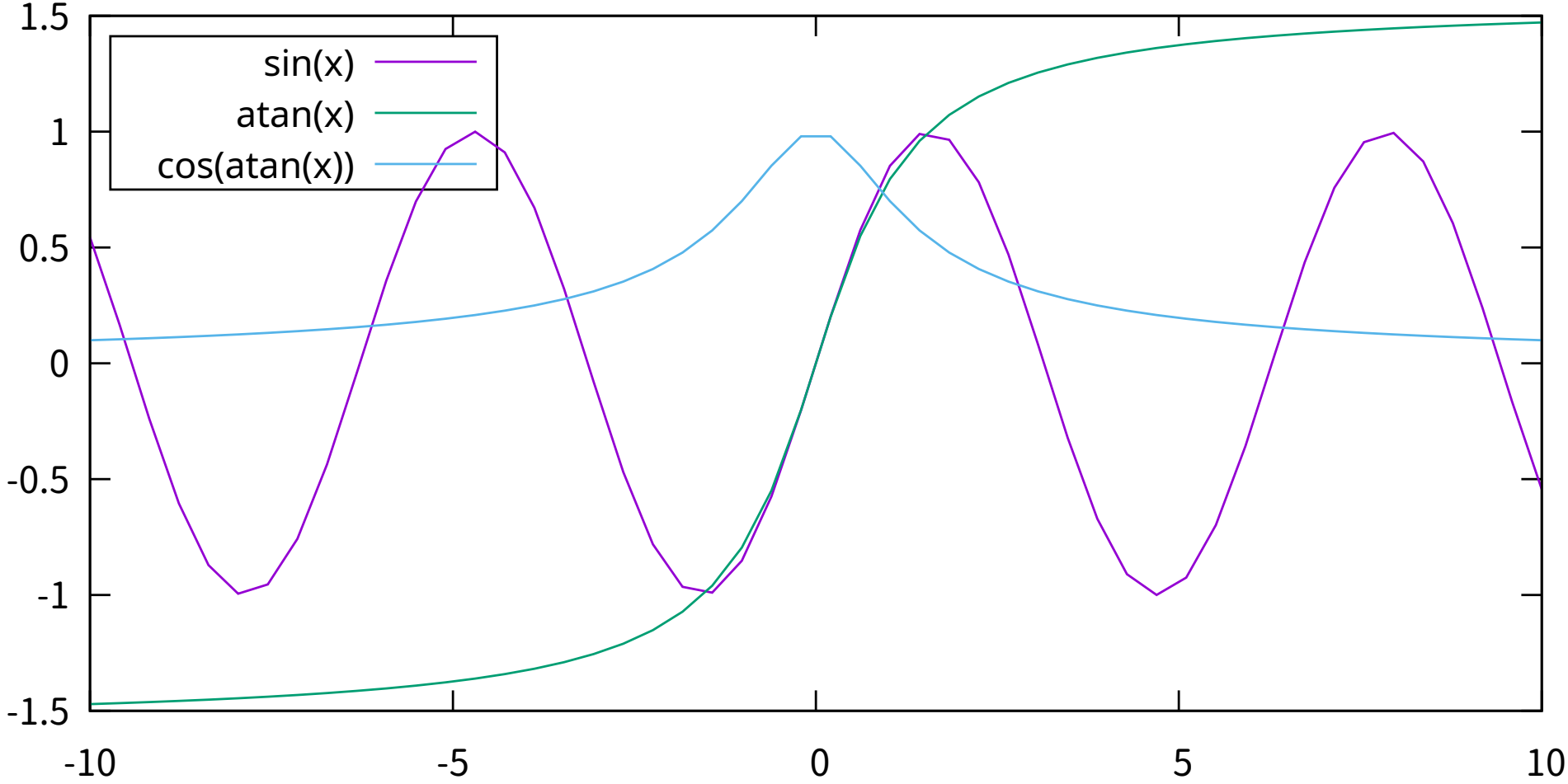
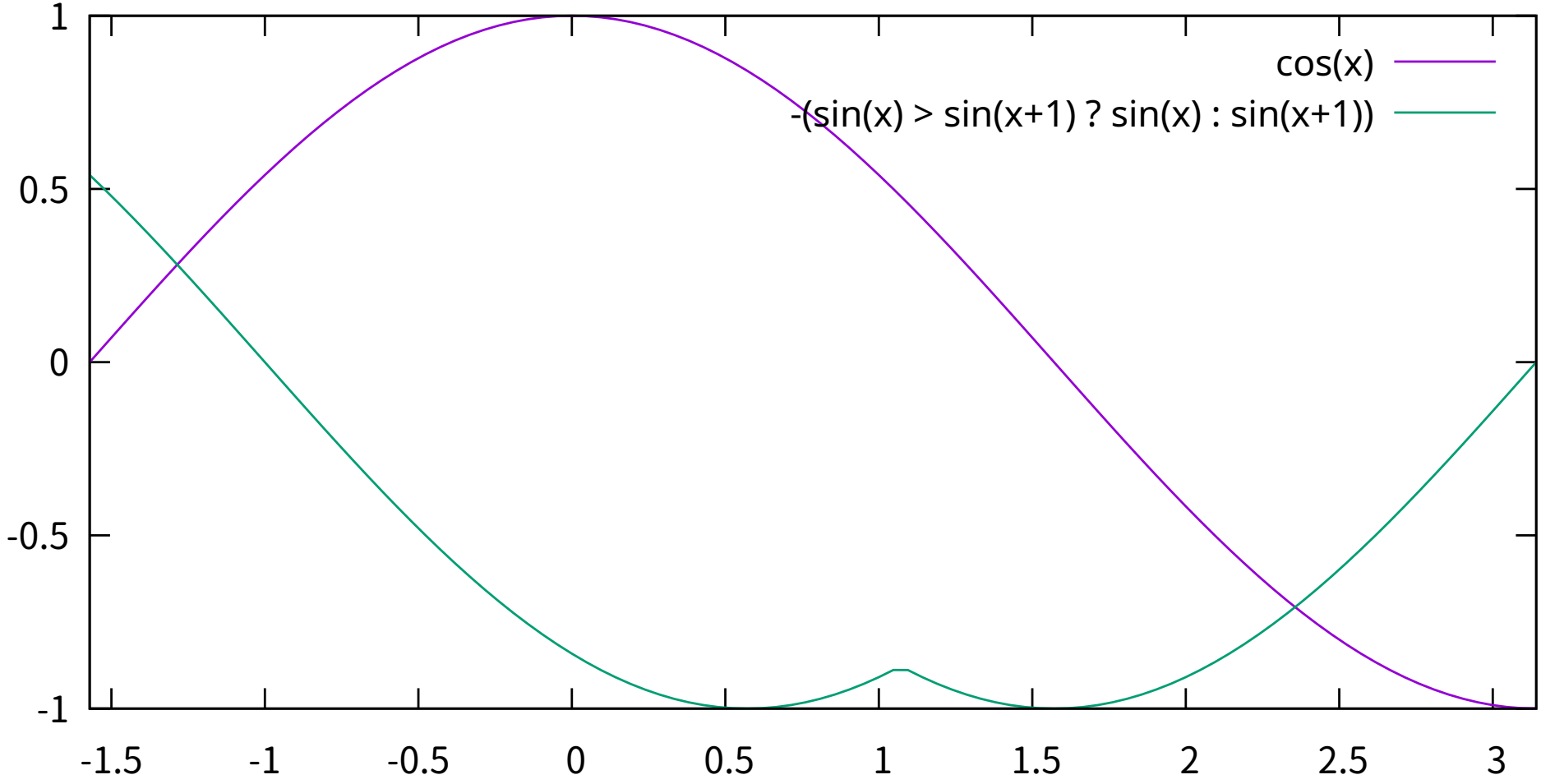
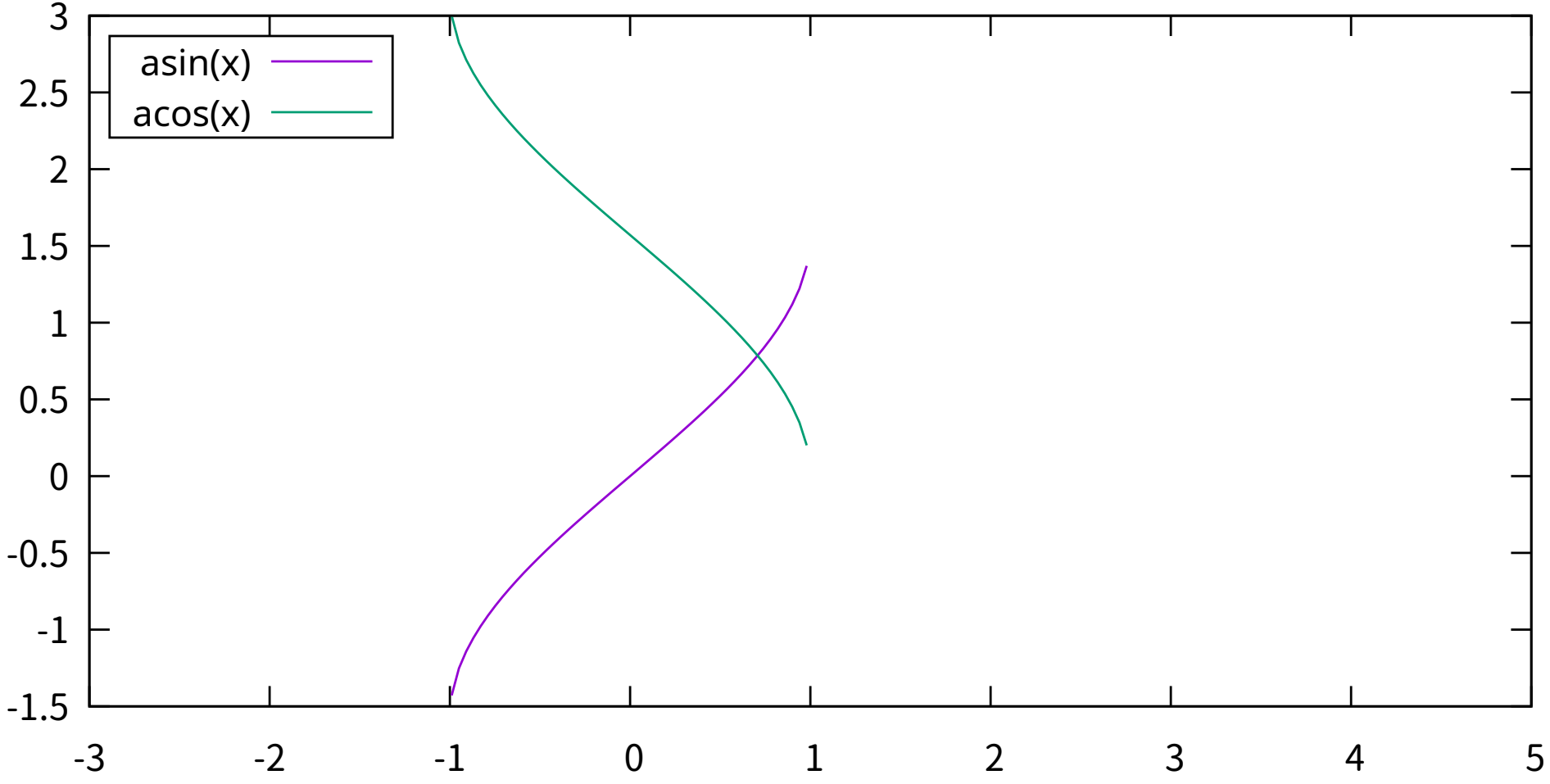
Simple Plots



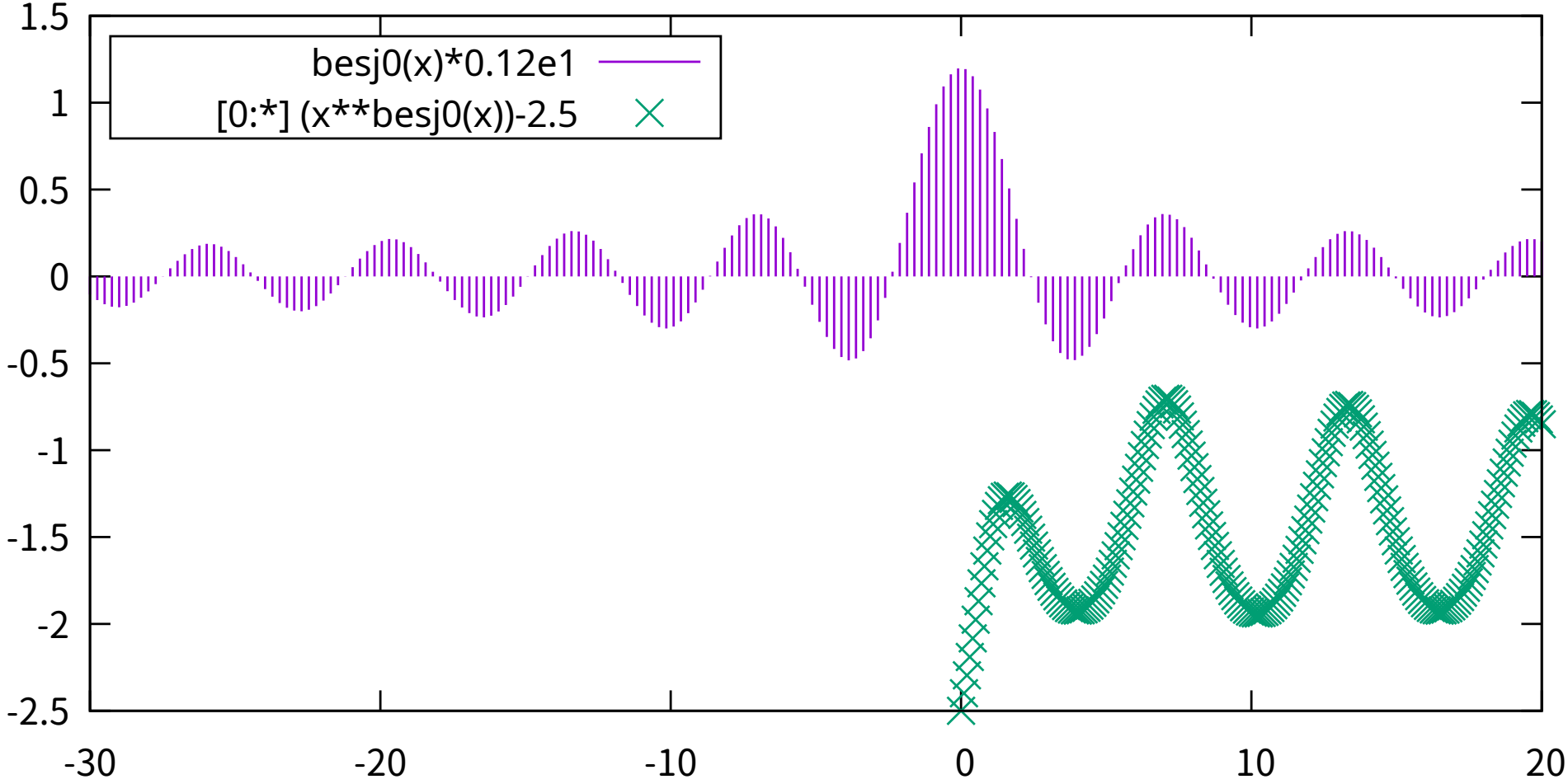
Simple Plots



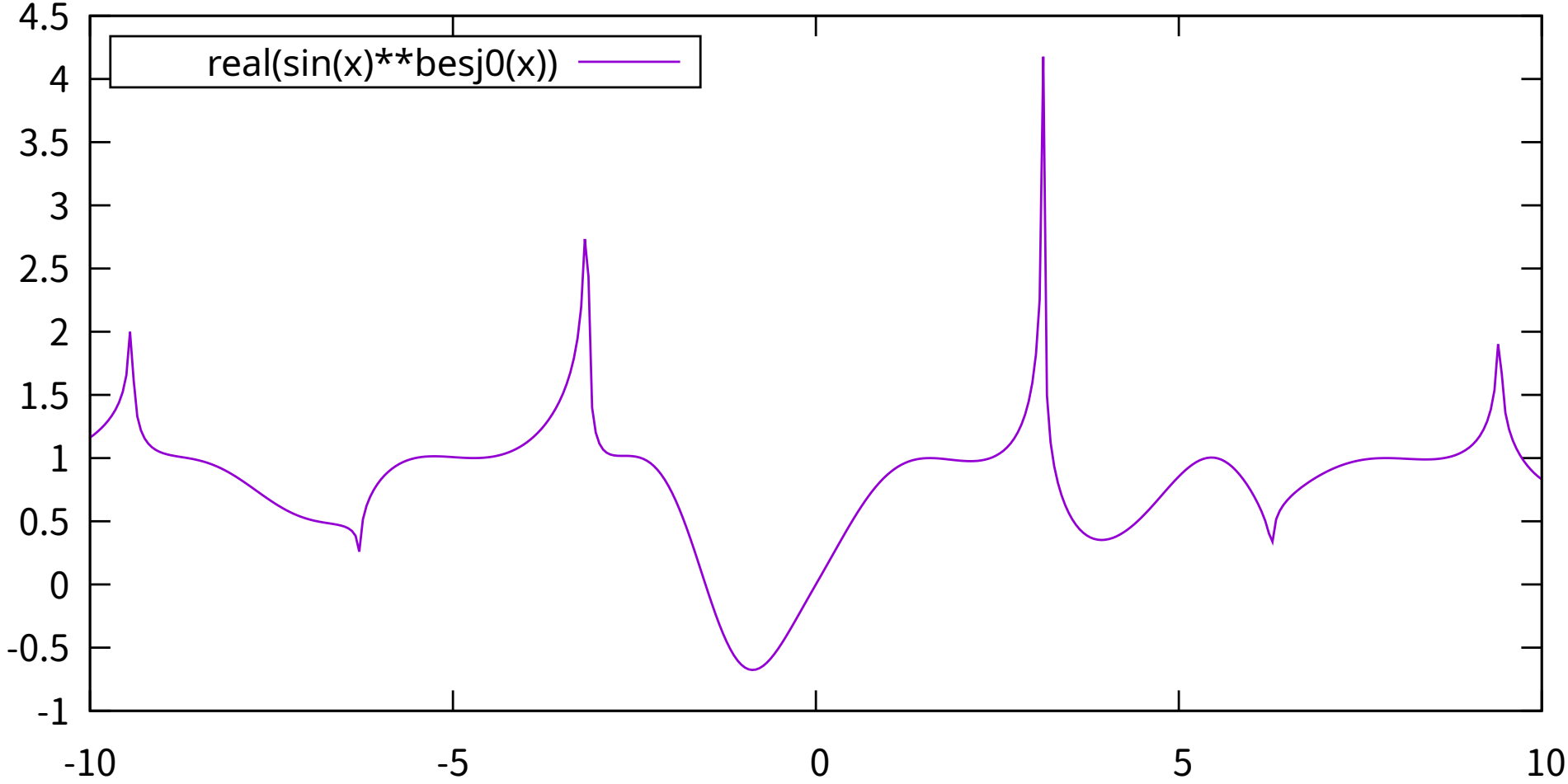
Simple Plots



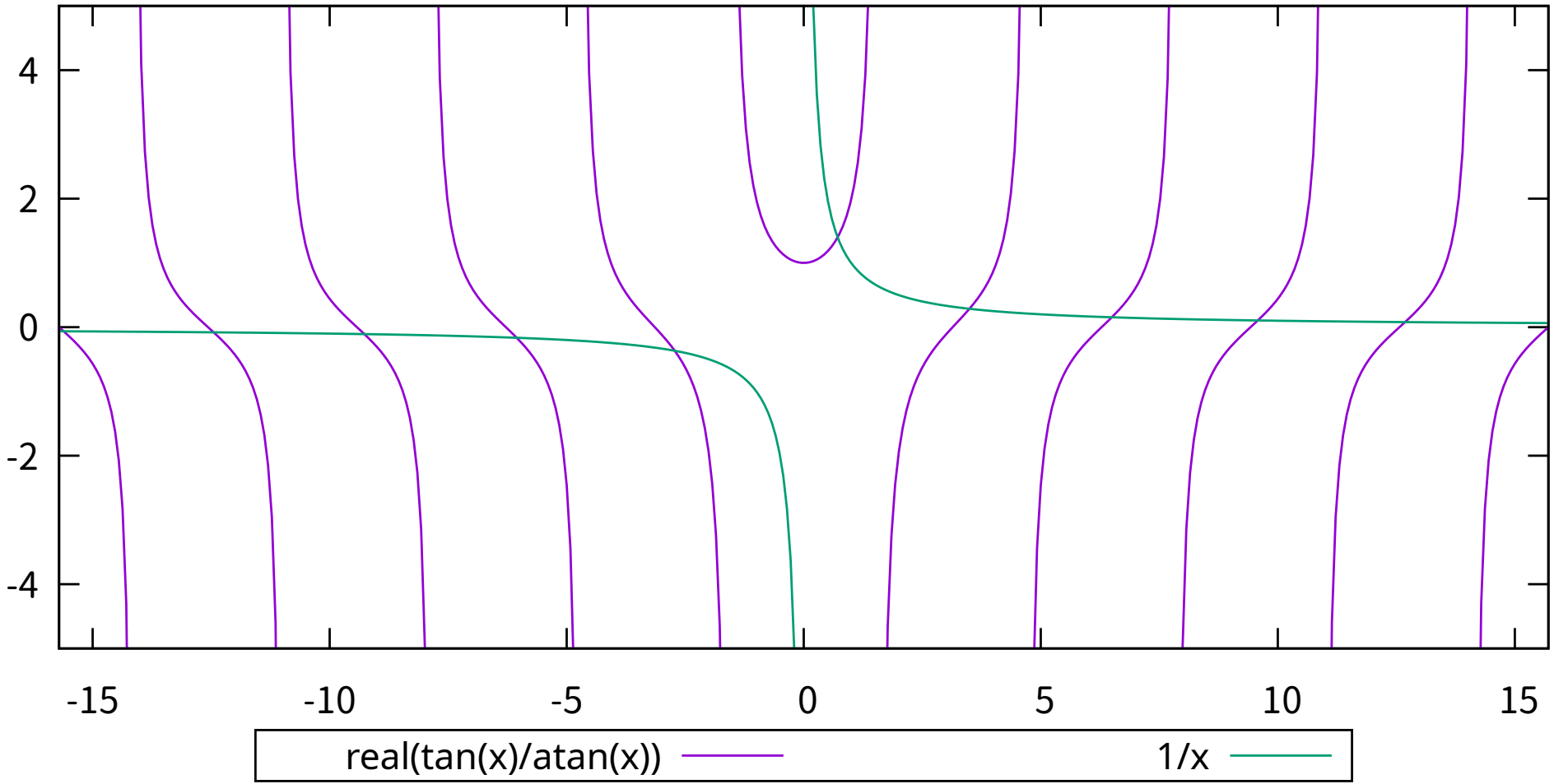
Simple Plots



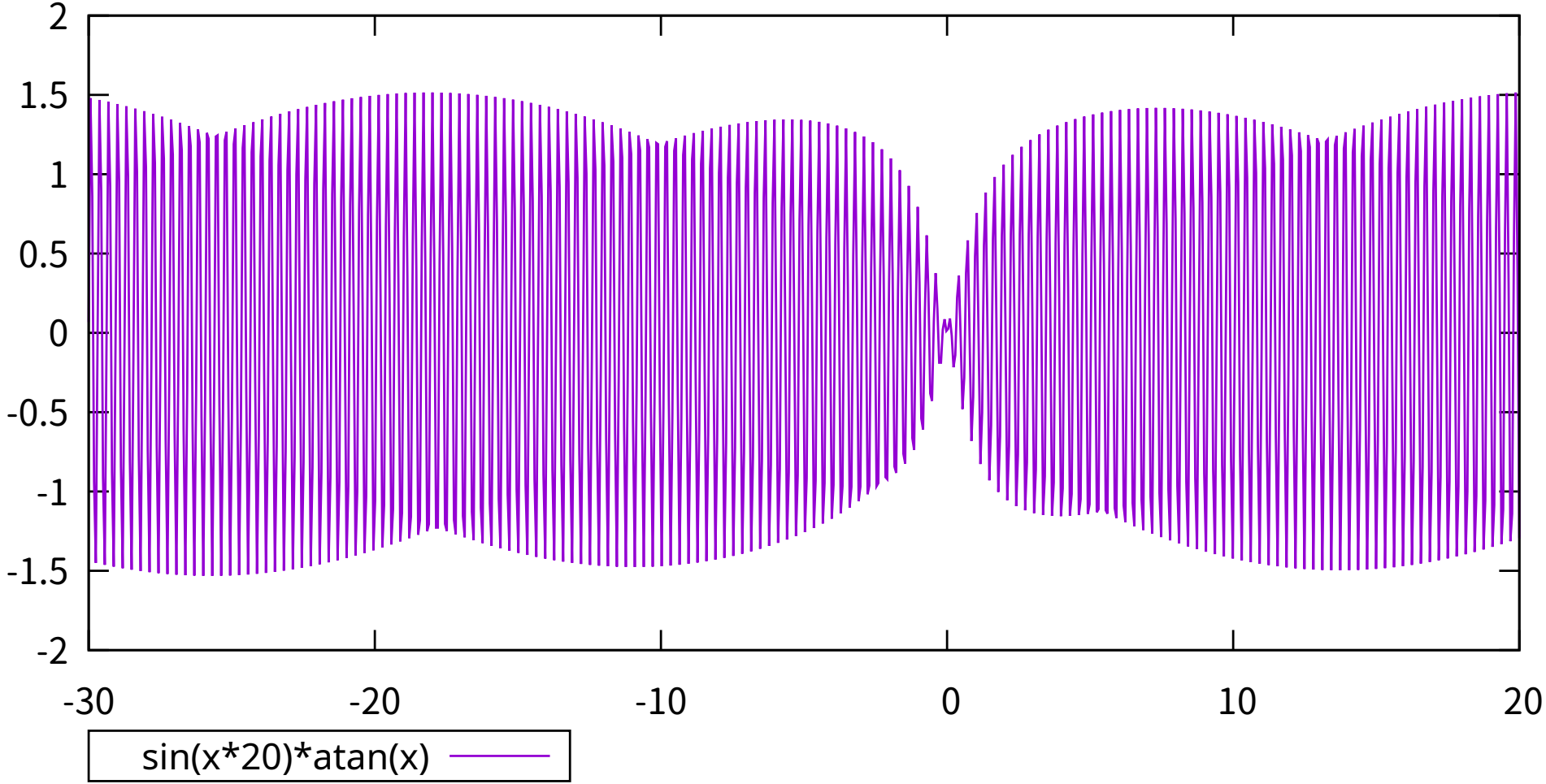
Simple Plots



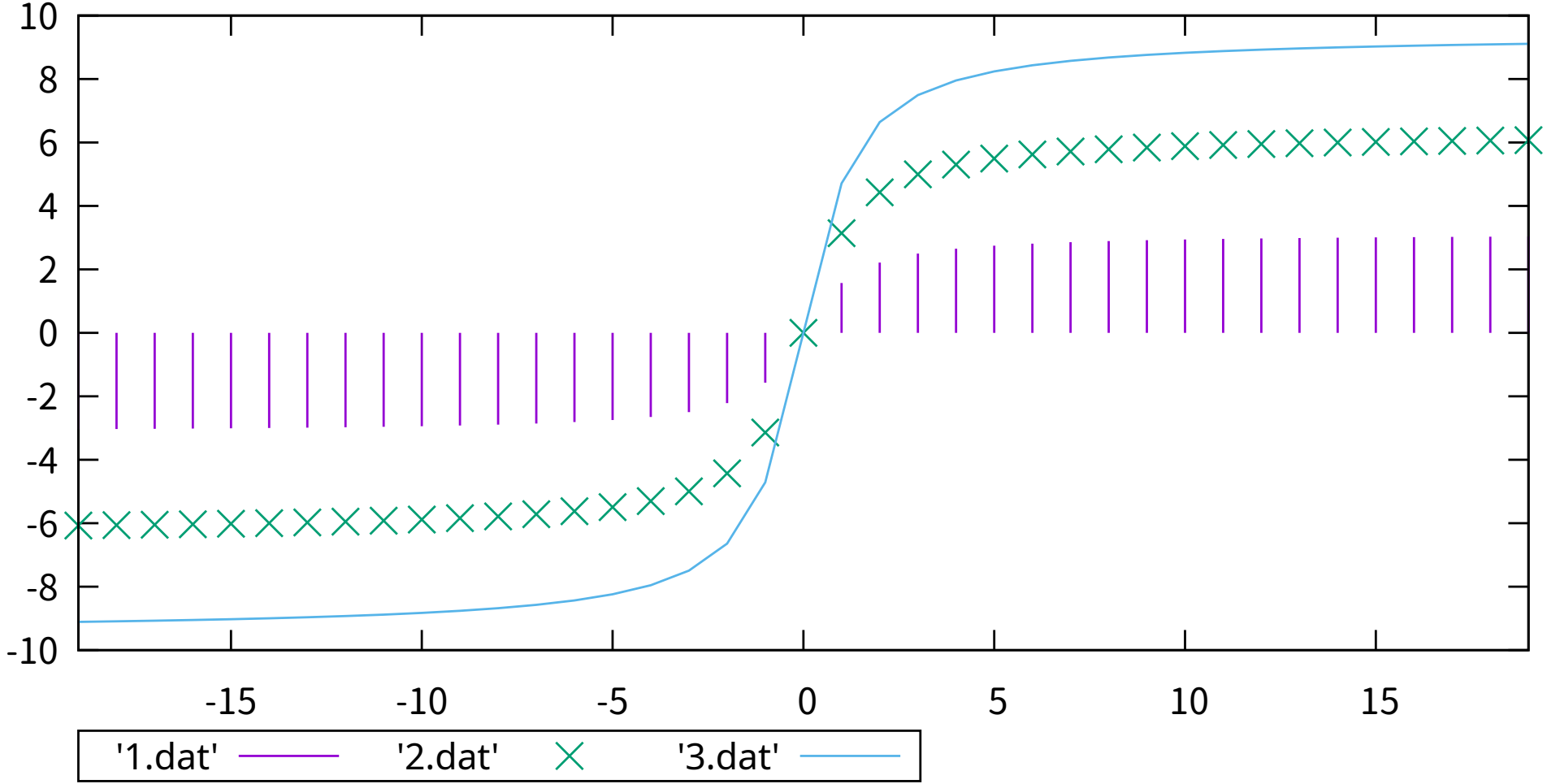
Simple Plots

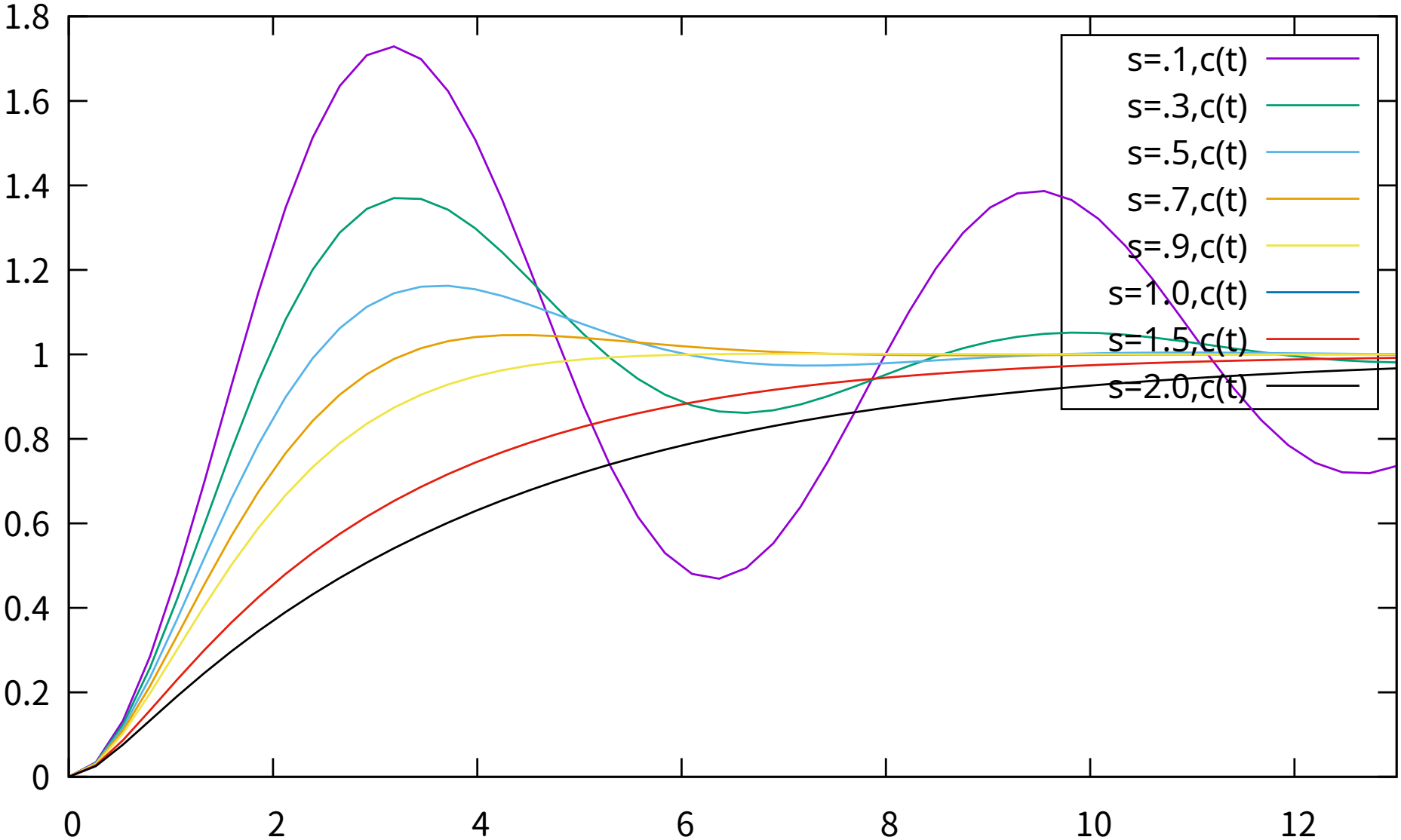


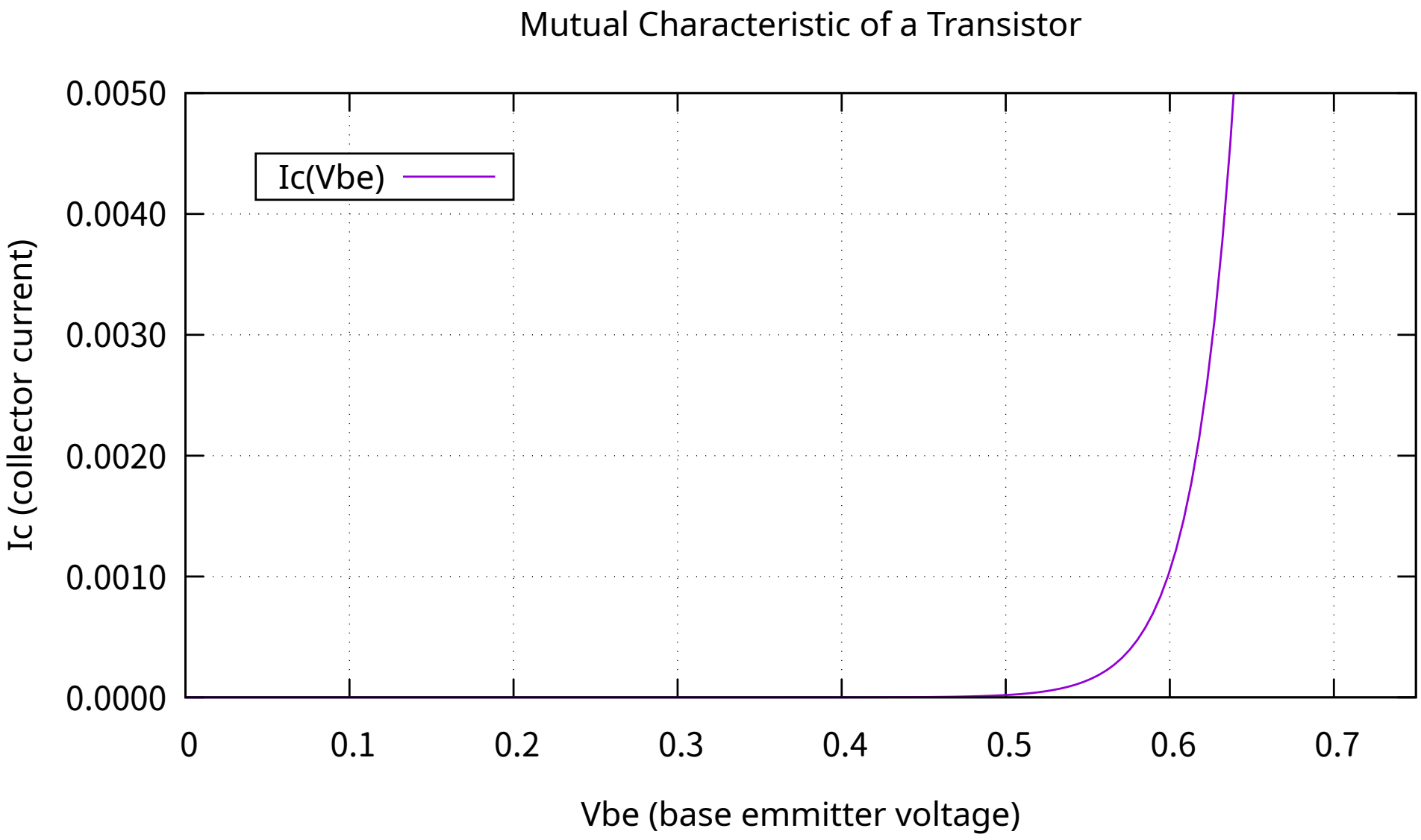
Simple Plots

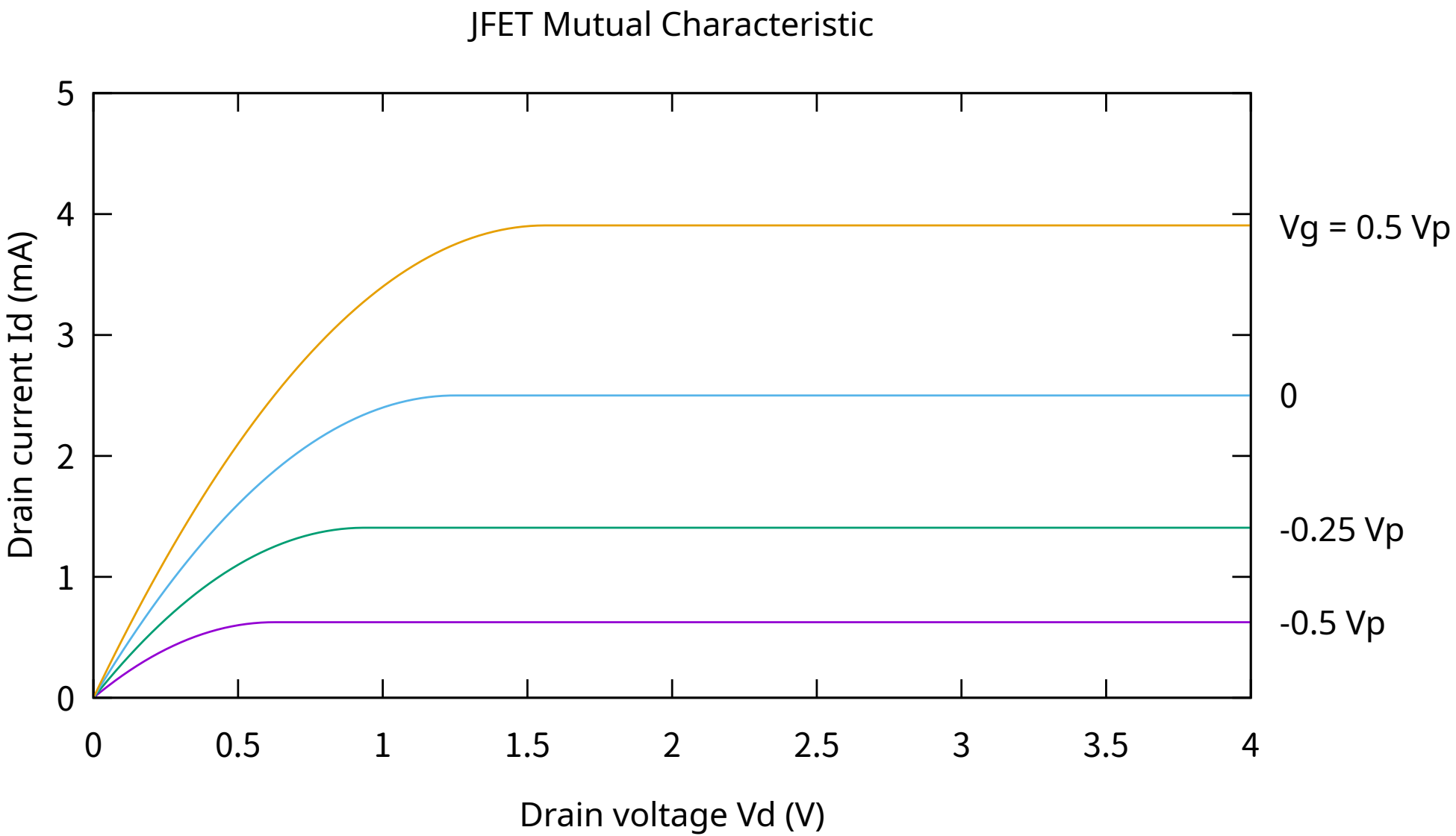


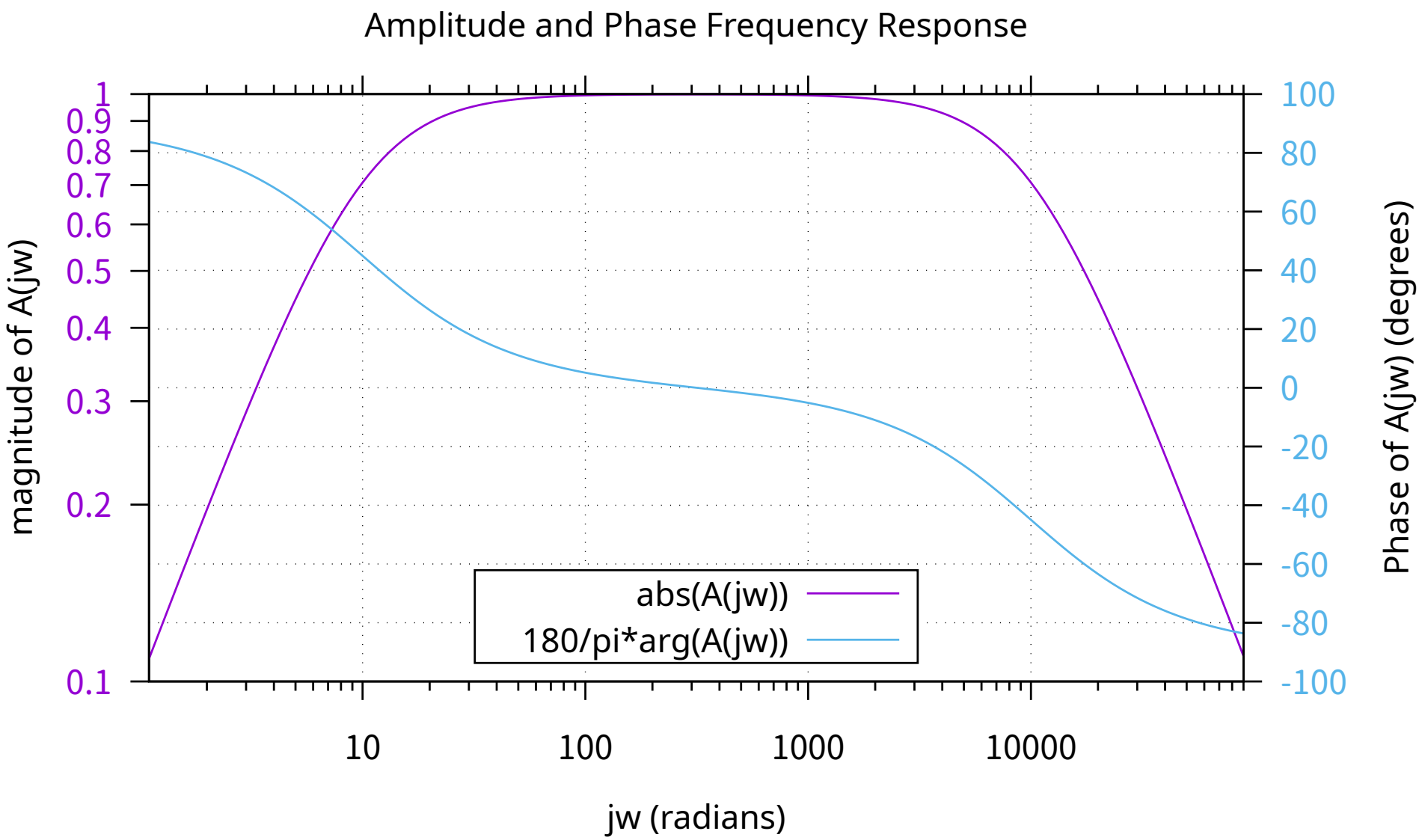
Simple Plots

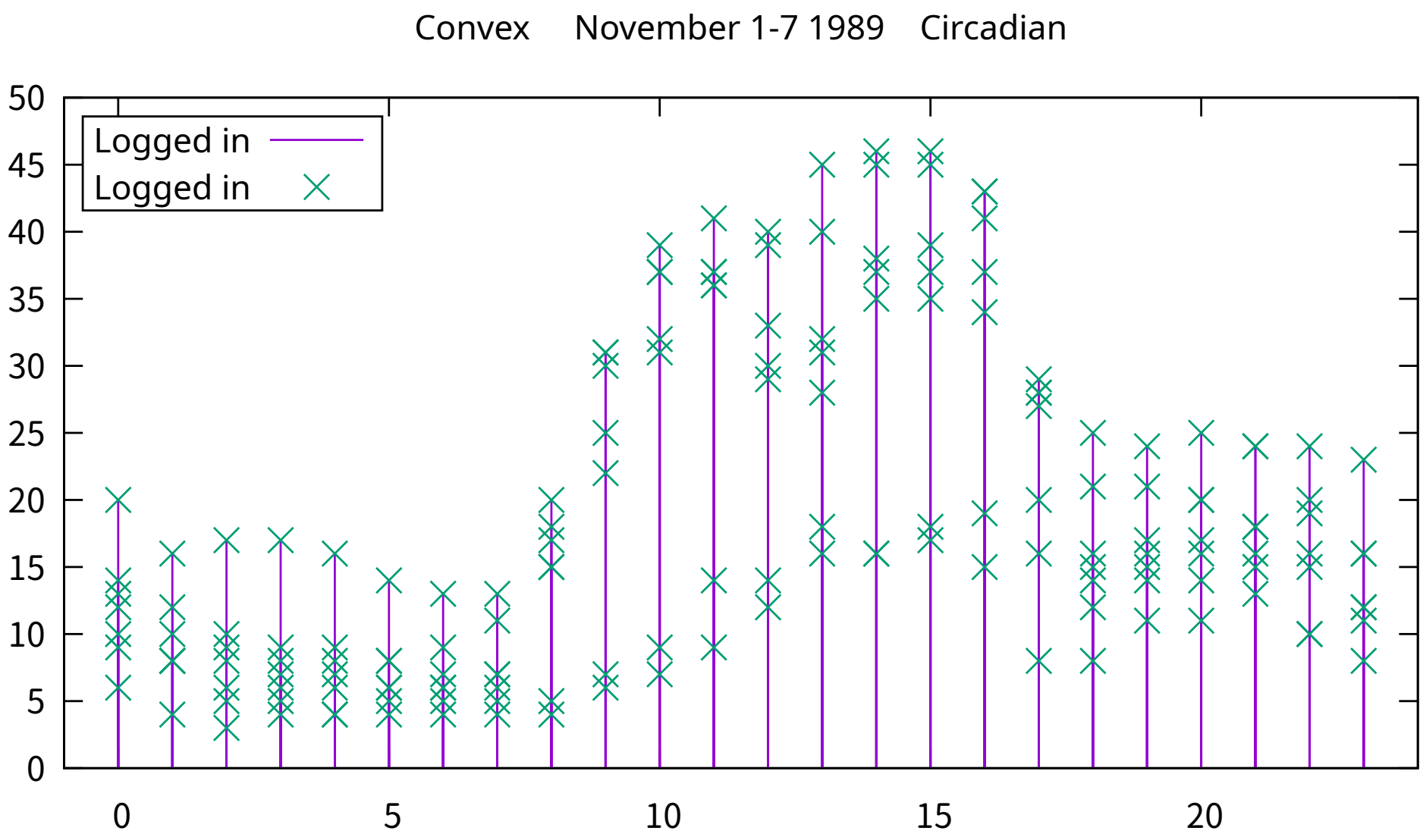


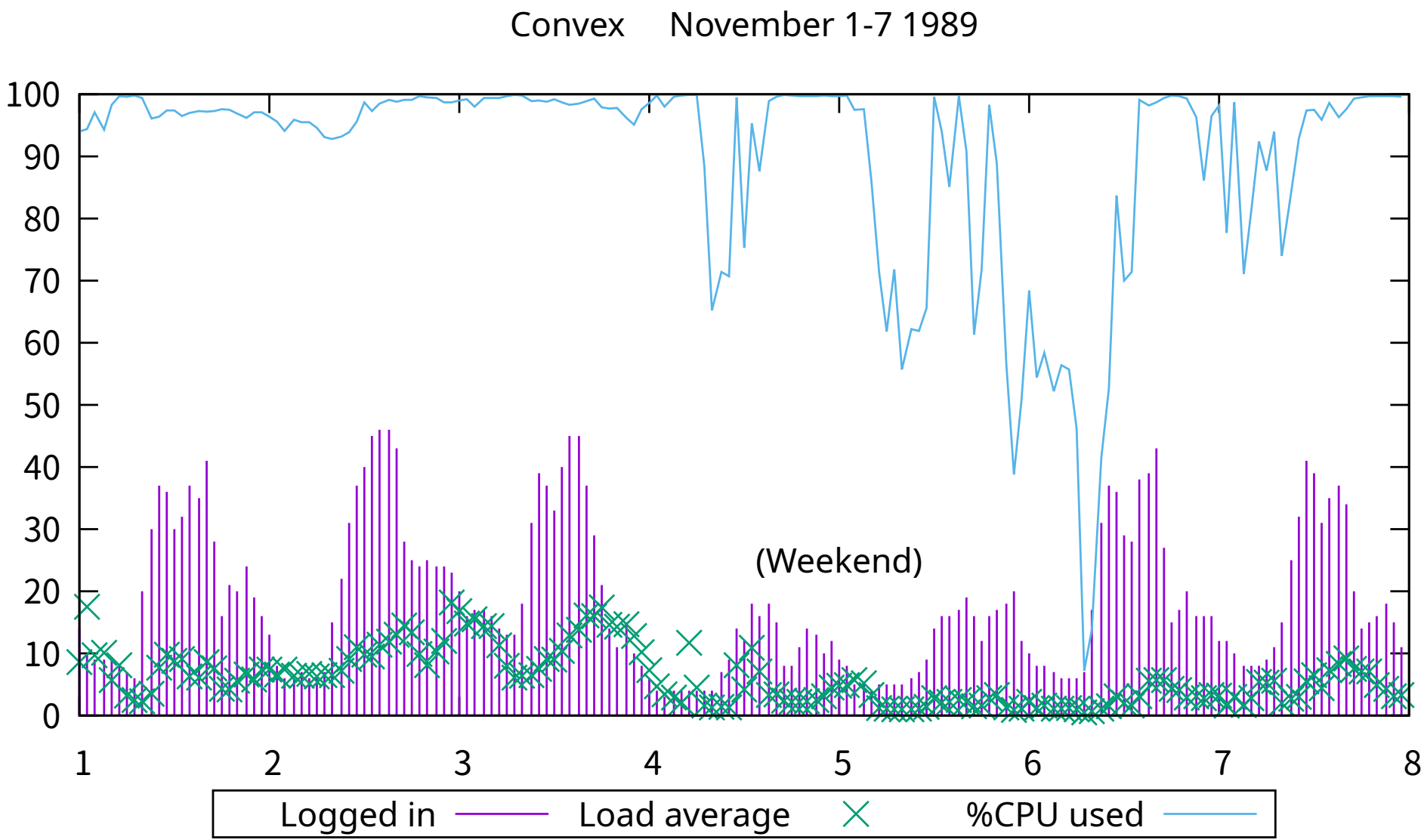

1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
12
s=.1,c(t)
s=.3,c(t)
s=.5,c(t)
s=.7,c(t)
s=.9,c(t)
s=1.0,c(t)
s=1.5,c(t)
s=2.0,c(t)


Mutual Characteristic of a Transistor
Ic(Vbe)
Ic (collector current)
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
Vbe (base emmitter voltage)

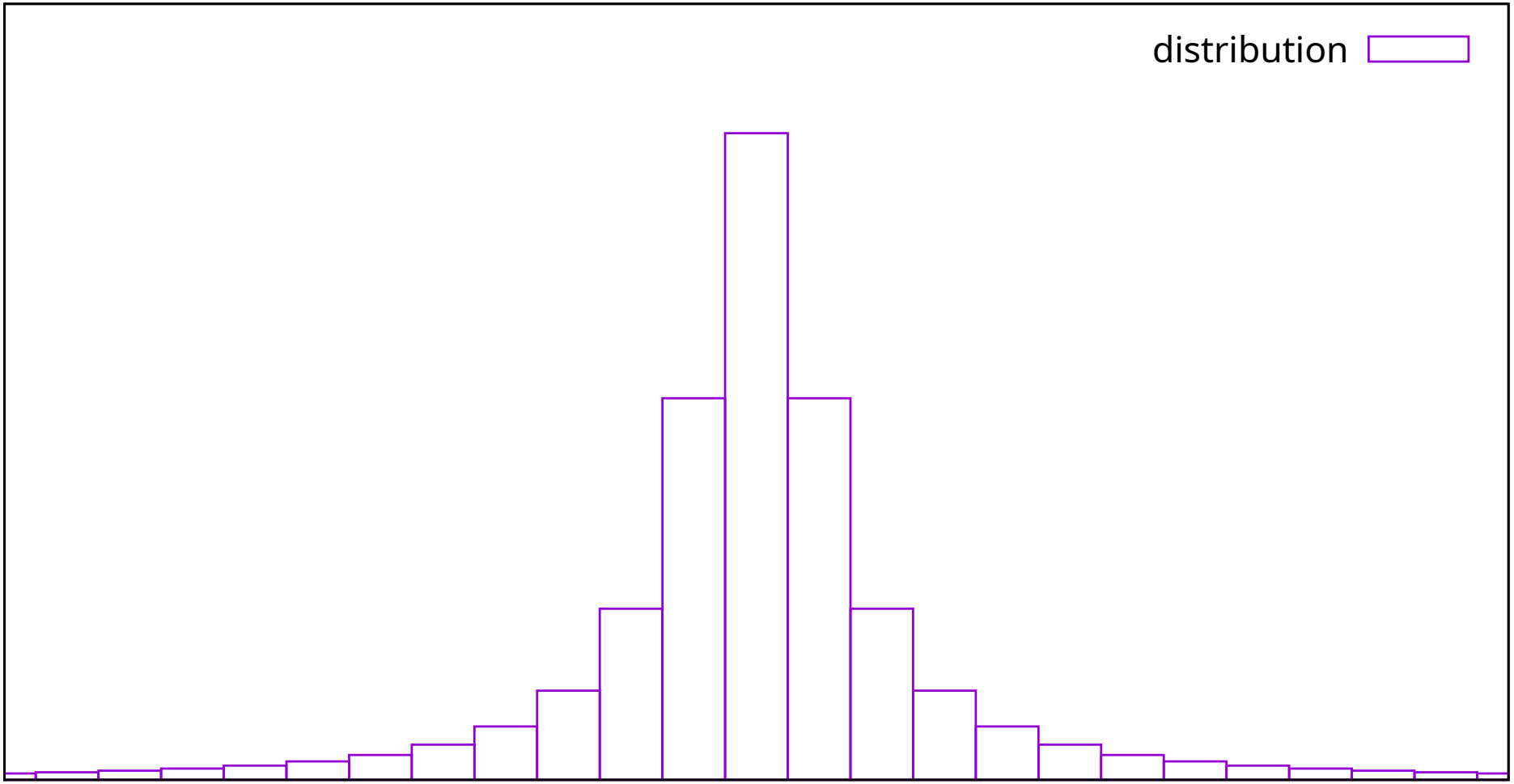

JFET Mutual Characteristic
Drain current Id (mA)
Drain voltage Vd (V)
0
1
2
3
4
5
0
0.5
1
1.5
2
2.5
3
3.5
4
Vg = 0.5 Vp
0
-0.25 Vp
-0.5 Vp


Amplitude and Phase Frequency Response
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

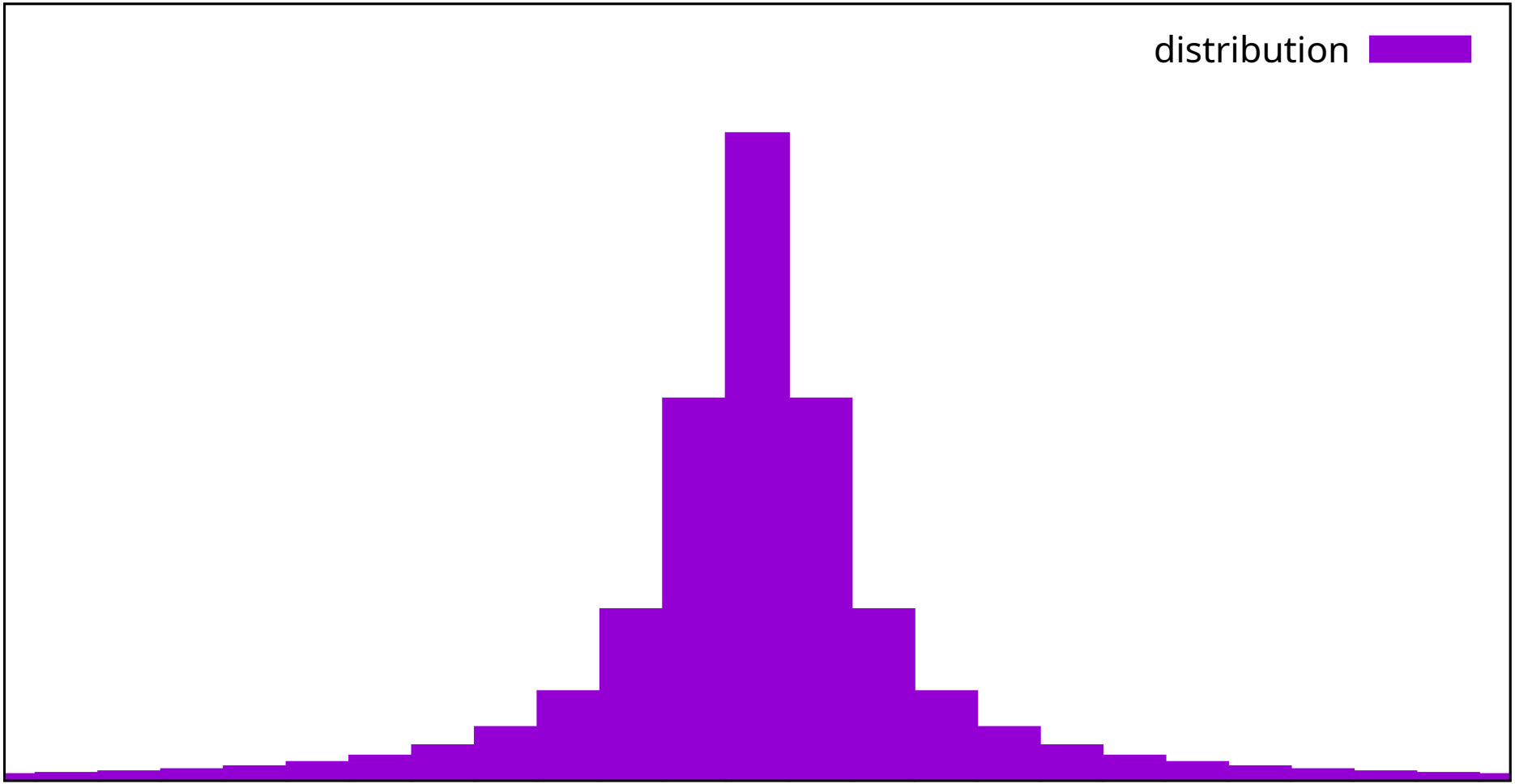

Convex November 1-7 1989 Circadian
Logged in
Logged in
50
45
40
35
30
25
20
15
10
5
0
0
5
10
15
20


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

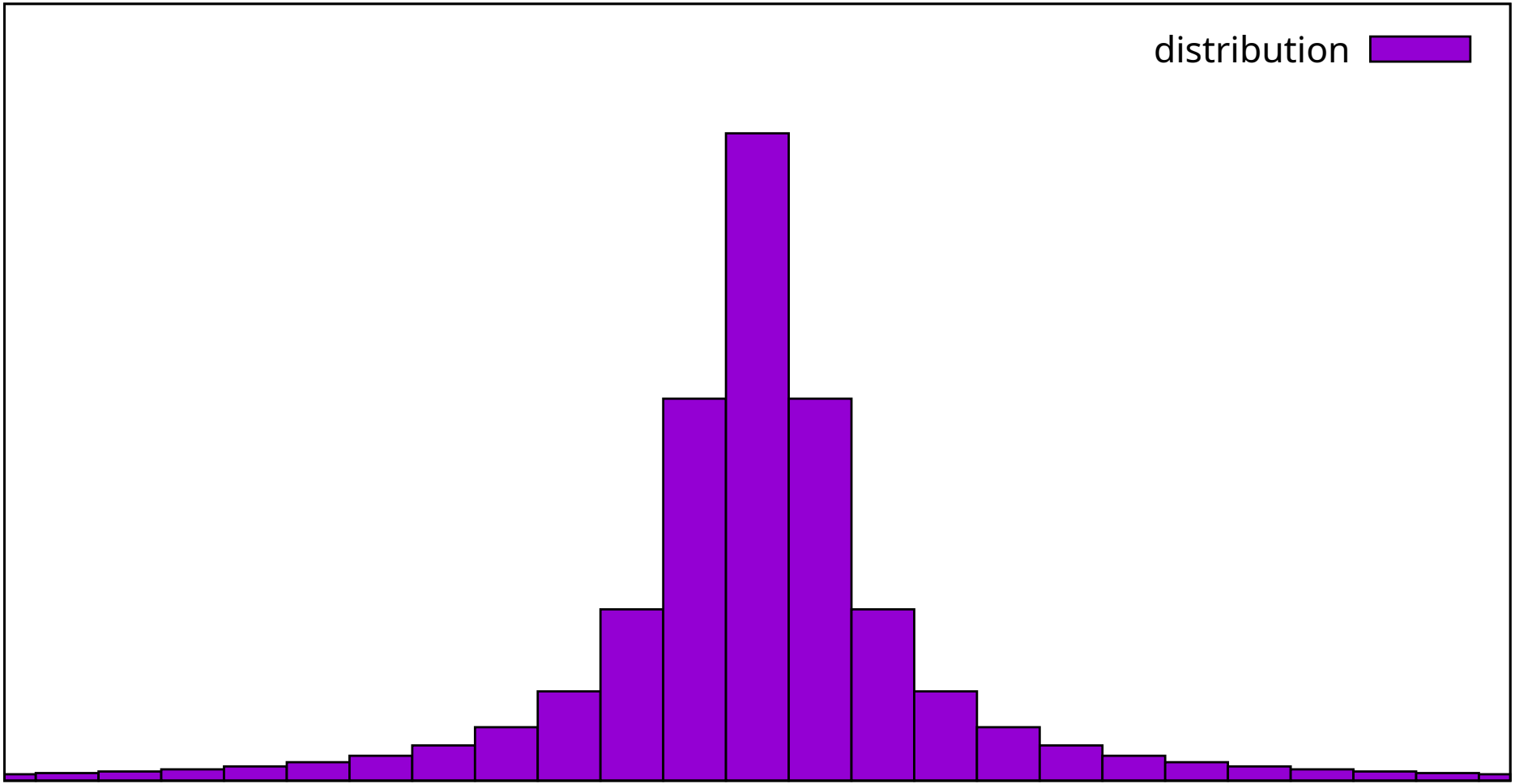
A demonstration of boxes with default properties



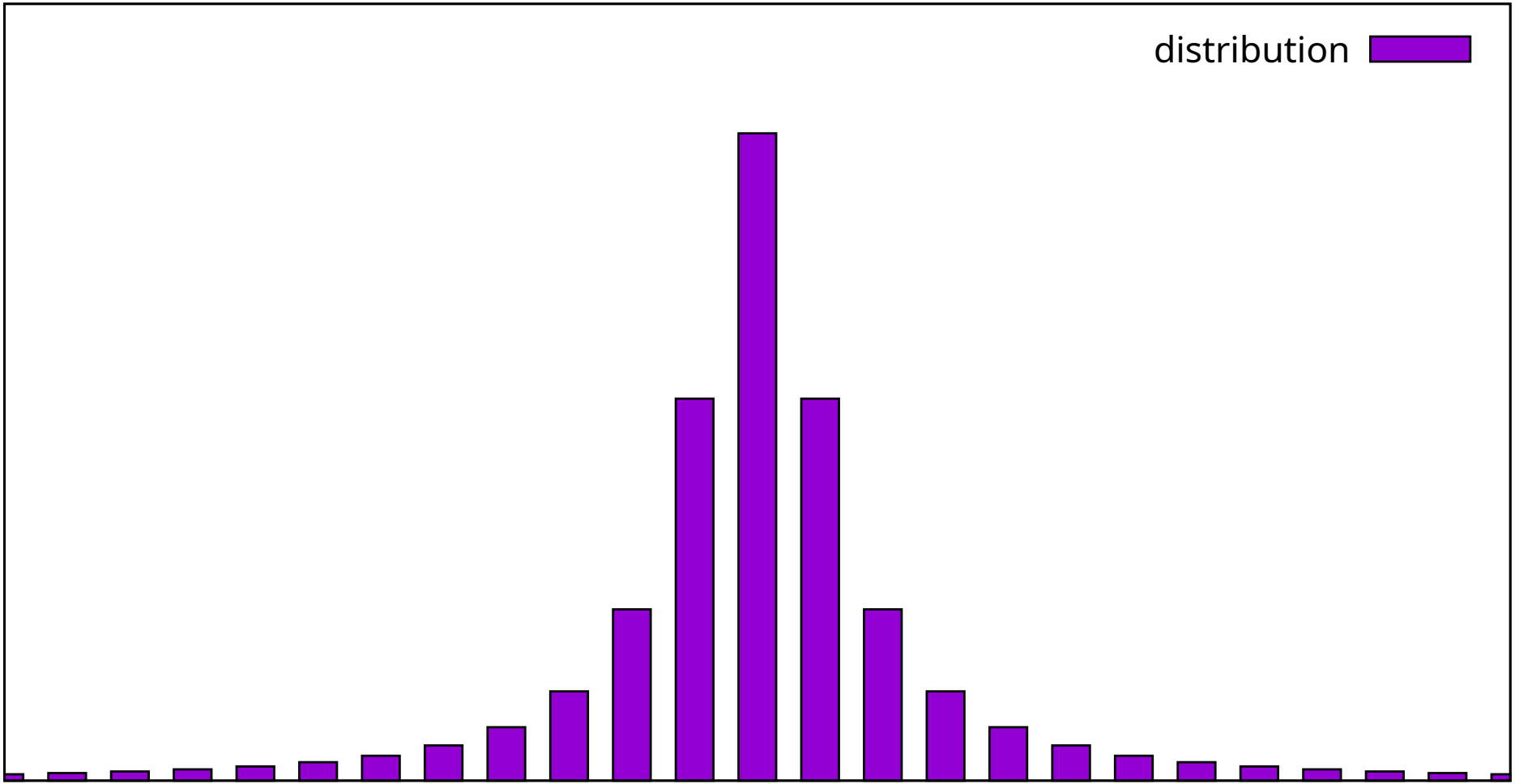
A demonstration of boxes with style fill solid 1.0



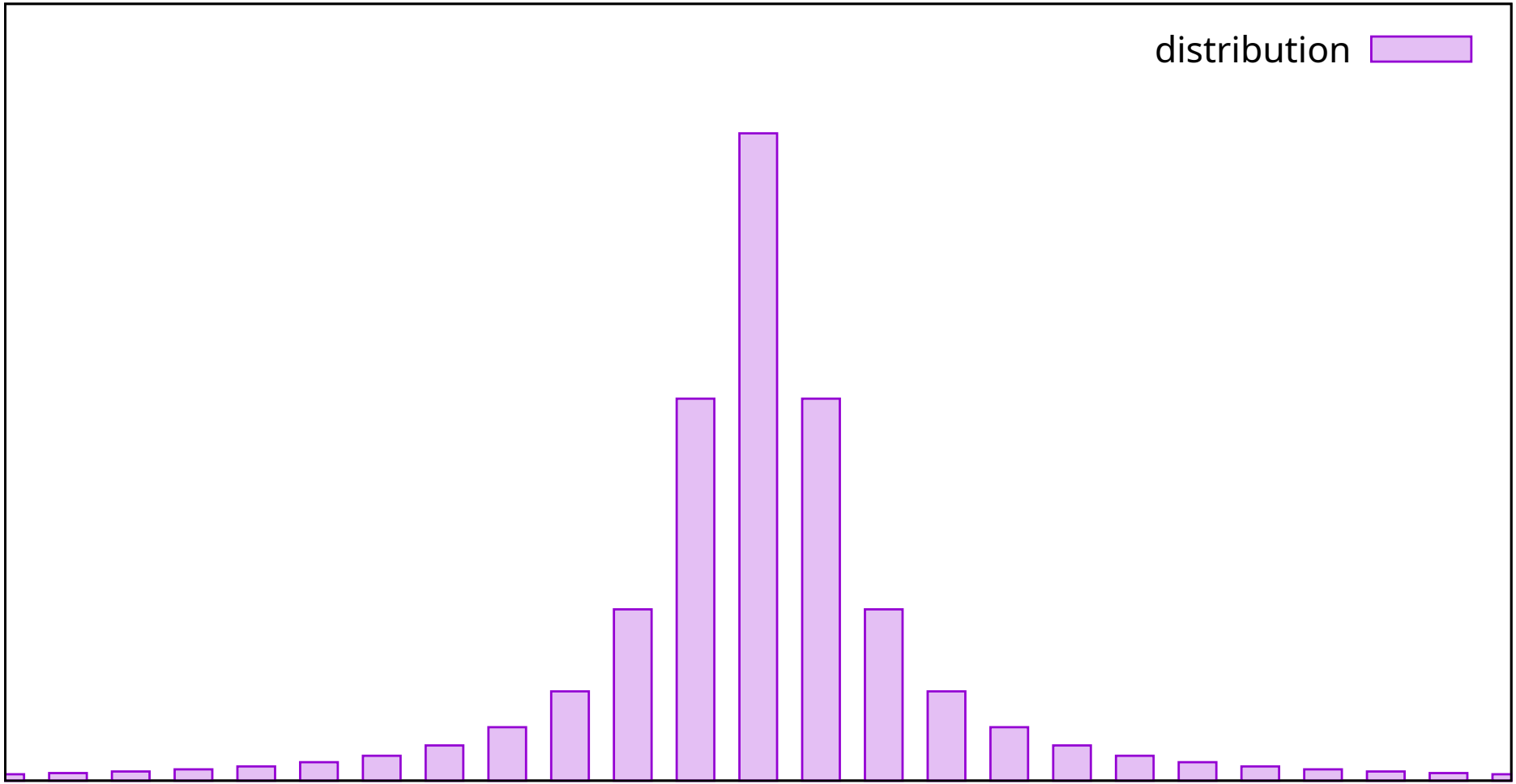
A demonstration of boxes with style fill solid border -1



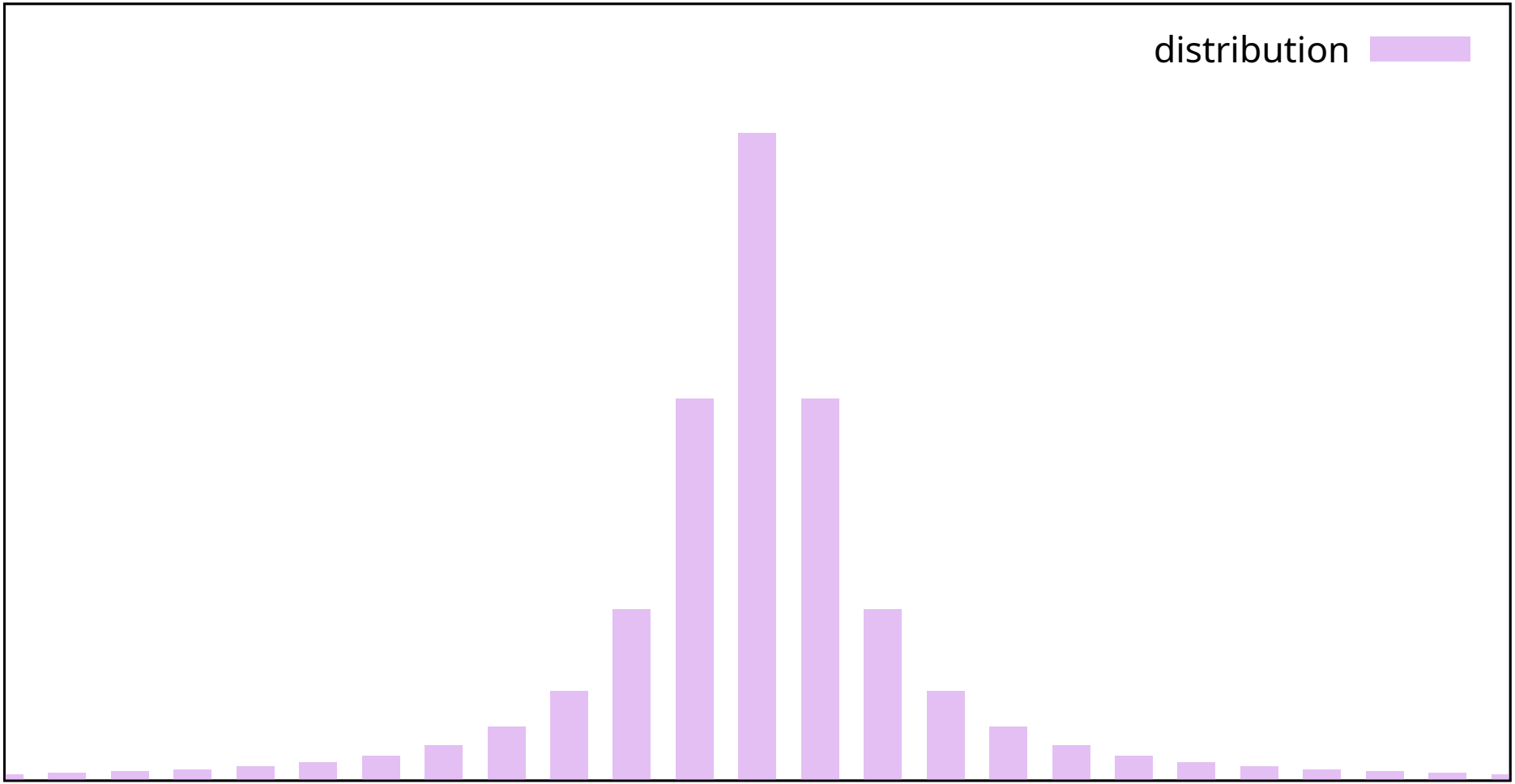
Filled boxes of reduced width



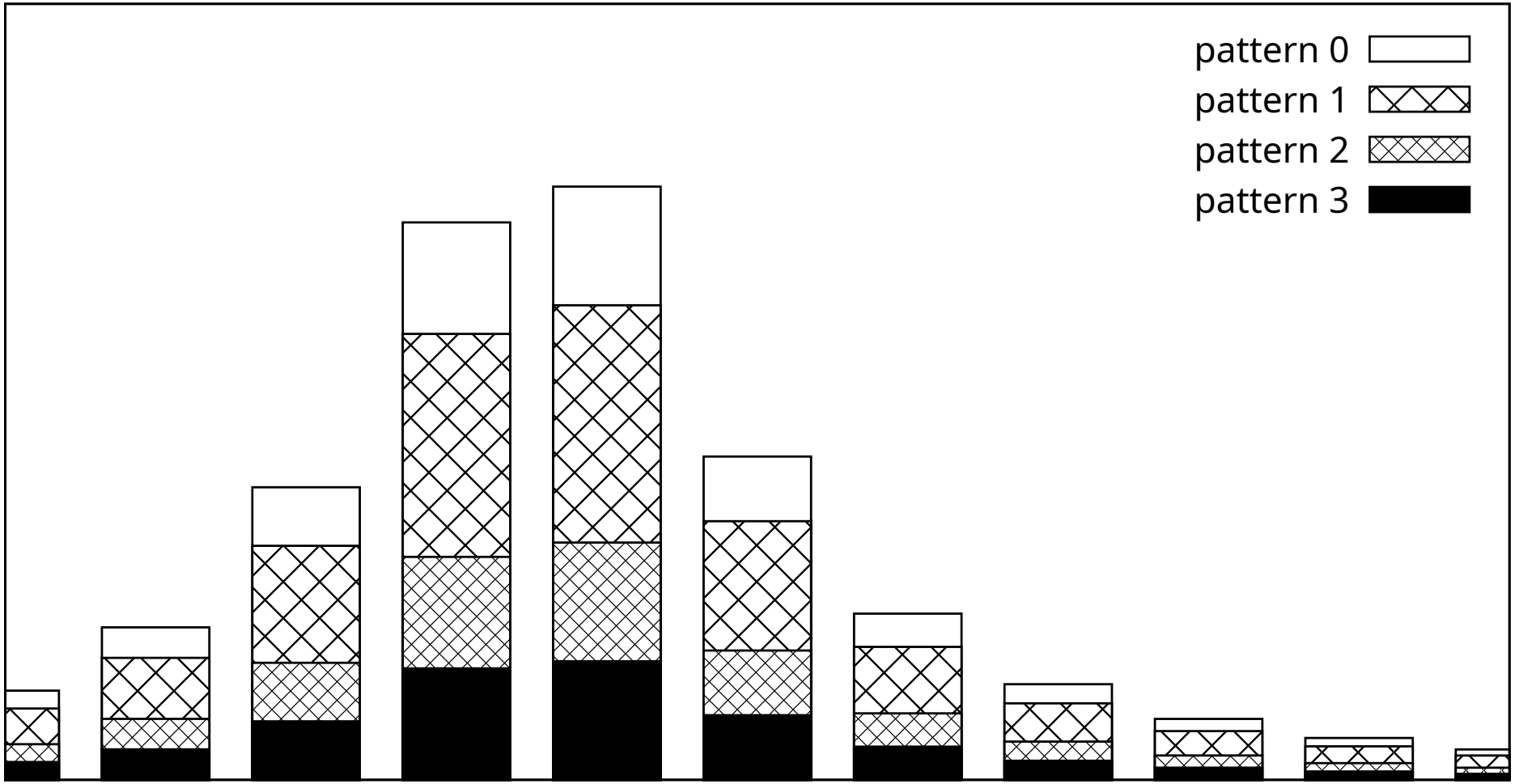
Filled boxes at 50% fill density



A demonstration of boxes with style fill solid 0.25 noborder



A demonstration of boxes in mono with style fill pattern



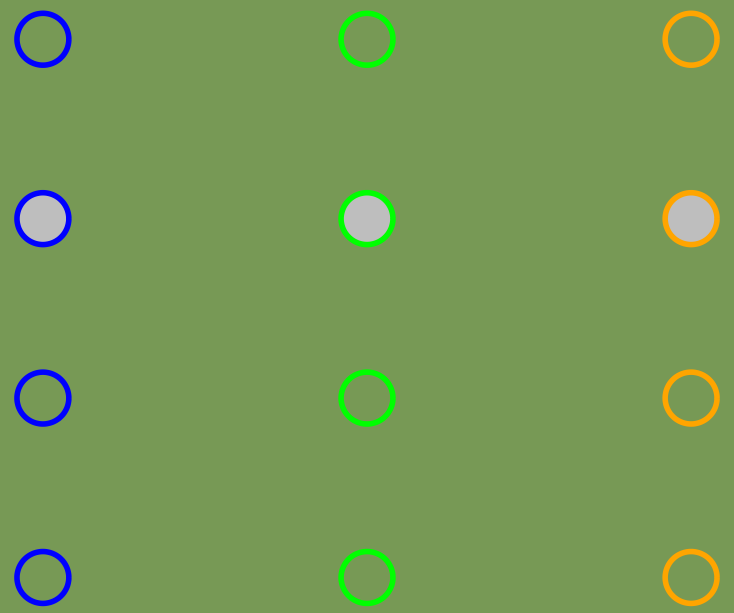
All borders should be drawn in color
gnuplot 6.1 patchlevel 0

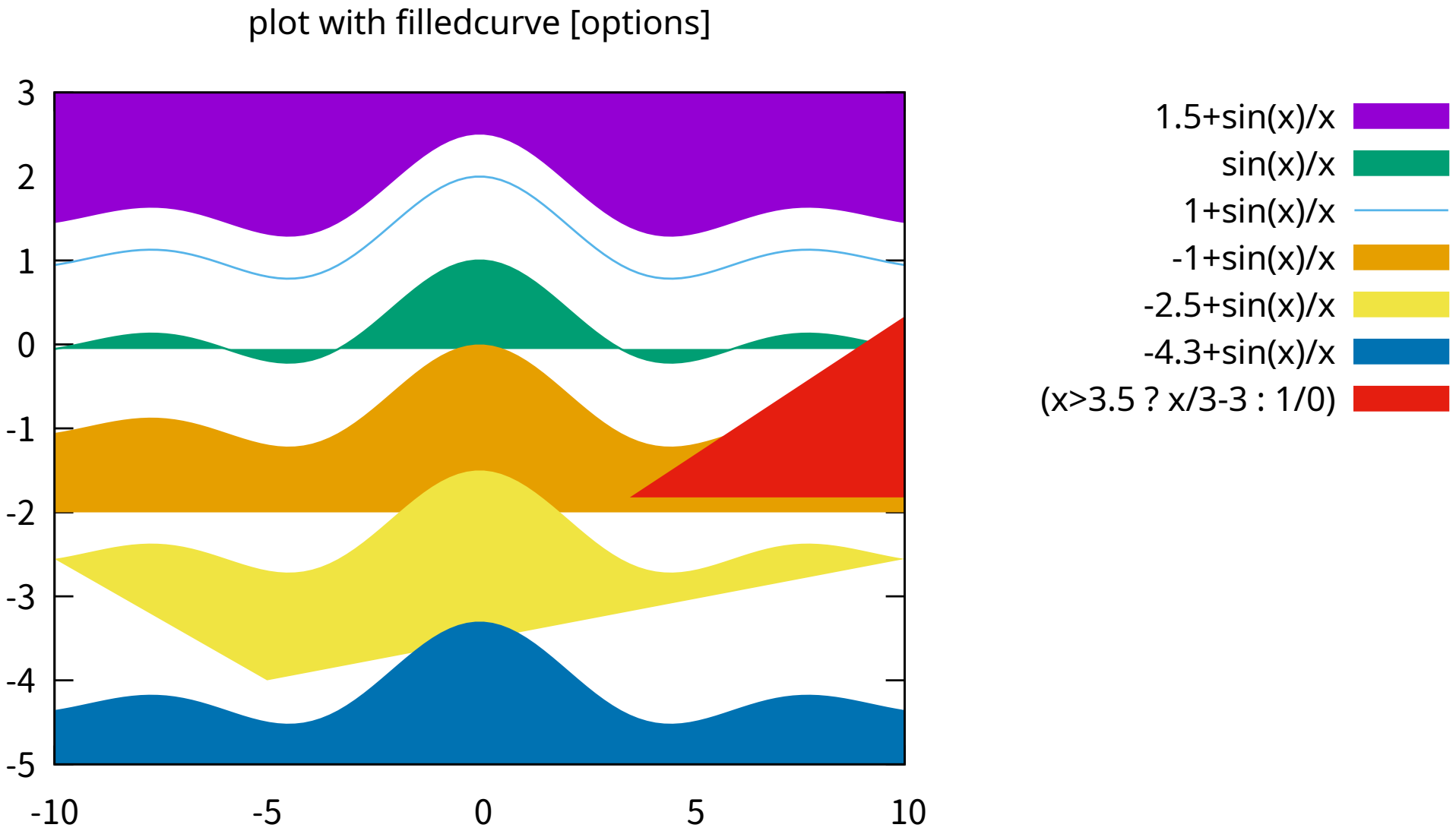
set obj fs empty border <color>

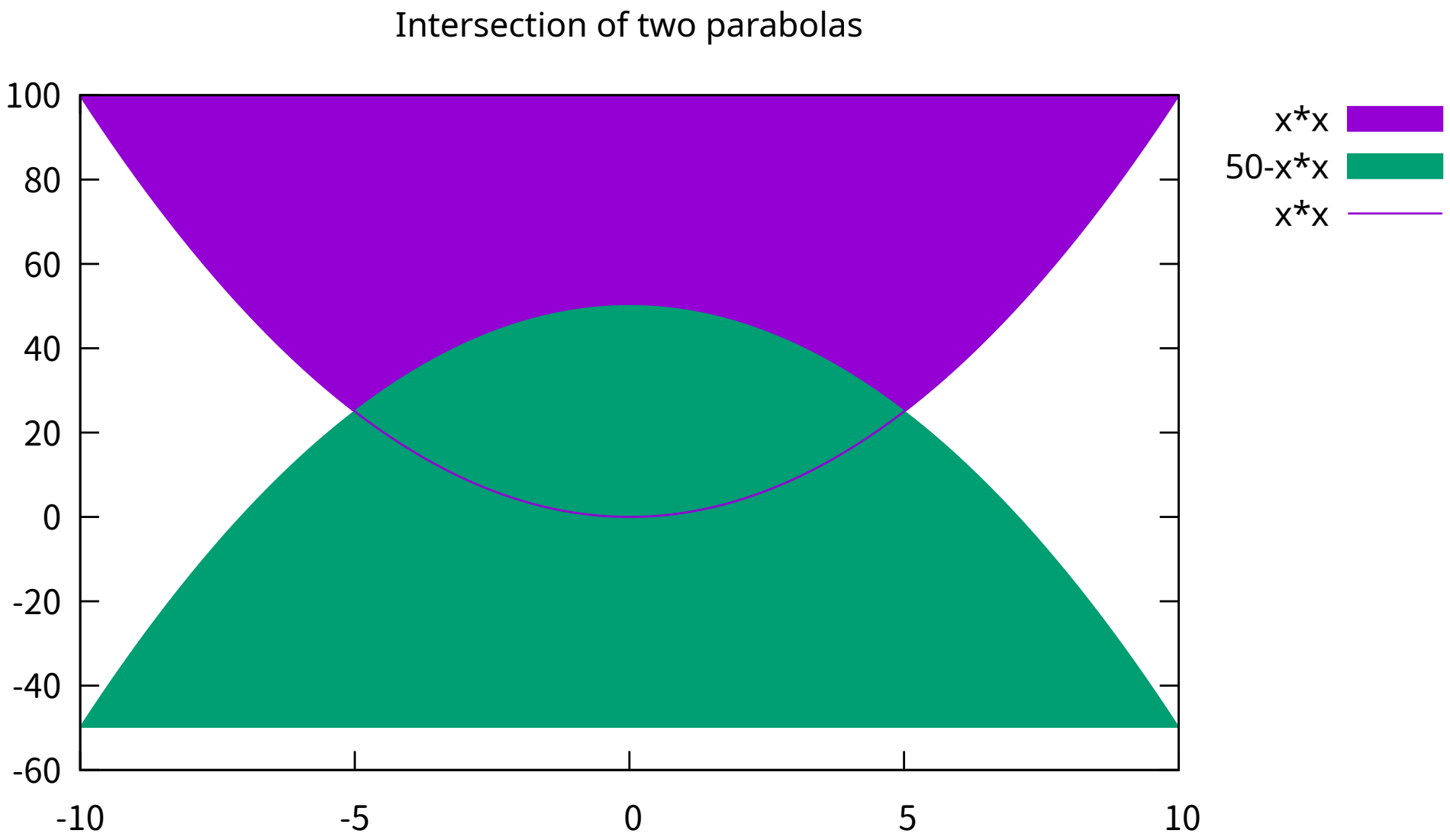
set obj fs solid border <color> fc 'gray'

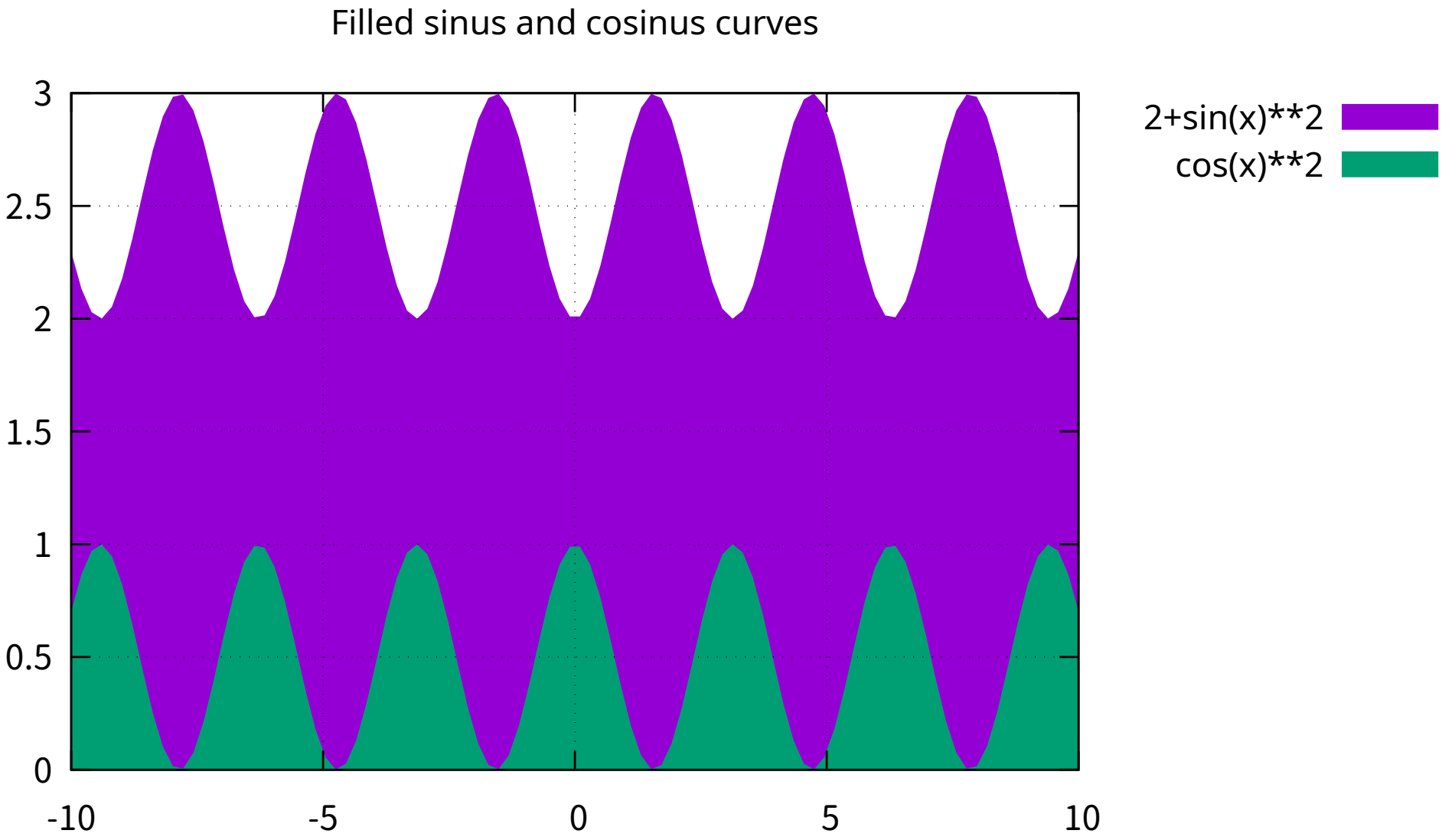
set obj fs empty border
set obj lc <color>

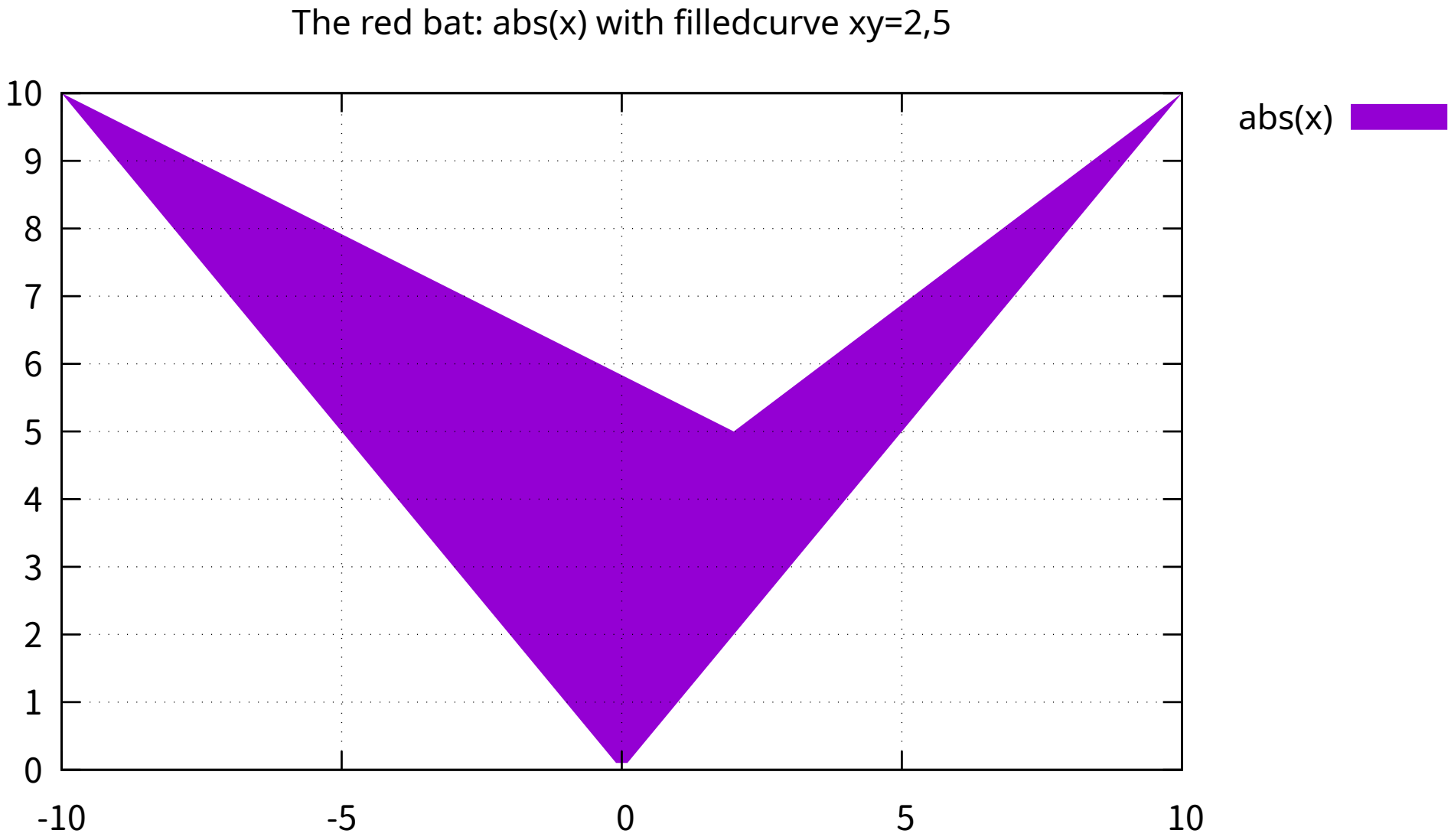
set obj fs empty border lc 'black'
set obj lc <color>

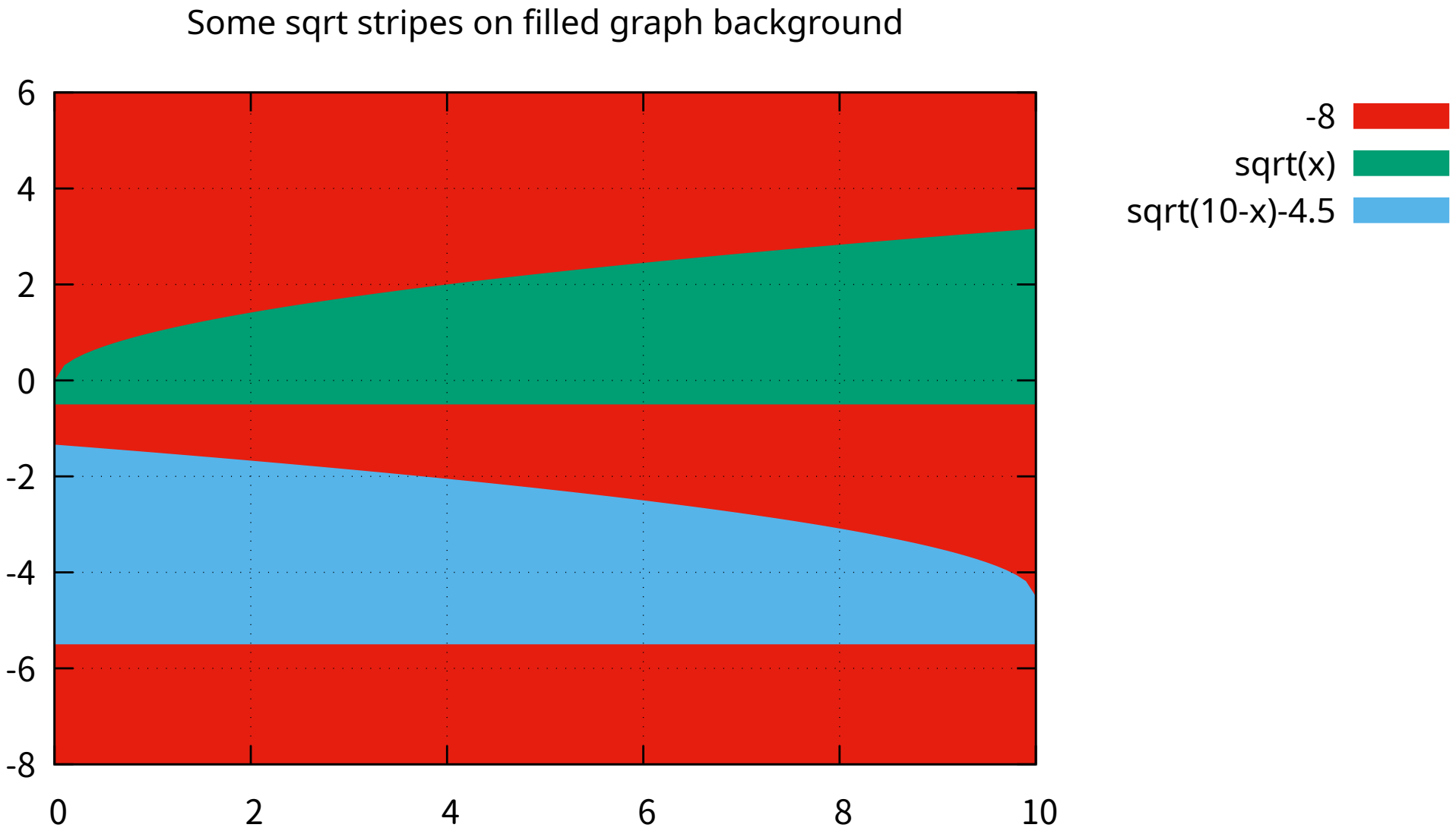



plot with filledcurve [options]
3
2
1
0
-1
-2
-3
-4
-5
-10
-5
0
5
10
1.5+sin(x)/x
sin(x)/x
1+sin(x)/x
-1+sin(x)/x
-2.5+sin(x)/x
-4.3+sin(x)/x
(x>3.5 ? x/3-3 : 1/0)

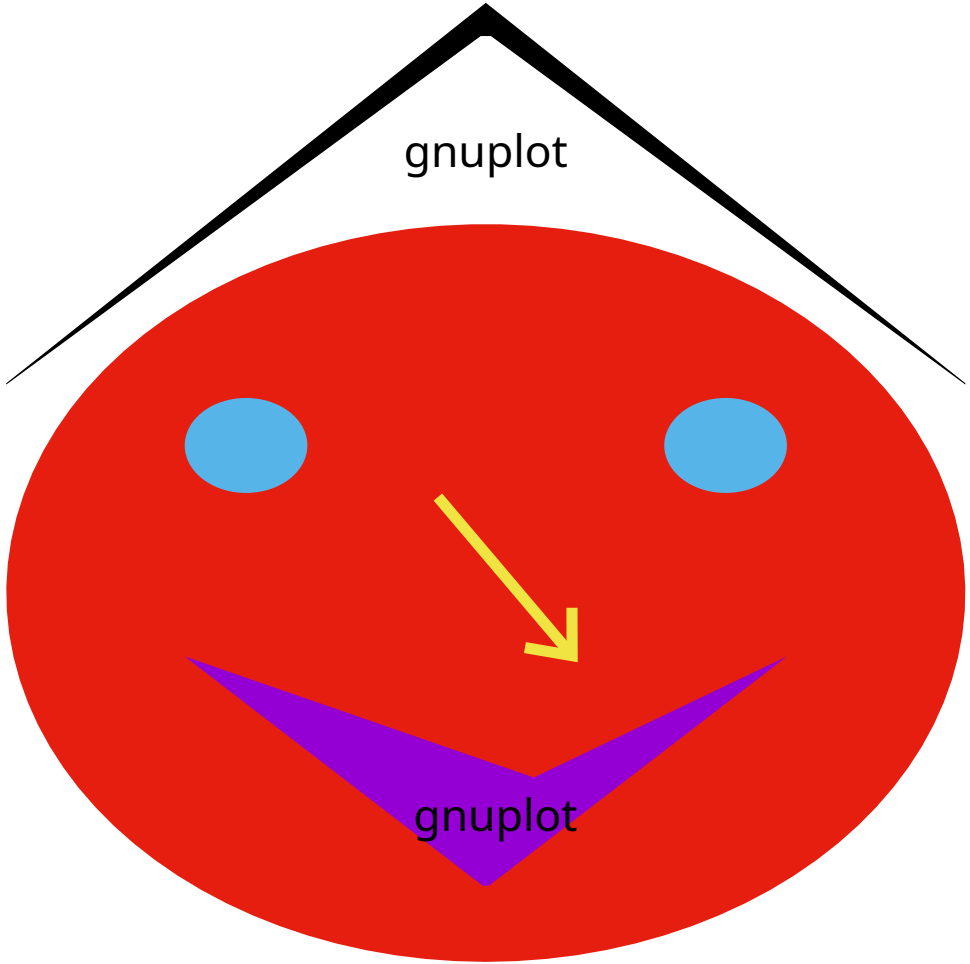

Intersection of two parabolas
100
80
60
40
20
0
-20
-40
-60
-10
-5
0
5
10
x*x
50-x*x
x*x


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

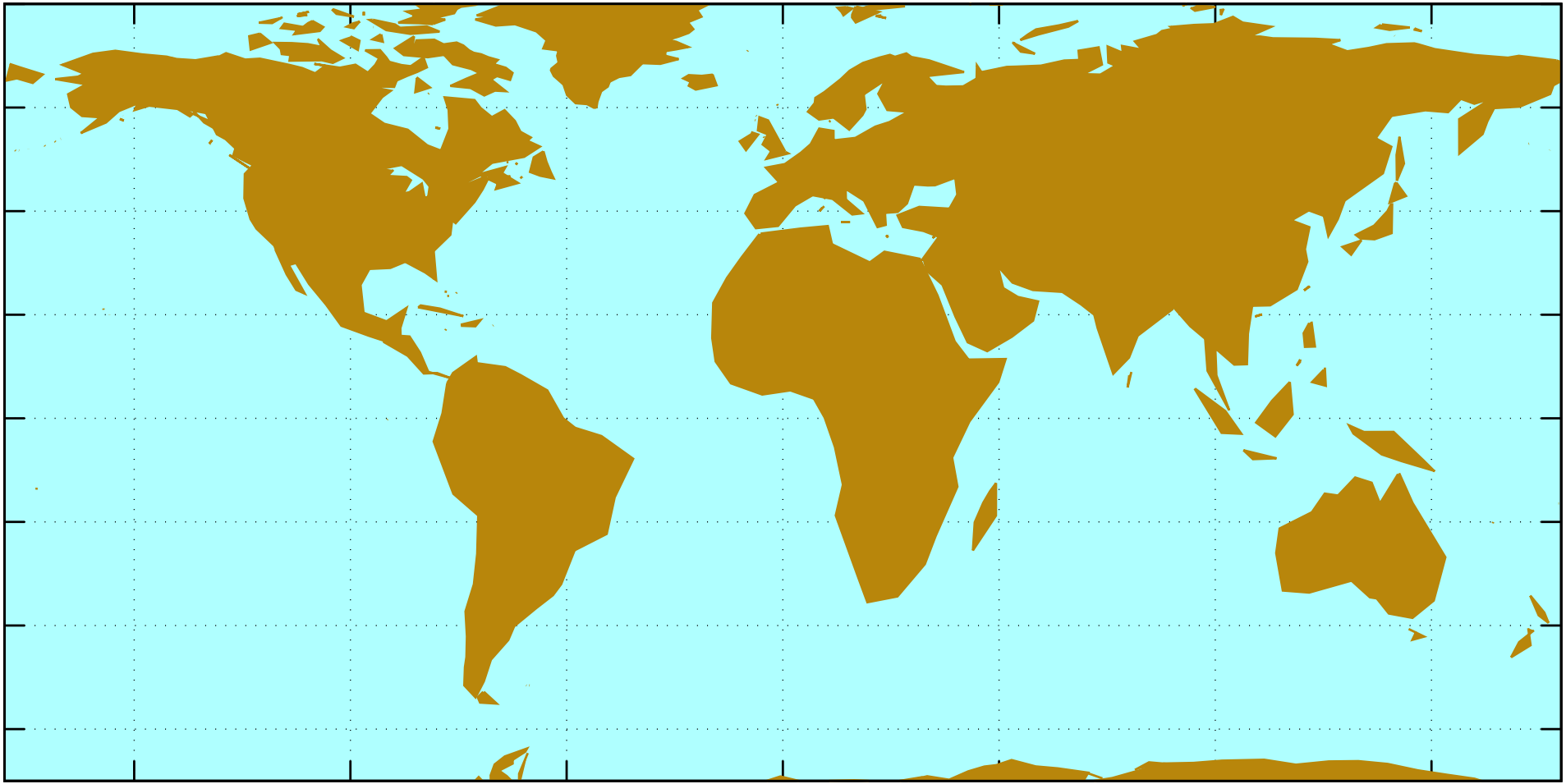

The red bat: abs(x) with filledcurve xy=2,5
abs(x)
10
9
8
7
6
5
4
3
2
1
0
-10
-5
0
5
10

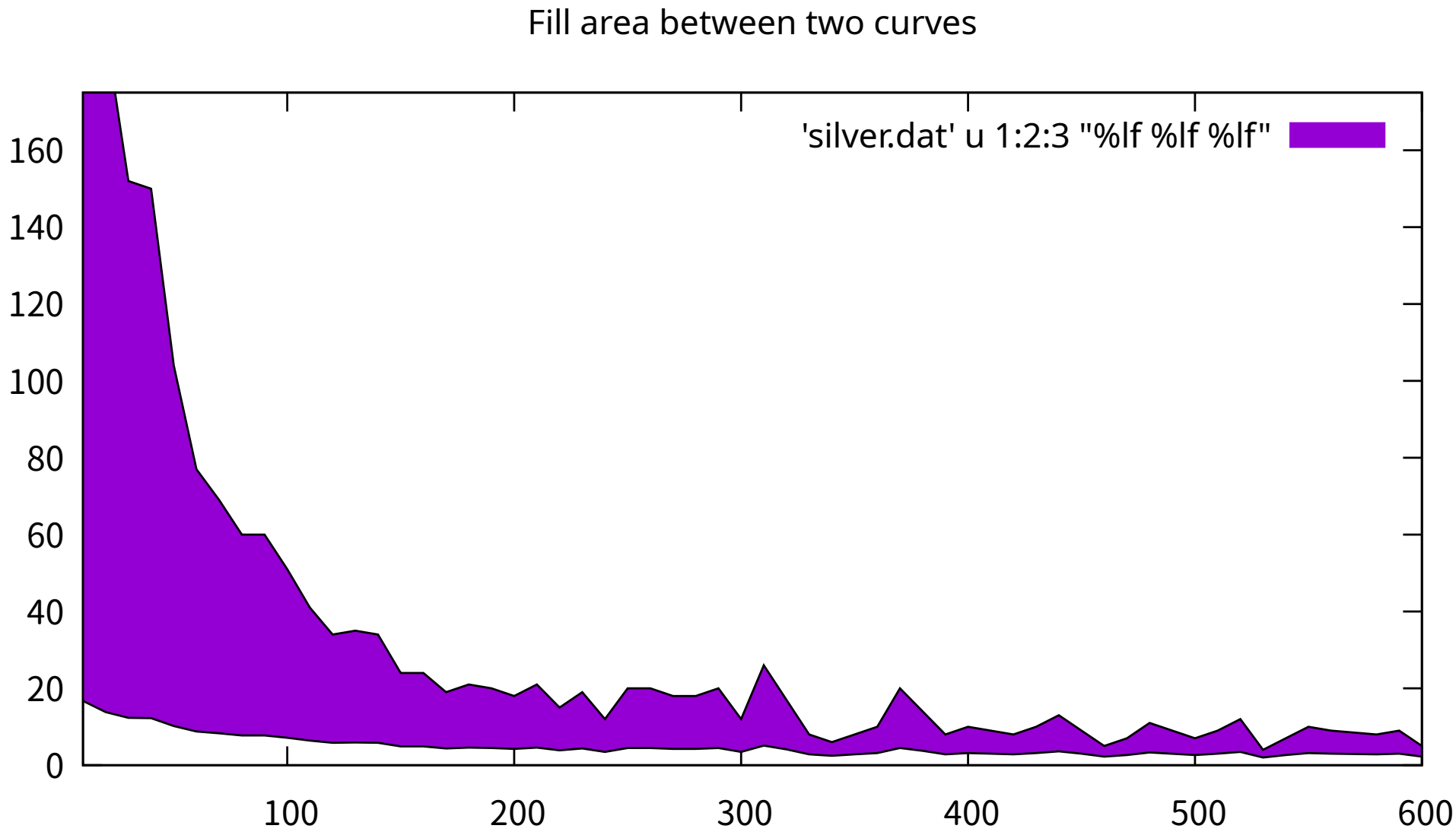

Some sqrt stripes on filled graph background
6
4
2
0
-2
-4
-6
-8
0
2
4
6
8
10
-8
sqrt(x)
sqrt(10-x)-4.5

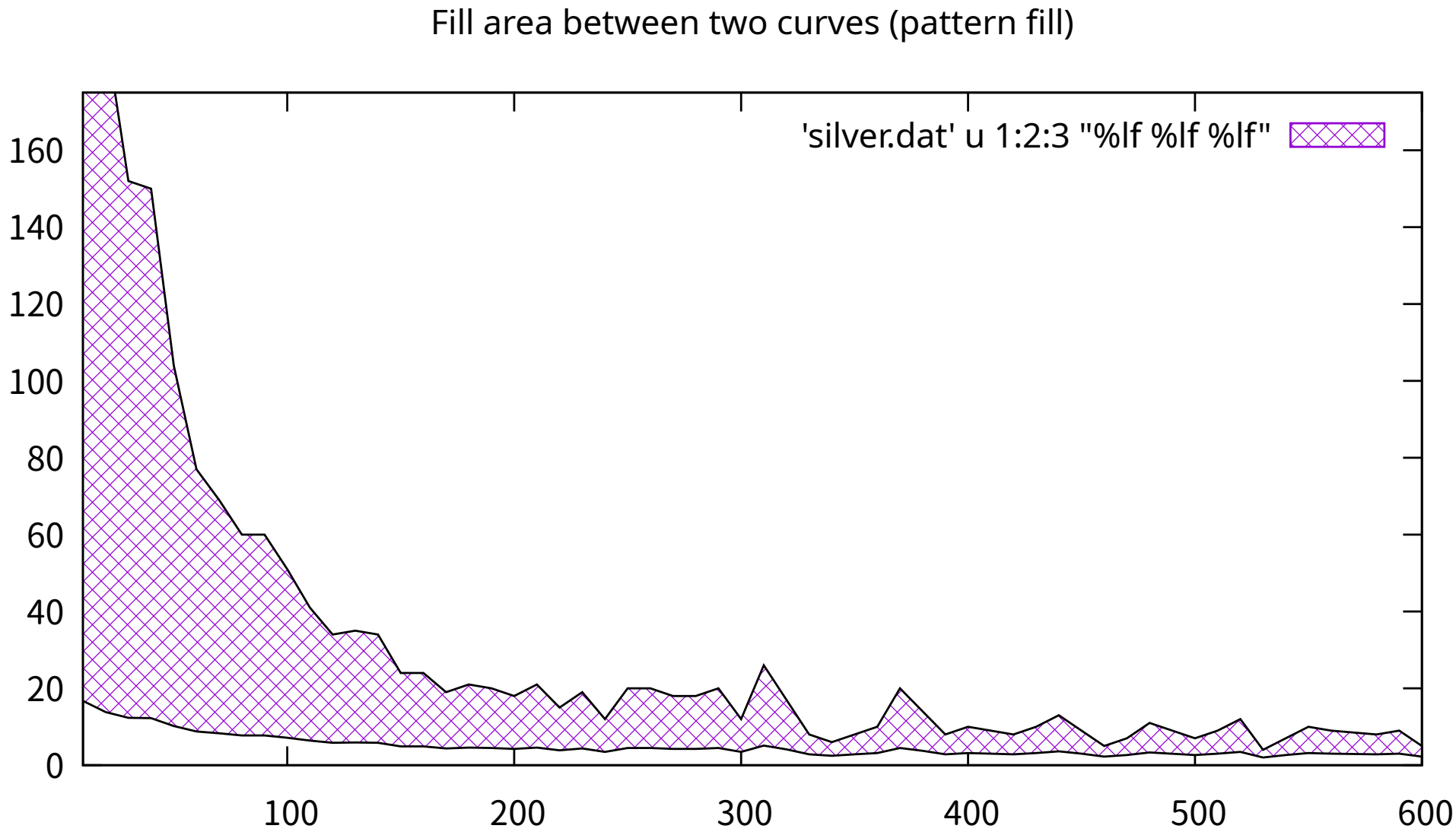
Let's smile with parametric filled curves

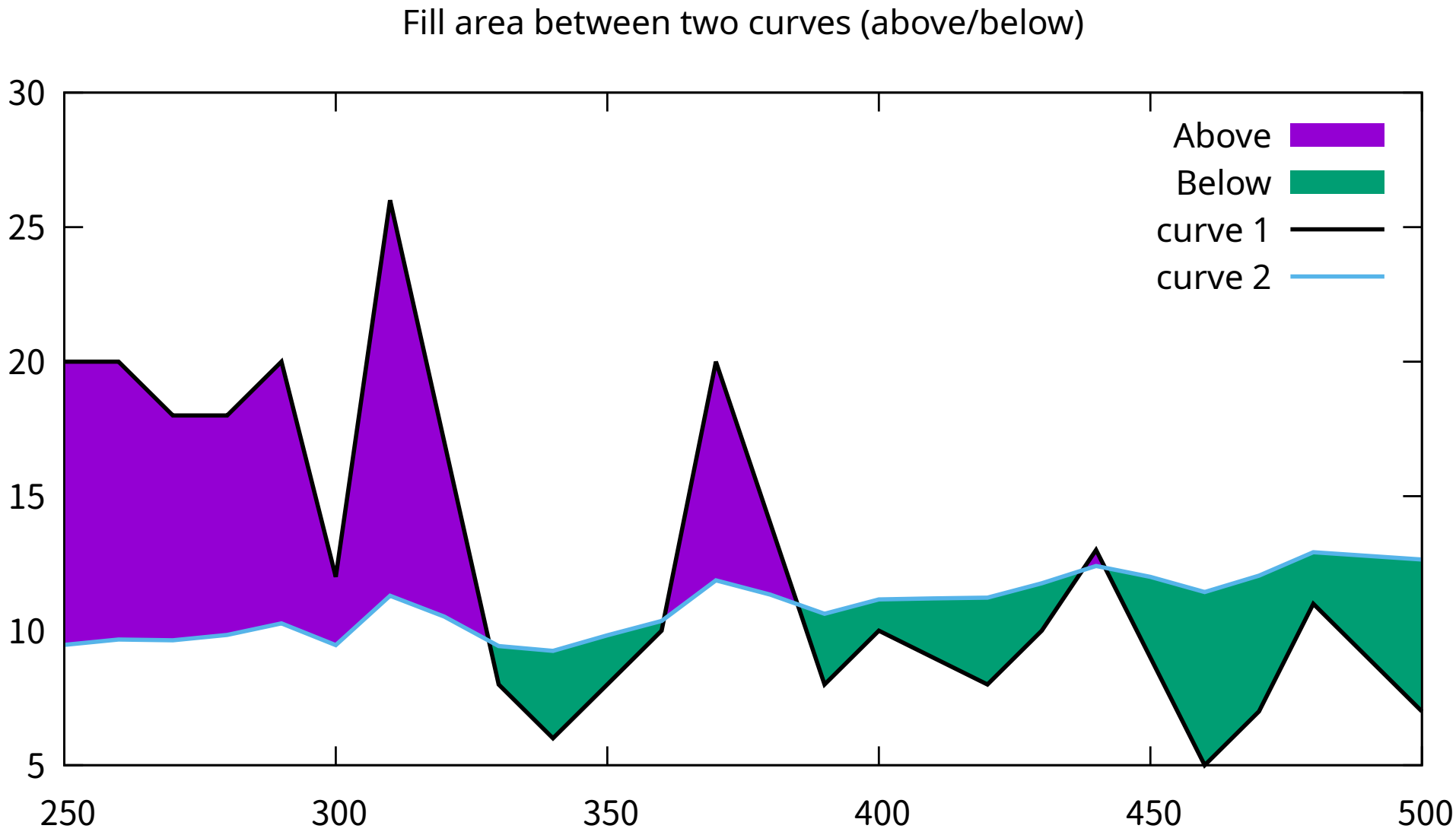


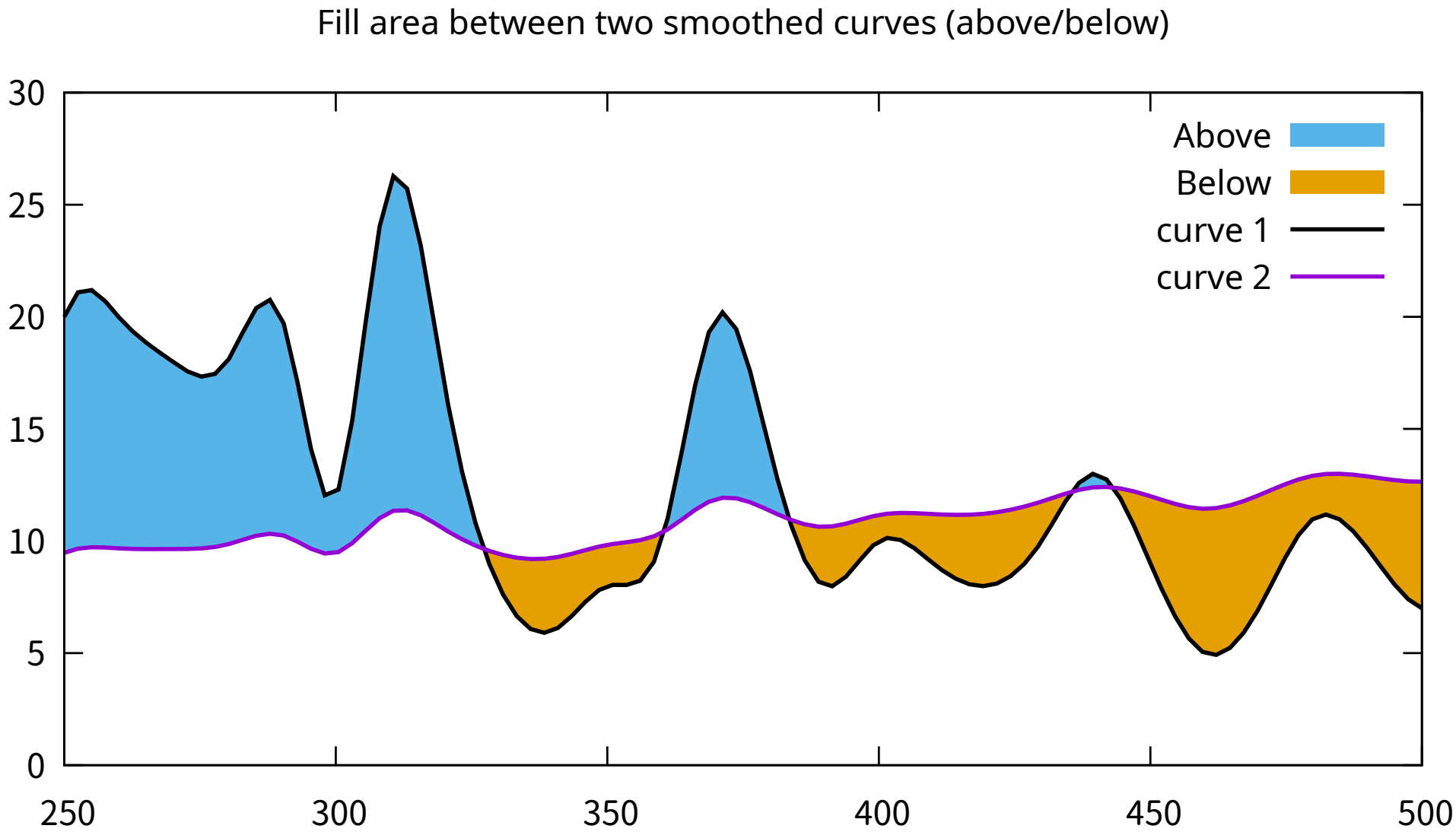
world.dat plotted with filledcurves

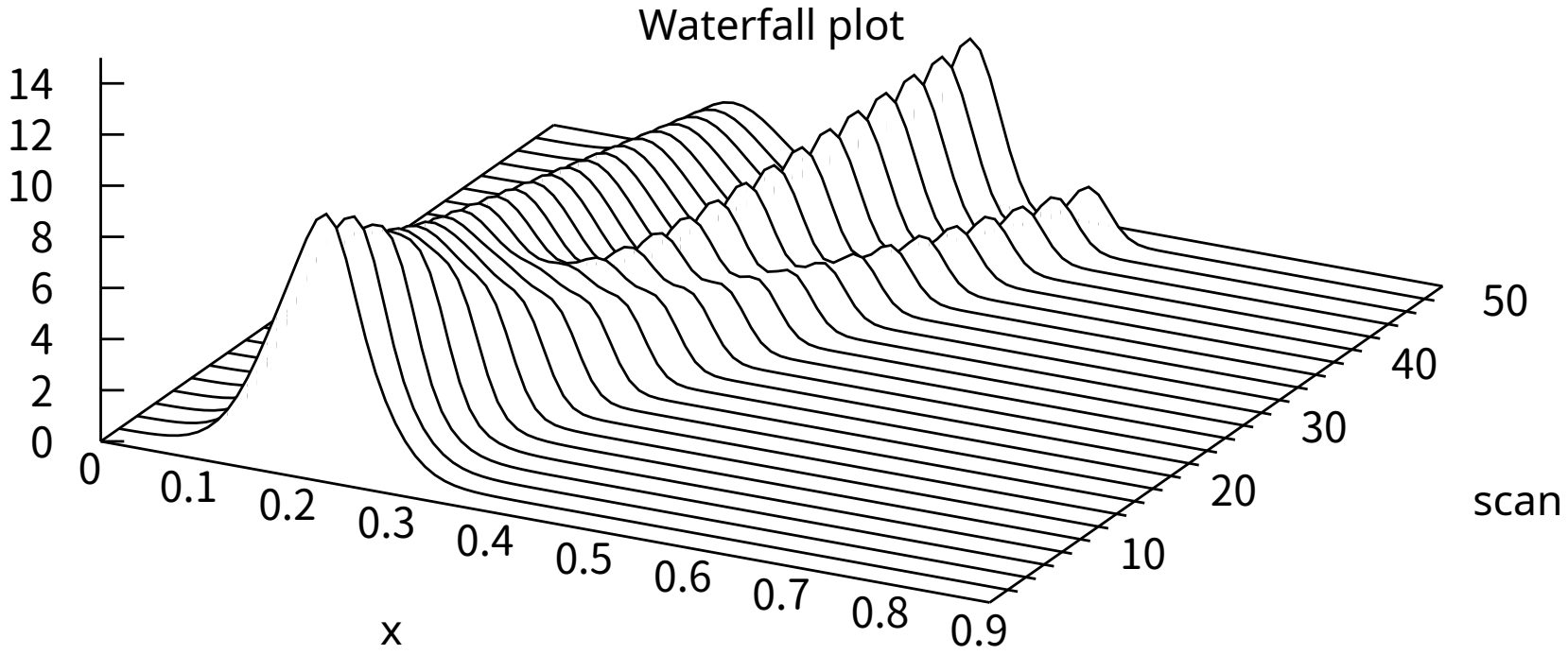


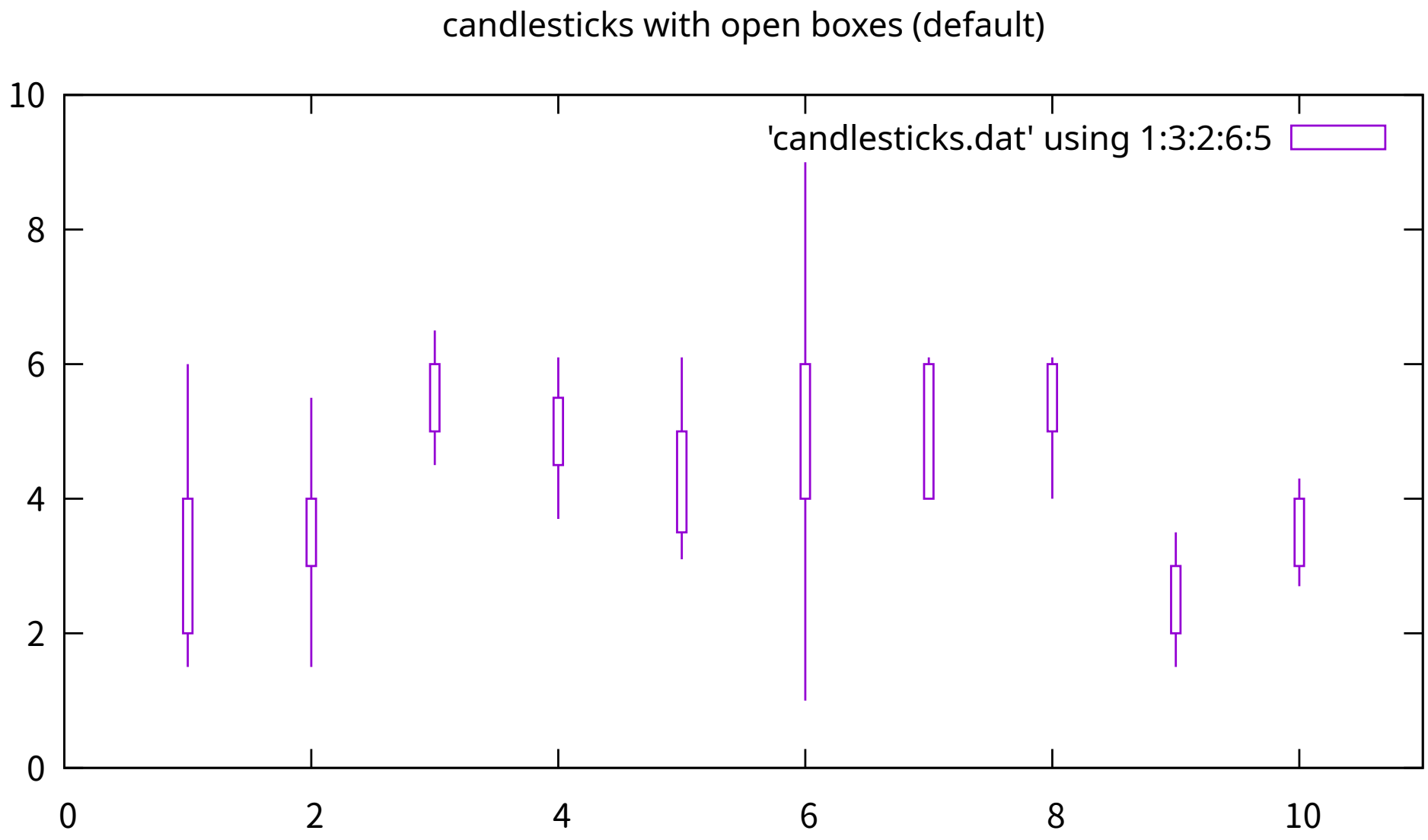

Fill area between two curves
'silver.dat' u 1:2:3 "%lf %lf %lf"
160
140
120
100
80
60
40
20
0
100
200
300
400
500
600

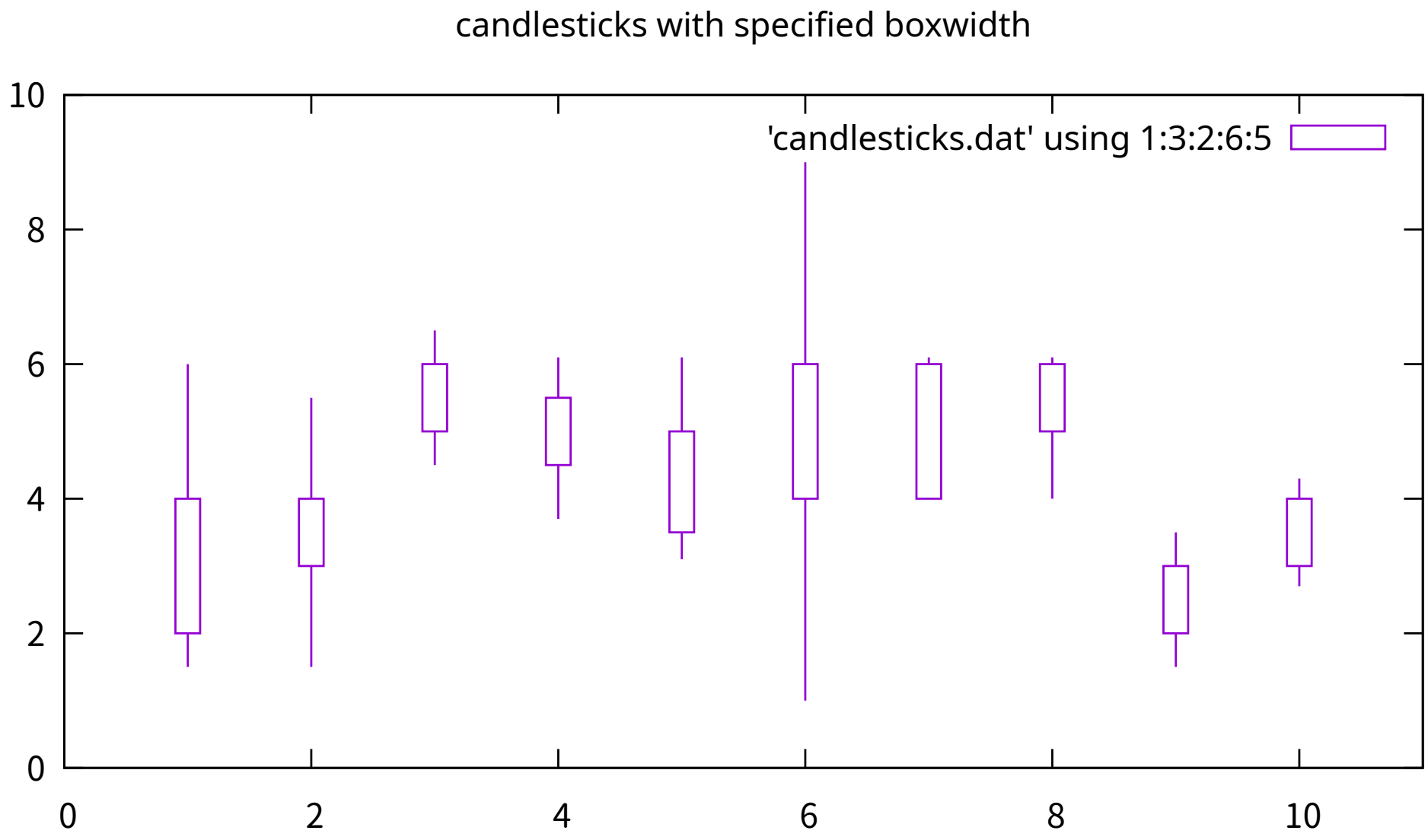

Fill area between two curves (pattern fill)
'silver.dat' u 1:2:3 "%lf %lf %lf"
160
140
120
100
80
60
40
20
0
100
200
300
400
500
600

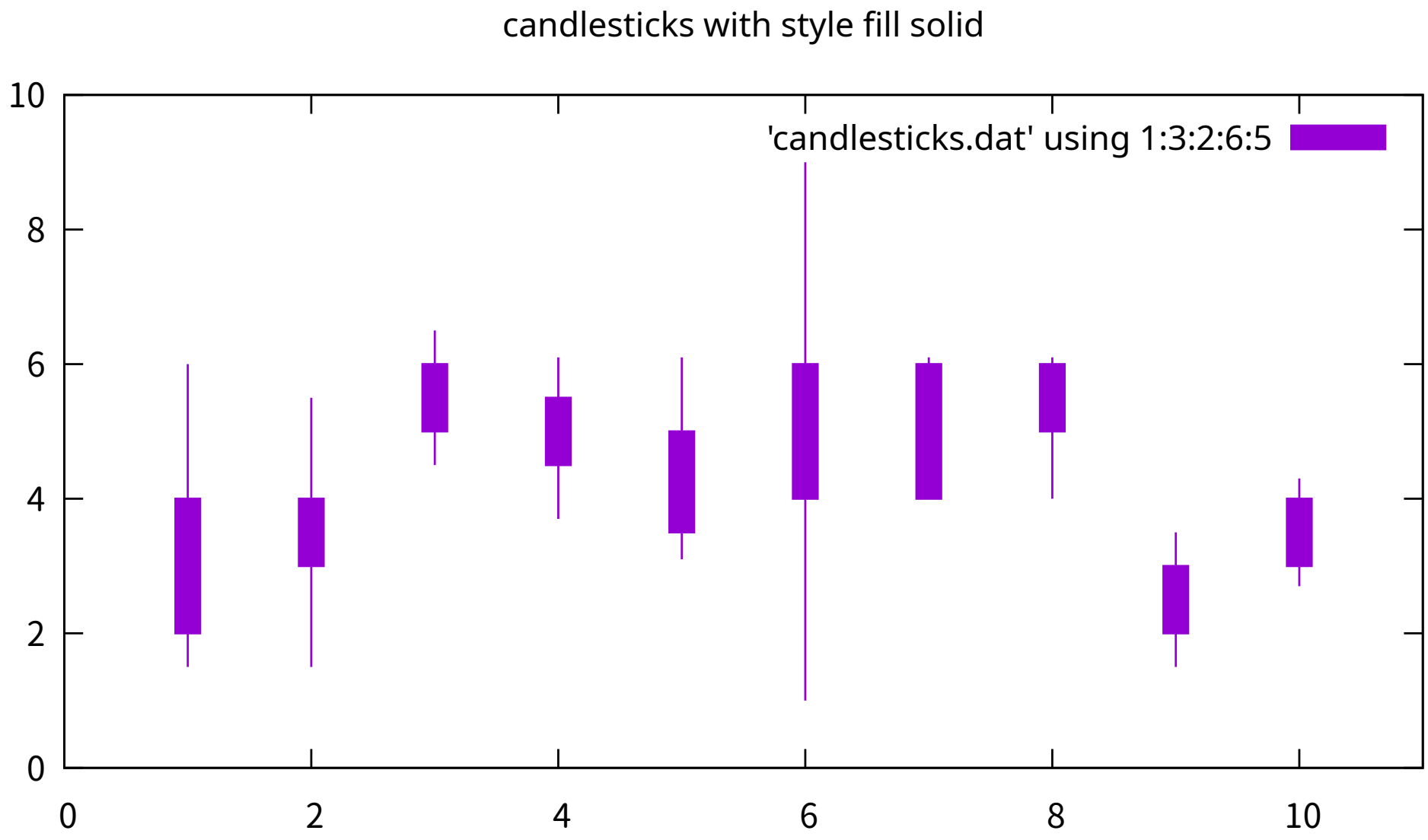

Fill area between two curves (above/below)
Above
Below
curve 1
curve 2
30
25
20
15
10
5
250
300
350
400
450
500

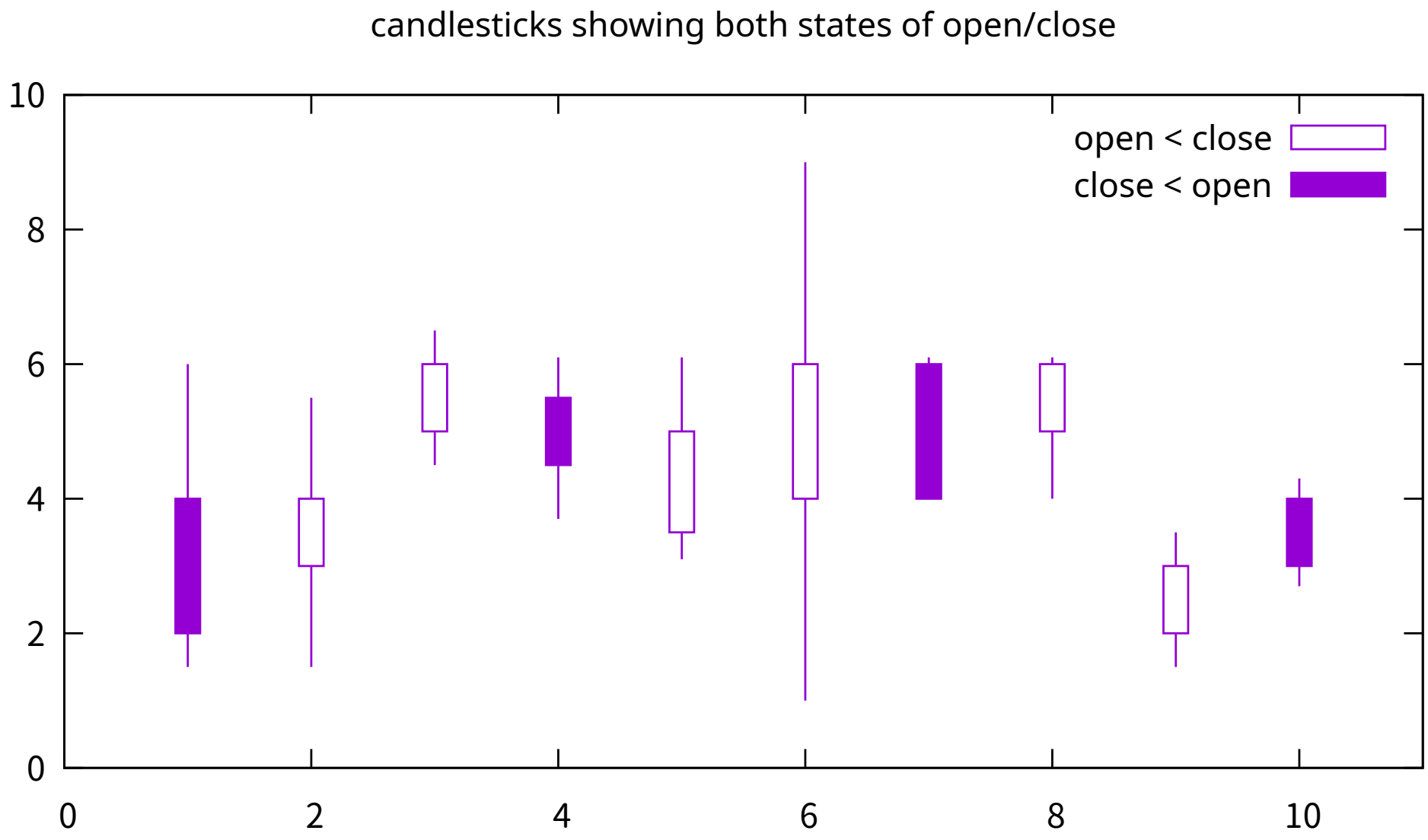

Fill area between two smoothed curves (above/below)
Above
Below
curve 1
curve 2
30
25
20
15
10
5
0
250
300
350
400
450
500


Waterfall plot
14
12
10
8
6
4
2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
x
50
40
30
20
10
scan

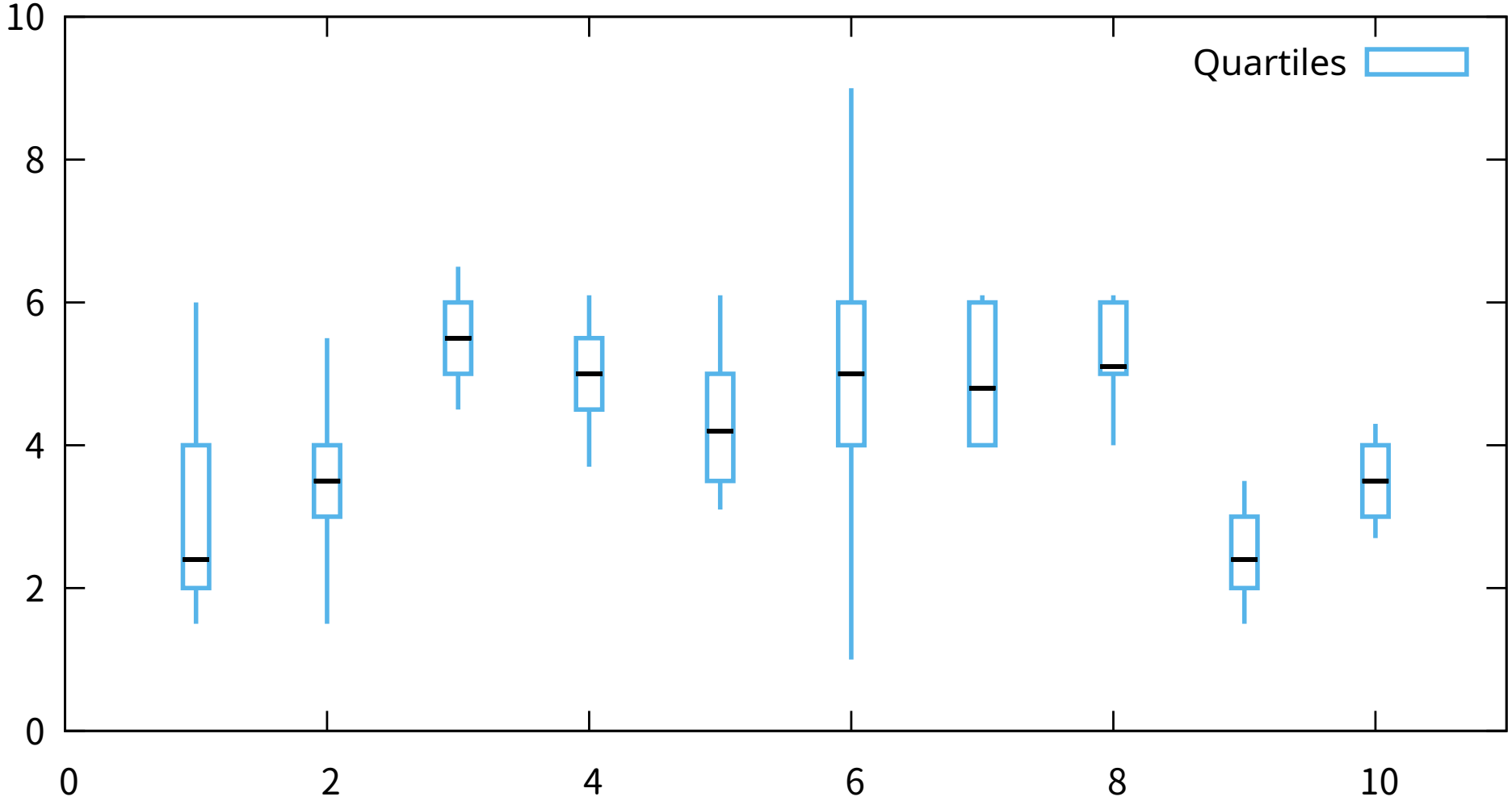

candlesticks with open boxes (default)
'candlesticks.dat' using 1:3:2:6:5
10
8
6
4
2
0
0
2
4
6
8
10


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

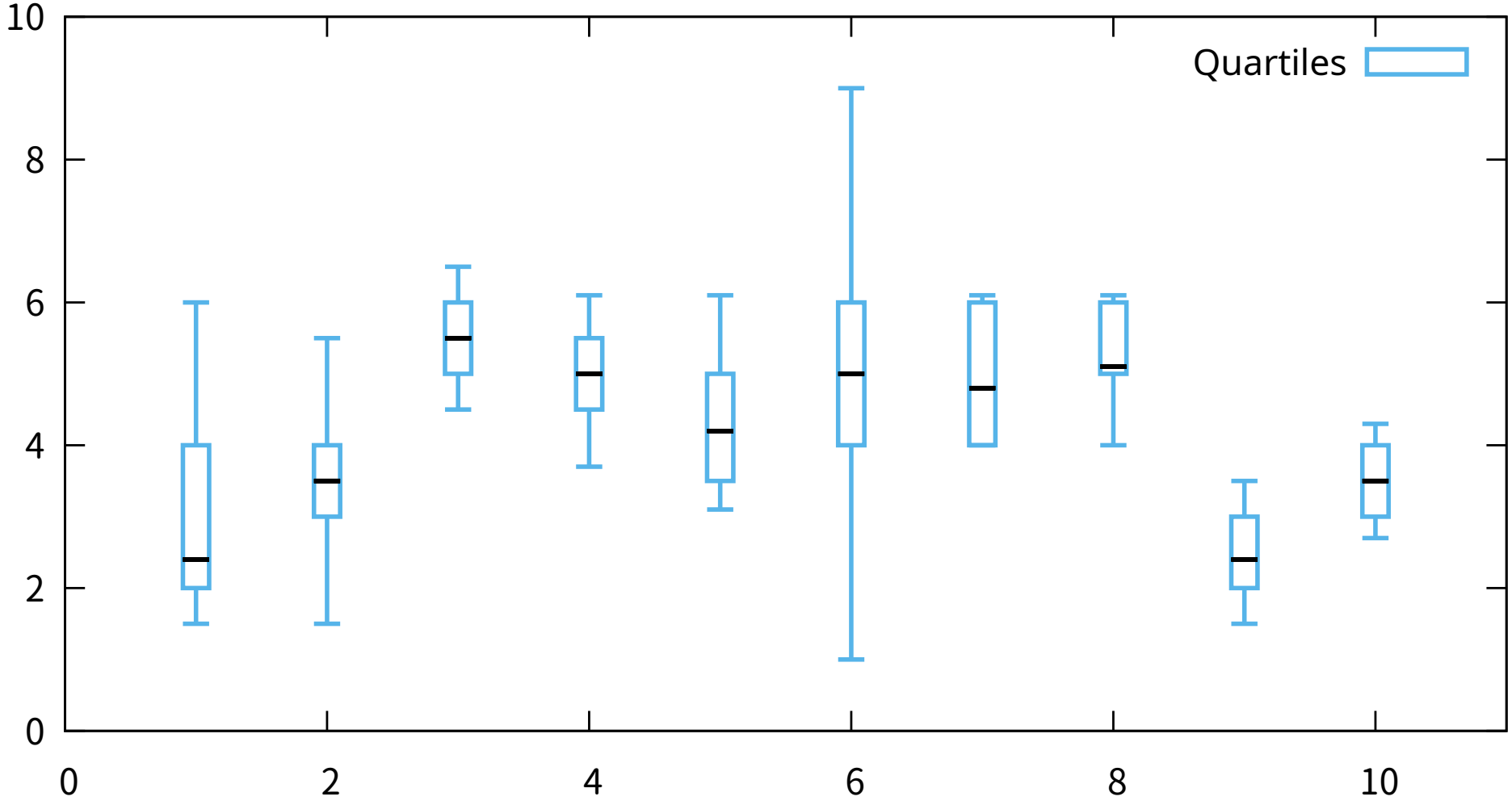

candlesticks with style fill solid
'candlesticks.dat' using 1:3:2:6:5
10
8
6
4
2
0
0
2
4
6
8
10

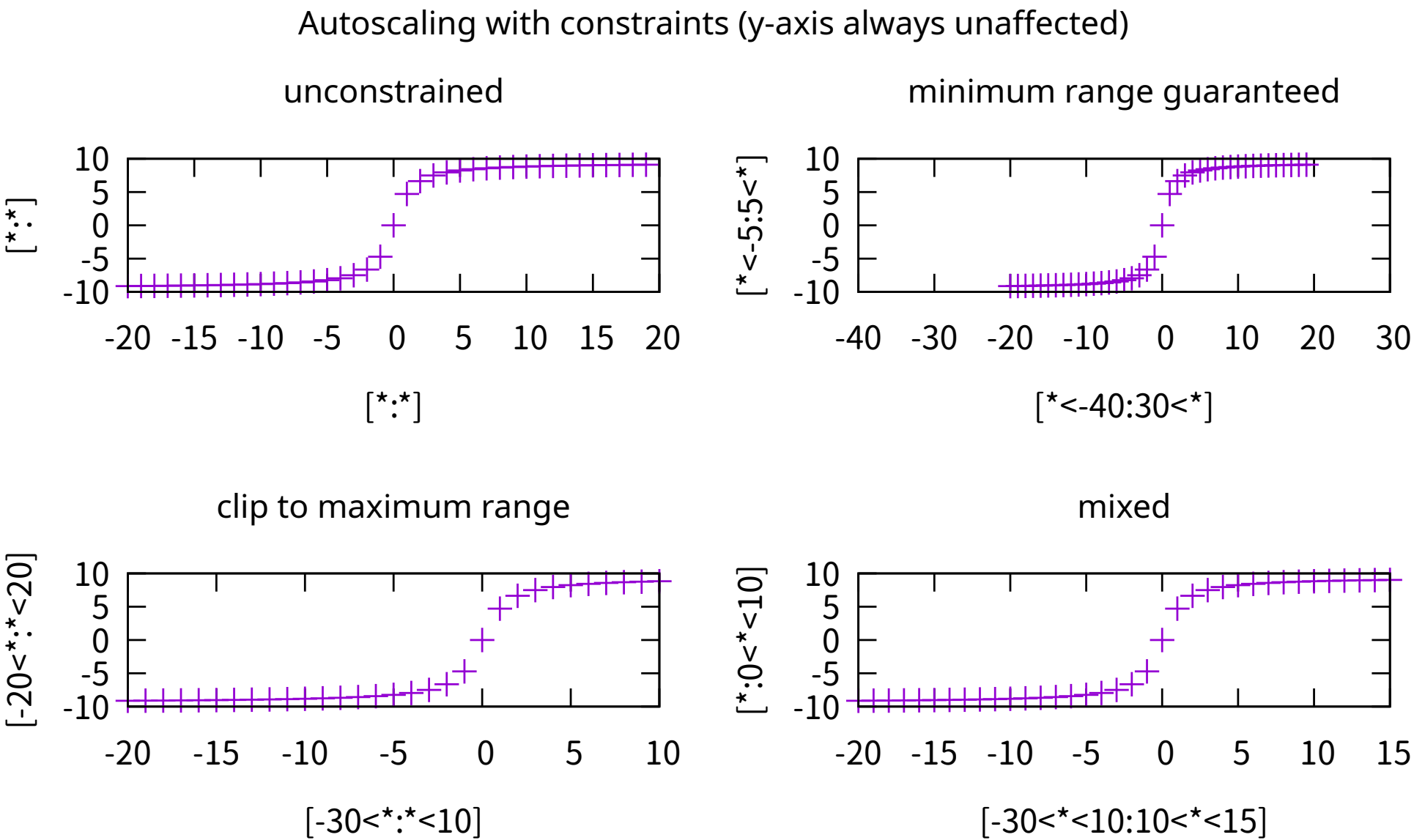

candlesticks showing both states of open/close
open < close
close < open
10
8
6
4
2
0
0
2
4
6
8
10

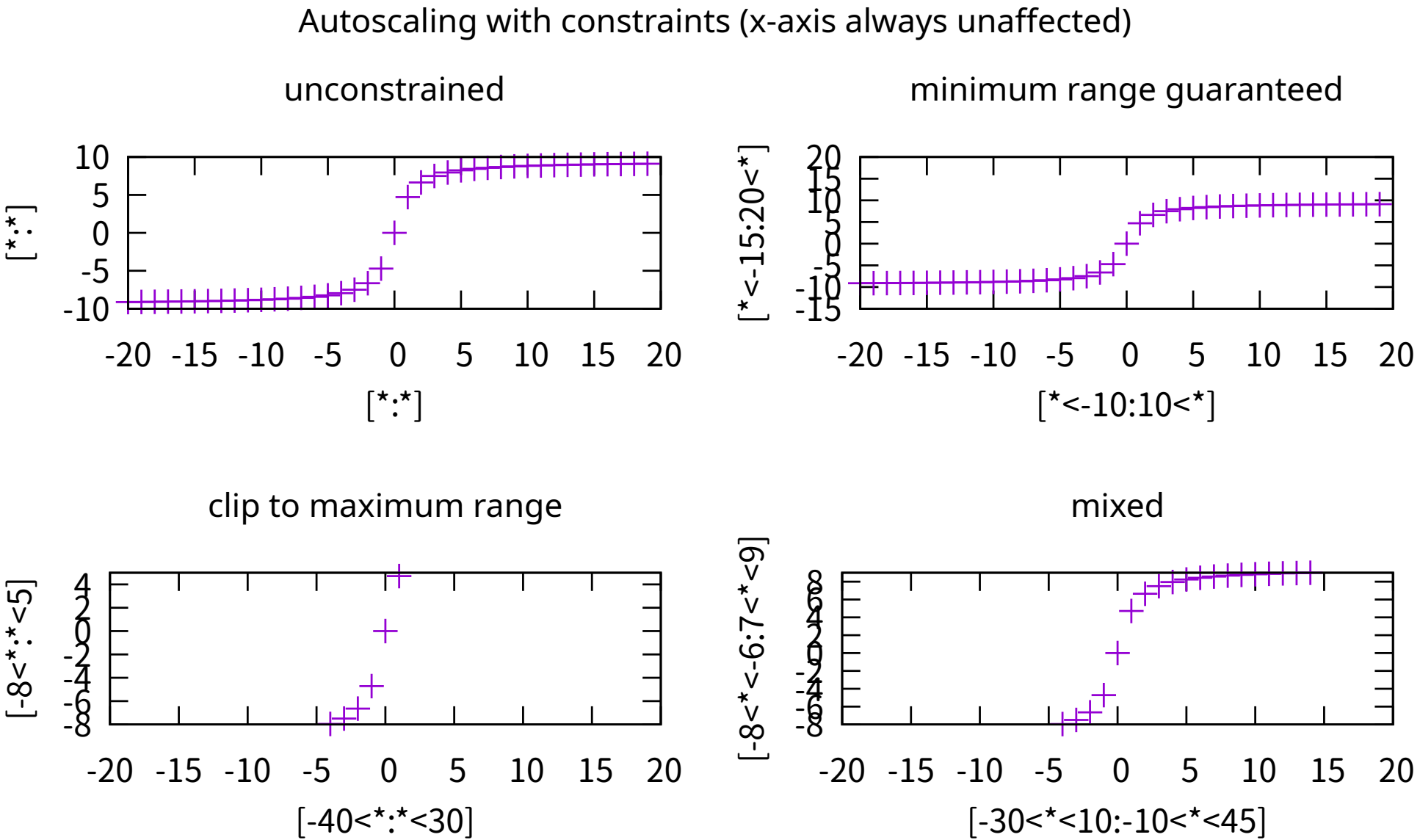
box-and-whisker plot adding median value as bar



box-and-whisker with median bar and whiskerbars

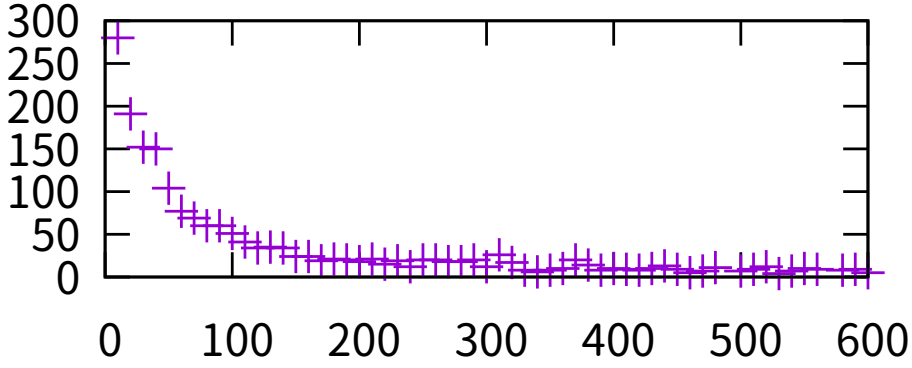



Autoscaling with constraints (y-axis always unaffected)
unconstrained
[*:*]
[*:*]
minimum range guaranteed
[*<-5:5<*]
[*<-40:30<*]
clip to maximum range
[-20<*:*<20]
[-30<*:*<10]
mixed
[*:0<*<10]
[-30<*<10:10<*<15]

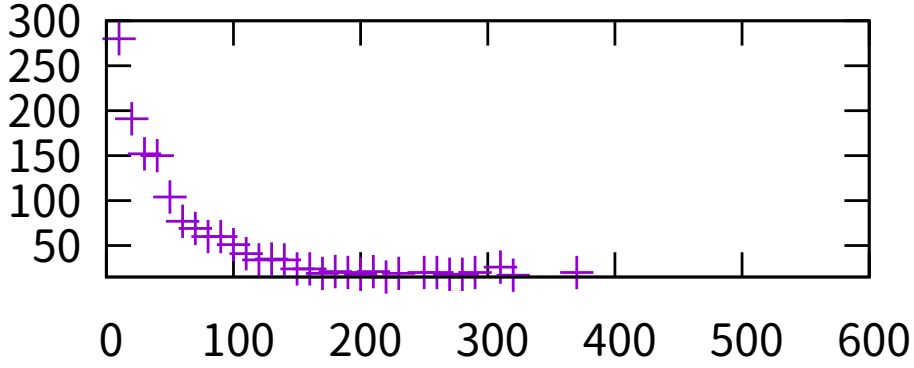

Autoscaling with constraints (x-axis always unaffected)
unconstrained
[*:*]
[*:*]
minimum range guaranteed
[*<-15:20<*]
[*<-10:10<*]
clip to maximum range
[-8<*:*<5]
[-40<*:*<30]
mixed
[-8<*<-6:7<*<9]
[-30<*<10:-10<*<45]

Autoscaling with constraints

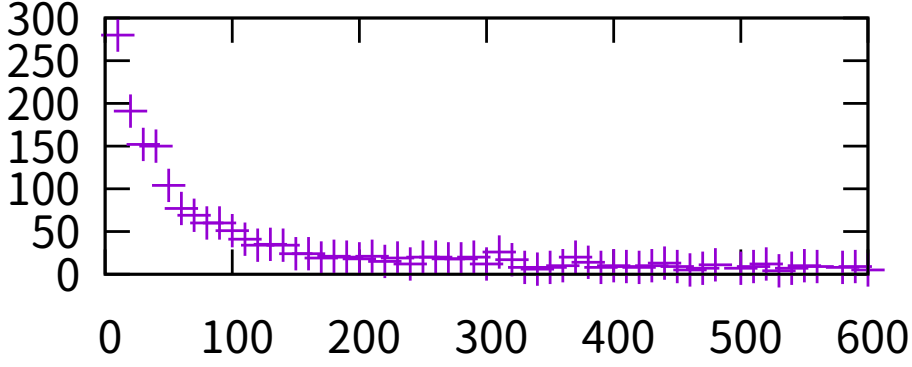
autoscale xy



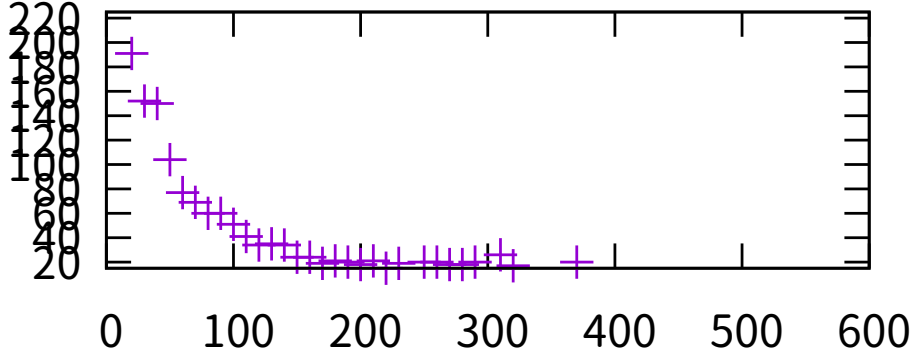
set yrange [15<*<25:*]

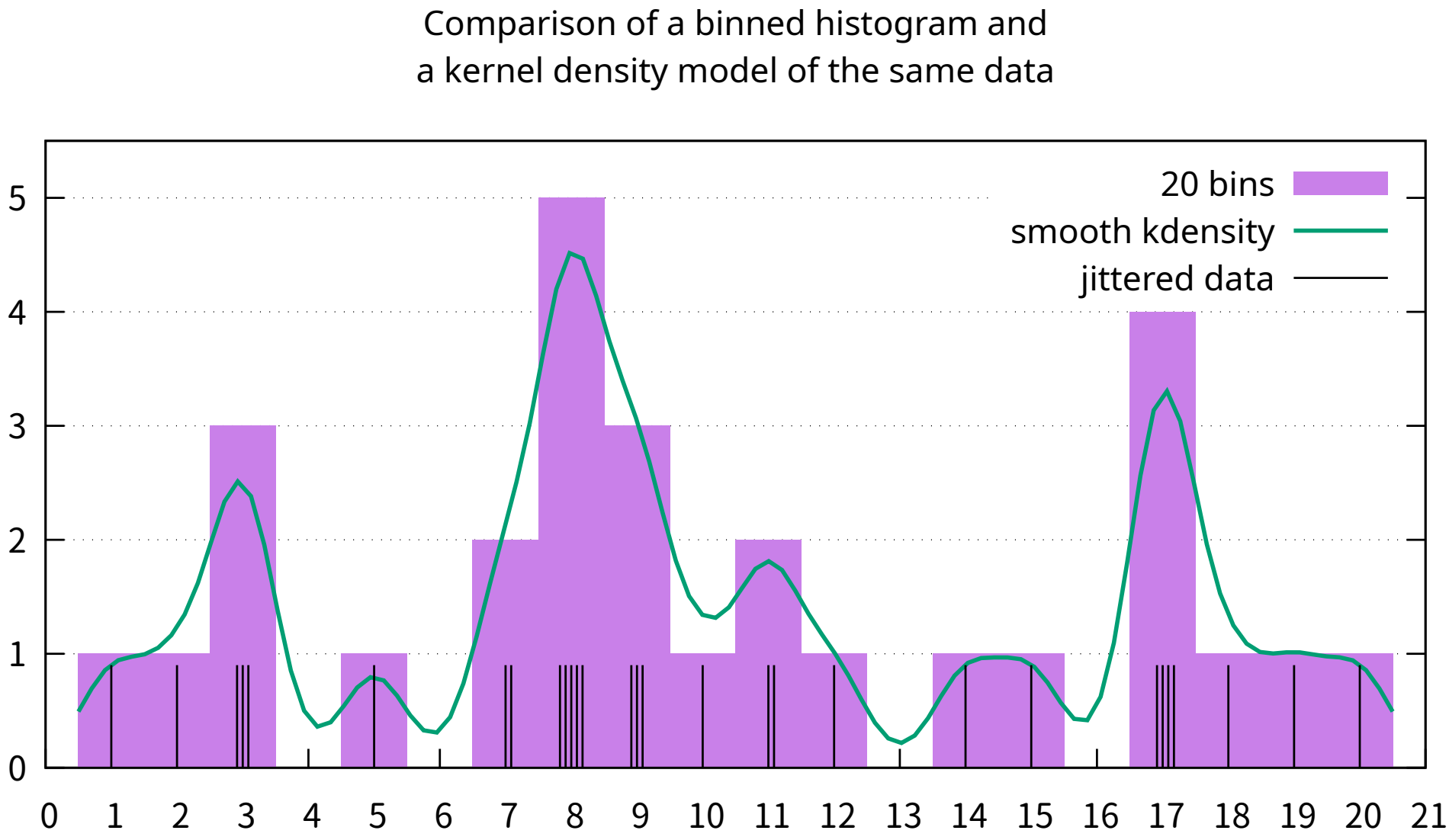


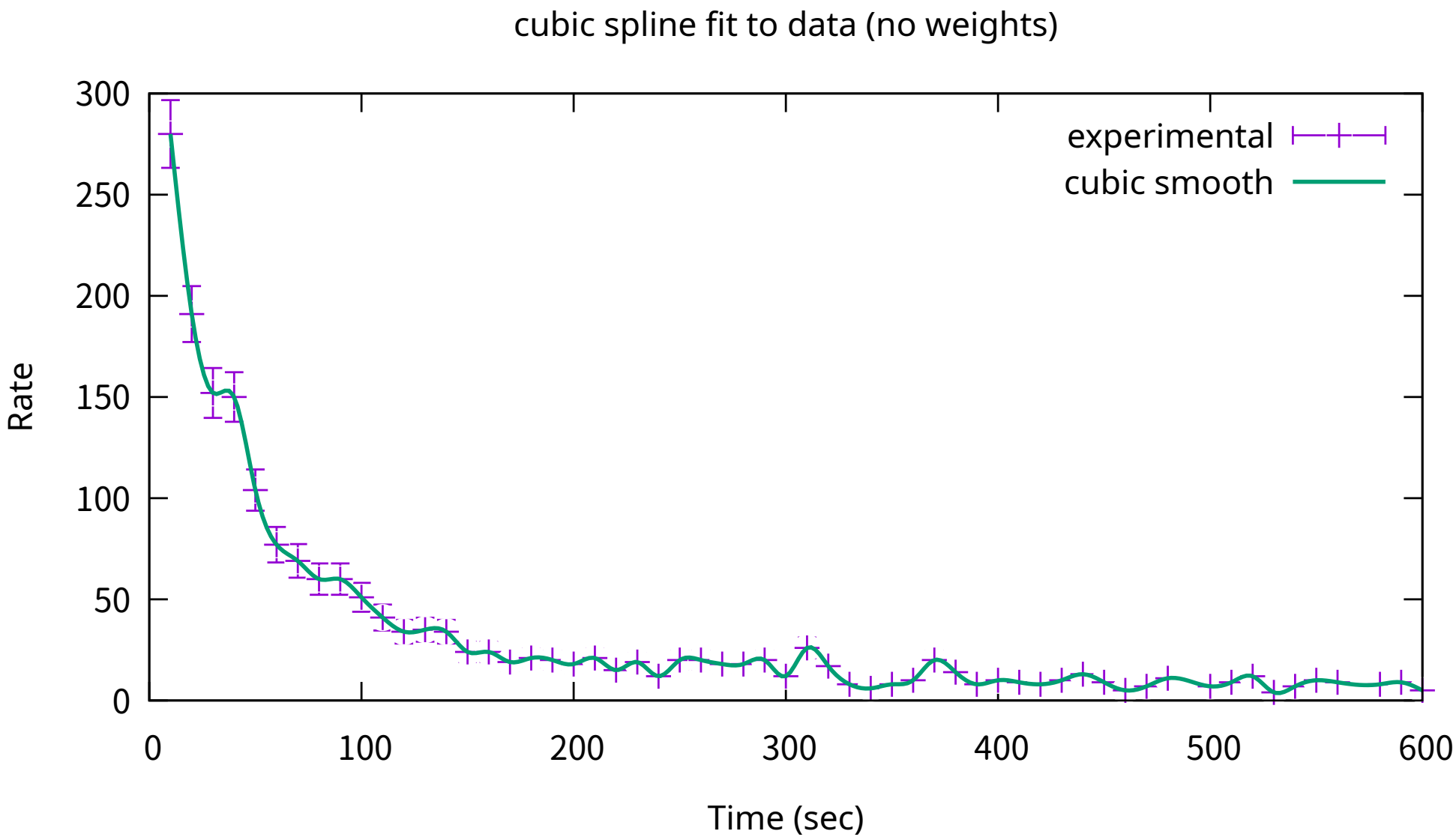
set autoscale ymin

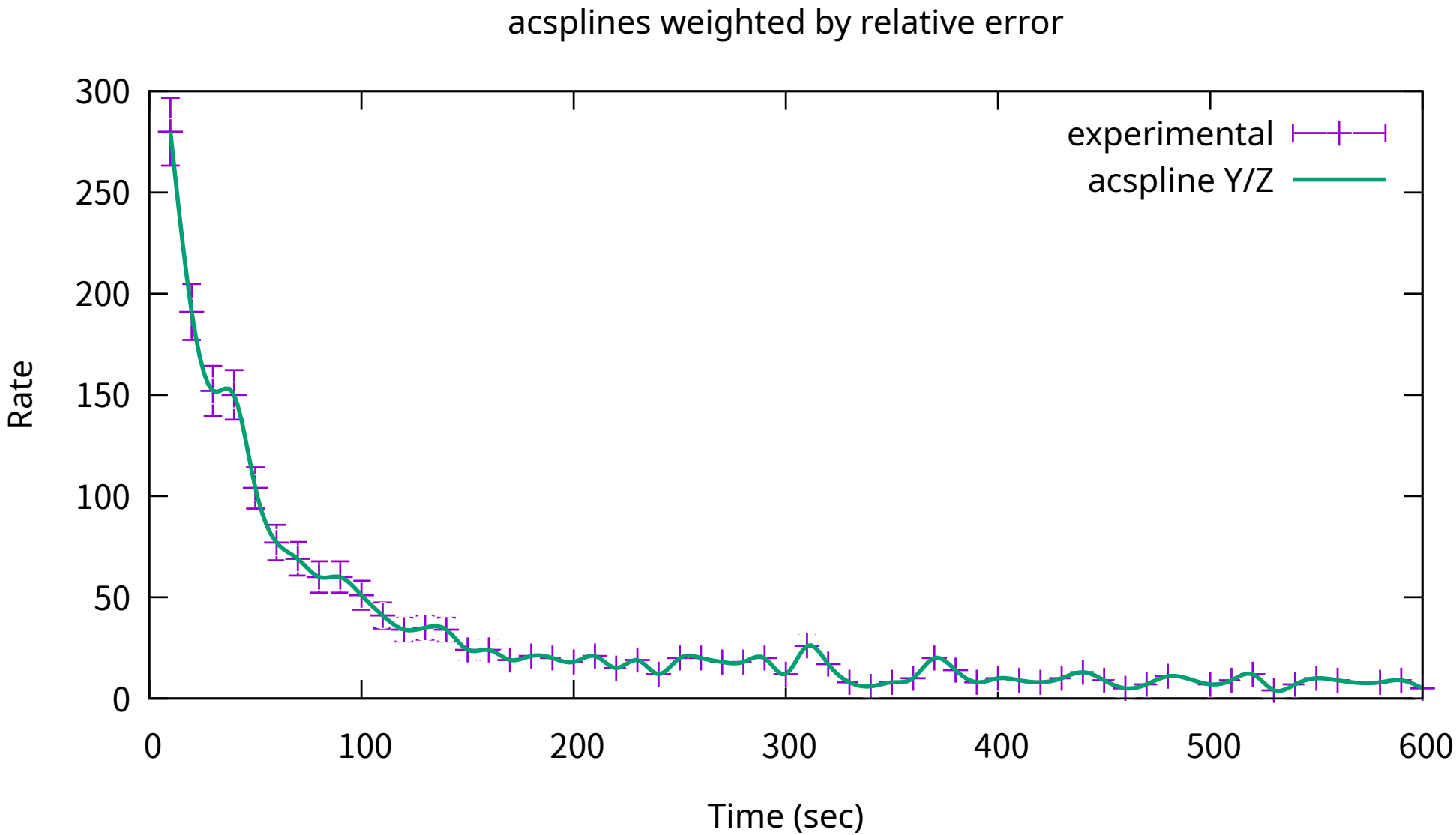


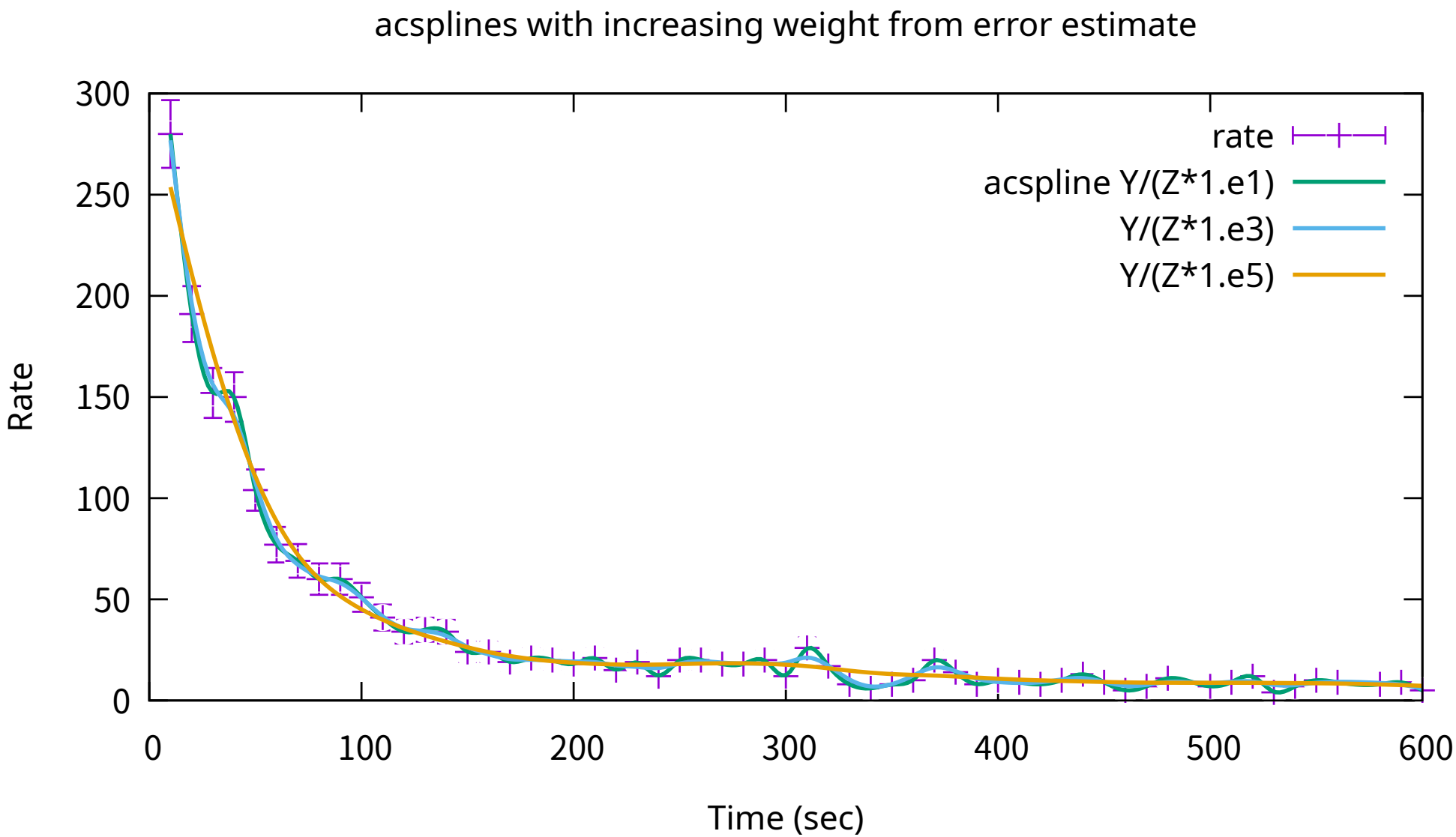
set yrange [15<*<25:135<*<225]

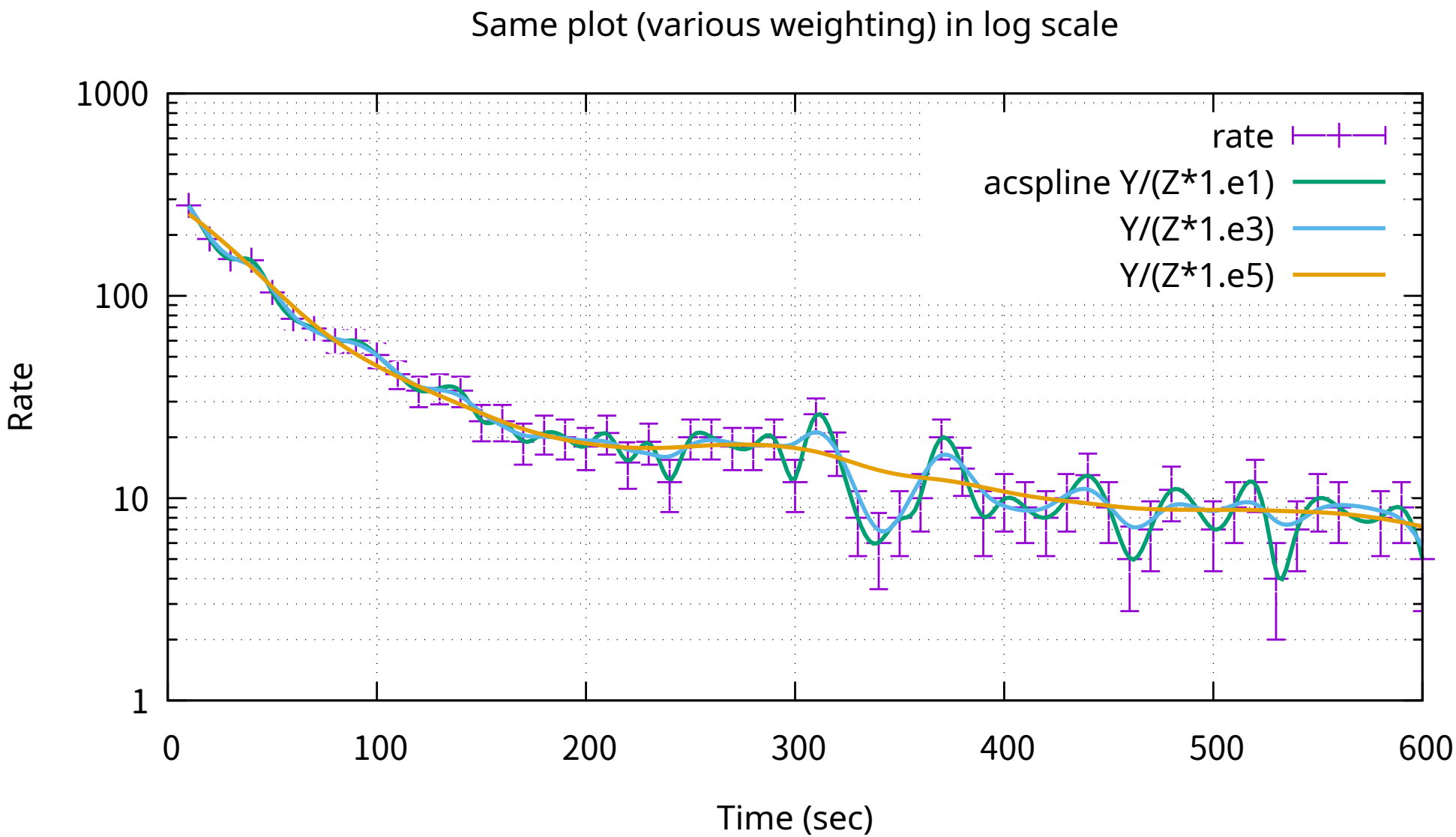


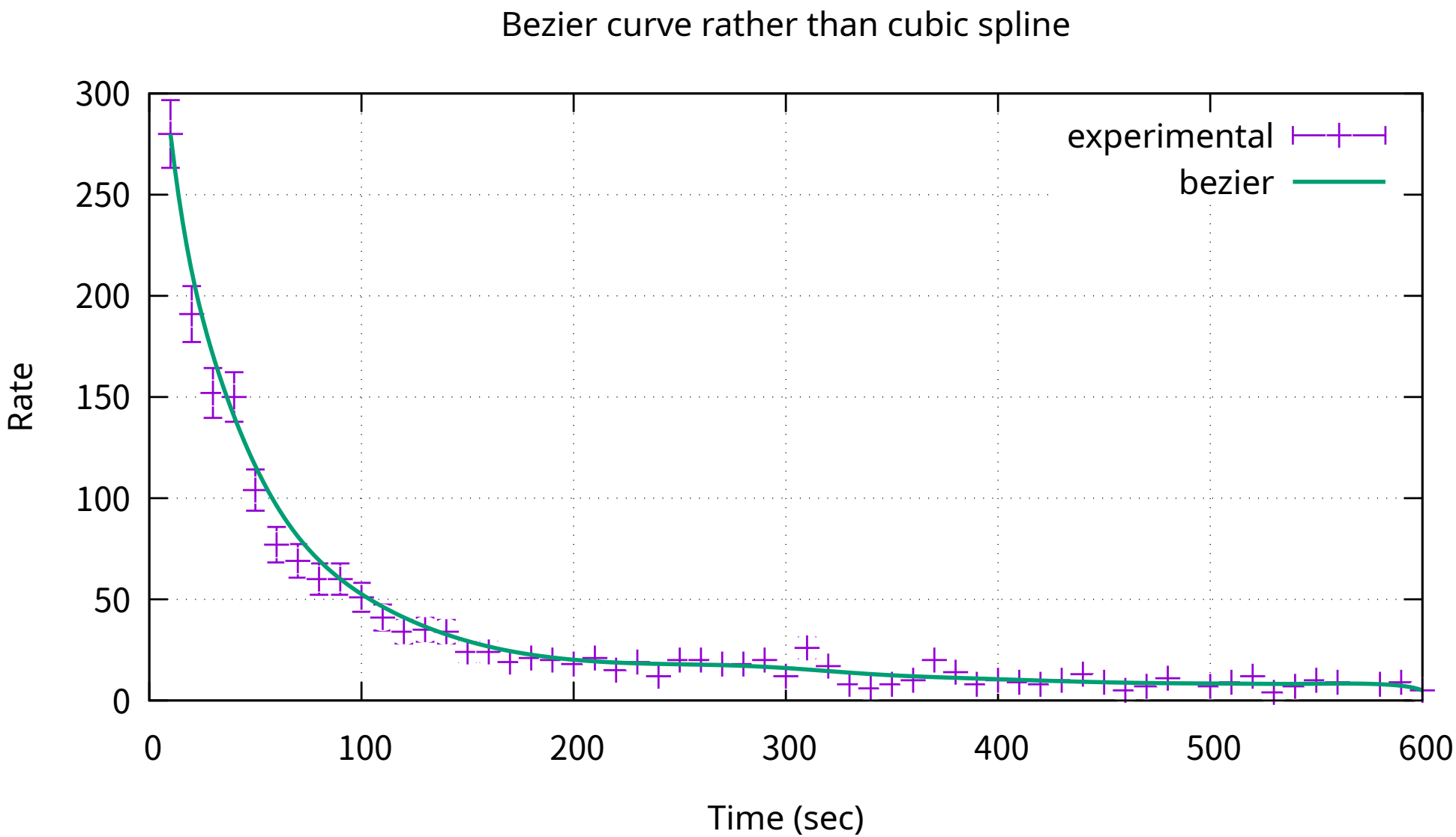

Comparison of a binned histogram and
a kernel density model of the same data
20 bins
smooth kdensity
jittered data
0
1
2
3
4
5
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

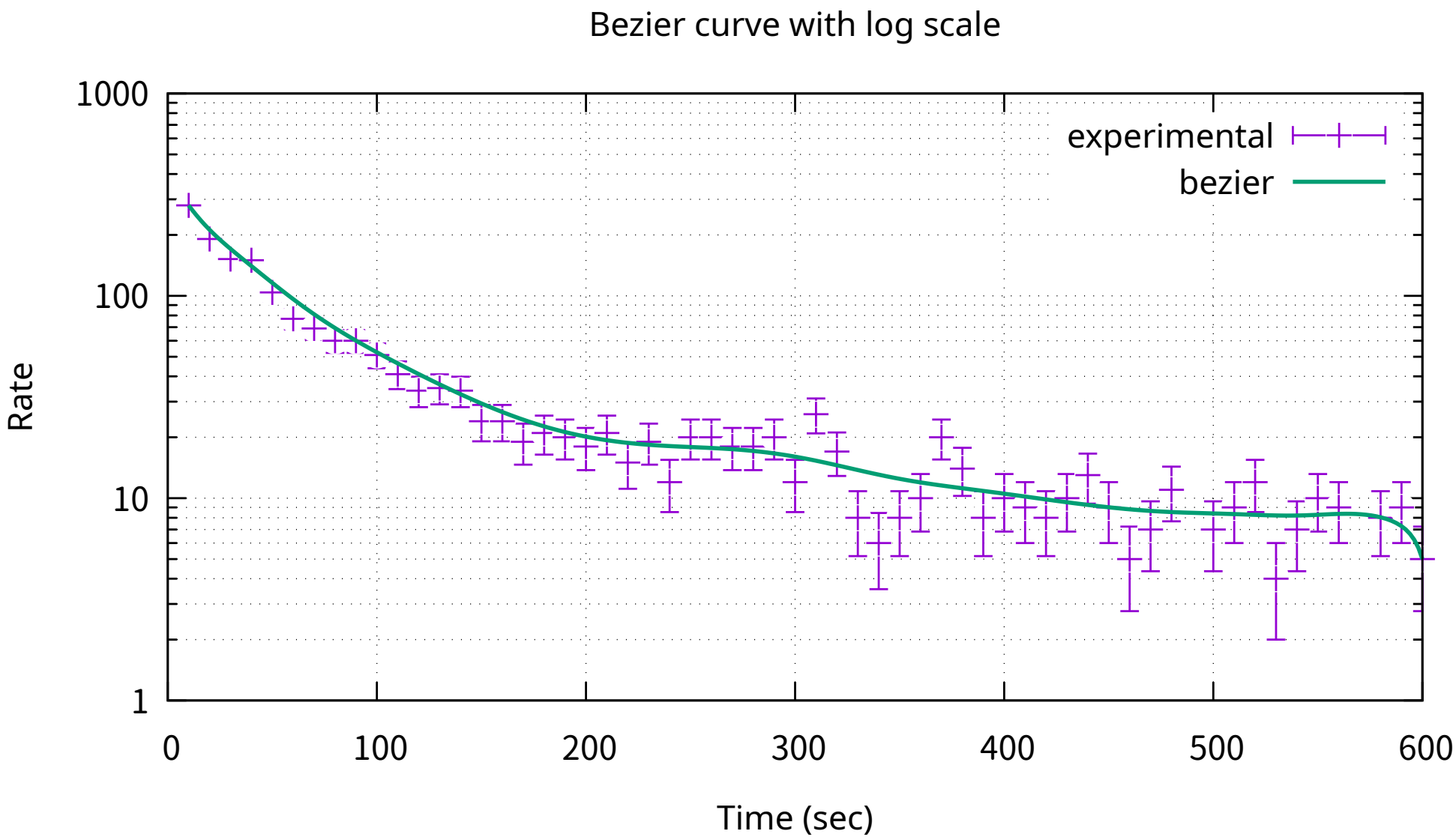

cubic spline fit to data (no weights)
experimental
cubic smooth
Rate
Time (sec)
0
50
100
150
200
250
300
0
100
200
300
400
500
600

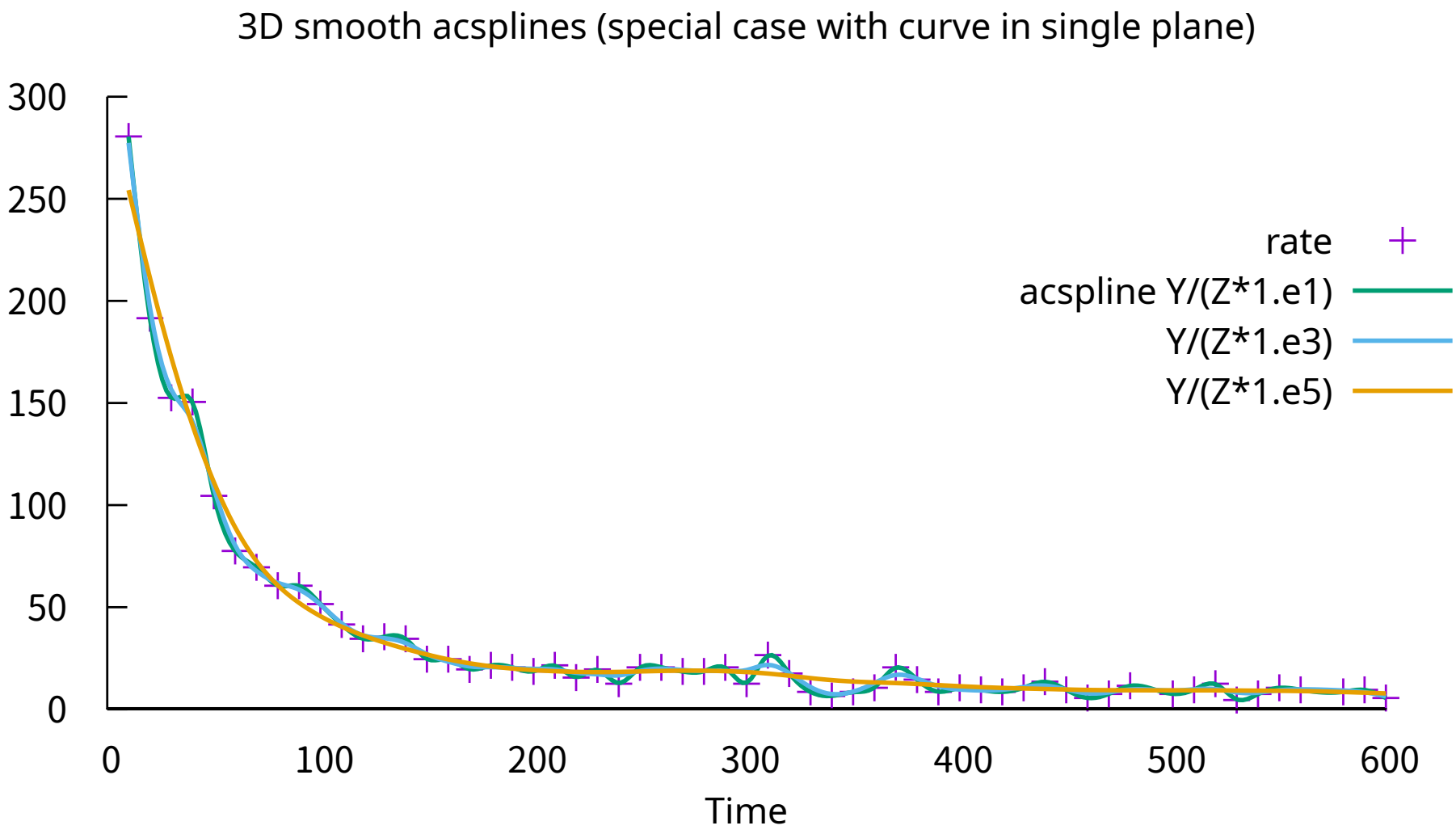

acsplines weighted by relative error
experimental
acspline Y/Z
Rate
Time (sec)
0
50
100
150
200
250
300
0
100
200
300
400
500
600

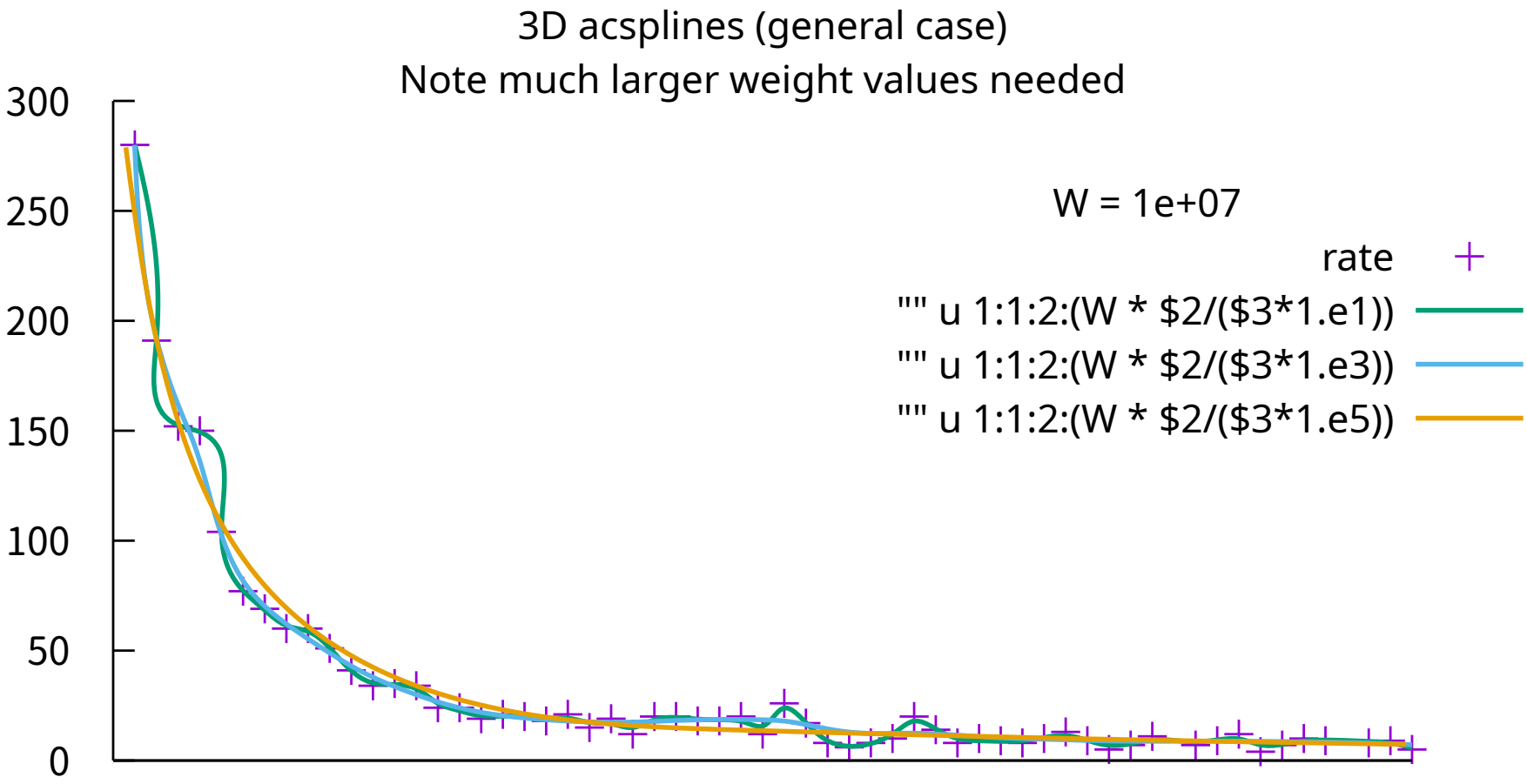

acsplines with increasing weight from error estimate
rate
acspline Y/(Z*1.e1)
Y/(Z*1.e3)
Y/(Z*1.e5)
Rate
Time (sec)
0
50
100
150
200
250
300
0
100
200
300
400
500
600

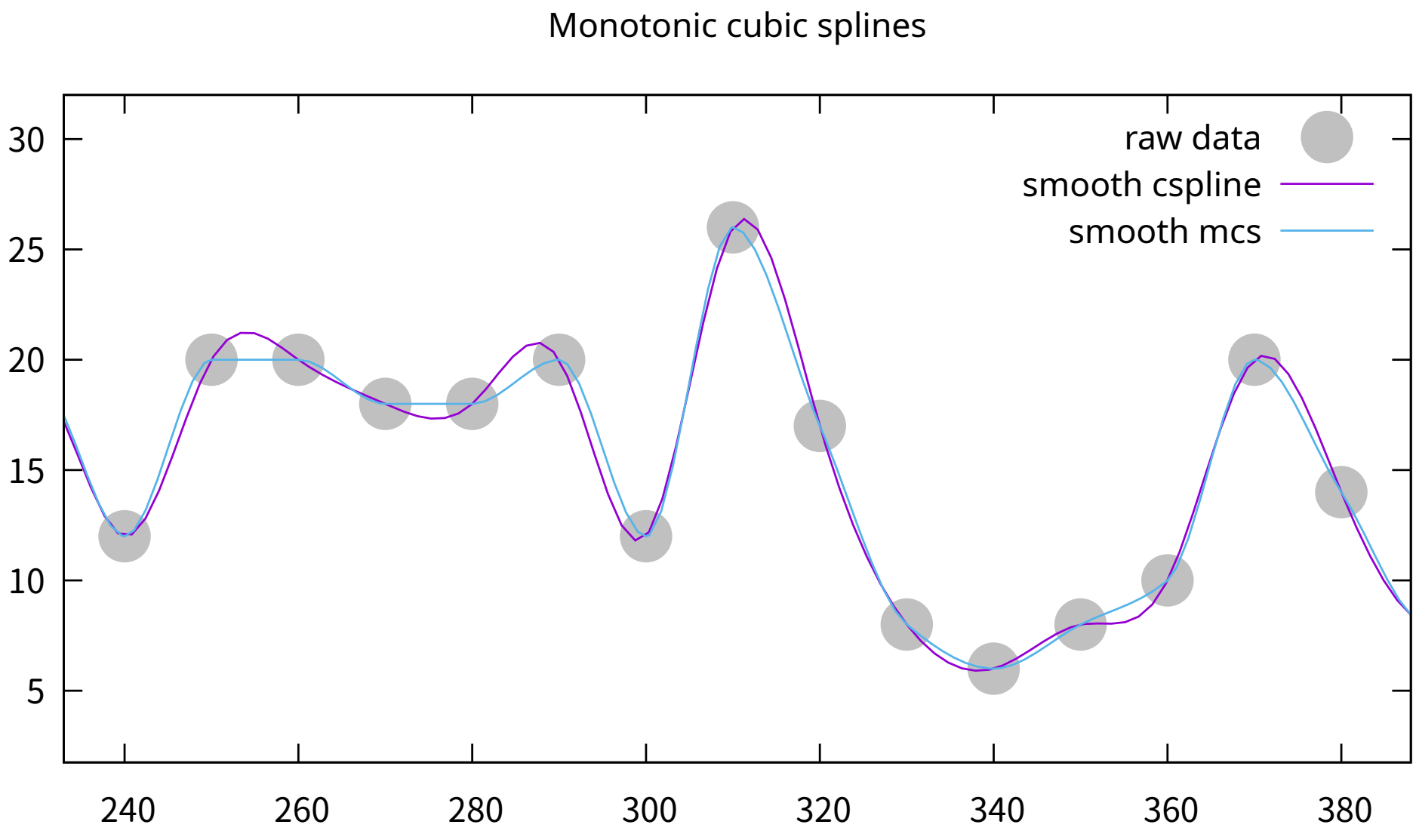

Same plot (various weighting) in log scale
rate
acspline Y/(Z*1.e1)
Y/(Z*1.e3)
Y/(Z*1.e5)
Rate
Time (sec)
1000
100
10
1
0
100
200
300
400
500
600

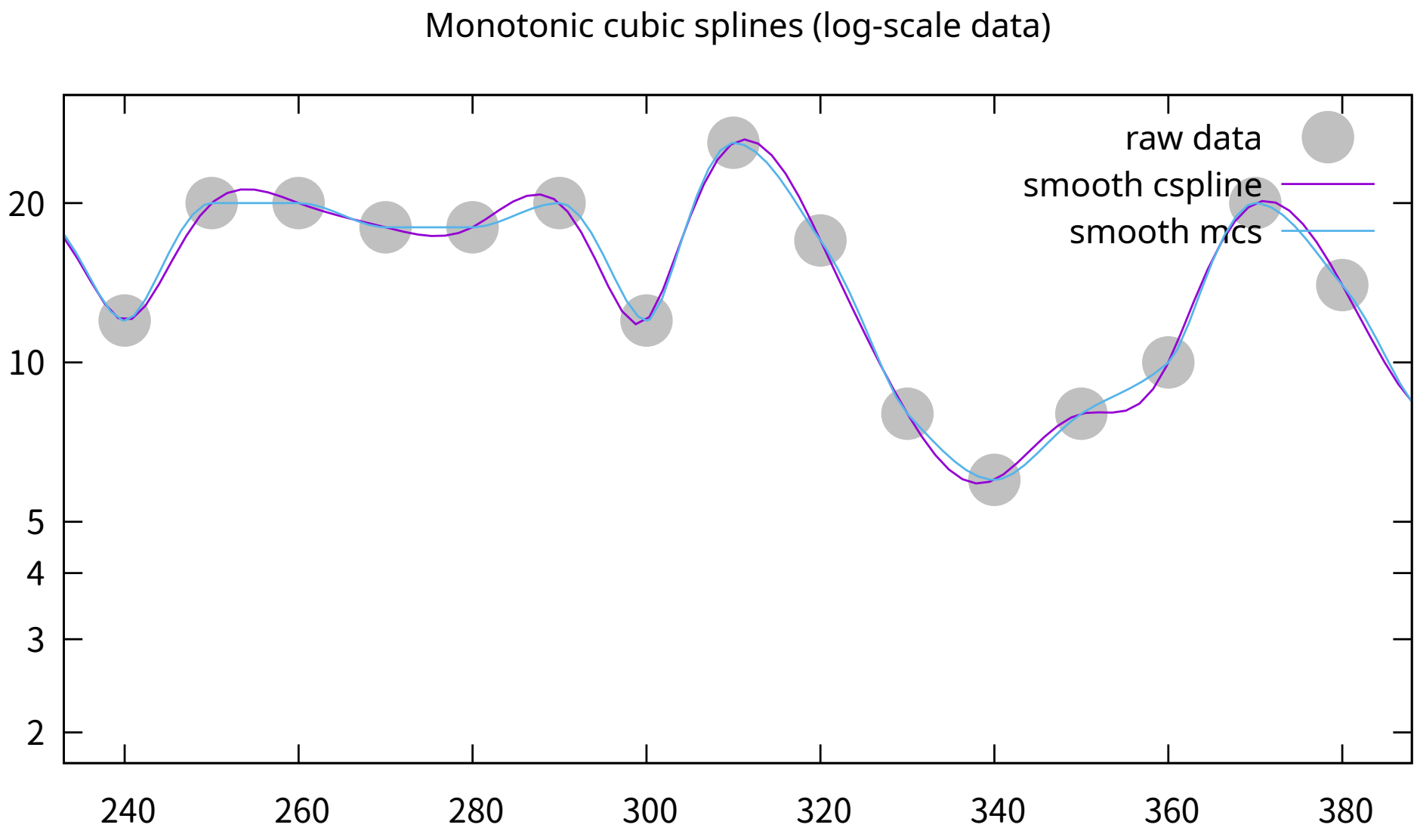

Bezier curve rather than cubic spline
experimental
bezier
Rate
Time (sec)
0
50
100
150
200
250
300
0
100
200
300
400
500
600

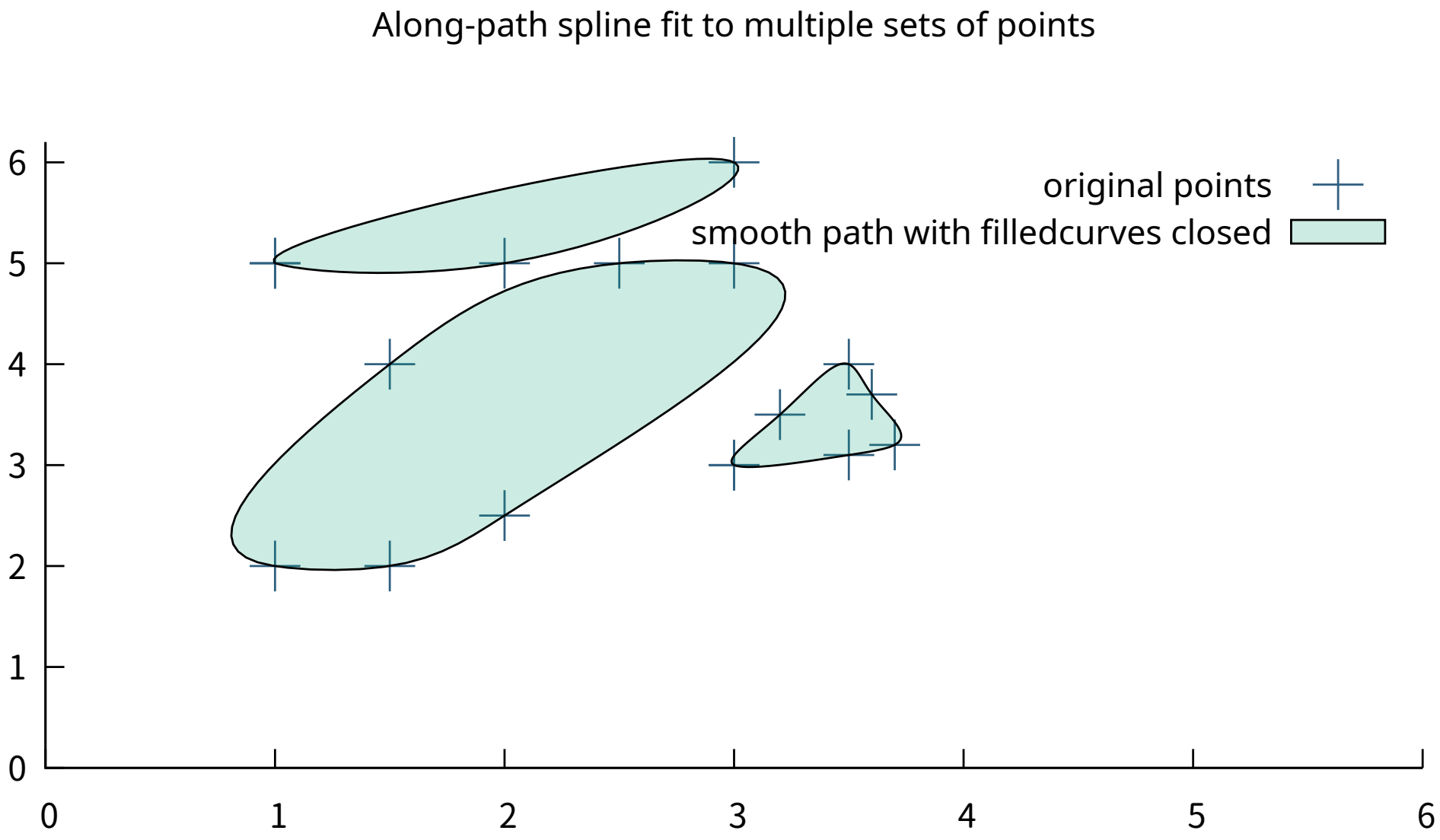

Bezier curve with log scale
experimental
bezier
Rate
Time (sec)
1000
100
10
1
0
100
200
300
400
500
600

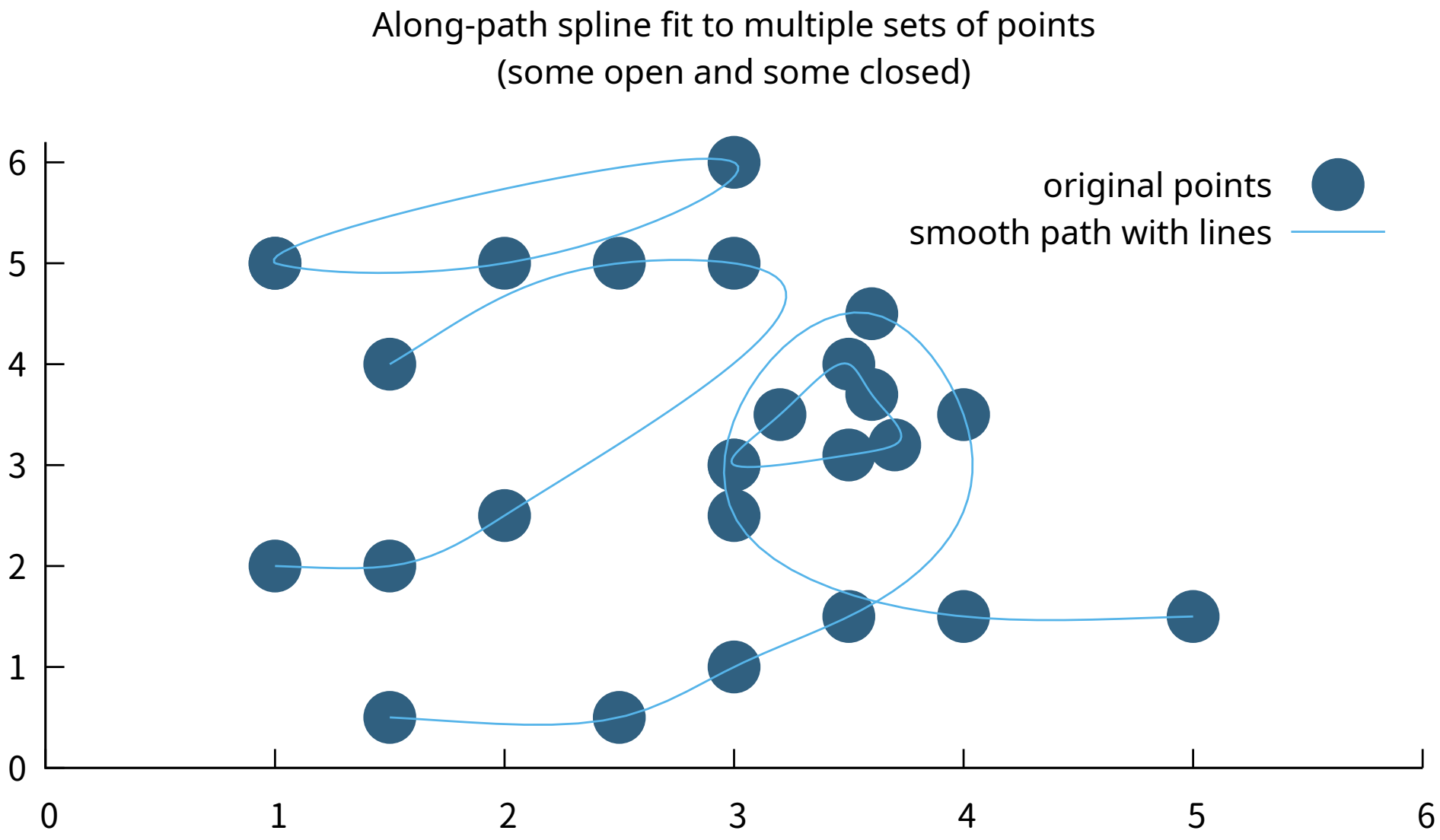

3D smooth acsplines (special case with curve in single plane)
300
250
200
150
100
50
0
0
100
200
300
400
500
600
Time
rate
acspline Y/(Z*1.e1)
Y/(Z*1.e3)
Y/(Z*1.e5)


3D acsplines (general case)
Note much larger weight values needed
300
250
200
150
100
50
0
W = 1e+07
rate
"" u 1:1:2:(W * $2/($3*1.e1))
"" u 1:1:2:(W * $2/($3*1.e3))
"" u 1:1:2:(W * $2/($3*1.e5))

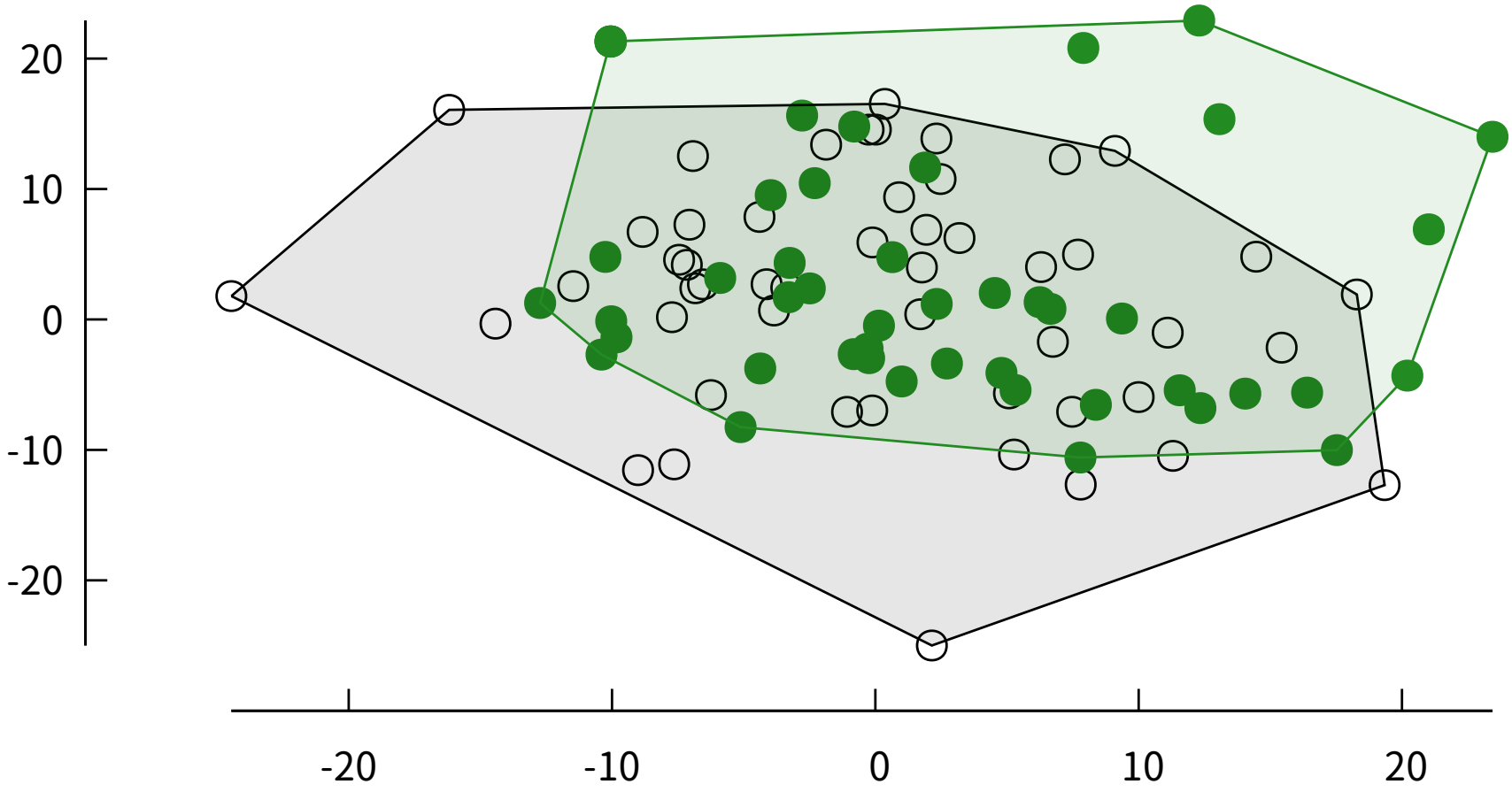

Monotonic cubic splines
raw data
smooth cspline
smooth mcs
30
25
20
15
10
5
240
260
280
300
320
340
360
380


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

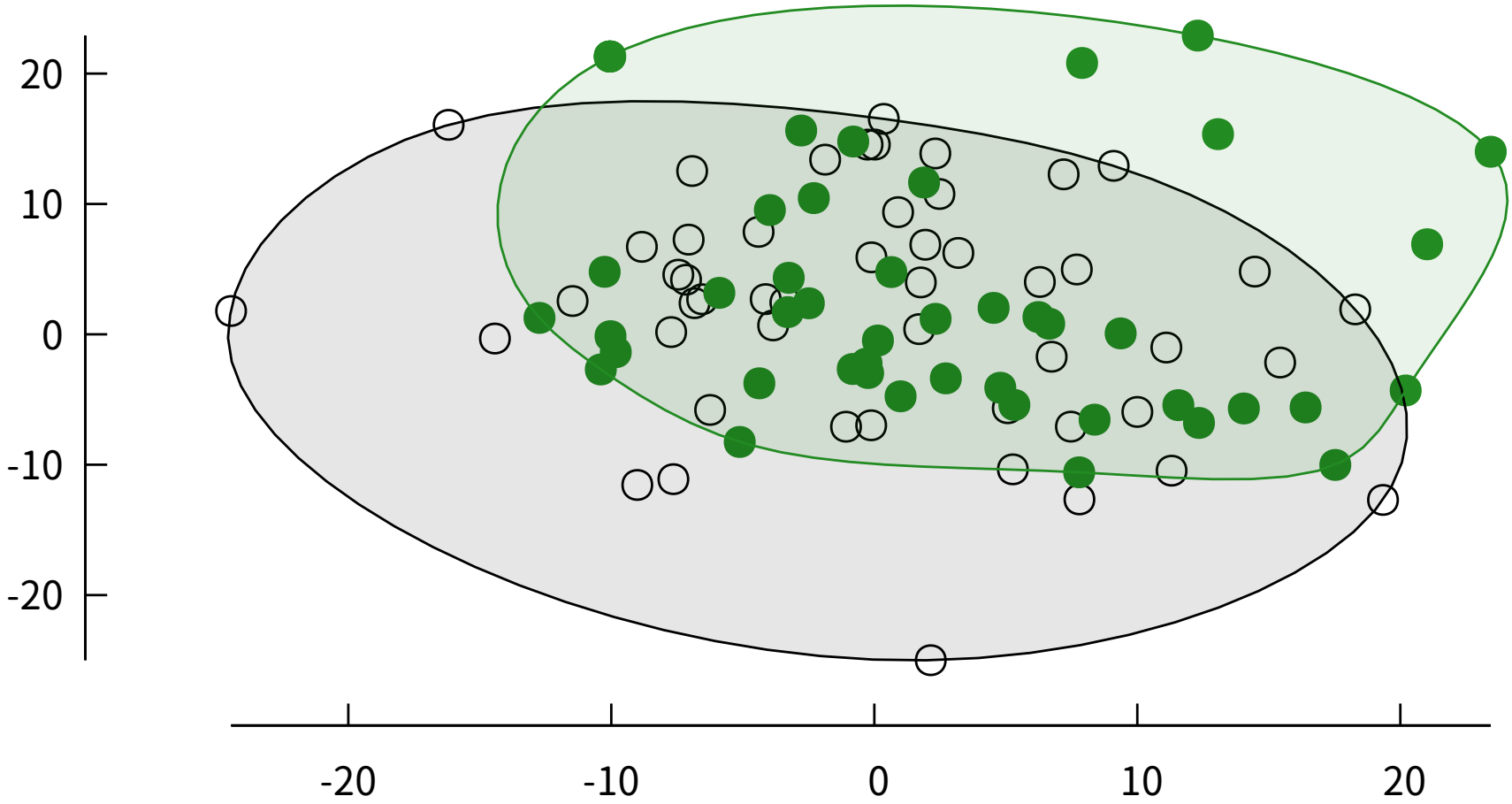

Along-path spline fit to multiple sets of points
original points
smooth path with filledcurves closed
0
1
2
3
4
5
6
0
1
2
3
4
5
6


Along-path spline fit to multiple sets of points
(some open and some closed)
original points
smooth path with lines
0
1
2
3
4
5
6
0
1
2
3
4
5
6

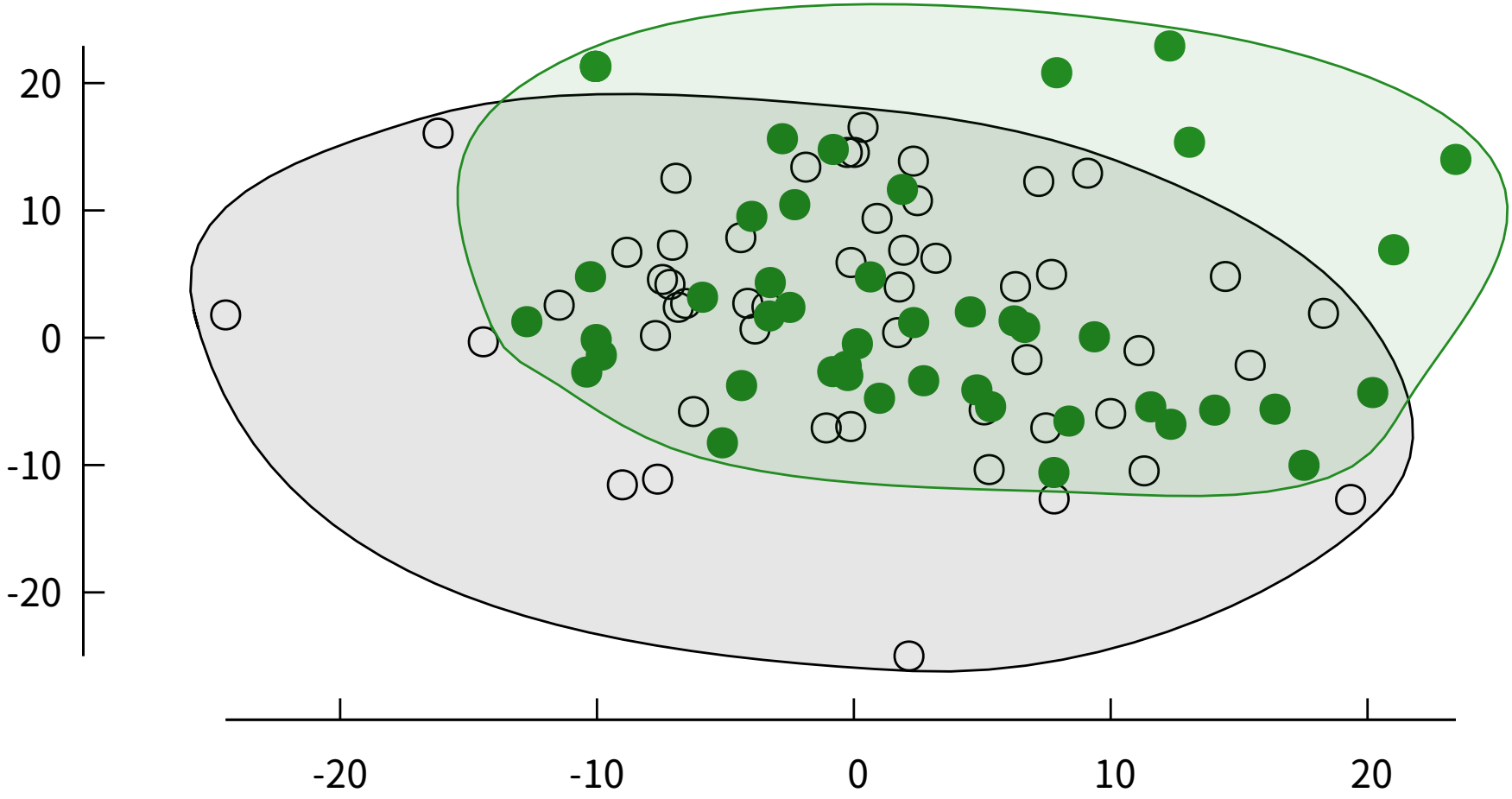
Generation of convex hull bounding scattered points

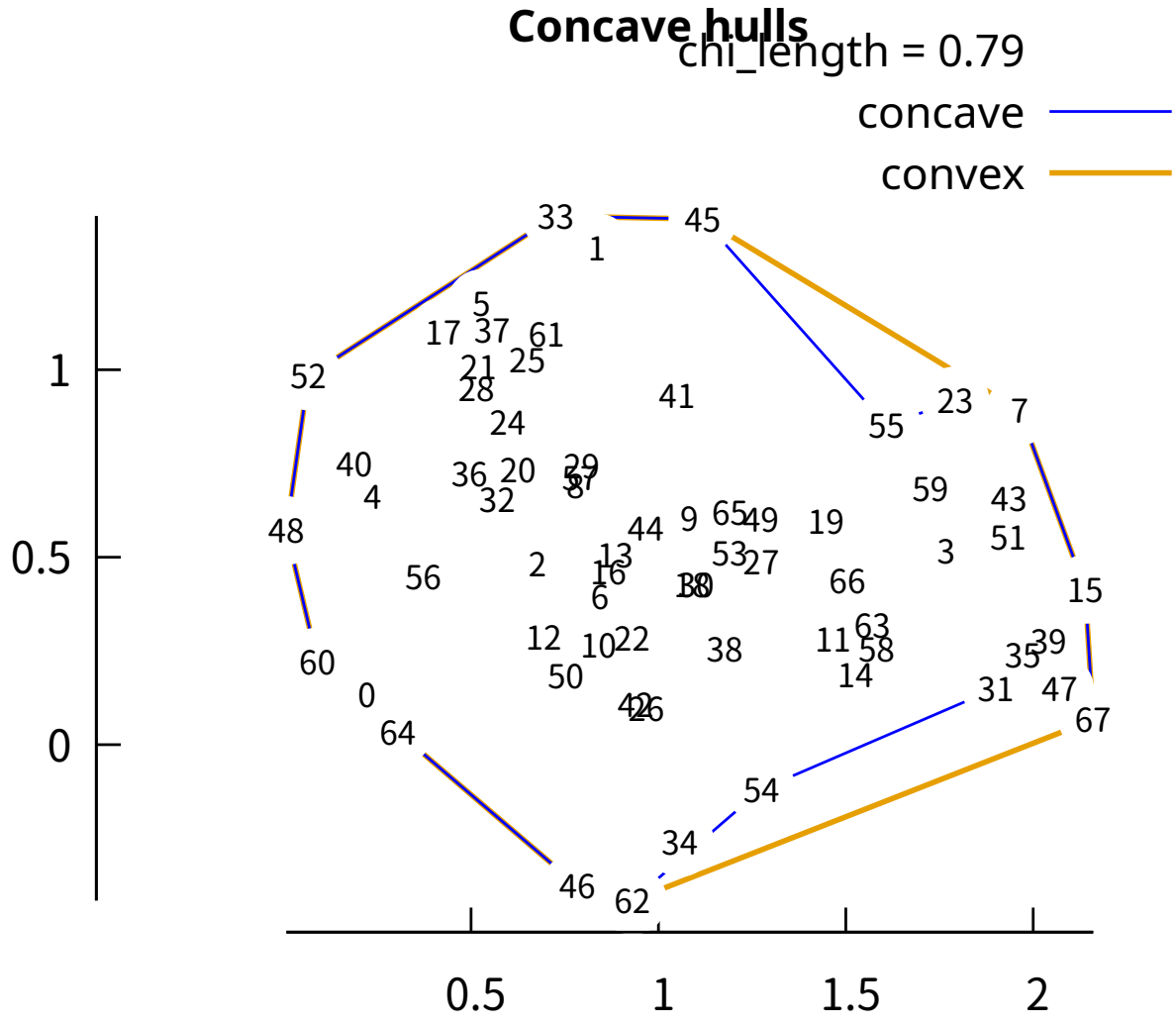


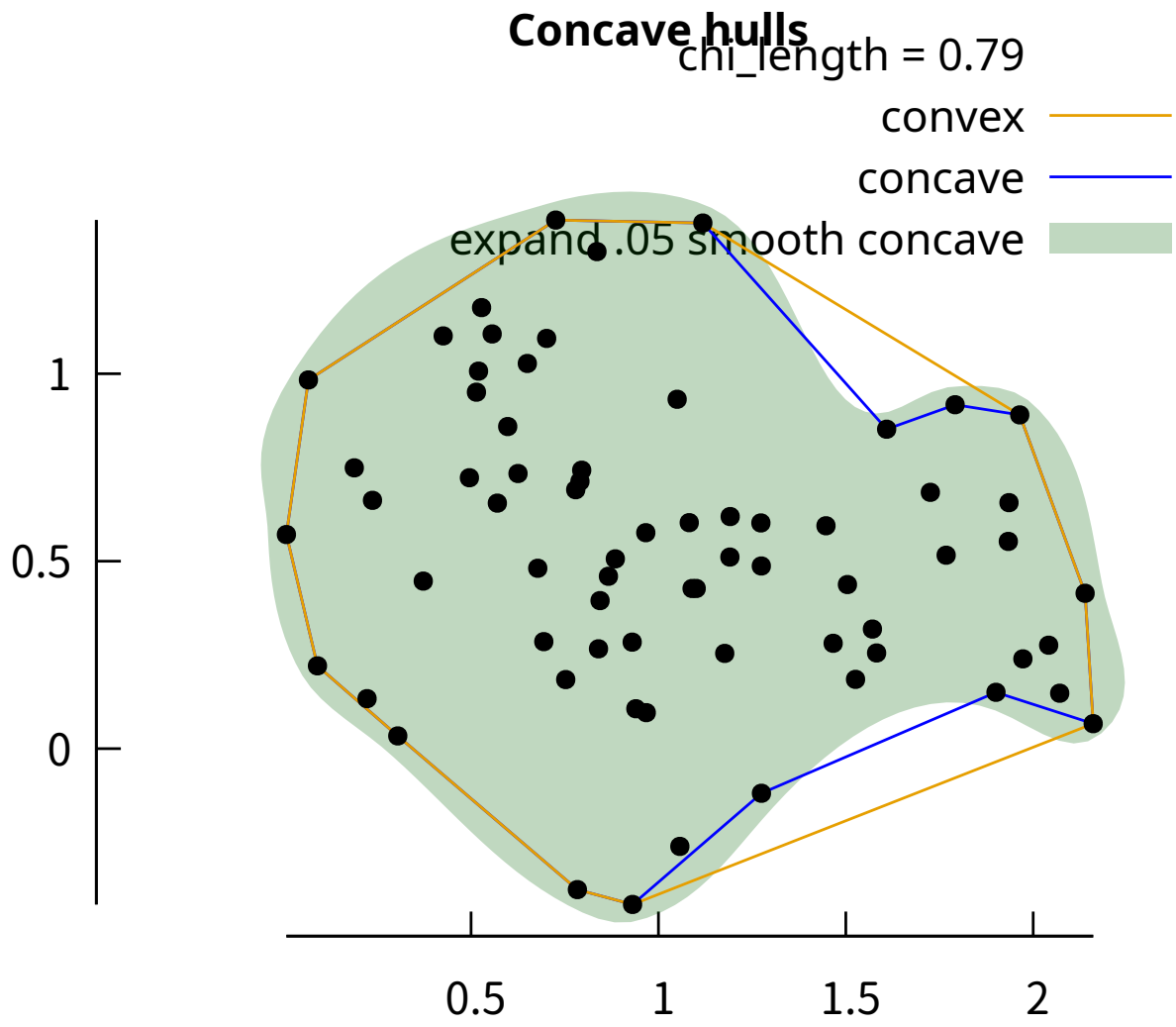
Smooth convex hull

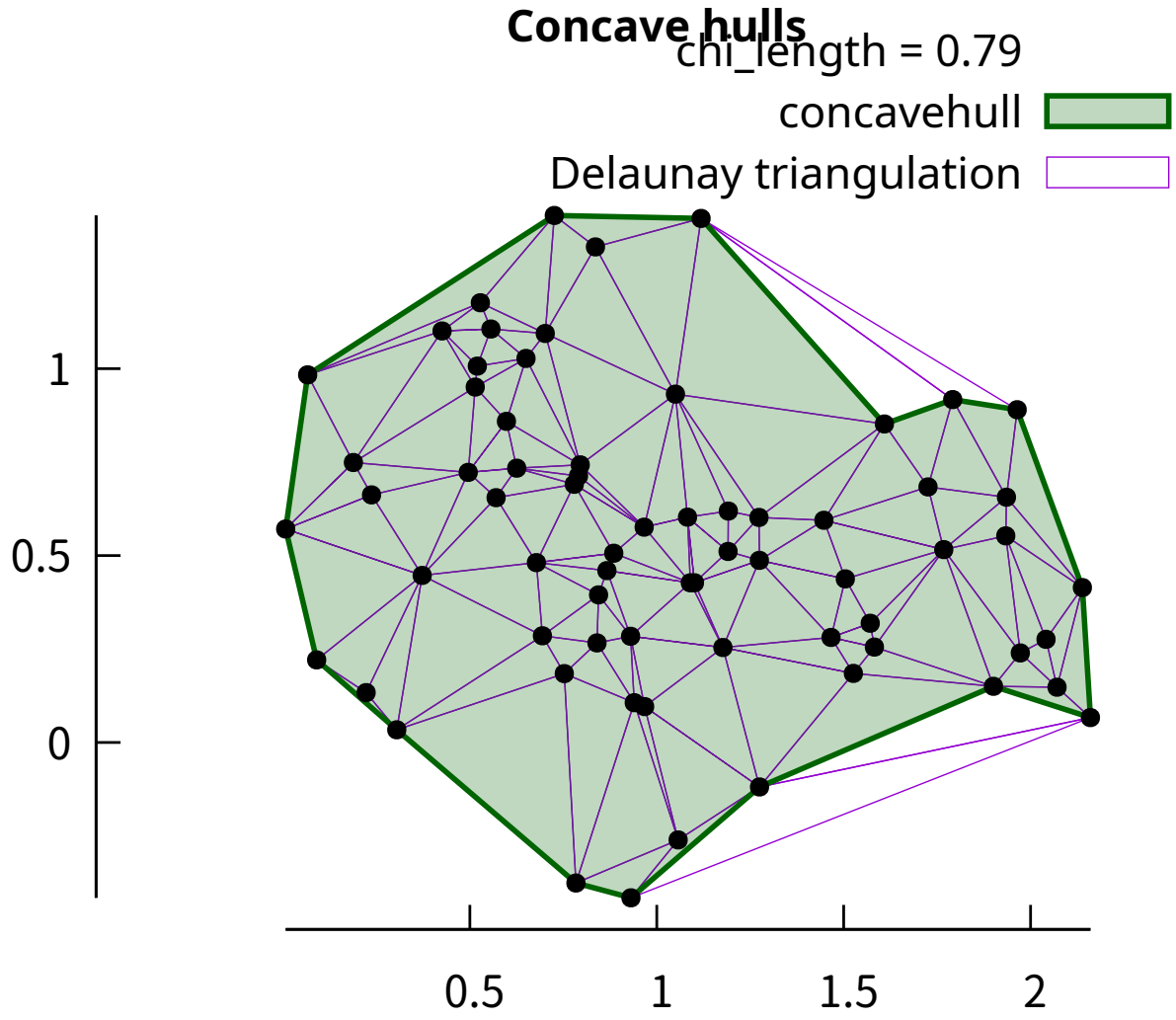


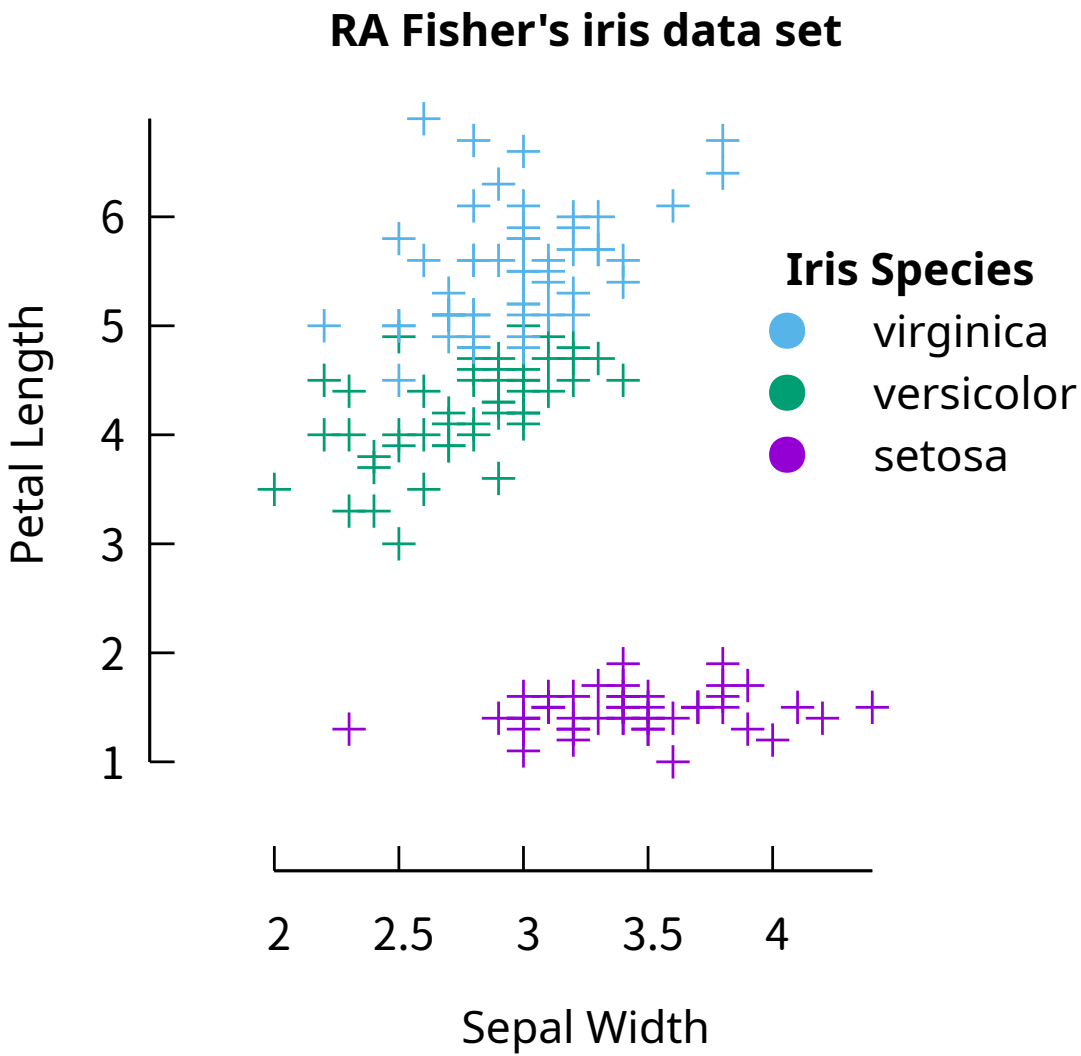
Smooth convex hull expanded by increment of 1.5

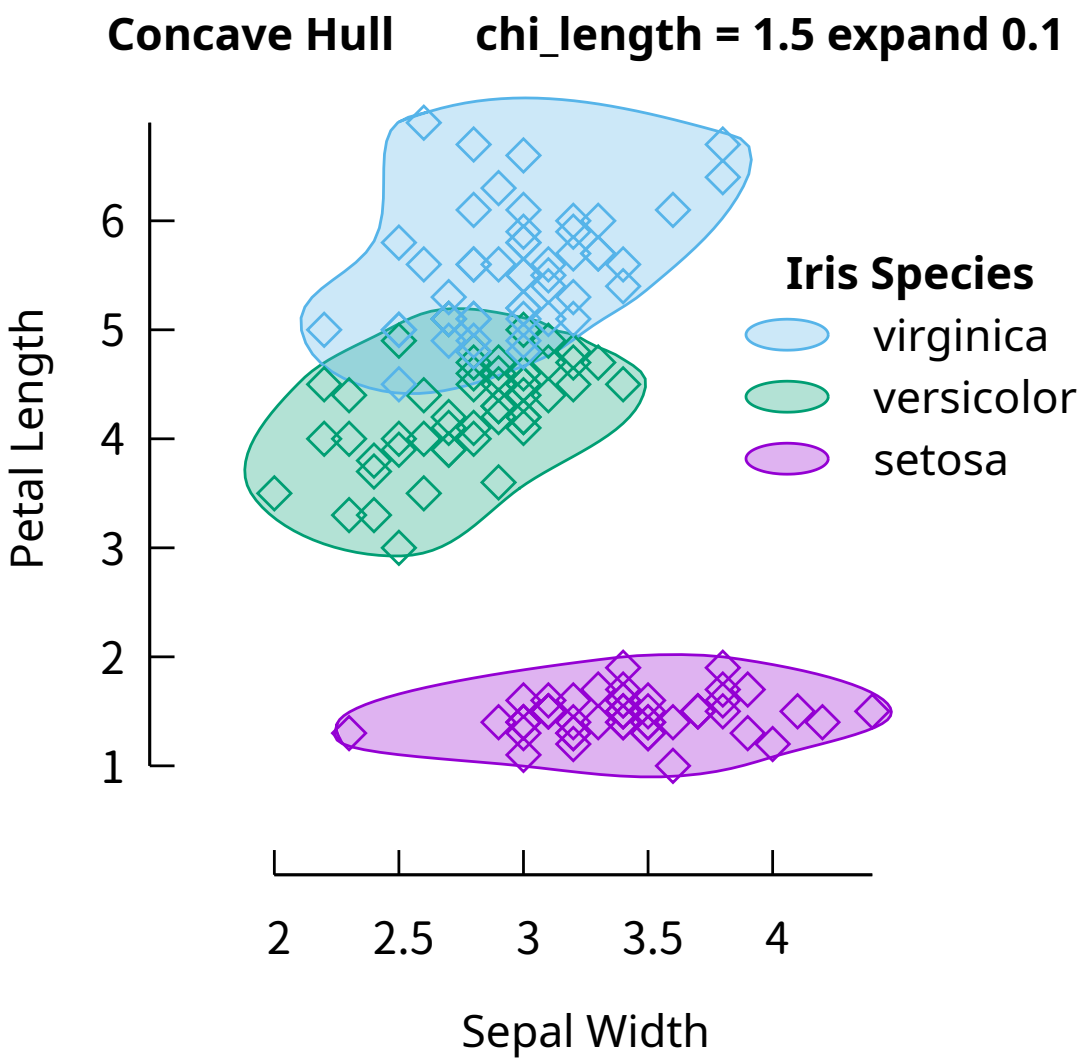



Concave hulls
chi_length = 0.79
concave
convex
1
0.5
0
0.5
1
1.5
2

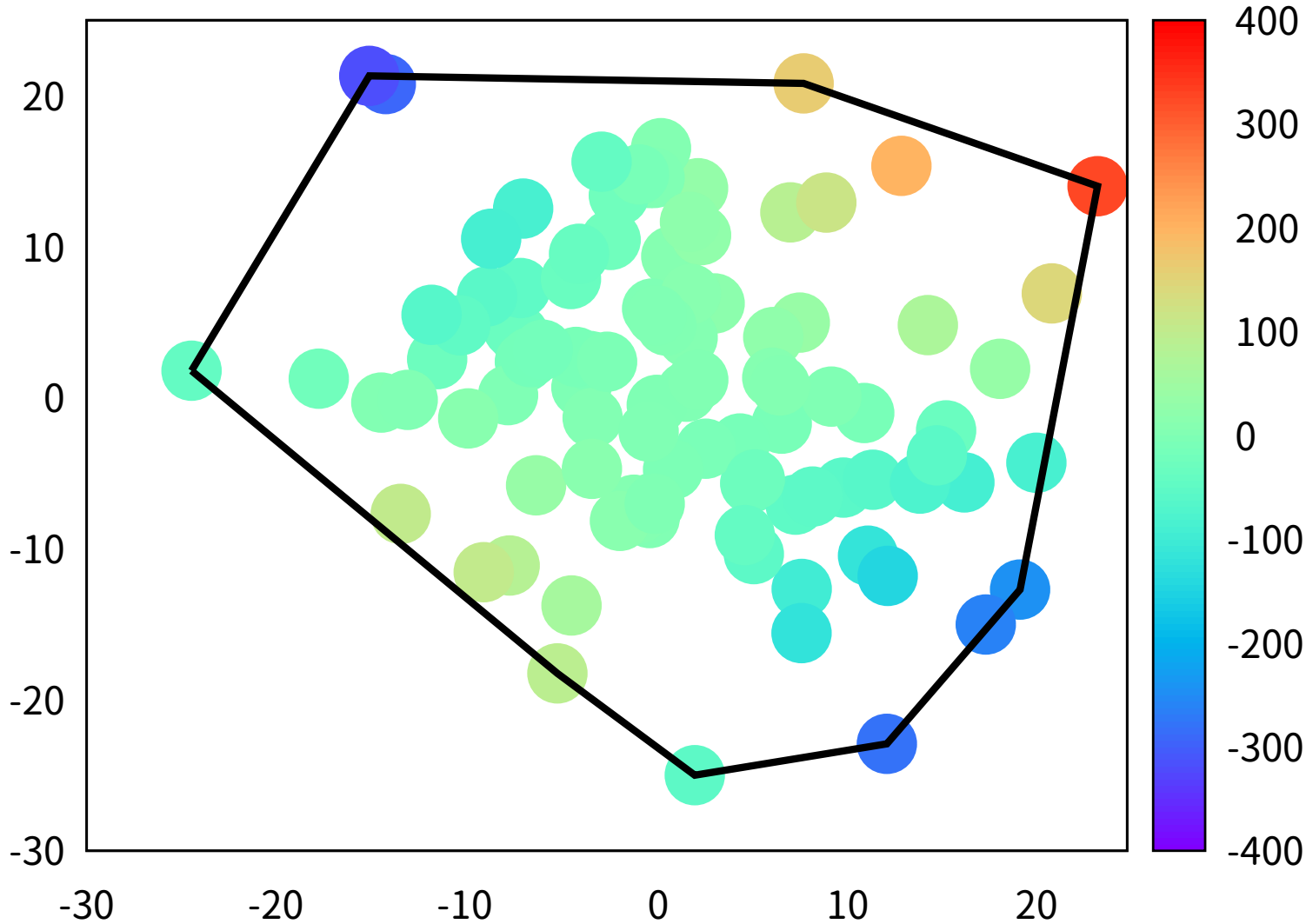

Concave hulls
chi_length = 0.79
convex
concave
expand .05 smooth concave
1
0.5
0
0.5
1
1.5
2


Concave hulls
chi_length = 0.79
concavehull
Delaunay triangulation
1
0.5
0
0.5
1
1.5
2

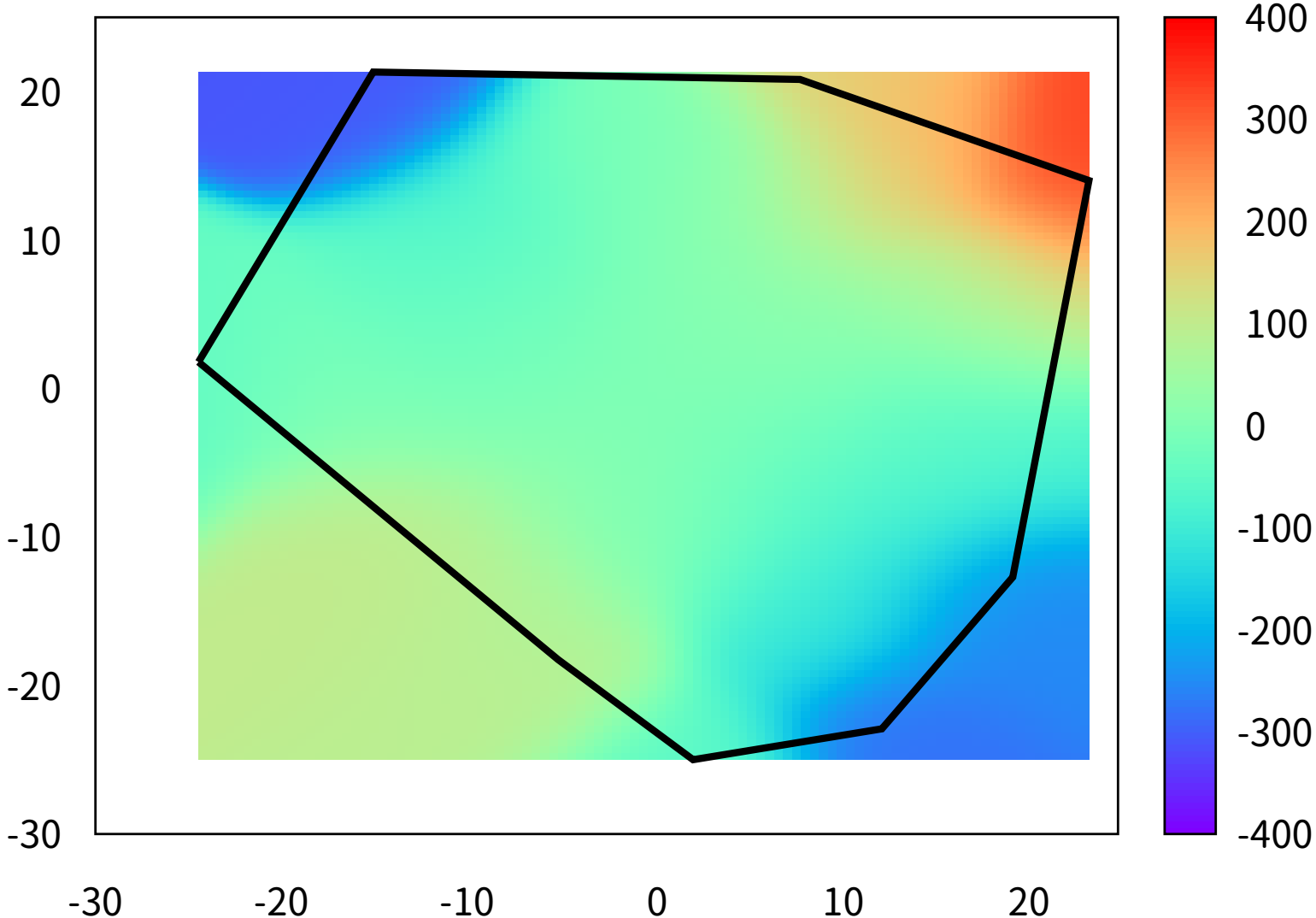

RA Fisher's iris data set
Petal Length
1
2
3
4
5
6
2
2.5
3
3.5
4
Sepal Width
Iris Species
virginica
versicolor
setosa


Concave Hull chi_length = 1.5 expand 0.1
Petal Length
6
5
4
3
2
1
2
2.5
3
3.5
4
Sepal Width
Iris Species
virginica
versicolor
setosa

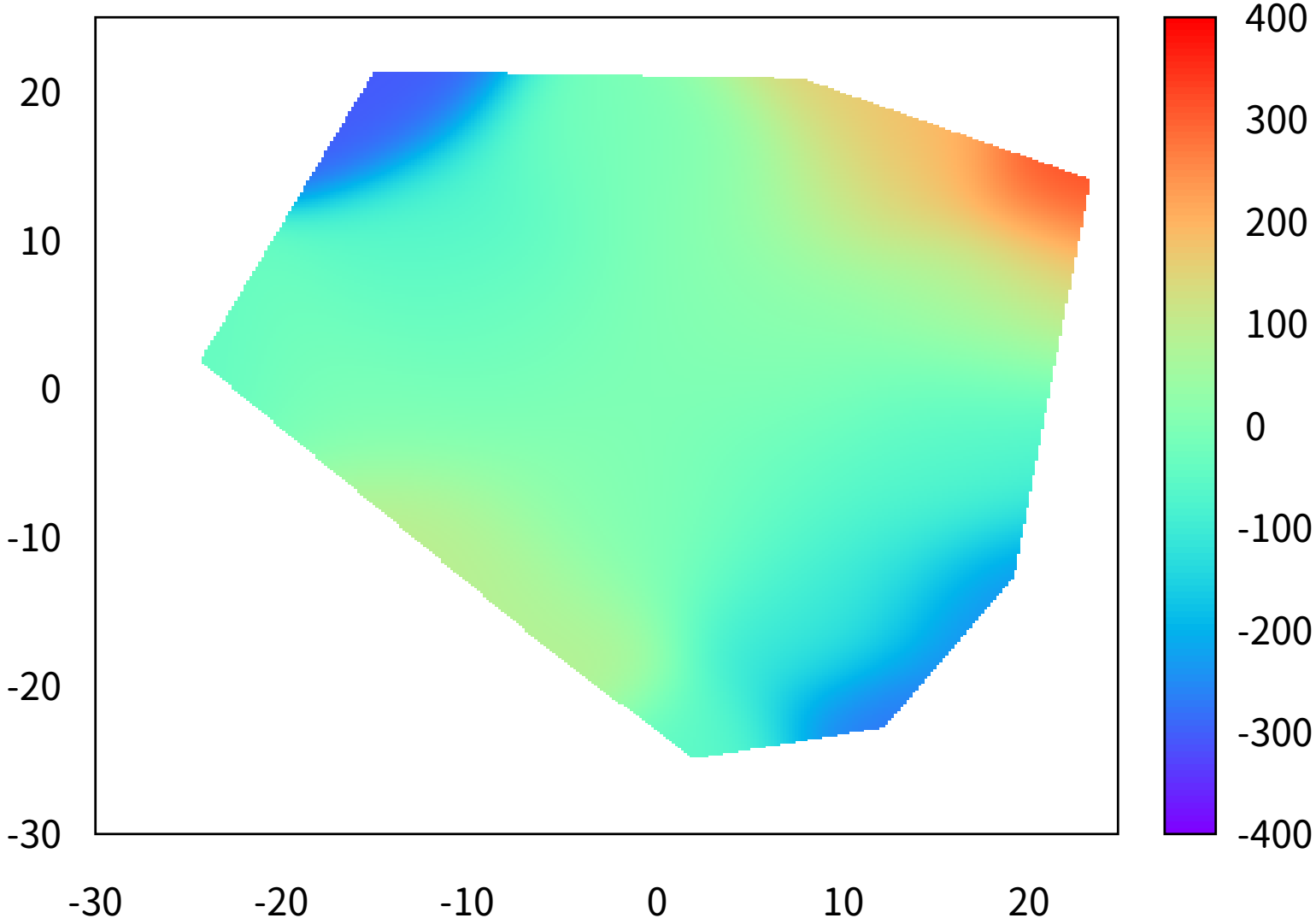
Convex hull constructed around scattered points

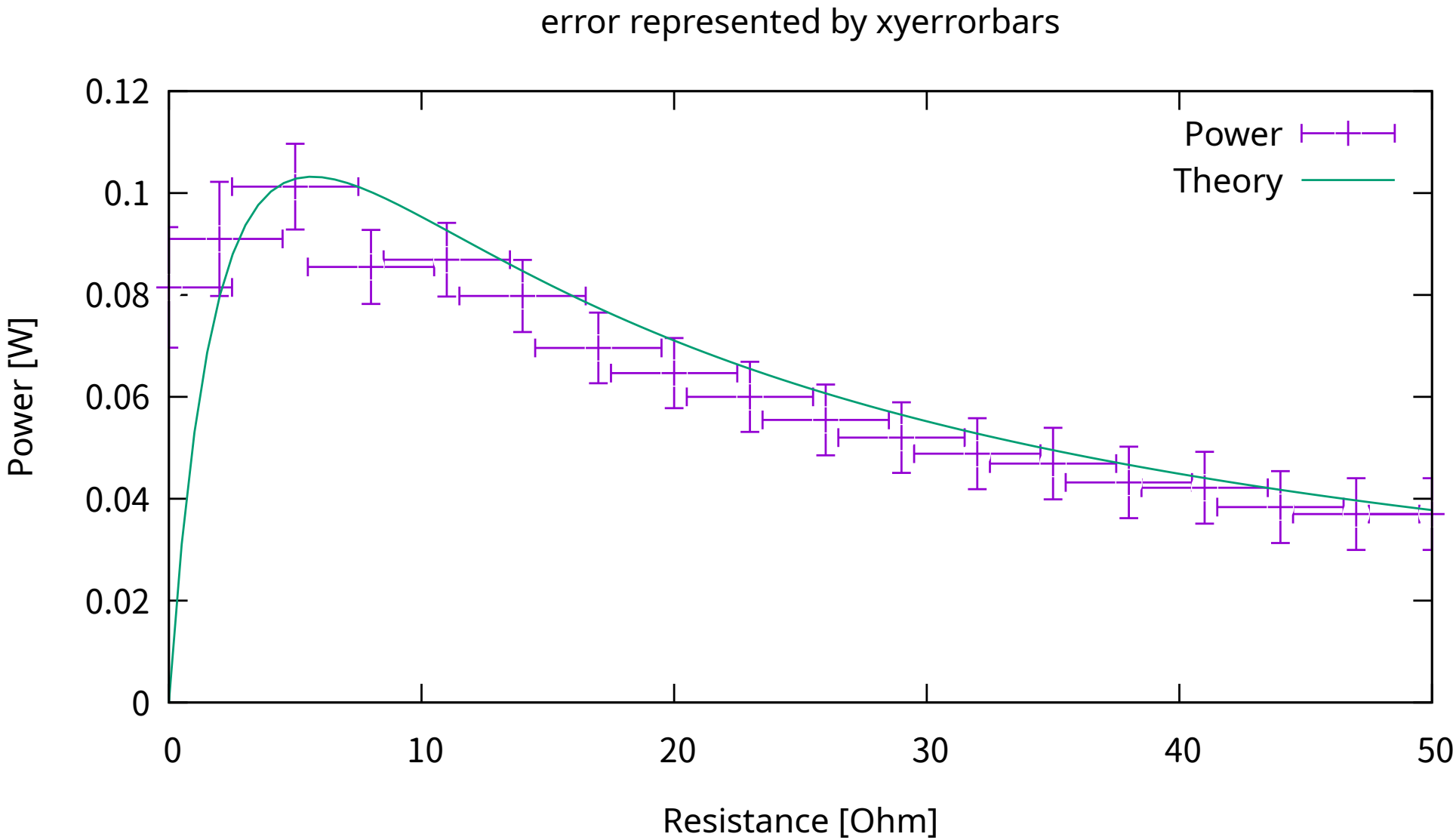


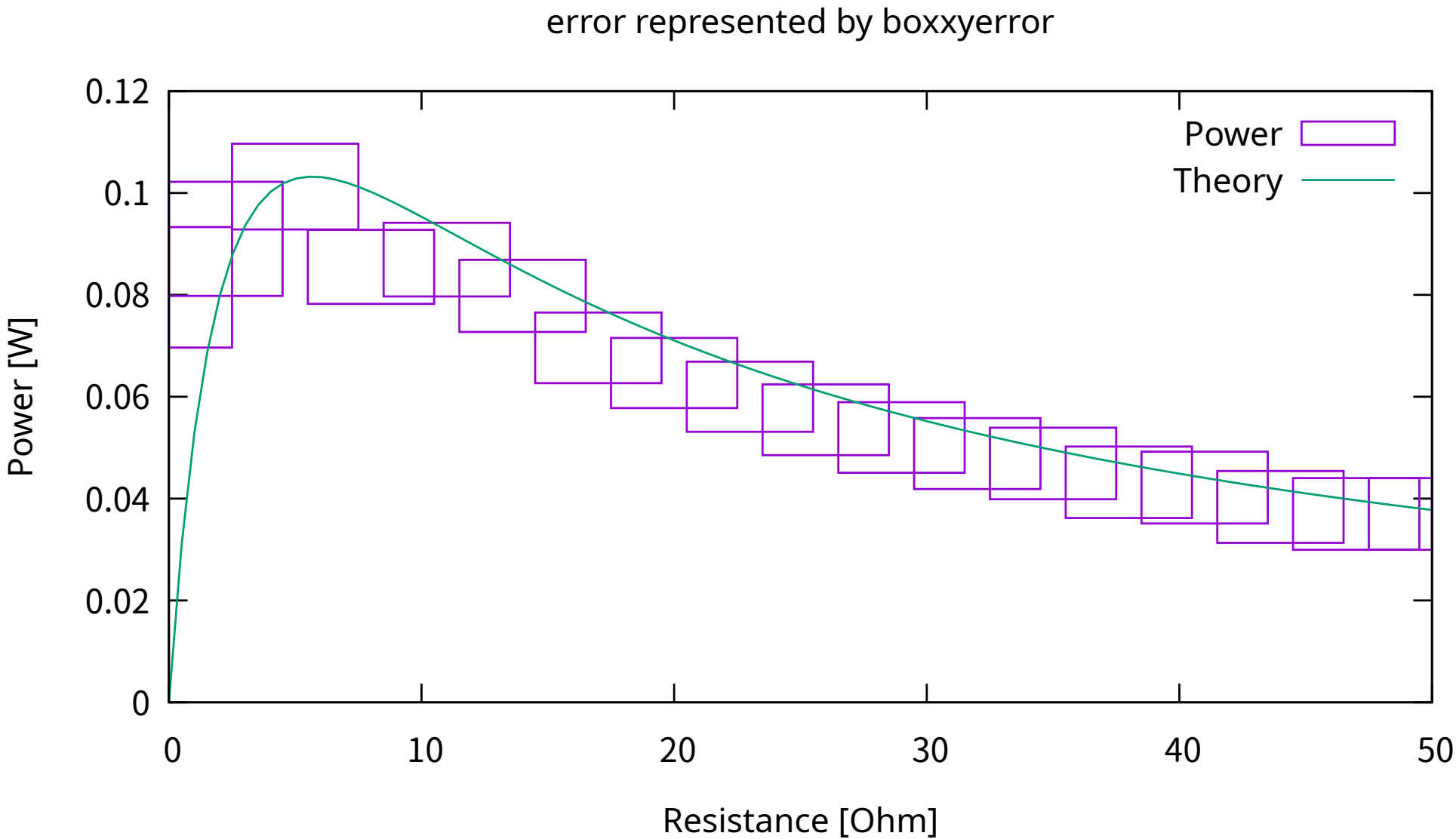
pm3d surface generated from points by dgrid3d

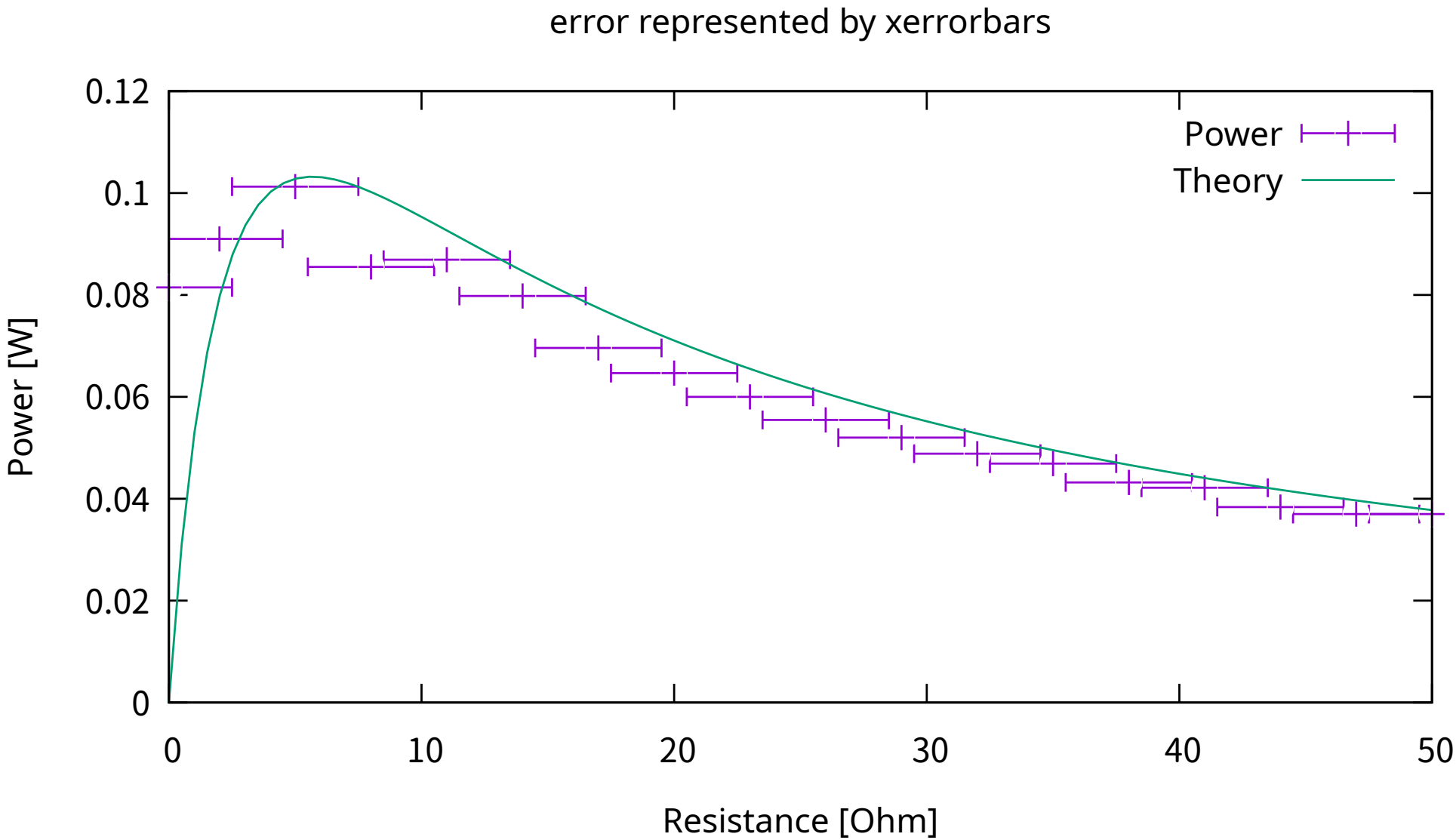


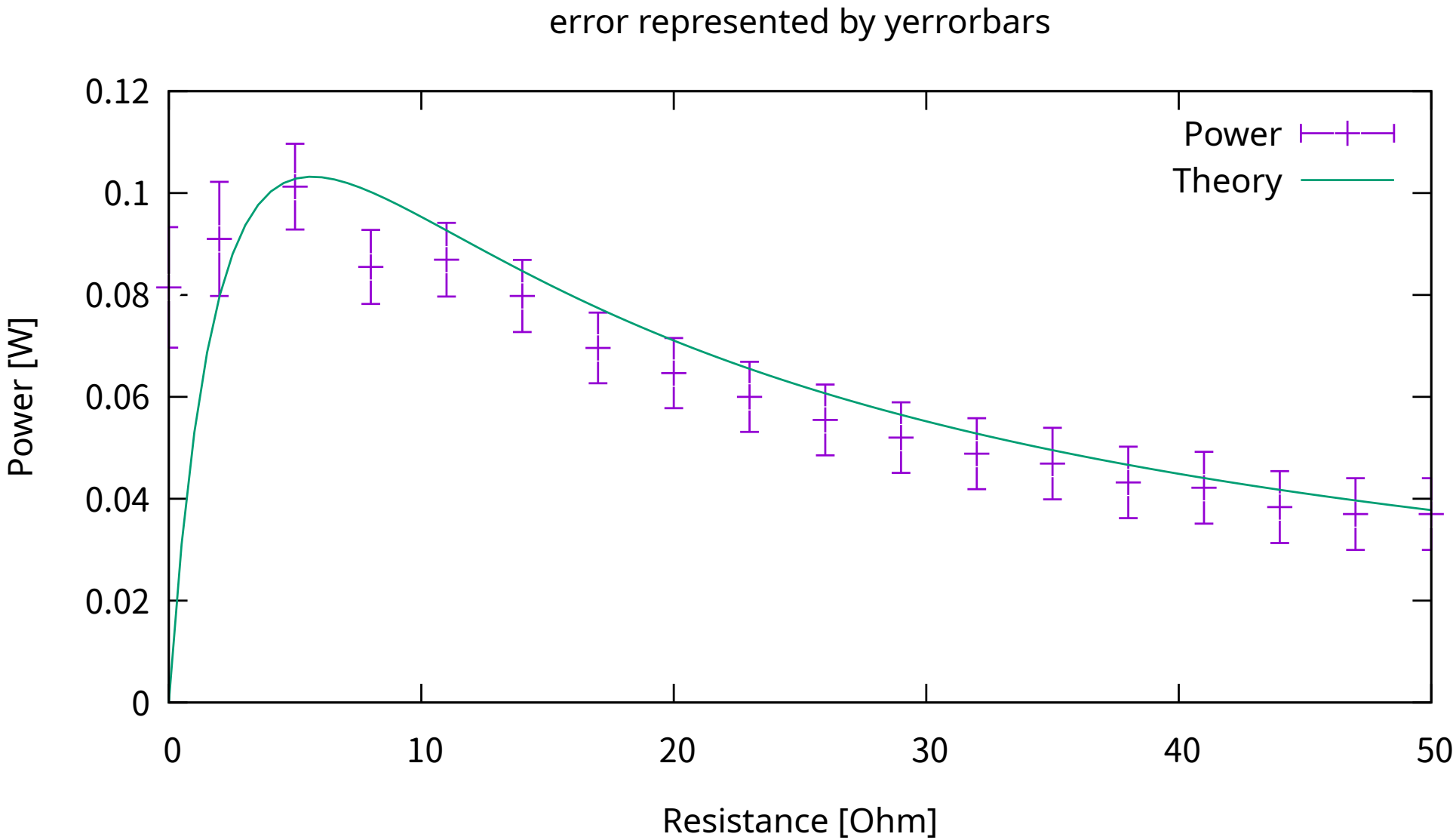
pm3d surface masked by convex hull

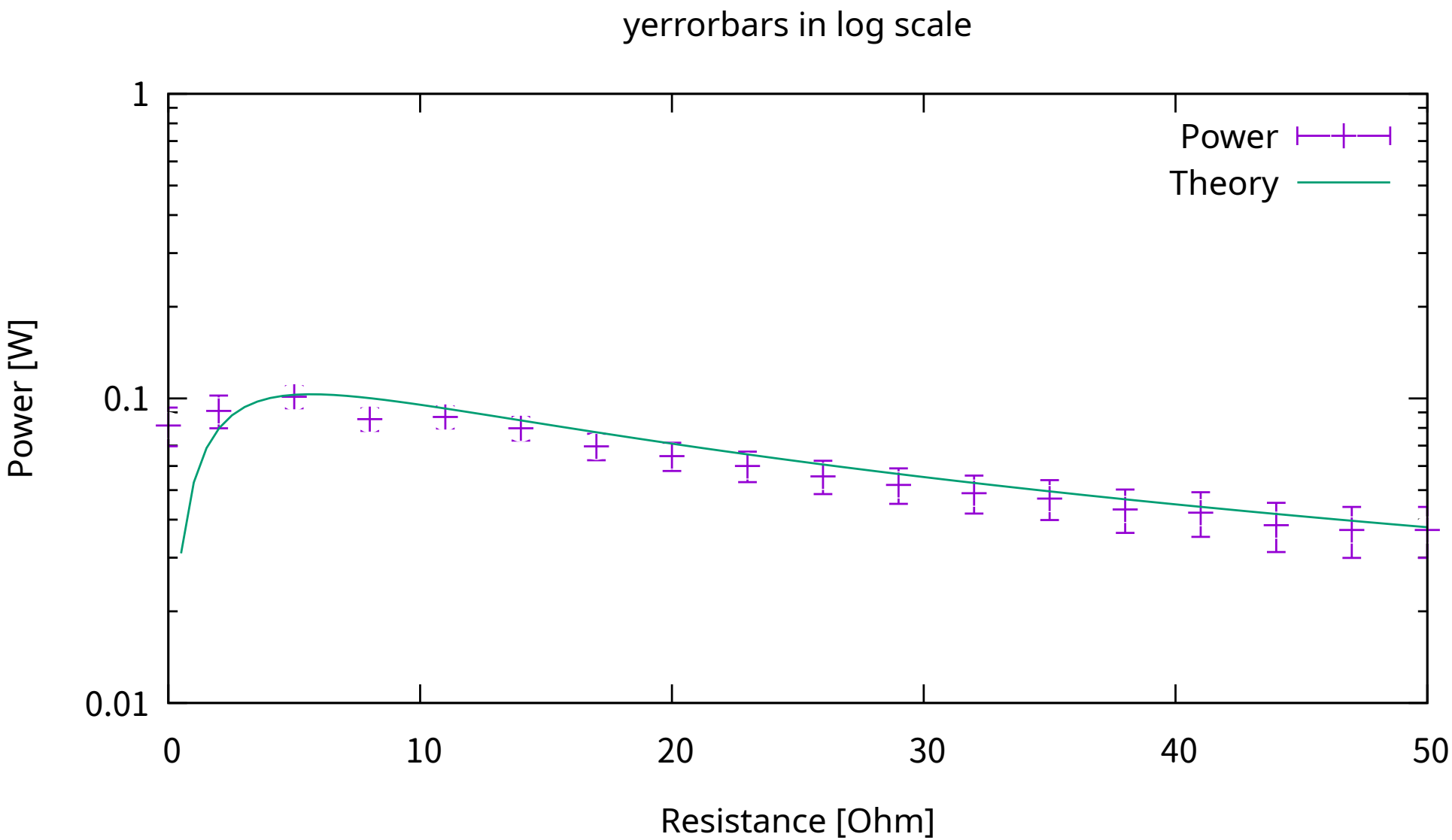


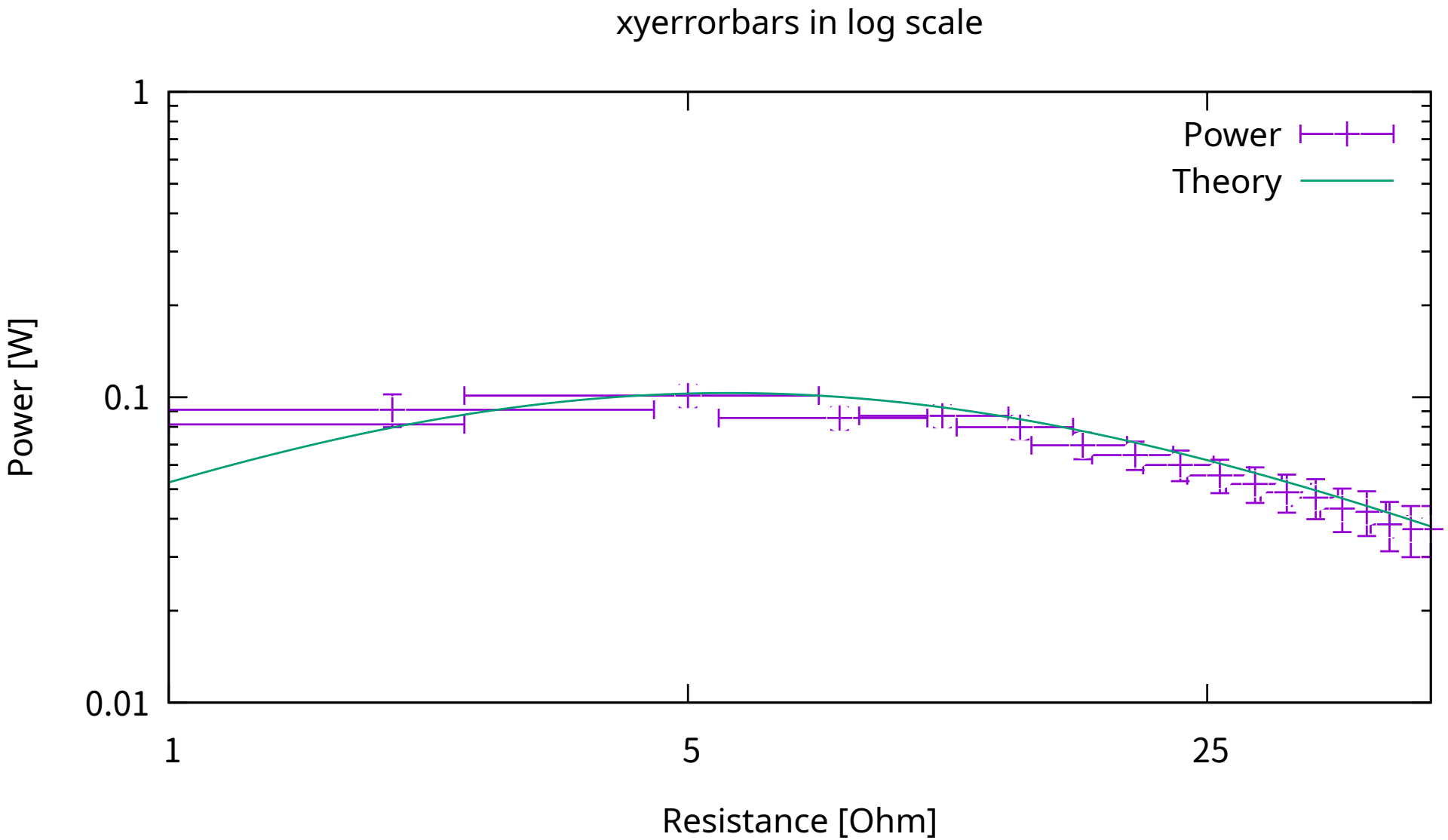

error represented by xyerrorbars
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

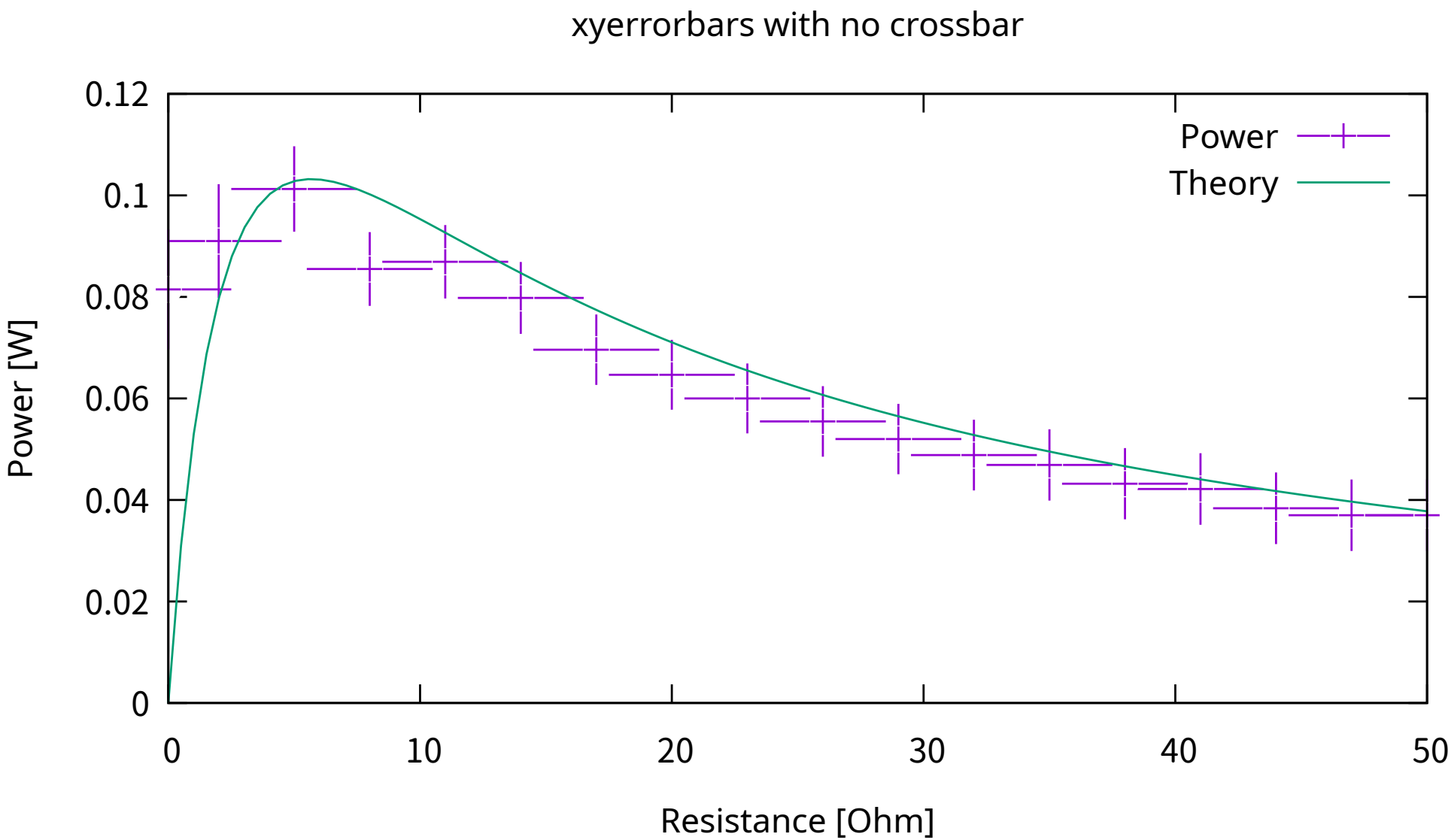

error represented by boxxyerror
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

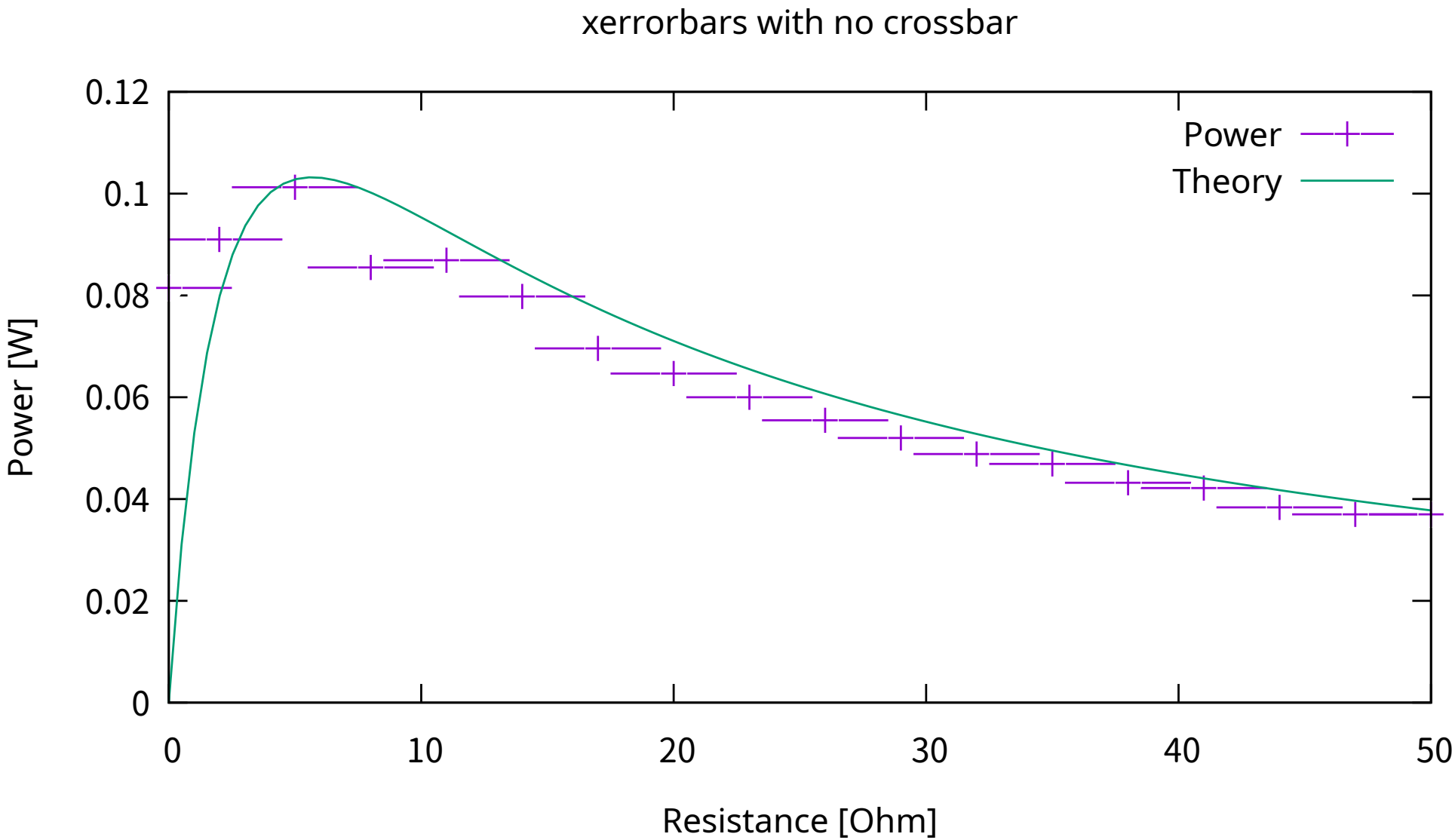

error represented by xerrorbars
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

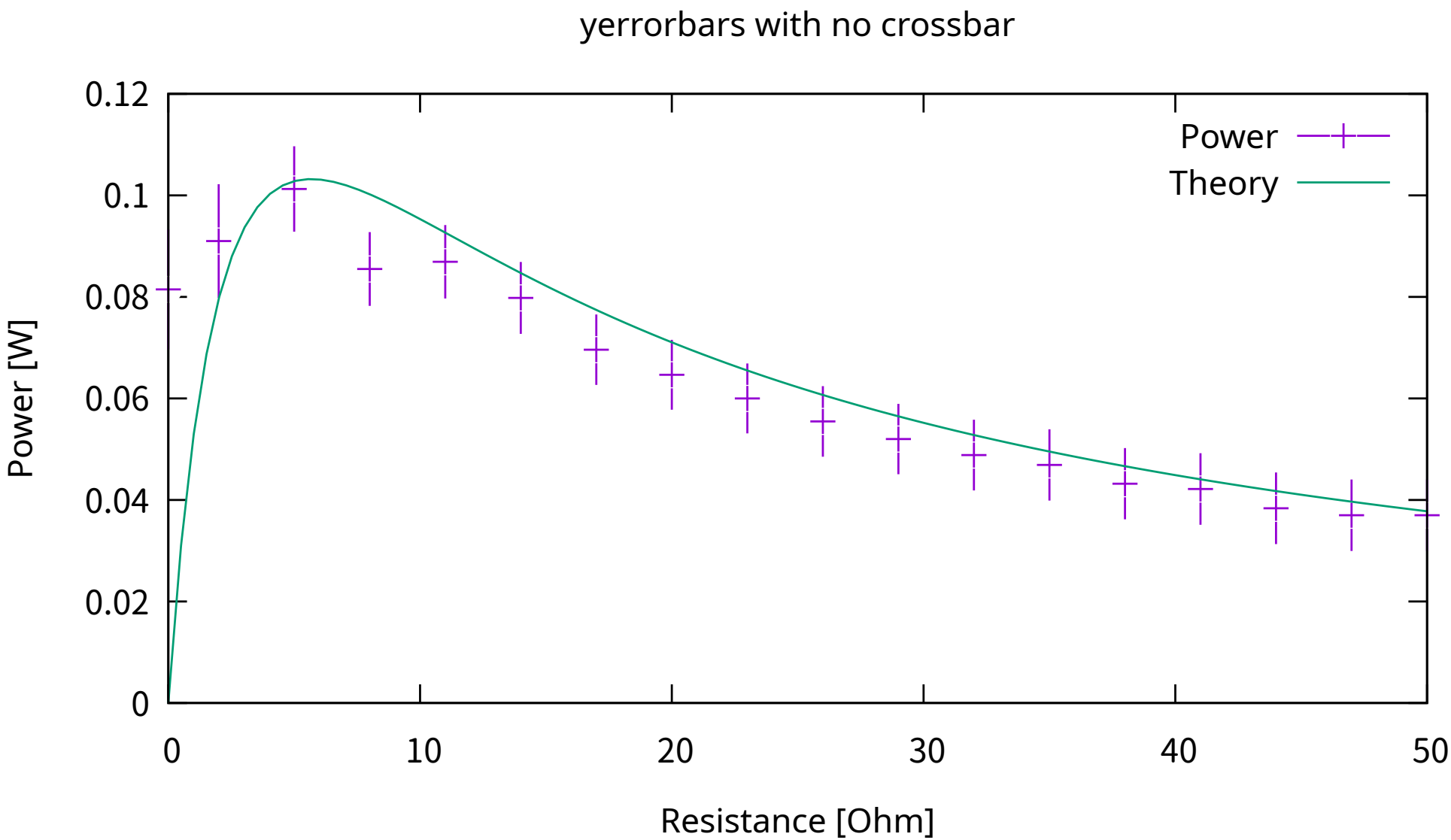

error represented by yerrorbars
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

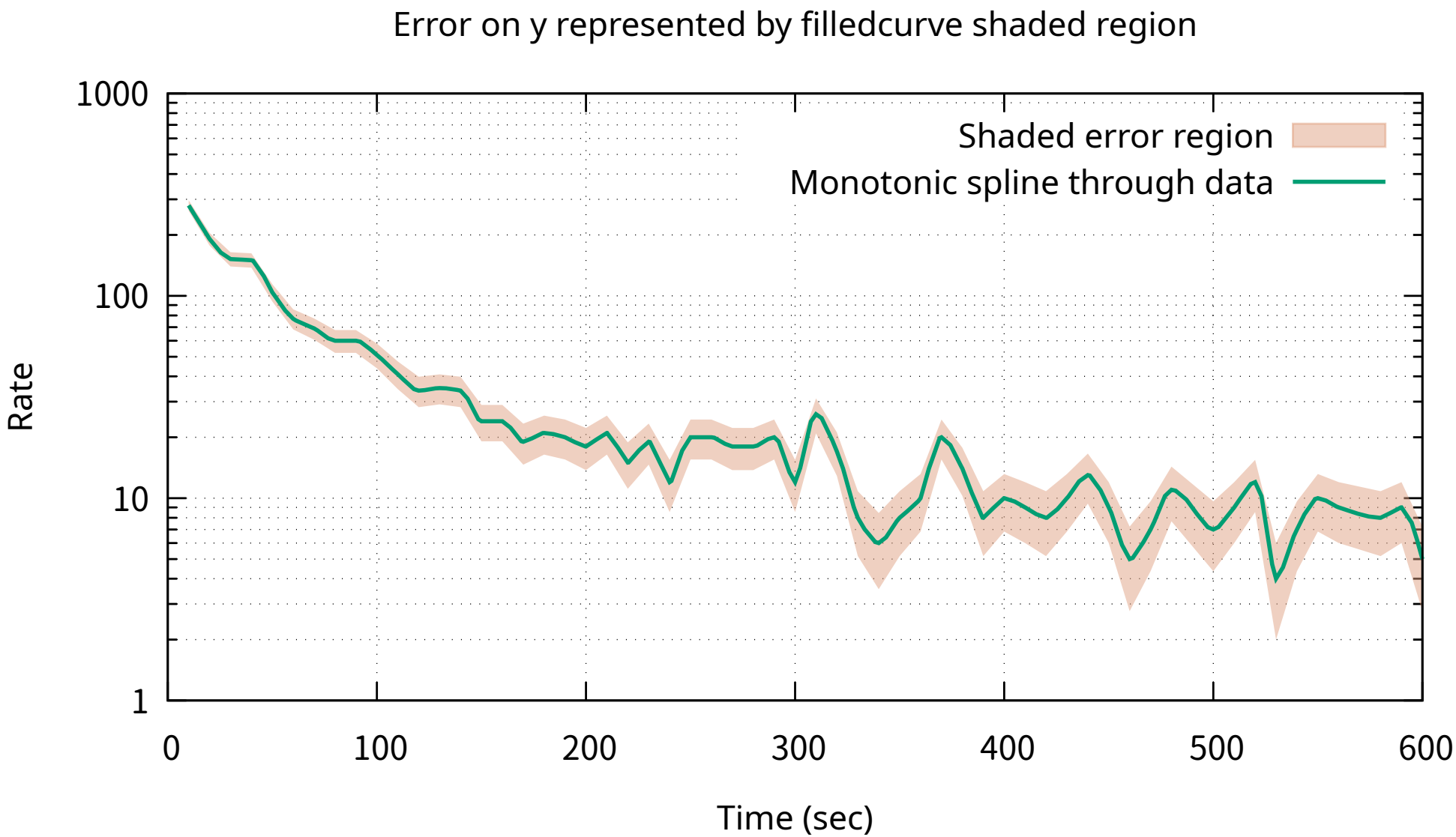

yerrorbars in log scale
Power
Theory
Power [W]
1
0.1
0.01
0
10
20
30
40
50
Resistance [Ohm]

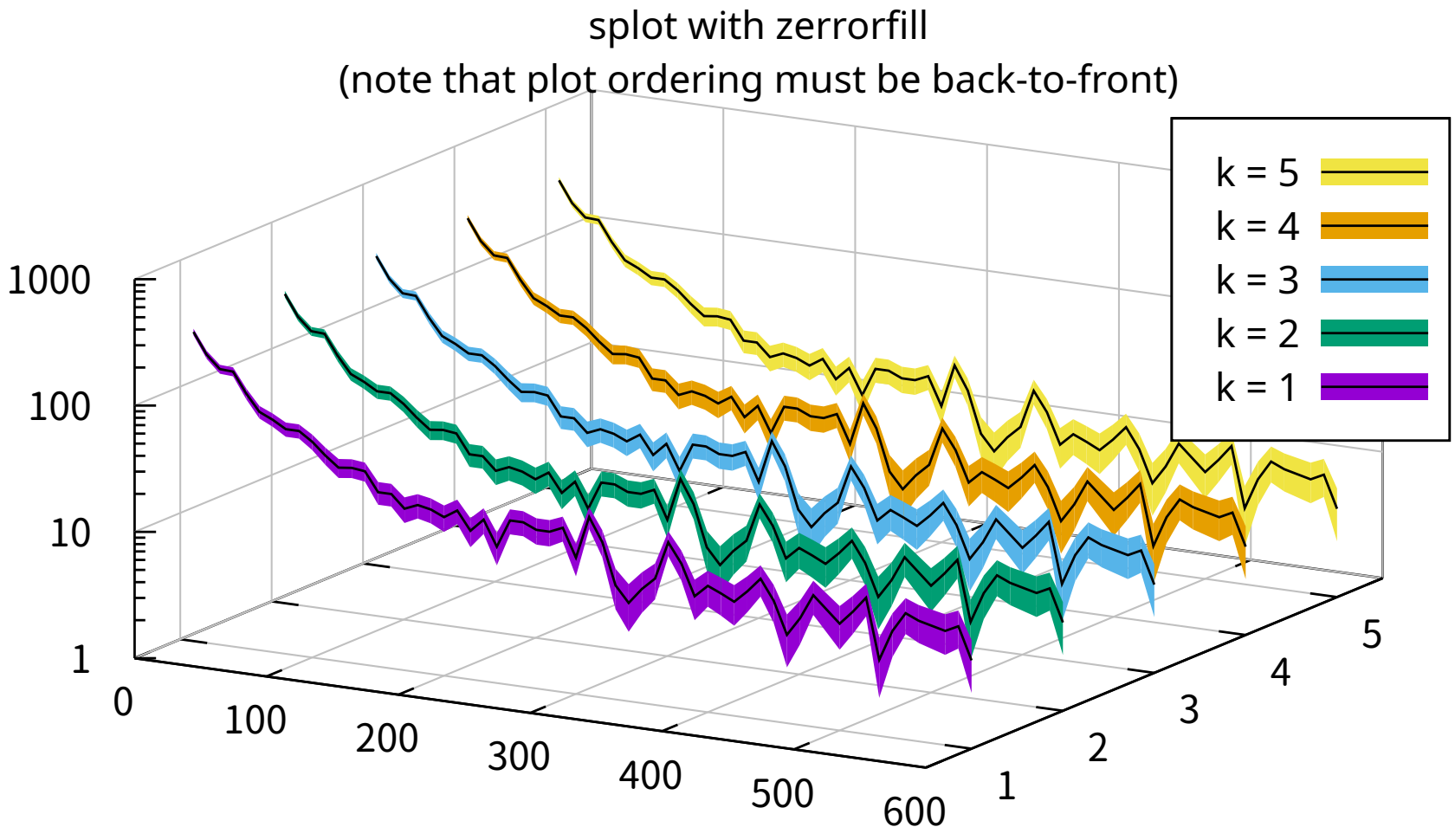

xyerrorbars in log scale
Power
Theory
Power [W]
1
0.1
0.01
1
5
25
Resistance [Ohm]


xyerrorbars with no crossbar
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

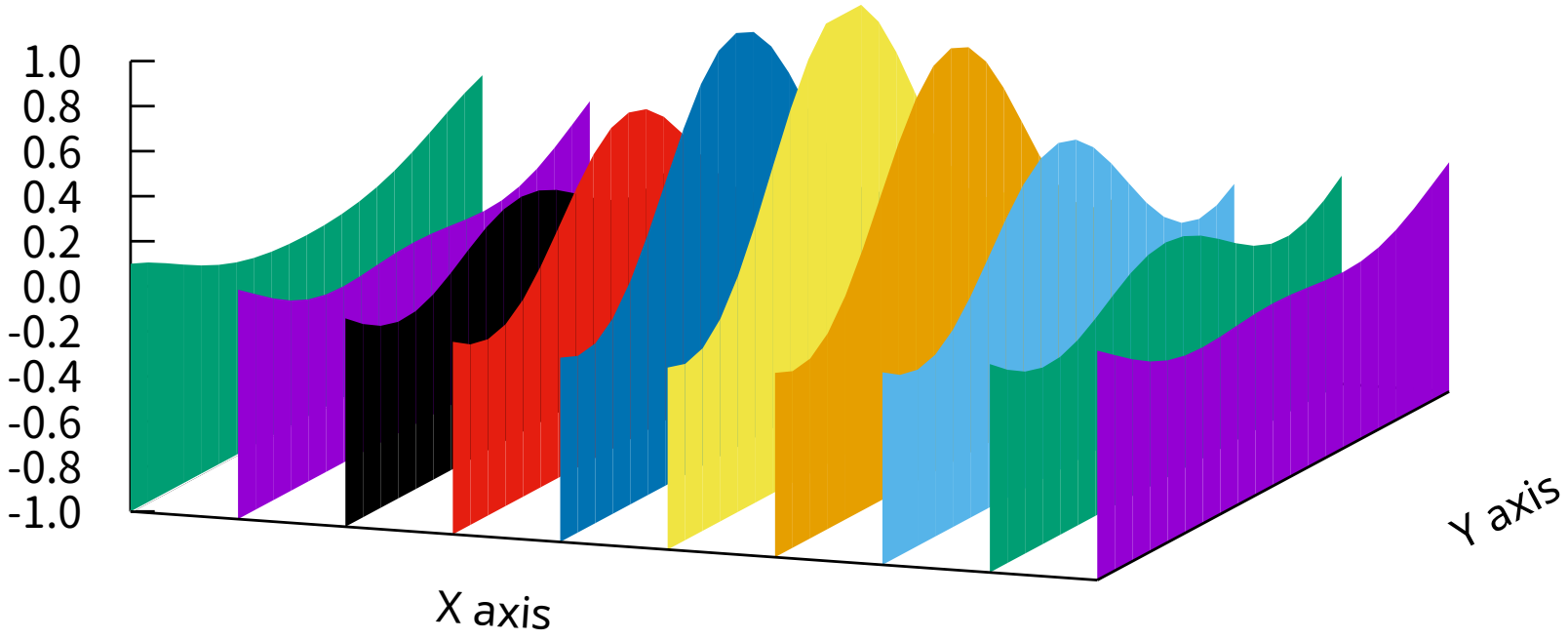

xerrorbars with no crossbar
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

