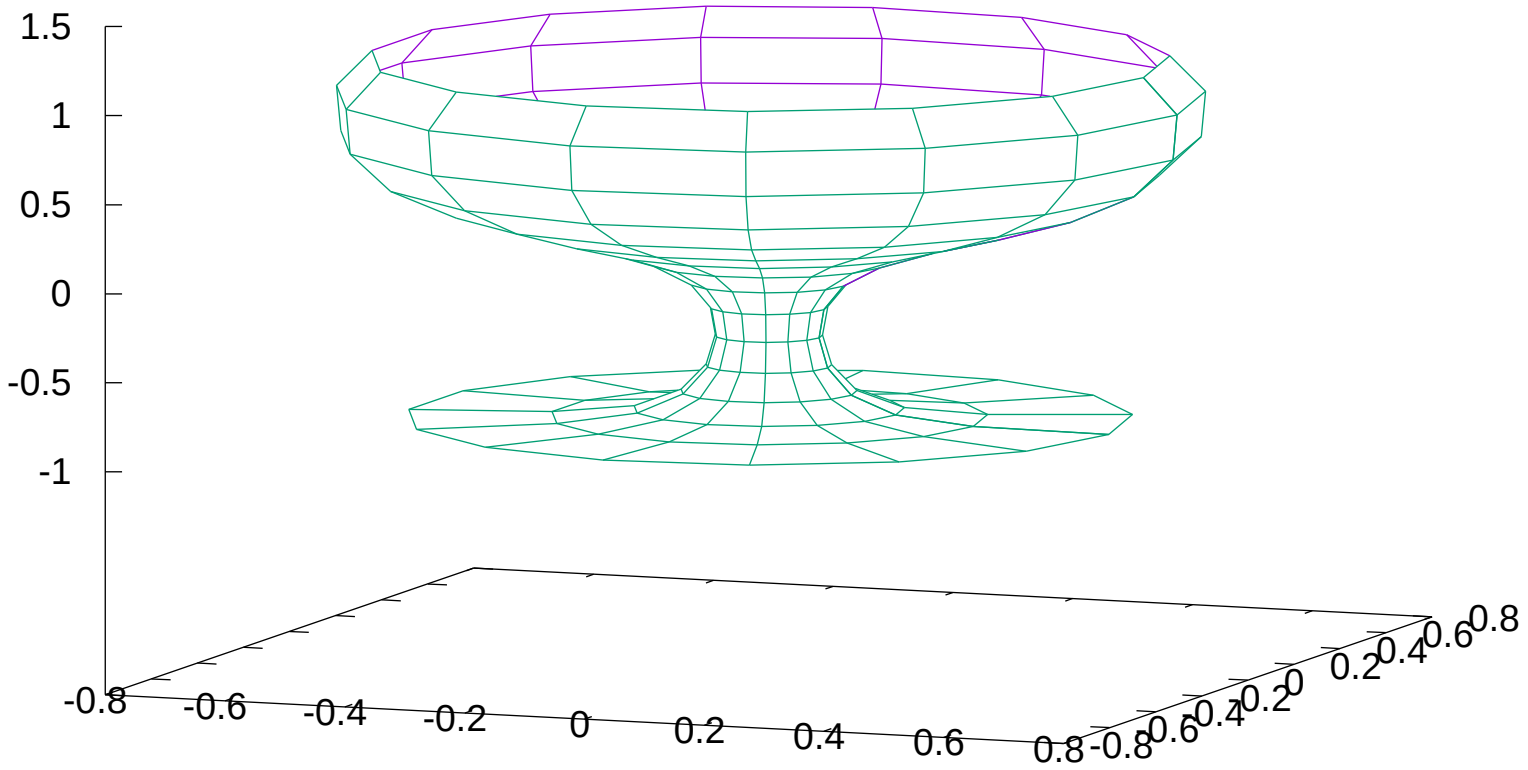

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

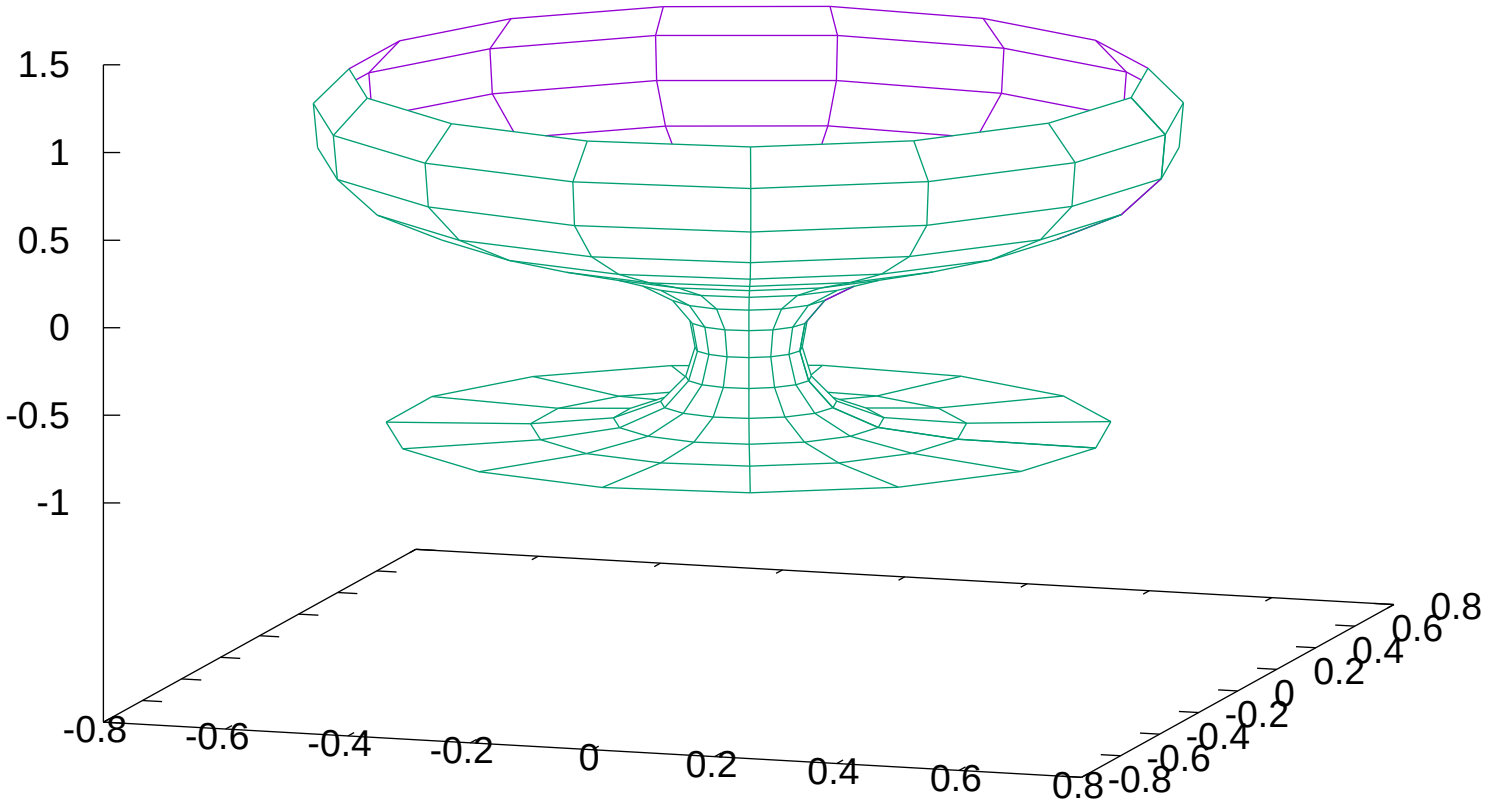

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

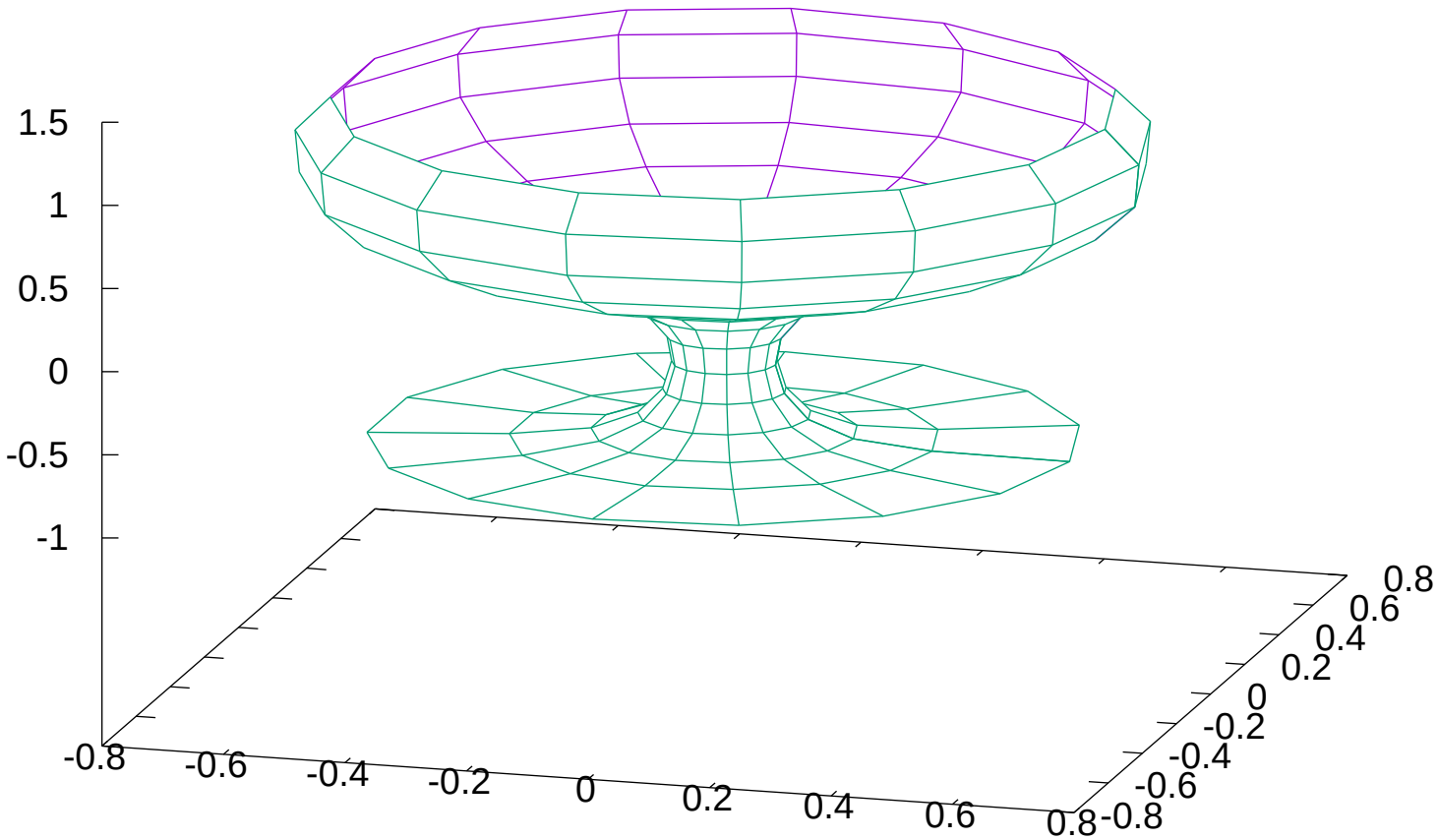

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

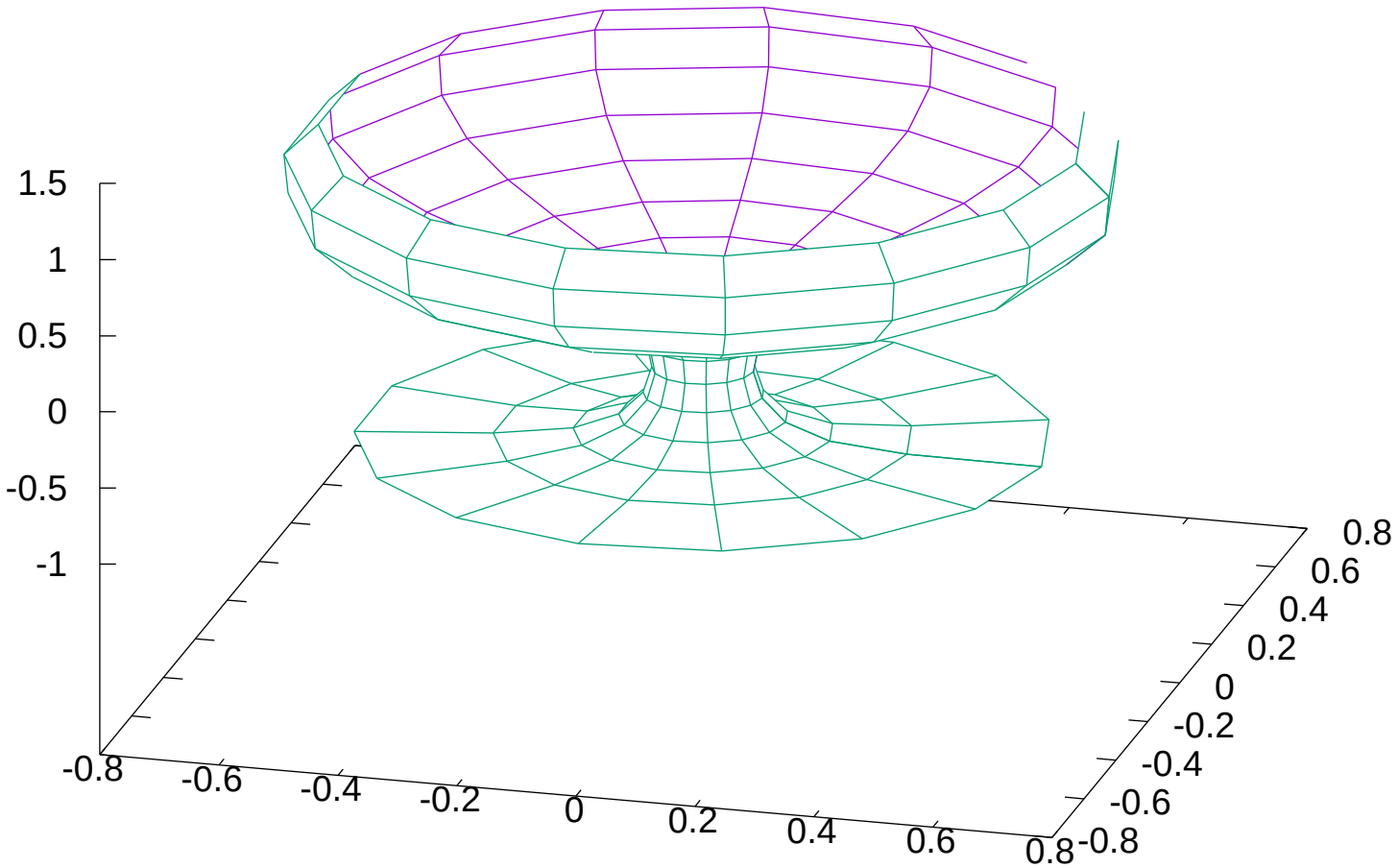

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

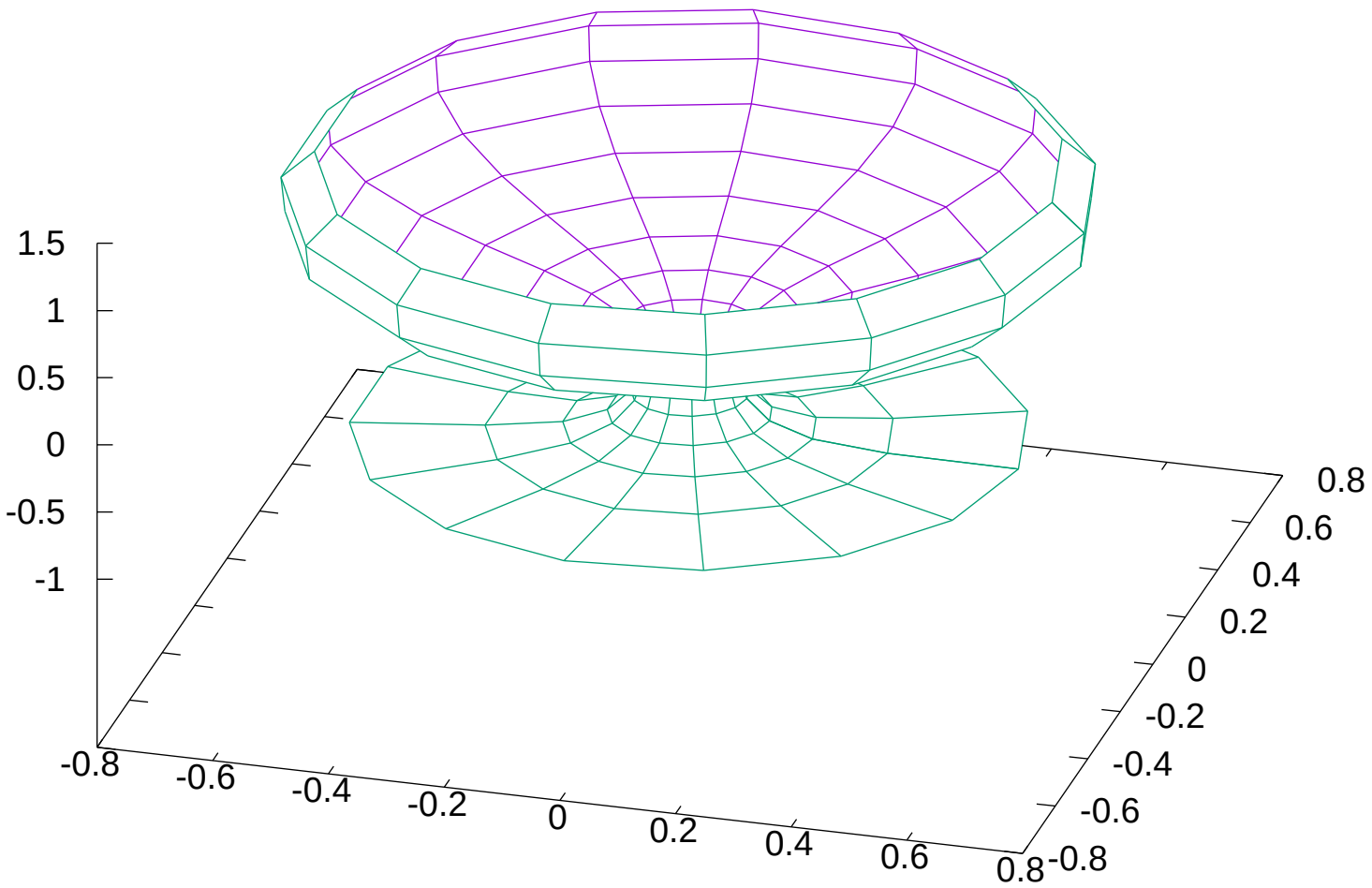

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

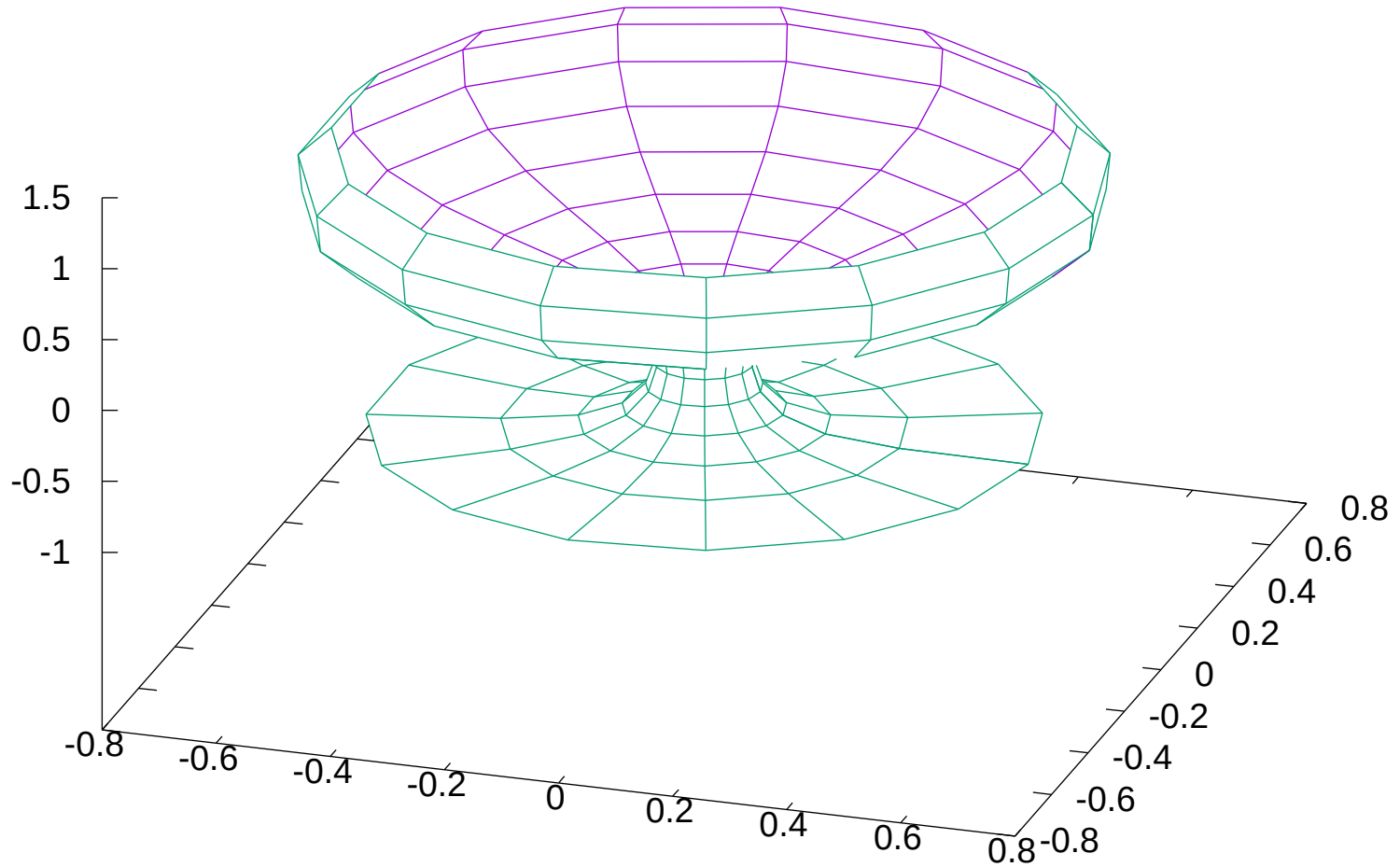

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

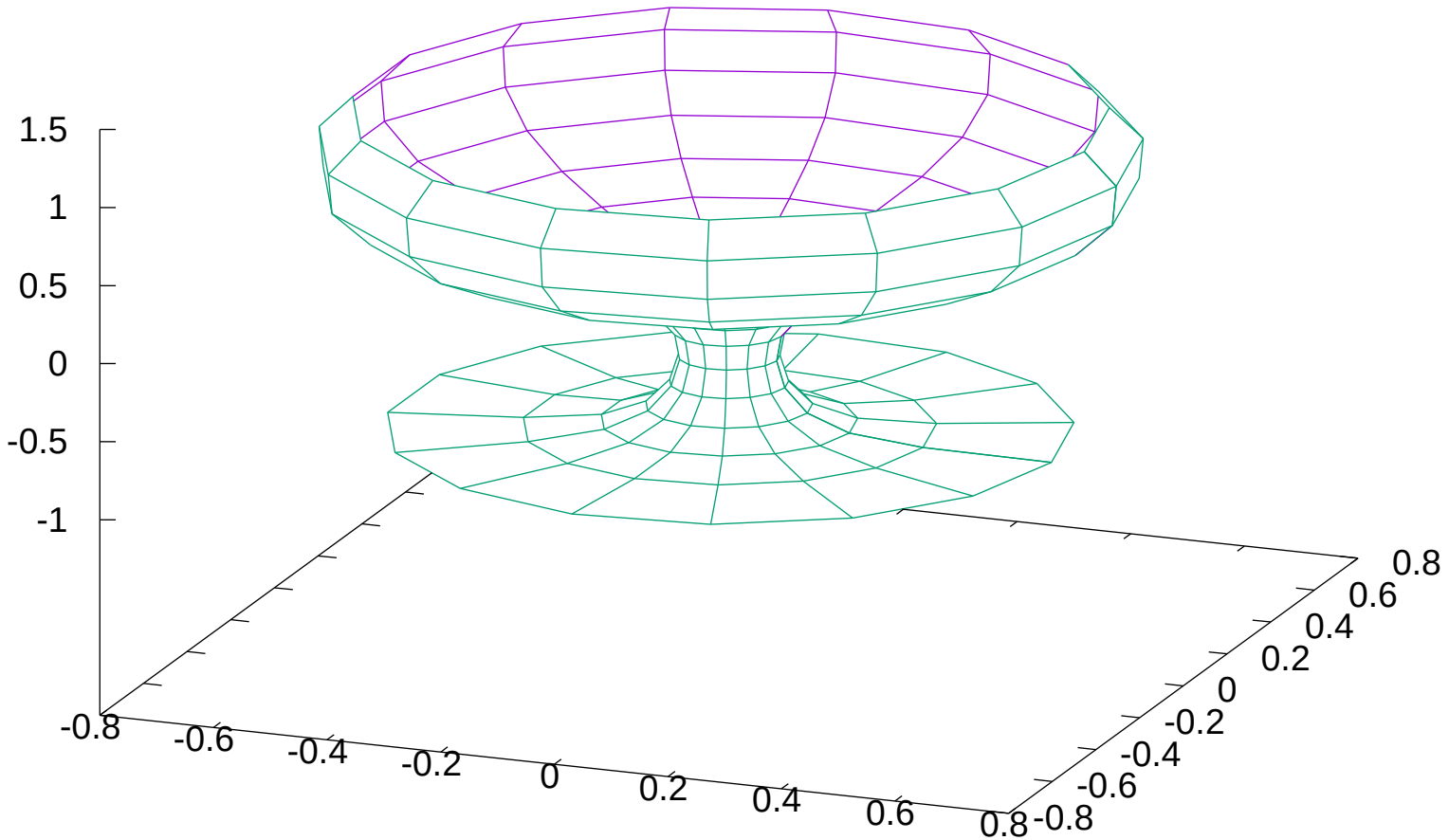

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

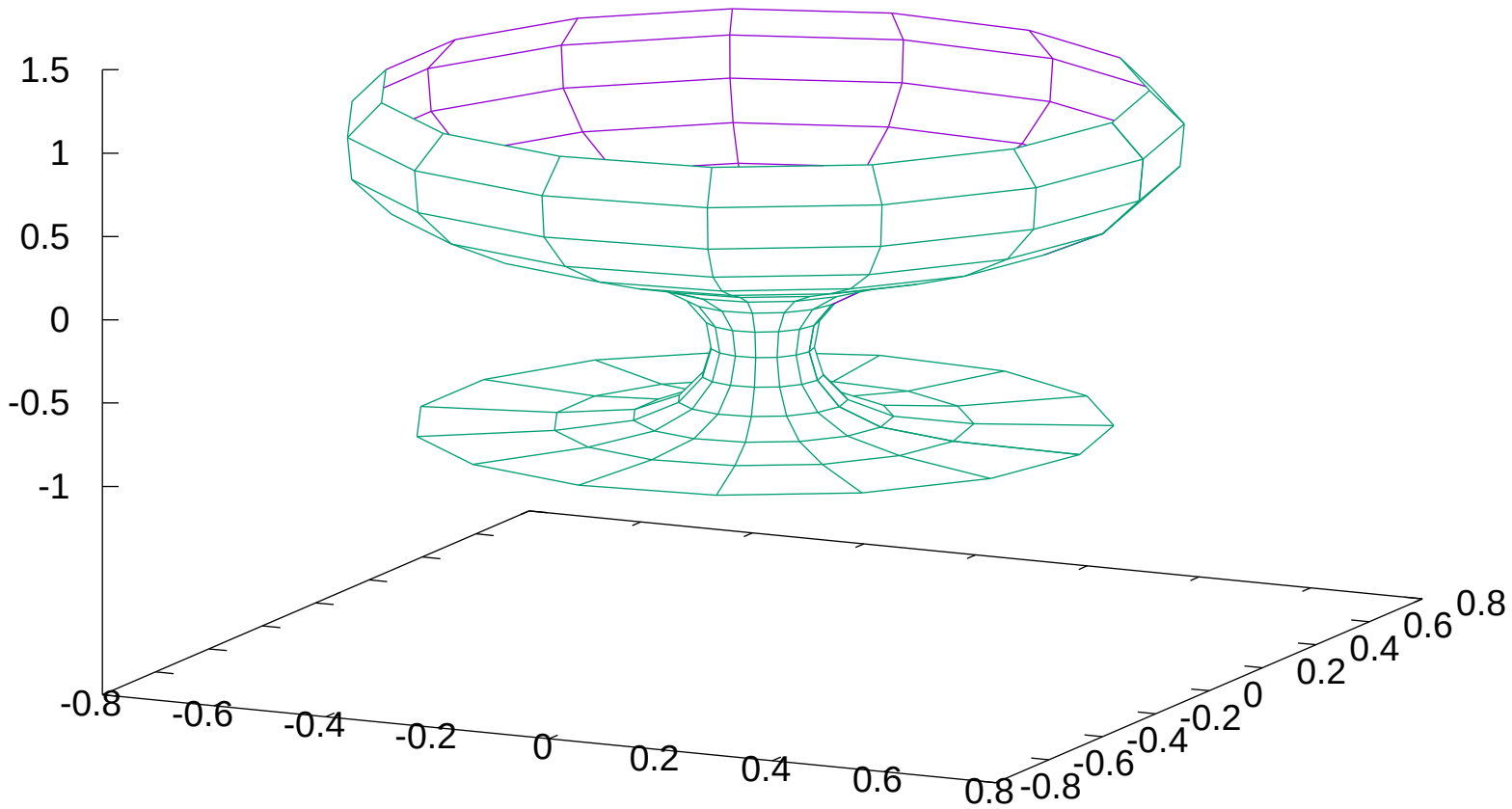

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

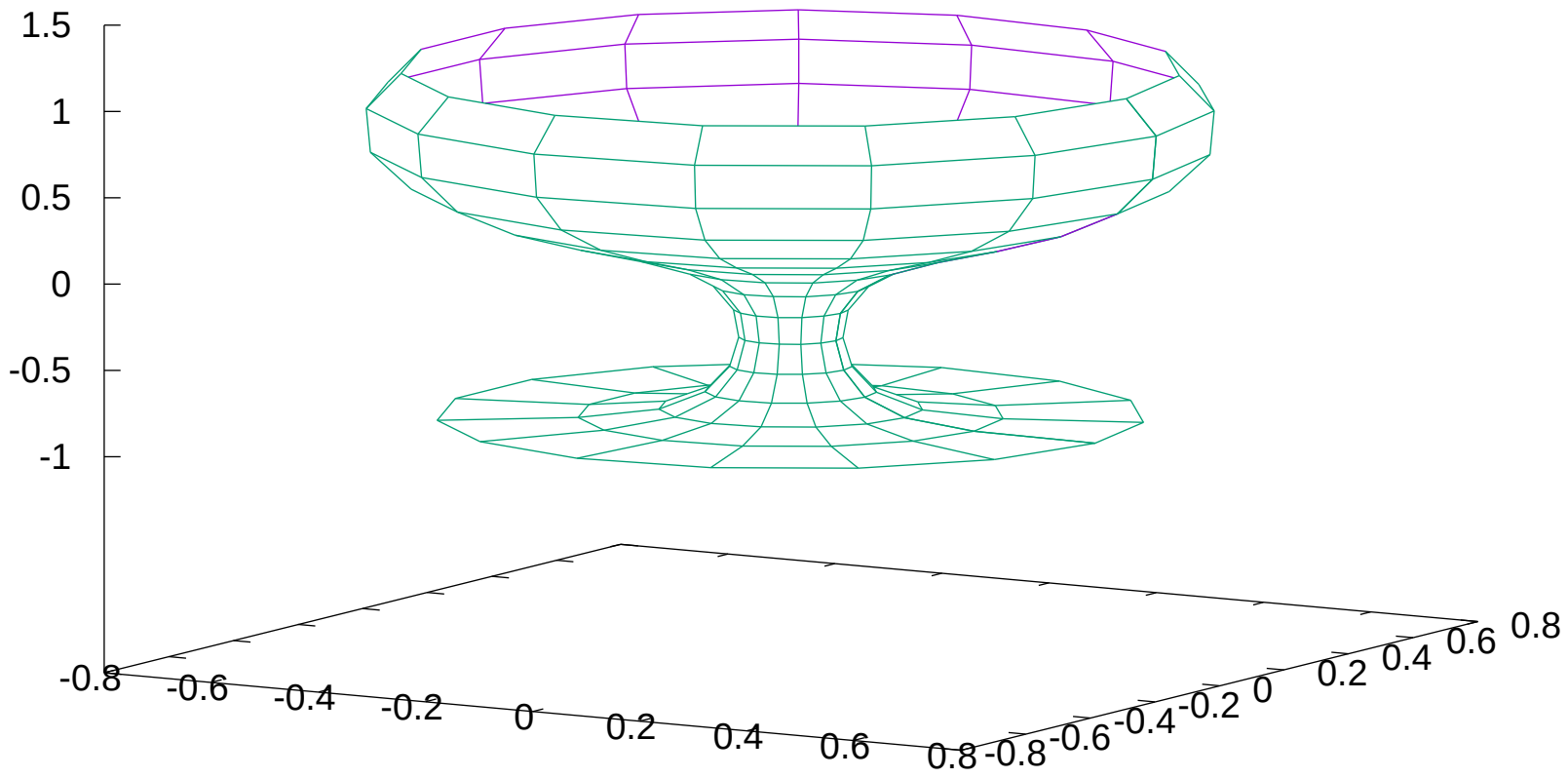

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8

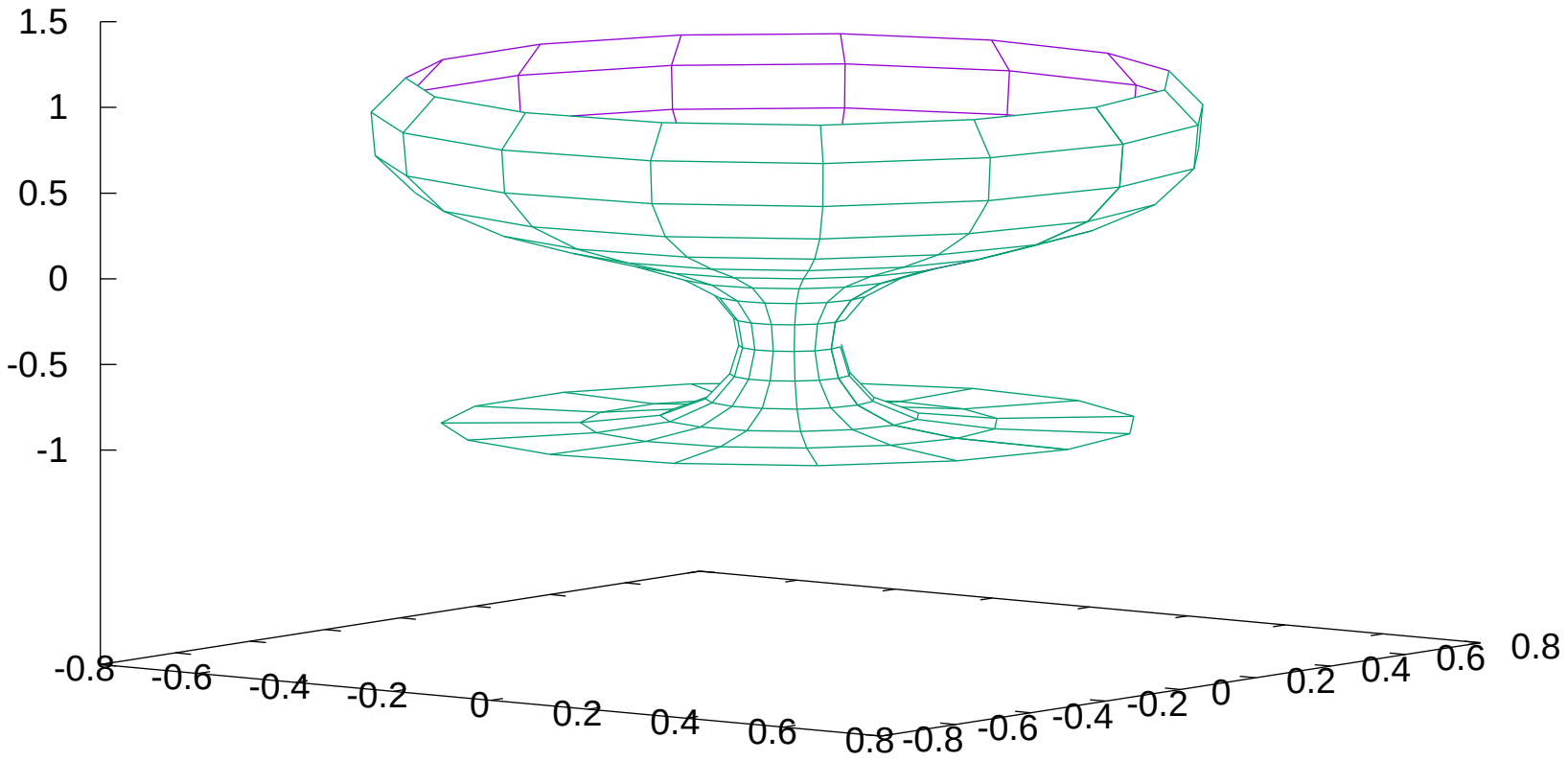

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8

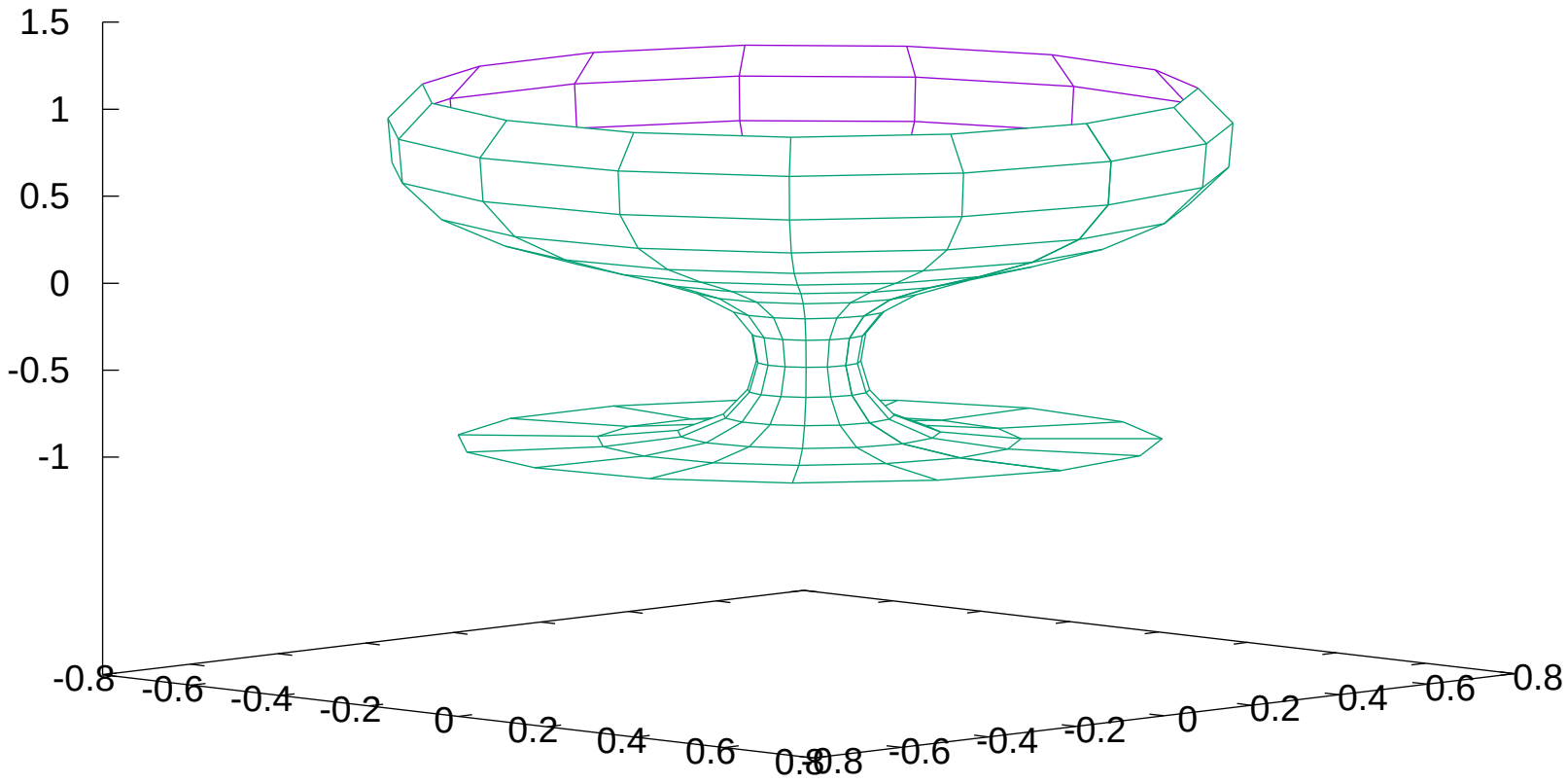

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8

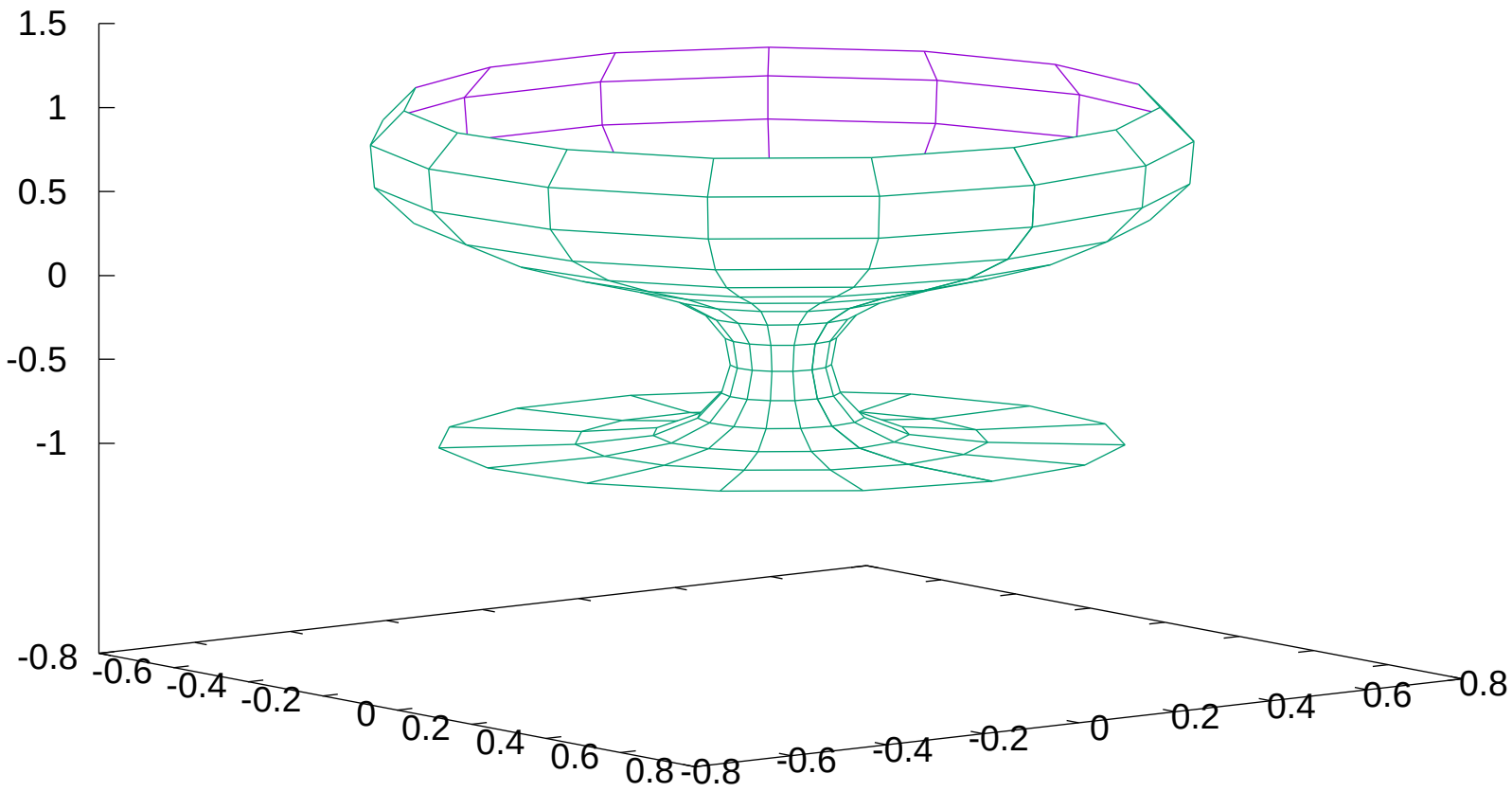

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

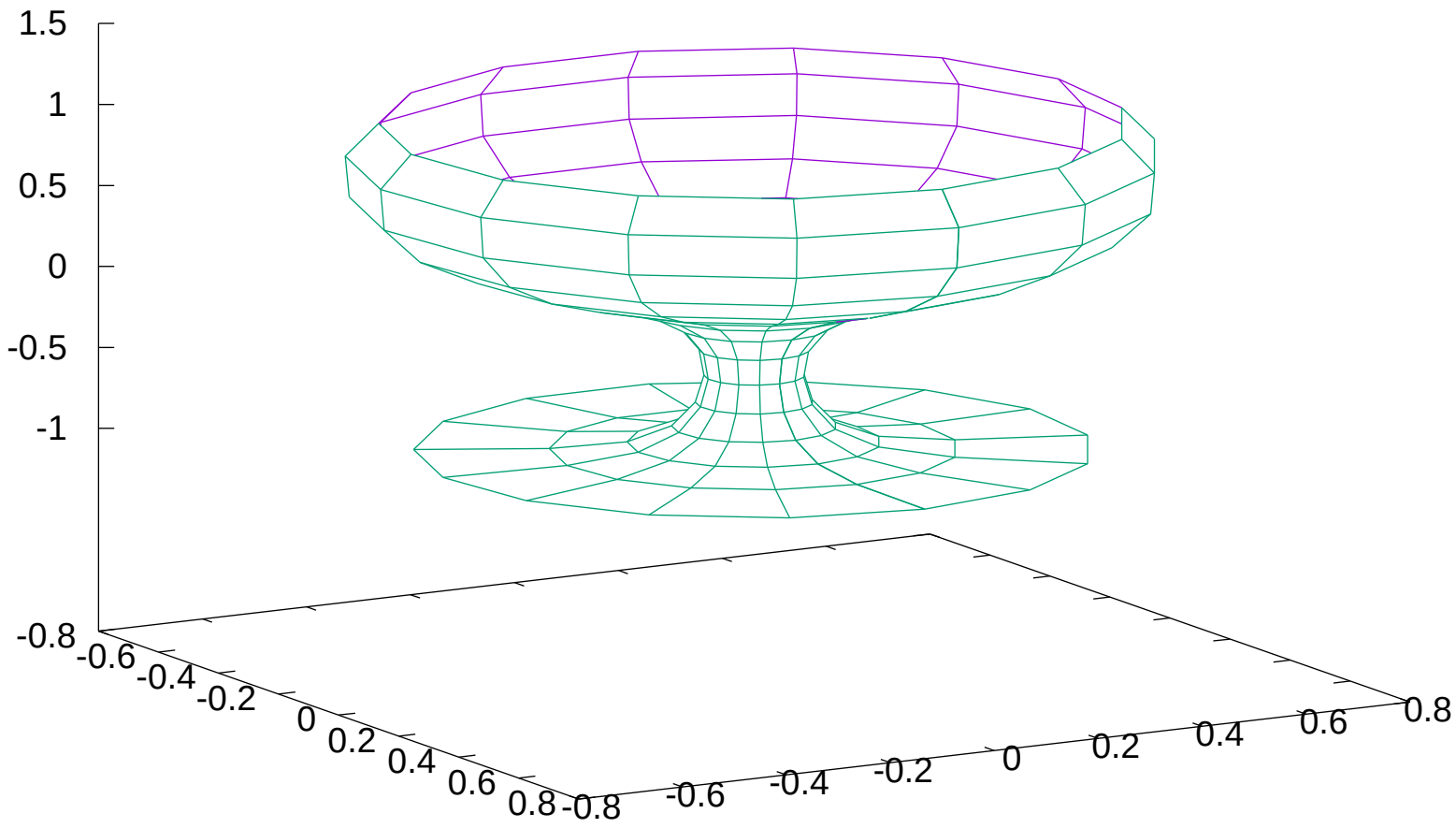

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

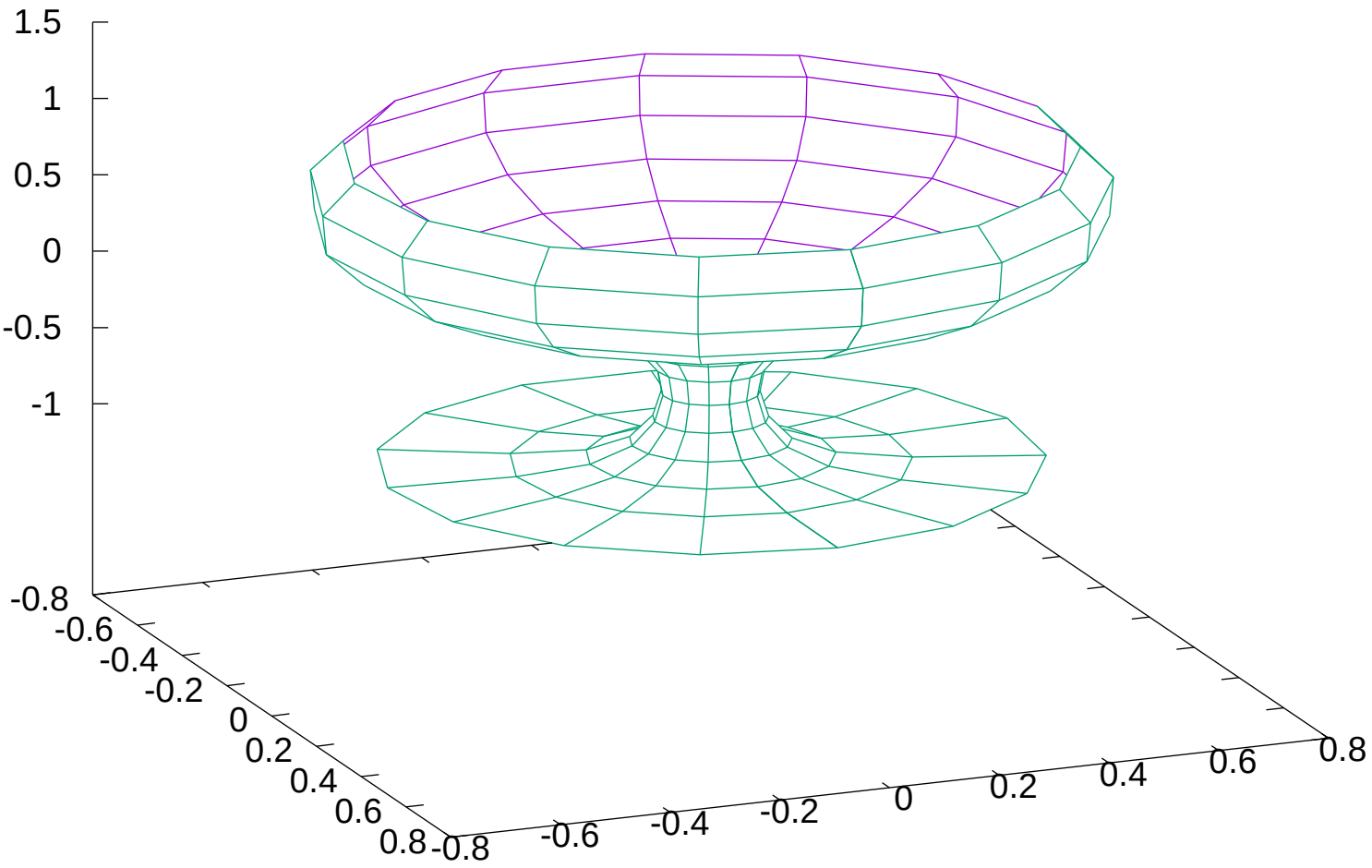

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

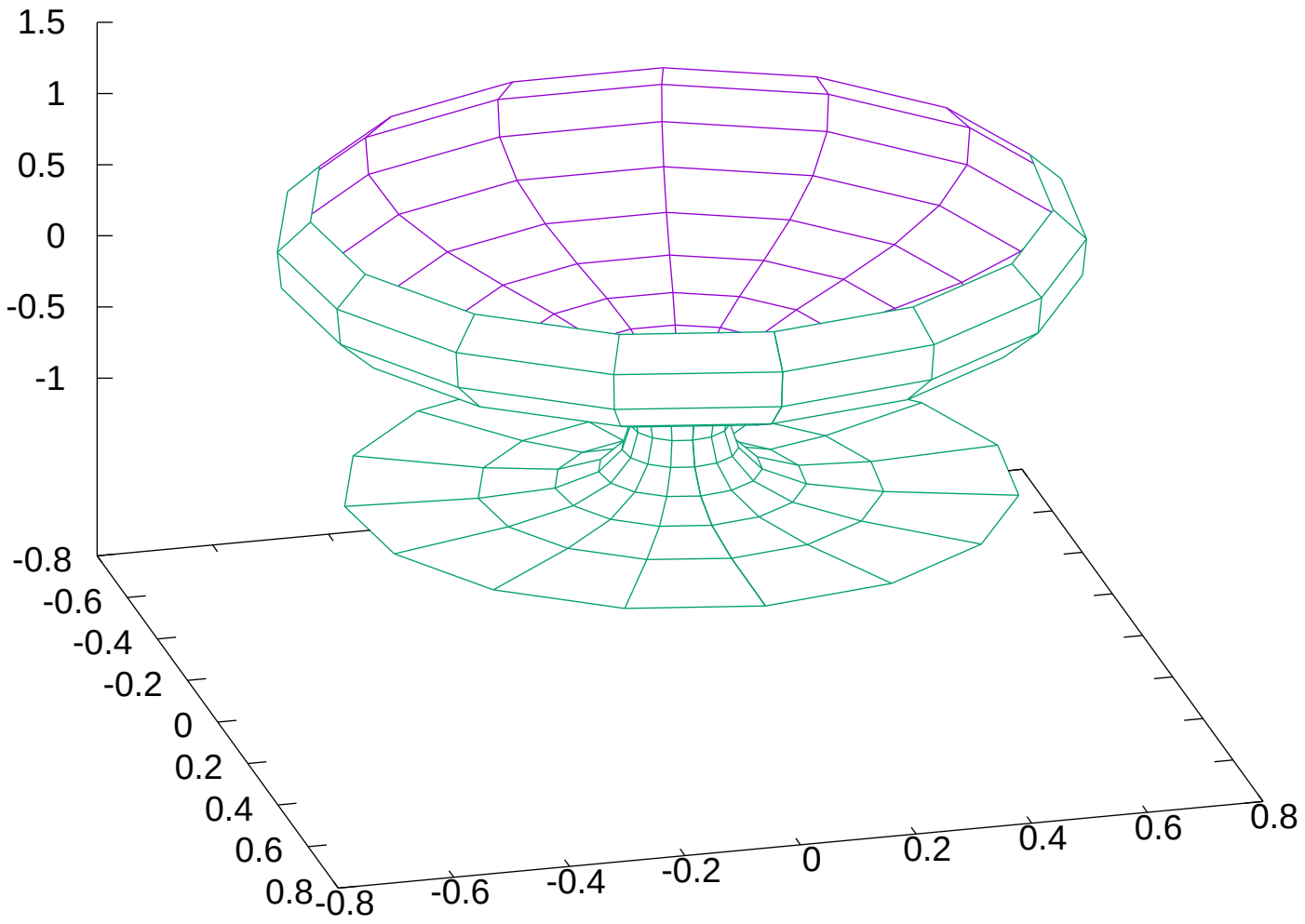

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

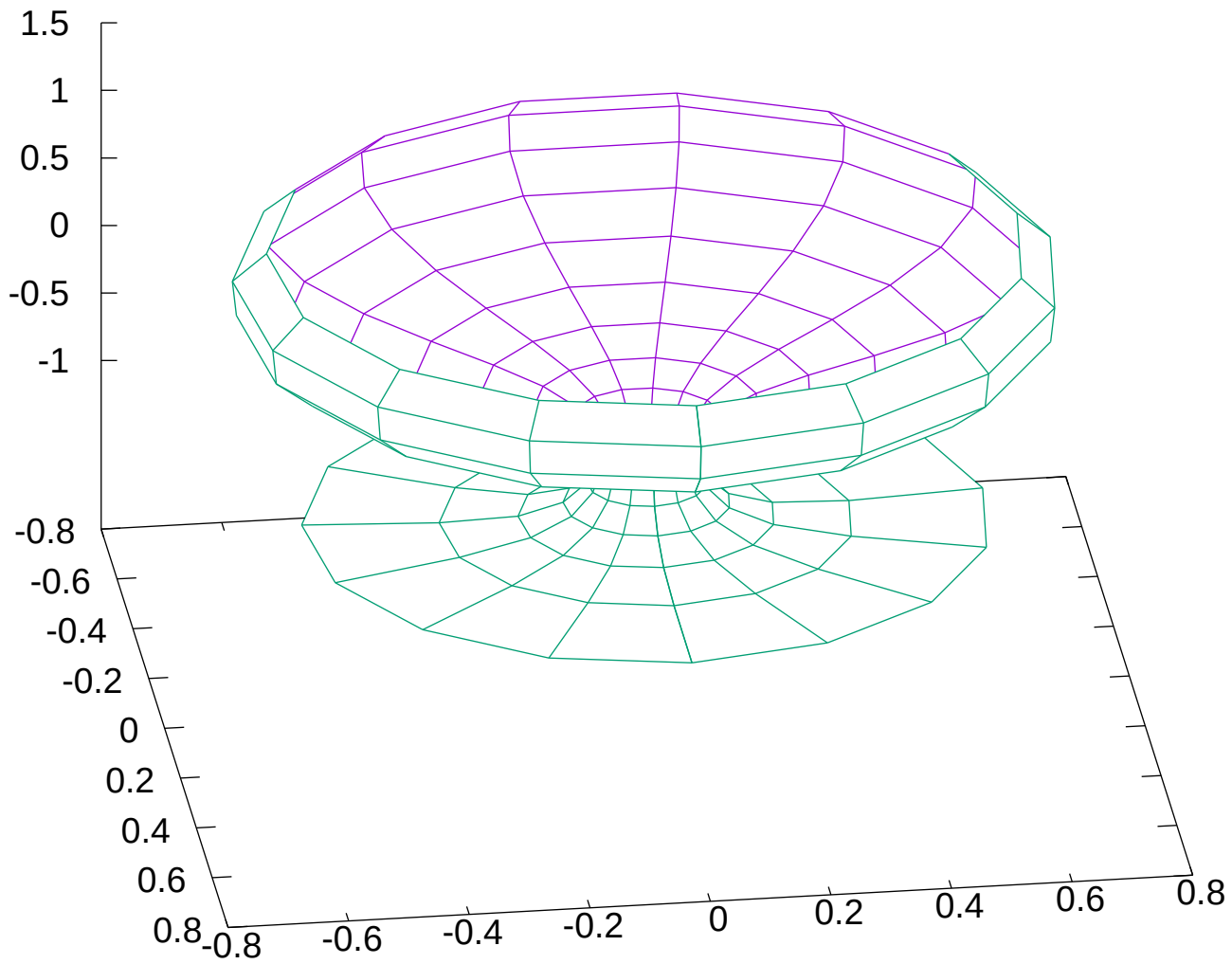

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8


1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

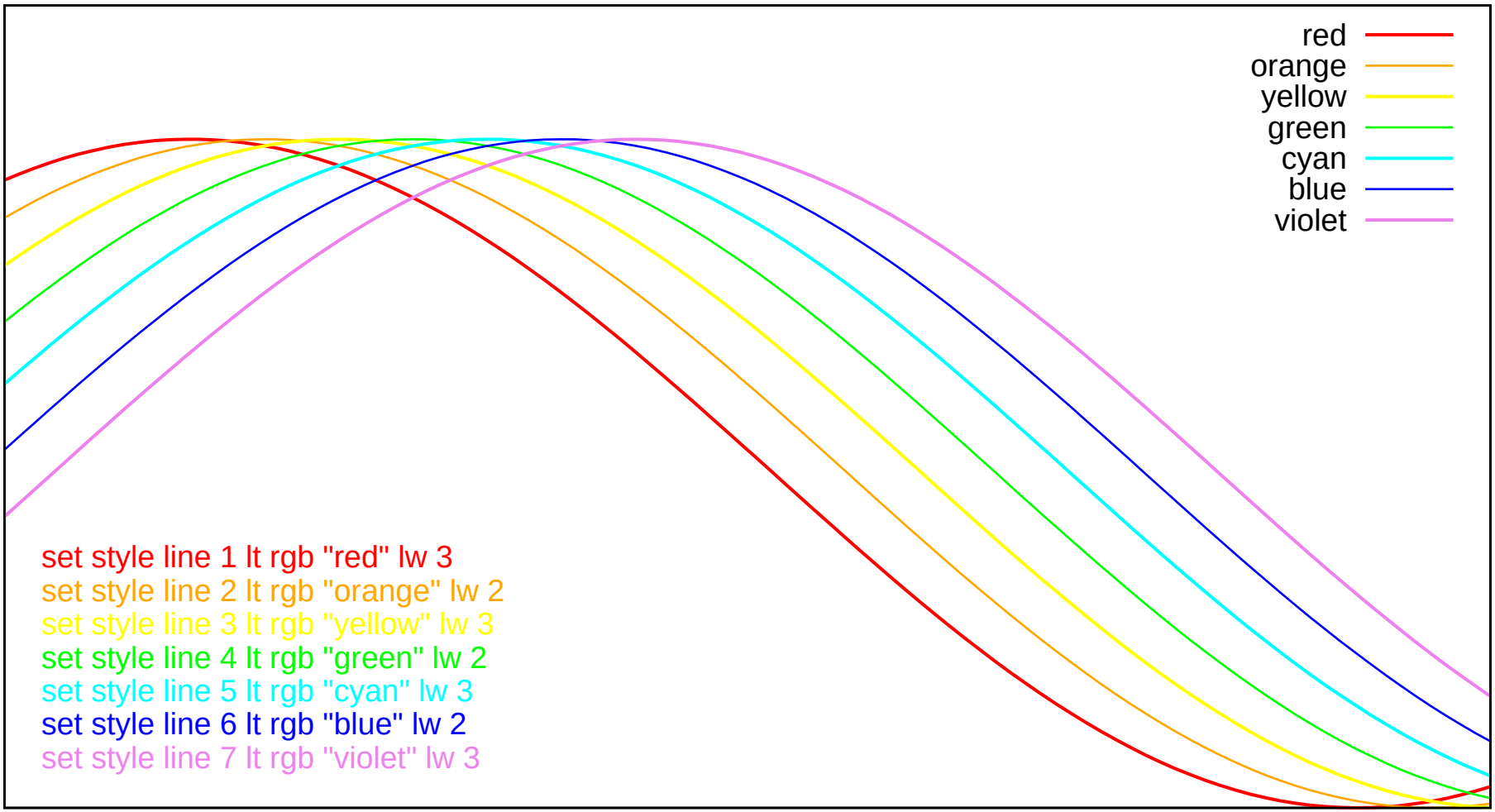

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8


1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8


1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

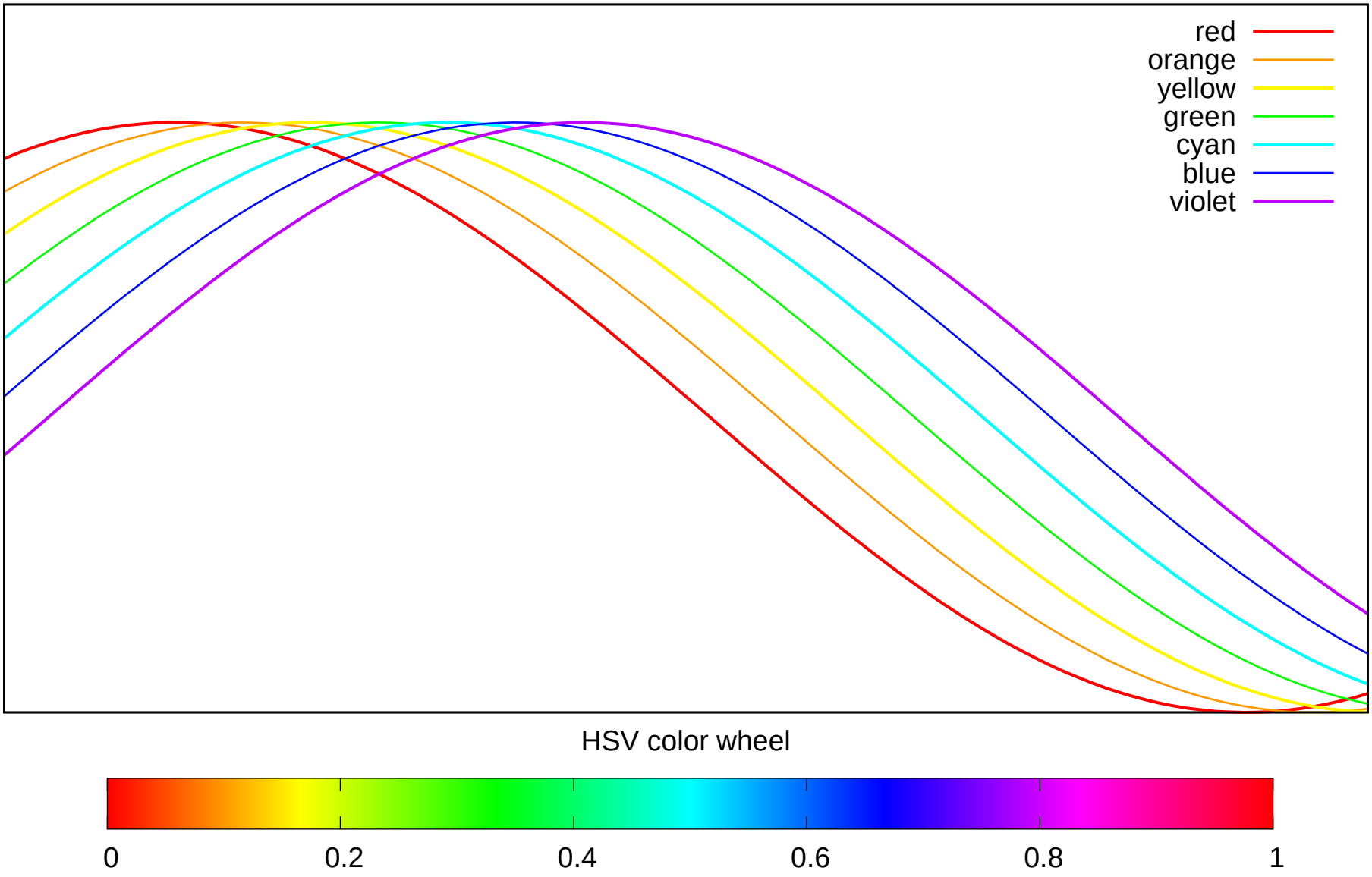

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

Terminal-independent RGB colors in 2D

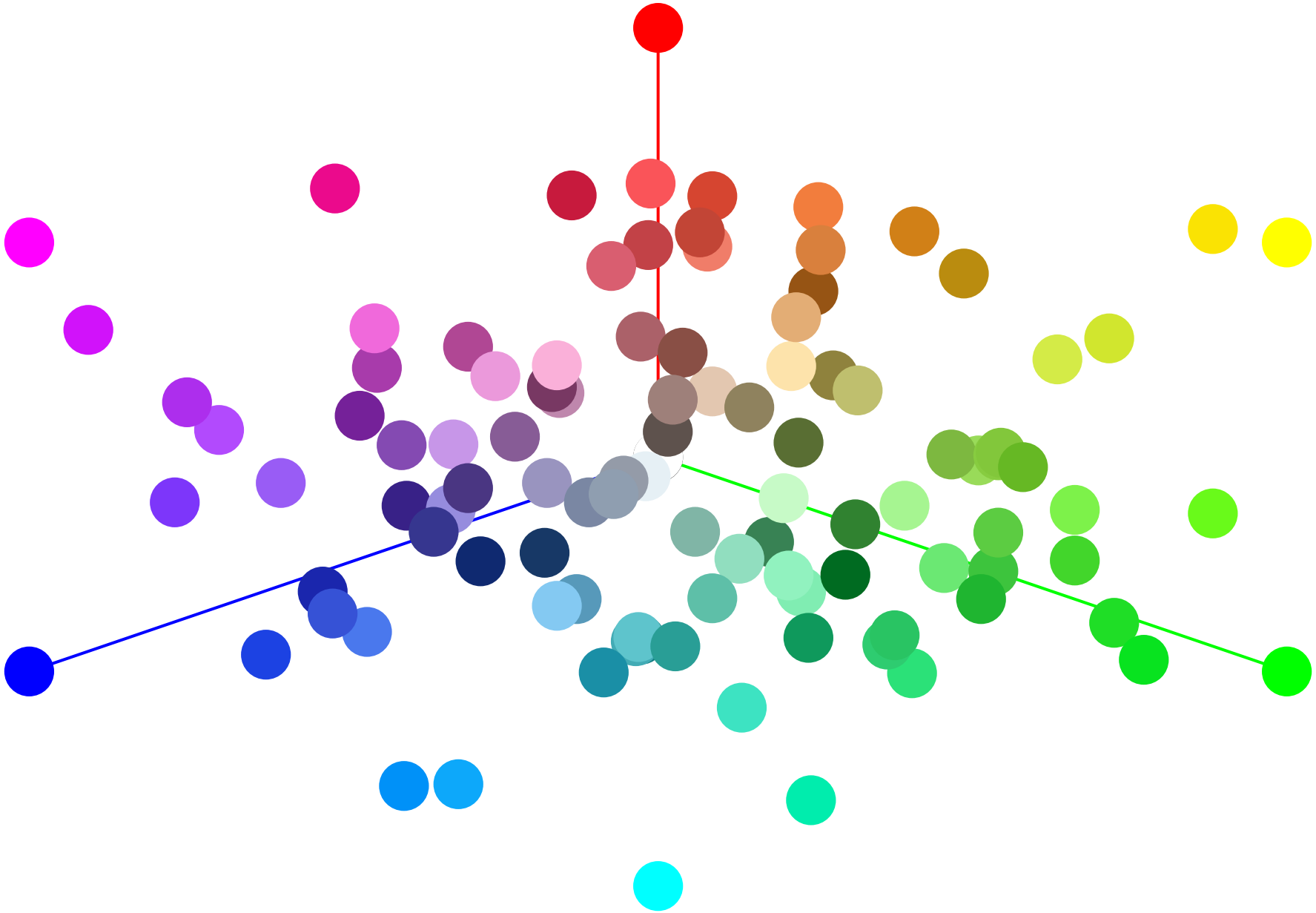


Implemented using built-in rgb color names
(only works for terminals that can do full rgb color)

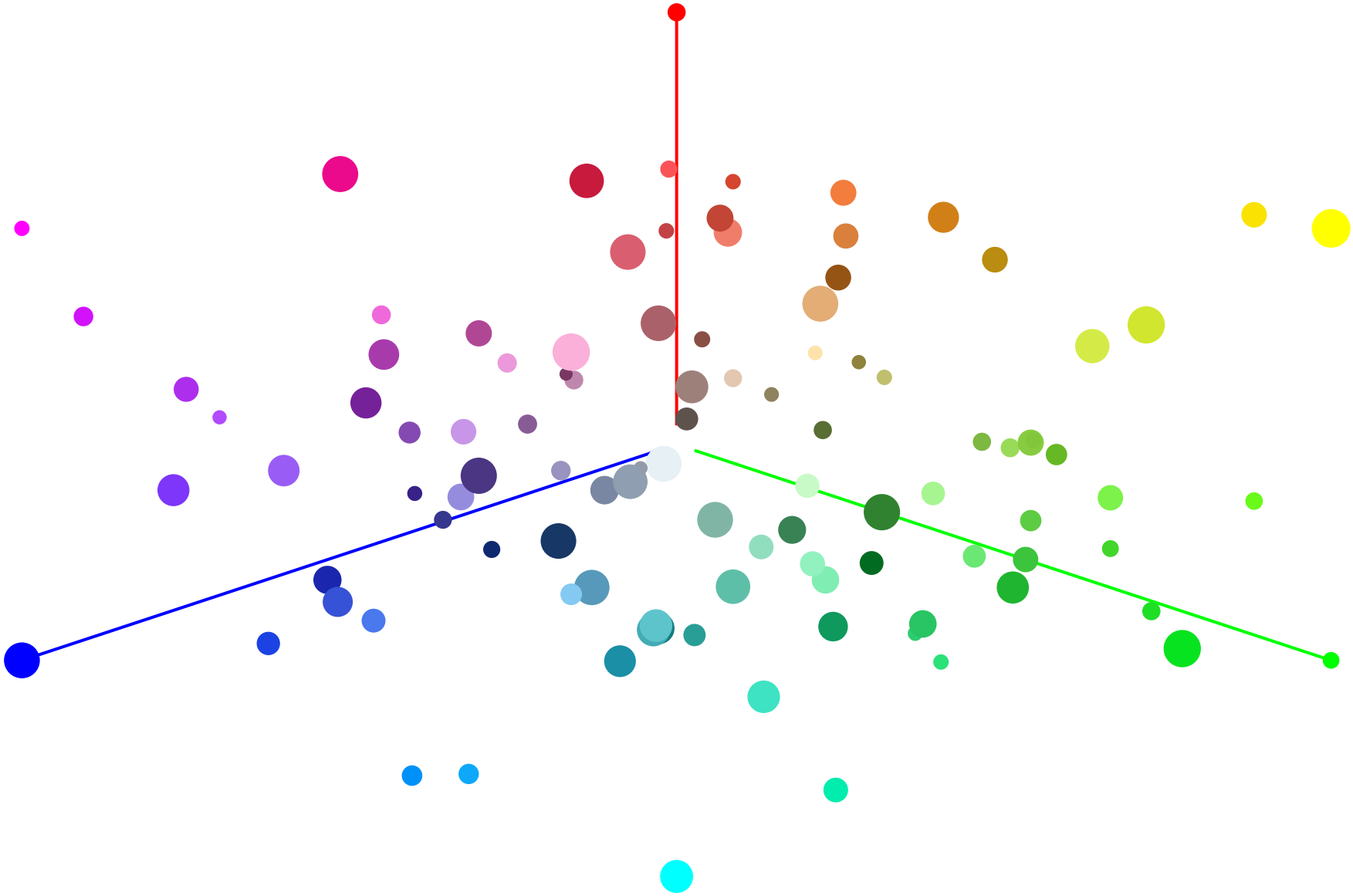
Terminal-independent palette colors in 2D
Implemented using command line macros referring to a fixed HSV palette

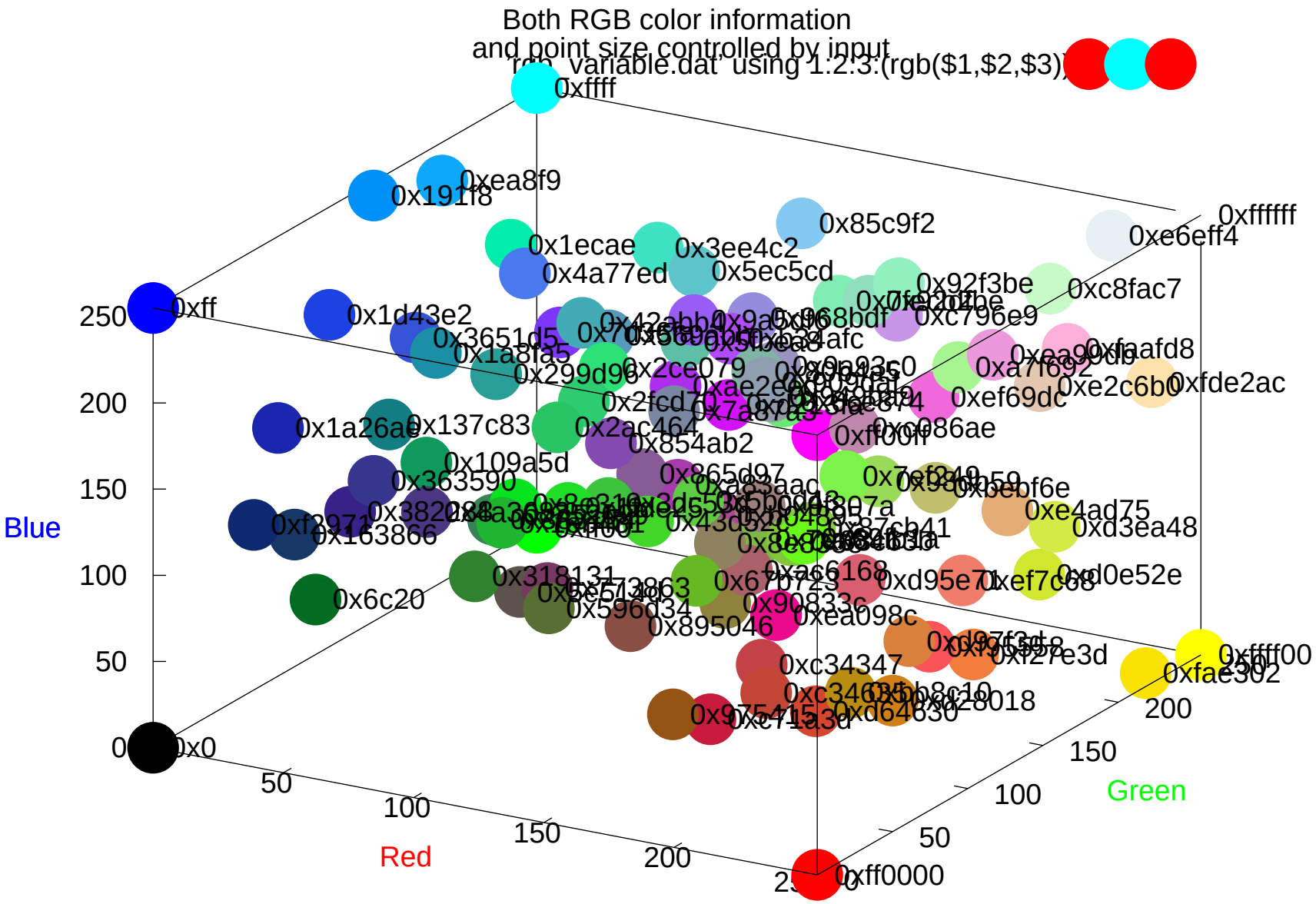


RGB color information read from data file

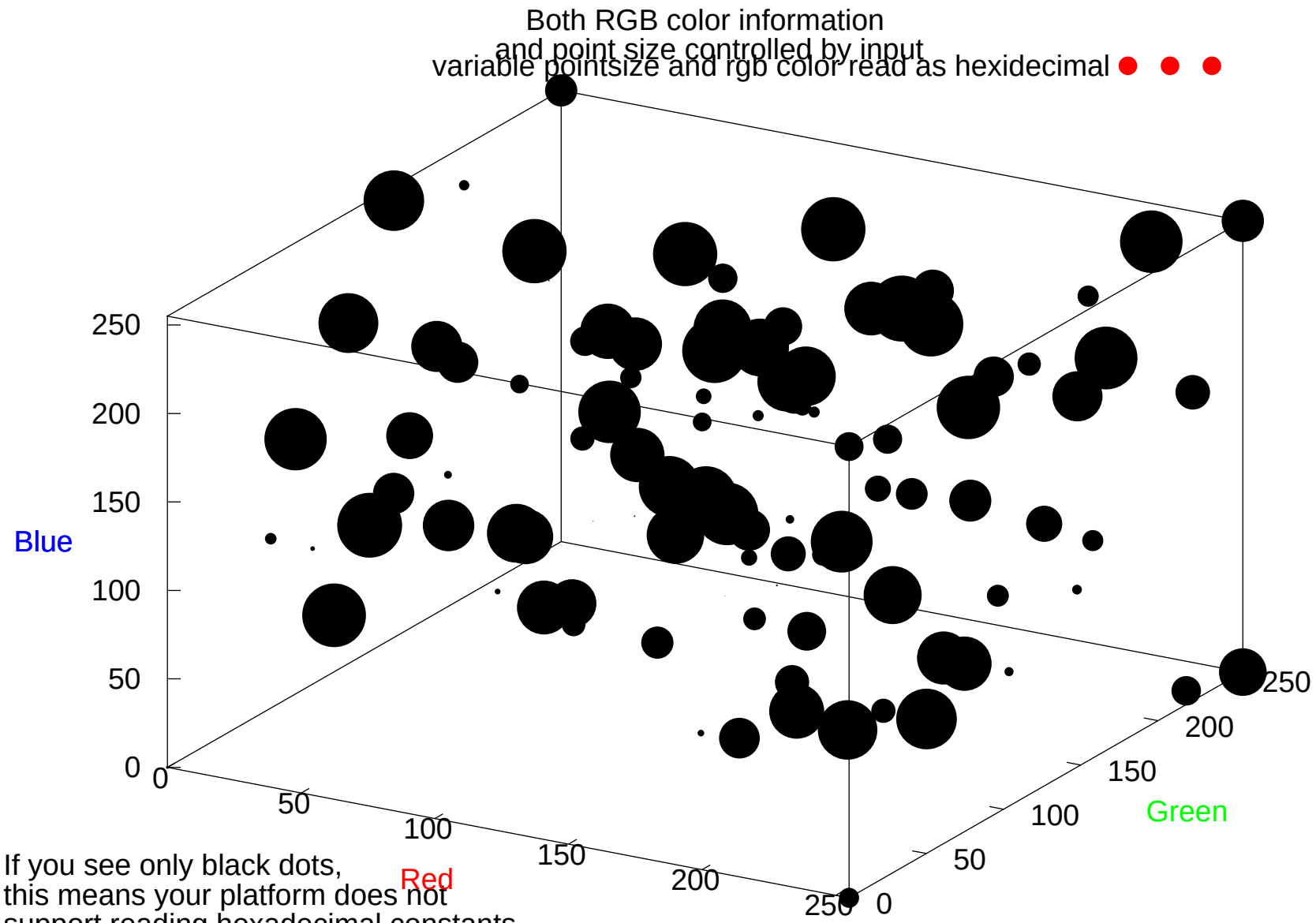


Both RGB color information
and point size controlled by input

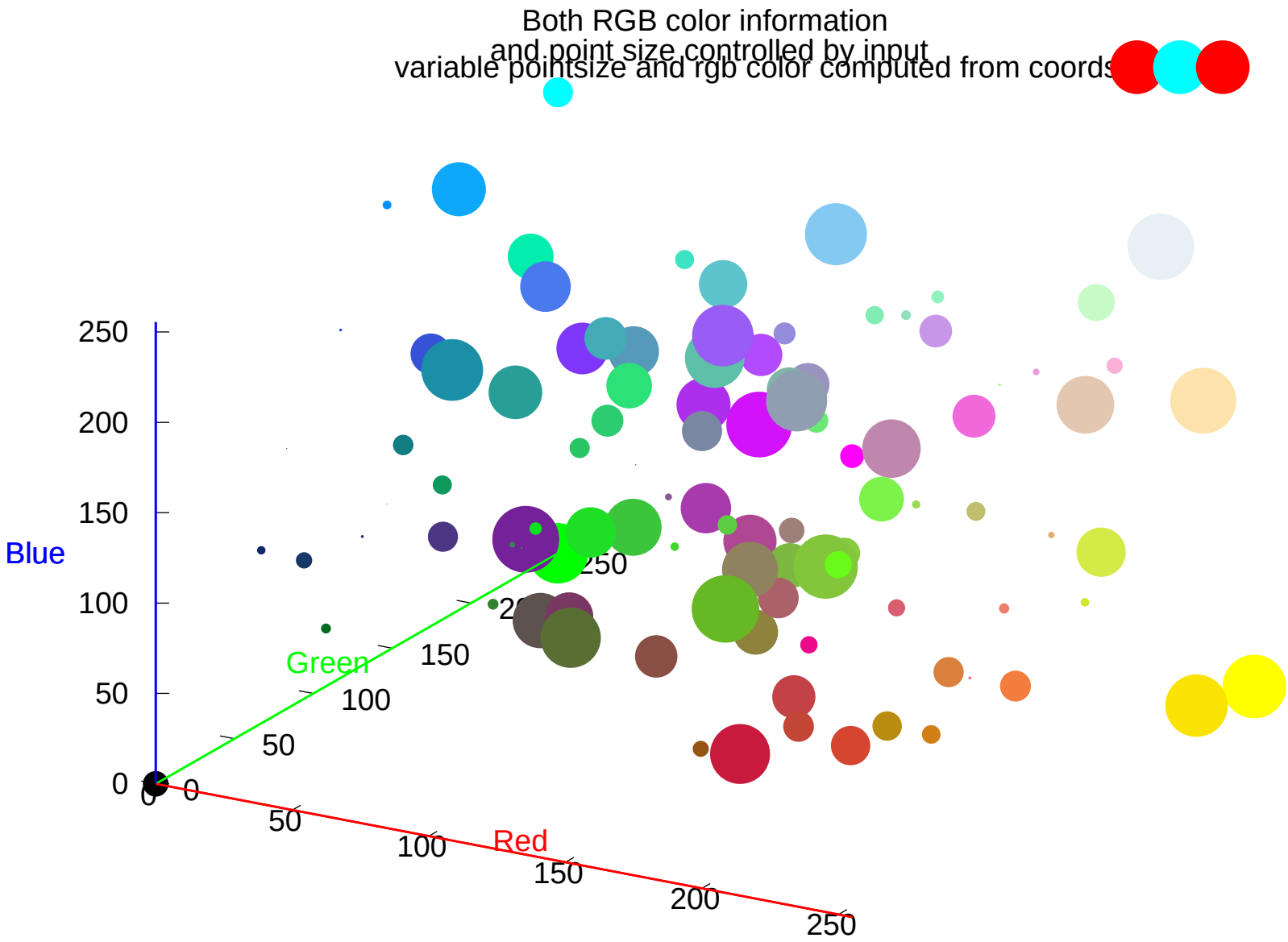


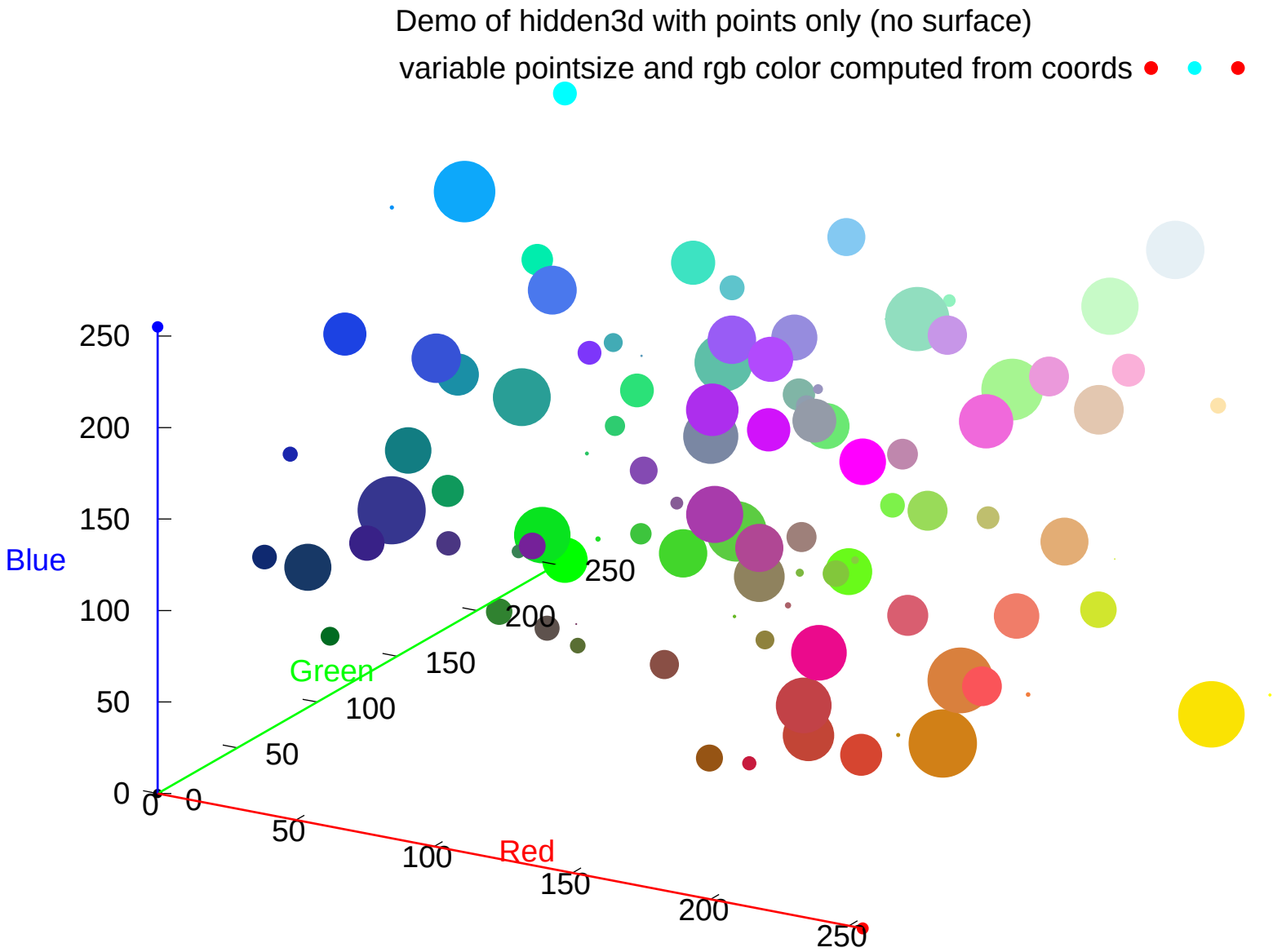


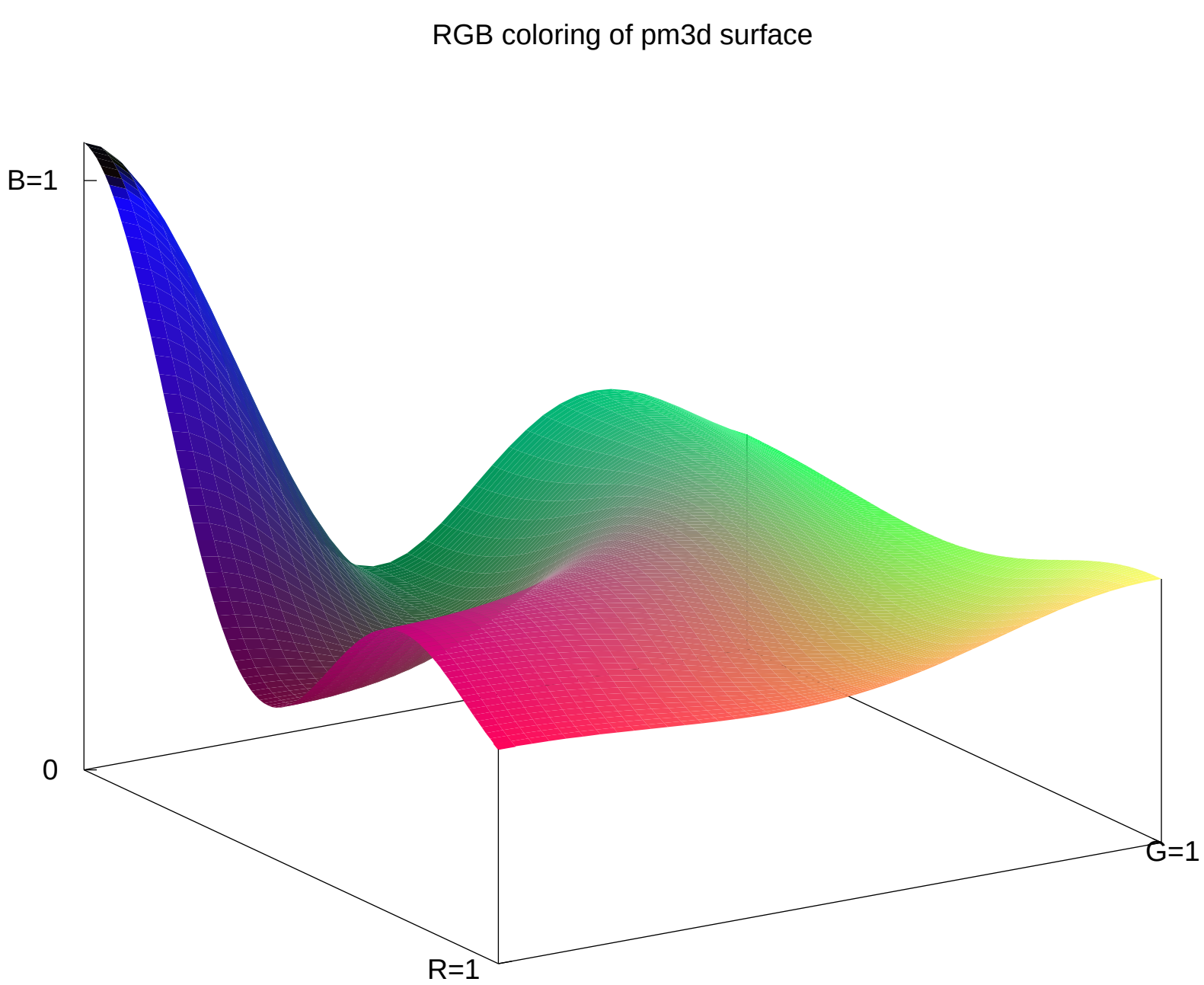
Both RGB color information
and point size controlled by input
rgb_variable.dat' using 1:2:3:(rgb($1,$2,$3))
Blue
Red
Green
0xffff
0xea8f9
0x191f8
0x85c9f2
0xffffff
0xe6eff4
0x1ecae
0x3ee4c2
0x4a77ed
0x5ec5cd
0x92f3be
0xc8fac7
0xff
0x1d43e2
0xc796e9
0x3651d5
0x1a8fa5
0x299d96
0x2ce079
0xa7f692
0xfaafd8
0xe2c6b0
0xfde2ac
0xef69dc
0x1a26ae
0x137c83
0x2ac464
0x854ab2
0xff00ff
0x109a5d
0x363590
0xe4ad75
0xd3ea48
0x163866
0xd0e52e
0x6c20
0x318131
0x596d34
0x895046
0xea098c
0xc34347
0xf27e3d
0xfae302
0xd64630
0x0
0xff0000
0
50
100
150
200
250

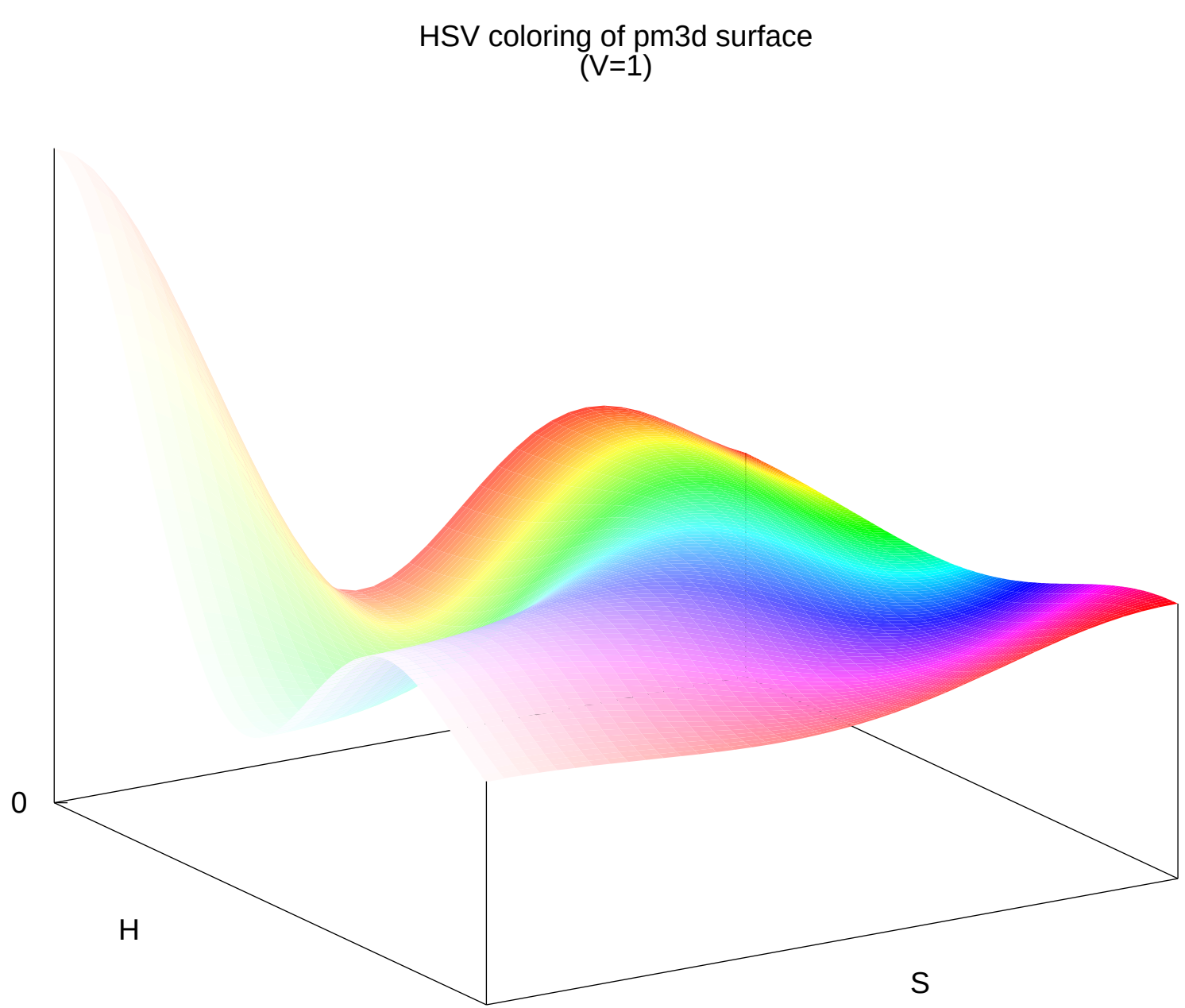


If you see only black dots,
this means your platform does not
support reading hexadecimal constants
from a data file. Get a newer libc.

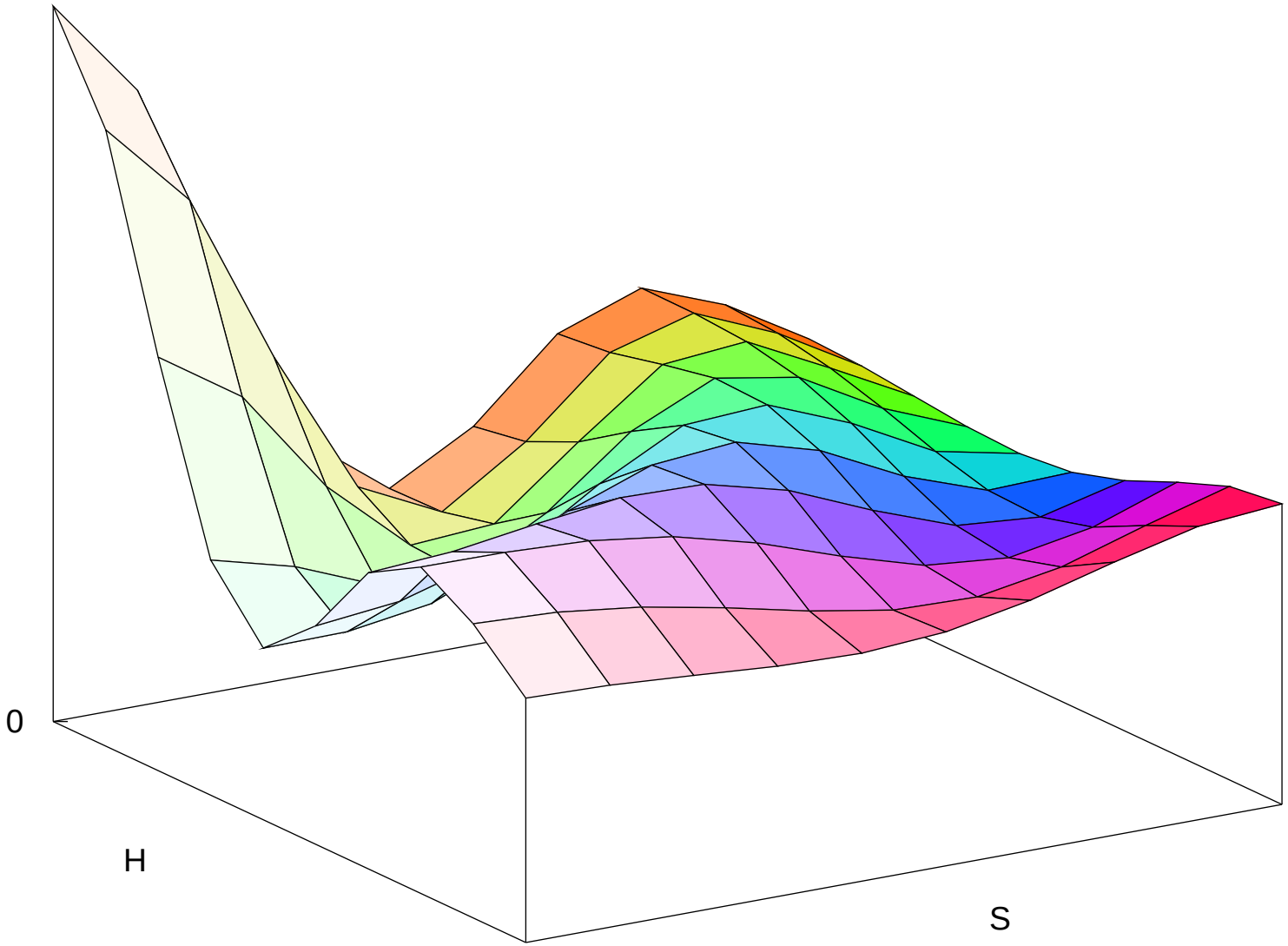

Both RGB color information
and point size controlled by input
variable pointsize and rgb color computed from coords
Blue
Green
Red
0
50
100
150
200
250

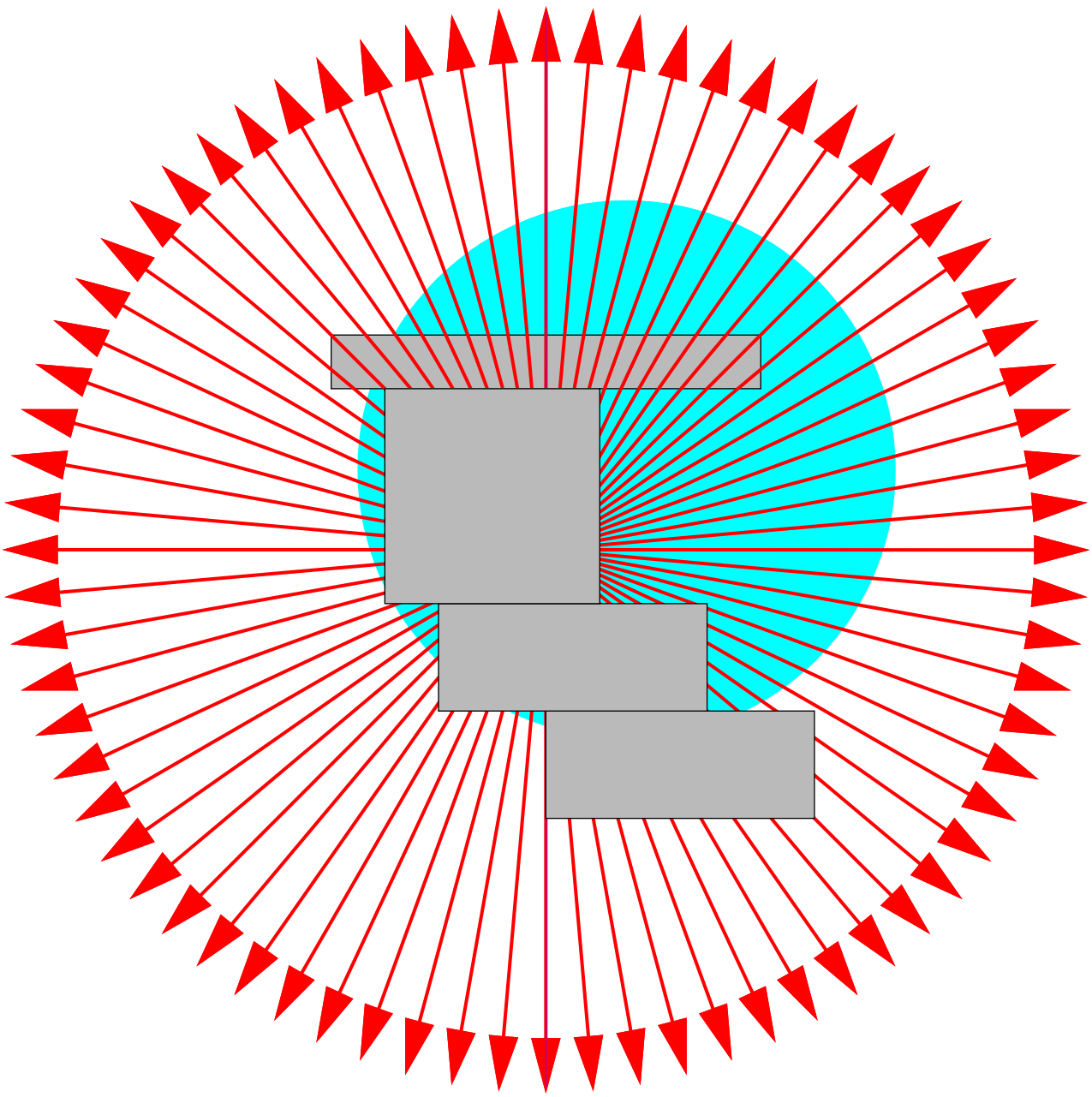

Demo of hidden3d with points only (no surface)
variable pointsize and rgb color computed from coords
Blue
Green
Red
0
50
100
150
200
250

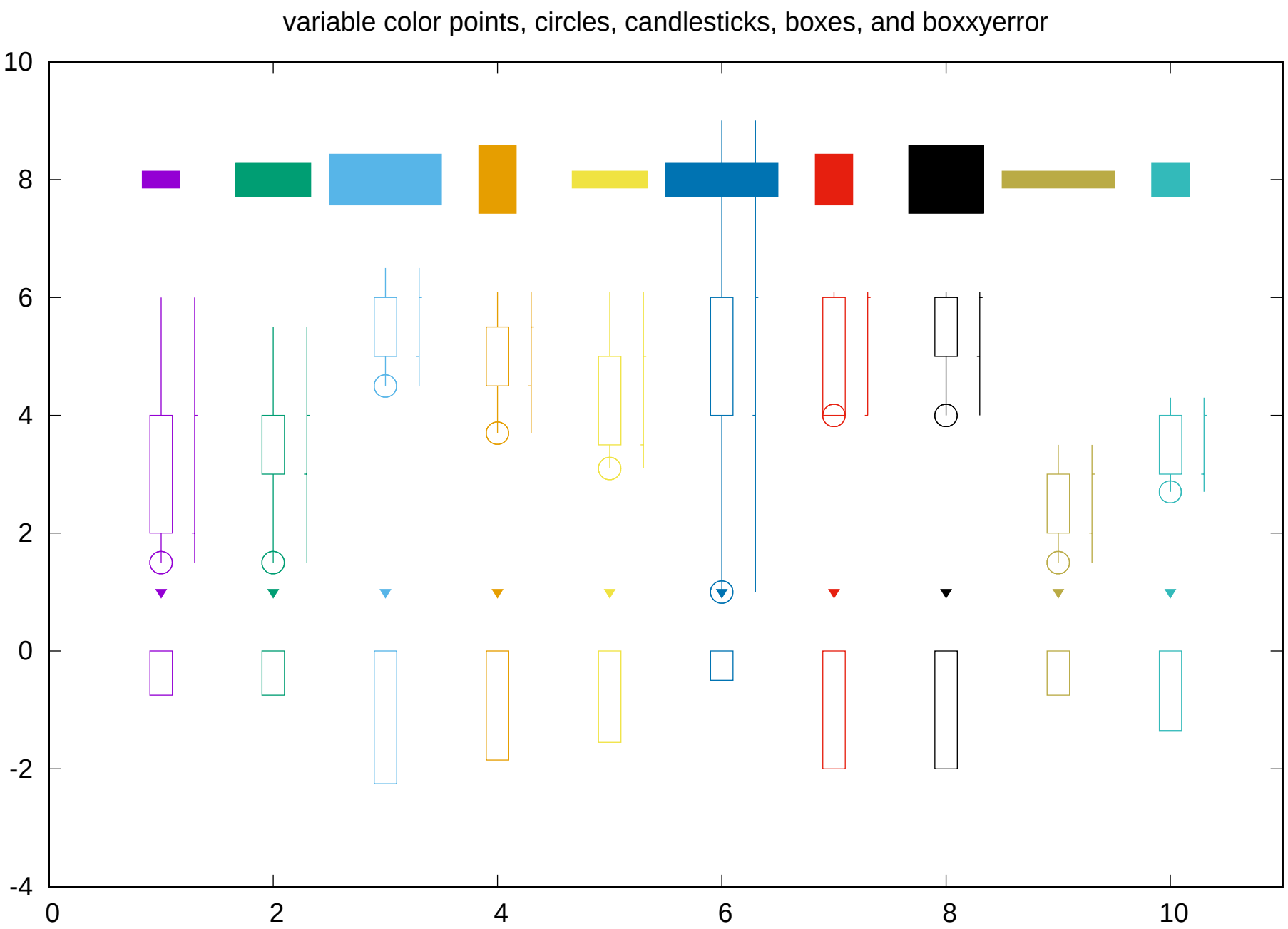

RGB coloring of pm3d surface
B=1
0
R=1
G=1

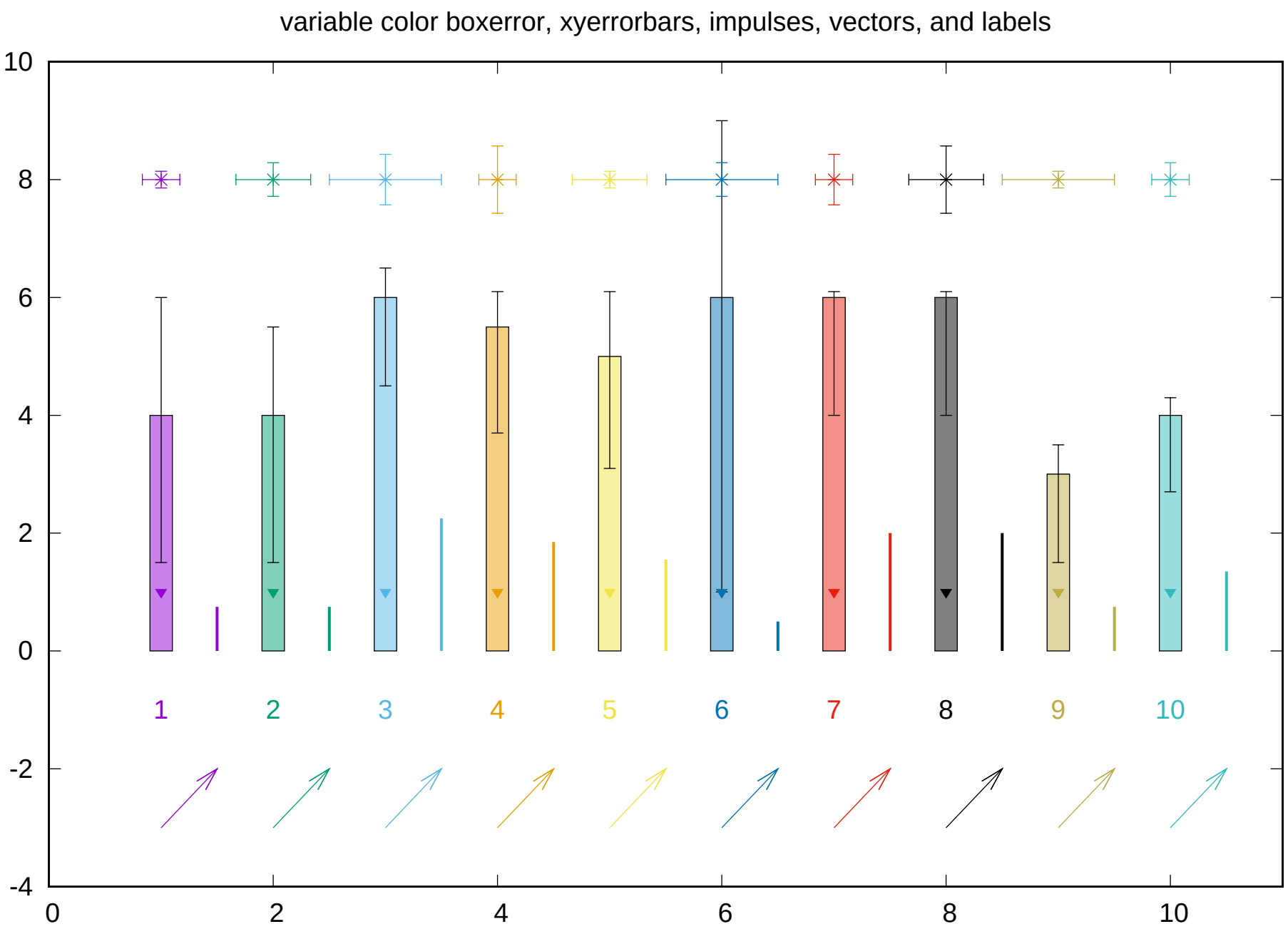

HSV coloring of pm3d surface
(V=1)
0
H
S

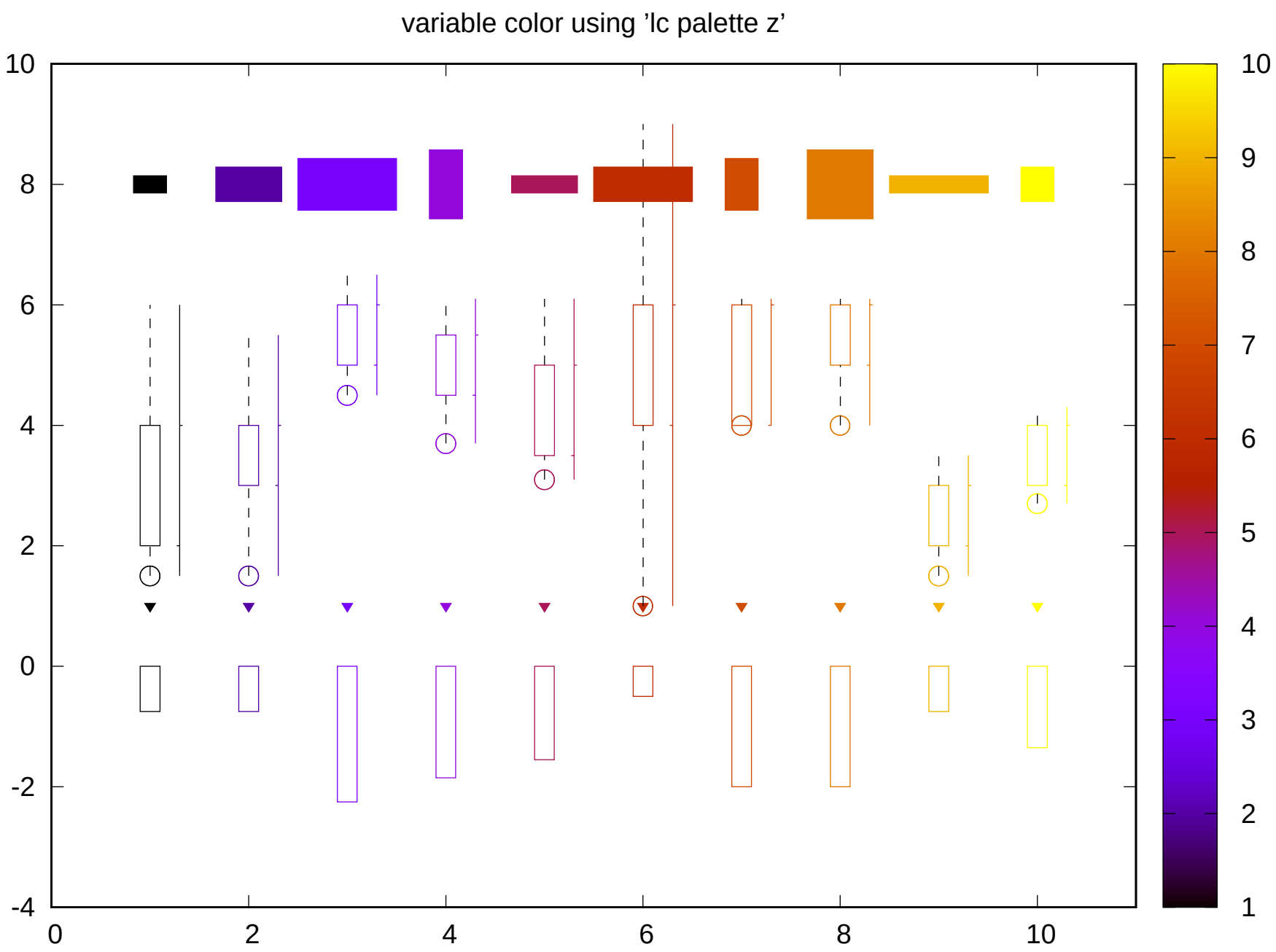
Explicit borders for pm3d tiling

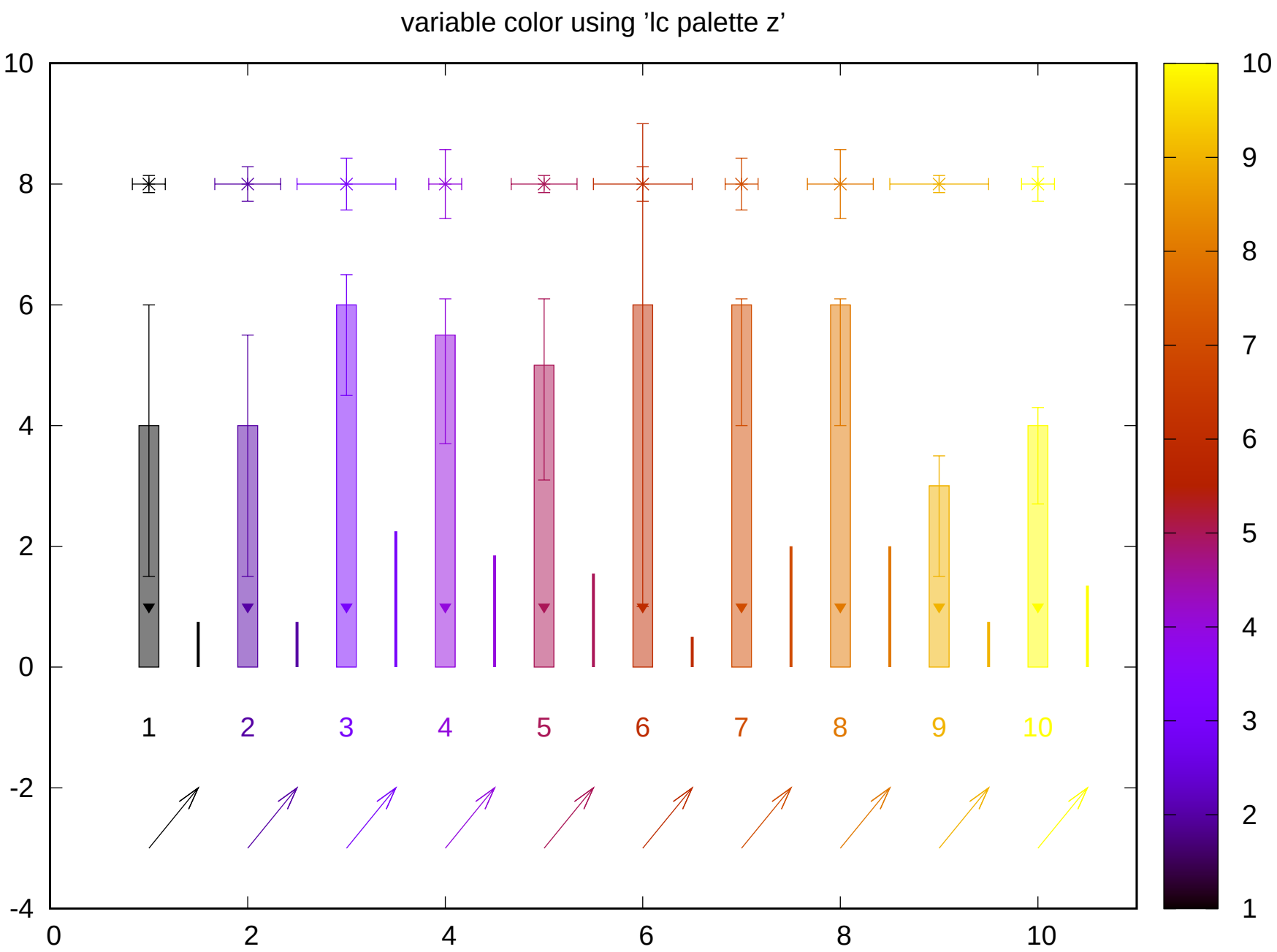


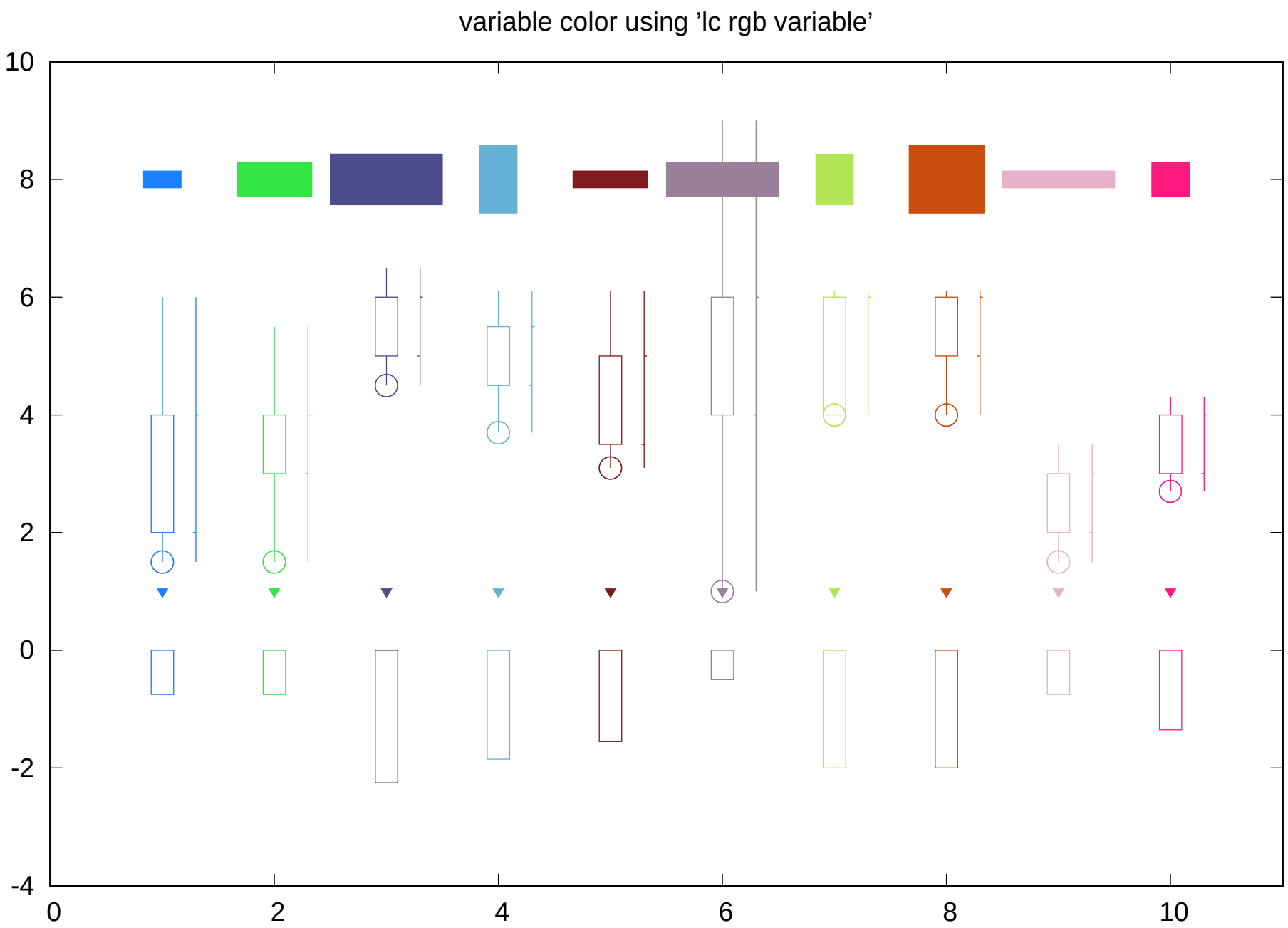


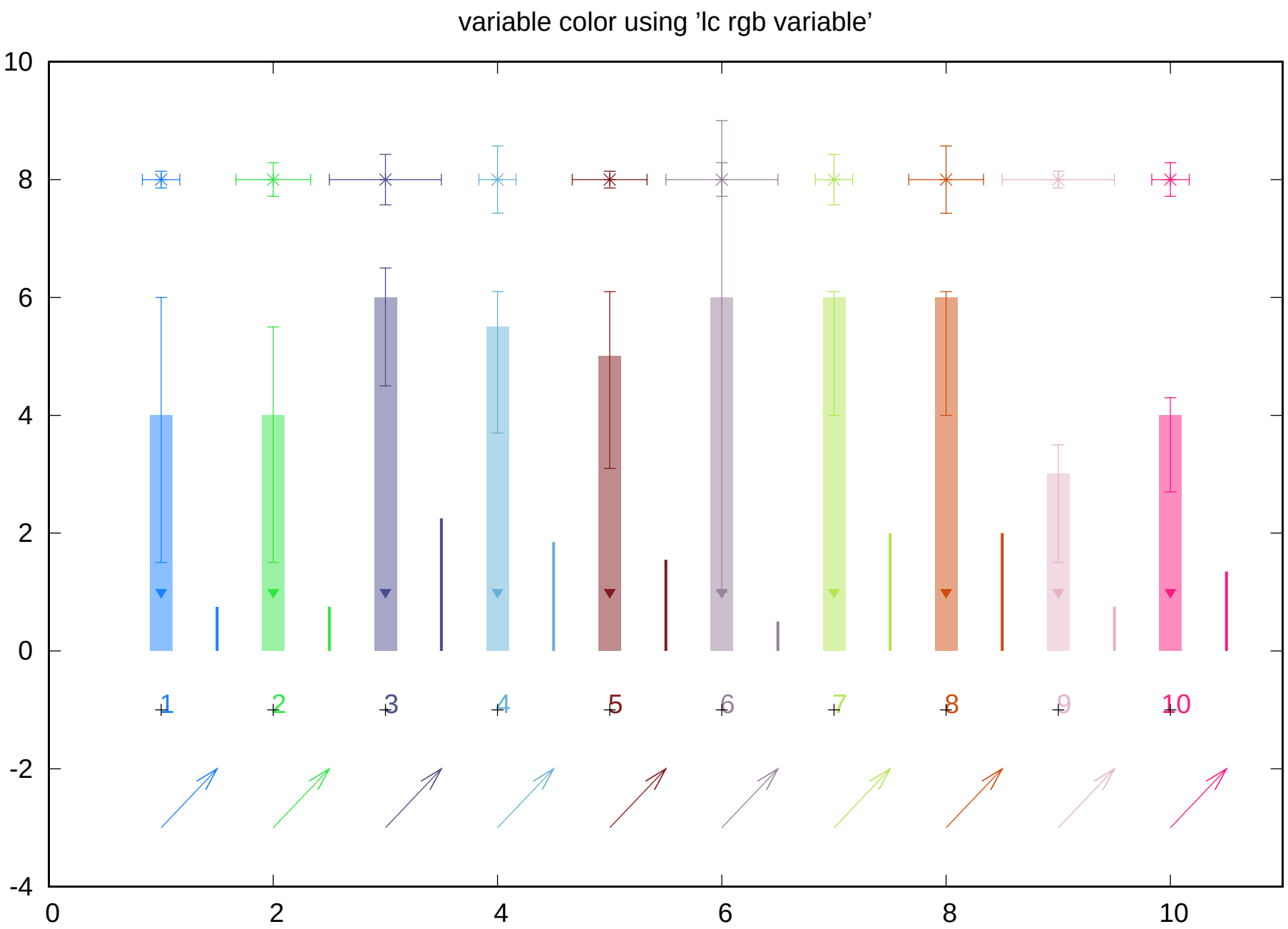

variable color points, circles, candlesticks, boxes, and boxxyerror
10
8
6
4
2
0
-2
-4
0
2
4
6
8
10

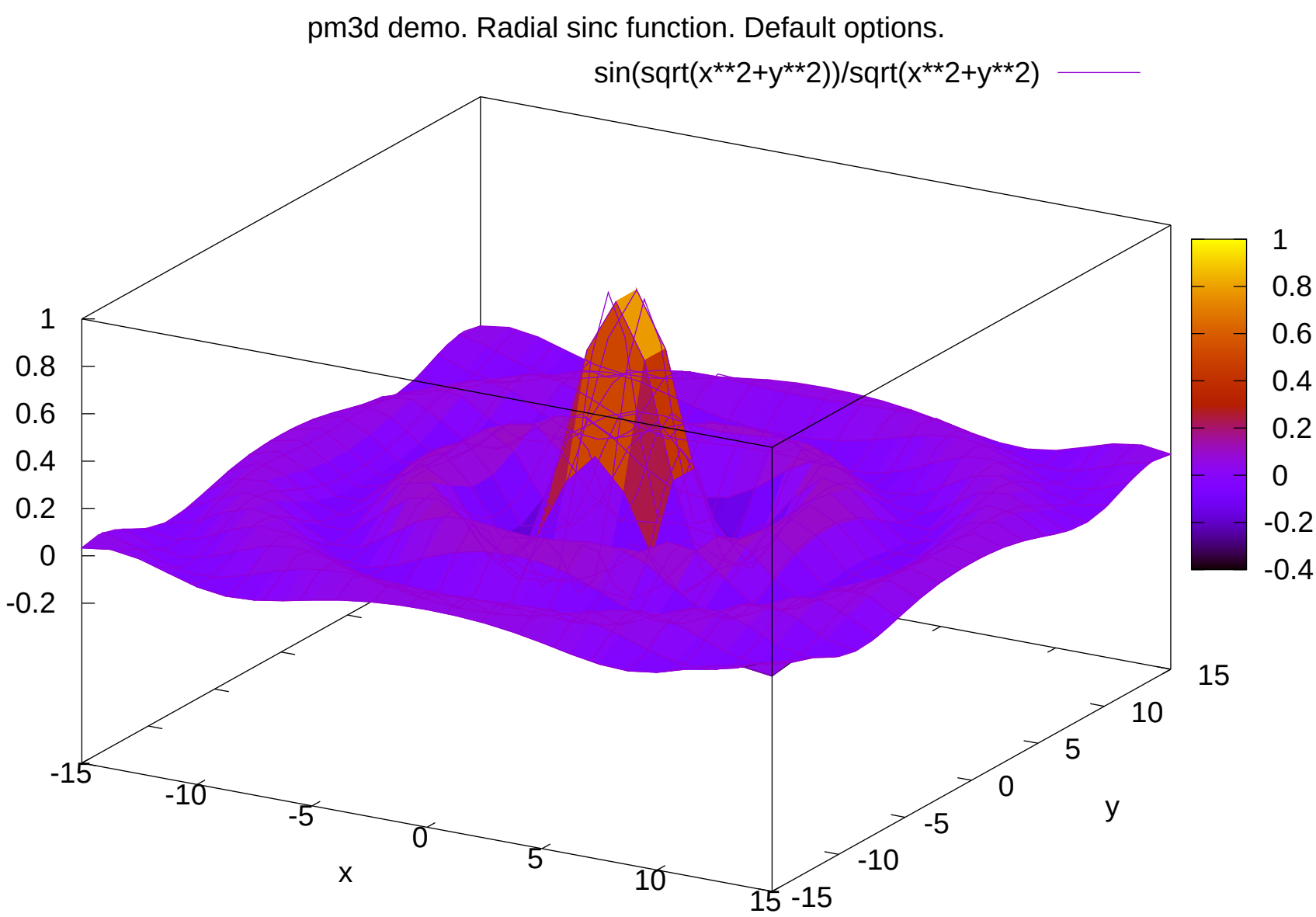

variable color boxerror, xyerrorbars, impulses, vectors, and labels
10
8
6
4
2
0
-2
-4
0
2
4
6
8
10
1
2
3
4
5
6
7
8
9
10

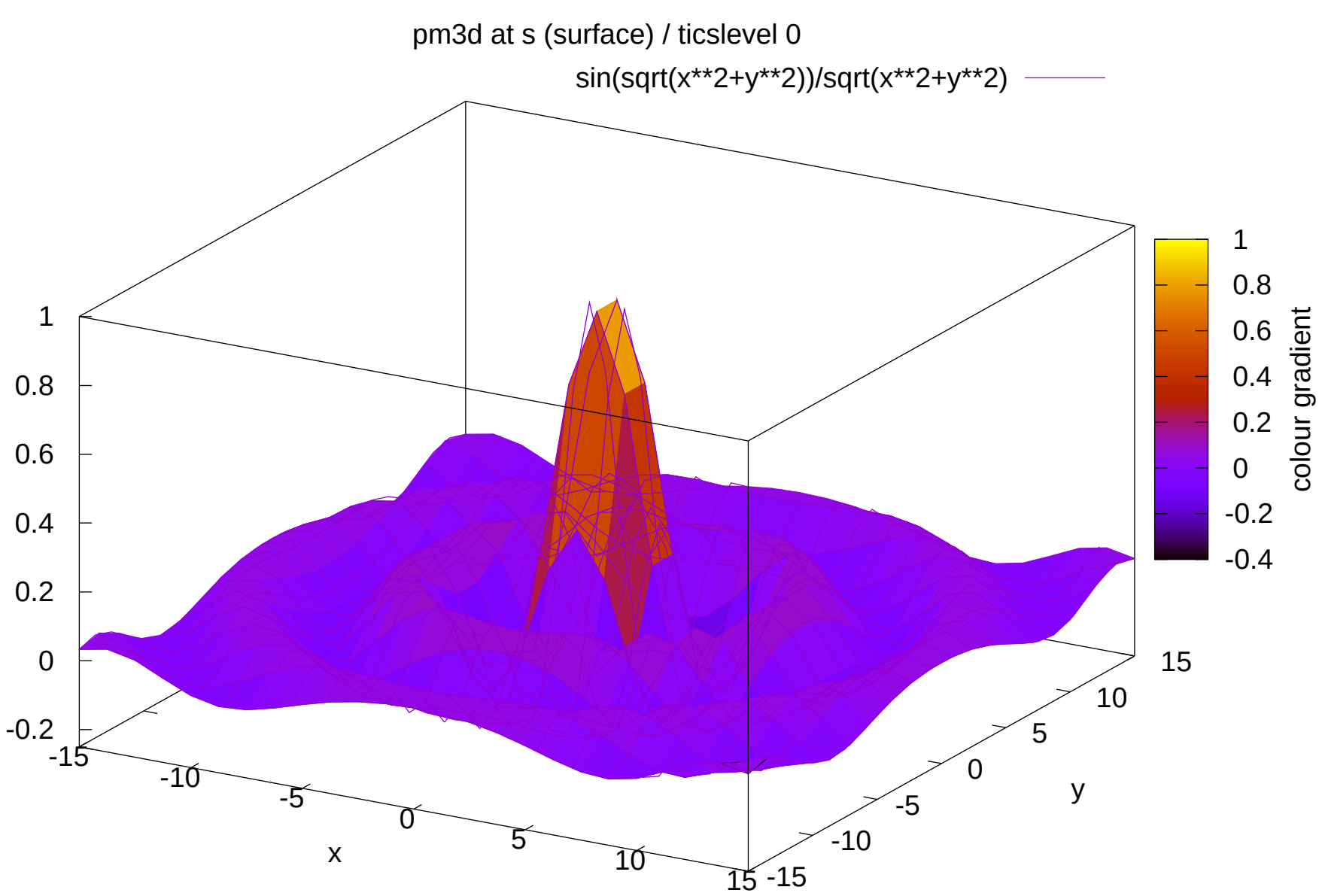

variable color using ’lc palette z’
10
8
6
4
2
0
-2
-4
0
2
4
6
8
10
10
9
8
7
6
5
4
3
2
1

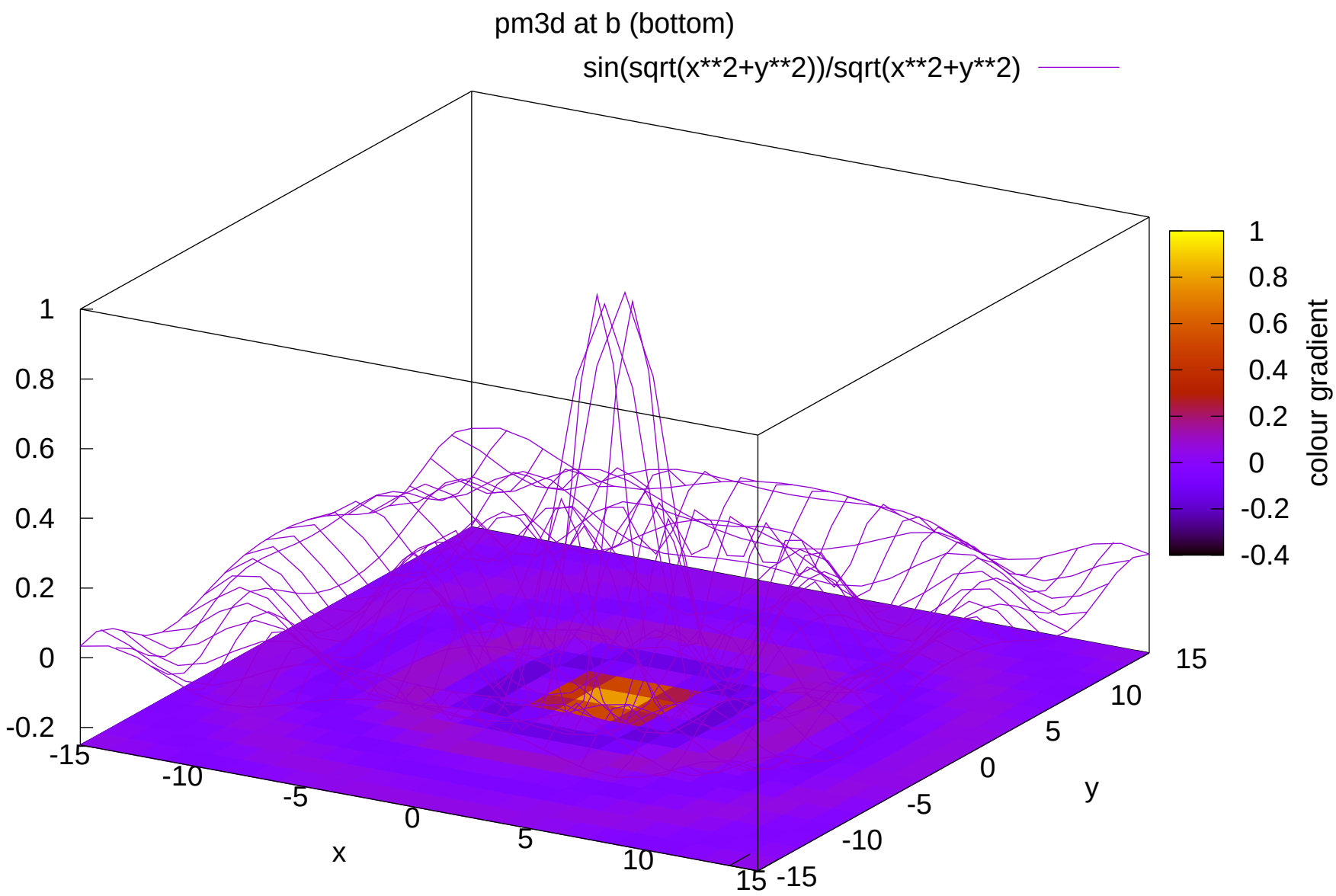

variable color using ’lc palette z’
10
8
6
4
2
0
-2
-4
0
2
4
6
8
10
1
2
3
4
5
6
7
8
9
10
10
9
8
7
6
5
4
3
2
1

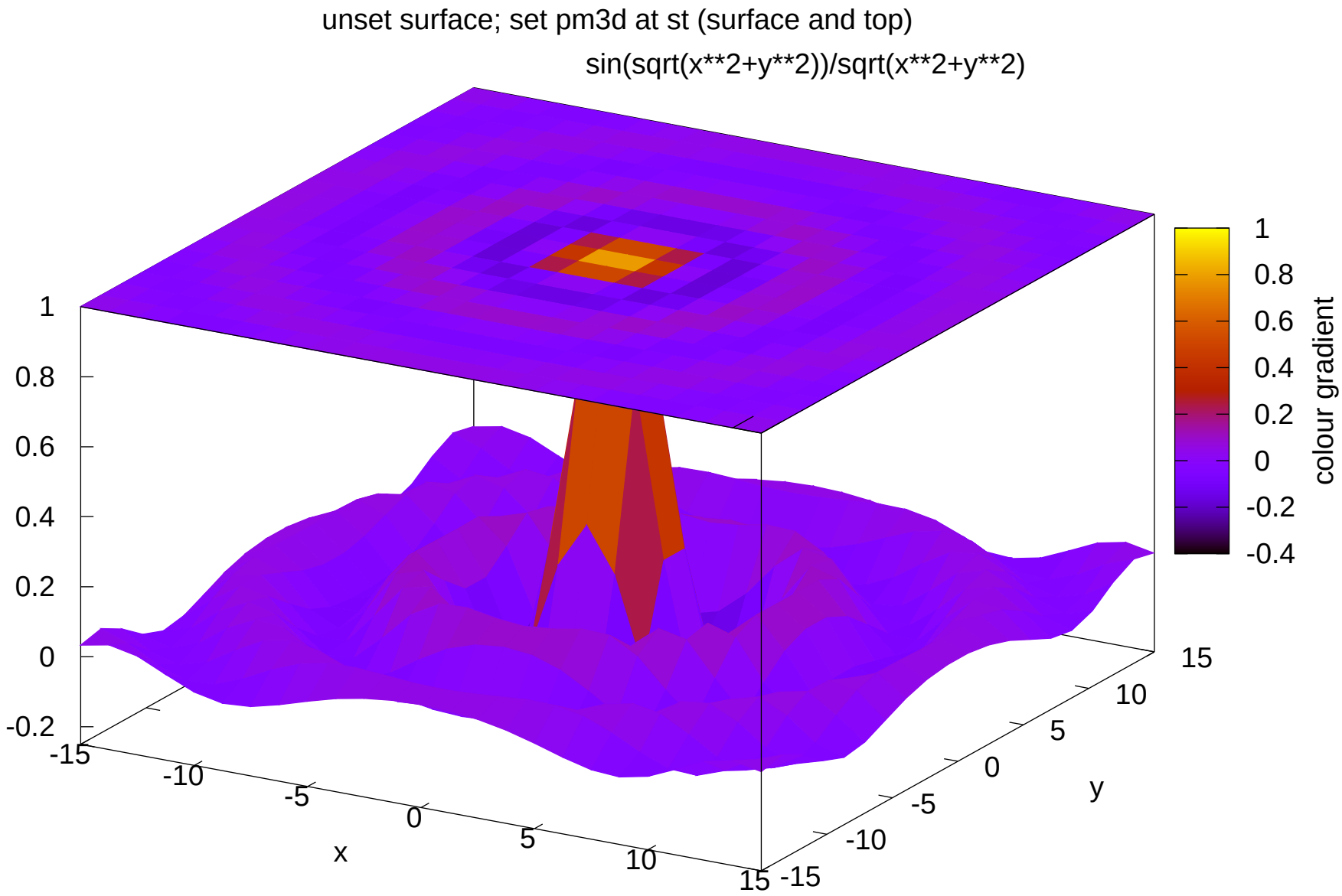

variable color using ’lc rgb variable’
10
8
6
4
2
0
-2
-4
0
2
4
6
8
10

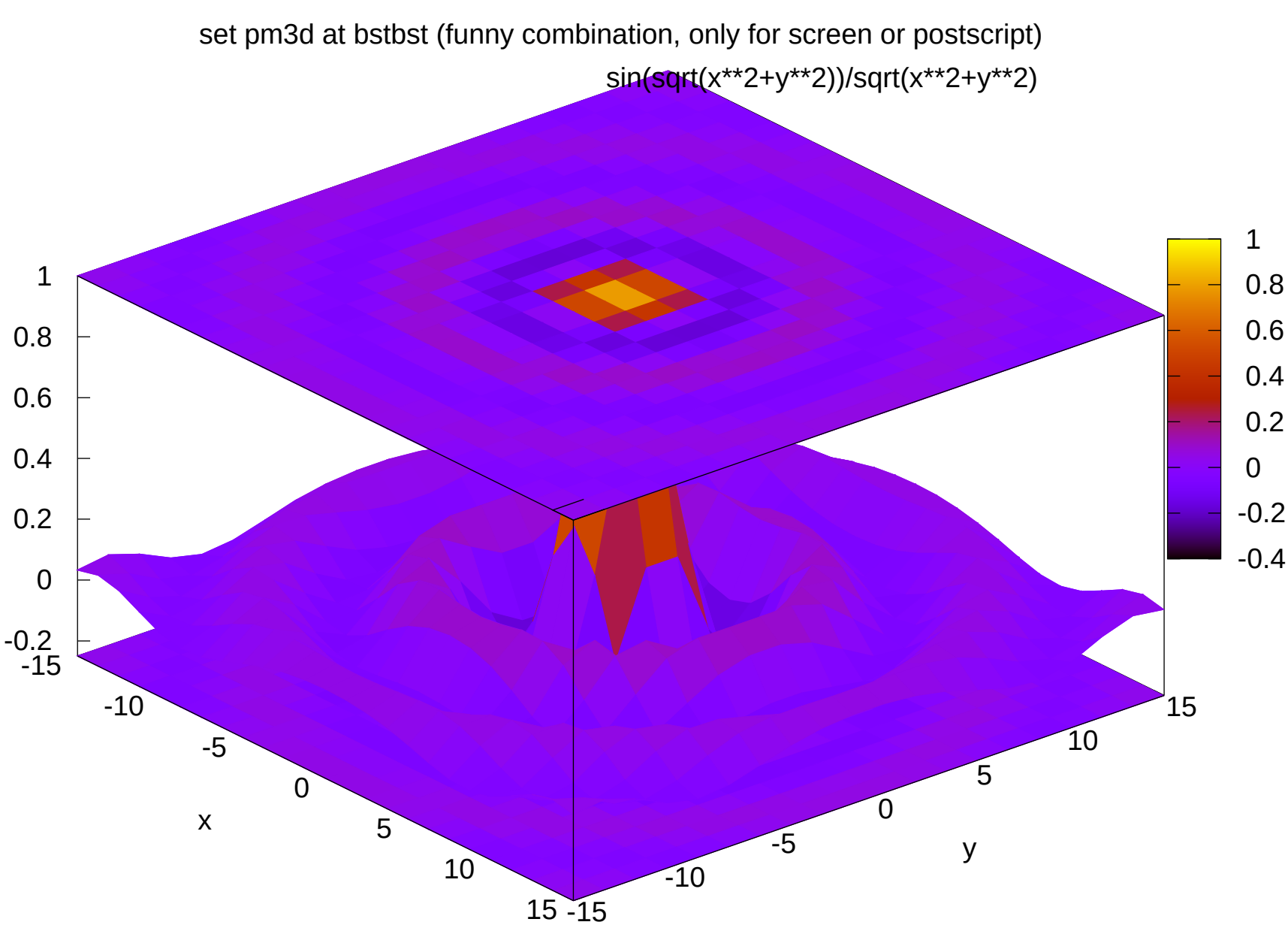

variable color using ’lc rgb variable’
10
8
6
4
2
0
-2
-4
0
2
4
6
8
10
1
2
3
4
5
6
7
8
9
10

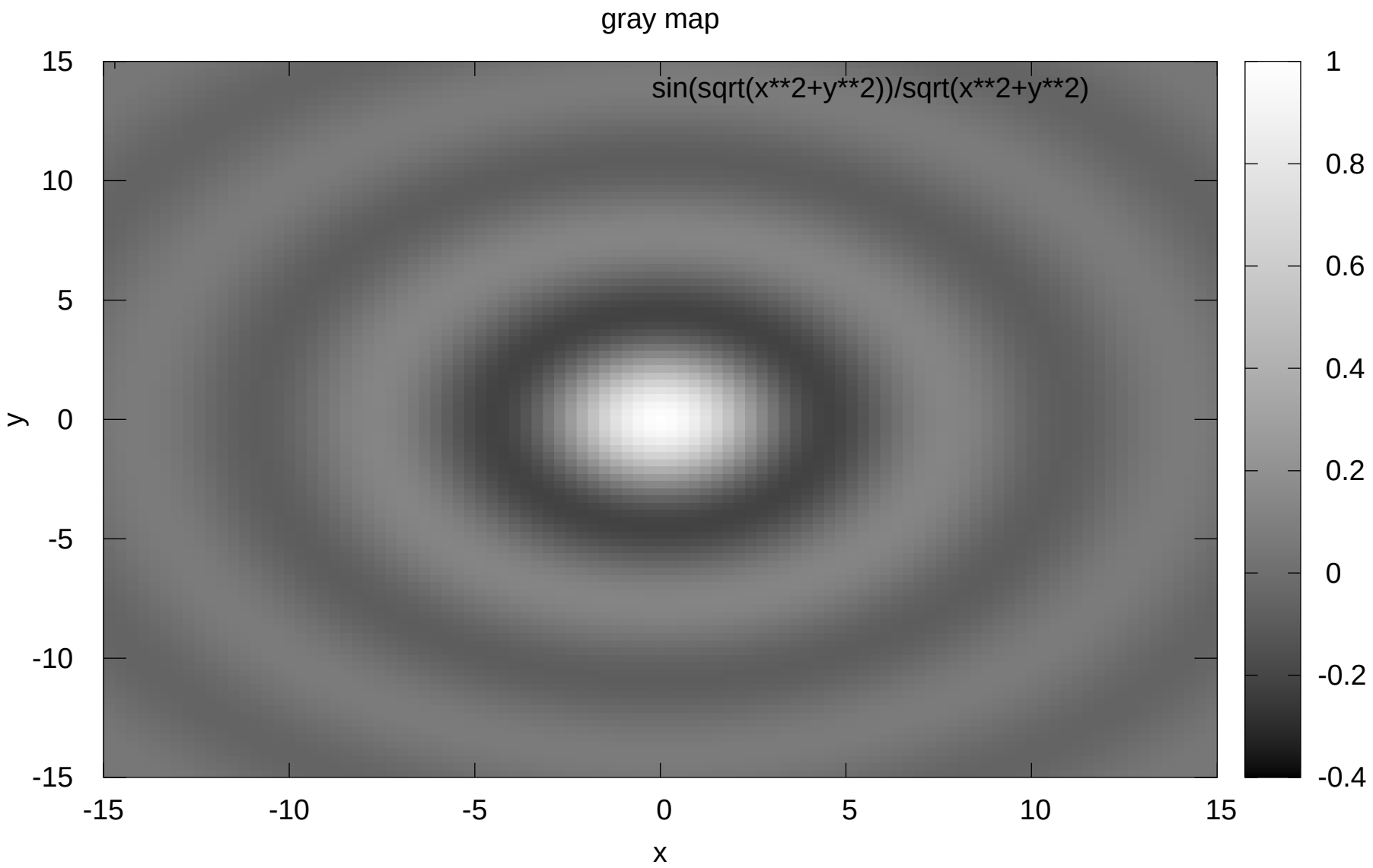

pm3d demo. Radial sinc function. Default options.
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x
y
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

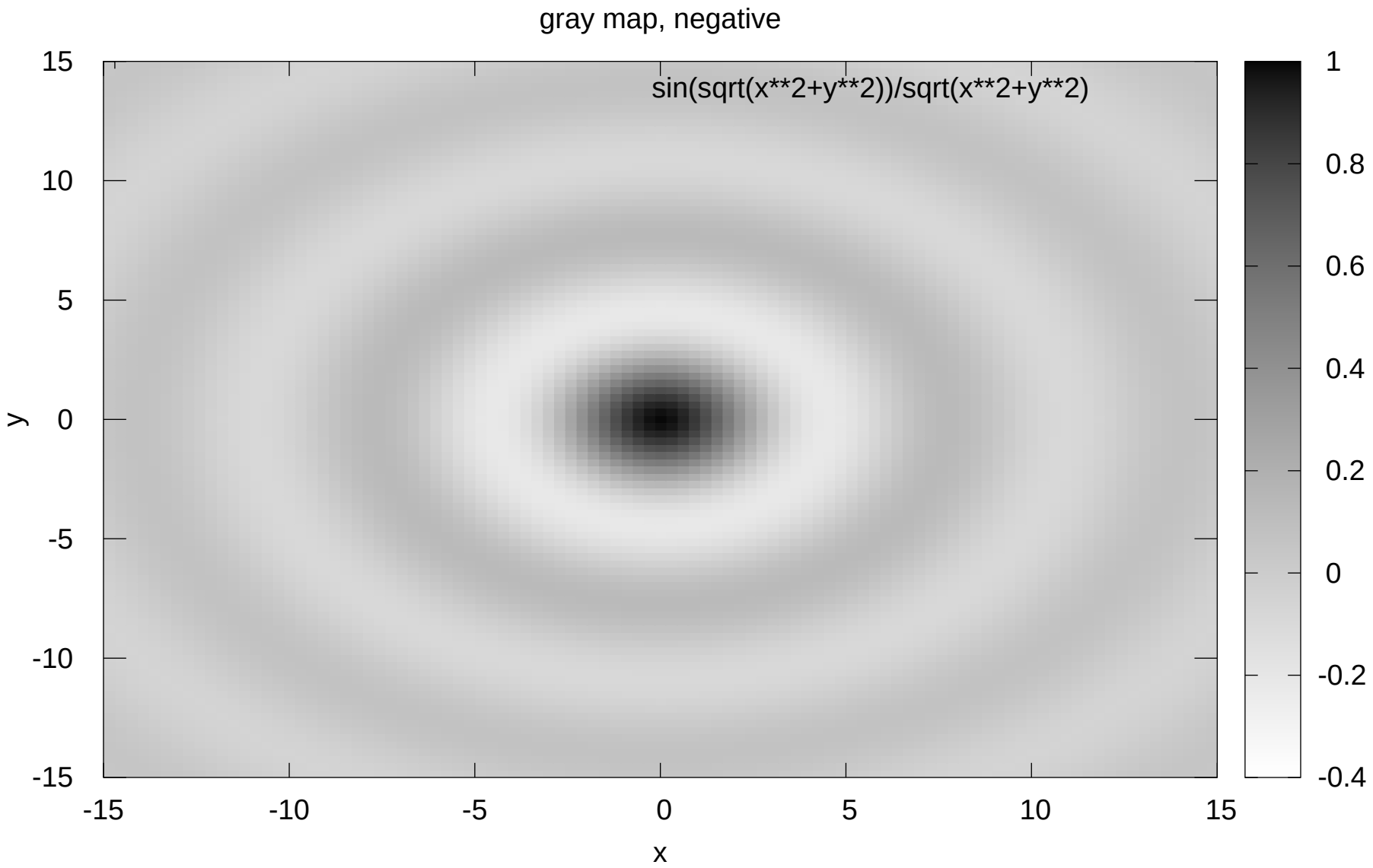

pm3d at s (surface) / ticslevel 0
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x
-15
-10
-5
0
5
10
15
y
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
colour gradient

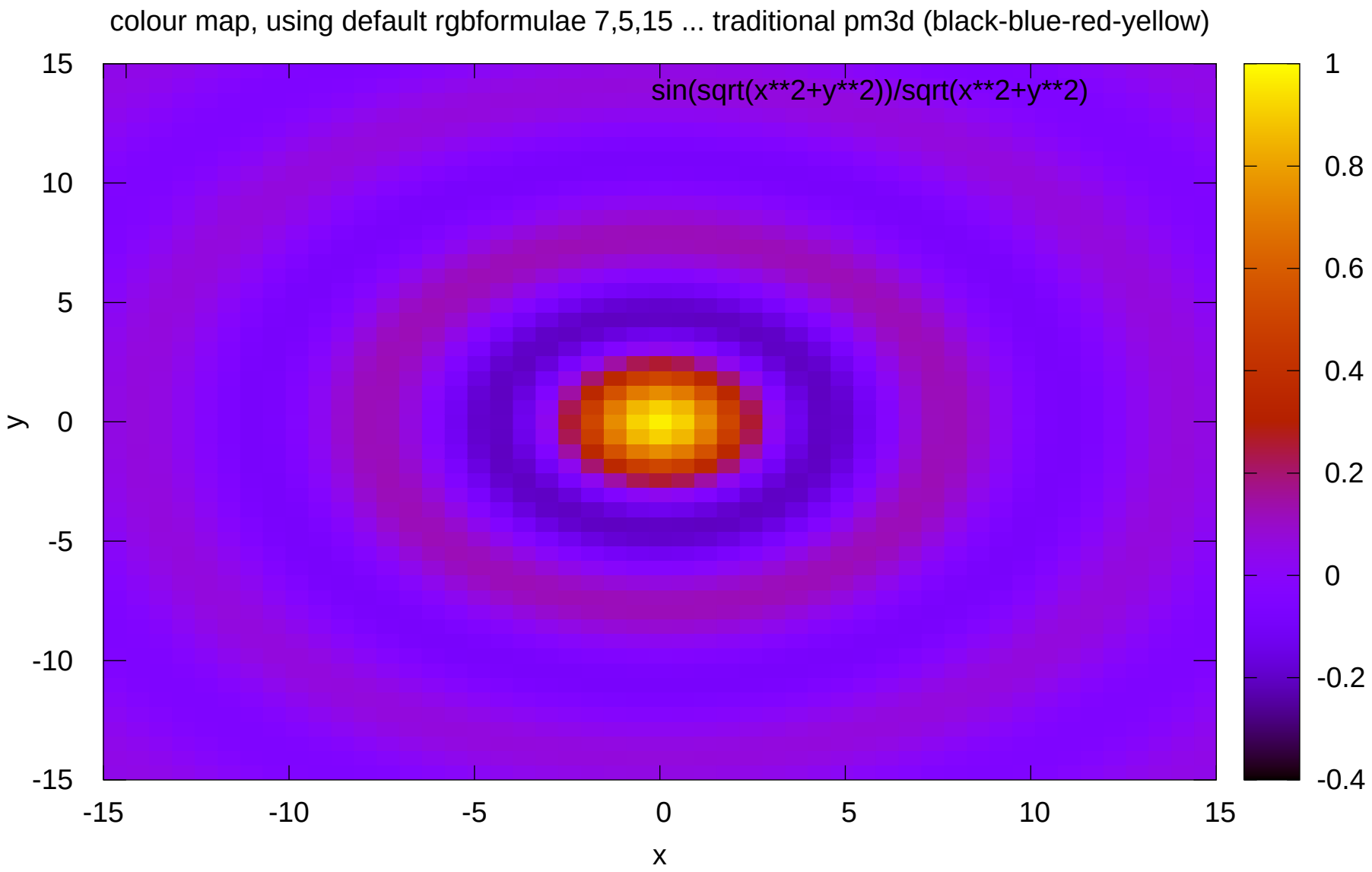

pm3d at b (bottom)
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x
y
colour gradient

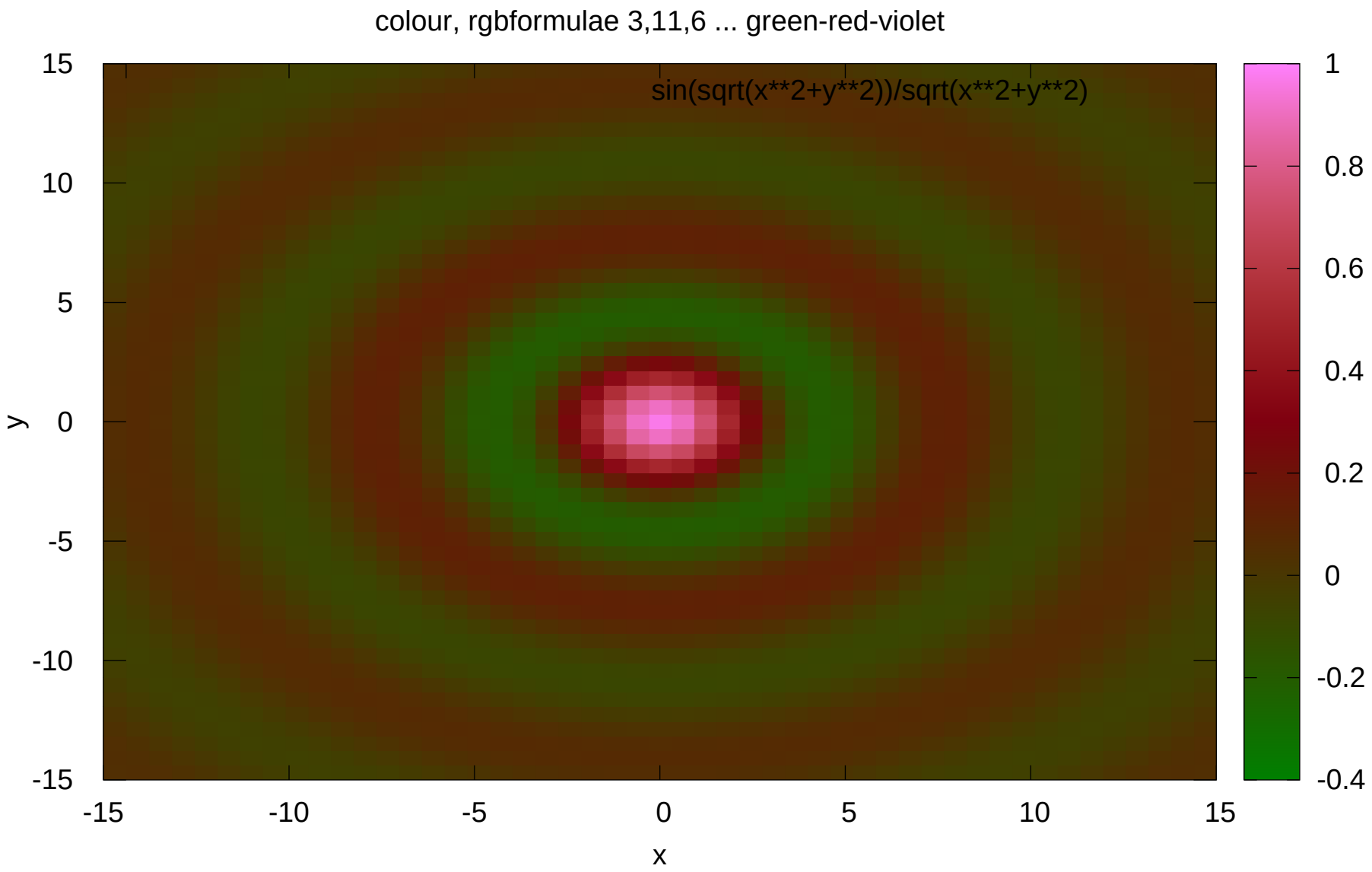

unset surface; set pm3d at st (surface and top)
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x
-15
-10
-5
0
5
10
15
y
colour gradient

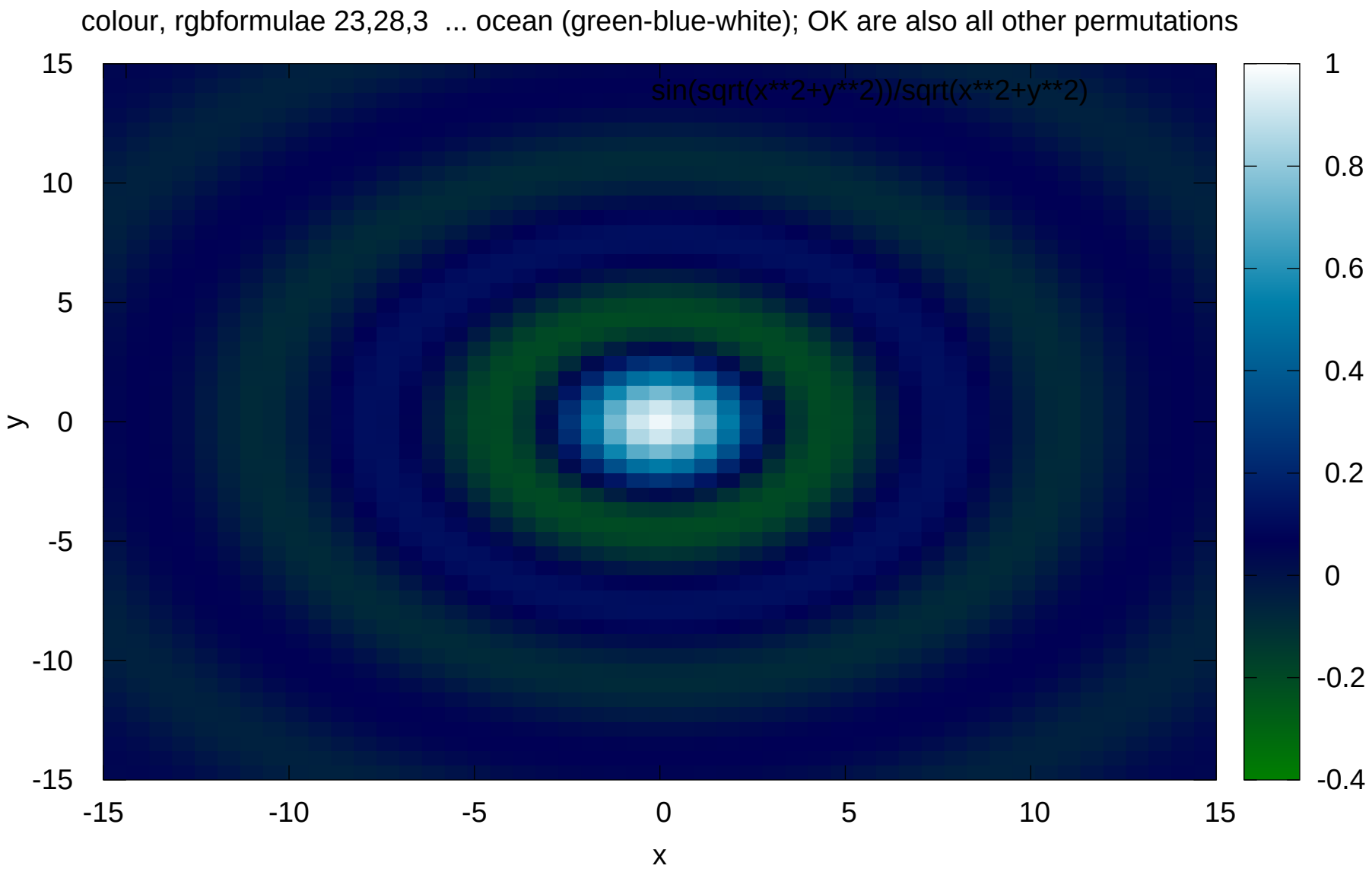

set pm3d at bstbst (funny combination, only for screen or postscript)
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x
y

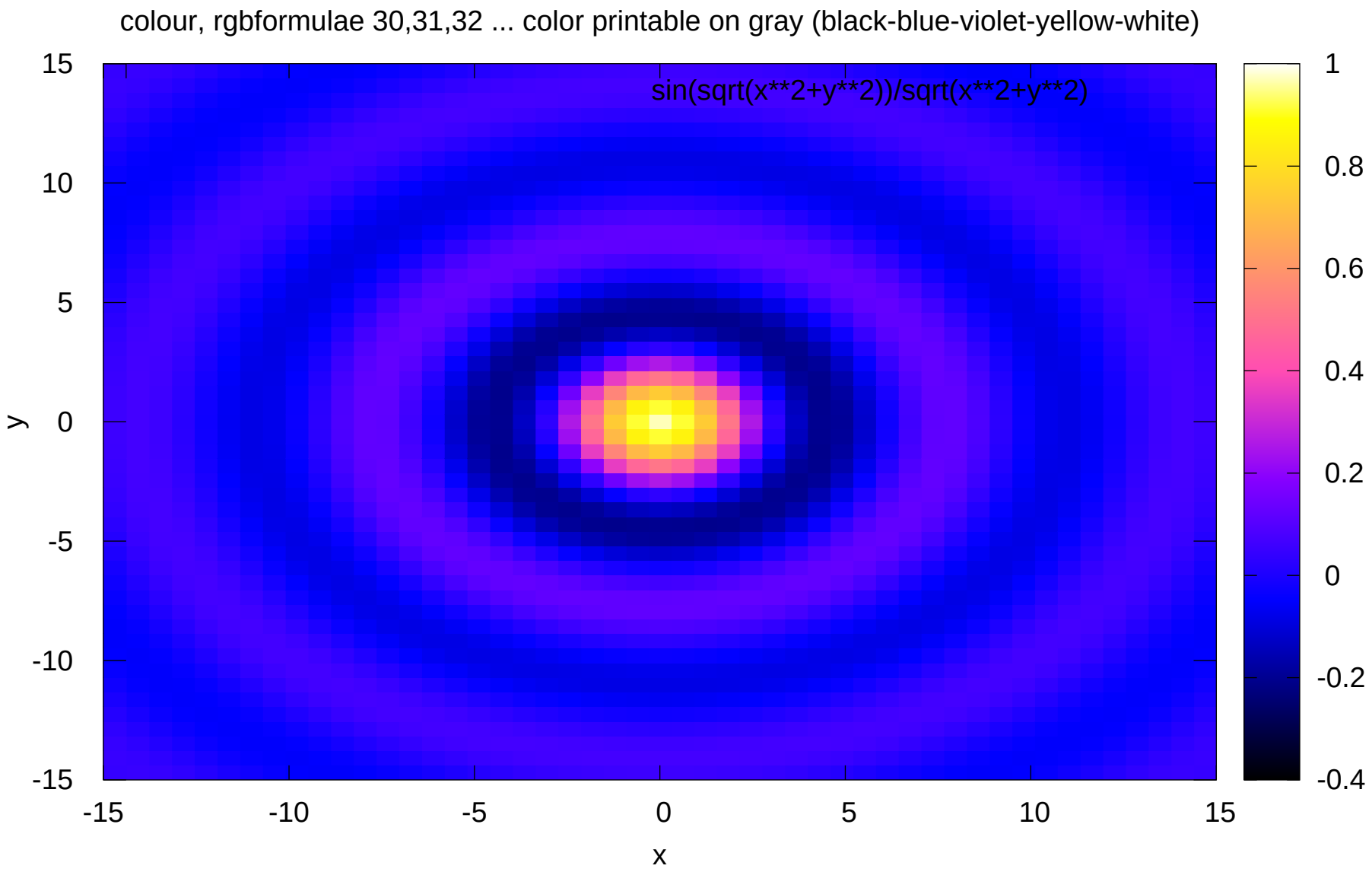

gray map
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

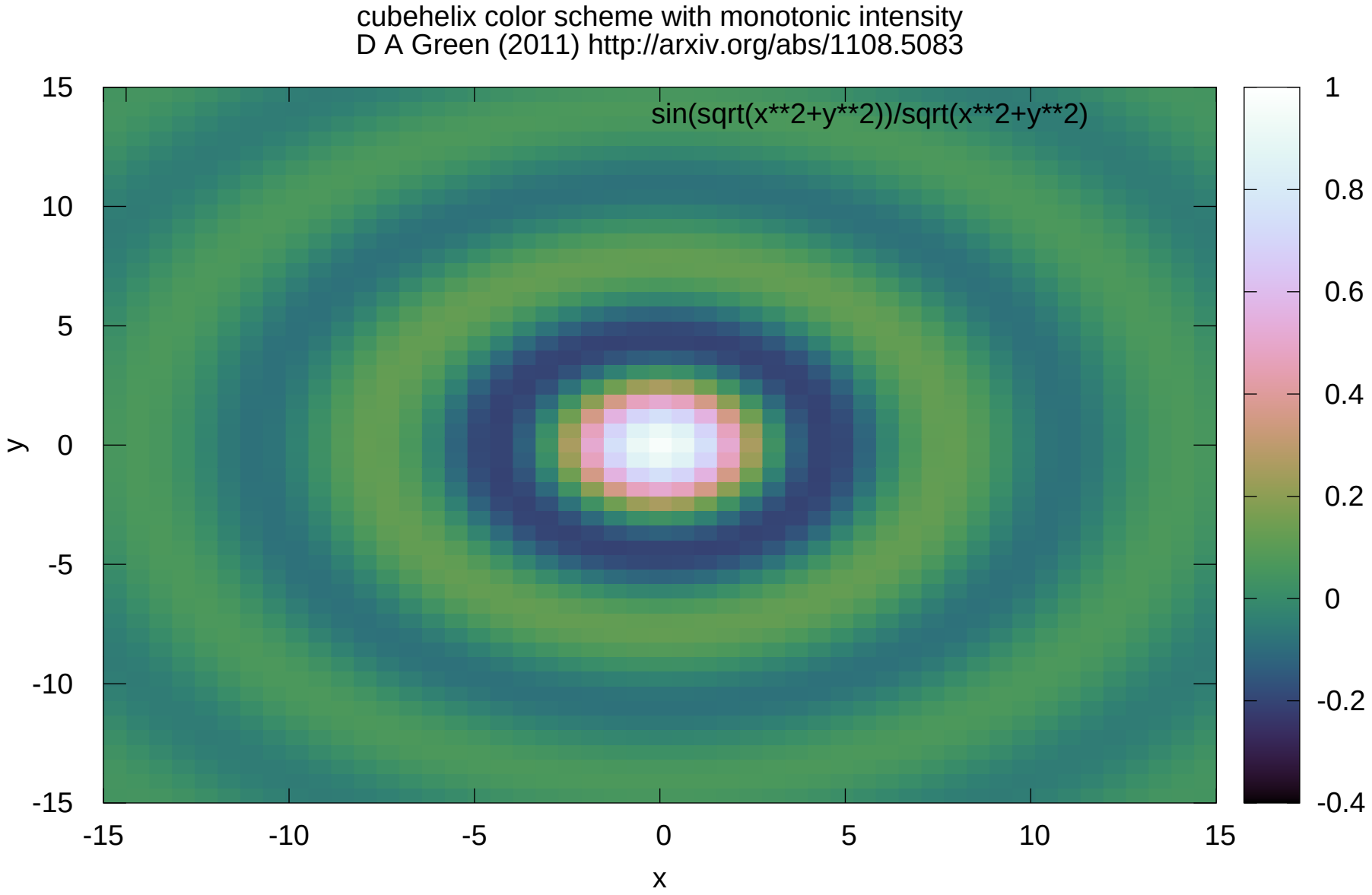

gray map, negative
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

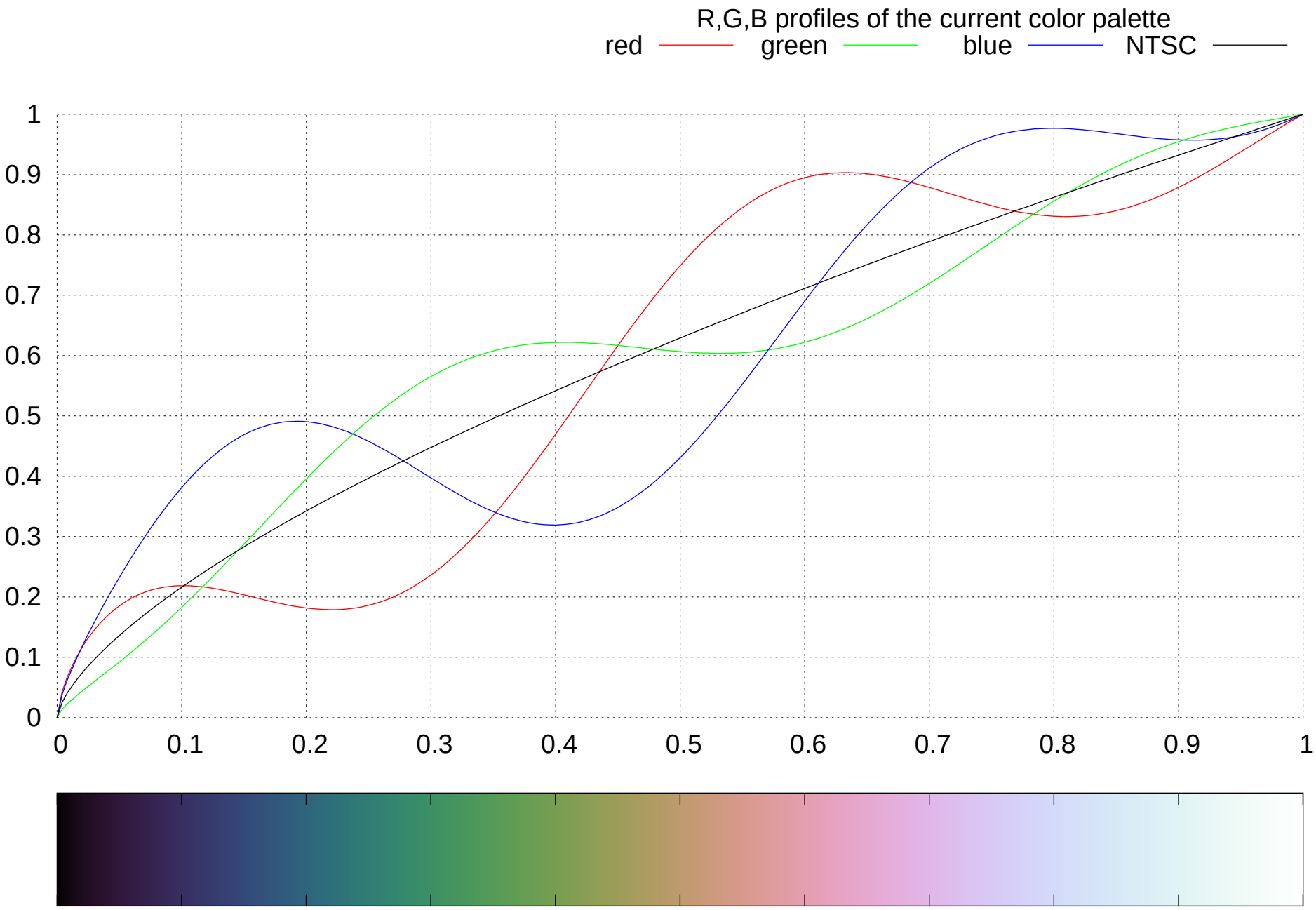

colour map, using default rgbformulae 7,5,15 ... traditional pm3d (black-blue-red-yellow)
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

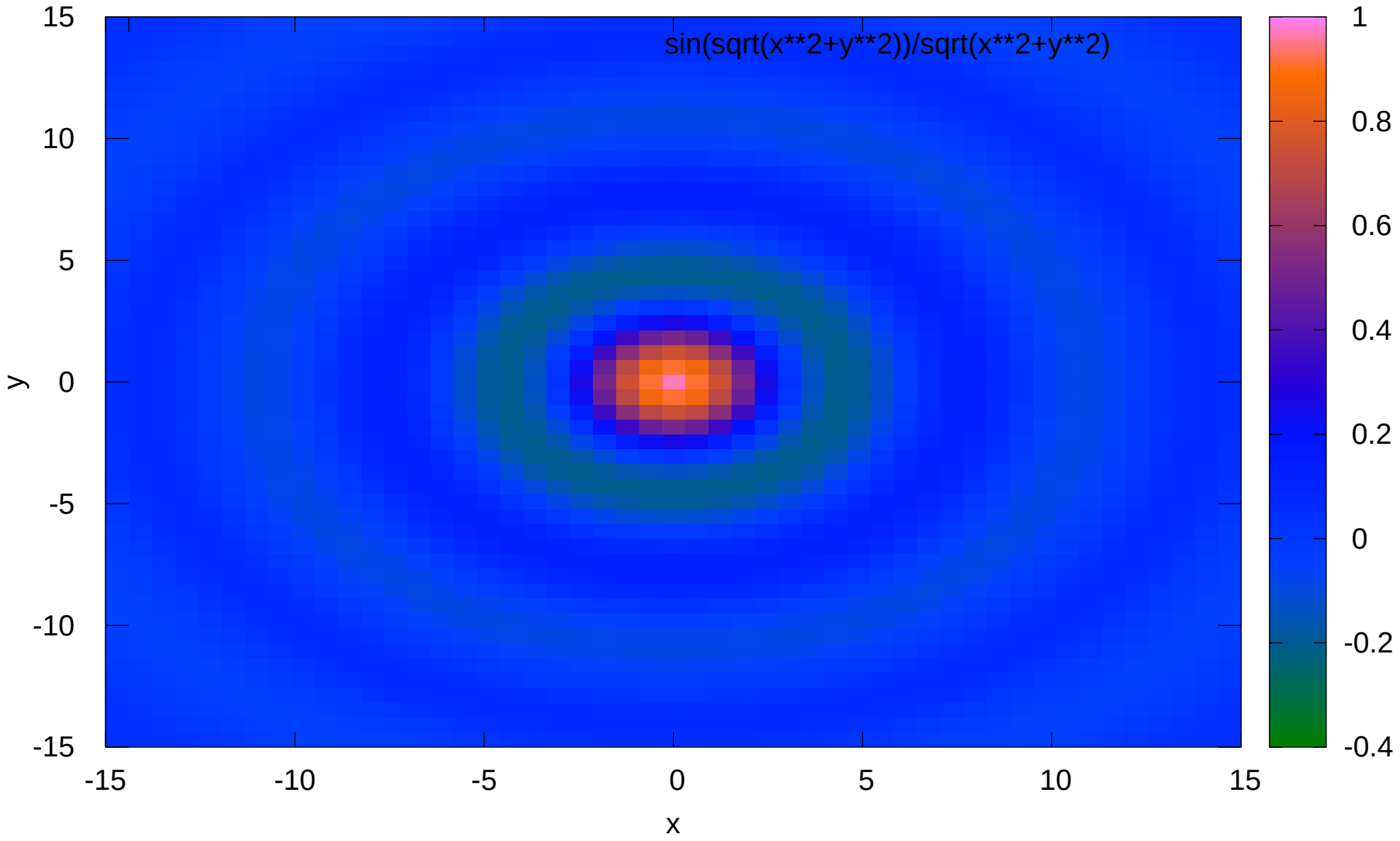

colour, rgbformulae 3,11,6 ... green-red-violet
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

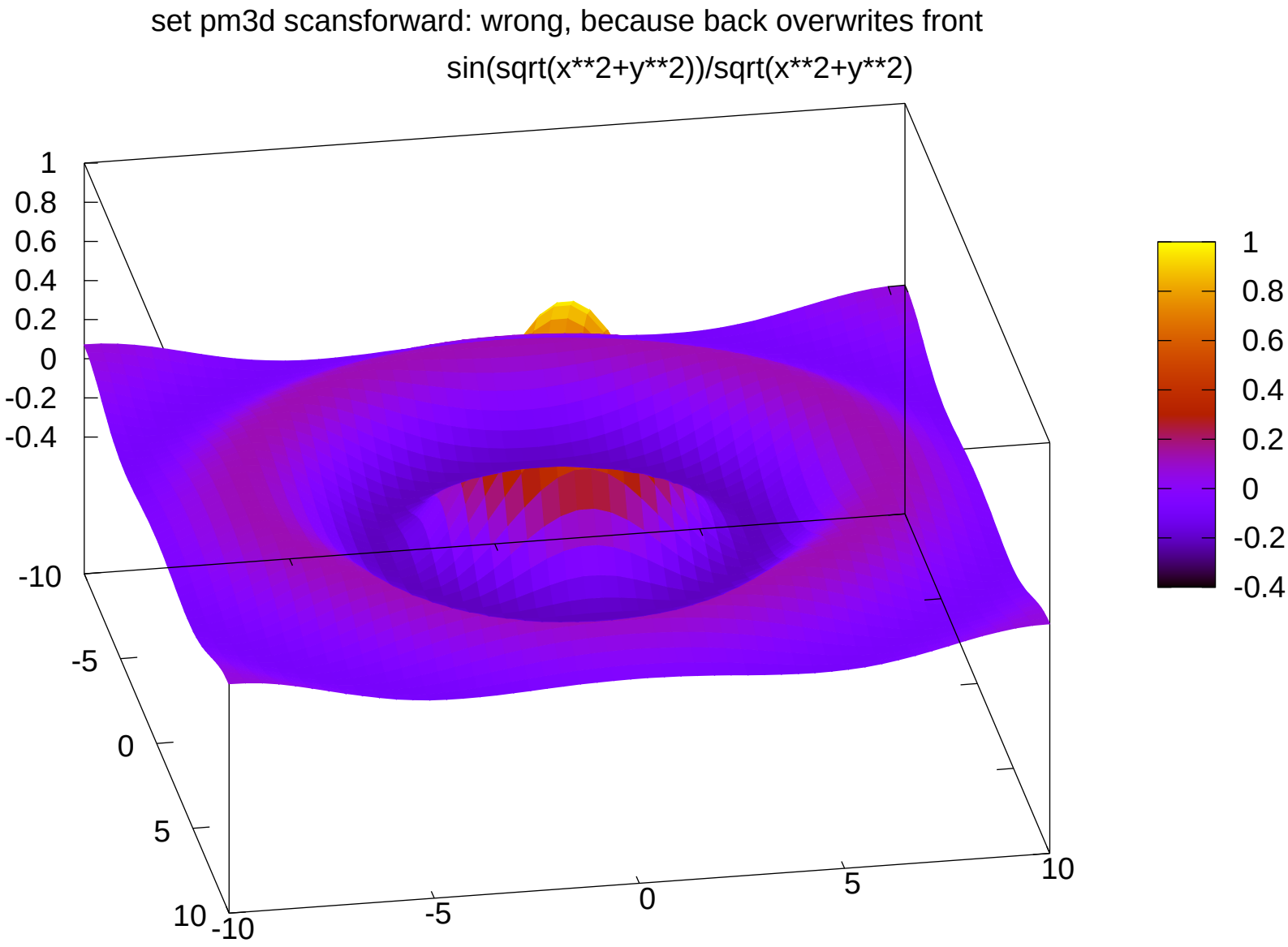

colour, rgbformulae 23,28,3 ... ocean (green-blue-white); OK are also all other permutations
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

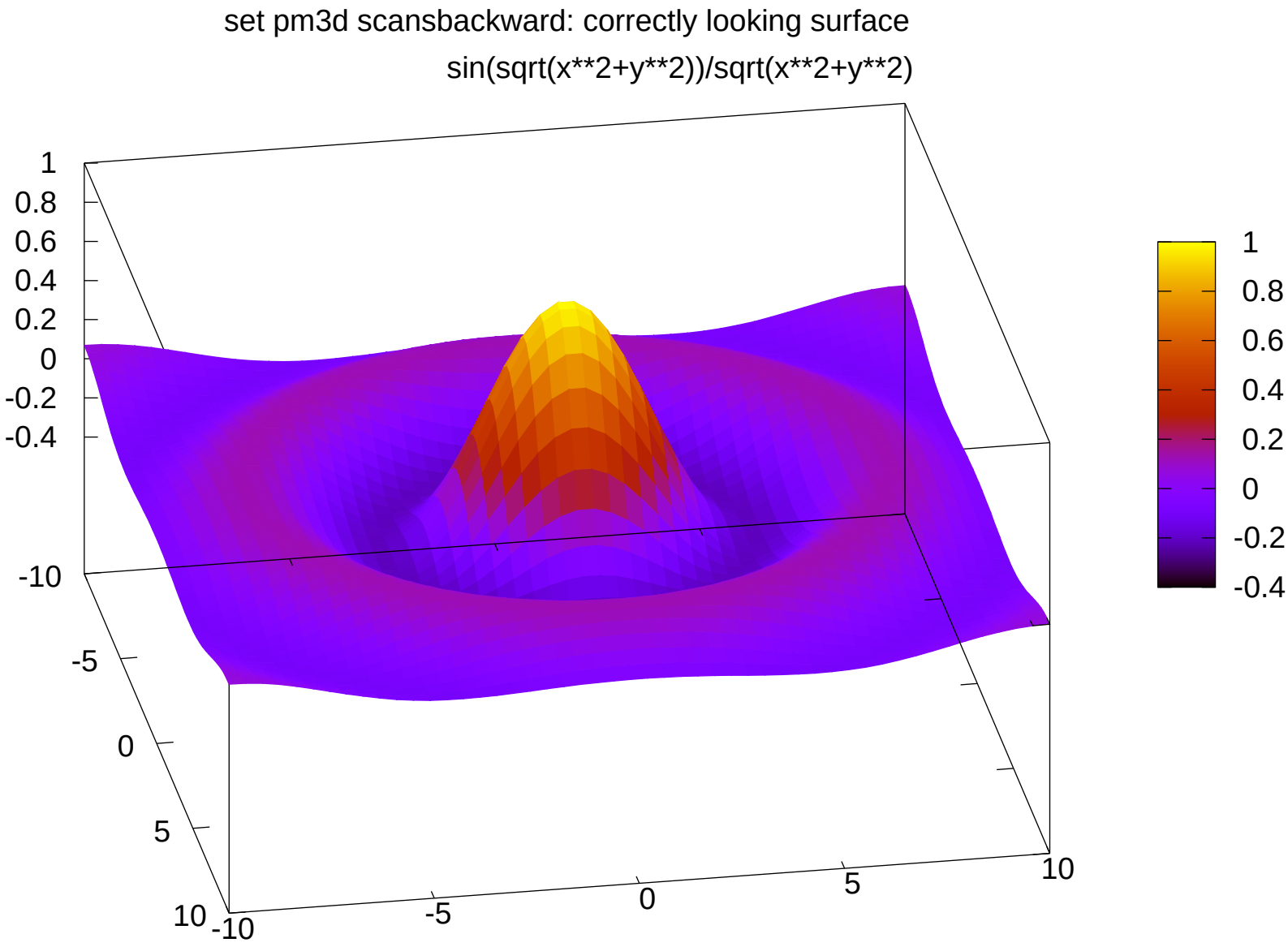

colour, rgbformulae 30,31,32 ... color printable on gray (black-blue-violet-yellow-white)
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

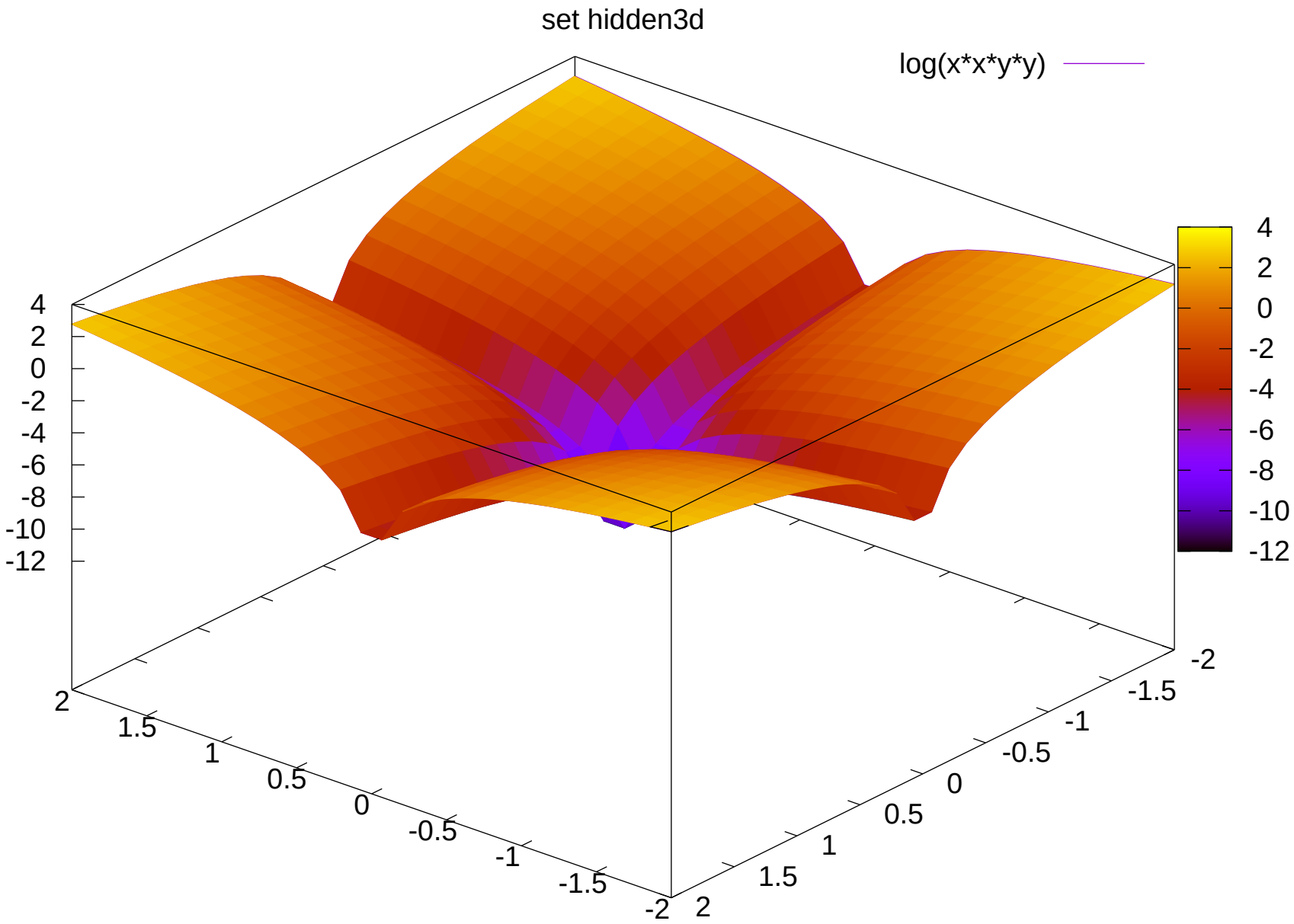

cubehelix color scheme with monotonic intensity
D A Green (2011) http://arxiv.org/abs/1108.5083
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

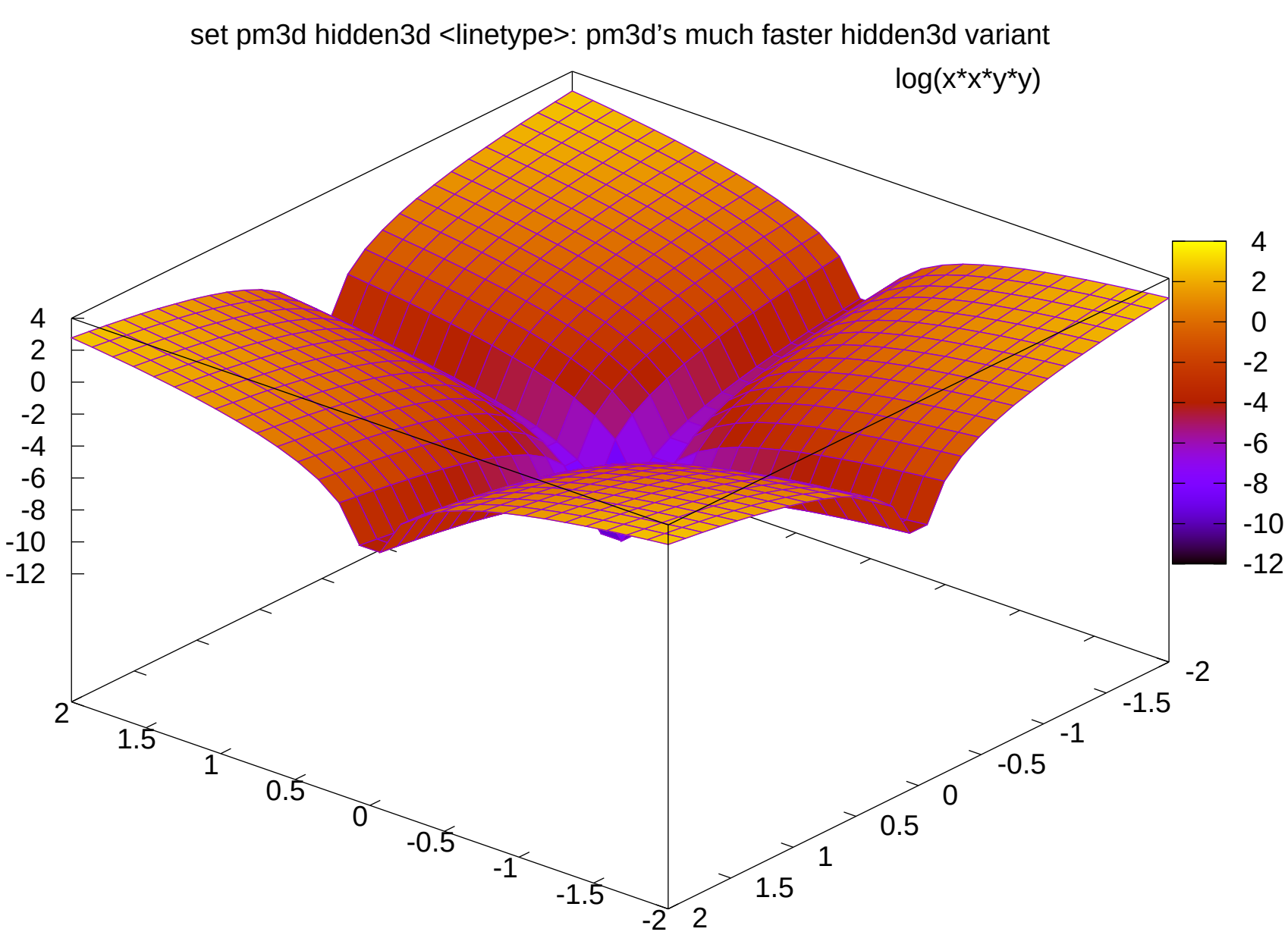

R,G,B profiles of the current color palette
red
green
blue
NTSC
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

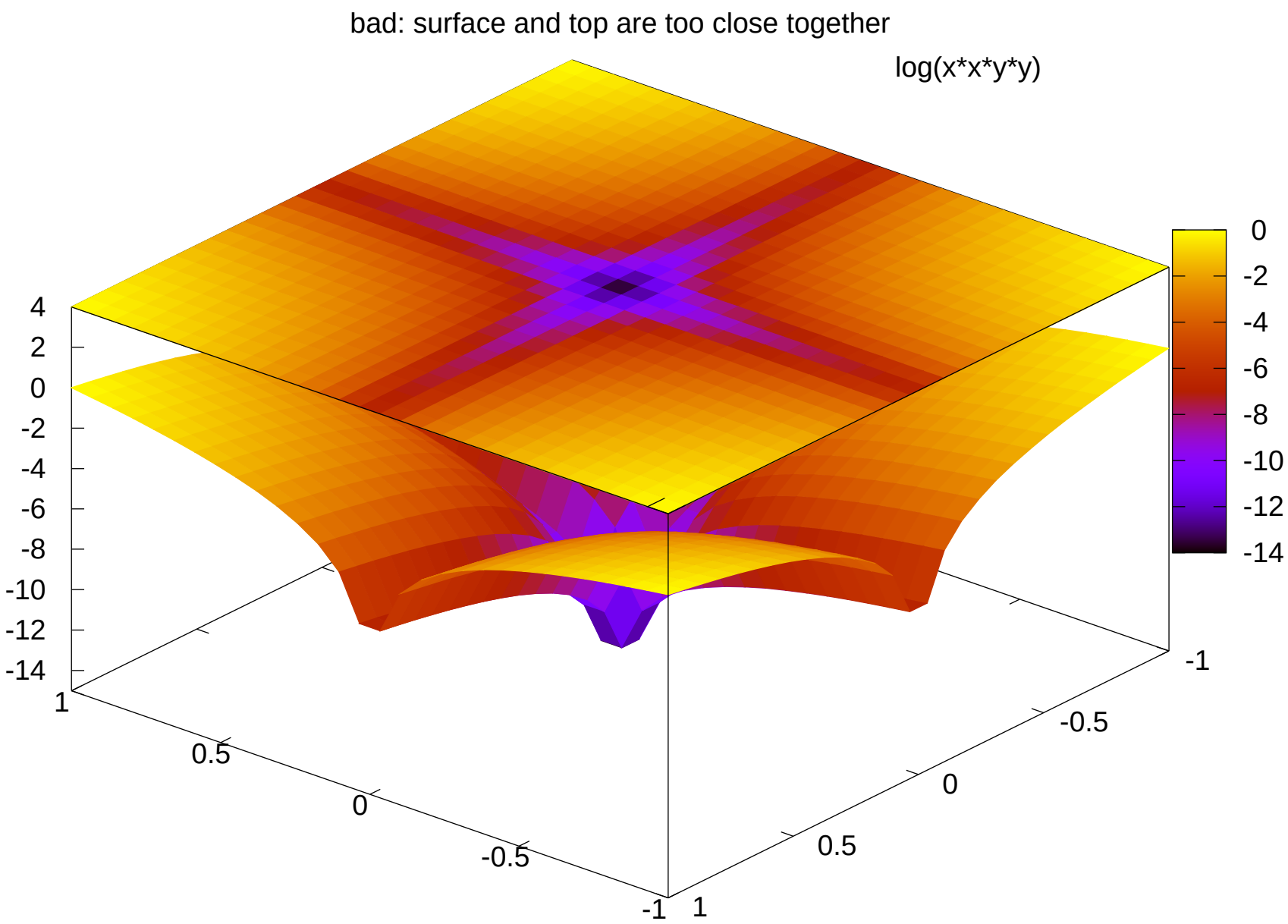

rgbformulae 31,-11,32: negative formula number=inverted color
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

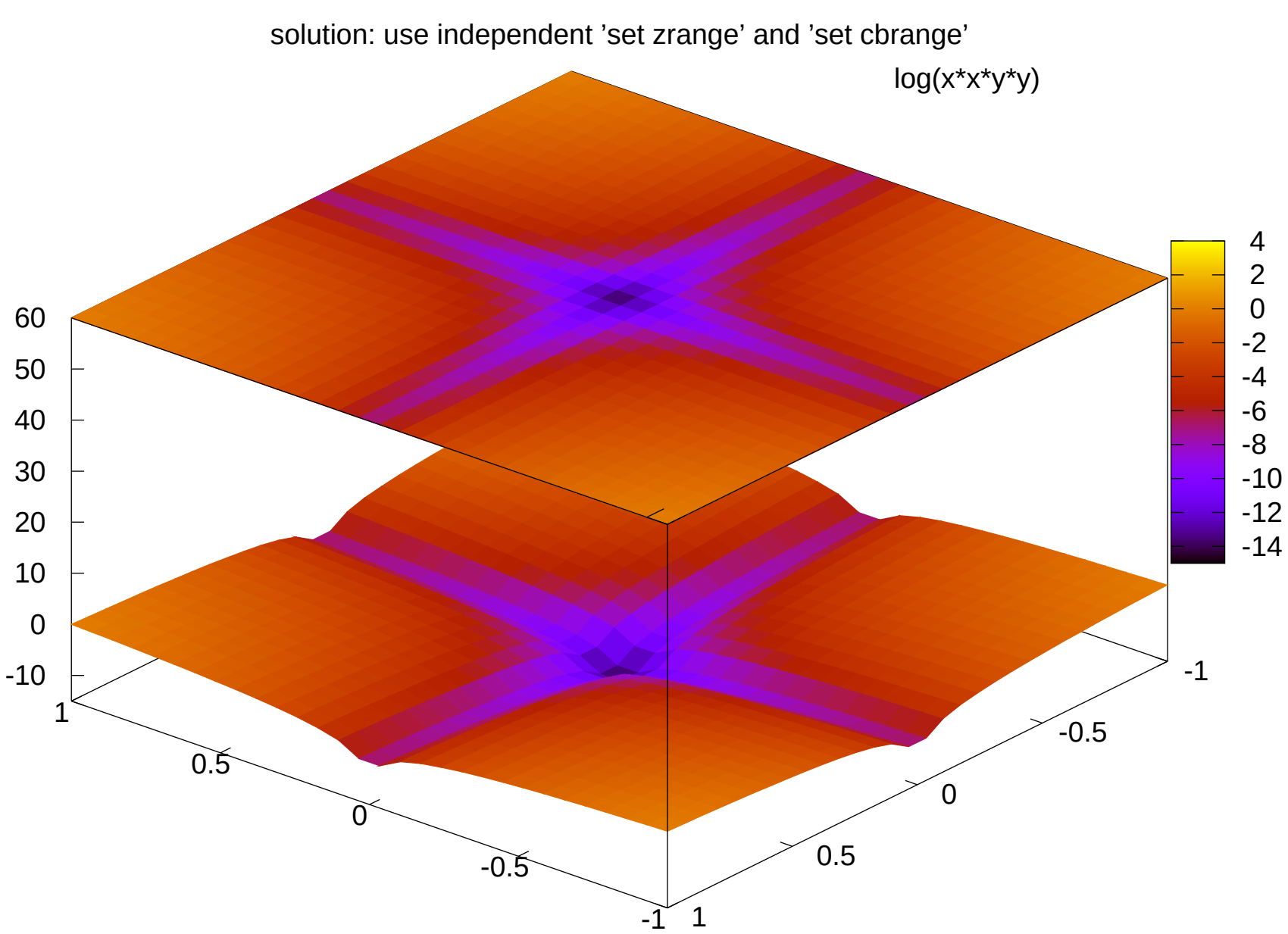

set pm3d scansforward: wrong, because back overwrites front
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10
-10
-5
0
5
10
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

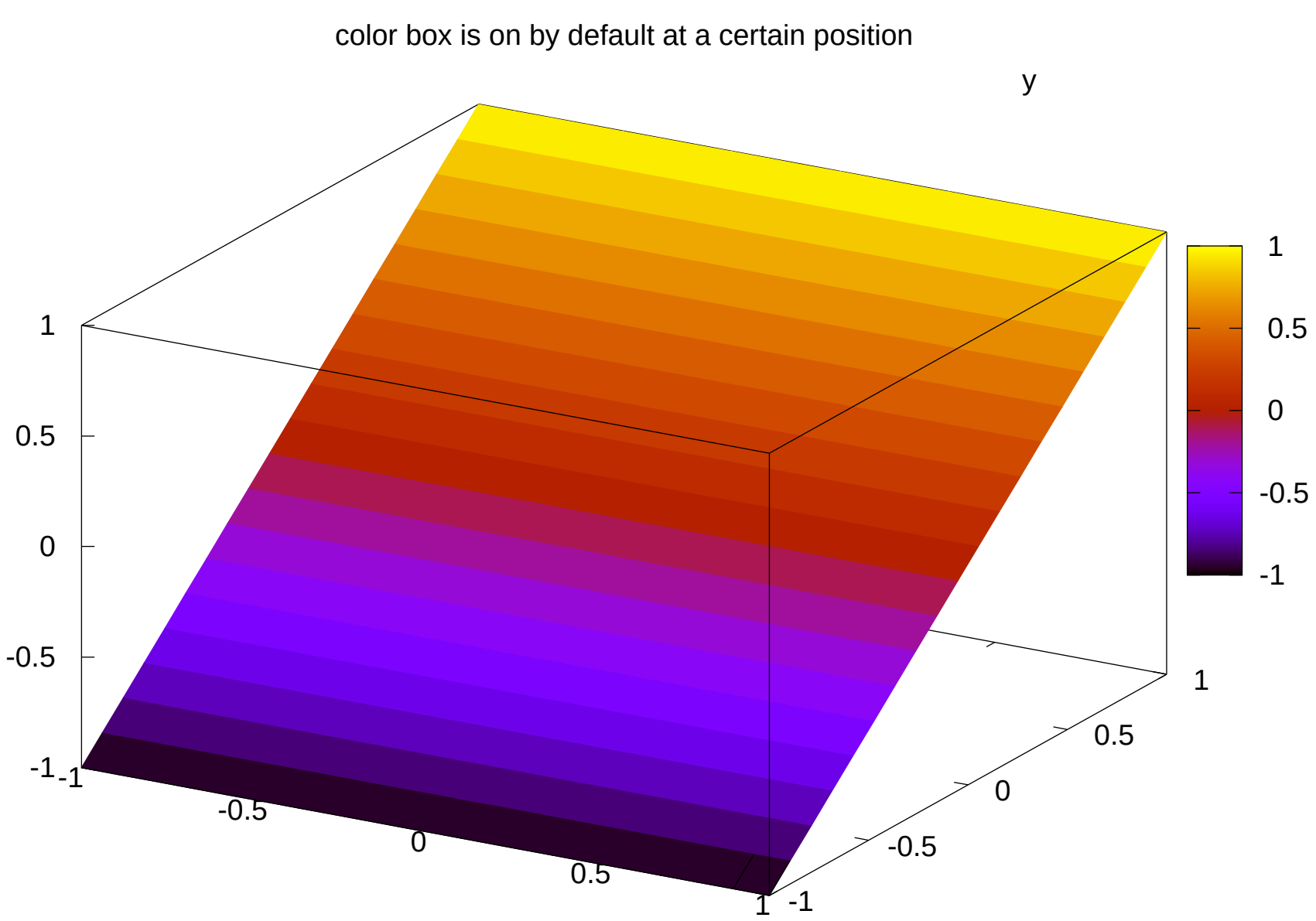

set pm3d scansbackward: correctly looking surface
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10
-10
-5
0
5
10
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

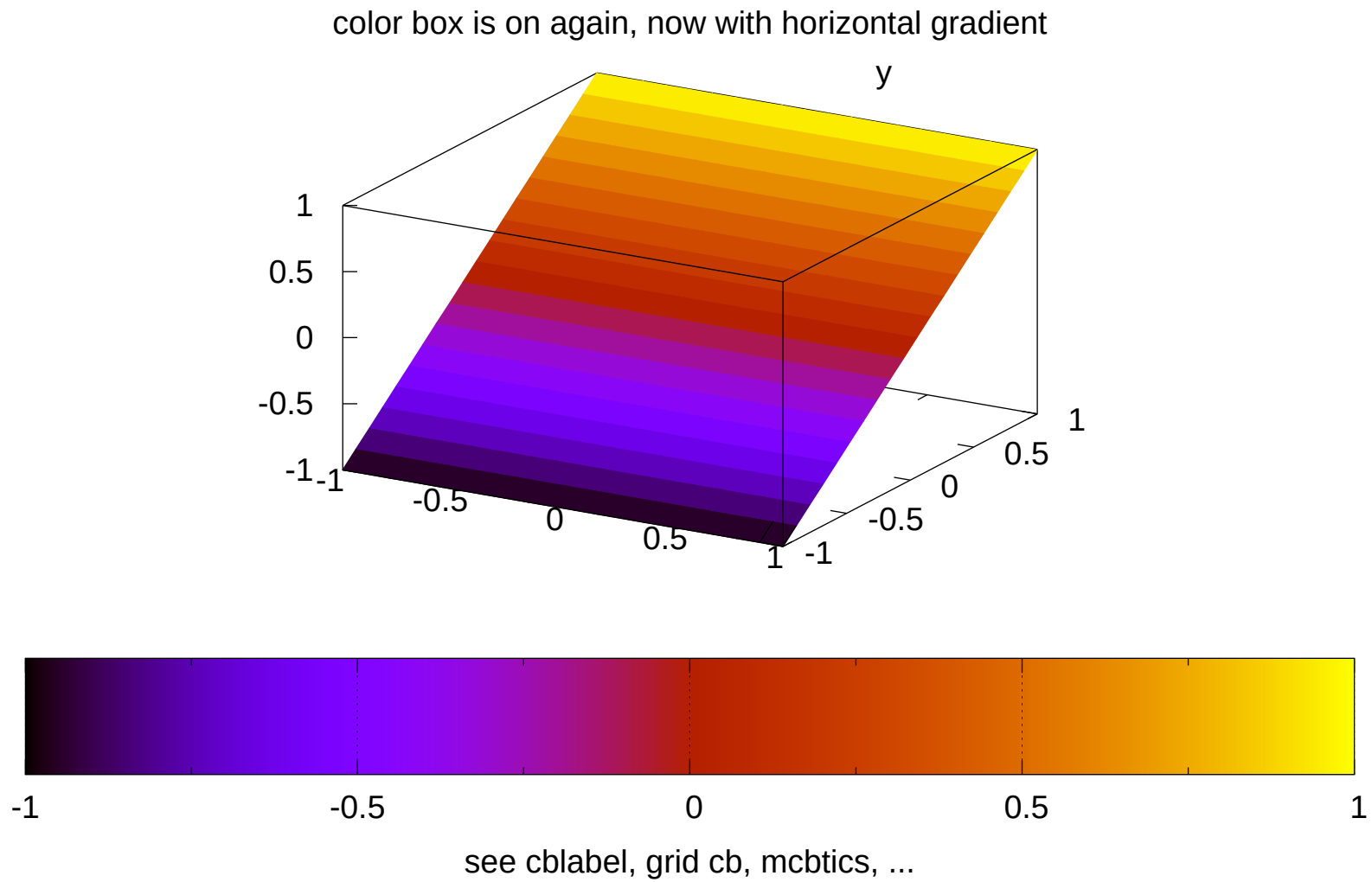

set hidden3d
log(x*x*y*y)
4
2
0
-2
-4
-6
-8
-10
-12
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2

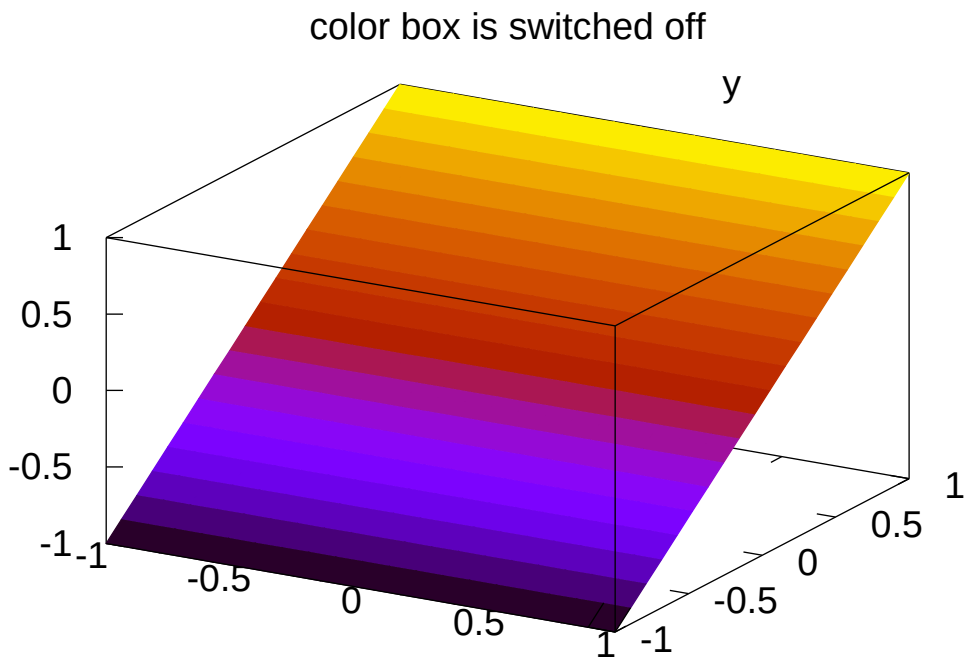

set pm3d hidden3d <linetype>: pm3d's much faster hidden3d variant
log(x*x*y*y)
4
2
0
-2
-4
-6
-8
-10
-12
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2

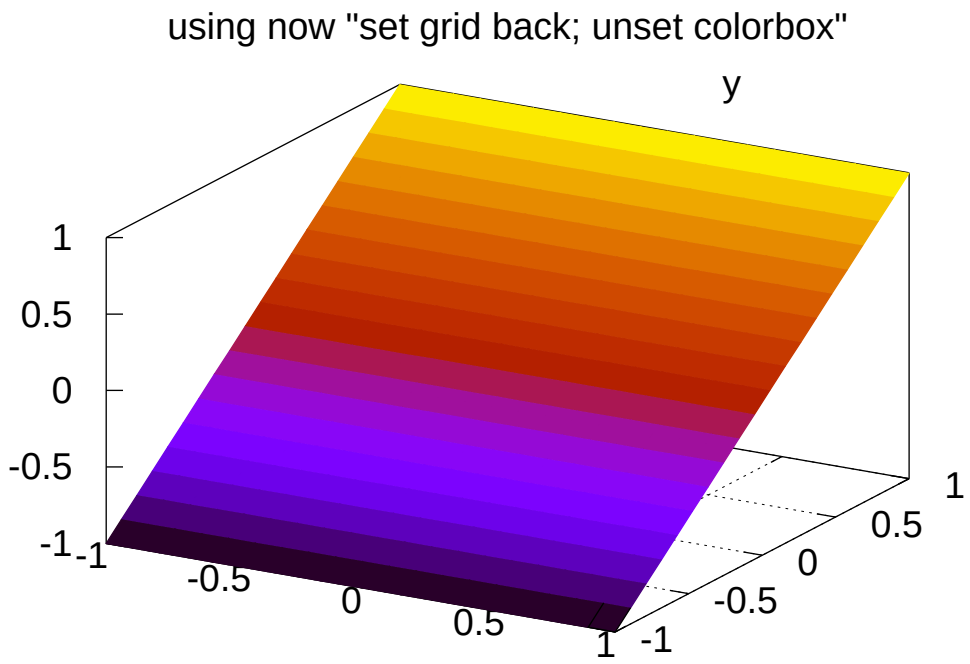

bad: surface and top are too close together
log(x*x*y*y)
4
2
0
-2
-4
-6
-8
-10
-12
-14
1
0.5
0
-0.5
-1
1
0.5
0
-0.5
-1
0
-2
-4
-6
-8
-10
-12
-14

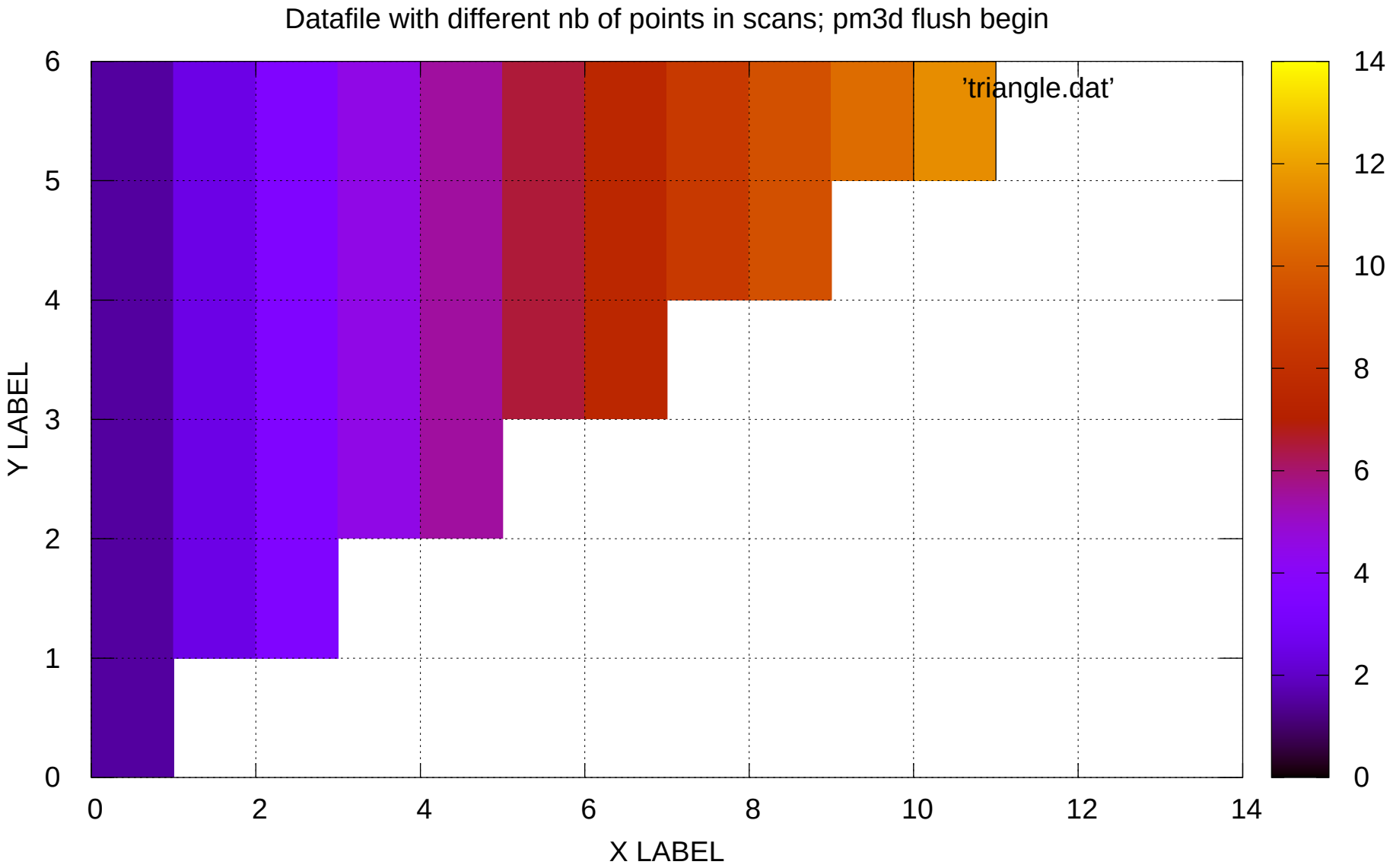

solution: use independent ’set zrange’ and ’set cbrange’
log(x*x*y*y)
60
50
40
30
20
10
0
-10
1
0.5
0
-0.5
-1
1
0.5
0
-0.5
-1
4
2
0
-2
-4
-6
-8
-10
-12
-14

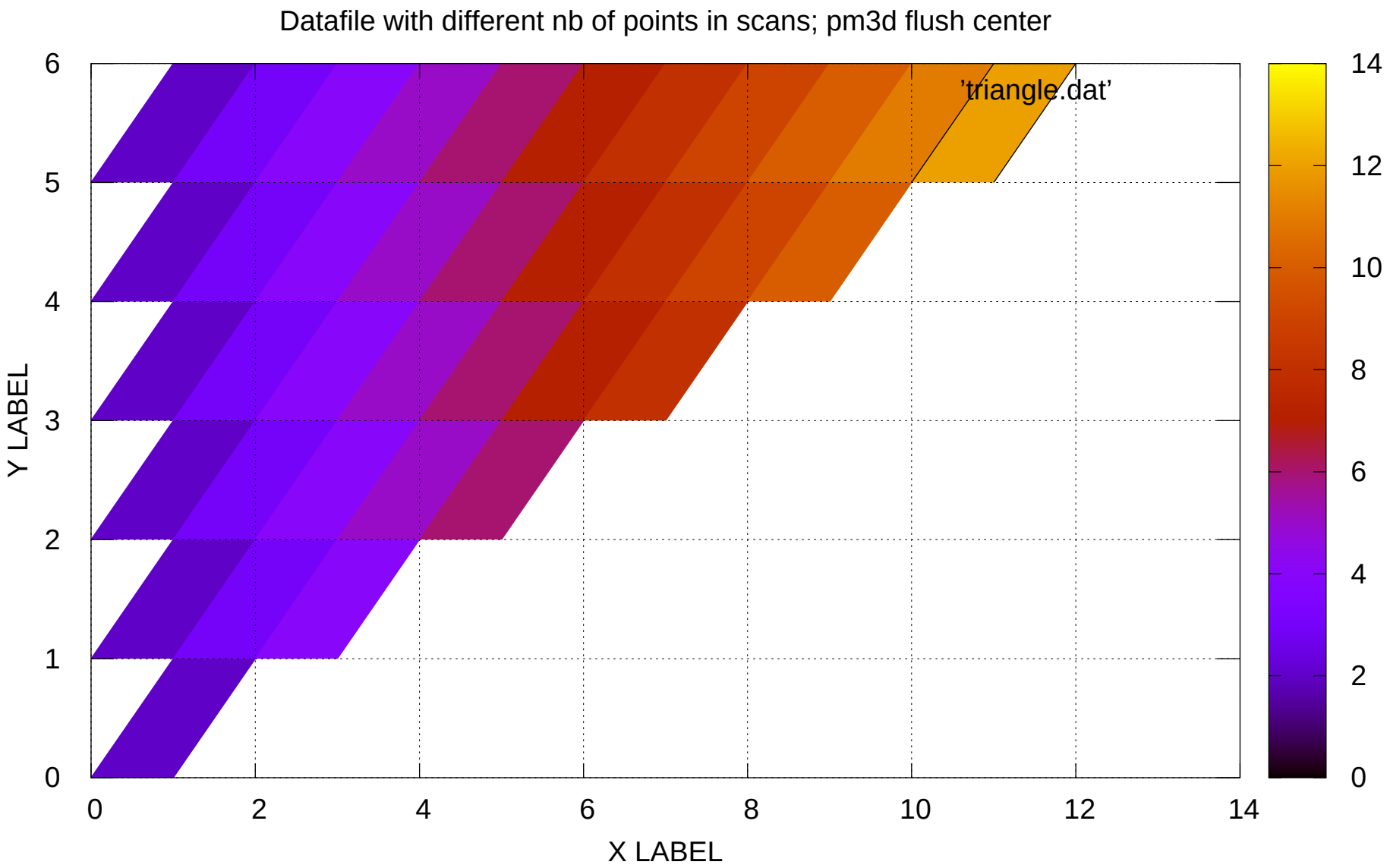

color box is on by default at a certain position
y
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
1
0.5
0
-0.5
-1
1
0.5
0
-0.5
-1

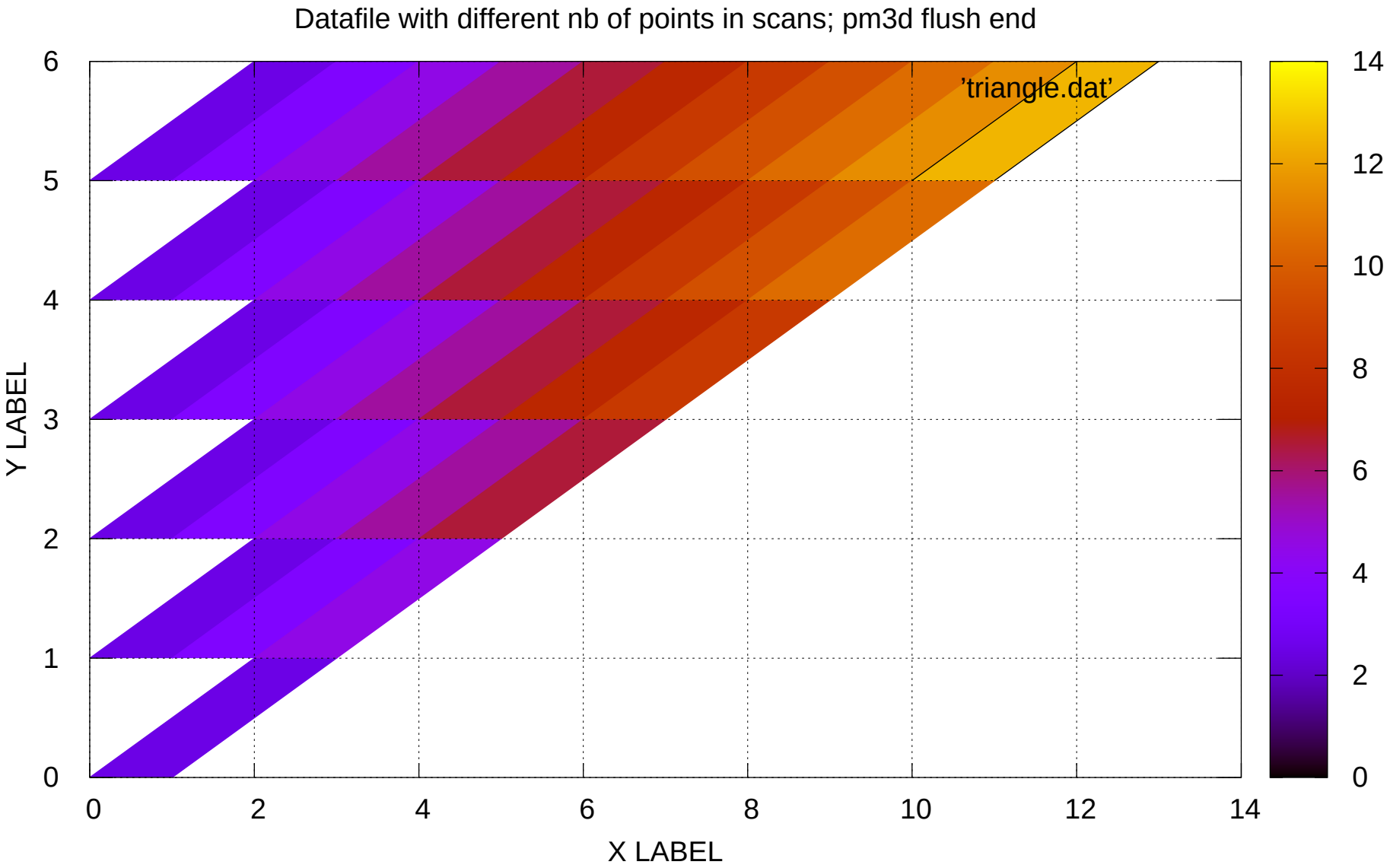

color box is on again, now with horizontal gradient
y
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1
see cblabel, grid cb, mcbtics, ...

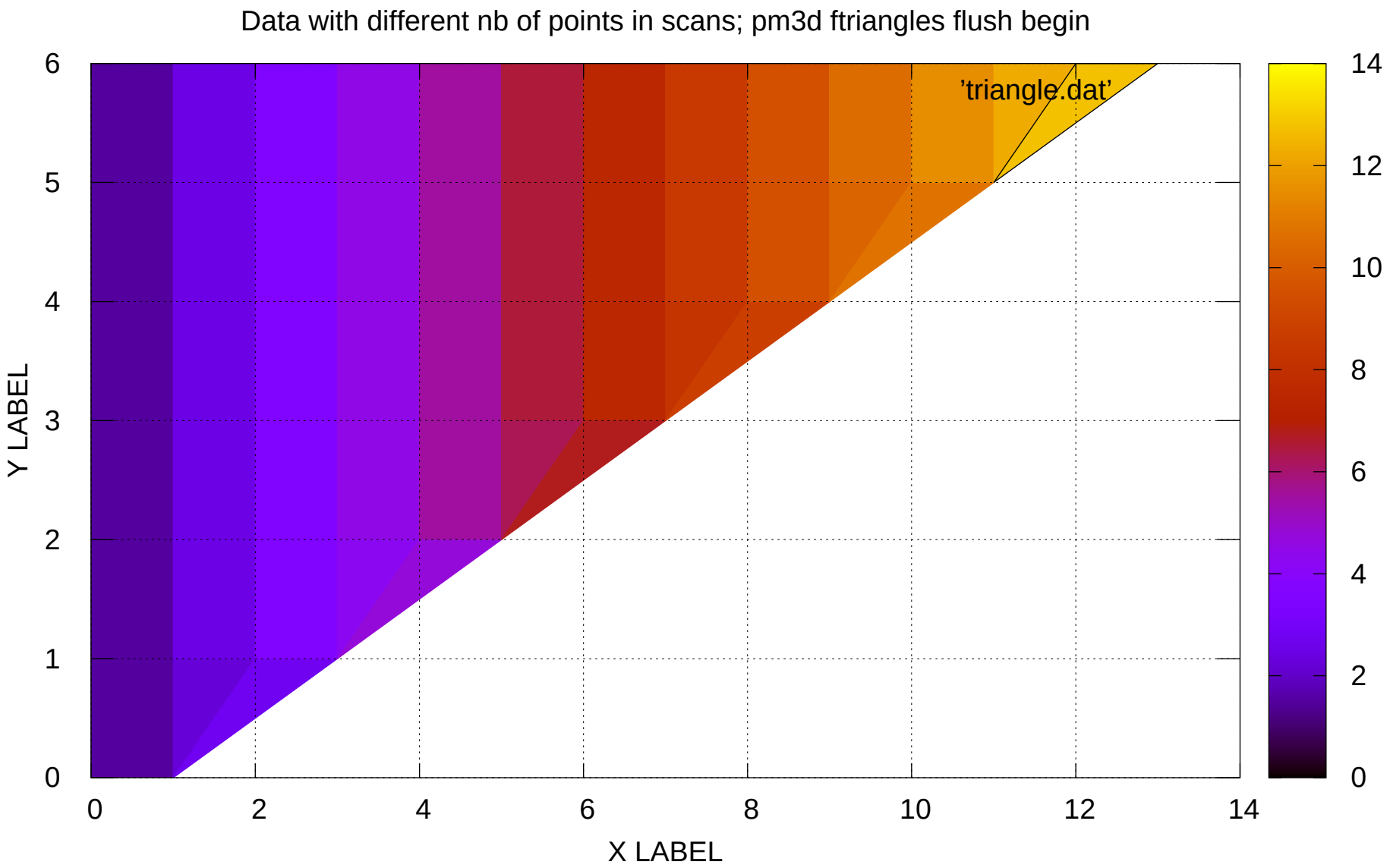

color box is switched off
y
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1

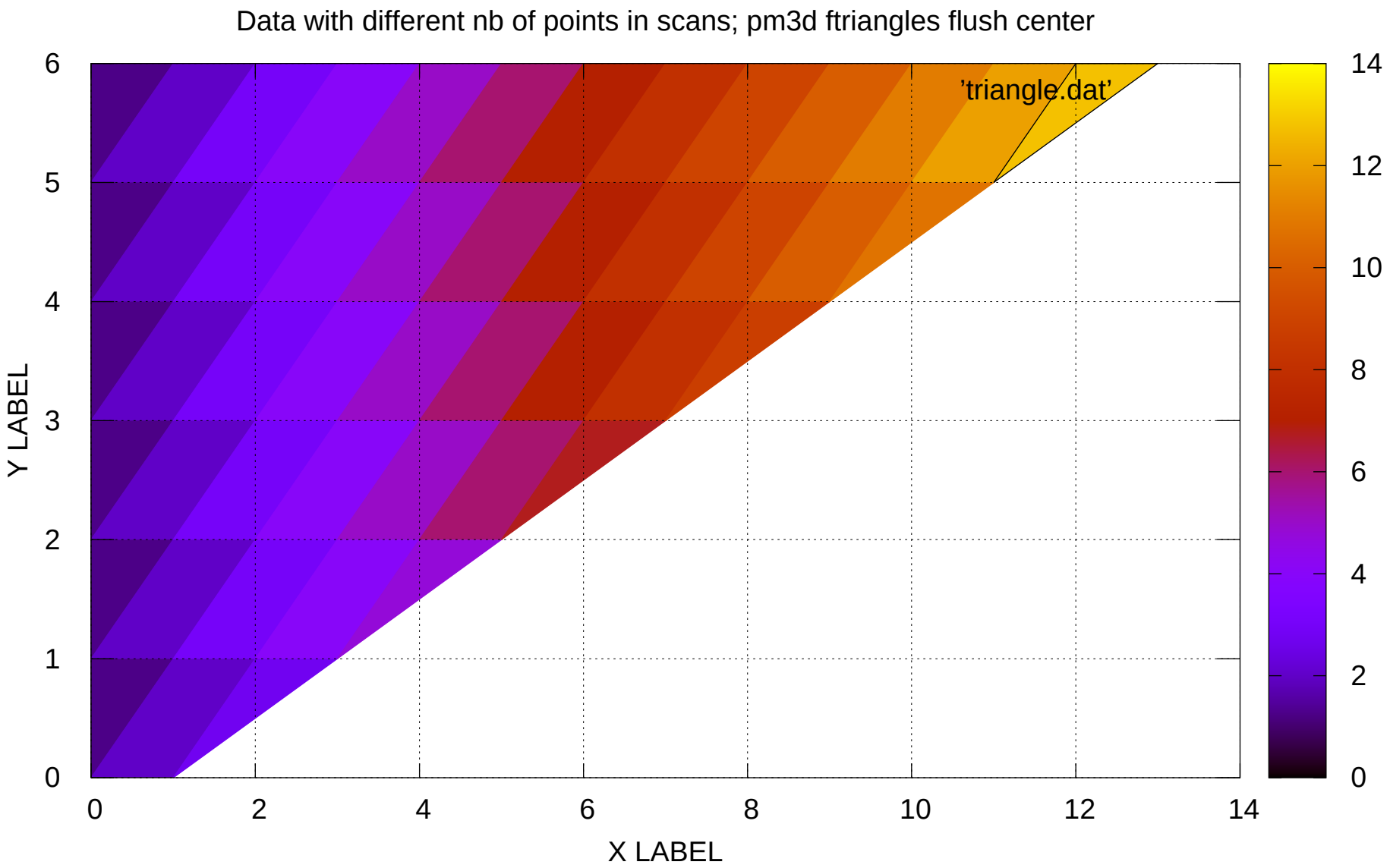

using now "set grid back; unset colorbox"
y
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1

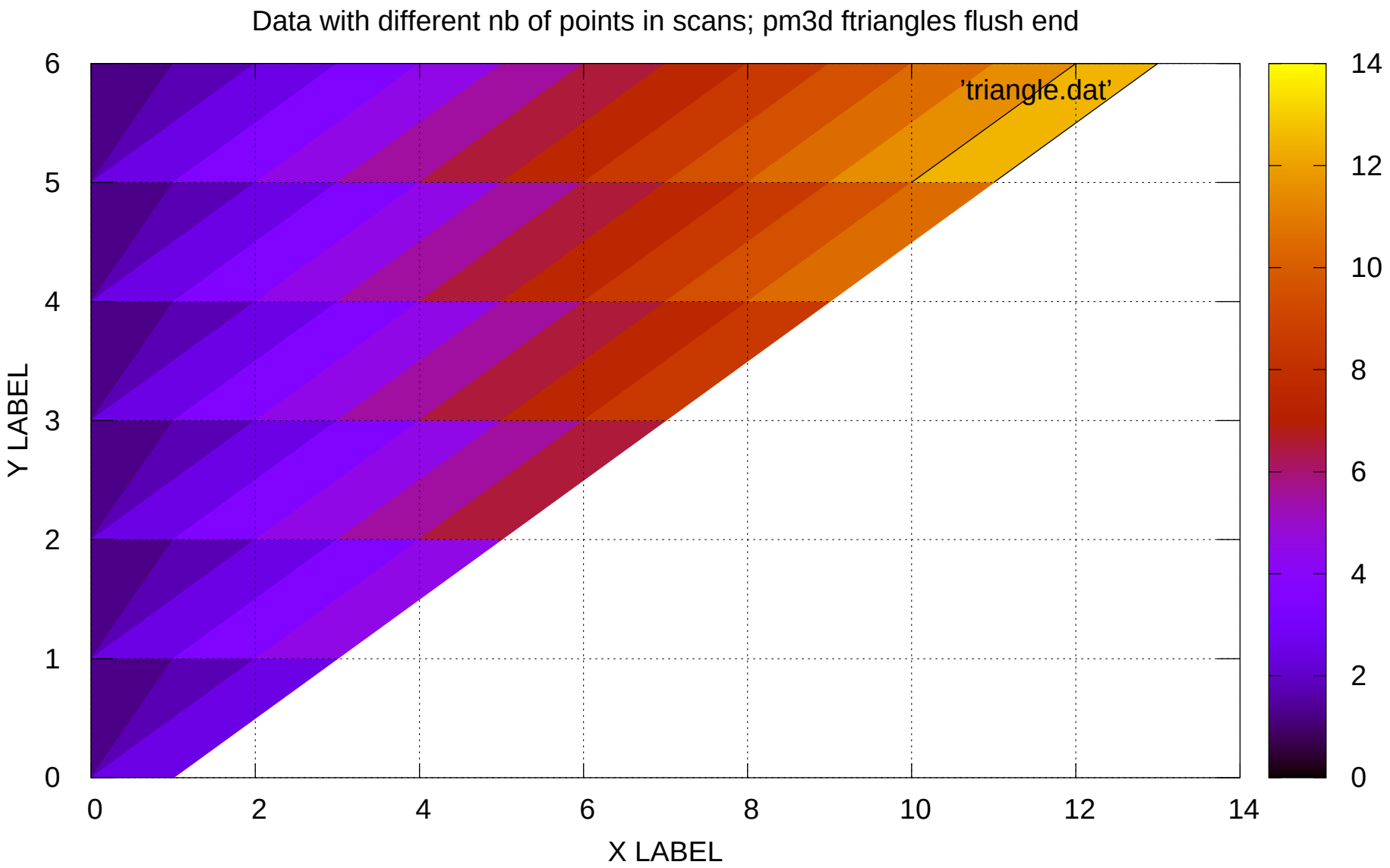

Datafile with different nb of points in scans; pm3d flush begin
'triangle.dat'
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

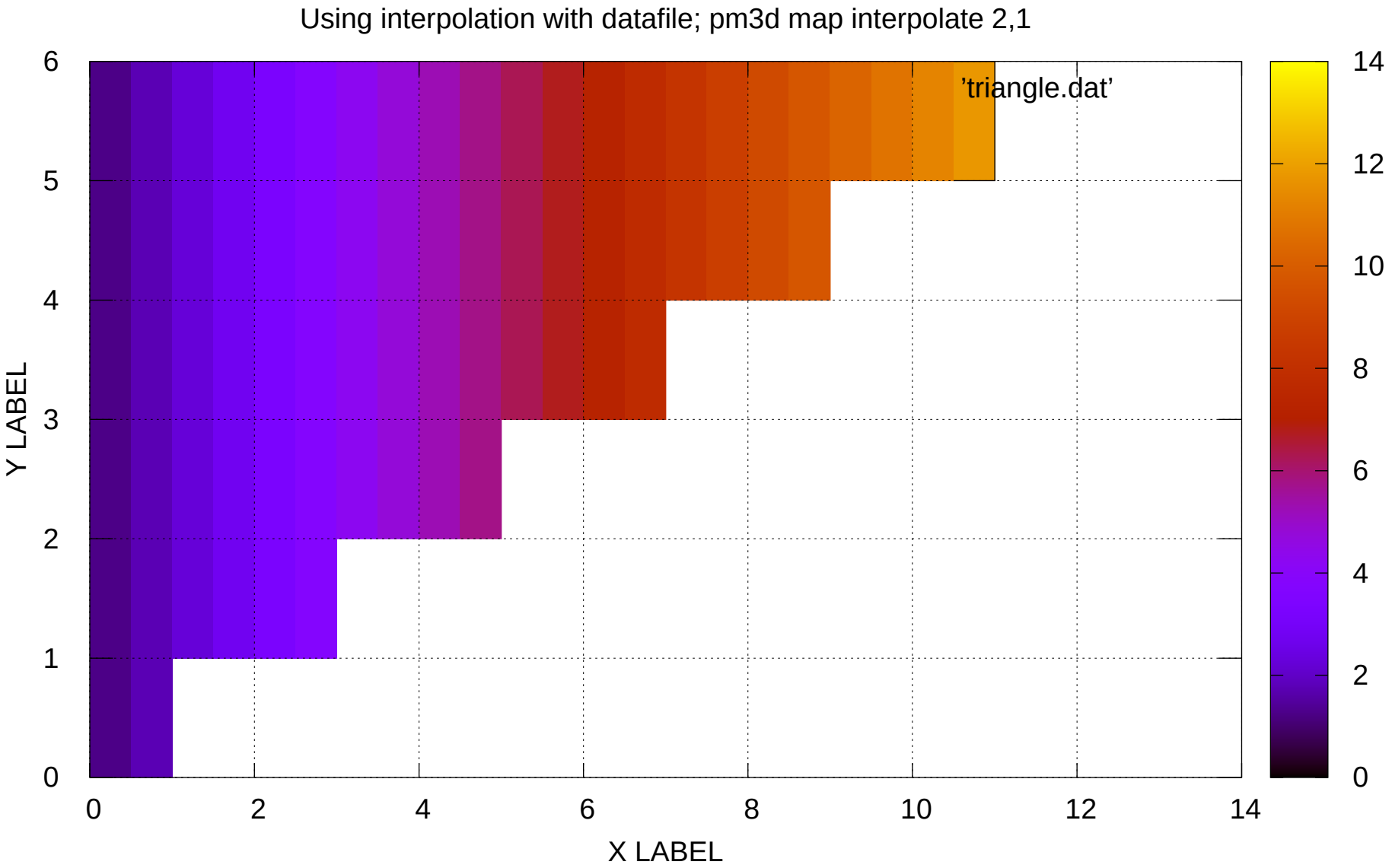

Datafile with different nb of points in scans; pm3d flush center
'triangle.dat'
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

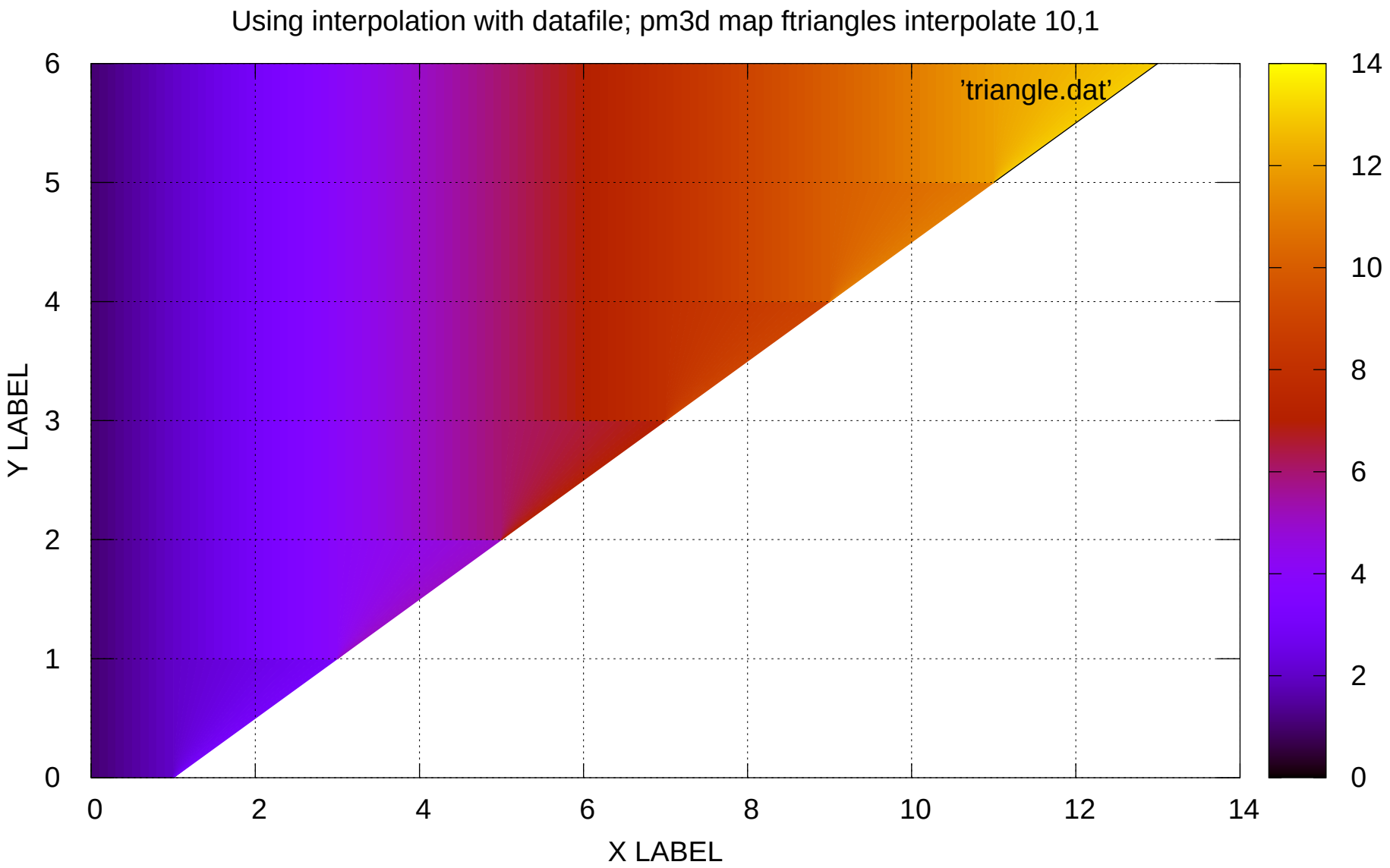

Datafile with different nb of points in scans; pm3d flush end
'triangle.dat'
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

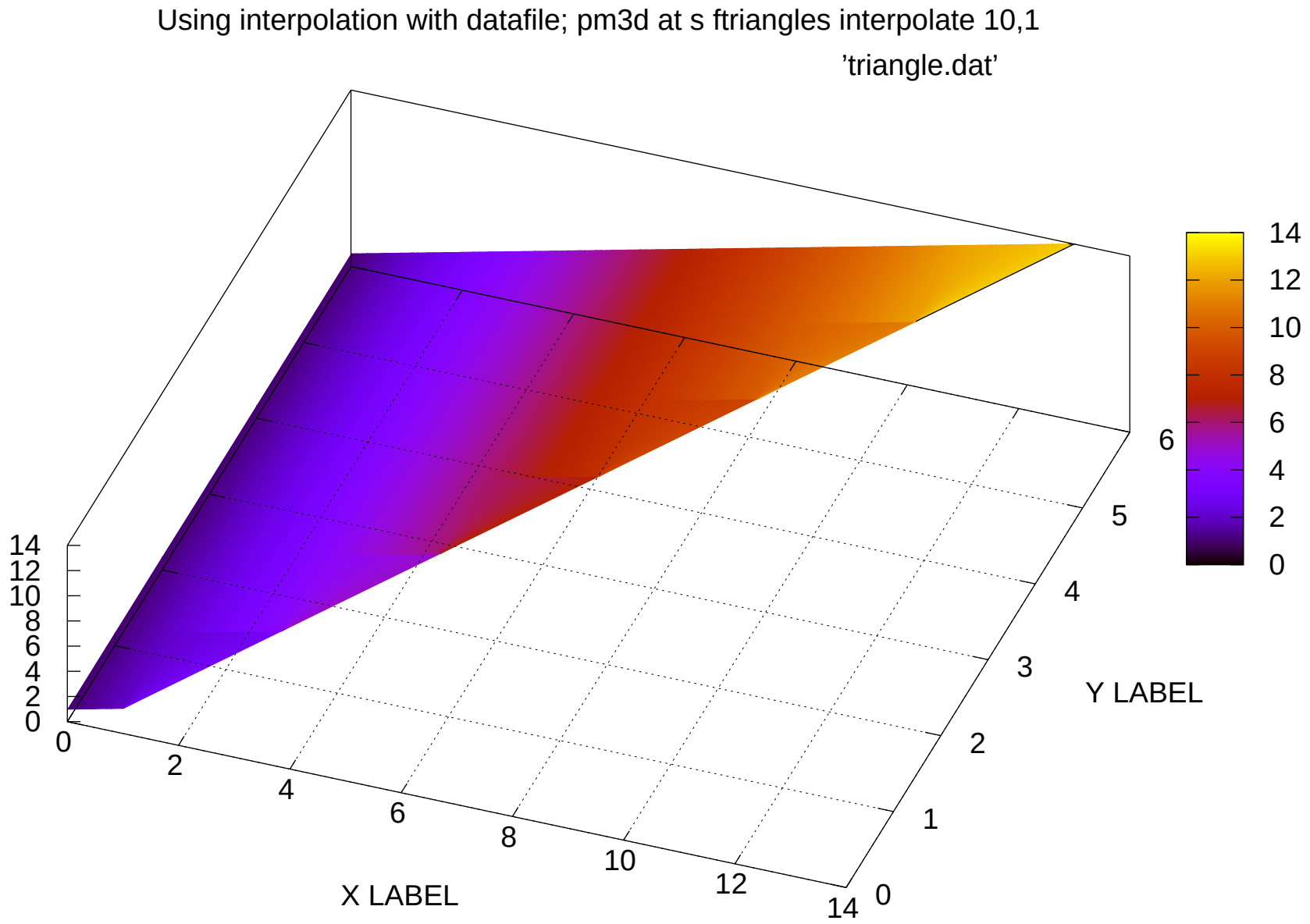

Data with different nb of points in scans; pm3d ftriangles flush begin
'triangle.dat'
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

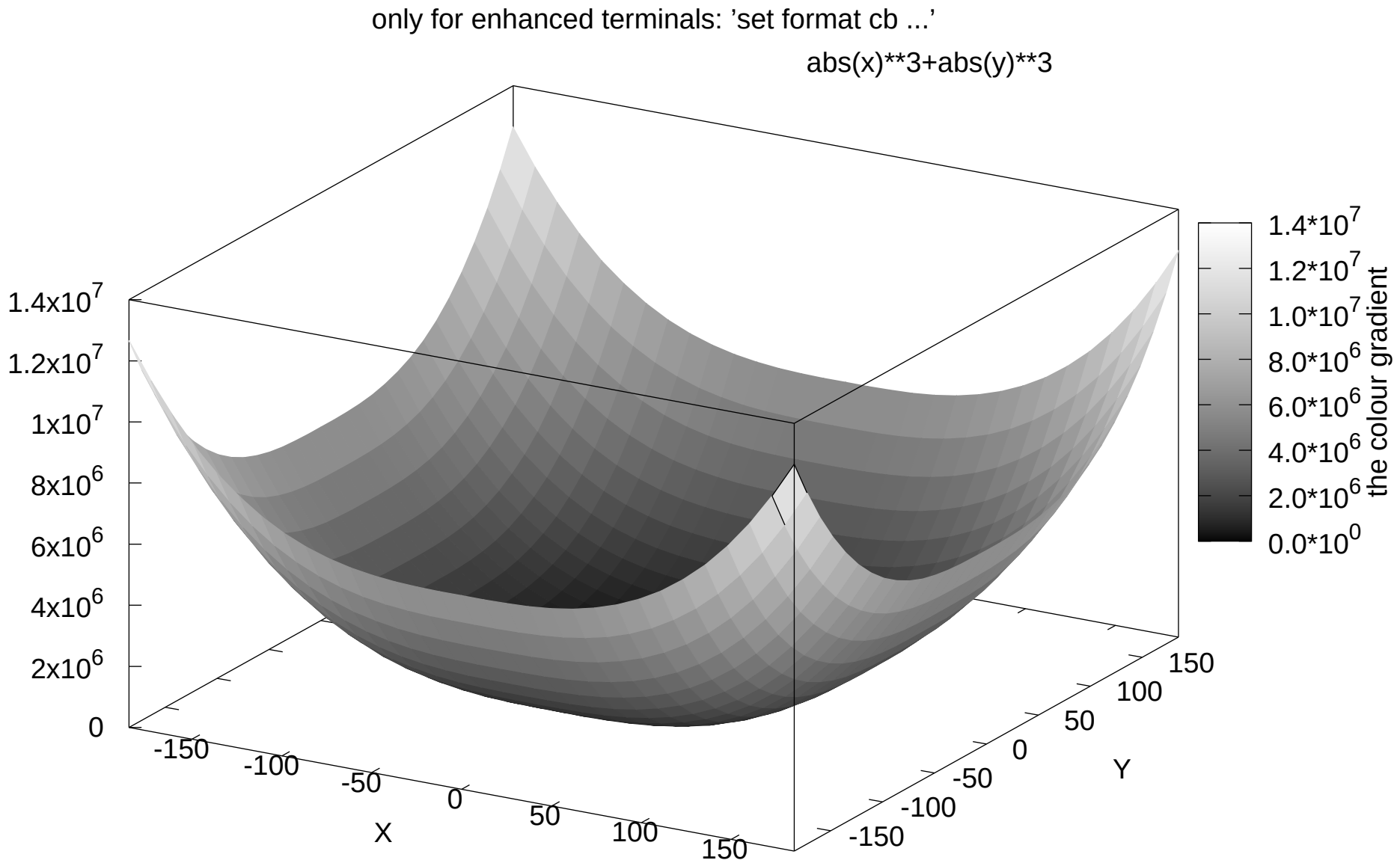

Data with different nb of points in scans; pm3d ftriangles flush center
'triangle.dat'
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

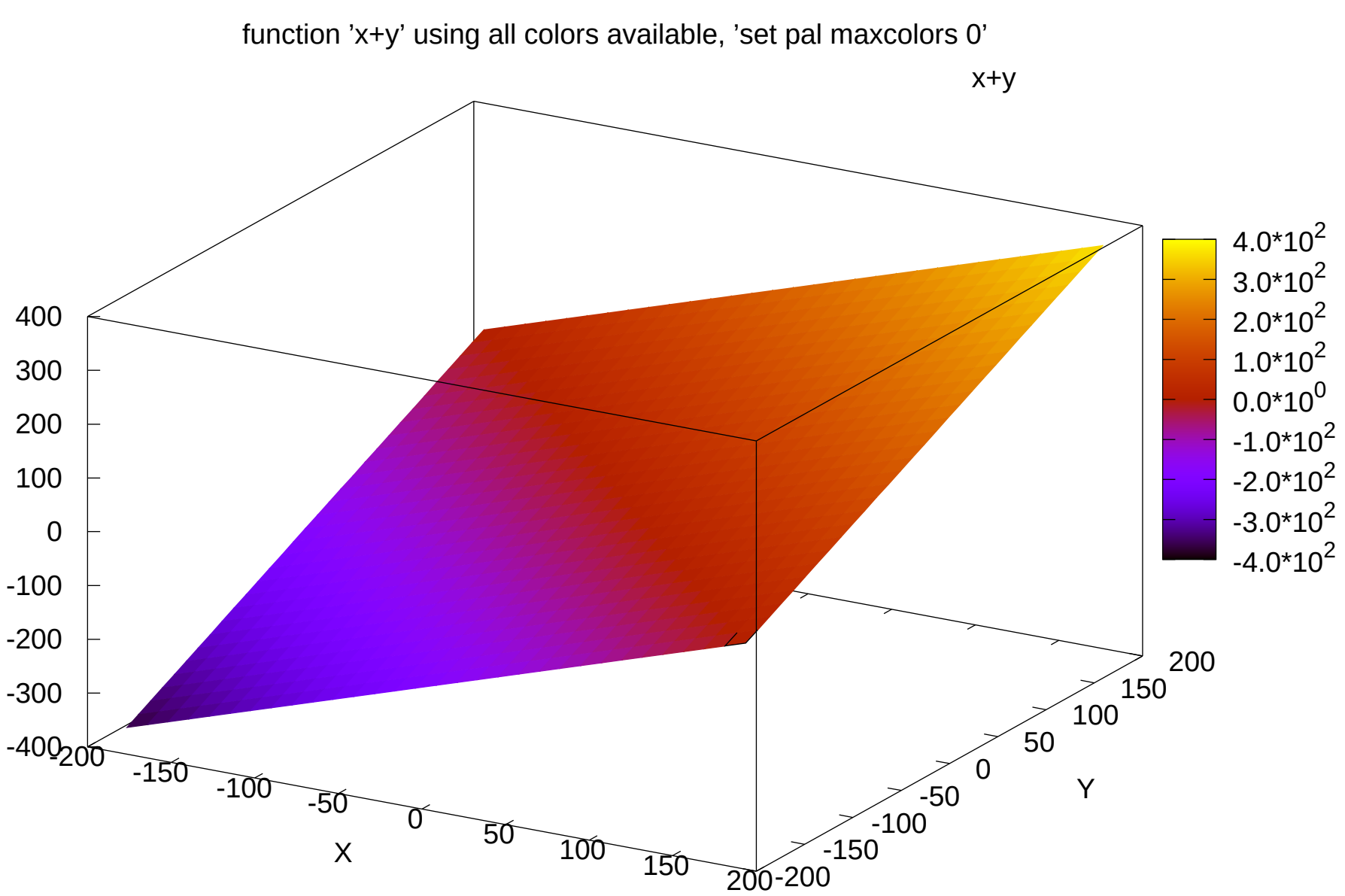

Data with different nb of points in scans; pm3d ftriangles flush end
'triangle.dat'
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

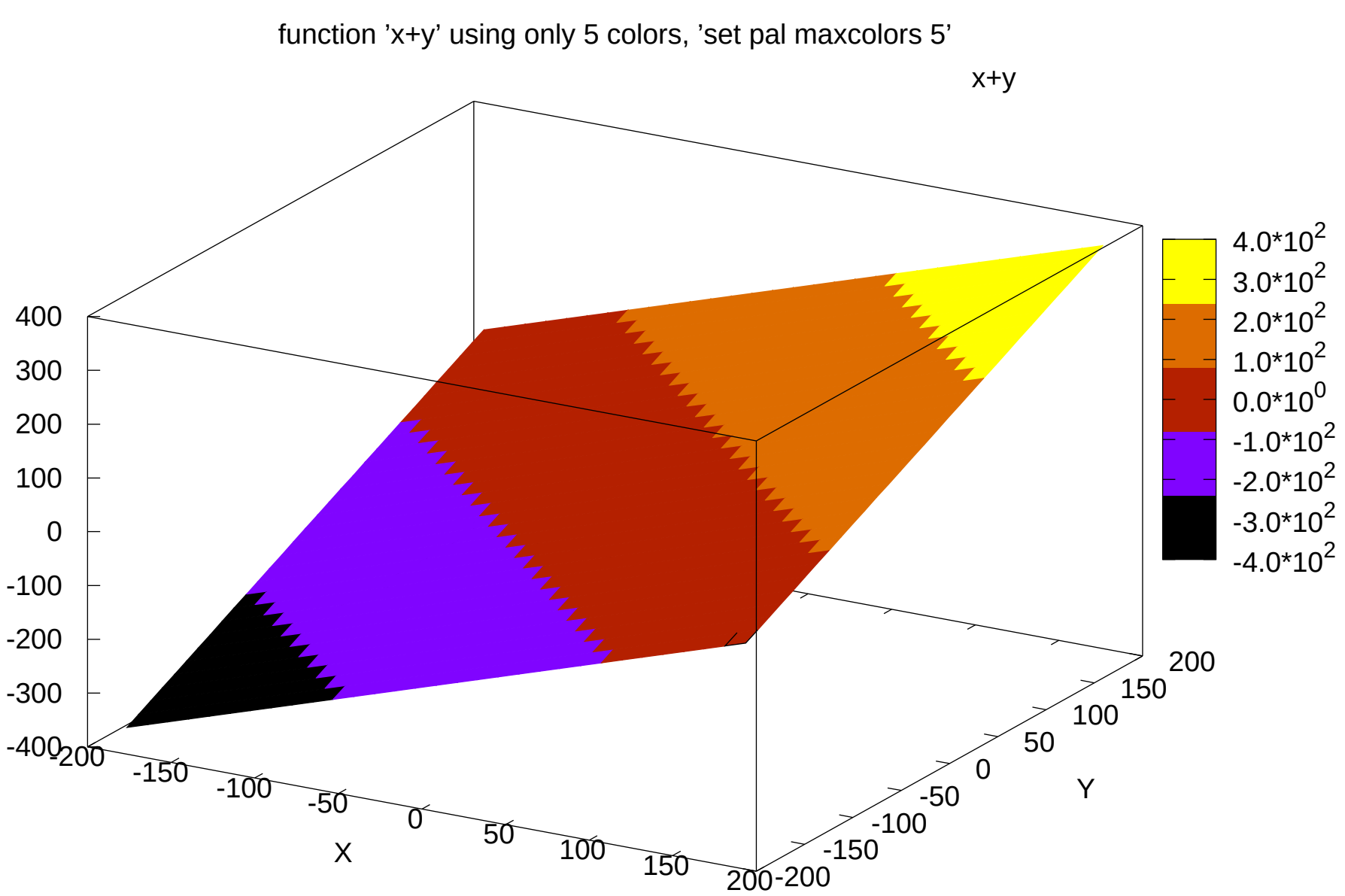

Using interpolation with datafile; pm3d map interpolate 2,1
'triangle.dat'
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

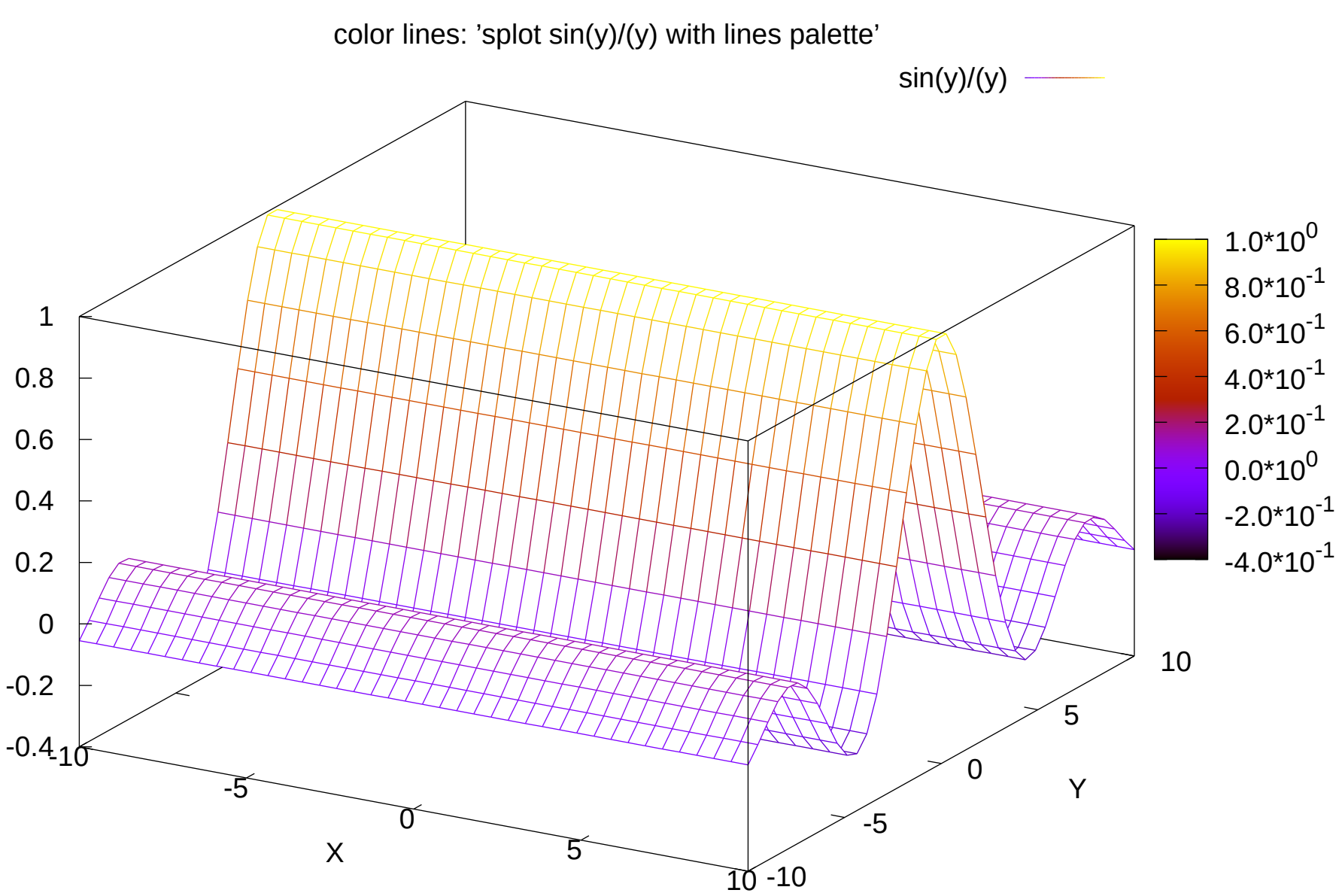

Using interpolation with datafile; pm3d map ftriangles interpolate 10,1
'triangle.dat'
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

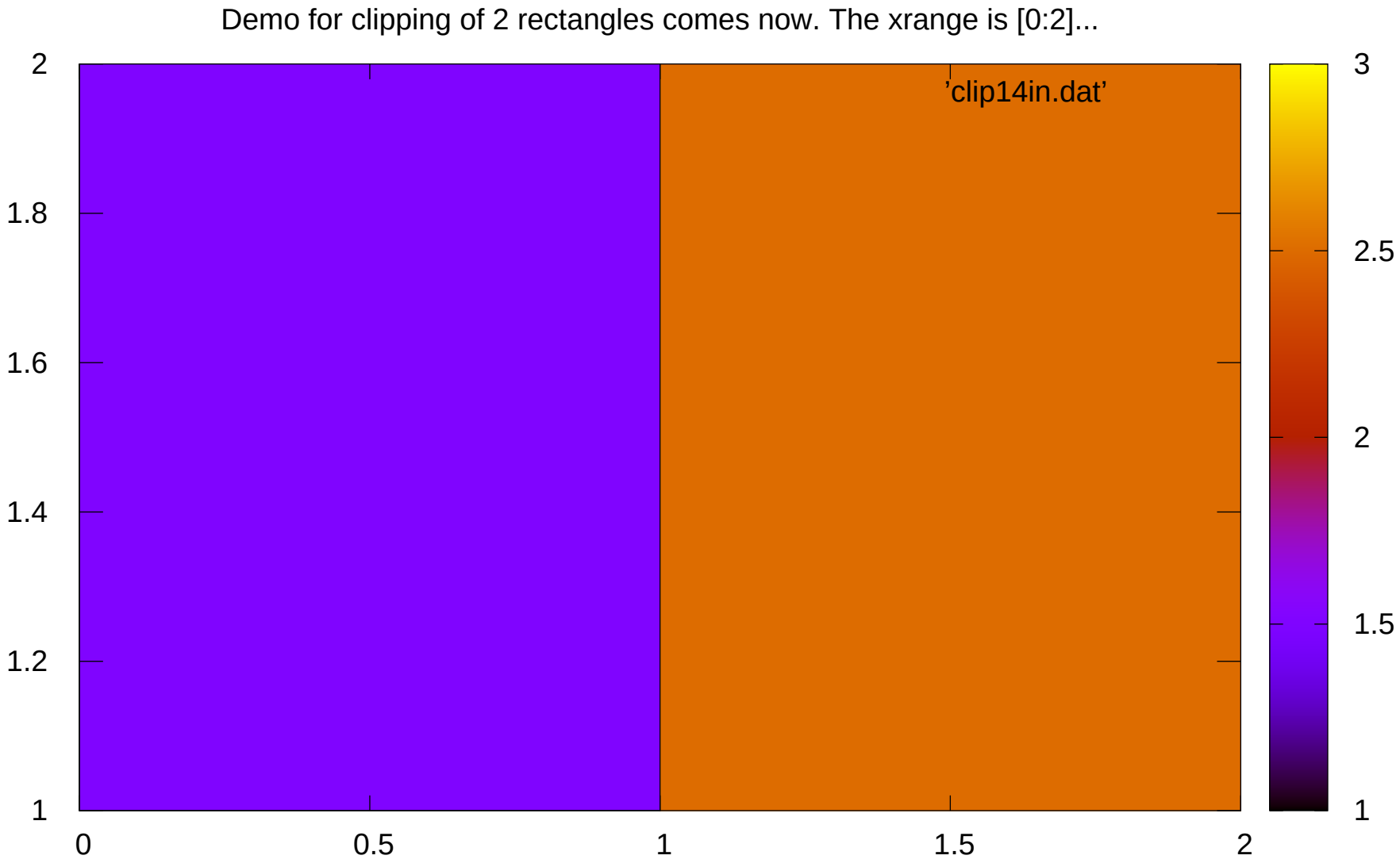

Using interpolation with datafile; pm3d at s ftriangles interpolate 10,1
'triangle.dat'
14
12
10
8
6
4
2
0
0
2
4
6
8
10
12
14
X LABEL
0
1
2
3
4
5
Y LABEL
6
14
12
10
8
6
4
2
0