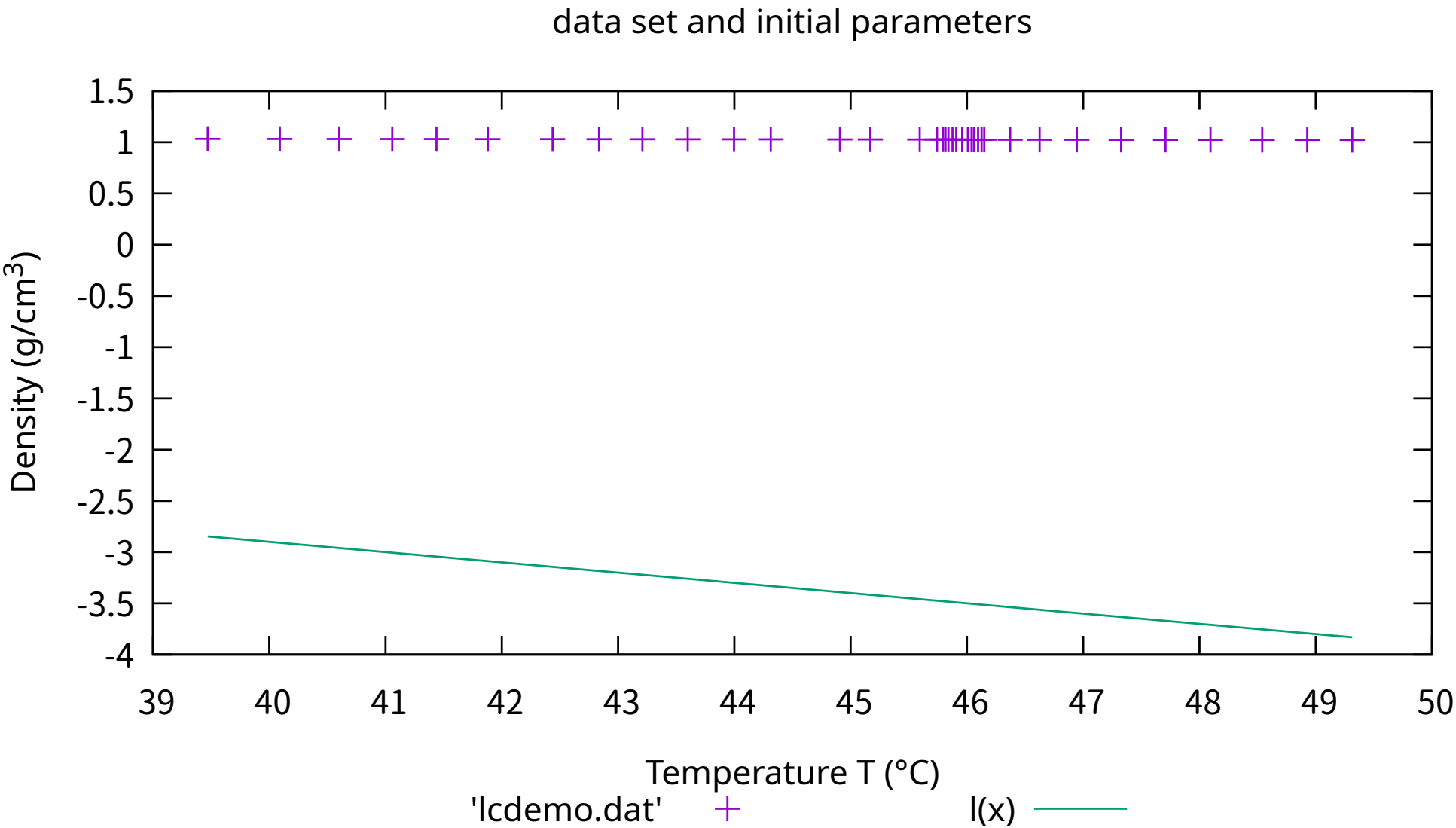

yerrorbars with no crossbar
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

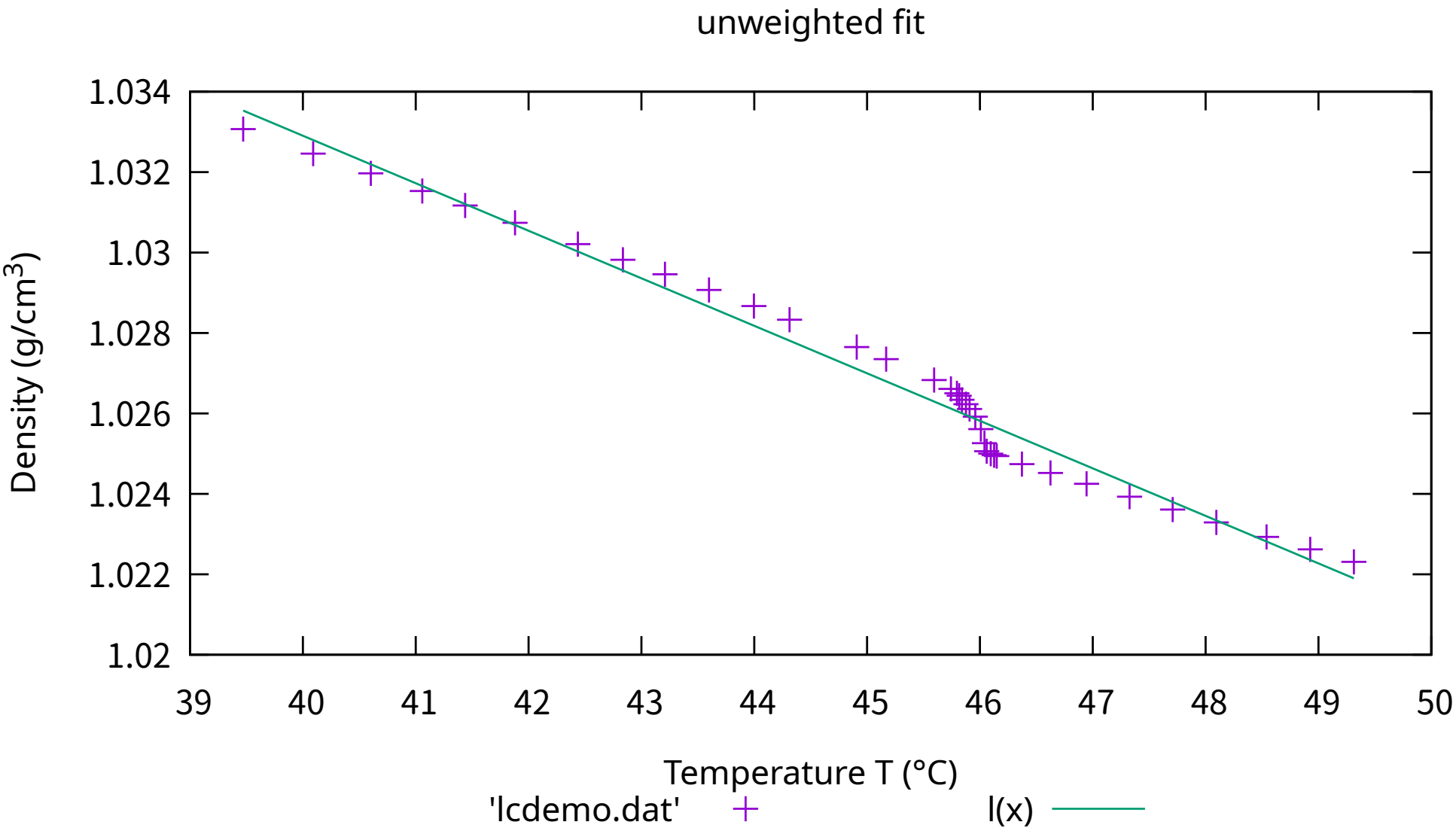

Error on y represented by filledcurve shaded region
Shaded error region
Monotonic spline through data
Rate
1000
100
10
1
0
100
200
300
400
500
600
Time (sec)

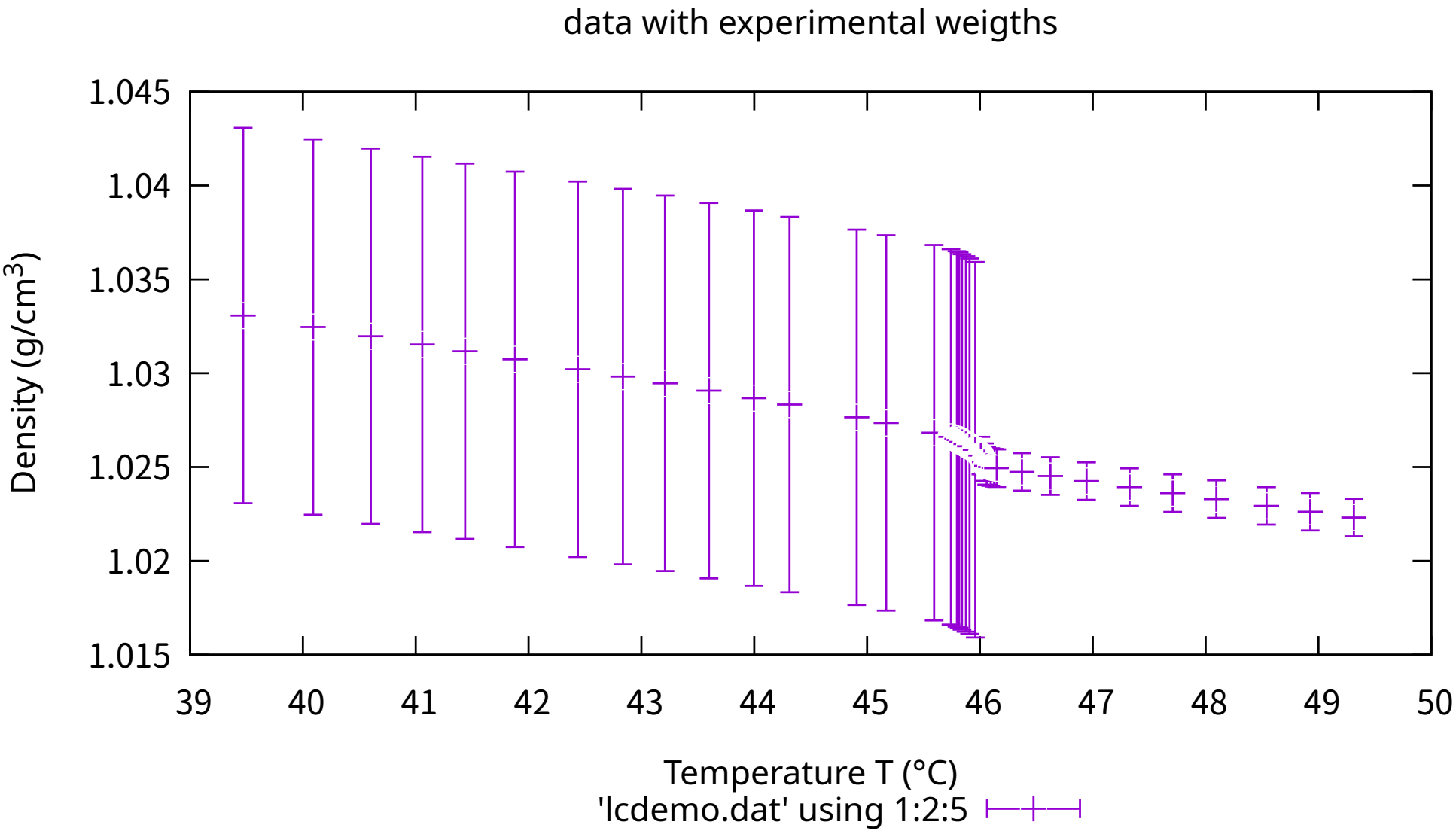

splot with zerrorfill
(note that plot ordering must be back-to-front)
k = 5
k = 4
k = 3
k = 2
k = 1
1000
100
10
1
0
100
200
300
400
500
600
1
2
3
4
5

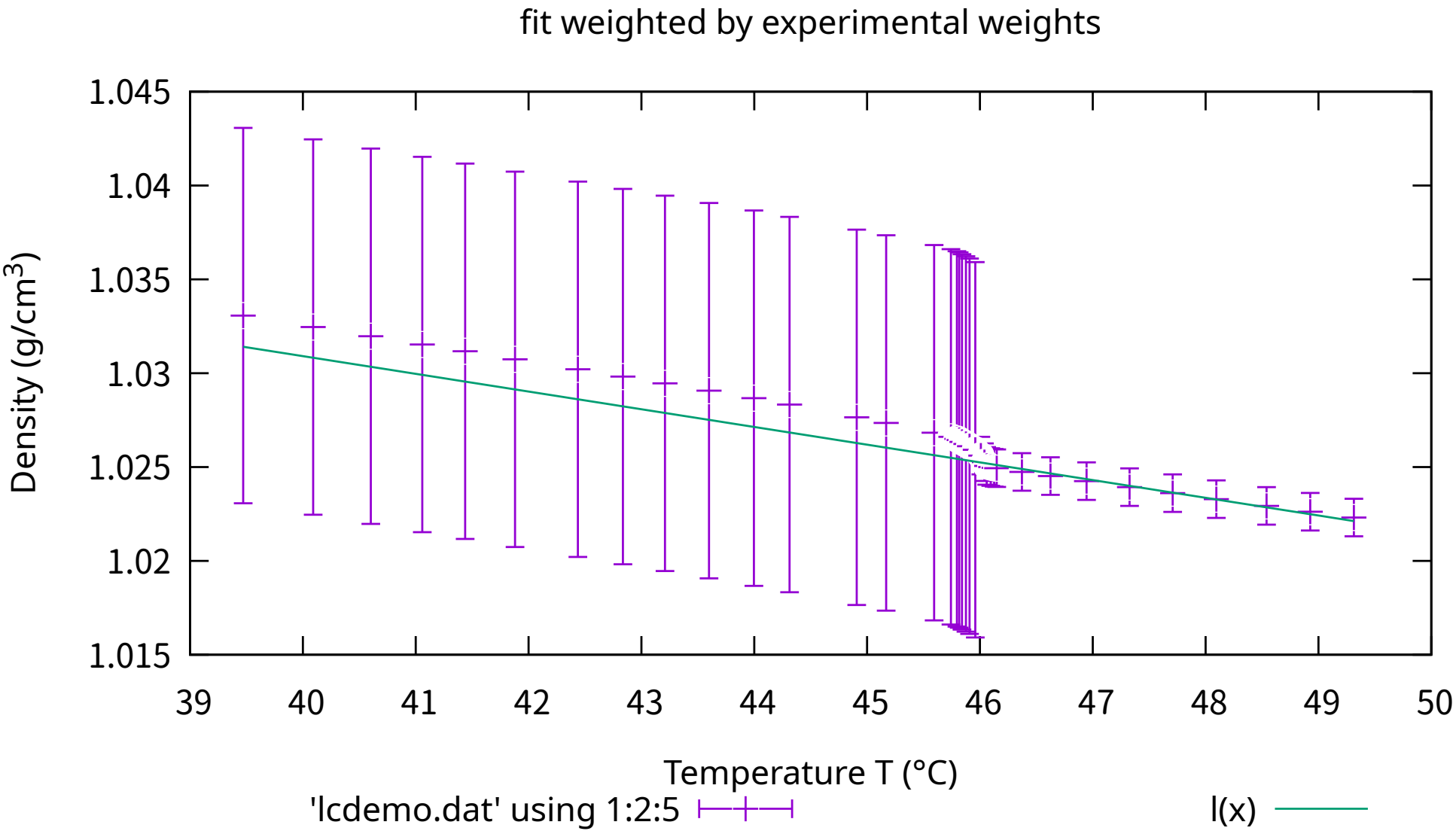
fence plot constructed with zerrorfill

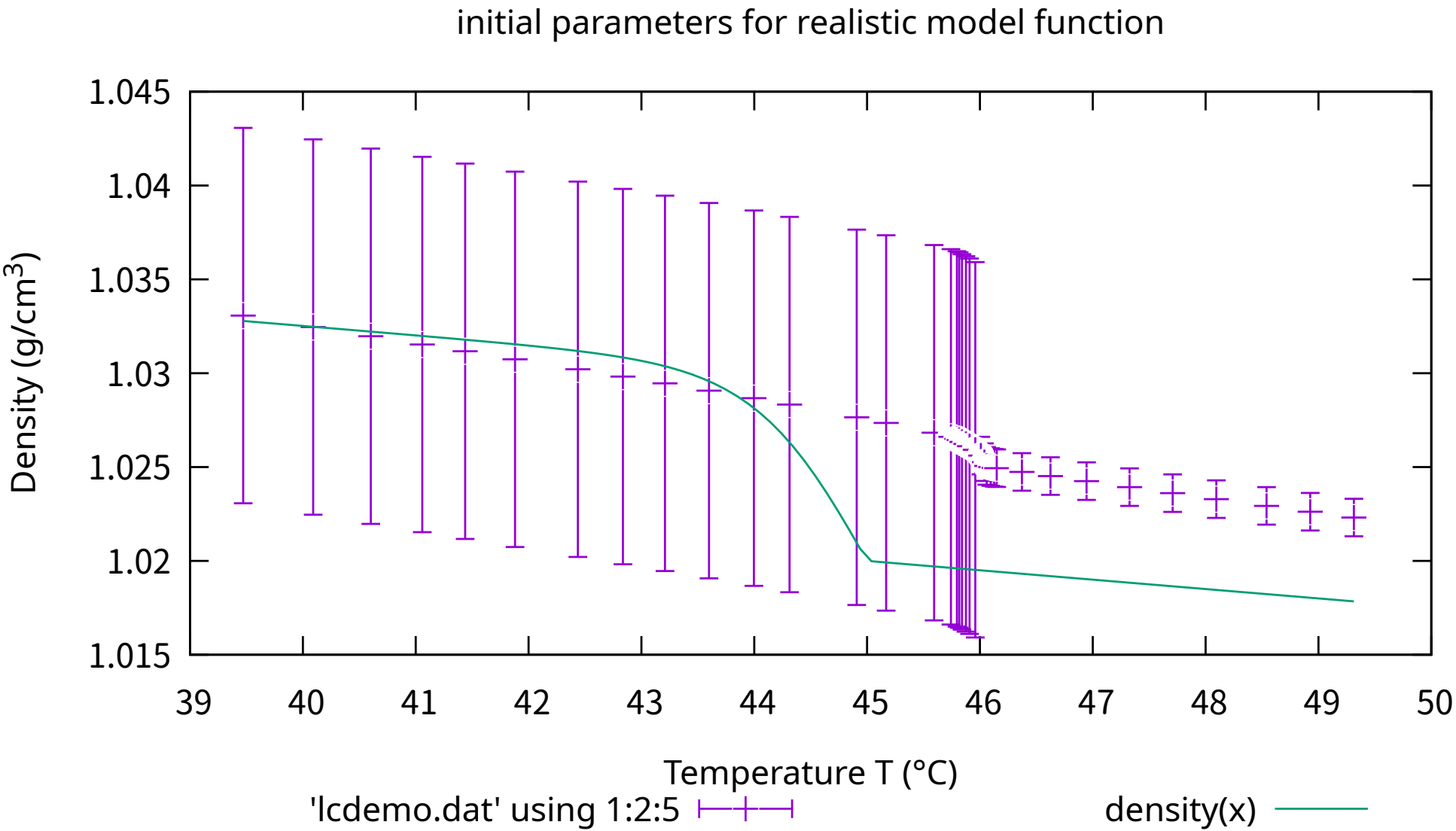


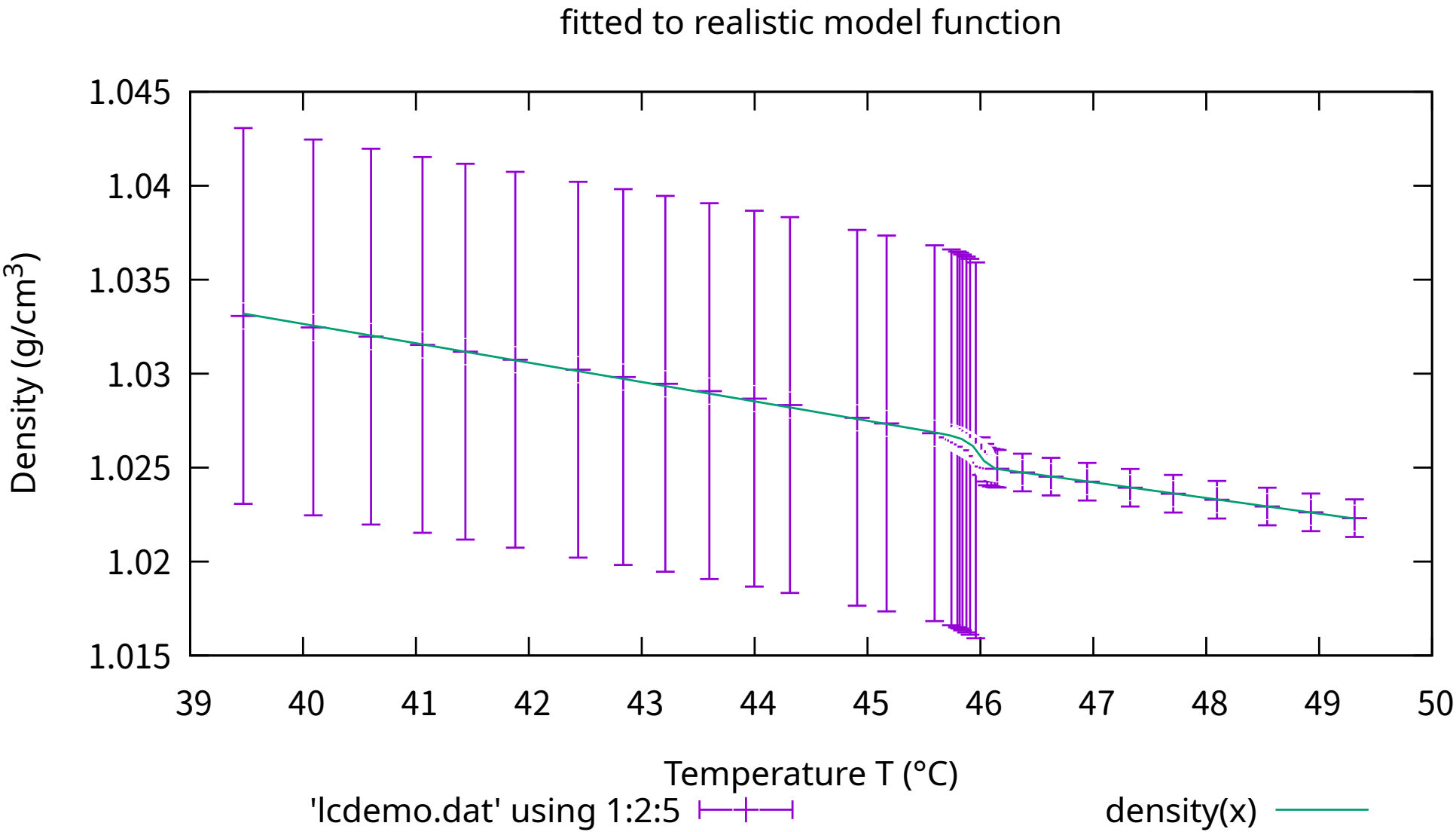

data set and initial parameters
Density (g/cm^3)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T (°C)
'lcdemo.dat'
l(x)


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 (°C)
'lcdemo.dat'
l(x)

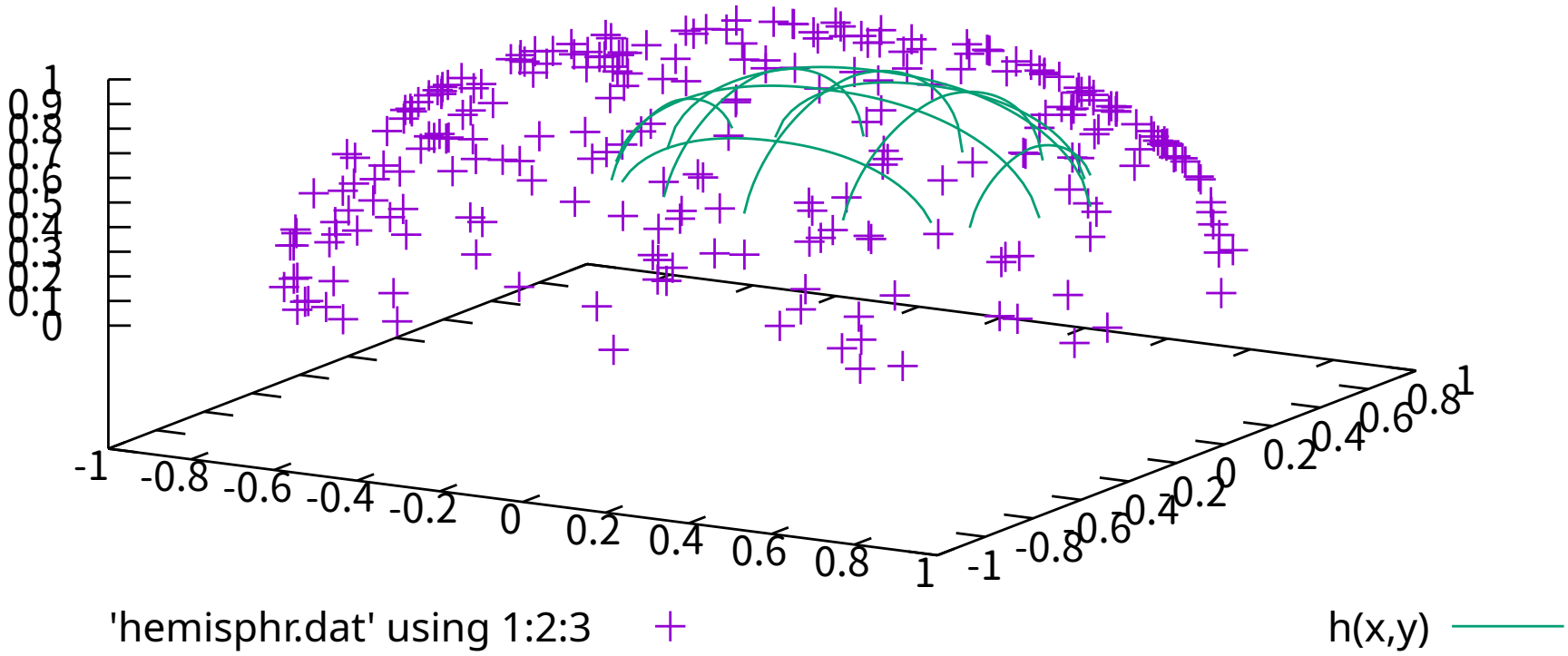

data with experimental weigths
1.045
1.04
1.035
1.03
1.025
1.02
1.015
Density (g/cm^3)
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T (°C)
'lcdemo.dat' using 1:2:5


fit weighted by experimental weights
1.045
1.04
1.035
1.03
1.025
1.02
1.015
Density (g/cm^3)
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T (°C)
'lcdemo.dat' using 1:2:5
l(x)

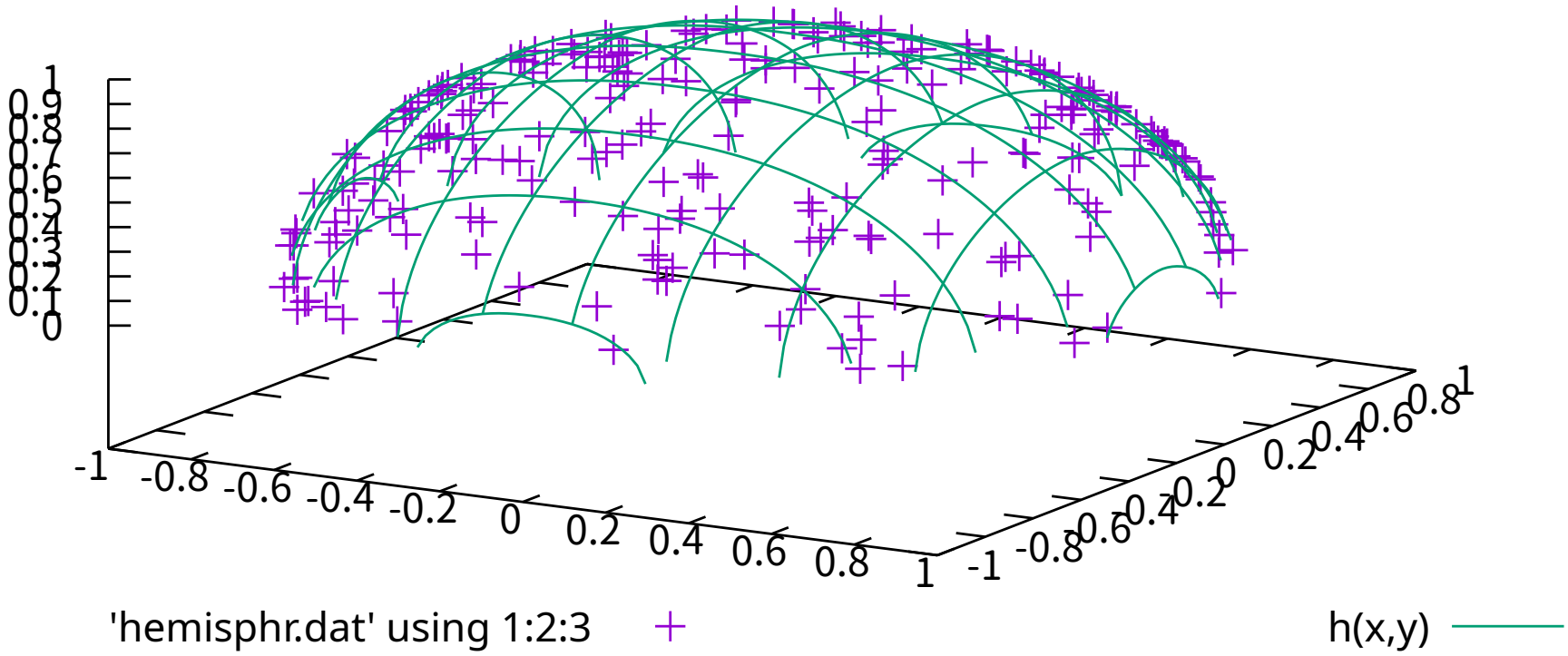

initial parameters for realistic model function
1.045
1.04
1.035
1.03
1.025
1.02
1.015
Density (g/cm3)
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T (°C)
'lcdemo.dat' using 1:2:5
density(x)

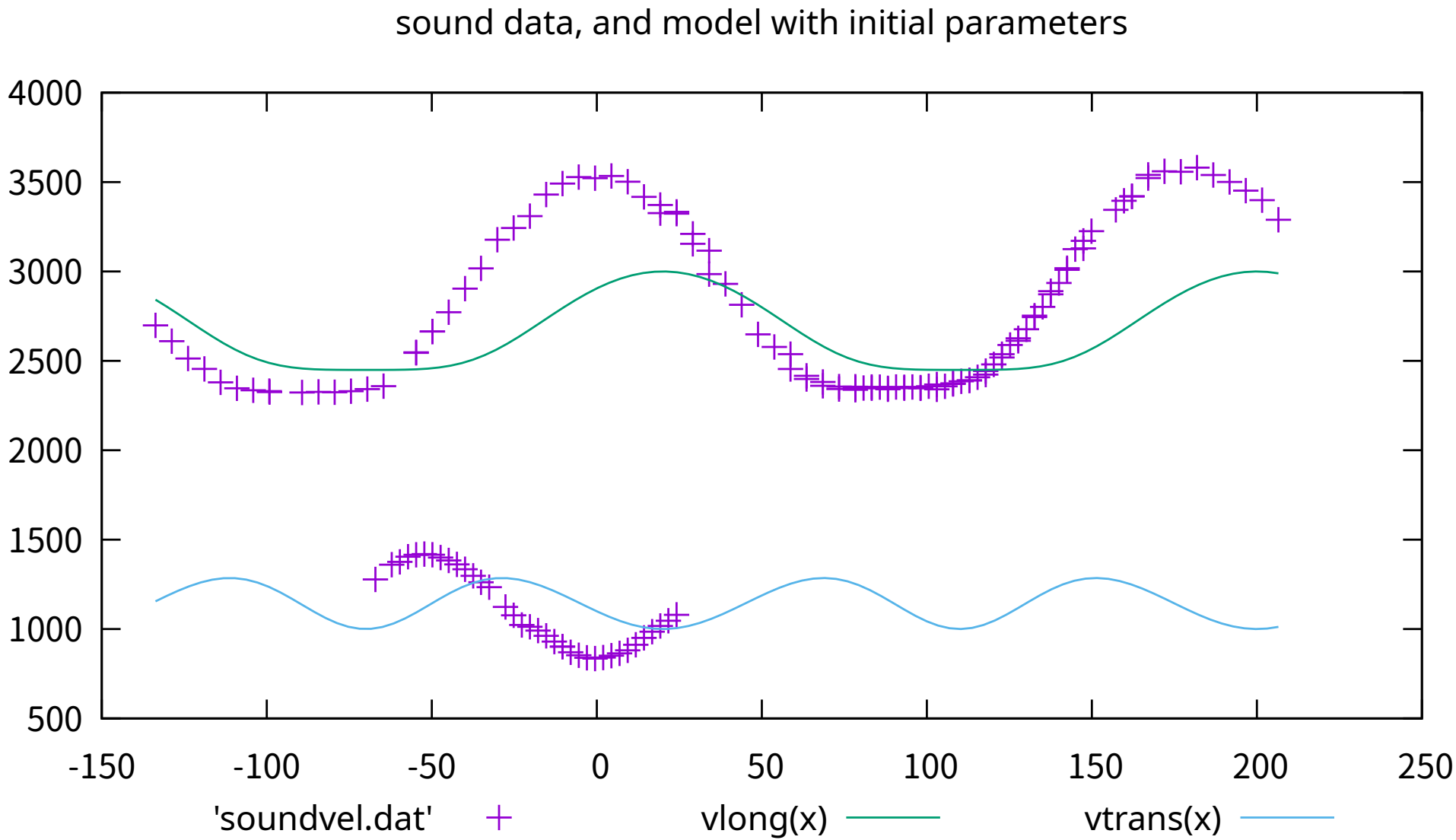

fitted to realistic model function
1.045
1.04
1.035
1.03
1.025
1.02
1.015
Density (g/cm^3)
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T (°C)
'lcdemo.dat' using 1:2:5
density(x)

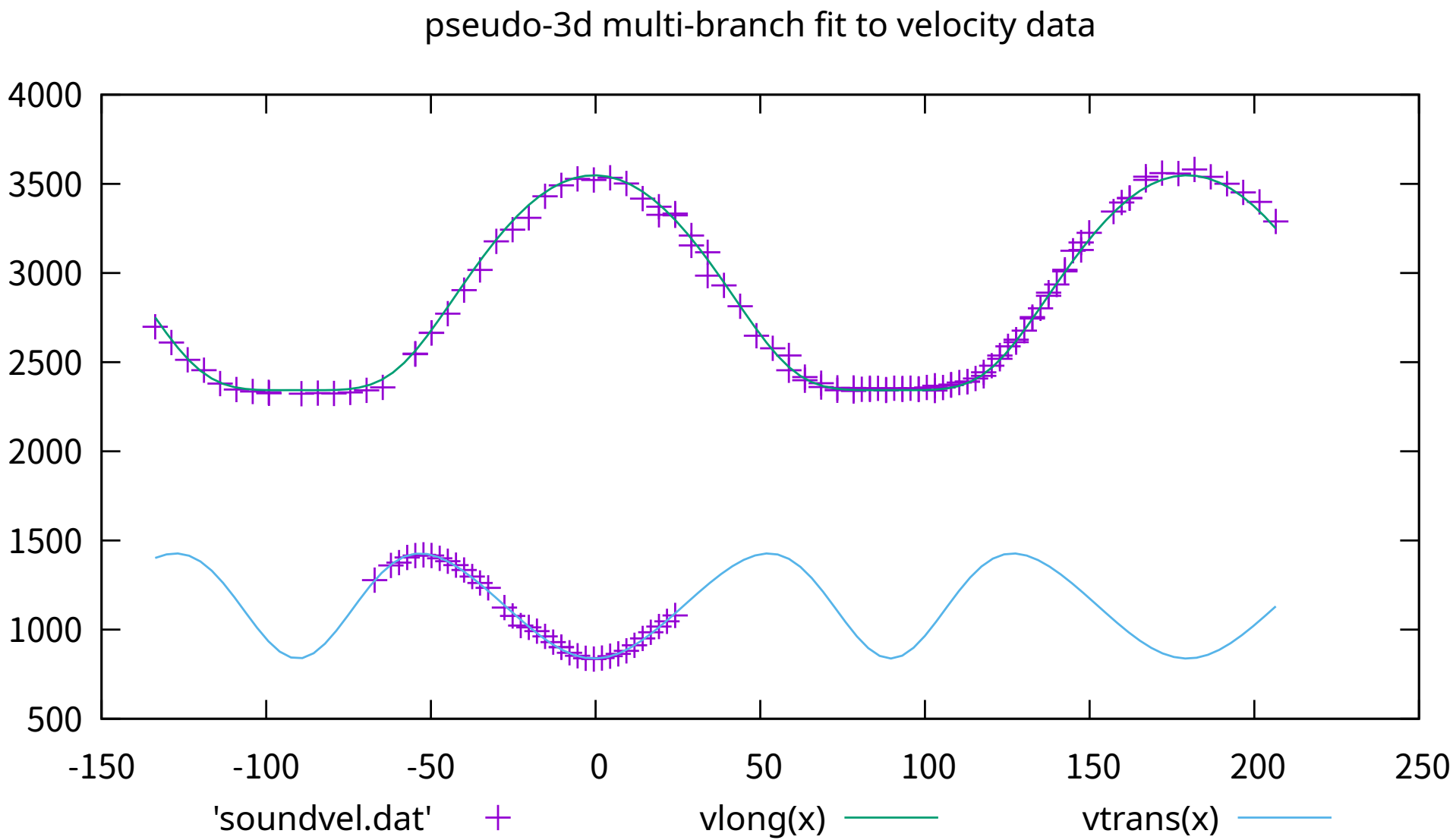
the scattered points, and the initial parameter

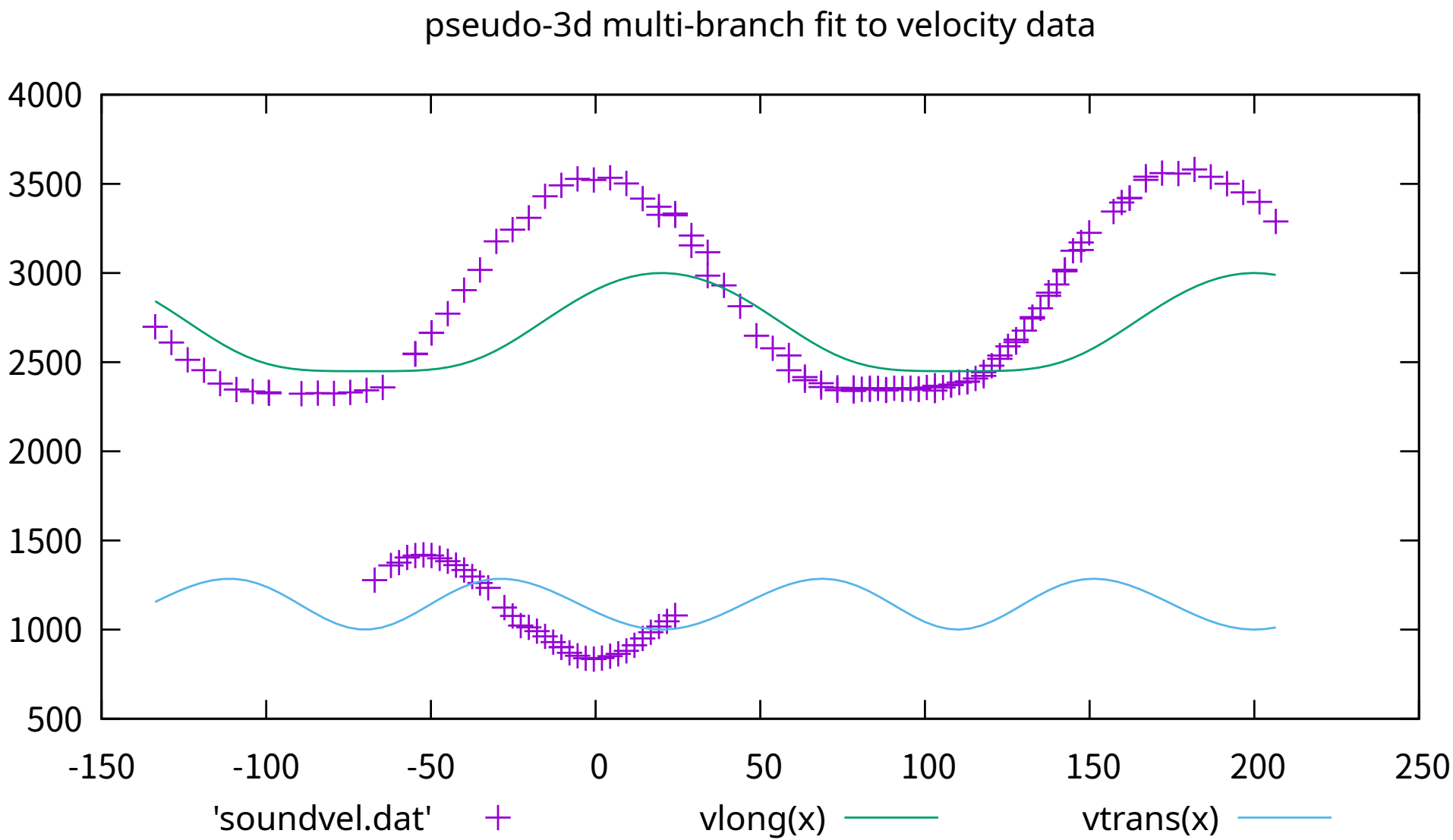


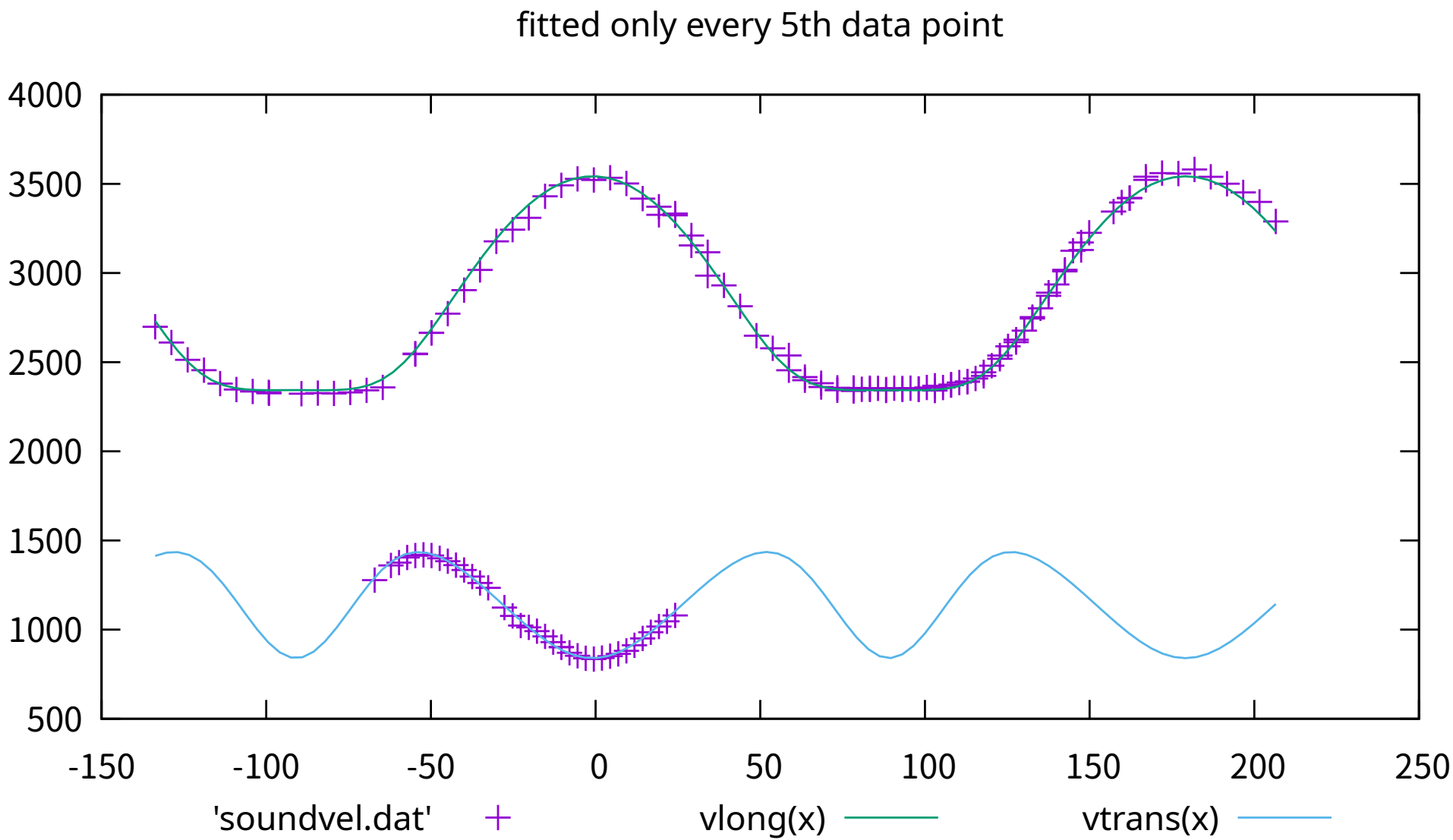
the scattered points, fitted curve

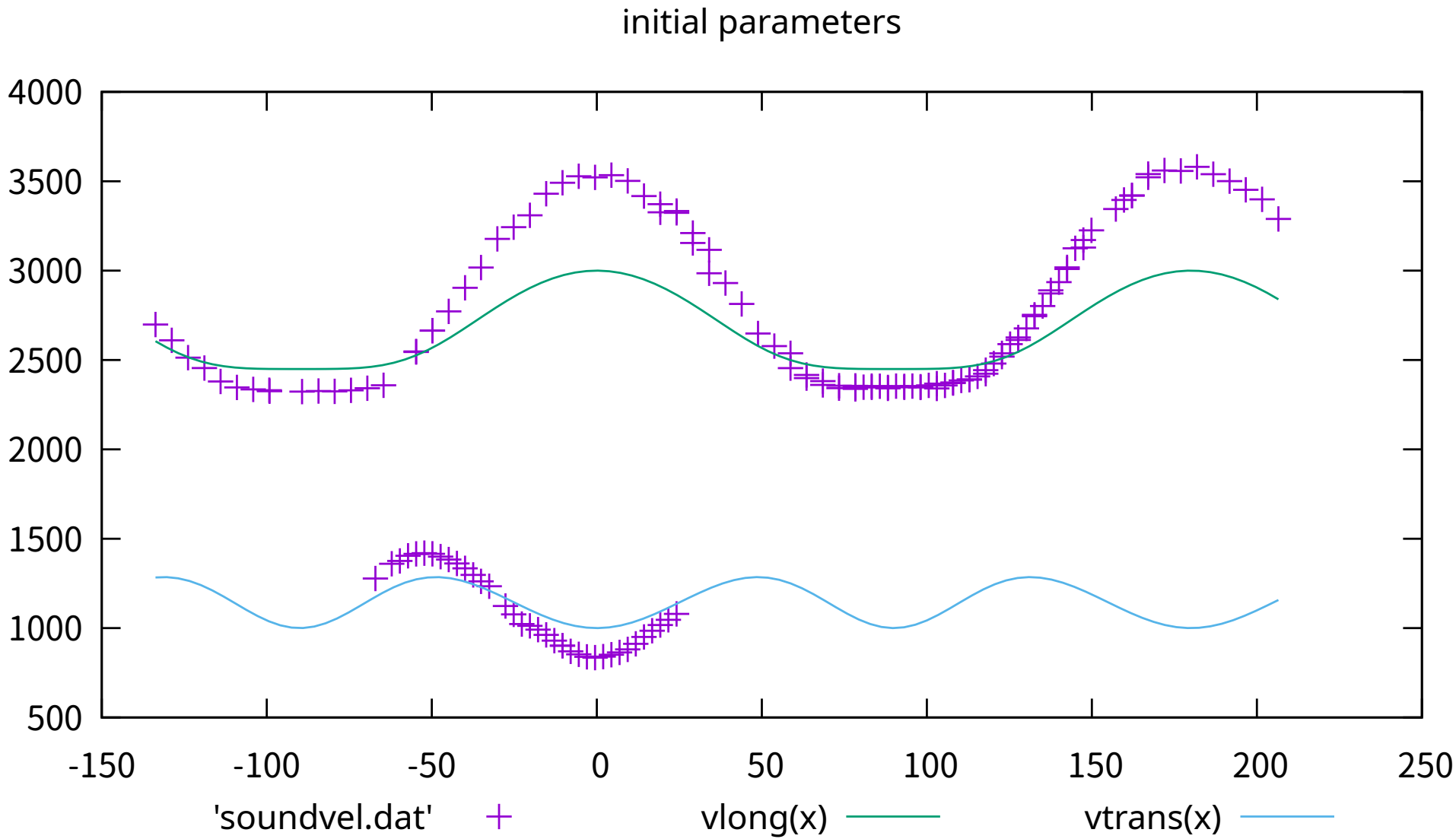


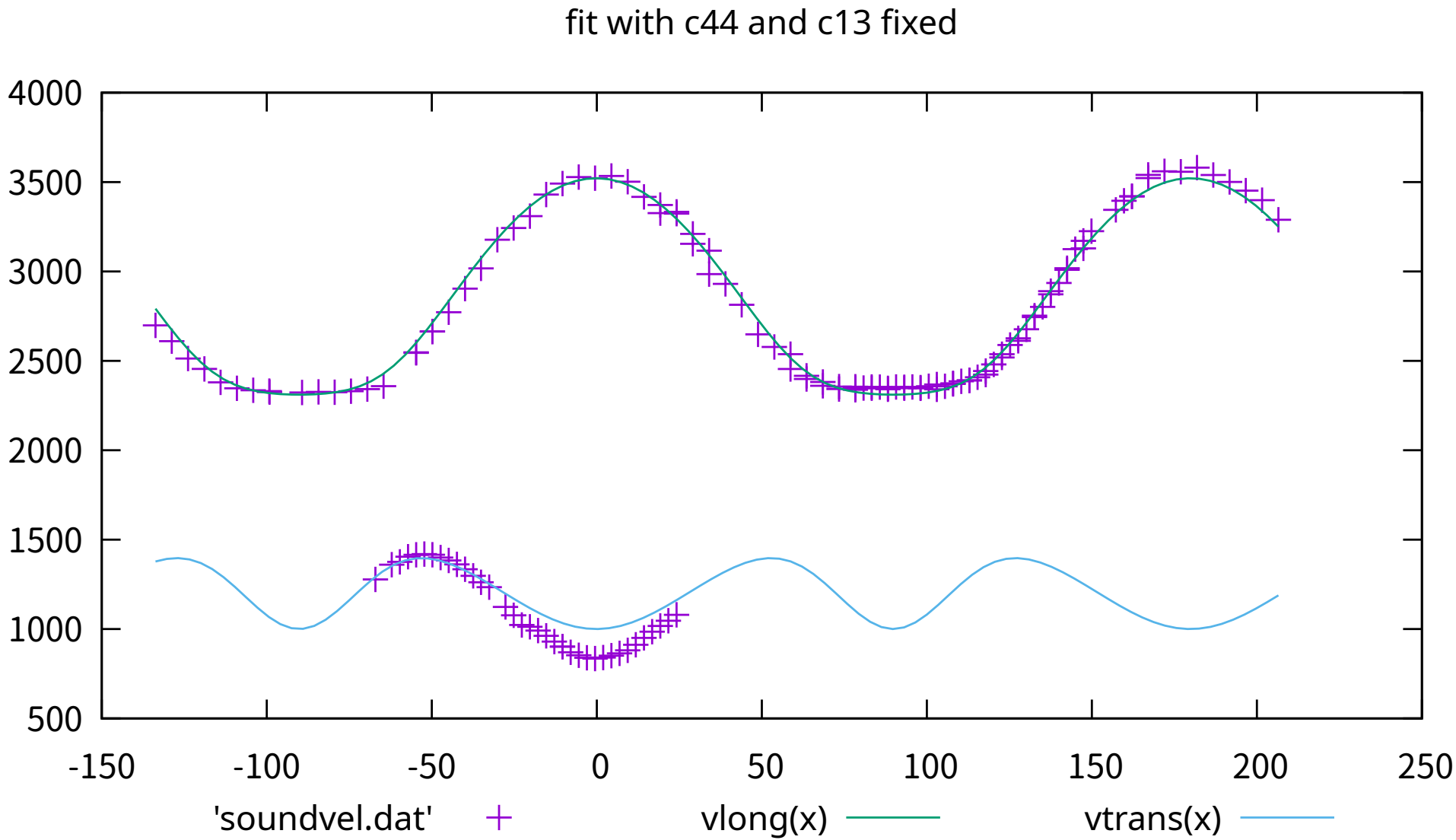

sound data, and model with initial parameters
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
'soundvel.dat'
vlong(x)
vtrans(x)

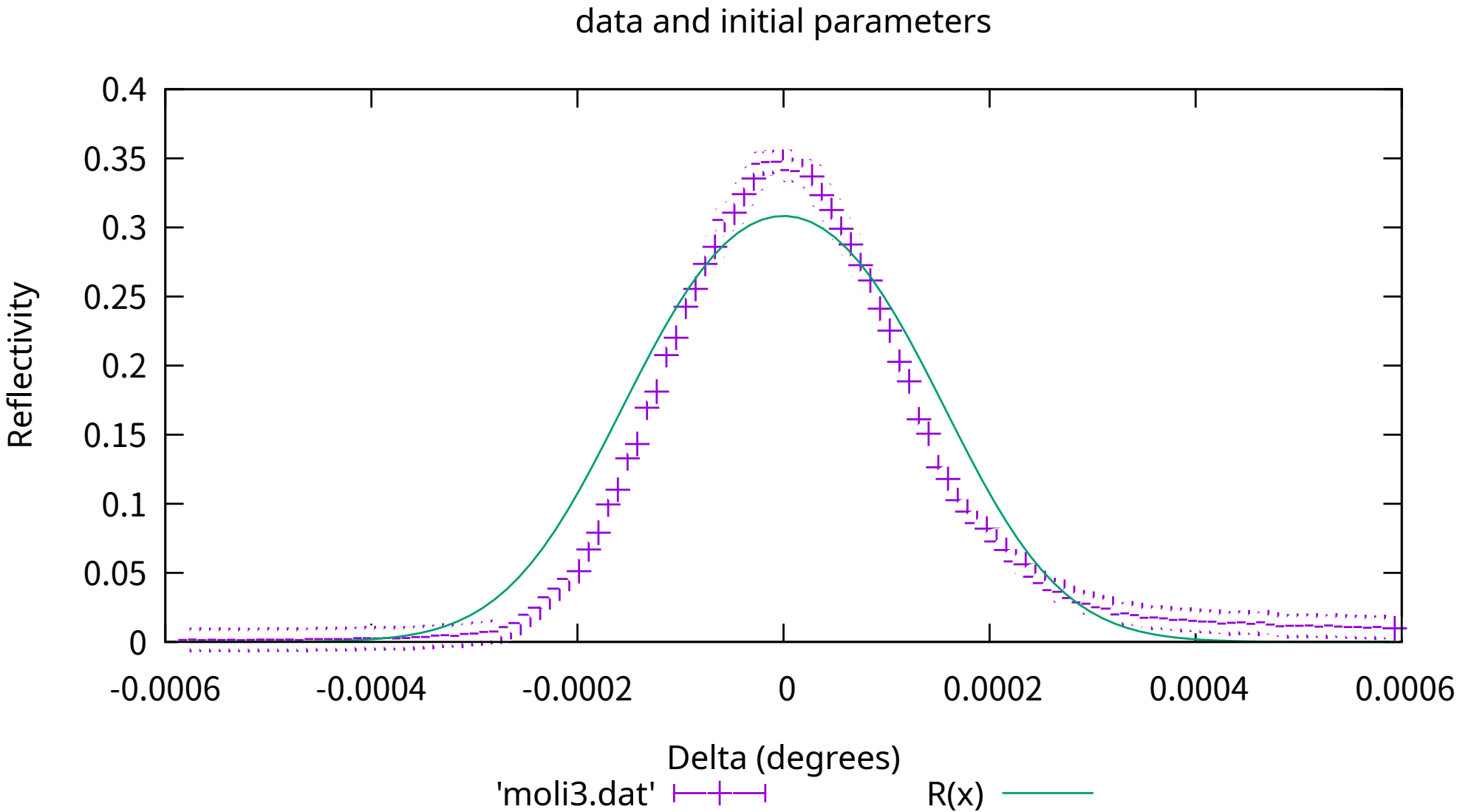

pseudo-3d multi-branch fit to velocity data
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
'soundvel.dat'
vlong(x)
vtrans(x)

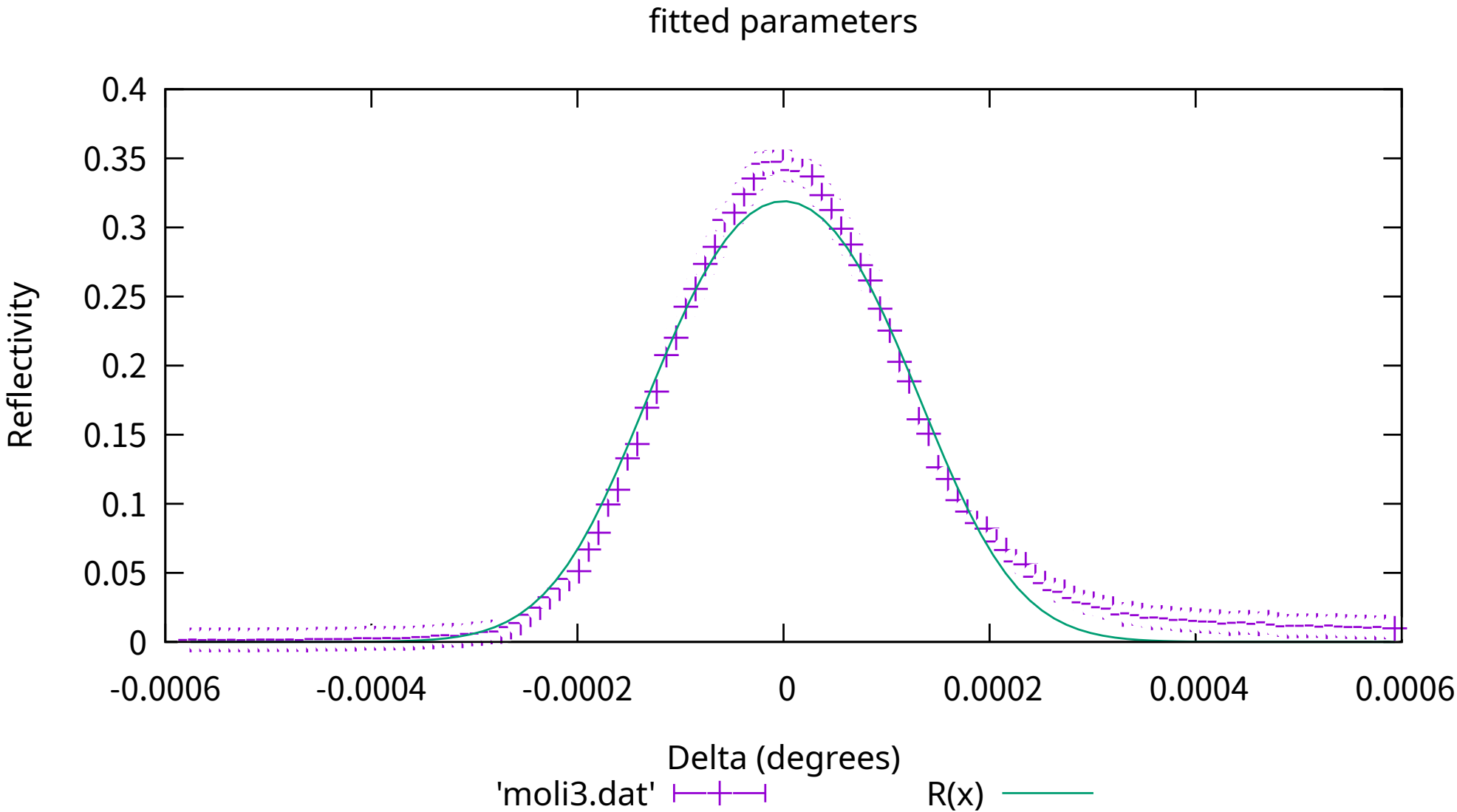

pseudo-3d multi-branch fit to velocity data
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
'soundvel.dat'
vlong(x)
vtrans(x)

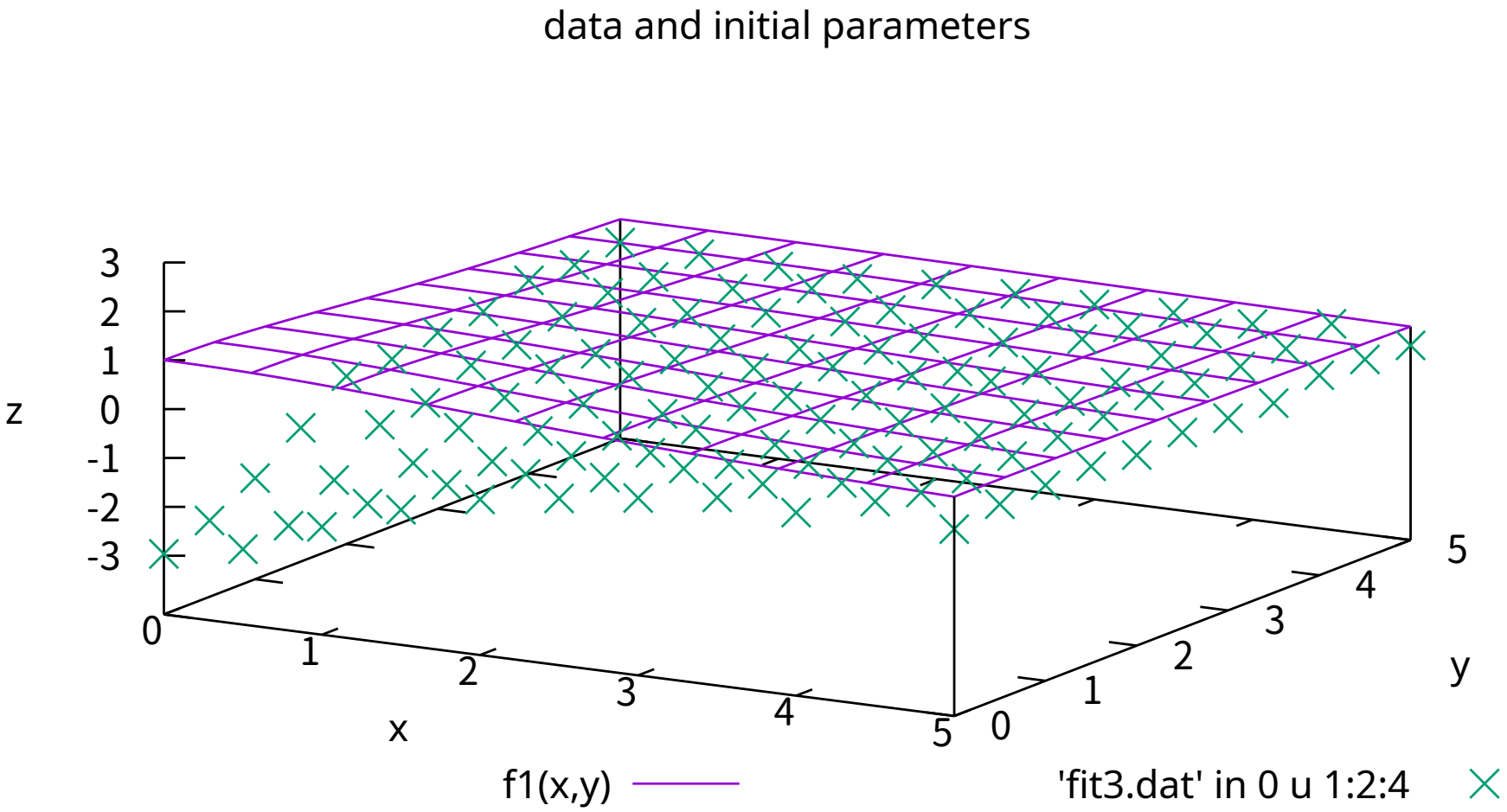

fitted only every 5th data point
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
'soundvel.dat'
vlong(x)
vtrans(x)

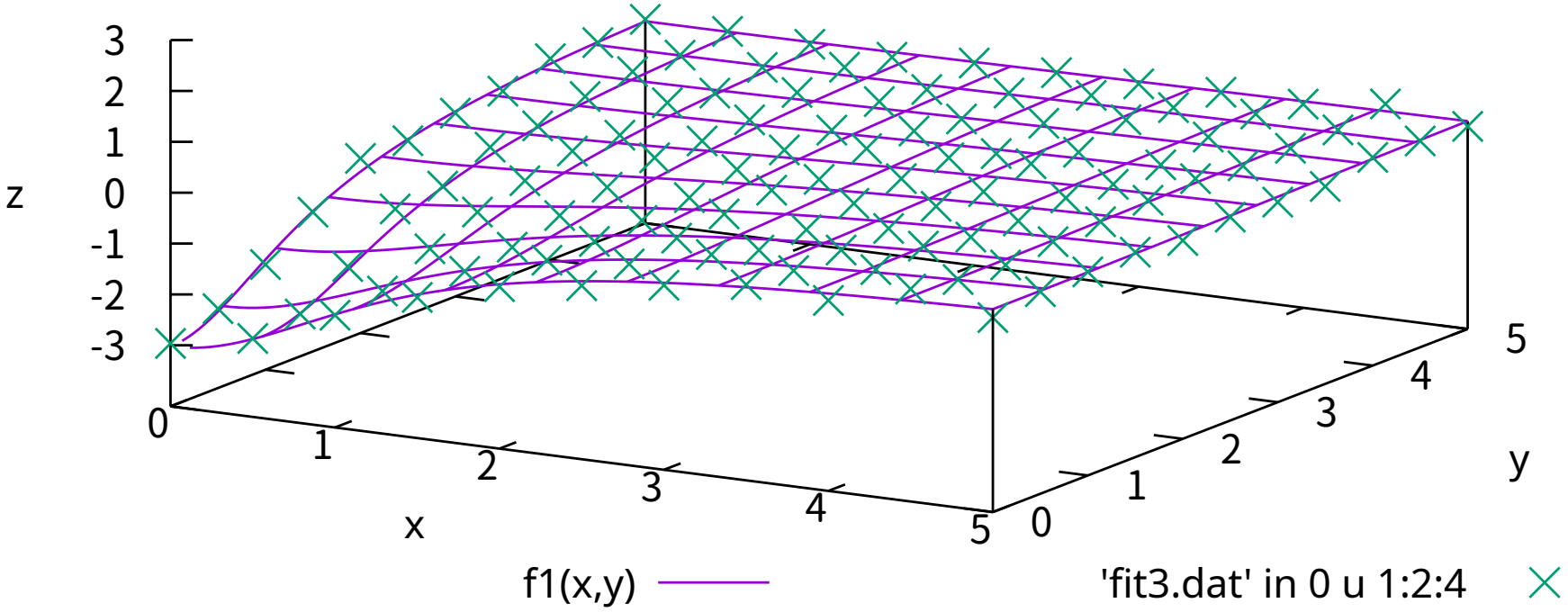

initial parameters
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
'soundvel.dat'
vlong(x)
vtrans(x)


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)

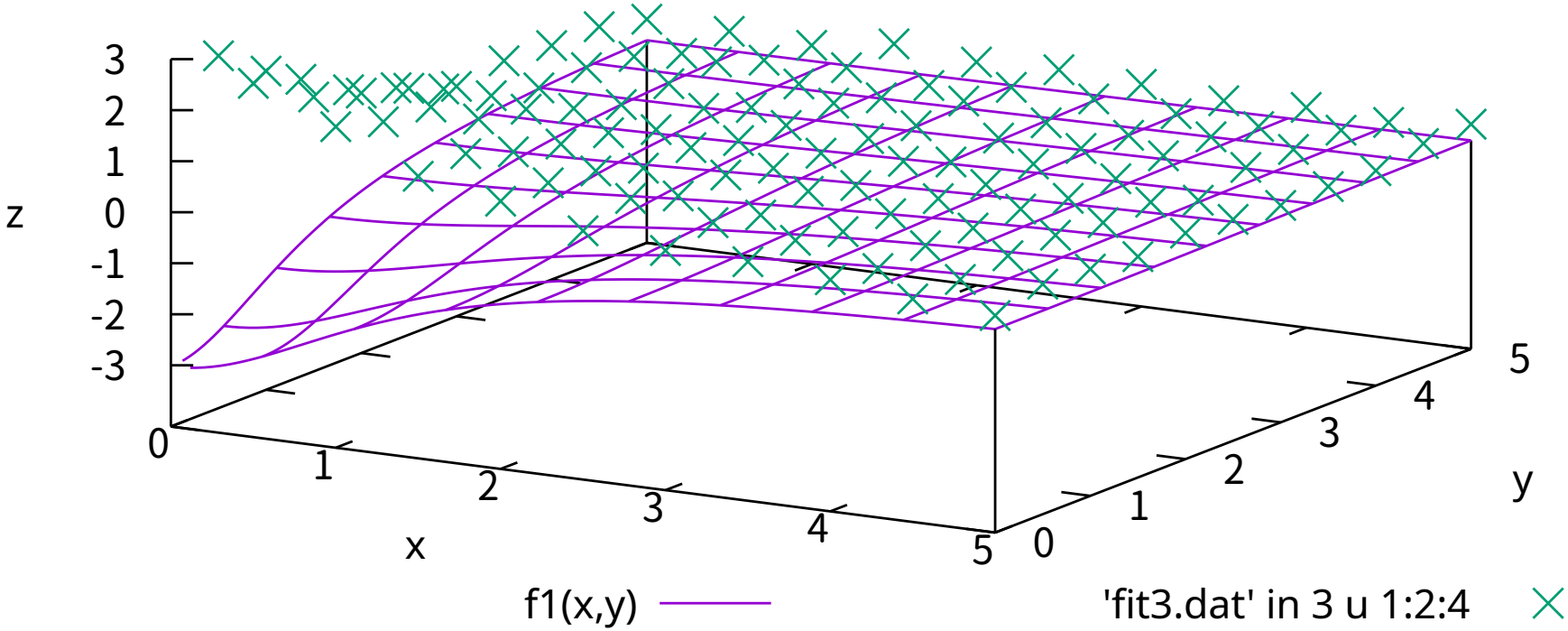

data and initial parameters
Reflectivity
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.0006
-0.0004
-0.0002
0
0.0002
0.0004
0.0006
Delta (degrees)
'moli3.dat'
R(x)


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)

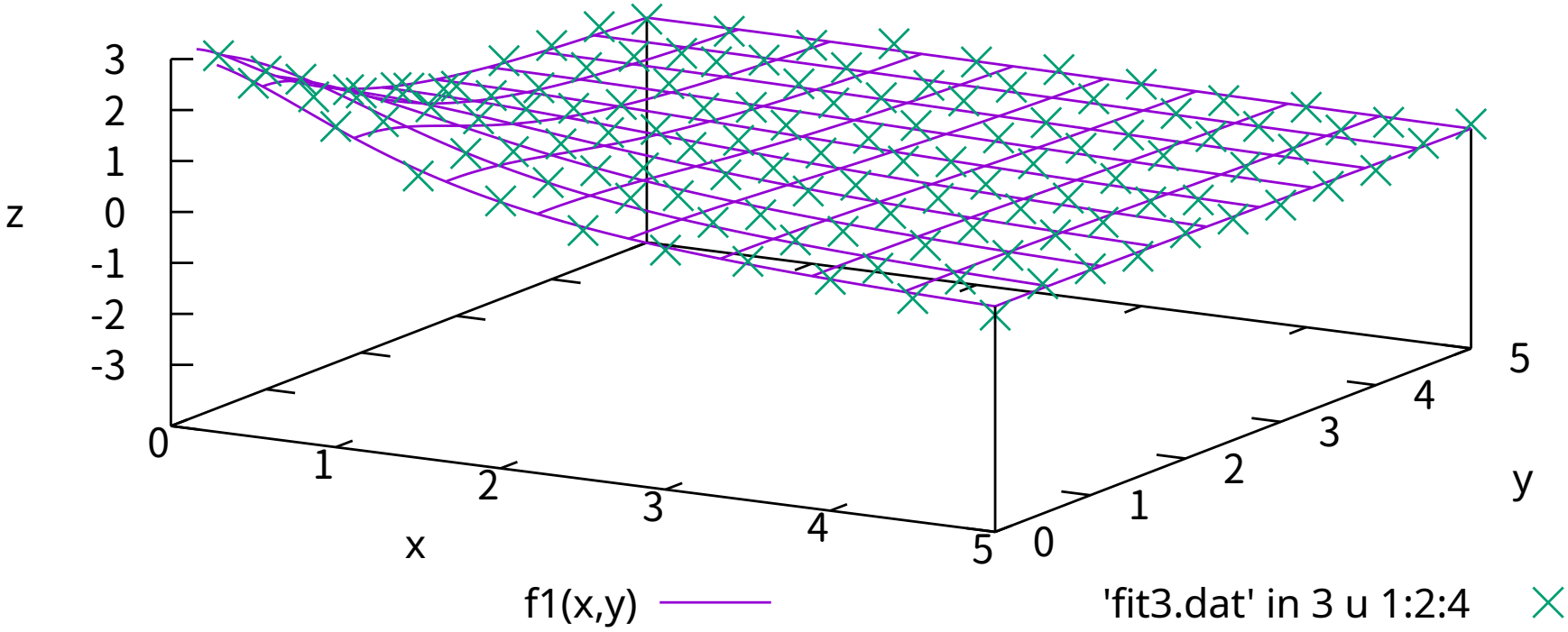

data and initial parameters
z
3
2
1
0
-1
-2
-3
0
1
2
3
4
5
x
0
1
2
3
4
5
y
f1(x,y)
'fit3.dat' in 0 u 1:2:4

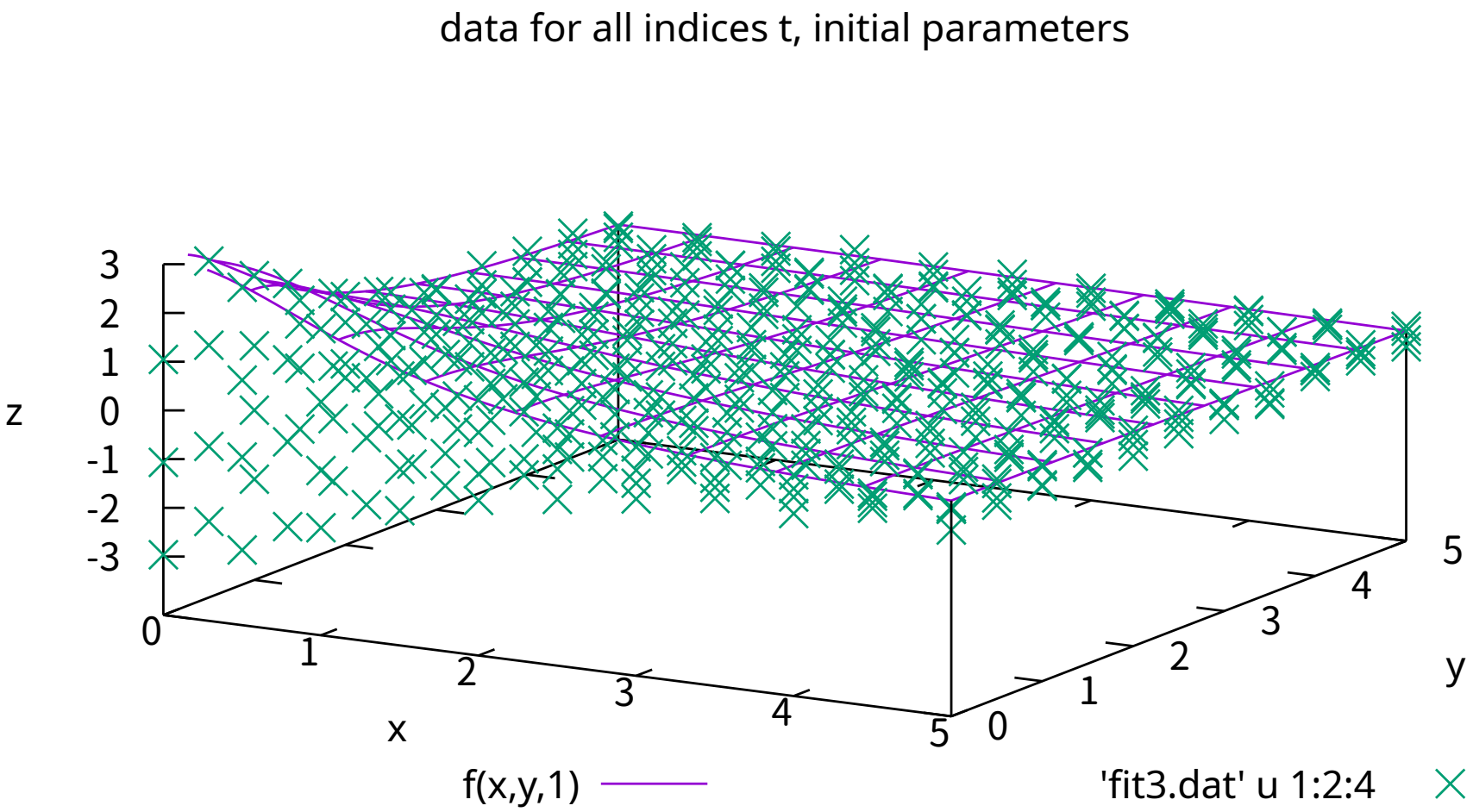

fit to data with t = -3
z
3
2
1
0
-1
-2
-3
0
1
2
3
4
5
x
0
1
2
3
4
5
y
f1(x,y)
'fit3.dat' in 0 u 1:2:4

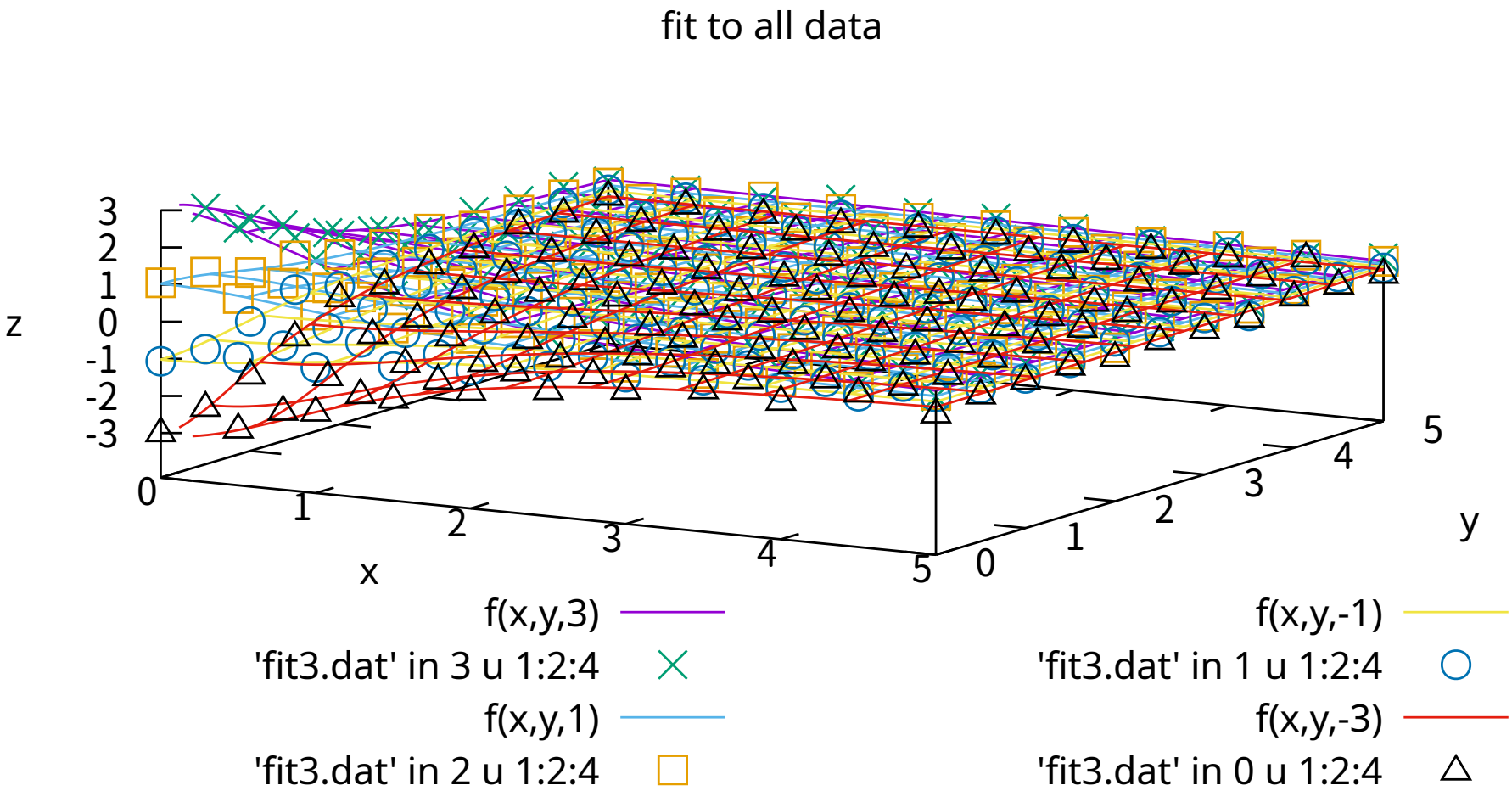
fit to data with t = +3, initial parameters

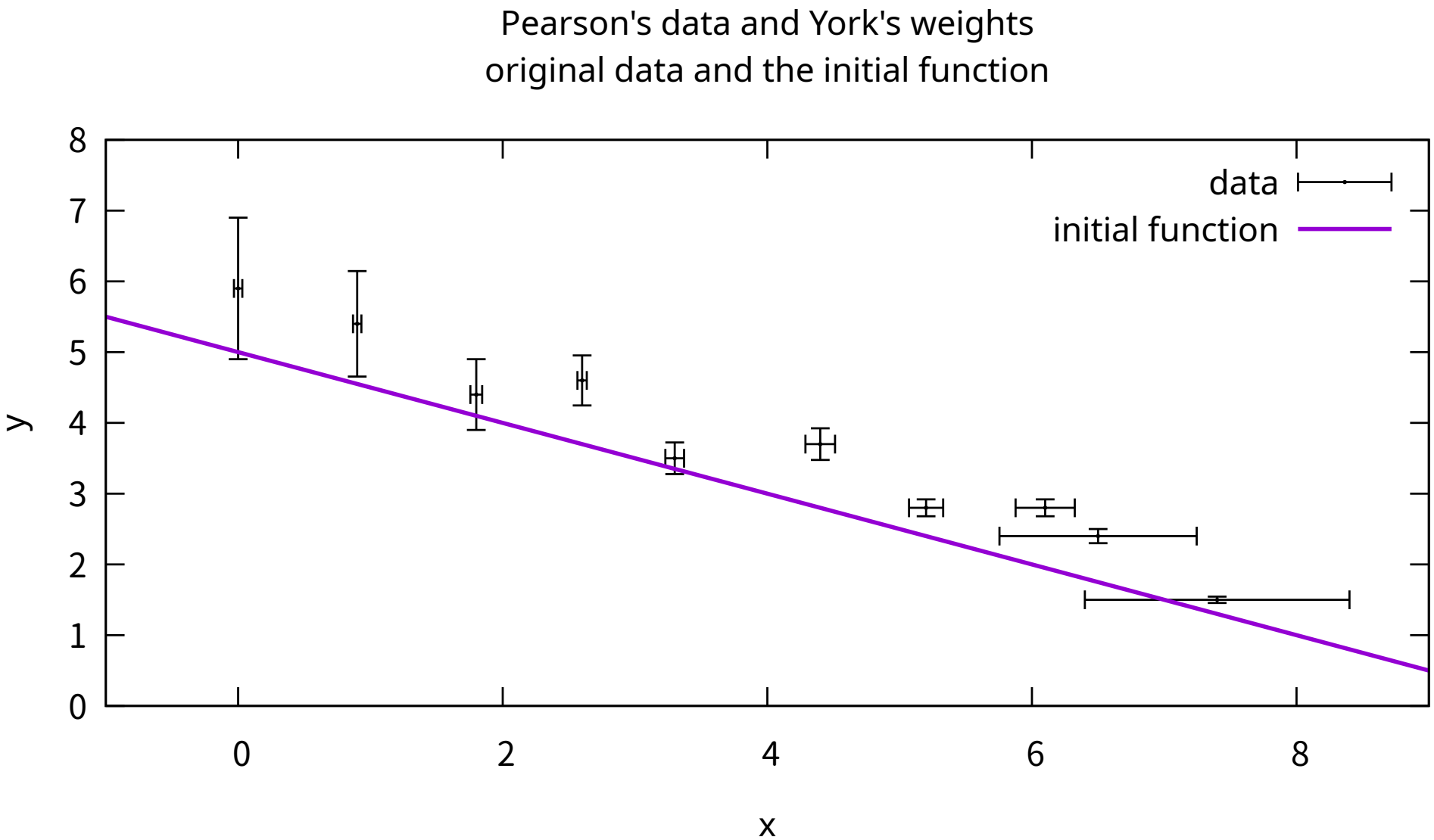


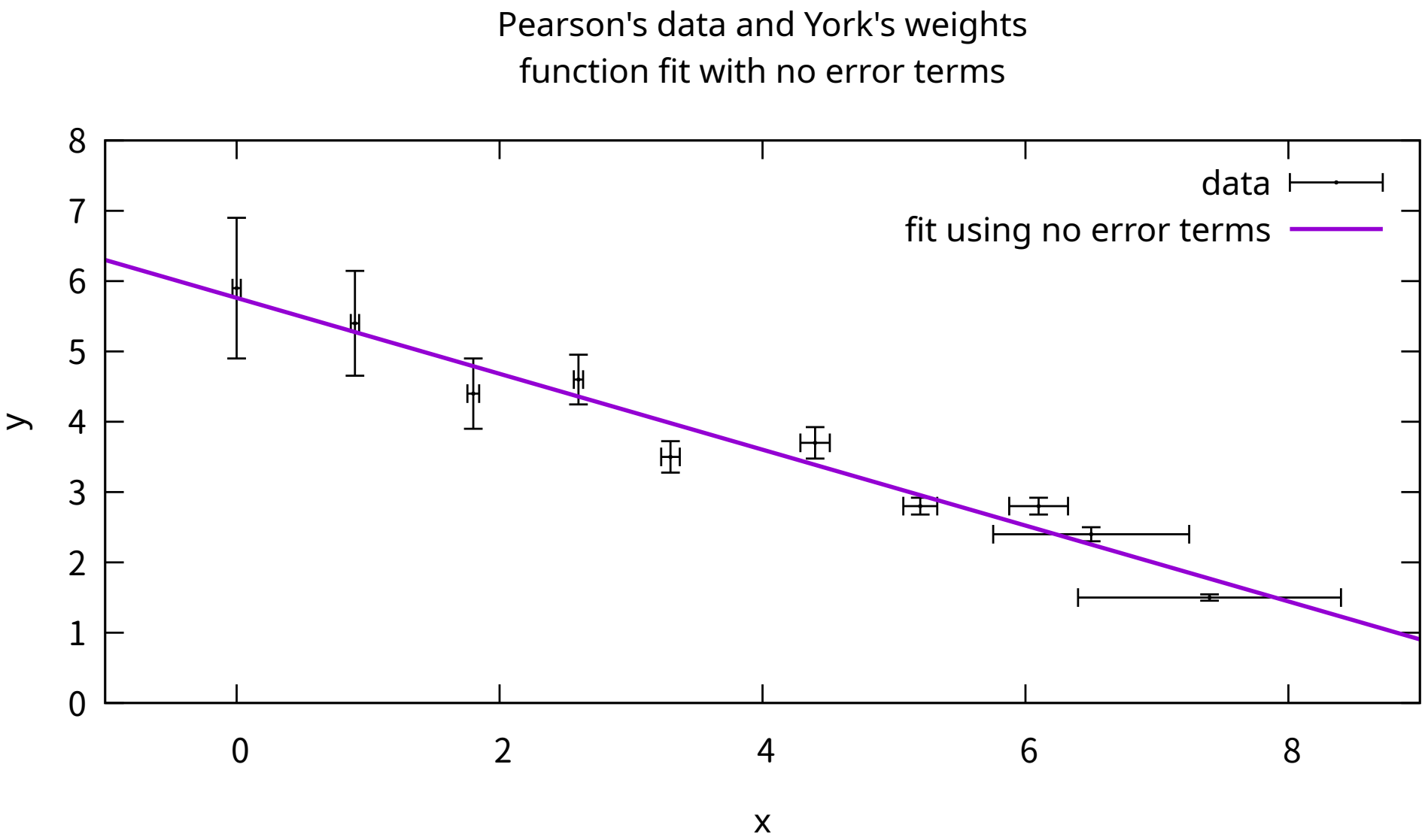
fit to data with t = +3

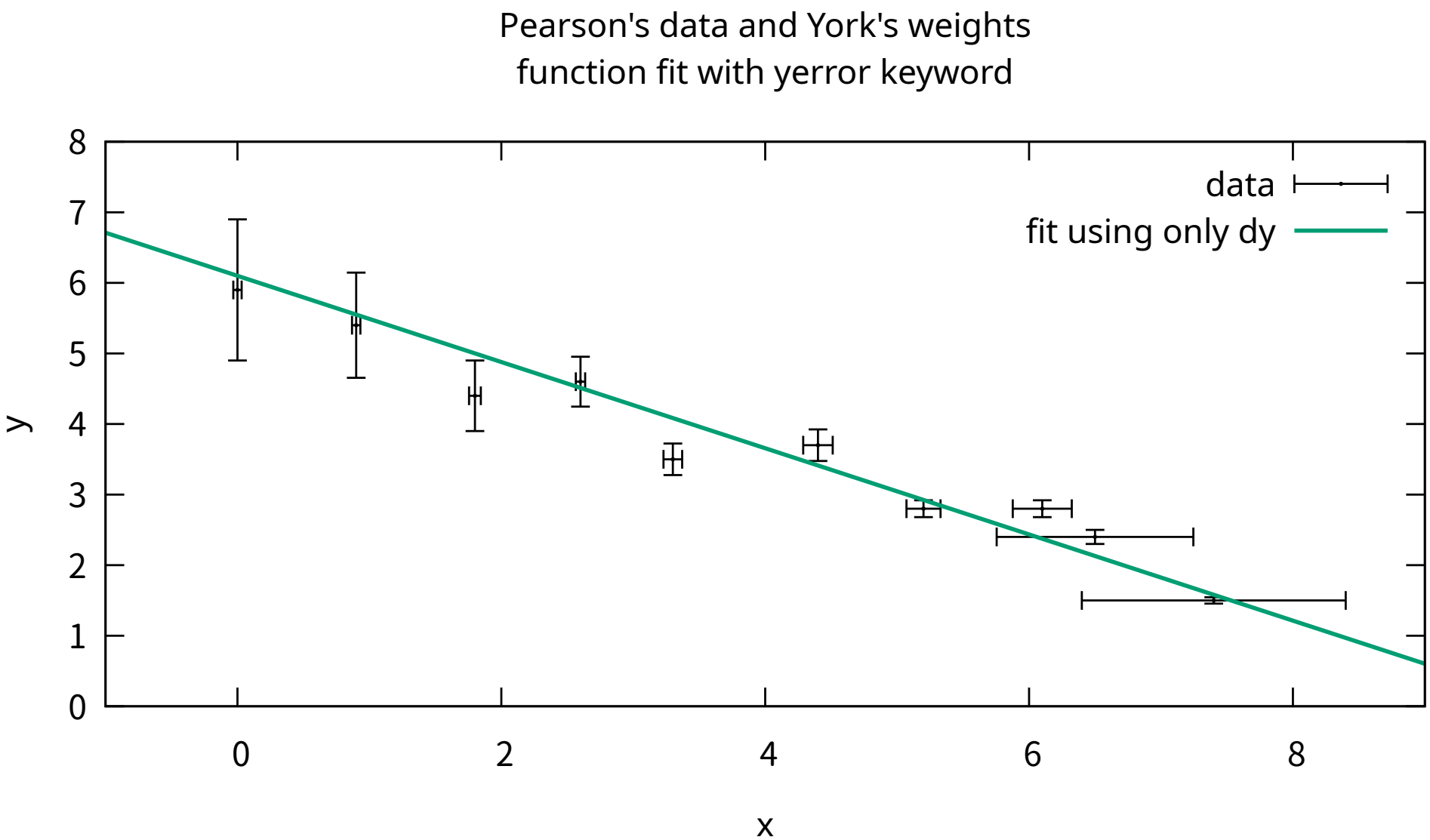


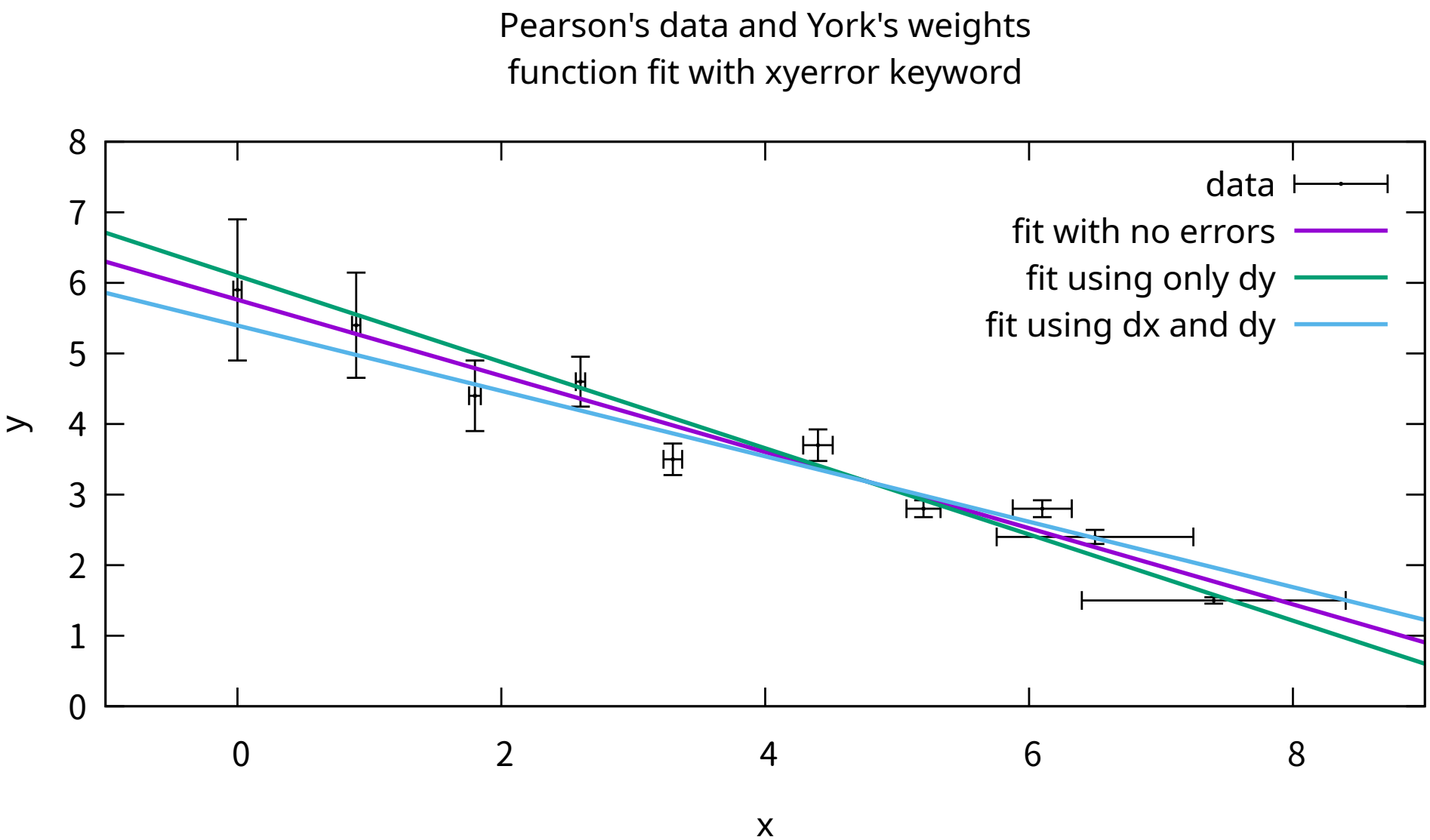

data for all indices t, initial parameters
z
3
2
1
0
-1
-2
-3
0
1
2
3
4
5
x
0
1
2
3
4
5
y
f(x,y,1)
'fit3.dat' u 1:2:4

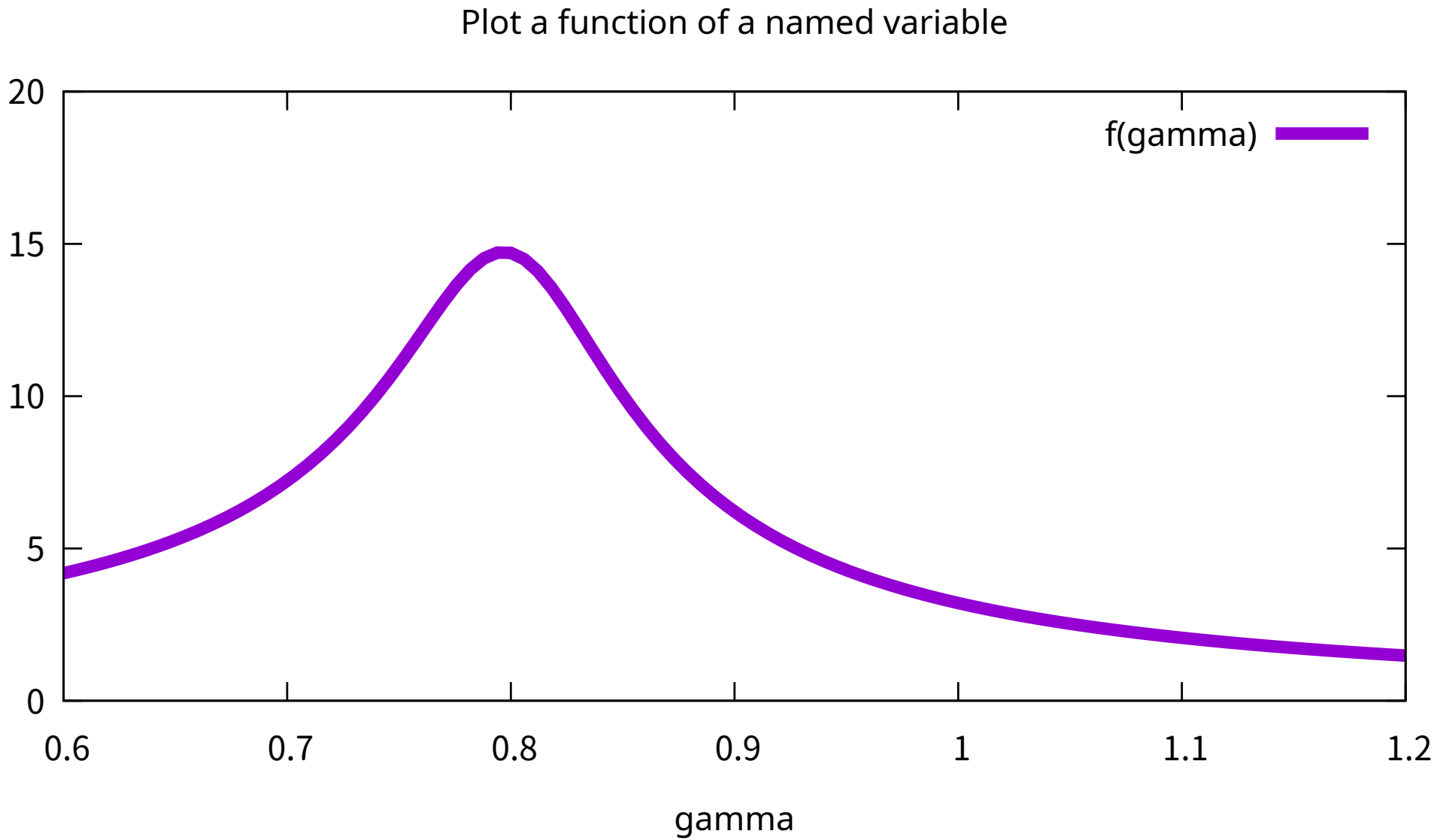

fit to all data
z
3
2
1
0
-1
-2
-3
0
1
2
3
4
5
x
0
1
2
3
4
5
y
f(x,y,3)
'fit3.dat' in 3 u 1:2:4
f(x,y,1)
'fit3.dat' in 2 u 1:2:4
f(x,y,-1)
'fit3.dat' in 1 u 1:2:4
f(x,y,-3)
'fit3.dat' in 0 u 1:2:4

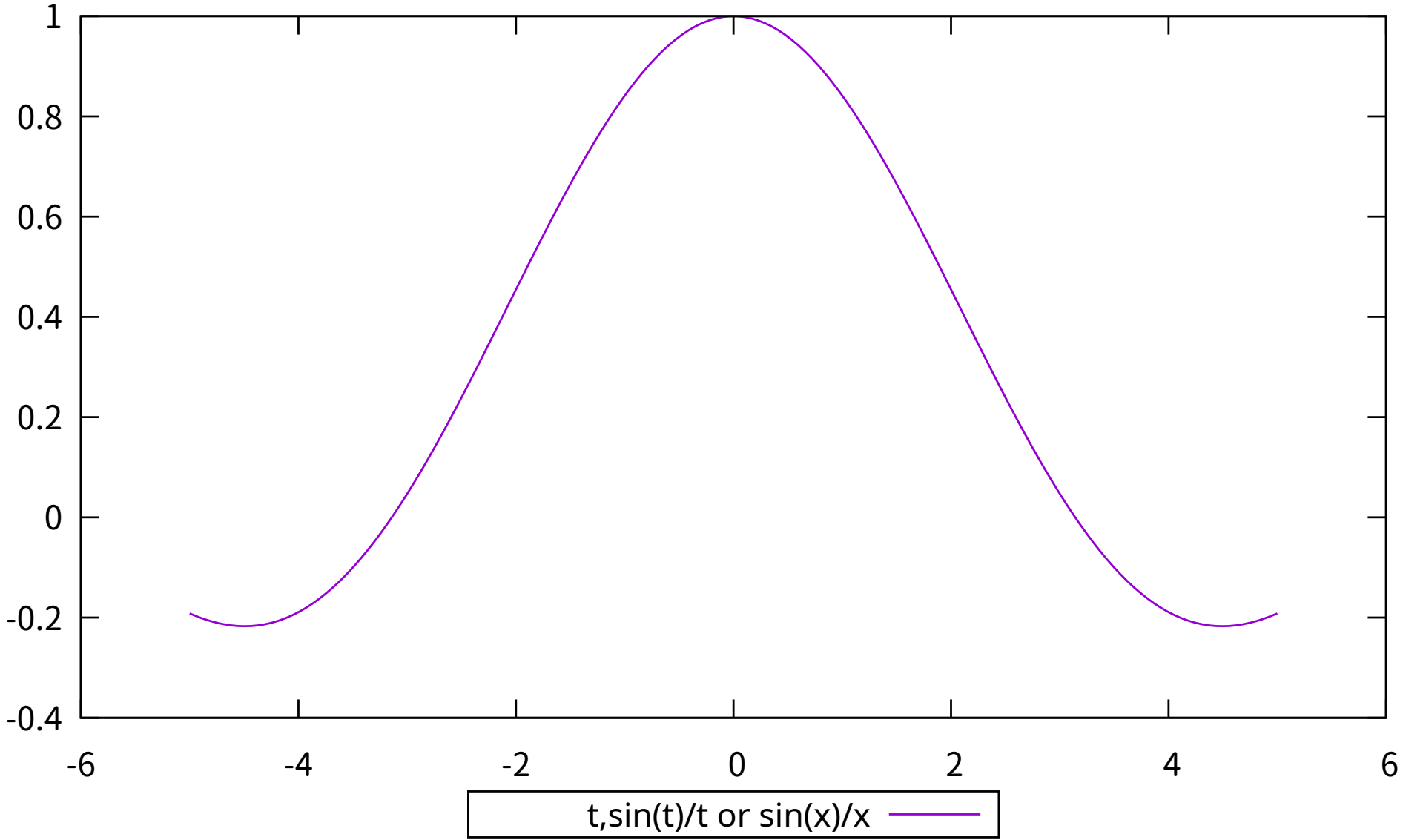

Pearson's data and York's weights
original data and the initial function
data
initial function
y
x
0
1
2
3
4
5
6
7
8
0
2
4
6
8

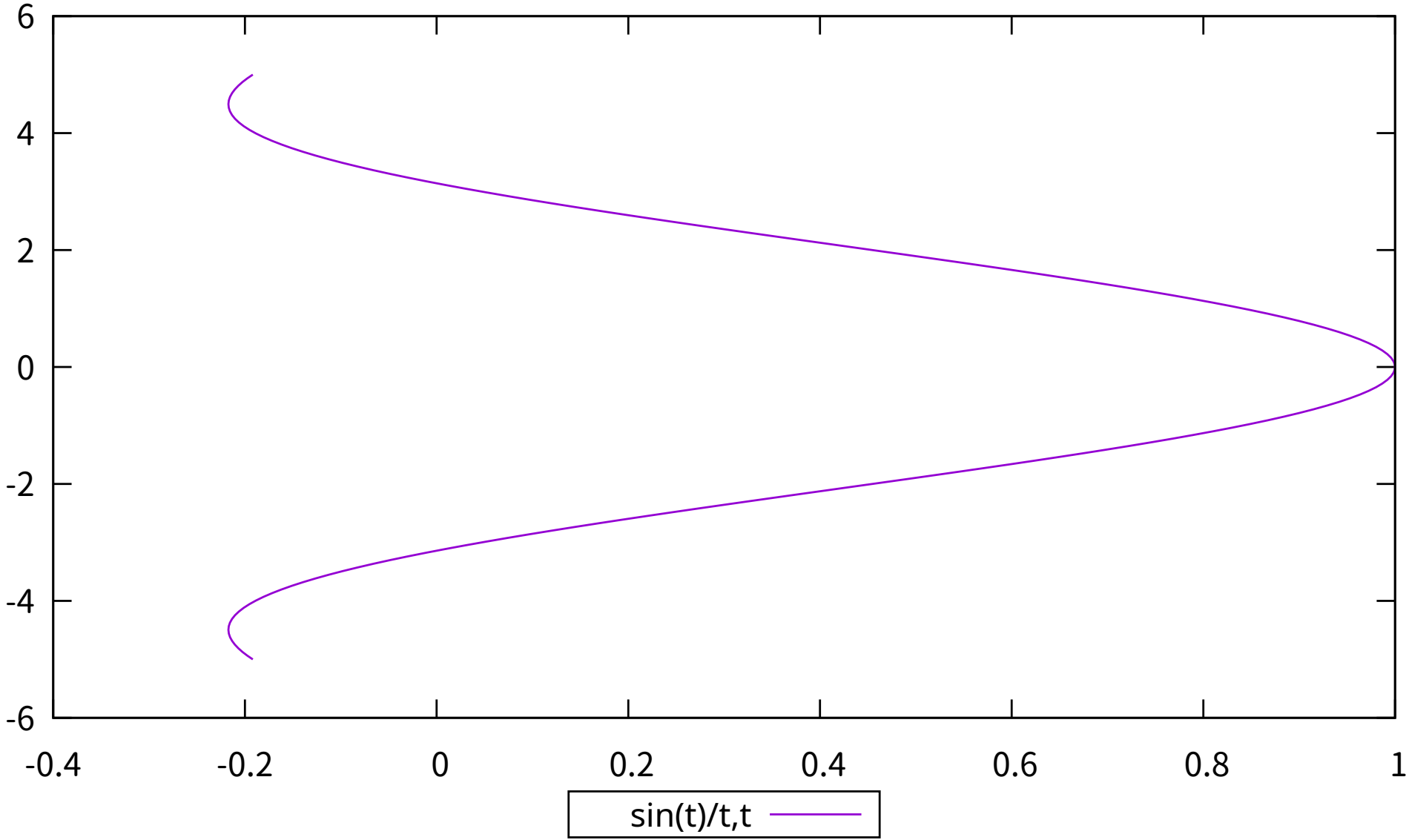

Pearson's data and York's weights
function fit with no error terms
data
fit using no error terms
y
x
0
1
2
3
4
5
6
7
8
0
2
4
6
8

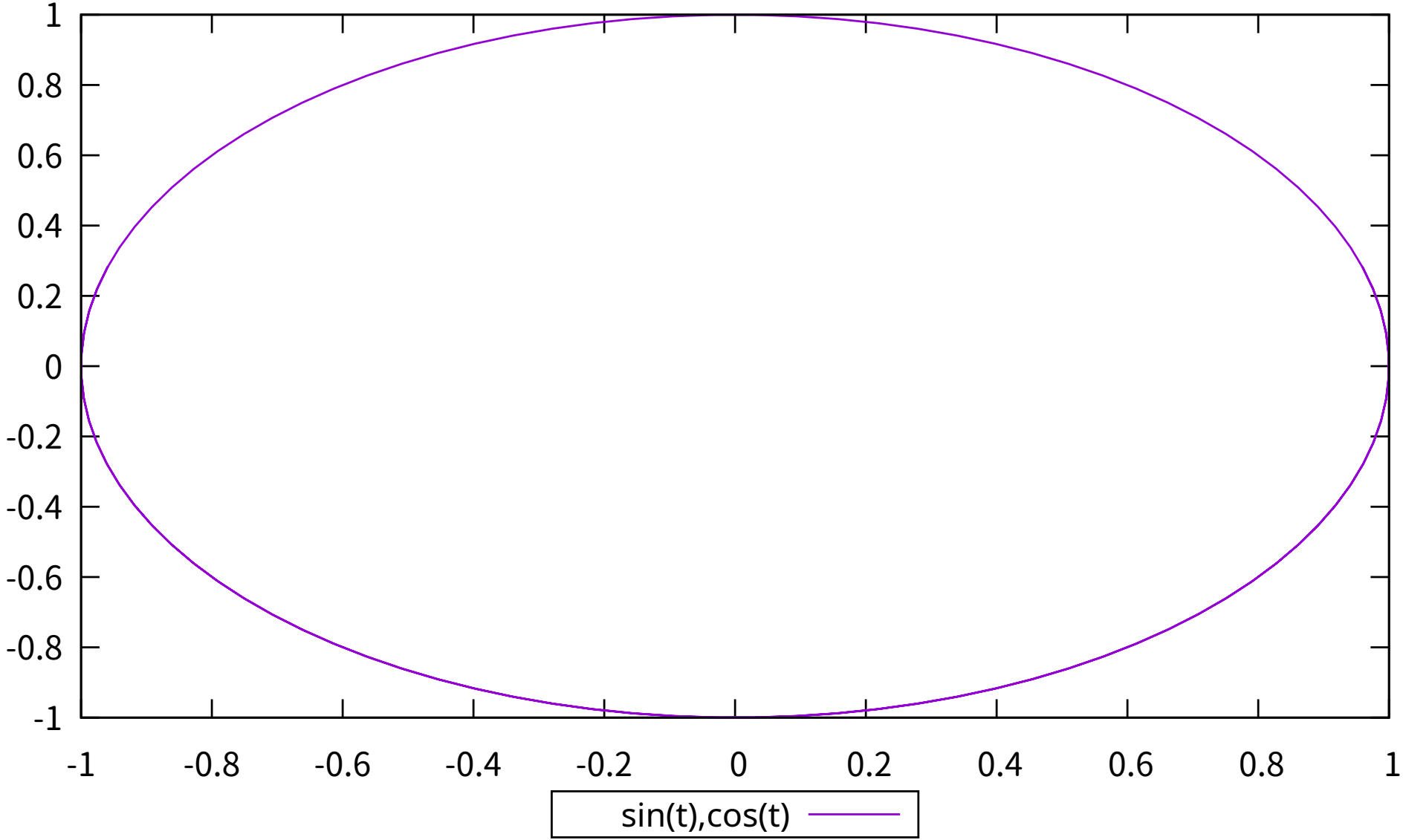

Pearson's data and York's weights
function fit with yerror keyword
data
fit using only dy
y
x
0
1
2
3
4
5
6
7
8
0
2
4
6
8

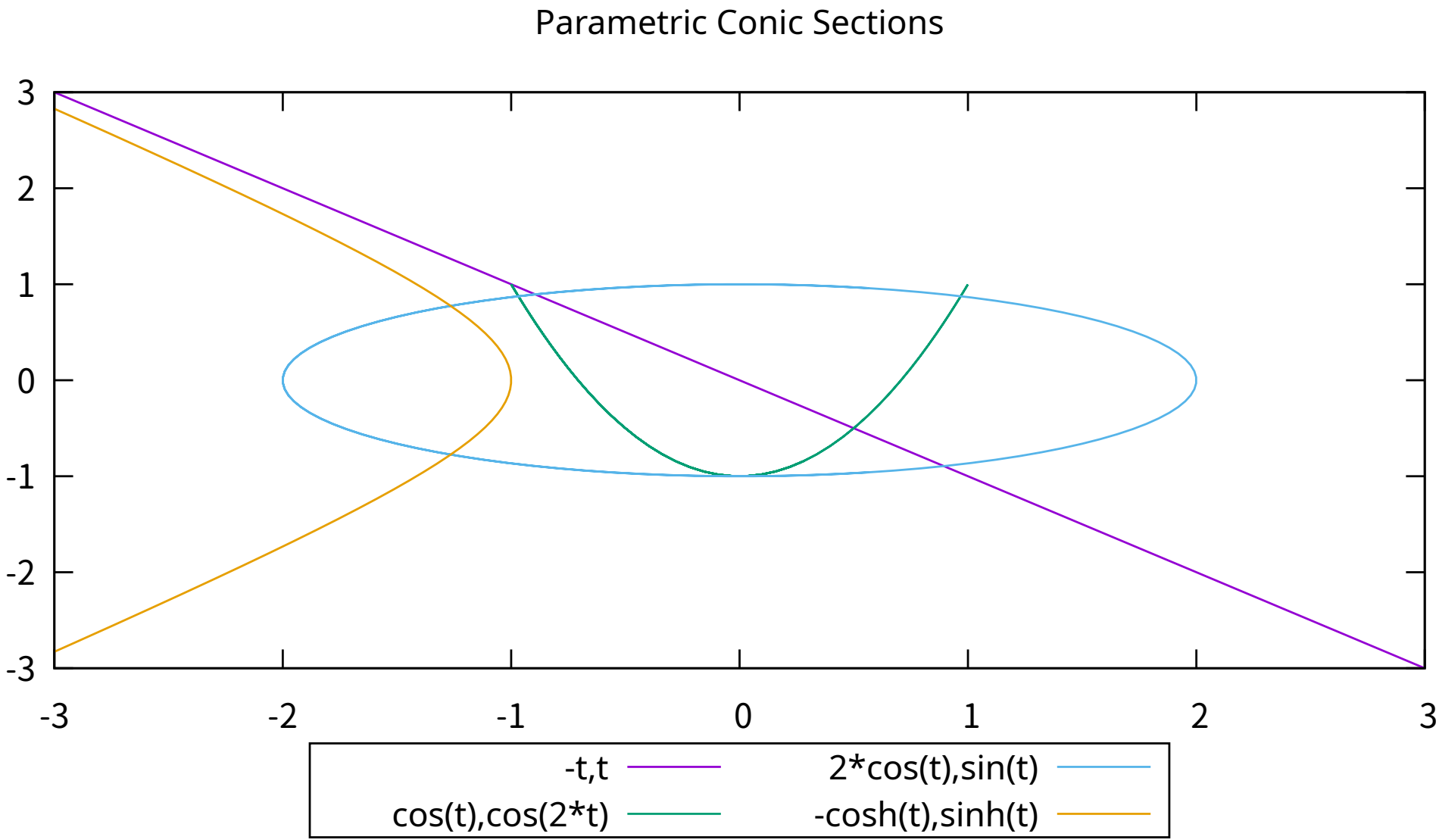

Pearson's data and York's weights
function fit with xyerror keyword
data
fit with no errors
fit using only dy
fit using dx and dy
y
x
0
1
2
3
4
5
6
7
8
0
2
4
6
8

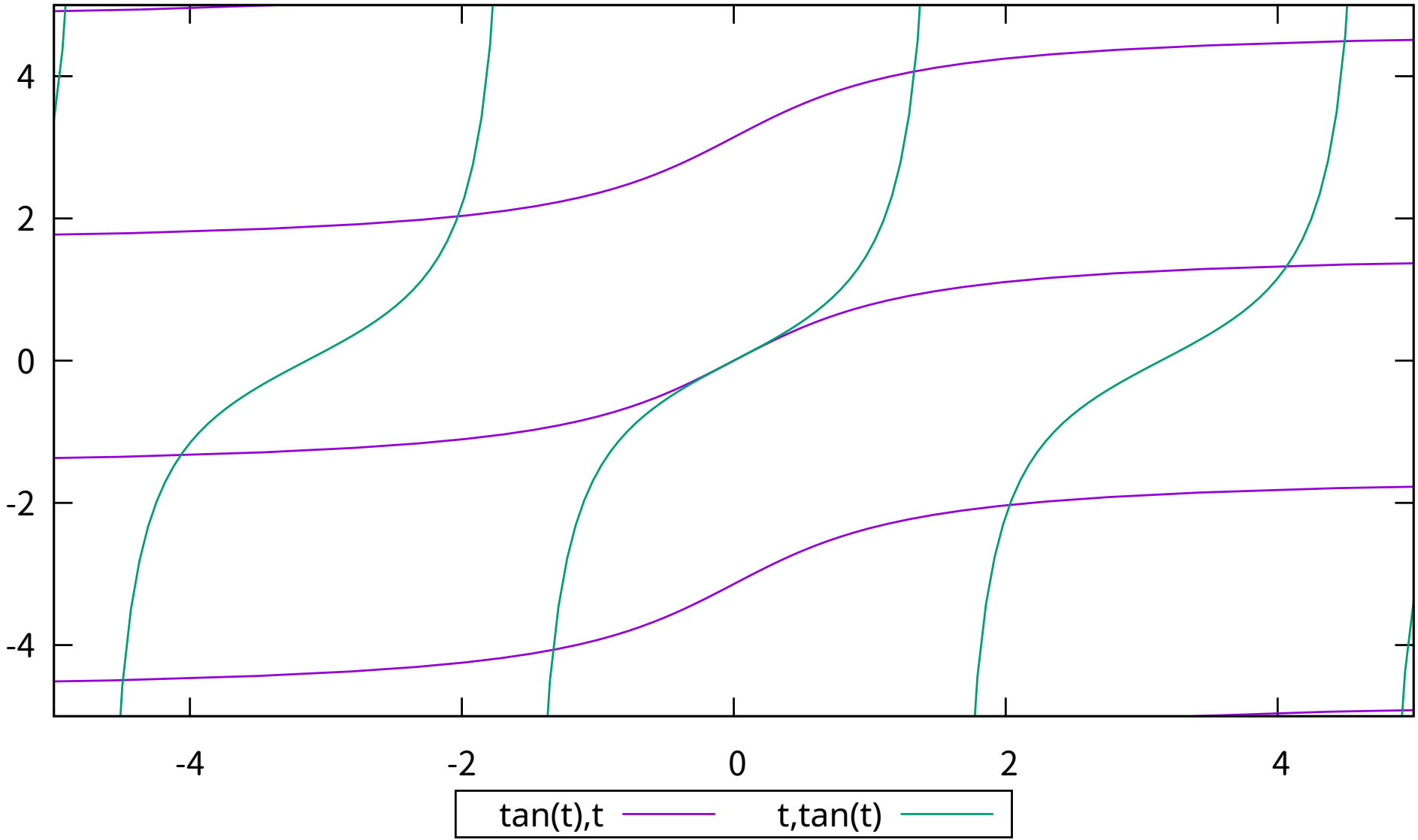

Plot a function of a named variable
f(gamma)
20
15
10
5
0
0.6
0.7
0.8
0.9
1
1.1
1.2
gamma

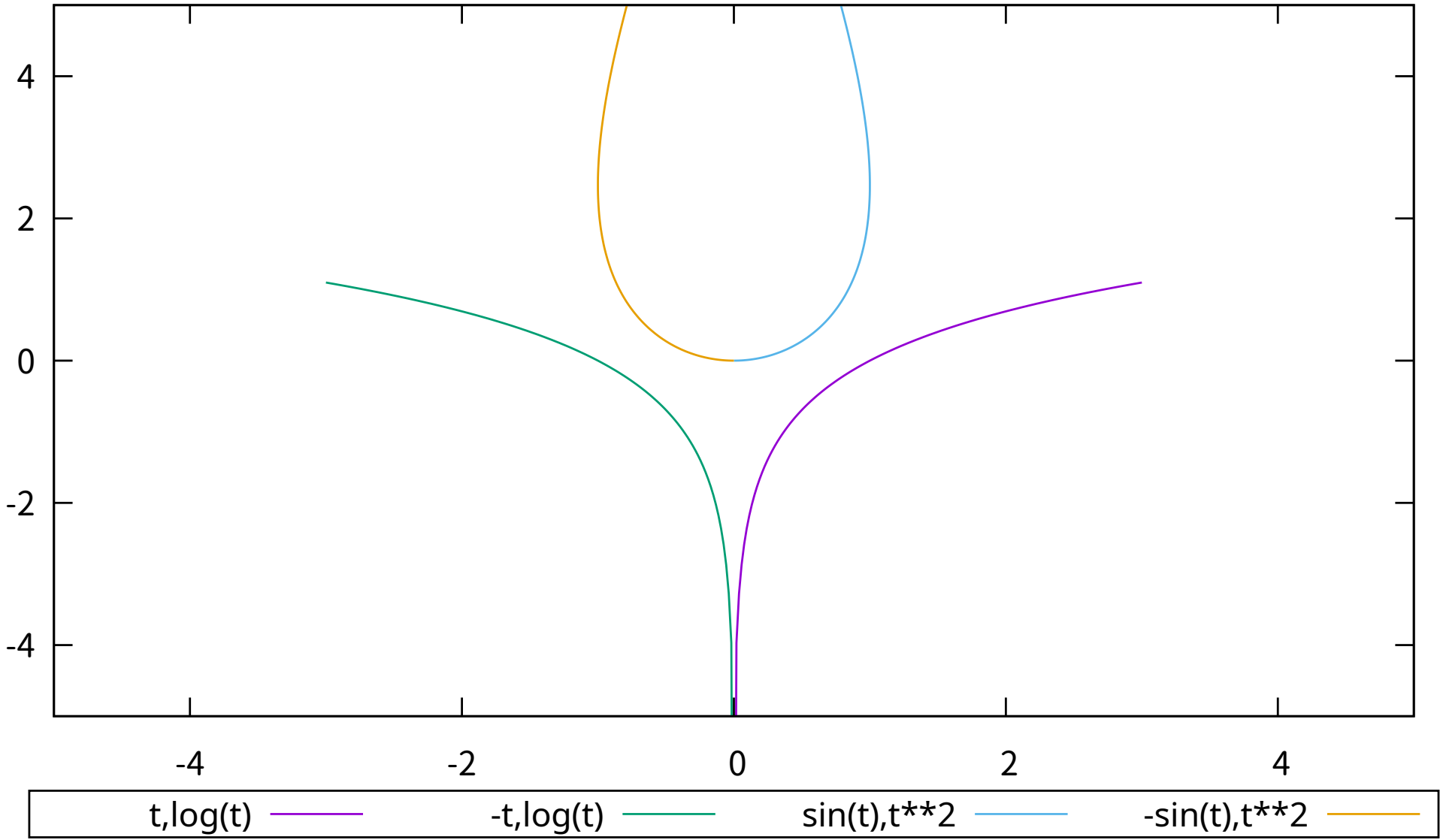

1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-6
-4
-2
0
2
4
6
t,sin(t)/t or sin(x)/x

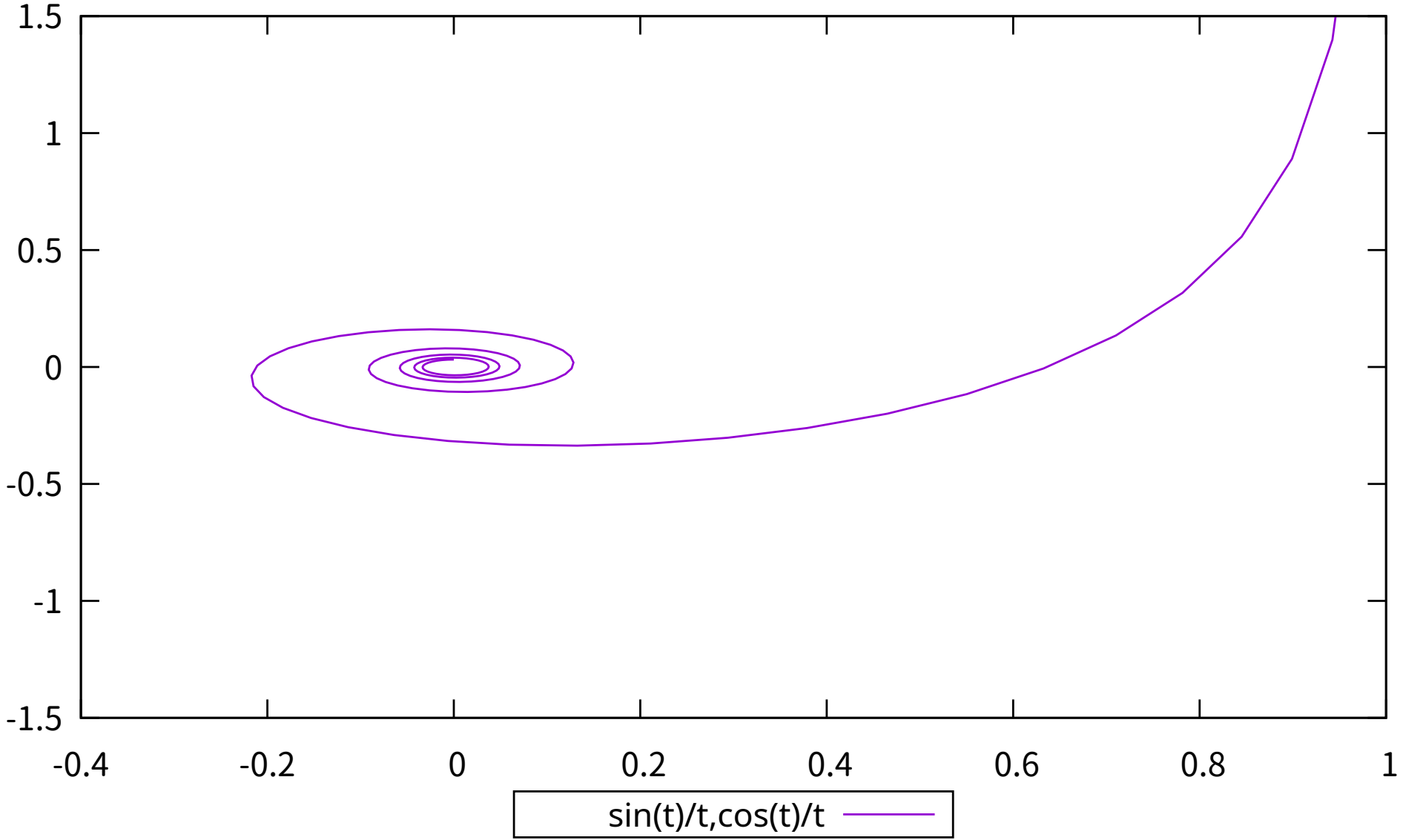

6
4
2
0
-2
-4
-6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
sin(t)/t,t


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)

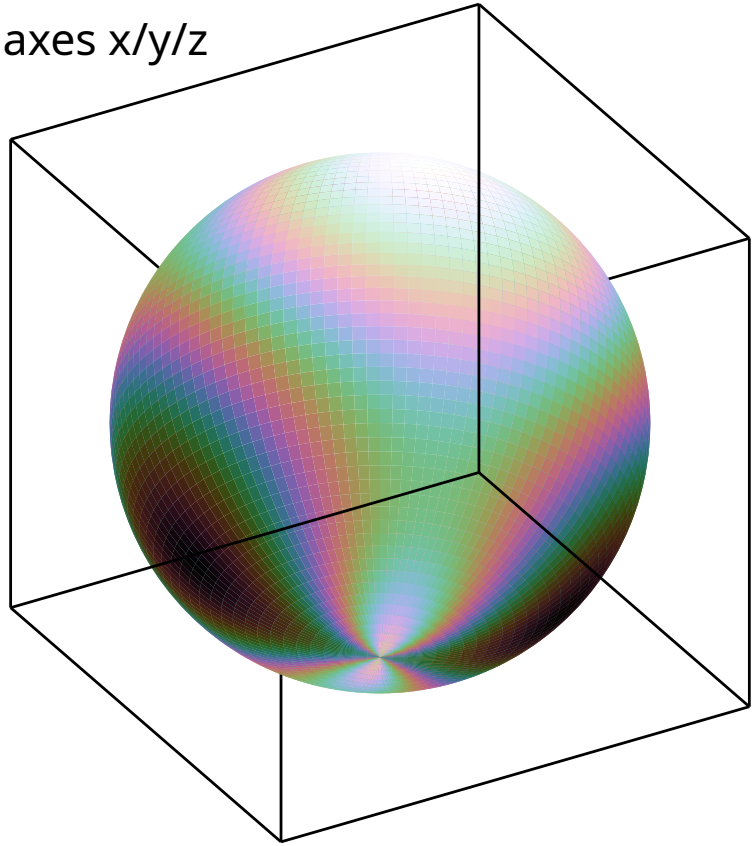

Parametric Conic Sections
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
-t,t
2*cos(t),sin(t)
cos(t),cos(2*t)
-cosh(t),sinh(t)

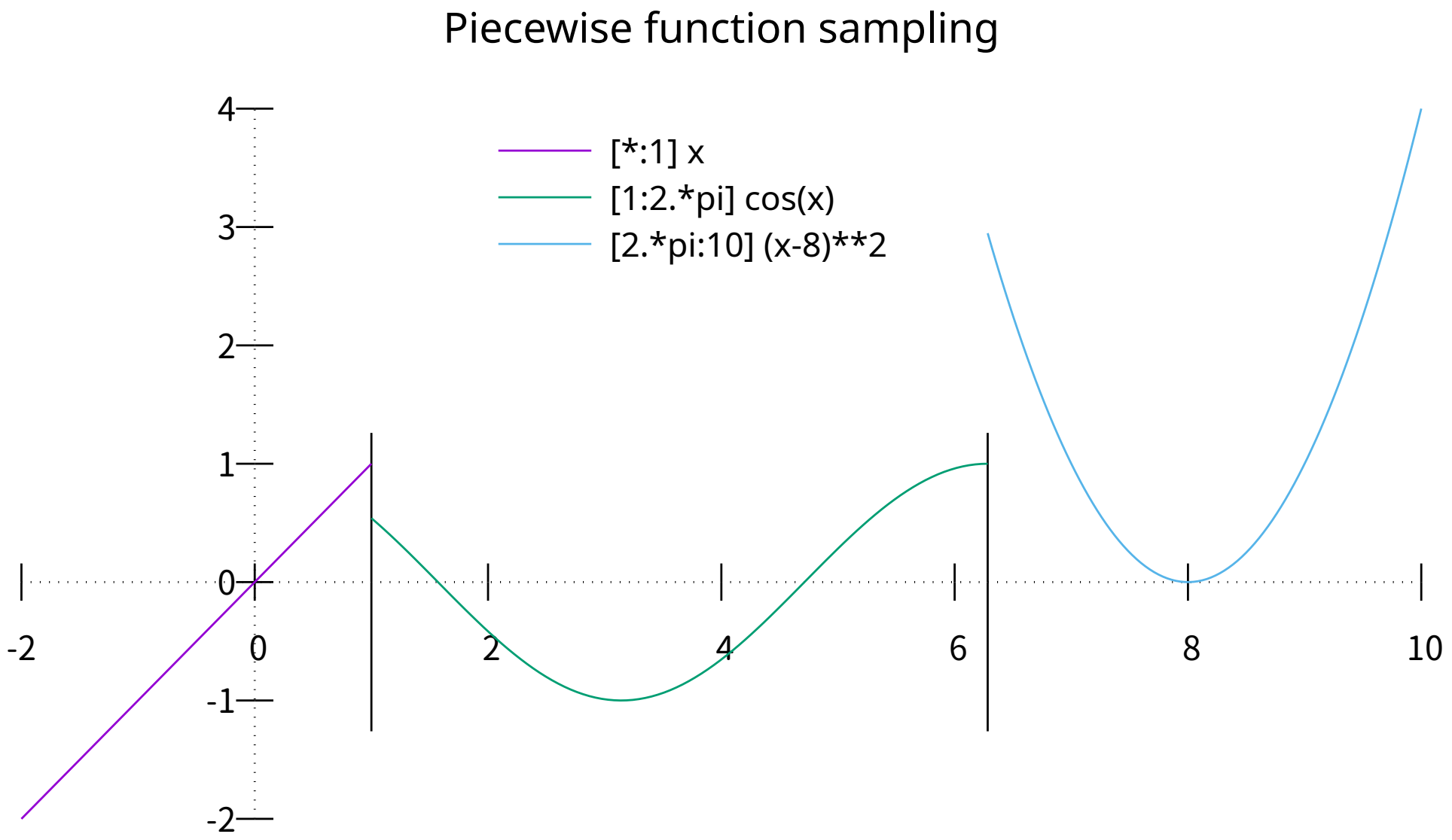

4
2
0
-2
-4
-4
-2
0
2
4
tan(t),t
t,tan(t)

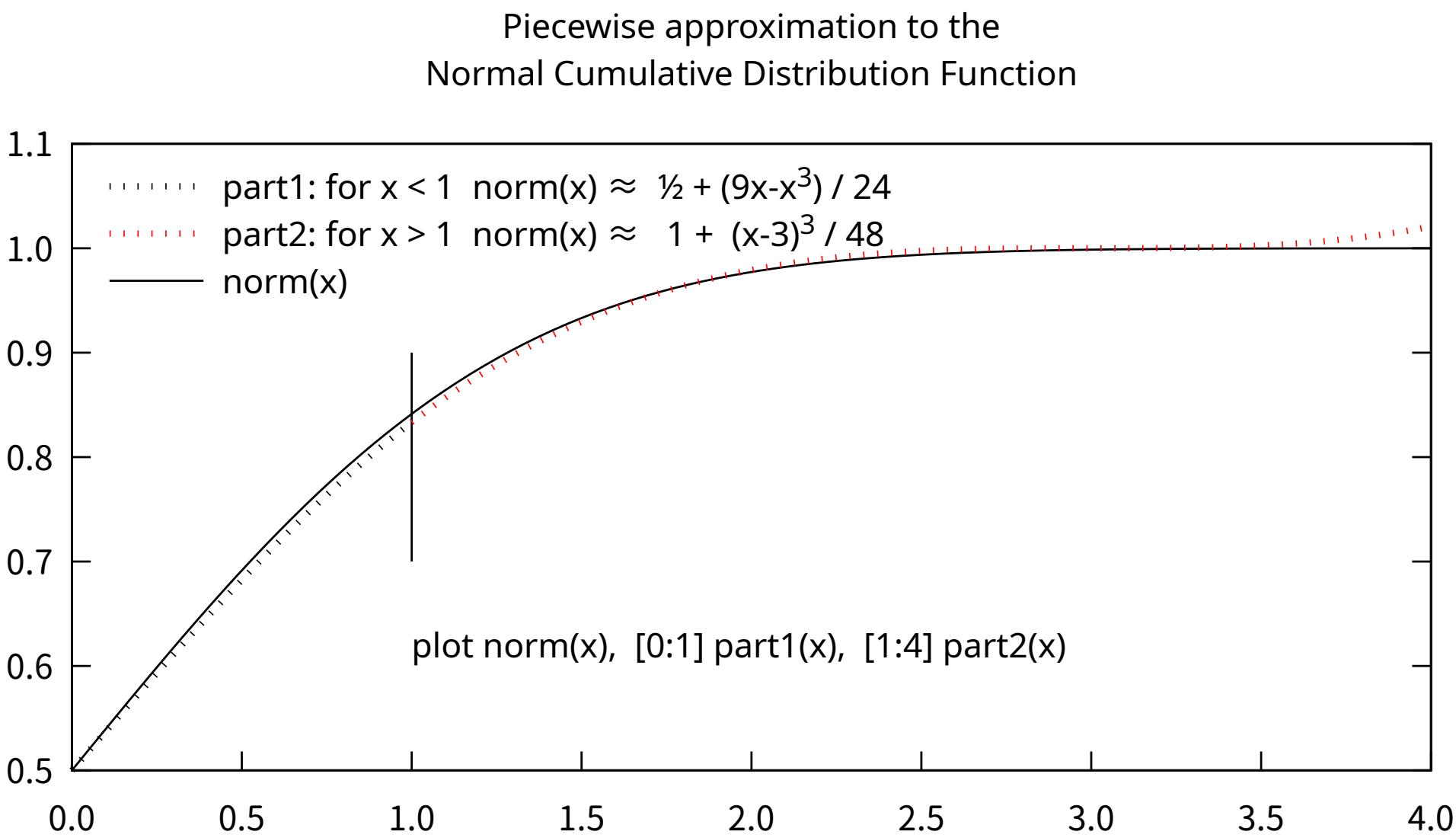

4
2
0
-2
-4
-4
-2
0
2
4
t,log(t)
-t,log(t)
sin(t),t**2
-sin(t),t**2


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

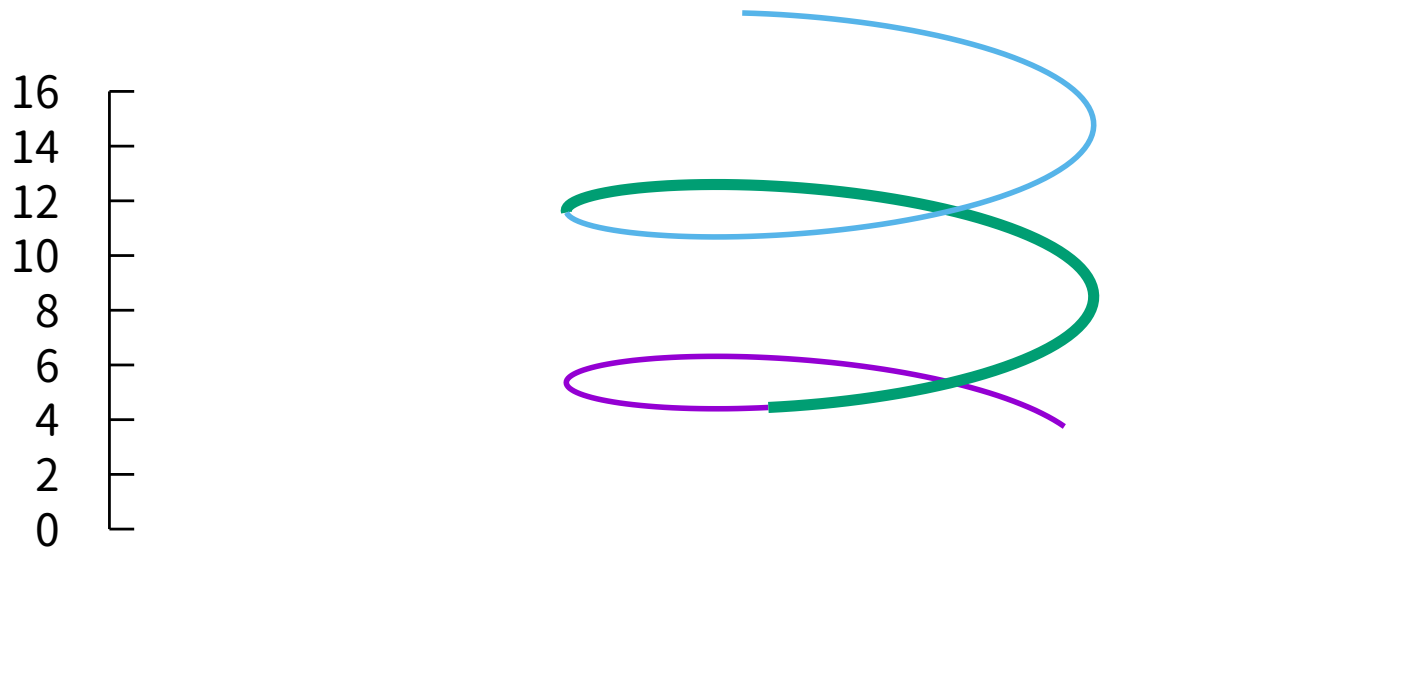
Decouple range of parametric axes u/v
from that of display axes x/y/z

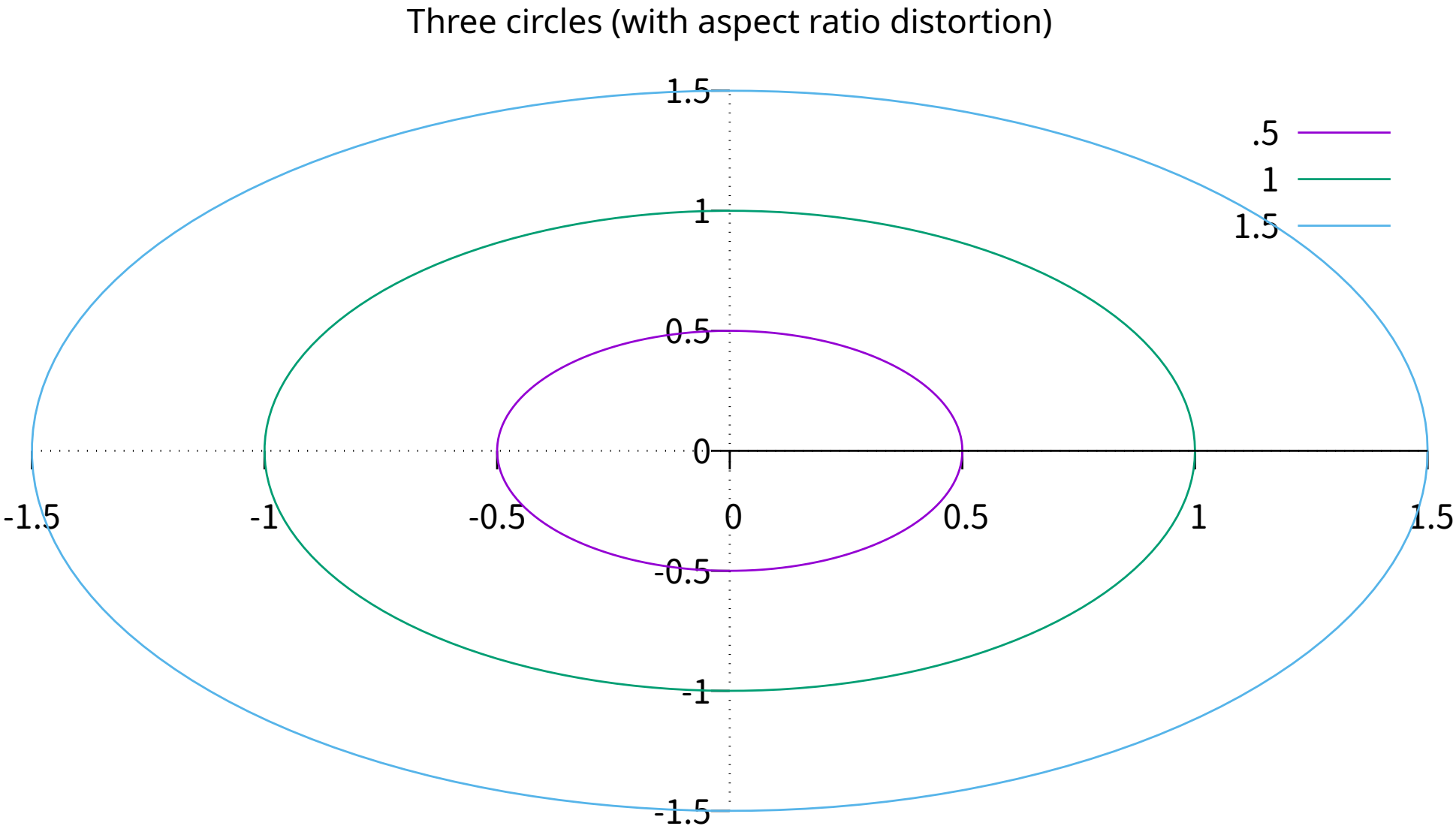


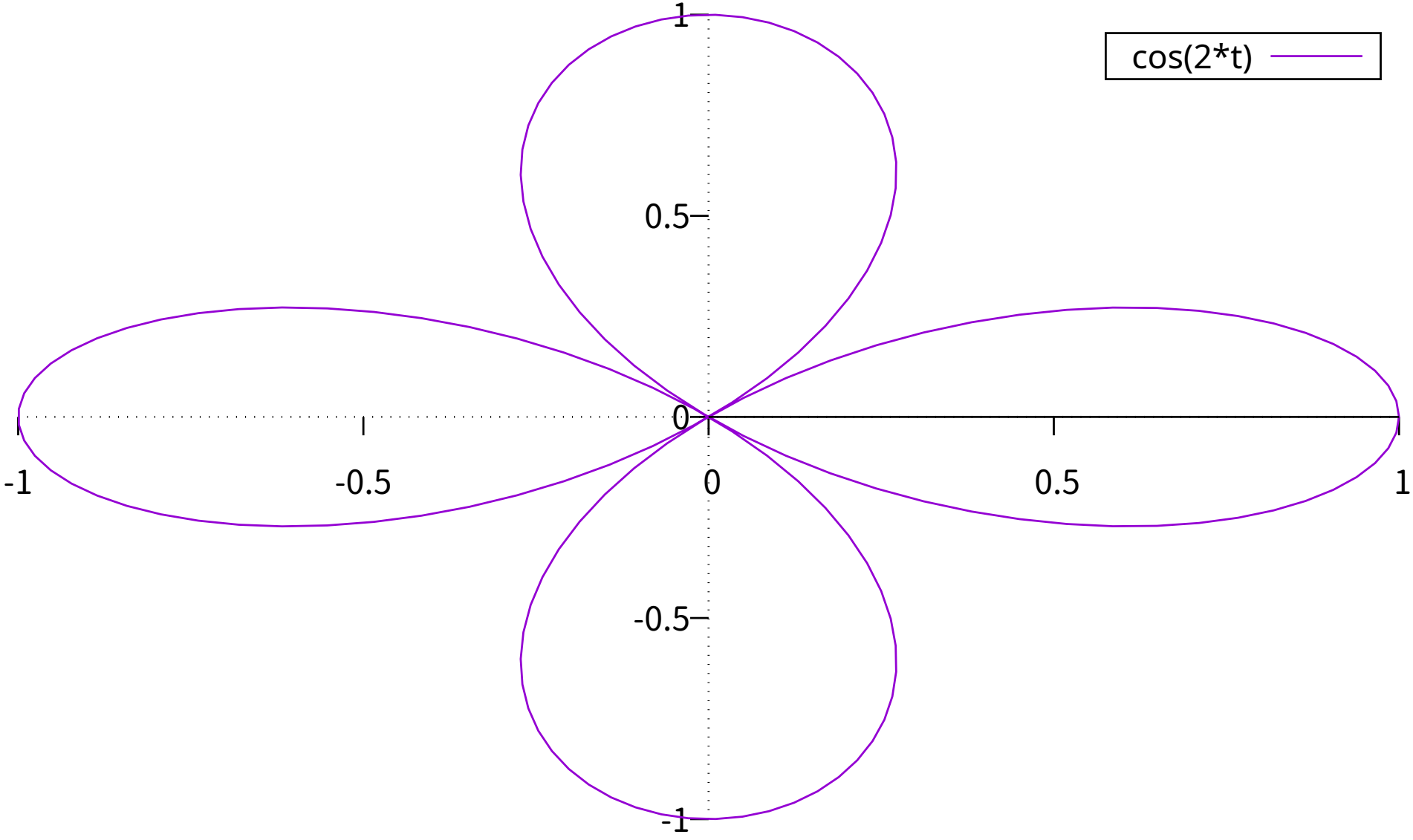

Piecewise function sampling
[*:1] x
[1:2.*pi] cos(x)
[2.*pi:10] (x-8)**2
4
3
2
1
0
-1
-2
-2
0
2
4
6
8
10

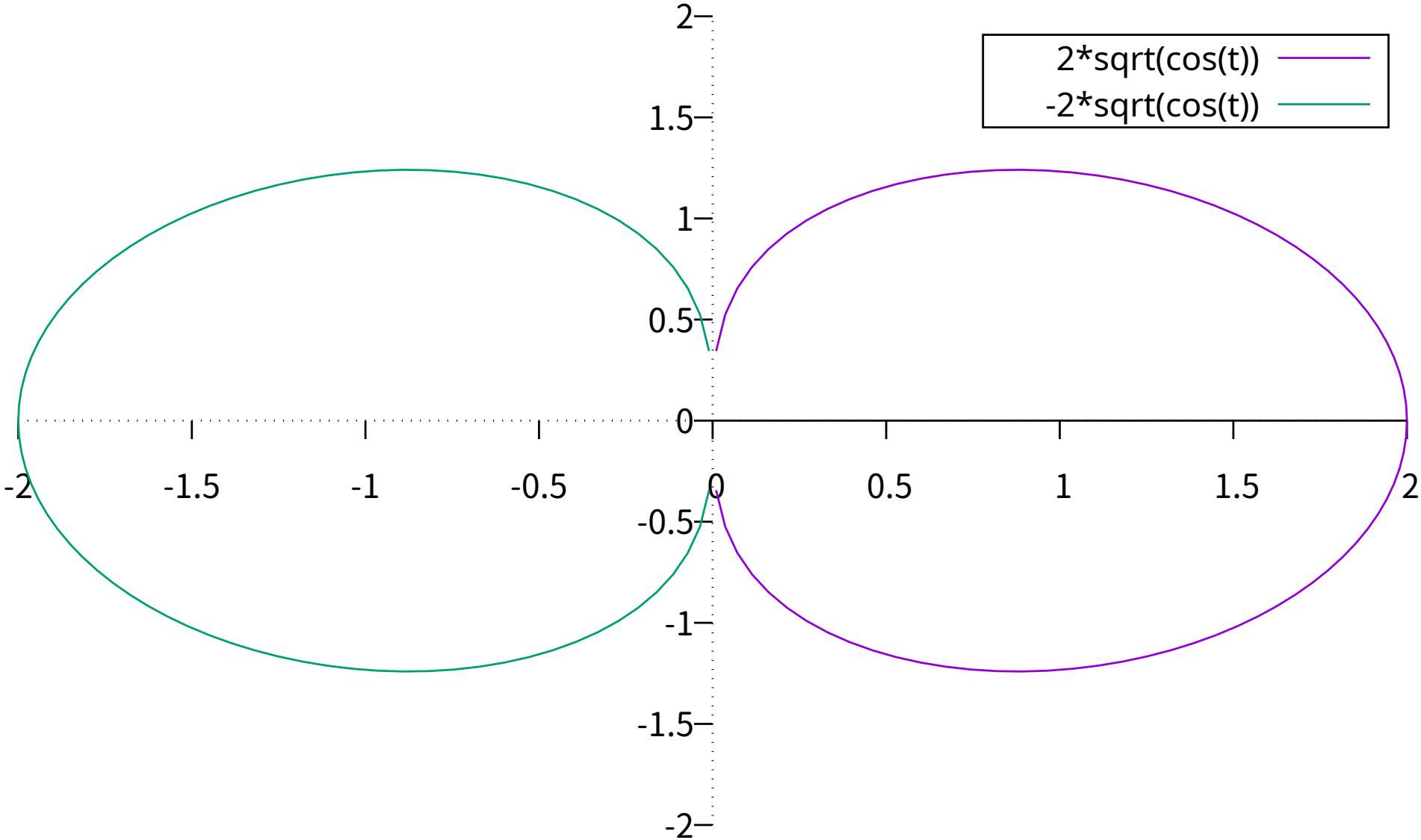

Piecewise approximation to the
Normal Cumulative Distribution Function
part1: for x < 1 norm(x) ≈ ½ + (9x-x^3) / 24
part2: for x > 1 norm(x) ≈ 1 + (x-3)^3 / 48
norm(x)
plot norm(x), [0:1] part1(x), [1:4] part2(x)
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

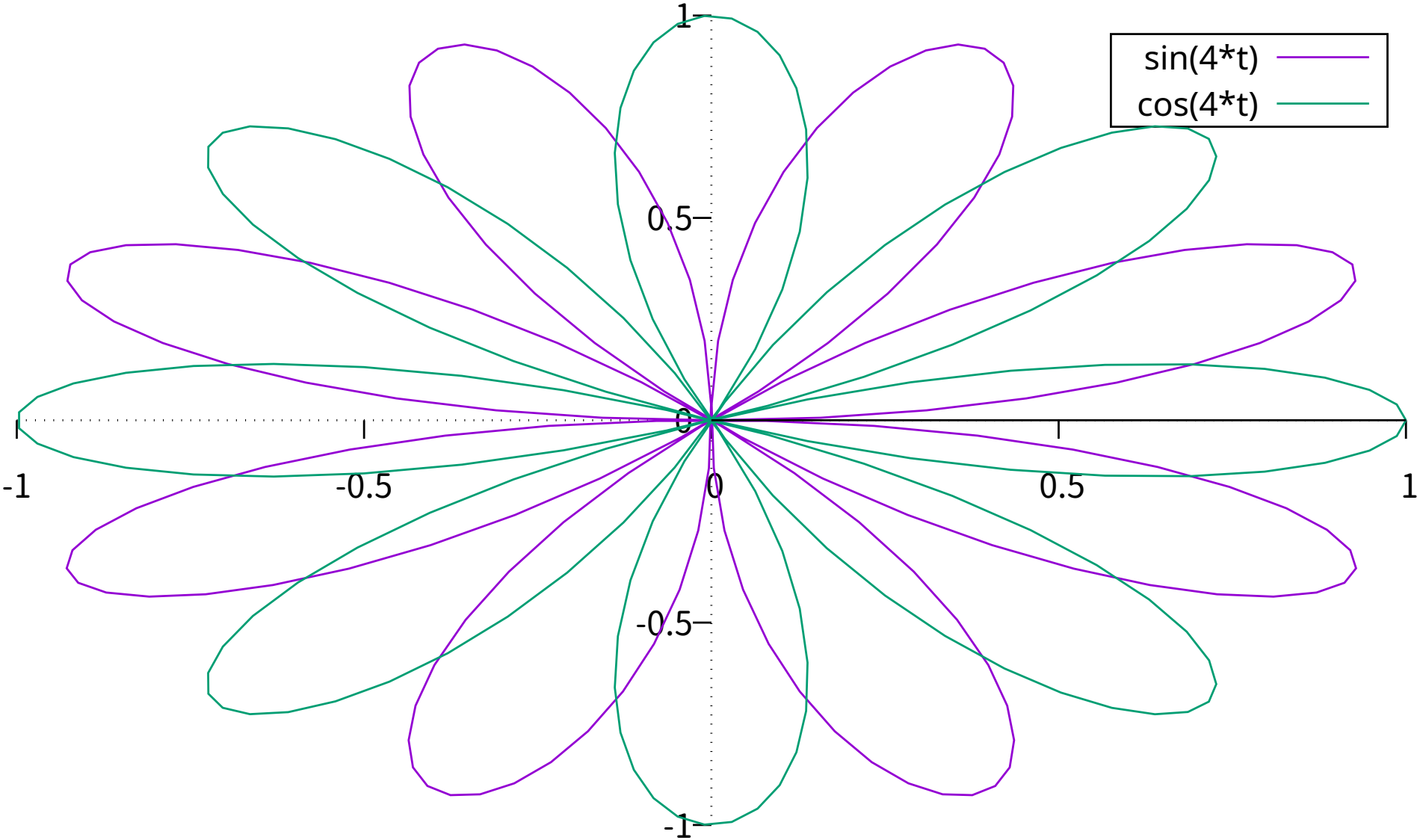
Piecewise function of one parameter in 3D

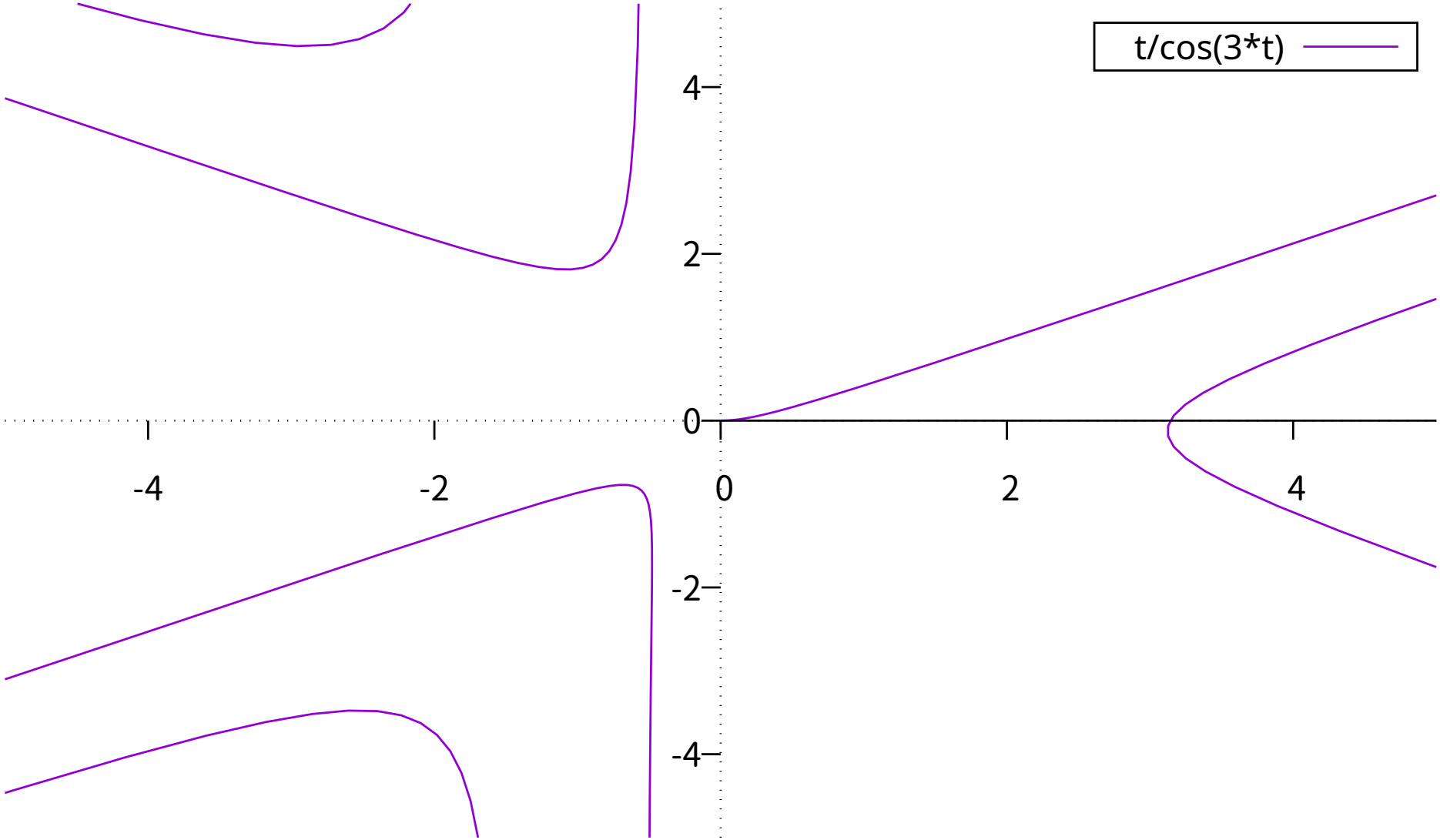


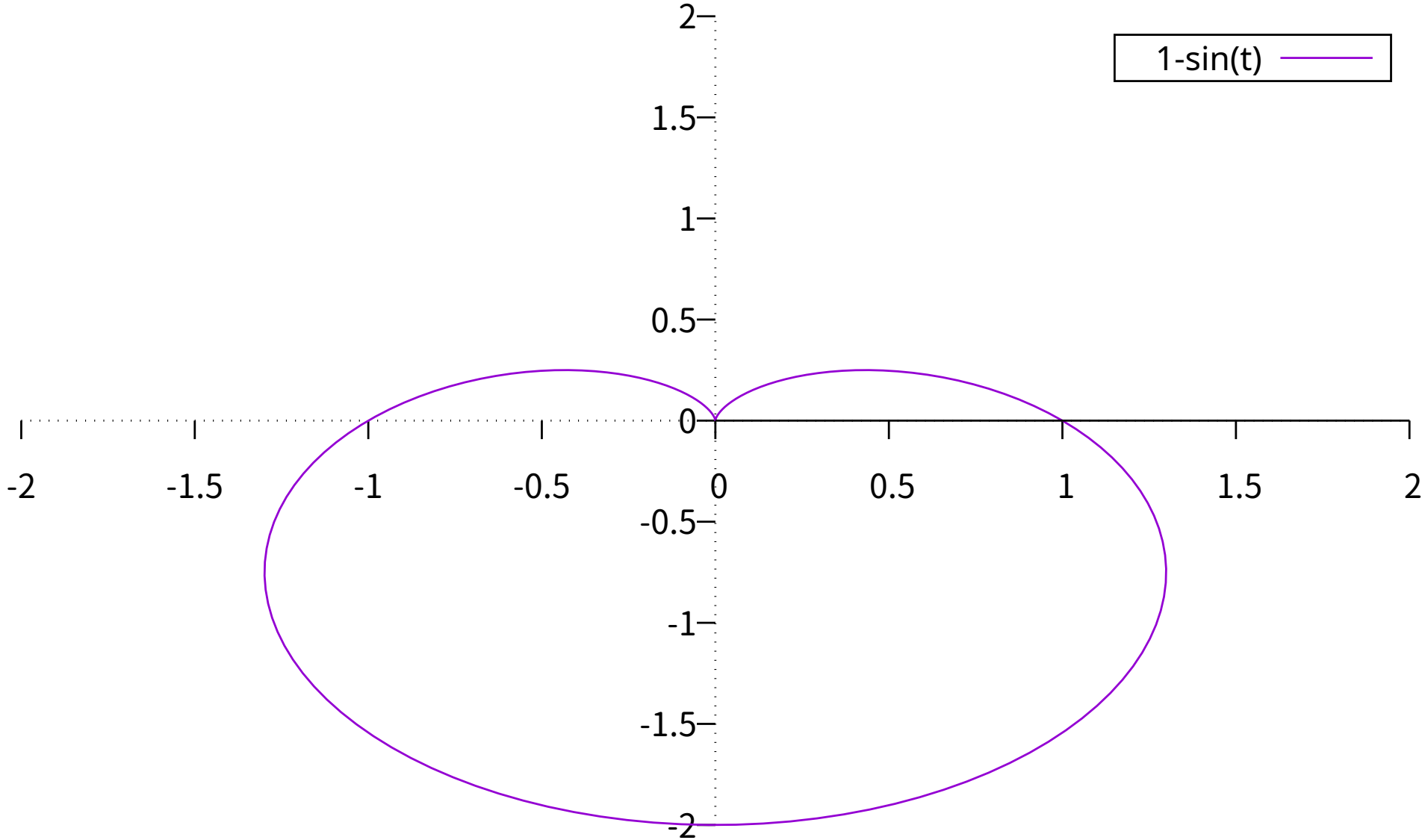

Three circles (with aspect ratio distortion)
.5
1
1.5
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5

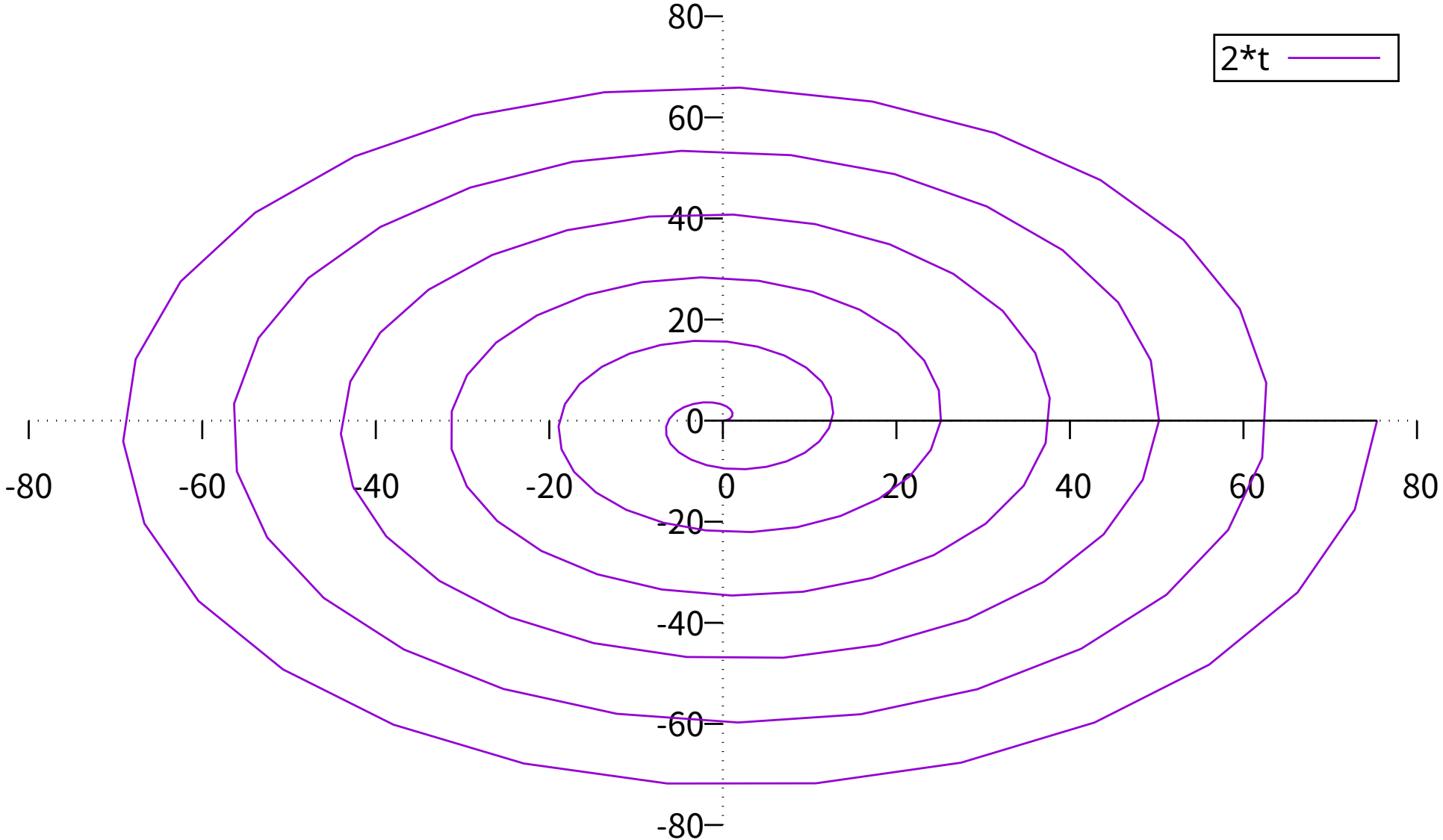

cos(2*t)
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1


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

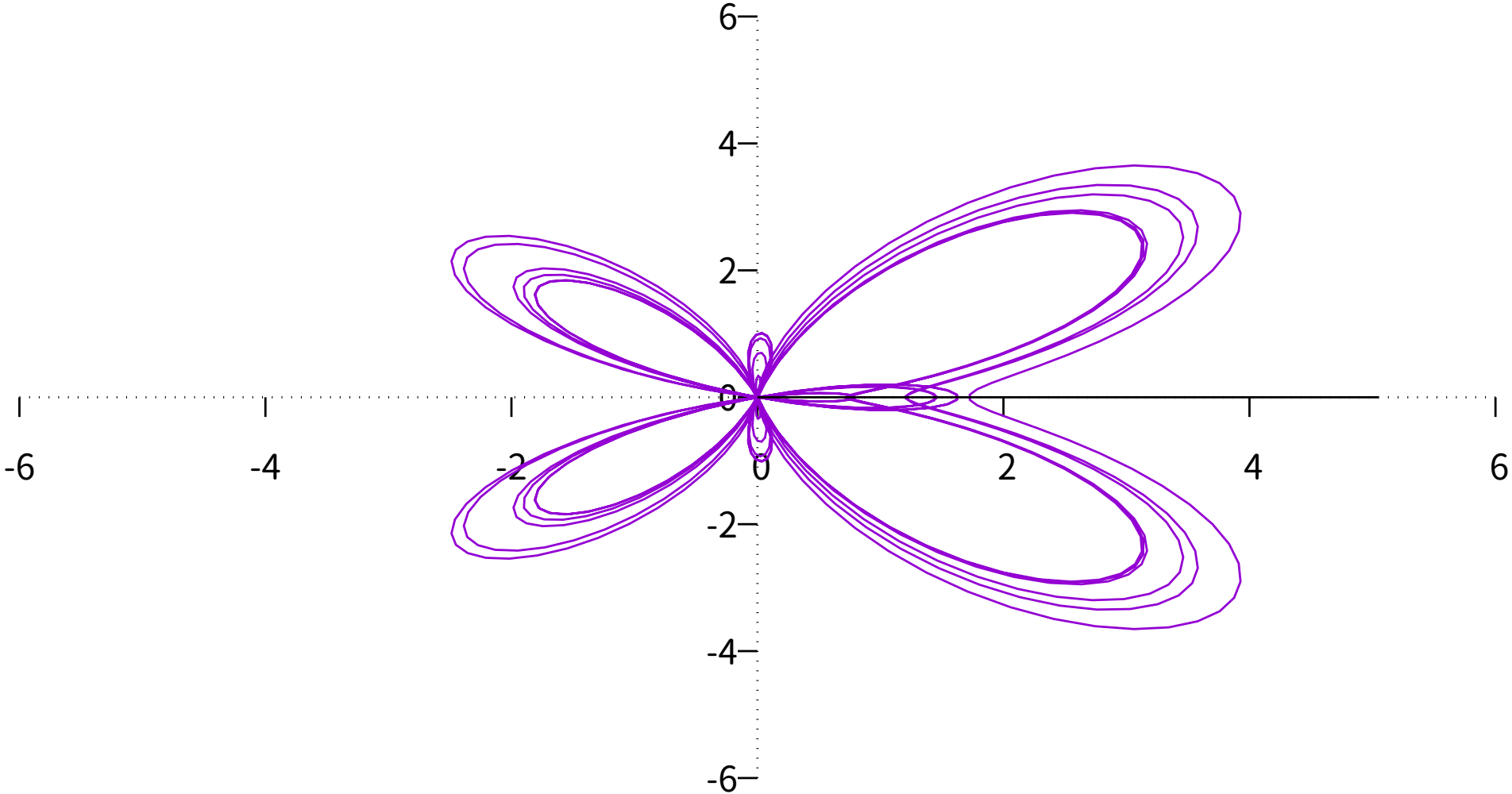

sin(4*t)
cos(4*t)
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

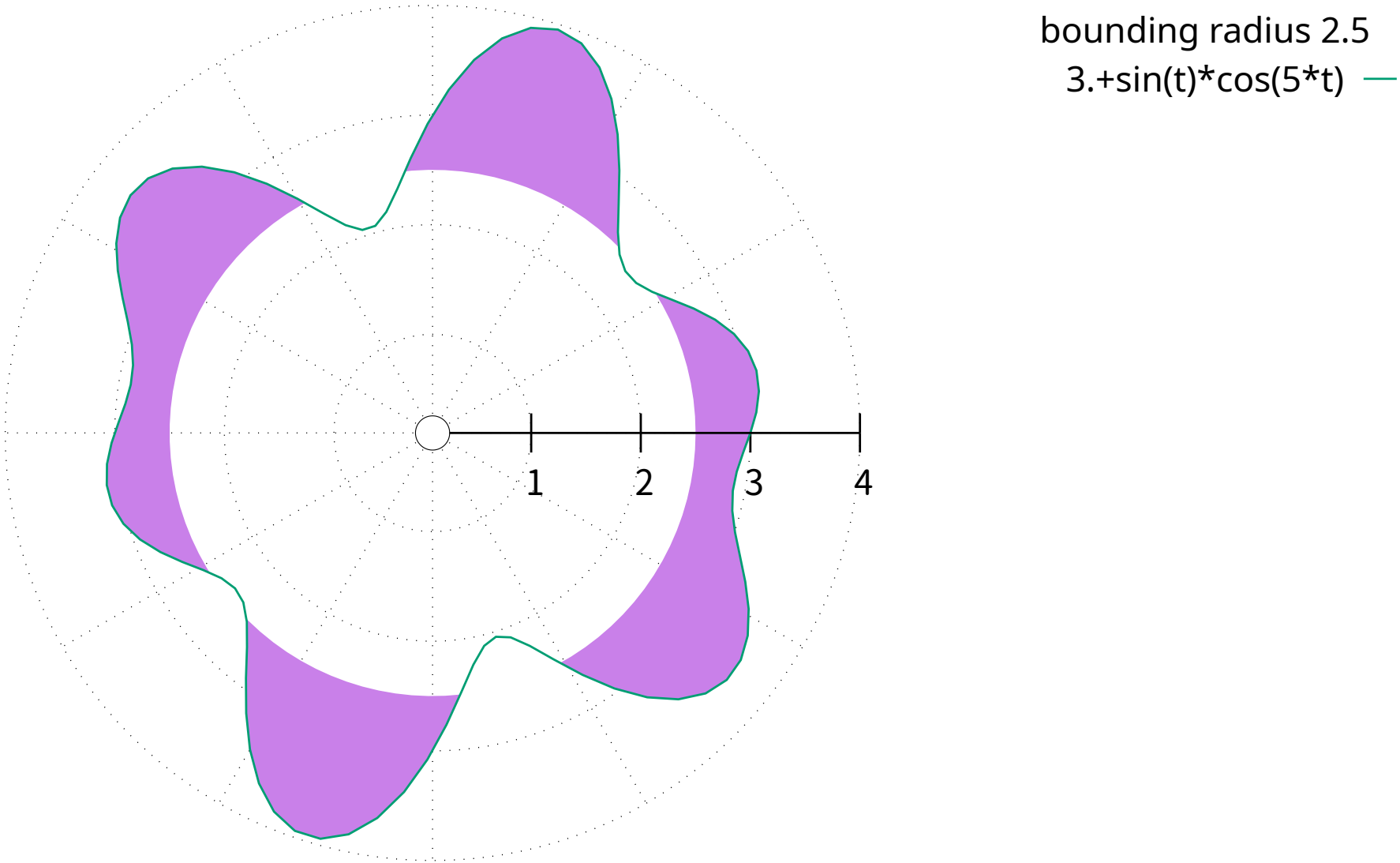

t/cos(3*t)
4
2
0
-2
-4
-4
-2
0
2
4


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

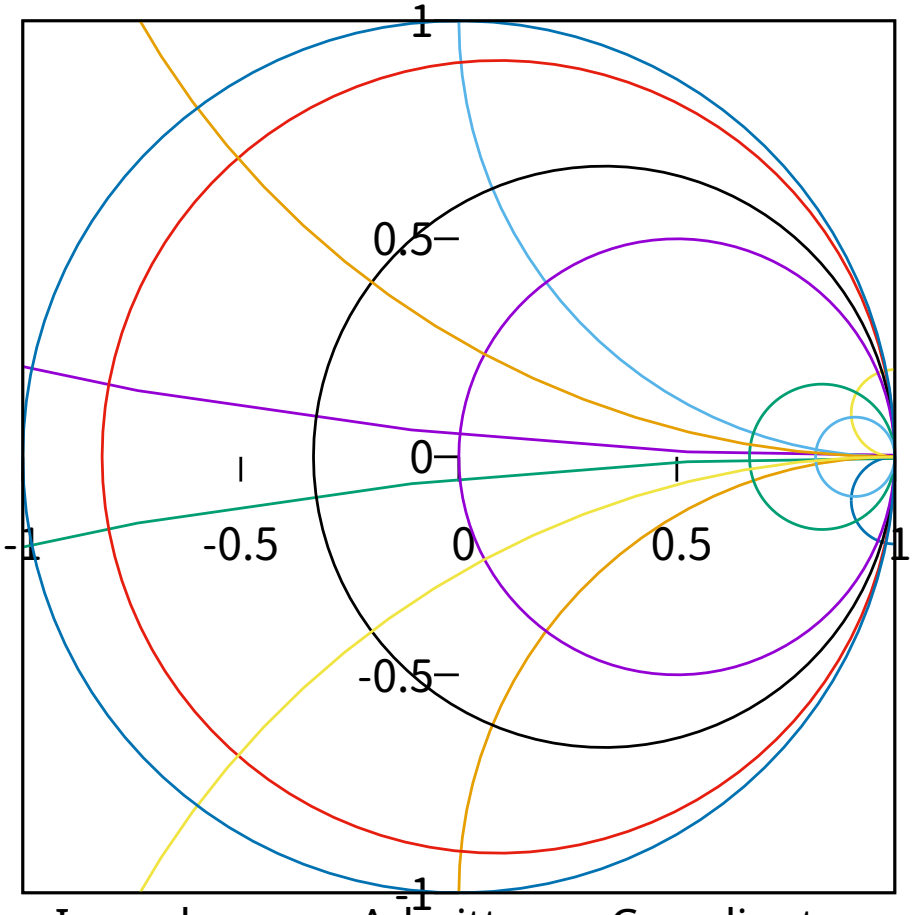

2*t
80
60
40
20
0
-20
-40
-60
-80
-80
-60
-40
-20
0
20
40
60
80

Butterfly



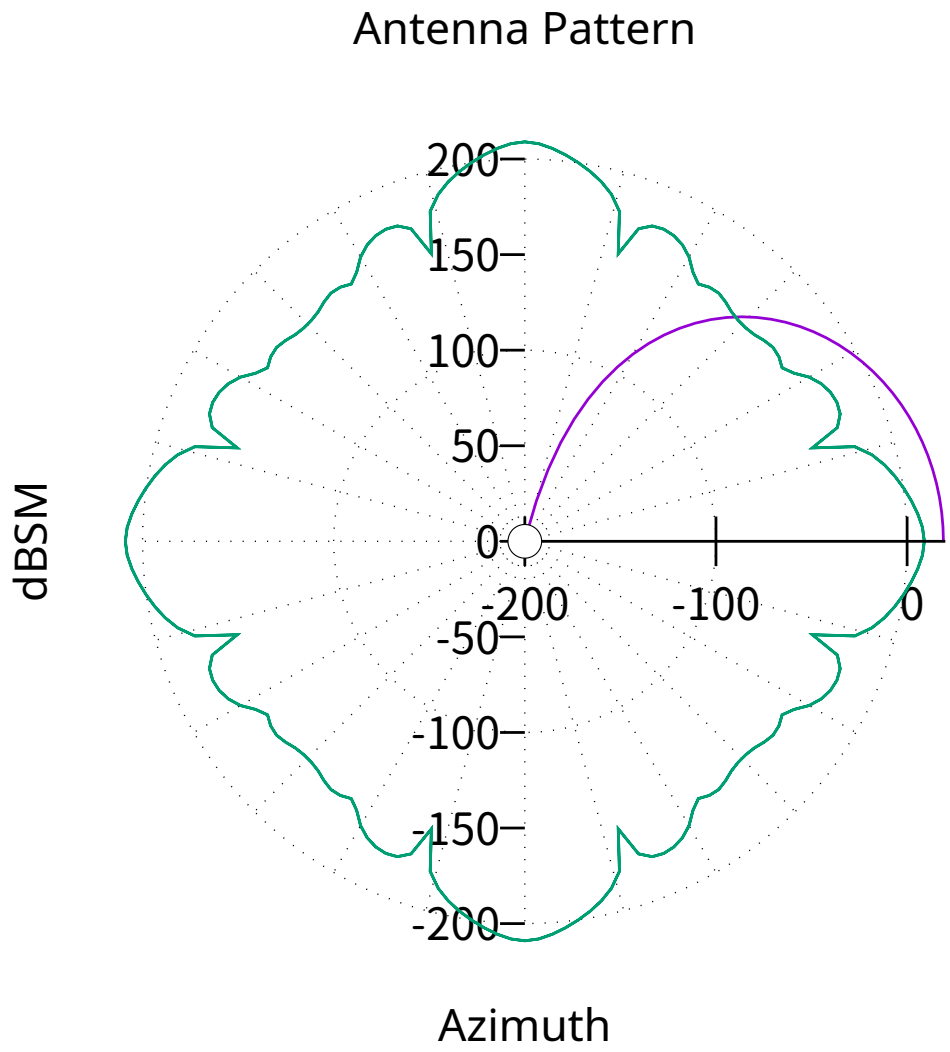

bounding radius 2.5
3.+sin(t)*cos(5*t)
1
2
3
4

Primitive Smith Chart



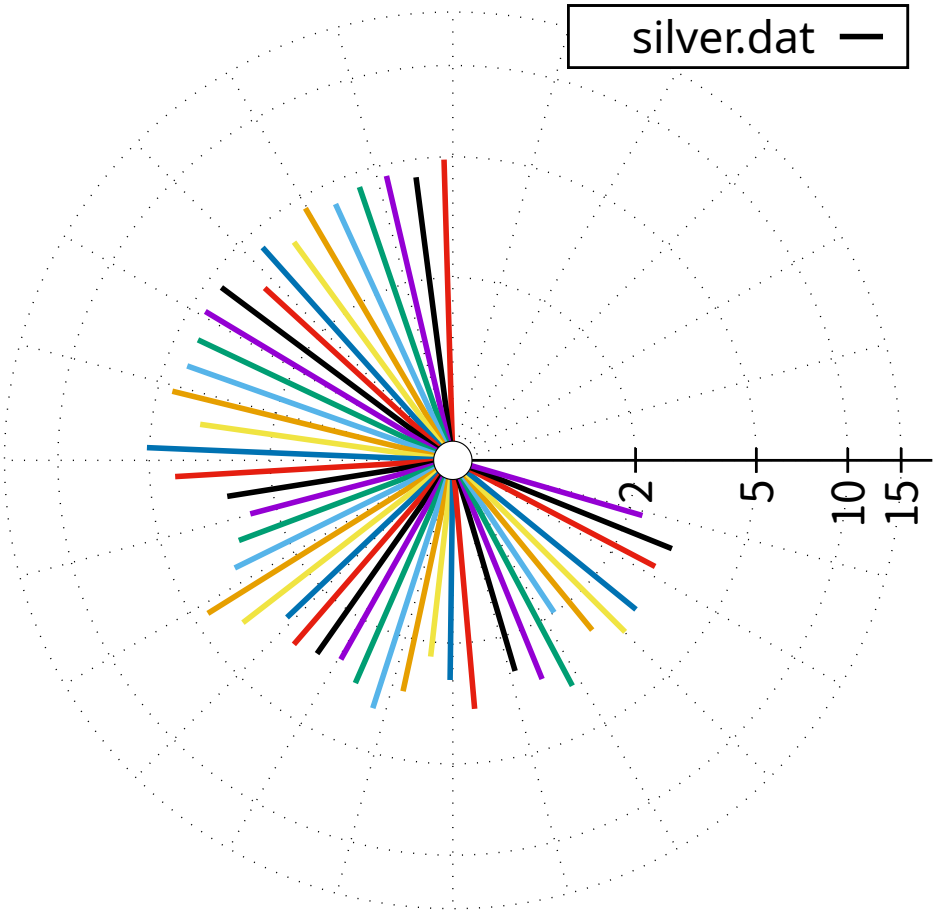
Impedance or Admittance Coordinates

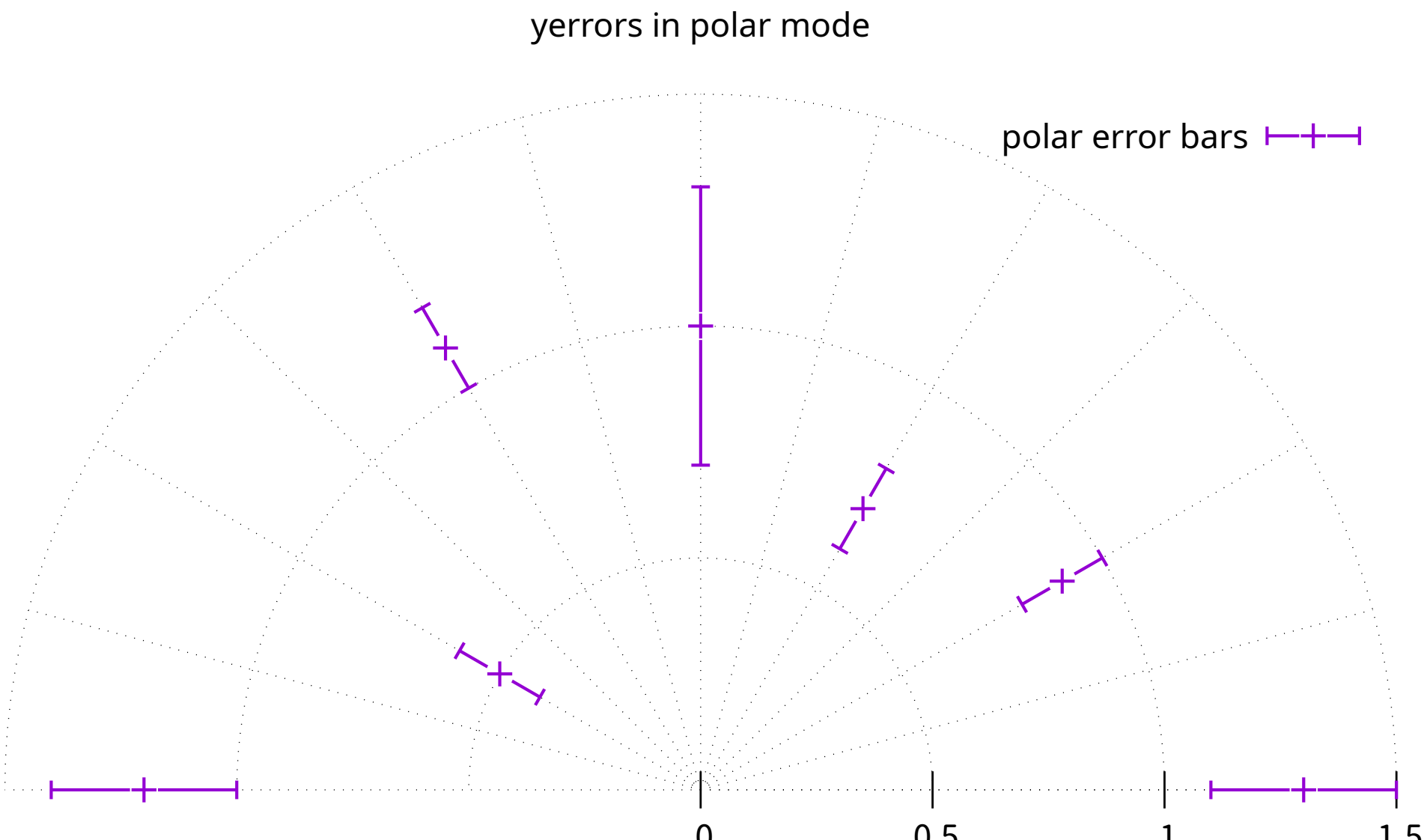
火 25 2月 2025

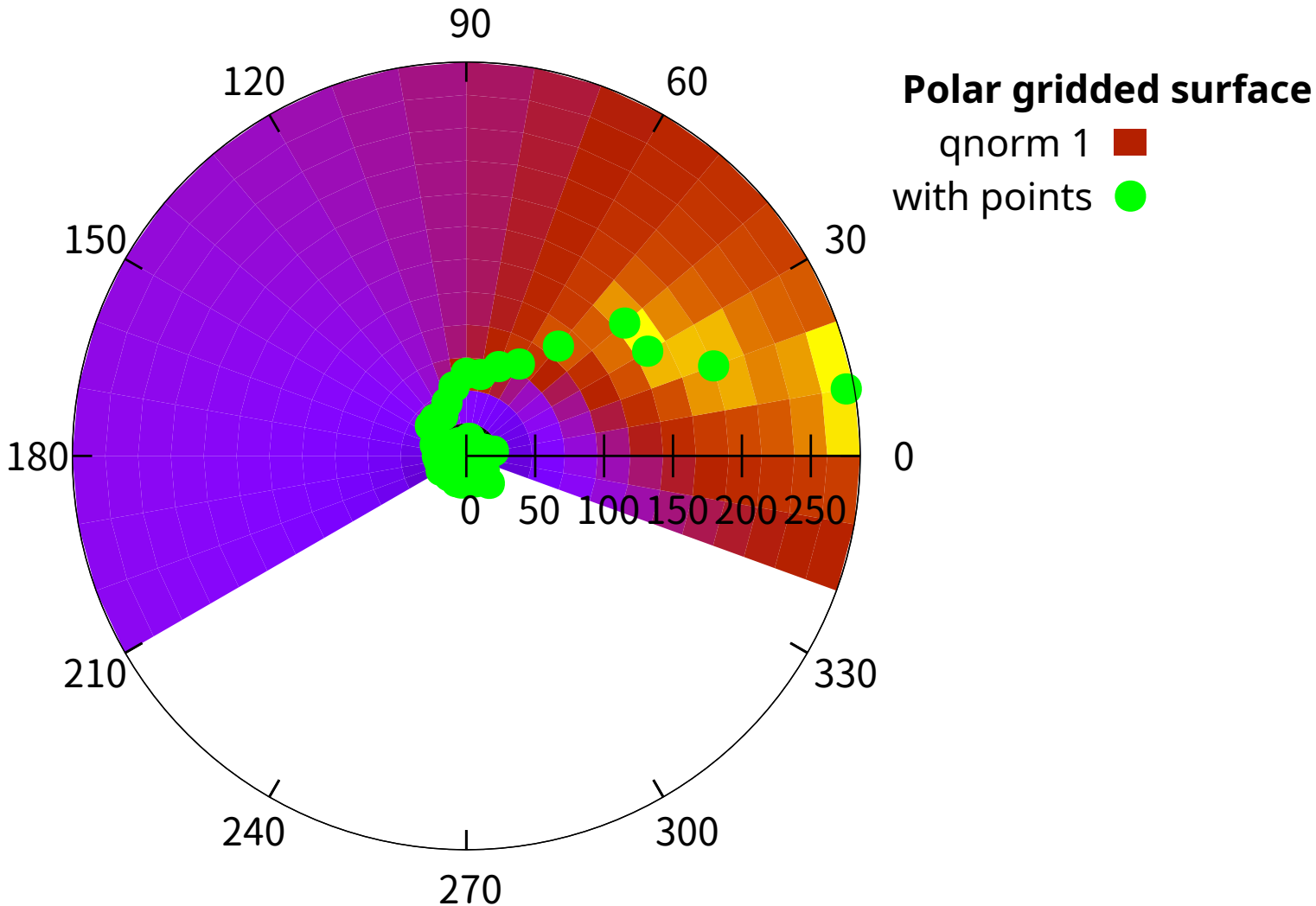

Antenna Pattern
200
150
100
50
0
-50
-100
-150
-200
dBSM
-200
-100
0
Azimuth

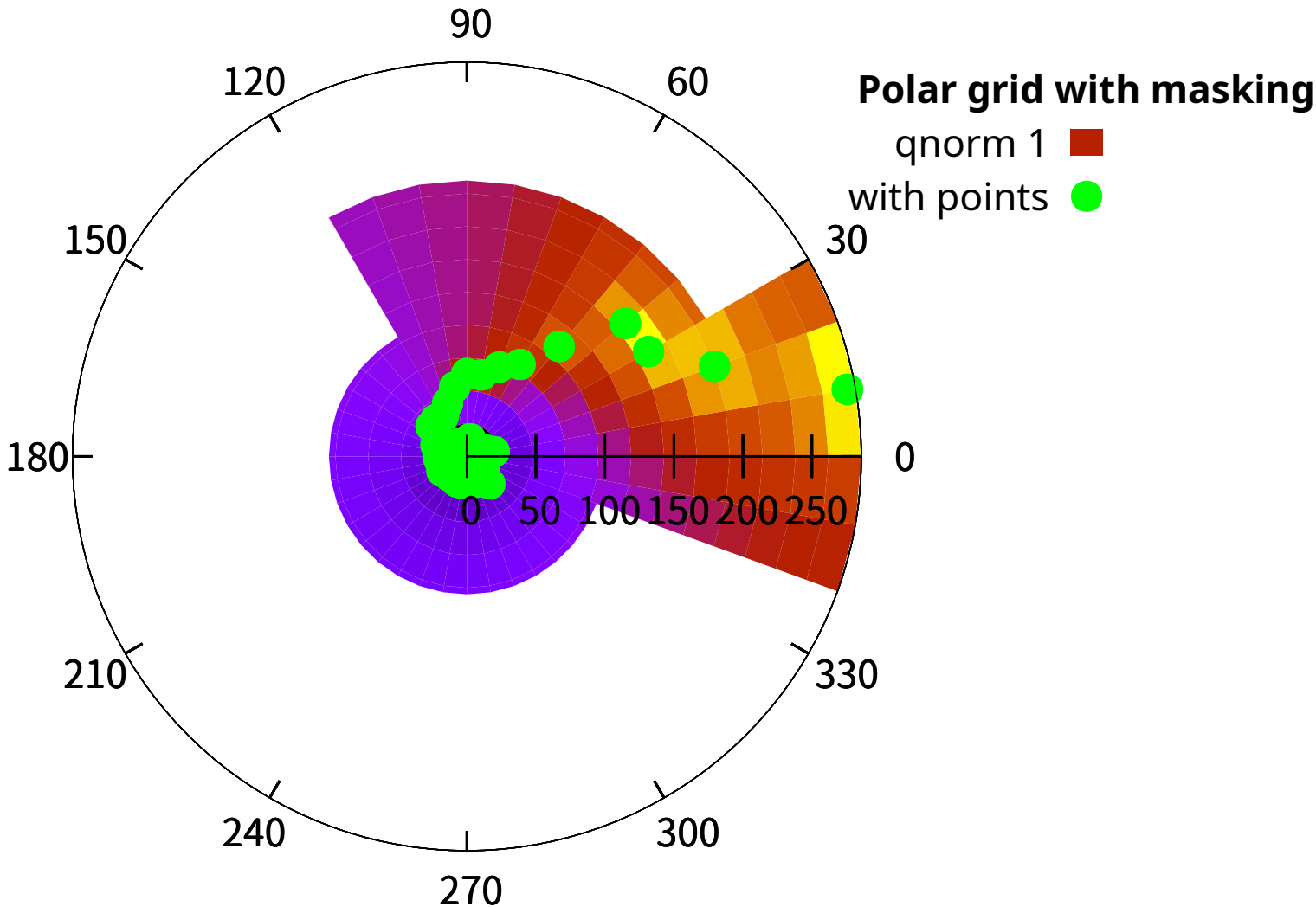
火 25 2月 2025

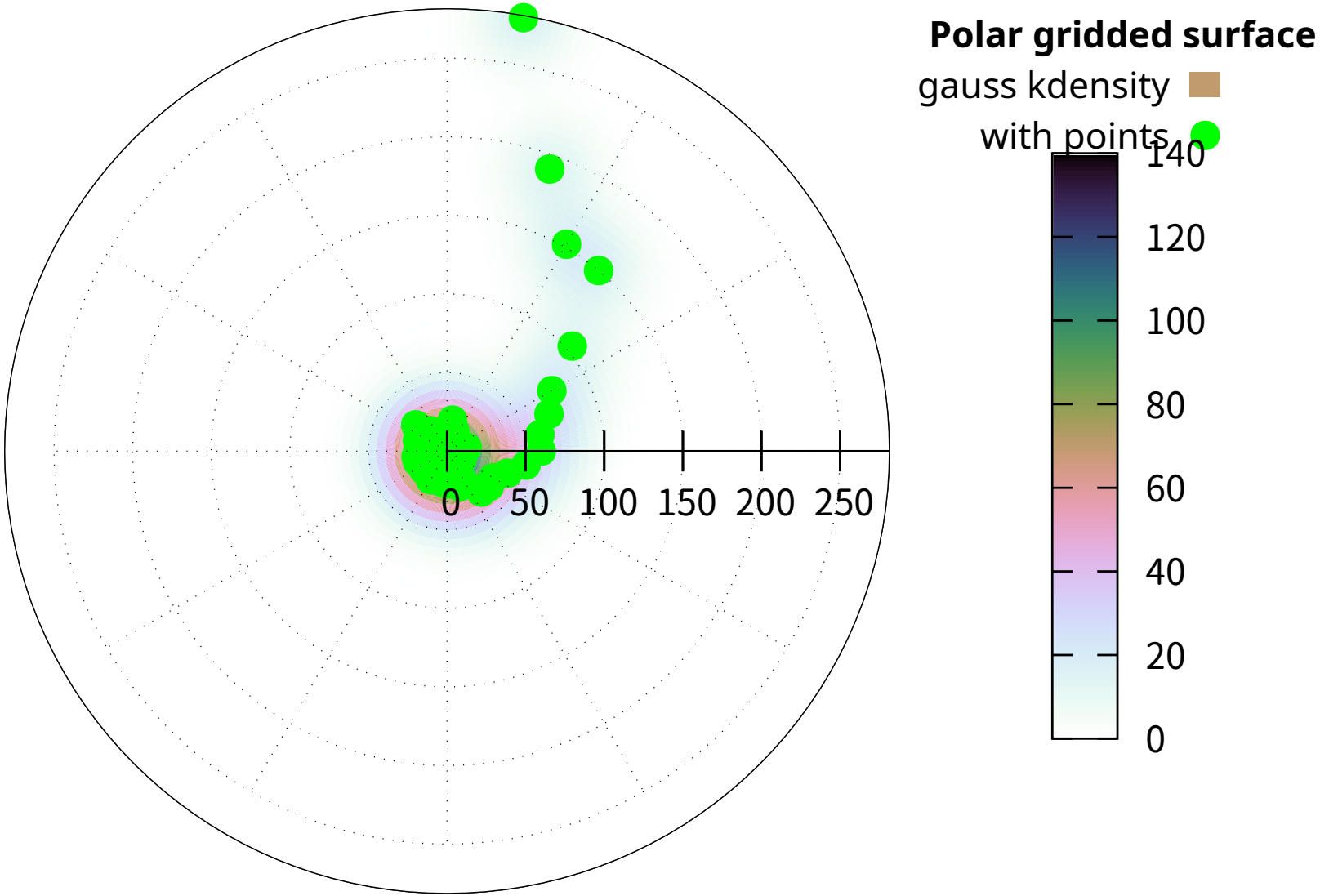
log scale polar axis, trange in degrees

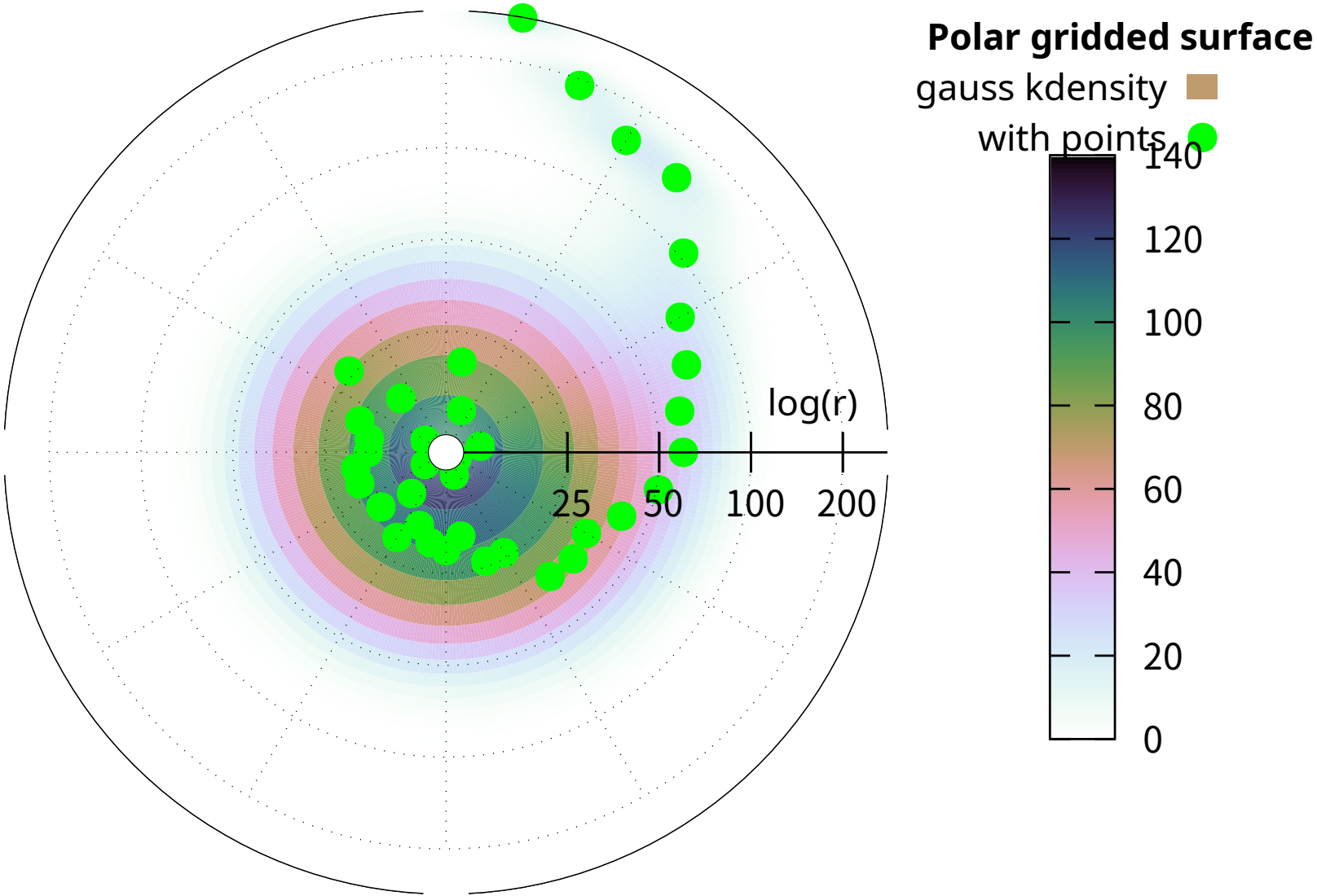



yerrors in polar mode
polar error bars
0
0.5
1
1.5

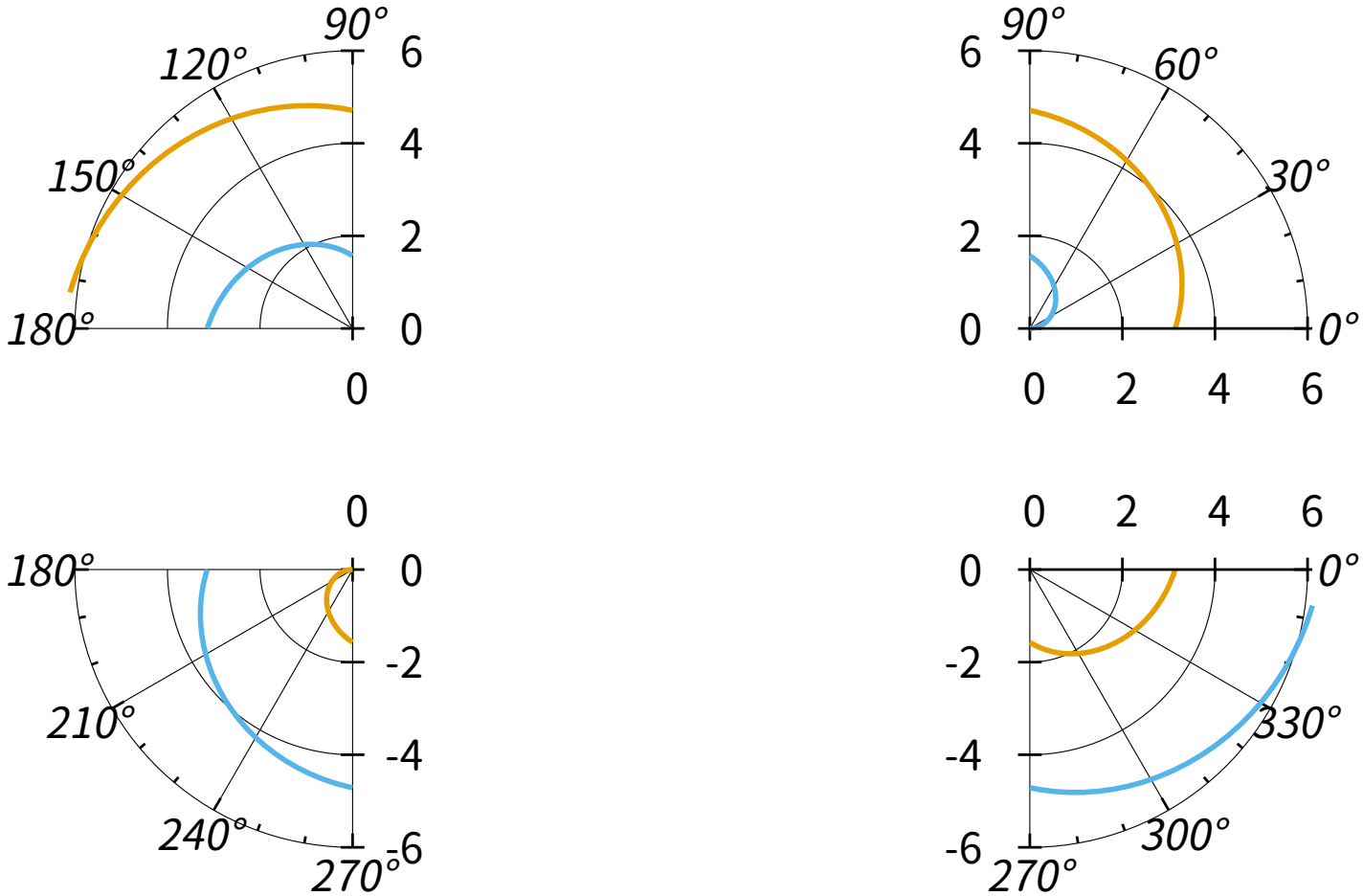

Polar gridded surface
qnorm 1
with points
90
60
30
0
330
300
270
240
210
180
150
120
0
50
100
150
200
250


Polar grid with masking
qnorm 1
with points
90
60
30
0
330
300
270
240
210
180
150
120
0
50
100
150
200
250

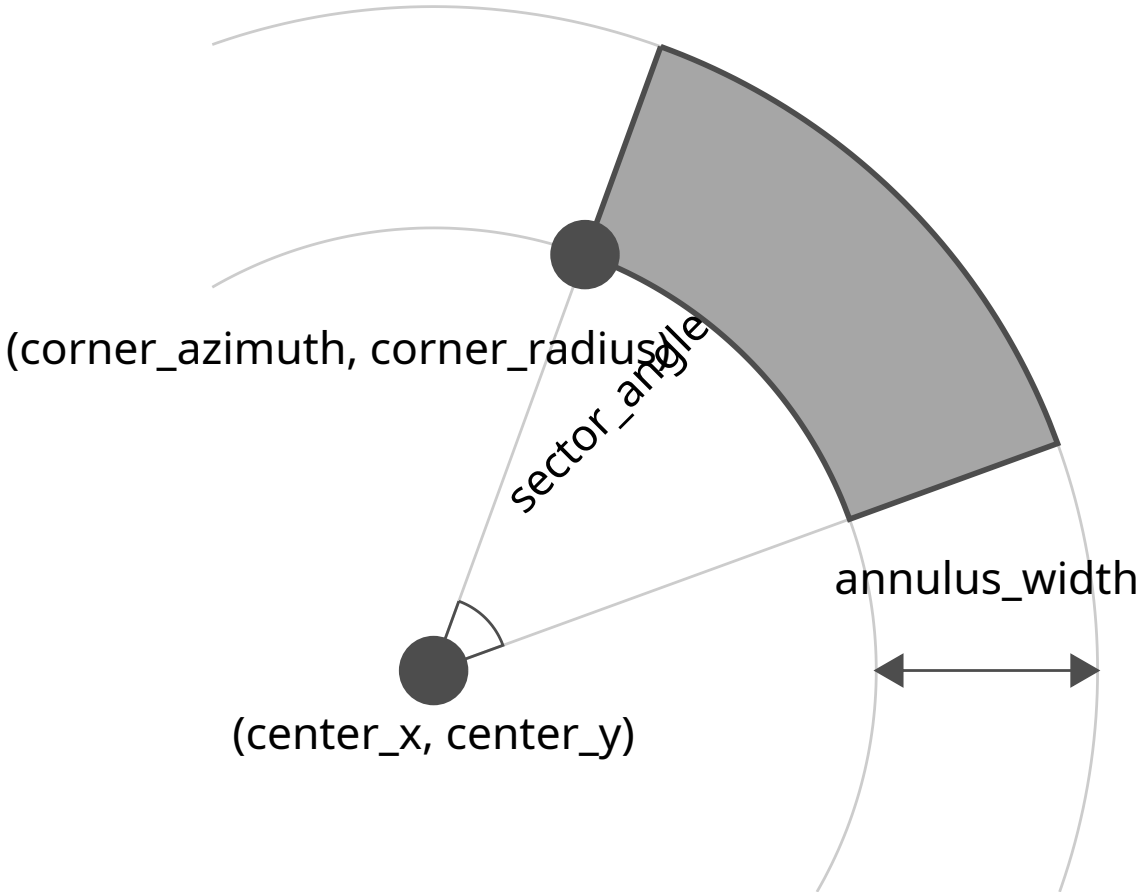

Polar gridded surface
gauss kdensity
with points
0
50
100
150
200
250
140
120
100
80
60
40
20
0


Polar gridded surface
gauss kdensity
with points
log(r)
25
50
100
200
140
120
100
80
60
40
20
0

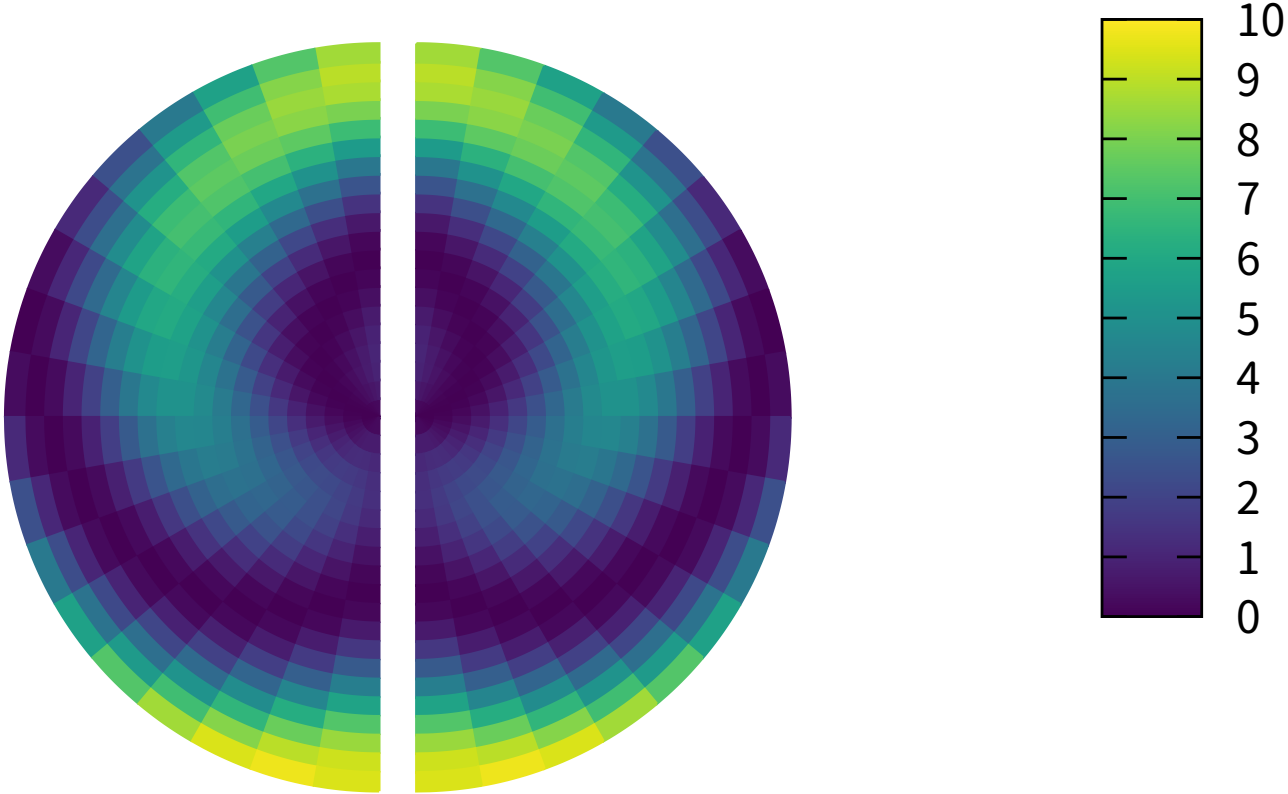
Polar Quadrants



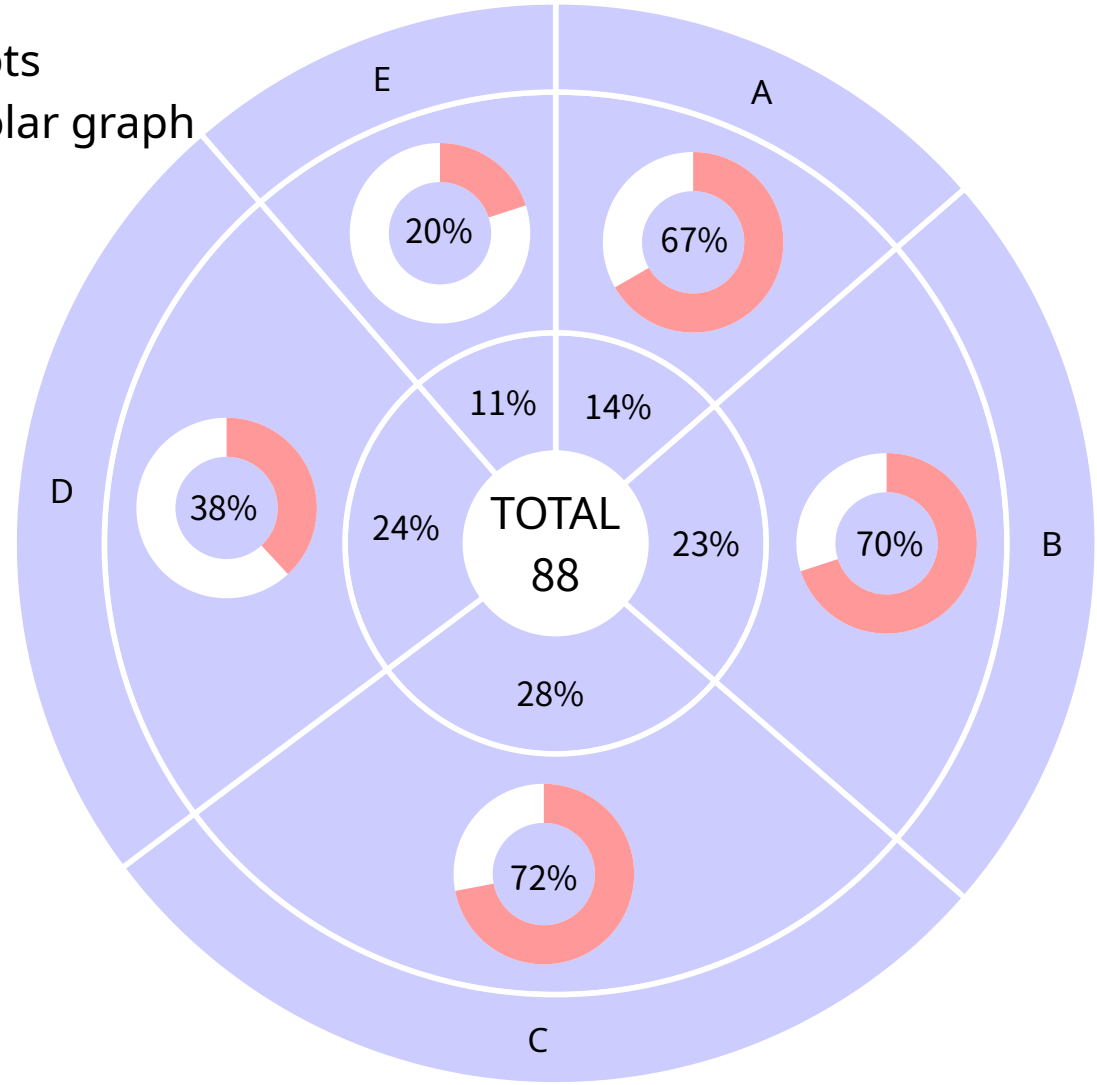
A single annular sector in plot style "with sectors"



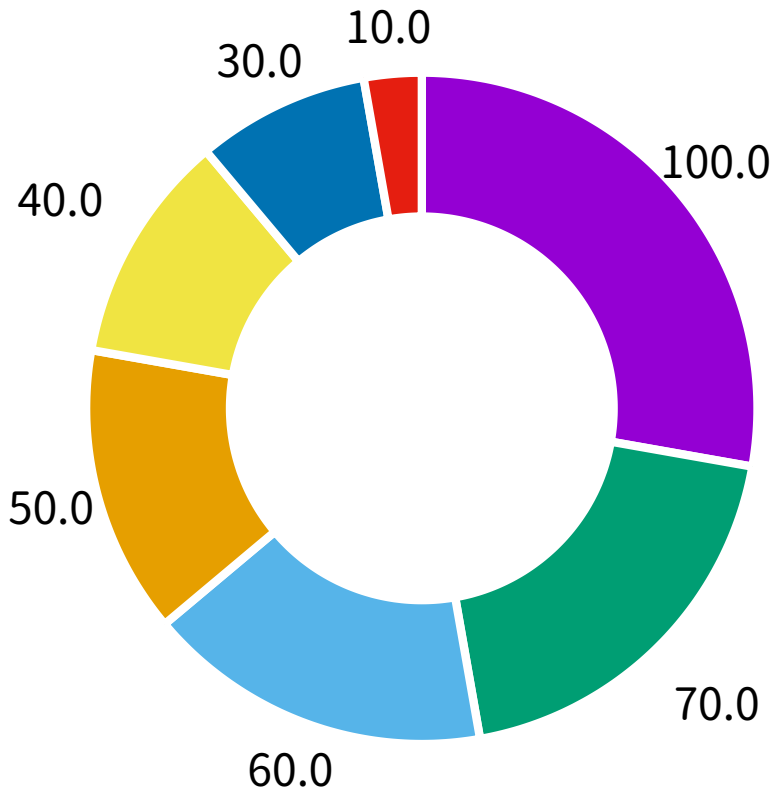
Polar heatmap composed of sectors
positioned on a cartesian x/y plane



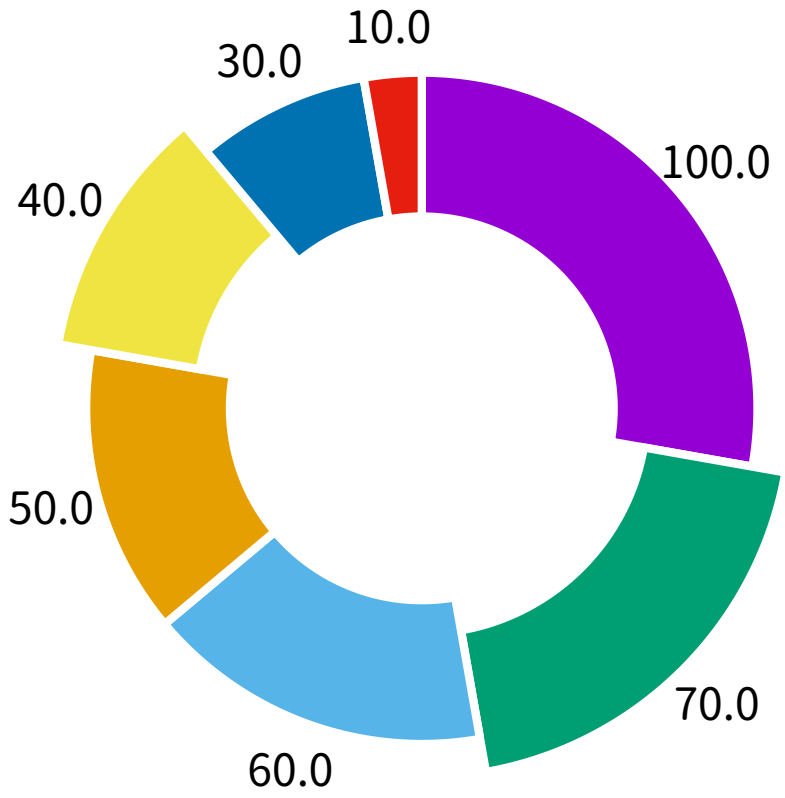
Multiple sector plots
positioned on a polar graph



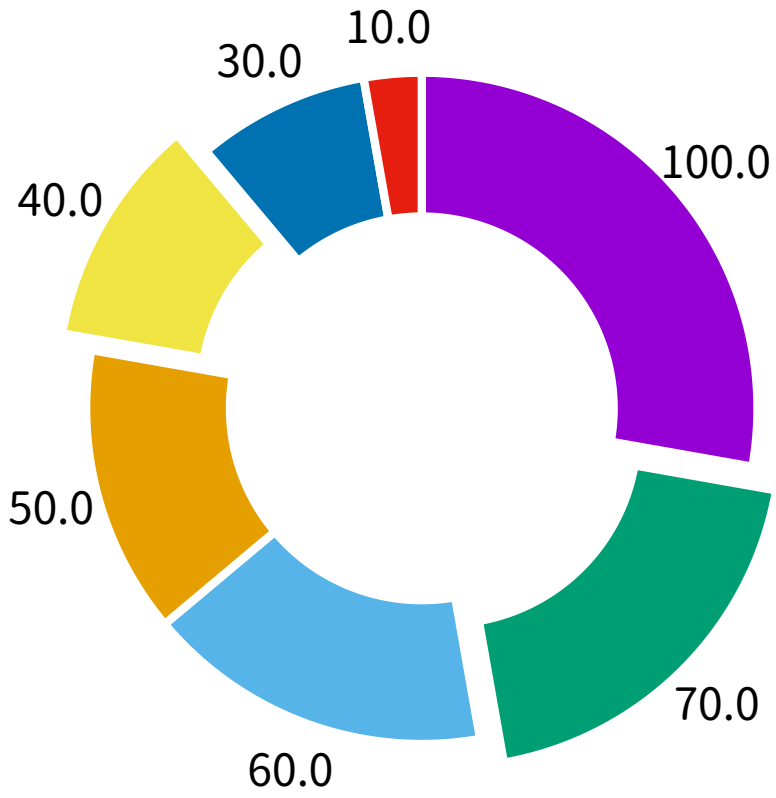
Pie (donut) chart drawn with sectors



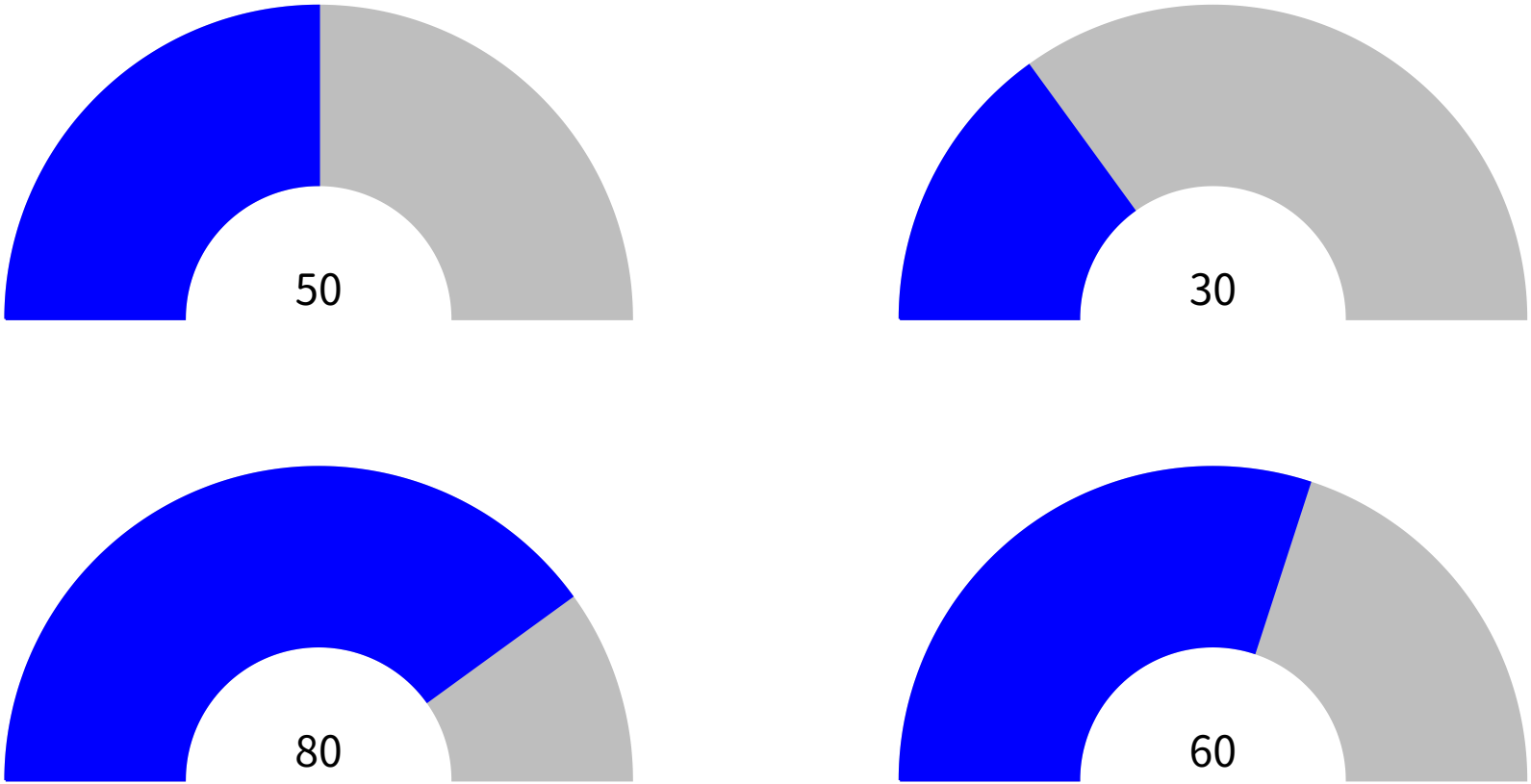
Pie (donut) chart some slices have radial shift

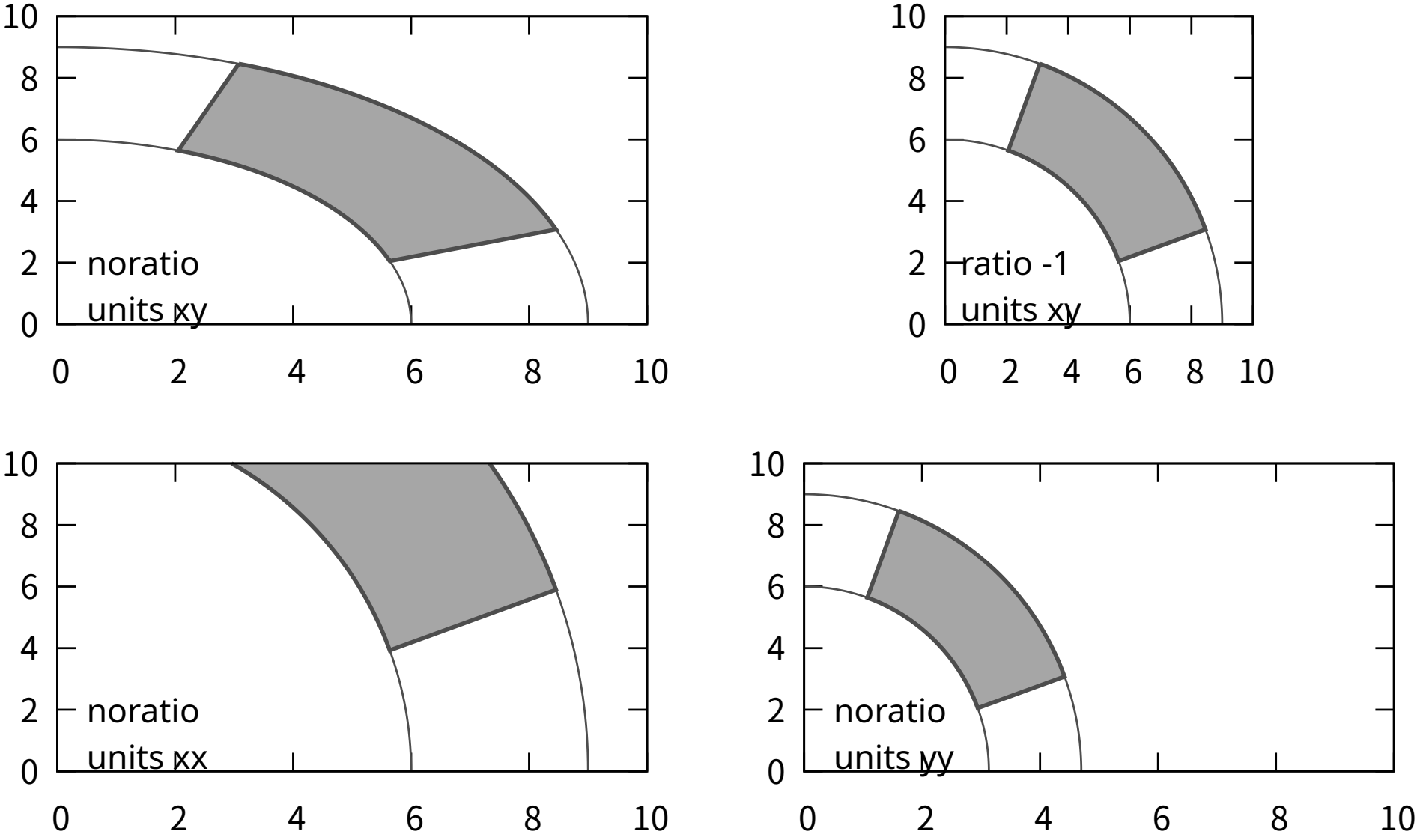


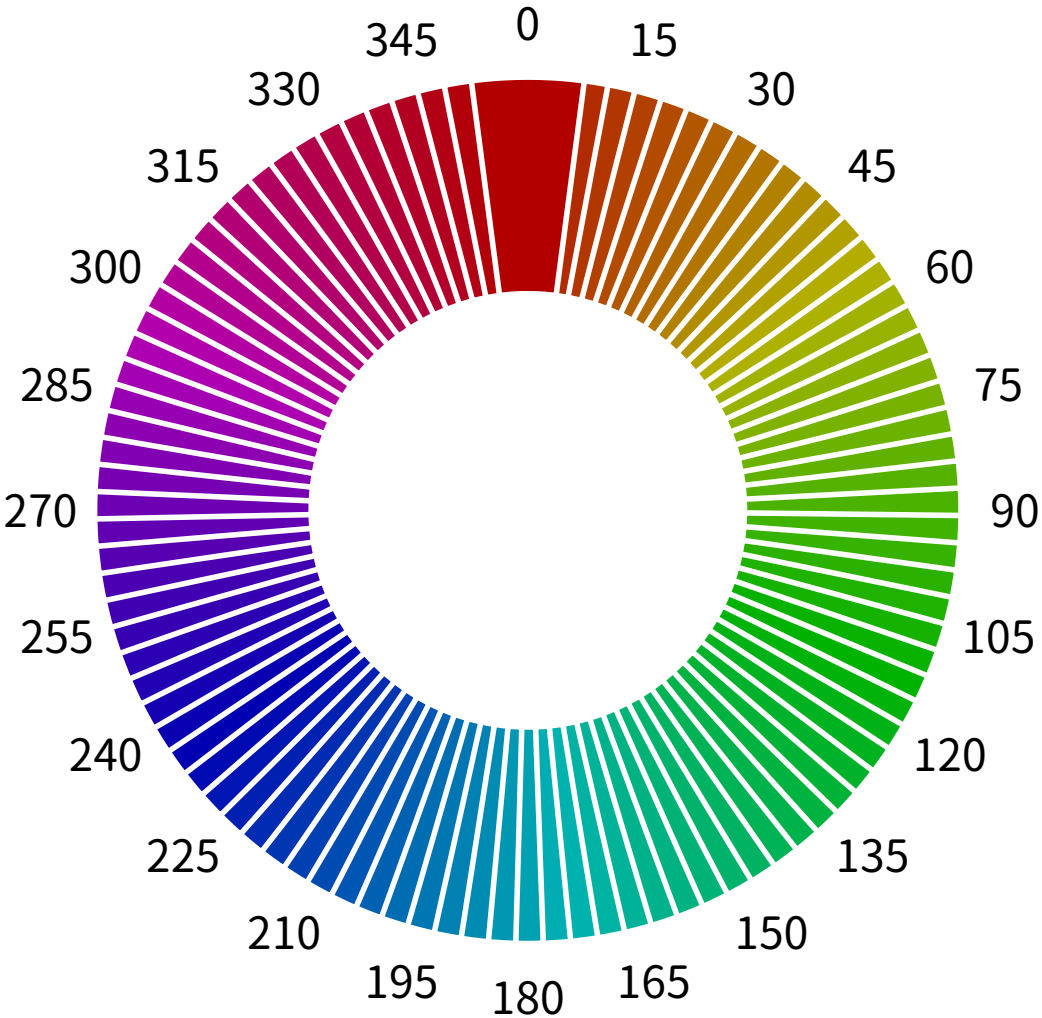
Pie (donut) chart some slices have offset origin



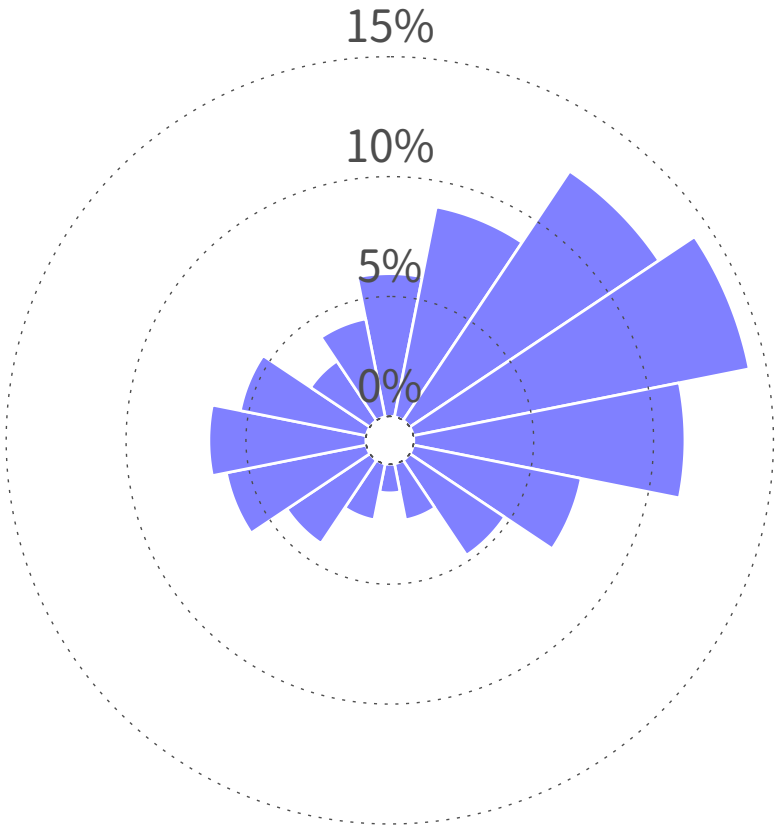
Dial charts drawn with sectors

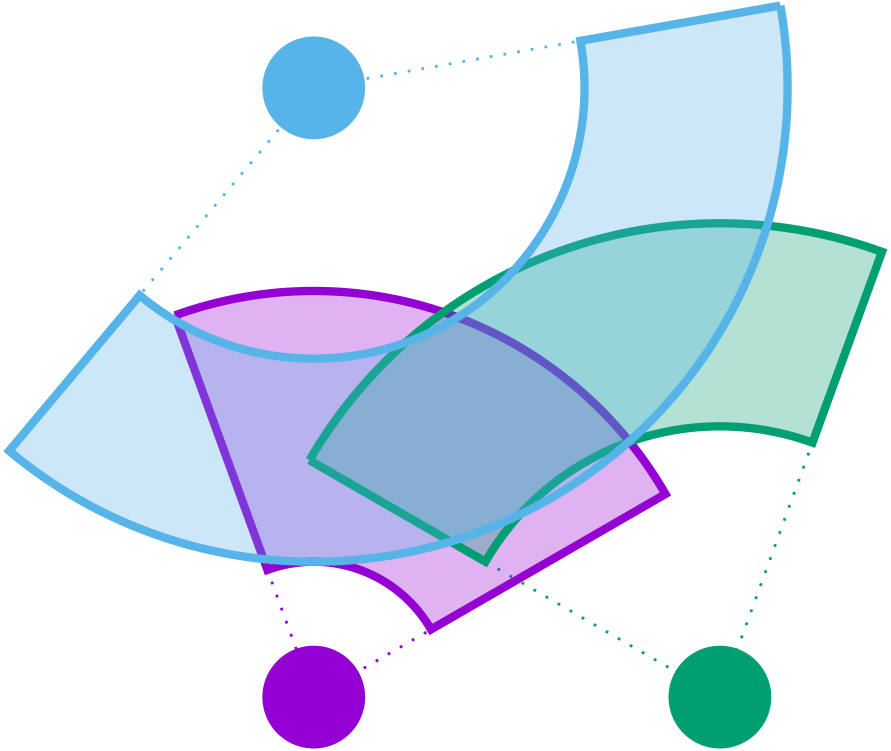


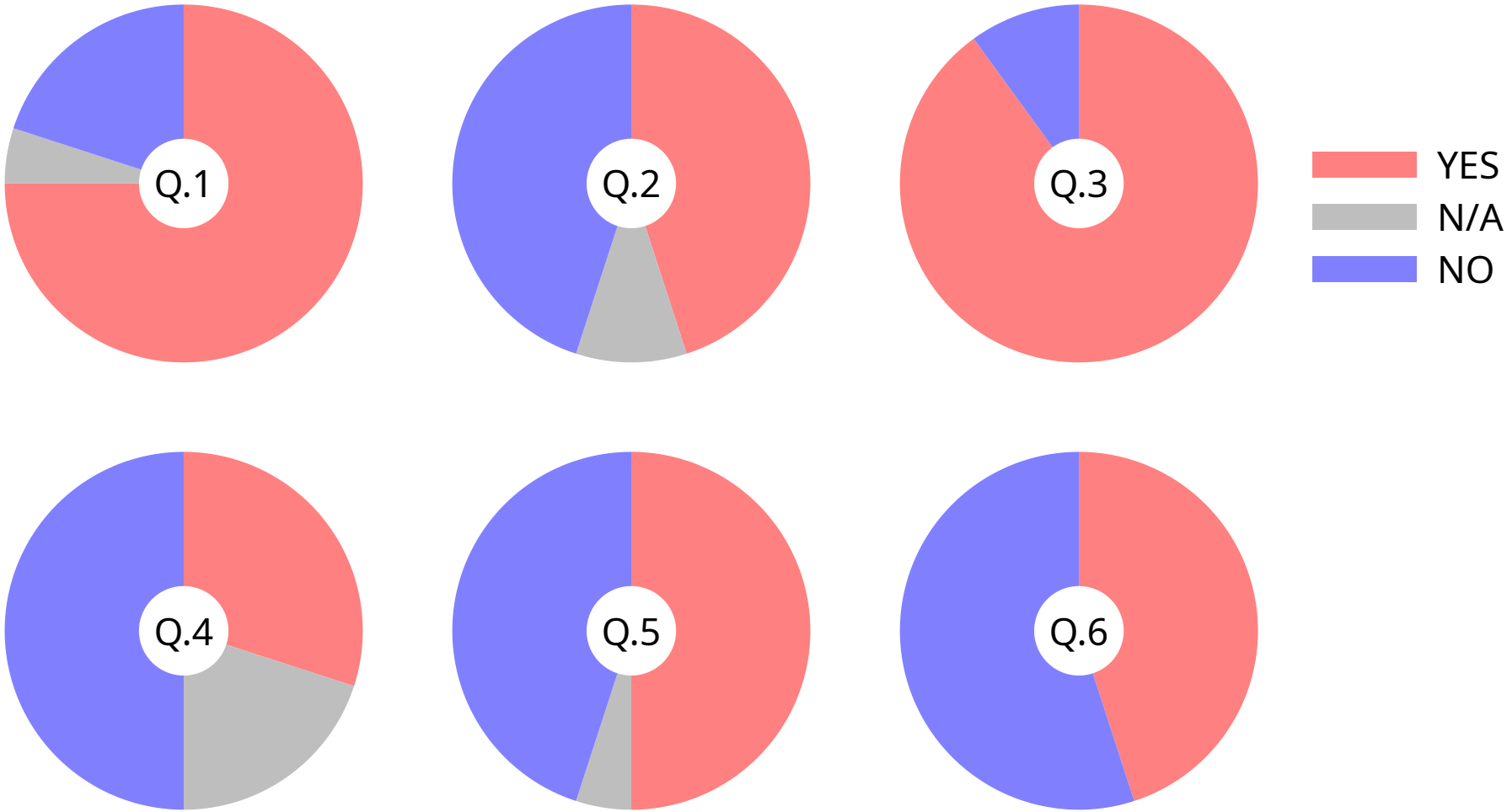

noratio
units xy
ratio -1
units xy
noratio
units xx
noratio
units yy
0
2
4
6
8
10

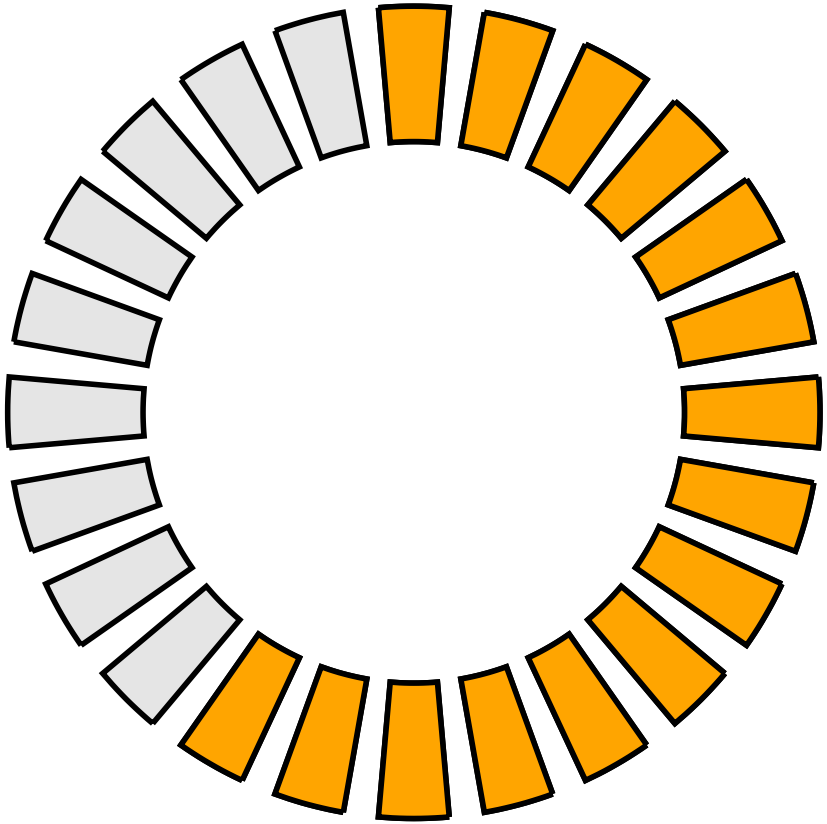

0
15
30
45
60
75
90
105
120
135
150
165
180
195
210
225
240
255
270
285
300
315
330
345

Wind rose (polar coordinate histogram using sectors)

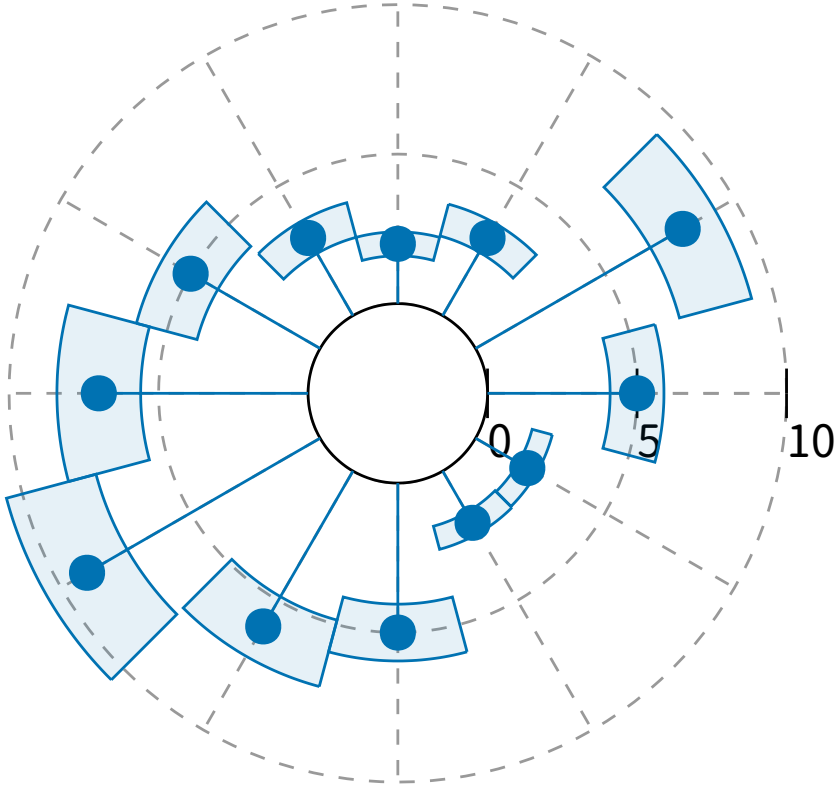


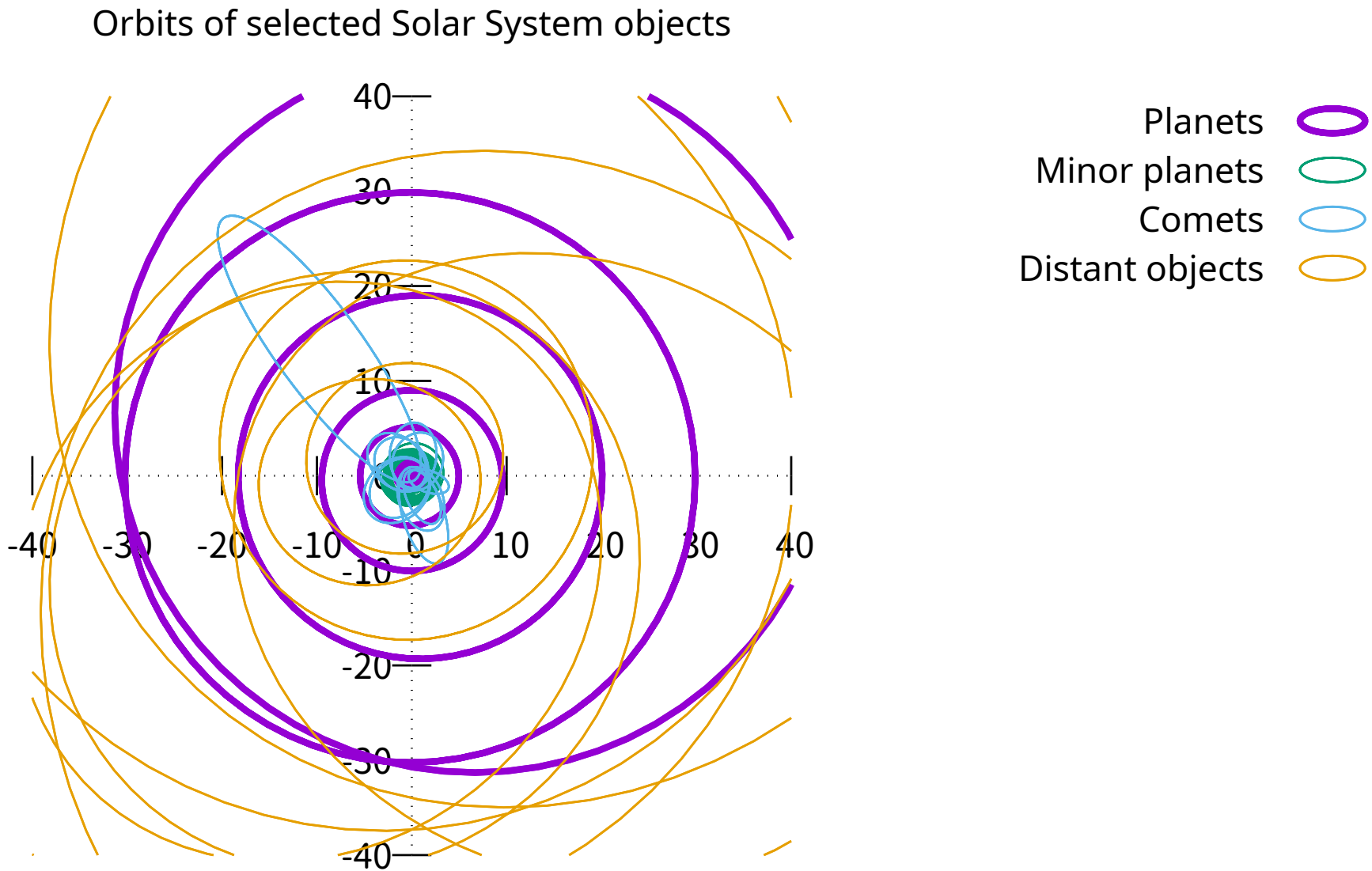


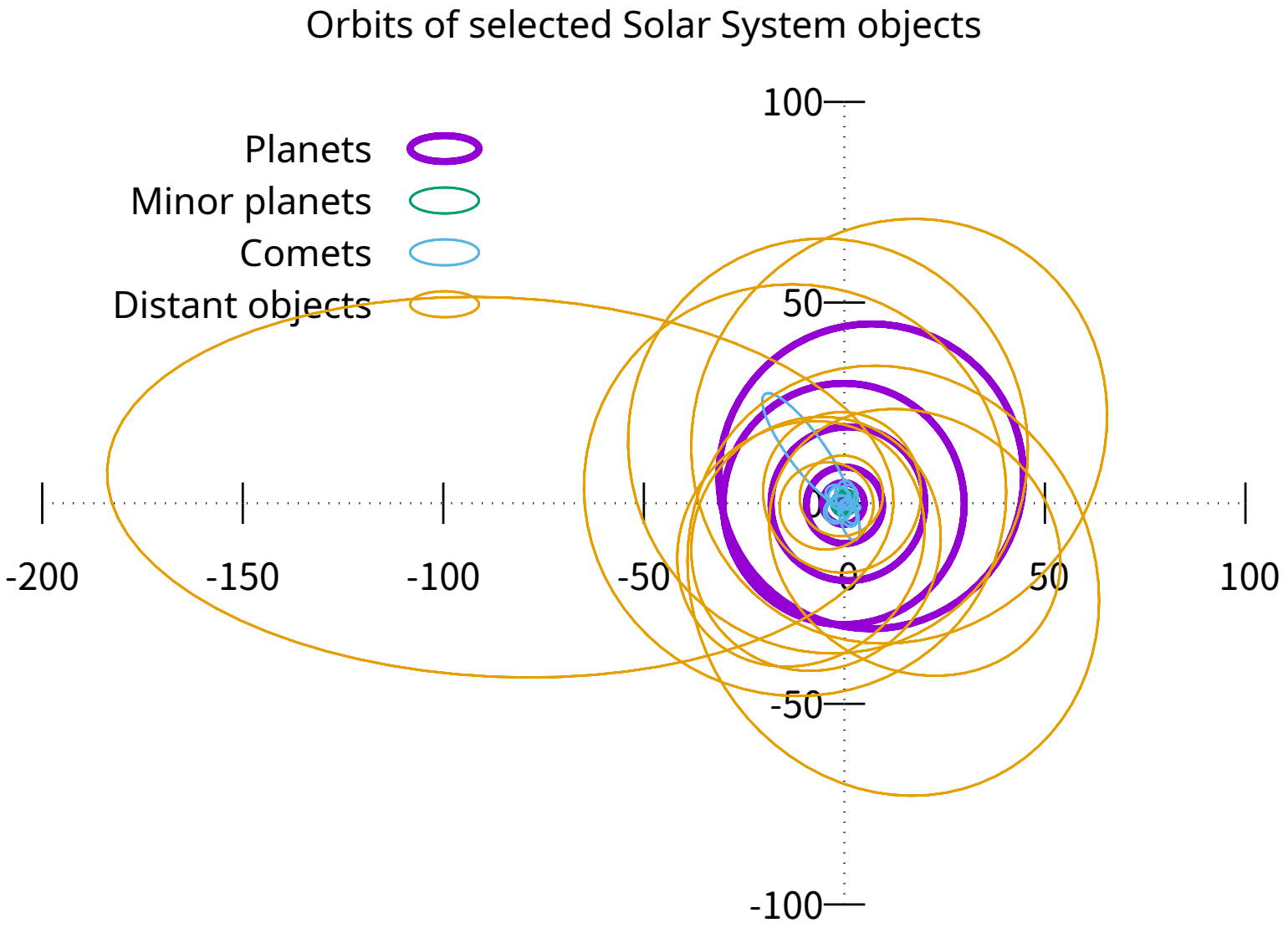

Q.1
Q.2
Q.3
Q.4
Q.5
Q.6
YES
N/A
NO



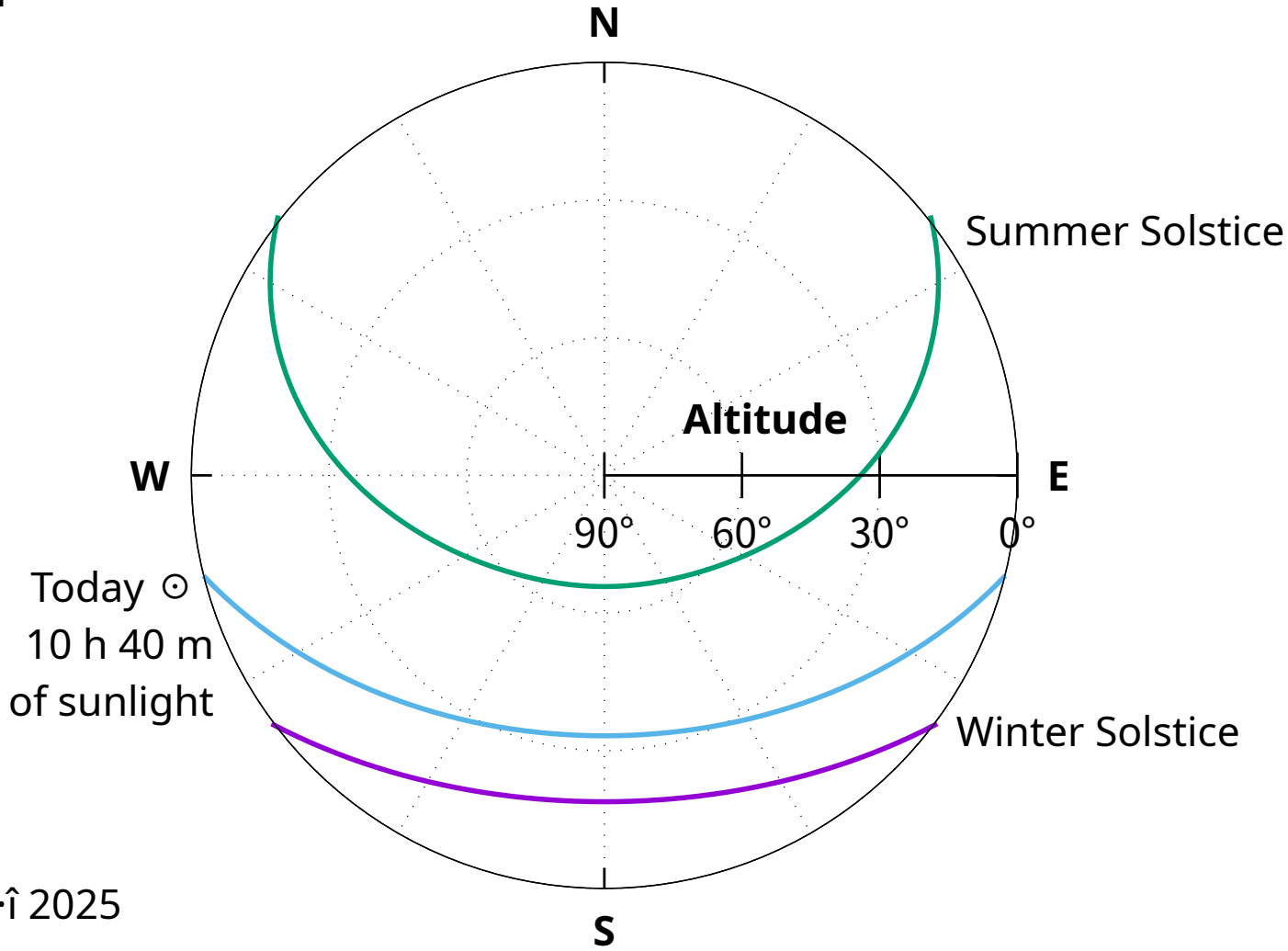
Polar coordinate points with both radial and angular error represented by sectors




Orbits of selected Solar System objects
40
30
20
10
-10
-20
-30
-40
-40
-30
-20
-10
0
10
20
30
40
Planets
Minor planets
Comets
Distant objects

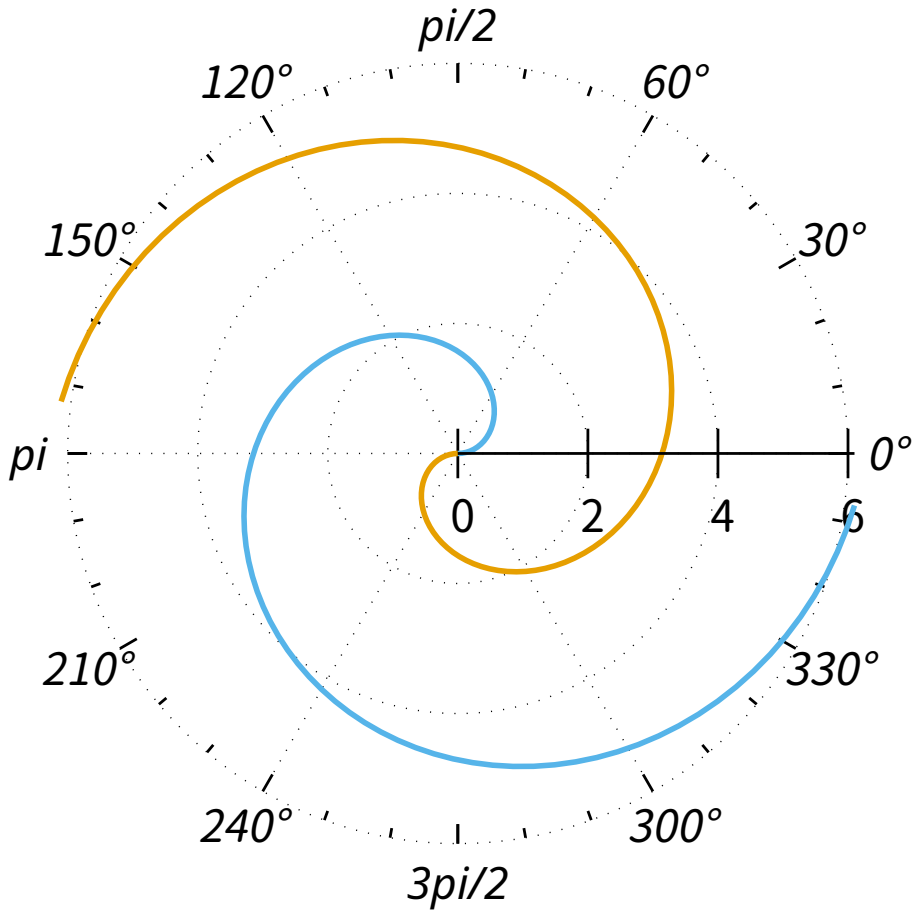

Orbits of selected Solar System objects
Planets
Minor planets
Comets
Distant objects
100
50
0
-50
-100
-200
-150
-100
-50
0
50
100

Solar path at
Latitude 47.67 N

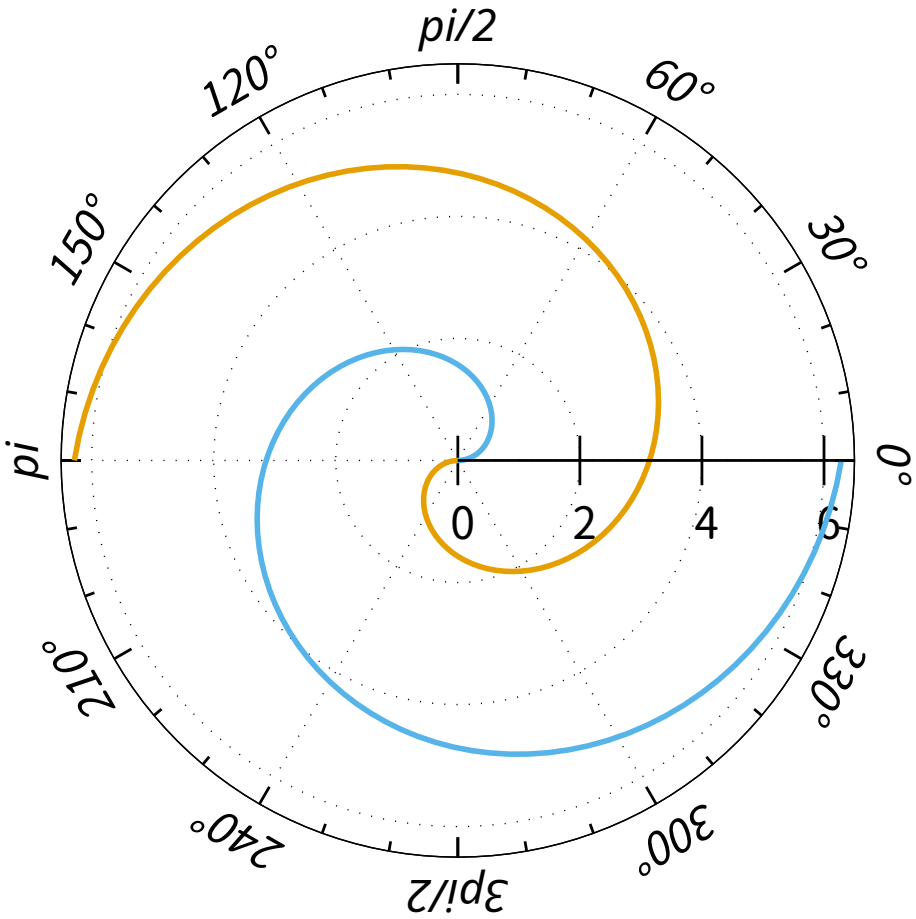


Seattle - 25 2·î 2025

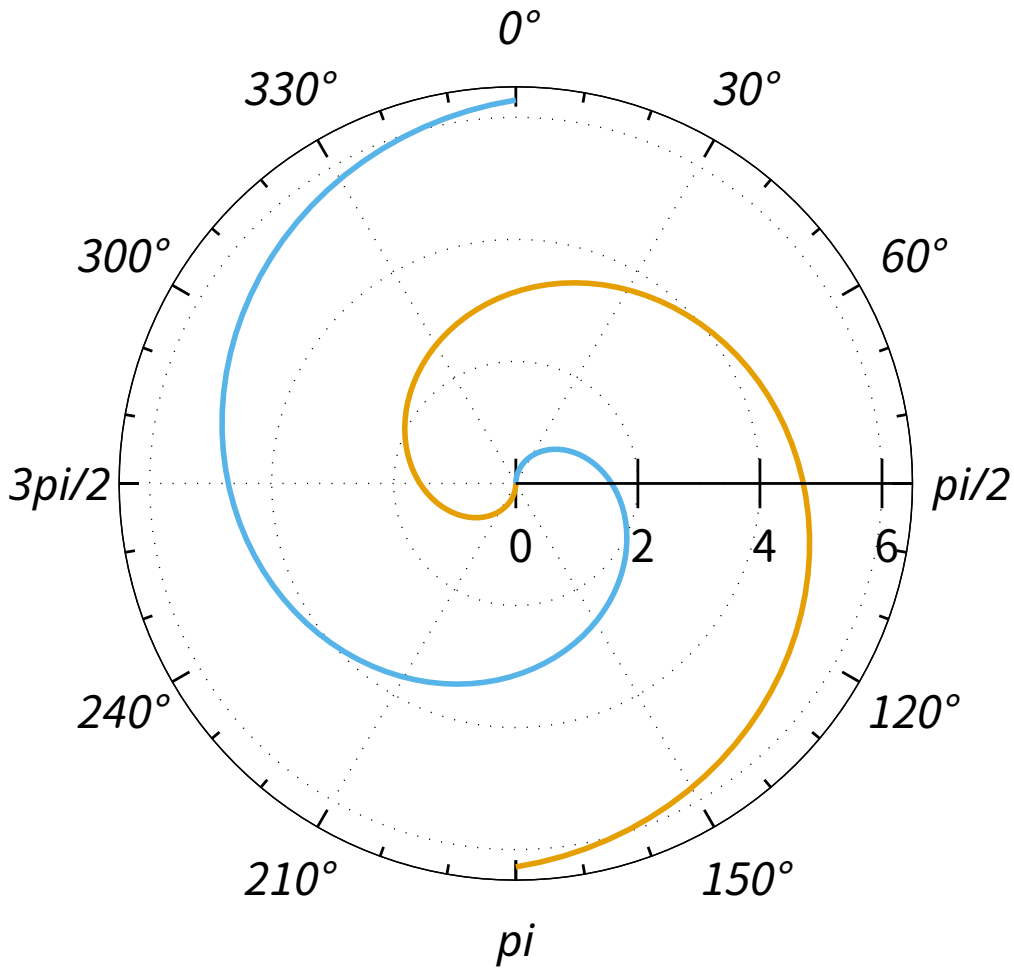
Angle labels (ttics) for polar plots

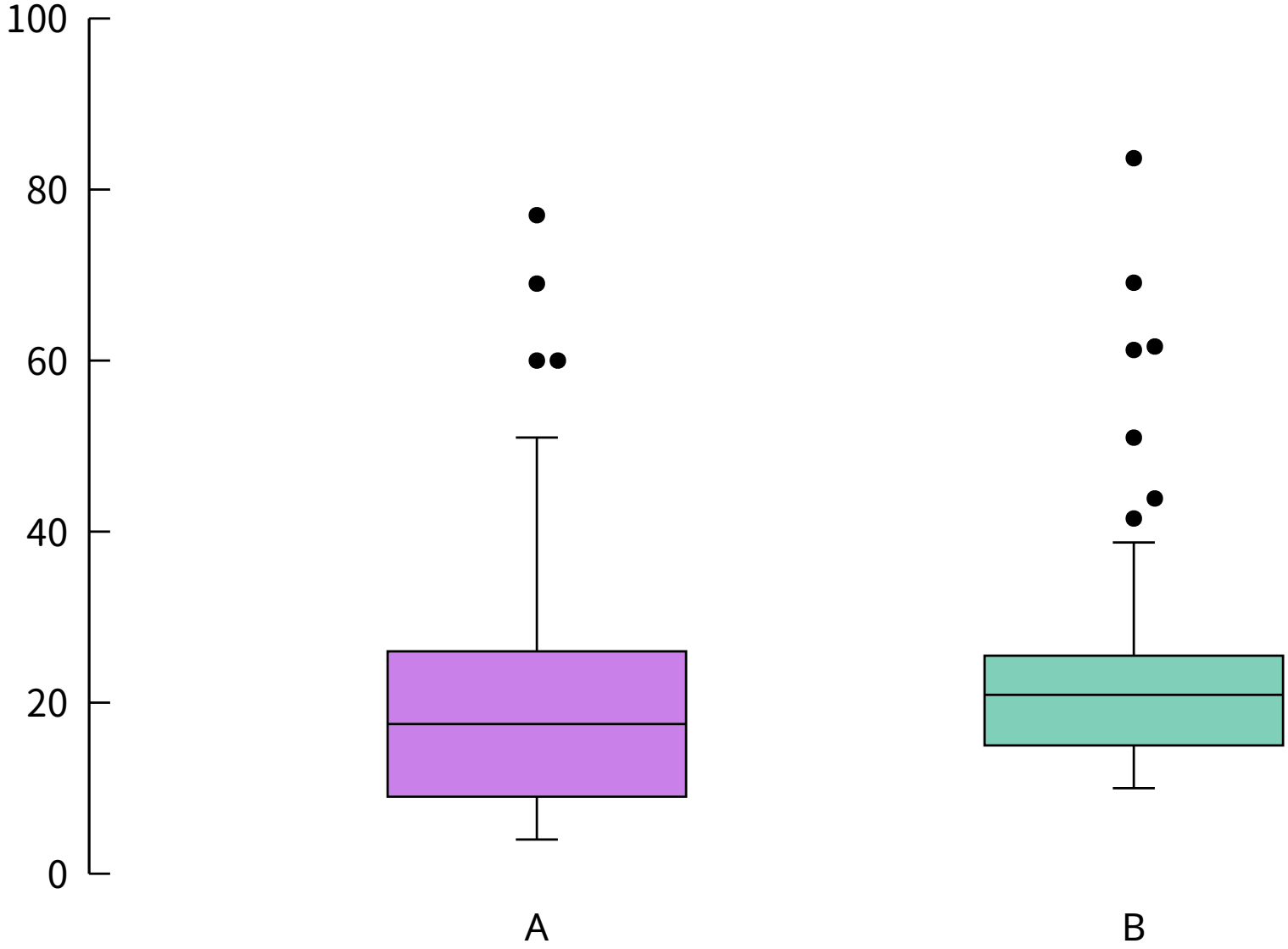


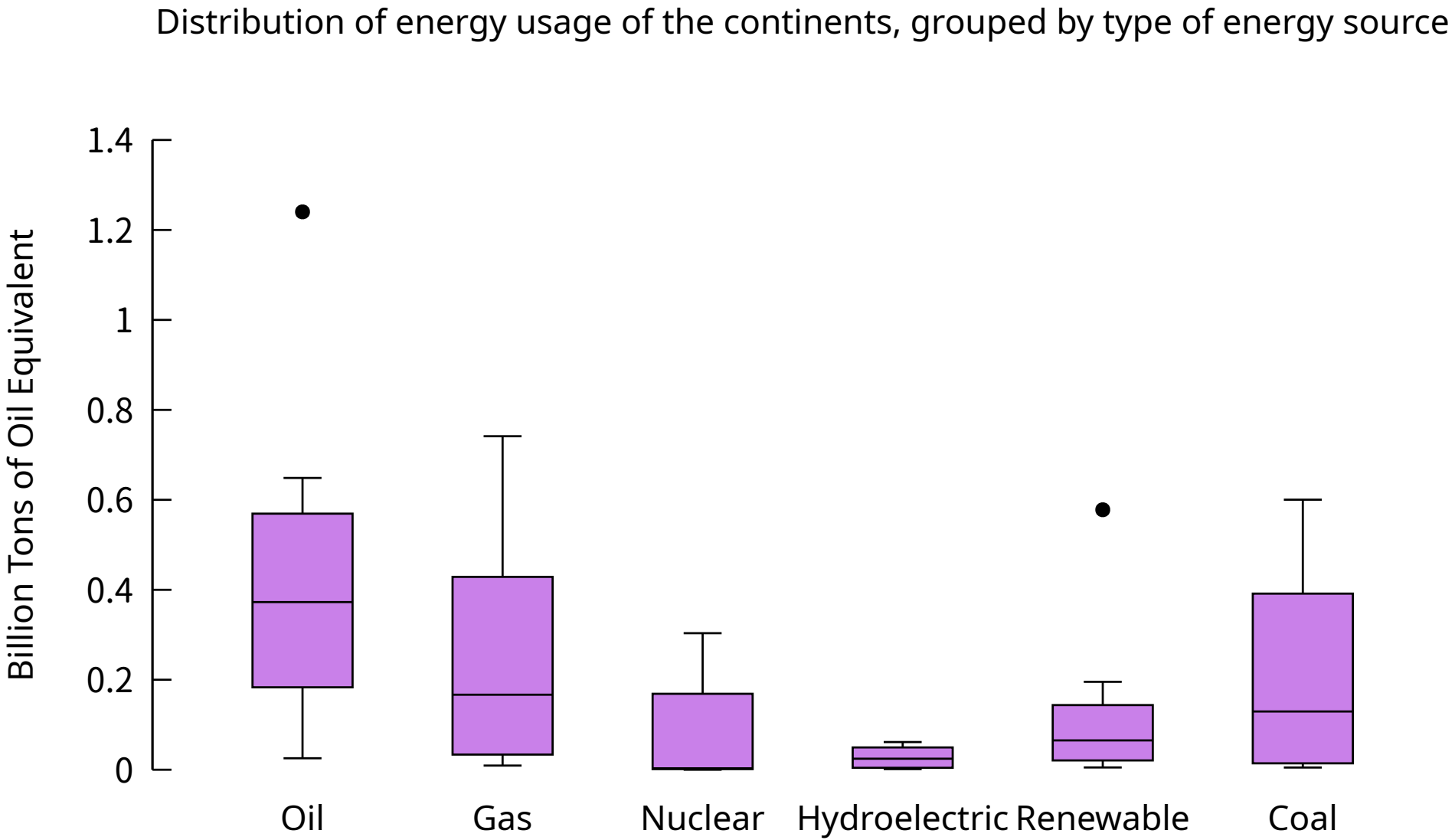
Polar plot with border and rotated labels for ttics



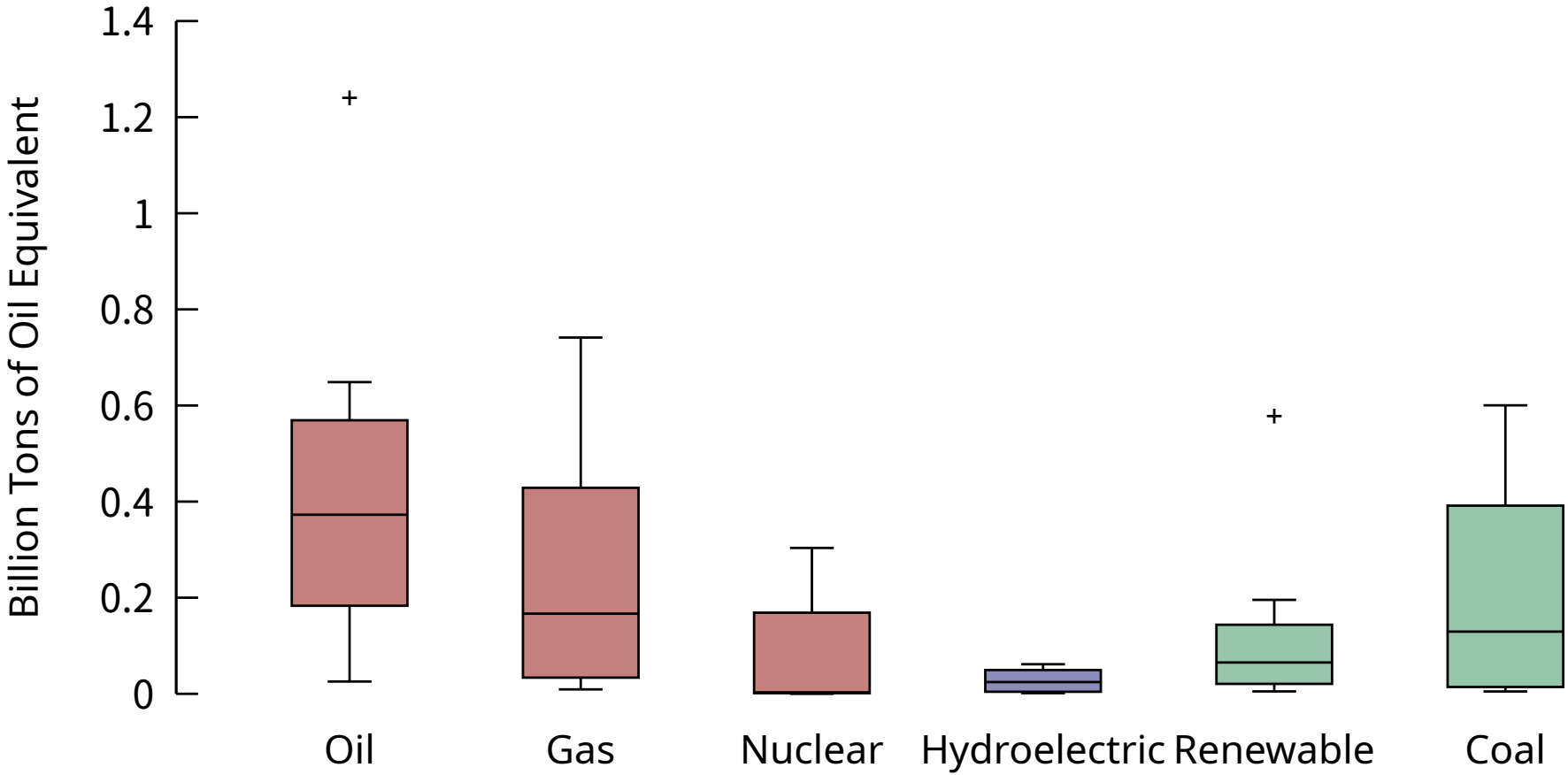
Theta origin at top, increasing clockwise

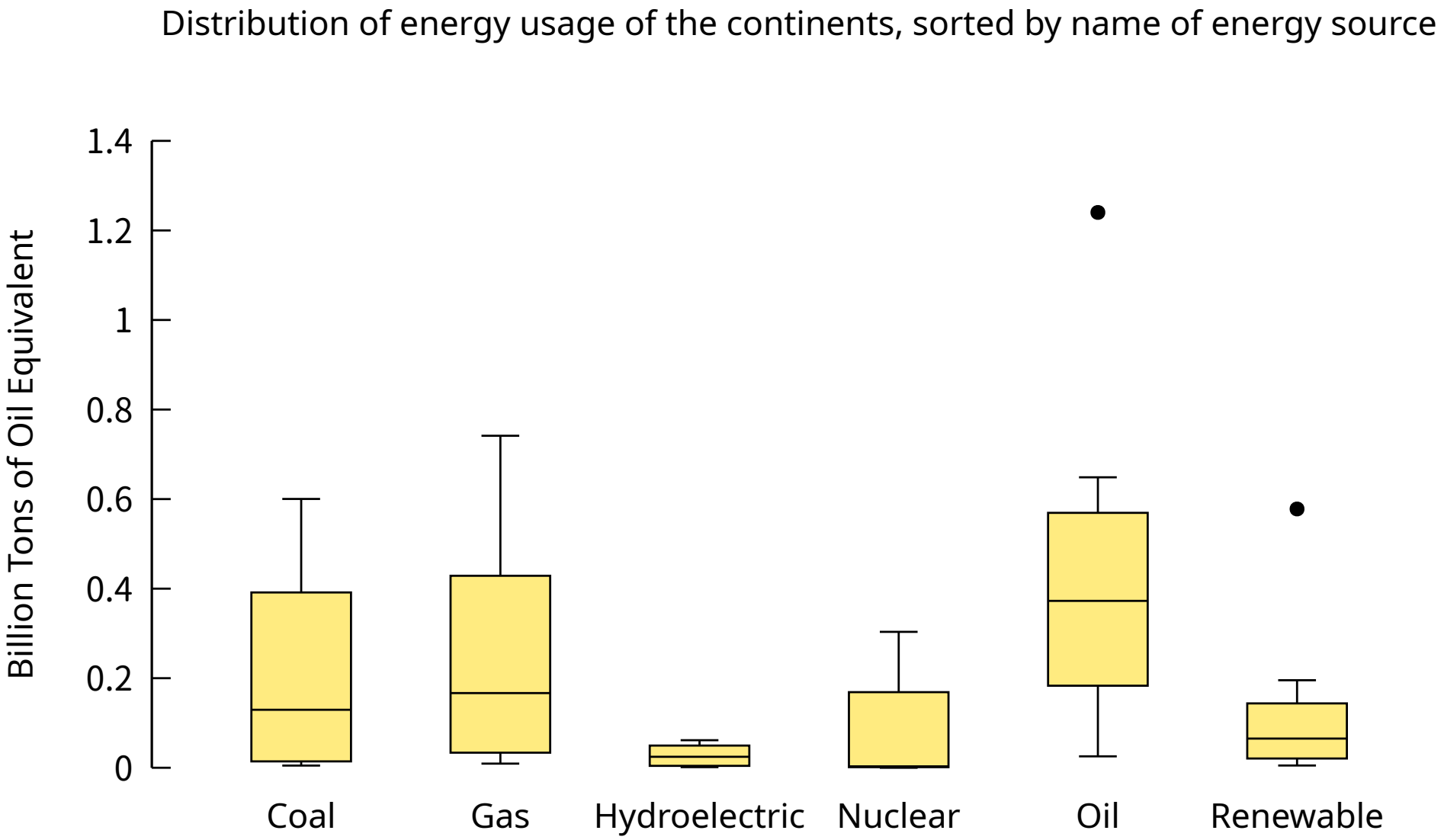


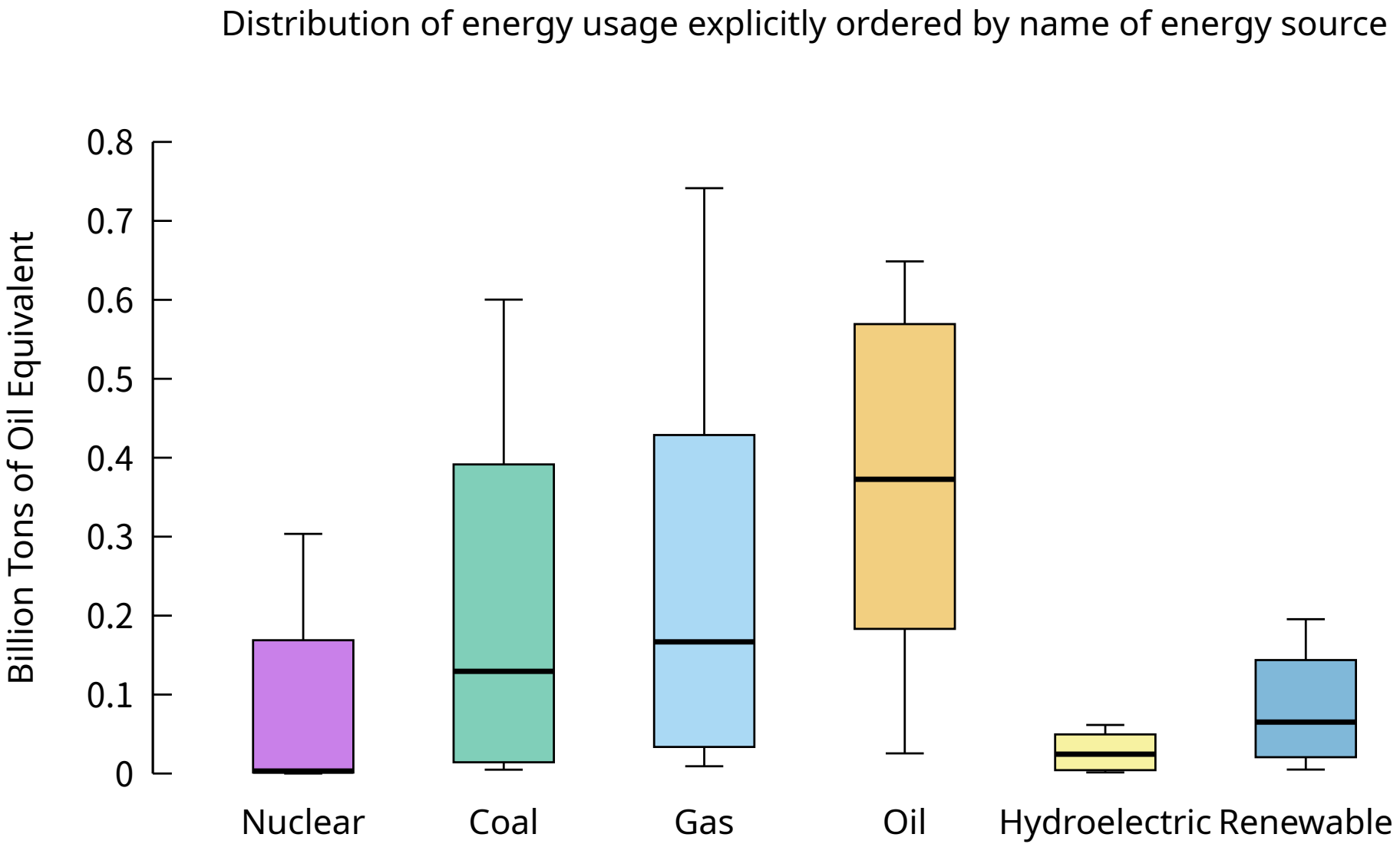

100
80
60
40
20
0
A
B

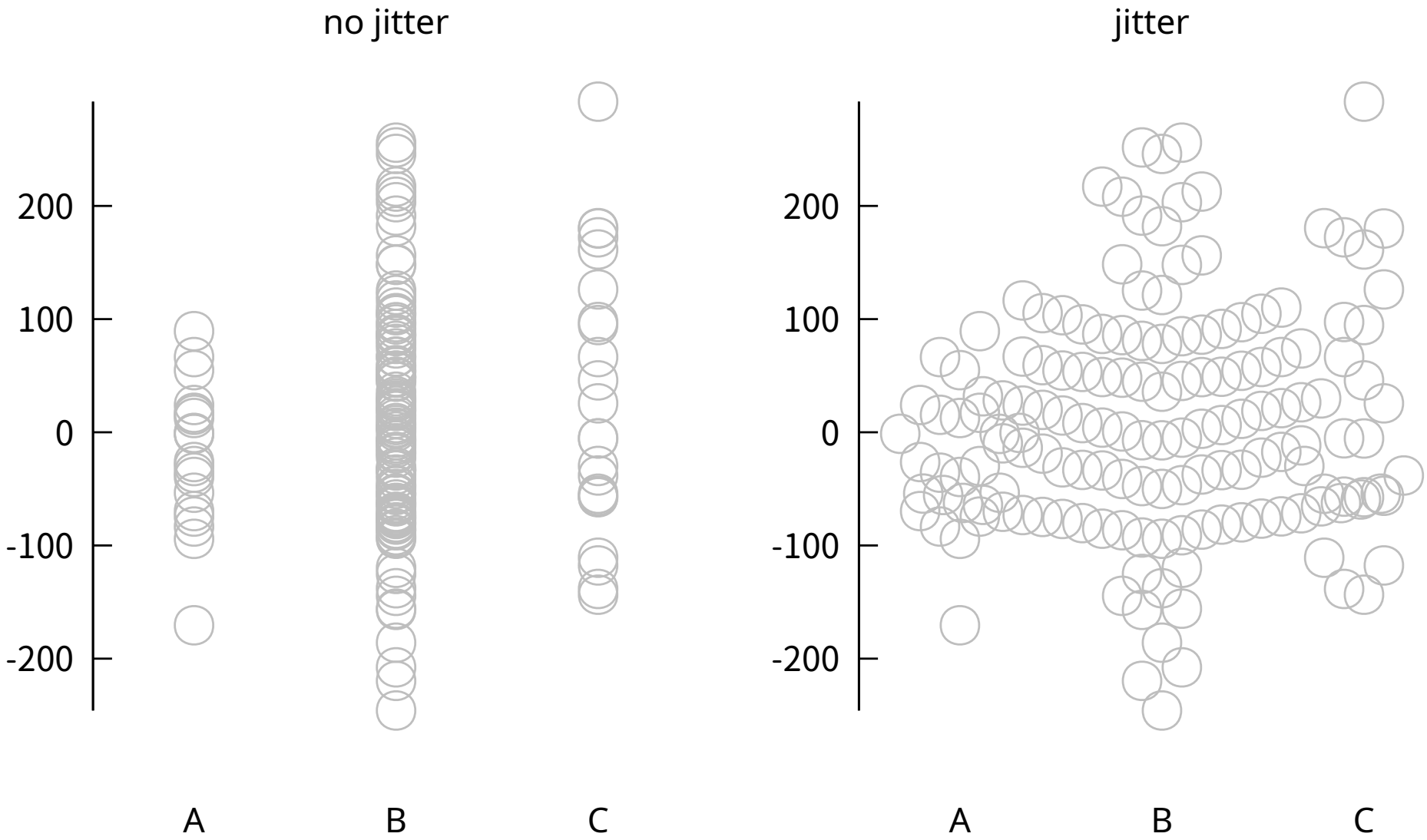

Distribution of energy usage of the continents, grouped by type of energy source
Billion Tons of Oil Equivalent
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Oil
Gas
Nuclear
Hydroelectric
Renewable
Coal

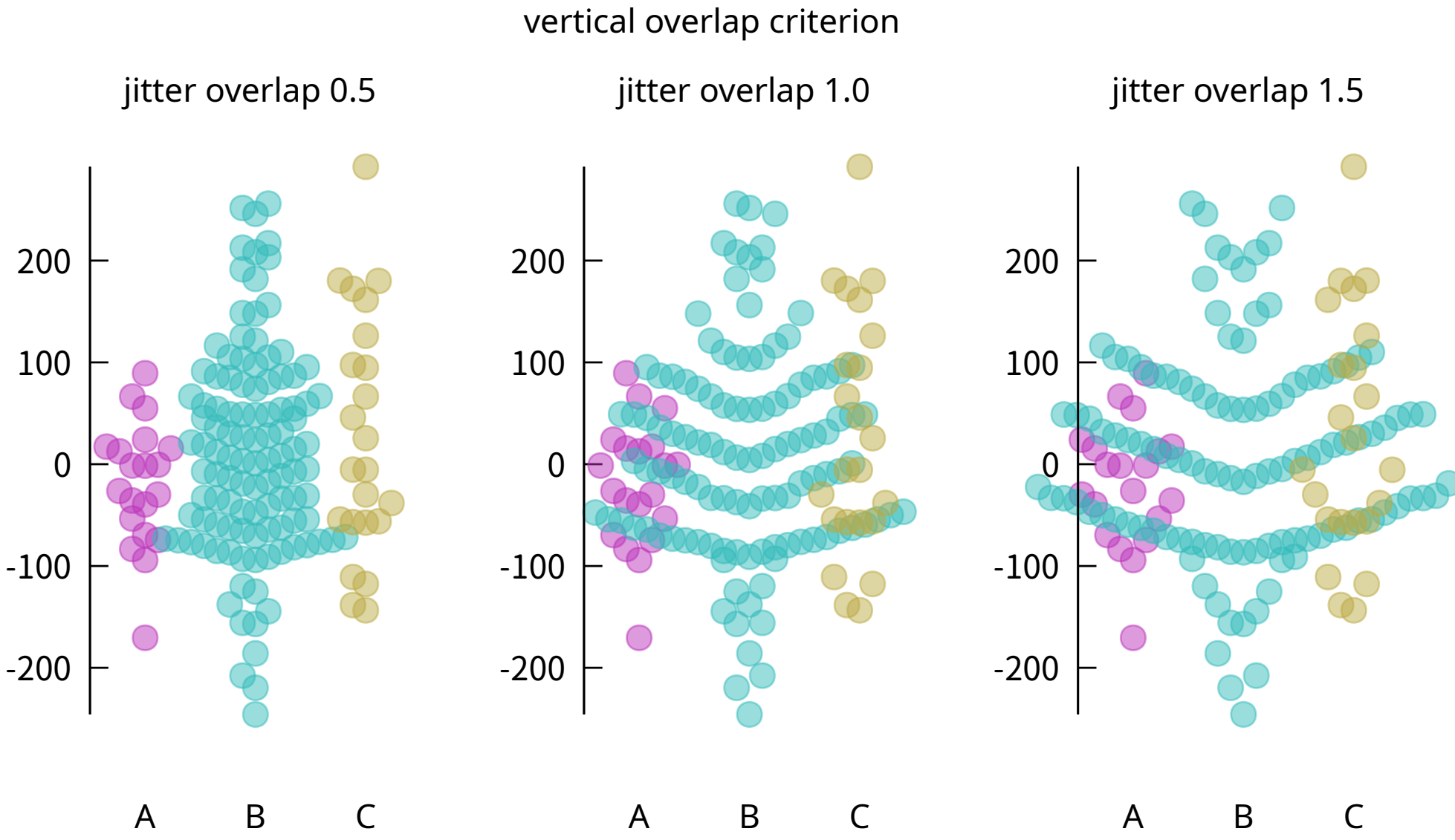
Distribution of energy usage of the continents, grouped by type of energy source, assign individual colors (linetypes) to the factors taken from column 4

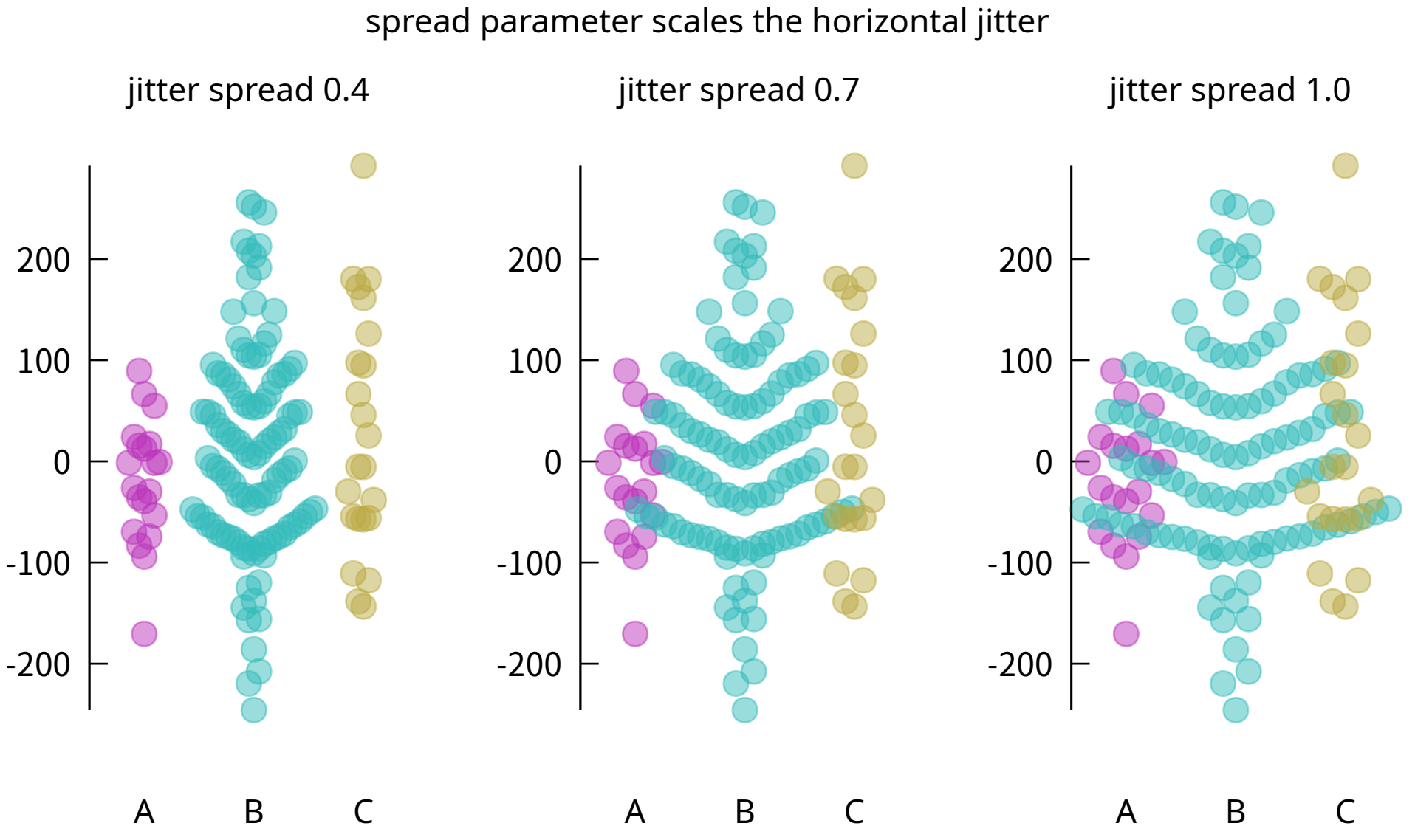



Distribution of energy usage of the continents, sorted by name of energy source
Billion Tons of Oil Equivalent
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Coal
Gas
Hydroelectric
Nuclear
Oil
Renewable