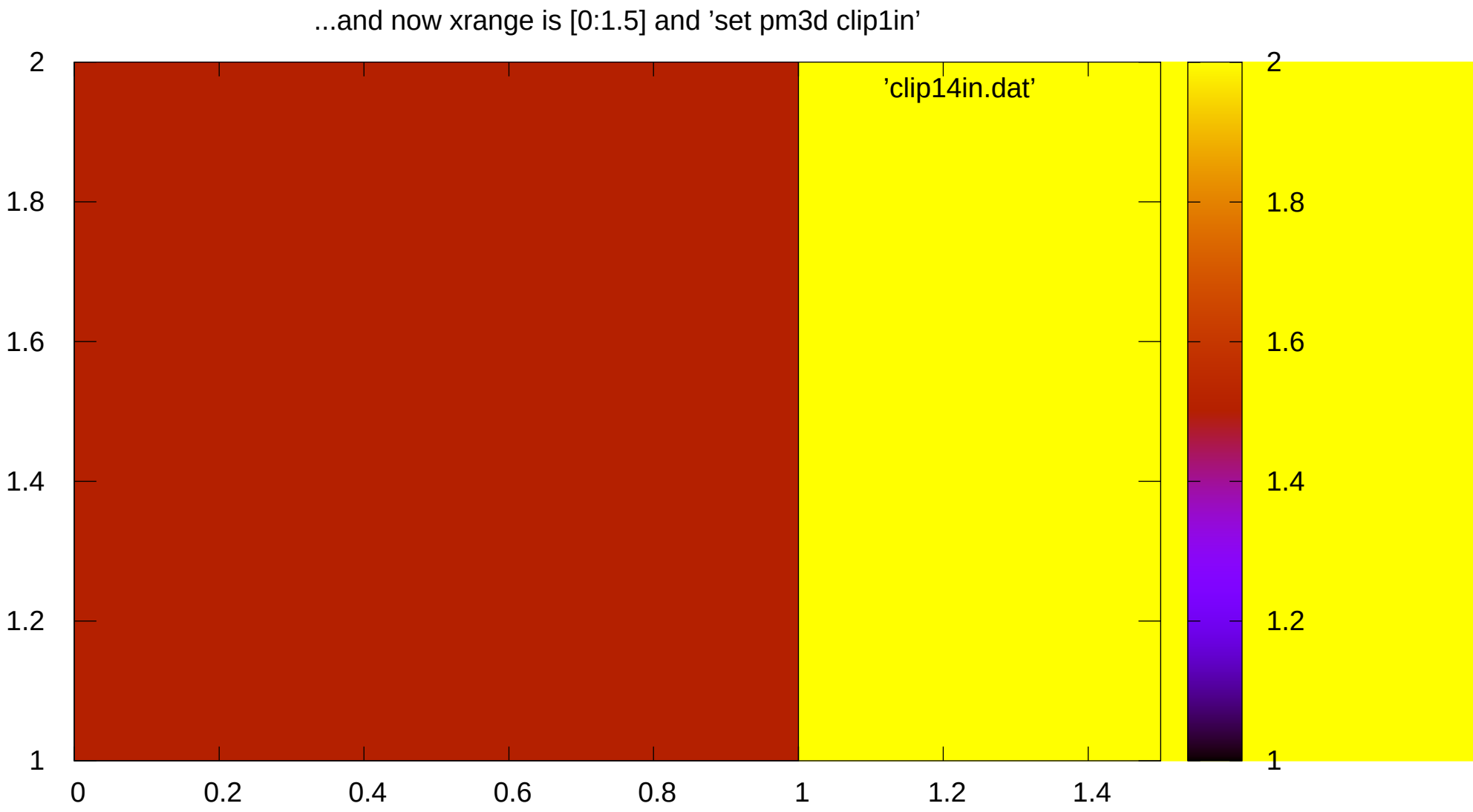

only for enhanced terminals: 'set format cb ...'
abs(x)**3+abs(y)**3
1.4x10^7
1.2x10^7
1x10^7
8x10^6
6x10^6
4x10^6
2x10^6
0
-150
-100
-50
0
50
100
150
X
-150
-100
-50
0
50
100
150
Y
1.4*10^7
1.2*10^7
1.0*10^7
8.0*10^6
6.0*10^6
4.0*10^6
2.0*10^6
0.0*10^0
the colour gradient

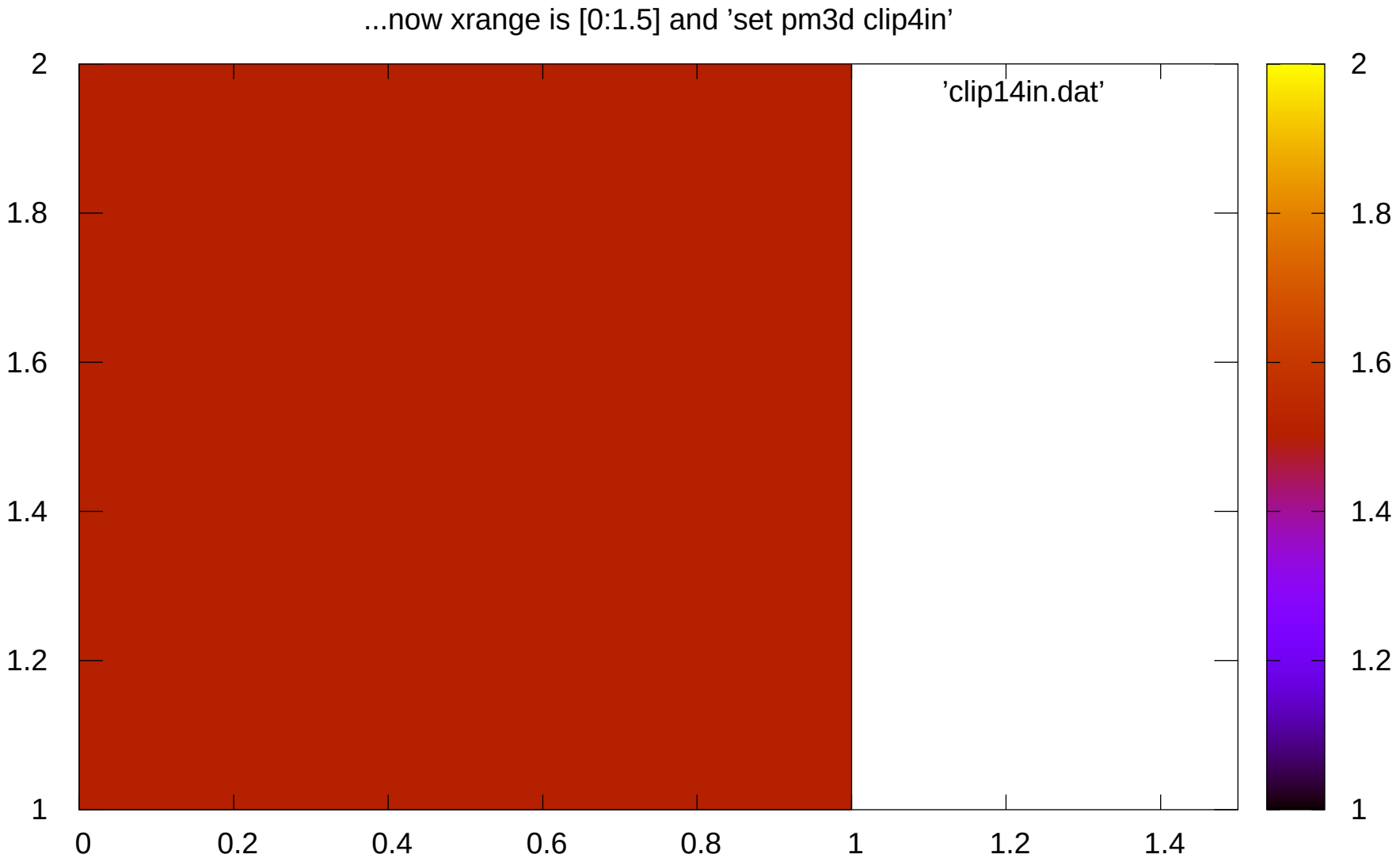

function ’x+y’ using all colors available, ’set pal maxcolors 0’
x+y
400
300
200
100
0
-100
-200
-300
-400
-200
-150
-100
-50
0
50
100
150
200
X
-200
-150
-100
-50
0
50
100
150
200
Y
4.0*10^2
3.0*10^2
2.0*10^2
1.0*10^2
0.0*10^0
-1.0*10^2
-2.0*10^2
-3.0*10^2
-4.0*10^2

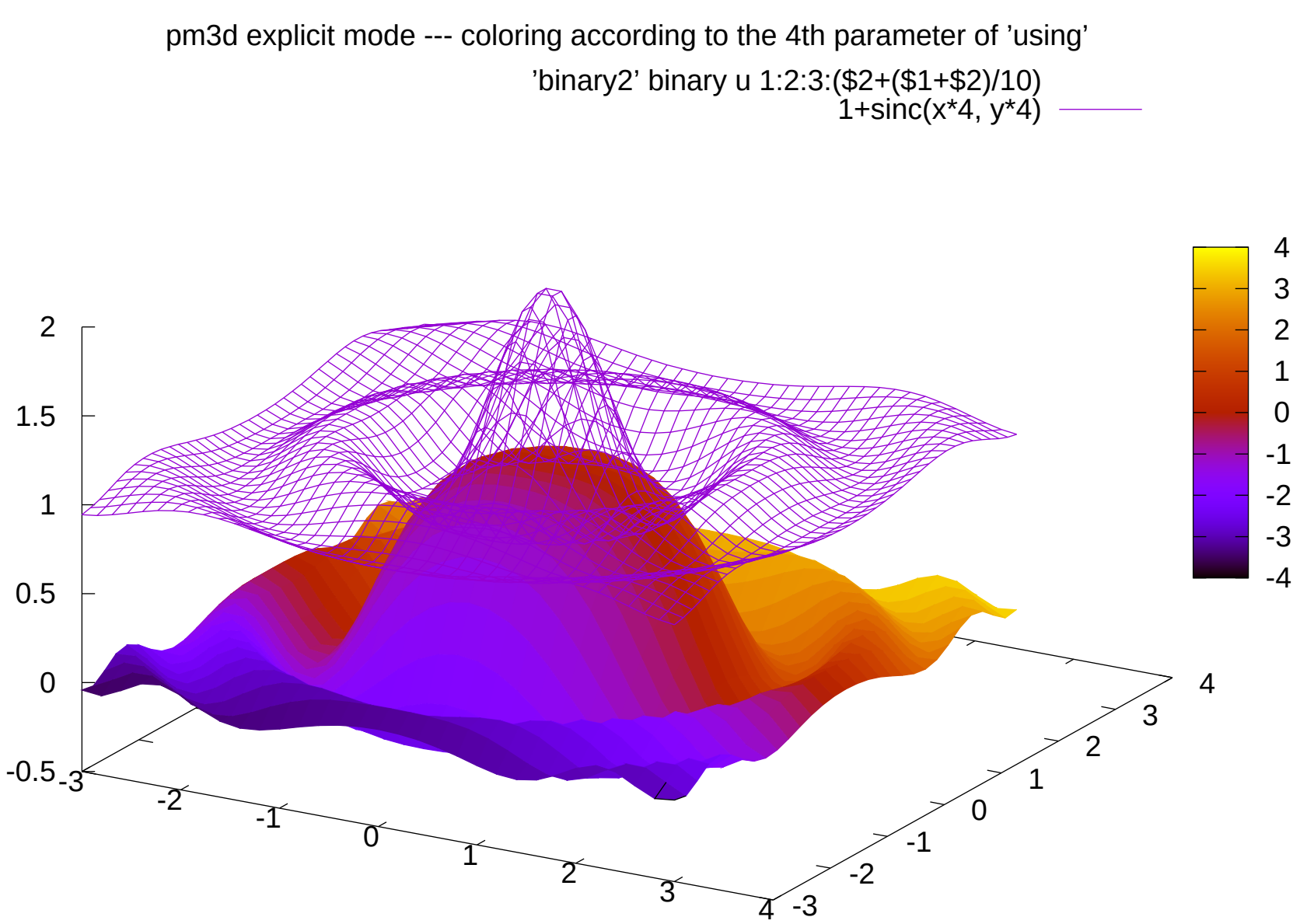

function ’x+y’ using only 5 colors, ’set pal maxcolors 5’
x+y
400
300
200
100
0
-100
-200
-300
-400
-200
-150
-100
-50
0
50
100
150
200
X
-200
-150
-100
-50
0
50
100
150
200
Y
4.0*10^2
3.0*10^2
2.0*10^2
1.0*10^2
0.0*10^0
-1.0*10^2
-2.0*10^2
-3.0*10^2
-4.0*10^2


color lines: ’splot sin(y)/(y) with lines palette’
sin(y)/(y)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10
X
-10
-5
0
5
10
Y
1.0*10^0
8.0*10^-1
6.0*10^-1
4.0*10^-1
2.0*10^-1
0.0*10^0
-2.0*10^-1
-4.0*10^-1

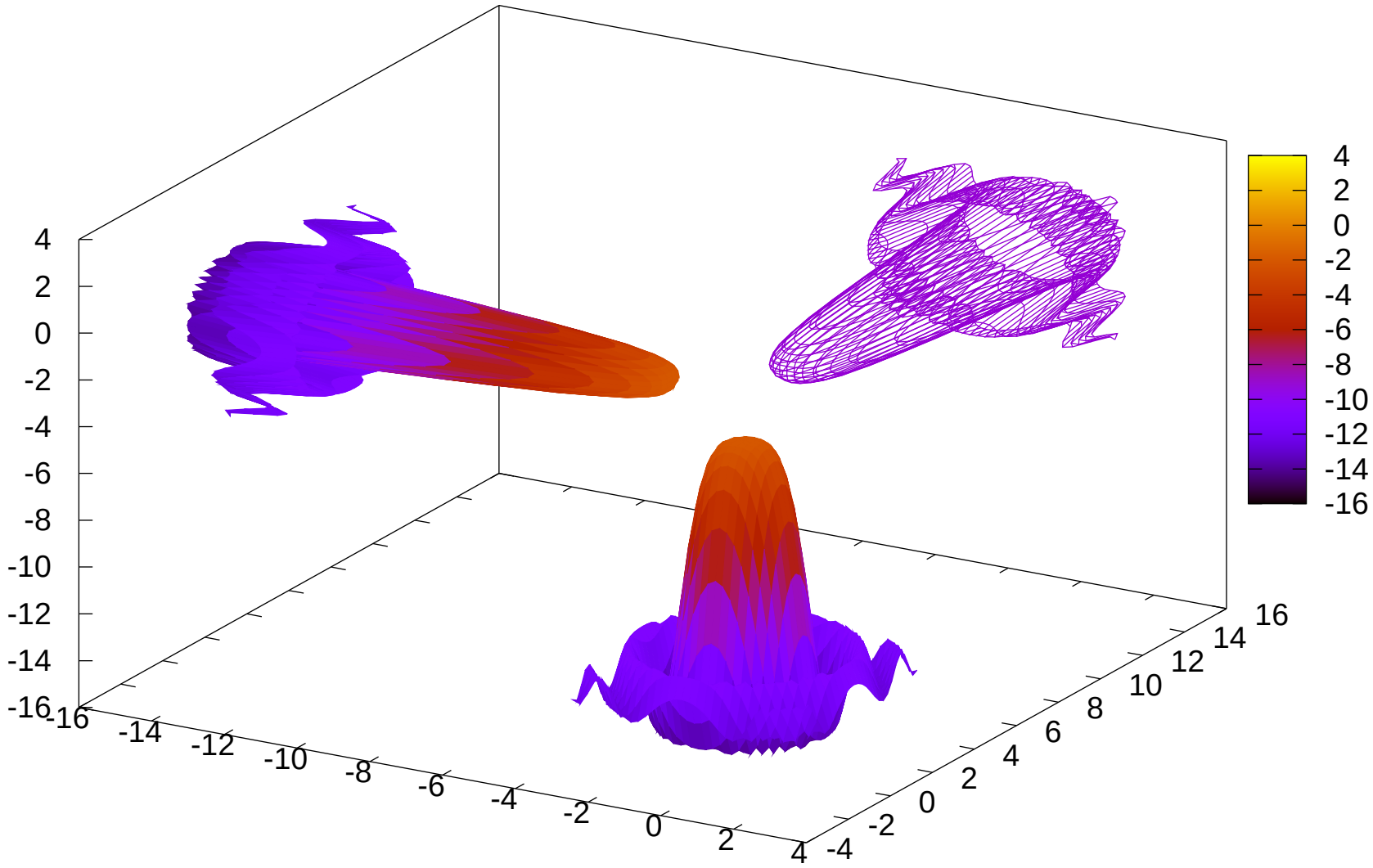

Demo for clipping of 2 rectangles comes now. The xrange is [0:2]...
'clip14in.dat'
2
1.8
1.6
1.4
1.2
1
0
0.5
1
1.5
2
3
2.5
2
1.5
1

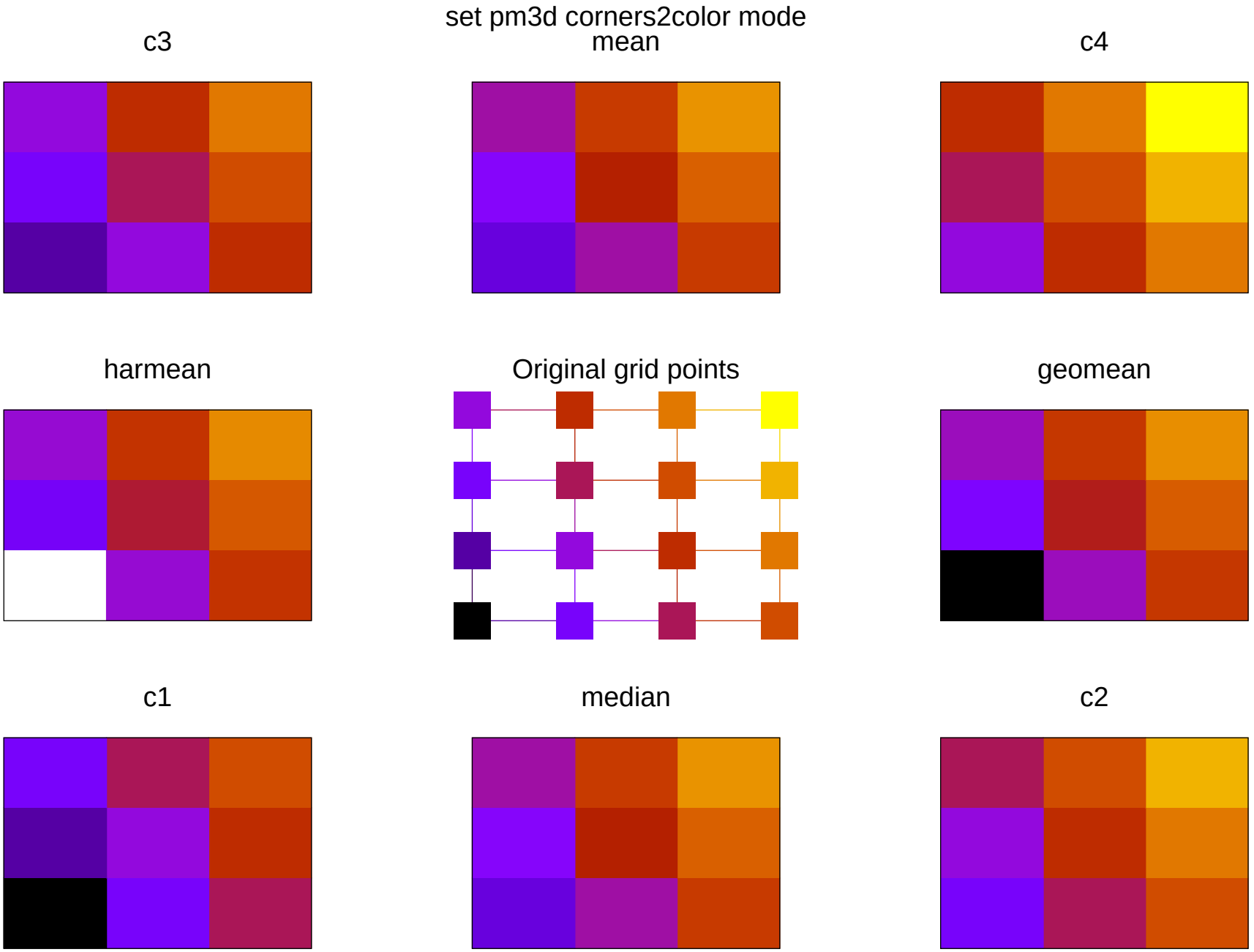

...and now xrange is [0:1.5] and ’set pm3d clip1in’
’clip14in.dat’
2
1.8
1.6
1.4
1.2
1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
2
1.8
1.6
1.4
1.2
1


...now xrange is [0:1.5] and ’set pm3d clip4in’
’clip14in.dat’
2
1.8
1.6
1.4
1.2
1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
2
1.8
1.6
1.4
1.2
1

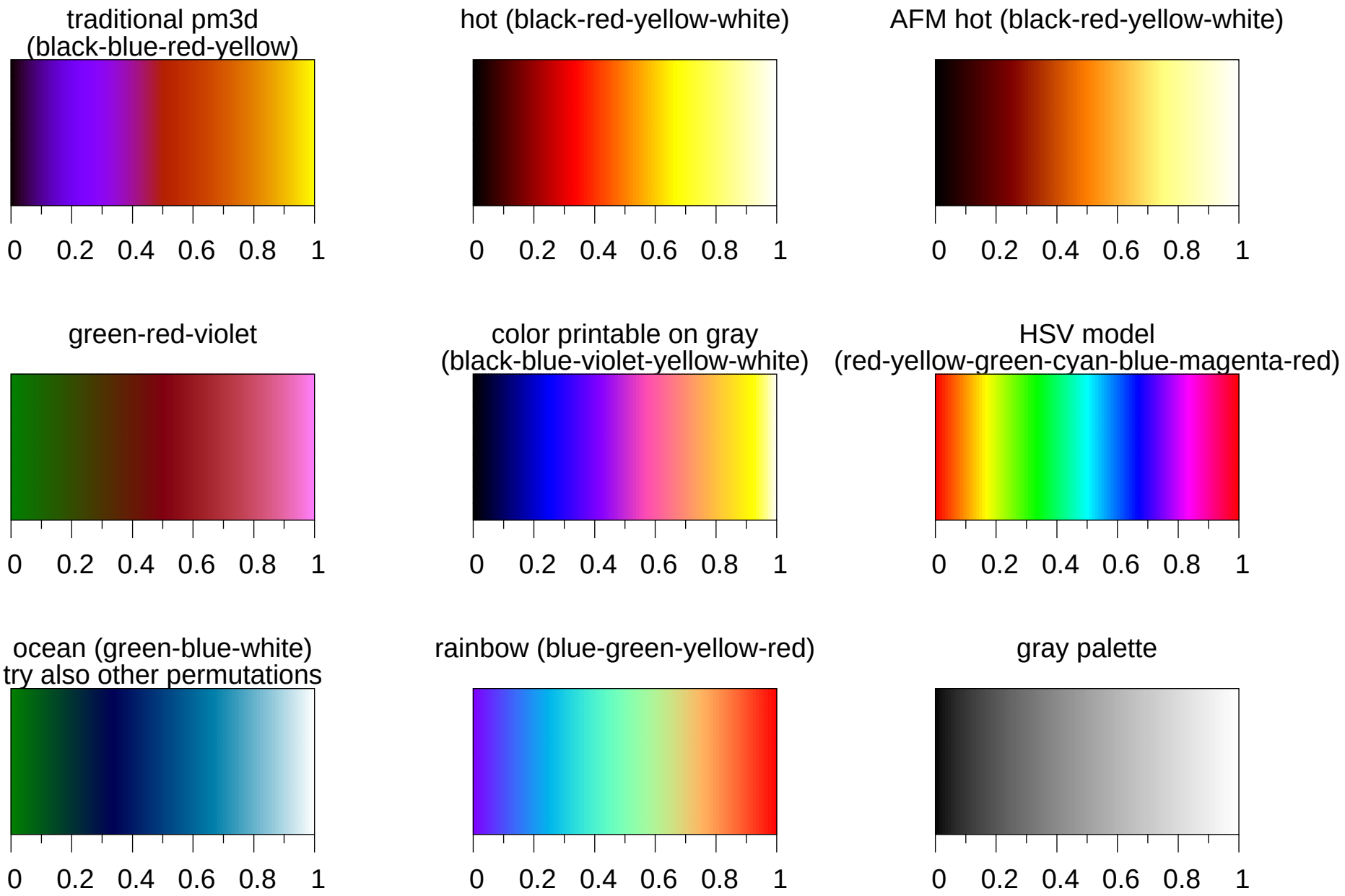

pm3d explicit mode --- coloring according to the 4th parameter of 'using'
'binary2' binary u 1:2:3:($2+($1+$2)/10)
1+sinc(x*4, y*4)
2
1.5
1
0.5
0
-0.5
-3
-2
-1
0
1
2
3
4
-3
-2
-1
0
1
2
3
4
4
3
2
1
0
-1
-2
-3
-4

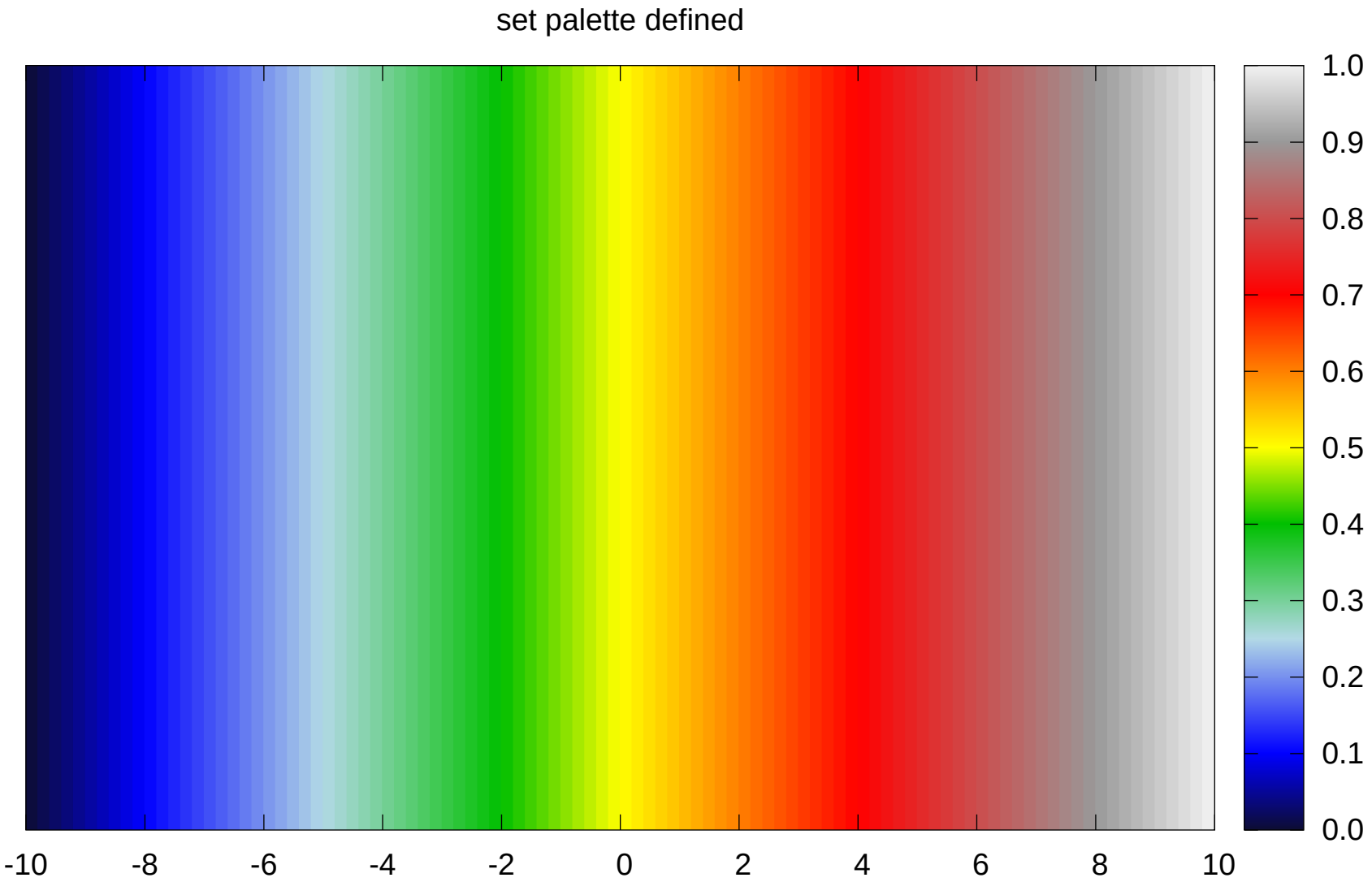
coloring according to the 3rd ’using’ parameter (left) and to the z-value (bottom)

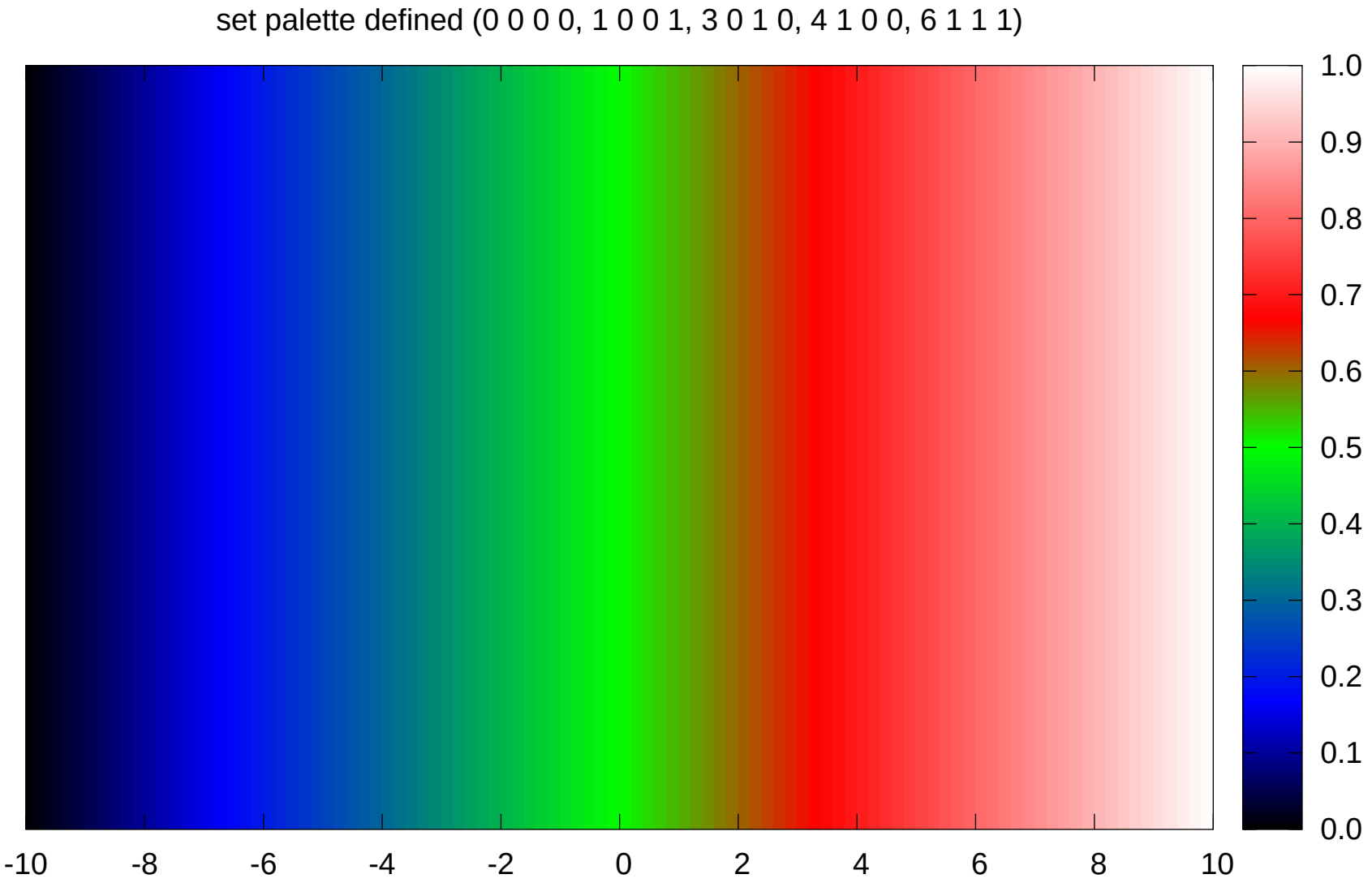


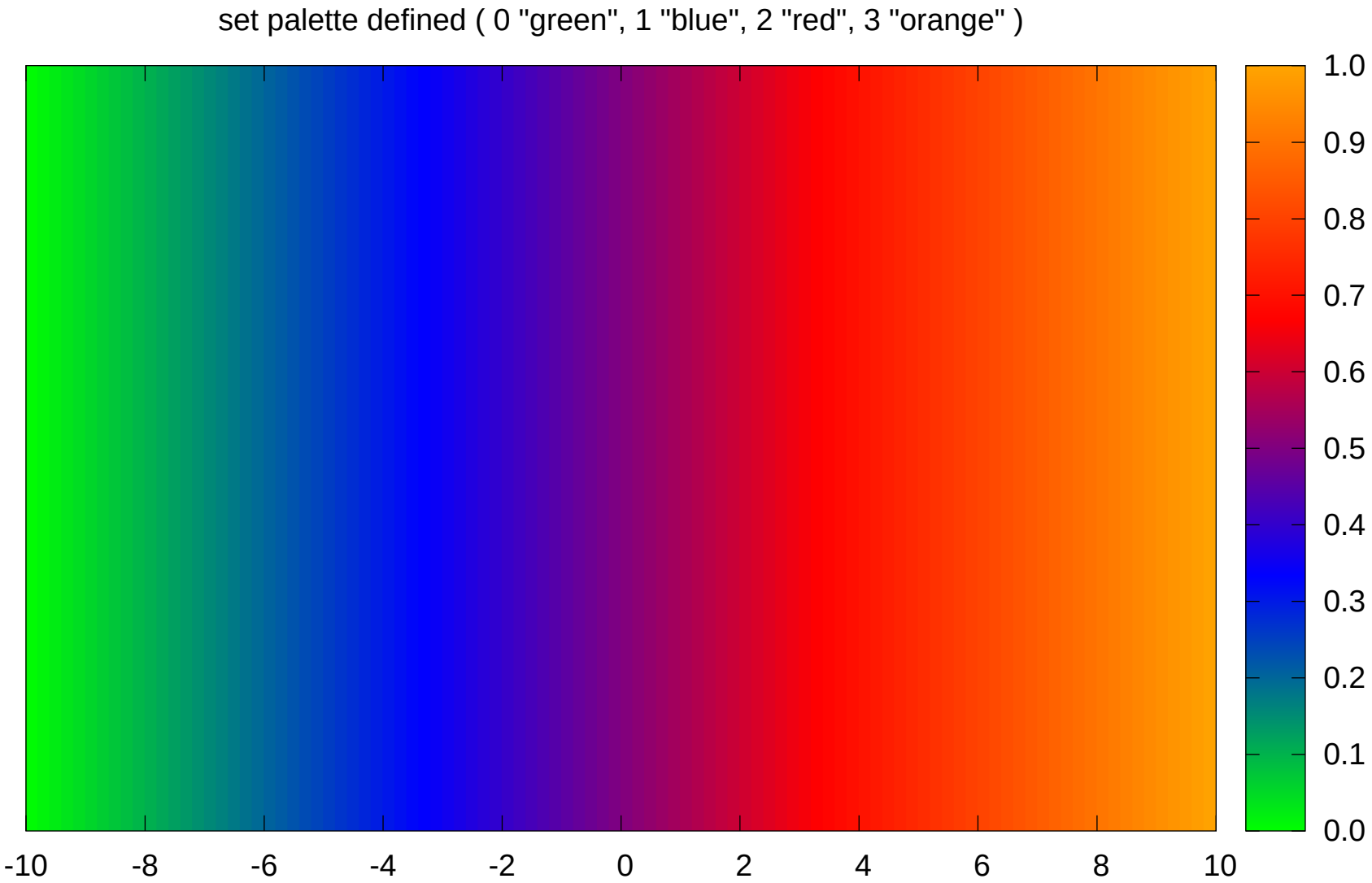

set pm3d corners2color mode
mean
c3
c4
harmean
Original grid points
geomean
c1
median
c2

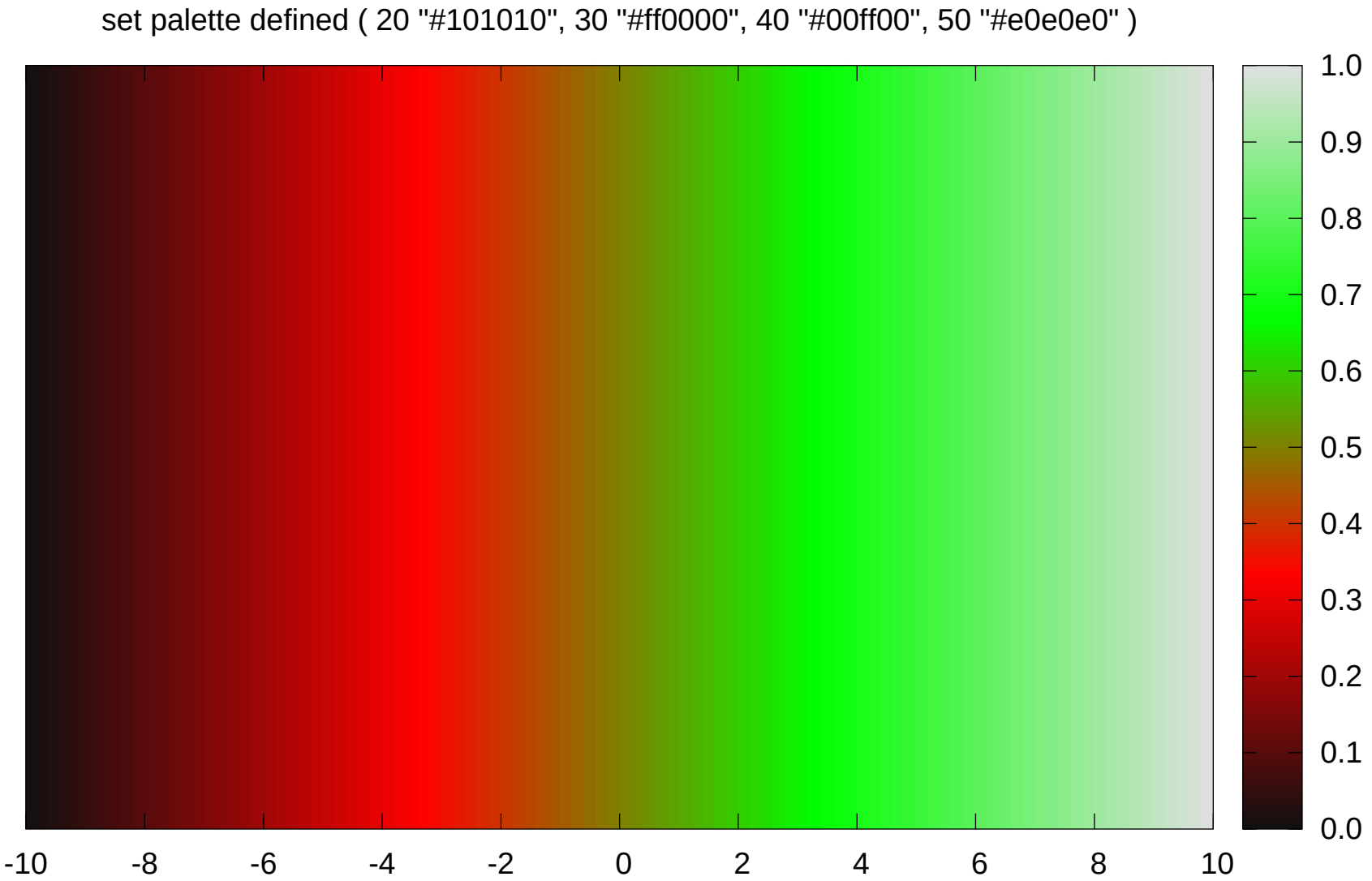
Palettes according to 'help palette rgbformulae'

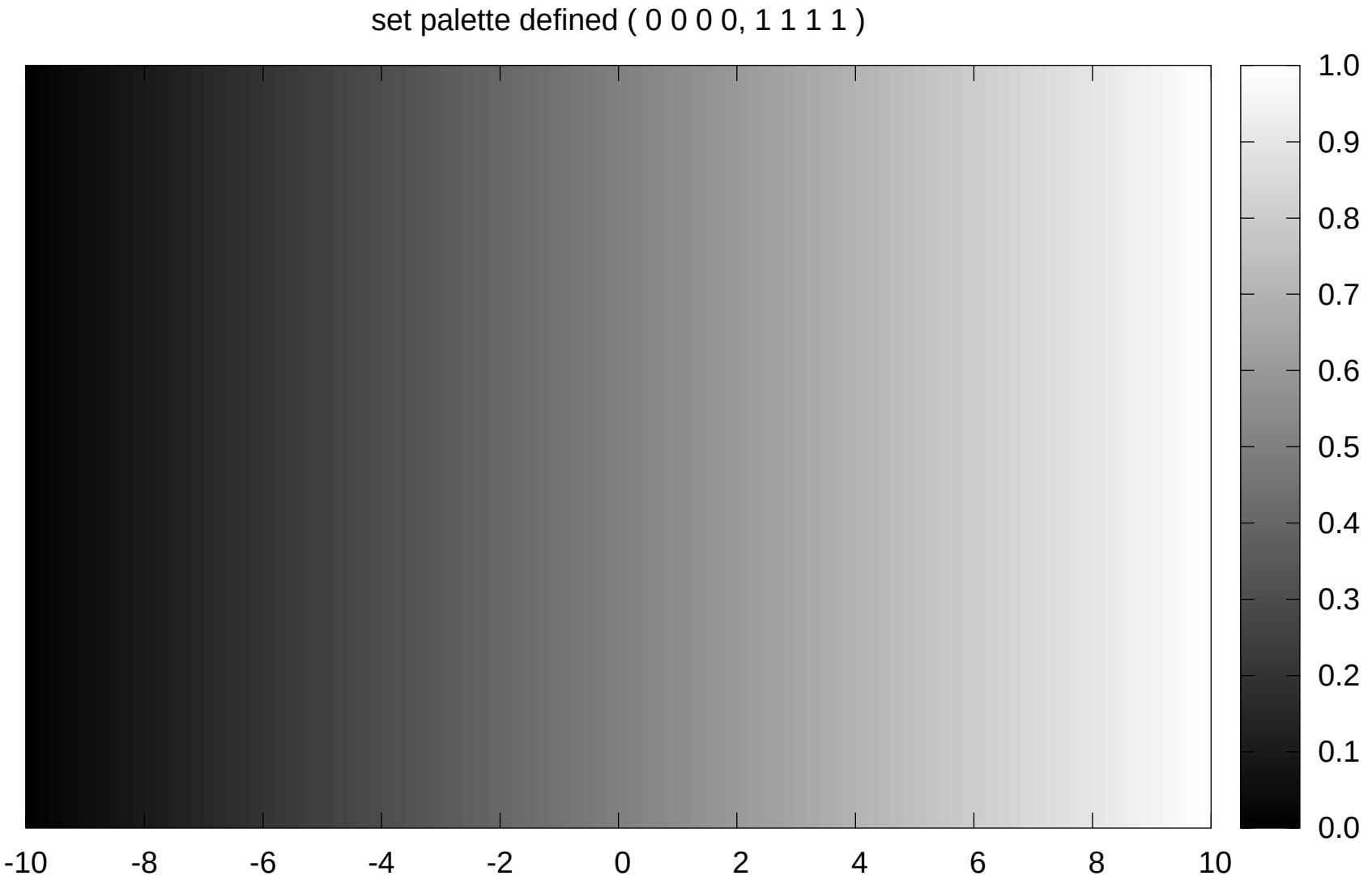


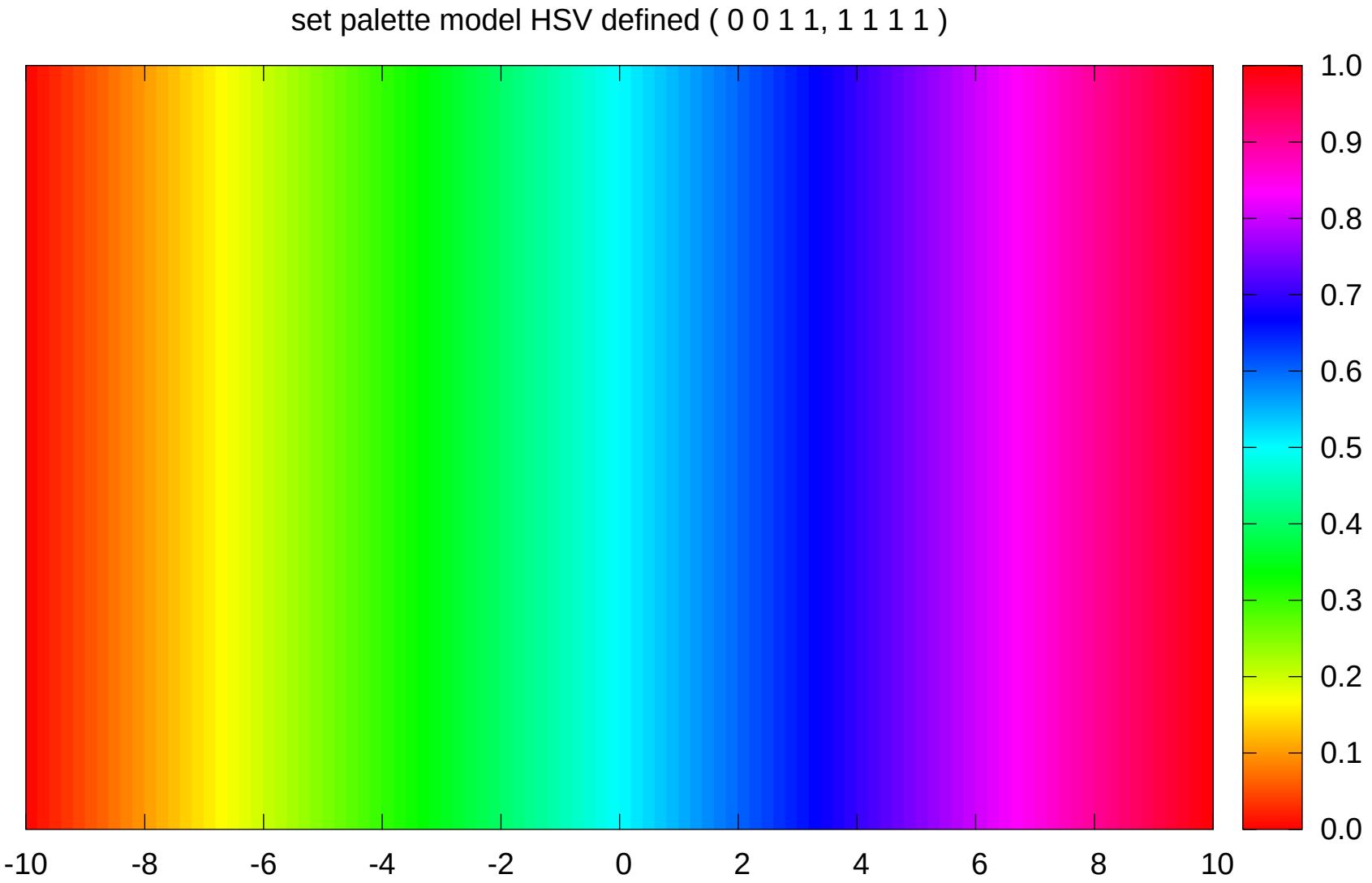

set palette defined
-10
-8
-6
-4
-2
0
2
4
6
8
10
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

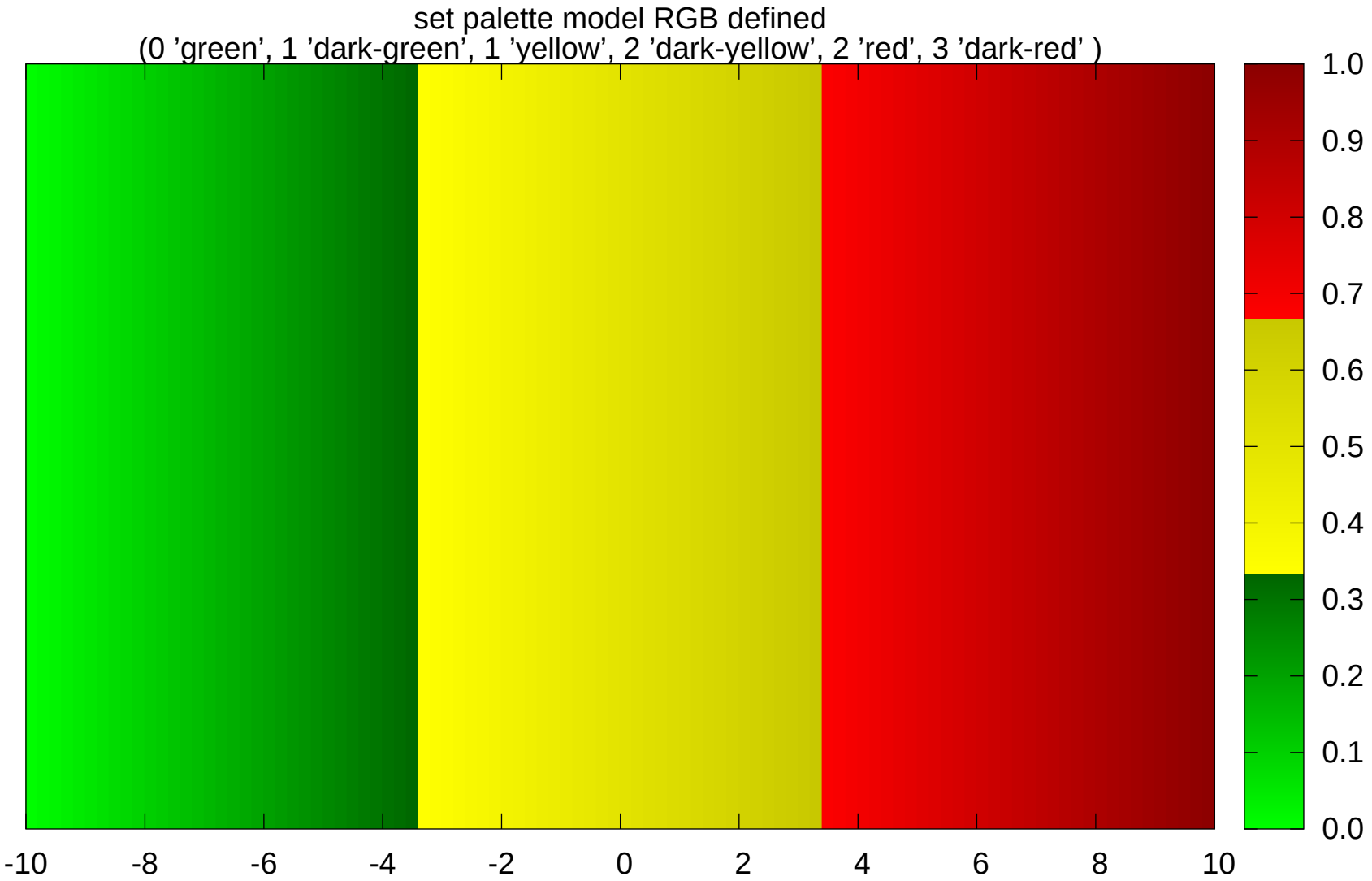

set palette defined (0 0 0 0, 1 0 0 1, 3 0 1 0, 4 1 0 0, 6 1 1 1)
-10
-8
-6
-4
-2
0
2
4
6
8
10
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

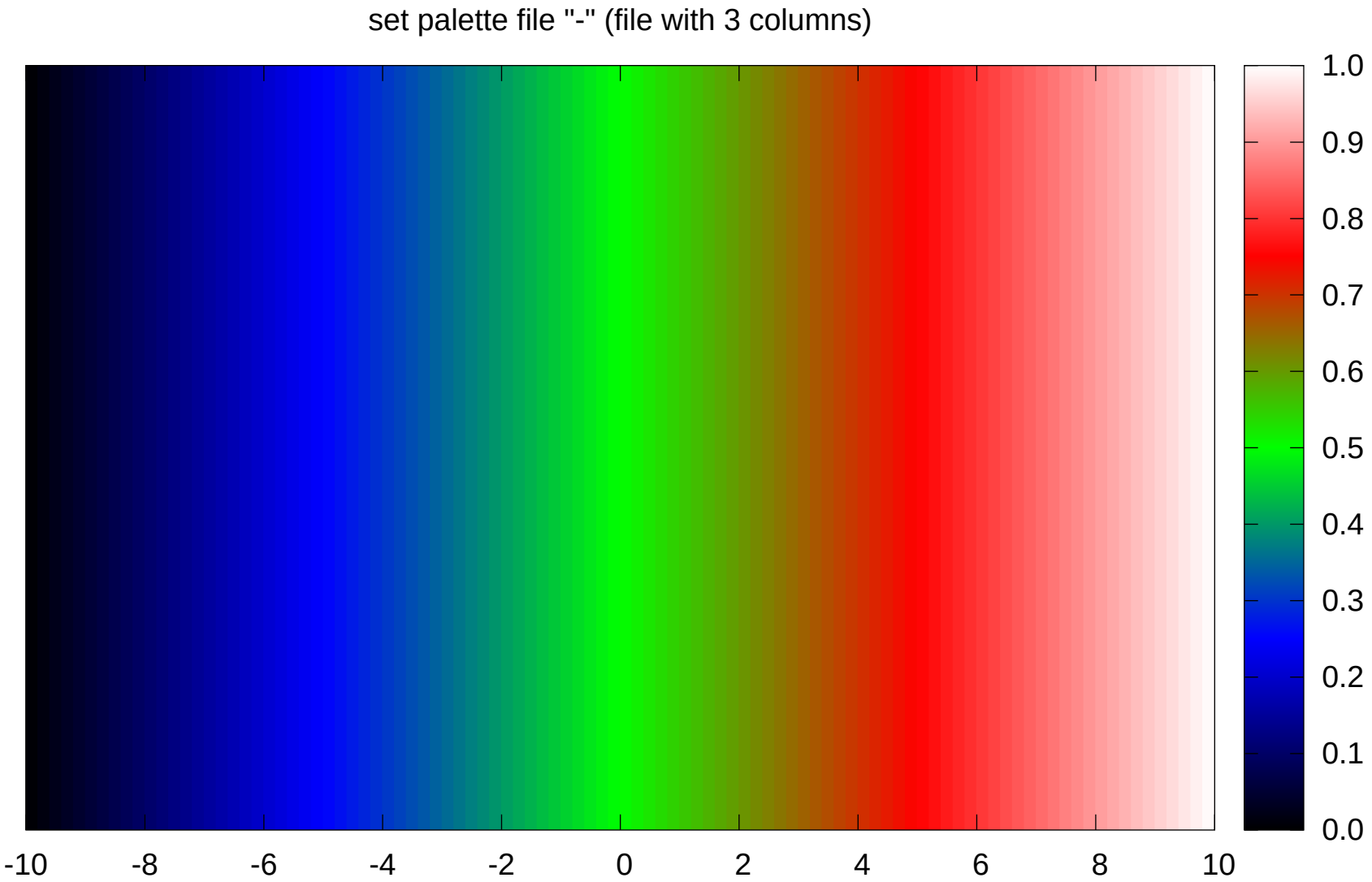

set palette defined (0 "green", 1 "blue", 2 "red", 3 "orange")
-10
-8
-6
-4
-2
0
2
4
6
8
10
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

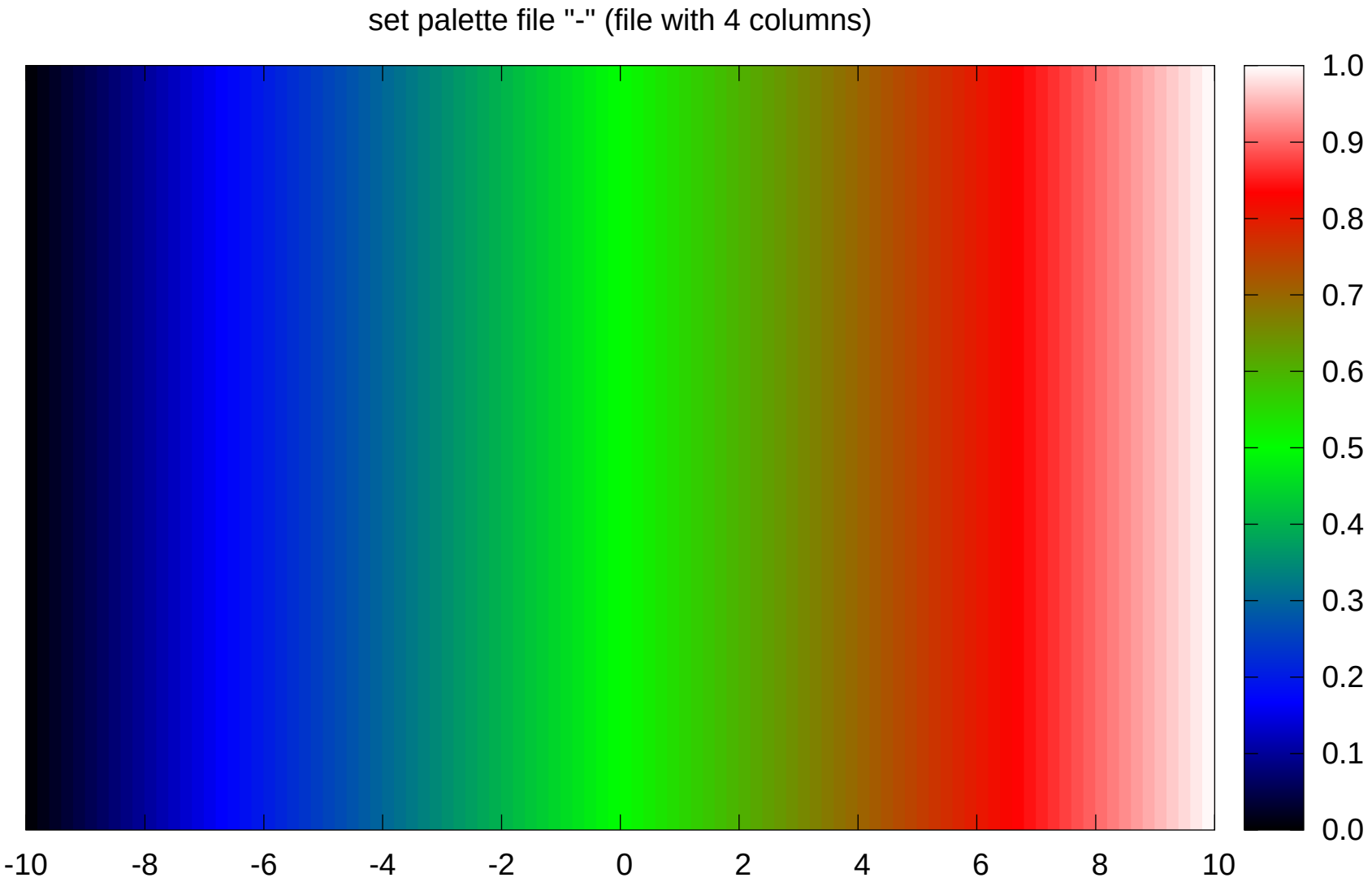

set palette defined (20 "#101010", 30 "#ff0000", 40 "#00ff00", 50 "#e0e0e0")
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-10
-8
-6
-4
-2
0
2
4
6
8
10

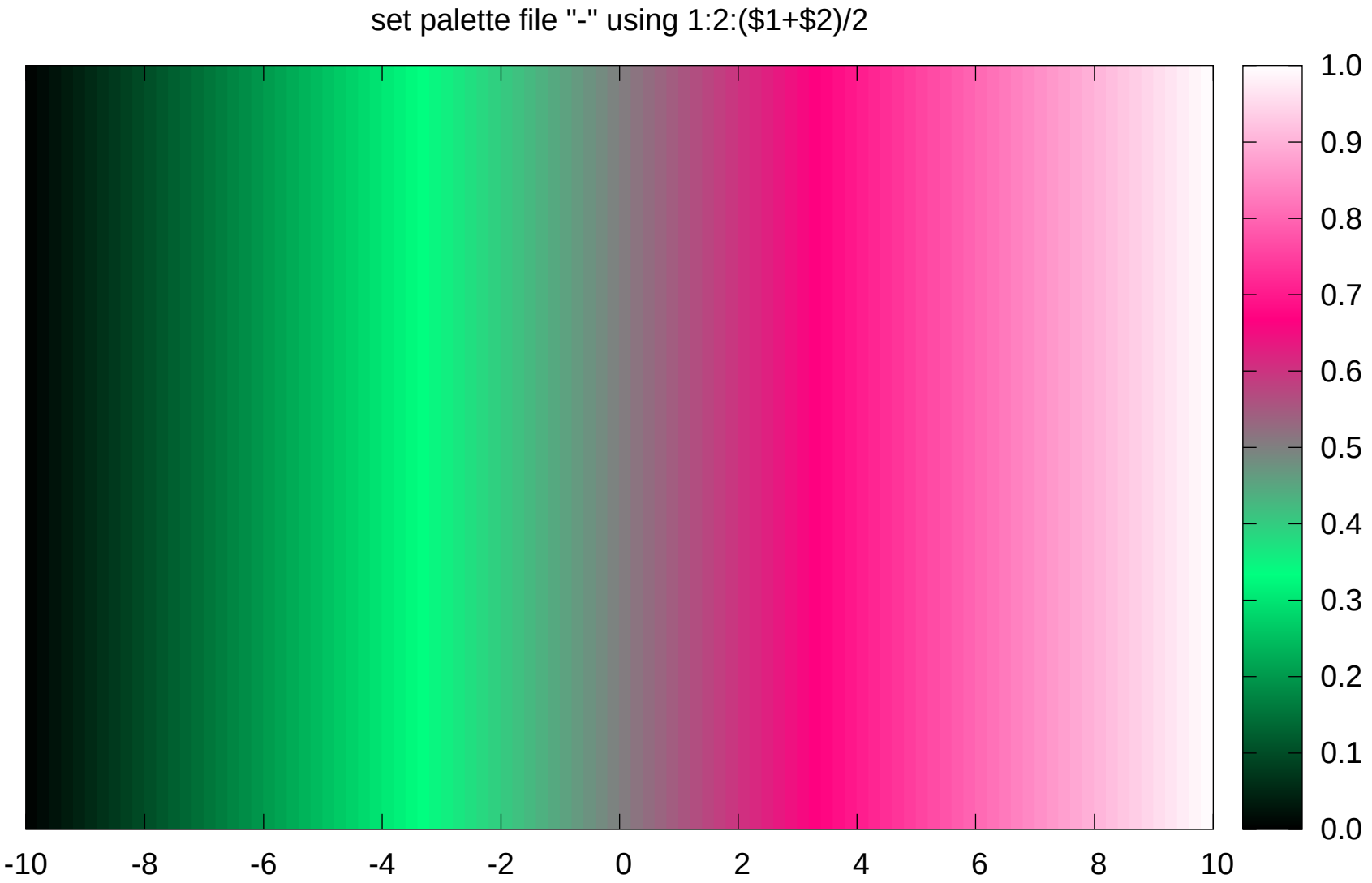

set palette defined (0 0 0 0, 1 1 1 1)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-10
-8
-6
-4
-2
0
2
4
6
8
10

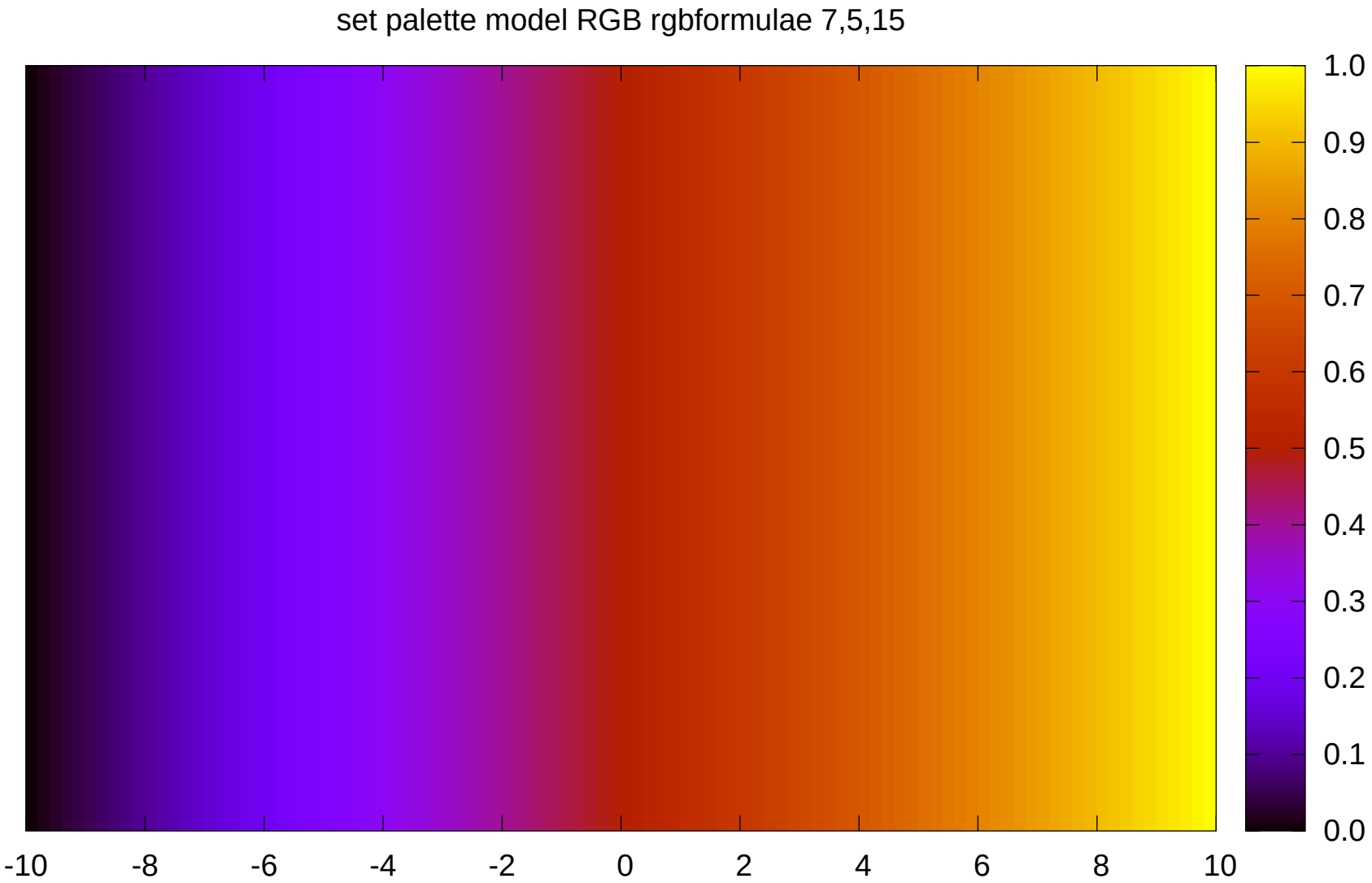

set palette model HSV defined (0 0 1 1, 1 1 1 1)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-10
-8
-6
-4
-2
0
2
4
6
8
10

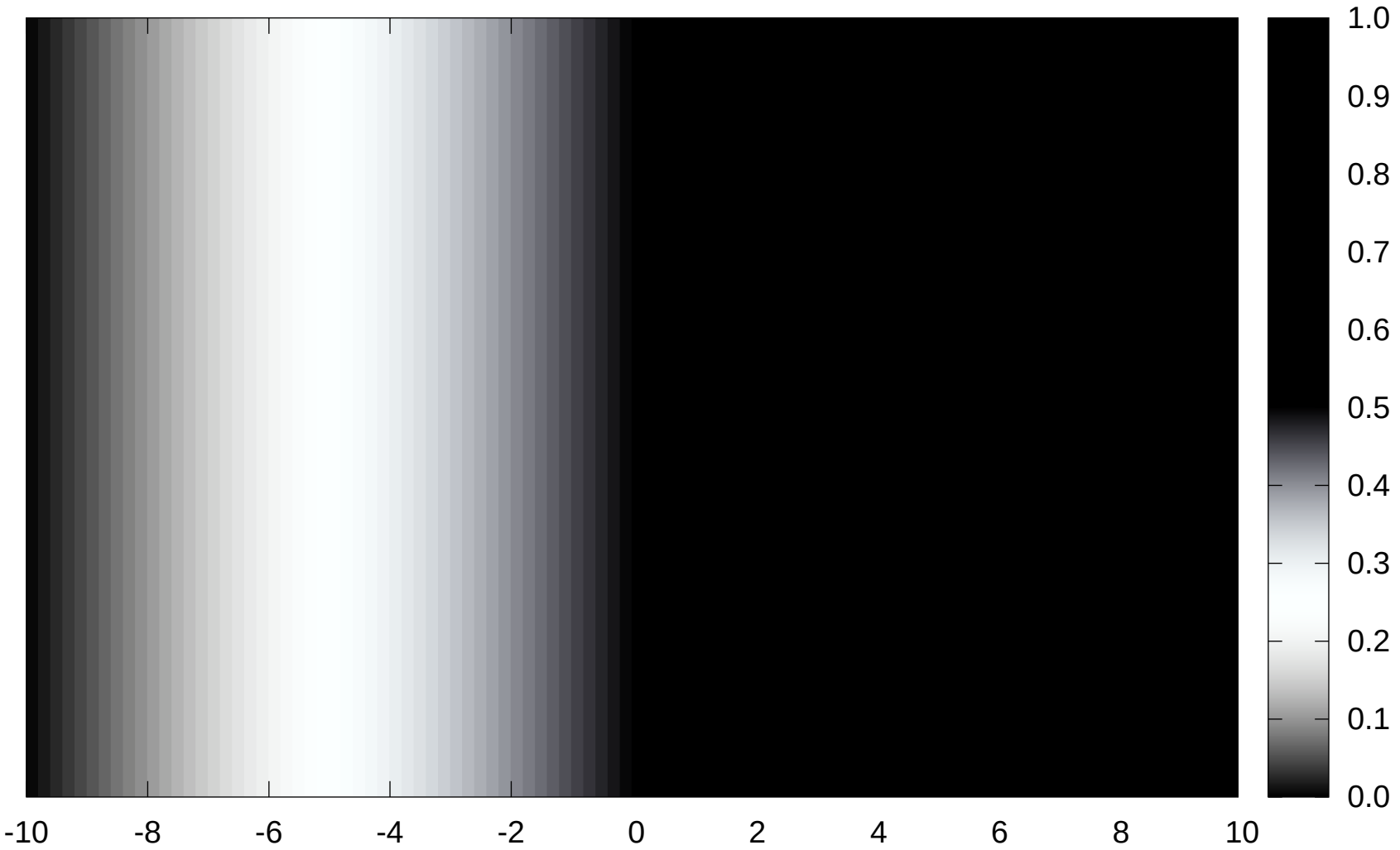

set palette model RGB defined
(0 ’green’, 1 ’dark-green’, 1 ’yellow’, 2 ’dark-yellow’, 2 ’red’, 3 ’dark-red’)
-10
-8
-6
-4
-2
0
2
4
6
8
10
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0


set palette file "-" (file with 3 columns)
-10
-8
-6
-4
-2
0
2
4
6
8
10
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

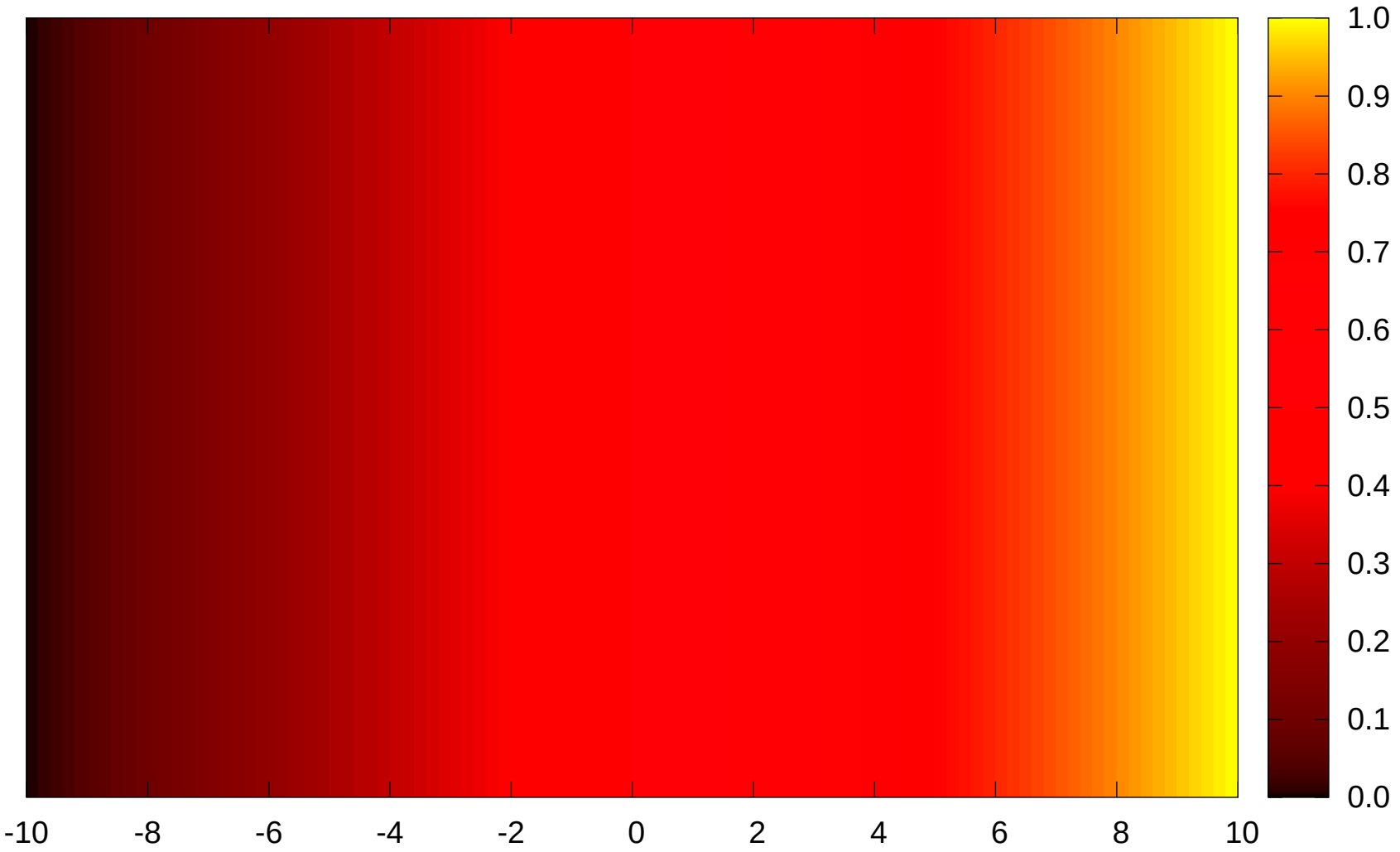

set palette file "-" (file with 4 columns)
-10
-8
-6
-4
-2
0
2
4
6
8
10
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

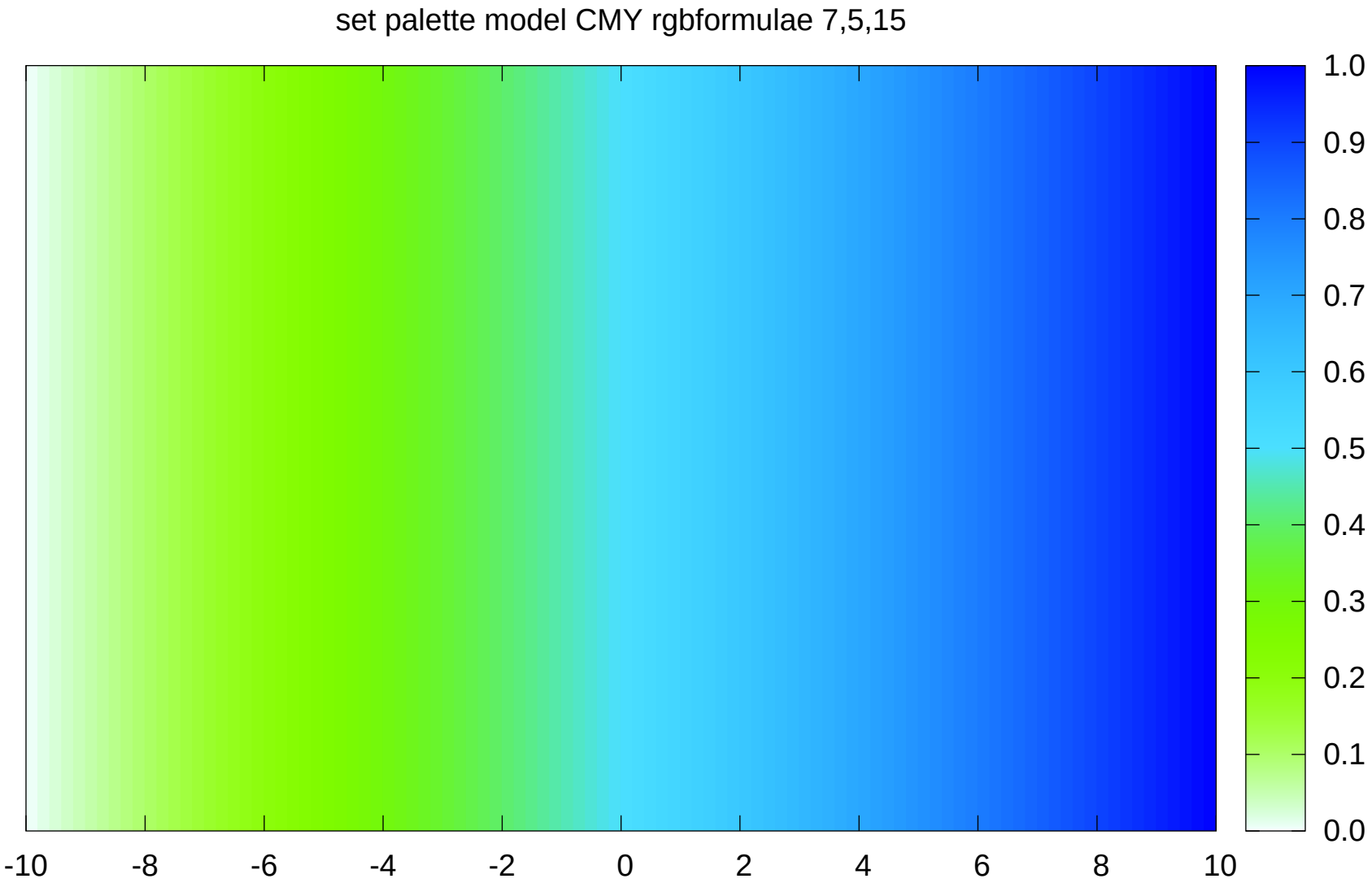

set palette file "-" using 1:2:($1+$2)/2
-10
-8
-6
-4
-2
0
2
4
6
8
10
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

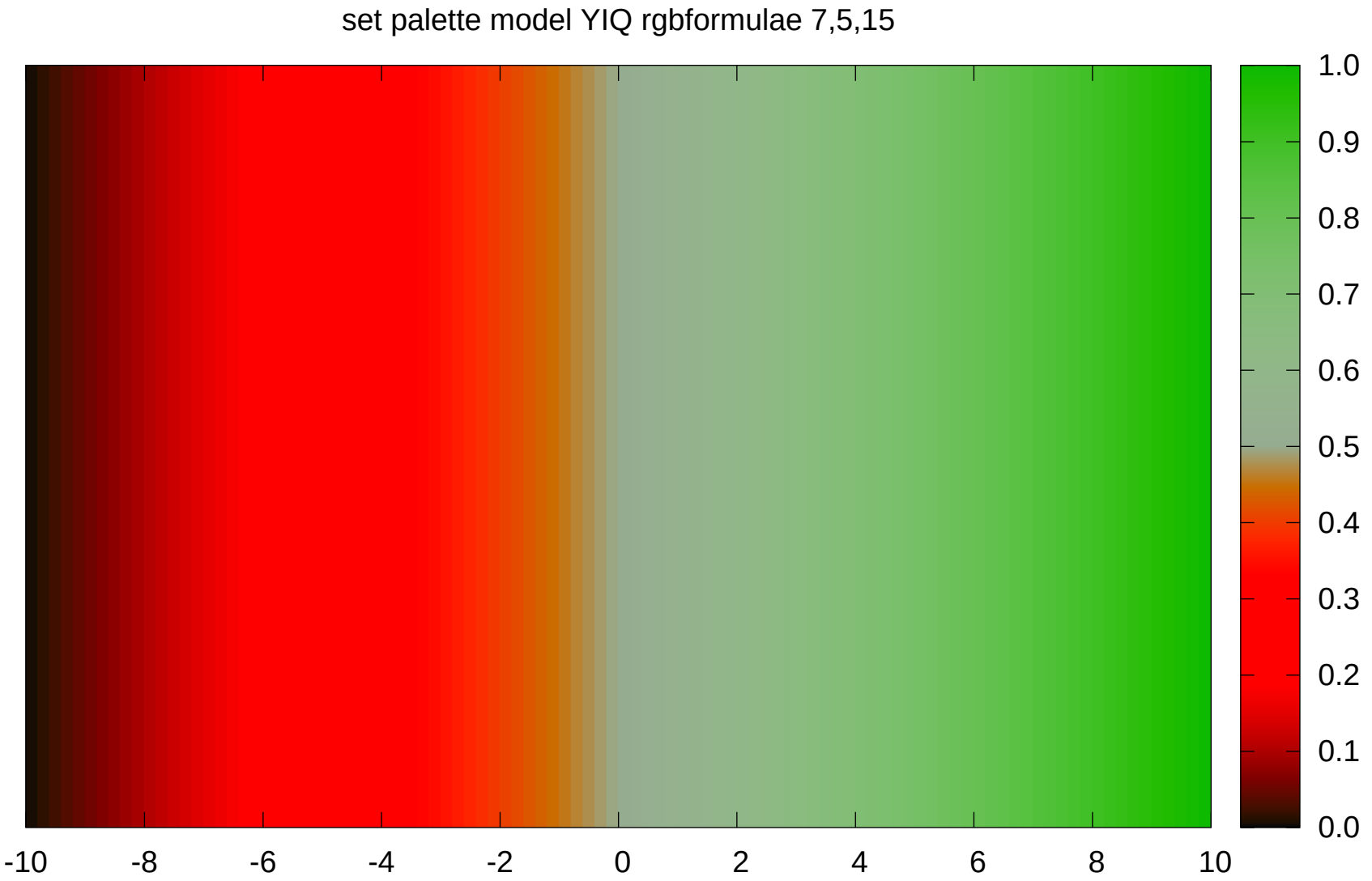

set palette model RGB rgbformulae 7,5,15
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-10
-8
-6
-4
-2
0
2
4
6
8
10

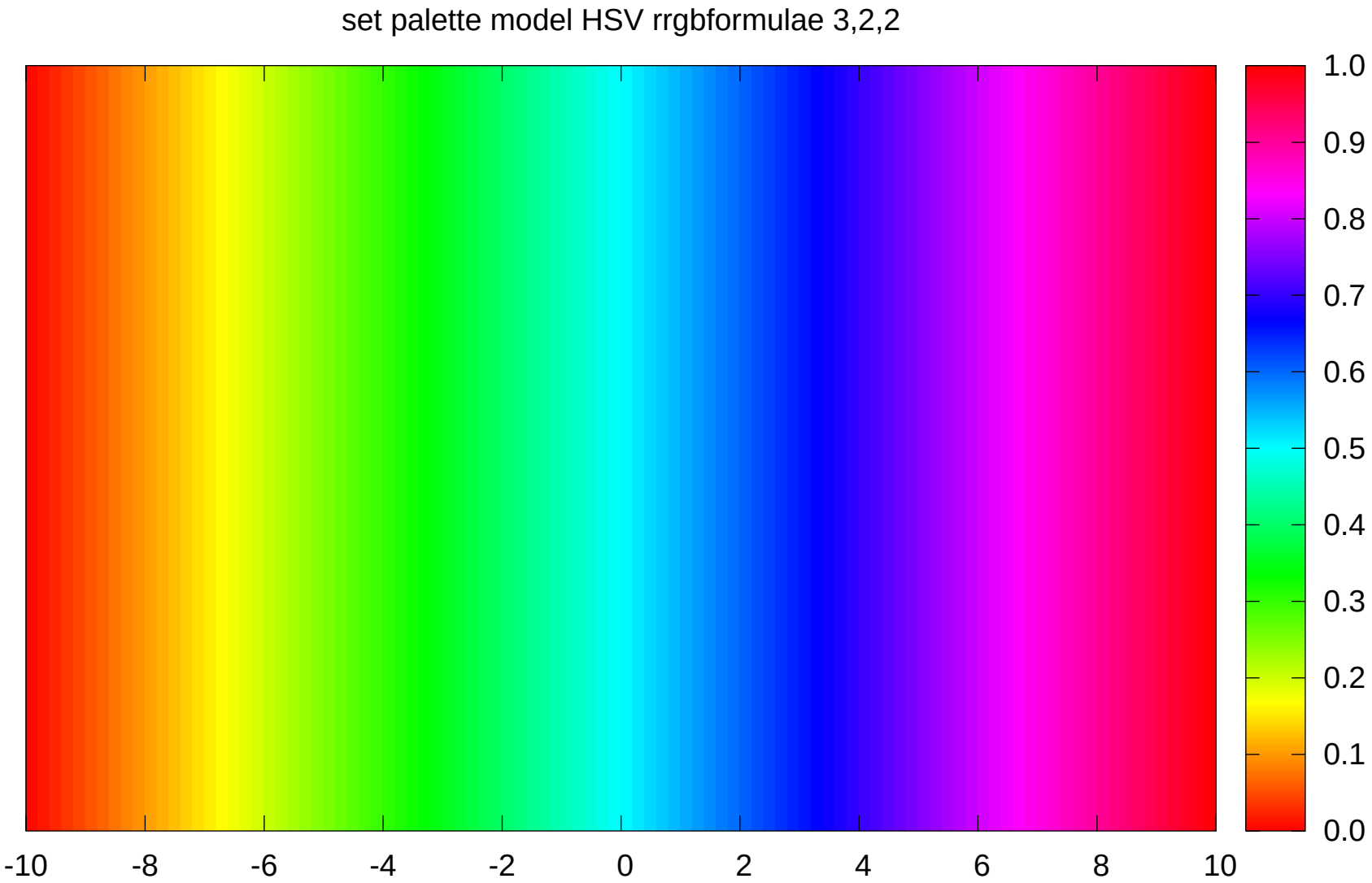

set palette model HSV rgbformulae 7,5,15
-10
-8
-6
-4
-2
0
2
4
6
8
10
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

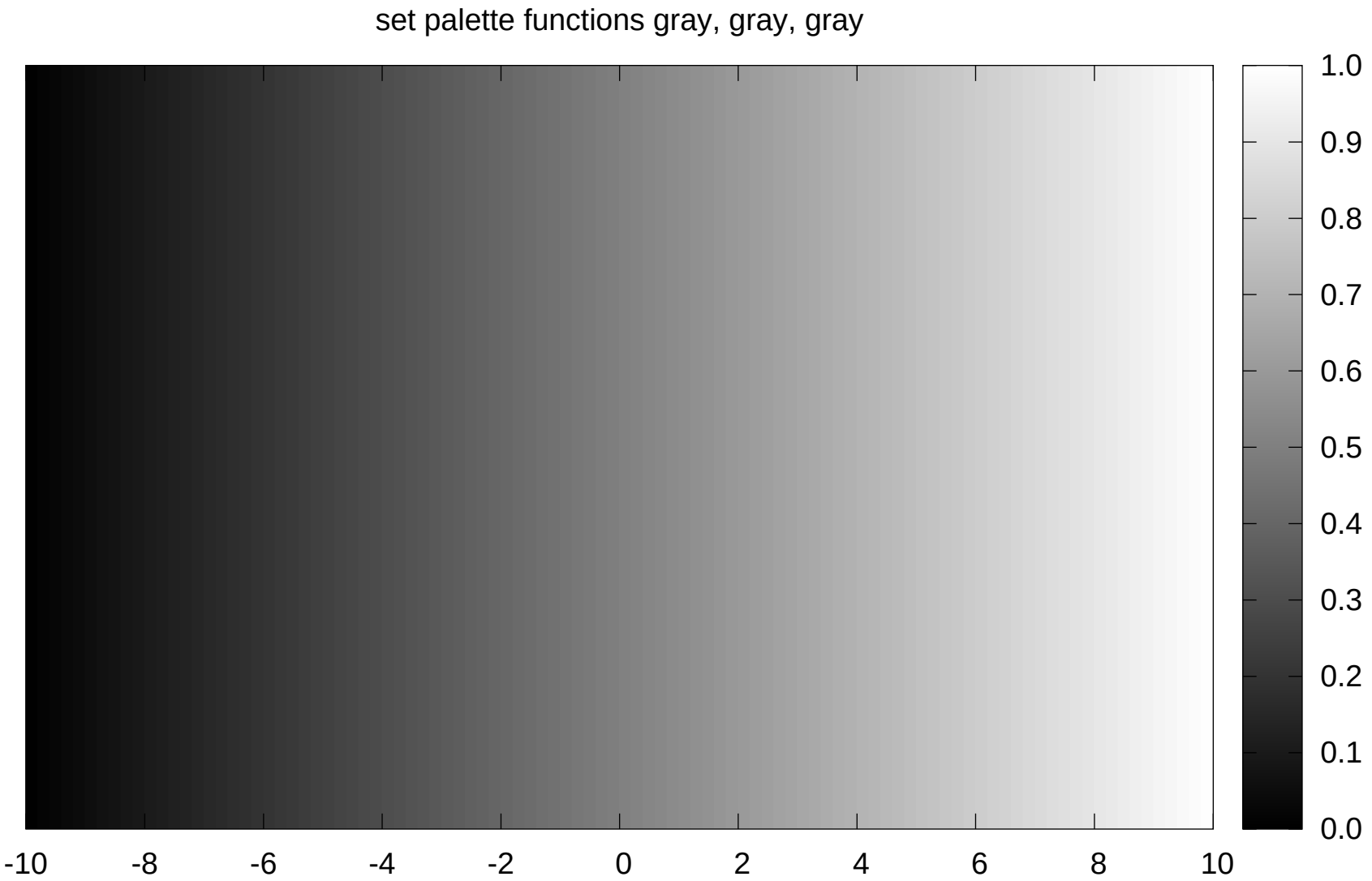
set palette model XYZ rgbformulae 7,5,15

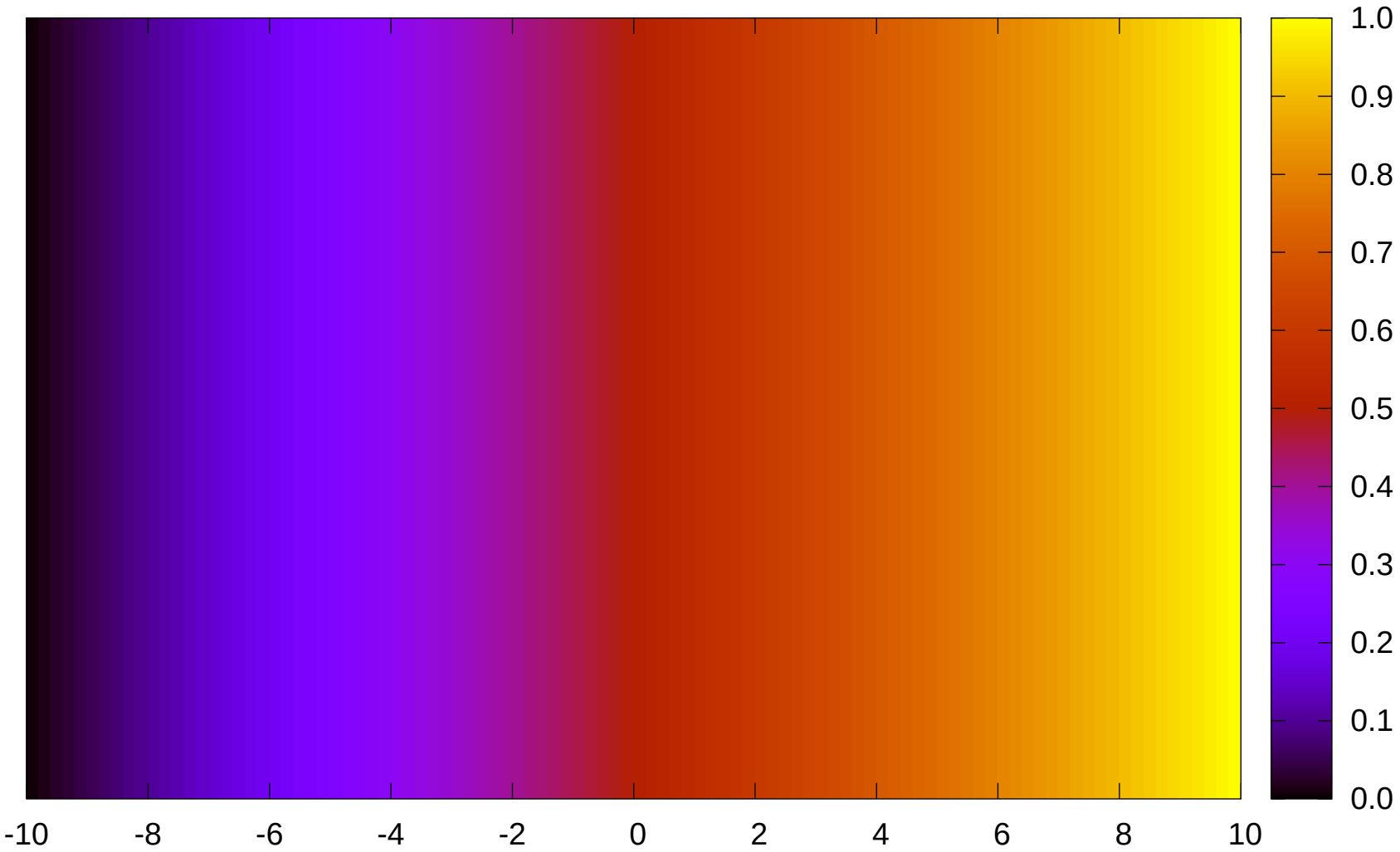


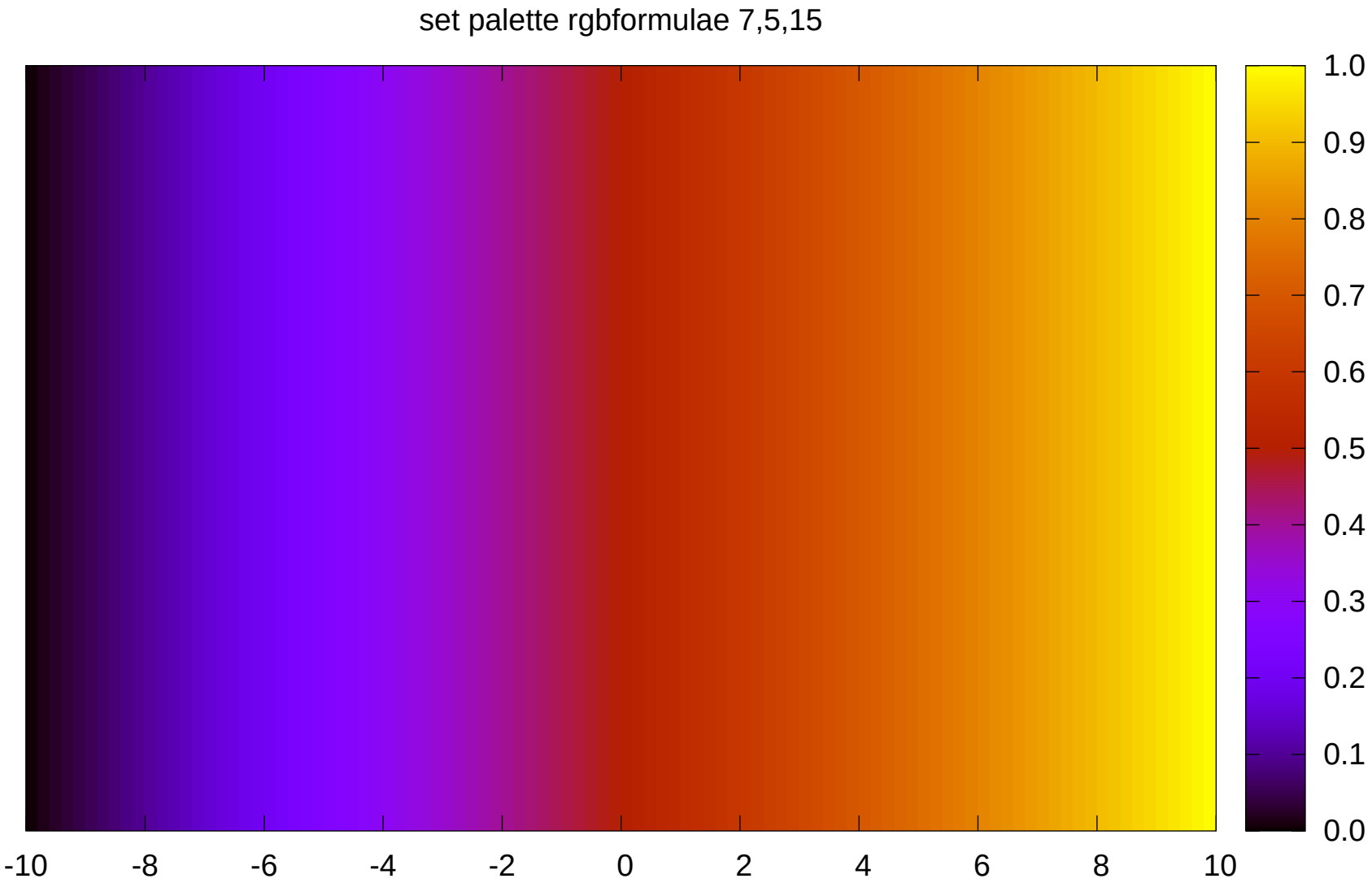

set palette model CMY rgbformulae 7,5,15
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-10
-8
-6
-4
-2
0
2
4
6
8
10

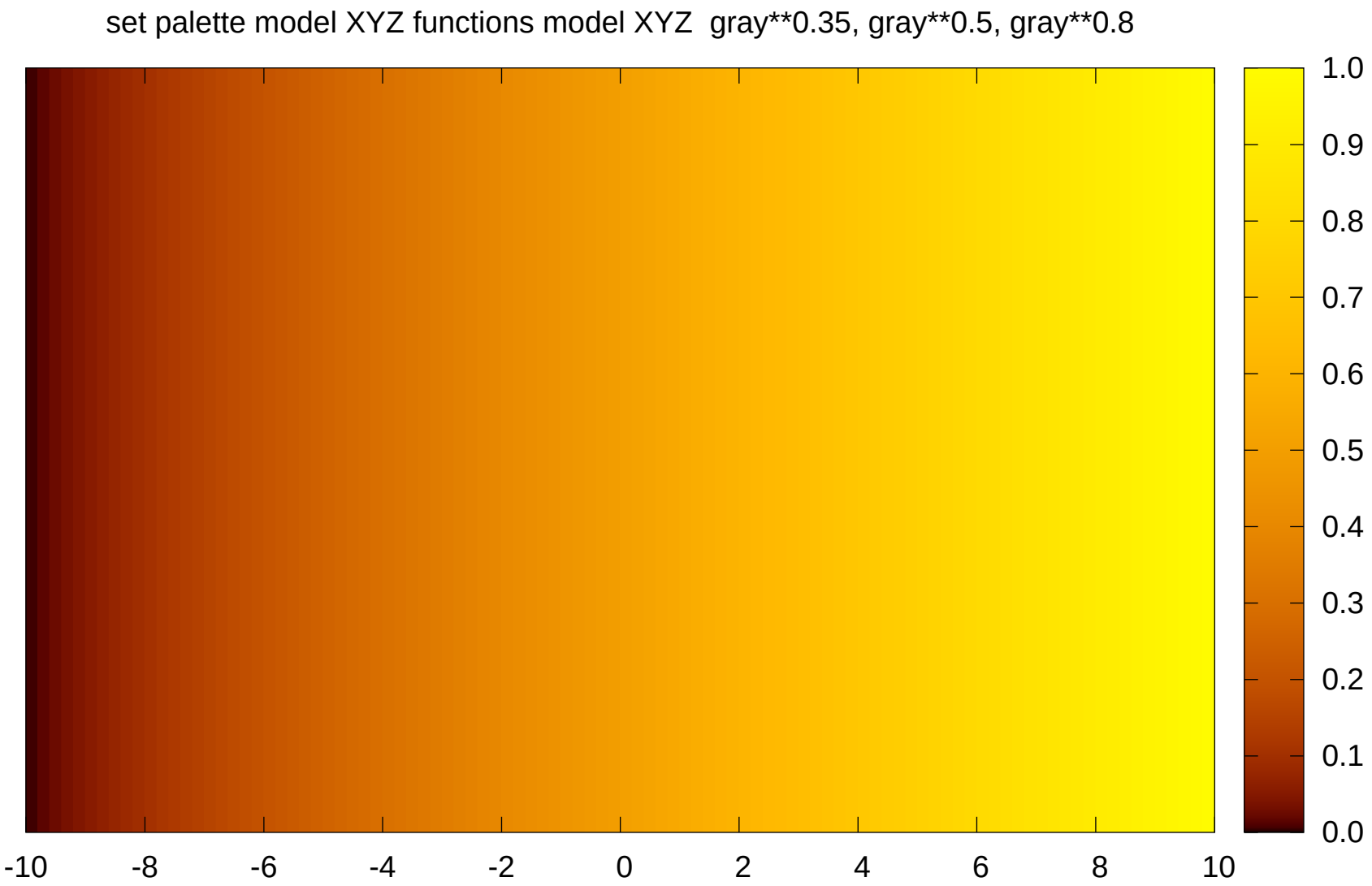

set palette model YIQ rgbformulae 7,5,15
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-10
-8
-6
-4
-2
0
2
4
6
8
10

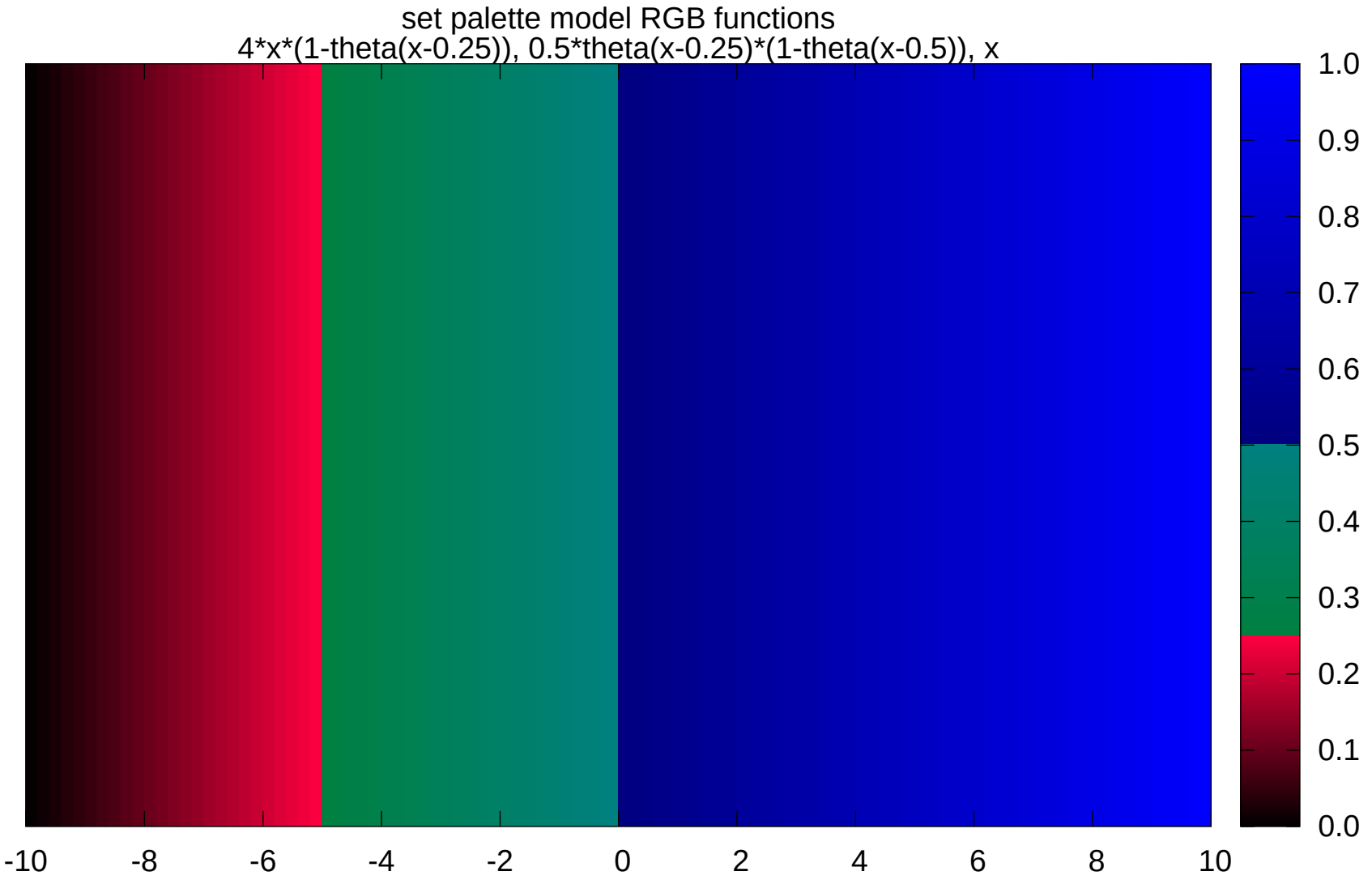

set palette model HSV rrgbformulae 3,2,2
-10
-8
-6
-4
-2
0
2
4
6
8
10
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

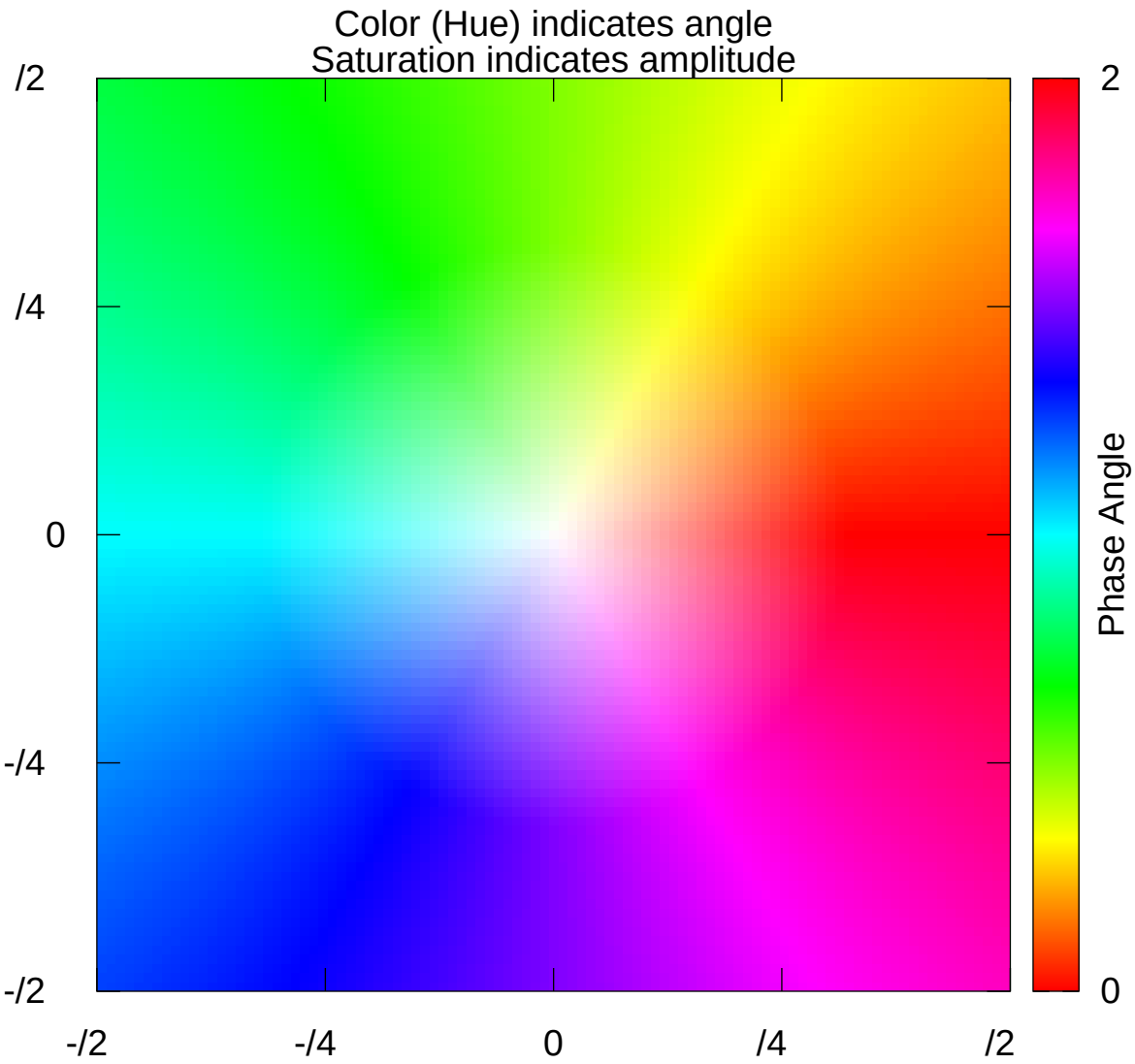

set palette functions gray, gray, gray
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-10
-8
-6
-4
-2
0
2
4
6
8
10

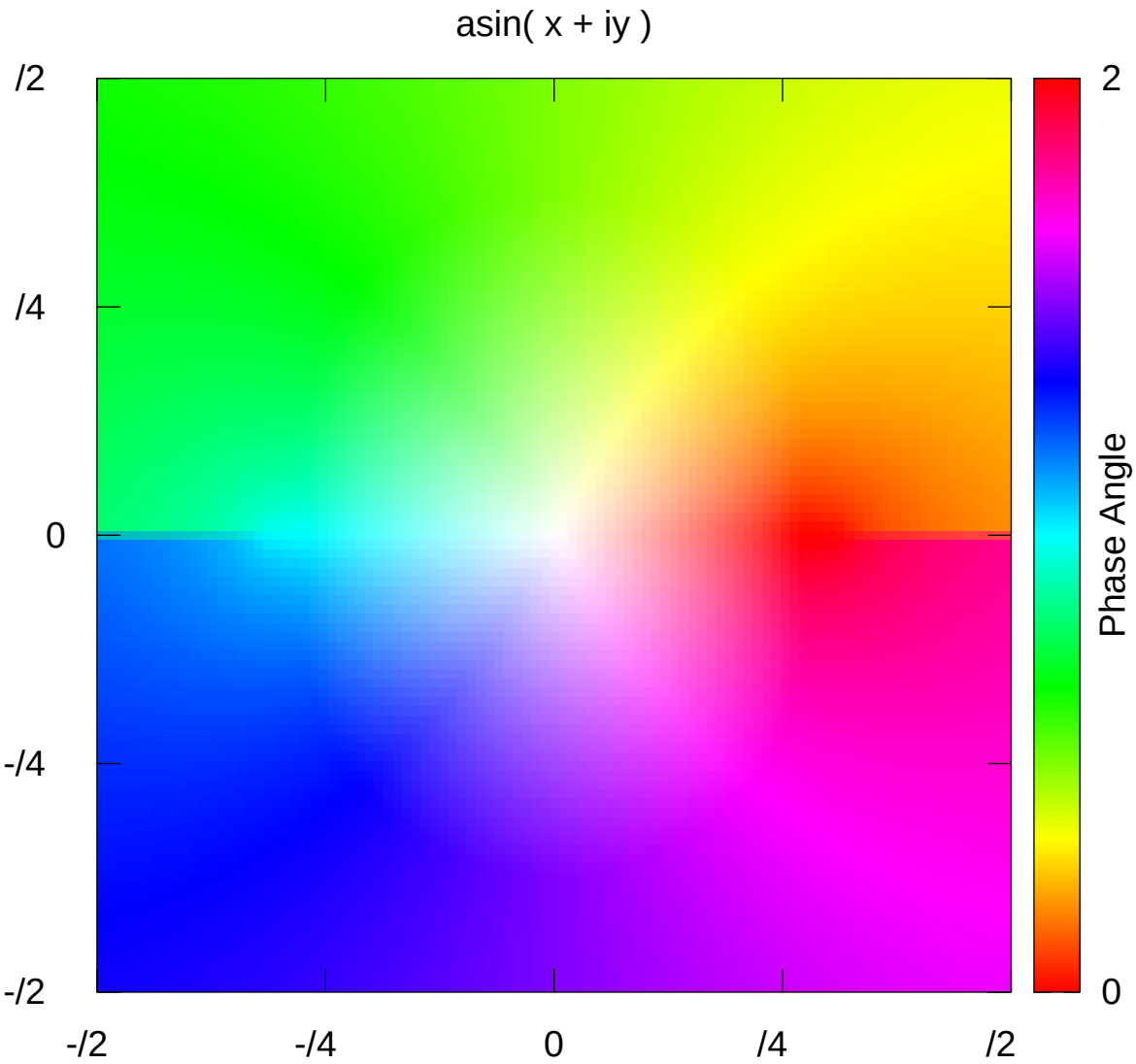

set palette functions sqrt(gray), gray**3, sin(gray*2*pi) <--> 7,5,15
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-10
-8
-6
-4
-2
0
2
4
6
8
10

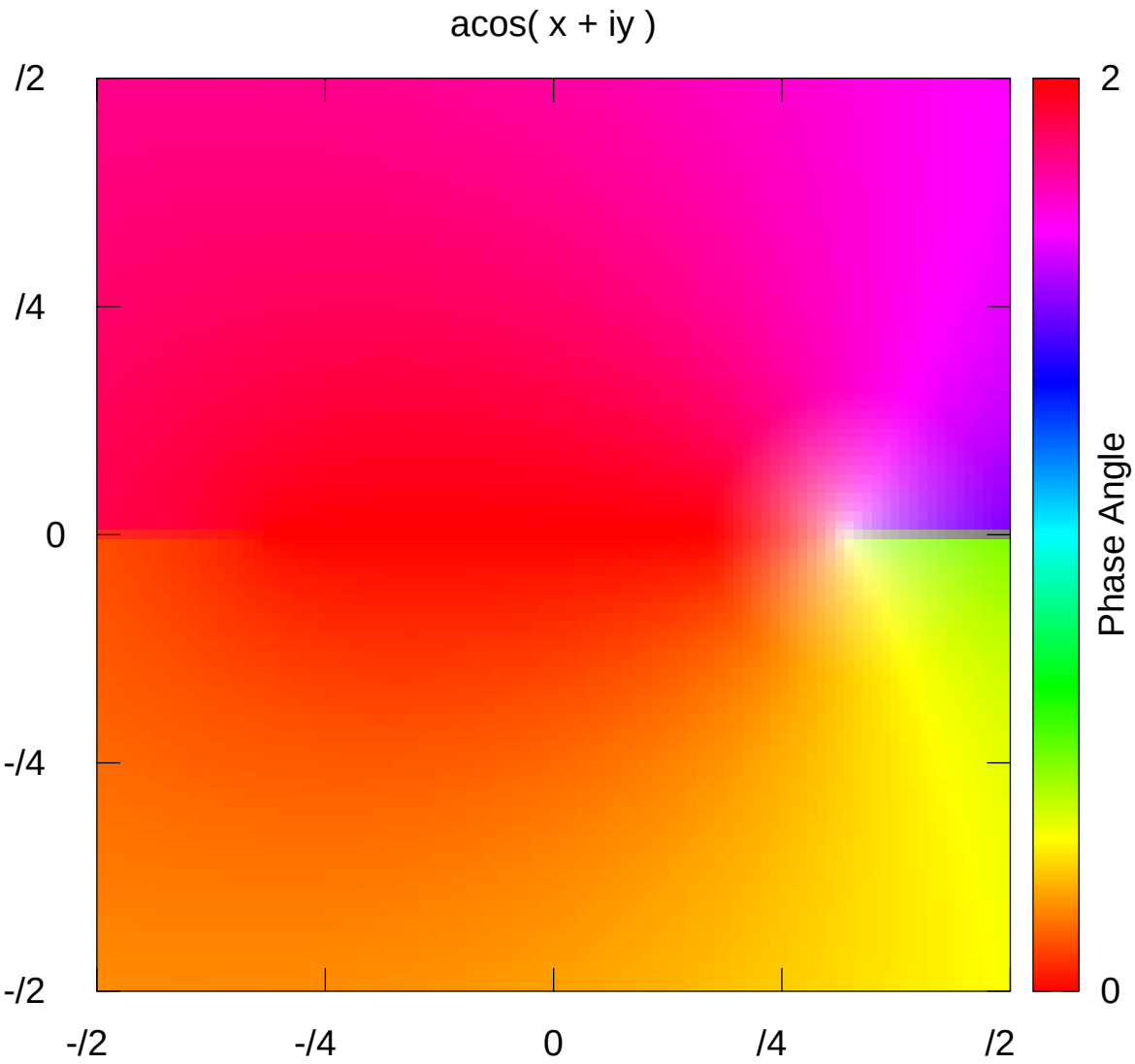

set palette rgbformulae 7,5,15
-10
-8
-6
-4
-2
0
2
4
6
8
10
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

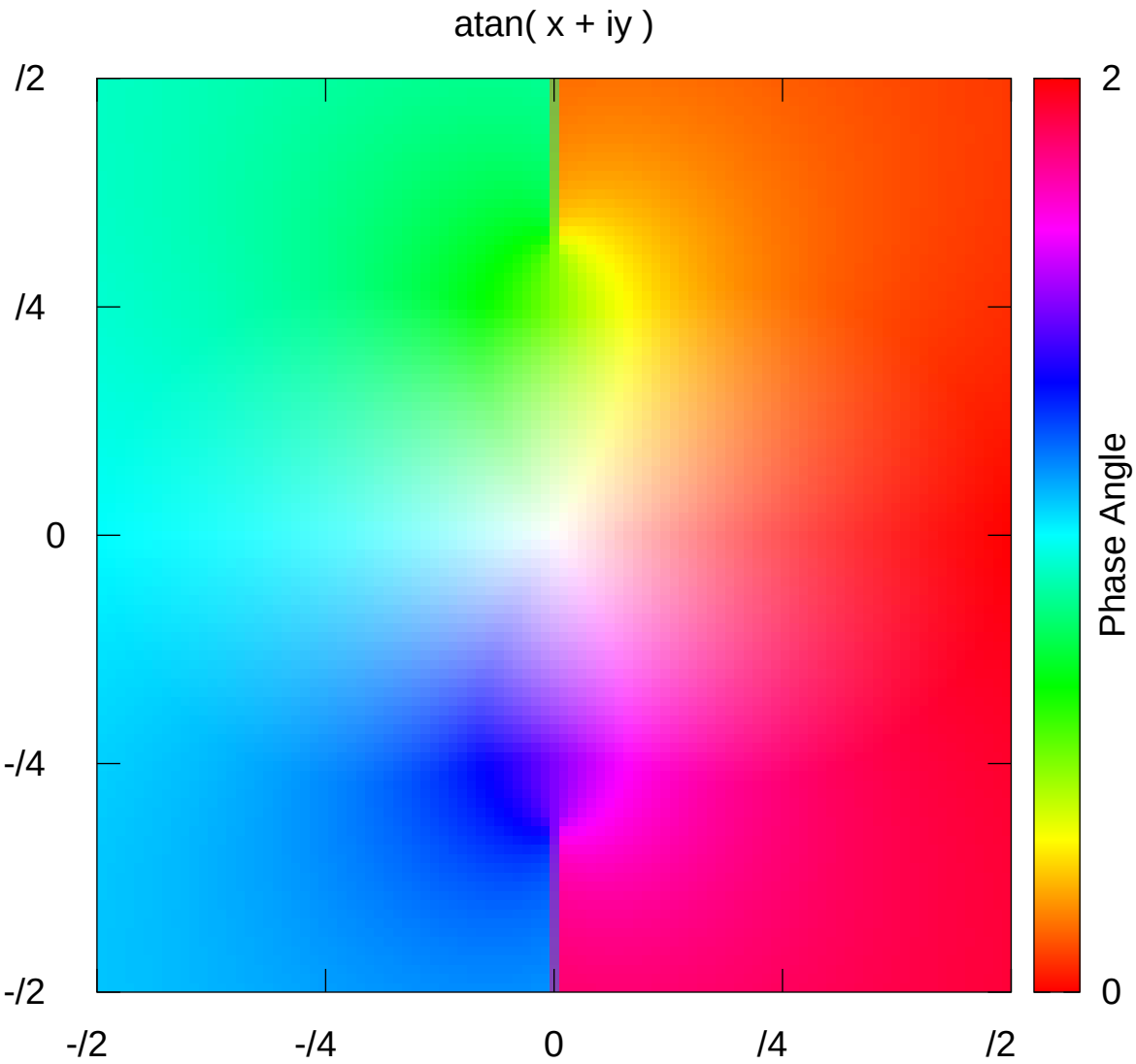

set palette model XYZ functions model XYZ gray**0.35, gray**0.5, gray**0.8
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-10
-8
-6
-4
-2
0
2
4
6
8
10

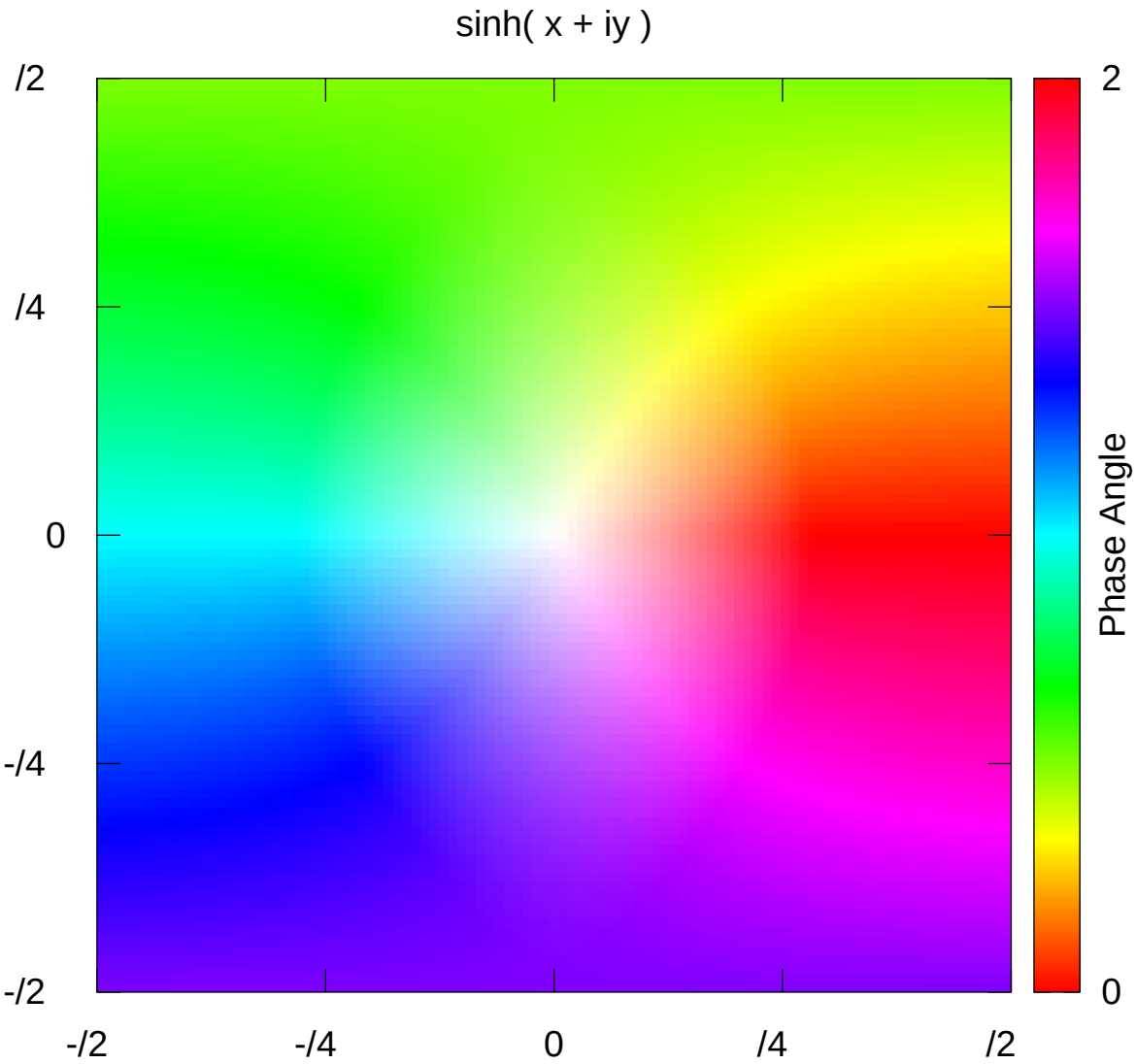

set palette model RGB functions
4*x*(1-theta(x-0.25)), 0.5*theta(x-0.25)*(1-theta(x-0.5)), x
-10
-8
-6
-4
-2
0
2
4
6
8
10
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

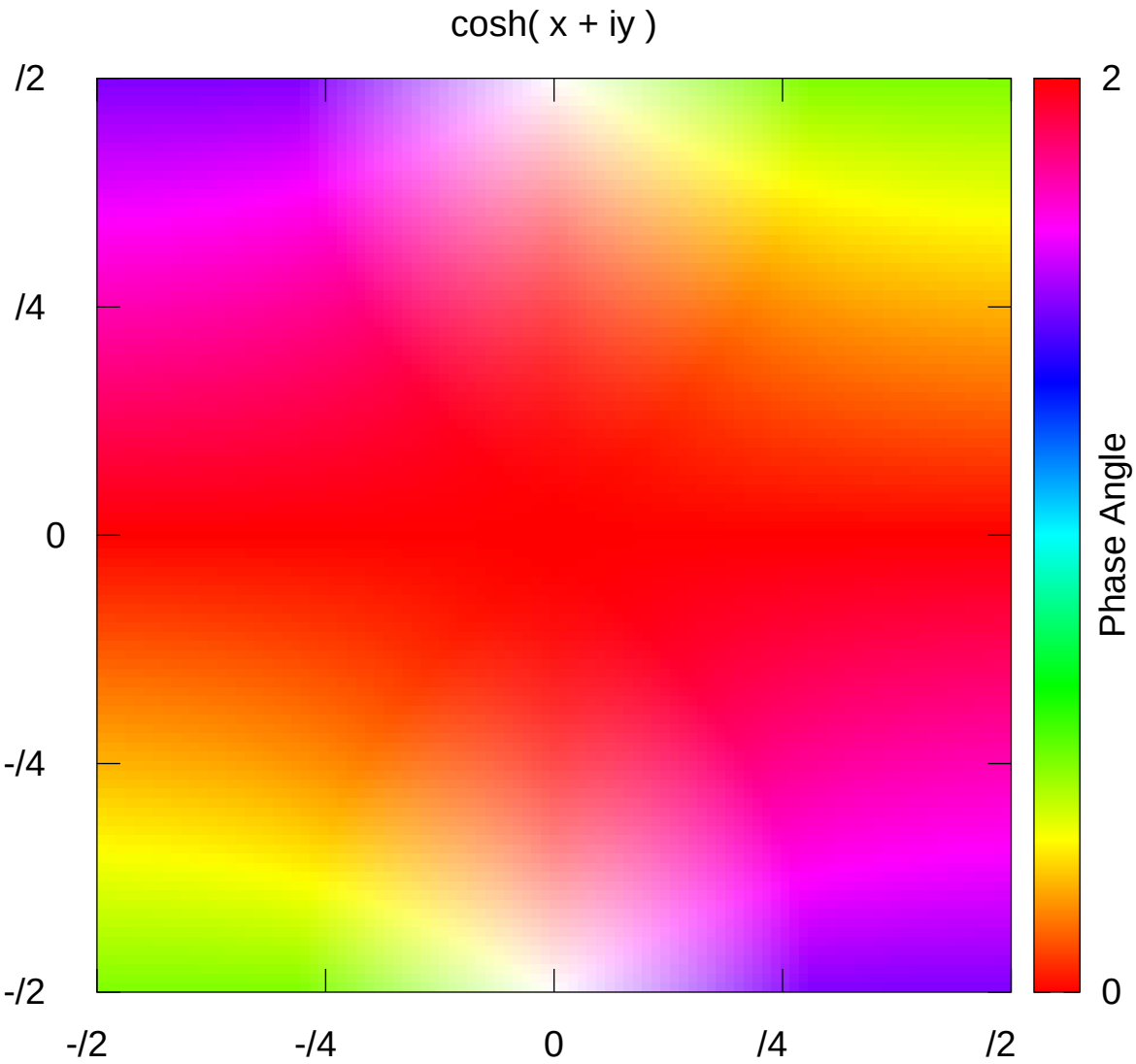

Color (Hue) indicates angle
Saturation indicates amplitude
/2
/4
0
-/4
-/2
-/2
-/4
0
/4
/2
Phase Angle
2
0