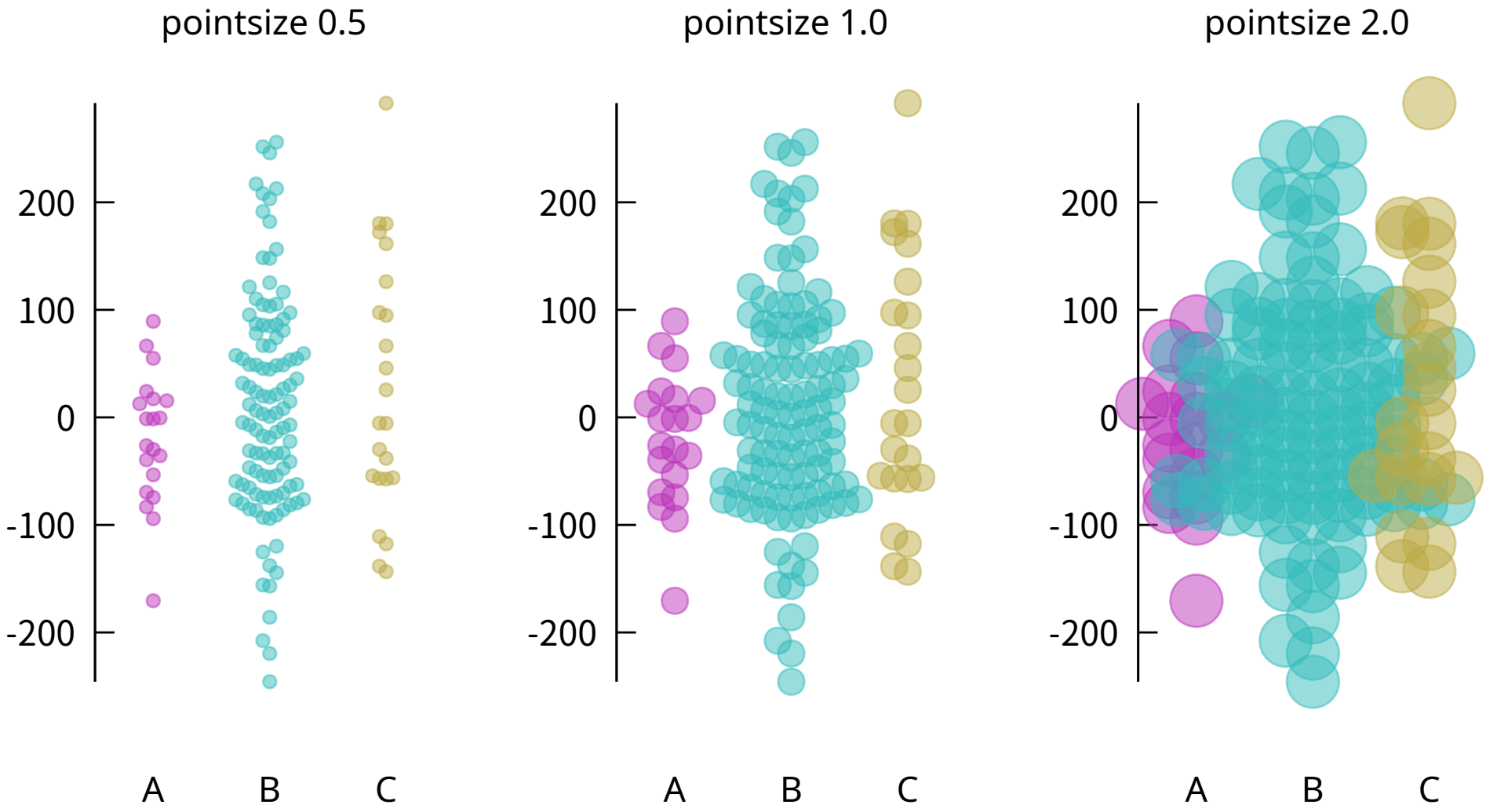

Distribution of energy usage explicitly ordered by name of energy source
Billion Tons of Oil Equivalent
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Nuclear
Coal
Gas
Oil
Hydroelectric
Renewable

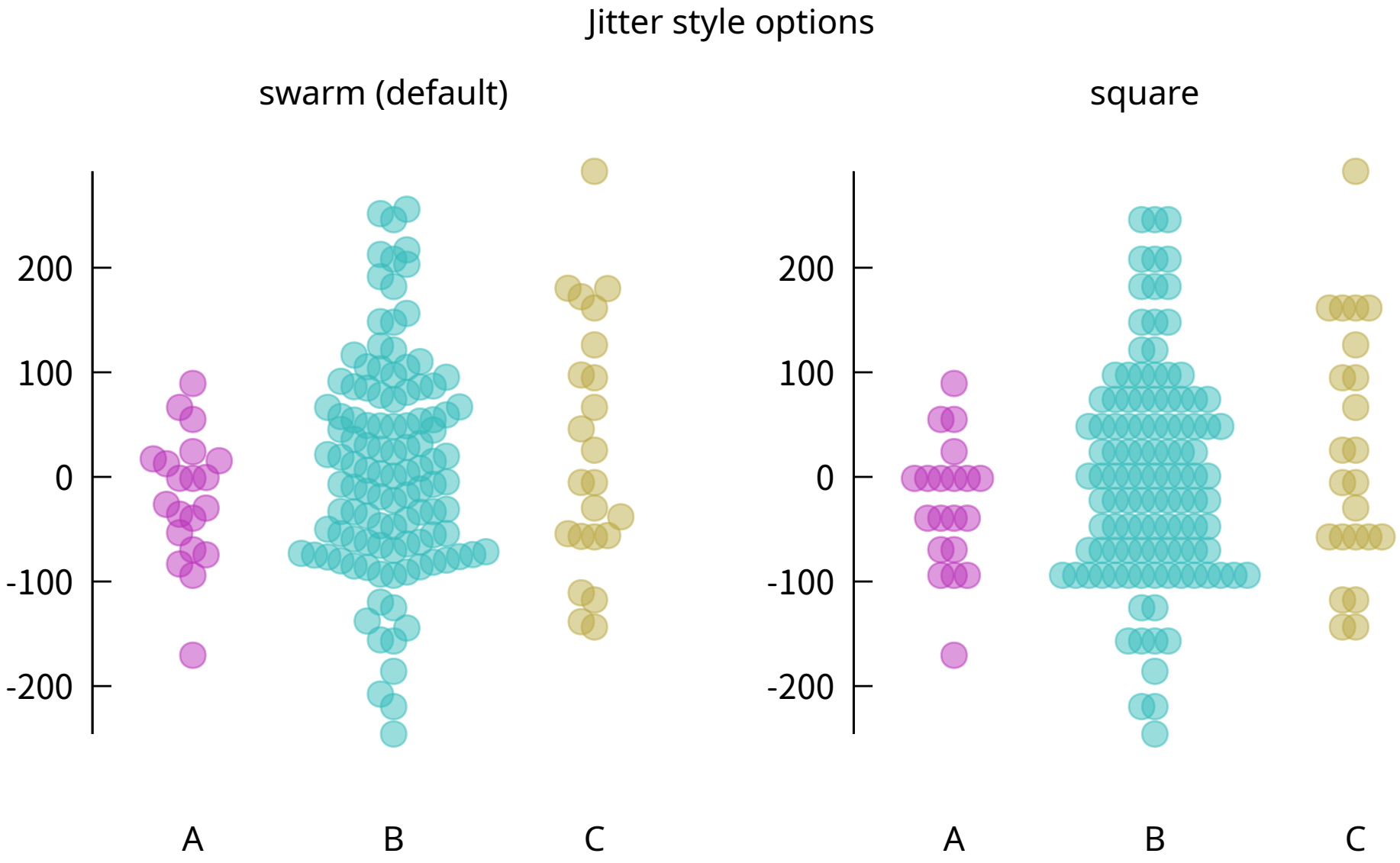

no jitter
jitter
200
100
0
-100
-200
A
B
C

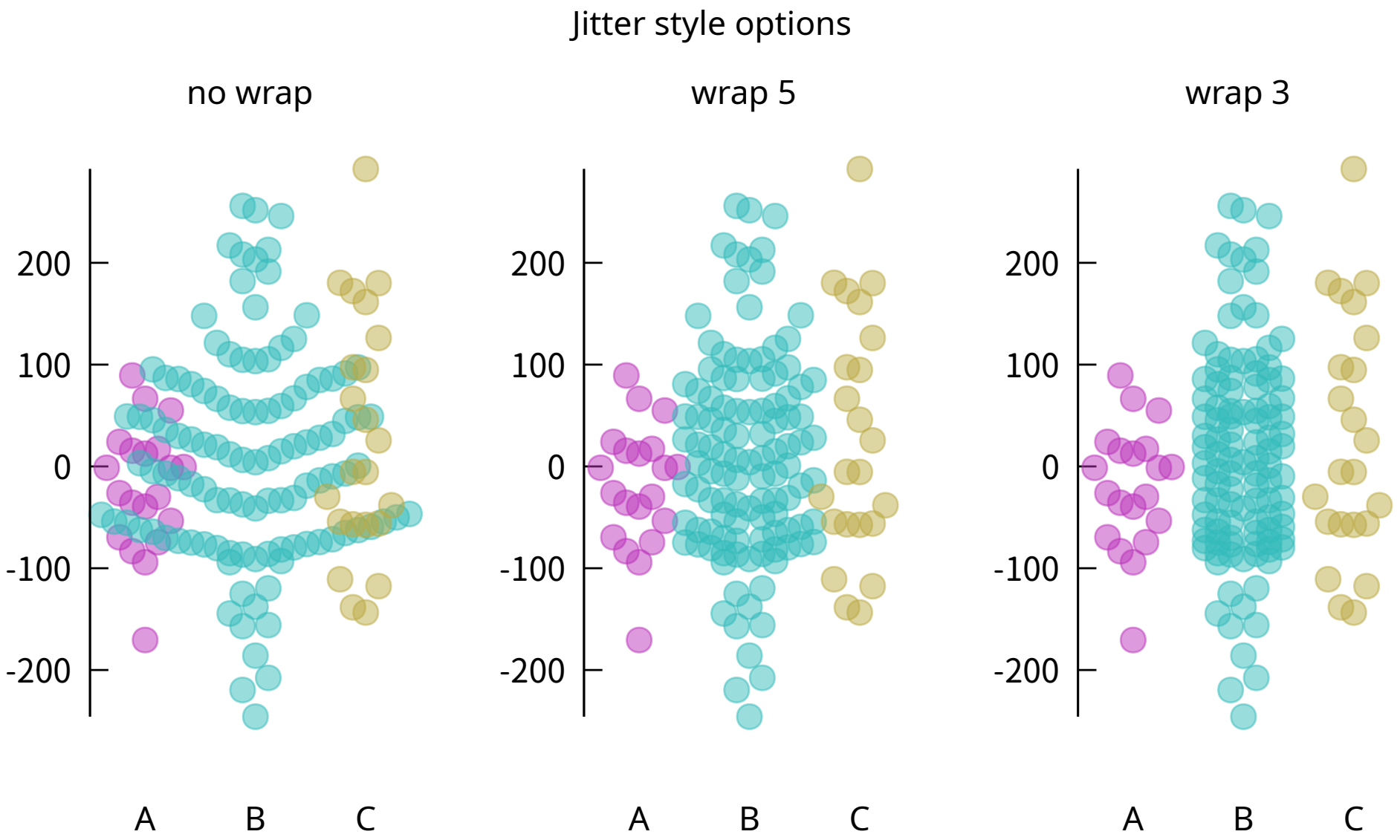

vertical overlap criterion
jitter overlap 0.5
jitter overlap 1.0
jitter overlap 1.5
200
100
0
-100
-200
A
B
C

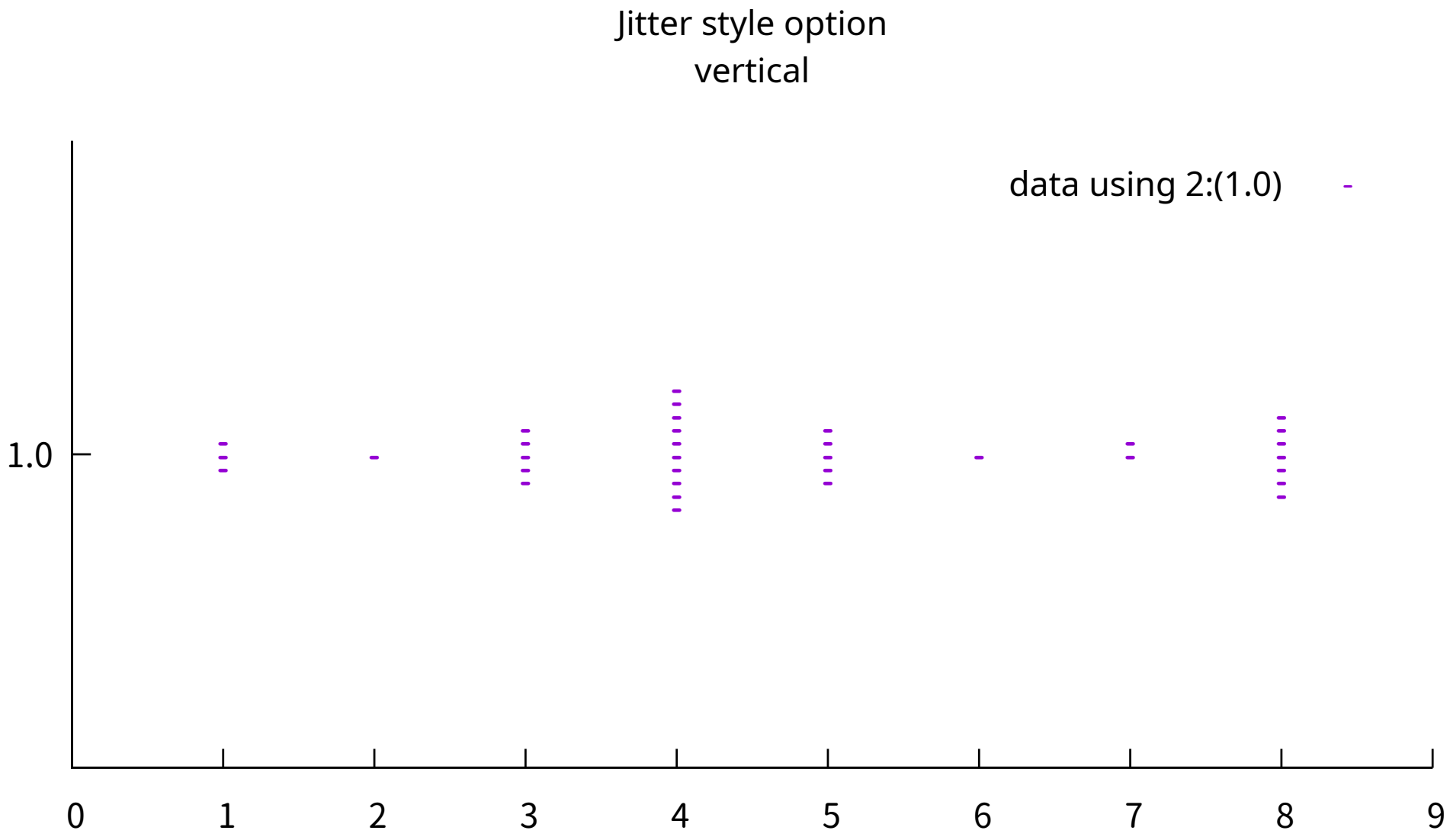

spread parameter scales the horizontal jitter
jitter spread 0.4
jitter spread 0.7
jitter spread 1.0
200
100
0
-100
-200
A
B
C

Plot appearance is also affected by point size

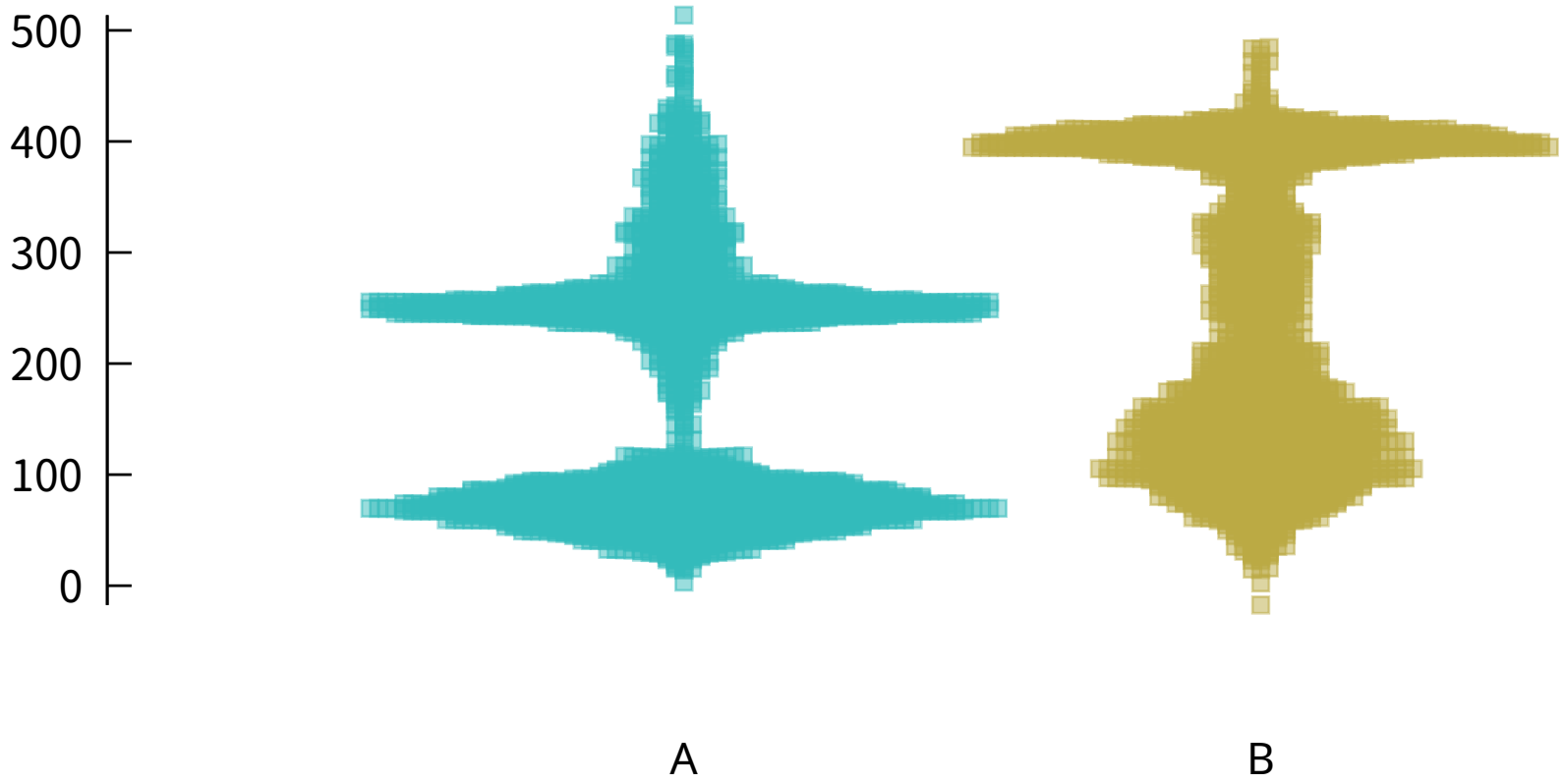


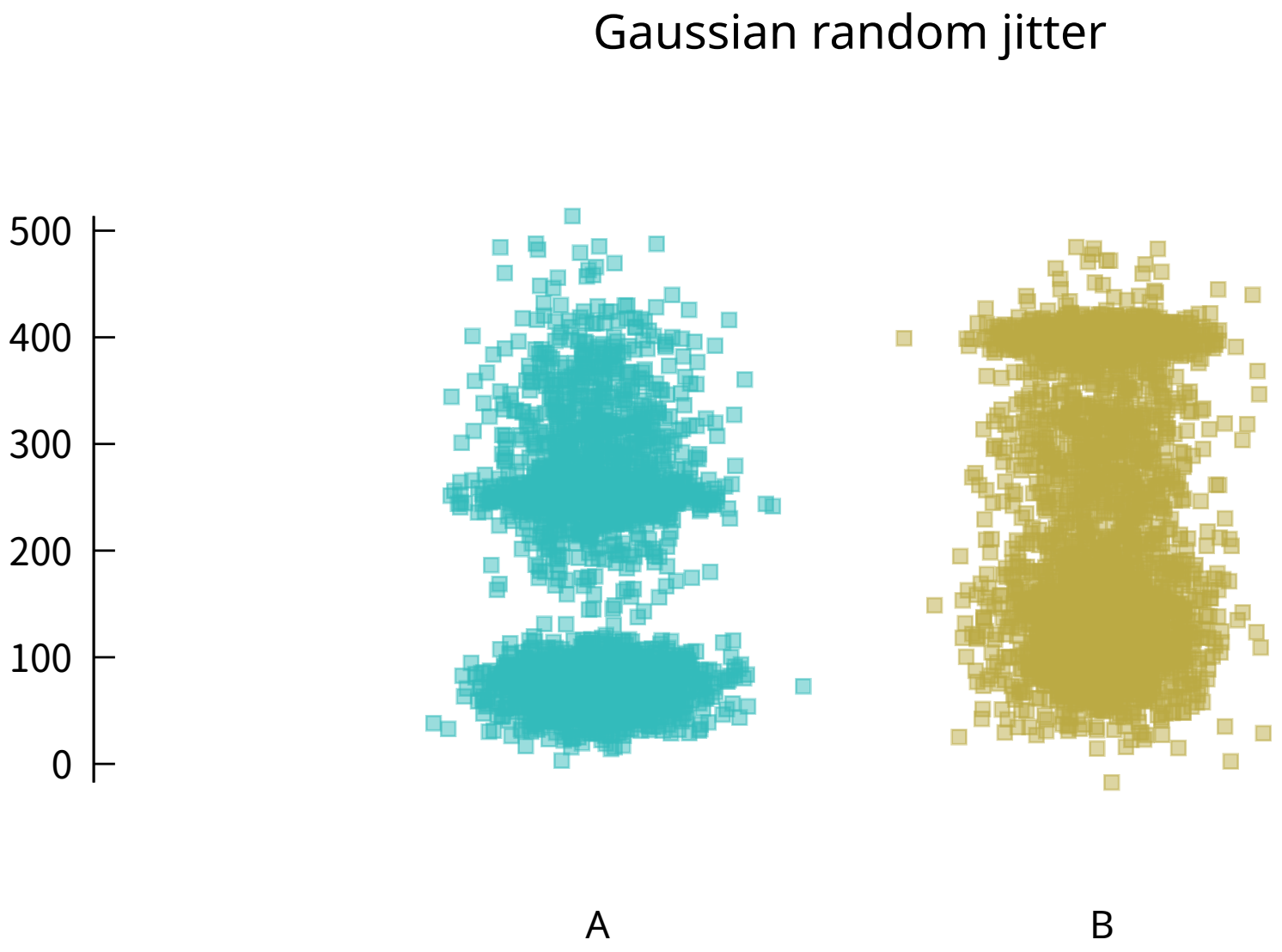

Jitter style options
swarm (default)
square
200
100
0
-100
-200
A
B
C

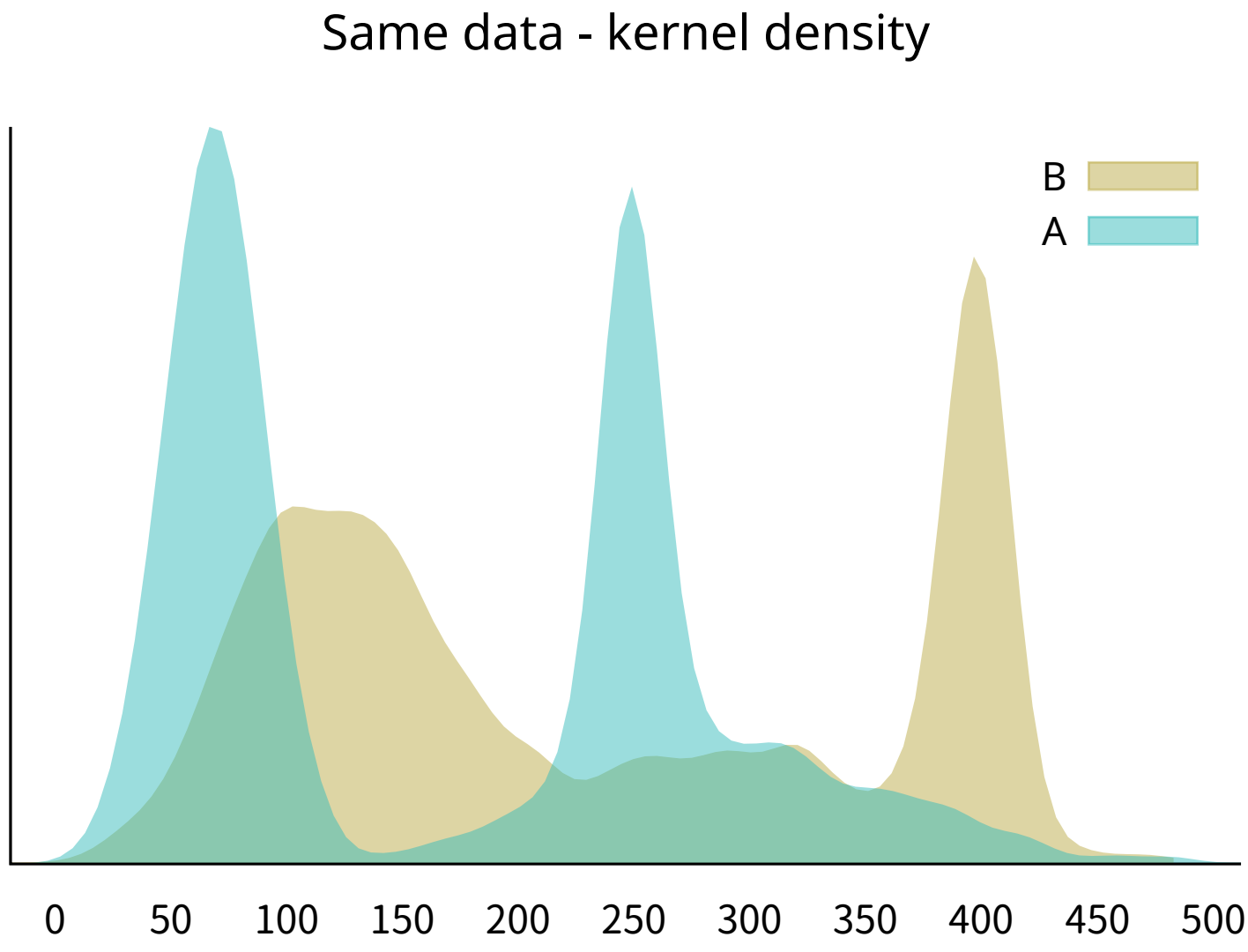

Jitter style options
no wrap
wrap 5
wrap 3
200
100
0
-100
-200
A
B
C


Jitter style option
vertical
data using 2:(1.0)
1.0
0
1
2
3
4
5
6
7
8
9

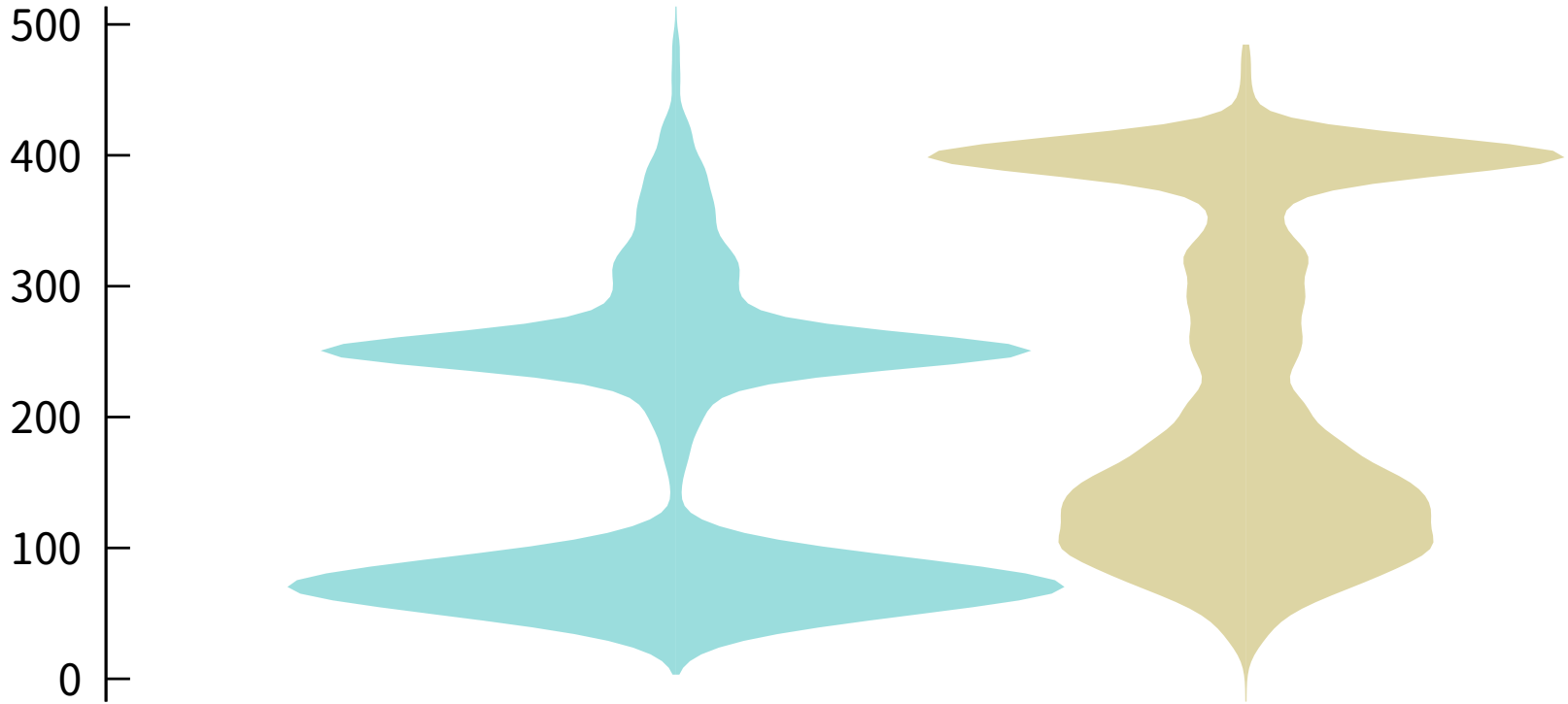
swarm jitter with a large number of points approximates a violin plot



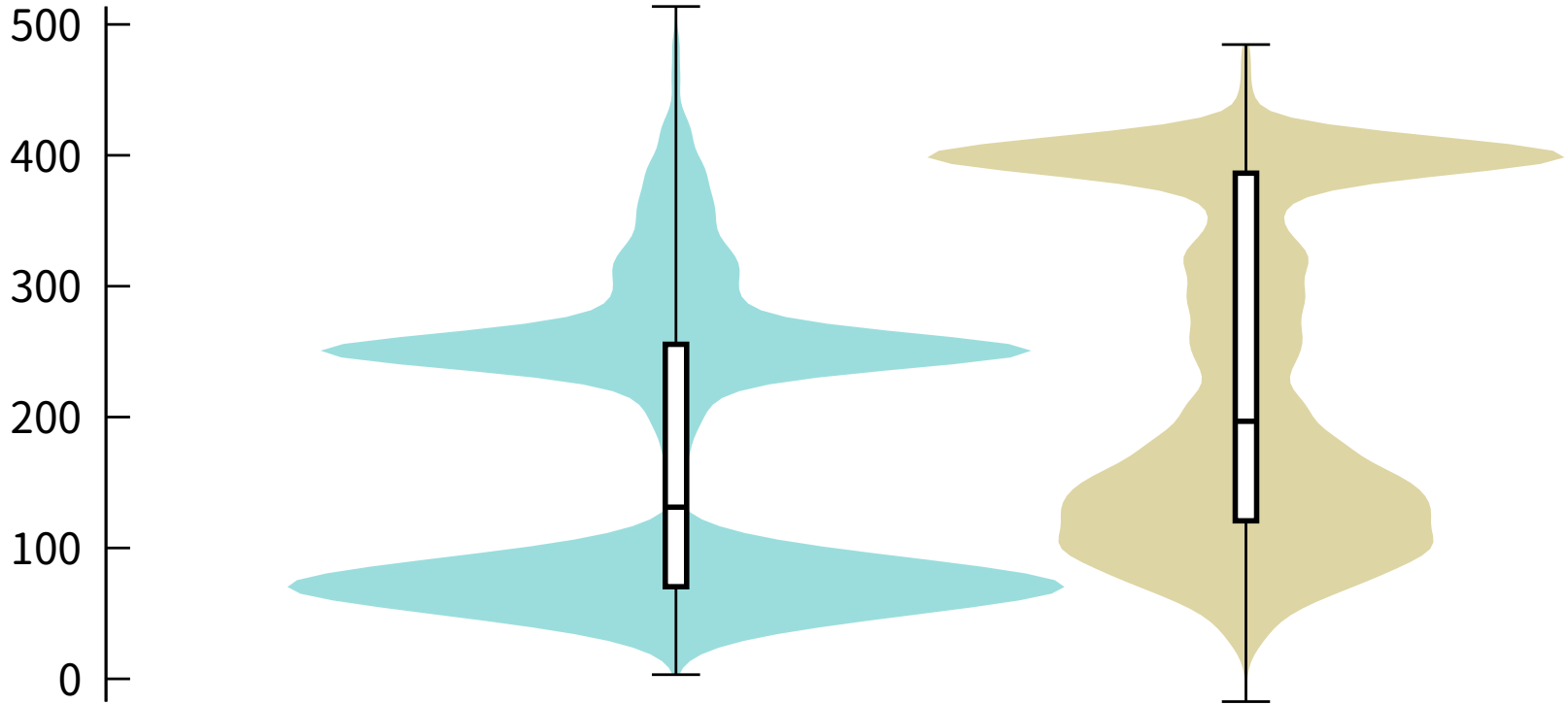

Gaussian random jitter
500
400
300
200
100
0
A
B


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

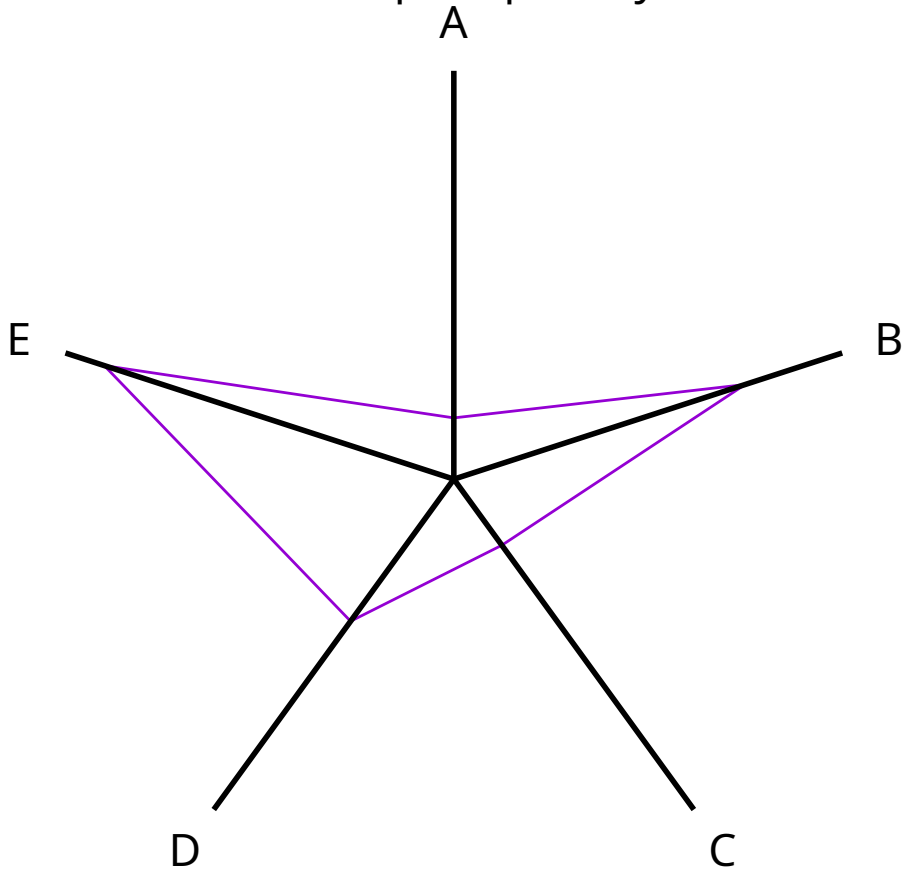
kdensity mirrored sideways to give a violin plot



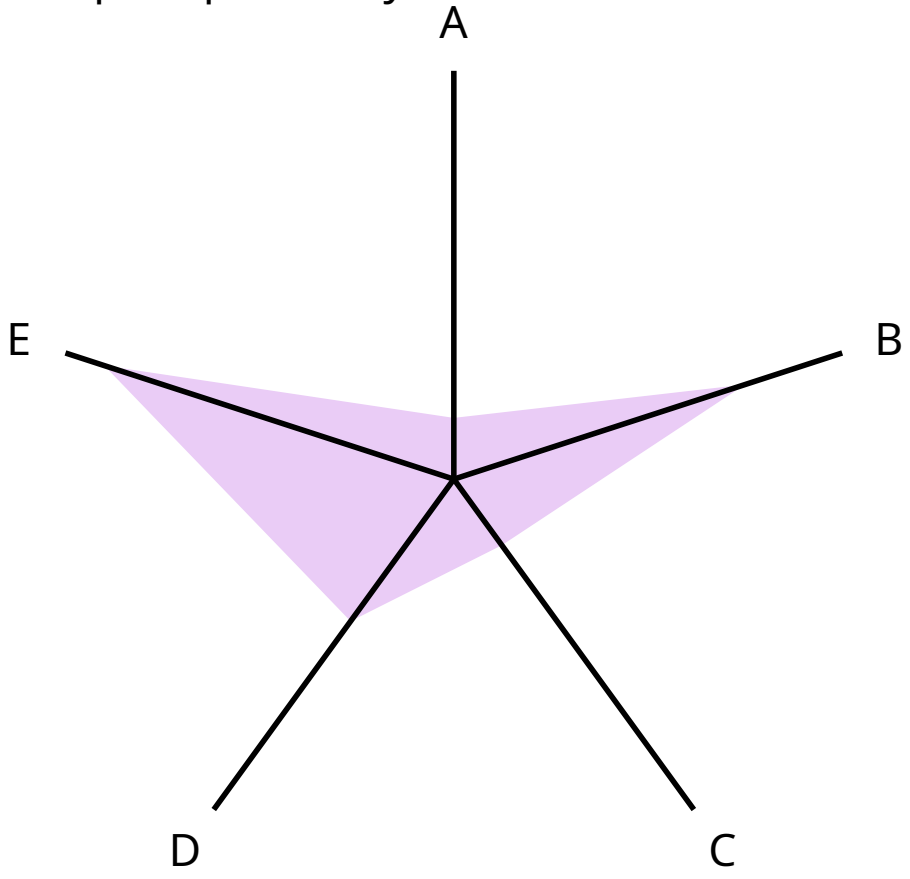
Superimposed violin plot and box plot



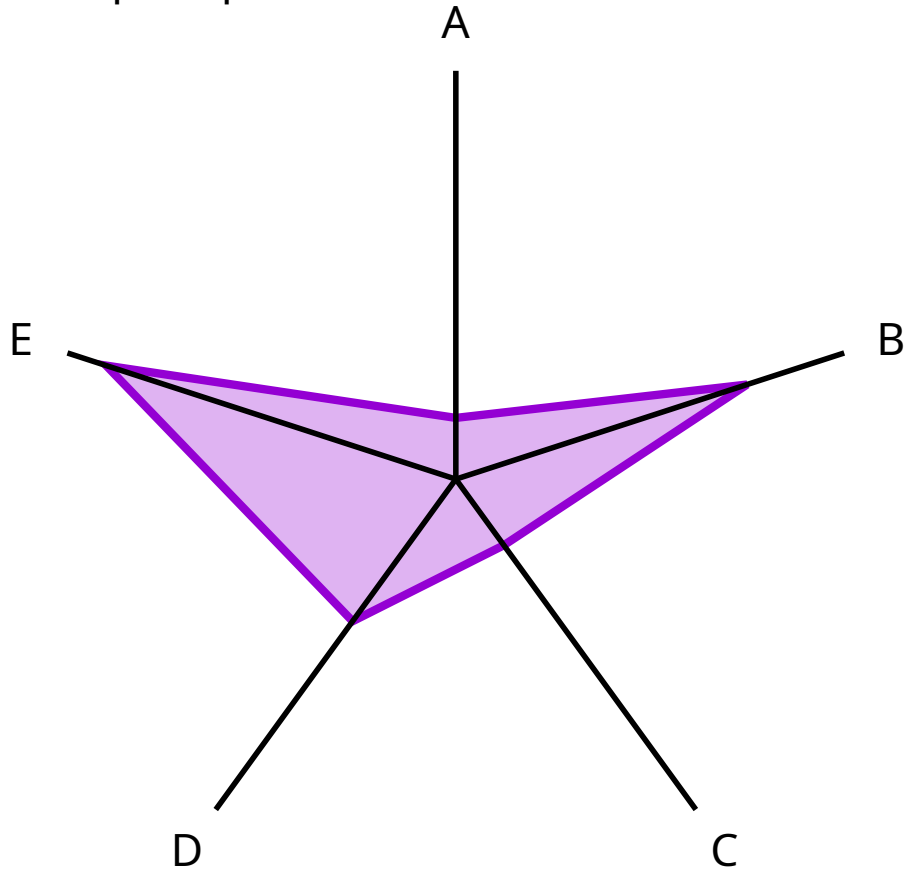
default spiderplot style



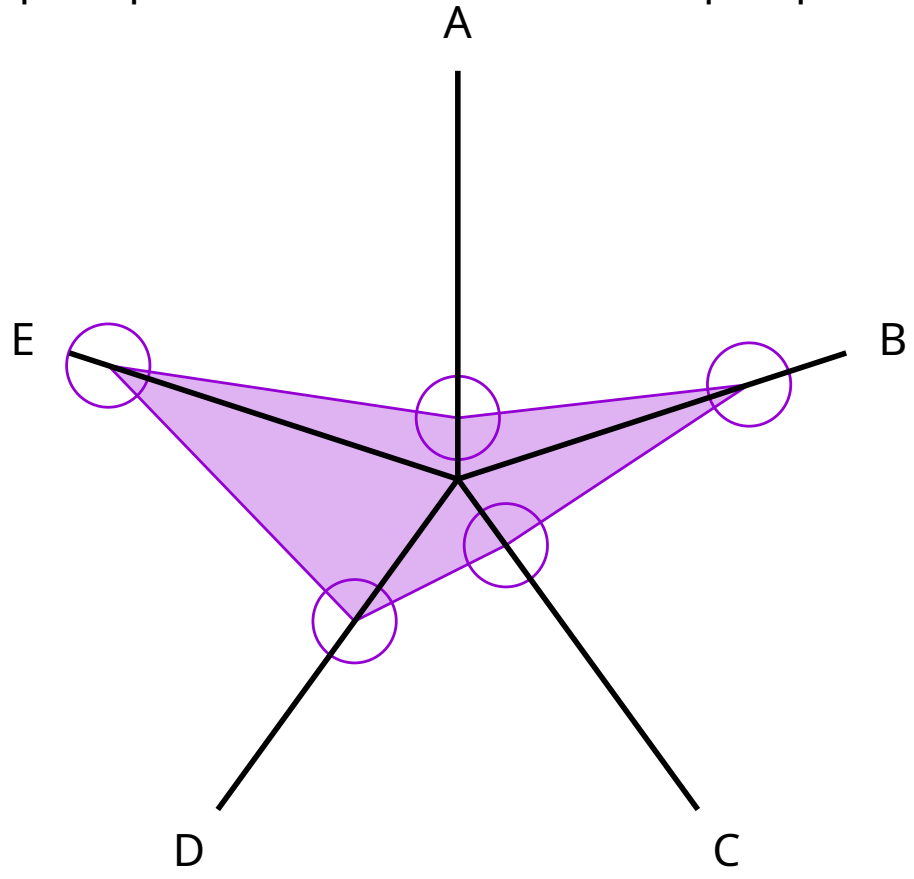
spiderplot fillstyle solid 0.2 noborder

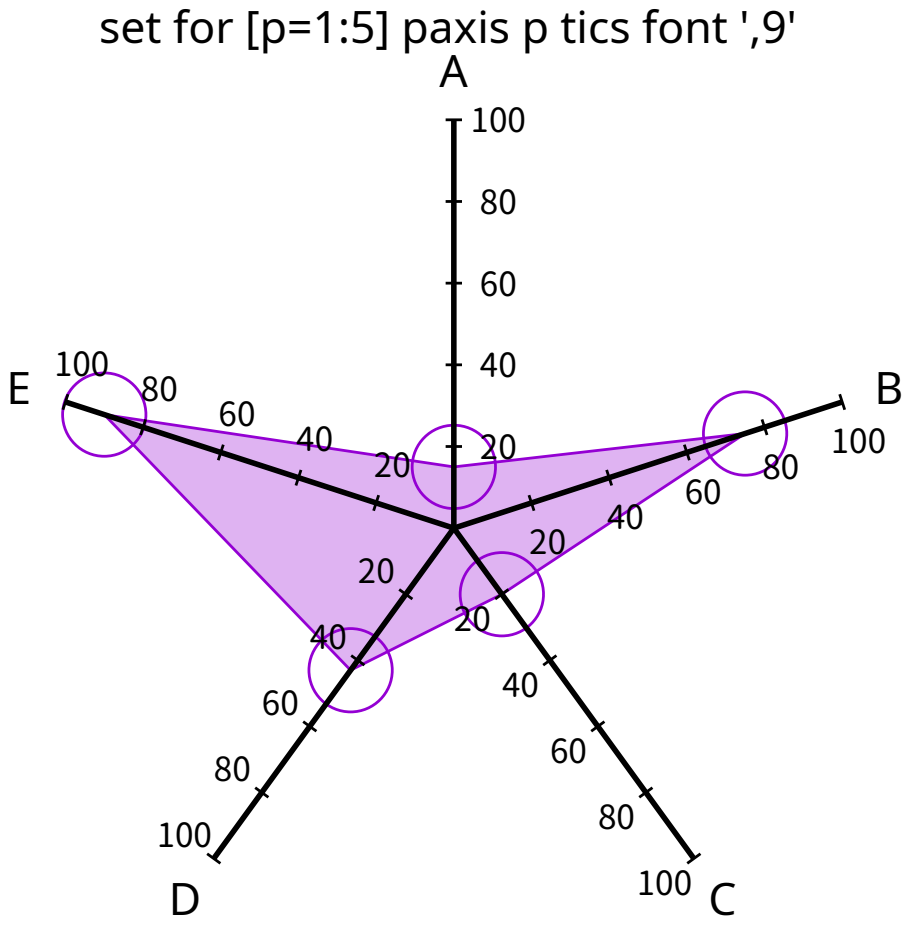


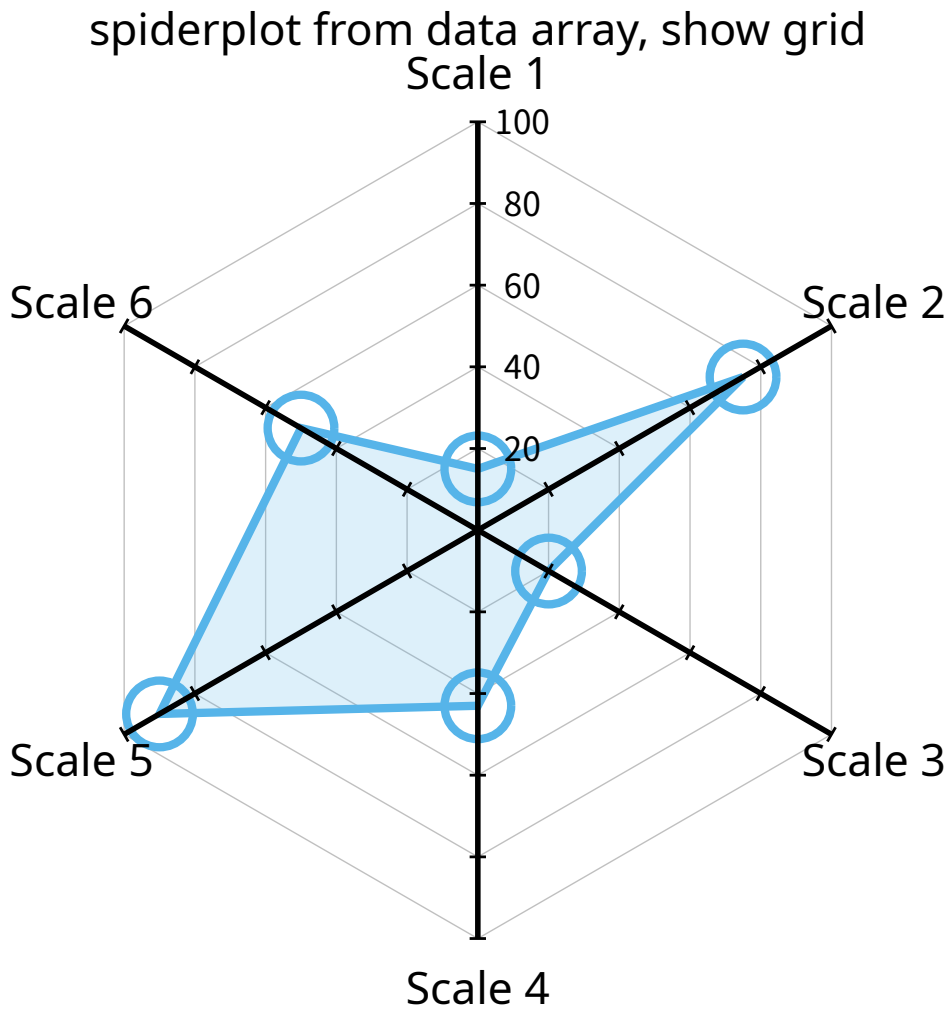
spiderplot fs solid 0.3 border lw 3.0



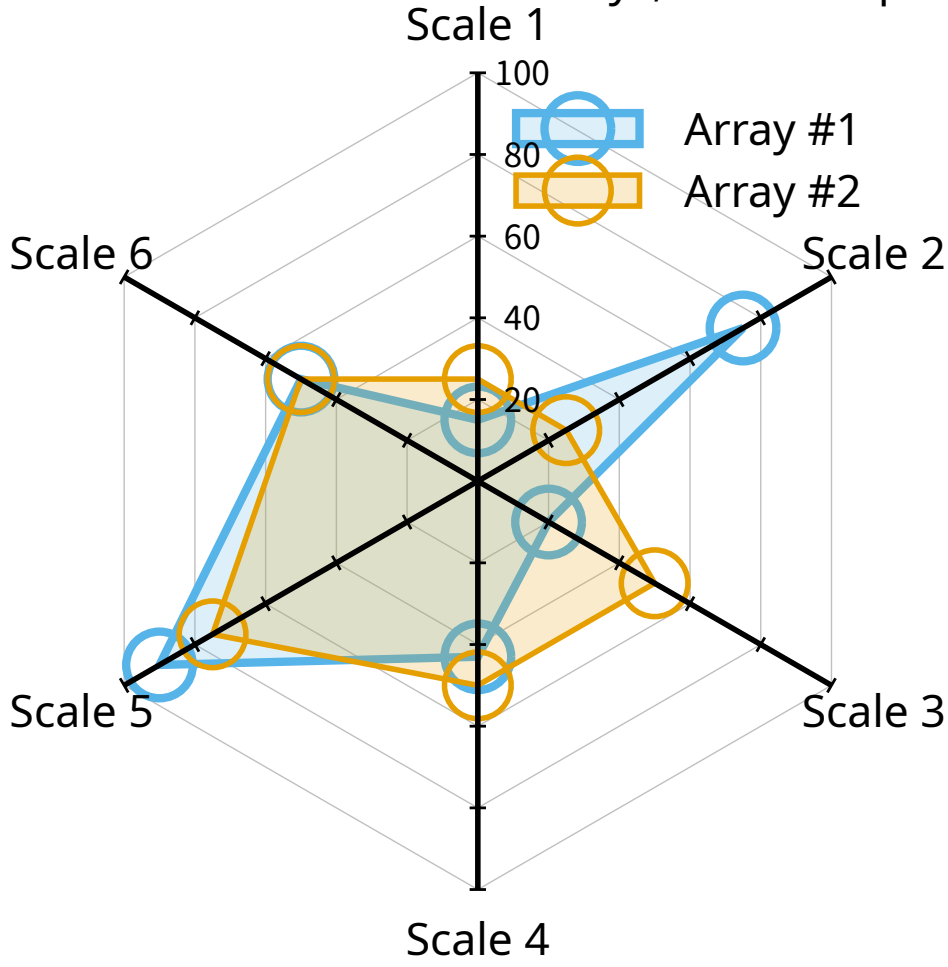
spiderplot fs solid 0.3 border lw 1 pt 6 ps 2.5

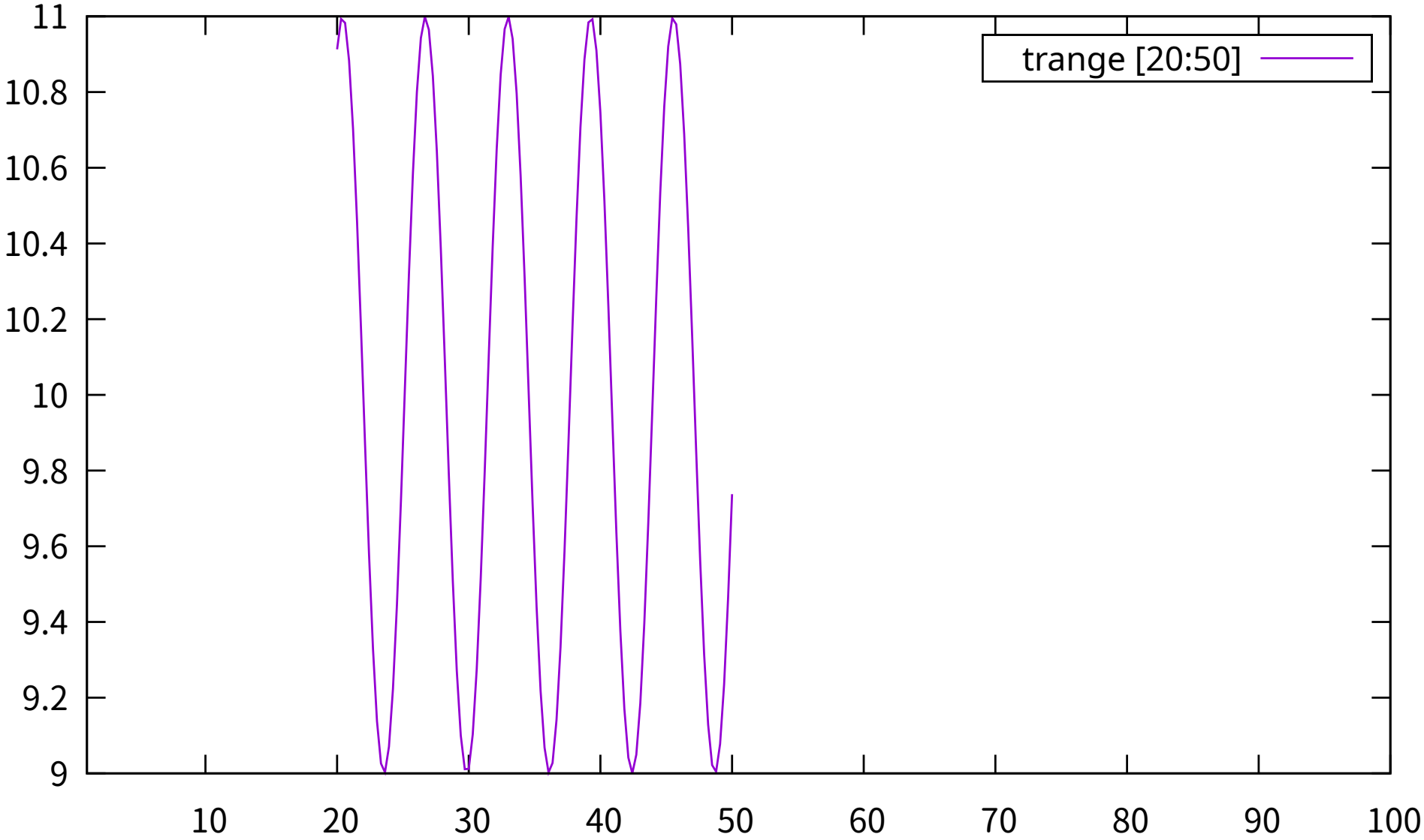


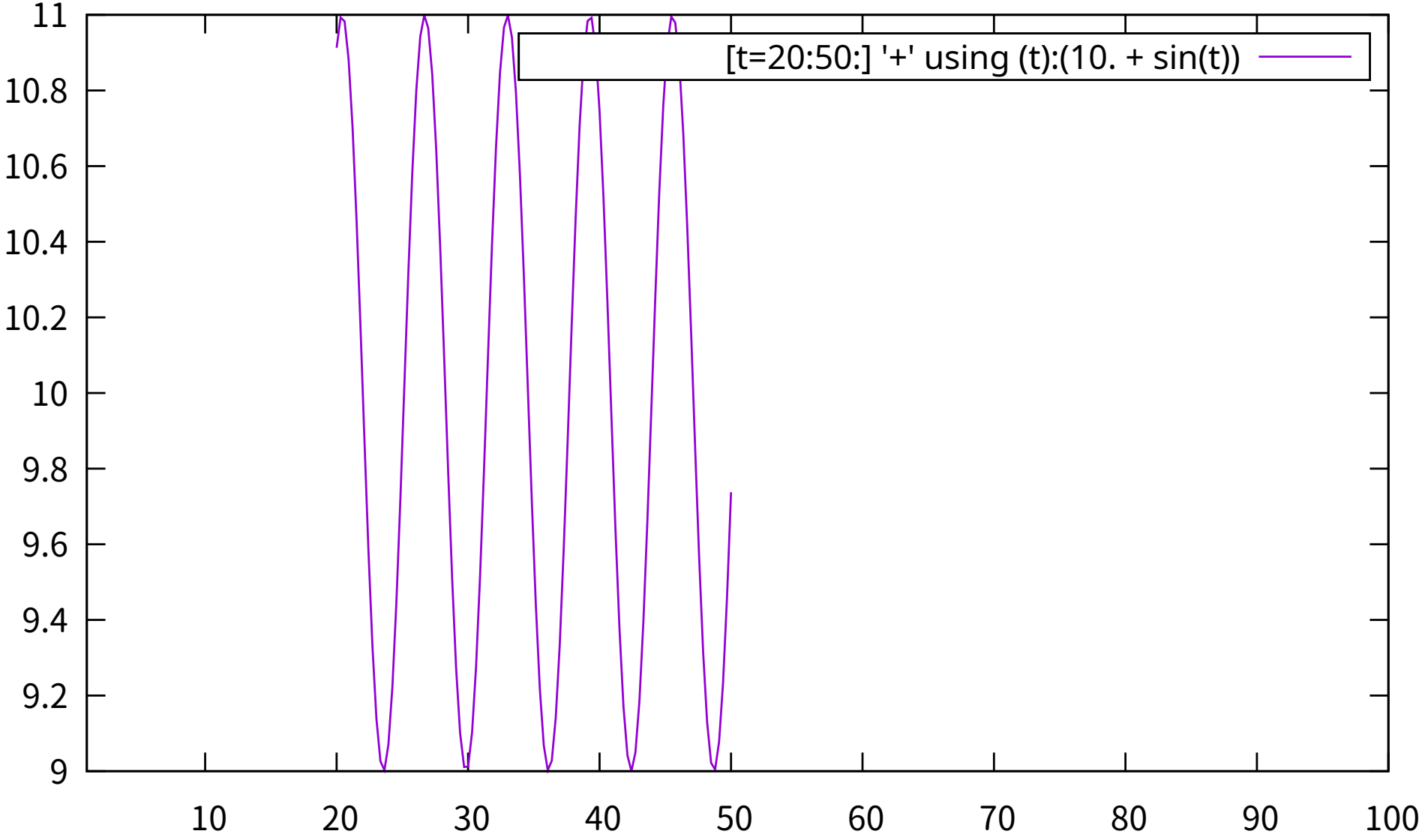

set for [p=1:5] paxis p tics font ',9'
A
B
C
D
E
100
80
60
40
20

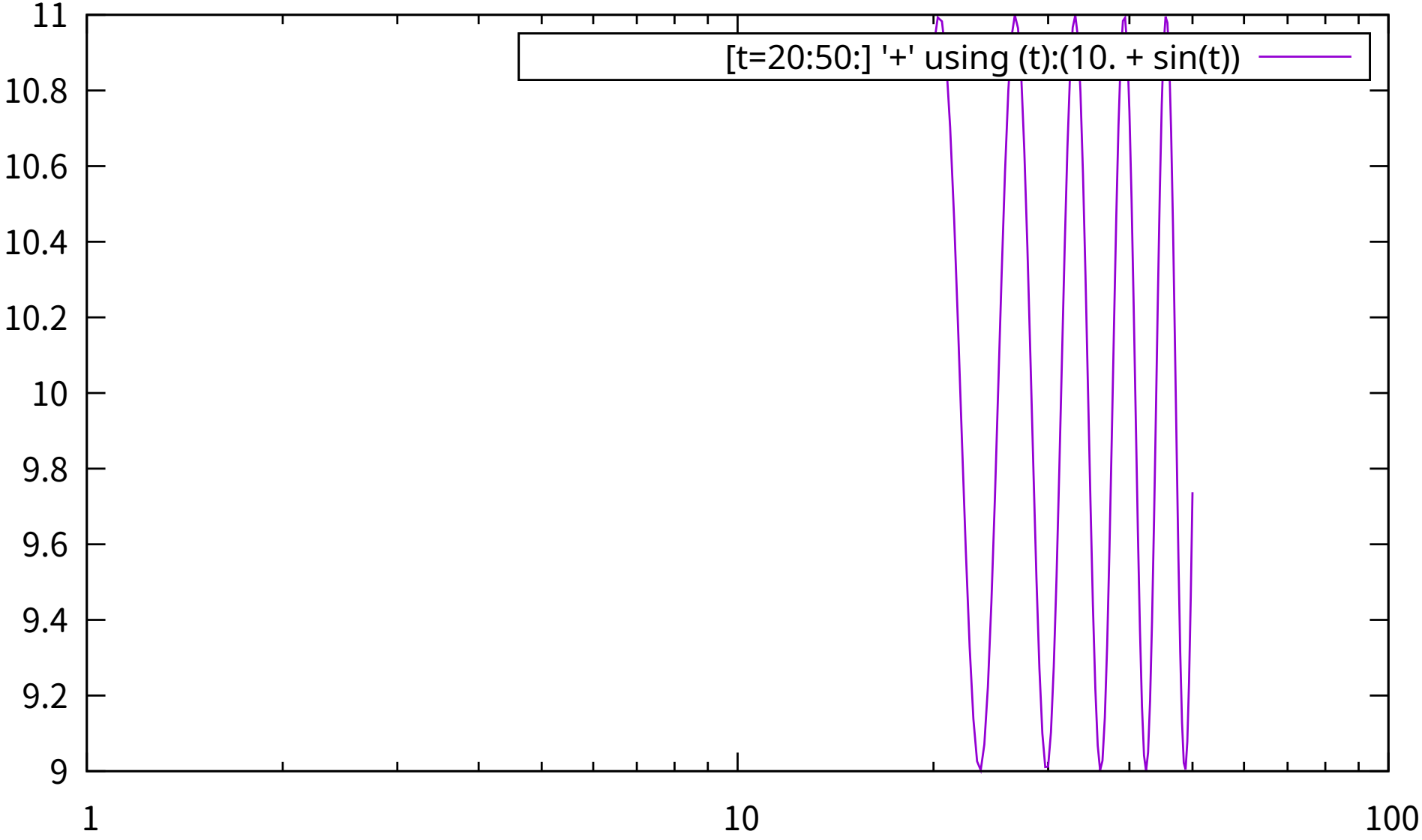

spiderplot from data array, show grid
Scale 1
100
80
60
40
20
Scale 2
Scale 3
Scale 4
Scale 5
Scale 6

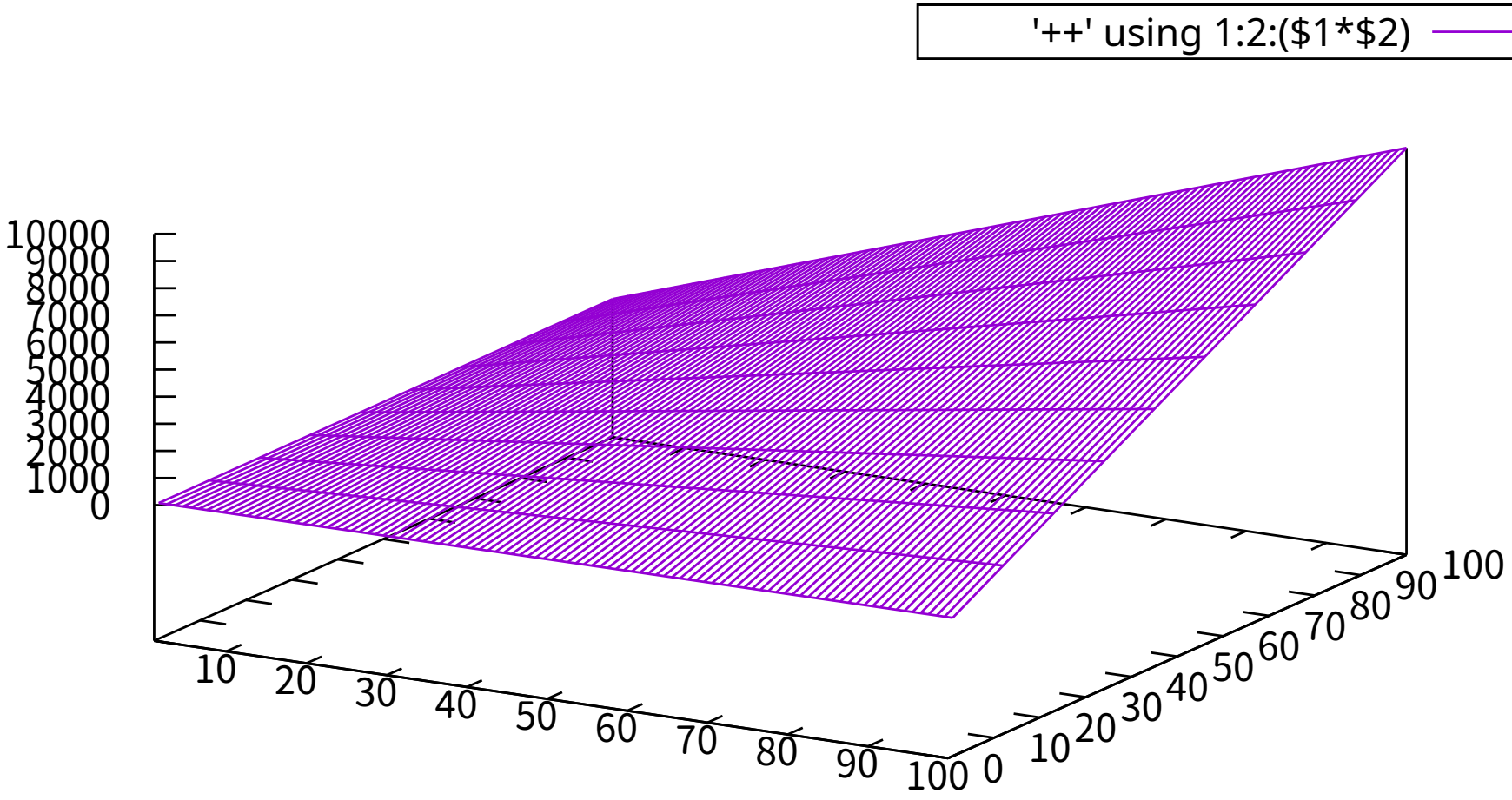
To plot from 2 different files or arrays, use 'newspiderplot'

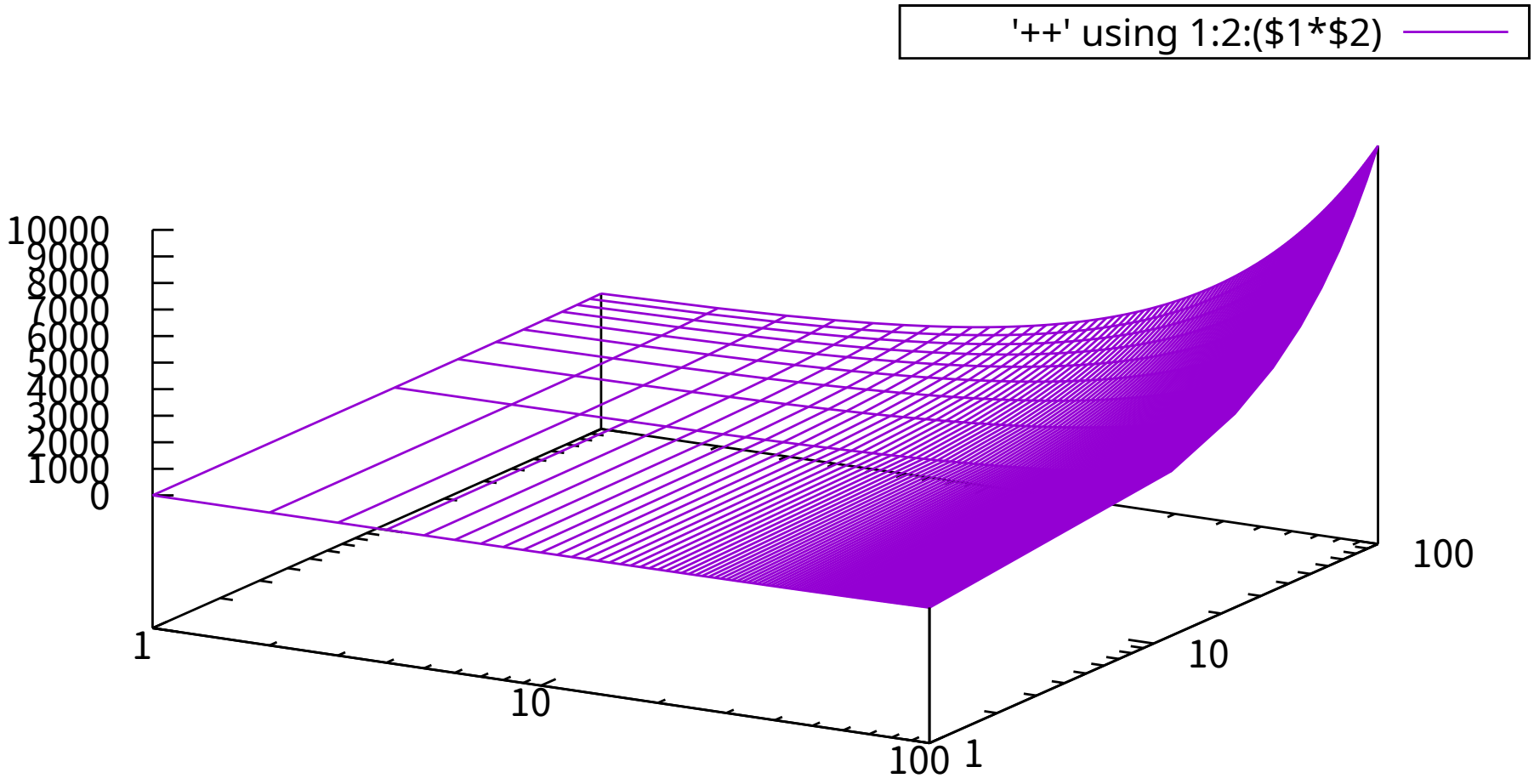


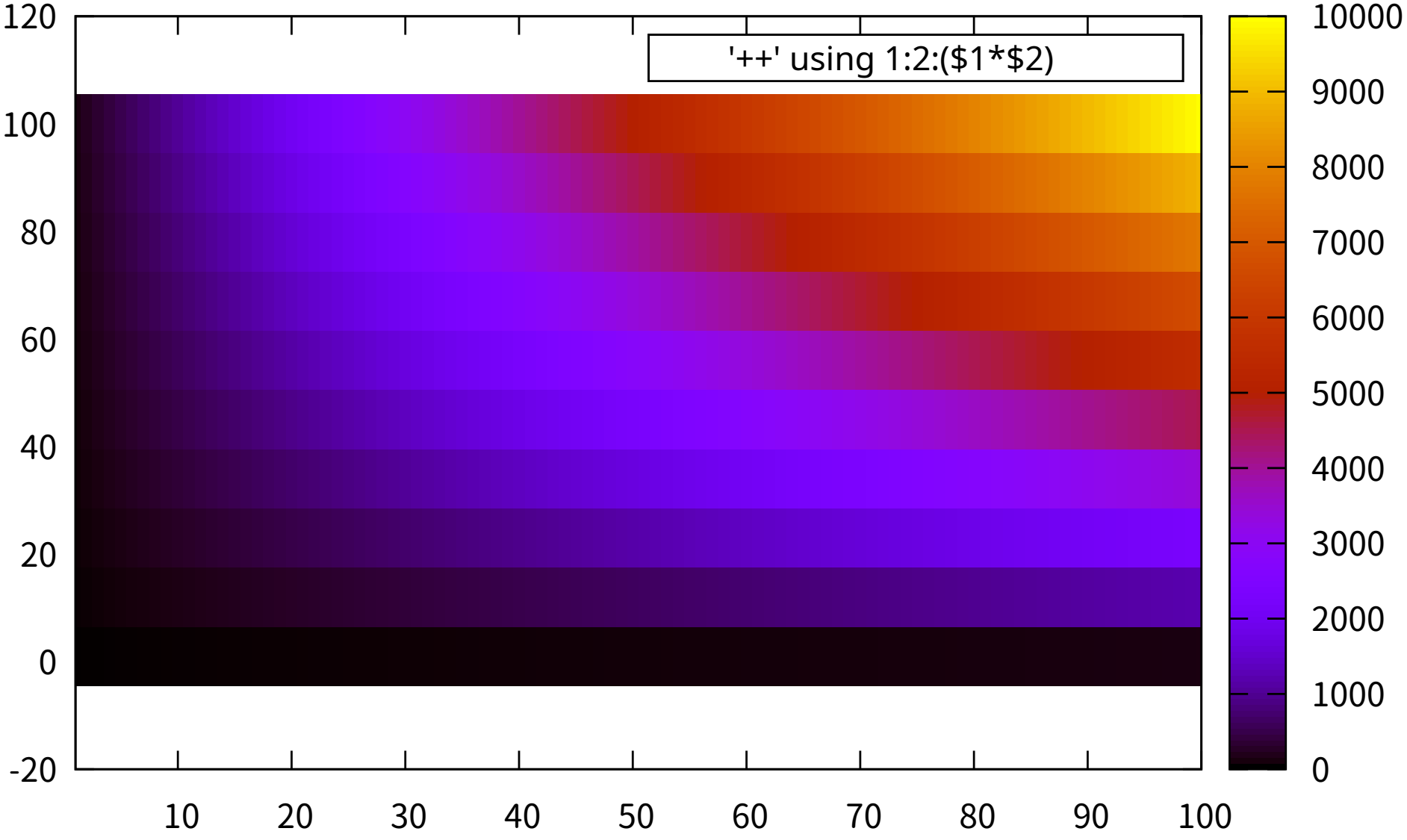

trange [20:50]
11
10.8
10.6
10.4
10.2
10
9.8
9.6
9.4
9.2
9
10
20
30
40
50
60
70
80
90
100

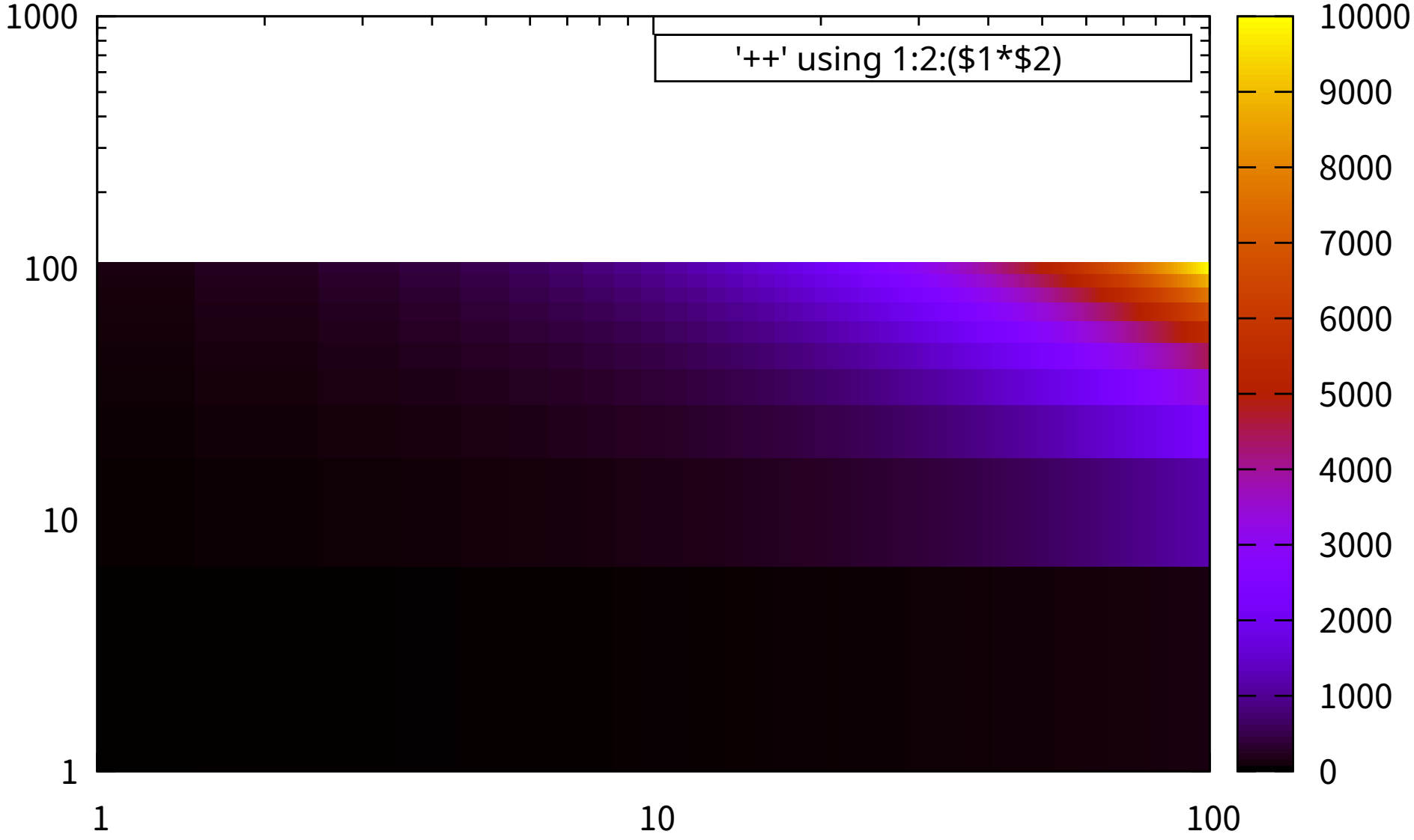

[t=20:50:] '+' using (t):(10. + sin(t))
11
10.8
10.6
10.4
10.2
10
9.8
9.6
9.4
9.2
9
10
20
30
40
50
60
70
80
90
100

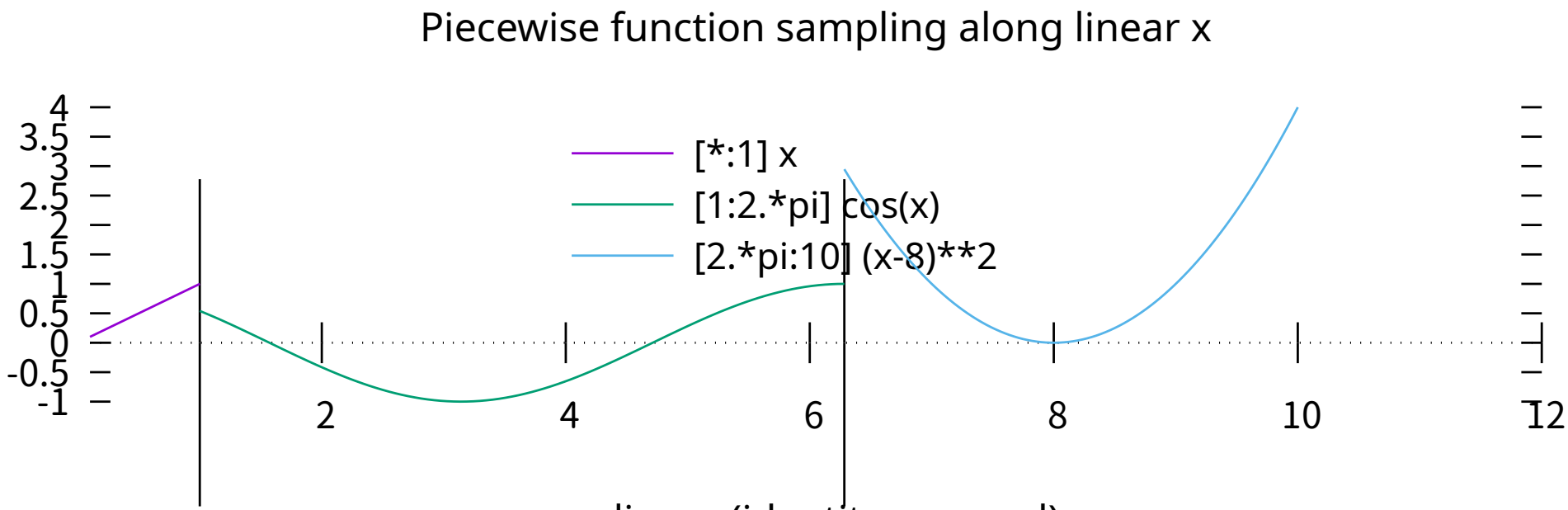

[t=20:50:] '+' using (t):(10. + sin(t))
11
10.8
10.6
10.4
10.2
10
9.8
9.6
9.4
9.2
9
1
10
100

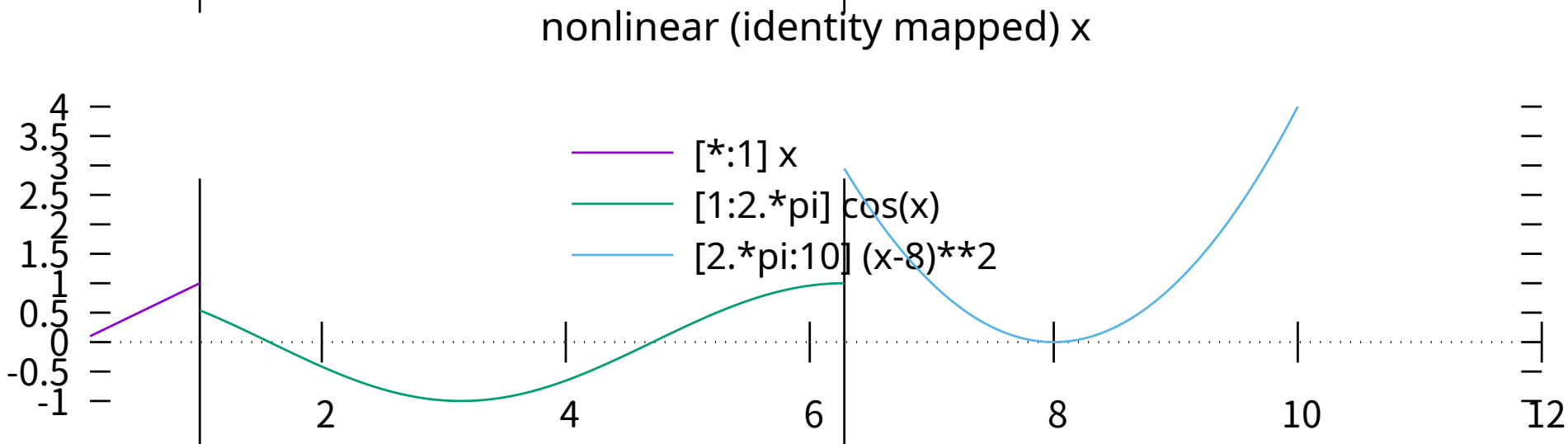

'++' using 1:2:($1*$2)
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
10
20
30
40
50
60
70
80
90
100
0
10
20
30
40
50
60
70
80
90
100

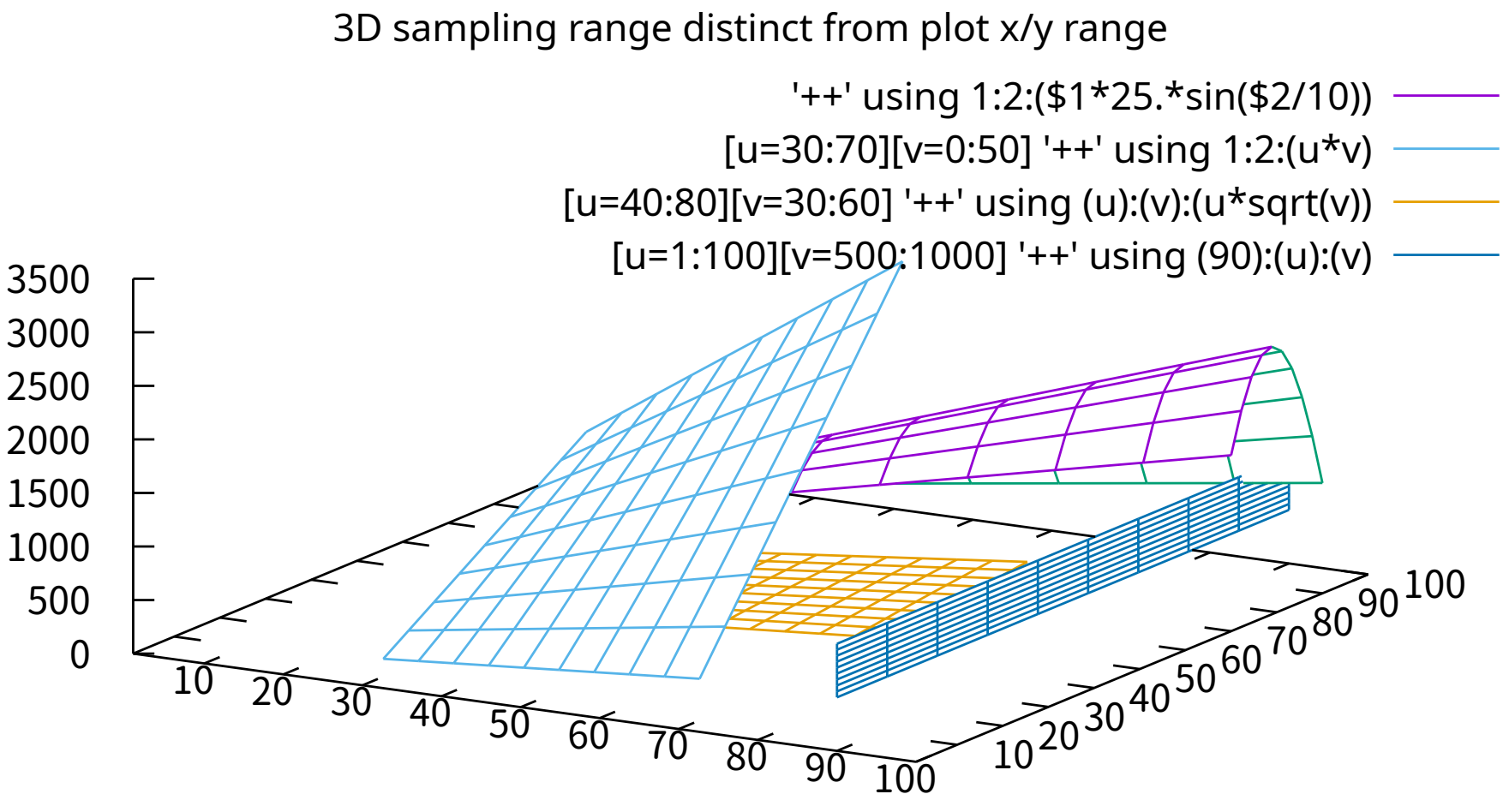

'++' using 1:2:($1*$2)
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
1
10
100
1
10
100


'++' using 1:2:($1*$2)
120
100
80
60
40
20
0
-20
10
20
30
40
50
60
70
80
90
100
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0

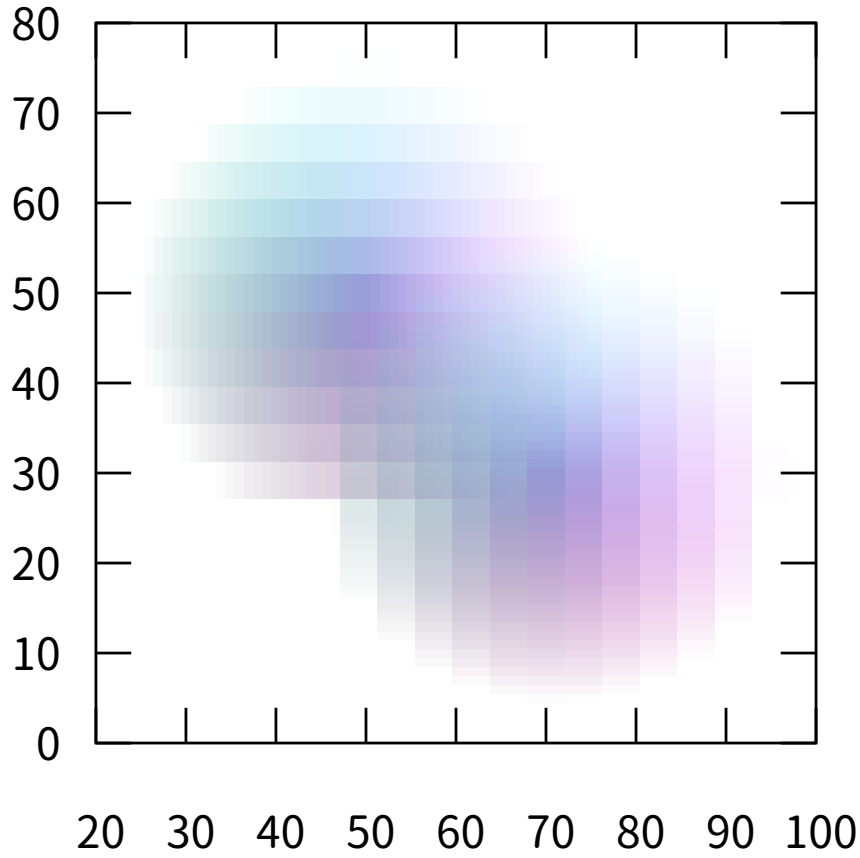

'++' using 1:2:($1*$2)
1000
100
10
1
1
10
100
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0

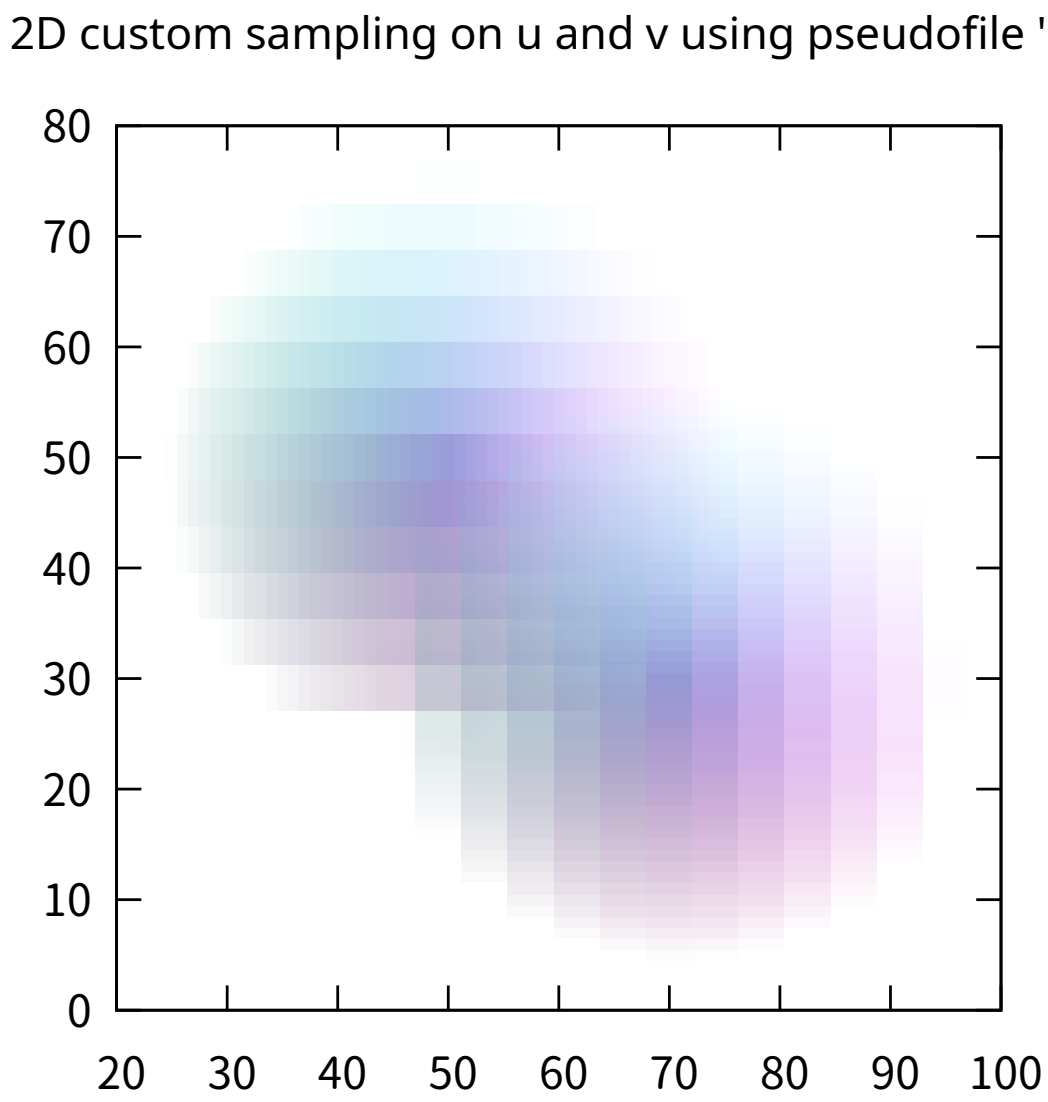

Piecewise function sampling along linear x
[*:1] x
[1:2.*pi] cos(x)
[2.*pi:10] (x-8)**2
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
2
4
6
8
10
12

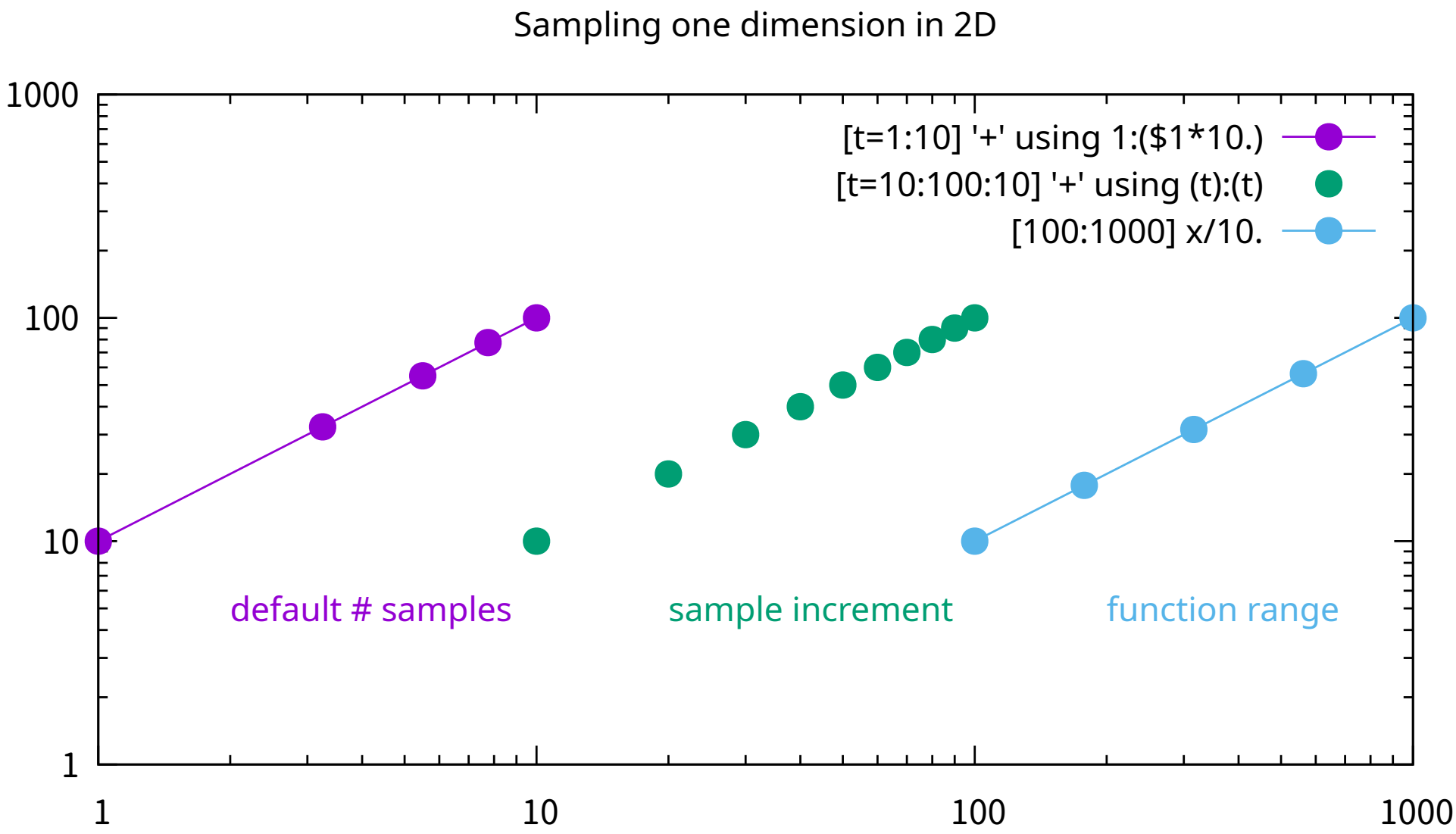

nonlinear (identity mapped) x
[*:1] x
[1:2.*pi] cos(x)
[2.*pi:10] (x-8)**2
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
2
4
6
8
10
12

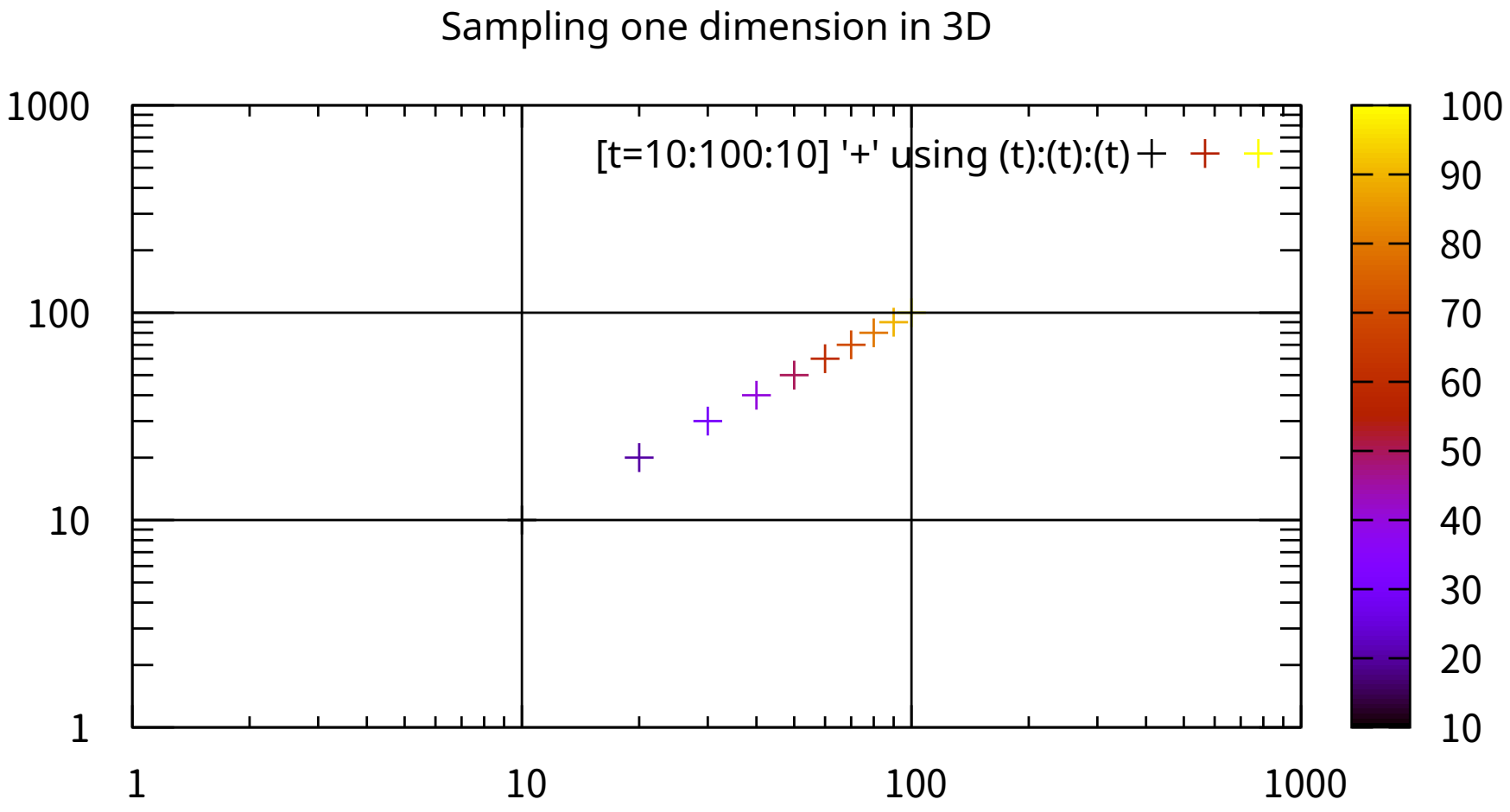

3D sampling range distinct from plot x/y range
'++' using 1:2:($1*25.*sin($2/10))
[u=30:70][v=0:50] '++' using 1:2:(u*v)
[u=40:80][v=30:60] '++' using (u):(v):(u*sqrt(v))
[u=1:100][v=500:1000] '++' using (90):(u):(v)
3500
3000
2500
2000
1500
1000
500
0
10
20
30
40
50
60
70
80
90
100
10
20
30
40
50
60
70
80
90
100

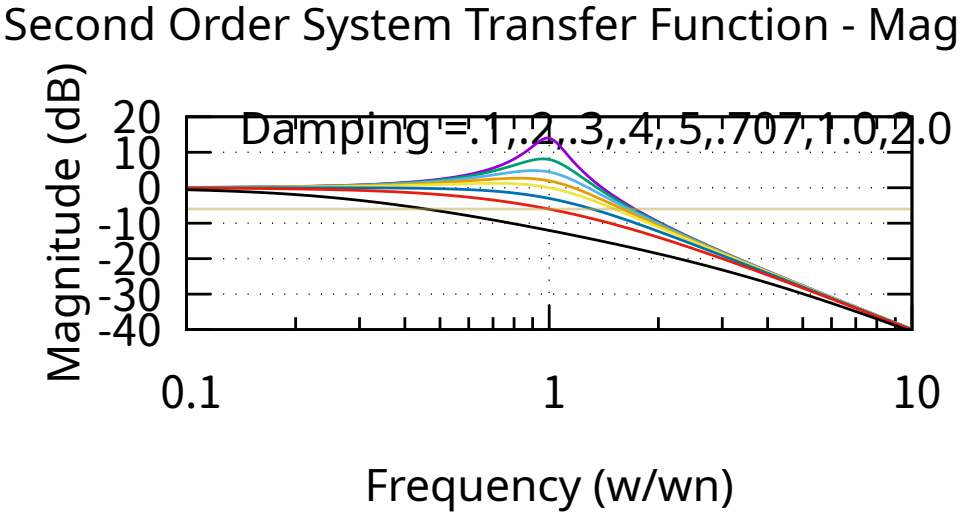
3D custom sampling on u and v using pseudofile '++'

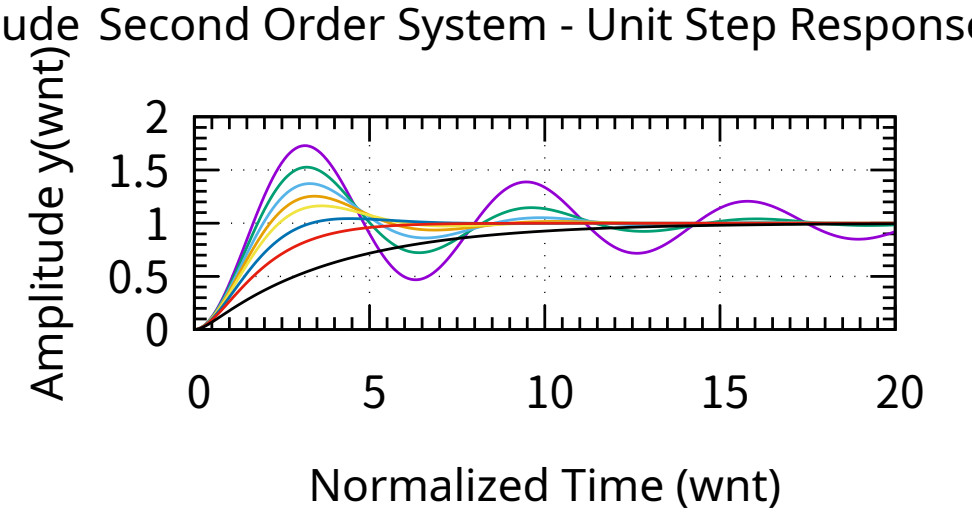


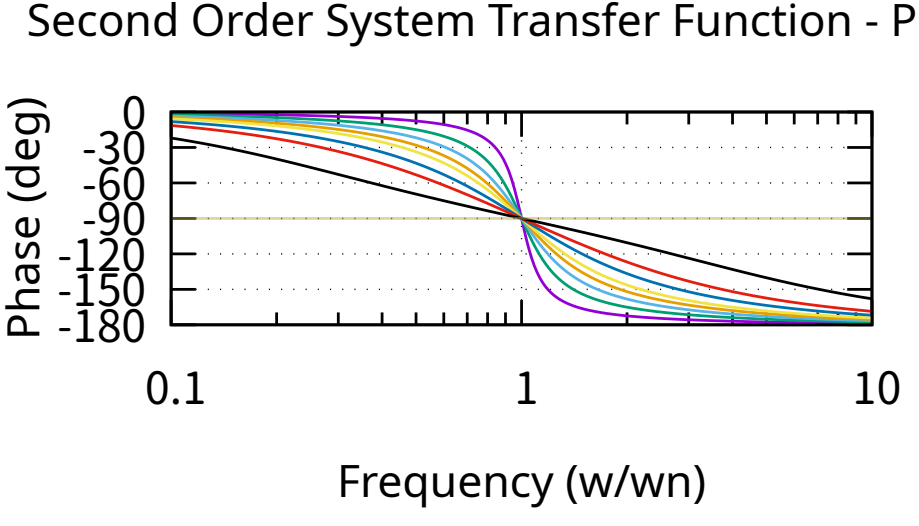

2D custom sampling on u and v using pseudofile '++'
80
70
60
50
40
30
20
10
0
20
30
40
50
60
70
80
90
100

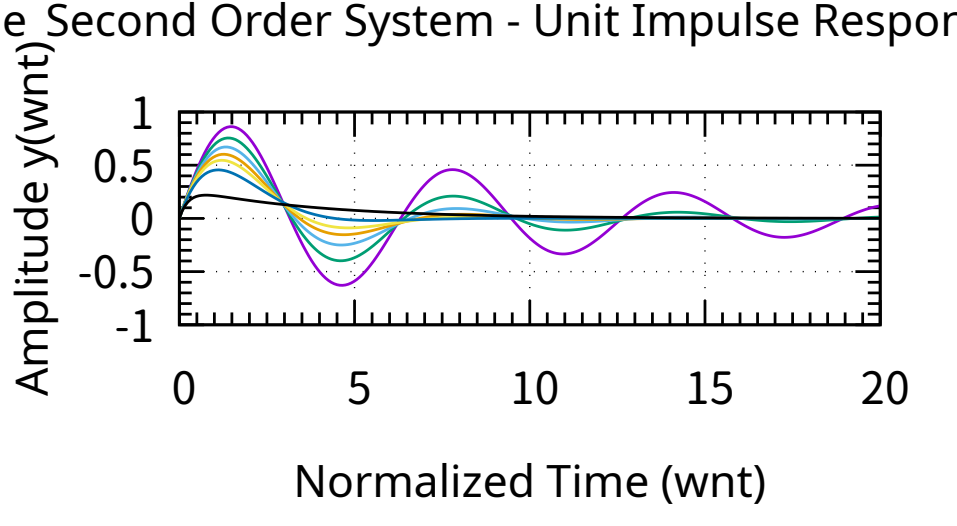

Sampling one dimension in 2D
[t=1:10] '+' using 1:($1*10.)
[t=10:100:10] '+' using (t):(t)
[100:1000] x/10.
default # samples
sample increment
function range
1000
100
10
1
1
10
100
1000

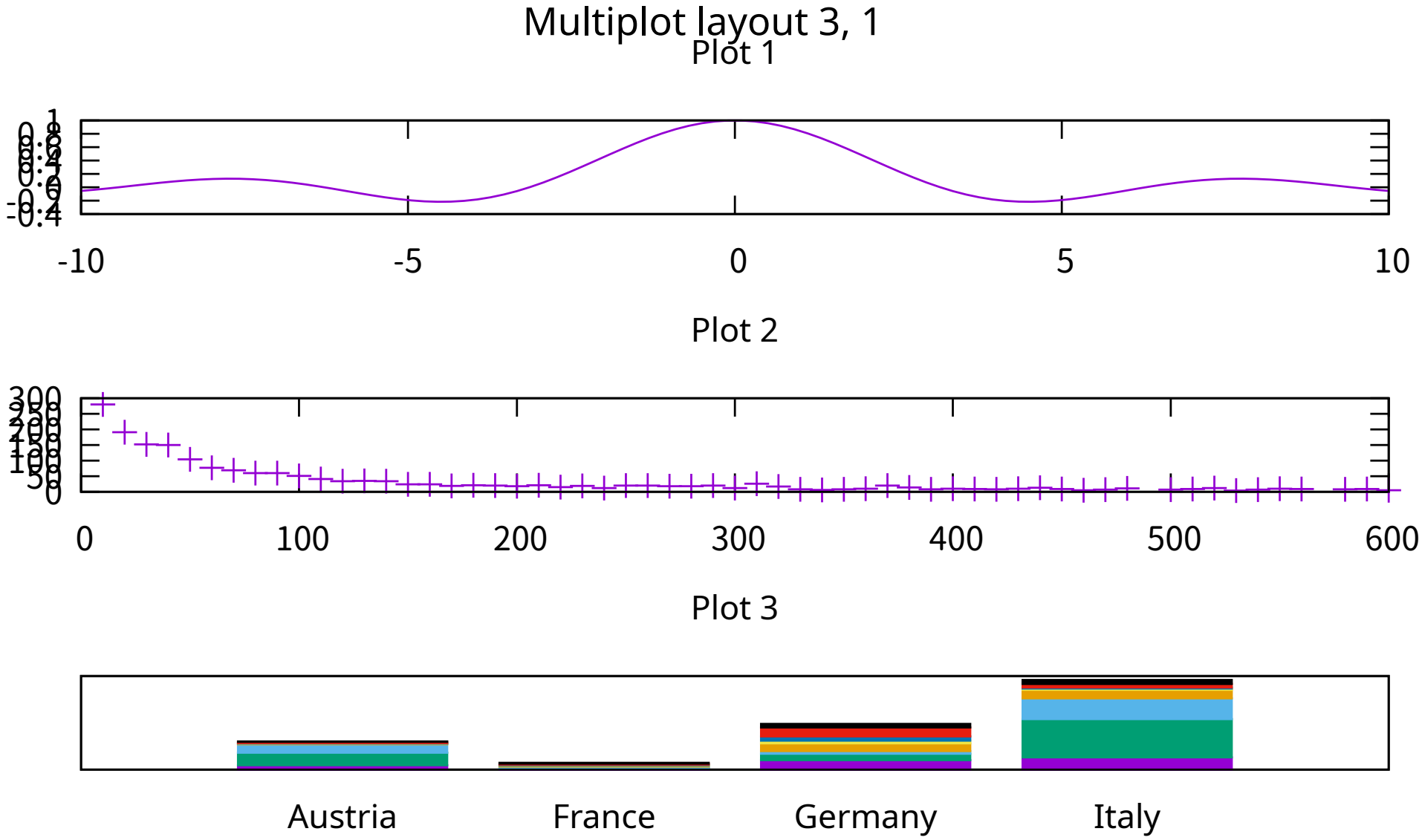

Sampling one dimension in 3D
[t=10:100:10] '+' using (t):(t):(t)
1000
100
10
1
1
10
100
1000
100
90
80
70
60
50
40
30
20
10

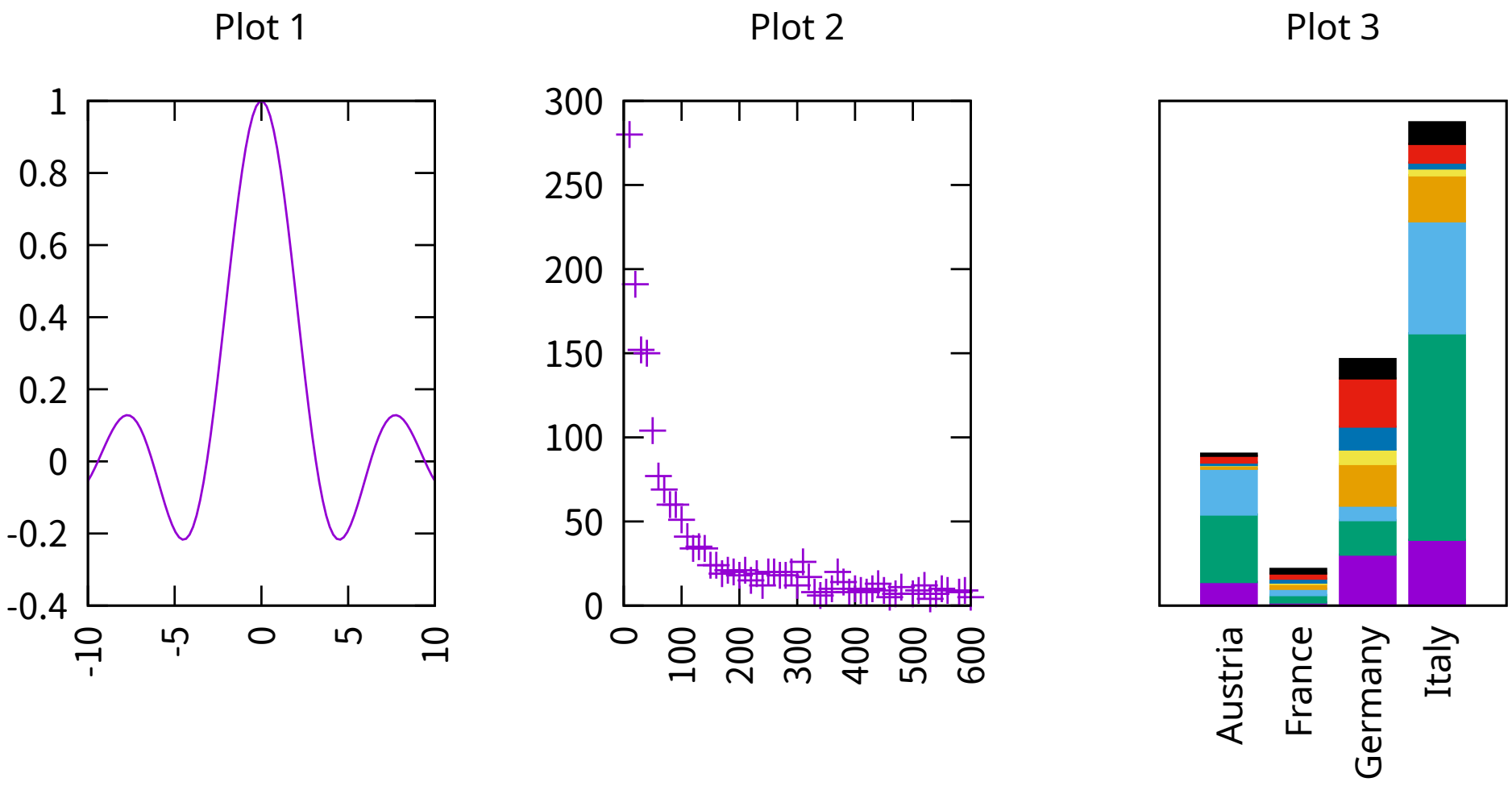

Second Order System Transfer Function - Magnitude
Damping =.1,.2,.3,.4,.5,.707,1.0,2.0
Magnitude (dB)
20
10
0
-10
-20
-30
-40
0.1
1
10
Frequency (w/wn)


Second Order System - Unit Step Respons
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)

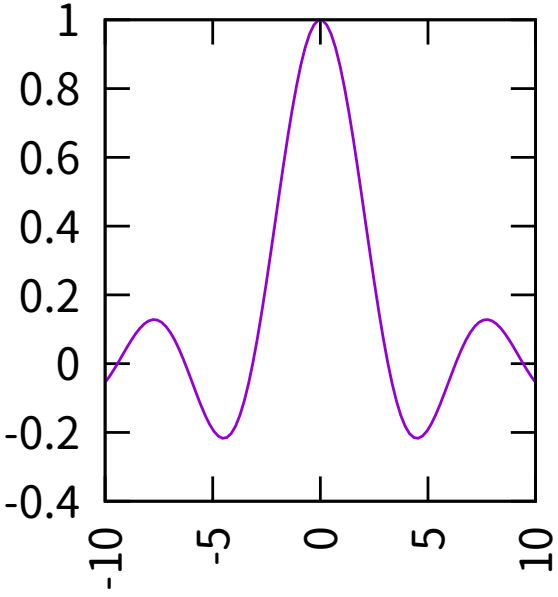

Second Order System - Unit Impulse Respor
Amplitude y(wnt)
1
0.5
0
-0.5
-1
0
5
10
15
20
Normalized Time (wnt)


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

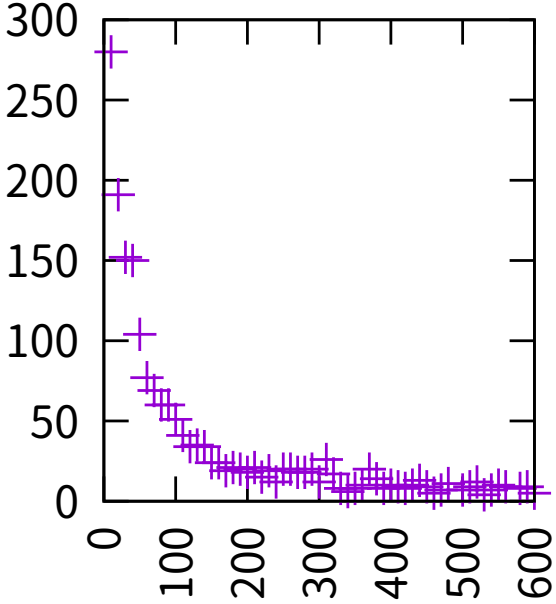

Multiplot layout 1, 3
Plot 1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10
Plot 2
300
250
200
150
100
50
0
0
100
200
300
400
500
600
Plot 3
Austria
France
Germany
Italy

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

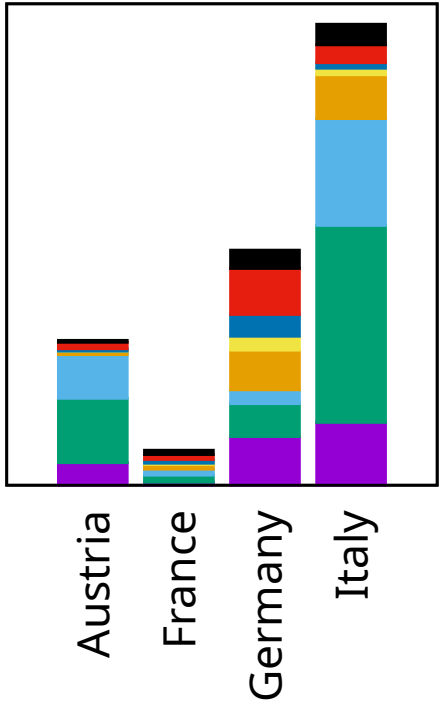
Plot 1

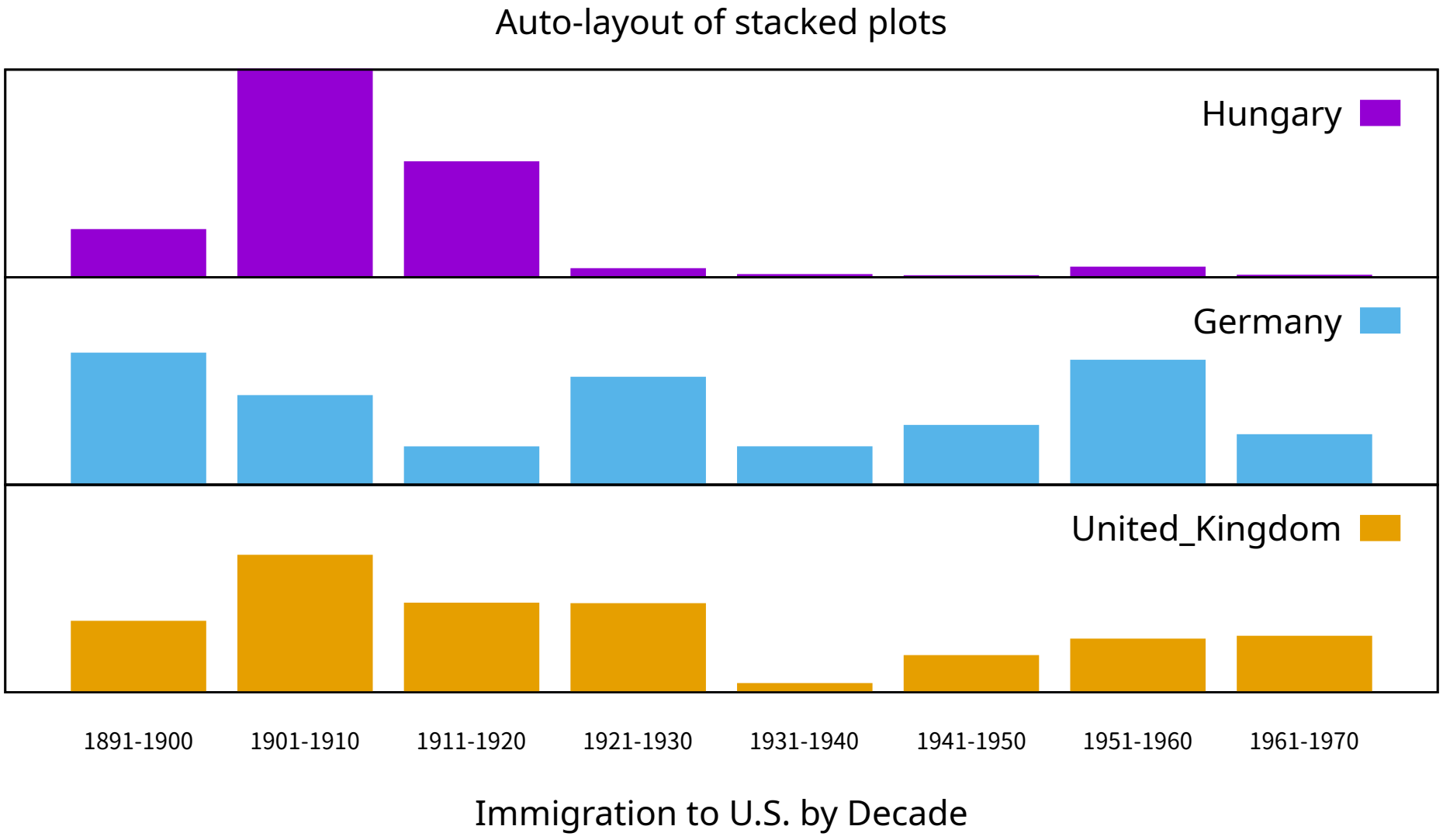


Plot 2

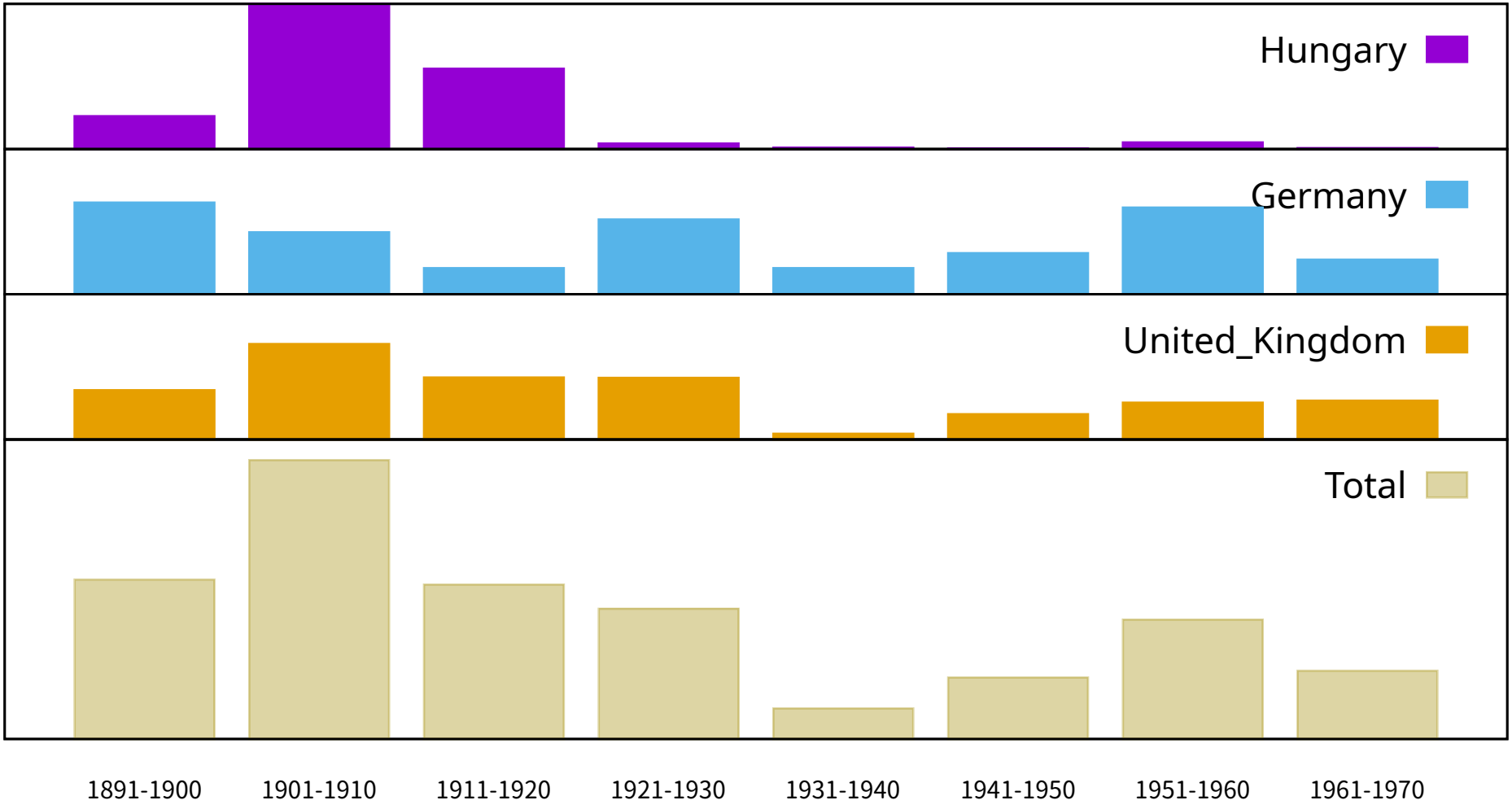


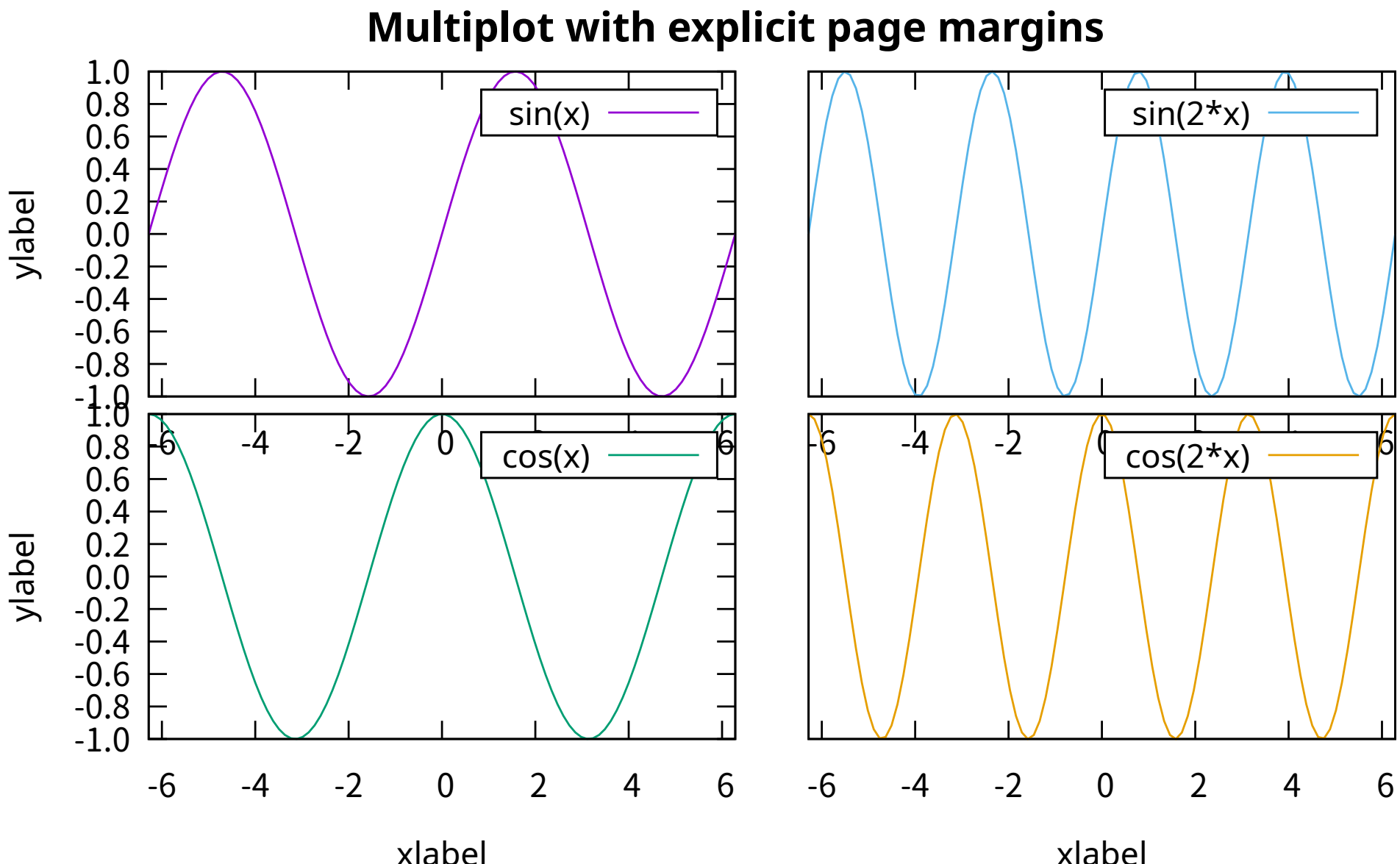
Plot 3



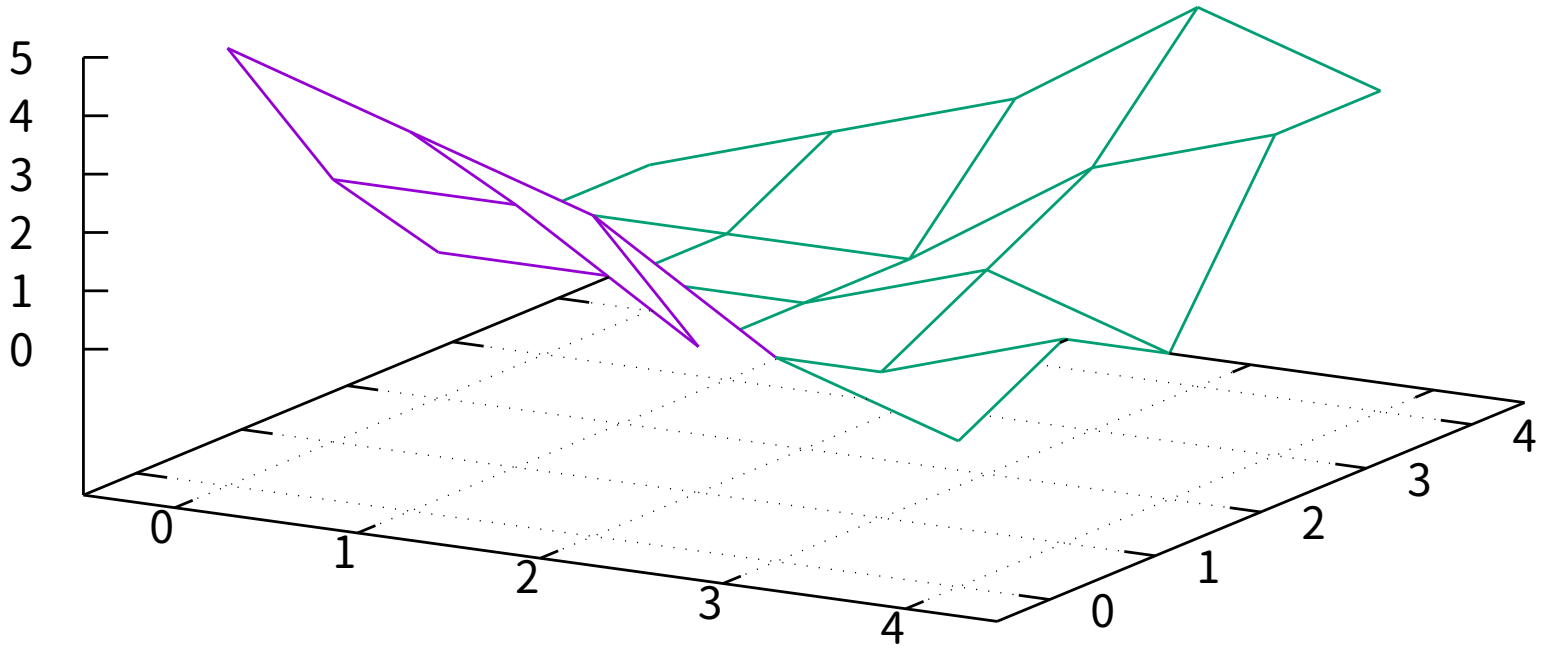

Auto-layout of stacked plots
Hungary
Germany
United_Kingdom
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Immigration to U.S. by Decade

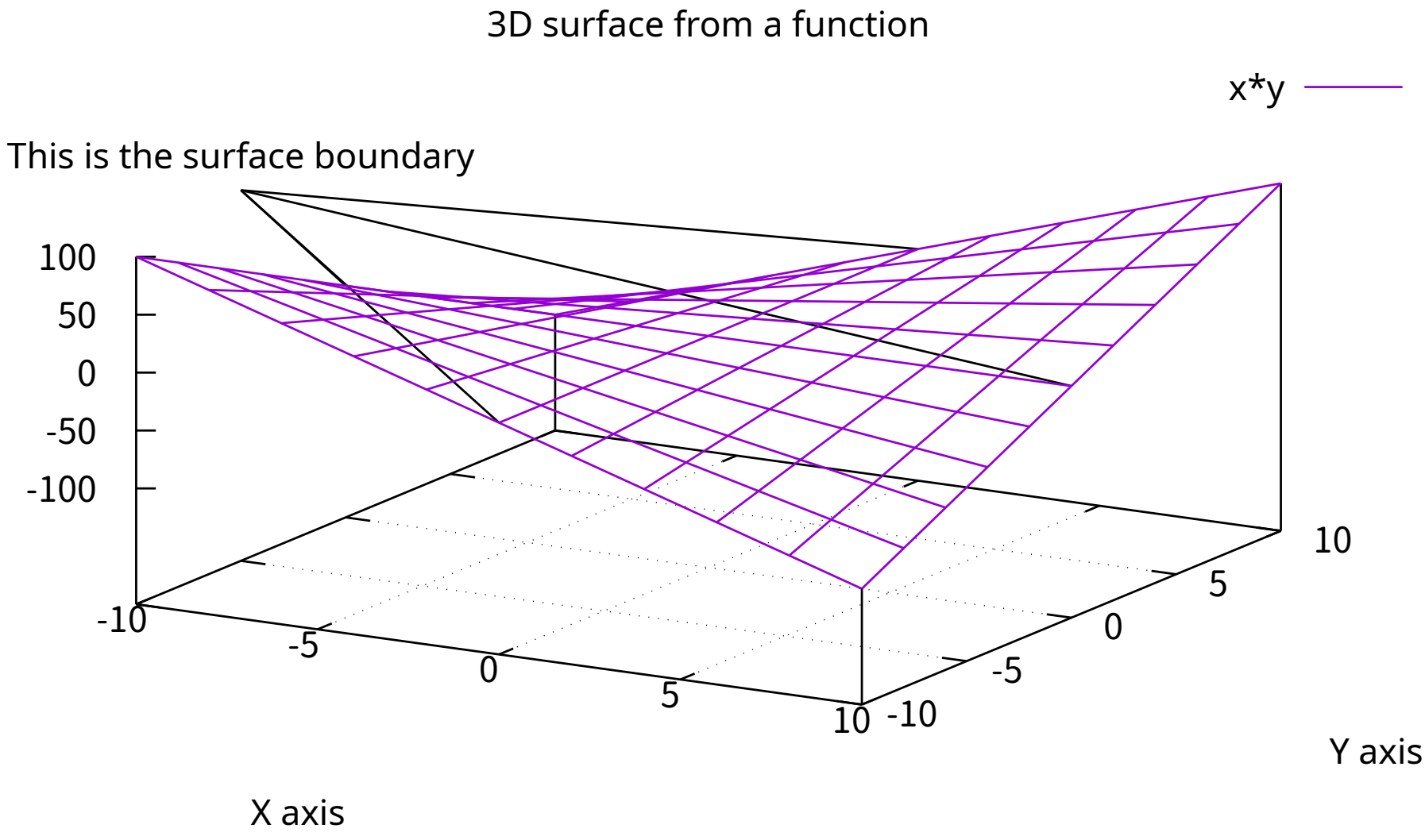
Expanding one of the plots to use additional space

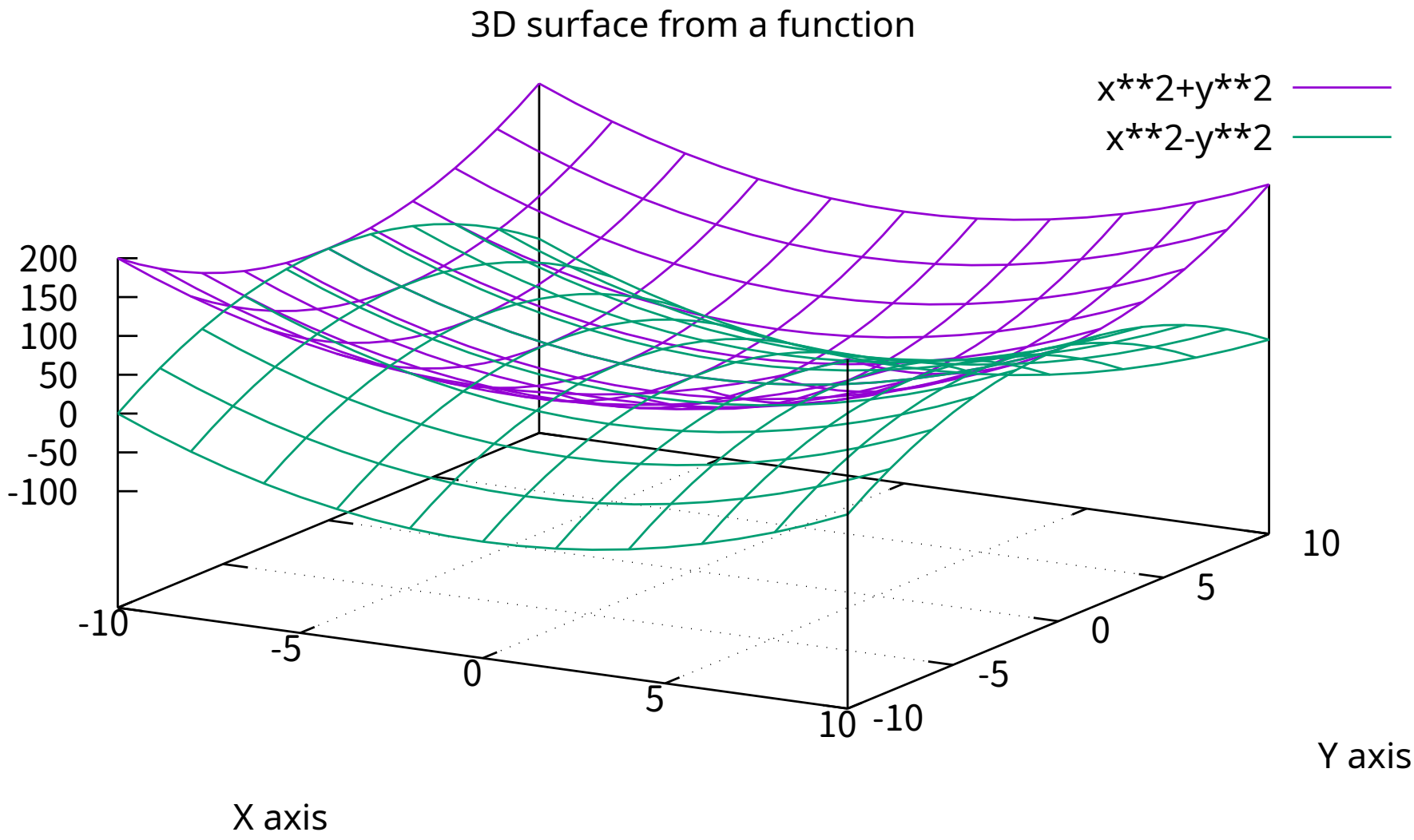


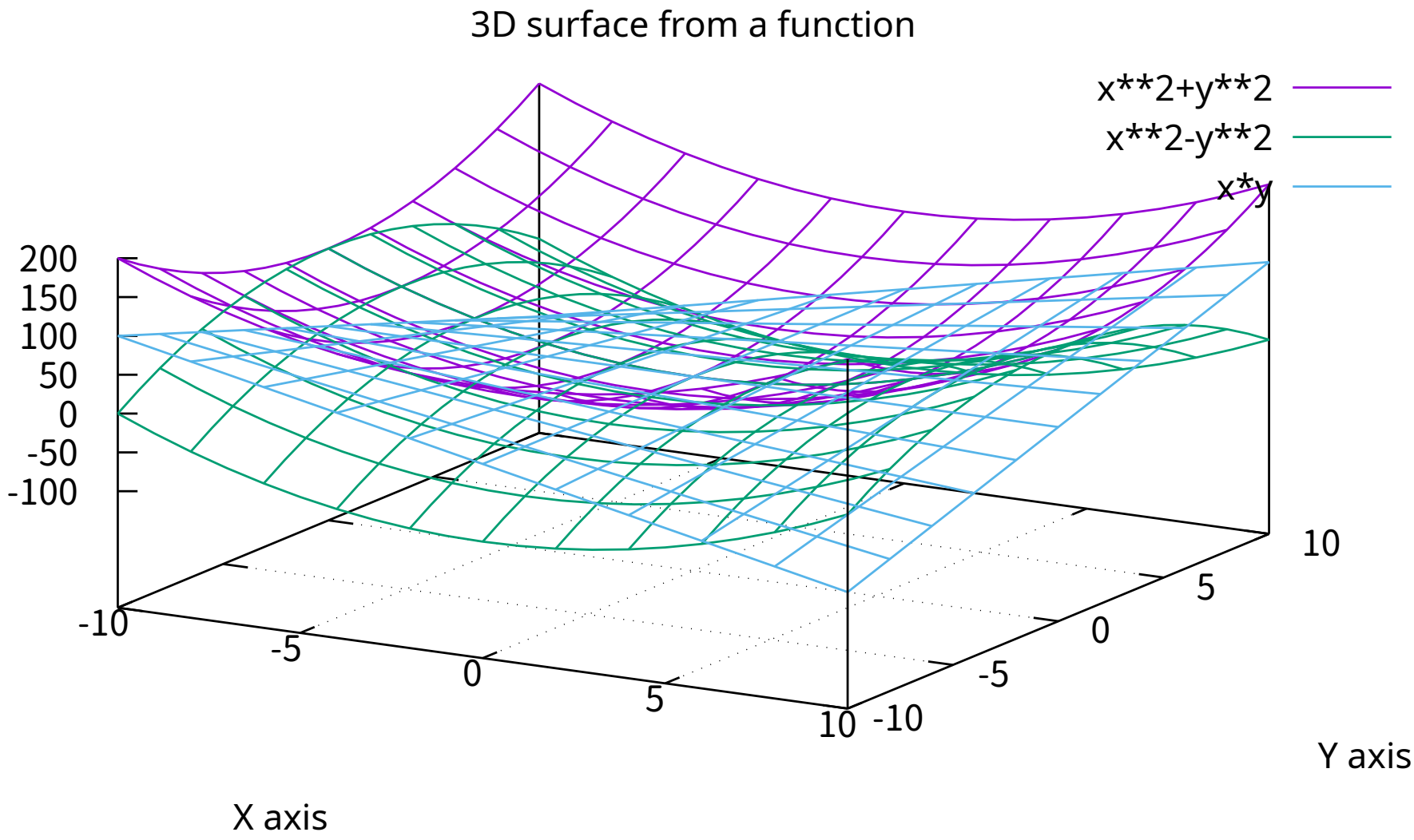

Multiplot with explicit page margins
sin(x)
sin(2*x)
cos(x)
cos(2*x)
ylabel
ylabel
xlabel
xlabel
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-6
-4
-2
0
2
4
6

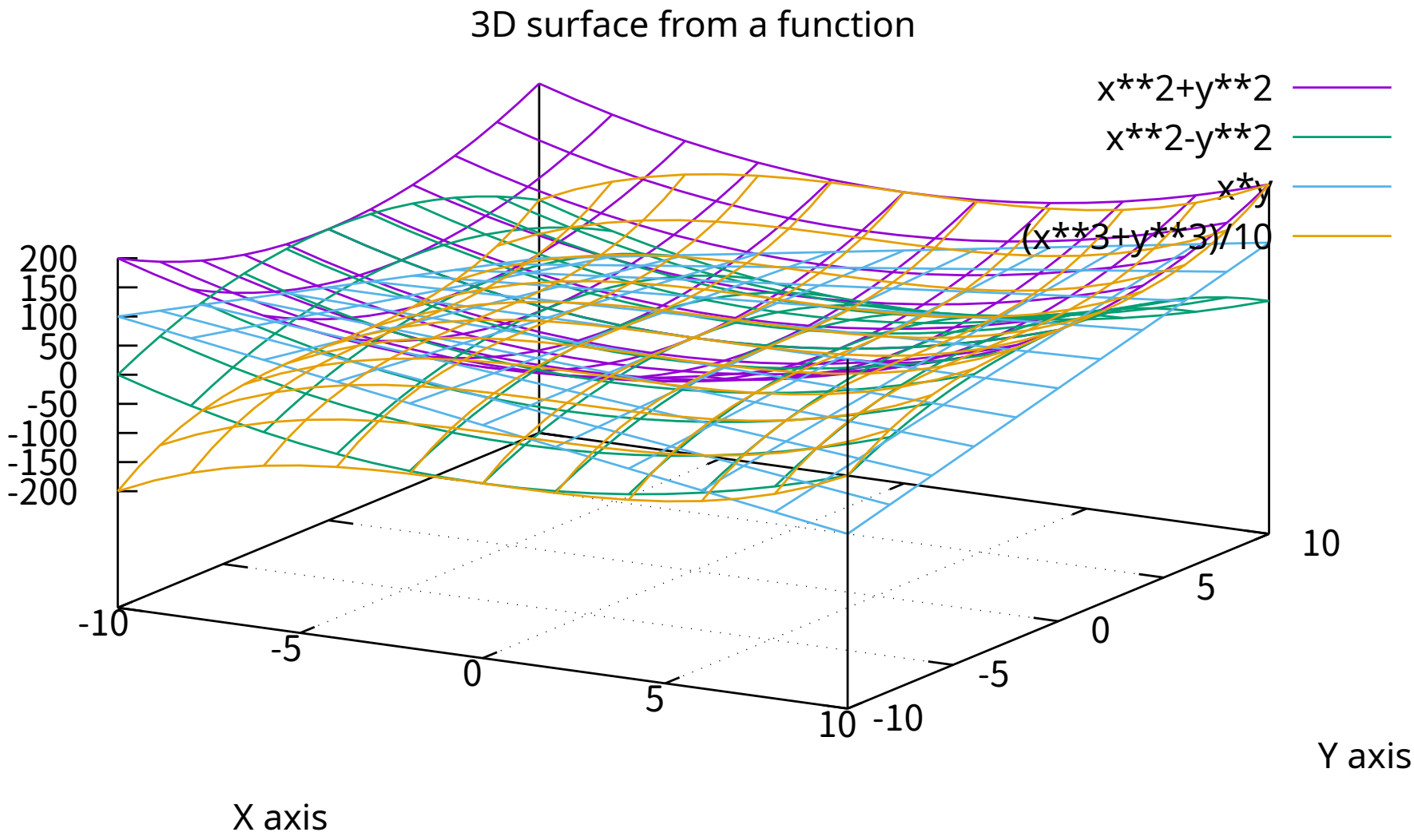
3D surface from a grid (matrix) of Z values

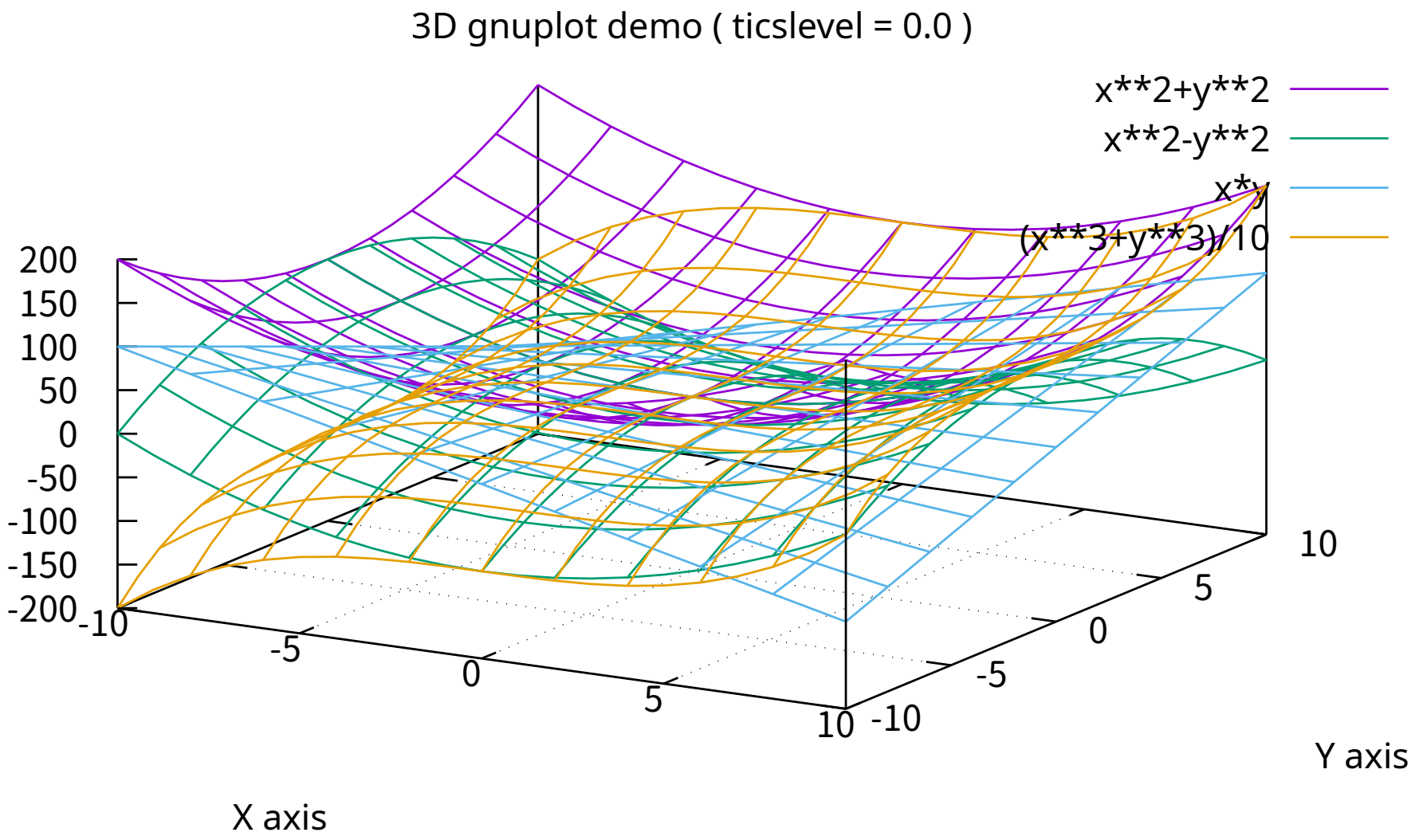


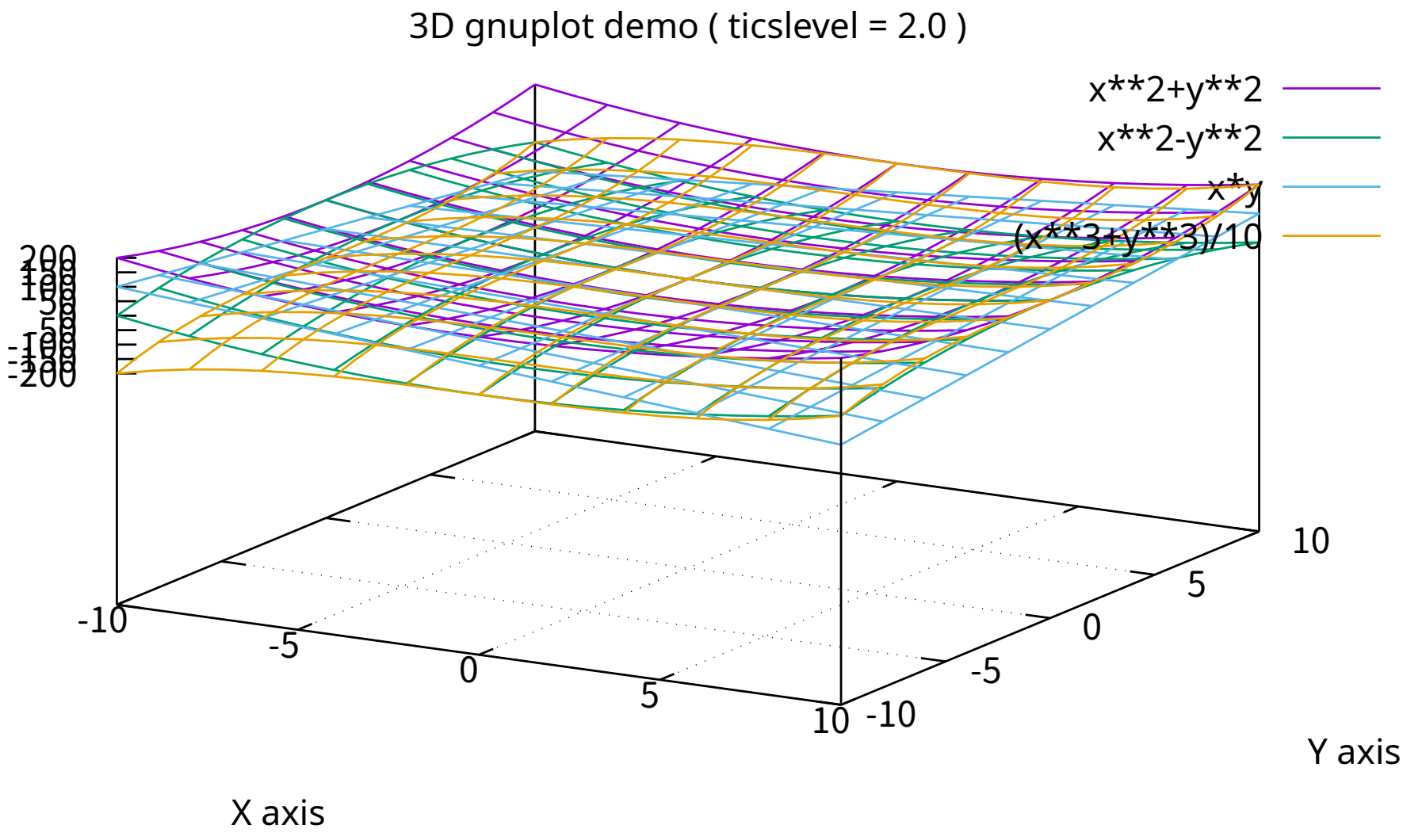

3D surface from a function
x*y
This is the surface boundary
100
50
0
-50
-100
-10
-5
0
5
10
-10
-5
0
5
10
X axis
Y axis

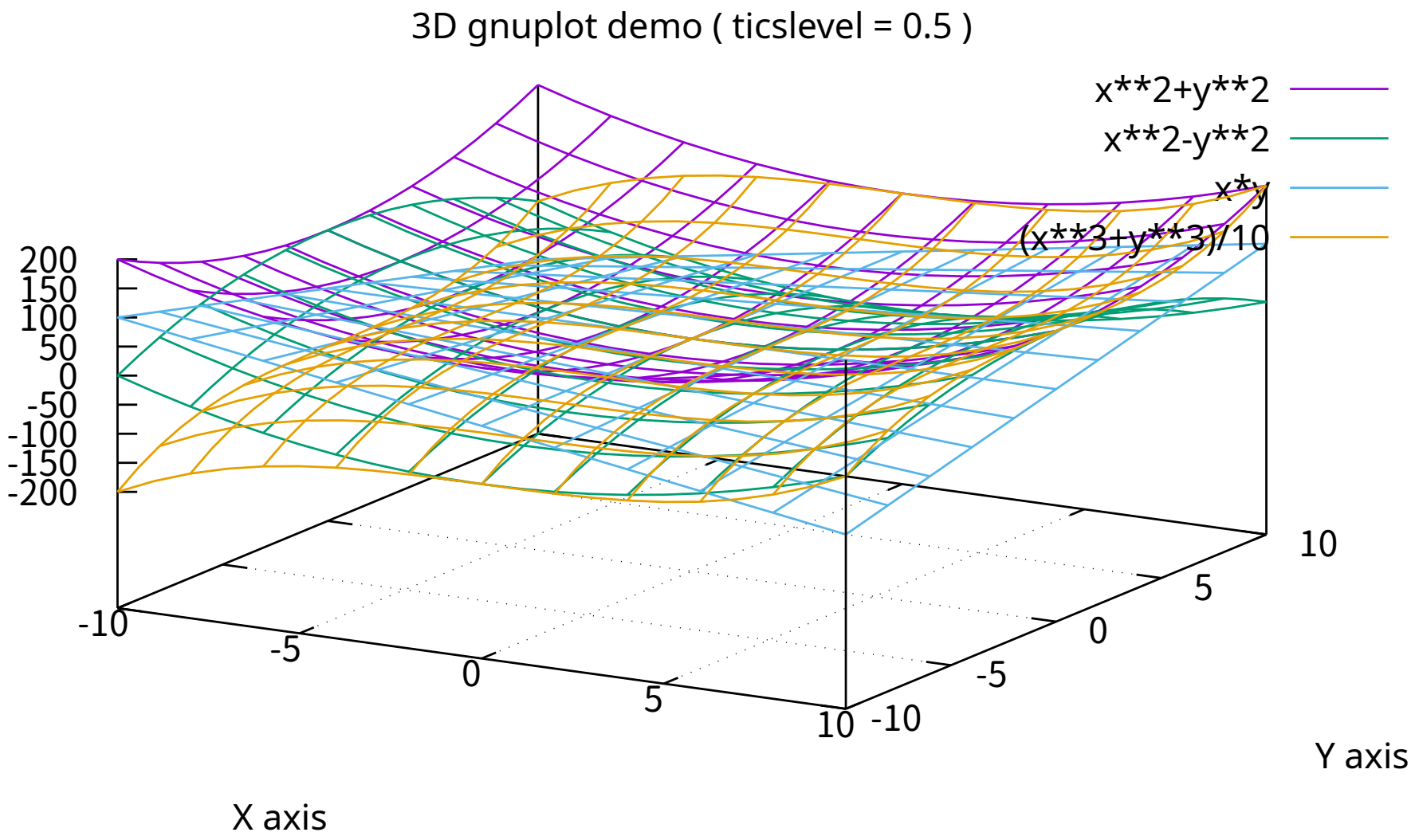

3D surface from a function
x**2+y**2
x**2-y**2
200
150
100
50
0
-50
-100
-10
-5
0
5
10
-10
-5
0
5
10
X axis
Y axis

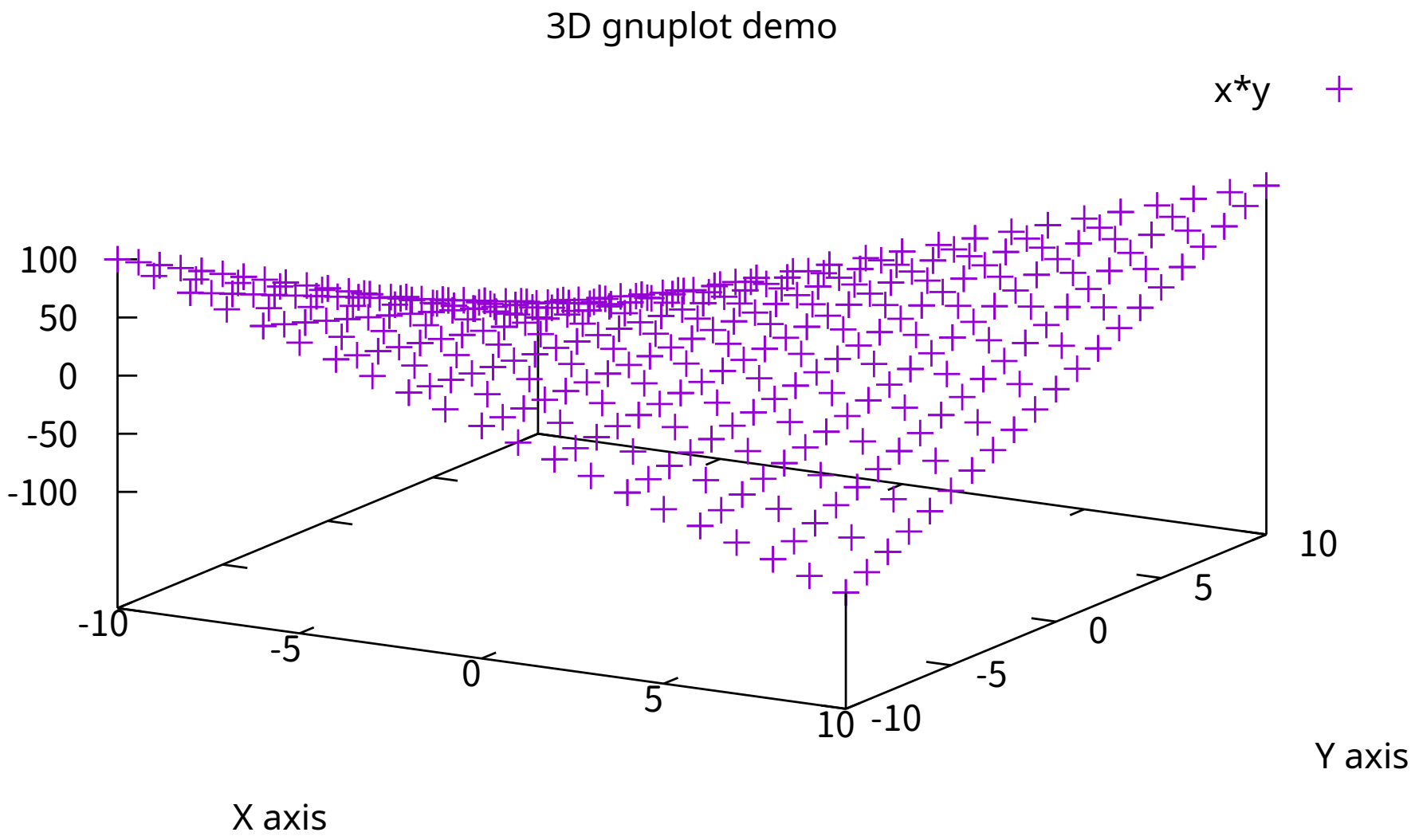

3D surface from a function
x**2+y**2
x**2-y**2
x*y
200
150
100
50
0
-50
-100
-10
-5
0
5
10
-10
-5
0
5
10
X axis
Y axis

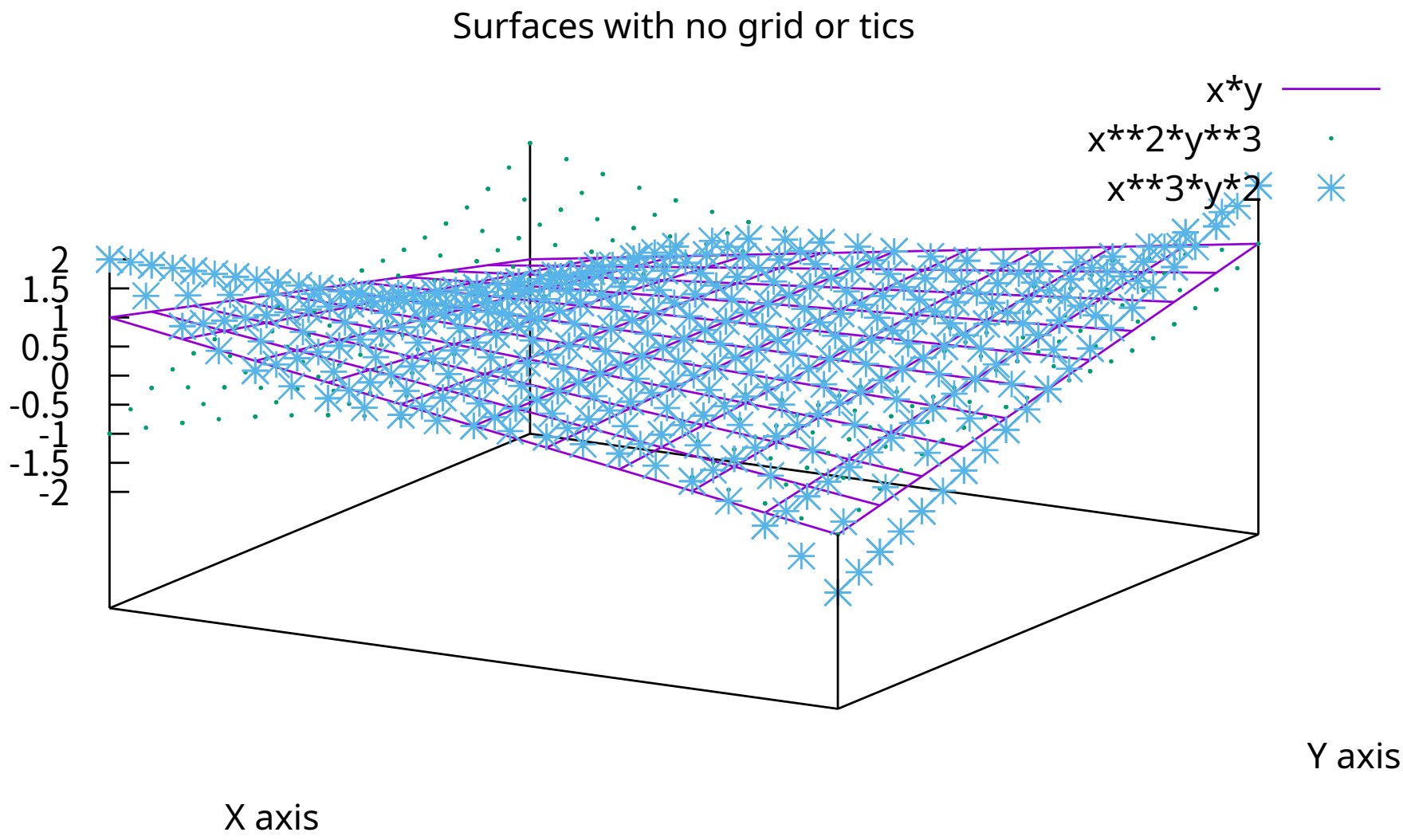

3D surface from a function
x**2+y**2
x**2-y**2
x*y
(x**3+y**3)/10
200
150
100
50
0
-50
-100
-150
-200
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

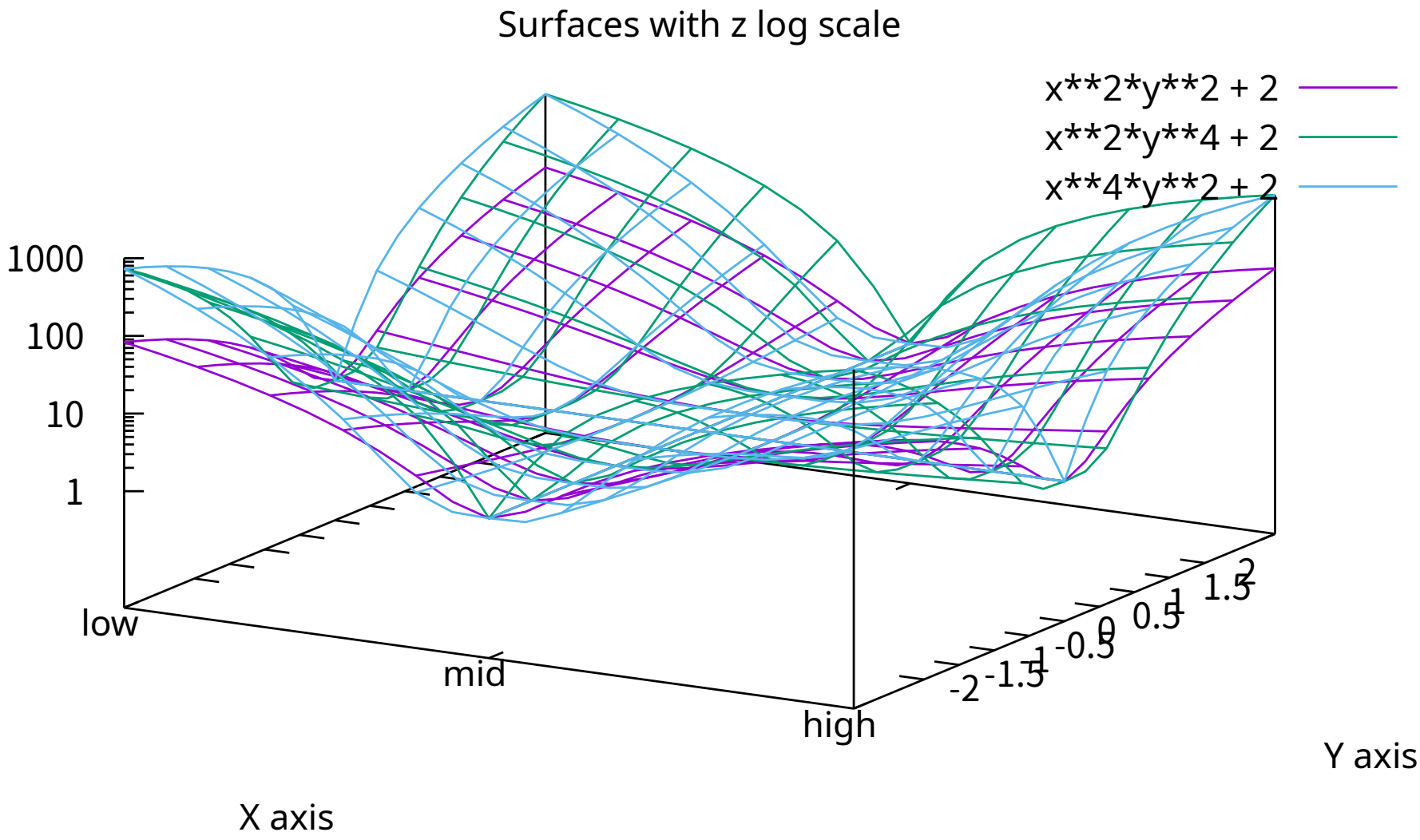

3D gnuplot demo (ticslevel = 0.0)
x**2+y**2
x**2-y**2
x*y
(x**3+y**3)/10
200
150
100
50
0
-50
-100
-150
-200
-10
-5
0
5
10
-10
-5
0
5
10
X axis
Y axis

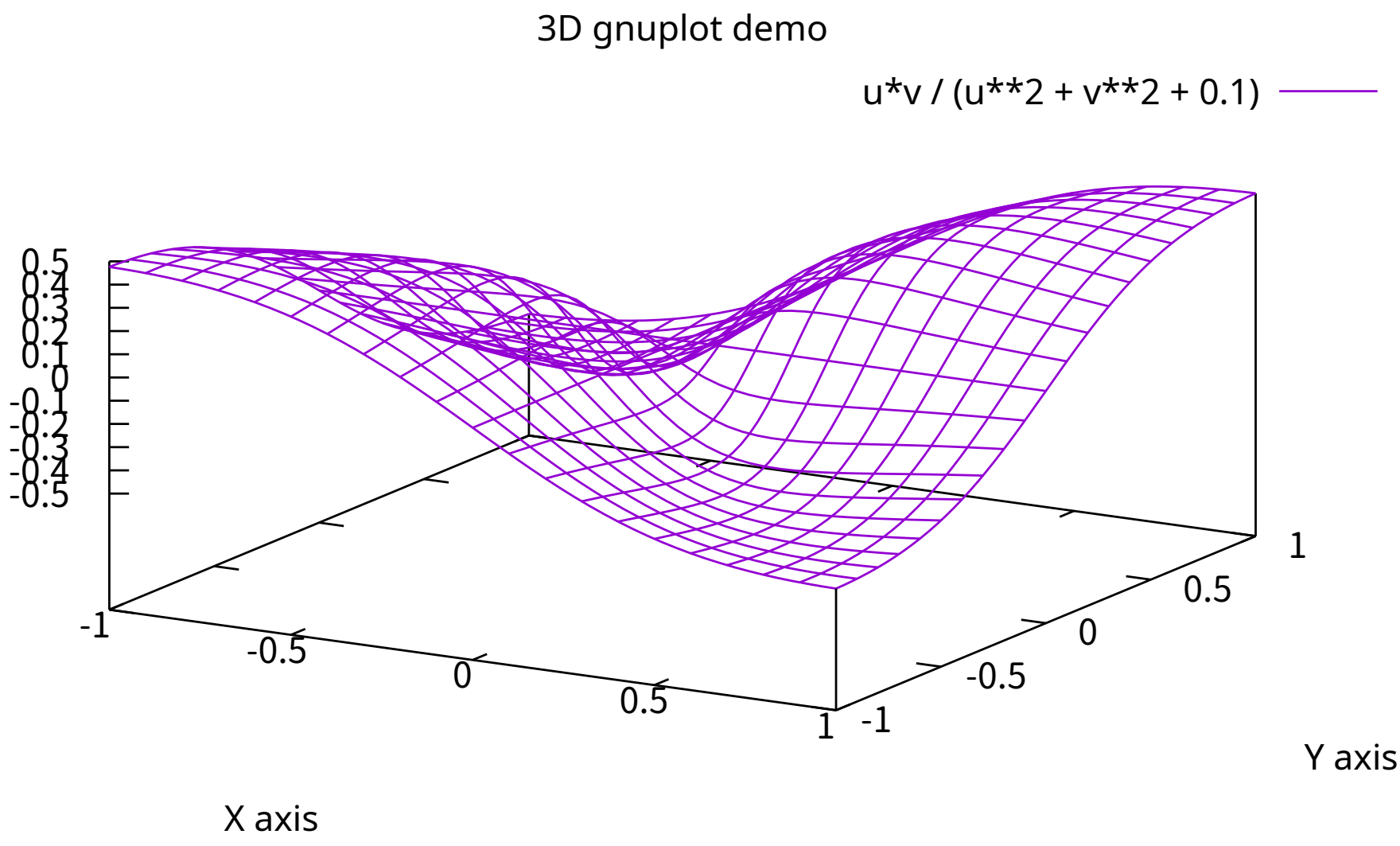

3D gnuplot demo (ticslevel = 2.0)
x**2+y**2
x**2-y**2
x*y
(x**3+y**3)/10
200
-200
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

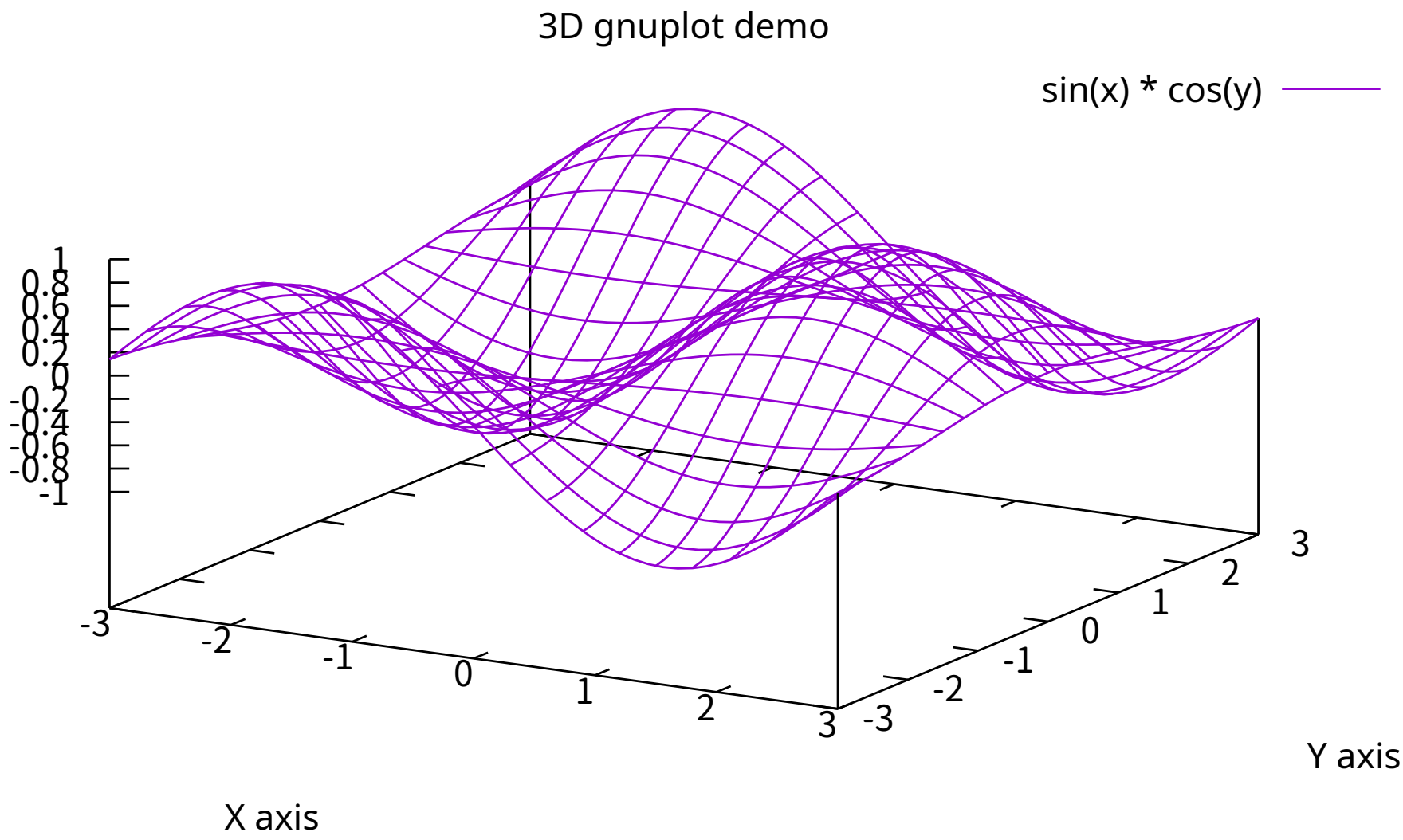

3D gnuplot demo (ticslevel = 0.5)
x**2+y**2
x**2-y**2
x*y
(x**3+y**3)/10
200
150
100
50
0
-50
-100
-150
-200
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

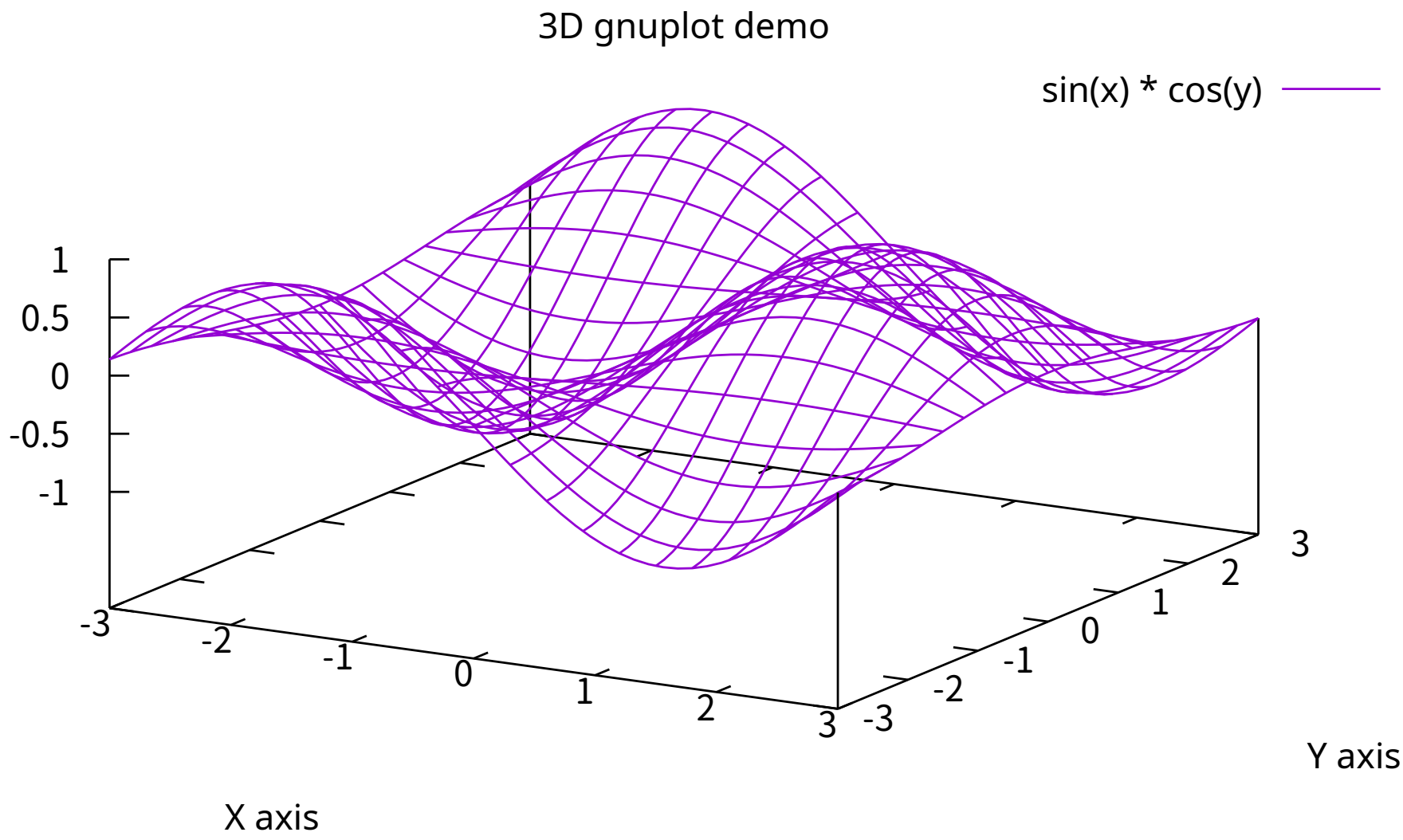

3D gnuplot demo
x*y
100
50
0
-50
-100
-10
-5
0
5
10
-10
-5
0
5
10
X axis
Y axis

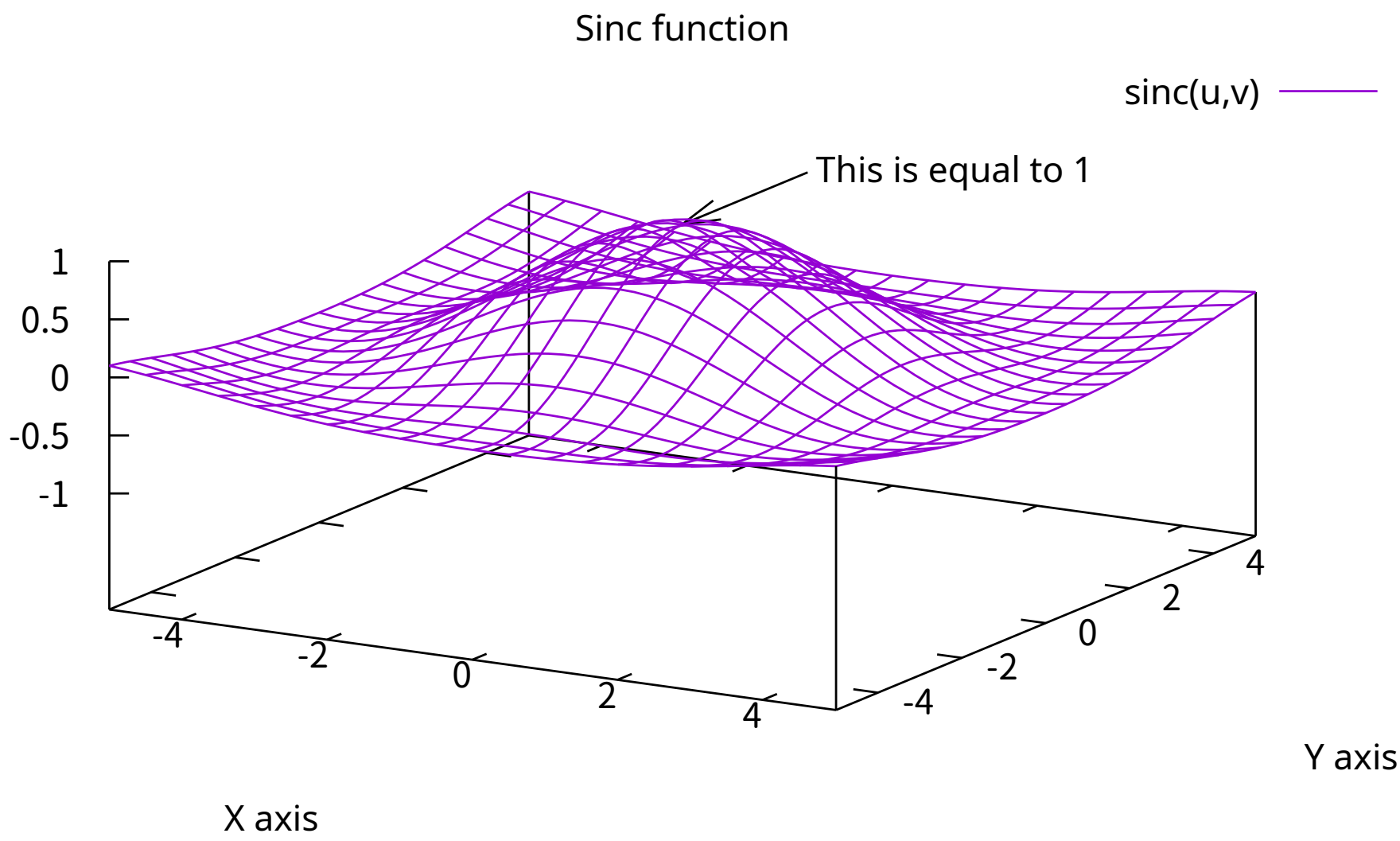

Surfaces with no grid or tics
x*y
x**2*y**3
x**3*y*2
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
Y axis
X axis

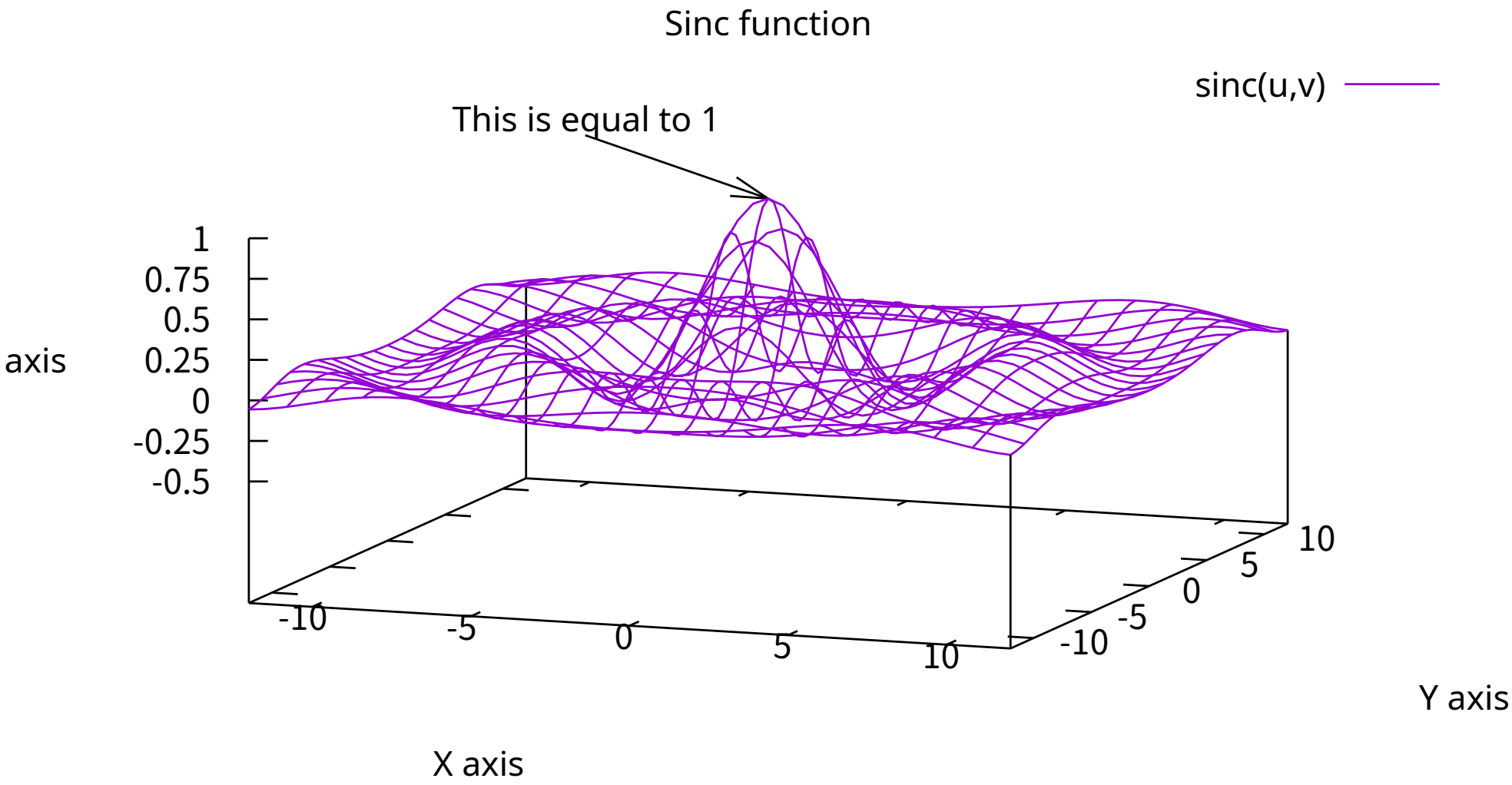

Surfaces with z log scale
x**2*y**2 + 2
x**2*y**4 + 2
x**4*y**2 + 2
1000
100
10
1
low
mid
high
X axis
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
Y axis


3D gnuplot demo
u*v / (u**2 + v**2 + 0.1)
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

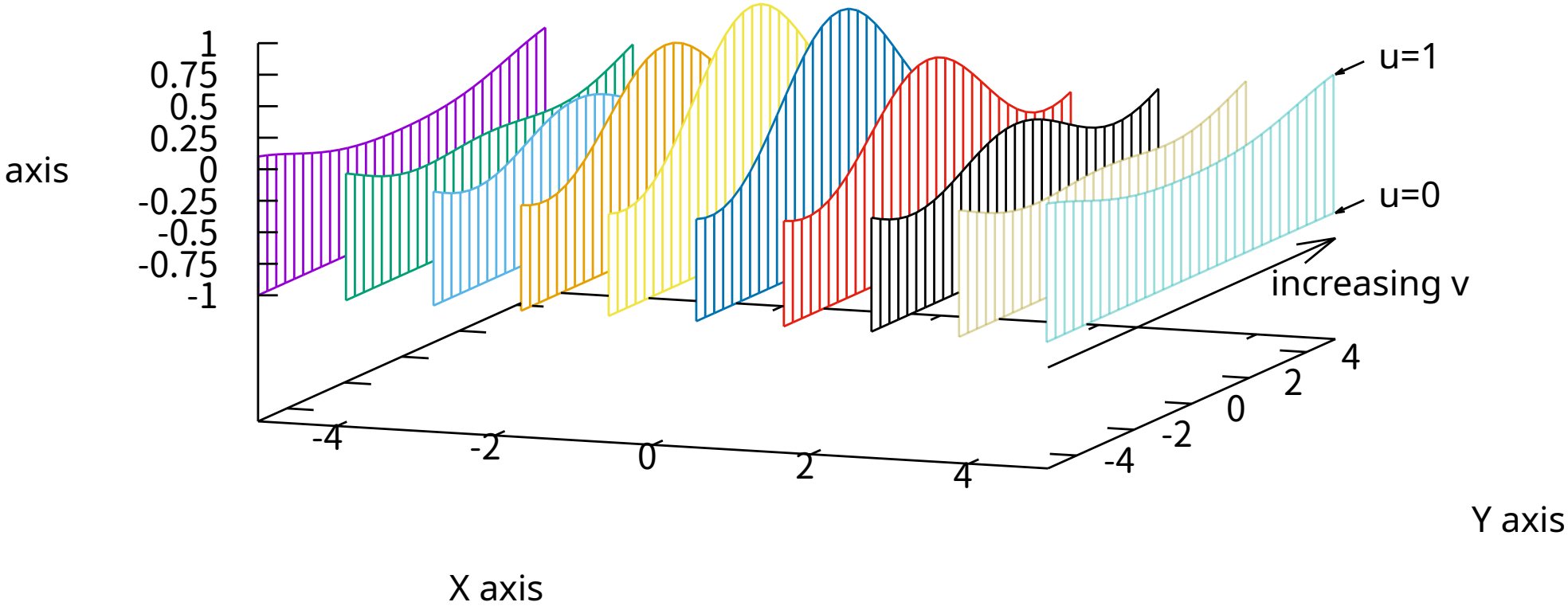

3D gnuplot demo
sin(x) * cos(y)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-3
-2
-1
0
1
2
3
X axis
-3
-2
-1
0
1
2
3
Y axis


3D gnuplot demo
sin(x) * cos(y)
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

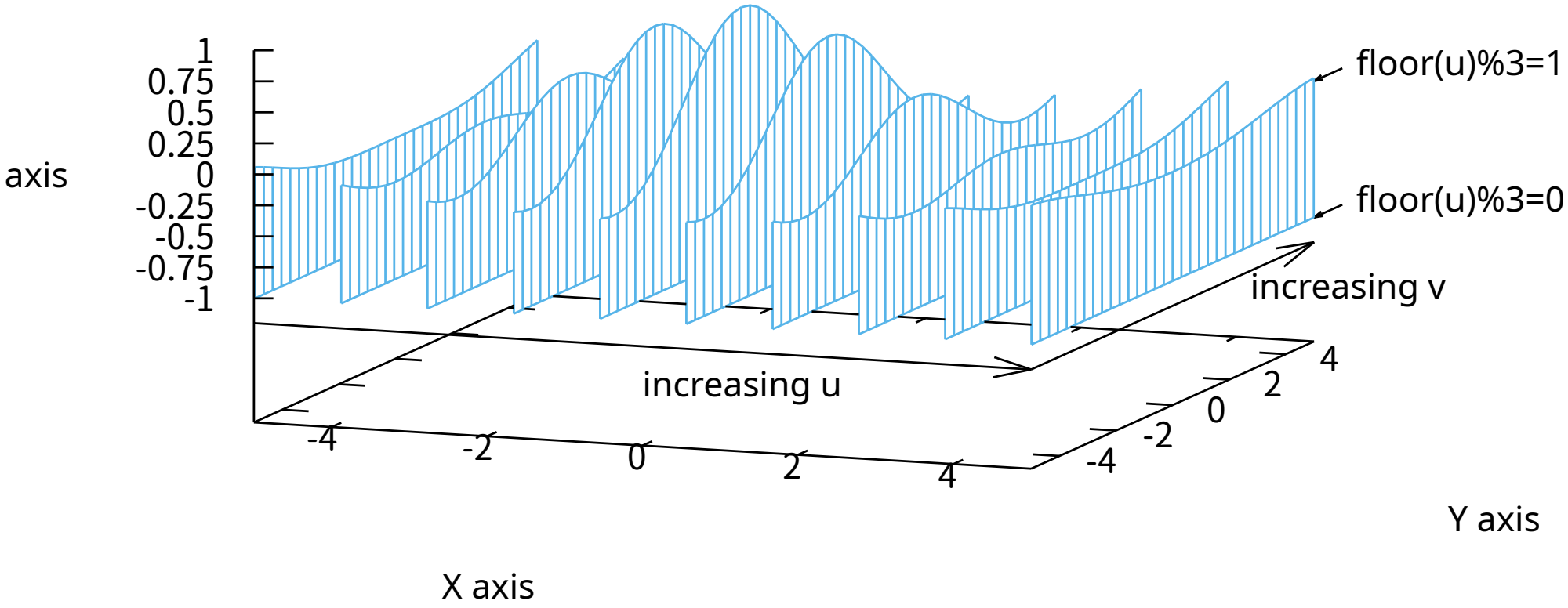

Sinc function
sinc(u,v)
This is equal to 1
1
0.5
0
-0.5
-1
-4
-2
0
2
4
-4
-2
0
2
4
Y axis
X axis

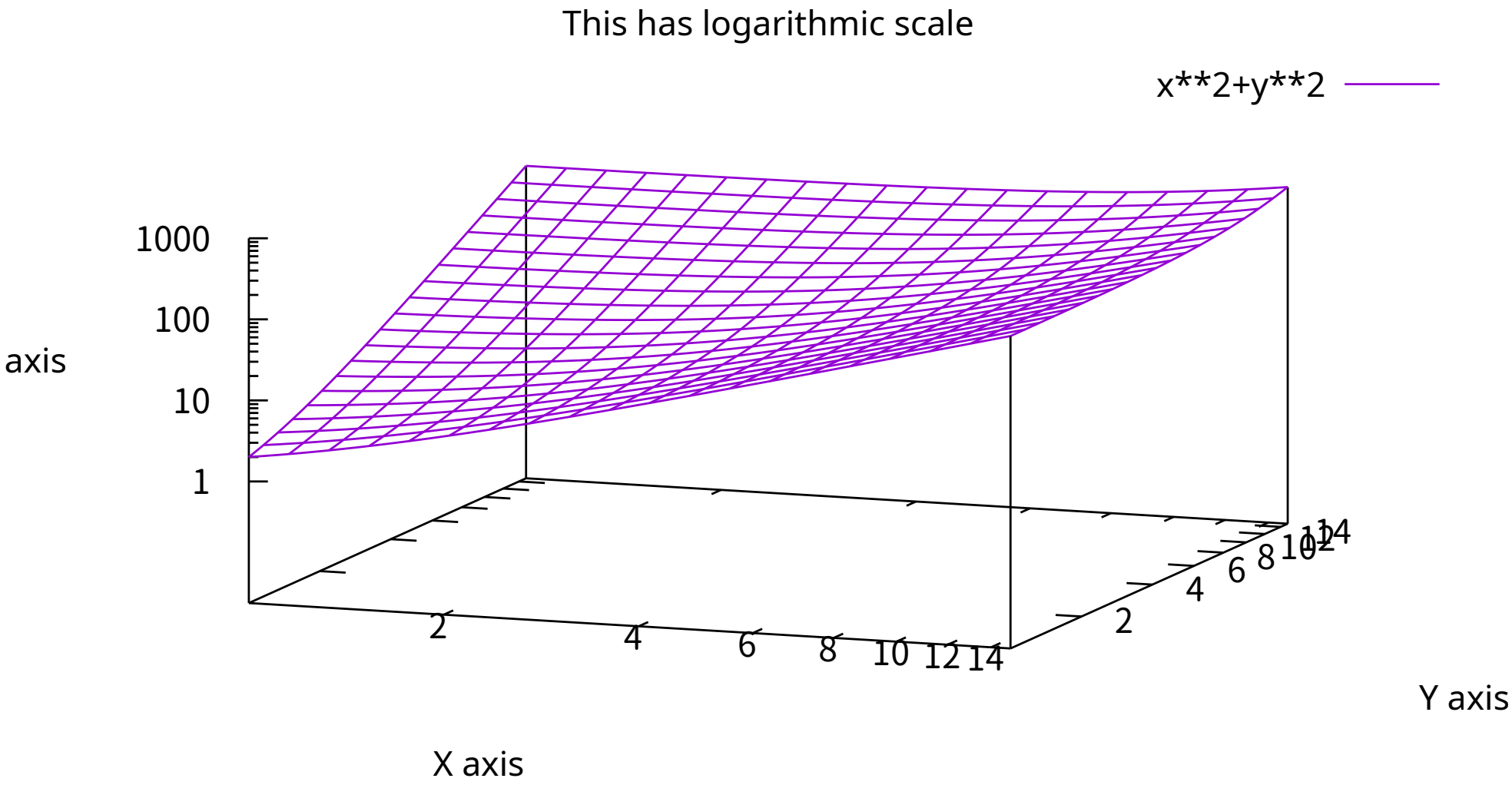

Sinc function
sinc(u,v)
This is equal to 1
1
0.75
0.5
0.25
0
-0.25
-0.5
axis
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

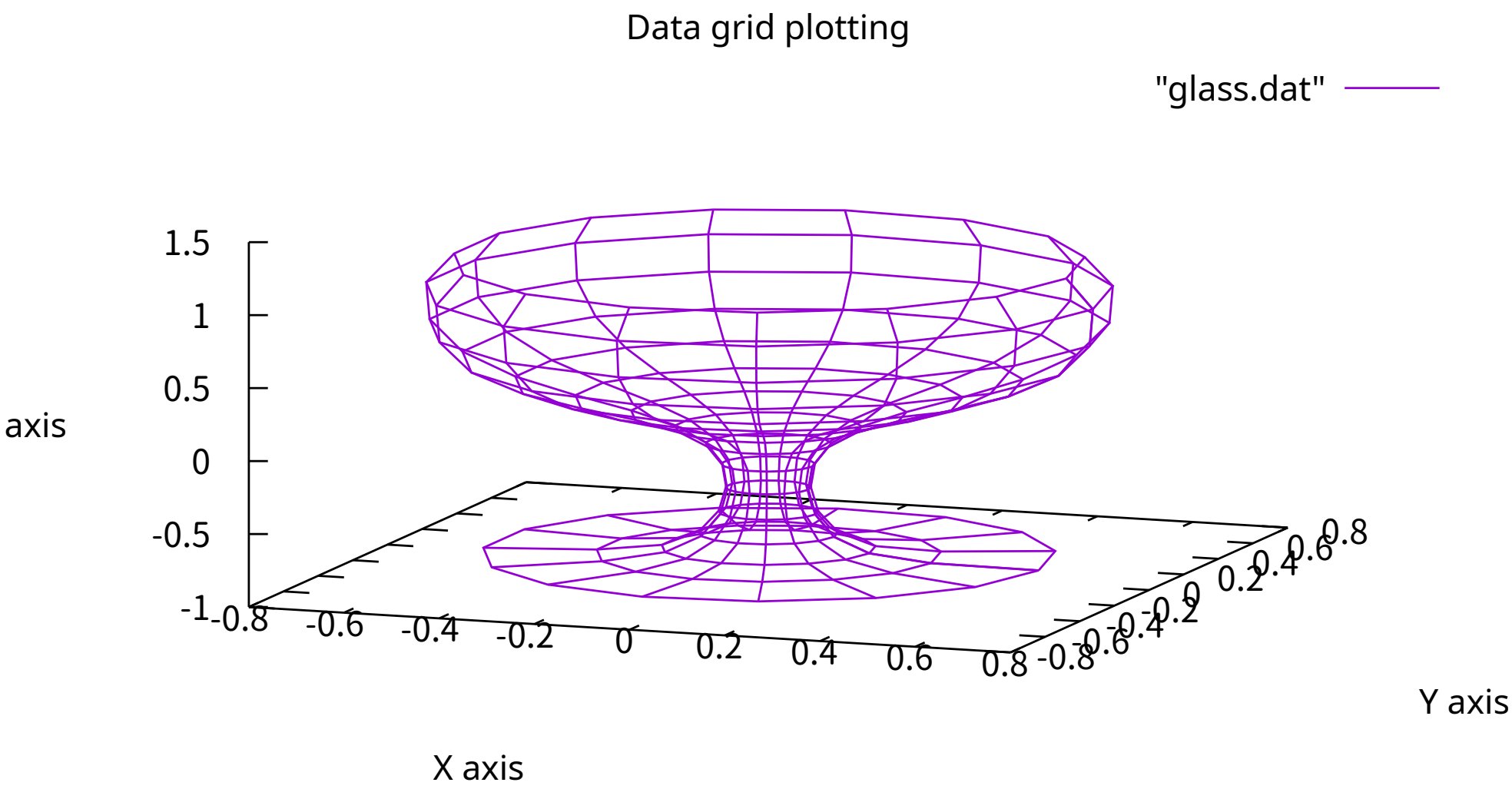
fence plot constructed with separate parametric surfaces

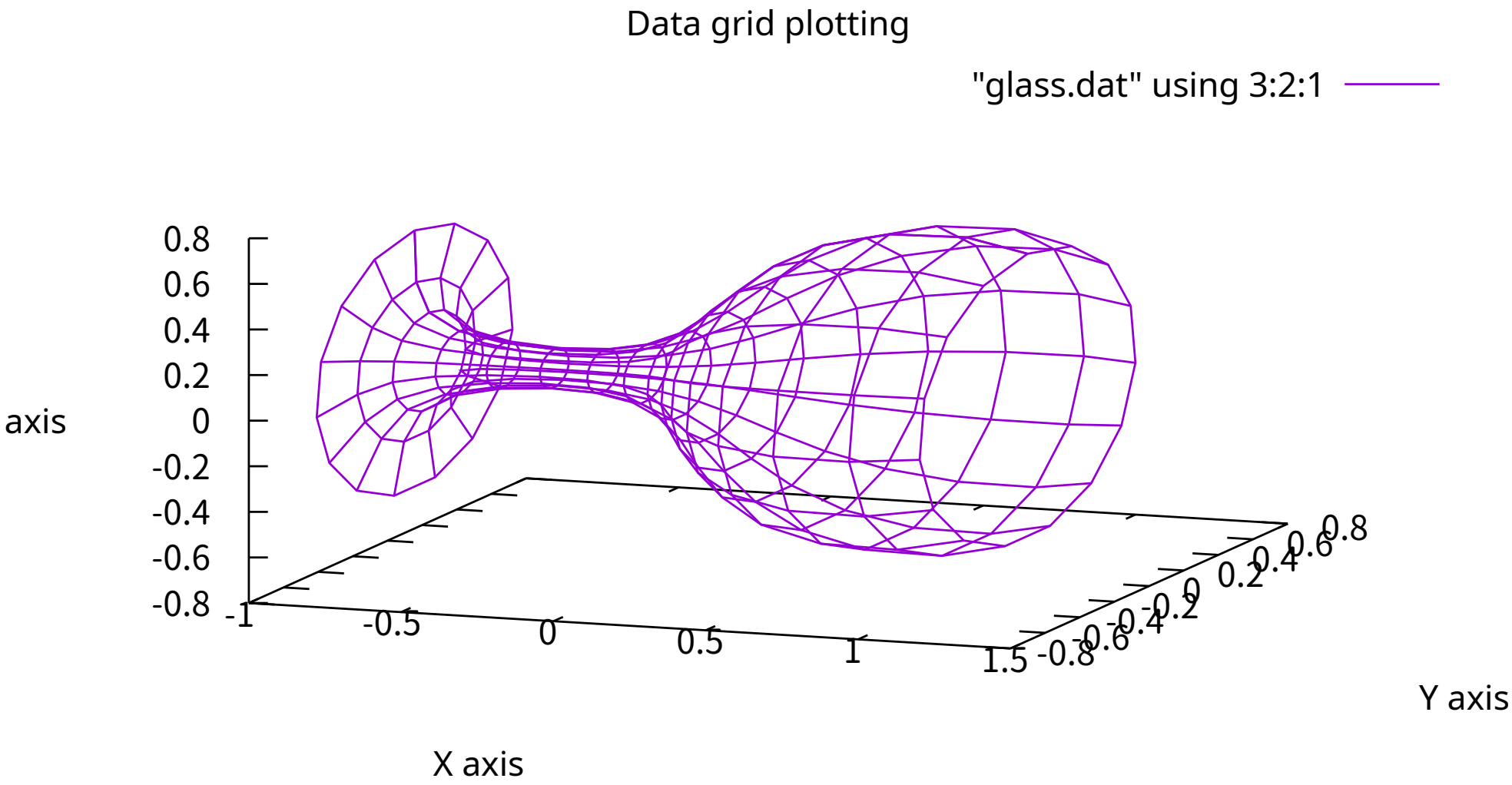


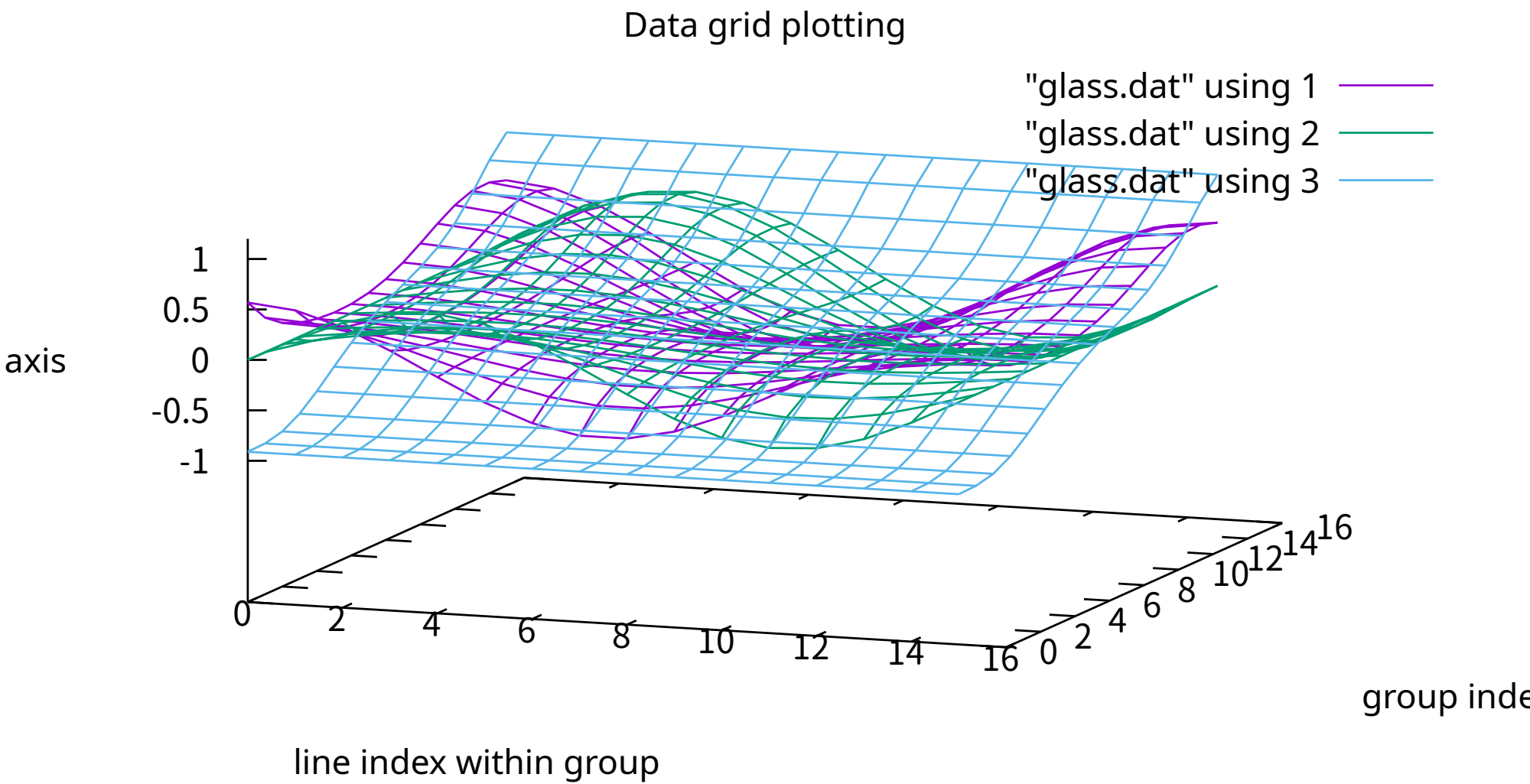
"fence plot" using single parametric surface with undefined points

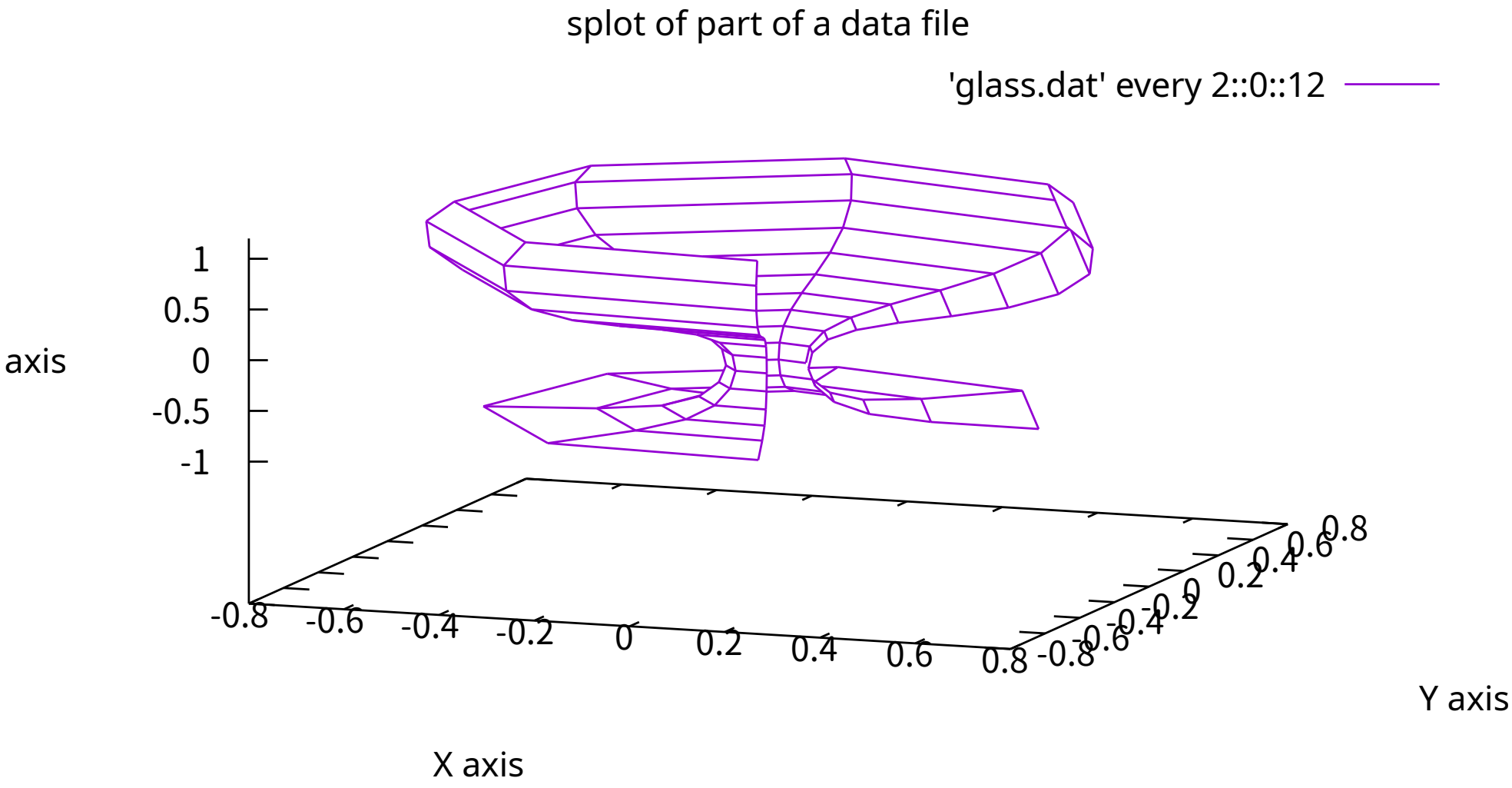


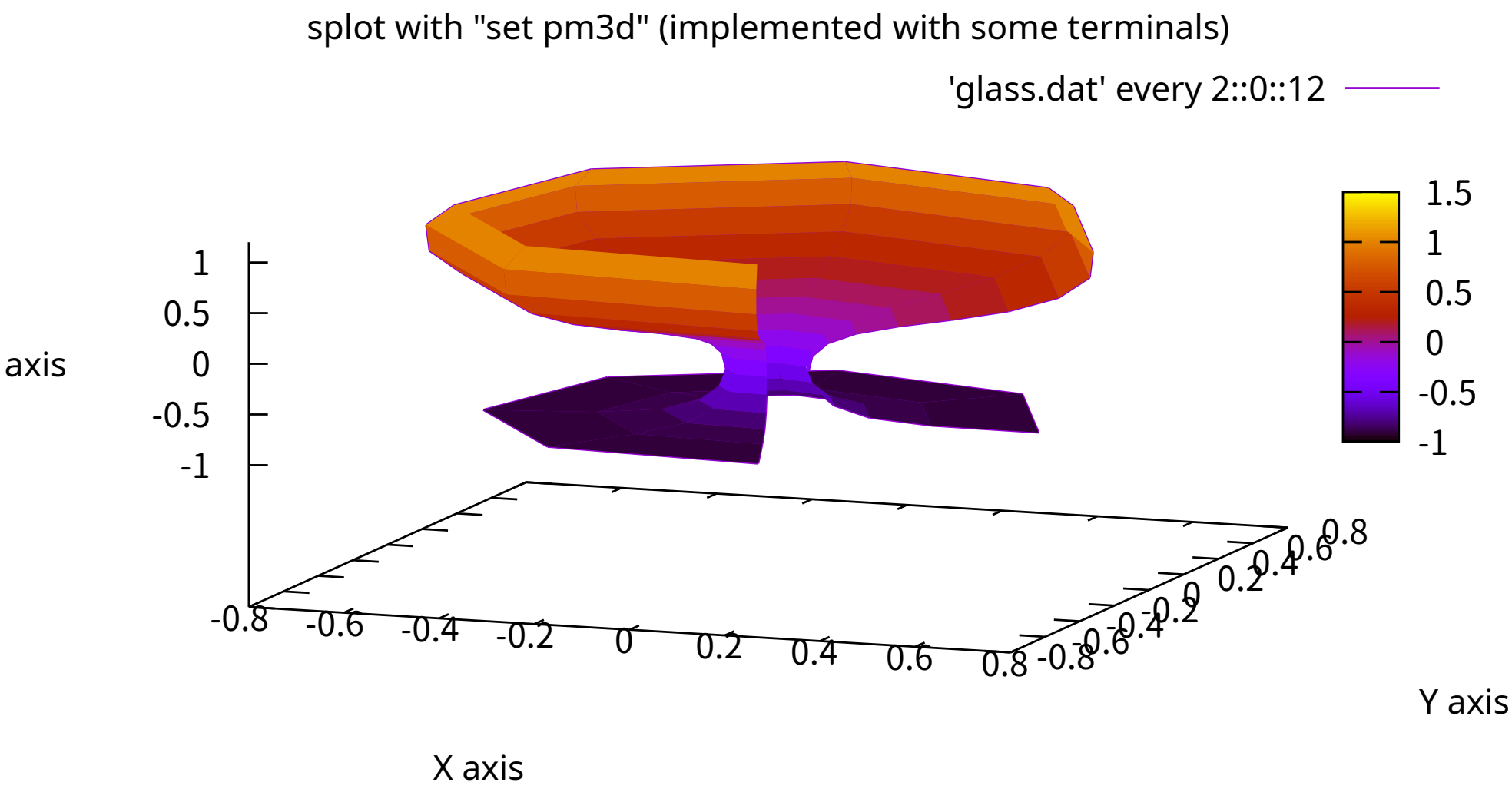

This has logarithmic scale
x**2+y**2
1000
100
10
1
axis
2
4
6
8
10
12
14
X axis
2
4
6
8
10
12
14
Y axis

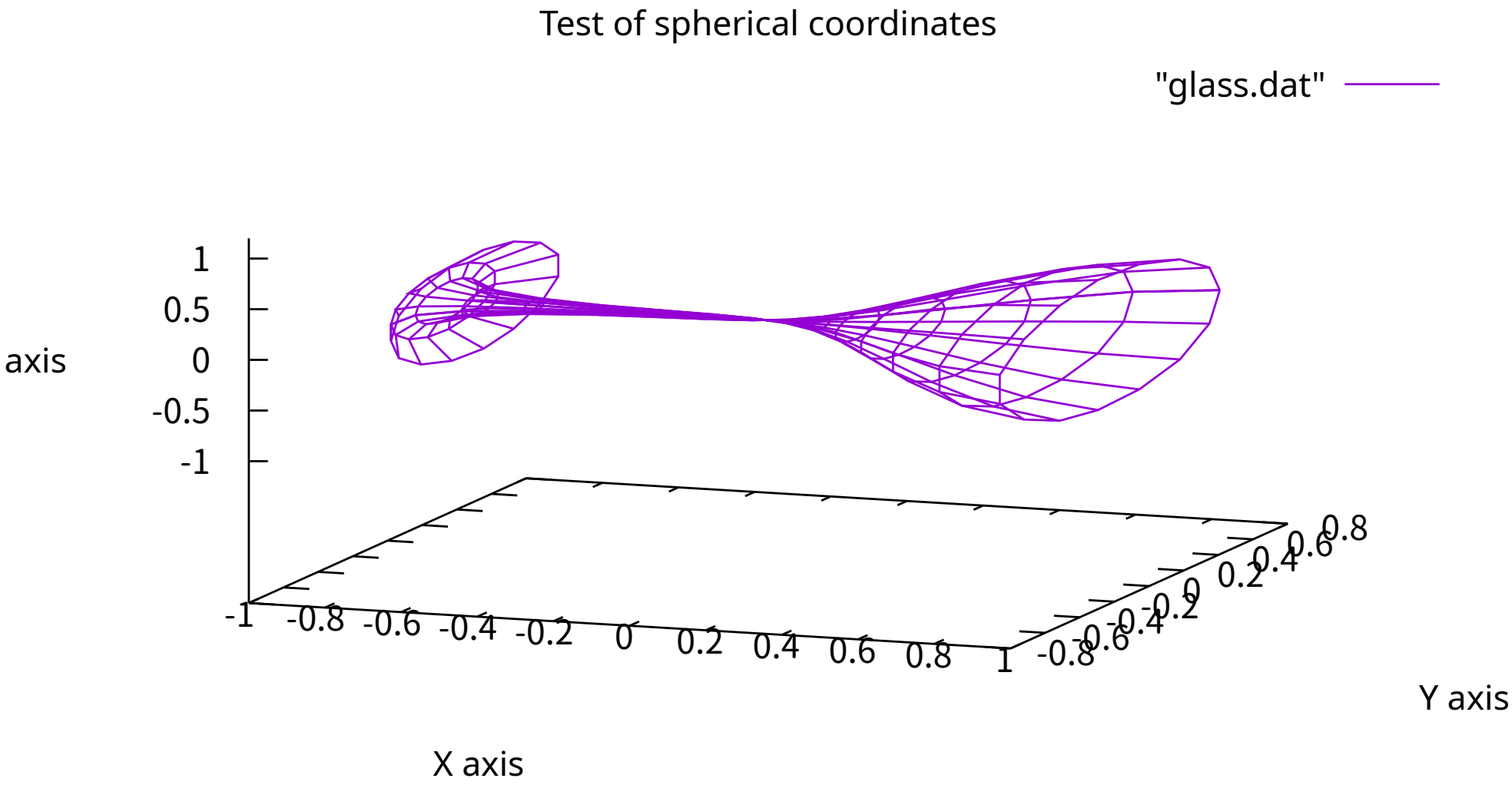

Data grid plotting
"glass.dat"
1.5
1
0.5
0
-0.5
-1
axis
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
X axis
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Y axis

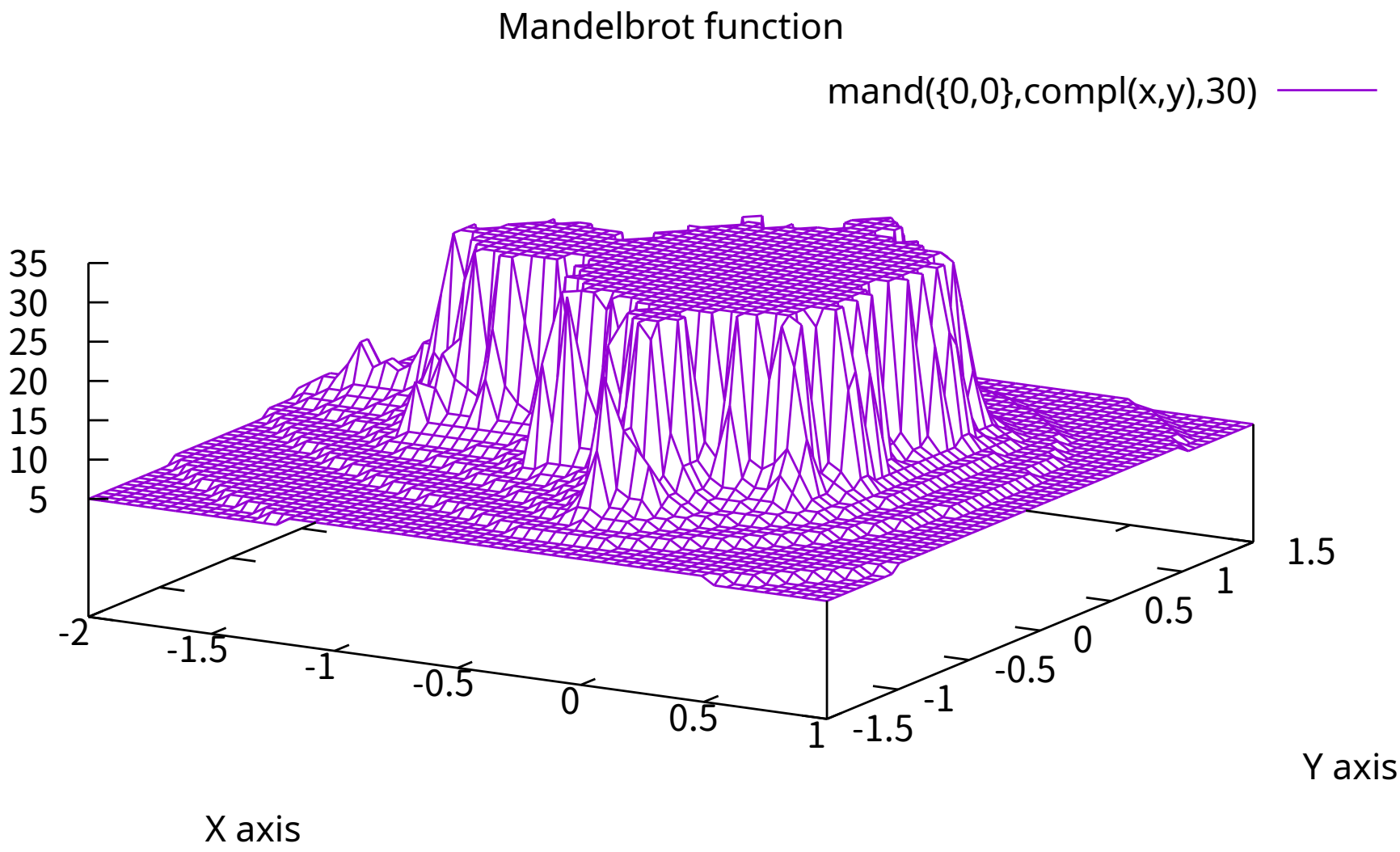

Data grid plotting
"glass.dat" using 3:2:1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
axis
-1
-0.5
0
0.5
1
1.5
X axis
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Y axis

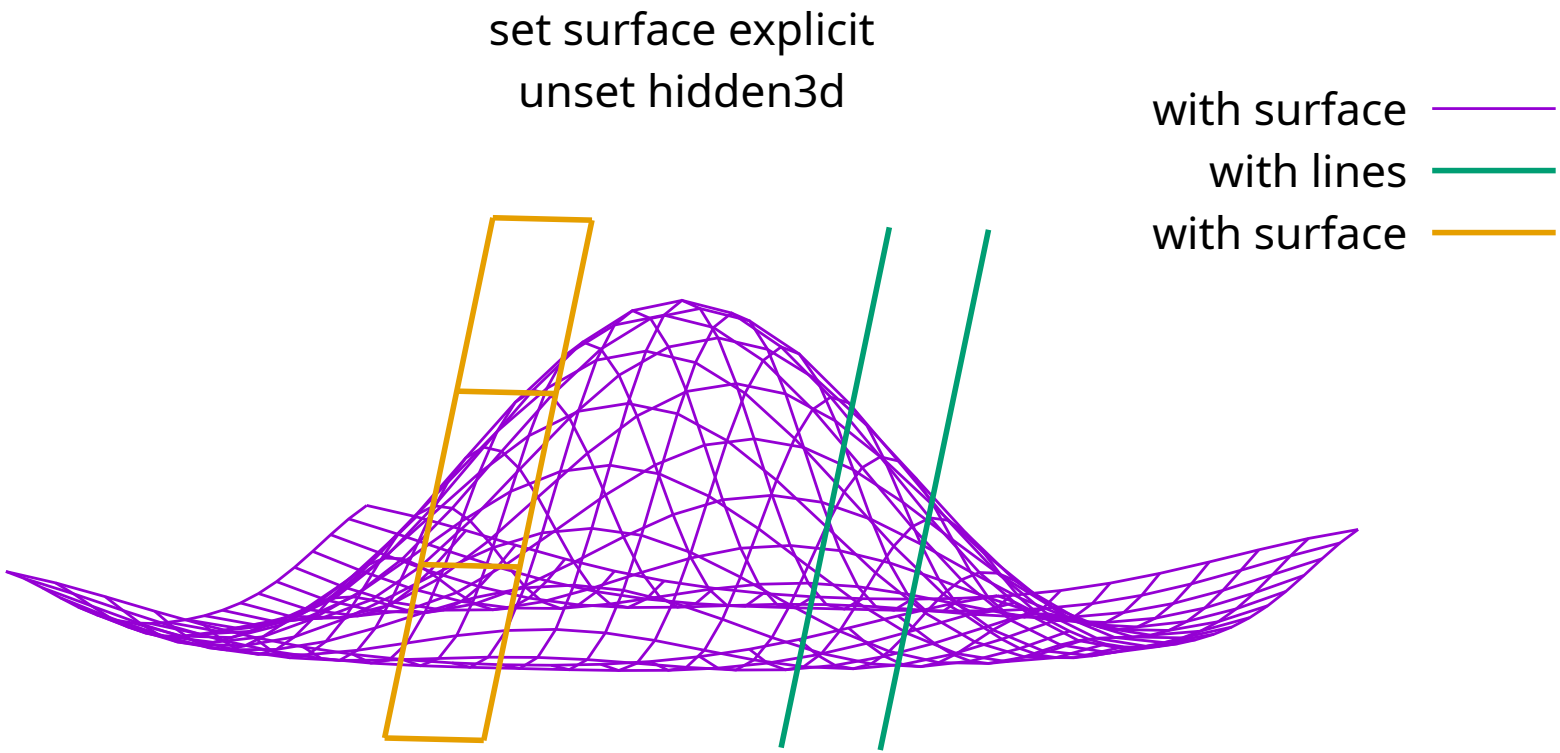

Data grid plotting
"glass.dat" using 1
"glass.dat" using 2
"glass.dat" using 3
1
0.5
0
-0.5
-1
axis
0
2
4
6
8
10
12
14
16
0
2
4
6
8
10
12
14
16
group inde
line index within group


splot of part of a data file
'glass.dat' every 2::0::12
1
0.5
0
-0.5
-1
axis
-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

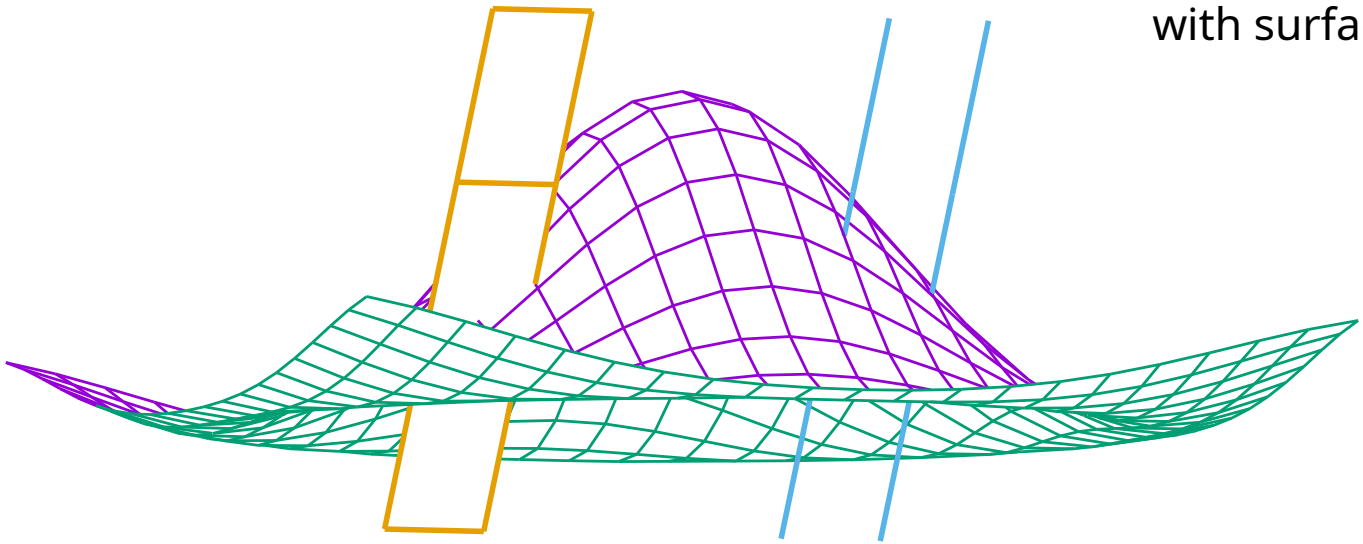

splot with "set pm3d" (implemented with some terminals)
'glass.dat' every 2::0::12
1.5
1
0.5
0
-0.5
-1
1
0.5
0
-0.5
-1
axis
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
X axis
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Y axis

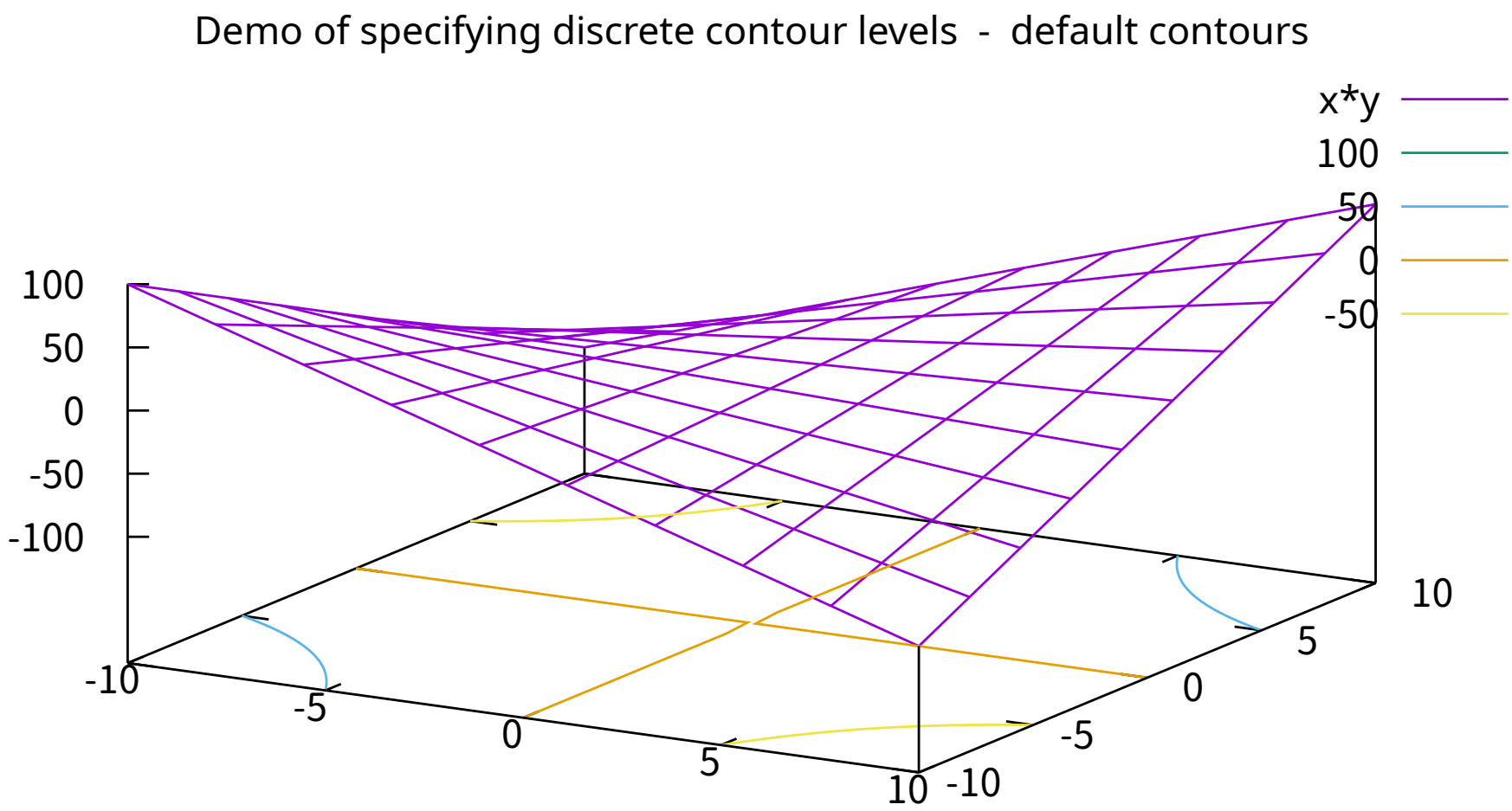

Test of spherical coordinates
"glass.dat"
1
0.5
0
-0.5
-1
axis
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
X axis
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Y axis

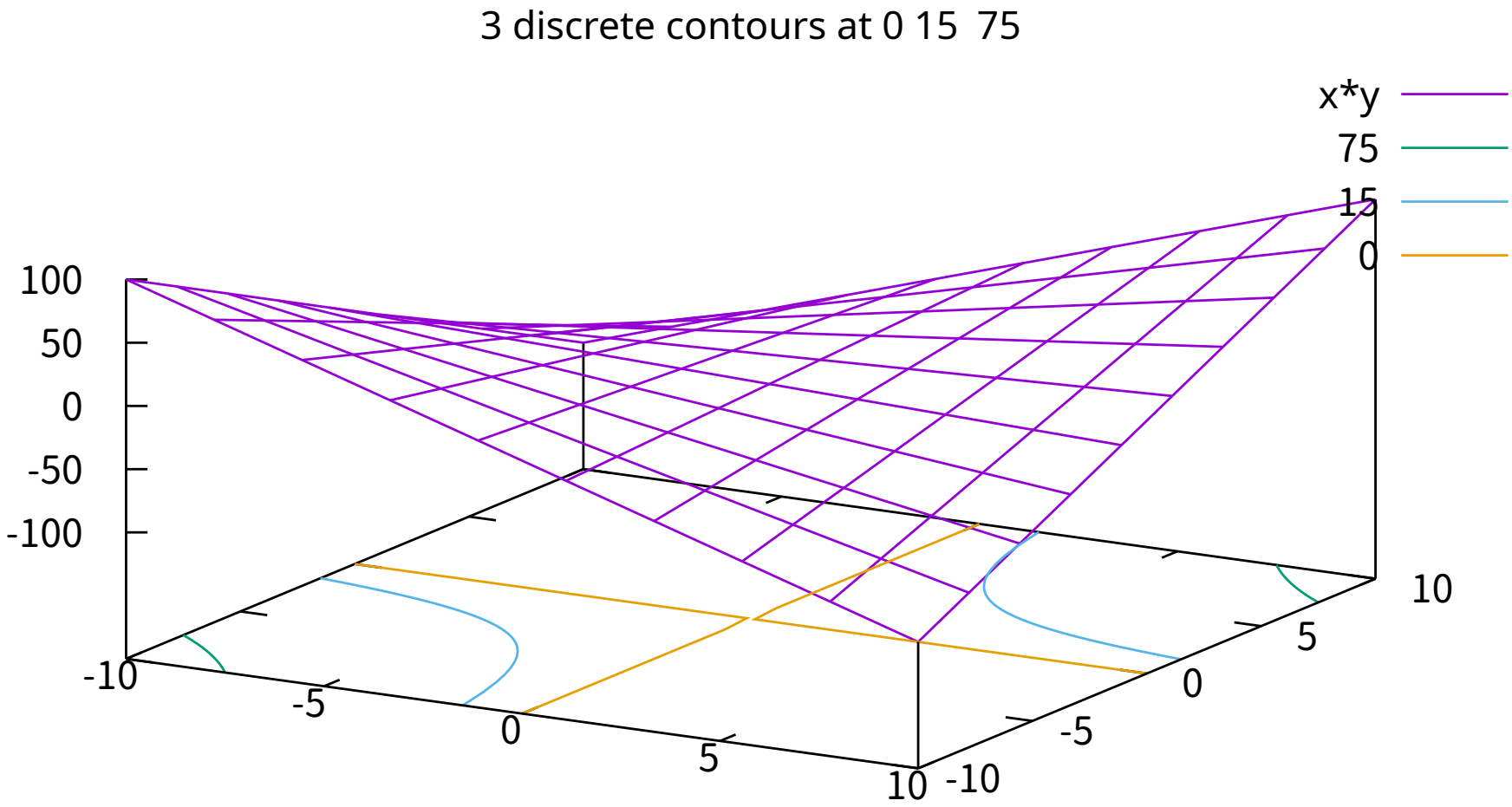

Mandelbrot function
mand({0,0},compl(x,y),30)
35
30
25
20
15
10
5
-2
-1.5
-1
-0.5
0
0.5
1
-1.5
-1
-0.5
0
0.5
1
1.5
X axis
Y axis


set surface explicit
unset hidden3d
with surface
with lines
with surface

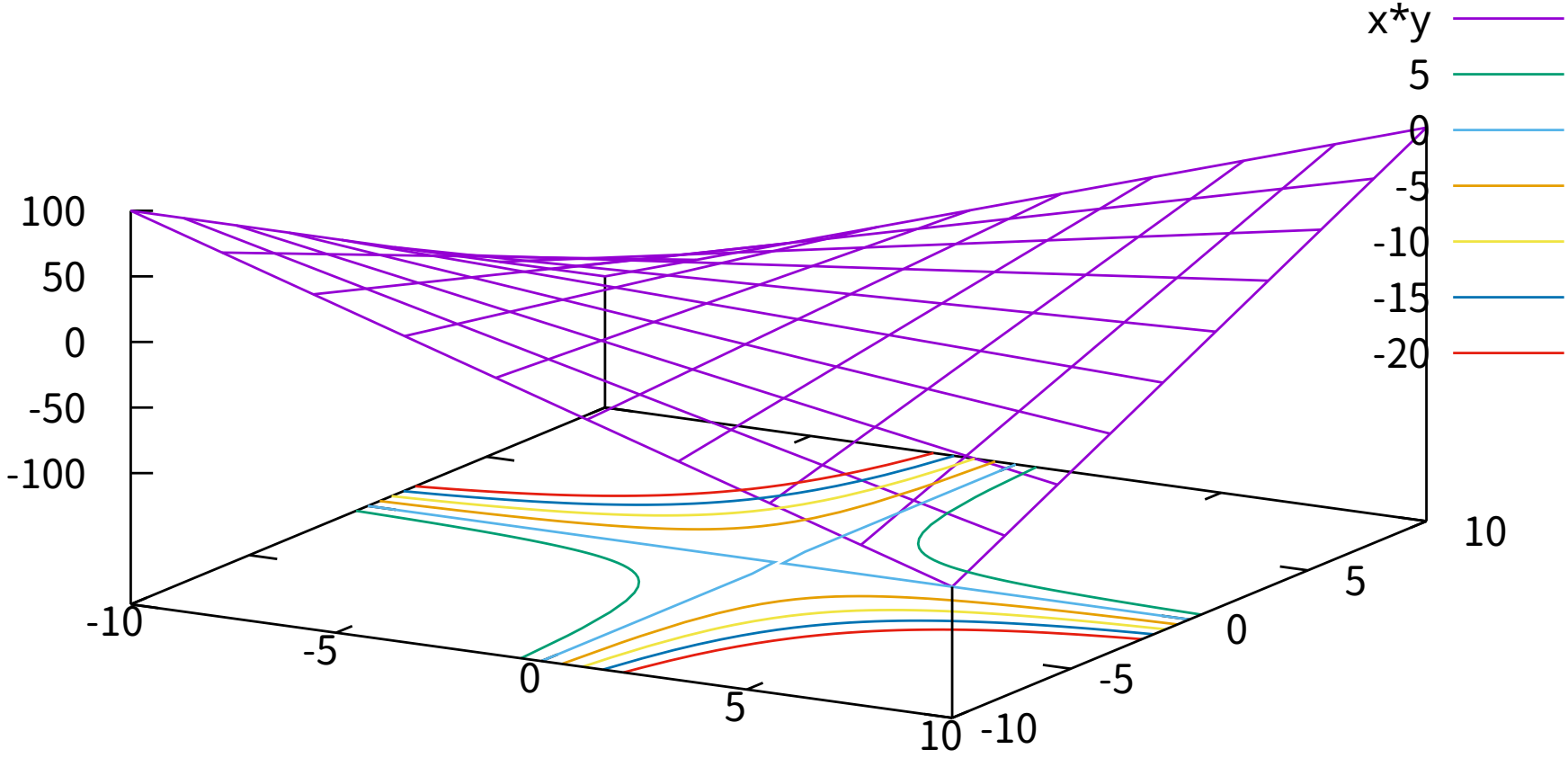
set surface explicit
set hidden3d



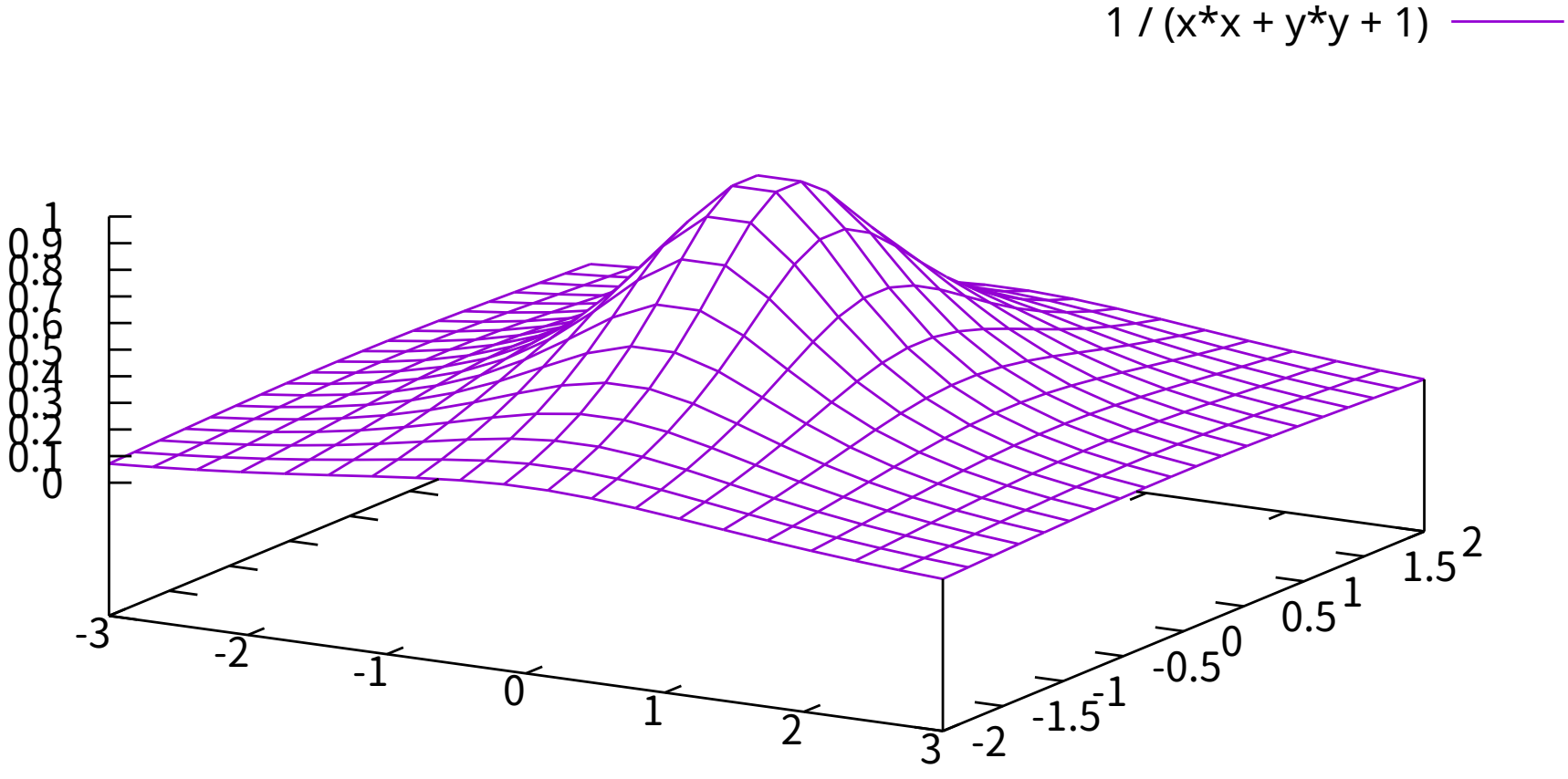

Demo of specifying discrete contour levels - default contours
x*y
100
50
0
-50
100
50
0
-50
-100
-10
-5
0
5
10
-10
-5
0
5
10

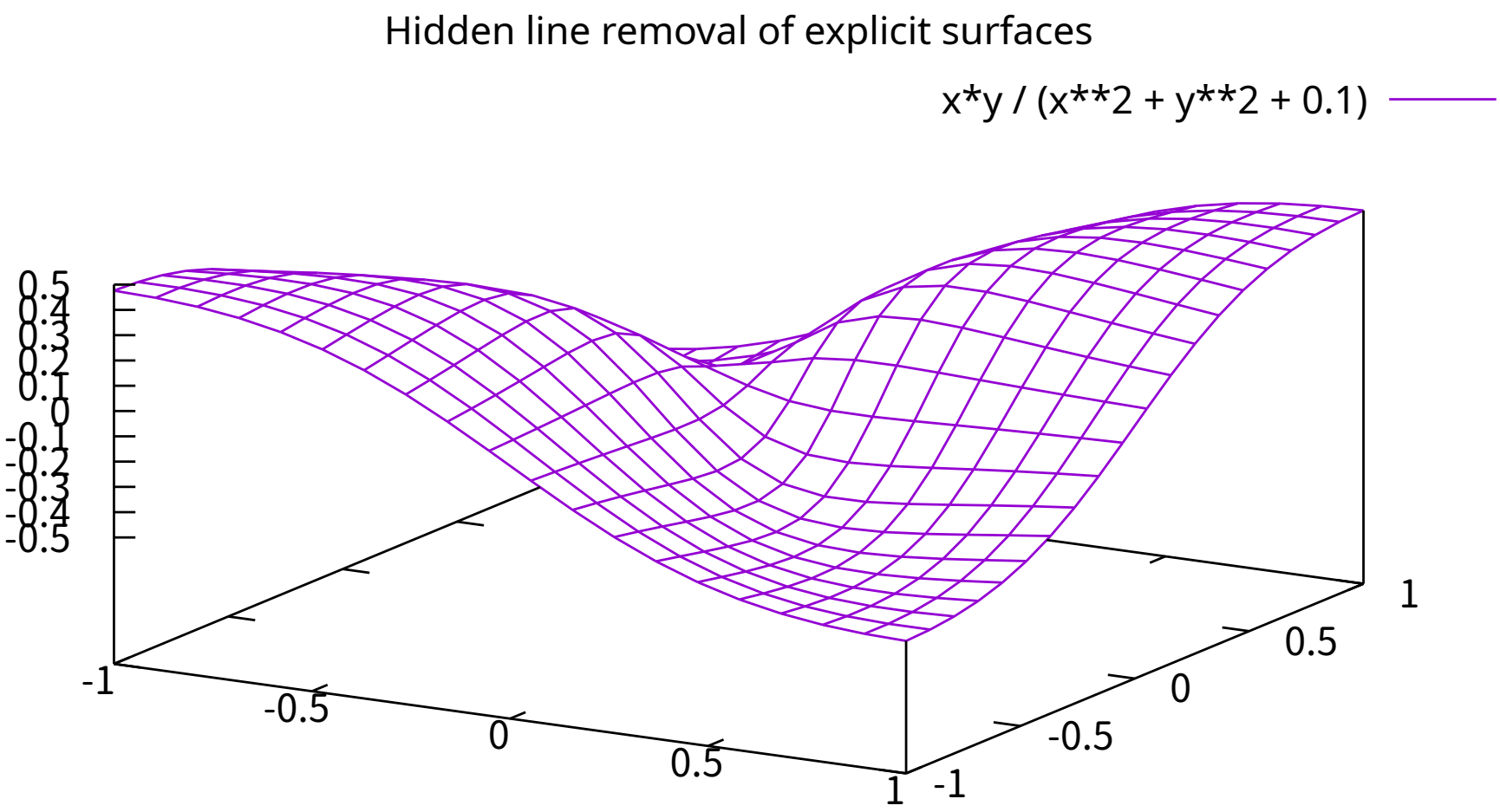

3 discrete contours at 0 15 75
x*y
75
15
0
100
50
0
-50
-100
-10
-5
0
5
10
-10
-5
0
5
10

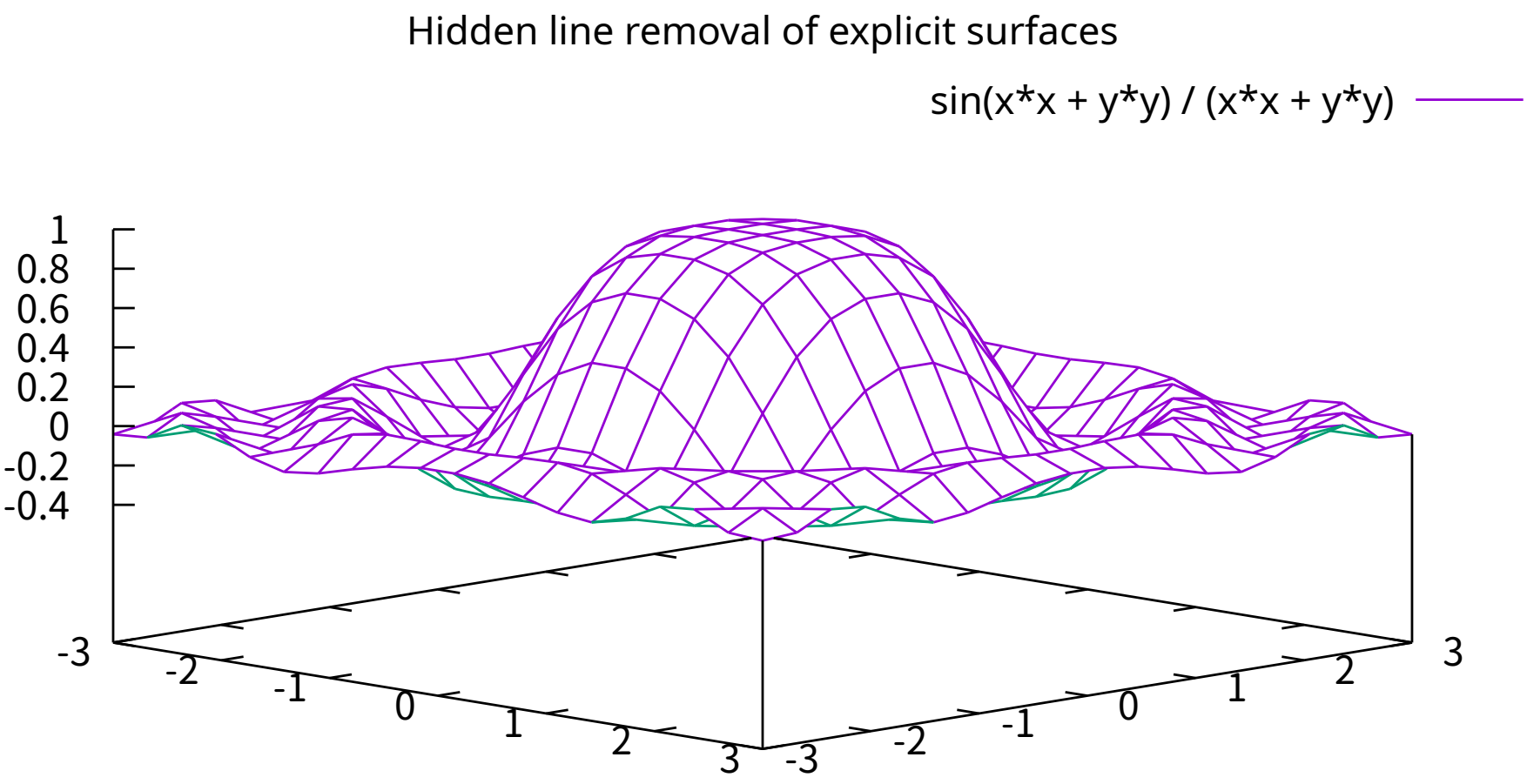
incremental contours starting at -20, stepping by 5

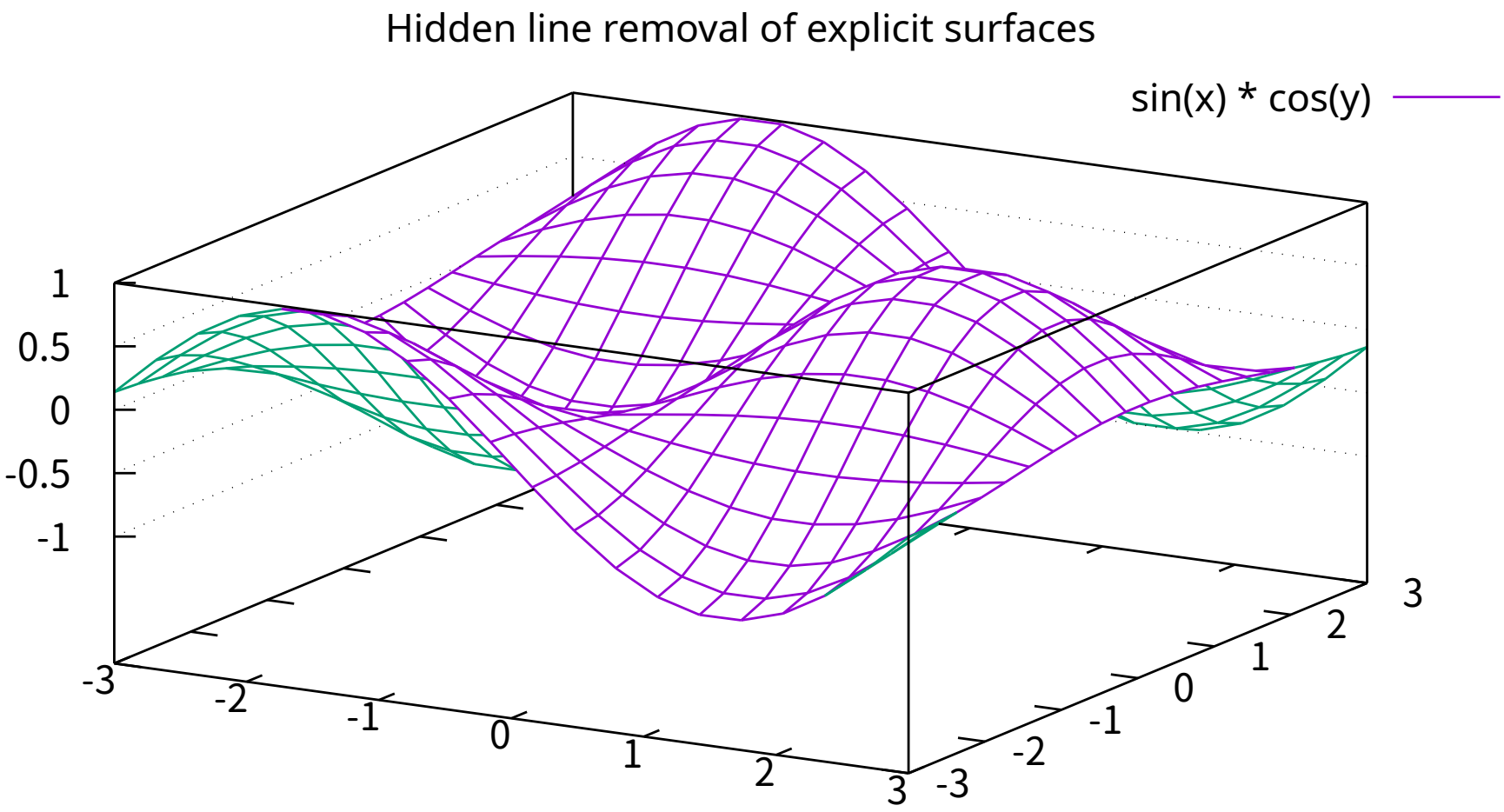


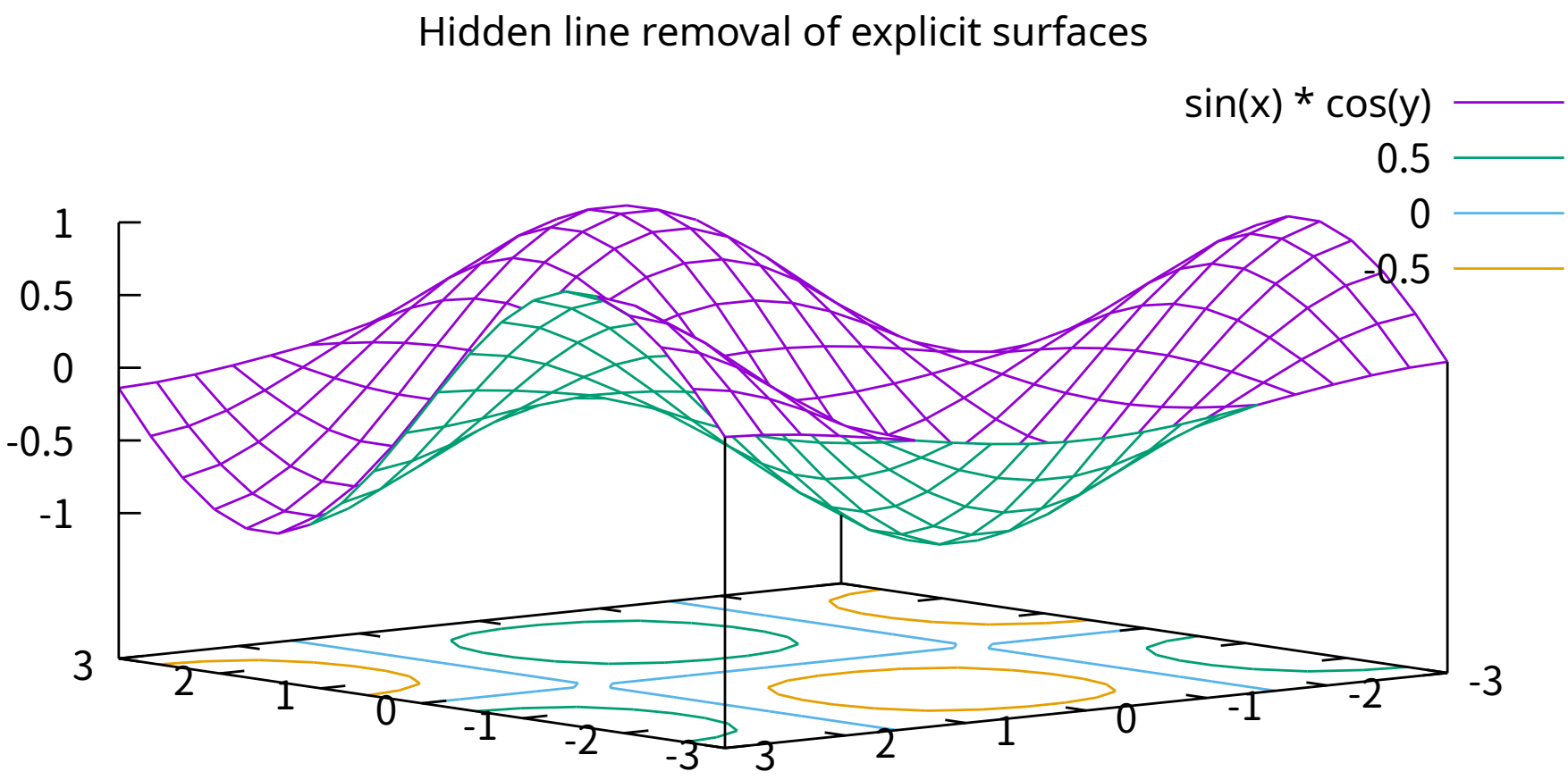
Hidden line removal of explicit surfaces

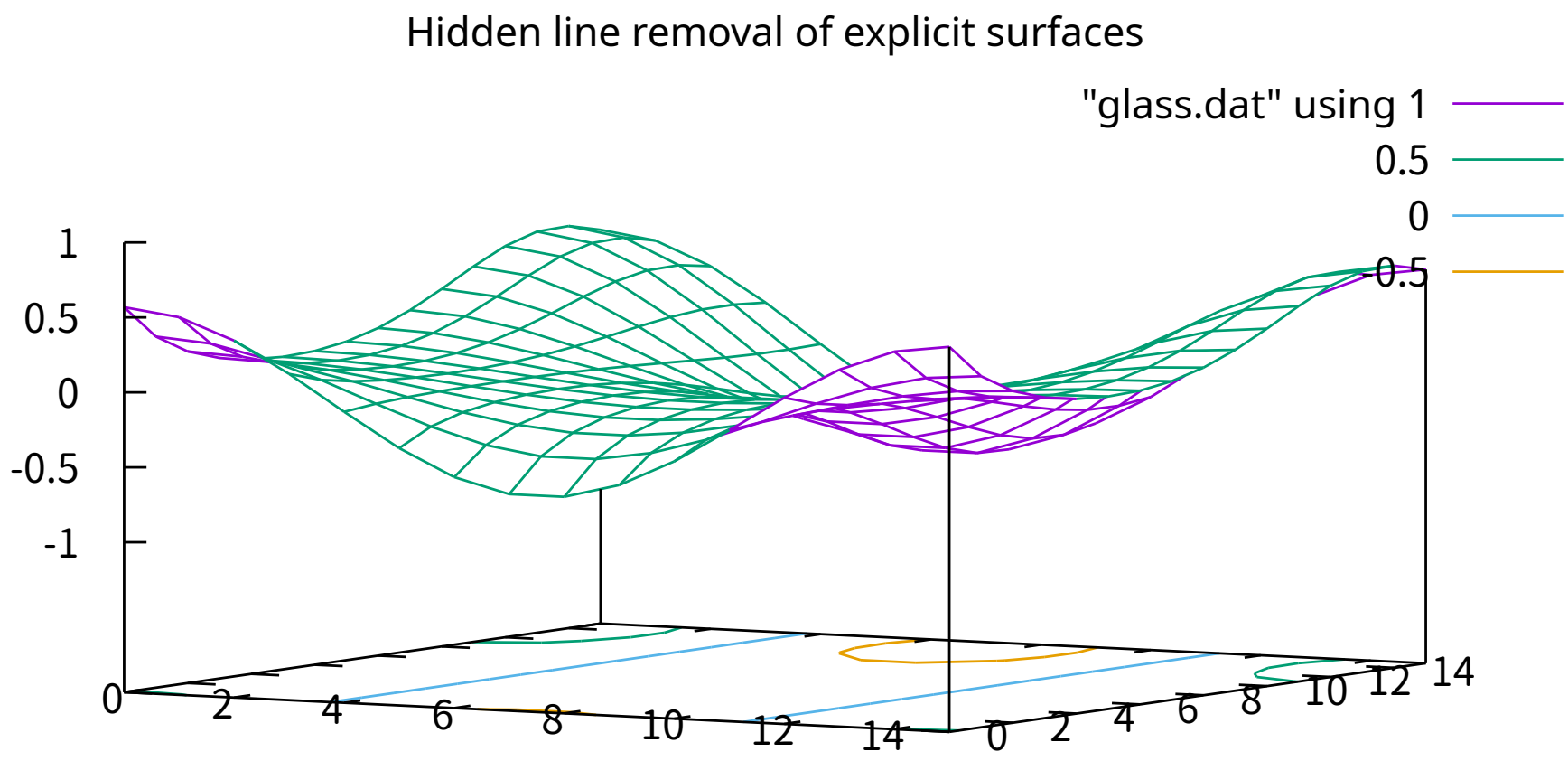


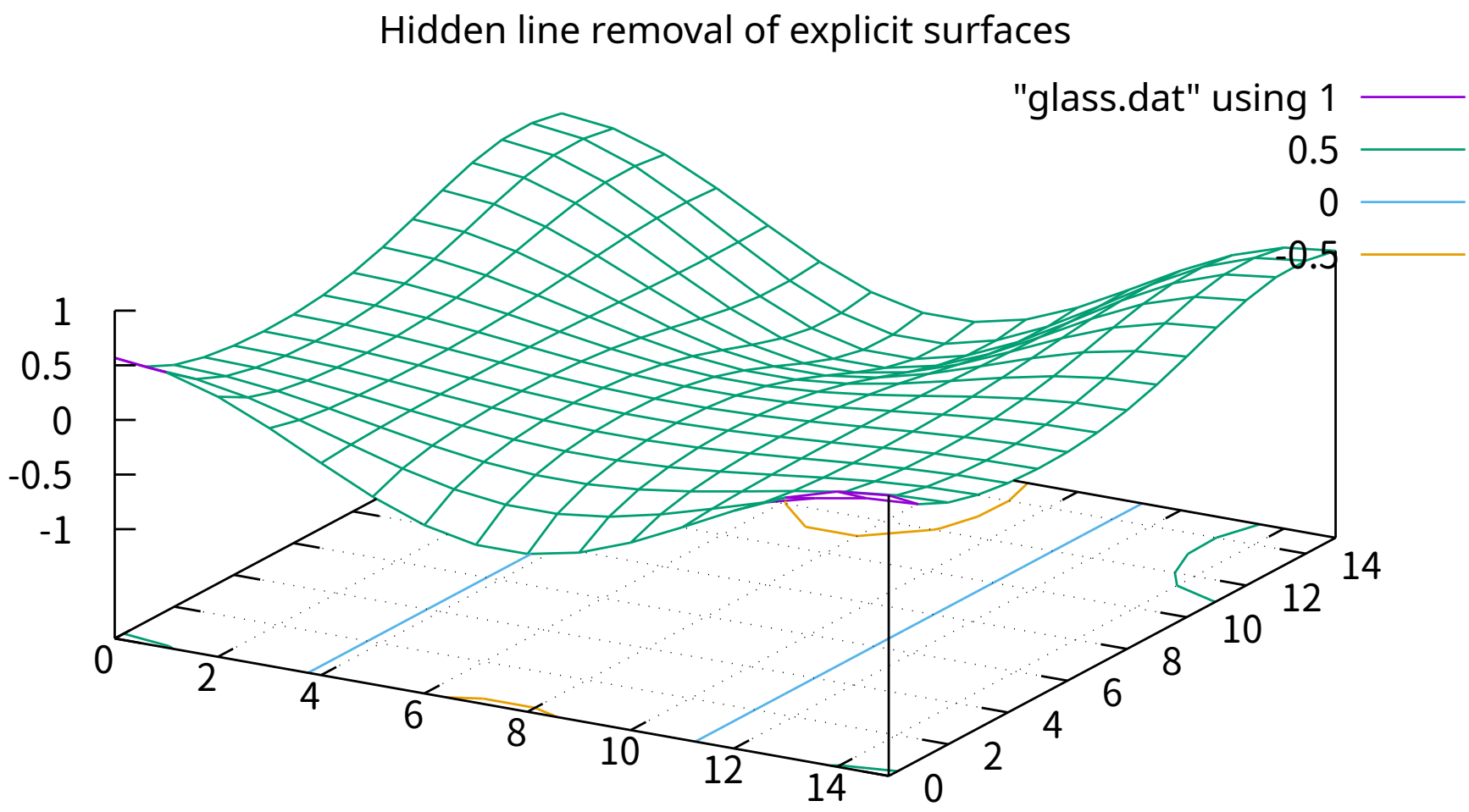

Hidden line removal of explicit surfaces
x*y / (x**2 + y**2 + 0.1)
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1


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

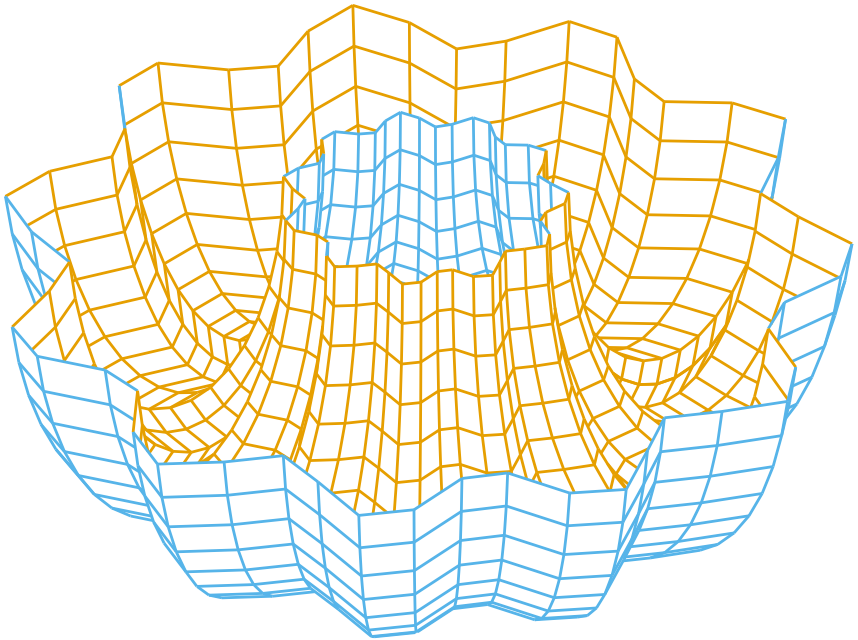

Hidden line removal of explicit surfaces
sin(x) * cos(y)
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3


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

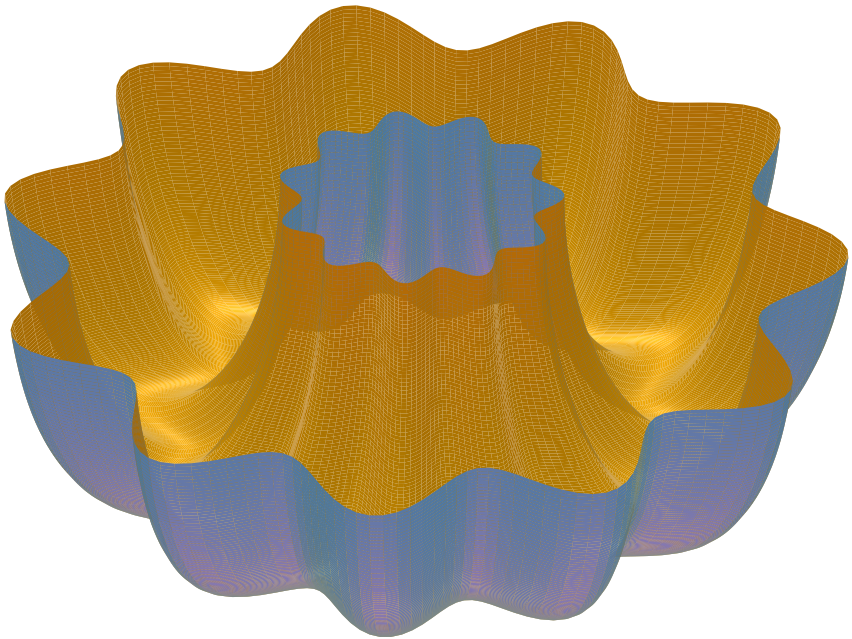

Hidden line removal of explicit surfaces
"glass.dat" using 1
0.5
0
-0.5
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14