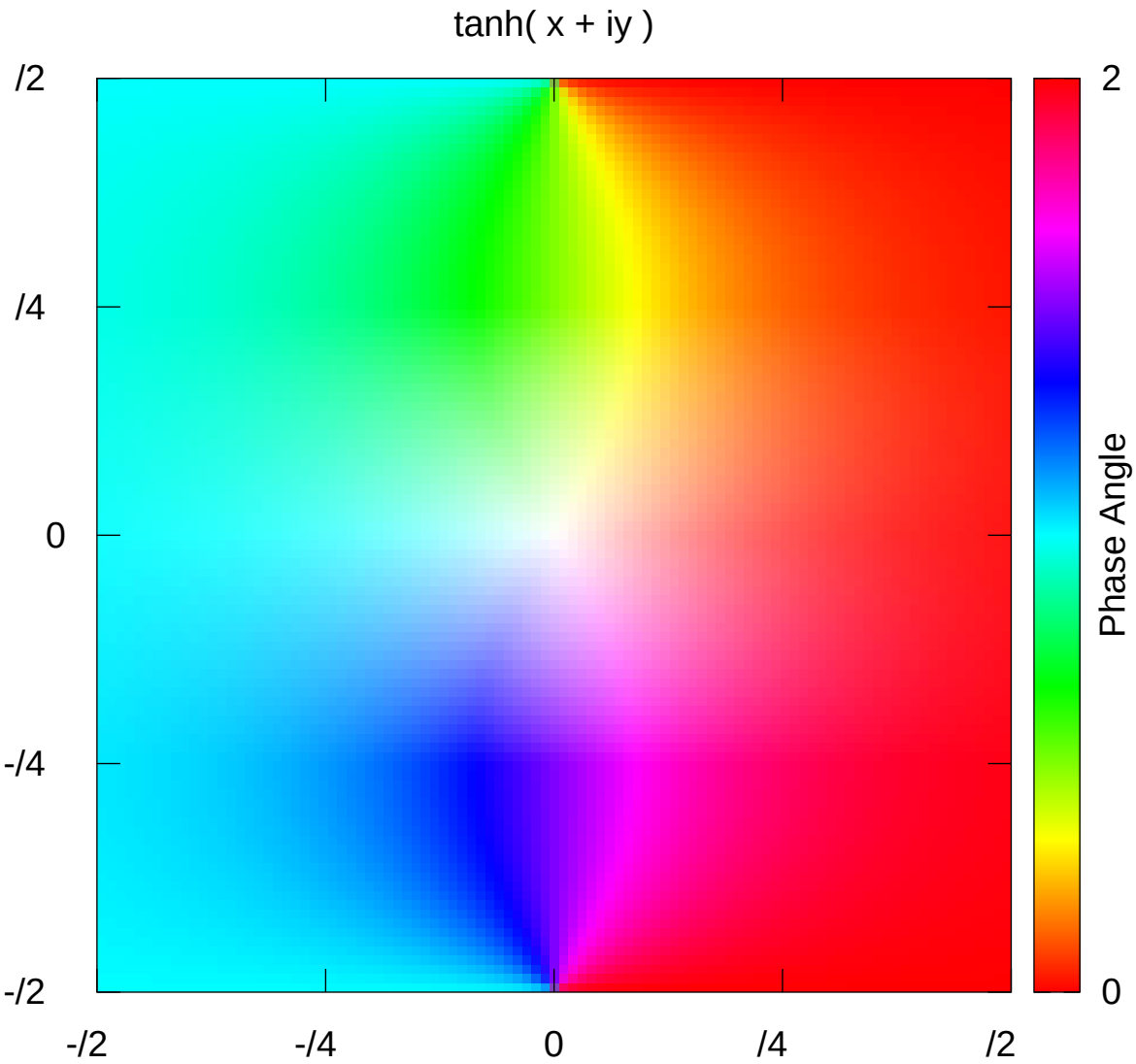

asin(x + iy)
/2
/4
0
-/4
-/2
-/2
-/4
0
/4
/2
2
Phase Angle
0

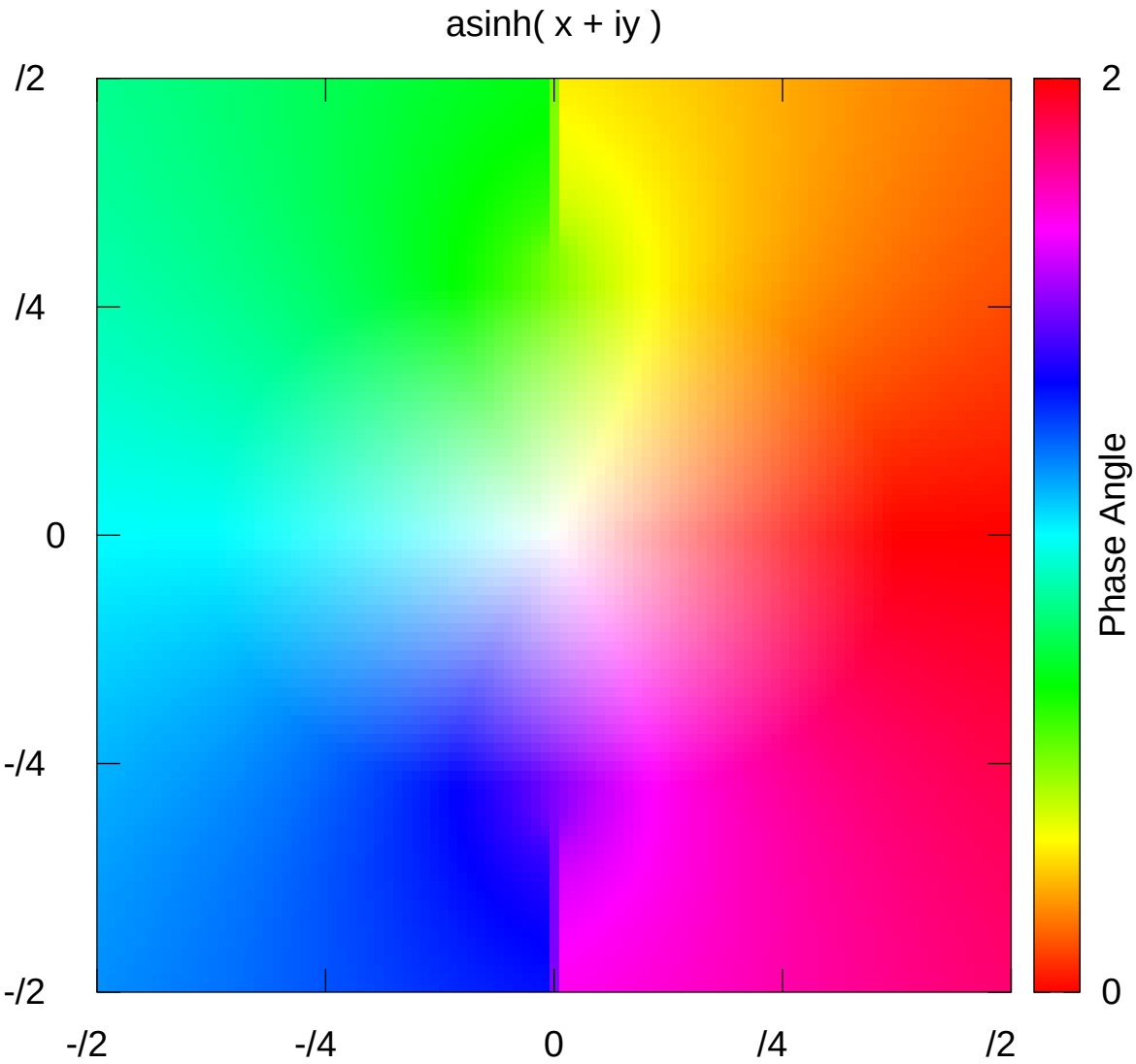

acos(x + iy)
/2
/4
0
-/4
-/2
-/2
-/4
0
/4
/2
Phase Angle
2
0

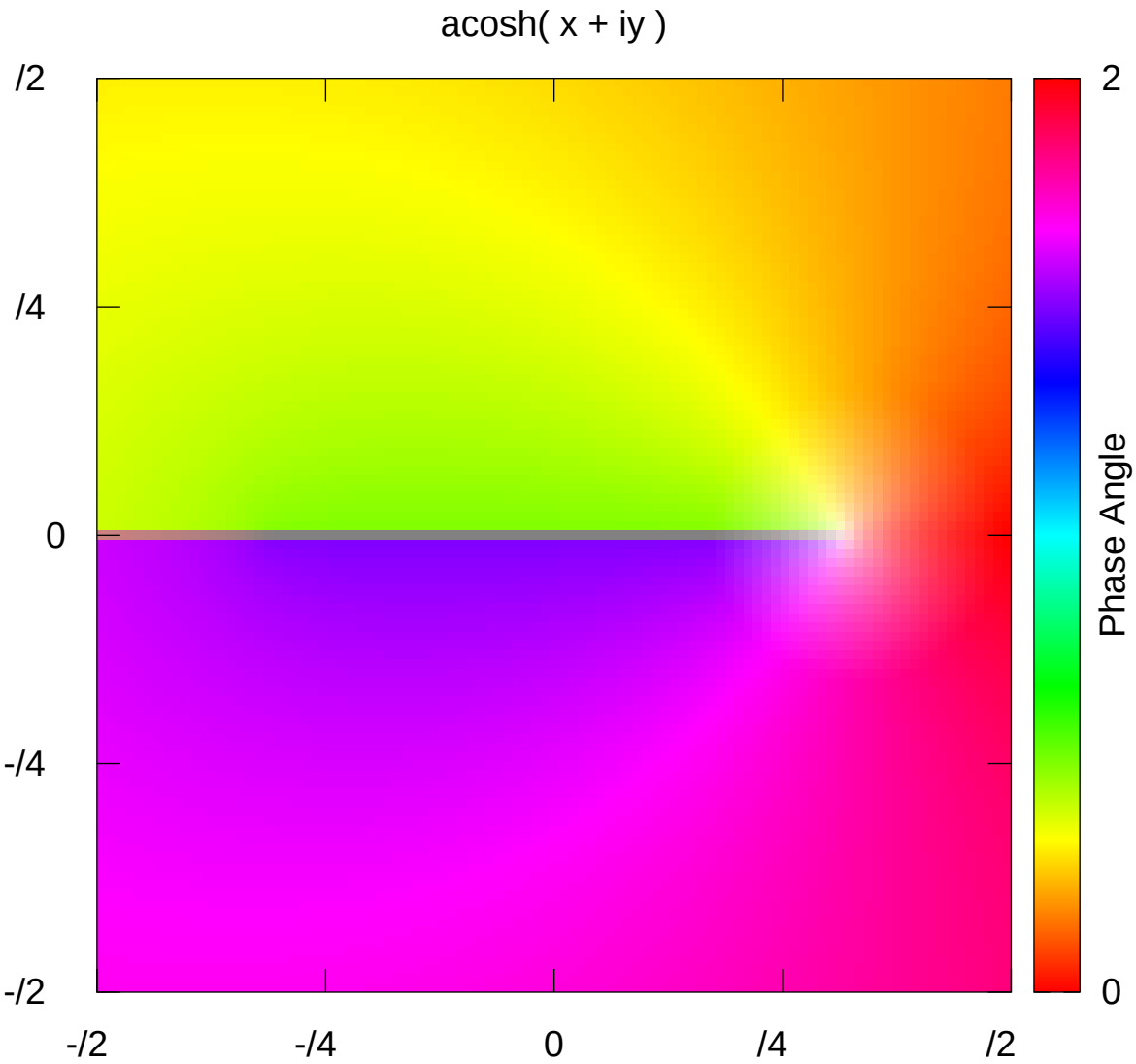

atan(x + iy)
/2
/4
0
-/4
-/2
-/2
-/4
0
/4
/2
2
Phase Angle
0

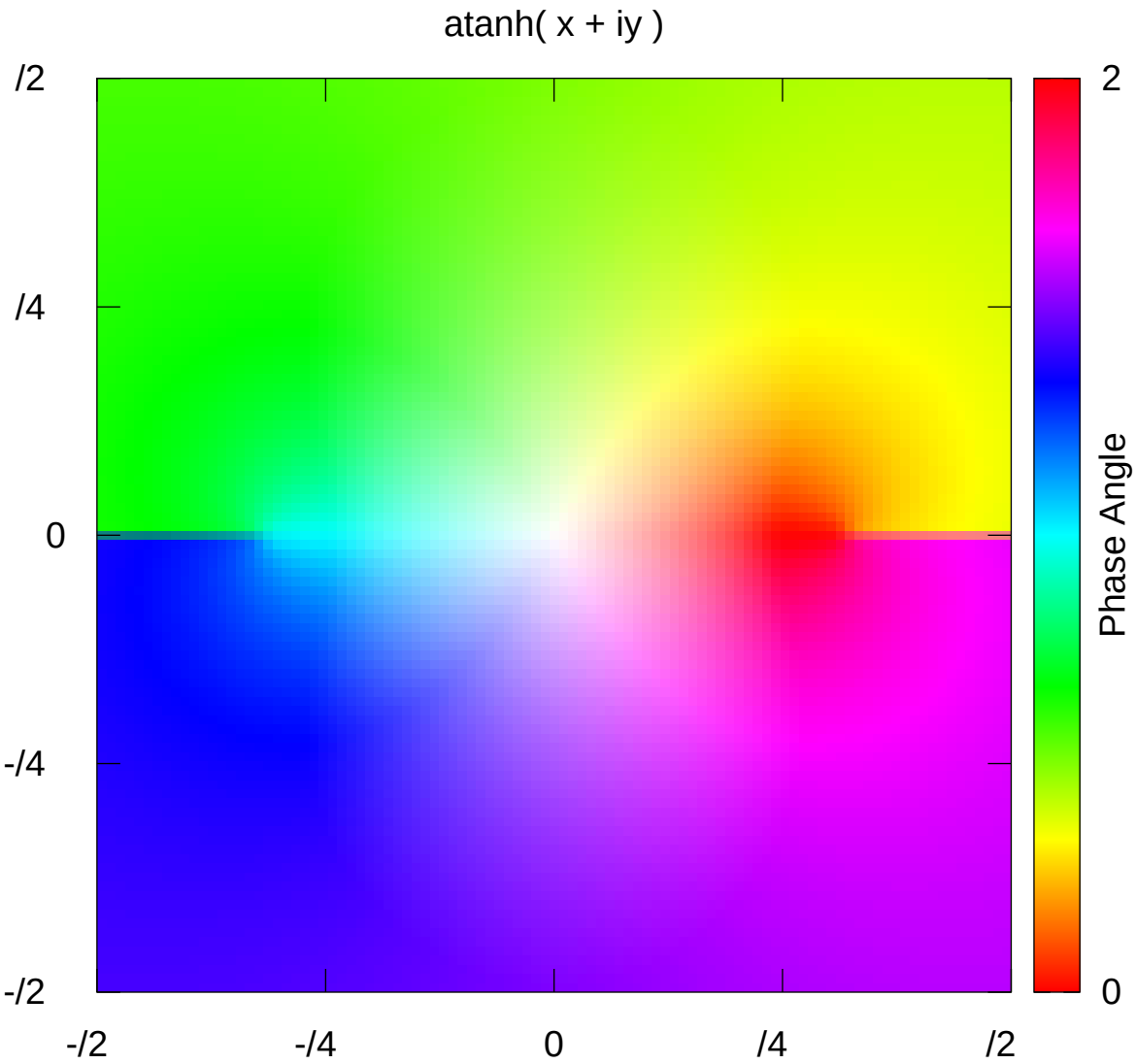

sinh(x + iy)
/2
/4
0
-/4
-/2
-/2
-/4
0
/4
/2
2
Phase Angle
0

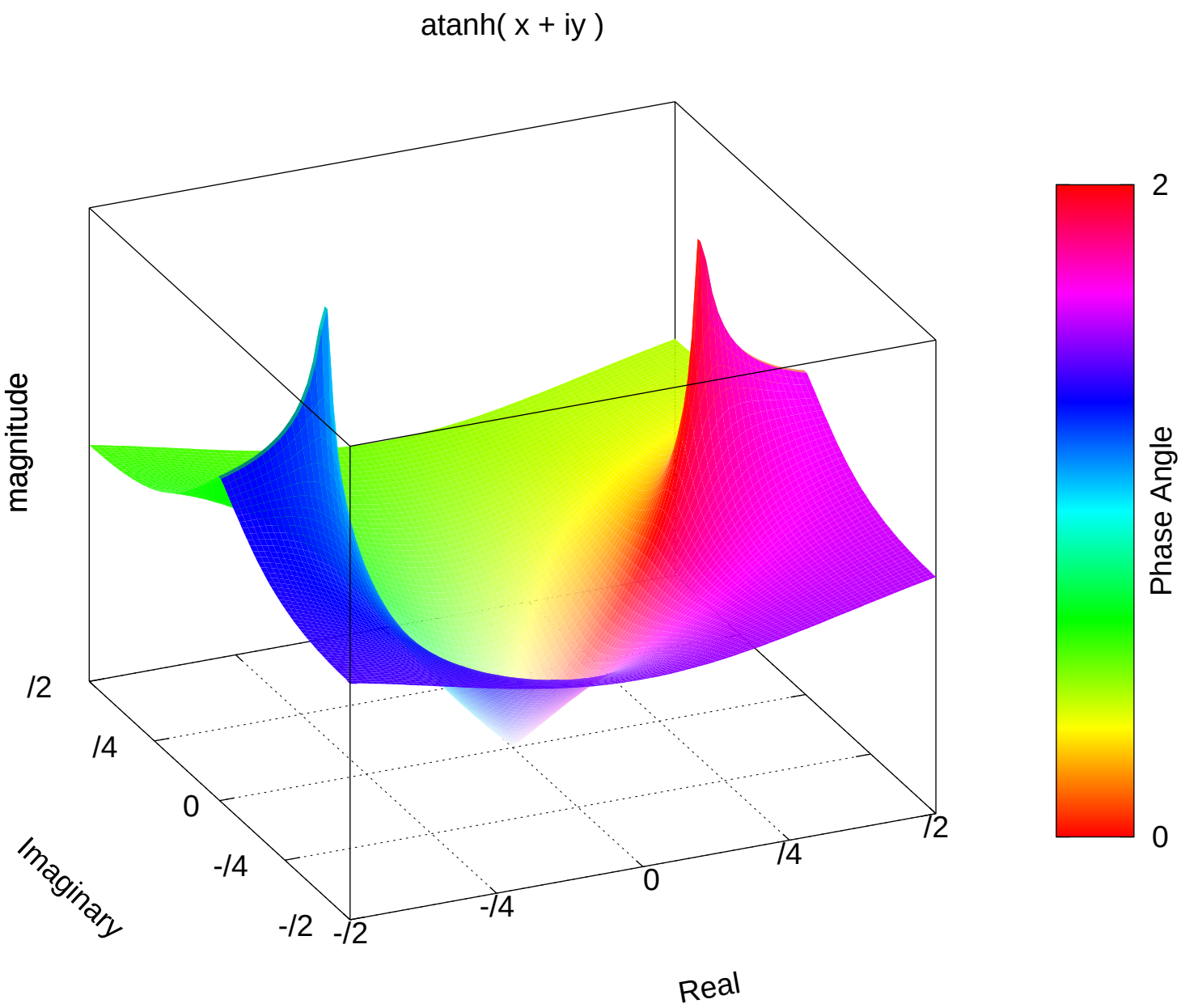

cosh(x + iy)
/2
/4
0
-/4
-/2
-/2
-/4
0
/4
/2
2
Phase Angle
0

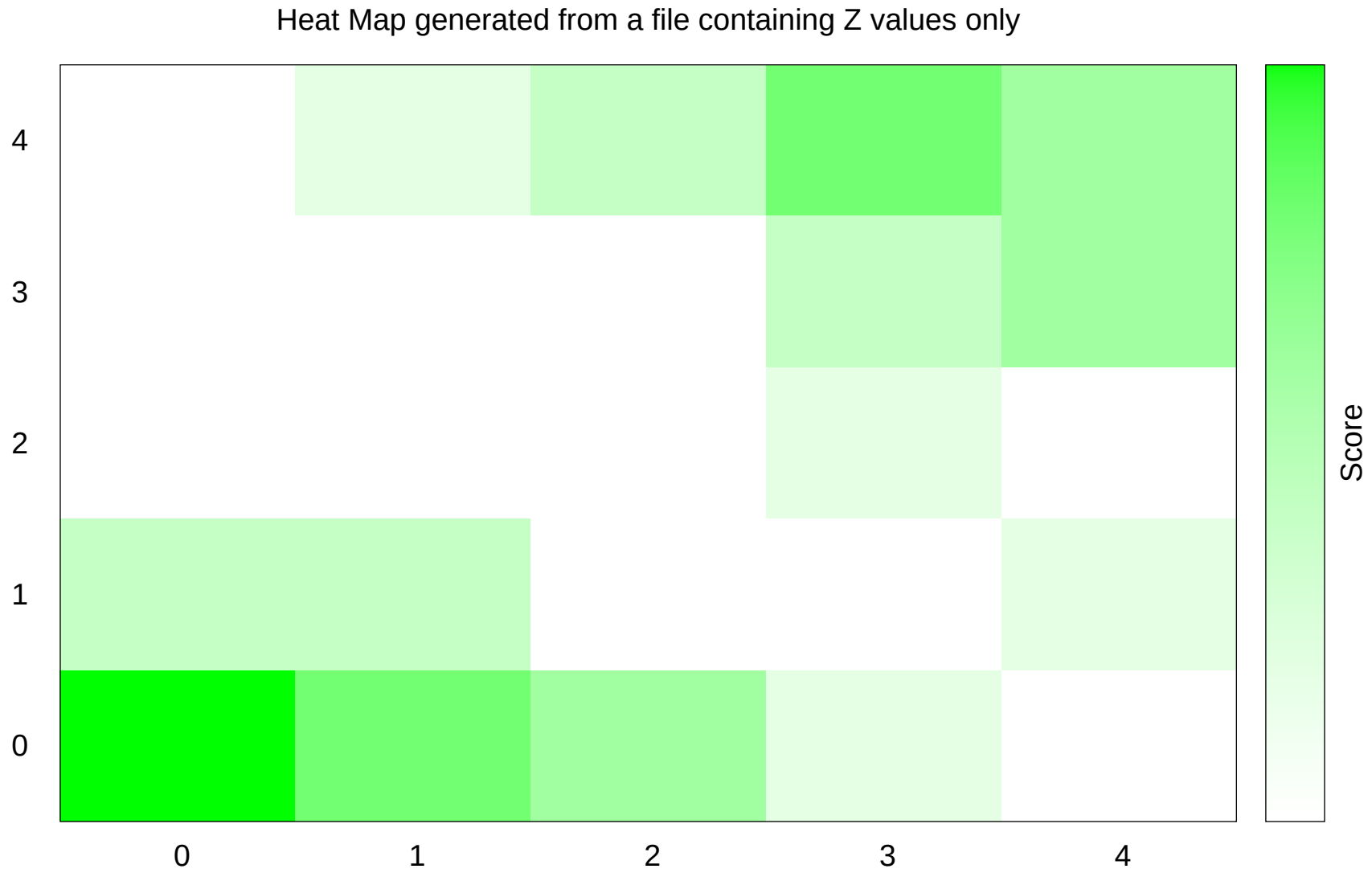

tanh(x + iy)
/2
/4
0
-/4
-/2
-/2
-/4
0
/4
/2
Phase Angle
2
0

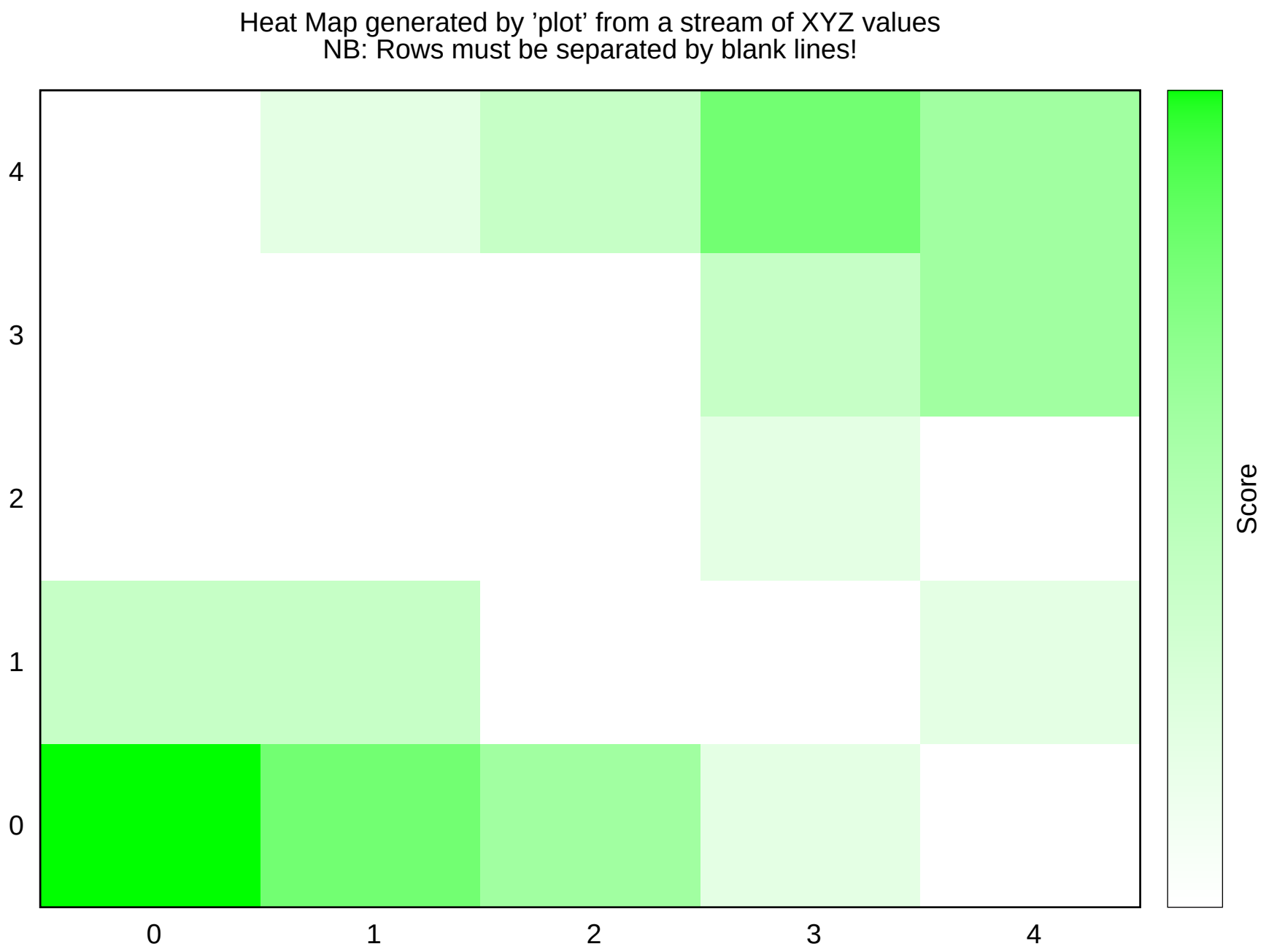

asinh(x + iy)
/2
/4
0
-/4
-/2
-/2
-/4
0
/4
/2
Phase Angle
2
0

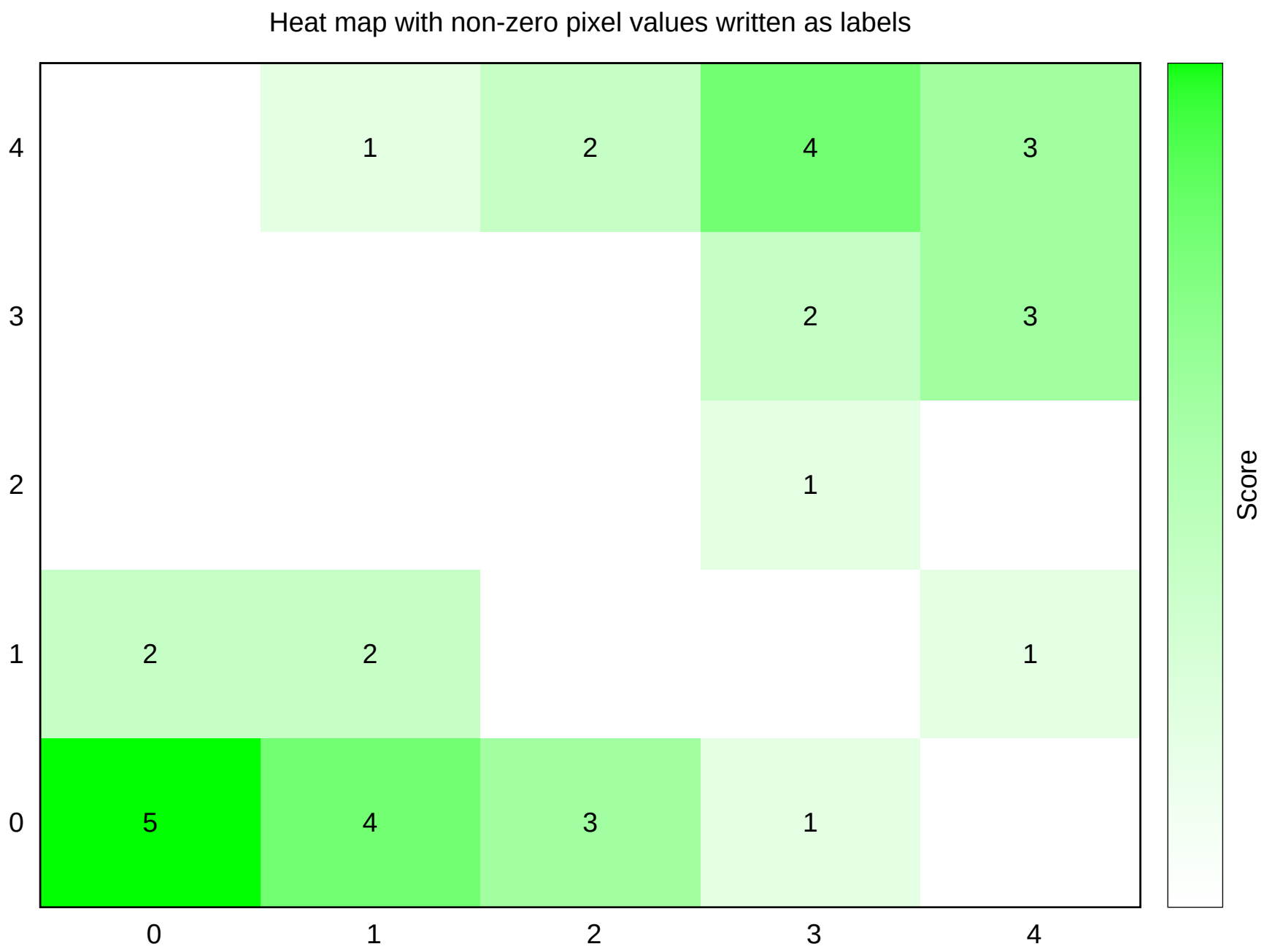

acosh(x + iy)
/2
/4
0
-/4
-/2
-/2
-/4
0
/4
/2
Phase Angle
2
0

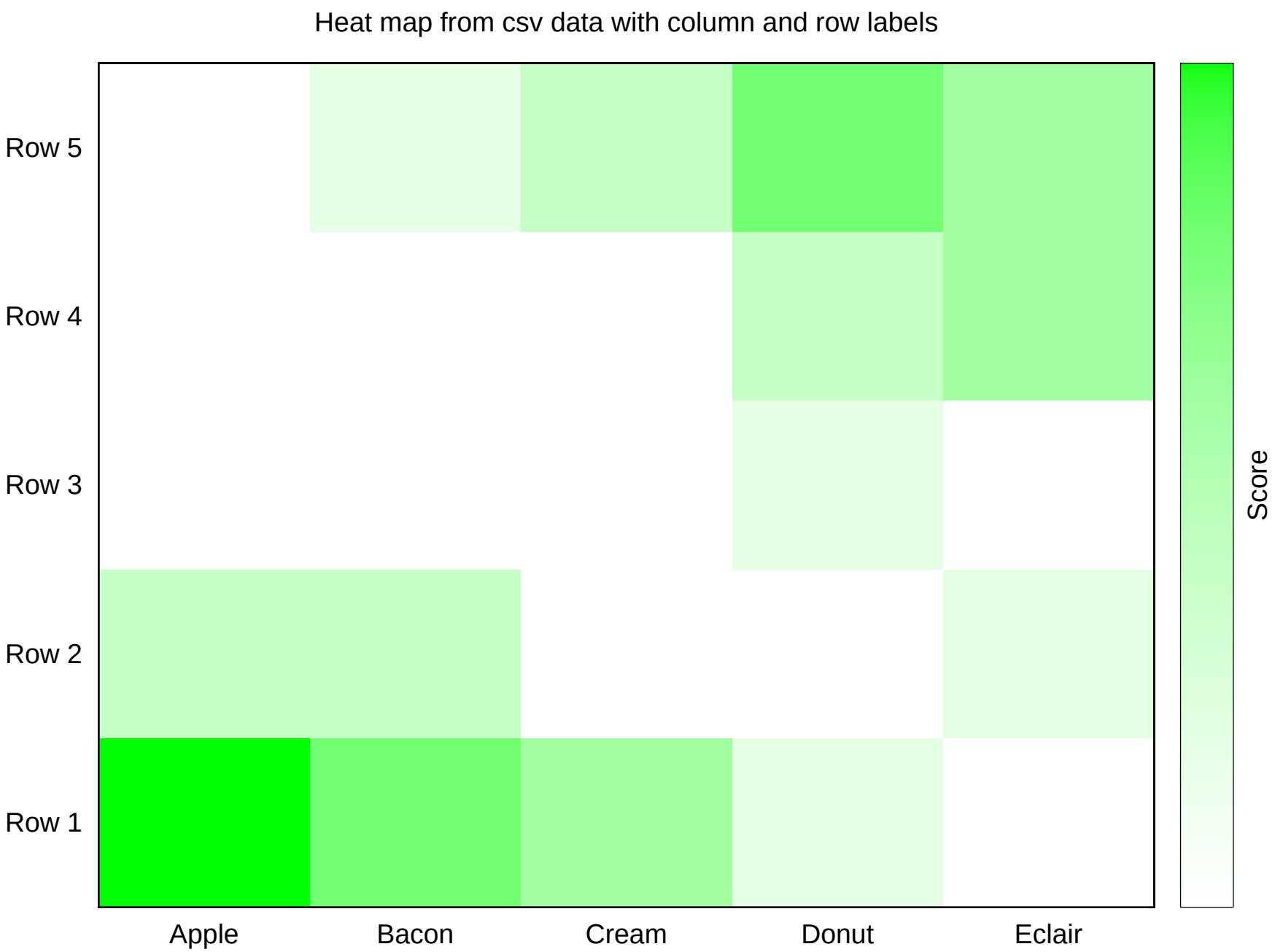

atanh(x + iy)
/2
/4
0
-/4
-/2
-/2
-/4
0
/4
/2
2
Phase Angle
0


atanh(x + iy)
magnitude
/2
/4
0
-/4
-/2
Imaginary
-/2
-/4
0
/4
/2
Real
2
0
Phase Angle


Heat Map generated from a file containing Z values only
4
3
2
1
0
0
1
2
3
4
Score

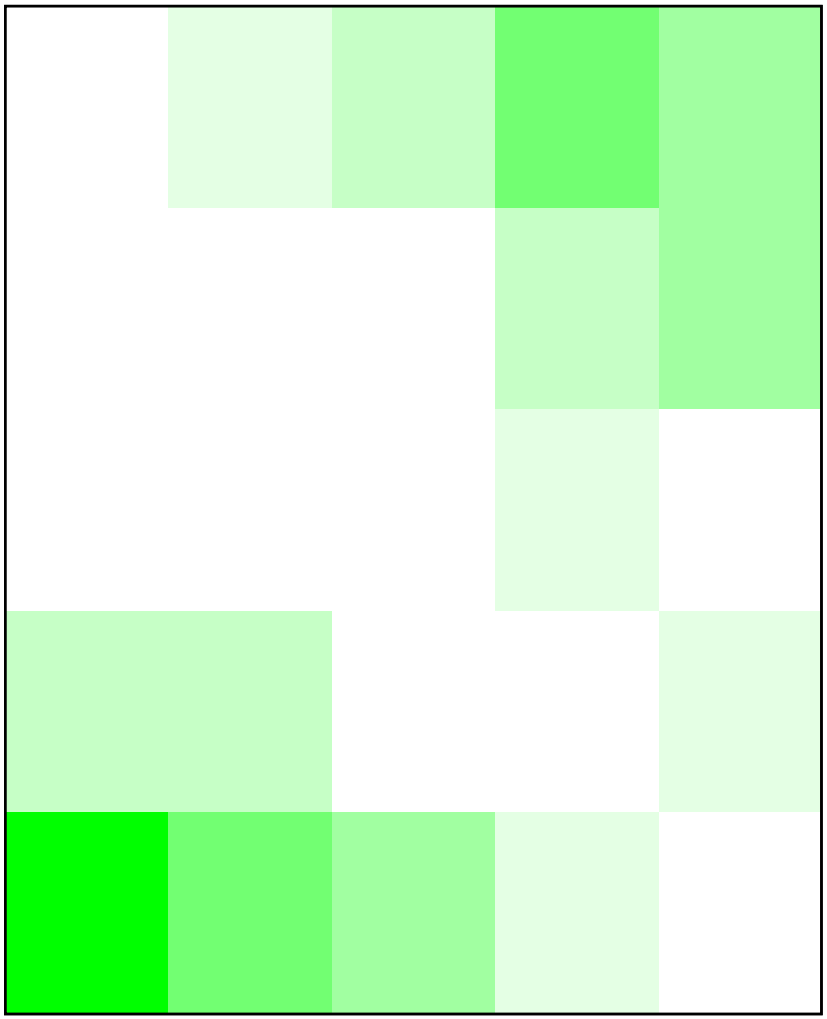

Heat Map generated by 'plot' from a stream of XYZ values
NB: Rows must be separated by blank lines!
4
3
2
1
0
0
1
2
3
4
Score


Heat map with non-zero pixel values written as labels
4
3
2
1
0
0
1
2
3
4
1
2
4
3
2
3
1
2
2
1
5
4
3
1
Score

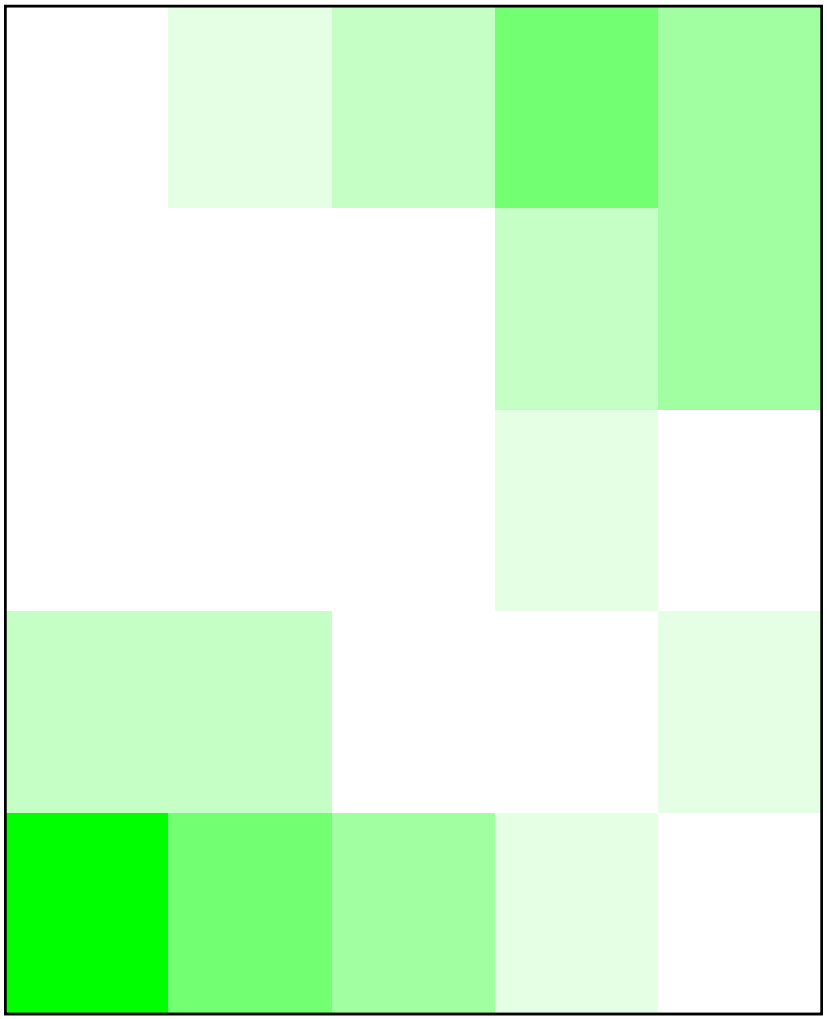

Heat map from csv data with column and row labels
Row 5
Row 4
Row 3
Row 2
Row 1
Apple
Bacon
Cream
Donut
Eclair
Score

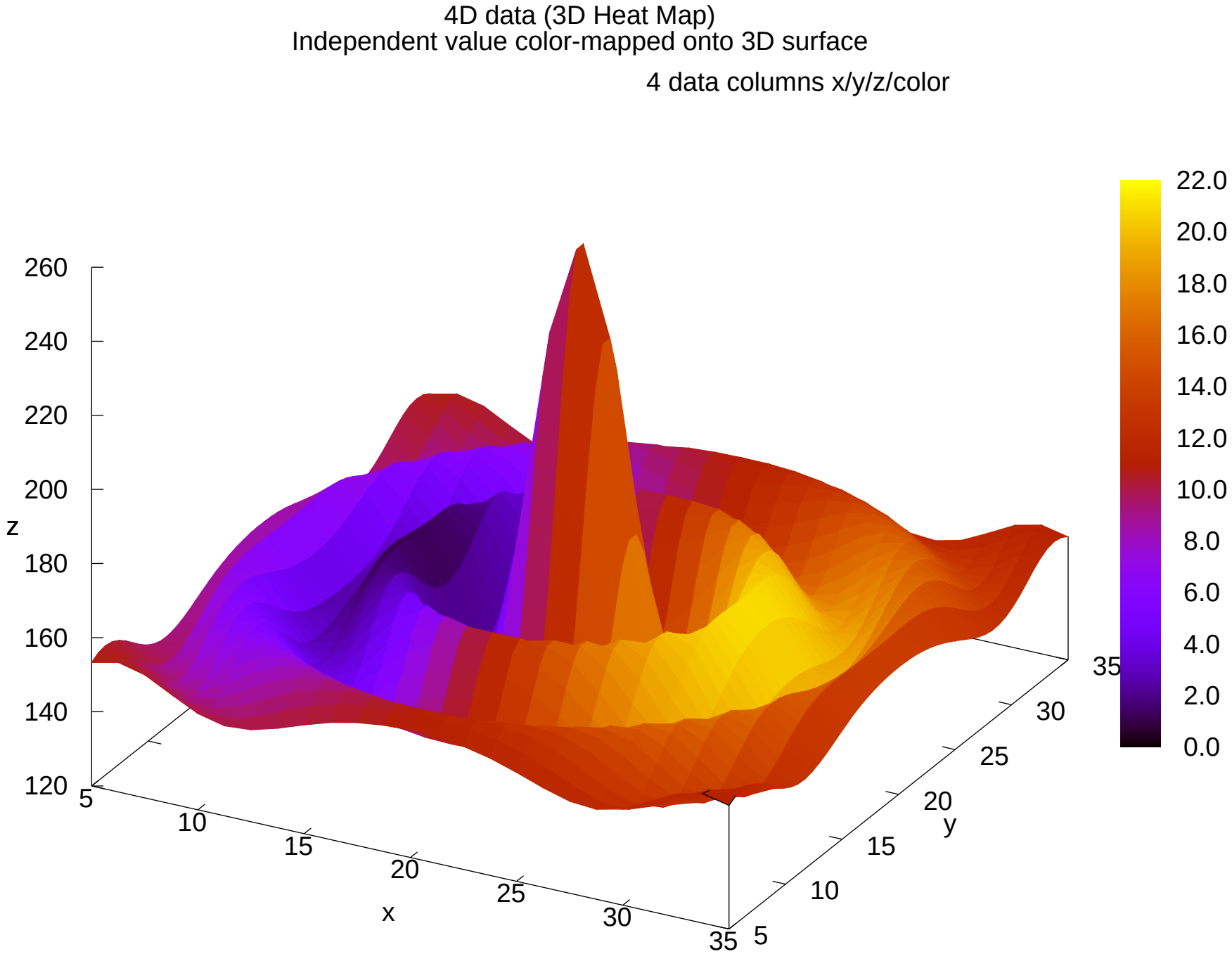
Compare 'image' and 'image pixels' modes

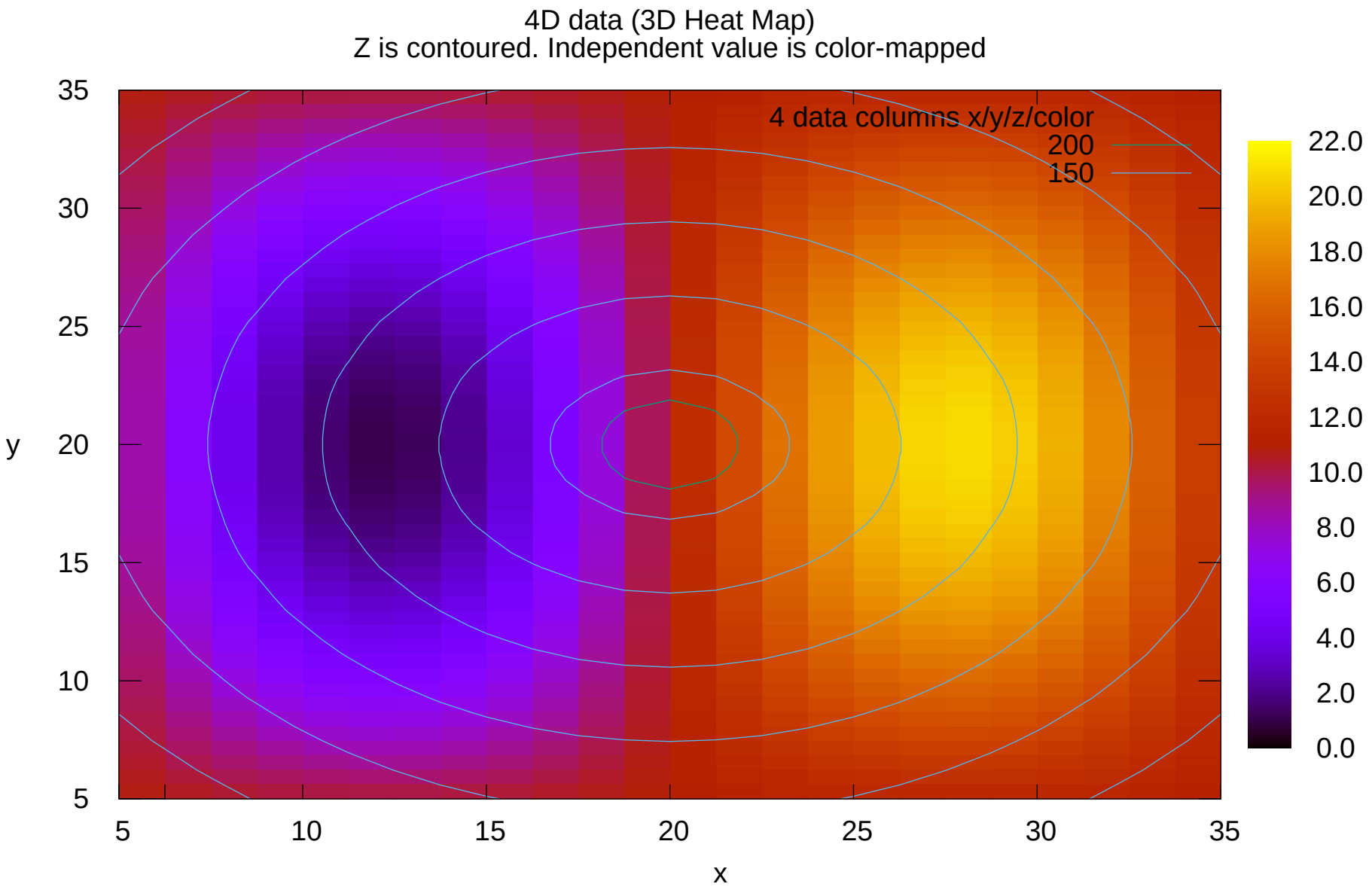
plot with image

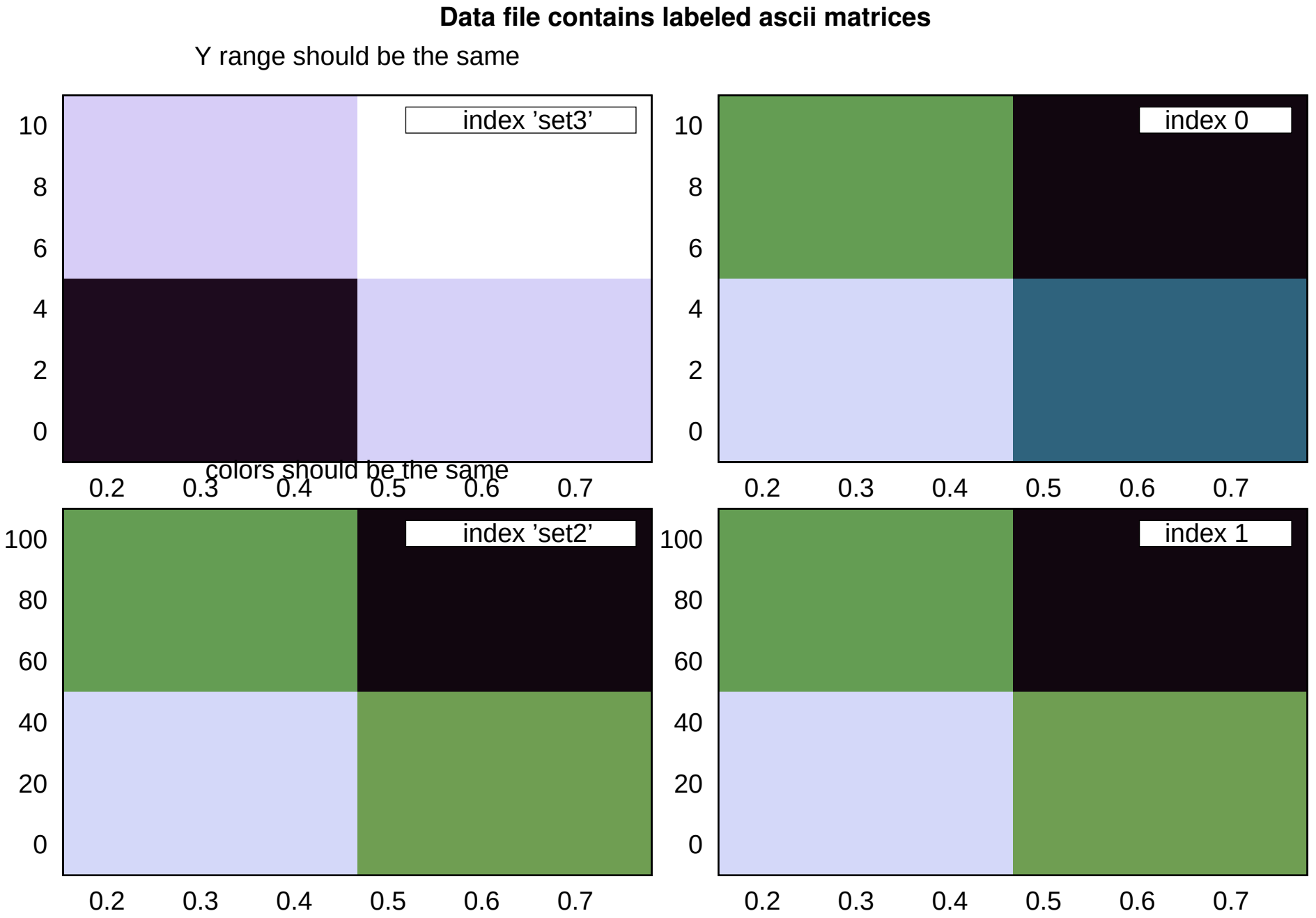


plot with image pixels

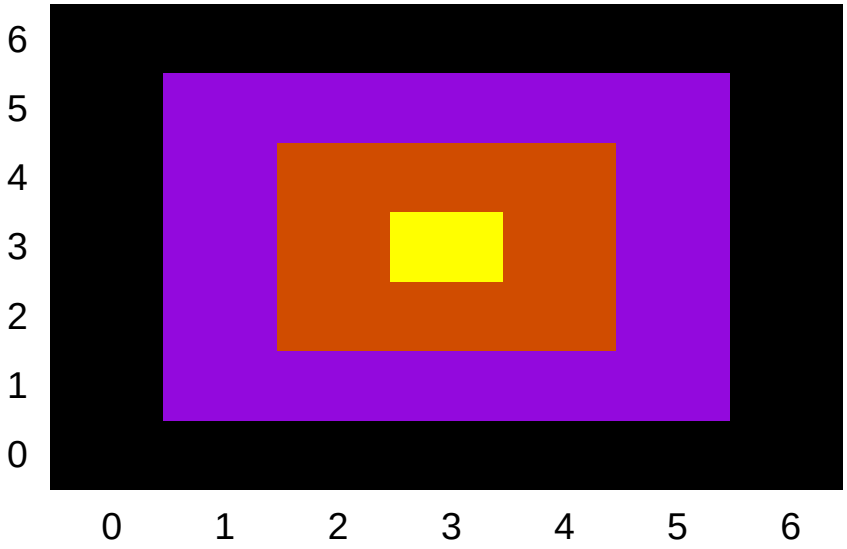



4D data (3D Heat Map)
Independent value color-mapped onto 3D surface
4 data columns x/y/z/color
z
260
240
220
200
180
160
140
120
x
5
10
15
20
25
30
35
y
5
10
15
20
25
30
35
22.0
20.0
18.0
16.0
14.0
12.0
10.0
8.0
6.0
4.0
2.0
0.0

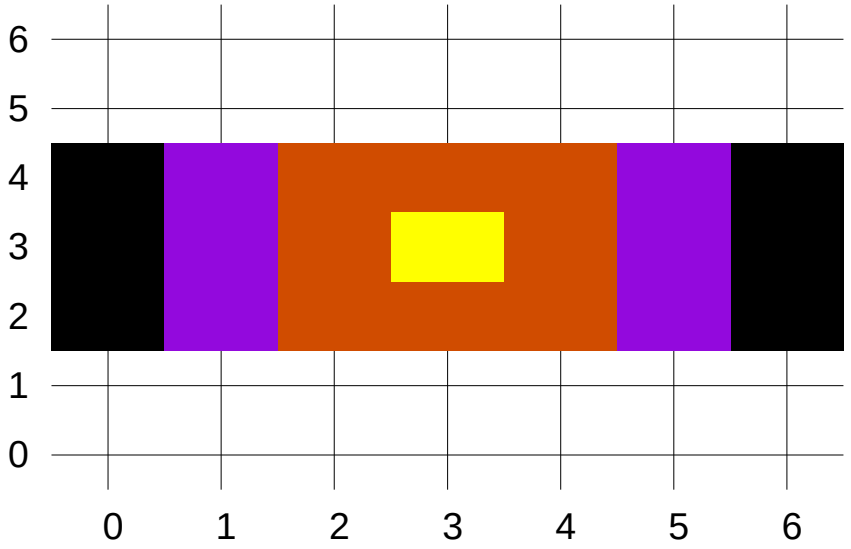

4D data (3D Heat Map)
Z is contoured. Independent value is color-mapped
4 data columns x/y/z/color
200
150
y
x
35
30
25
20
15
10
5
5
10
15
20
25
30
35
22.0
20.0
18.0
16.0
14.0
12.0
10.0
8.0
6.0
4.0
2.0
0.0


Data file contains labeled ascii matrices
Y range should be the same
index 'set3'
index 0
colors should be the same
index 'set2'
index 1
10
8
6
4
2
0
100
80
60
40
20
0
0.2
0.3
0.4
0.5
0.6
0.7

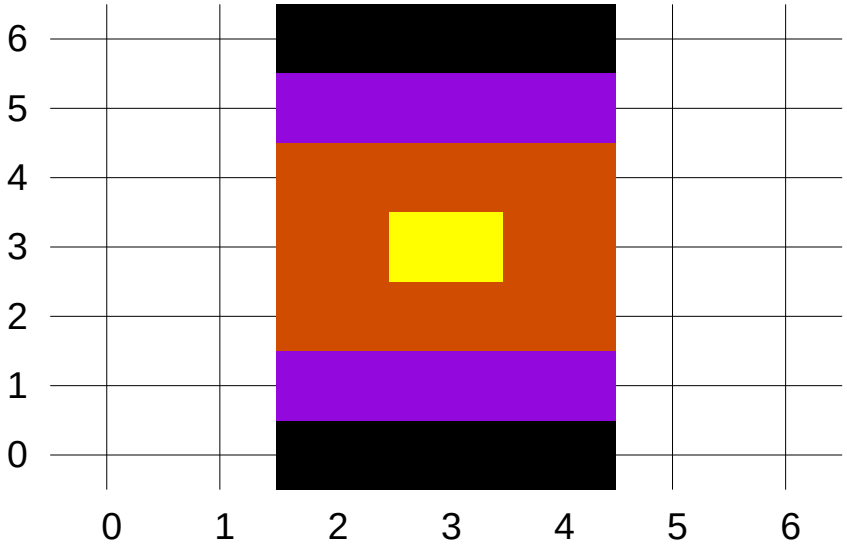
Full 7x7 matrix



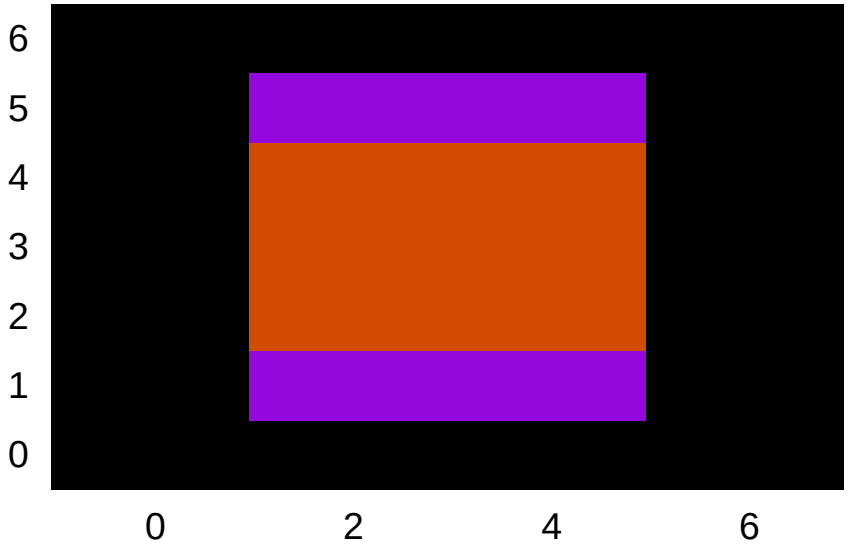
Subsample rows by every :::2::4

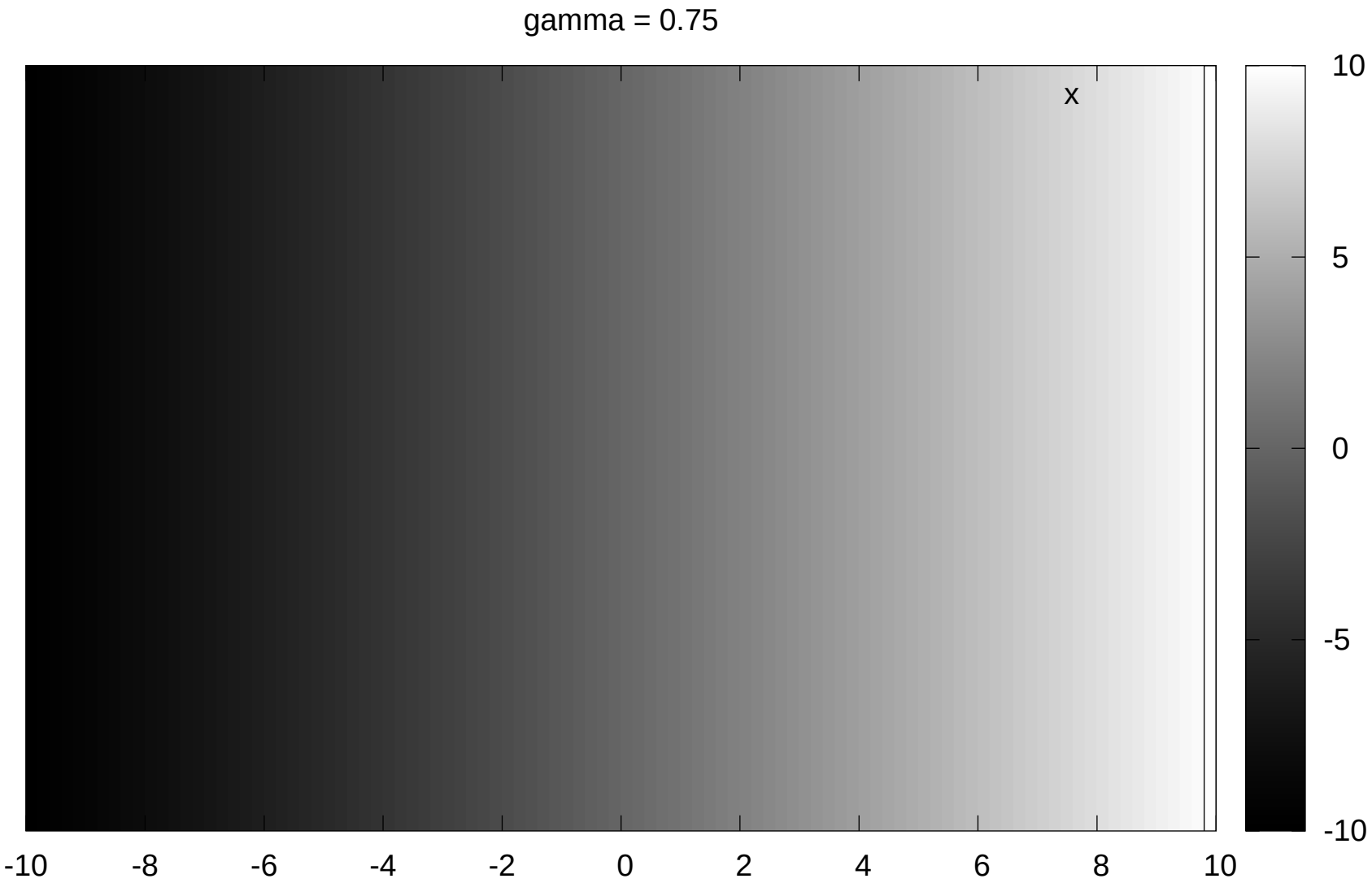


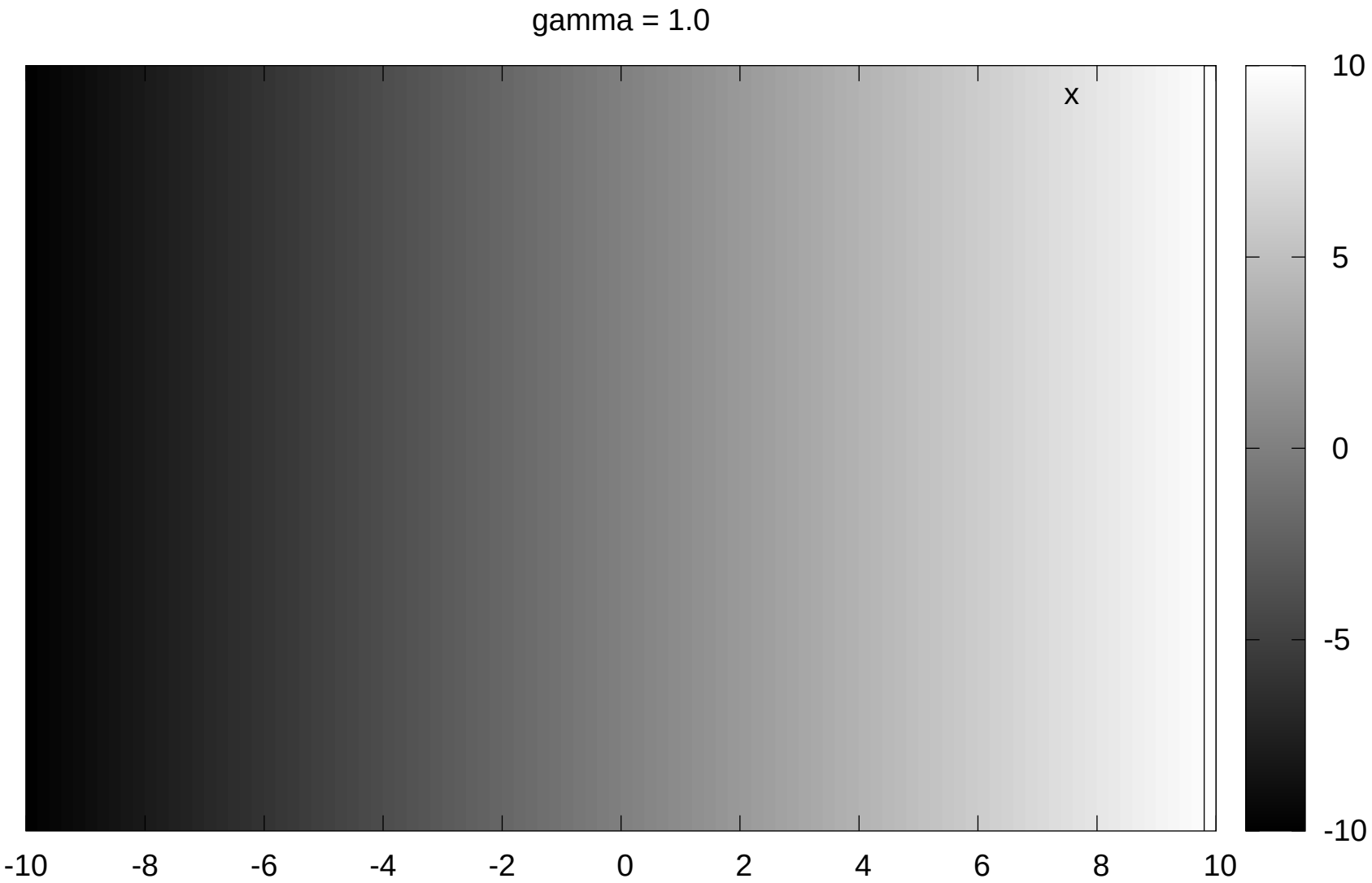
Subsample columns by every ::2::4

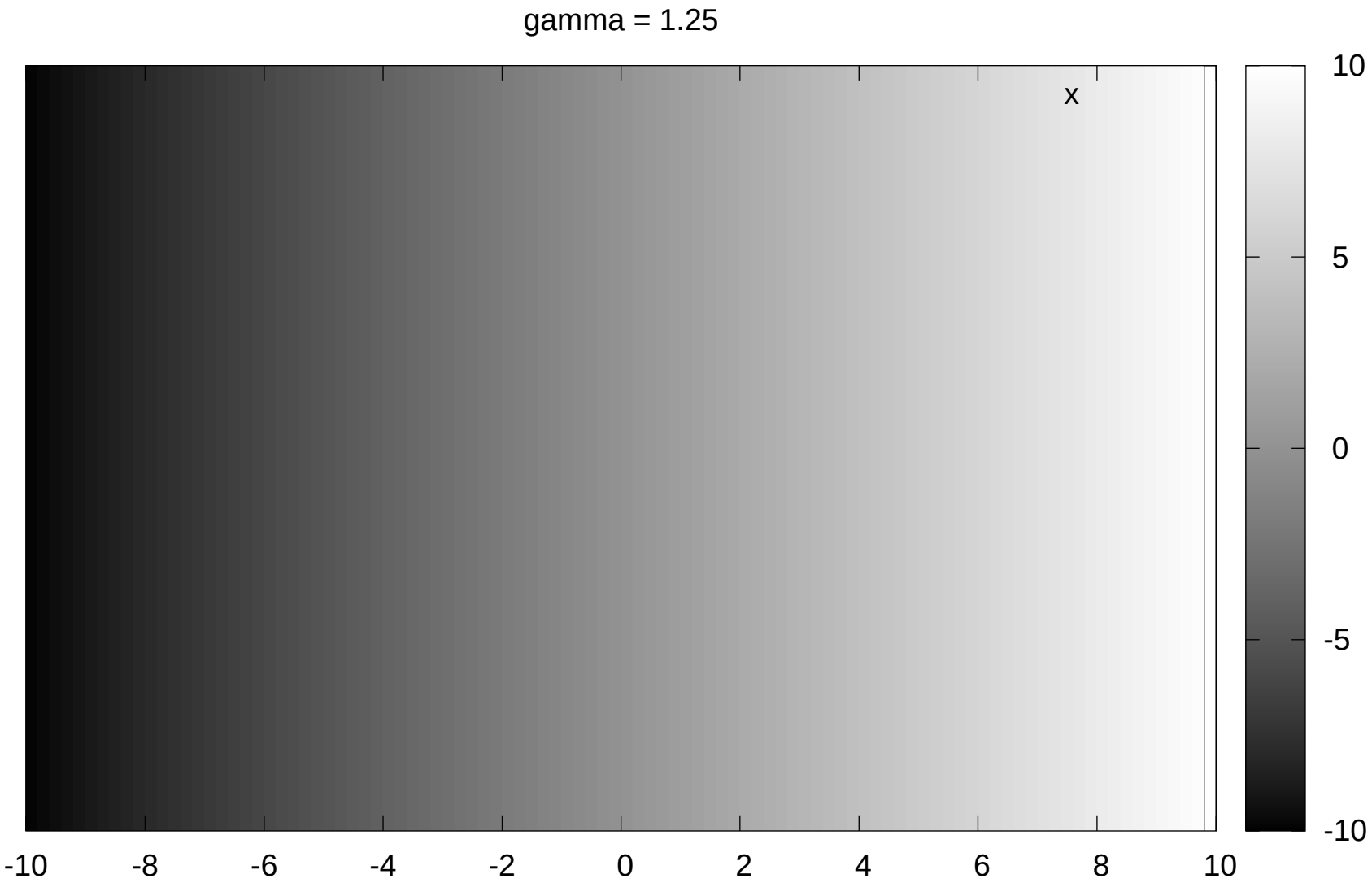


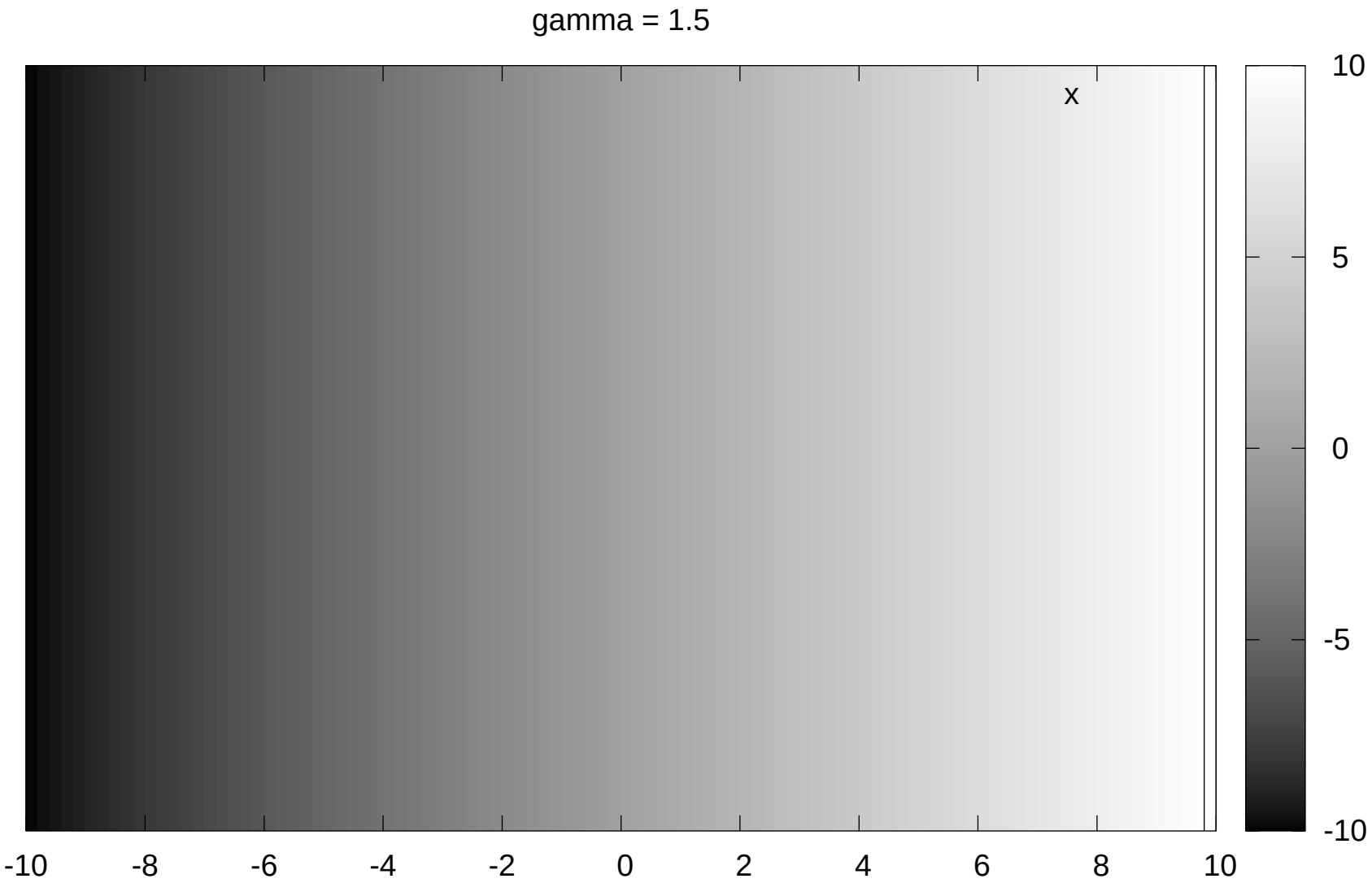
Sample alternate columns by every 2

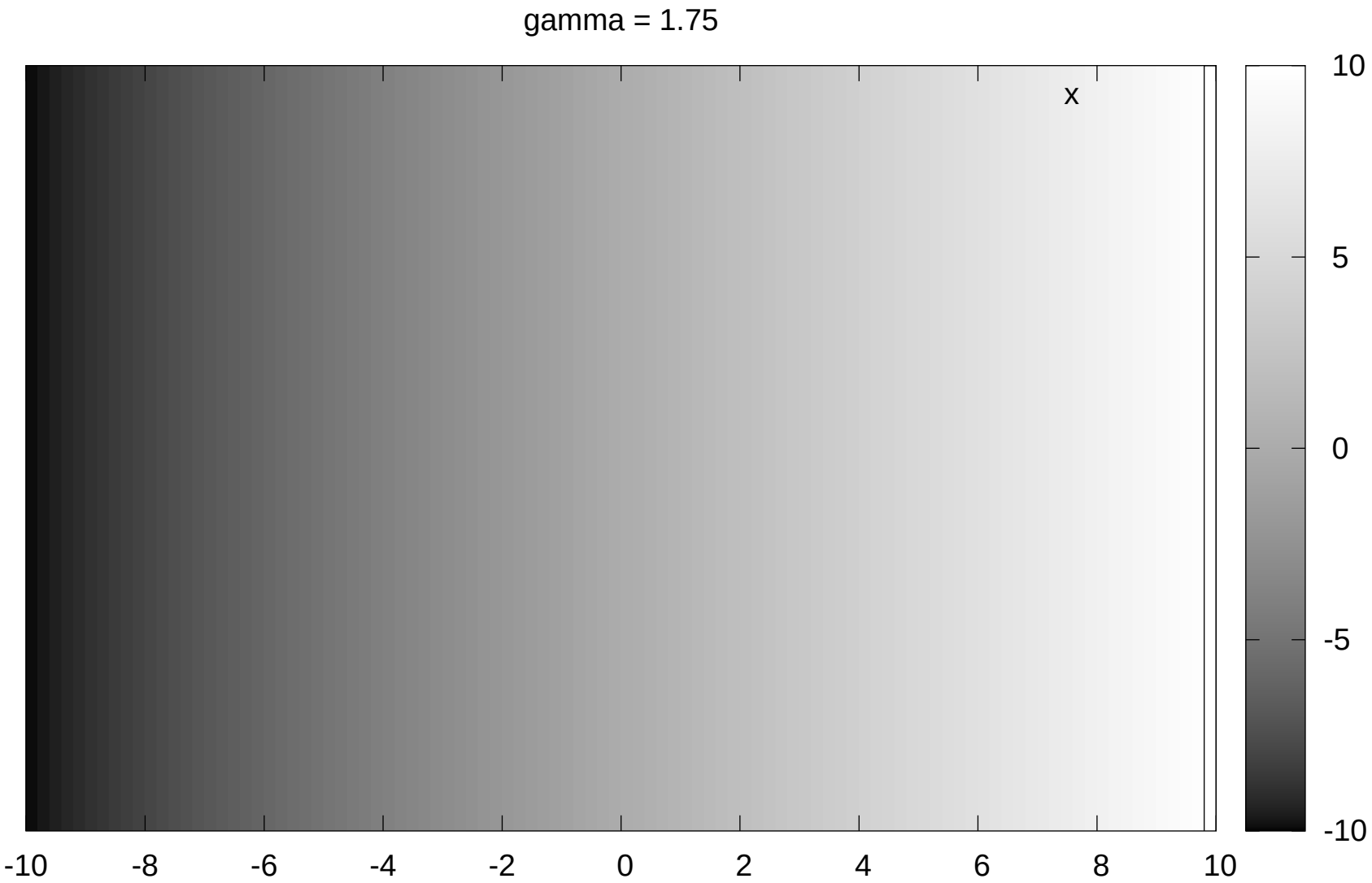


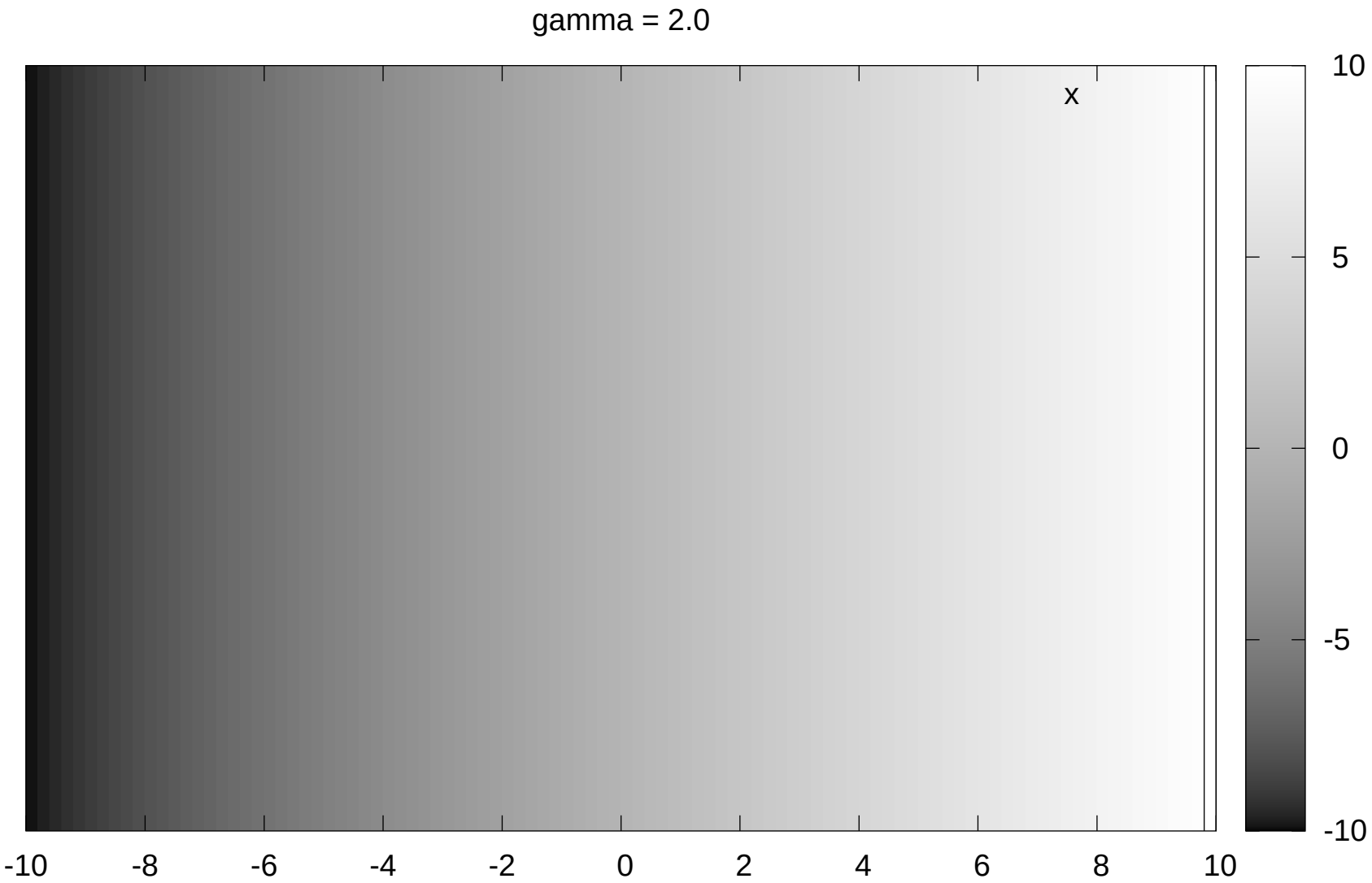

gamma = 0.75
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10


gamma = 1.0
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

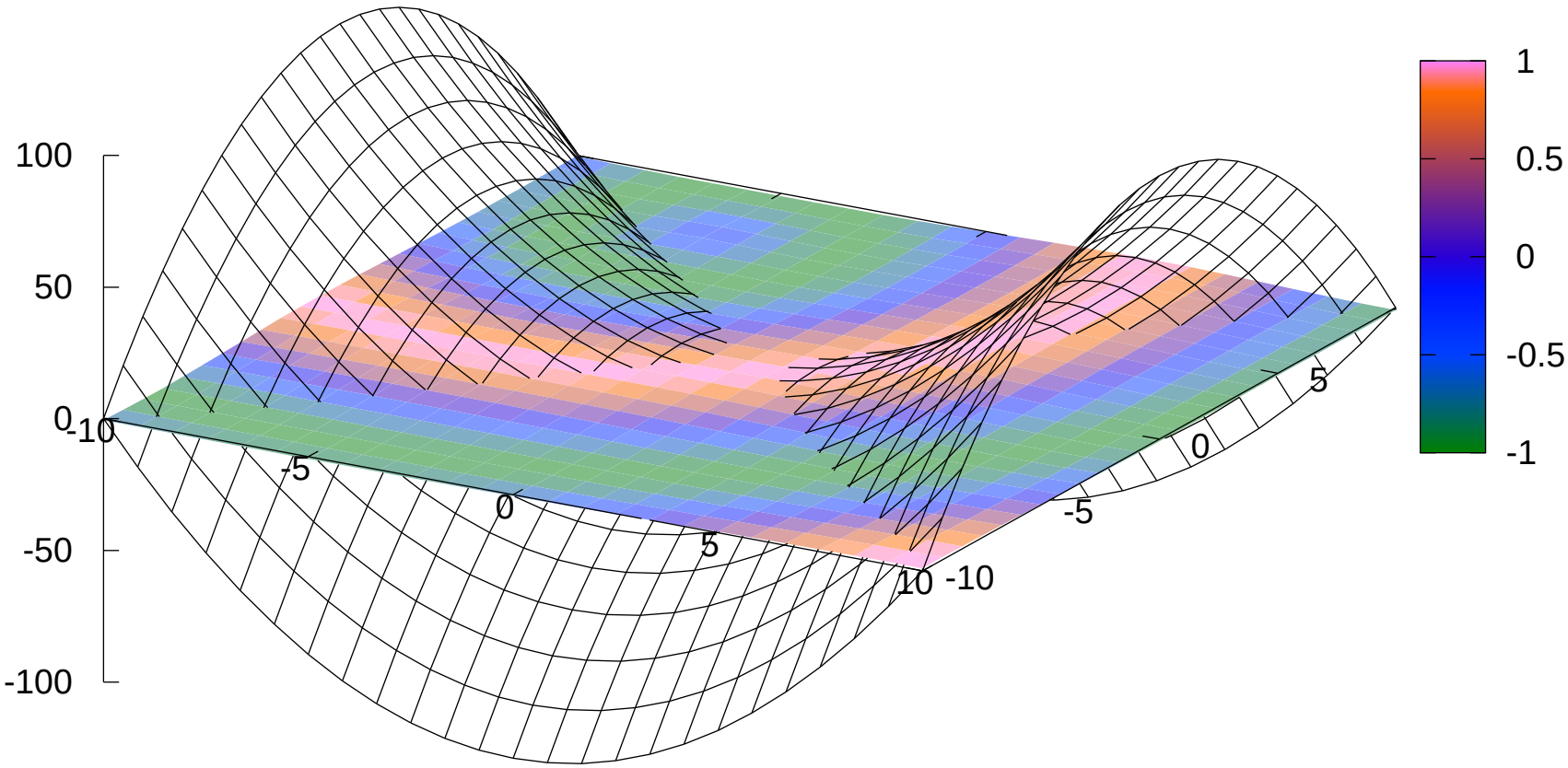

gamma = 1.25
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10


gamma = 1.5
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

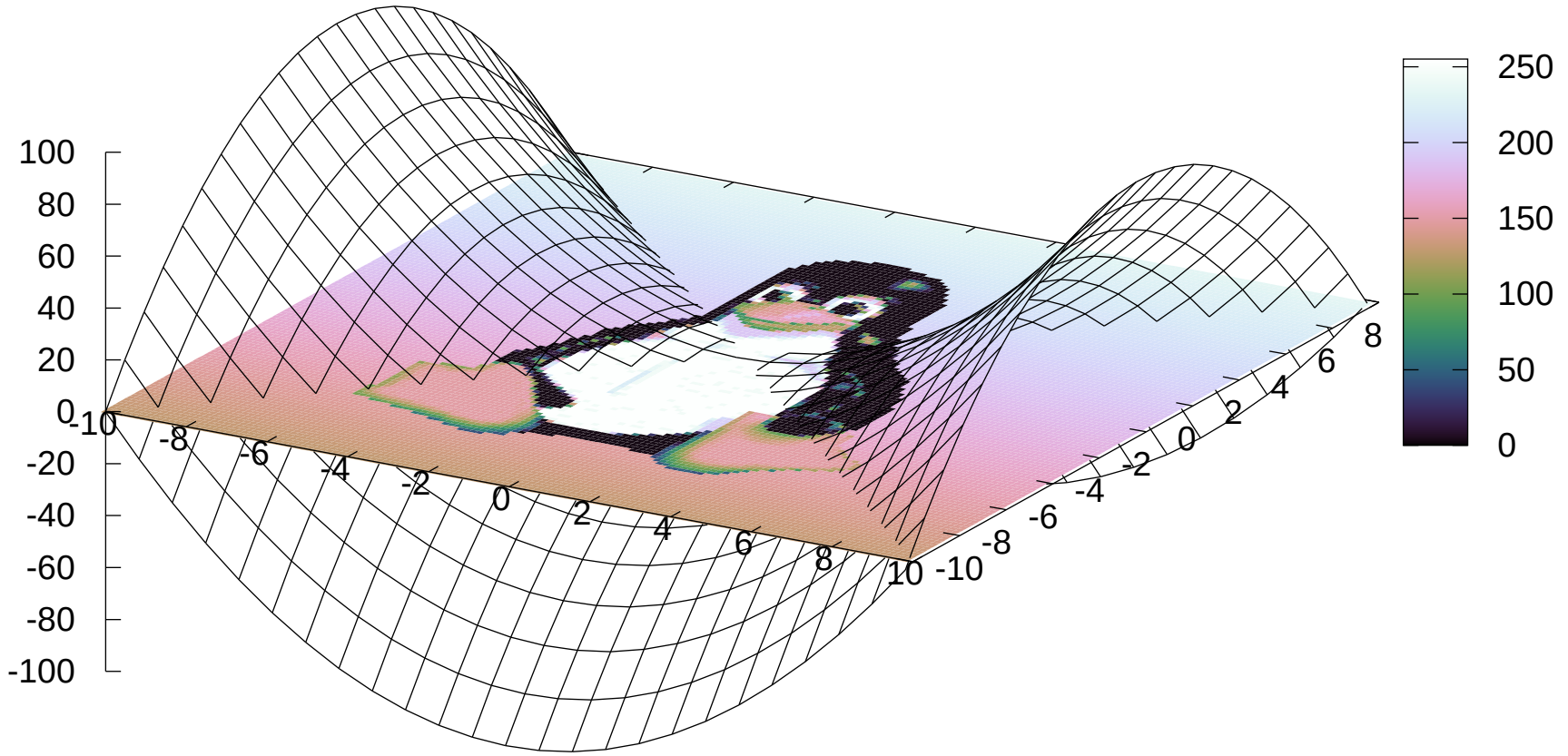

gamma = 1.75
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10


gamma = 2.0
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

Mixing pm3d surfaces with hidden-line plots



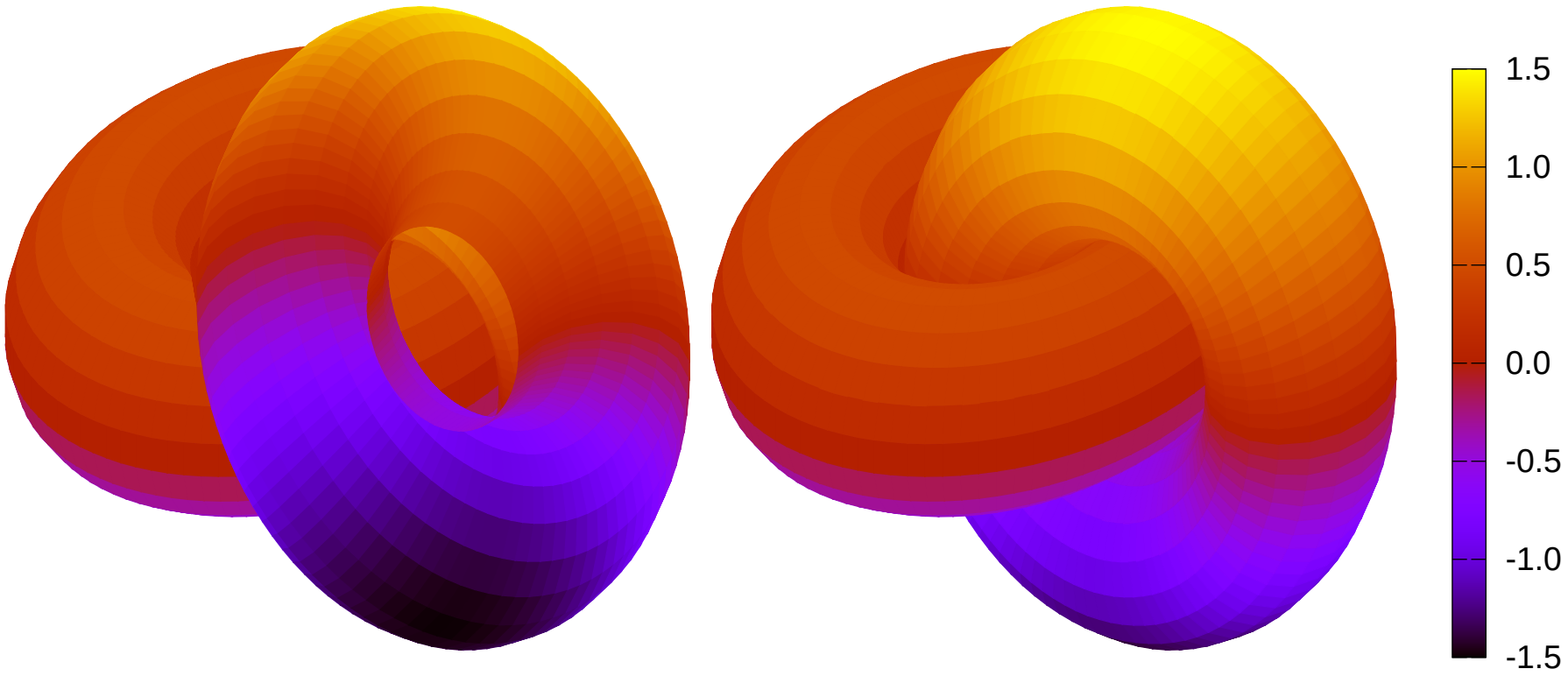
Mixing image surface with hidden-line plots

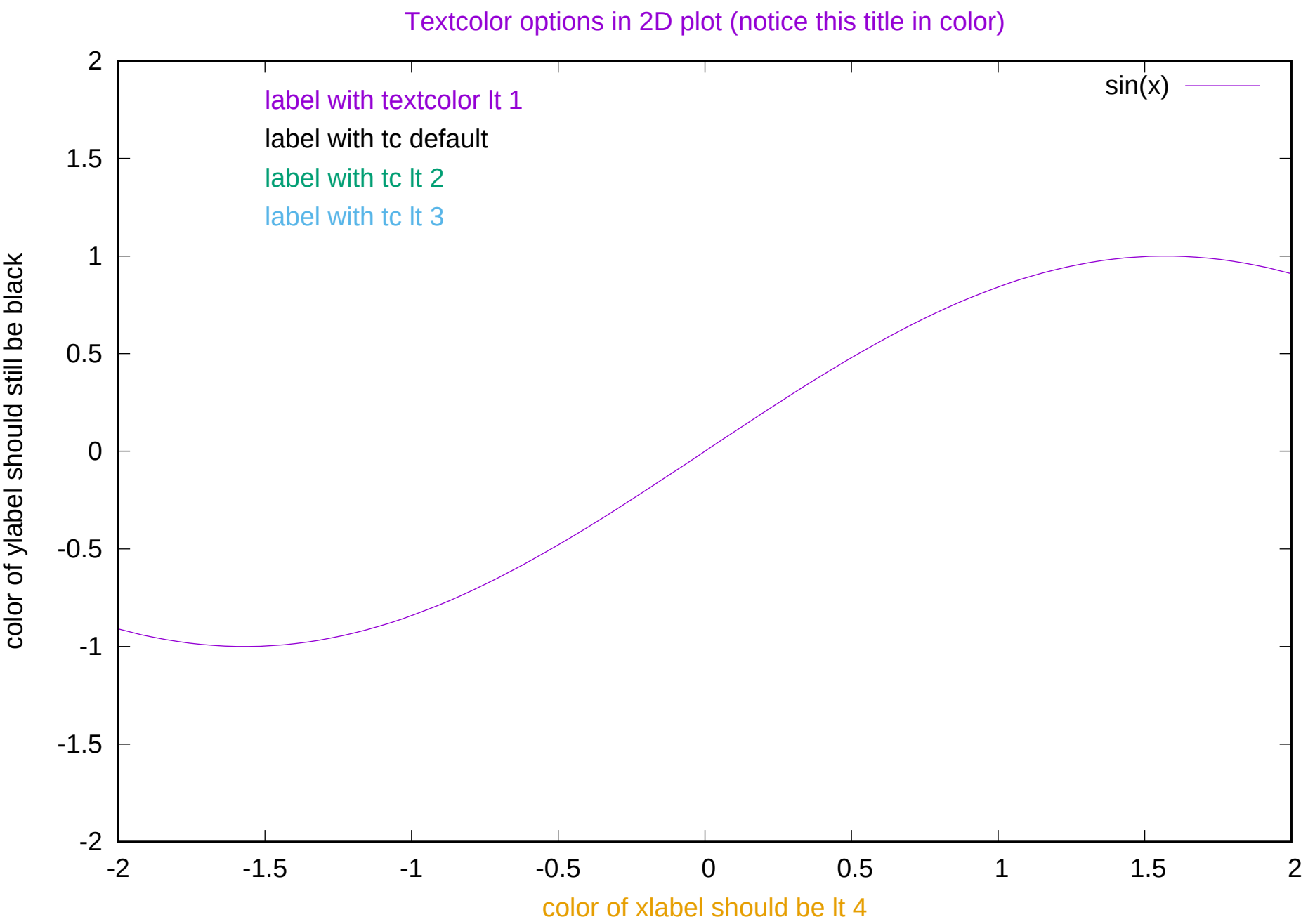


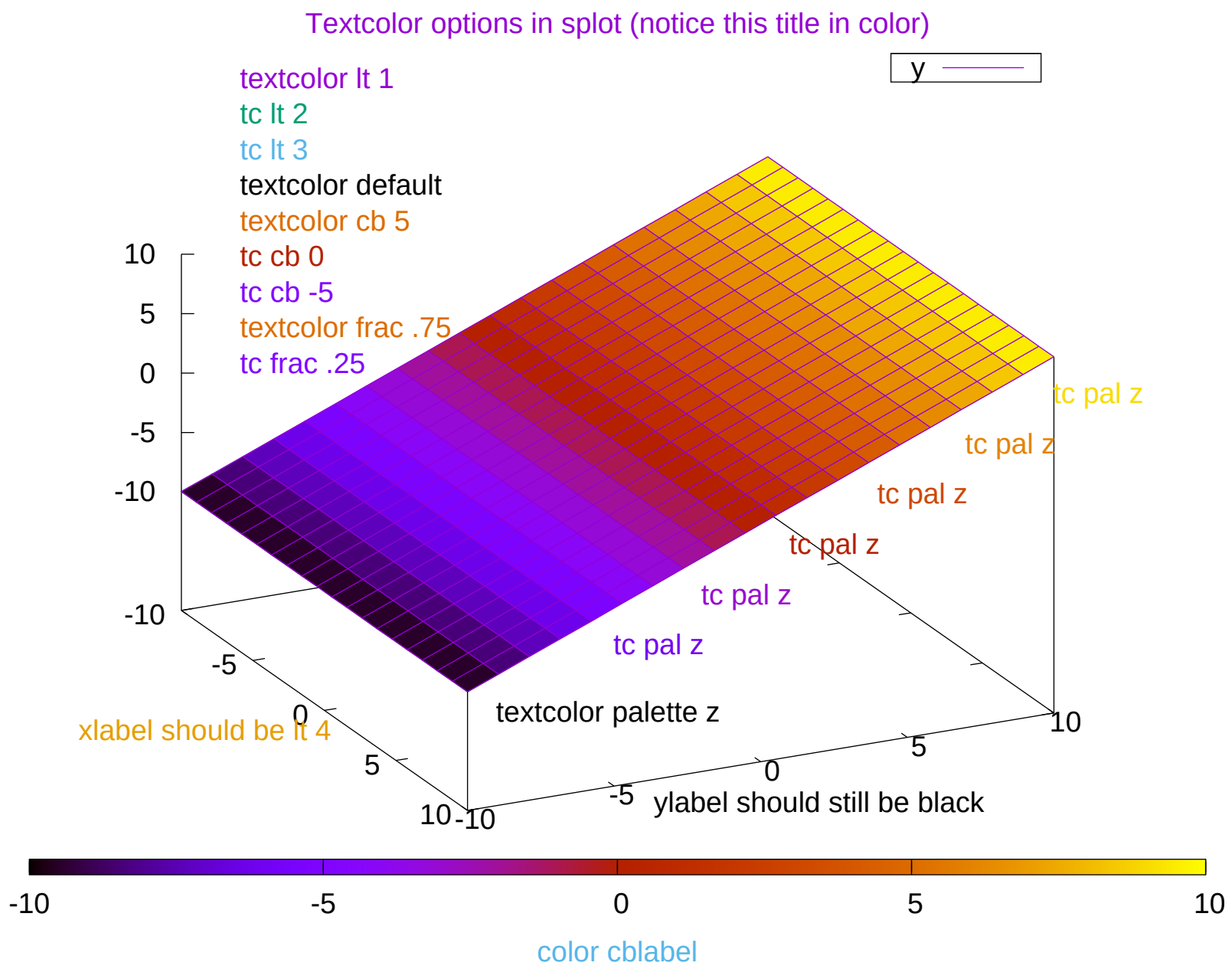
Interlocking Tori

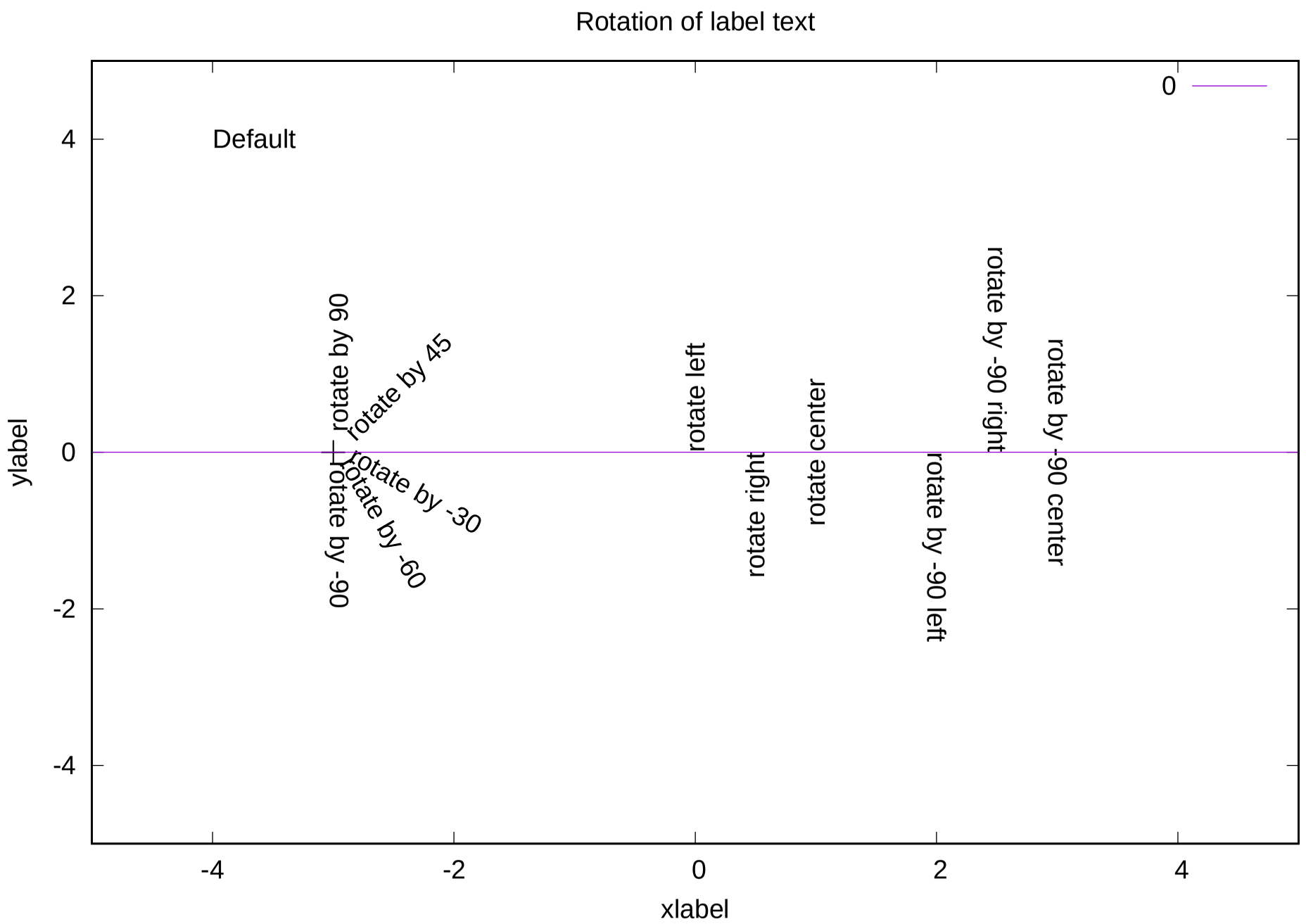
PM3D surface
no depth sorting

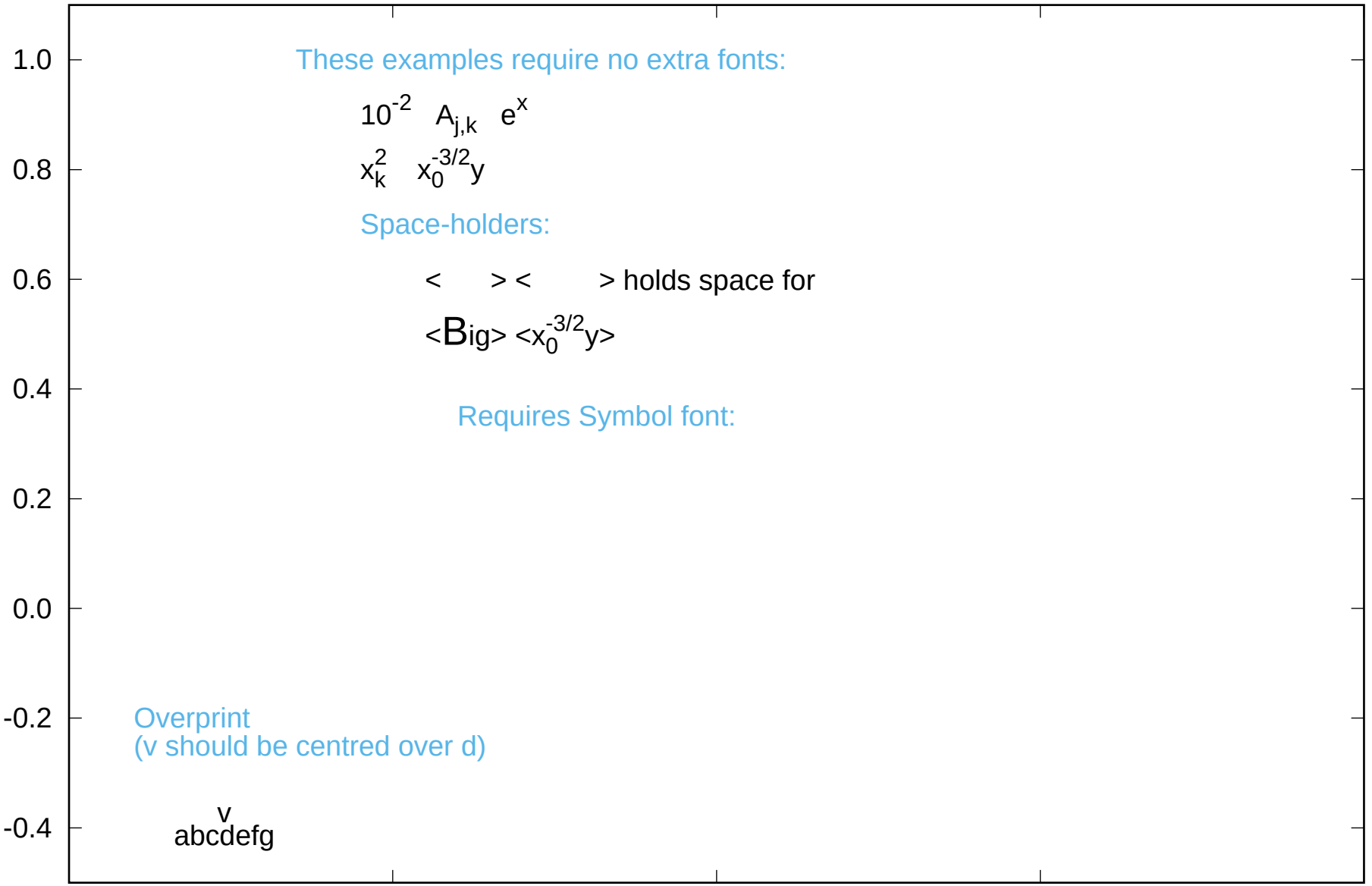
PM3D surface
depth sorting

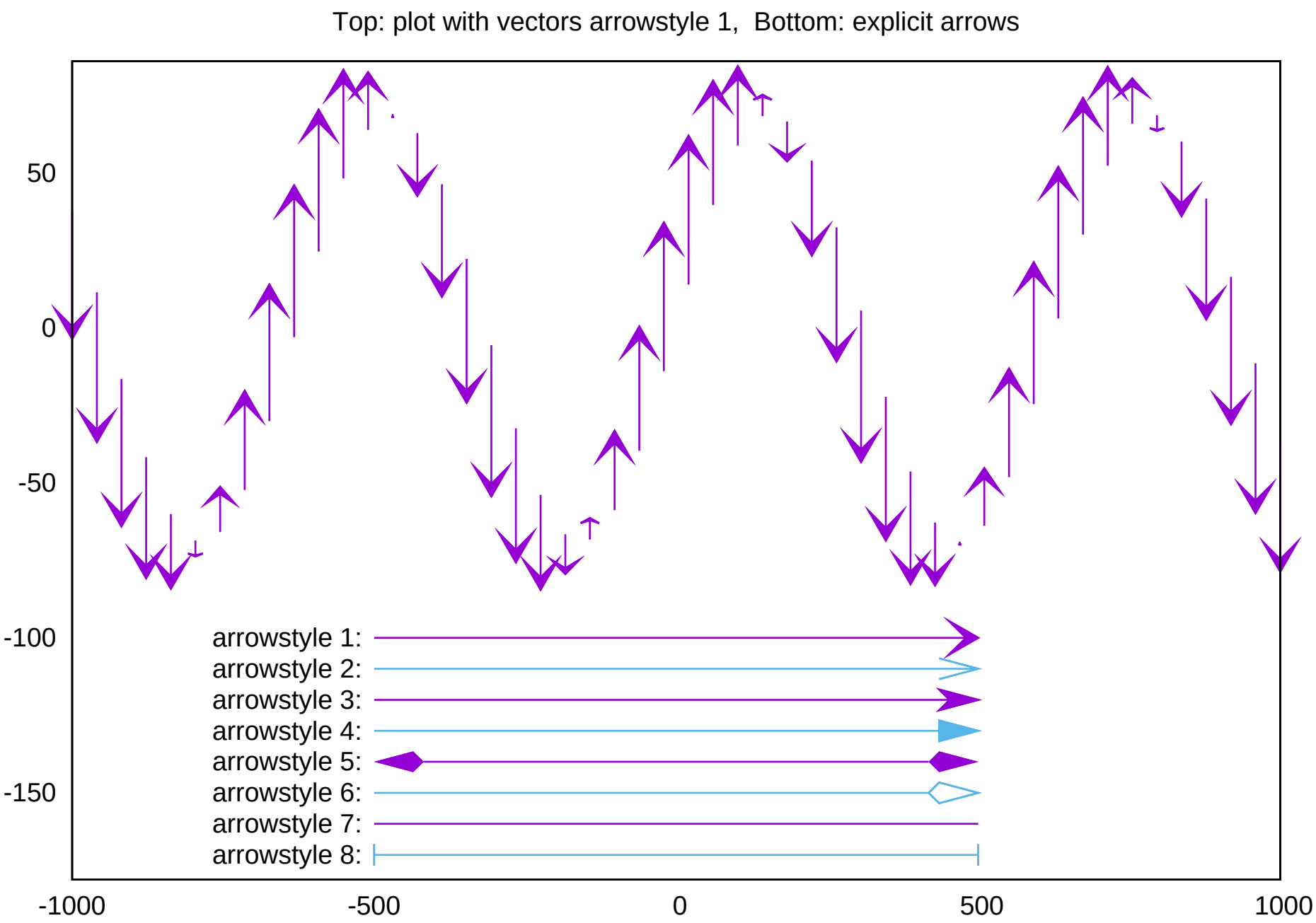


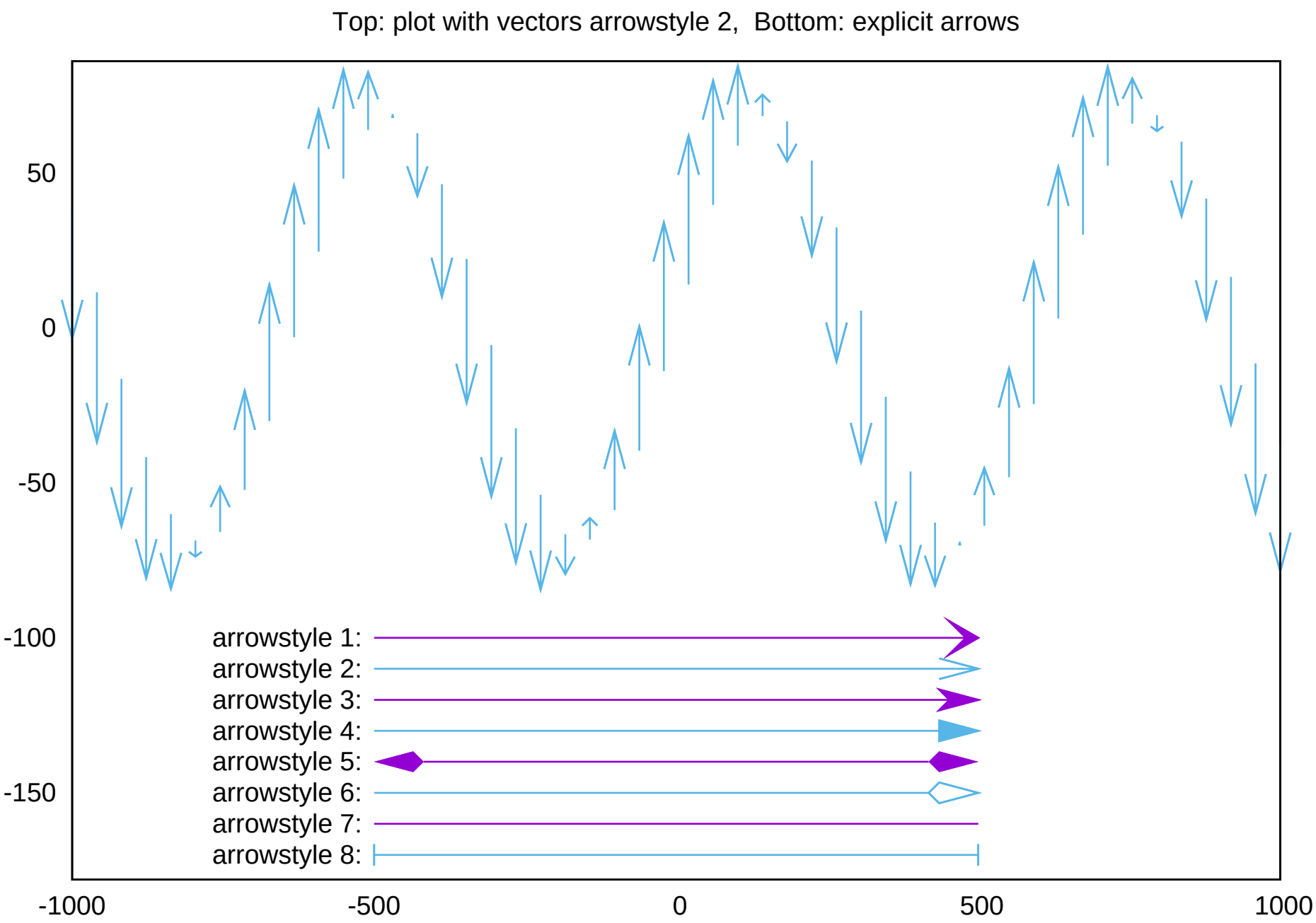

Textcolor options in 2D plot (notice this title in color)
label with textcolor lt 1
label with tc default
label with tc lt 2
label with tc lt 3
sin(x)
color of ylabel should still be black
color of xlabel should be lt 4
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

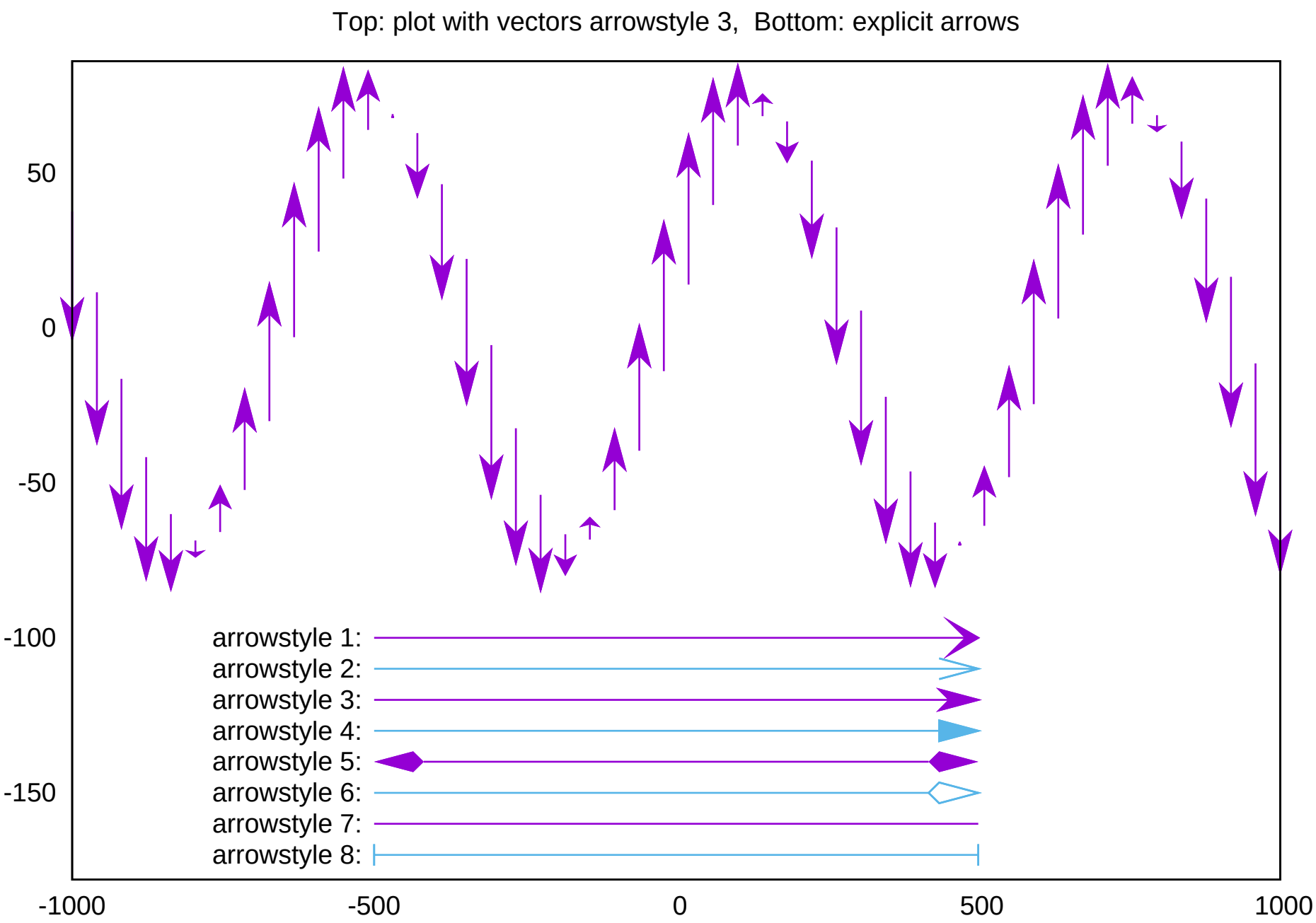

Textcolor options in splot (notice this title in color)
y
textcolor lt 1
tc lt 2
tc lt 3
textcolor default
textcolor cb 5
tc cb 0
tc cb -5
textcolor frac .75
tc frac .25
tc pal z
tc pal z
tc pal z
tc pal z
tc pal z
tc pal z
textcolor palette z
xlabel should be lt 4
ylabel should still be black
10
5
0
-5
-10
-10
-5
0
5
10
-10
-5
0
5
10
color cblabel

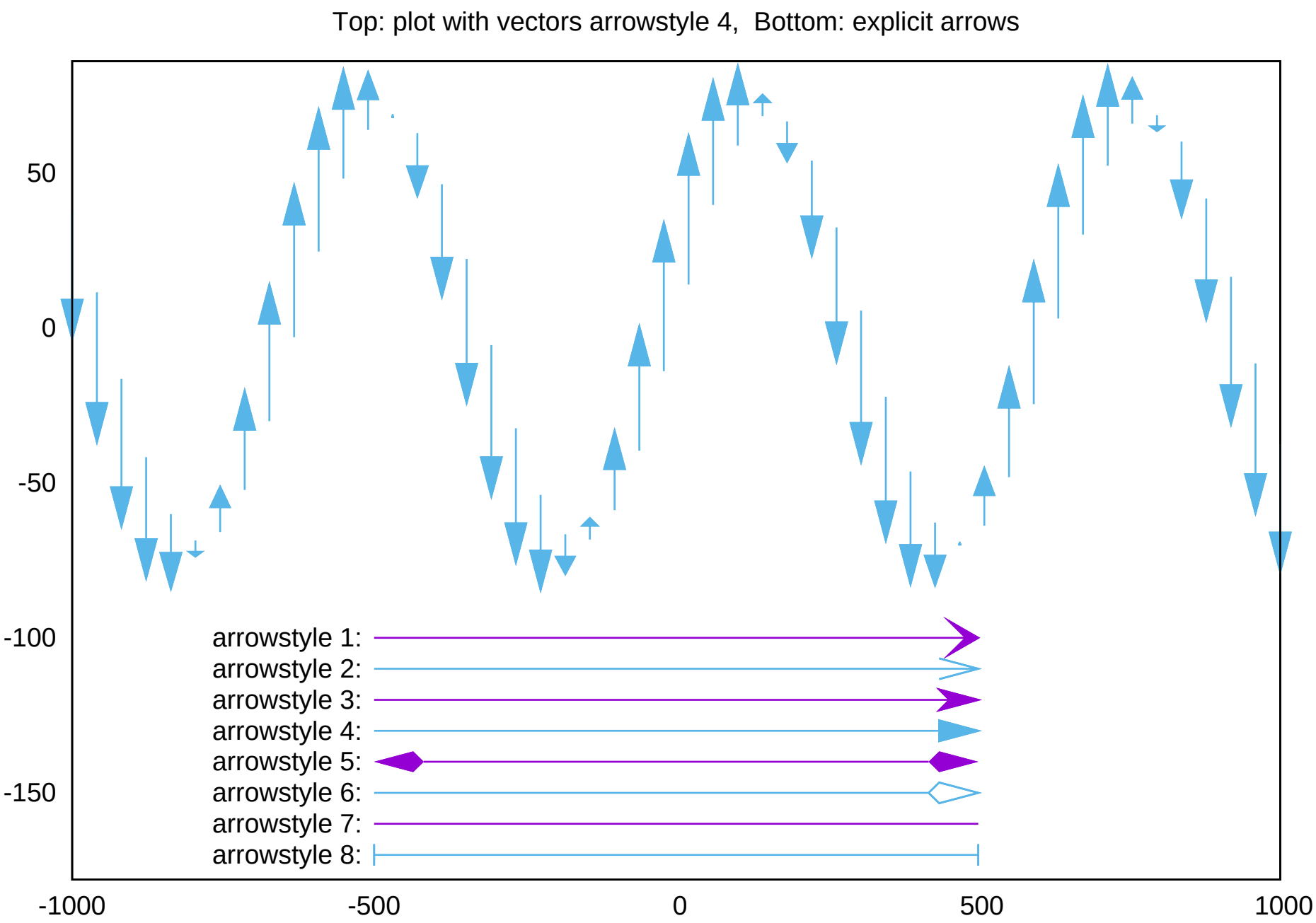

Rotation of label text
Default
0
rotate by 90
rotate by 45
rotate by -30
rotate by -60
rotate by -90
rotate left
rotate right
rotate center
rotate by -90 left
rotate by -90 right
rotate by -90 center
ylabel
xlabel
4
2
0
-2
-4
-4
-2
0
2
4

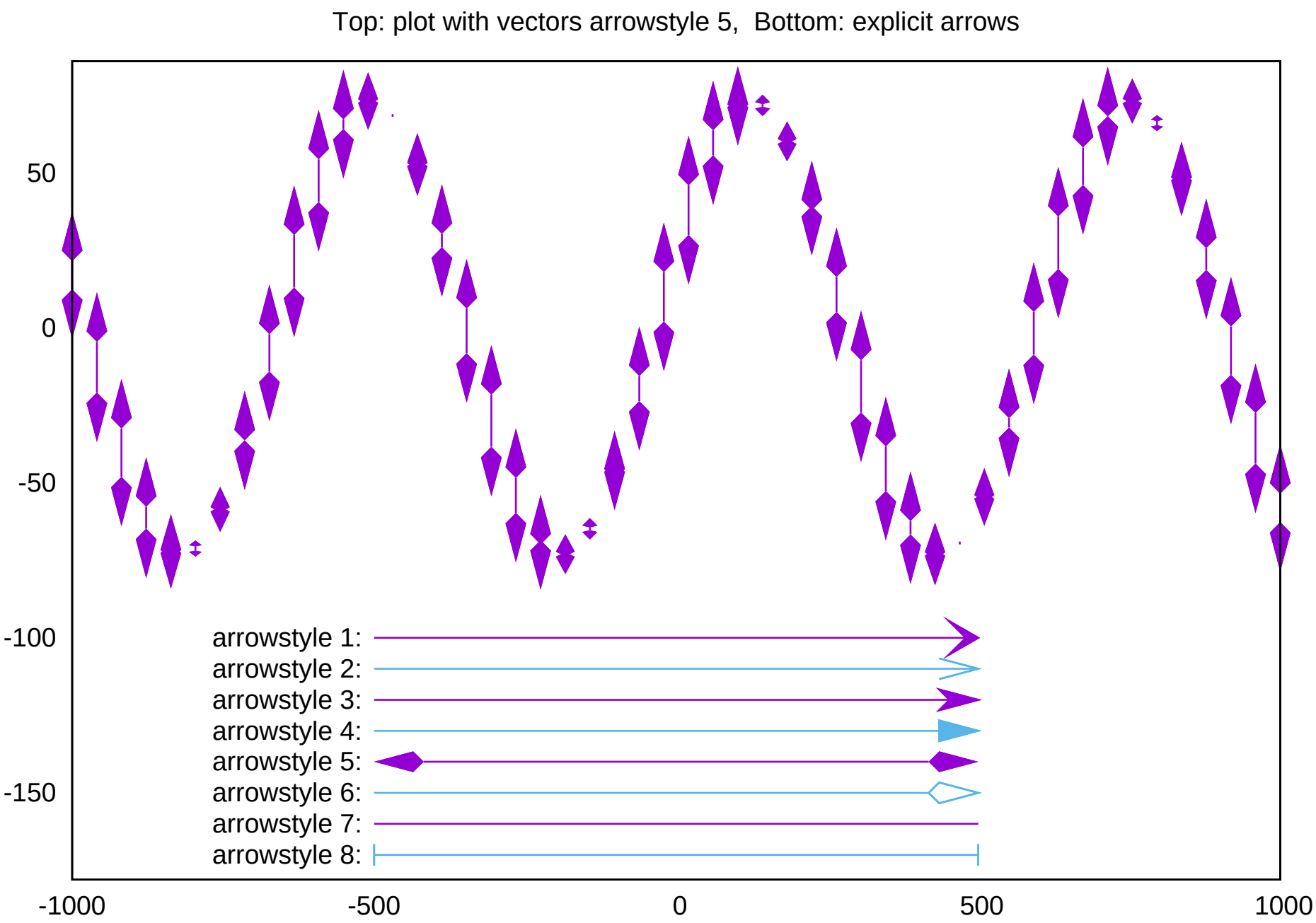

These examples require no extra fonts:
10^{-2} $A_{j,k}$ e^{x}
x_k^2 $x_0^{-3/2}y$
Space-holders:
< > < > holds space for
<Big> $<x_0^{-3/2}y>$
Requires Symbol font:
Overprint
(v should be centred over d)
v
abcdefg
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-1.0
-0.5
0.0
0.5
1.0

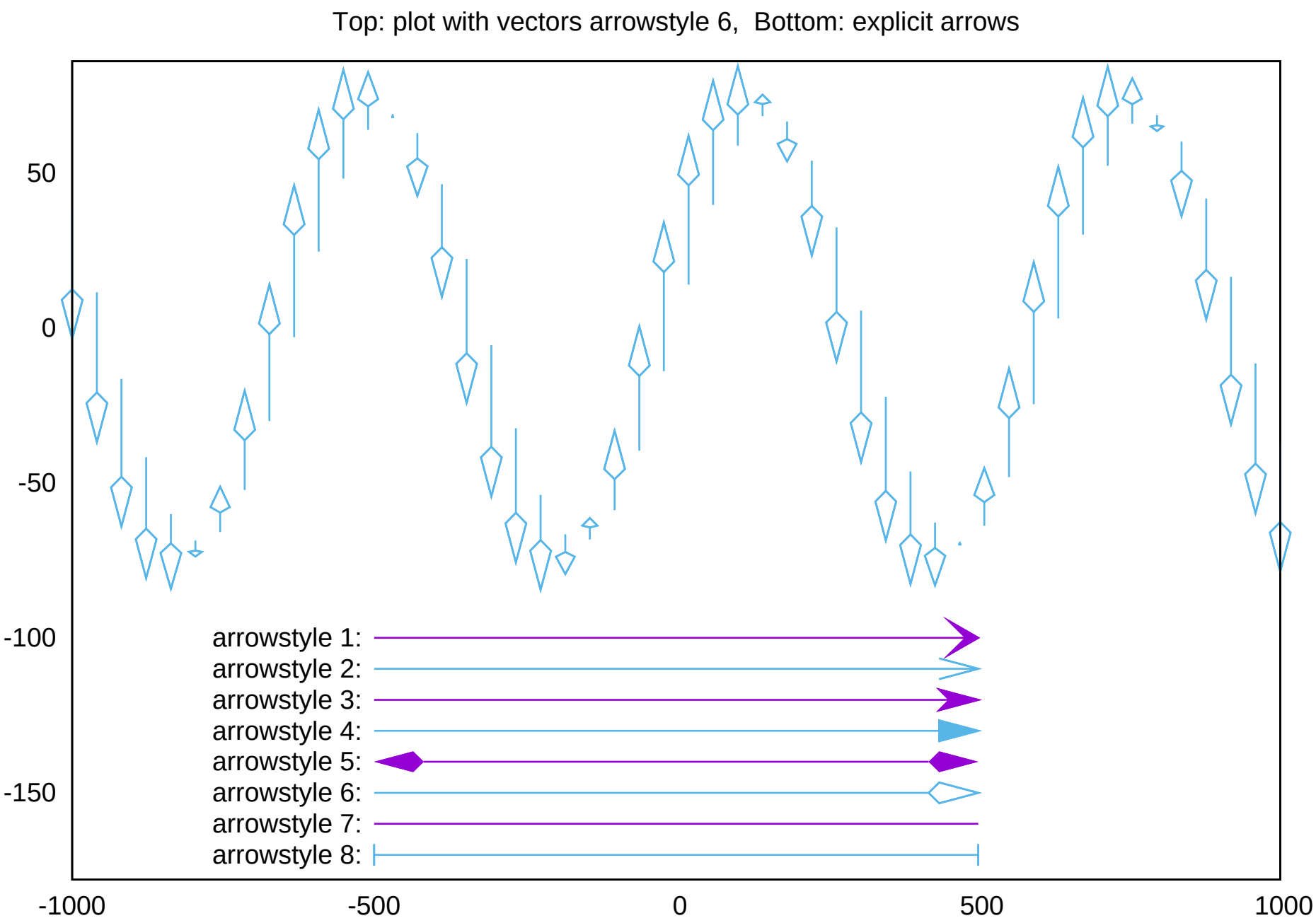

Top: plot with vectors arrowstyle 1, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:

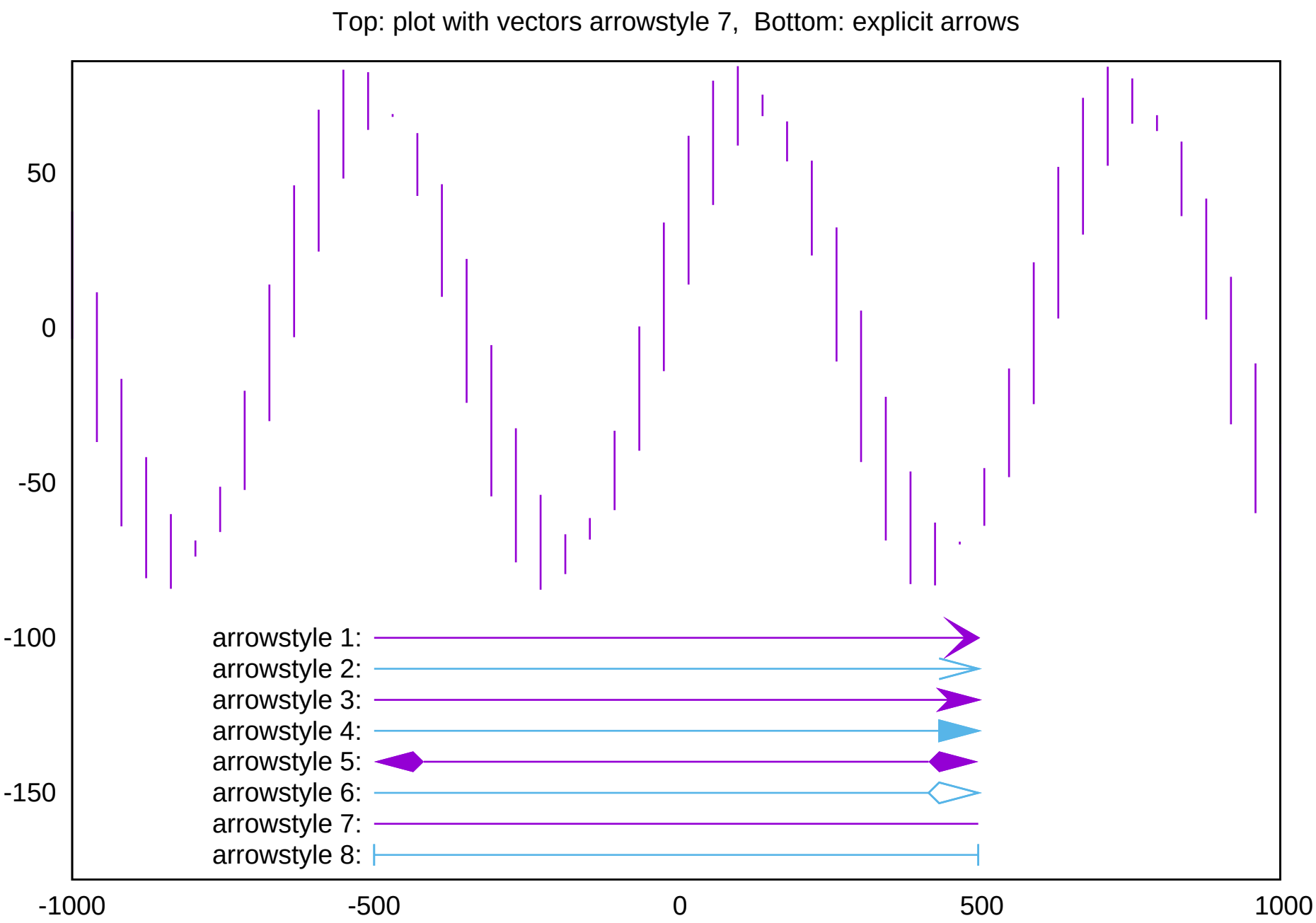

Top: plot with vectors arrowstyle 2, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:

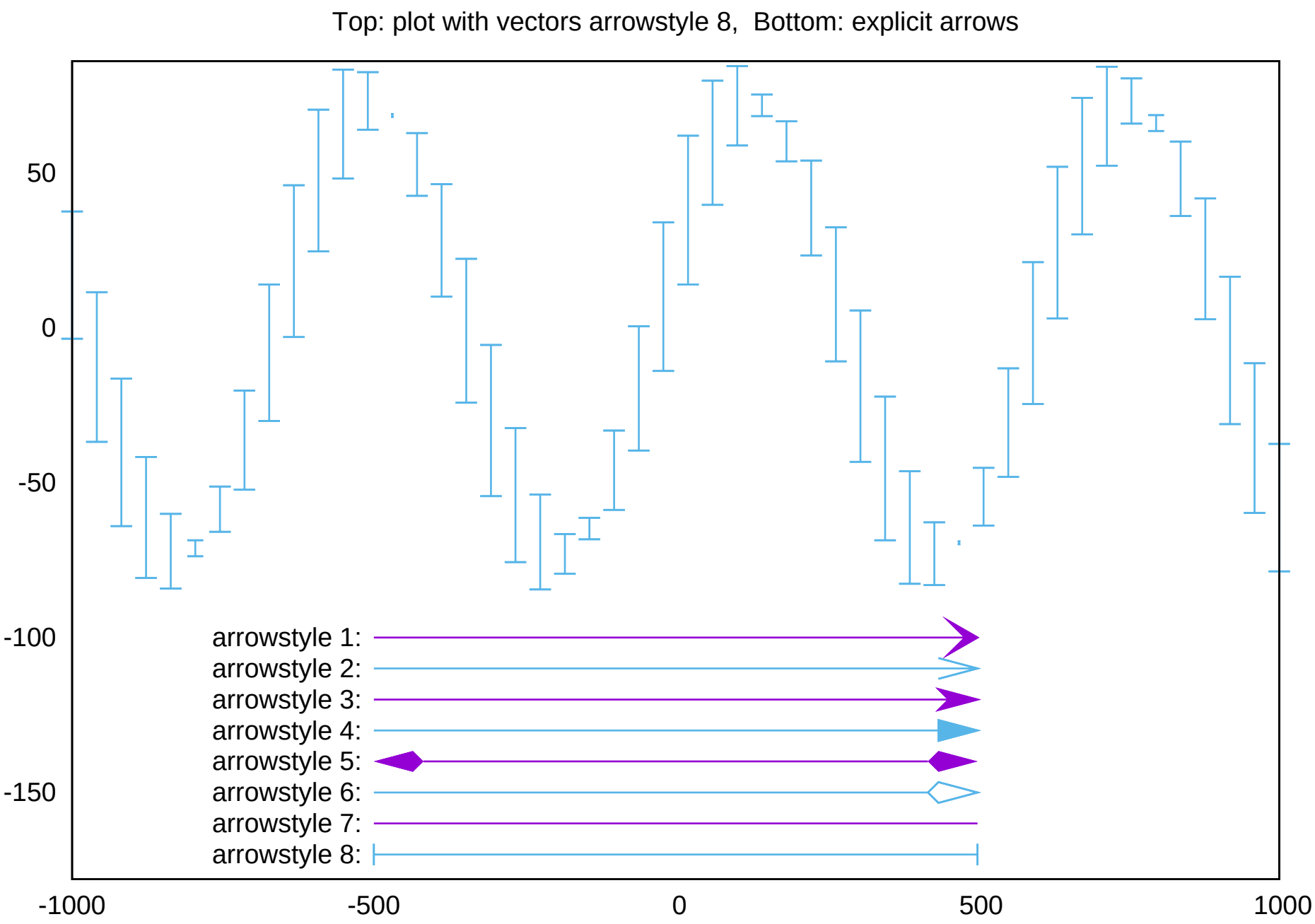

Top: plot with vectors arrowstyle 3, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:

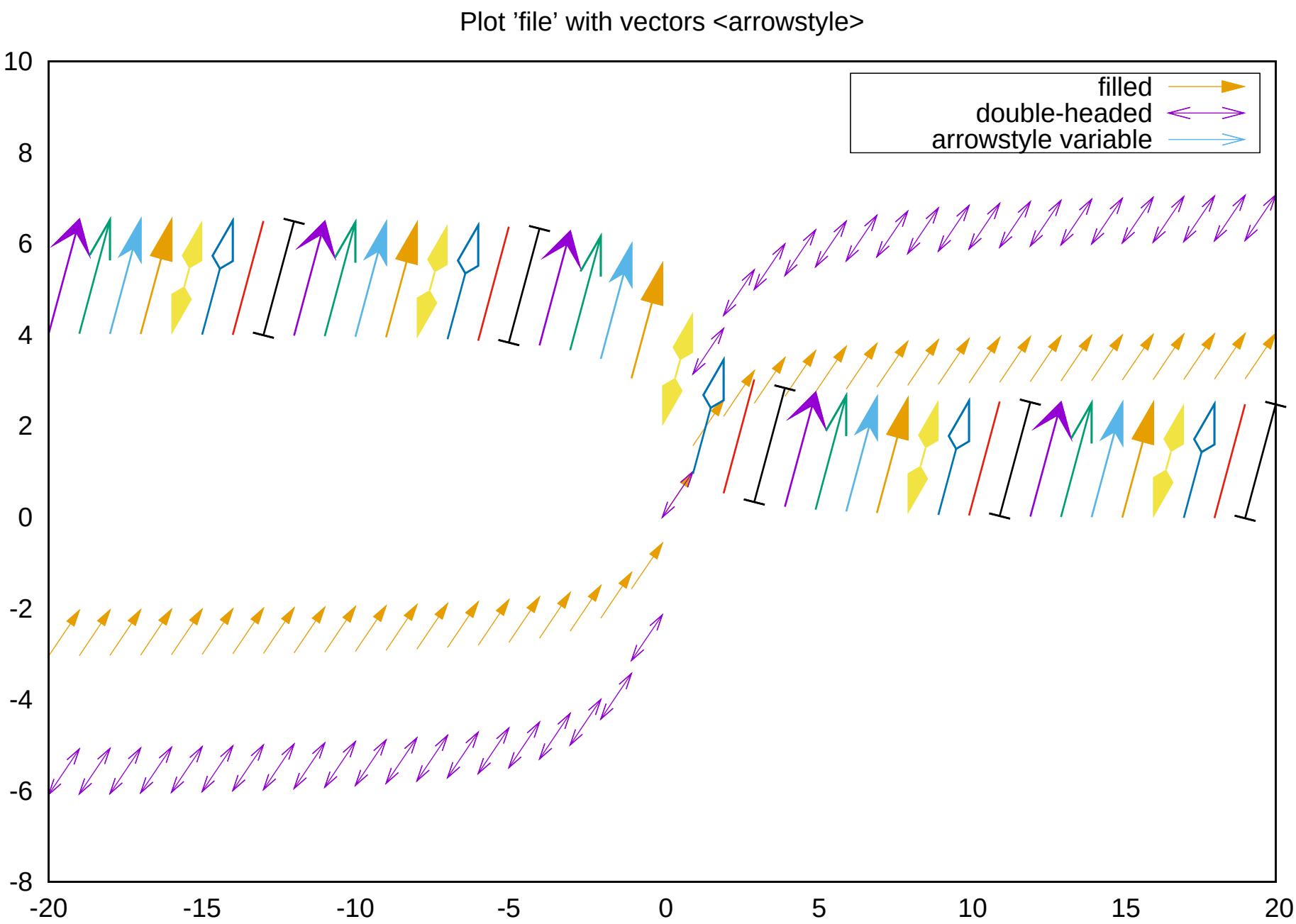

Top: plot with vectors arrowstyle 4, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:


Top: plot with vectors arrowstyle 5, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:


Top: plot with vectors arrowstyle 6, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:


Top: plot with vectors arrowstyle 7, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:


Top: plot with vectors arrowstyle 8, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:


Plot 'file' with vectors <arrowstyle>
filled
double-headed
arrowstyle variable
10
8
6
4
2
0
-2
-4
-6
-8
-20
-15
-10
-5
0
5
10
15
20

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 "-"

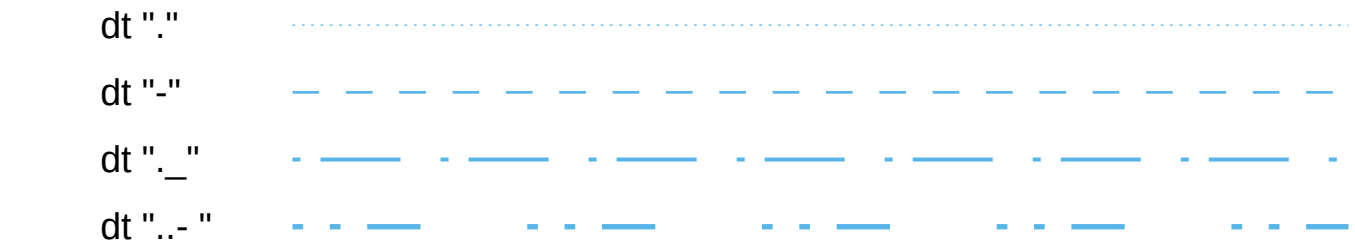
dt "._"

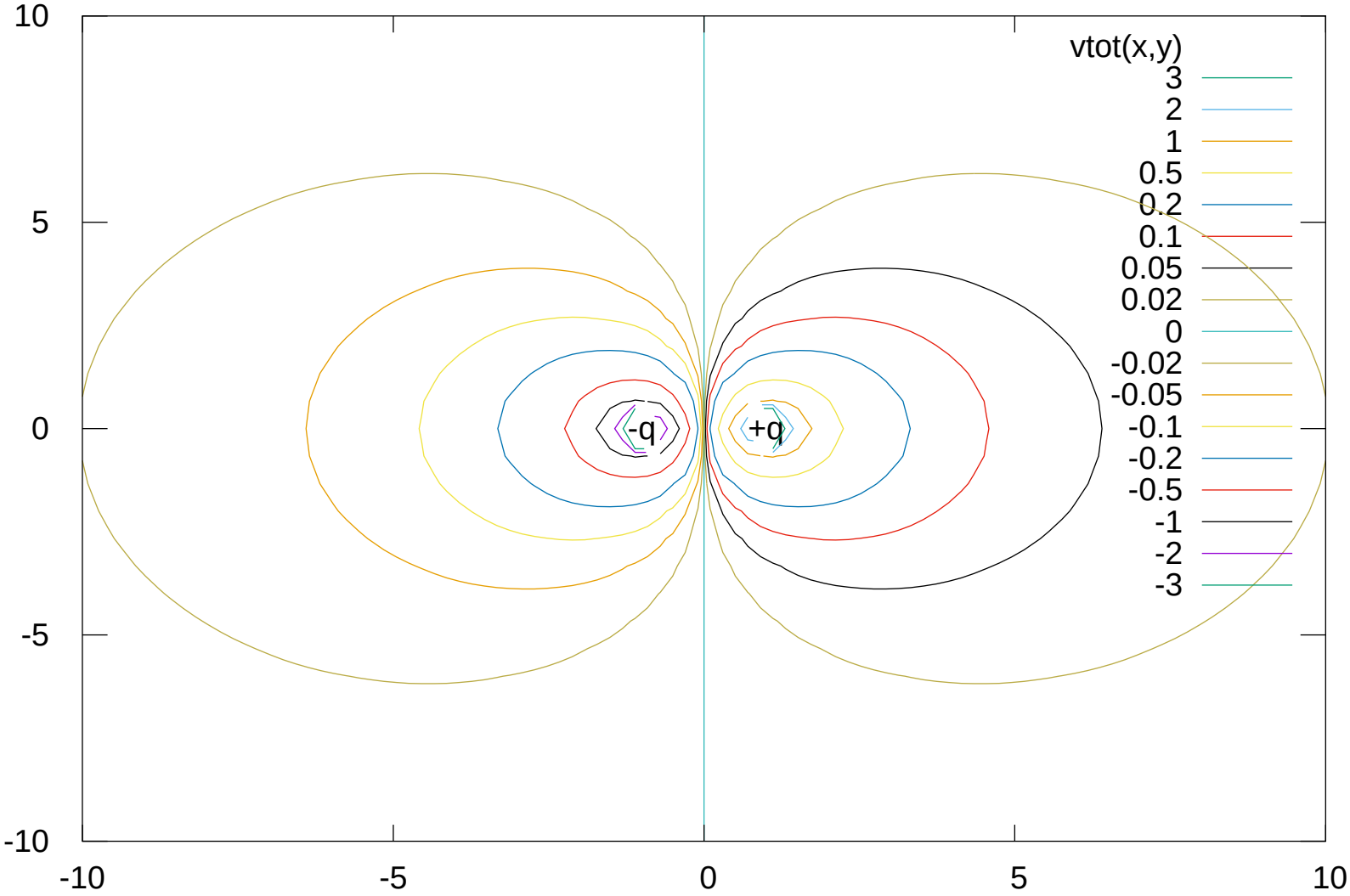
dt "..- "

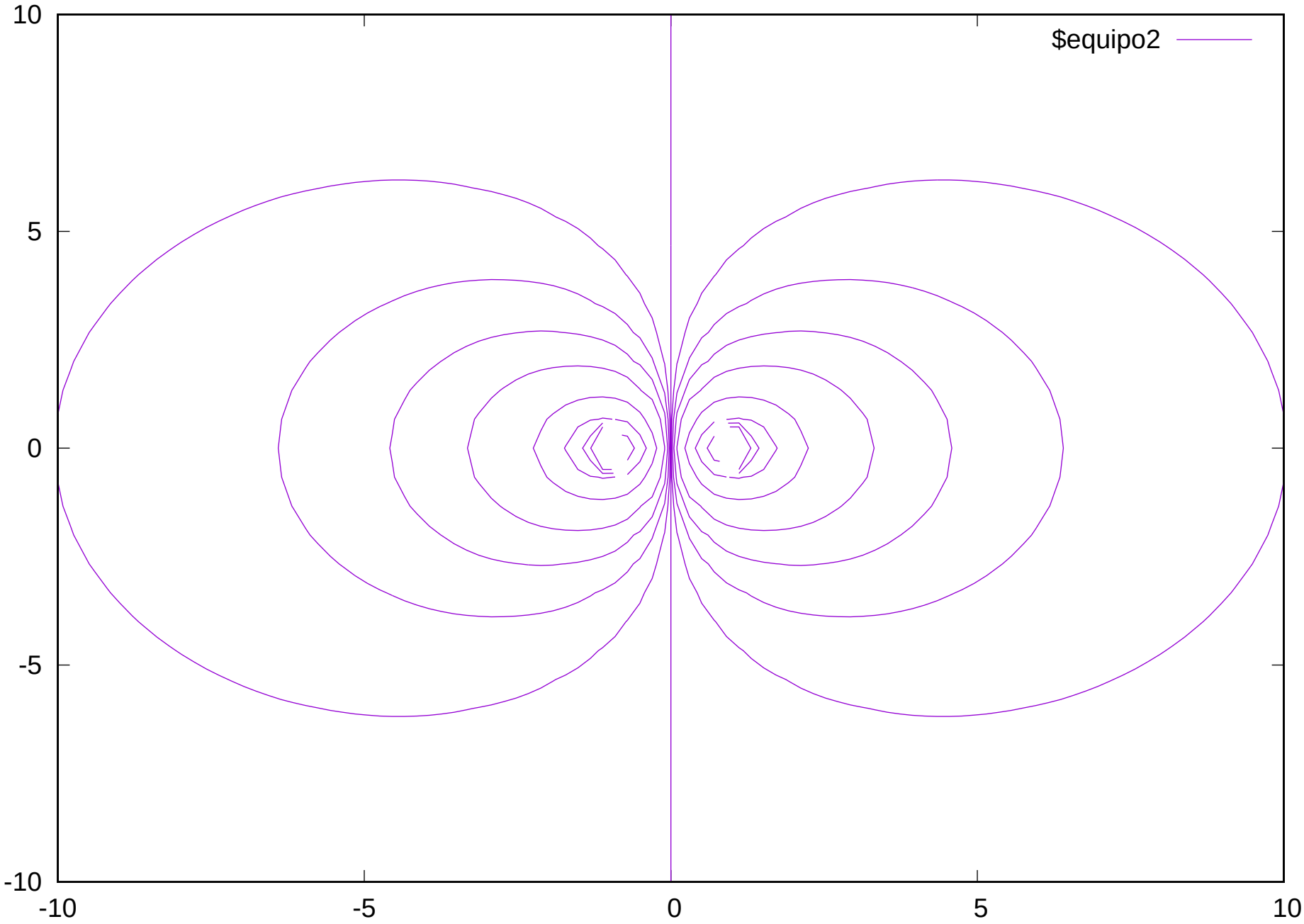
dt (50,6,2,6)

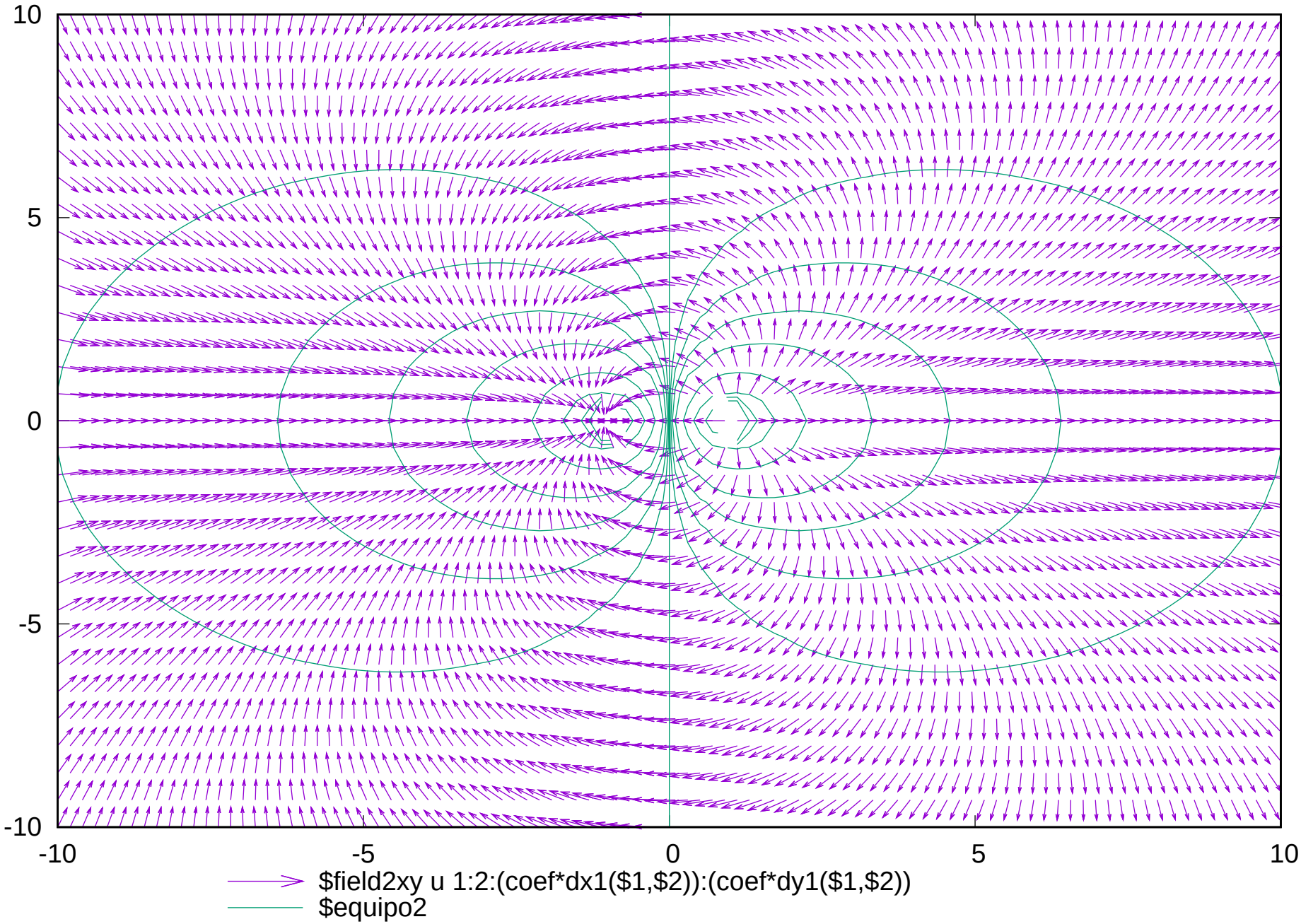
Terminal’s native dashtypes

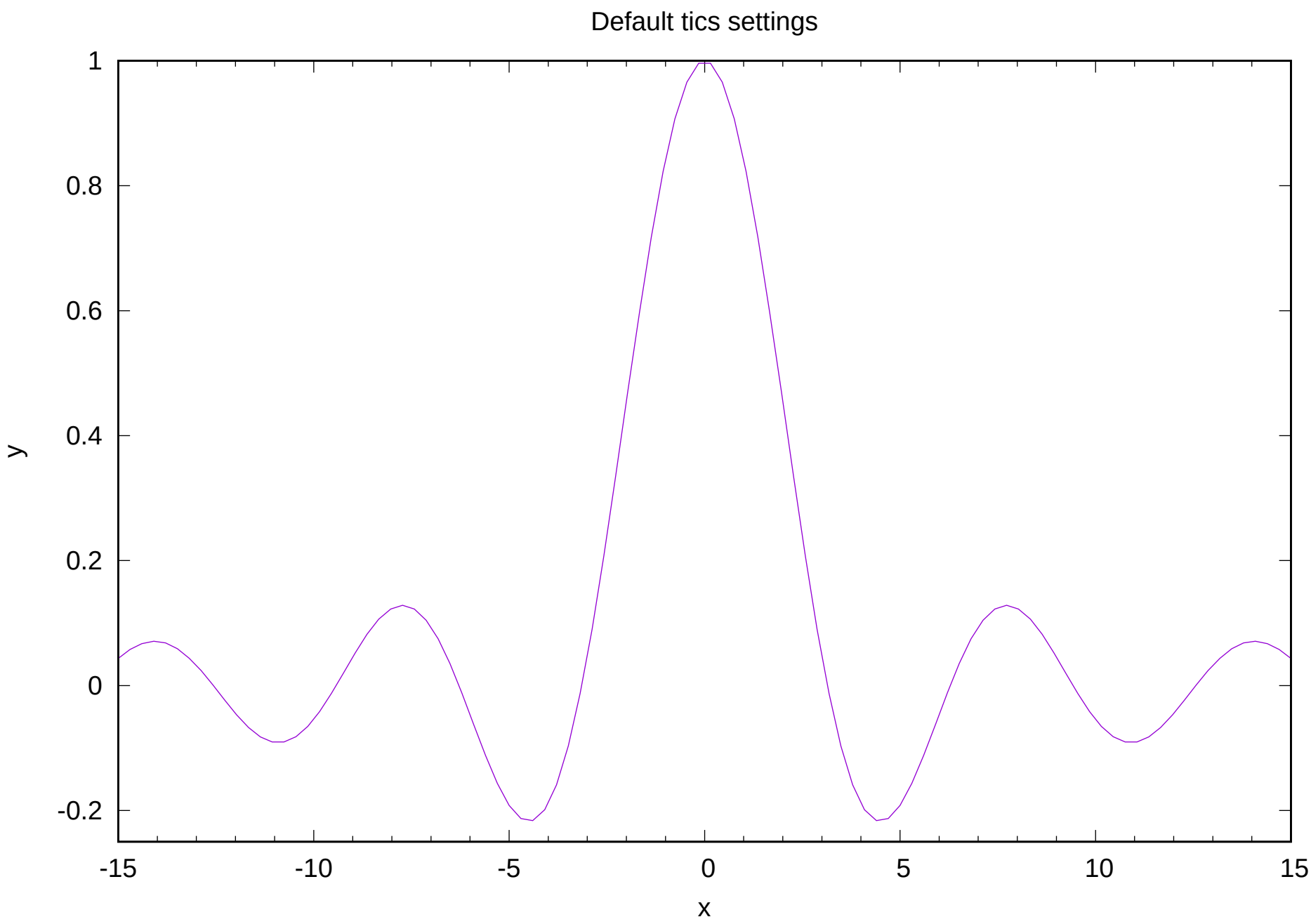
Custom dashtypes



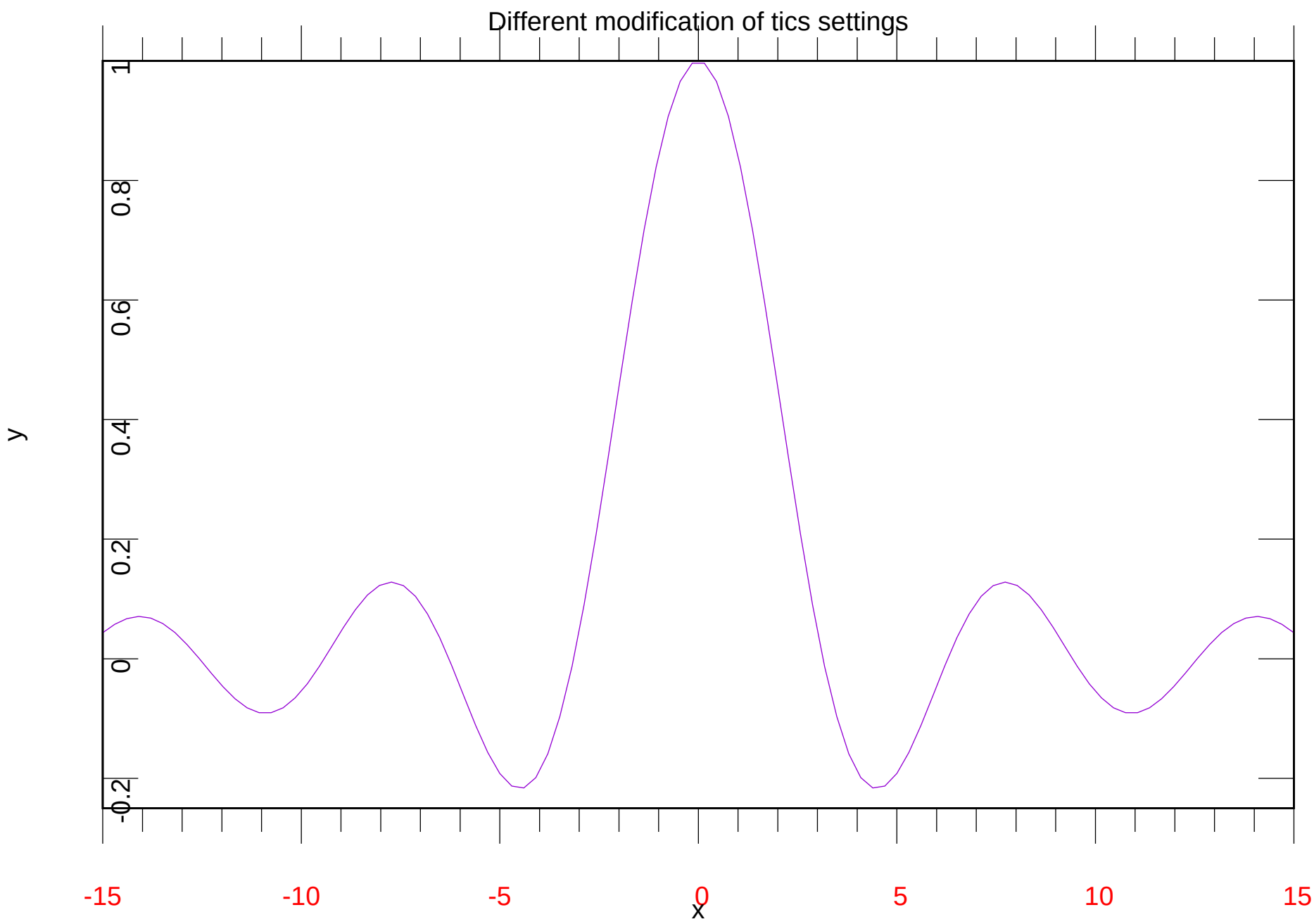

vtot(x,y)
3
2
1
0.5
0.2
0.1
0.05
0.02
0
-0.02
-0.05
-0.1
-0.2
-0.5
-1
-2
-3
-q
+q
10
5
0
-5
-10
-10
-5
0
5
10

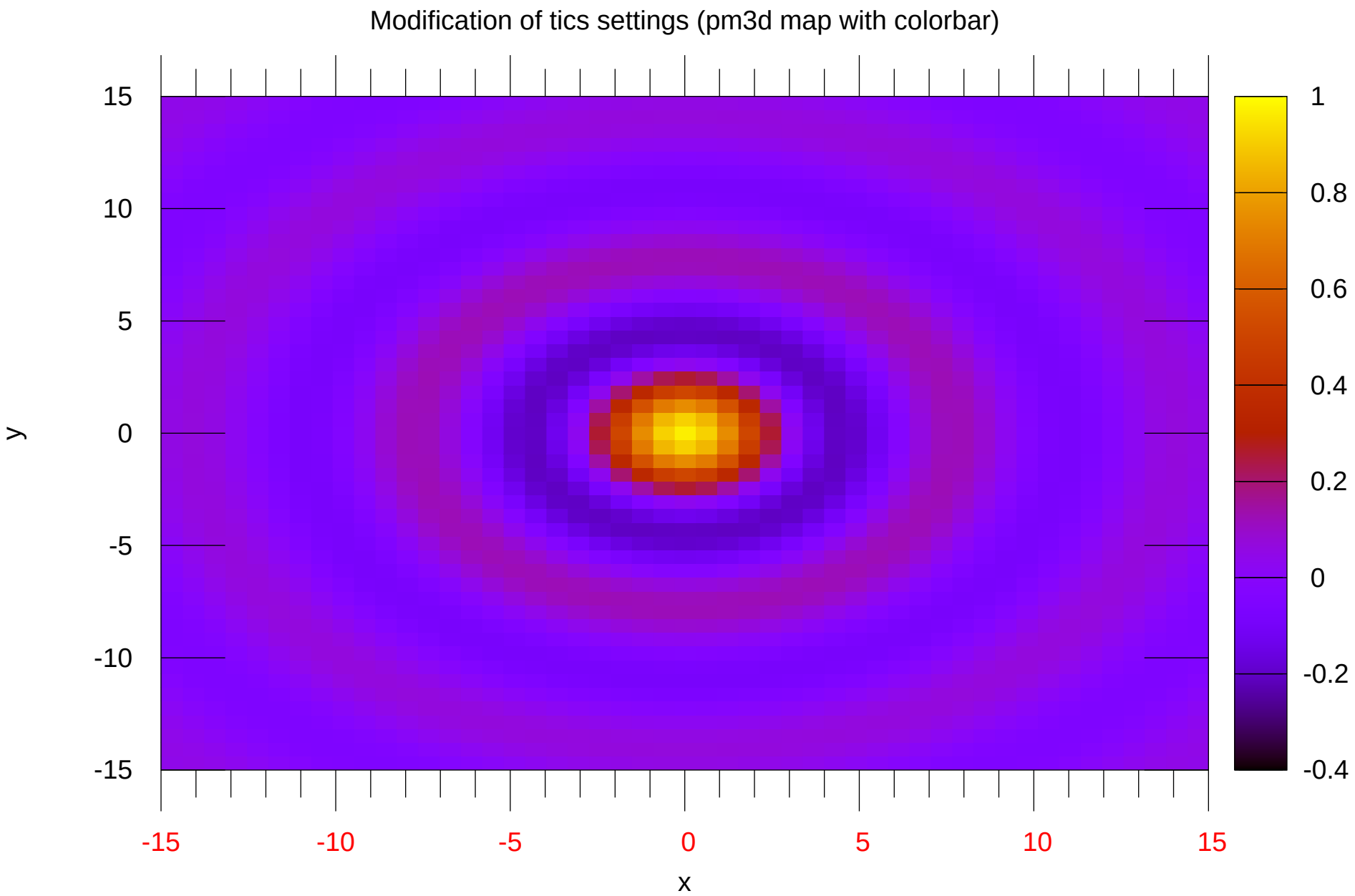

$equipo2
10
5
0
-5
-10
-10
-5
0
5
10


10
5
0
-5
-10
-10
-5
0
5
10
$field2xy u 1:2:(coef*dx1($1,$2)):(coef*dy1($1,$2))
$equipo2

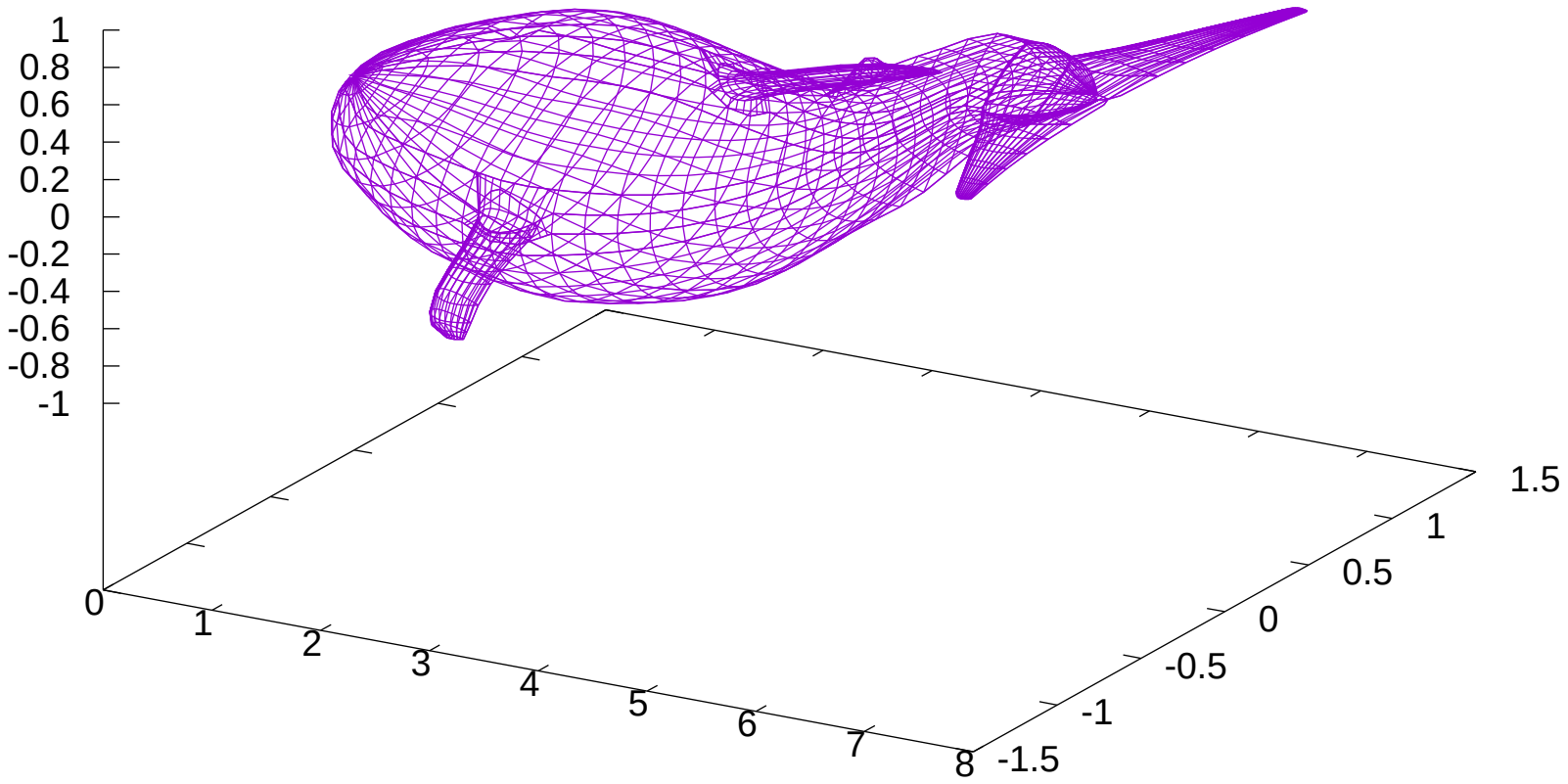

Default tics settings
1
0.8
0.6
0.4
0.2
0
-0.2
y
-15
-10
-5
0
5
10
15
x

Different modification of tics settings
y
x
1
0.8
0.6
0.4
0.2
0
-0.2
15
10
5
0
-5
-10
-15

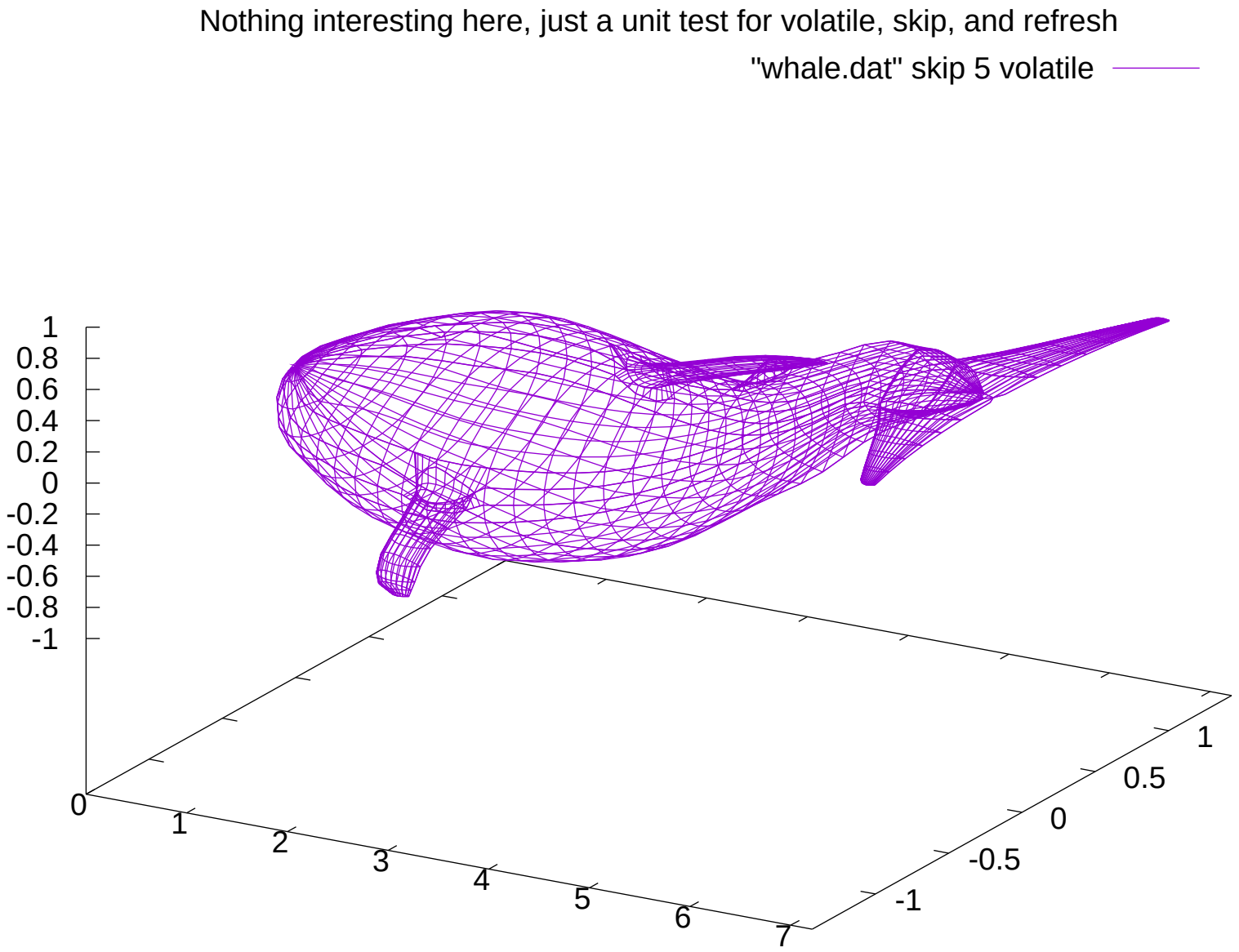

Different modification of tics settings
y
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x

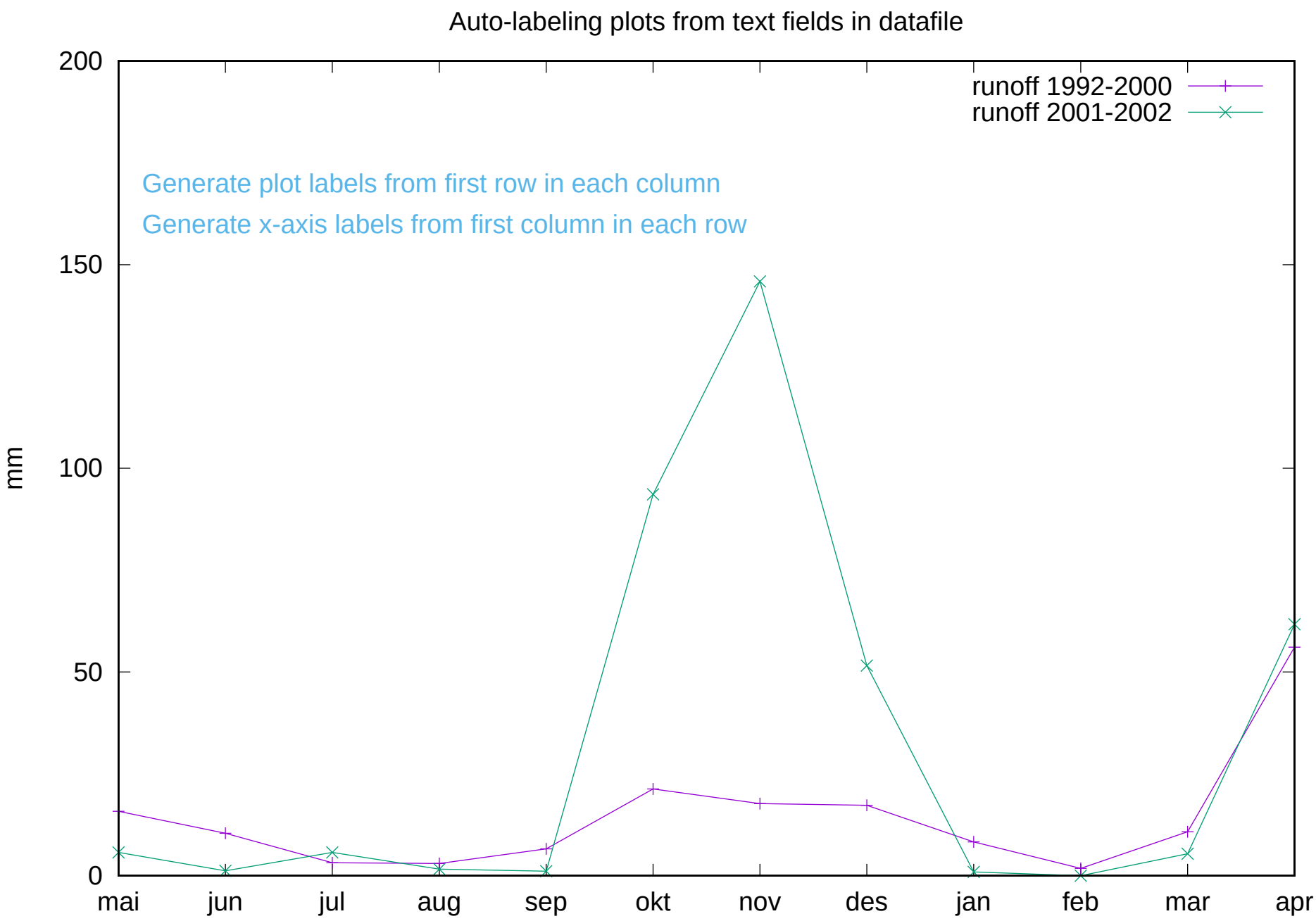

Modification of tics settings (pm3d map with colorbar)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

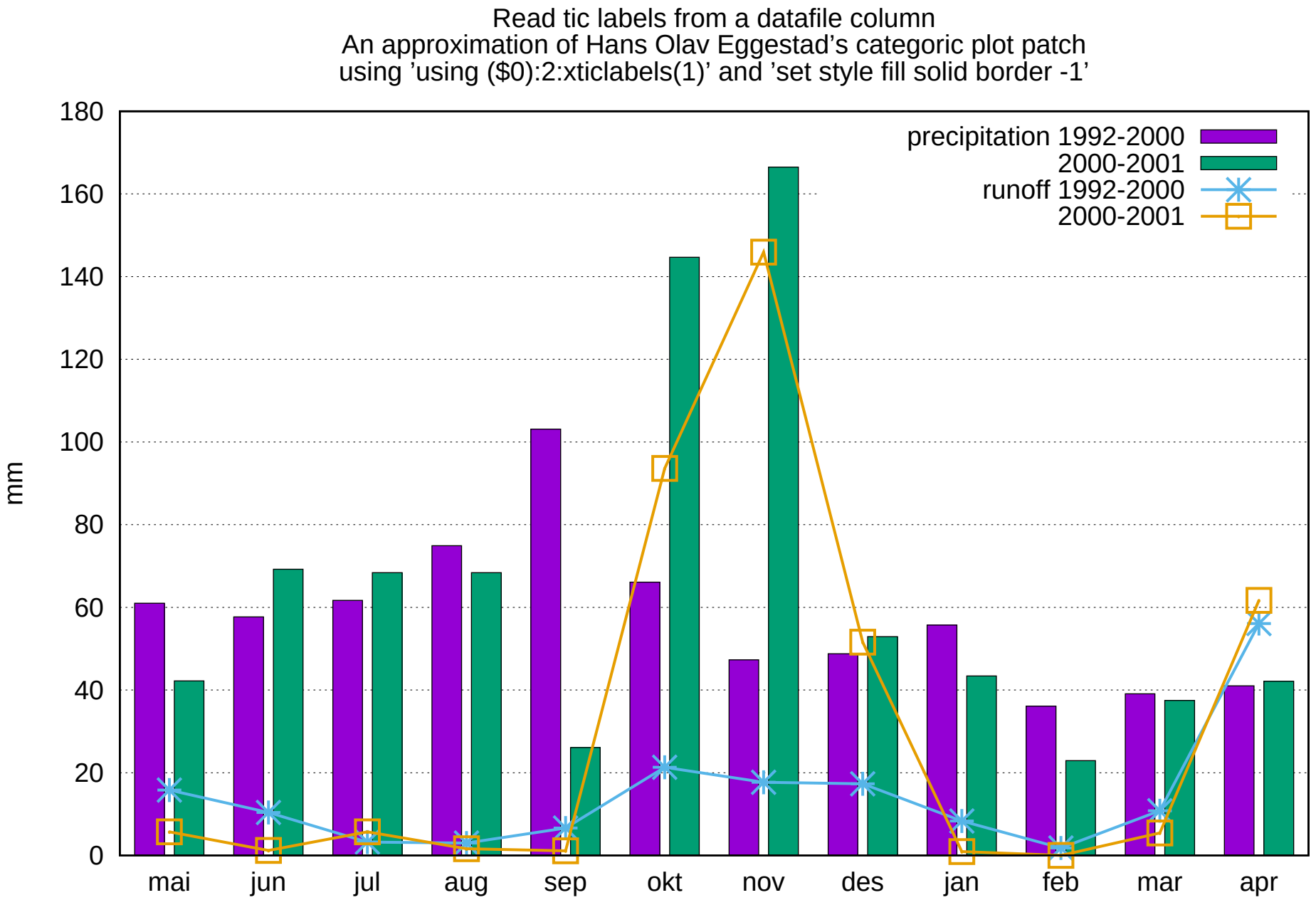
Nothing interesting here, just a unit test for volatile, skip, and refresh

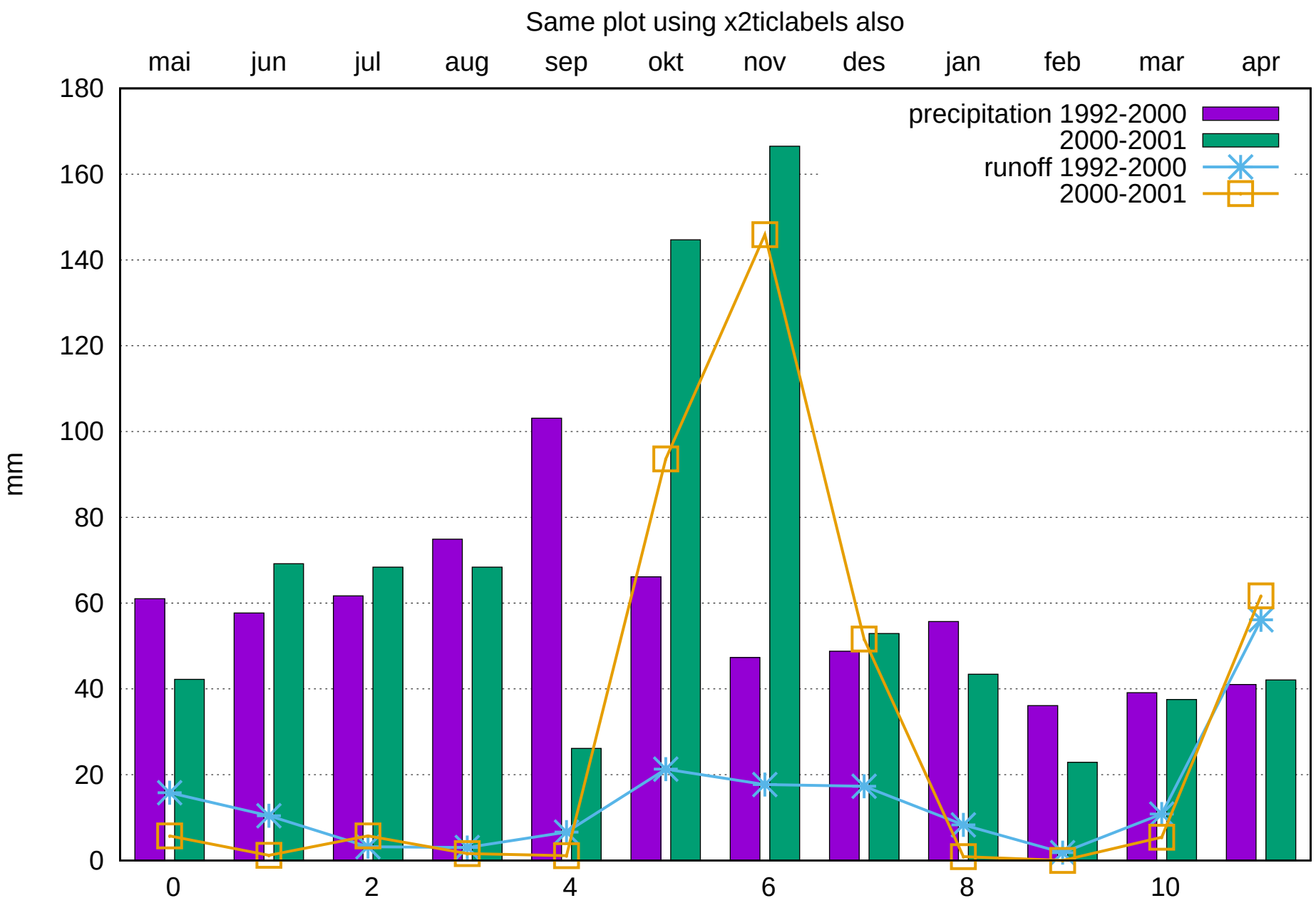


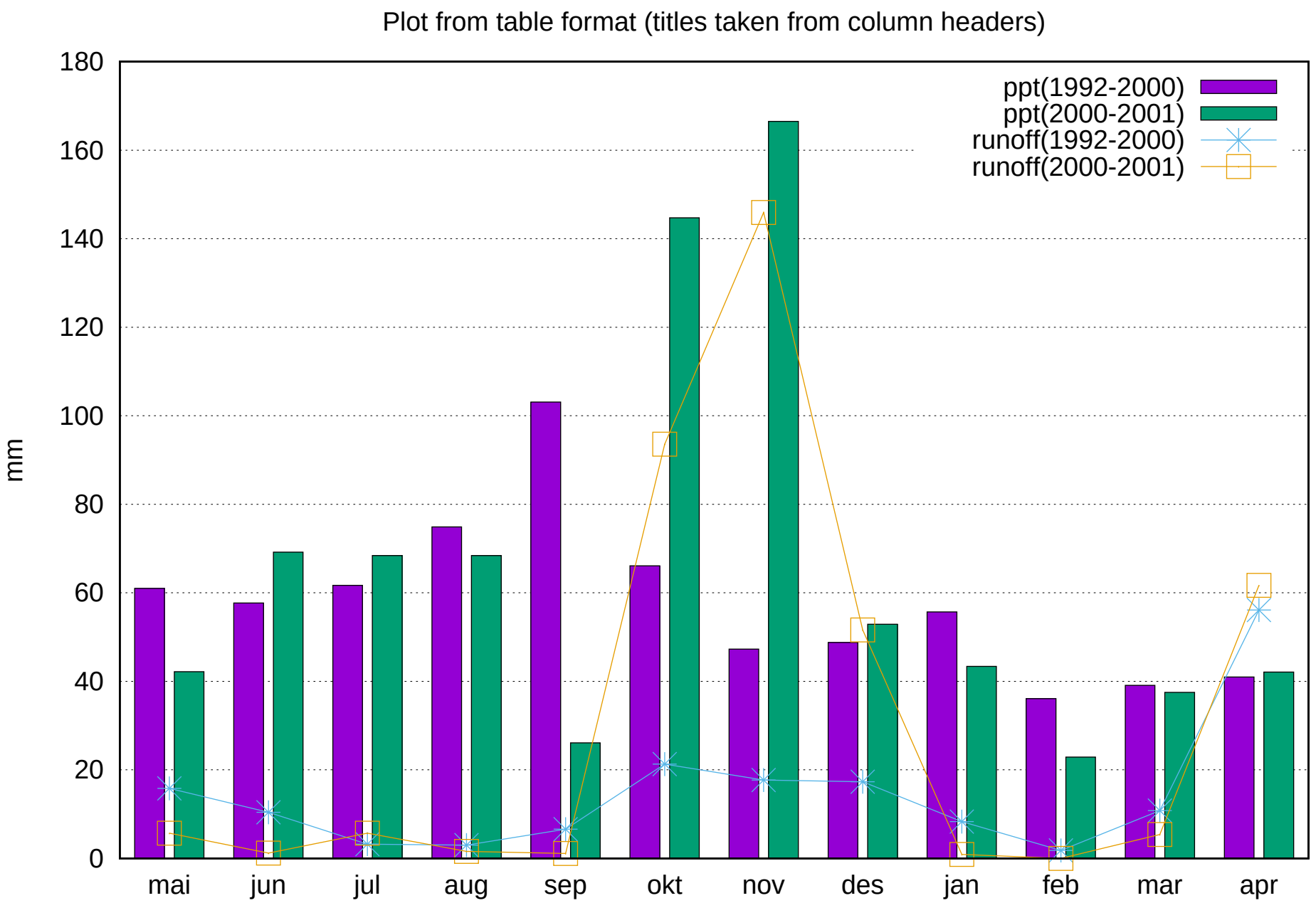
Nothing interesting here, just a unit test for volatile, skip, and refresh

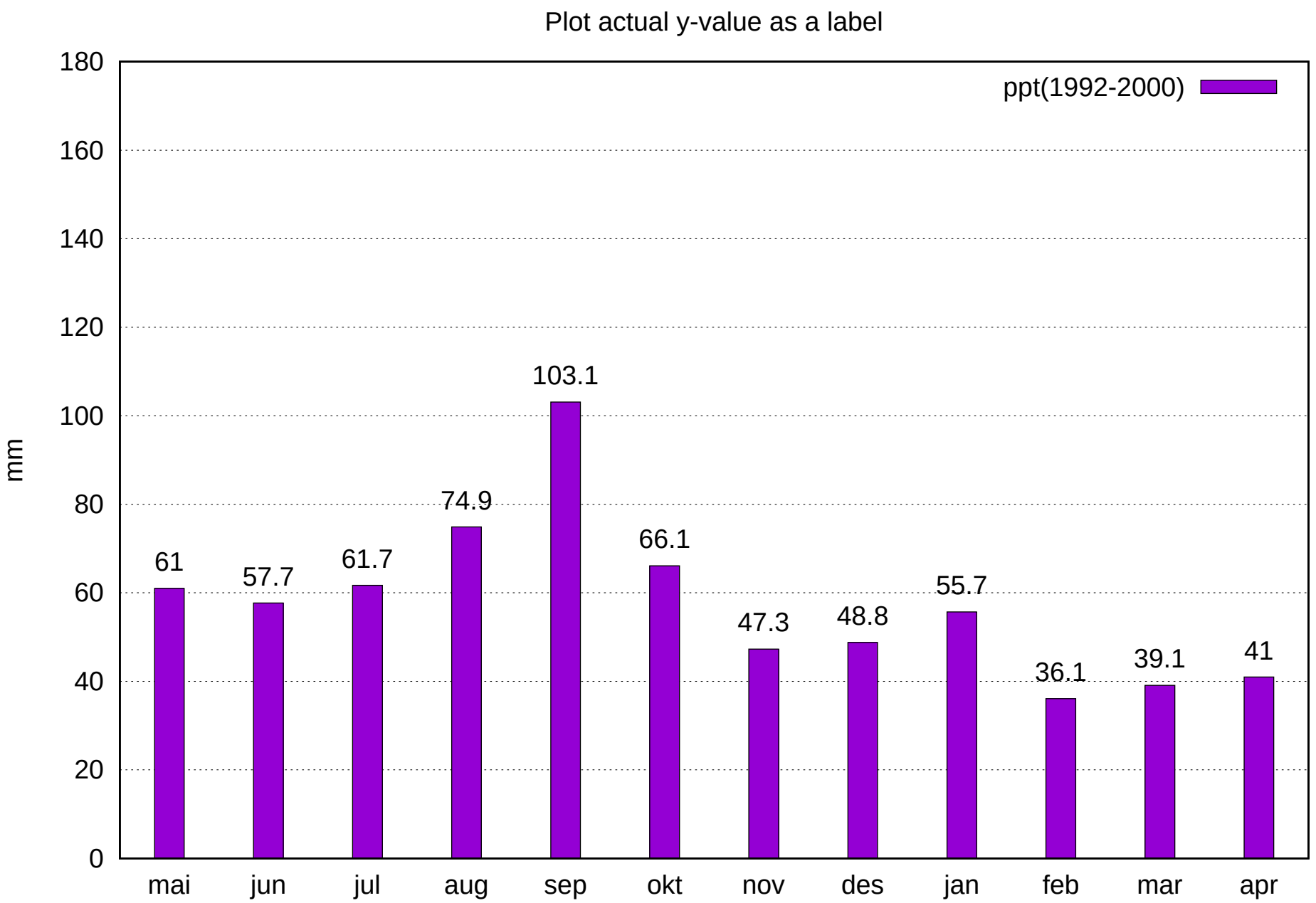


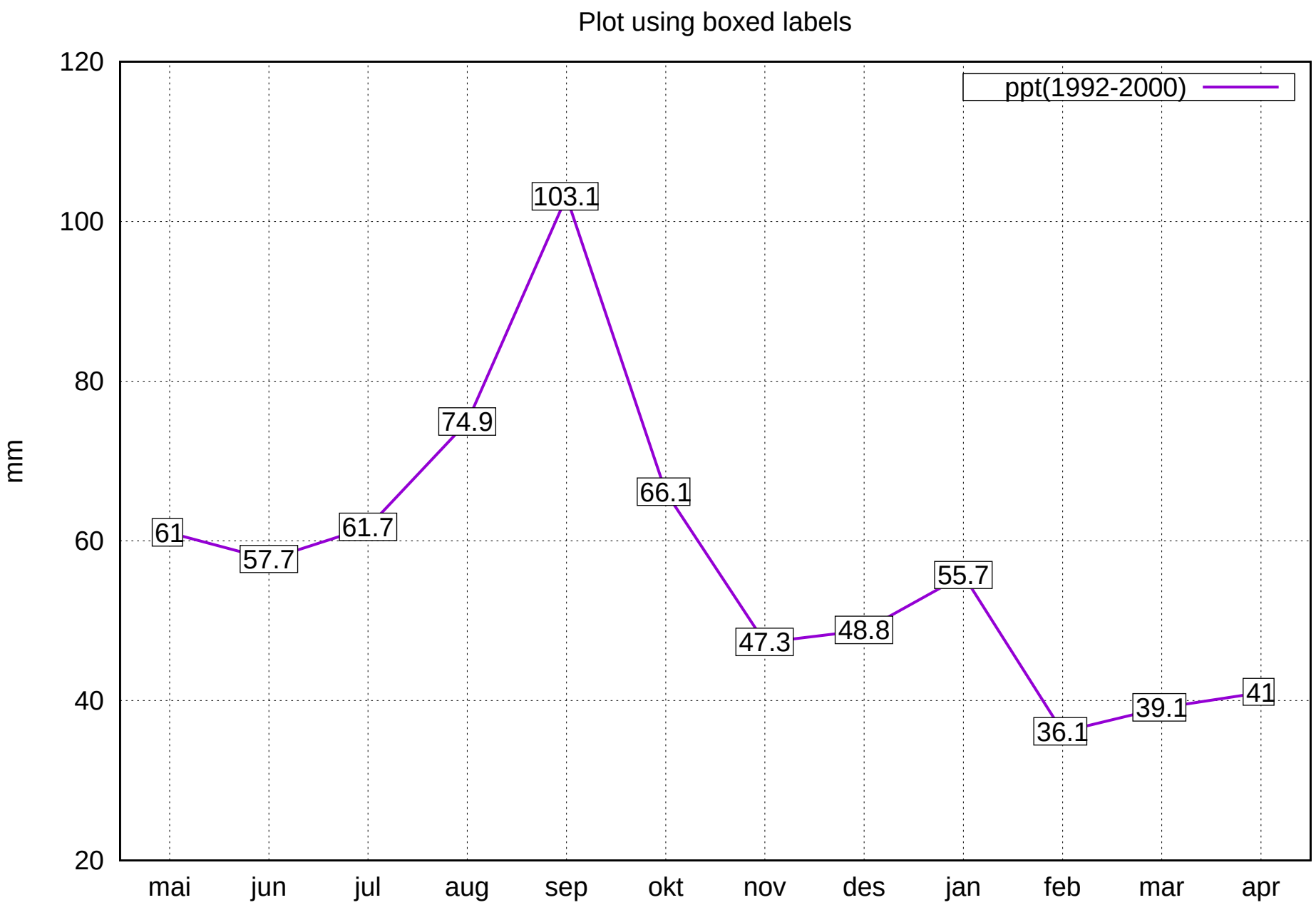

Auto-labeling plots from text fields in datafile
runoff 1992-2000
runoff 2001-2002
Generate plot labels from first row in each column
Generate x-axis labels from first column in each row
mm
200
150
100
50
0
mai
jun
jul
aug
sep
okt
nov
des
jan
feb
mar
apr

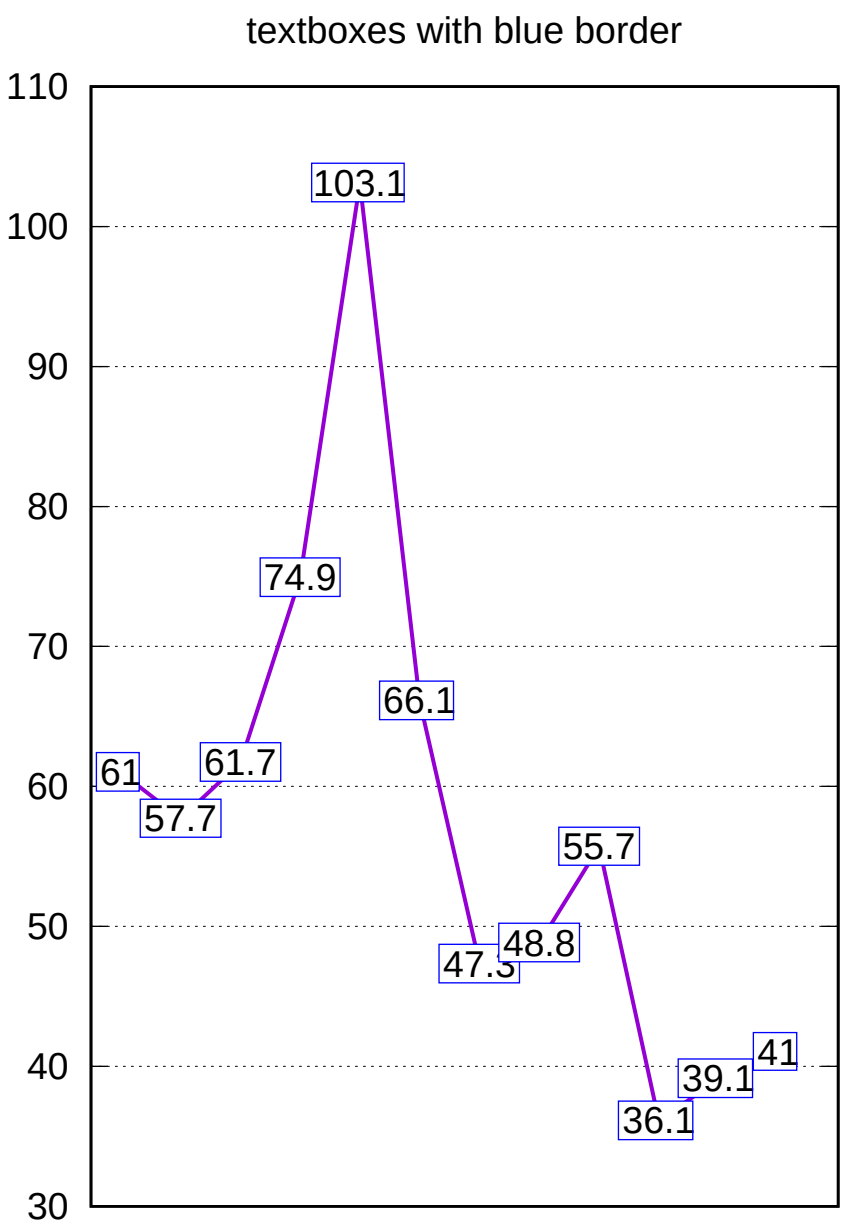

Read tic labels from a datafile column
An approximation of Hans Olav Eggestad's categoric plot patch
using 'using ($0):2:xticlabels(1)' and 'set style fill solid border -1'
mm
180
160
140
120
100
80
60
40
20
0
mai
jun
jul
aug
sep
okt
nov
des
jan
feb
mar
apr
precipitation 1992-2000
2000-2001
runoff 1992-2000
2000-2001

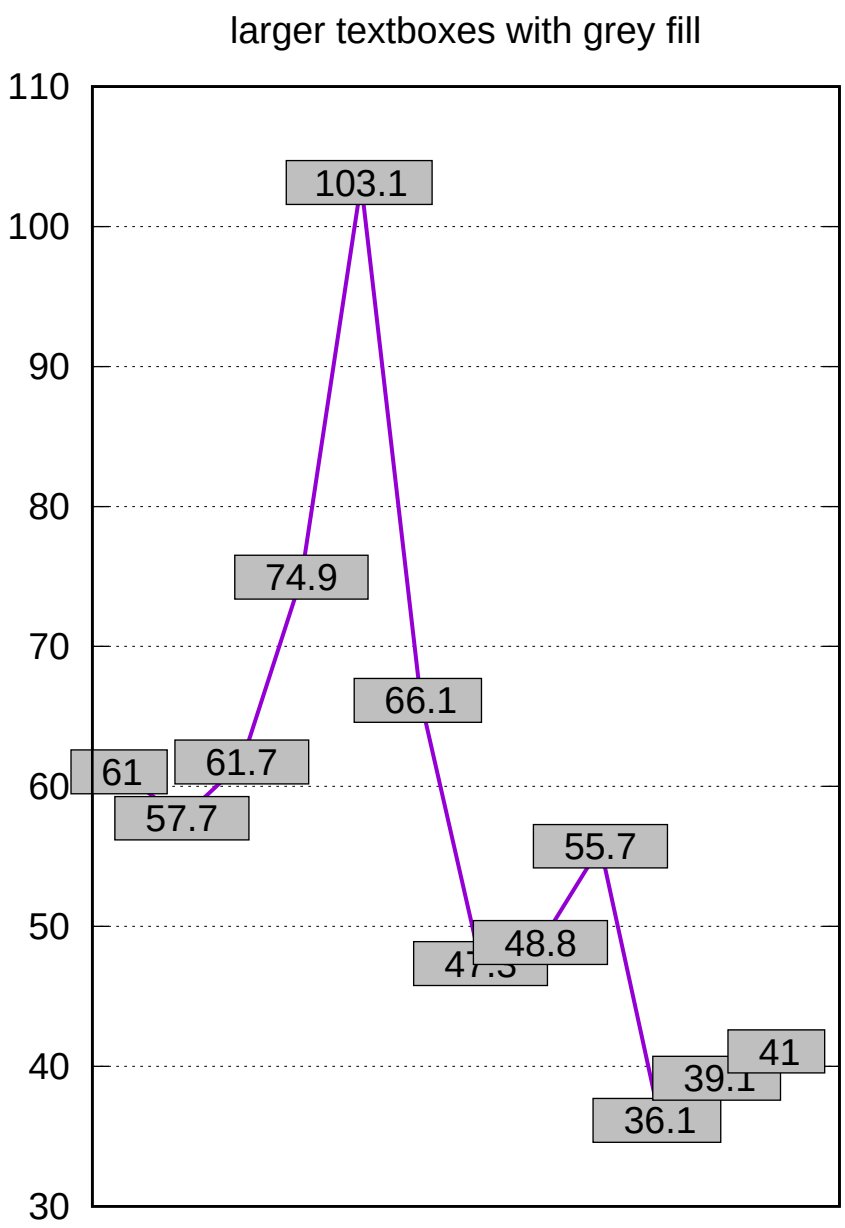

Same plot using x2ticlabels also
mai
jun
jul
aug
sep
okt
nov
des
jan
feb
mar
apr
180
160
140
120
100
80
60
40
20
0
mm
0
2
4
6
8
10
precipitation 1992-2000
2000-2001
runoff 1992-2000
2000-2001


Plot from table format (titles taken from column headers)
mm
180
160
140
120
100
80
60
40
20
0
mai
jun
jul
aug
sep
okt
nov
des
jan
feb
mar
apr
ppt(1992-2000)
ppt(2000-2001)
runoff(1992-2000)
runoff(2000-2001)

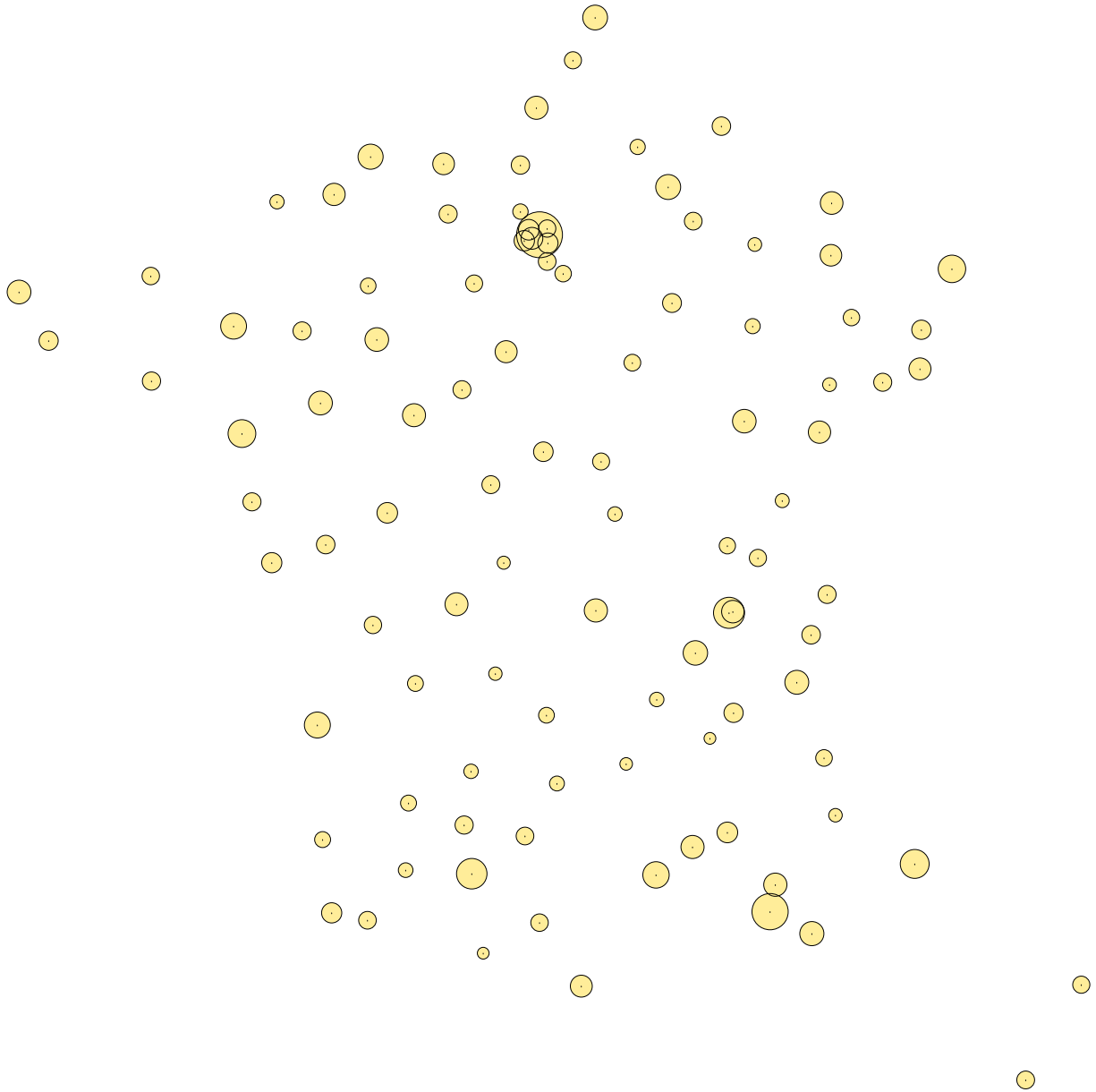

Plot actual y-value as a label
ppt(1992-2000)
mm
180
160
140
120
100
80
60
40
20
0
61
57.7
61.7
74.9
103.1
66.1
47.3
48.8
55.7
36.1
39.1
41
mai
jun
jul
aug
sep
okt
nov
des
jan
feb
mar
apr

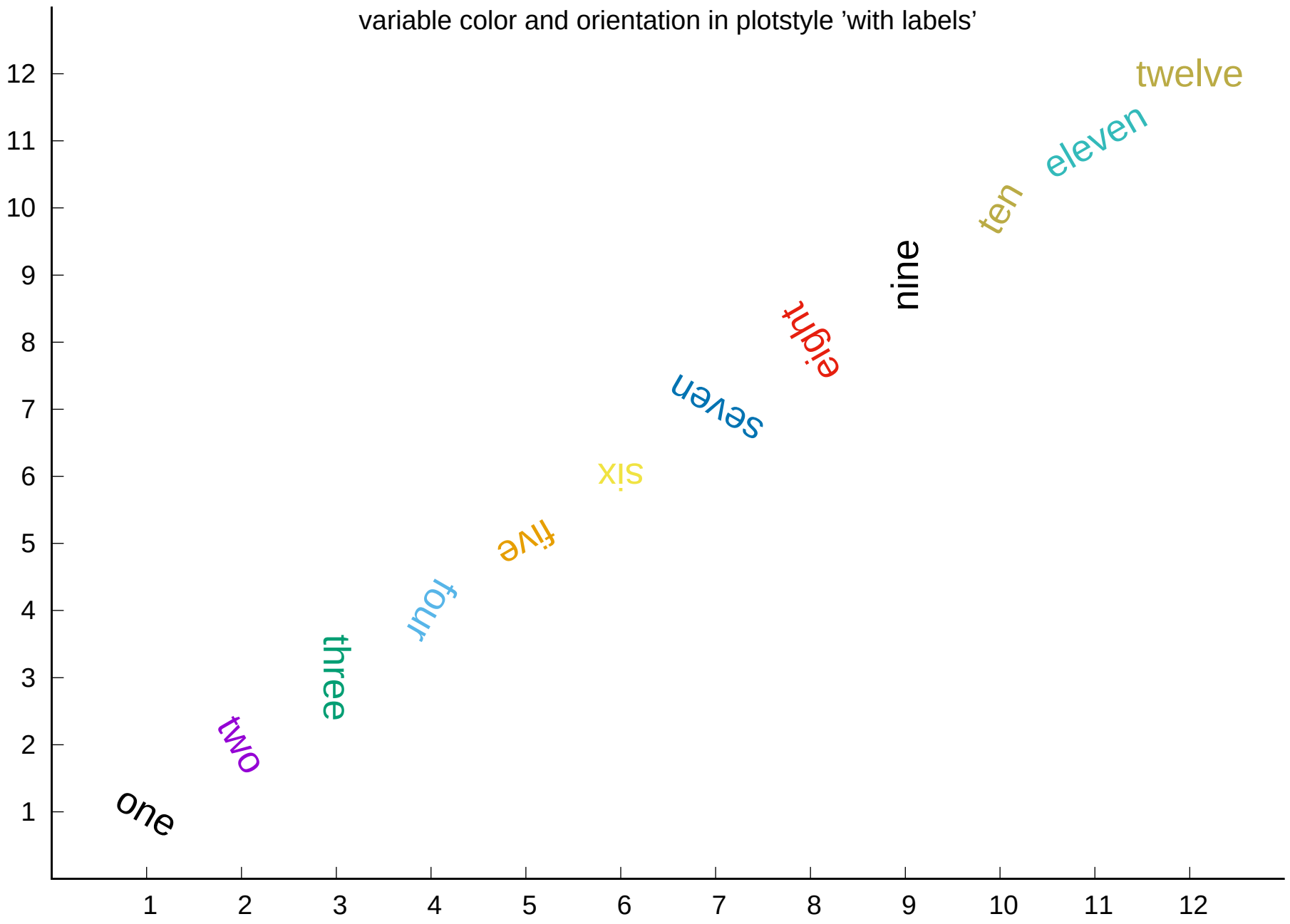

Plot using boxed labels
ppt(1992-2000)
mm
120
100
80
60
40
20
mai
jun
jul
aug
sep
okt
nov
des
jan
feb
mar
apr
61
57.7
61.7
74.9
103.1
66.1
47.3
48.8
55.7
36.1
39.1
41

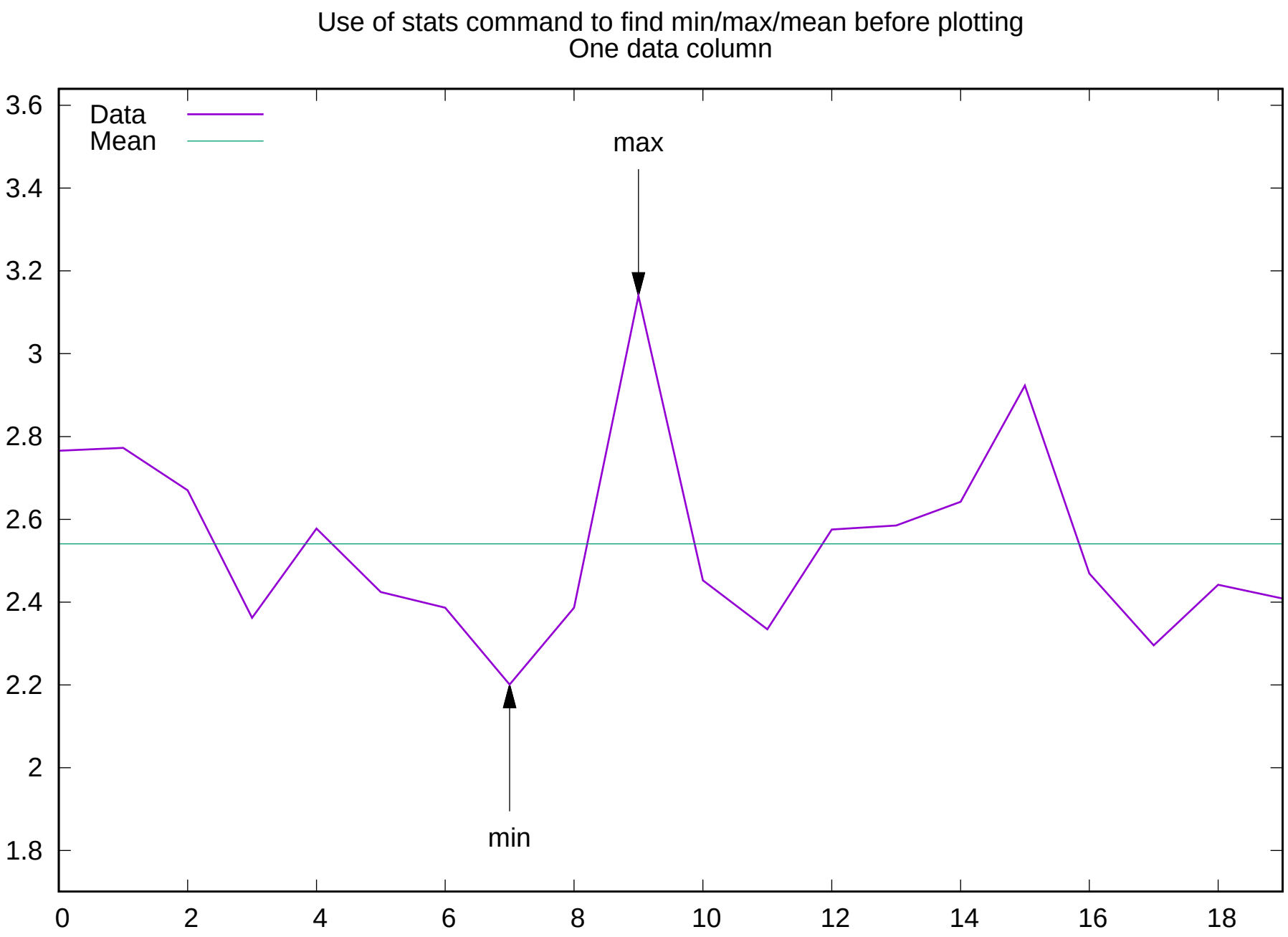

textboxes with blue border
110
100
90
80
70
60
50
40
30
103.1
74.9
66.1
61
61.7
57.7
55.7
48.8
47.3
41
39.1
36.1

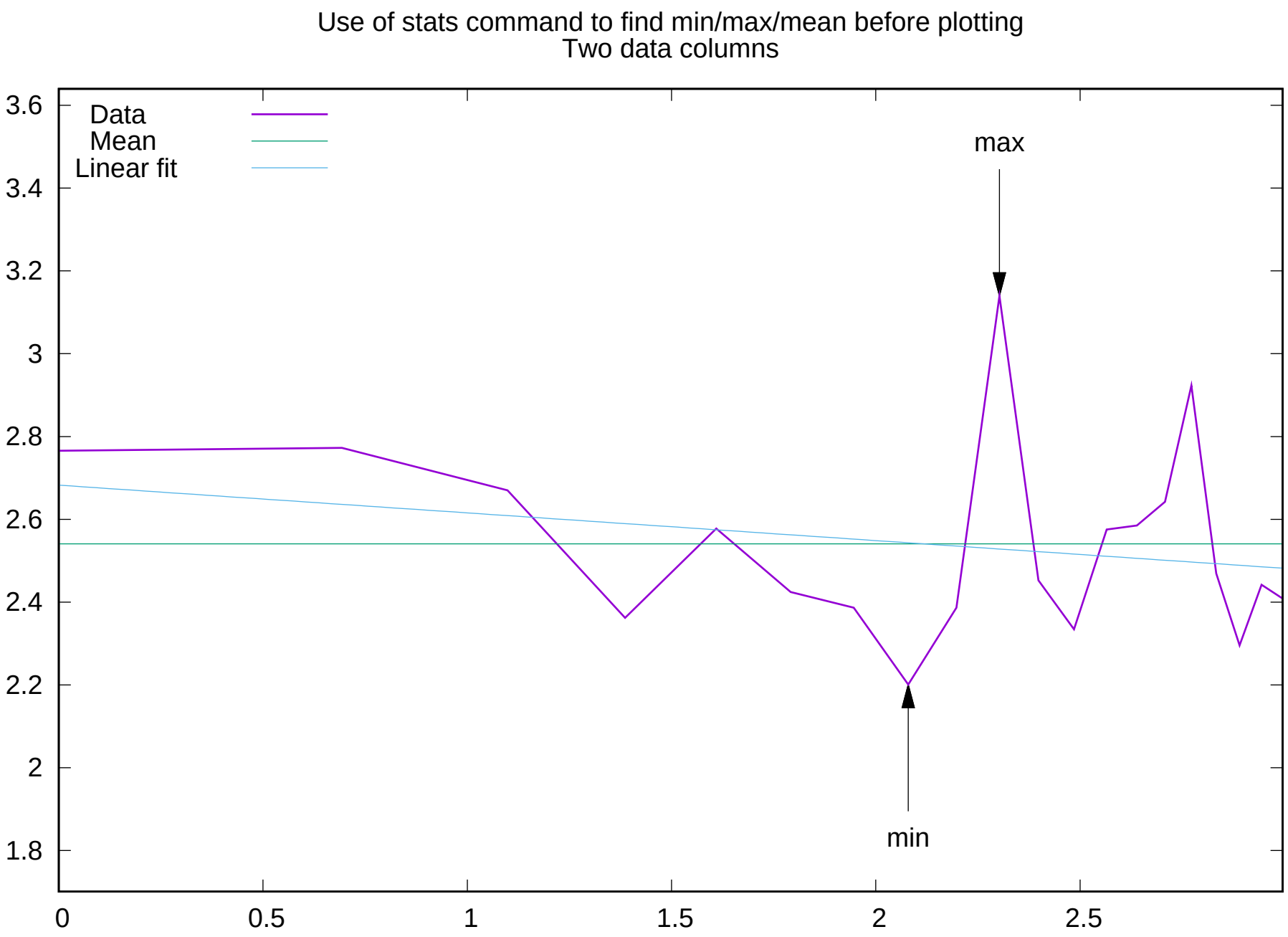

larger textboxes with grey fill
110
100
90
80
70
60
50
40
30
103.1
74.9
66.1
61
61.7
57.7
55.7
48.8
41
39.1
36.1

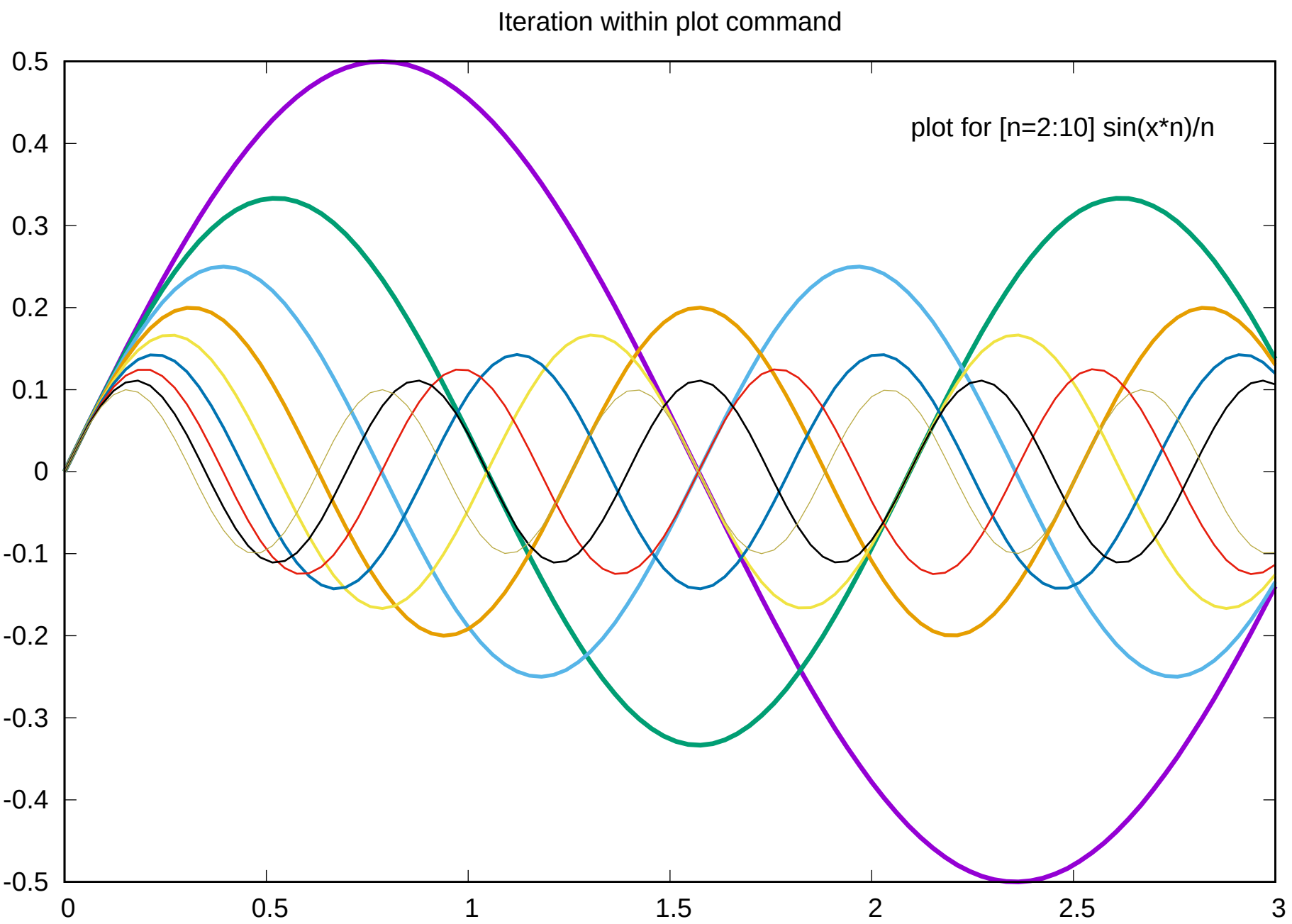
Hypertext is shown when the mouse is over a point



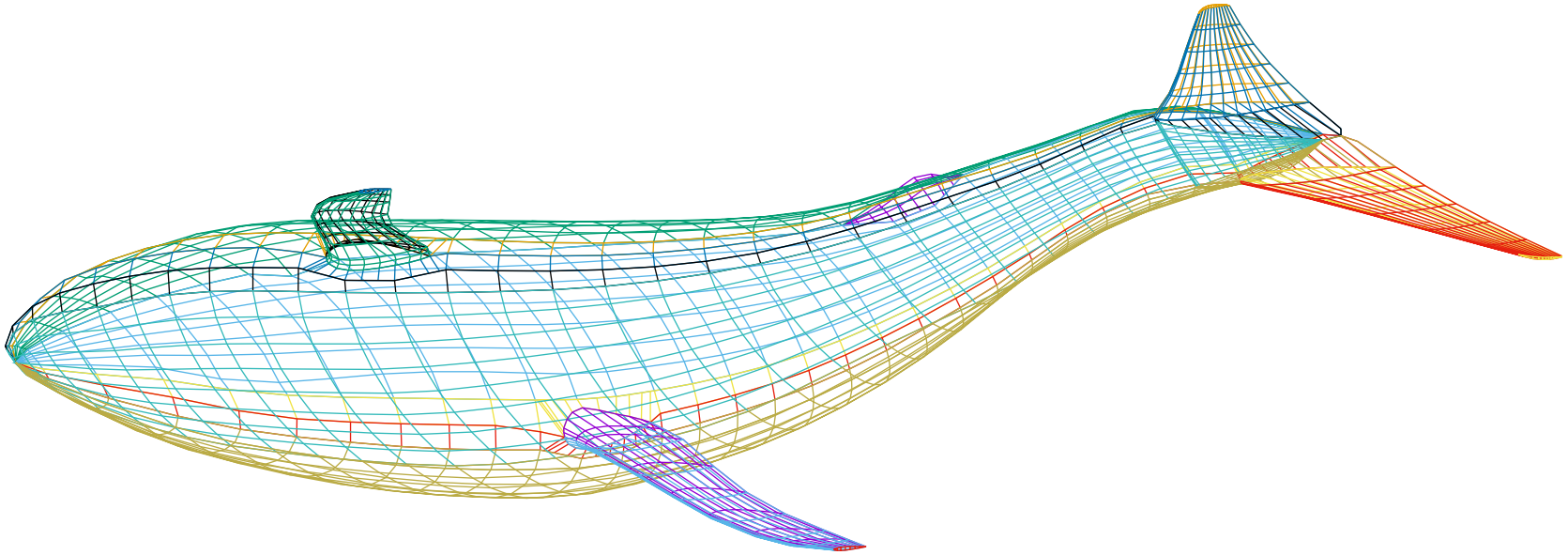

variable color and orientation in plotstyle 'with labels'
one
two
three
four
five
six
seven
eight
nine
ten
eleven
twelve
1
2
3
4
5
6
7
8
9
10
11
12


Use of stats command to find min/max/mean before plotting
One data column
Data
Mean
max
min
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
2
1.8
0
2
4
6
8
10
12
14
16
18


Use of stats command to find min/max/mean before plotting
Two data columns
Data
Mean
Linear fit
max
min
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
2
1.8
0
0.5
1
1.5
2
2.5

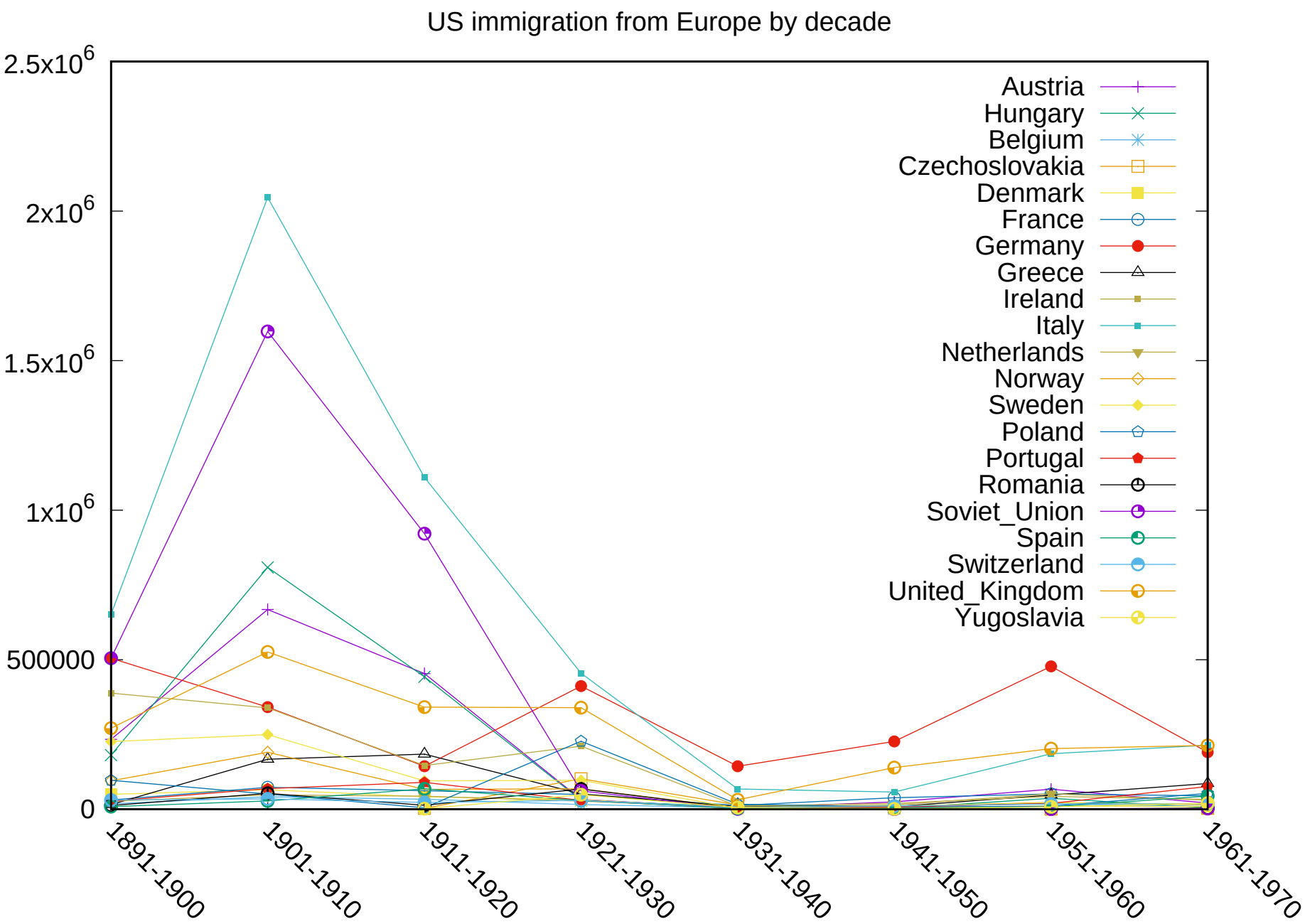

Iteration within plot command
plot for [n=2:10] sin(x*n)/n
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
0
0.5
1
1.5
2
2.5
3

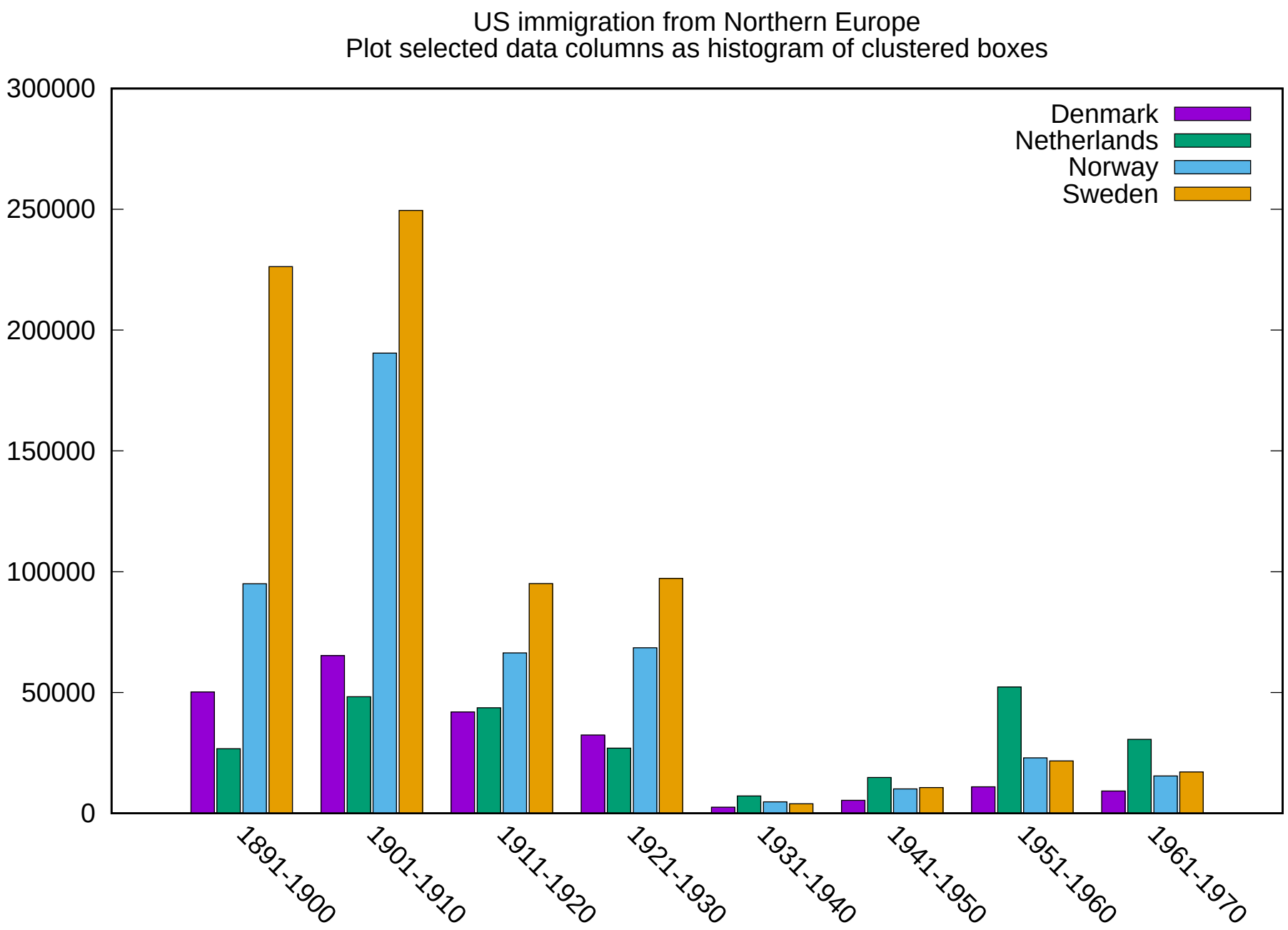
Iteration over all available data in a file

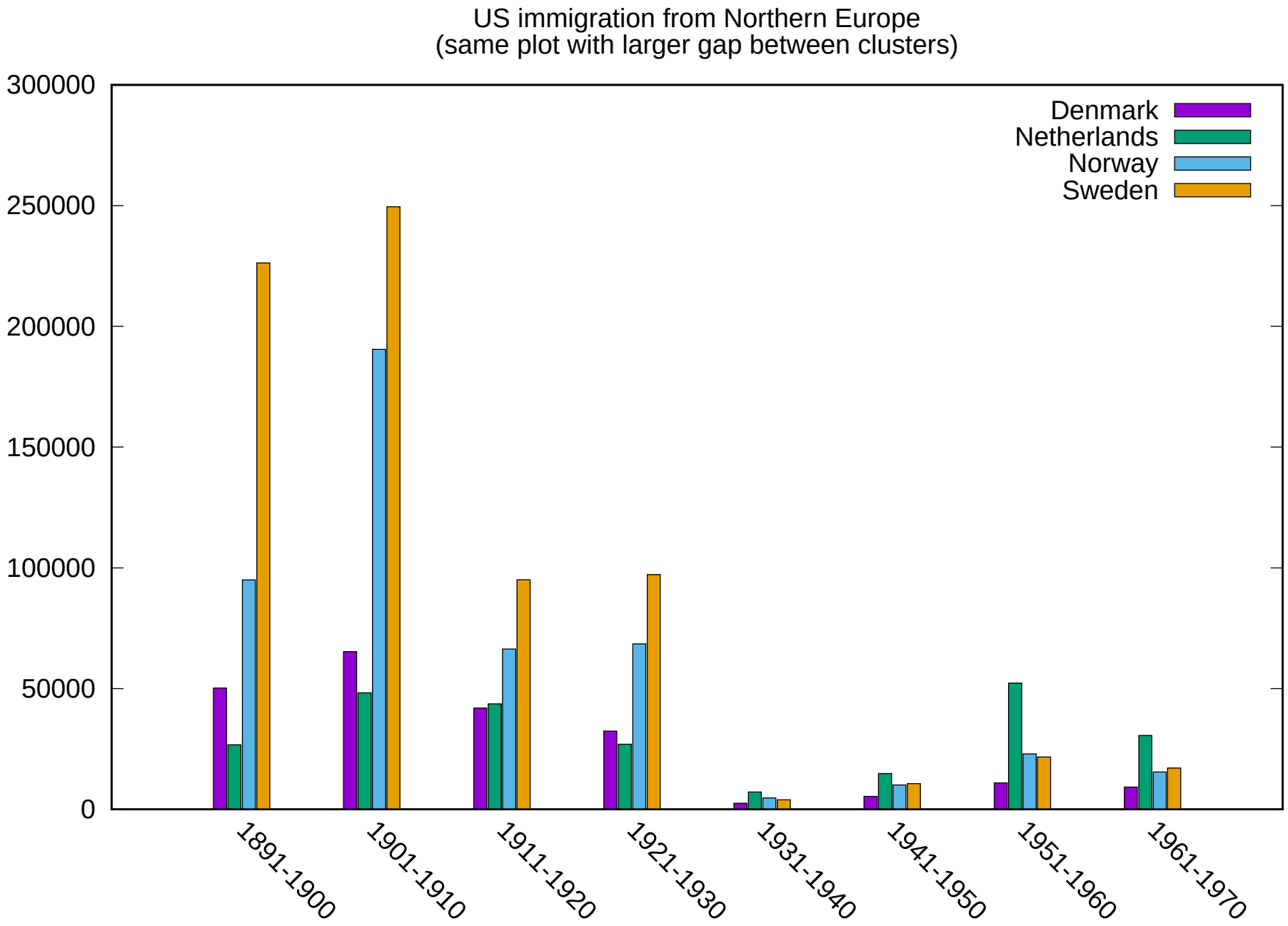


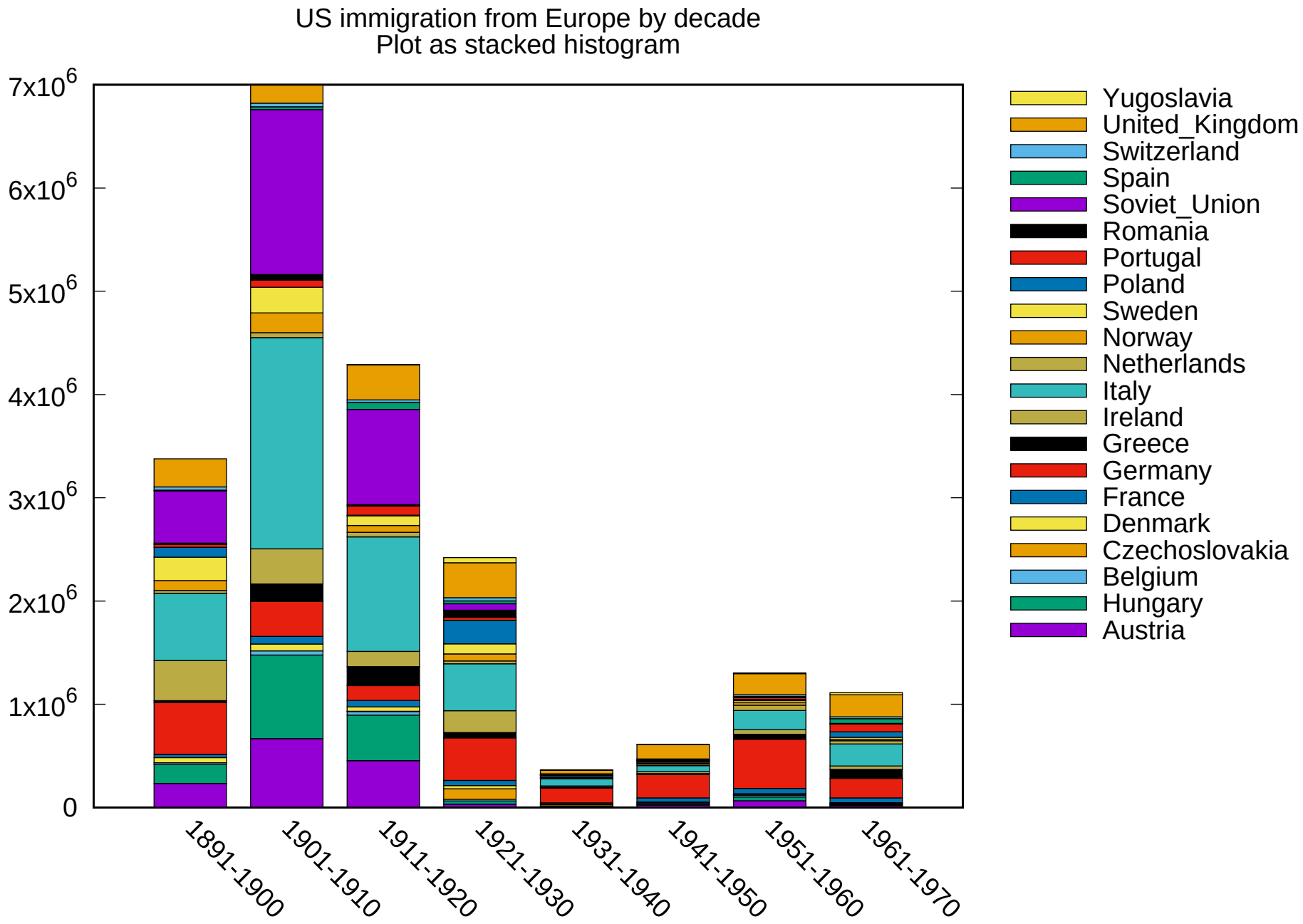
splot for [scan=1:*] 'whale.dat' index scan

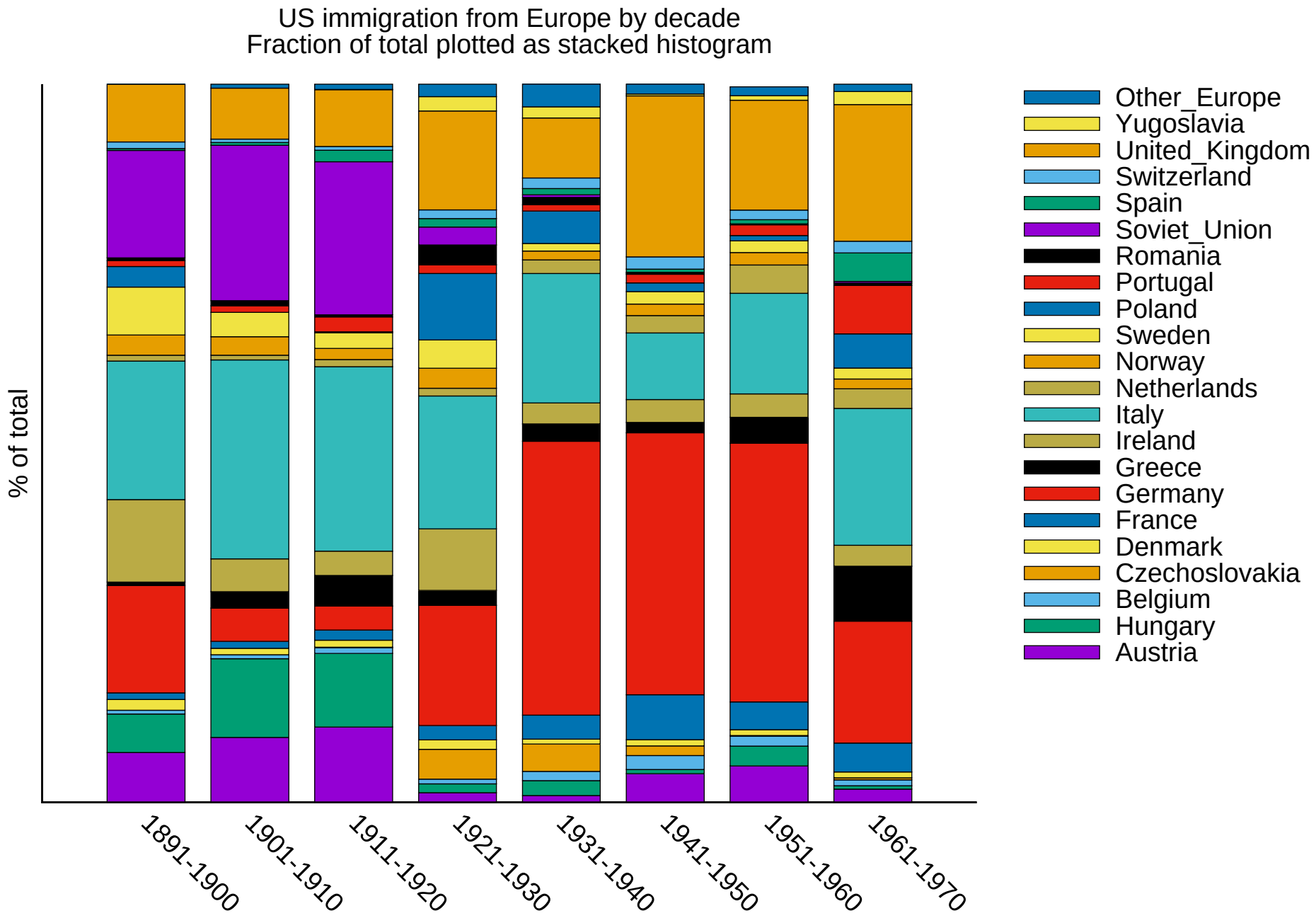
scan 1
scan 2
scan 3
scan 4
scan 5
scan 6
scan 7
scan 8
scan 9
scan 10
scan 11
scan 12
scan 13
scan 14
scan 15
scan 16
scan 17
scan 18
scan 19
scan 20
scan 21
scan 22
scan 23

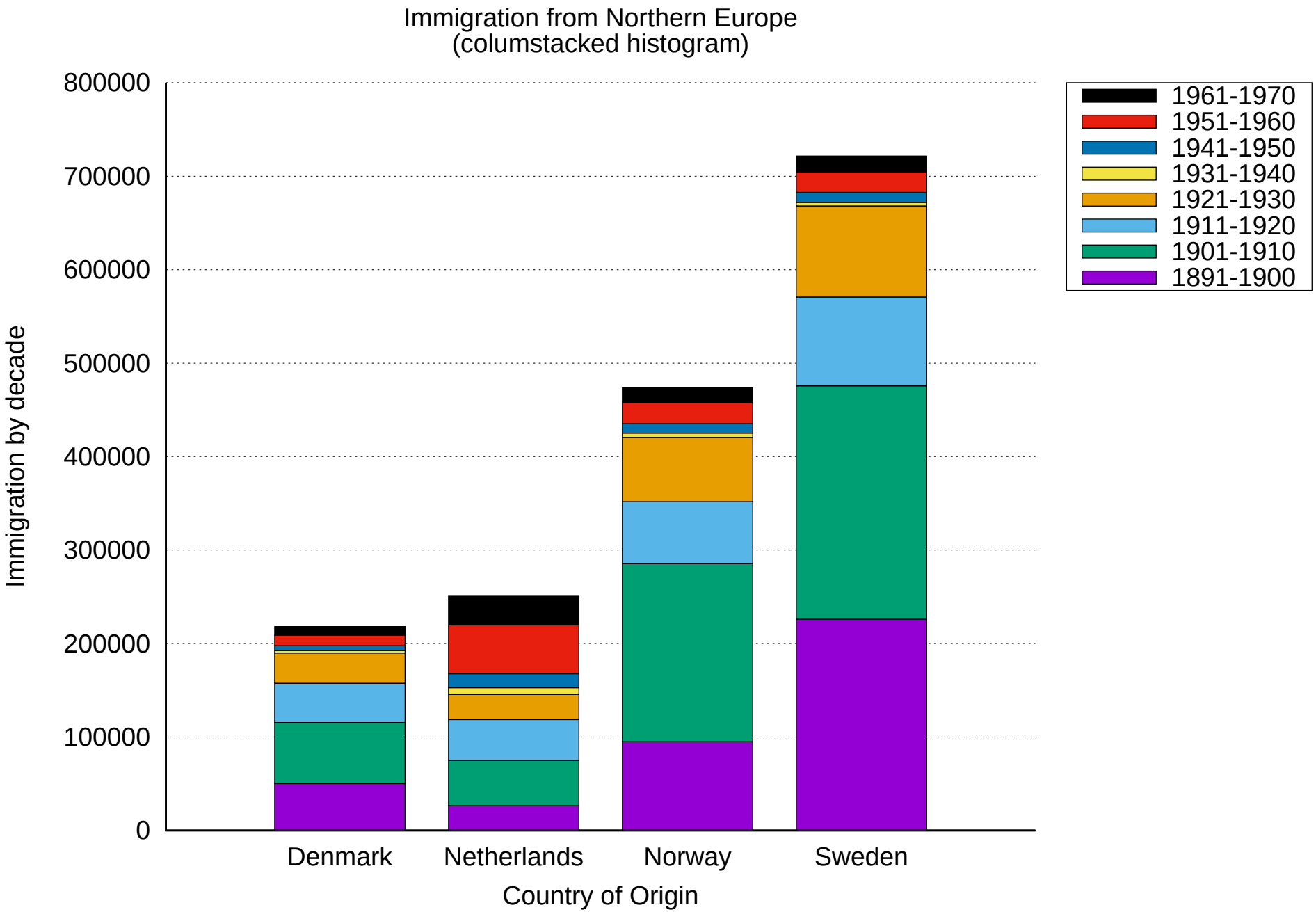

US immigration from Europe by decade
2.5x10^6
2x10^6
1.5x10^6
1x10^6
500000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Austria
Hungary
Belgium
Czechoslovakia
Denmark
France
Germany
Greece
Ireland
Italy
Netherlands
Norway
Sweden
Poland
Portugal
Romania
Soviet_Union
Spain
Switzerland
United_Kingdom
Yugoslavia

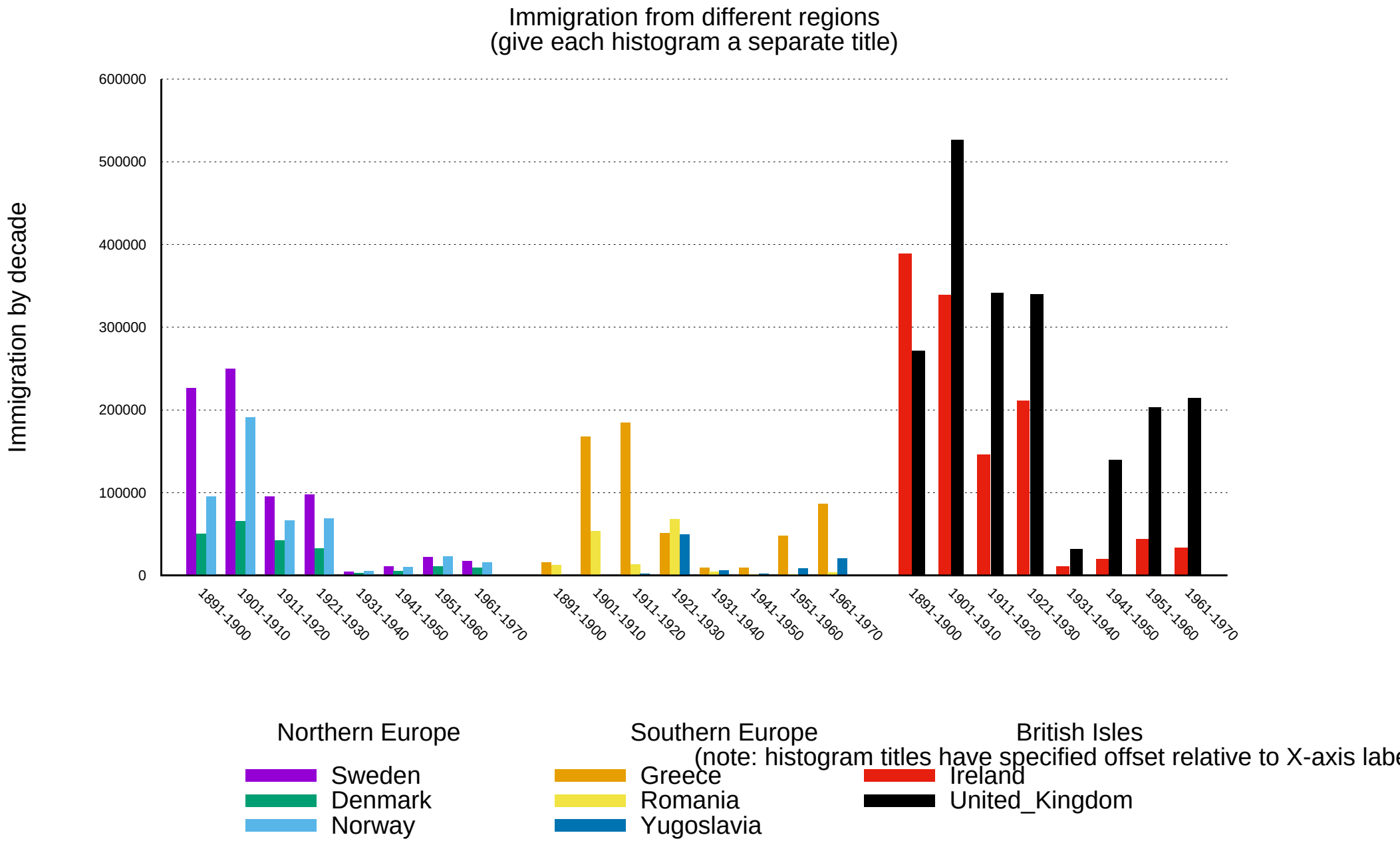

US immigration from Northern Europe
Plot selected data columns as histogram of clustered boxes
300000
250000
200000
150000
100000
50000
0
Denmark
Netherlands
Norway
Sweden
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970

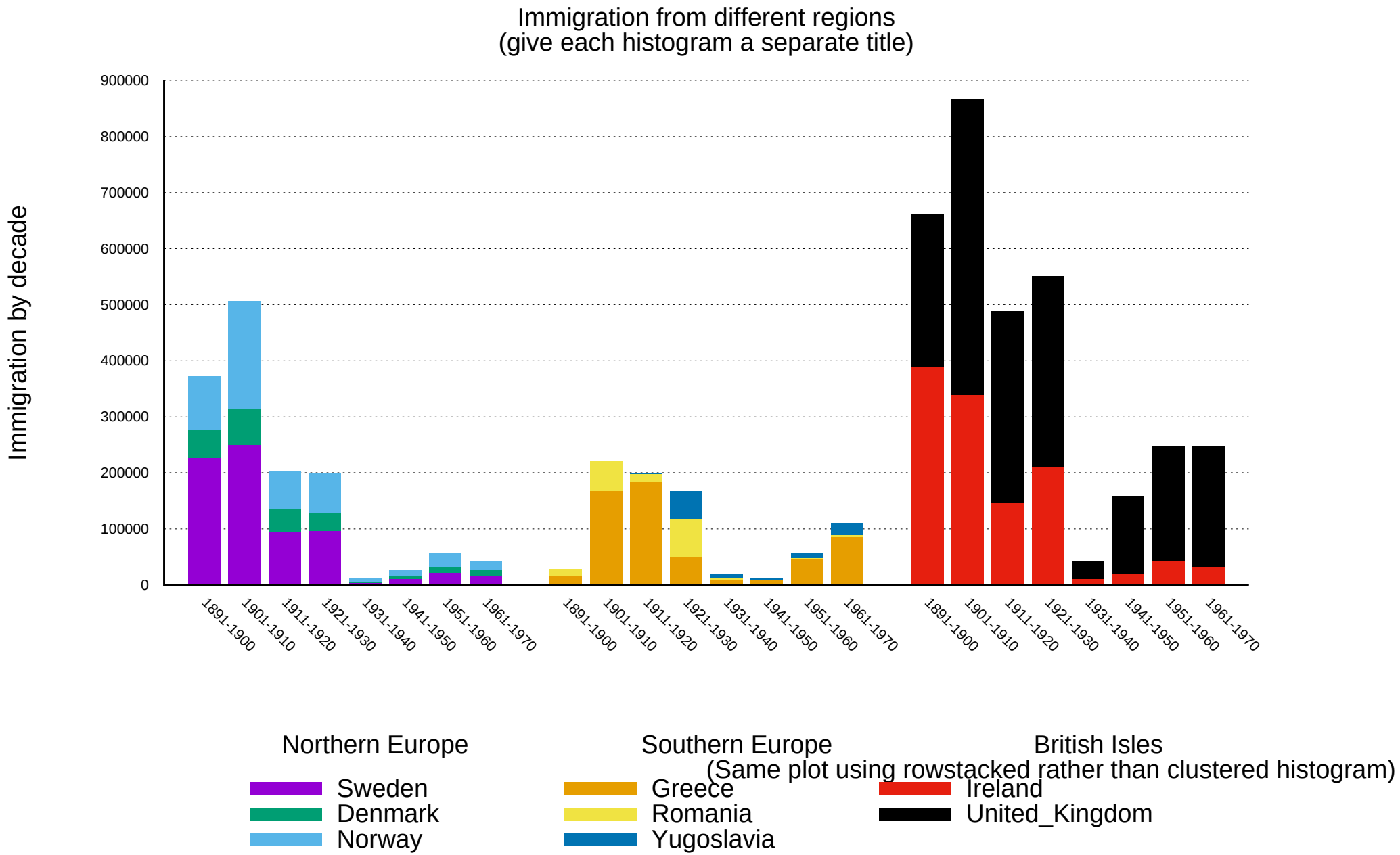

US immigration from Northern Europe
(same plot with larger gap between clusters)
300000
250000
200000
150000
100000
50000
0
Denmark
Netherlands
Norway
Sweden
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970

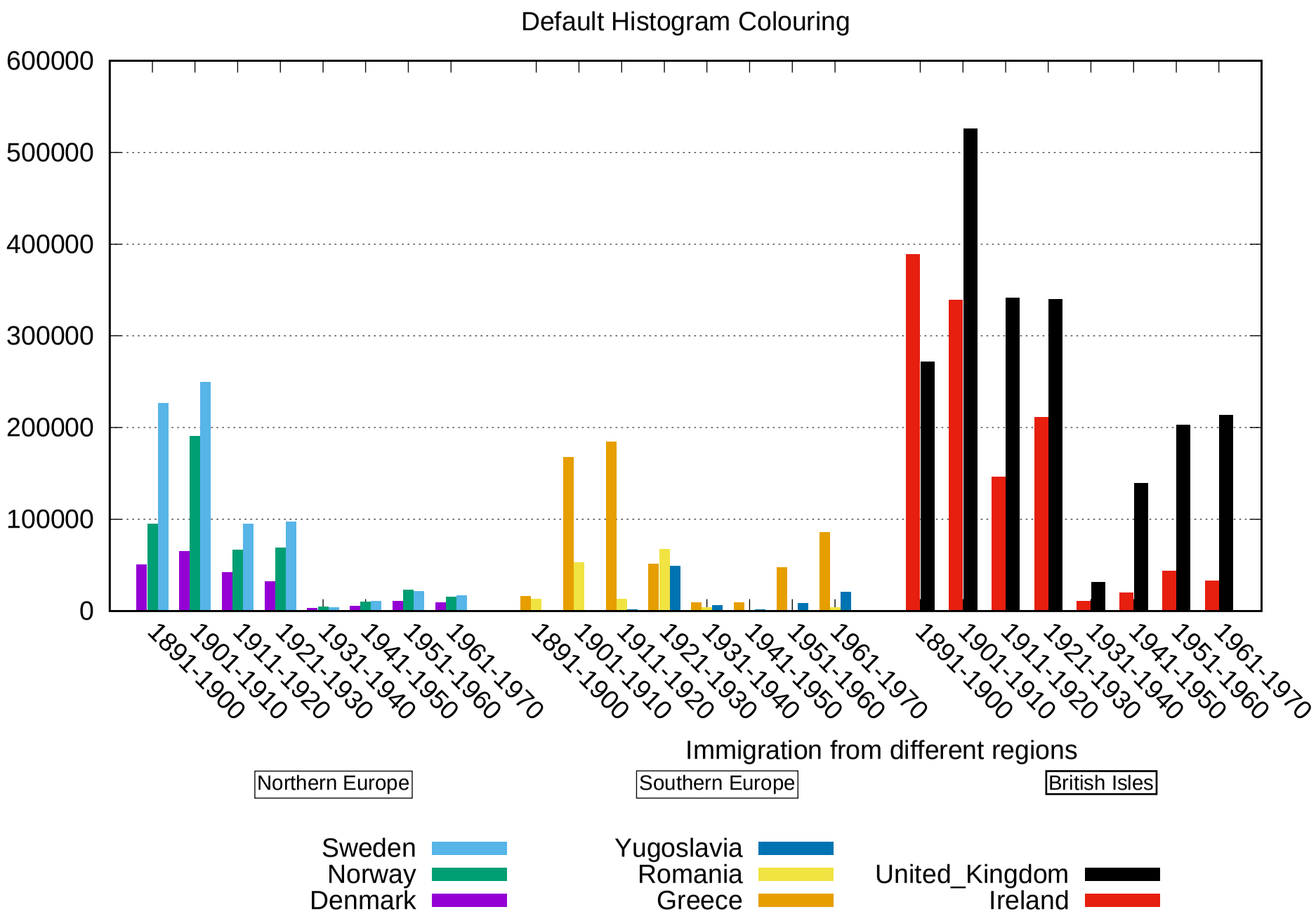

US immigration from Europe by decade
Plot as stacked histogram
7x10^6
6x10^6
5x10^6
4x10^6
3x10^6
2x10^6
1x10^6
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Yugoslavia
United_Kingdom
Switzerland
Spain
Soviet_Union
Romania
Portugal
Poland
Sweden
Norway
Netherlands
Italy
Ireland
Greece
Germany
France
Denmark
Czechoslovakia
Belgium
Hungary
Austria

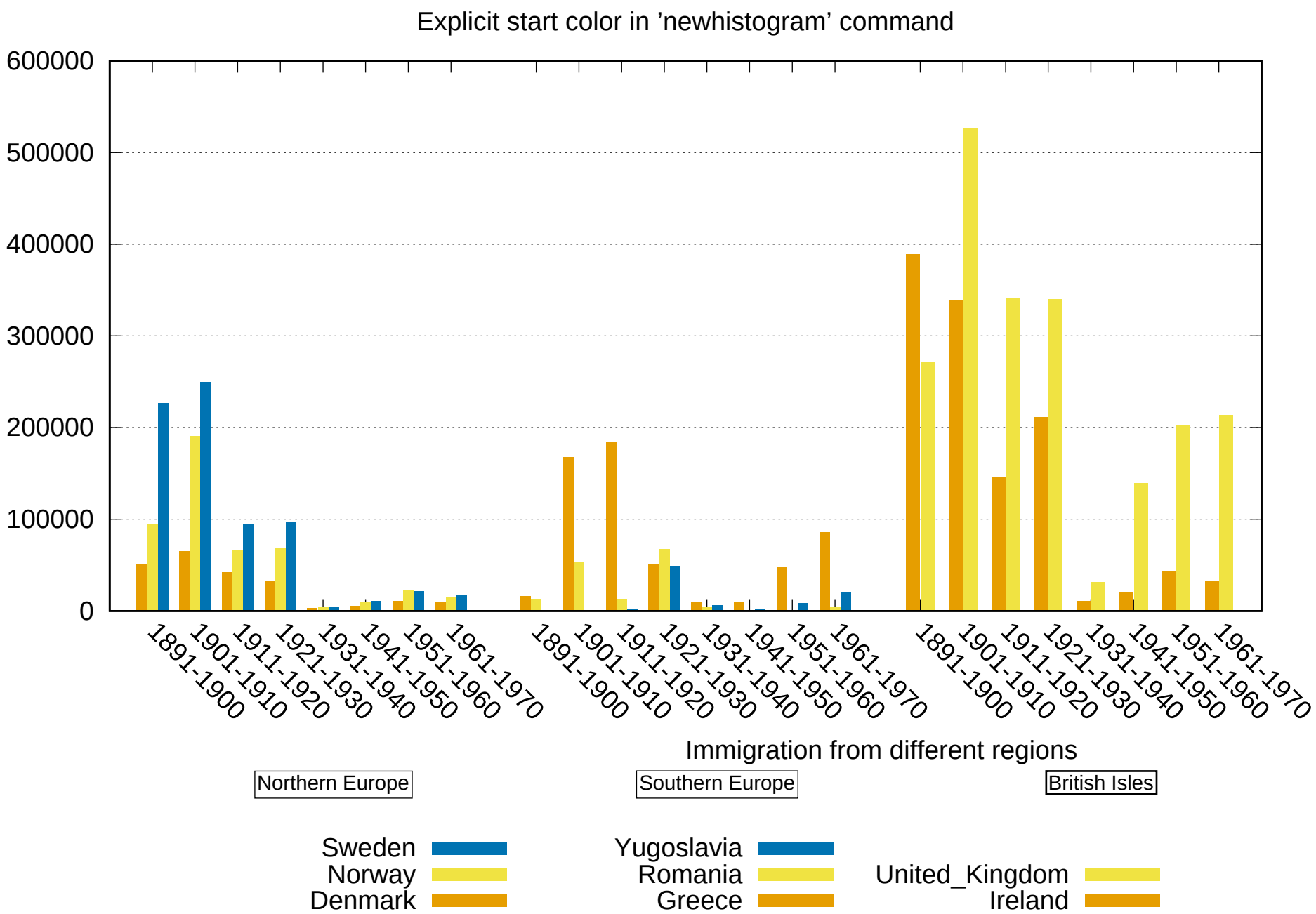

US immigration from Europe by decade
Fraction of total plotted as stacked histogram
% of total
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Other_Europe
Yugoslavia
United_Kingdom
Switzerland
Spain
Soviet_Union
Romania
Portugal
Poland
Sweden
Norway
Netherlands
Italy
Ireland
Greece
Germany
France
Denmark
Czechoslovakia
Belgium
Hungary
Austria

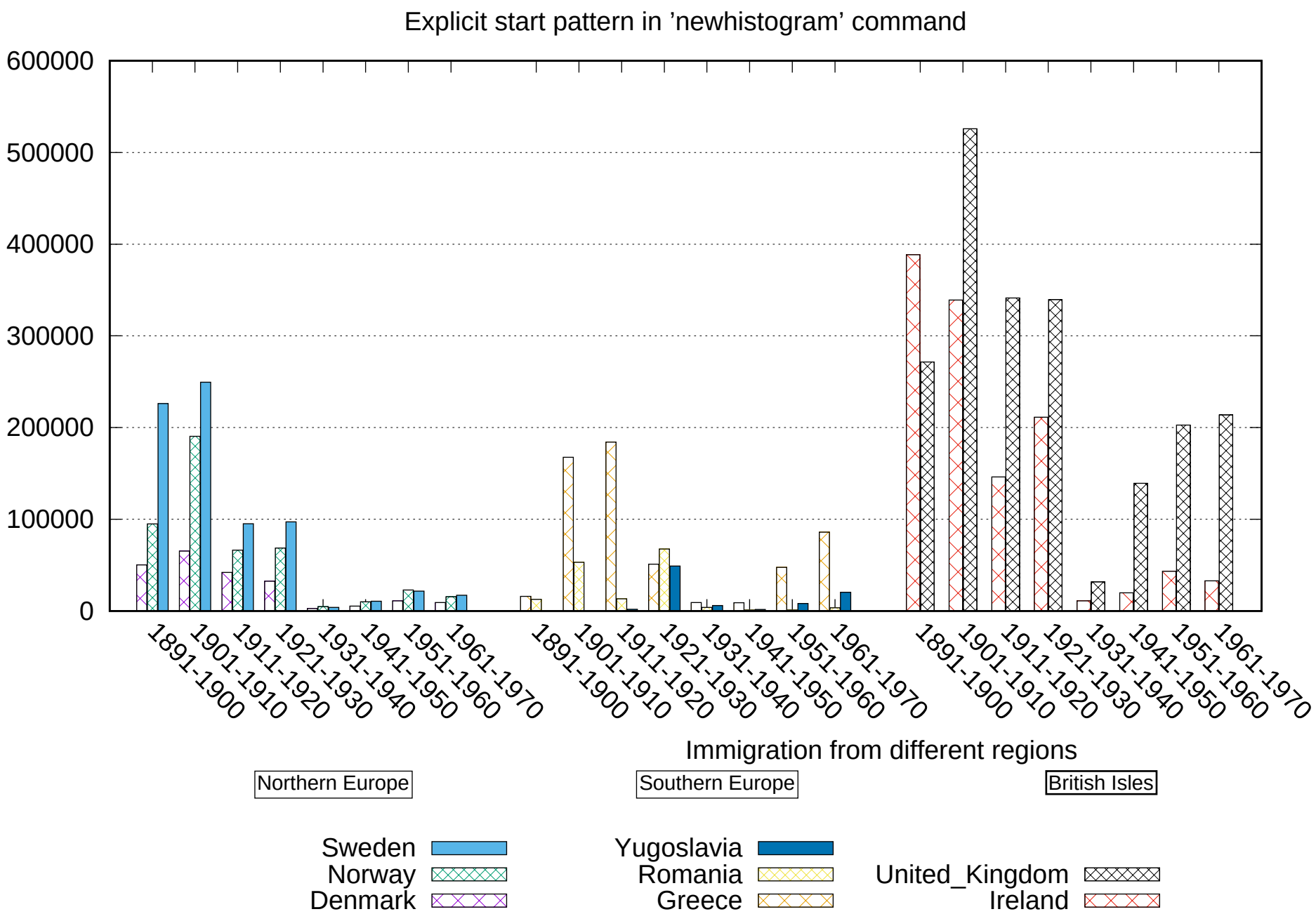

Immigration from Northern Europe
(columstacked histogram)
800000
700000
600000
500000
400000
300000
200000
100000
0
Immigration by decade
Denmark
Netherlands
Norway
Sweden
Country of Origin
1961-1970
1951-1960
1941-1950
1931-1940
1921-1930
1911-1920
1901-1910
1891-1900

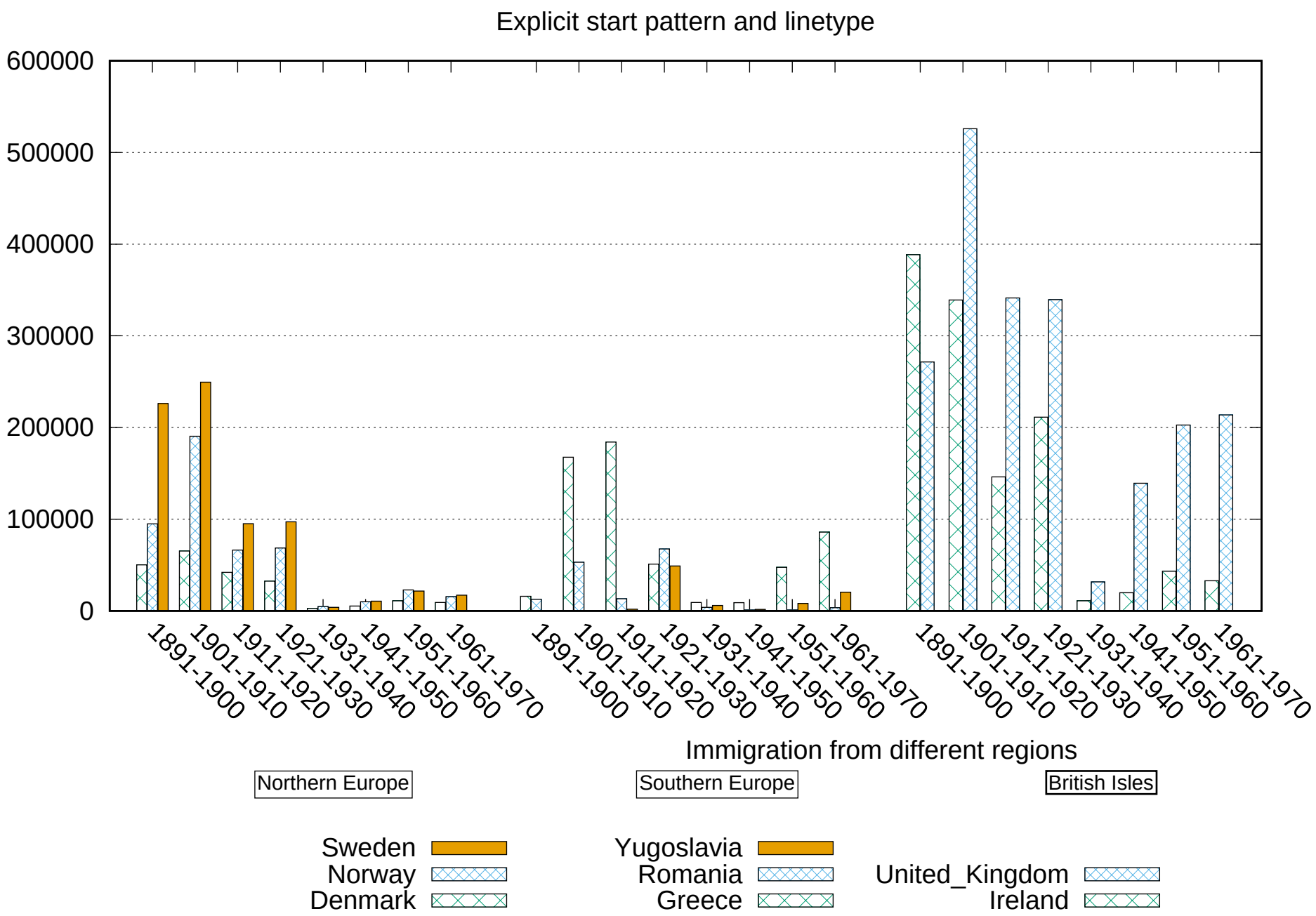

Immigration from different regions
(give each histogram a separate title)
Immigration by decade
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Northern Europe
Sweden
Denmark
Norway
Southern Europe
Greece
Romania
Yugoslavia
British Isles
Ireland
United_Kingdom
(note: histogram titles have specified offset relative to X-axis labe

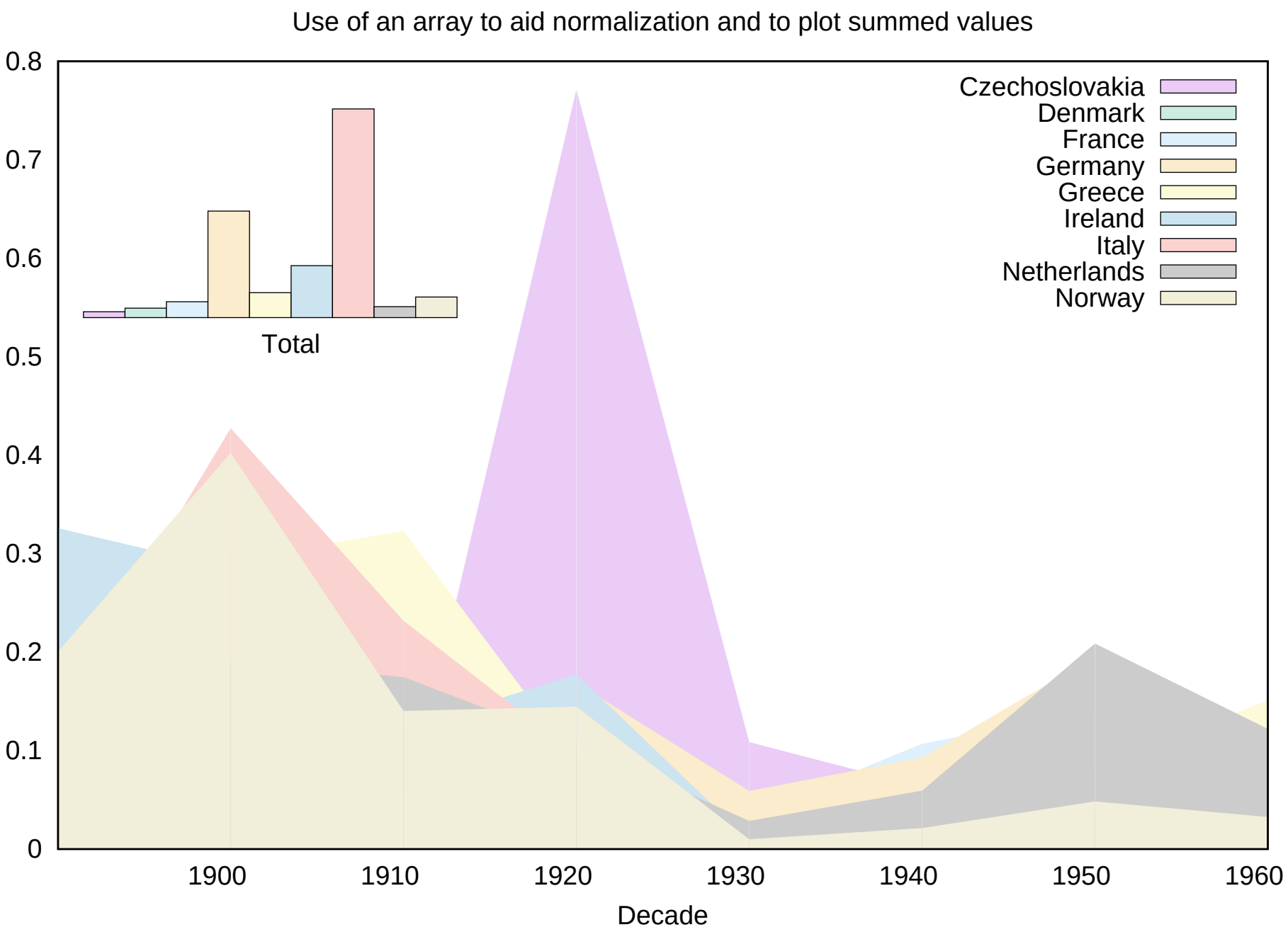

Immigration from different regions
(give each histogram a separate title)
Immigration by decade
900000
800000
700000
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Northern Europe
Sweden
Denmark
Norway
Southern Europe
Greece
Romania
Yugoslavia
British Isles
Ireland
United_Kingdom
(Same plot using rowstacked rather than clustered histogram)