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

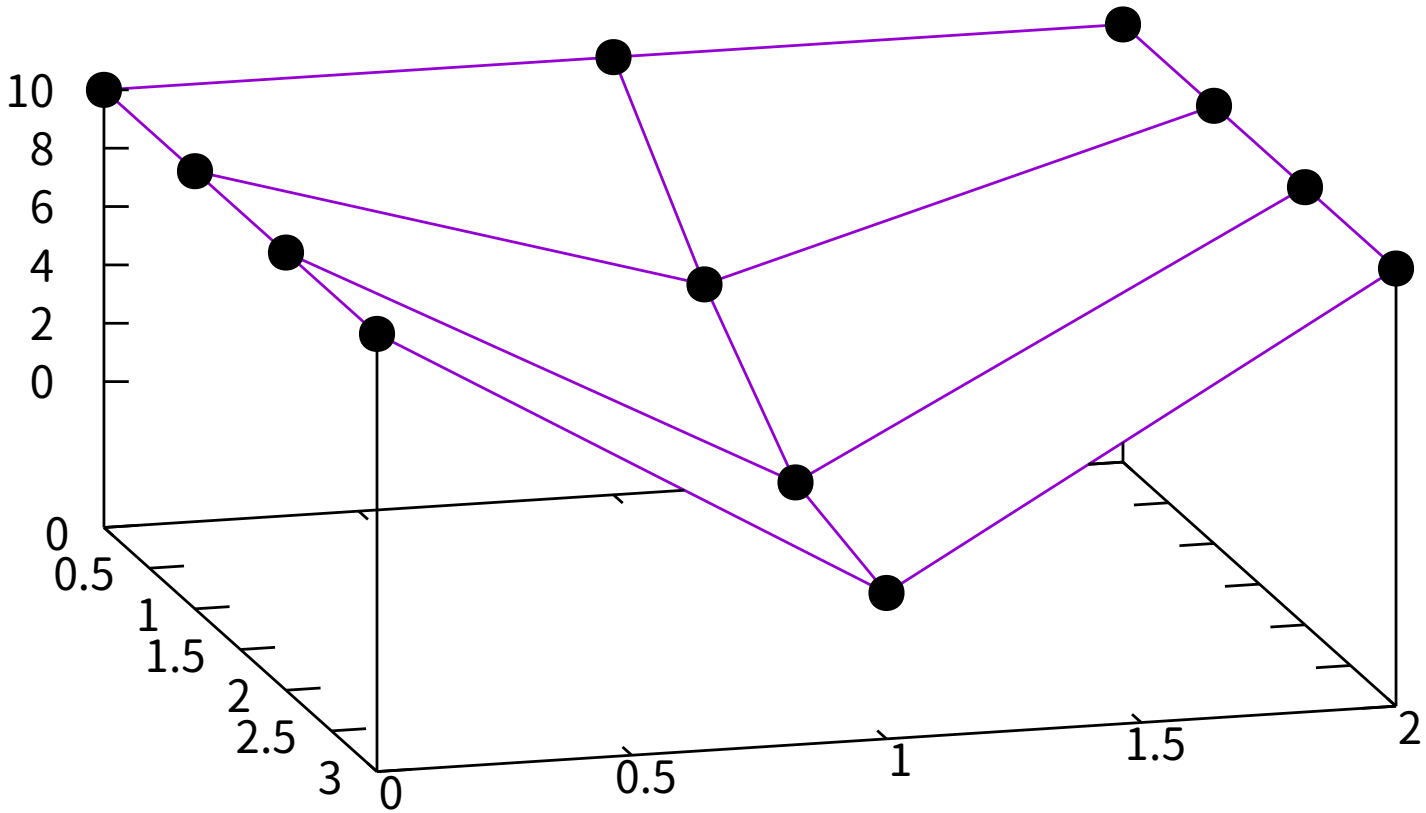
hidden3d 2-color surface

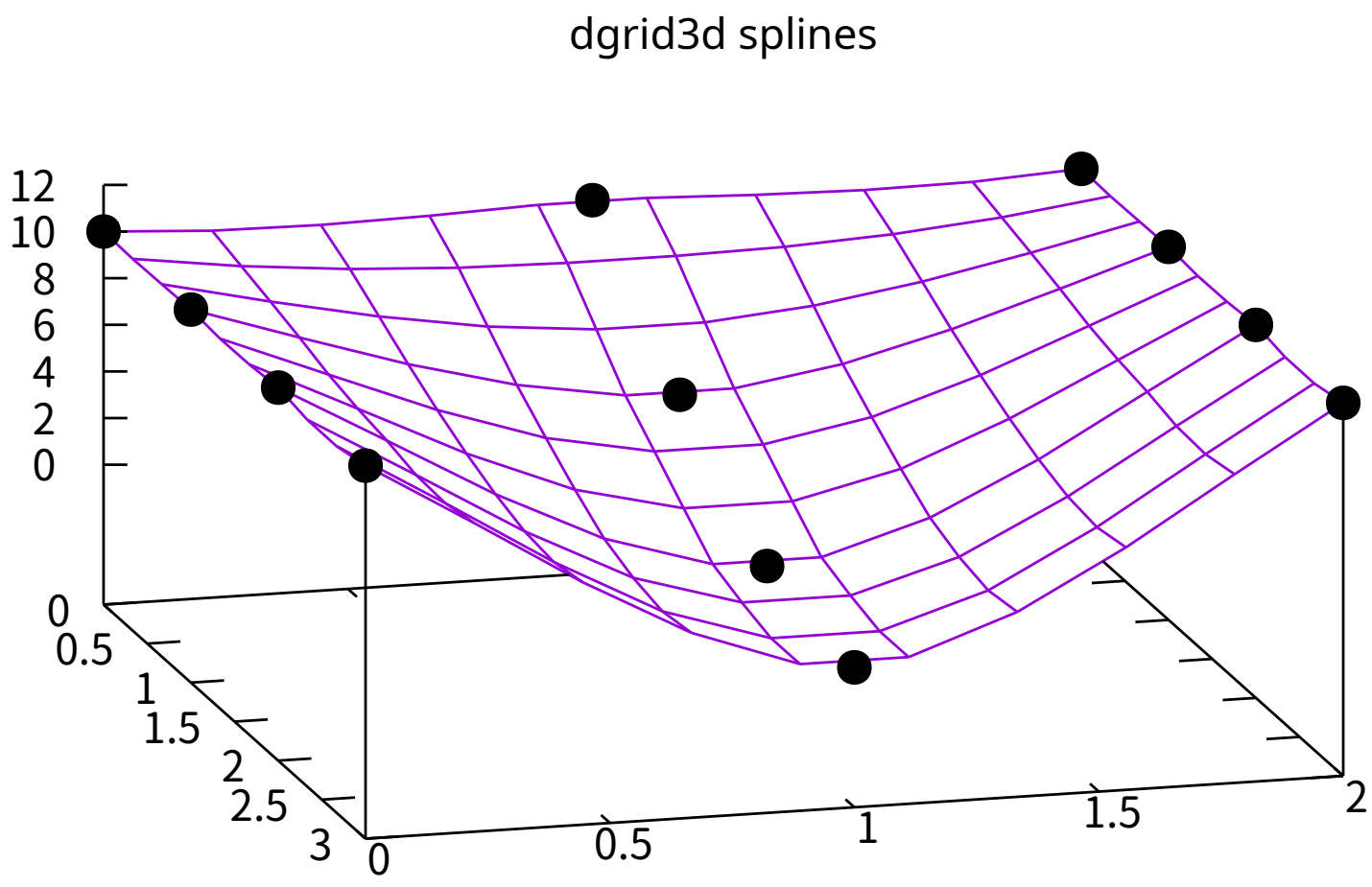


pm3d 2-color surface

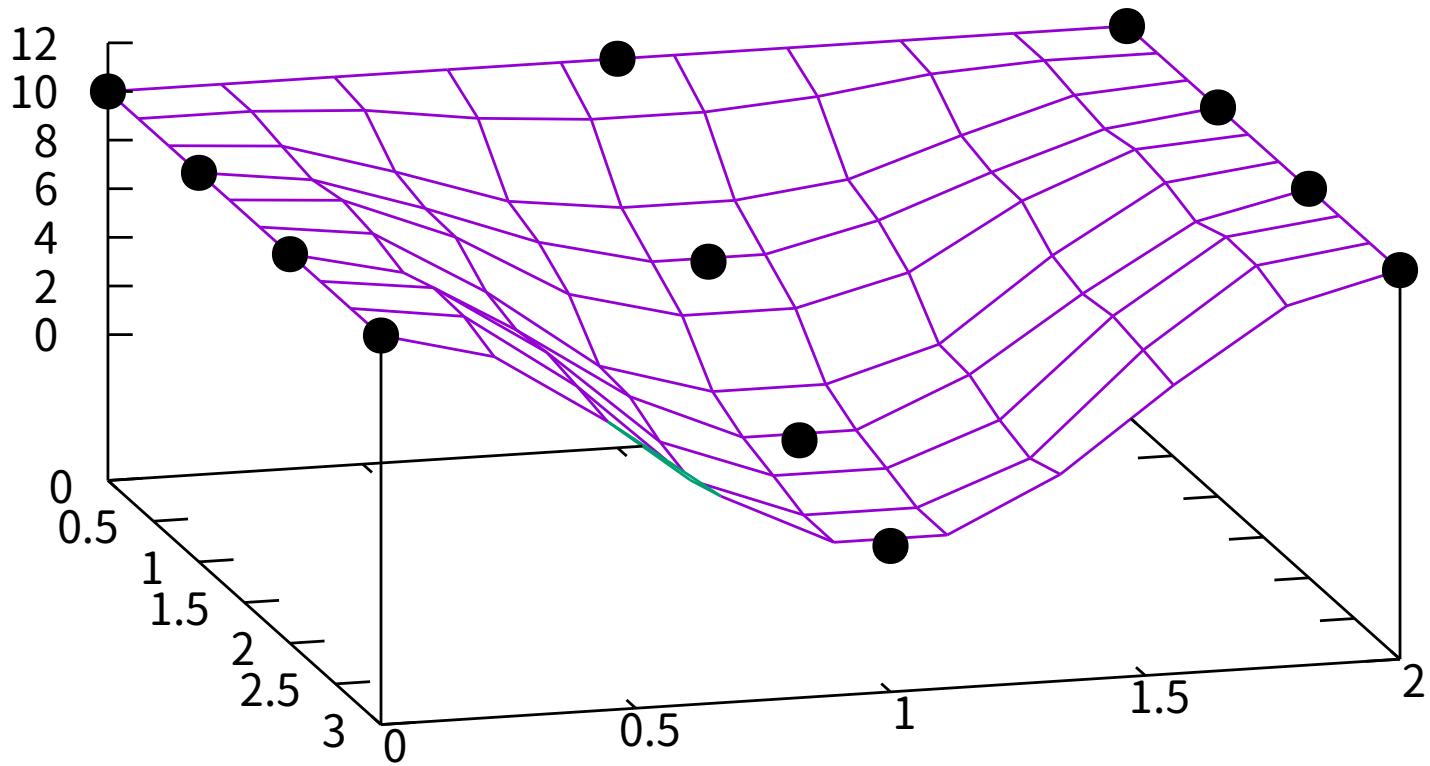


The Valley of the Gnu

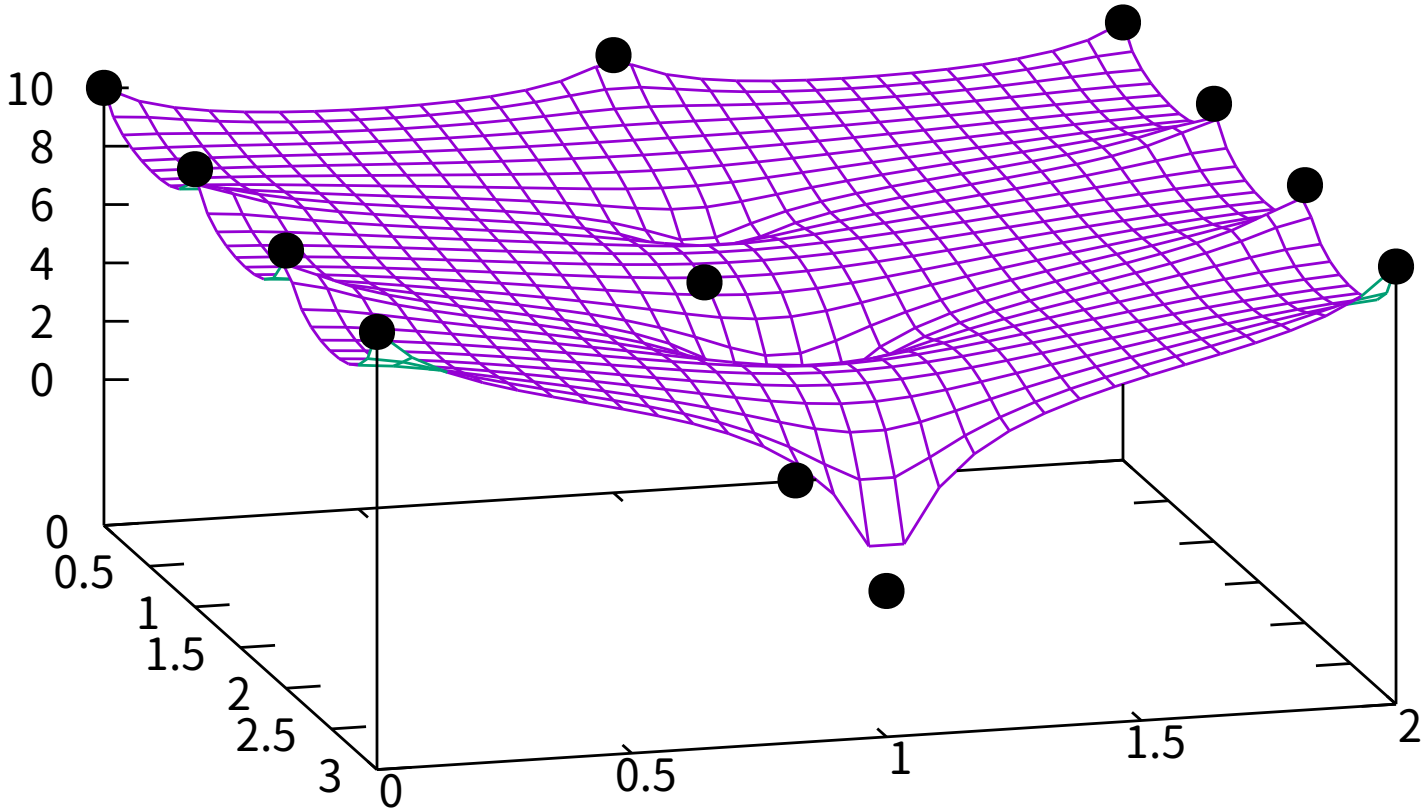



dgrid3d splines
12
10
8
6
4
2
0
0
0.5
1
1.5
2
2.5
3
0
0.5
1
1.5
2

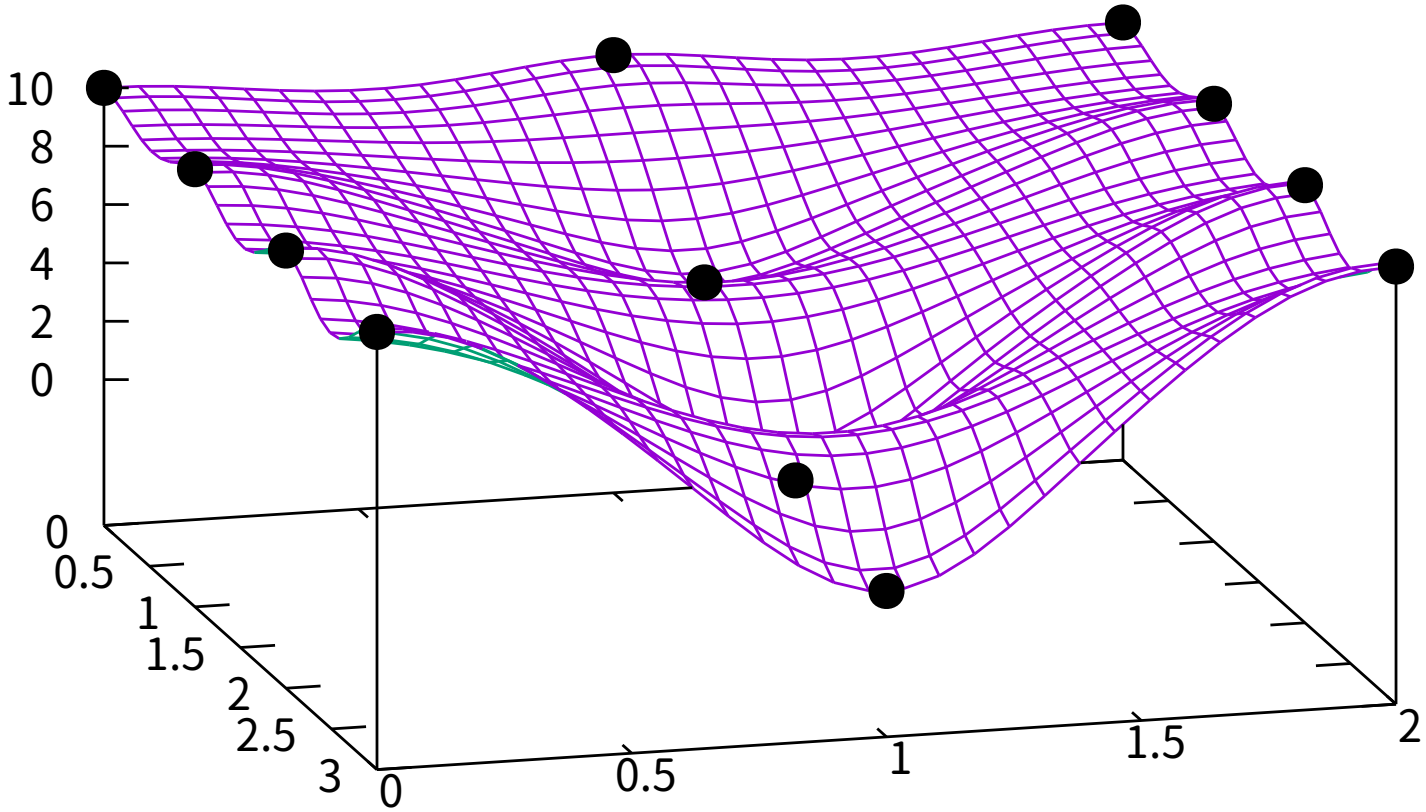
dgrid3d Hann function



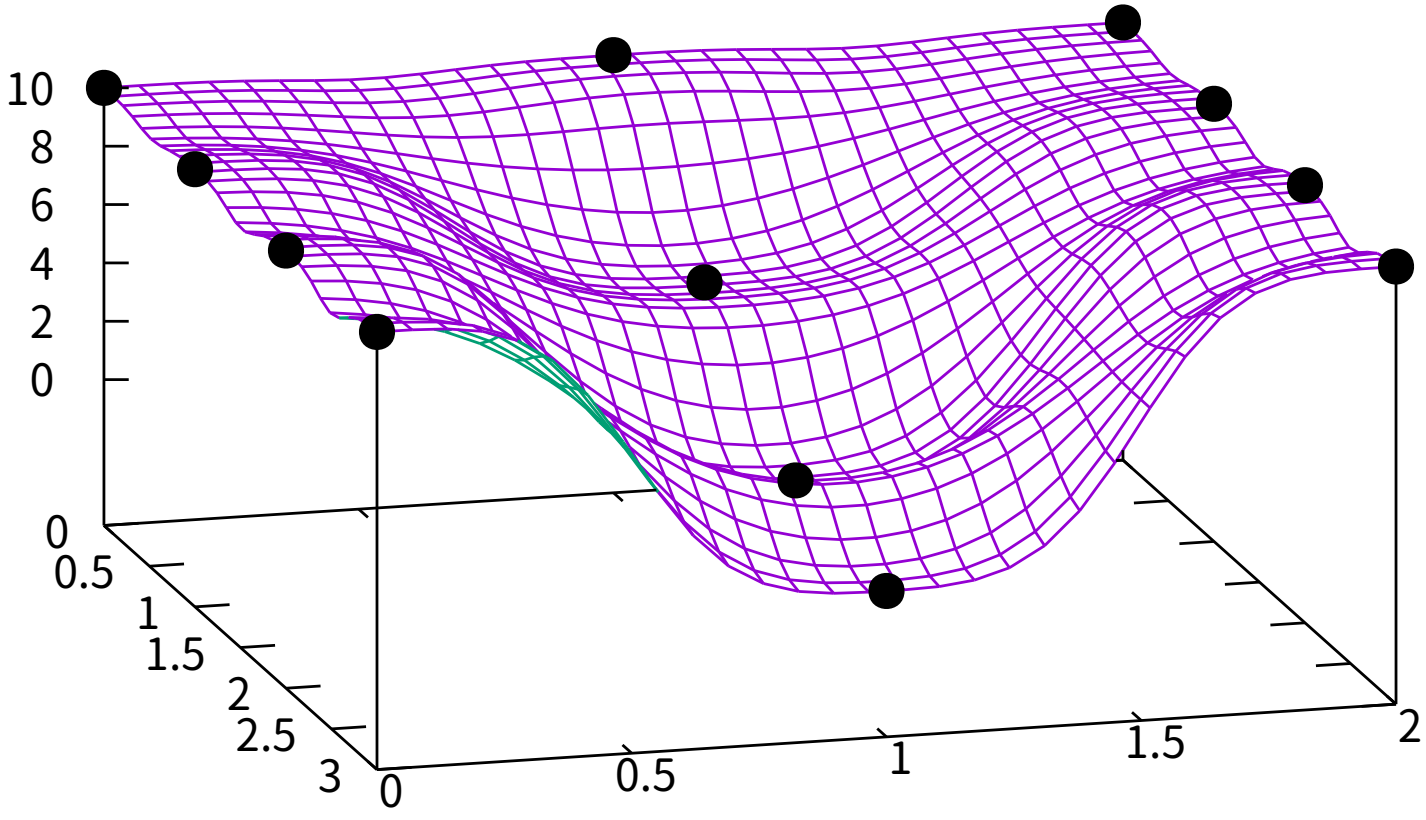
dgrid3d 30,30 qnorm 1



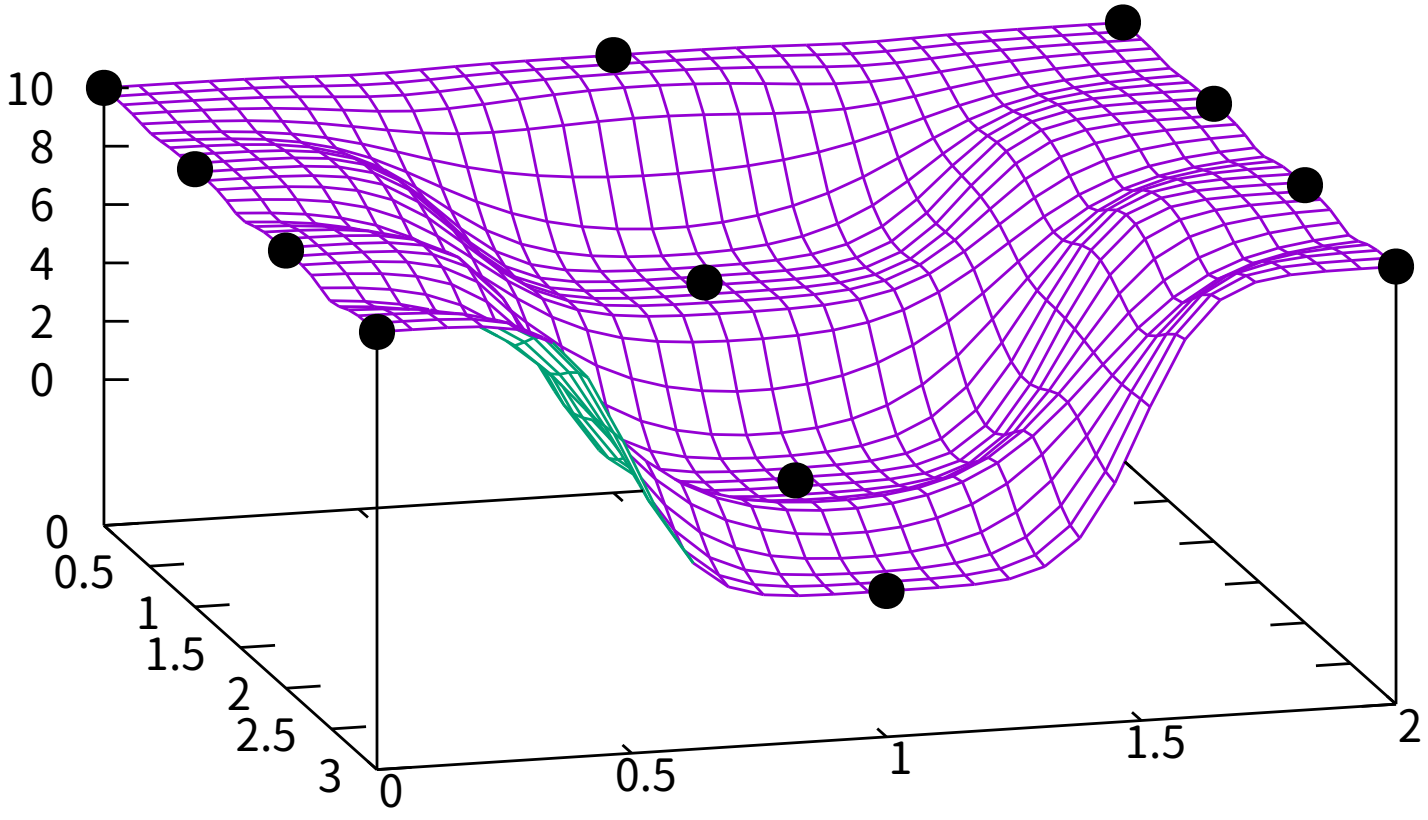
dgrid3d 30,30 qnorm 2



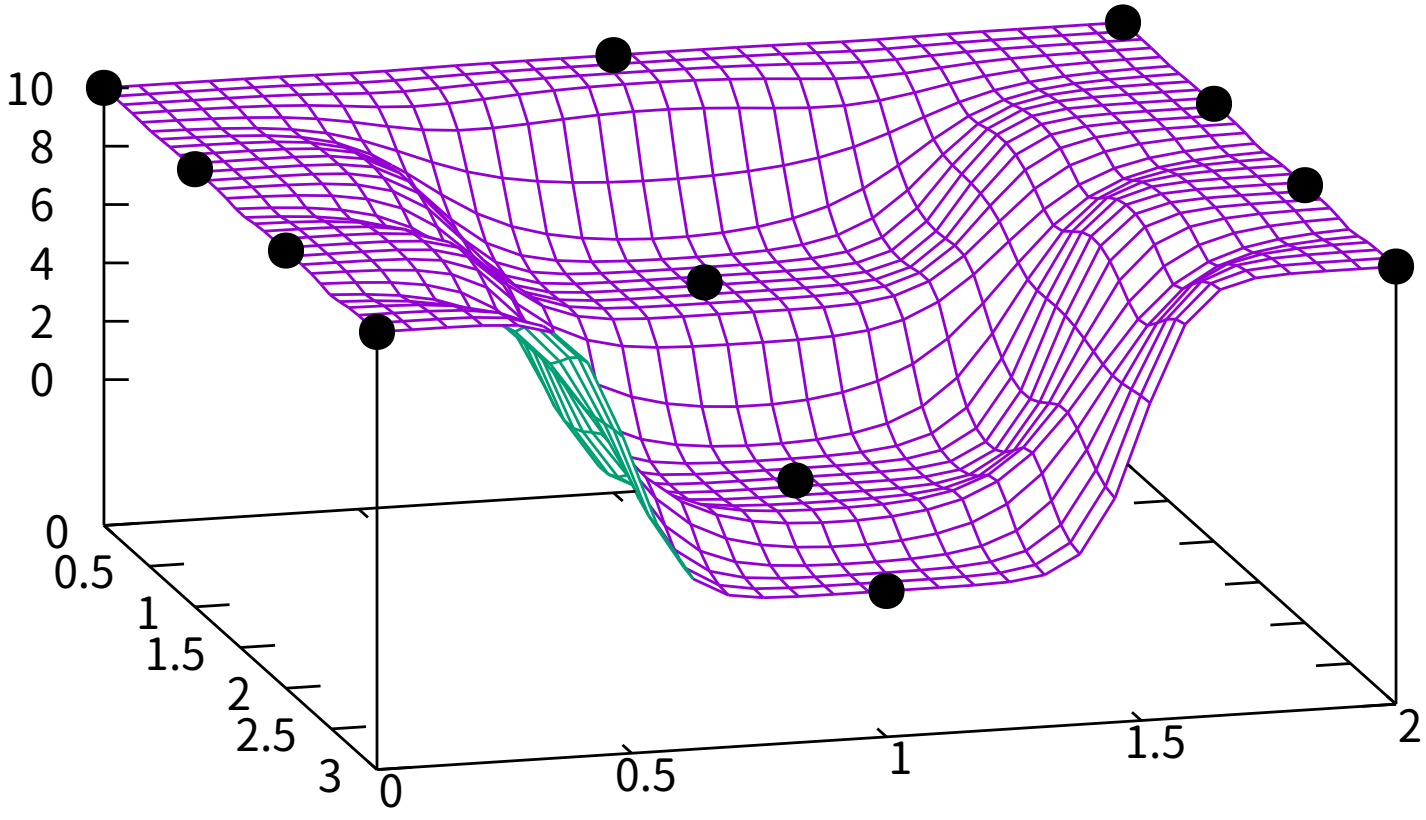
dgrid3d 30,30 qnorm 3



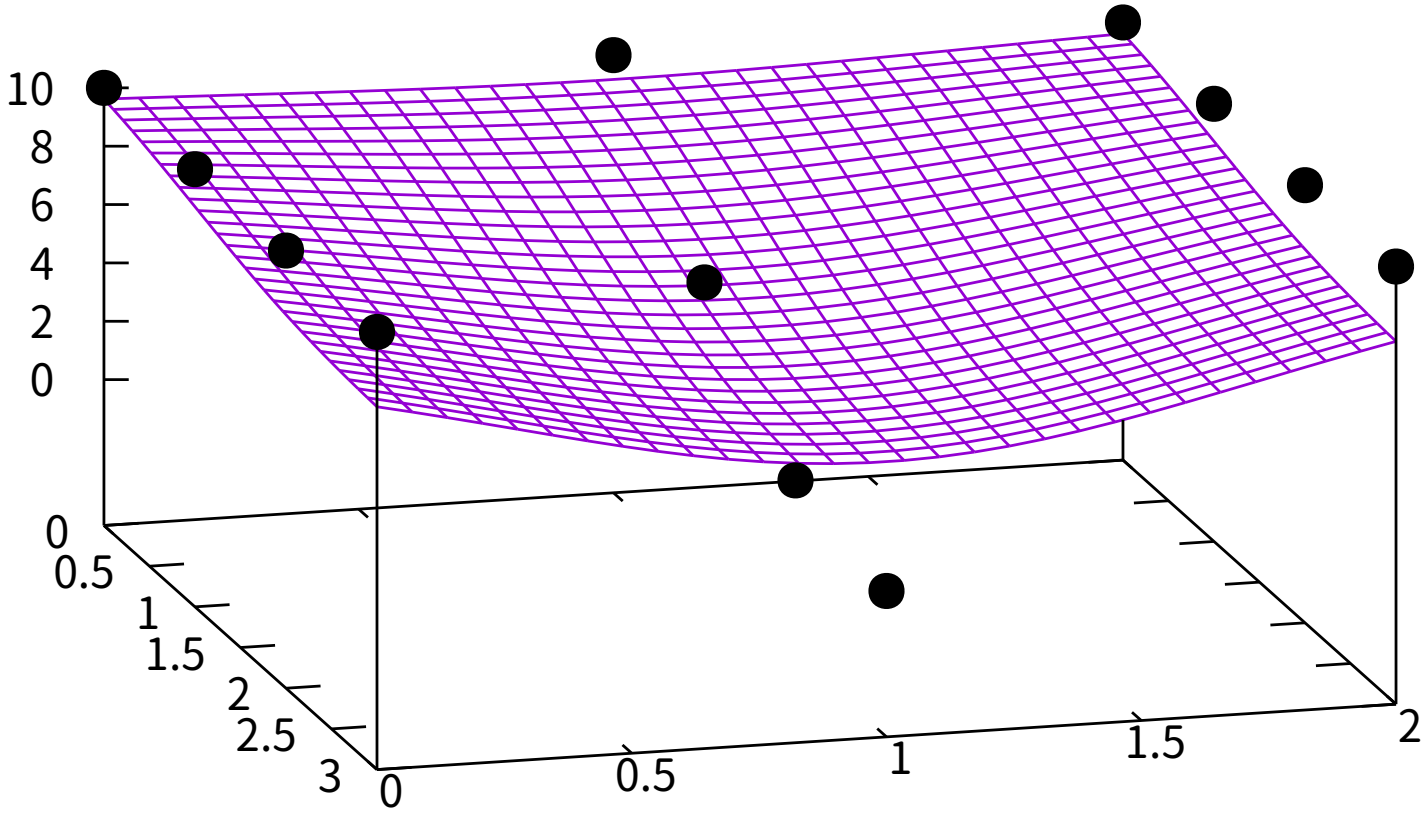
dgrid3d 30,30 qnorm 4



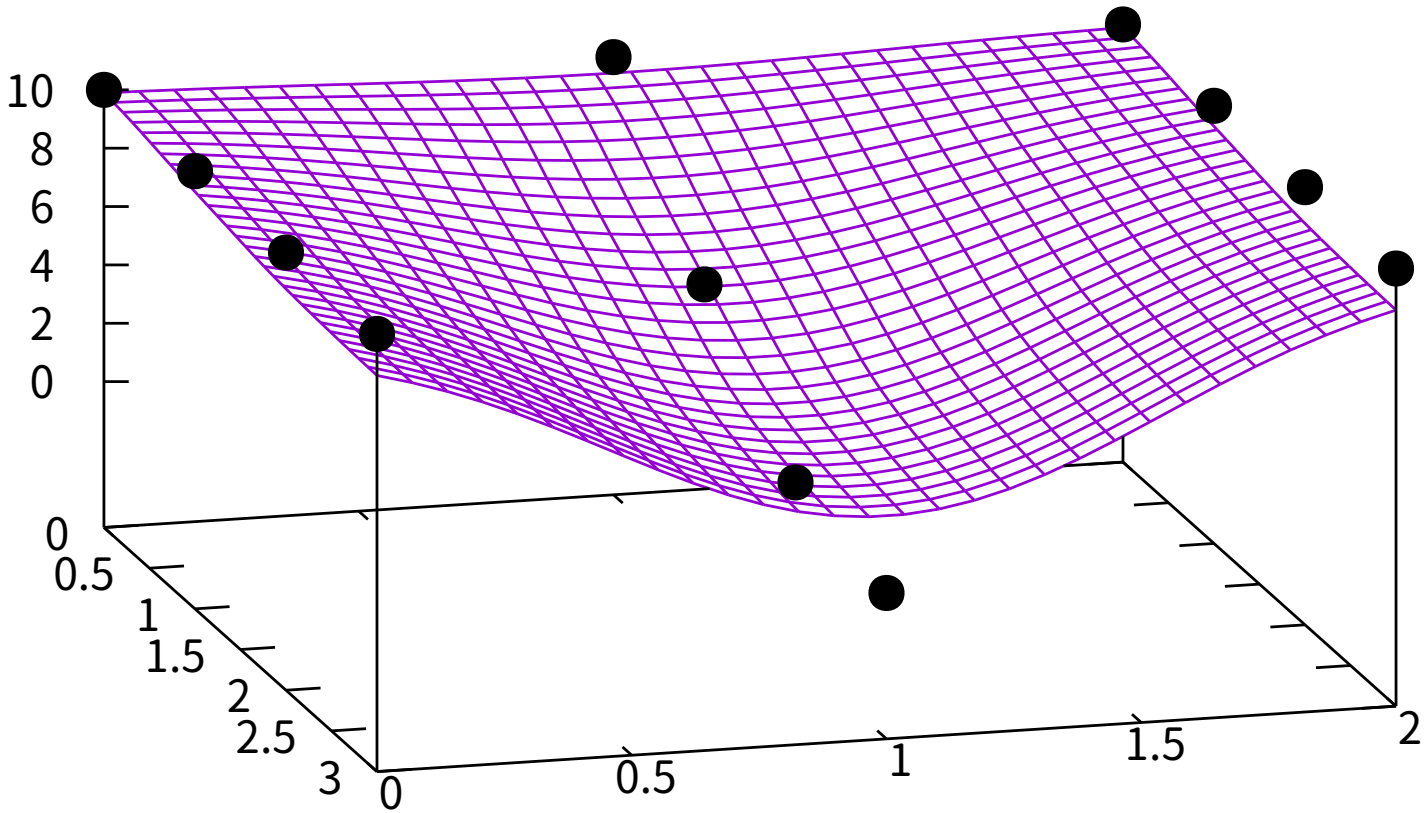
dgrid3d 30,30 qnorm 5



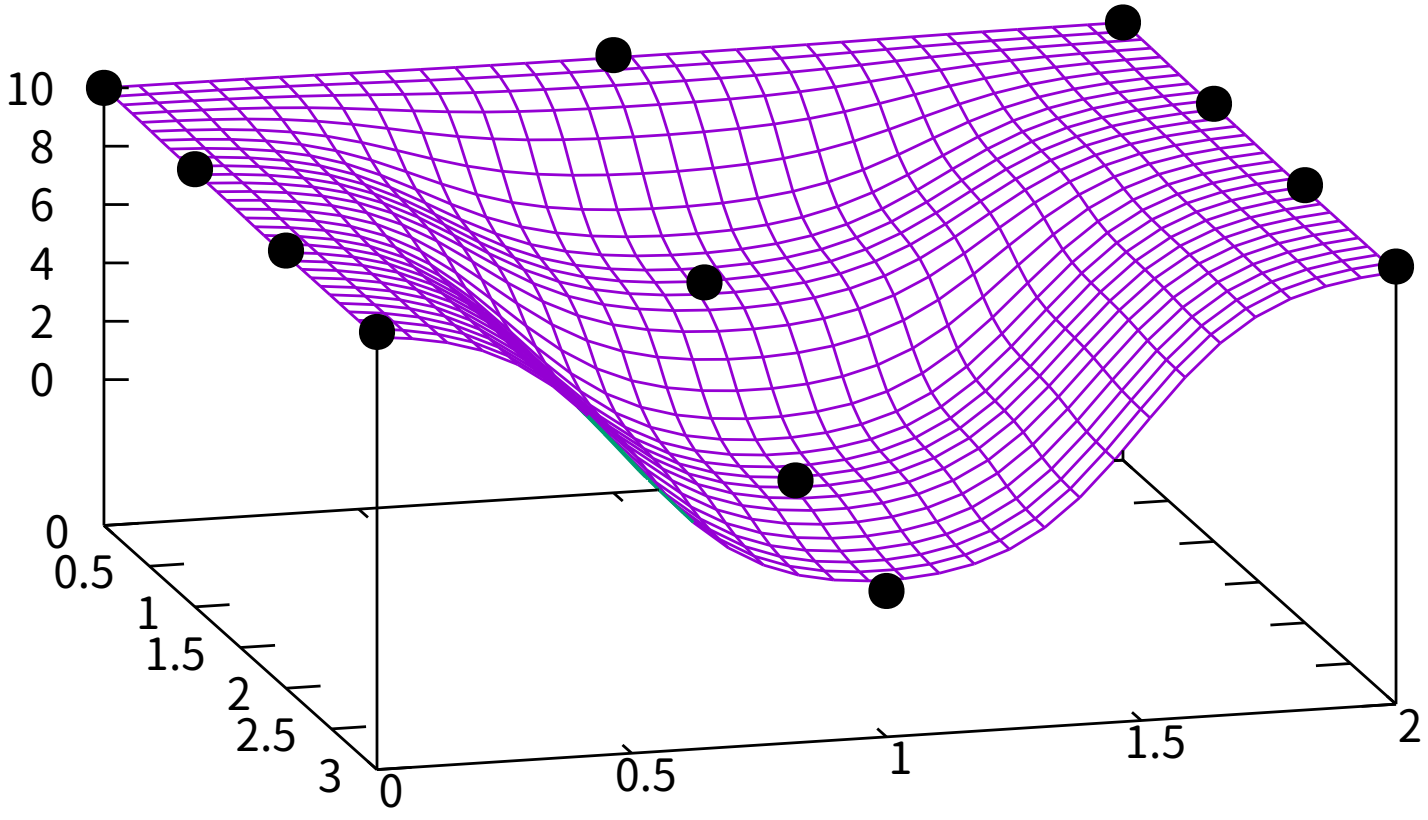
dgrid3d 30,30 gauss 1



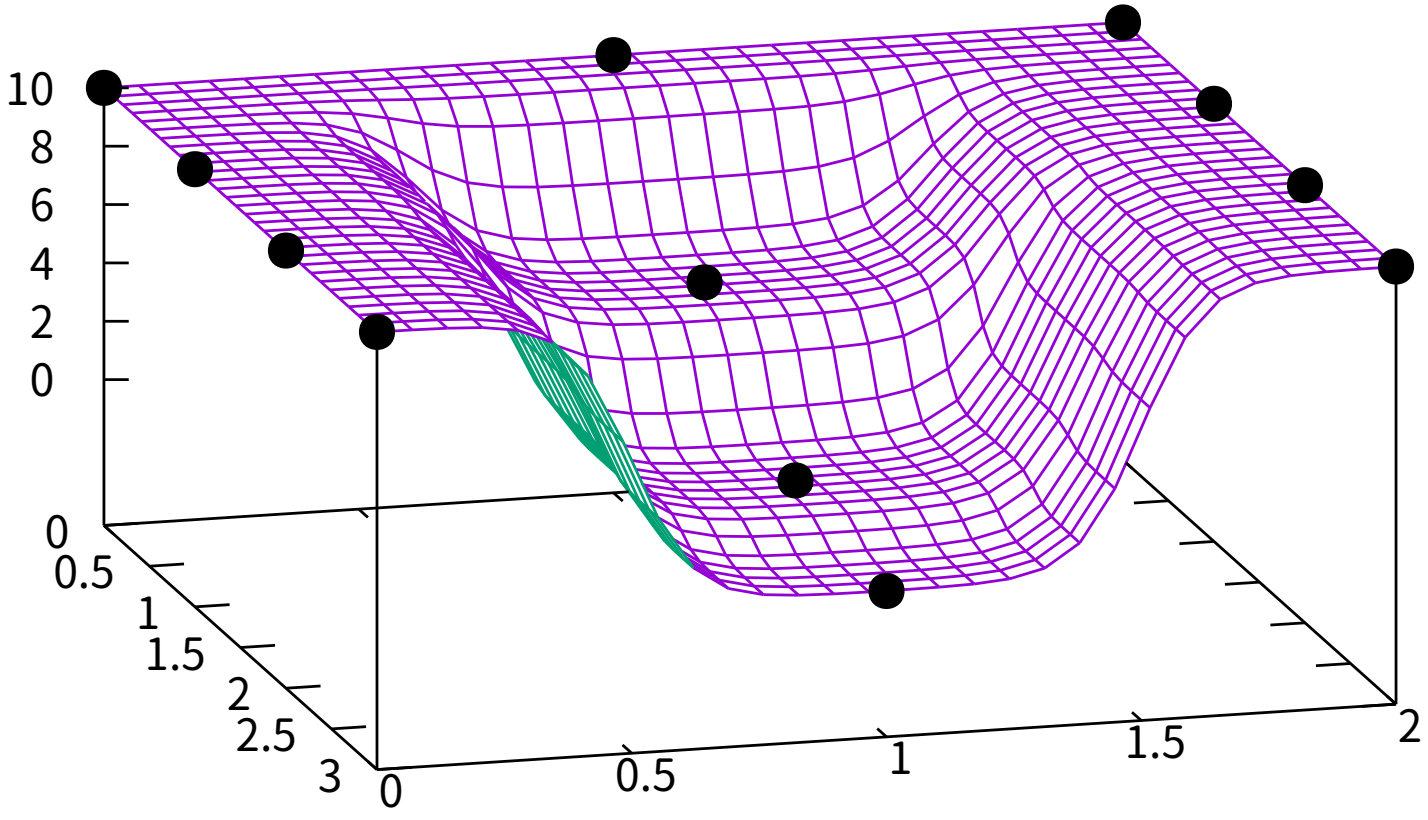
dgrid3d 30,30 gauss .75



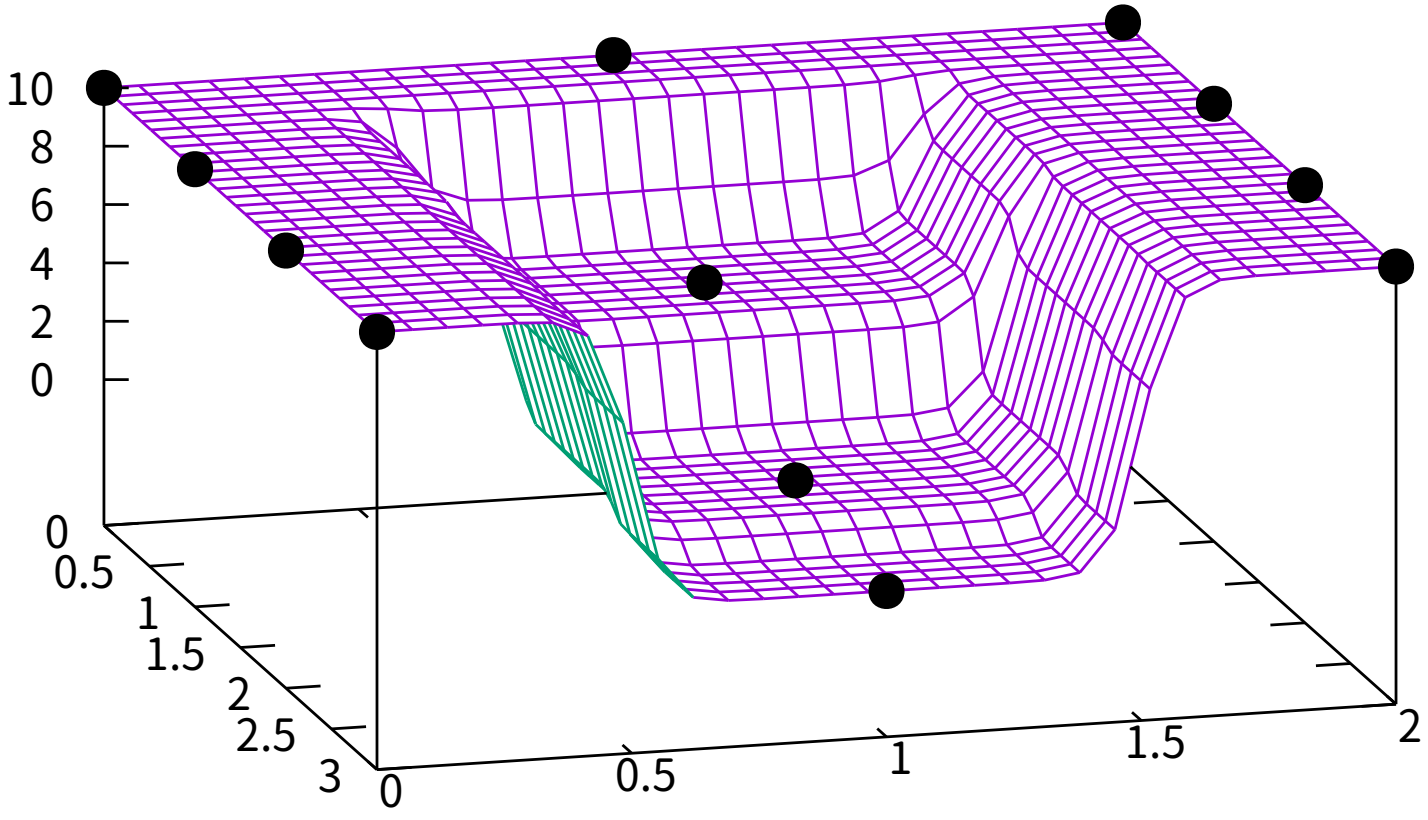
dgrid3d 30,30 gauss .5



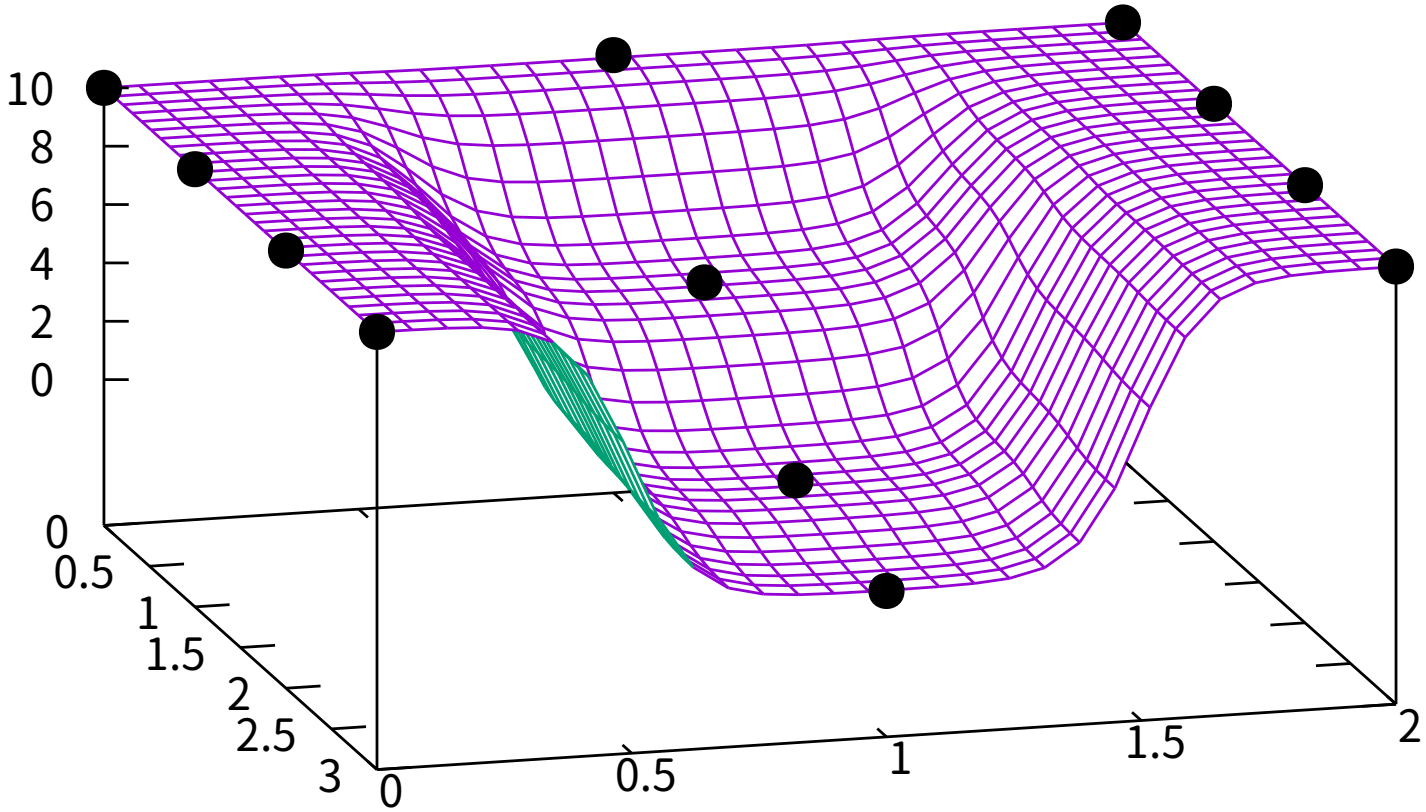
dgrid3d 30,30 gauss .35



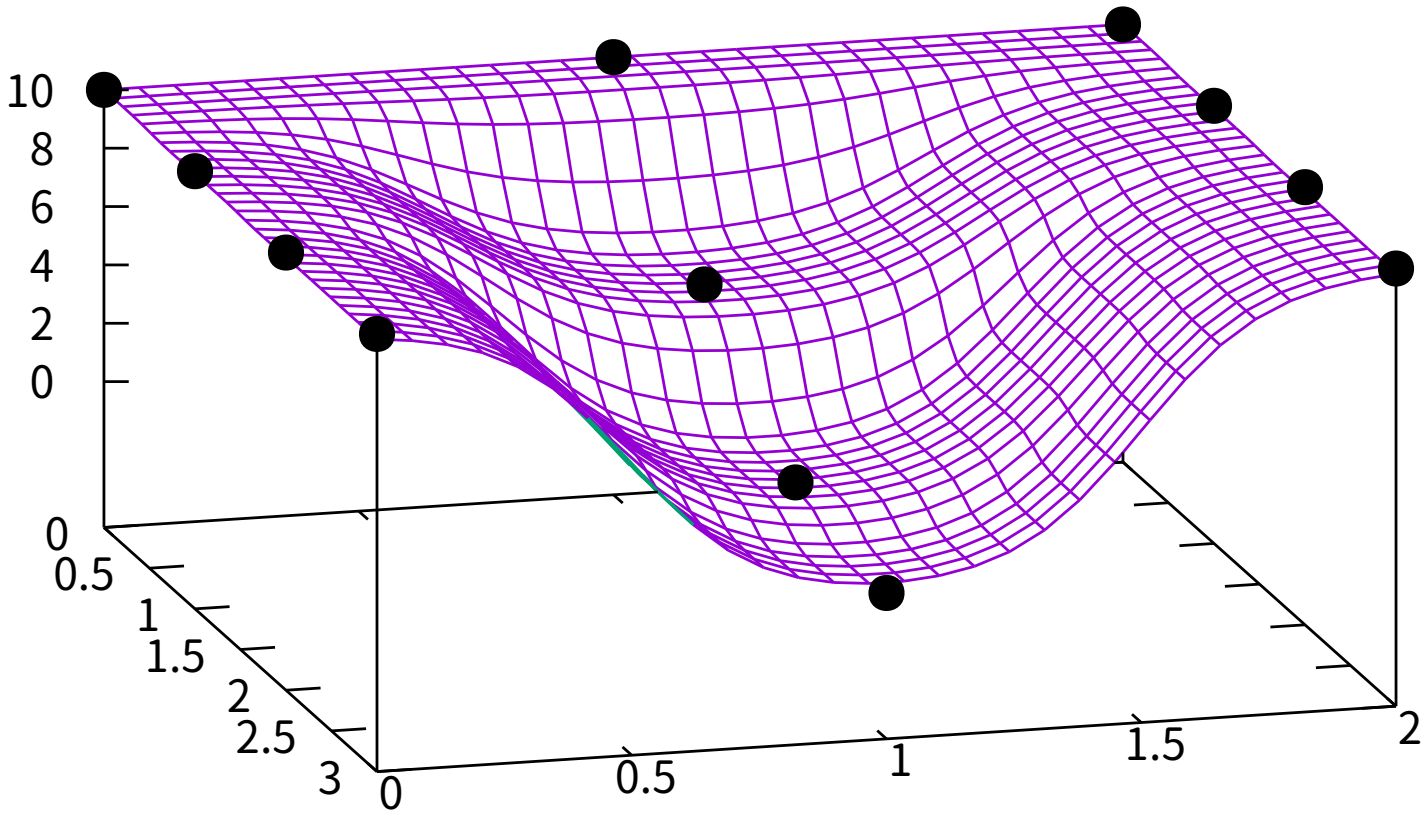
dgrid3d 30,30 gauss .25



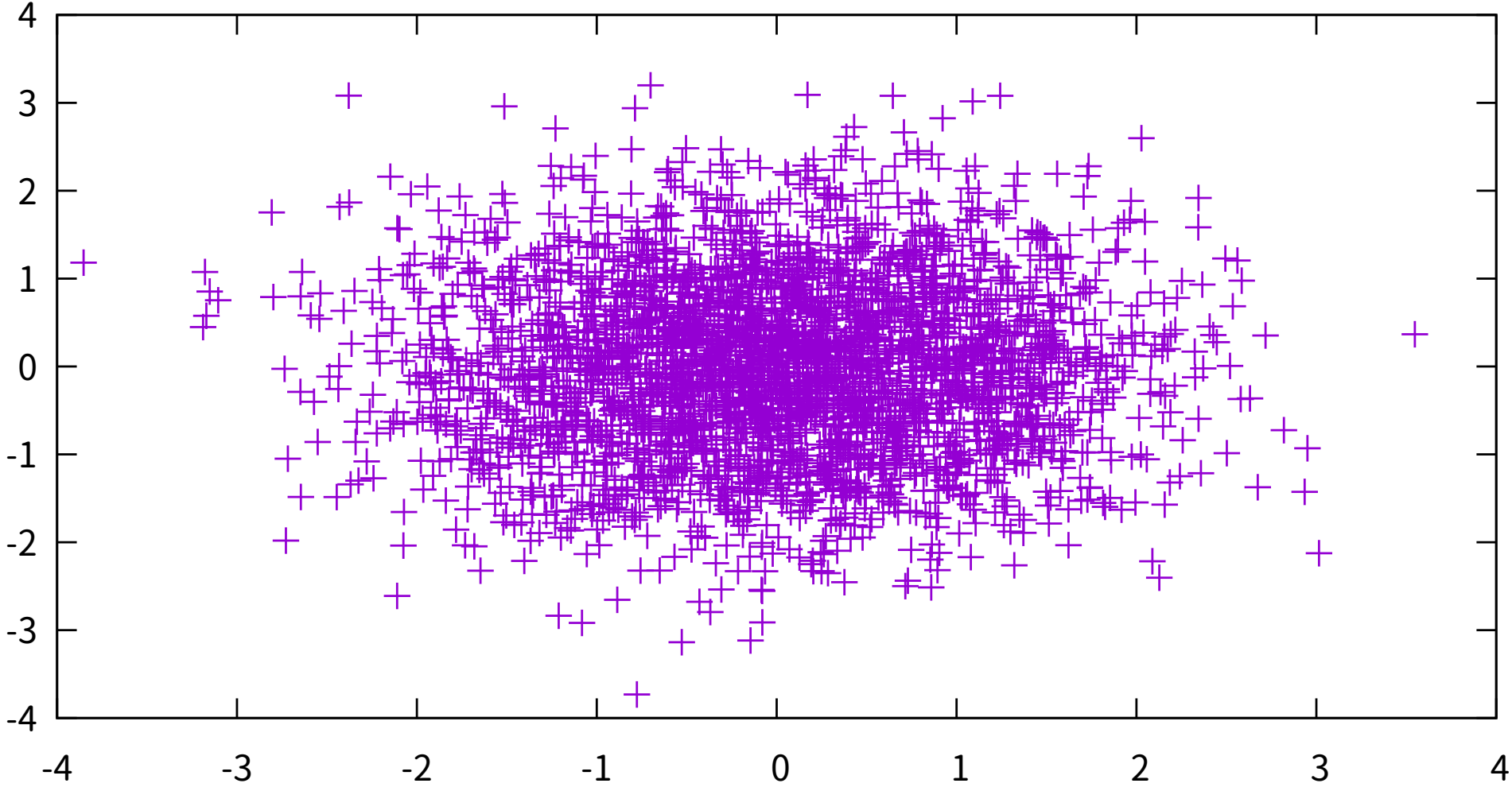
dgrid3d 30,30 gauss .5,.35



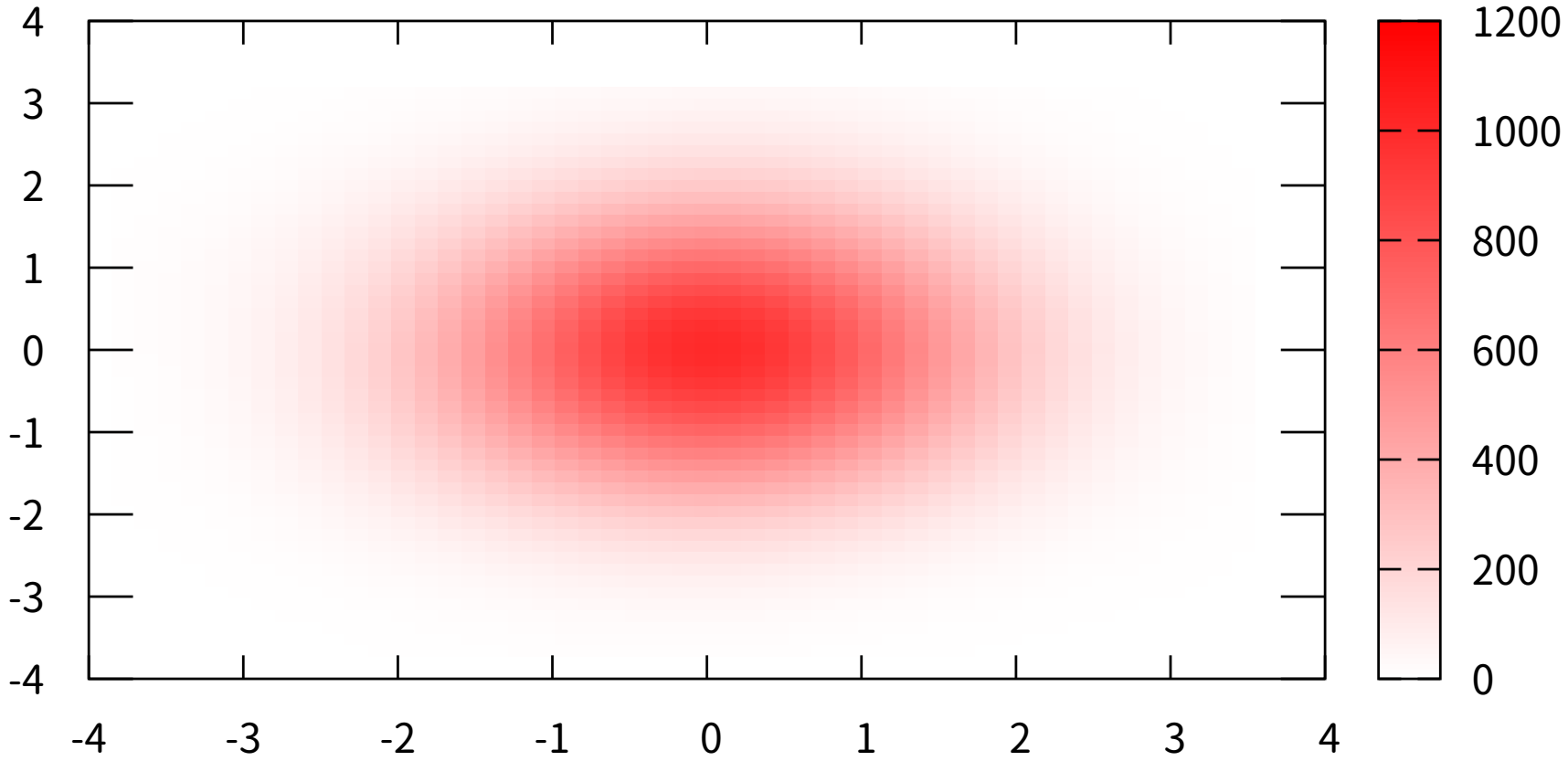
dgrid3d 30,30 gauss .35,.5



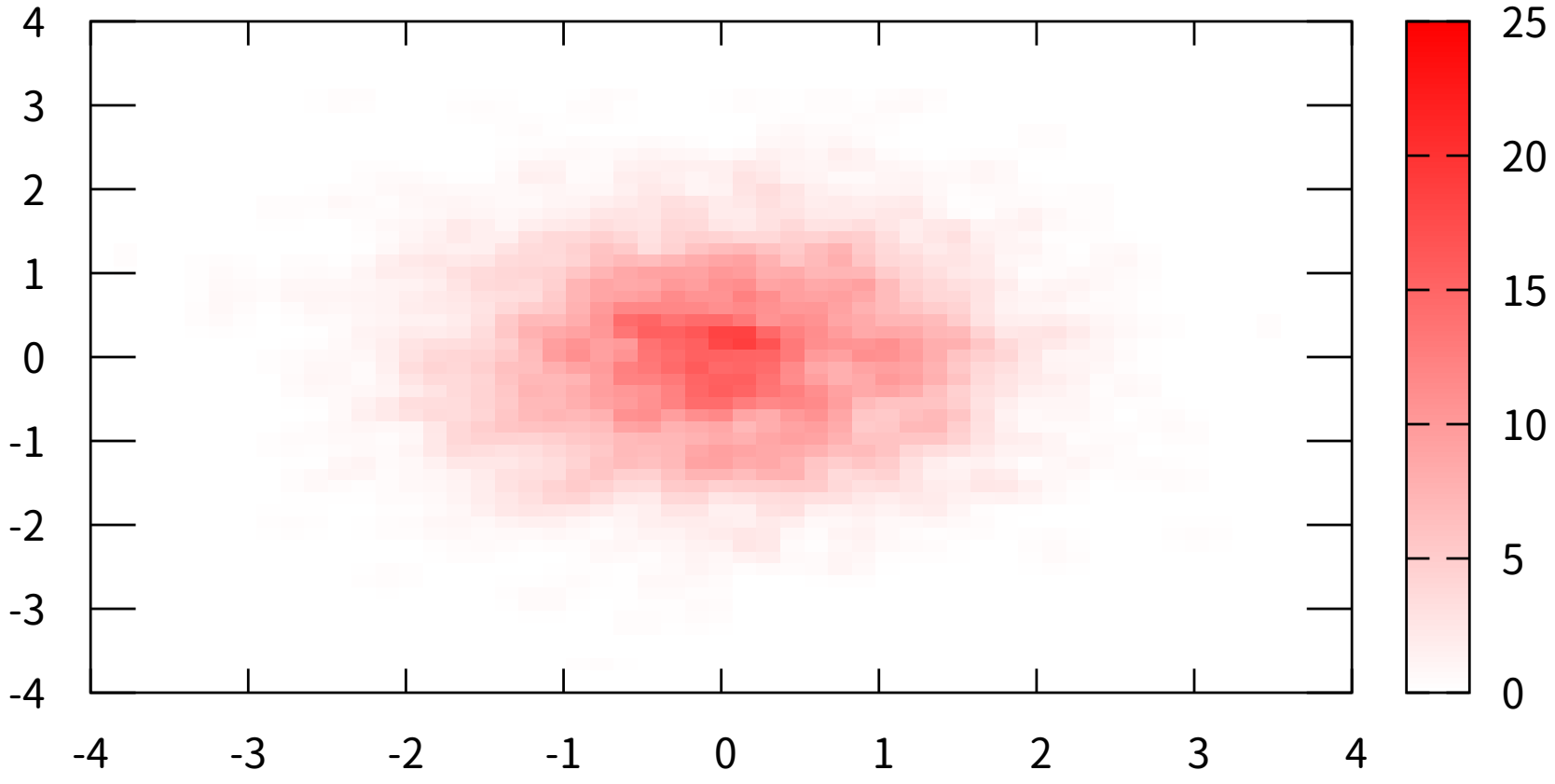
How to plot a kernel density estimate for this 2D dataset?

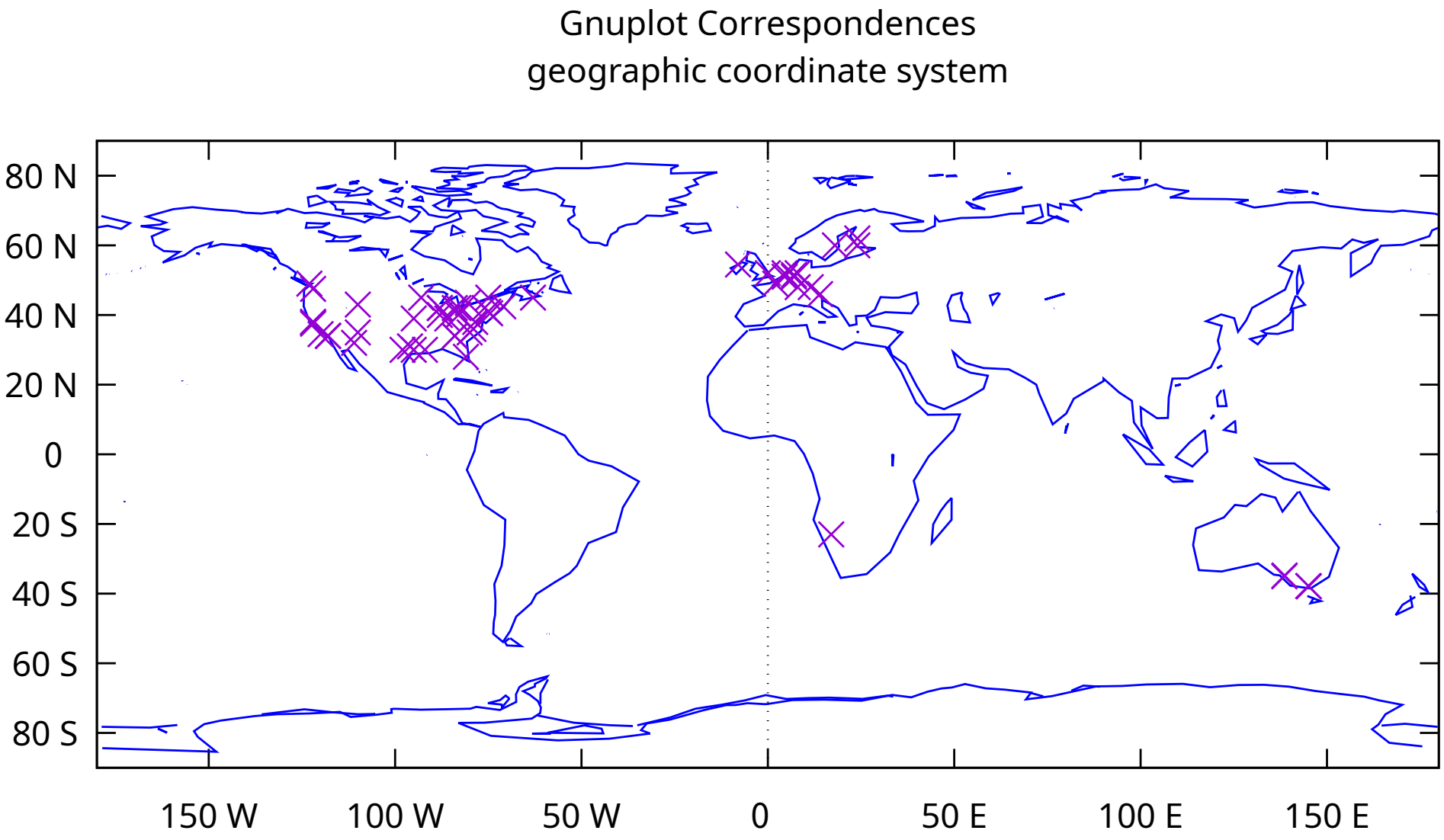


set dgrid3d 50,50 gauss kdensity

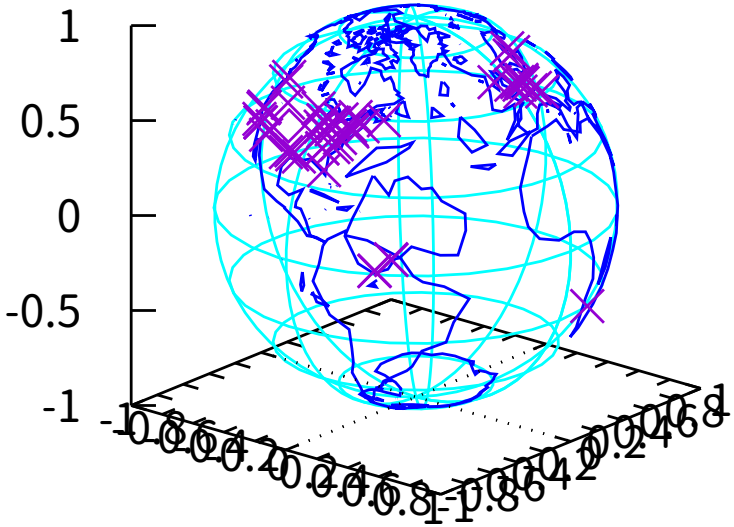


set dgrid3d 50,50 gauss kdensity 0.1

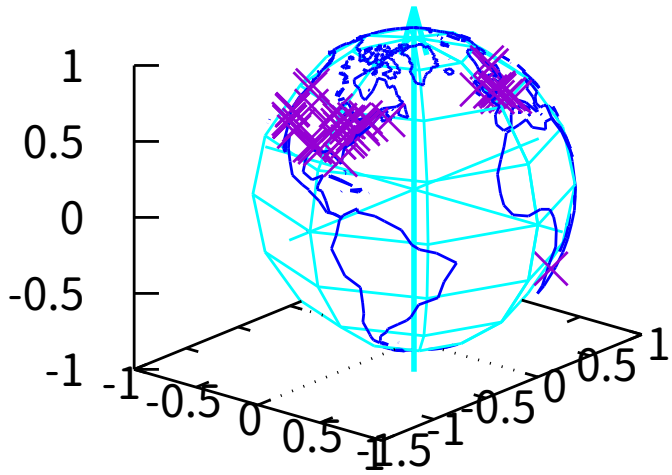



Gnuplot Correspondences
geographic coordinate system
80 N
60 N
40 N
20 N
0
20 S
40 S
60 S
80 S
150 W
100 W
50 W
0
50 E
100 E
150 E

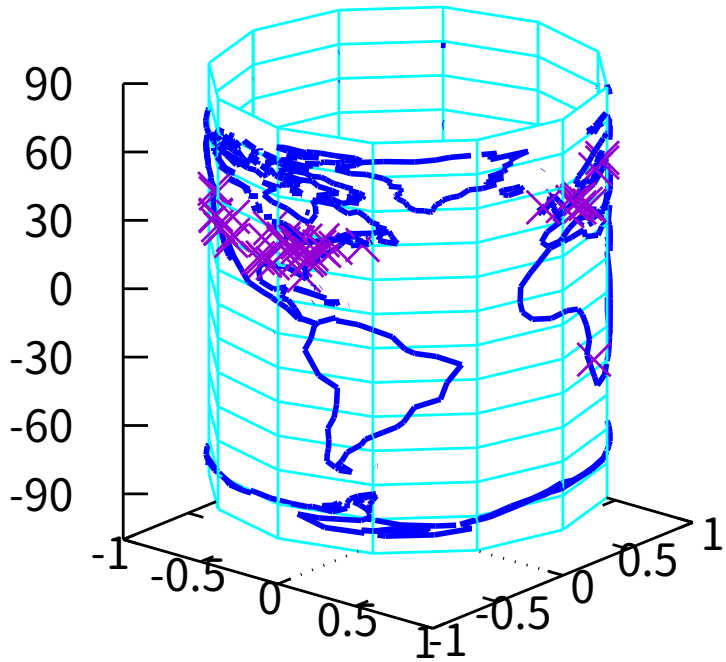
3D version using spherical coordinate system

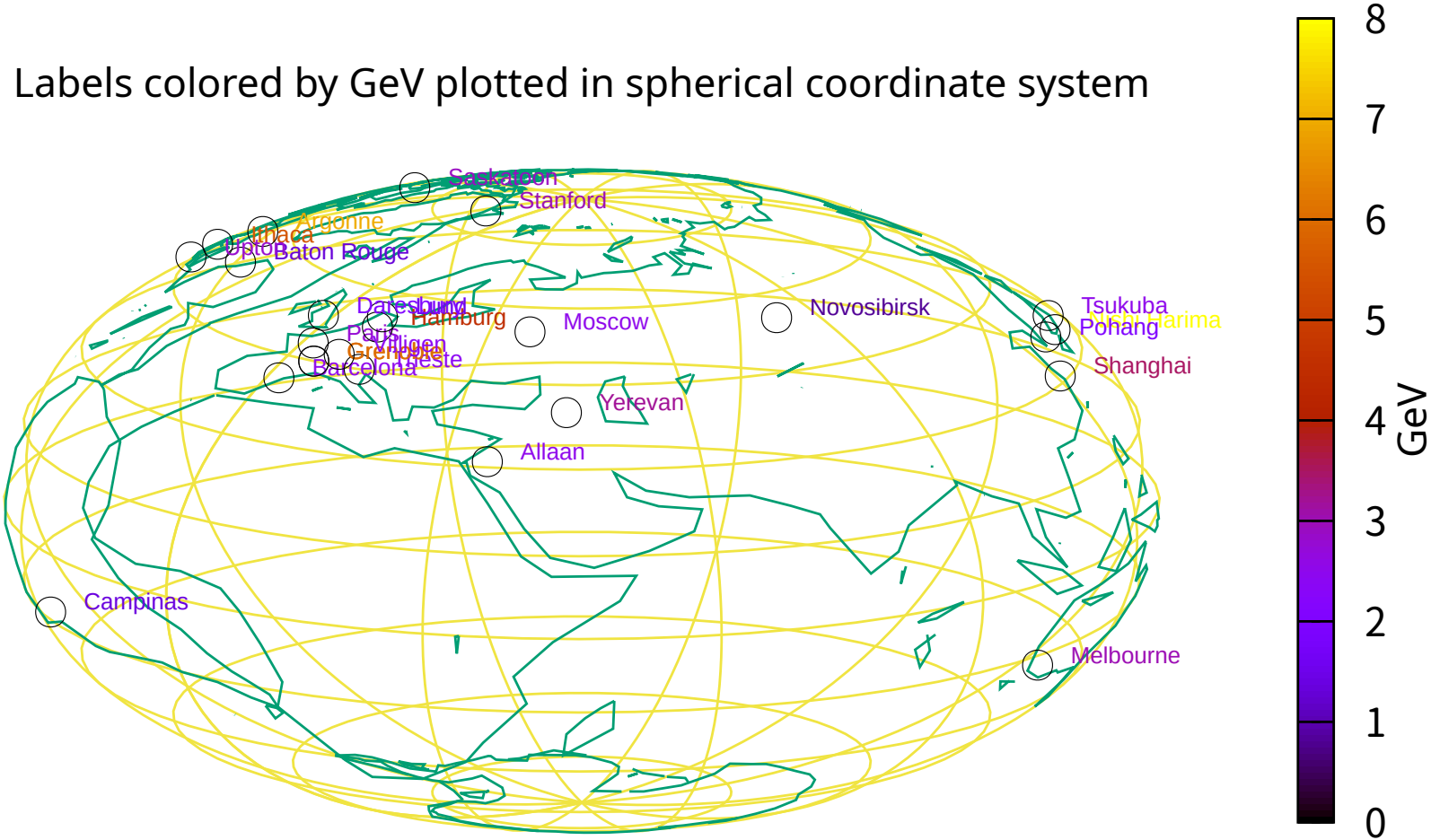


3D solid version with hidden line removal

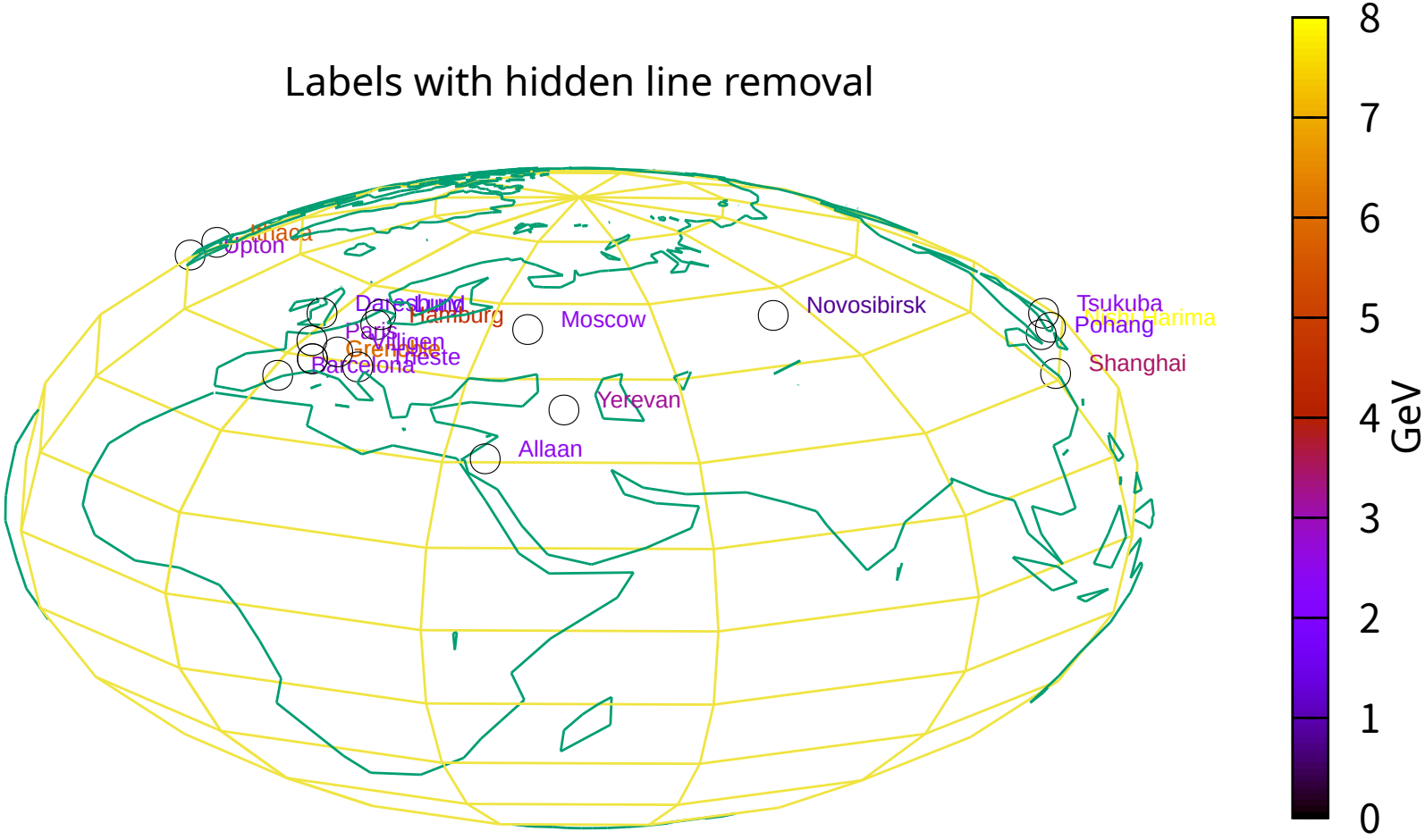


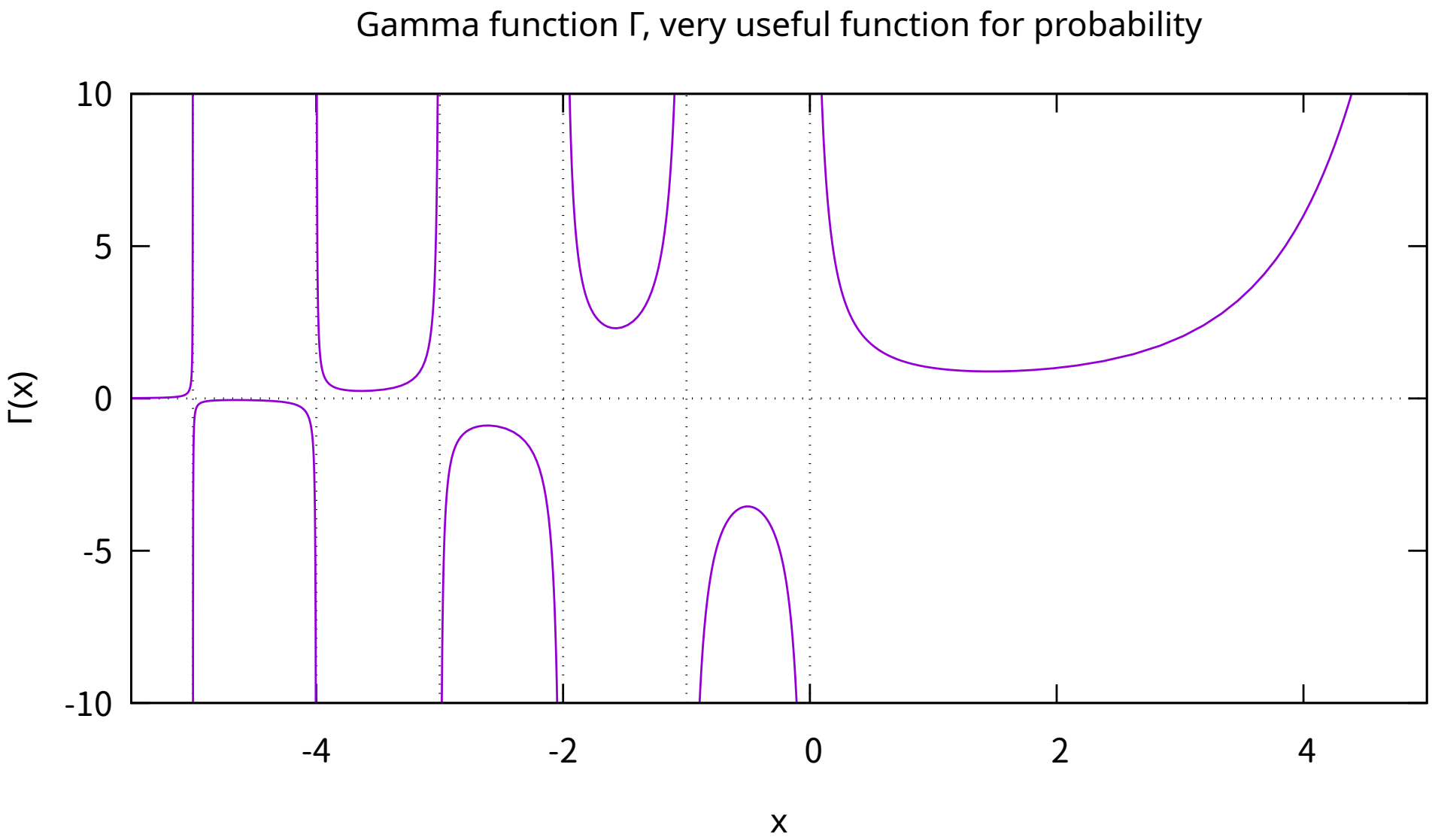
3D version using cylindrical coordinate system

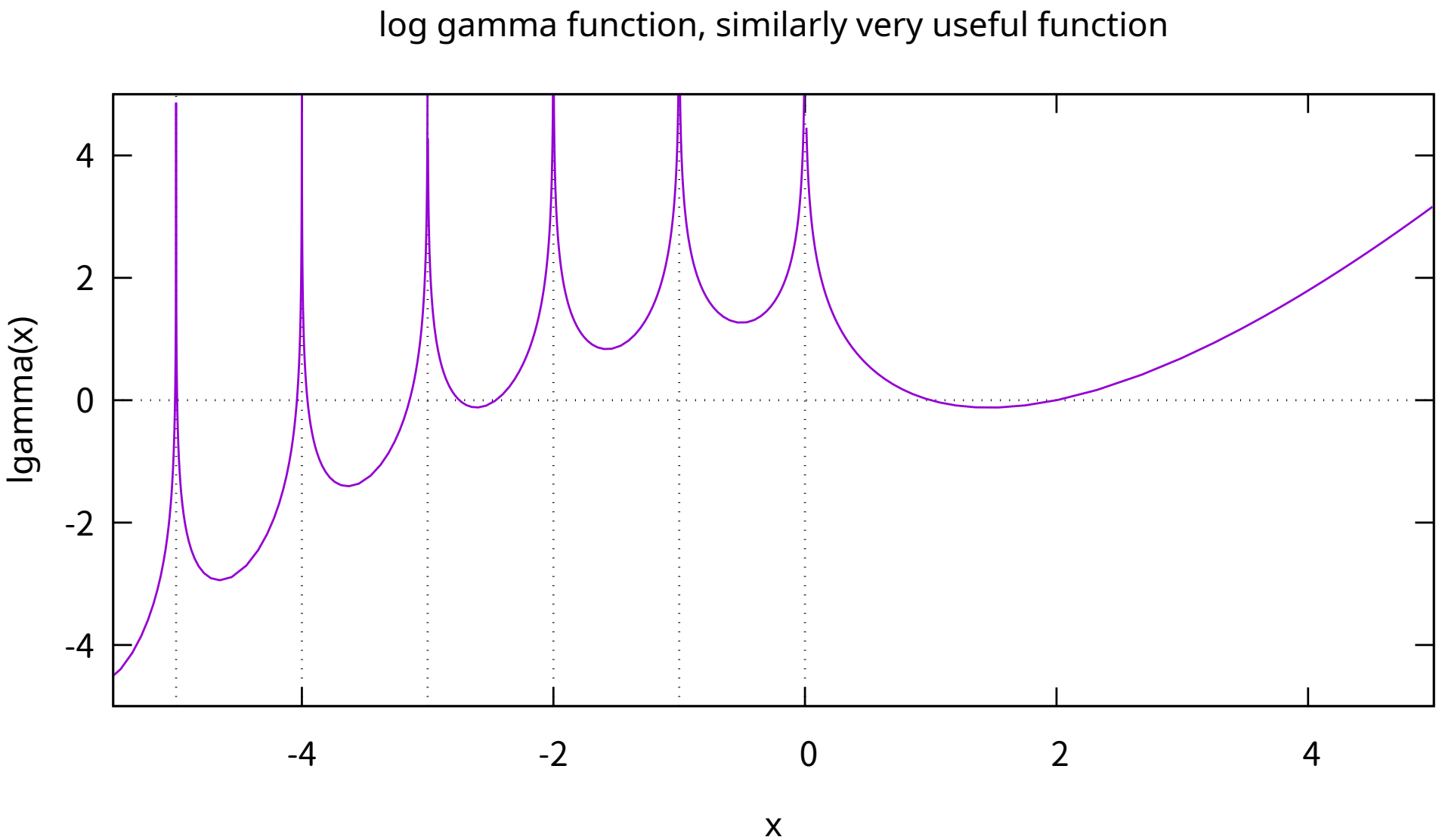


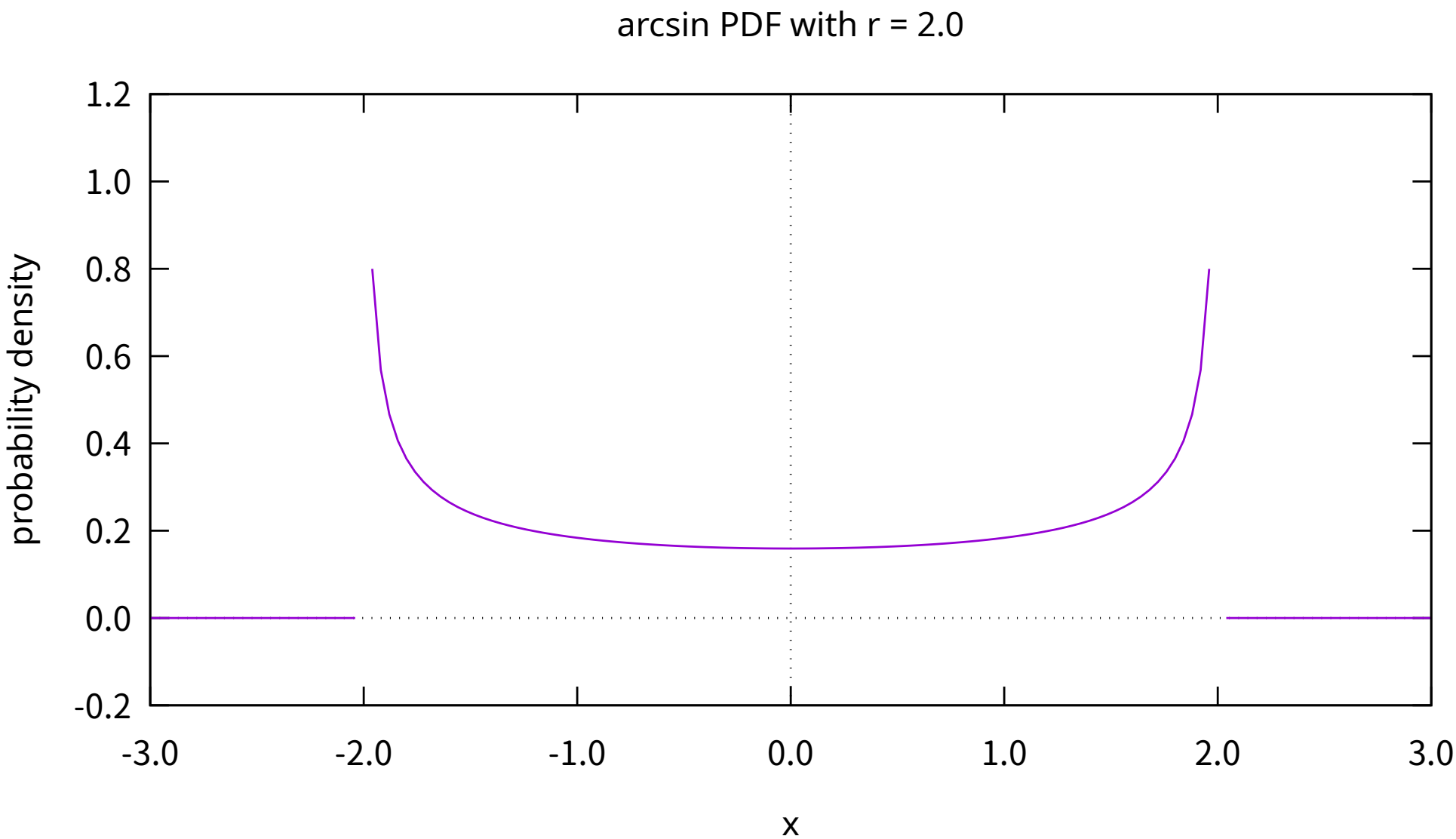

Labels colored by GeV plotted in spherical coordinate system
Saskatoon
Stanford
Argonne
Ithaca
Upto
Baton Rouge
Darmstadt
Hamburg
Moscow
Novosibirsk
Tsukuba
Pohang
Nishinarima
Paris
Villigen
Grenoble
Trieste
Barcelona
Shanghai
Yerevan
Allaan
Campinas
Melbourne
GeV
0
1
2
3
4
5
6
7
8

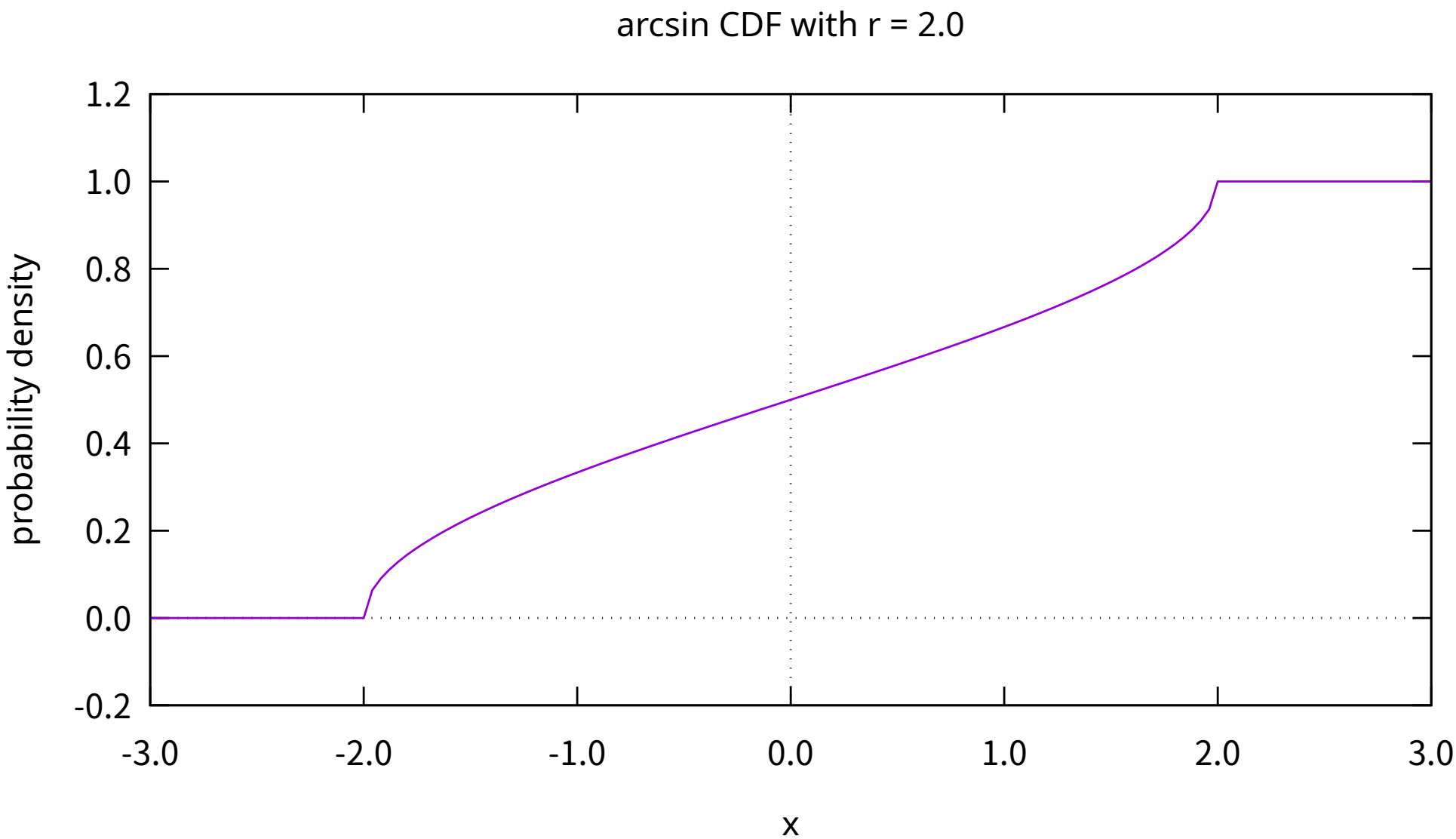
Labels with hidden line removal

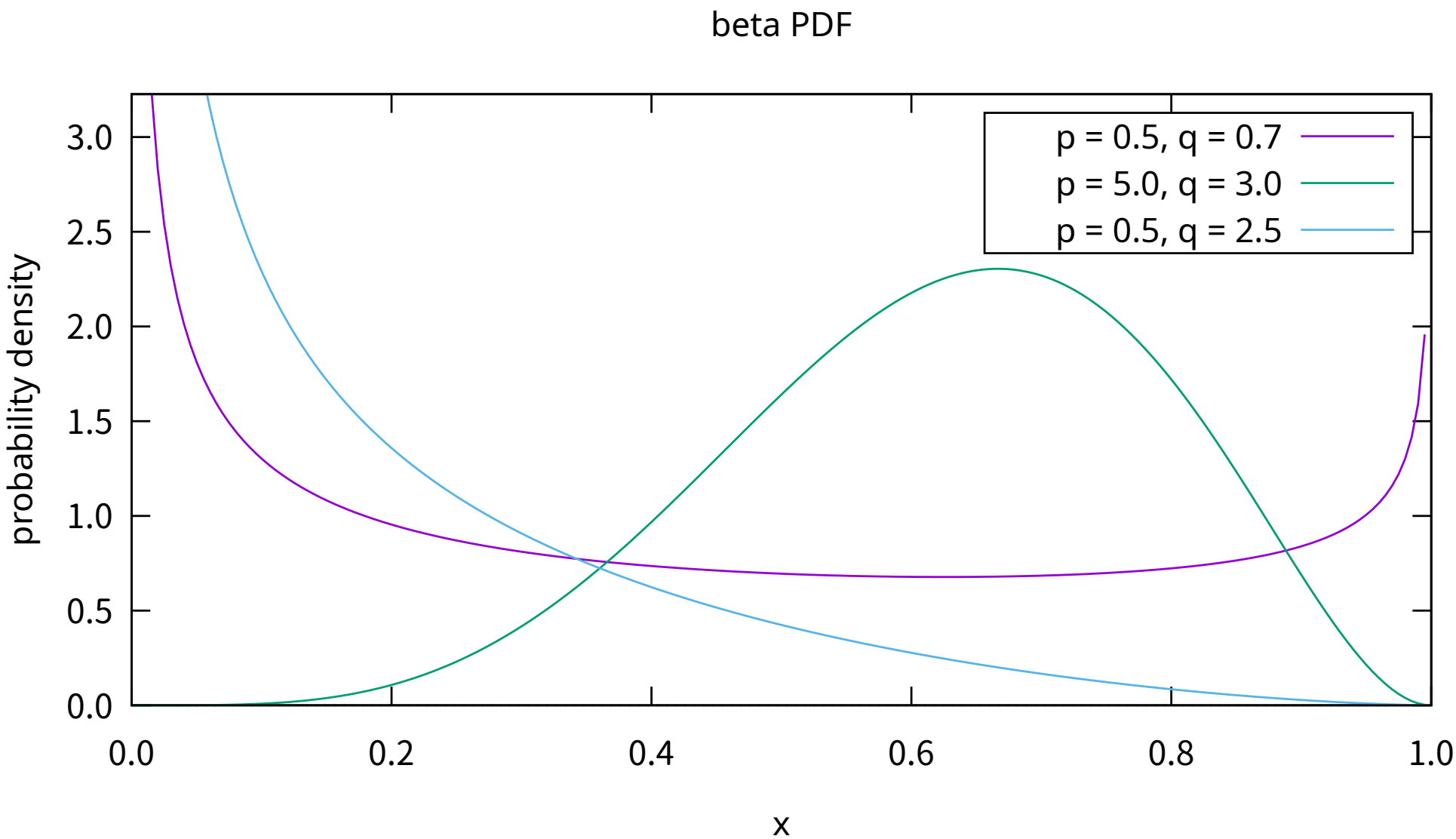


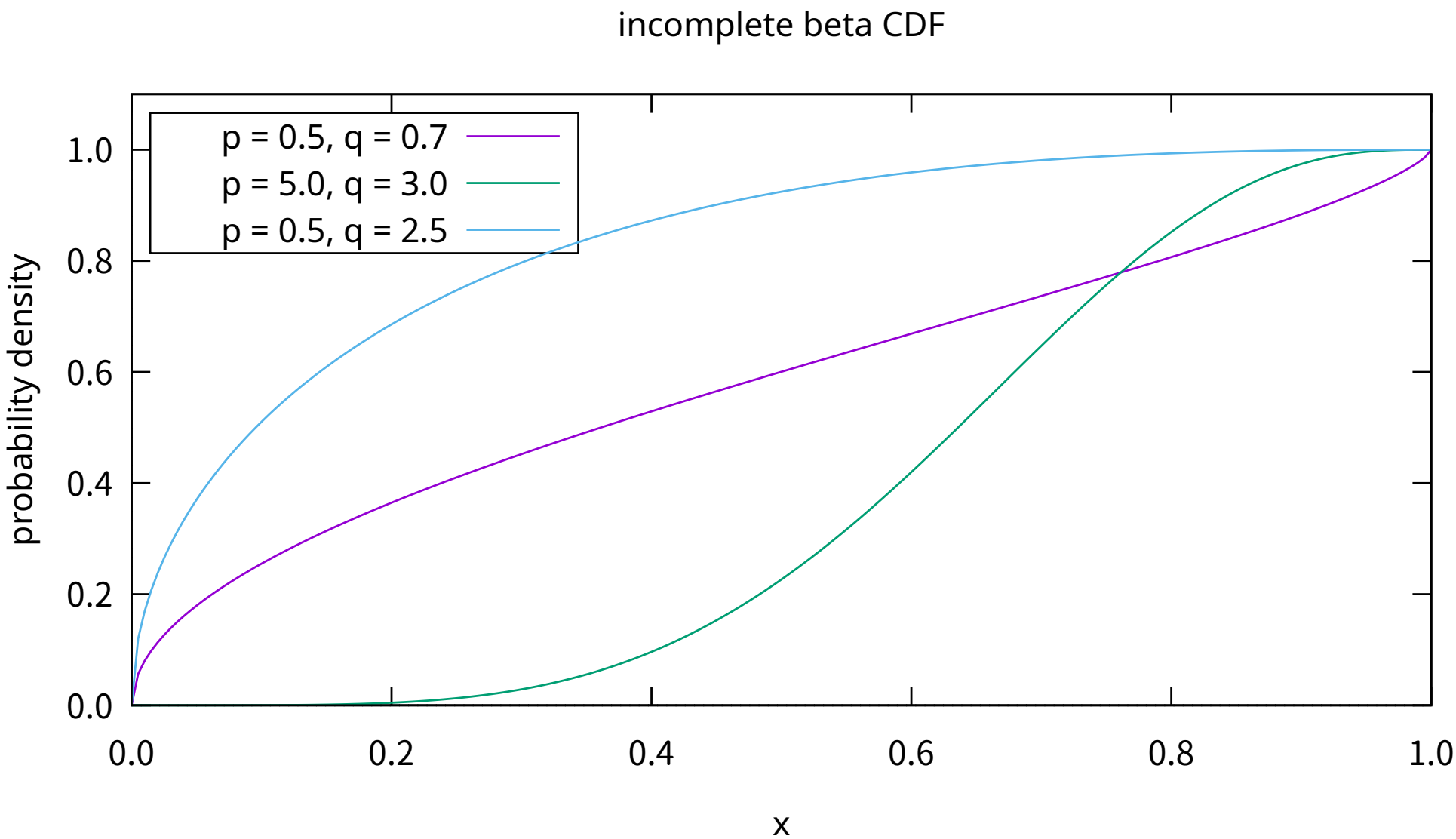

Gamma function Γ, very useful function for probability
10
5
0
-5
-10
Γ(x)
-4
-2
0
2
4
x

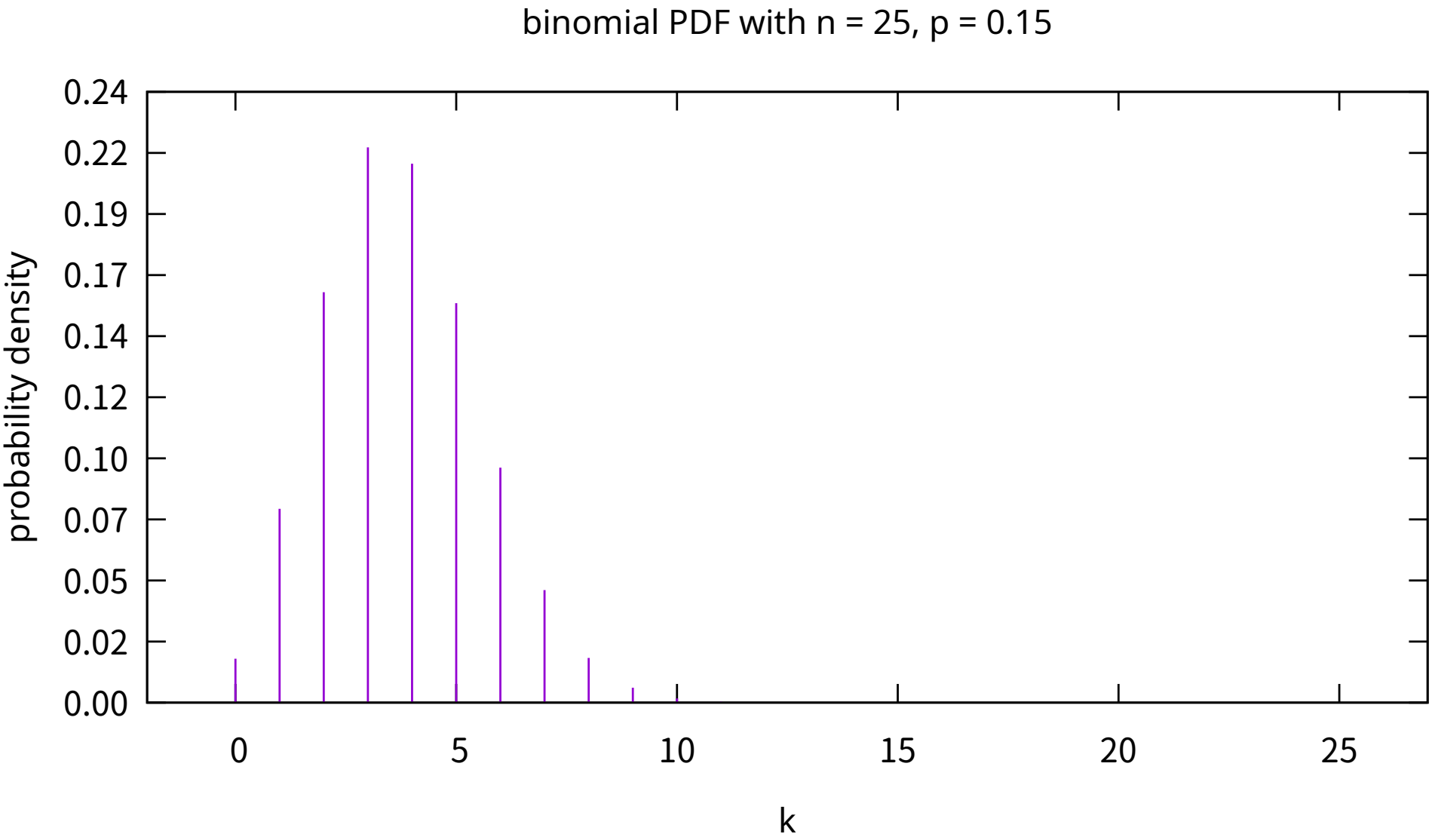

log gamma function, similarly very useful function
lgamma(x)
x
4
2
0
-2
-4
-4
-2
0
2
4

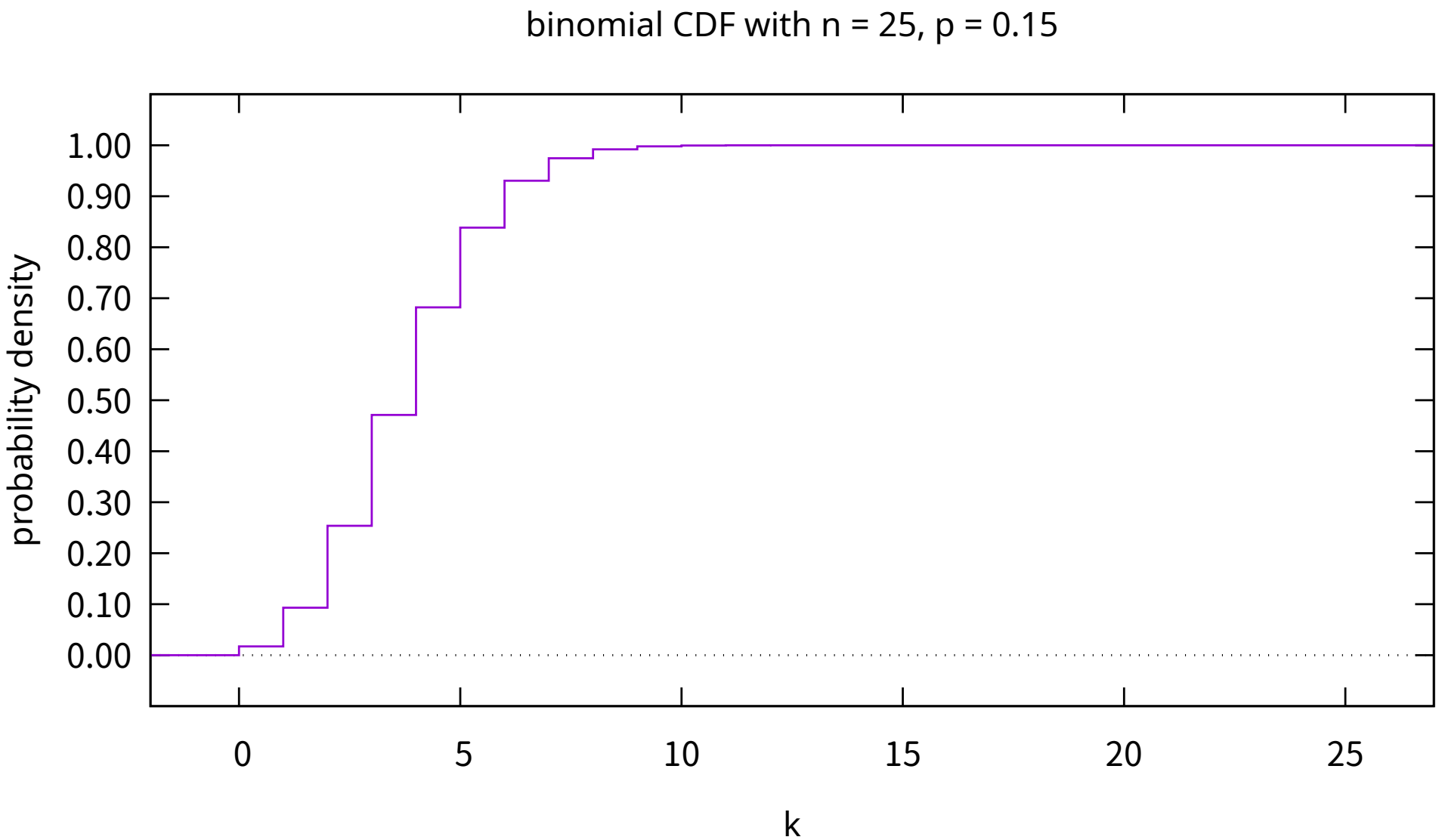

arcsin PDF with r = 2.0
probability density
x
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

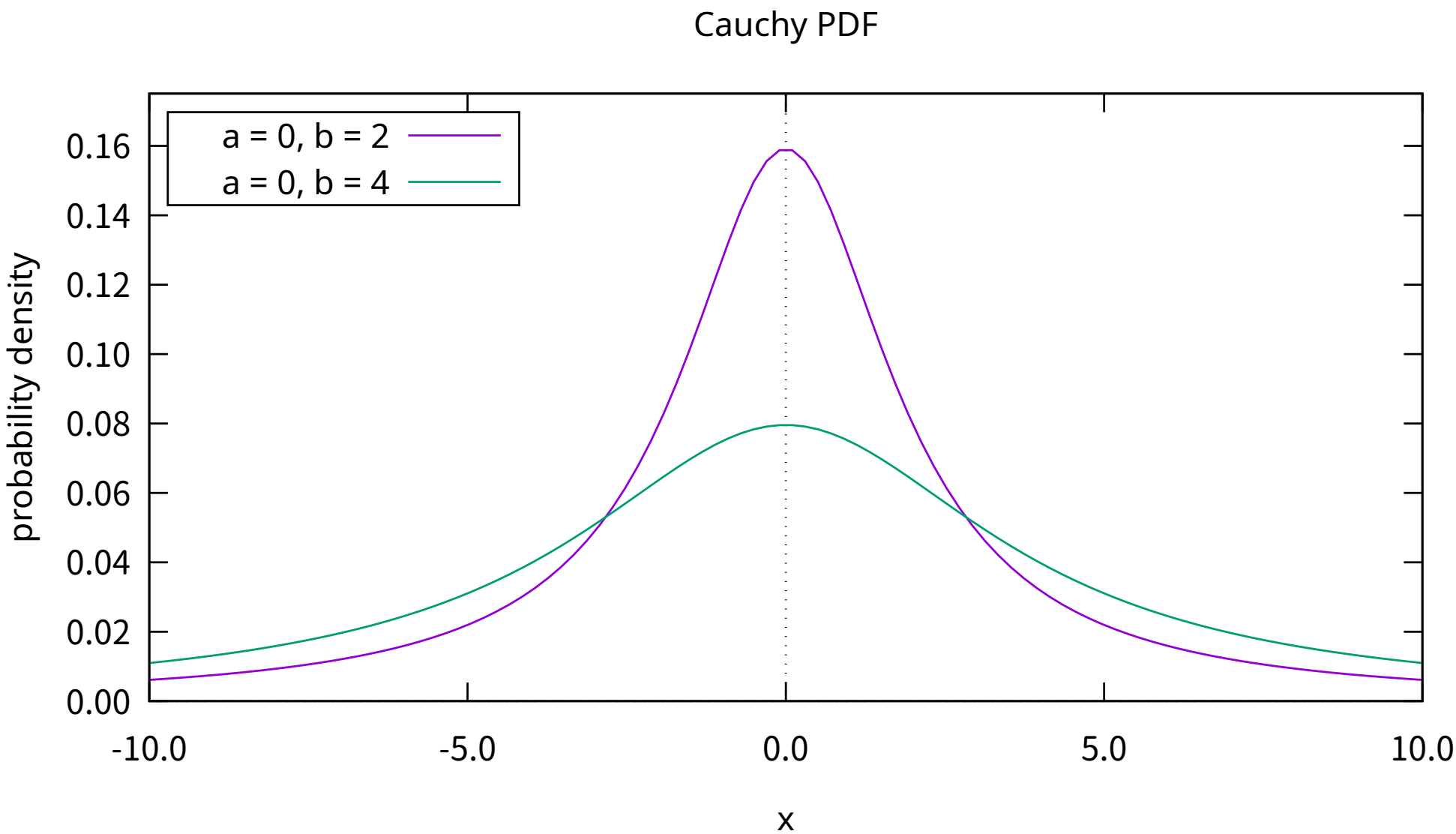

arcsin CDF with r = 2.0
probability density
x
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

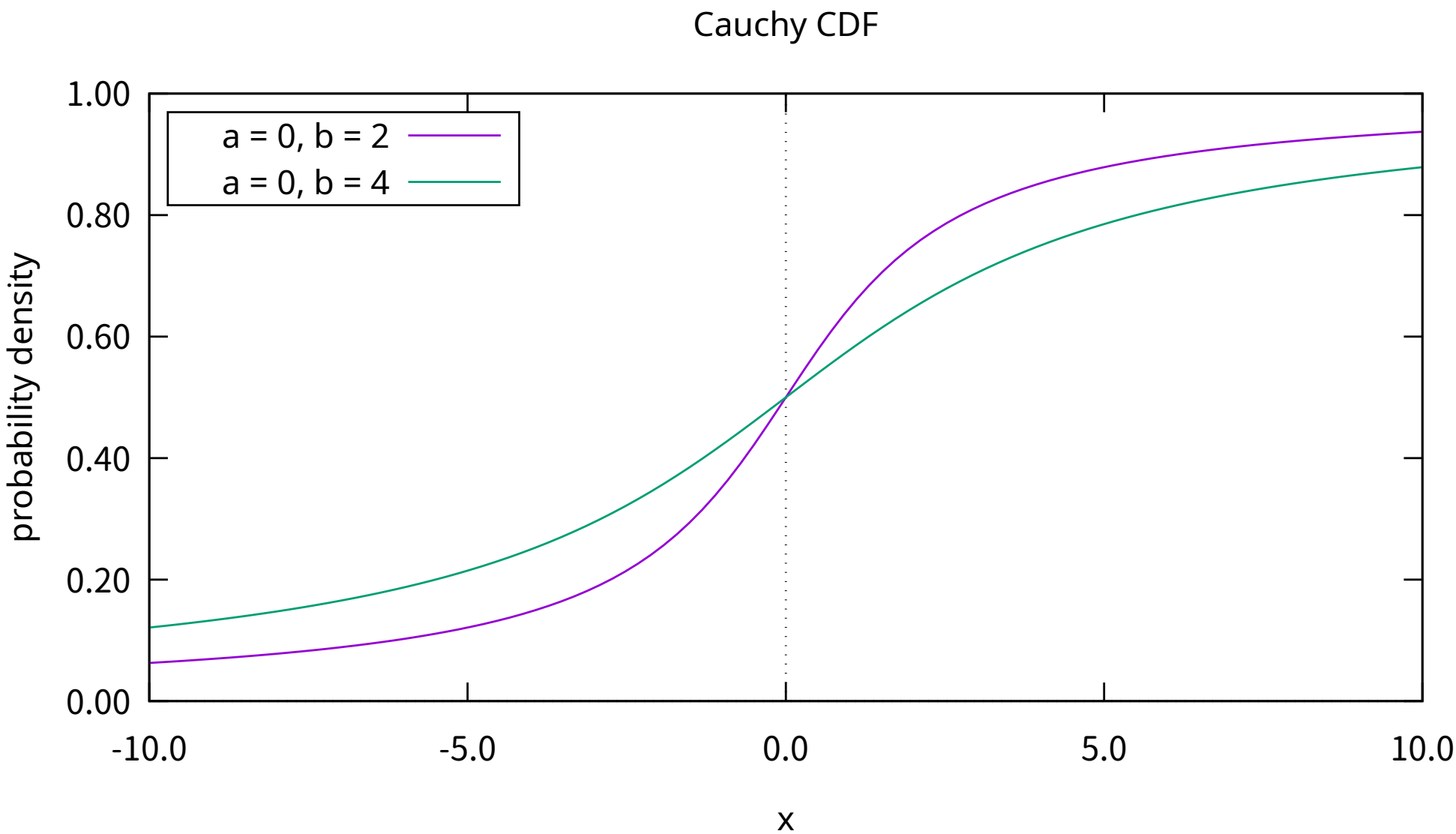

beta PDF
p = 0.5, q = 0.7
p = 5.0, q = 3.0
p = 0.5, q = 2.5
probability density
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0
x

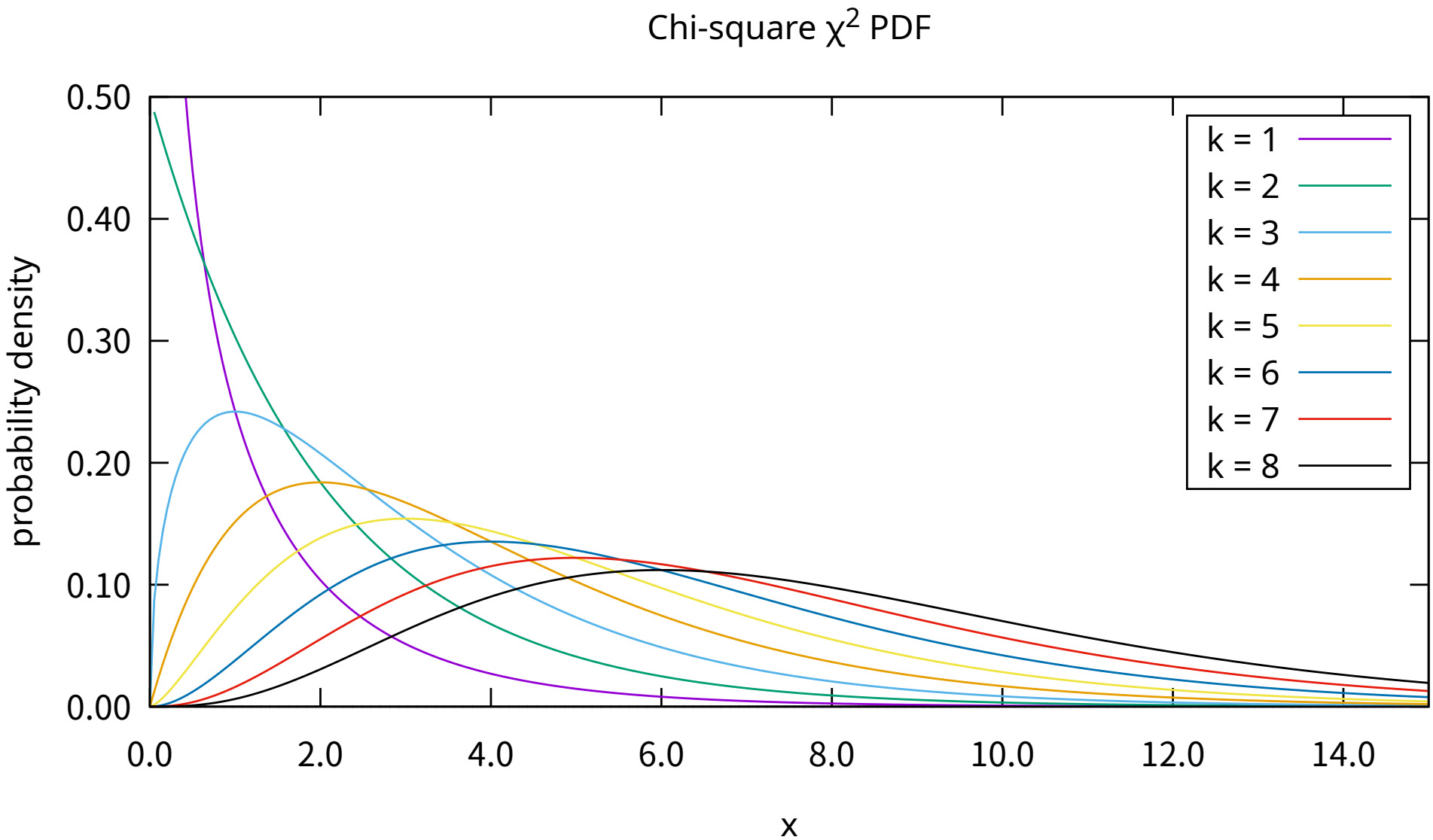

incomplete beta CDF
p = 0.5, q = 0.7
p = 5.0, q = 3.0
p = 0.5, q = 2.5
probability density
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
x

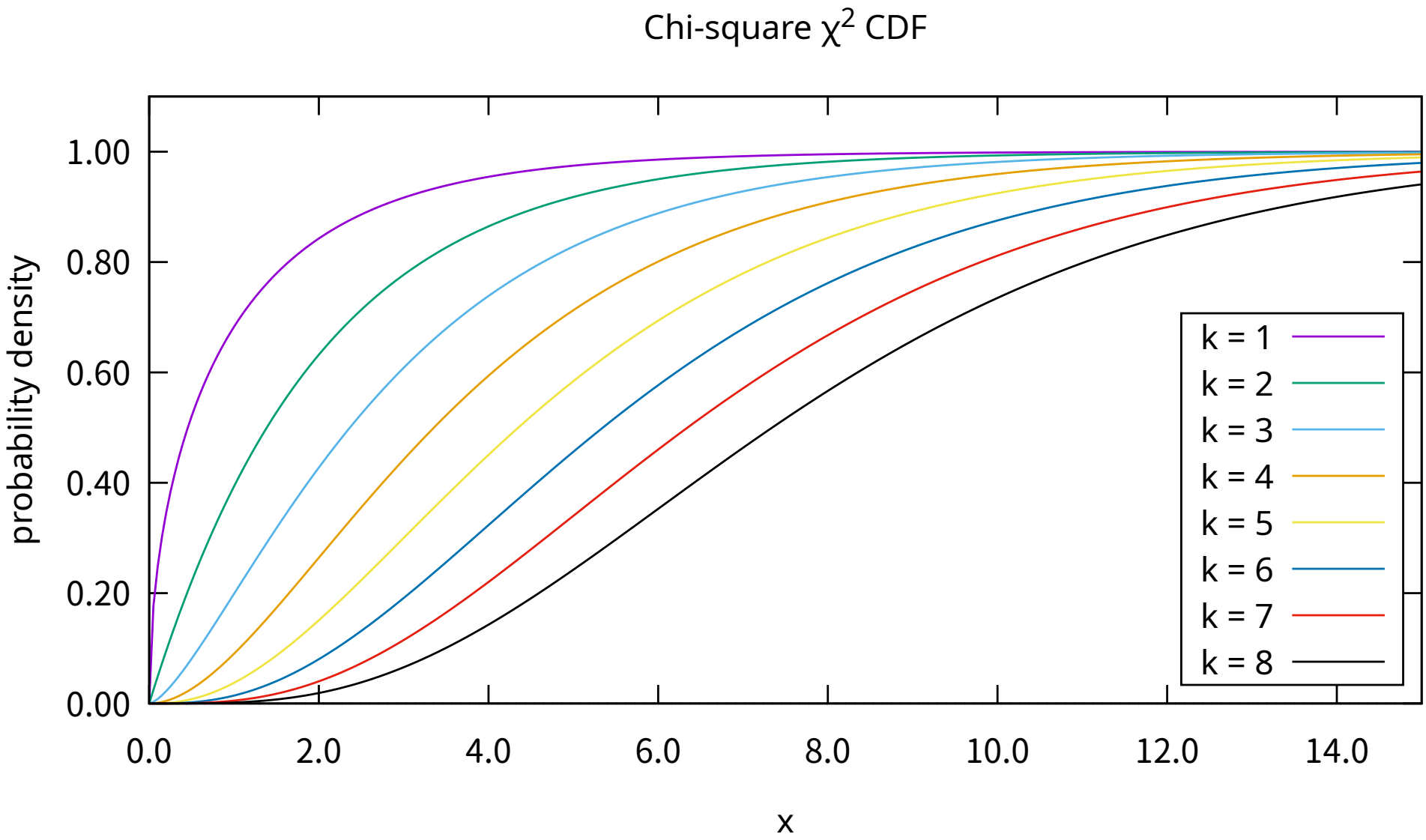

binomial PDF with n = 25, p = 0.15
probability density
0.24
0.22
0.19
0.17
0.14
0.12
0.10
0.07
0.05
0.02
0.00
0
5
10
15
20
25
k

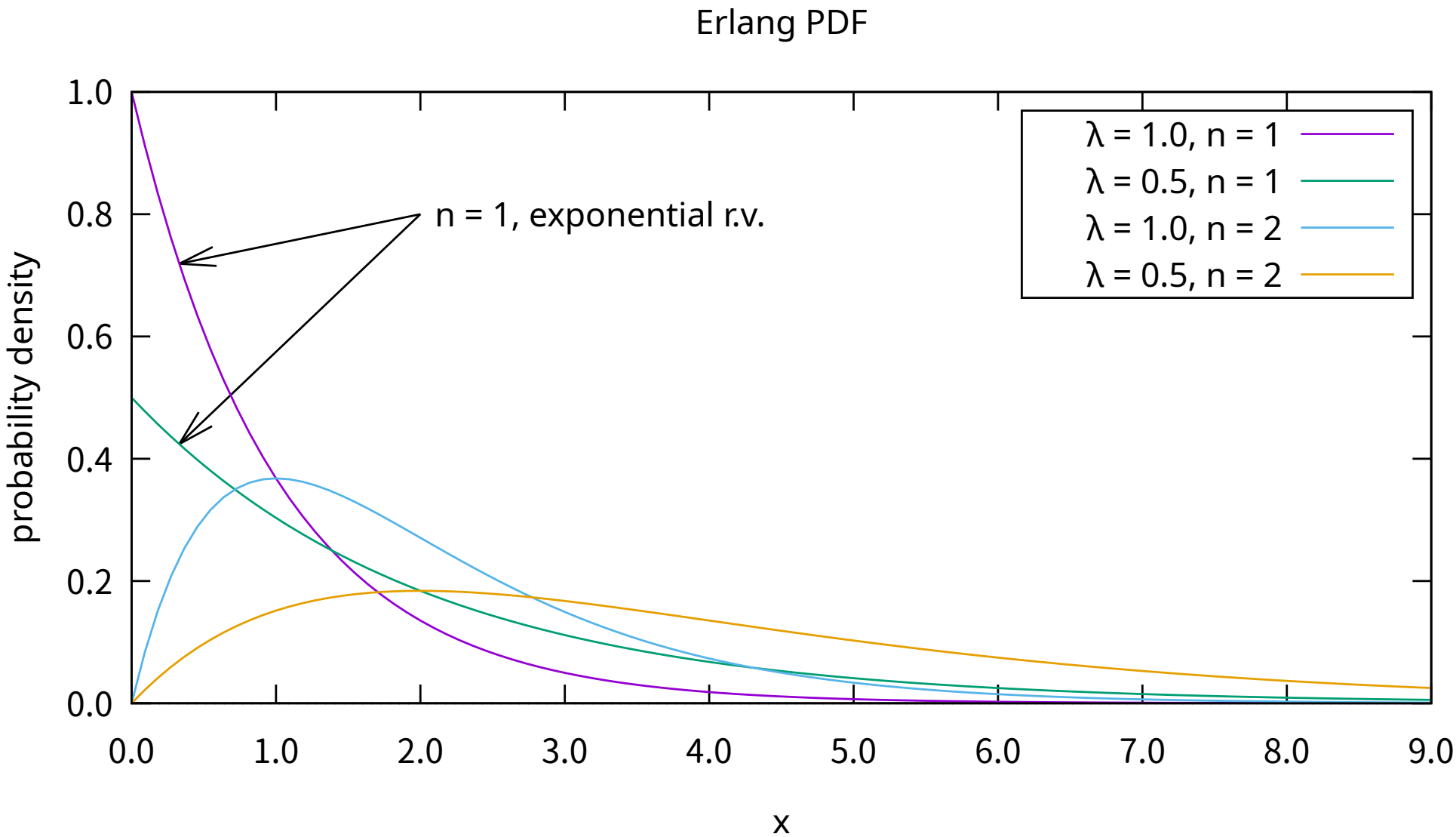

binomial CDF with n = 25, p = 0.15
probability density
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0
5
10
15
20
25
k

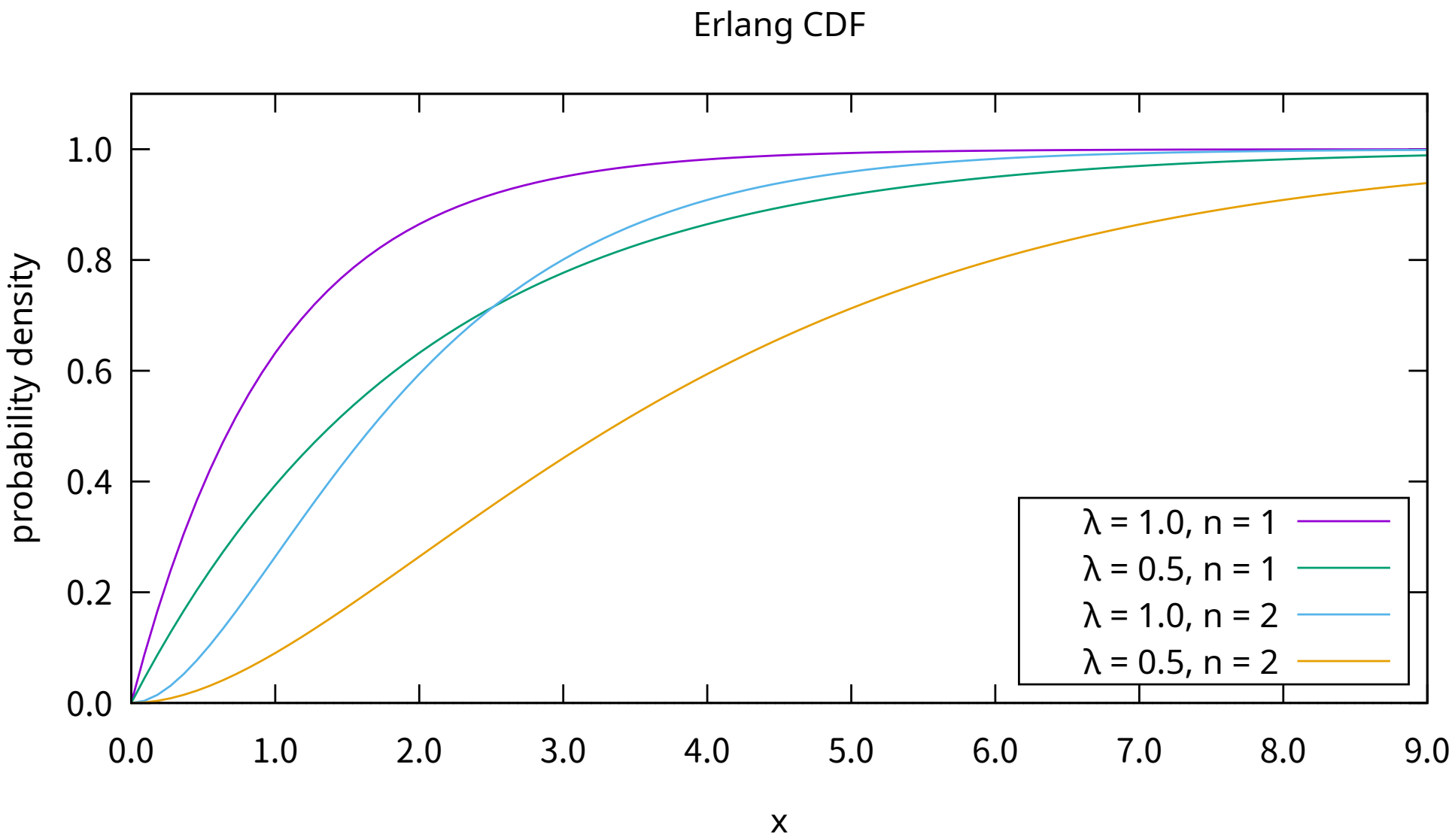

Cauchy PDF
a = 0, b = 2
a = 0, b = 4
probability density
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
-10.0
-5.0
0.0
5.0
10.0
x

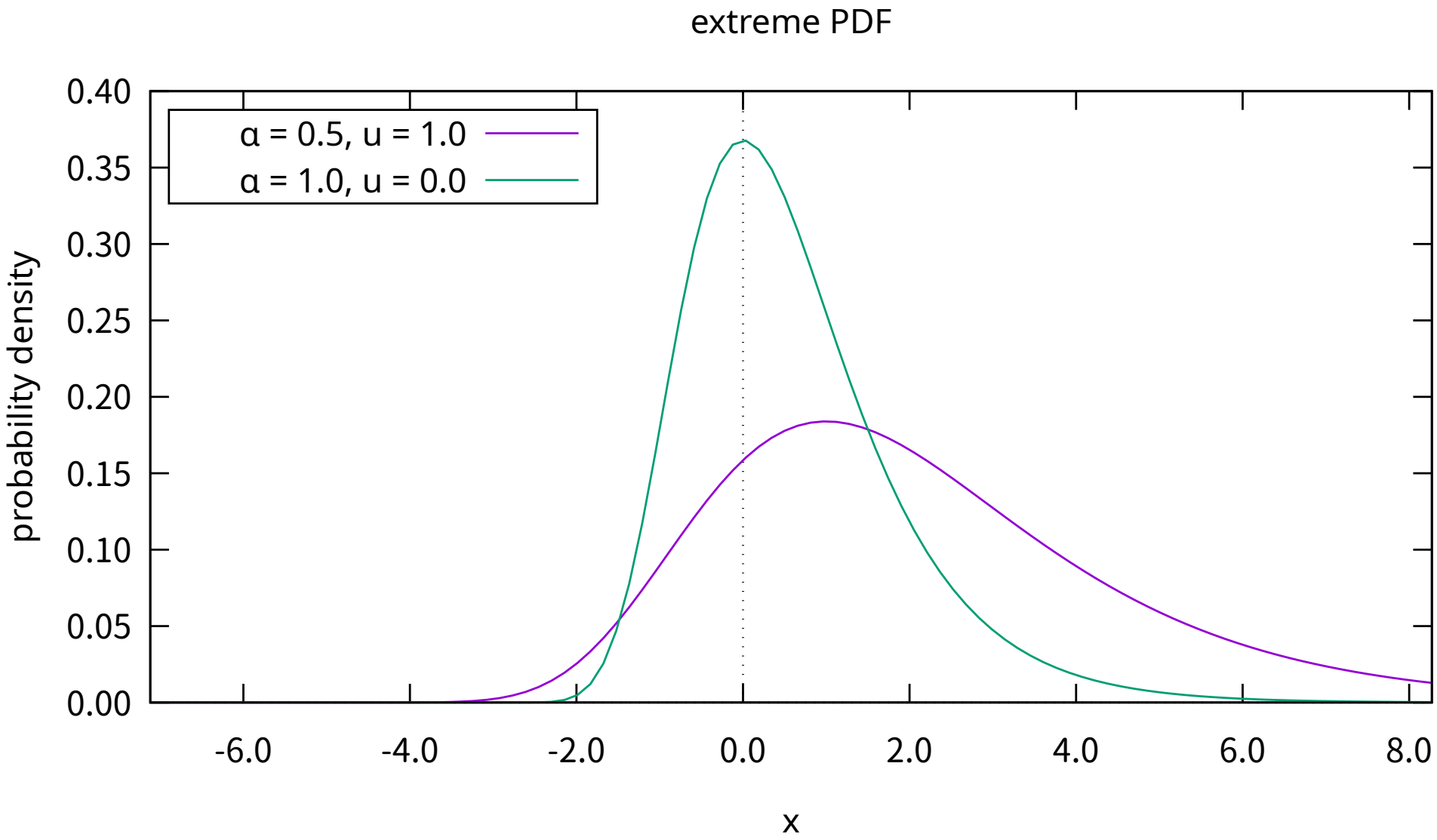

Cauchy CDF
a = 0, b = 2
a = 0, b = 4
probability density
1.00
0.80
0.60
0.40
0.20
0.00
-10.0
-5.0
0.0
5.0
10.0
x

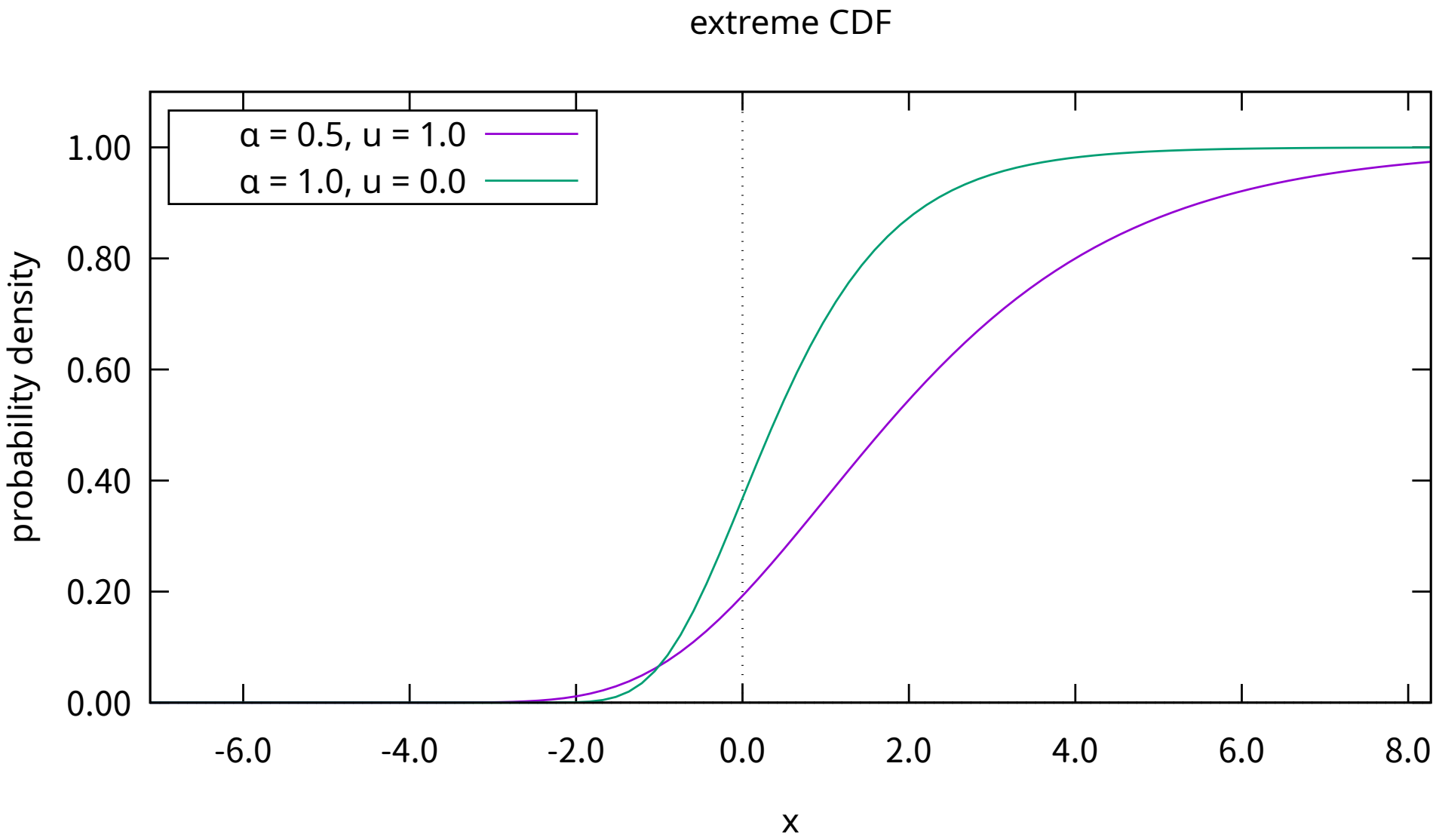

Chi-square χ2 PDF
probability density
0.50
0.40
0.30
0.20
0.10
0.00
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
x
k = 1
k = 2
k = 3
k = 4
k = 5
k = 6
k = 7
k = 8

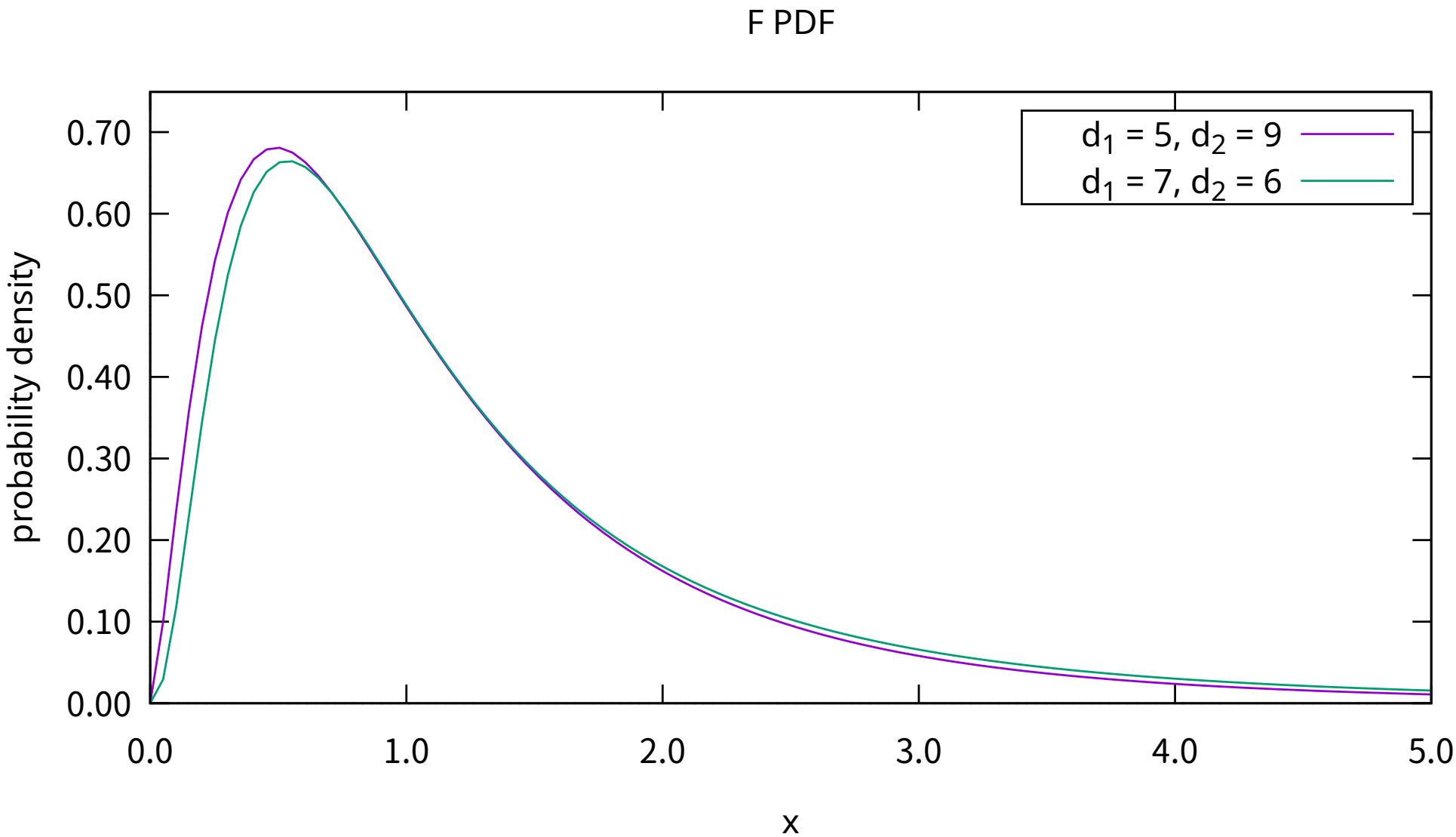

Chi-square χ2 CDF
probability density
1.00
0.80
0.60
0.40
0.20
0.00
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
x
k = 1
k = 2
k = 3
k = 4
k = 5
k = 6
k = 7
k = 8

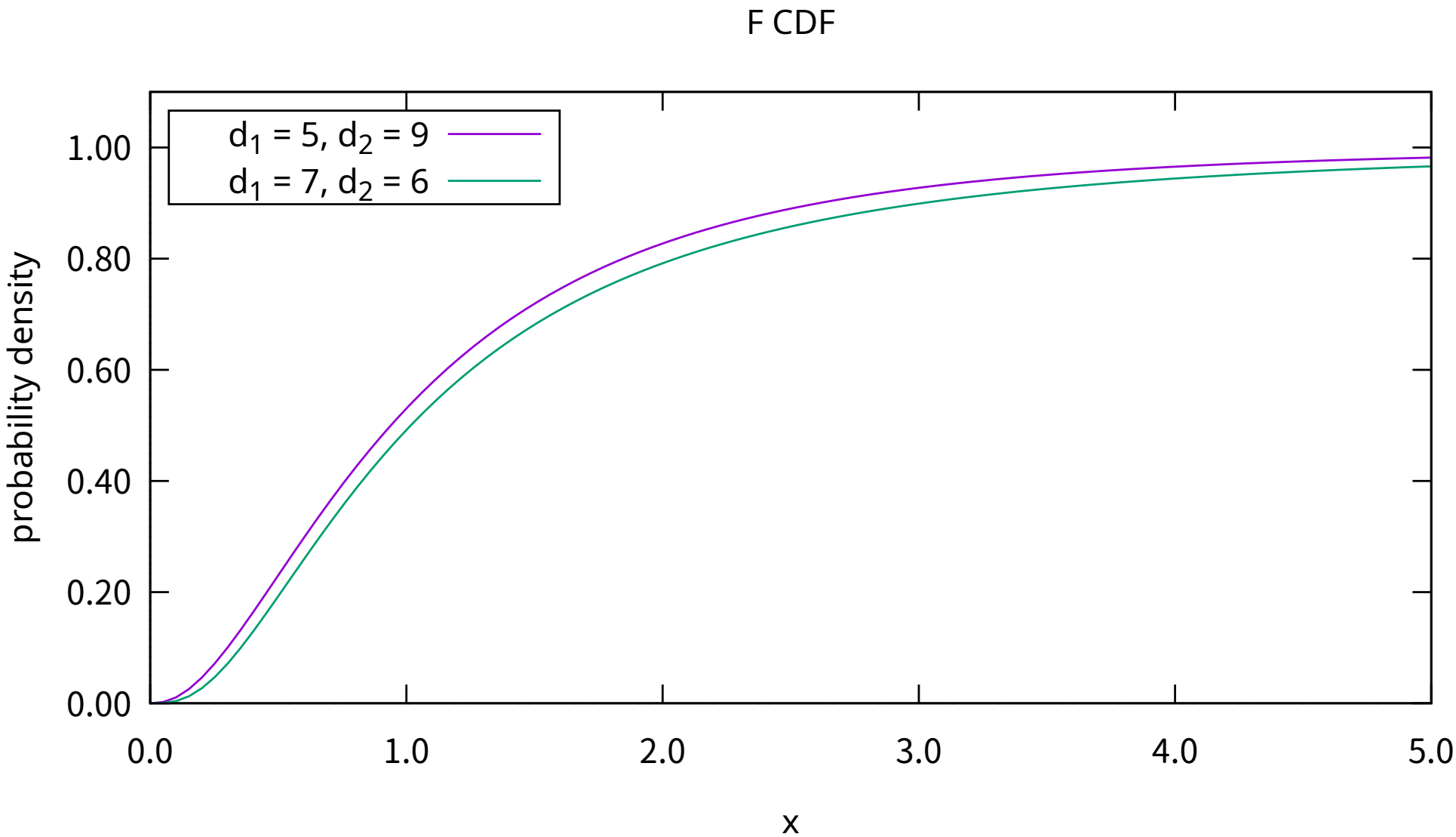

Erlang PDF
λ = 1.0, n = 1
λ = 0.5, n = 1
λ = 1.0, n = 2
λ = 0.5, n = 2
n = 1, exponential r.v.
probability density
x
1.0
0.8
0.6
0.4
0.2
0.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0

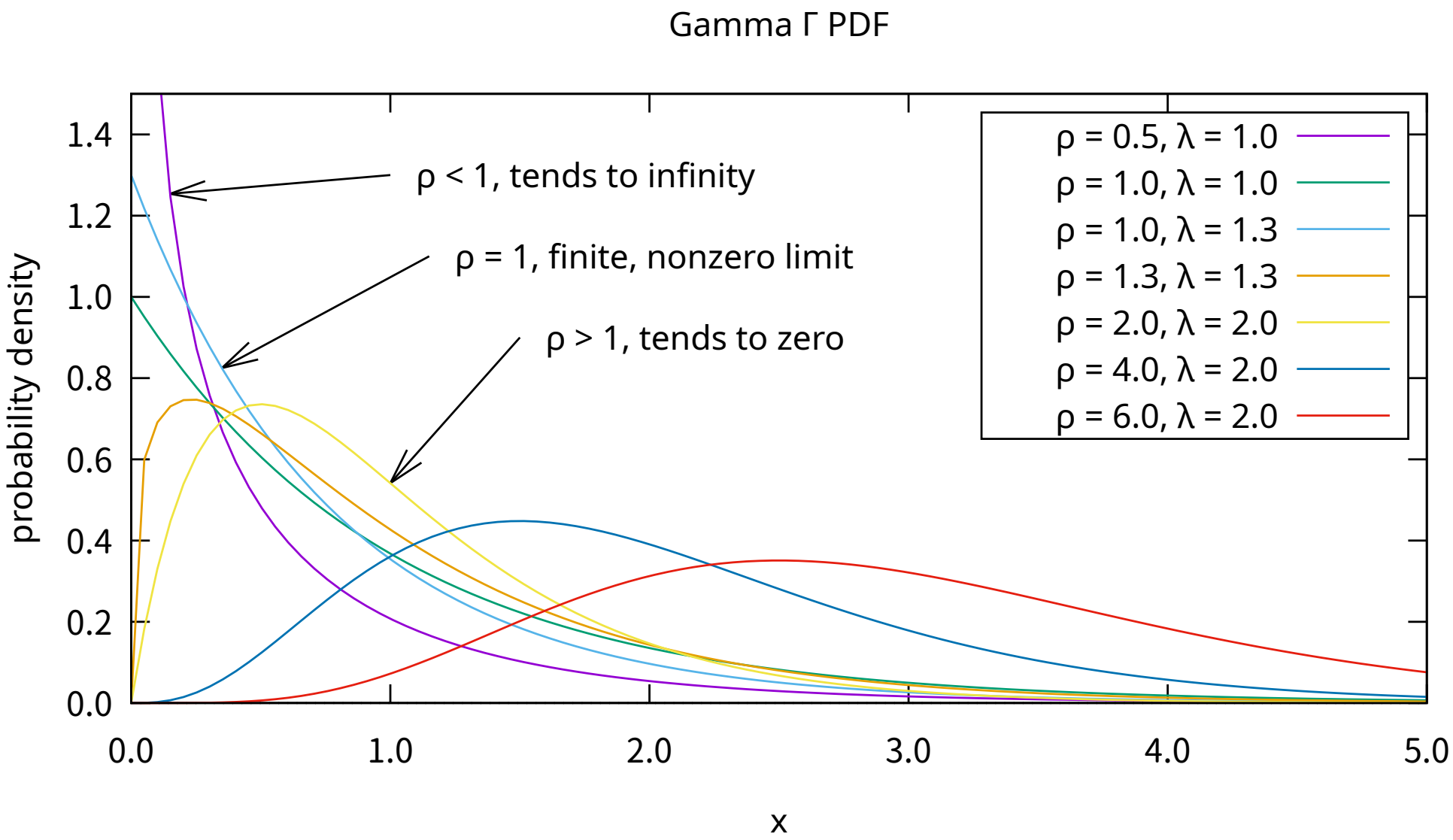

Erlang CDF
probability density
1.0
0.8
0.6
0.4
0.2
0.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
x
λ = 1.0, n = 1
λ = 0.5, n = 1
λ = 1.0, n = 2
λ = 0.5, n = 2

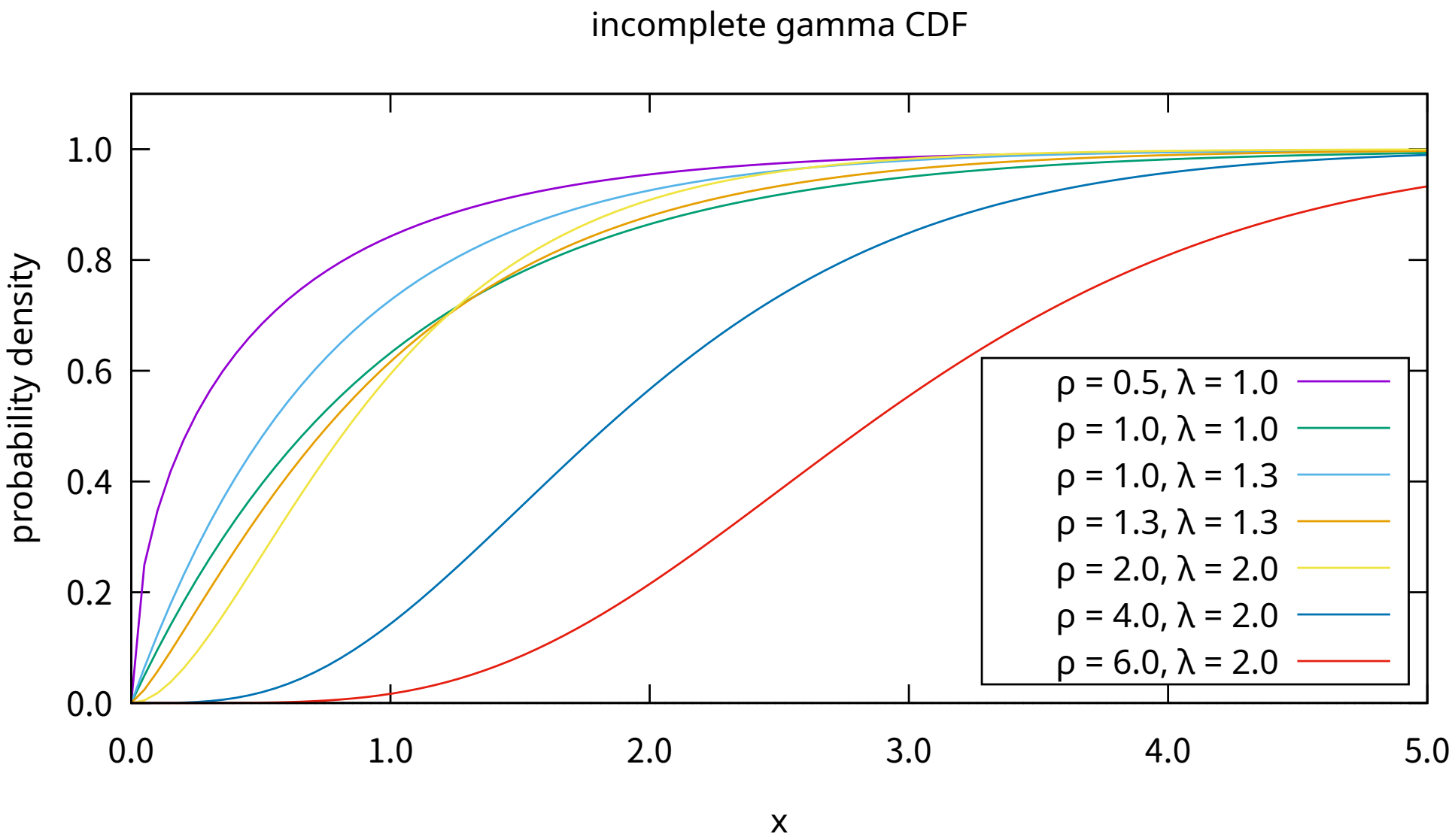

extreme PDF
α = 0.5, u = 1.0
α = 1.0, u = 0.0
probability density
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-6.0
-4.0
-2.0
0.0
2.0
4.0
6.0
8.0
x

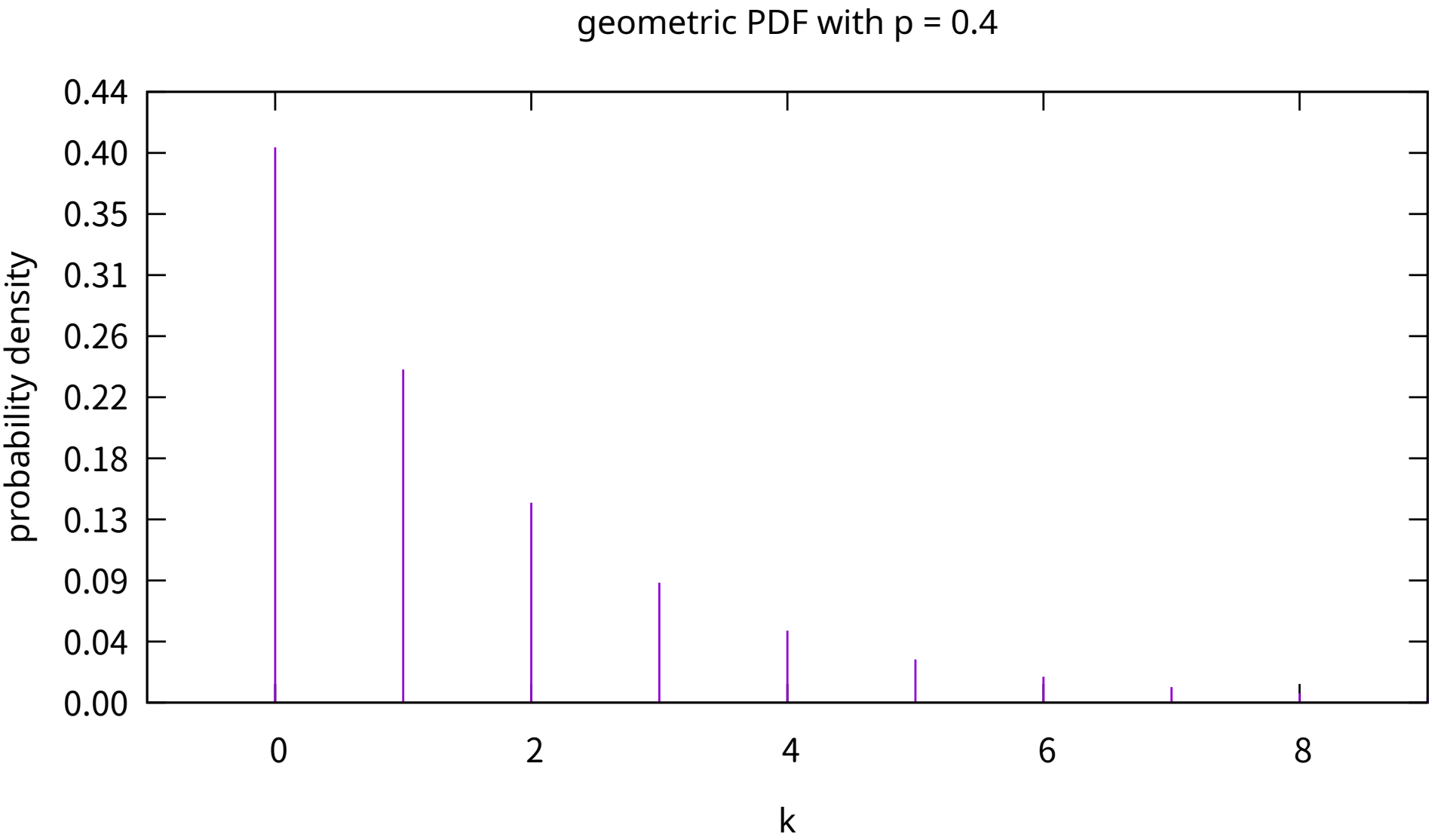

extreme CDF
α = 0.5, u = 1.0
α = 1.0, u = 0.0
probability density
1.00
0.80
0.60
0.40
0.20
0.00
-6.0
-4.0
-2.0
0.0
2.0
4.0
6.0
8.0
x

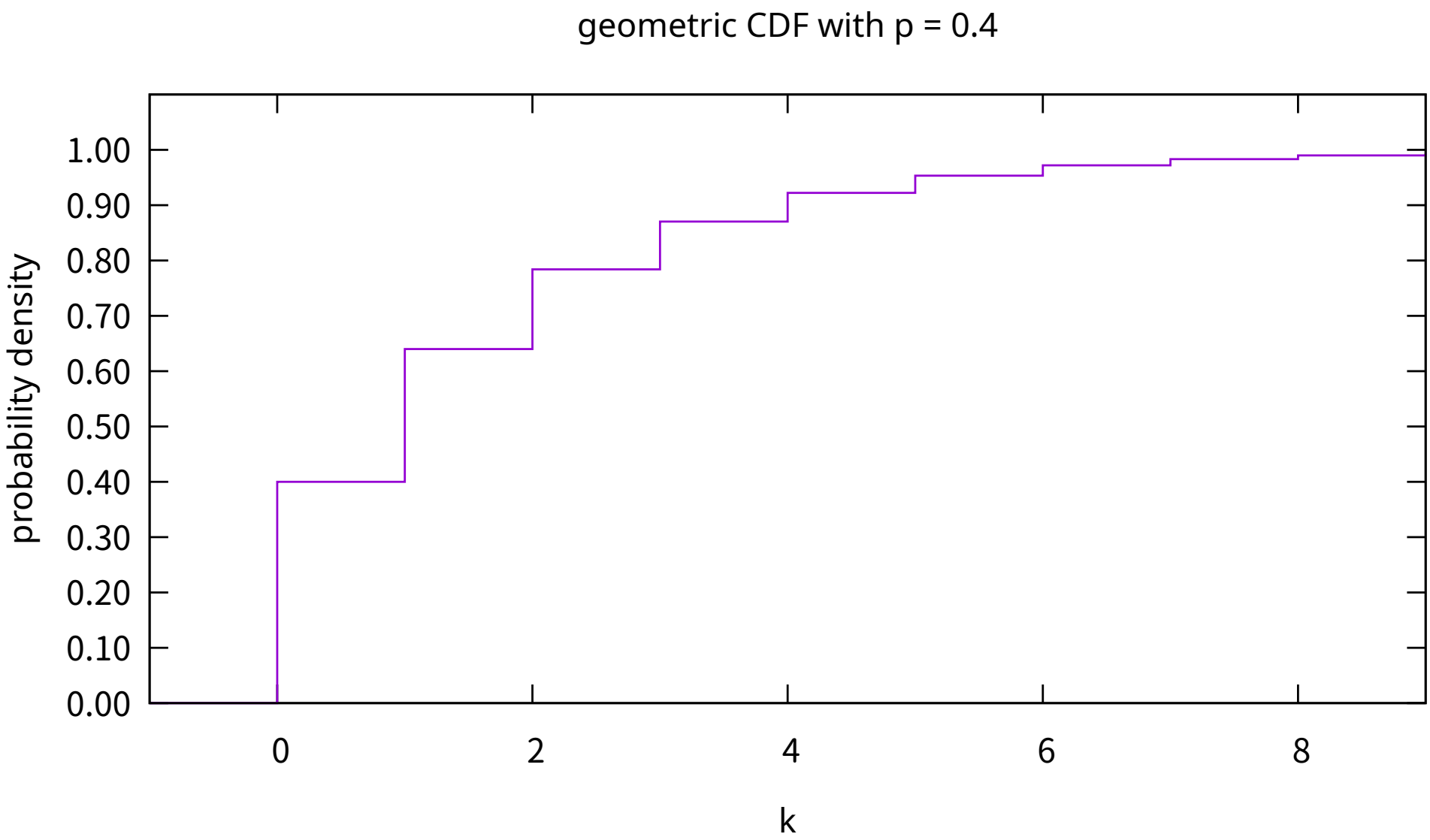

F PDF
probability density
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0.0
1.0
2.0
3.0
4.0
5.0
x
d1 = 5, d2 = 9
d1 = 7, d2 = 6

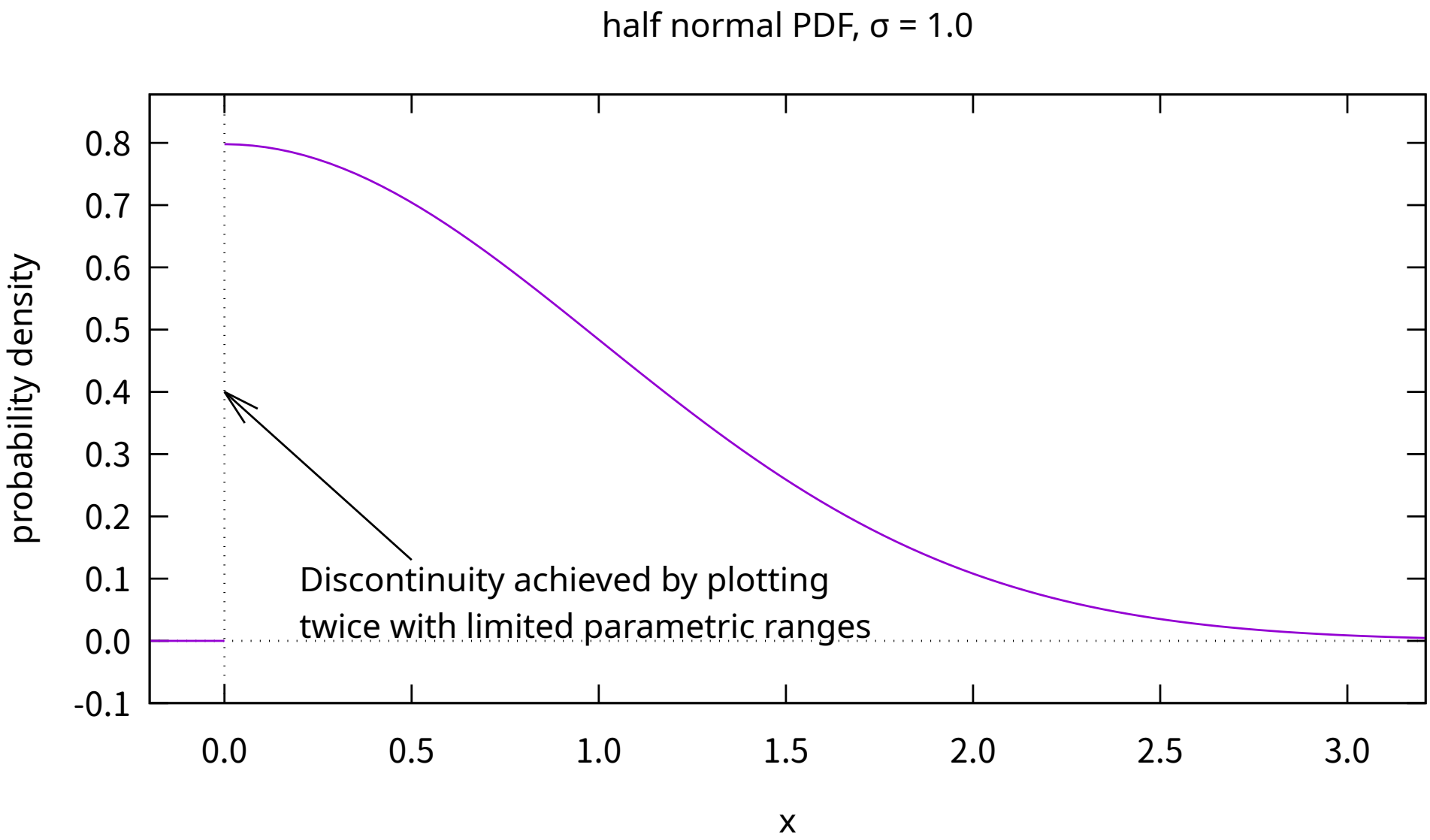

F CDF
d1 = 5, d2 = 9
d1 = 7, d2 = 6
probability density
1.00
0.80
0.60
0.40
0.20
0.00
0.0
1.0
2.0
3.0
4.0
5.0
x

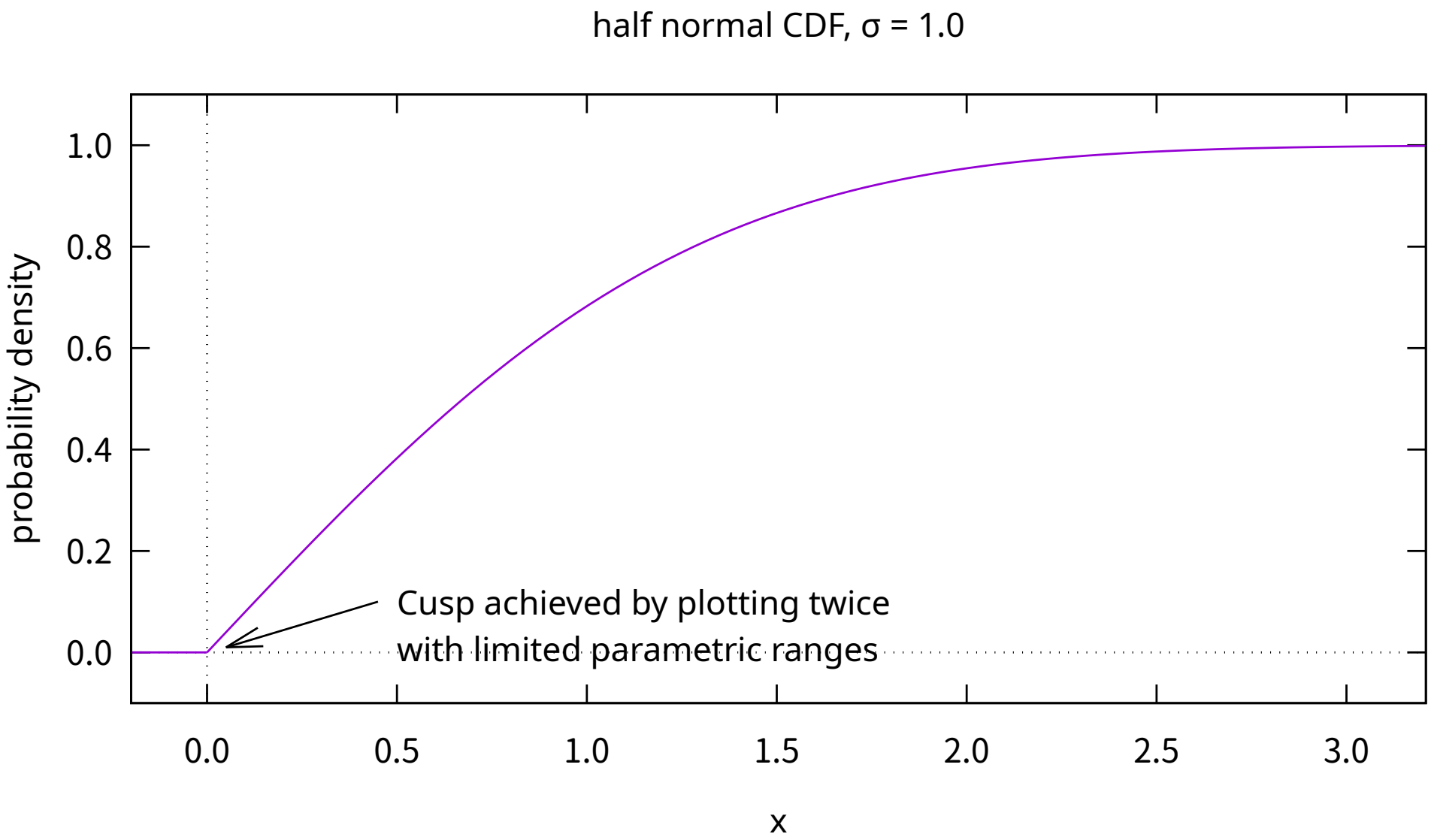

Gamma Γ PDF
ρ < 1, tends to infinity
ρ = 1, finite, nonzero limit
ρ > 1, tends to zero
ρ = 0.5, λ = 1.0
ρ = 1.0, λ = 1.0
ρ = 1.0, λ = 1.3
ρ = 1.3, λ = 1.3
ρ = 2.0, λ = 2.0
ρ = 4.0, λ = 2.0
ρ = 6.0, λ = 2.0
probability density
x
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0.0
1.0
2.0
3.0
4.0
5.0

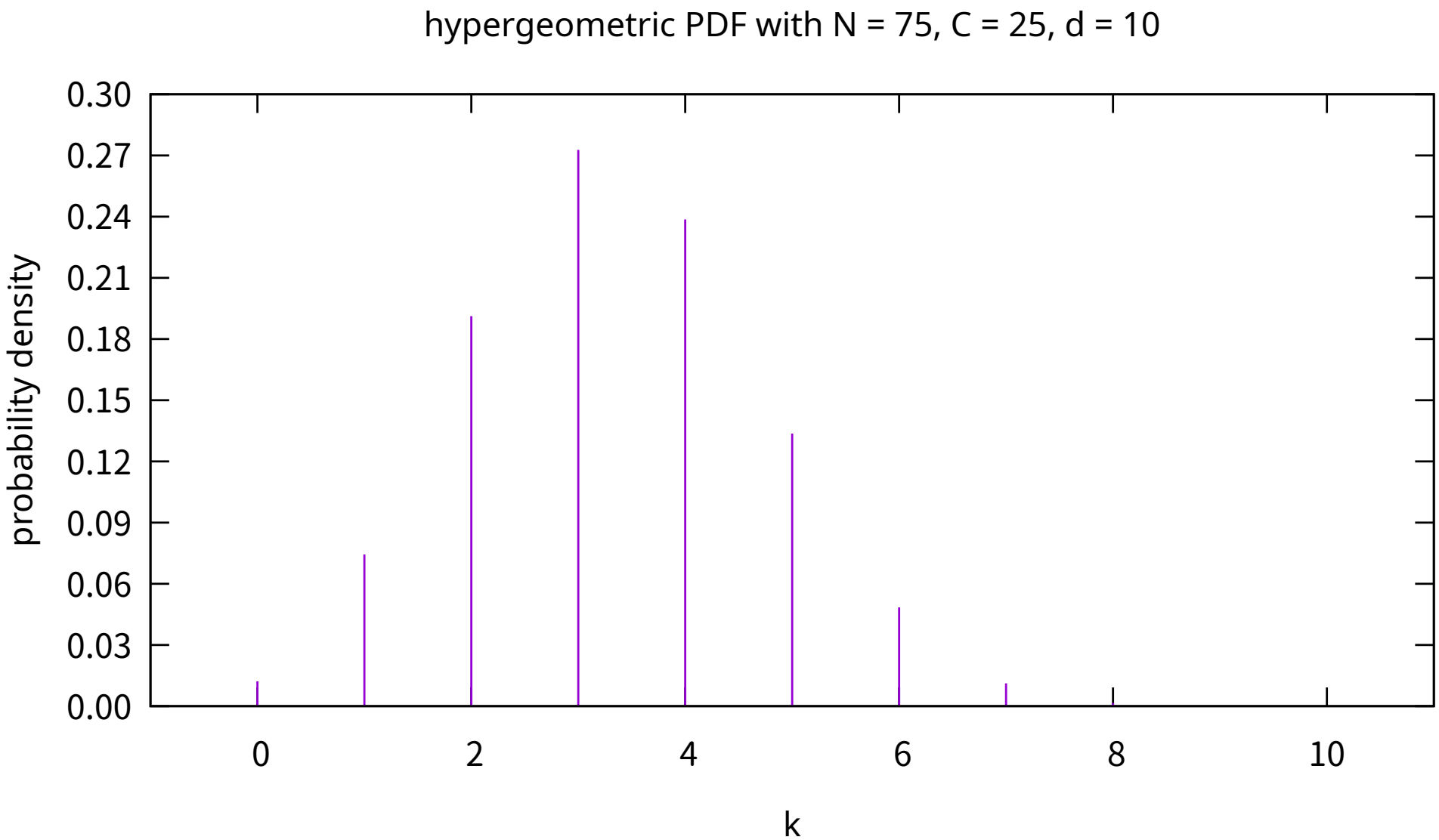

incomplete gamma CDF
probability density
x
0.0
0.2
0.4
0.6
0.8
1.0
0.0
1.0
2.0
3.0
4.0
5.0
ρ = 0.5, λ = 1.0
ρ = 1.0, λ = 1.0
ρ = 1.0, λ = 1.3
ρ = 1.3, λ = 1.3
ρ = 2.0, λ = 2.0
ρ = 4.0, λ = 2.0
ρ = 6.0, λ = 2.0

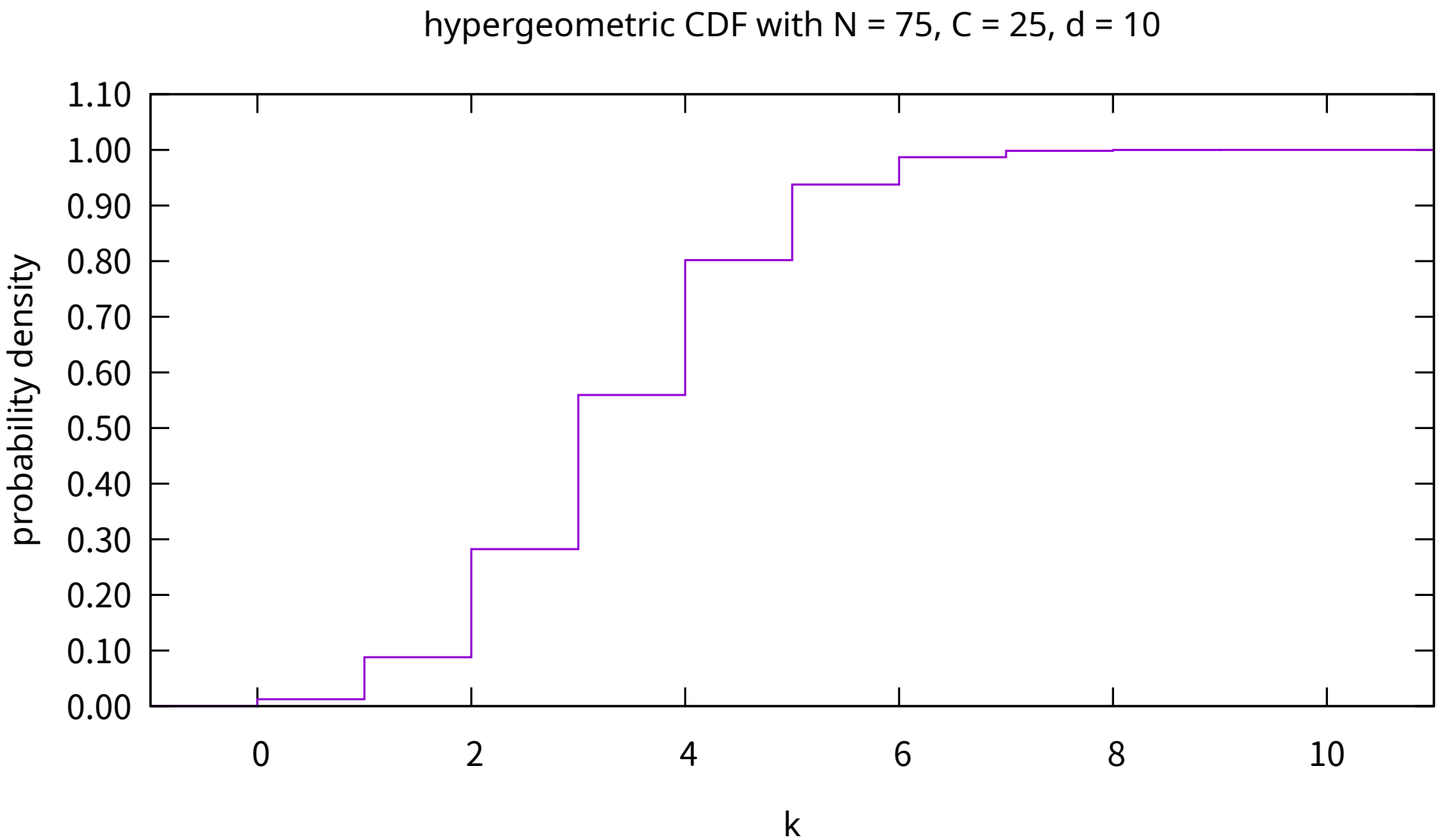

geometric PDF with p = 0.4
probability density
0.44
0.40
0.35
0.31
0.26
0.22
0.18
0.13
0.09
0.04
0.00
0
2
4
6
8
k

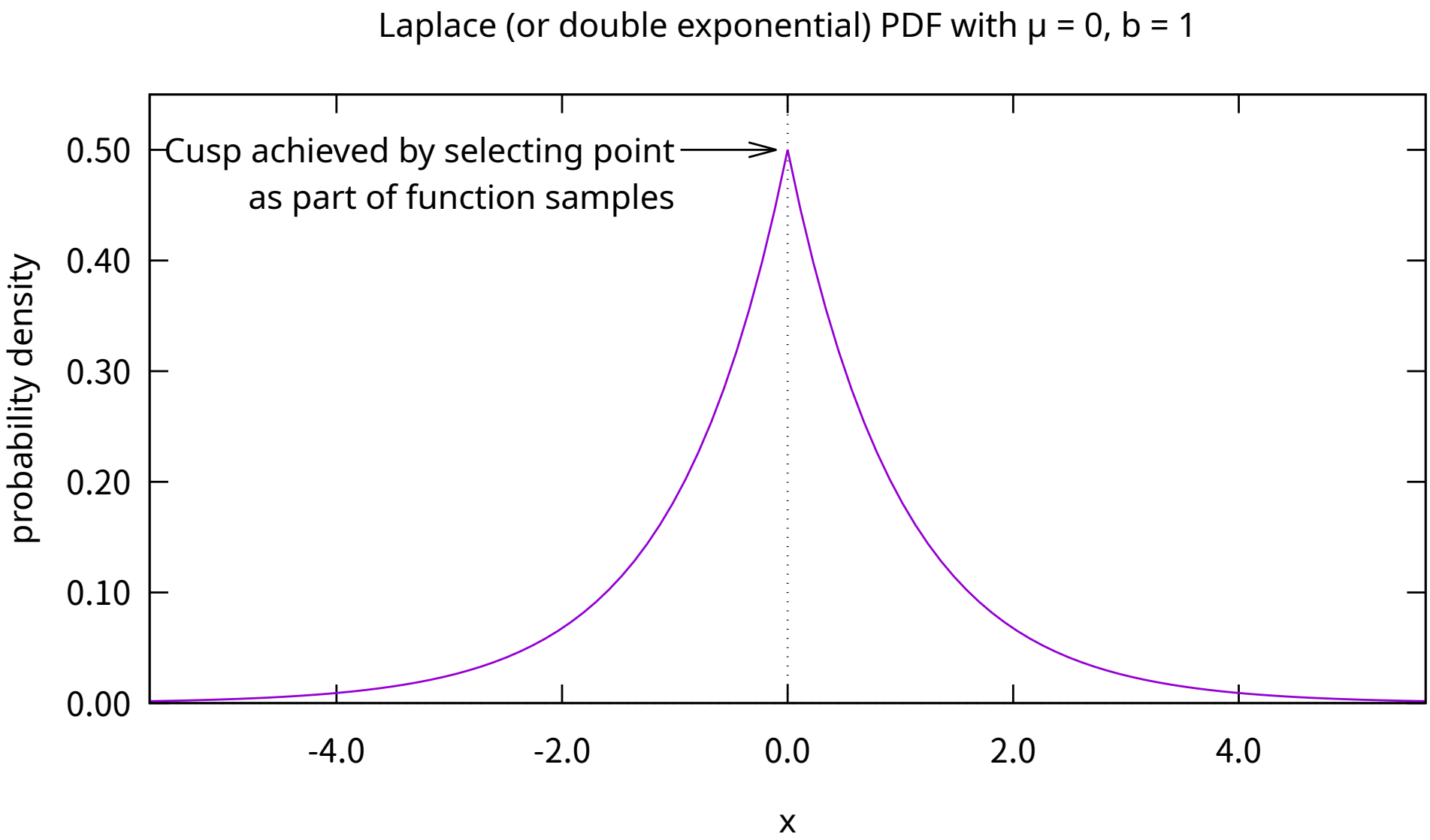

geometric CDF with p = 0.4
probability density
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0
2
4
6
8
k

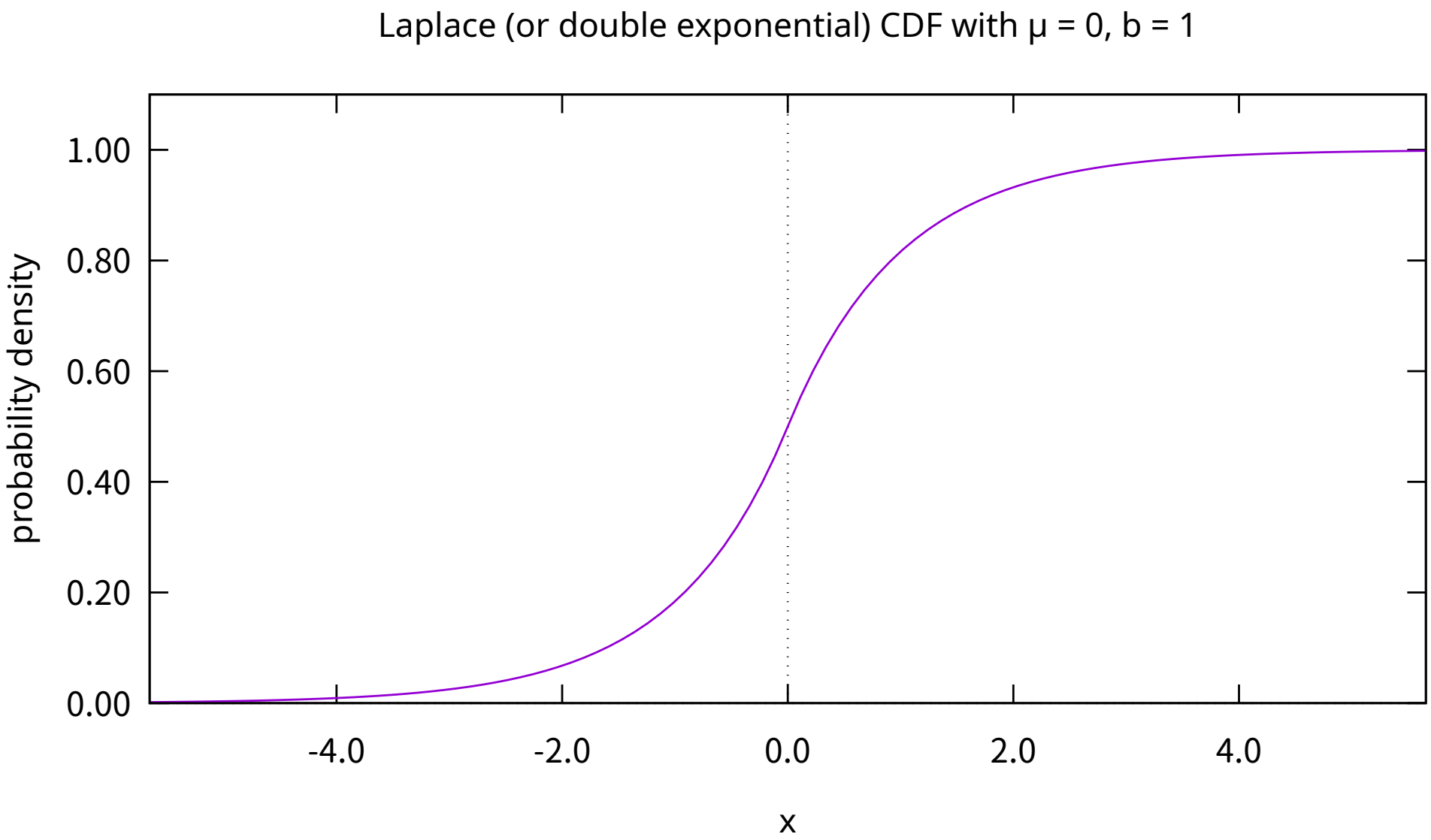

half normal PDF, σ = 1.0
probability density
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
0.0
0.5
1.0
1.5
2.0
2.5
3.0
x
Discontinuity achieved by plotting
twice with limited parametric ranges

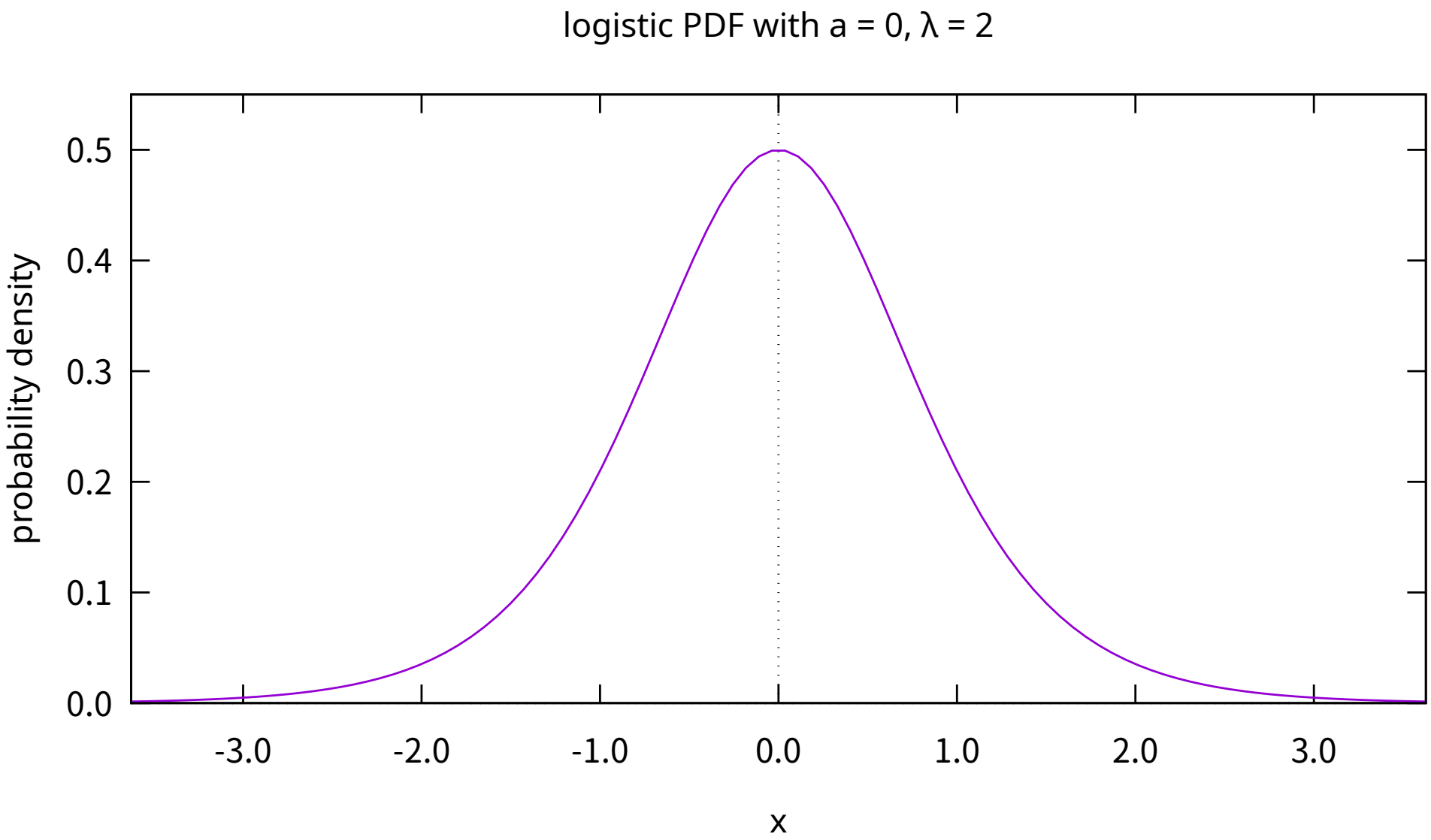

half normal CDF, σ = 1.0
probability density
1.0
0.8
0.6
0.4
0.2
0.0
Cusp achieved by plotting twice
with limited parametric ranges
0.0
0.5
1.0
1.5
2.0
2.5
3.0
x

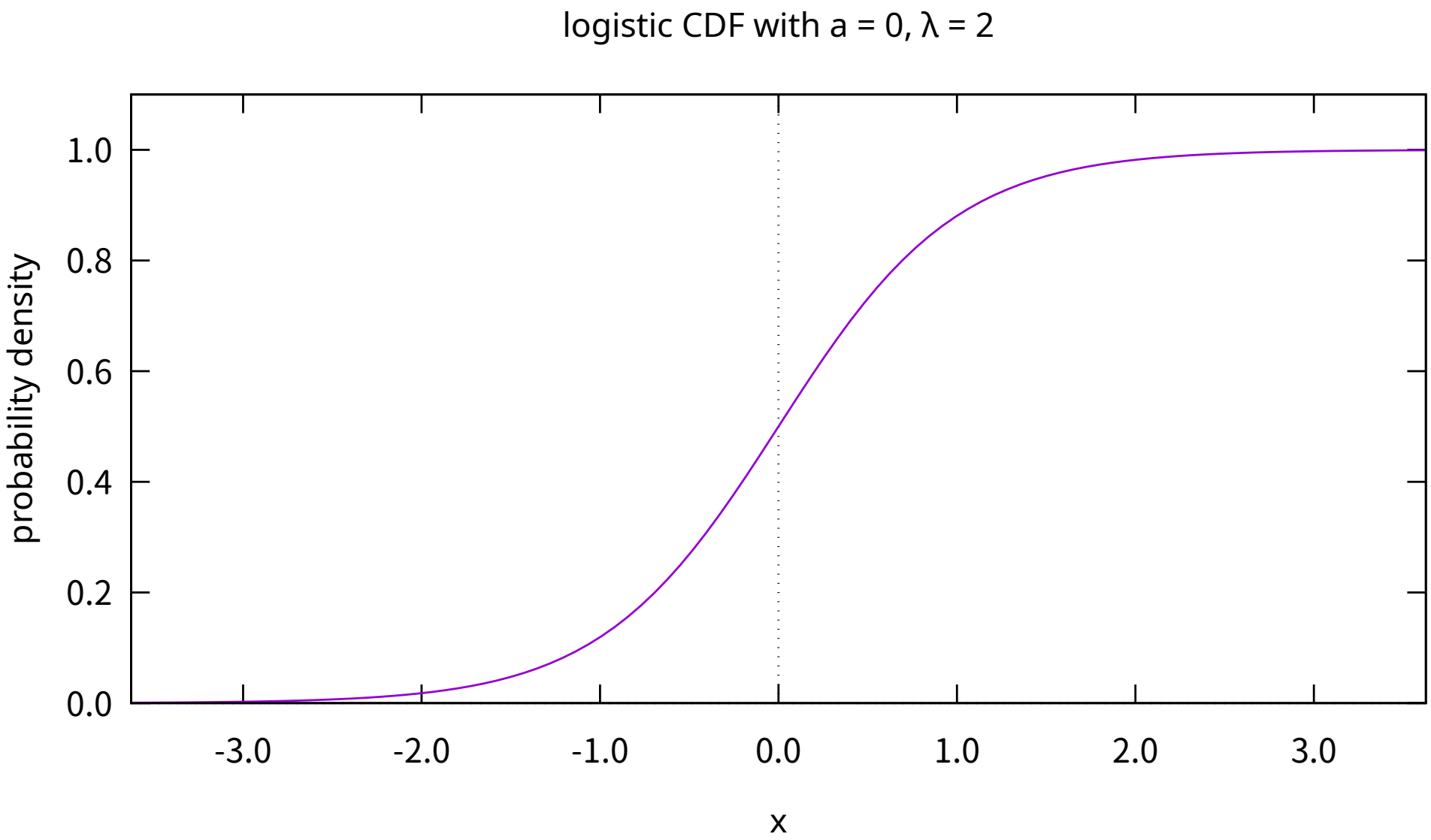

hypergeometric PDF with N = 75, C = 25, d = 10
probability density
0.30
0.27
0.24
0.21
0.18
0.15
0.12
0.09
0.06
0.03
0.00
0
2
4
6
8
10
k

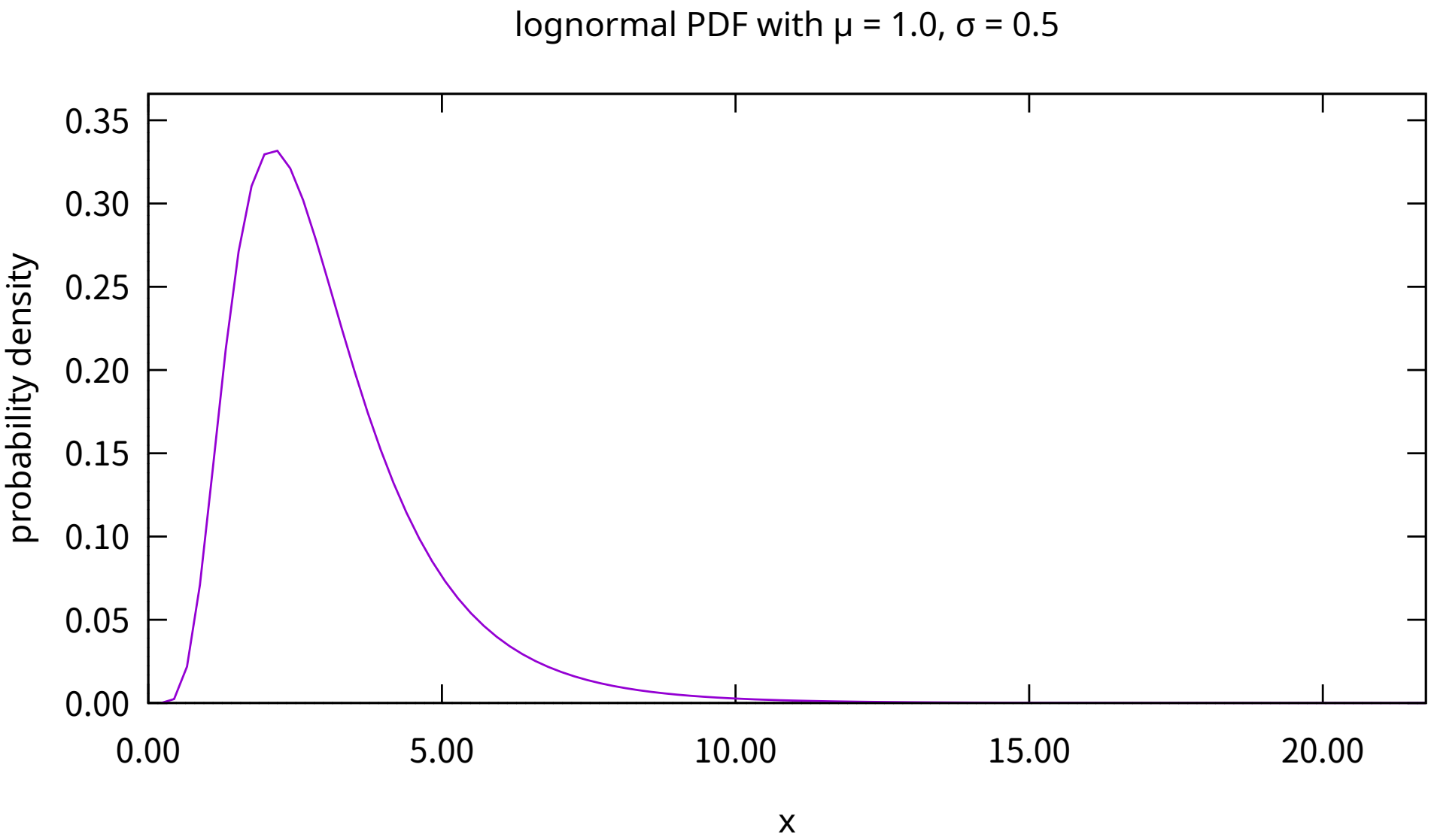

hypergeometric CDF with N = 75, C = 25, d = 10
probability density
1.10
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0
2
4
6
8
10
k

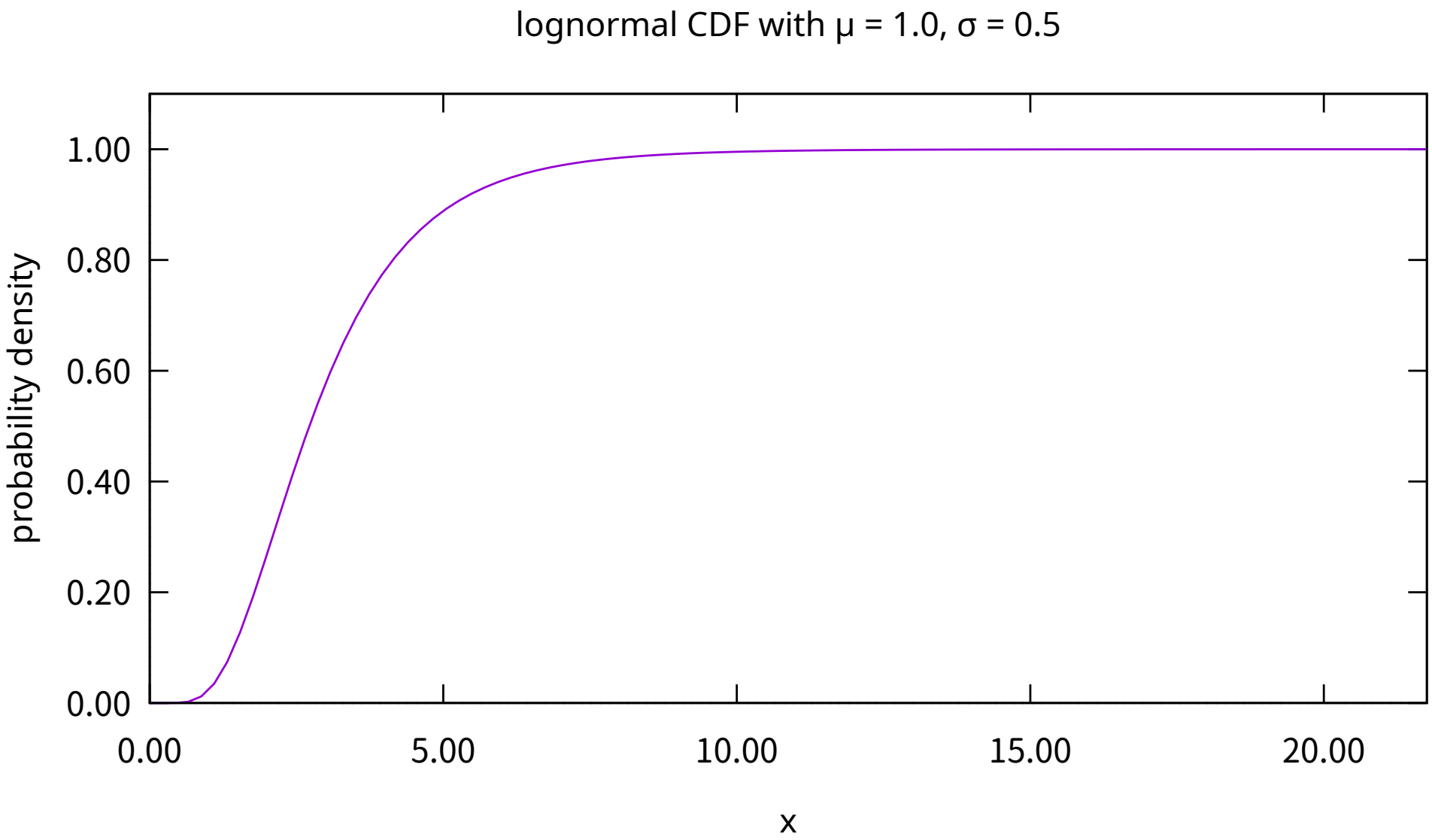

Laplace (or double exponential) PDF with μ = 0, b = 1
Cusp achieved by selecting point
as part of function samples
probability density
0.50
0.40
0.30
0.20
0.10
0.00
-4.0
-2.0
0.0
2.0
4.0
x

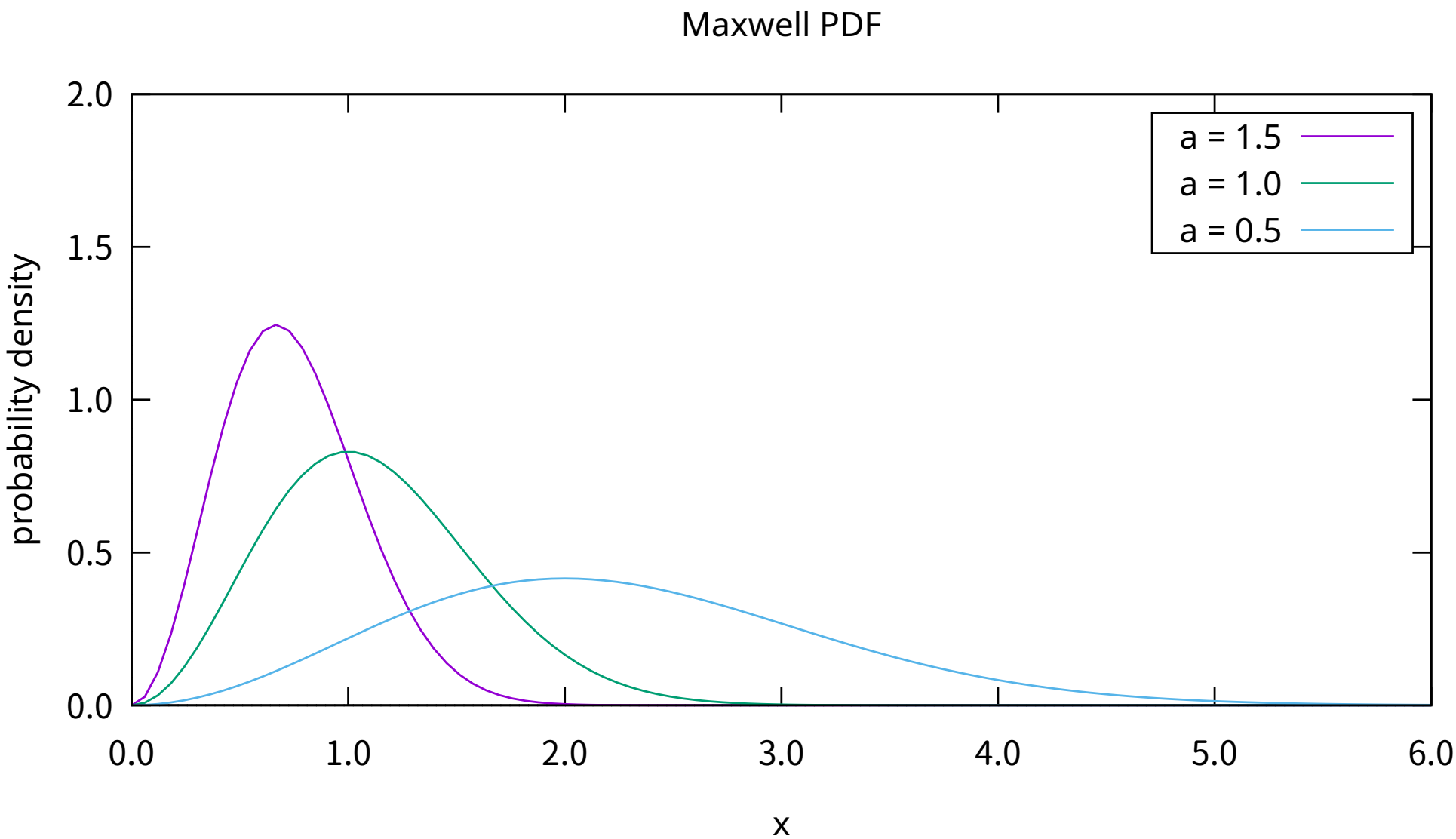

Laplace (or double exponential) CDF with μ = 0, b = 1
probability density
1.00
0.80
0.60
0.40
0.20
0.00
-4.0
-2.0
0.0
2.0
4.0
x

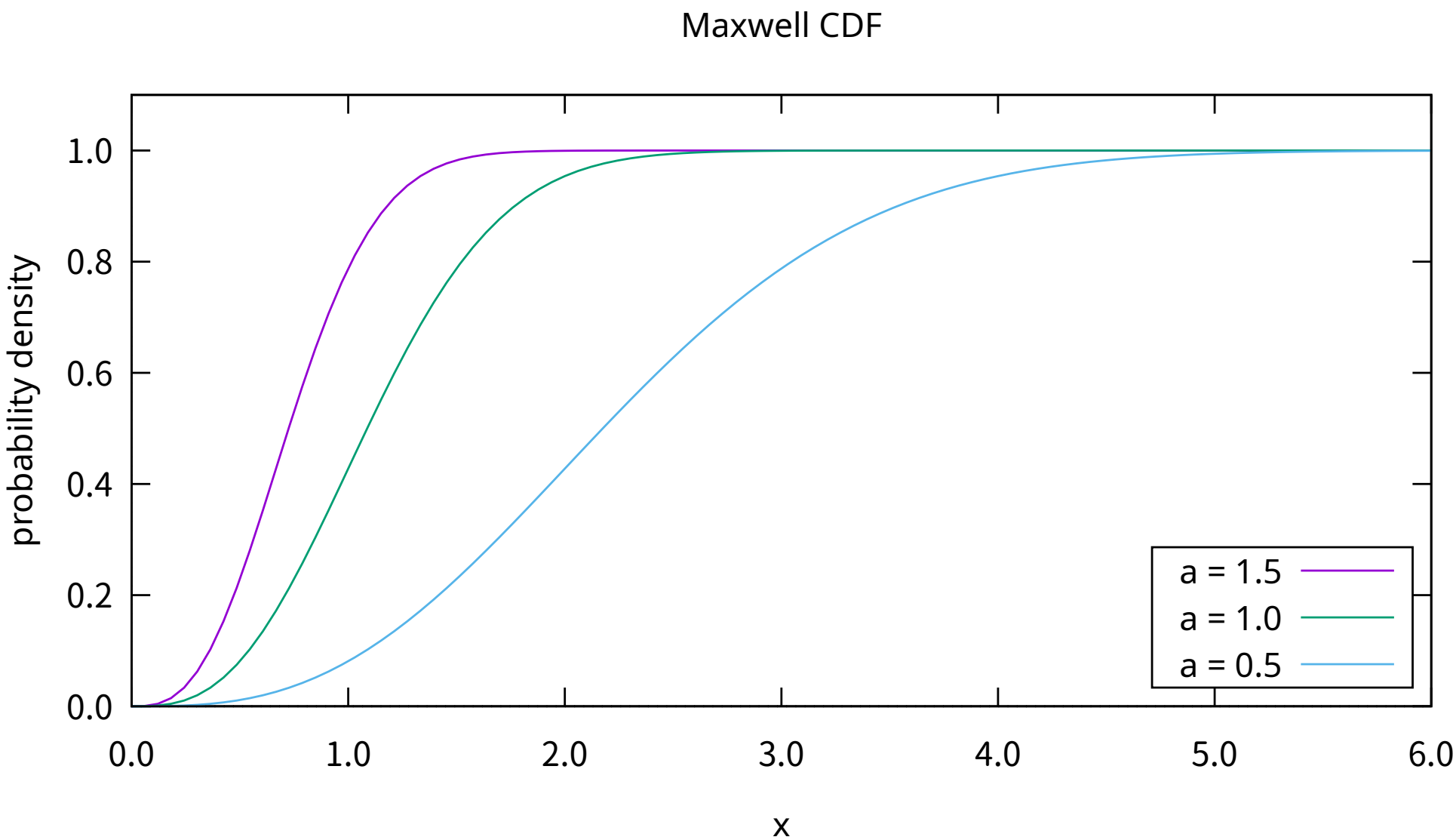

logistic PDF with a = 0, λ = 2
probability density
0.5
0.4
0.3
0.2
0.1
0.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
x

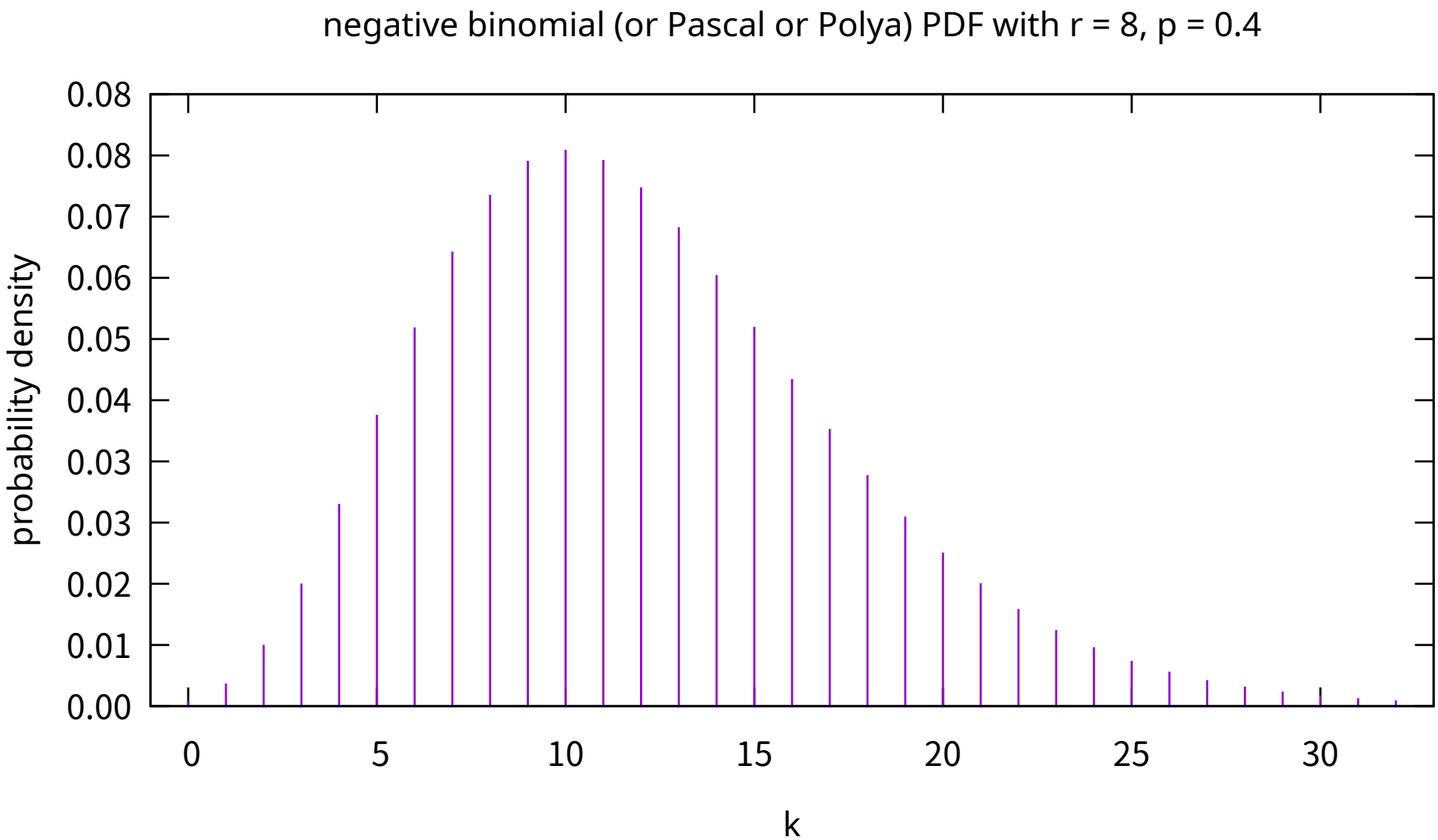

logistic CDF with a = 0, λ = 2
probability density
1.0
0.8
0.6
0.4
0.2
0.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
x

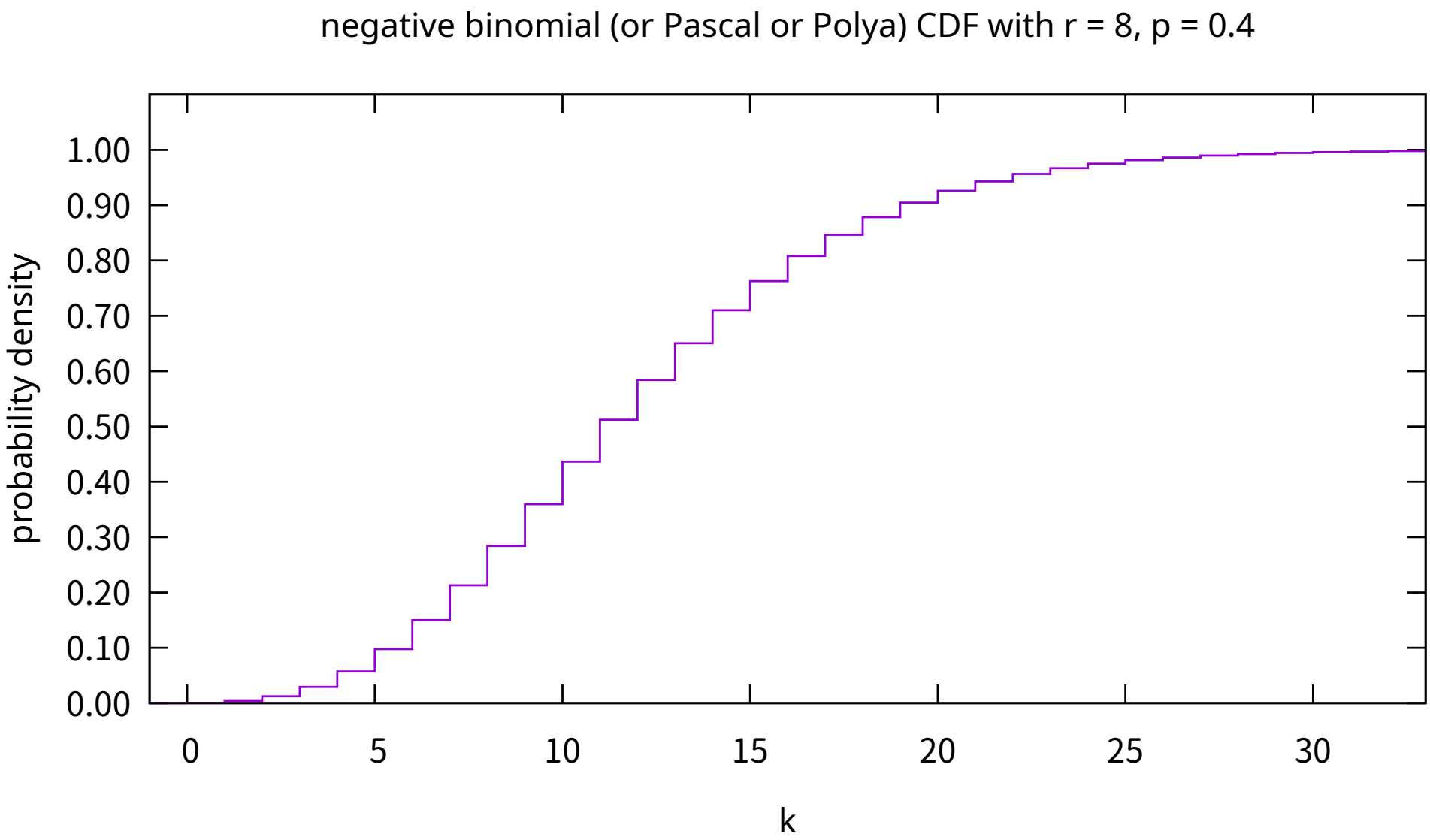

lognormal PDF with μ = 1.0, σ = 0.5
probability density
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.00
5.00
10.00
15.00
20.00
x

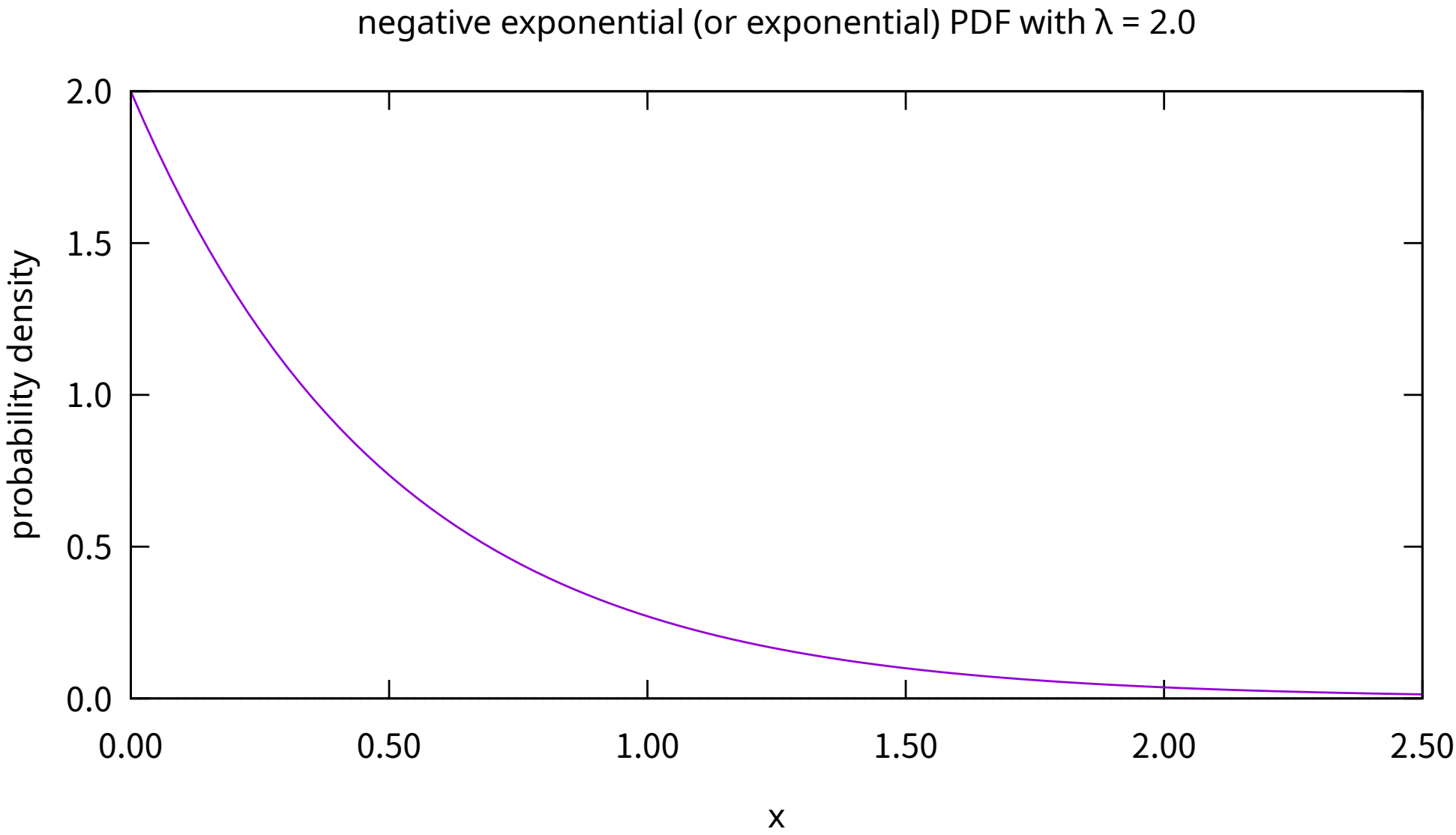

lognormal CDF with μ = 1.0, σ = 0.5
probability density
1.00
0.80
0.60
0.40
0.20
0.00
0.00
5.00
10.00
15.00
20.00
x

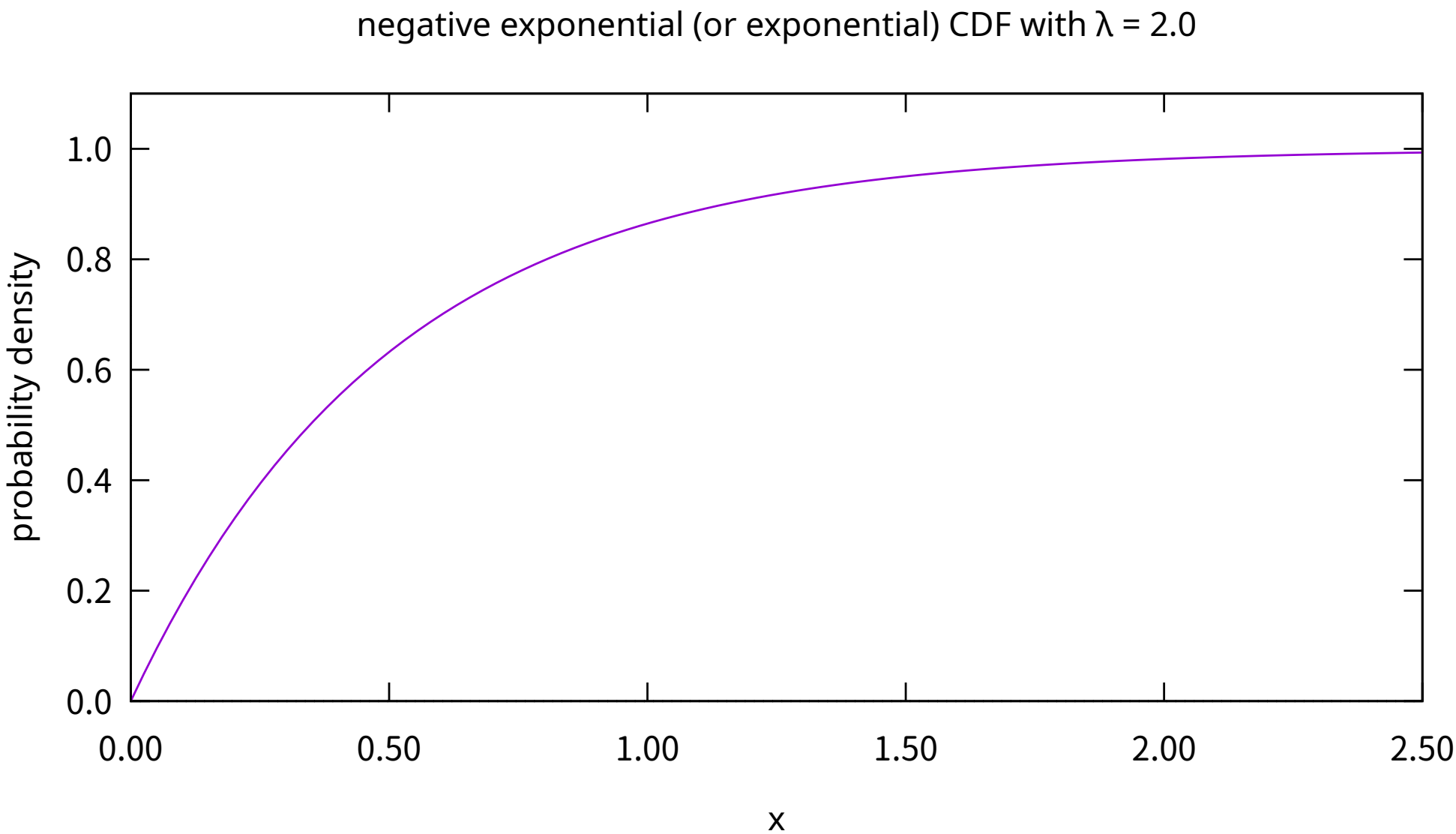

Maxwell PDF
a = 1.5
a = 1.0
a = 0.5
probability density
x
0.0
0.5
1.0
1.5
2.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0

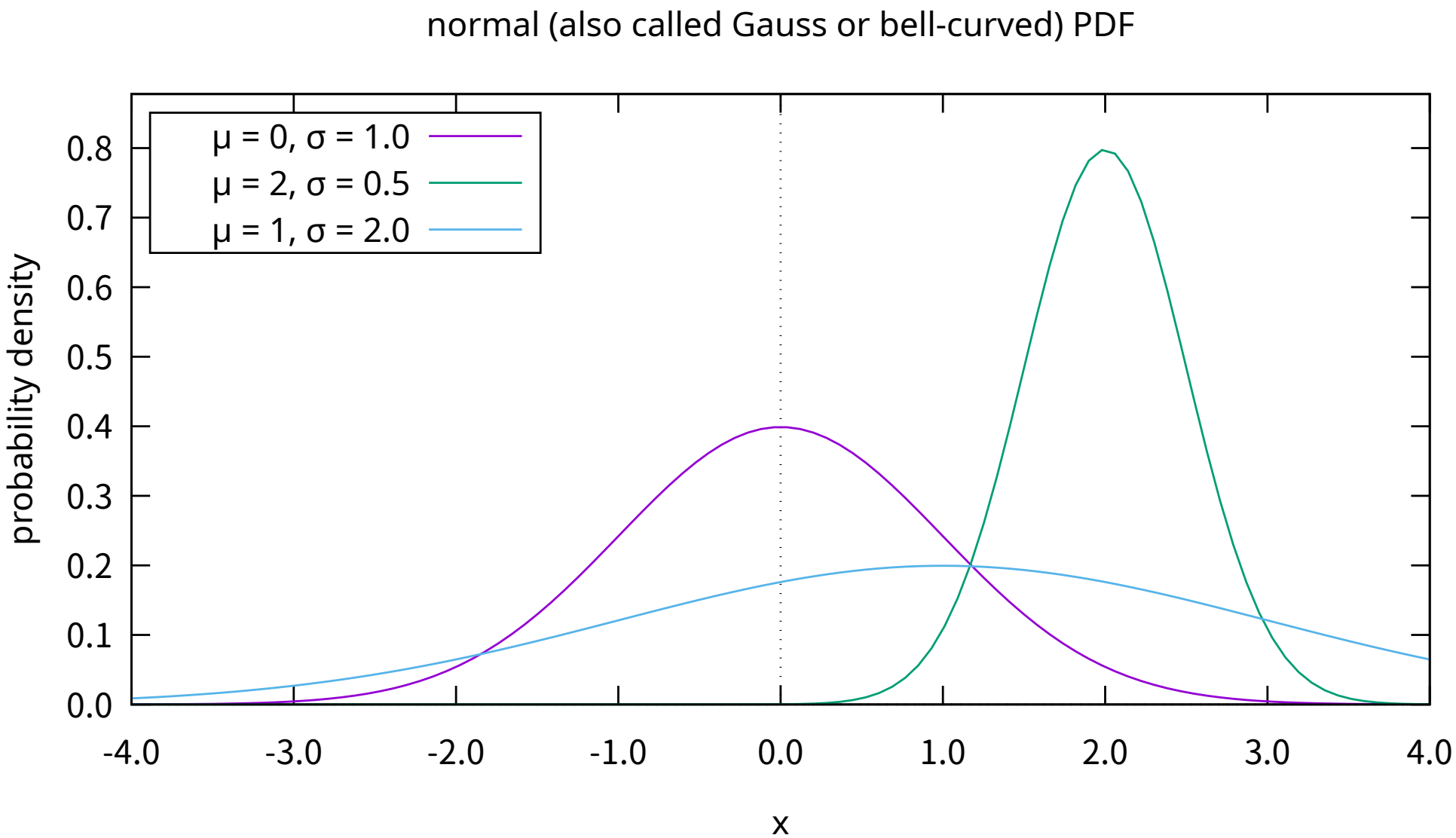

Maxwell CDF
probability density
1.0
0.8
0.6
0.4
0.2
0.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
x
a = 1.5
a = 1.0
a = 0.5

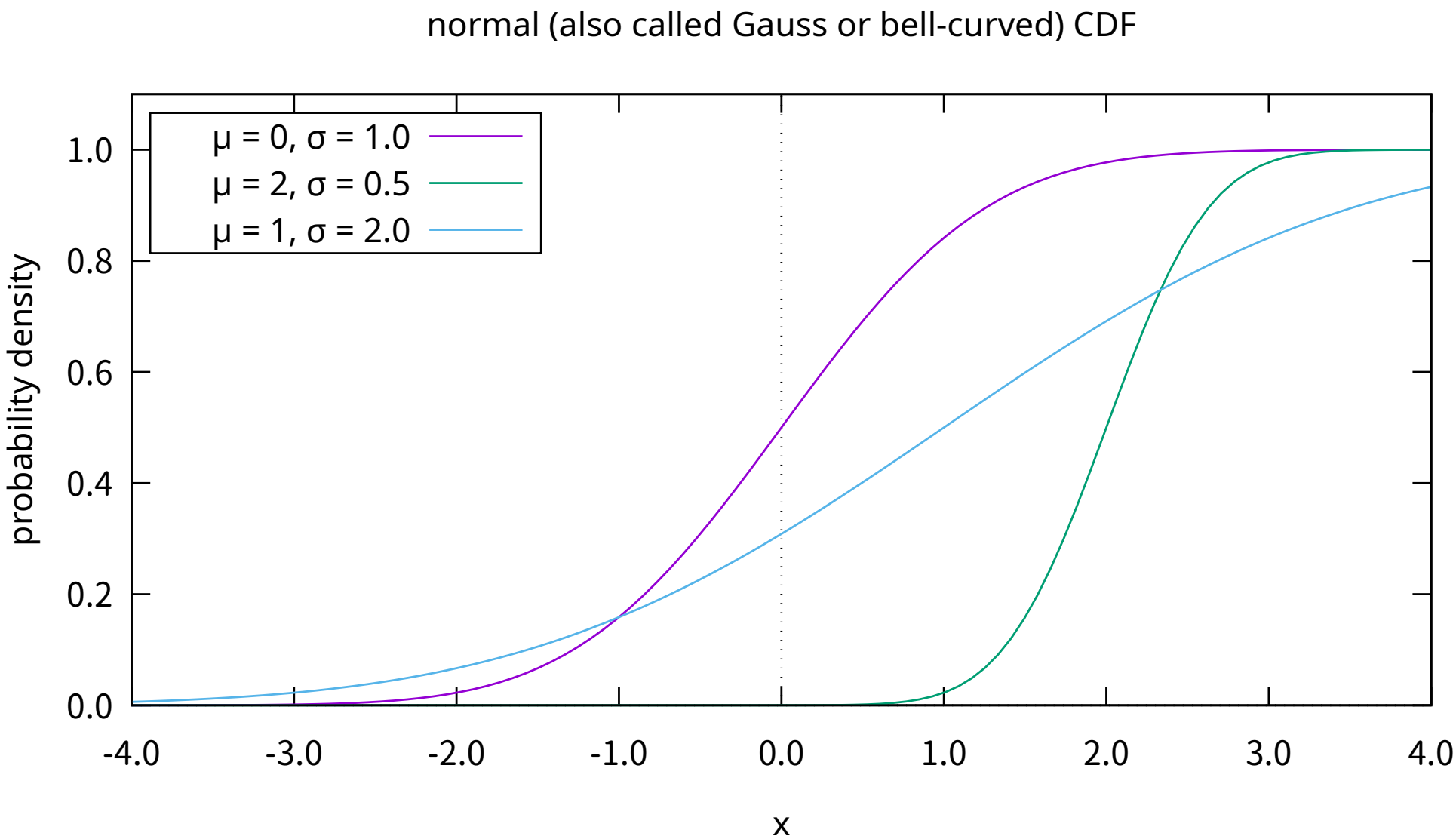

negative binomial (or Pascal or Polya) PDF with r = 8, p = 0.4
probability density
0.08
0.08
0.07
0.06
0.05
0.04
0.03
0.03
0.02
0.01
0.00
0
5
10
15
20
25
30
k

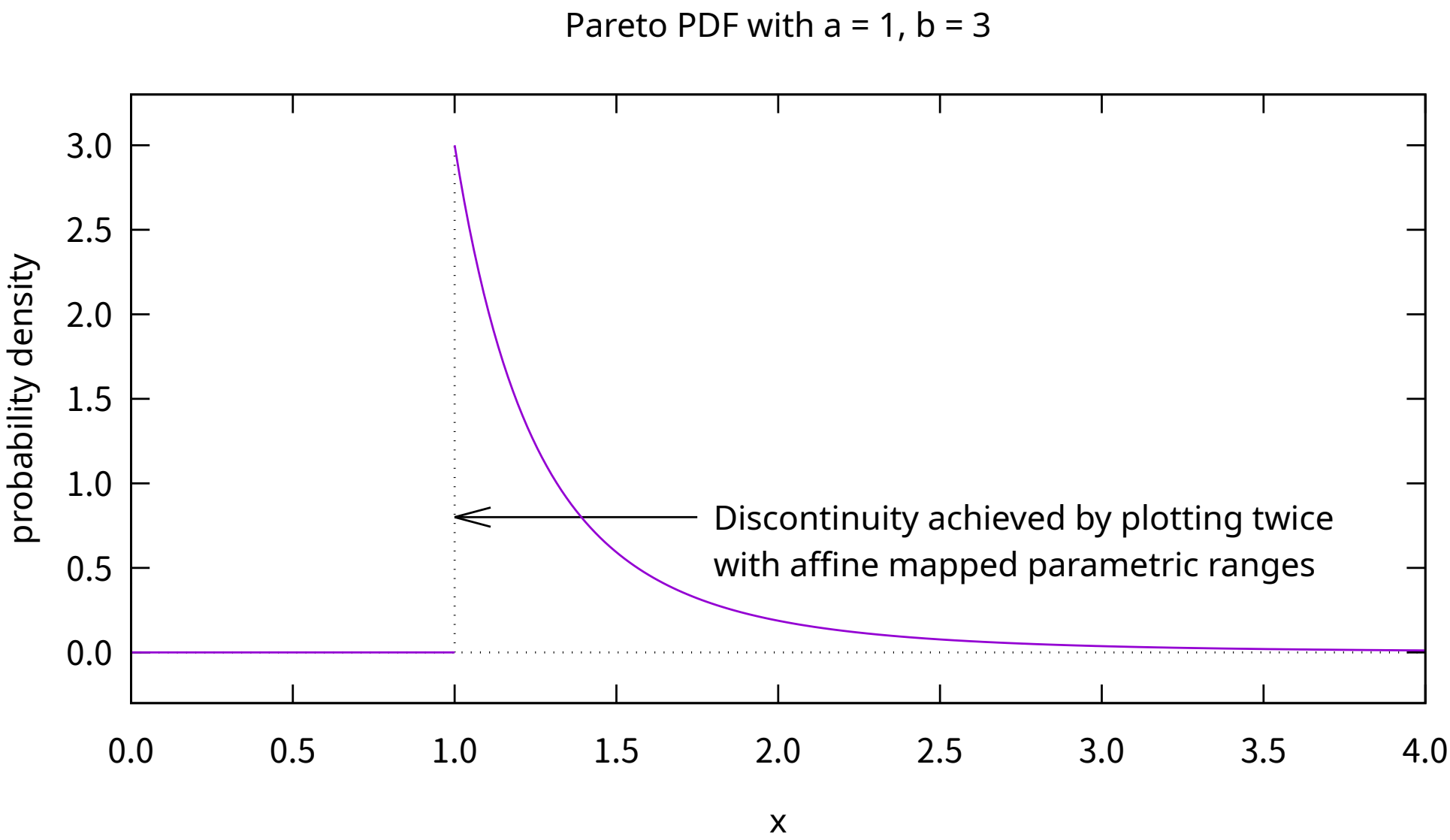

negative binomial (or Pascal or Polya) CDF with r = 8, p = 0.4
probability density
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0
5
10
15
20
25
30
k

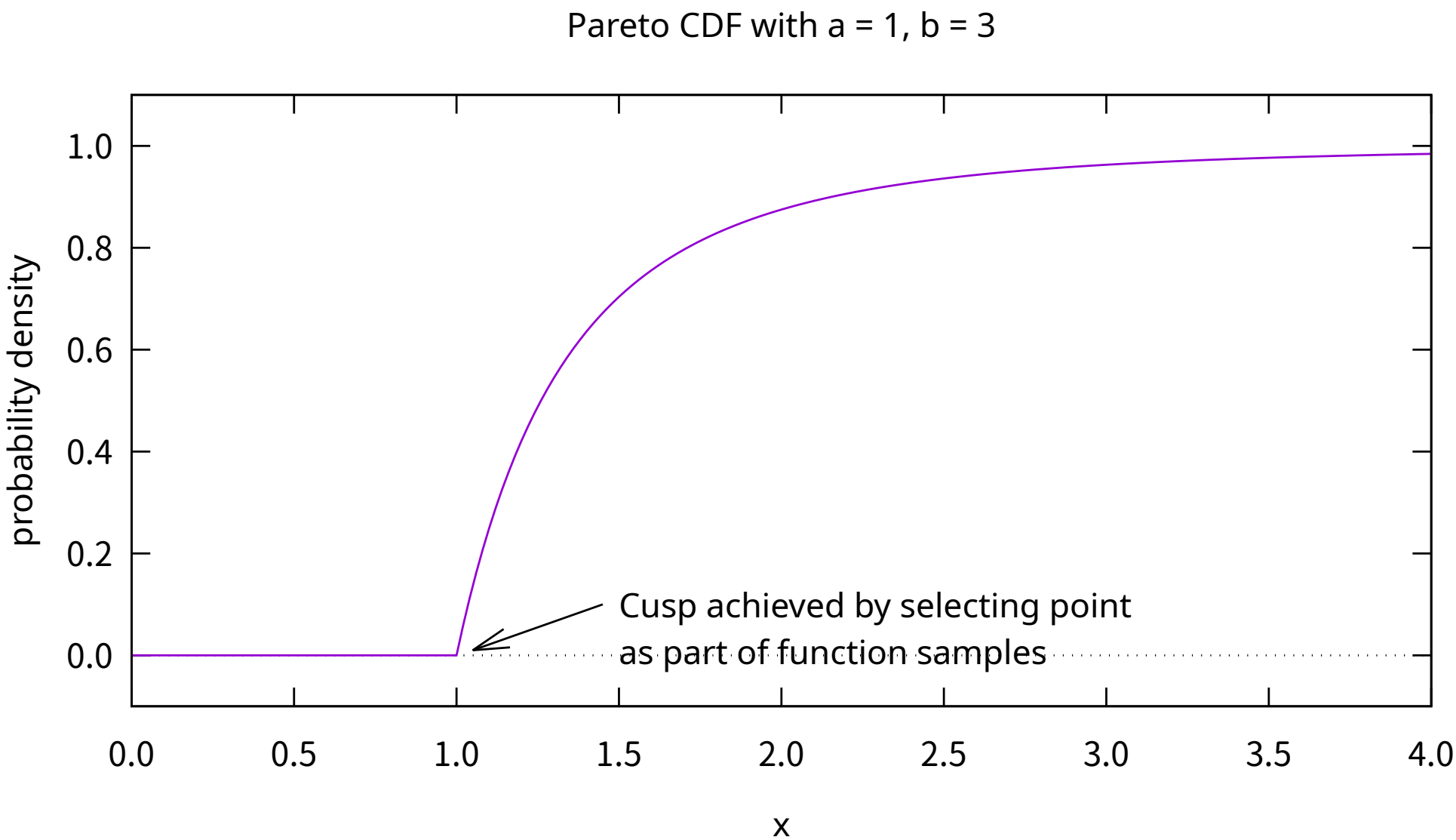

negative exponential (or exponential) PDF with λ = 2.0
probability density
2.0
1.5
1.0
0.5
0.0
0.00
0.50
1.00
1.50
2.00
2.50
x

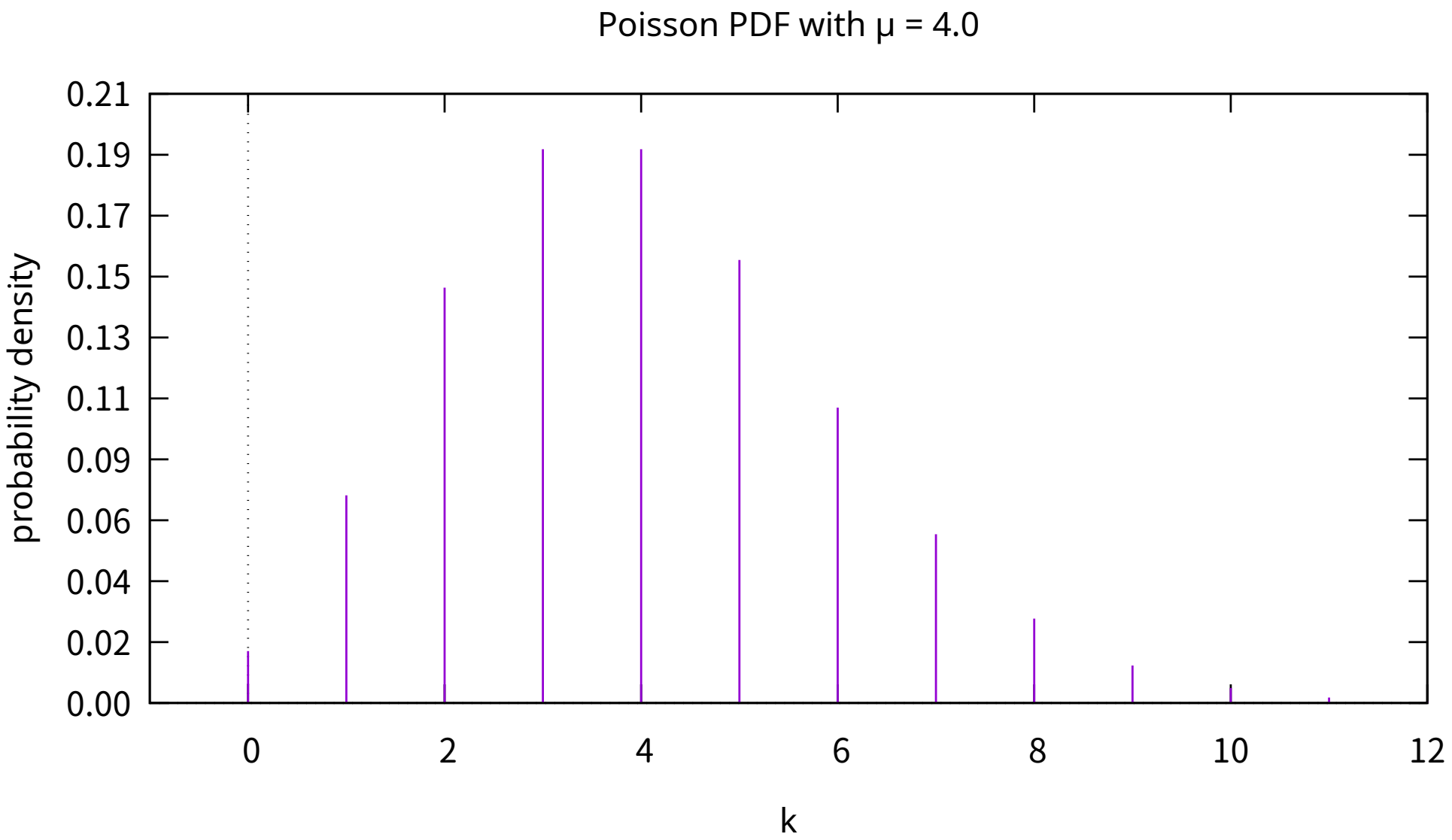

negative exponential (or exponential) CDF with λ = 2.0
probability density
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.50
1.00
1.50
2.00
2.50
x

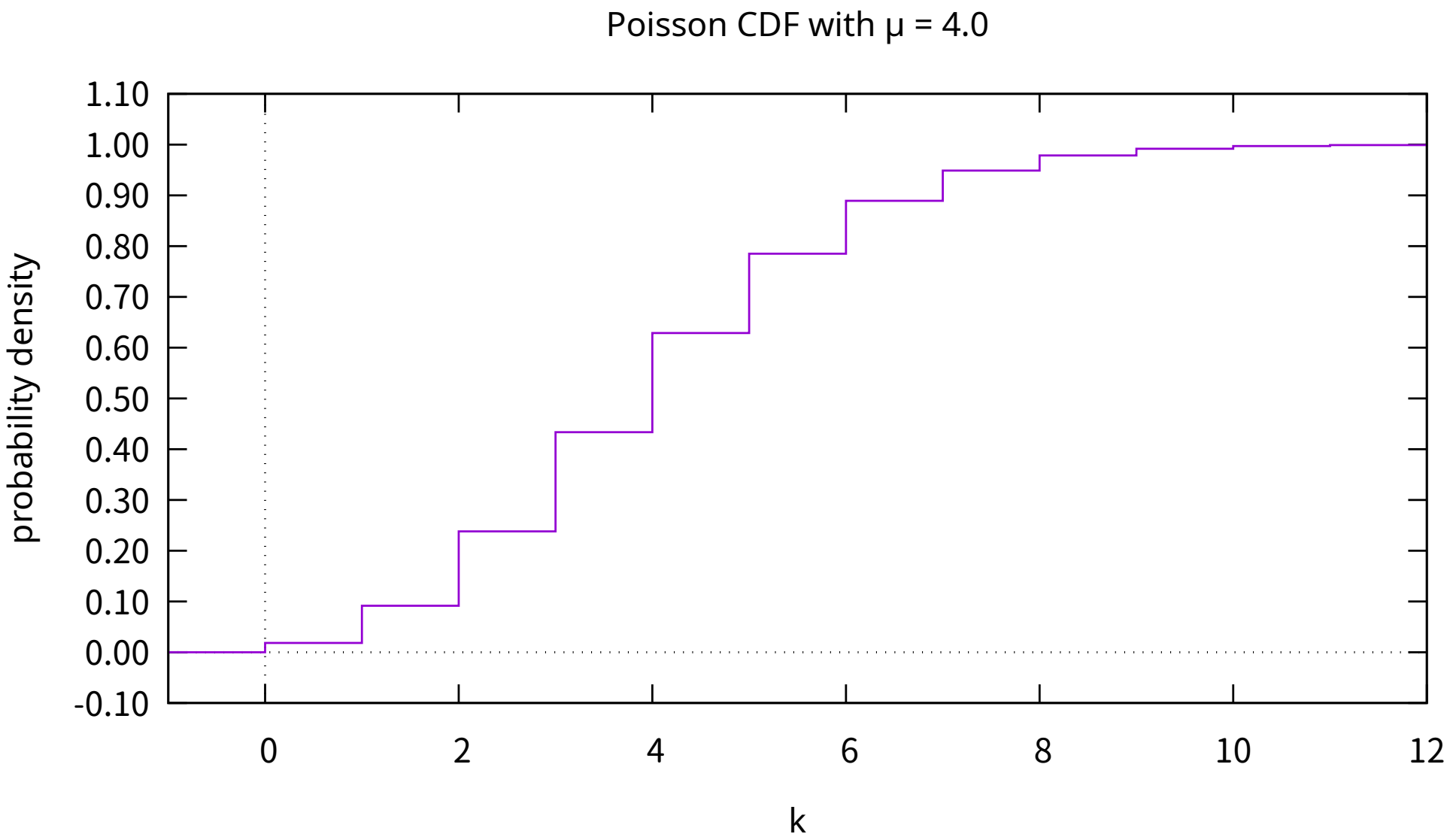

normal (also called Gauss or bell-curved) PDF
μ = 0, σ = 1.0
μ = 2, σ = 0.5
μ = 1, σ = 2.0
probability density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
-4.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0
x

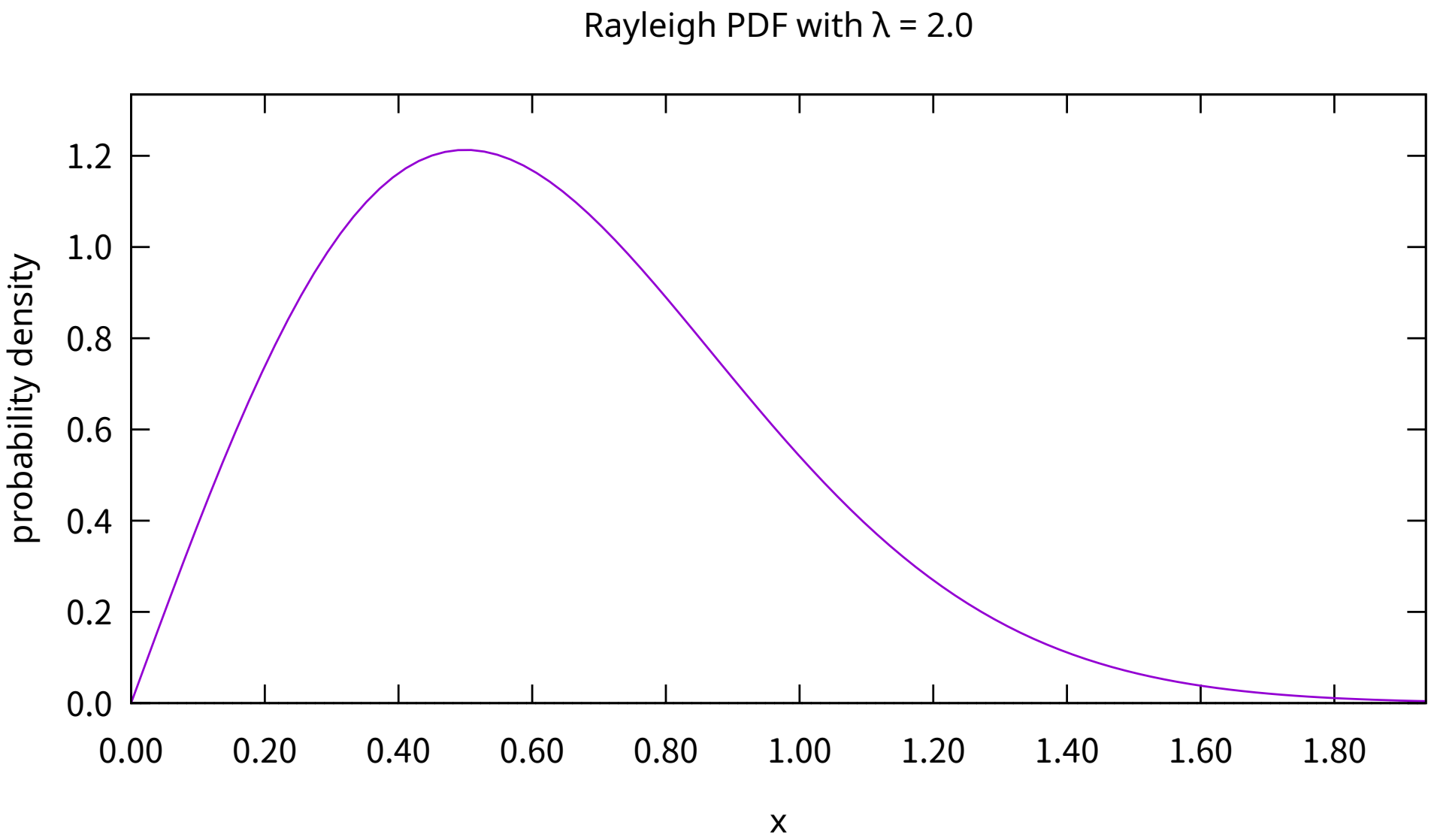

normal (also called Gauss or bell-curved) CDF
μ = 0, σ = 1.0
μ = 2, σ = 0.5
μ = 1, σ = 2.0
probability density
1.0
0.8
0.6
0.4
0.2
0.0
-4.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0
x

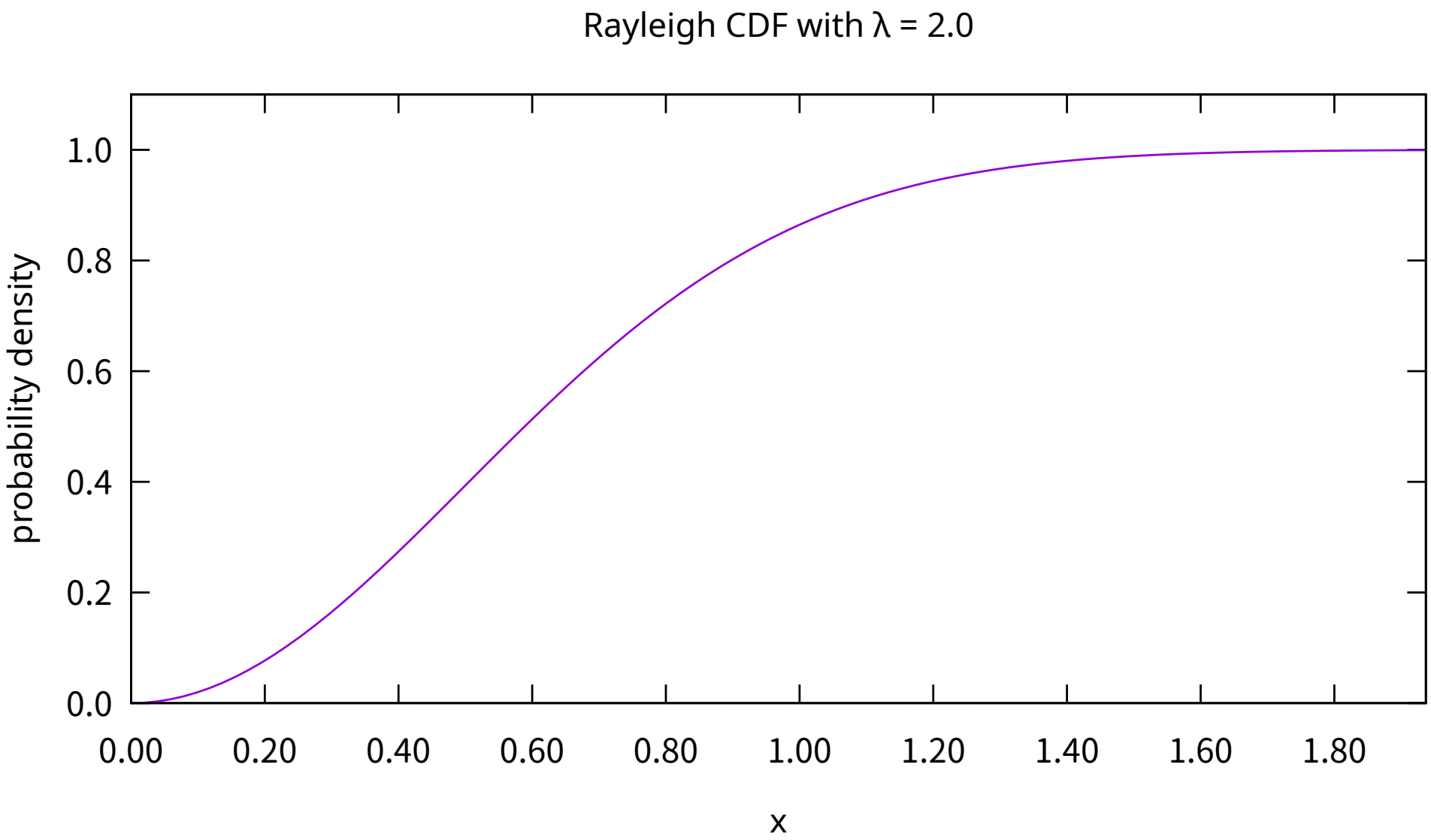

Pareto PDF with a = 1, b = 3
probability density
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
x
Discontinuity achieved by plotting twice
with affine mapped parametric ranges

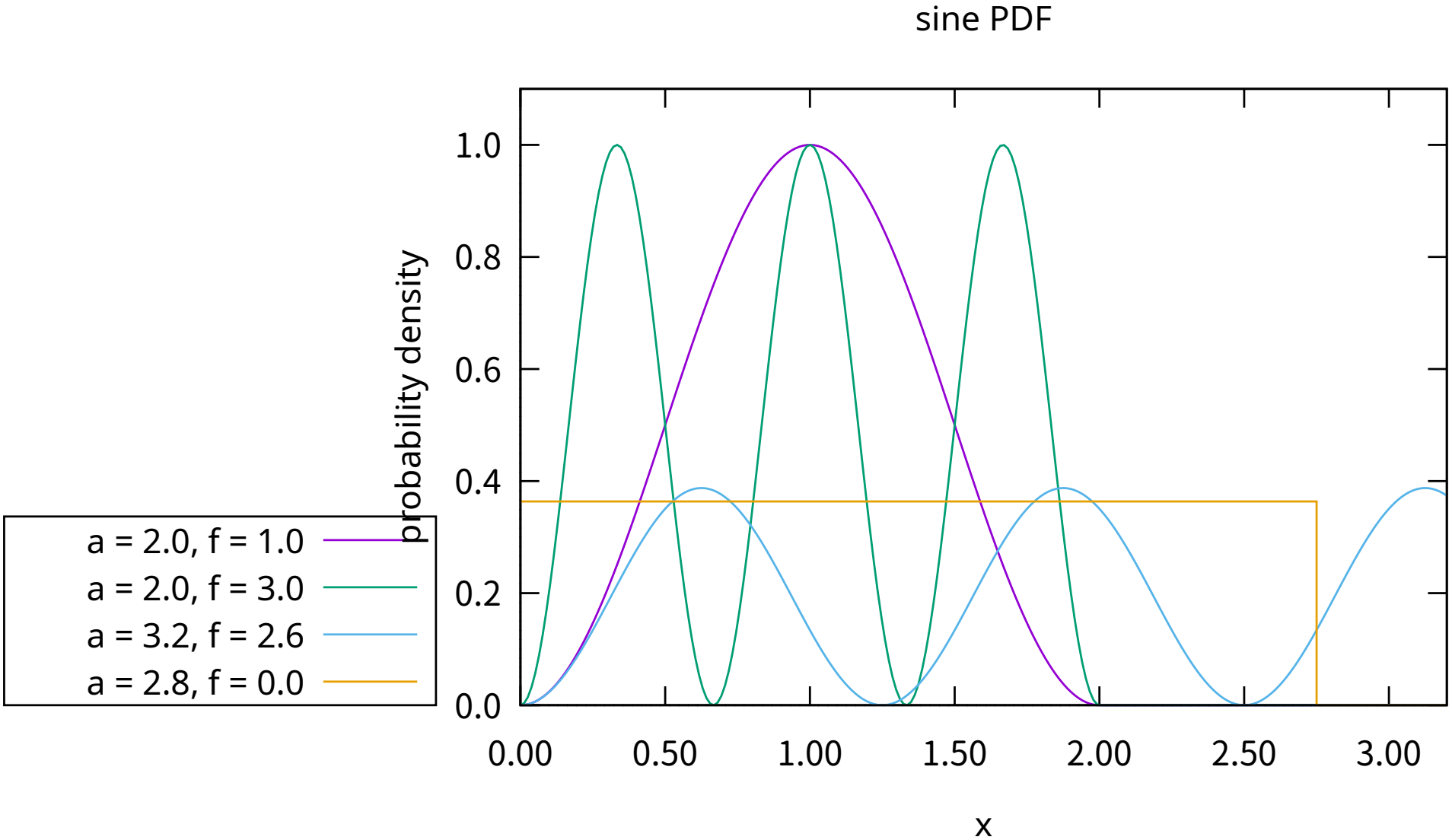

Pareto CDF with a = 1, b = 3
probability density
1.0
0.8
0.6
0.4
0.2
0.0
Cusp achieved by selecting point
as part of function samples
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
x

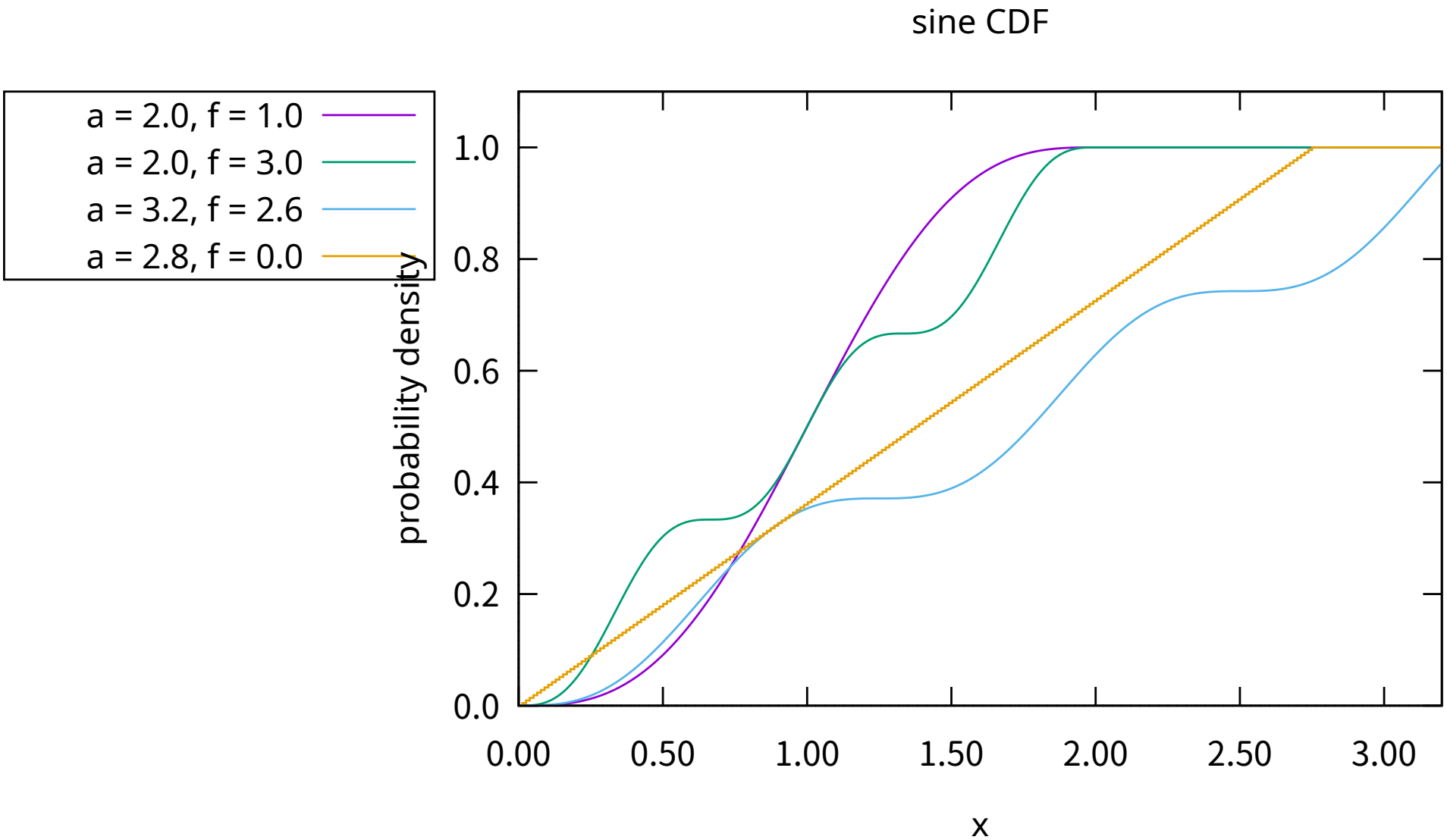

Poisson PDF with μ = 4.0
probability density
0.21
0.19
0.17
0.15
0.13
0.11
0.09
0.06
0.04
0.02
0.00
0
2
4
6
8
10
12
k

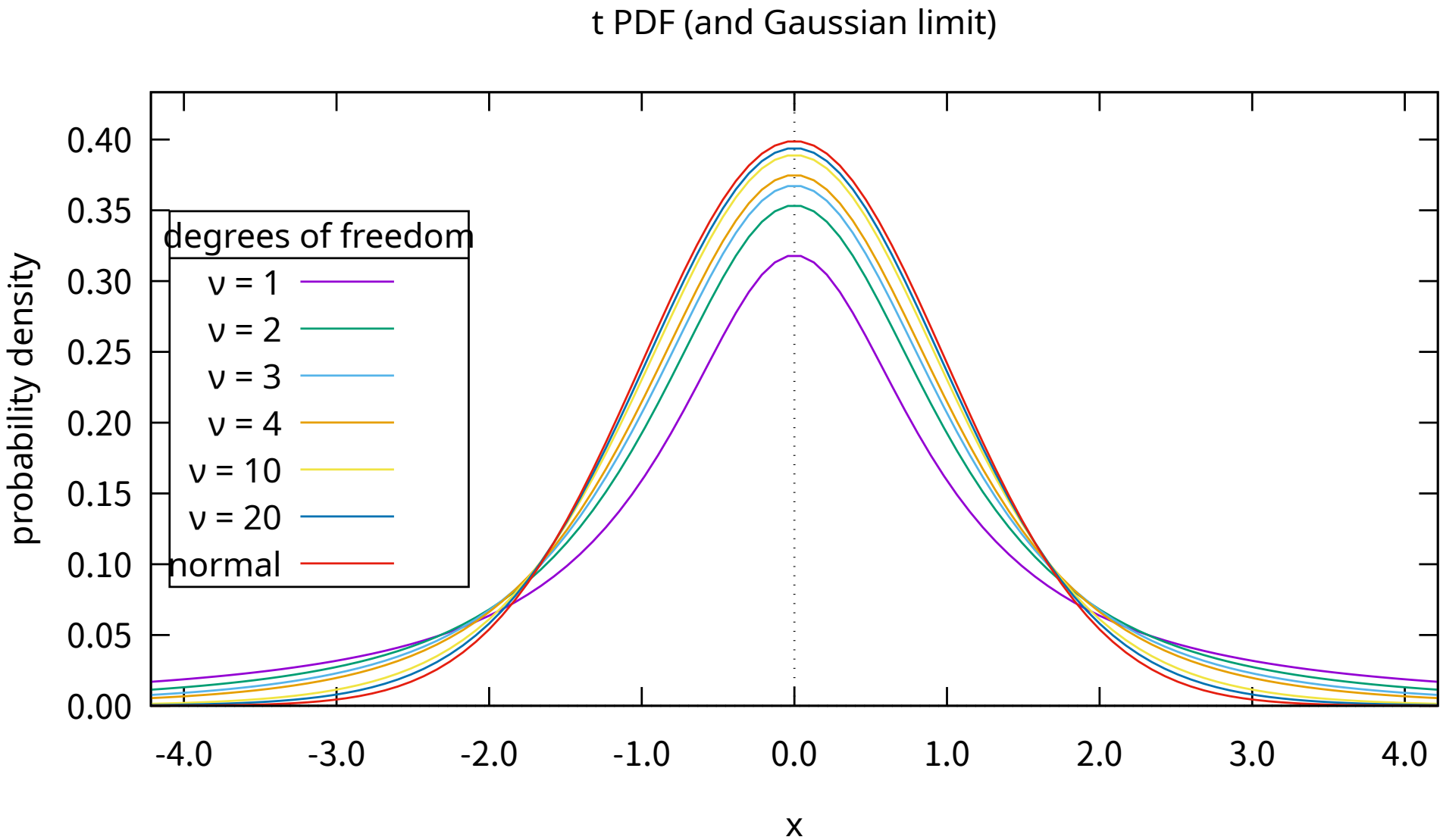

Poisson CDF with μ = 4.0
probability density
1.10
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
-0.10
0
2
4
6
8
10
12
k

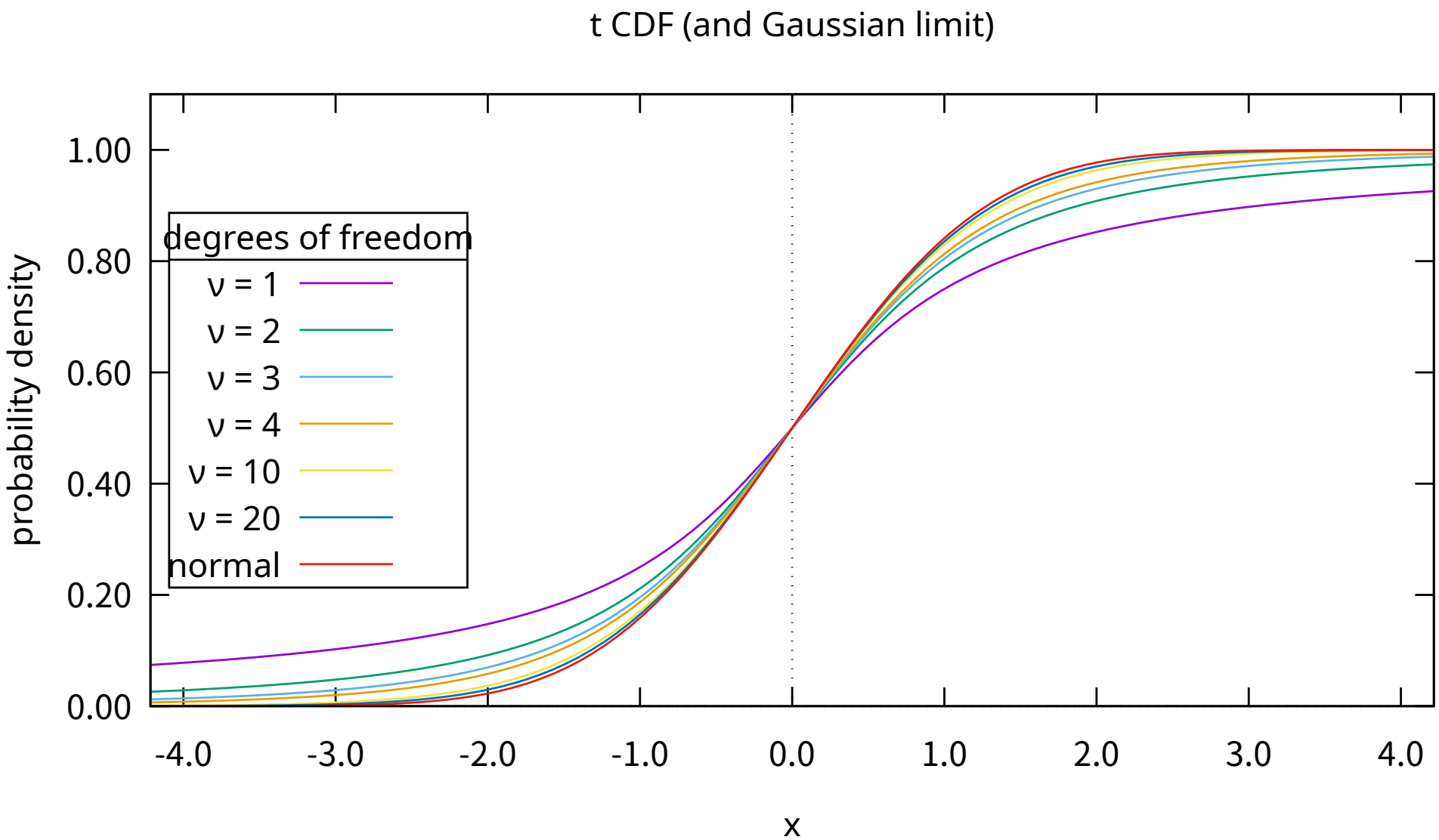

Rayleigh PDF with λ = 2.0
probability density
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.20
0.40
0.60
0.80
1.00
1.20
1.40
1.60
1.80
x

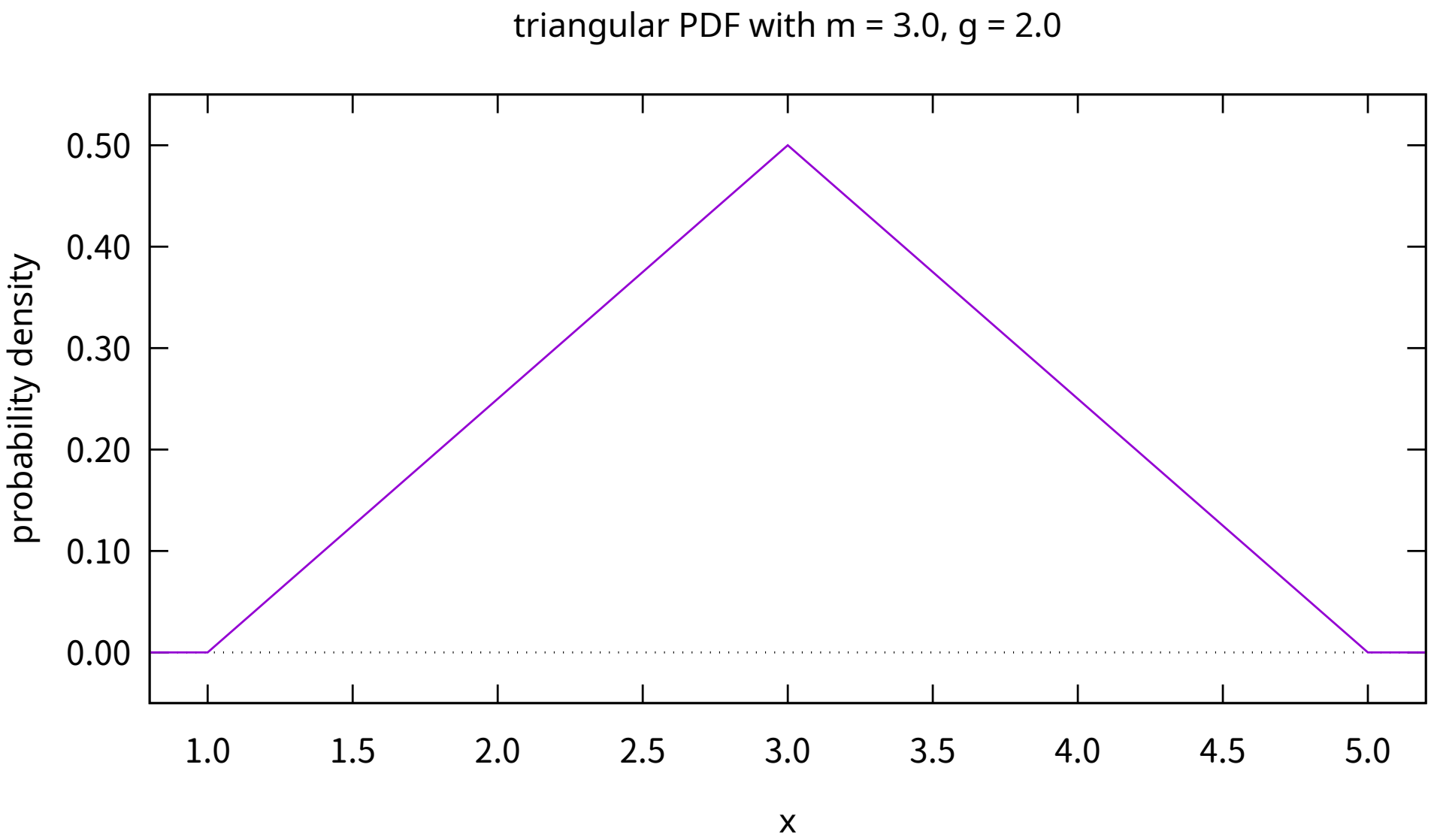

Rayleigh CDF with λ = 2.0
probability density
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.20
0.40
0.60
0.80
1.00
1.20
1.40
1.60
1.80
x

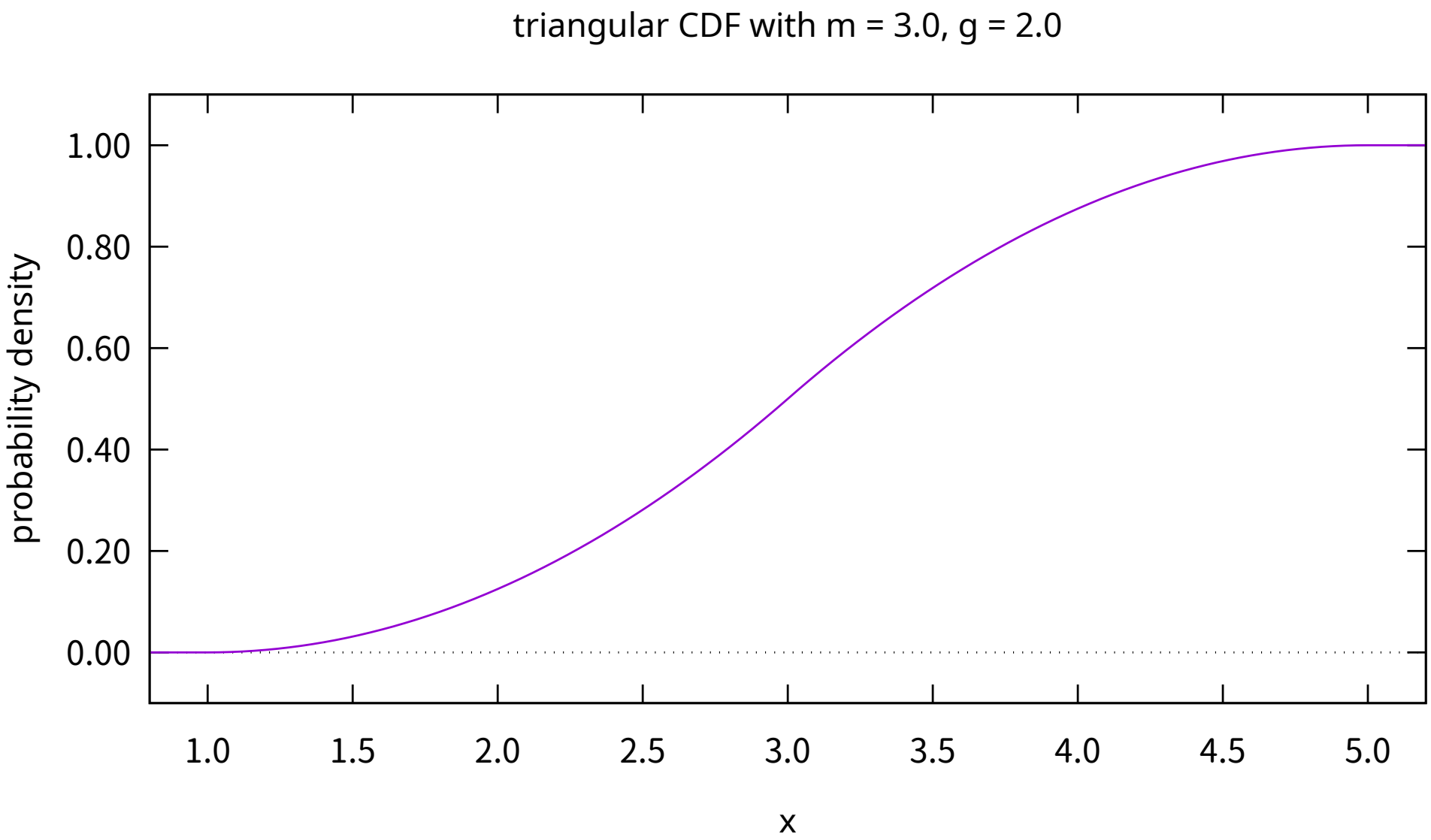

sine PDF
probability density
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.50
1.00
1.50
2.00
2.50
3.00
x
a = 2.0, f = 1.0
a = 2.0, f = 3.0
a = 3.2, f = 2.6
a = 2.8, f = 0.0

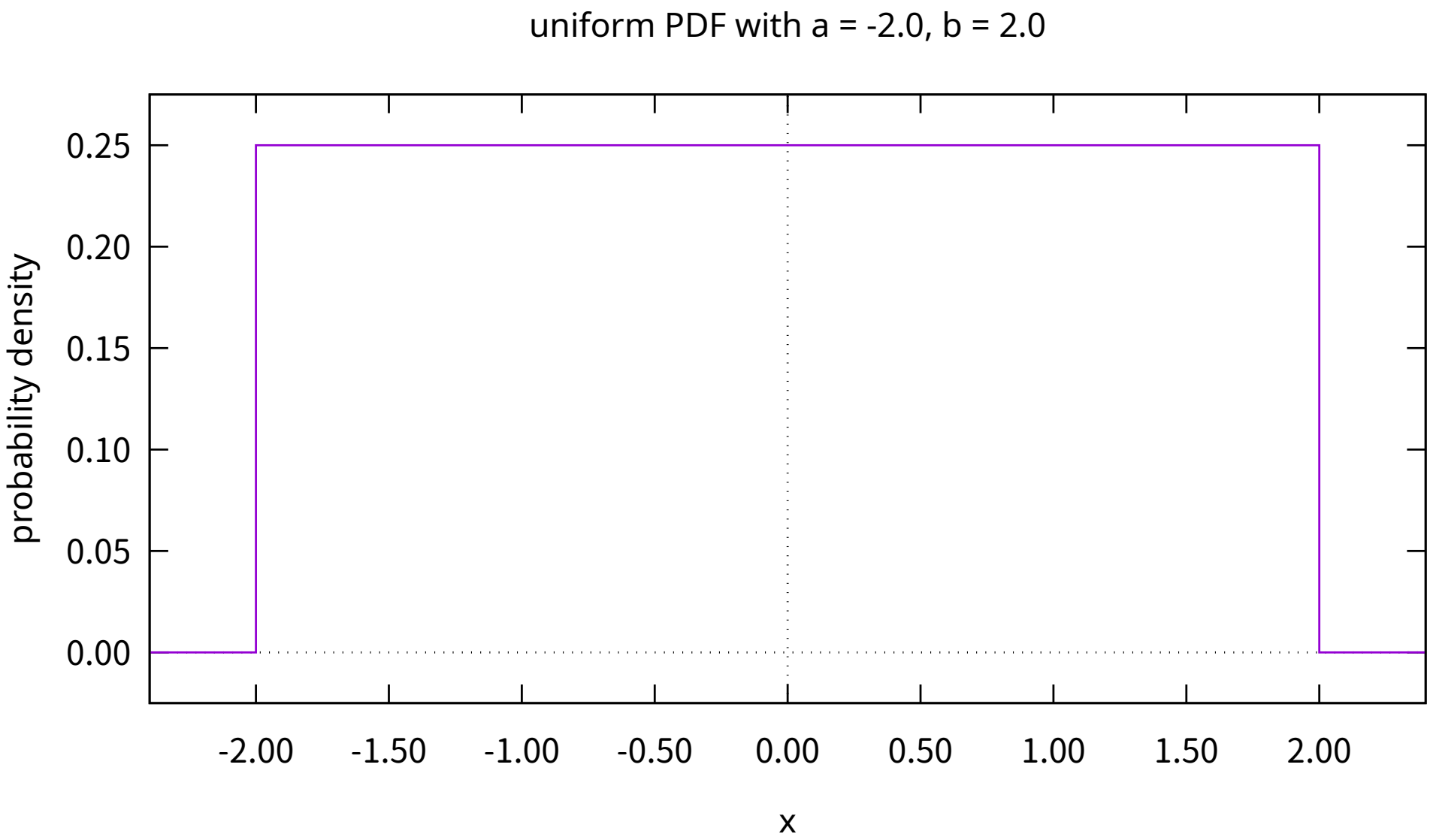

sine CDF
a = 2.0, f = 1.0
a = 2.0, f = 3.0
a = 3.2, f = 2.6
a = 2.8, f = 0.0
probability density
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.50
1.00
1.50
2.00
2.50
3.00
x

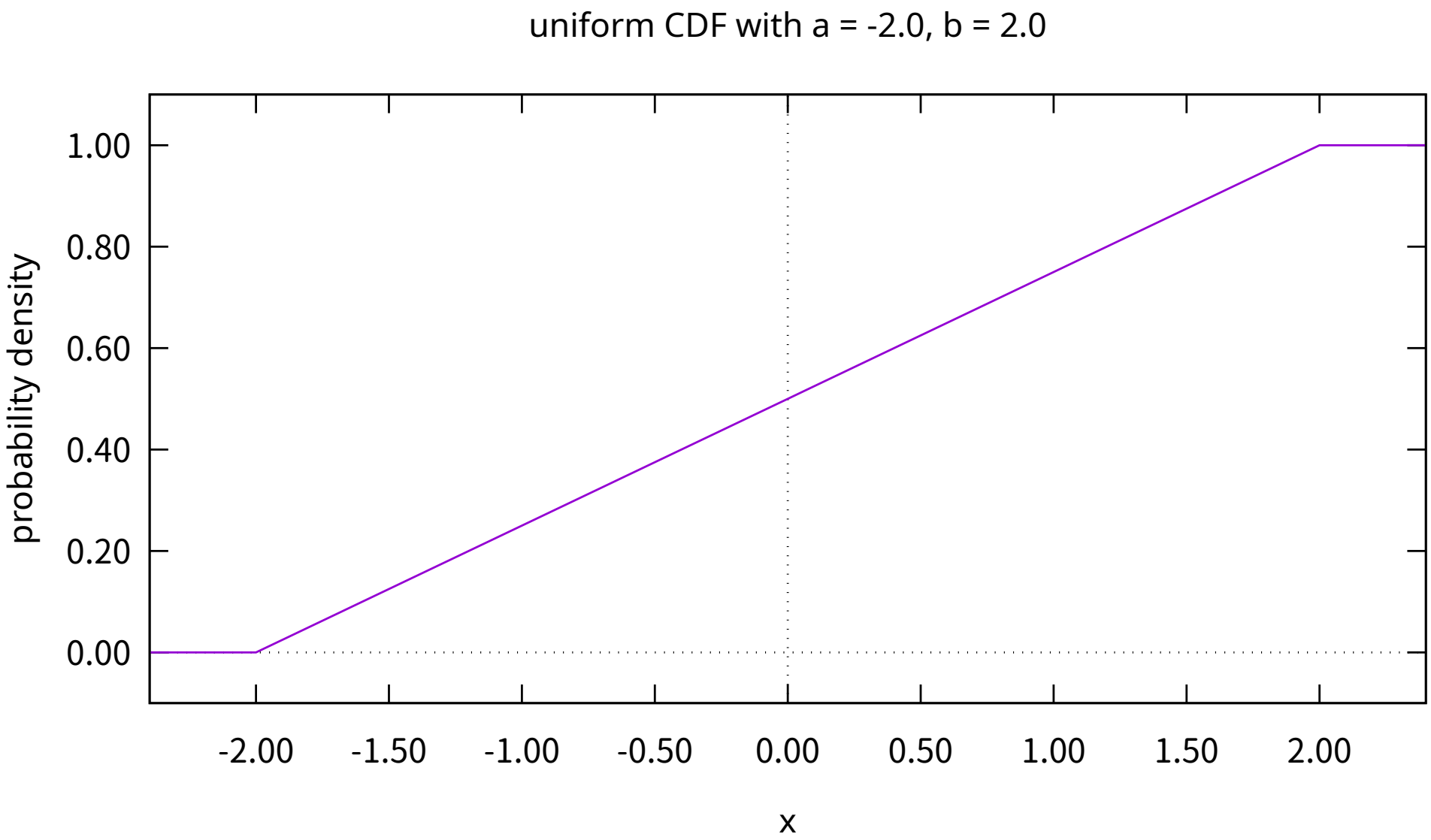

t PDF (and Gaussian limit)
degrees of freedom
ν = 1
ν = 2
ν = 3
ν = 4
ν = 10
ν = 20
normal
probability density
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-4.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0
x

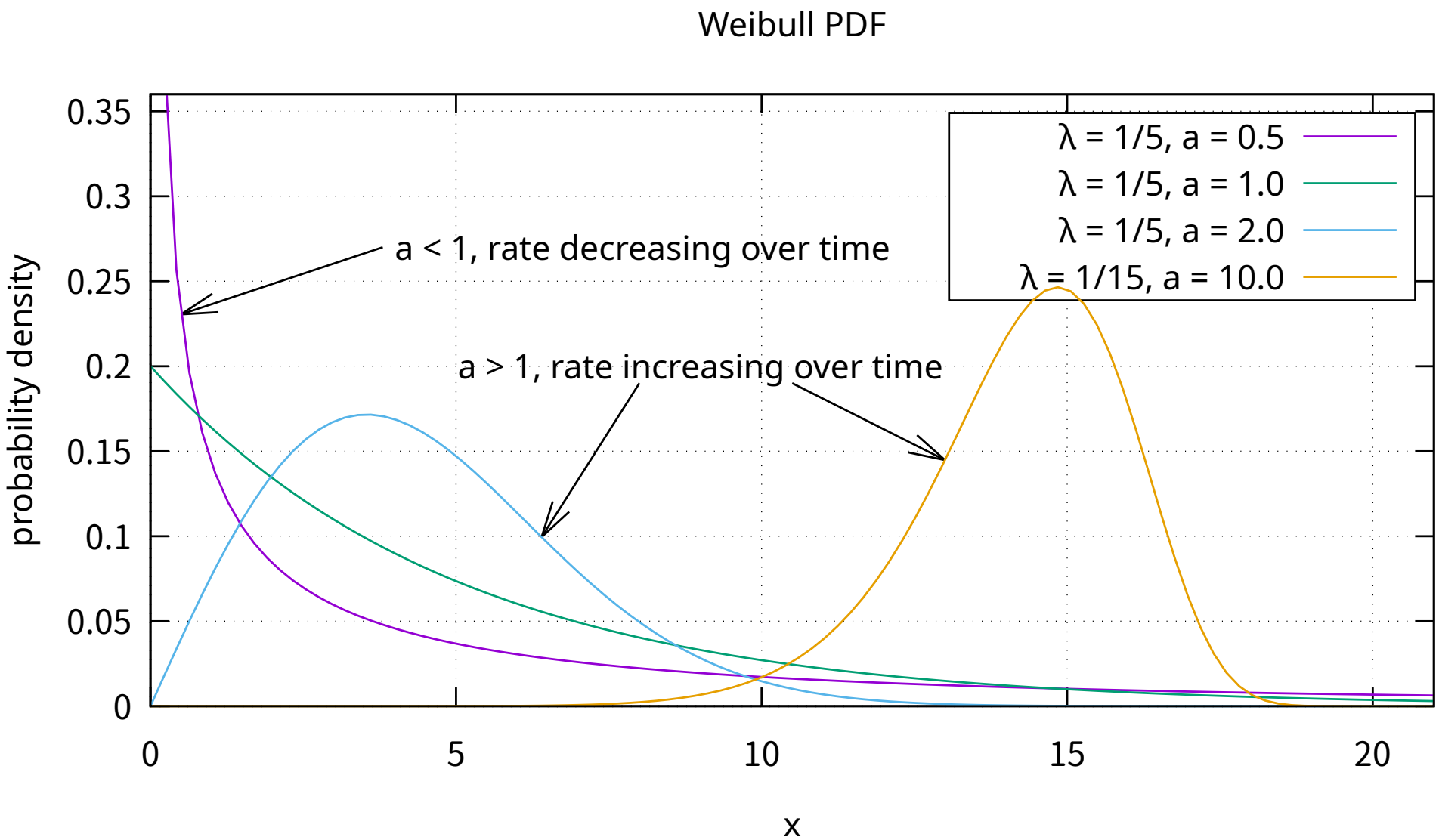

t CDF (and Gaussian limit)
degrees of freedom
ν = 1
ν = 2
ν = 3
ν = 4
ν = 10
ν = 20
normal
probability density
1.00
0.80
0.60
0.40
0.20
0.00
-4.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0
x

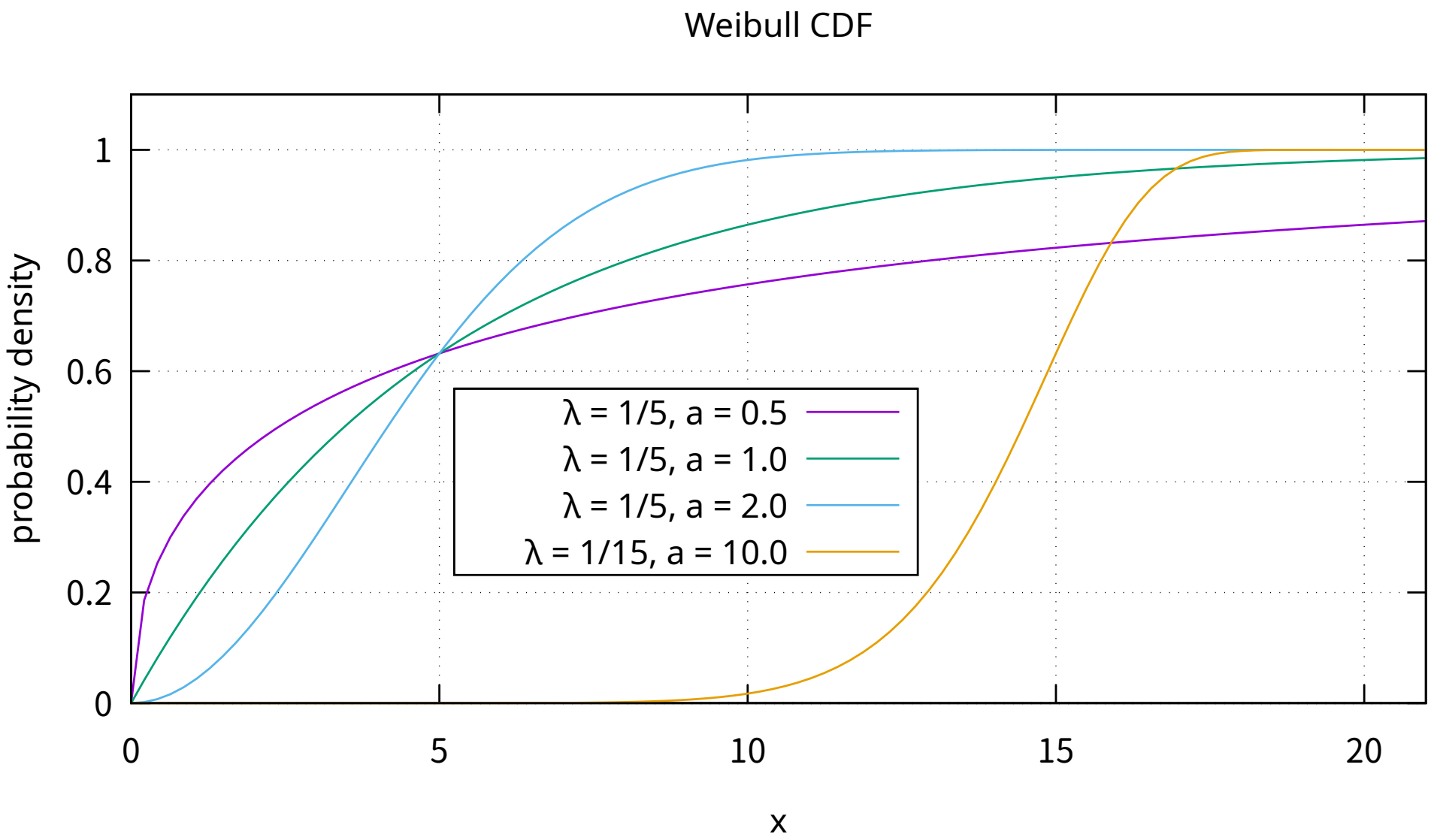

triangular PDF with m = 3.0, g = 2.0
probability density
0.50
0.40
0.30
0.20
0.10
0.00
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
x

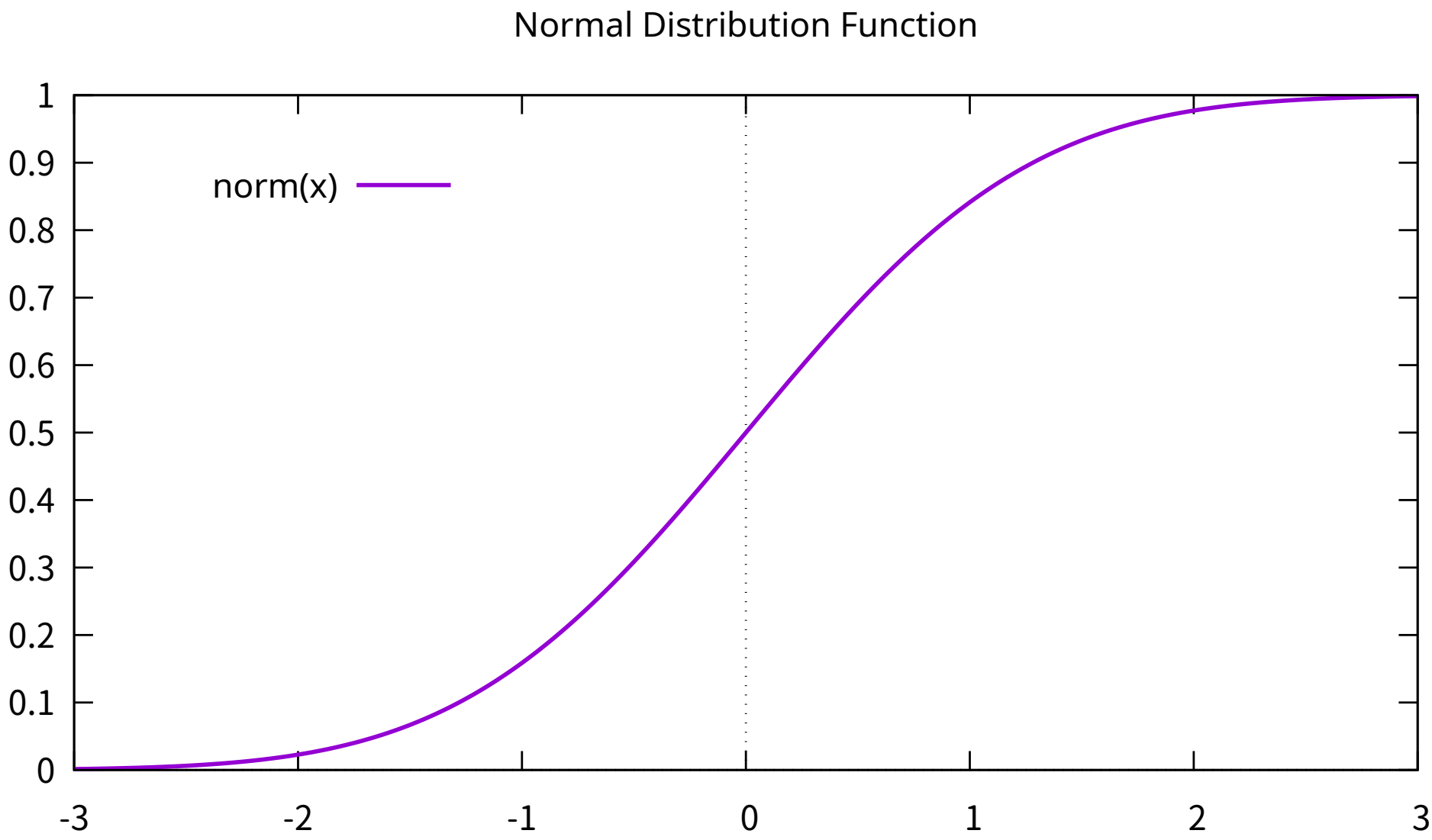

triangular CDF with m = 3.0, g = 2.0
probability density
1.00
0.80
0.60
0.40
0.20
0.00
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
x

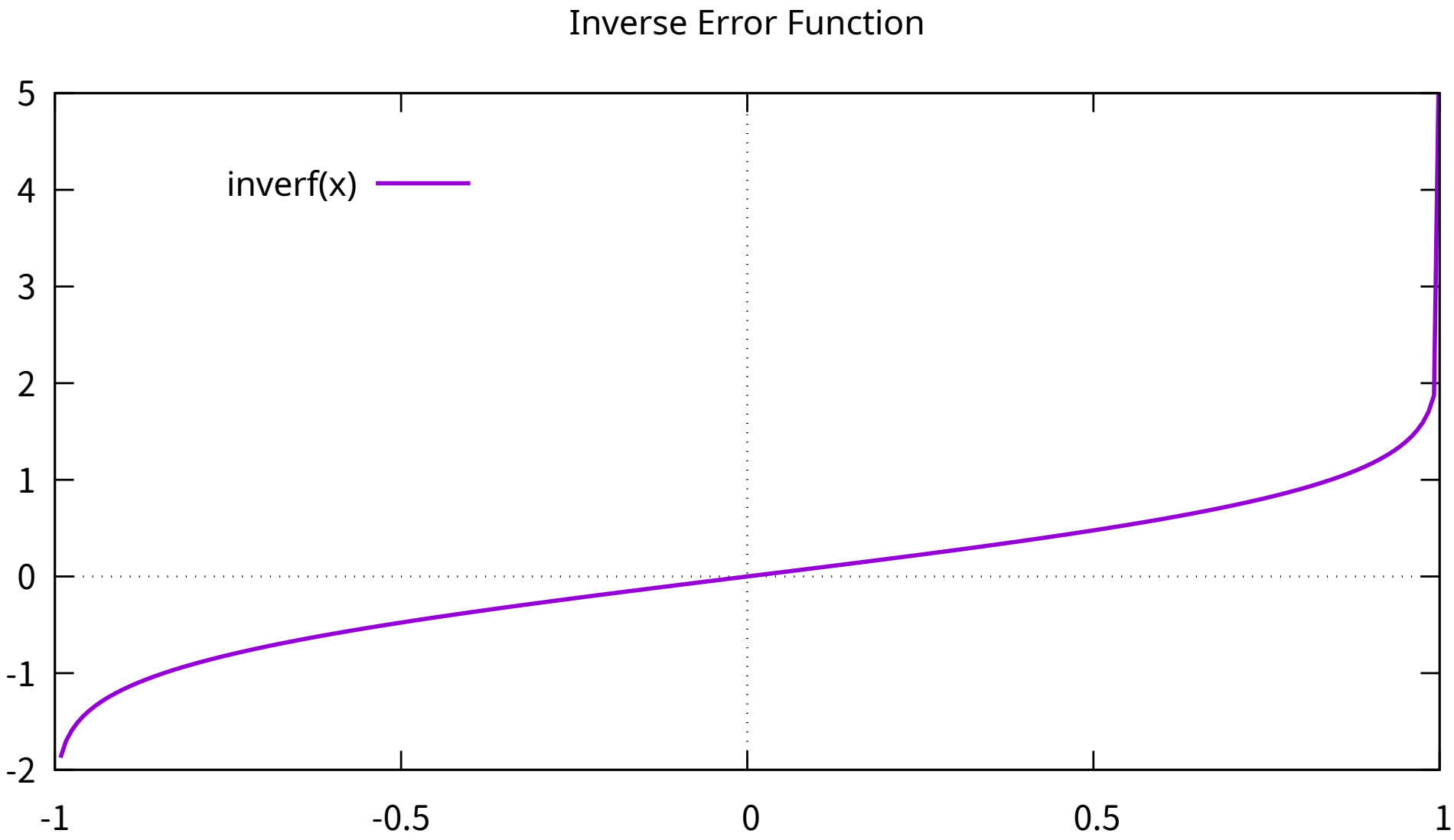

uniform PDF with a = -2.0, b = 2.0
probability density
0.25
0.20
0.15
0.10
0.05
0.00
-2.00
-1.50
-1.00
-0.50
0.00
0.50
1.00
1.50
2.00
x

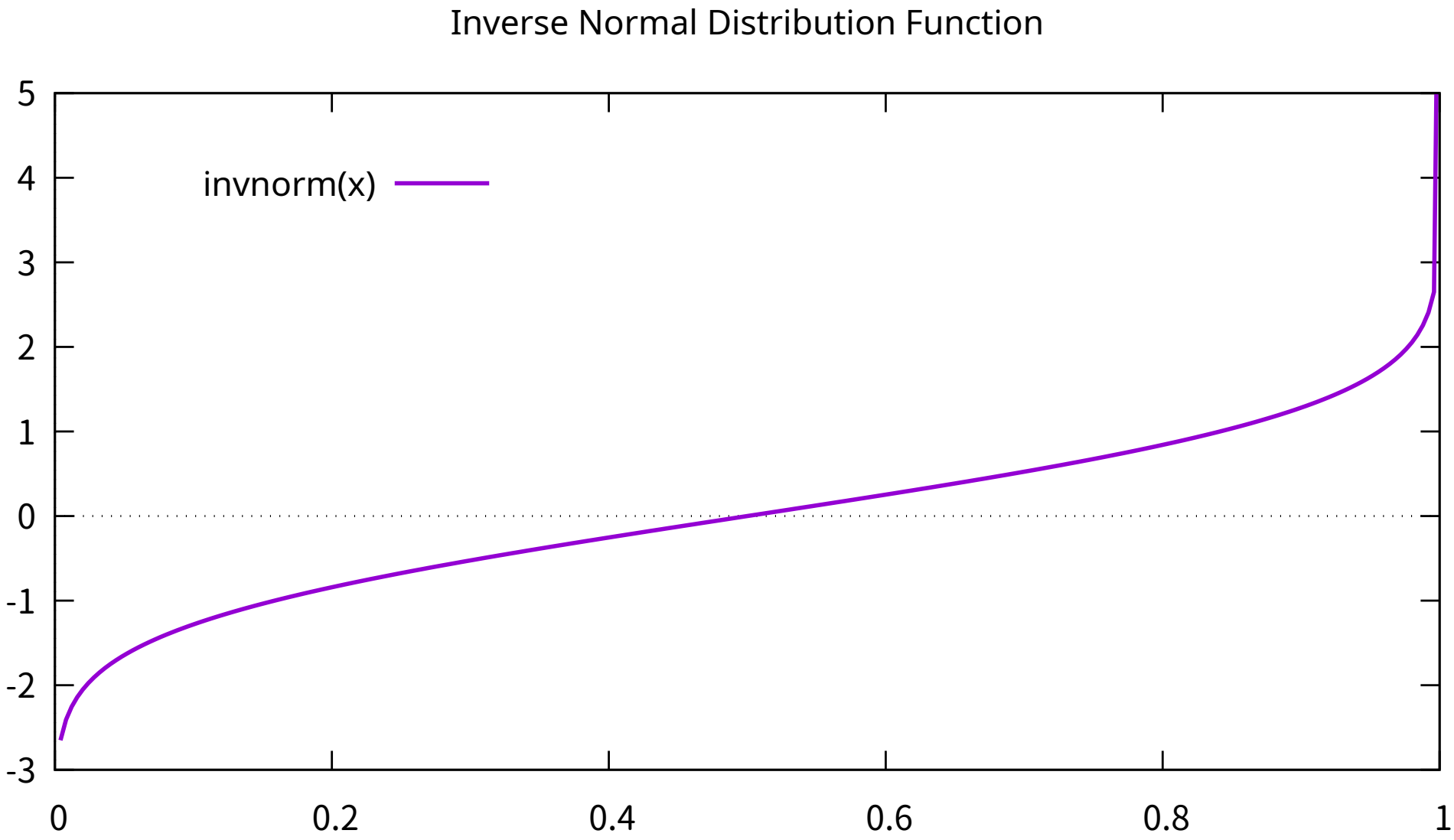

uniform CDF with a = -2.0, b = 2.0
probability density
1.00
0.80
0.60
0.40
0.20
0.00
-2.00
-1.50
-1.00
-0.50
0.00
0.50
1.00
1.50
2.00
x

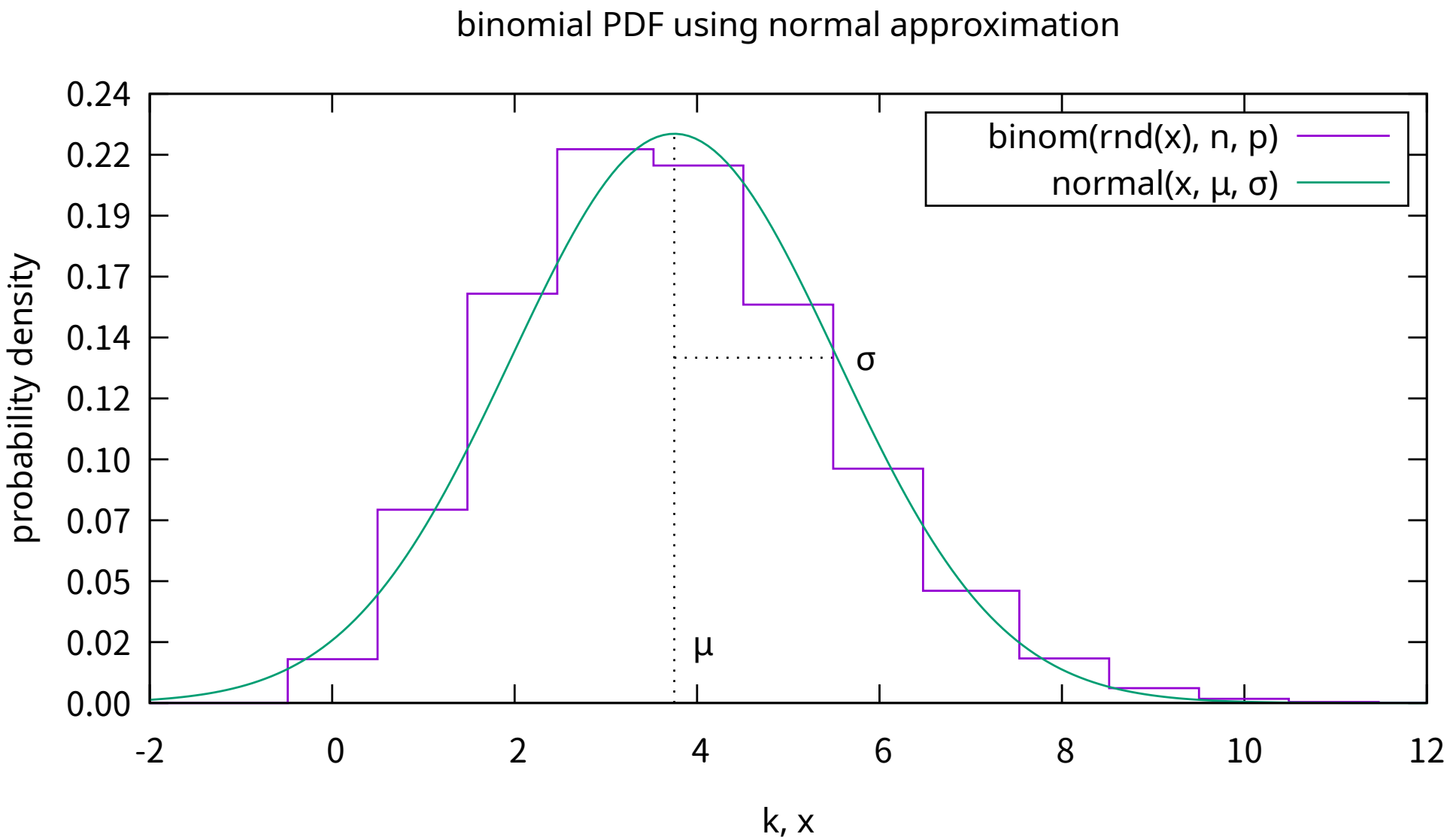

Weibull PDF
probability density
x
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
λ = 1/5, a = 0.5
λ = 1/5, a = 1.0
λ = 1/5, a = 2.0
λ = 1/15, a = 10.0
a < 1, rate decreasing over time
a > 1, rate increasing over time

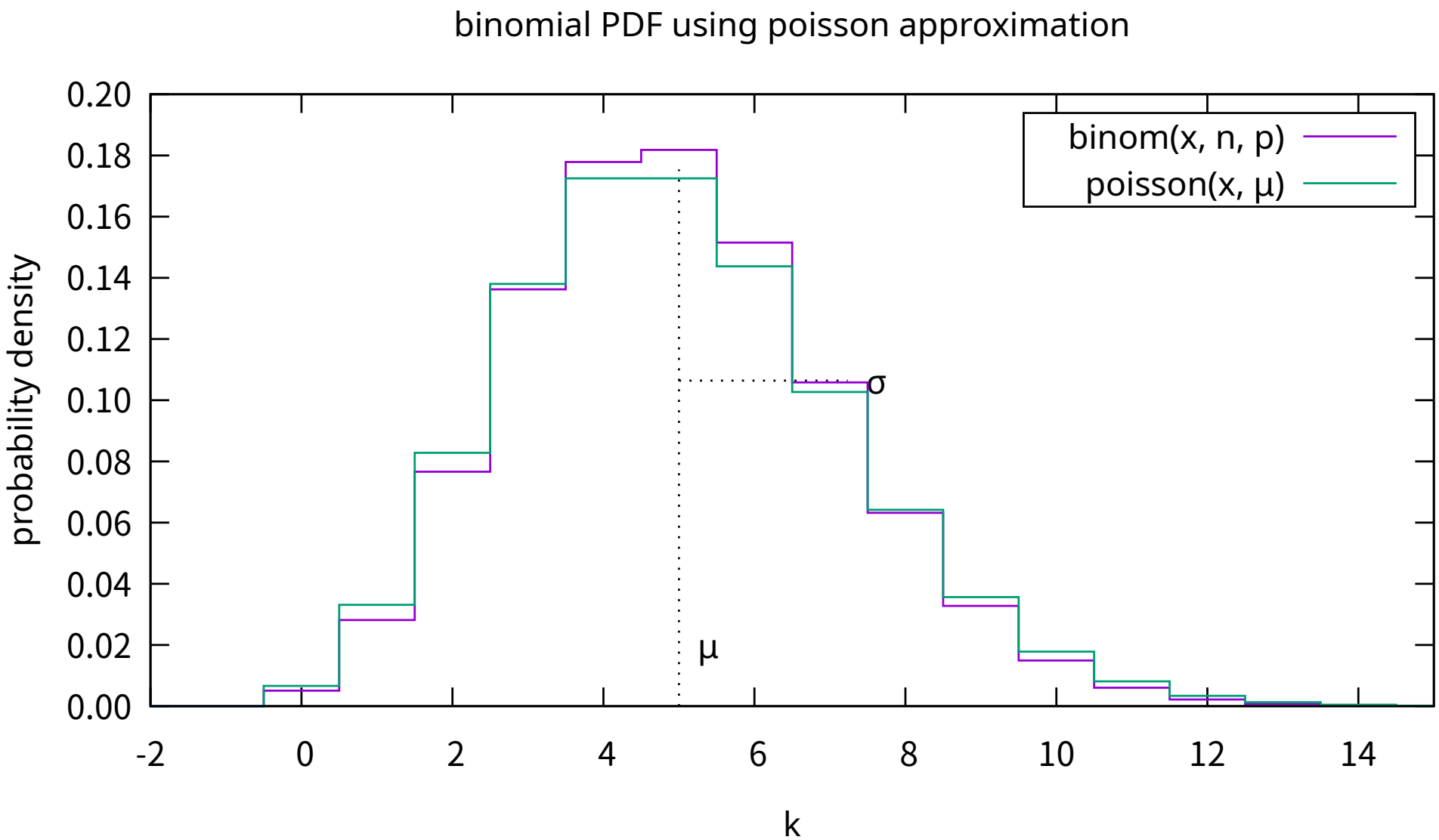

Weibull CDF
probability density
x
0
0.2
0.4
0.6
0.8
1
0
5
10
15
20
λ = 1/5, a = 0.5
λ = 1/5, a = 1.0
λ = 1/5, a = 2.0
λ = 1/15, a = 10.0

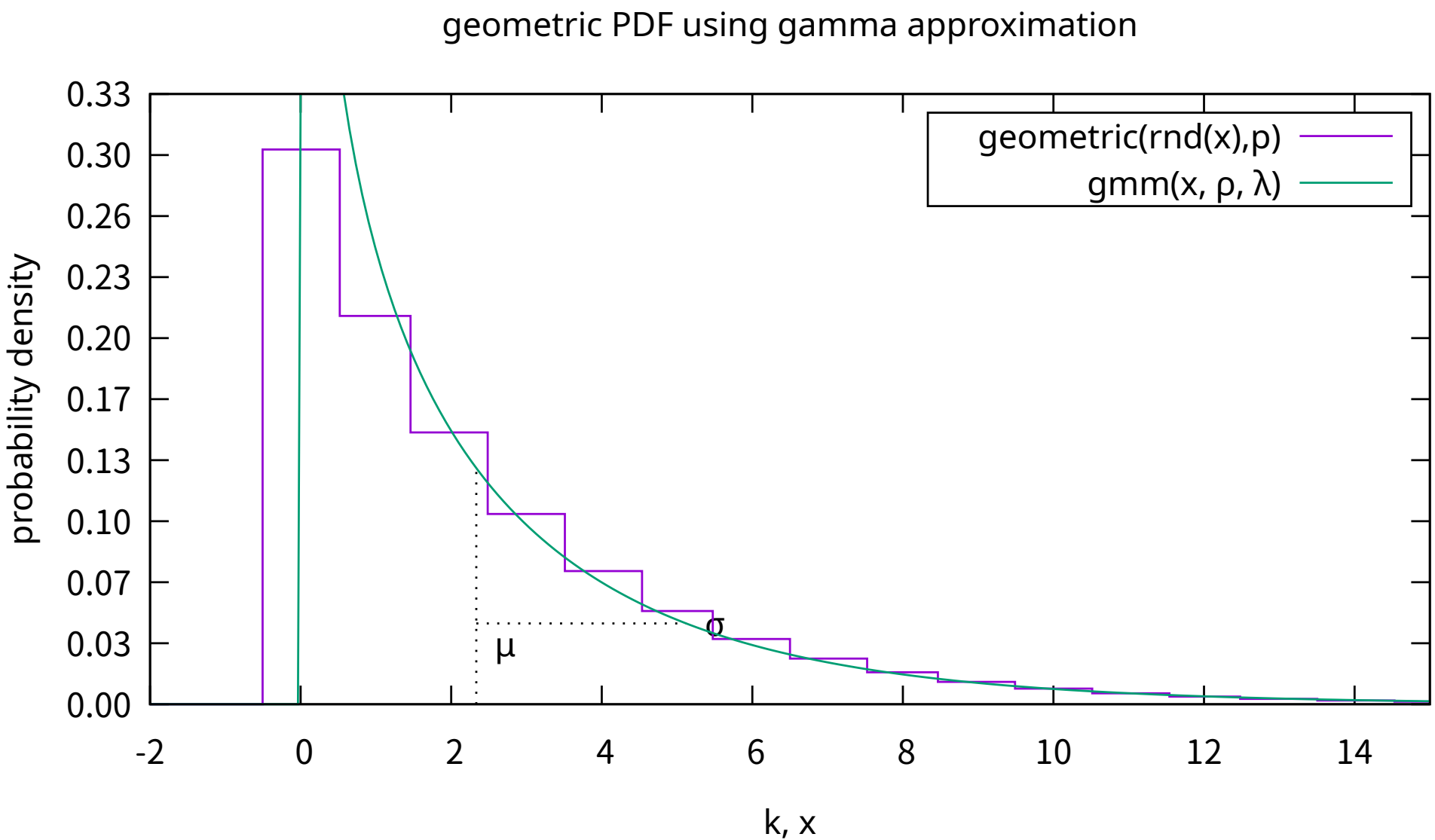

Normal Distribution Function
norm(x)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-3
-2
-1
0
1
2
3

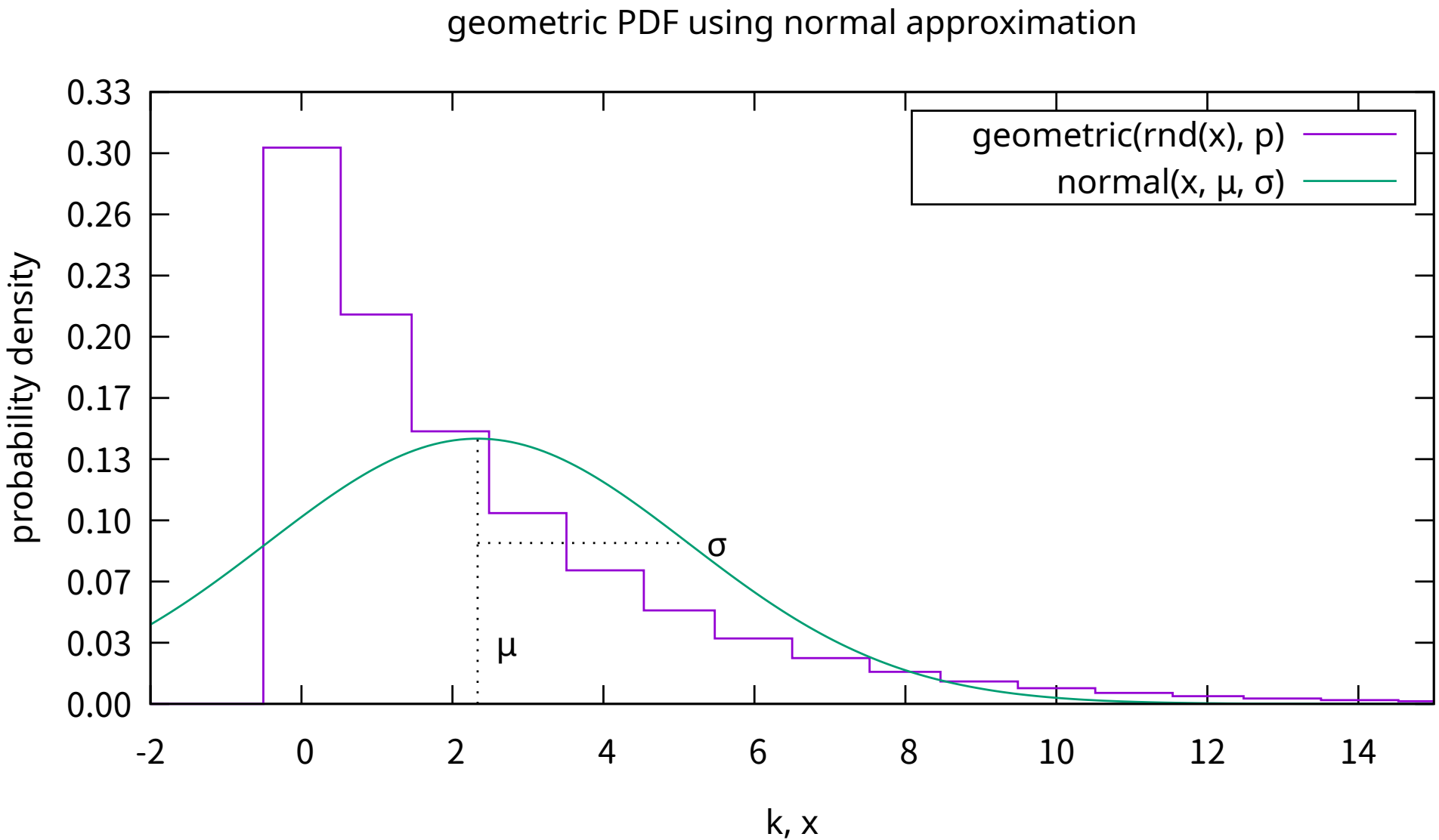

Inverse Error Function
inverf(x)
5
4
3
2
1
0
-1
-2
-1
-0.5
0
0.5
1

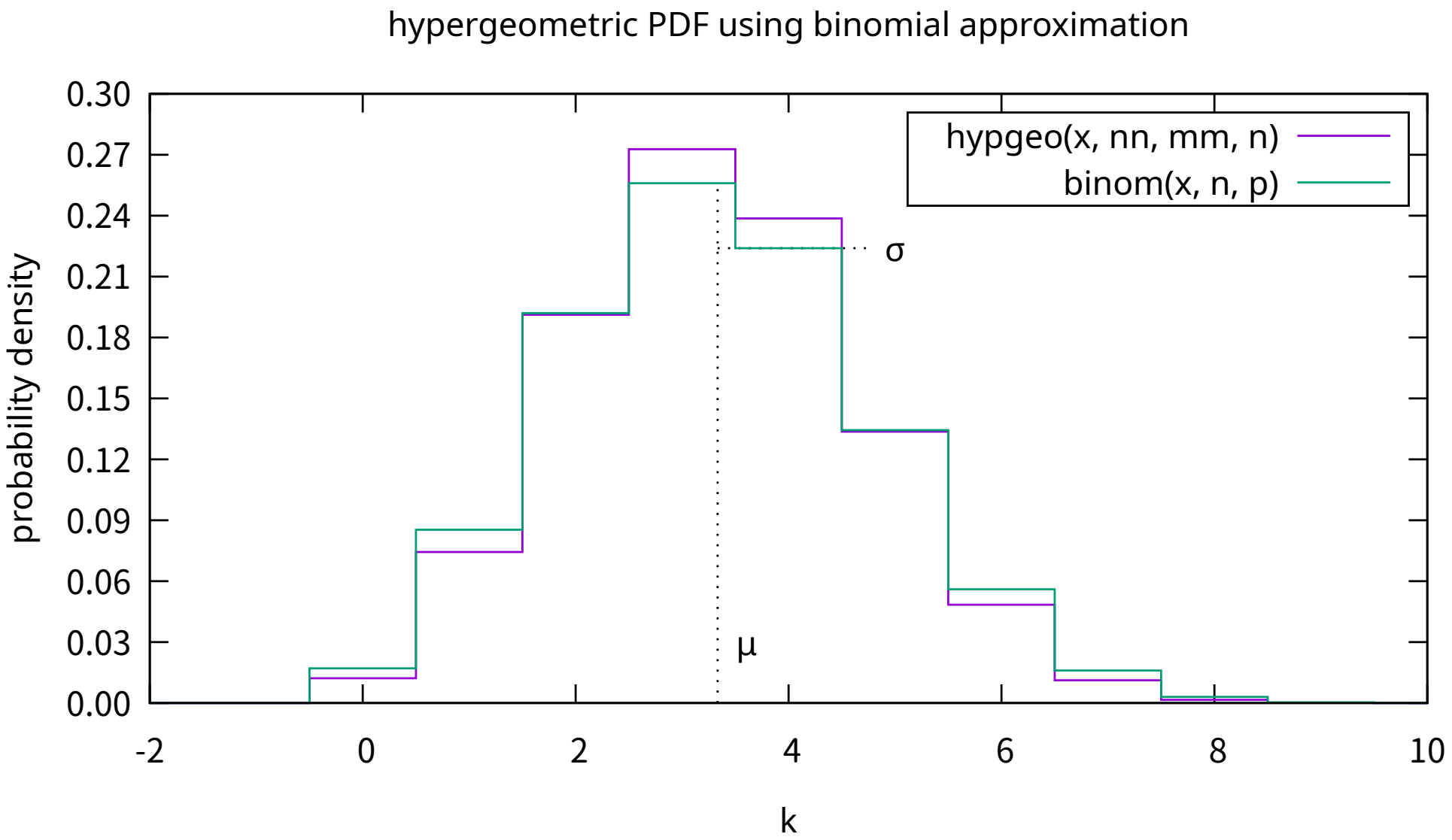

Inverse Normal Distribution Function
invnorm(x)
5
4
3
2
1
0
-1
-2
-3
0
0.2
0.4
0.6
0.8
1

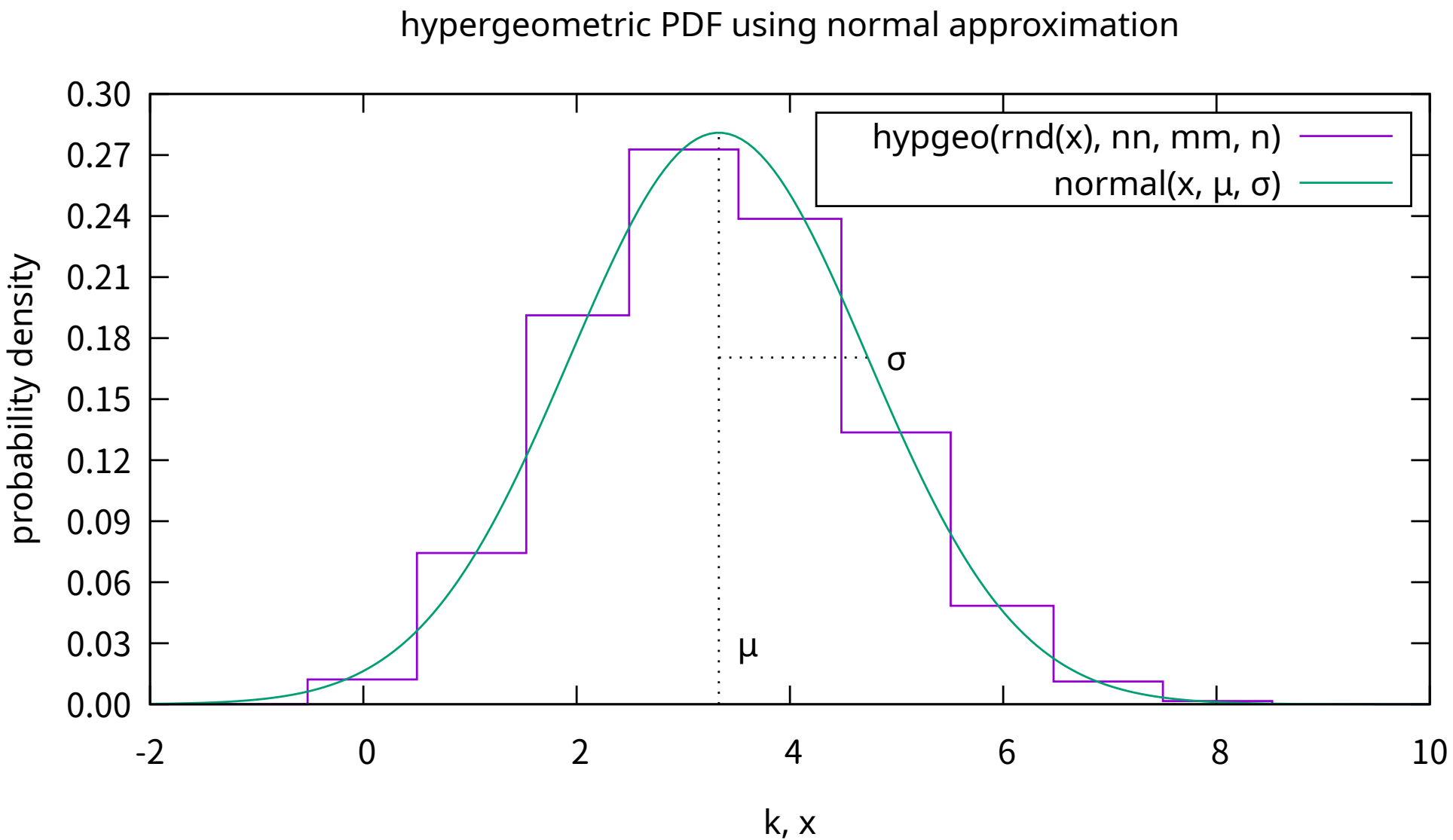

binomial PDF using normal approximation
binom(rnd(x), n, p)
normal(x, μ, σ)
σ
μ
probability density
k, x
0.24
0.22
0.19
0.17
0.14
0.12
0.10
0.07
0.05
0.02
0.00
-2
0
2
4
6
8
10
12

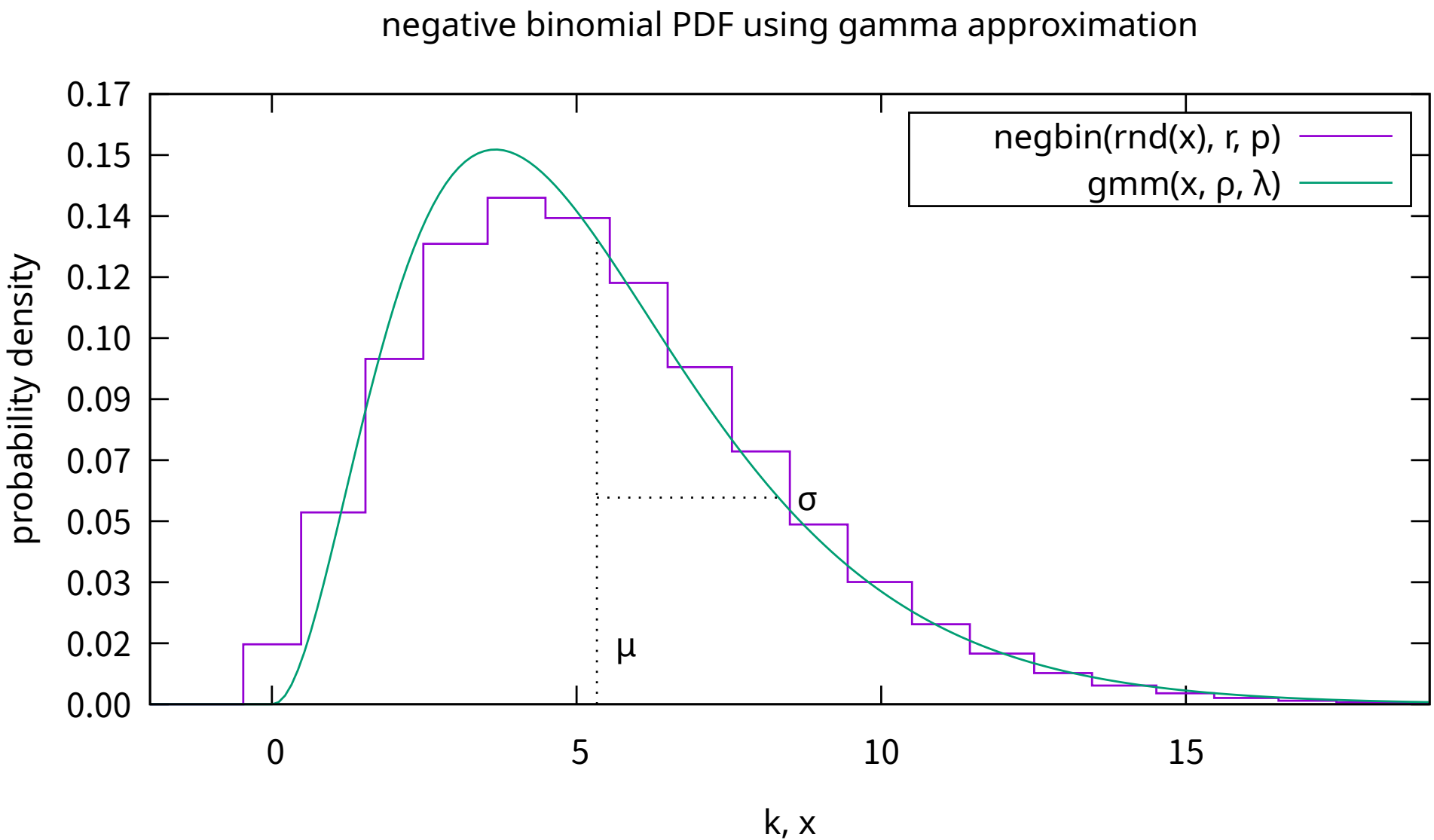

binomial PDF using poisson approximation
binom(x, n, p)
poisson(x, μ)
σ
μ
probability density
k
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
-2
0
2
4
6
8
10
12
14

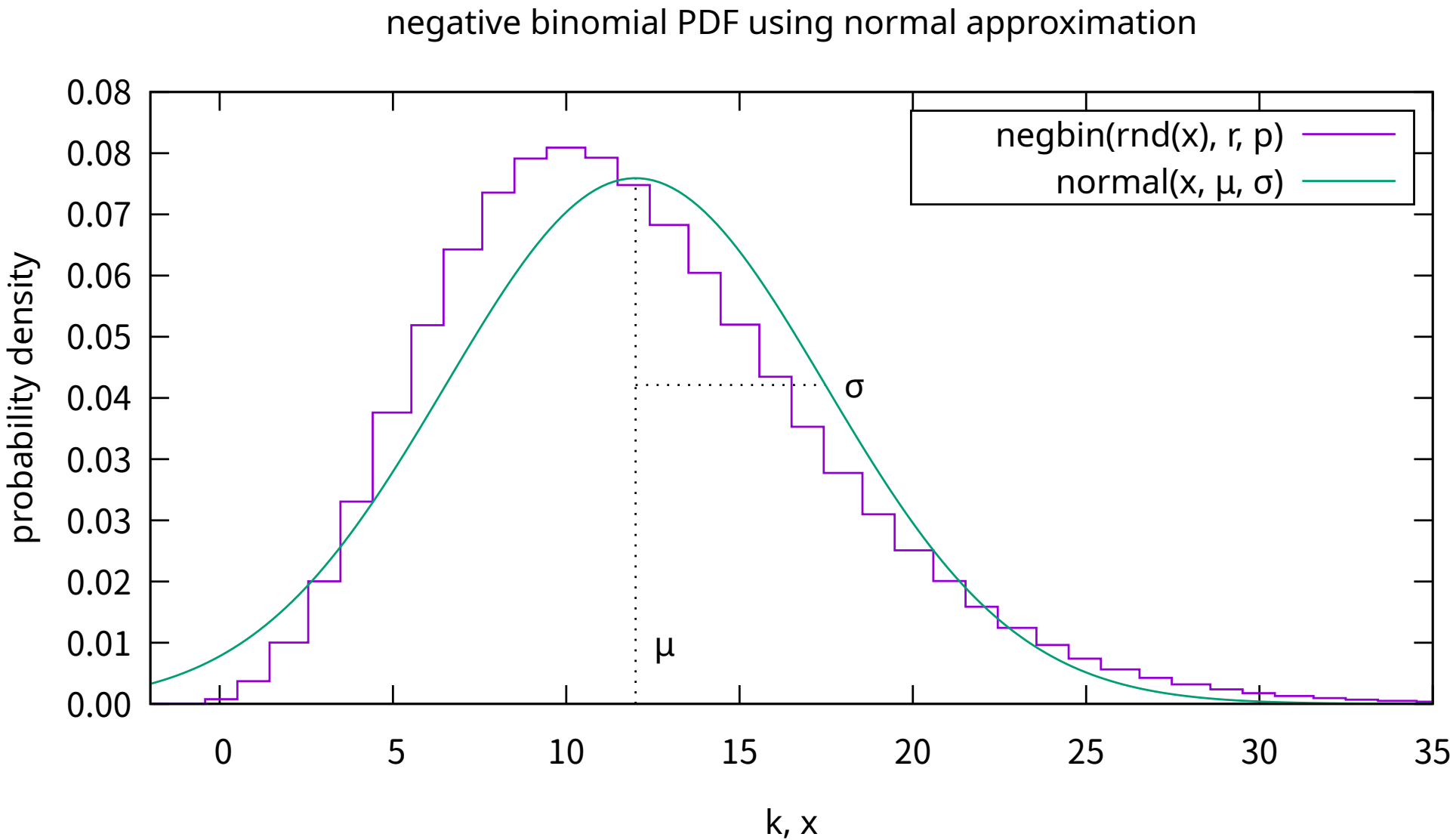

geometric PDF using gamma approximation
geometric(rnd(x),p)
gmm(x, ρ, λ)
probability density
k, x
μ
σ
0.33
0.30
0.26
0.23
0.20
0.17
0.13
0.10
0.07
0.03
0.00
-2
0
2
4
6
8
10
12
14

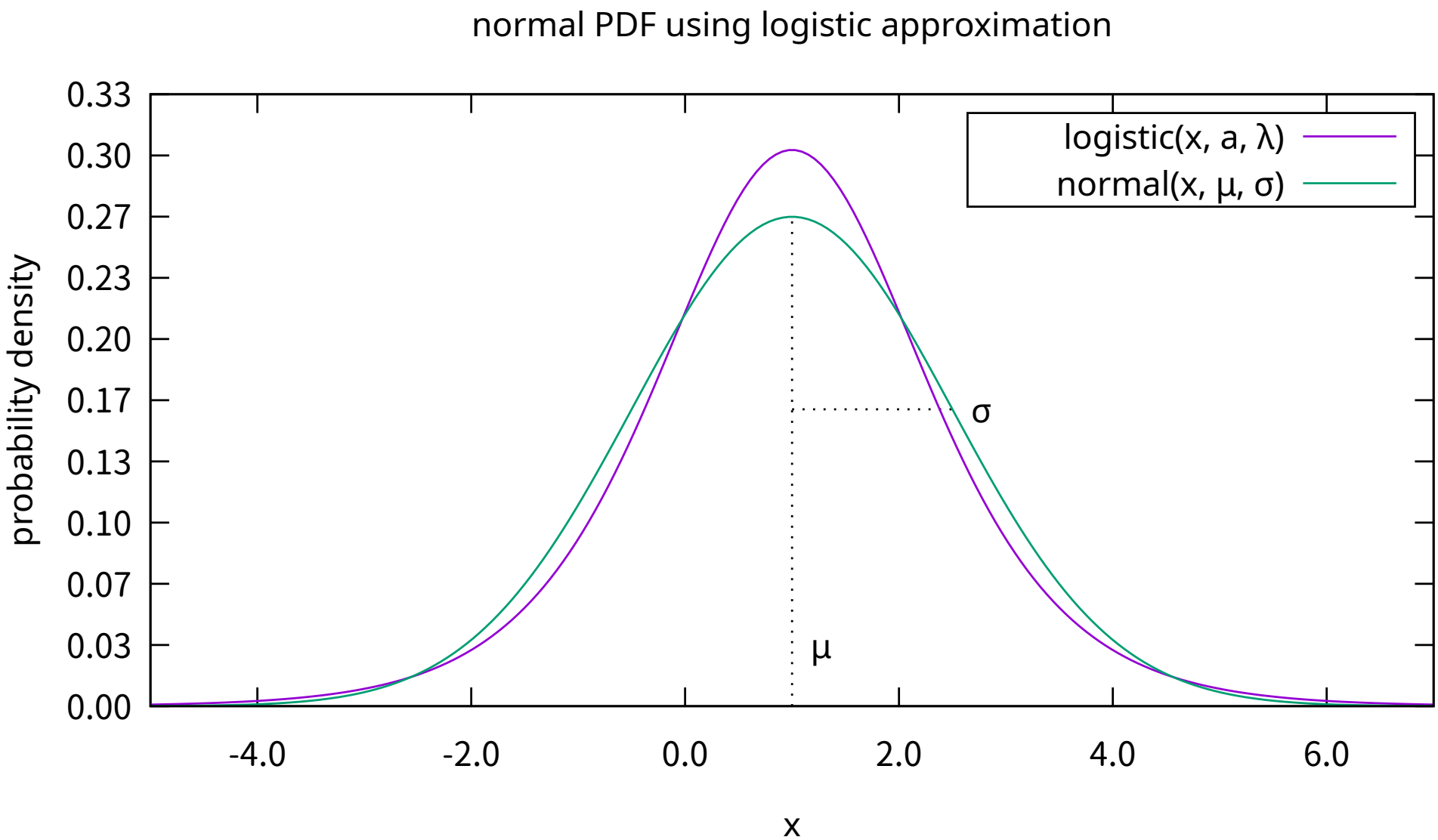

geometric PDF using normal approximation
geometric(rnd(x), p)
normal(x, μ, σ)
probability density
0.33
0.30
0.26
0.23
0.20
0.17
0.13
0.10
0.07
0.03
0.00
-2
0
2
4
6
8
10
12
14
k, x
σ
μ

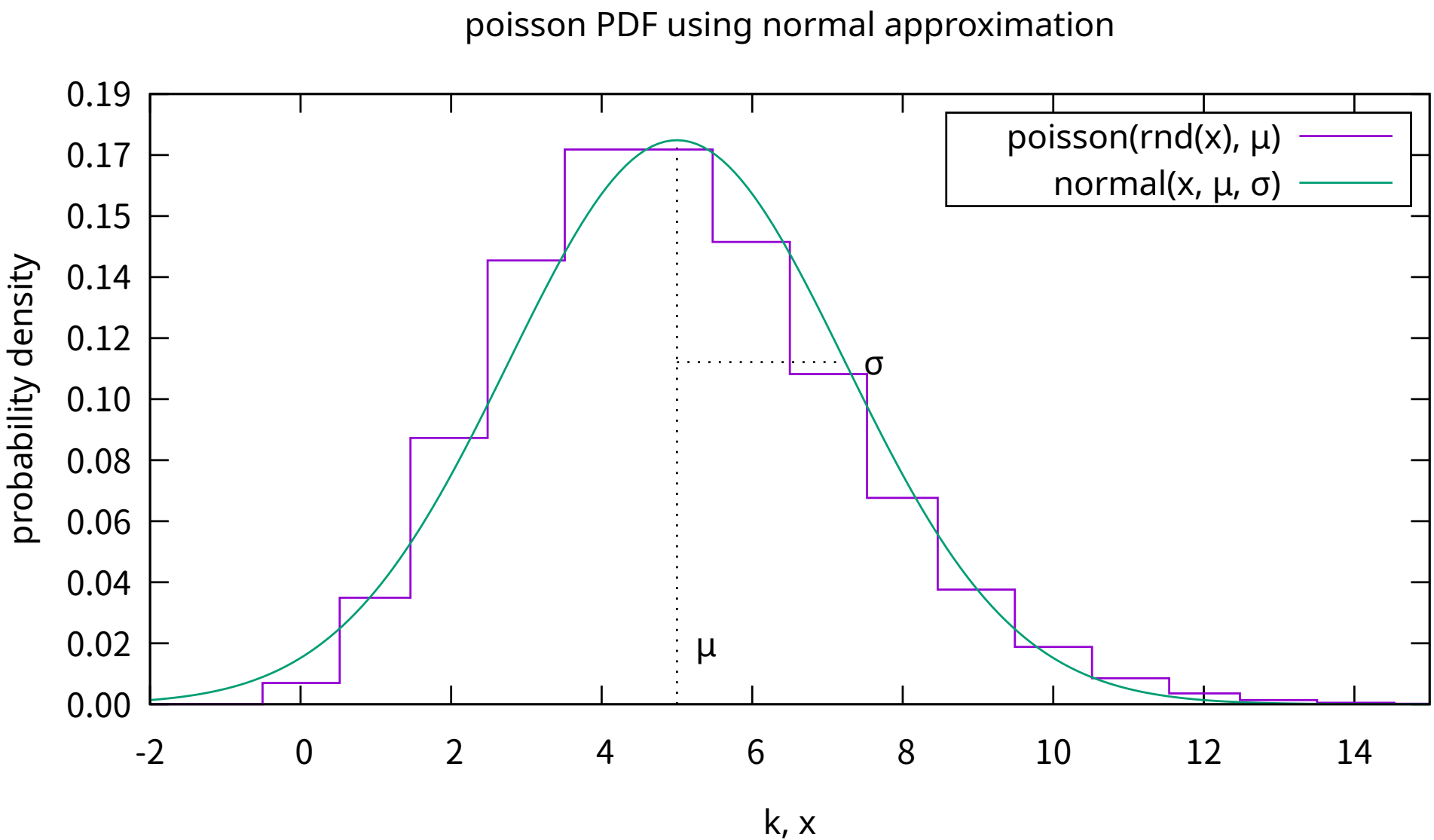

hypergeometric PDF using binomial approximation
hypgeo(x, nn, mm, n)
binom(x, n, p)
σ
μ
probability density
k
0.30
0.27
0.24
0.21
0.18
0.15
0.12
0.09
0.06
0.03
0.00
-2
0
2
4
6
8
10

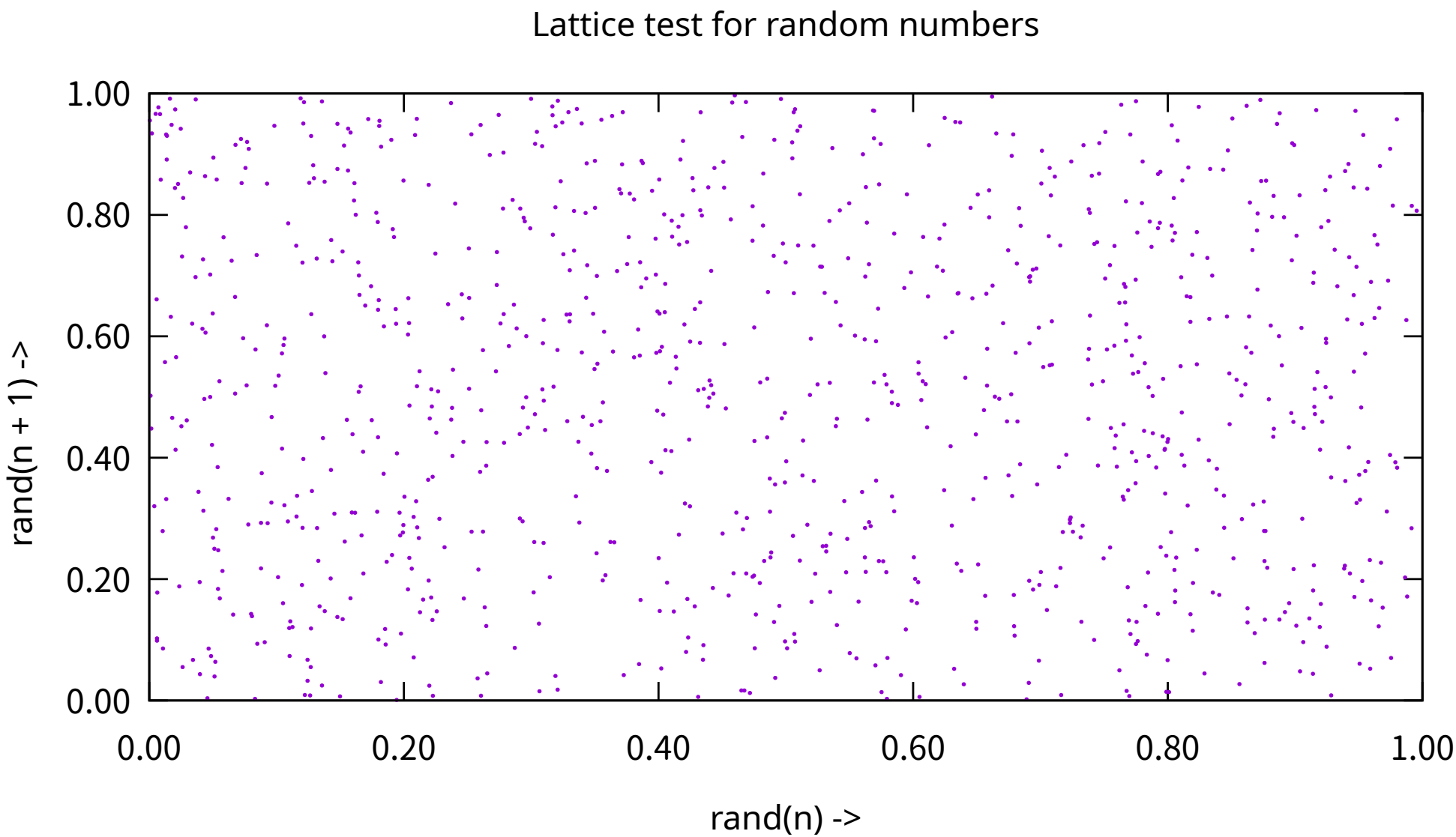

hypergeometric PDF using normal approximation
hypgeo(rnd(x), nn, mm, n)
normal(x, μ, σ)
σ
μ
probability density
k, x
0.30
0.27
0.24
0.21
0.18
0.15
0.12
0.09
0.06
0.03
0.00
-2
0
2
4
6
8
10

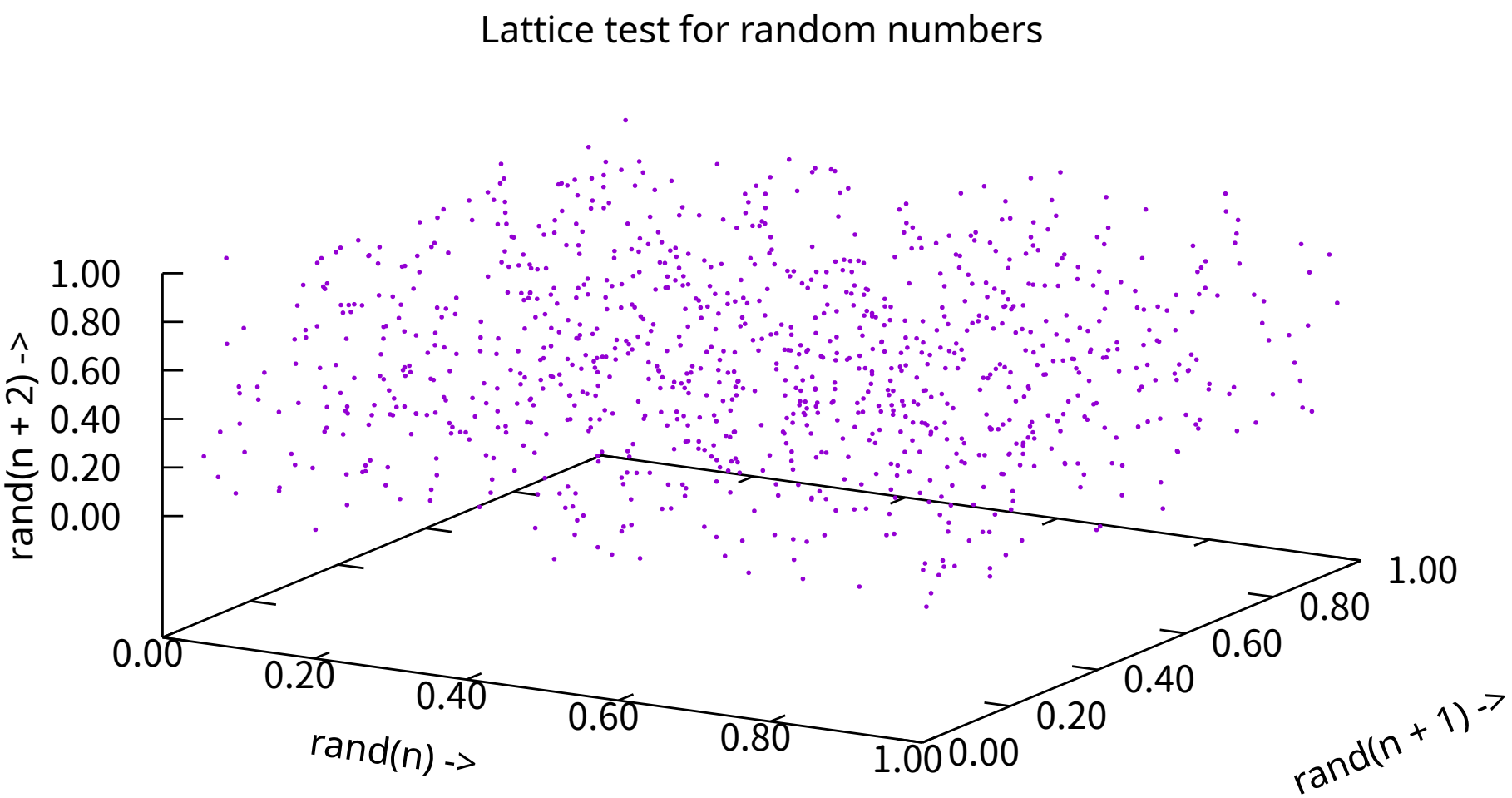

negative binomial PDF using gamma approximation
negbin(rnd(x), r, p)
gmm(x, ρ, λ)
probability density
0.17
0.15
0.14
0.12
0.10
0.09
0.07
0.05
0.03
0.02
0.00
0
5
10
15
k, x
σ
μ


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

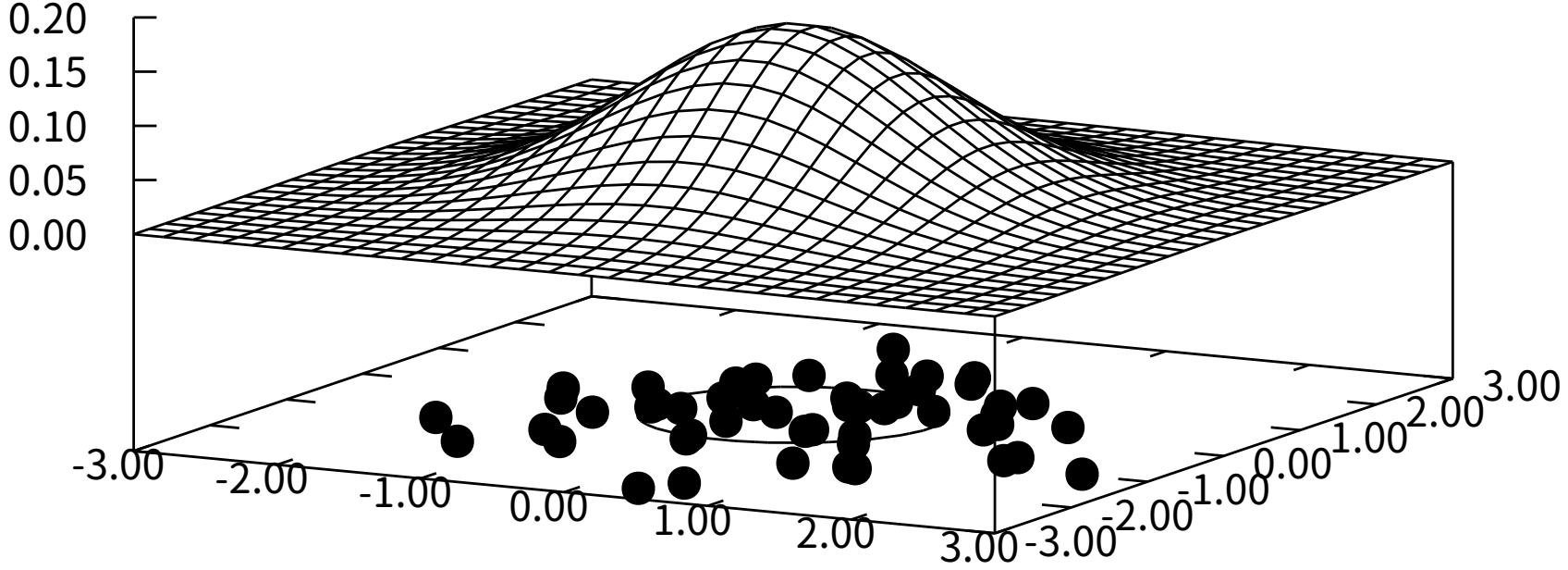

normal PDF using logistic approximation
logistic(x, a, λ)
normal(x, μ, σ)
σ
μ
probability density
0.33
0.30
0.27
0.23
0.20
0.17
0.13
0.10
0.07
0.03
0.00
-4.0
-2.0
0.0
2.0
4.0
6.0
x

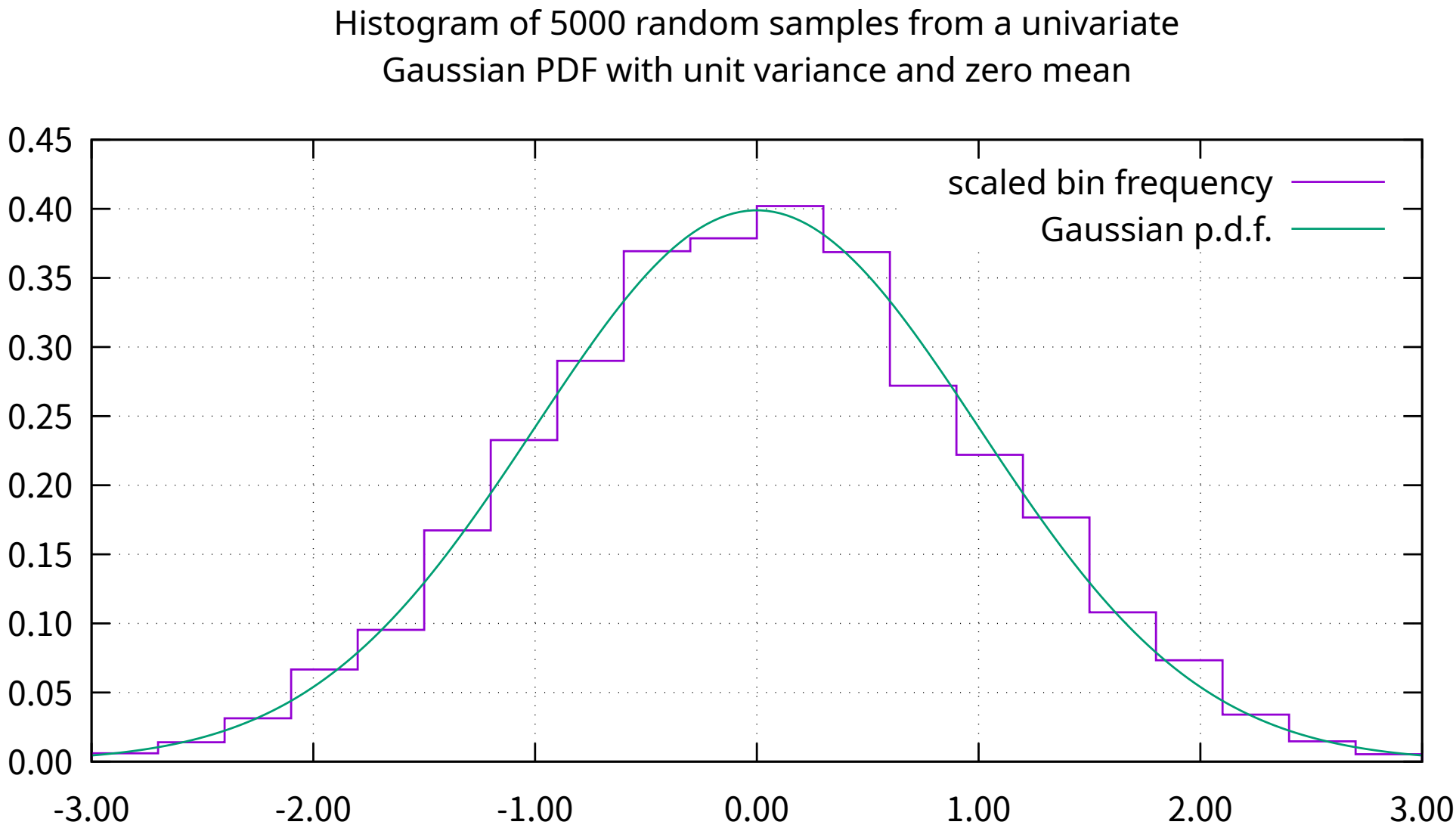

poisson PDF using normal approximation
poisson(rnd(x), μ)
normal(x, μ, σ)
σ
μ
probability density
k, x
0.19
0.17
0.15
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
-2
0
2
4
6
8
10
12
14


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) ->

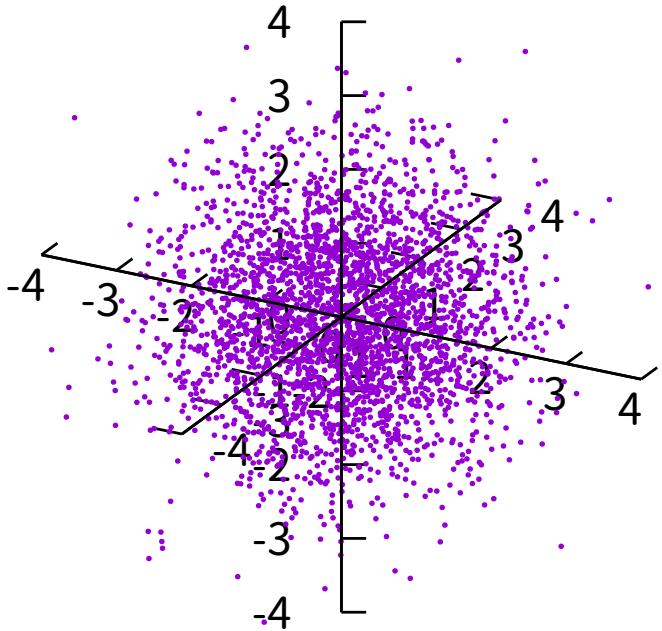

Lattice test for random numbers
rand(n + 2) ->
1.00
0.80
0.60
0.40
0.20
0.00
0.00
0.20
0.40
0.60
0.80
1.00
rand(n) ->
0.00
0.20
0.40
0.60
0.80
1.00
rand(n + 1) ->

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

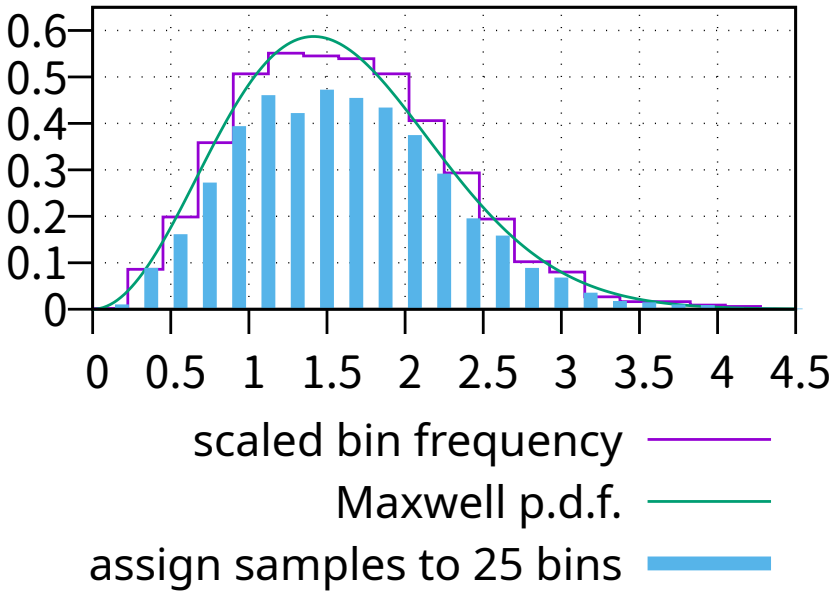



Histogram of 5000 random samples from a univariate
Gaussian PDF with unit variance and zero mean
scaled bin frequency
Gaussian p.d.f.
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-3.00
-2.00
-1.00
0.00
1.00
2.00
3.00

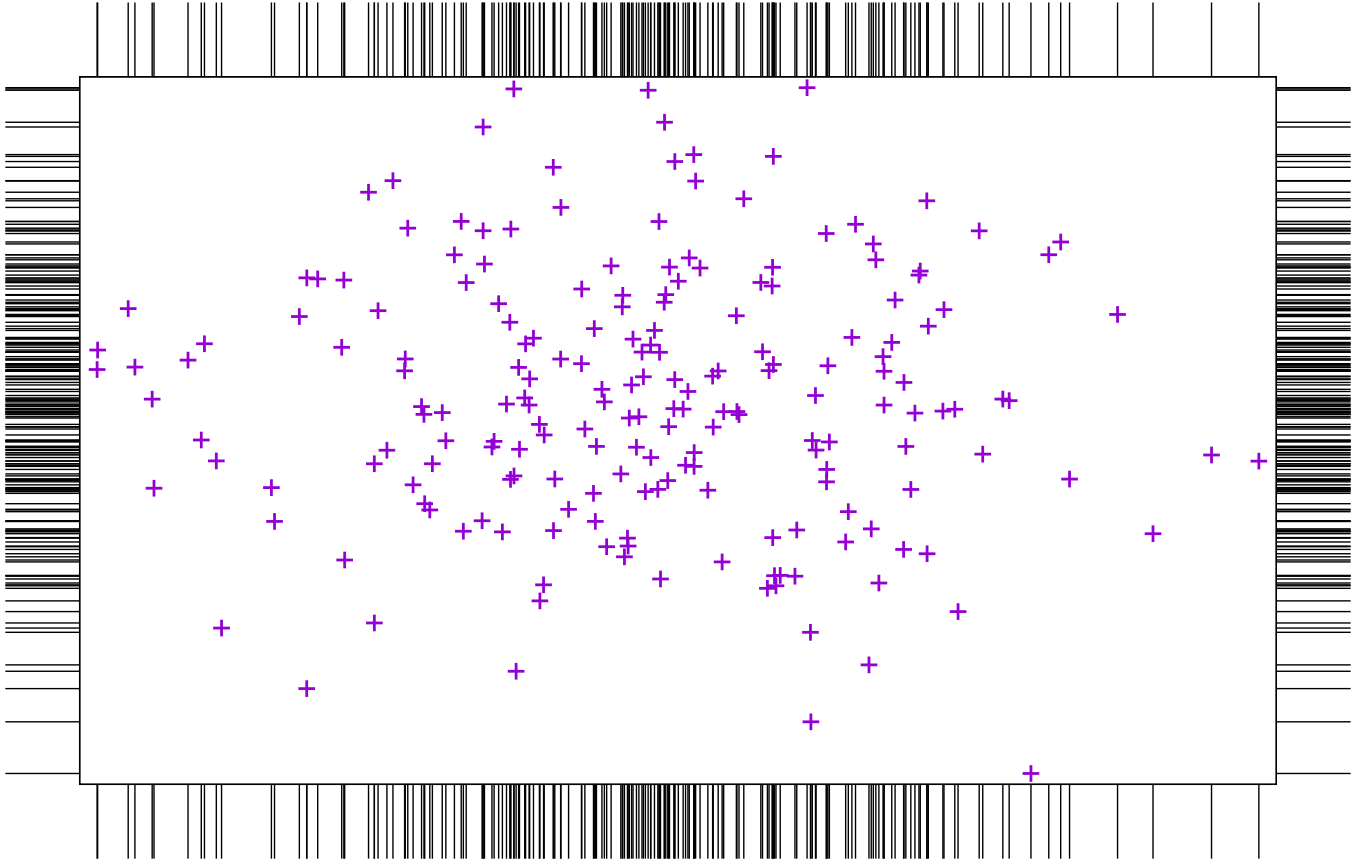
Gaussian 3D cloud of 3000 random samples

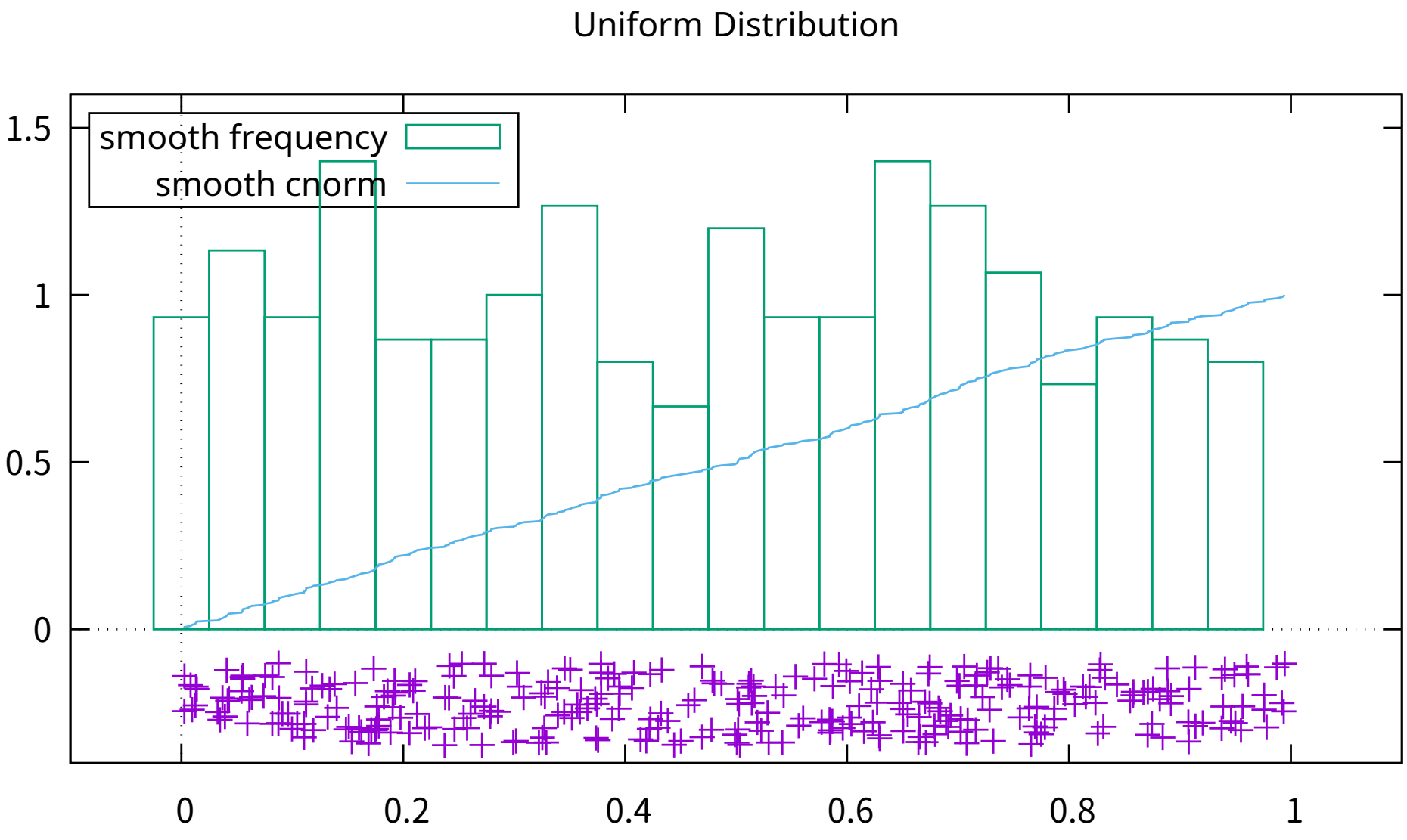


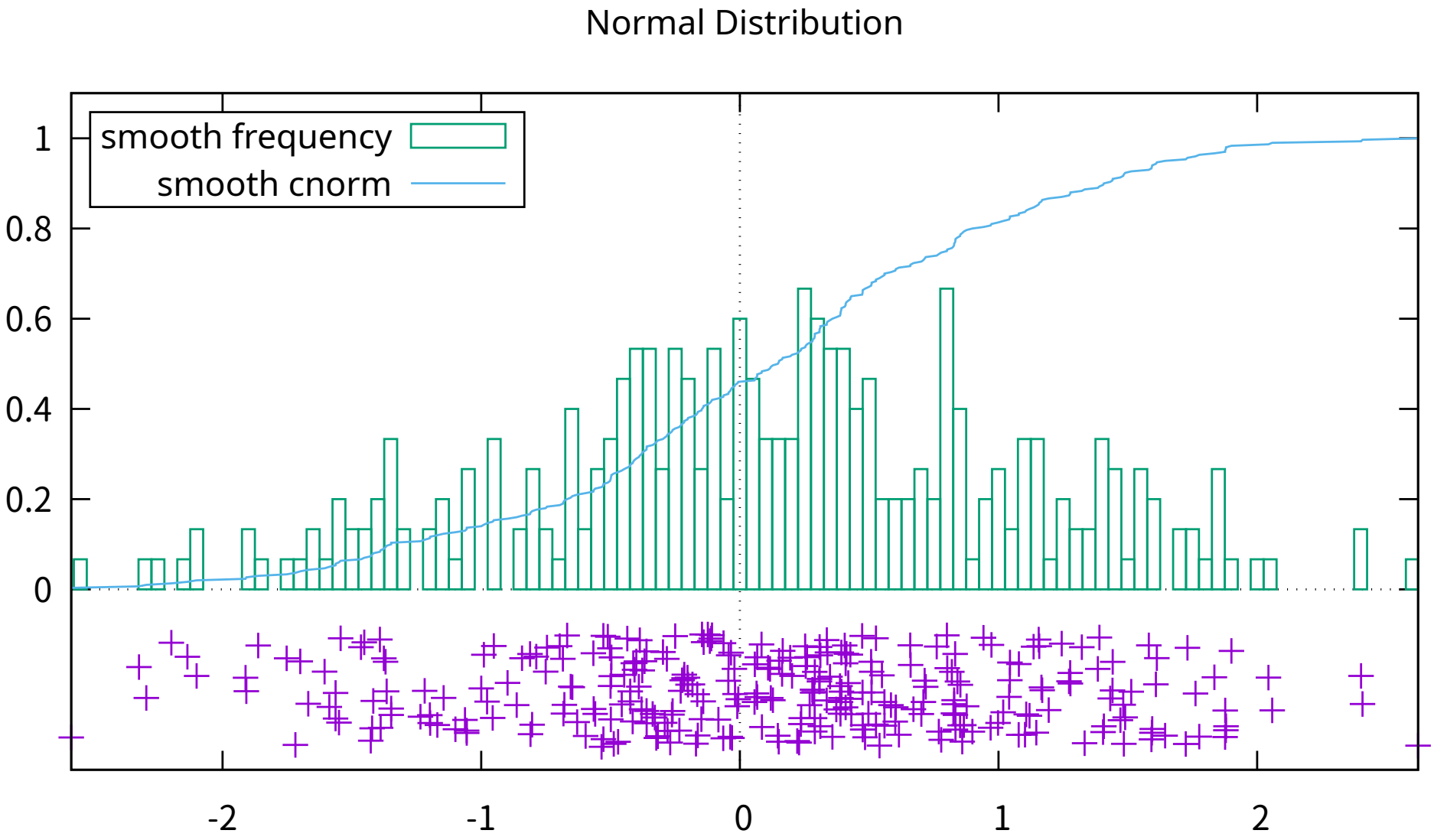
Histogram of distance from origin of 3000 multivariate unit variance samples

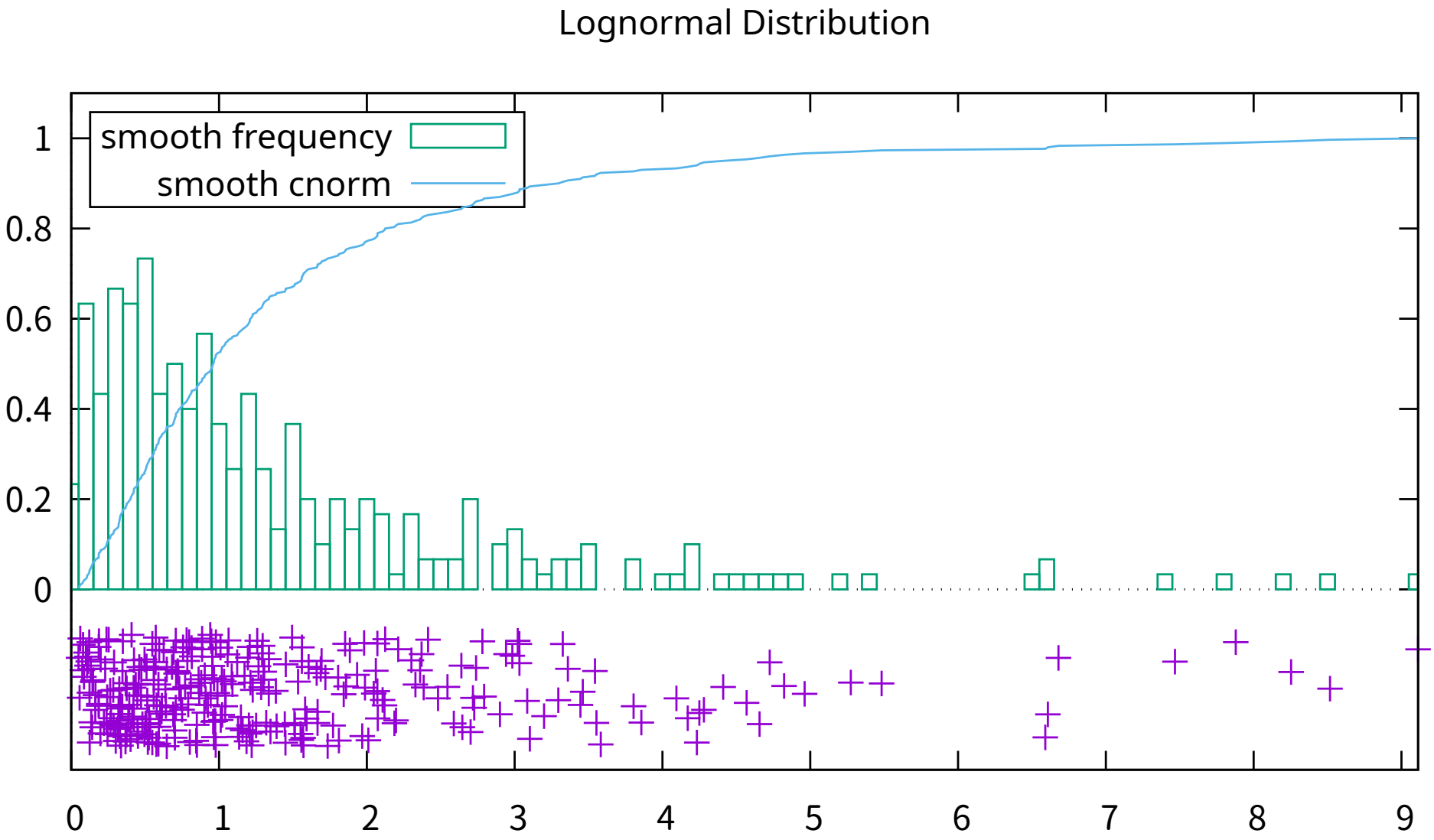


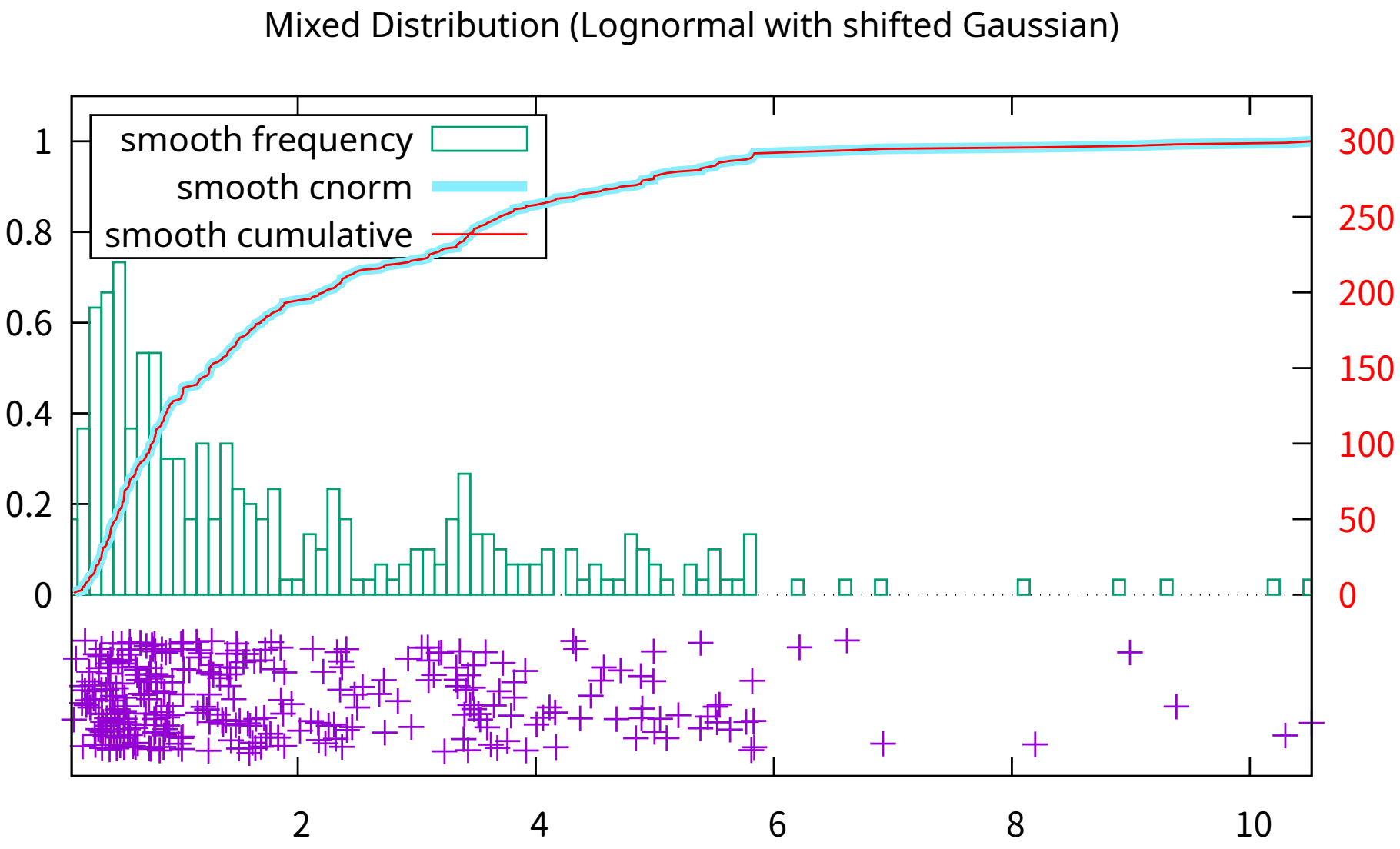
Rug Plot

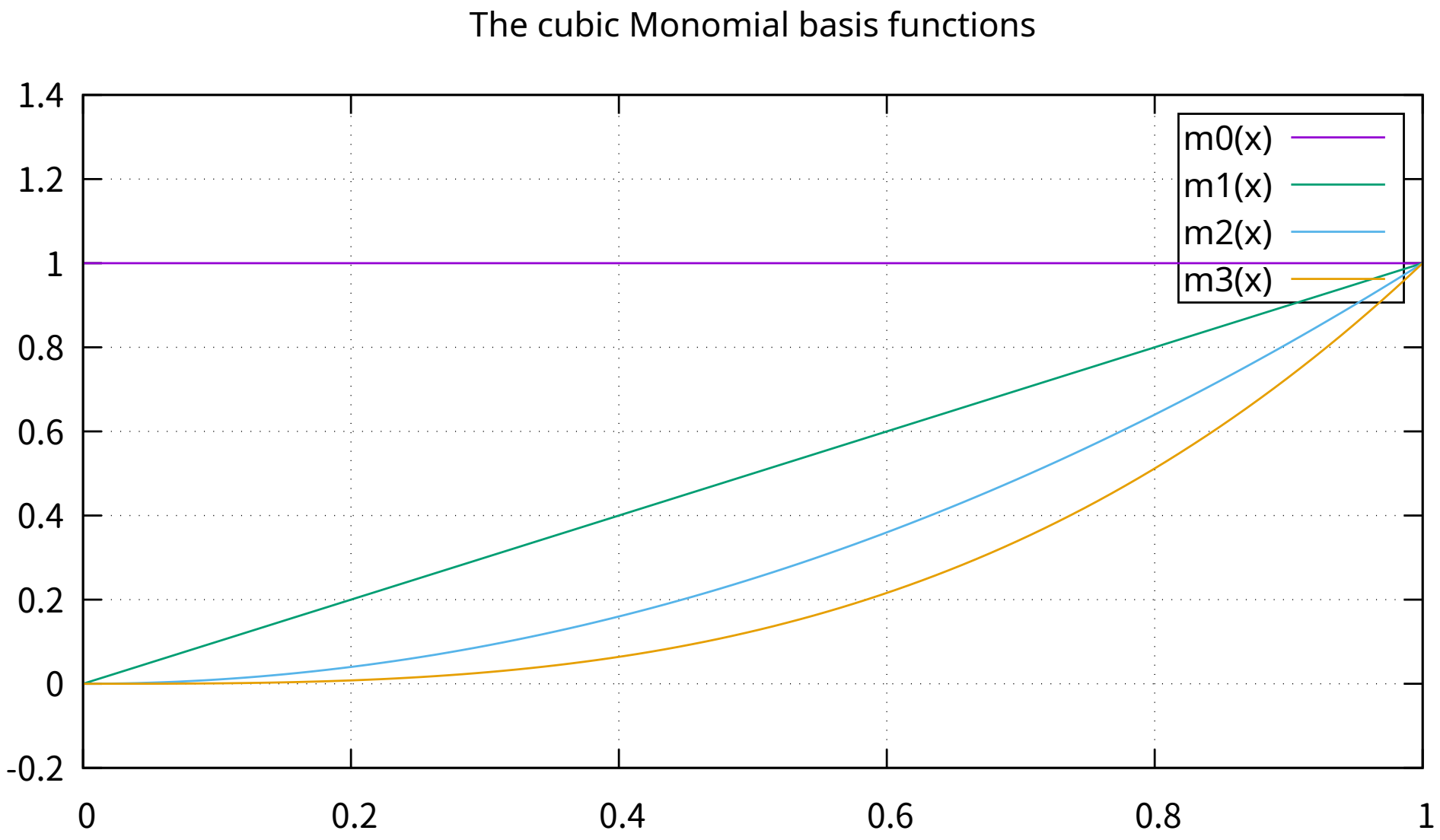


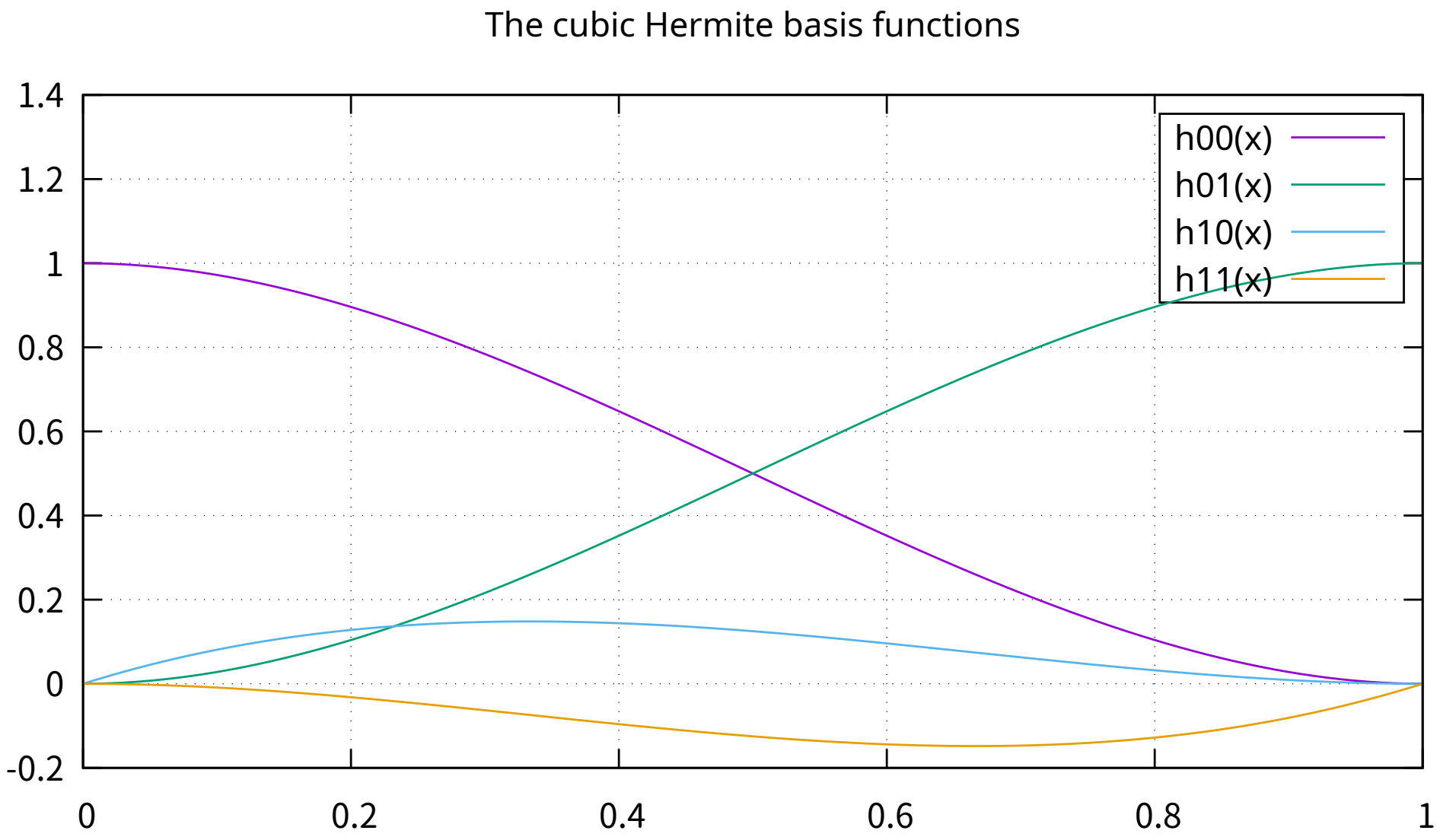

Uniform Distribution
smooth frequency
smooth cnorm
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1

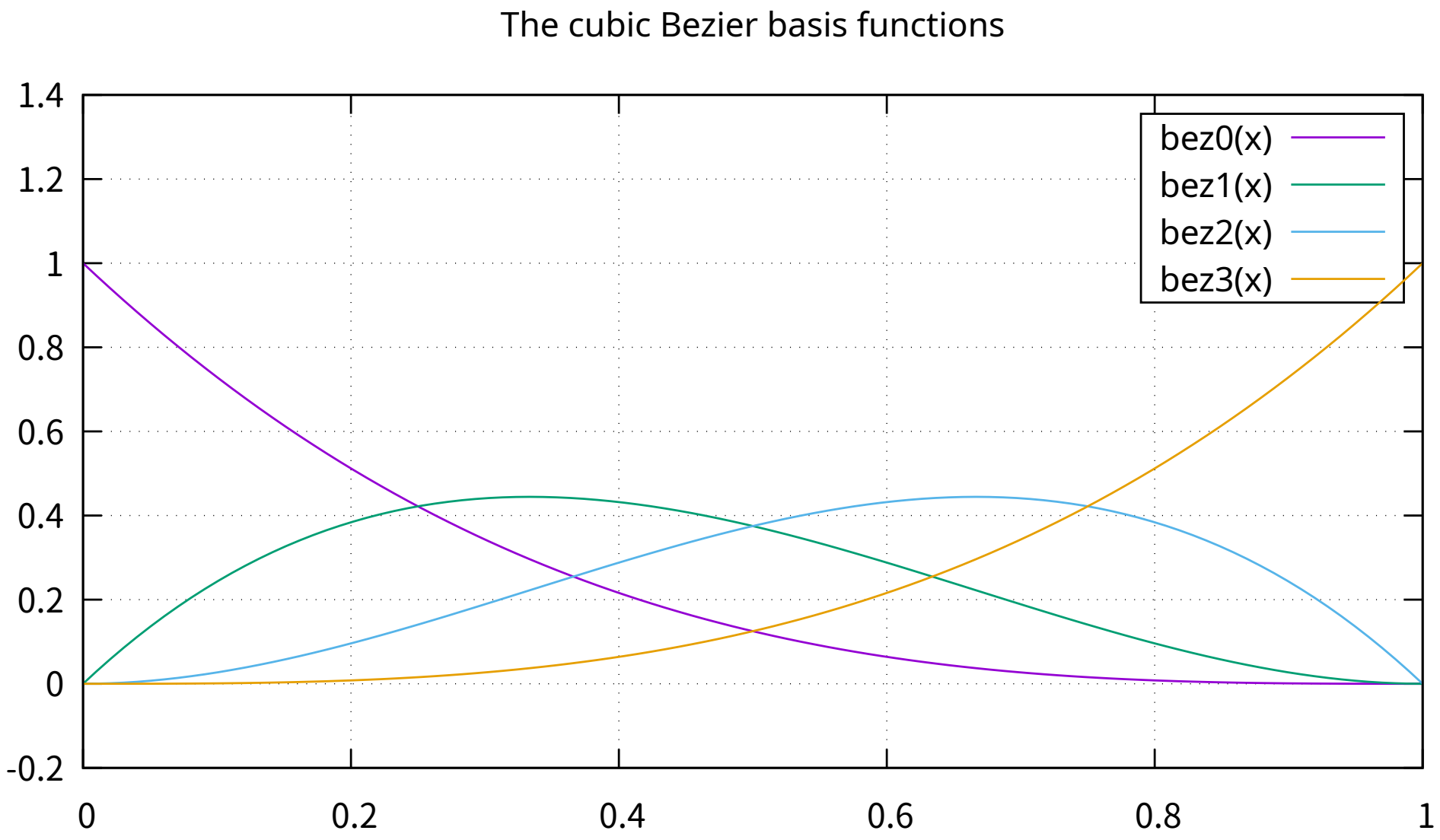

Normal Distribution
smooth frequency
smooth cnorm
1
0.8
0.6
0.4
0.2
0
-2
-1
0
1
2

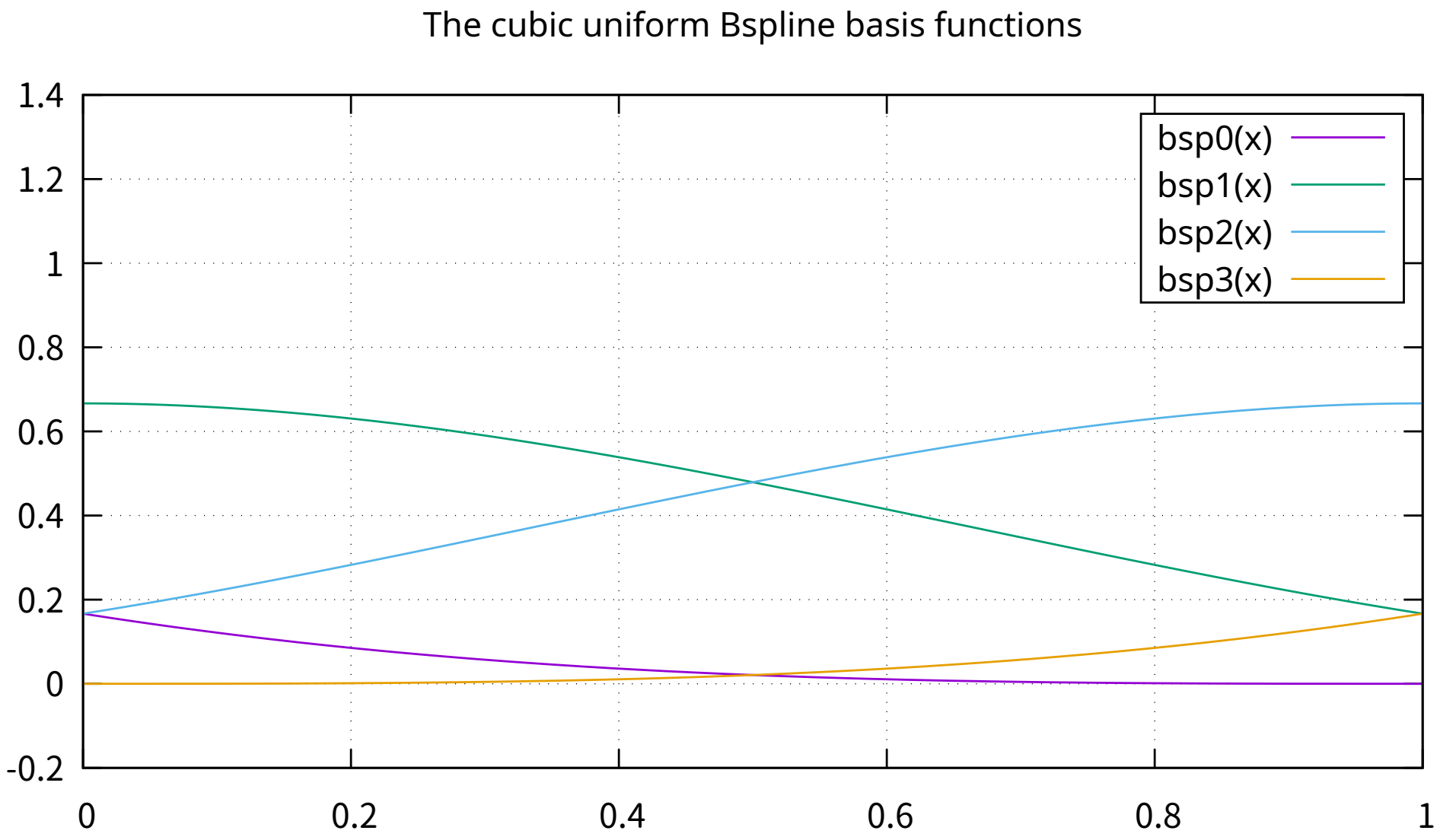

Lognormal Distribution
smooth frequency
smooth cnorm
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
7
8
9

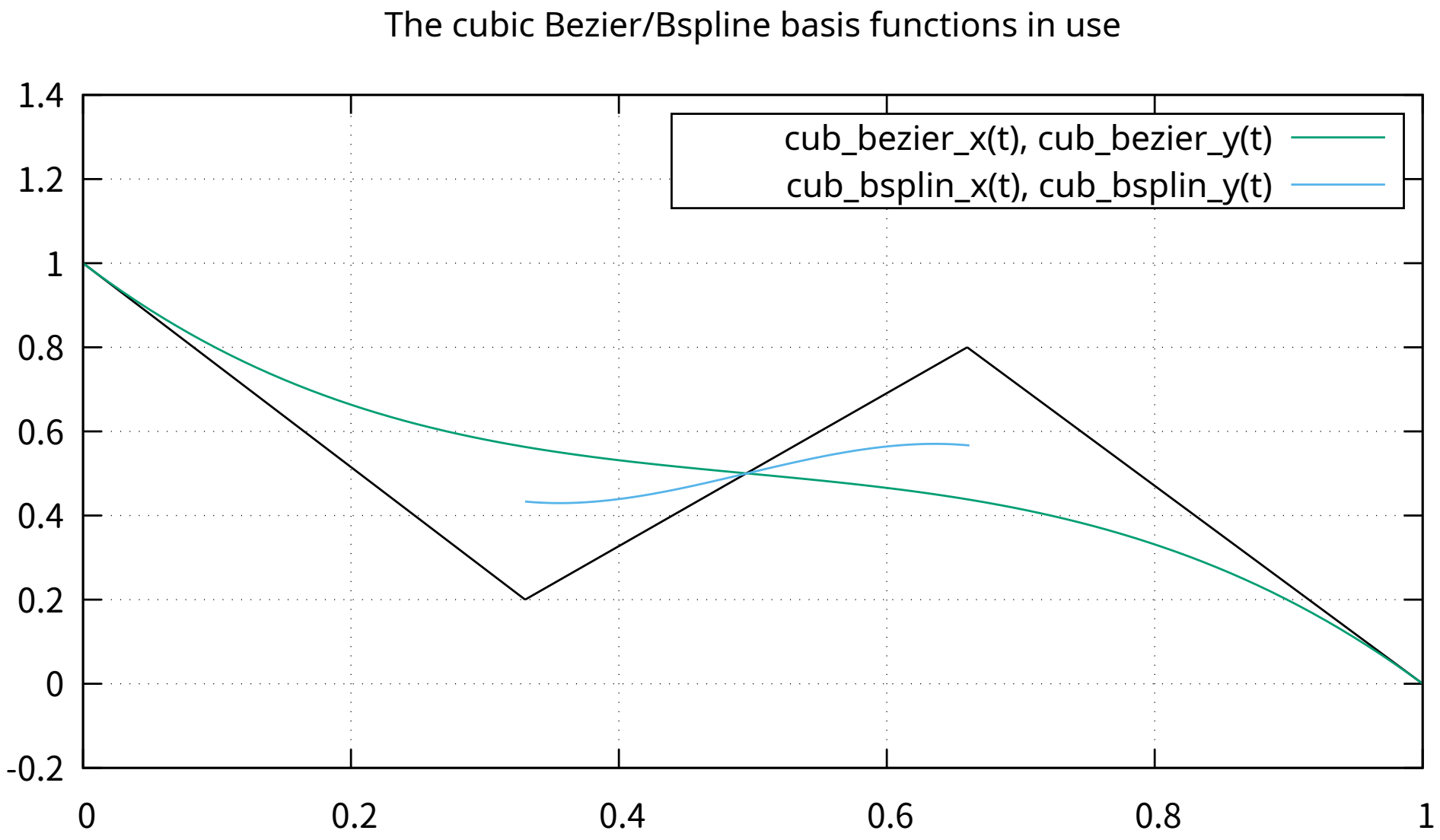

Mixed Distribution (Lognormal with shifted Gaussian)
smooth frequency
smooth cnorm
smooth cumulative
1
0.8
0.6
0.4
0.2
0
300
250
200
150
100
50
0
2
4
6
8
10

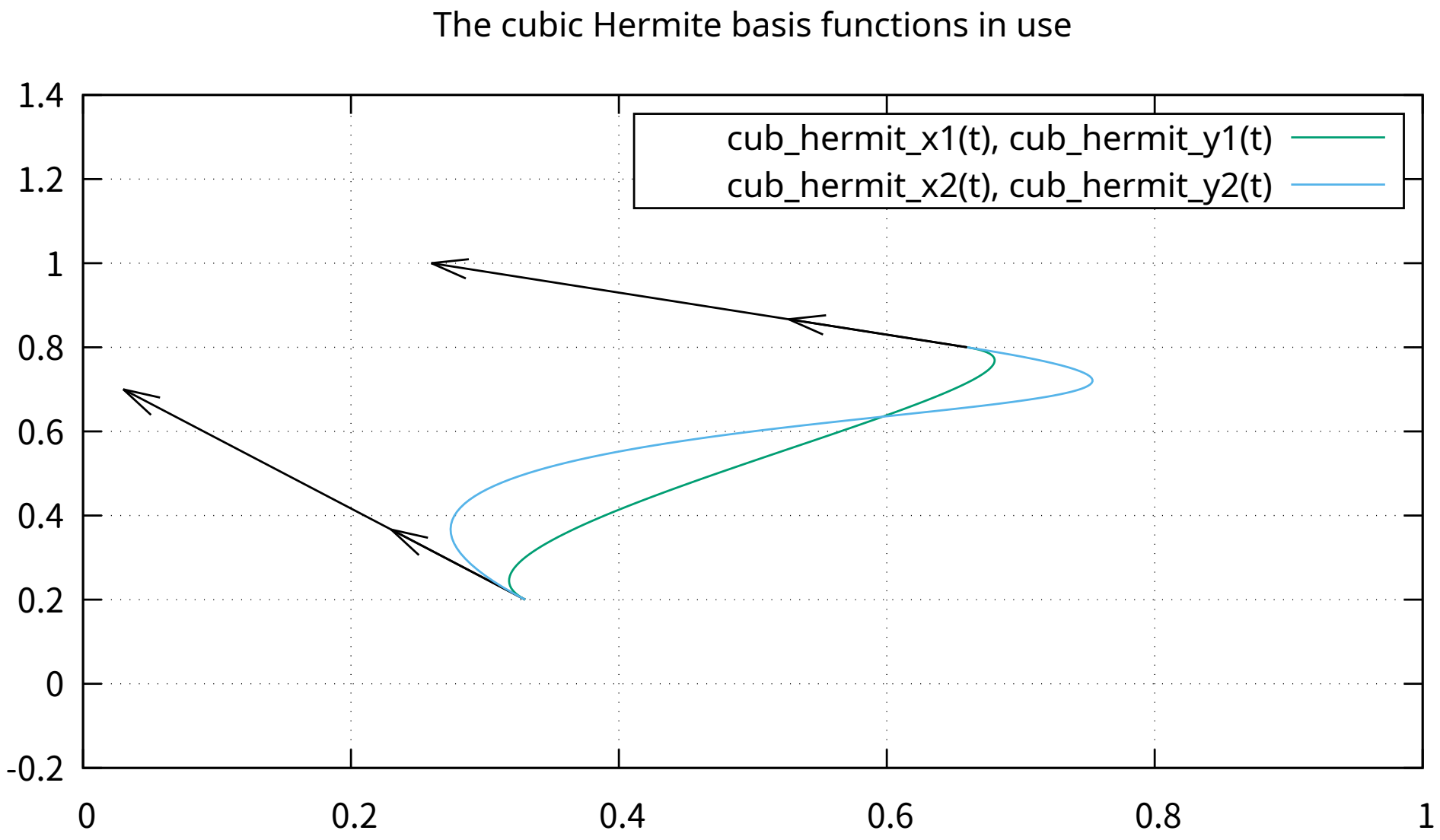

The cubic Monomial basis functions
m0(x)
m1(x)
m2(x)
m3(x)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1

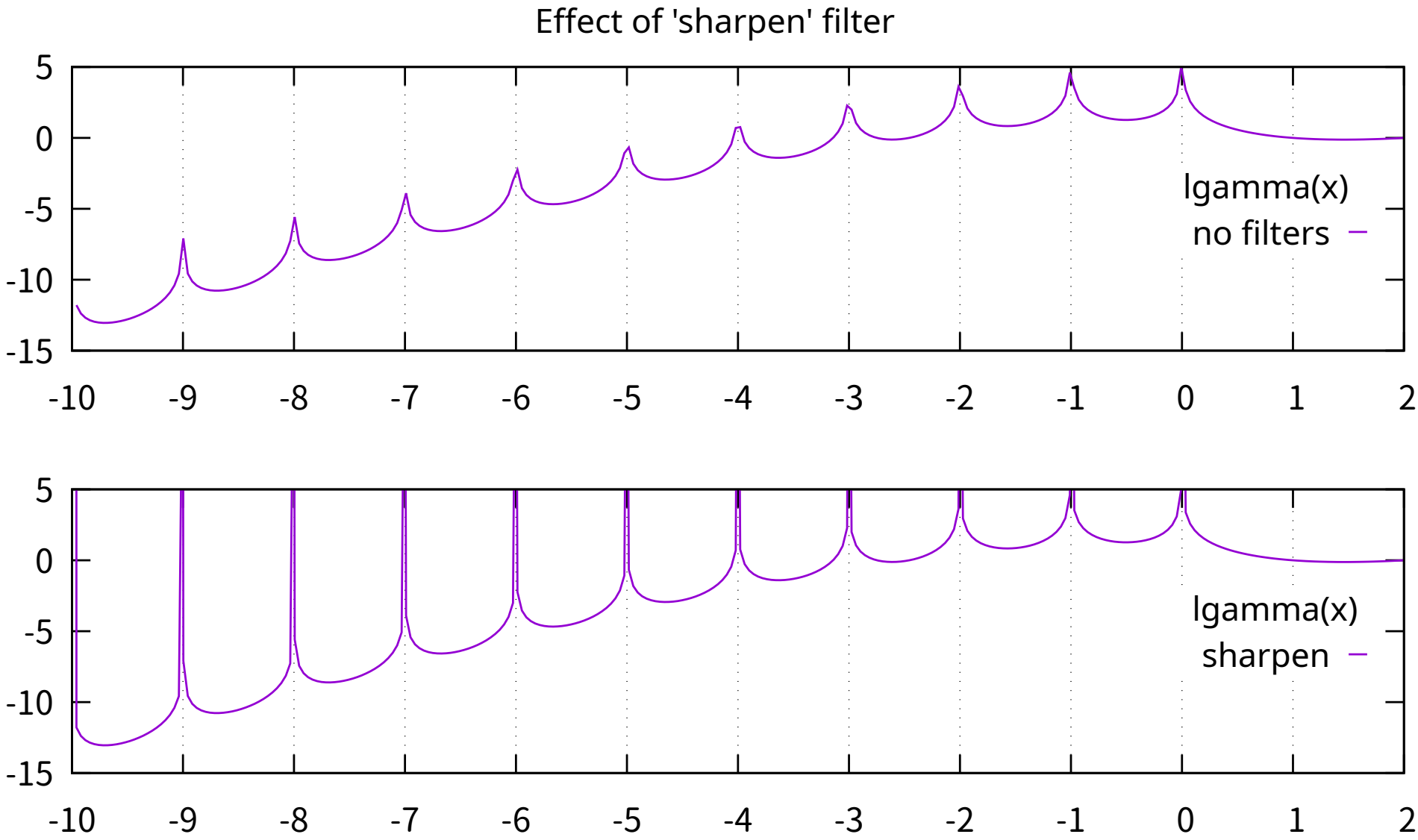

The cubic Hermite basis functions
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1
h00(x)
h01(x)
h10(x)
h11(x)


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

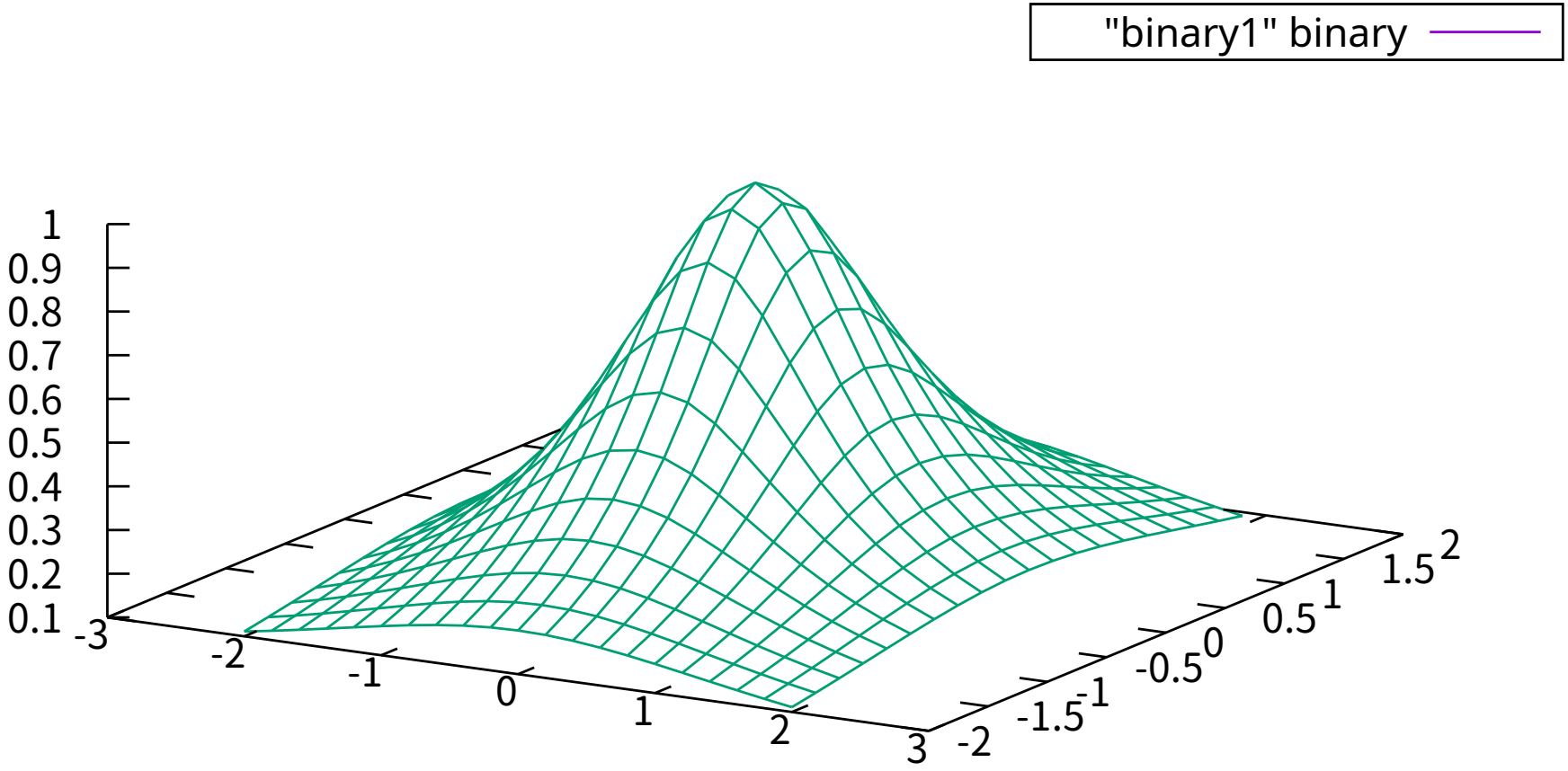

The cubic uniform Bspline basis functions
bsp0(x)
bsp1(x)
bsp2(x)
bsp3(x)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1

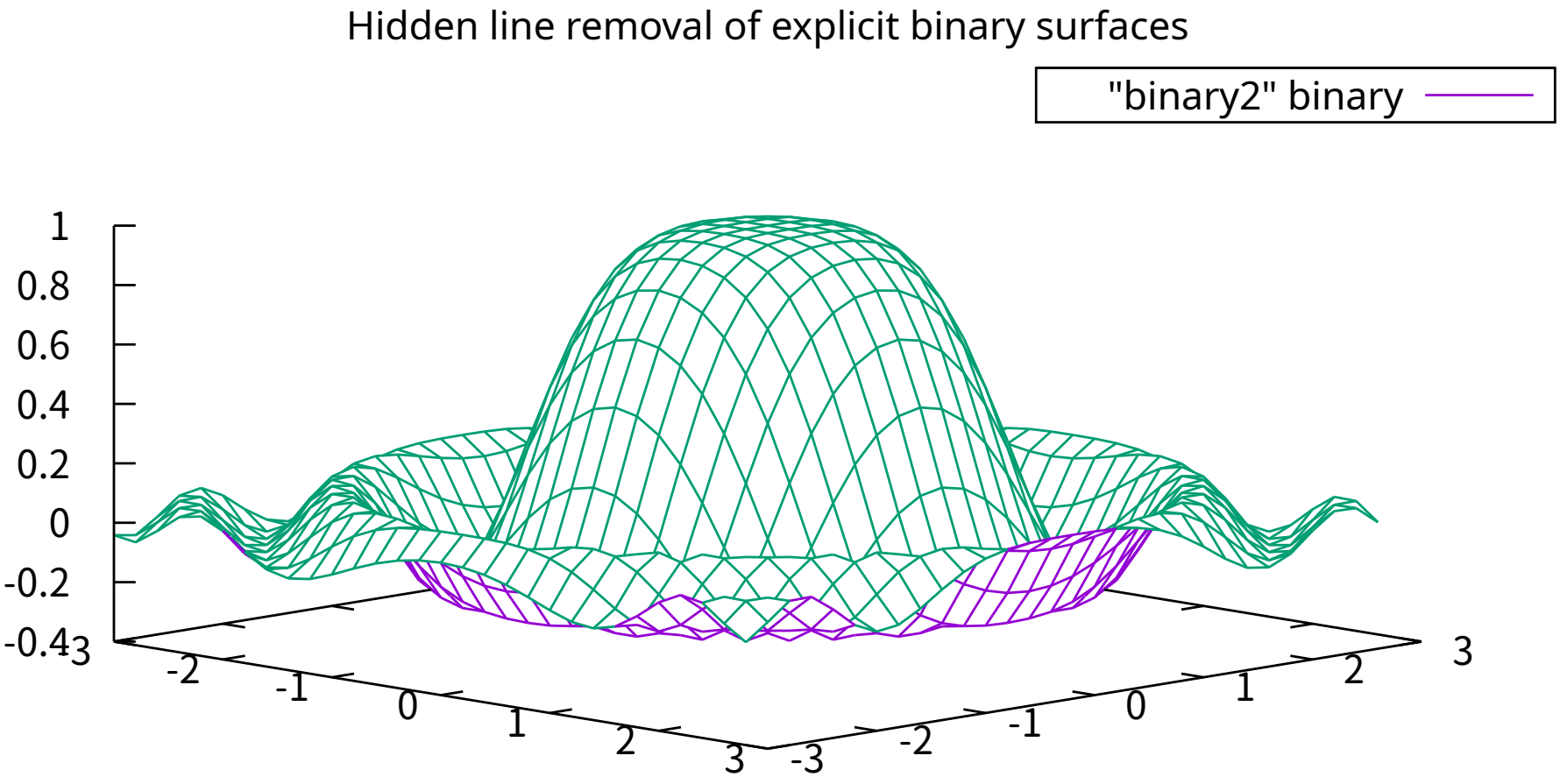

The cubic Bezier/Bspline basis functions in use
cub_bezier_x(t), cub_bezier_y(t)
cub_bsplin_x(t), cub_bsplin_y(t)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1


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

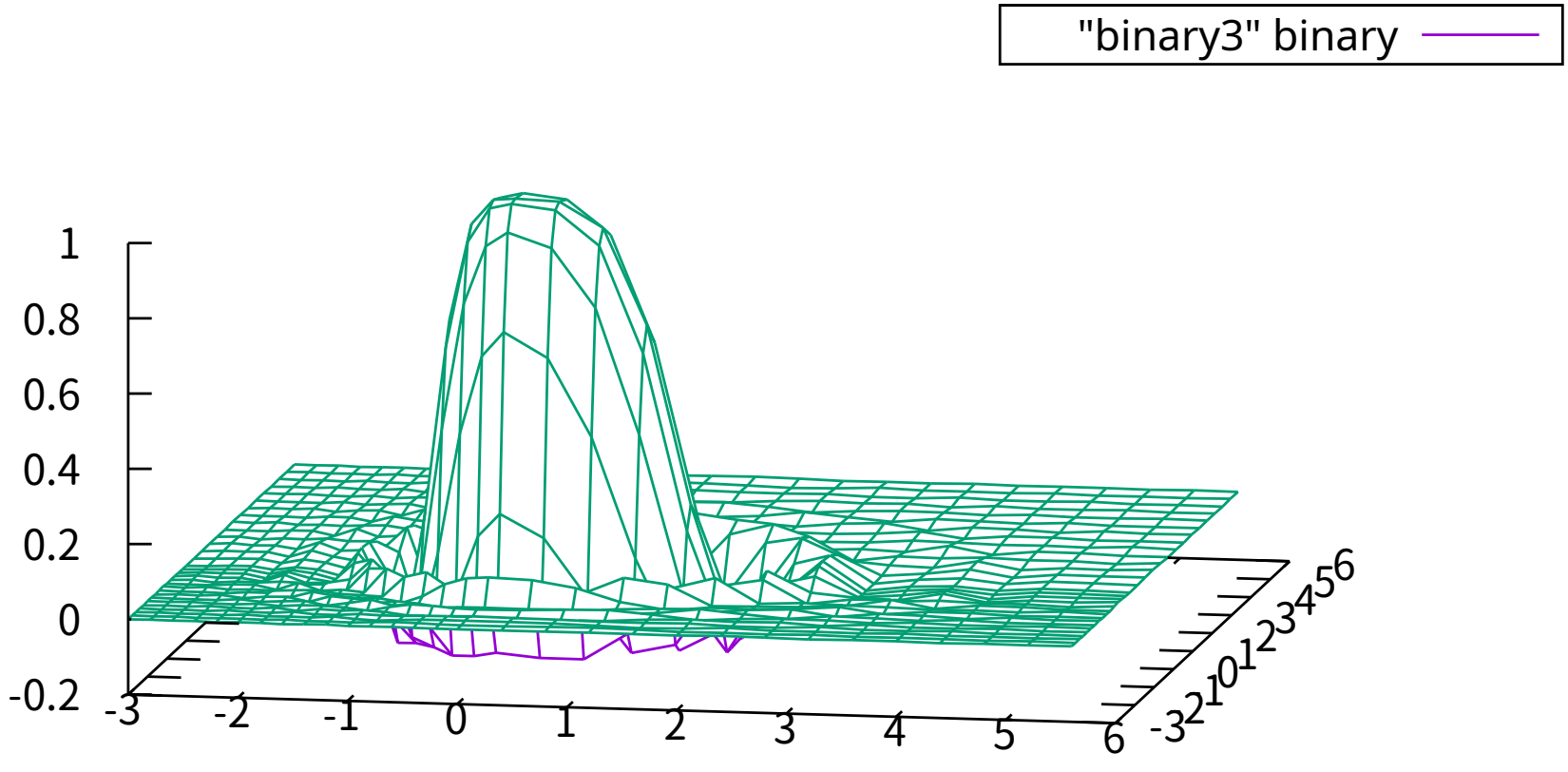

Effect of 'sharpen' filter
5
0
-5
-10
-15
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1
0
1
2
lgamma(x)
no filters
lgamma(x)
sharpen

Hidden line removal of explicit binary surfaces



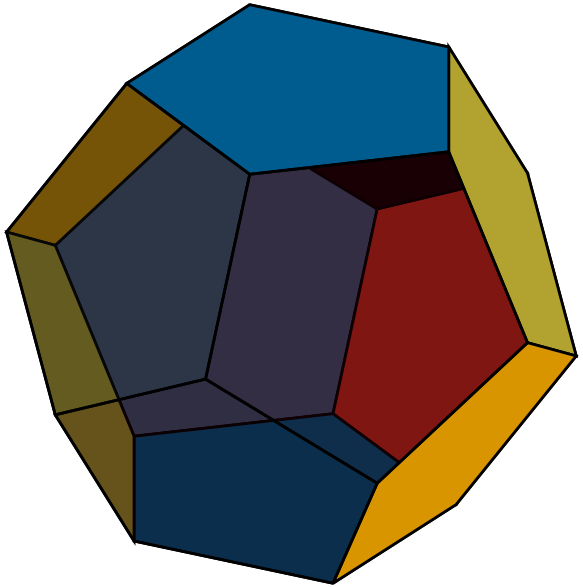

Hidden line removal of explicit binary surfaces
"binary2" binary
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
3
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3

Notice that sampling rate can change

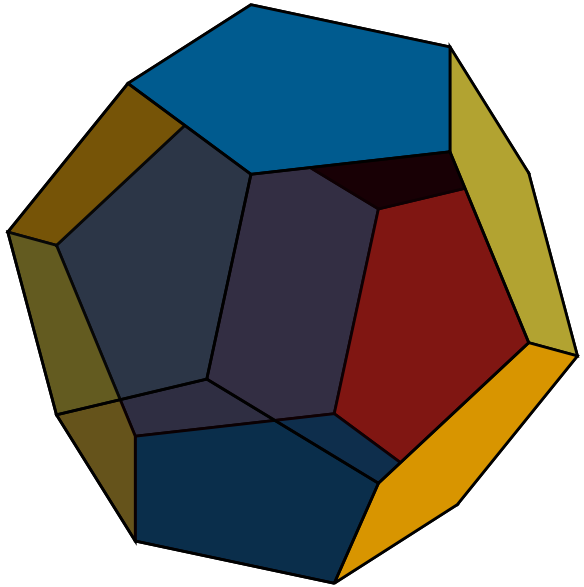


Compare text and binary input data

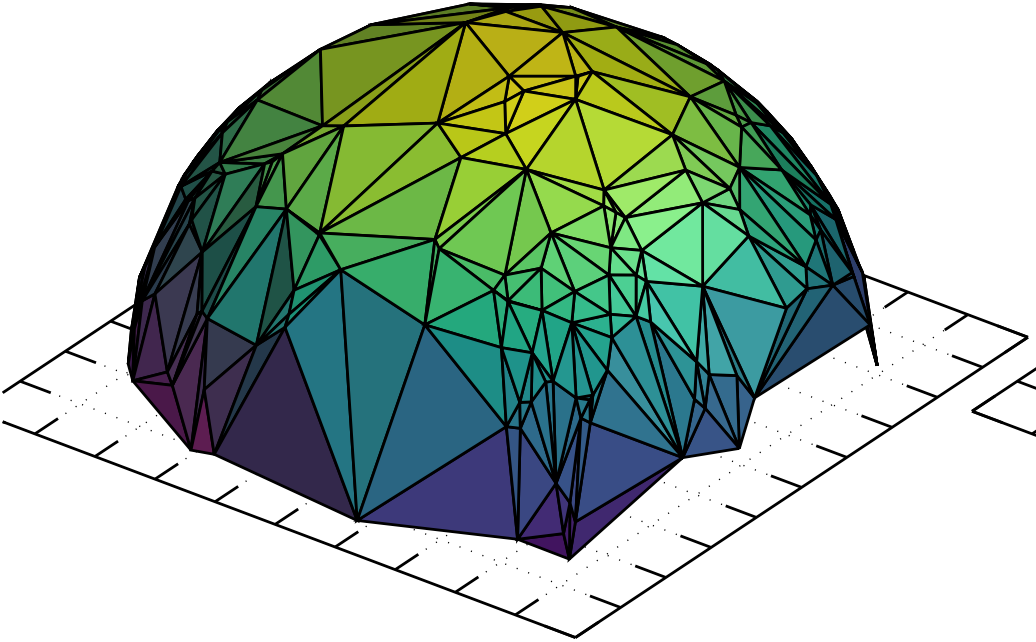
text input data
using 1:2:3:4 with polygons lc variable



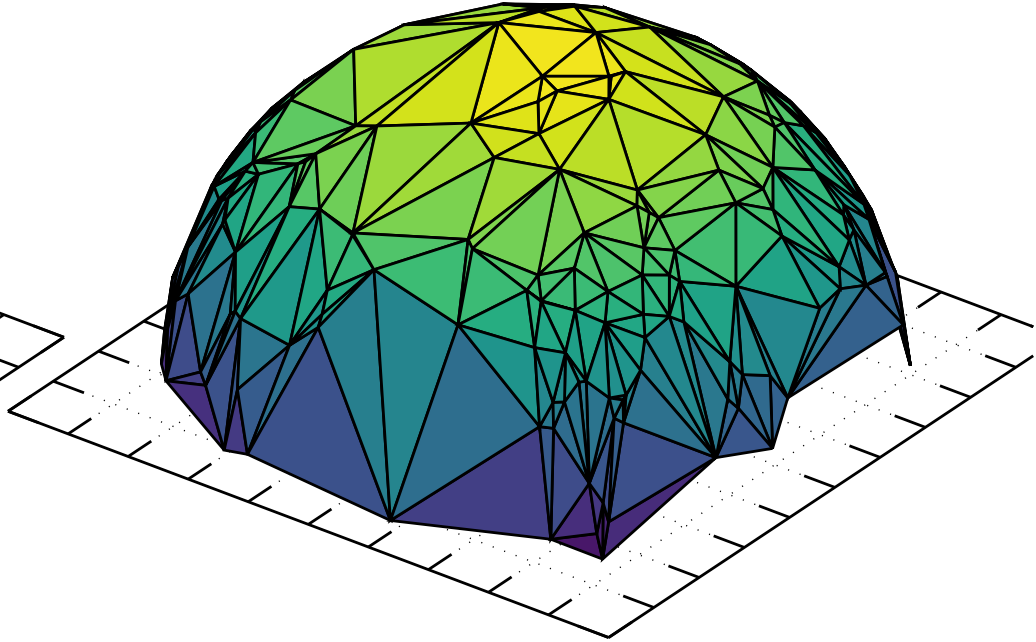
binary format='%3float64%int32' blank=NaN
using 1:2:3:4 with polygons lc variable

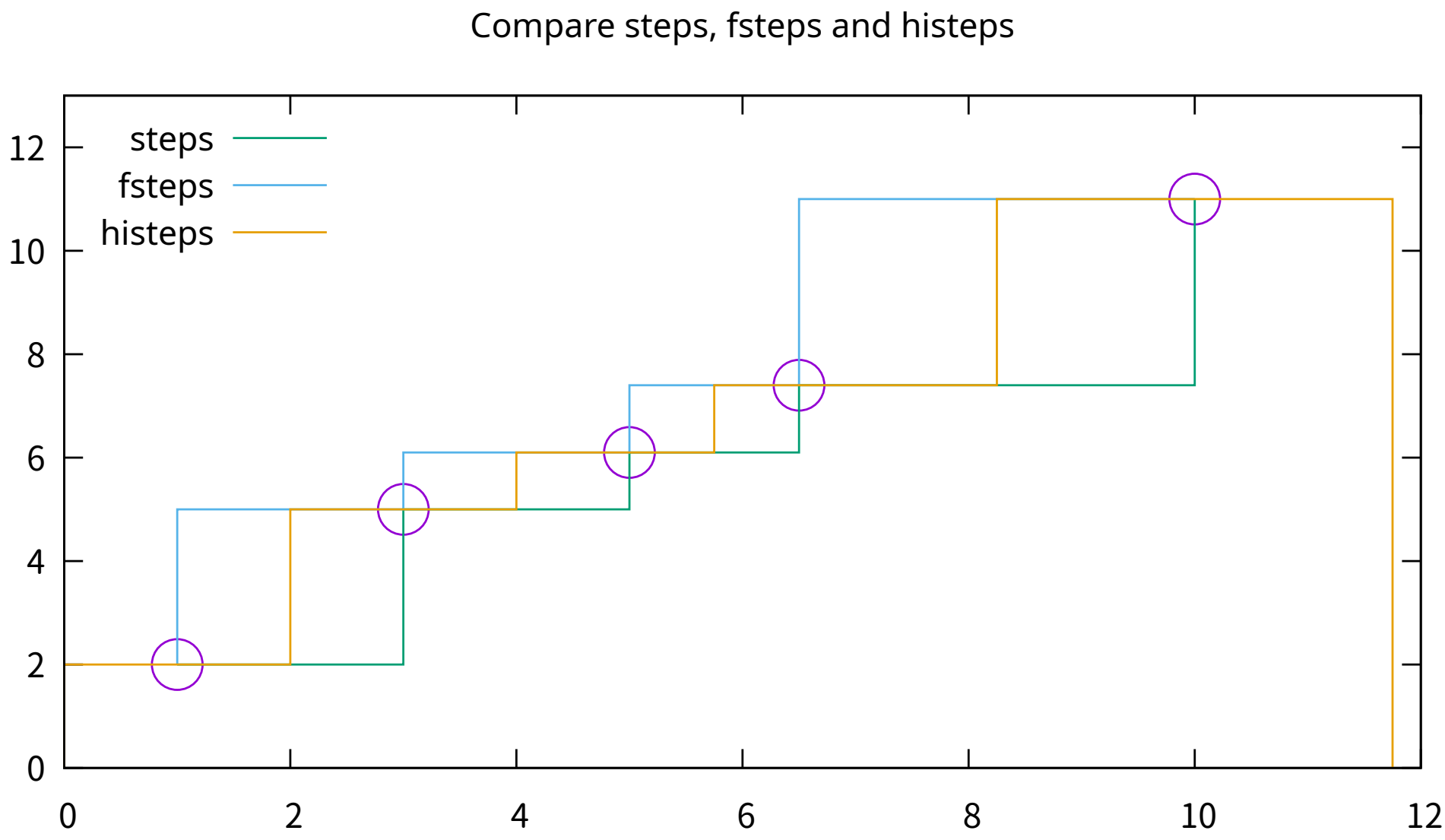


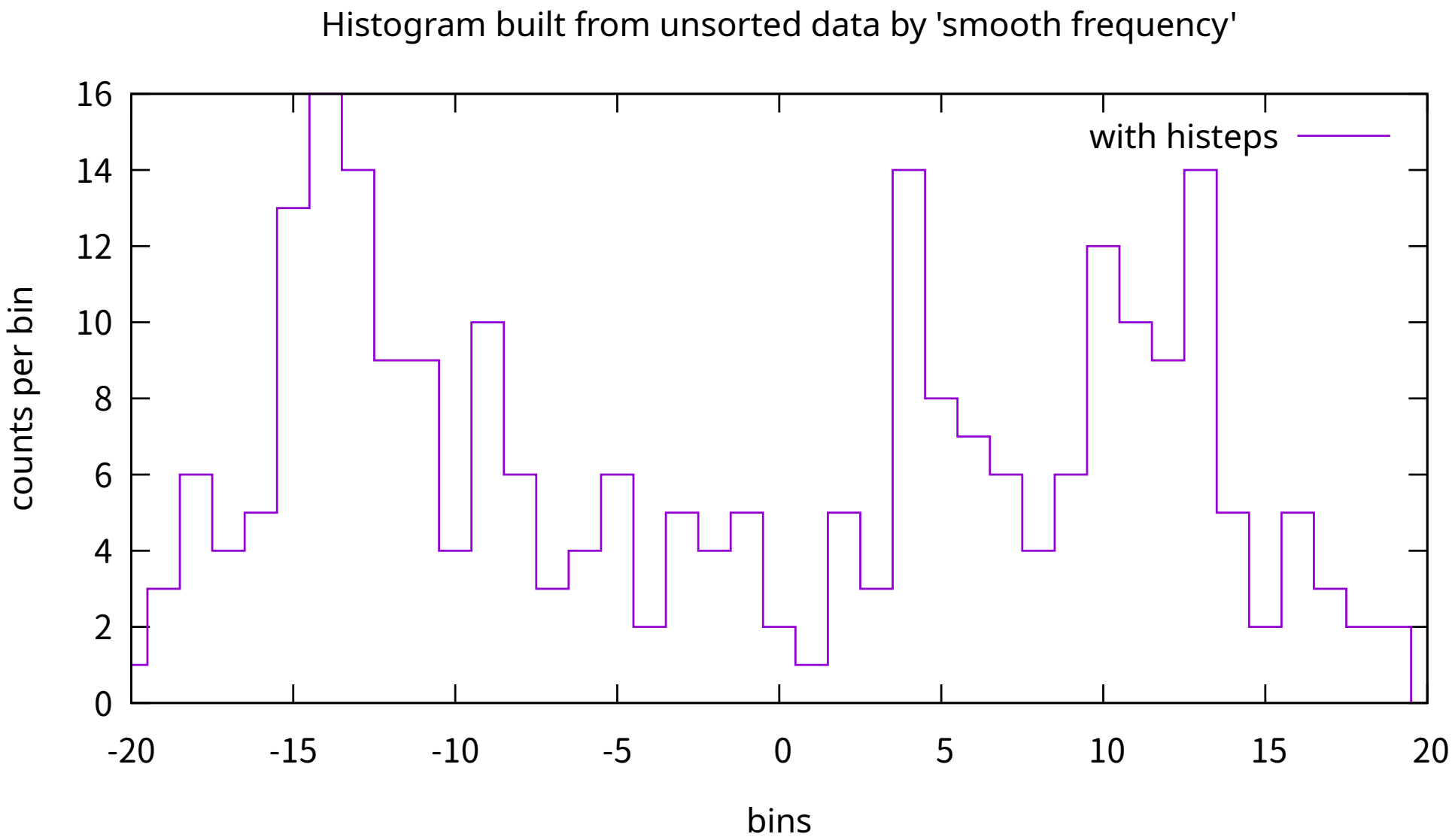
pm3d lighting

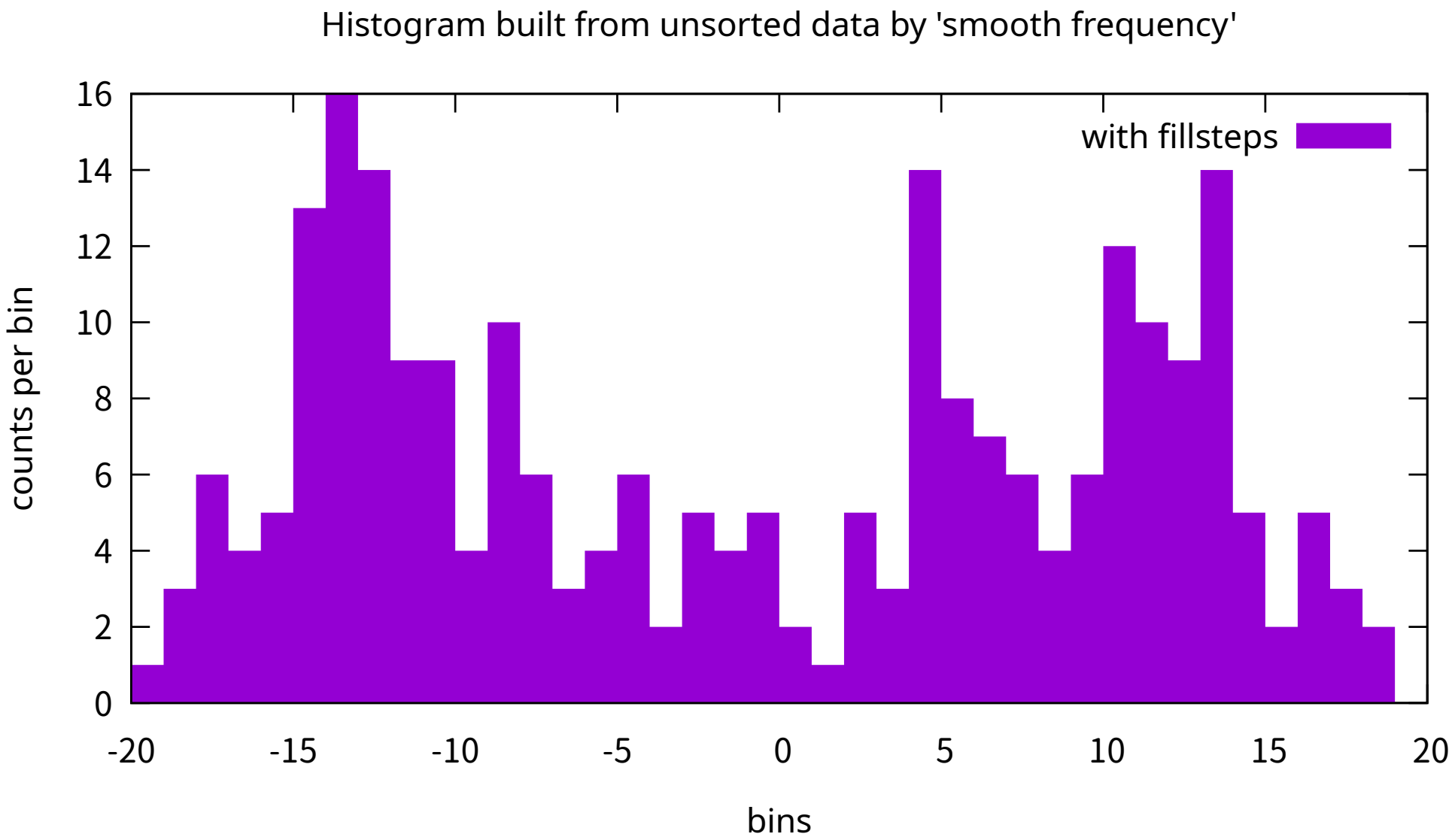


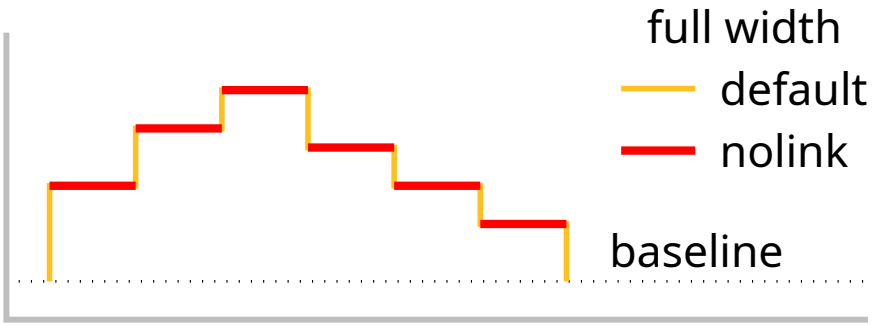
no lighting model

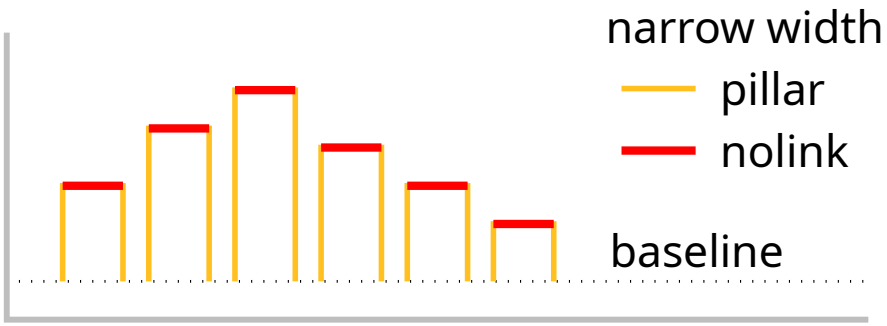


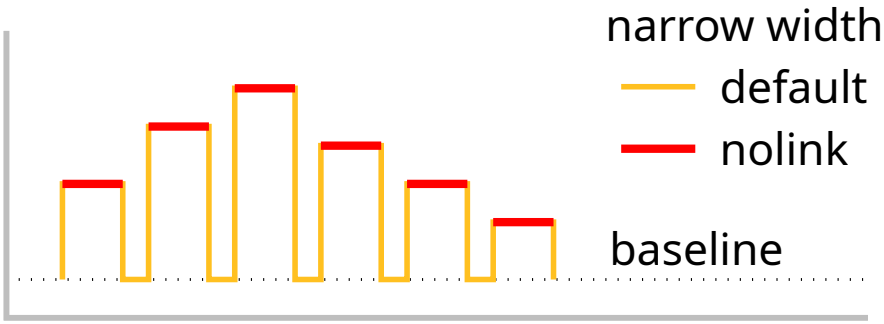

Compare steps, fsteps and histeps
steps
fsteps
histeps
0
2
4
6
8
10
12
0
2
4
6
8
10
12

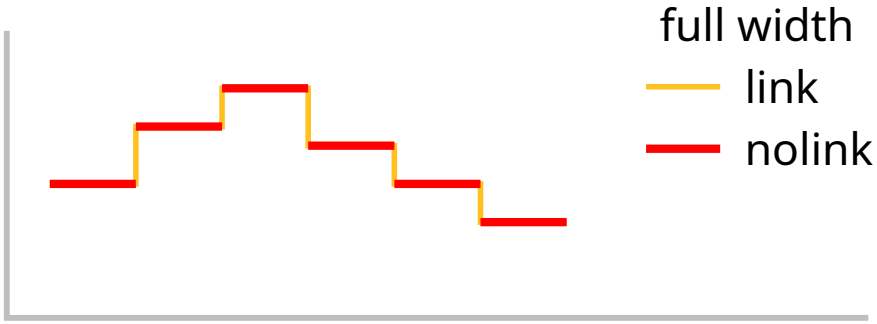

Histogram built from unsorted data by 'smooth frequency'
with histeps
counts per bin
bins
16
14
12
10
8
6
4
2
0
-20
-15
-10
-5
0
5
10
15
20

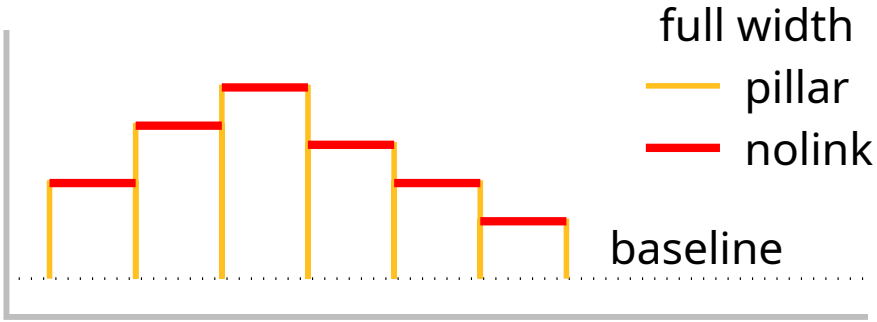

Histogram built from unsorted data by 'smooth frequency'
with fillsteps
counts per bin
bins
16
14
12
10
8
6
4
2
0
-20
-15
-10
-5
0
5
10
15
20

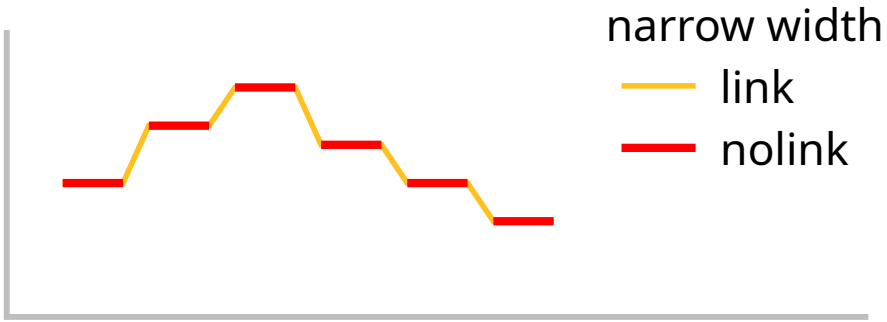

full width
default
nolink
baseline

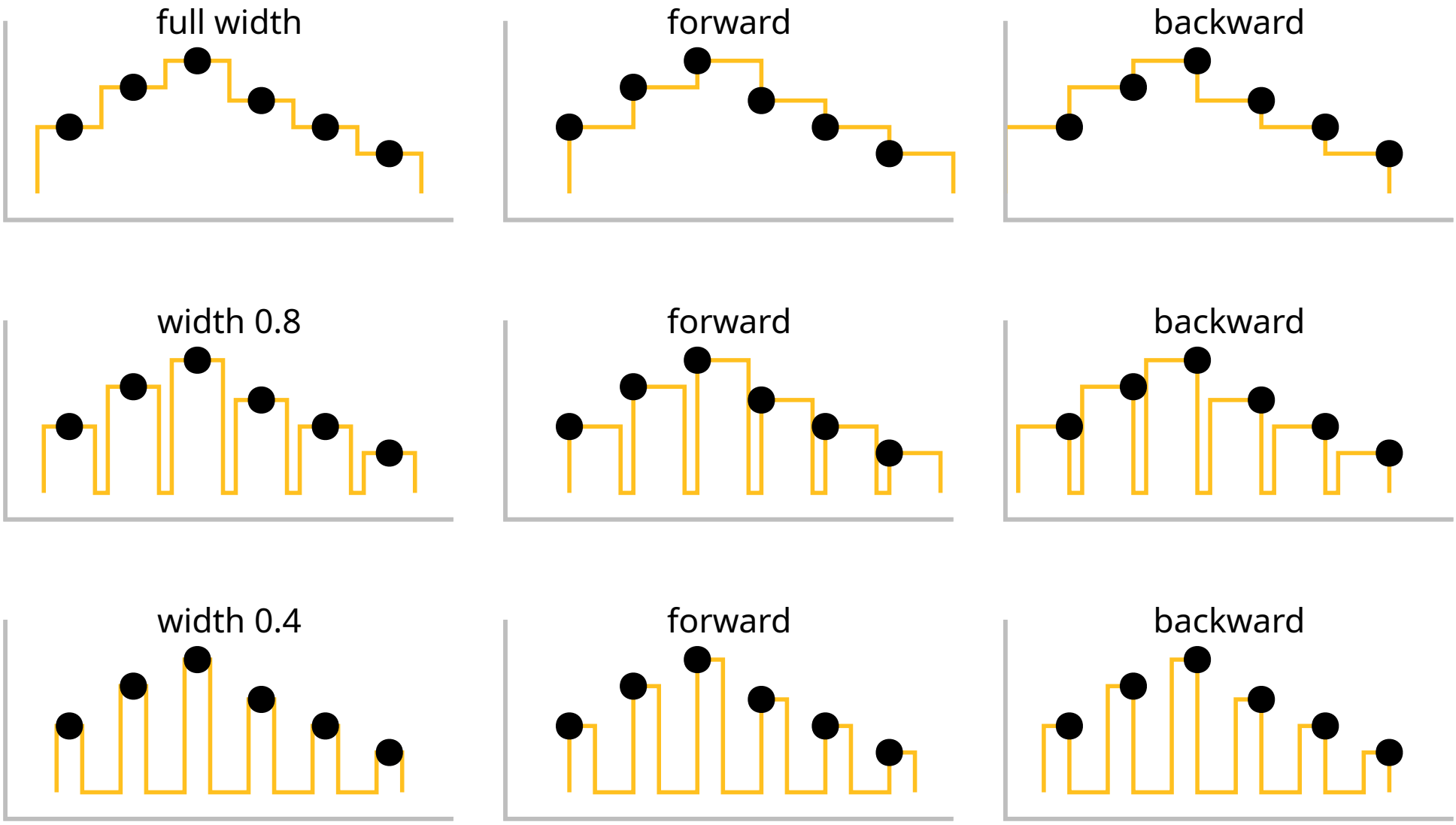

narrow width
pillar
nolink
baseline


narrow width
default
nolink
baseline

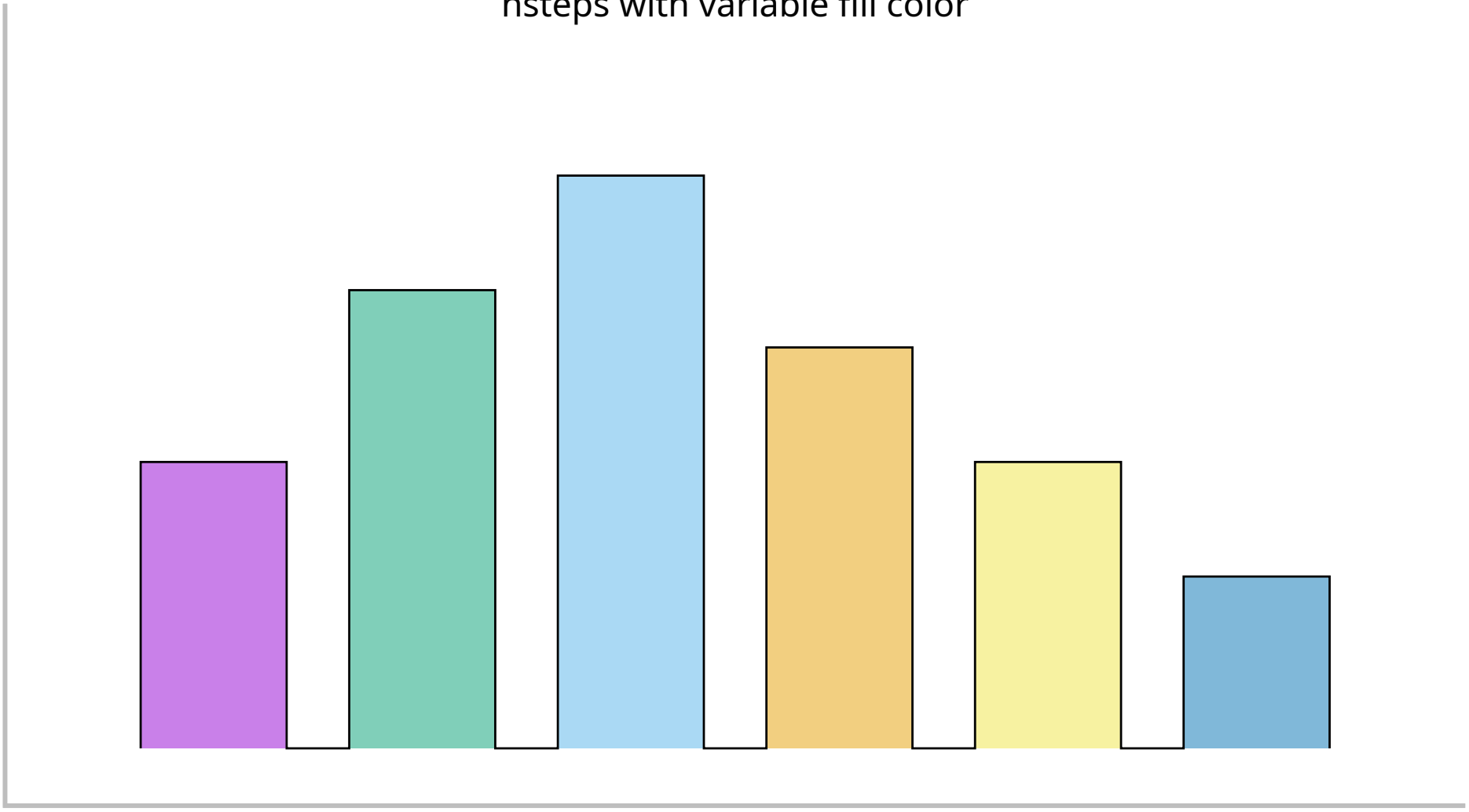

full width
link
nolink


full width
pillar
nolink
baseline

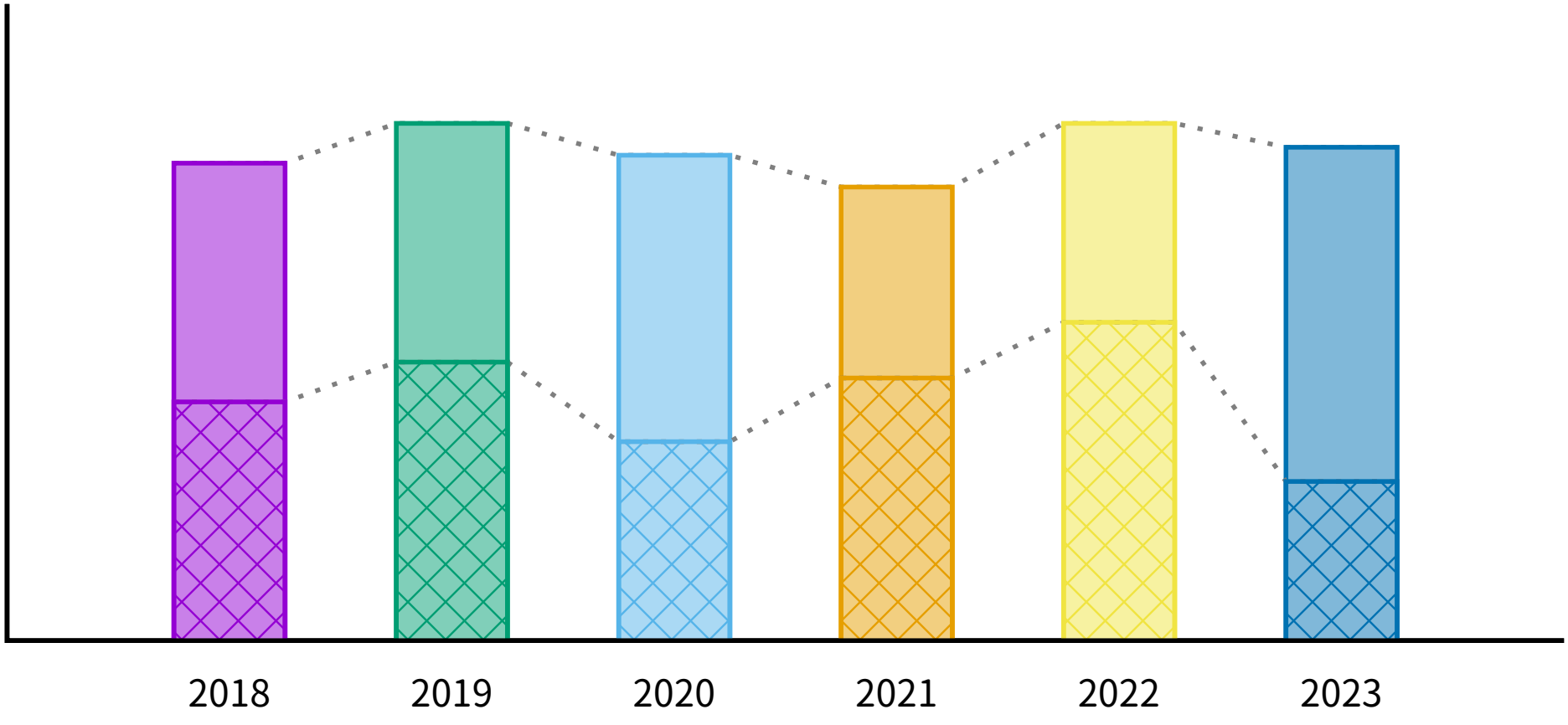

narrow width
link
nolink

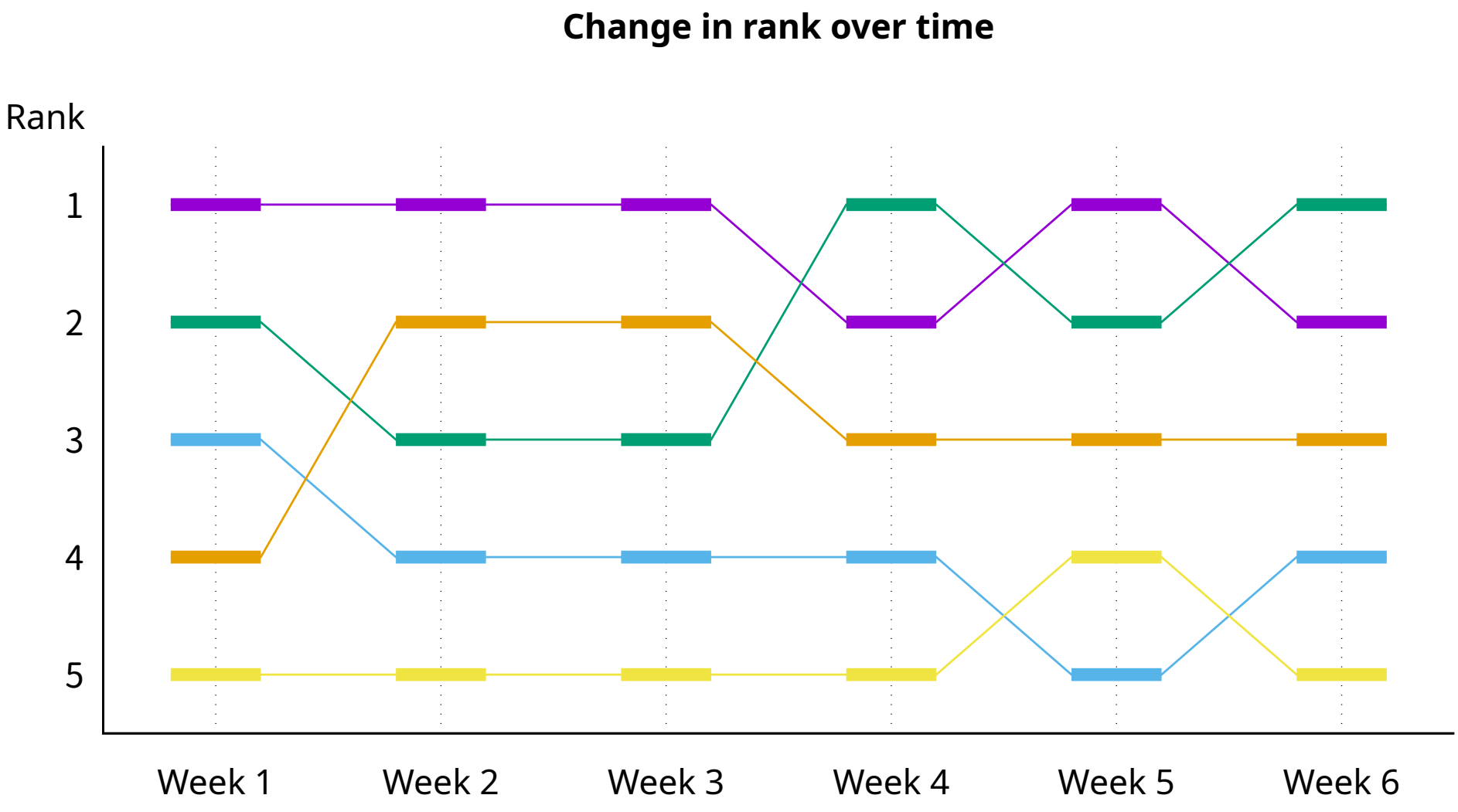

full width
forward
backward
width 0.8
forward
backward
width 0.4
forward
backward

hsteps with variable fill color

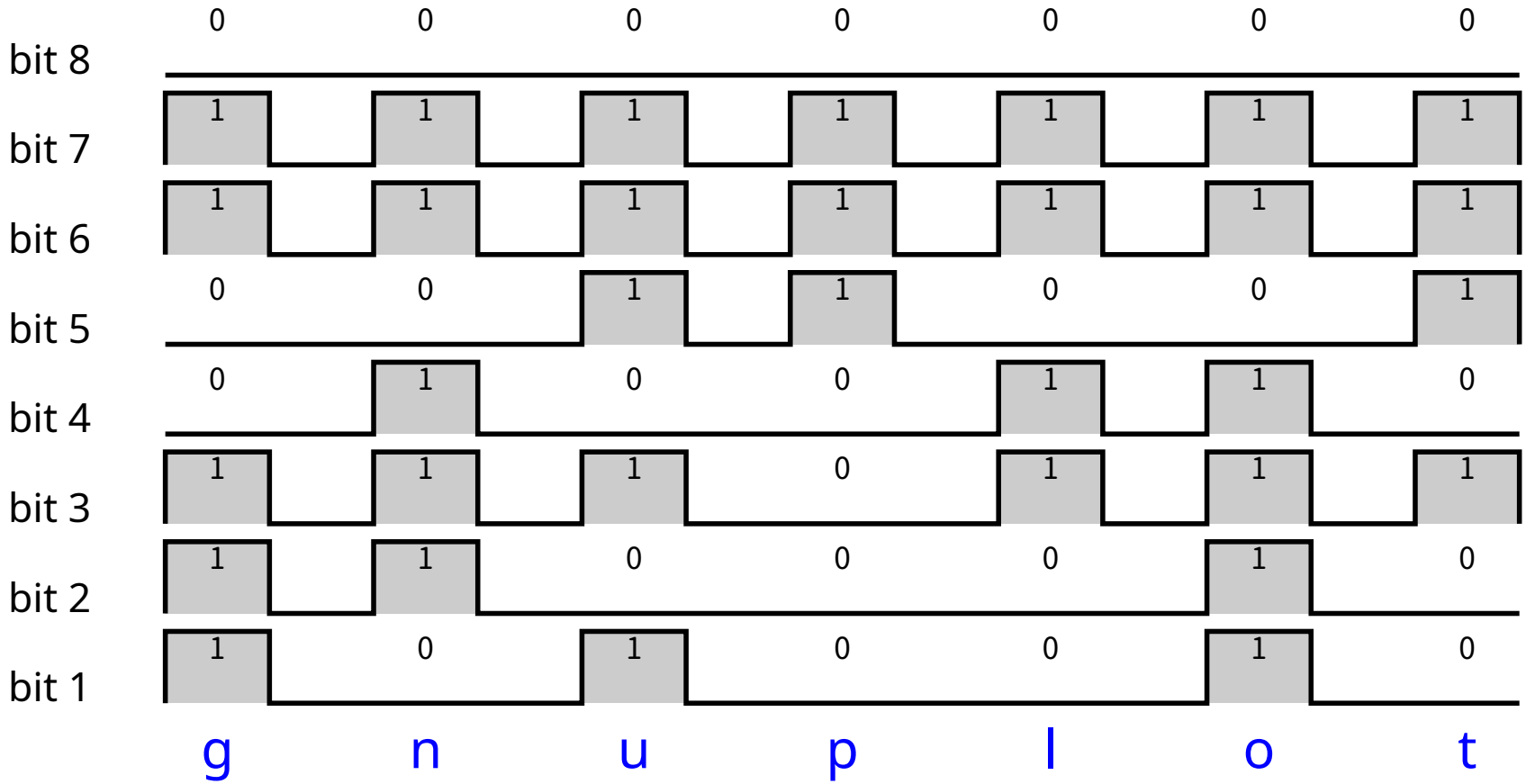


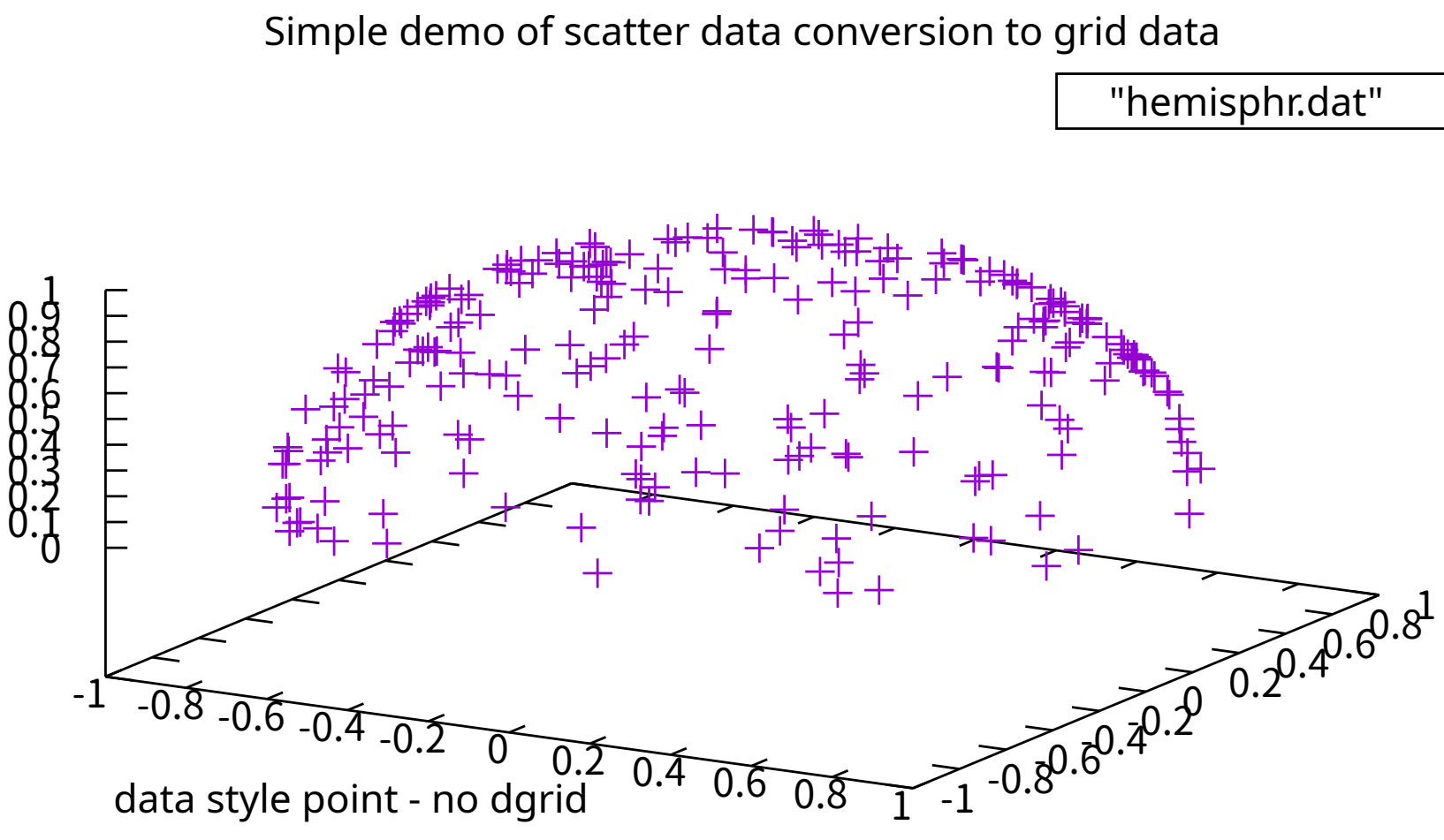
Stacked histogram constructed from plot style hsteps

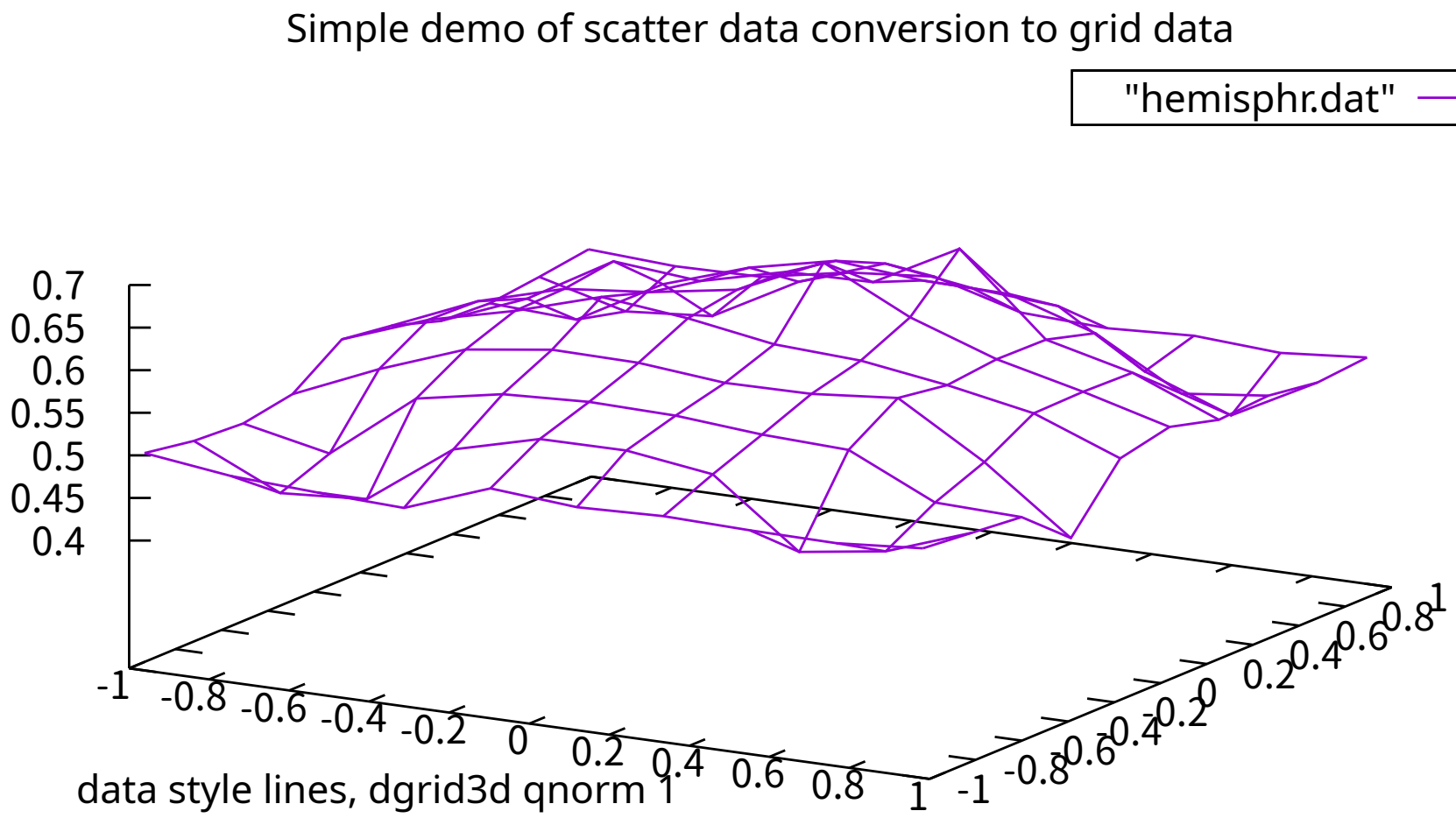


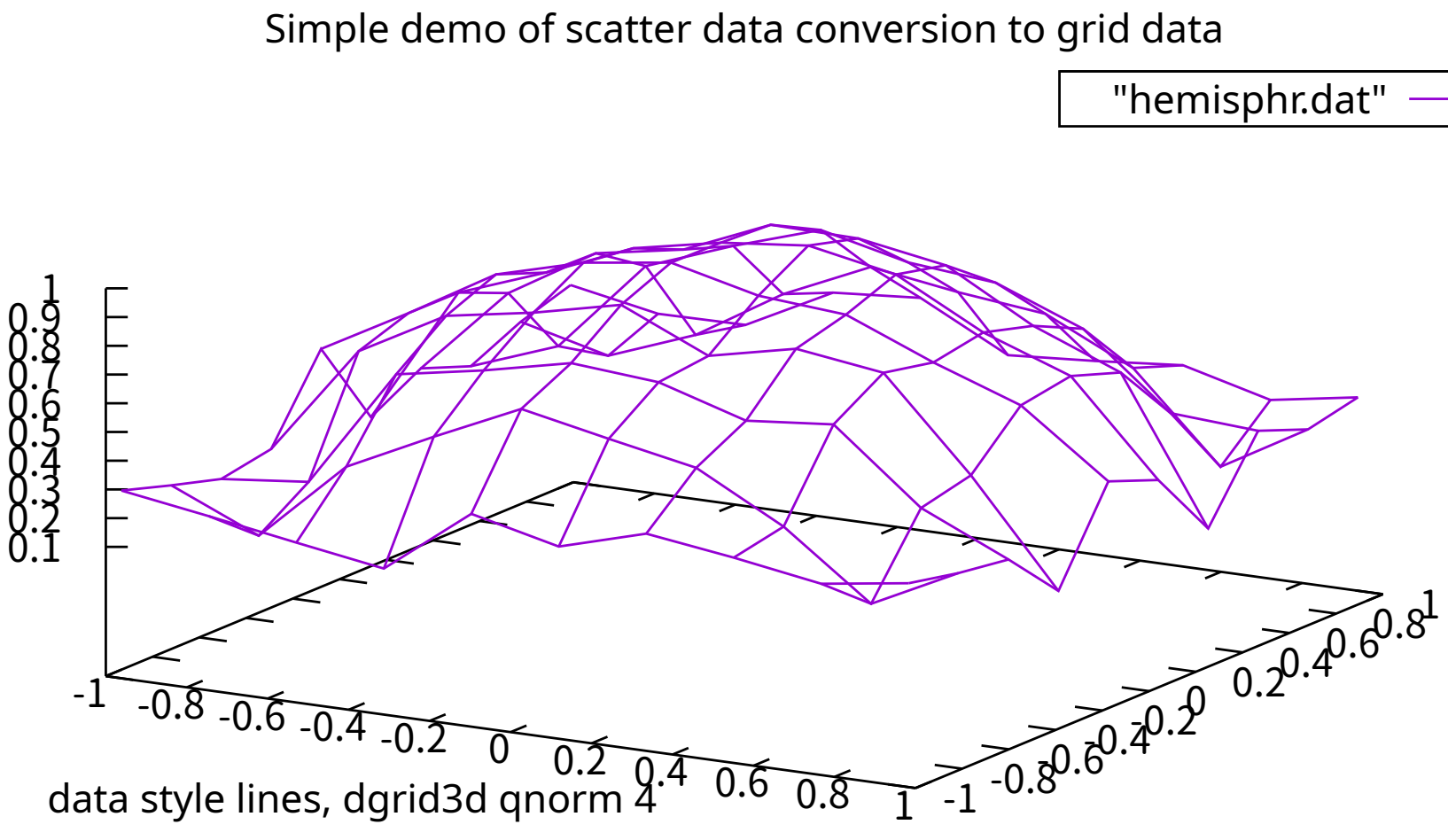

Change in rank over time
Rank
1
2
3
4
5
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6

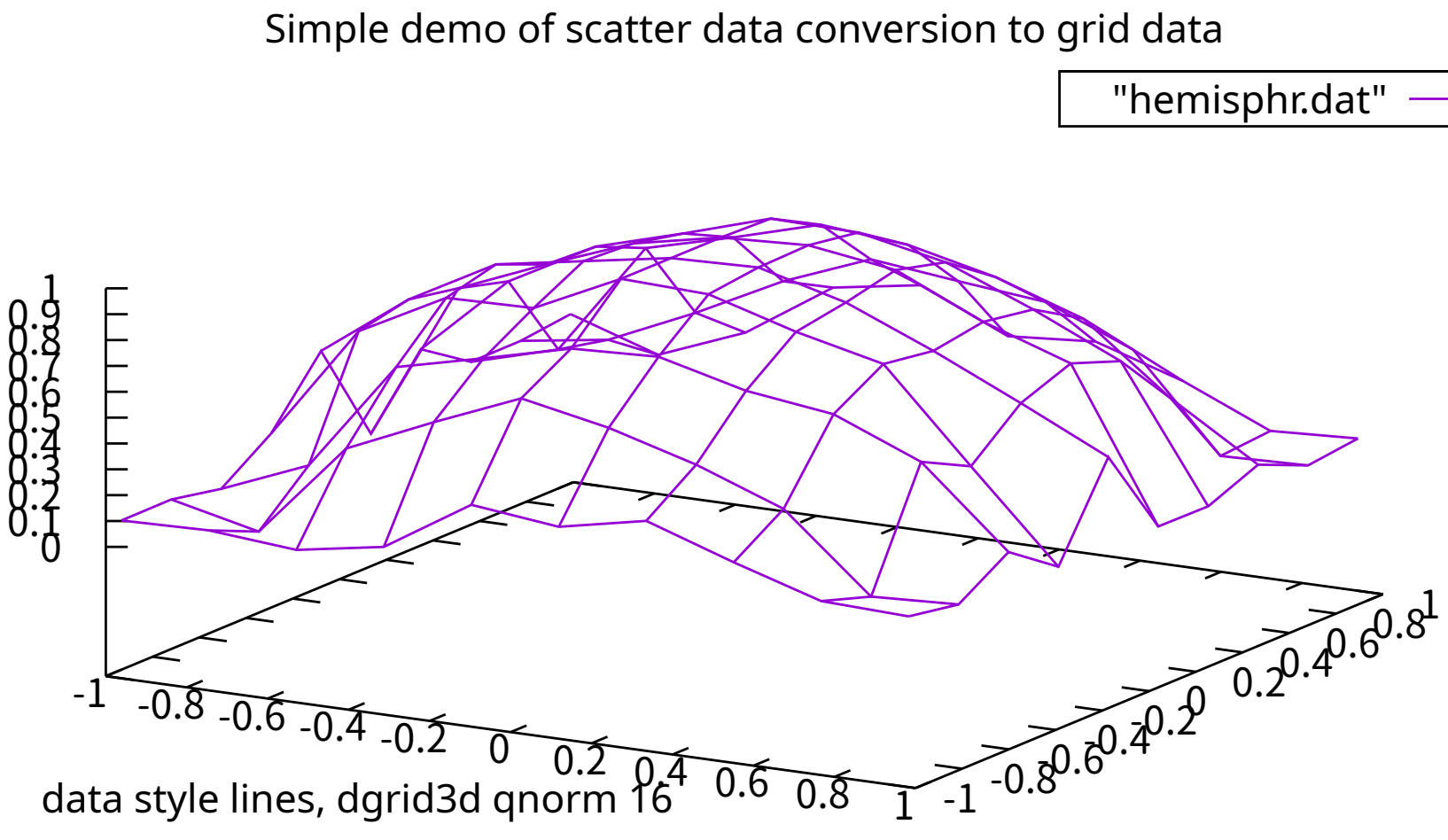
Logic timing of bits in sequential ASCII characters

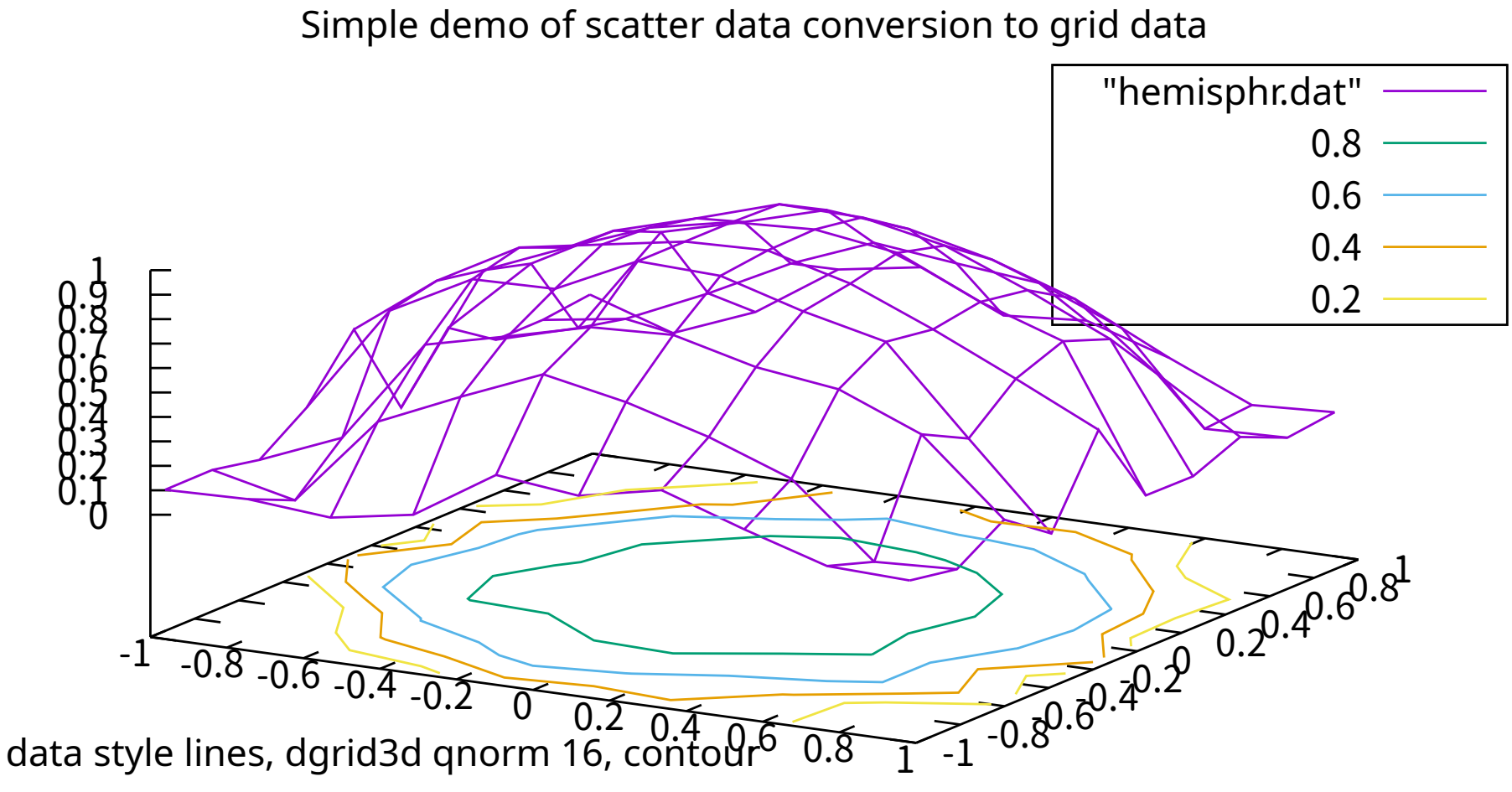


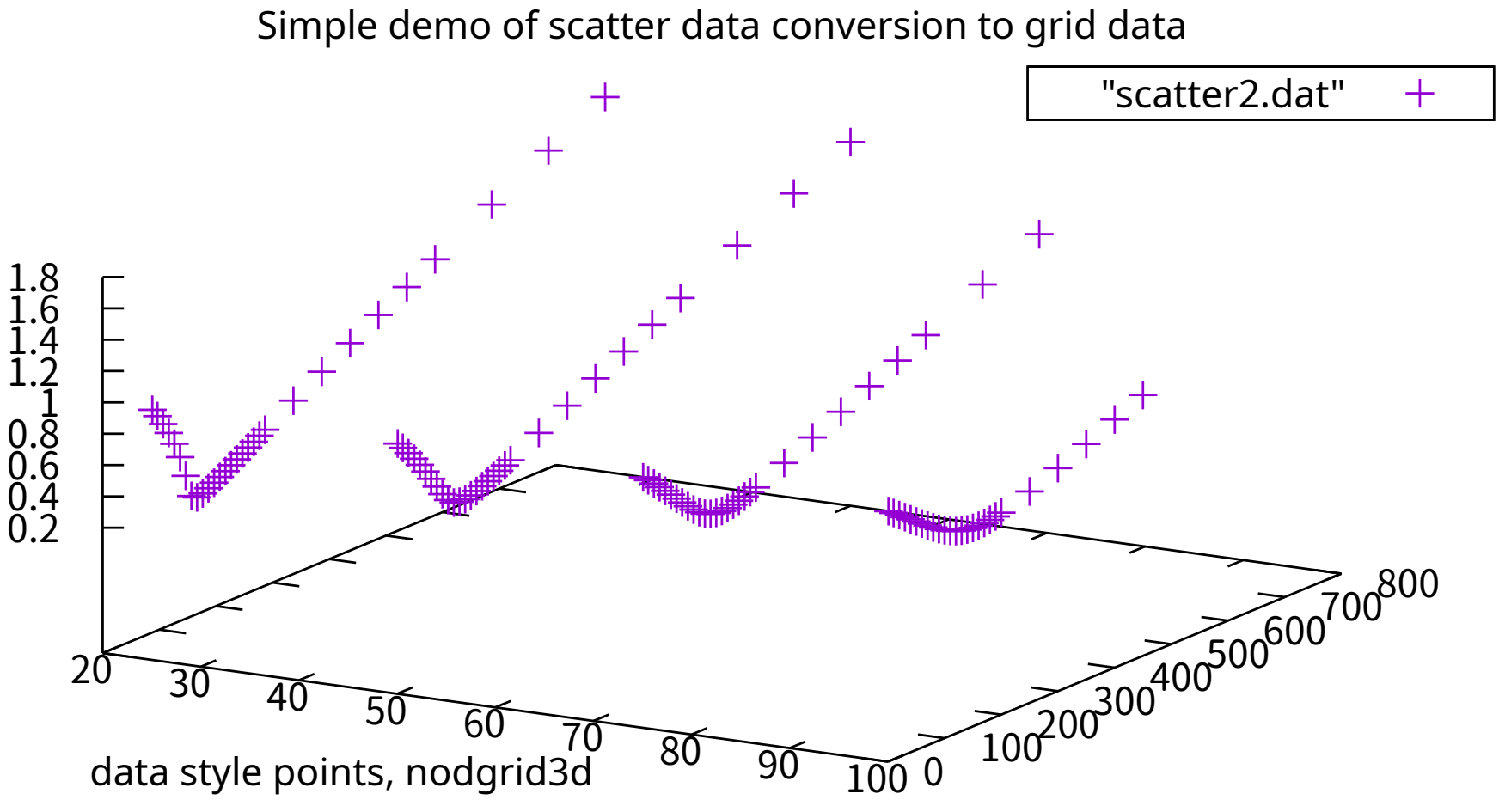

Simple demo of scatter data conversion to grid data
"hemisphr.dat"
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
data style point - no dgrid

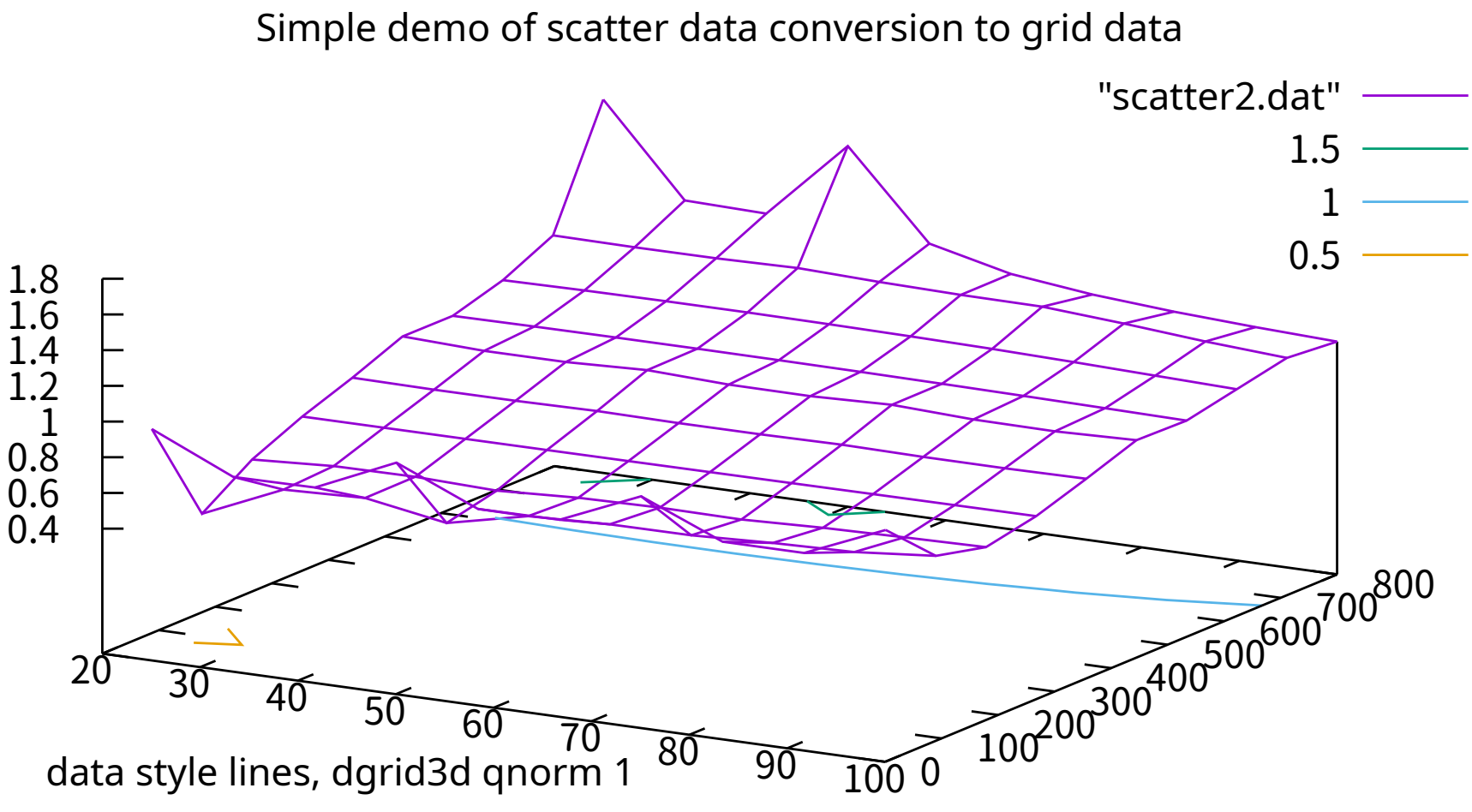

Simple demo of scatter data conversion to grid data
"hemisphr.dat"
0.7
0.65
0.6
0.55
0.5
0.45
0.4
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
data style lines, dgrid3d qnorm 1

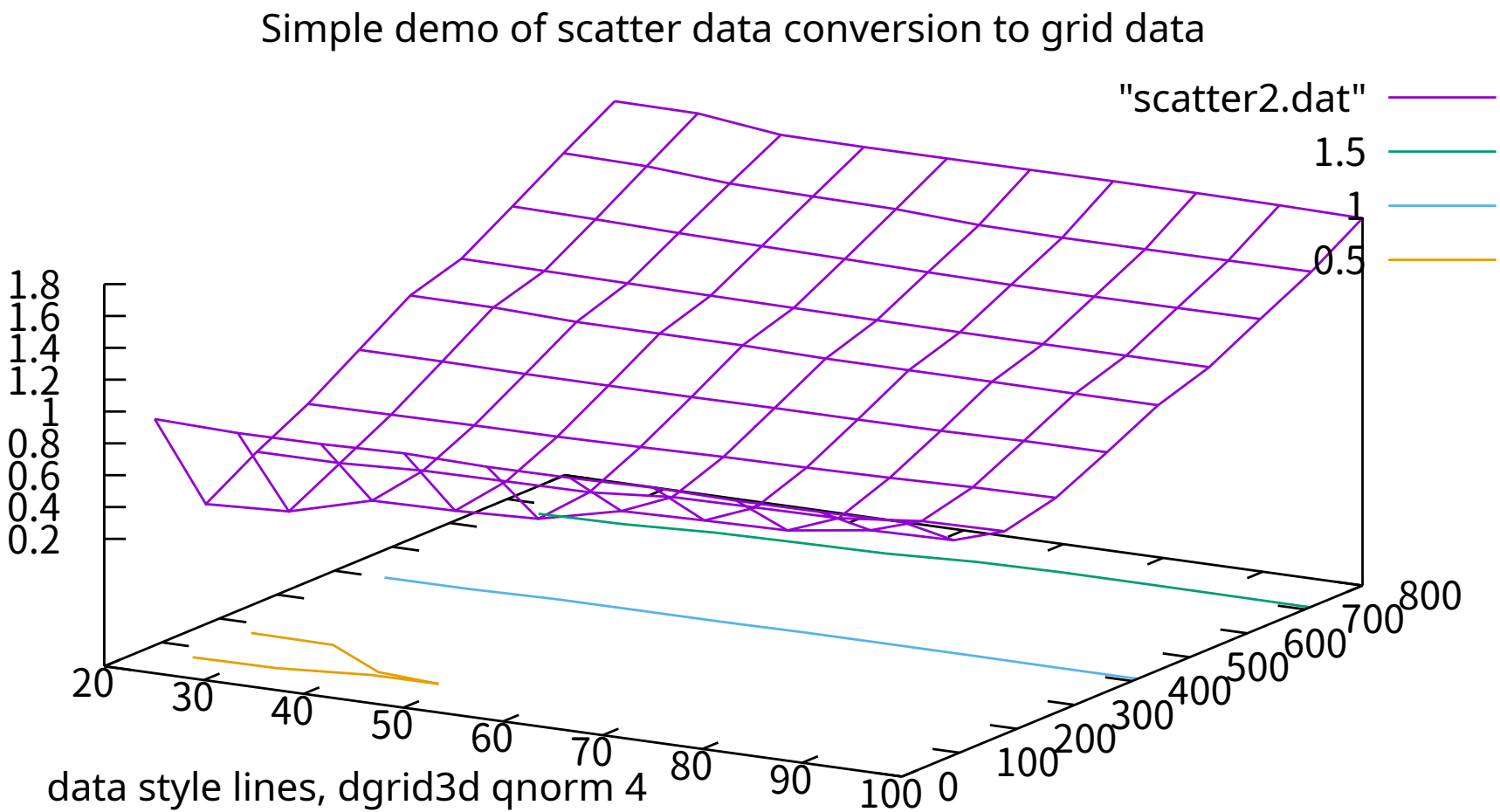

Simple demo of scatter data conversion to grid data
"hemisphr.dat"
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
data style lines, dgrid3d qnorm 4

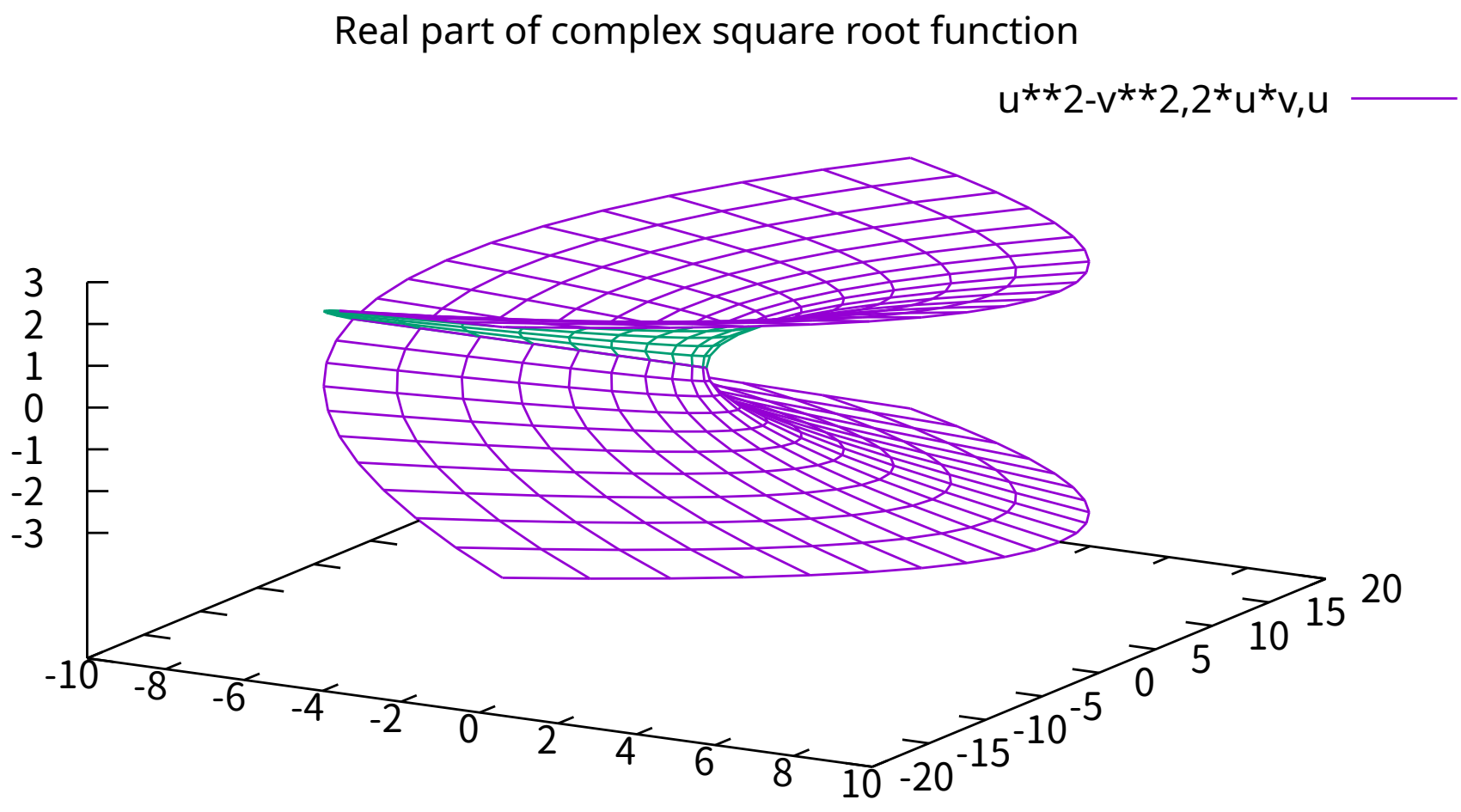

Simple demo of scatter data conversion to grid data
"hemisphr.dat"
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
data style lines, dgrid3d qnorm 16


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
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
data style lines, dgrid3d qnorm 16, contour

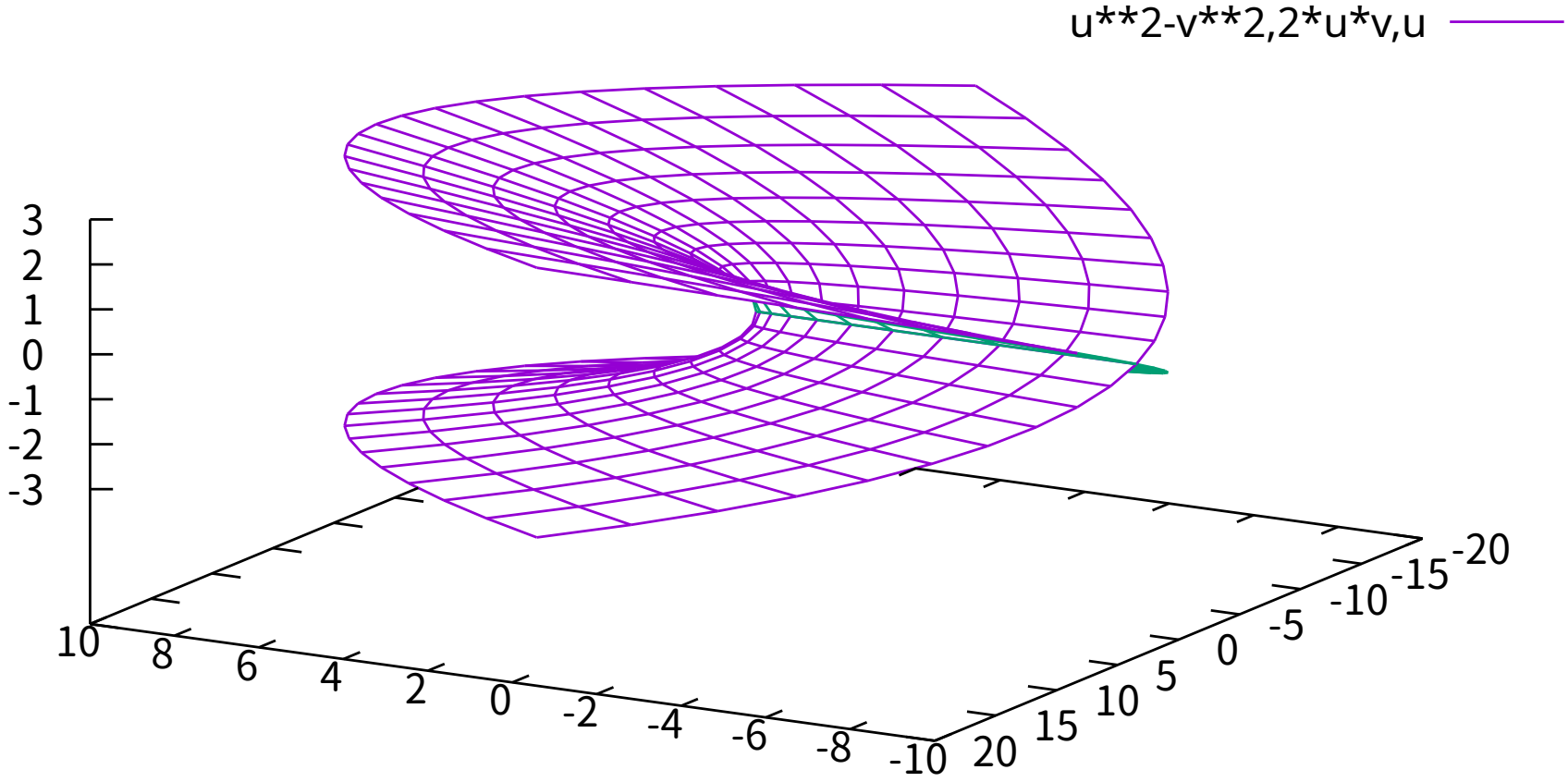

Simple demo of scatter data conversion to grid data
"scatter2.dat"
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500
600
700
800
data style points, nodgrid3d

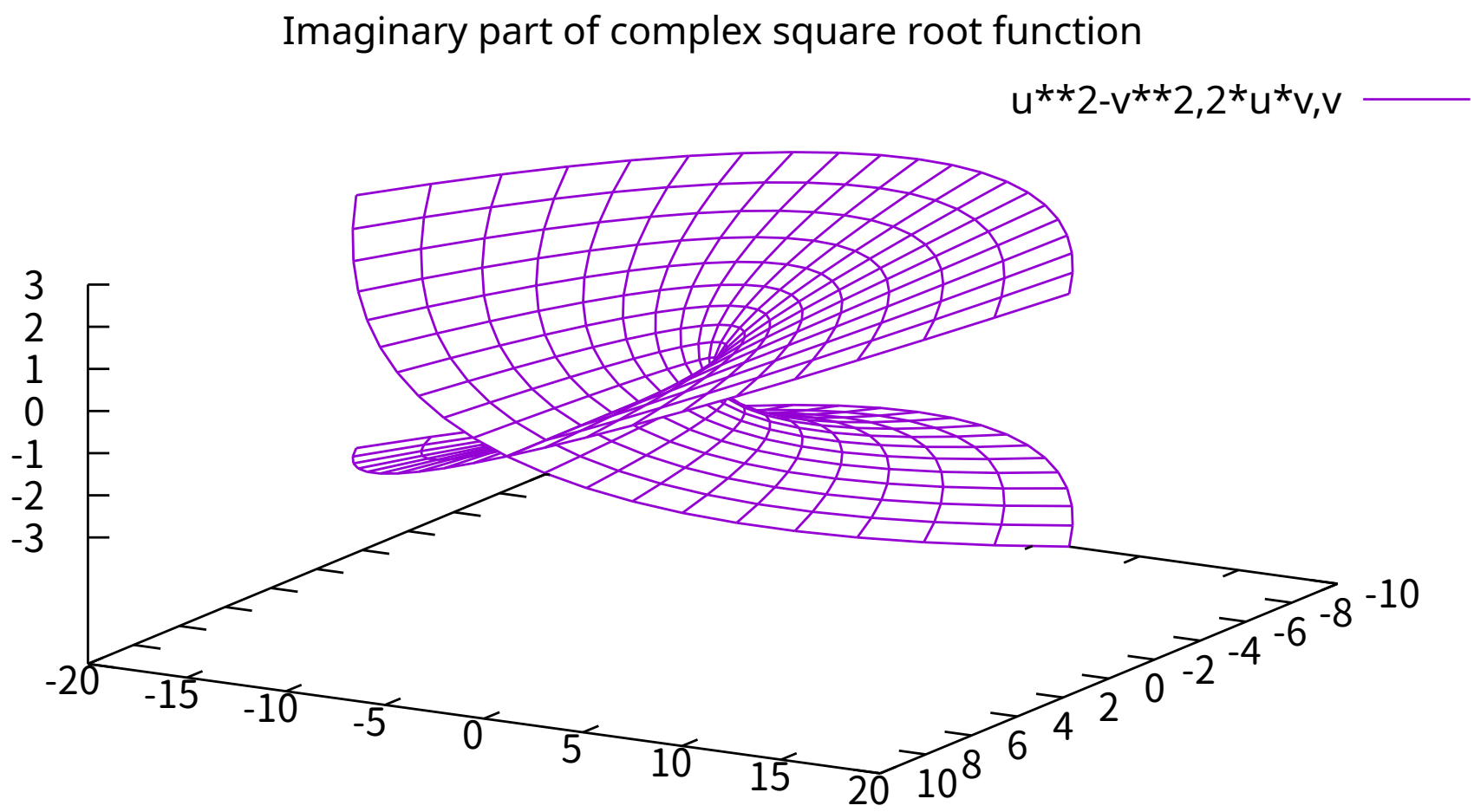

Simple demo of scatter data conversion to grid data
"scatter2.dat"
1.5
1
0.5
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500
600
700
800
data style lines, dgrid3d qnorm 1


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 qnorm 4

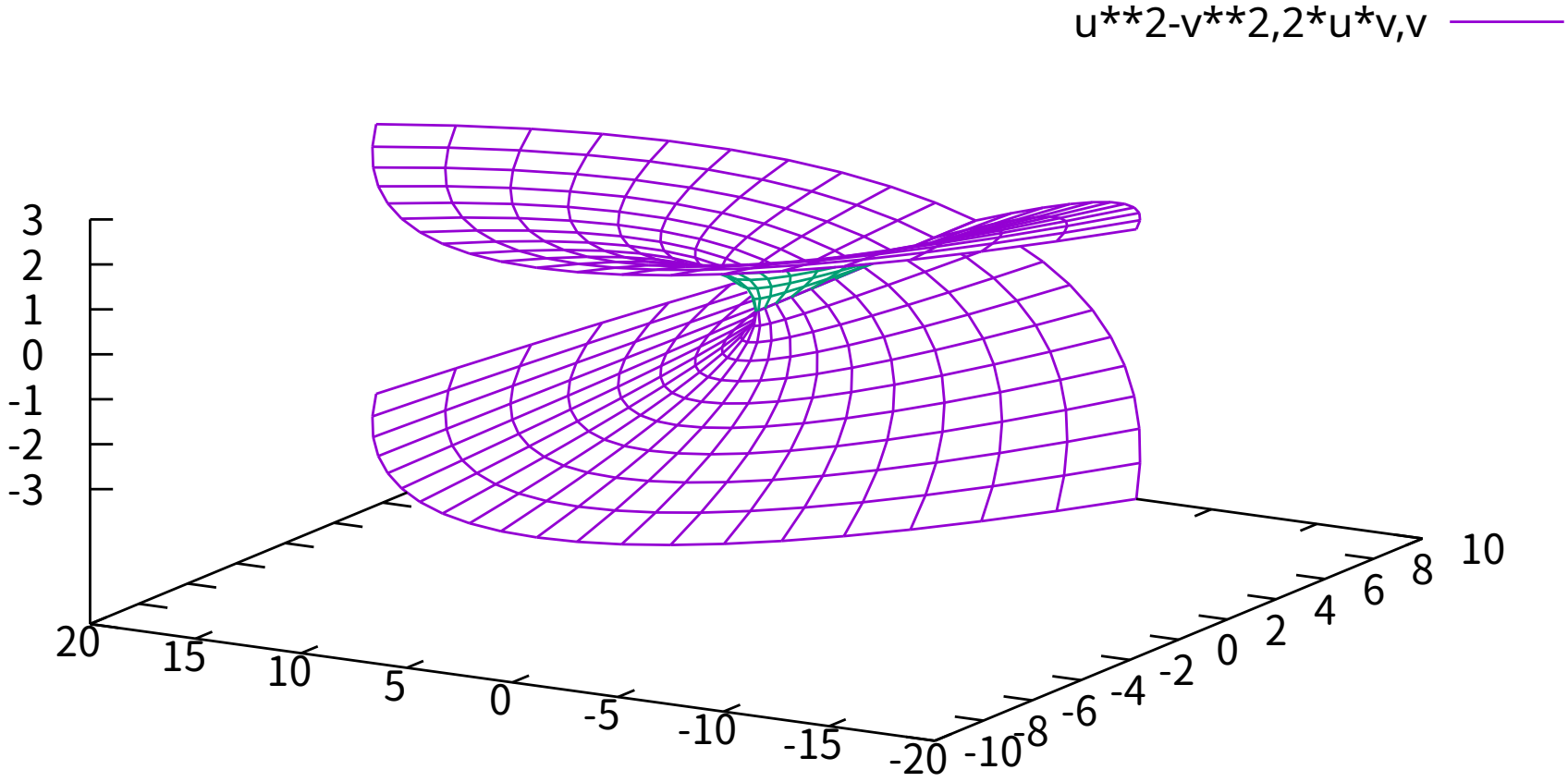

Real part of complex square root function
u**2-v**2,2*u*v,u
3
2
1
0
-1
-2
-3
-10
-8
-6
-4
-2
0
2
4
6
8
10
-20
-15
-10
-5
0
5
10
15
20

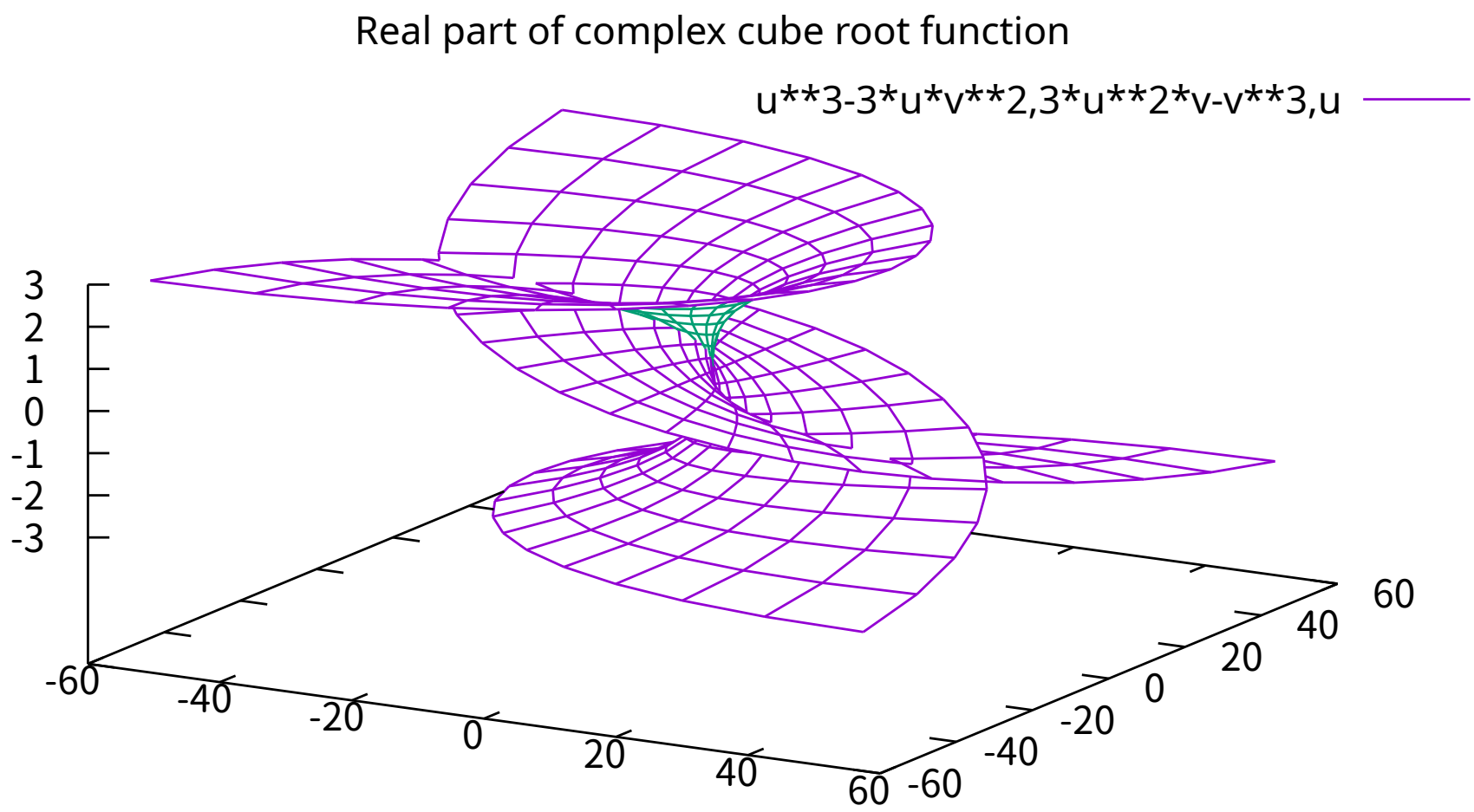
Real part of complex square root function (different view)



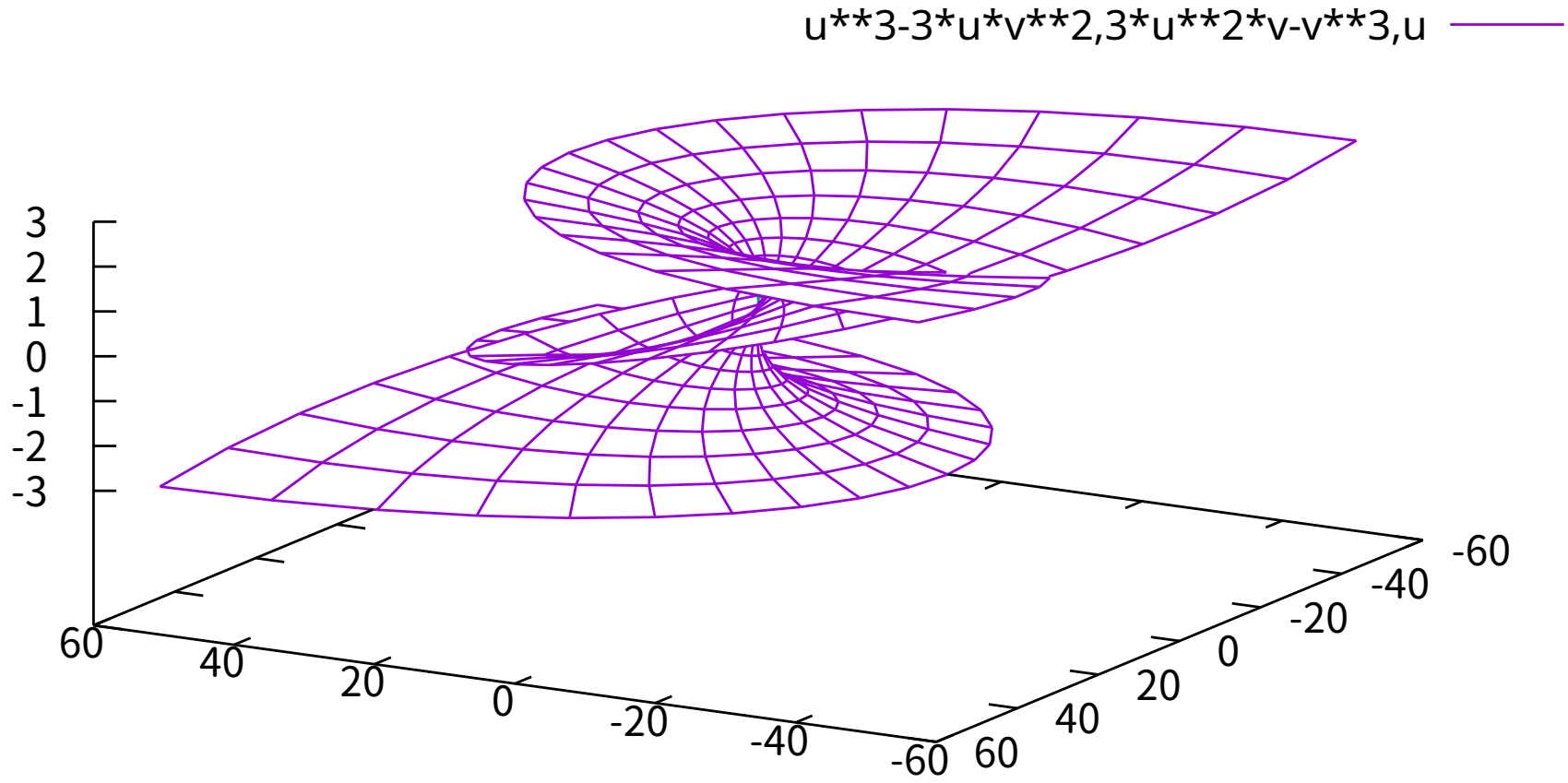

Imaginary part of complex square root function
u**2-v**2,2*u*v,v
3
2
1
0
-1
-2
-3
-20
-15
-10
-5
0
5
10
15
20
10
8
6
4
2
0
-2
-4
-6
-8
-10

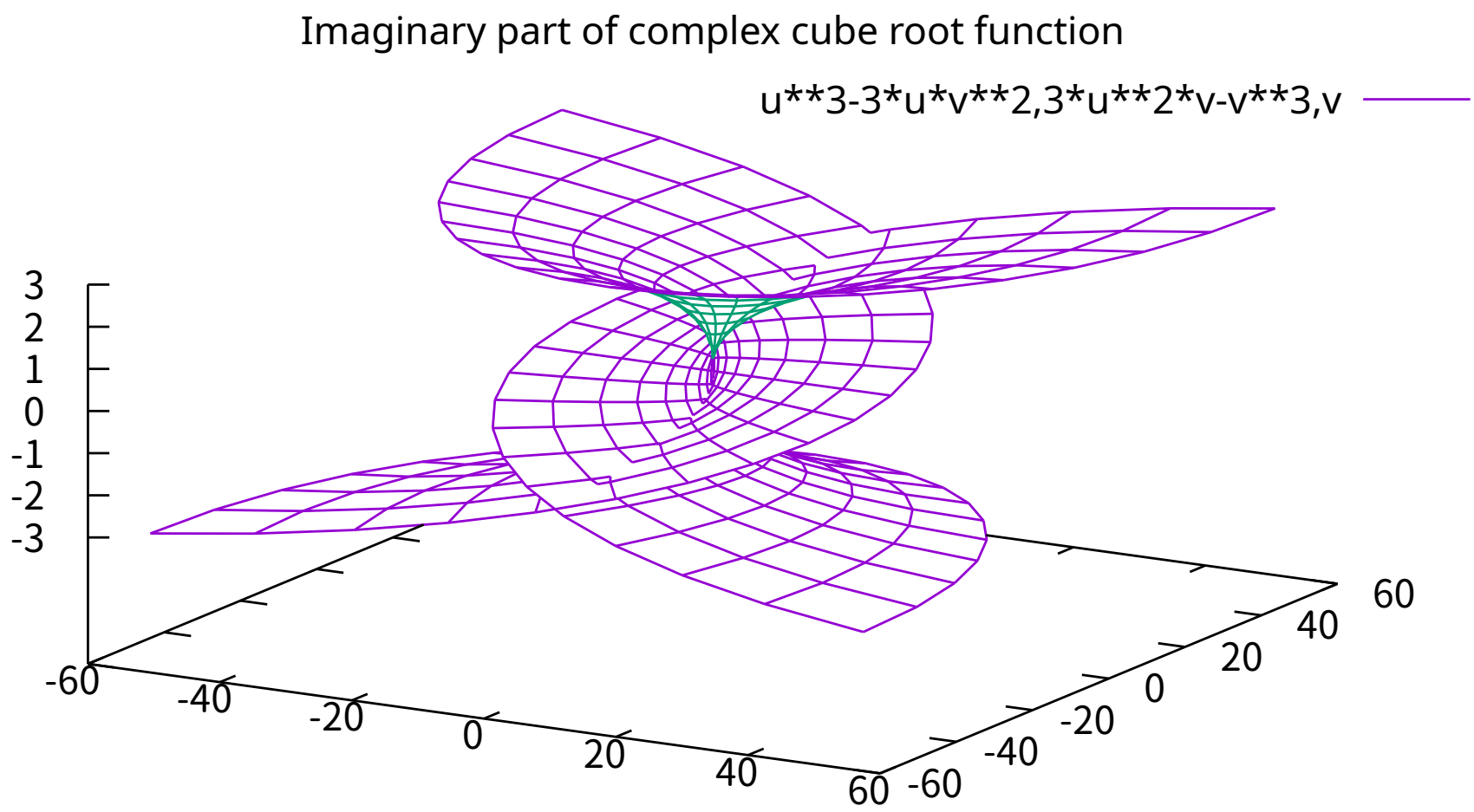
Imaginary part of complex square root function (different view)



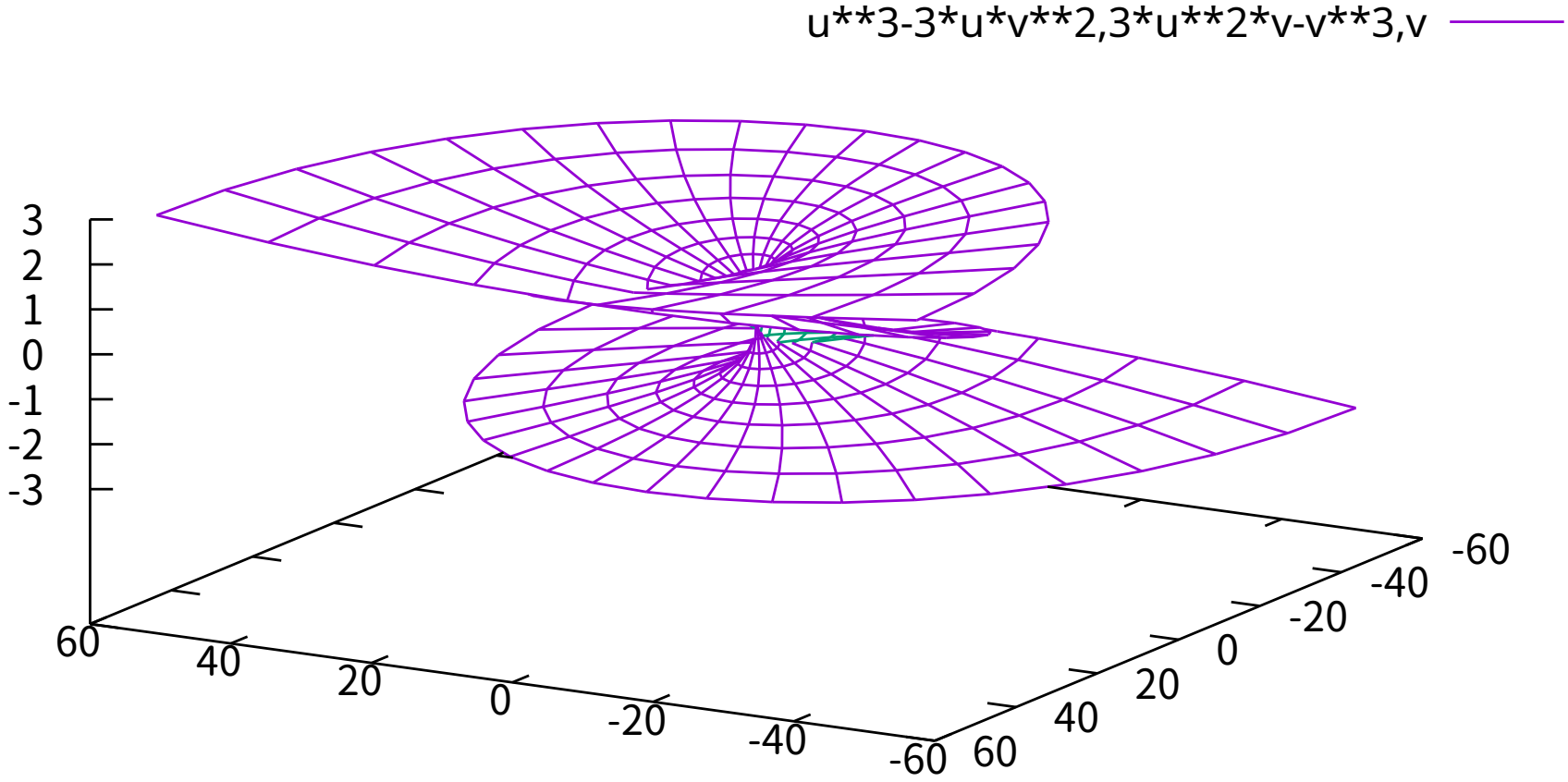

Real part of complex cube root function
u**3-3*u*v**2,3*u**2*v-v**3,u
3
2
1
0
-1
-2
-3
-60
-40
-20
0
20
40
60
-60
-40
-20
0
20
40
60

Real part of complex cube root function (different view)

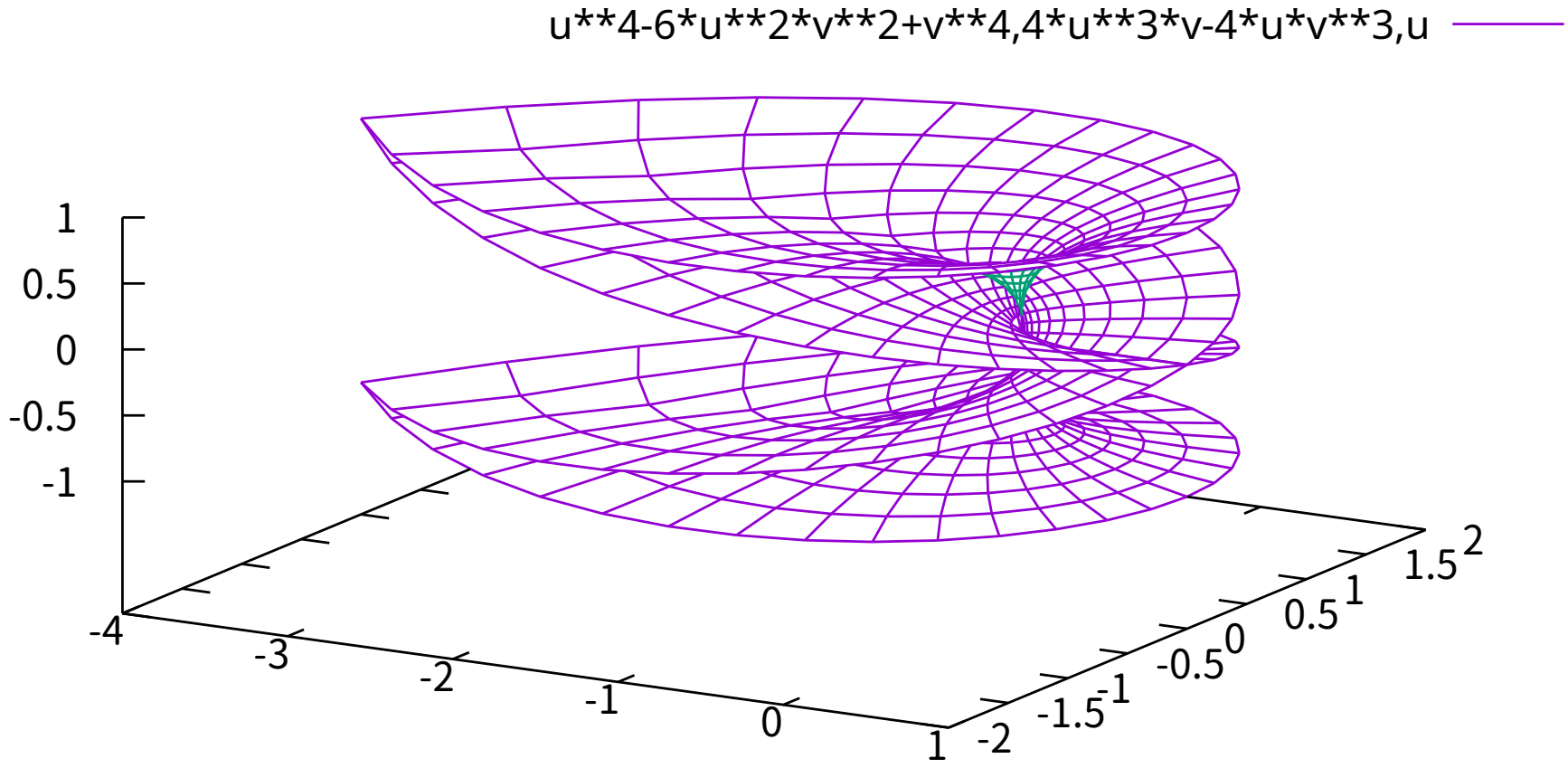



Imaginary part of complex cube root function
u**3-3*u*v**2,3*u**2*v-v**3,v
3
2
1
0
-1
-2
-3
-60
-40
-20
0
20
40
60
-60
-40
-20
0
20
40
60

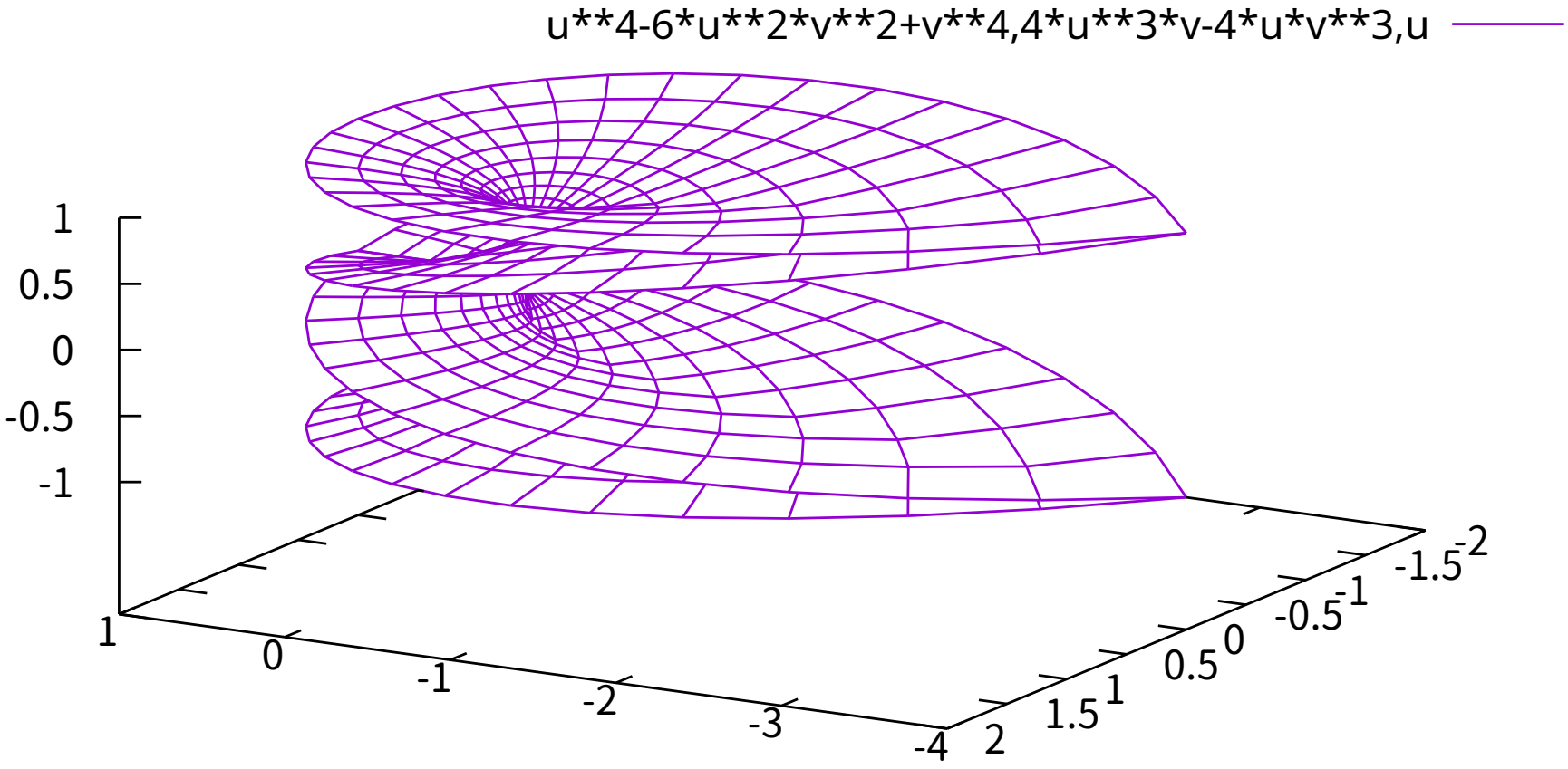
Imaginary part of complex cube root function (different view)

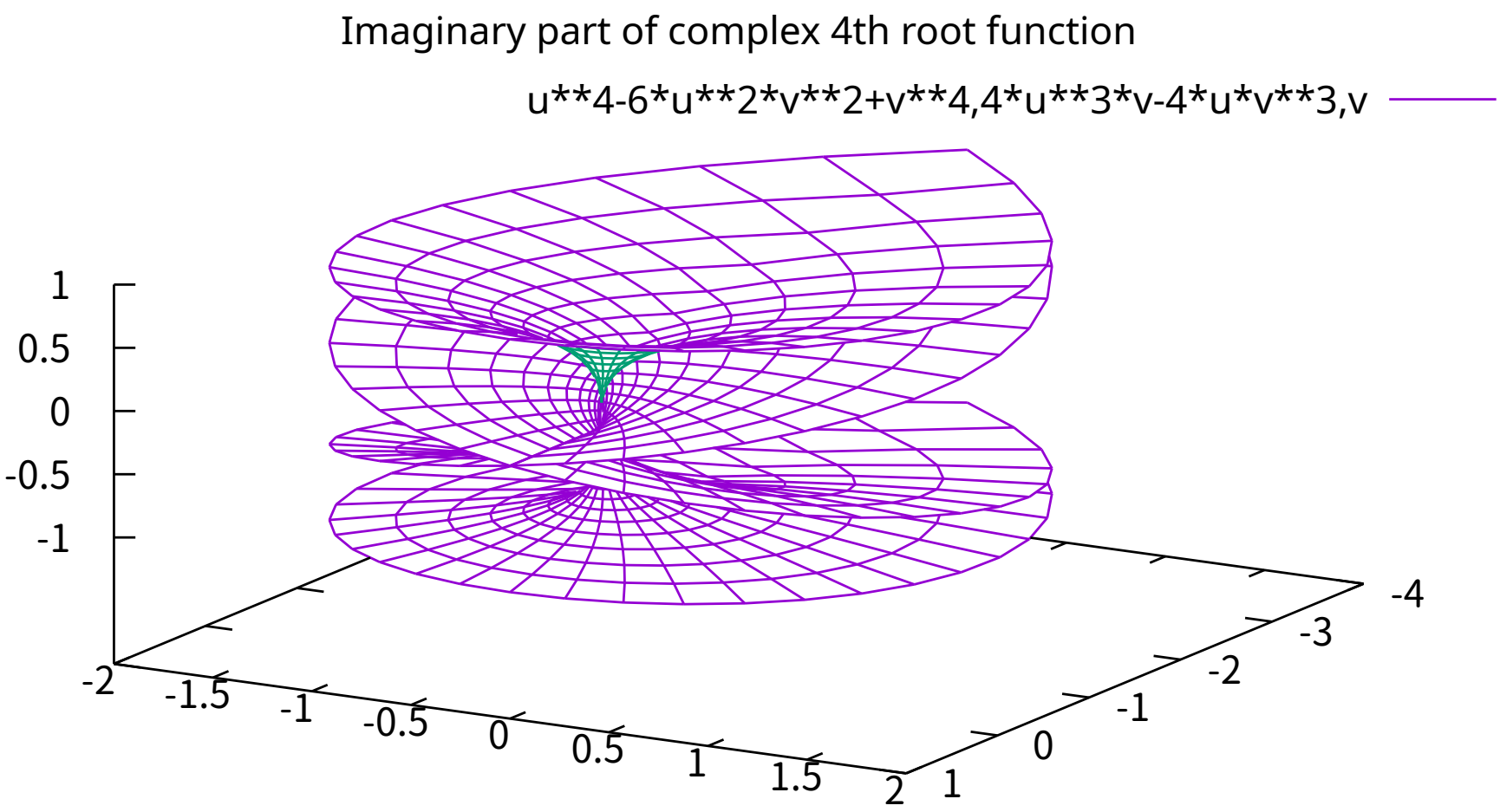


Real part of complex 4th root function

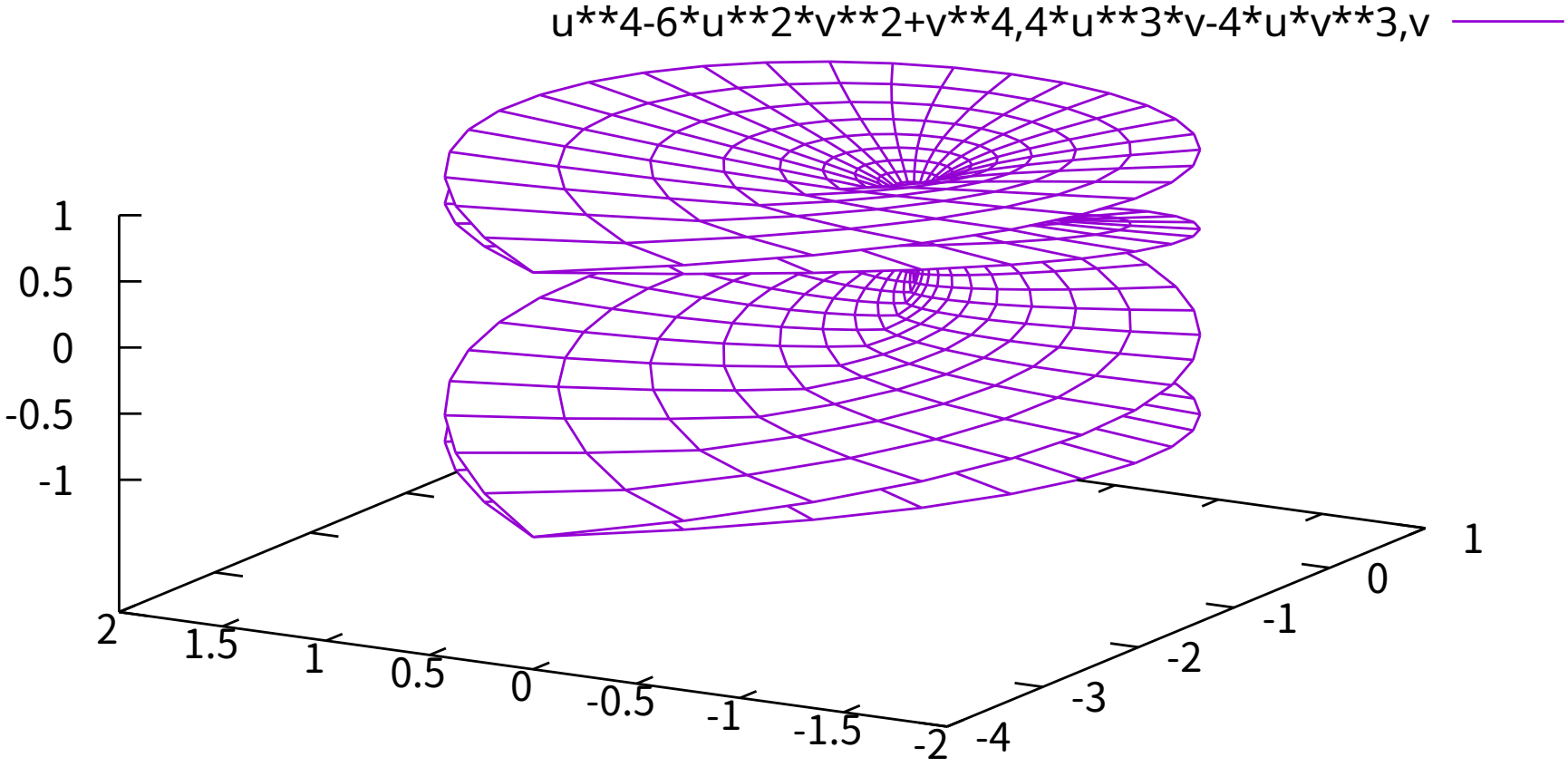


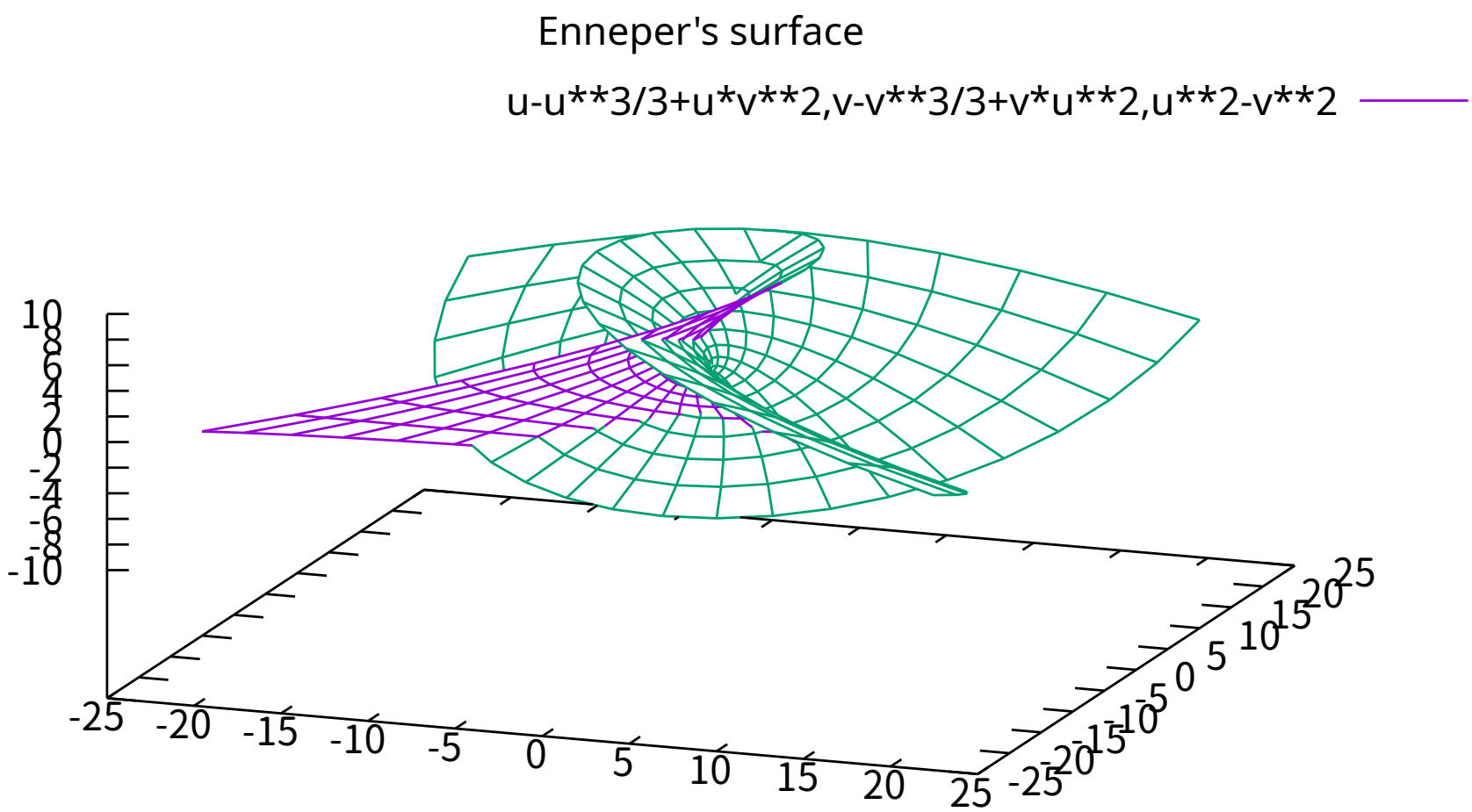
Real part of complex 4th root function (different view)



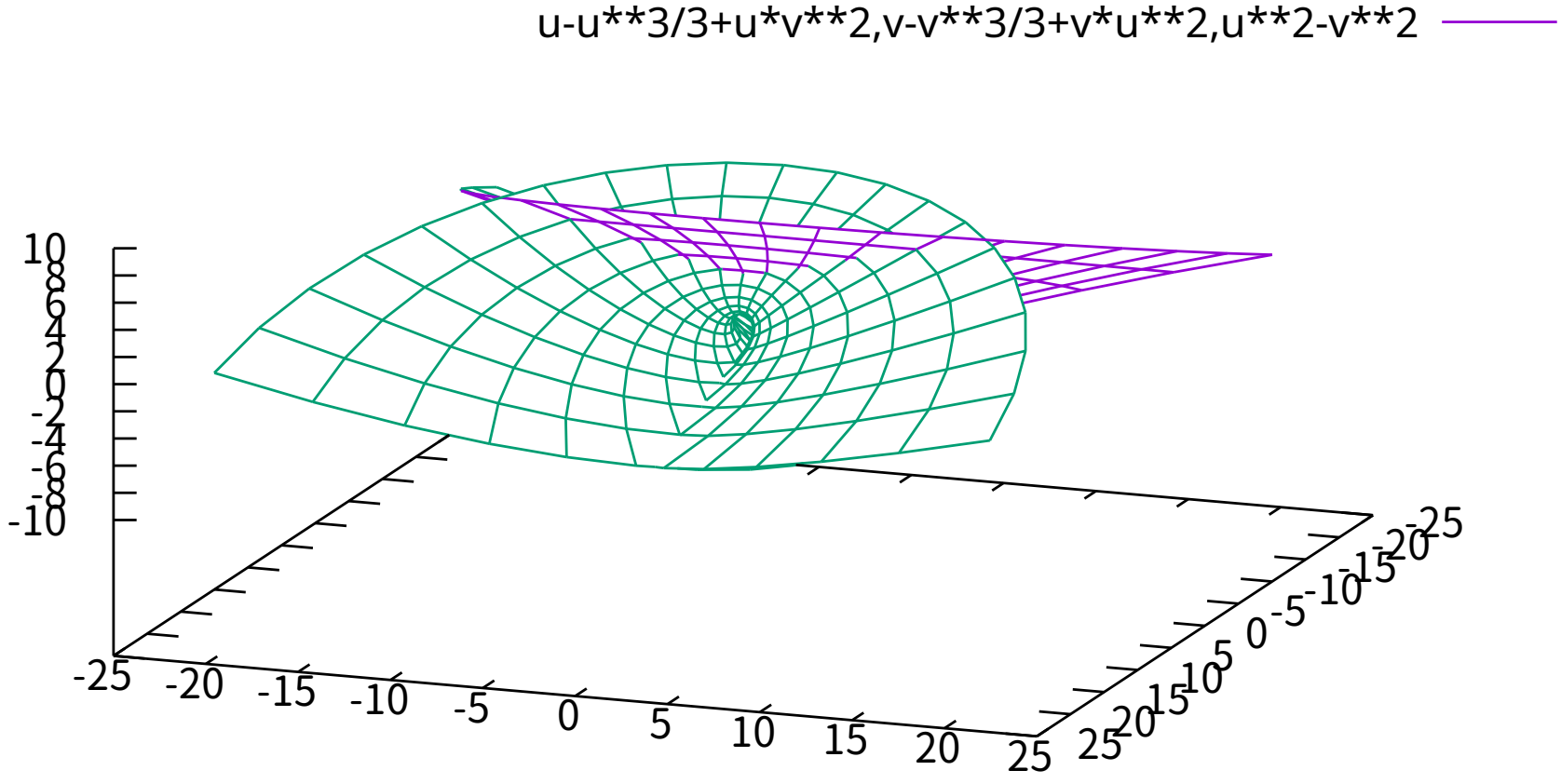

Imaginary part of complex 4th root function
u**4-6*u**2*v**2+v**4,4*u**3*v-4*u*v**3,v
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
1
0
-1
-2
-3
-4

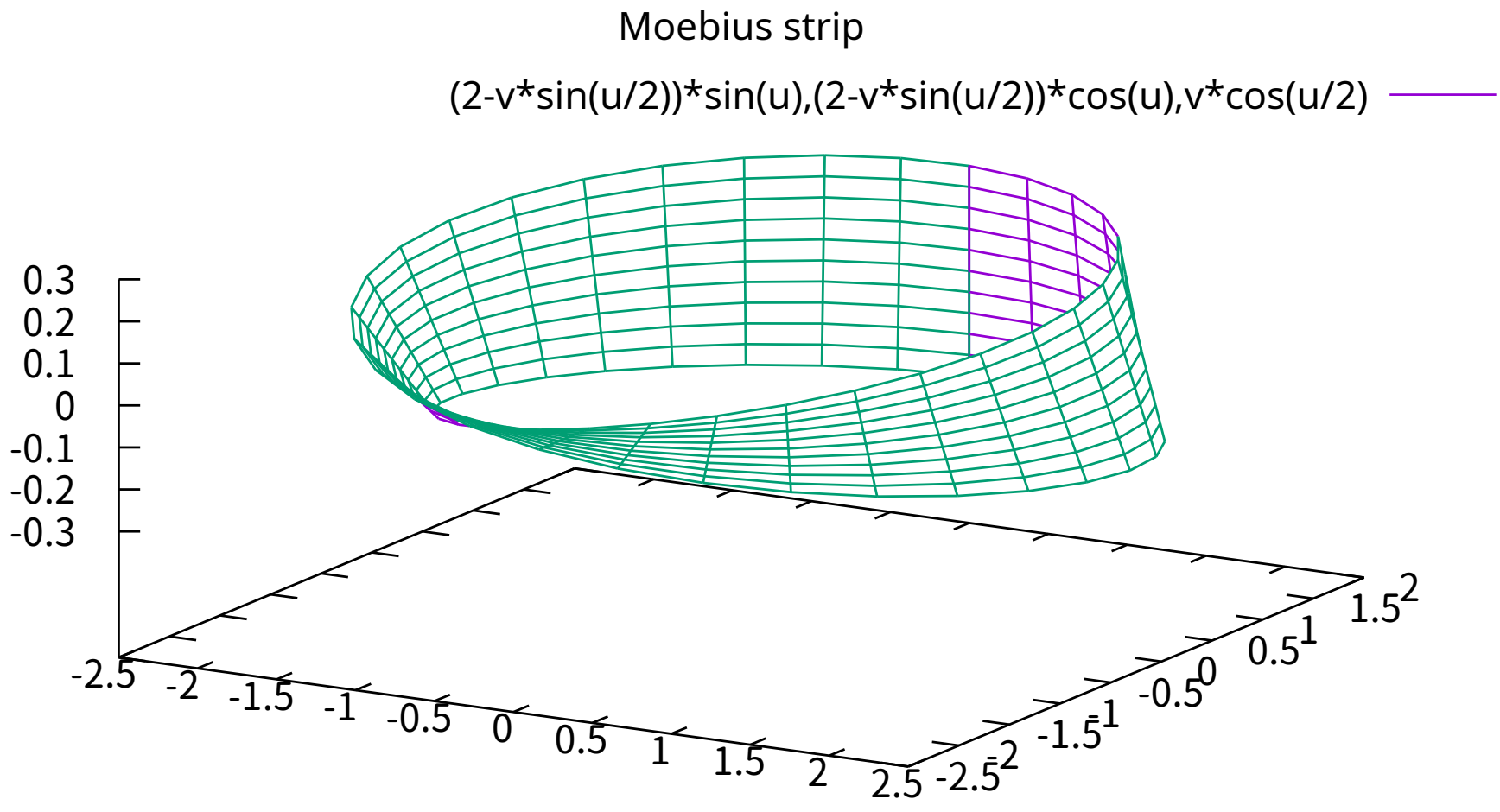
Imaginary part of complex 4th root function (different view)



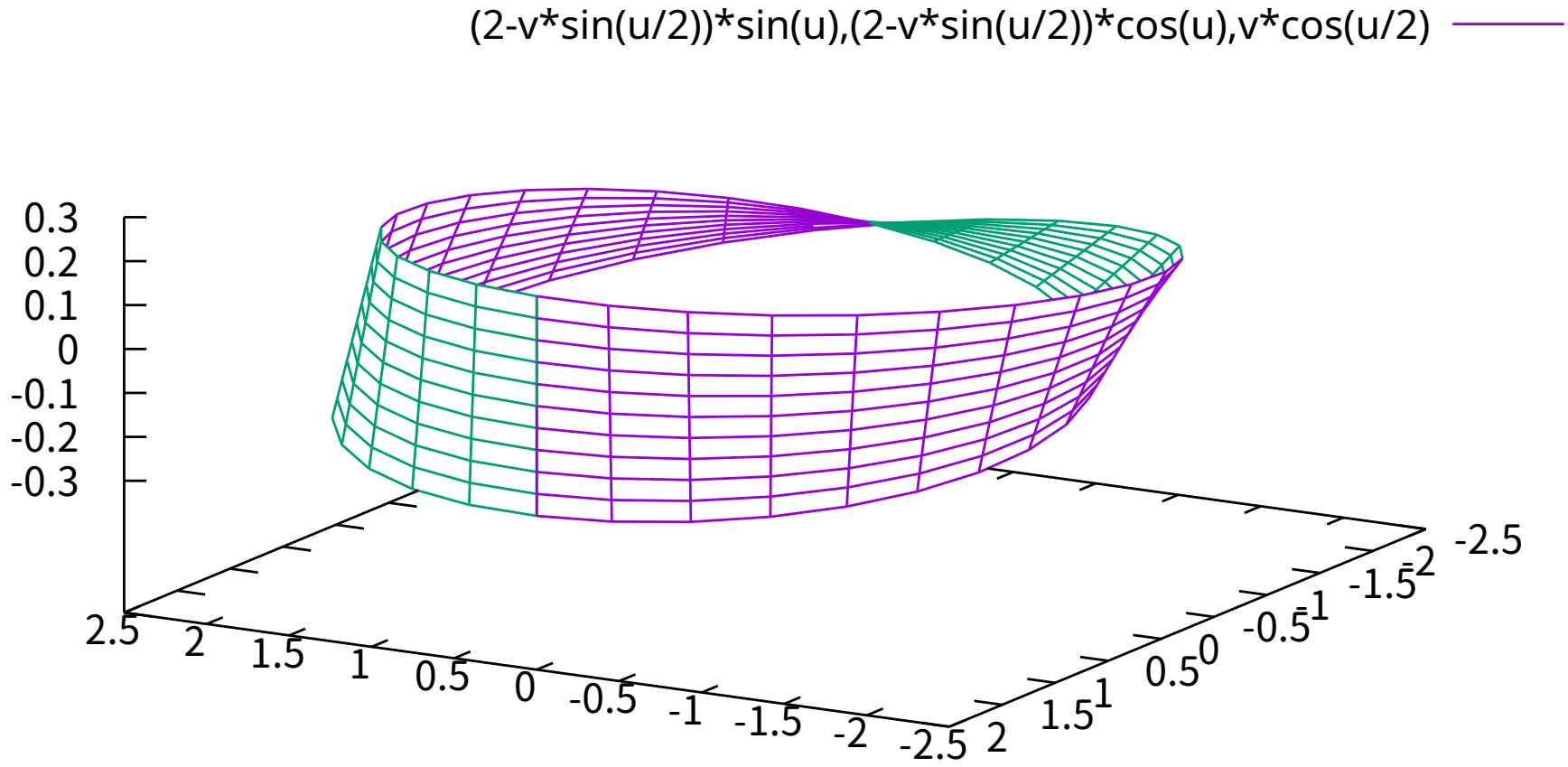

Enneper's surface
u-u**3/3+u*v**2,v-v**3/3+v*u**2,u**2-v**2
10
8
6
4
2
0
-2
-4
-6
-8
-10
-25
-20
-15
-10
-5
0
5
10
15
20
25
-25
-20
-15
-10
-5
0
5
10
15
20
25

Enneper's surface (different view)

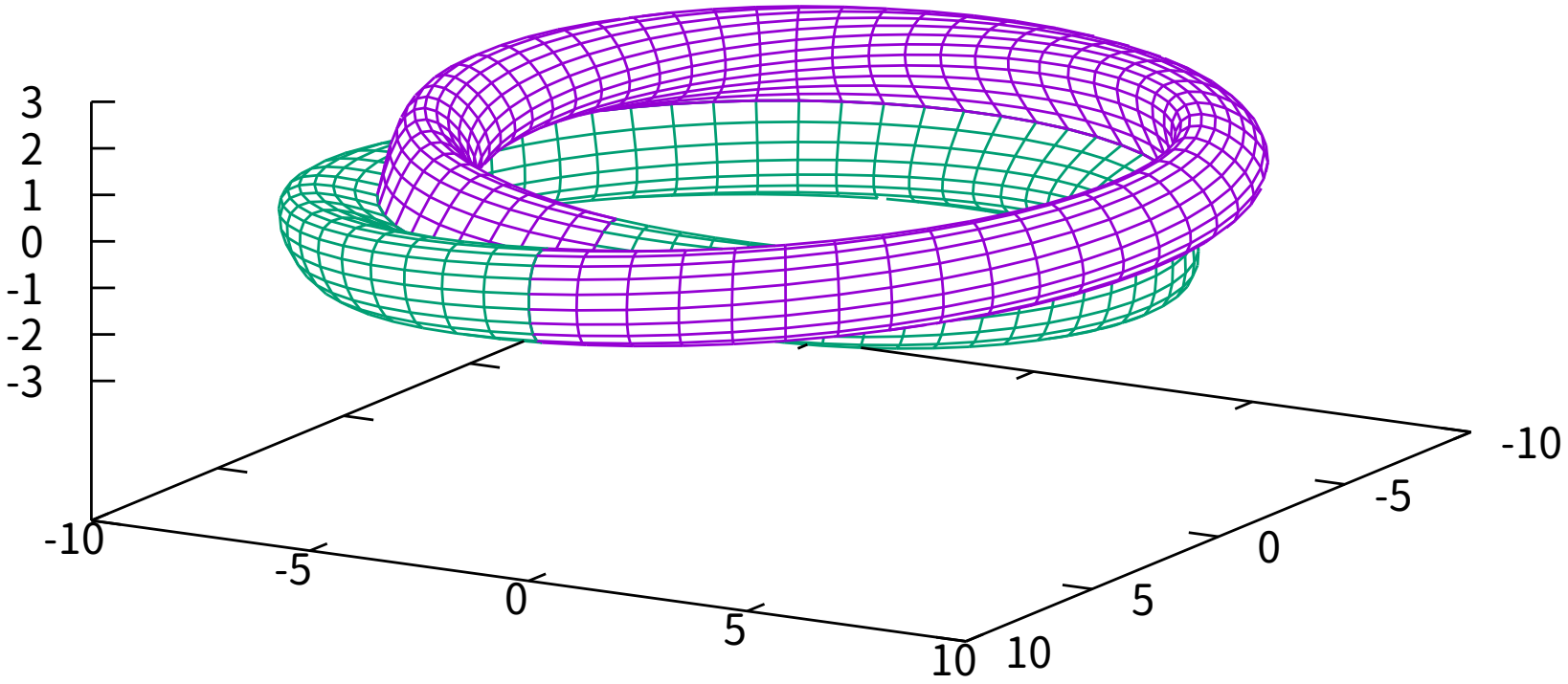



Moebius strip
(2-v*sin(u/2))*sin(u),(2-v*sin(u/2))*cos(u),v*cos(u/2)
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

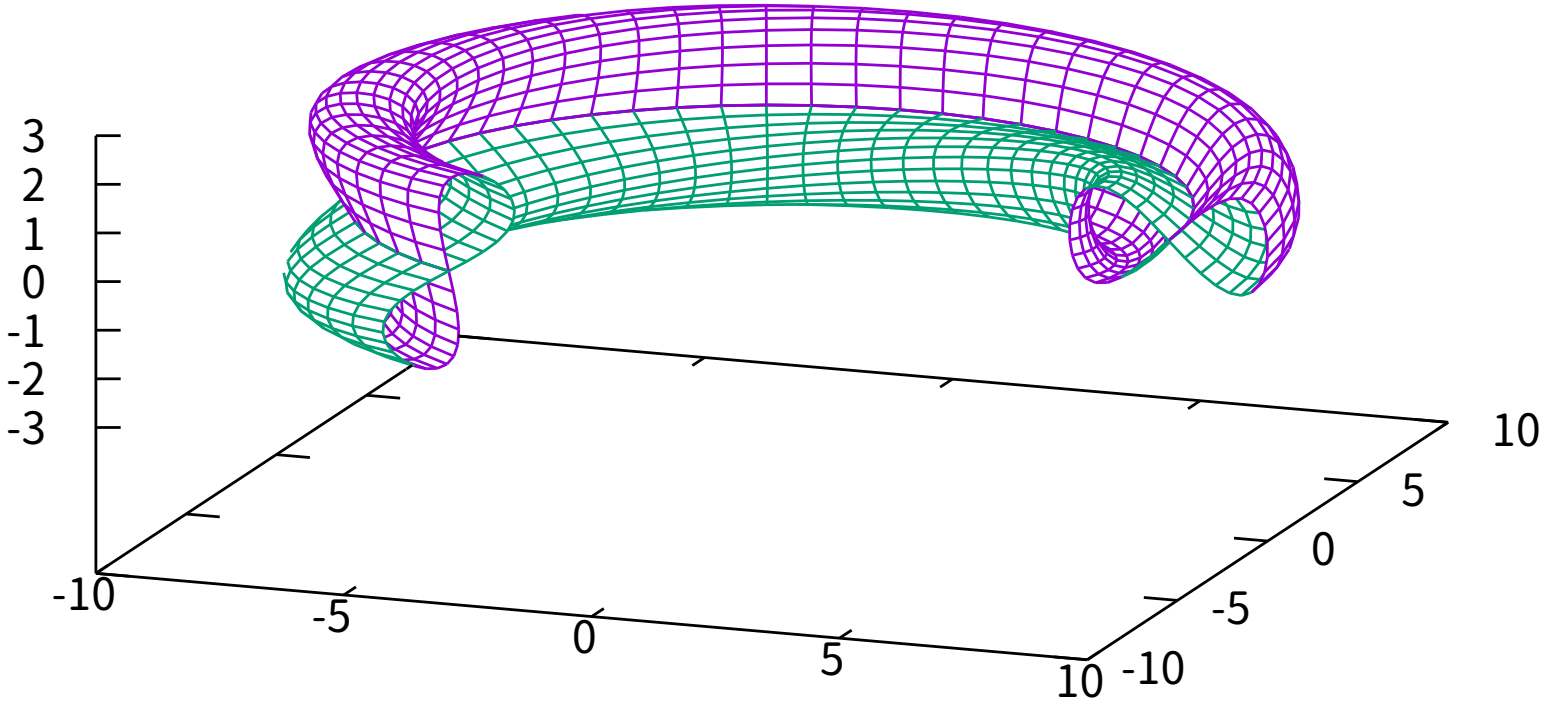
Moebius strip (view from opposite side)



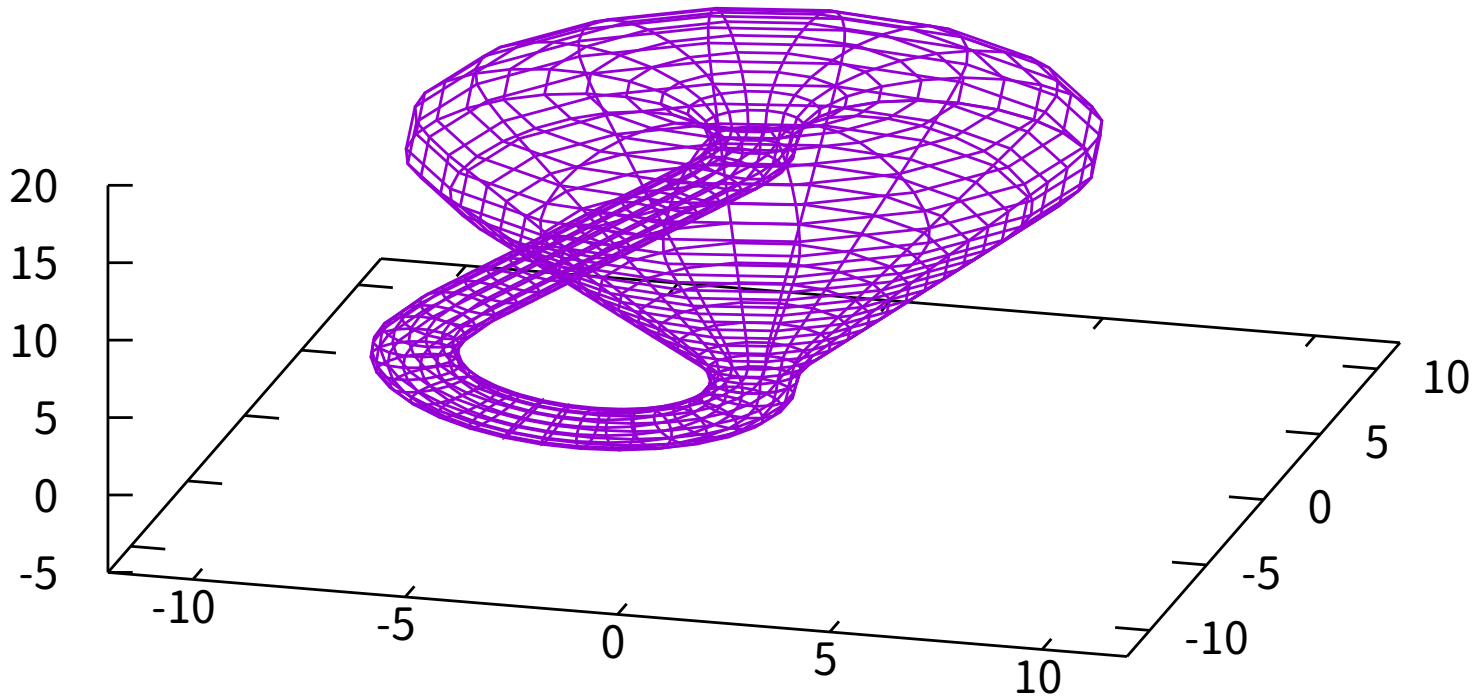
Klein bottle



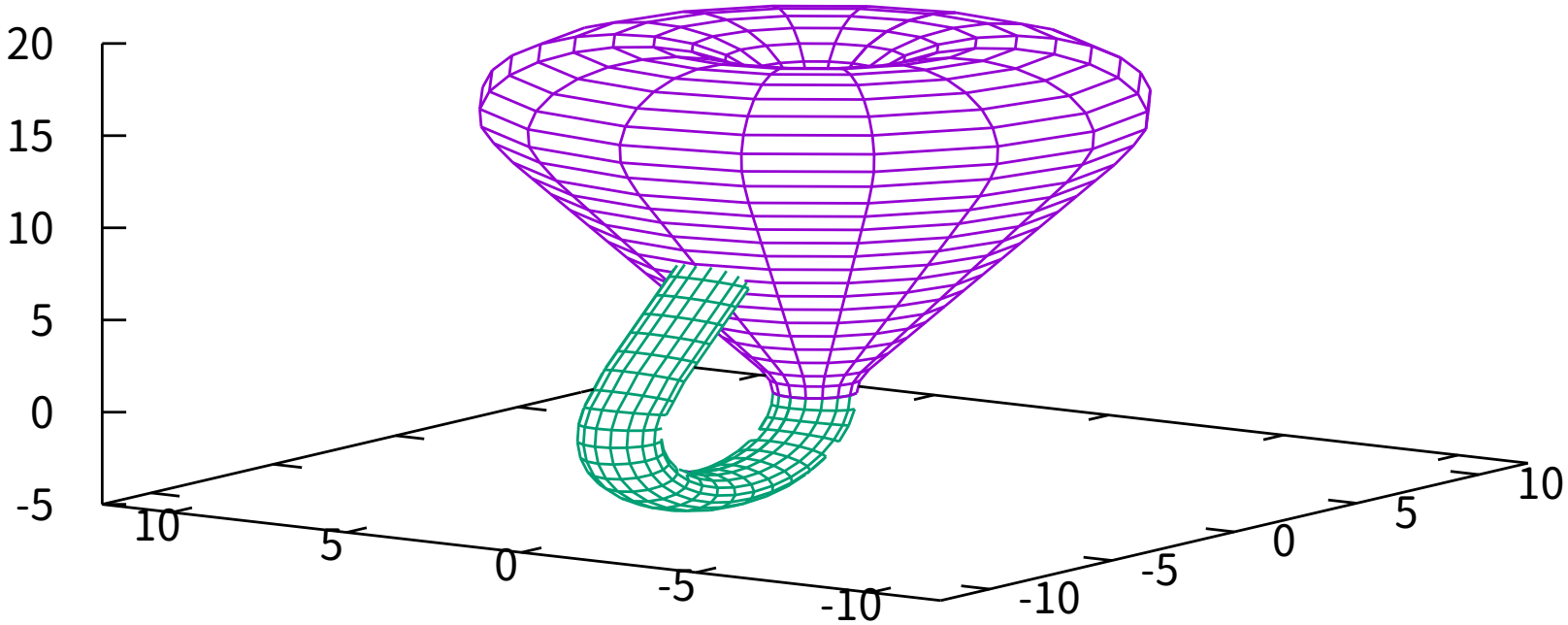
Klein bottle with look at the 'inside'

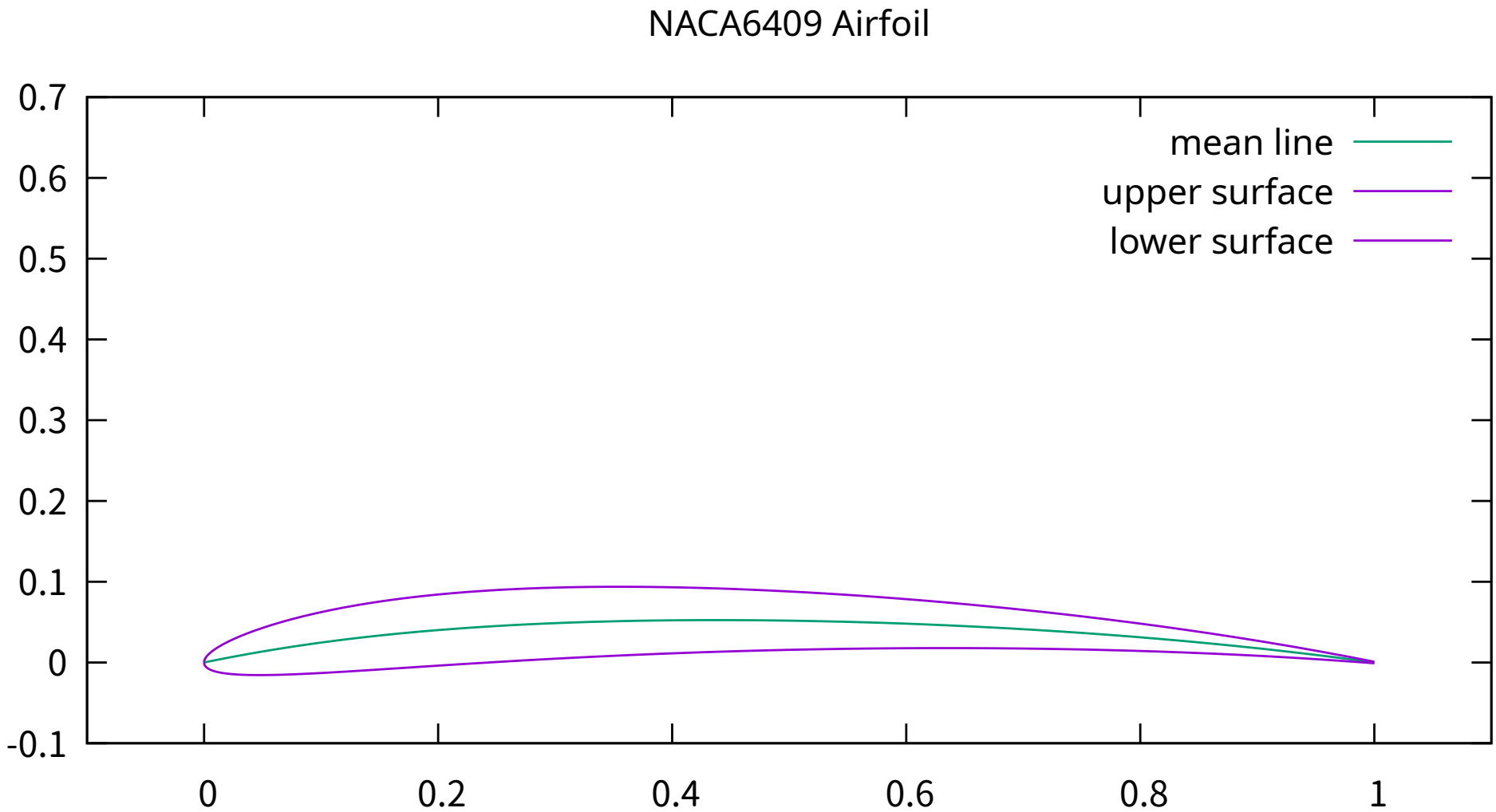


Klein bottle, glassblowers' version (look-through)

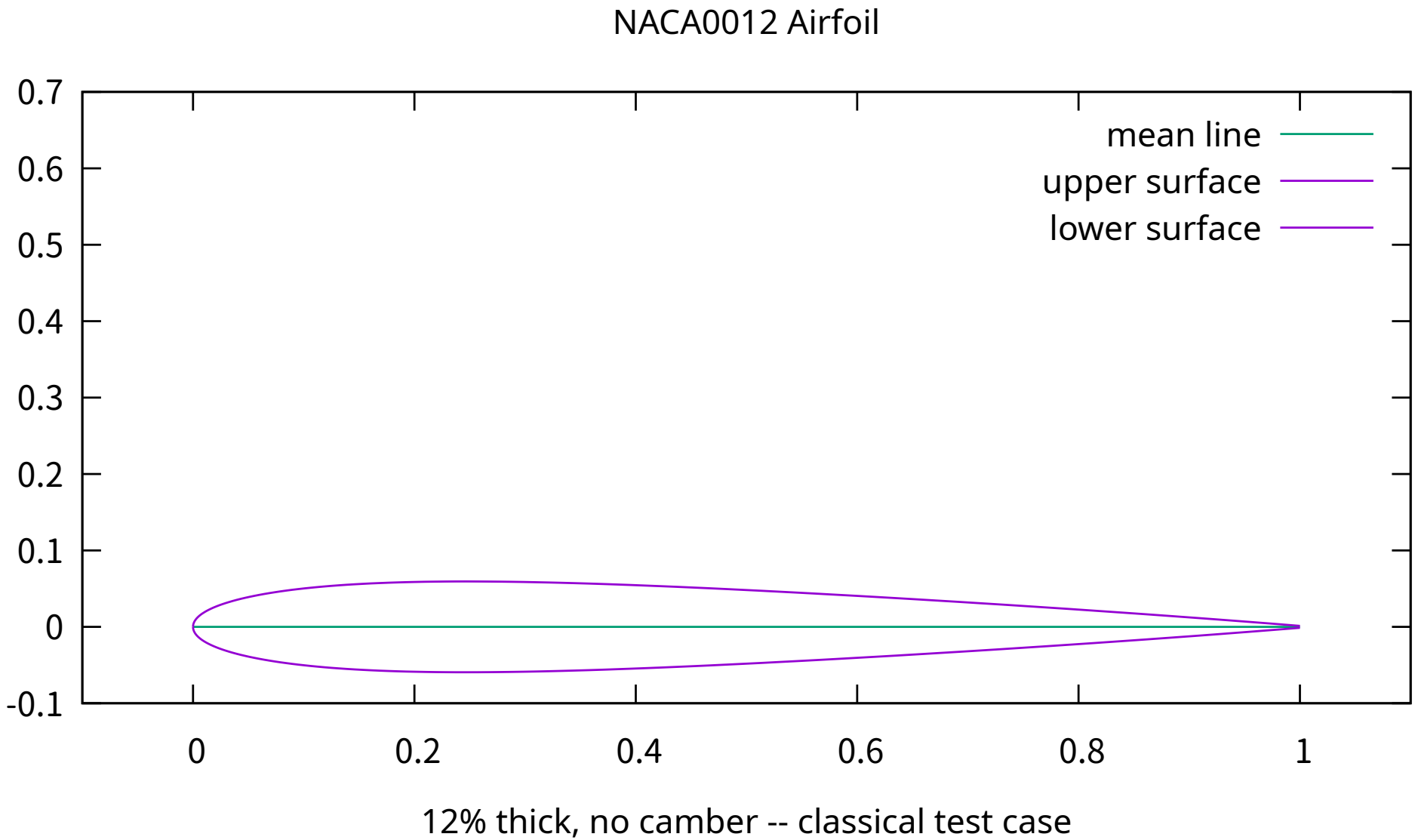


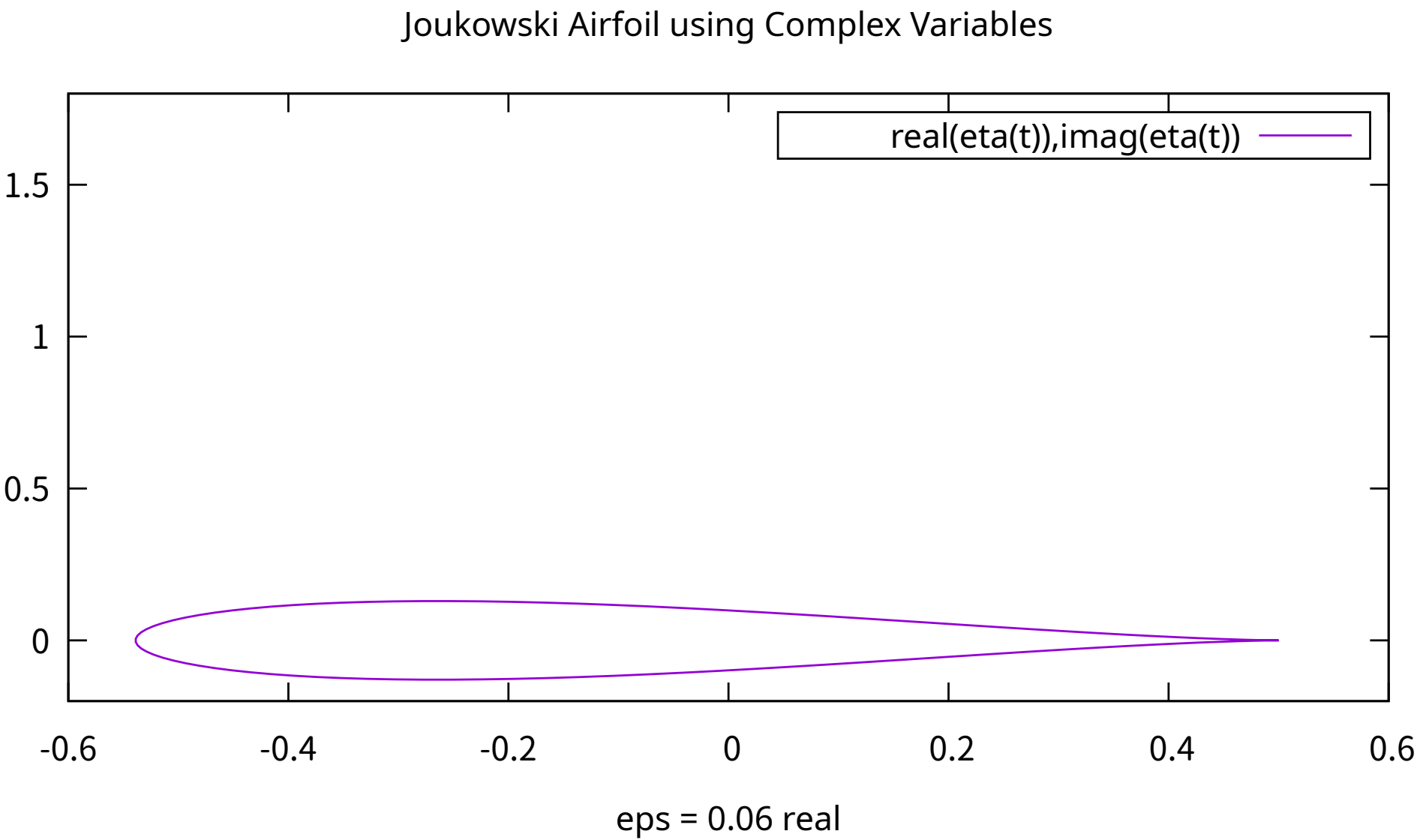
Klein bottle, glassblowers' version (solid)

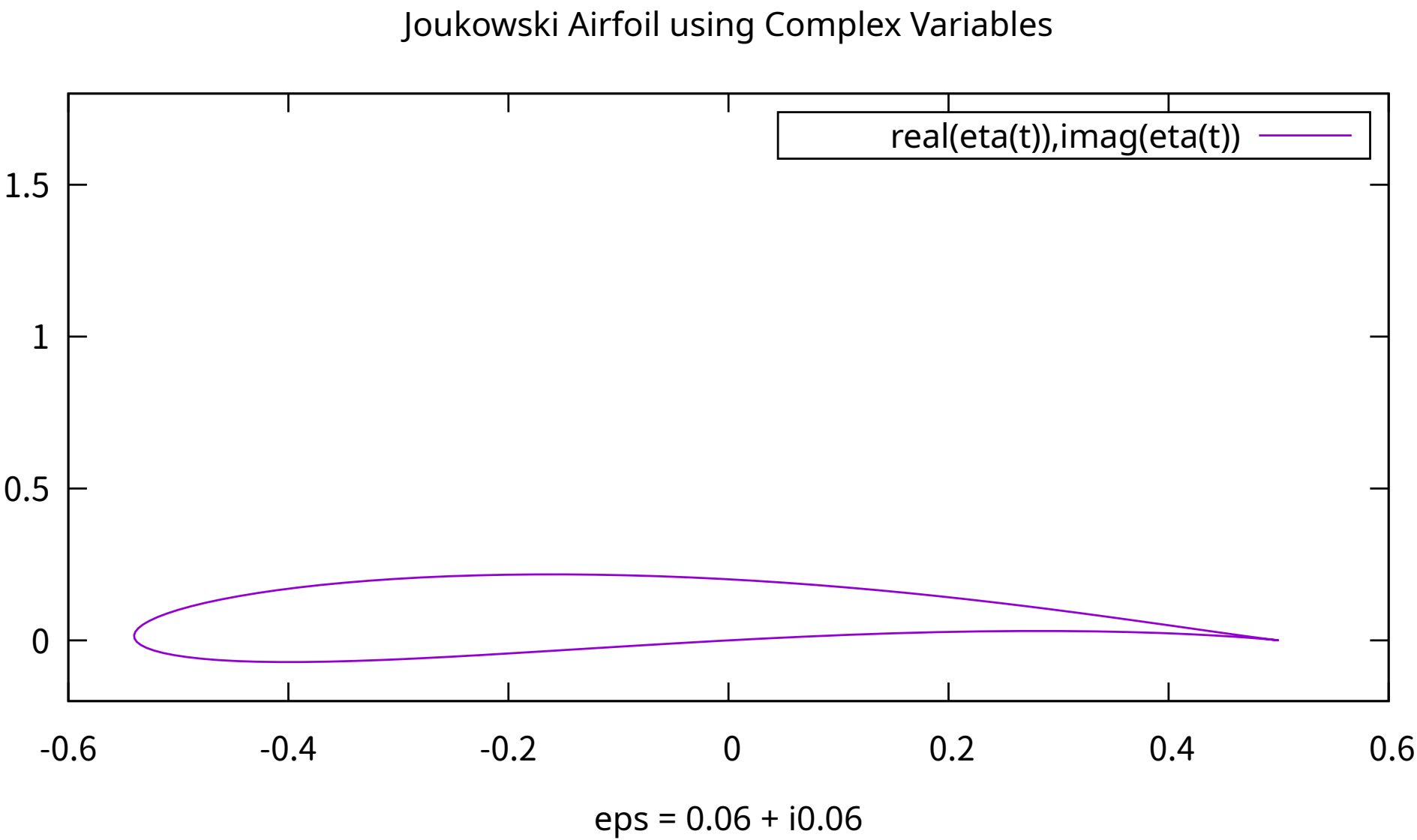




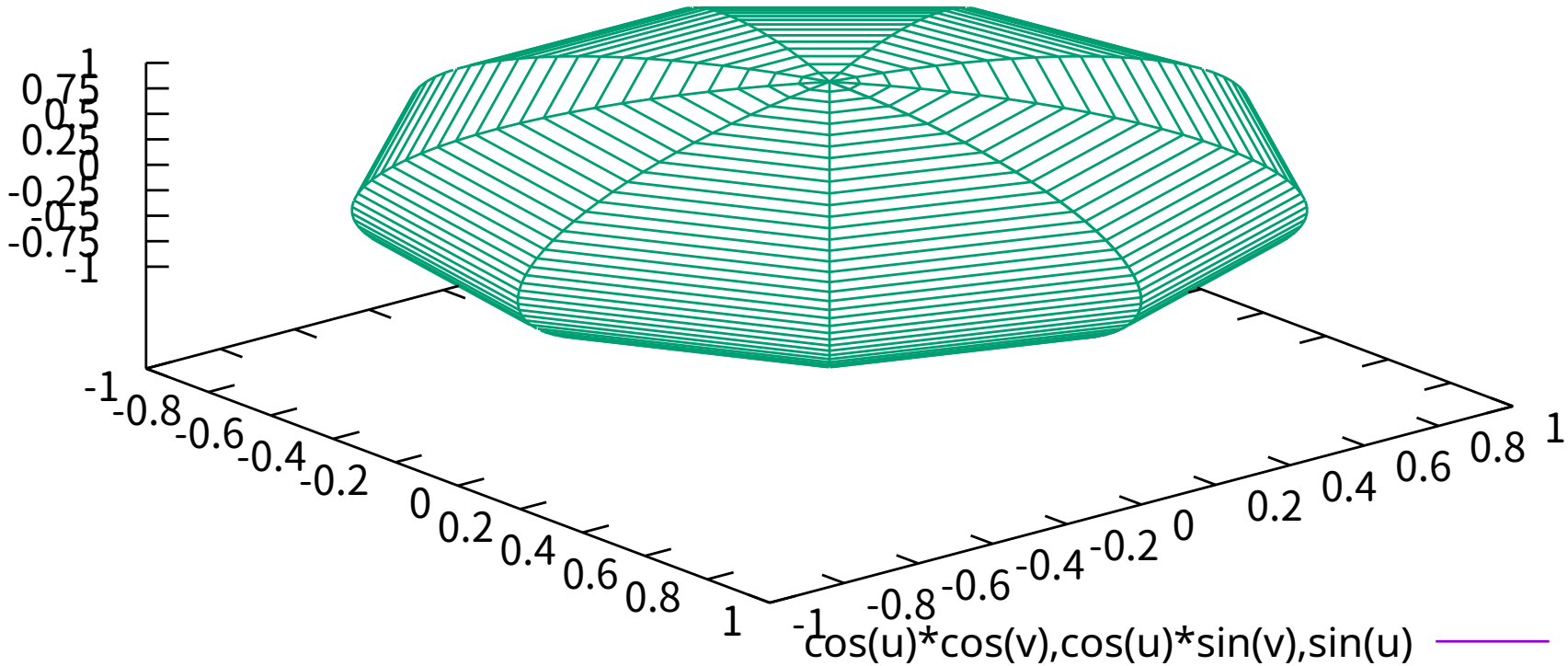
NACA6409 -- 9% thick, 40% max camber, 6% camber


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

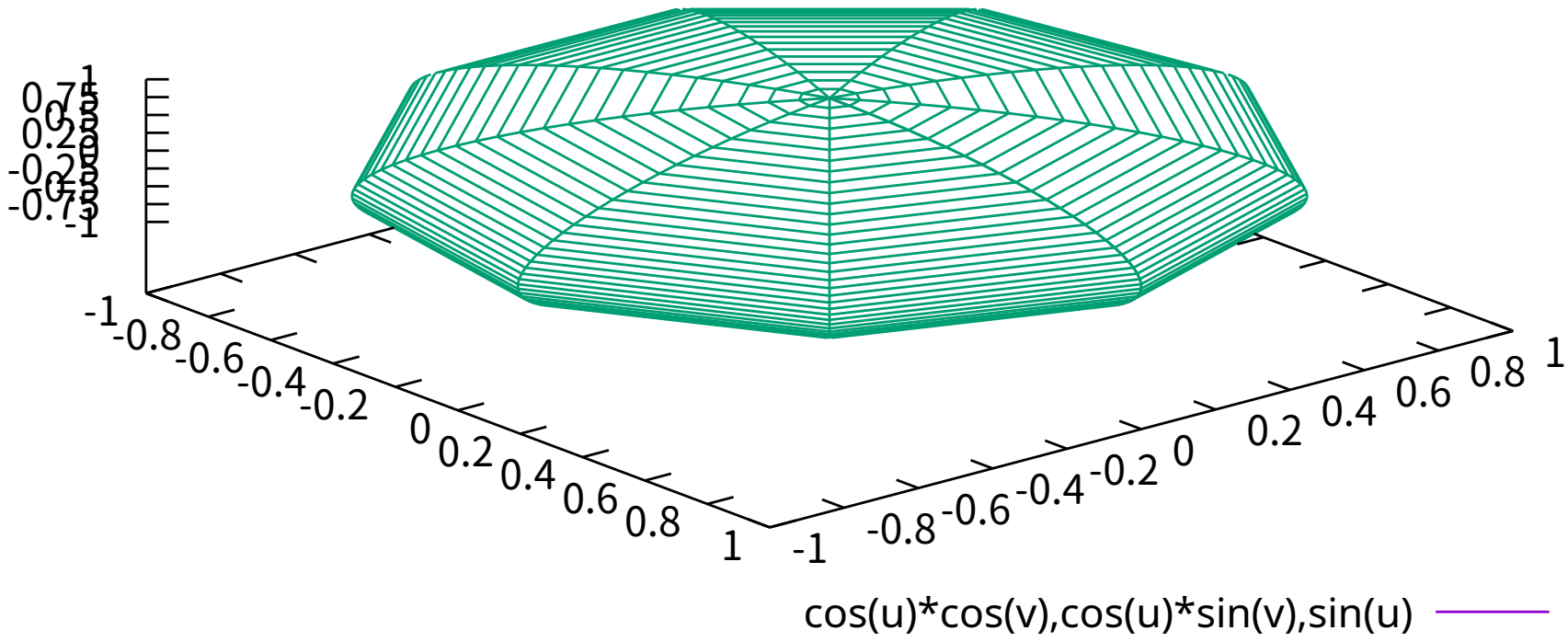

Joukowski Airfoil using Complex Variables
real(eta(t)),imag(eta(t))
1.5
1
0.5
0
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
eps = 0.06 real


Joukowski Airfoil using Complex Variables
real(eta(t)),imag(eta(t))
1.5
1
0.5
0
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
eps = 0.06 + i0.06

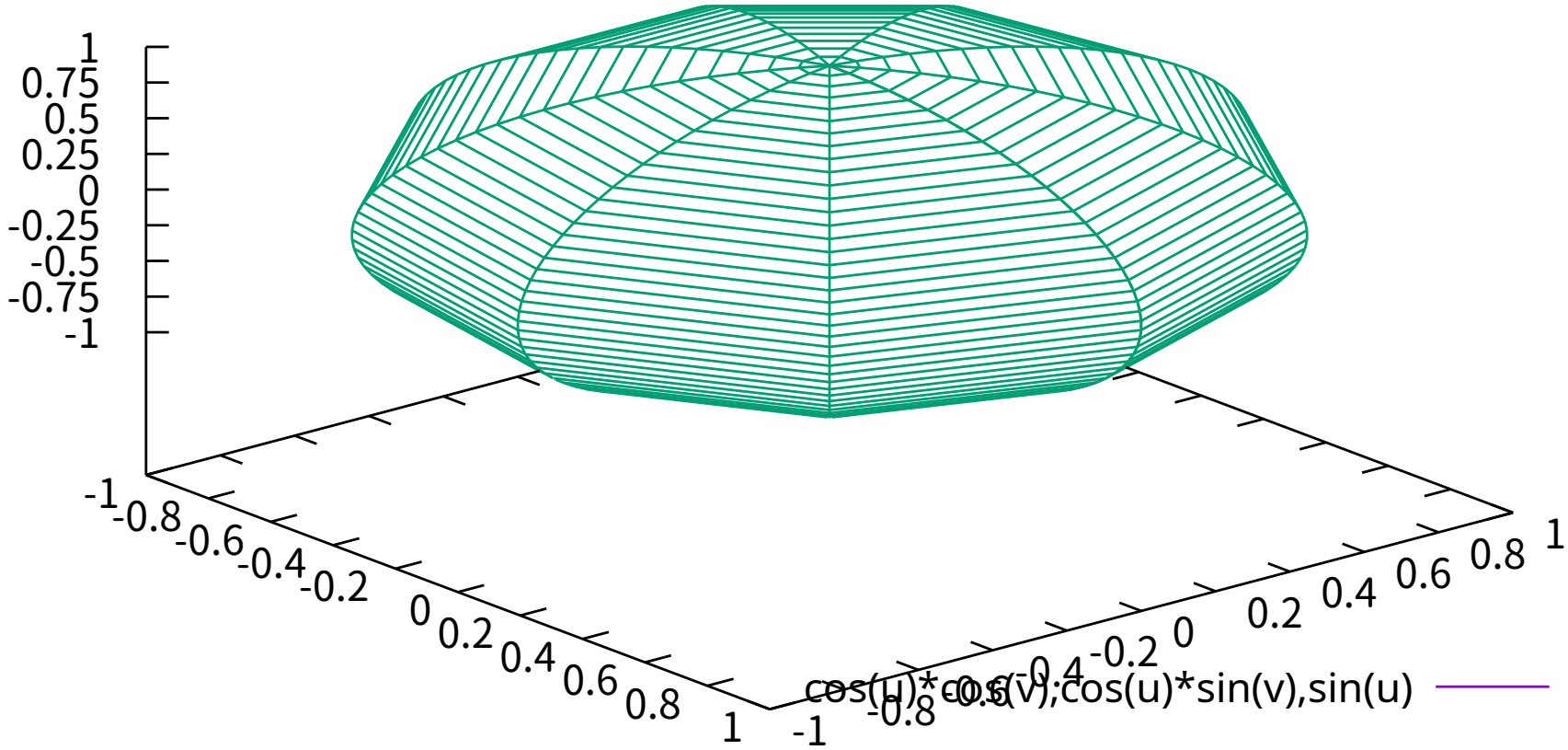
Parametric Sphere



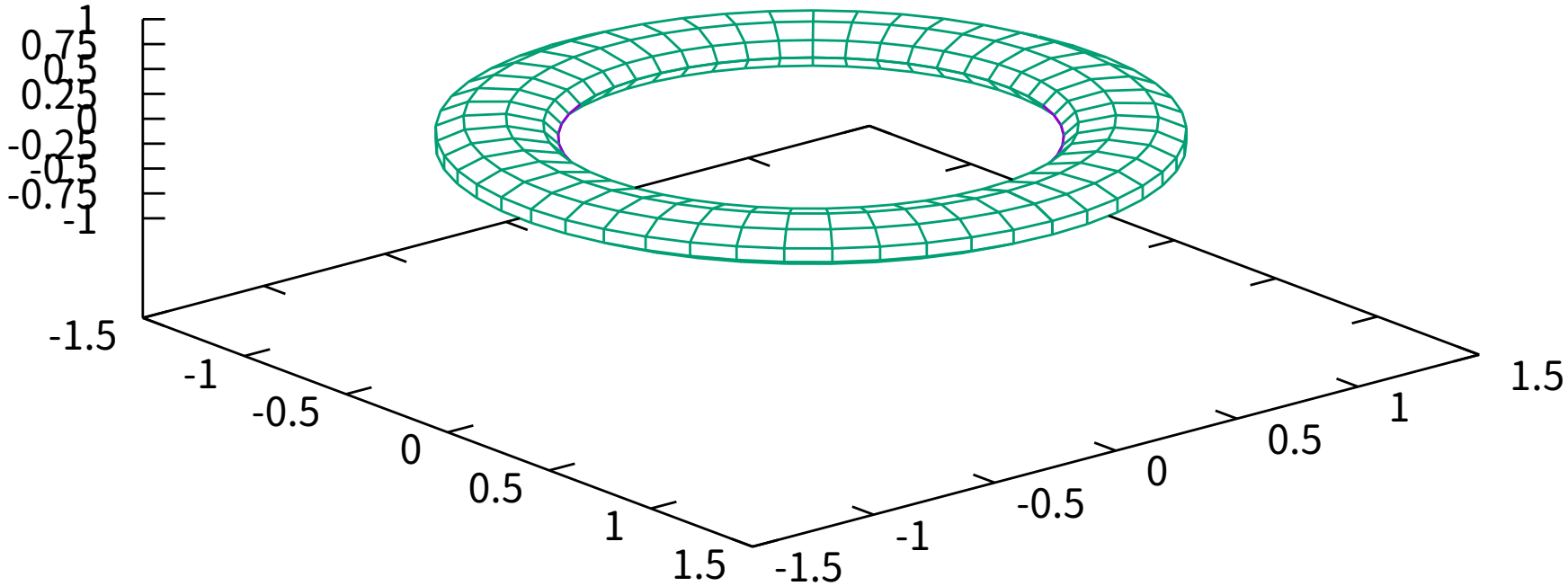
Parametric Sphere, crunched z axis



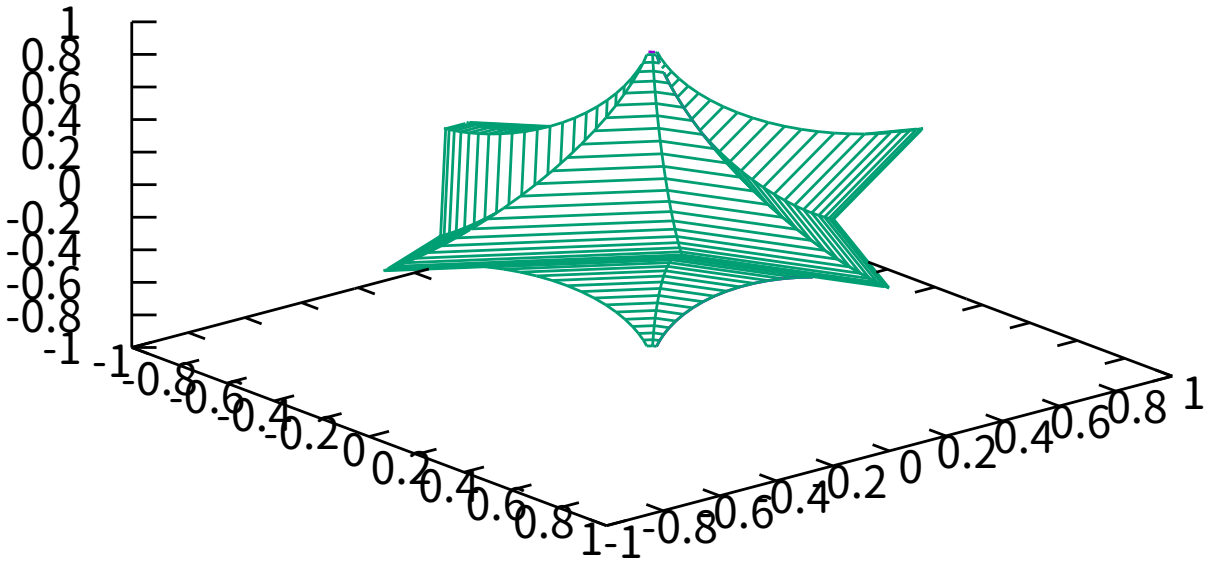
Parametric Sphere, enlarged z axis



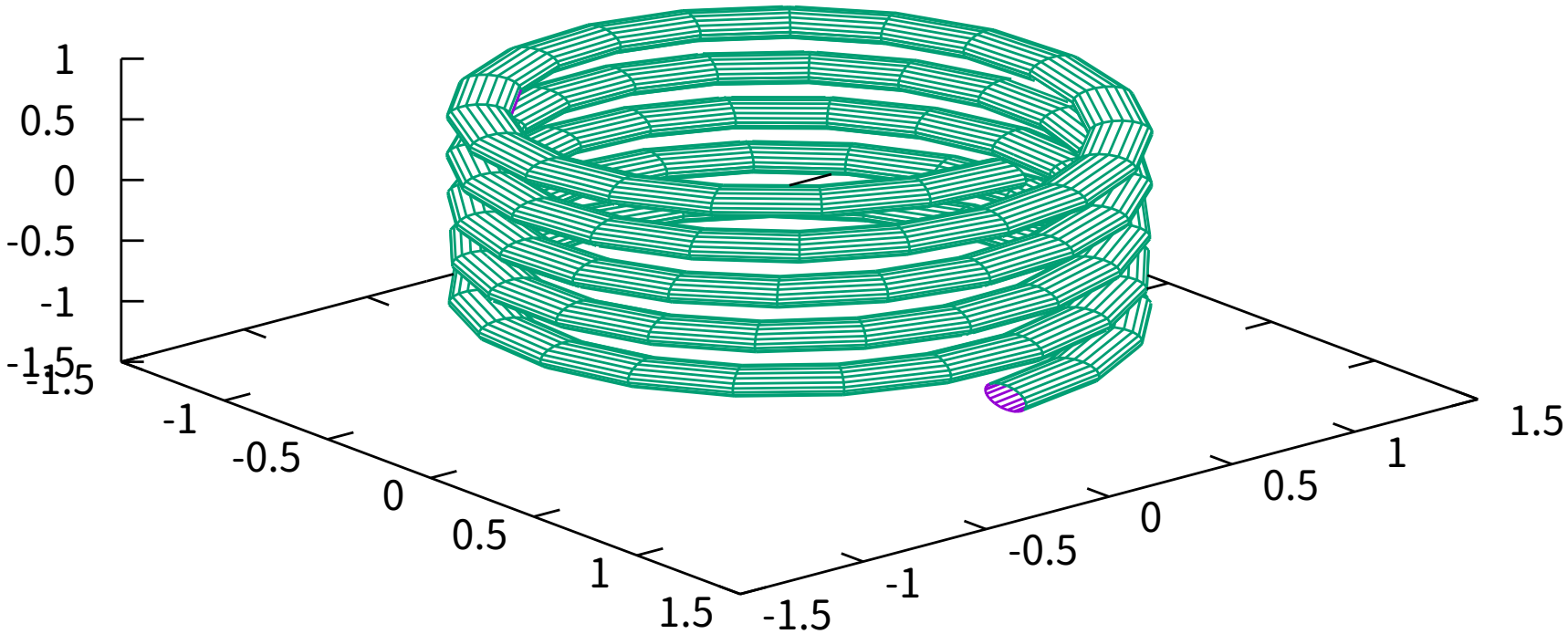
Parametric Torus



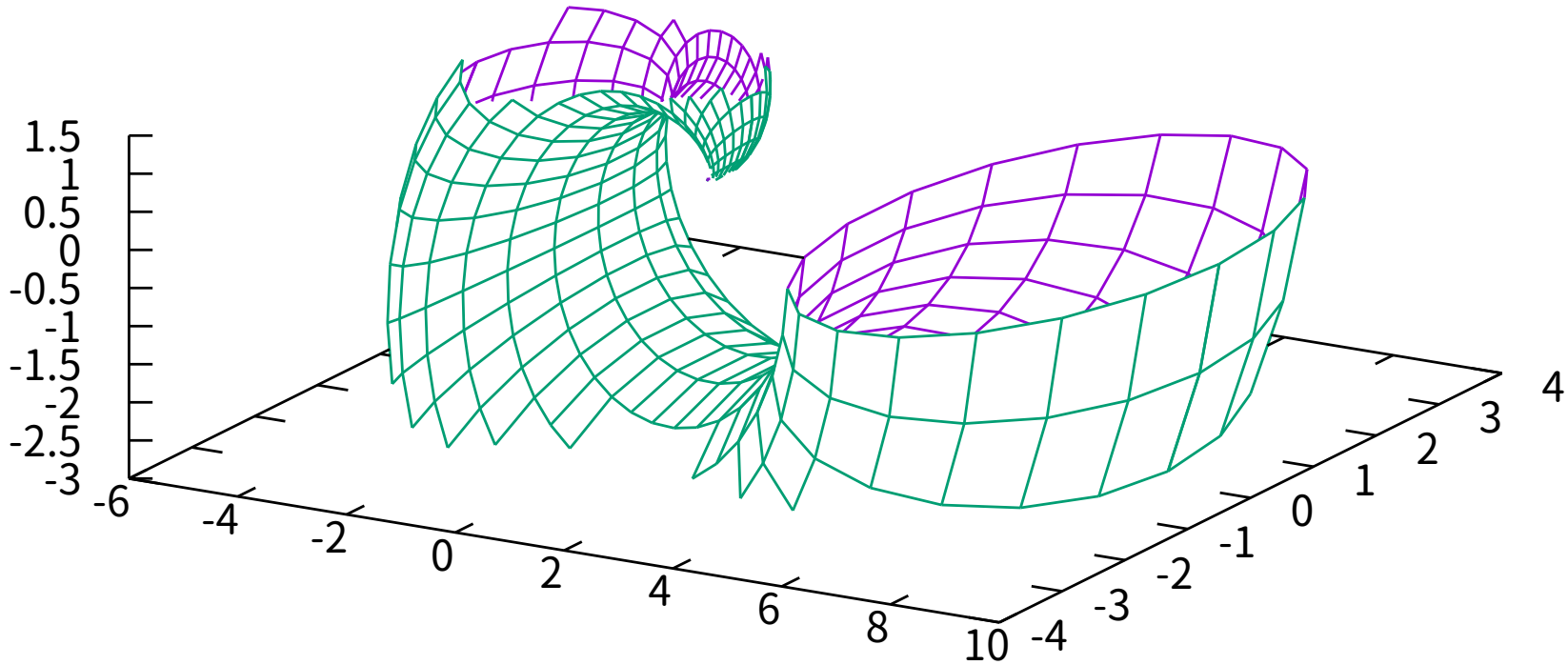
Parametric Hexagon



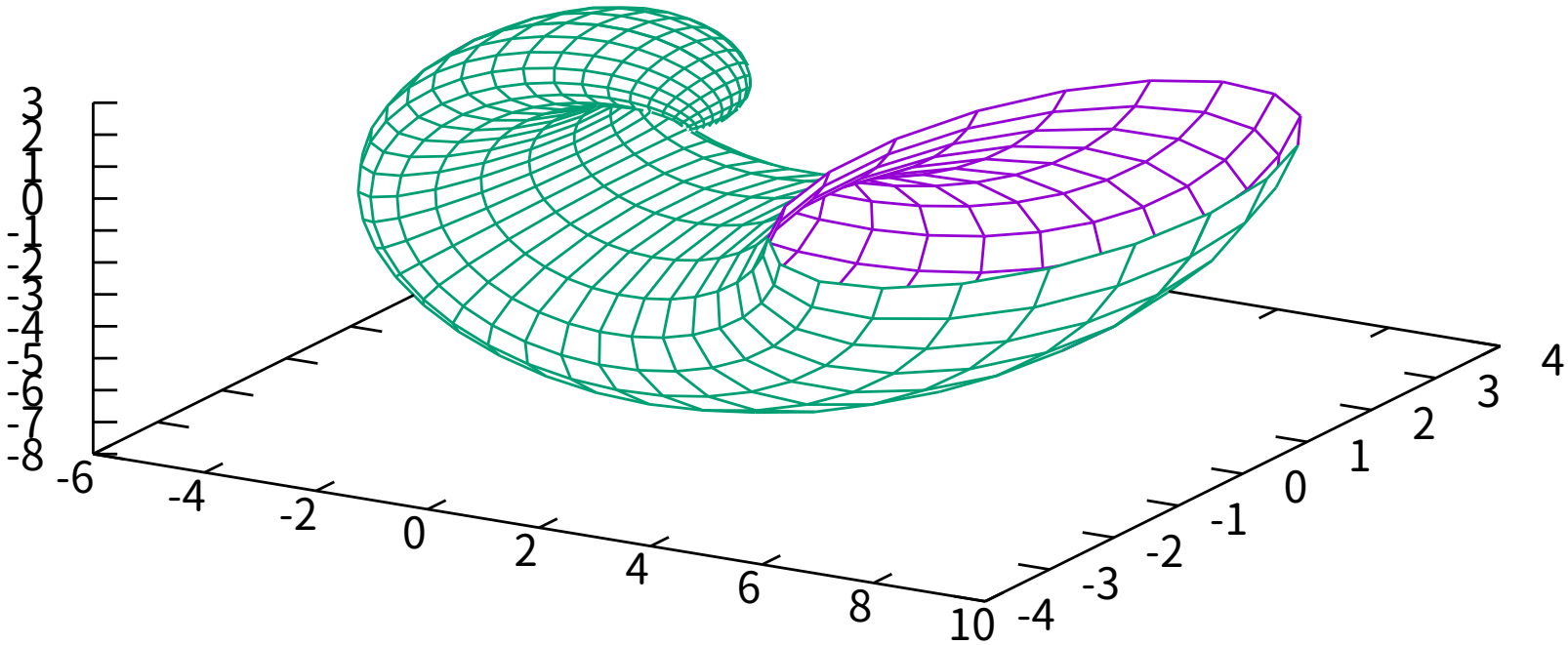
Parametric Helix



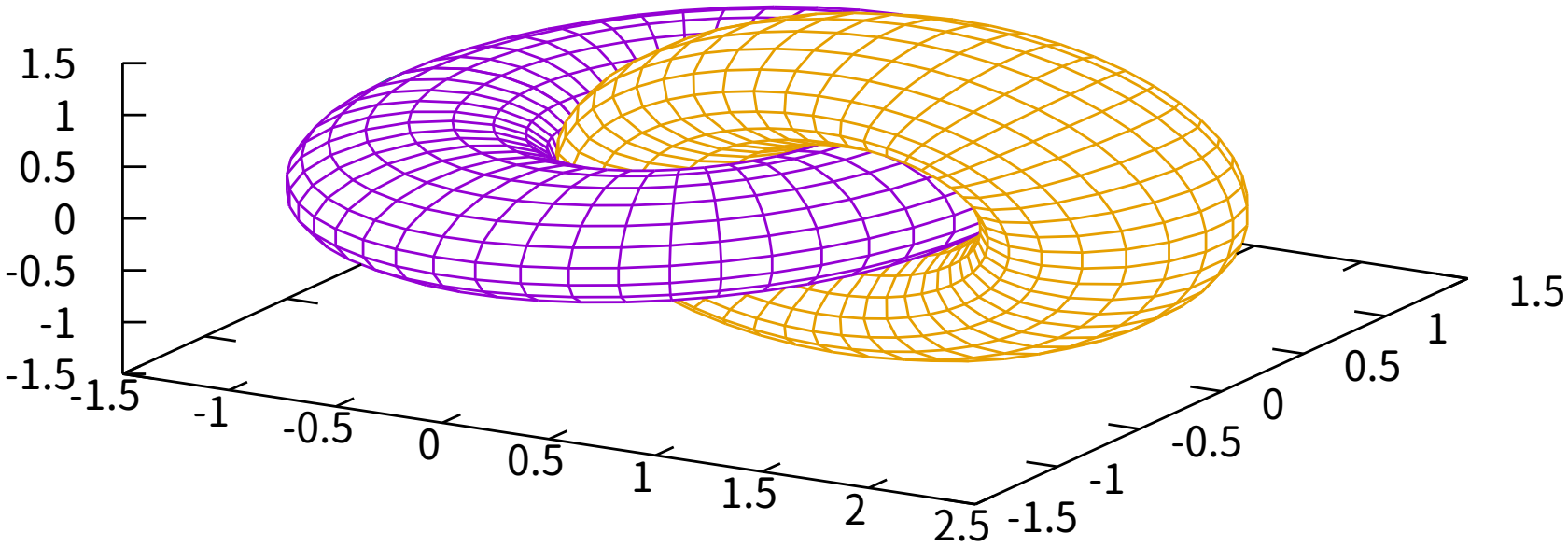
Parametric Shell (clipped to limited z range)