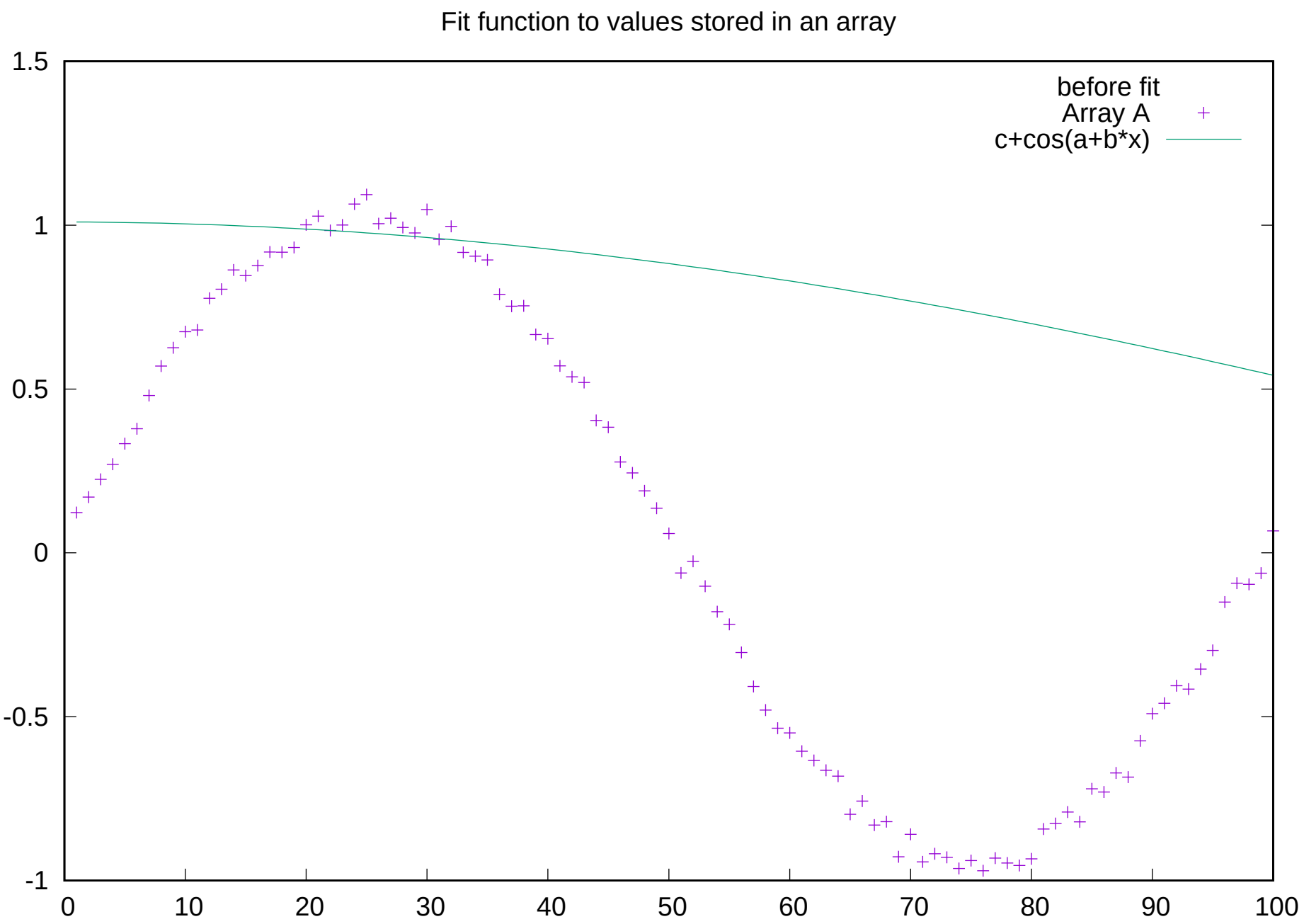

Default Histogram Colouring
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Immigration from different regions
Northern Europe
Southern Europe
British Isles
Sweden
Norway
Denmark
Yugoslavia
Romania
Greece
United_Kingdom
Ireland

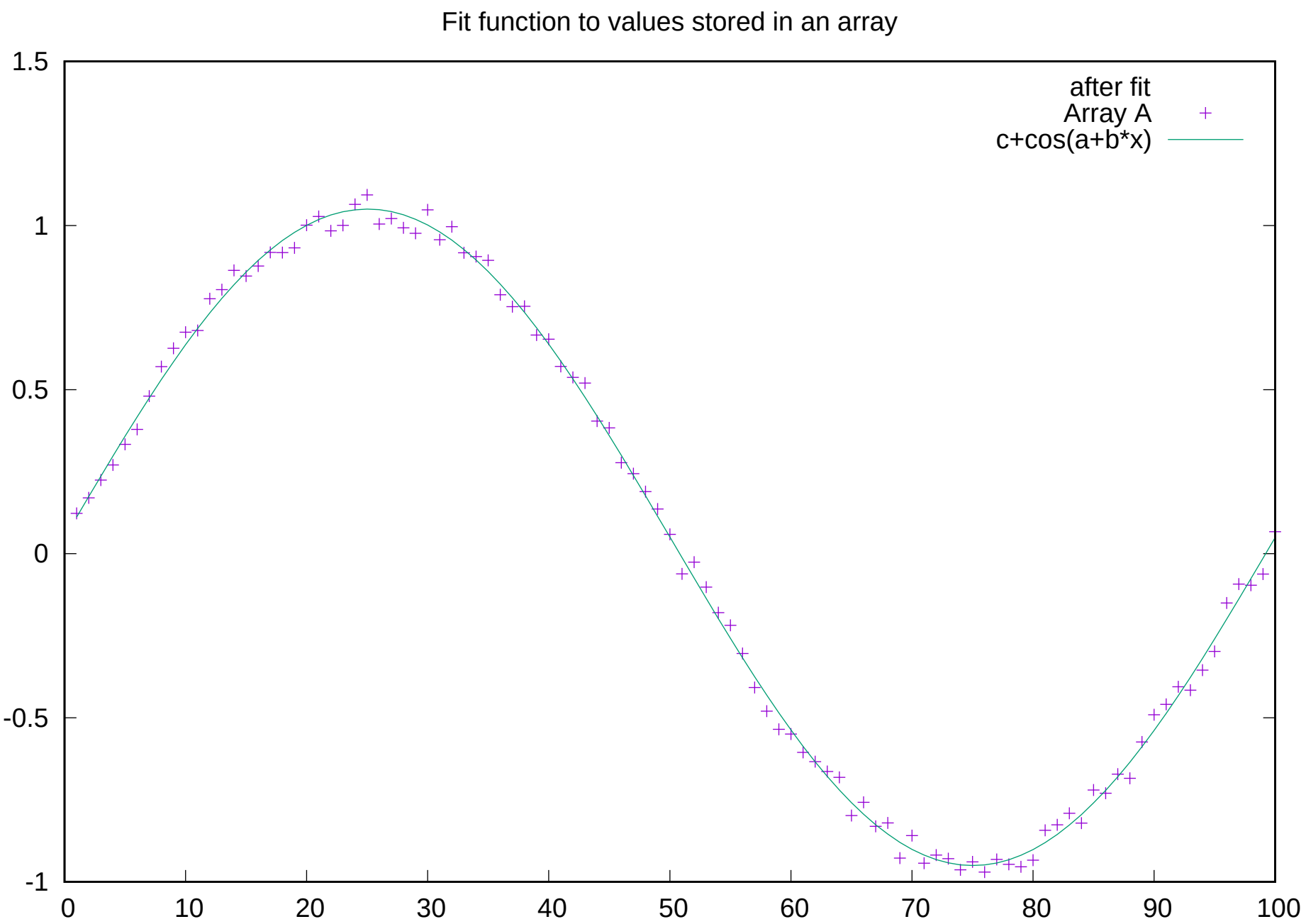

Explicit start color in 'newhistogram' command
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Immigration from different regions
Northern Europe
Southern Europe
British Isles
Sweden
Norway
Denmark
Yugoslavia
Romania
Greece
United_Kingdom
Ireland

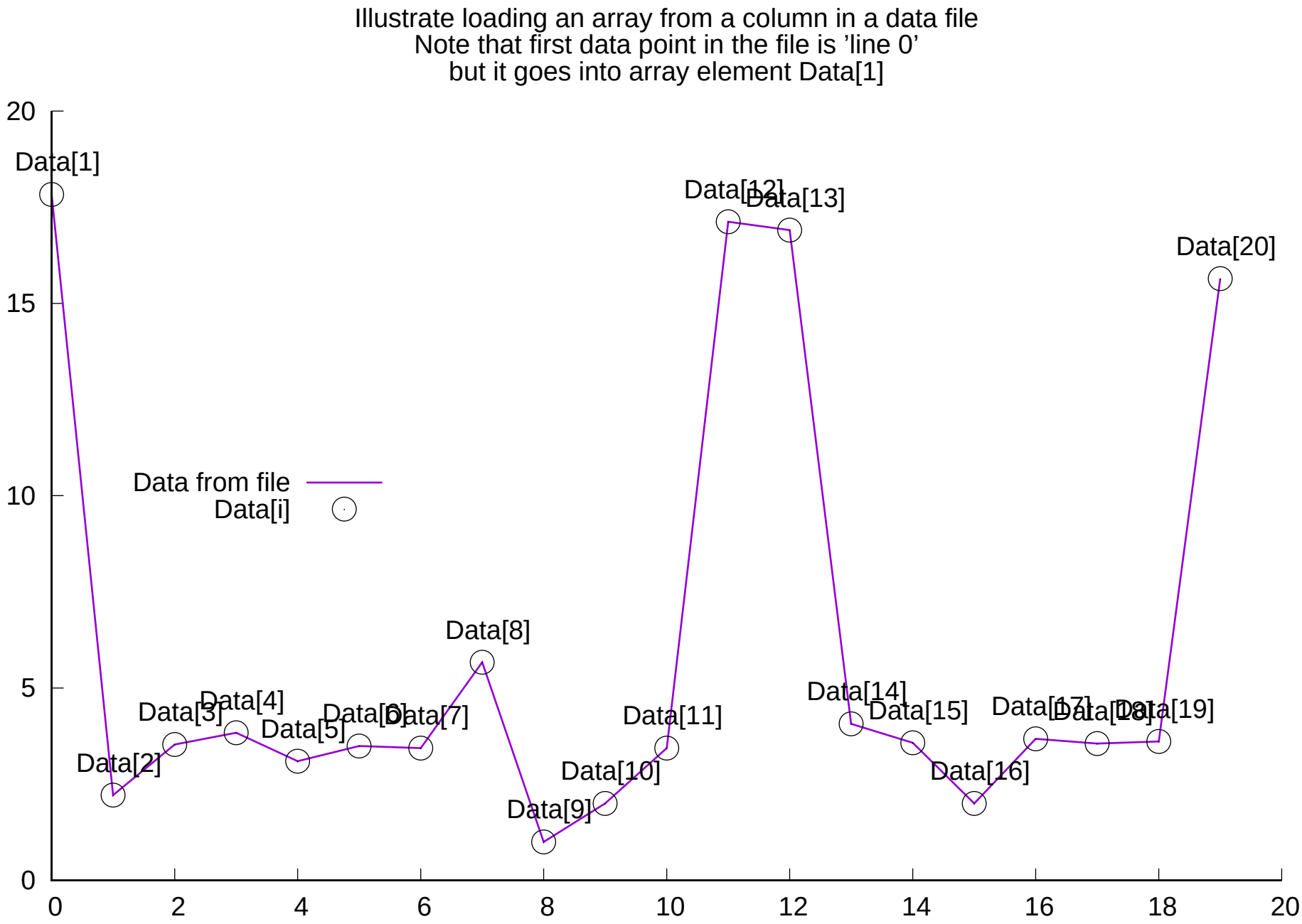

Explicit start pattern in 'newhistogram' command
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Immigration from different regions
Northern Europe
Southern Europe
British Isles
Sweden
Norway
Denmark
Yugoslavia
Romania
Greece
United_Kingdom
Ireland


Explicit start pattern and linetype
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Immigration from different regions
Northern Europe
Southern Europe
British Isles
Sweden
Norway
Denmark
Yugoslavia
Romania
Greece
United_Kingdom
Ireland

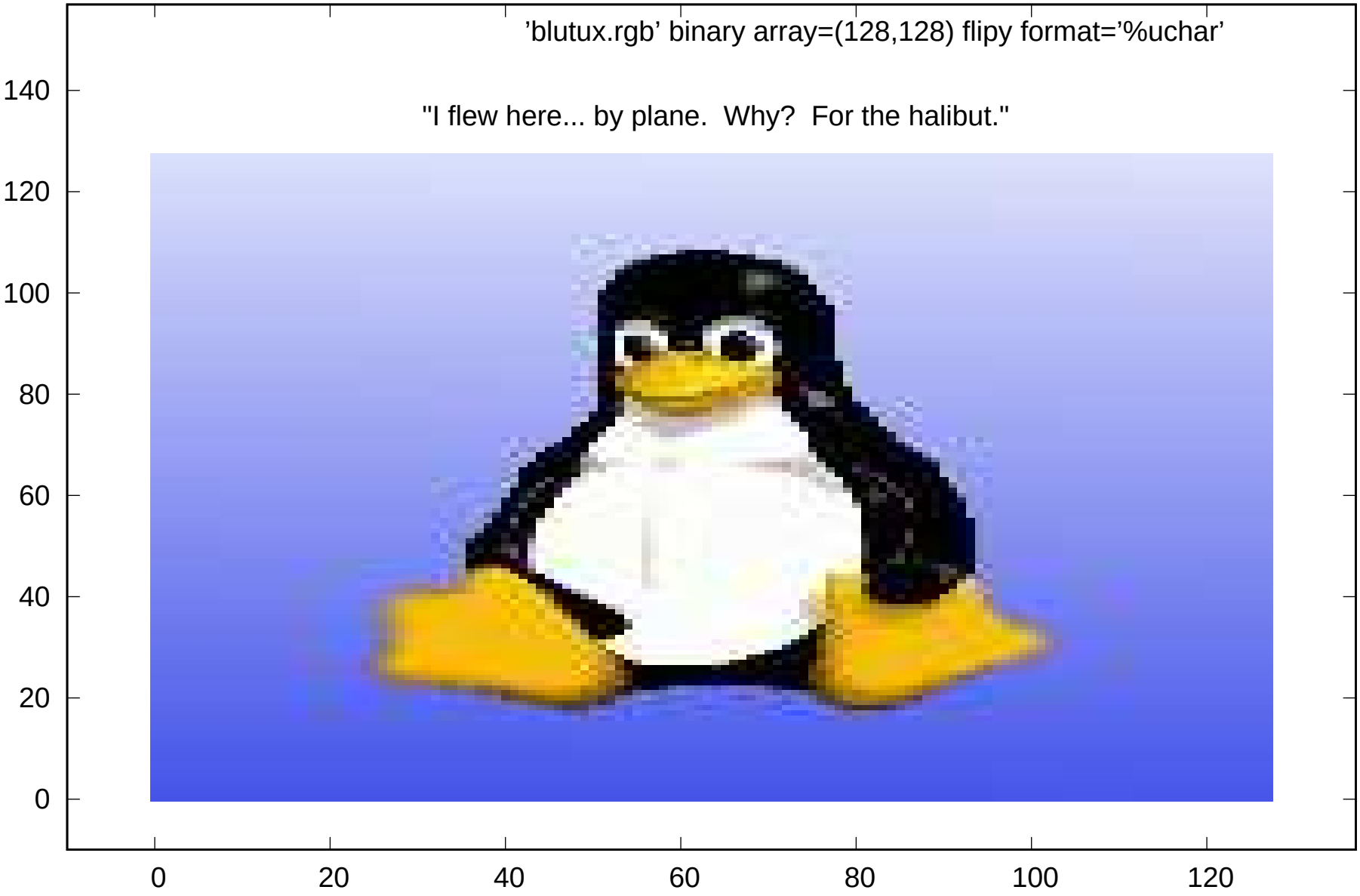

Use of an array to aid normalization and to plot summed values
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1900
1910
1920
1930
1940
1950
1960
Decade
Czechoslovakia
Denmark
France
Germany
Greece
Ireland
Italy
Netherlands
Norway
Total

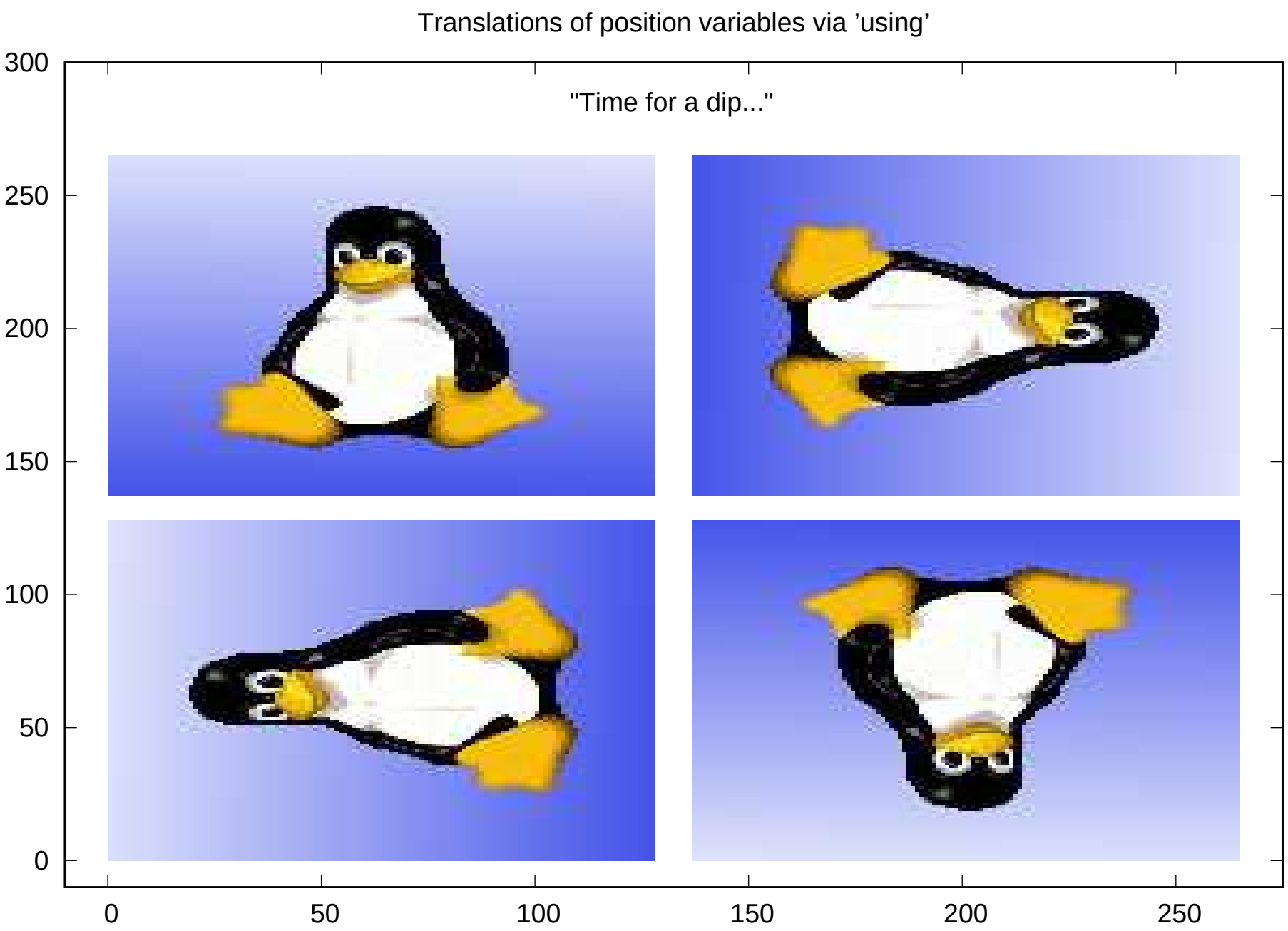

Fit function to values stored in an array
before fit
Array A
c+cos(a+b*x)
1.5
1
0.5
0
-0.5
-1
0
10
20
30
40
50
60
70
80
90
100


Fit function to values stored in an array
after fit
Array A
c+cos(a+b*x)
1.5
1
0.5
0
-0.5
-1
0
10
20
30
40
50
60
70
80
90
100

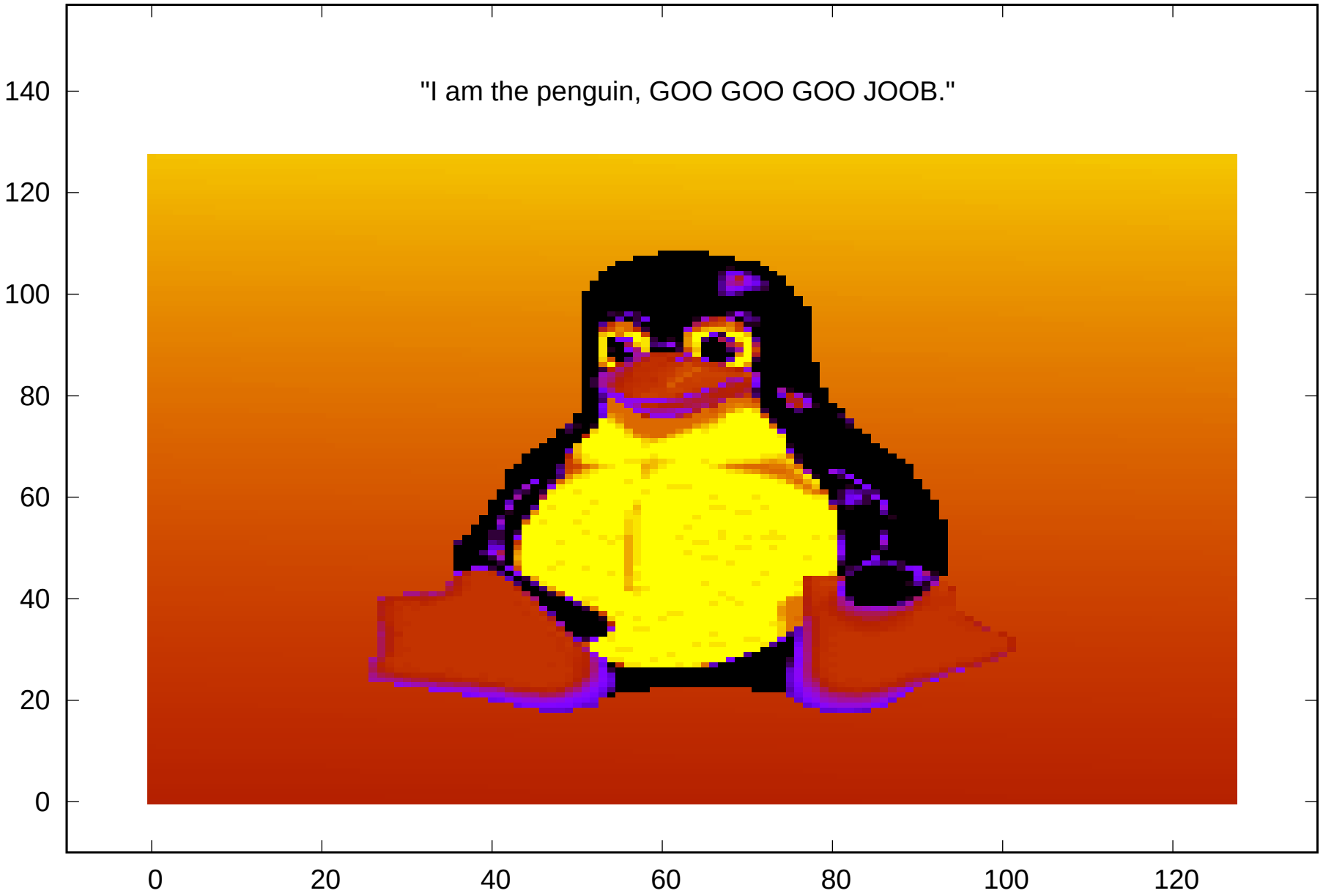

Illustrate loading an array from a column in a data file
Note that first data point in the file is 'line 0'
but it goes into array element Data[1]
Data from file
Data[i]
Data[1]
Data[2]
Data[3]
Data[4]
Data[5]
Data[6]
Data[7]
Data[8]
Data[9]
Data[10]
Data[11]
Data[12]
Data[13]
Data[14]
Data[15]
Data[16]
Data[17]
Data[18]
Data[19]
Data[20]
0
5
10
15
20
0
2
4
6
8
10
12
14
16
18
20

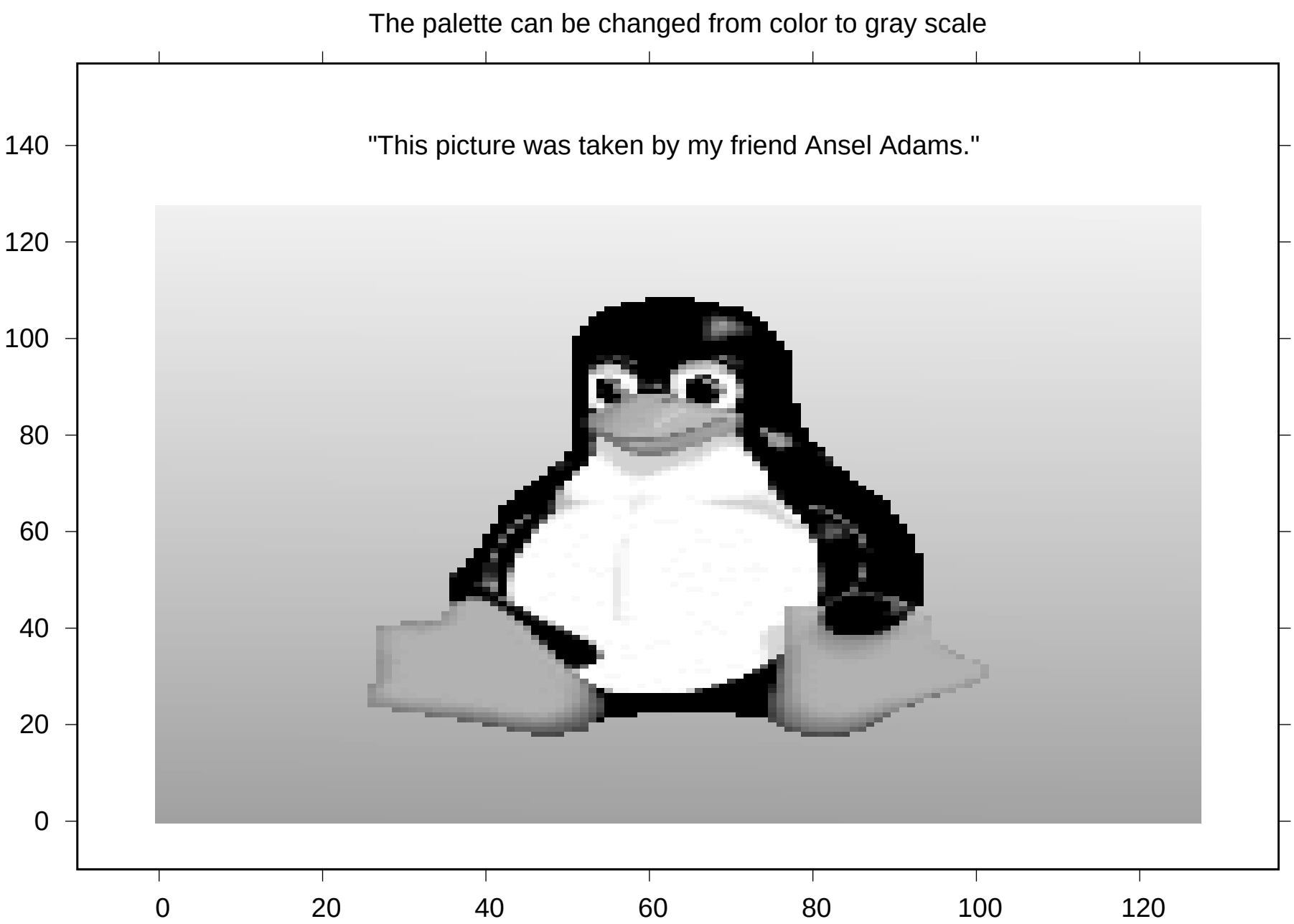
Larry Ewing’s GIMP penguin on vacation basking in
the balmy waters off the coast of Murmansk

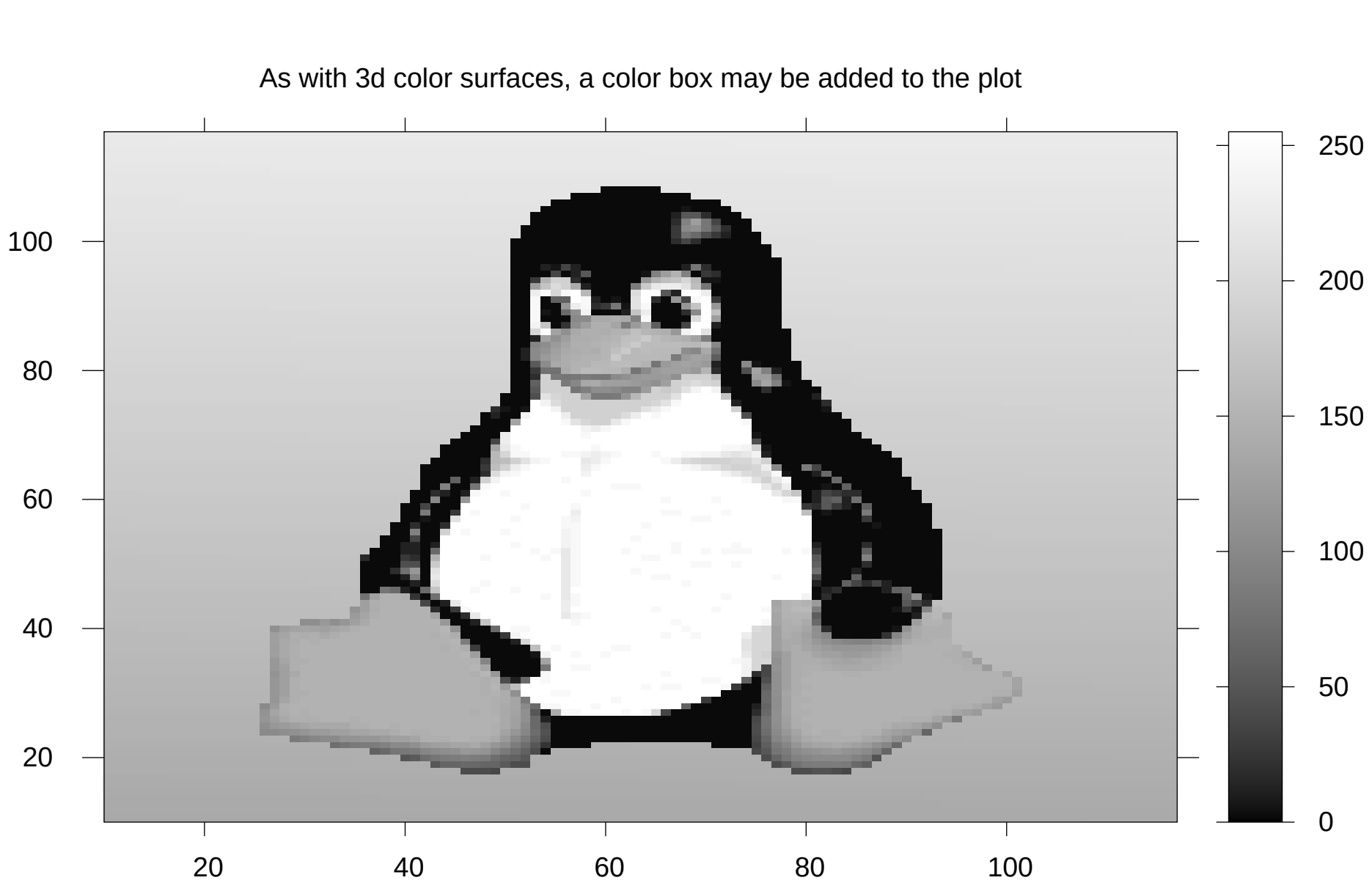


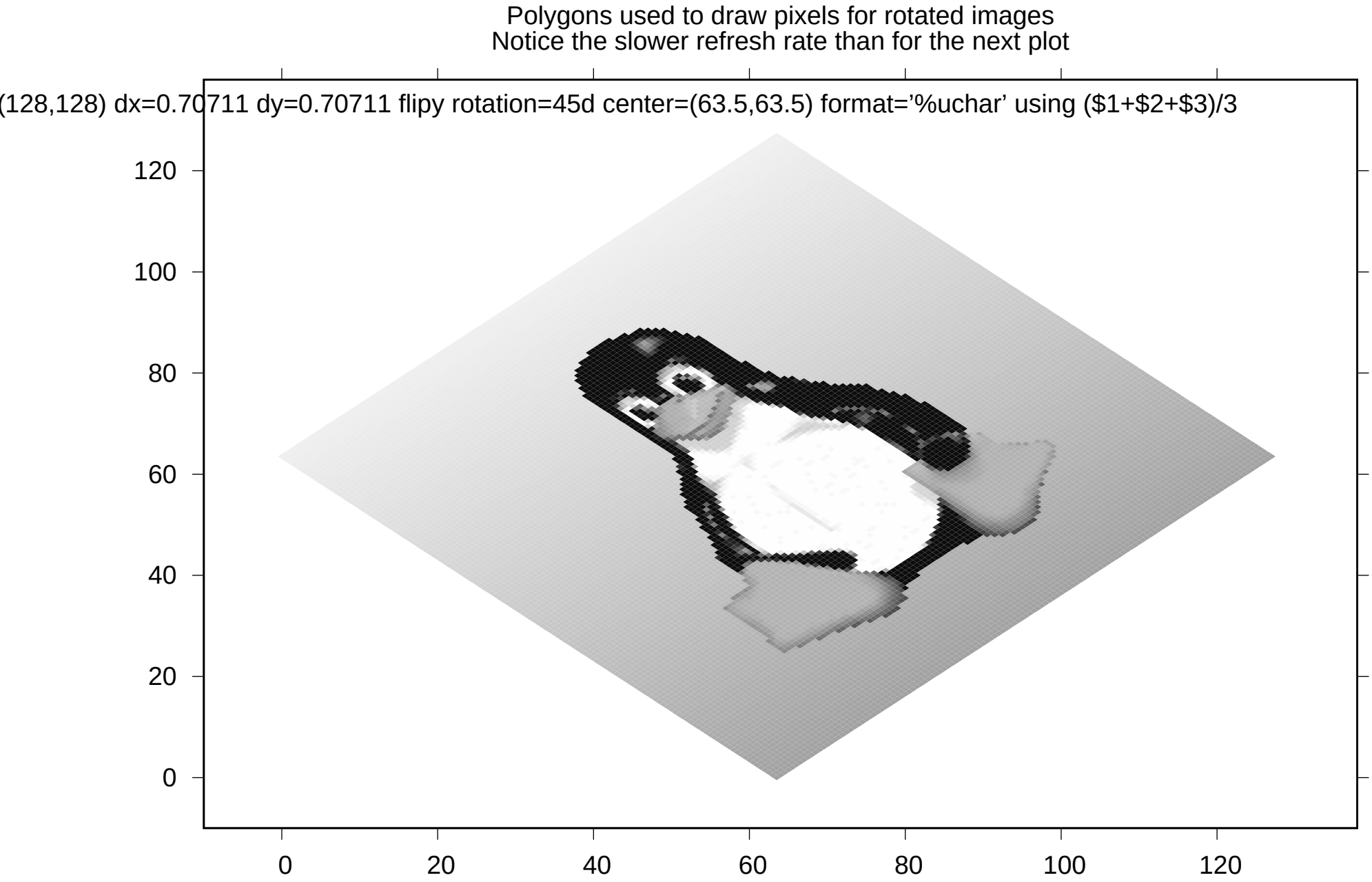

Translations of position variables via ’using’
"Time for a dip..."
300
250
200
150
100
50
0
0
50
100
150
200
250

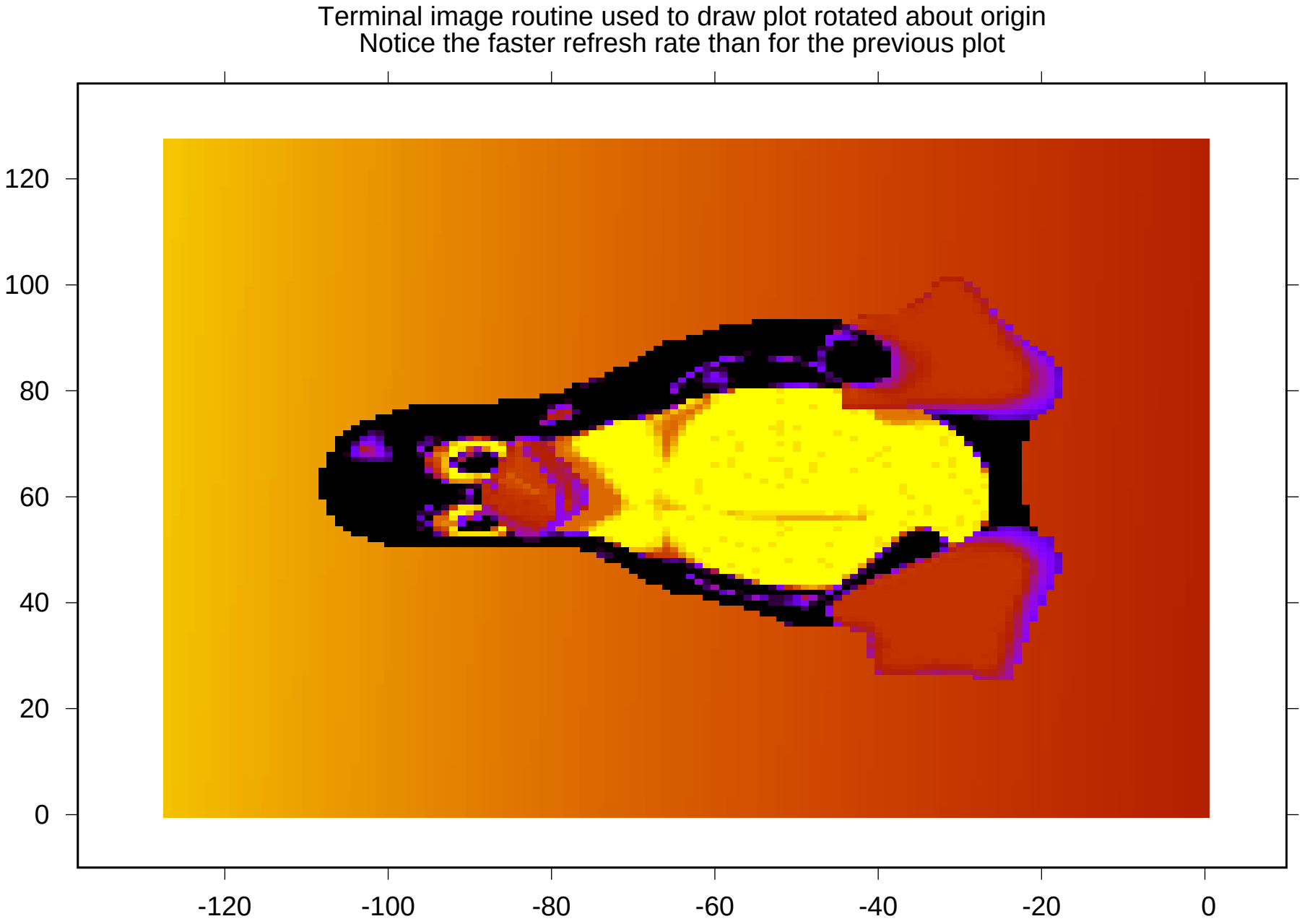
Palette mode ’image’ used to produce psychedelic bird



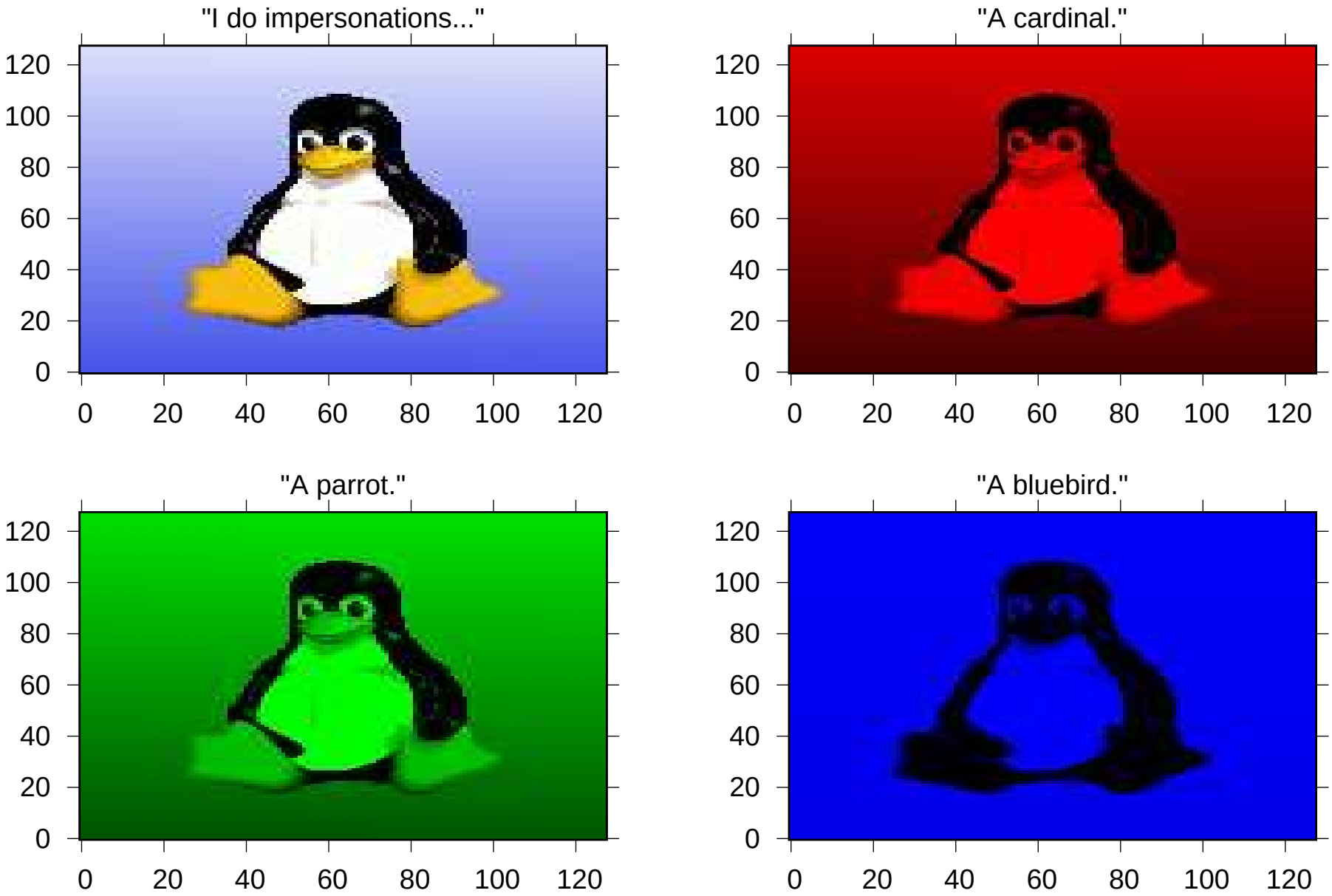

The palette can be changed from color to gray scale
"This picture was taken by my friend Ansel Adams."
140
120
100
80
60
40
20
0
0
20
40
60
80
100
120


As with 3d color surfaces, a color box may be added to the plot
100
80
60
40
20
20
40
60
80
100
250
200
150
100
50
0

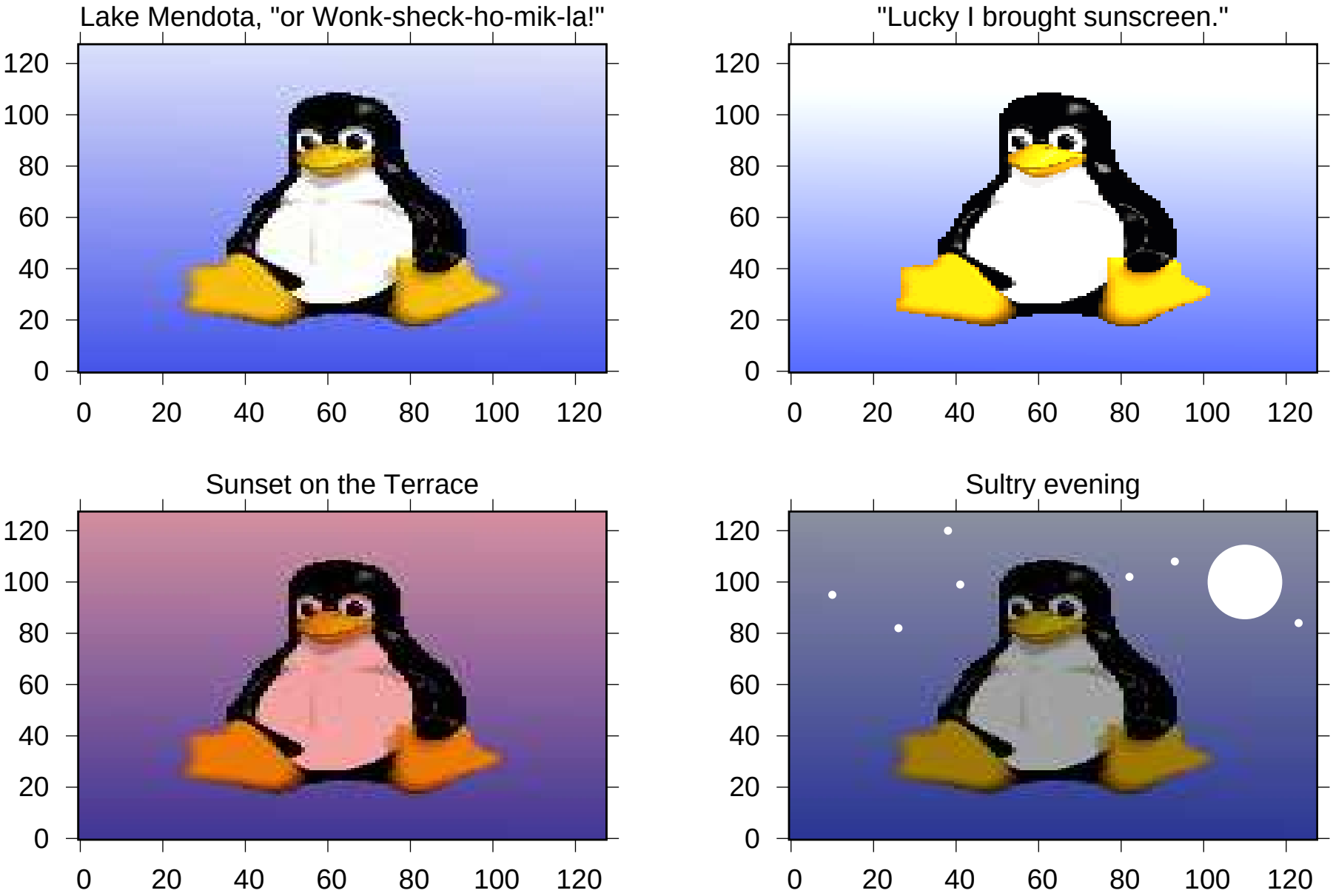

Polygons used to draw pixels for rotated images
Notice the slower refresh rate than for the next plot
(128,128) dx=0.70711 dy=0.70711 flipy rotation=45d center=(63.5,63.5) format='%uchar' using ($1+$2+$3)/3
120
100
80
60
40
20
0
0
20
40
60
80
100
120

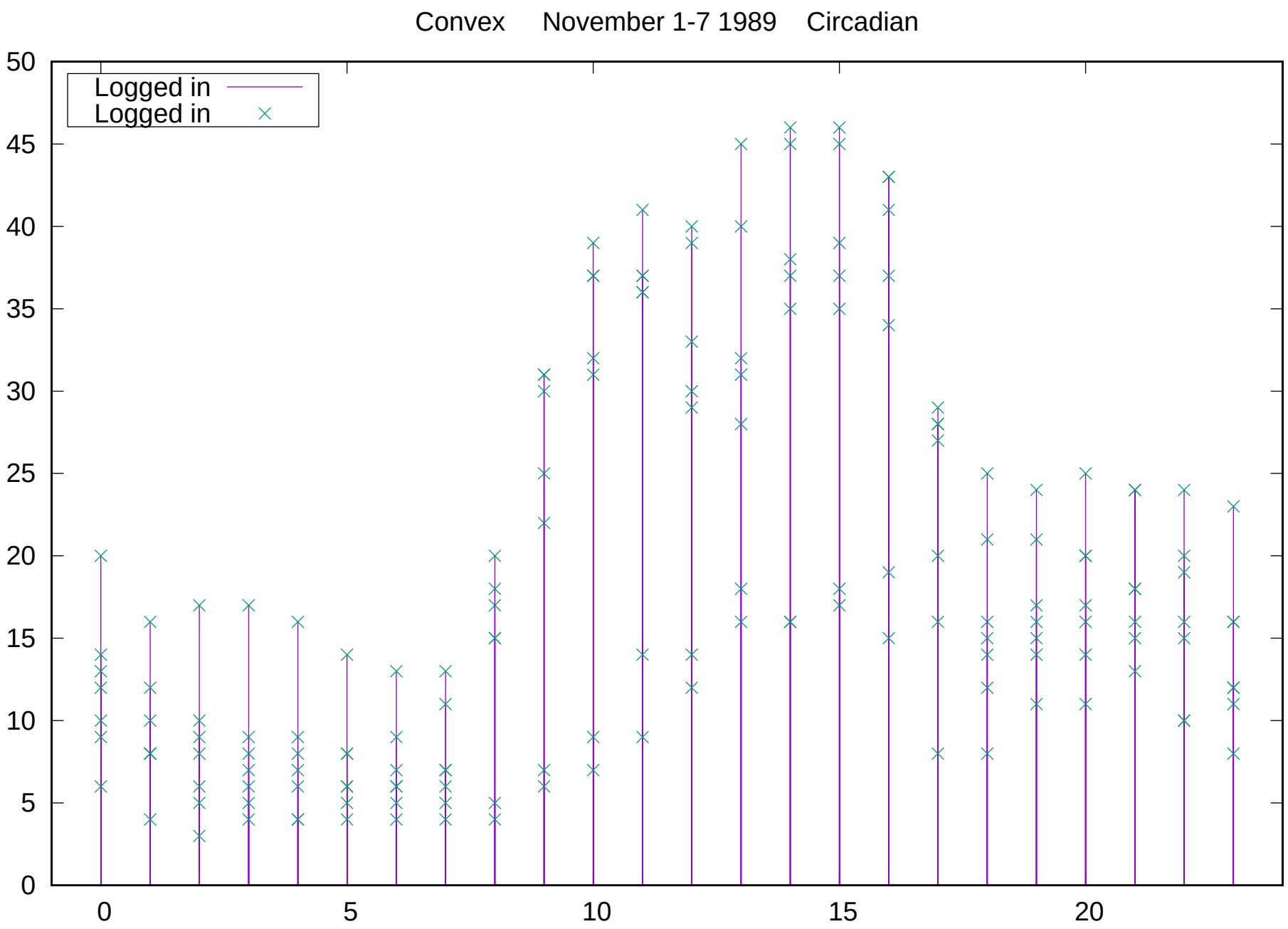

Terminal image routine used to draw plot rotated about origin
Notice the faster refresh rate than for the previous plot
120
100
80
60
40
20
0
-120
-100
-80
-60
-40
-20
0

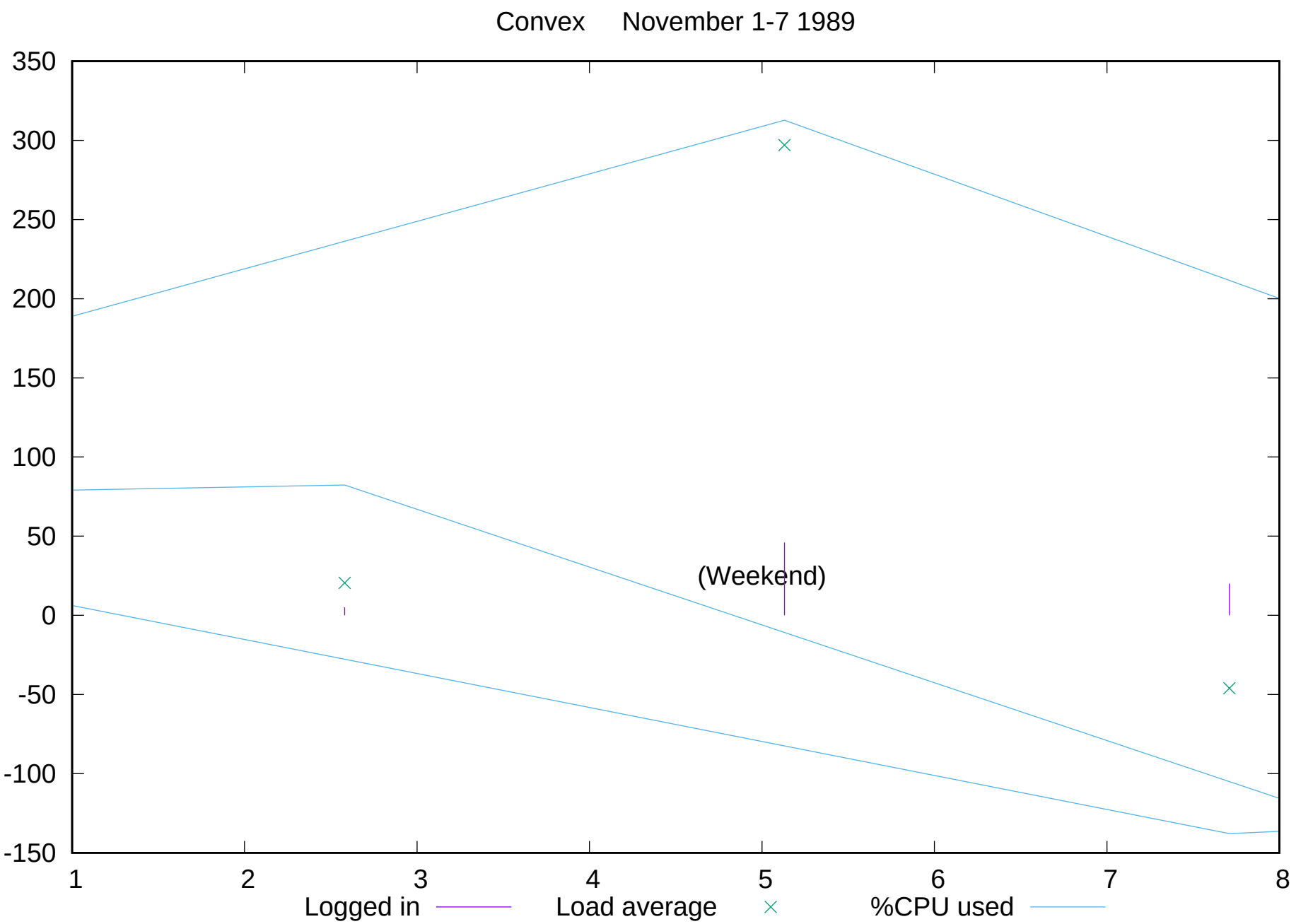
Selection of the input channels via ‘using‘

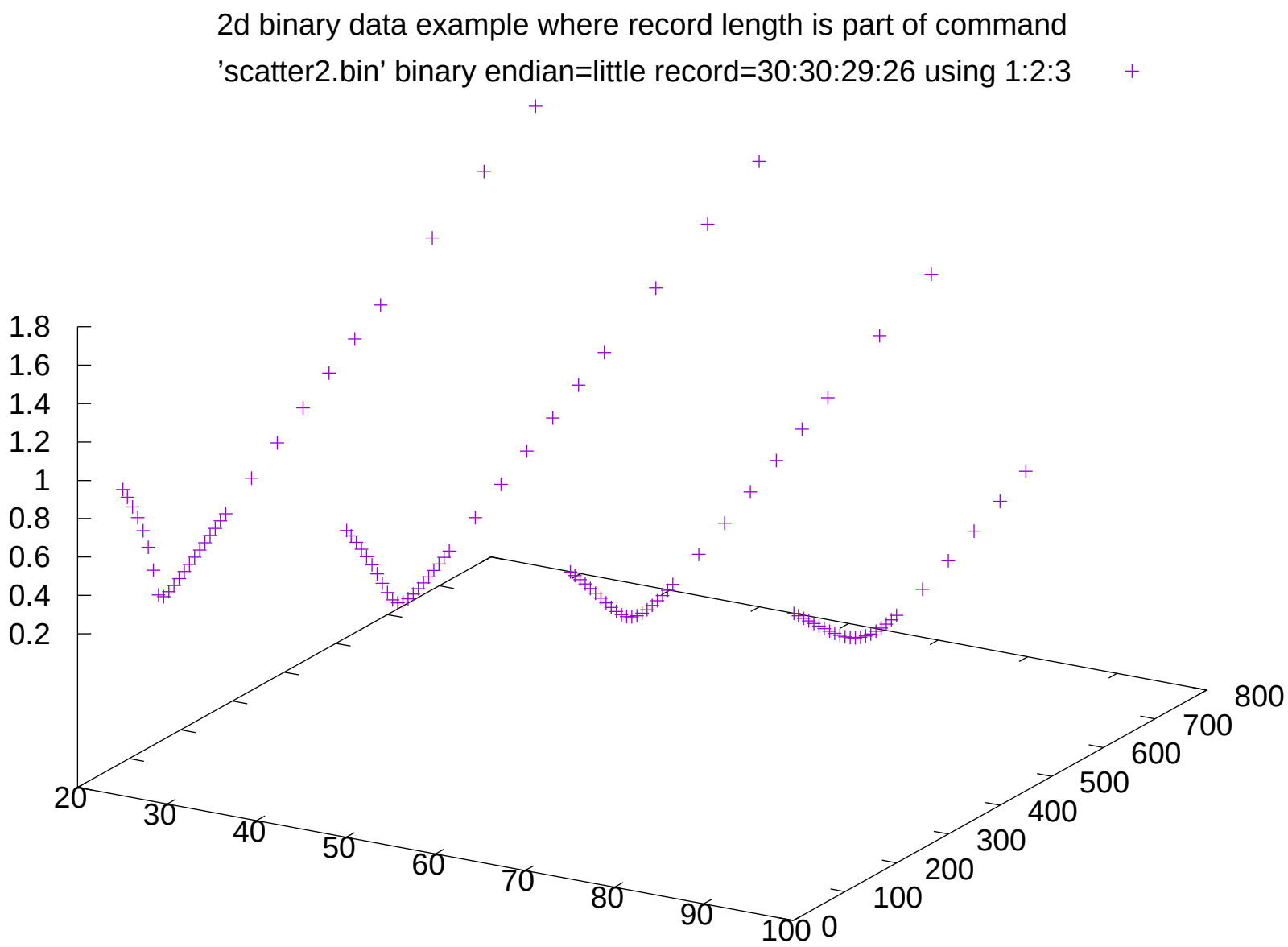


Luminance adjustment via ‘cbrange‘

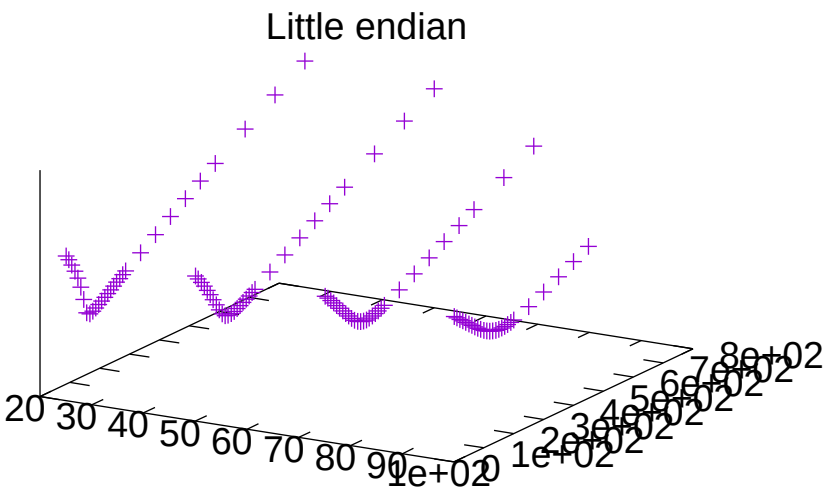


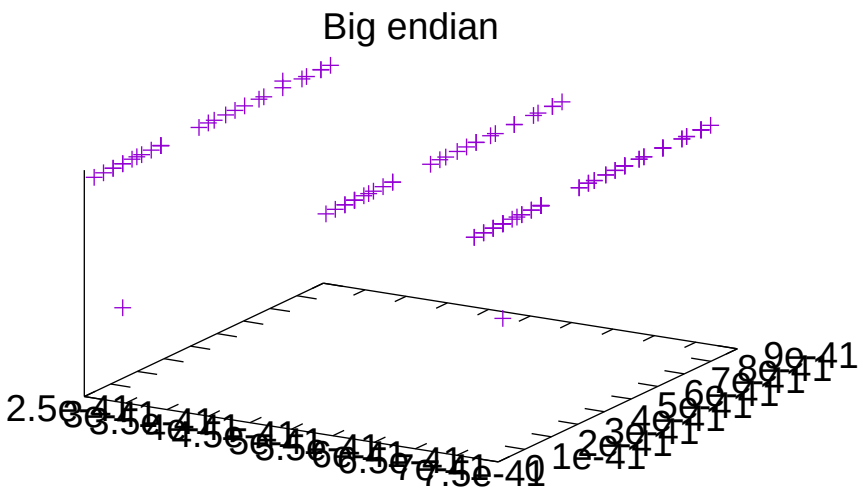

Convex November 1-7 1989 Circadian
Logged in
Logged in
50
45
40
35
30
25
20
15
10
5
0
0
5
10
15
20

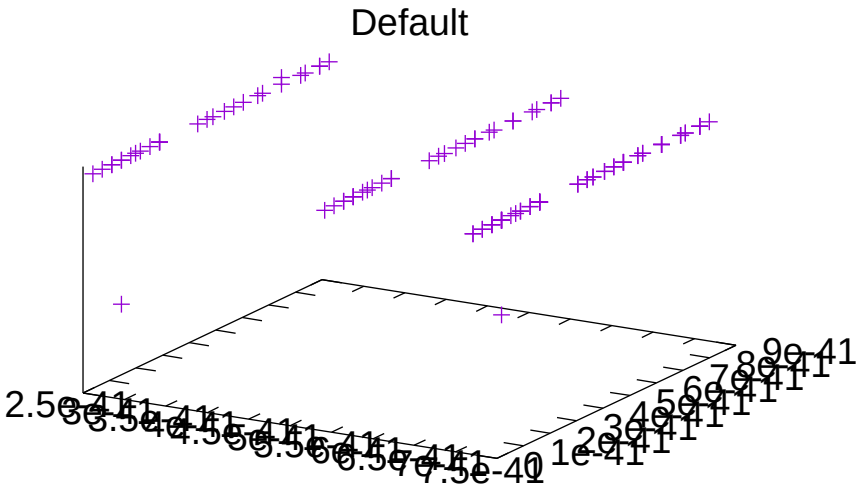

Convex November 1-7 1989
350
300
250
200
150
100
50
0
-50
-100
-150
1
2
3
4
5
6
7
8
(Weekend)
Logged in
Load average
%CPU used

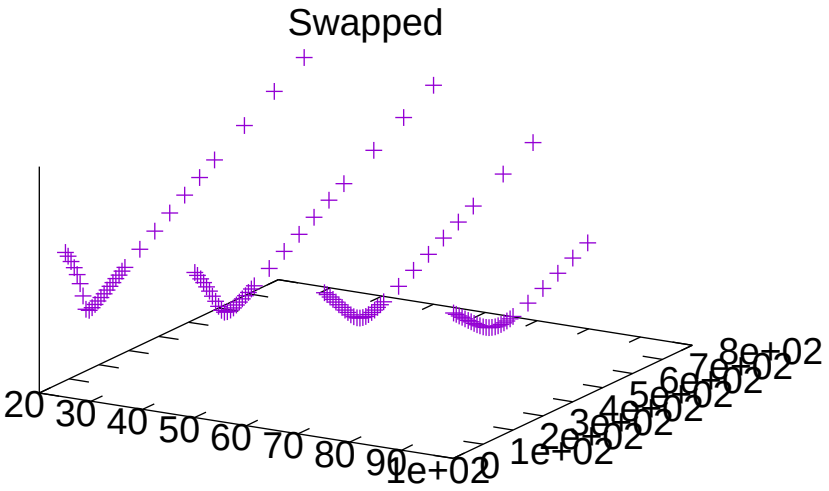

2d binary data example where record length is part of command
'scatter2.bin' binary endian=little record=30:30:29:26 using 1:2:3
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500
600
700
800

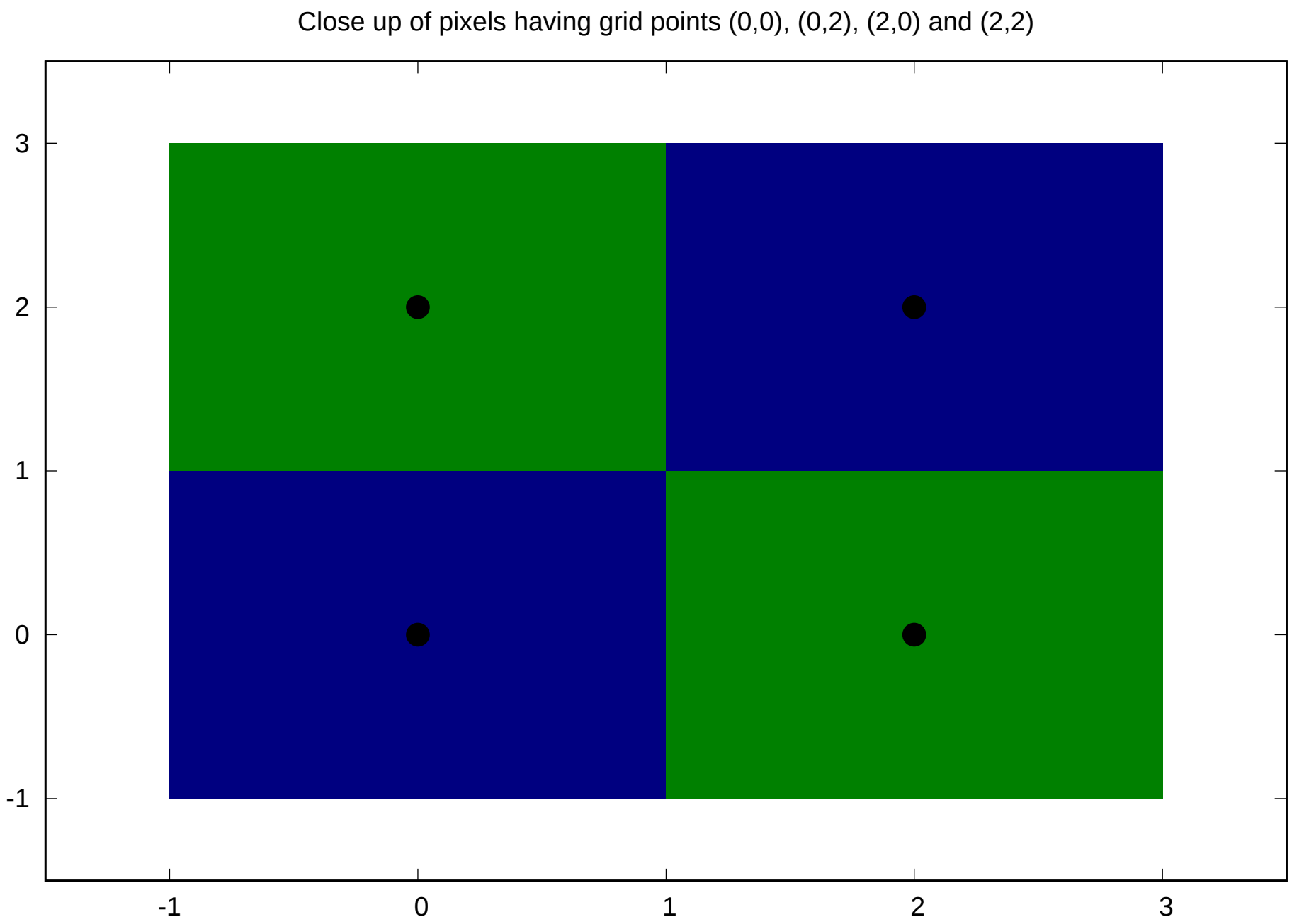
If plots in columns match, your compiler is little endian





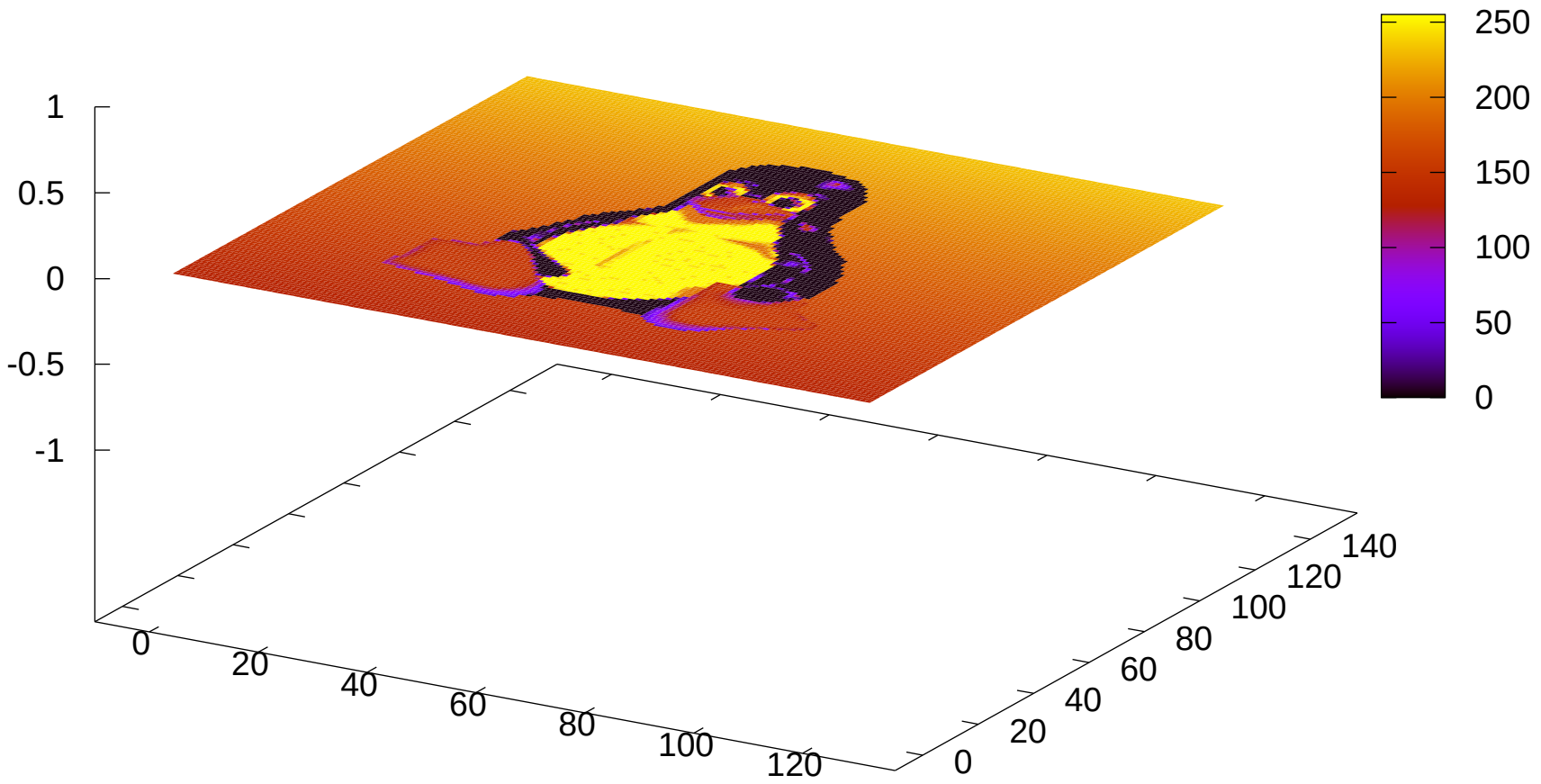





Close up of pixels having grid points (0,0), (0,2), (2,0) and (2,2)
3
2
1
0
-1
-1
0
1
2
3

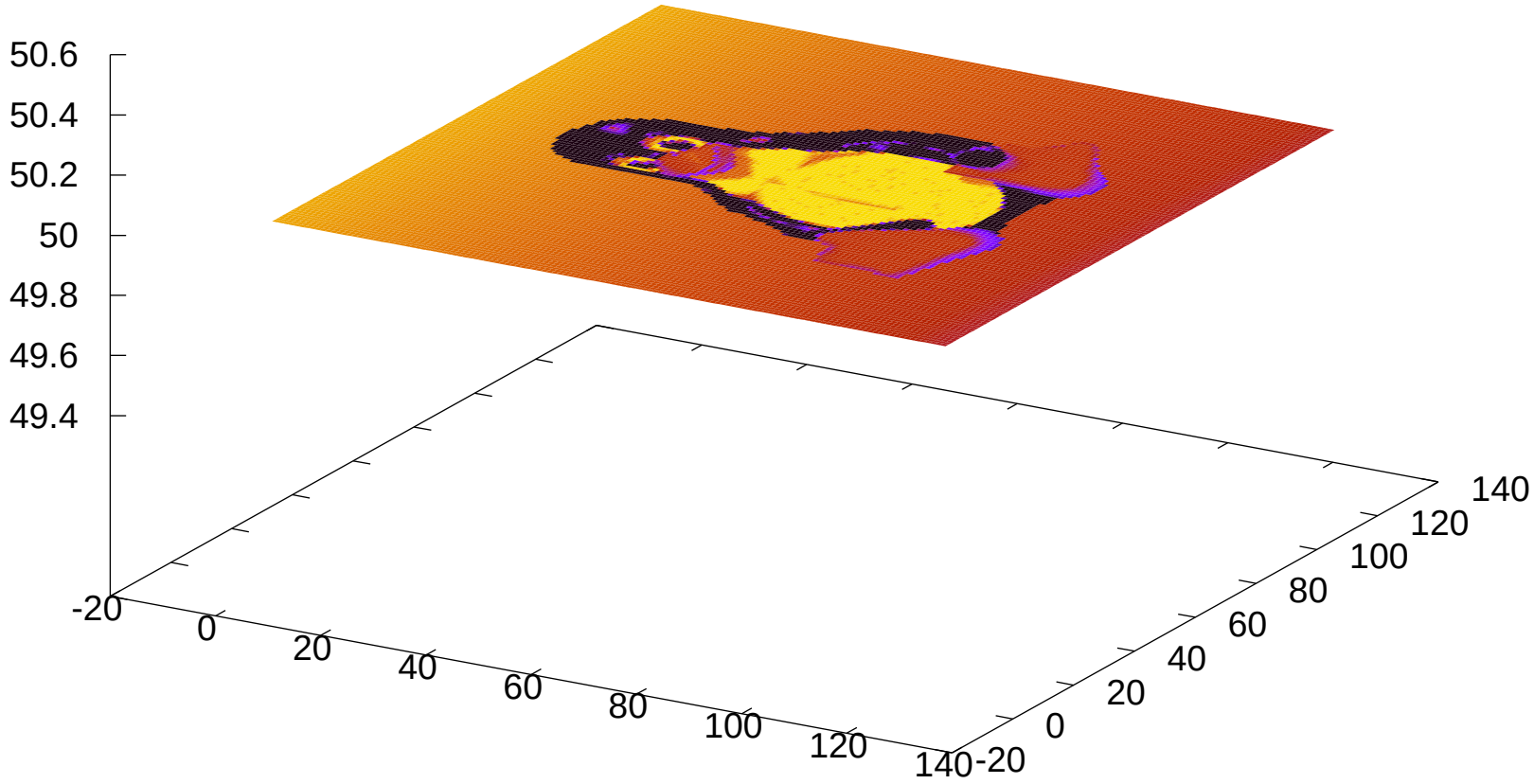
Simple extension of a two dimensional image into three dimensions

'blutux.rgb' binary array=(128,128) flip=y format='%uchar%uchar%uchar' using ($1+$2+$3)/3



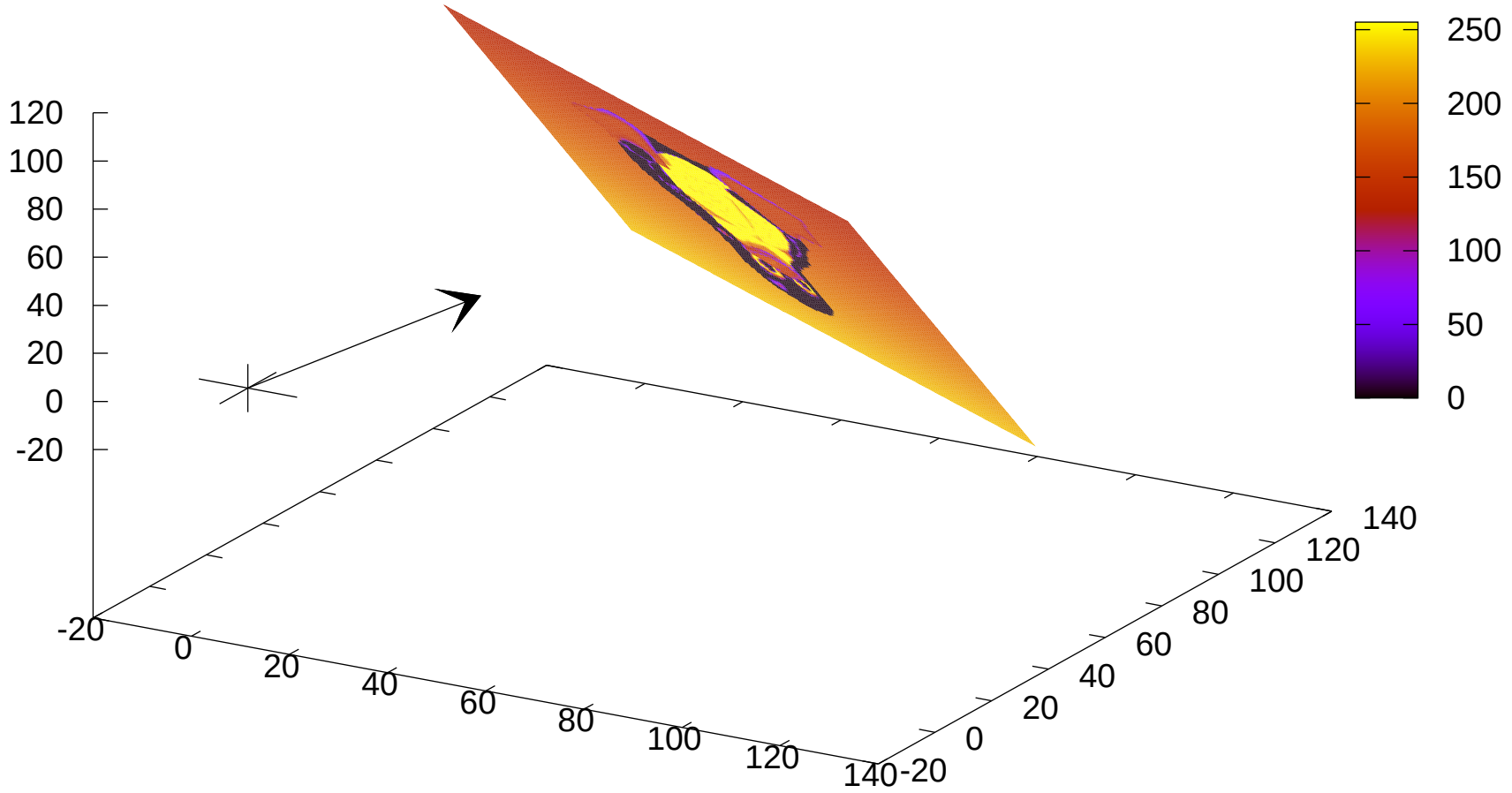
Orientation operations from ’plot’ also apply to to ’splot’

128,128) flipy rotate=90d center = (63.5,63.5,50) format=’%uchar%uchar%uchar’ using ($1+$2+$3)

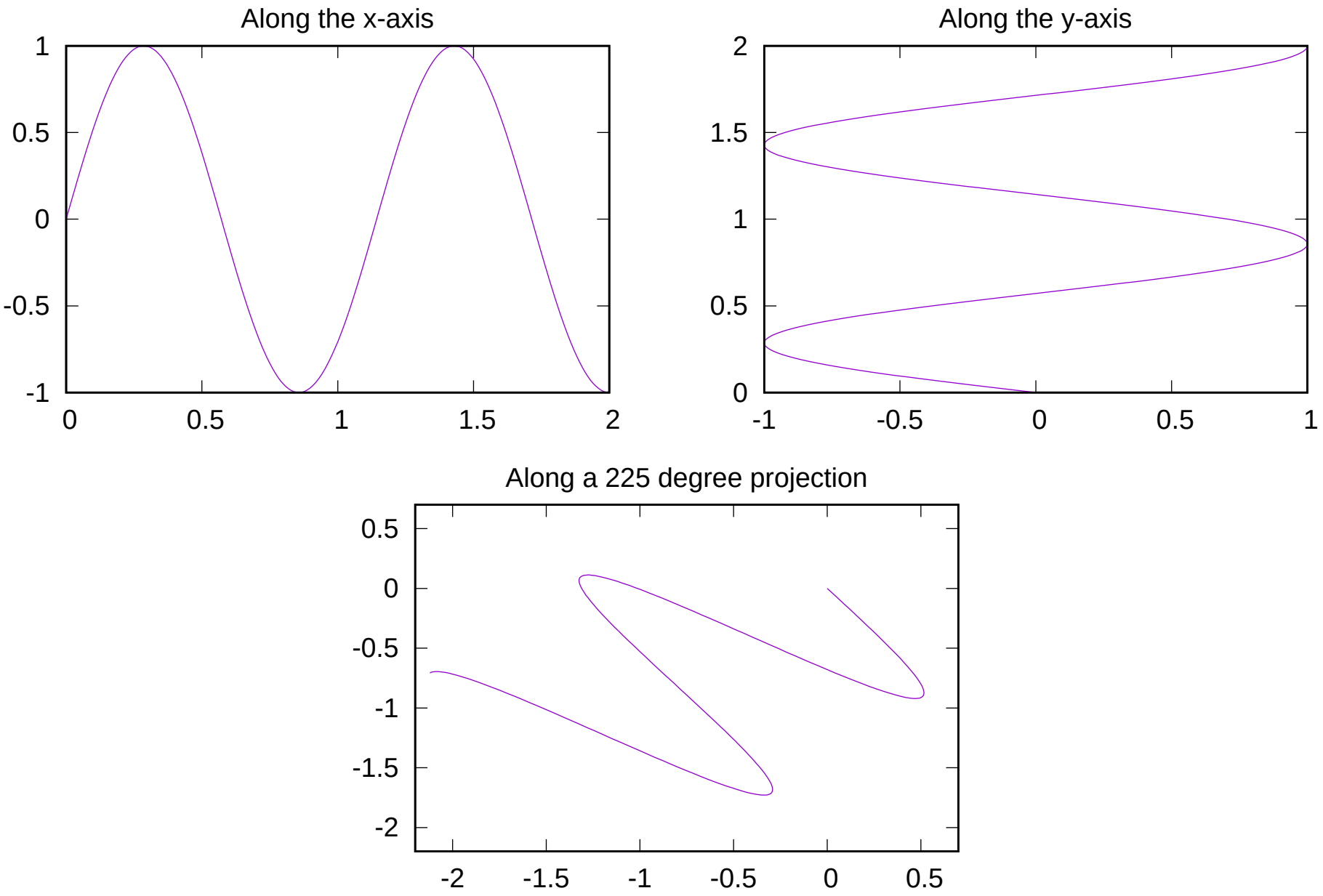


The key word ’perpendicular’ applies only to ’splot’

rot=1.0pi center = (63.5,63.5,50) perp=(1,1,1) format=’%uchar%uchar%uchar’ using ($1+$2+$3)/3

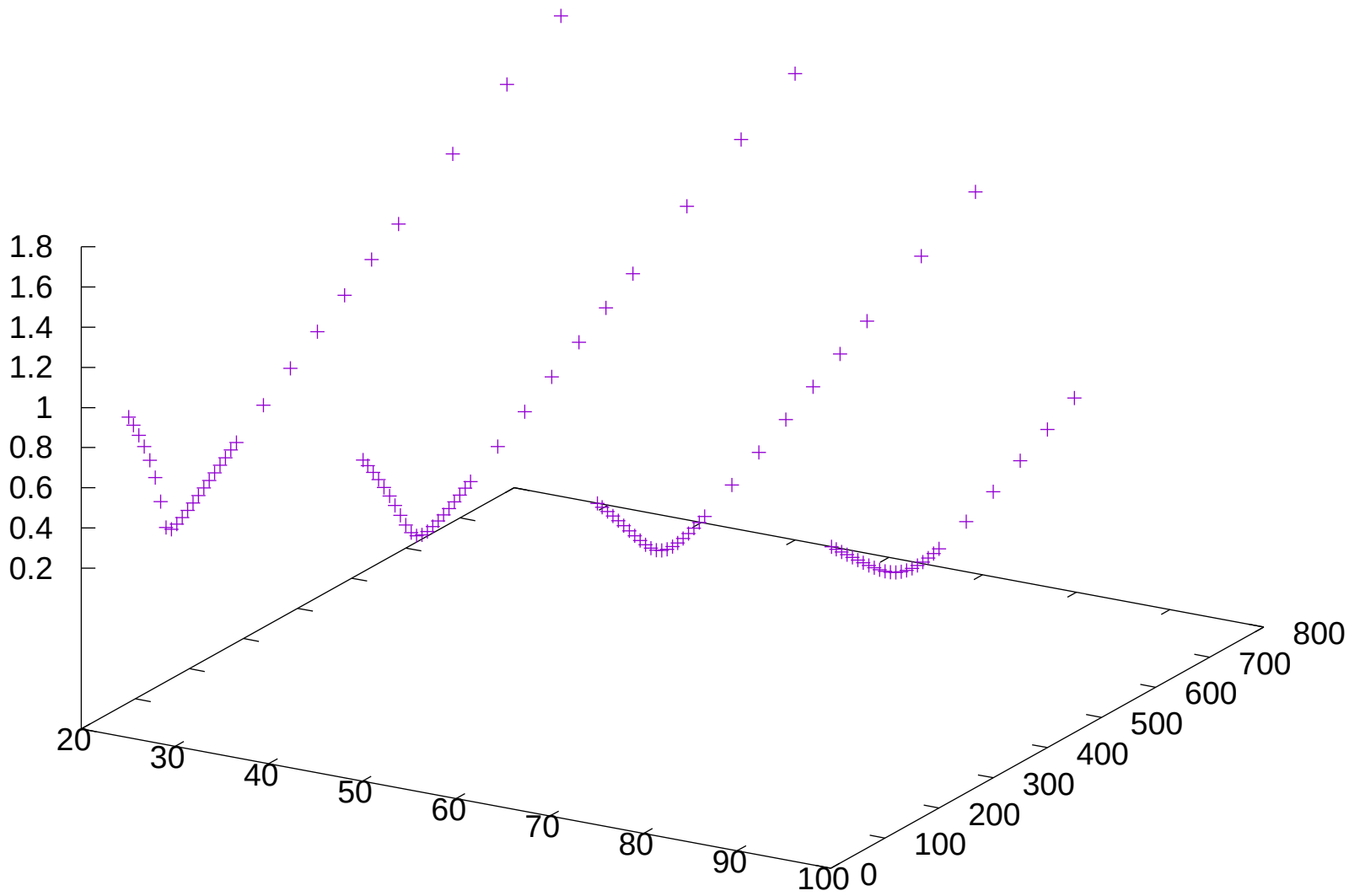


Temporal data having one generated coordinate



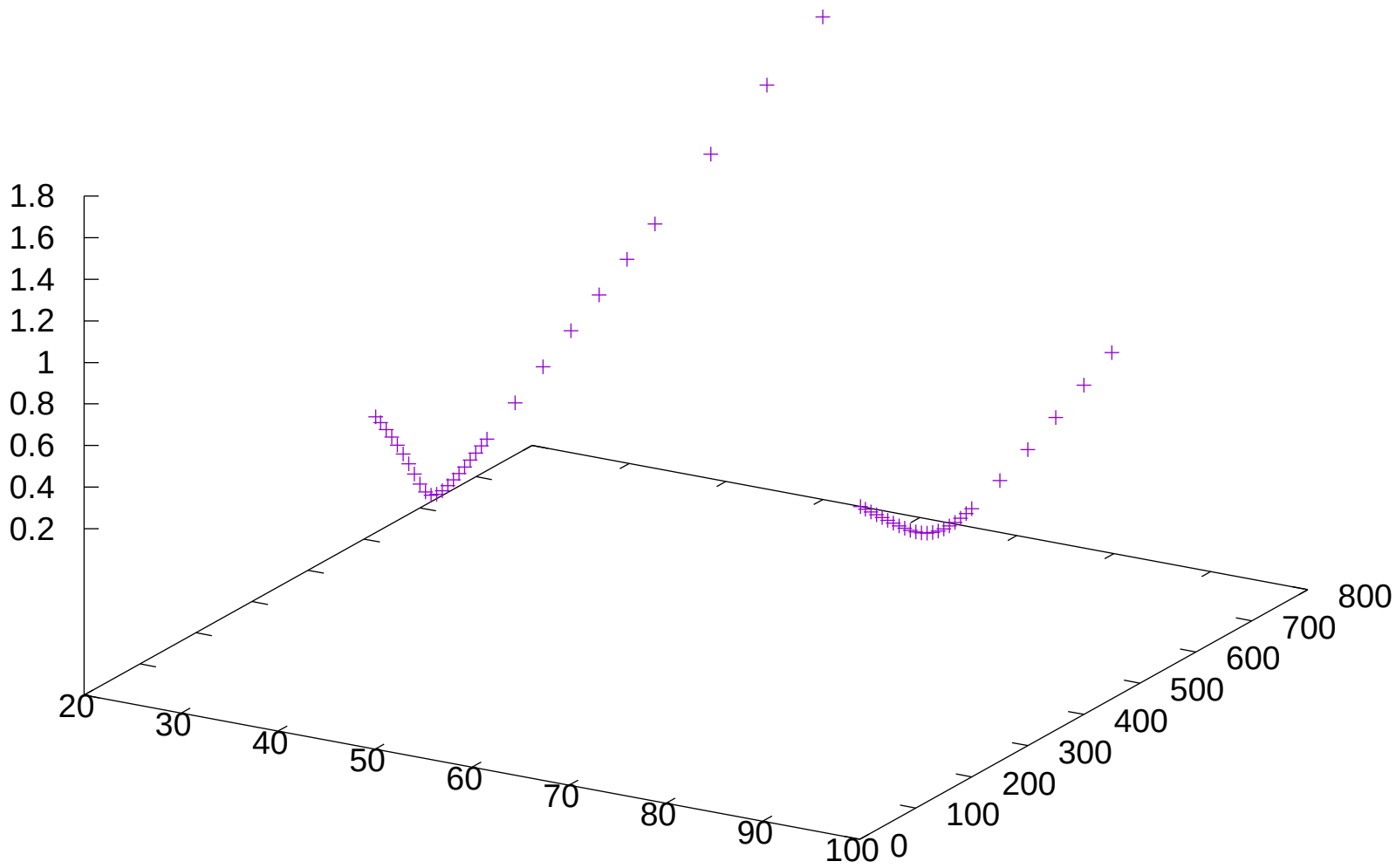
2d binary data example where x coordinate is ignored then generated

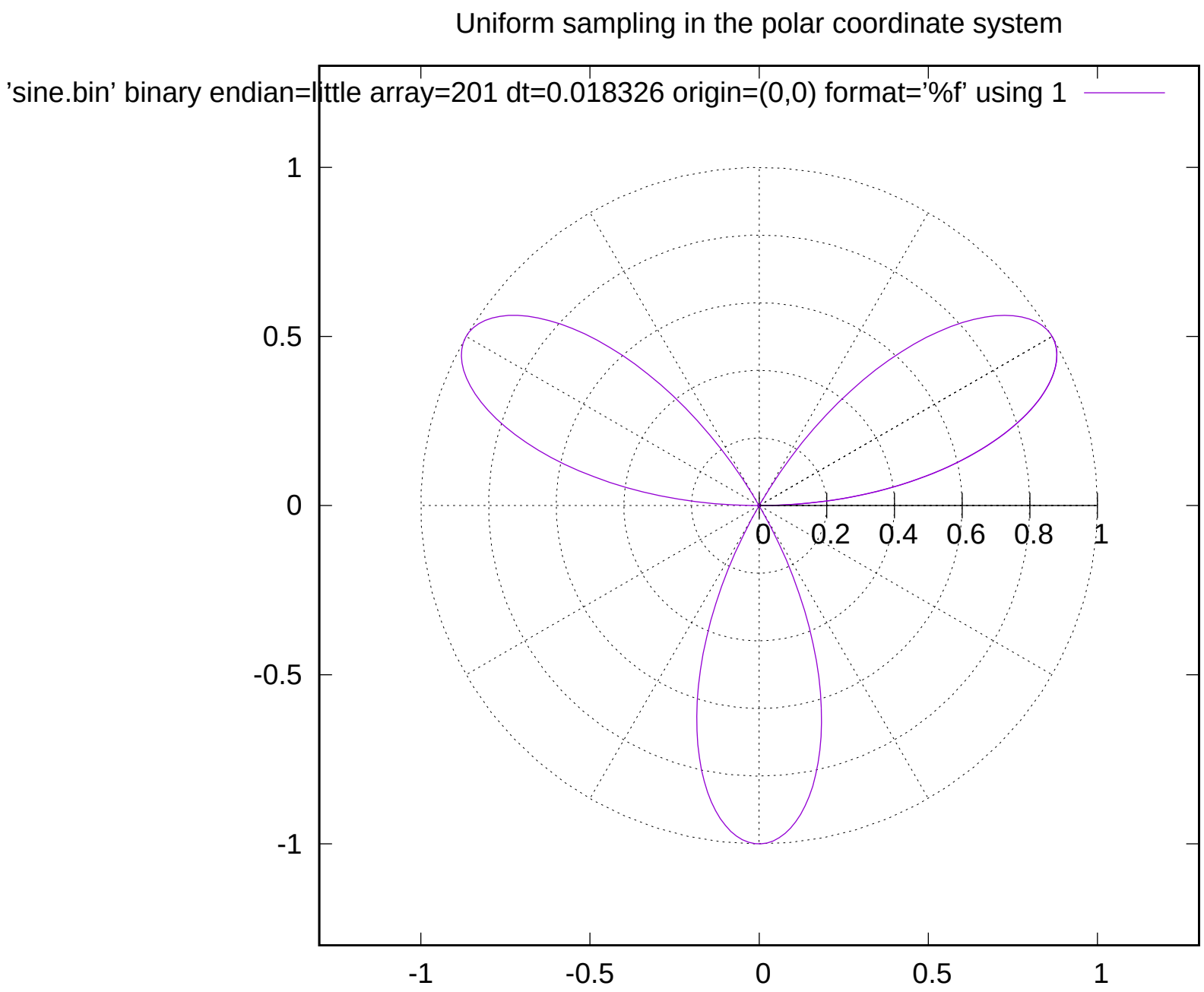
lian=little record=30:30:29:26 origin=(25,0,0):(50,0,0):(75,0,0):(100,0,0) format='%f%f' using (0):2:3



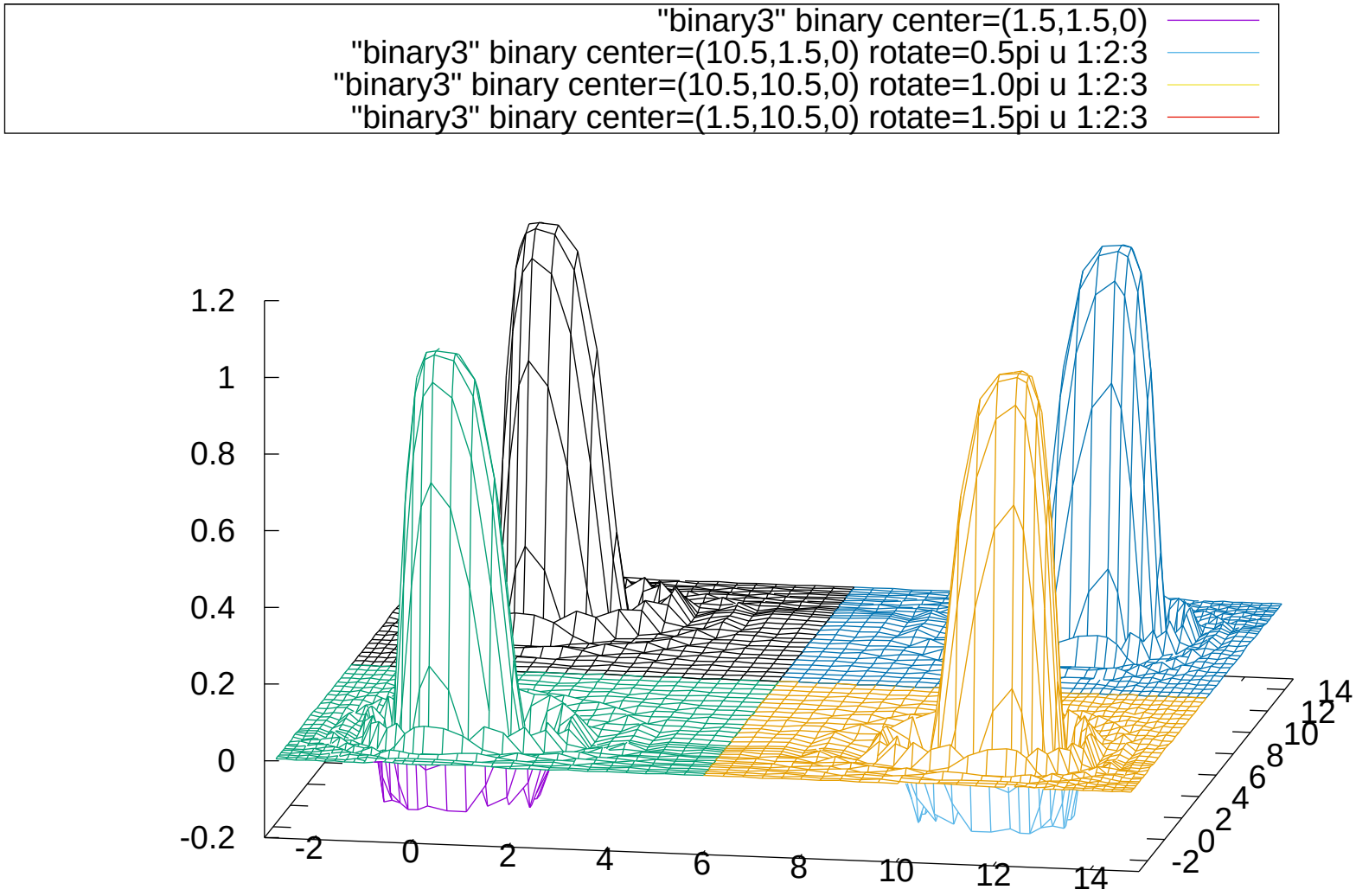
The key word ’skip’ used to ignore some data

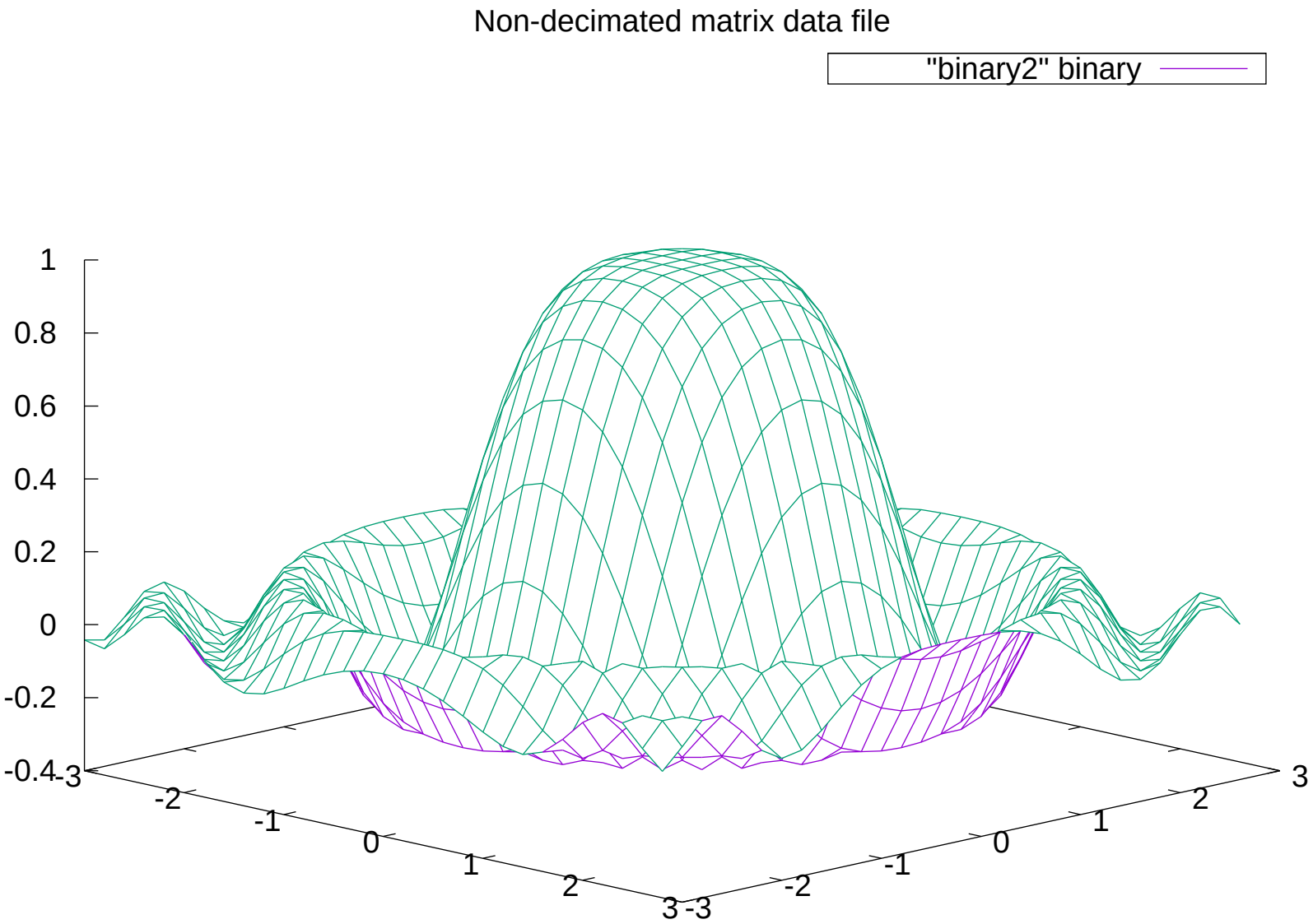
inary endian=little record=30:26 skip=360:348 origin=(50,0,0):(100,0,0) format=’%f%f’ using (0):2:3

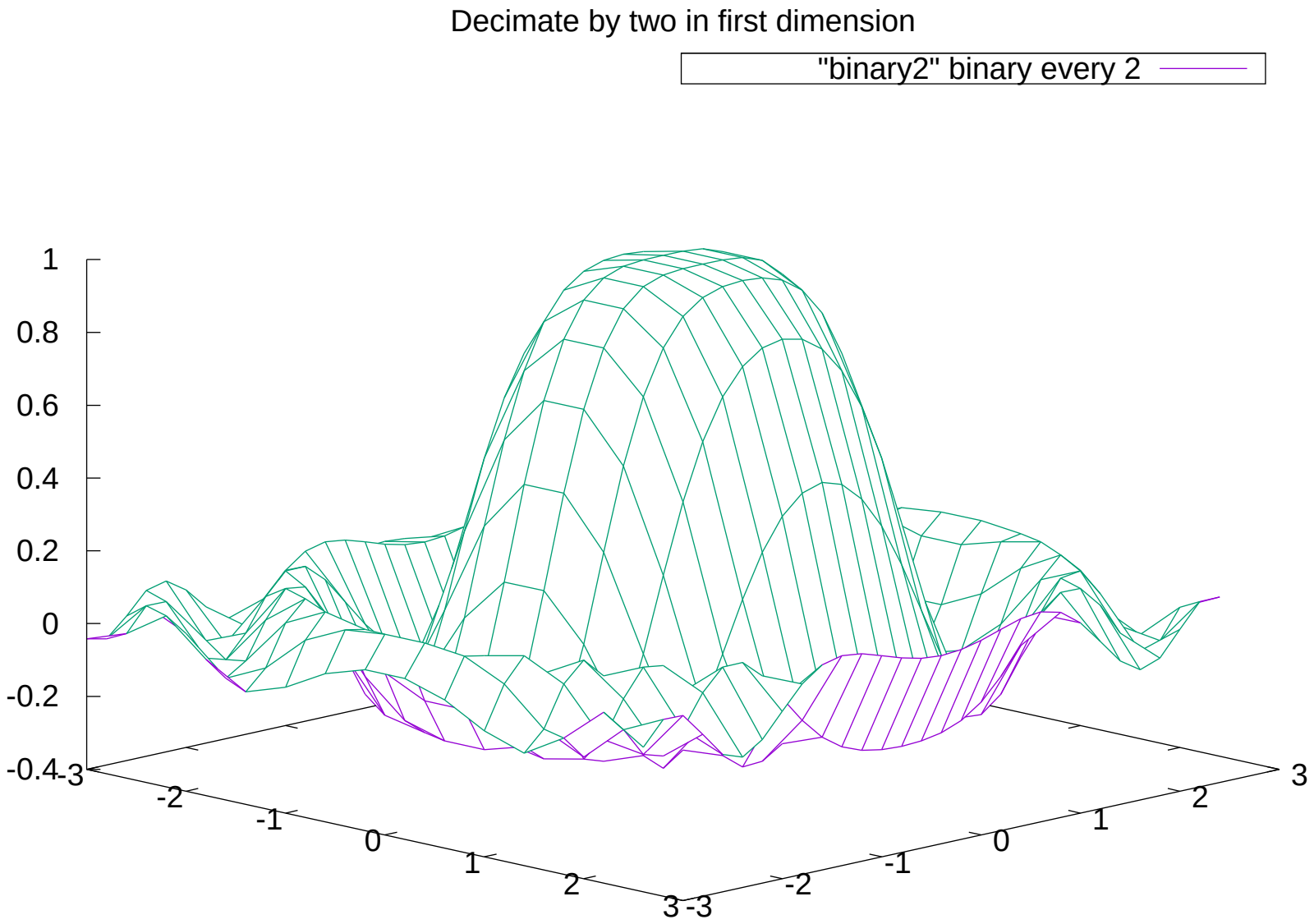


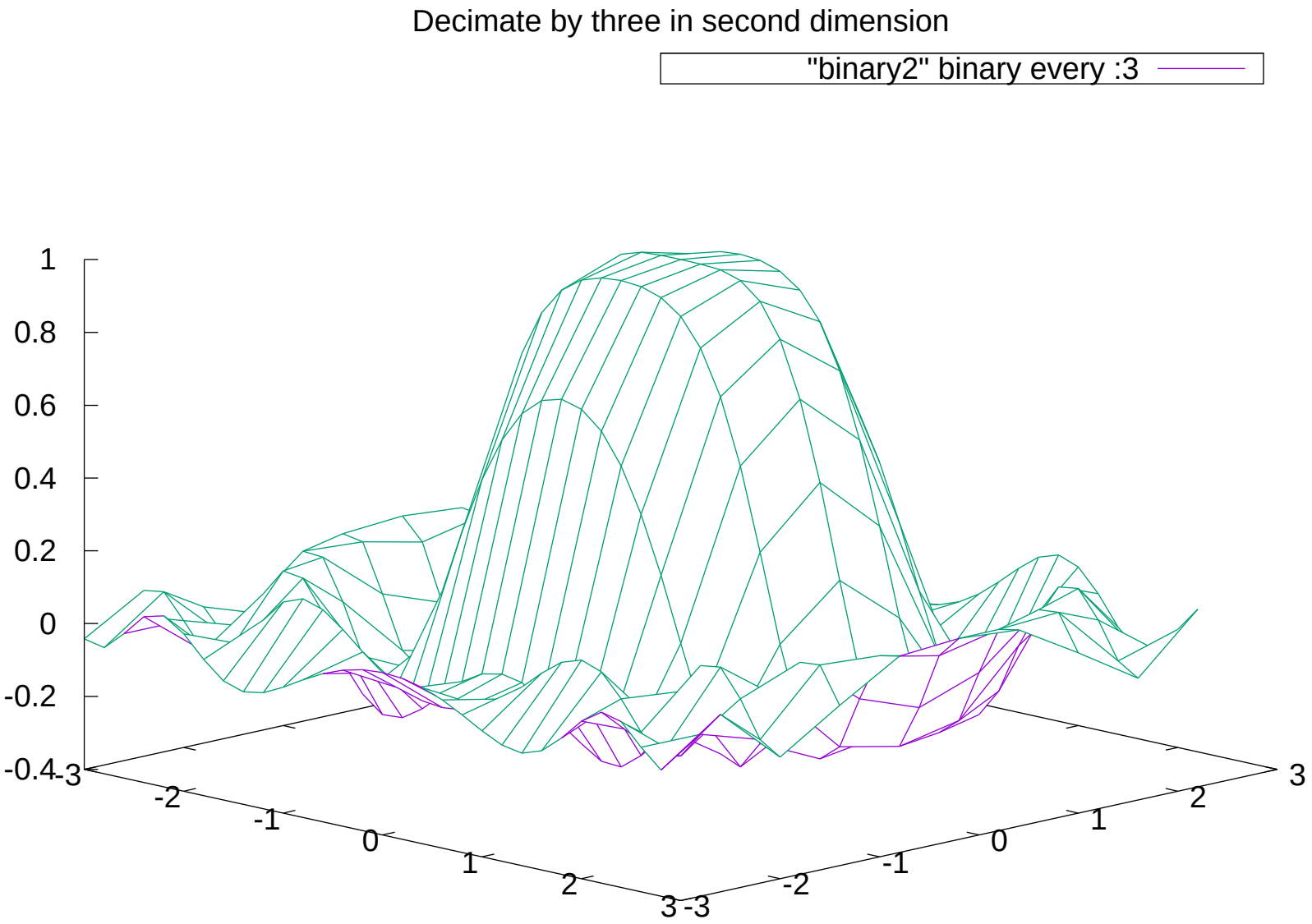

Uniform sampling in the polar coordinate system
'sine.bin' binary endian=little array=201 dt=0.018326 origin=(0,0) format='%f' using 1
1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1
-1
-0.5
0
0.5
1

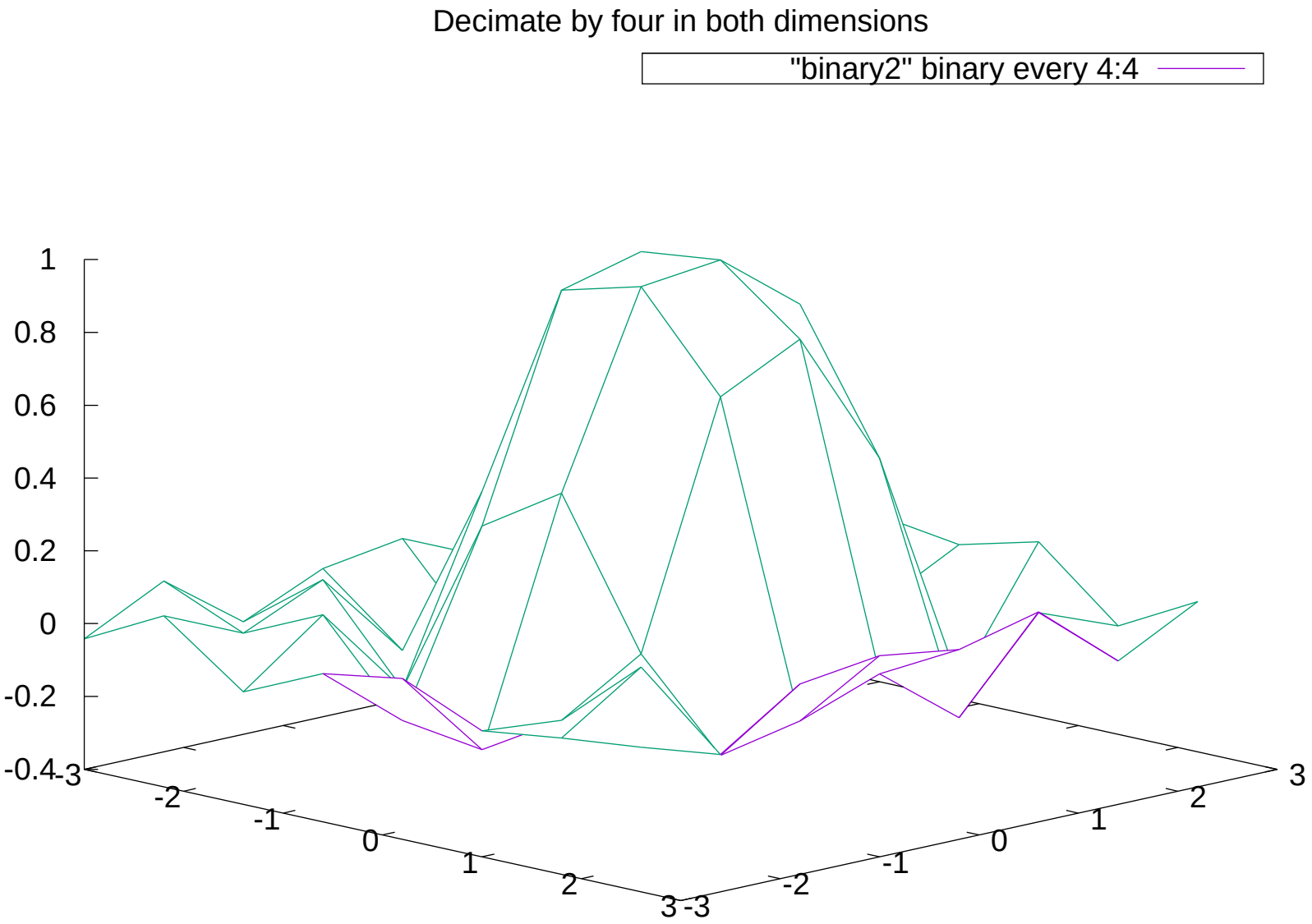
Matrix binary data (gnuplot binary) translated

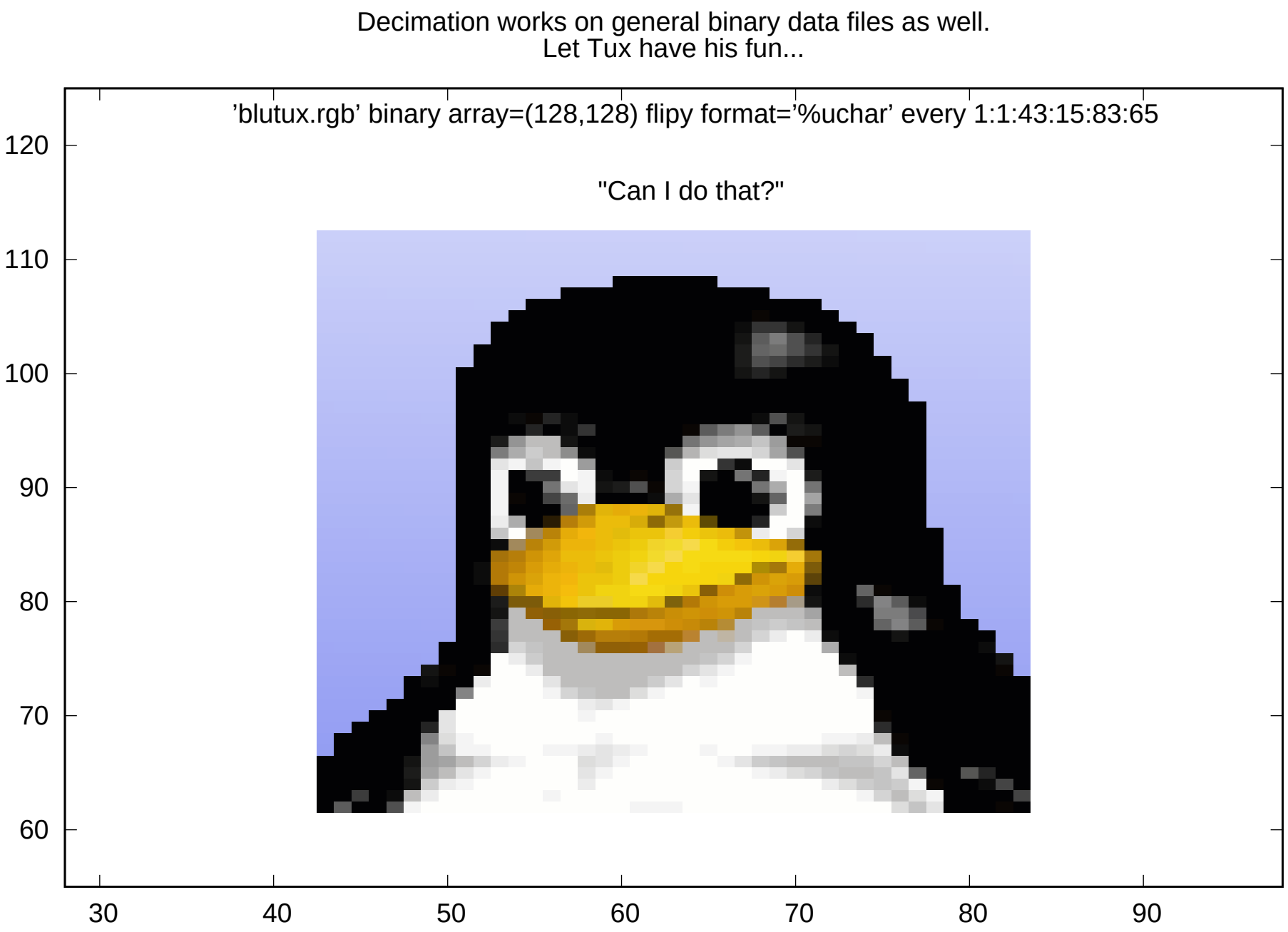



Non-decimated matrix data file
"binary2" binary
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-3
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3

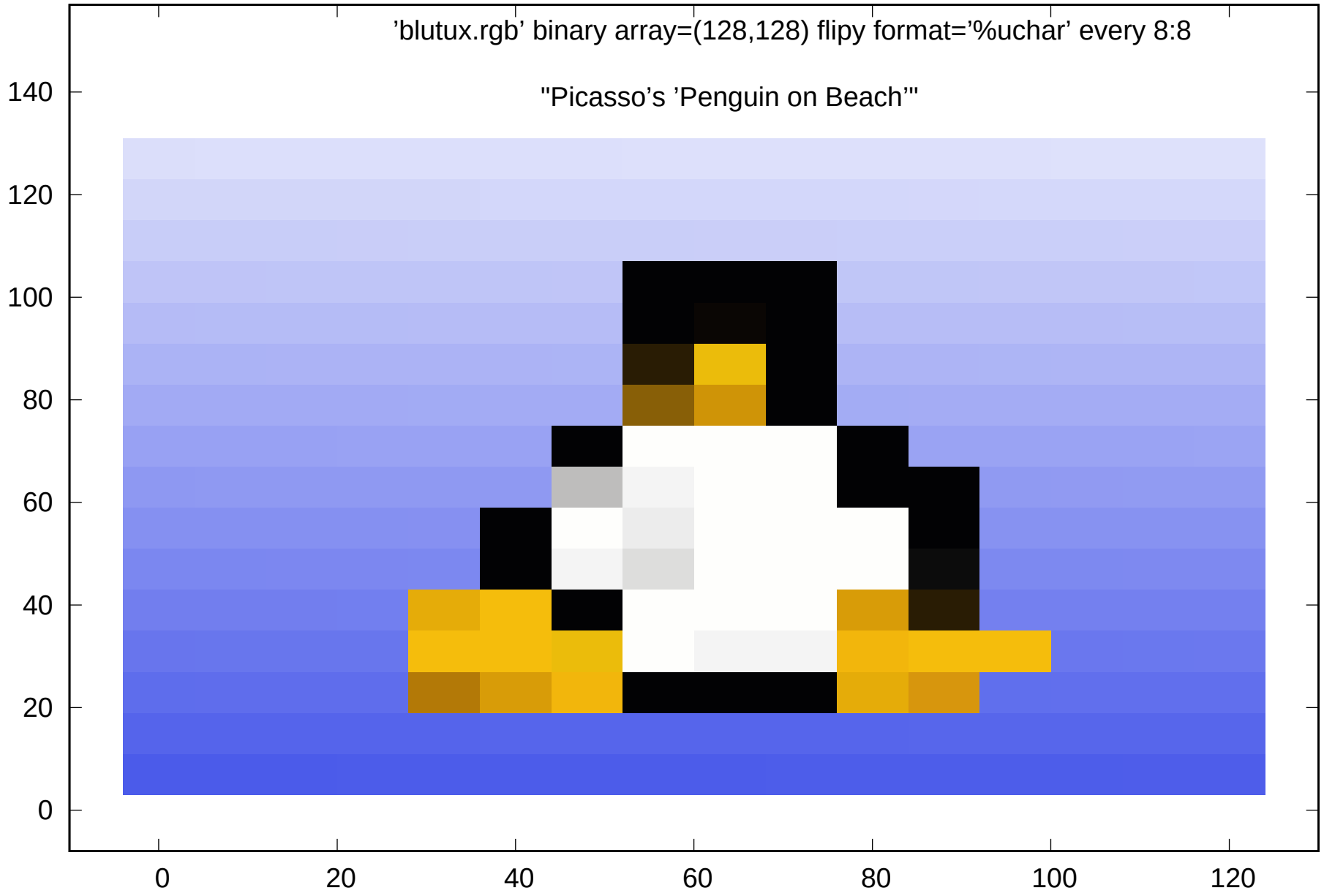

Decimate by two in first dimension
"binary2" binary every 2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-3
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3

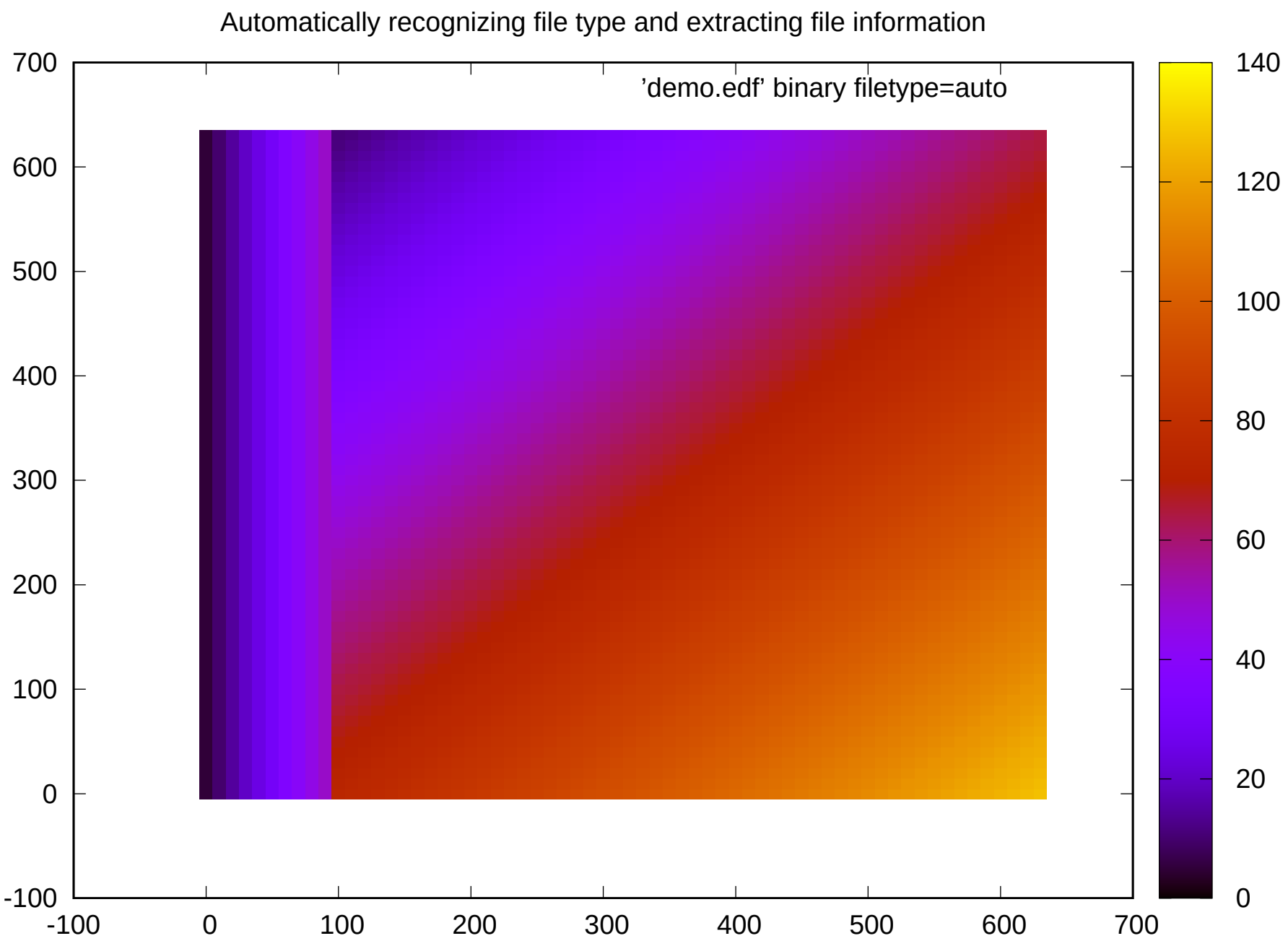

Decimate by three in second dimension
"binary2" binary every :3
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-3
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3

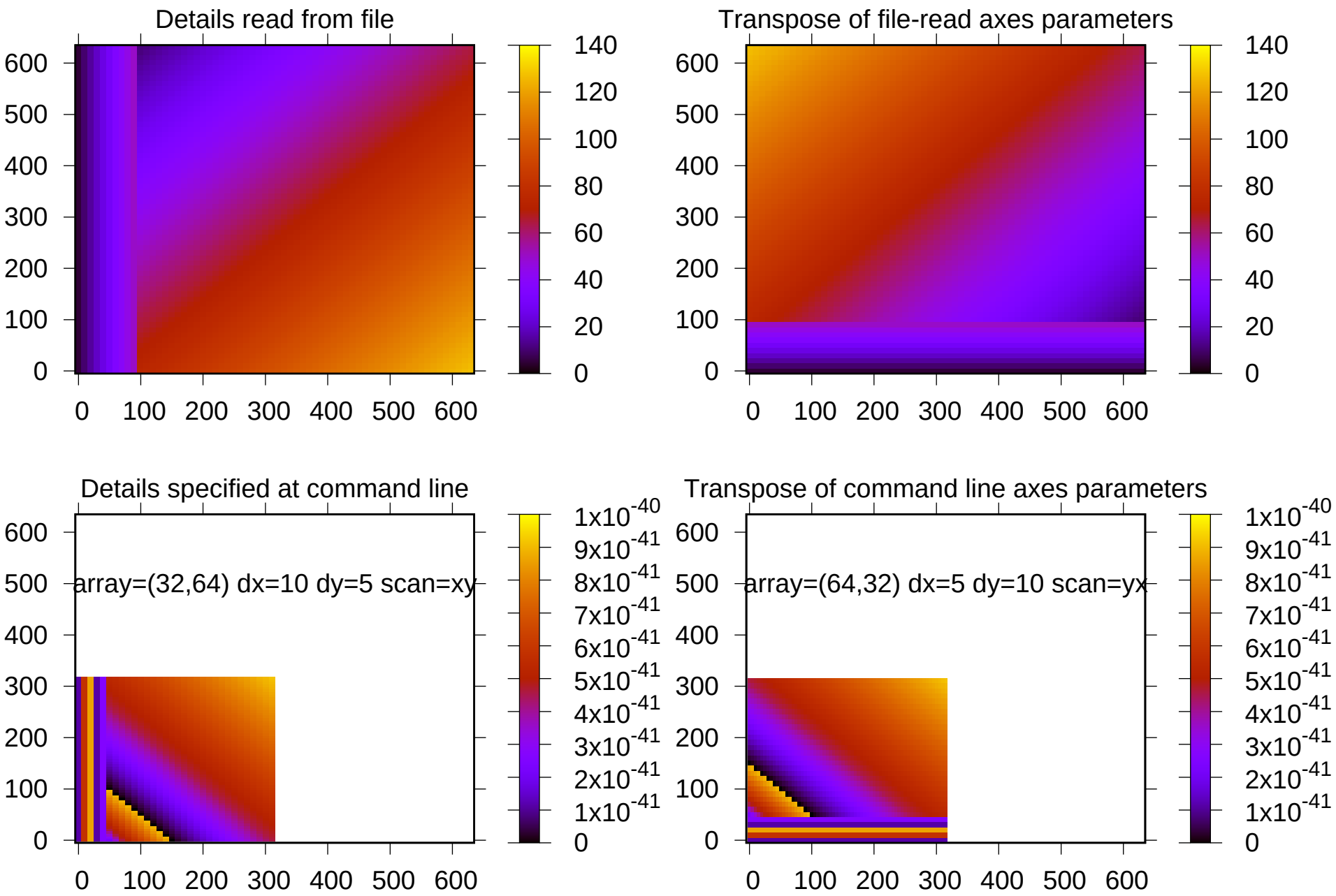

Decimate by four in both dimensions
"binary2" binary every 4:4
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-3
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3

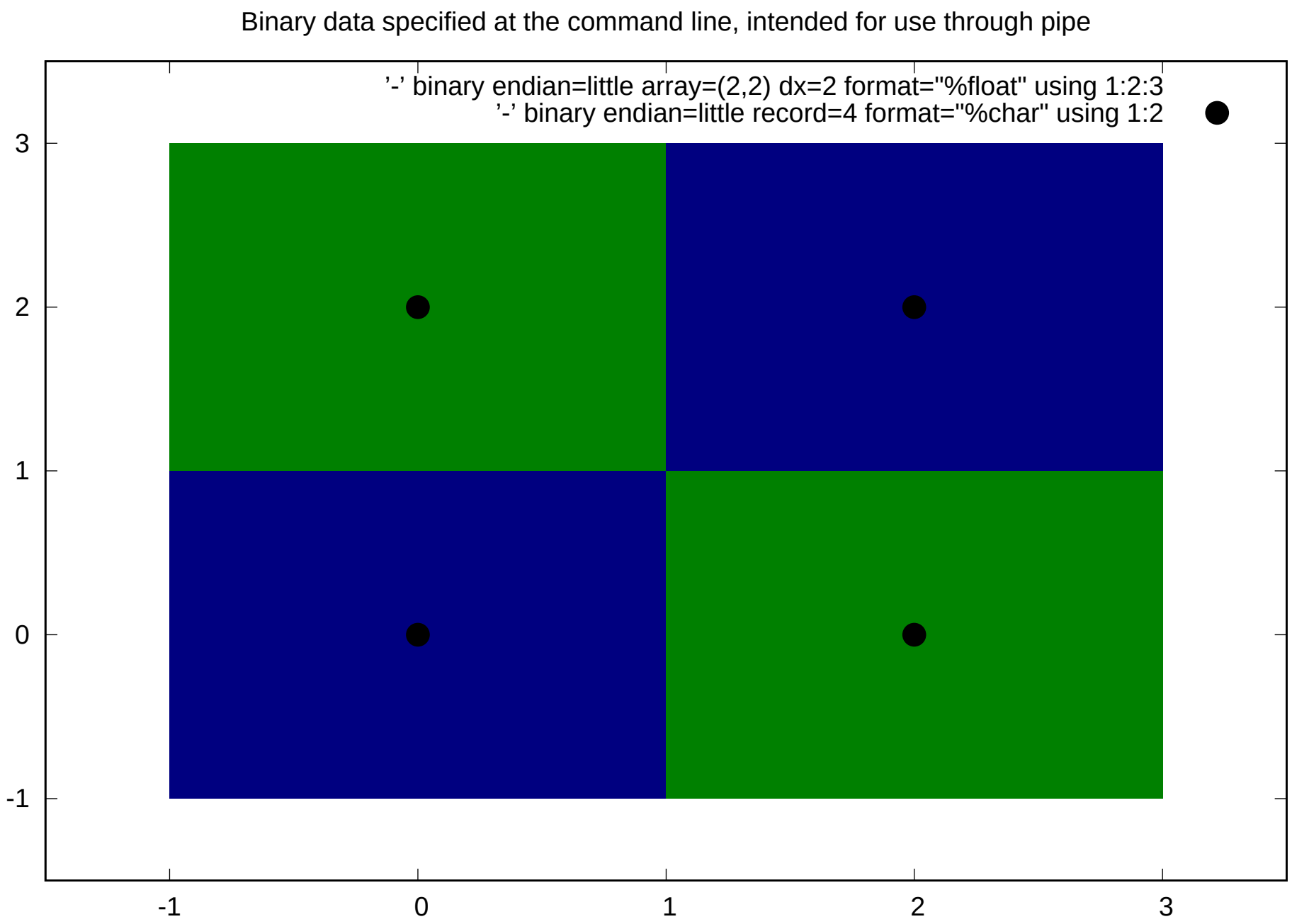

Decimation works on general binary data files as well.
Let Tux have his fun...
'blutux.rgb' binary array=(128,128) flipy format='%uchar' every 1:1:43:15:83:65
"Can I do that?"
120
110
100
90
80
70
60
30
40
50
60
70
80
90

... Sure, go ahead.