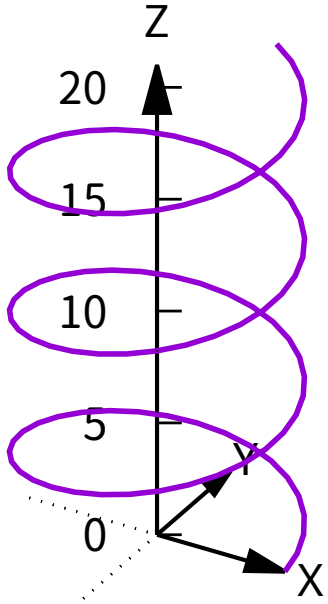
Parametric Shell (automatic z range)



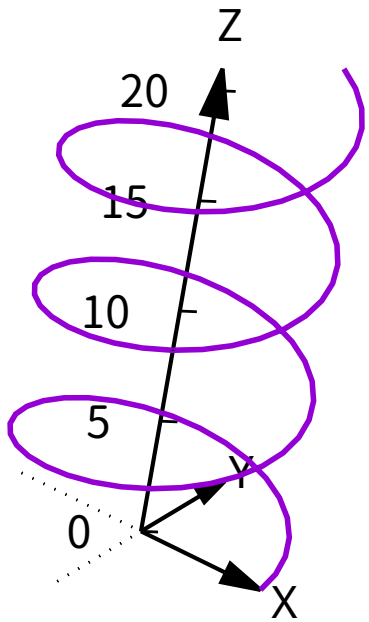
Interlocking Tori



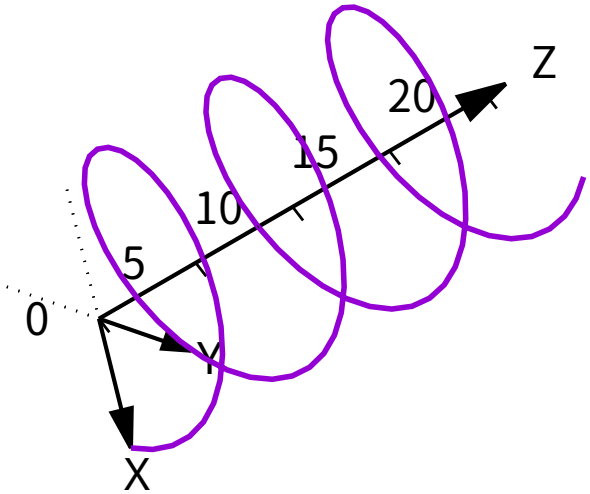
azimuth 0



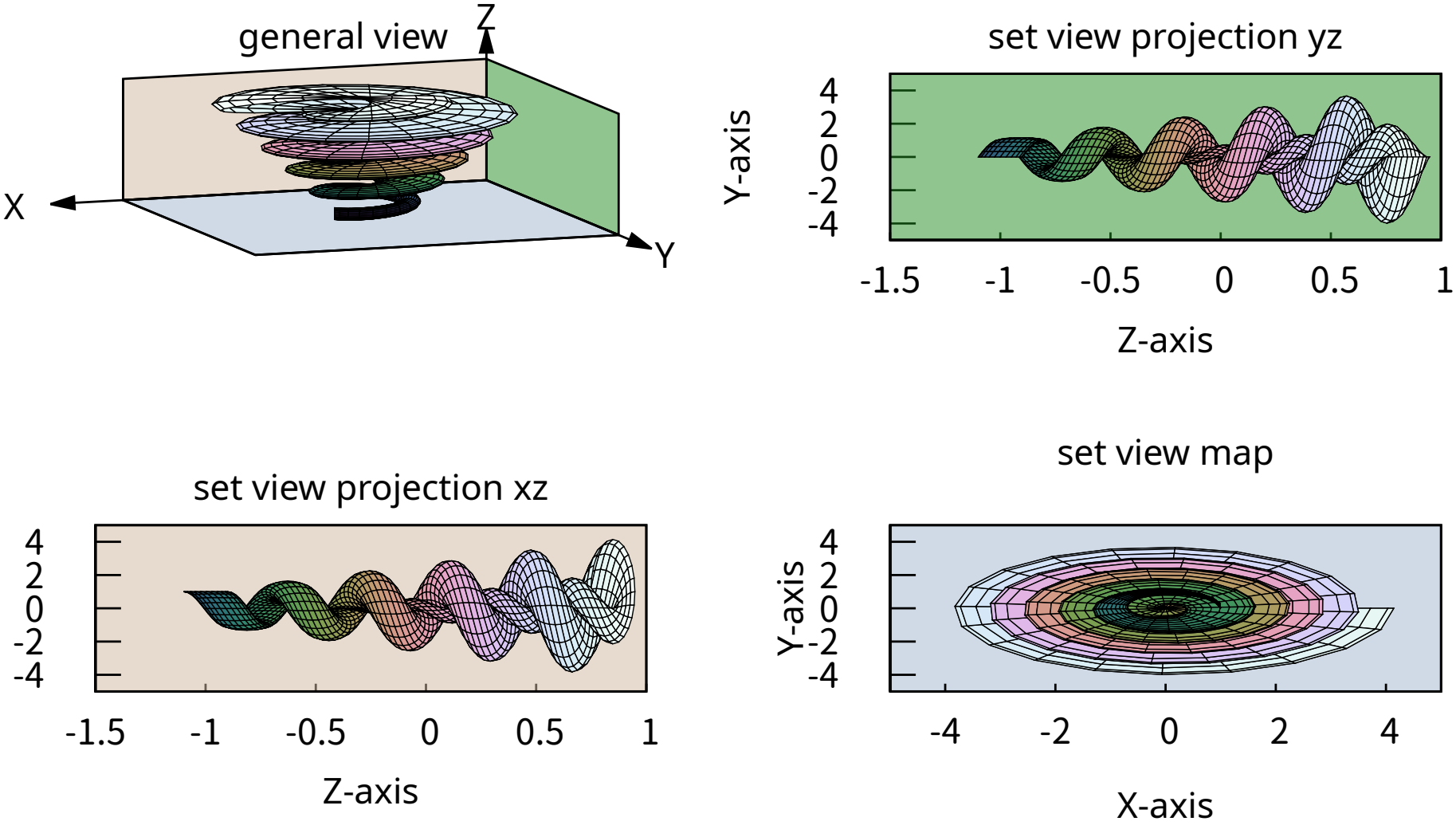
azimuth 10

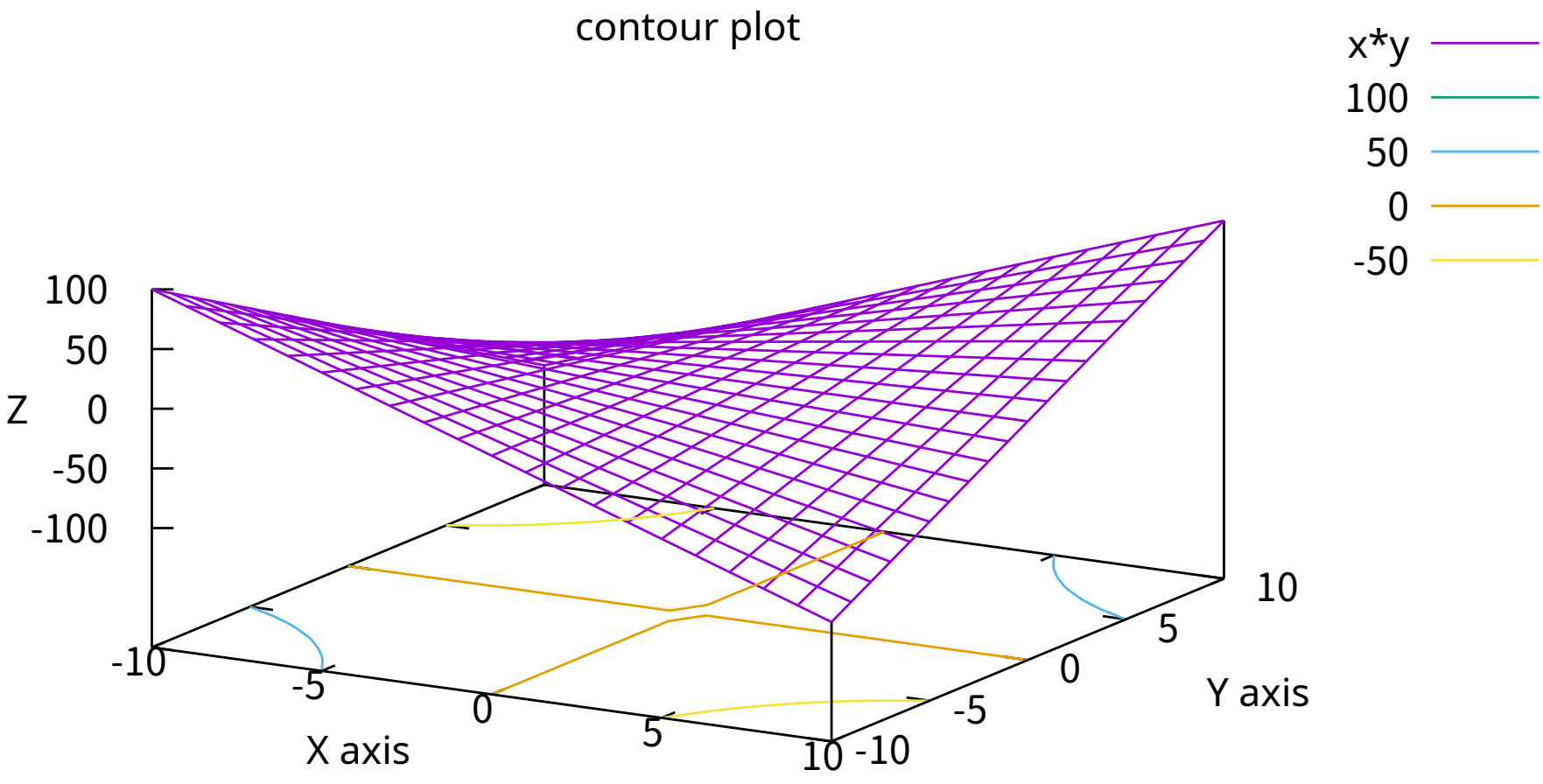


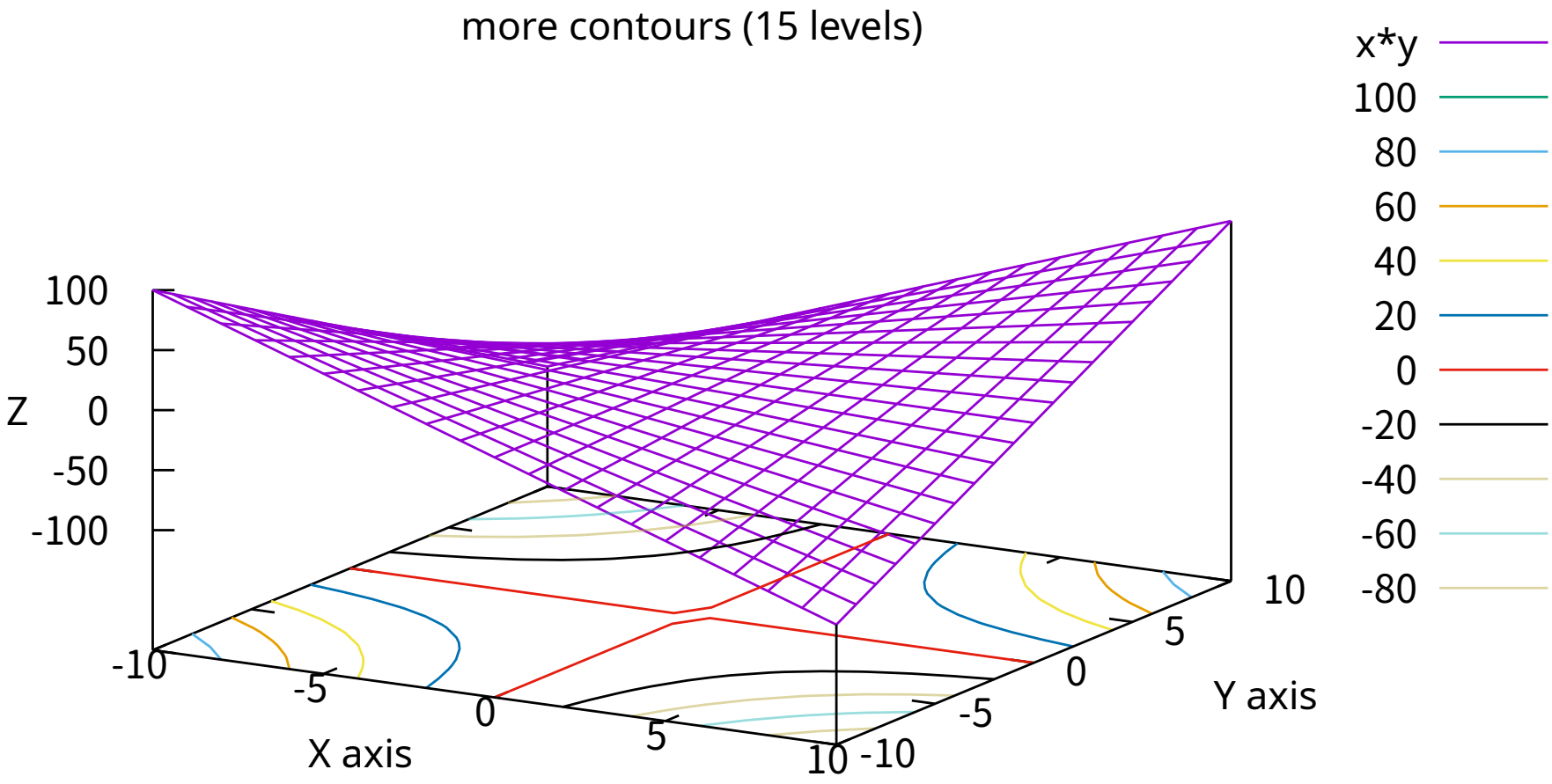
azimuth 60

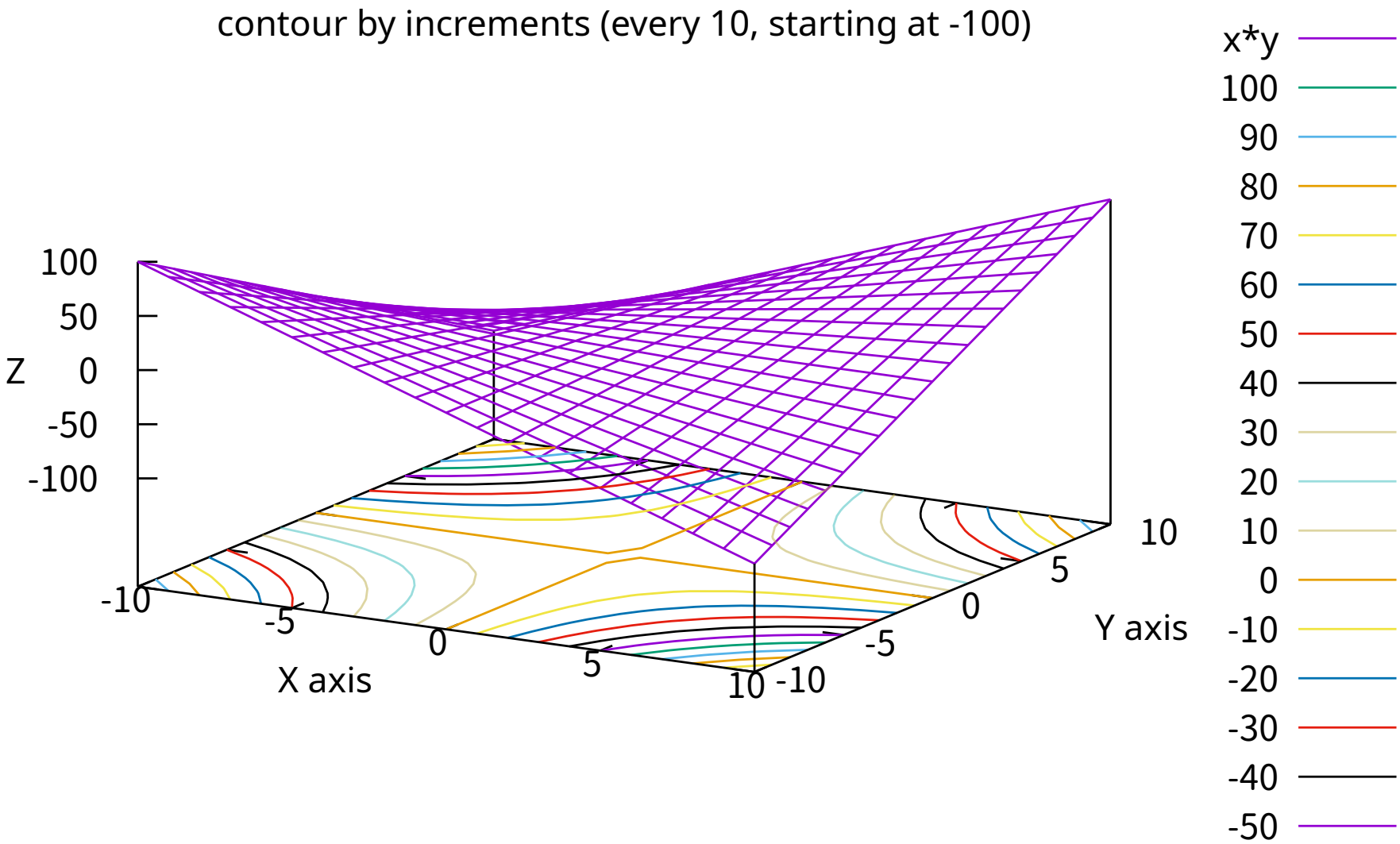


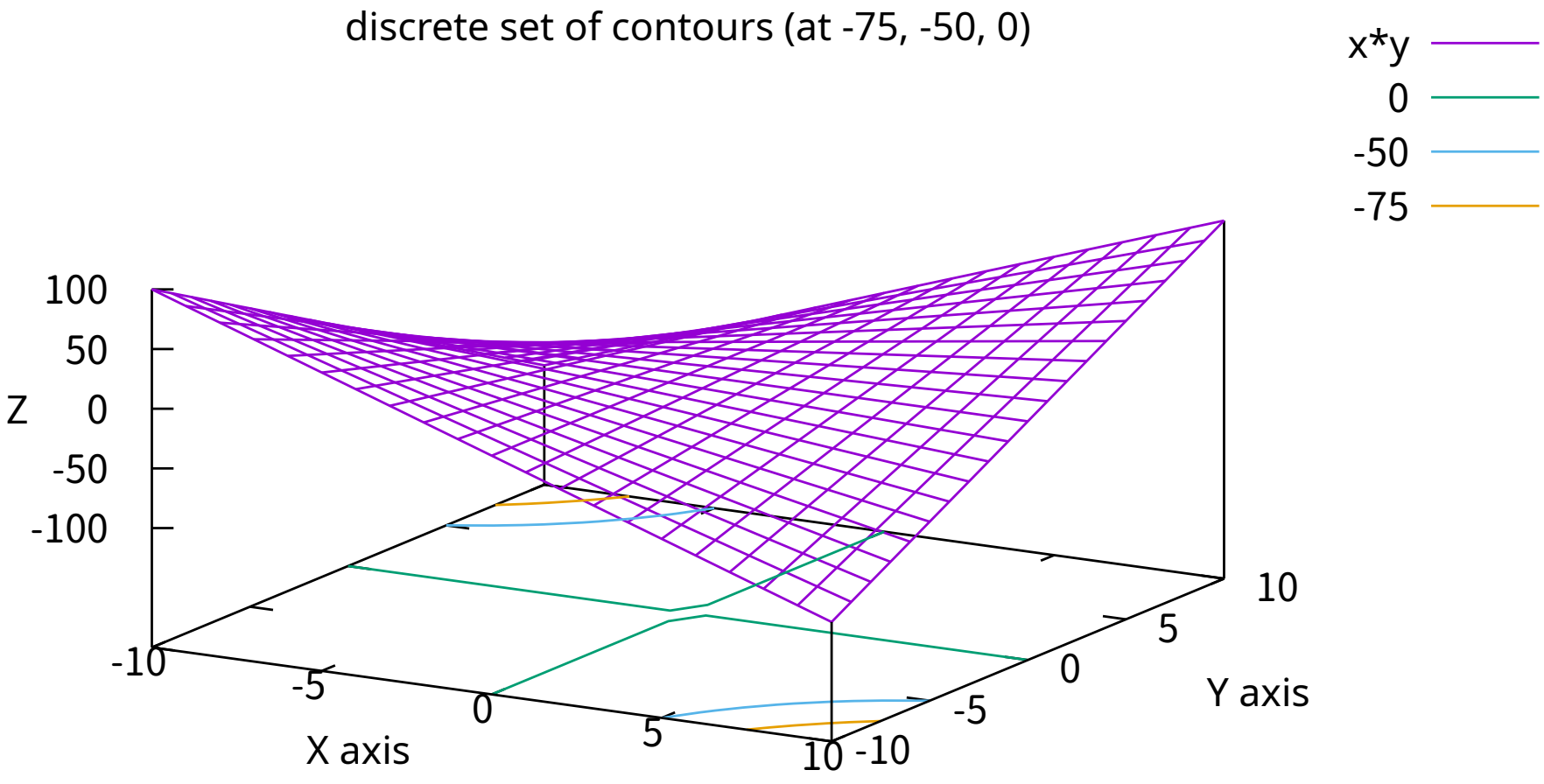
2D projections of a 3D surface

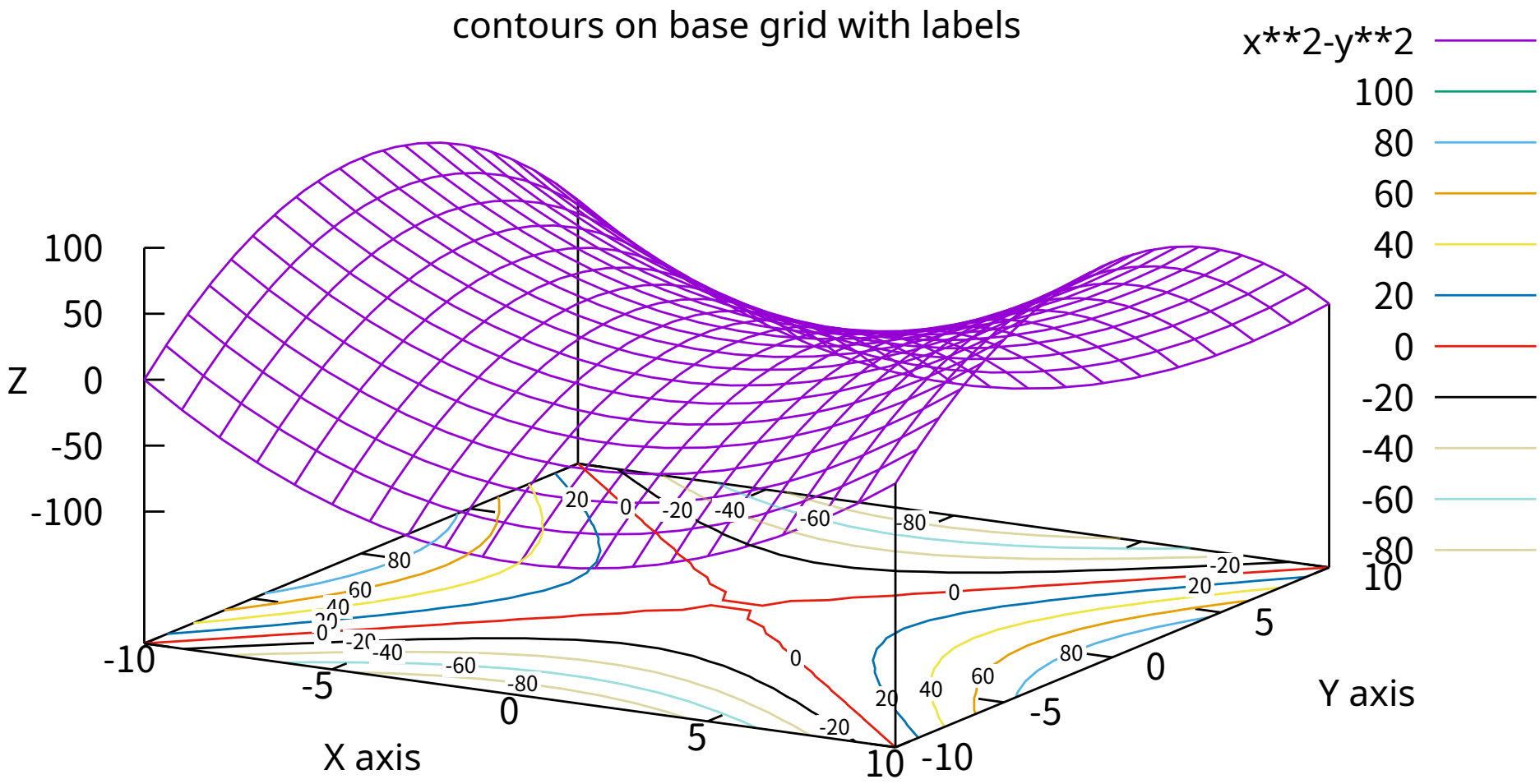


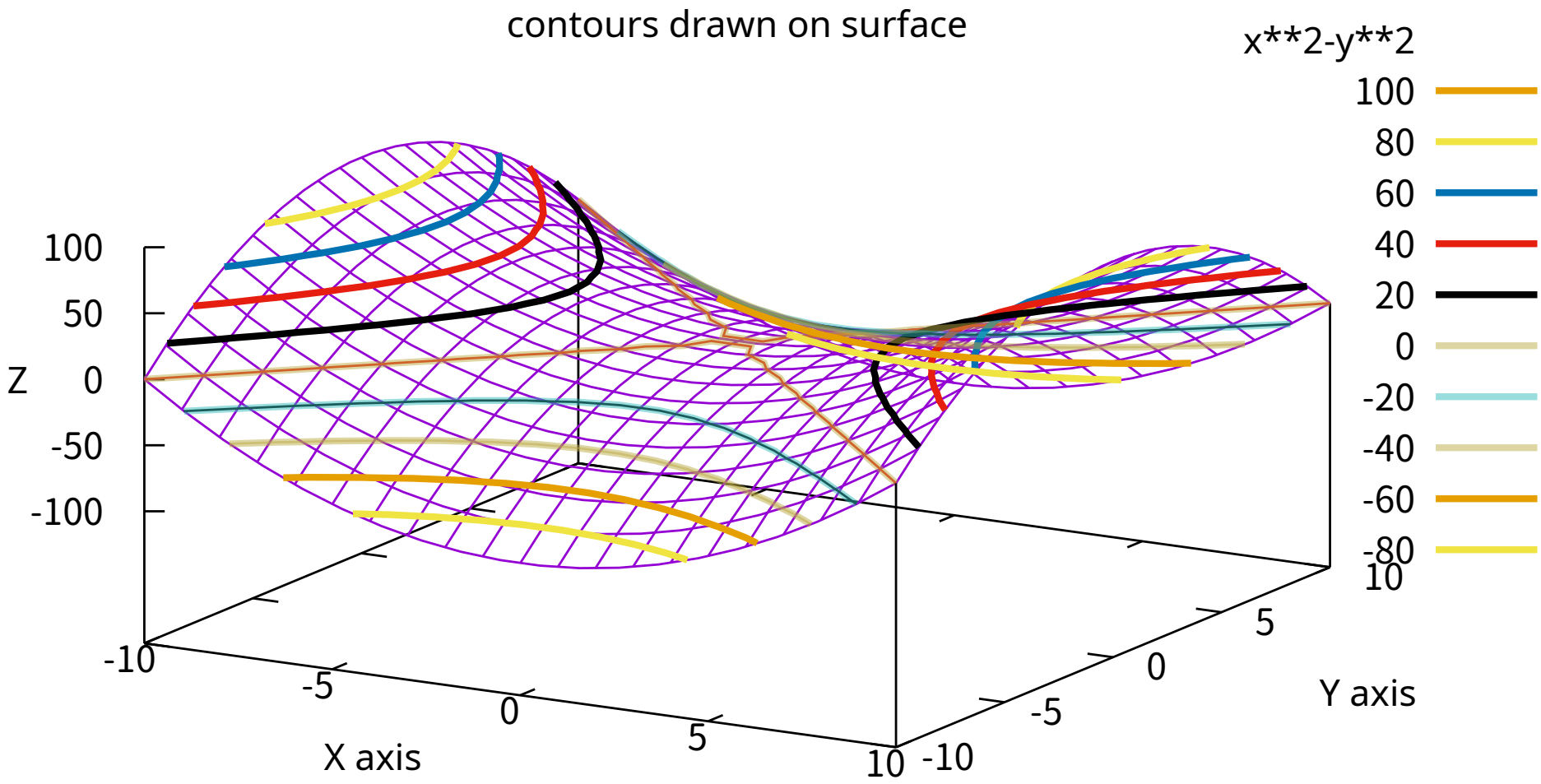

contour plot
x*y
100
50
0
-50
Z
100
50
0
-50
-100
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

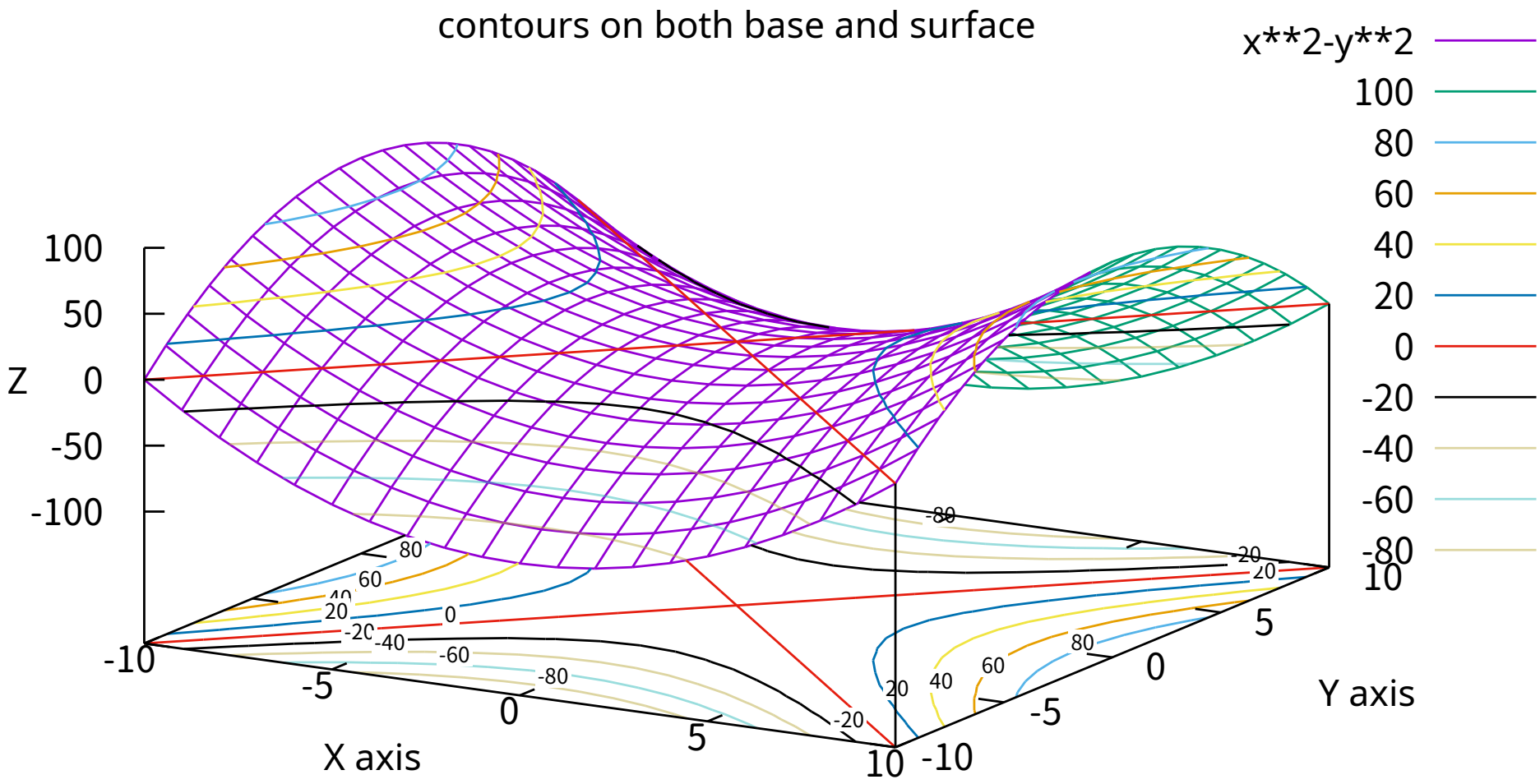

more contours (15 levels)
x*y
100
80
60
40
20
0
-20
-40
-60
-80
Z
100
50
0
-50
-100
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

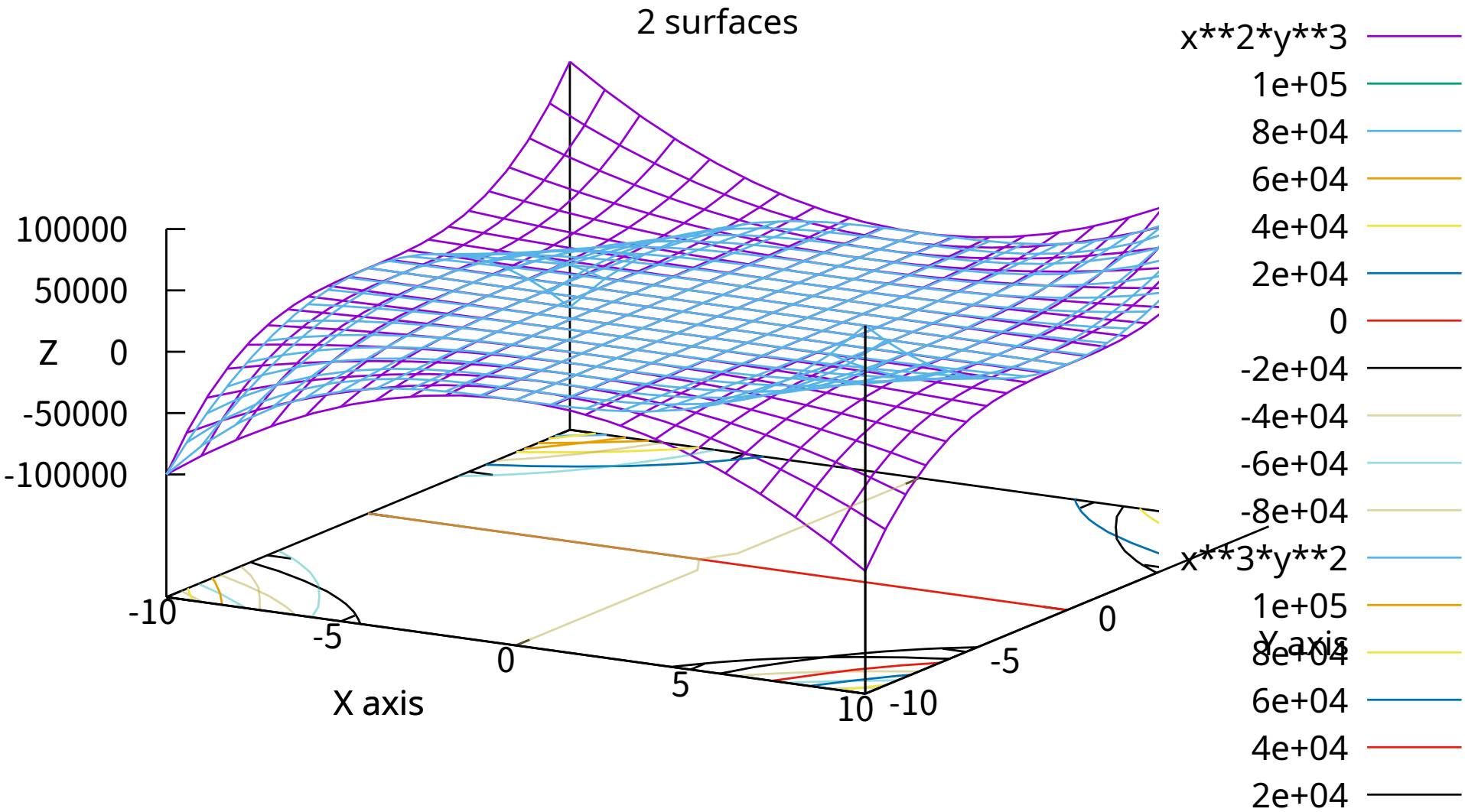

contour by increments (every 10, starting at -100)
Z
100
50
0
-50
-100
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis
x*y
100
90
80
70
60
50
40
30
20
10
0
-10
-20
-30
-40
-50


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

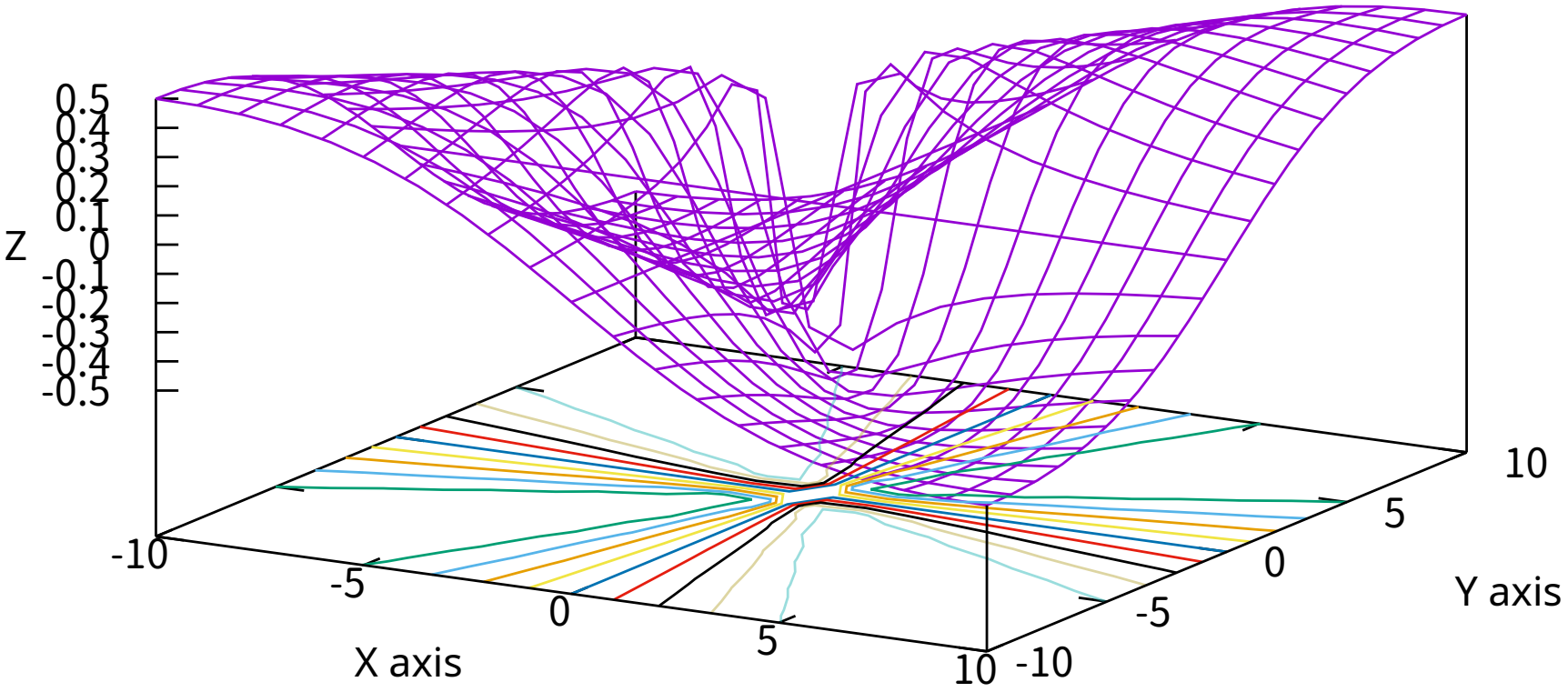

contours on base grid with labels
x**2-y**2
100
80
60
40
20
0
-20
-40
-60
-80
Z
100
50
0
-50
-100
X axis
-10
-5
0
5
10
Y axis
-10
-5
0
5
10

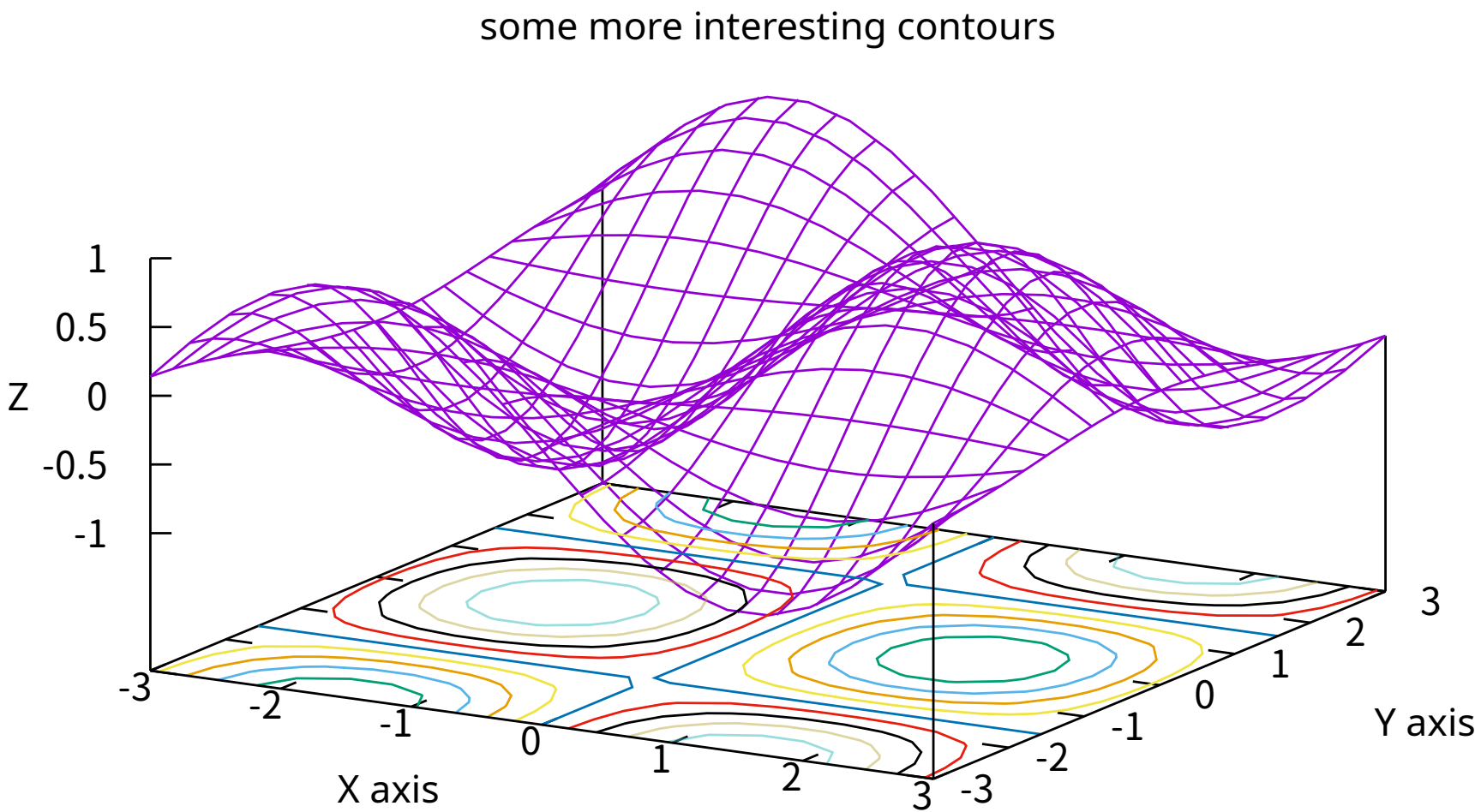

contours drawn on surface
x**2-y**2
100
80
60
40
20
0
-20
-40
-60
-80
Z
100
50
0
-50
-100
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

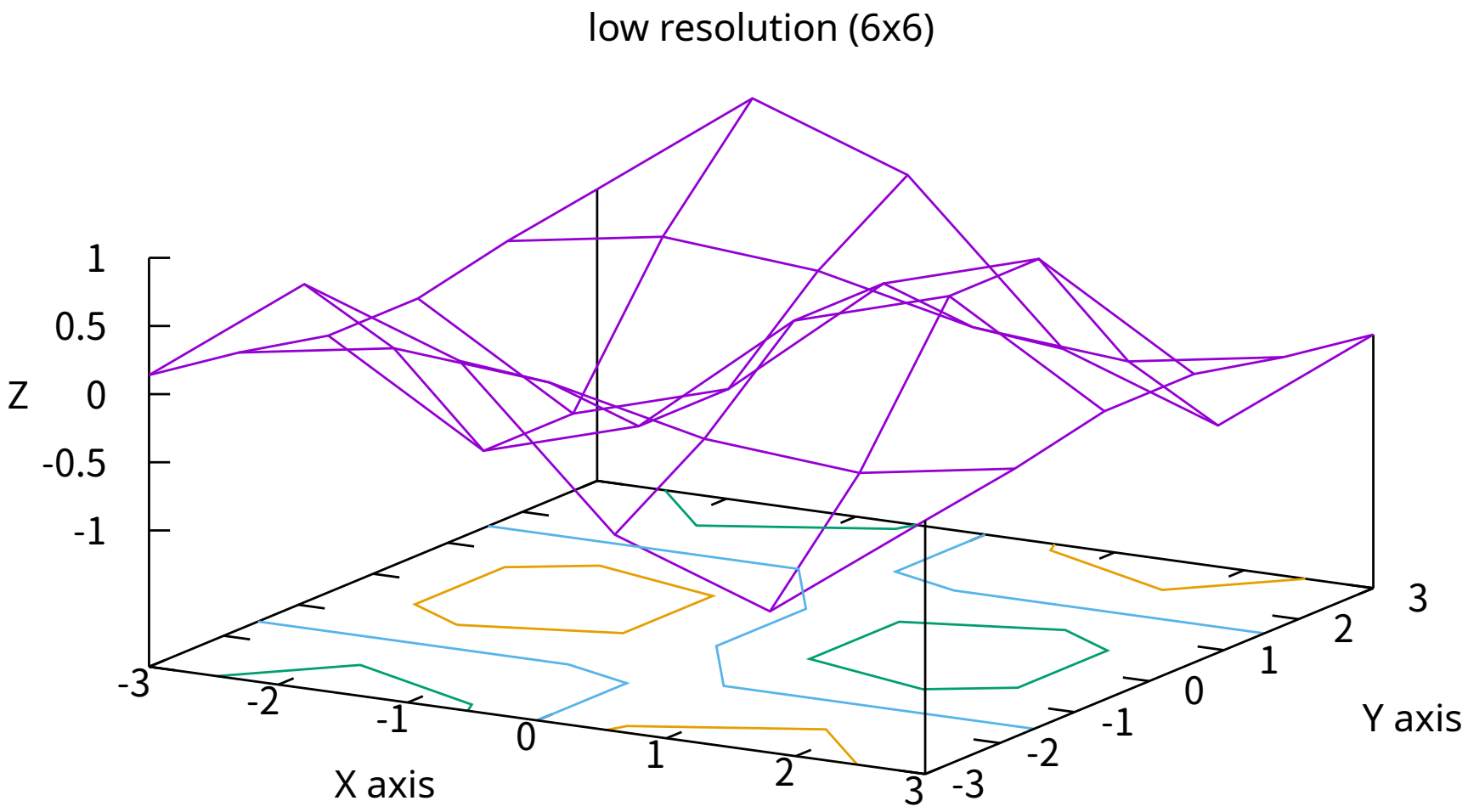

contours on both base and surface
x**2-y**2
100
80
60
40
20
0
-20
-40
-60
-80
Z
100
50
0
-50
-100
X axis
-10
-5
0
5
10
Y axis
-10
-5
0
5
10

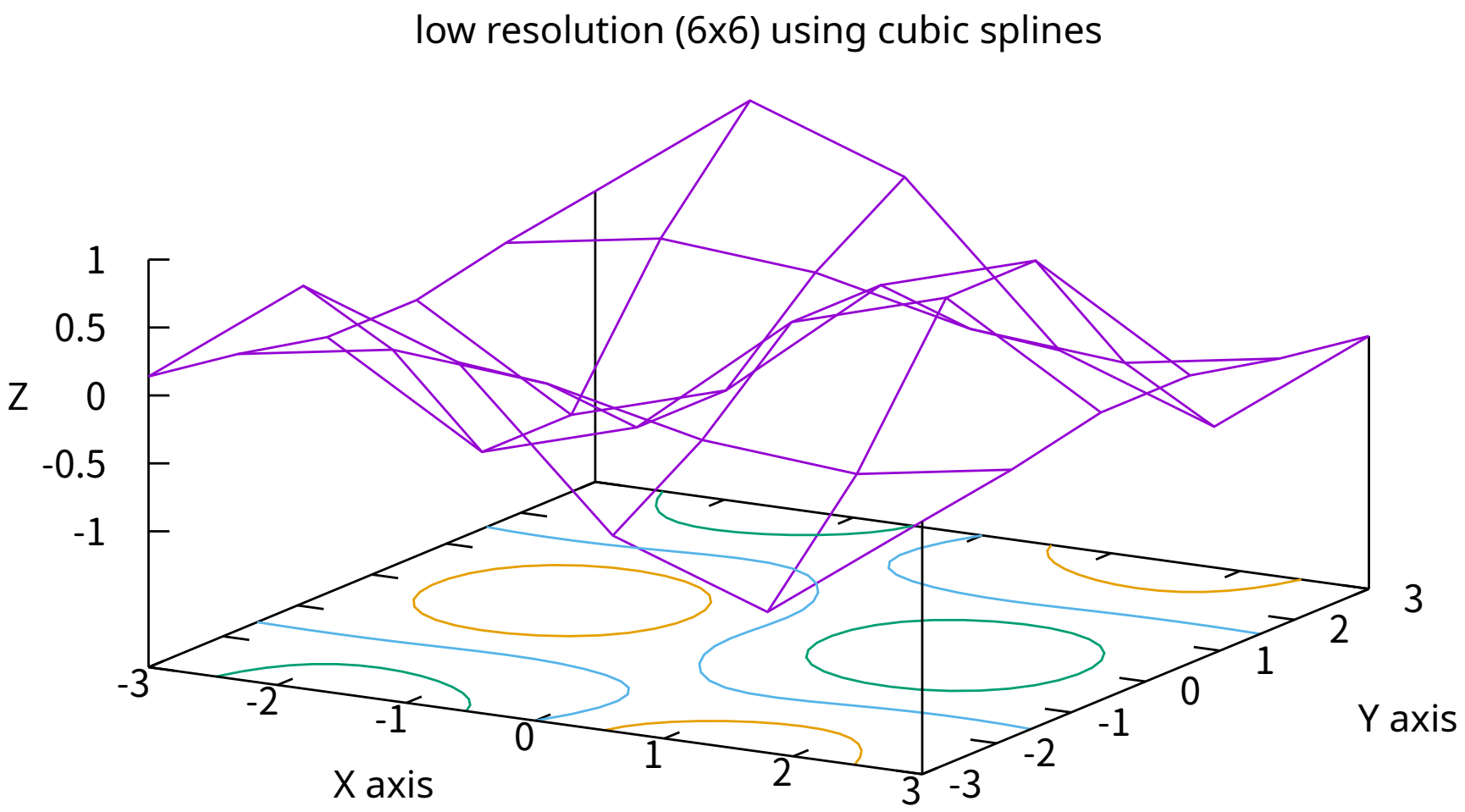

2 surfaces
x**2*y**3
1e+05
8e+04
6e+04
4e+04
2e+04
0
-2e+04
-4e+04
-6e+04
-8e+04
x**3*y**2
1e+05
8e+04
6e+04
4e+04
2e+04
100000
50000
Z 0
-50000
-100000
-10
-5
0
5
10
X axis
-10
-5
0
Y axis

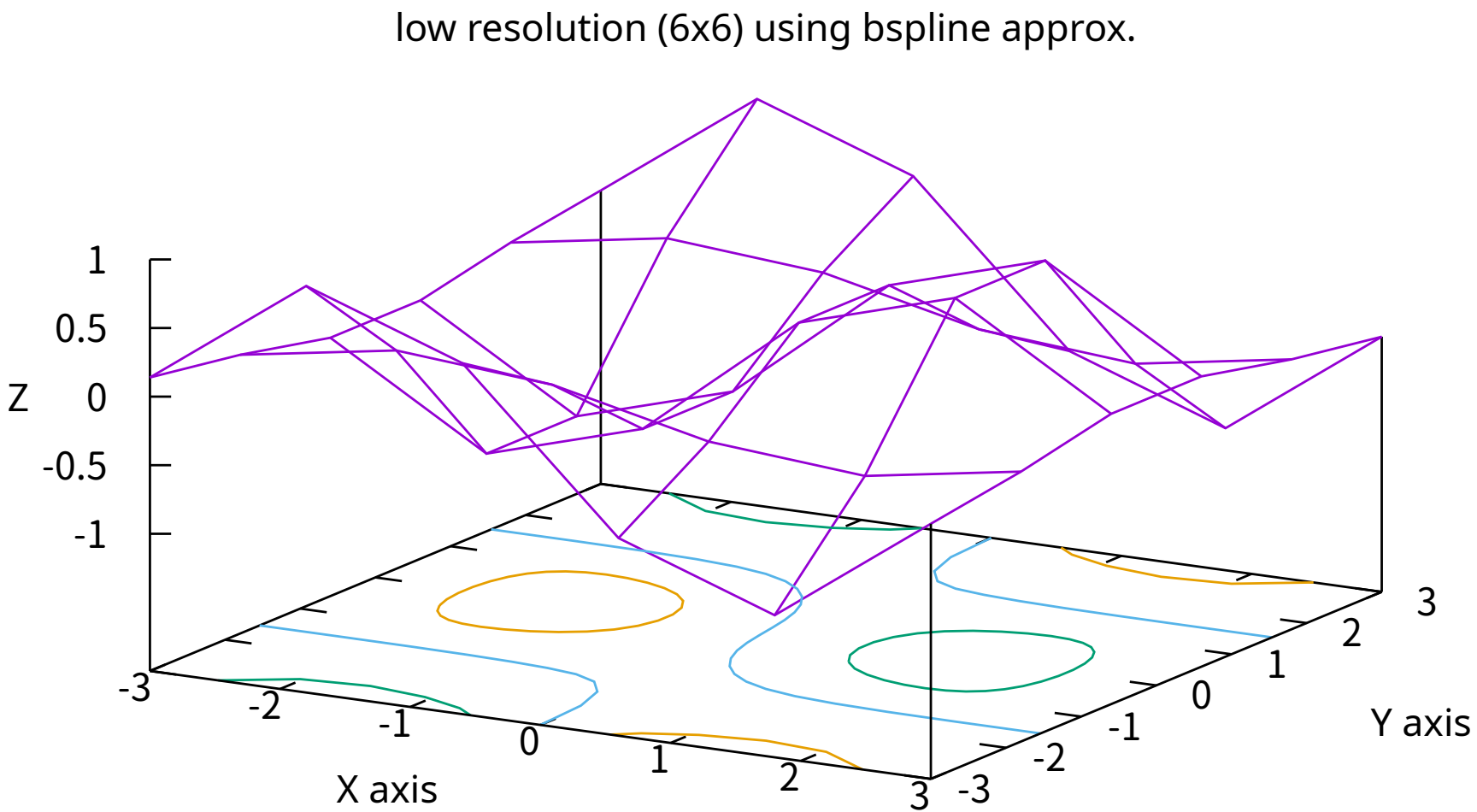
some more interesting contours



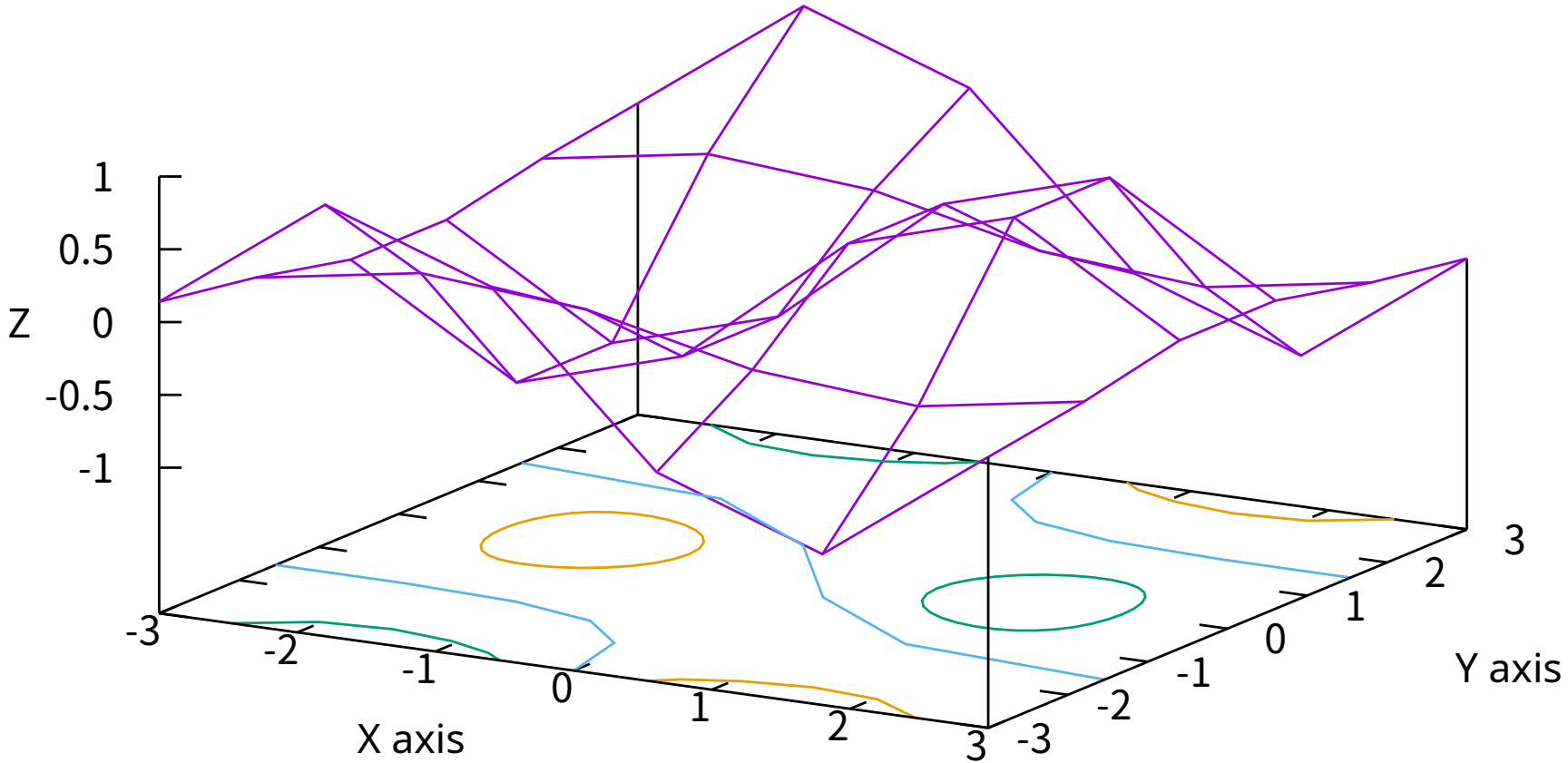

some more interesting contours
Z
1
0.5
0
-0.5
-1
X axis
-3
-2
-1
0
1
2
3
Y axis
-3
-2
-1
0
1
2
3

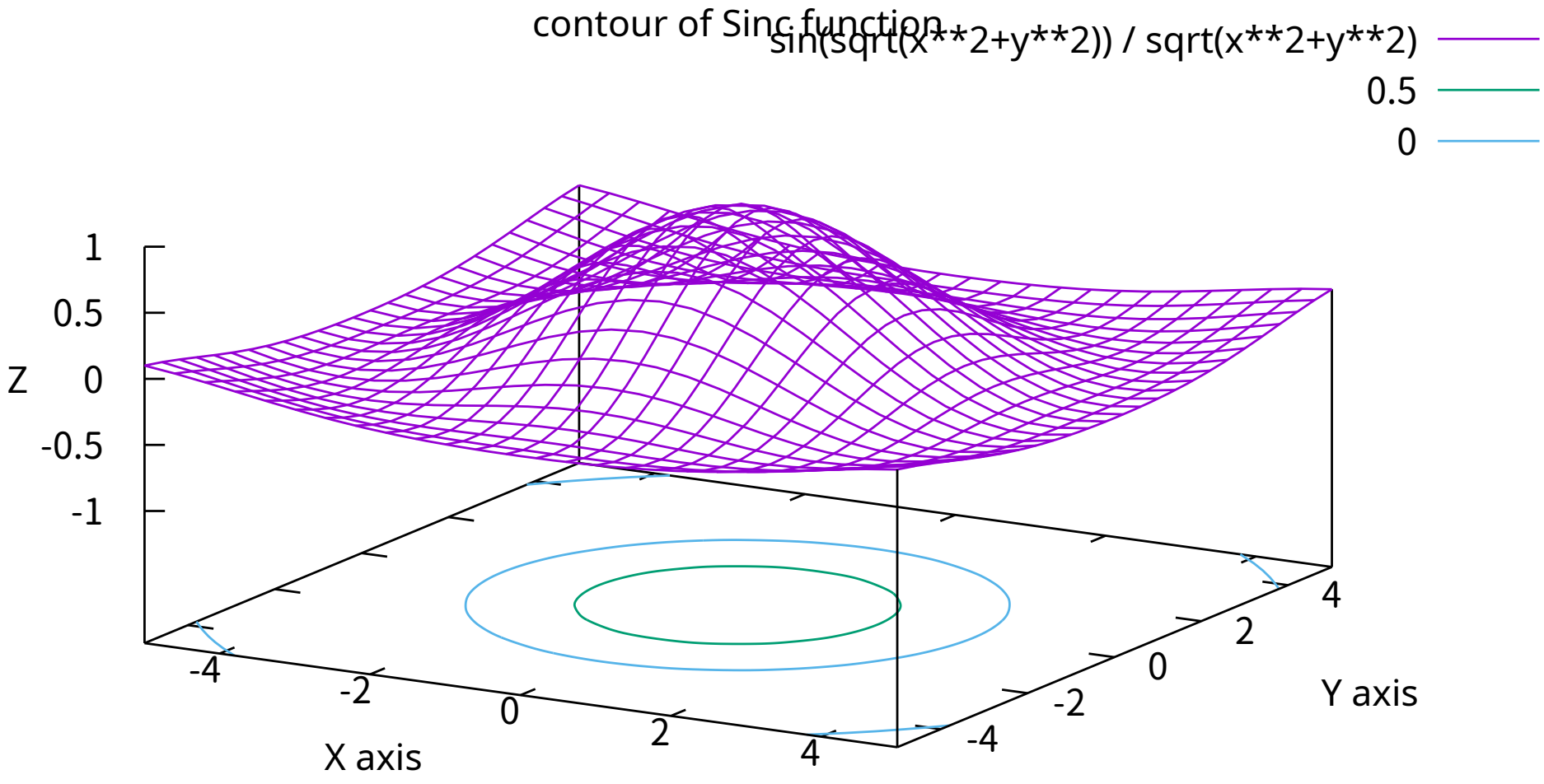

low resolution (6x6)
Z
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3
X axis
-3
-2
-1
0
1
2
3
Y axis

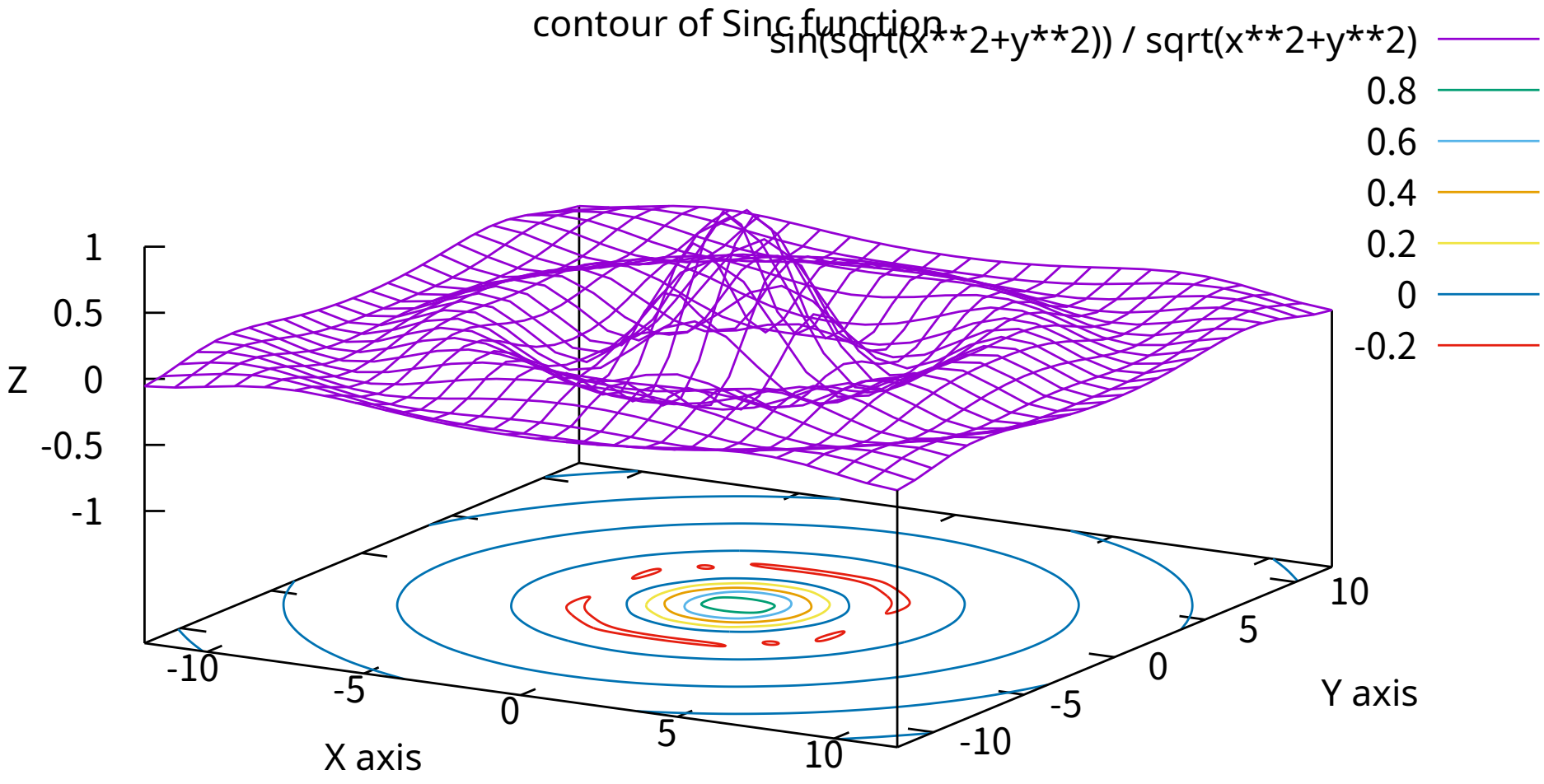

low resolution (6x6) using cubic splines
Z
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3
X axis
-3
-2
-1
0
1
2
3
Y axis

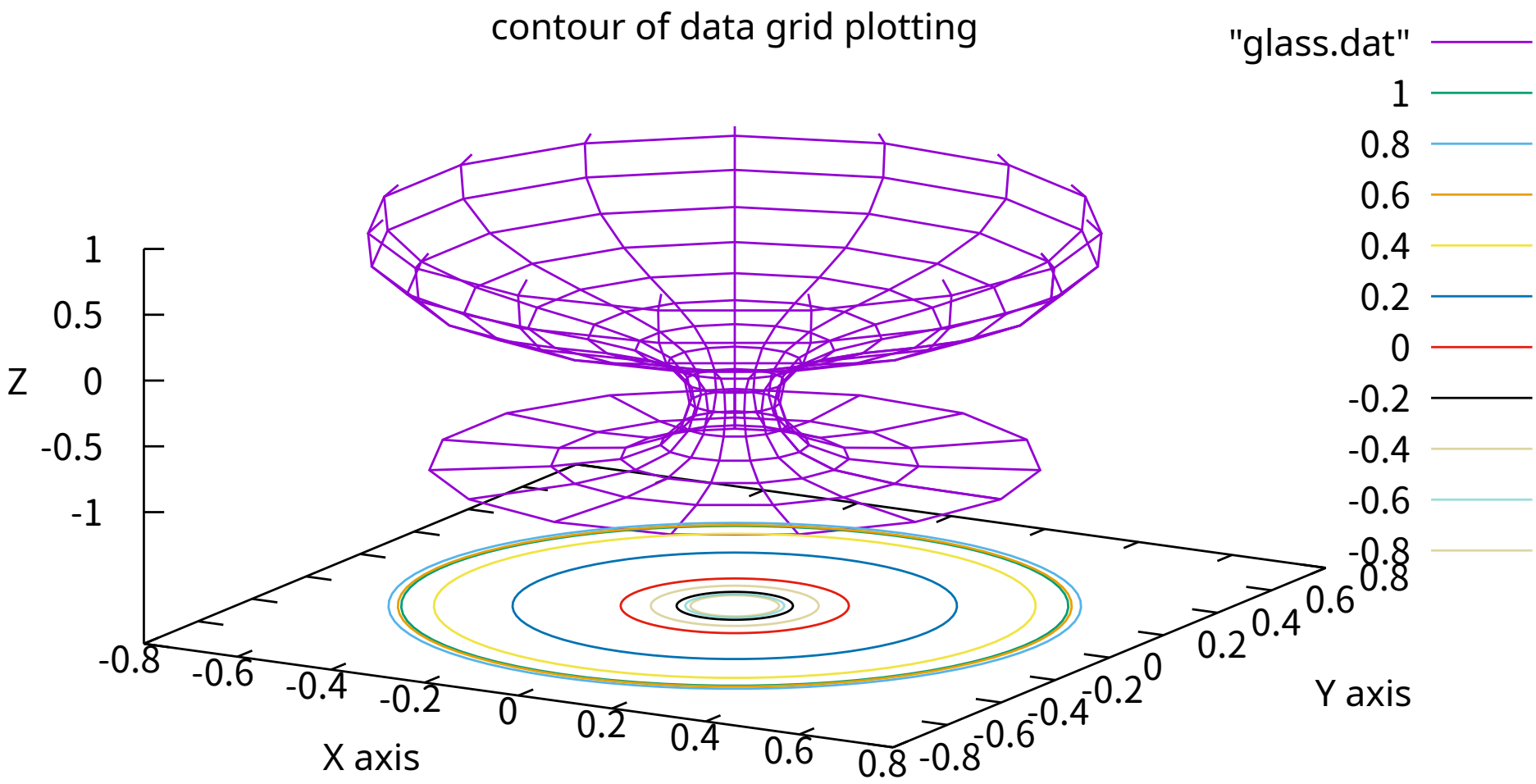

low resolution (6x6) using bspline approx.
Z
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3
X axis
-3
-2
-1
0
1
2
3
Y axis

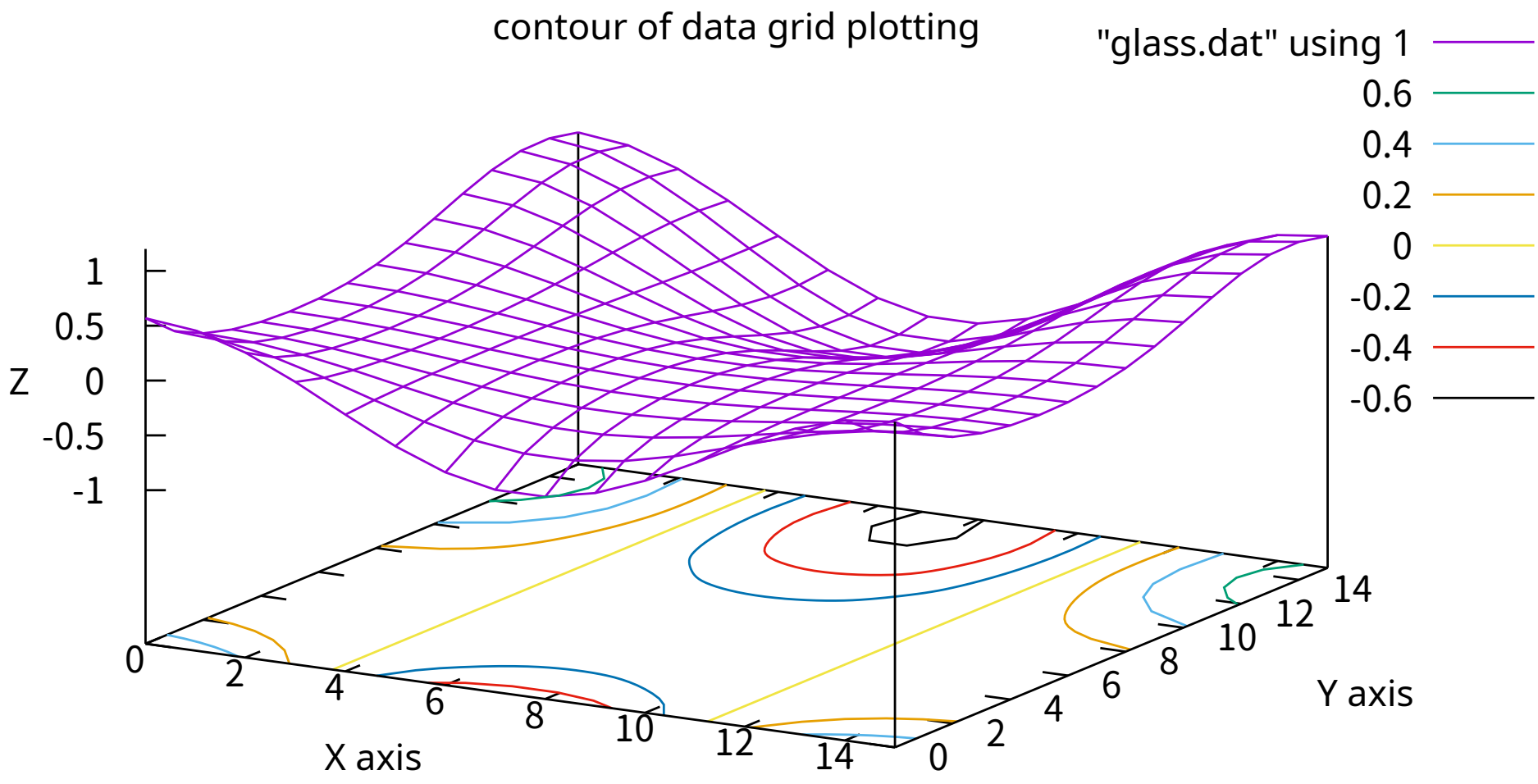
low resolution (6x6) raise bspline order.

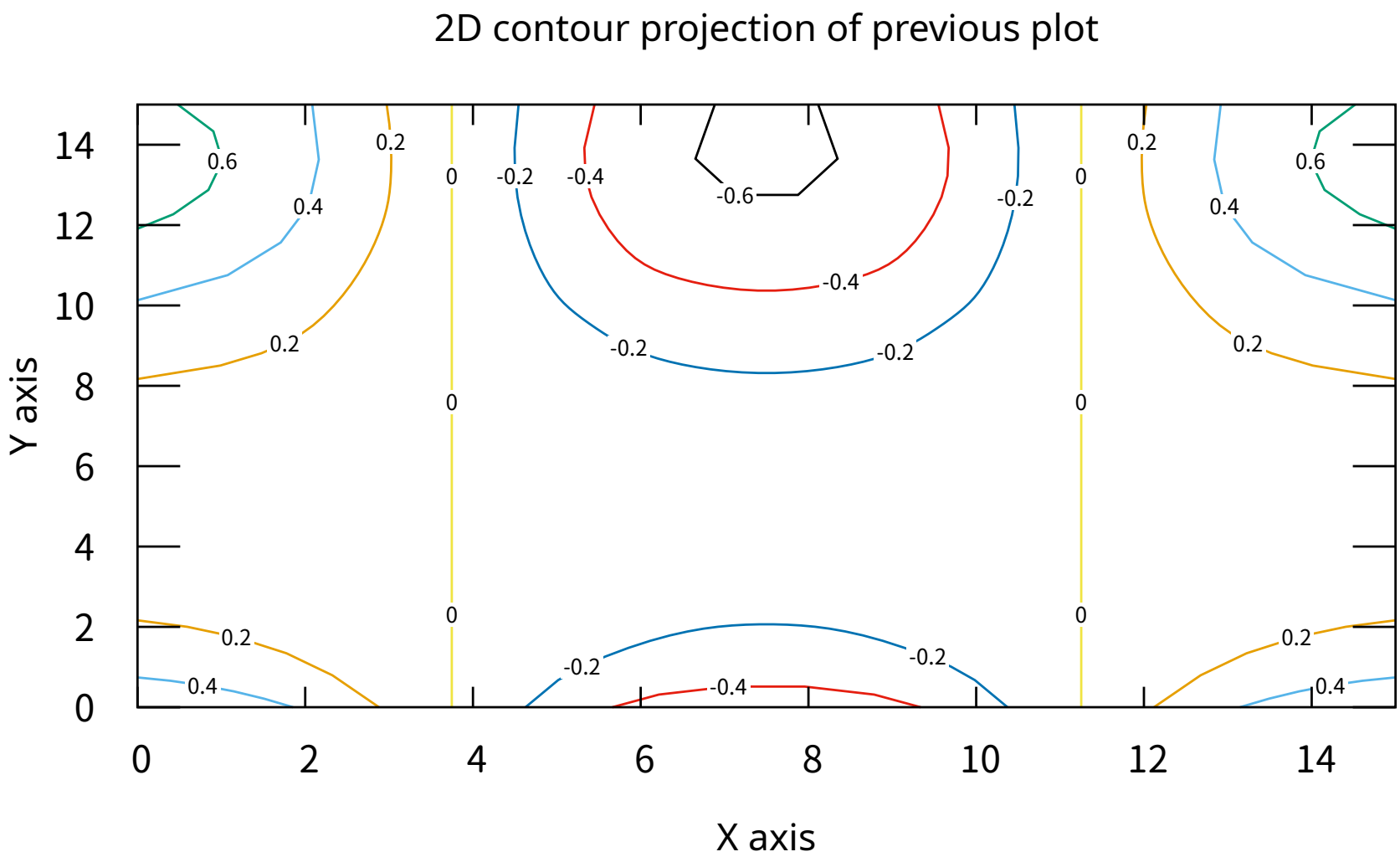



contour of Sinc function
sin(sqrt(x**2+y**2)) / sqrt(x**2+y**2)
0.5
0
1
0.5
Z
0
-0.5
-1
-4
-2
0
2
4
X axis
-4
-2
0
2
4
Y axis

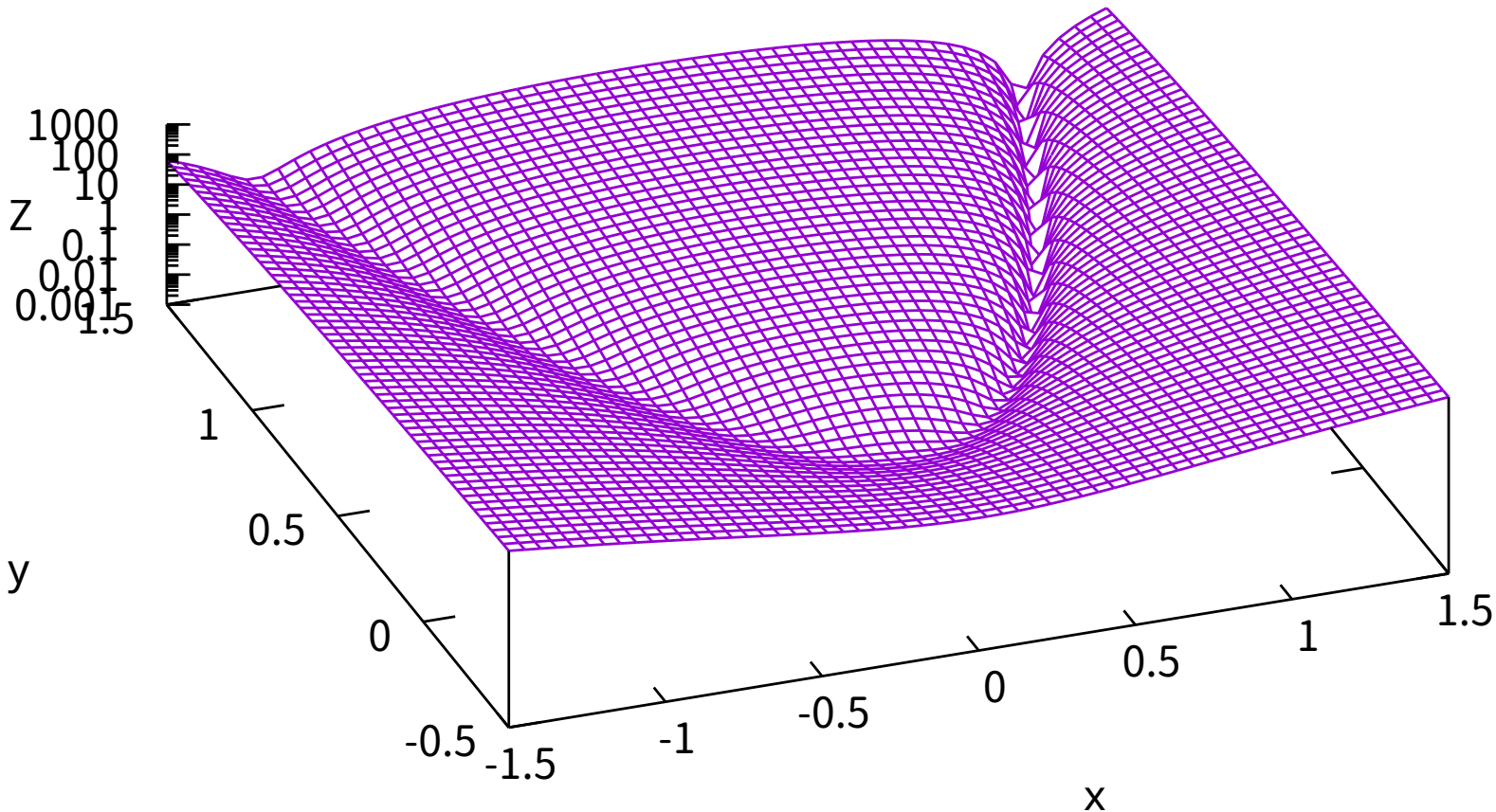

contour of Sinc function
sin(sqrt(x**2+y**2)) / sqrt(x**2+y**2)
0.8
0.6
0.4
0.2
0
-0.2
Z
1
0.5
0
-0.5
-1
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis


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

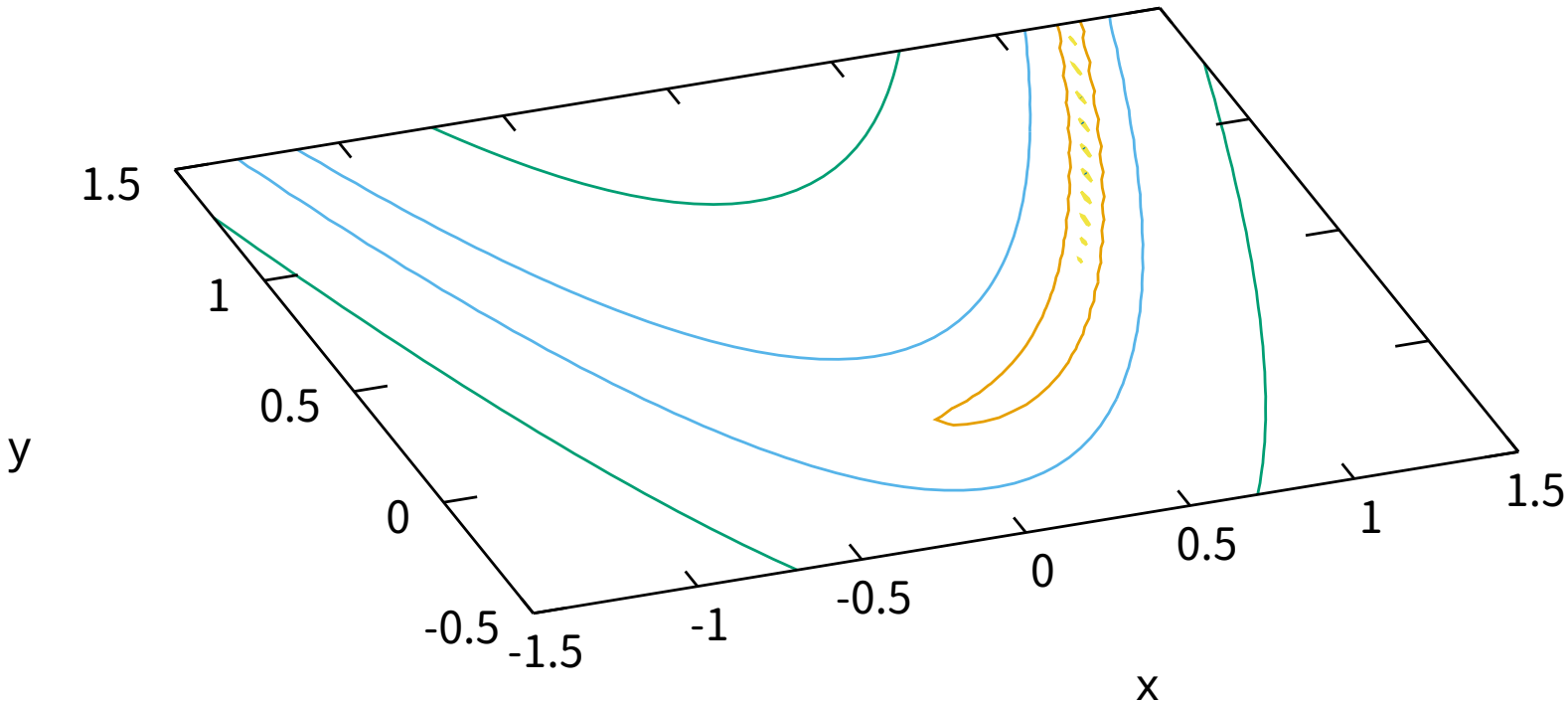

contour of data grid plotting
"glass.dat" using 1
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
Z
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
X axis
0
2
4
6
8
10
12
14
Y axis


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

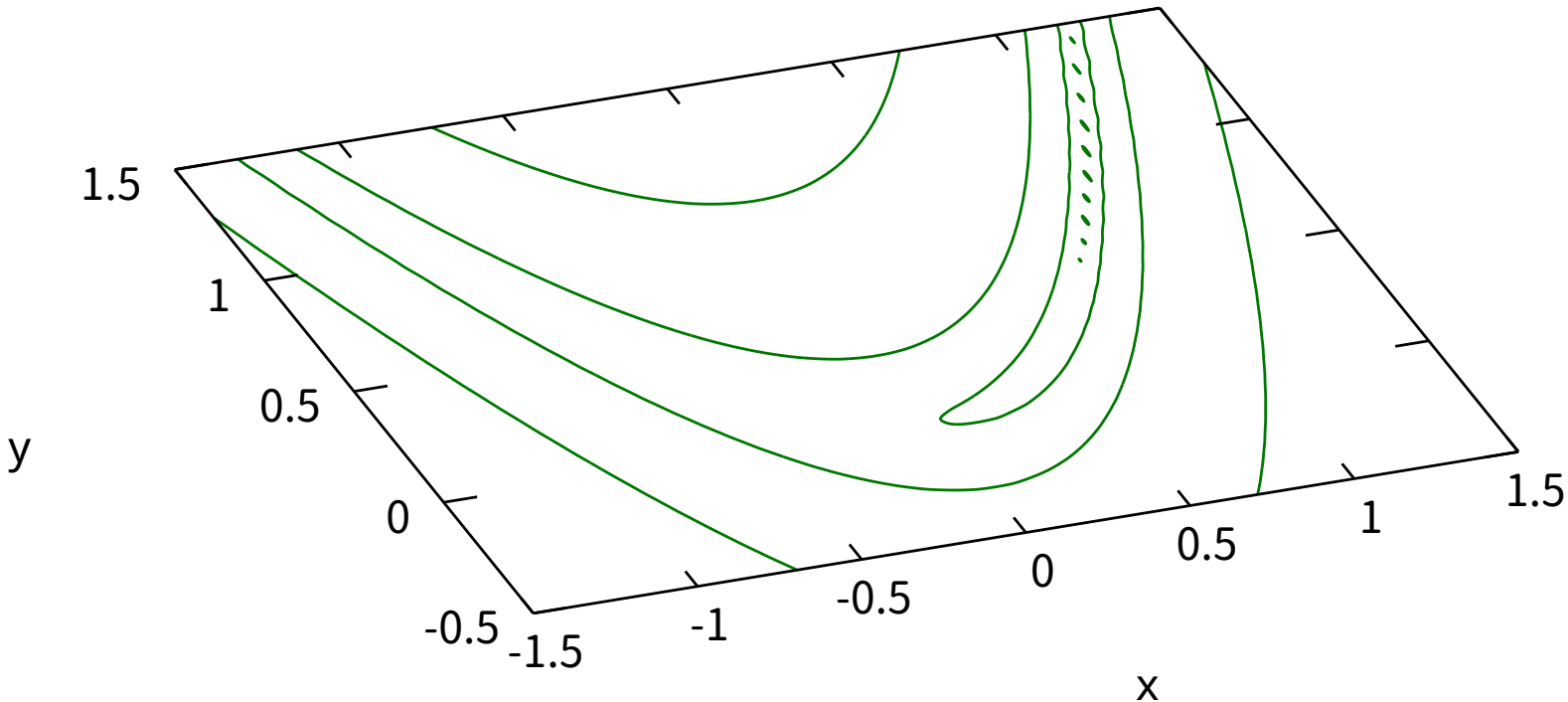
Rosenbrock Function



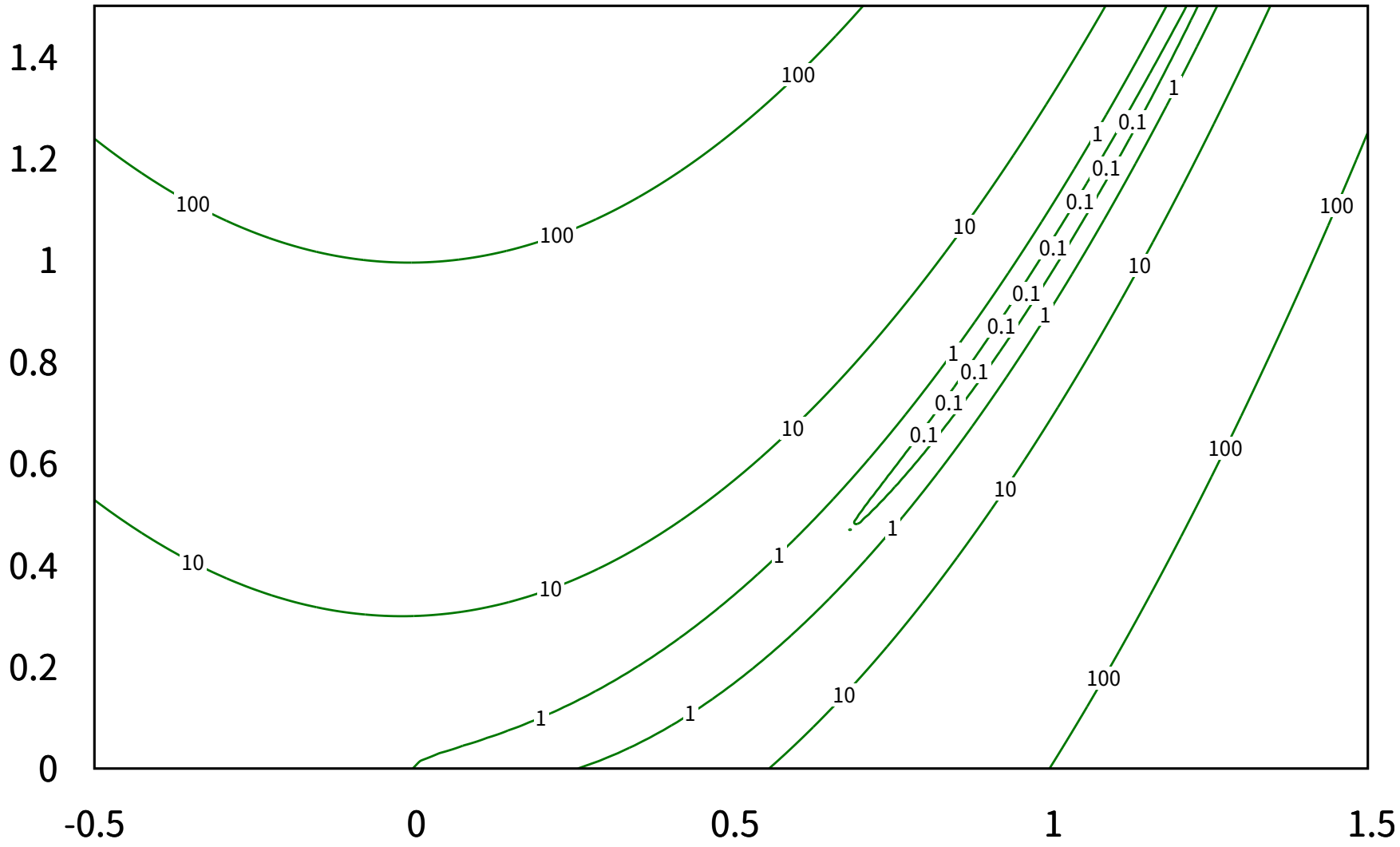
Rosenbrock Function

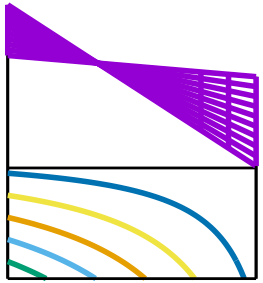


All contours drawn in a single color



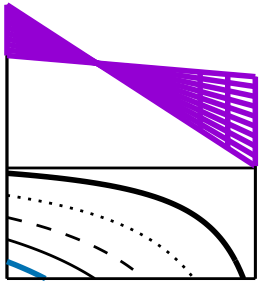
Sometimes it helps to use multiplot





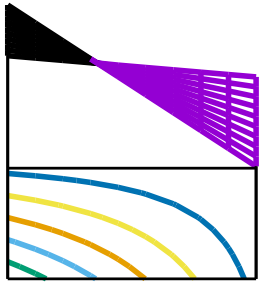
default

x*y
50
40
30
20
10



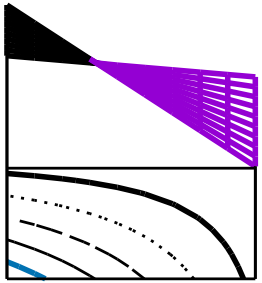
firstline sorted

x*y
10
20
30
40
50



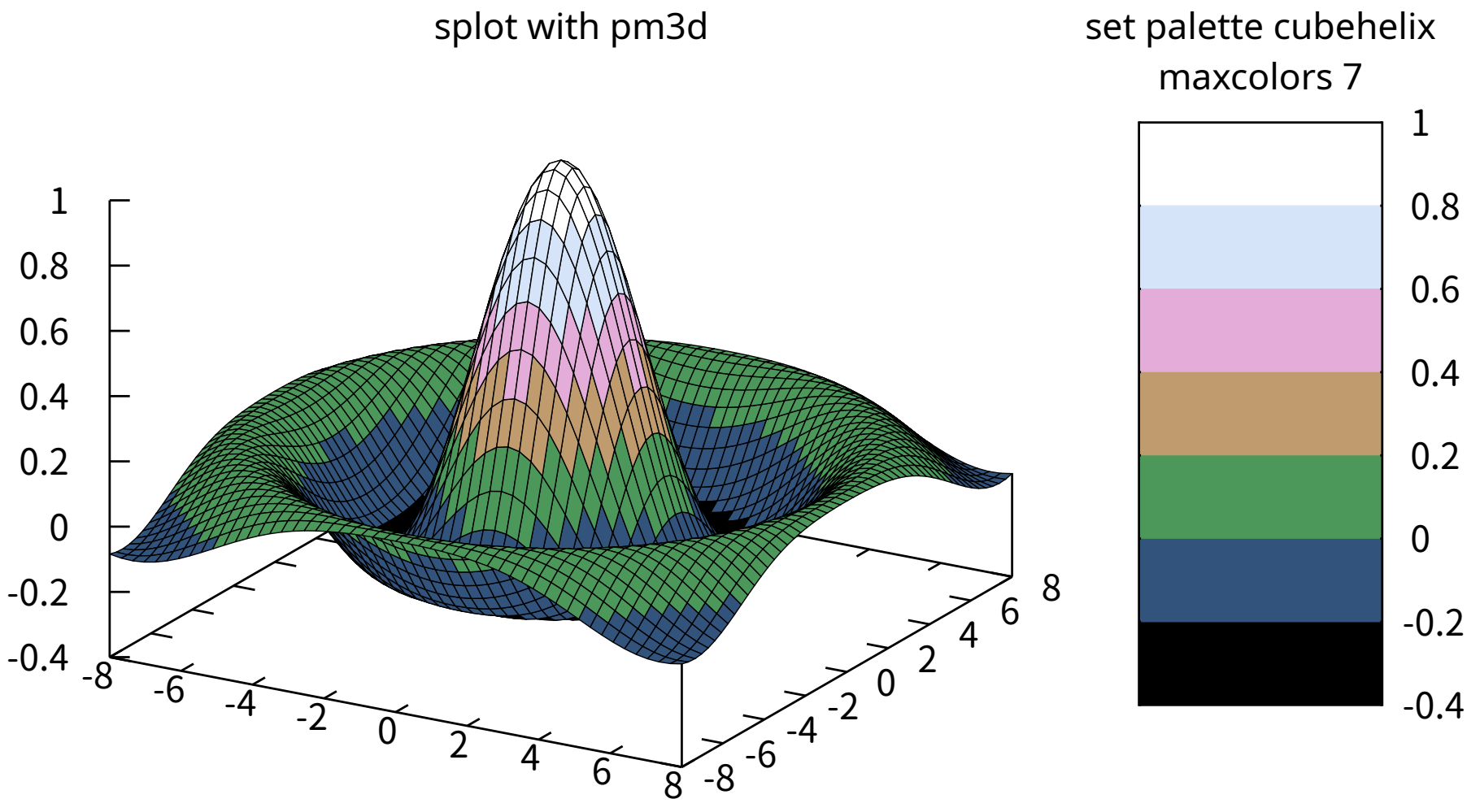
hidden3d

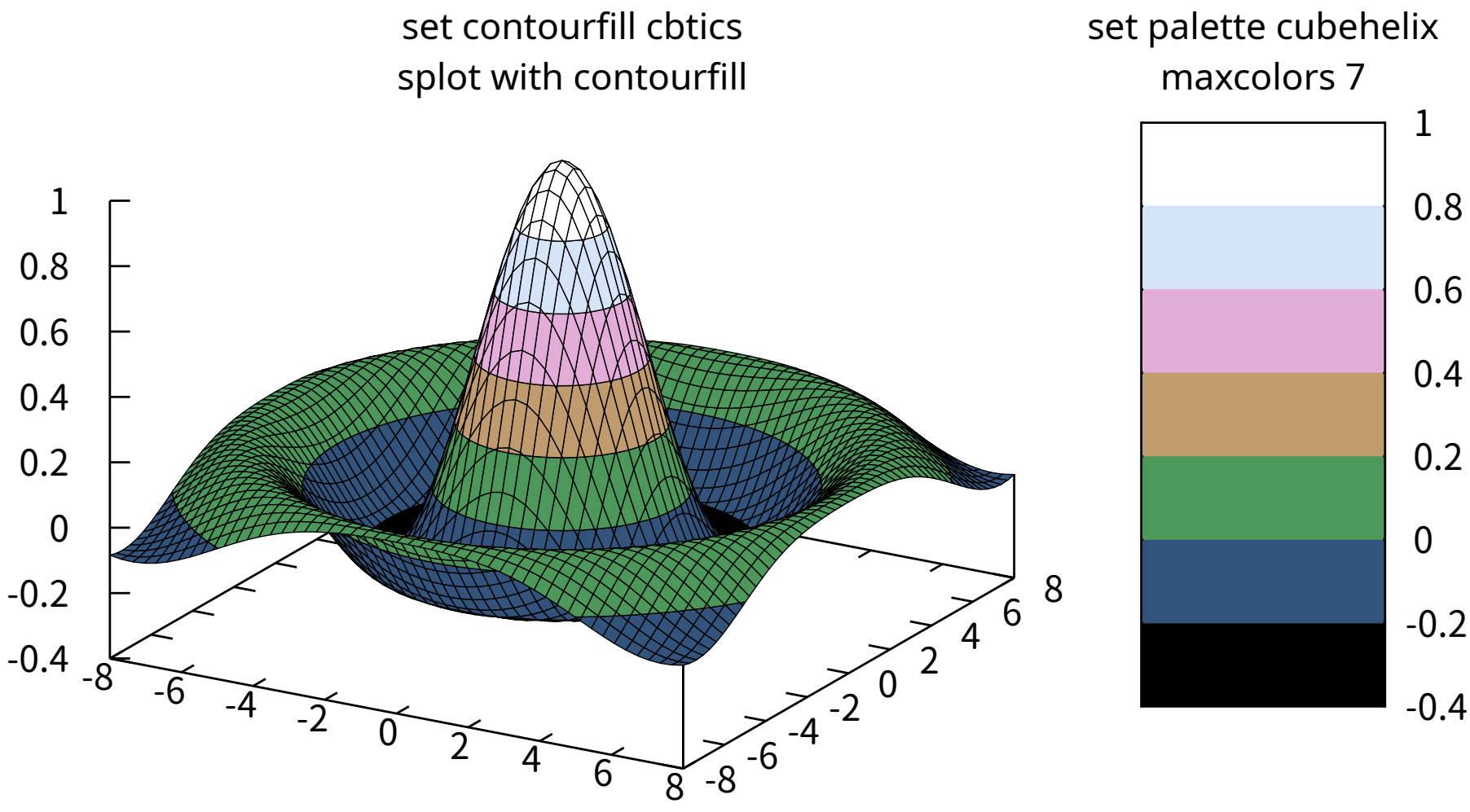
x*y
50
40
30
20
10

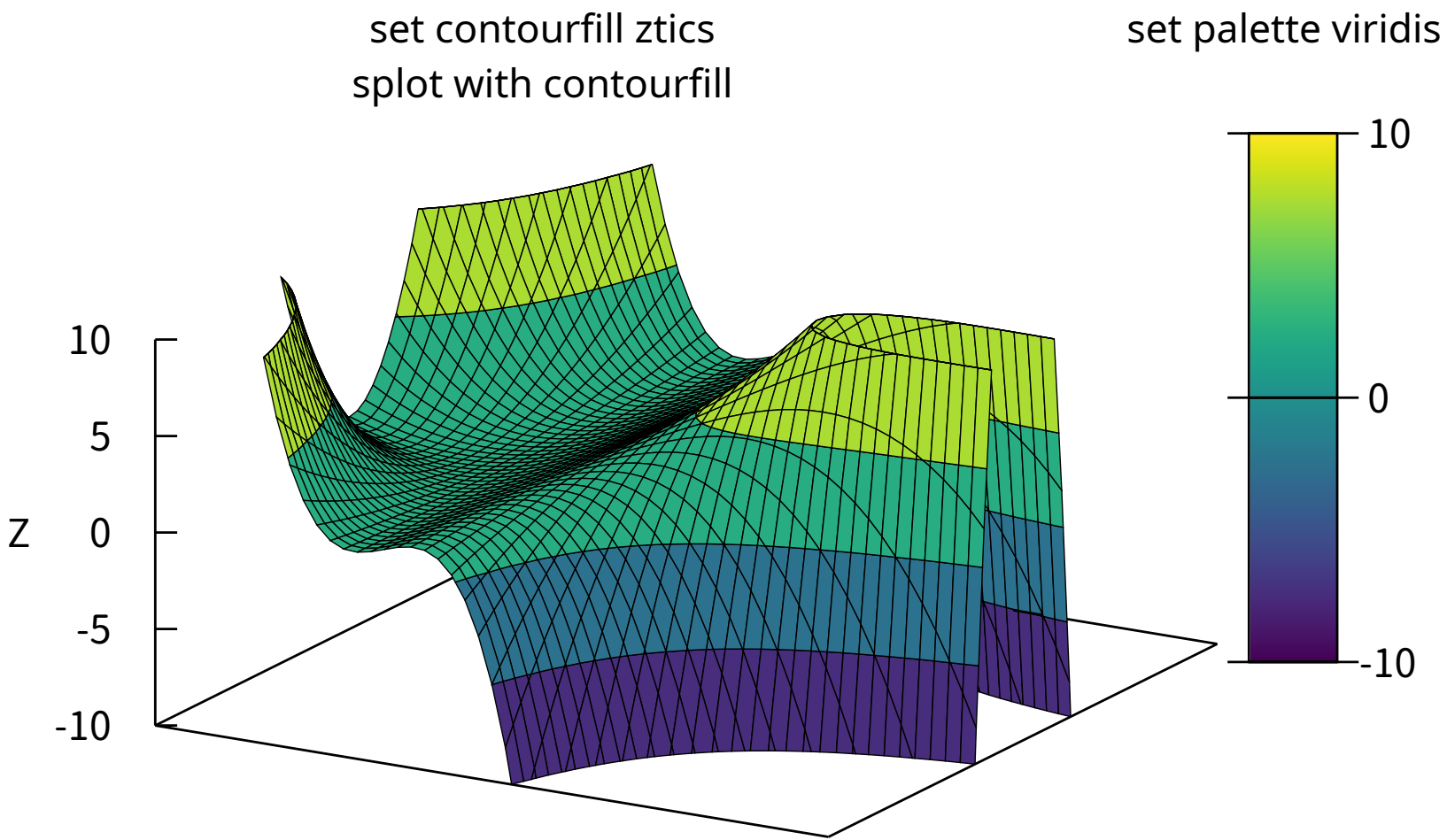


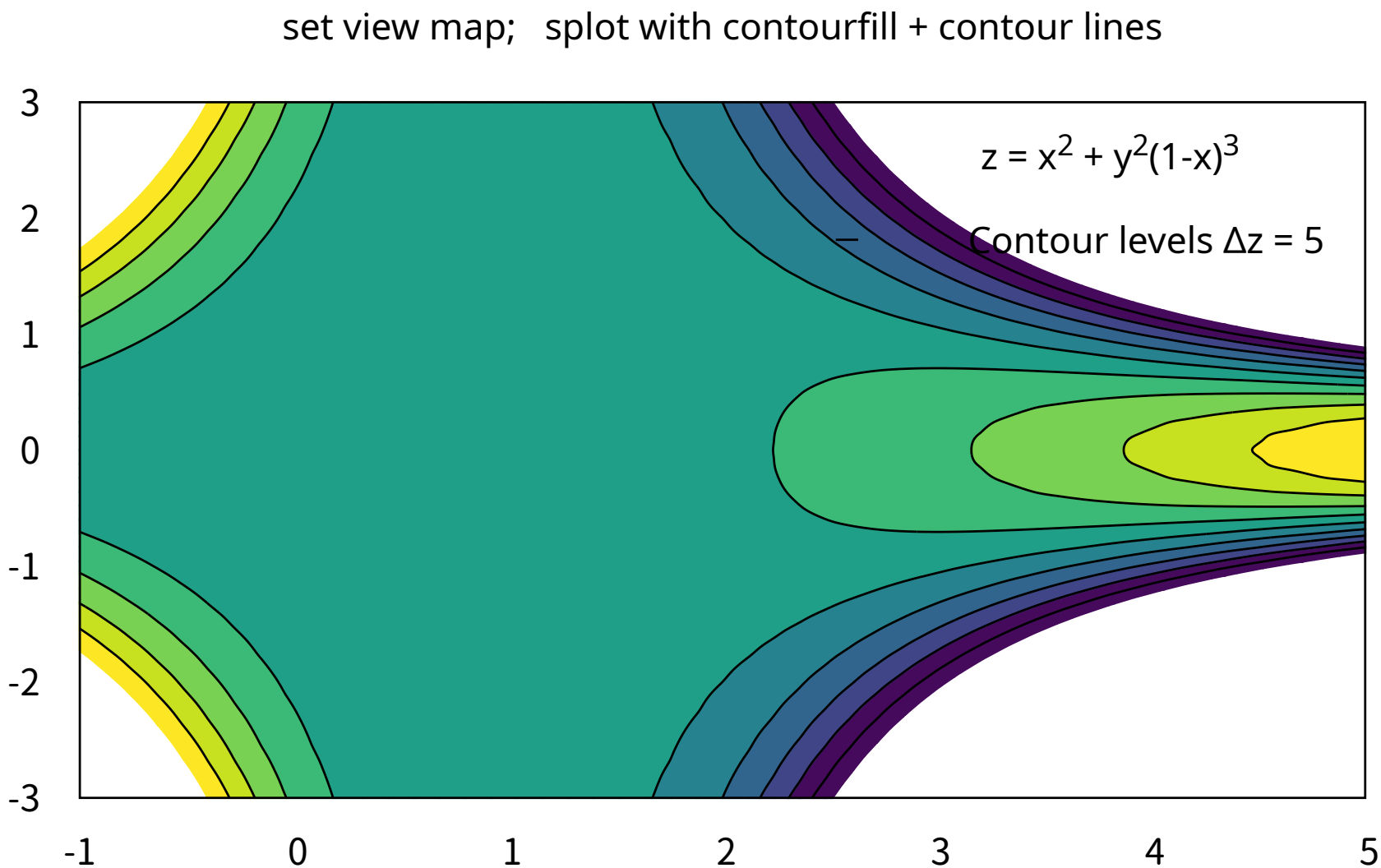
hidden3d firstline sorted

x*y
10
20
30
40
50


splot with pm3d
set palette cubehelix
maxcolors 7
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-8
-6
-4
-2
0
2
4
6
8


set contourfill cbtics
splot with contourfill
set palette cubehelix
maxcolors 7
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-8
-6
-4
-2
0
2
4
6
8

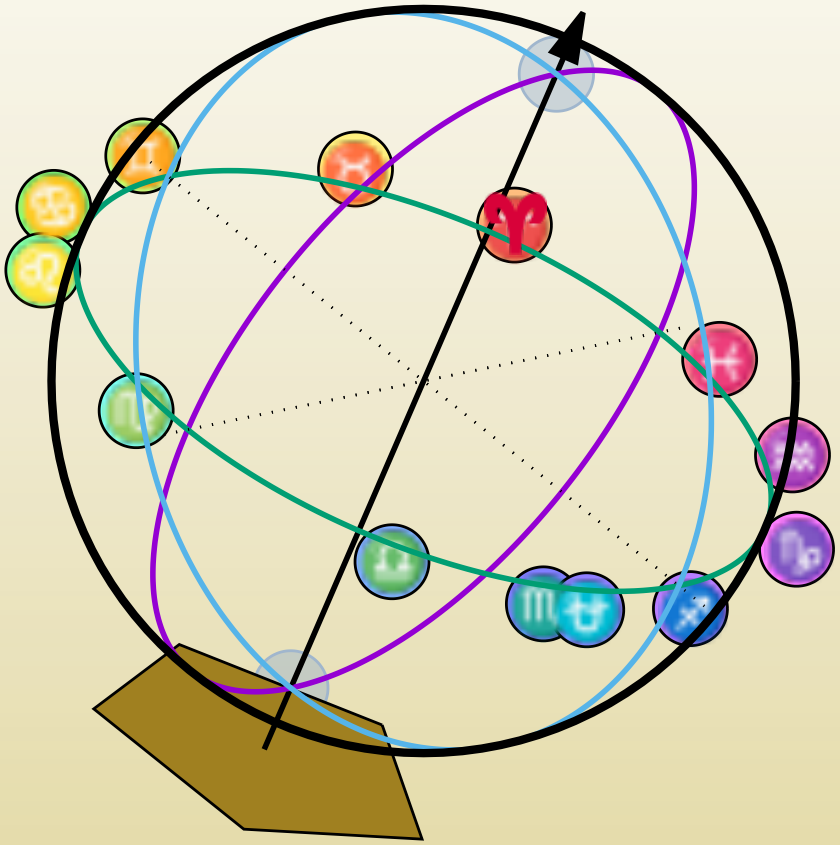

set contourfill ztics
splot with contourfill
set palette viridis
10
5
Z
0
-5
-10
10
0
-10

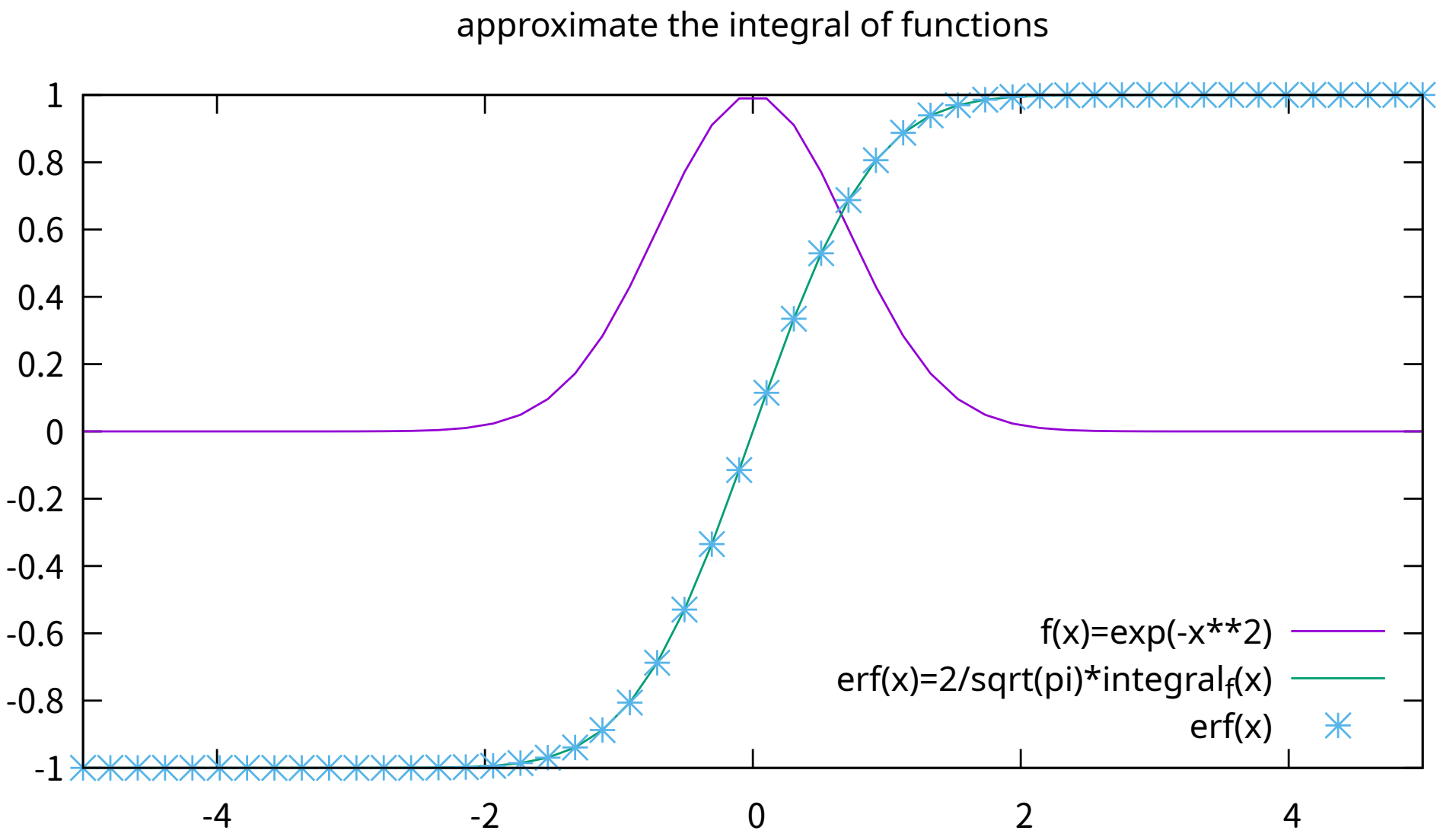

set view map; splot with contourfill + contour lines
z = x^2 + y^2(1-x)^3
Contour levels Δz = 5
3
2
1
0
-1
-2
-3
-1
0
1
2
3
4
5

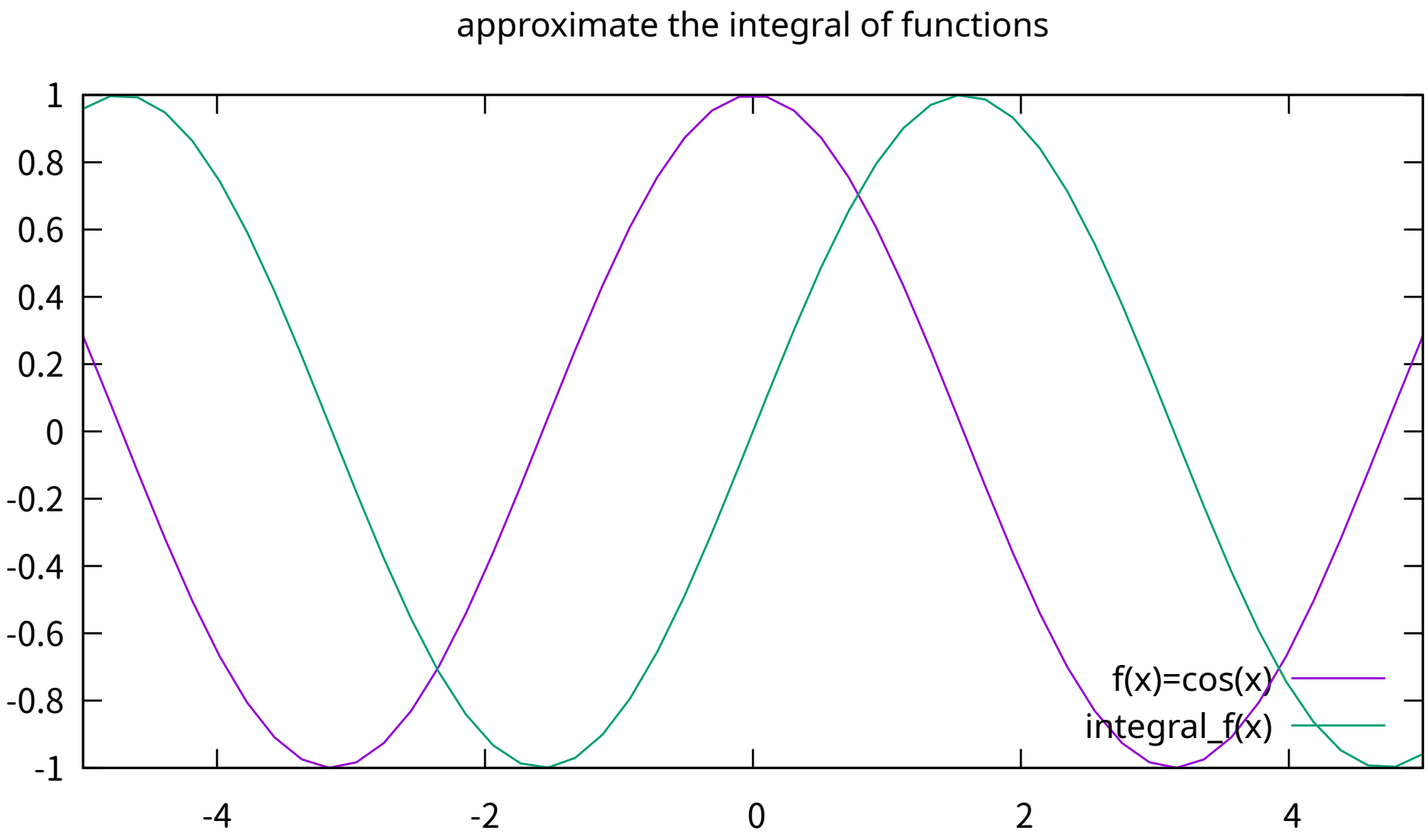
Circle and polygon objects in 3D

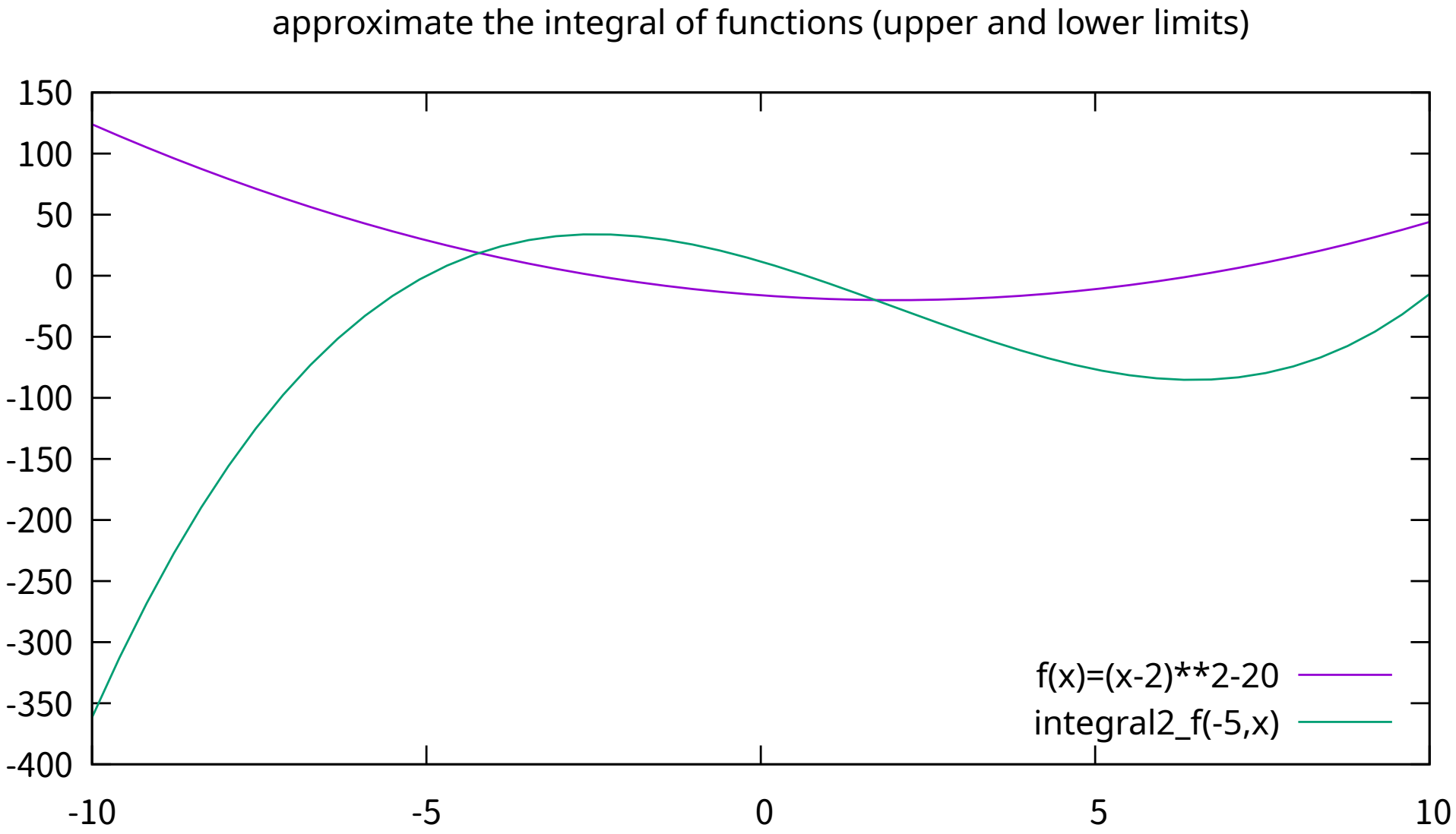
Pixmap use

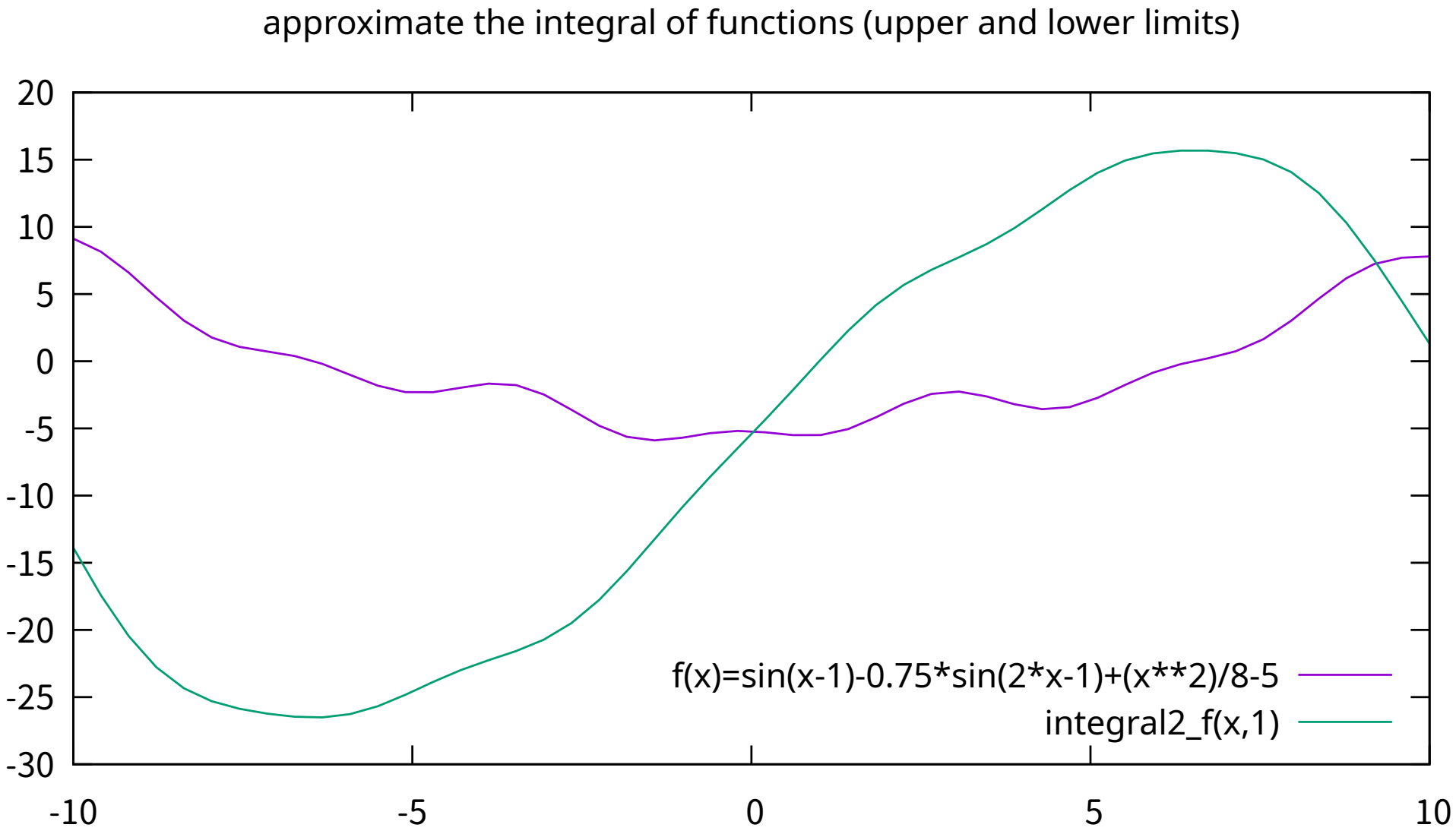
- Gradient used as a backdrop
- Project logo bottom left
- Icon (Aries) as plot element



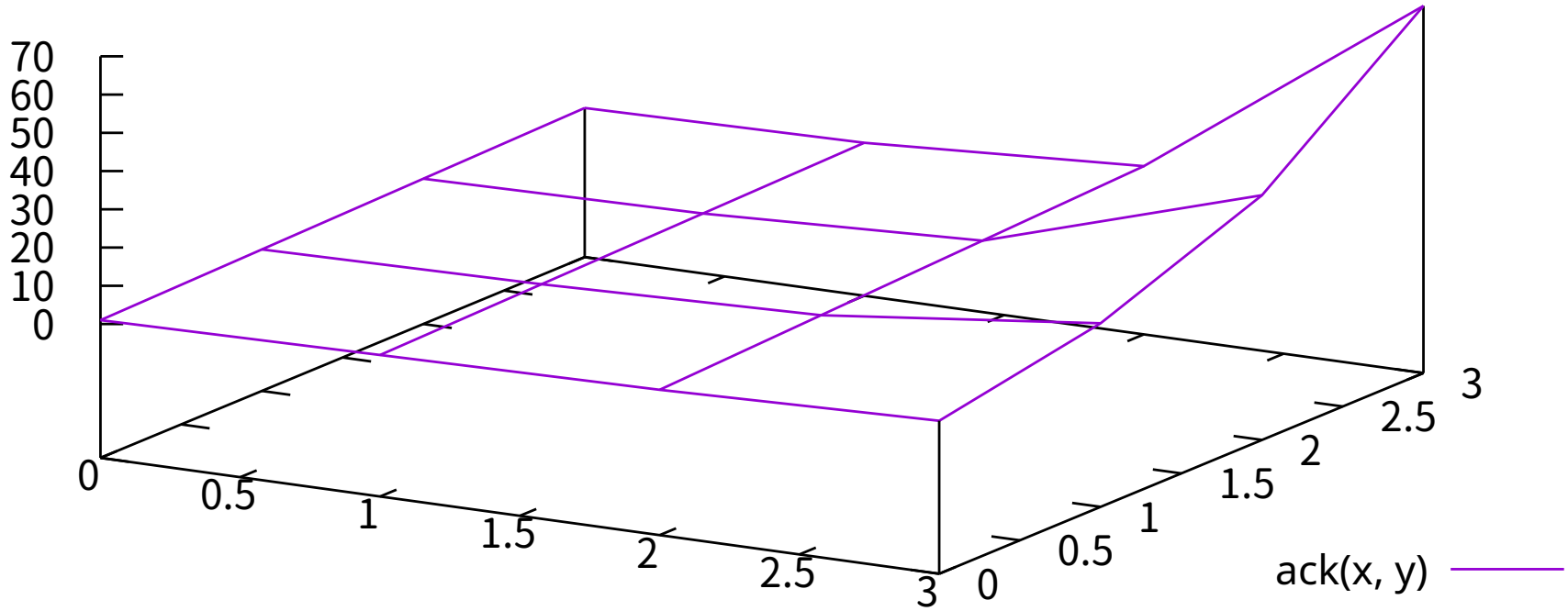

approximate the integral of functions
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-4
-2
0
2
4
f(x)=exp(-x**2)
erf(x)=2/sqrt(pi)*integral_f(x)
erf(x)

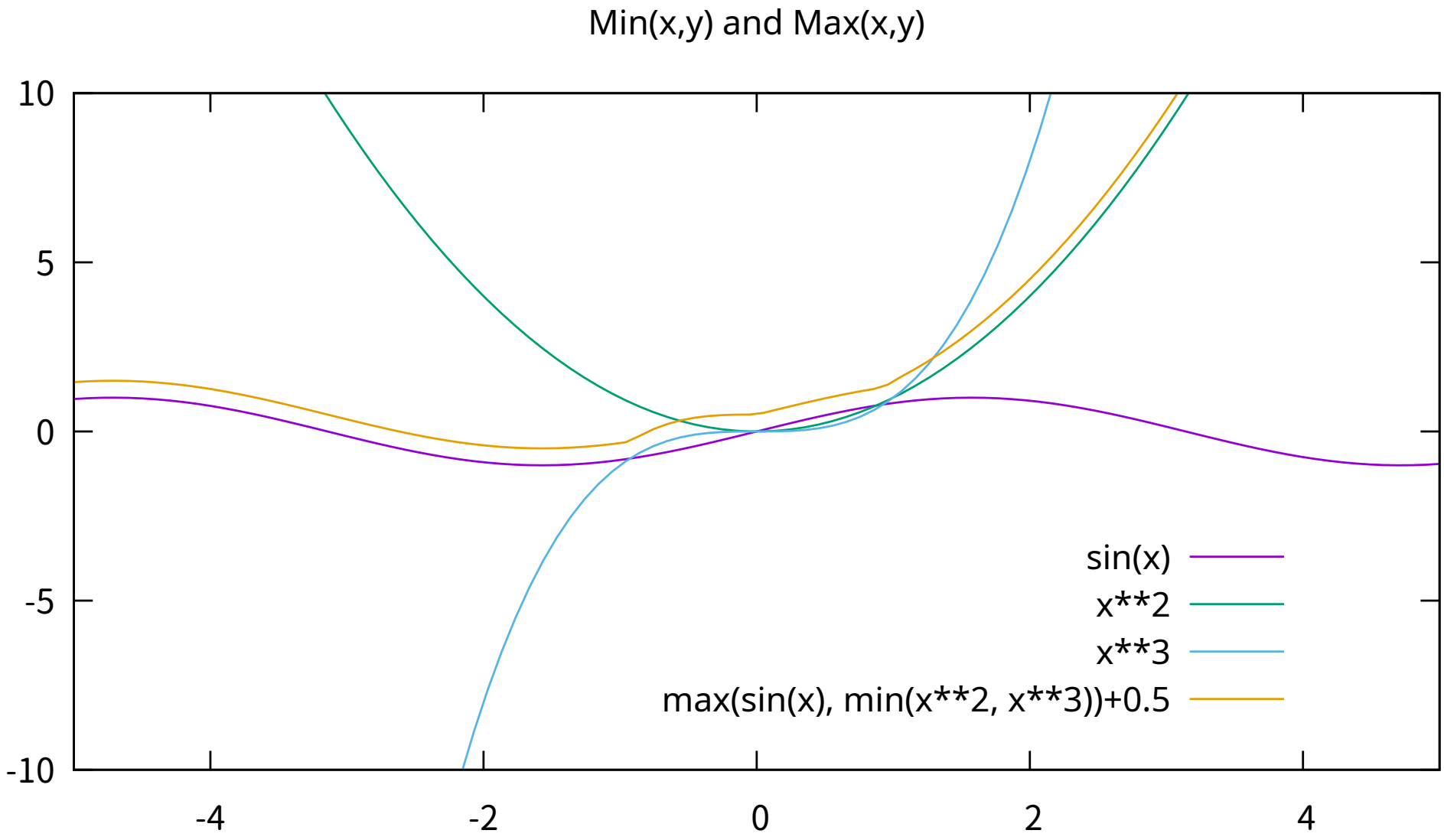

approximate the integral of functions
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-4
-2
0
2
4
f(x)=cos(x)
integral_f(x)

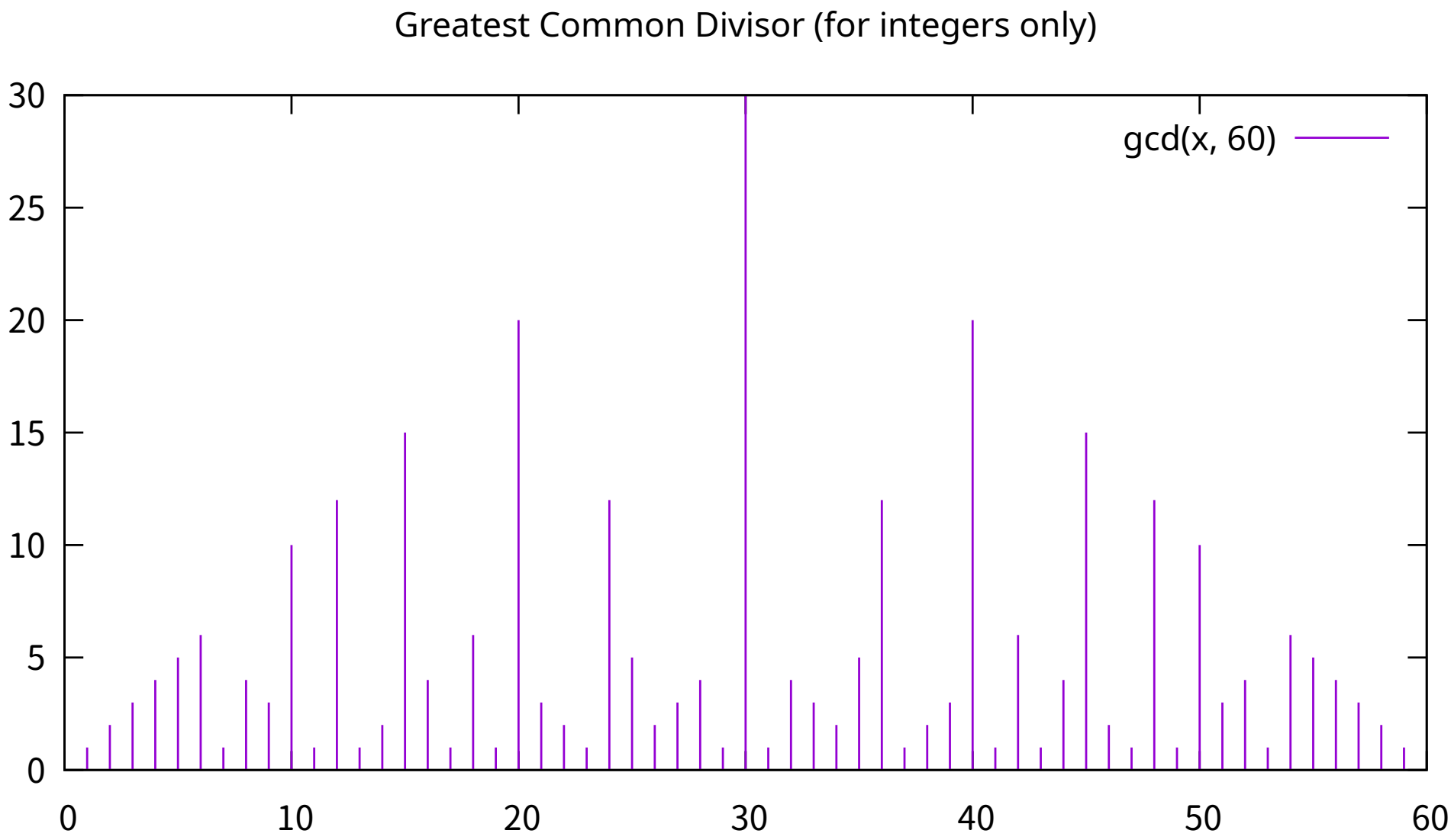

approximate the integral of functions (upper and lower limits)
150
100
50
0
-50
-100
-150
-200
-250
-300
-350
-400
-10
-5
0
5
10
f(x)=(x-2)**2-20
integral2_f(-5,x)

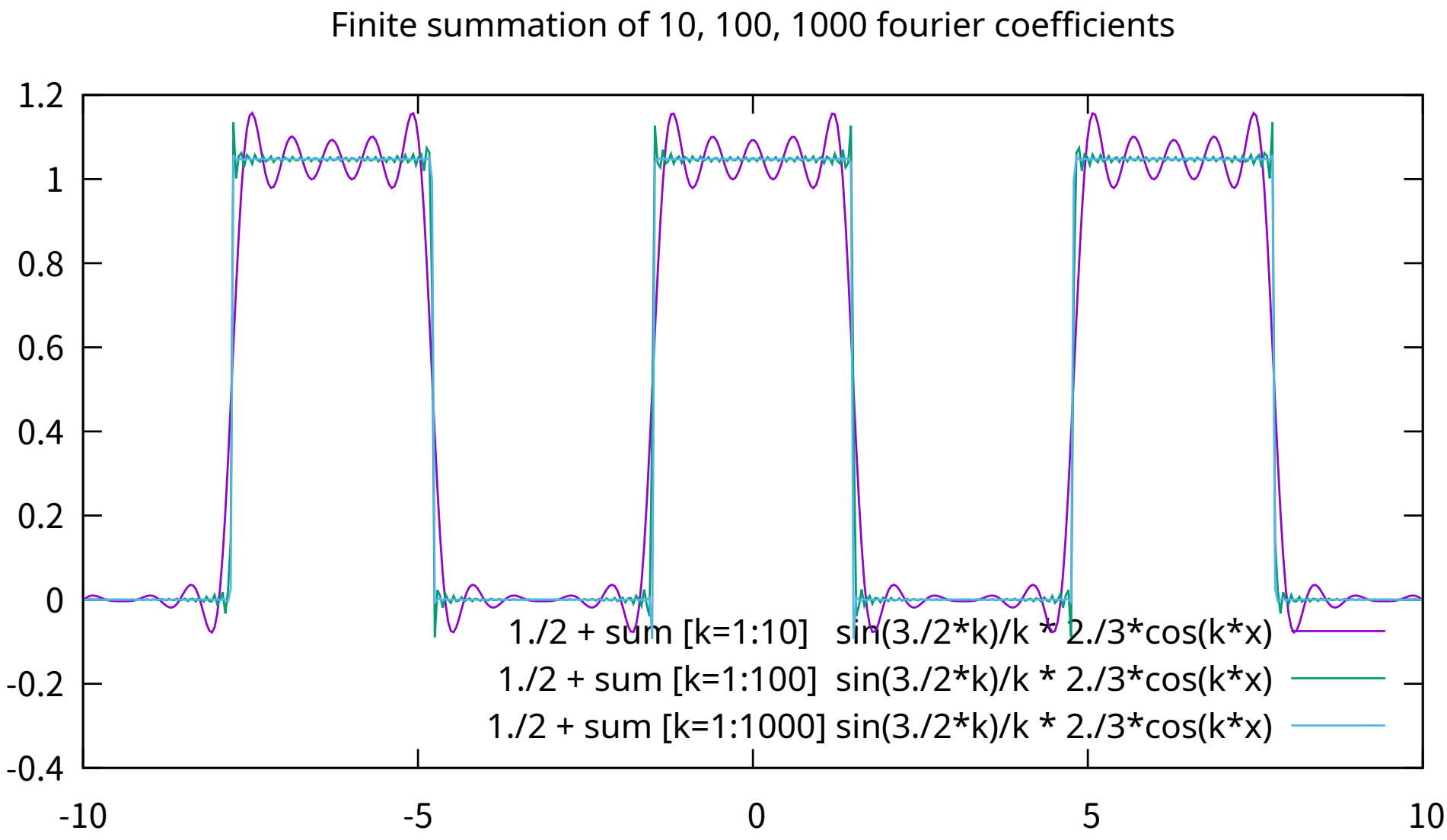

approximate the integral of functions (upper and lower limits)
20
15
10
5
0
-5
-10
-15
-20
-25
-30
-10
-5
0
5
10
f(x)=sin(x-1)-0.75*sin(2*x-1)+(x**2)/8-5
integral2_f(x,1)

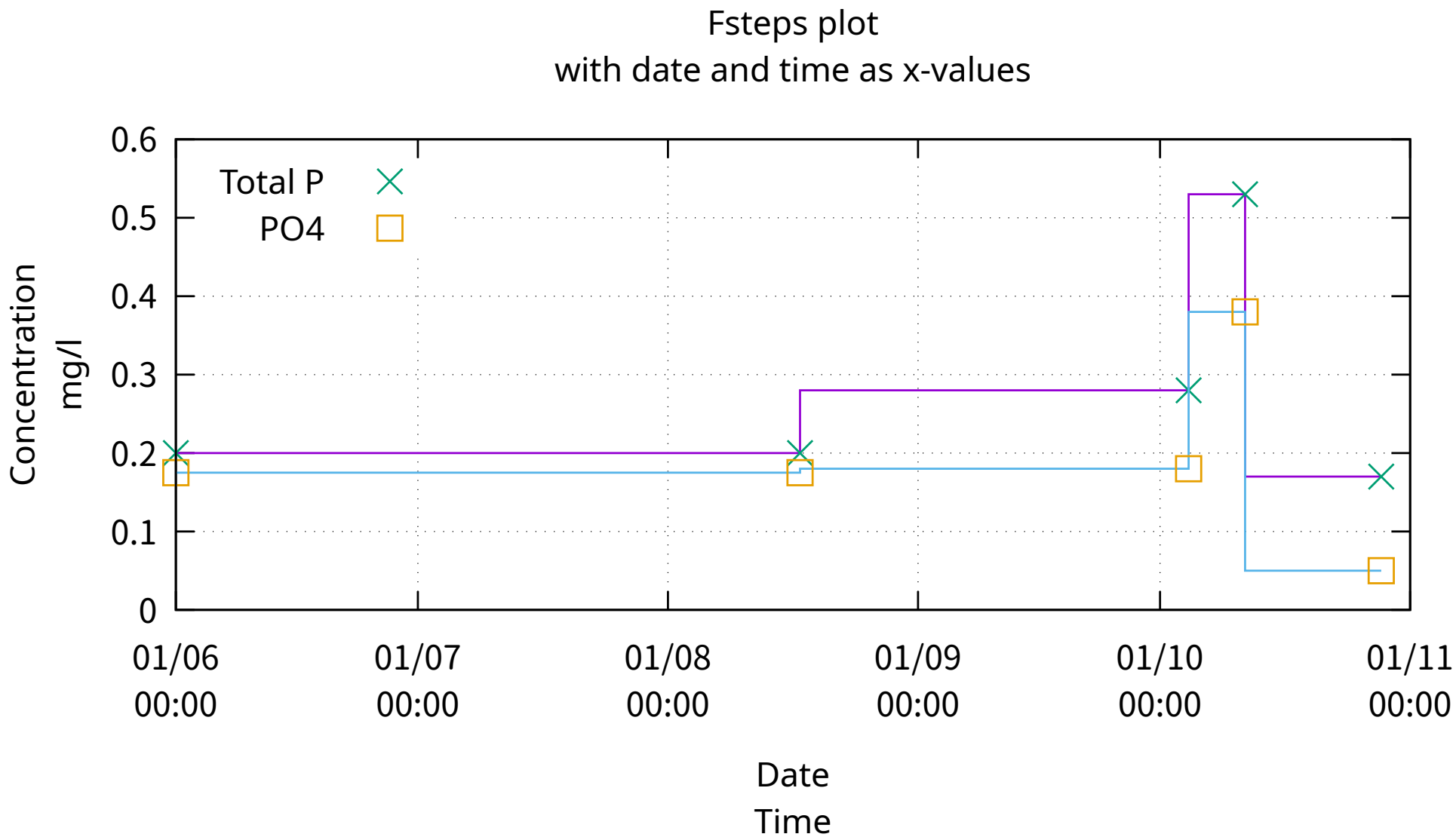
Plot of the ackermann function

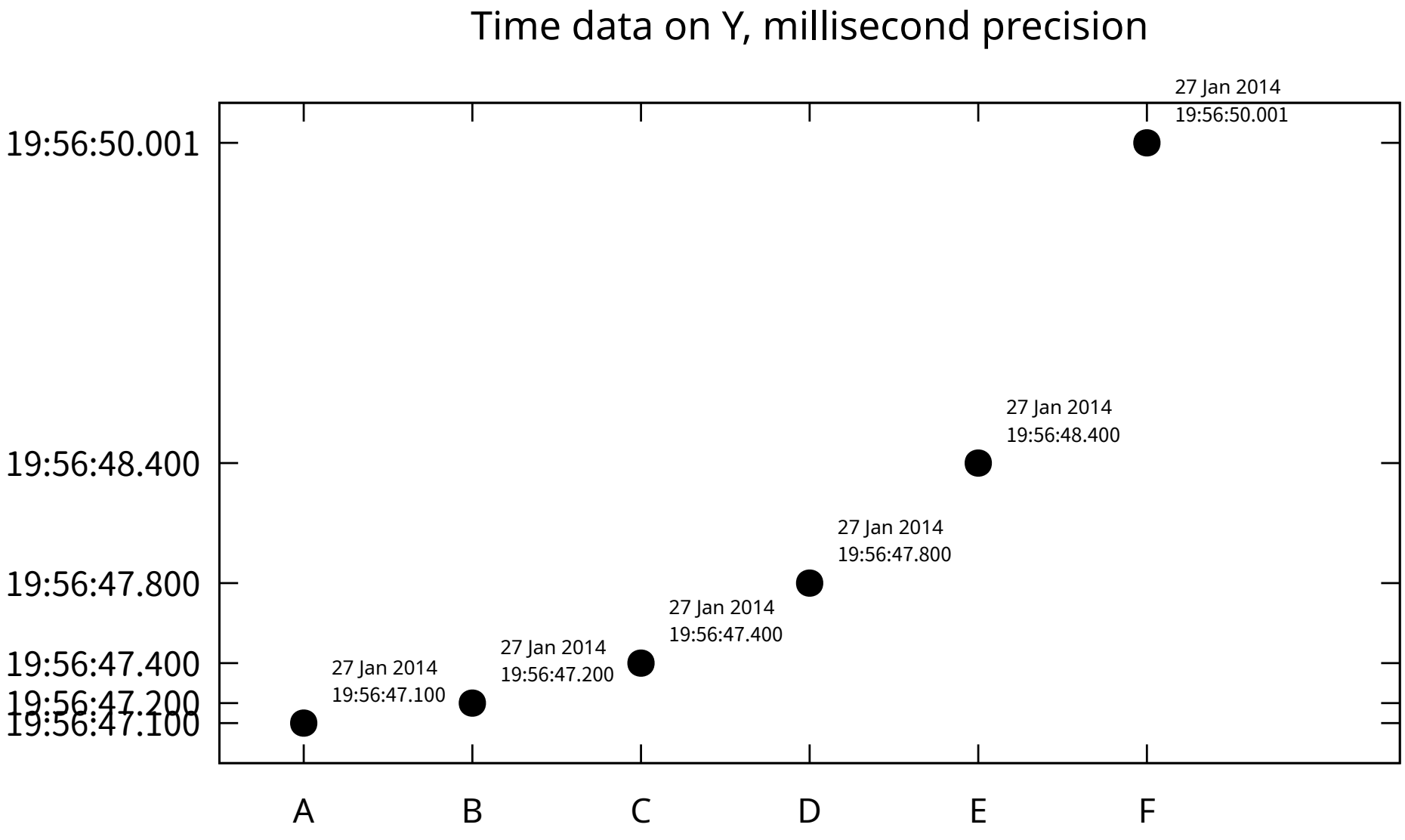



Min(x,y) and Max(x,y)
10
5
0
-5
-10
-4
-2
0
2
4
sin(x)
x**2
x**3
max(sin(x), min(x**2, x**3))+0.5

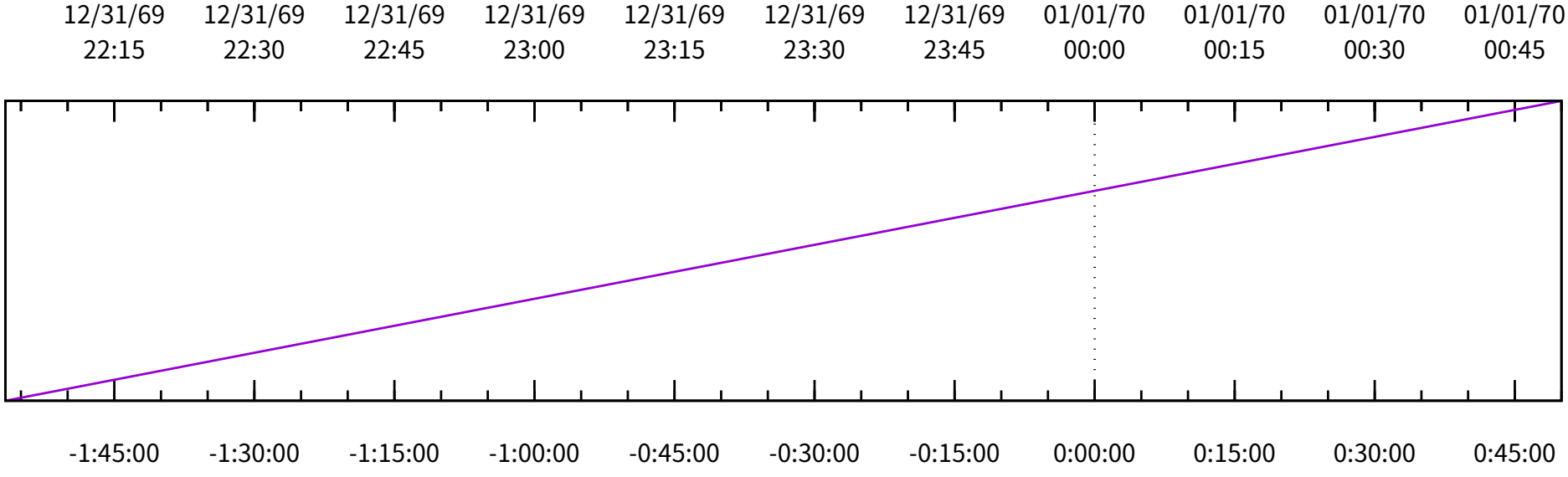

Greatest Common Divisor (for integers only)
gcd(x, 60)
0
5
10
15
20
25
30
0
10
20
30
40
50
60


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
00:00
01/07
00:00
01/08
00:00
01/09
00:00
01/10
00:00
01/11
00:00
Date
Time

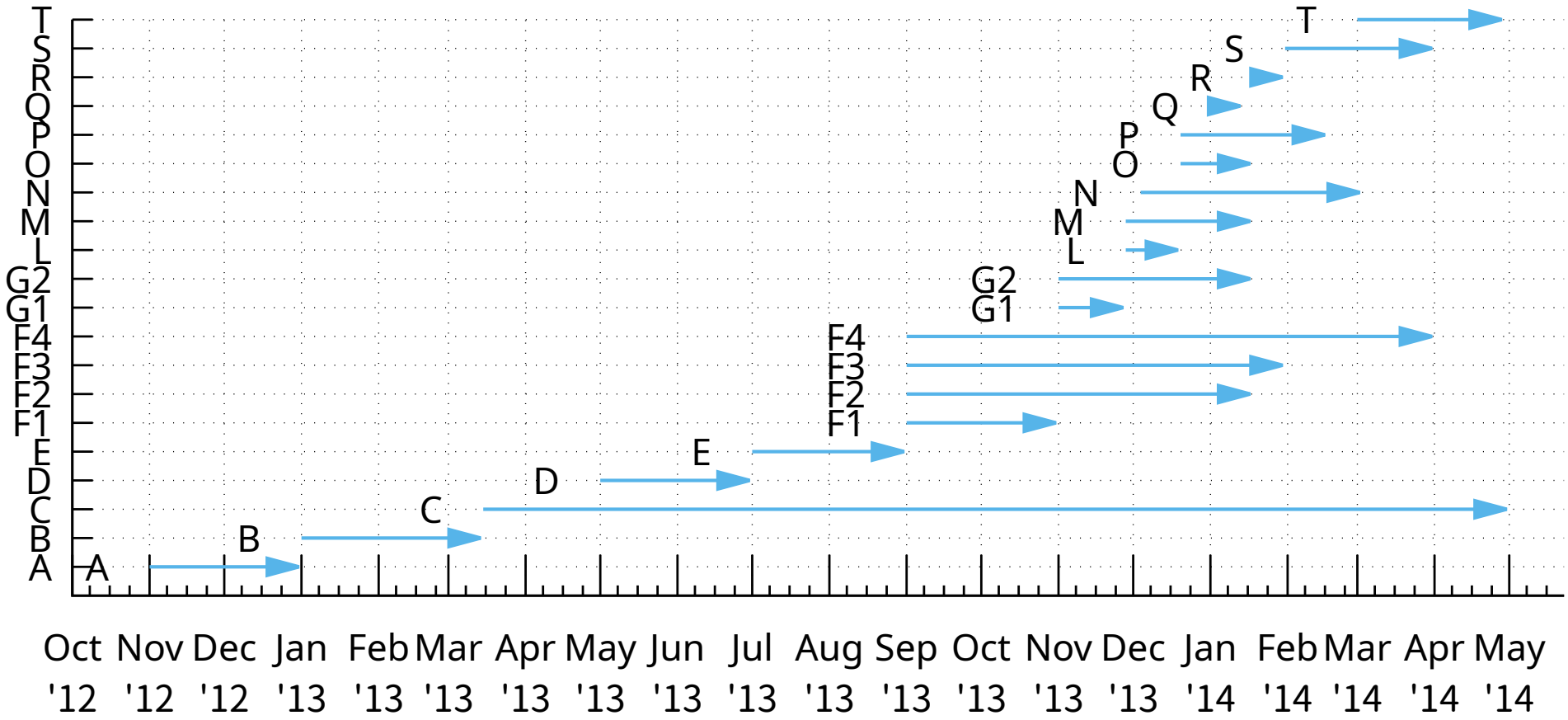

Time data on Y, millisecond precision
19:56:50.001
19:56:48.400
19:56:47.800
19:56:47.400
19:56:47.200
19:56:47.100
27 Jan 2014
19:56:50.001
27 Jan 2014
19:56:48.400
27 Jan 2014
19:56:47.800
27 Jan 2014
19:56:47.400
27 Jan 2014
19:56:47.200
27 Jan 2014
19:56:47.100
A
B
C
D
E
F

Date format (top) vs Time format (bottom)



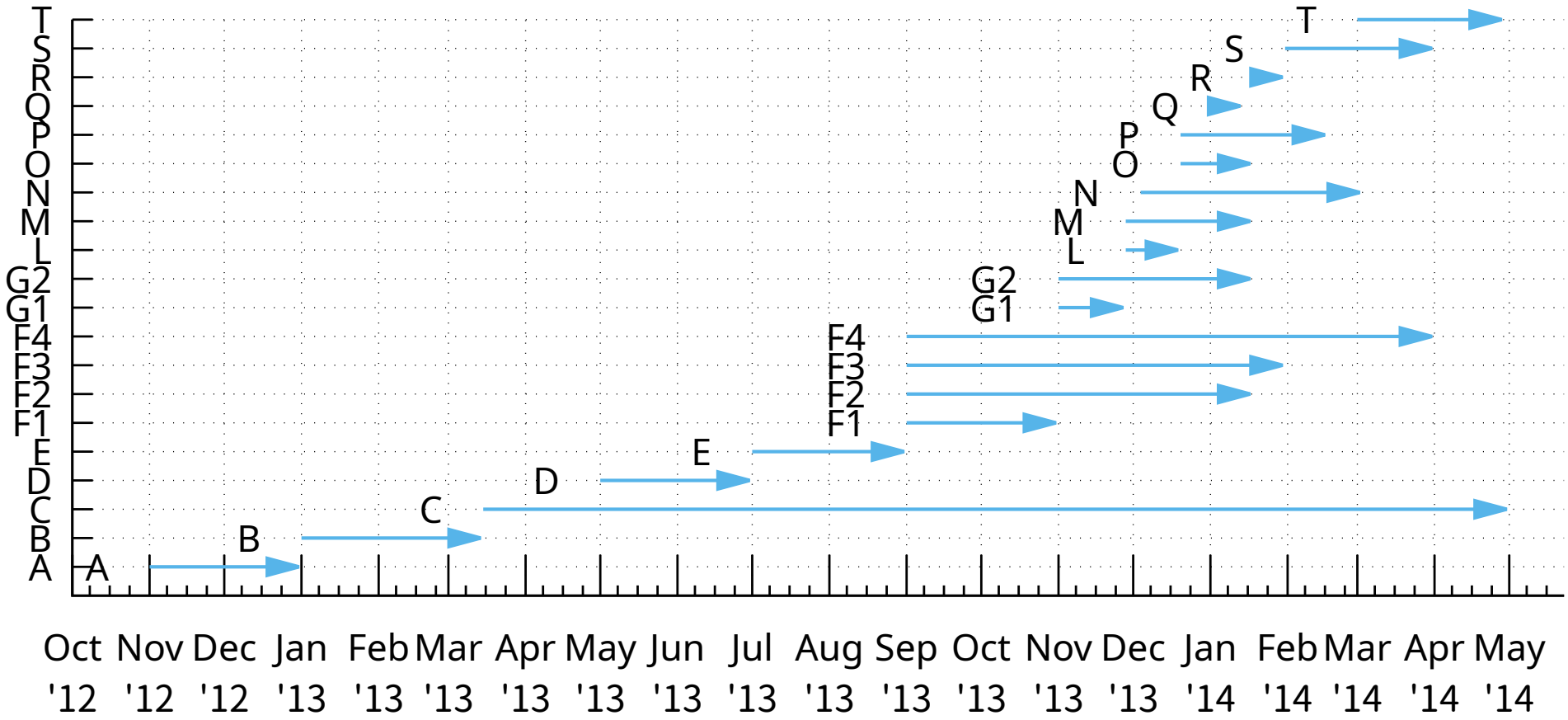
Simple Gantt Chart

Task start and end times in columns 2 and 3

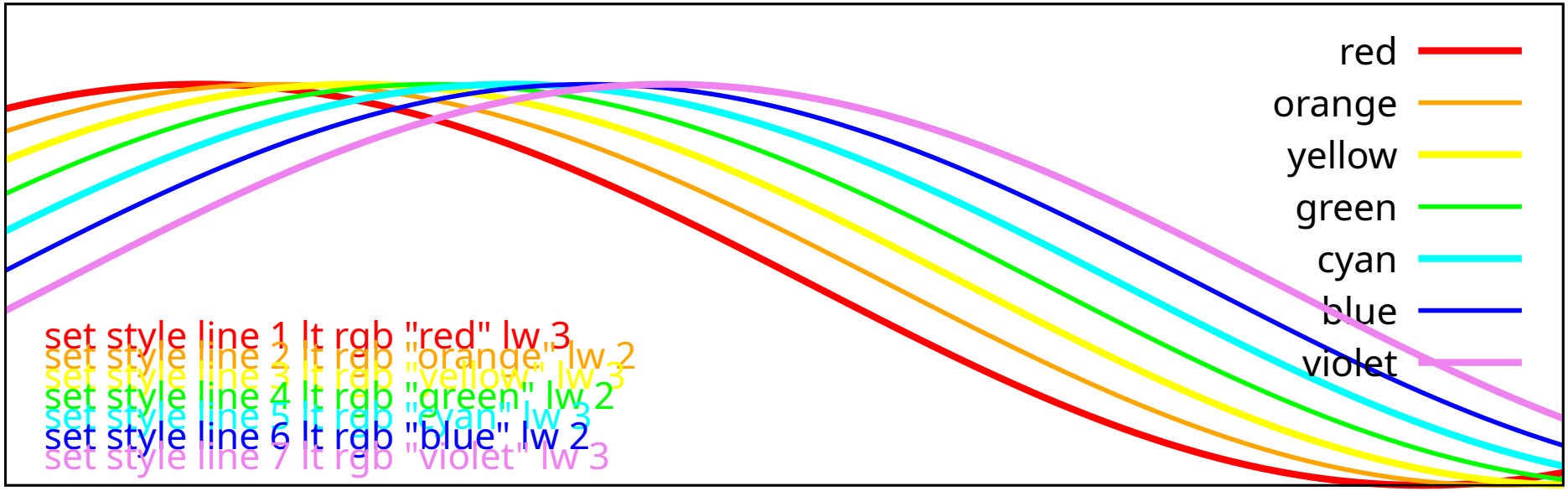


Simple Gantt Chart

Task start and end times in columns 2 and 3

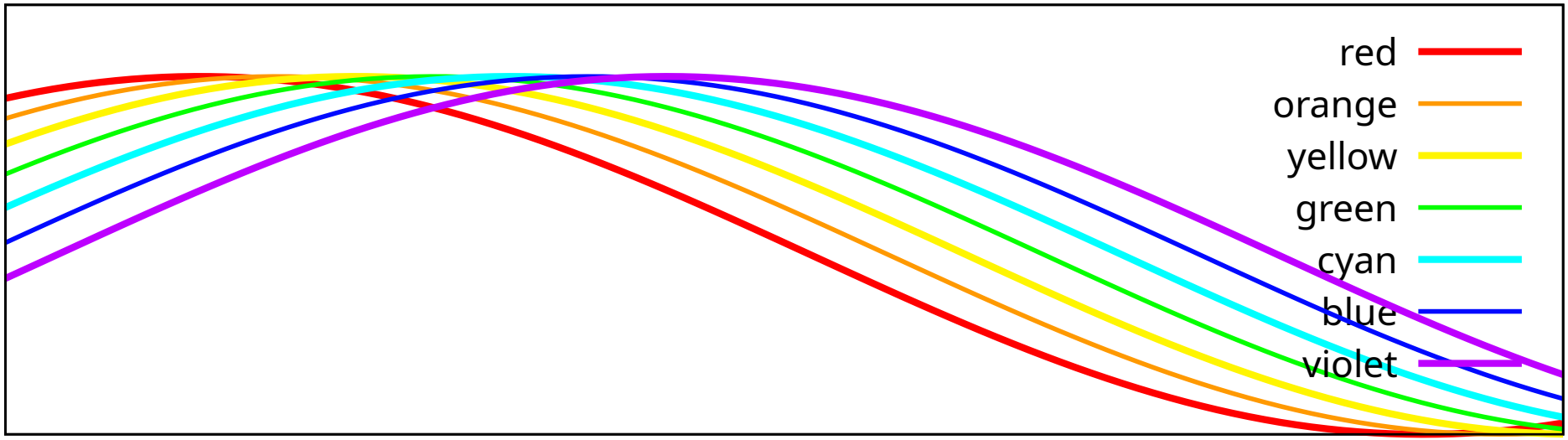


Terminal-independent RGB colors in 2D

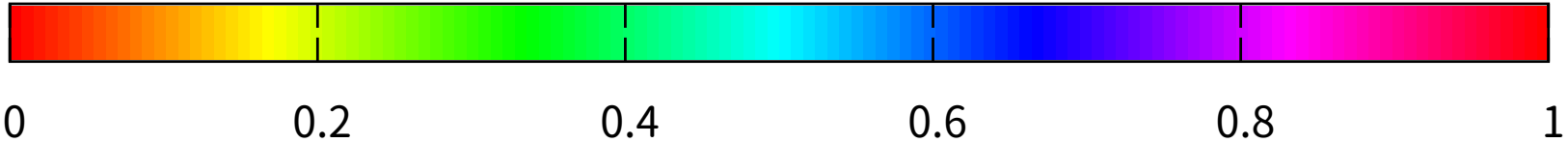


Implemented using built-in rgb color names
(only works for terminals that can do full rgb color)

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

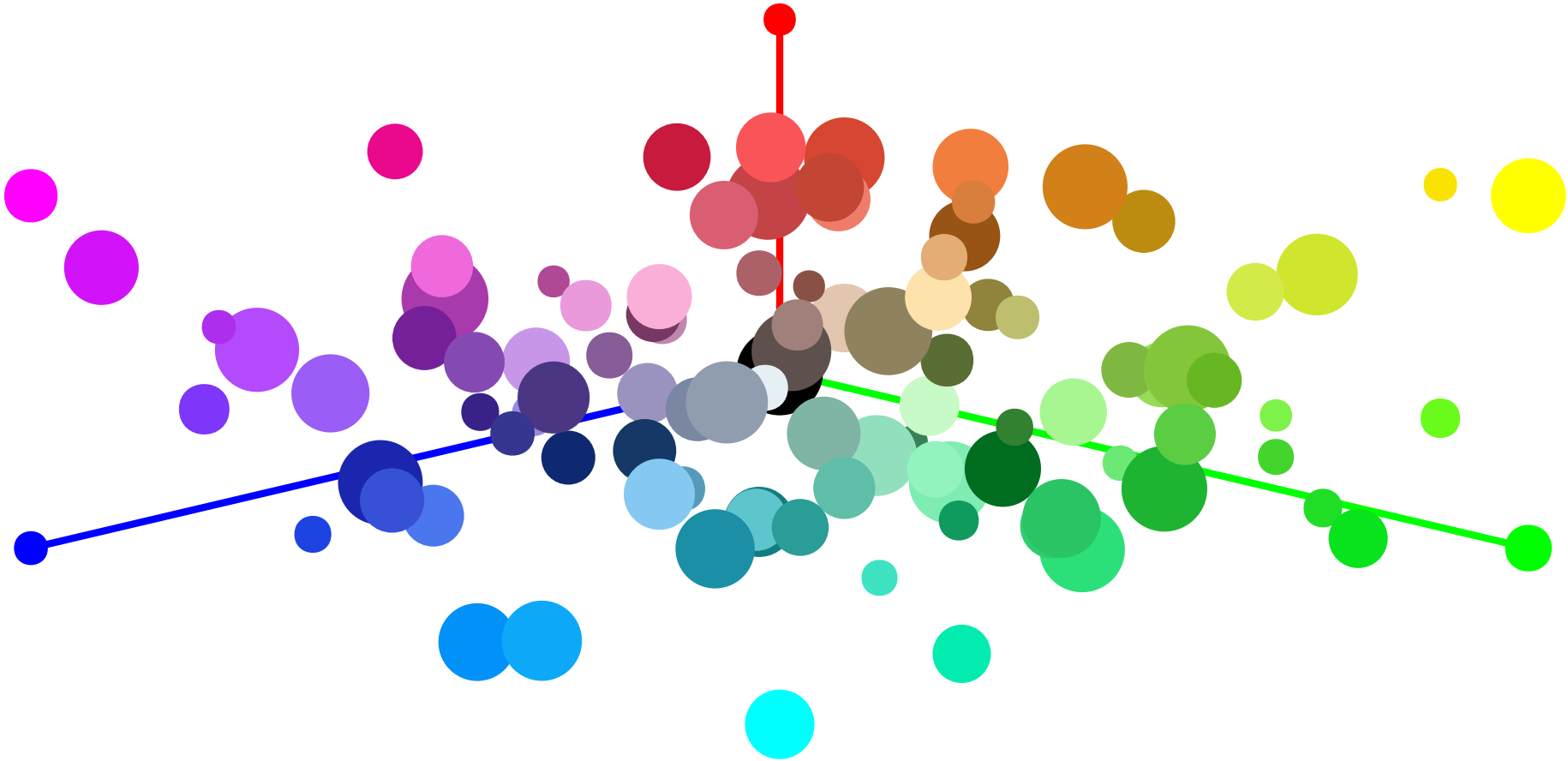


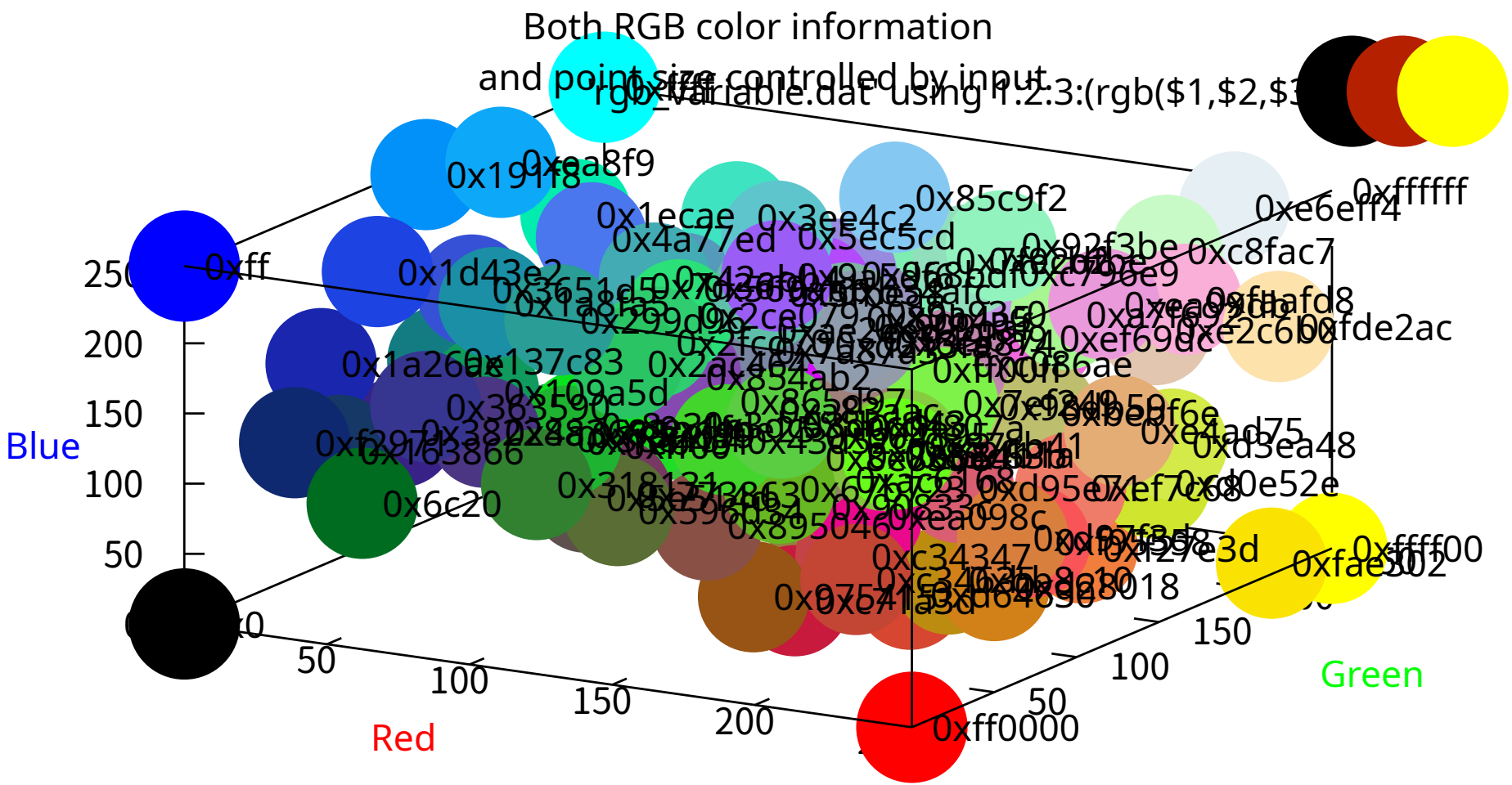
HSV color wheel

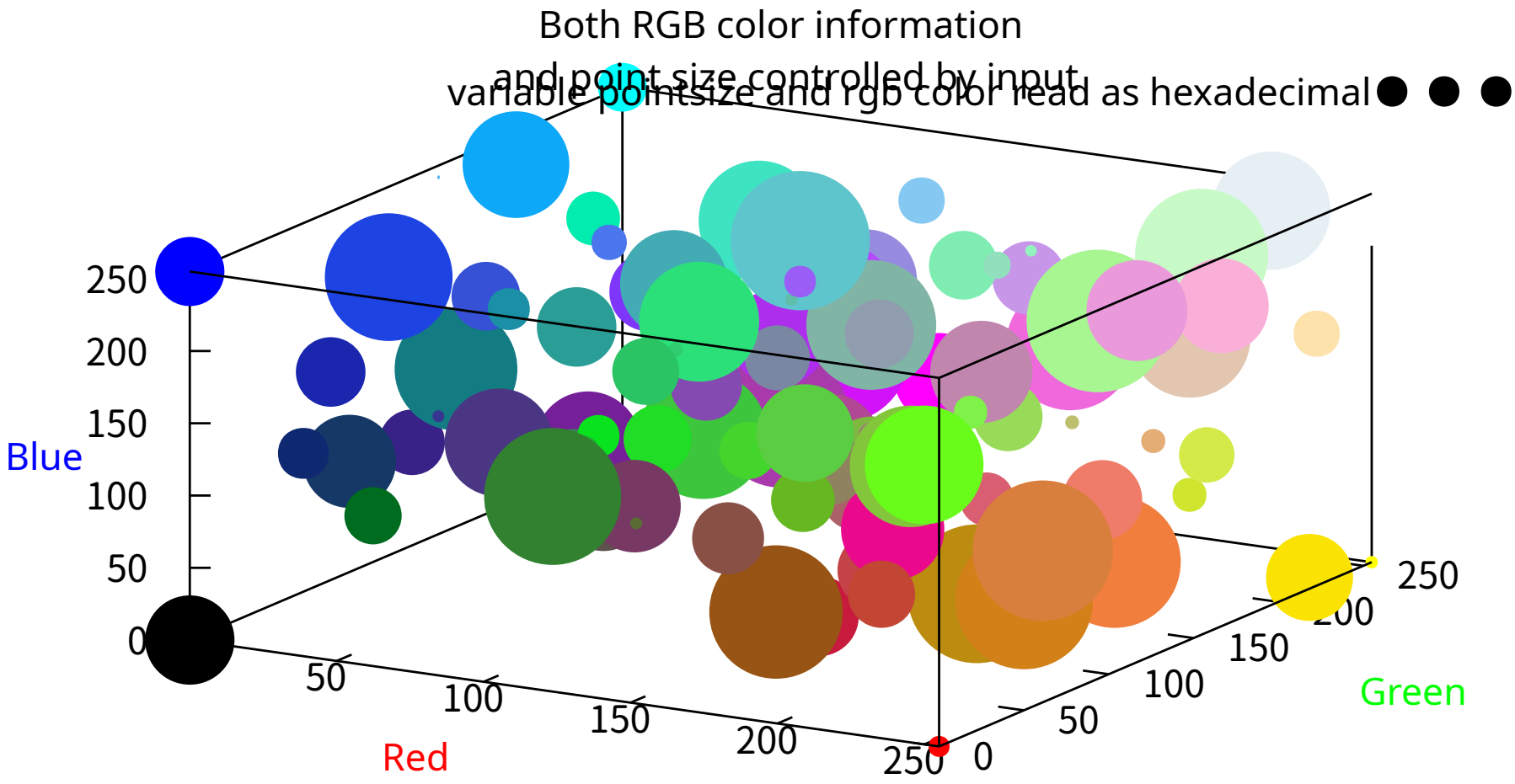


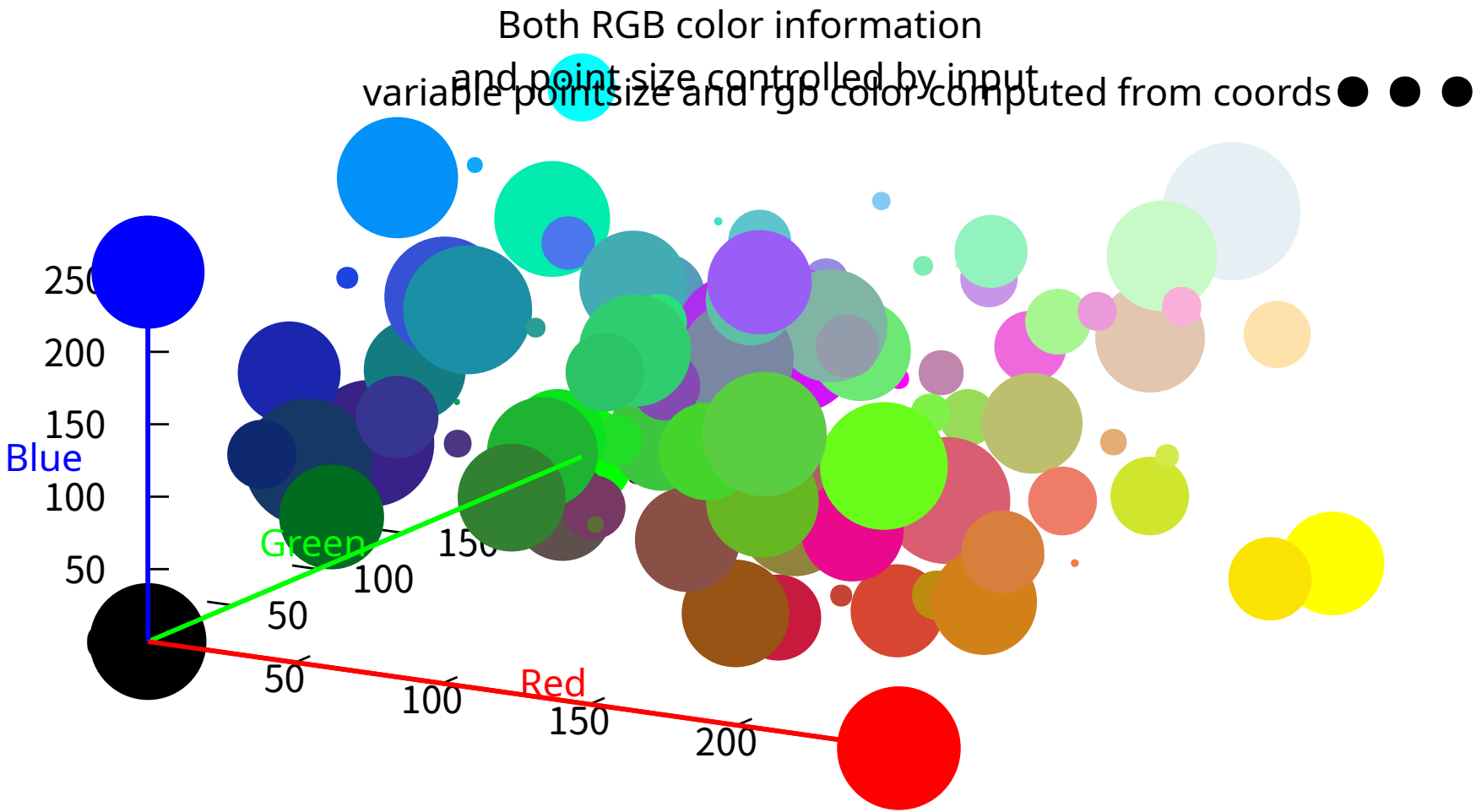
RGB color information read from data file

Both RGB color information
and point size controlled by input



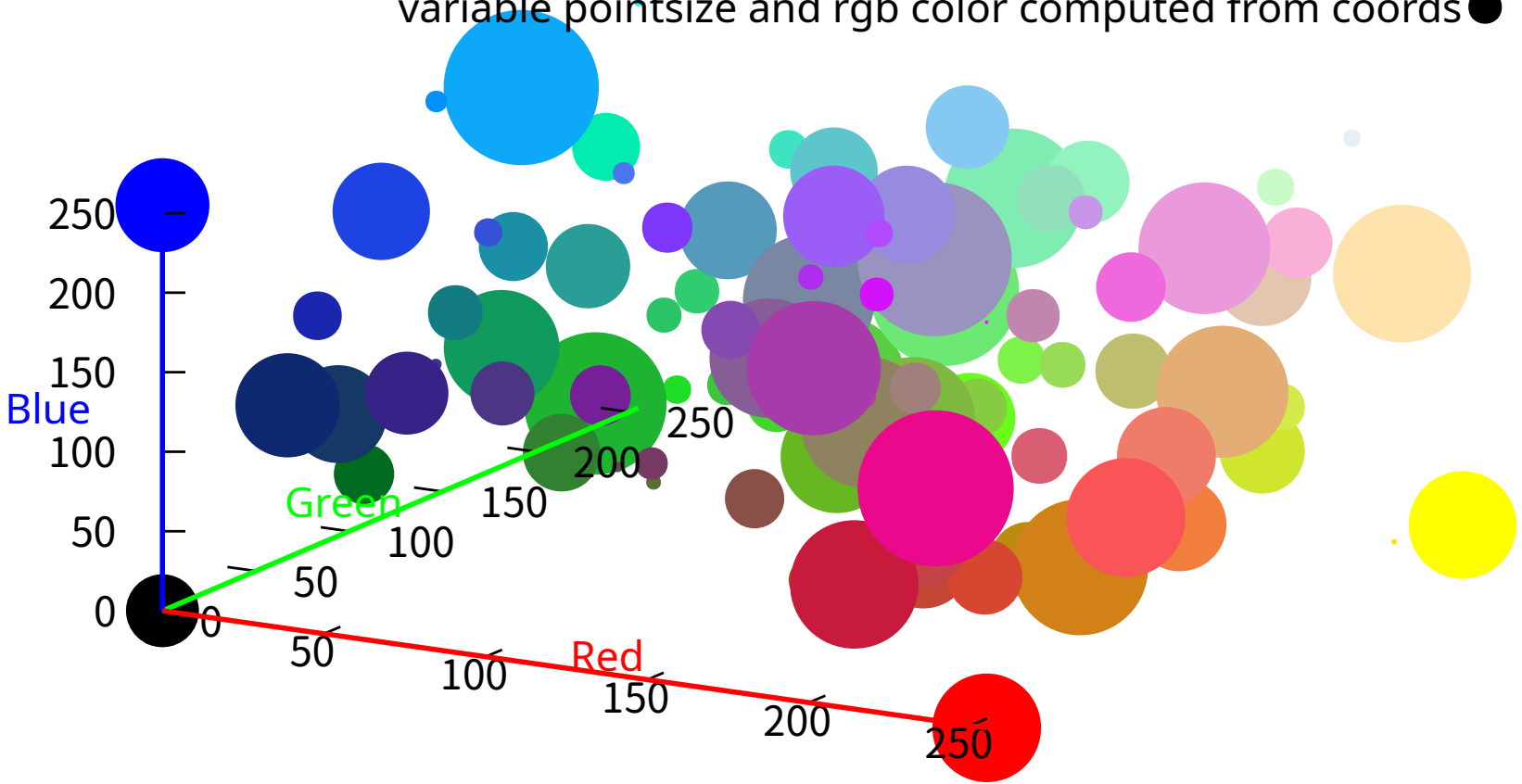

Both RGB color information
and point size controlled by input
rgb_variable.dat using 1:2:3:(rgb($1,$2,$3
Blue
Red
Green
0xff
0xffffff
0xff0000
0xe6eff4
0xc8fac7
0xd3ea48
0xd0e52e
0xfae302
0xffff00
0x191f8
0xea8f9
0x1ecae
0x3ee4c2
0x85c9f2
0x4a77ed
0x5ec5cd
0x92f3be
0x1d43e2
0x137c83
0x109a5d
0x163866
0x6c20
0x895046
0xc34347
0xc71a3d
0xd95e71
0xef7c68
0xe84ad75
0xe2c6b0
0xfde2ac
0xe3d
0x50
100
150
200
250


Both RGB color information
and point size controlled by input
variable pointsize and rgb color read as hexadecimal
Blue
Red
Green
0
50
100
150
200
250

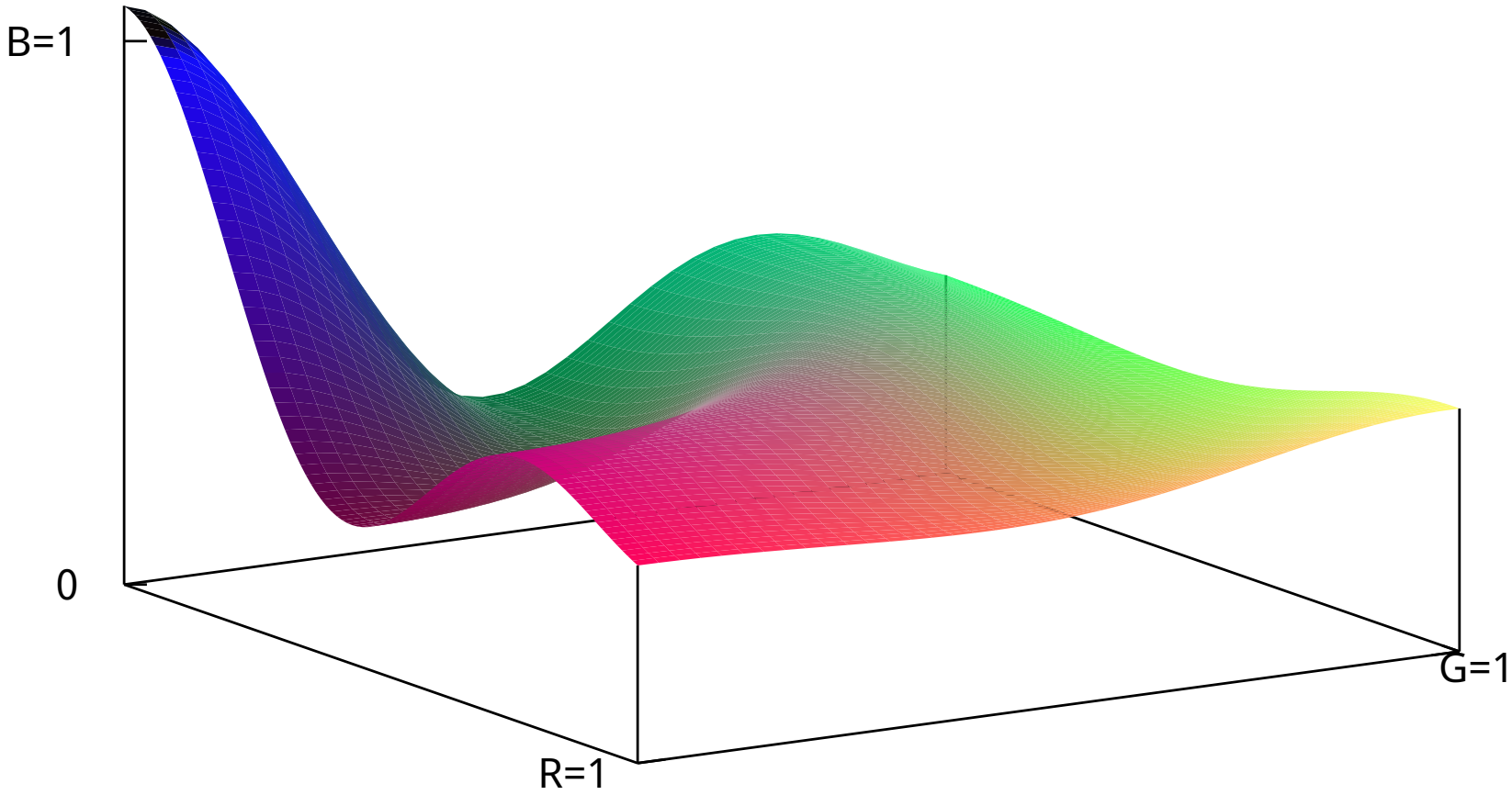

Both RGB color information
and point size controlled by input
variable pointsize and rgb color computed from coords
Blue
Green
Red
250
200
150
100
50
50
100
150
50
100
150
200

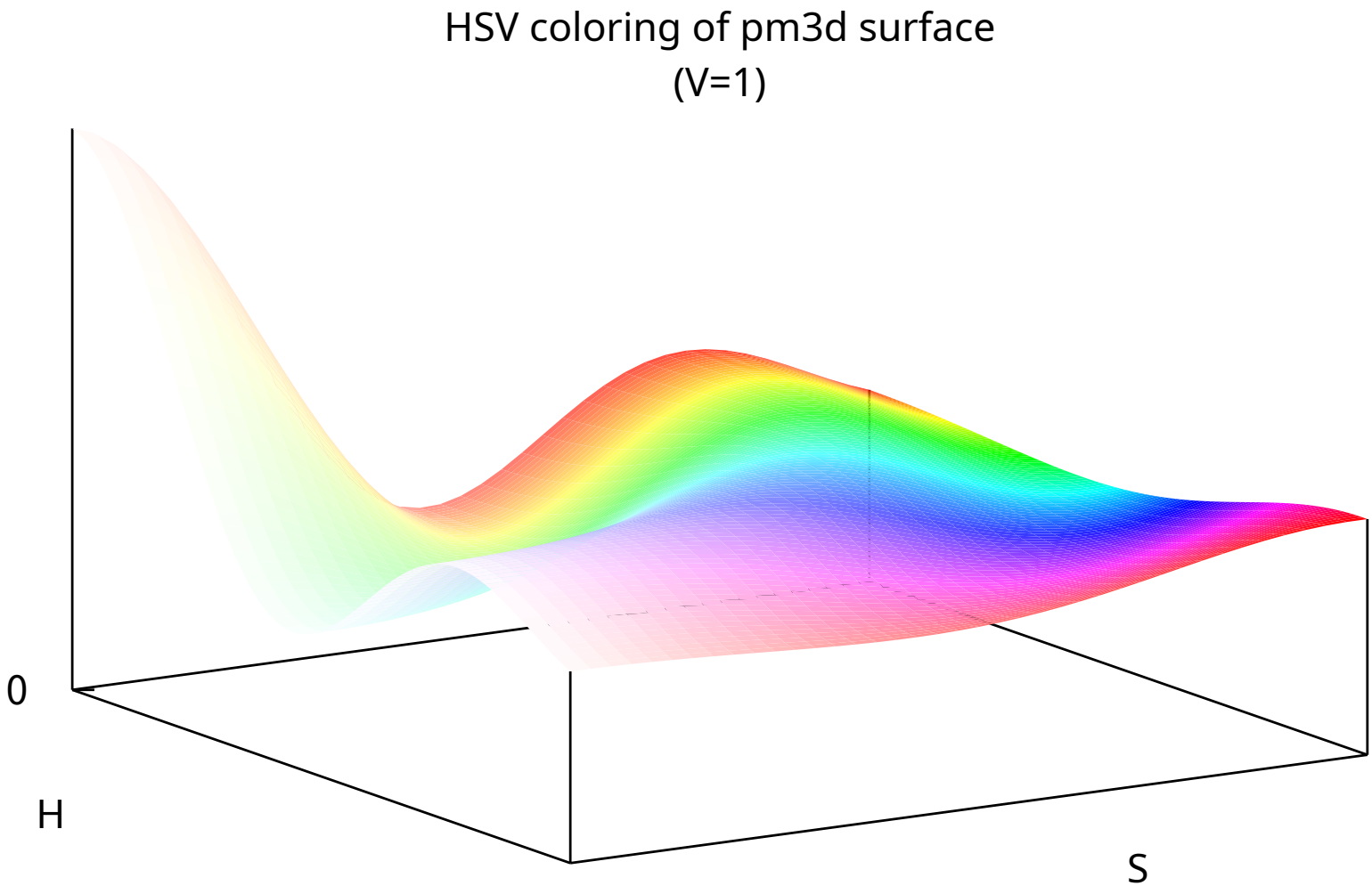
Demo of hidden3d with points only (no surface)

variable pointsize and rgb color computed from coords

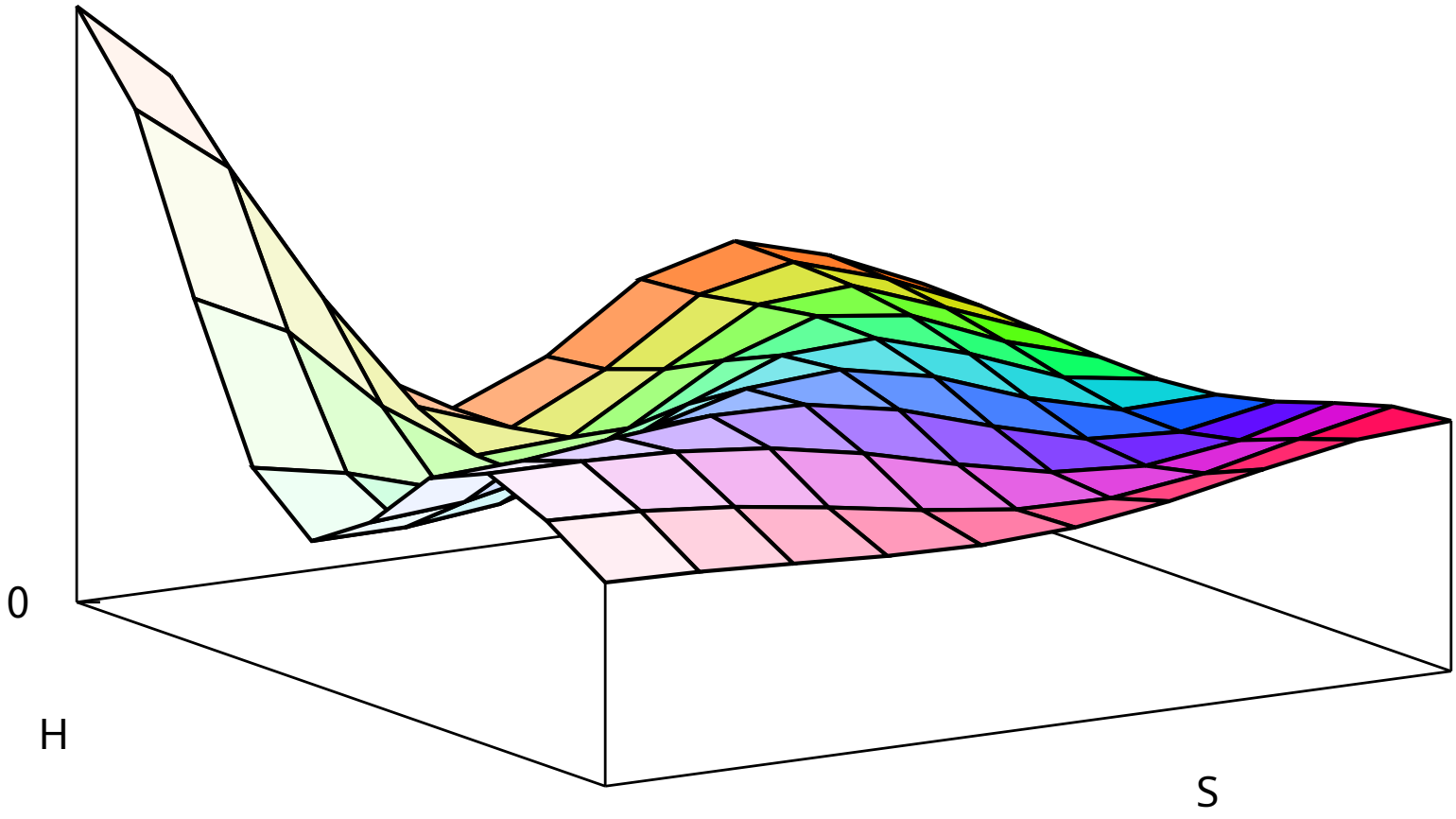


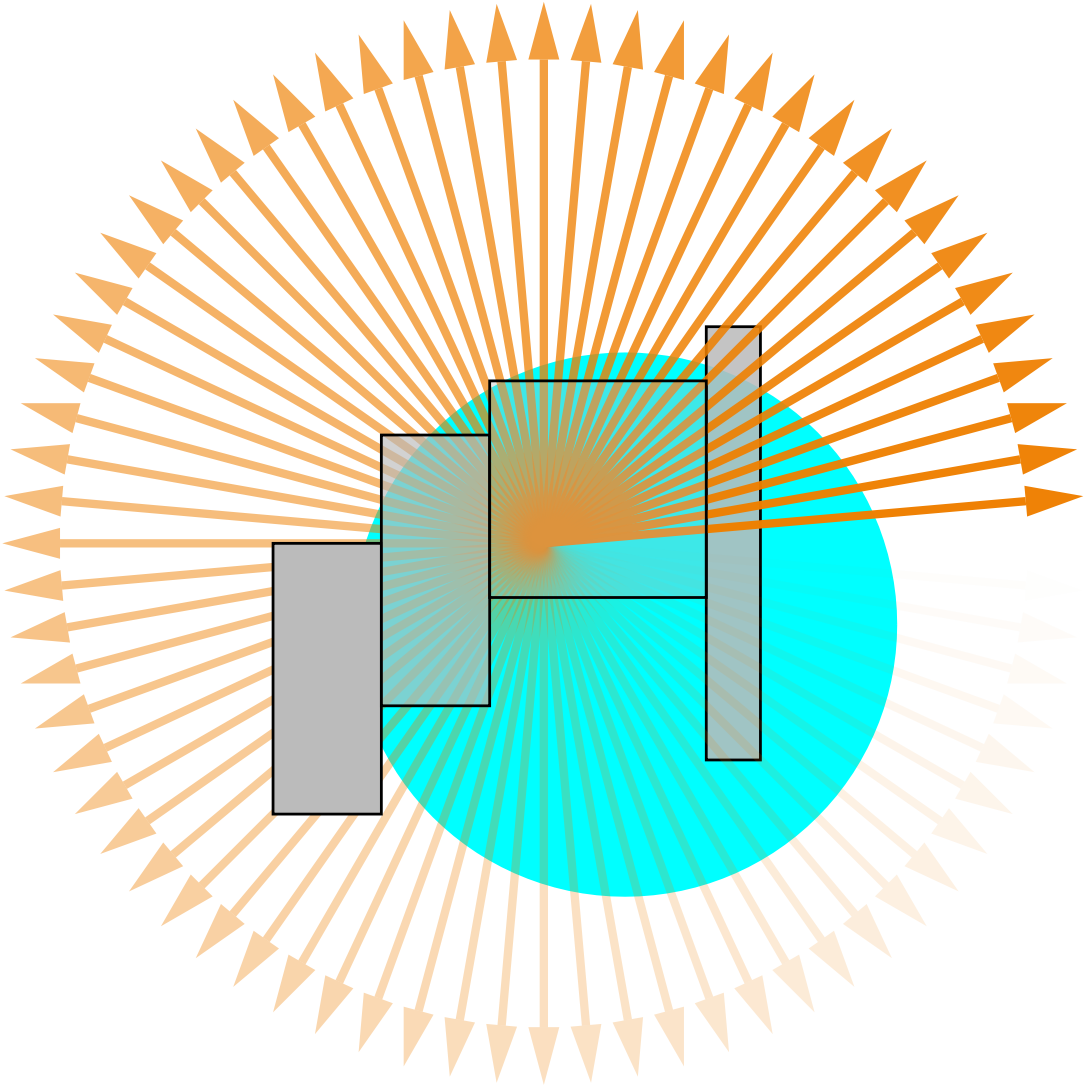
RGB coloring of pm3d surface

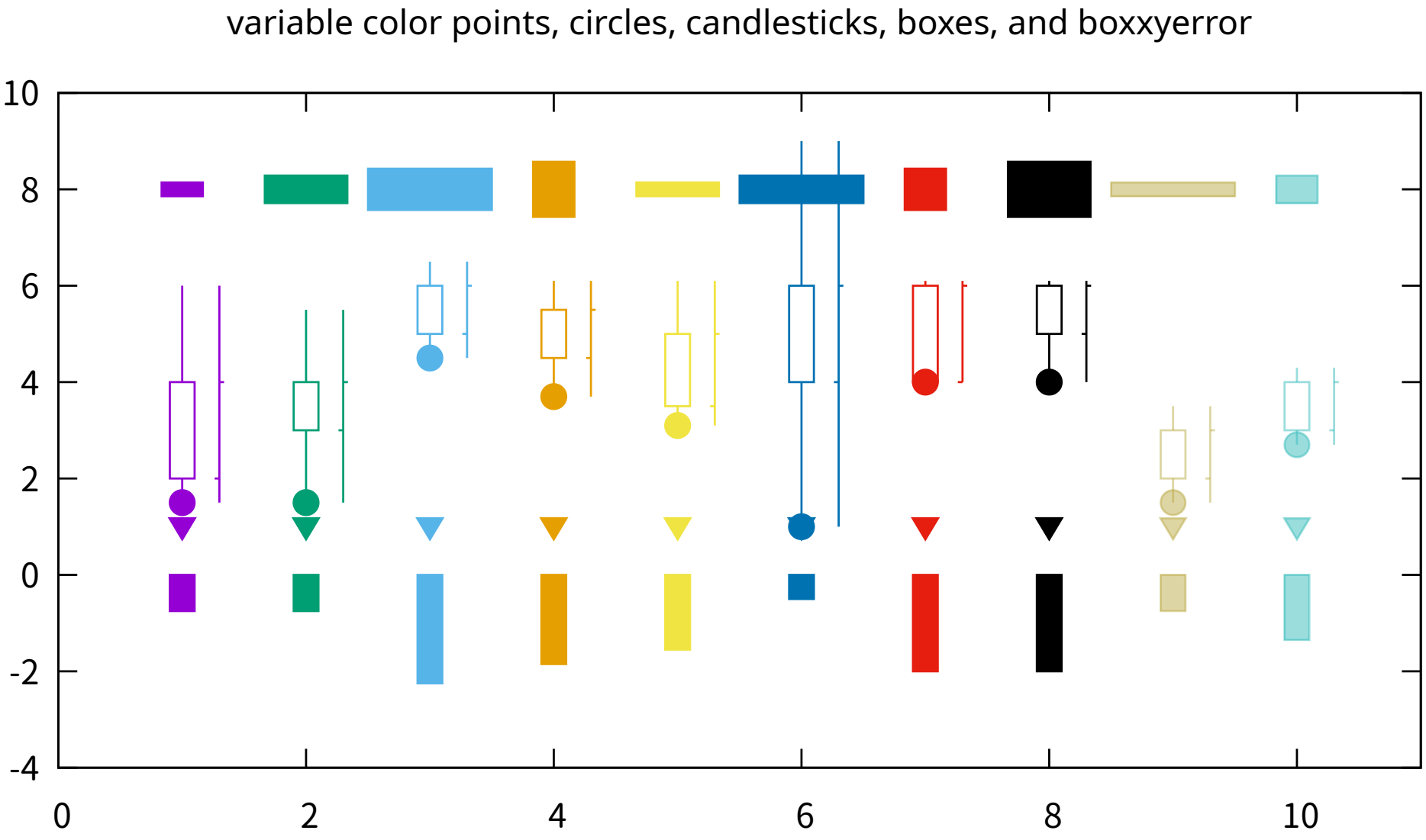


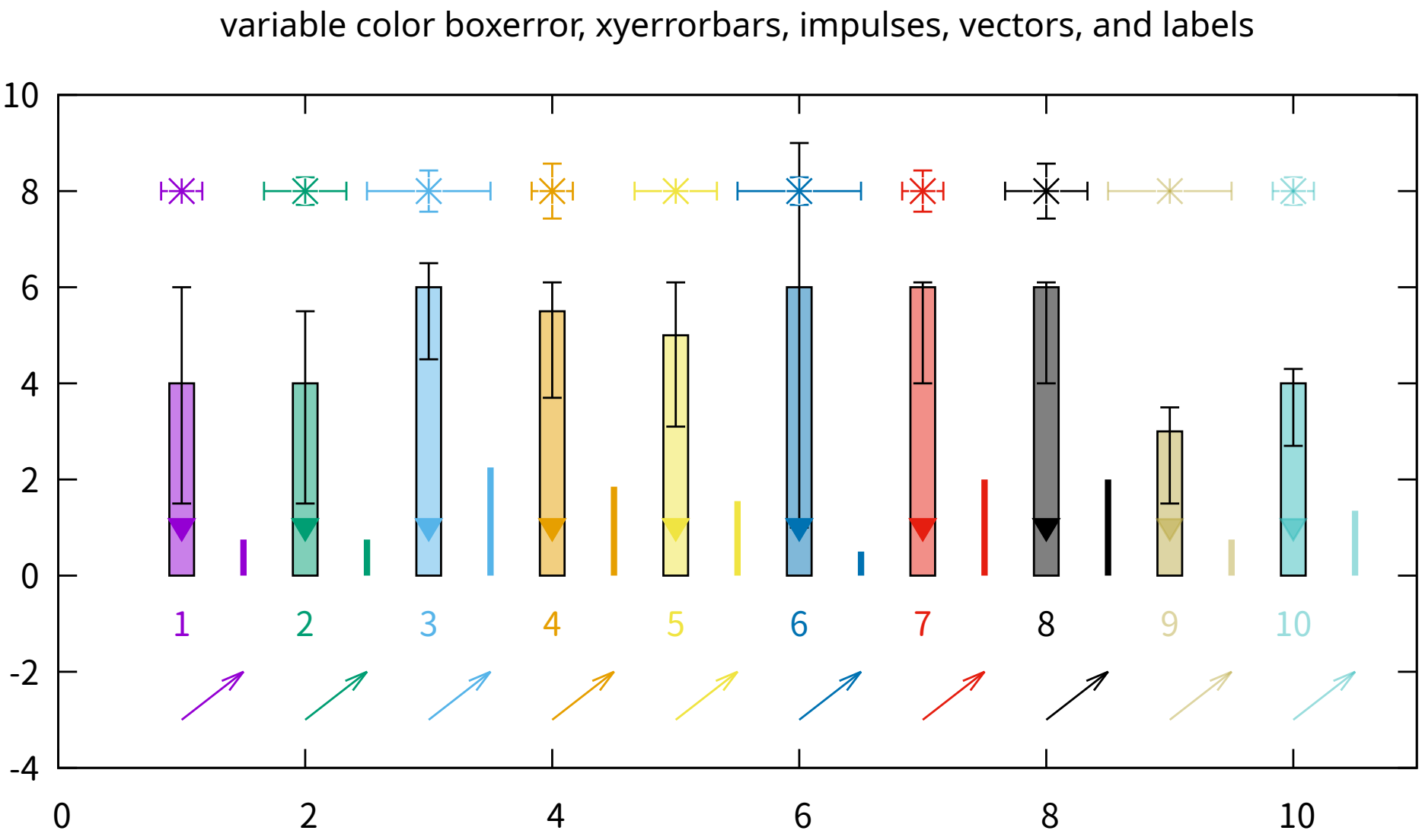

HSV coloring of pm3d surface
(V=1)
0
H
S

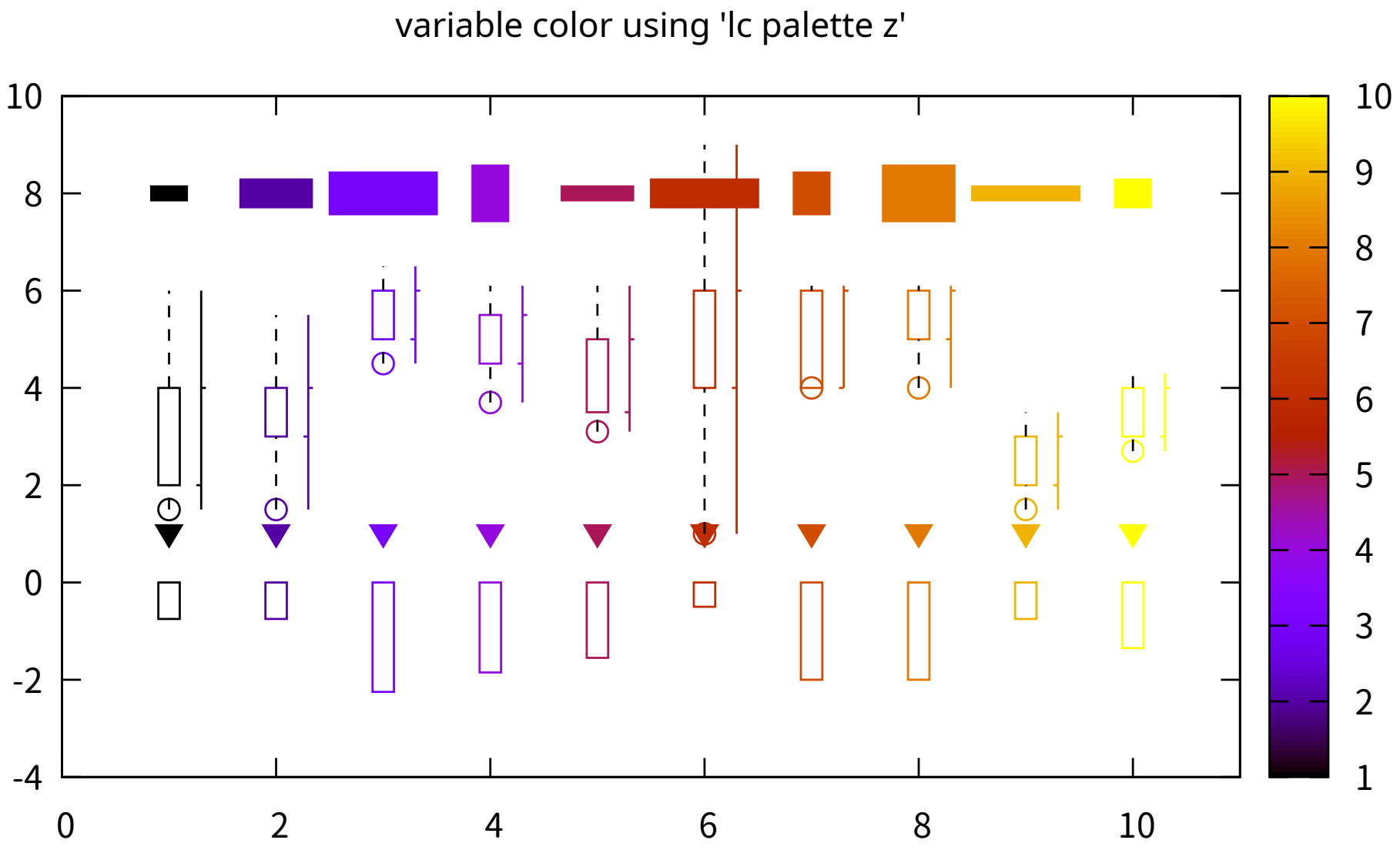
Explicit borders for pm3d tiling

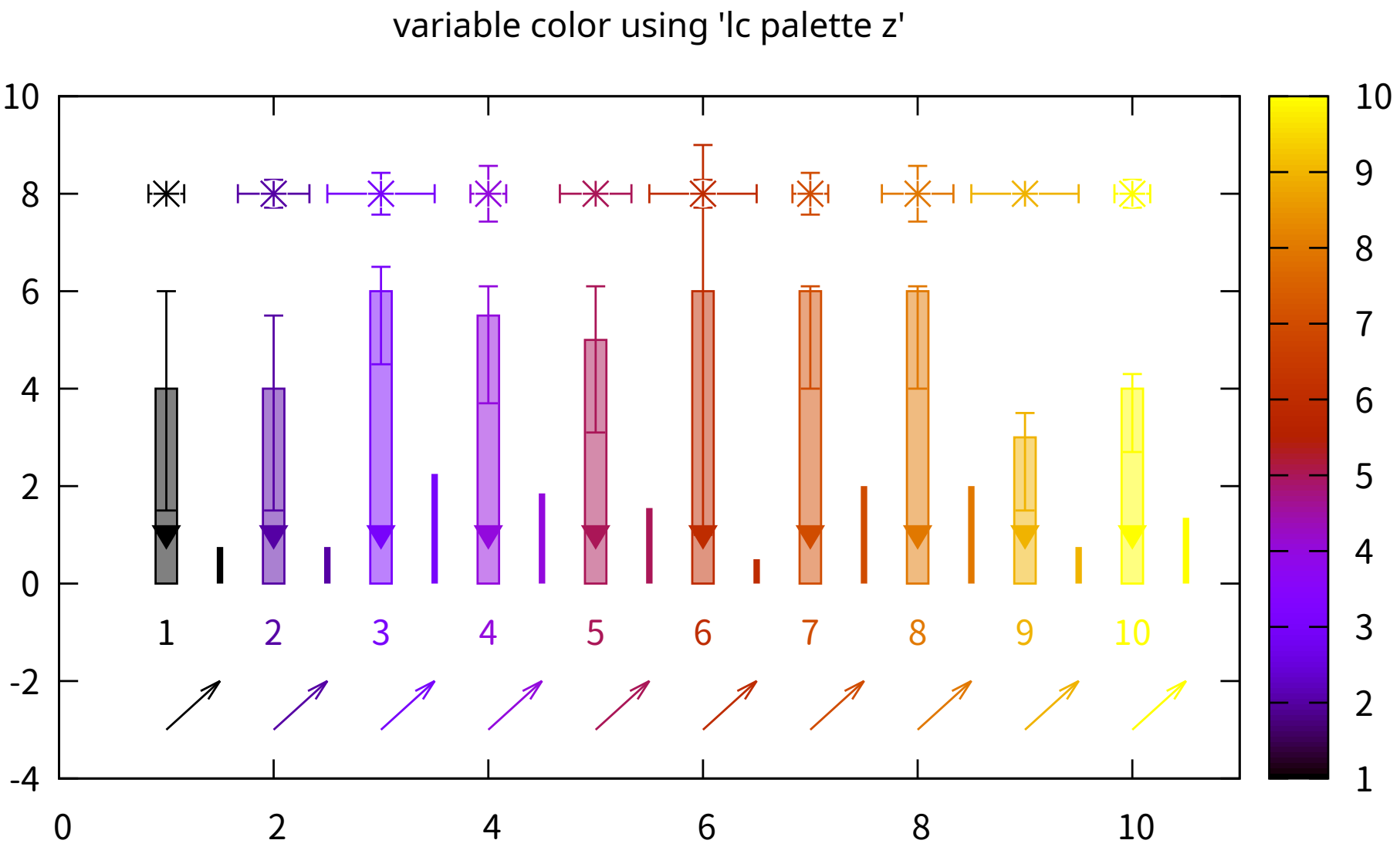


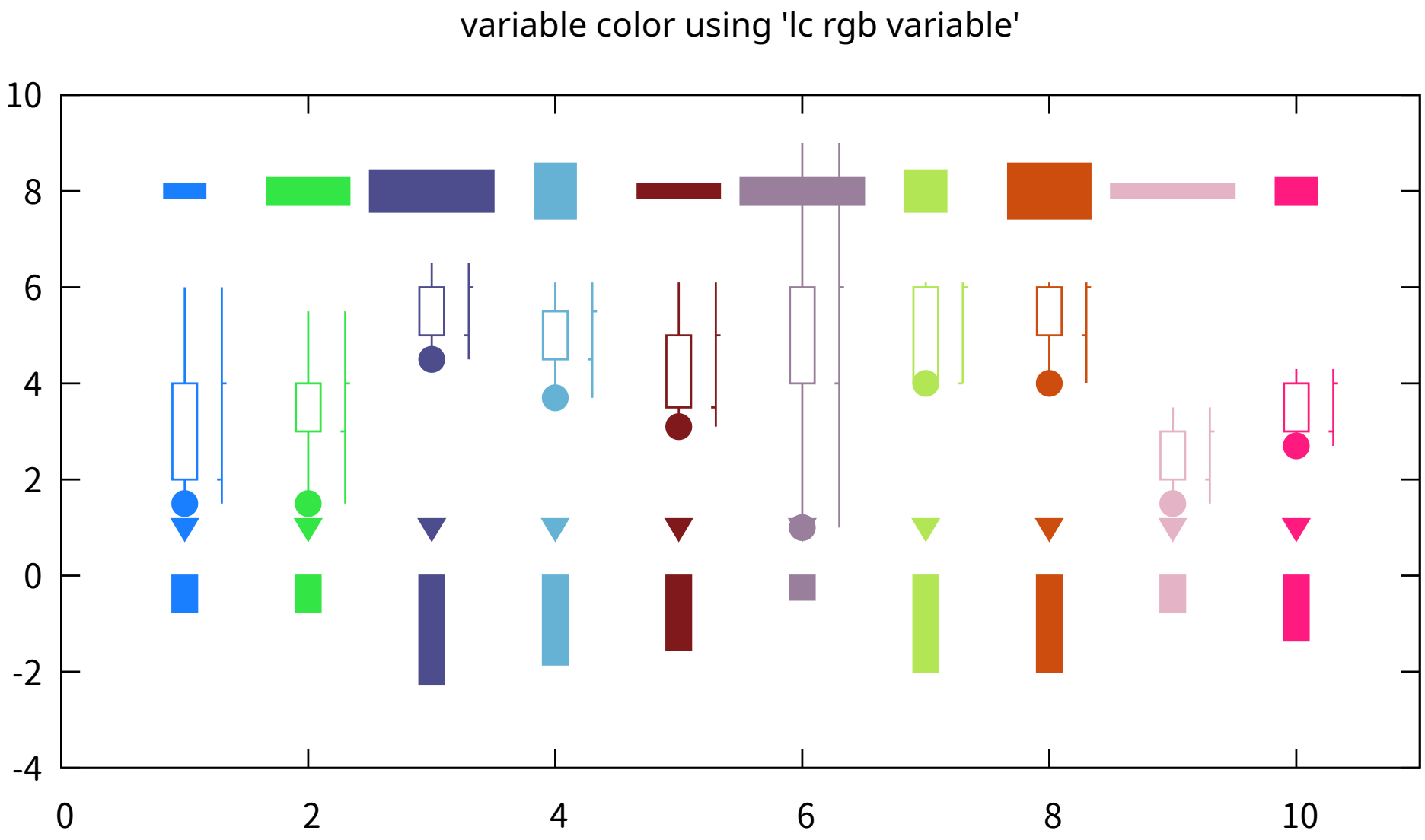


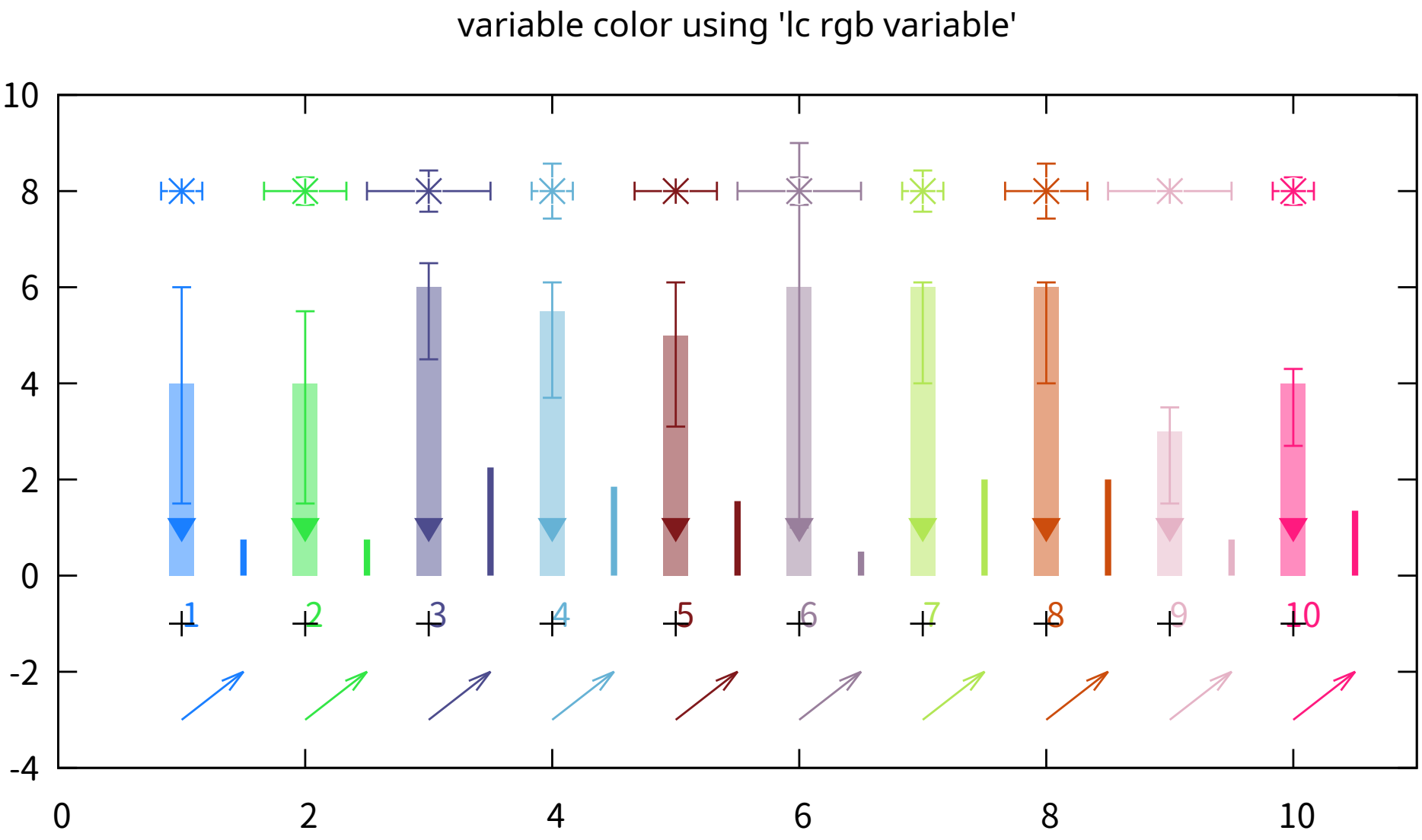

variable color points, circles, candlesticks, boxes, and boxxyerror
10
8
6
4
2
0
-2
-4
0
2
4
6
8
10

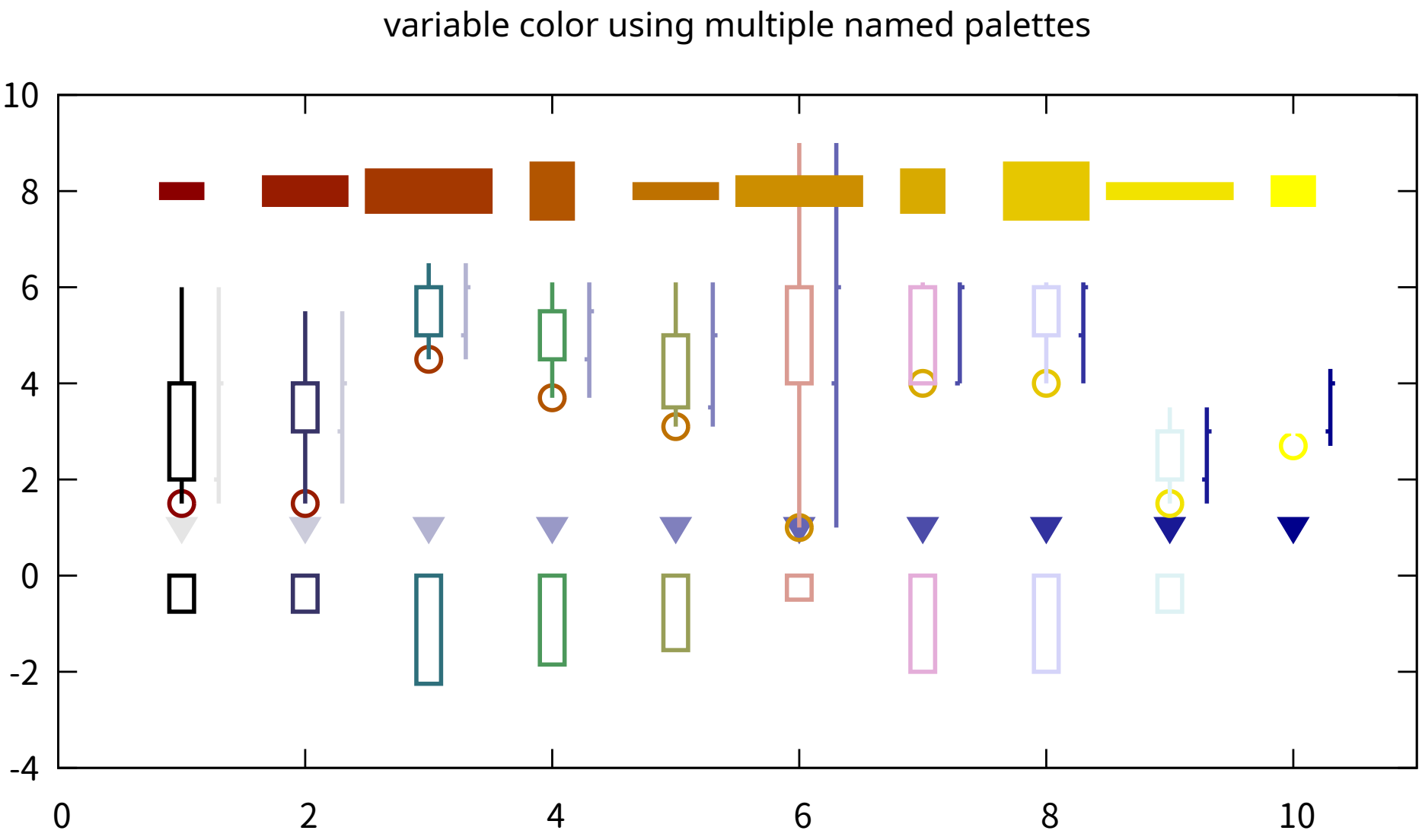

variable color boxerror, xyerrorbars, impulses, vectors, and labels
10
8
6
4
2
0
-2
-4
0
2
4
6
8
10
1
2
3
4
5
6
7
8
9
10

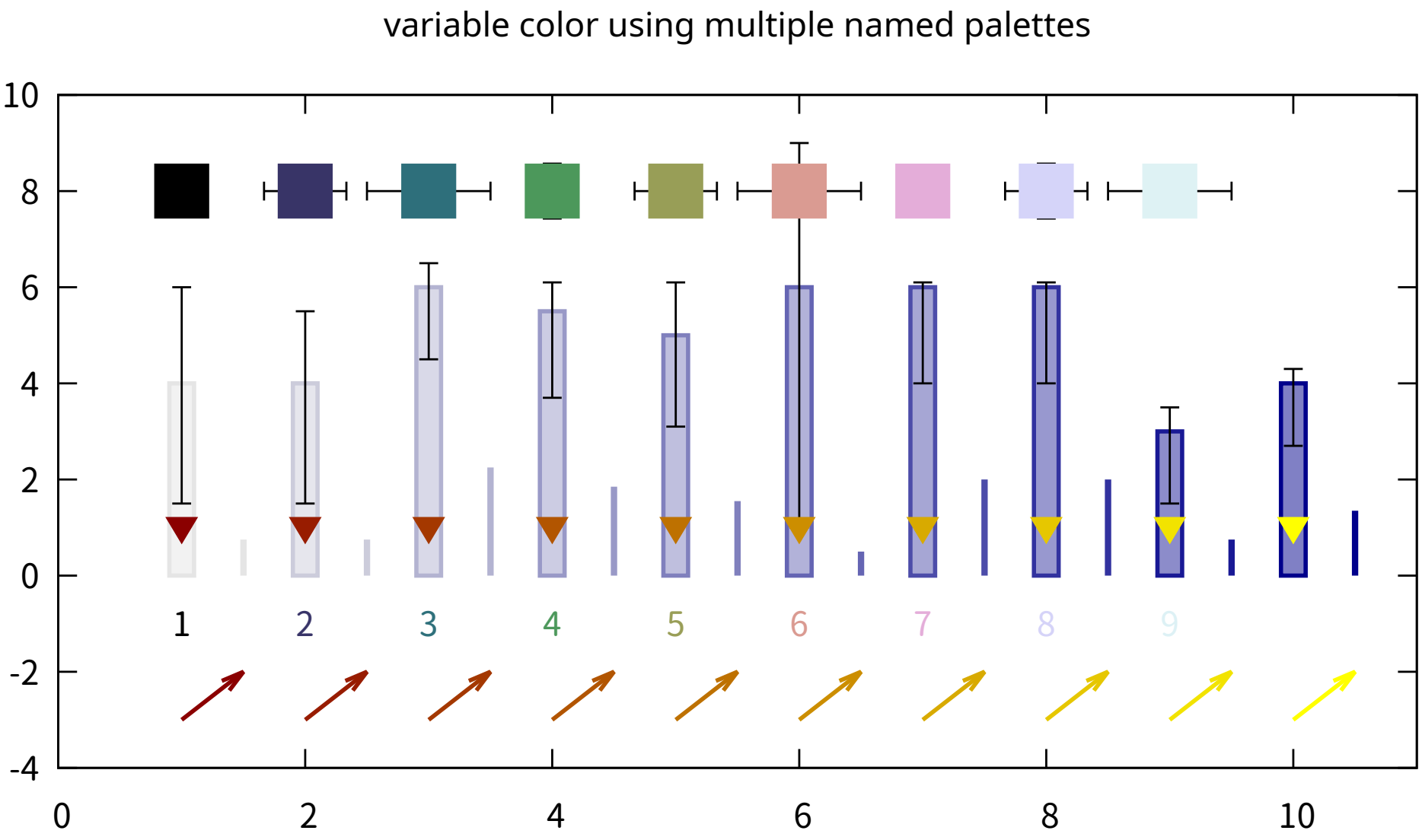

variable color using 'lc palette z'
10
8
6
4
2
0
-2
-4
0
2
4
6
8
10
10
9
8
7
6
5
4
3
2
1


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

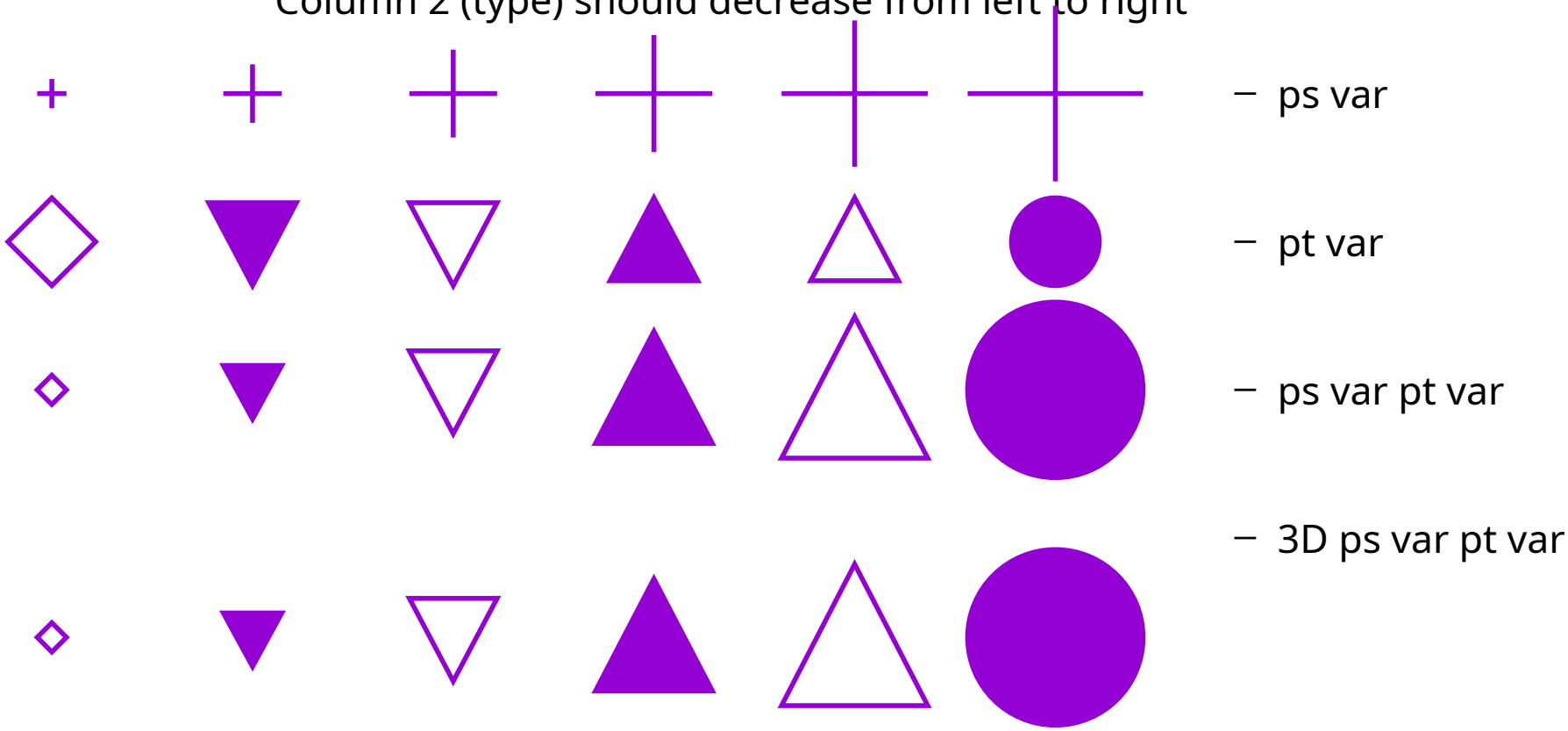

variable color using 'lc rgb variable'
10
8
6
4
2
0
-2
-4
0
2
4
6
8
10


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


variable color using multiple named palettes
10
8
6
4
2
0
-2
-4
0
2
4
6
8
10

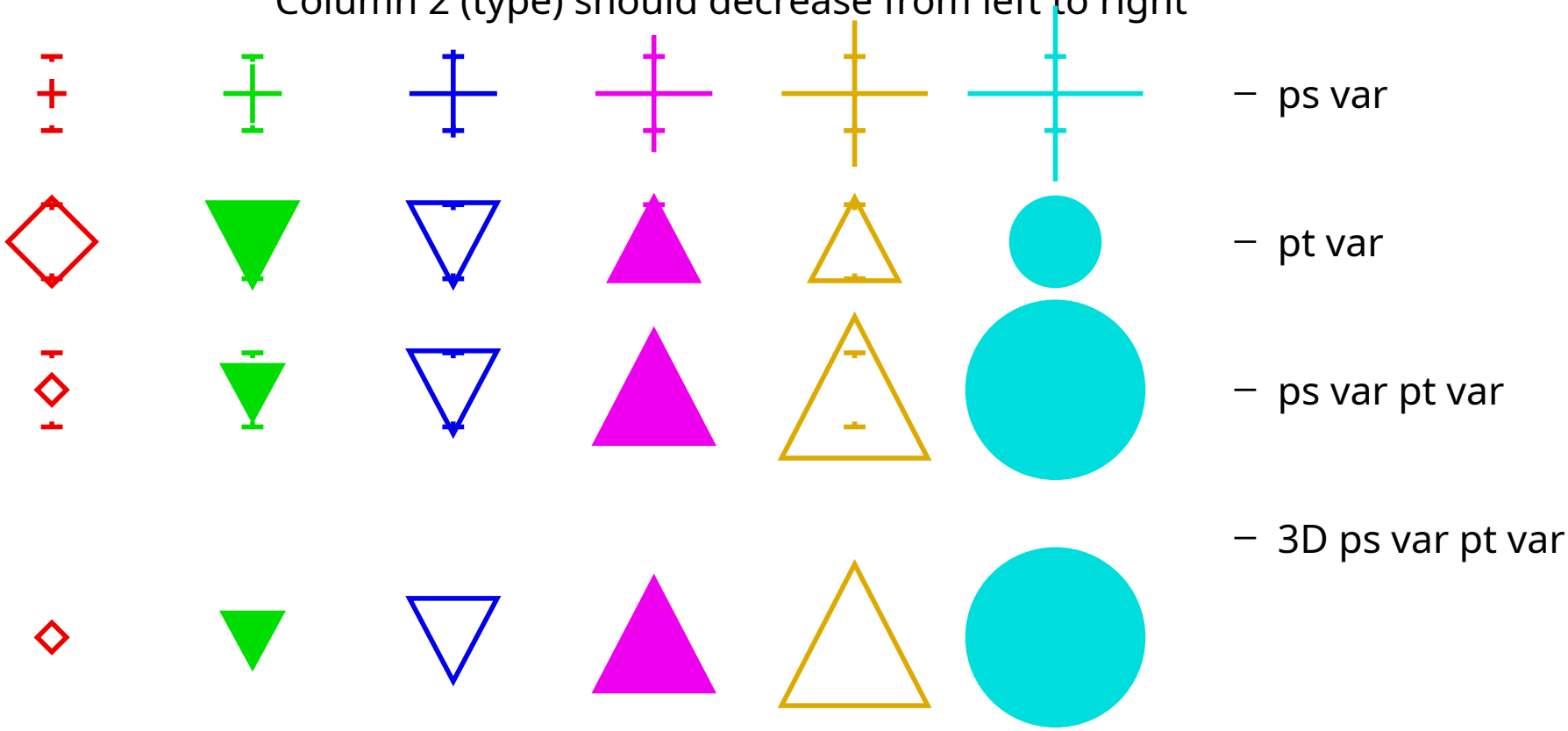

variable color using multiple named palettes
10
8
6
4
2
0
-2
-4
0
2
4
6
8
10
1
2
3
4
5
6
7
8
9

Column 1 (size) should increase from left to right
Column 2 (type) should decrease from left to right

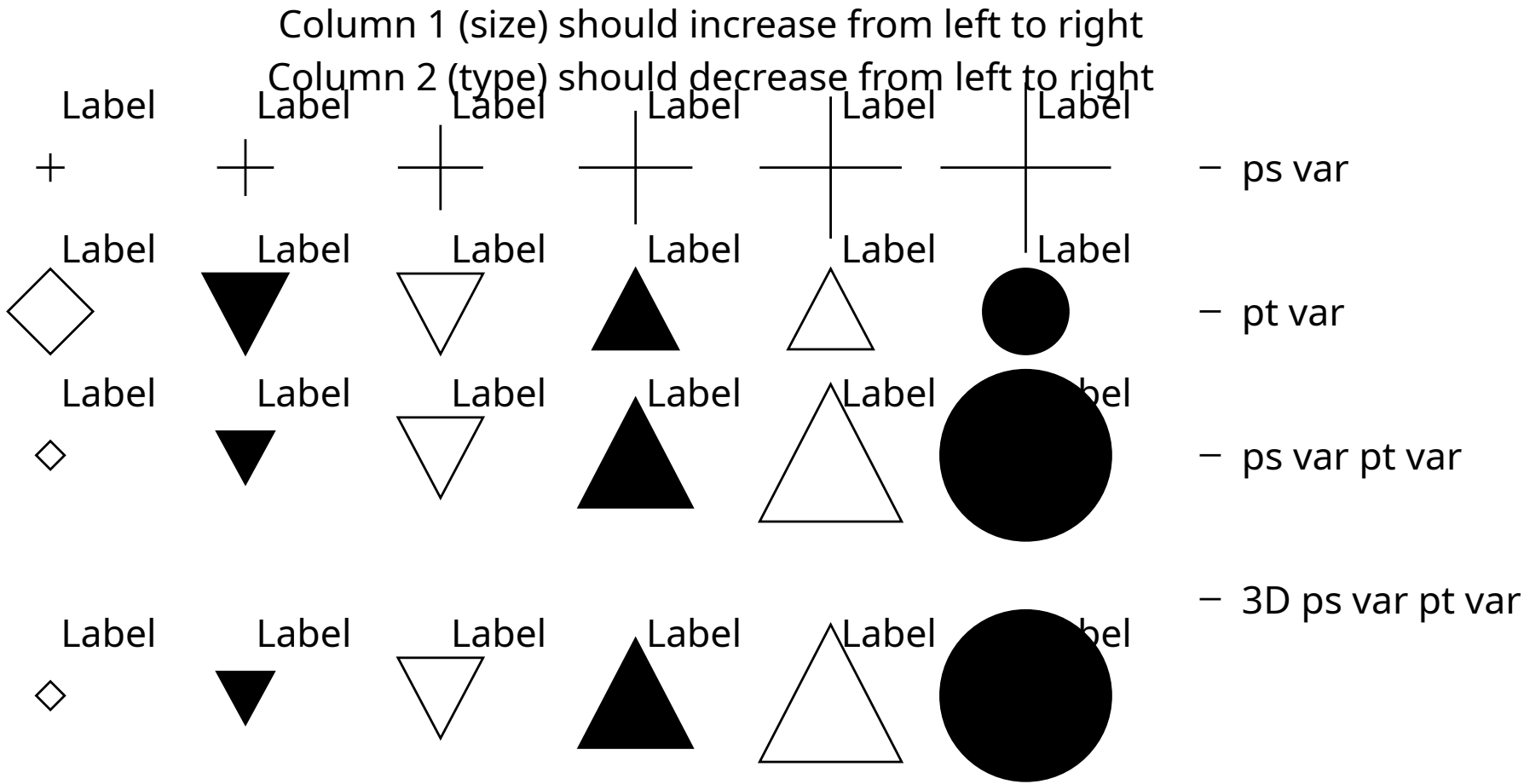


with points

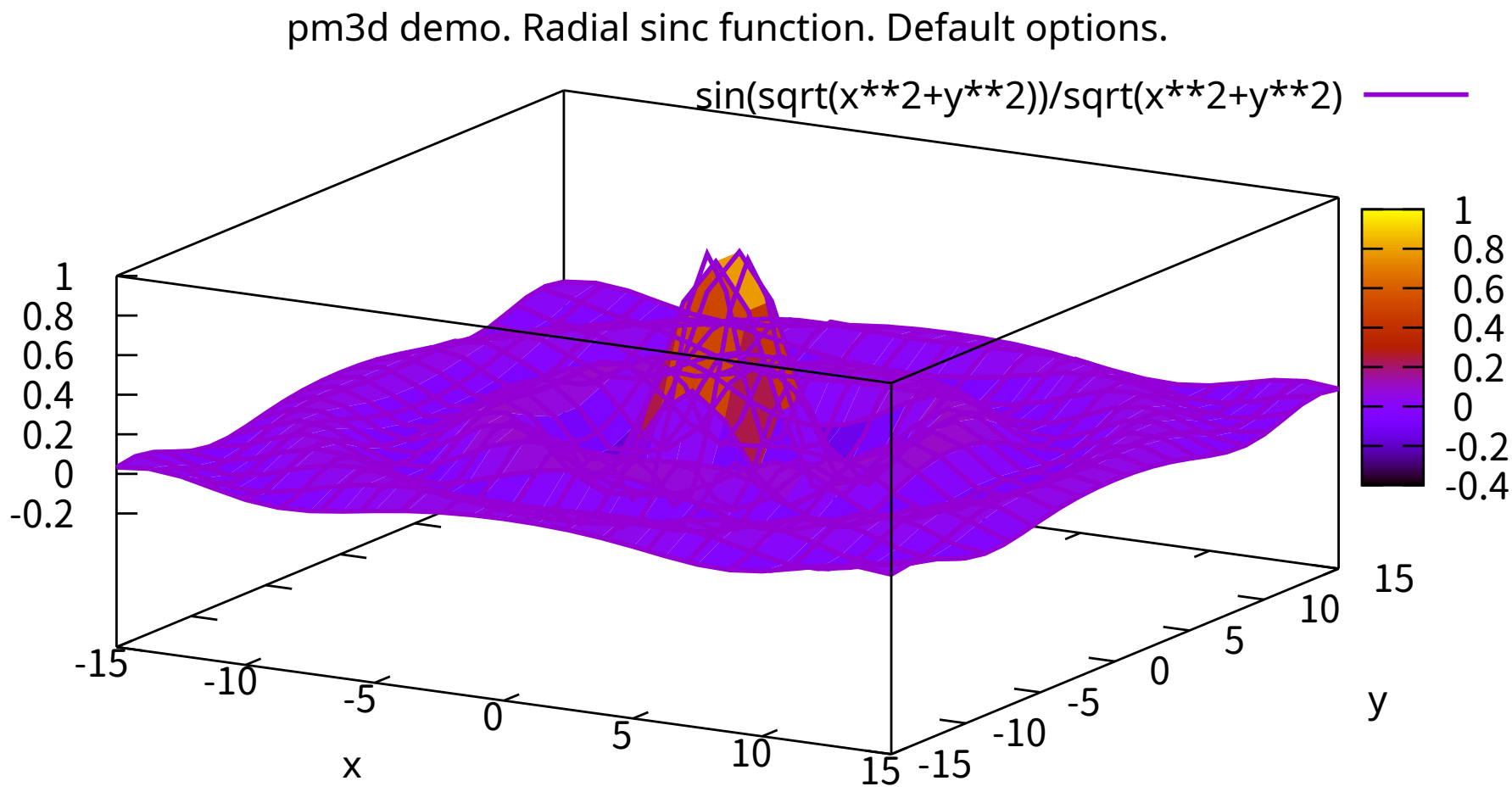
Column 1 (size) should increase from left to right
Column 2 (type) should decrease from left to right

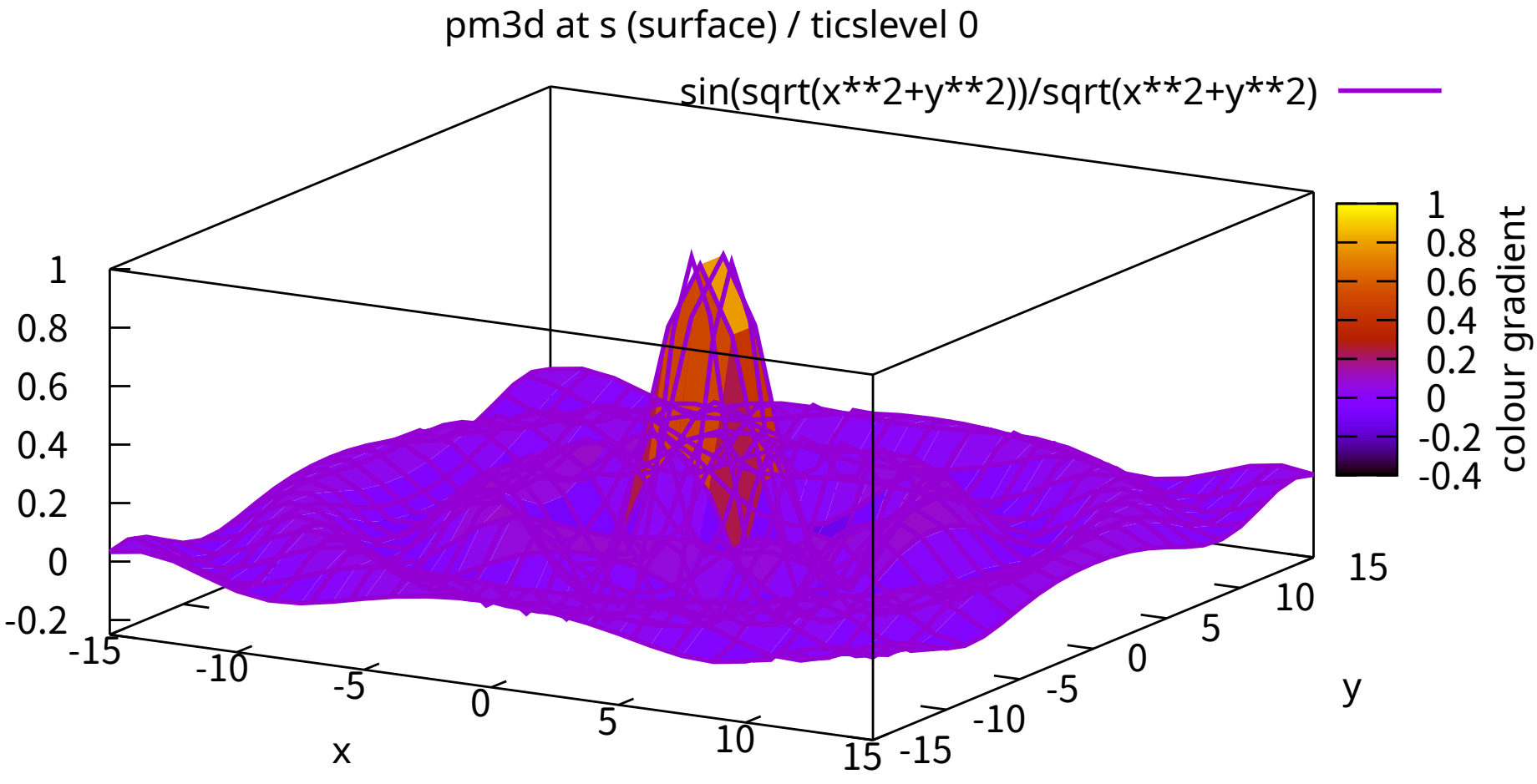


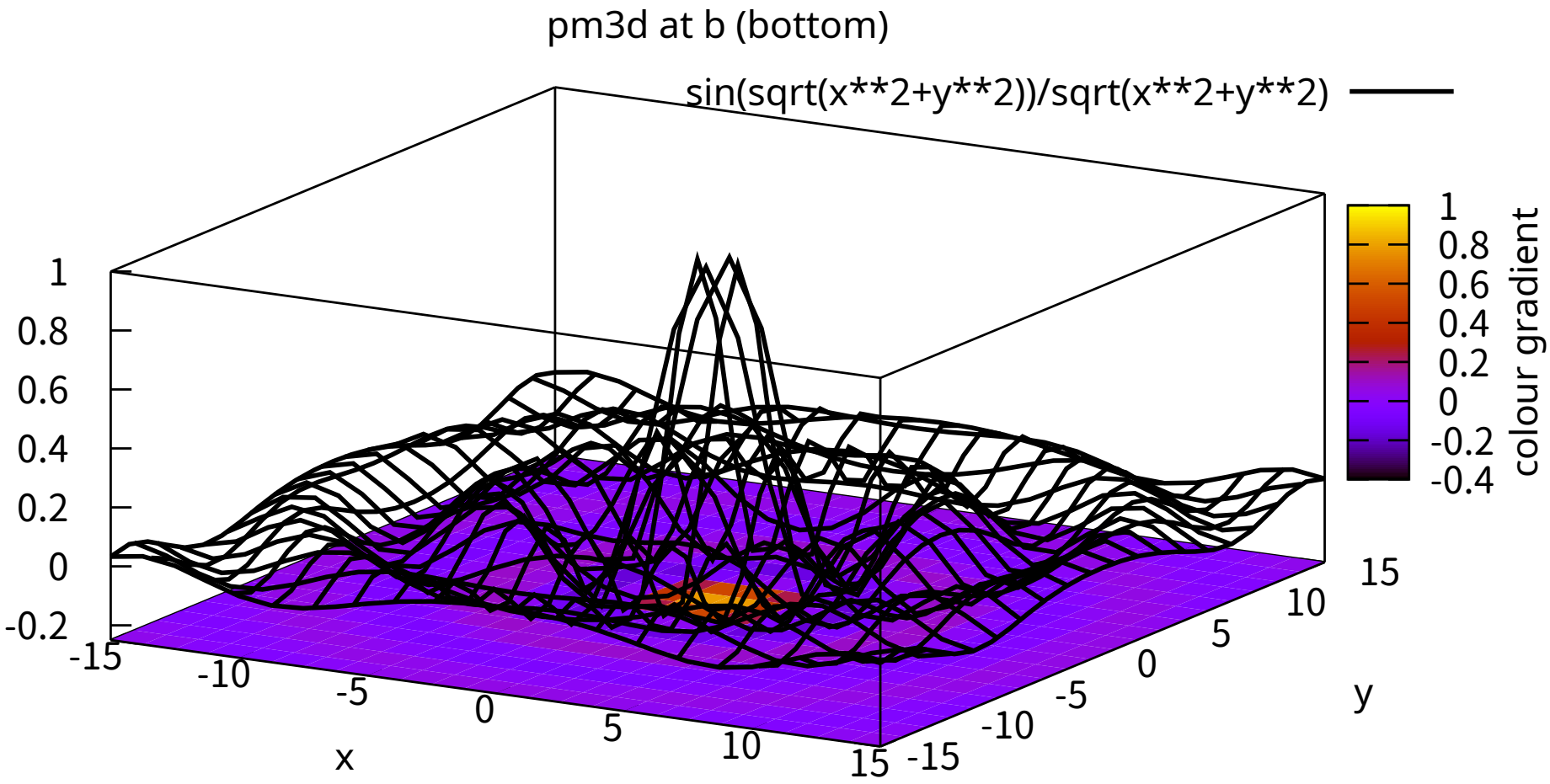
with yerrorbars

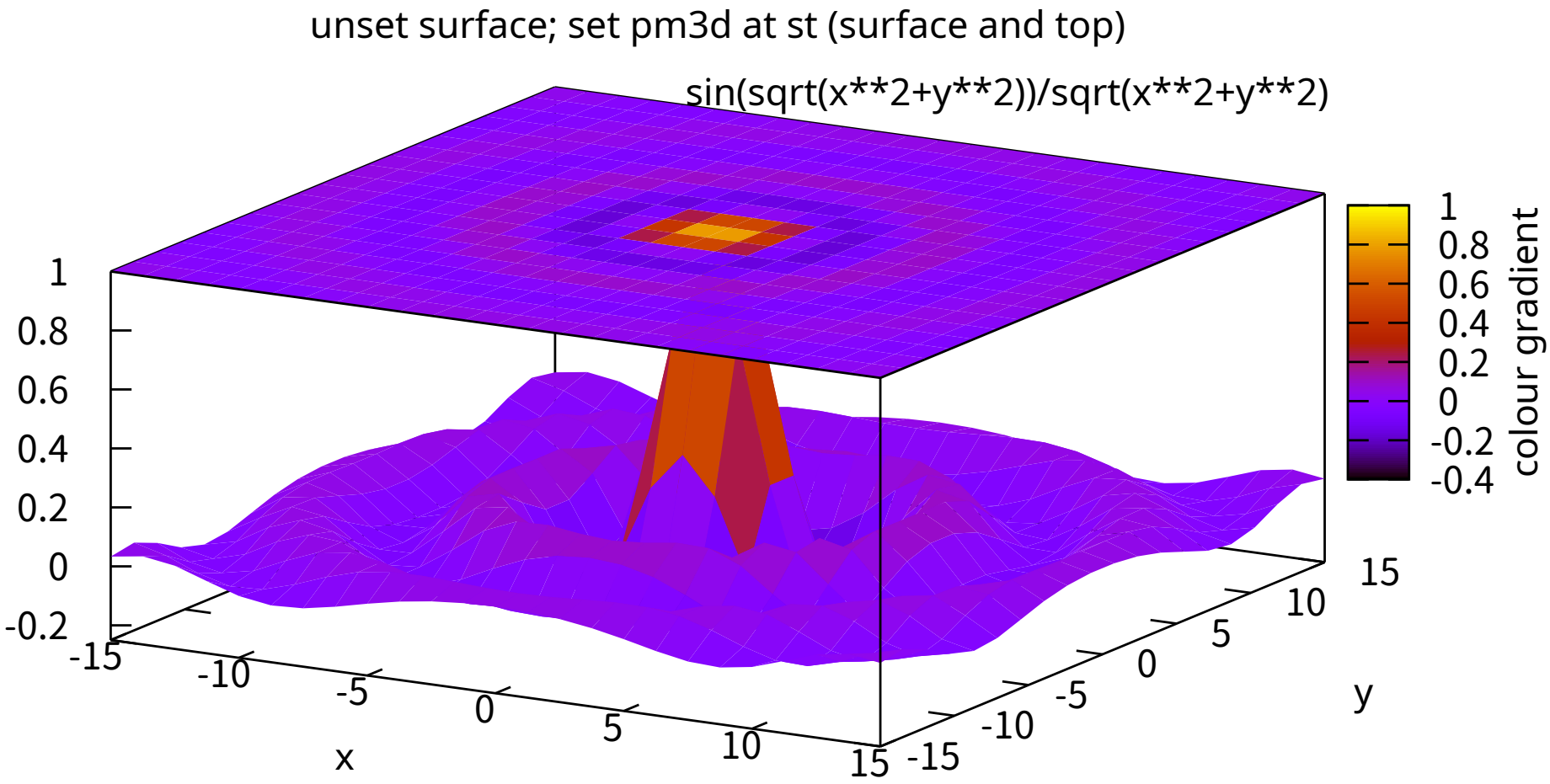

Column 1 (size) should increase from left to right
Column 2 (type) should decrease from left to right
Label
ps var
pt var
ps var pt var
3D ps var pt var

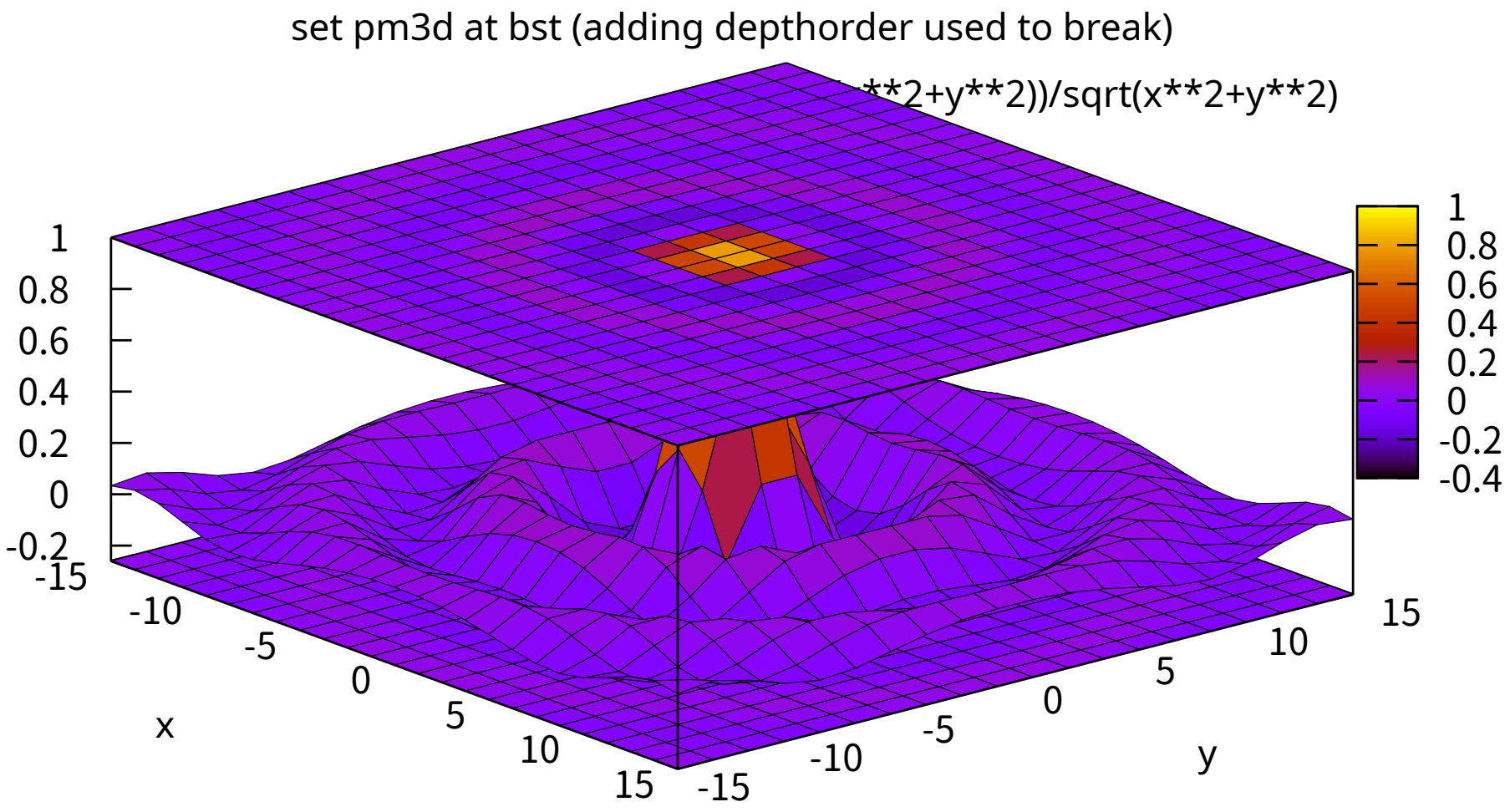
with labels

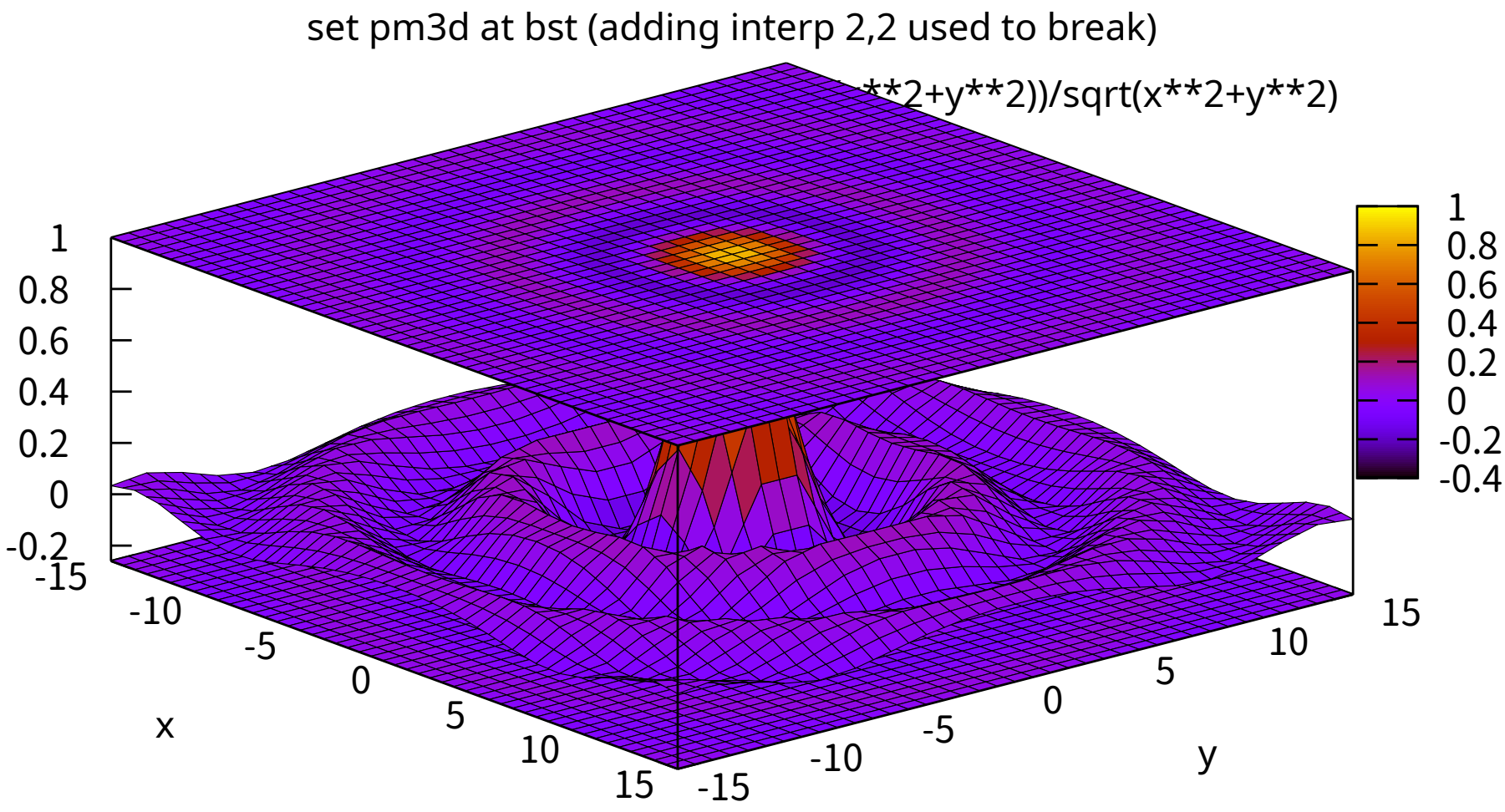

pm3d demo. Radial sinc function. Default options.
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
x
y

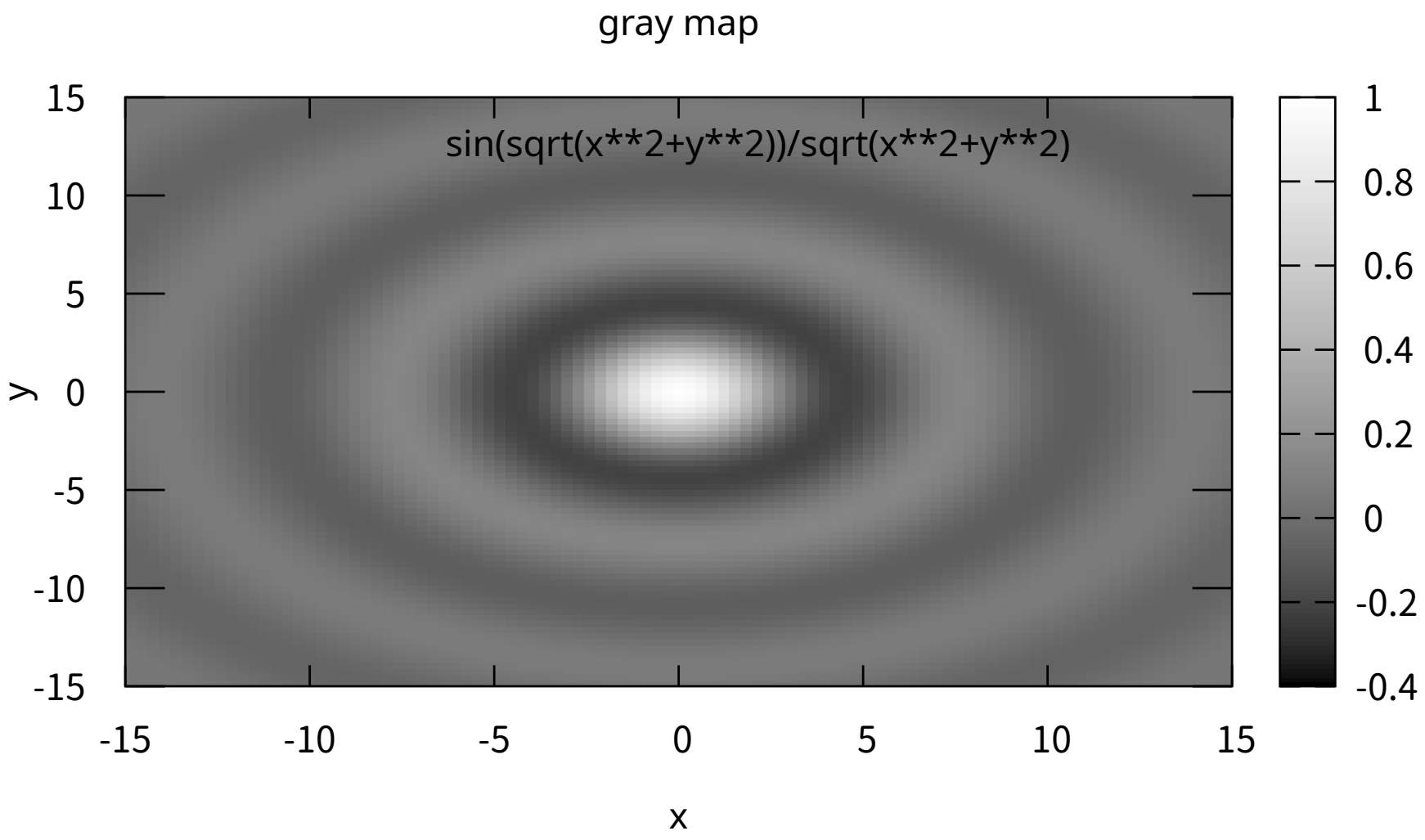

pm3d at s (surface) / ticslevel 0
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x
y
colour gradient
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

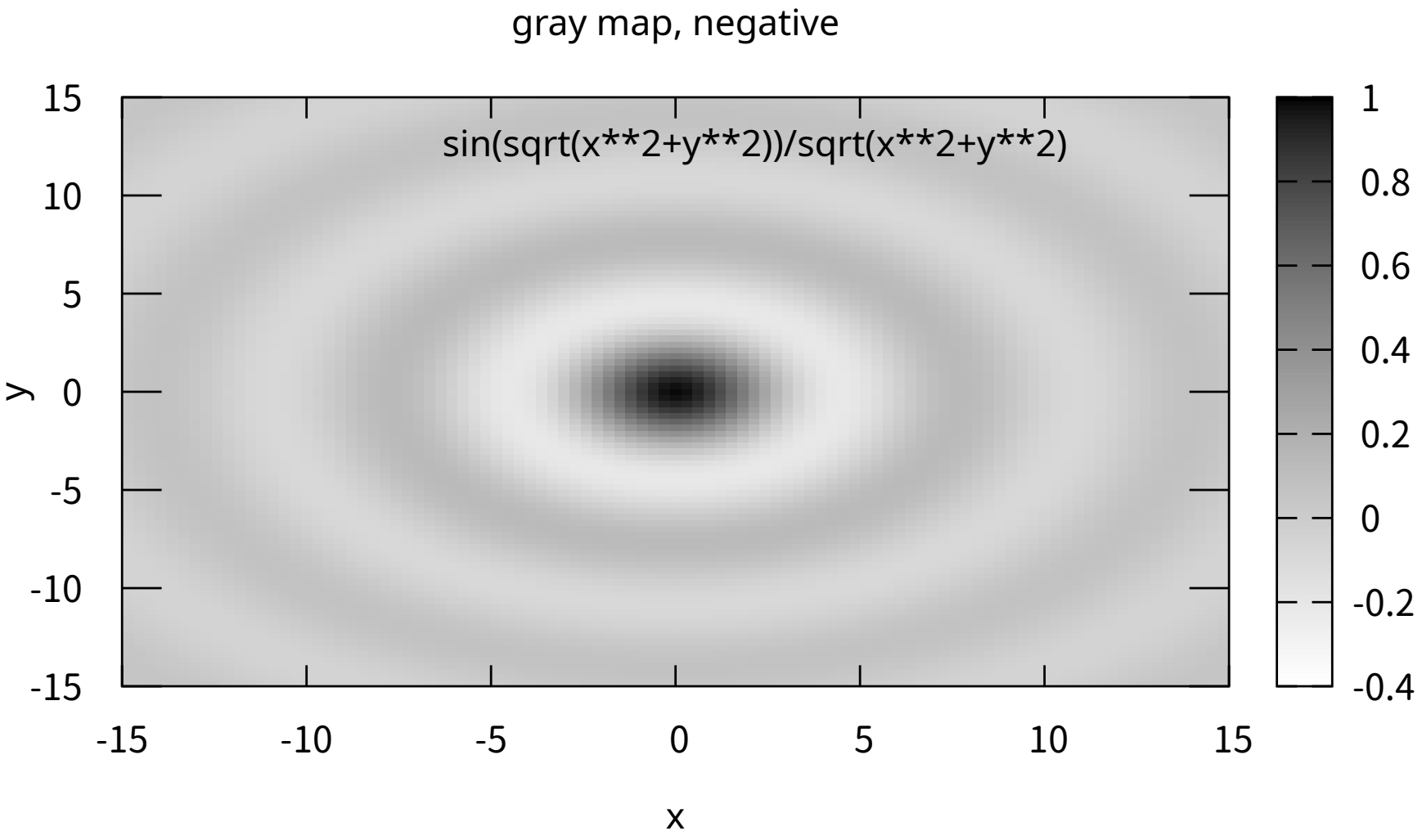

pm3d at b (bottom)
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x
-15
-10
-5
0
5
10
15
y
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
colour gradient


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
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
colour gradient

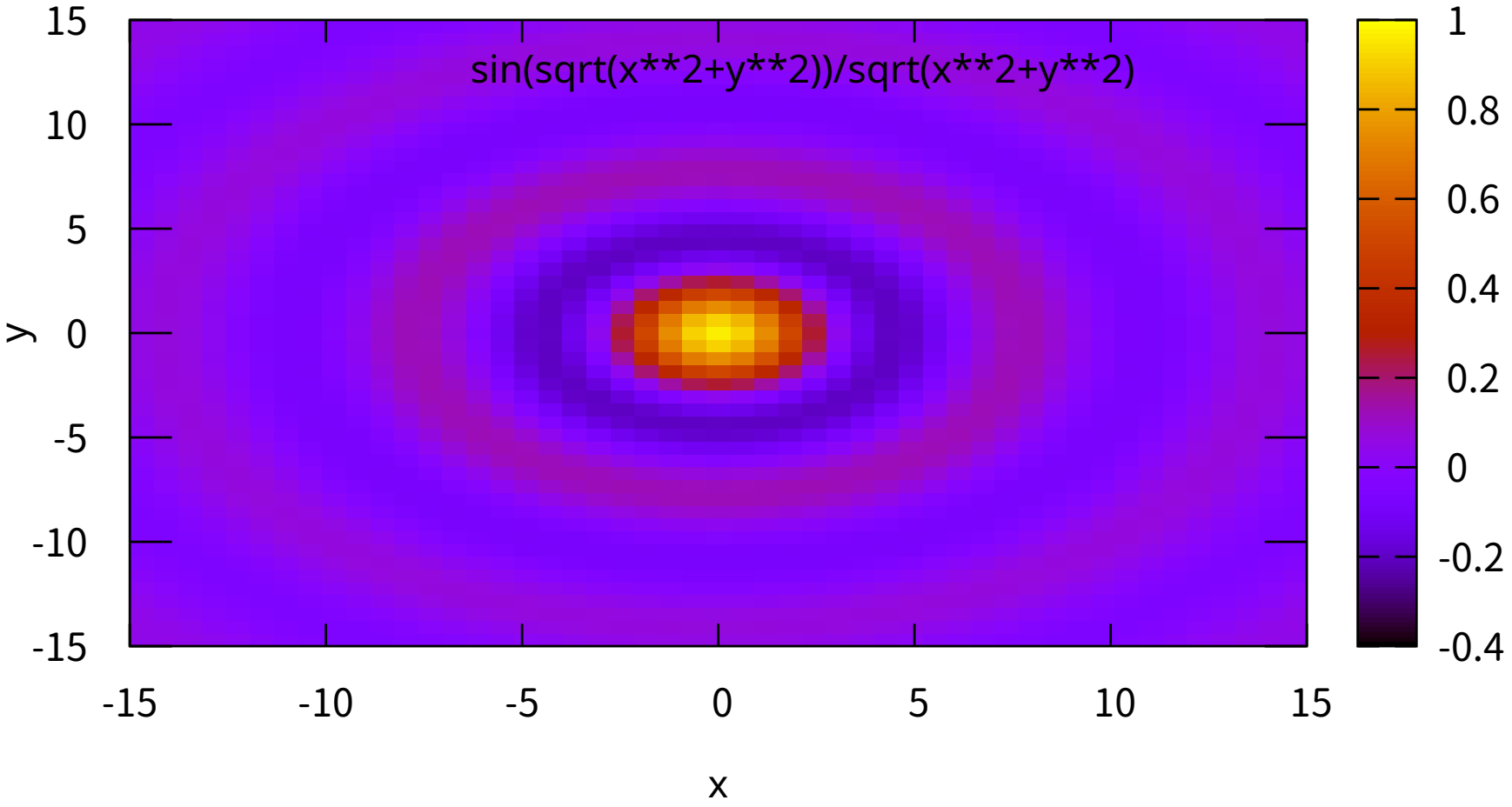

set pm3d at bst (adding depthorder used to break)
2+y2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-15
-10
-5
0
5
10
15
x
y

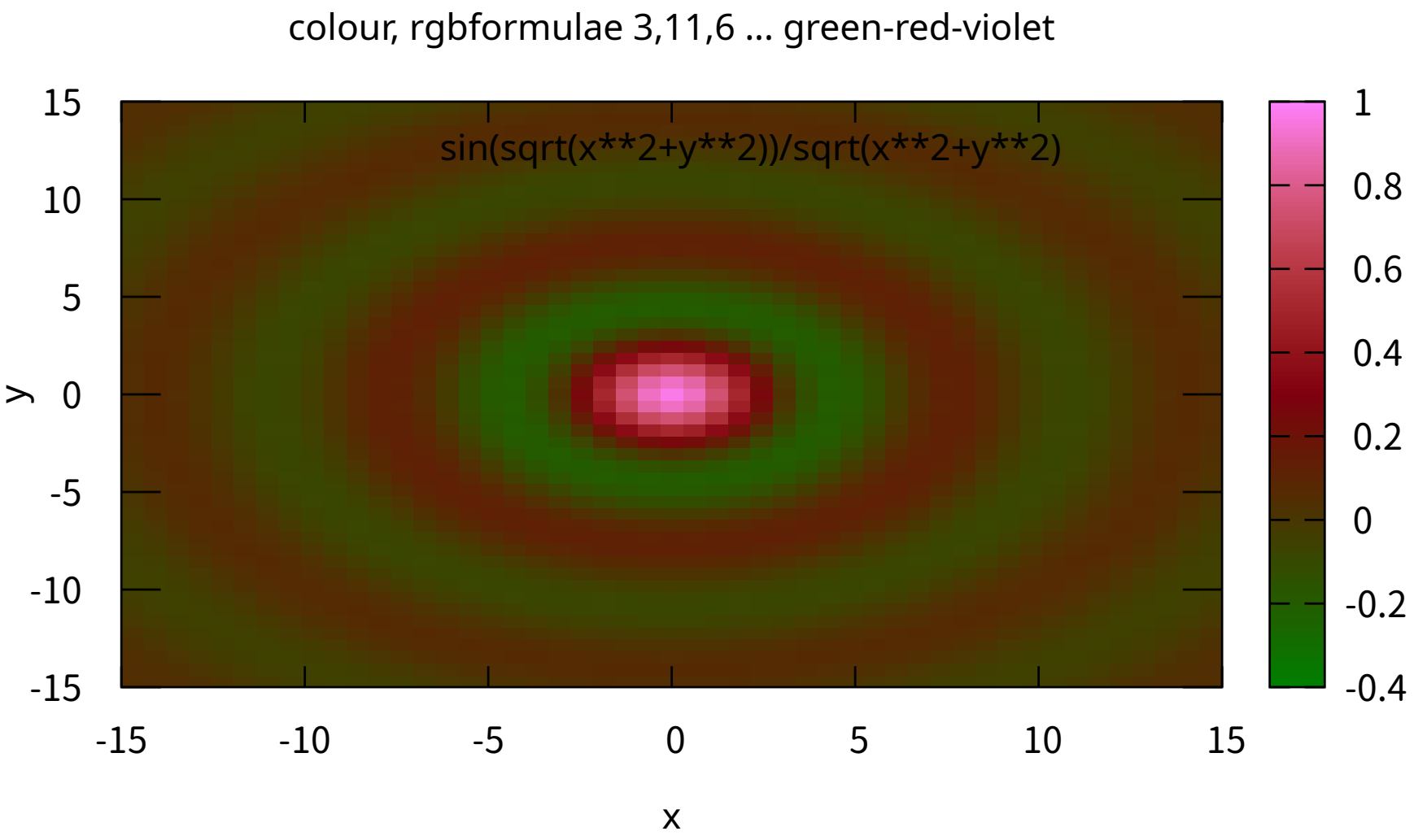

set pm3d at bst (adding interp 2,2 used to break)
2+y2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-15
-10
-5
0
5
10
15
x
y


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

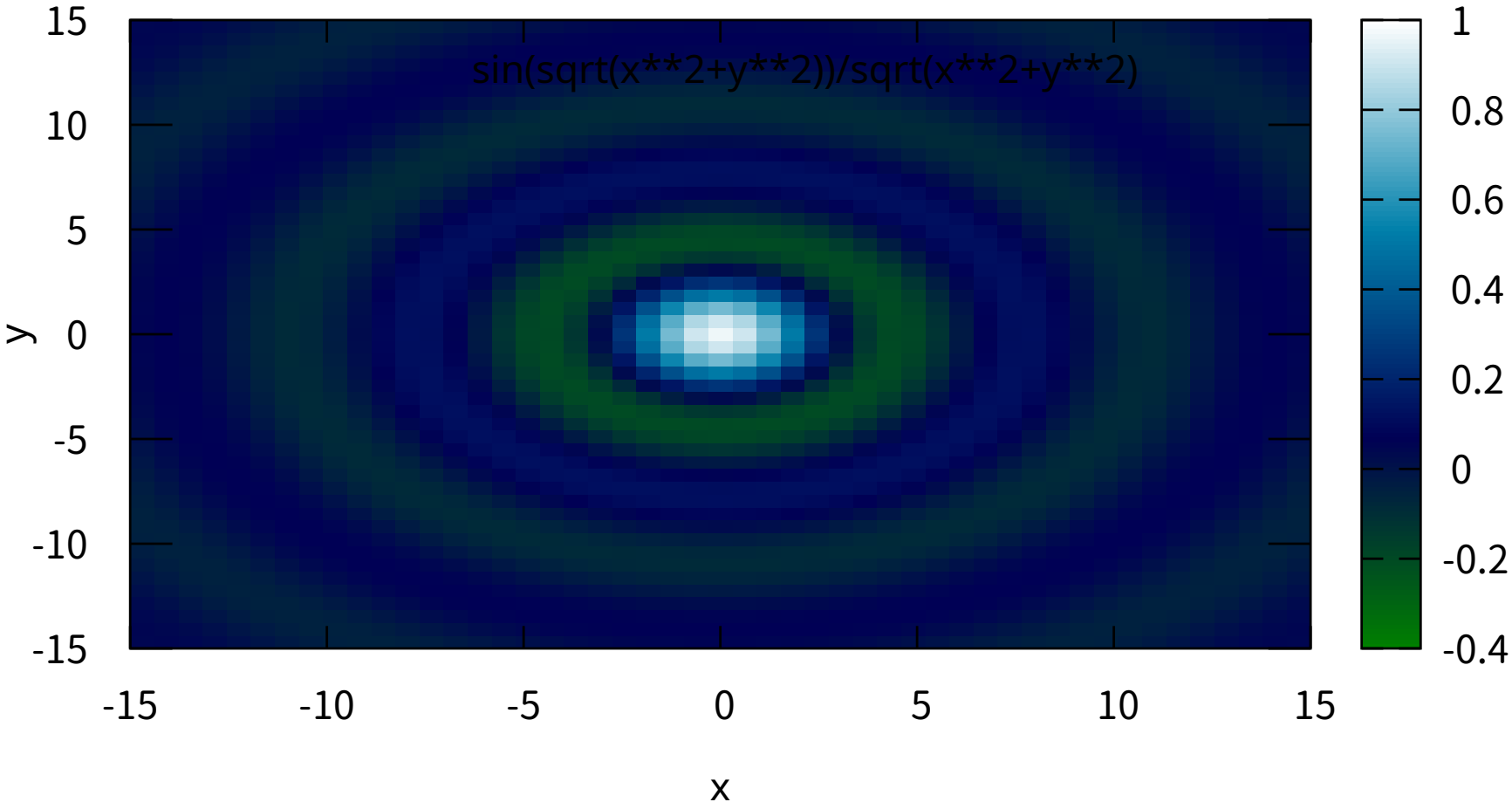

gray map, negative
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

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

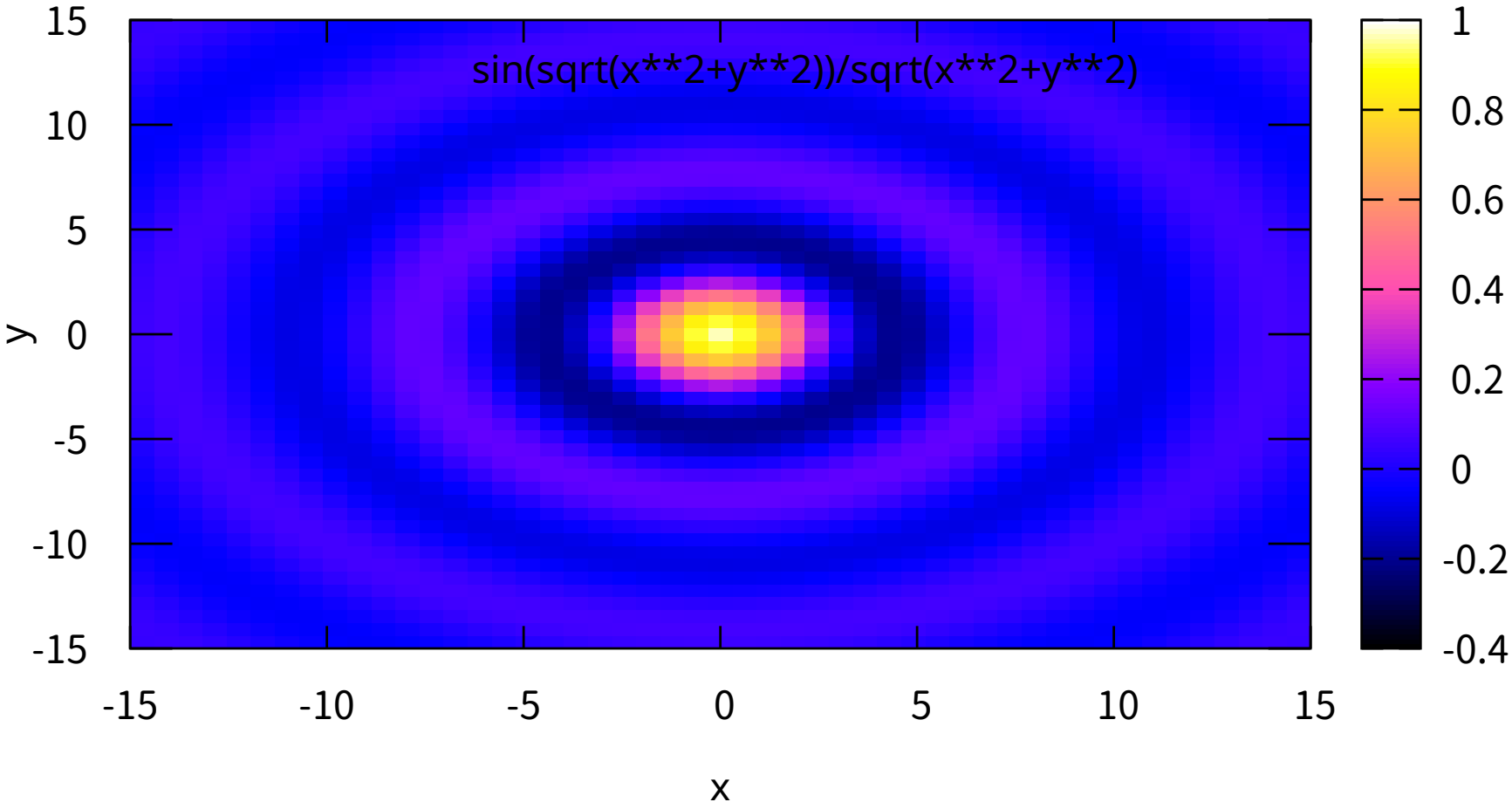


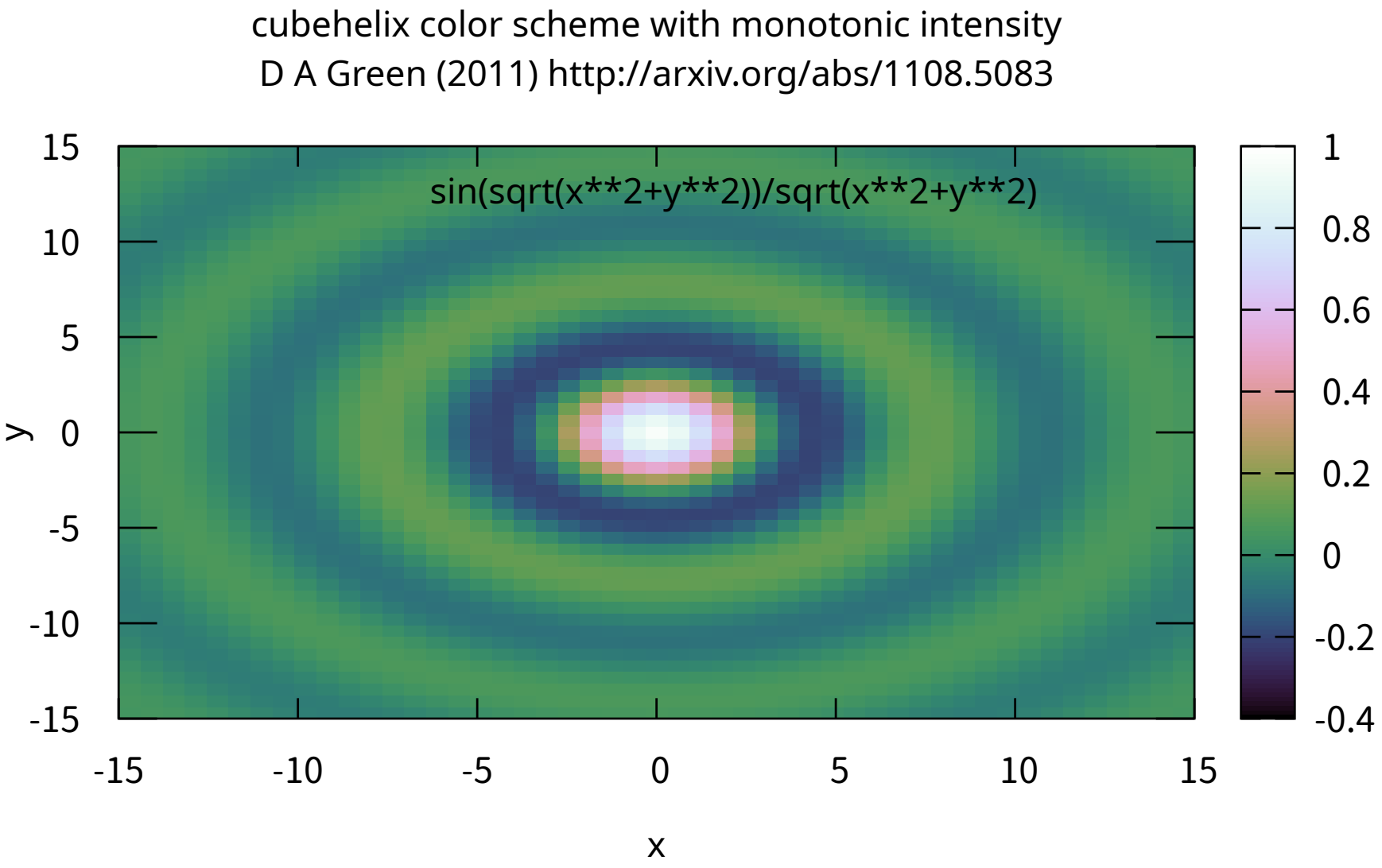

colour, rgbformulae 3,11,6 ... green-red-violet
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

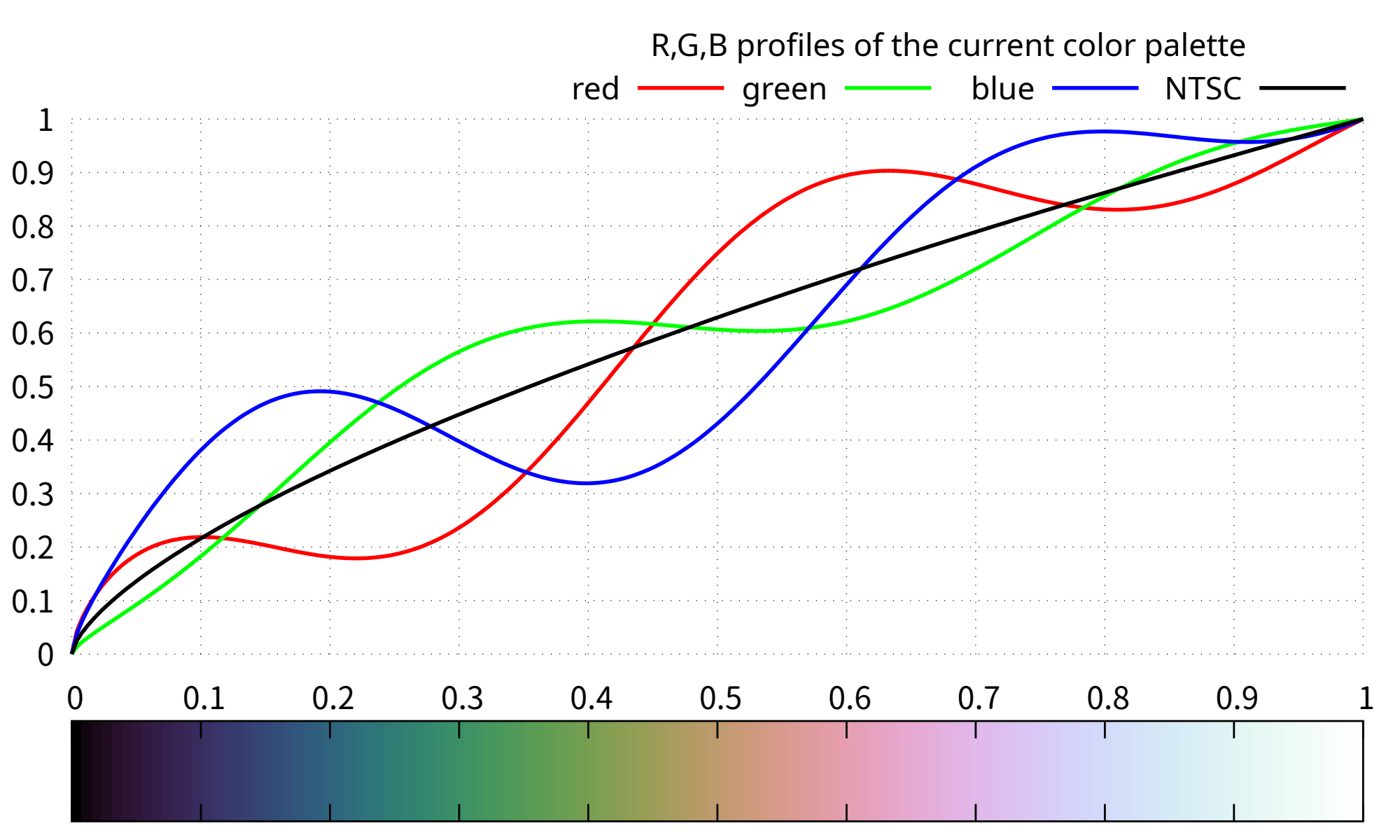
lour, rgbformulae 23,28,3 ... ocean (green-blue-white); OK are also all other permutations



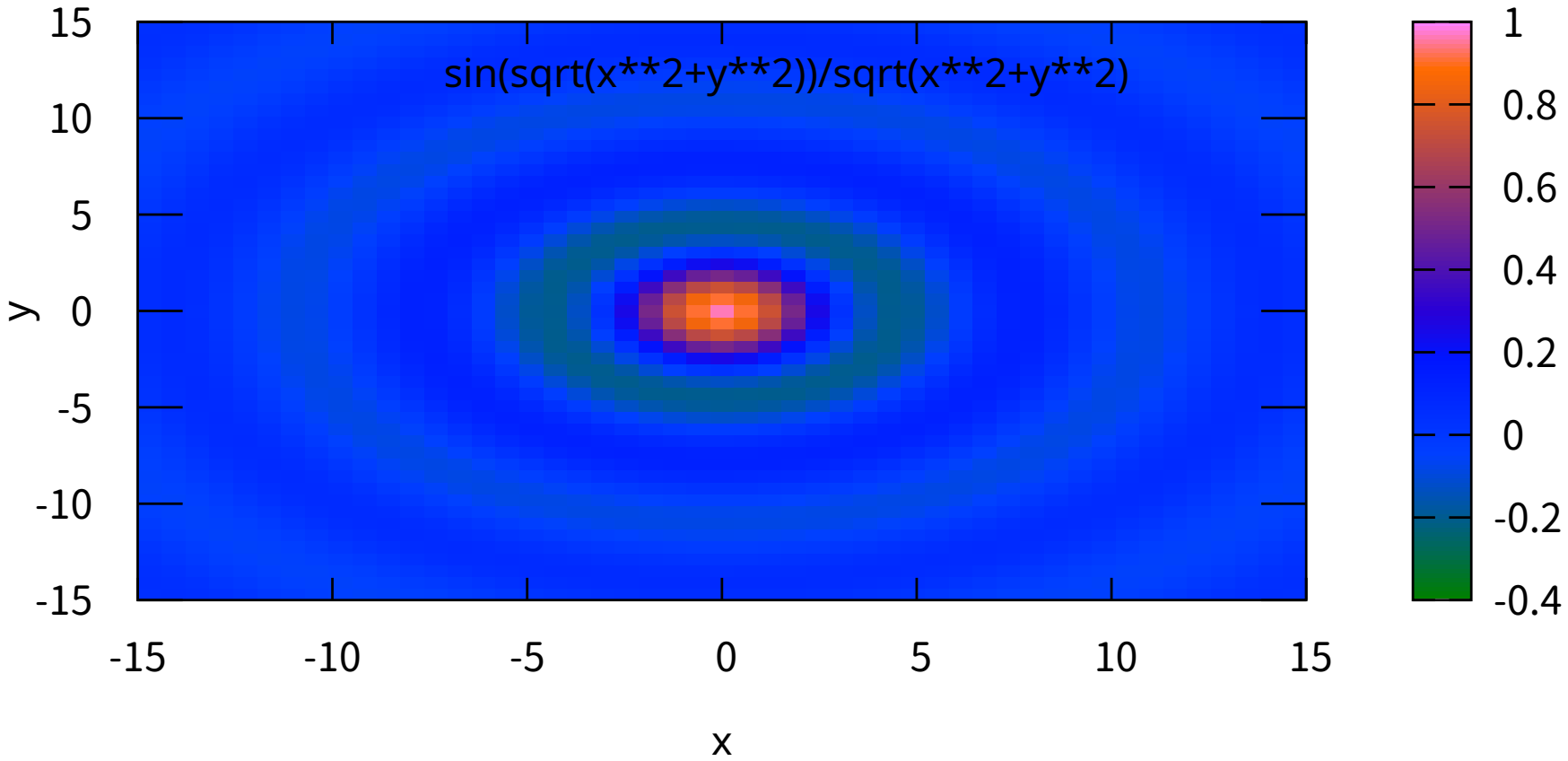
colour, rgbformulae 30,31,32 ... color printable on gray (black-blue-violet-yellow-white)



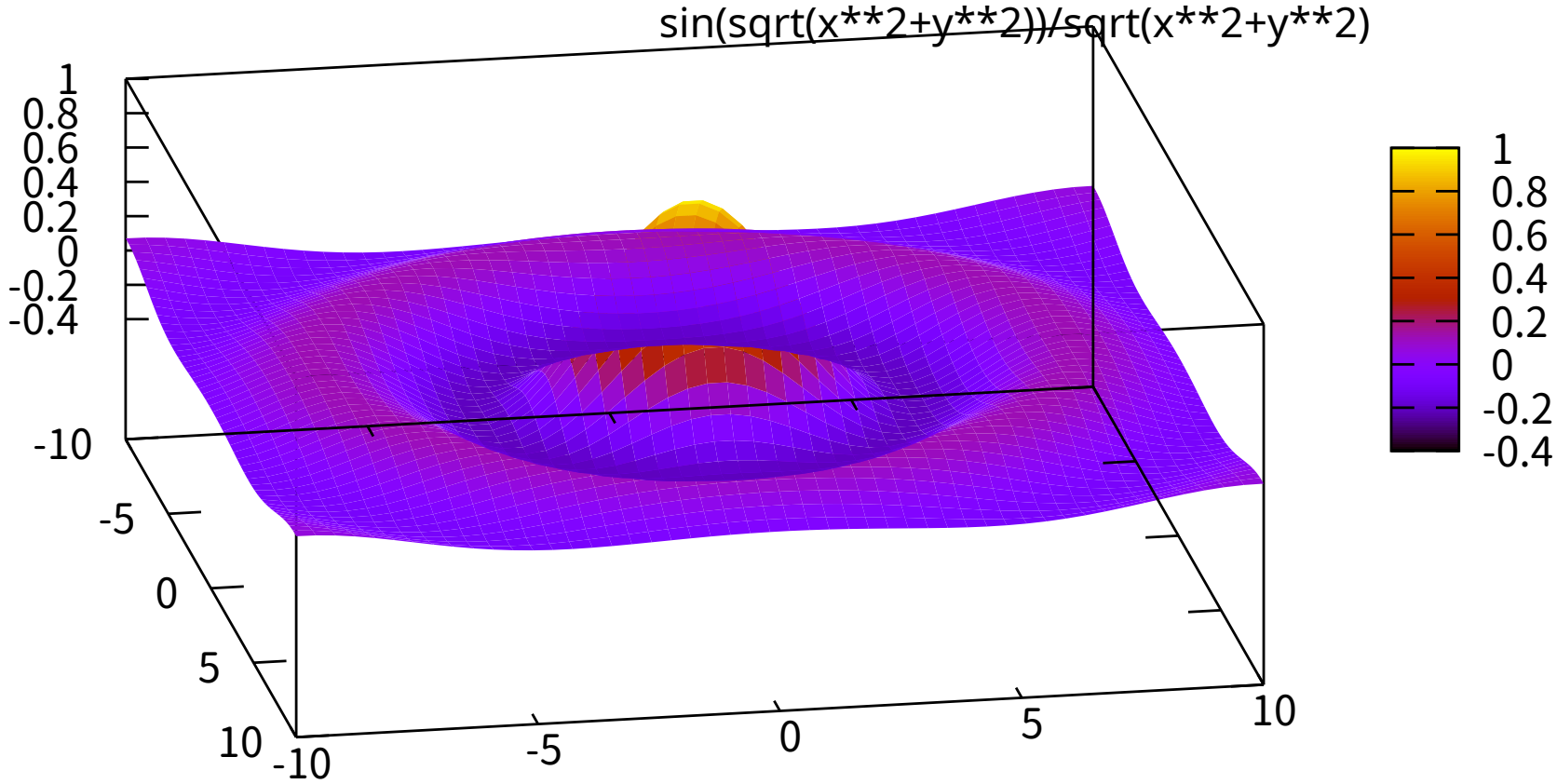

cubehelix color scheme with monotonic intensity
D A Green (2011) http://arxiv.org/abs/1108.5083
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4


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

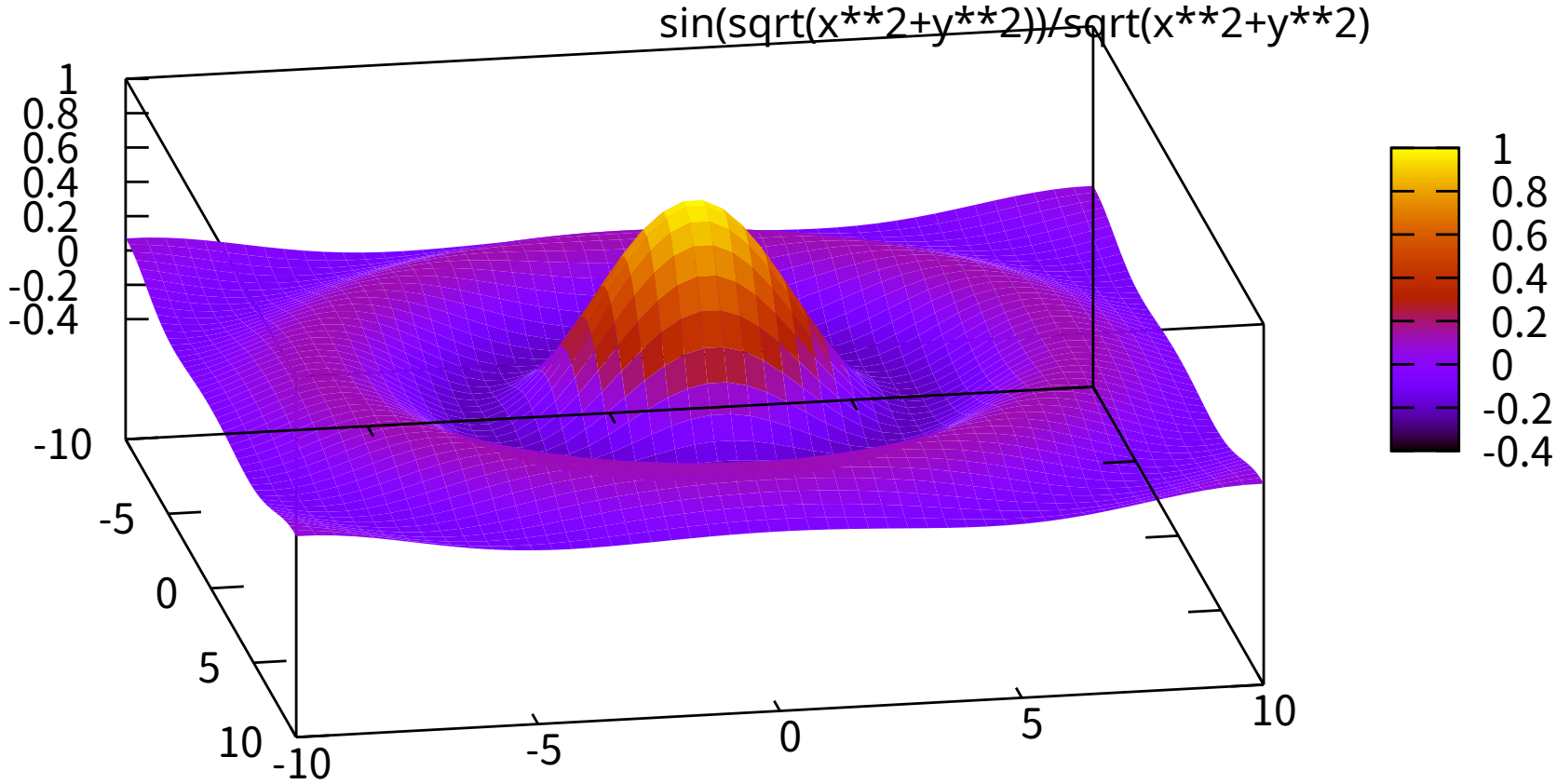
rgbformulae 31,-11,32: negative formula number=inverted color

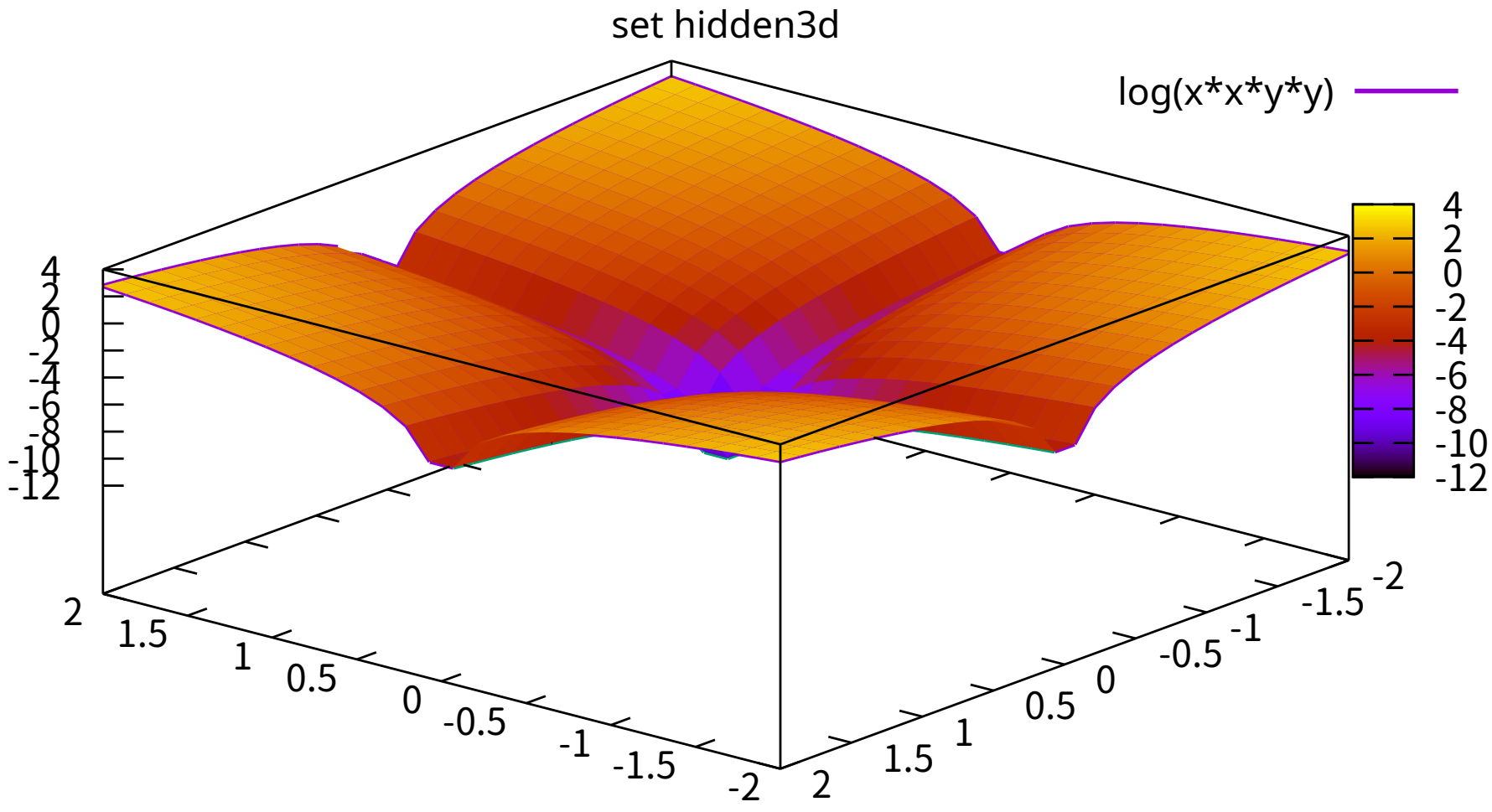


set pm3d scansforward: wrong, because back overwrites front

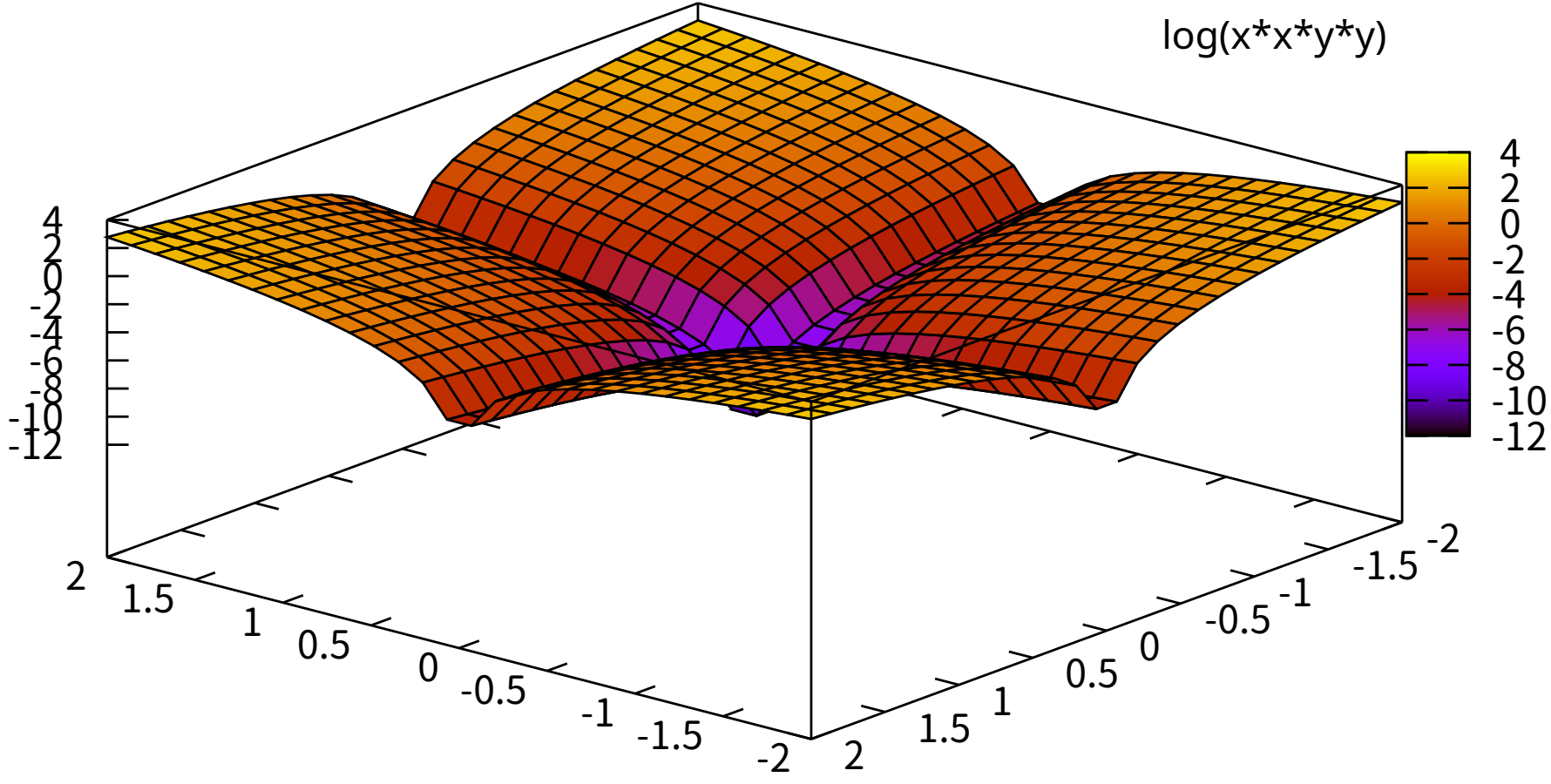


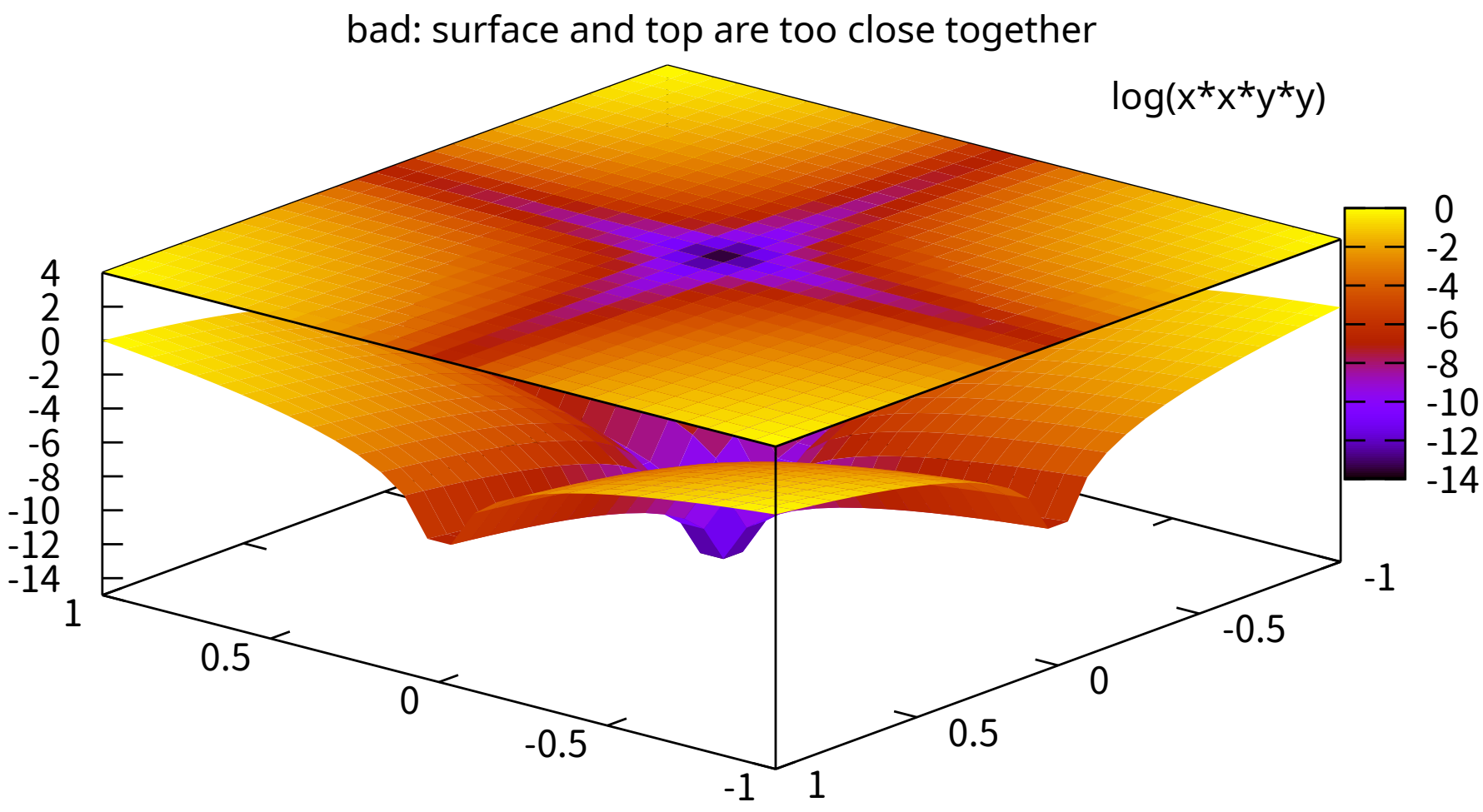
set pm3d scansbackward: correctly looking surface

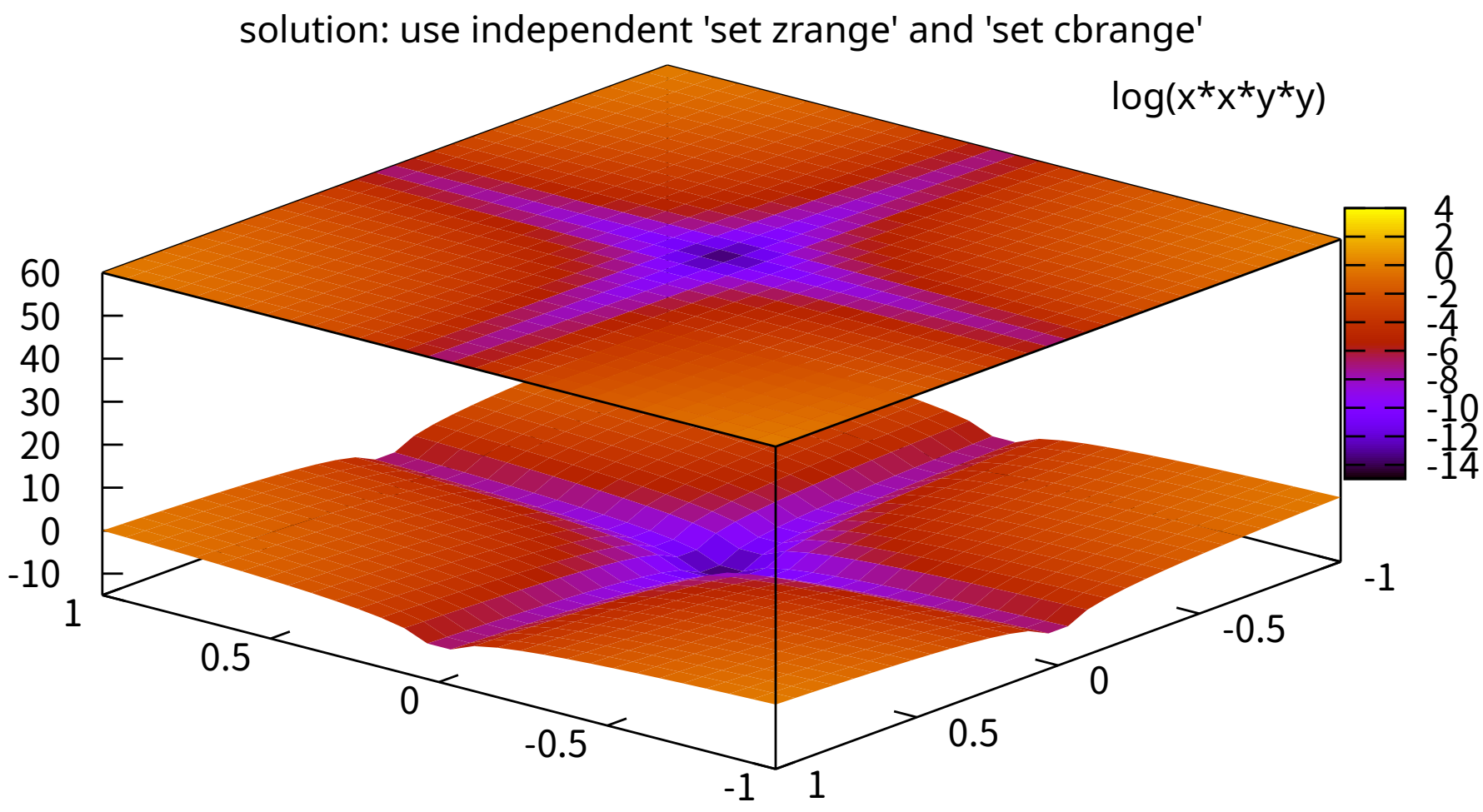


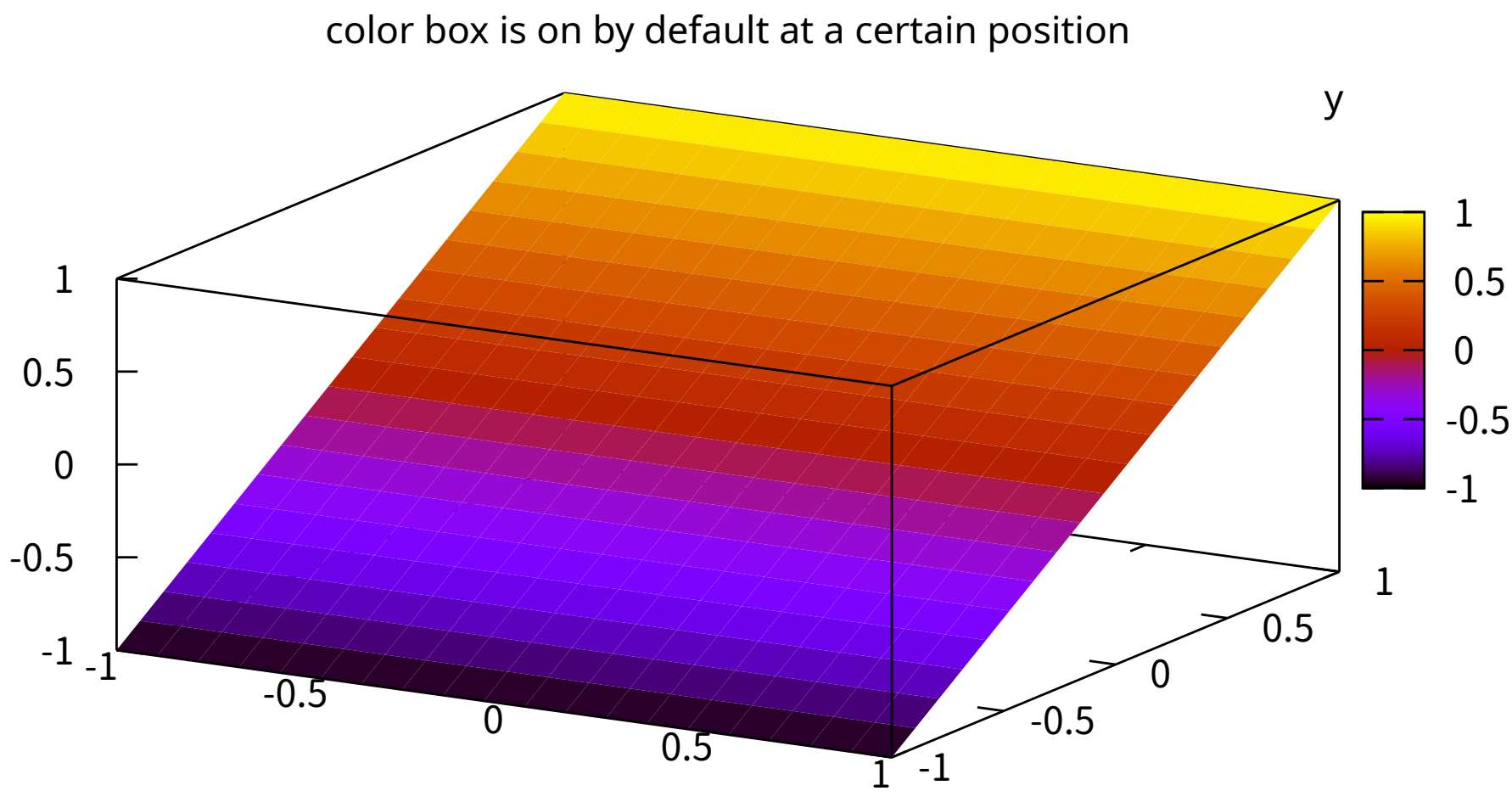

set hidden3d
log(x*x*y*y)
4
2
0
-2
-4
-6
-8
-10
-12
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
4
2
0
-2
-4
-6
-8
-10
-12

set pm3d depthorder border lc 'black' lw 1

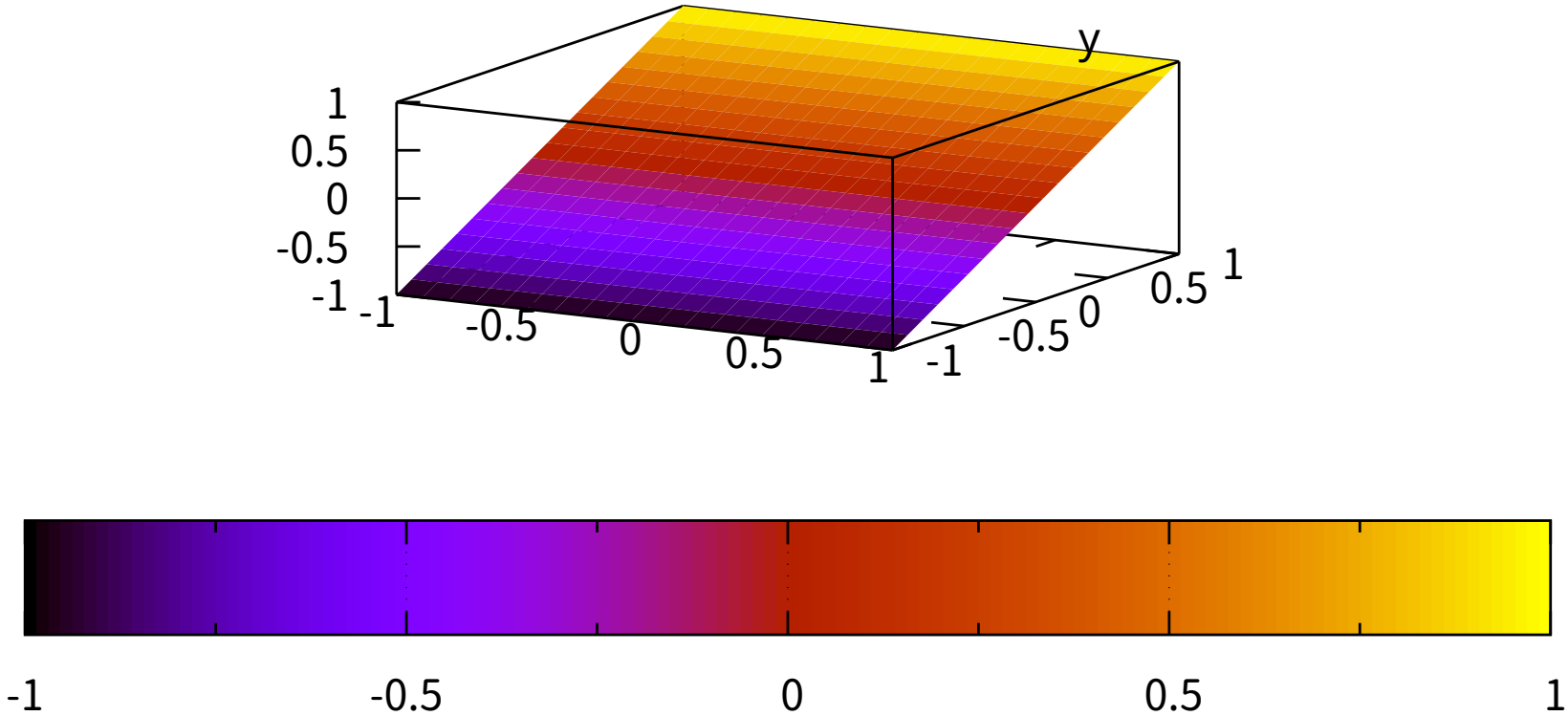



bad: surface and top are too close together
log(x*x*y*y)
4
2
0
-2
-4
-6
-8
-10
-12
-14
1
0.5
0
-0.5
-1
1
0.5
0
-0.5
-1
0
-2
-4
-6
-8
-10
-12
-14

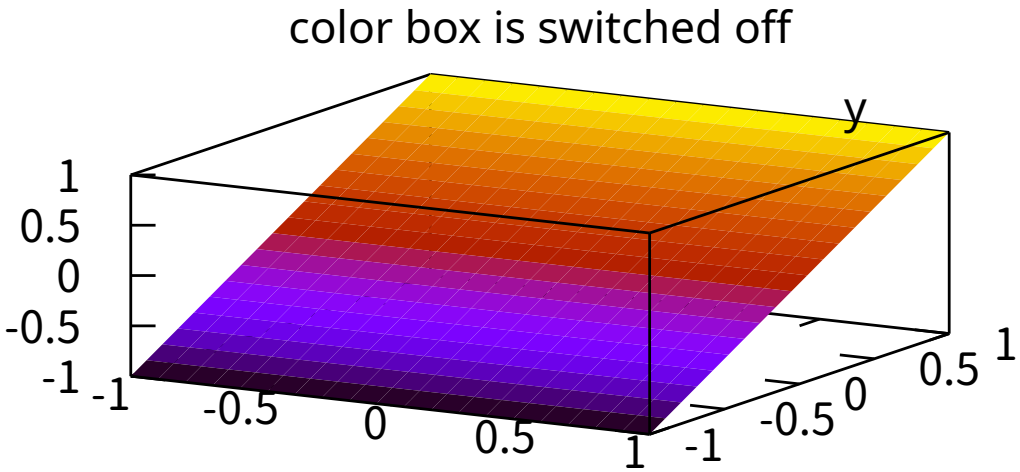

solution: use independent 'set zrange' and 'set cbrange'
log(x*x*y*y)
60
50
40
30
20
10
0
-10
1
0.5
0
-0.5
-1
1
0.5
0
-0.5
-1
4
2
0
-2
-4
-6
-8
-10
-12
-14


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

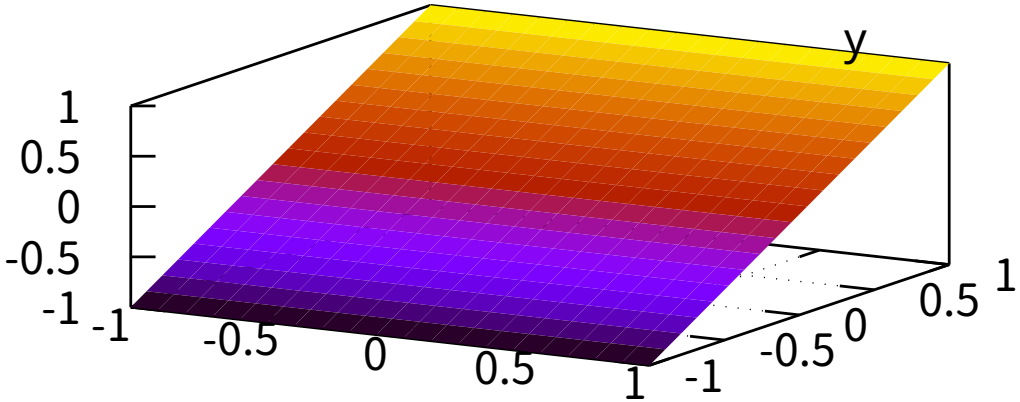
color box is on again, now with horizontal gradient

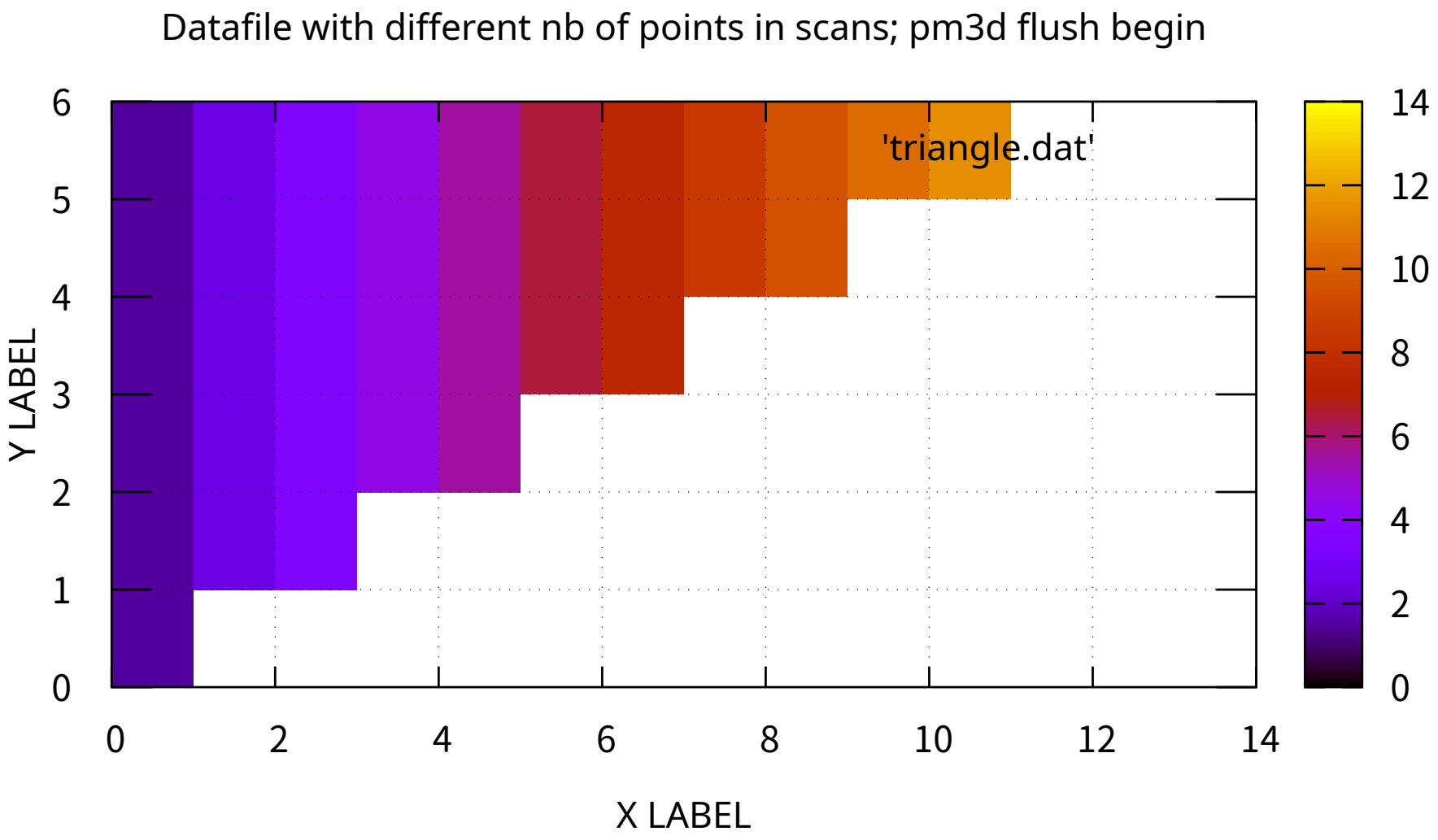


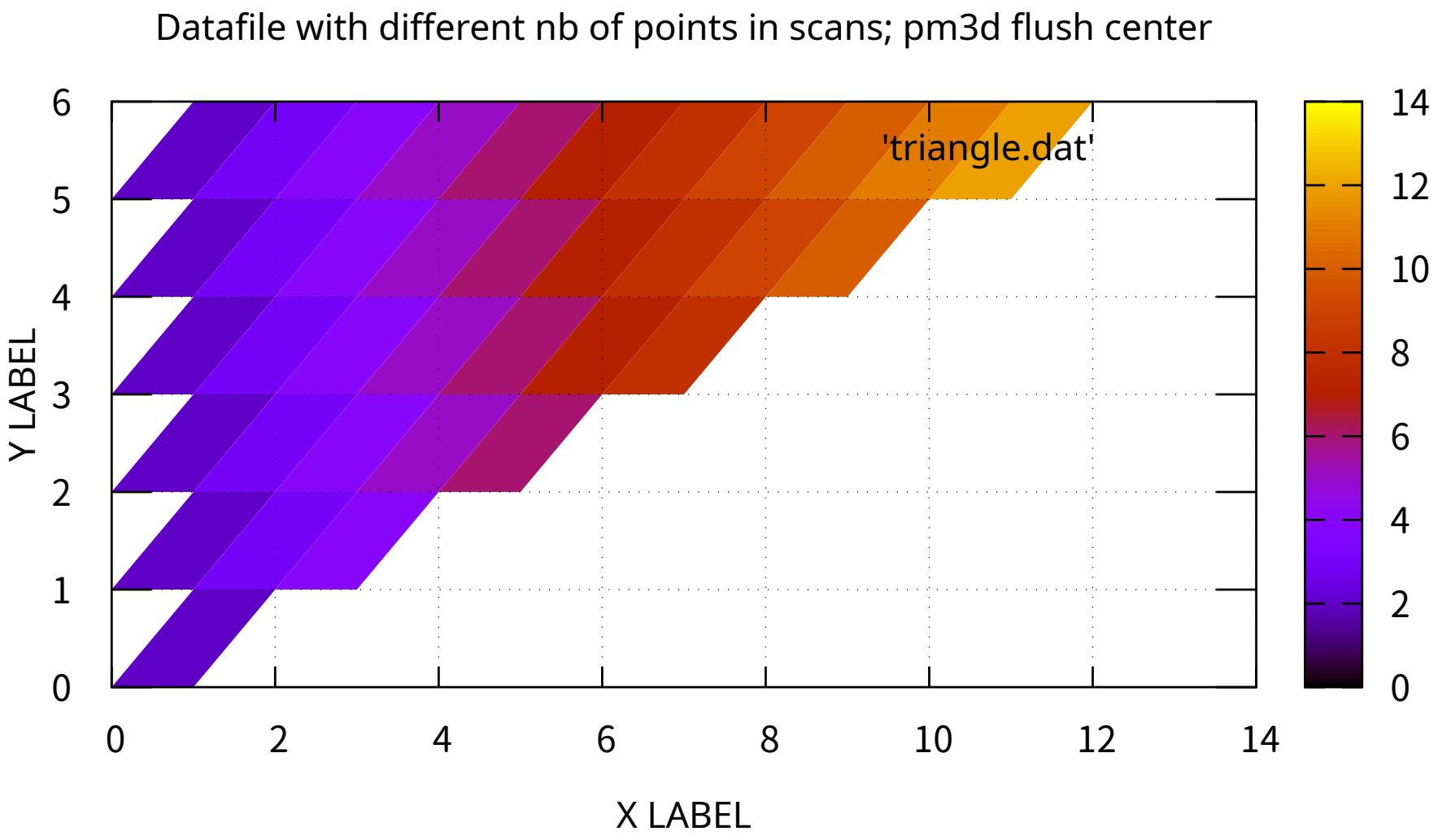
see cblabel, grid cb, mcbtics, ...

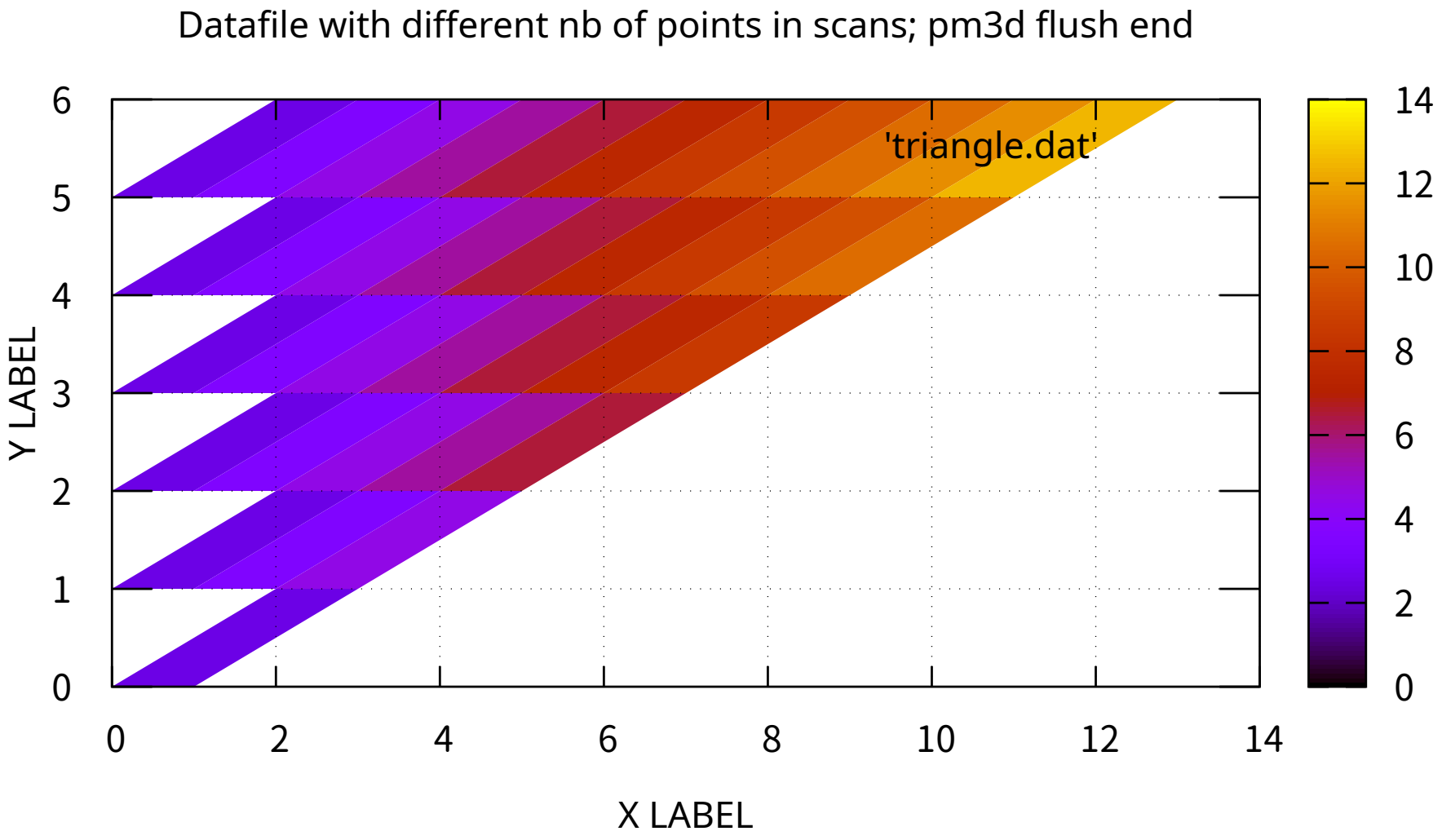

color box is switched off
y
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1

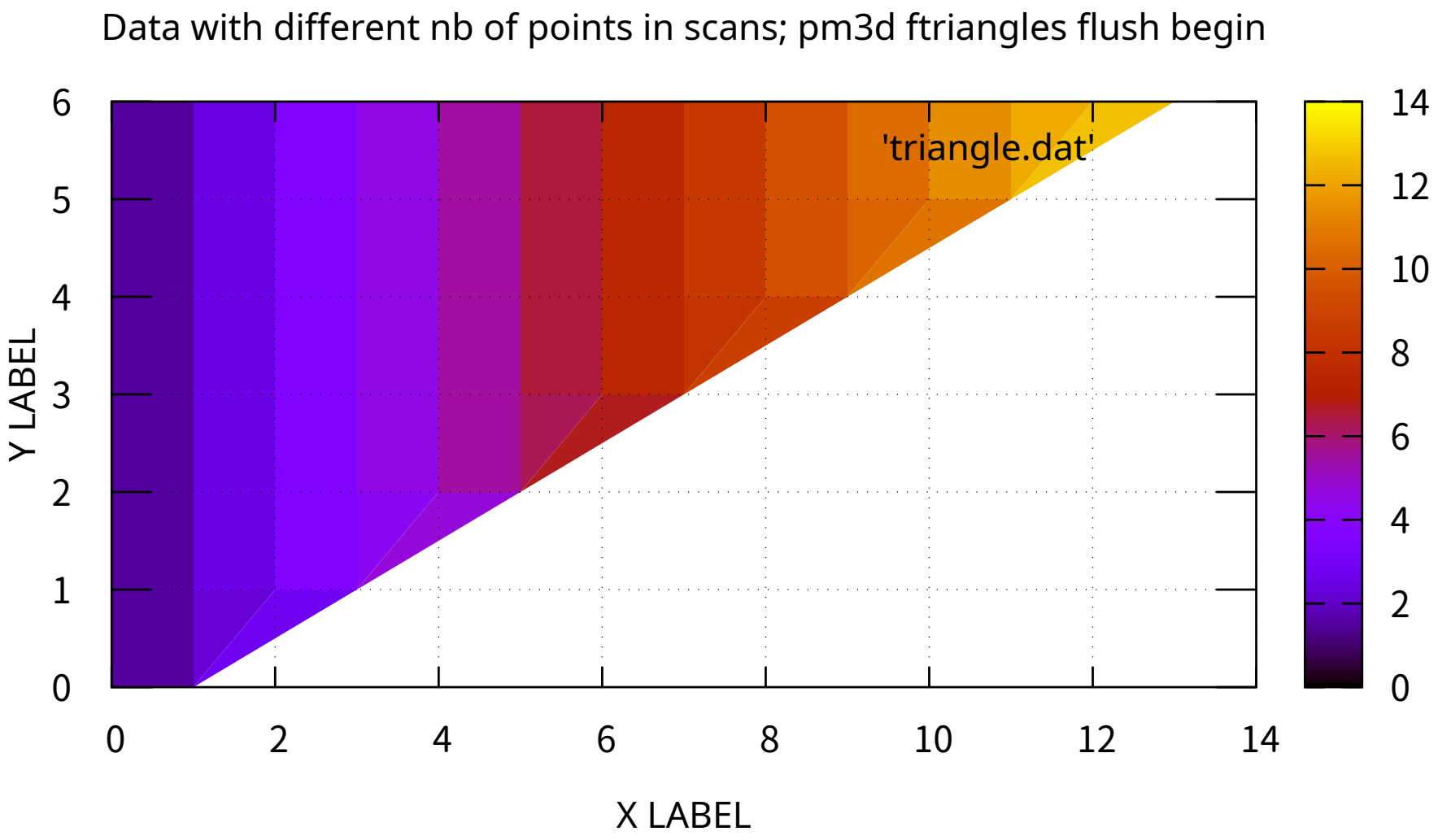
using now "set grid back; unset colorbox"

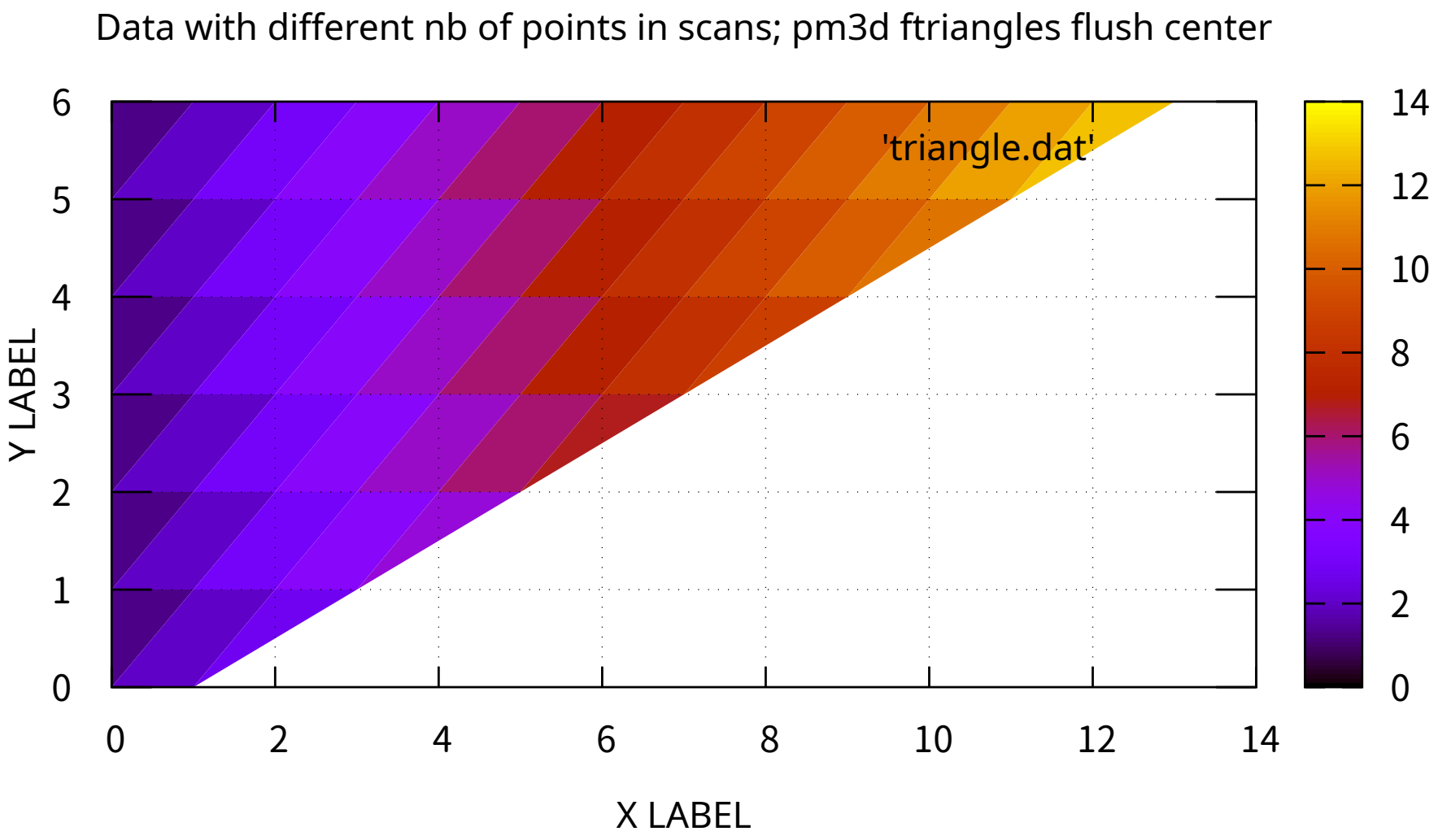


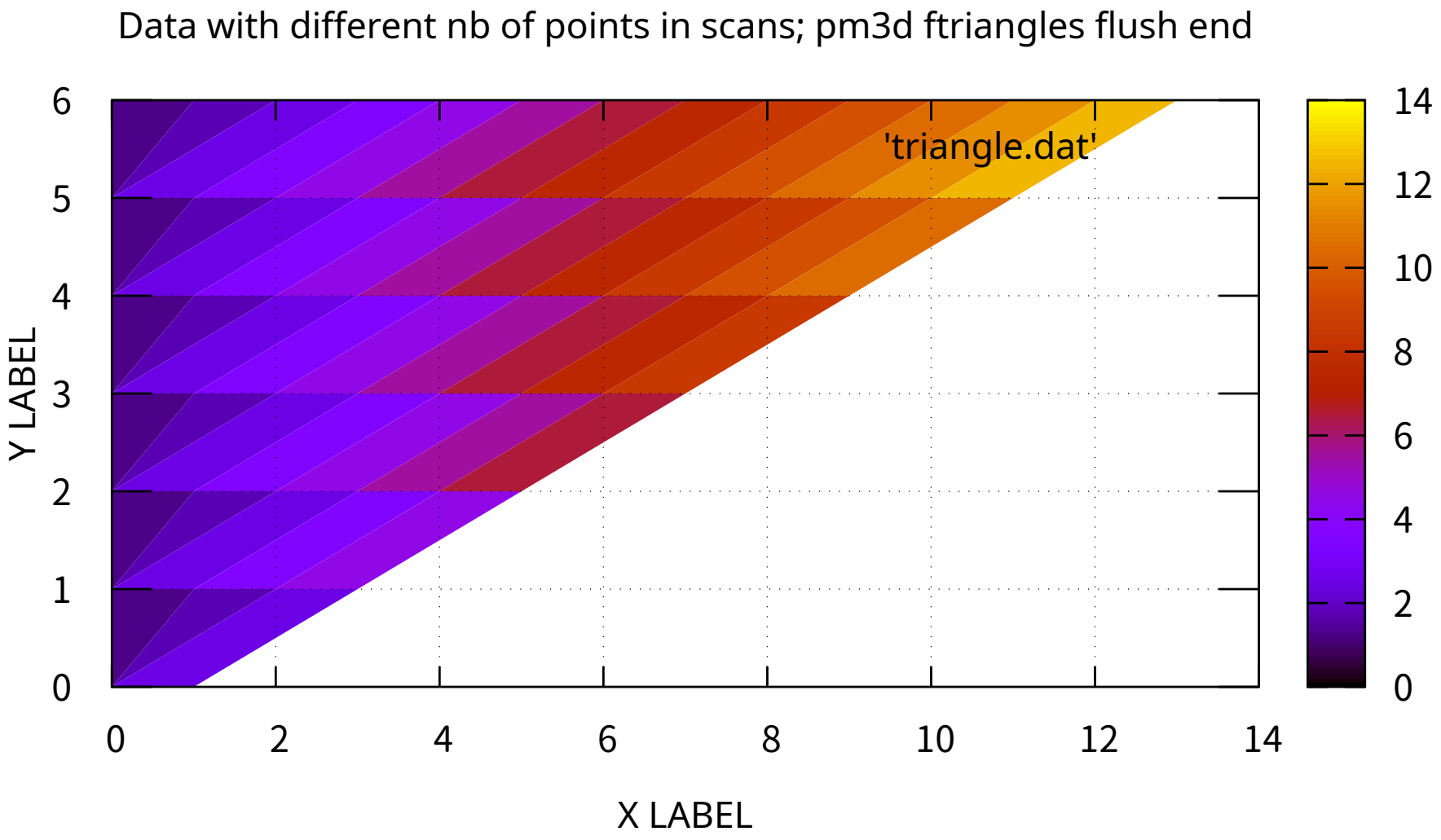

Datafile with different nb of points in scans; pm3d flush begin
'triangle.dat'
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

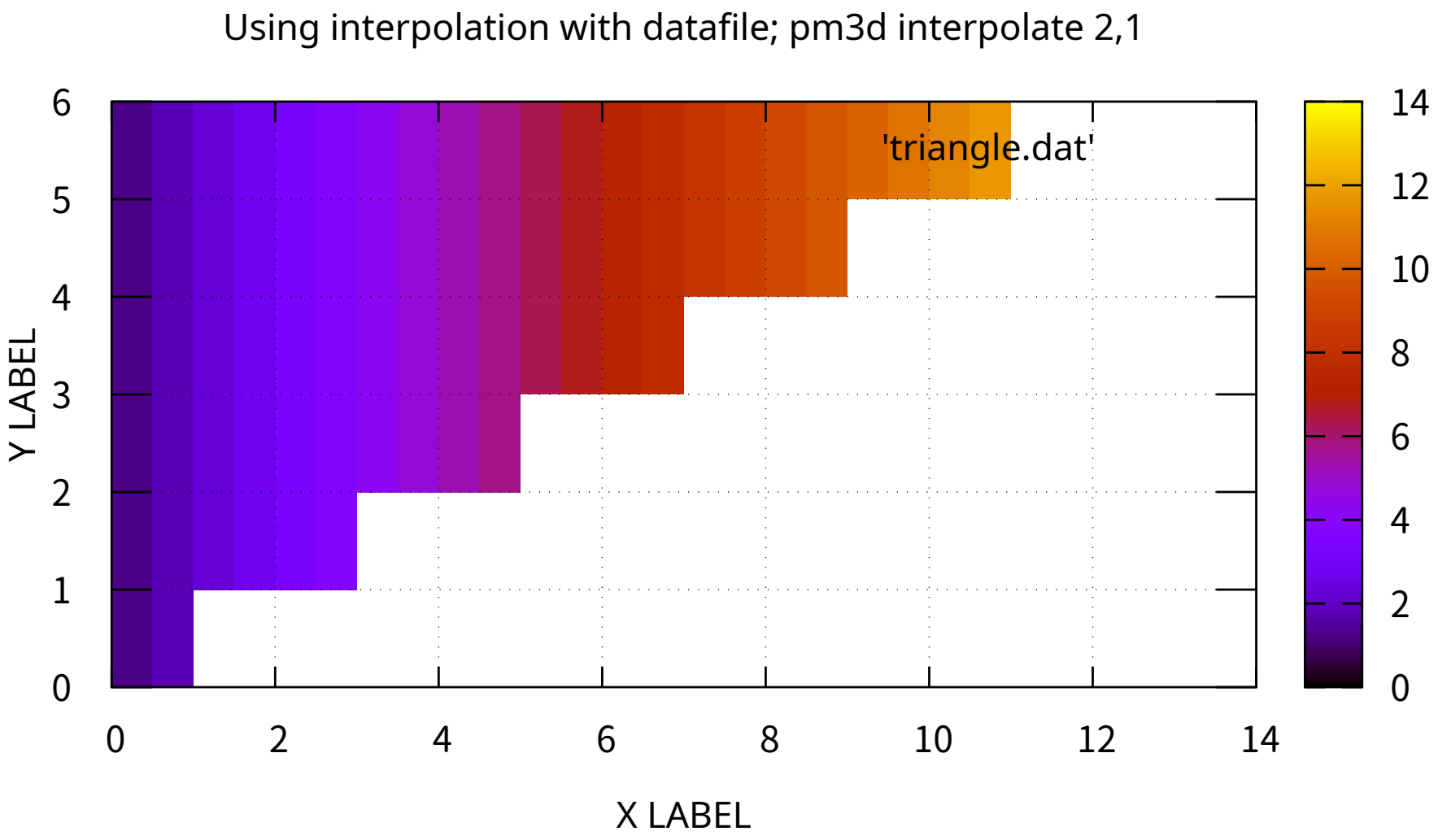

Datafile with different nb of points in scans; pm3d flush center
'triangle.dat'
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

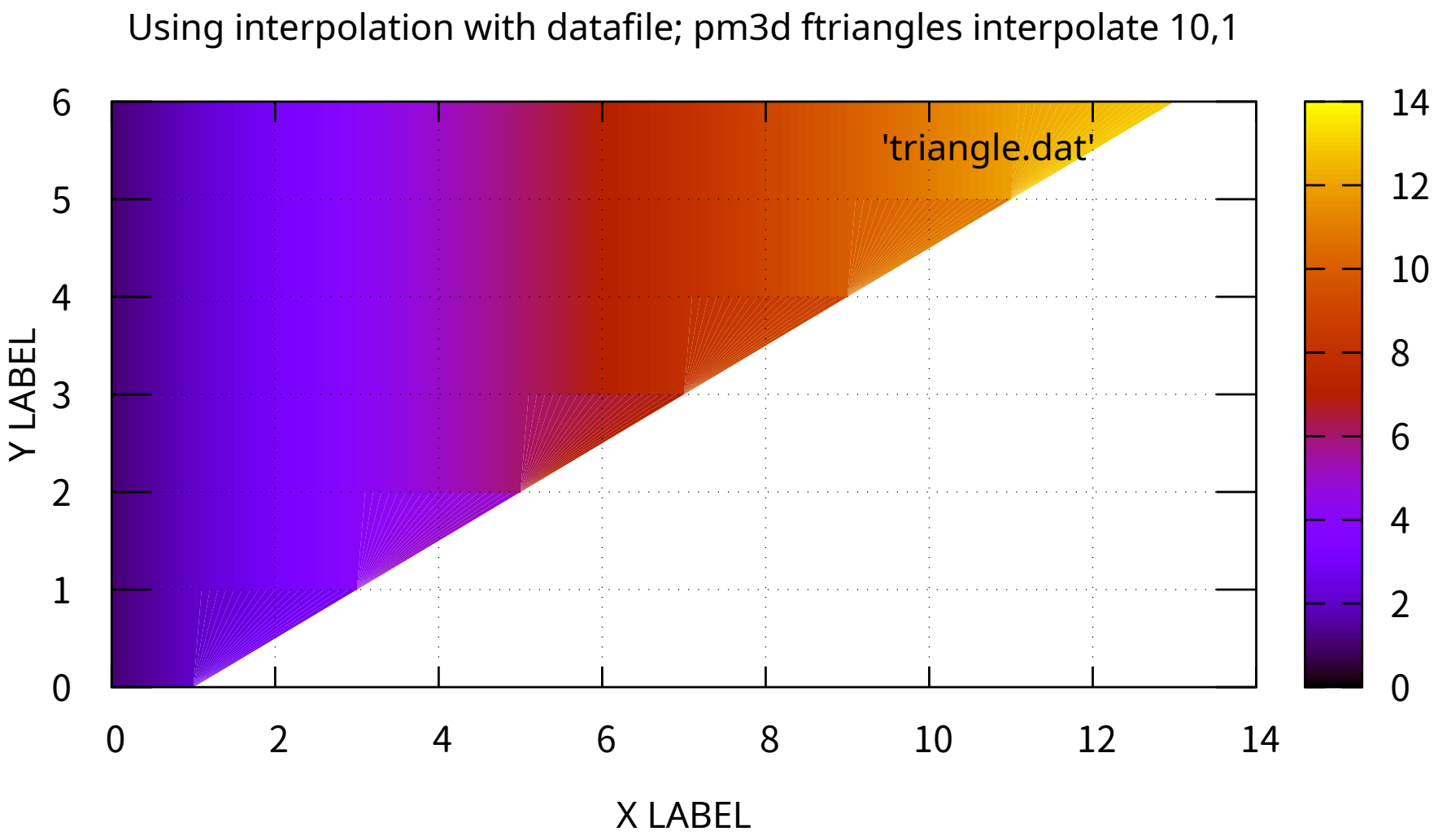

Datafile with different nb of points in scans; pm3d flush end
'triangle.dat'
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

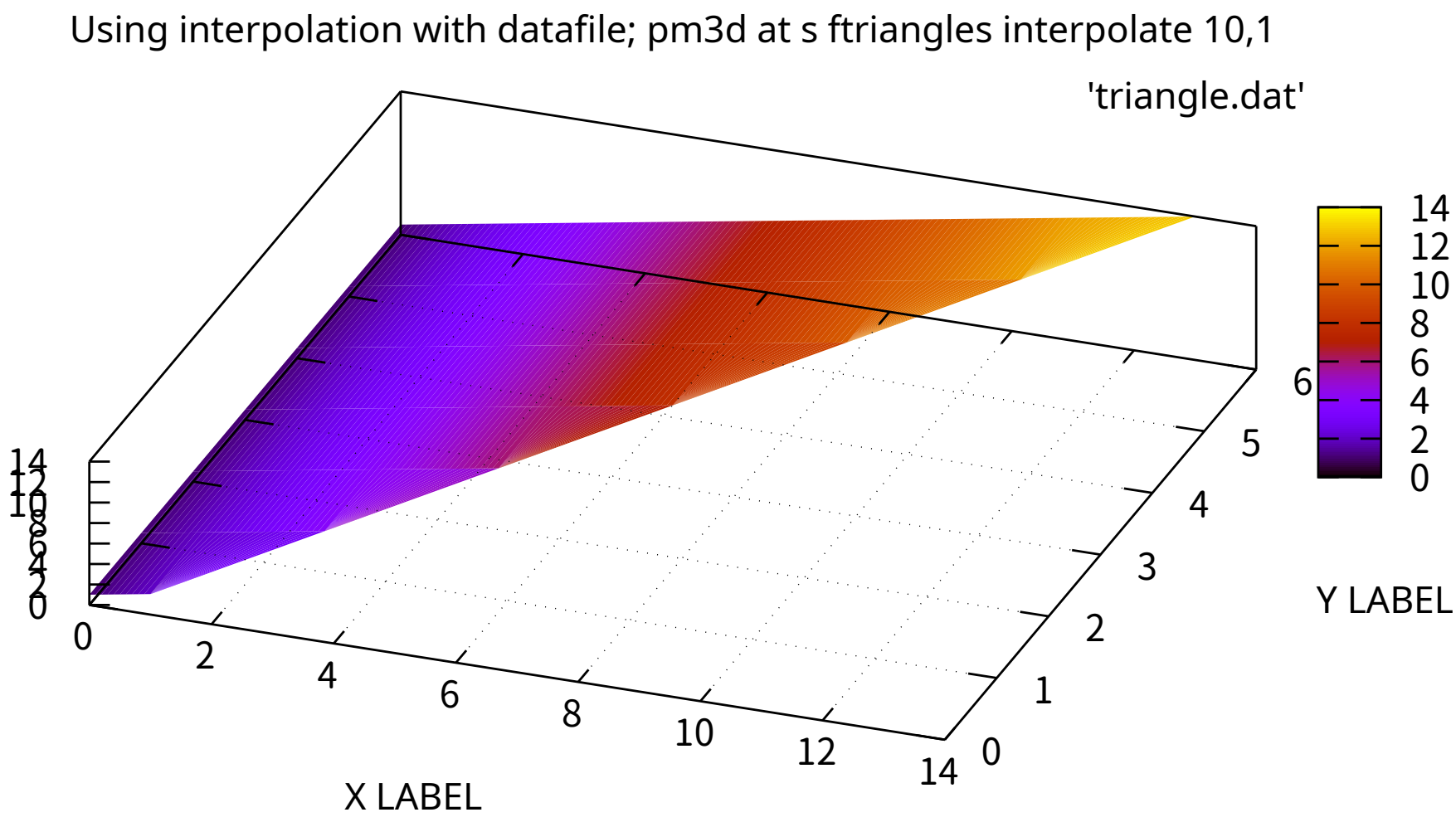

Data with different nb of points in scans; pm3d ftriangles flush begin
'triangle.dat'
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

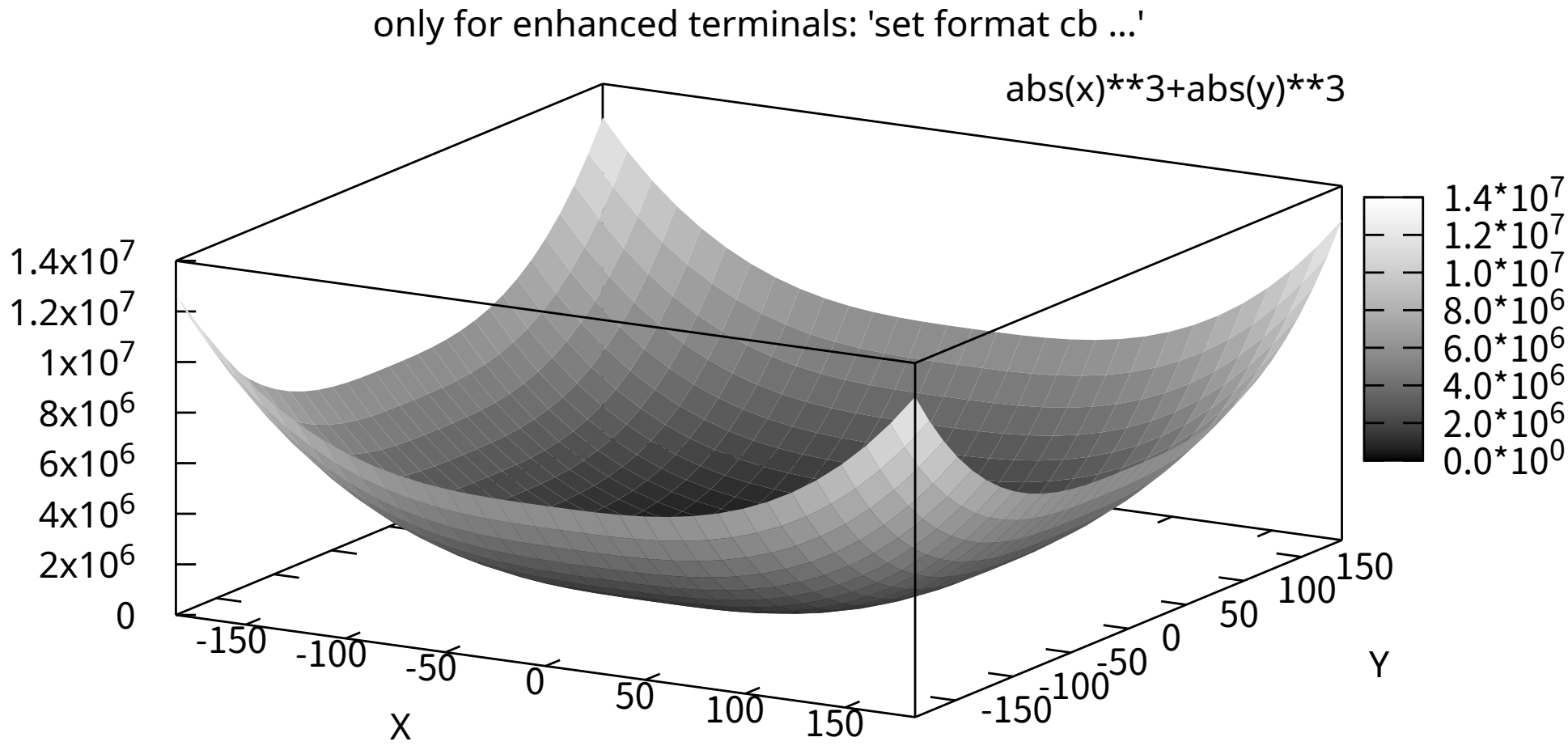

Data with different nb of points in scans; pm3d ftriangles flush center
'triangle.dat'
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14


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

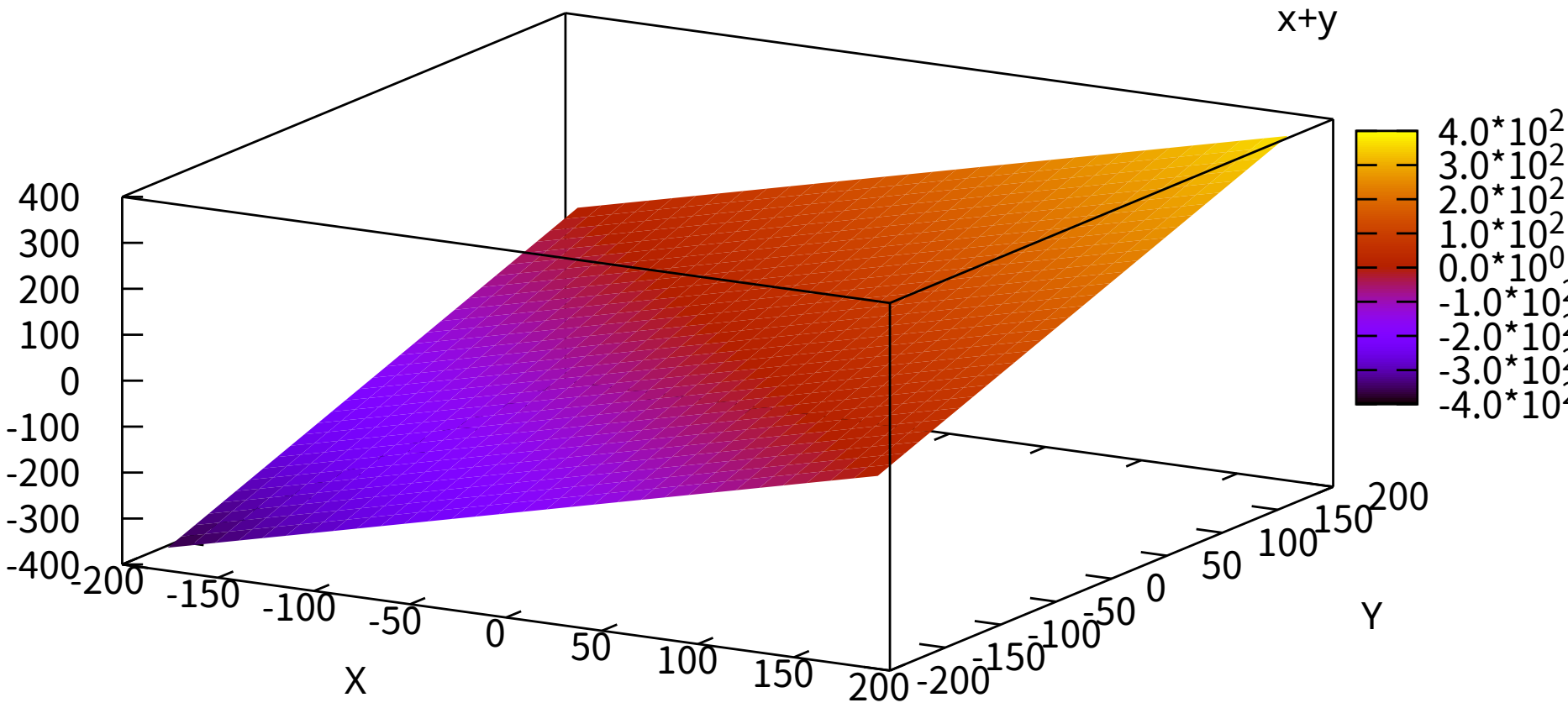

Using interpolation with datafile; pm3d interpolate 2,1
'triangle.dat'
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14


Using interpolation with datafile; pm3d 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

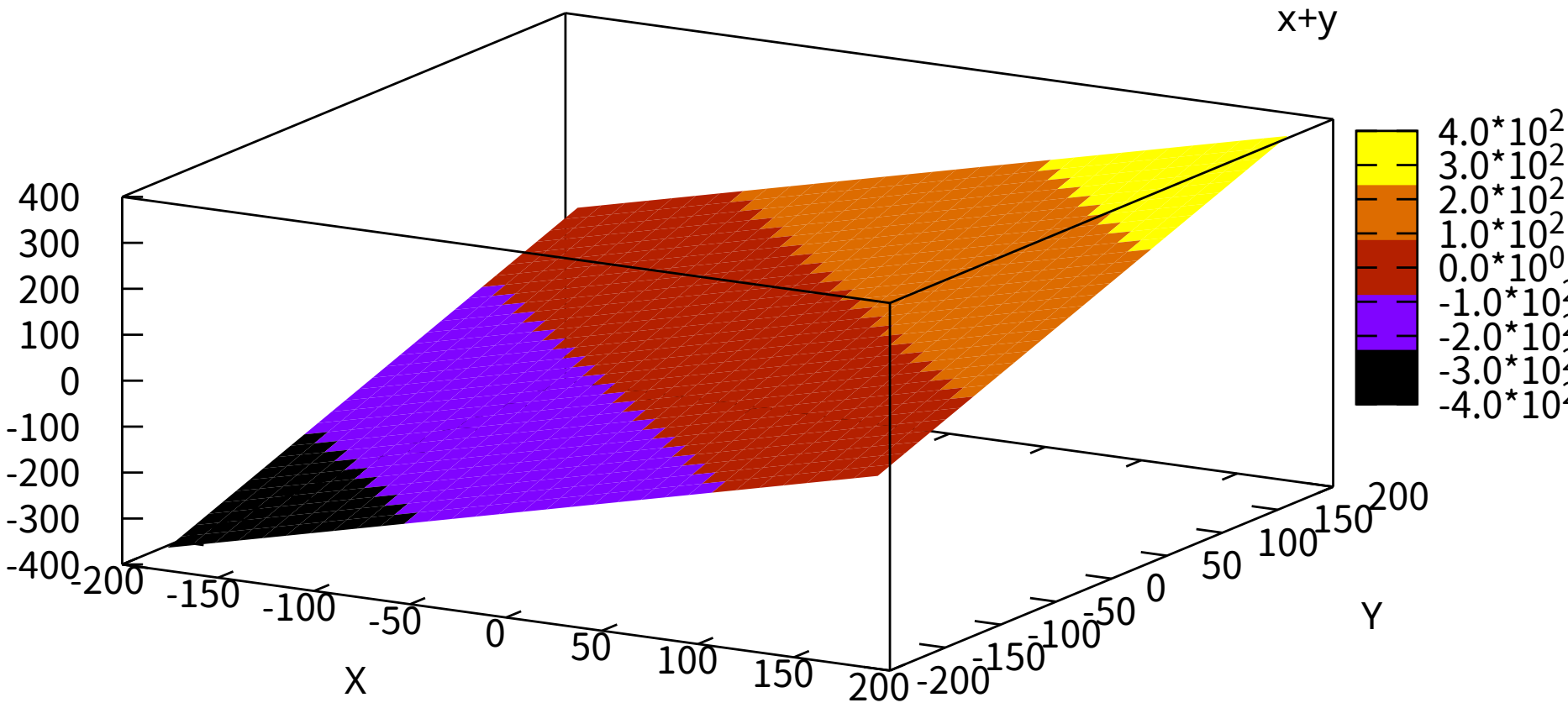

Using interpolation with datafile; pm3d at s ftriangles interpolate 10,1
'triangle.dat'
14
12
10
8
6
4
2
0
0
2
4
6
8
10
12
14
X LABEL
0
1
2
3
4
5
Y LABEL
14
12
10
8
6
4
2
0

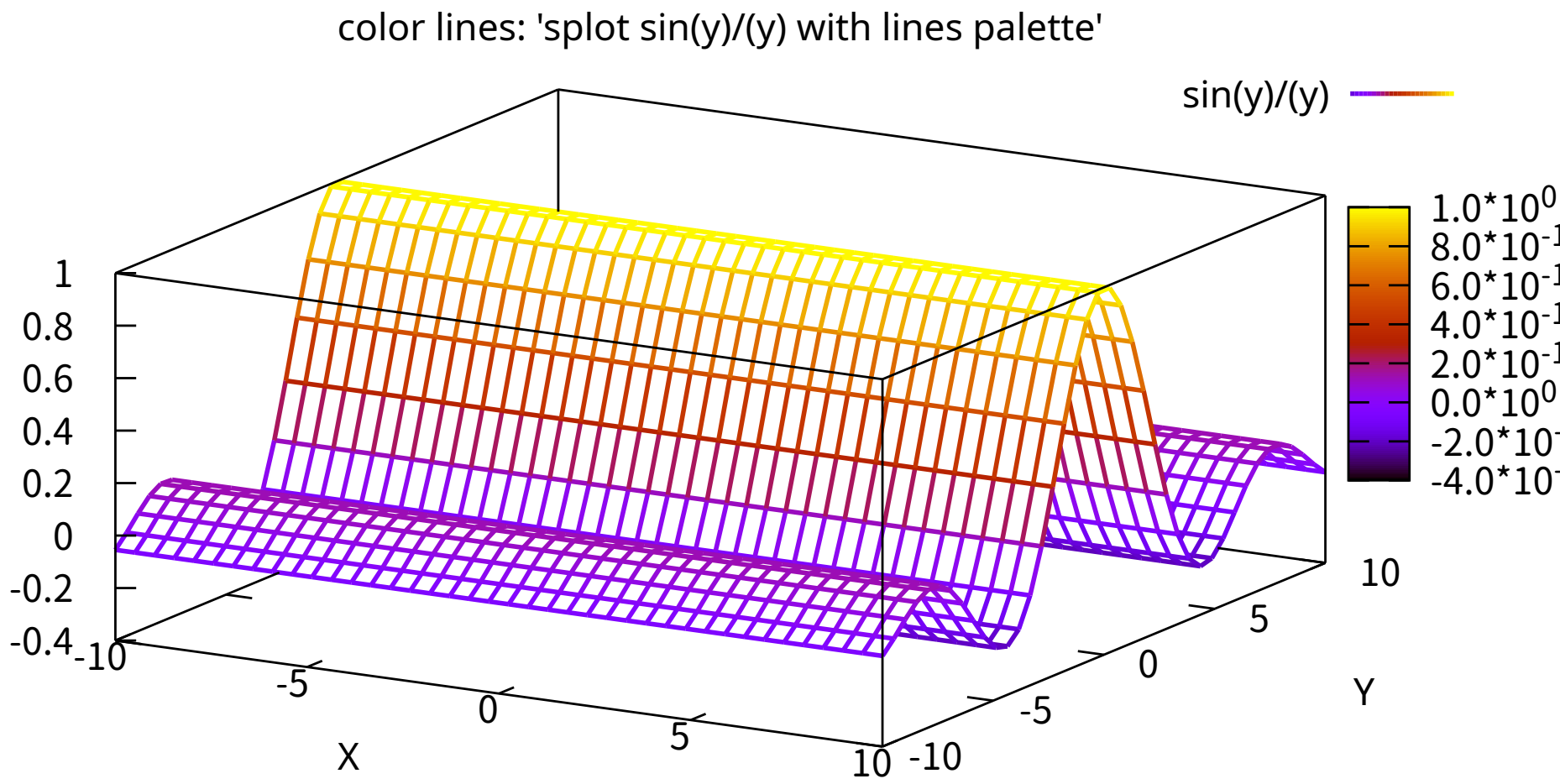

only for enhanced terminals: 'set format cb ...'
abs(x)**3+abs(y)**3
1.4x10^7
1.2x10^7
1x10^7
8x10^6
6x10^6
4x10^6
2x10^6
0
-150
-100
-50
0
50
100
150
X
-150
-100
-50
0
50
100
150
Y
1.4*10^7
1.2*10^7
1.0*10^7
8.0*10^6
6.0*10^6
4.0*10^6
2.0*10^6
0.0*10^0

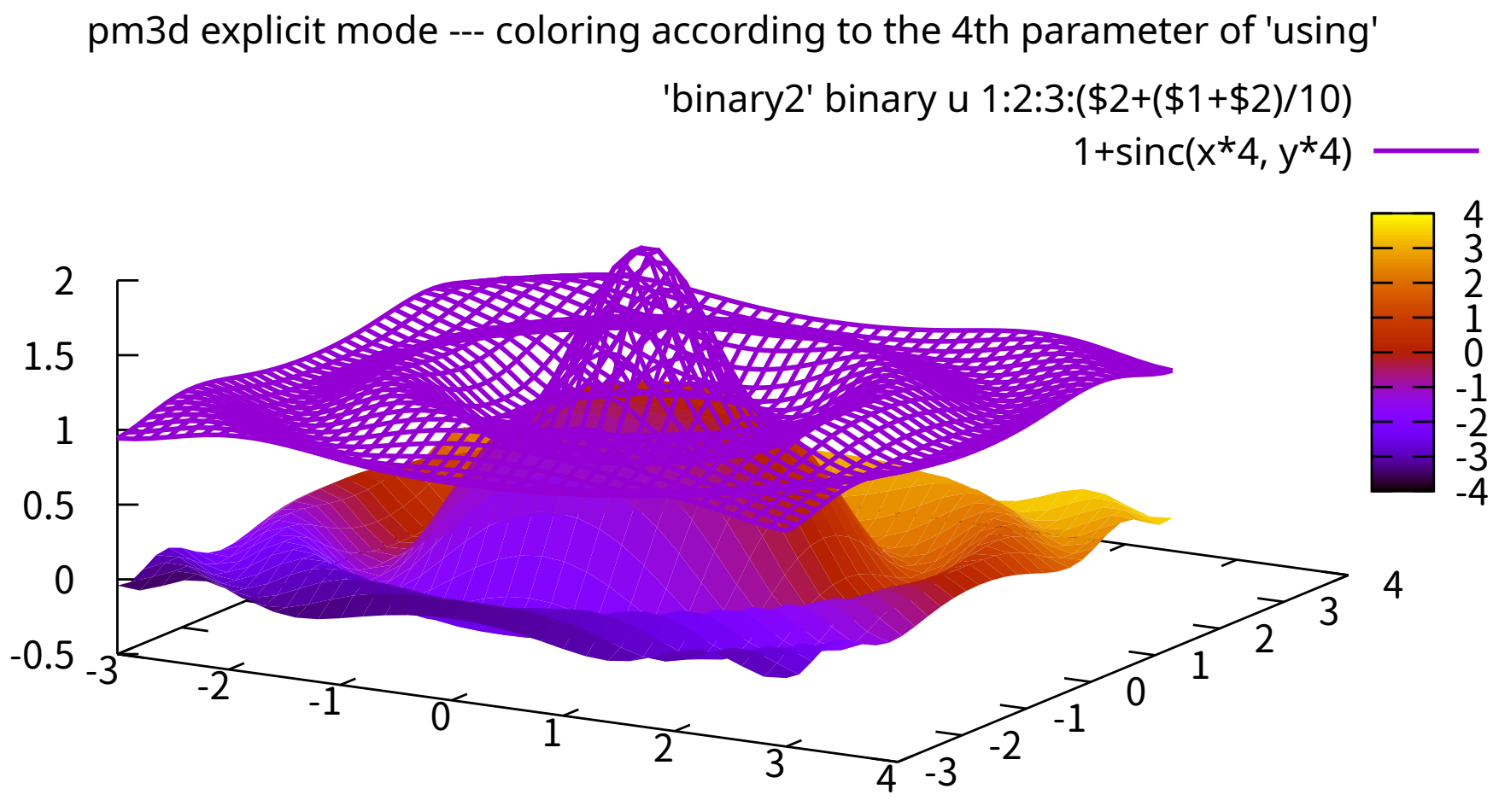
function 'x+y' using all colors available, 'set pal maxcolors 0'



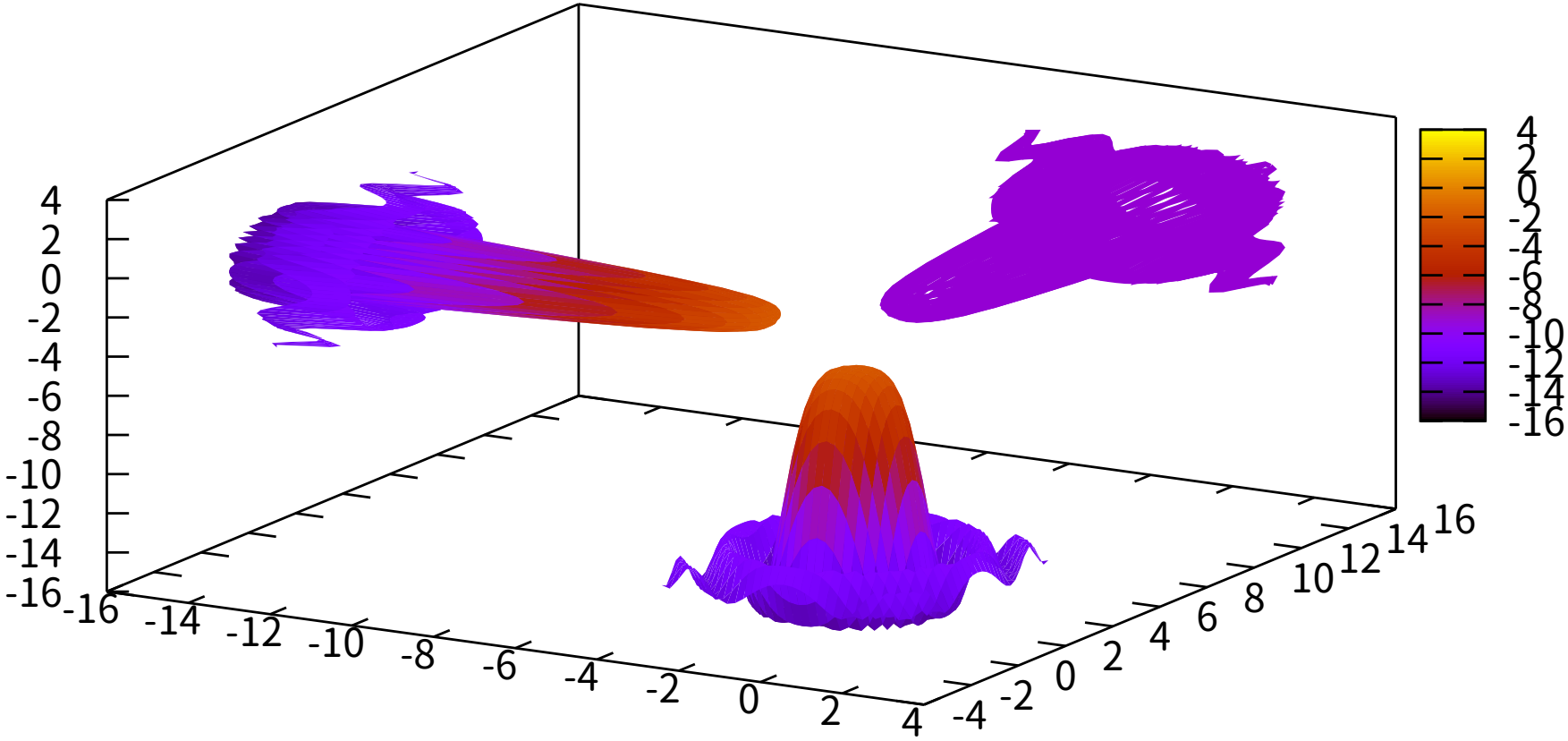
function 'x+y' using only 5 colors, 'set pal maxcolors 5'




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

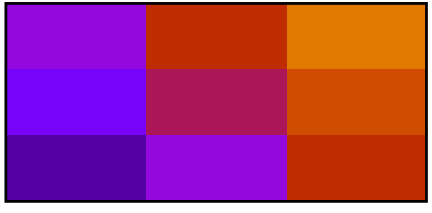

pm3d explicit mode --- coloring according to the 4th parameter of 'using'
'binary2' binary u 1:2:3:($2+($1+$2)/10)
1+sinc(x*4, y*4)
2
1.5
1
0.5
0
-0.5
-3
-2
-1
0
1
2
3
4
-3
-2
-1
0
1
2
3
4
4
3
2
1
0
-1
-2
-3
-4

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

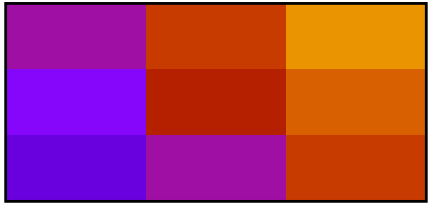


set pm3d corners2color mode

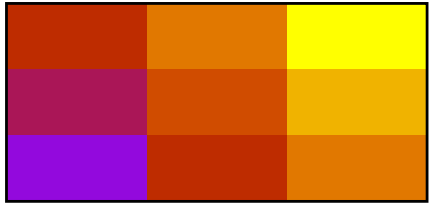
c3



mean

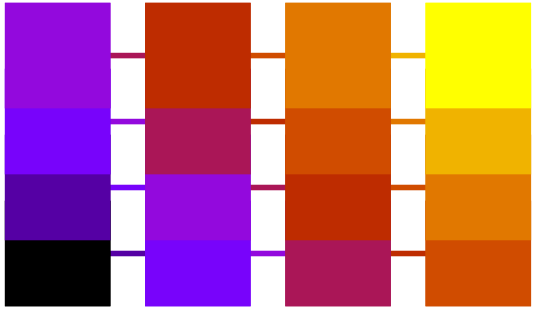


c4

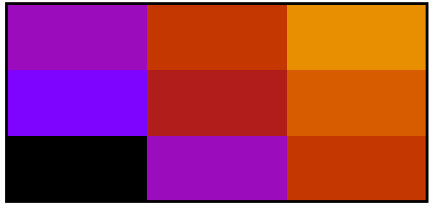


harmean

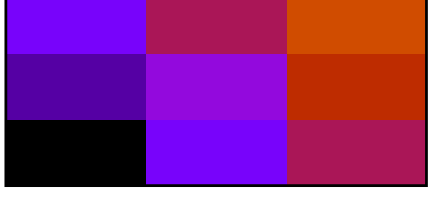
Original grid points



geomean



c1

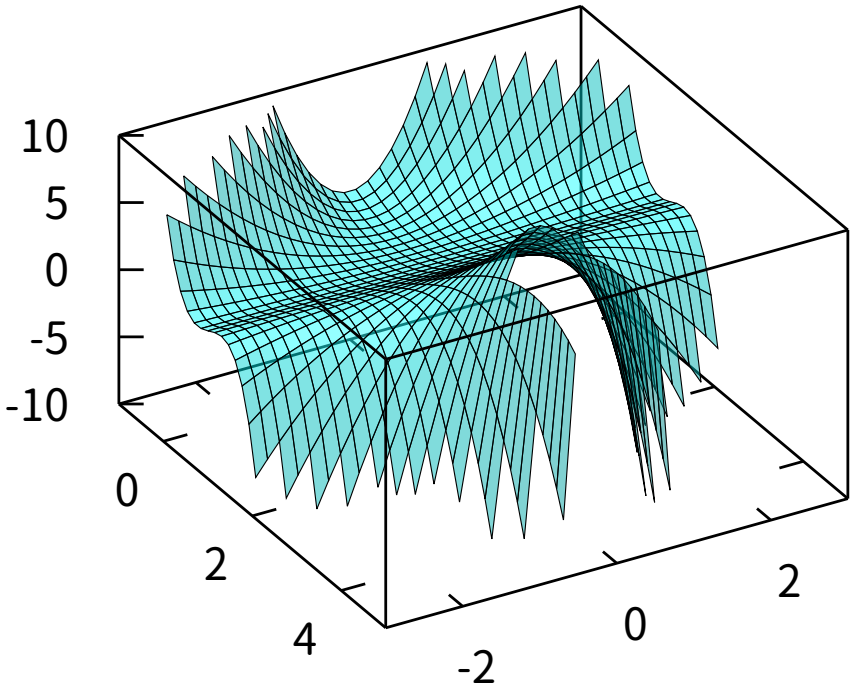


median

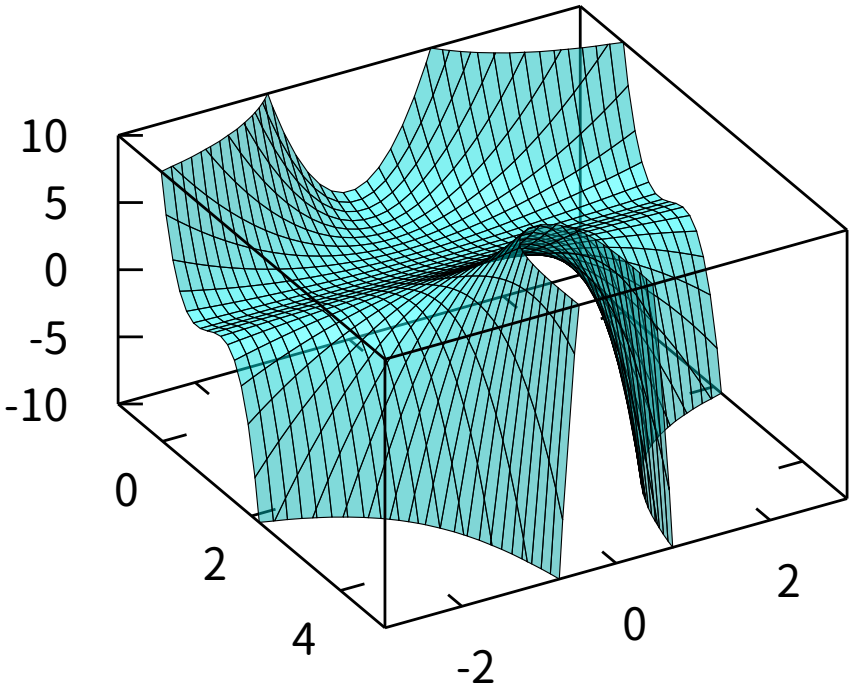
c2

Gnuplot 6 pm3d default is smooth clipping against zrange

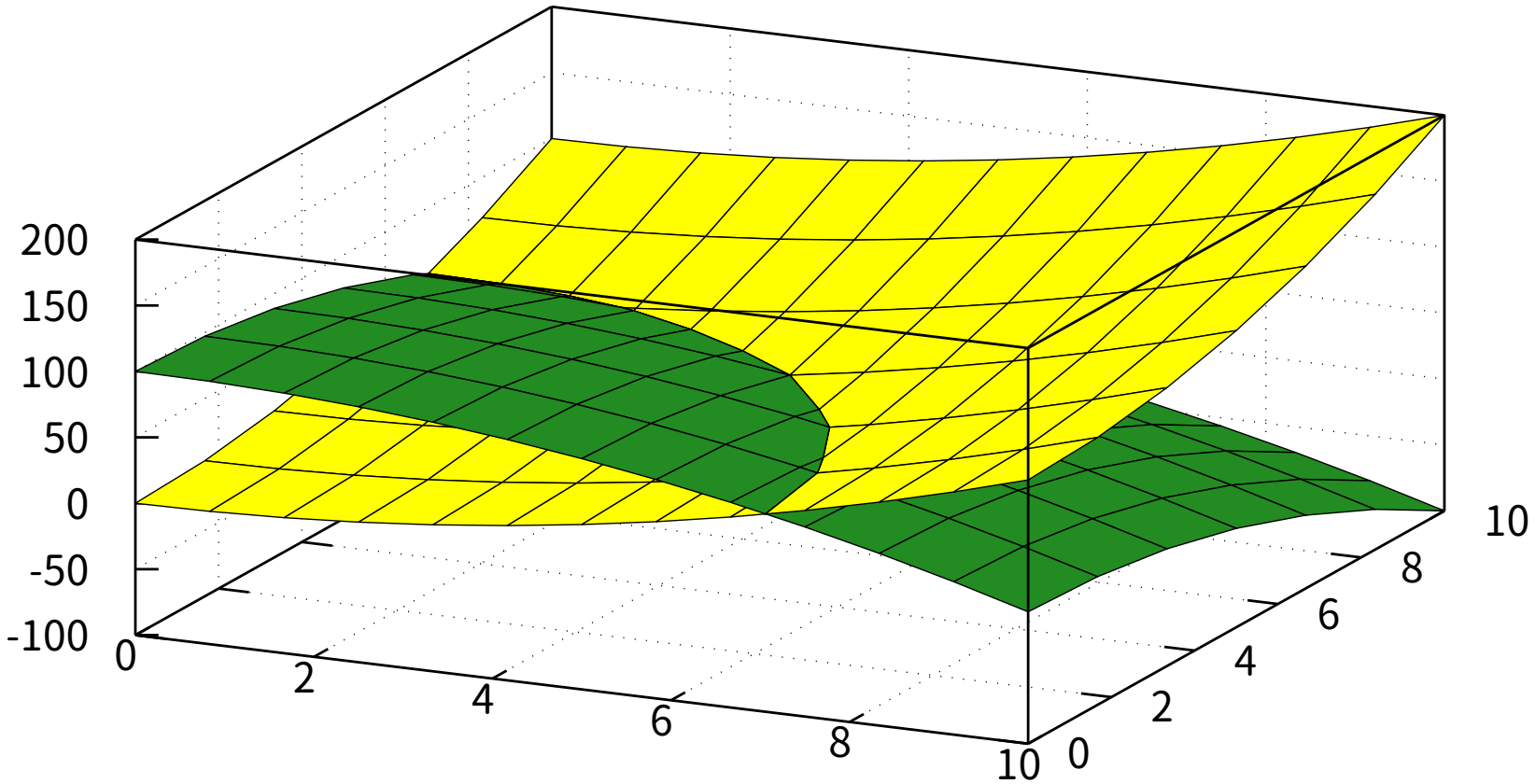
set pm3d clip4in (old default)

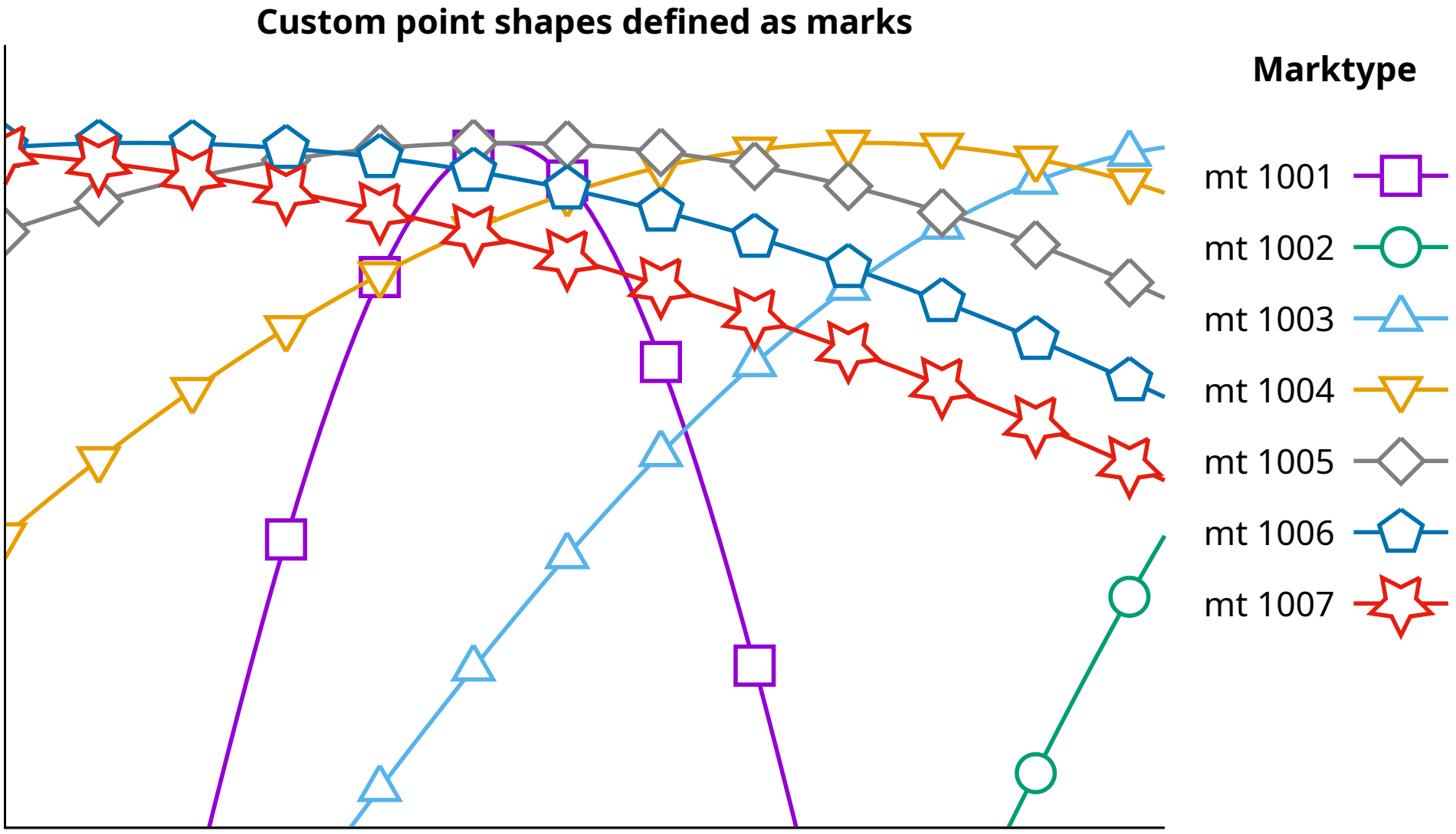


set pm3d clip (Gnuplot 6)

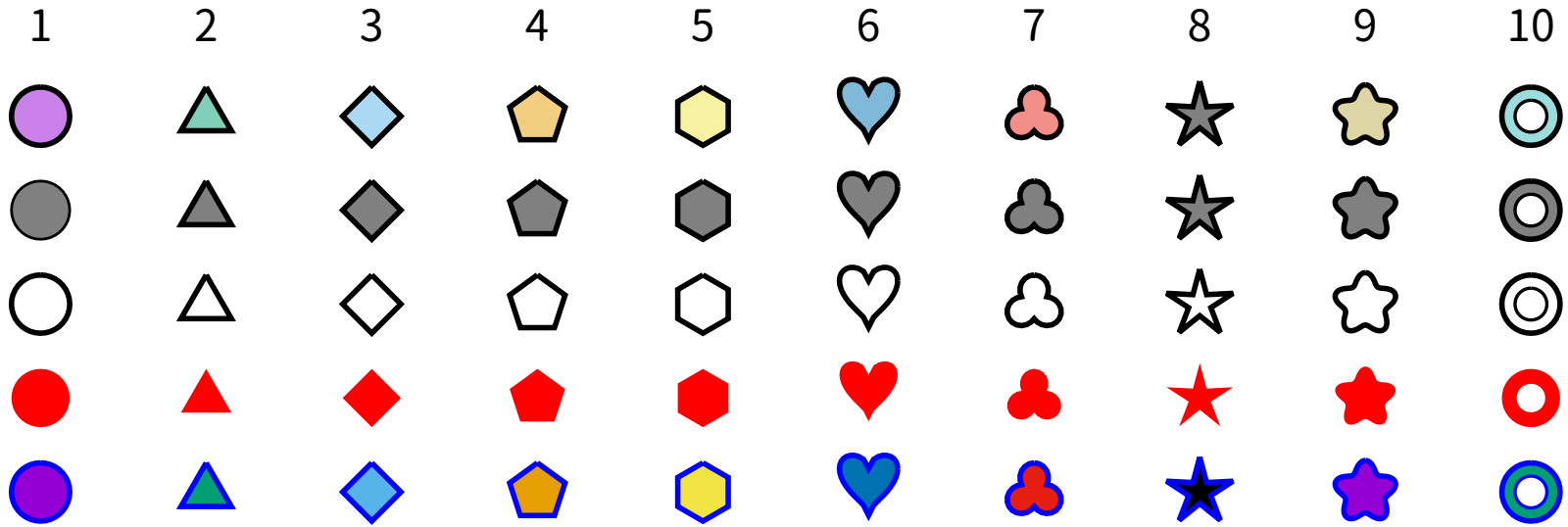


Smooth intersection of pm3d surfaces

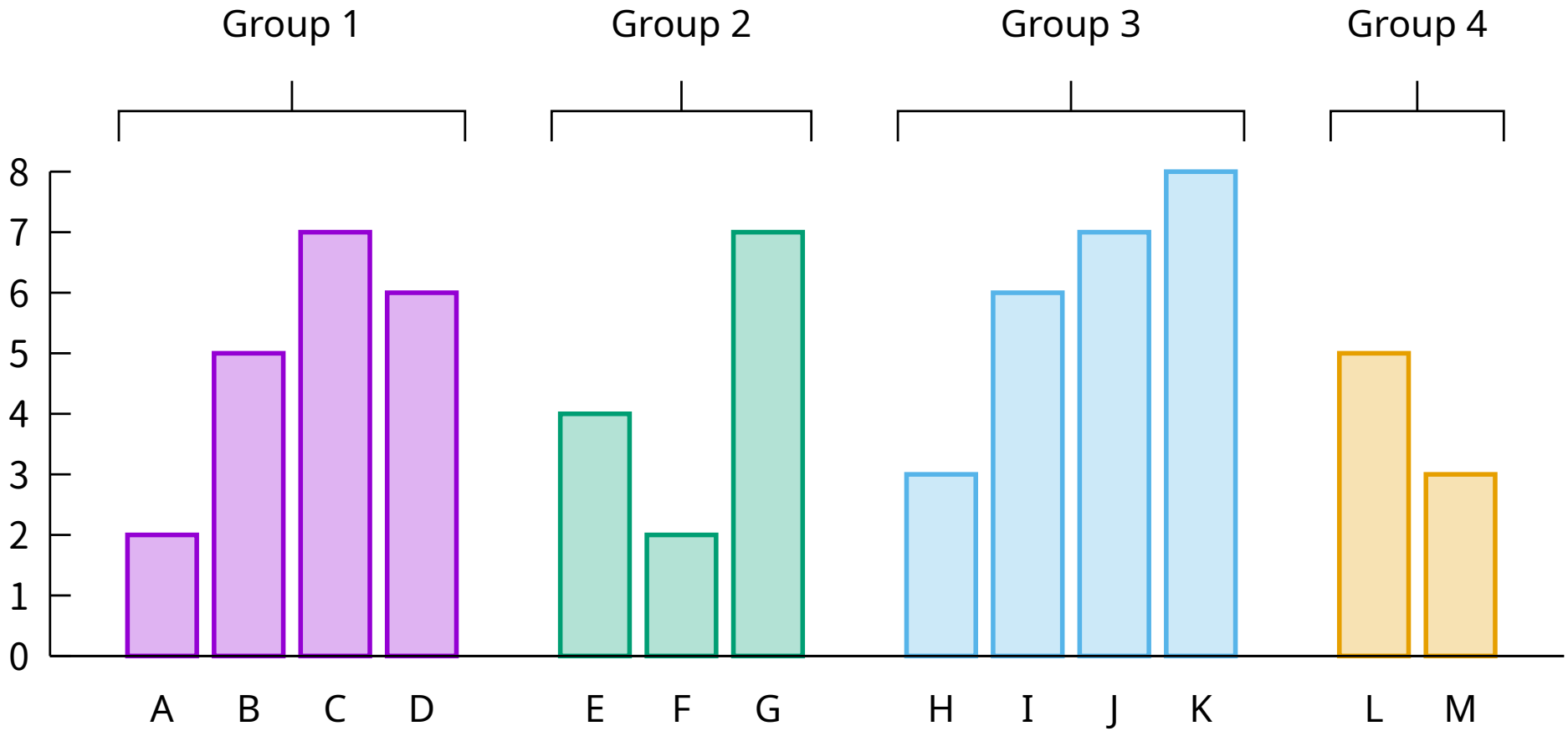



Custom point shapes defined as marks
Marktype
mt 1001
mt 1002
mt 1003
mt 1004
mt 1005
mt 1006
mt 1007

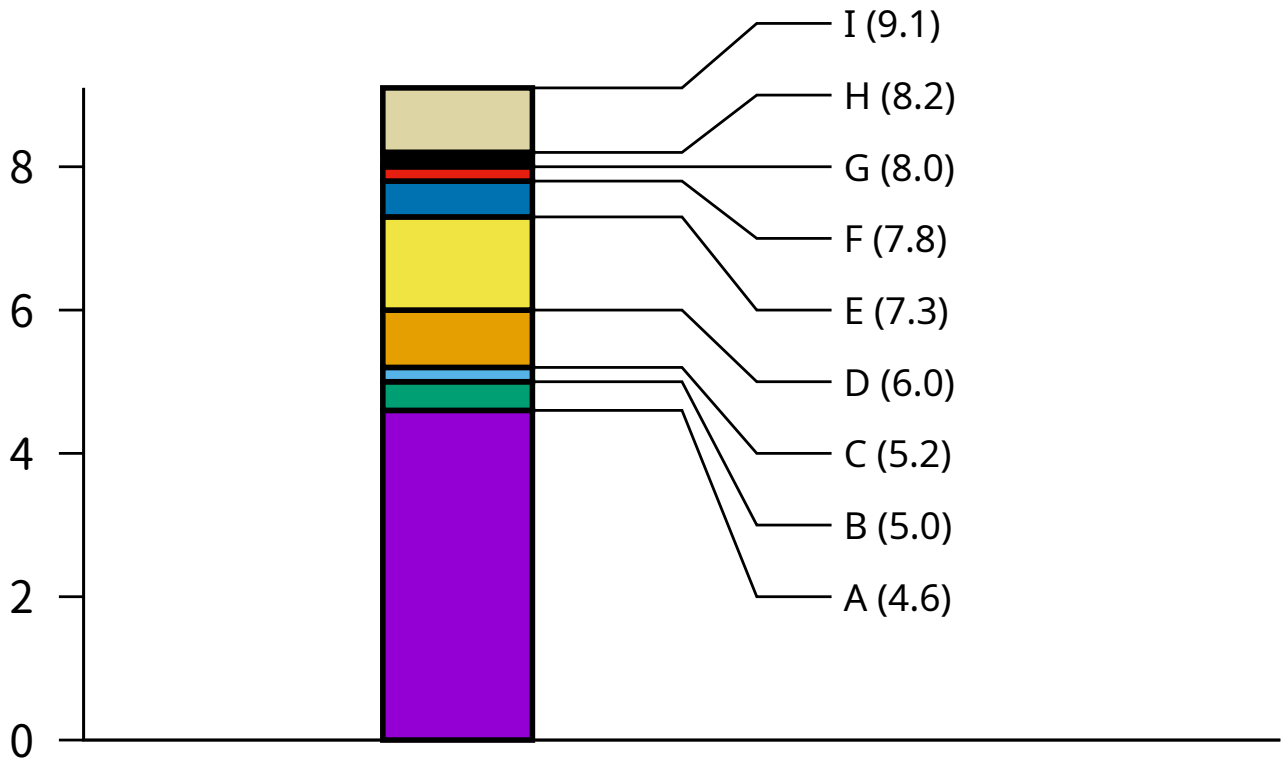
Mark shapes defined by formulas

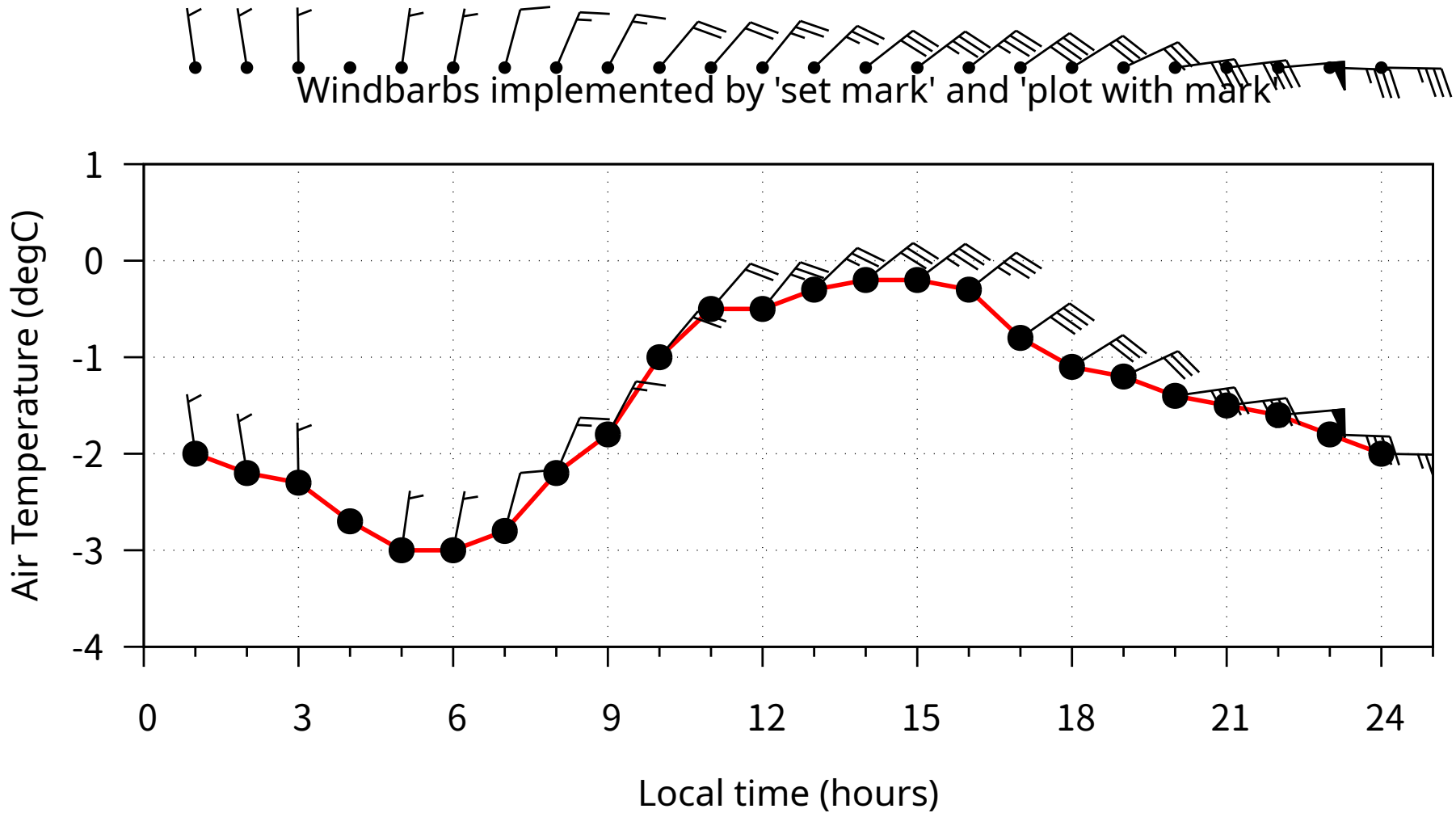


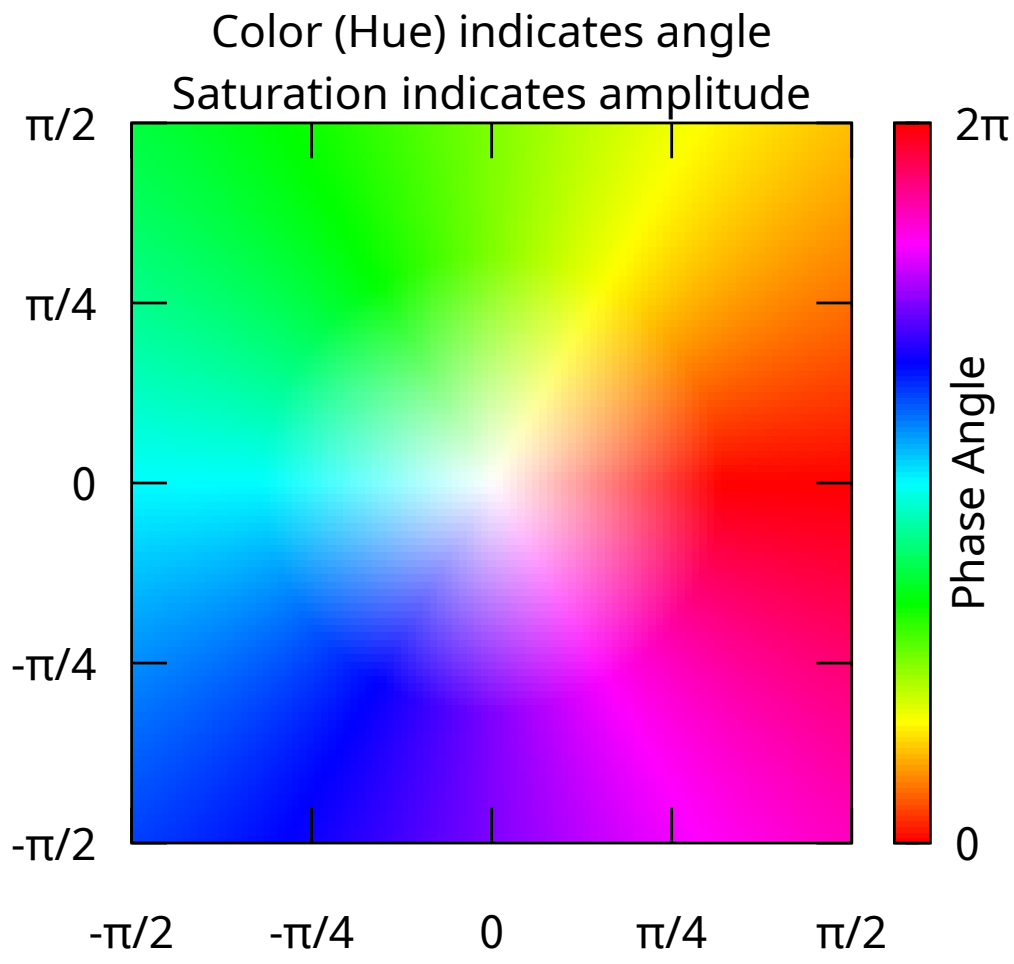
Adjustable width mark used to indicate grouping

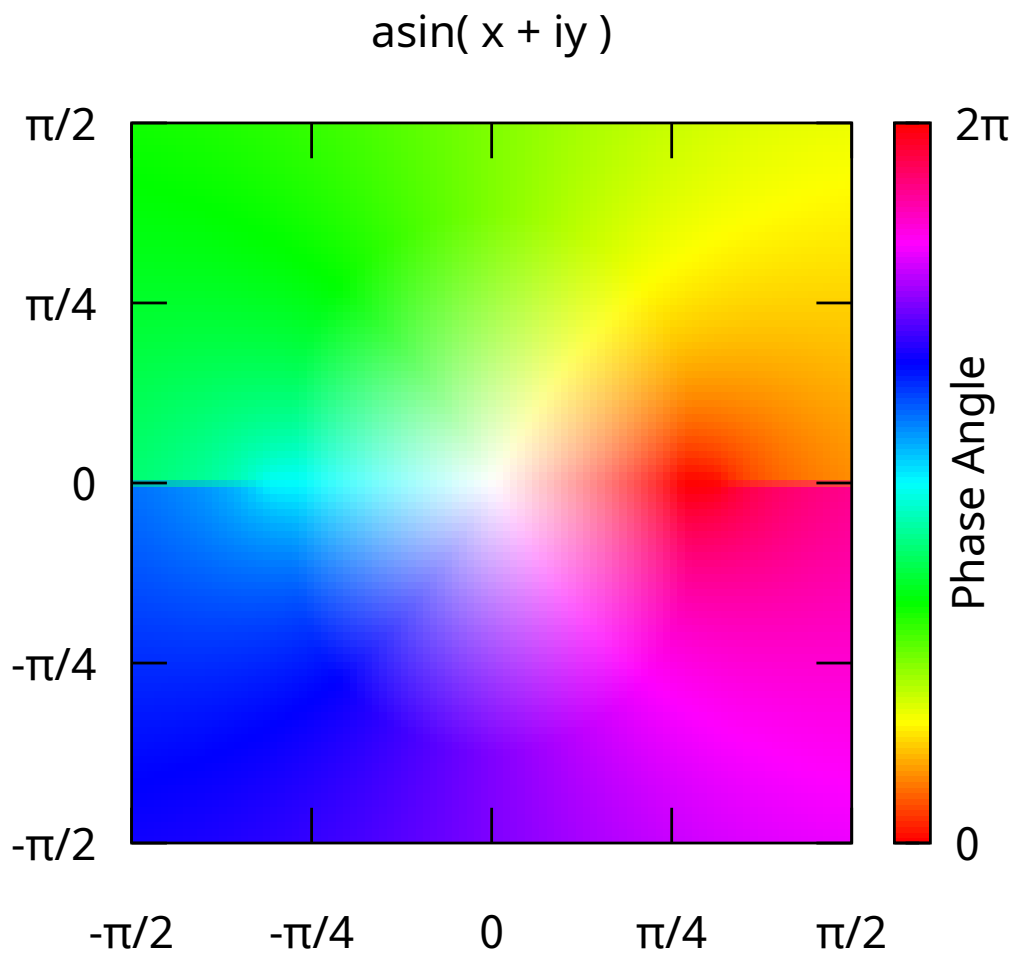


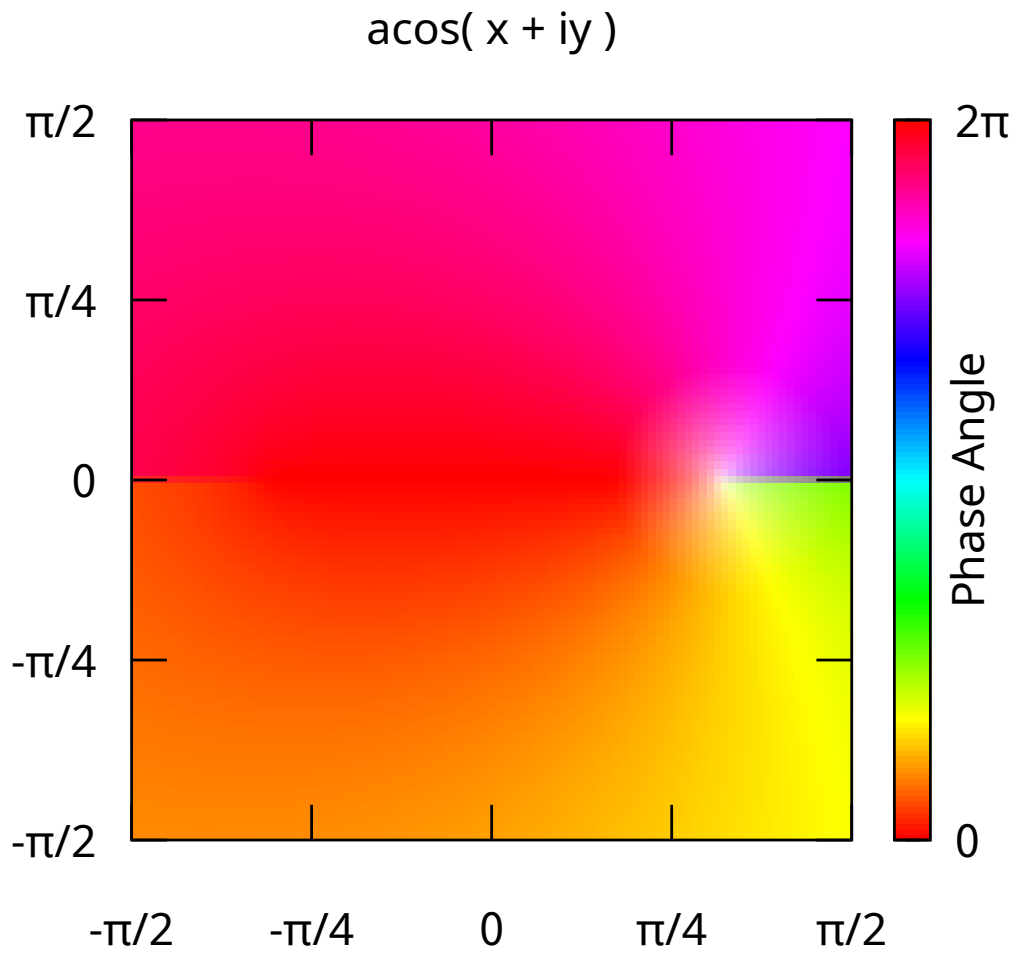
Adjustable height mark used to attach labels

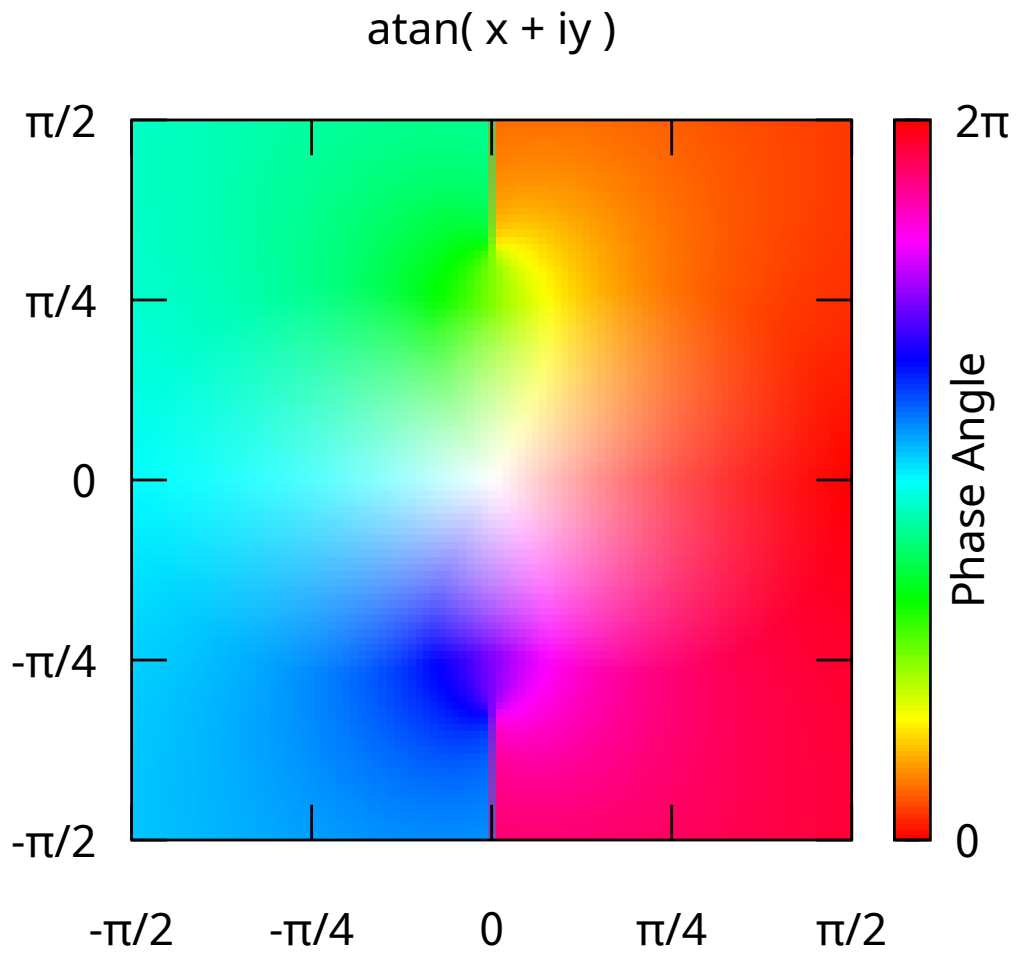


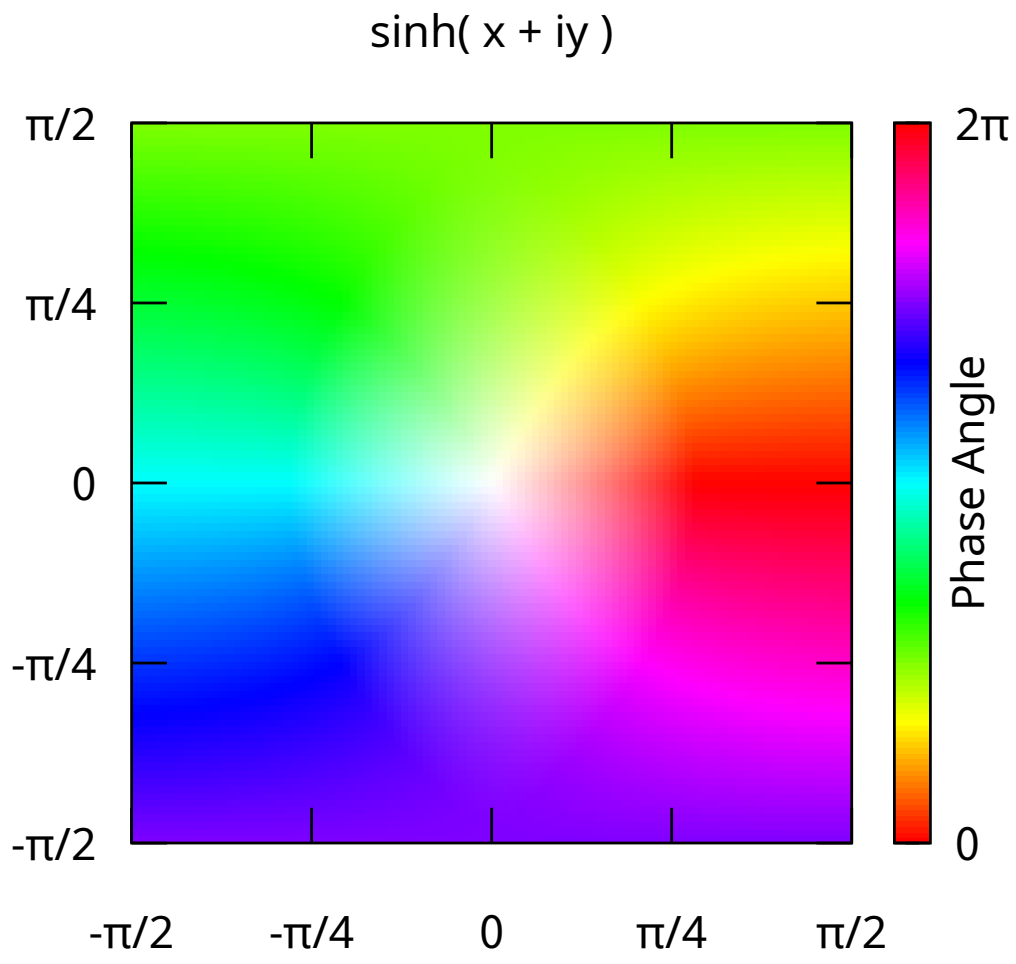

Windbarbs implemented by 'set mark' and 'plot with mark'
Air Temperature (degC)
1
0
-1
-2
-3
-4
0
3
6
9
12
15
18
21
24
Local time (hours)

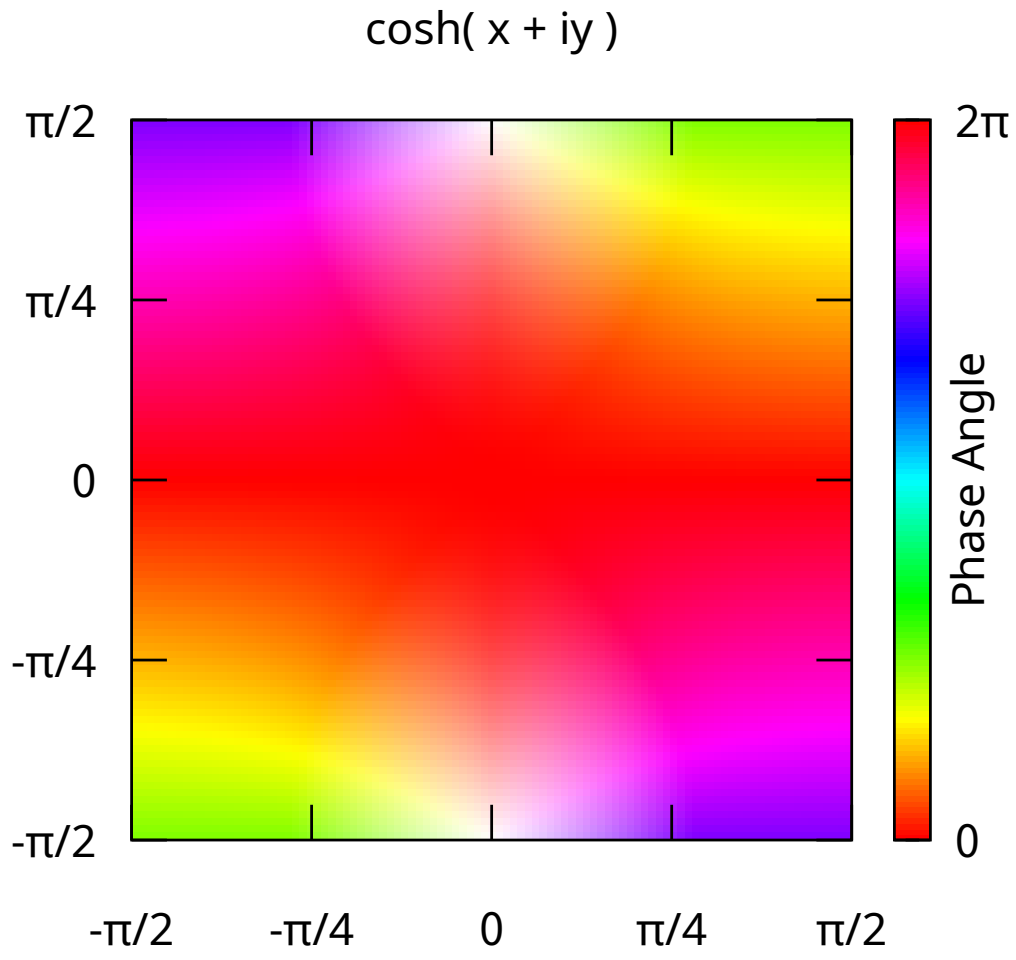

Color (Hue) indicates angle
Saturation indicates amplitude
π/2
π/4
0
-π/4
-π/2
-π/2
-π/4
0
π/4
π/2
2π
0
Phase Angle

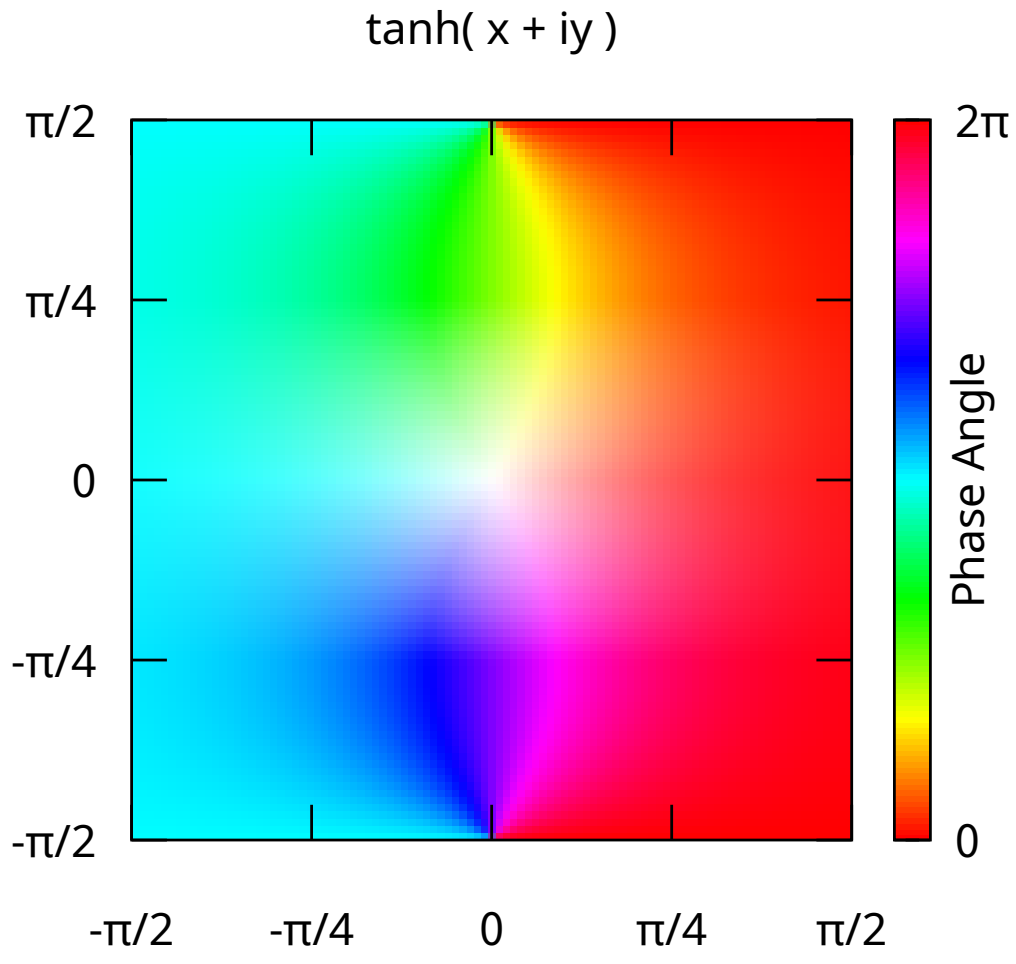

asin(x + iy)
π/2
π/4
0
-π/4
-π/2
-π/2
-π/4
0
π/4
π/2
2π
0
Phase Angle

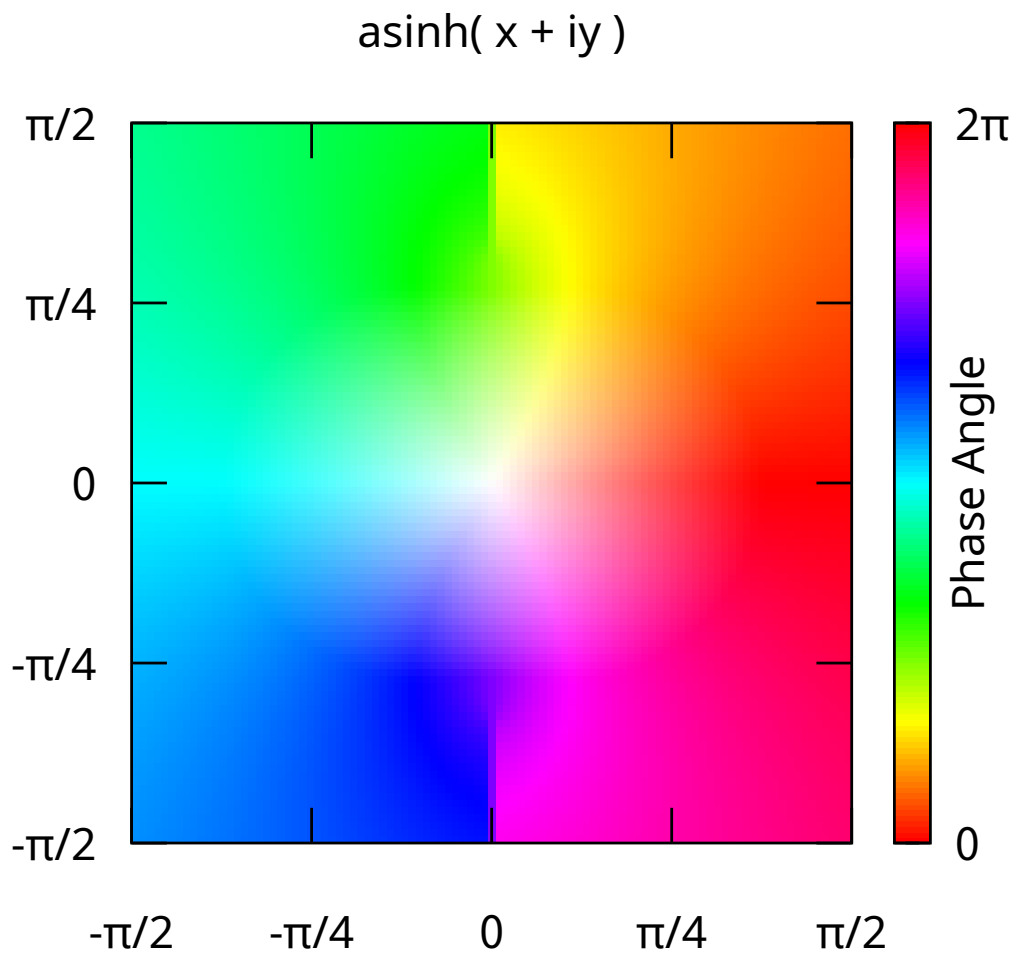

acos(x + iy)
π/2
π/4
0
-π/4
-π/2
-π/2
-π/4
0
π/4
π/2
2π
0
Phase Angle

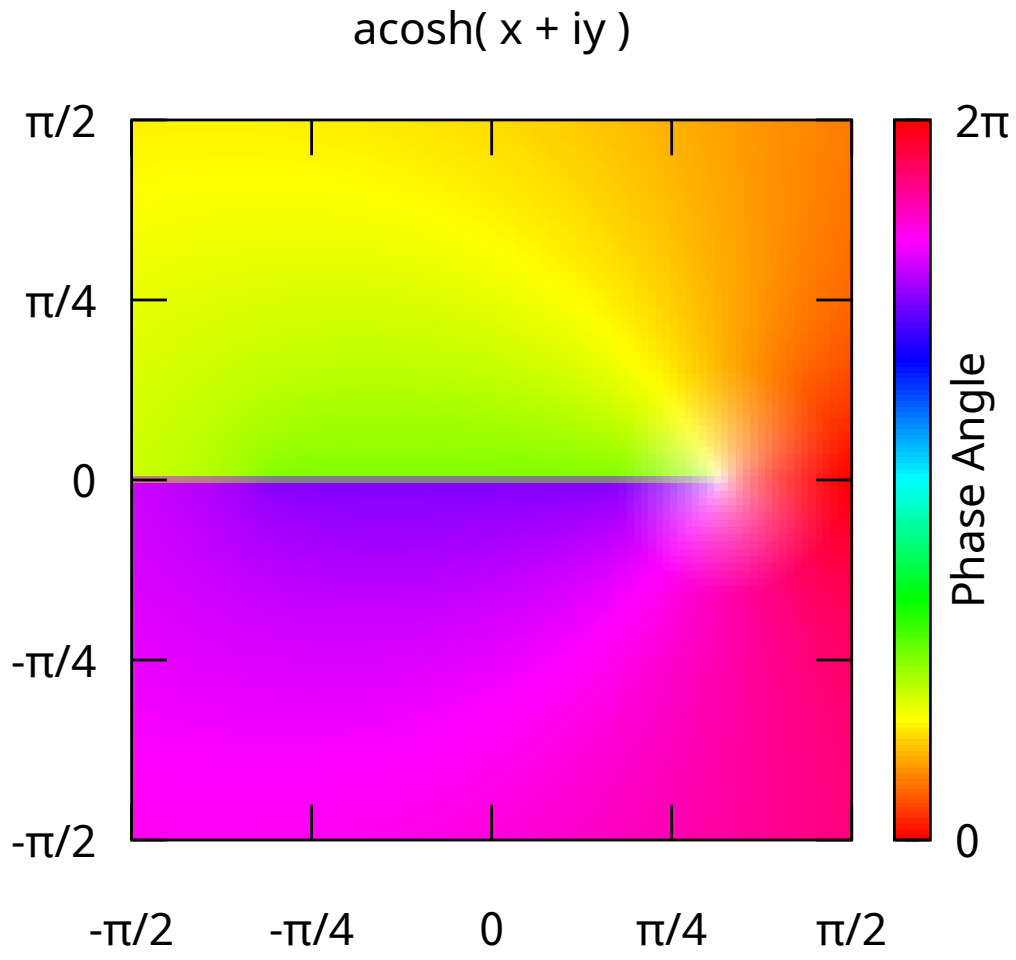

atan(x + iy)
π/2
π/4
0
-π/4
-π/2
-π/2
-π/4
0
π/4
π/2
2π
0
Phase Angle

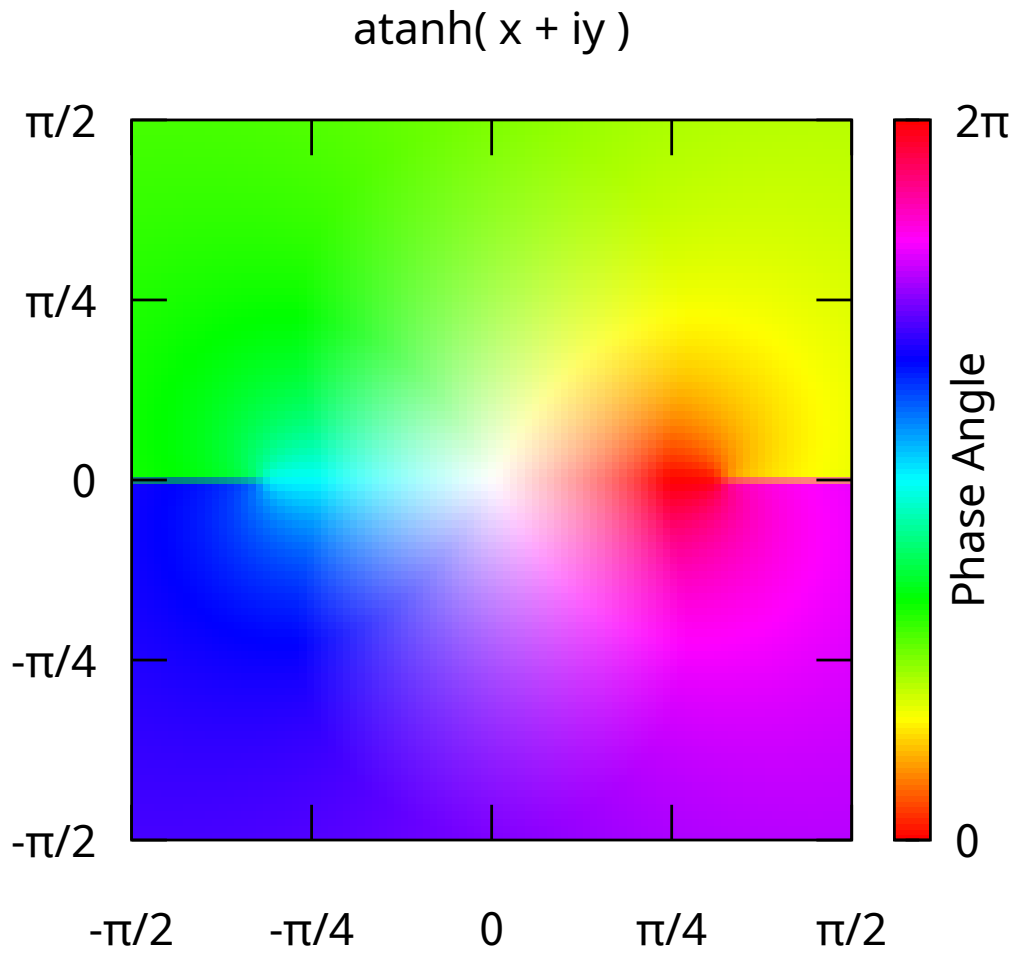

sinh(x + iy)
π/2
π/4
0
-π/4
-π/2
-π/2
-π/4
0
π/4
π/2
2π
0
Phase Angle

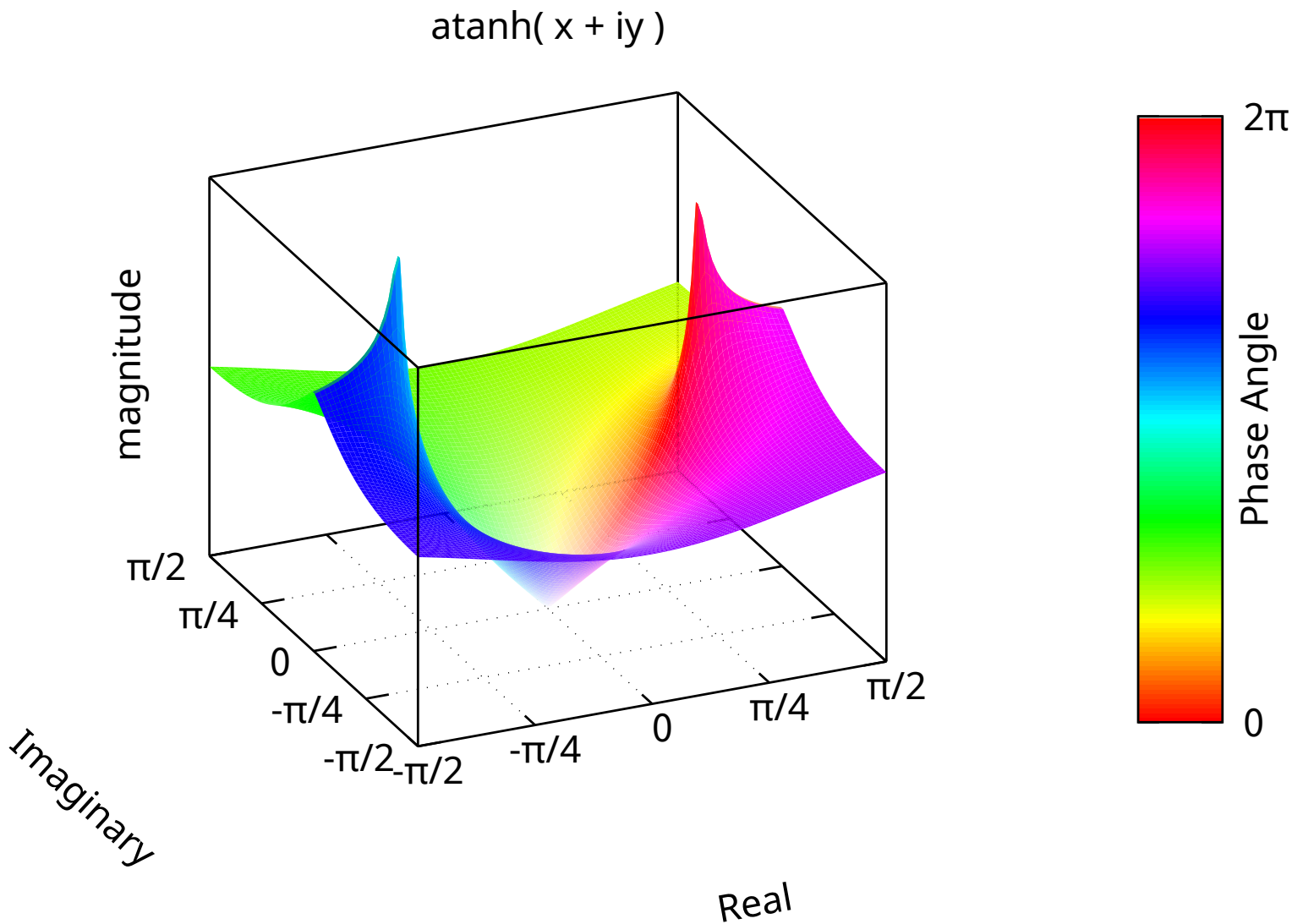

cosh(x + iy)
π/2
π/4
0
-π/4
-π/2
-π/2
-π/4
0
π/4
π/2
2π
0
Phase Angle

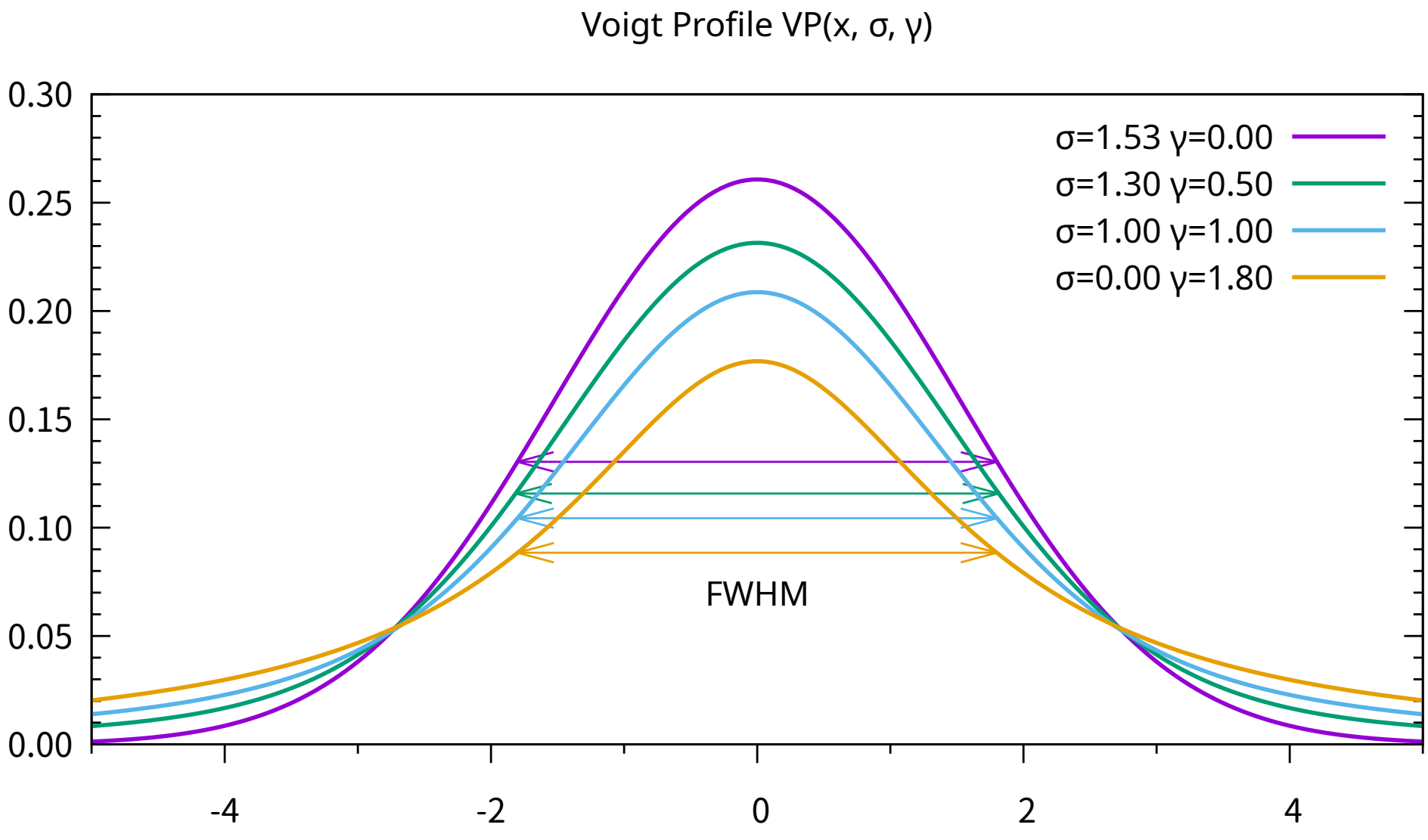

tanh(x + iy)
π/2
π/4
0
-π/4
-π/2
-π/2
-π/4
0
π/4
π/2
2π
0
Phase Angle

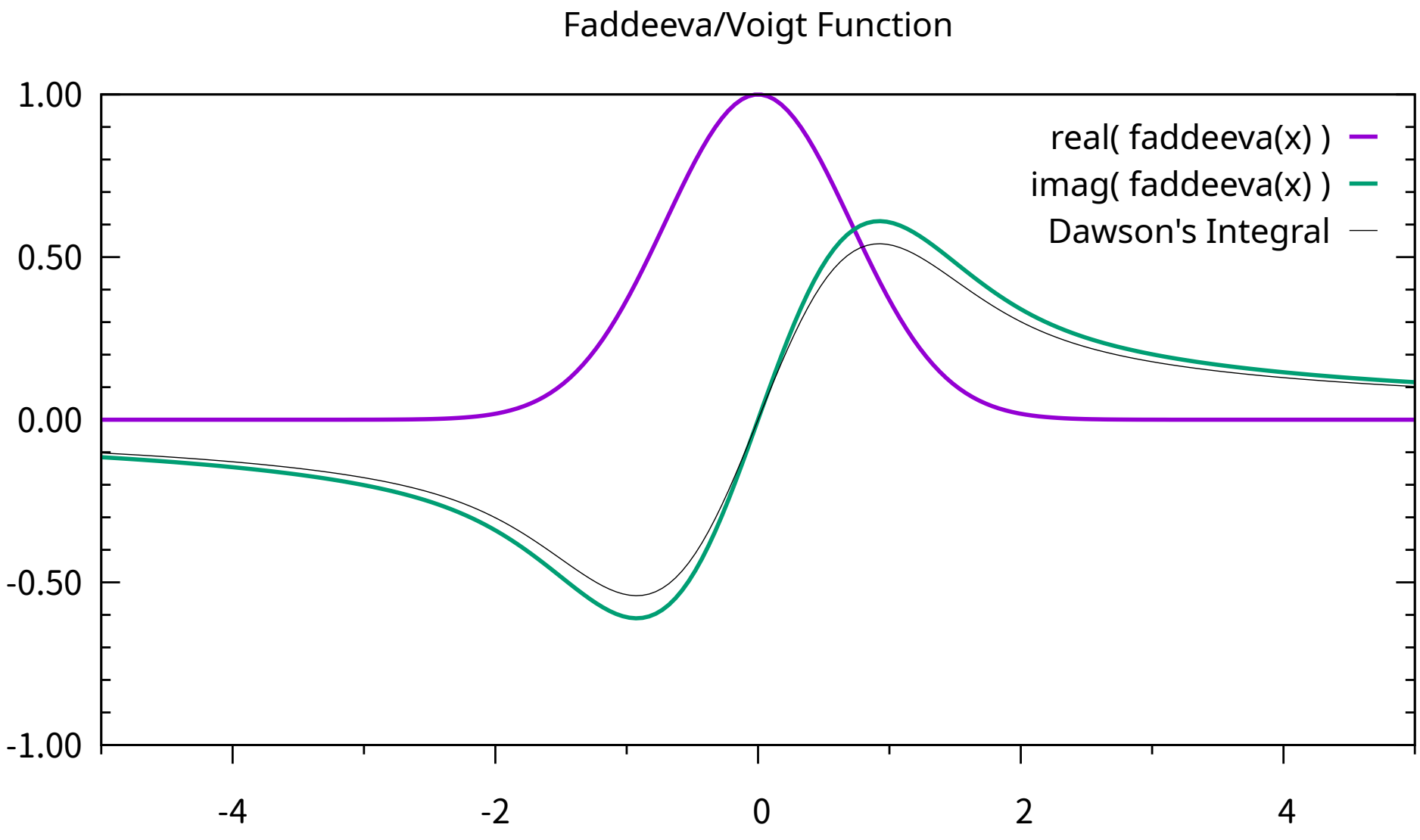

asinh(x + iy)
π/2
π/4
0
-π/4
-π/2
-π/2
-π/4
0
π/4
π/2
2π
0
Phase Angle


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

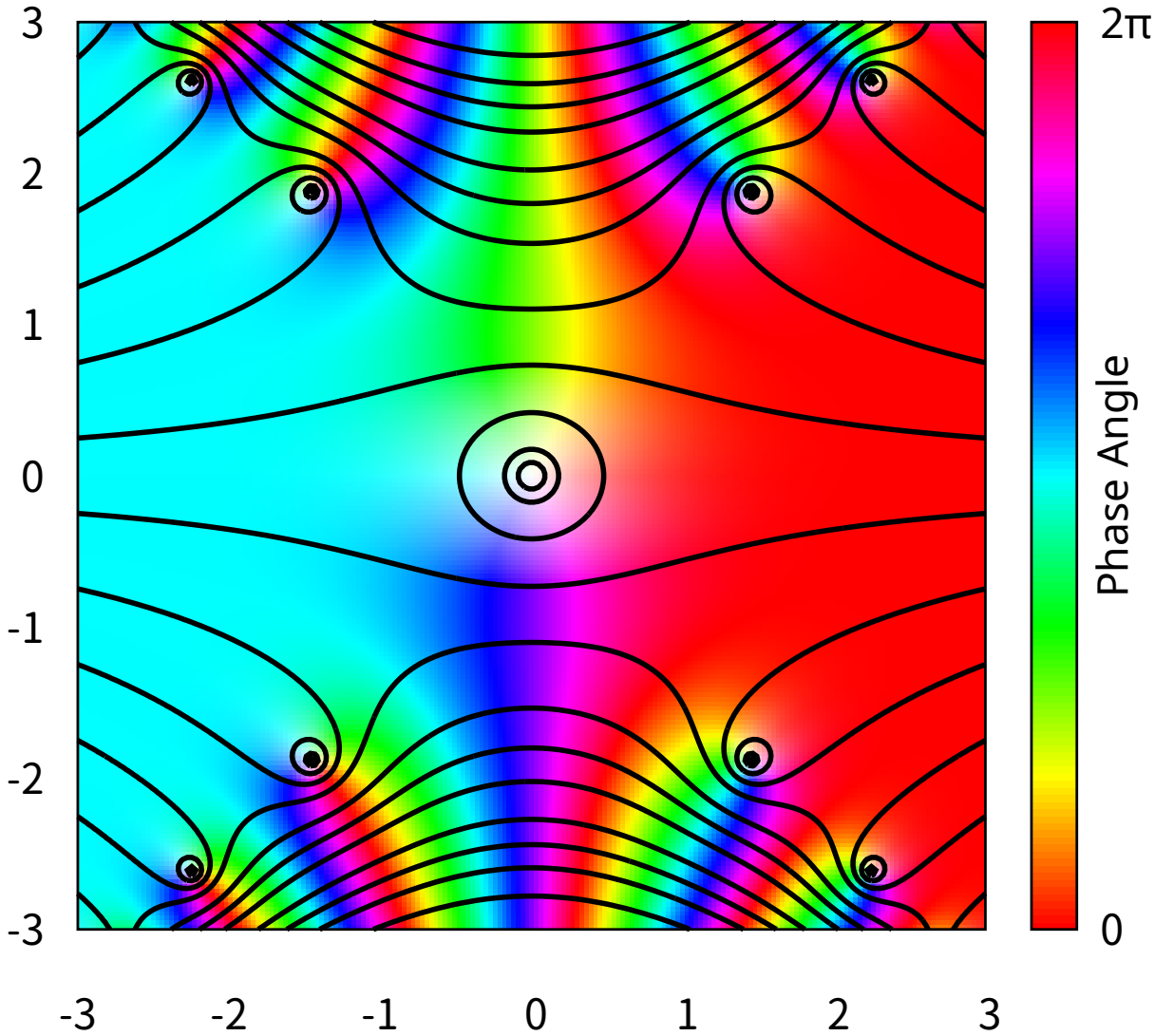

atanh(x + iy)
π/2
π/4
0
-π/4
-π/2
-π/2
-π/4
0
π/4
π/2
2π
0
Phase Angle

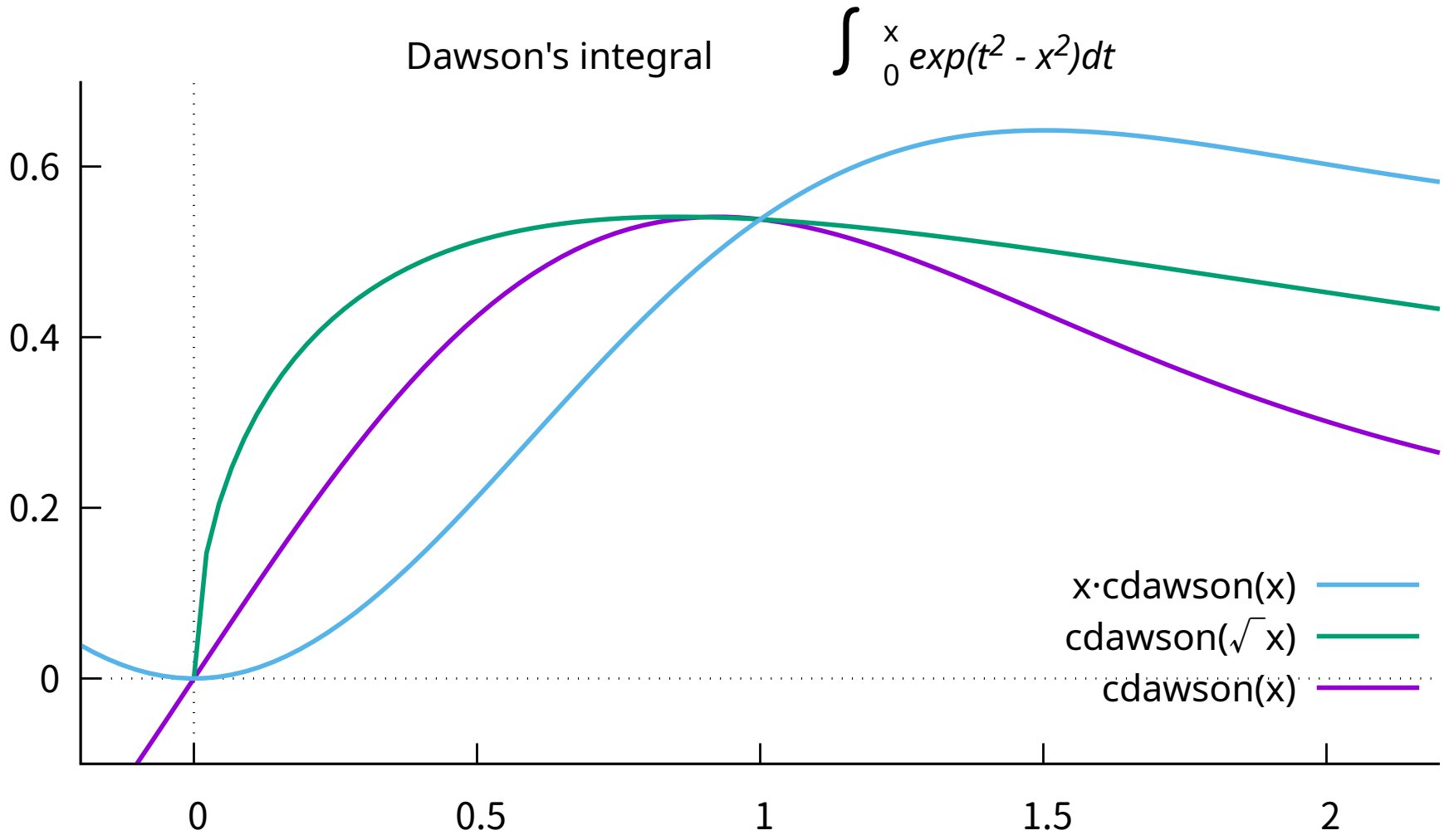

atanh(x + iy)
magnitude
Imaginary
Real
π/2
π/4
0
-π/4
-π/2
-π/2
-π/4
0
π/4
π/2
2π
0
Phase Angle


Voigt Profile VP(x, σ, γ)
σ=1.53 γ=0.00
σ=1.30 γ=0.50
σ=1.00 γ=1.00
σ=0.00 γ=1.80
FWHM
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-4
-2
0
2
4


Faddeeva/Voigt Function
real(faddeeva(x))
imag(faddeeva(x))
Dawson's Integral
1.00
0.50
0.00
-0.50
-1.00
-4
-2
0
2
4

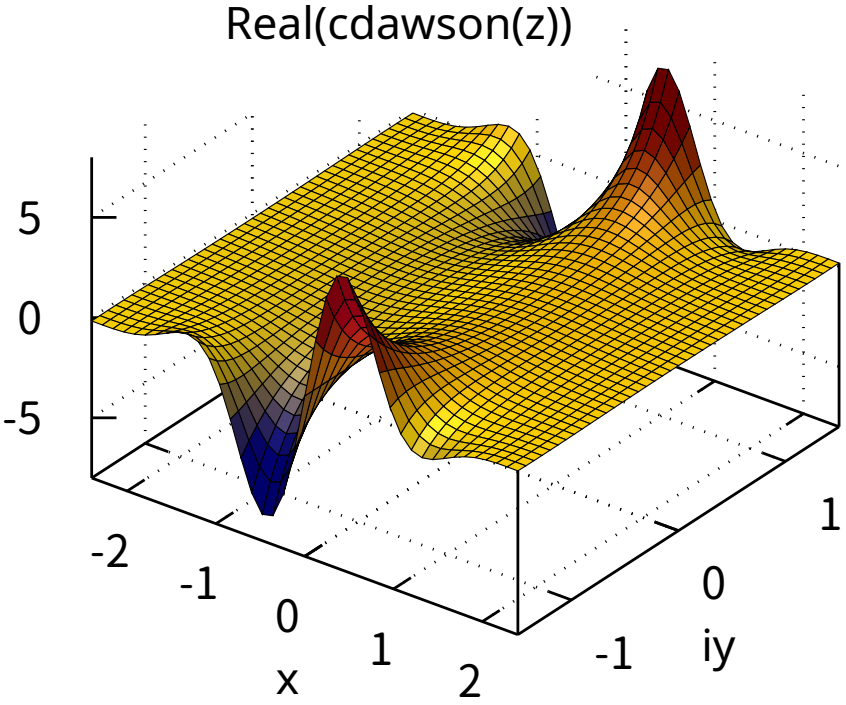
Complex error function cerf(x + iy)

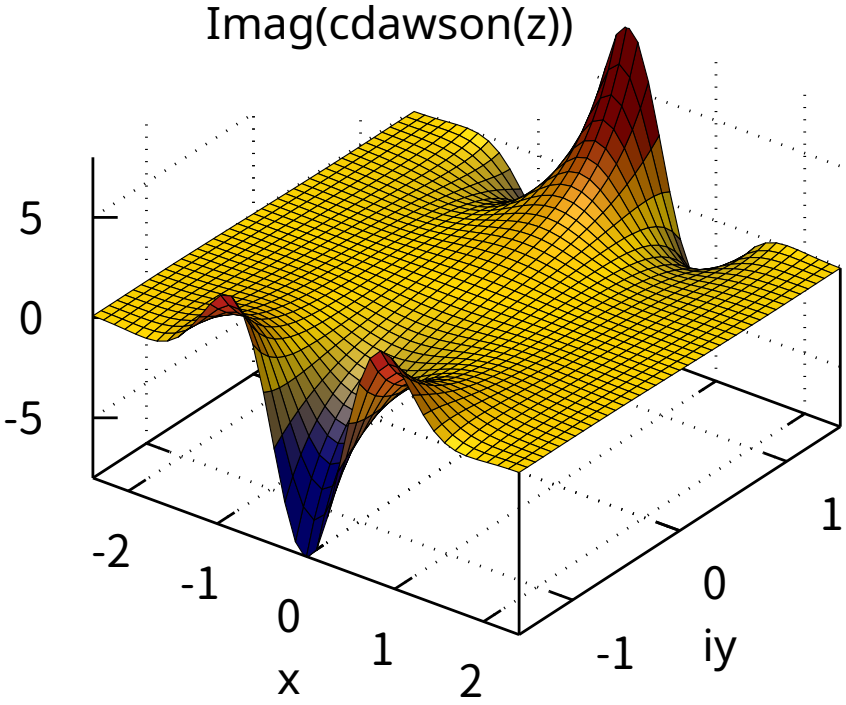


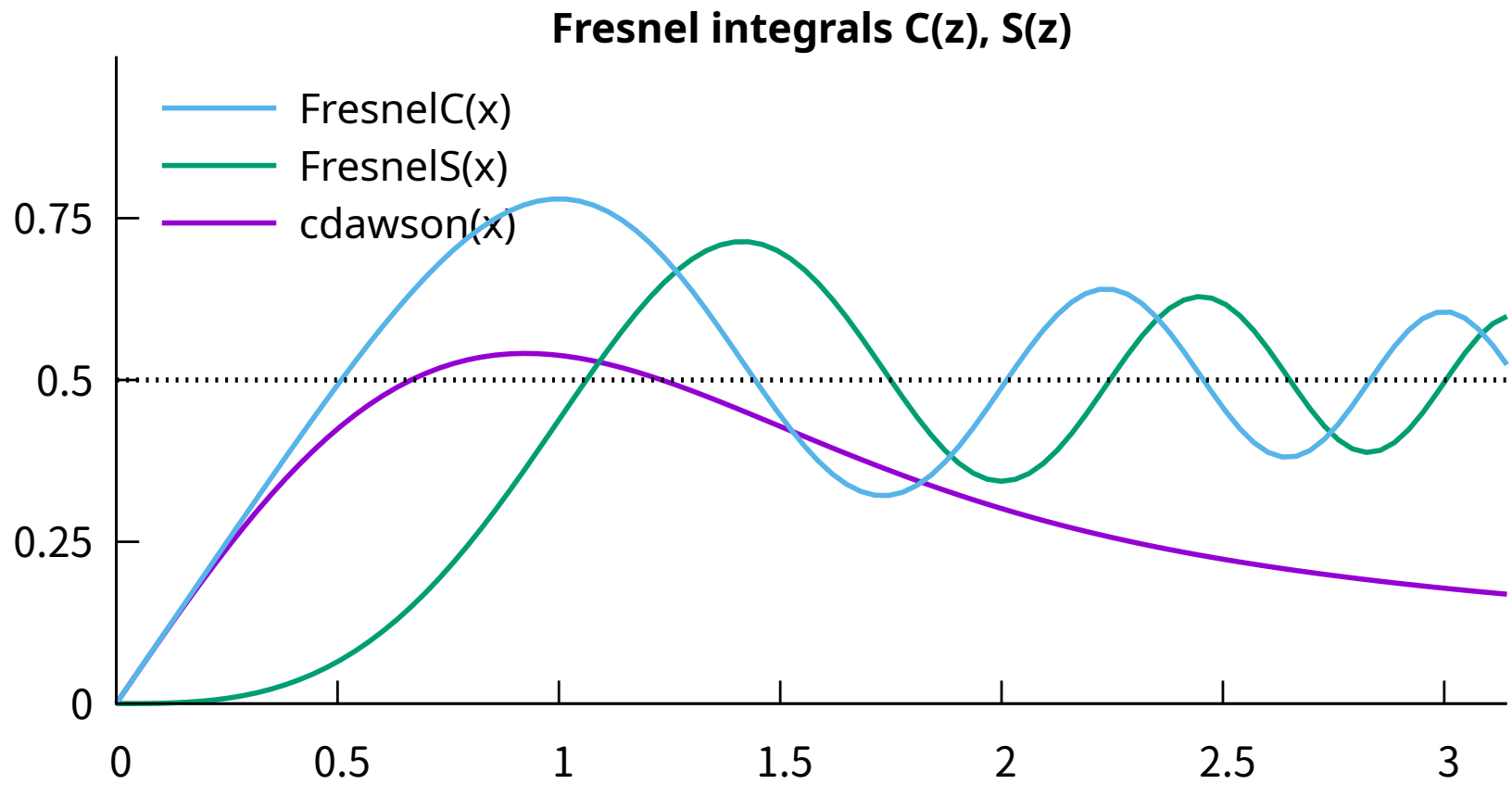

Dawson's integral
$\int_0^x exp(t^2 - x^2)dt$
0.6
0.4
0.2
0
0
0.5
1
1.5
2
x·cdawson(x)
cdawson(√x)
cdawson(x)

Dawson's integral of complex variable

z = x + iy



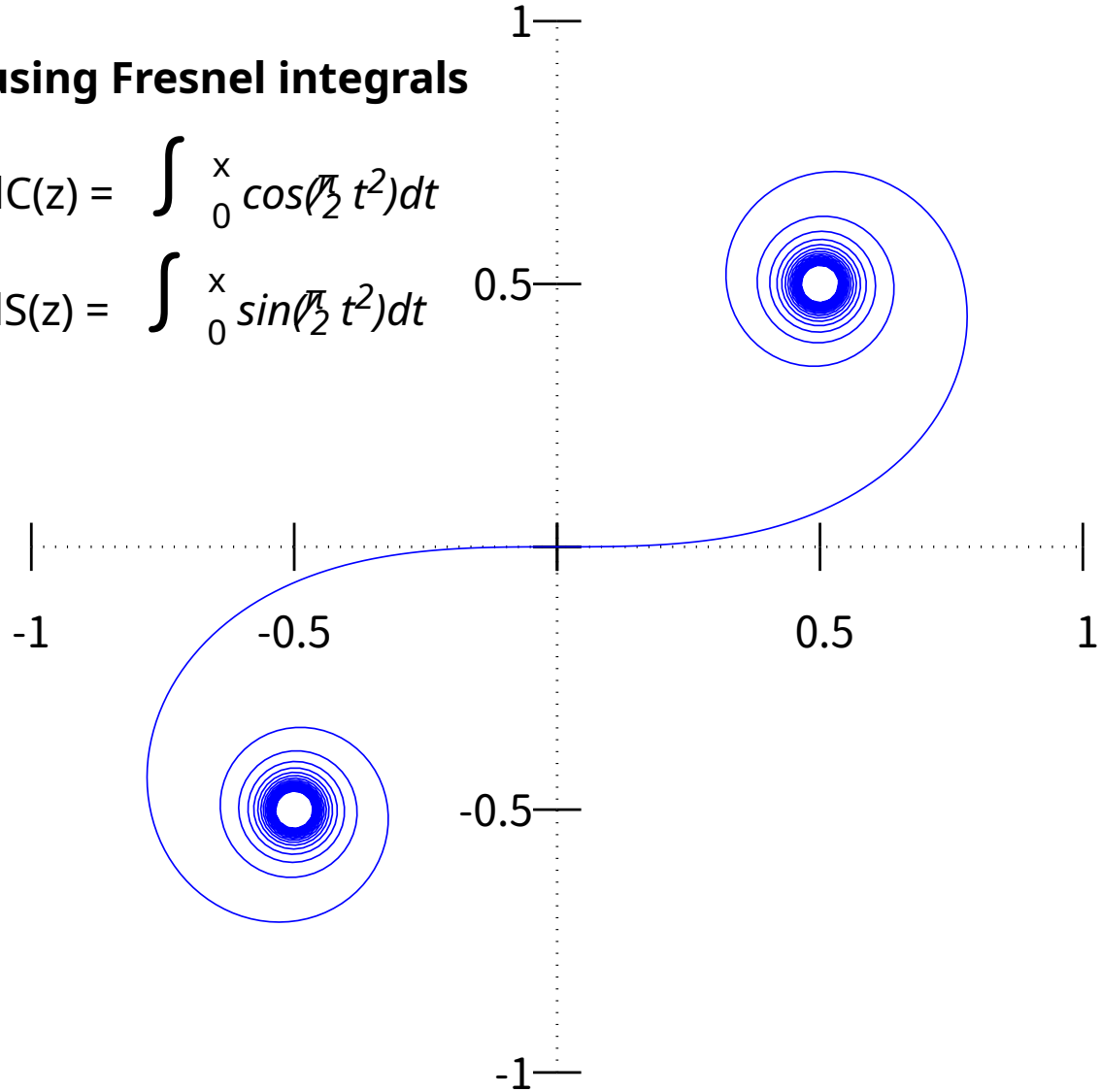


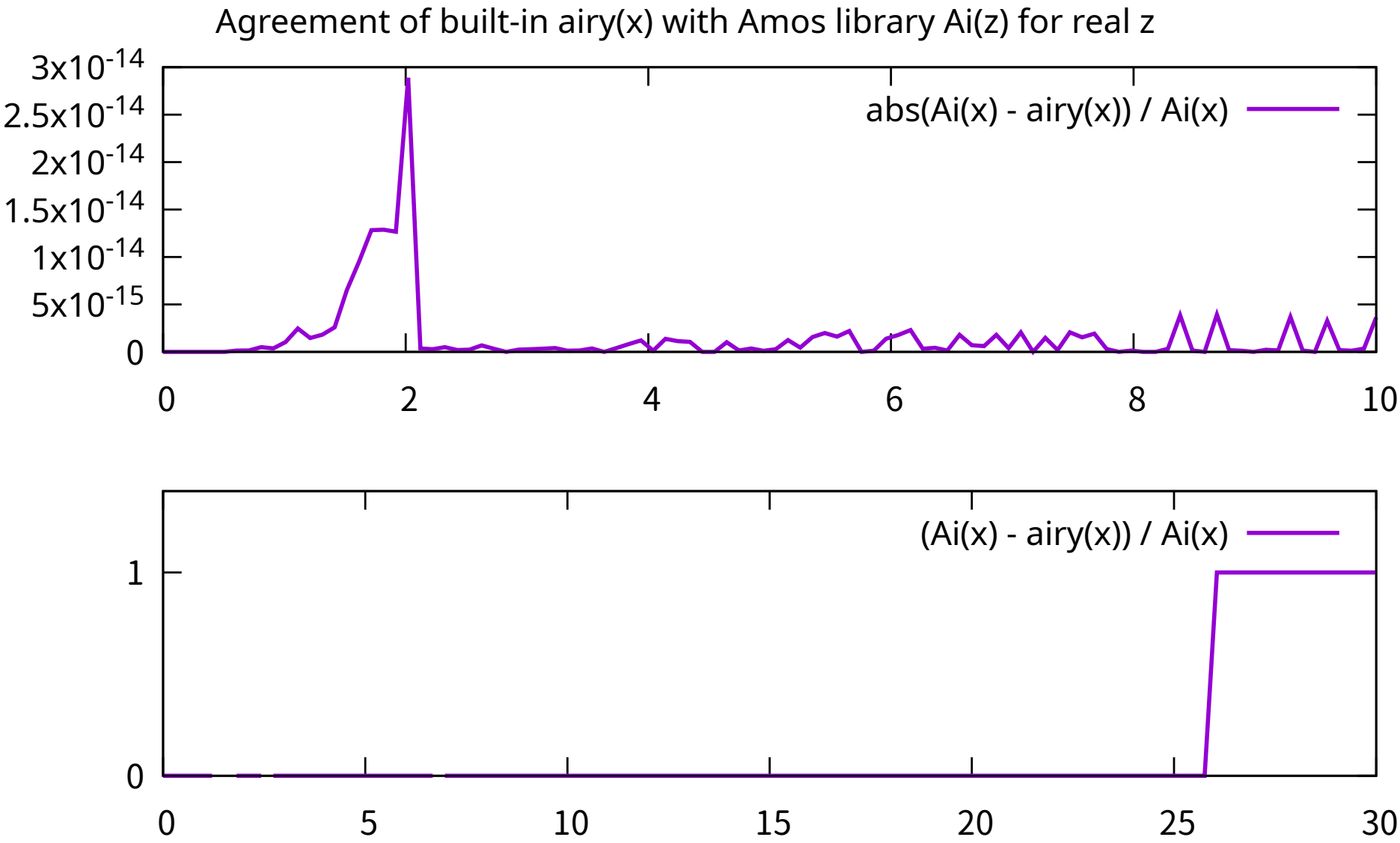

Fresnel integrals C(z), S(z)
FresnelC(x)
FresnelS(x)
cdawson(x)
0.75
0.5
0.25
0
0
0.5
1
1.5
2
2.5
3

Clothoid using Fresnel integrals

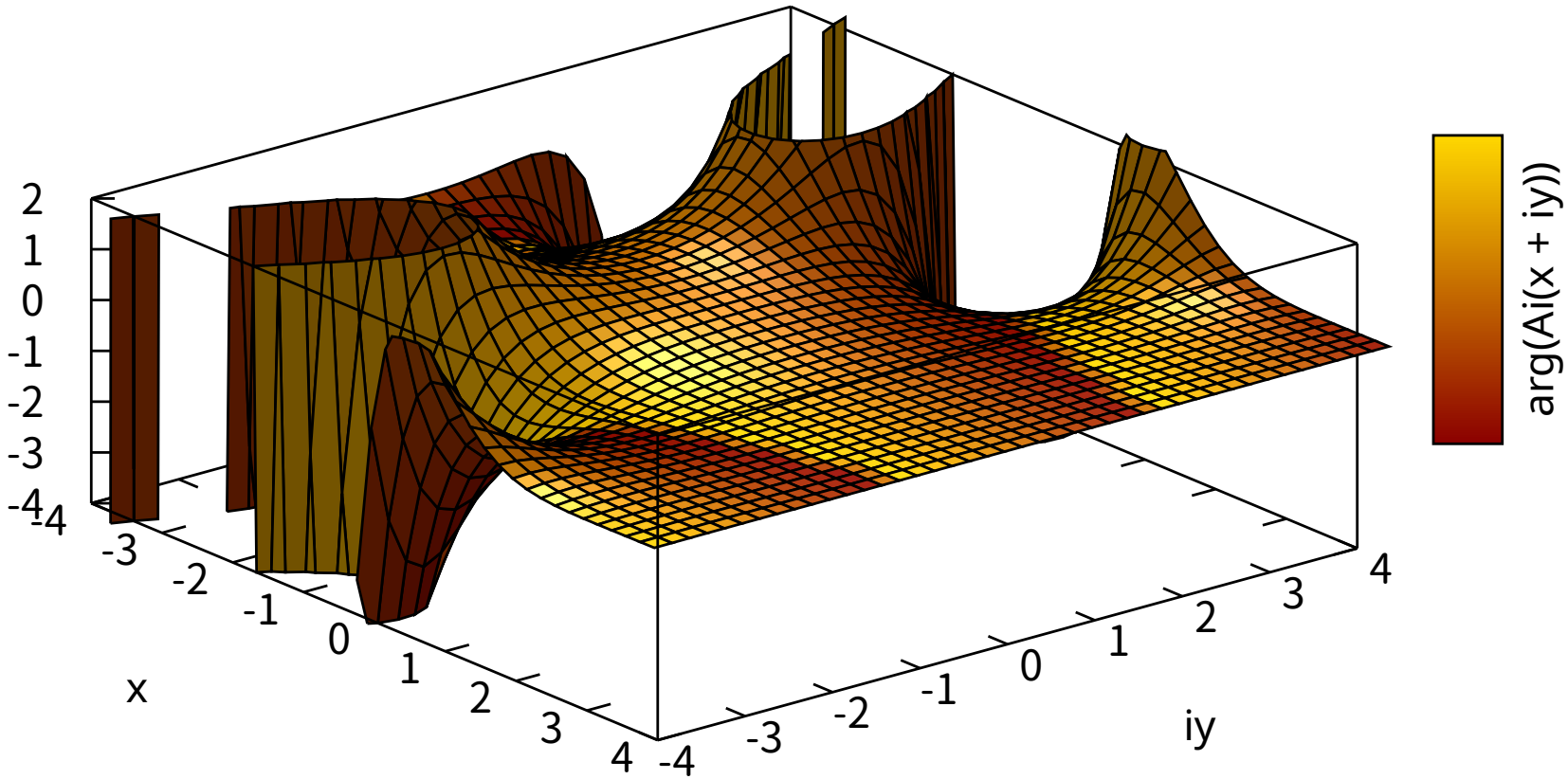
$$x = \text{FresnelC}(z) = \int_0^x \cos\left(\frac{\pi}{2} t^2\right) dt$$

$$y = \text{FresnelS}(z) = \int_0^x \sin\left(\frac{\pi}{2} t^2\right) dt$$

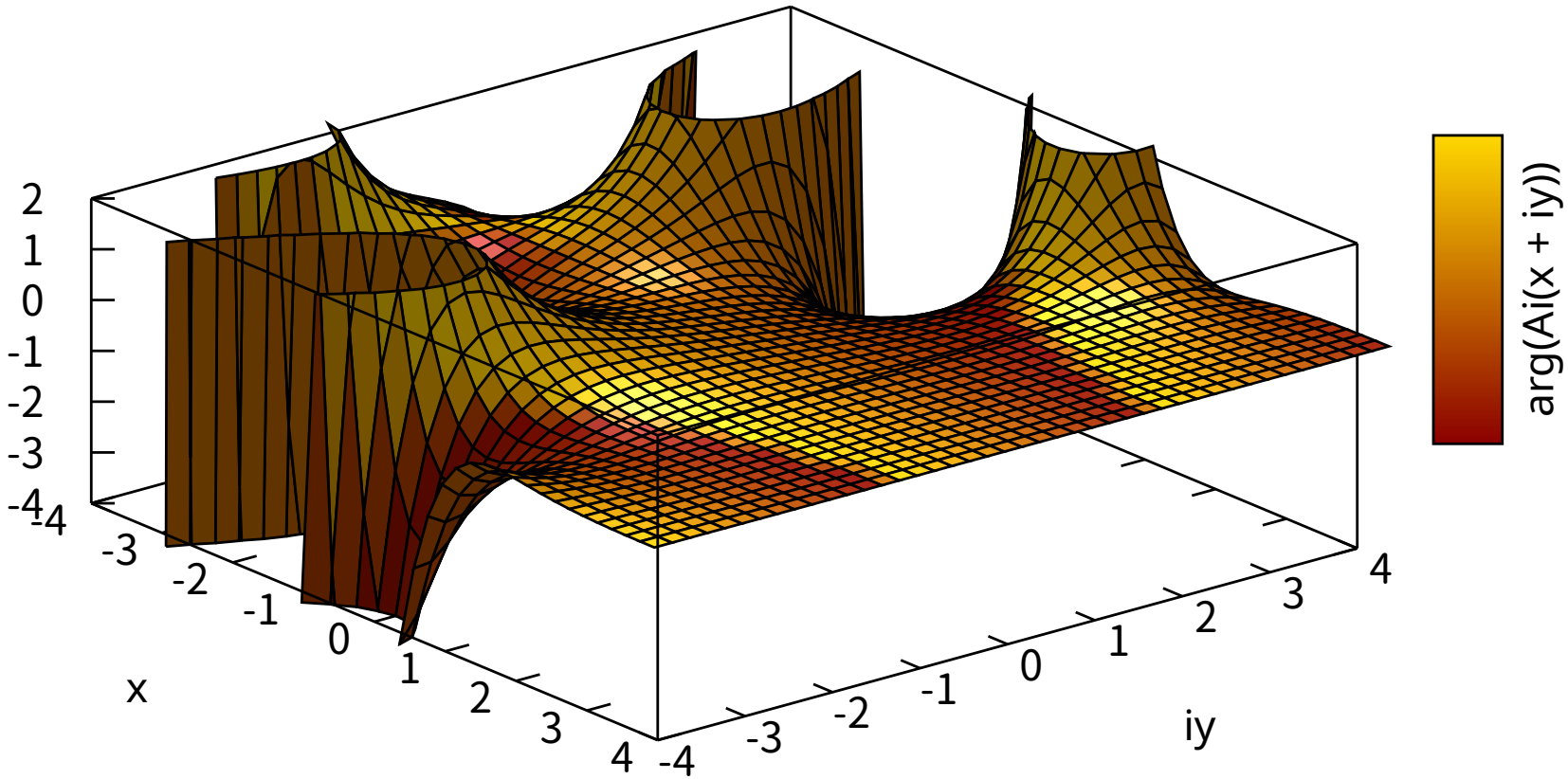


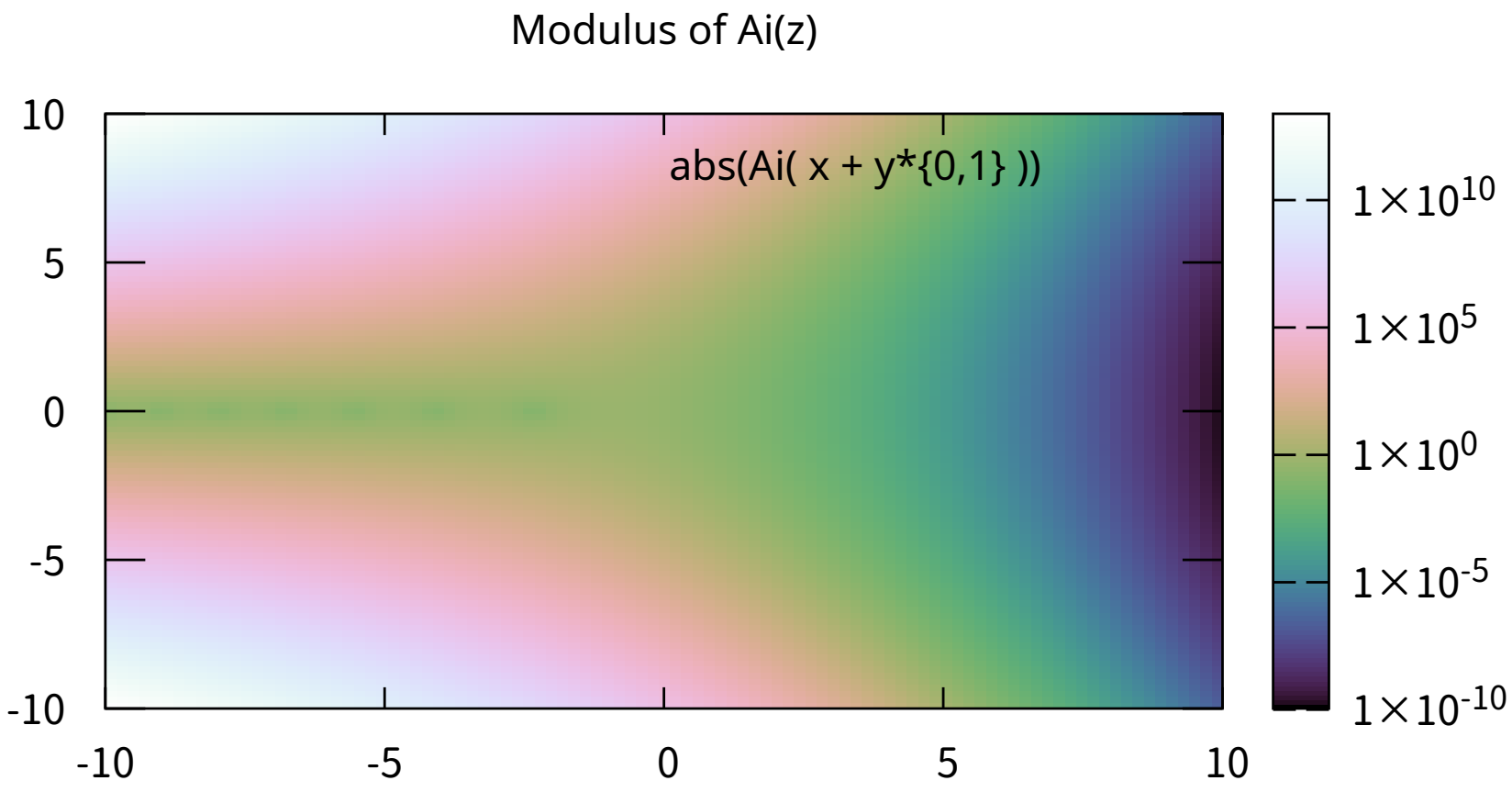

Agreement of built-in airy(x) with Amos library Ai(z) for real z
abs(Ai(x) - airy(x)) / Ai(x)
3x10^-14
2.5x10^-14
2x10^-14
1.5x10^-14
1x10^-14
5x10^-15
0
0
2
4
6
8
10
(Ai(x) - airy(x)) / Ai(x)
1
0
0
5
10
15
20
25
30

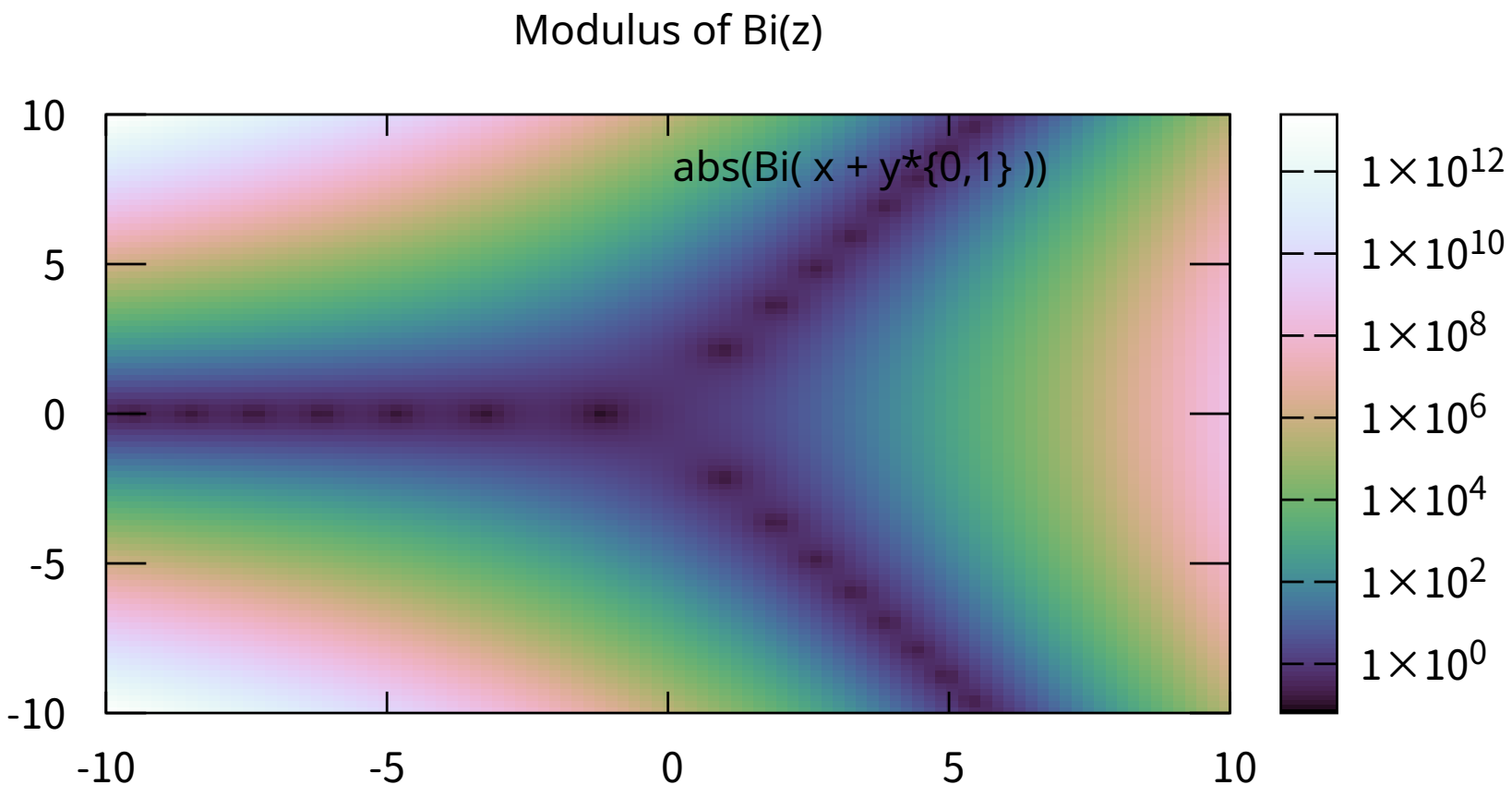
Real(Ai(x+iy))

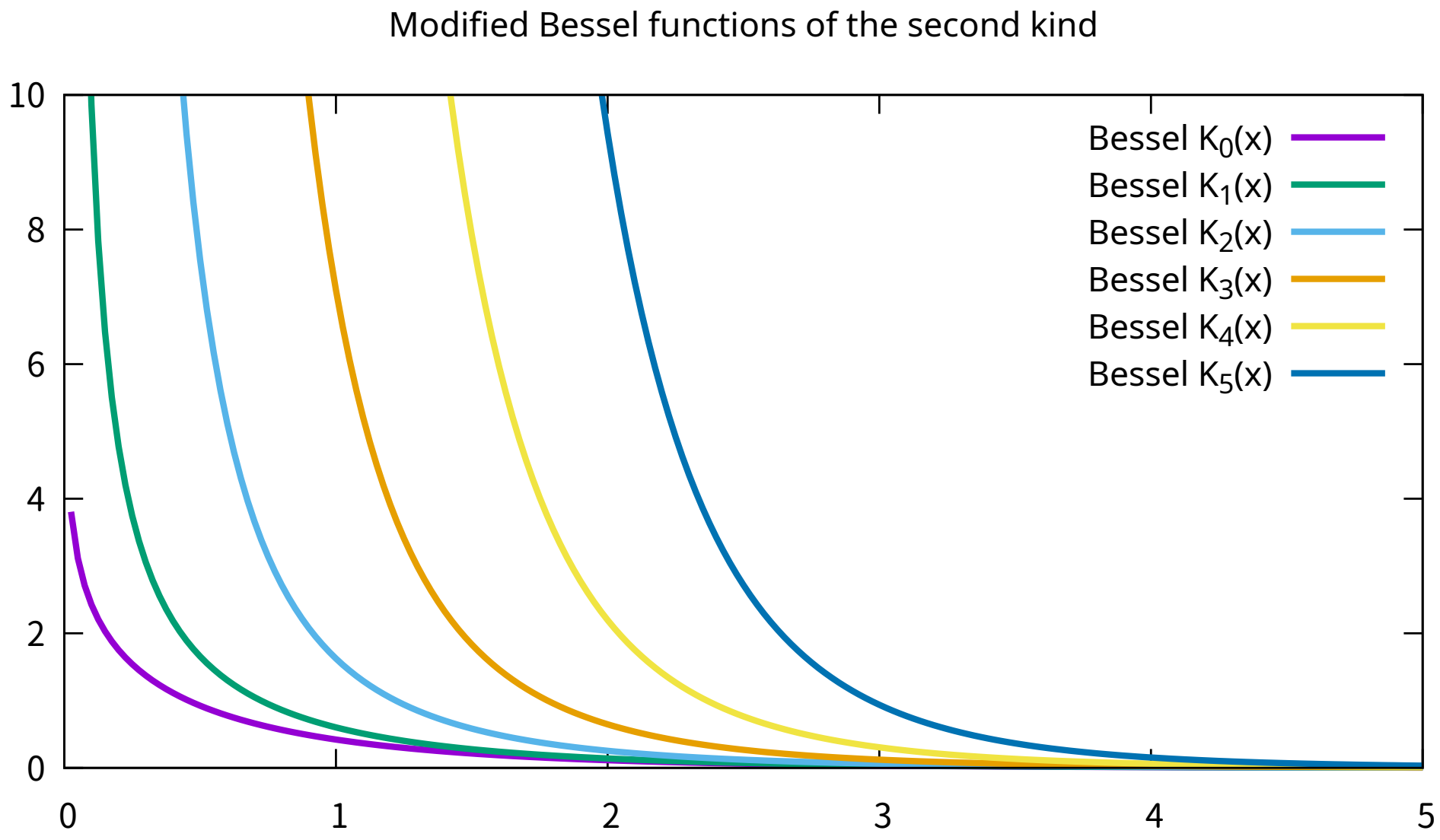


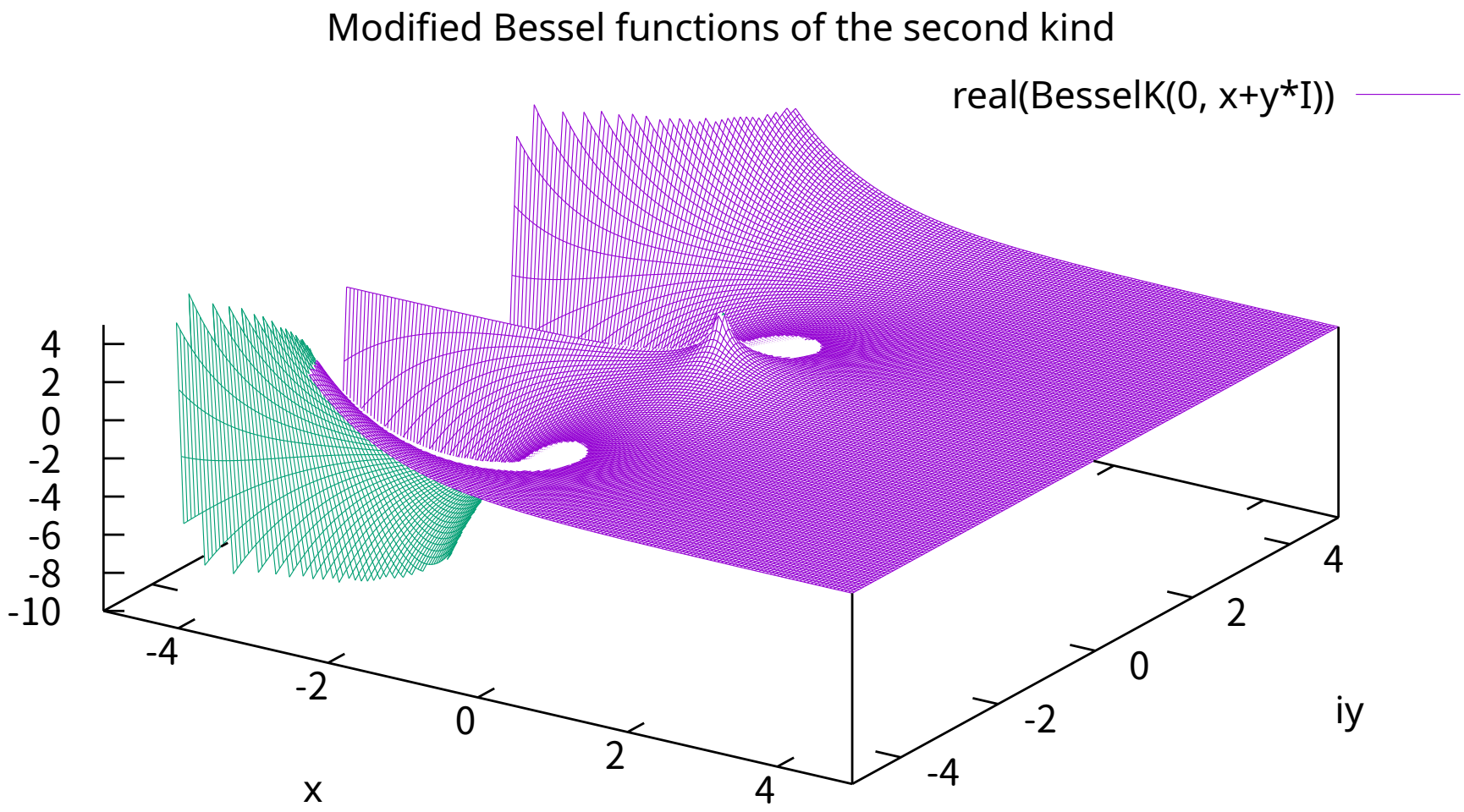
Imag(Ai(x+iy))



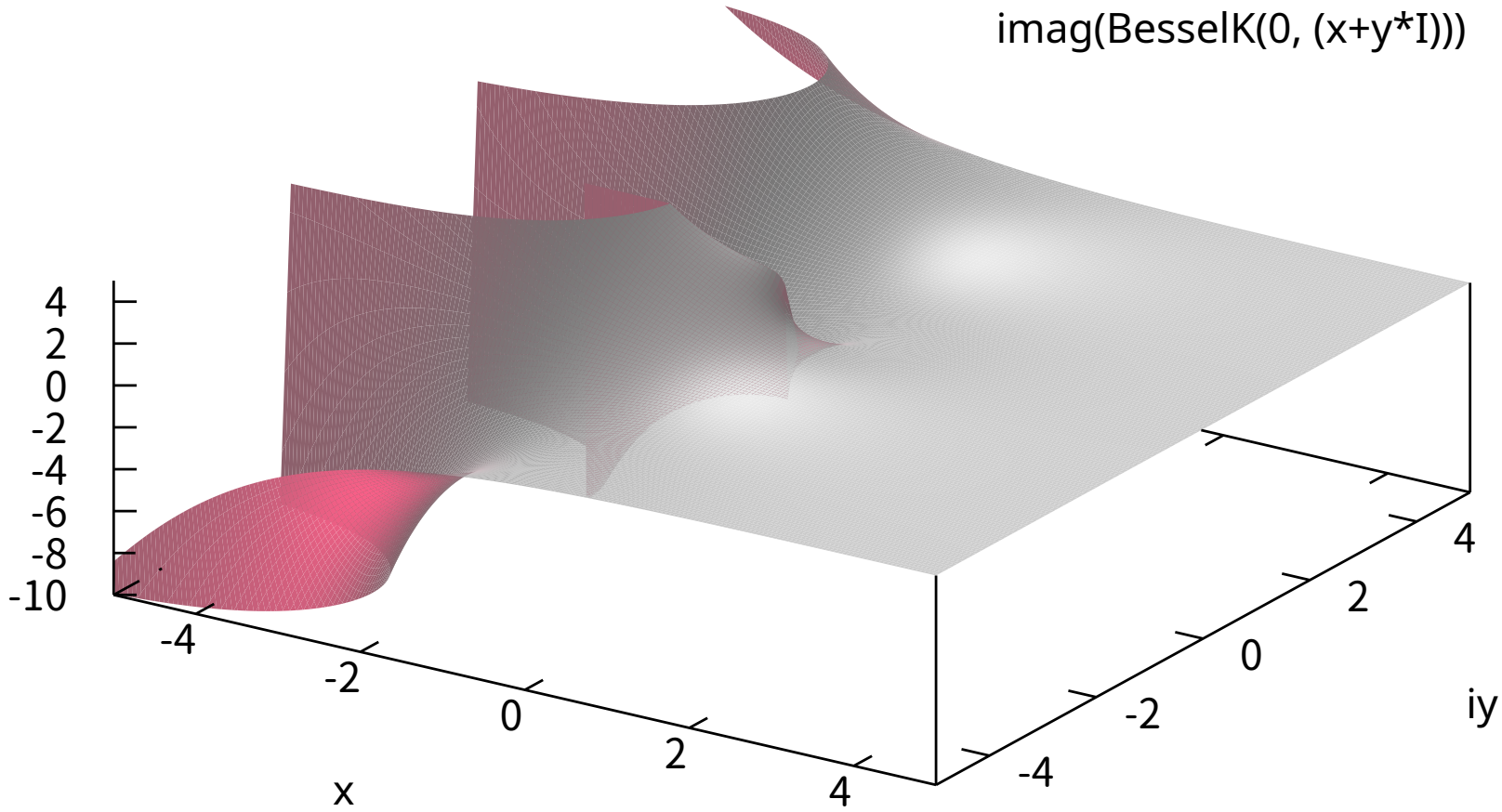

Modulus of Ai(z)
abs(Ai(x + y*{0,1}))
10
5
0
-5
-10
-10
-5
0
5
10
1×10^10
1×10^5
1×10^0
1×10^-5
1×10^-10


Modulus of Bi(z)
abs(Bi(x + y*{0,1}))
10
5
0
-5
-10
-10
-5
0
5
10
1×10^12
1×10^10
1×10^8
1×10^6
1×10^4
1×10^2
1×10^0

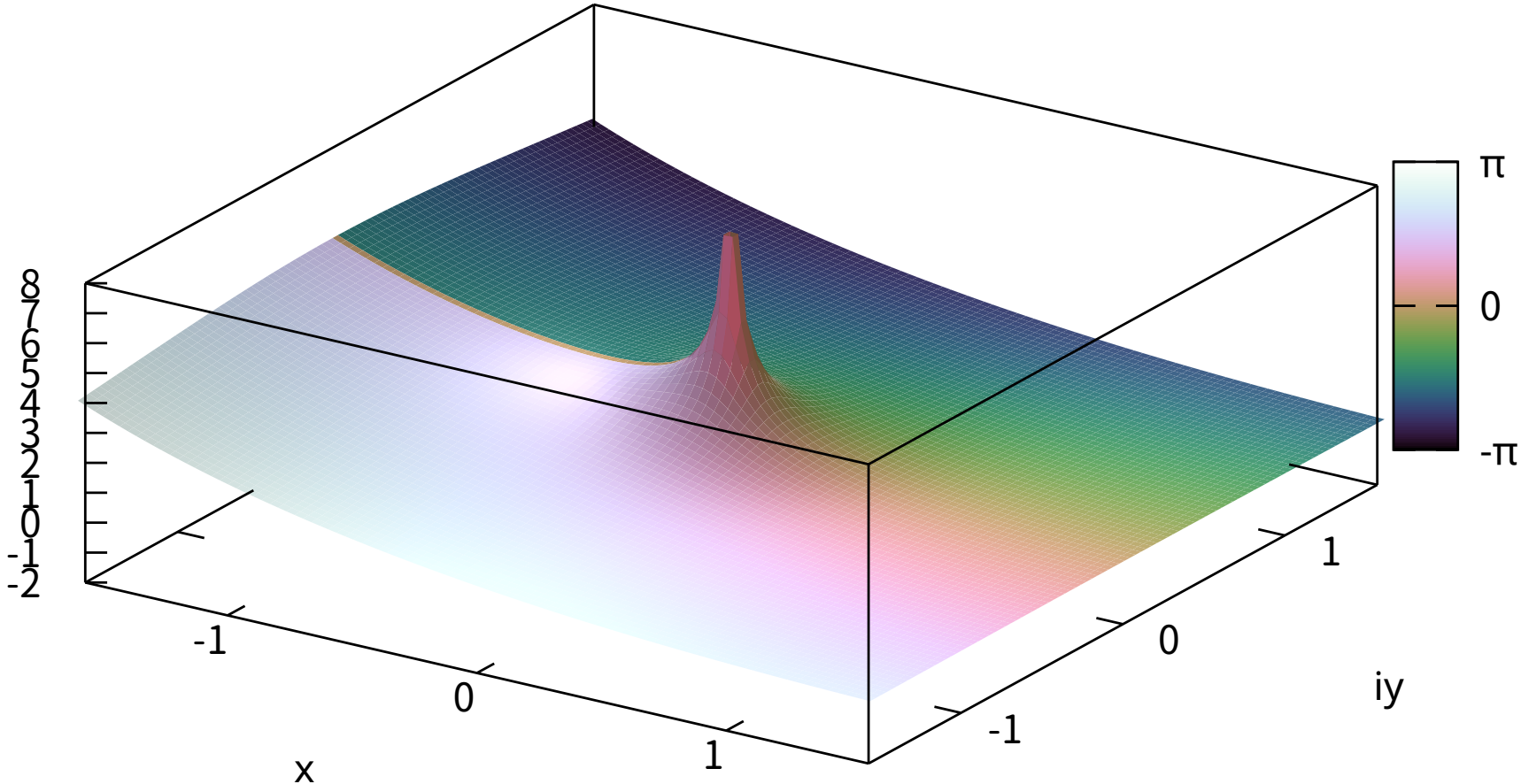

Modified Bessel functions of the second kind
Bessel $K_0(x)$
Bessel $K_1(x)$
Bessel $K_2(x)$
Bessel $K_3(x)$
Bessel $K_4(x)$
Bessel $K_5(x)$
0
2
4
6
8
10
0
1
2
3
4
5

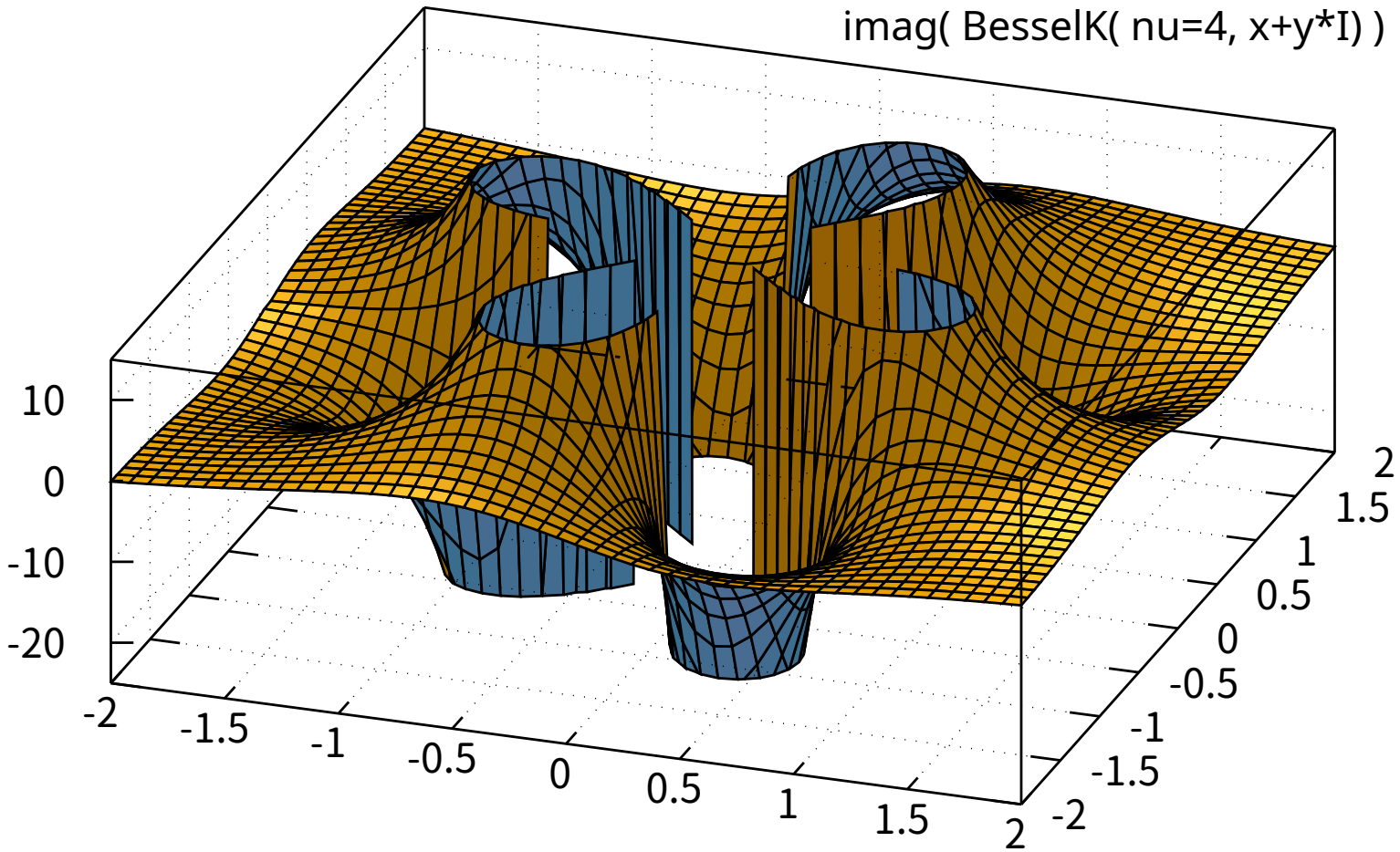

Modified Bessel functions of the second kind
real(BesselK(0, x+y*I))
4
2
0
-2
-4
-6
-8
-10
-4
-2
0
2
4
x
-4
-2
0
2
4
iy

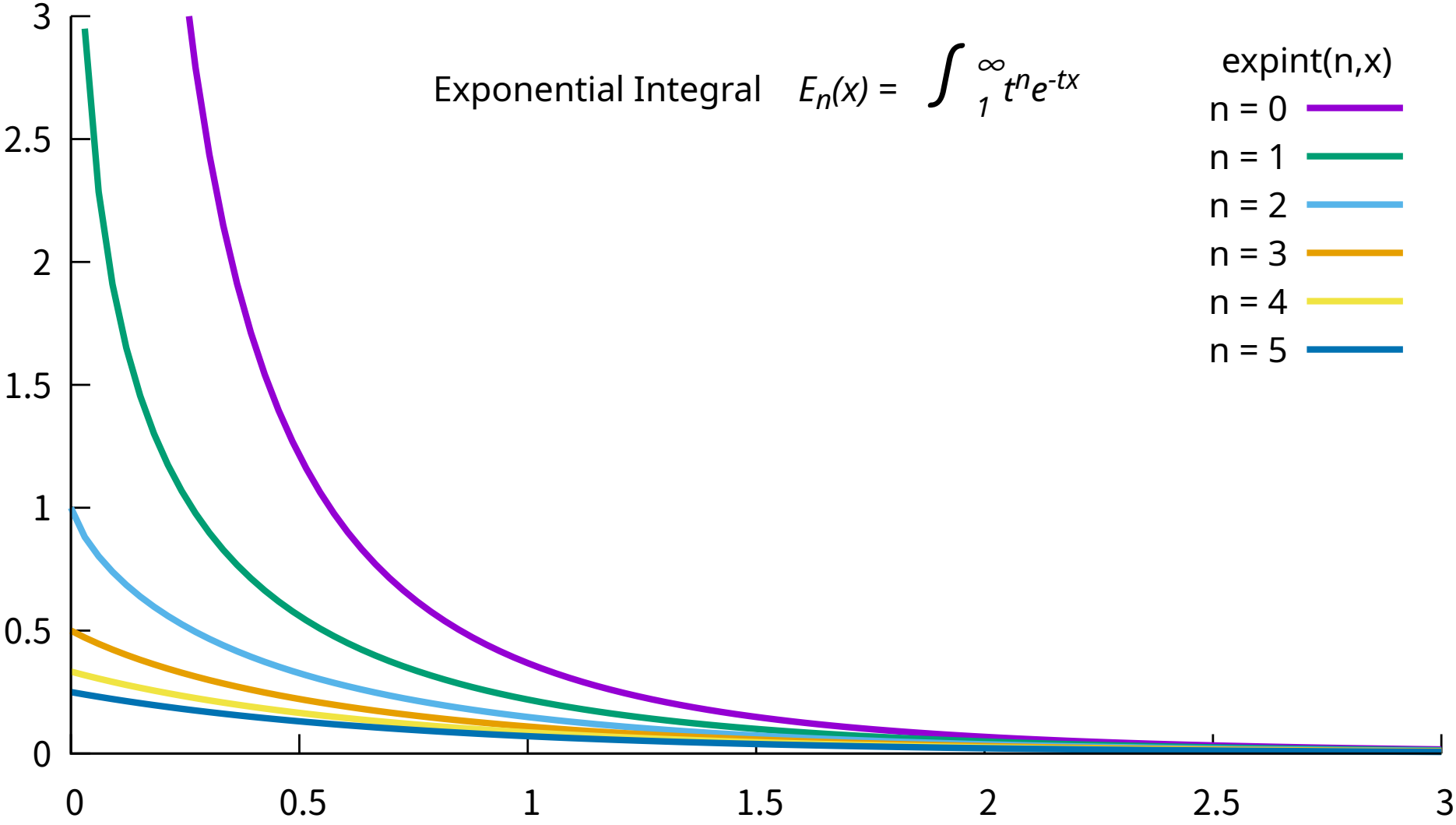
Modified Bessel functions of the second kind

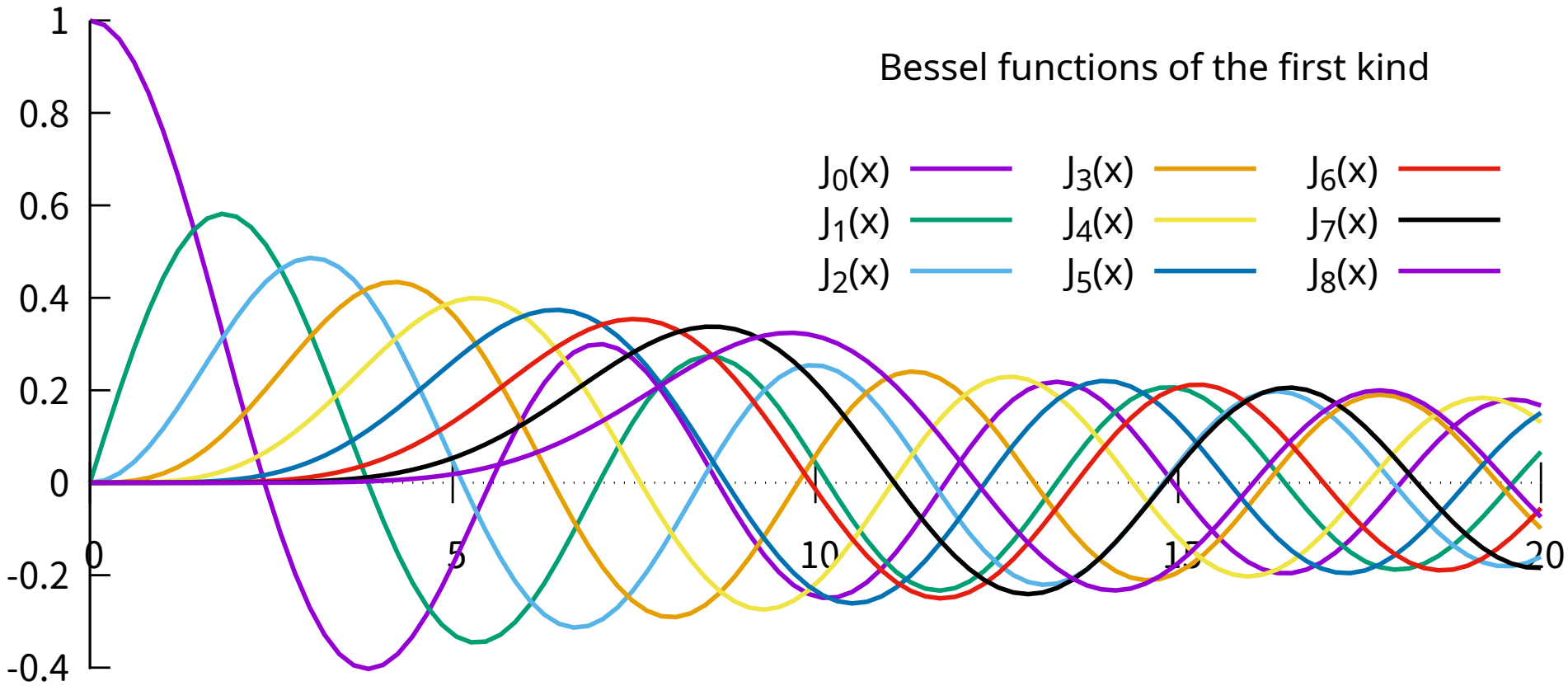


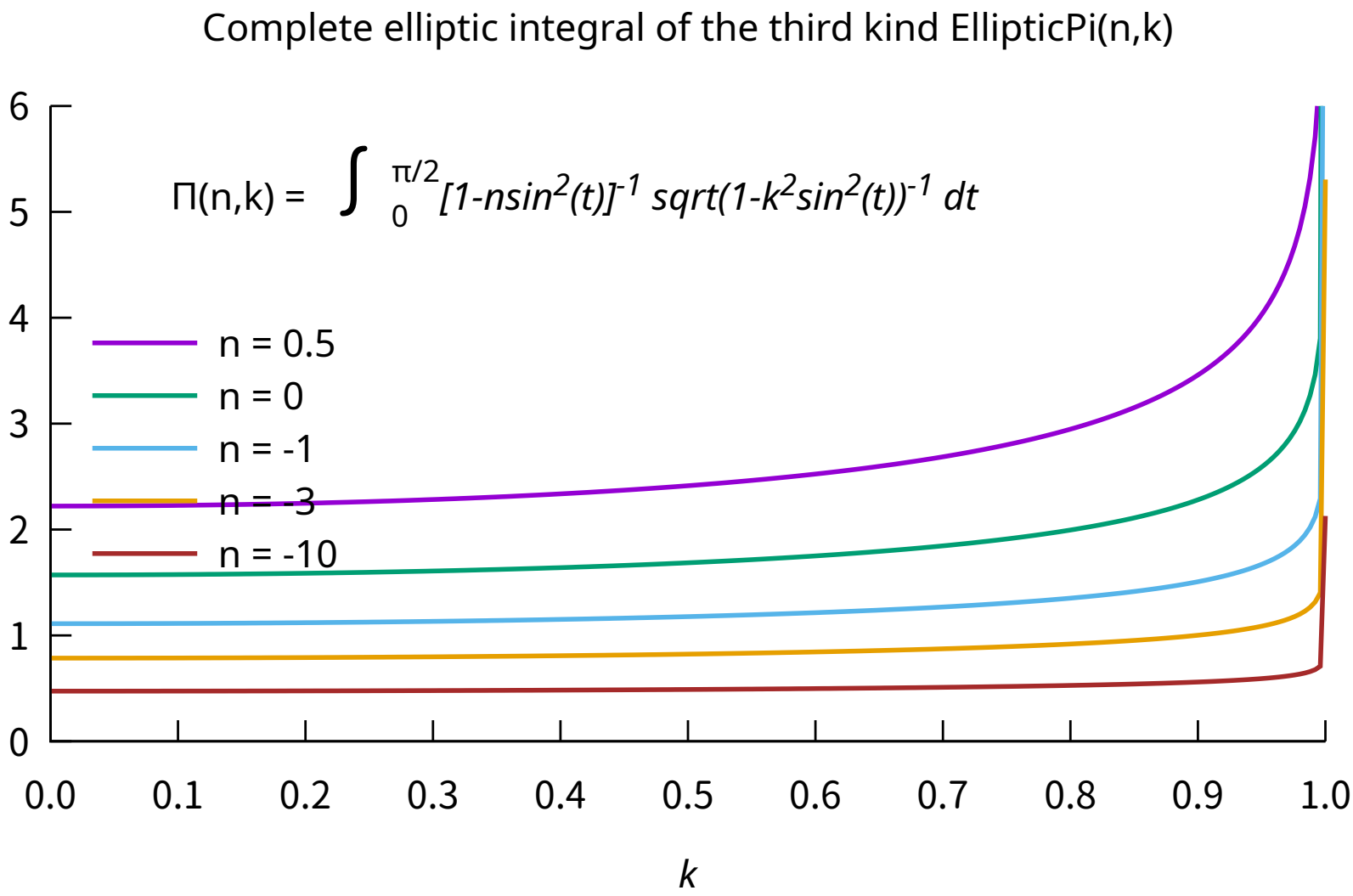
complex Bessel function $|K_{1/2}(x + iy)|$

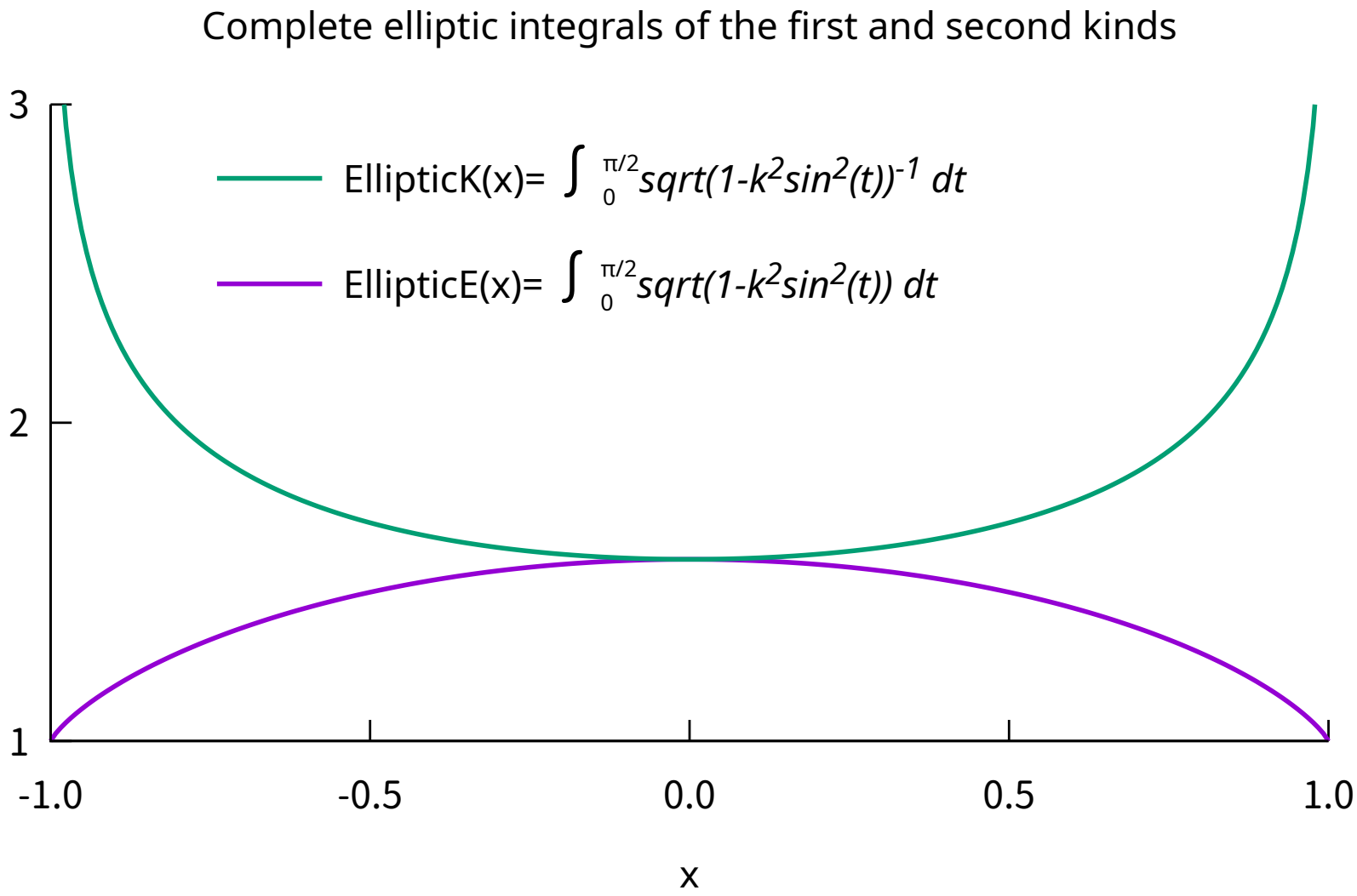



imag(BesselK(nu=4, x+y*I))
10
0
-10
-20
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2

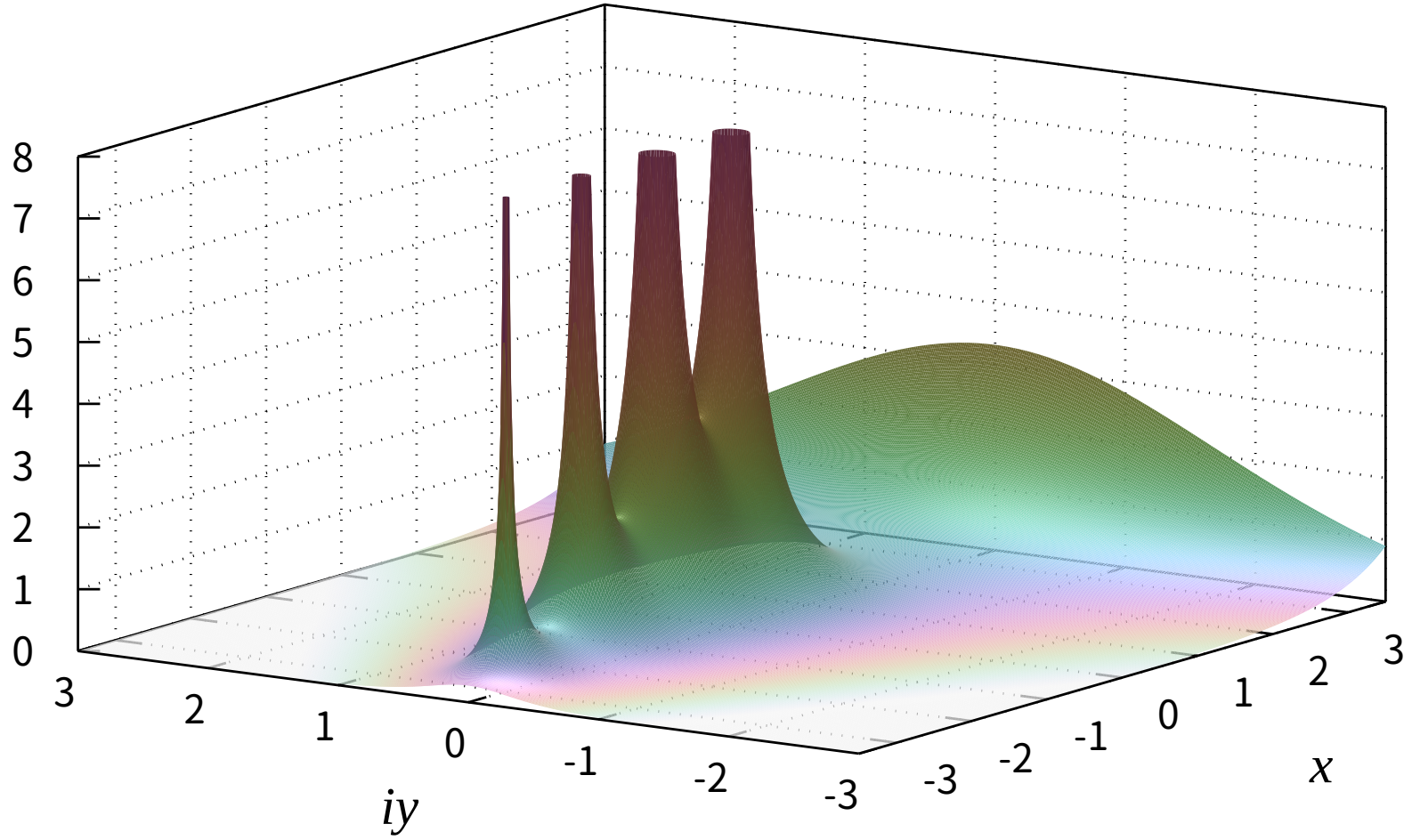

Exponential Integral $E_n(x) = \int_1^\infty t^n e^{-tx}$
expint(n,x)
n = 0
n = 1
n = 2
n = 3
n = 4
n = 5
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3

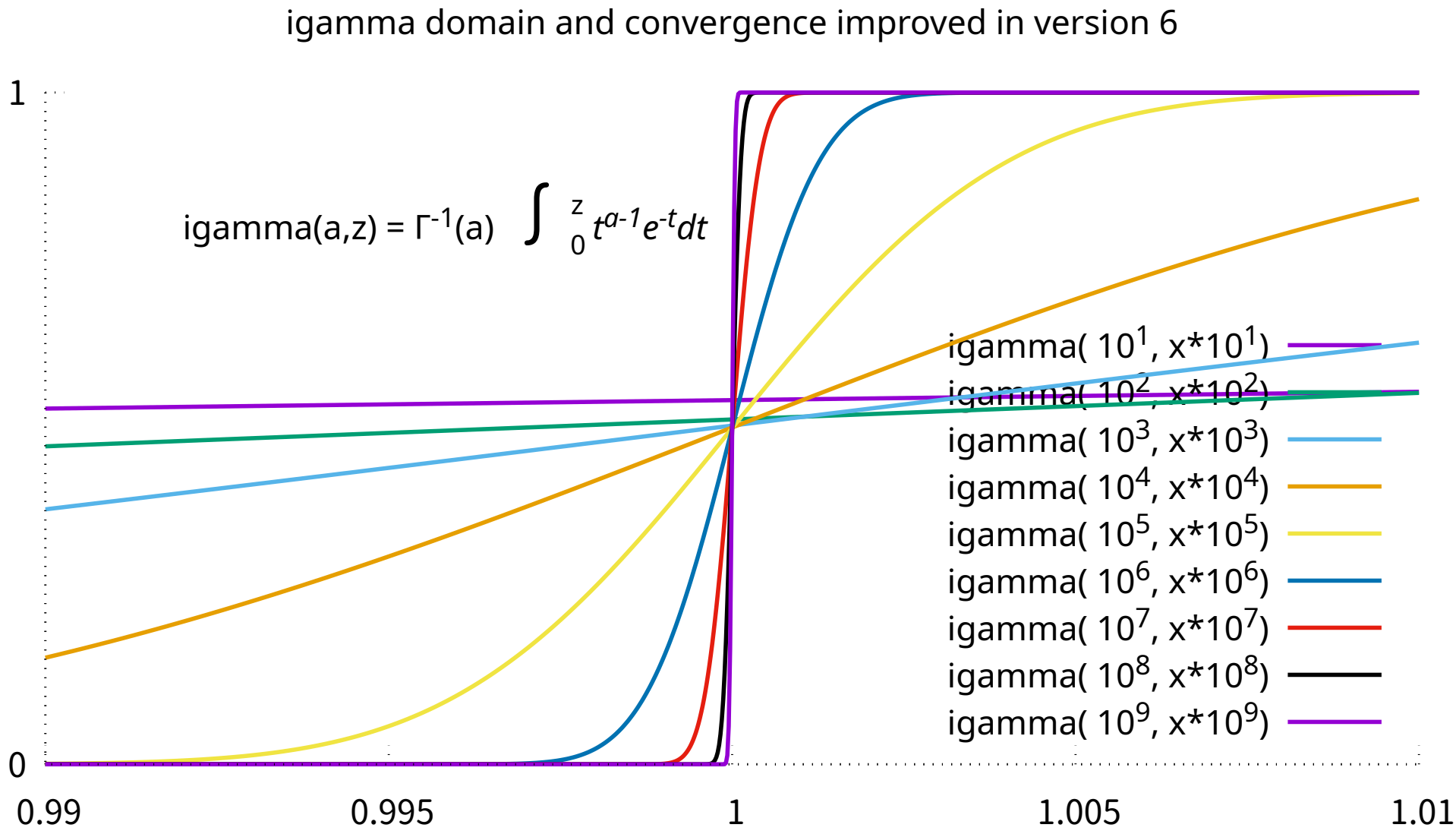

Bessel functions of the first kind
J0(x)
J1(x)
J2(x)
J3(x)
J4(x)
J5(x)
J6(x)
J7(x)
J8(x)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
5
10
15
20

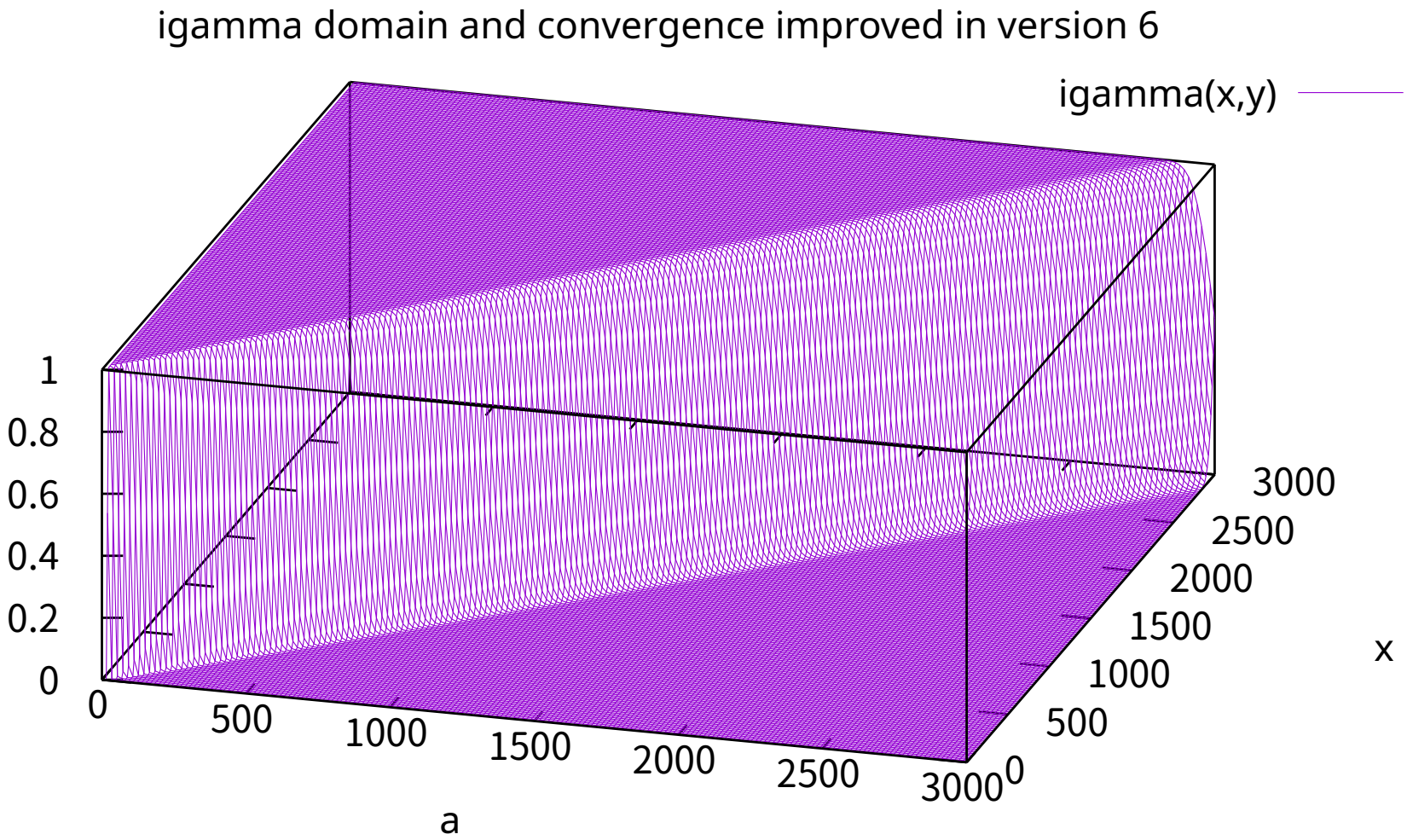

Complete elliptic integral of the third kind EllipticPi(n,k)
Π(n,k) = ∫ π/2 0 [1-nsin2(t)]-1 sqrt(1-k2sin2(t))-1 dt
n = 0.5
n = 0
n = -1
n = -3
n = -10
0
1
2
3
4
5
6
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
k

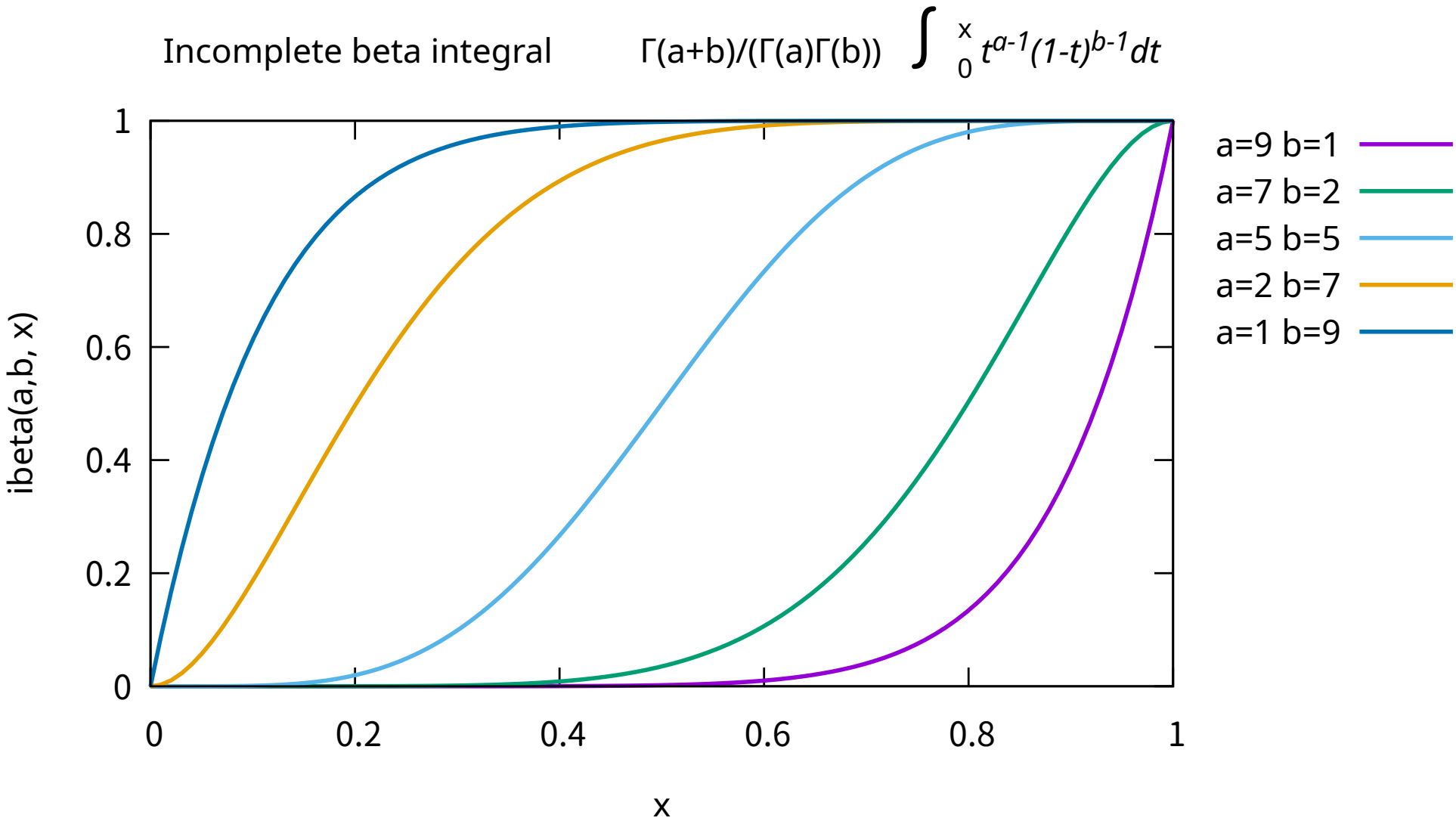

Complete elliptic integrals of the first and second kinds
EllipticK(x)= ∫ π/2 0 sqrt(1-k2sin2(t))-1 dt
EllipticE(x)= ∫ π/2 0 sqrt(1-k2sin2(t)) dt
3
2
1
-1.0
-0.5
0.0
0.5
1.0
x

$\Gamma(x+iy)$ = exp(lnGamma(x + y*I)

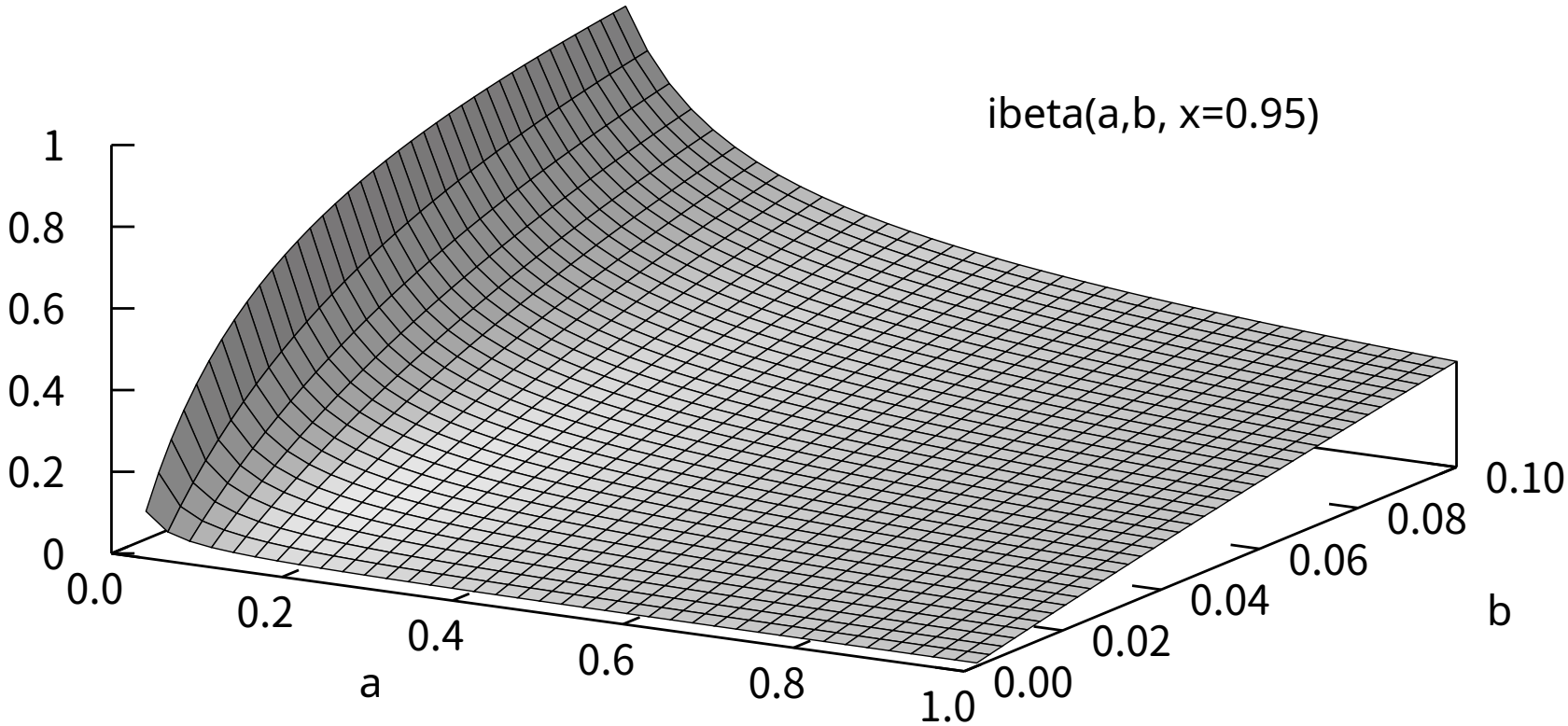


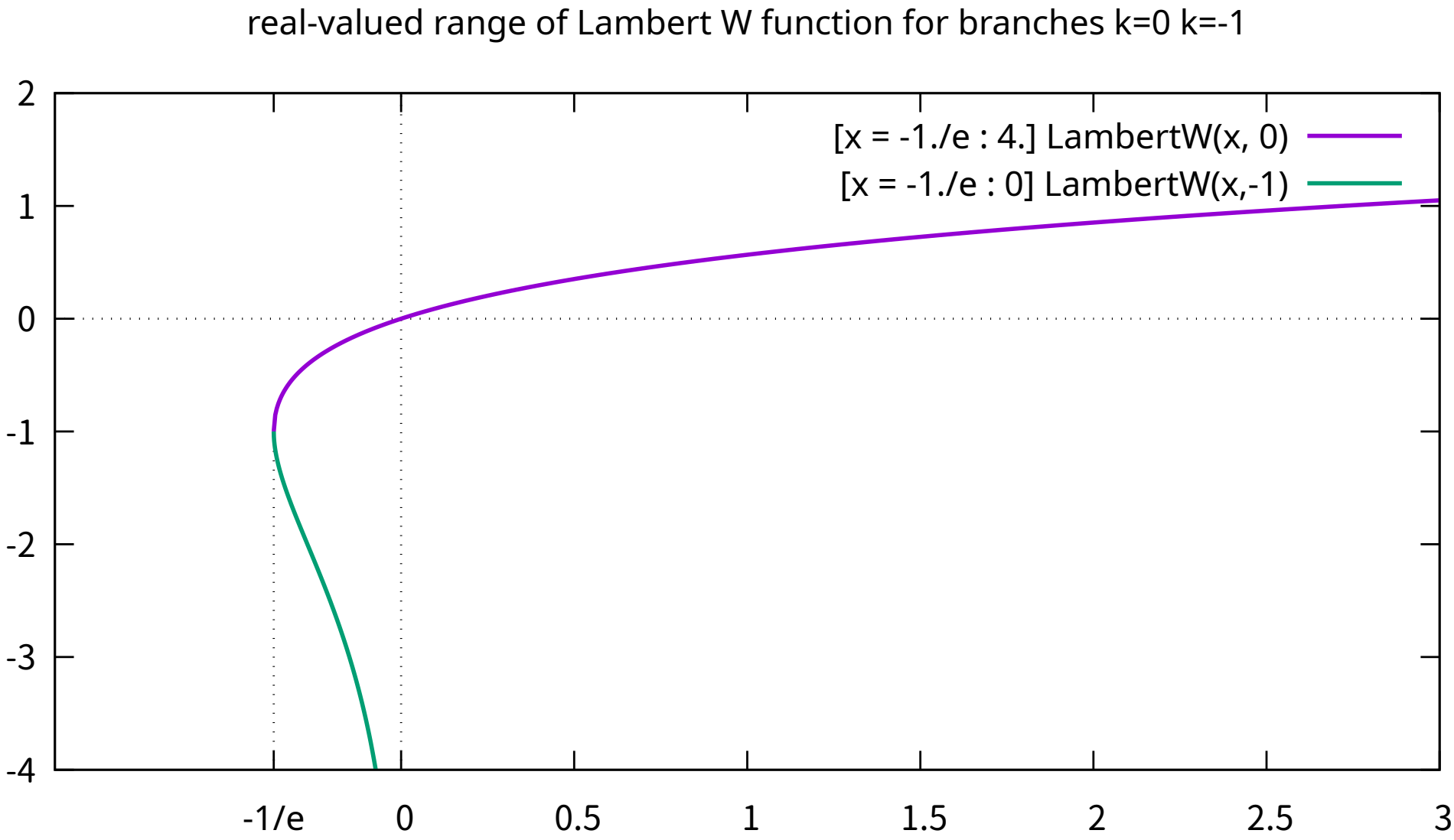

igamma domain and convergence improved in version 6
igamma(a,z) = Γ-1(a) ∫ z 0 ta-1e-tdt
1
0
0.99
0.995
1
1.005
1.01
igamma(101, x*101)
igamma(102, x*102)
igamma(103, x*103)
igamma(104, x*104)
igamma(105, x*105)
igamma(106, x*106)
igamma(107, x*107)
igamma(108, x*108)
igamma(109, x*109)

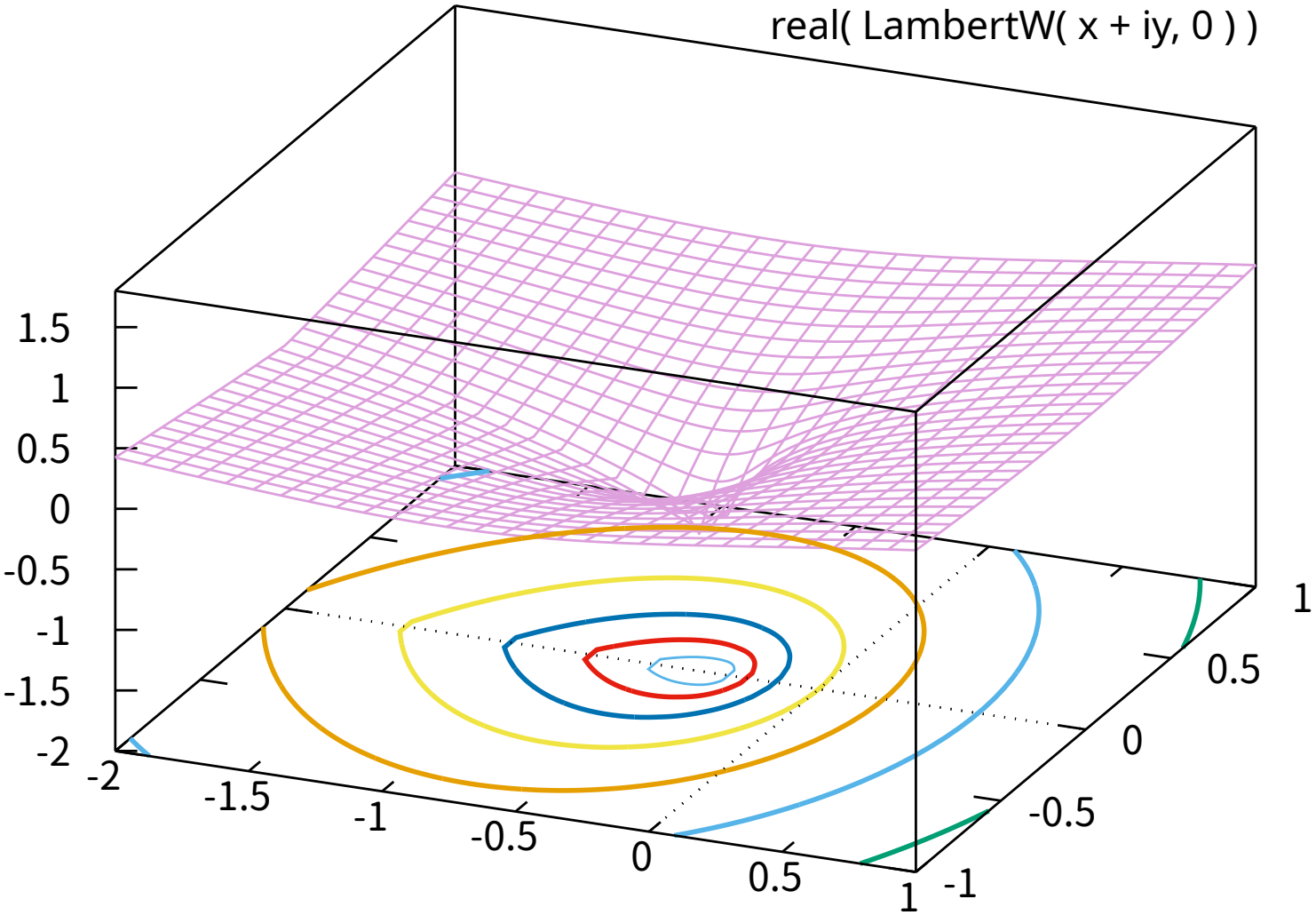

igamma domain and convergence improved in version 6
igamma(x,y)
1
0.8
0.6
0.4
0.2
0
0
500
1000
1500
2000
2500
3000
a
0
500
1000
1500
2000
2500
3000
x

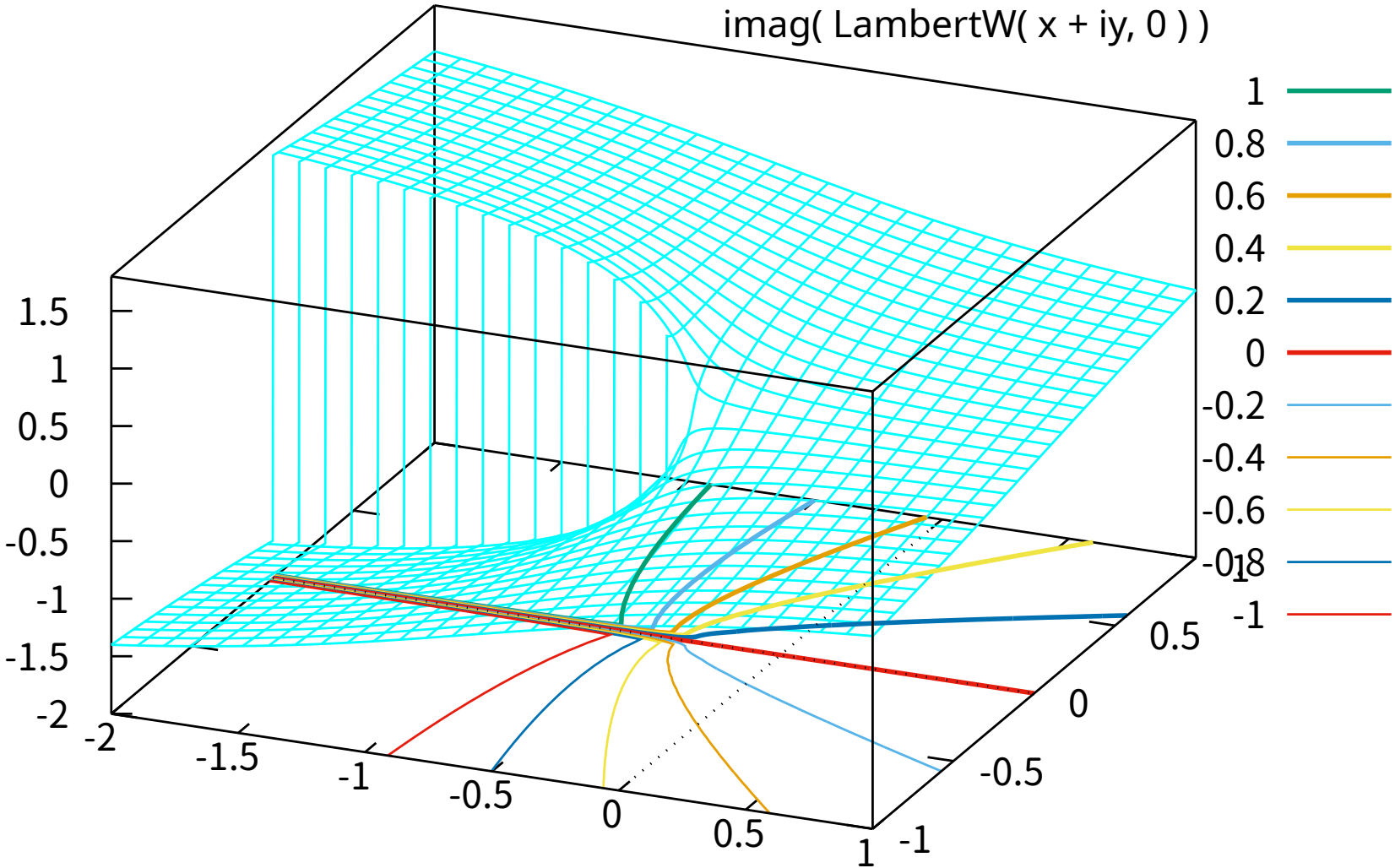

Incomplete beta integral
Γ(a+b)/(Γ(a)Γ(b)) ∫ x 0 t^(a-1)(1-t)^(b-1)dt
ibeta(a,b, x)
x
0
0.2
0.4
0.6
0.8
1
a=9 b=1
a=7 b=2
a=5 b=5
a=2 b=7
a=1 b=9

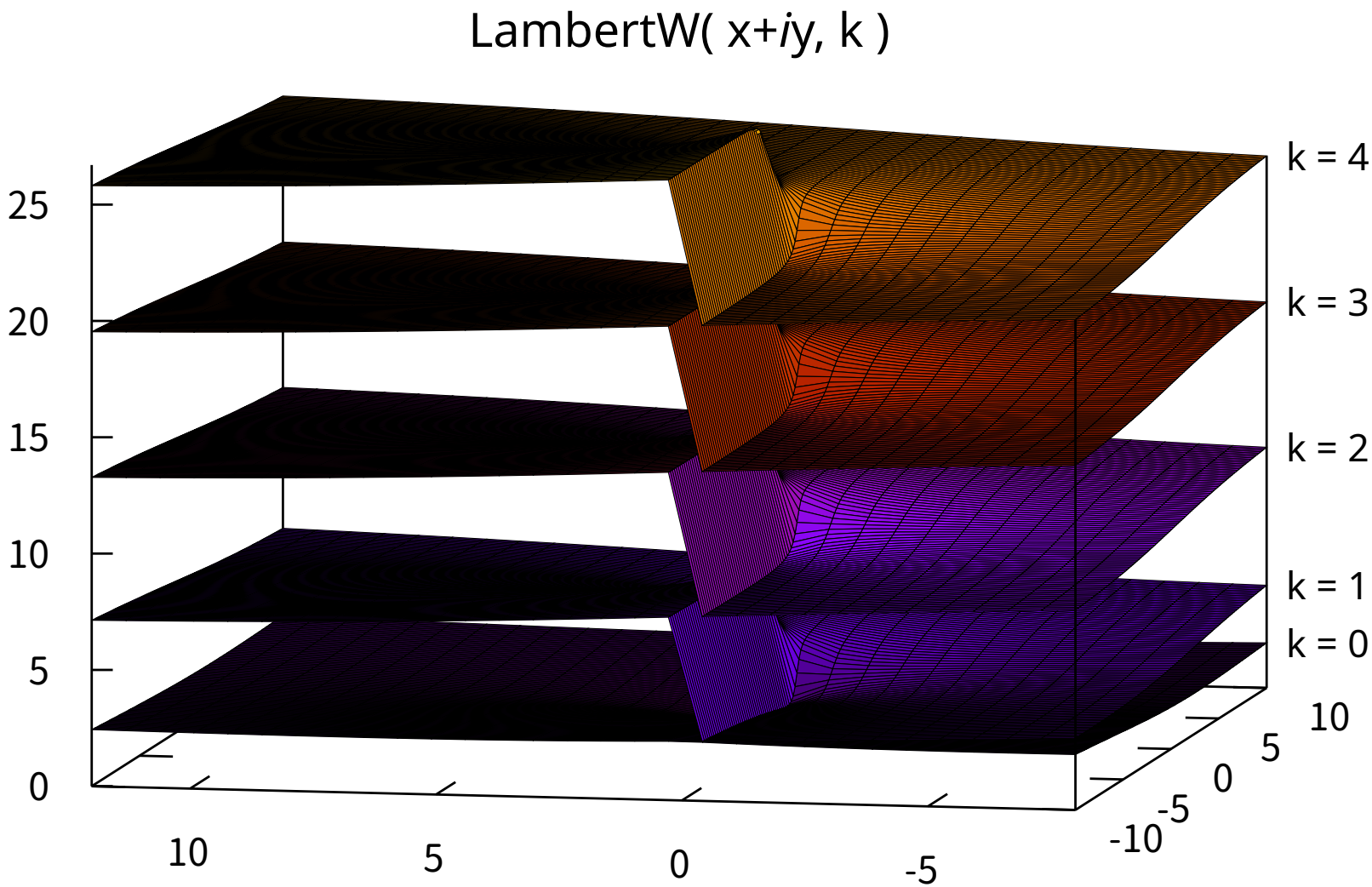
Incomplete beta integral $\Gamma(a+b)/(\Gamma(a)\Gamma(b)) \int_0^x t^{a-1}(1-t)^{b-1}dt$

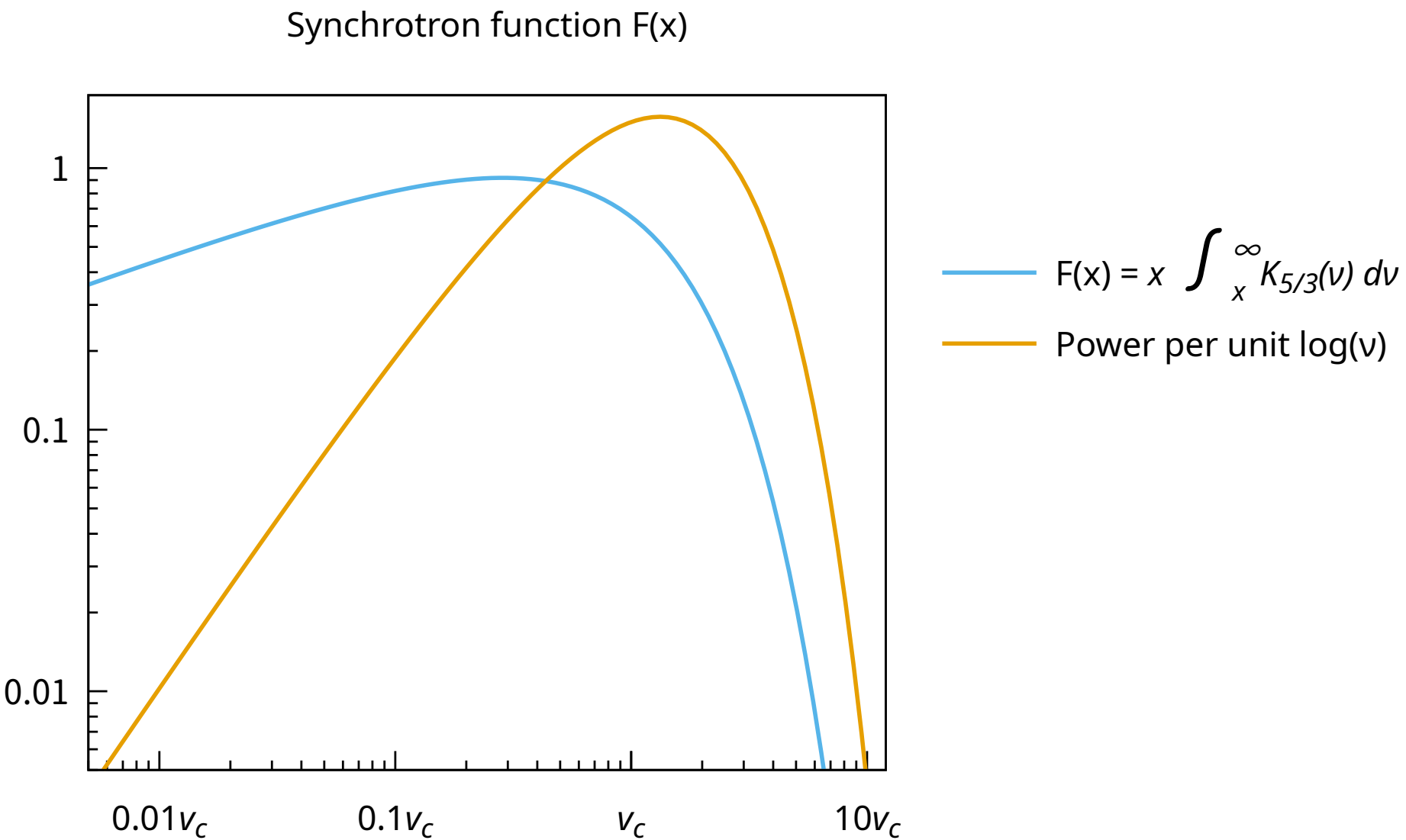


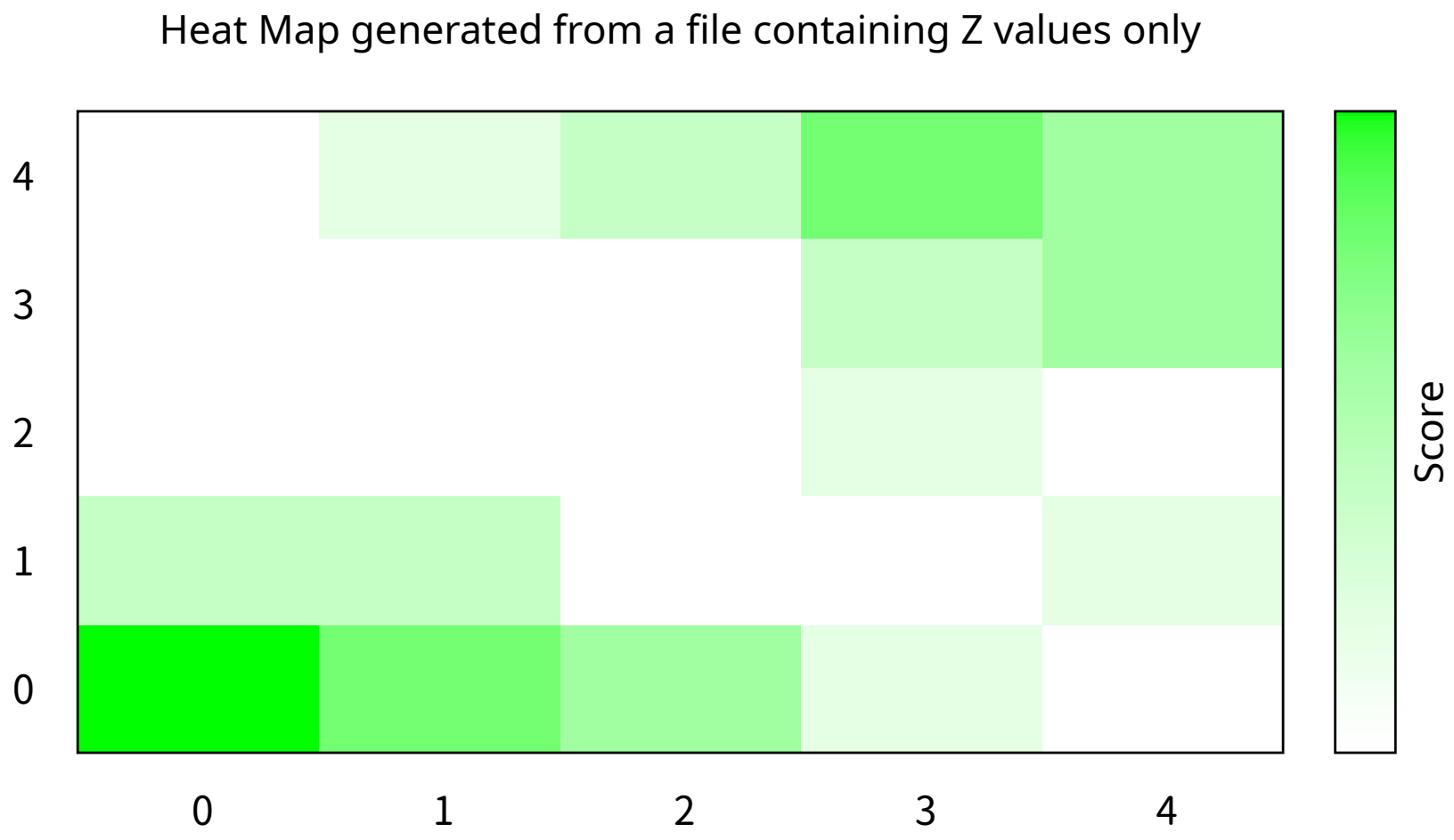

real-valued range of Lambert W function for branches k=0 k=-1
[x = -1./e : 4.] LambertW(x, 0)
[x = -1./e : 0] LambertW(x,-1)
2
1
0
-1
-2
-3
-4
-1/e
0
0.5
1
1.5
2
2.5
3


real(LambertW(x + iy, 0))
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1

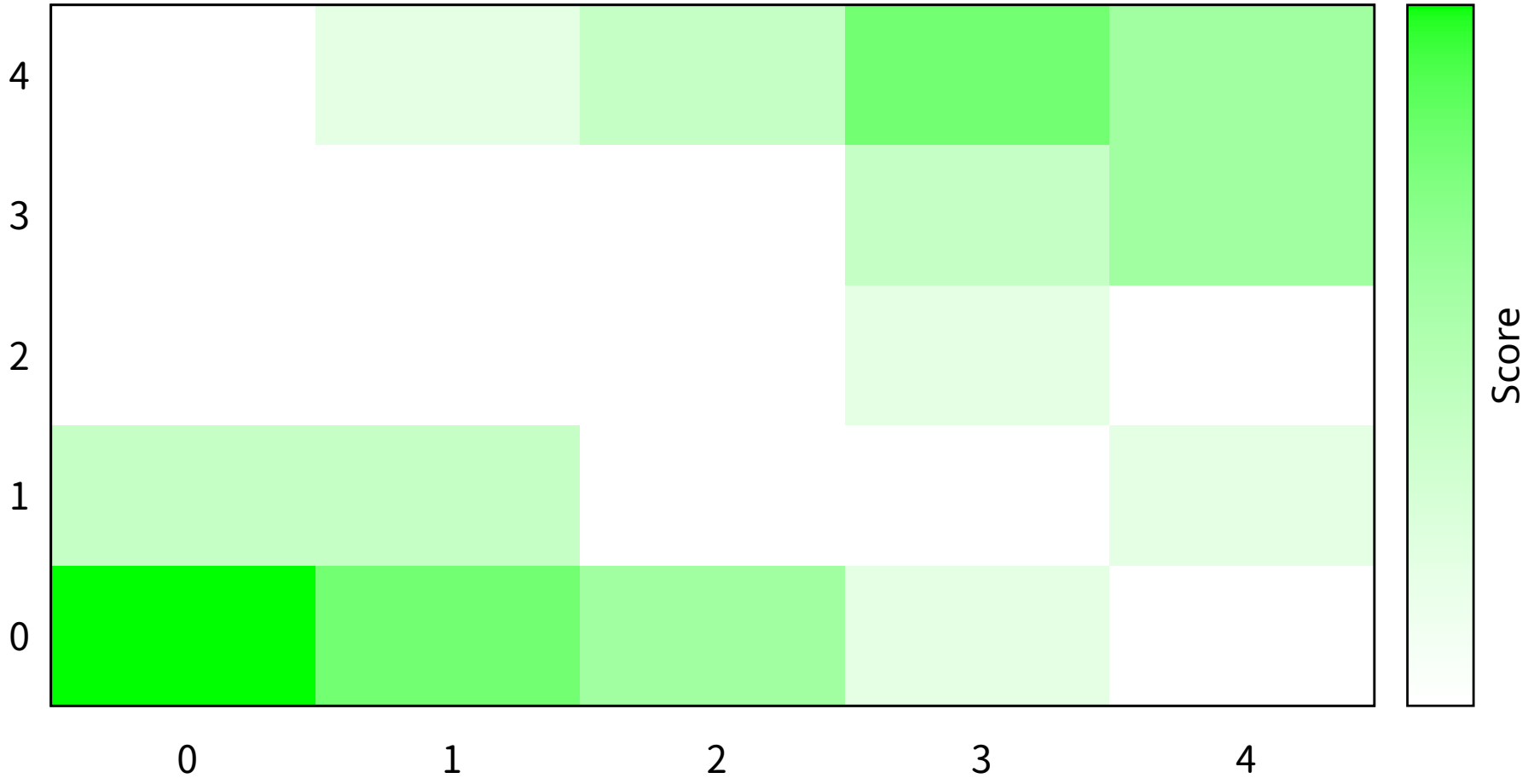

imag(LambertW(x + iy, 0))
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


LambertW(x+iy, k)
k = 4
k = 3
k = 2
k = 1
k = 0
25
20
15
10
5
0
10
5
0
-5
-10
-5
0
5
10

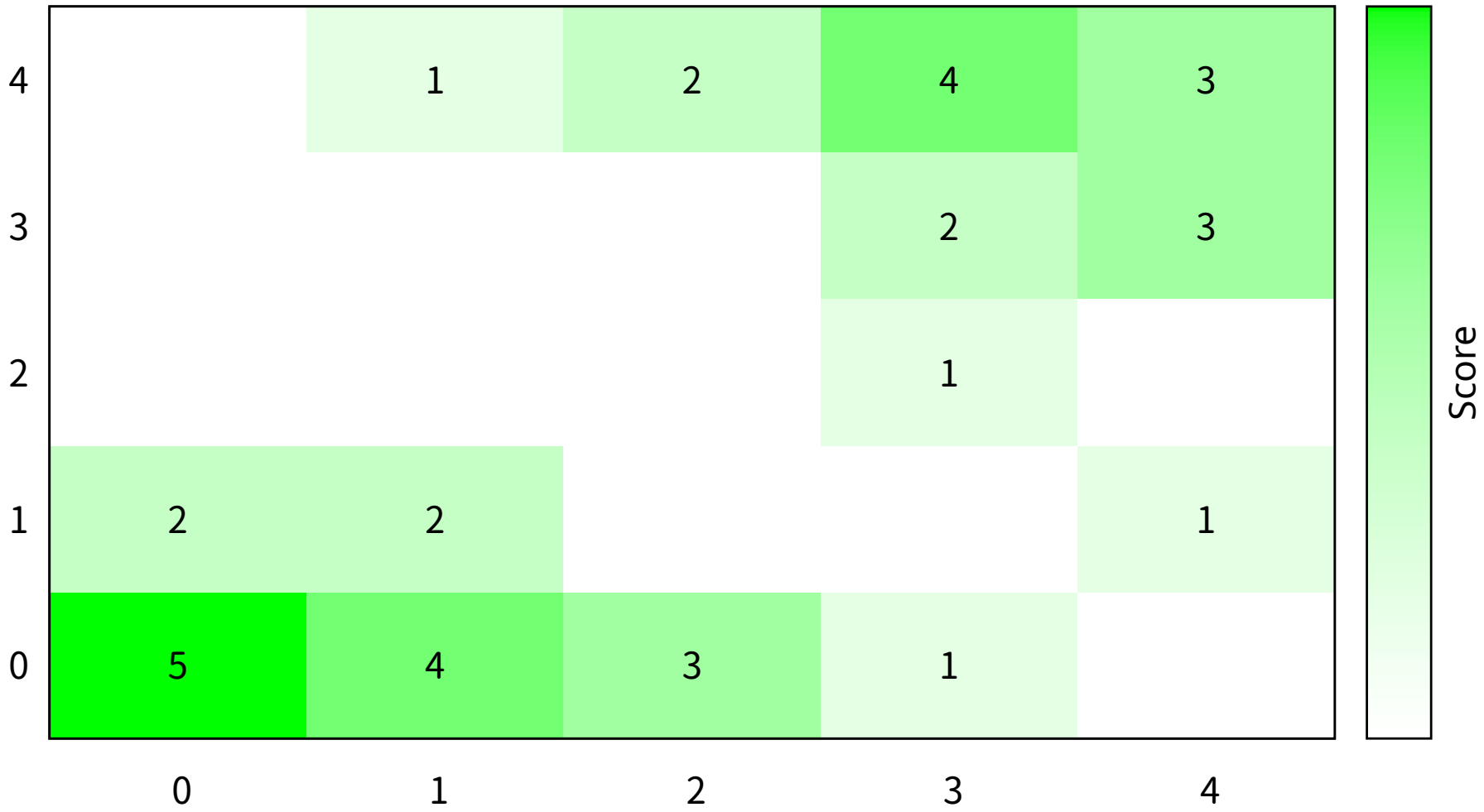

Synchrotron function F(x)
1
0.1
0.01
0.01v_c
0.1v_c
v_c
10v_c
F(x) = $x \int_x^\infty K_{5/3}(\nu)\, d\nu$
Power per unit log(ν)


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

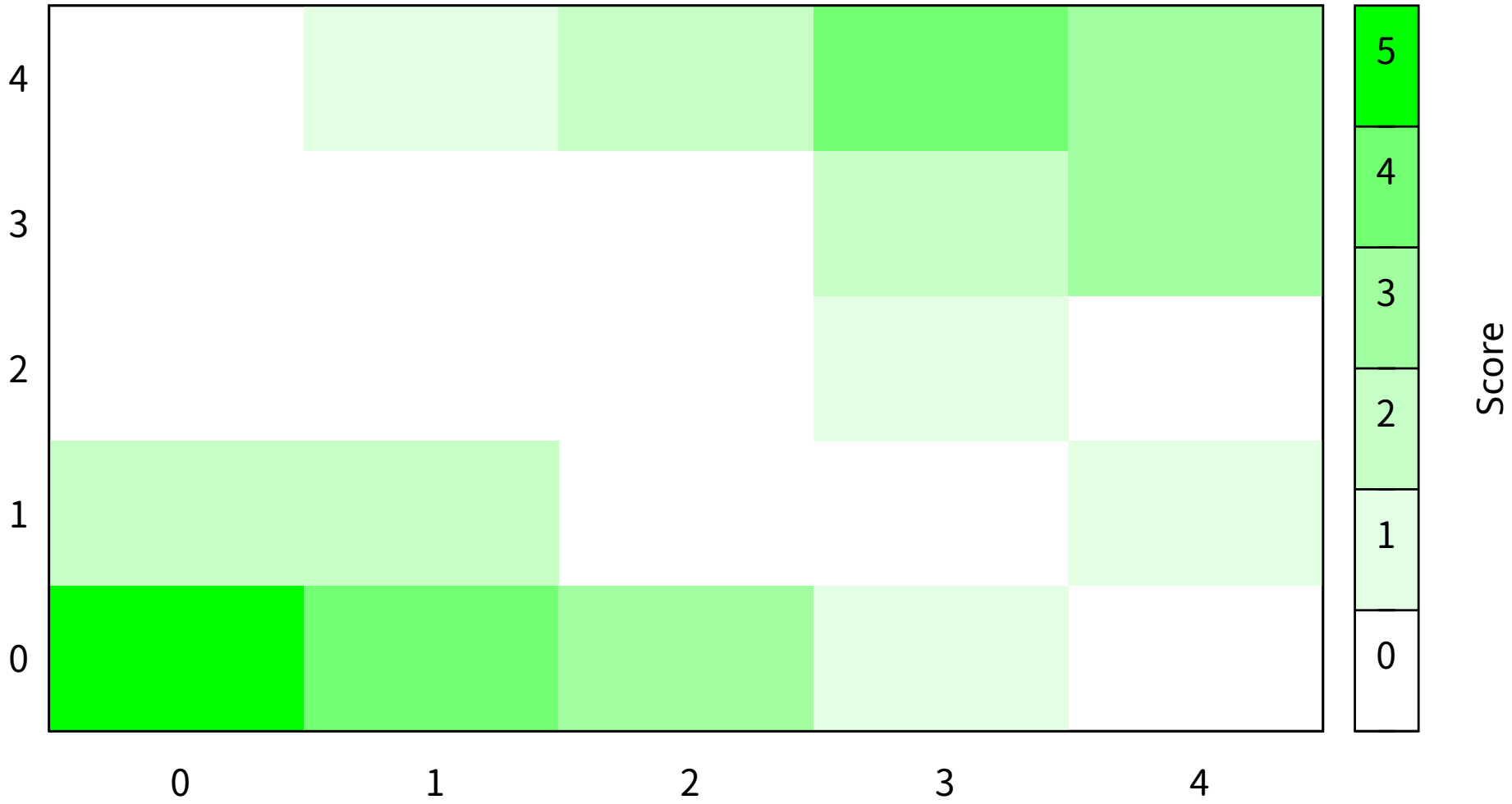
Heat Map generated by 'plot' from a stream of XYZ values
NB: Rows must be separated by blank lines!