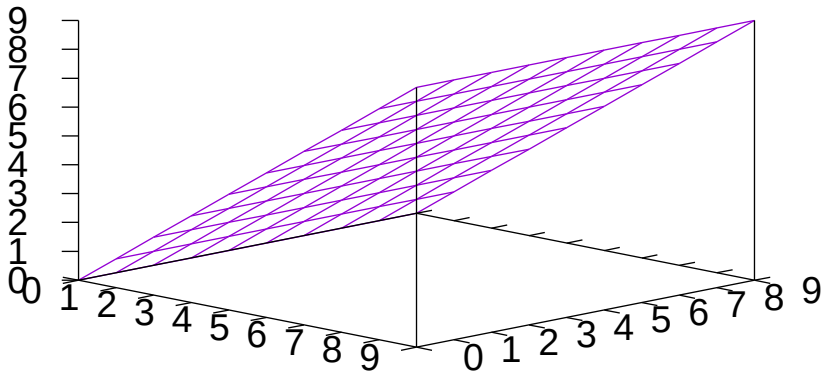



Automatically recognizing file type and extracting file information
'demo.edf' binary filetype=auto
700
600
500
400
300
200
100
0
-100
-100
0
100
200
300
400
500
600
700
140
120
100
80
60
40
20
0

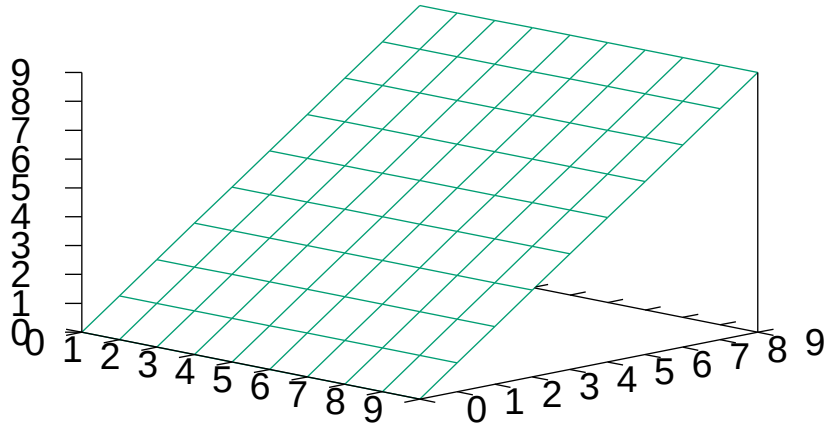

Details read from file
Transpose of file-read axes parameters
Details specified at command line
array=(32,64) dx=10 dy=5 scan=xy
Transpose of command line axes parameters
array=(64,32) dx=5 dy=10 scan=yx
0
100
200
300
400
500
600
0
20
40
60
80
100
120
140
1x10-40
9x10-41
8x10-41
7x10-41
6x10-41
5x10-41
4x10-41
3x10-41
2x10-41
1x10-41


Binary data specified at the command line, intended for use through pipe
'-' binary endian=little array=(2,2) dx=2 format="%float" using 1:2:3
'-' binary endian=little record=4 format="%char" using 1:2
3
2
1
0
-1
-1
0
1
2
3

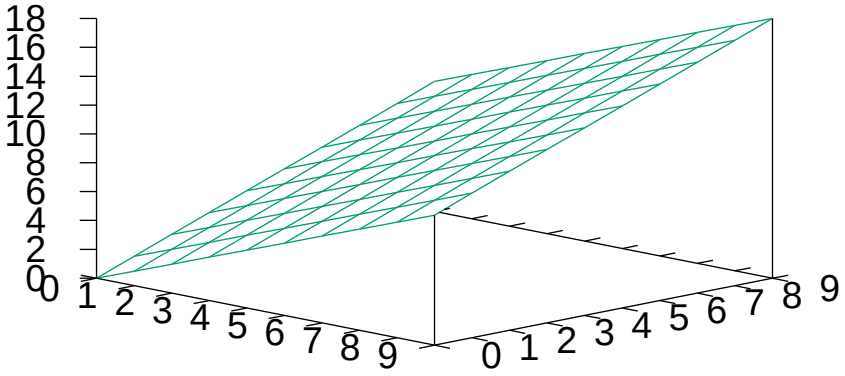
’asciimat.dat’ matrix index 0



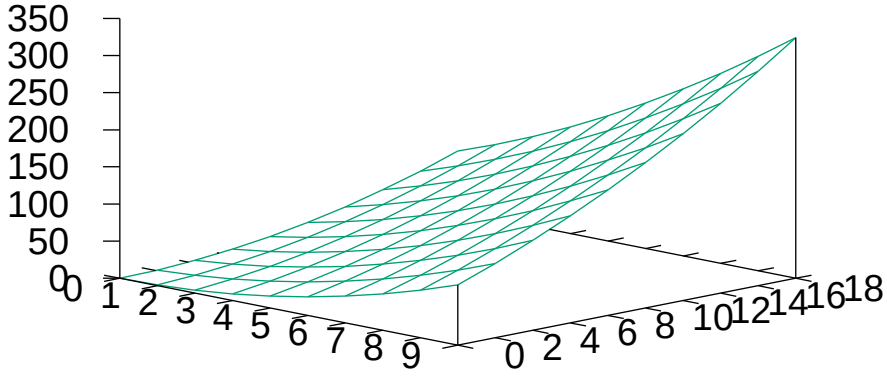
’asciimat.dat’ matrix index 1



’asciimat.dat’ matrix index 2

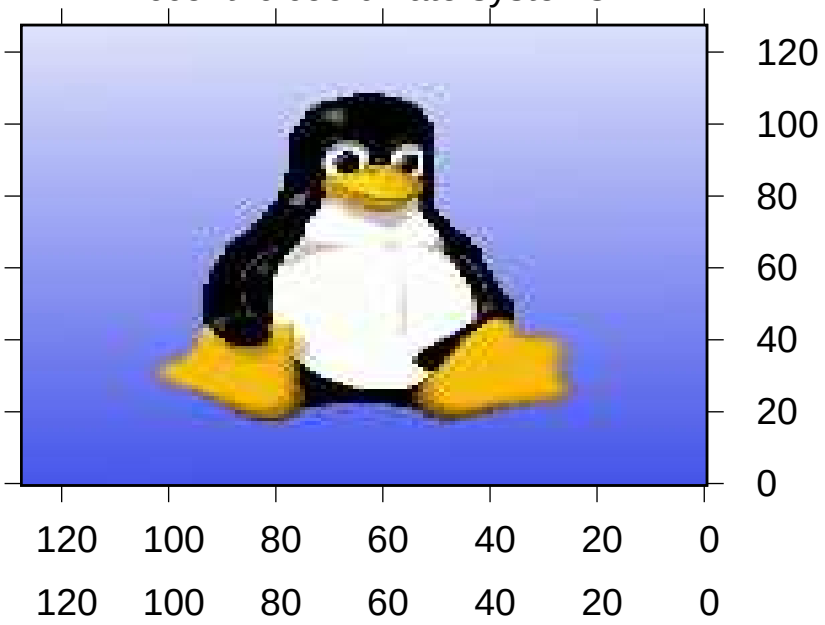


’asciimat.dat’ matrix index 2 using 1:(2*$2):($3*$3)

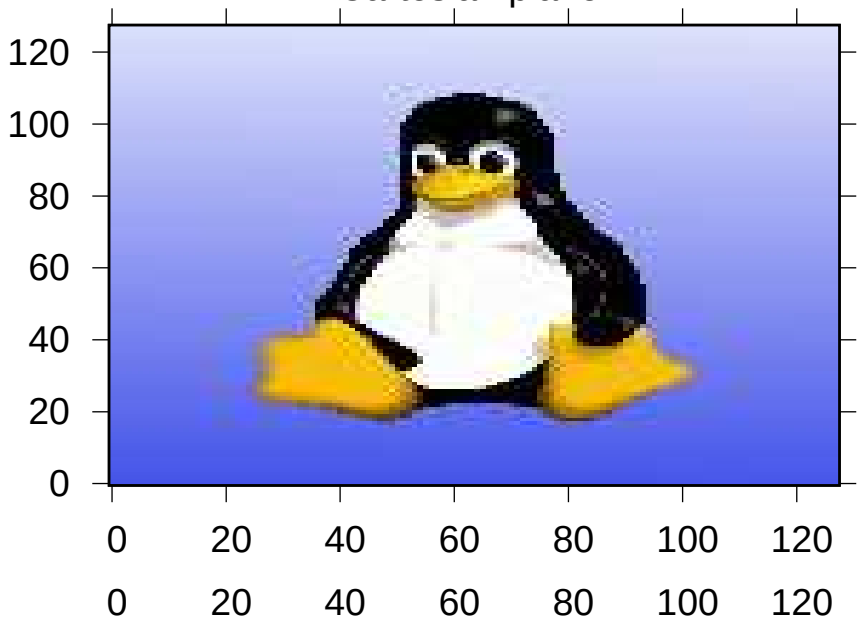


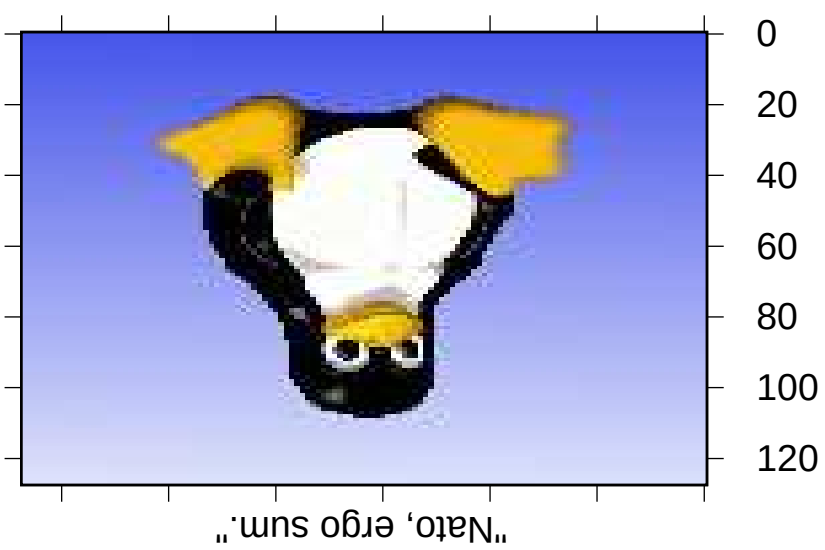
Images reverse according to axis orientation

"Eccentric coordinate systems"

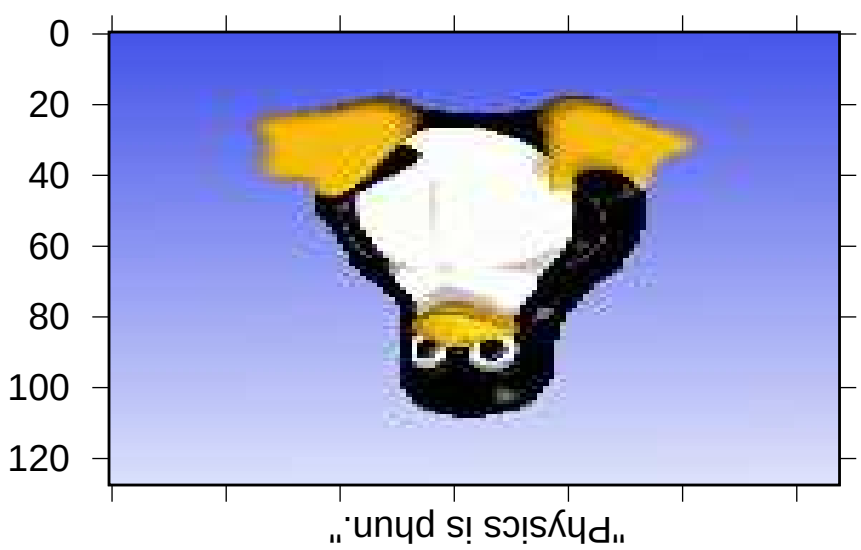


"Cartesian plane!"

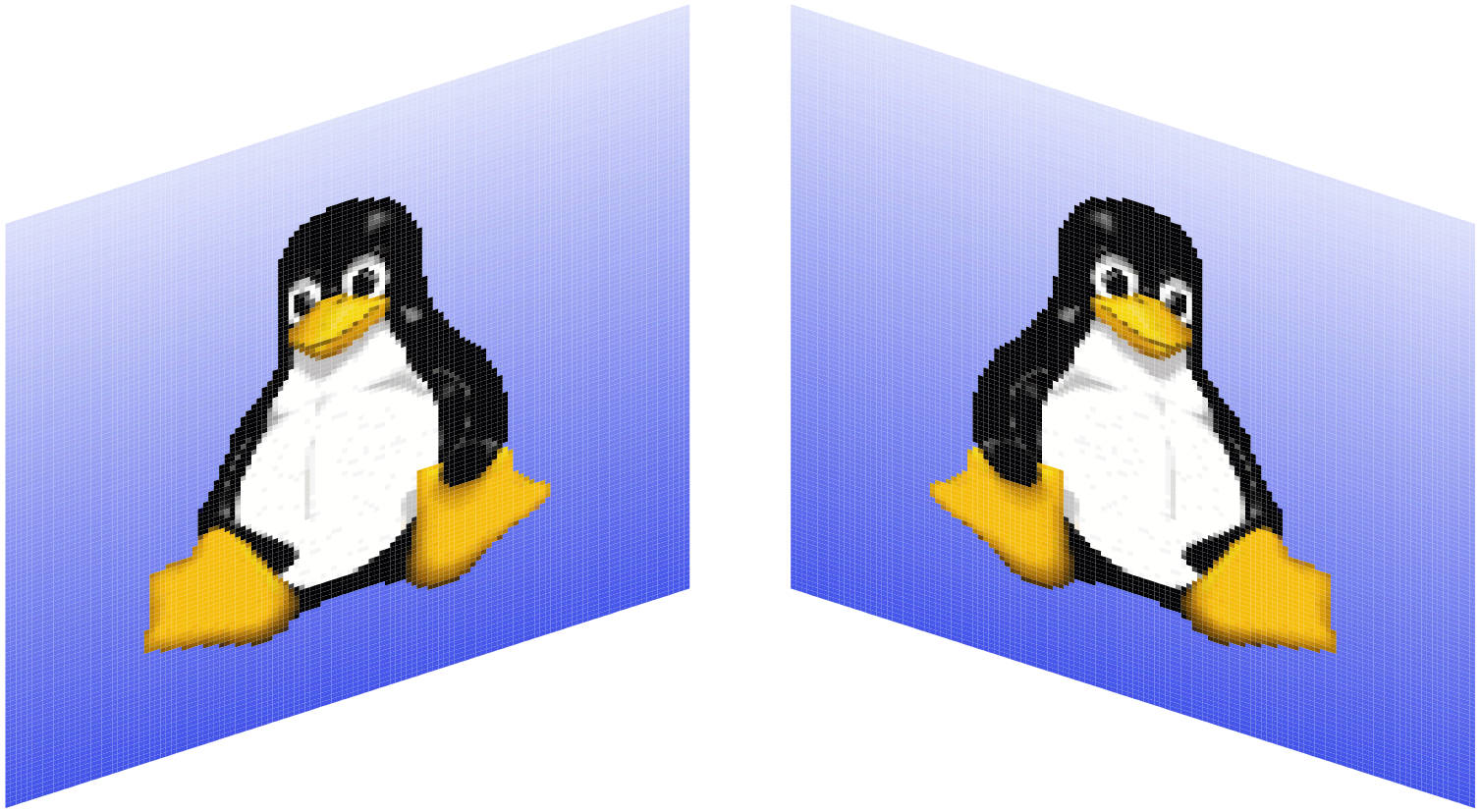




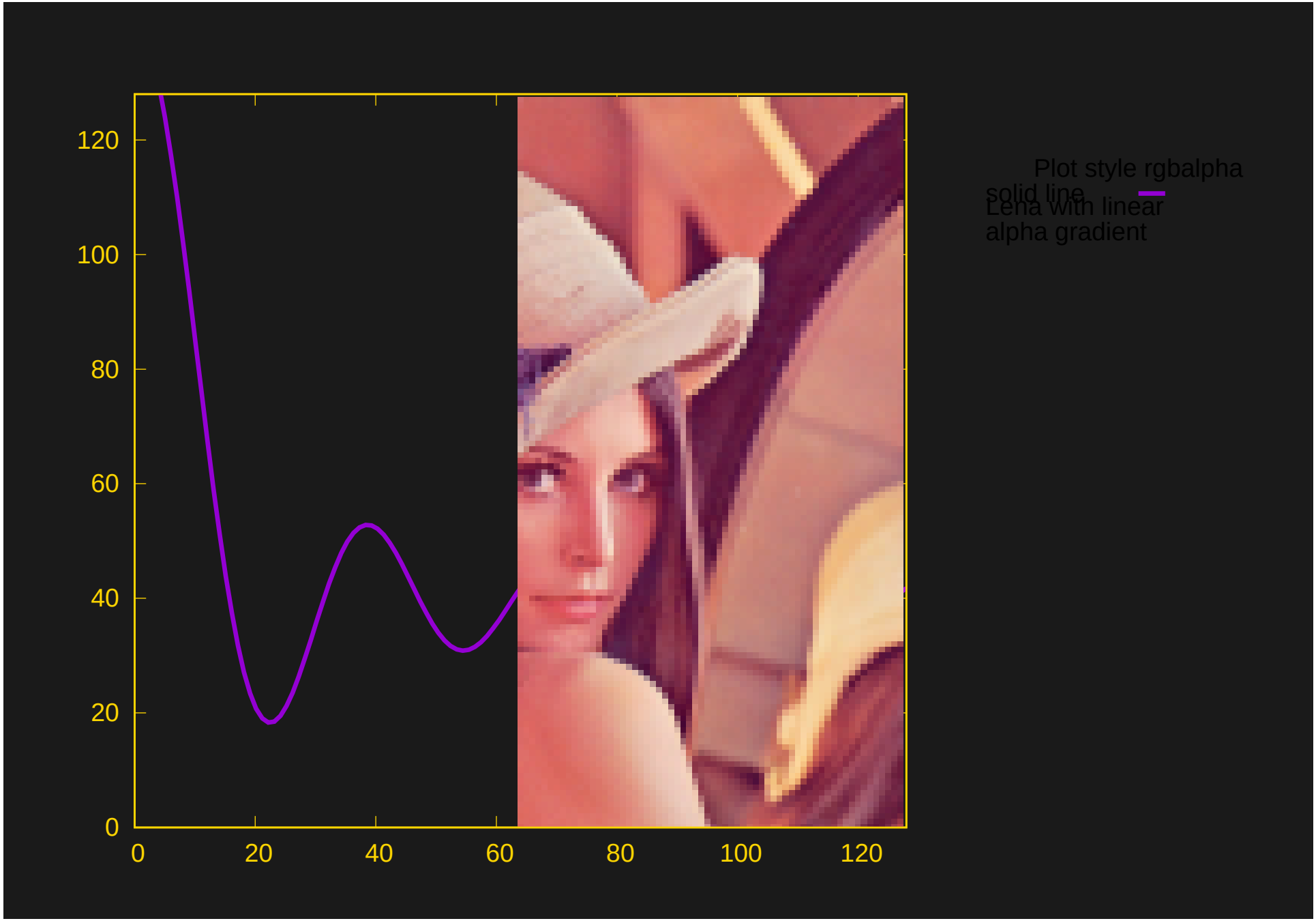
"Nato, ergo sum."

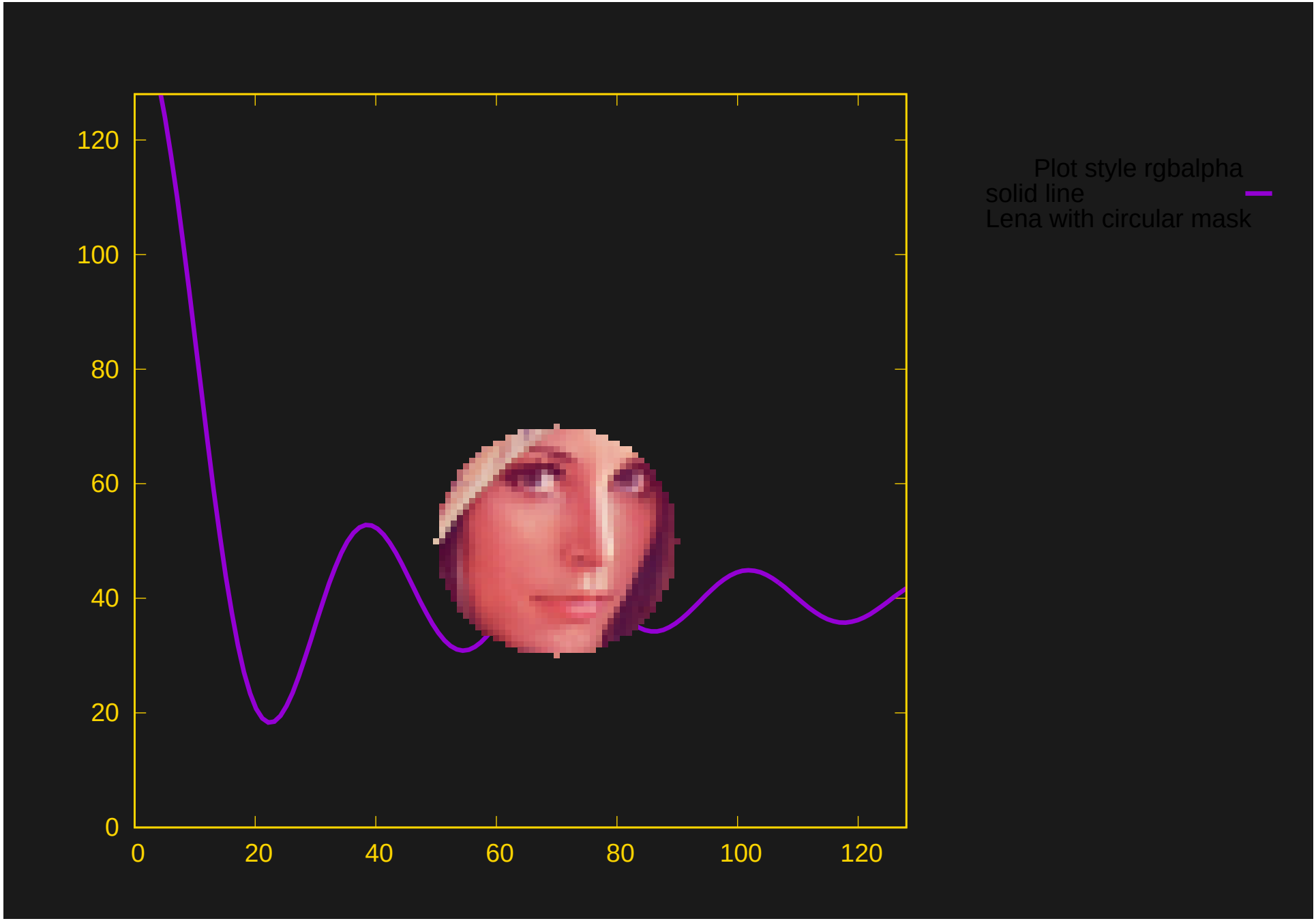


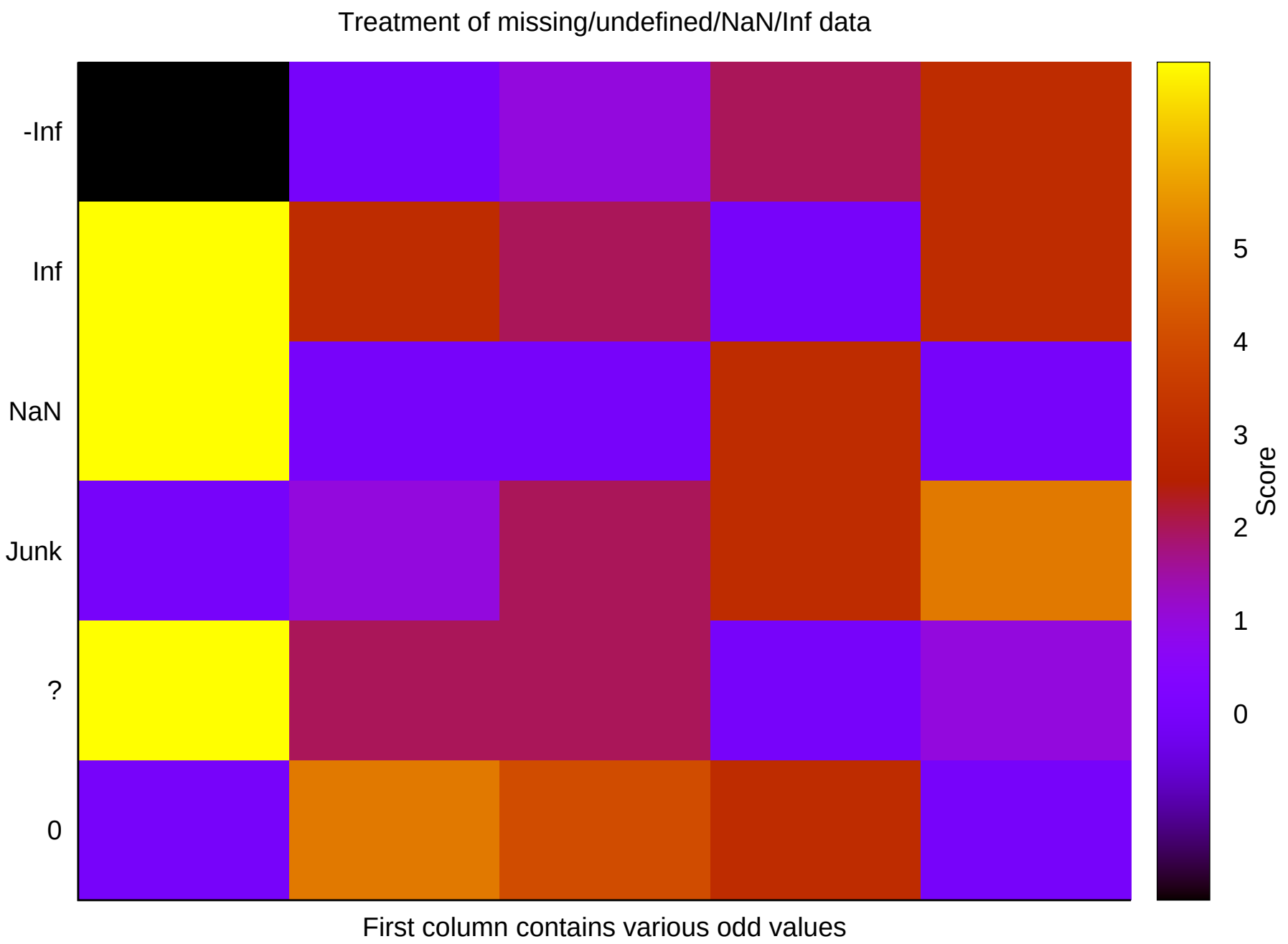
"Physics is phun."

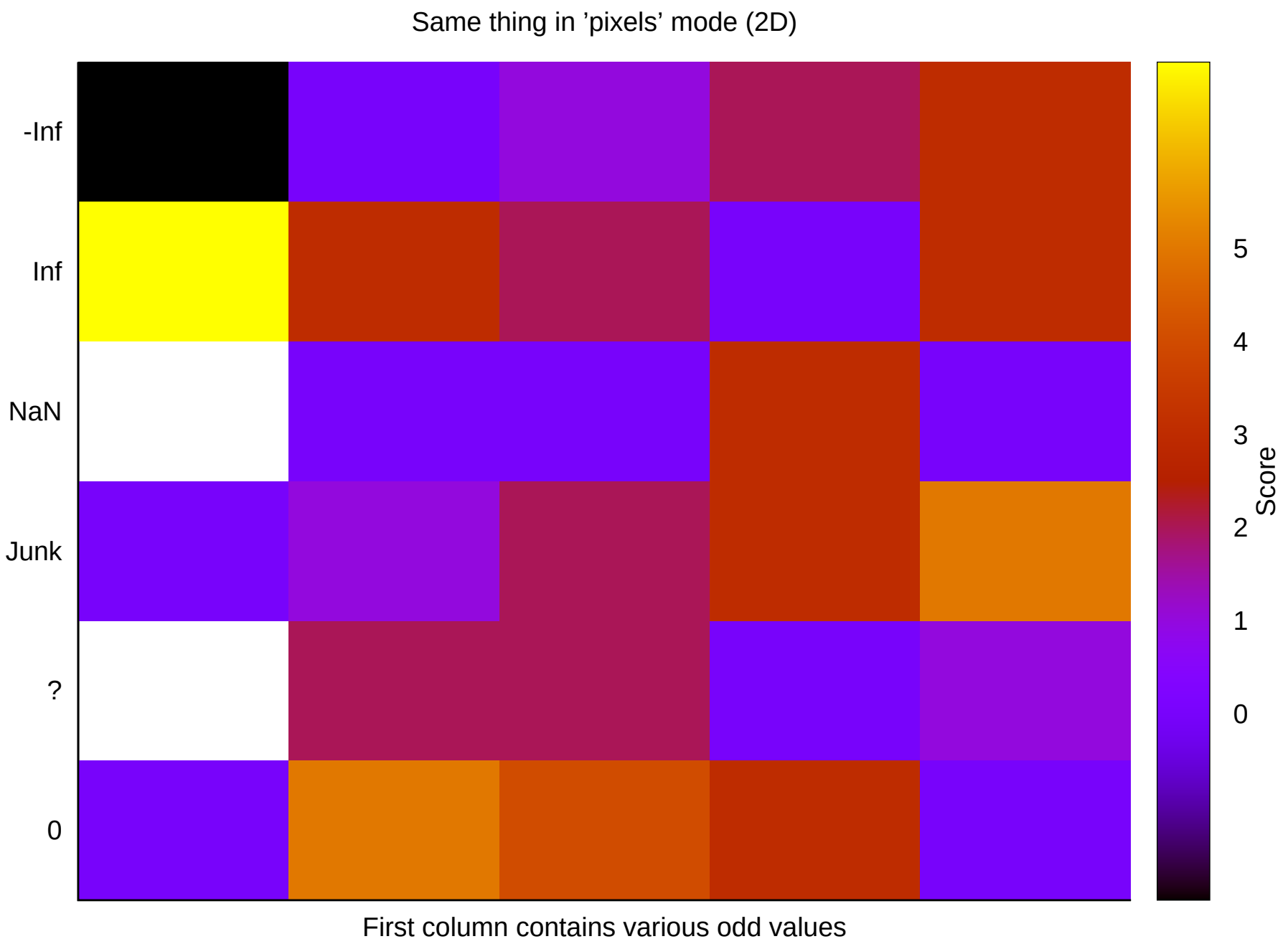


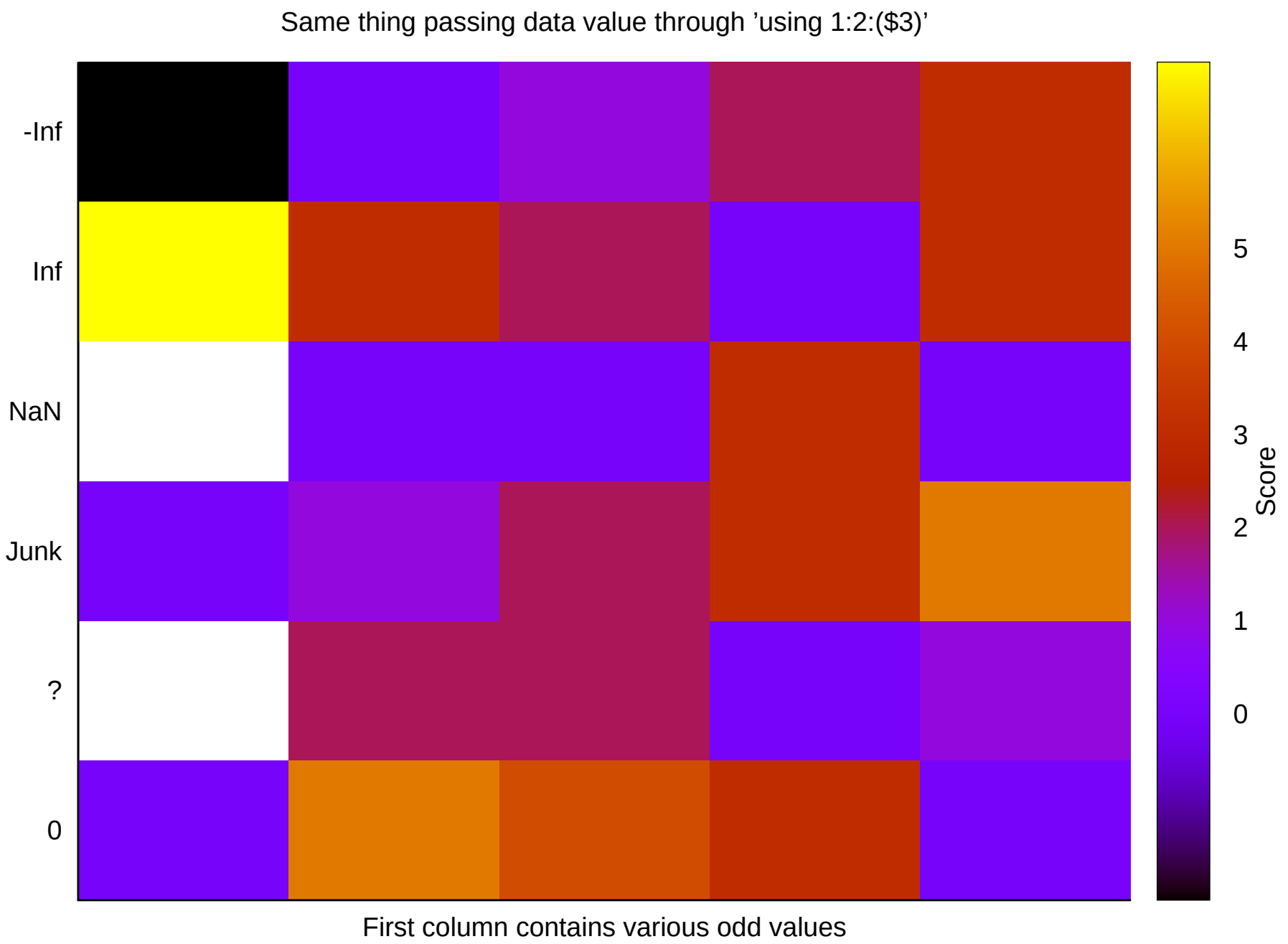
Tux in a reflective mood


120
100
80
60
40
20
0
0
20
40
60
80
100
120
Plot style rgbalpha
solid line
Lena with linear
alpha gradient

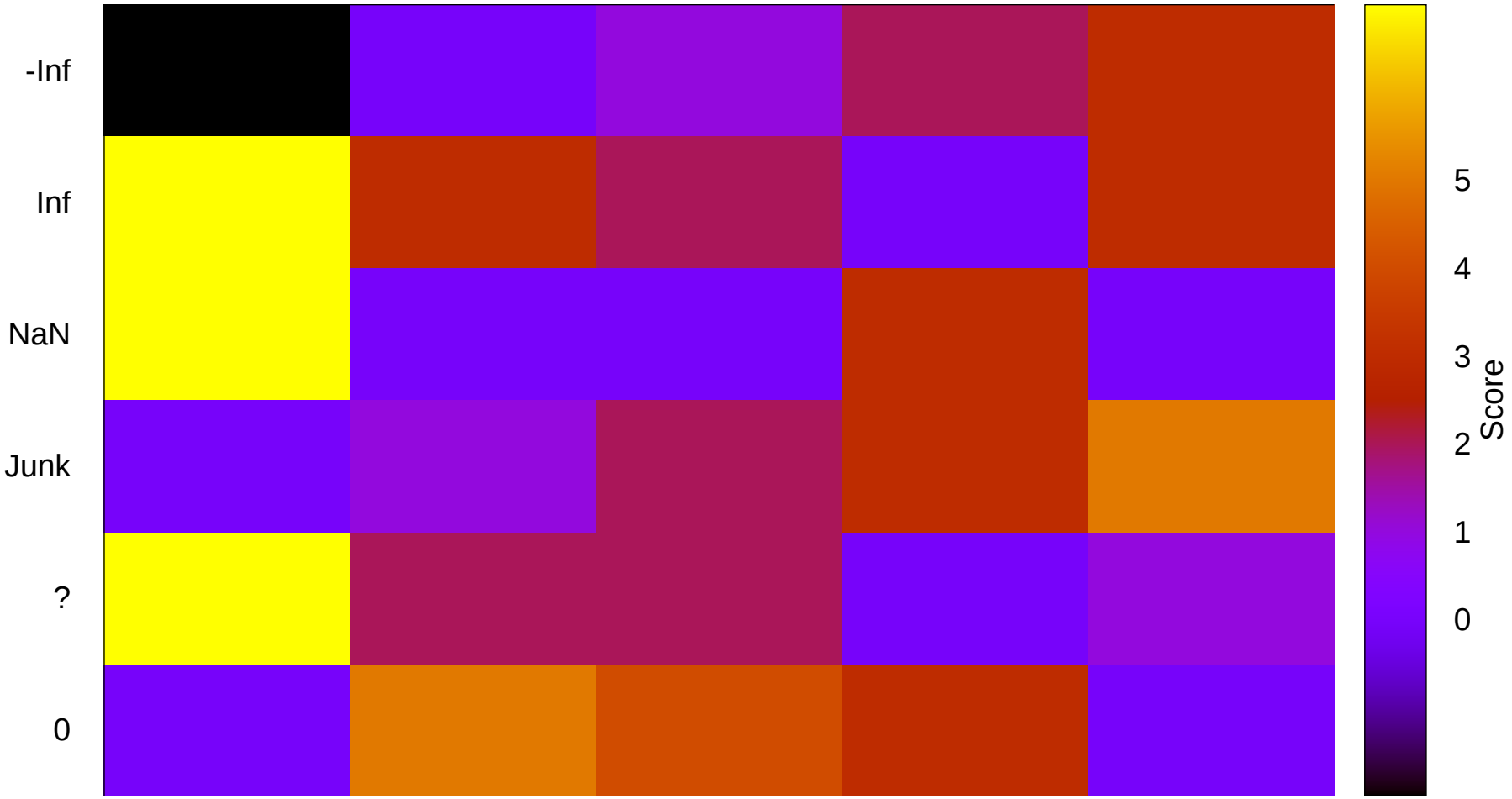

120
100
80
60
40
20
0
0
20
40
60
80
100
120
Plot style rgbalpha
solid line
Lena with circular mask


Treatment of missing/undefined/NaN/Inf data
-Inf
Inf
NaN
Junk
?
0
Score
5
4
3
2
1
0
First column contains various odd values


Same thing in ’pixels’ mode (2D)
-Inf
Inf
NaN
Junk
?
0
5
4
3
2
1
0
Score
First column contains various odd values

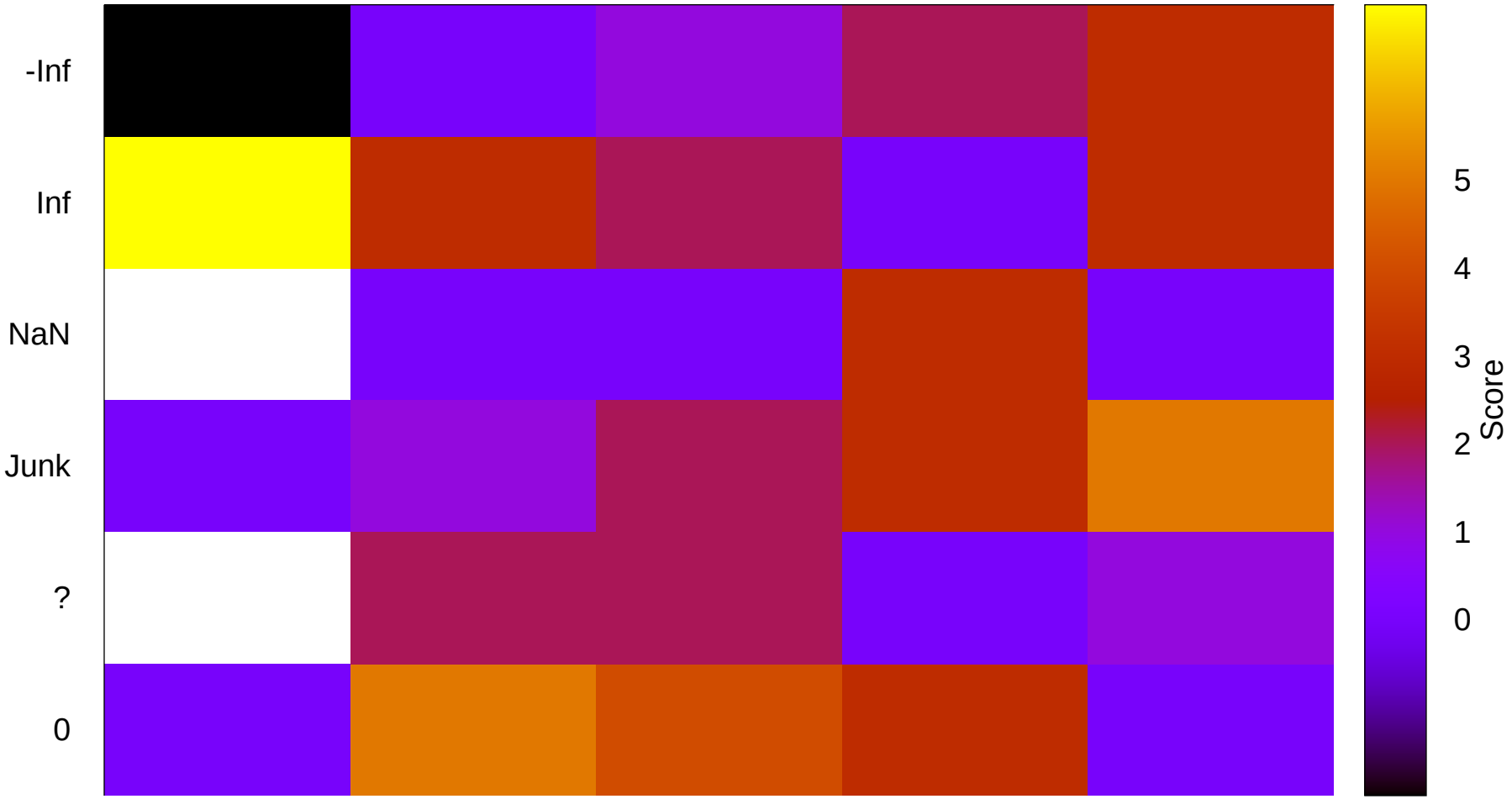

Same thing passing data value through 'using 1:2:($3)'
-Inf
Inf
NaN
Junk
?
0
5
4
3
2
1
0
Score
First column contains various odd values

Same thing in 3D mode

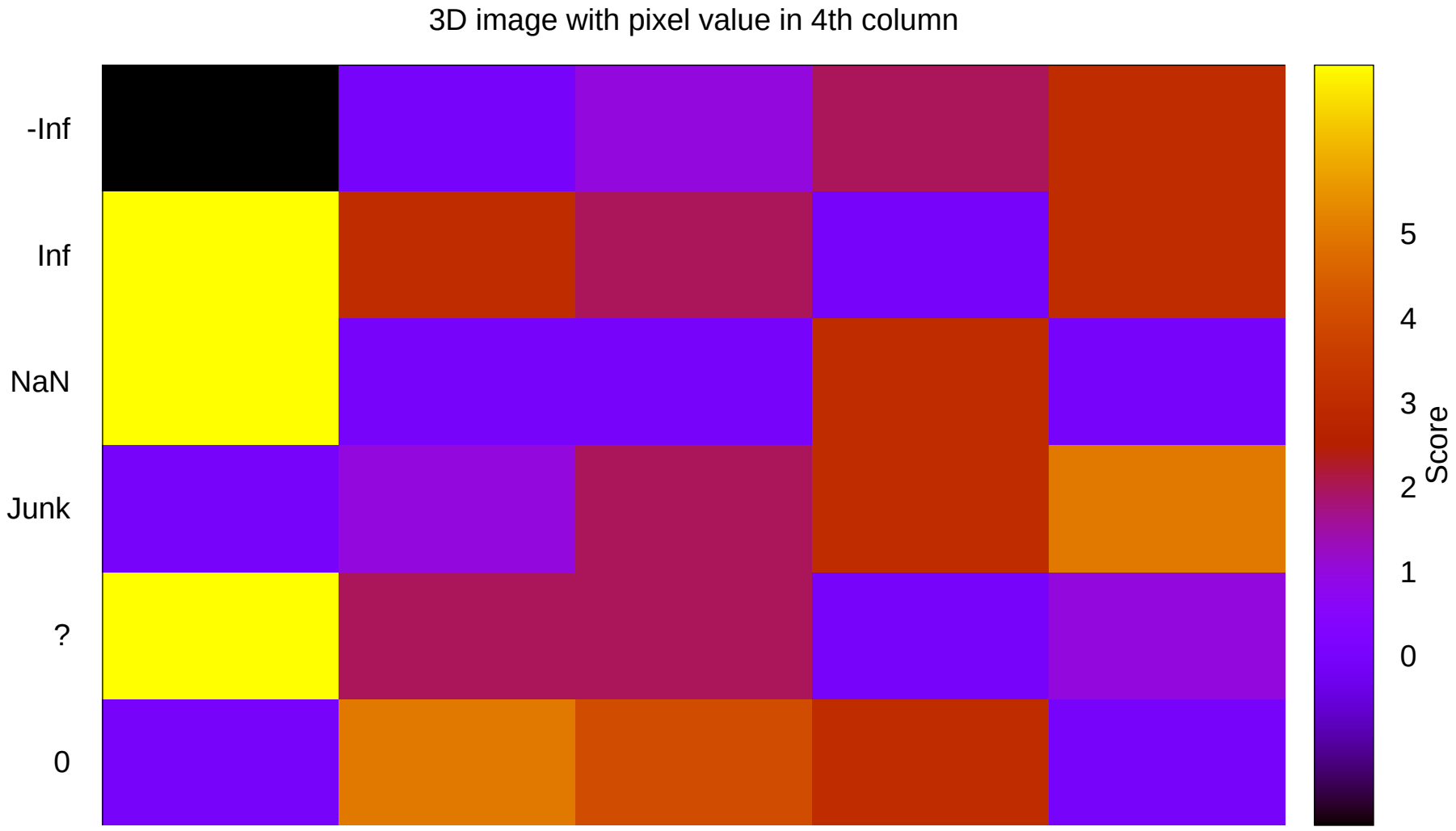


First column contains various odd values

Same thing in ’pixels’ mode (3D)



First column contains various odd values


3D image with pixel value in 4th column
-Inf
Inf
NaN
Junk
?
0
5
4
3
2
1
0
Score
First column contains various odd values

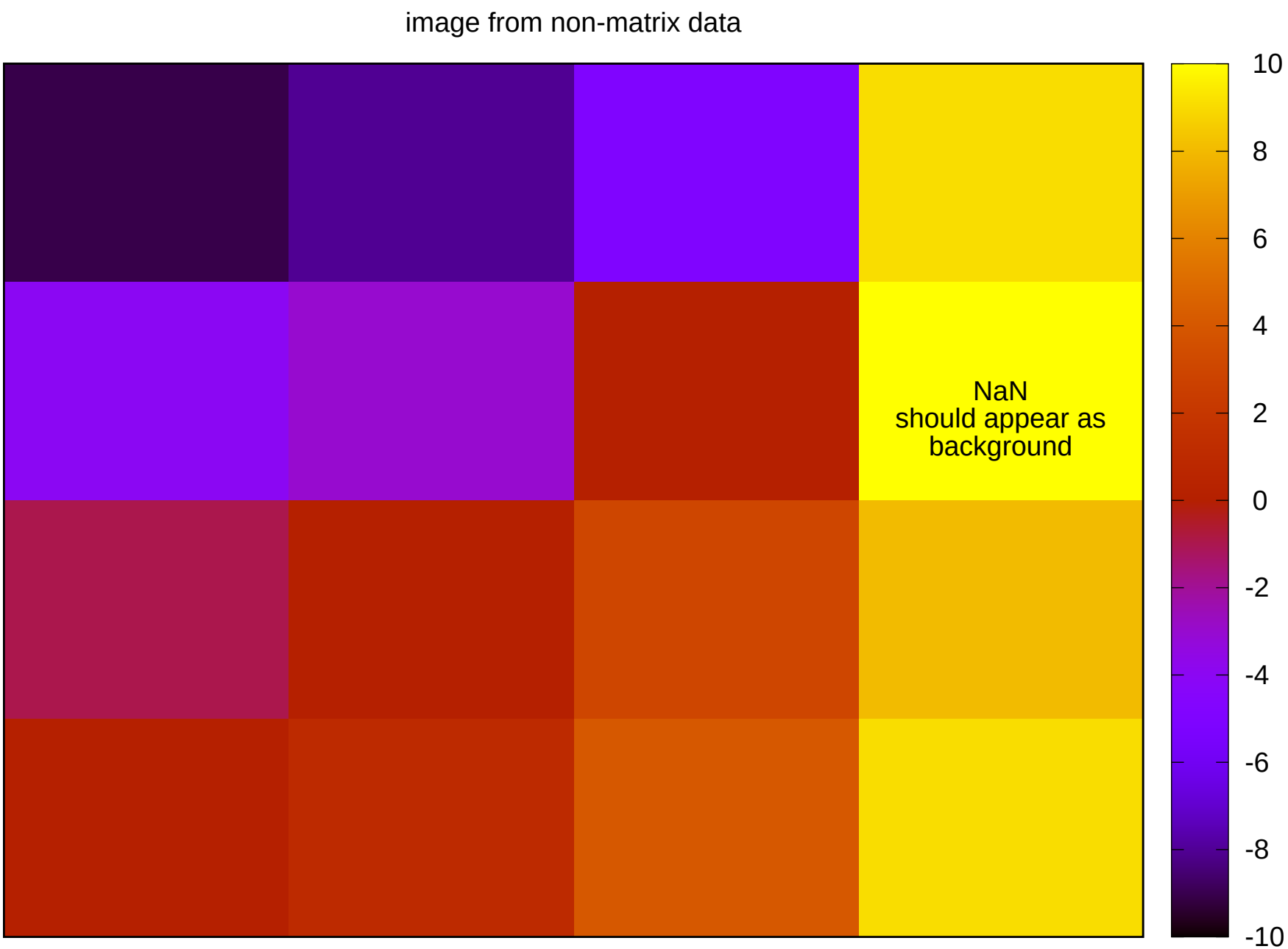
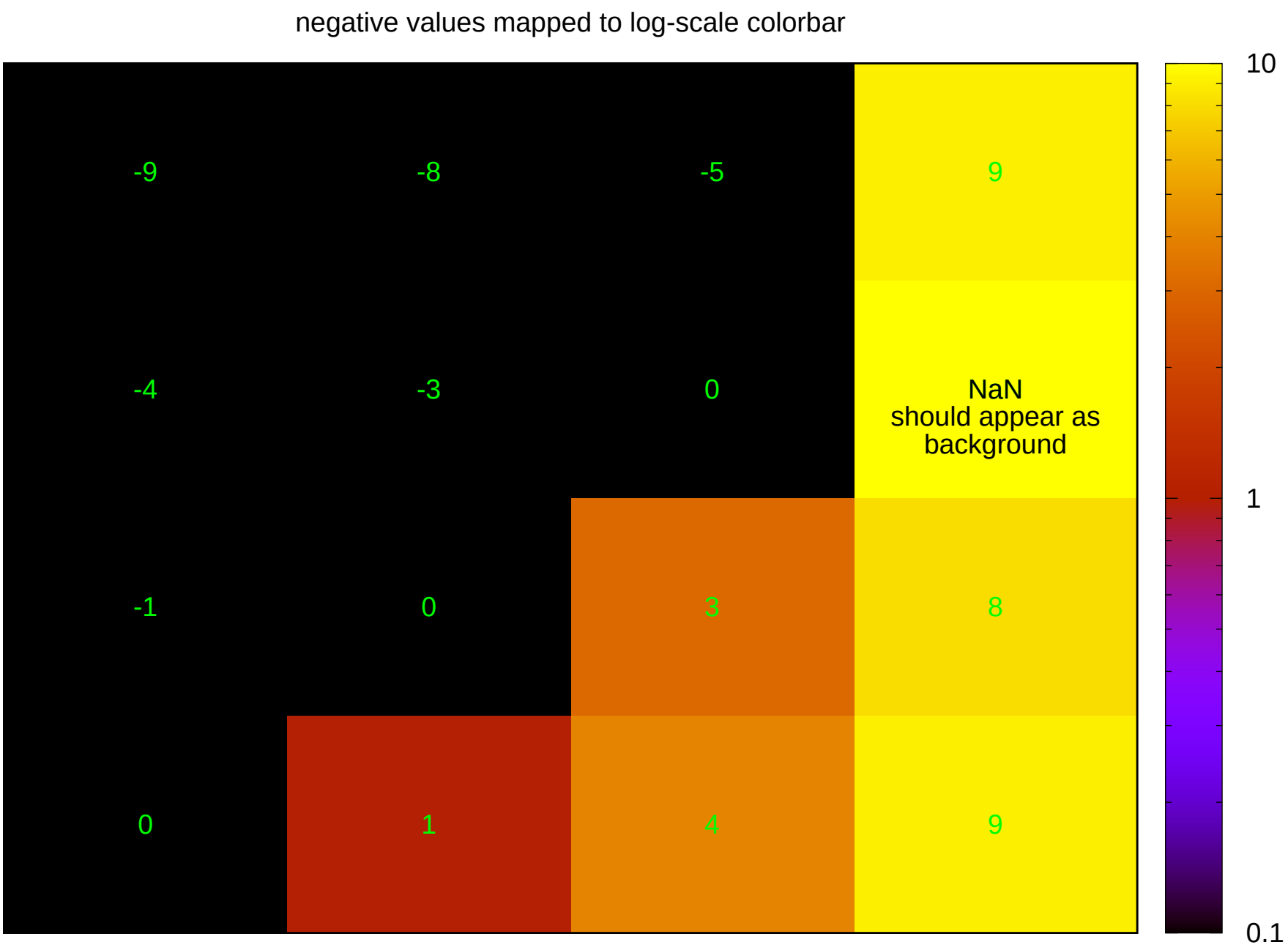
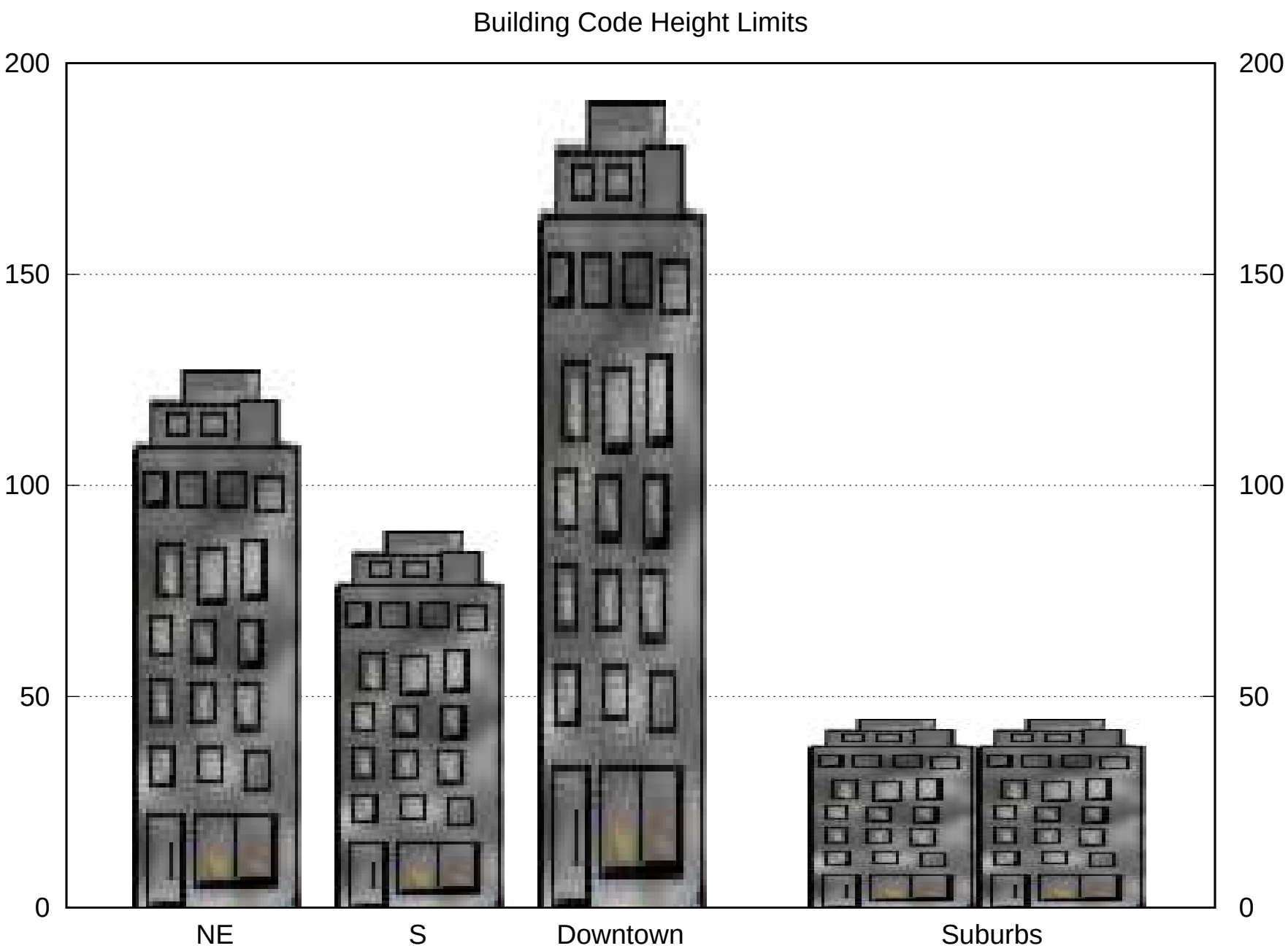


image from non-matrix data
NaN
should appear as
background
10
8
6
4
2
0
-2
-4
-6
-8
-10


negative values mapped to log-scale colorbar
-9
-8
-5
9
-4
-3
0
NaN
should appear as
background
-1
0
3
8
0
1
4
9
10
1
0.1


Building Code Height Limits
200
150
100
50
0
NE
S
Downtown
Suburbs

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

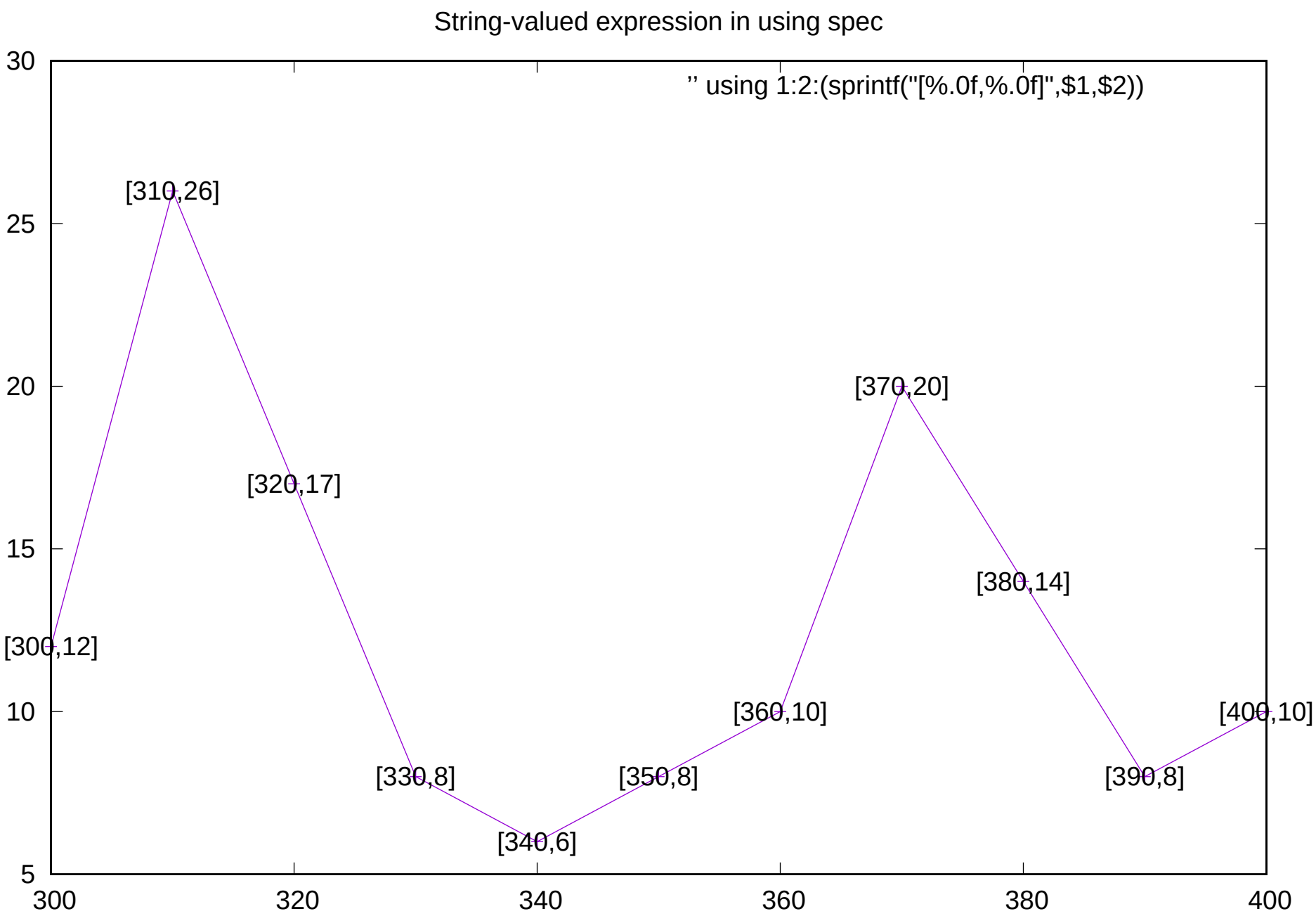
```
foo          = ABCDEF
strlen(foo)  = 6
substr(foo,3,4) = CD

haystack     = ‘date‘
haystack     = Fri Oct  7 14:41:11 JST 2016
needle       = :
S = strstrt(haystack,needle) = 14
haystack[S-2:S+2] = 14:41
It is now 14:41

words(haystack)   = 6
word(haystack,5)  = JST

sprintf output of long strings works OK
```

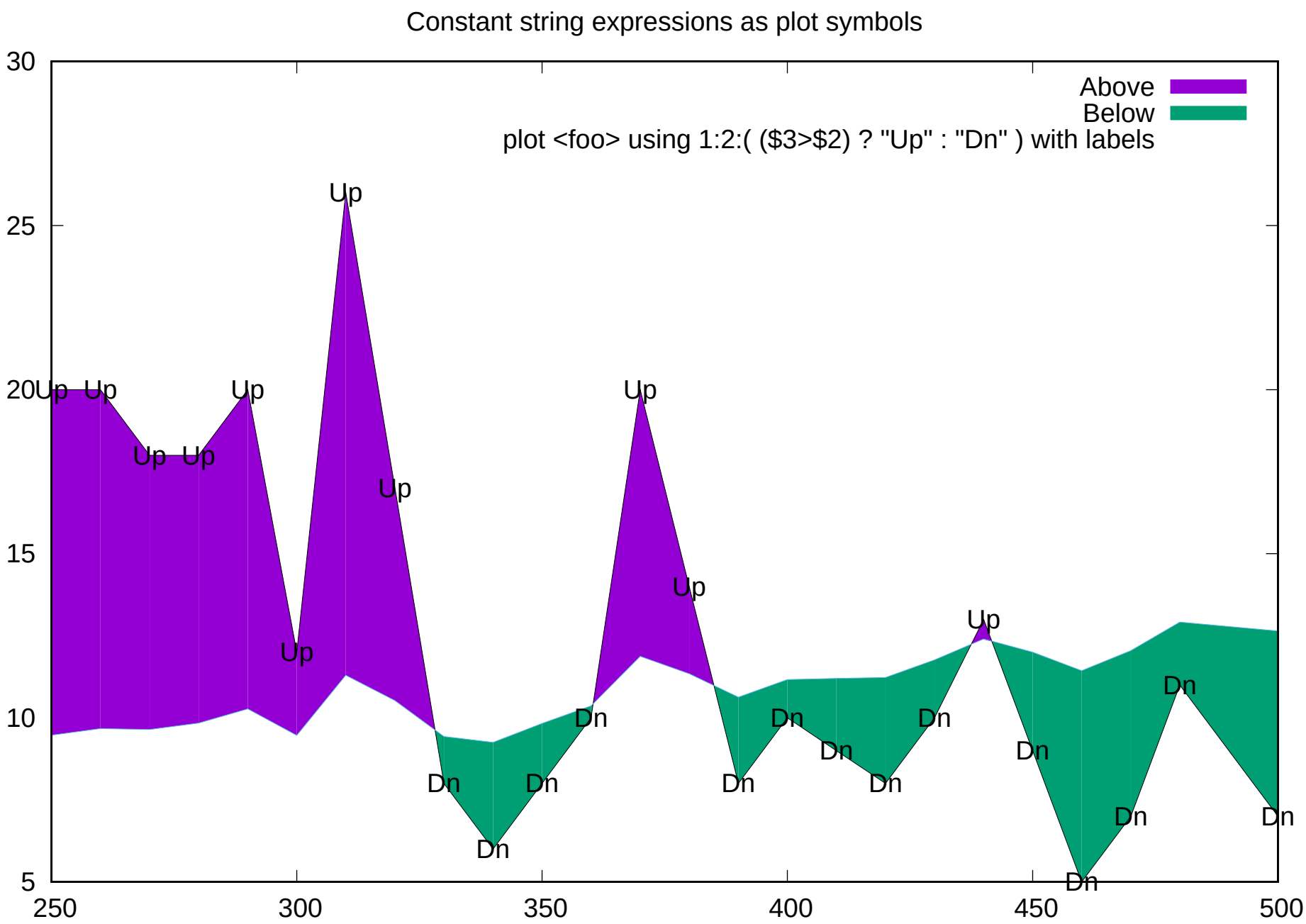
Exercise word and words functions

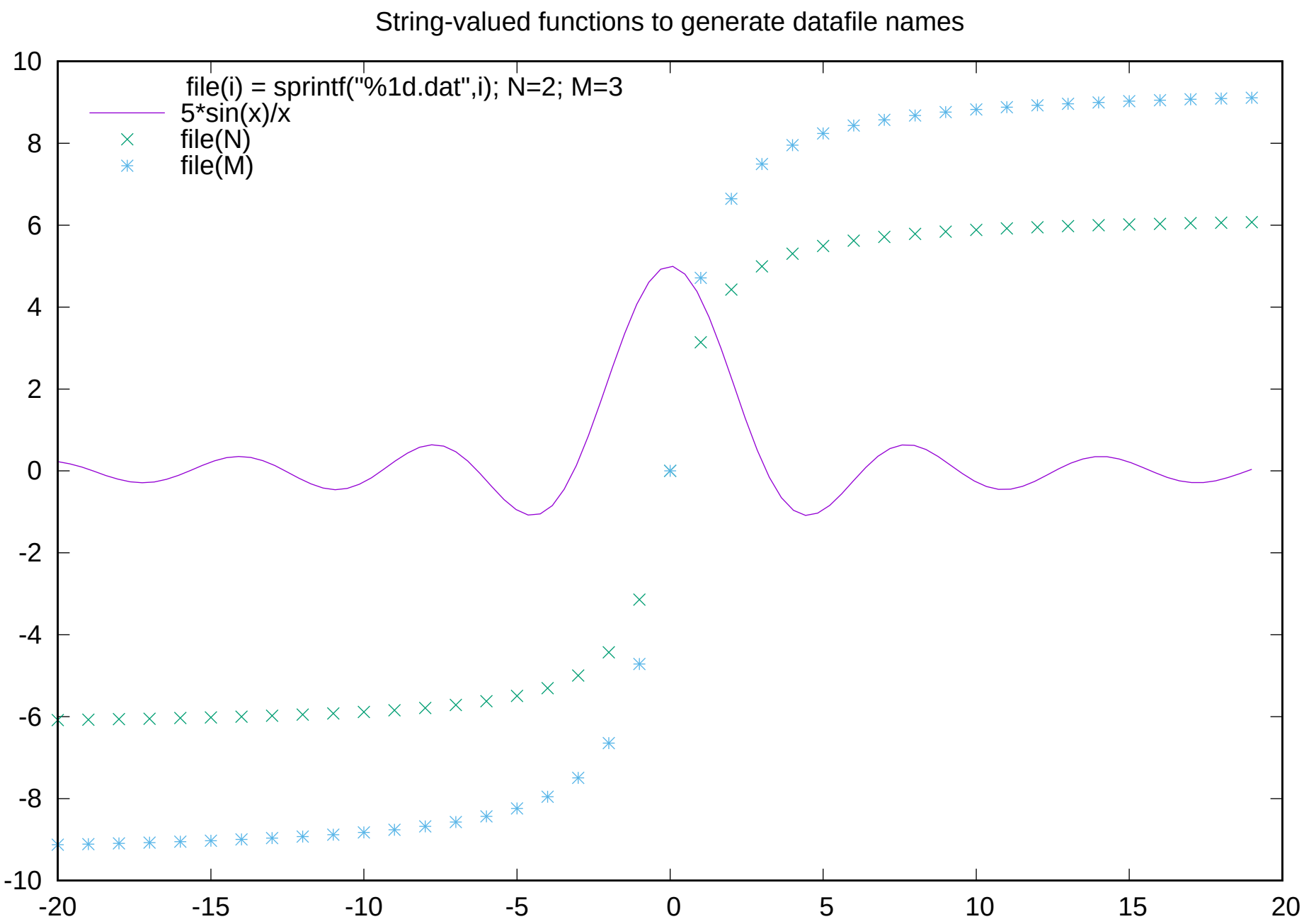
foo = word and words can handle ’quoted string’
words(foo) = 6
word(foo, 6) = quoted string

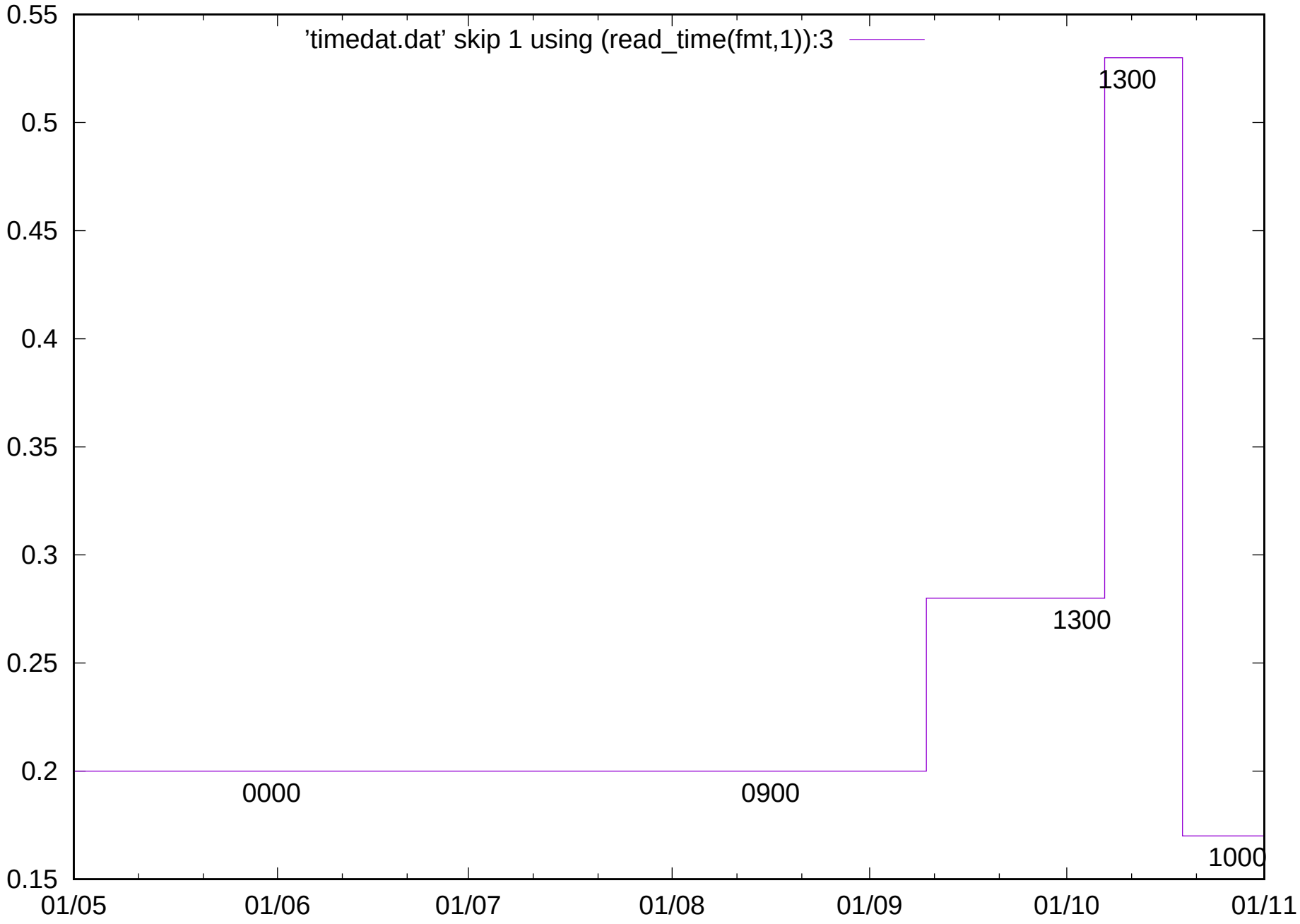
foo = "double quotes" or ’single quotes’
words(foo) = 3

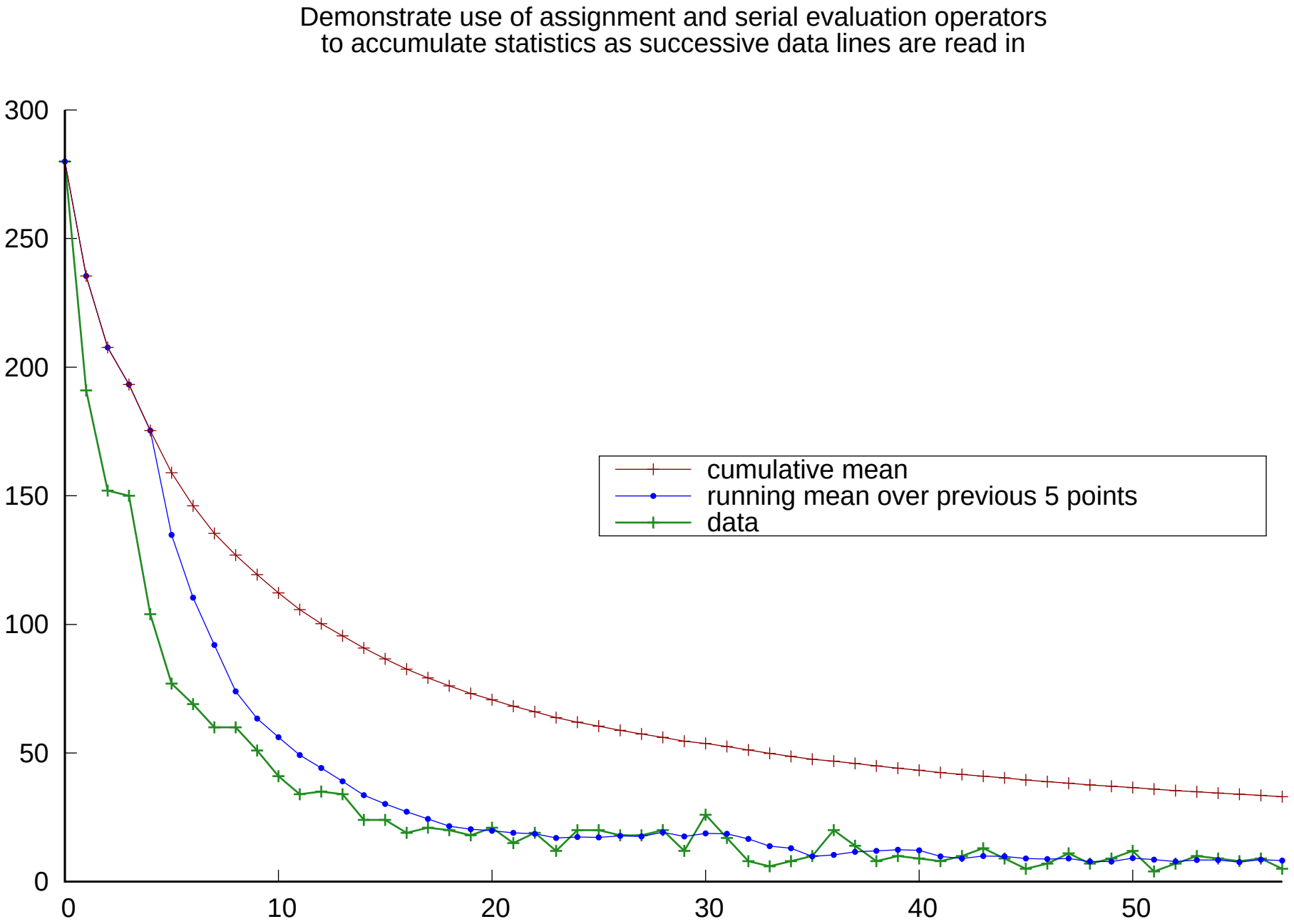
foo = Apostrophes inside words don’t matter
word(foo, 4) = don’t


String-valued expression in using spec
'' using 1:2:(sprintf("[%.0f,%.0f]",$1,$2))
30
25
20
15
10
5
300
320
340
360
380
400
[310,26]
[320,17]
[300,12]
[330,8]
[340,6]
[350,8]
[360,10]
[370,20]
[380,14]
[390,8]
[400,10]

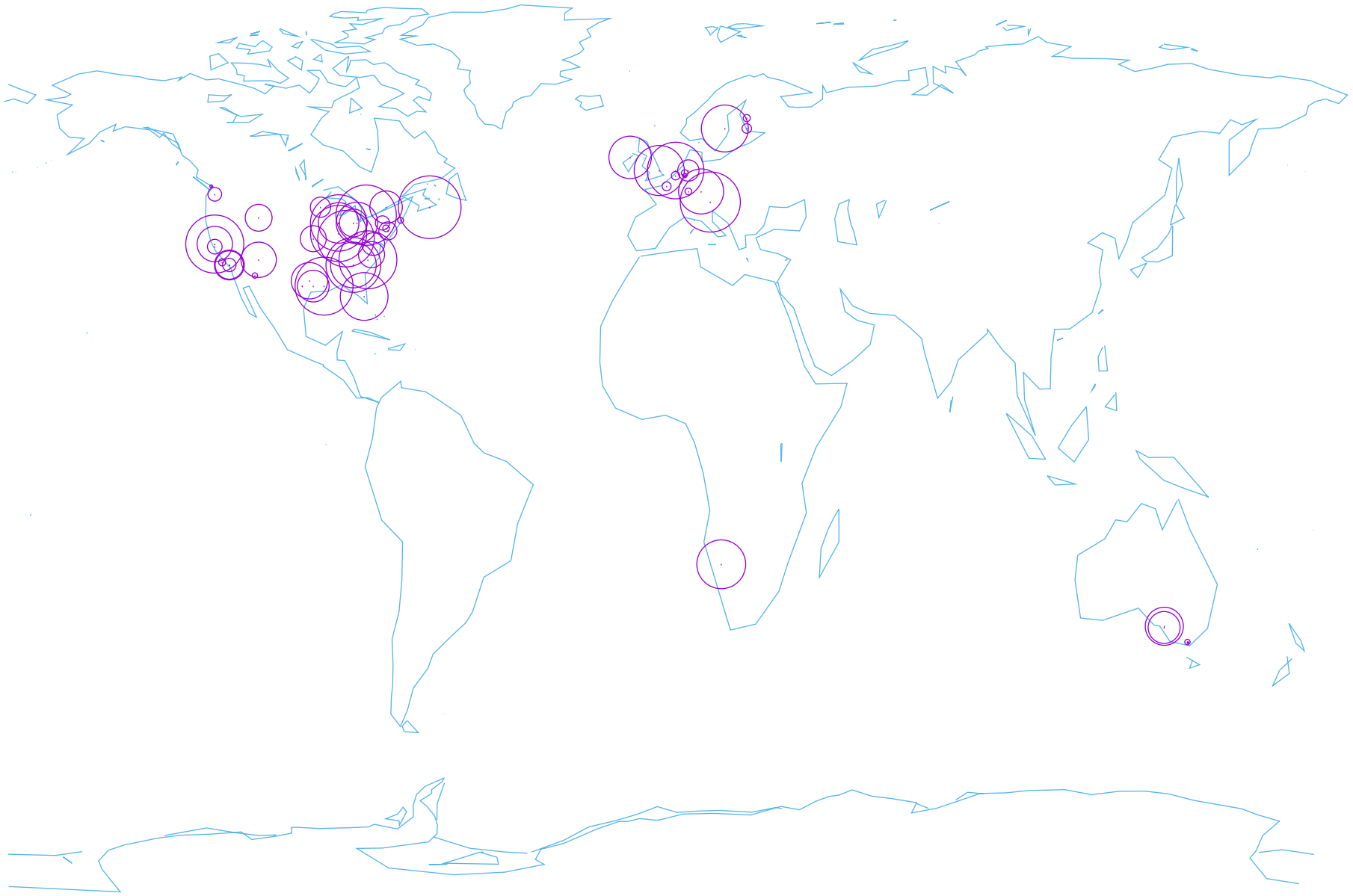

Constant string expressions as plot symbols
plot <foo> using 1:2:(($3>$2) ? "Up" : "Dn") with labels
Above
Below
30
25
20
15
10
5
250
300
350
400
450
500
Up
Up
Up
Up
Up
Up
Up
Up
Up
Up
Up
Dn
Dn
Dn
Dn
Dn
Dn
Dn
Dn
Dn
Dn
Dn
Dn
Dn

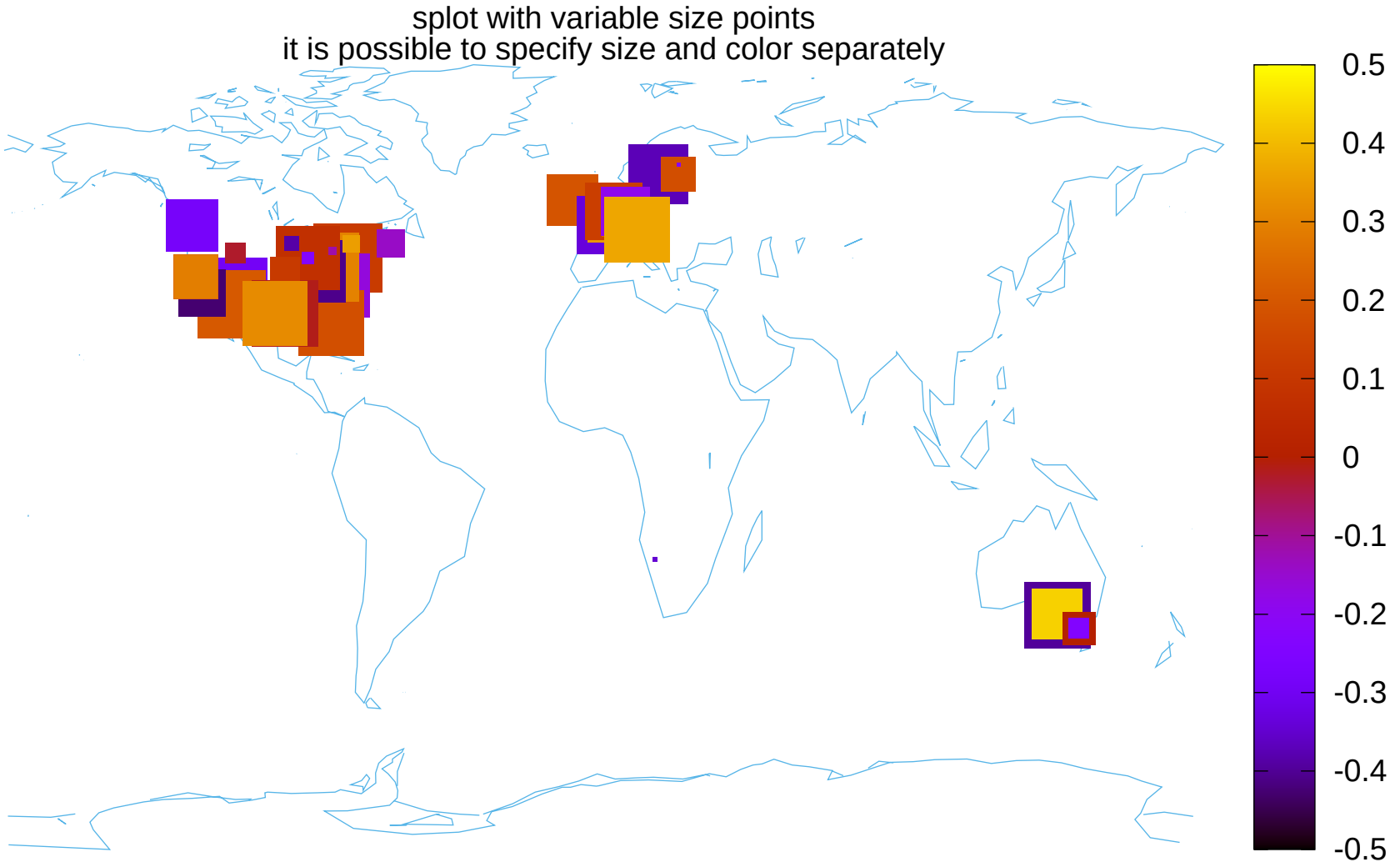

String-valued functions to generate datafile names
file(i) = sprintf("%1d.dat",i); N=2; M=3
5*sin(x)/x
file(N)
file(M)
10
8
6
4
2
0
-2
-4
-6
-8
-10
-20
-15
-10
-5
0
5
10
15
20


'timedat.dat' skip 1 using (read_time(fmt,1)):3
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
01/05
01/06
01/07
01/08
01/09
01/10
01/11
0000
0900
1300
1300
1000

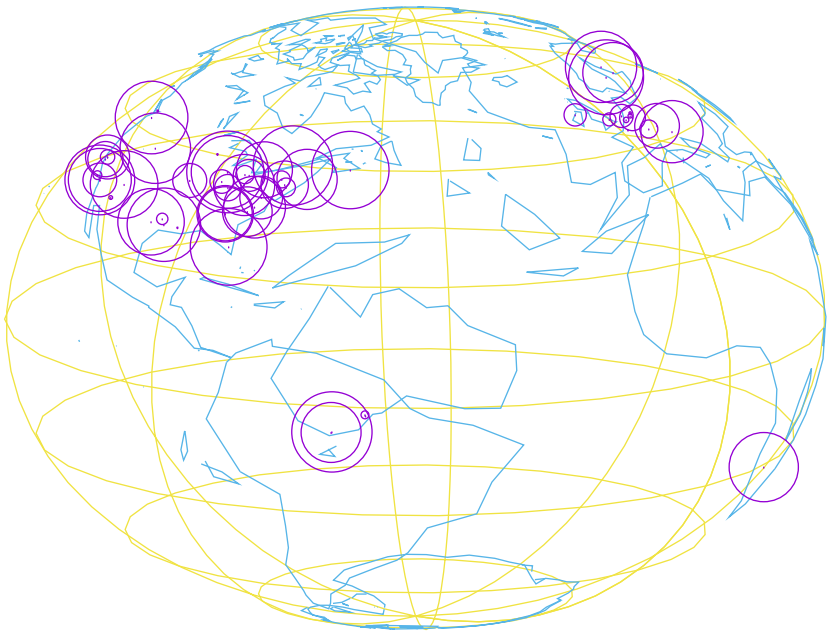

Demonstrate use of assignment and serial evaluation operators
to accumulate statistics as successive data lines are read in
300
250
200
150
100
50
0
0
10
20
30
40
50
cumulative mean
running mean over previous 5 points
data

plot with variable size points

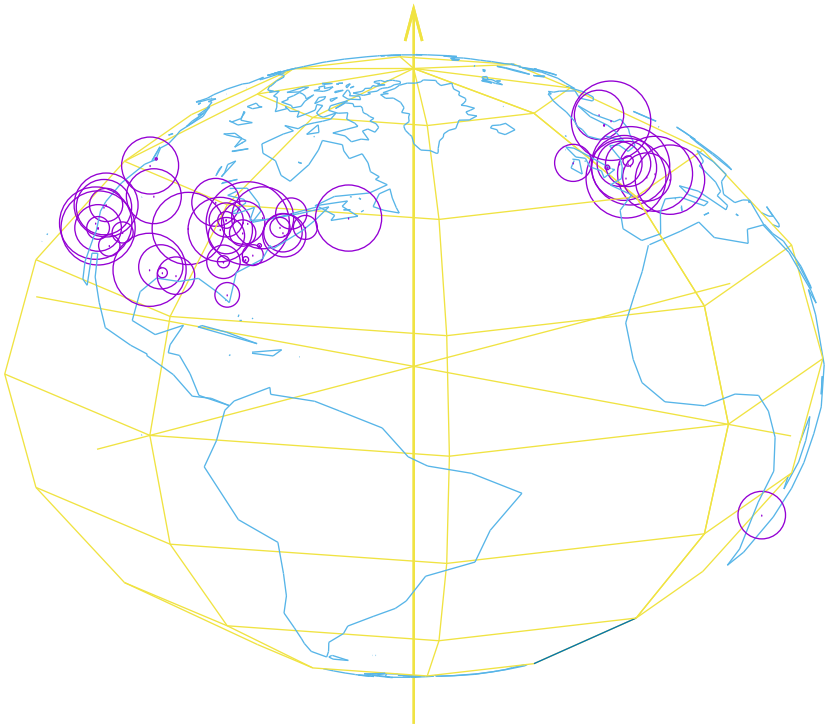


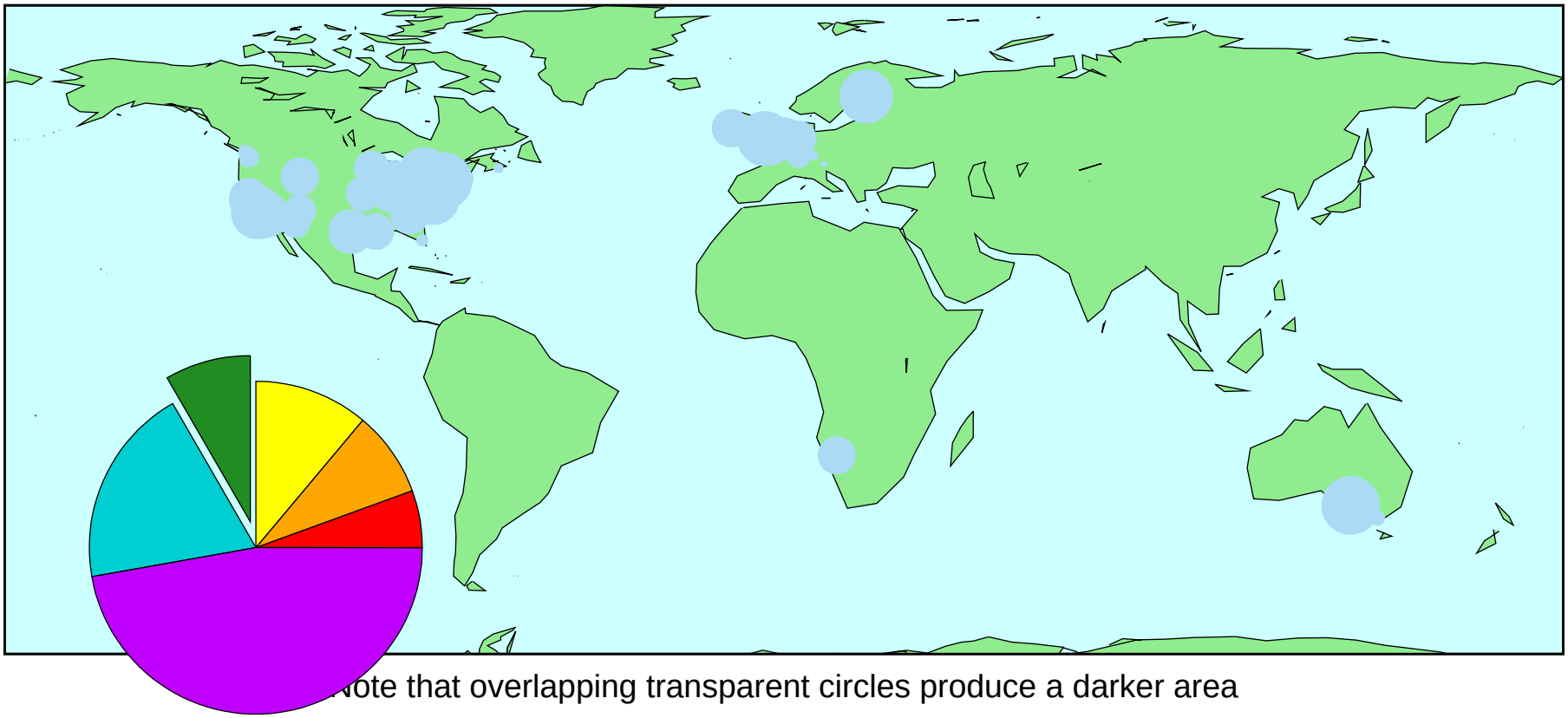

splot with variable size points
it is possible to specify size and color separately
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5

3D version using spherical coordinate system



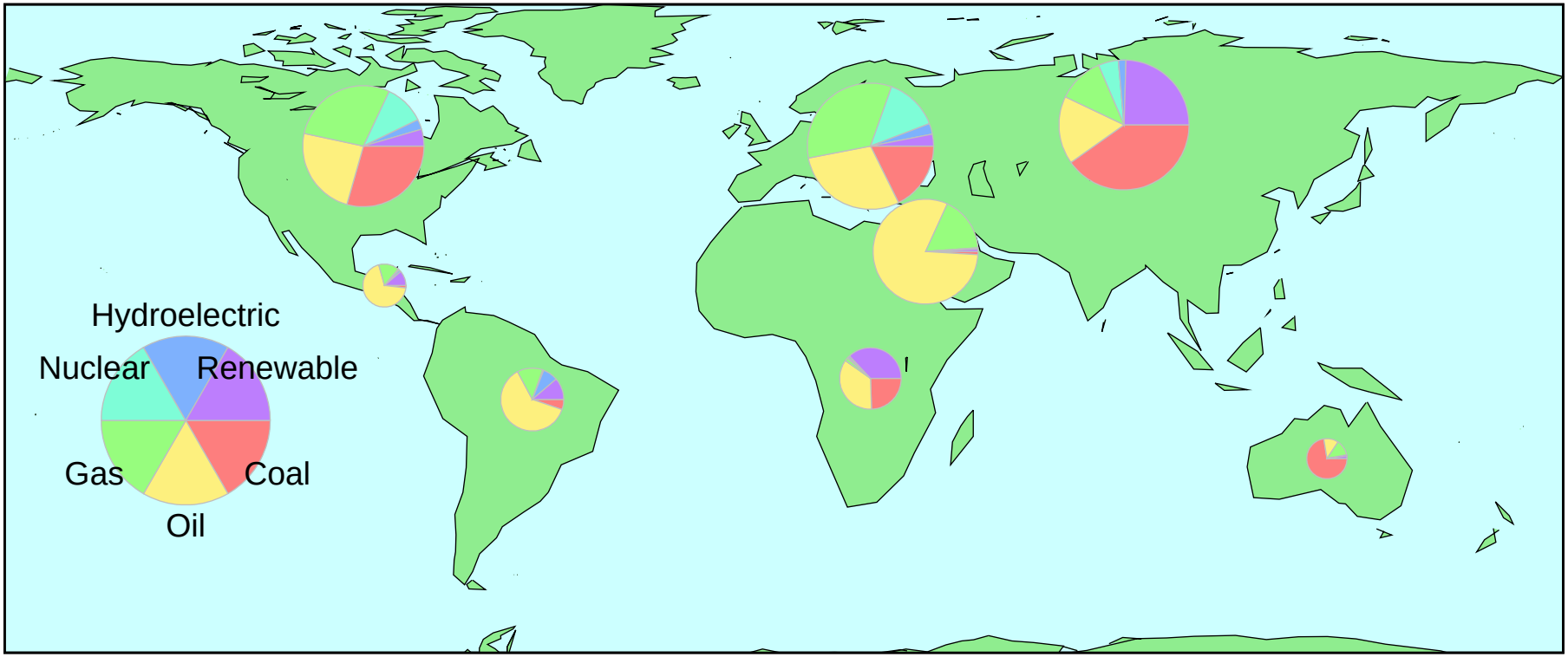
3D solid version through hiddenlining

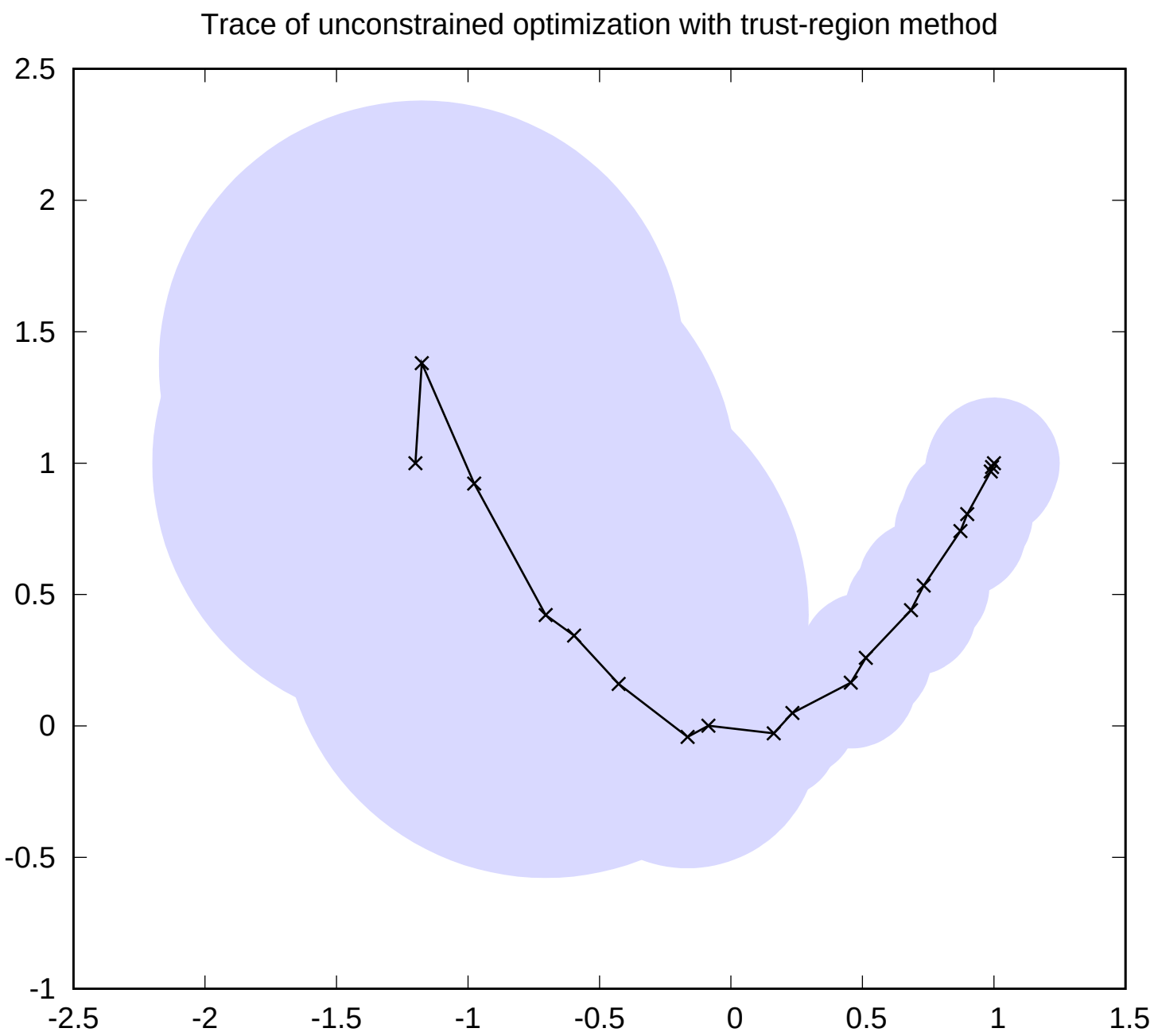




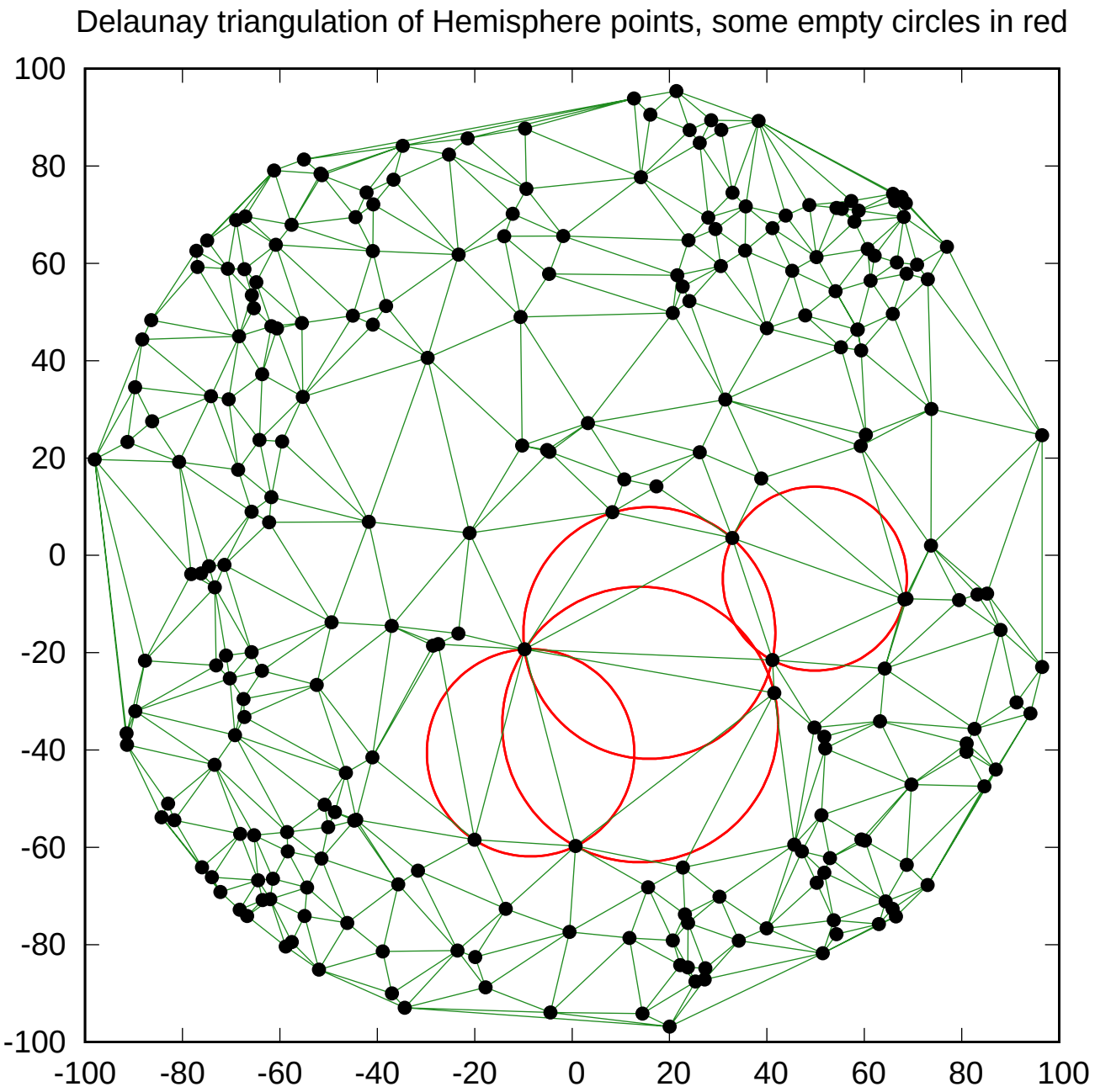
Note that overlapping transparent circles produce a darker area

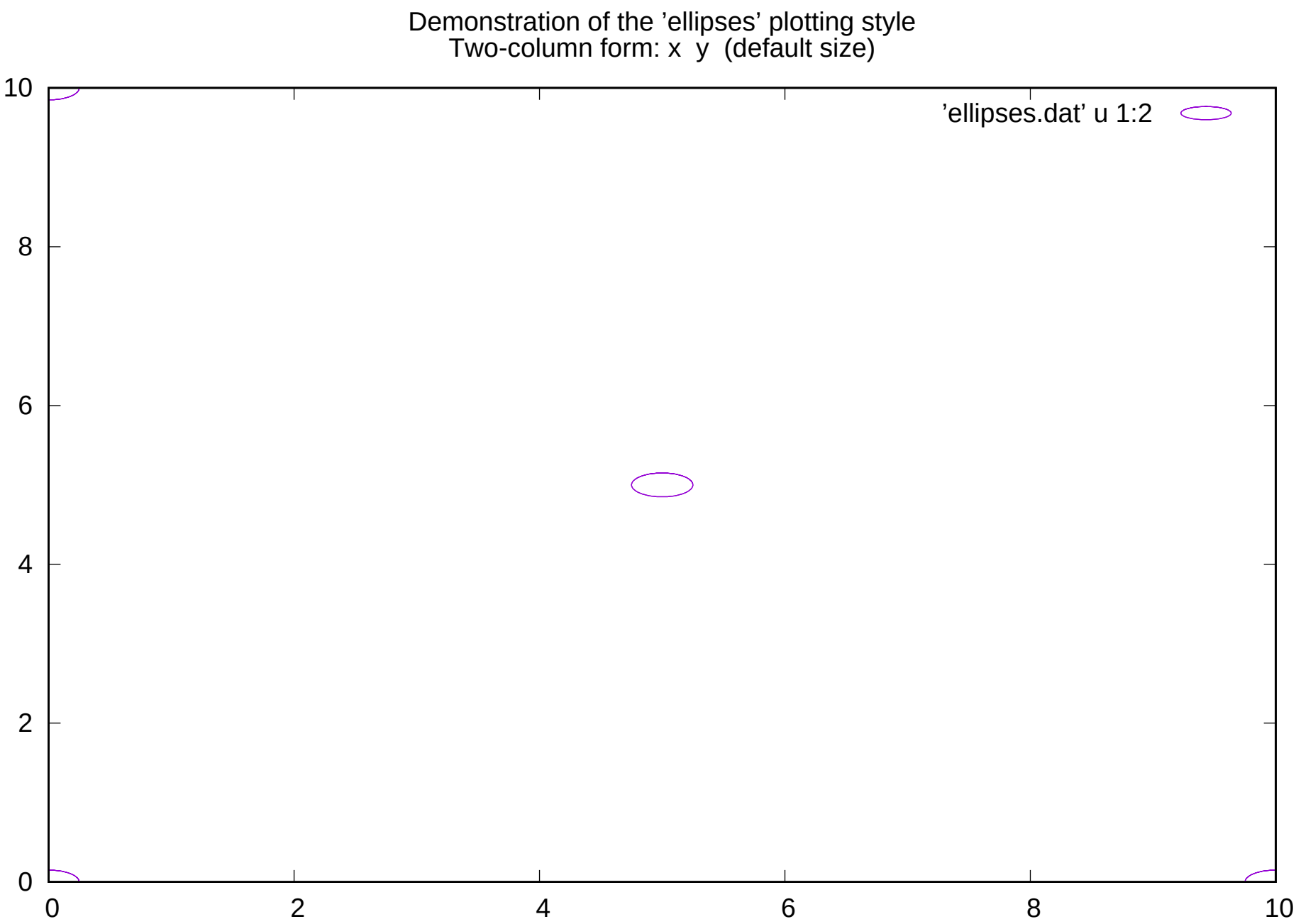
Sources of energy production, plotted for each continent

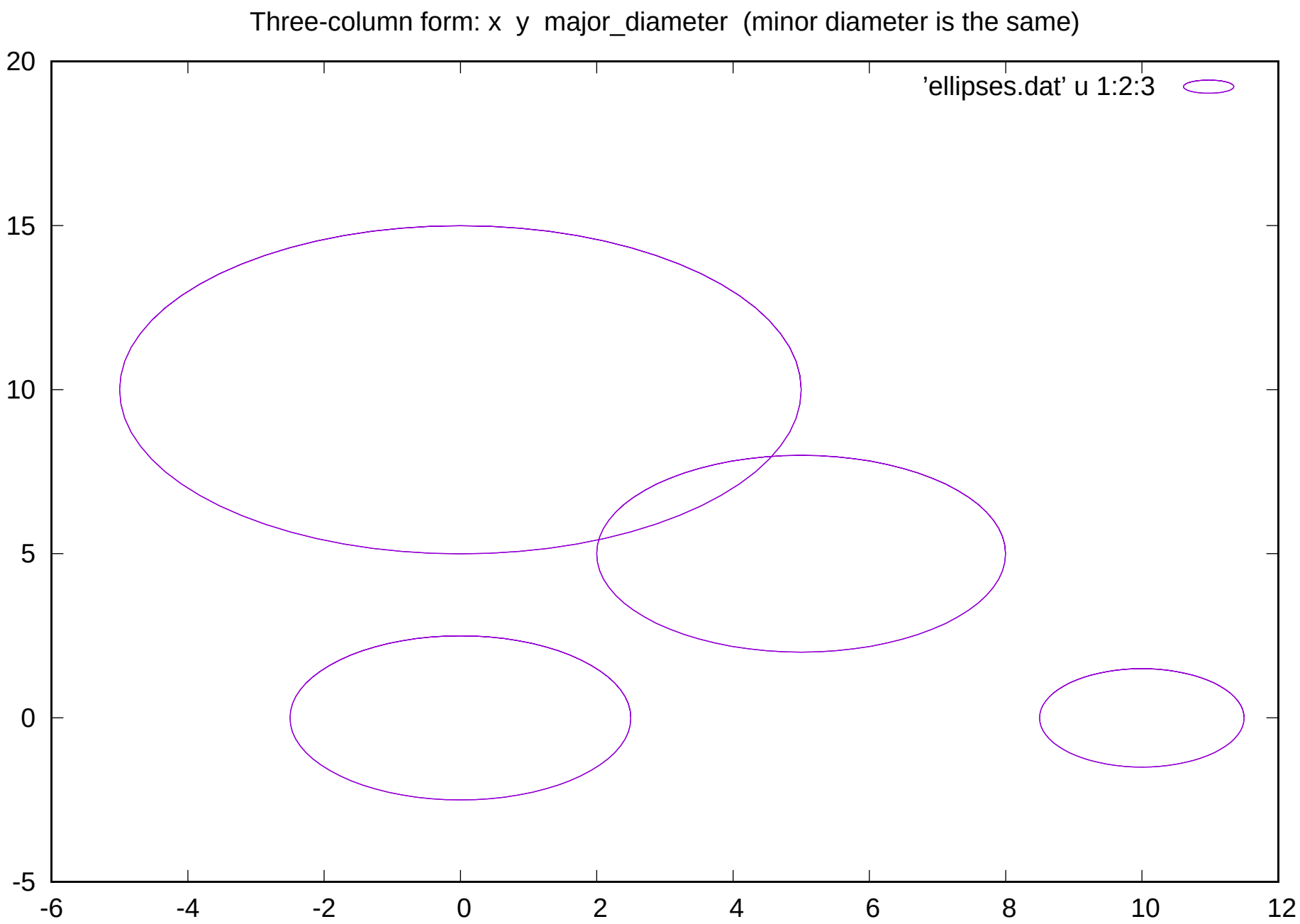


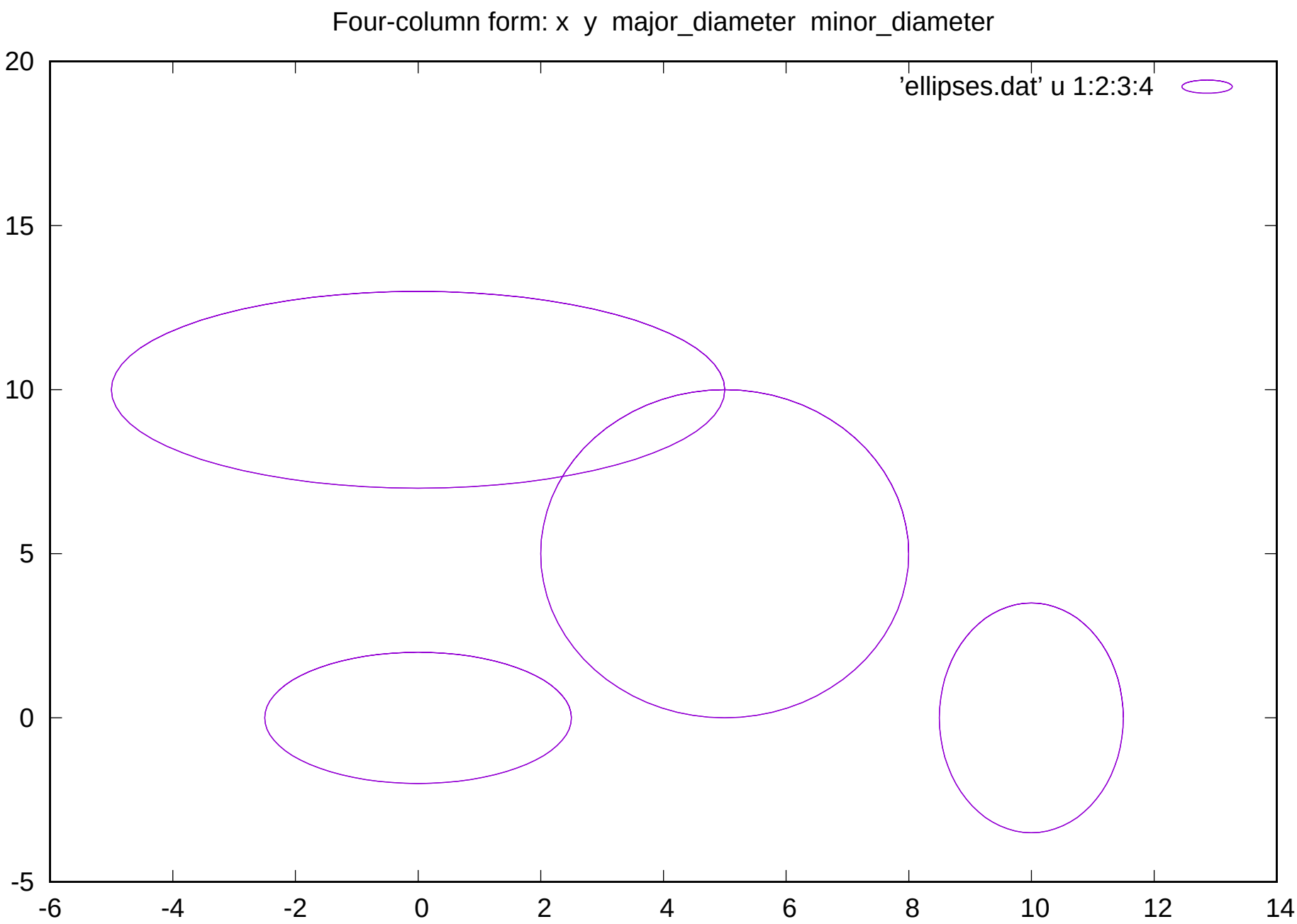

Trace of unconstrained optimization with trust-region method
2.5
2
1.5
1
0.5
0
-0.5
-1
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5

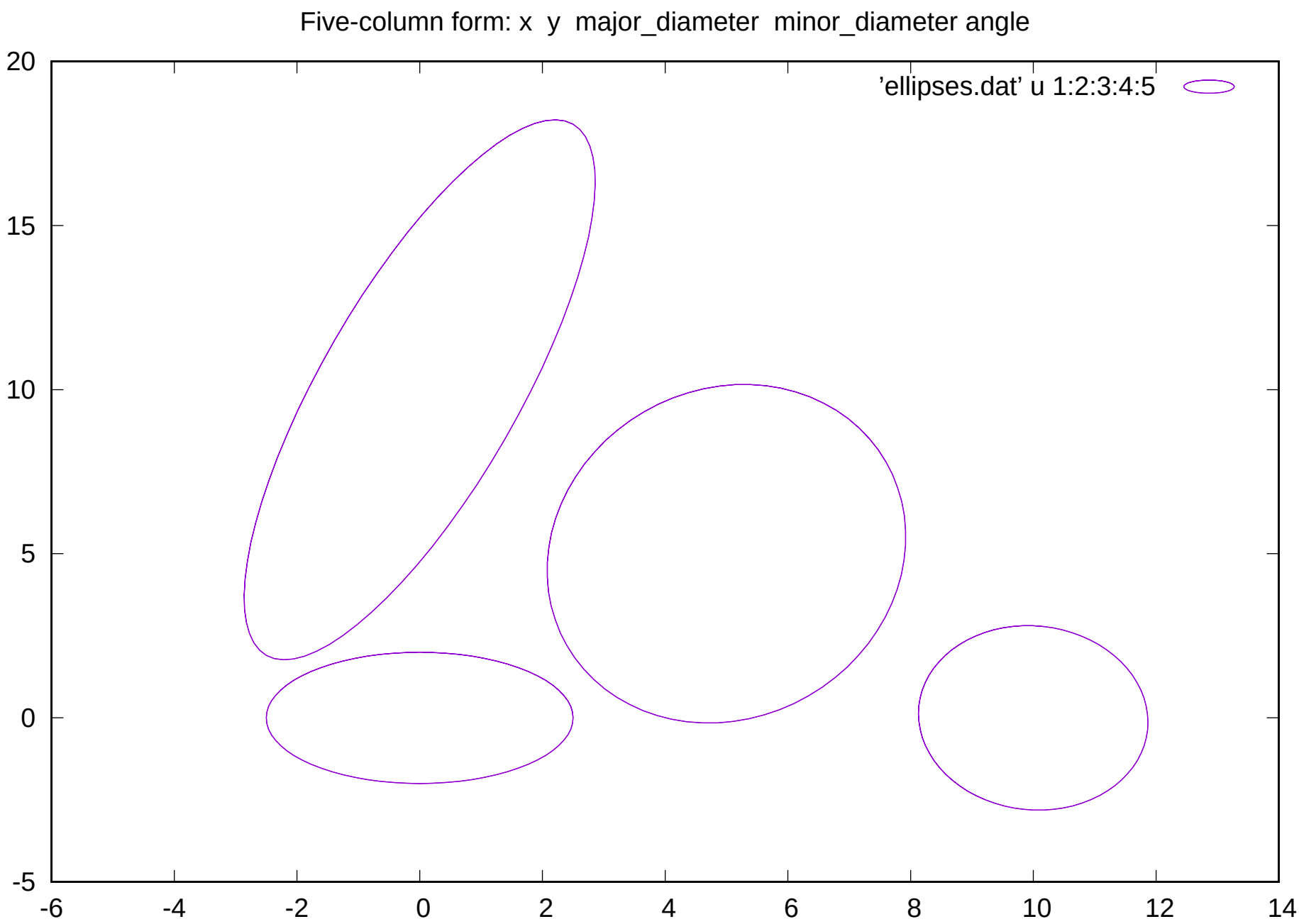
Lena’s key points

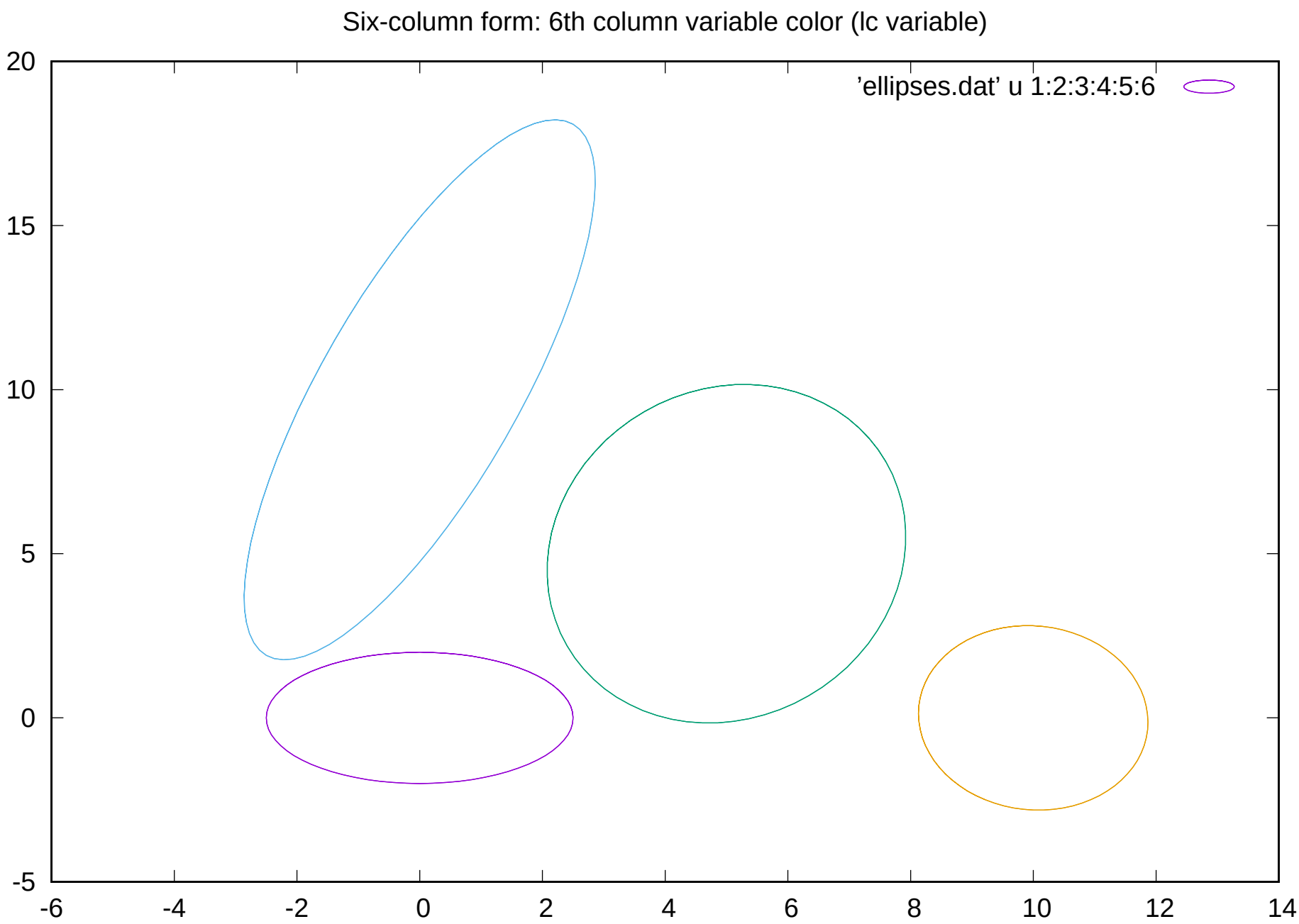

Delaunay triangulation of Hemisphere points, some empty circles in red
100
80
60
40
20
0
-20
-40
-60
-80
-100
-100
-80
-60
-40
-20
0
20
40
60
80
100

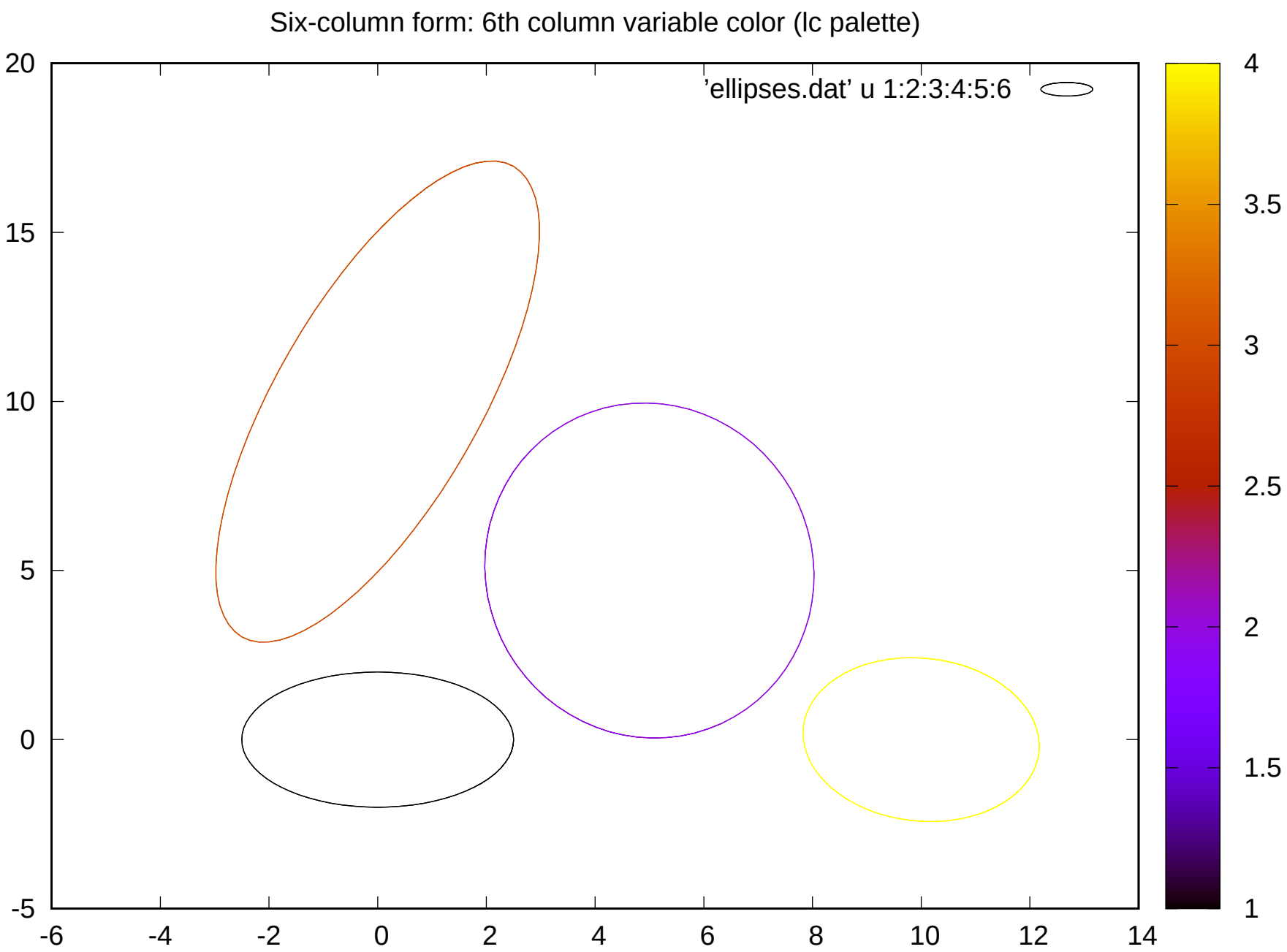

Demonstration of the ’ellipses’ plotting style
Two-column form: x y (default size)
’ellipses.dat’ u 1:2
10
8
6
4
2
0
0
2
4
6
8
10

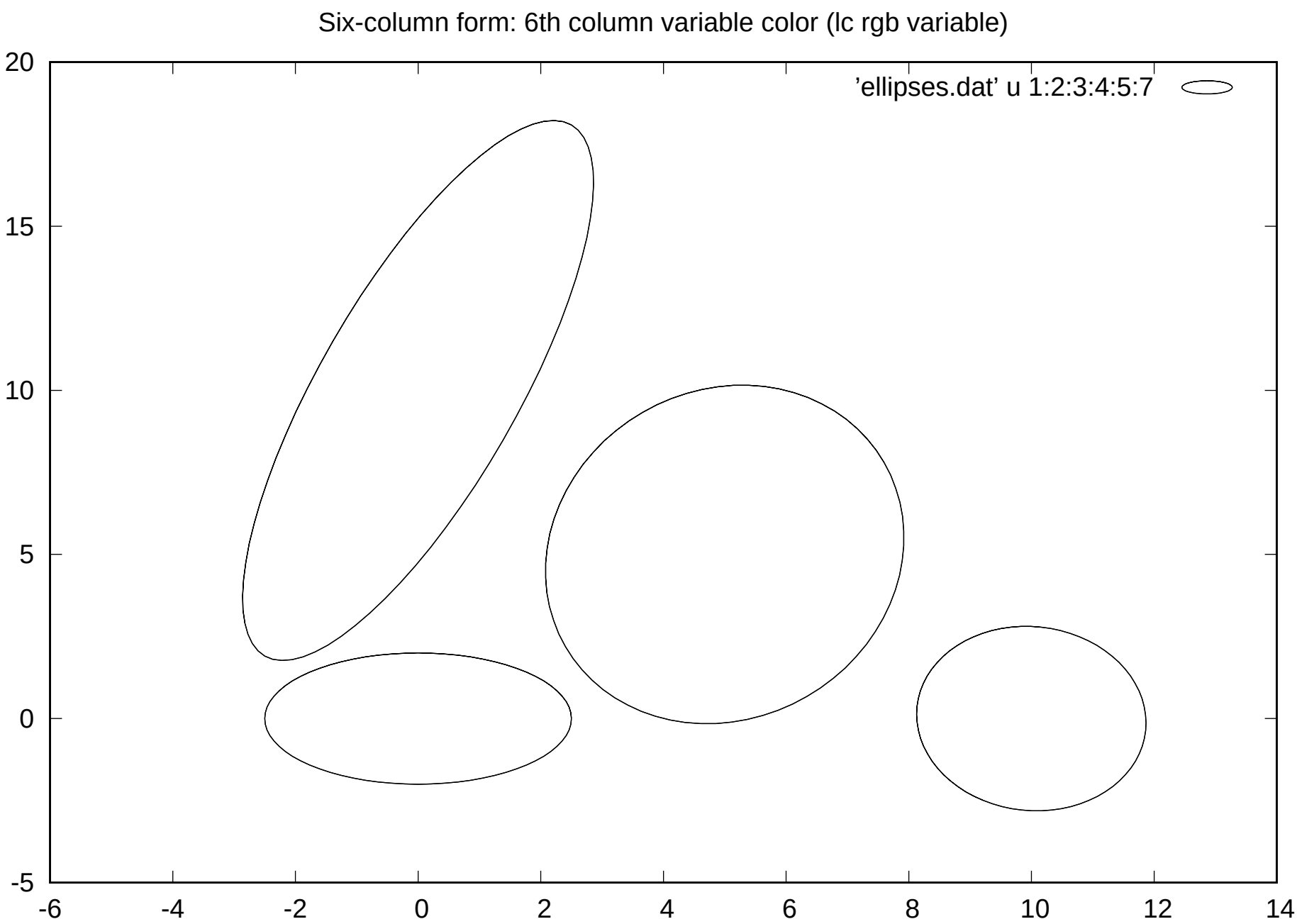

Three-column form: x y major_diameter (minor diameter is the same)
'ellipses.dat' u 1:2:3
20
15
10
5
0
-5
-6
-4
-2
0
2
4
6
8
10
12

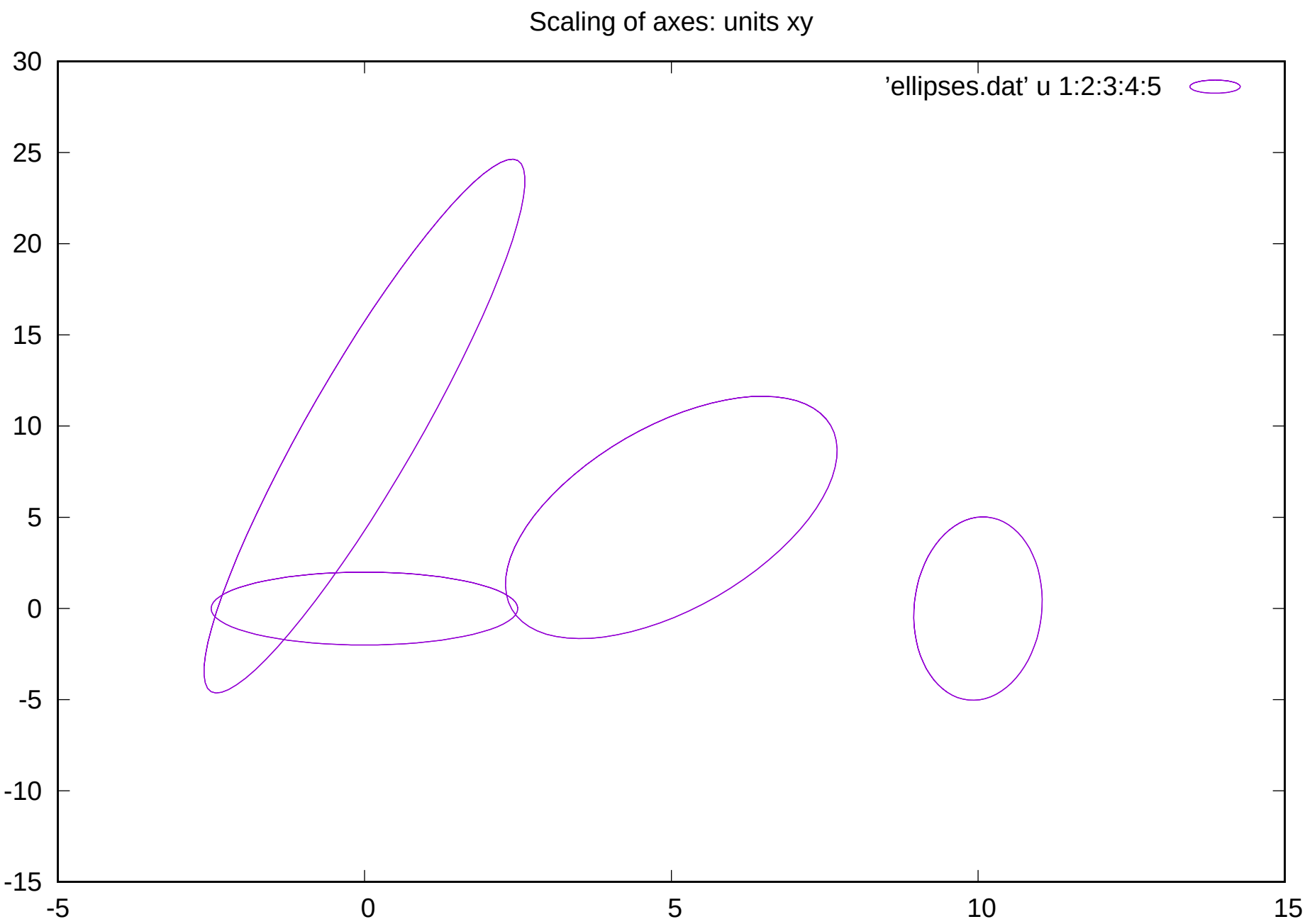

Four-column form: x y major_diameter minor_diameter
'ellipses.dat' u 1:2:3:4
20
15
10
5
0
-5
-6
-4
-2
0
2
4
6
8
10
12
14

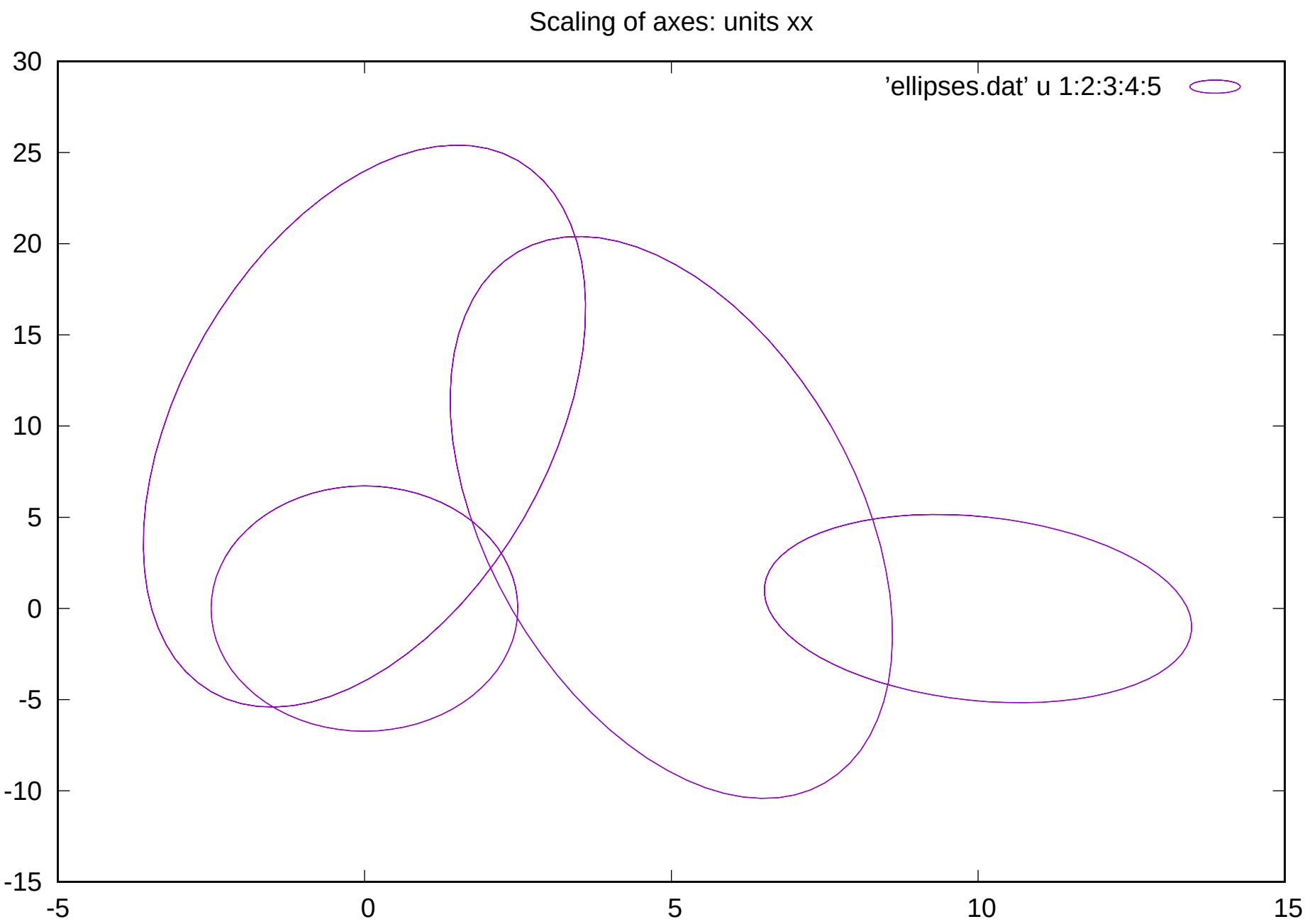

Five-column form: x y major_diameter minor_diameter angle
'ellipses.dat' u 1:2:3:4:5
20
15
10
5
0
-5
-6
-4
-2
0
2
4
6
8
10
12
14

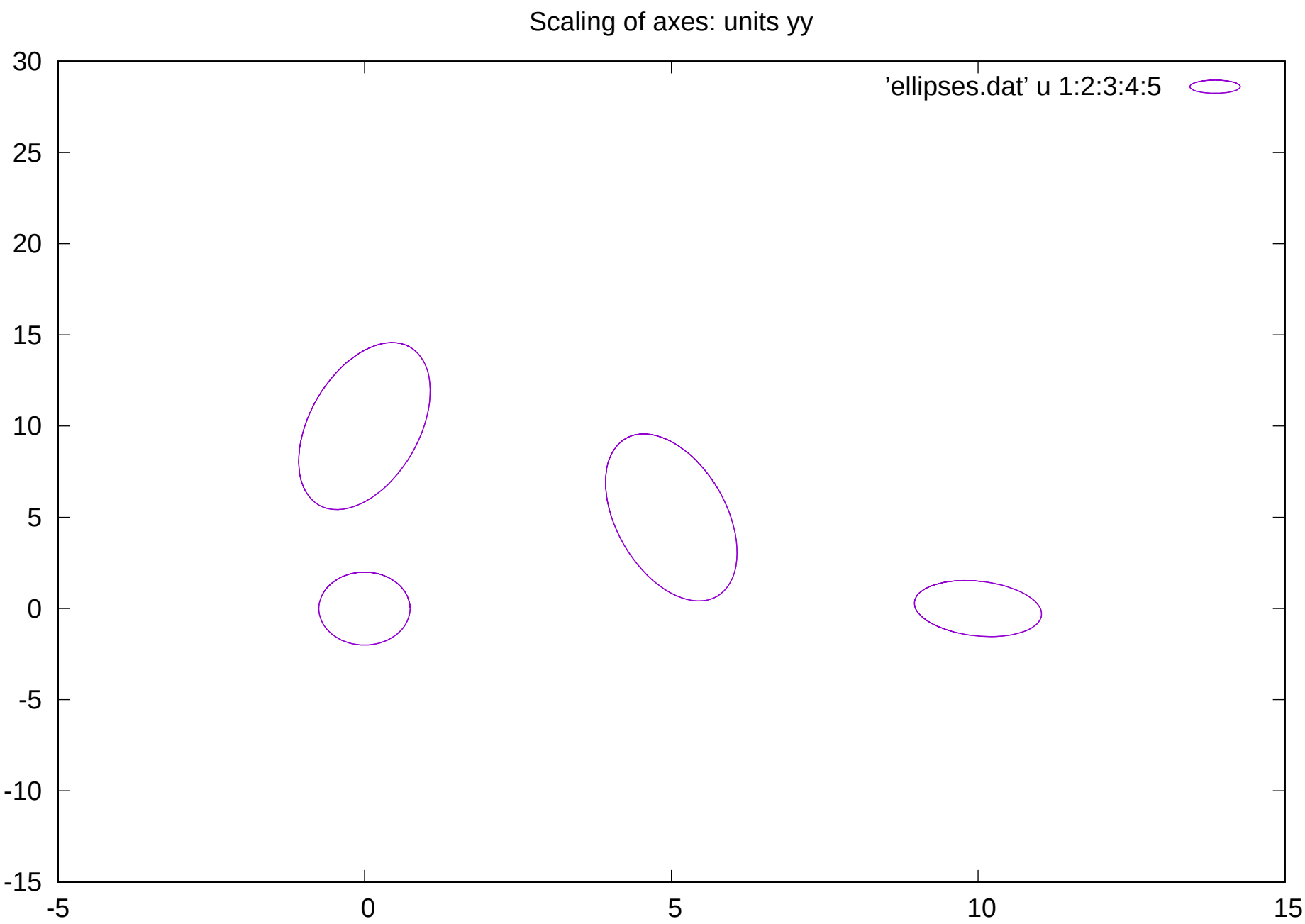

Six-column form: 6th column variable color (lc variable)
'ellipses.dat' u 1:2:3:4:5:6
20
15
10
5
0
-5
-6
-4
-2
0
2
4
6
8
10
12
14

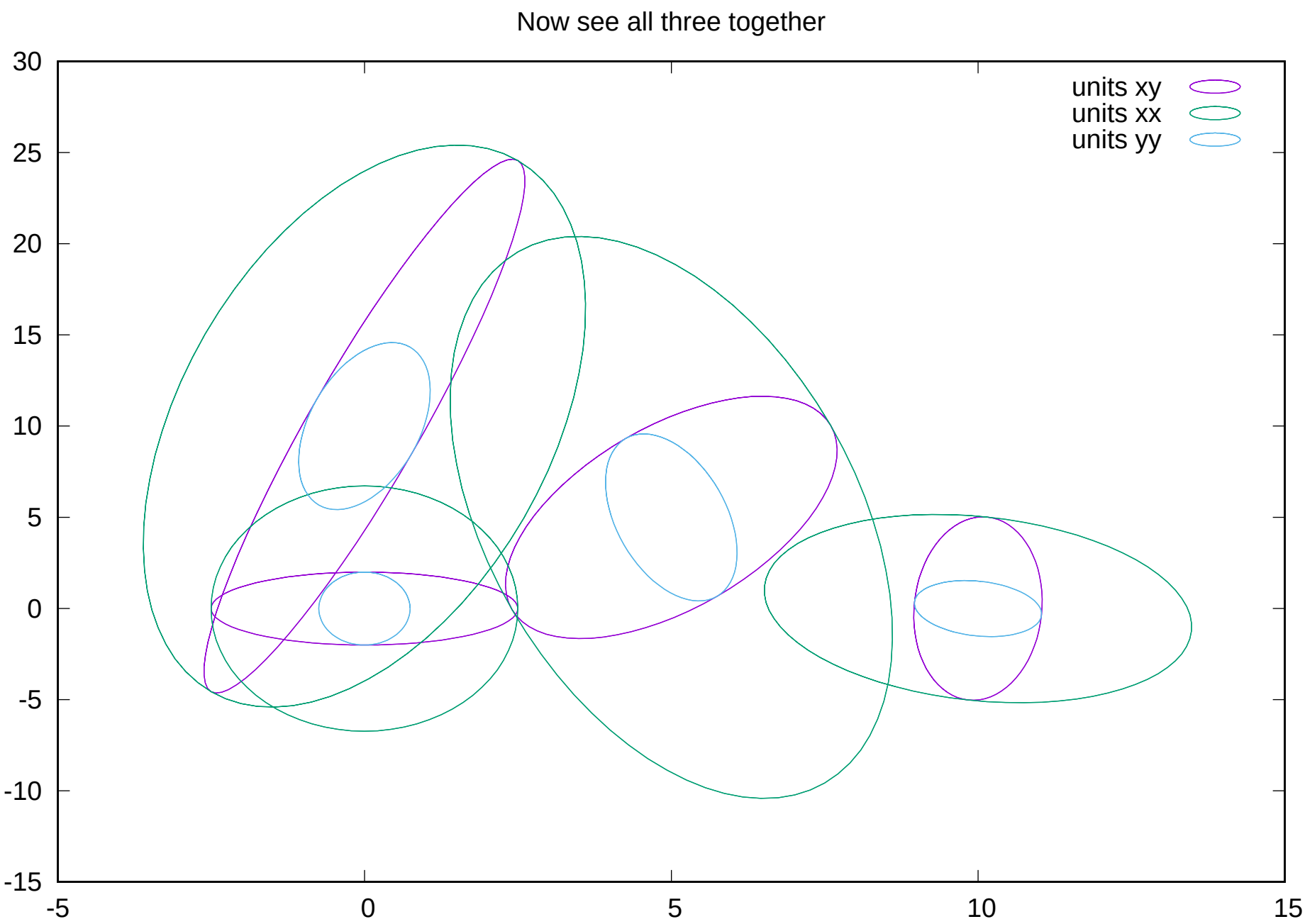

Six-column form: 6th column variable color (lc palette)
'ellipses.dat' u 1:2:3:4:5:6
20
15
10
5
0
-5
-6
-4
-2
0
2
4
6
8
10
12
14
4
3.5
3
2.5
2
1.5
1


Six-column form: 6th column variable color (lc rgb variable)
'ellipses.dat' u 1:2:3:4:5:7
20
15
10
5
0
-5
-6
-4
-2
0
2
4
6
8
10
12
14

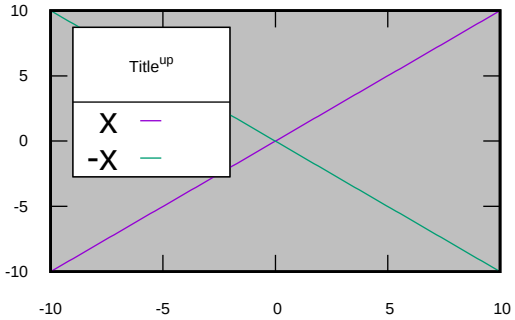

Scaling of axes: units xy
'ellipses.dat' u 1:2:3:4:5
30
25
20
15
10
5
0
-5
-10
-15
-5
0
5
10
15


Scaling of axes: units xx
'ellipses.dat' u 1:2:3:4:5
30
25
20
15
10
5
0
-5
-10
-15
-5
0
5
10
15

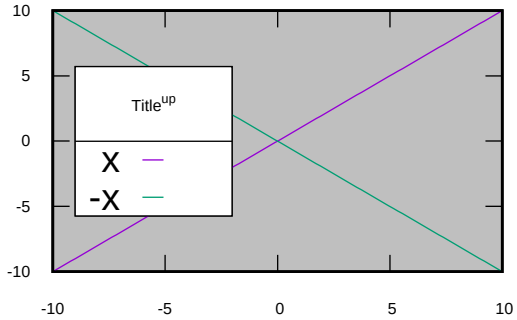

Scaling of axes: units yy
'ellipses.dat' u 1:2:3:4:5
30
25
20
15
10
5
0
-5
-10
-15
-5
0
5
10
15


Now see all three together
units xy
units xx
units yy
30
25
20
15
10
5
0
-5
-10
-15
-5
0
5
10
15

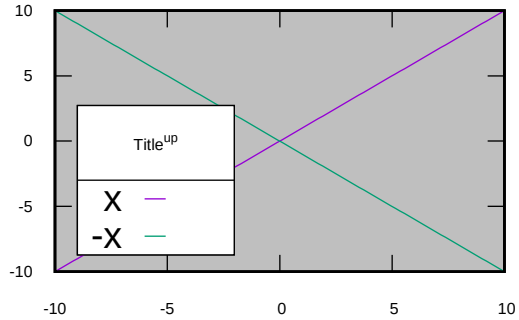
Key (ins vert left top)



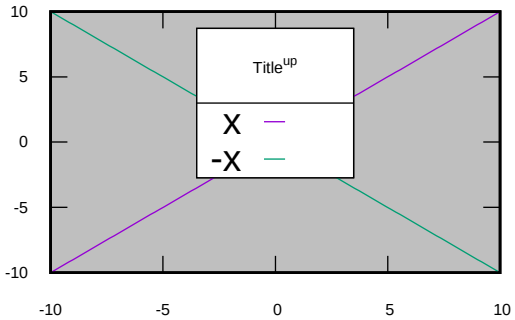
Key (ins vert center left)



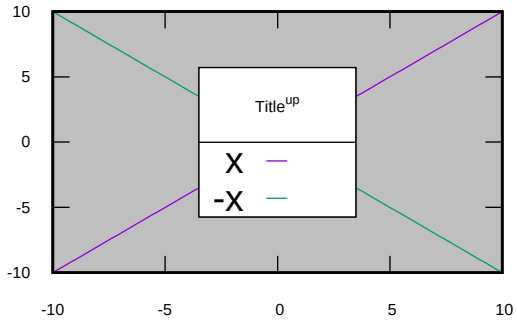
Key (ins vert bot left)



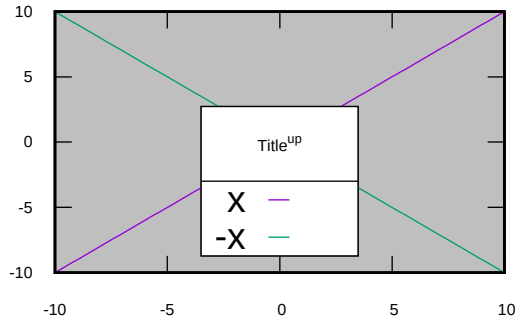
Key (ins vert center top)



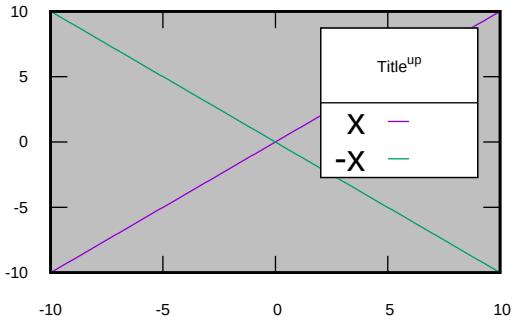
Key (inside vertical center)



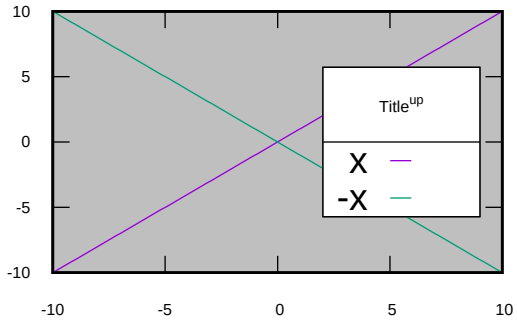
Key (ins vert bot center)



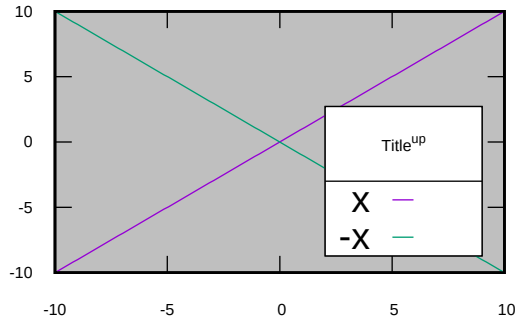
Key (ins vert right top)



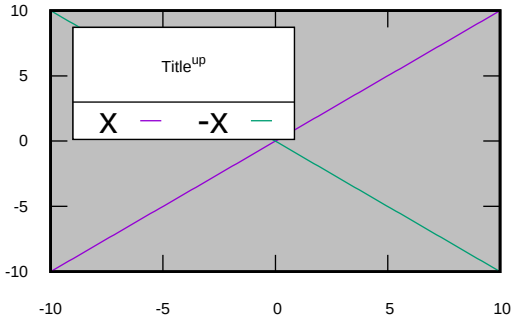
Key (ins vert cent right)



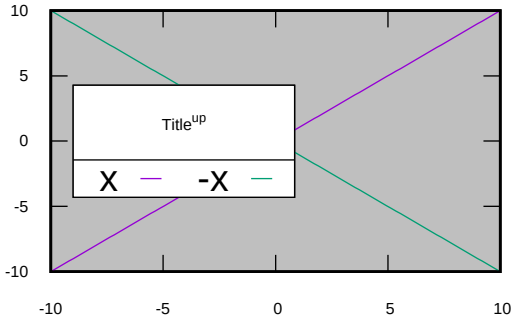
Key (ins vert bot right)



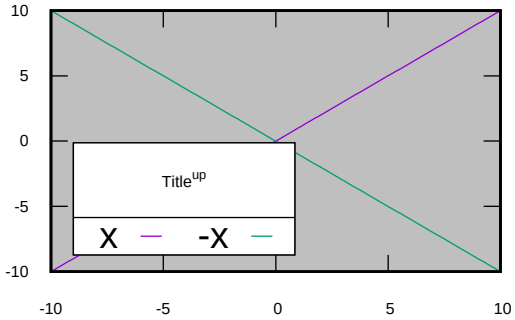
Key (ins horiz left top)



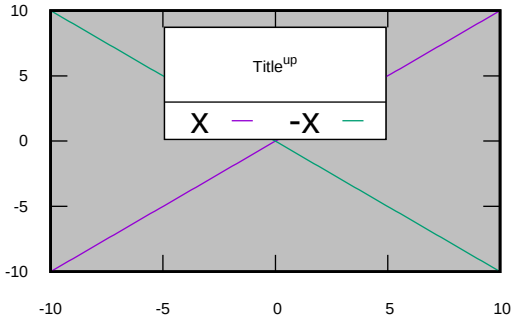
Key (ins horiz center left)



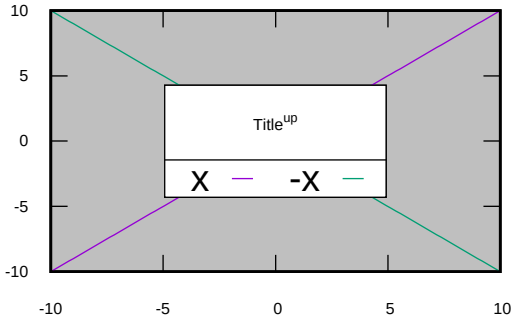
Key (ins horiz bot left)



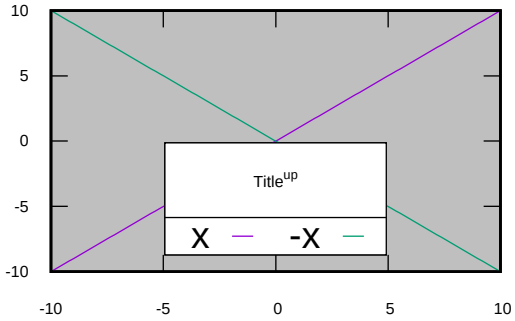
Key (ins horiz center top)



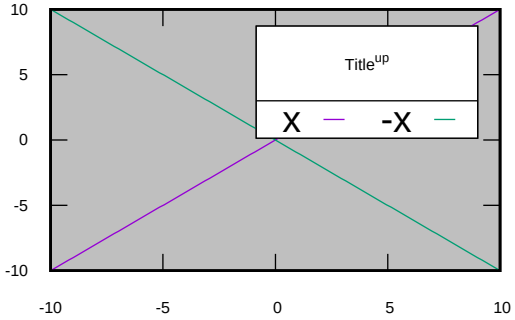
Key (inside horizontal center)



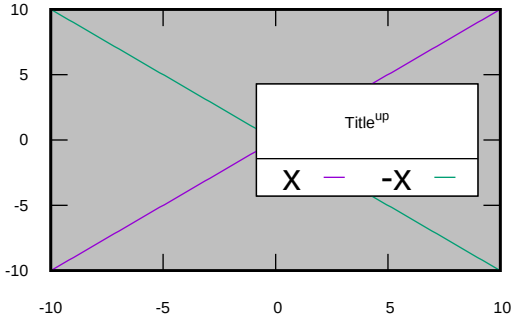
Key (ins horiz bot center)



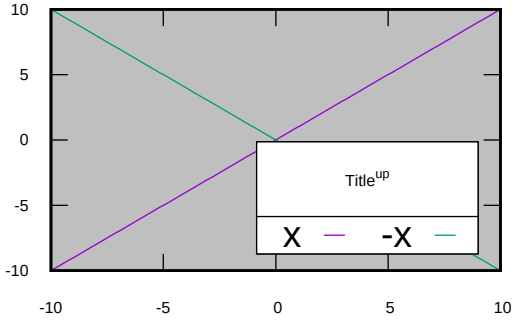
Key (ins horiz right top)



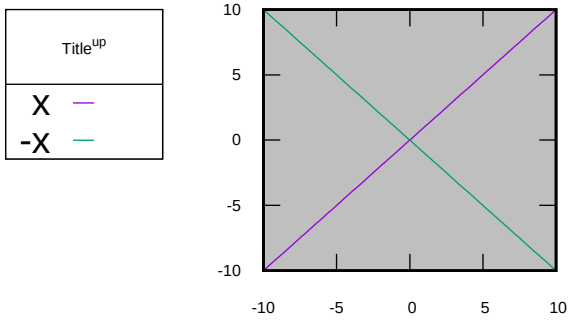
Key (ins horiz cent right)



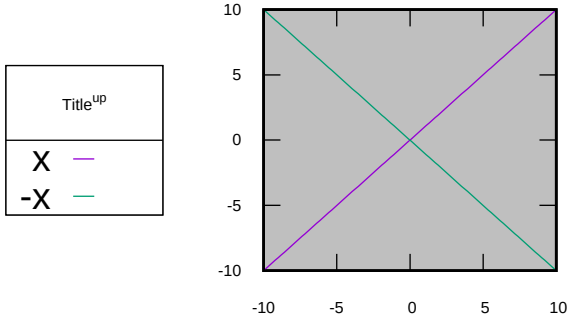
Key (ins horiz bot right)



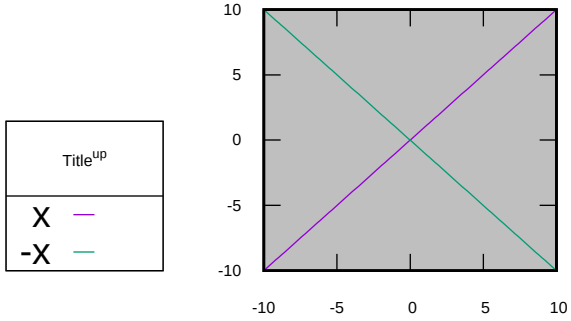
Key (out vert left top)



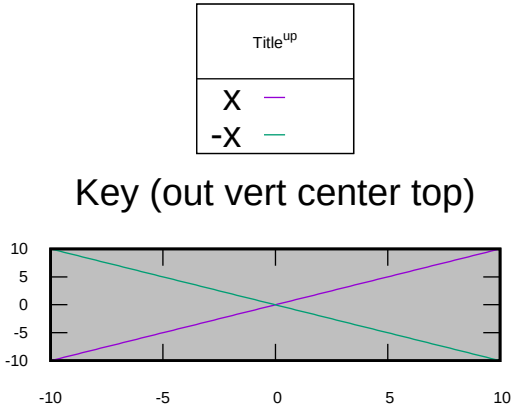
Key (out vert center left)



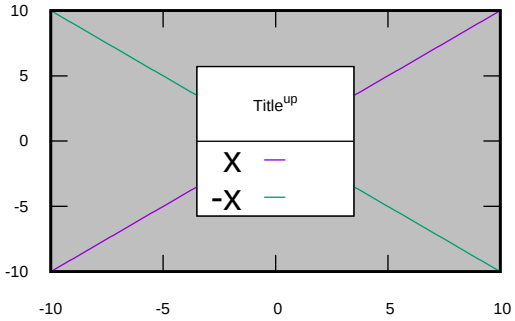
Key (out vert bot left)



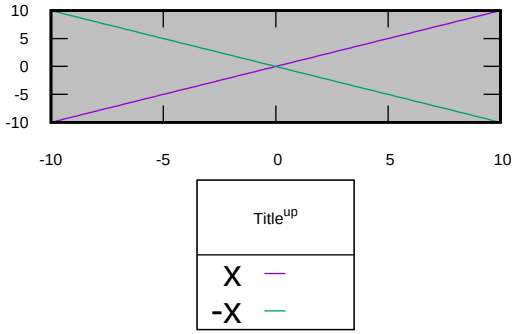
Key (out vert center top)



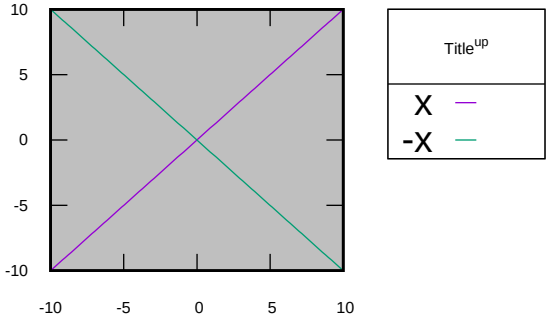
Key (outside vertical center)



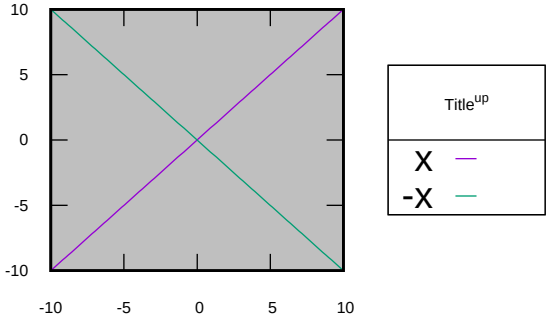
Key (out vert bot center)



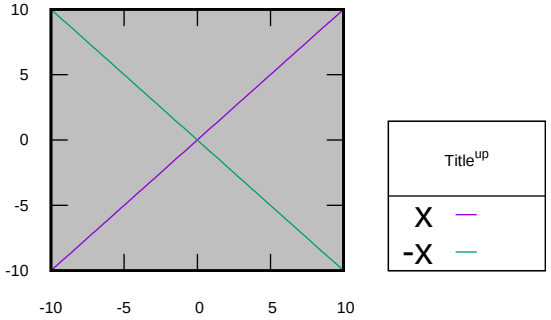
Key (out vert right top)



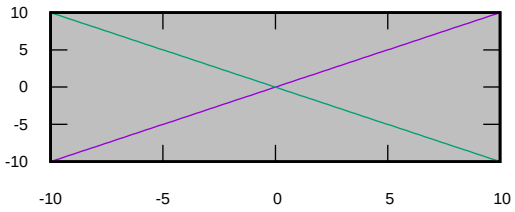
Key (out vert cent right)



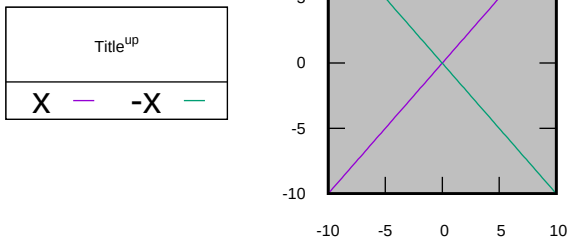
Key (out vert bot right)



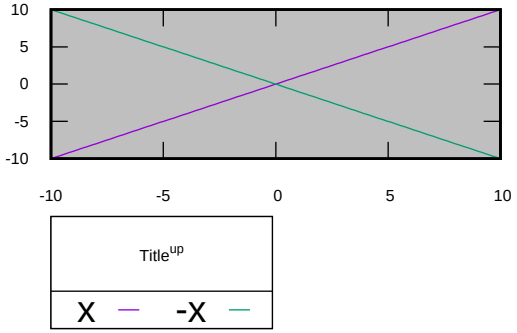
Key (out horiz left top)

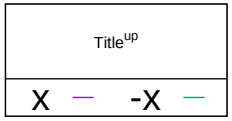


Key (out horiz center left)

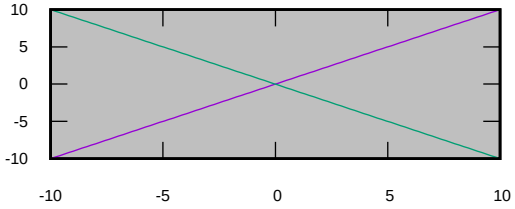


Key (out horiz bot left)

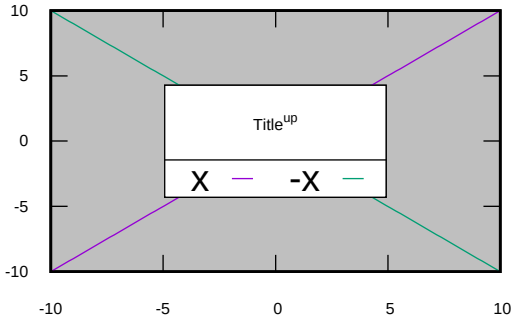




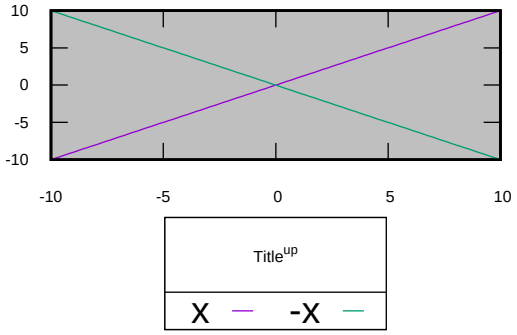
Key (out horiz center top)

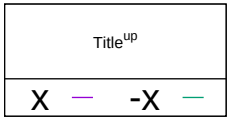


Key (outside horizontal center)

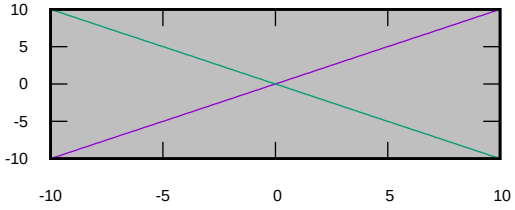


Key (out horiz bot center)

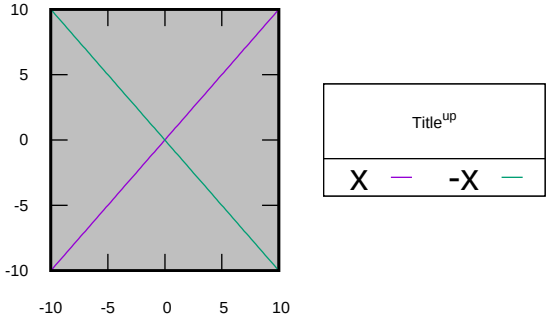




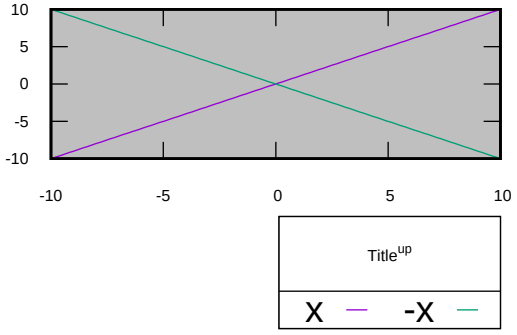
Key (out horiz right top)



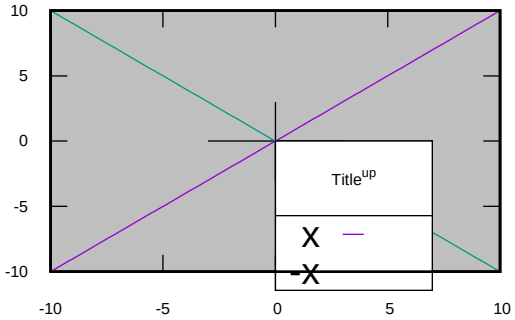
Key (out horiz cent right)



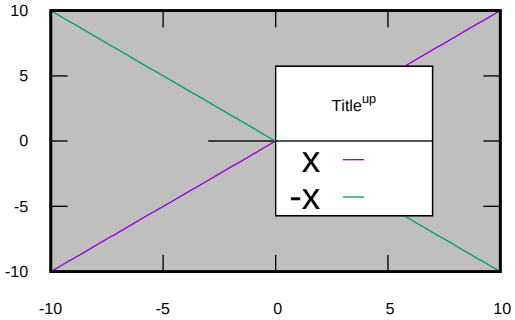
Key (out horiz bot right)



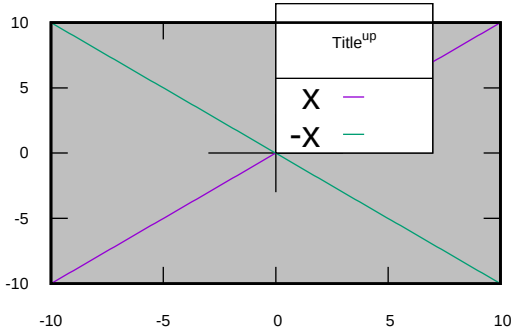
Key (<manual> vert left top)



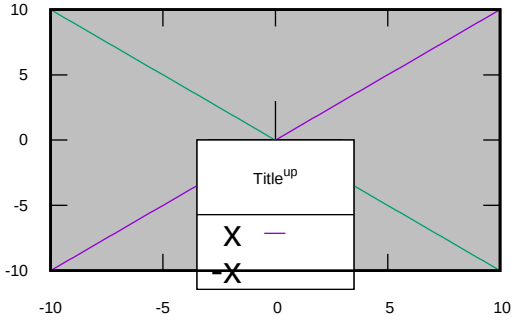
Key (<manual> vert center left)



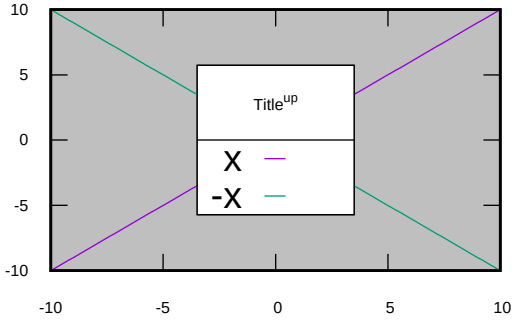
Key (<manual> vert bot left)



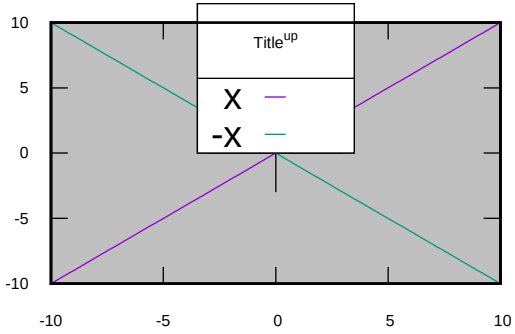
Key (<manual> vert center top)



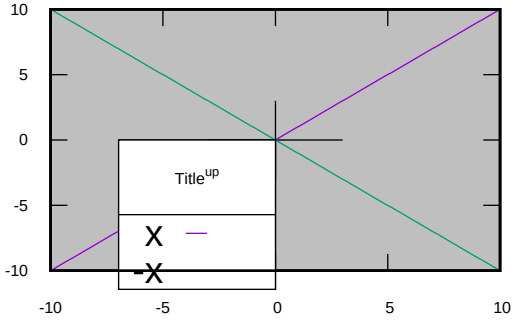
Key (<manual> vertical center)



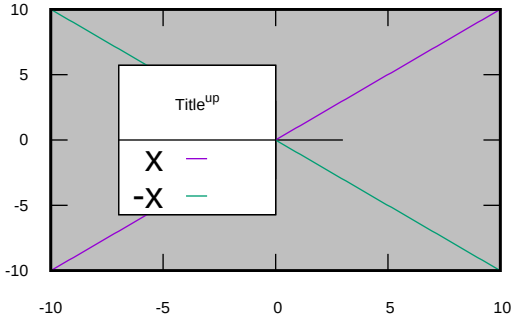
Key (<manual> vert bot center)



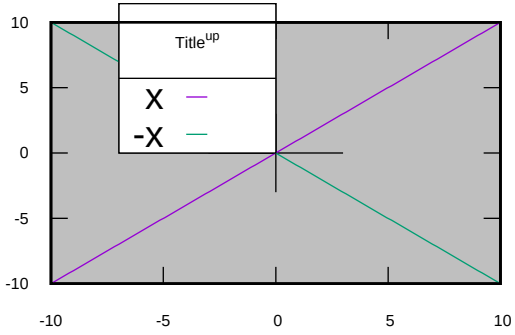
Key (<manual> vert right top)



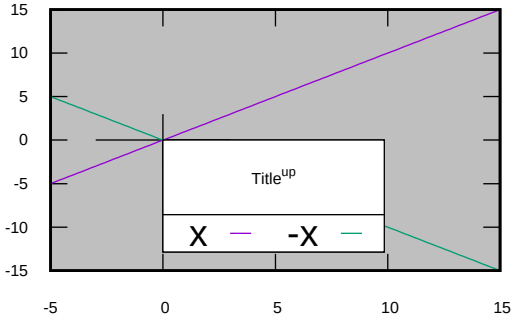
Key (<manual> vert cent right)



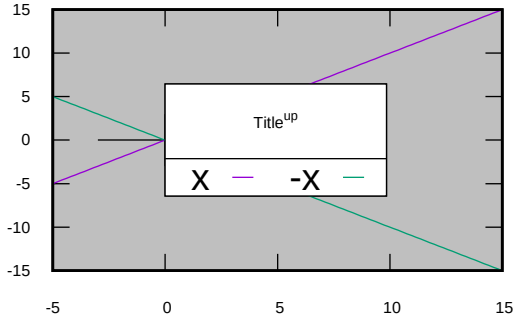
Key (<manual> vert bot right)



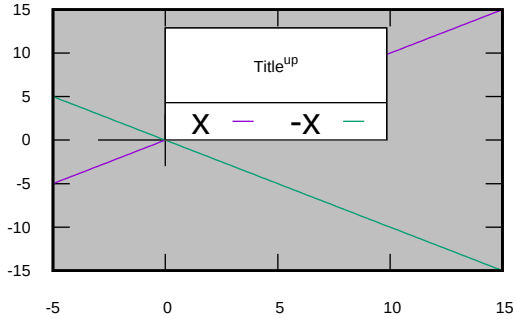
Key (<manual> horiz left top)



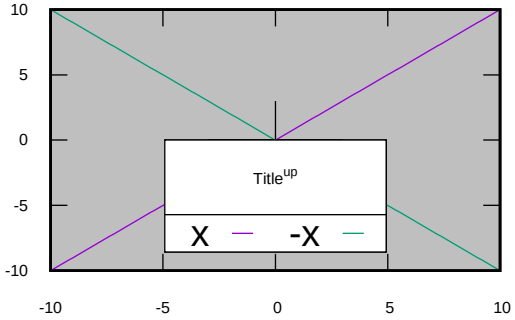
Key (<manual> horiz center left)



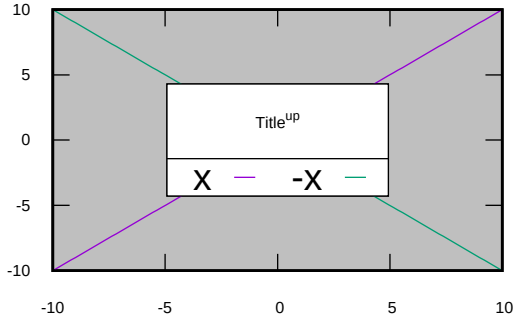
Key (<manual> horiz bot left)



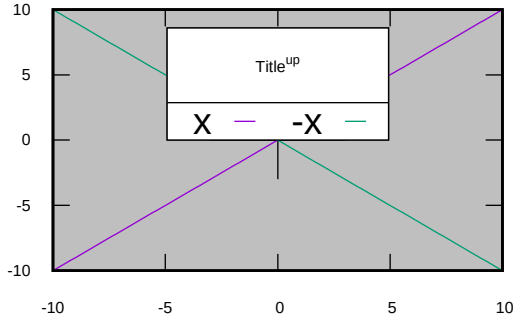
Key (<manual> horiz center top)



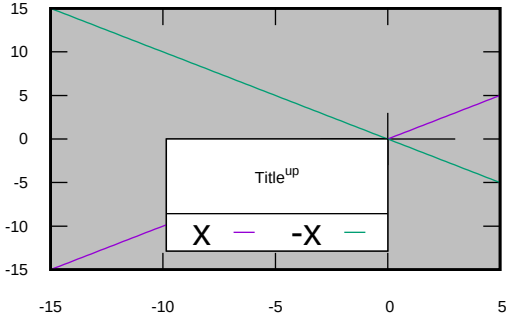
Key (<manual> horizontal center)



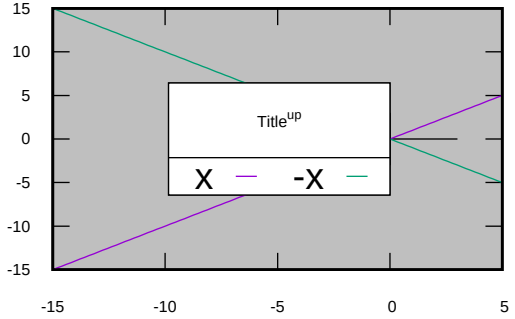
Key (<manual> horiz bot center)



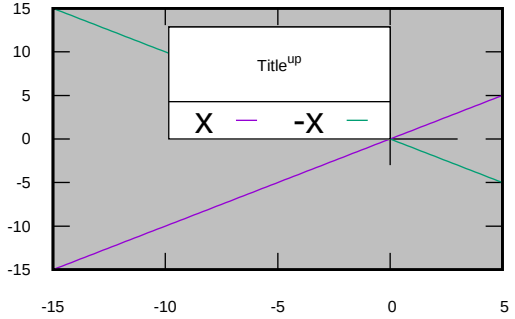
Key (<manual> horiz right top)



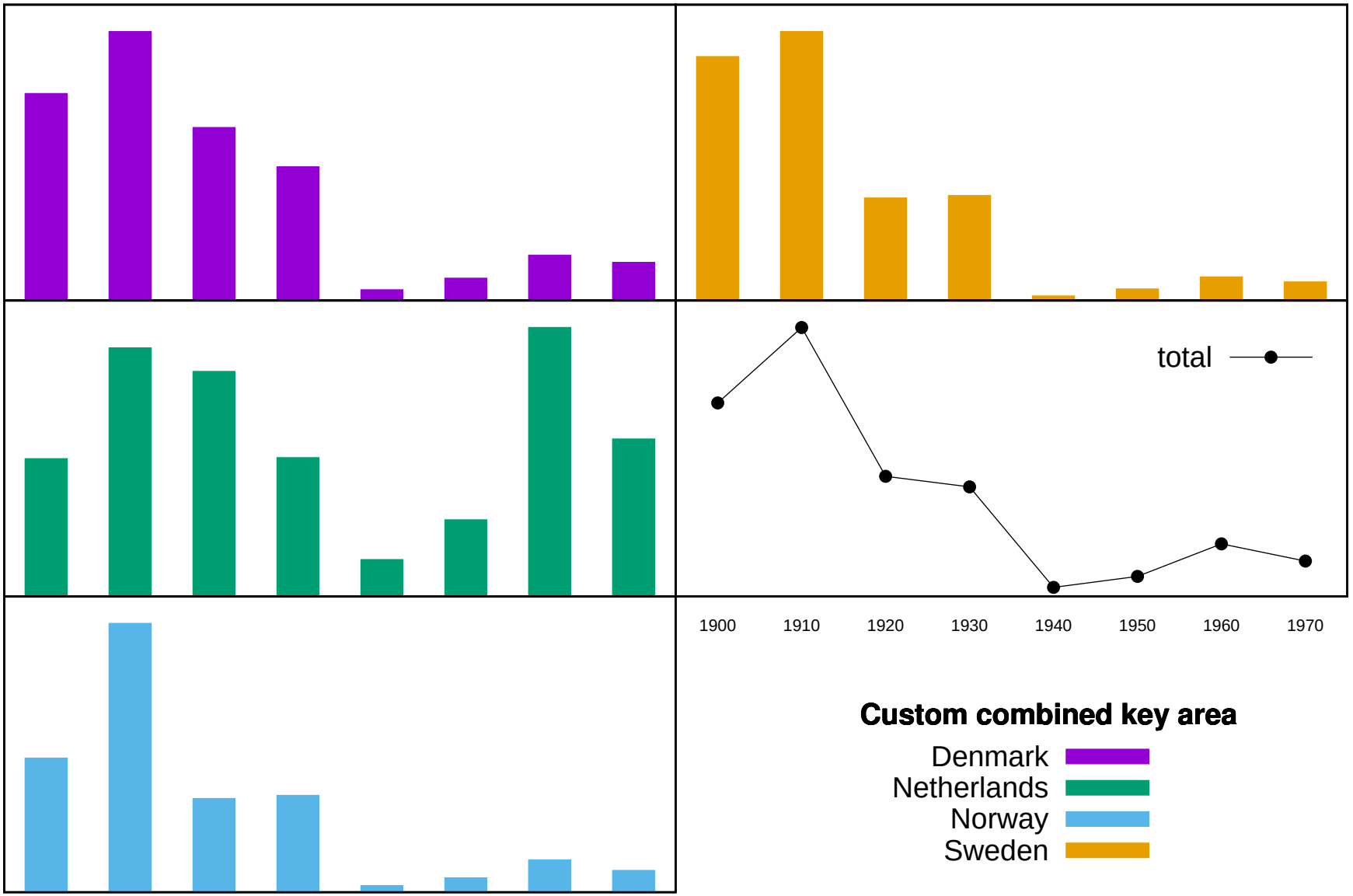
Key (<manual> horiz cent right)

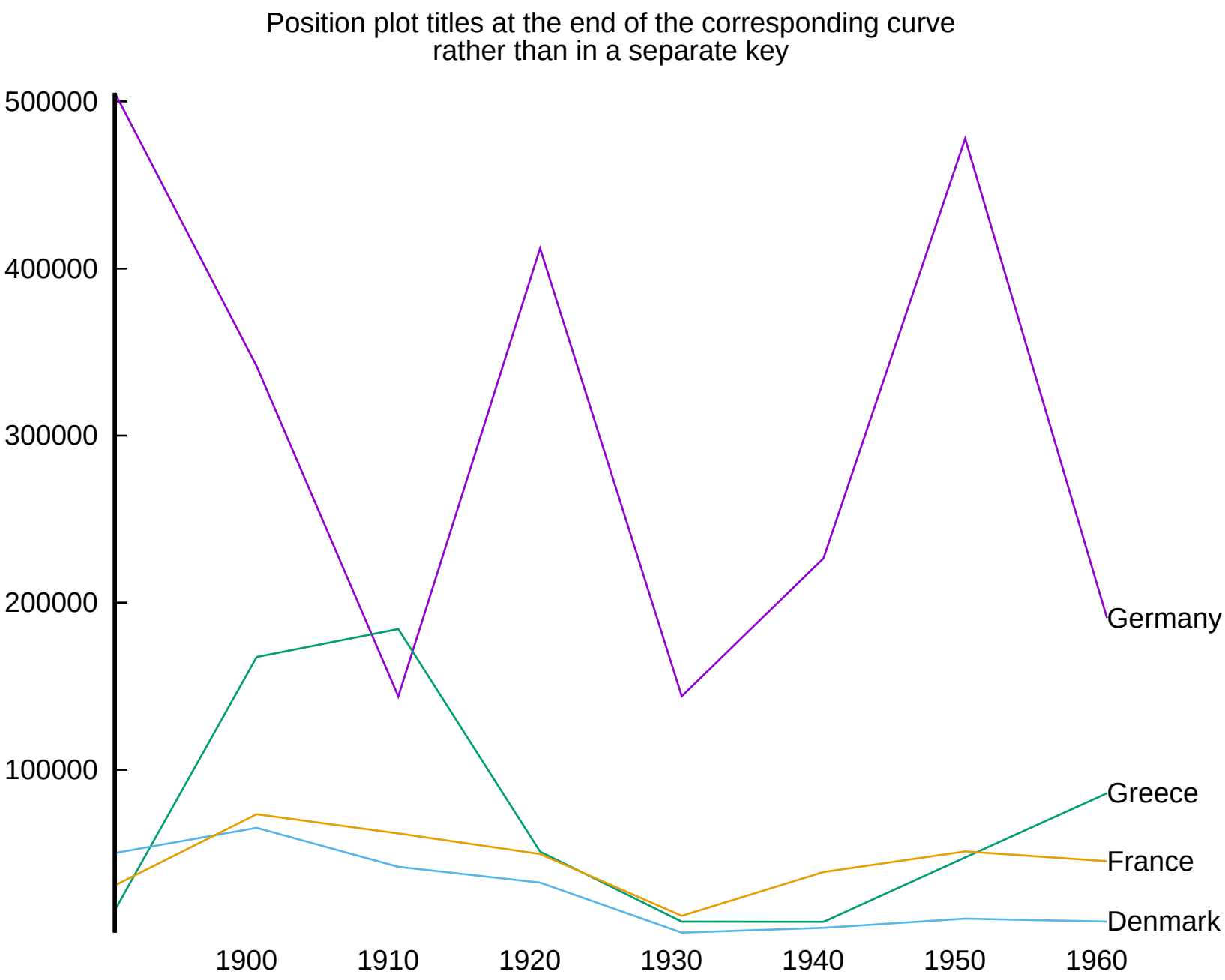


Key (<manual> horiz bot right)

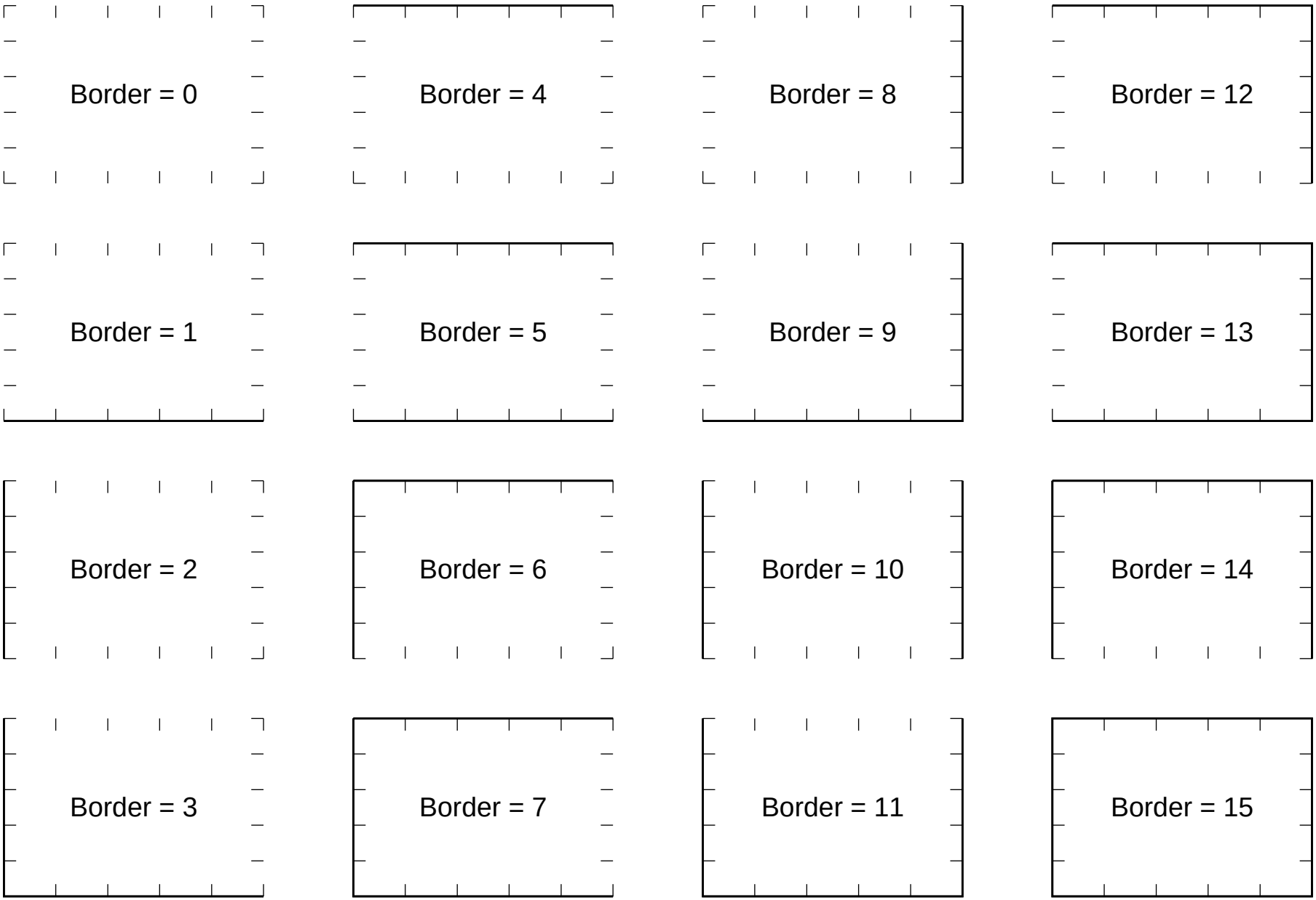


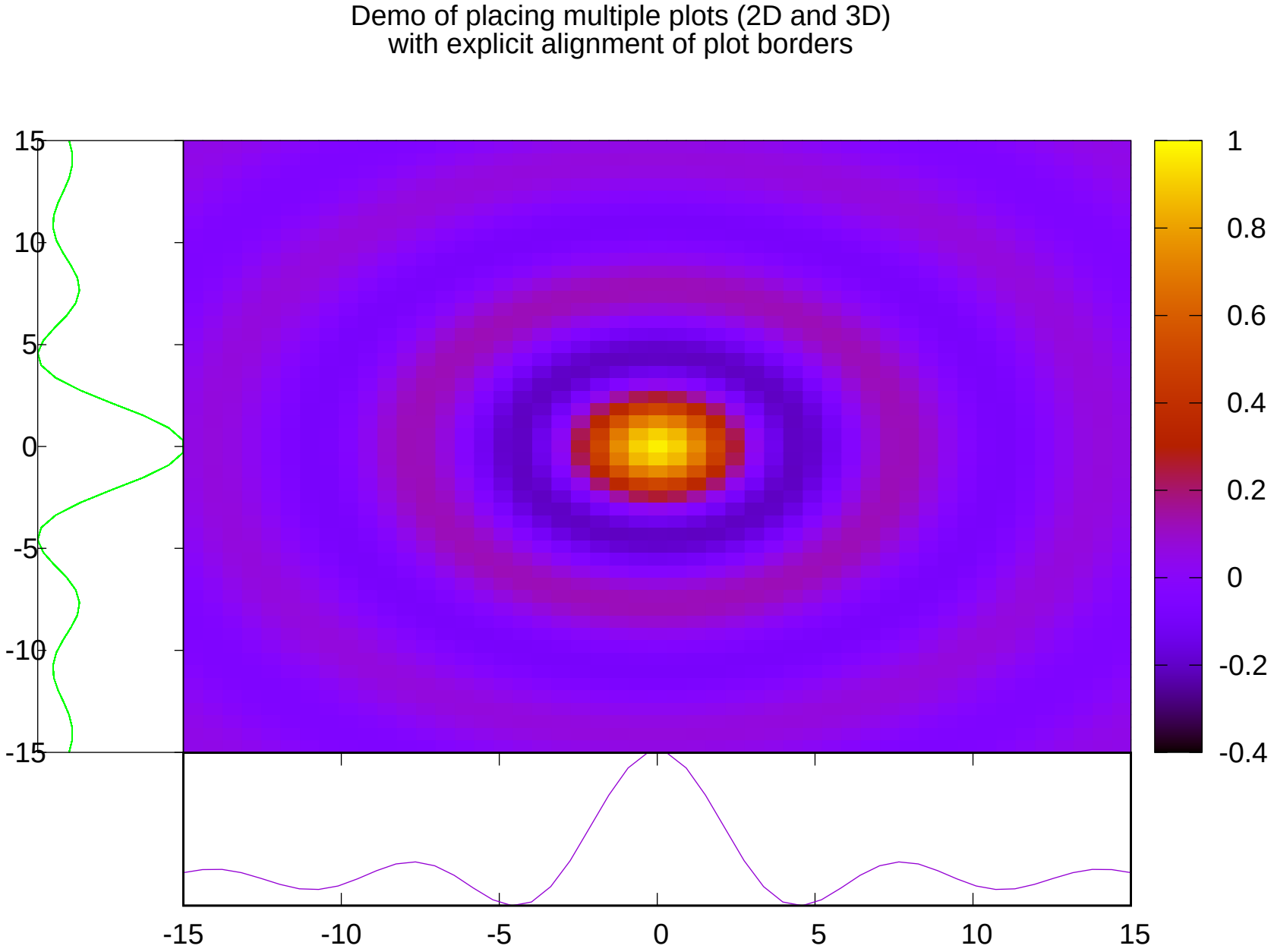
Illustrate use of a custom key area

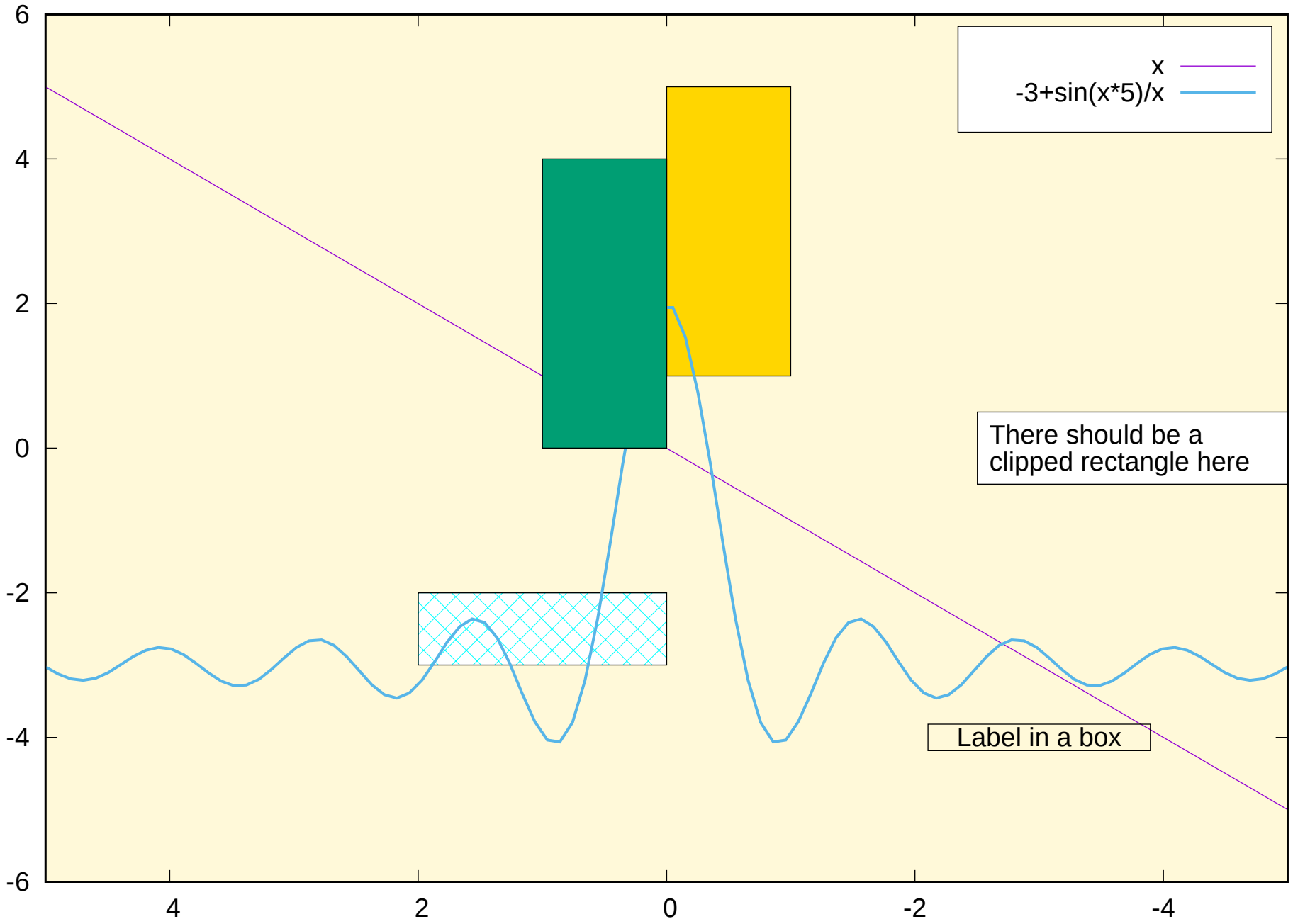


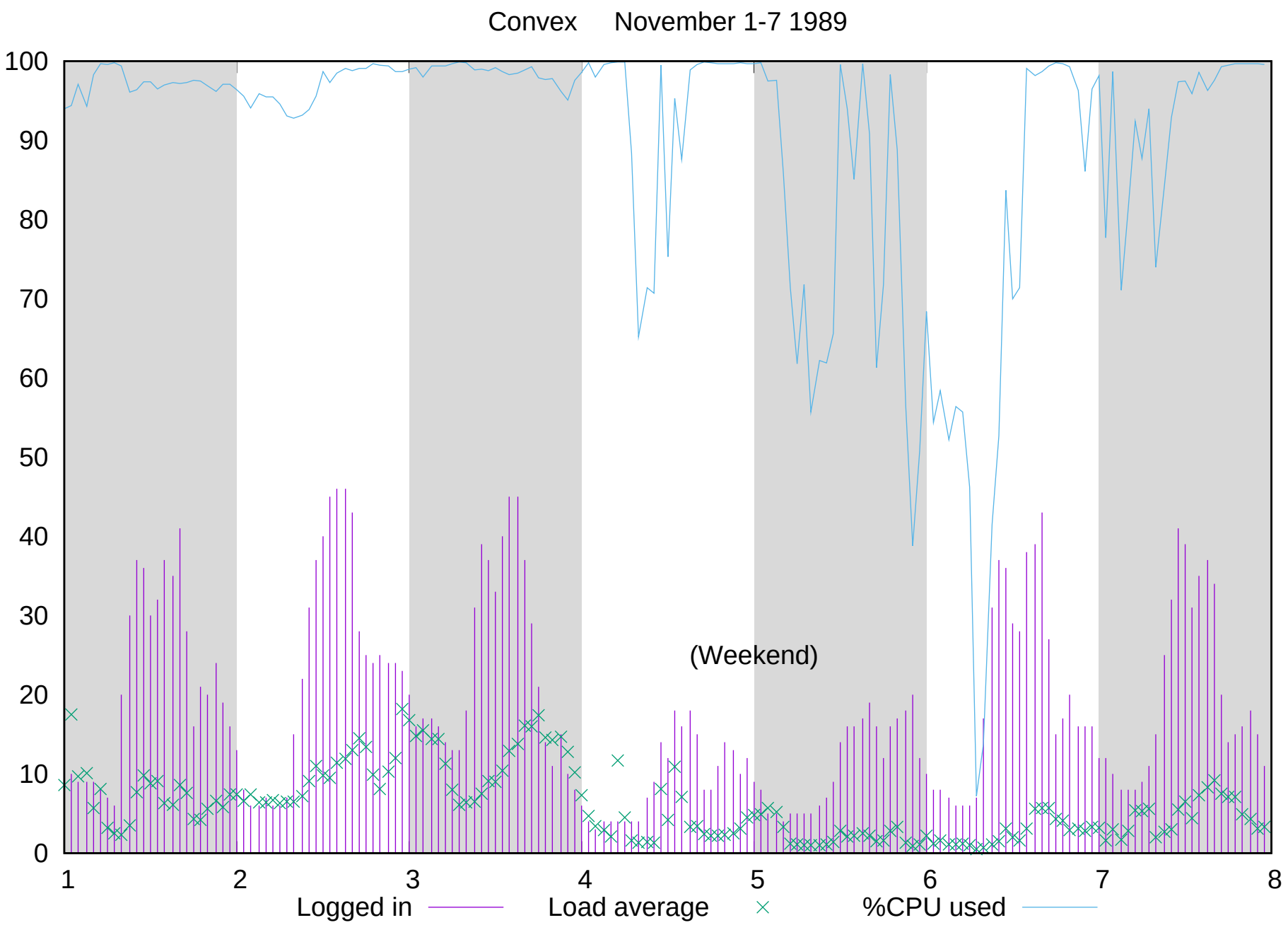

Position plot titles at the end of the corresponding curve
rather than in a separate key
500000
400000
300000
200000
100000
1900
1910
1920
1930
1940
1950
1960
Germany
Greece
France
Denmark

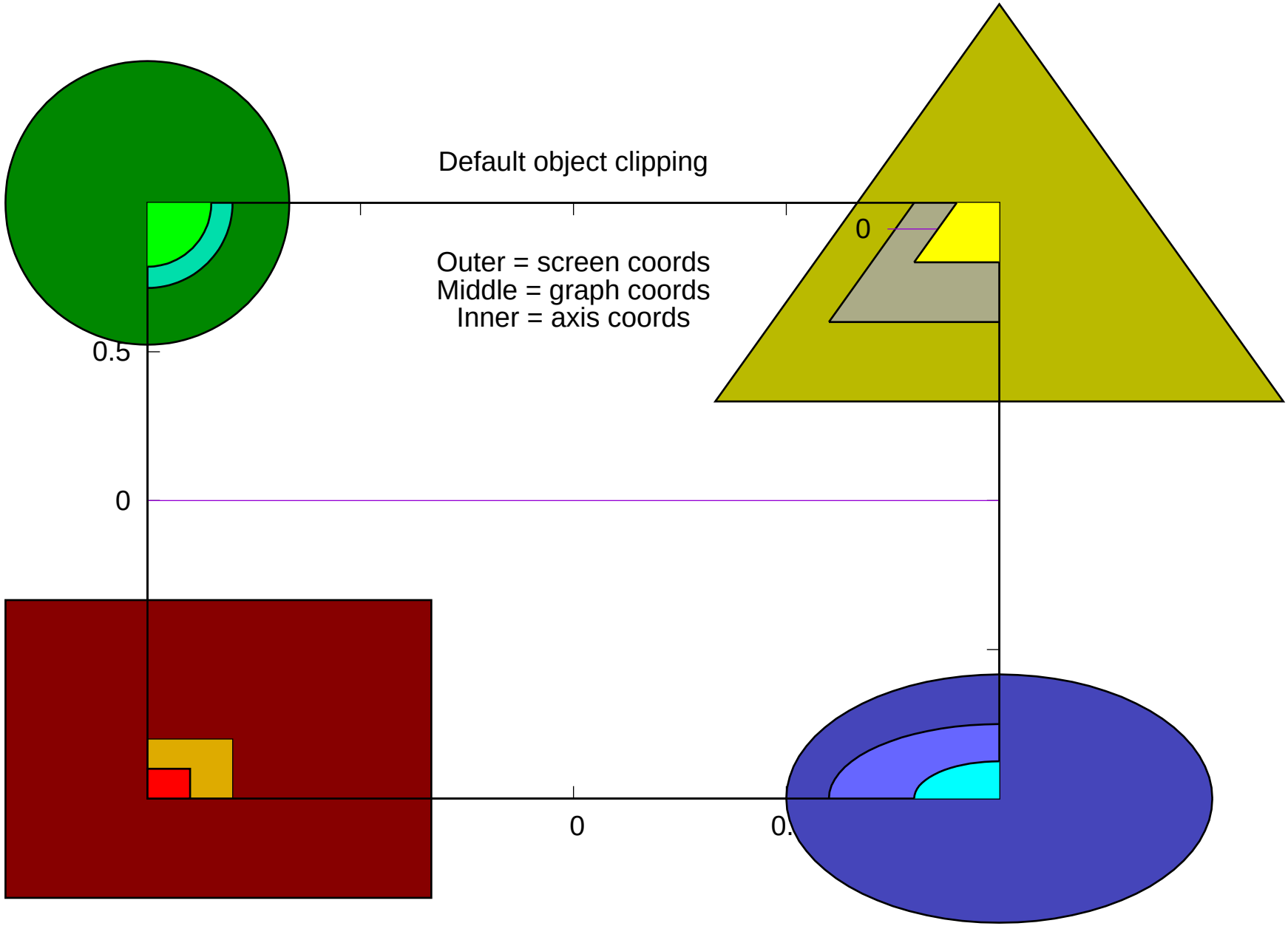
Demonstration of different border settings

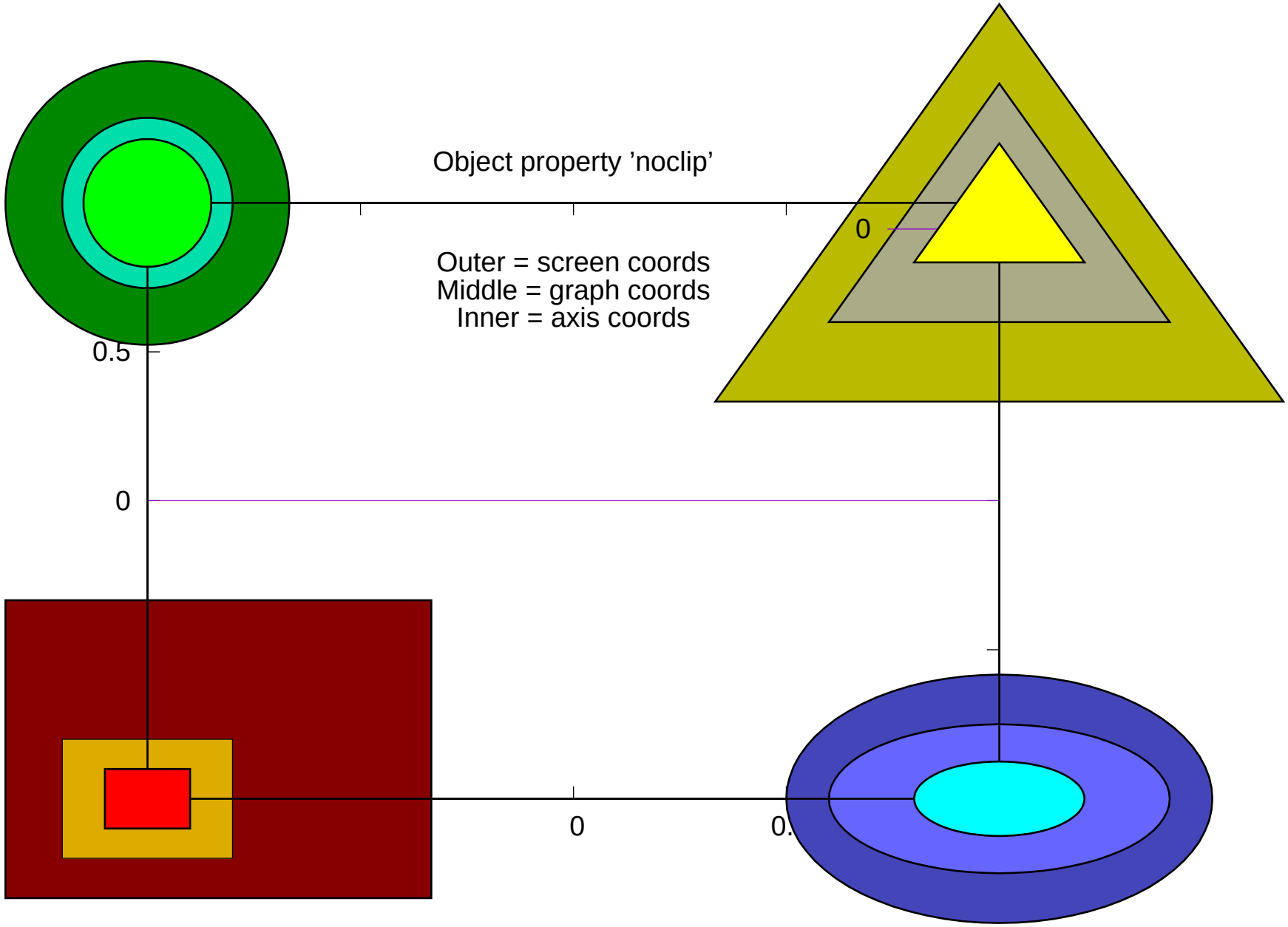


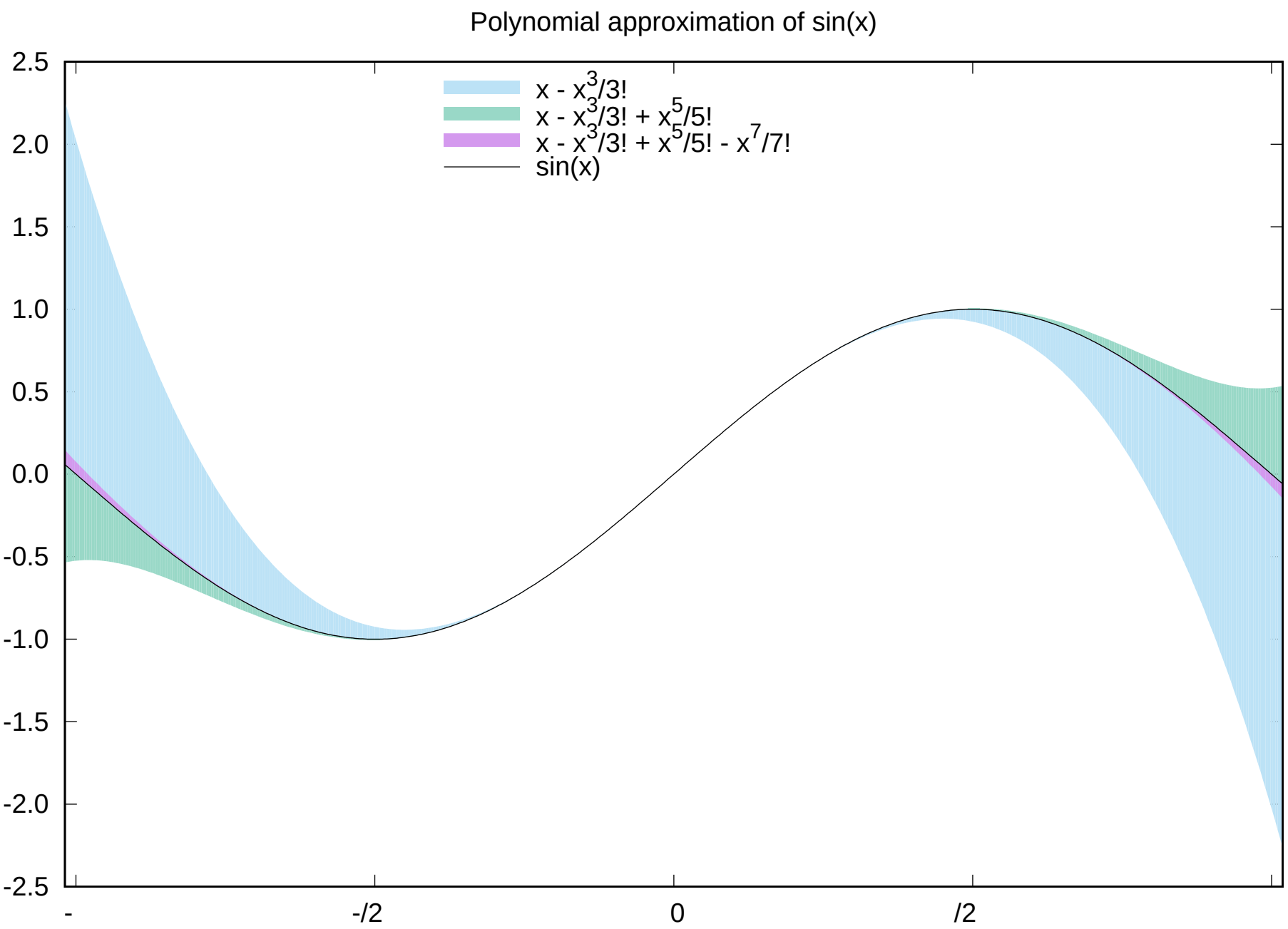

Demo of placing multiple plots (2D and 3D)
with explicit alignment of plot borders
15
10
5
0
-5
-10
-15
-15
-10
-5
0
5
10
15
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

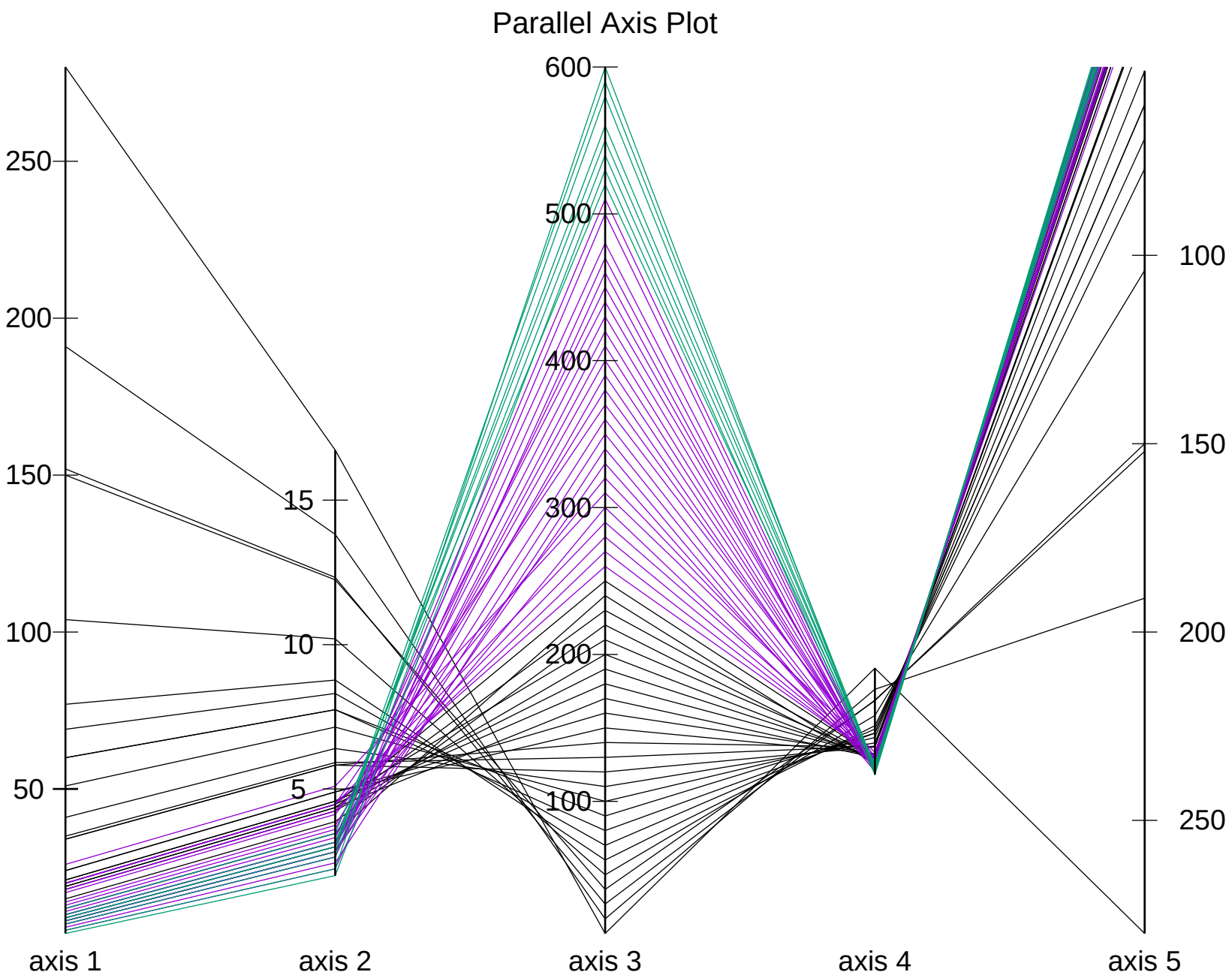

6
4
2
0
-2
-4
-6
4
2
0
-2
-4
x
-3+sin(x*5)/x
There should be a
clipped rectangle here
Label in a box

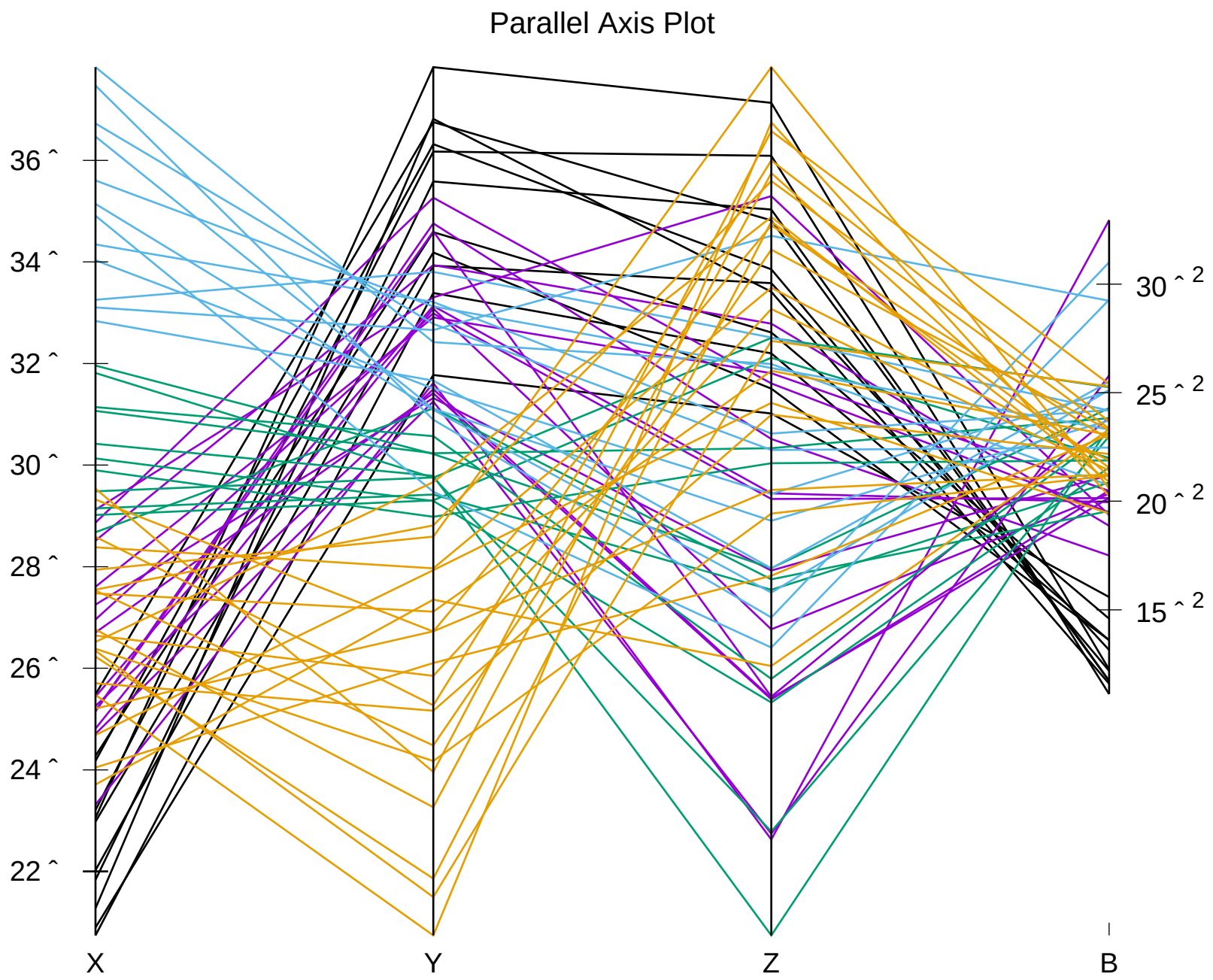

Convex November 1-7 1989
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
(Weekend)
Logged in
Load average
%CPU used

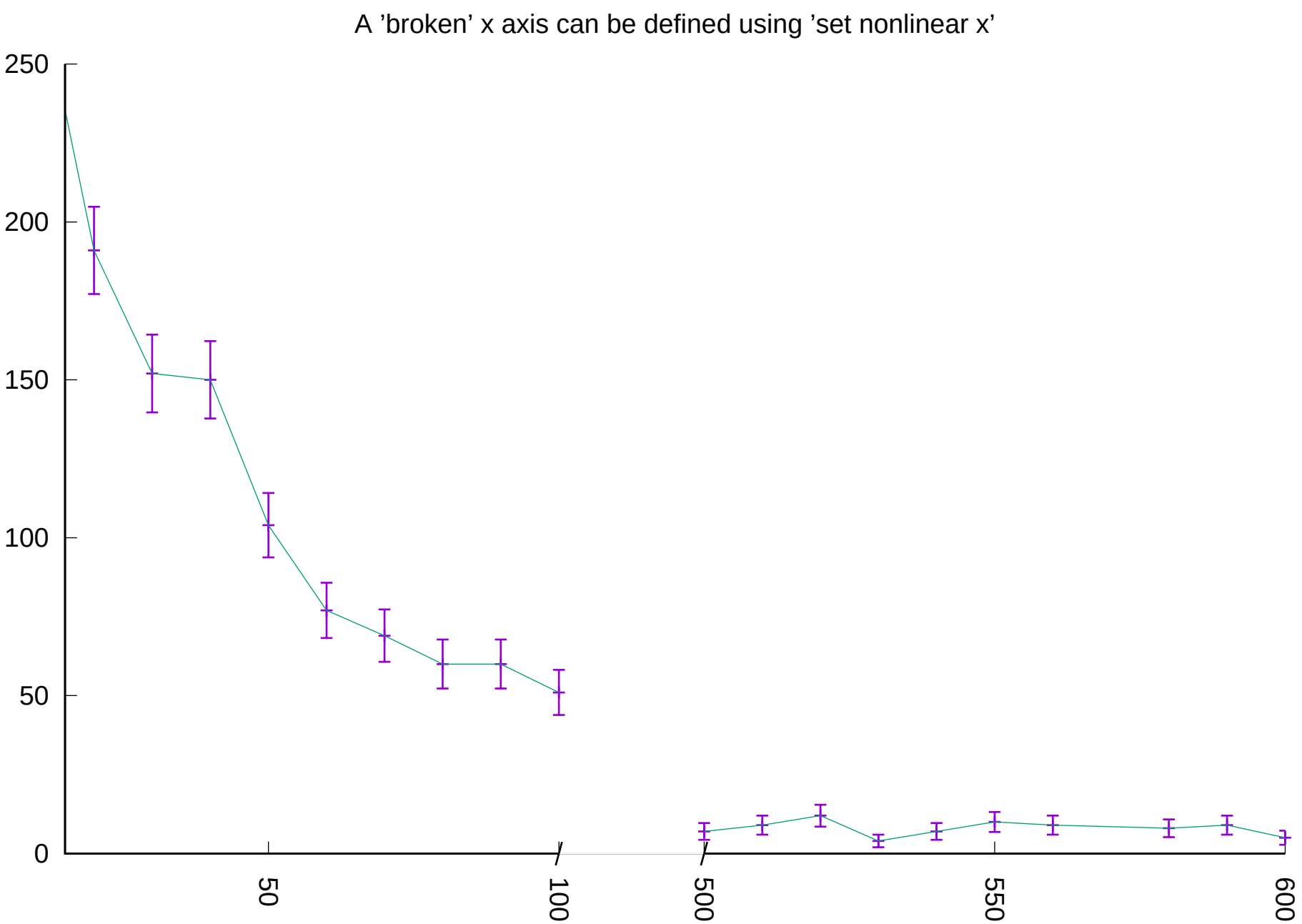

Default object clipping
Outer = screen coords
Middle = graph coords
Inner = axis coords
0.5
0
0
0
0.

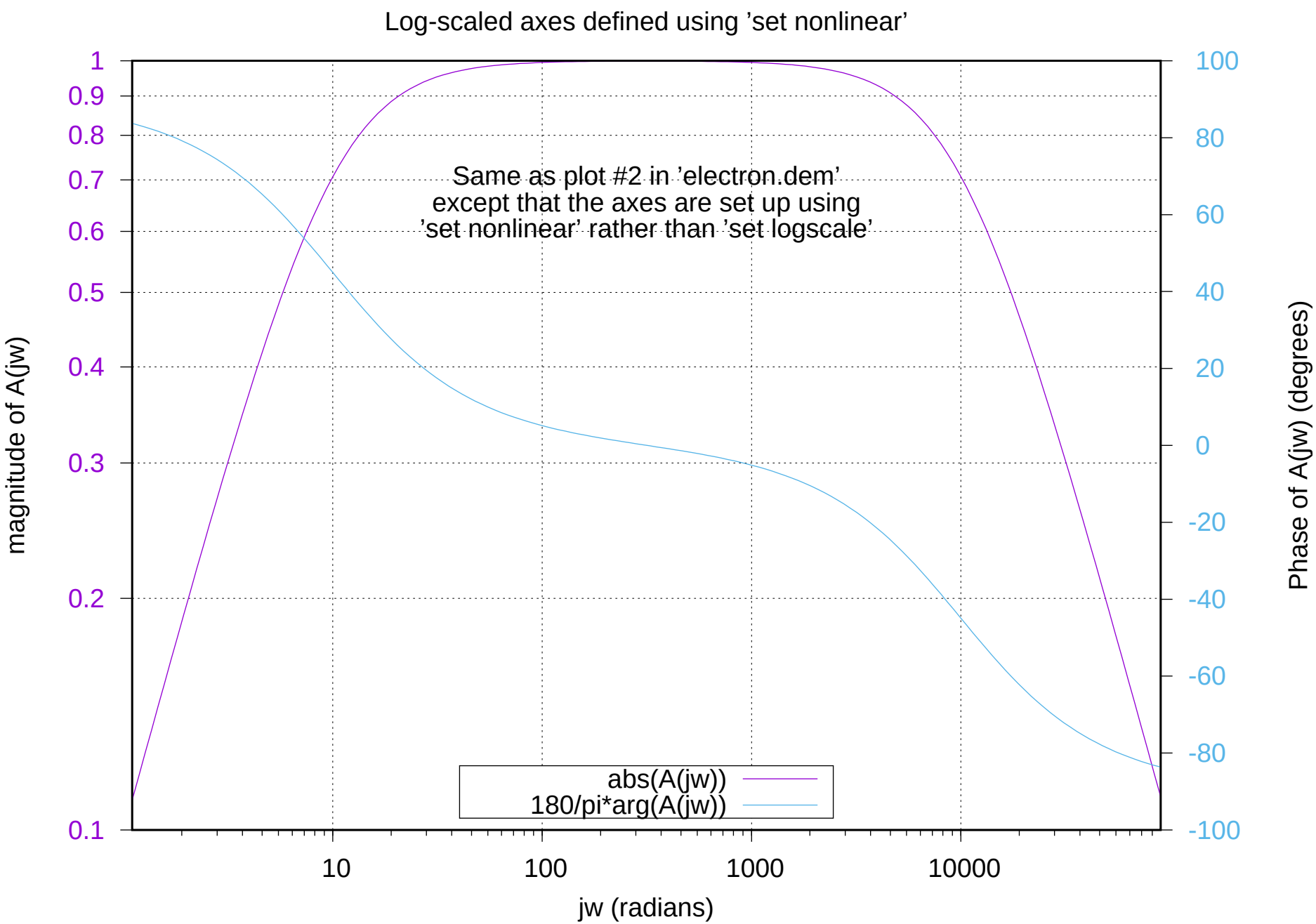

Object property ’noclip’
Outer = screen coords
Middle = graph coords
Inner = axis coords
0.5
0
0
0
0.


Polynomial approximation of sin(x)
x - x^3/3!
x - x^3/3! + x^5/5!
x - x^3/3! + x^5/5! - x^7/7!
sin(x)
2.5
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
-2.0
-2.5
-
-/2
0
/2


Parallel Axis Plot
250
200
150
100
50
15
10
5
600
500
400
300
200
100
100
150
200
250
axis 1
axis 2
axis 3
axis 4
axis 5

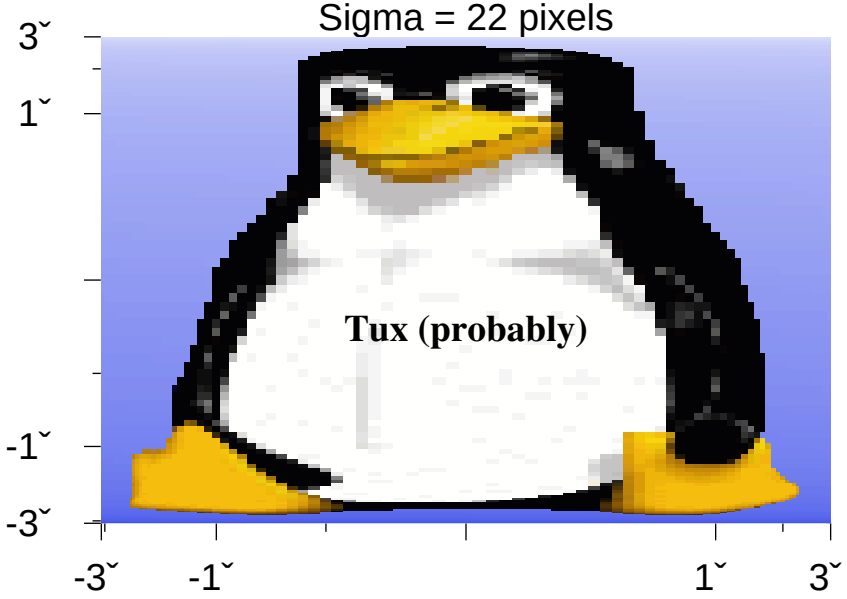

Parallel Axis Plot
36 ^
34 ^
32 ^
30 ^
28 ^
26 ^
24 ^
22 ^
30 ^ 2
25 ^ 2
20 ^ 2
15 ^ 2
X
Y
Z
B


A ’broken’ x axis can be defined using ’set nonlinear x’
250
200
150
100
50
0
50
100
500
550
600

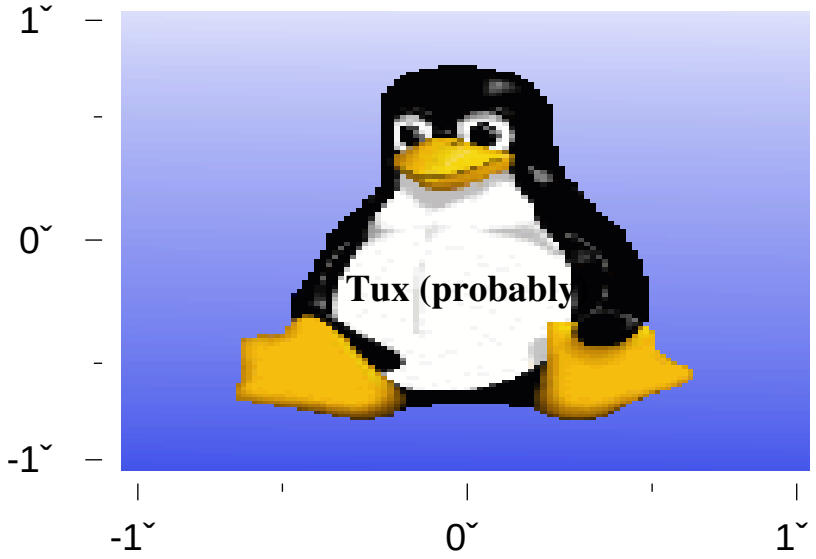

Log-scaled axes defined using ’set nonlinear’
Same as plot #2 in ’electron.dem’
except that the axes are set up using
’set nonlinear’ rather than ’set logscale’
magnitude of A(jw)
Phase of A(jw) (degrees)
jw (radians)
abs(A(jw))
180/pi*arg(A(jw))
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
100
80
60
40
20
0
-20
-40
-60
-80
-100
10
100
1000
10000

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

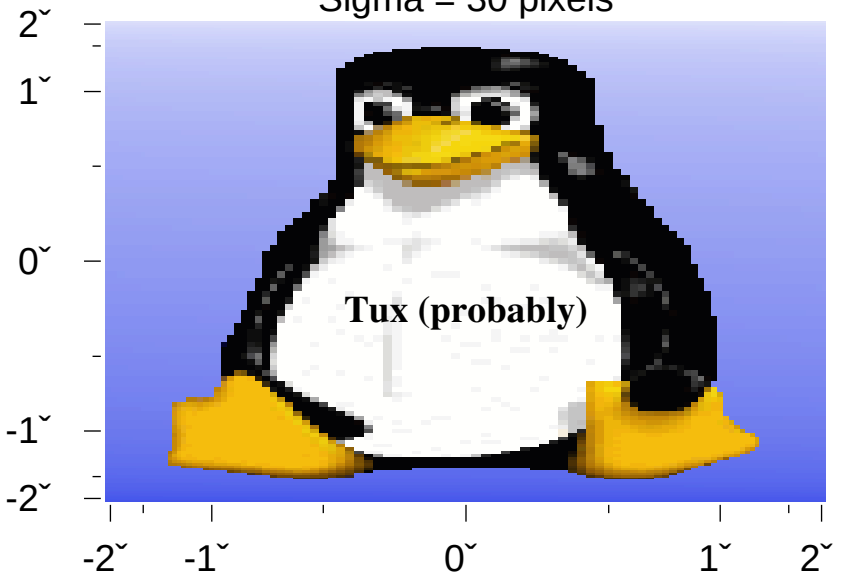
Sigma = 22 pixels



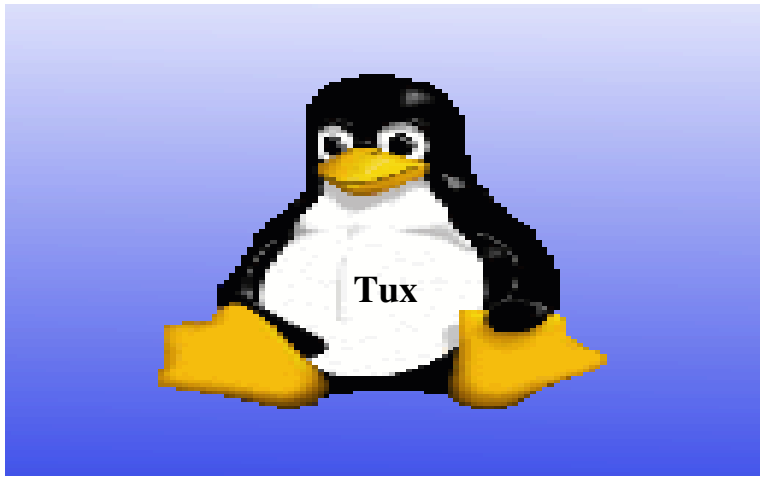
Sigma = 60 pixels

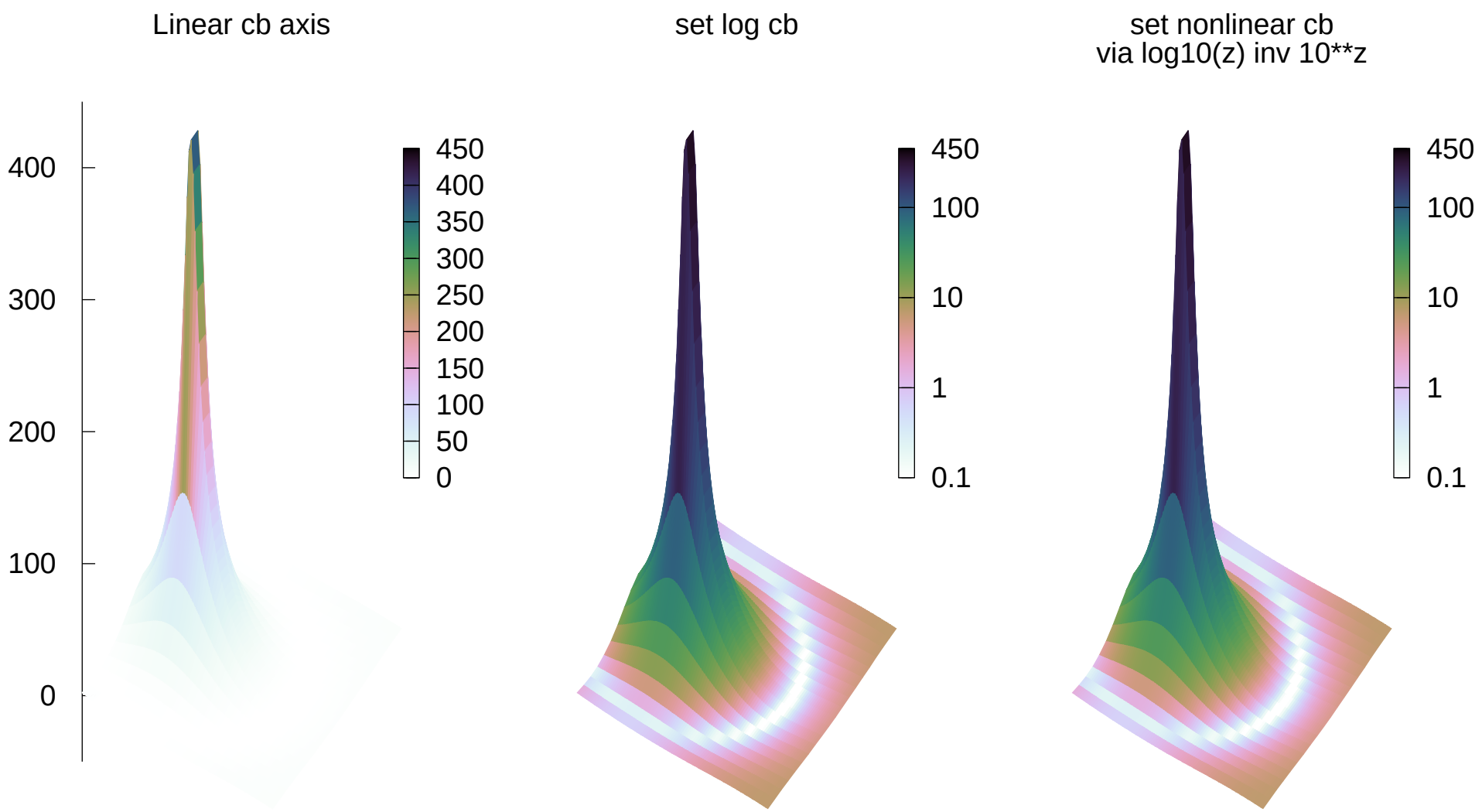


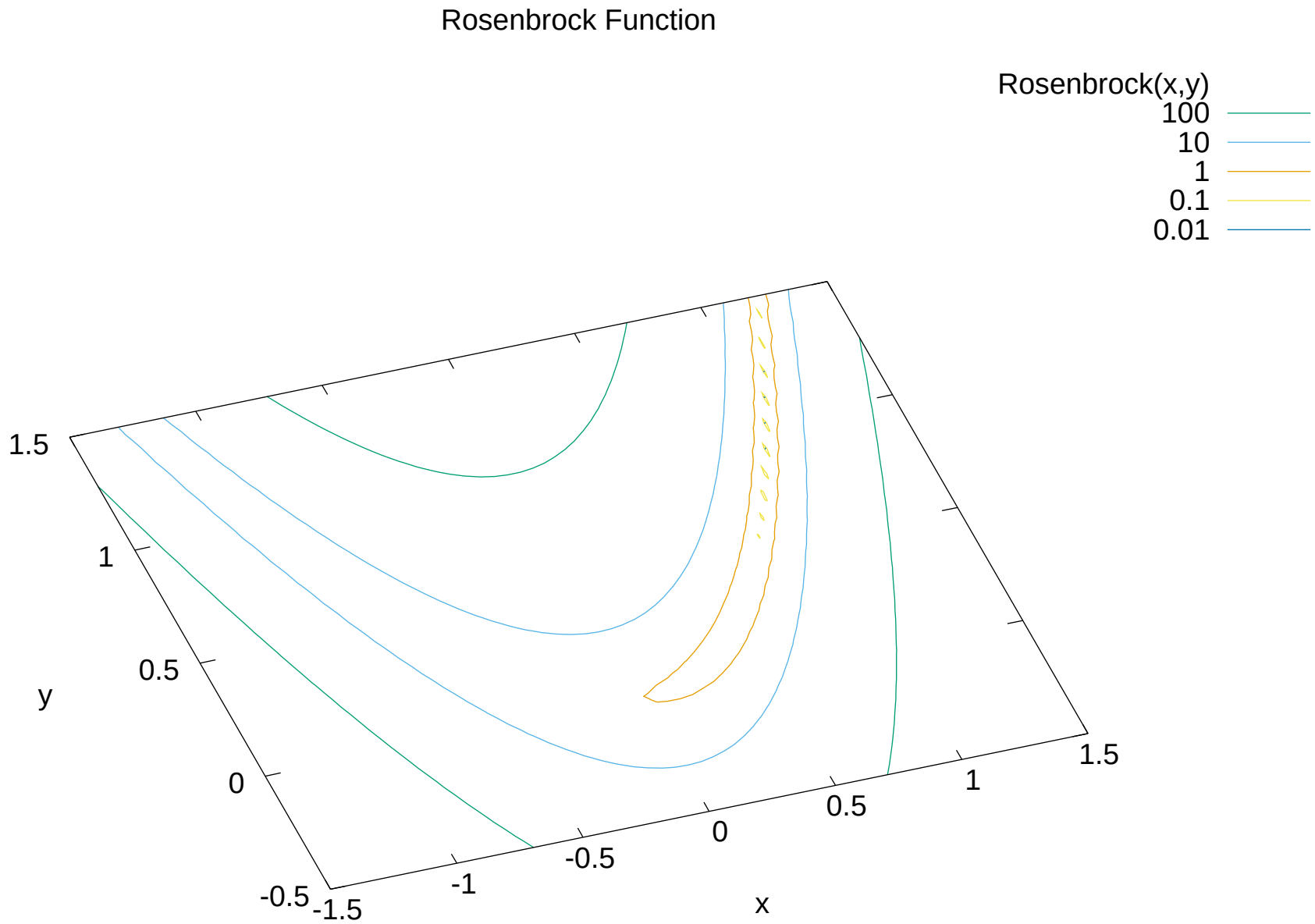
Sigma = 30 pixels

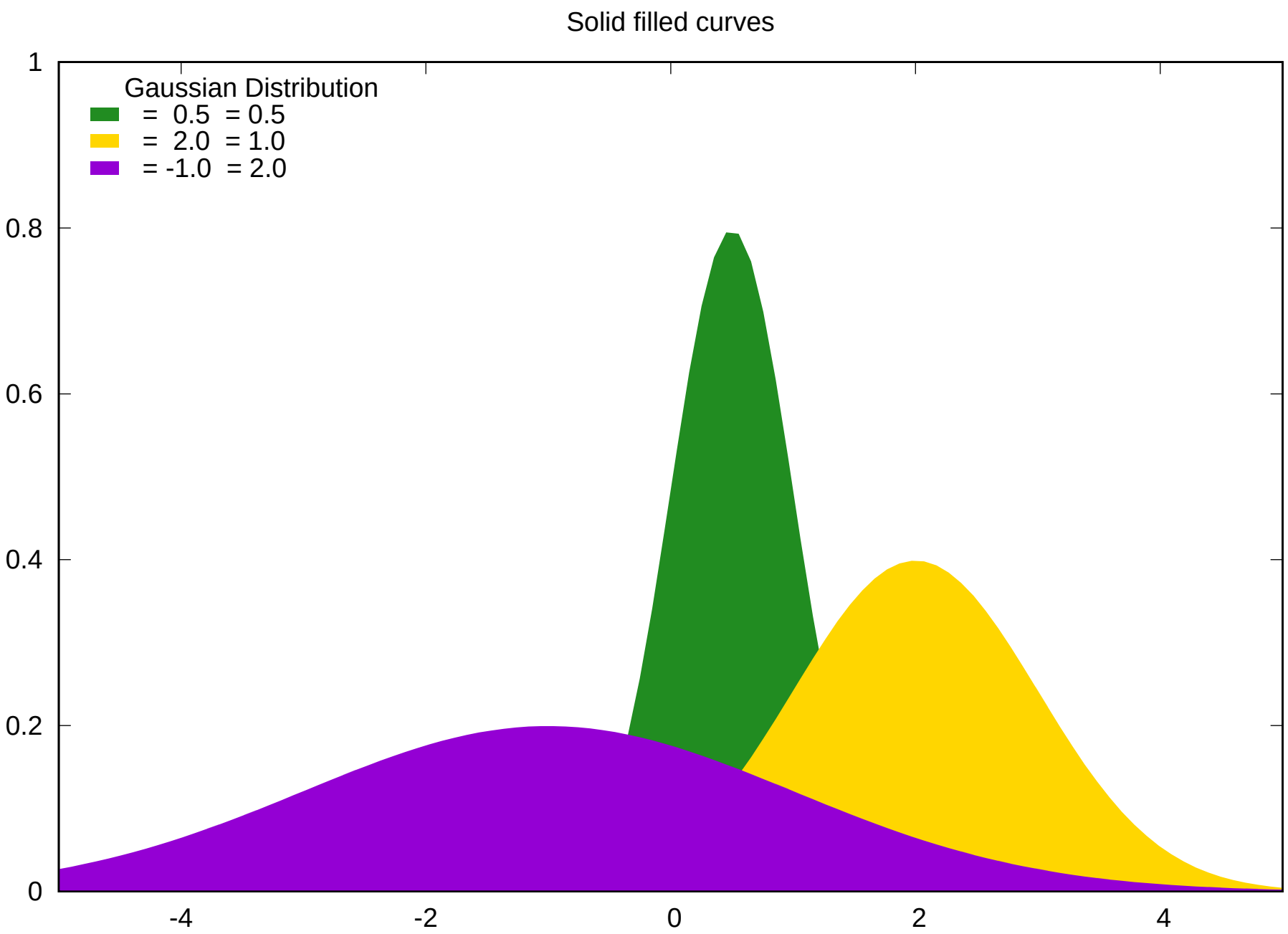


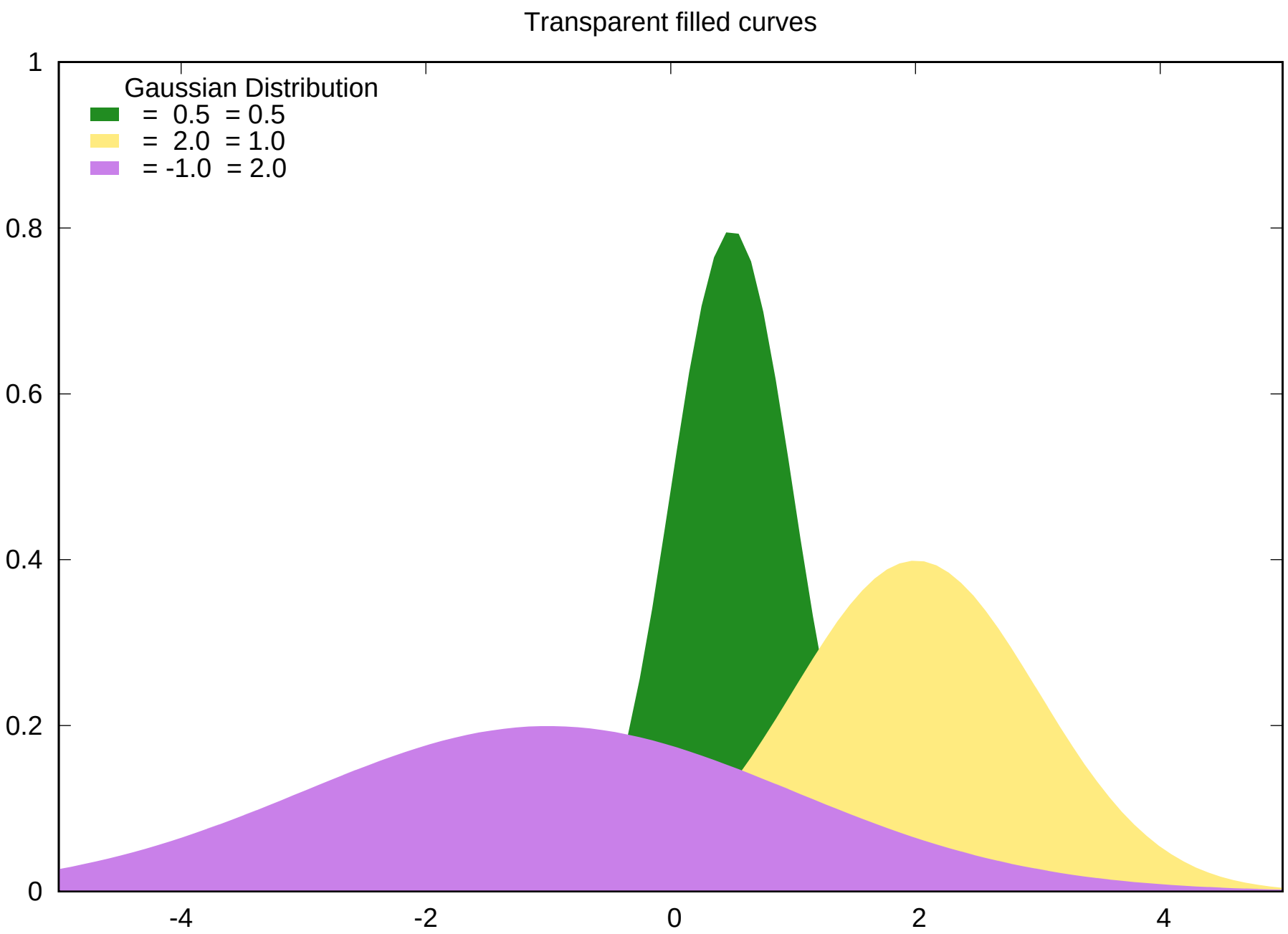
Linear Scale

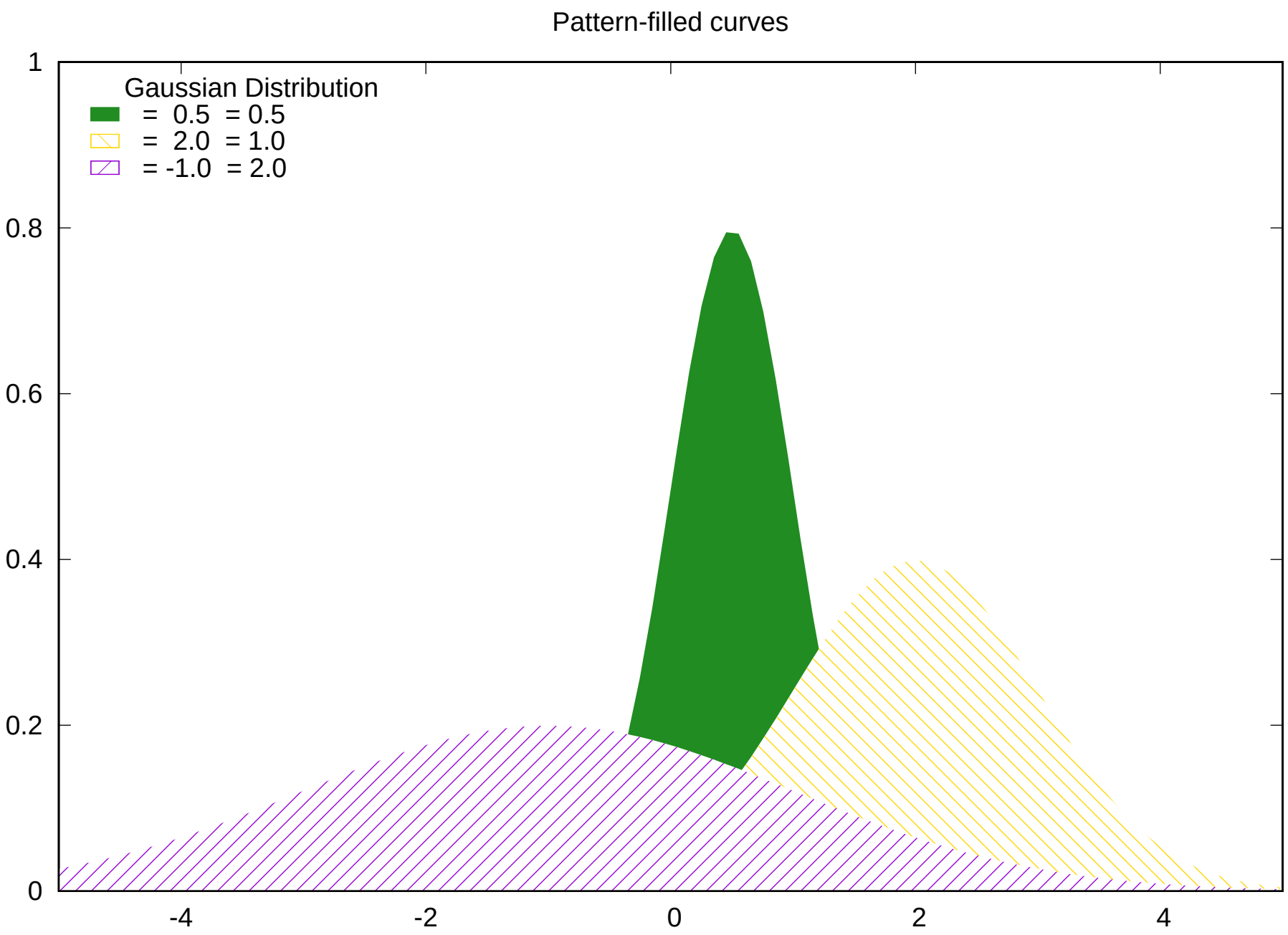


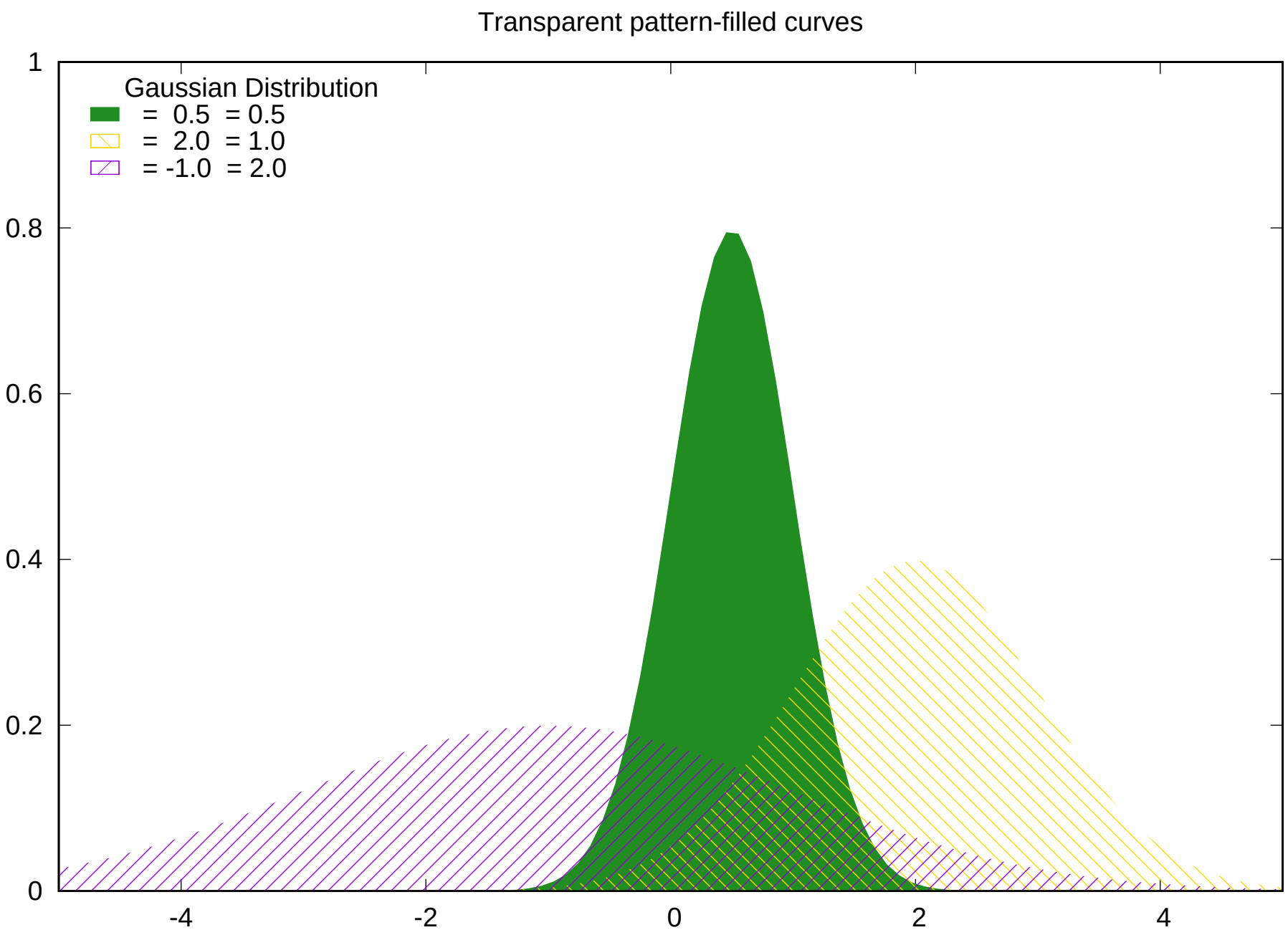

Linear cb axis
400
300
200
100
0
450
400
350
300
250
200
150
100
50
0
set log cb
450
100
10
1
0.1
set nonlinear cb
via log10(z) inv 10**z
450
100
10
1
0.1

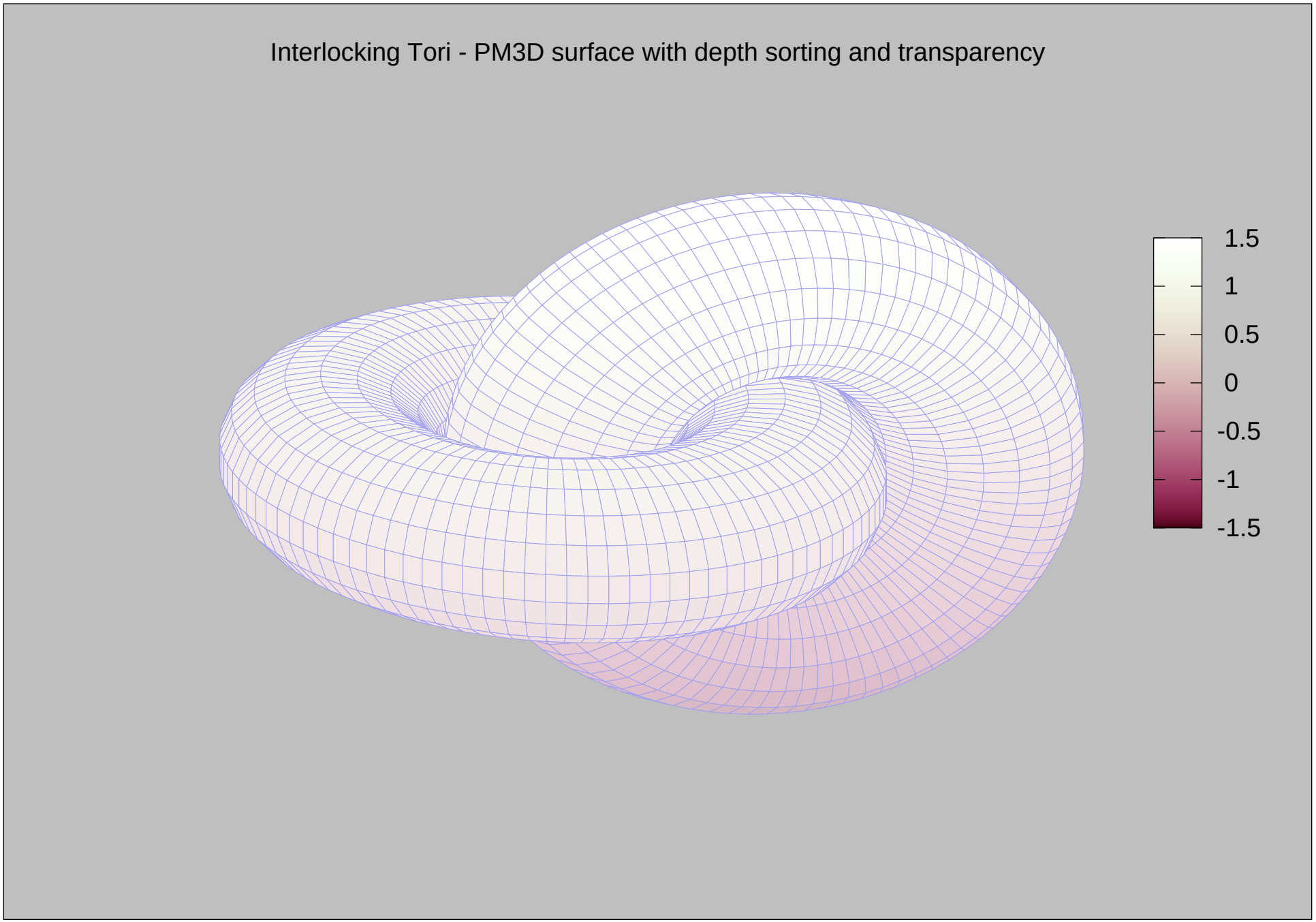

Rosenbrock Function
Rosenbrock(x,y)
100
10
1
0.1
0.01
1.5
1
0.5
0
-0.5
y
-1.5
-1
-0.5
0
0.5
1
1.5
x

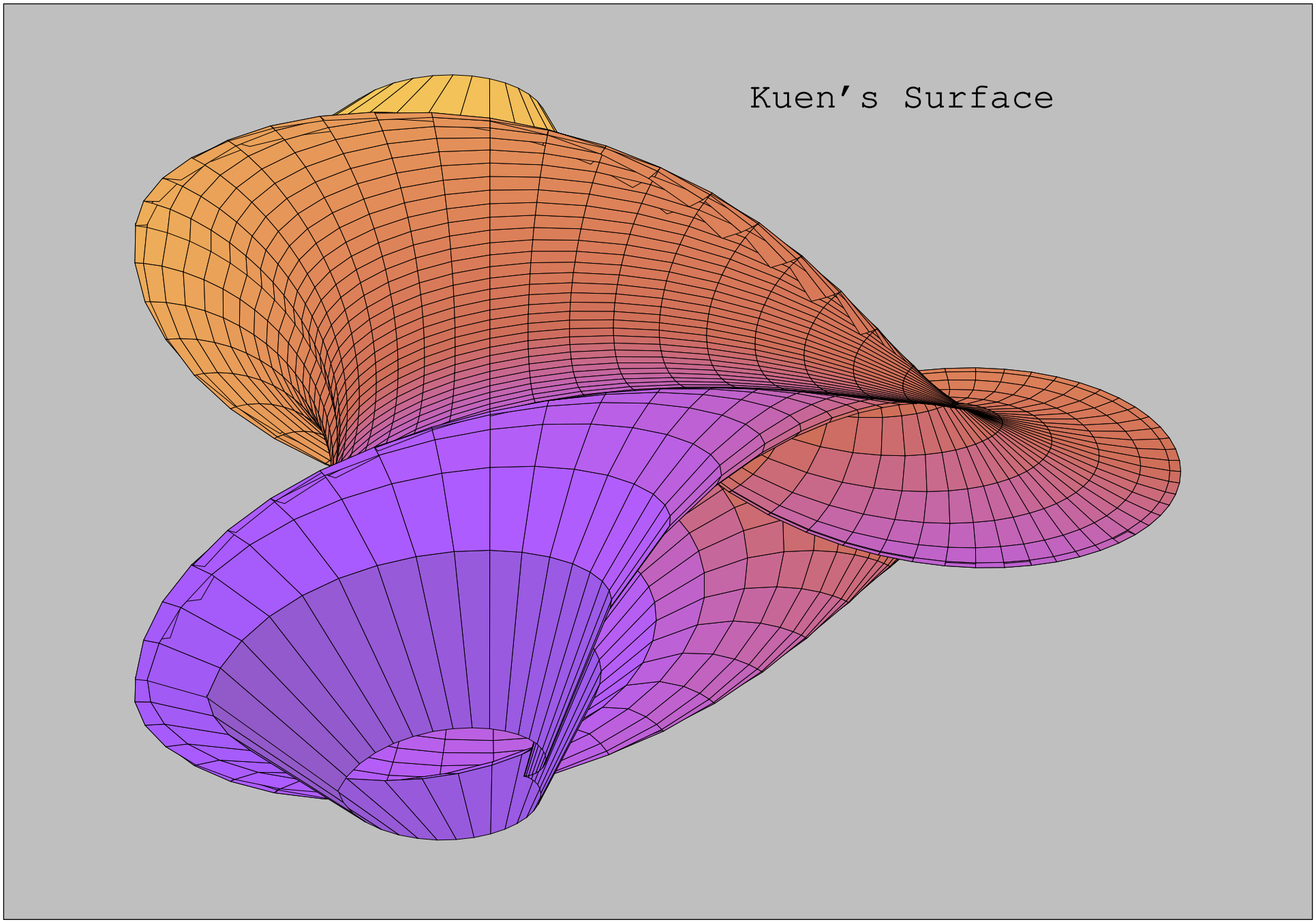

Solid filled curves
Gaussian Distribution
= 0.5 = 0.5
= 2.0 = 1.0
= -1.0 = 2.0
1
0.8
0.6
0.4
0.2
0
-4
-2
0
2
4


Transparent filled curves
Gaussian Distribution
= 0.5 = 0.5
= 2.0 = 1.0
= -1.0 = 2.0
1
0.8
0.6
0.4
0.2
0
-4
-2
0
2
4

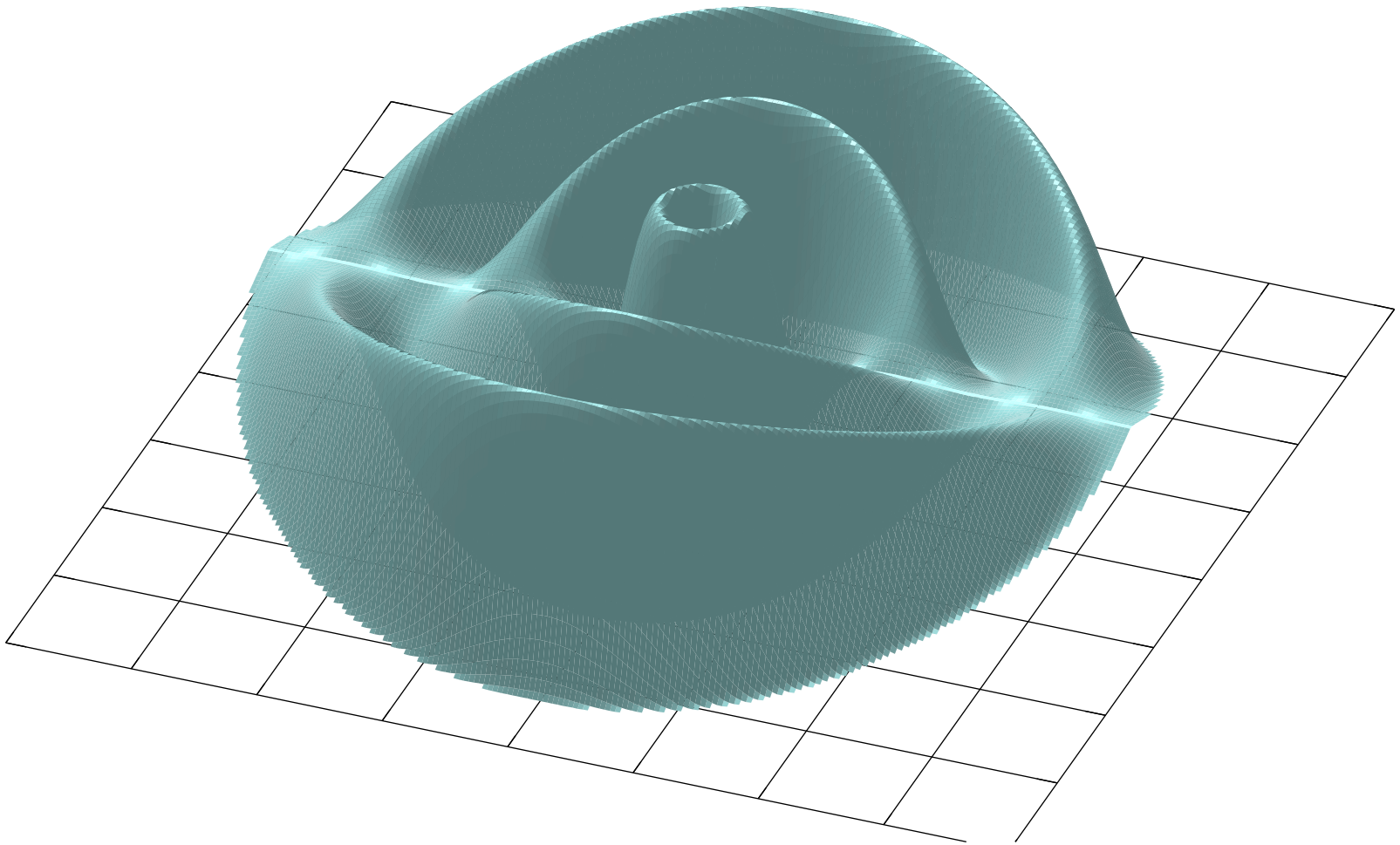

Pattern-filled curves
Gaussian Distribution
= 0.5 = 0.5
= 2.0 = 1.0
= -1.0 = 2.0
1
0.8
0.6
0.4
0.2
0
-4
-2
0
2
4

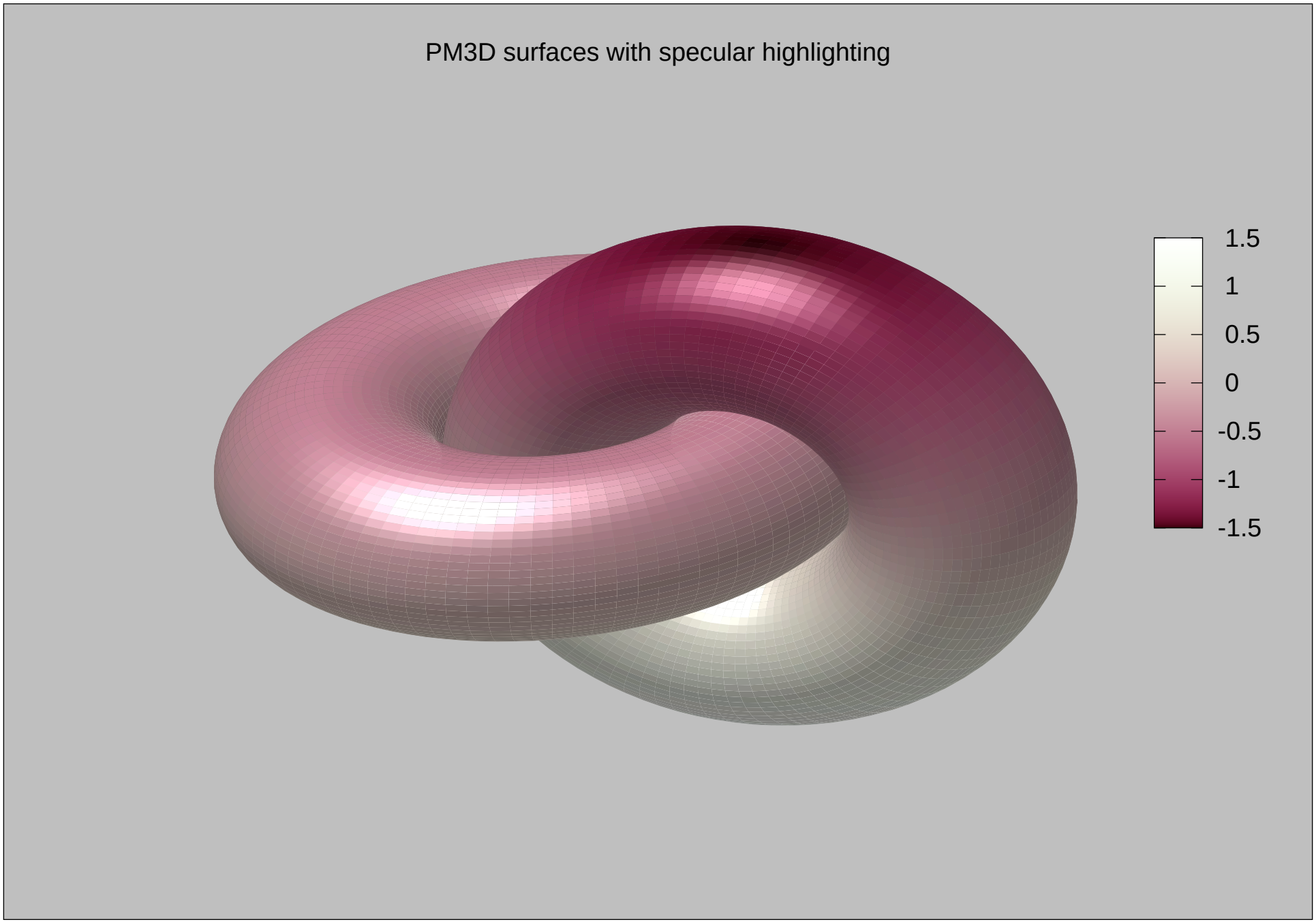

Transparent pattern-filled curves
Gaussian Distribution
= 0.5 = 0.5
= 2.0 = 1.0
= -1.0 = 2.0
1
0.8
0.6
0.4
0.2
0
-4
-2
0
2
4


Interlocking Tori - PM3D surface with depth sorting and transparency
1.5
1
0.5
0
-0.5
-1
-1.5


Kuen's Surface

pm3d lighting model with specular highlighting




PM3D surfaces with specular highlighting
1.5
1
0.5
0
-0.5
-1
-1.5