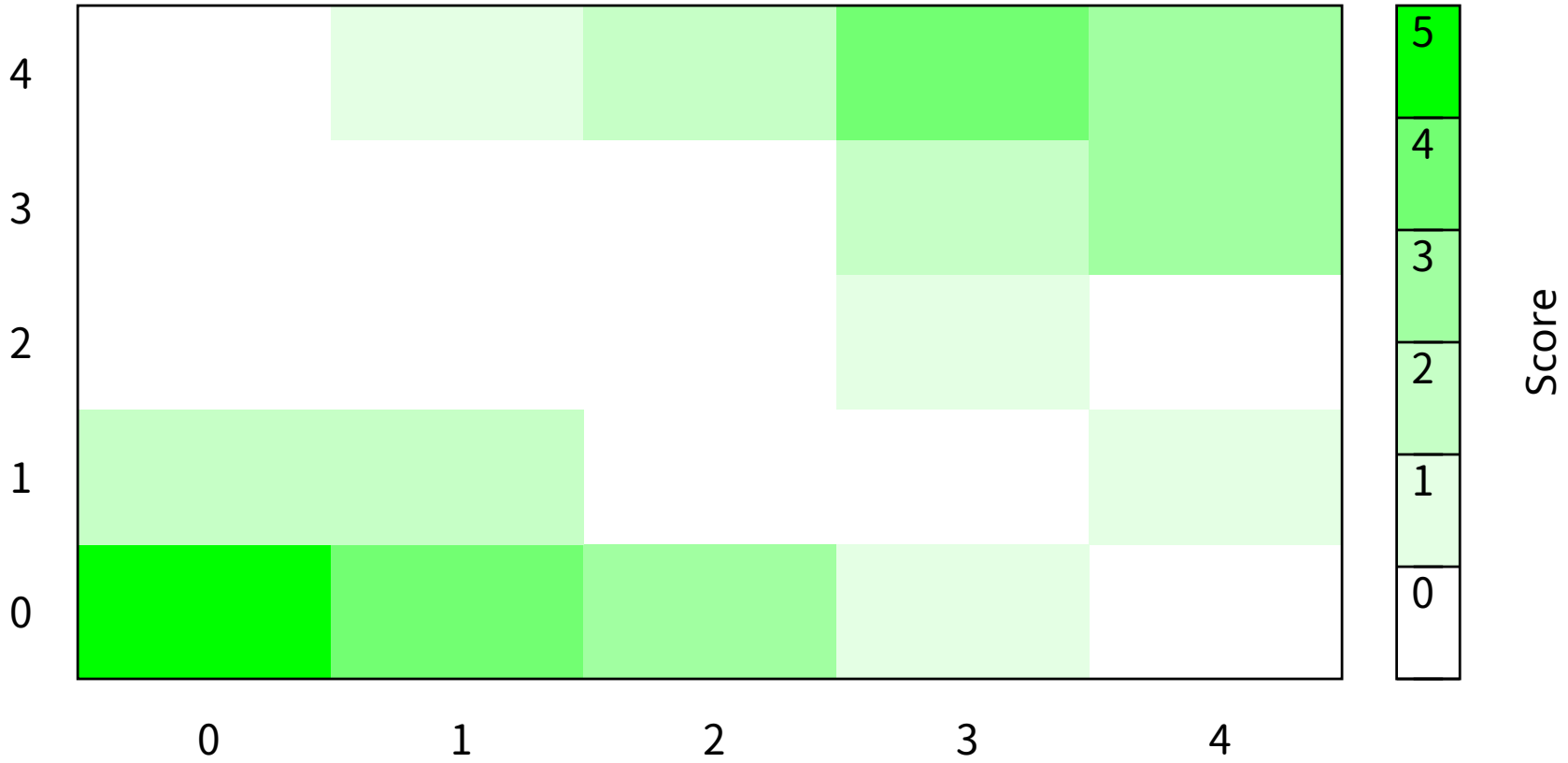
Heat map with non-zero pixel values written as labels

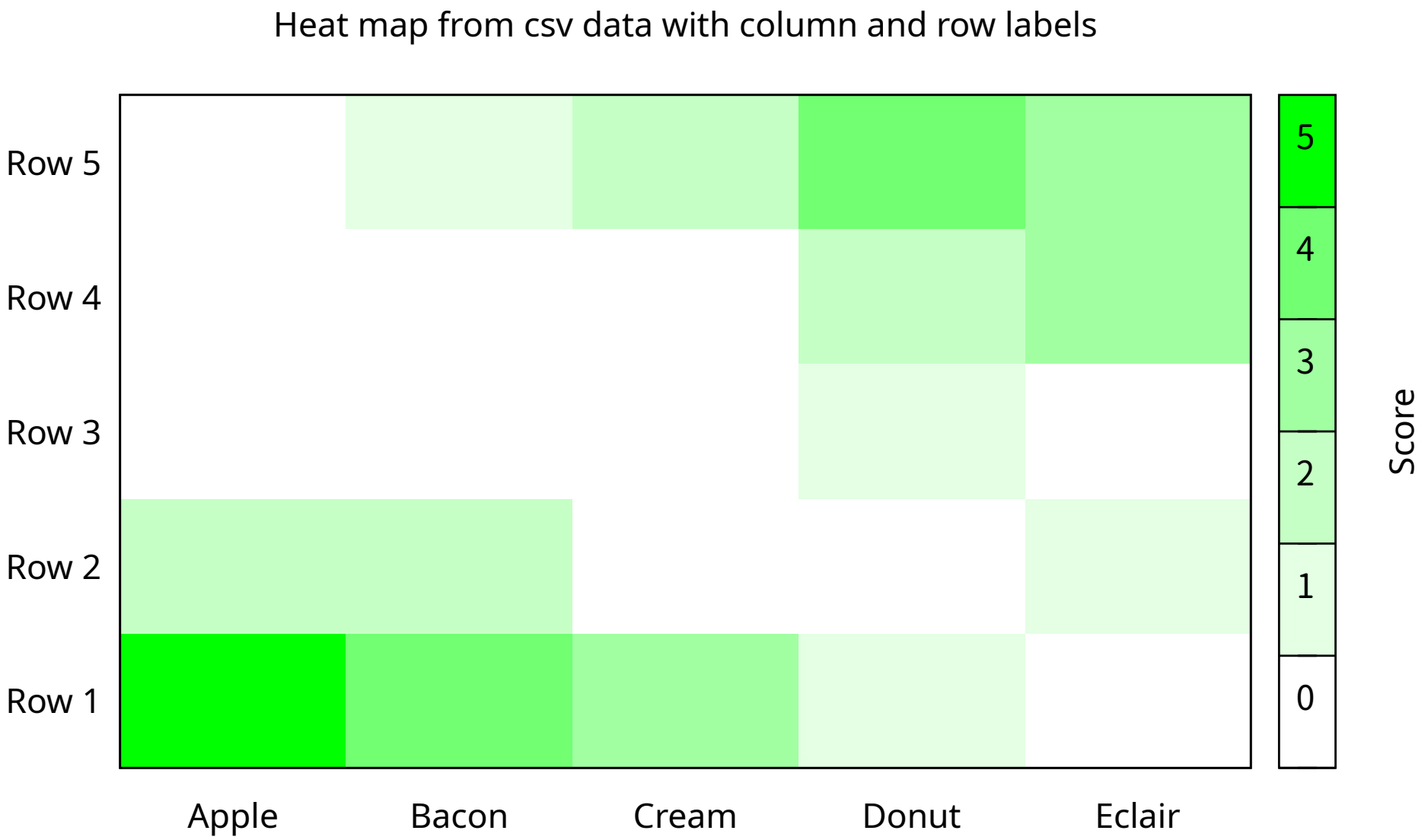


Same data input as a sparse matrix (non-zero values only)



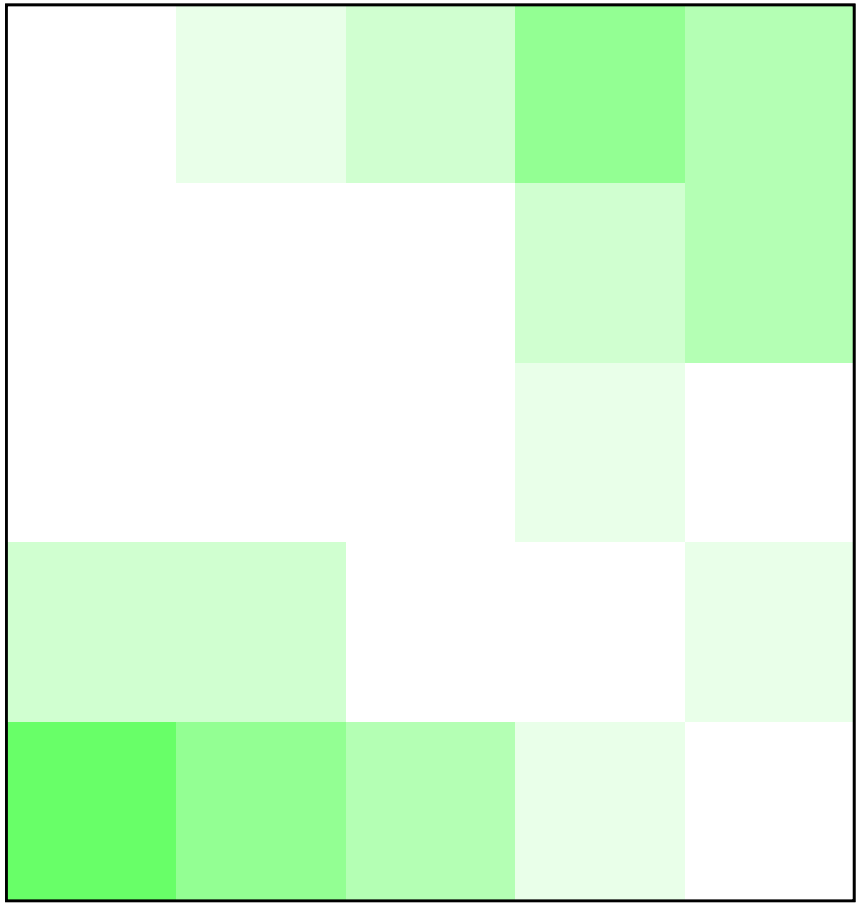
Sparse matrix handling is also possible with splot



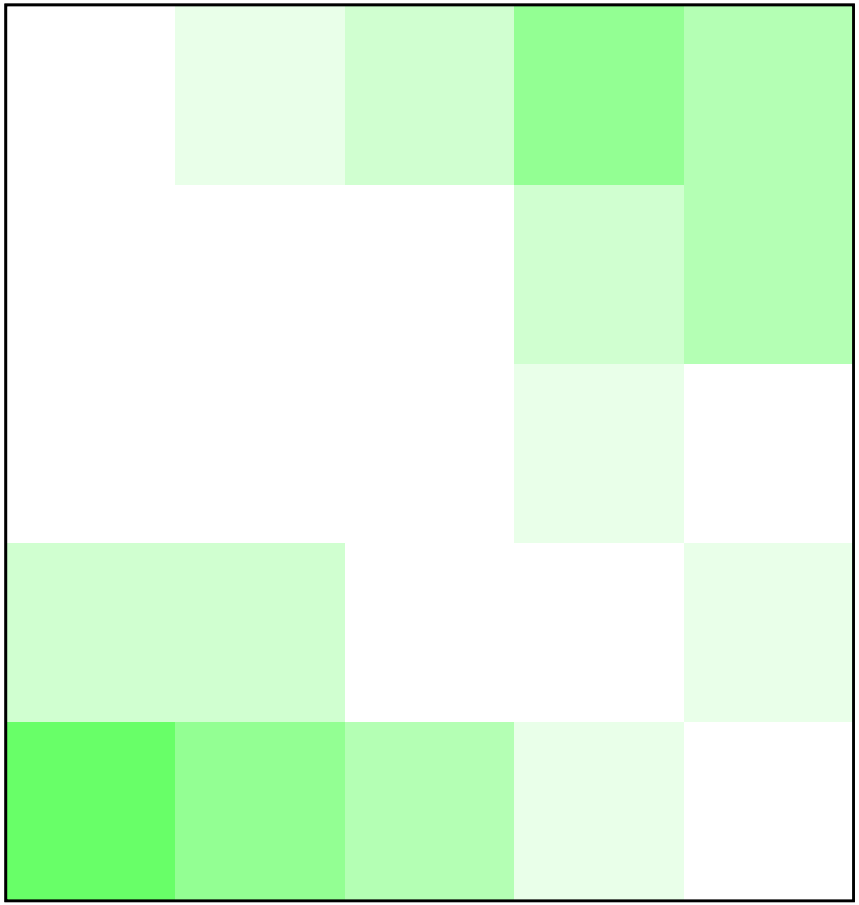

Heat map from csv data with column and row labels
Row 5
Row 4
Row 3
Row 2
Row 1
Apple
Bacon
Cream
Donut
Eclair
5
4
3
2
1
0
Score

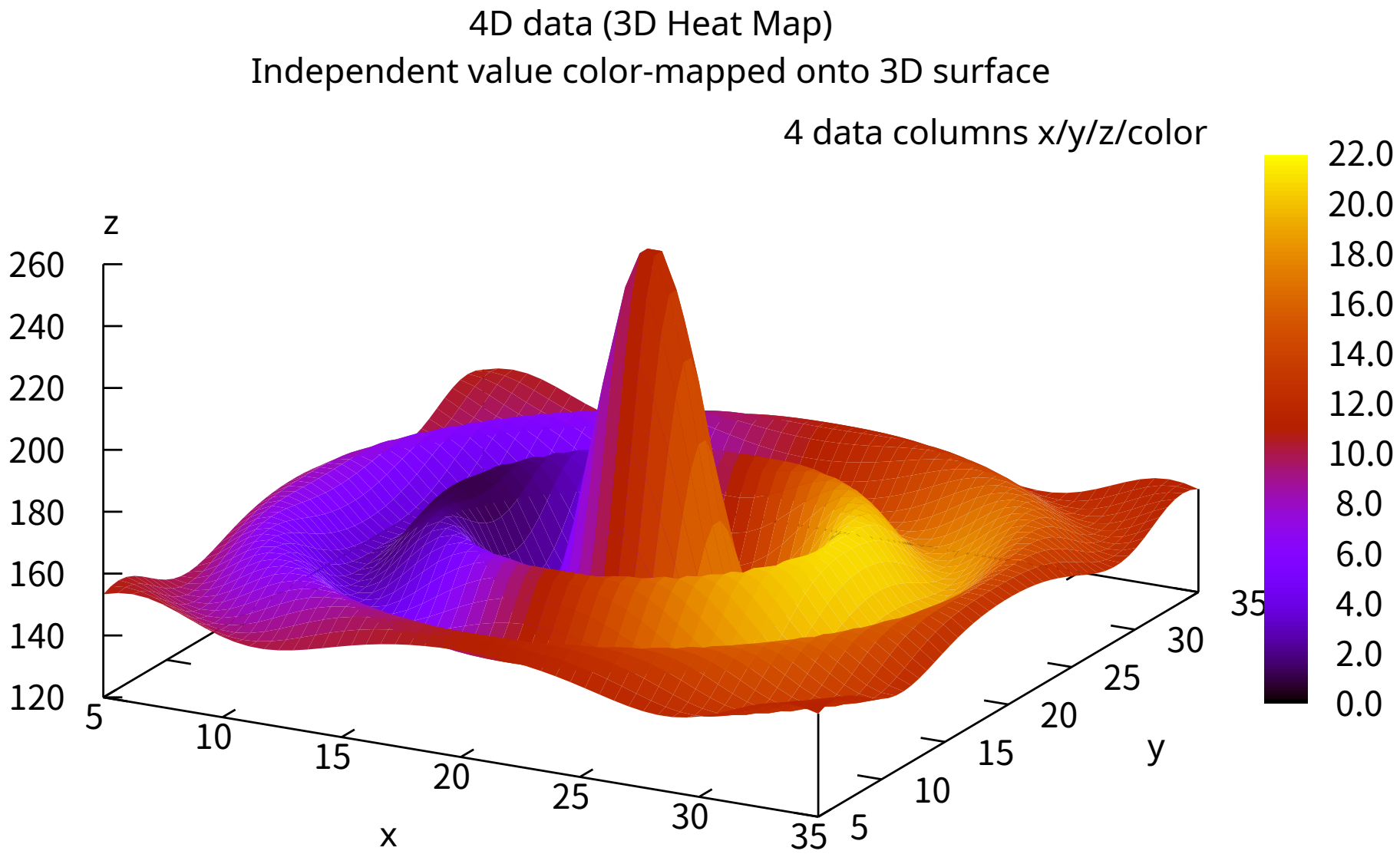
Compare 'image' and 'image pixels' modes

plot with image



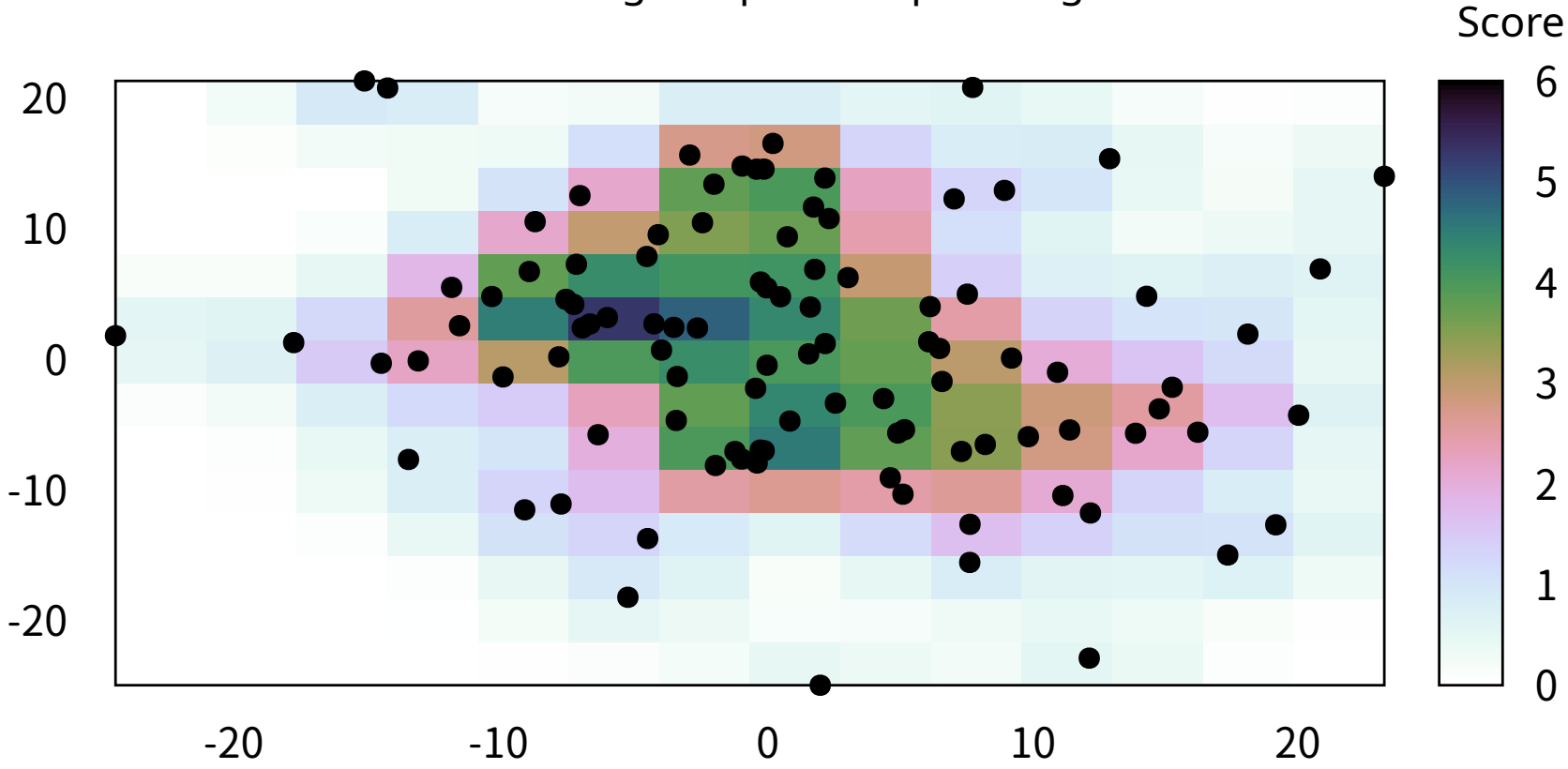
plot with image pixels




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
5
10
15
20
25
30
35
x
5
10
15
20
25
30
35
y
22.0
20.0
18.0
16.0
14.0
12.0
10.0
8.0
6.0
4.0
2.0
0.0

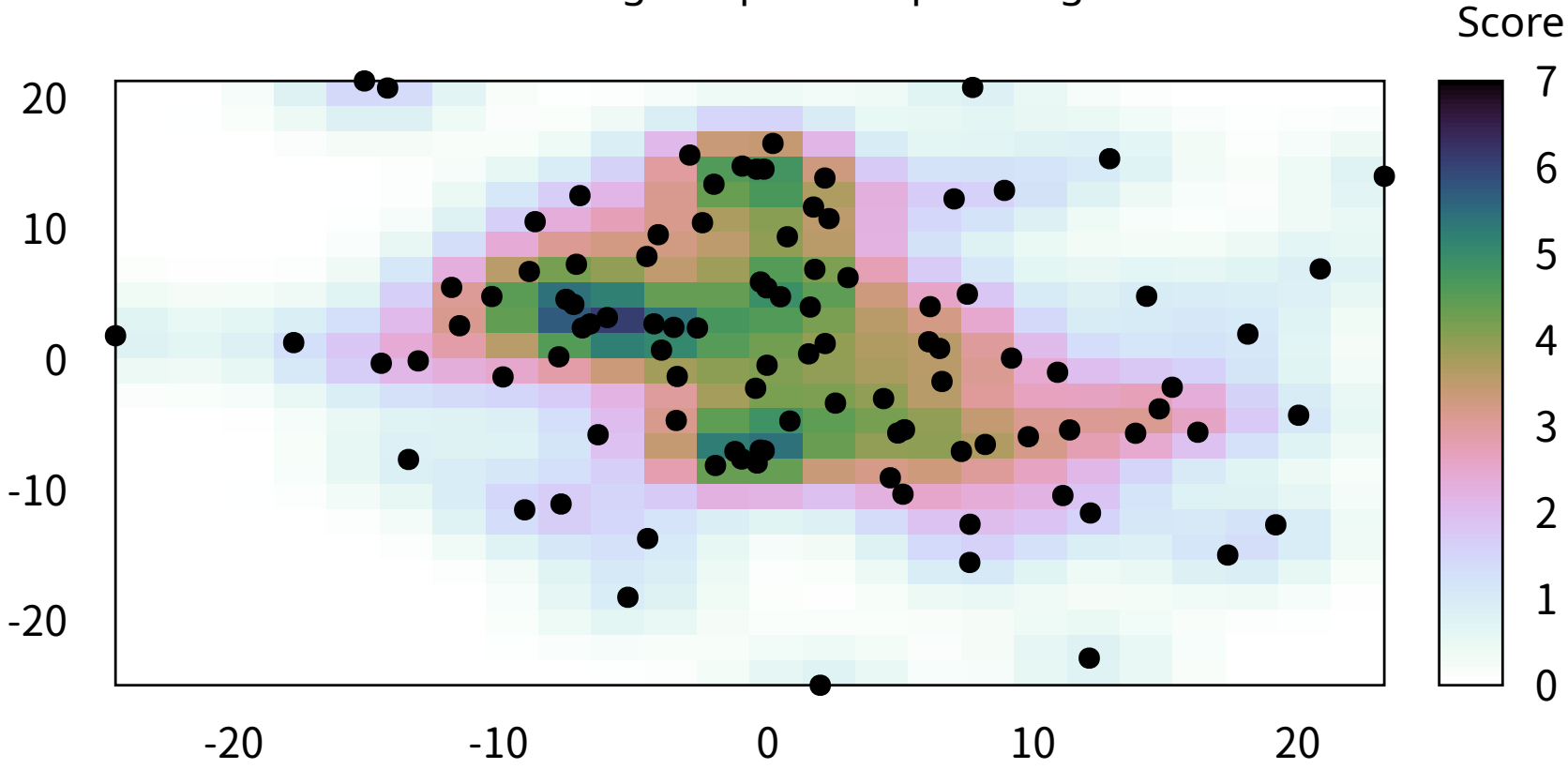
Heat map of point density

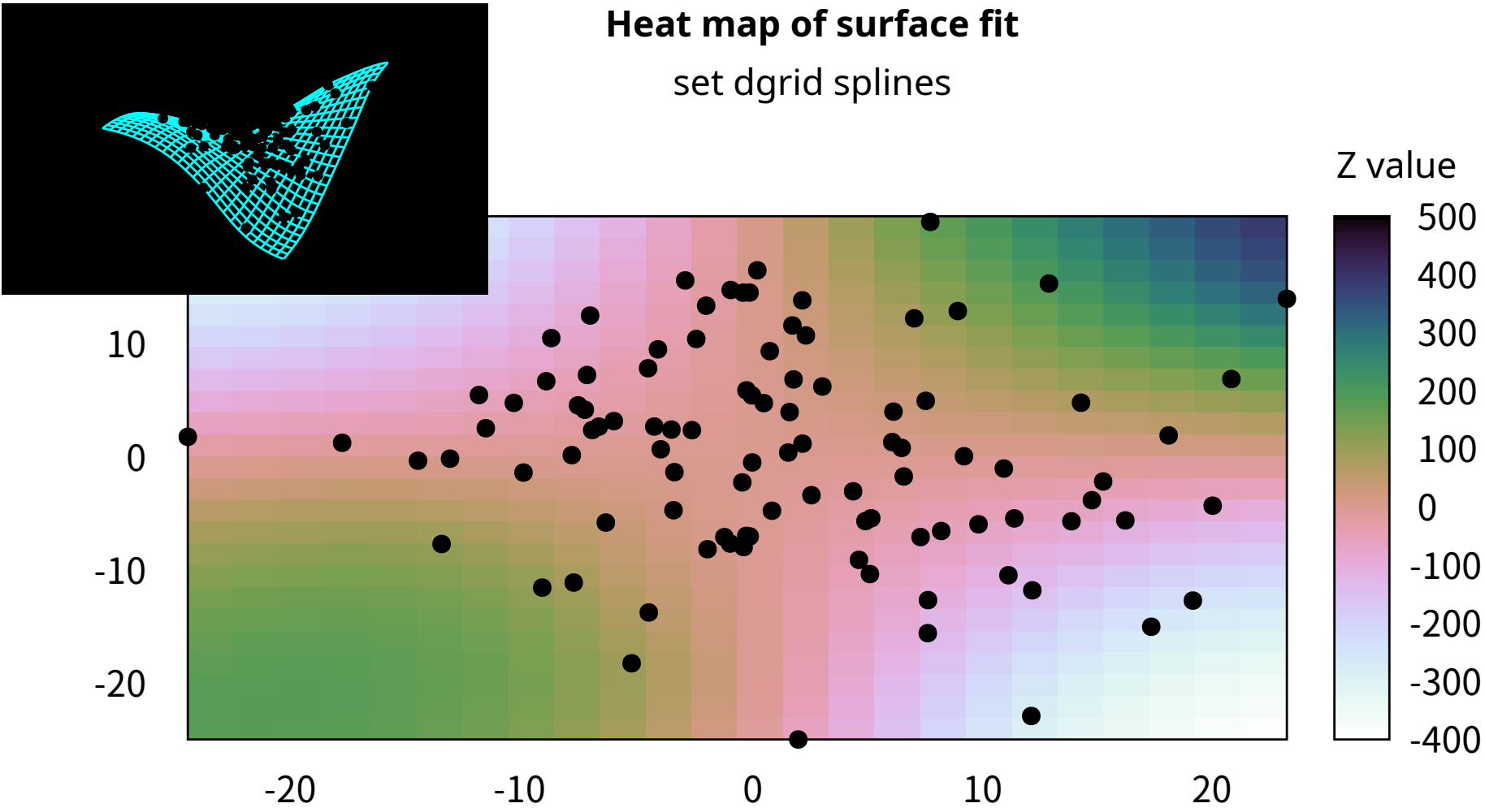
set dgrid 15,15 gauss kdensity 3, 3
assign all points equal weight



Heat map of point density

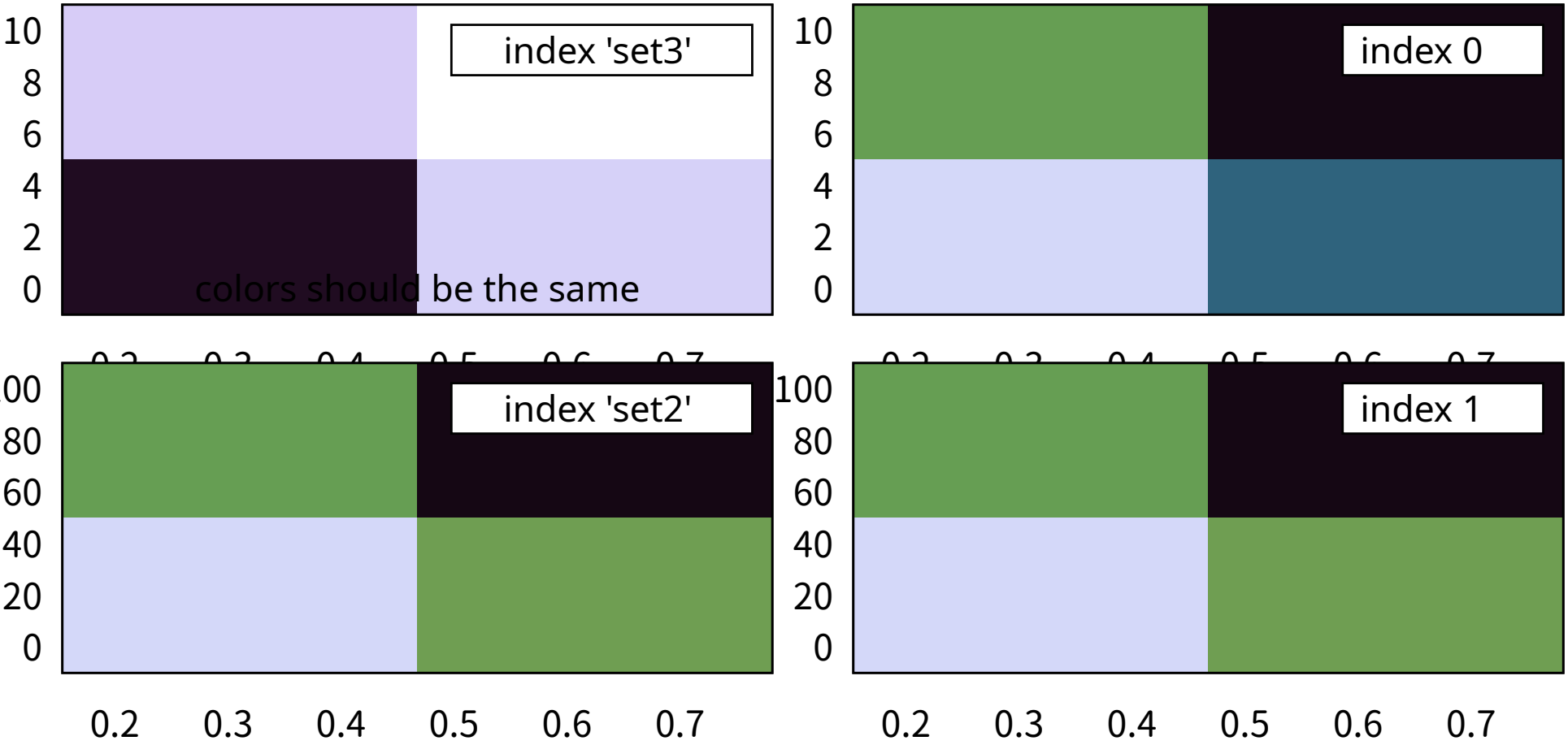
set dgrid 25,25 gauss kdensity 3, 3
assign all points equal weight

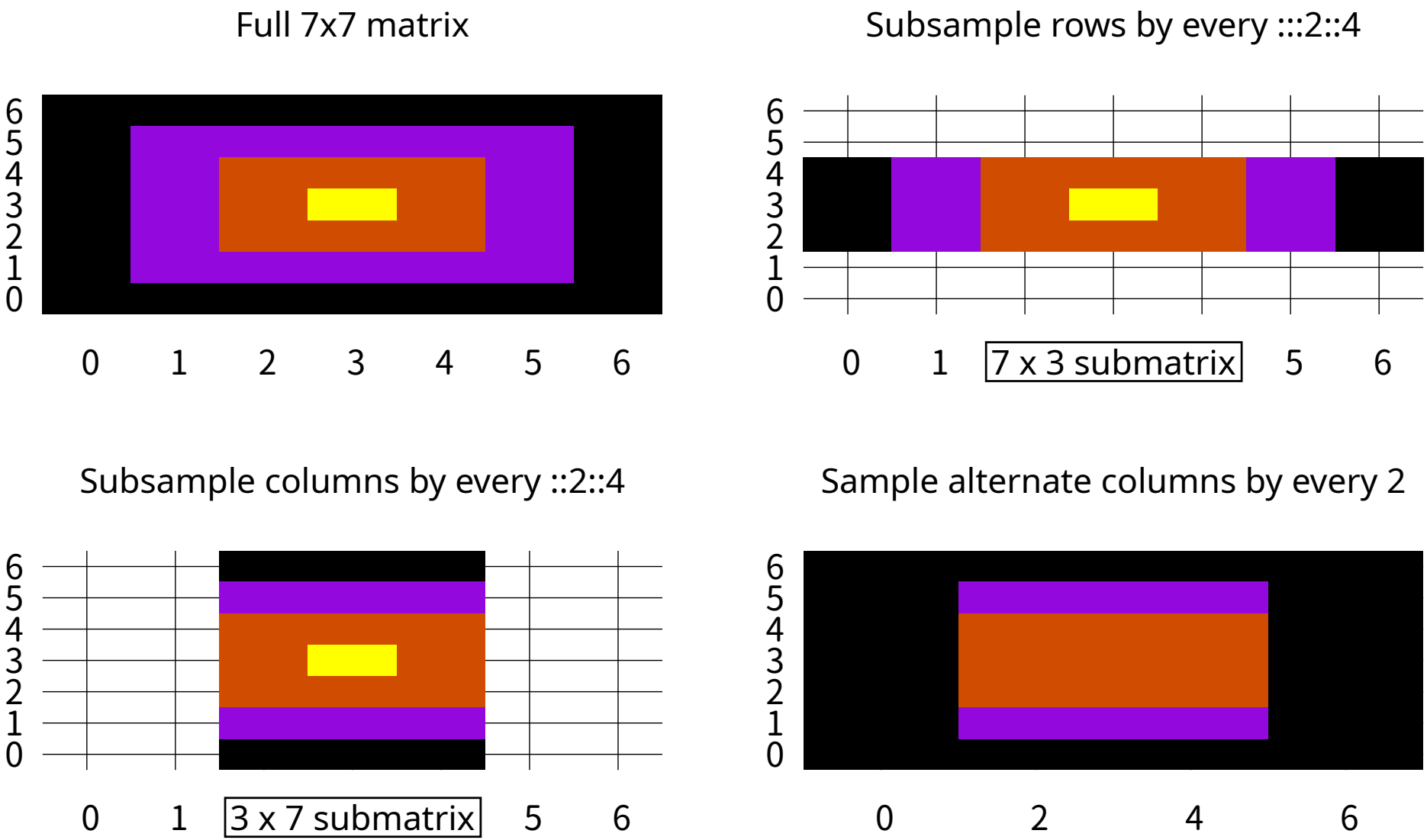



Heat map of surface fit
set dgrid splines
Z value
500
400
300
200
100
0
-100
-200
-300
-400
10
0
-10
-20
-20
-10
0
10
20

Data file contains labeled ascii matrices

Y range should be the same




Full 7x7 matrix
Subsample rows by every :::2::4
7 x 3 submatrix
Subsample columns by every ::2::4
3 x 7 submatrix
Sample alternate columns by every 2

gamma = 0.75

x

gamma = 1.0

x

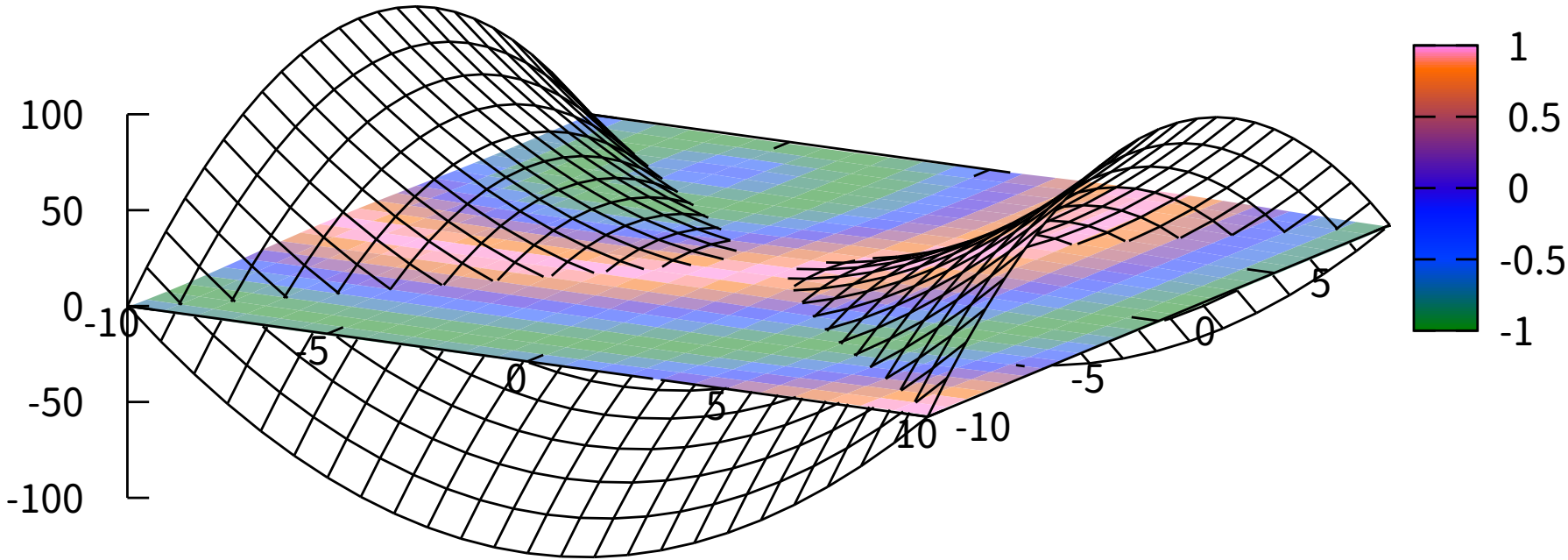
gamma = 1.5 (default)

x

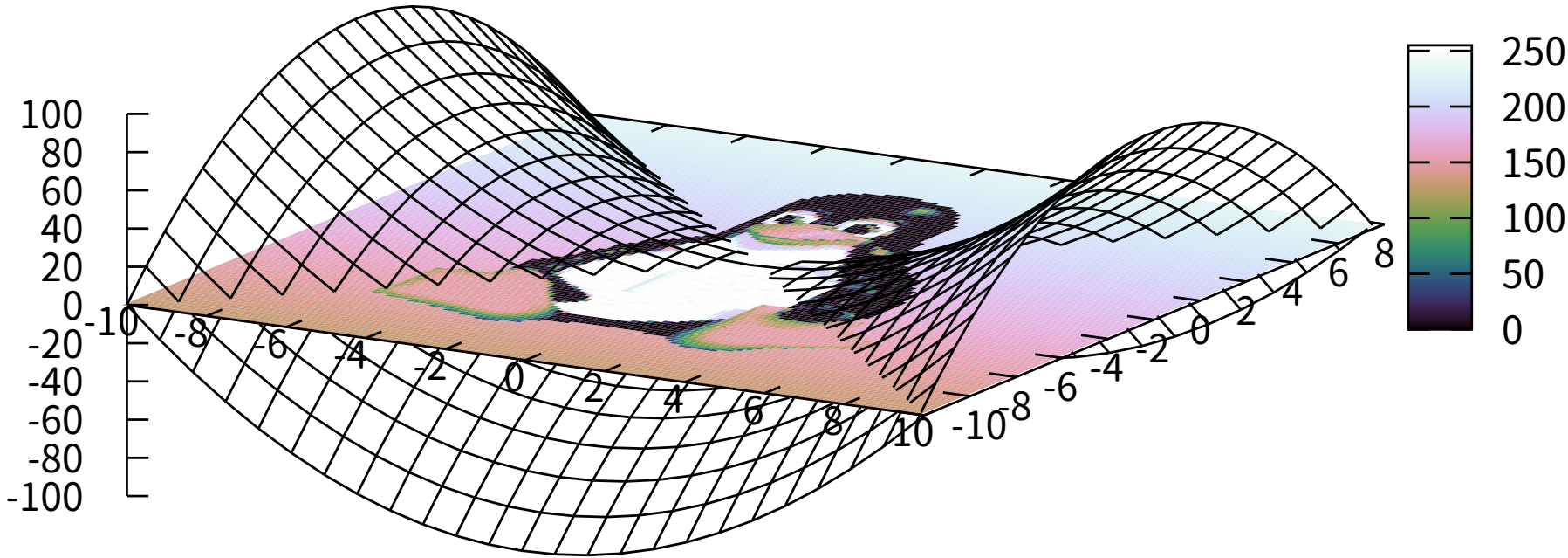
gamma = 2.0

x

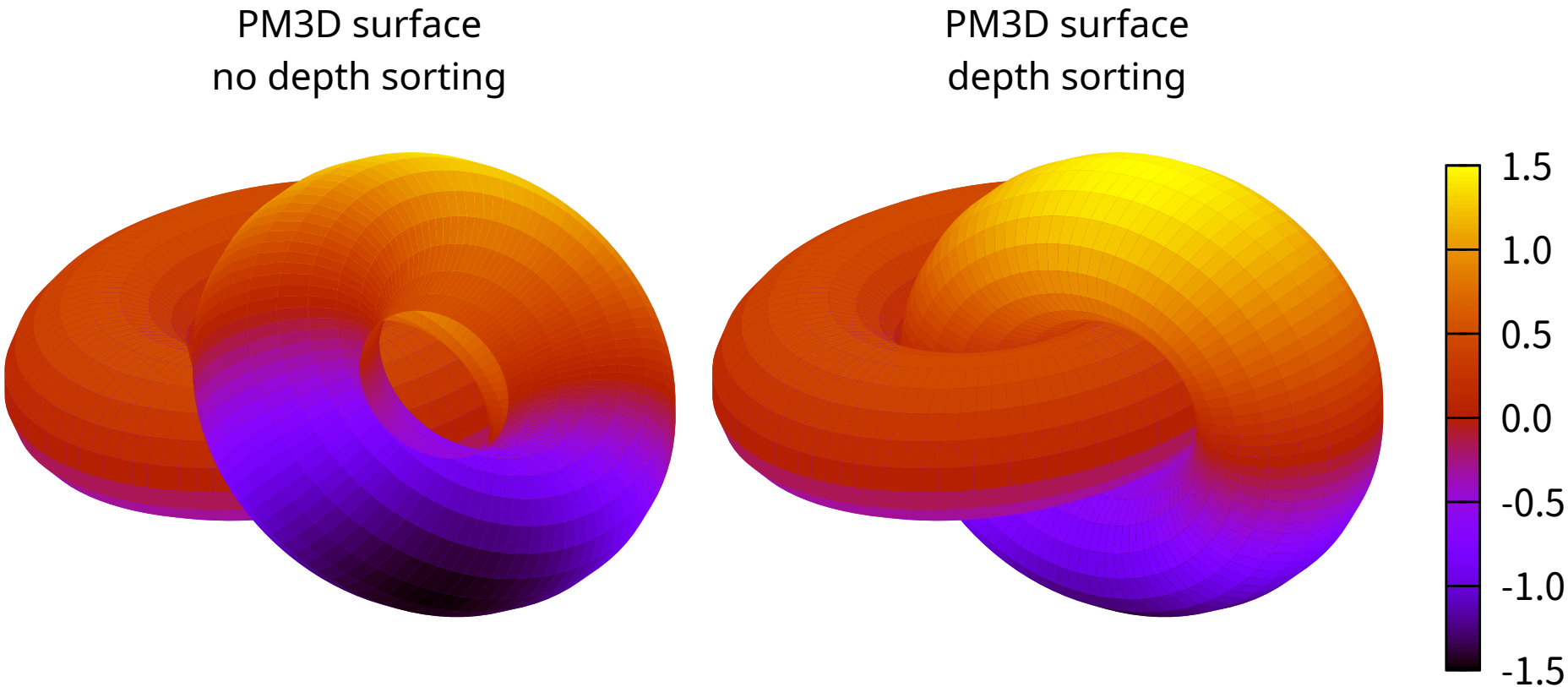
Mixing pm3d surfaces with hidden-line plots

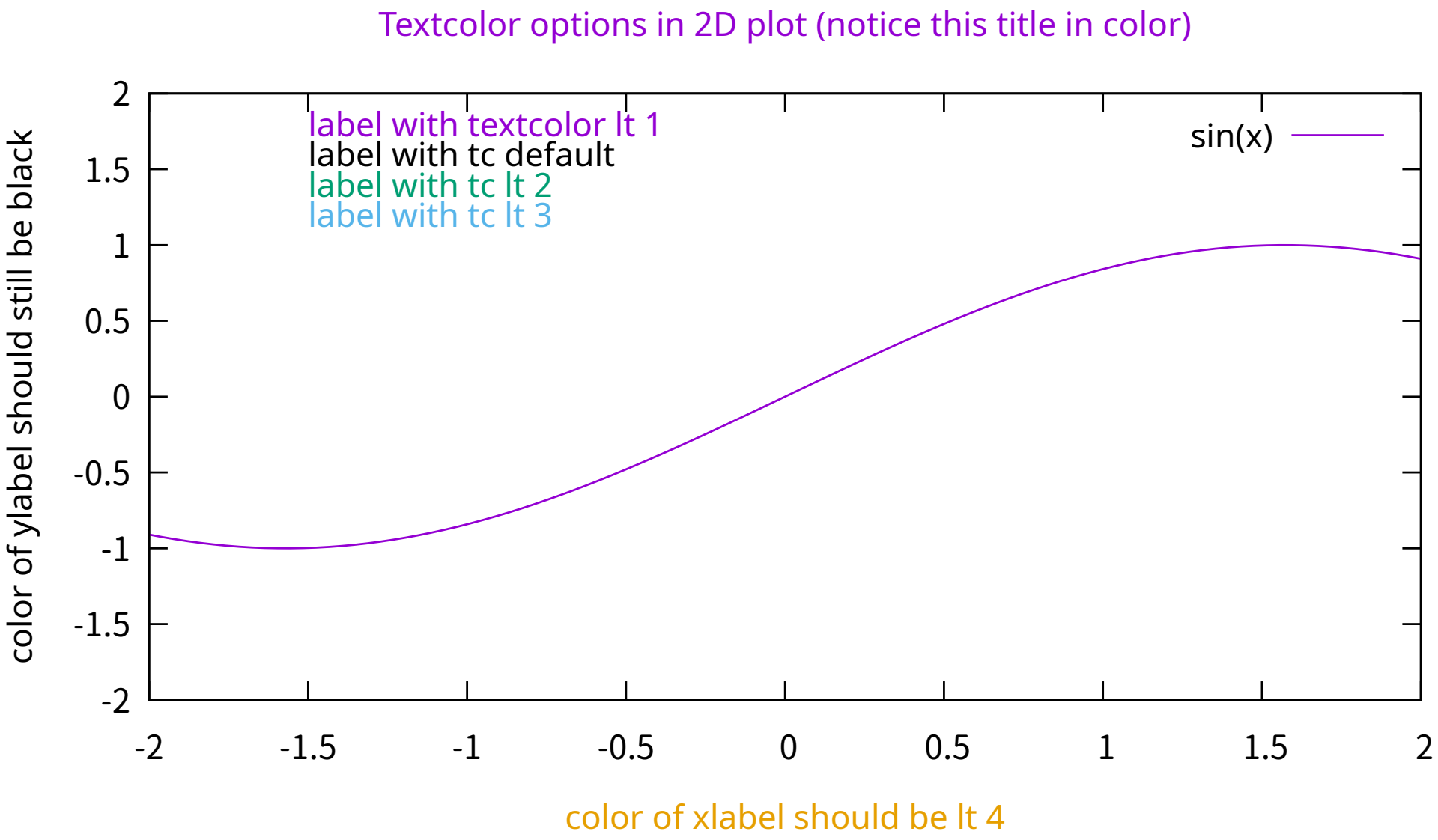


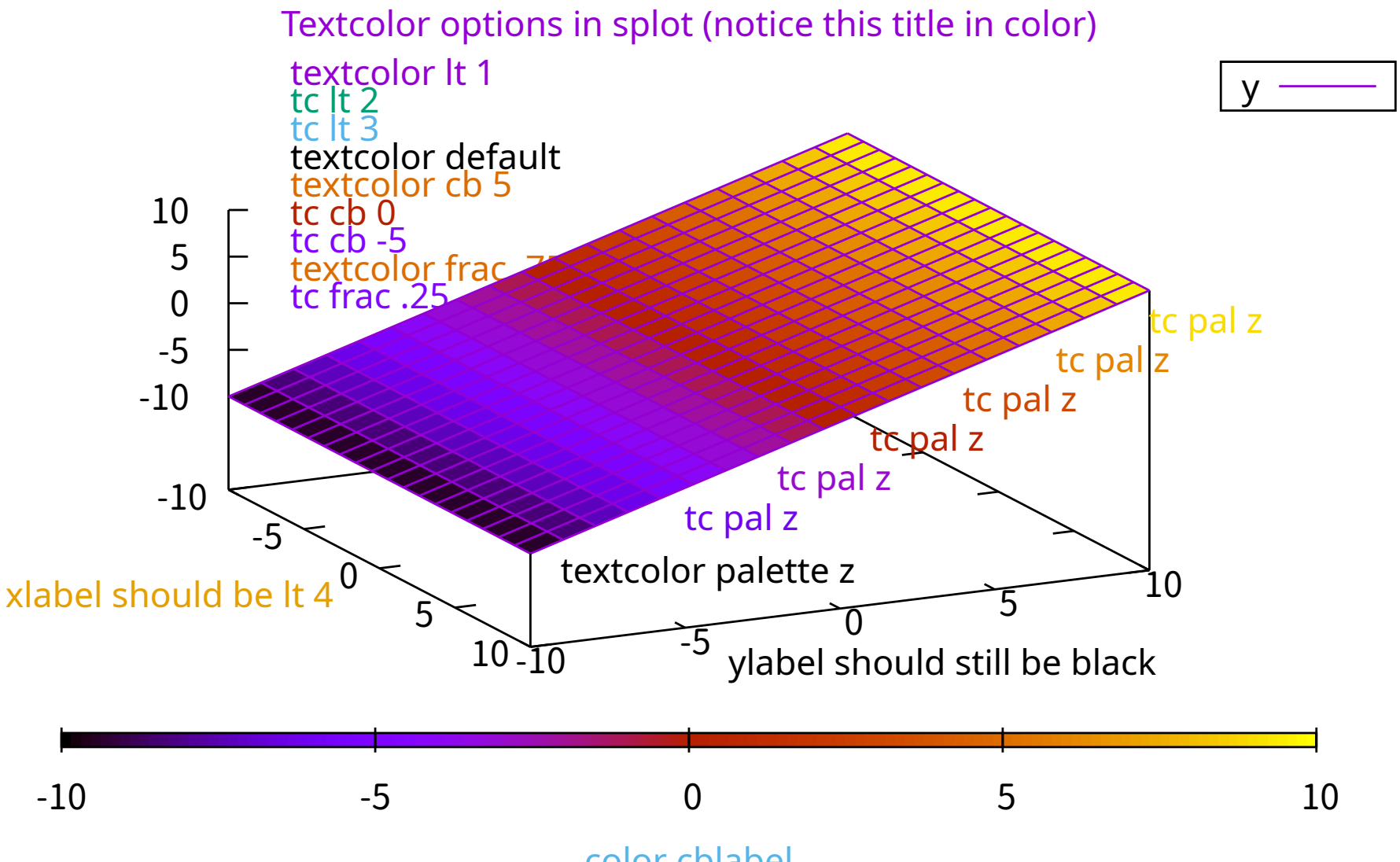
Mixing image surface with hidden-line plots

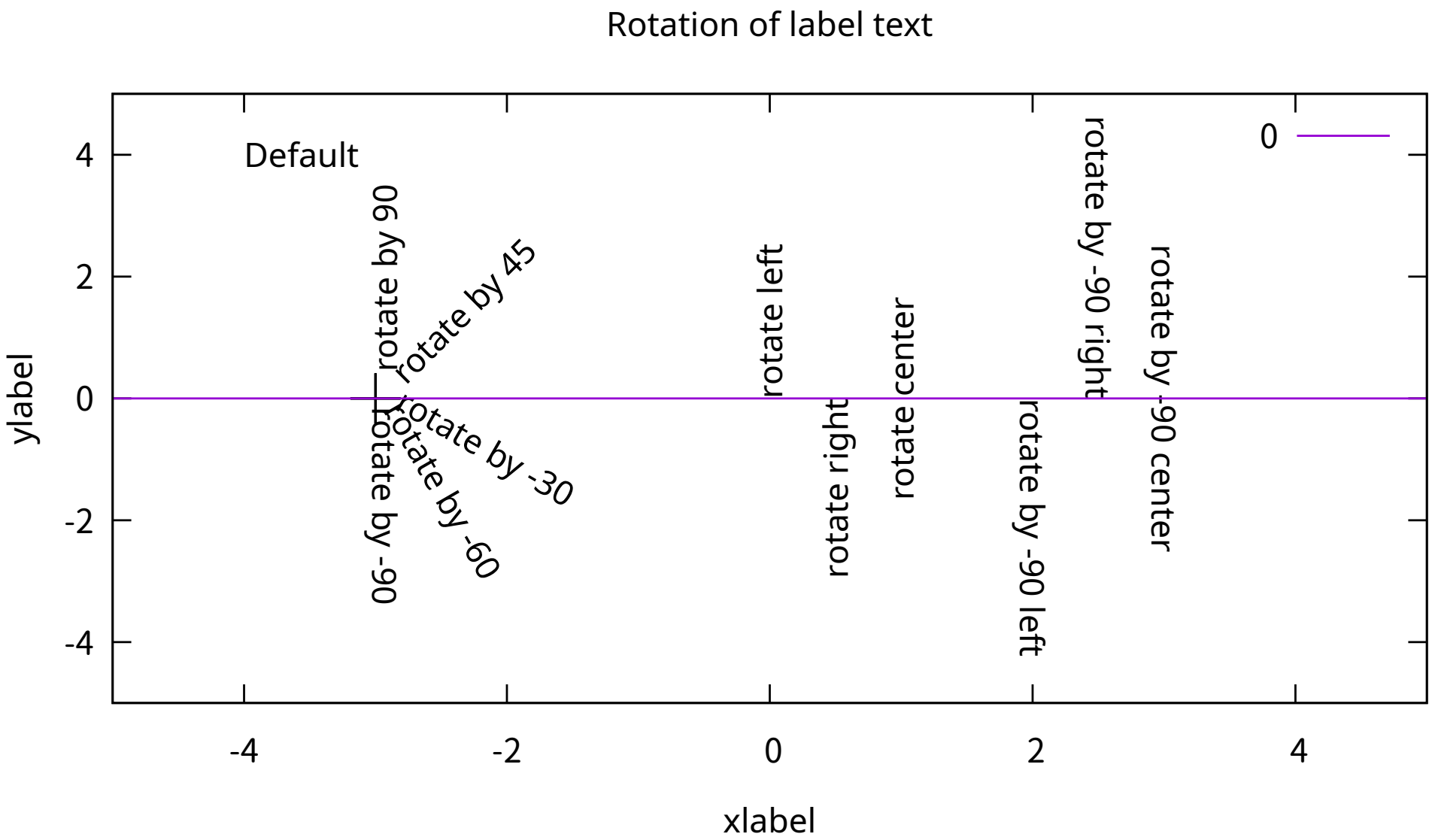


Interlocking Tori

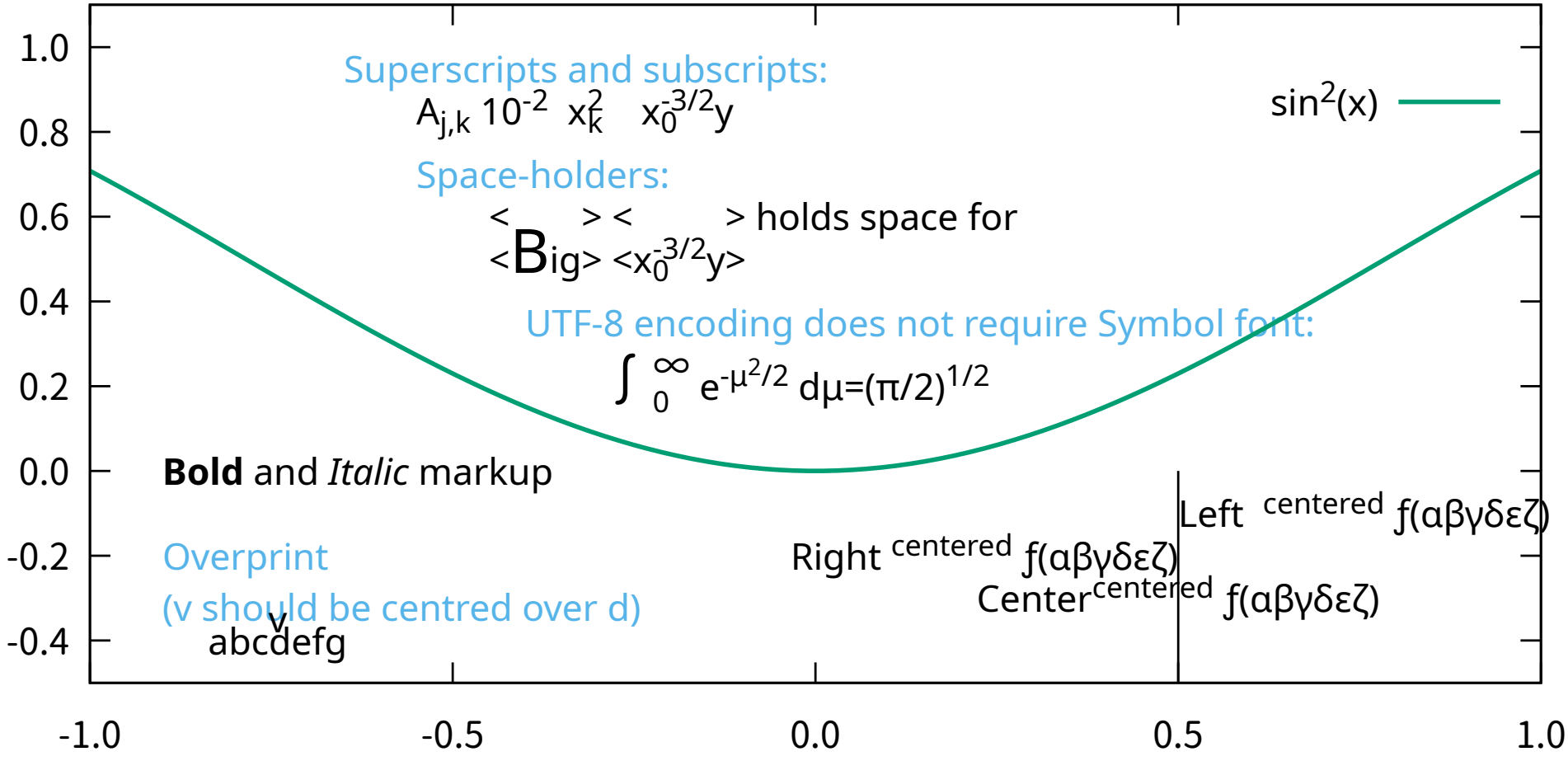



Textcolor options in 2D plot (notice this title in color)
label with textcolor lt 1
label with tc default
label with tc lt 2
label with tc lt 3
sin(x)
color of ylabel should still be black
color of xlabel should be lt 4
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2


Textcolor options in splot (notice this title in color)
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
y
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
-10
-5
0
5
10
color cblabel

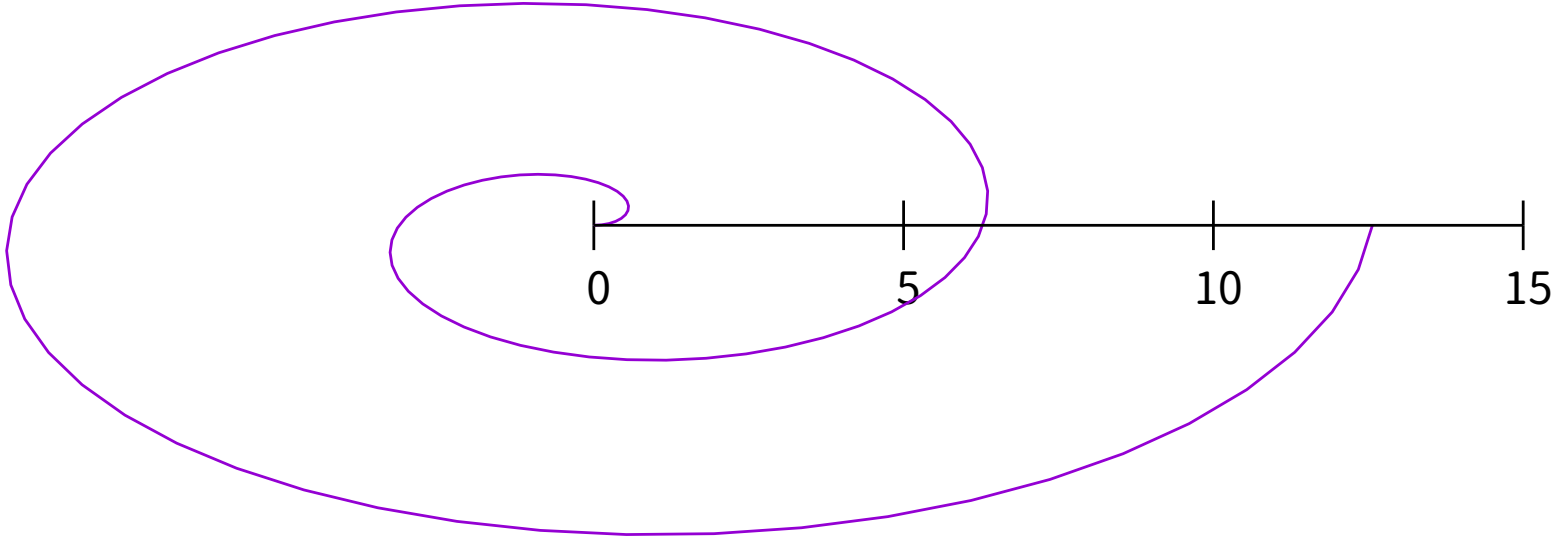

Rotation of label text
Default
rotate by 90
rotate by 45
rotate left
rotate center
rotate by -90 right
rotate by -90 center
rotate by -90
rotate by -60
rotate by -30
rotate right
rotate by -90 left
0
4
2
0
-2
-4
-4
-2
0
2
4
ylabel
xlabel

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



Enhanced text style markup

Default **Bold** ***Italic*** Default



Default *Italic* ***Bold*** Normal Default

Illustrate use of unicode escape sequences

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

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

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

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

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

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

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

Terminal's native dashtypes

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

dt 1

dt 2

dt 3

dt 4

dt 5

dt 6

dt 7

dt 8

dt 9

dt 10

Custom dashtypes

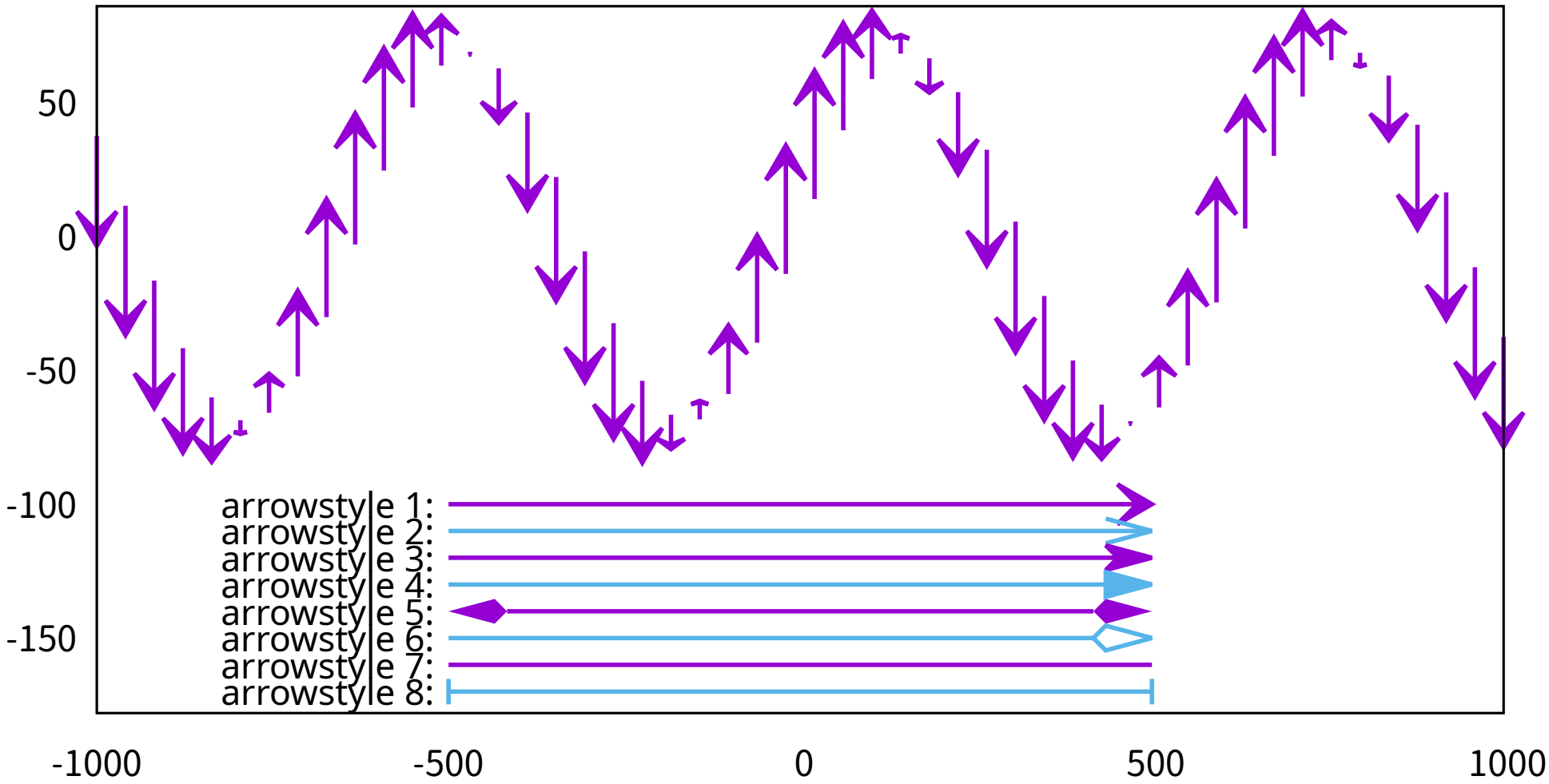
dt "."

dt "-"

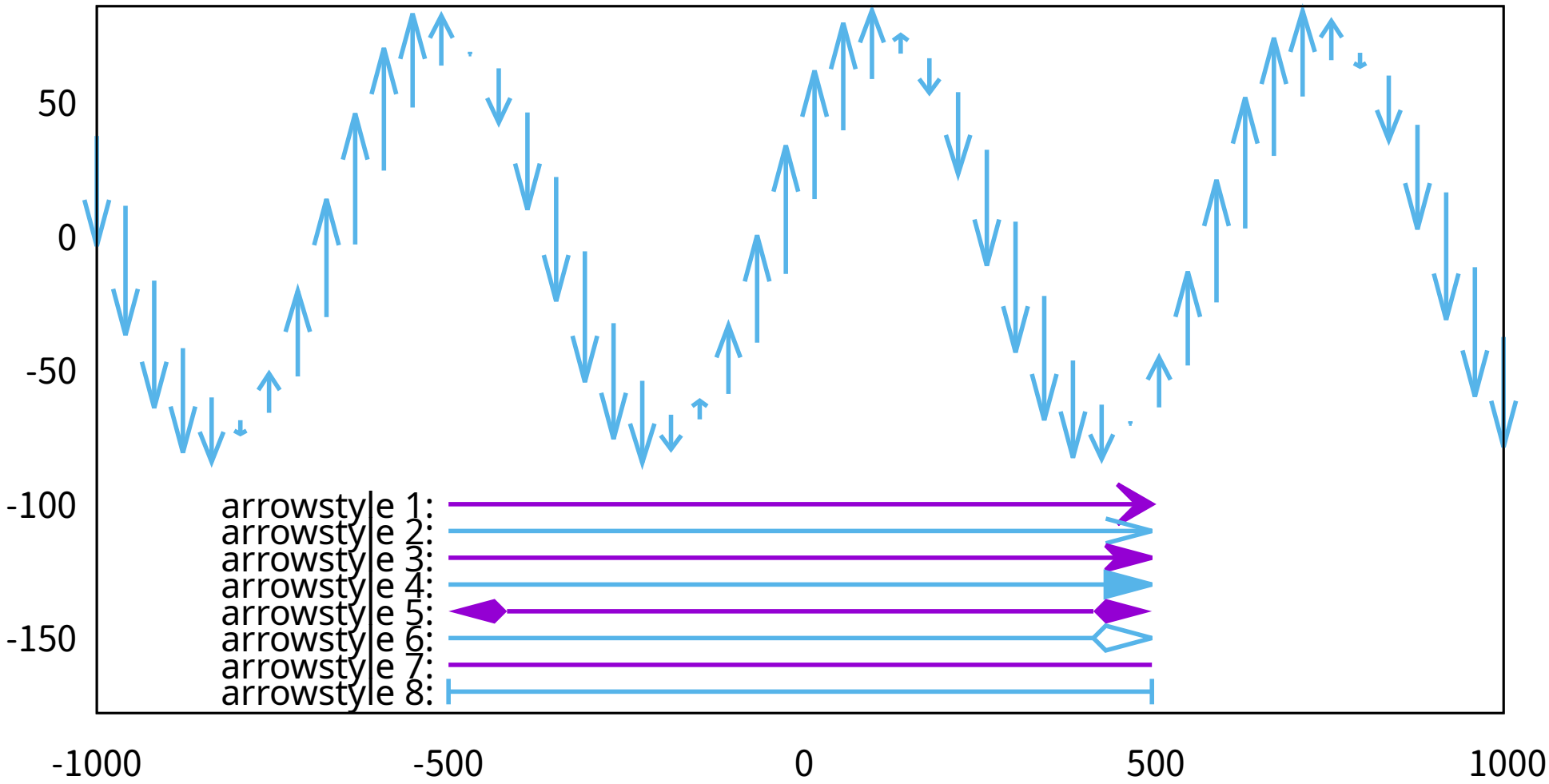
dt "._"

dt "..- "

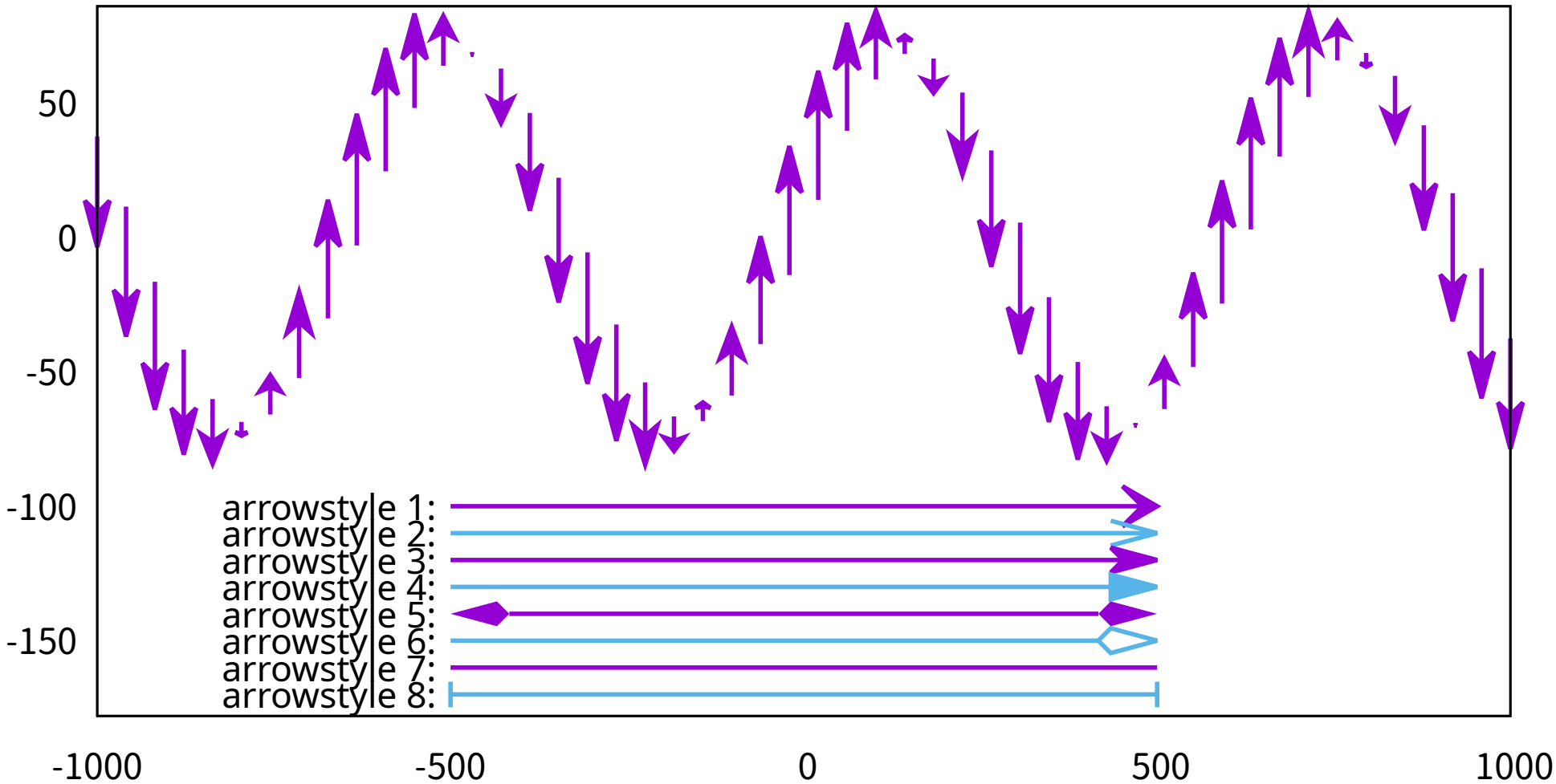
Top: plot with vectors arrowstyle 1, Bottom: explicit arrows



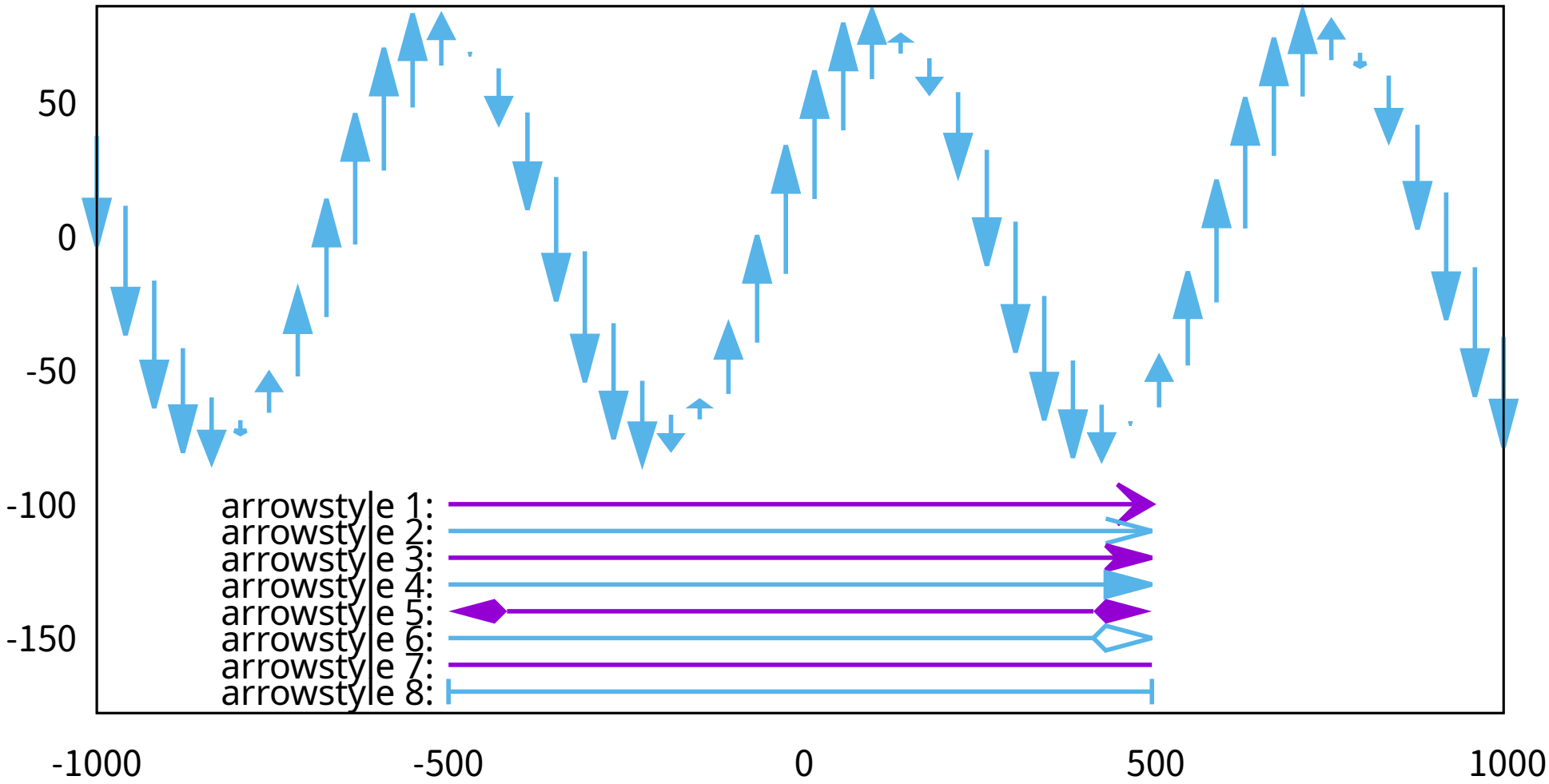
Top: plot with vectors arrowstyle 2, Bottom: explicit arrows



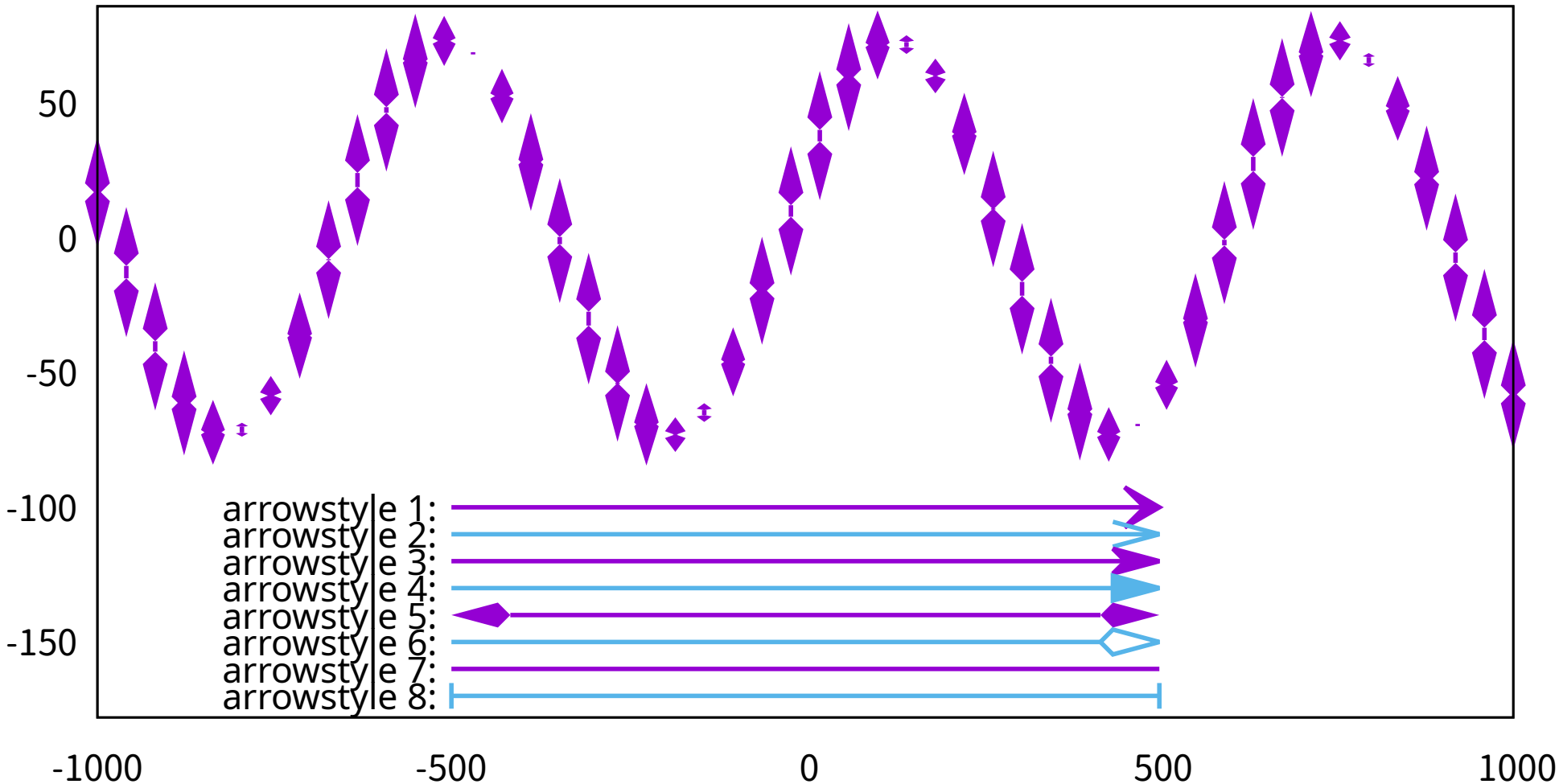
Top: plot with vectors arrowstyle 3, Bottom: explicit arrows



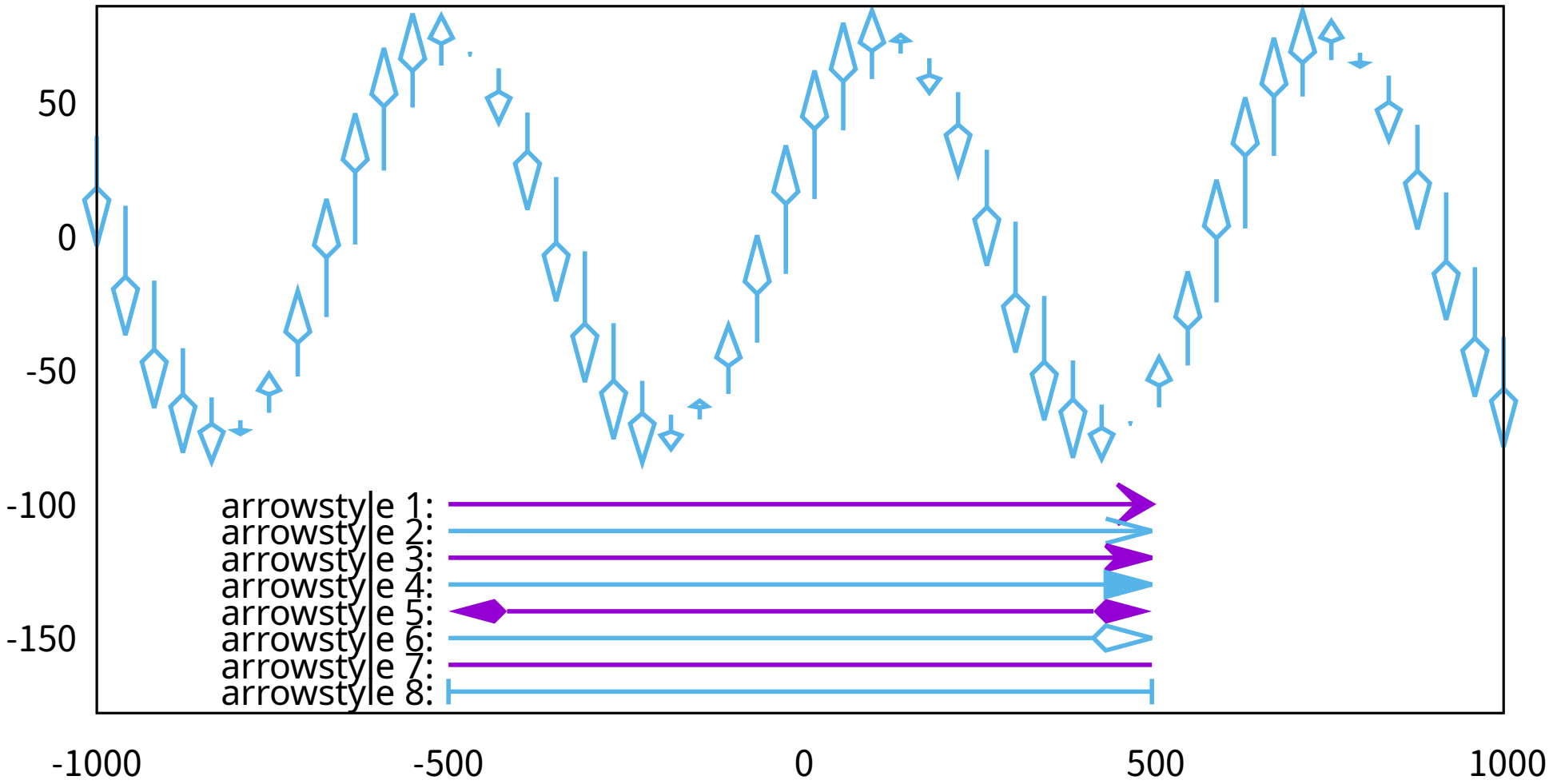
Top: plot with vectors arrowstyle 4, Bottom: explicit arrows



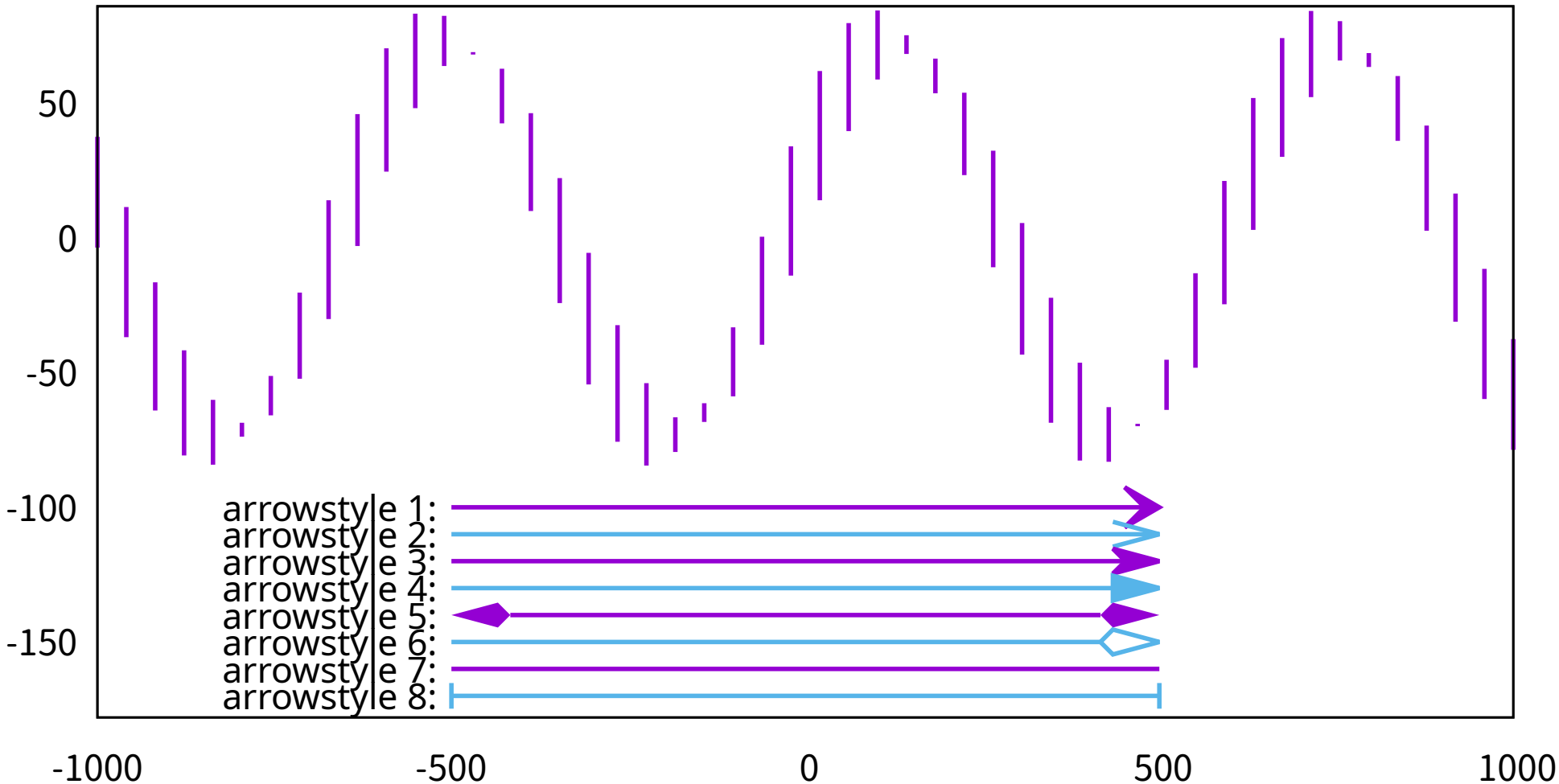
Top: plot with vectors arrowstyle 5, Bottom: explicit arrows



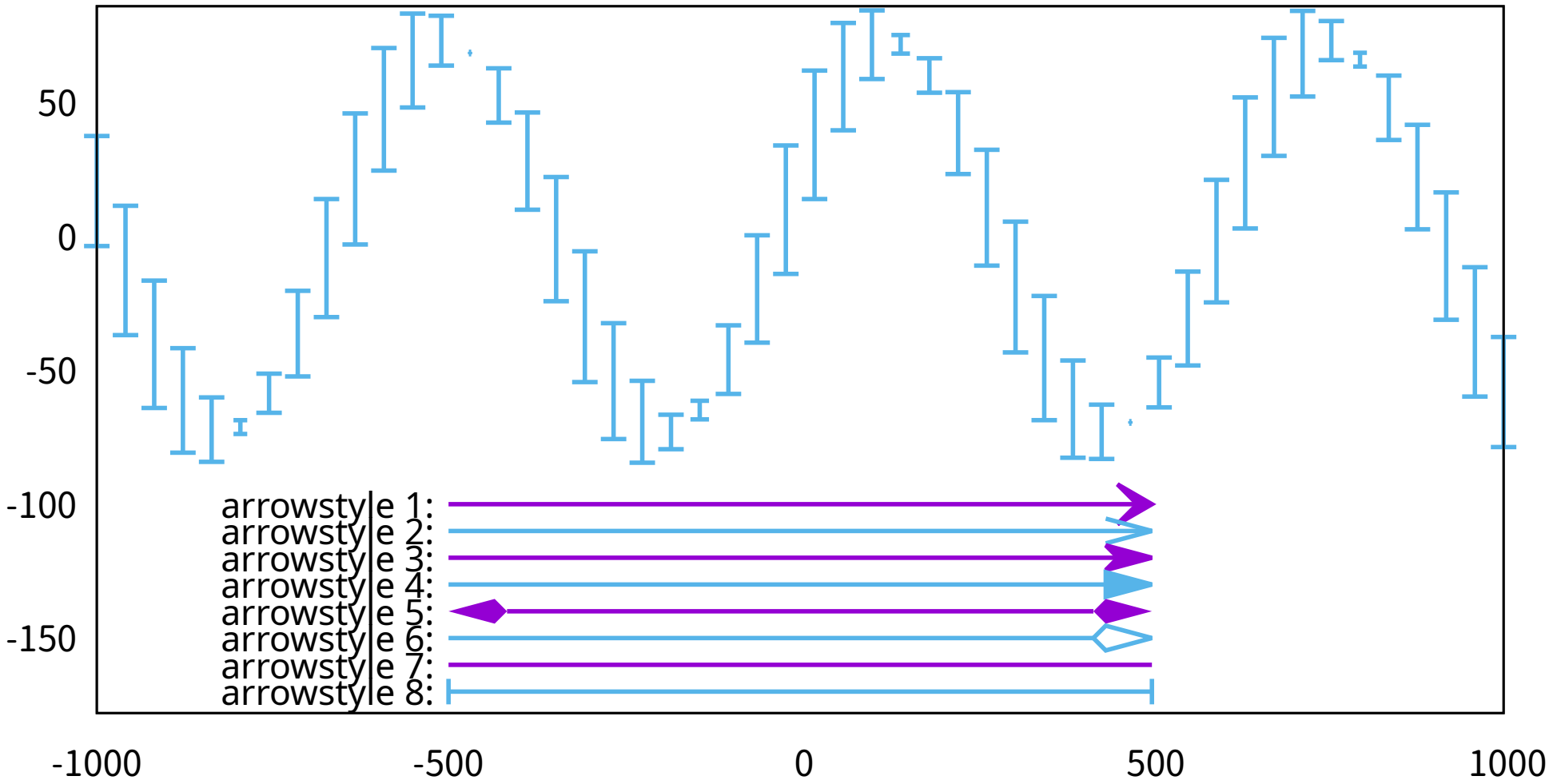
Top: plot with vectors arrowstyle 6, Bottom: explicit arrows

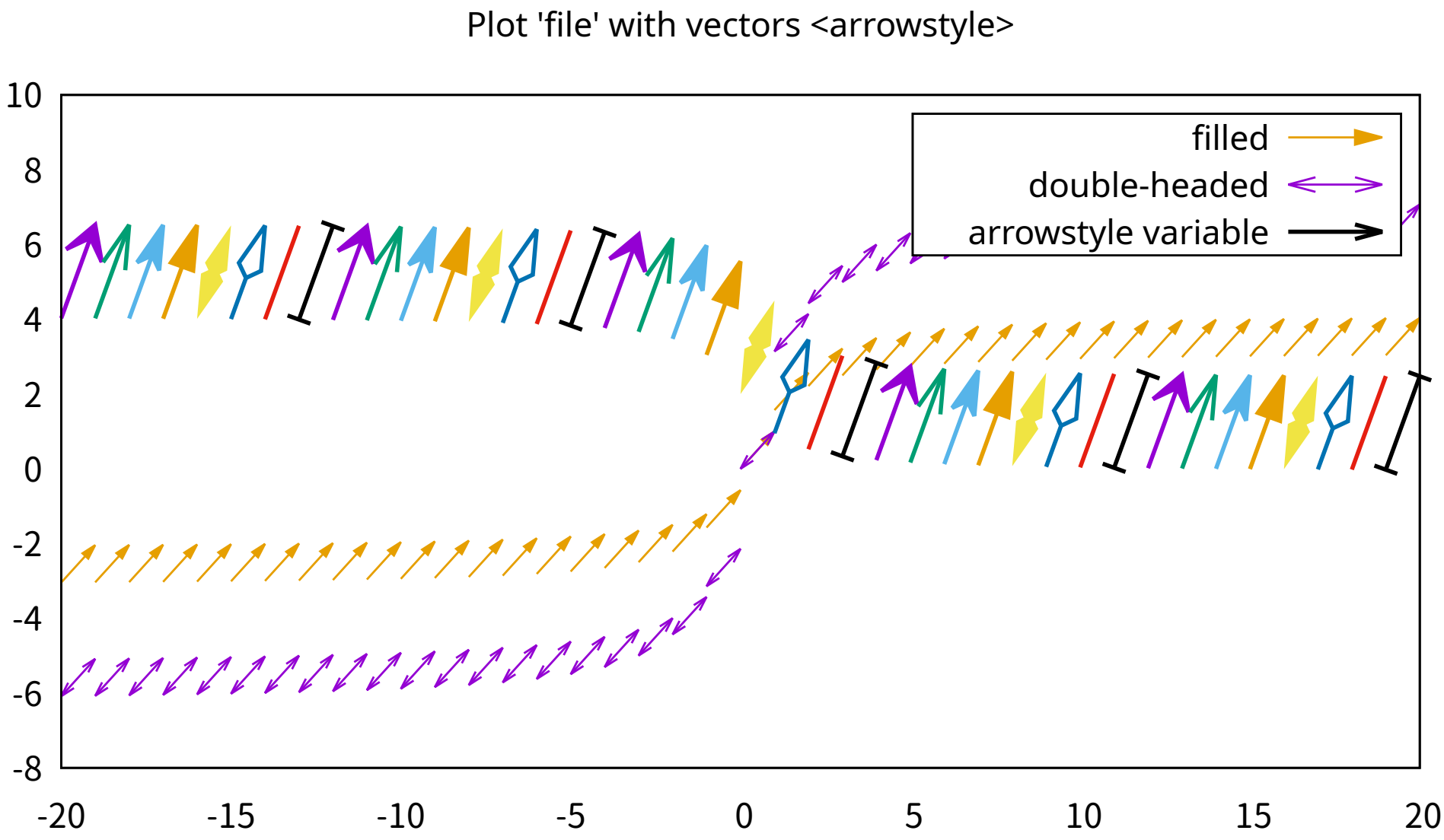


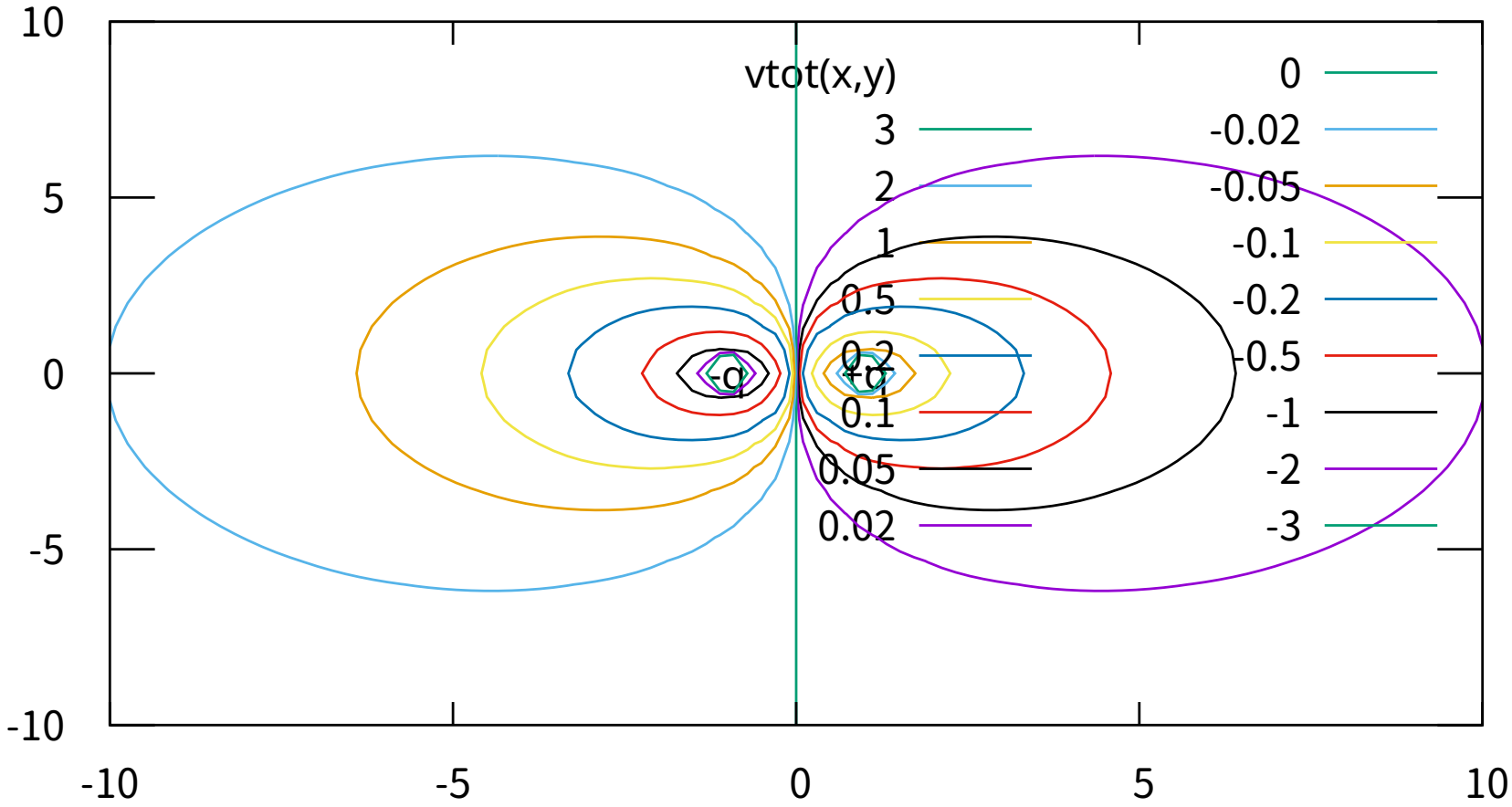
Top: plot with vectors arrowstyle 7, Bottom: explicit arrows

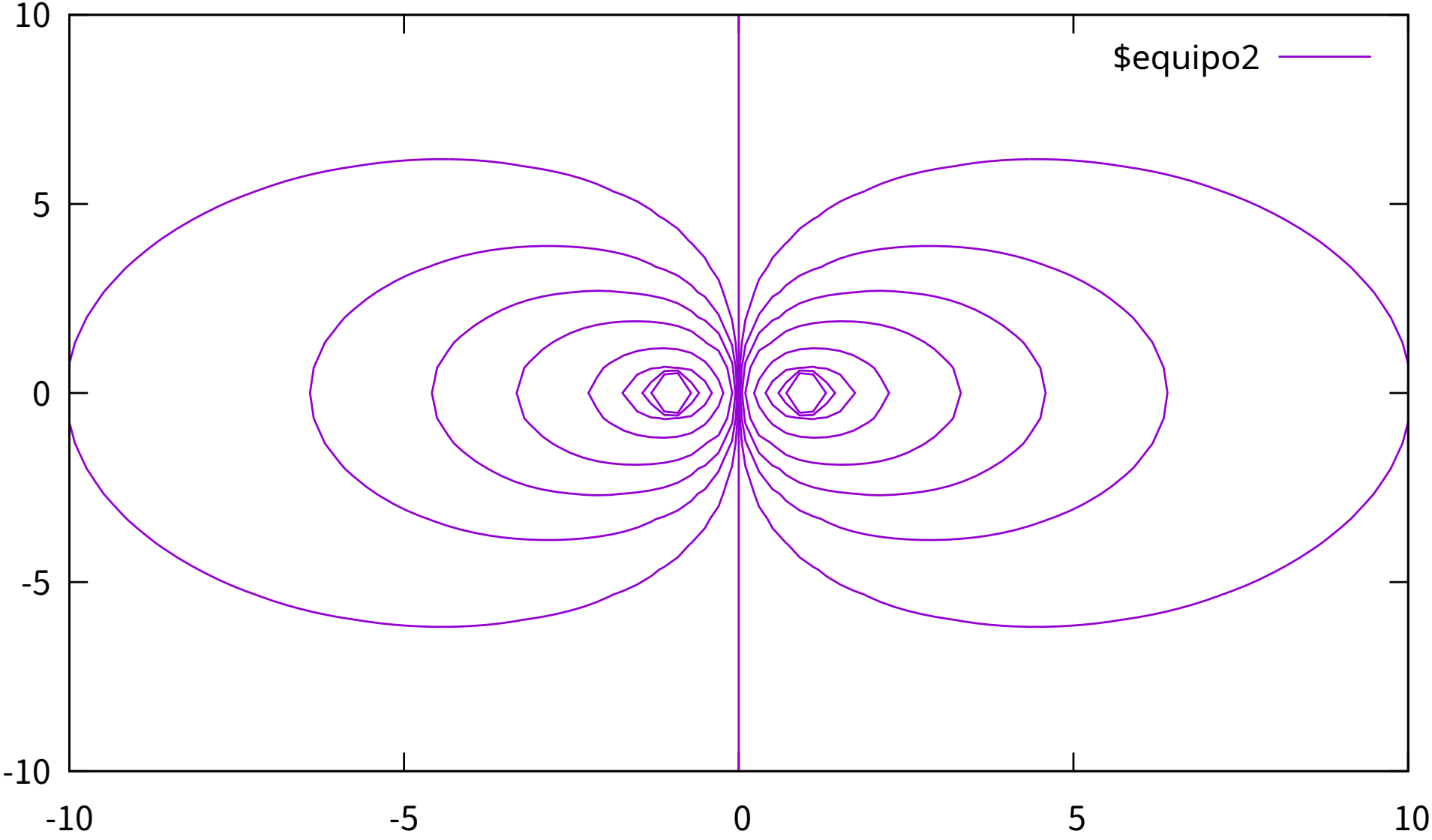


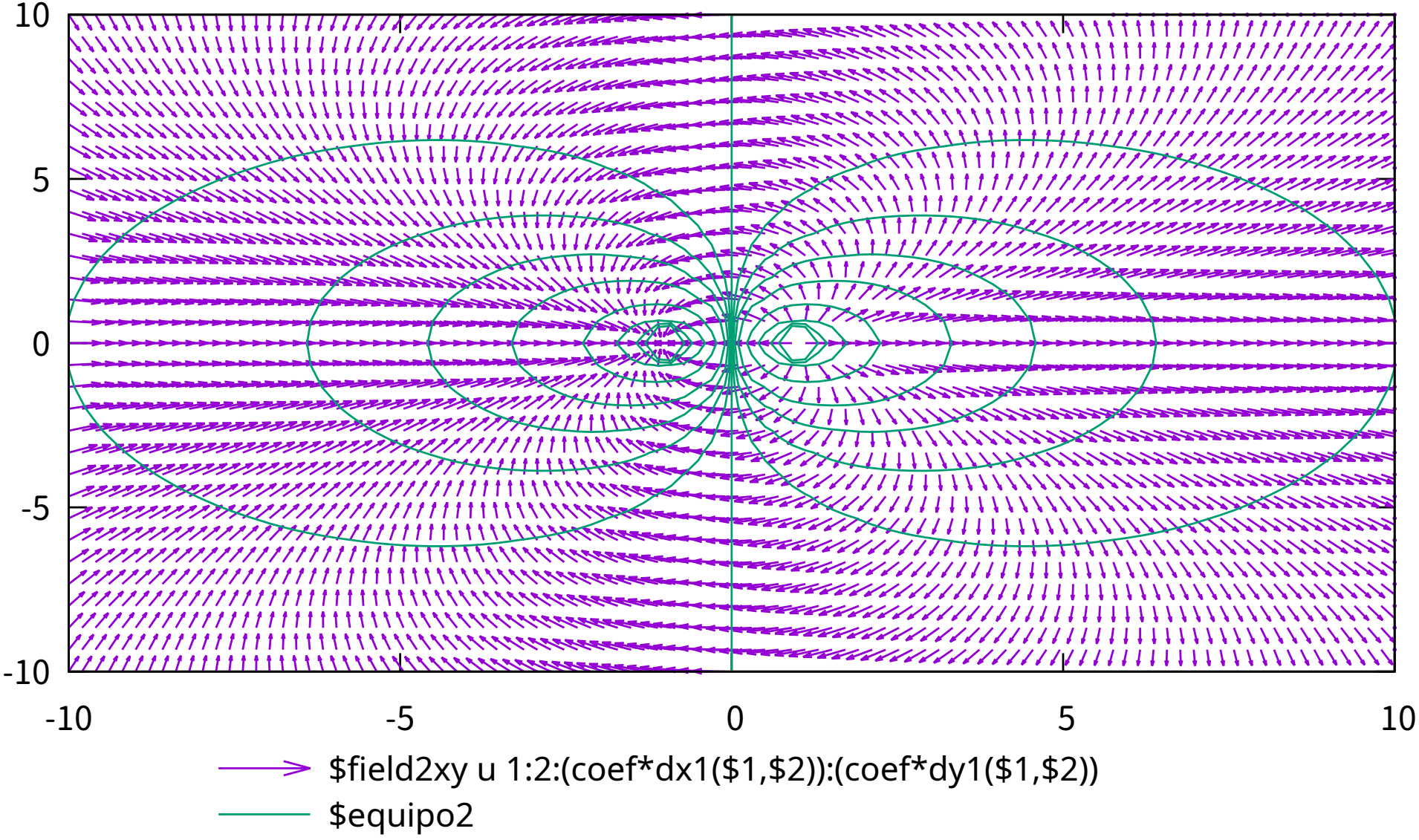
Top: plot with vectors arrowstyle 8, Bottom: explicit arrows

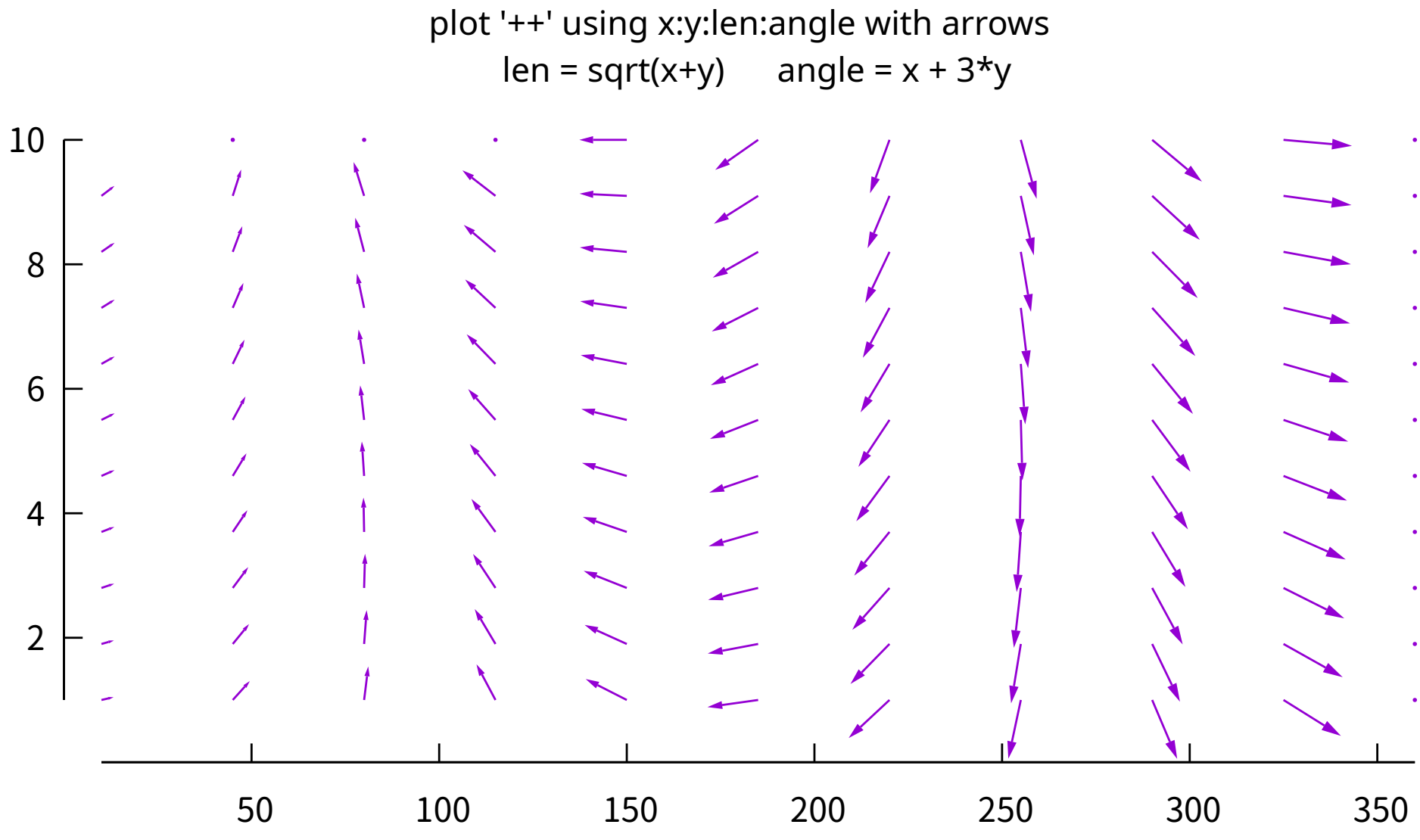


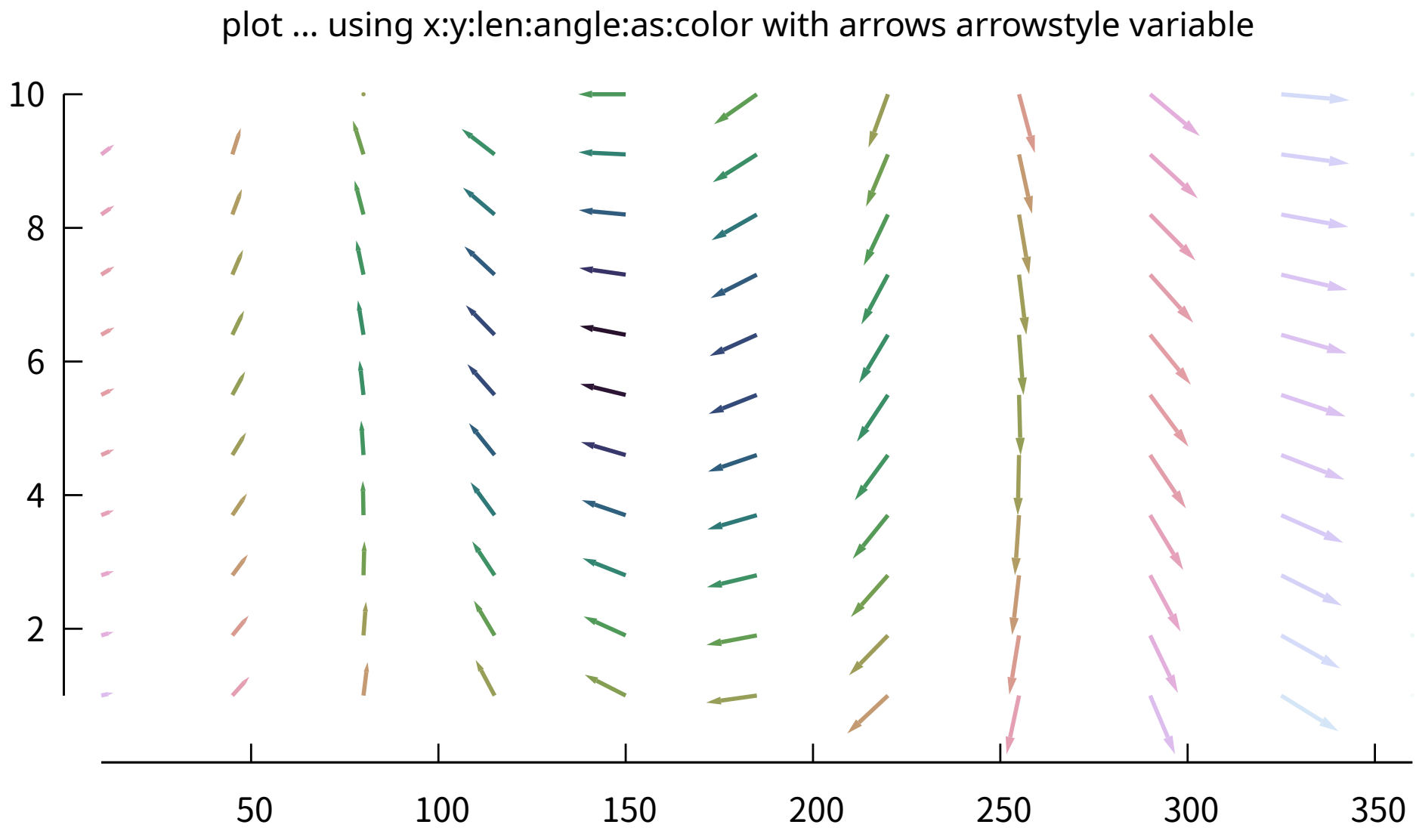

Plot 'file' with vectors <arrowstyle>
filled
double-headed
arrowstyle variable
10
8
6
4
2
0
-2
-4
-6
-8
-20
-15
-10
-5
0
5
10
15
20


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
-0
+0
10
5
0
-5
-10
-10
-5
0
5
10

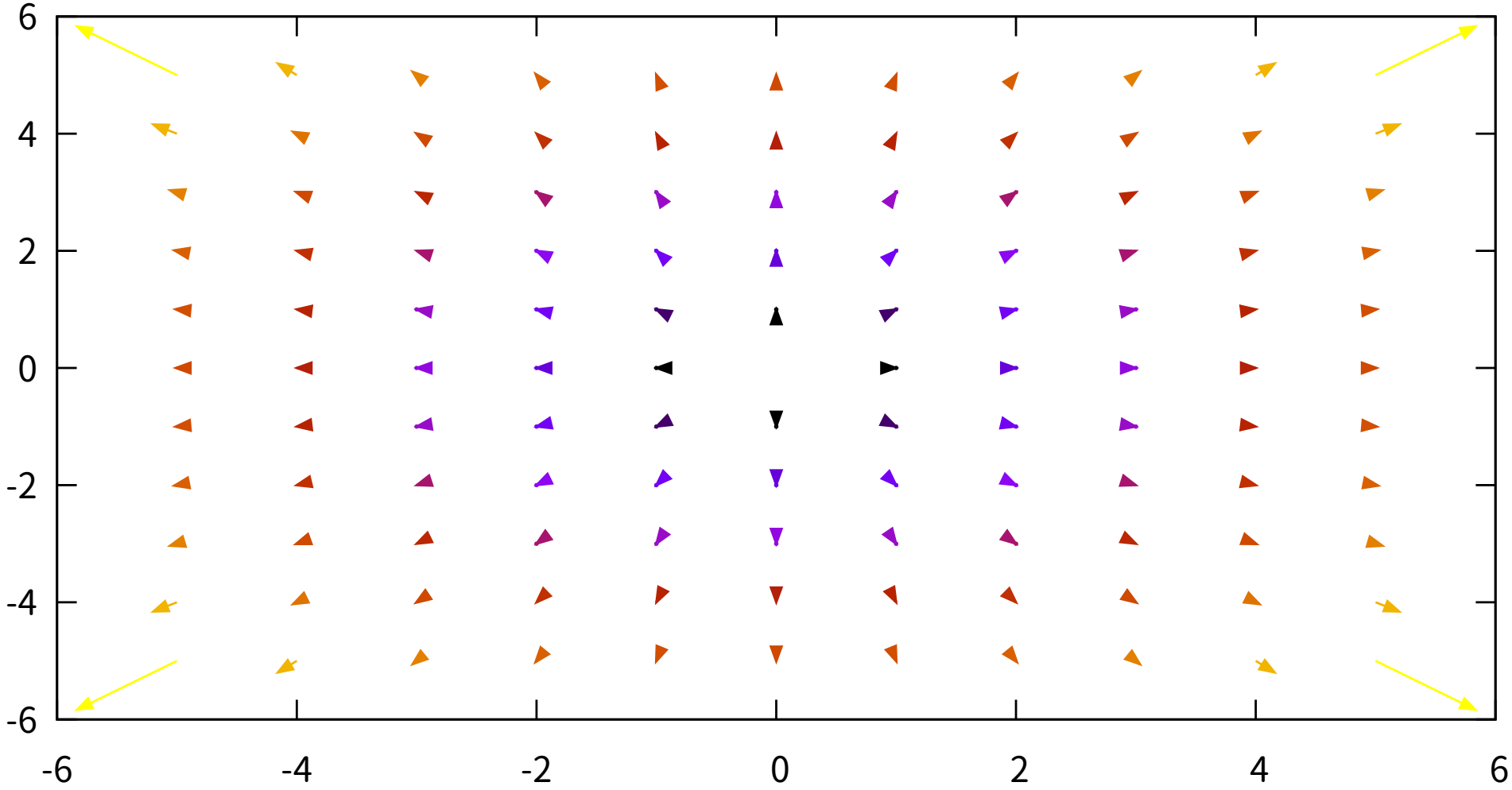

$equipo2
10
5
0
-5
-10
-10
-5
0
5
10

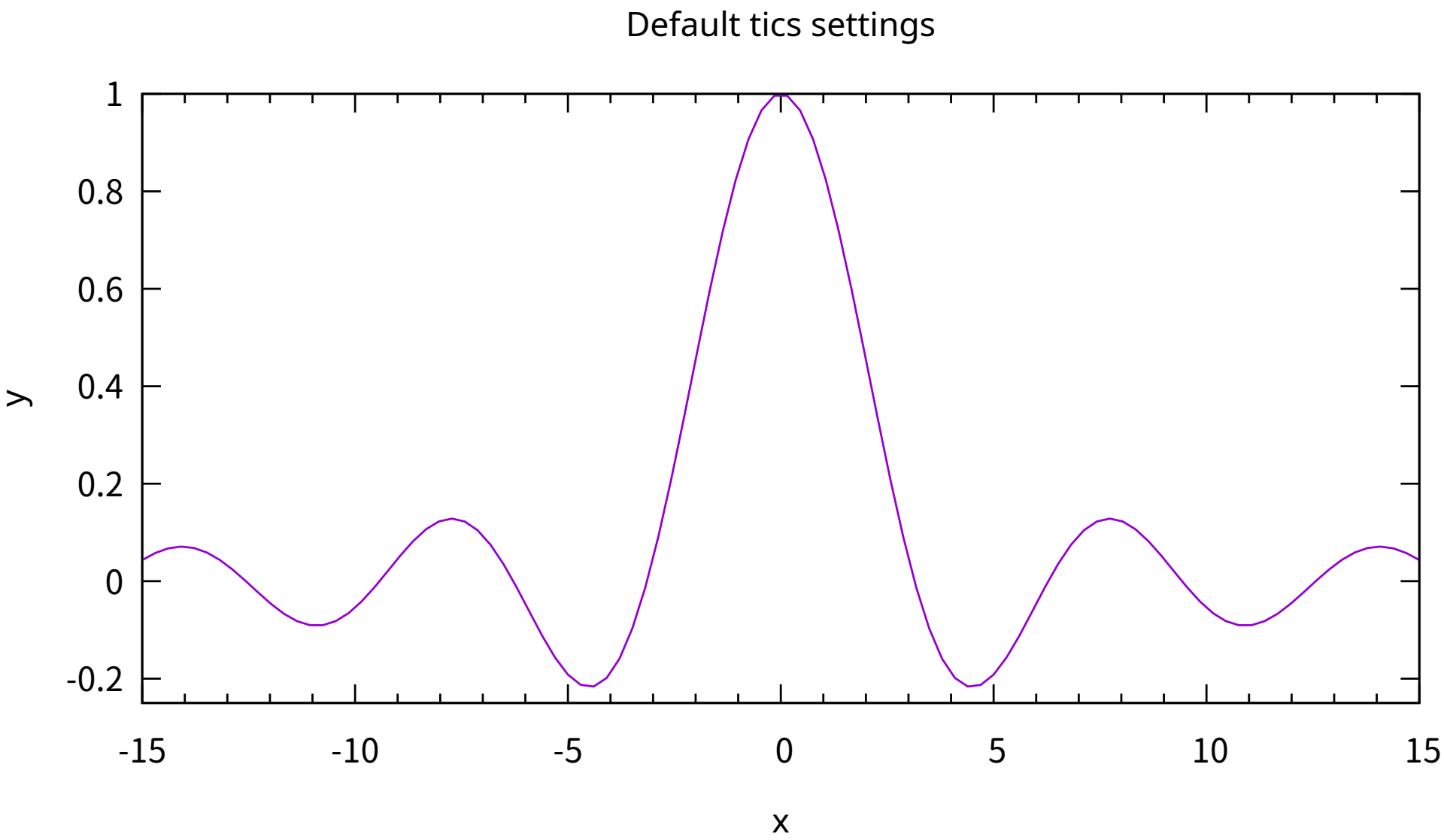

10
5
0
-5
-10
-10
-5
0
5
10
$field2xy u 1:2:(coef*dx1($1,$2)):(coef*dy1($1,$2))
$equipo2

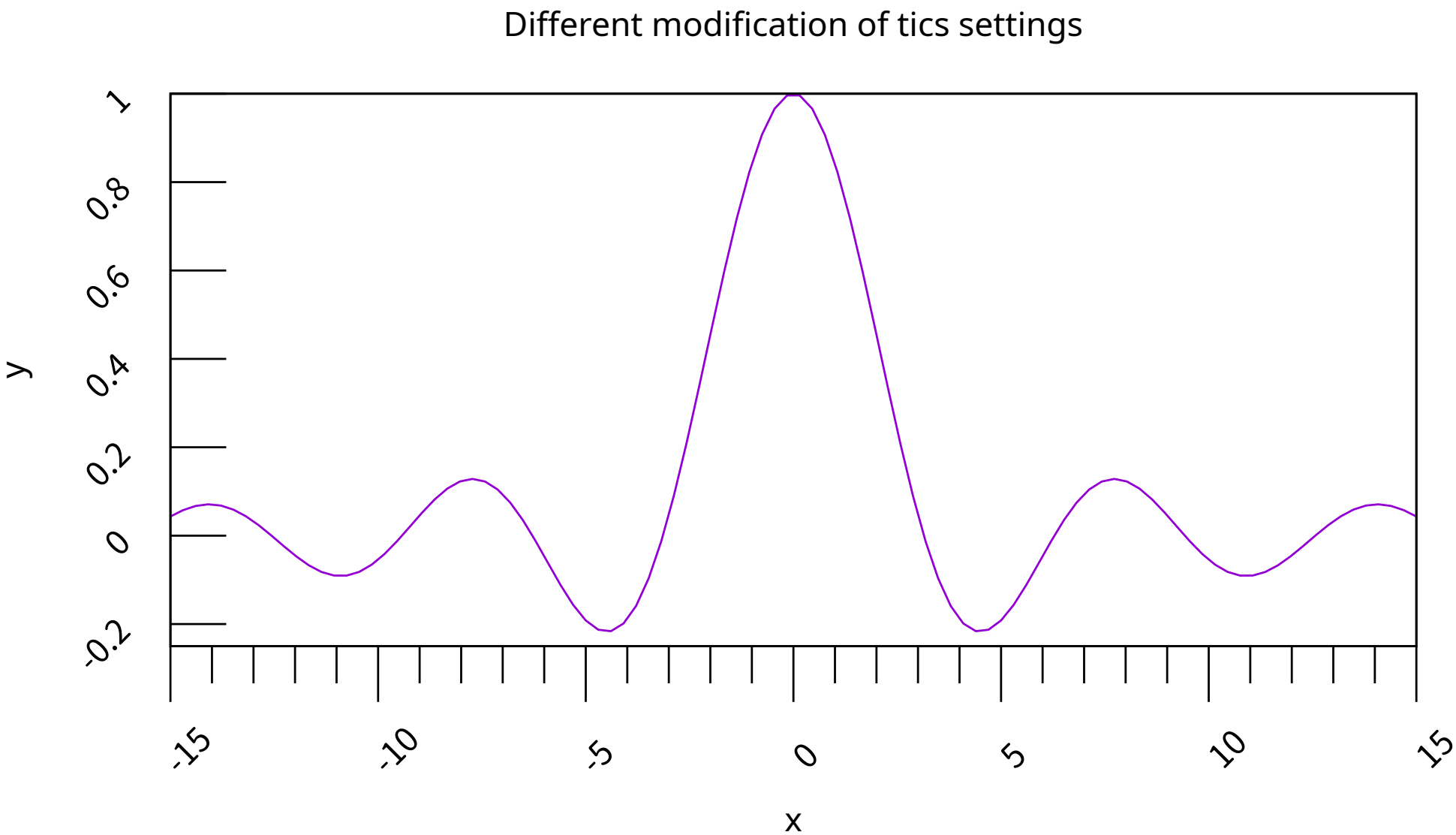

plot '++' using x:y:len:angle with arrows
len = sqrt(x+y) angle = x + 3*y
10
8
6
4
2
50
100
150
200
250
300
350

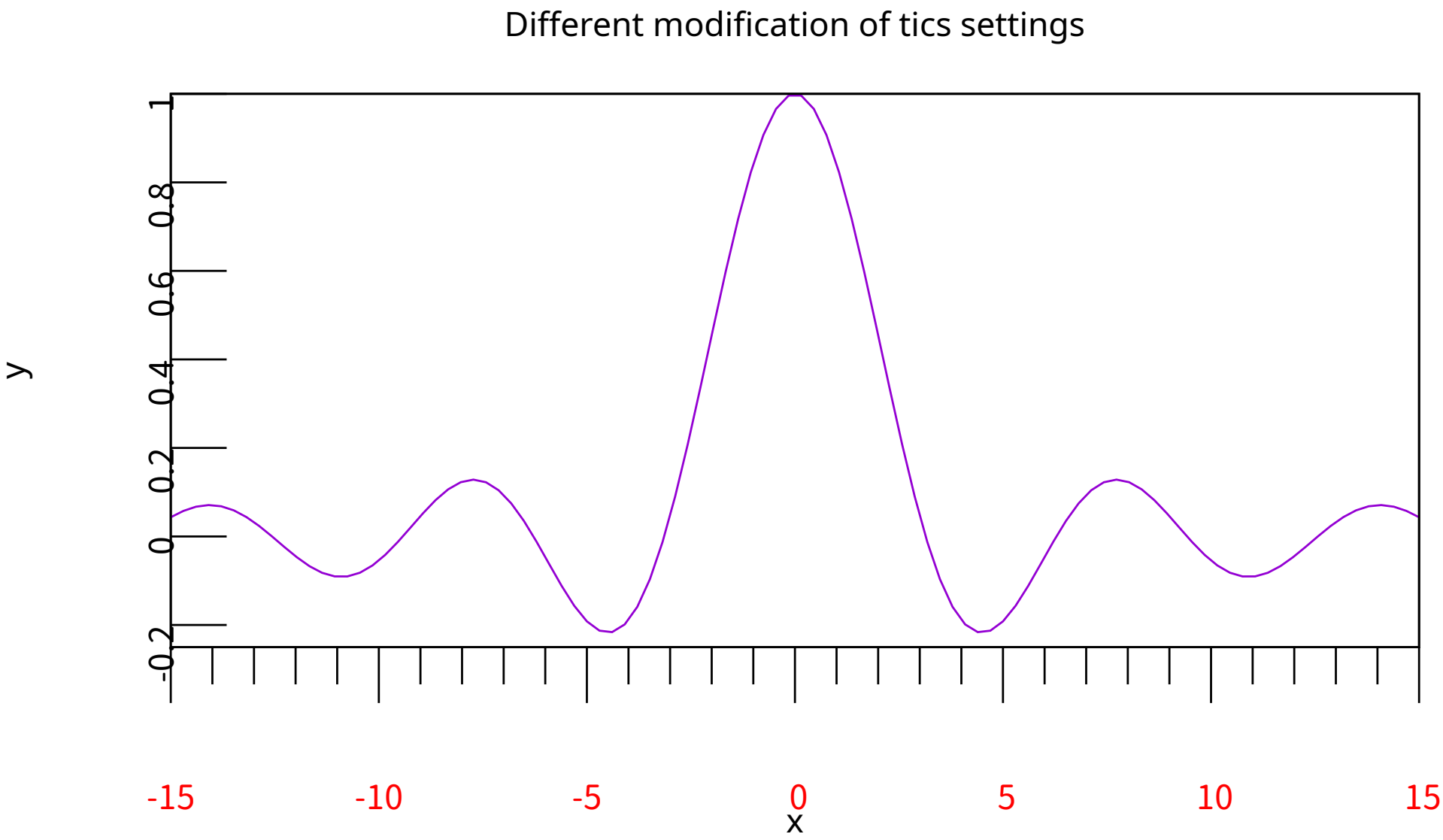

plot ... using x:y:len:angle:as:color with arrows arrowstyle variable
10
8
6
4
2
50
100
150
200
250
300
350

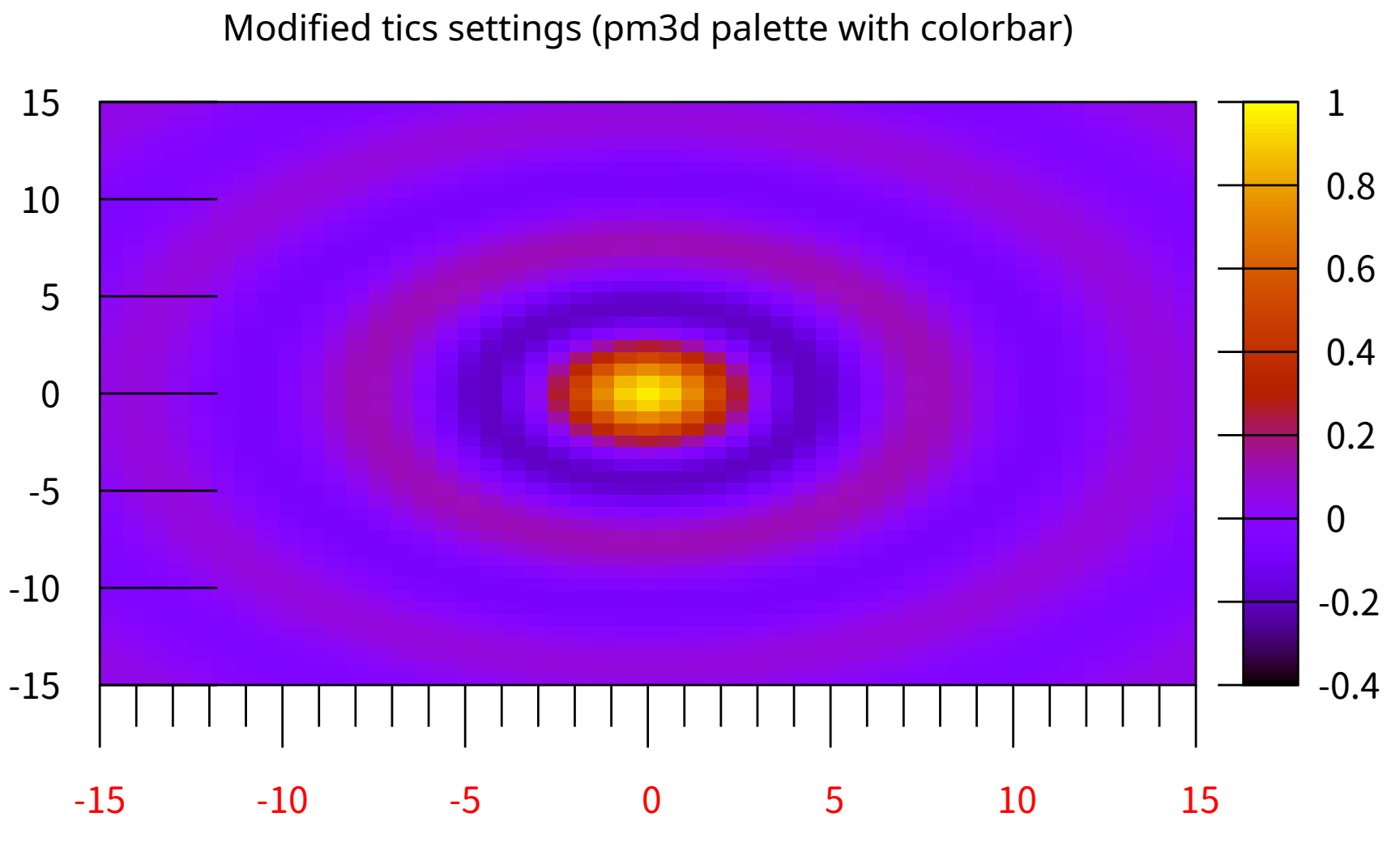
fixed size arrowheads for very short vectors



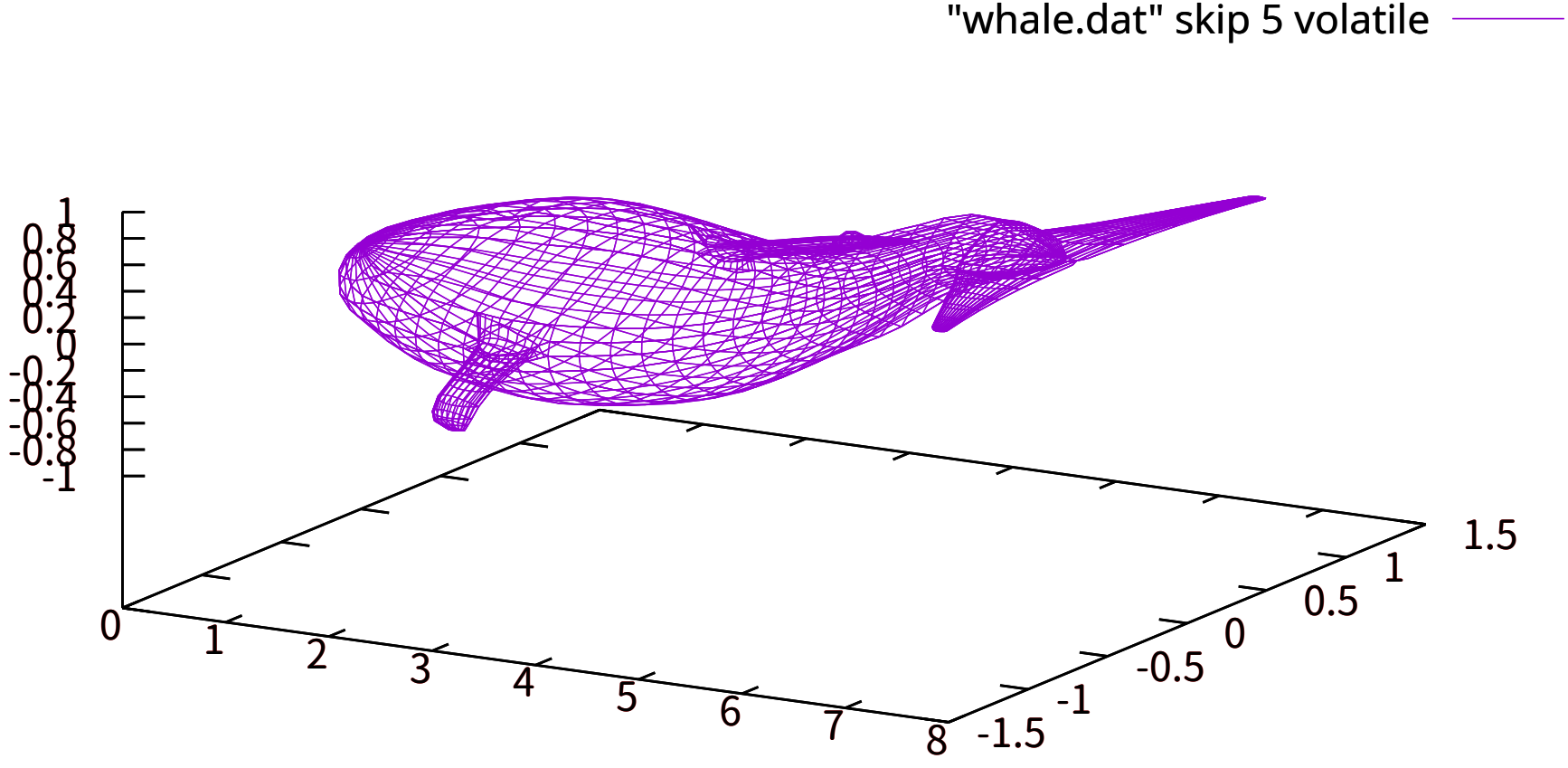

Default tics settings
1
0.8
0.6
0.4
0.2
0
-0.2
y
-15
-10
-5
0
5
10
15
x

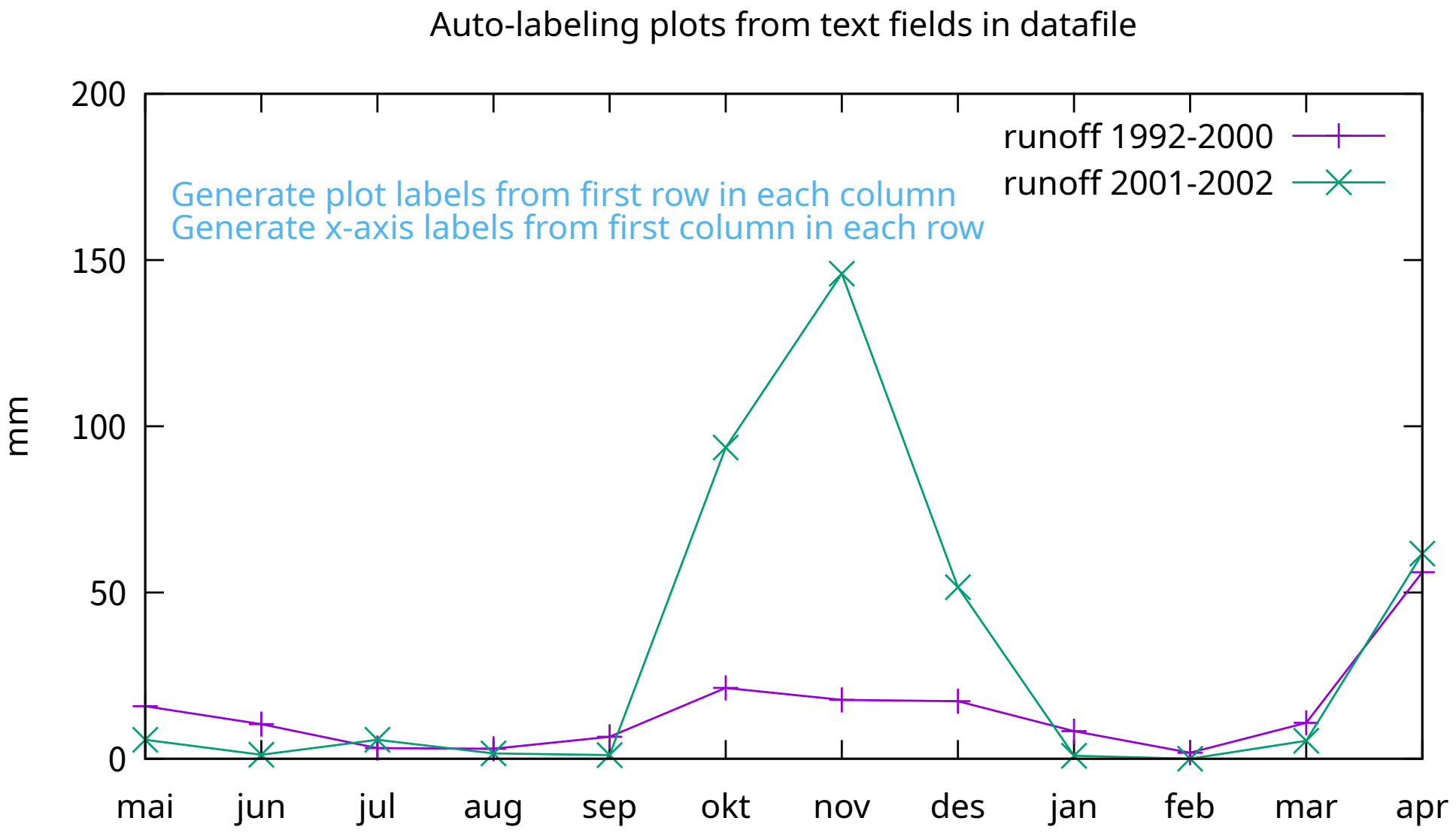

Different modification of tics settings
y
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x

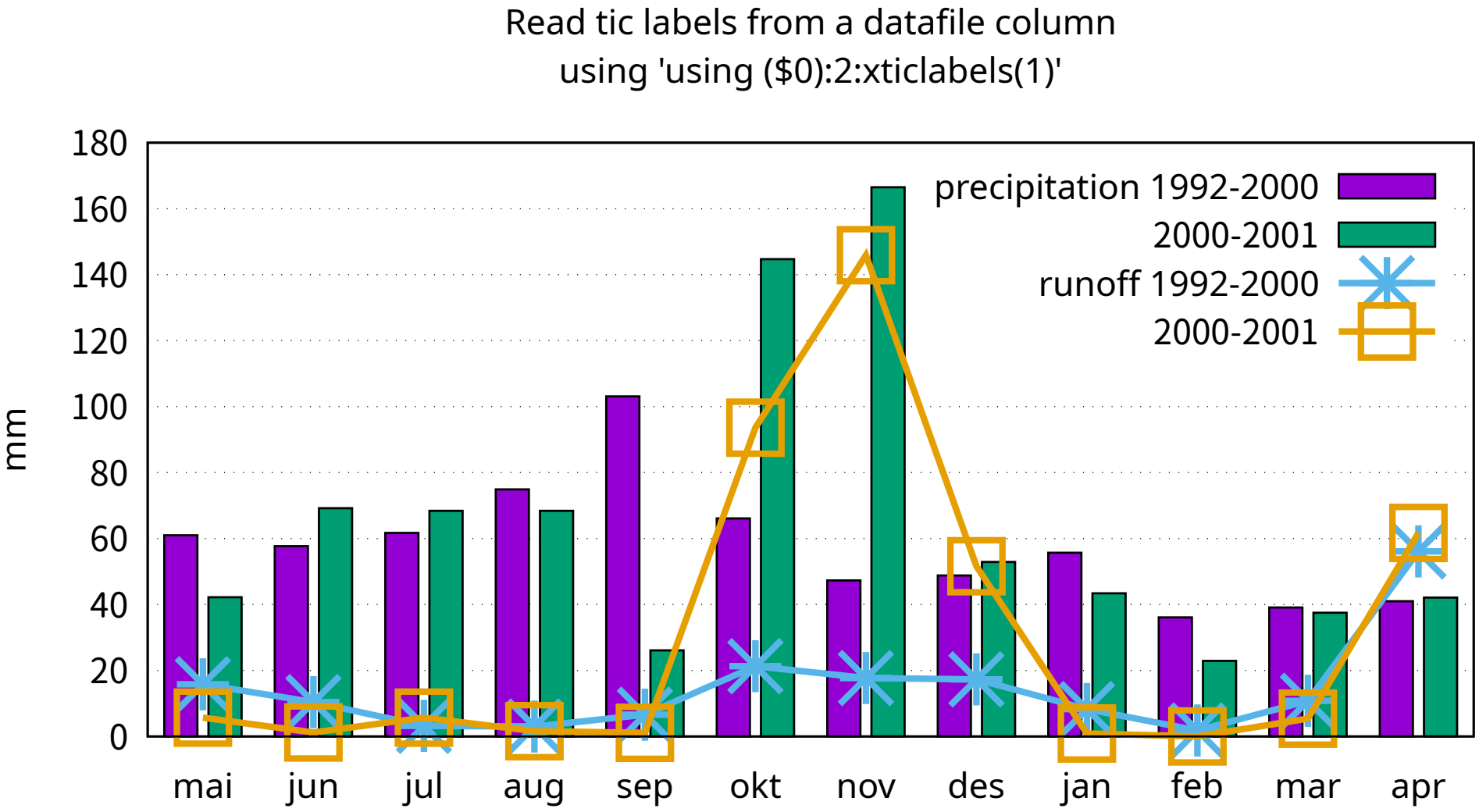

Different modification of tics settings
y
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x

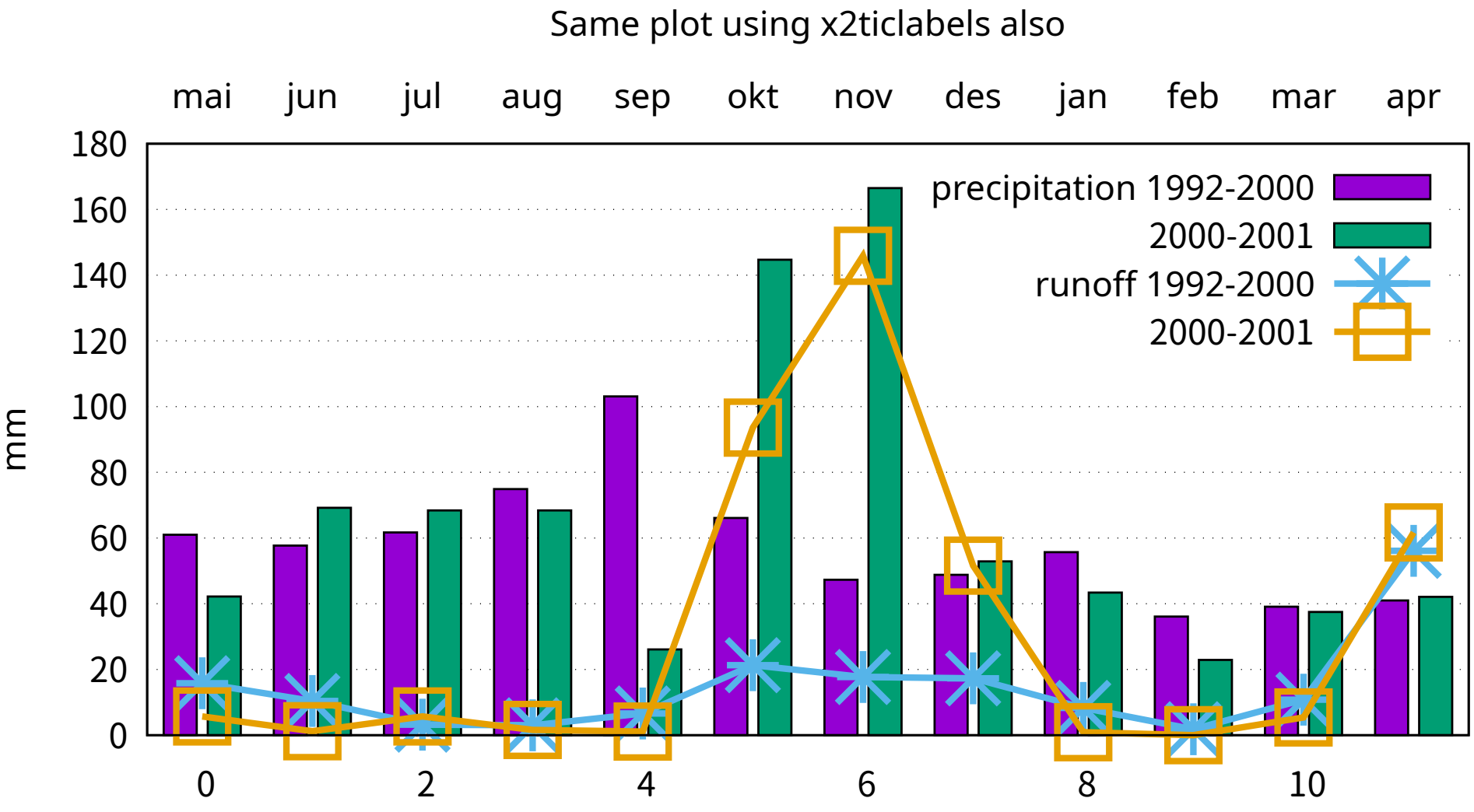

Modified tics settings (pm3d palette with colorbar)
15
10
5
0
-5
-10
-15
-15
-10
-5
0
5
10
15
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

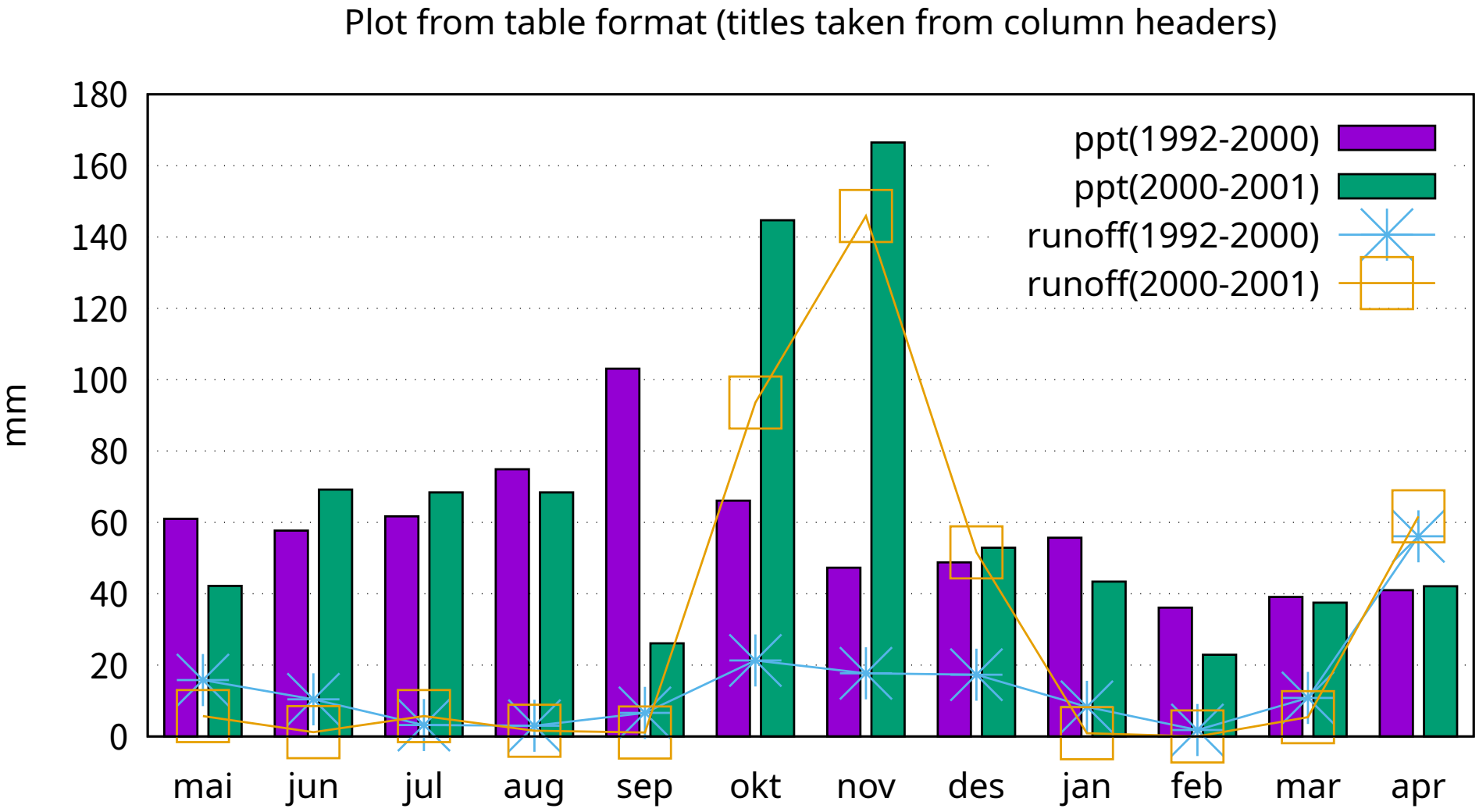
Nothing interesting here, just a unit test for volatile, skip, and refresh

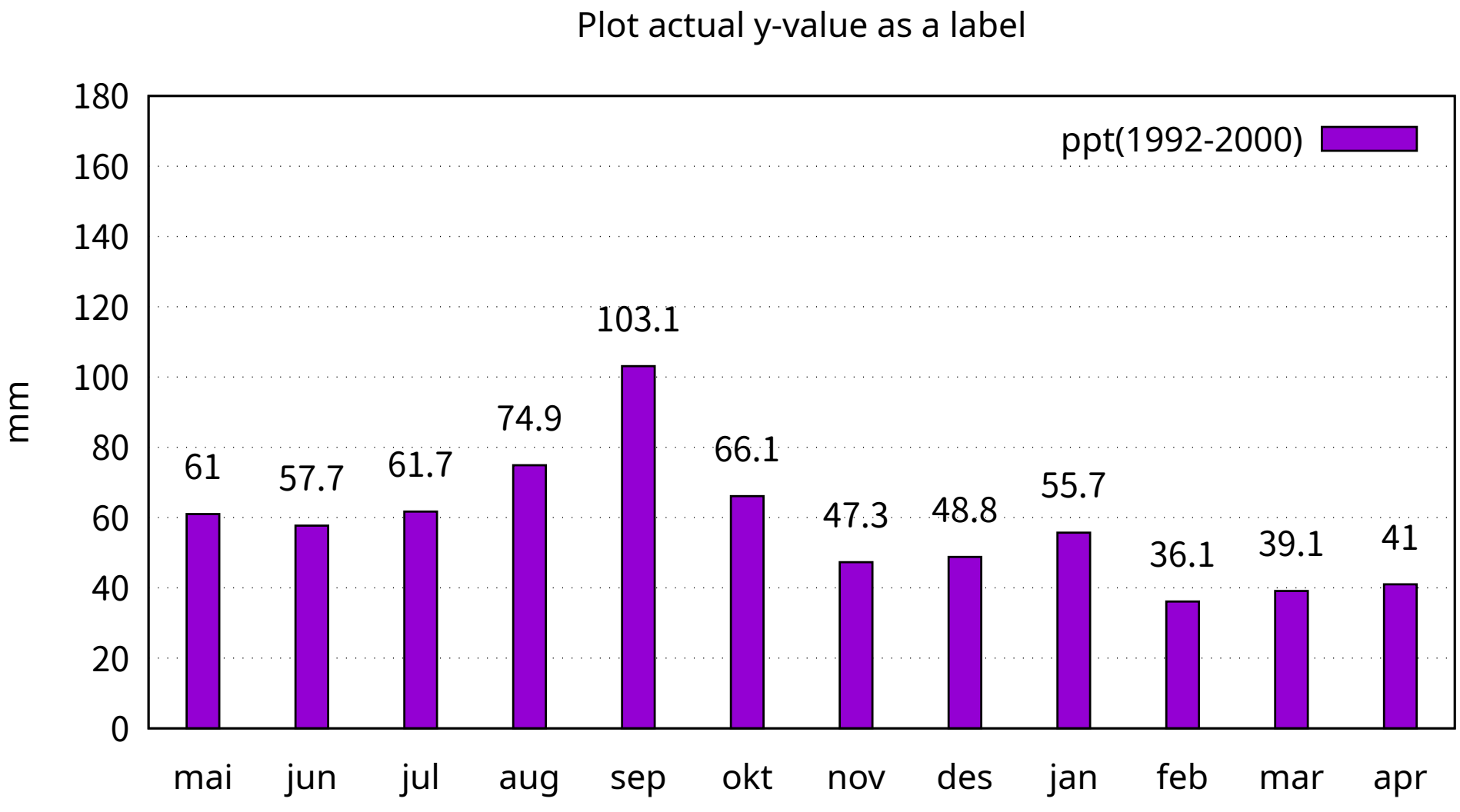


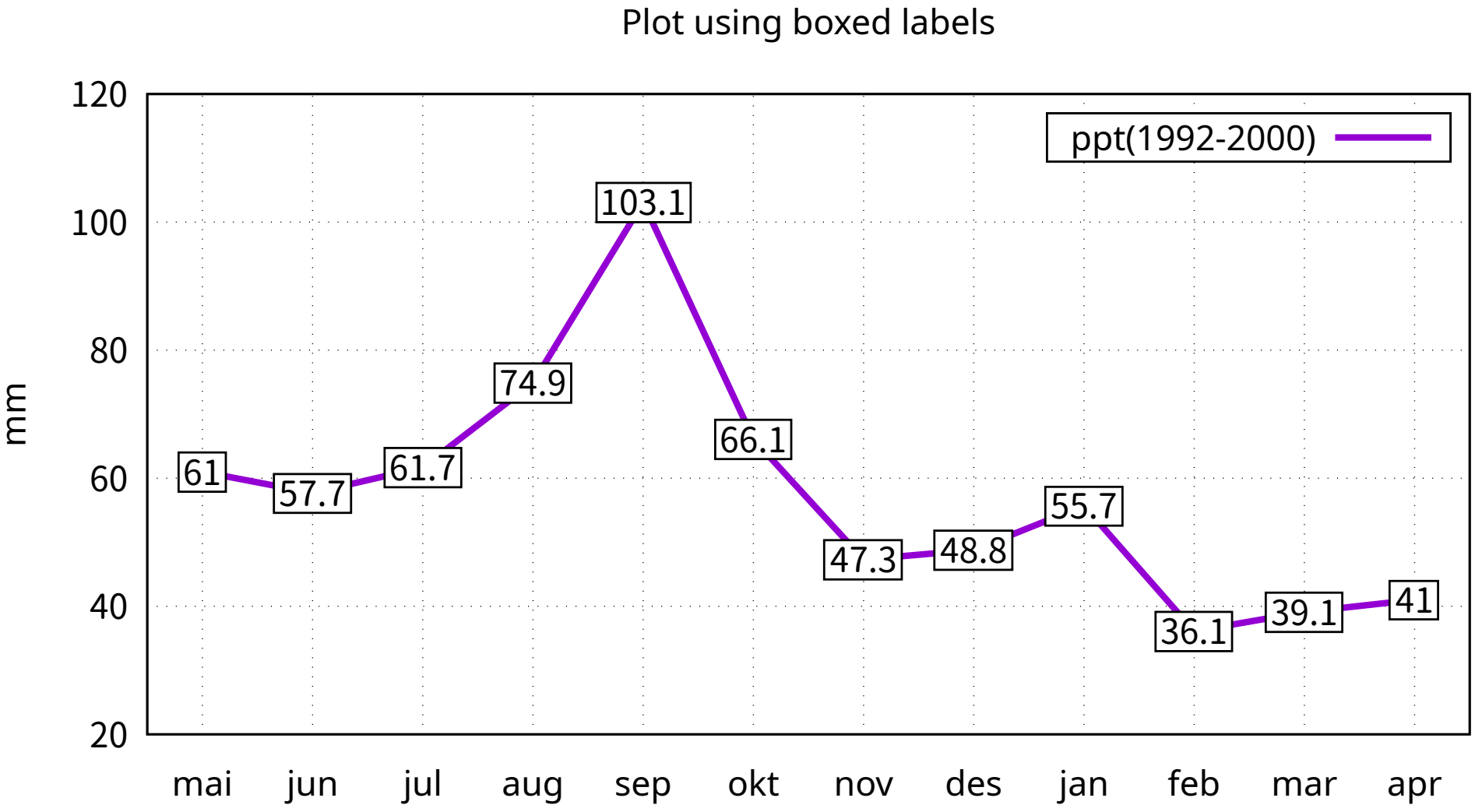

Auto-labeling plots from text fields in datafile
Generate plot labels from first row in each column
Generate x-axis labels from first column in each row
runoff 1992-2000
runoff 2001-2002
mm
200
150
100
50
0
mai
jun
jul
aug
sep
okt
nov
des
jan
feb
mar
apr

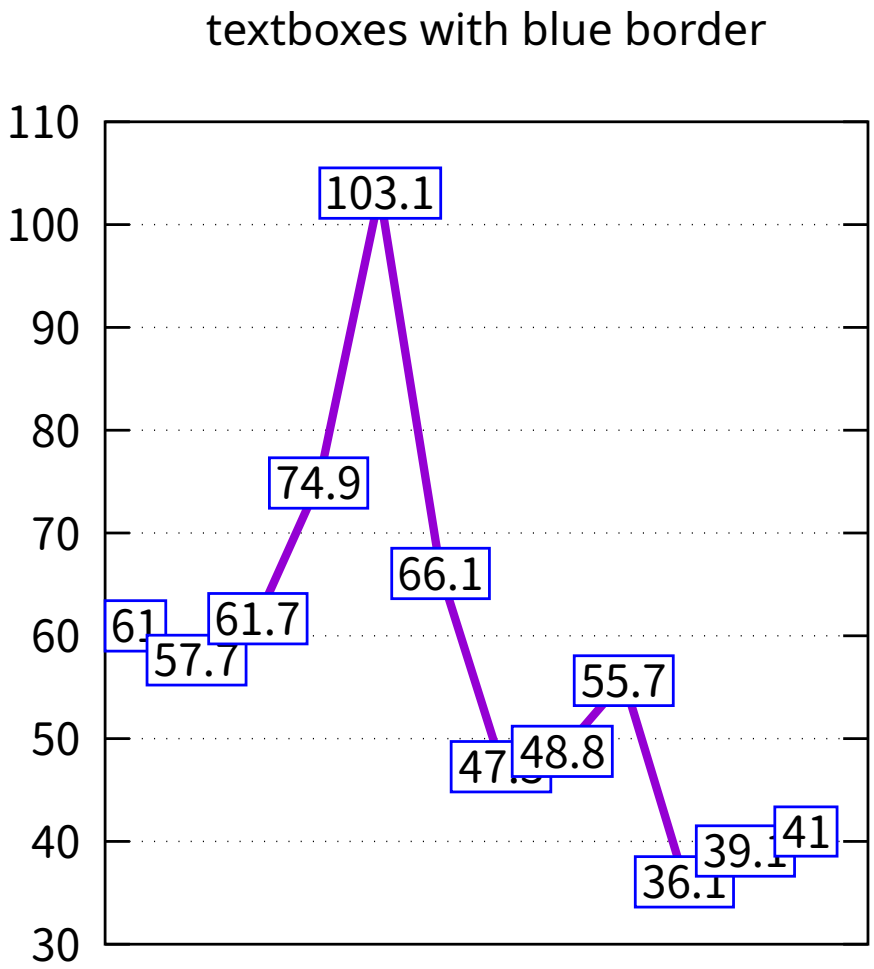

Read tic labels from a datafile column
using 'using ($0):2:xticlabels(1)'
mm
180
160
140
120
100
80
60
40
20
0
mai
jun
jul
aug
sep
okt
nov
des
jan
feb
mar
apr
precipitation 1992-2000
2000-2001
runoff 1992-2000
2000-2001

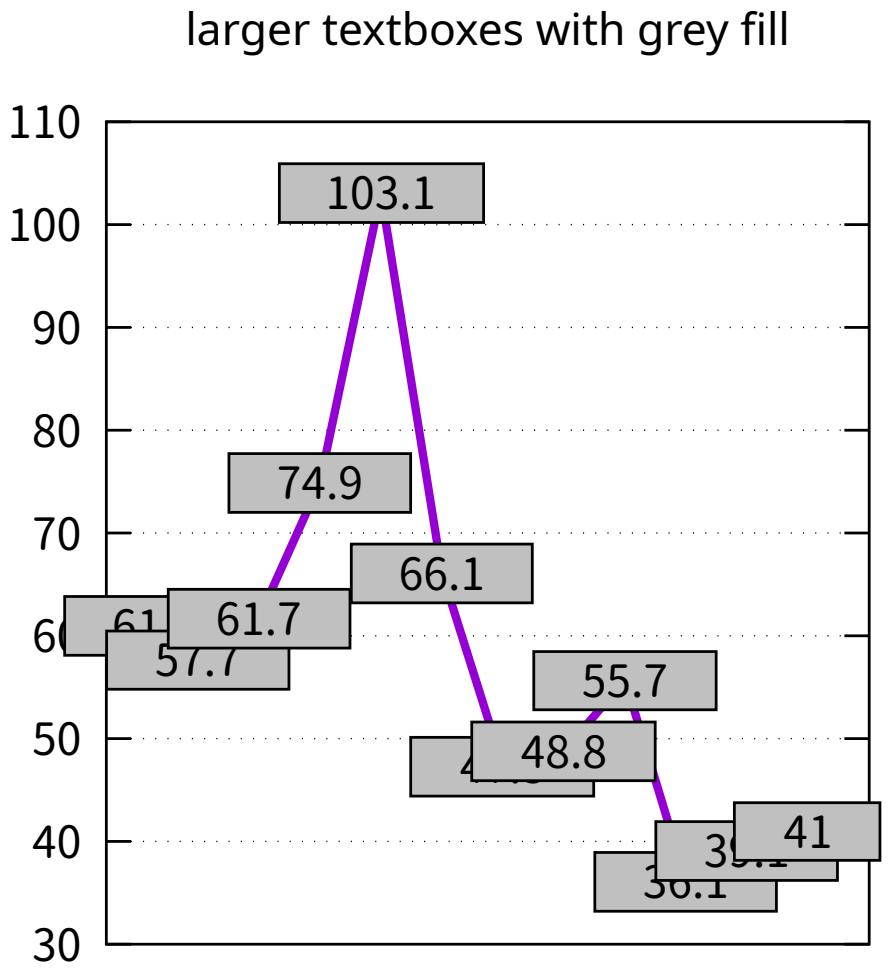

Same plot using x2ticlabels also
mai
jun
jul
aug
sep
okt
nov
des
jan
feb
mar
apr
180
160
140
120
100
80
60
40
20
0
mm
0
2
4
6
8
10
precipitation 1992-2000
2000-2001
runoff 1992-2000
2000-2001


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)

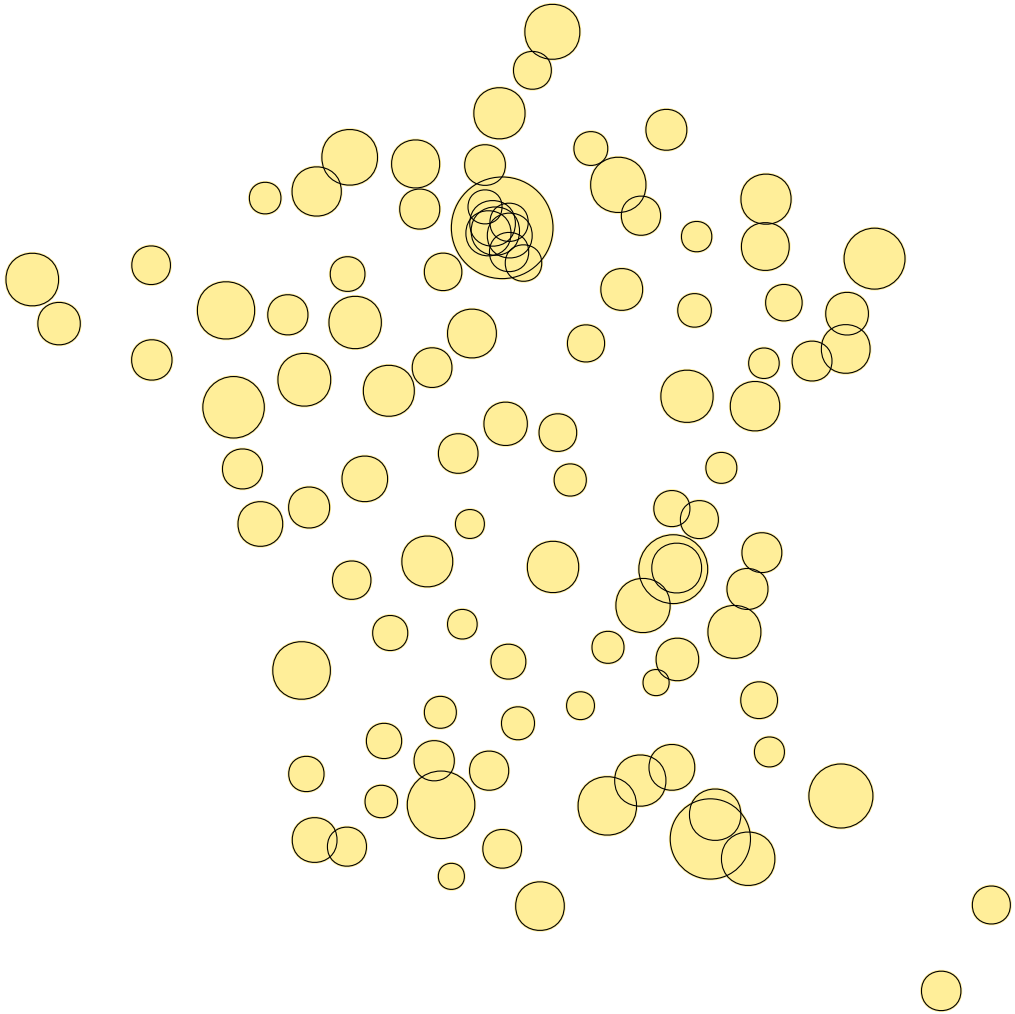

Plot actual y-value as a label
ppt(1992-2000)
mm
180
160
140
120
100
80
60
40
20
0
61
57.7
61.7
74.9
103.1
66.1
47.3
48.8
55.7
36.1
39.1
41
mai
jun
jul
aug
sep
okt
nov
des
jan
feb
mar
apr

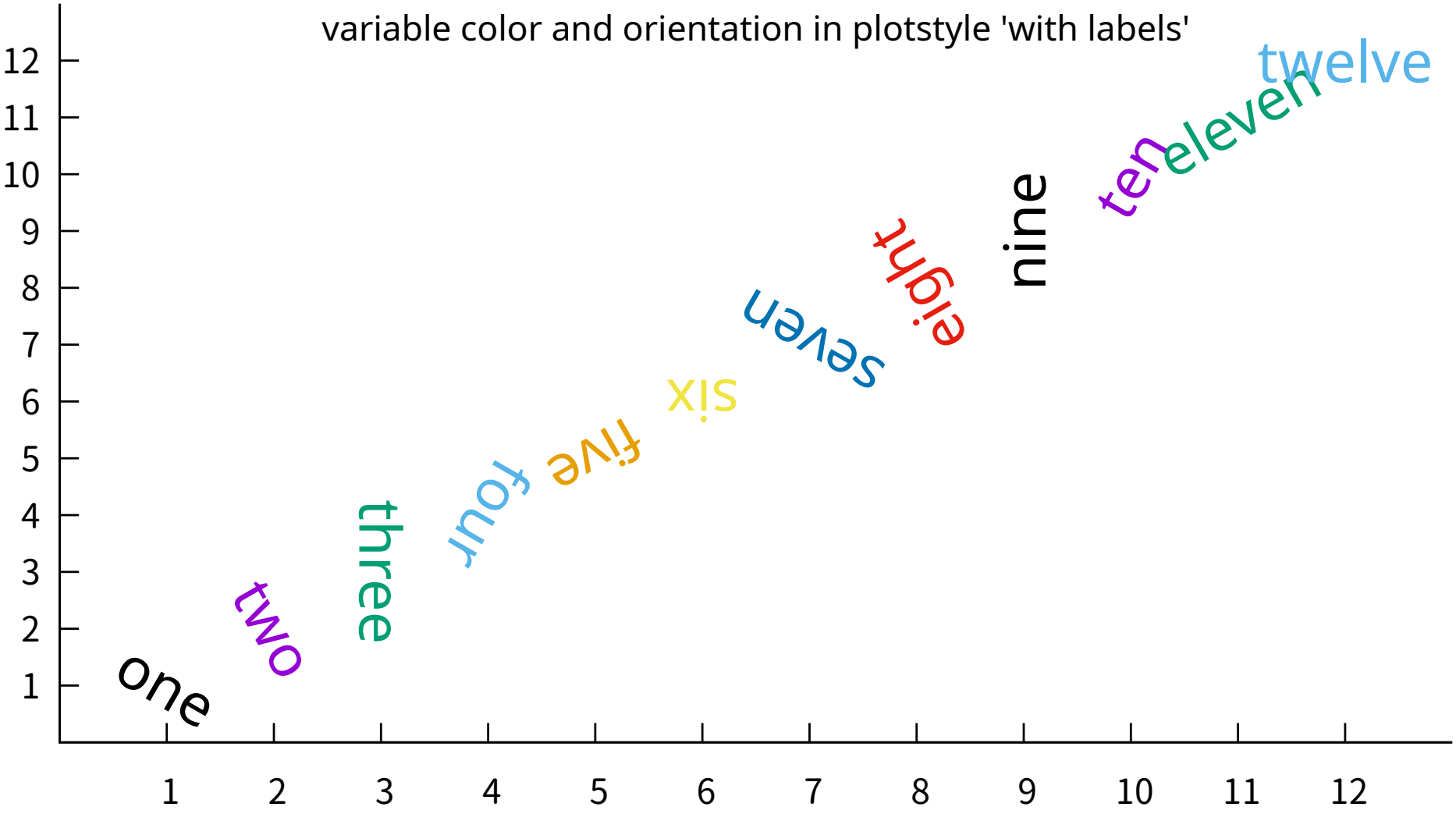

Plot using boxed labels
ppt(1992-2000)
mm
120
100
80
60
40
20
61
57.7
61.7
74.9
103.1
66.1
47.3
48.8
55.7
36.1
39.1
41
mai
jun
jul
aug
sep
okt
nov
des
jan
feb
mar
apr

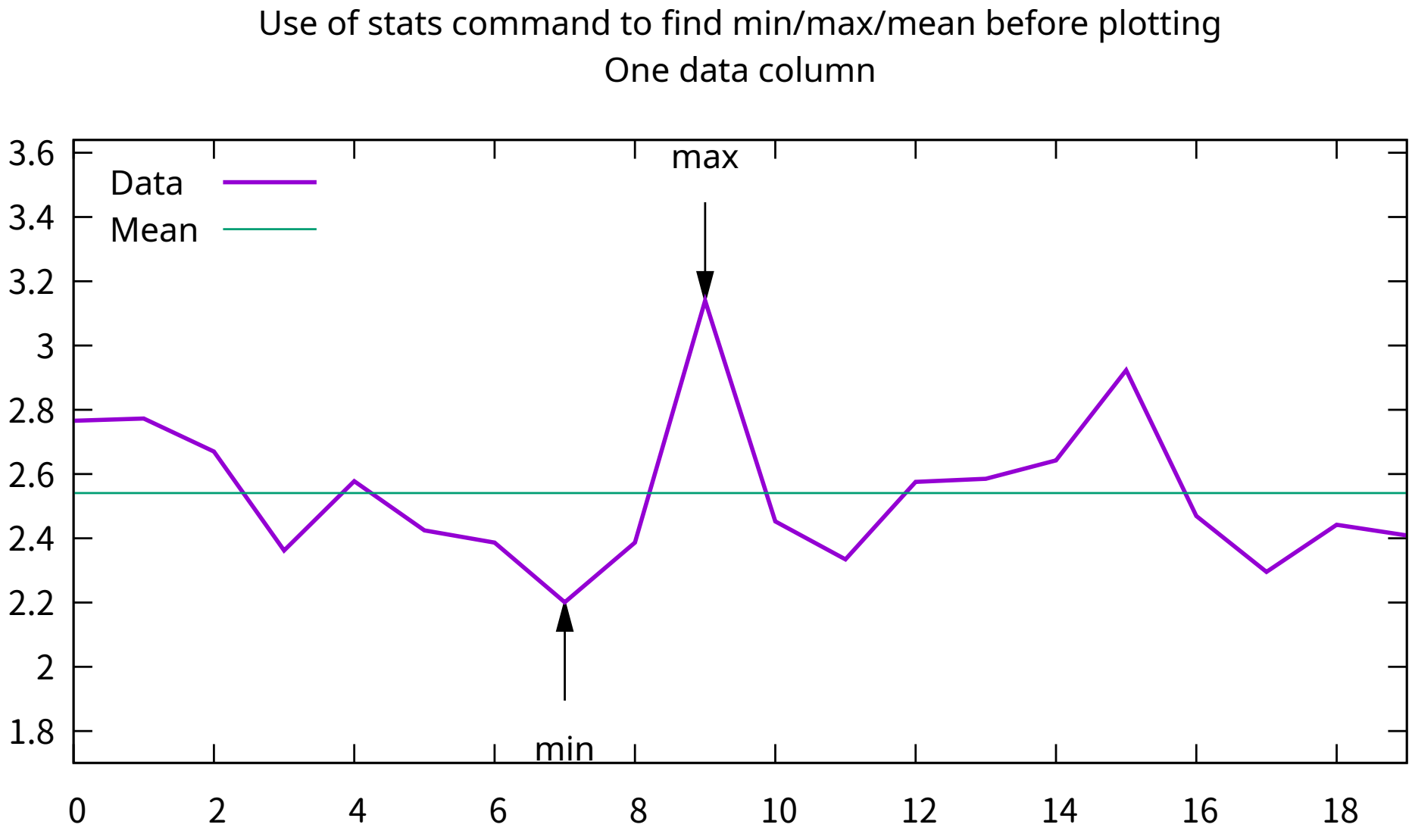

textboxes with blue border
110
100
90
80
70
60
50
40
30
103.1
74.9
66.1
61
61.7
57.7
55.7
48.8
41
39.1
36.1

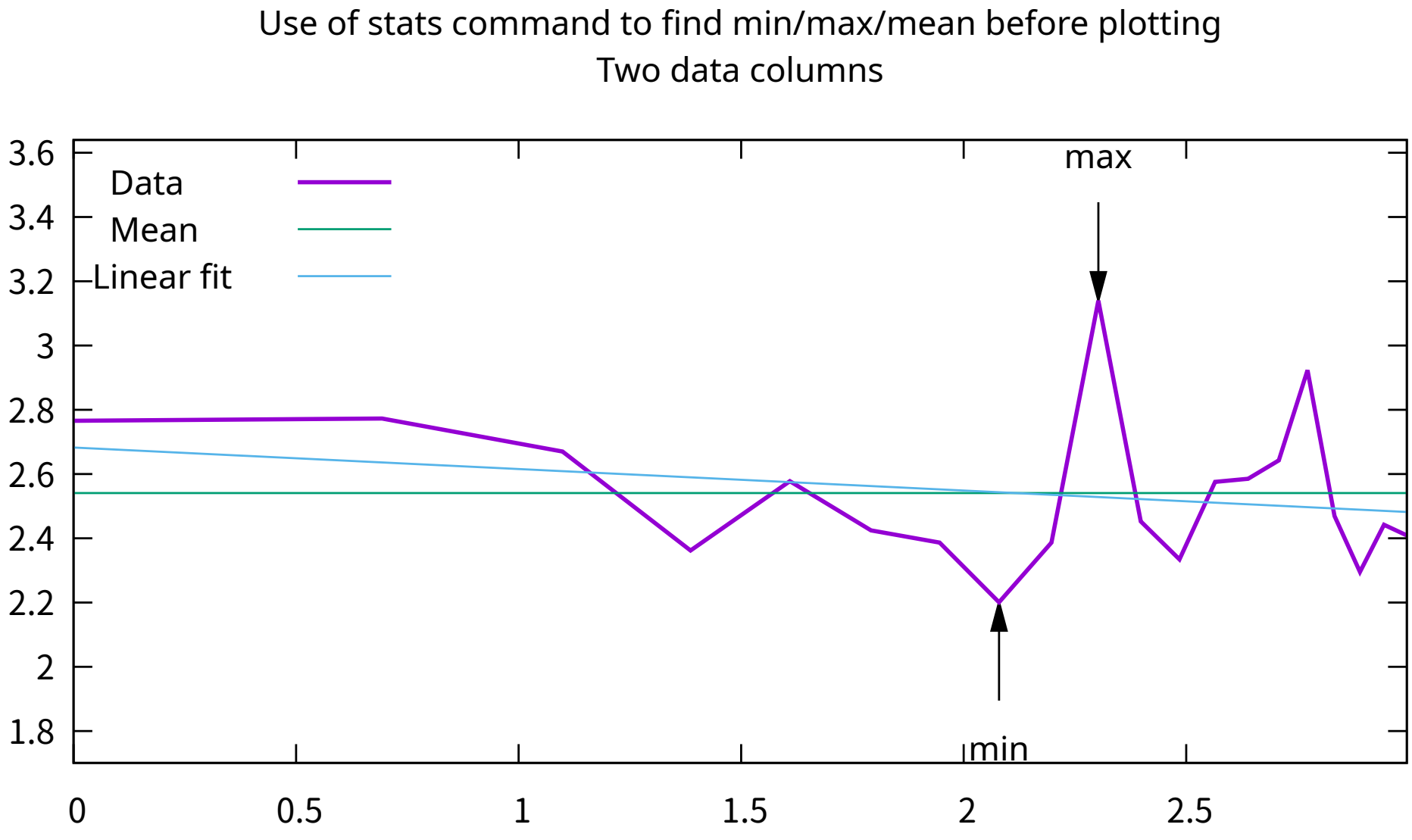

larger textboxes with grey fill
110
100
90
80
70
60
50
40
30
103.1
74.9
66.1
61.7
55.7
48.8
41

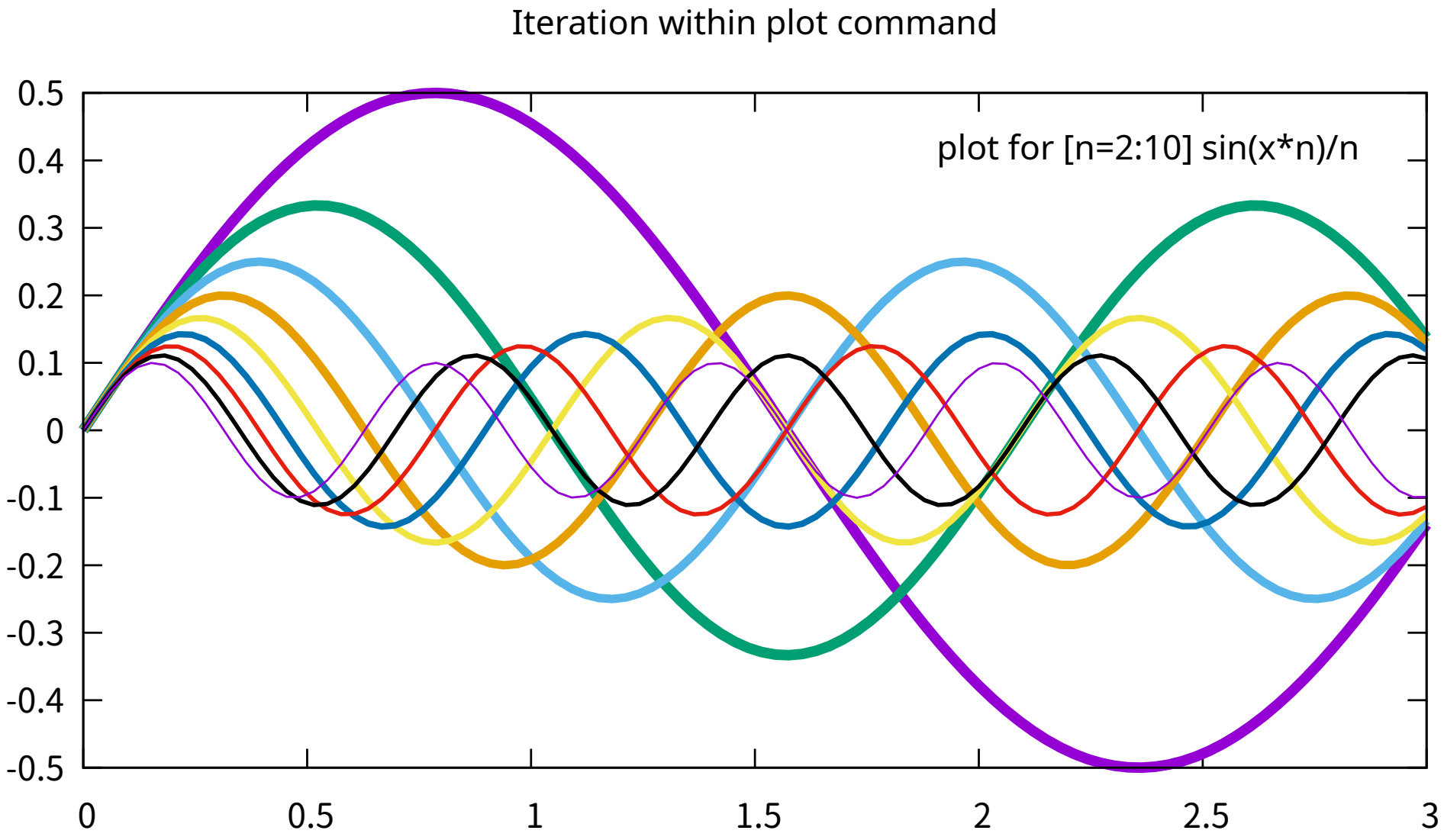
Hypertext is shown when the mouse is over a point



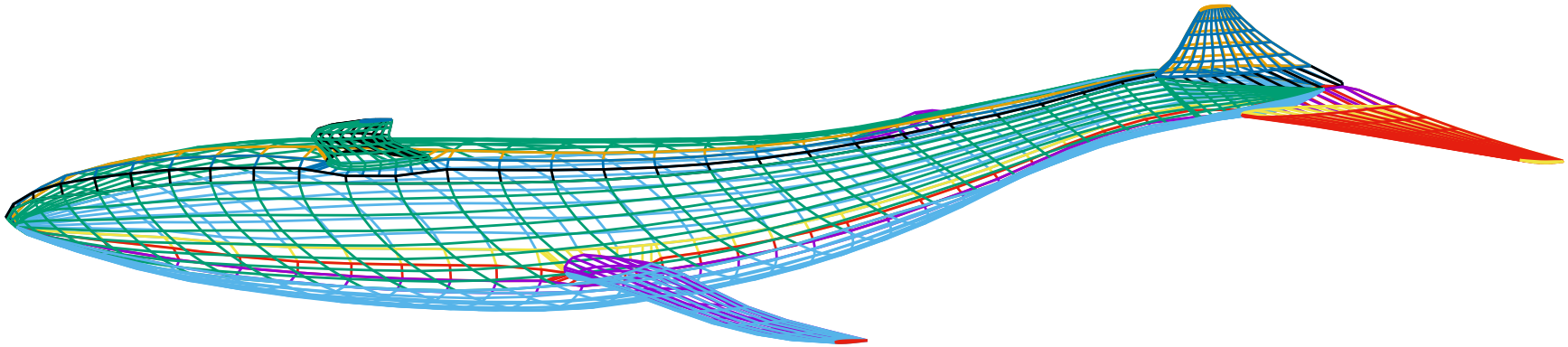

variable color and orientation in plotstyle 'with labels'
one
two
three
four
five
six
seven
eight
nine
ten
eleven
twelve
1
2
3
4
5
6
7
8
9
10
11
12


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

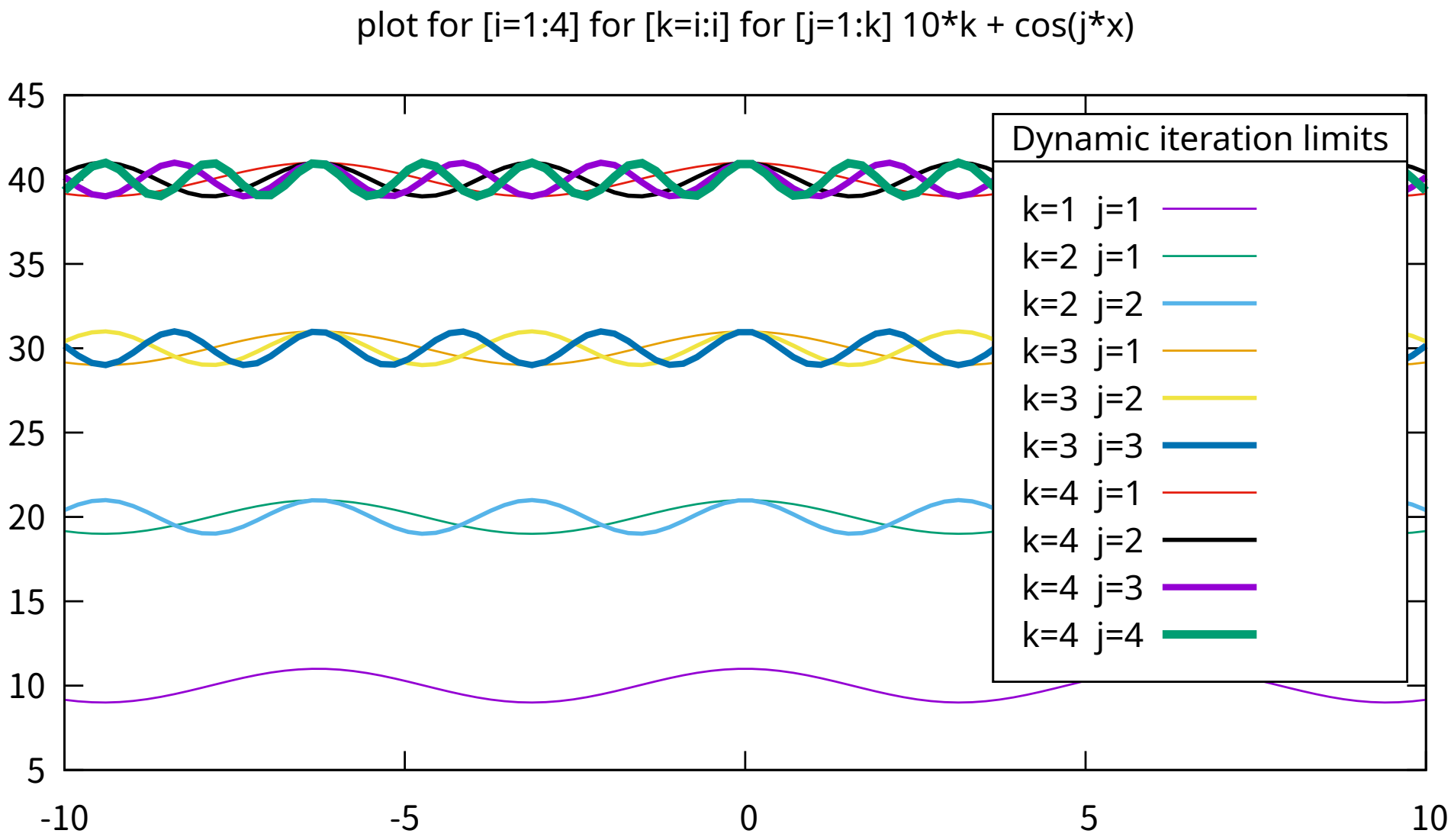

Iteration within plot command
plot for [n=2:10] sin(x*n)/n
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
0
0.5
1
1.5
2
2.5
3

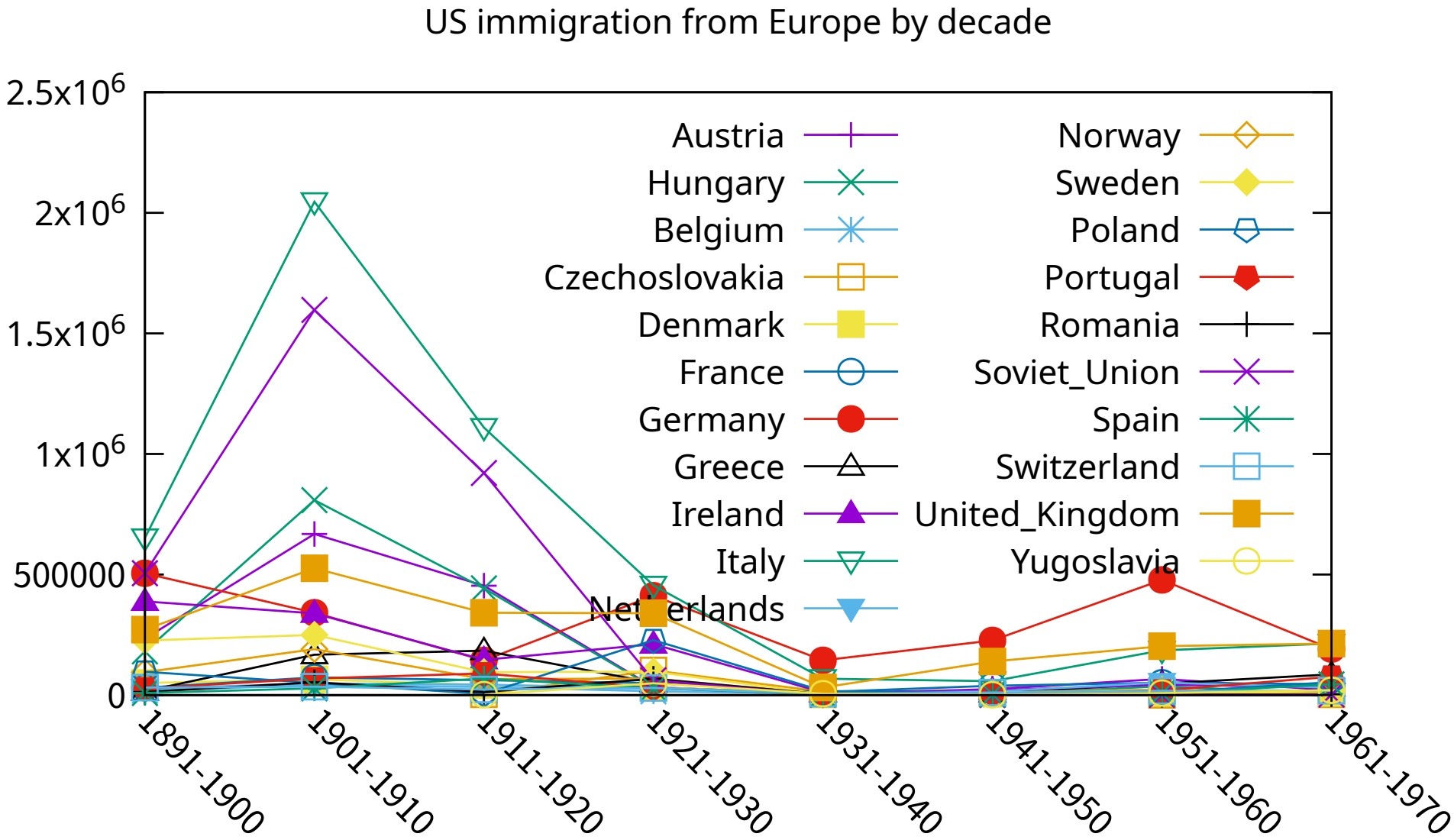
Iteration over all available data in a file

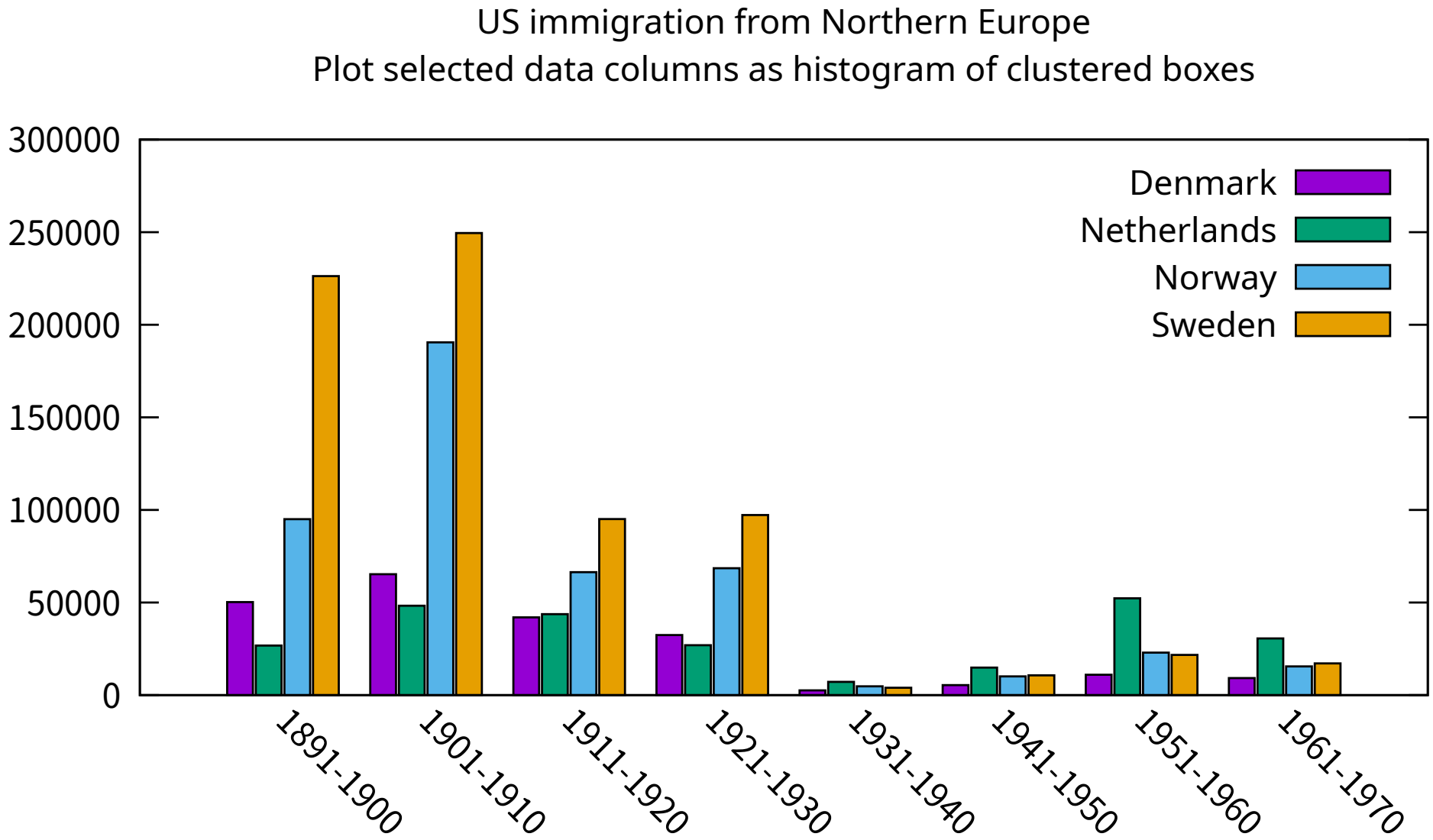


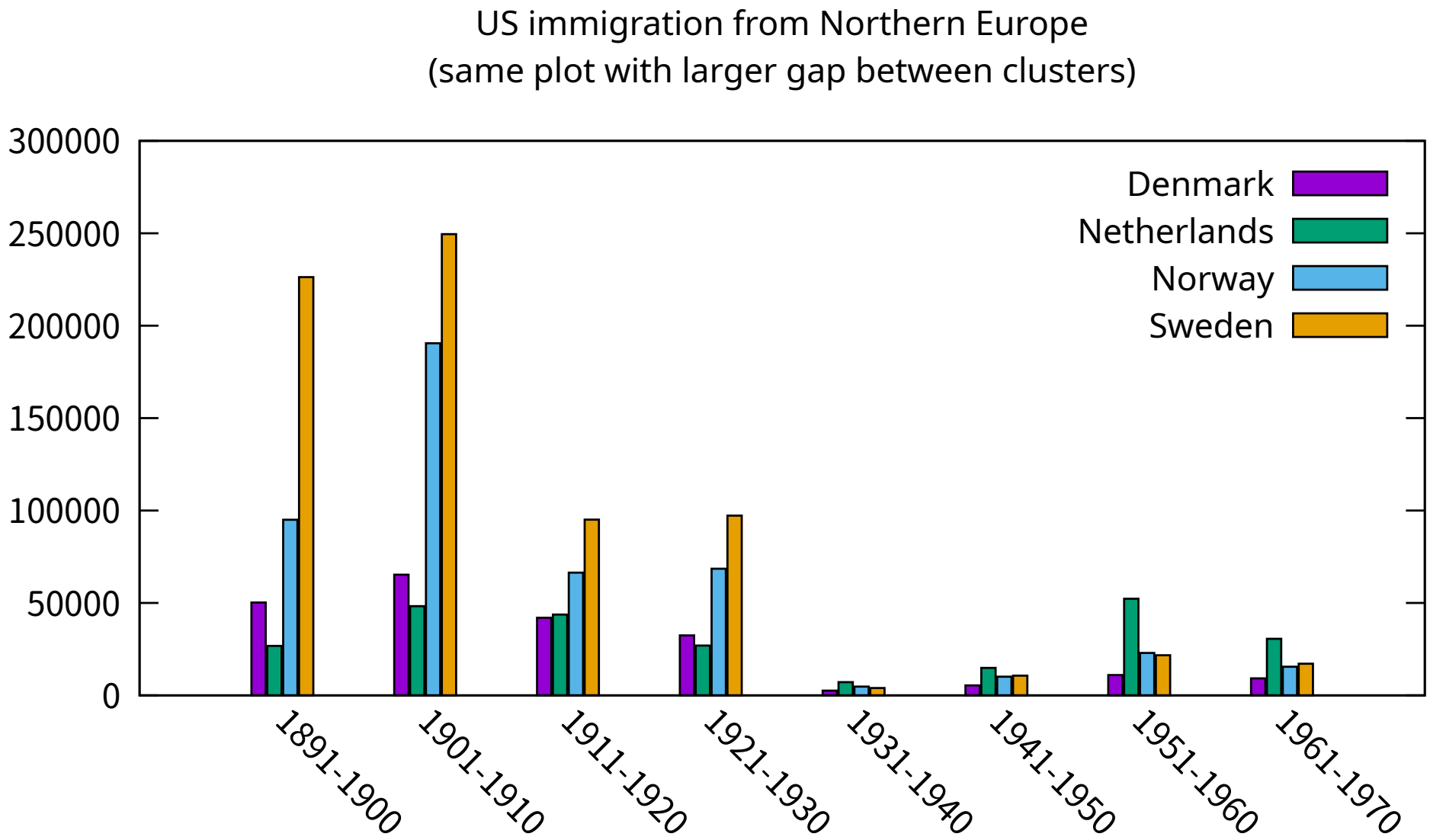
splot for [scan=1:*] 'whale.dat' index scan

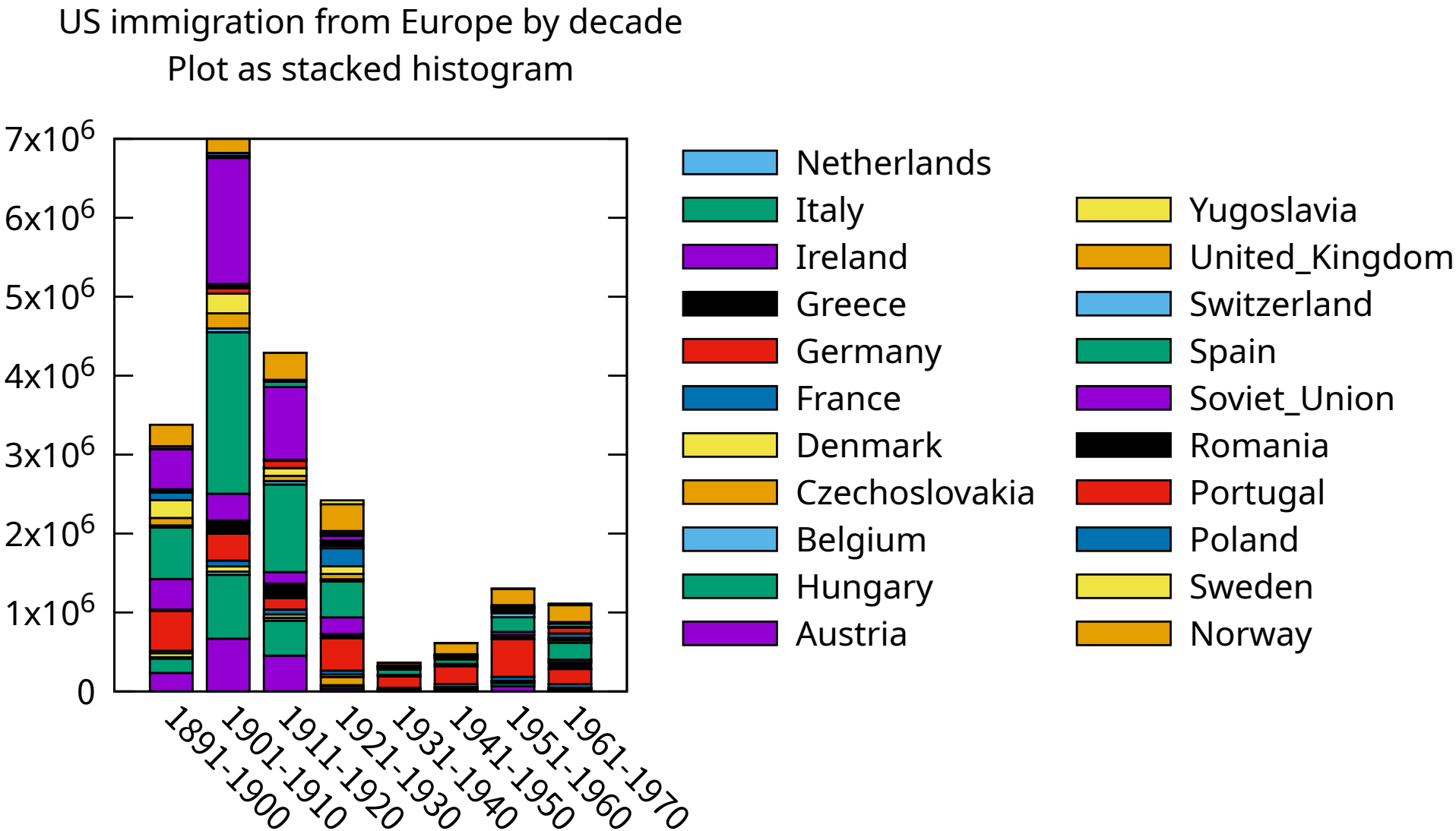
scan 1	scan 6	scan 11	scan 16	scan 21
scan 2	scan 7	scan 12	scan 17	scan 22
scan 3	scan 8	scan 13	scan 18	scan 23
scan 4	scan 9	scan 14	scan 19	
scan 5	scan 10	scan 15	scan 20	

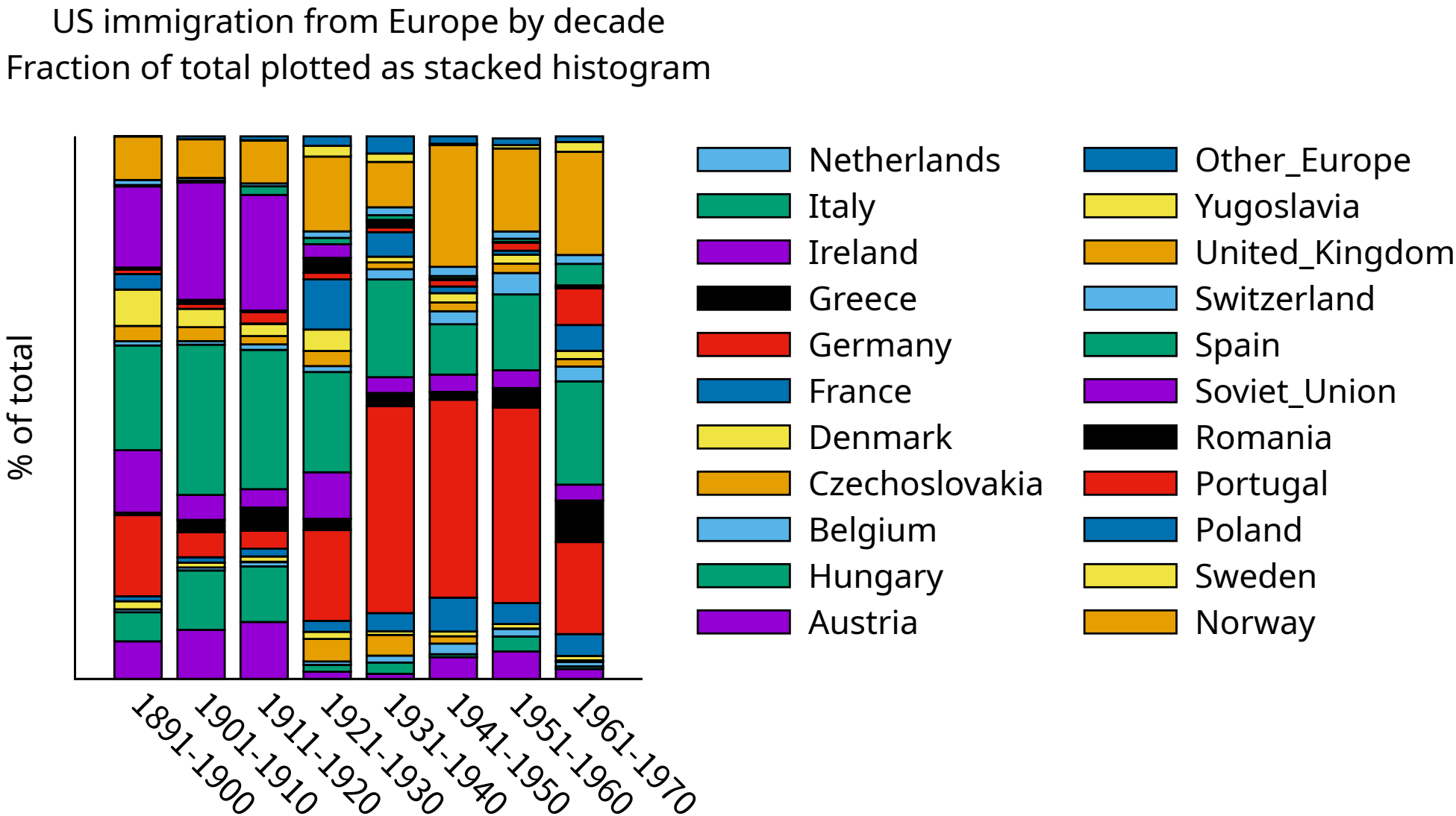

plot for [i=1:4] for [k=i:i] for [j=1:k] 10*k + cos(j*x)
Dynamic iteration limits
k=1 j=1
k=2 j=1
k=2 j=2
k=3 j=1
k=3 j=2
k=3 j=3
k=4 j=1
k=4 j=2
k=4 j=3
k=4 j=4
45
40
35
30
25
20
15
10
5
-10
-5
0
5
10

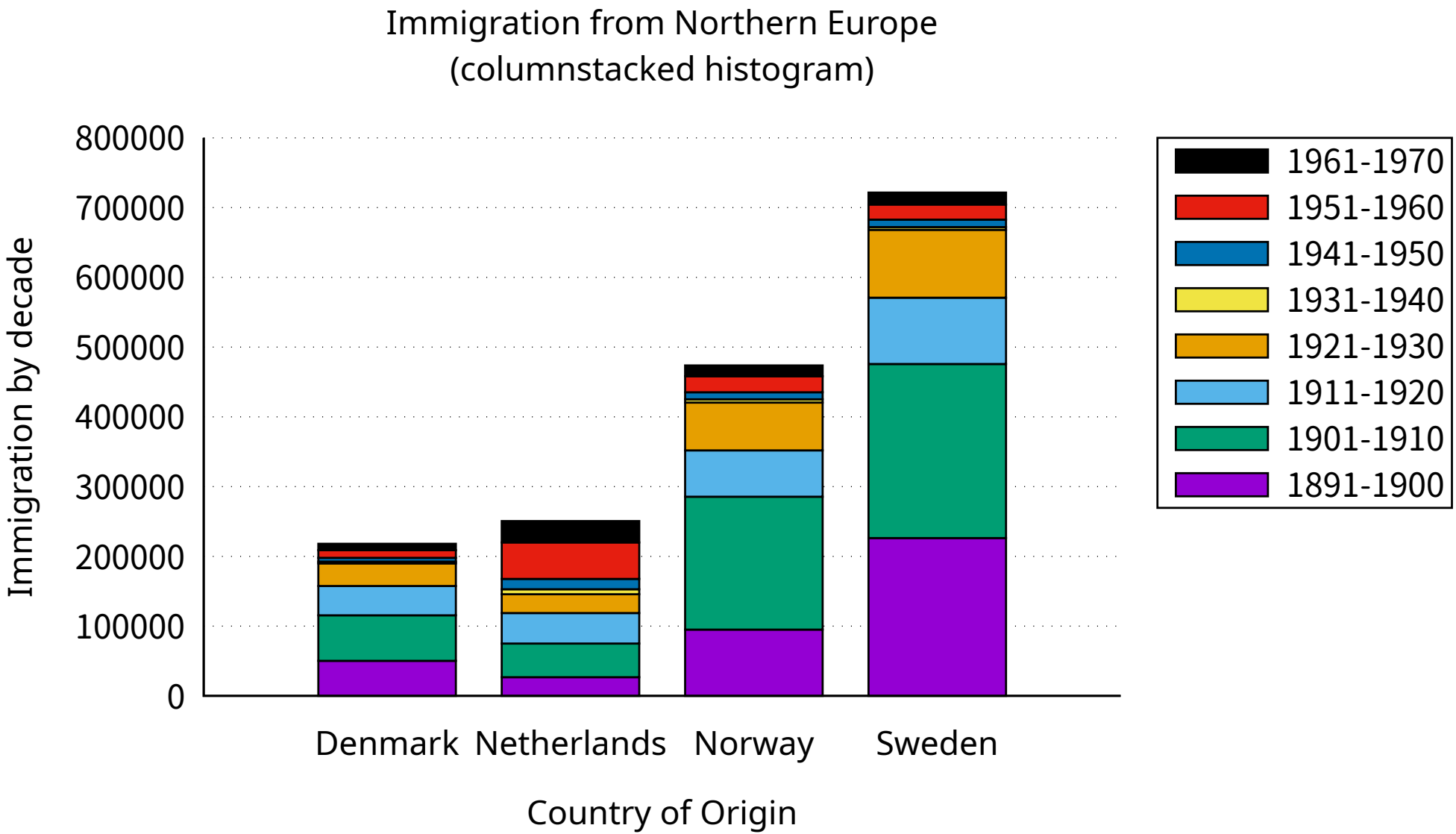

US immigration from Europe by decade
2.5x10^6
2x10^6
1.5x10^6
1x10^6
500000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Austria
Hungary
Belgium
Czechoslovakia
Denmark
France
Germany
Greece
Ireland
Italy
Netherlands
Norway
Sweden
Poland
Portugal
Romania
Soviet_Union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

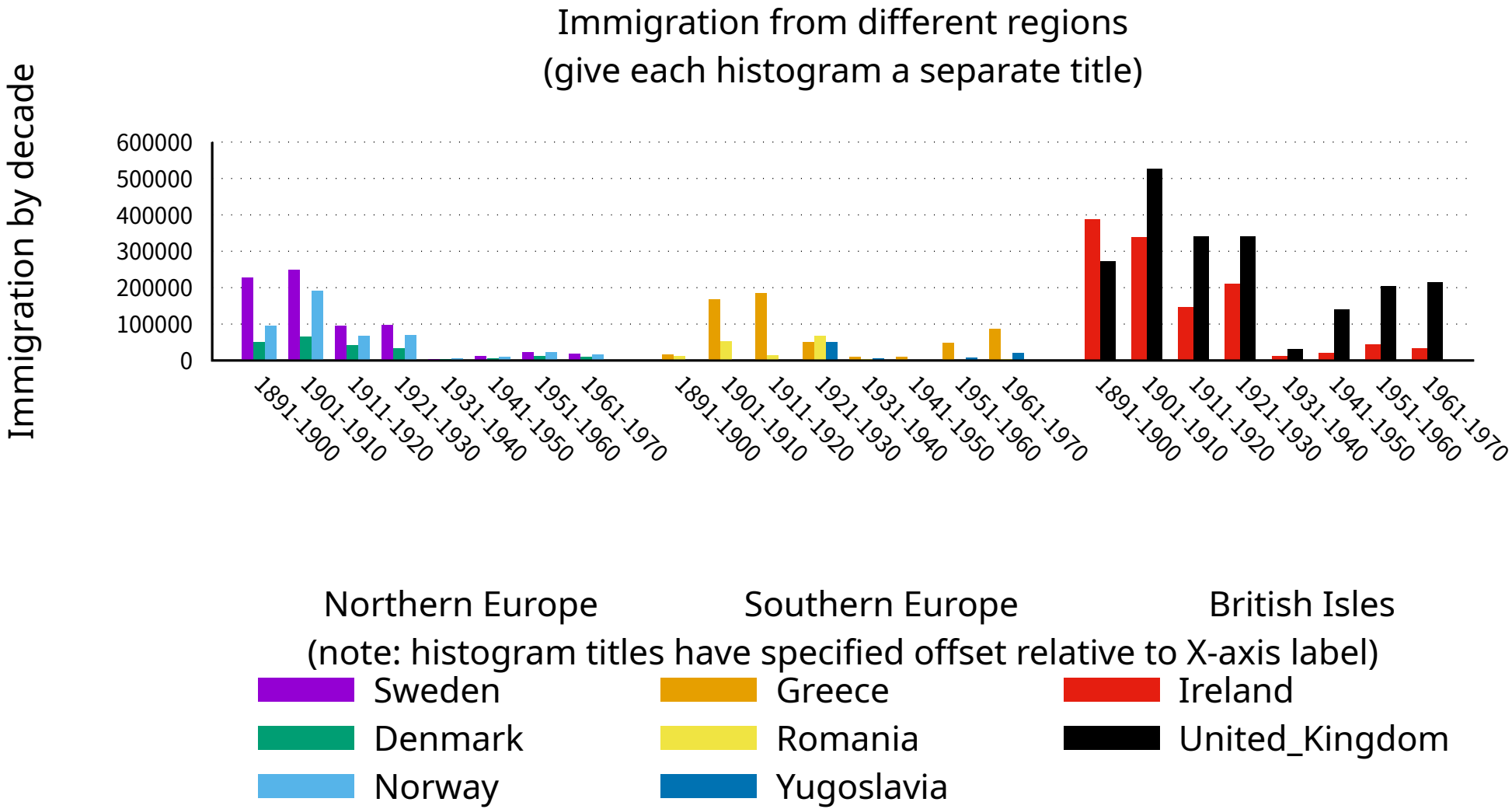

US immigration from Northern Europe
Plot selected data columns as histogram of clustered boxes
300000
250000
200000
150000
100000
50000
0
Denmark
Netherlands
Norway
Sweden
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970

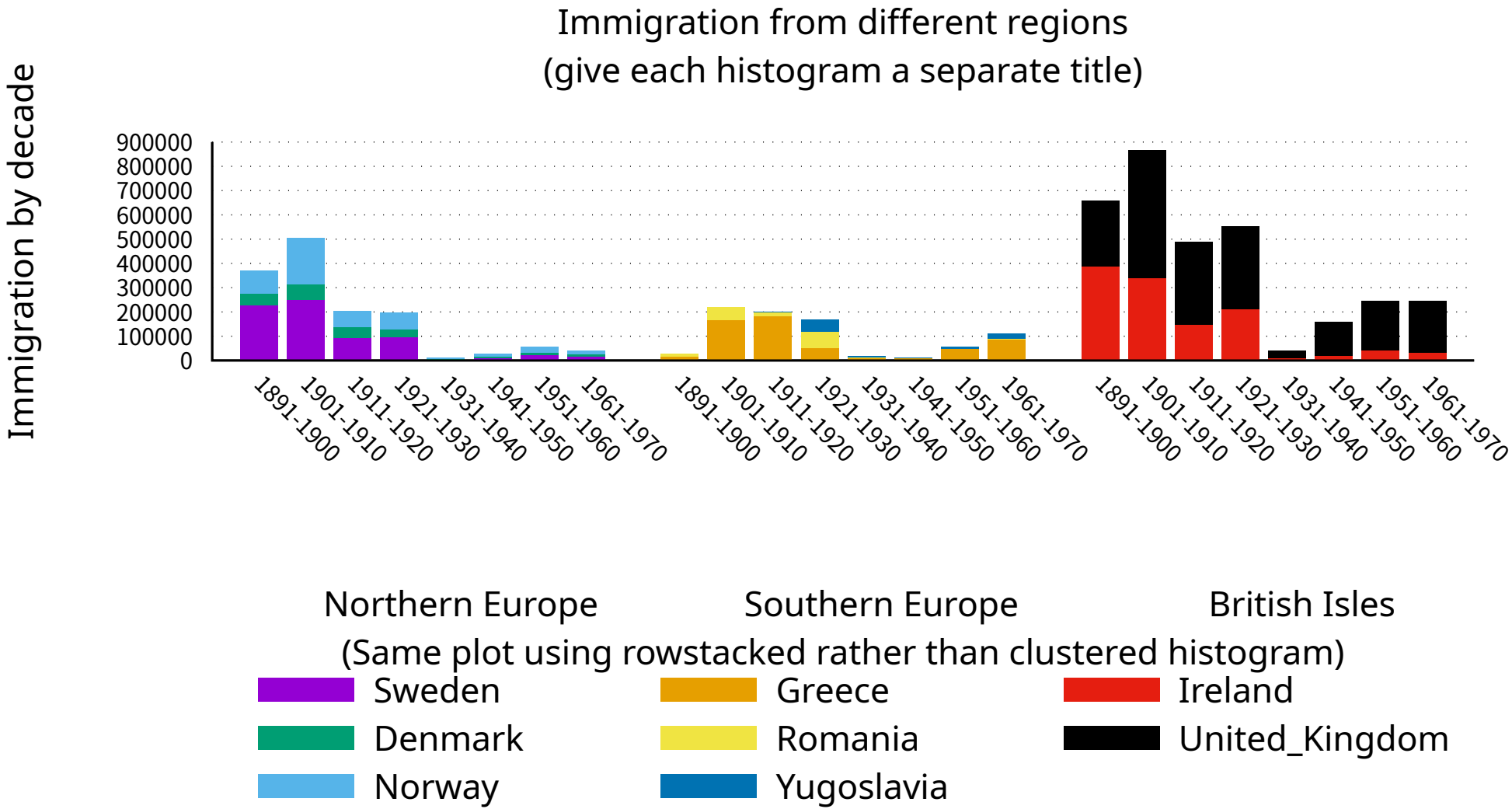

US immigration from Northern Europe
(same plot with larger gap between clusters)
300000
250000
200000
150000
100000
50000
0
Denmark
Netherlands
Norway
Sweden
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970

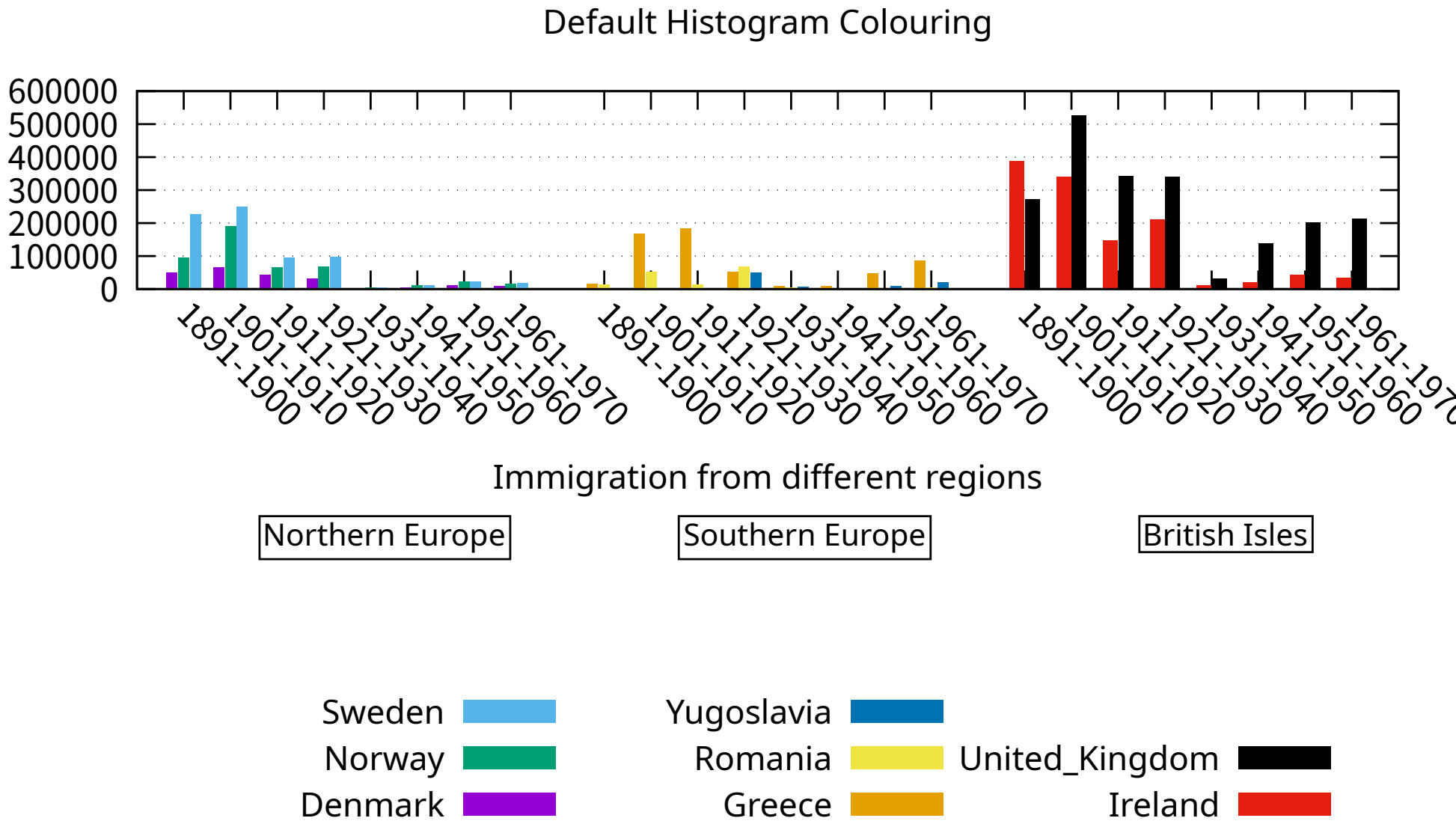

US immigration from Europe by decade
Plot as stacked histogram
7x10^6
6x10^6
5x10^6
4x10^6
3x10^6
2x10^6
1x10^6
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Netherlands
Italy
Ireland
Greece
Germany
France
Denmark
Czechoslovakia
Belgium
Hungary
Austria
Yugoslavia
United_Kingdom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

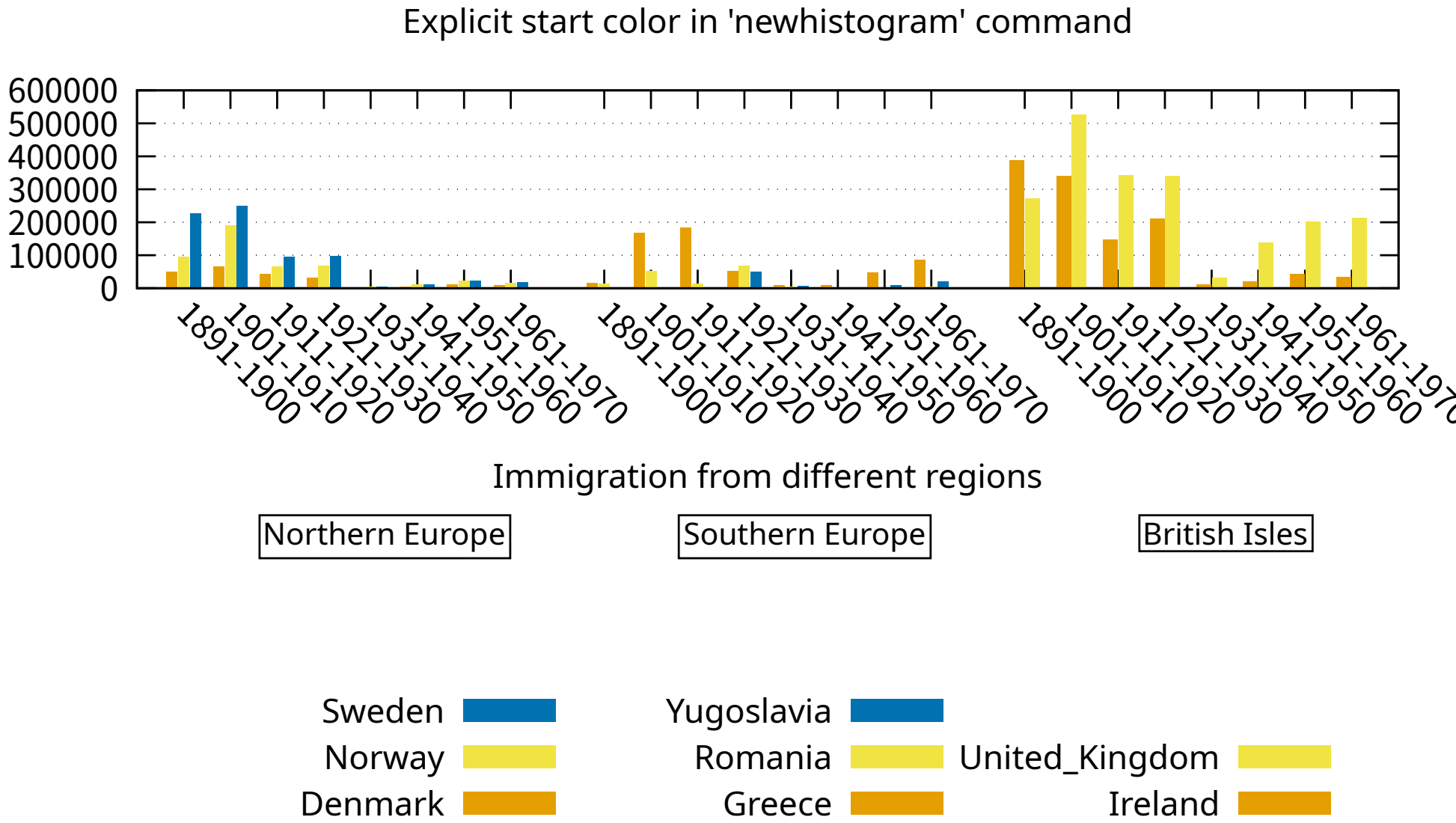

US immigration from Europe by decade
Fraction of total plotted as stacked histogram
% of total
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Netherlands
Italy
Ireland
Greece
Germany
France
Denmark
Czechoslovakia
Belgium
Hungary
Austria
Other_Europe
Yugoslavia
United_Kingdom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

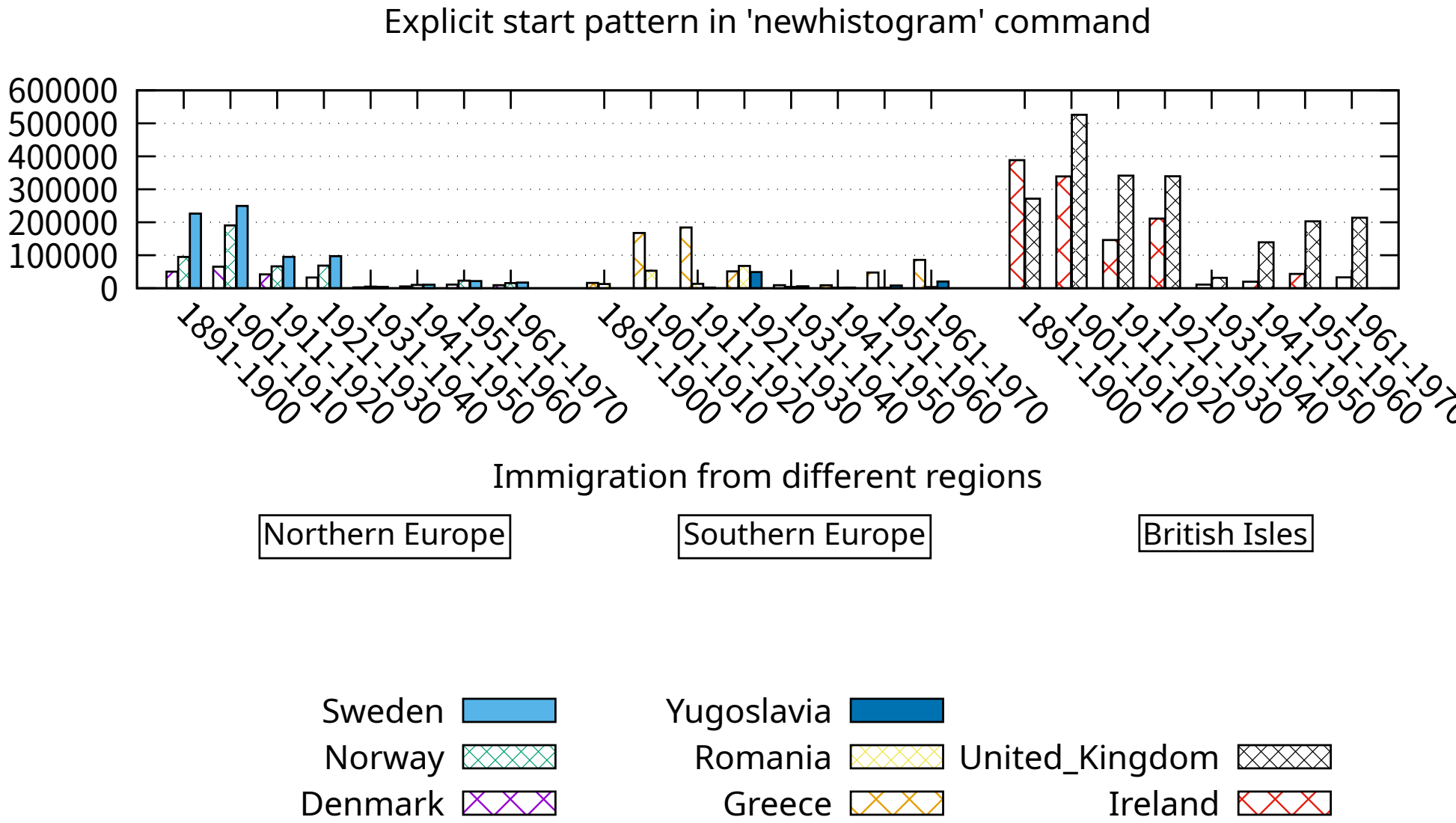

Immigration from Northern Europe
(columnstacked histogram)
Immigration by decade
800000
700000
600000
500000
400000
300000
200000
100000
0
Denmark
Netherlands
Norway
Sweden
Country of Origin
1961-1970
1951-1960
1941-1950
1931-1940
1921-1930
1911-1920
1901-1910
1891-1900

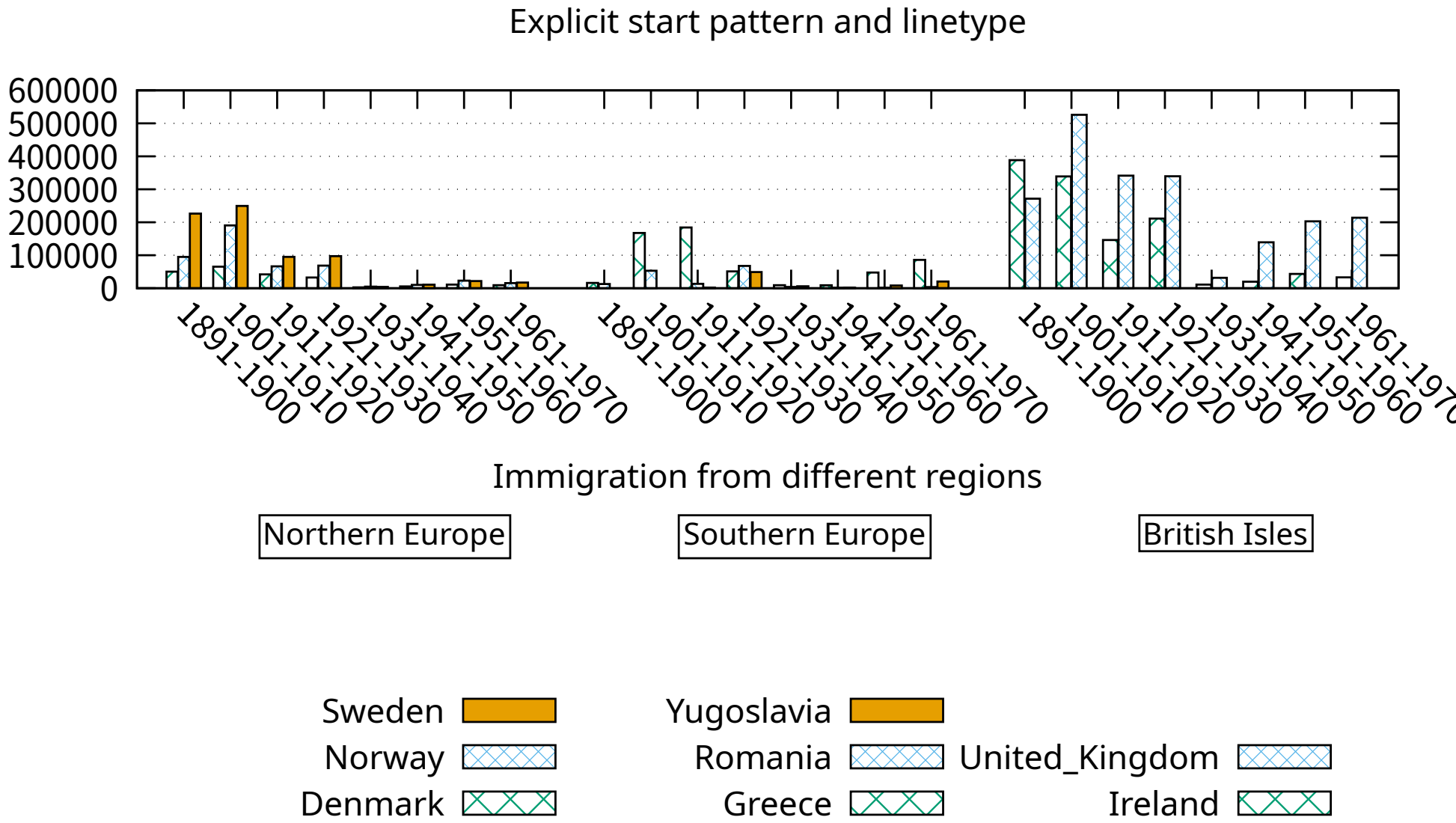

Immigration from different regions
(give each histogram a separate title)
Immigration by decade
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Northern Europe
Southern Europe
British Isles
(note: histogram titles have specified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

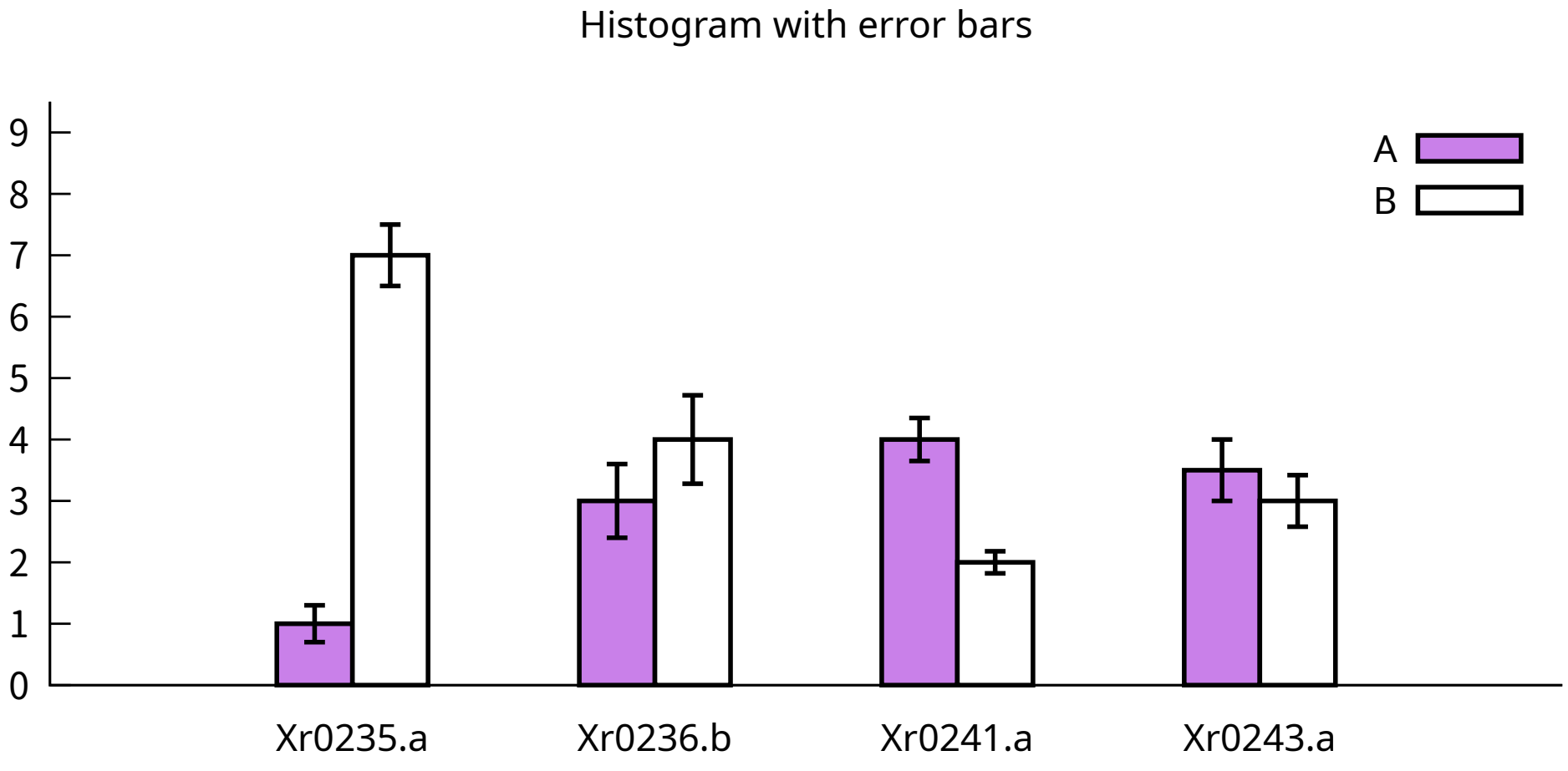

Immigration from different regions
(give each histogram a separate title)
Immigration by decade
900000
800000
700000
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Northern Europe
Southern Europe
British Isles
(Same plot using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

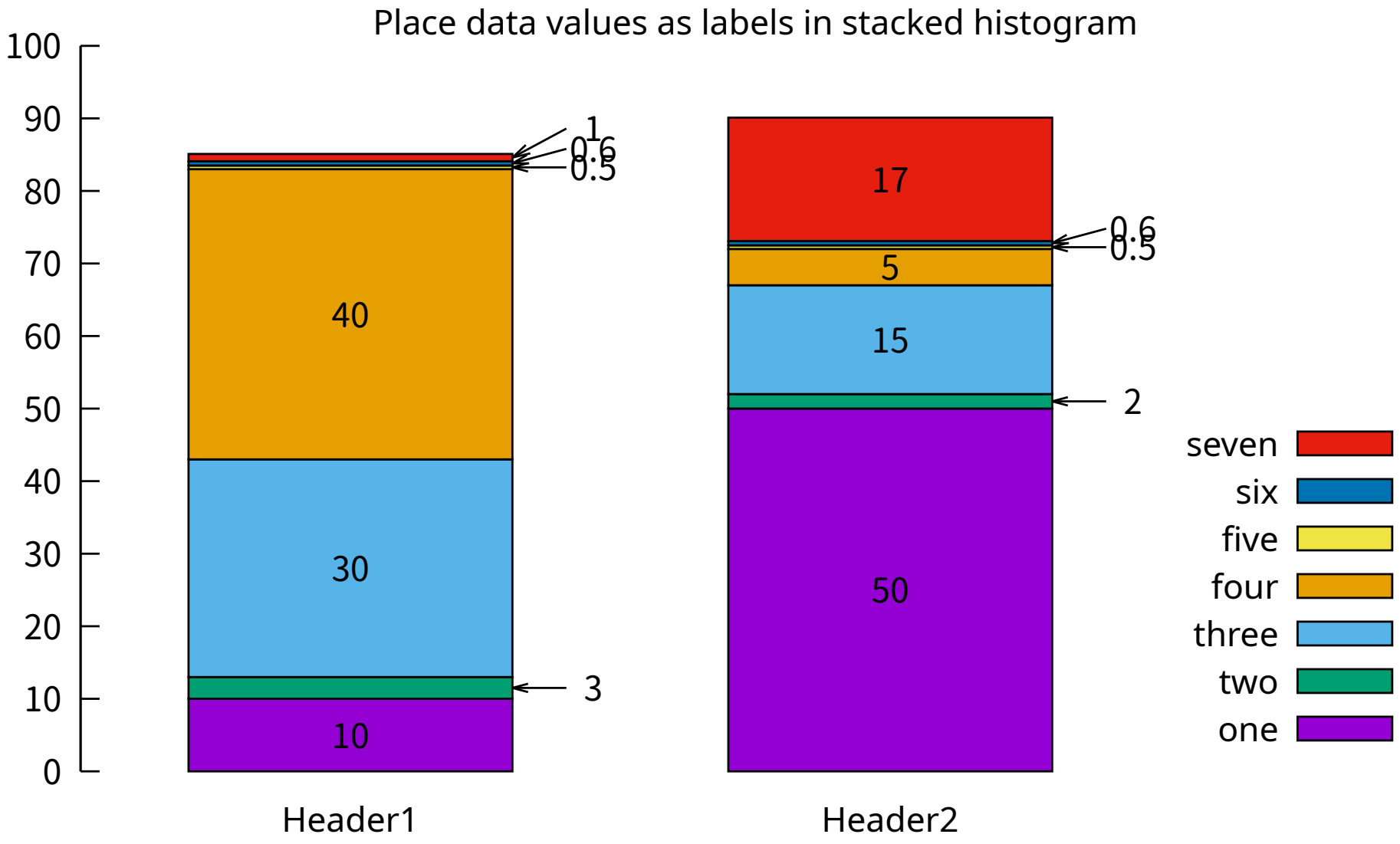

Default Histogram Colouring
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Immigration from different regions
Northern Europe
Southern Europe
British Isles
Sweden
Norway
Denmark
Yugoslavia
Romania
Greece
United_Kingdom
Ireland

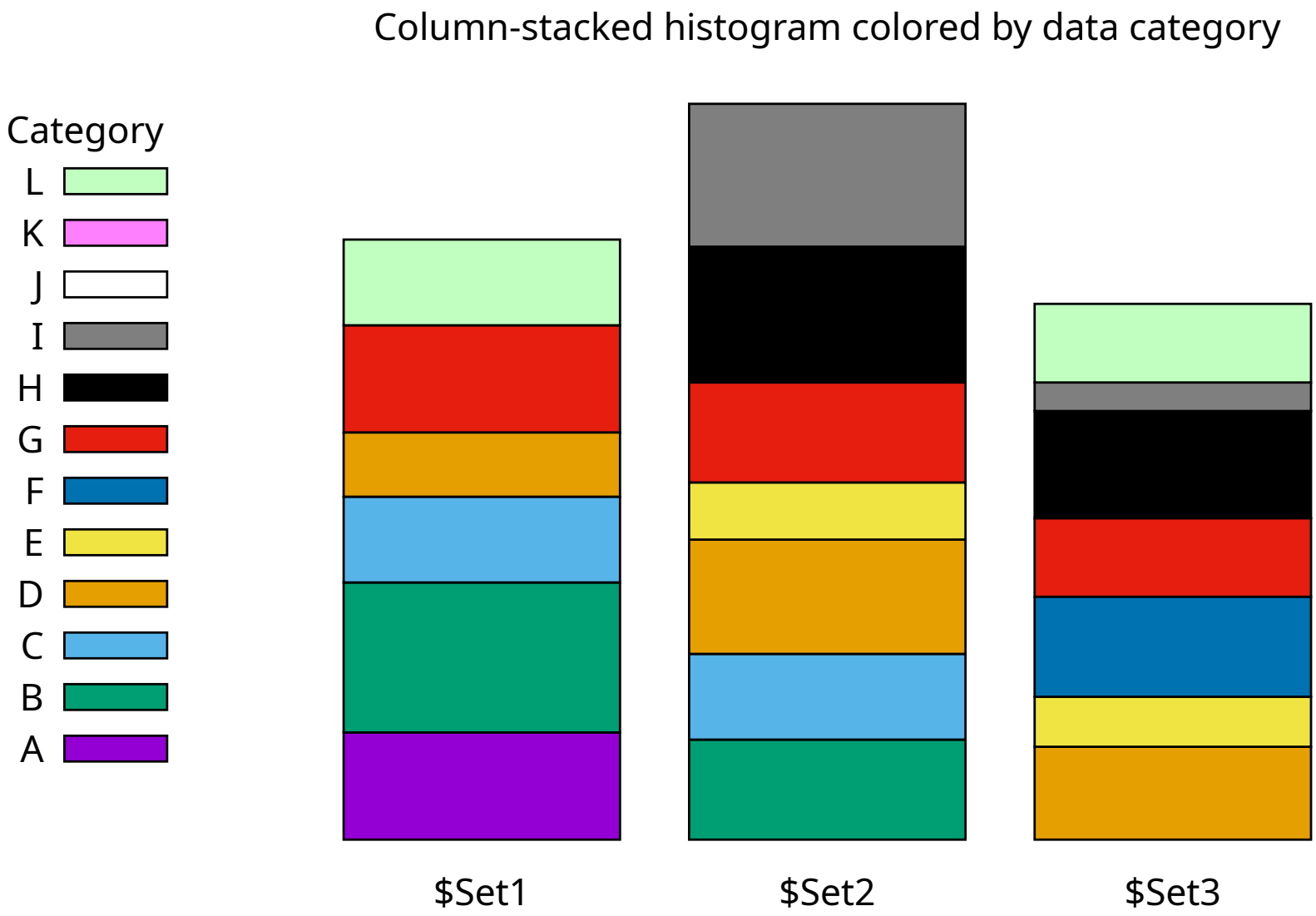

Explicit start color in 'newhistogram' command
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Immigration from different regions
Northern Europe
Southern Europe
British Isles
Sweden
Norway
Denmark
Yugoslavia
Romania
Greece
United_Kingdom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

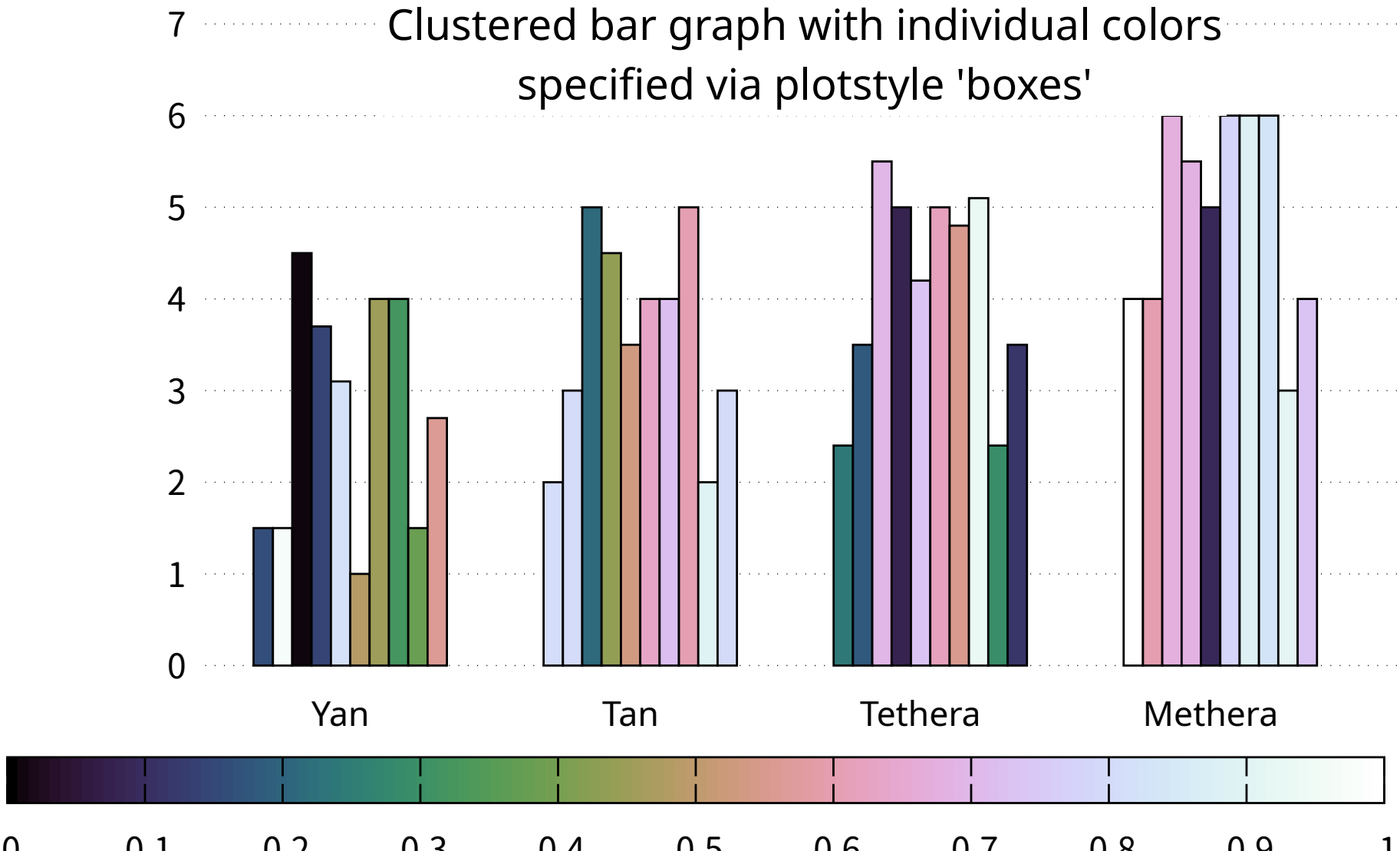

Explicit start pattern in 'newhistogram' command
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Immigration from different regions
Northern Europe
Southern Europe
British Isles
Sweden
Norway
Denmark
Yugoslavia
Romania
Greece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

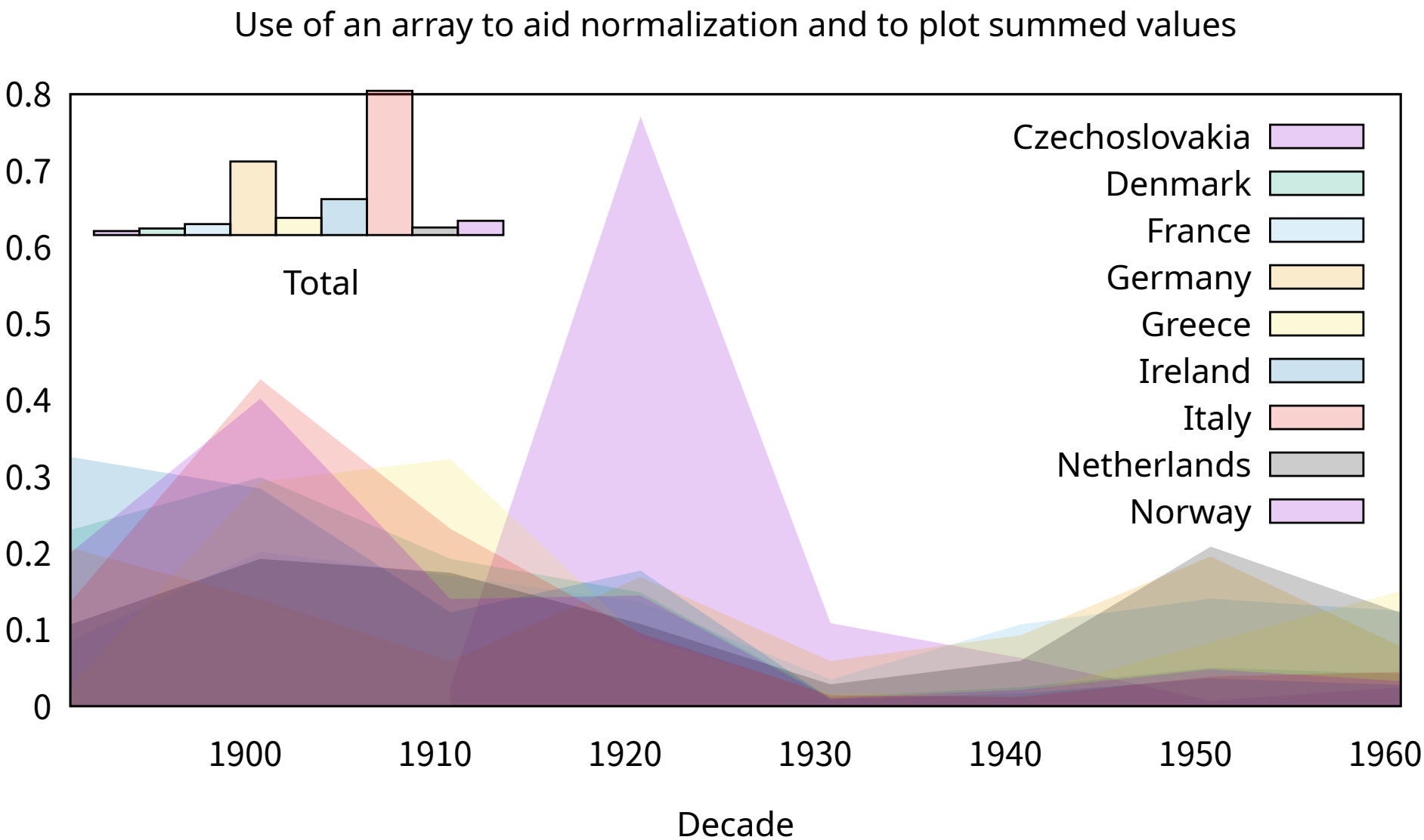

Explicit start pattern and linetype
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Immigration from different regions
Northern Europe
Southern Europe
British Isles
Sweden
Norway
Denmark
Yugoslavia
Romania
Greece
United_Kingdom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

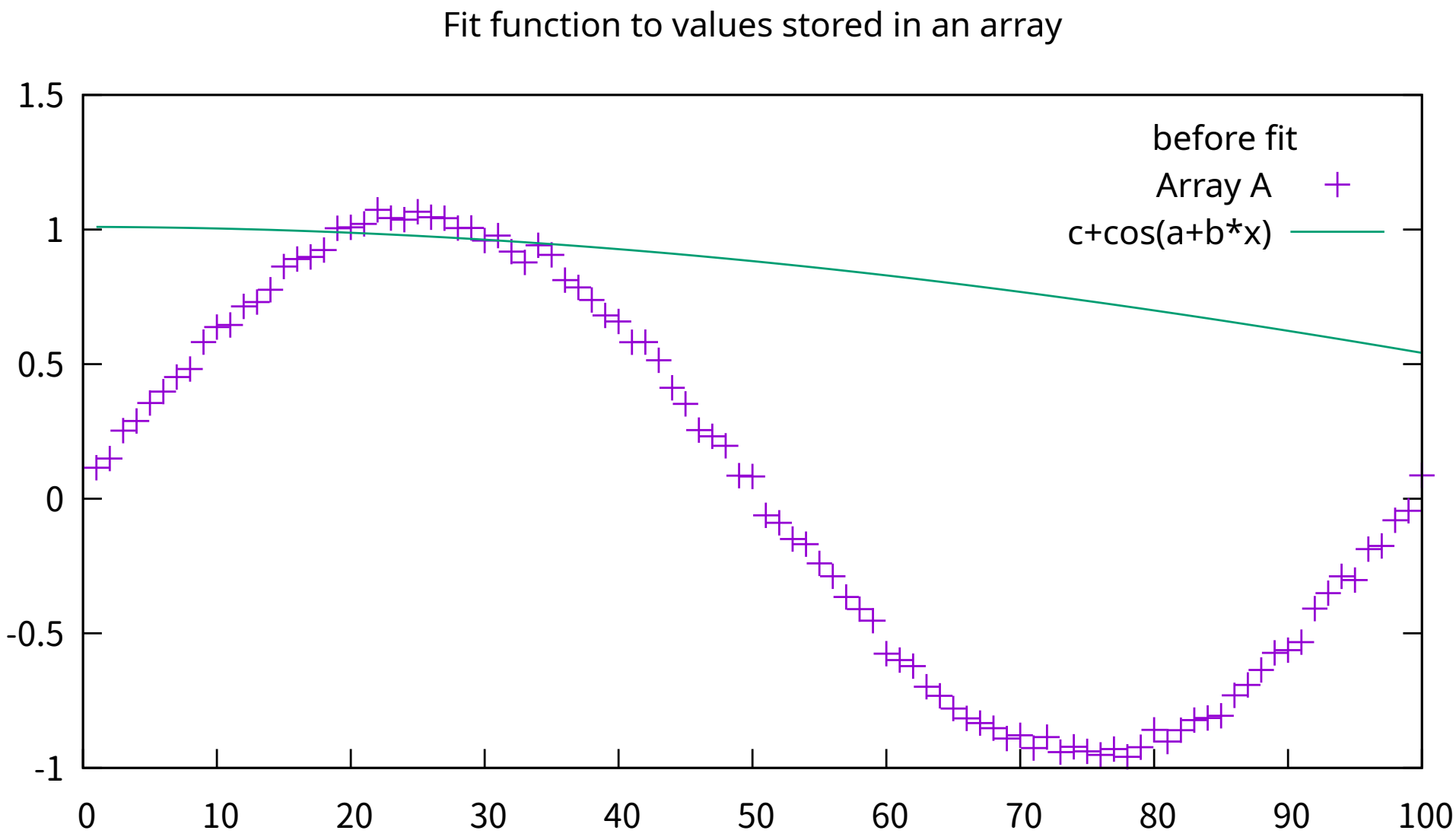

Histogram with error bars
9
8
7
6
5
4
3
2
1
0
A
B
Xr0235.a
Xr0236.b
Xr0241.a
Xr0243.a

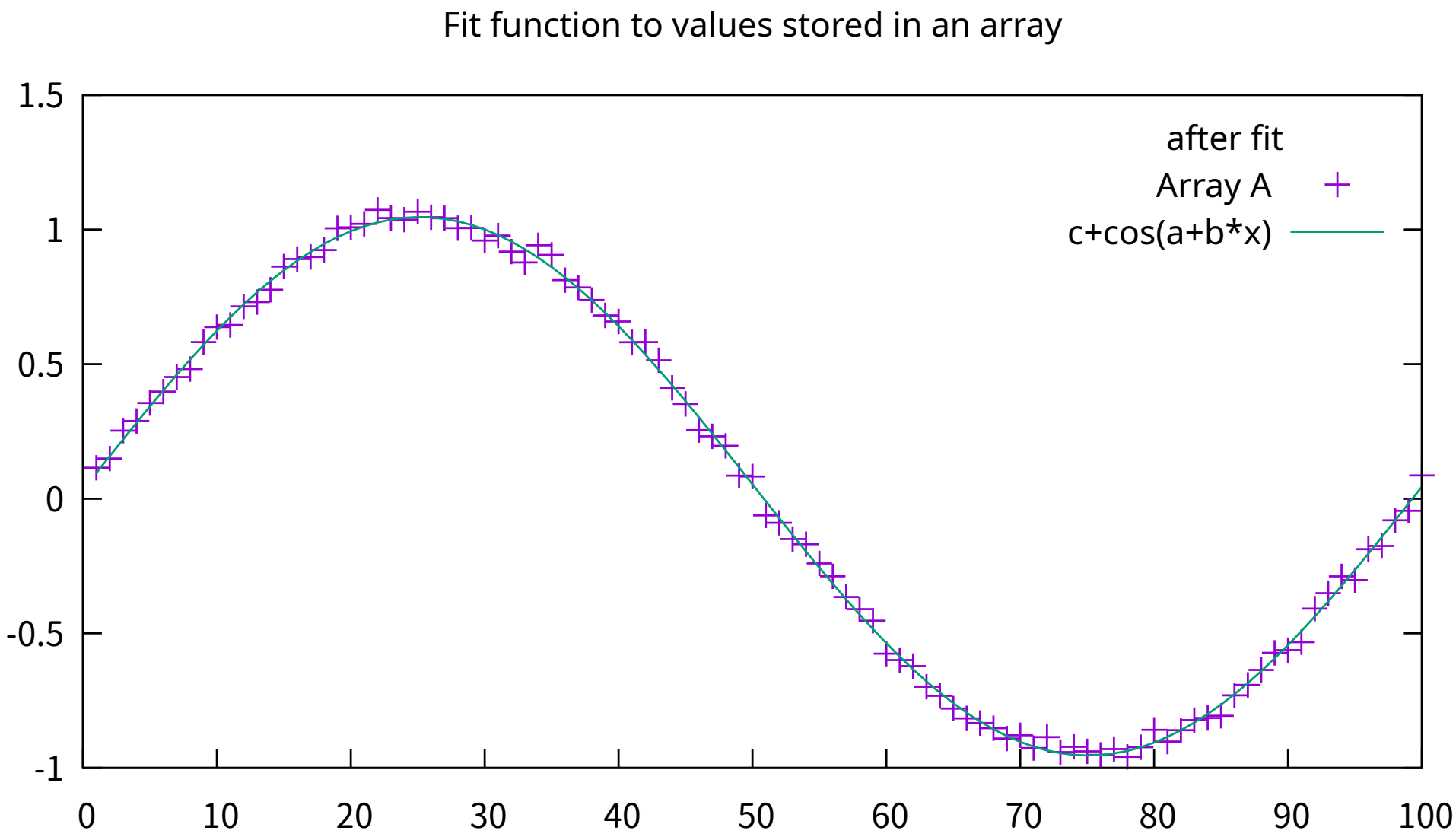

Place data values as labels in stacked histogram
100
90
80
70
60
50
40
30
20
10
0
1
0.6
0.5
40
30
3
10
17
0.6
0.5
5
15
2
50
seven
six
five
four
three
two
one
Header1
Header2

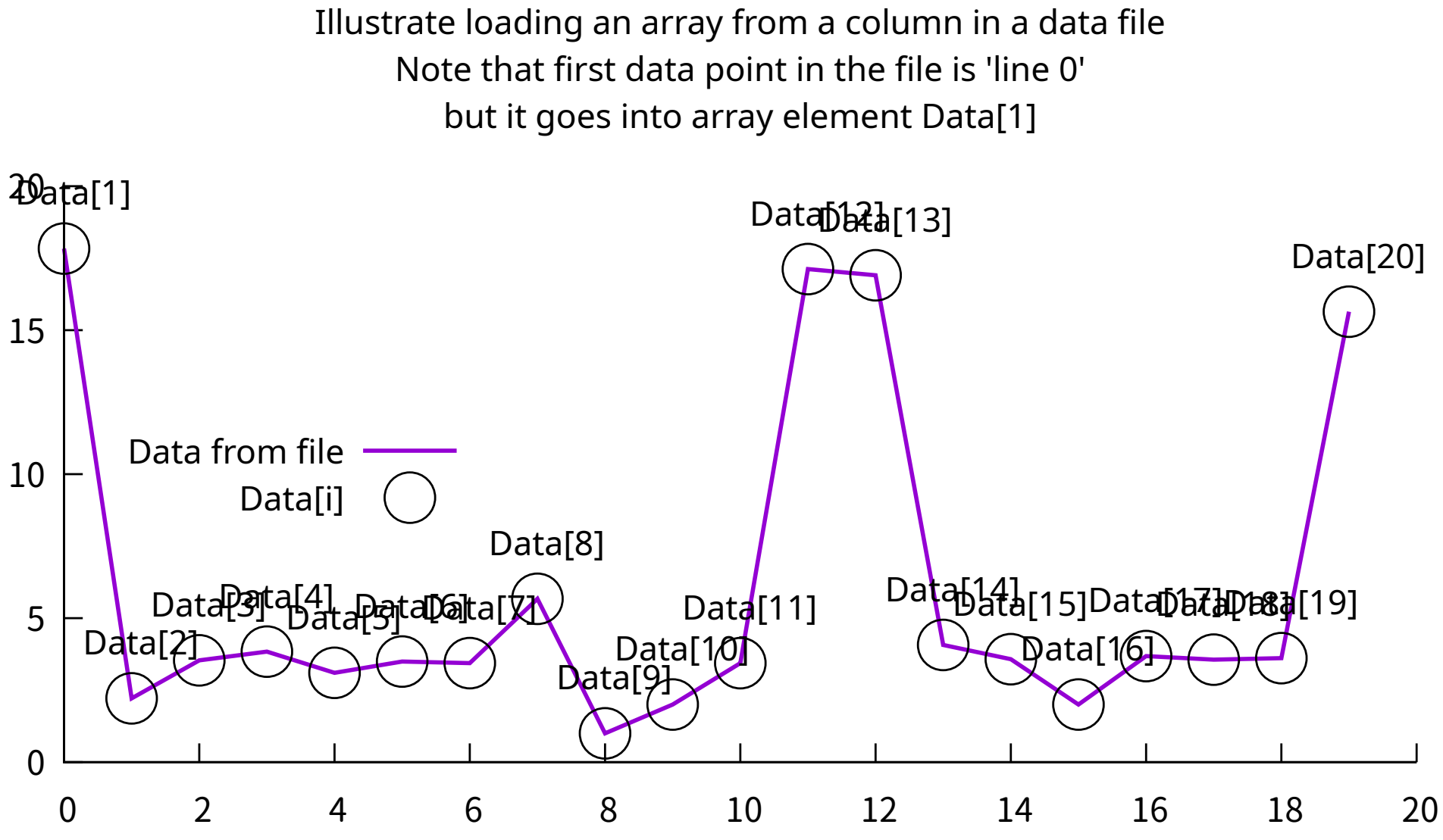

Column-stacked histogram colored by data category
Category
L
K
J
I
H
G
F
E
D
C
B
A
$Set1
$Set2
$Set3


Clustered bar graph with individual colors
specified via plotstyle 'boxes'
7
6
5
4
3
2
1
0
Yan
Tan
Tethera
Methera
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

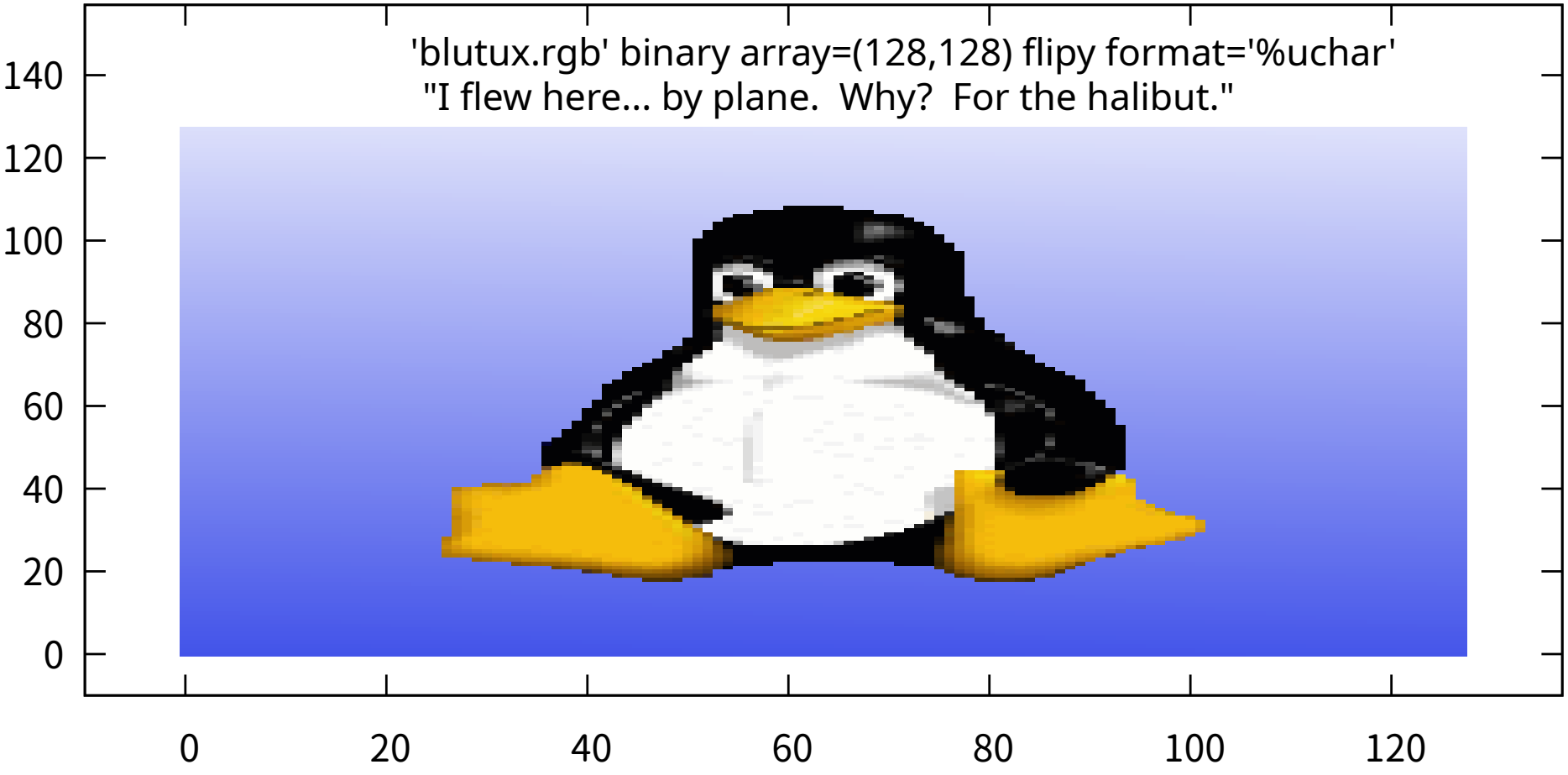

Use of an array to aid normalization and to plot summed values
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Total
Czechoslovakia
Denmark
France
Germany
Greece
Ireland
Italy
Netherlands
Norway
1900
1910
1920
1930
1940
1950
1960
Decade


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

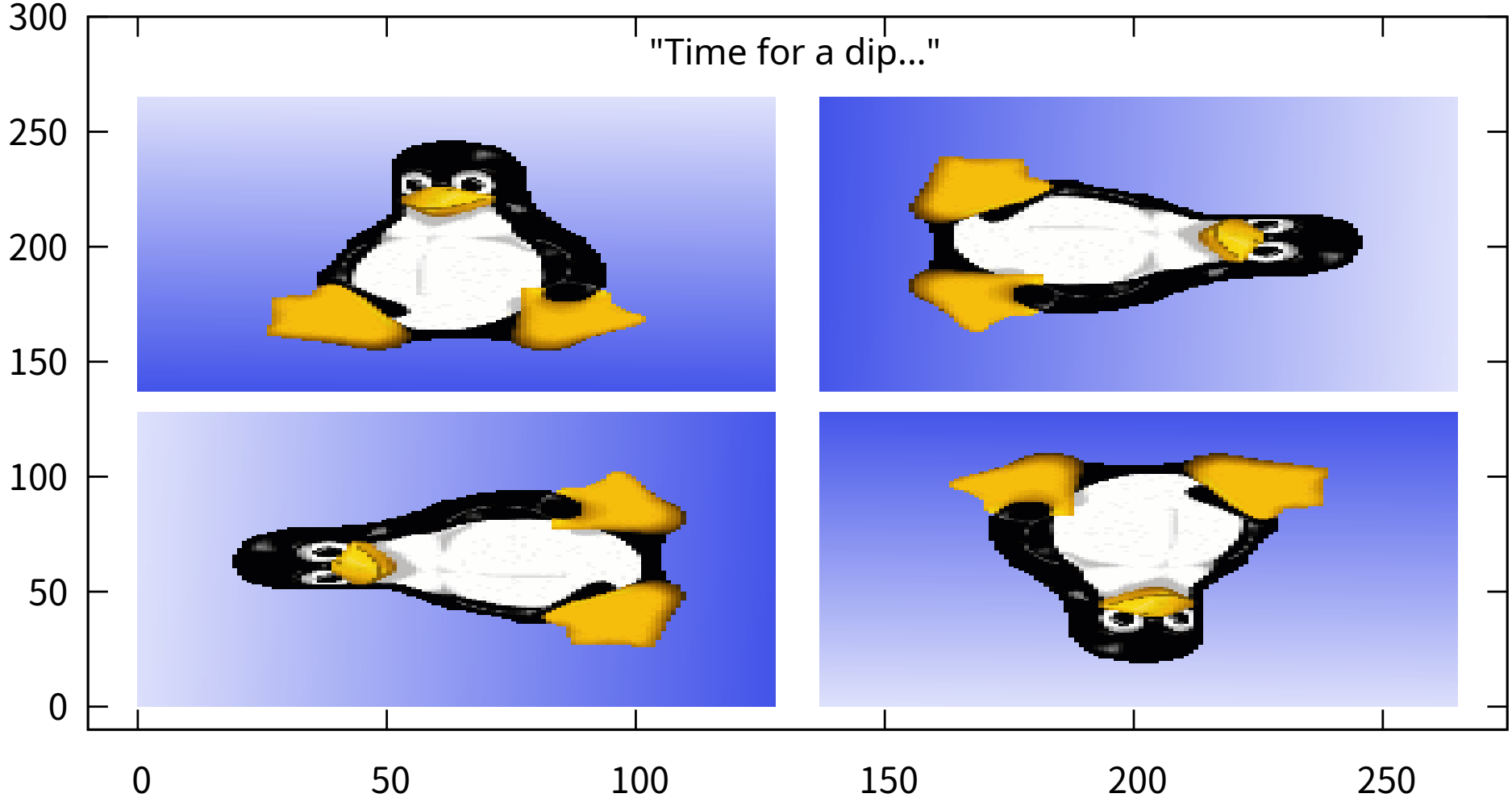

Fit function to values stored in an array
after fit
Array A
c+cos(a+b*x)
1.5
1
0.5
0
-0.5
-1
0
10
20
30
40
50
60
70
80
90
100


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

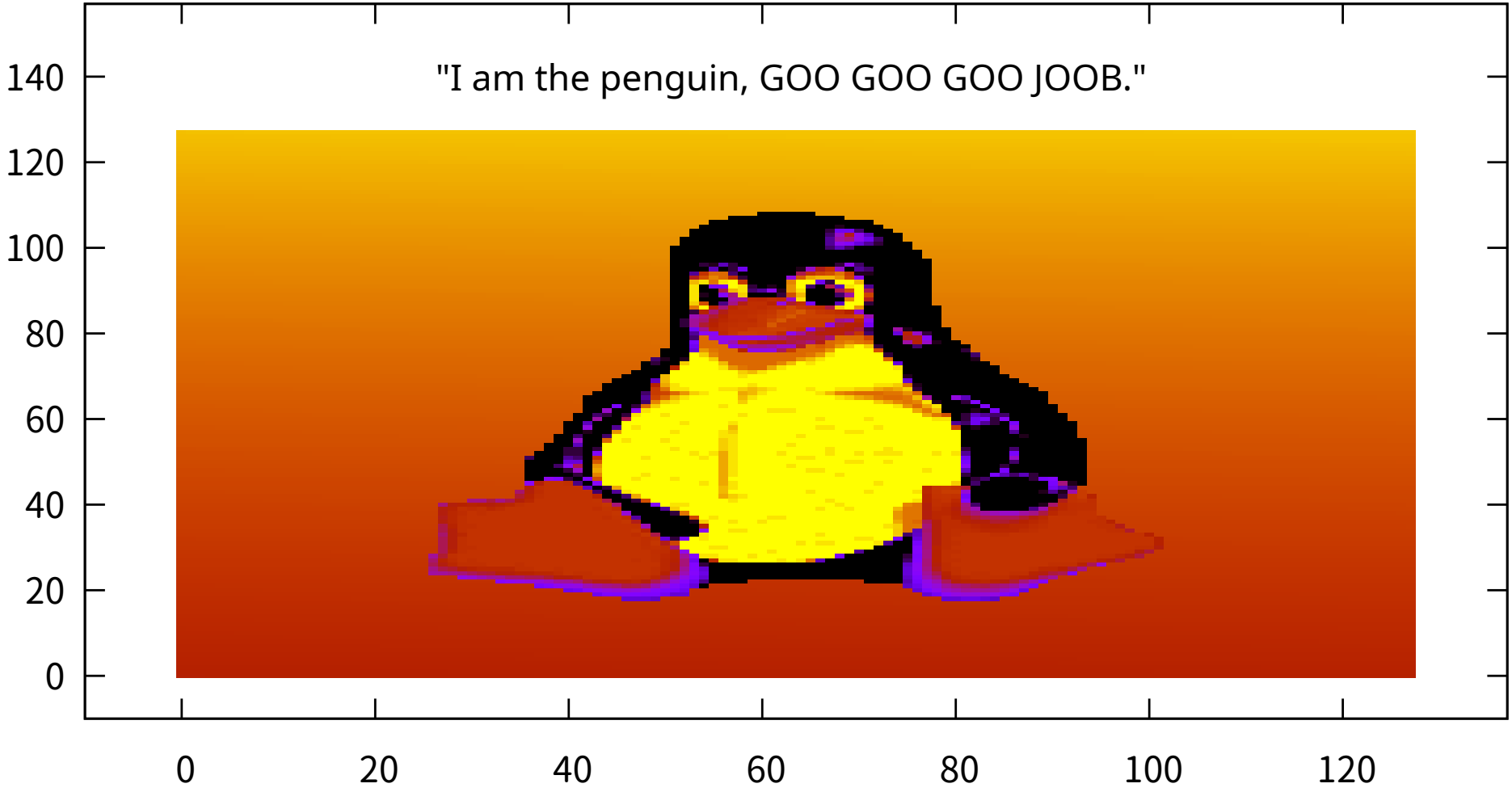
Larry Ewing's GIMP penguin on vacation basking in
the balmy waters off the coast of Murmansk



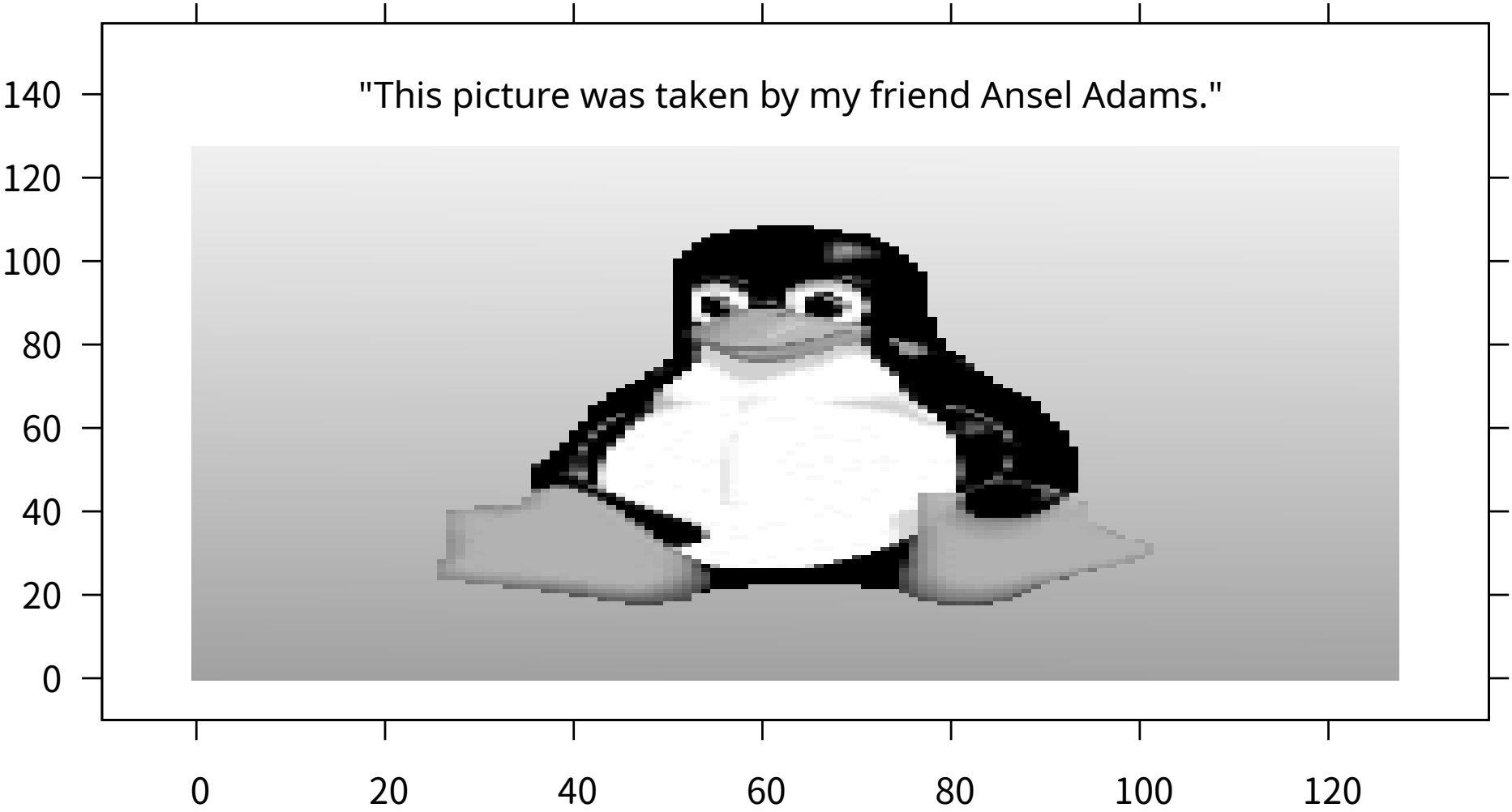
Translations of position variables via 'using'



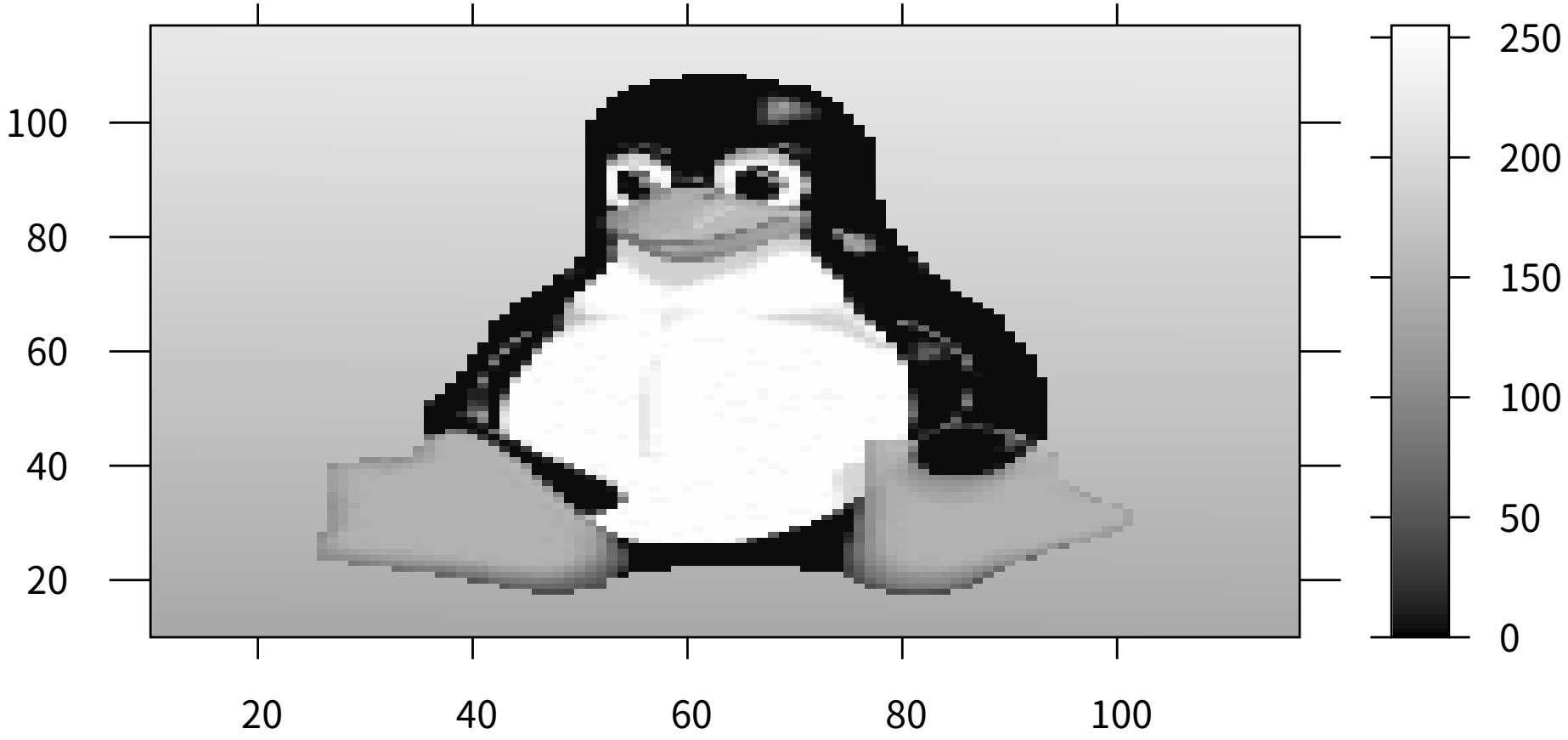
Palette mode 'image' used to produce psychedelic bird



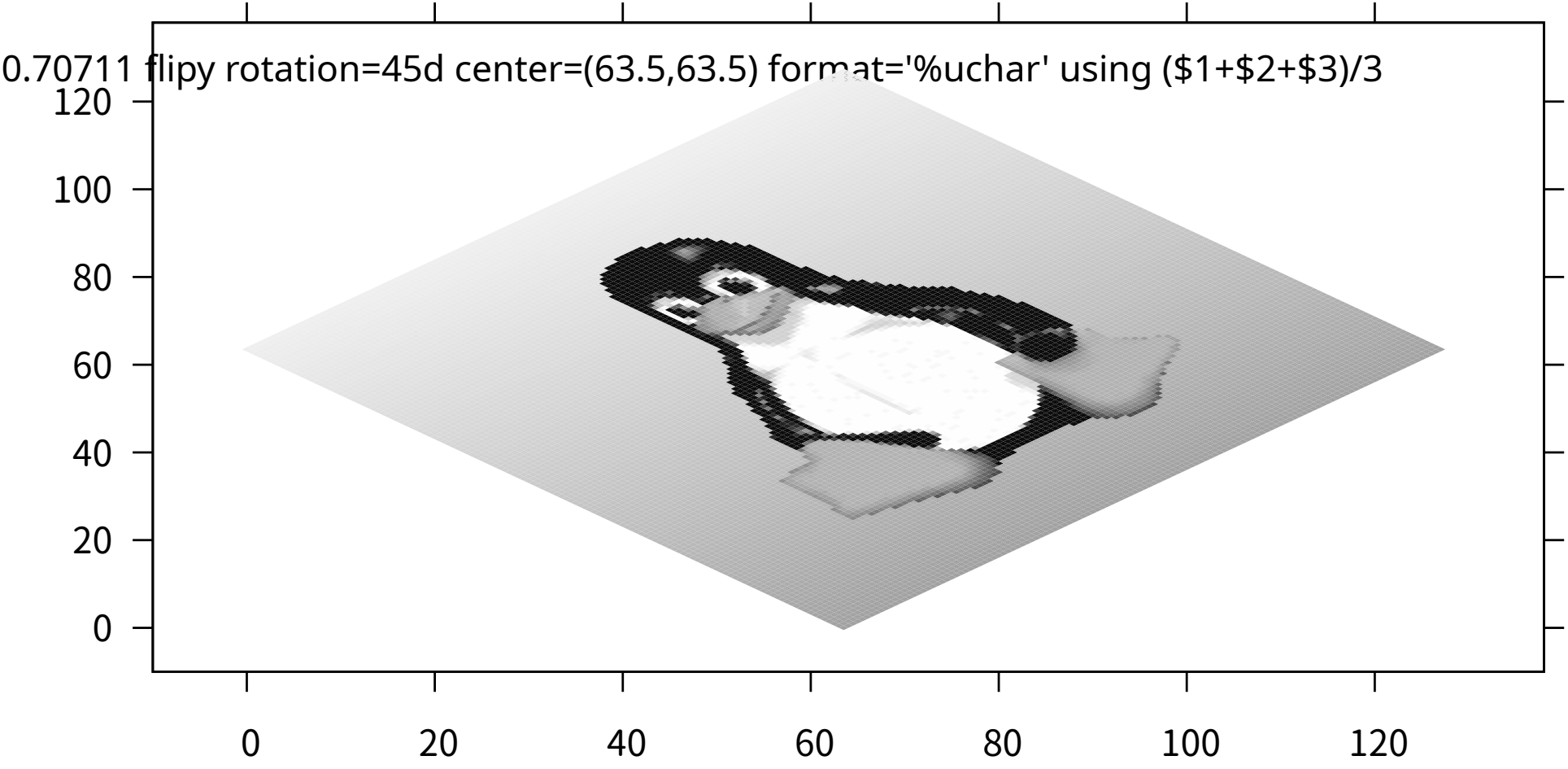
The palette can be changed from color to gray scale



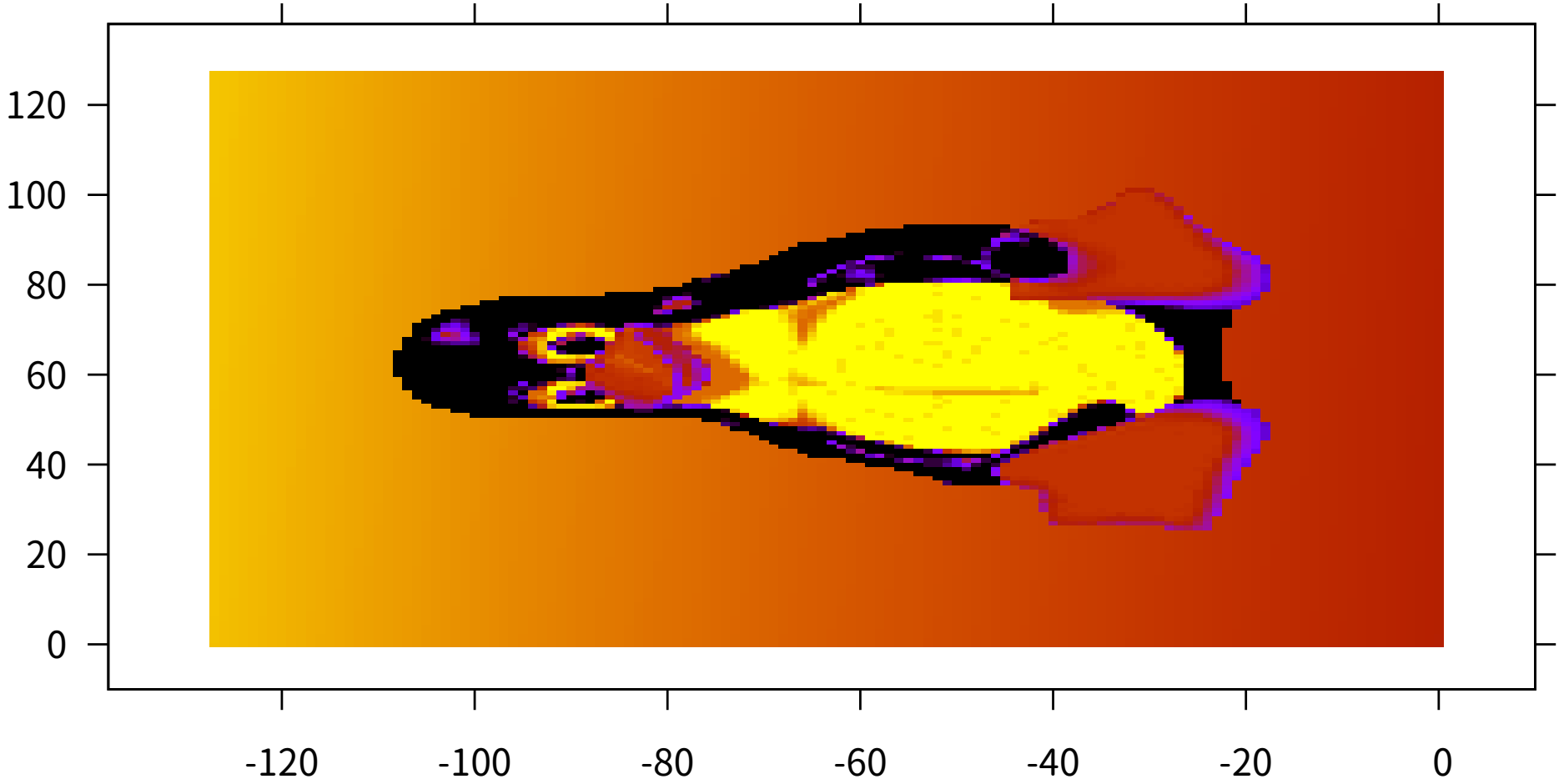
As with 3d color surfaces, a color box may be added to the plot



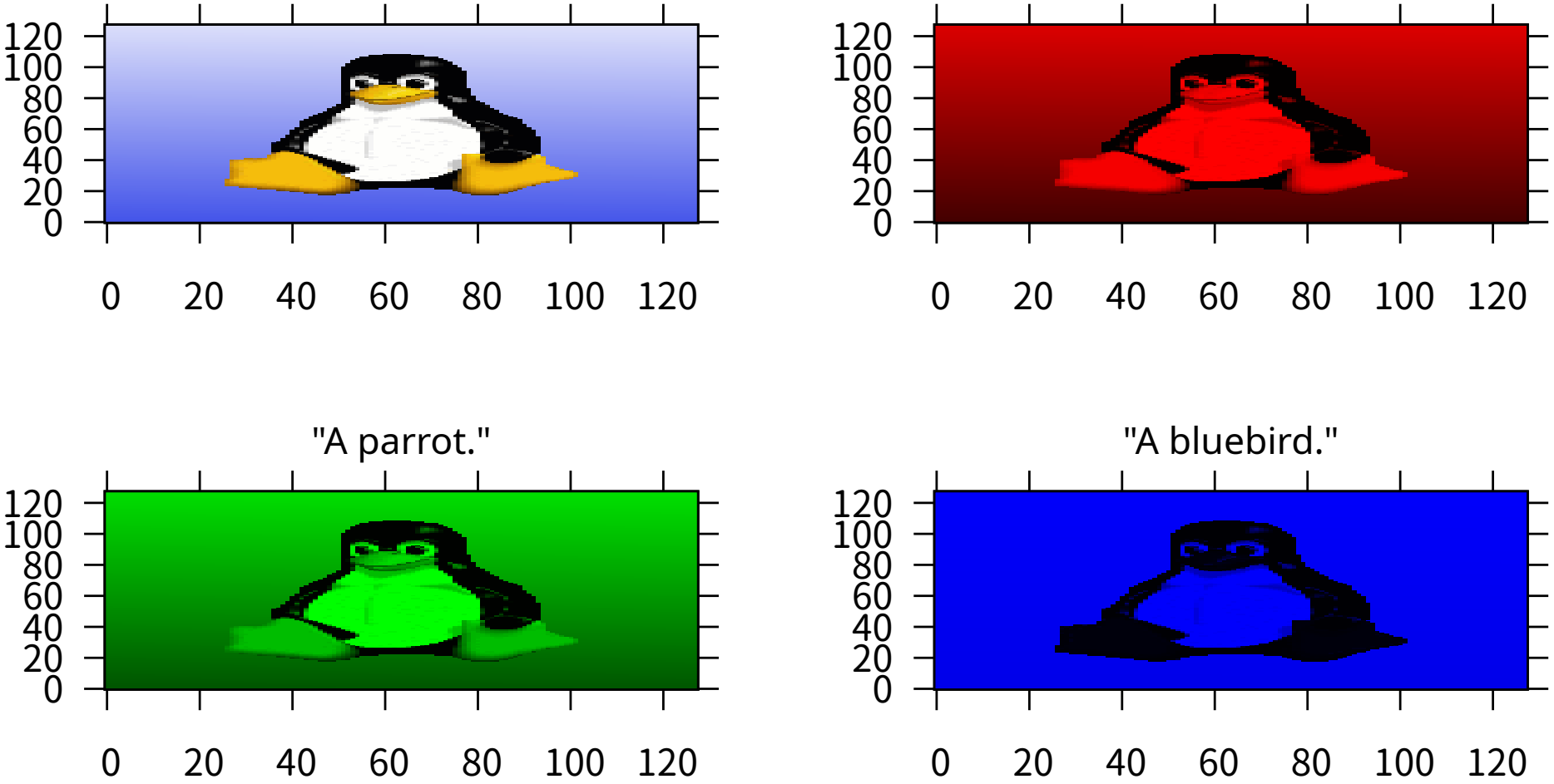
Polygons used to draw pixels for rotated images
Notice the slower refresh rate than for the next plot

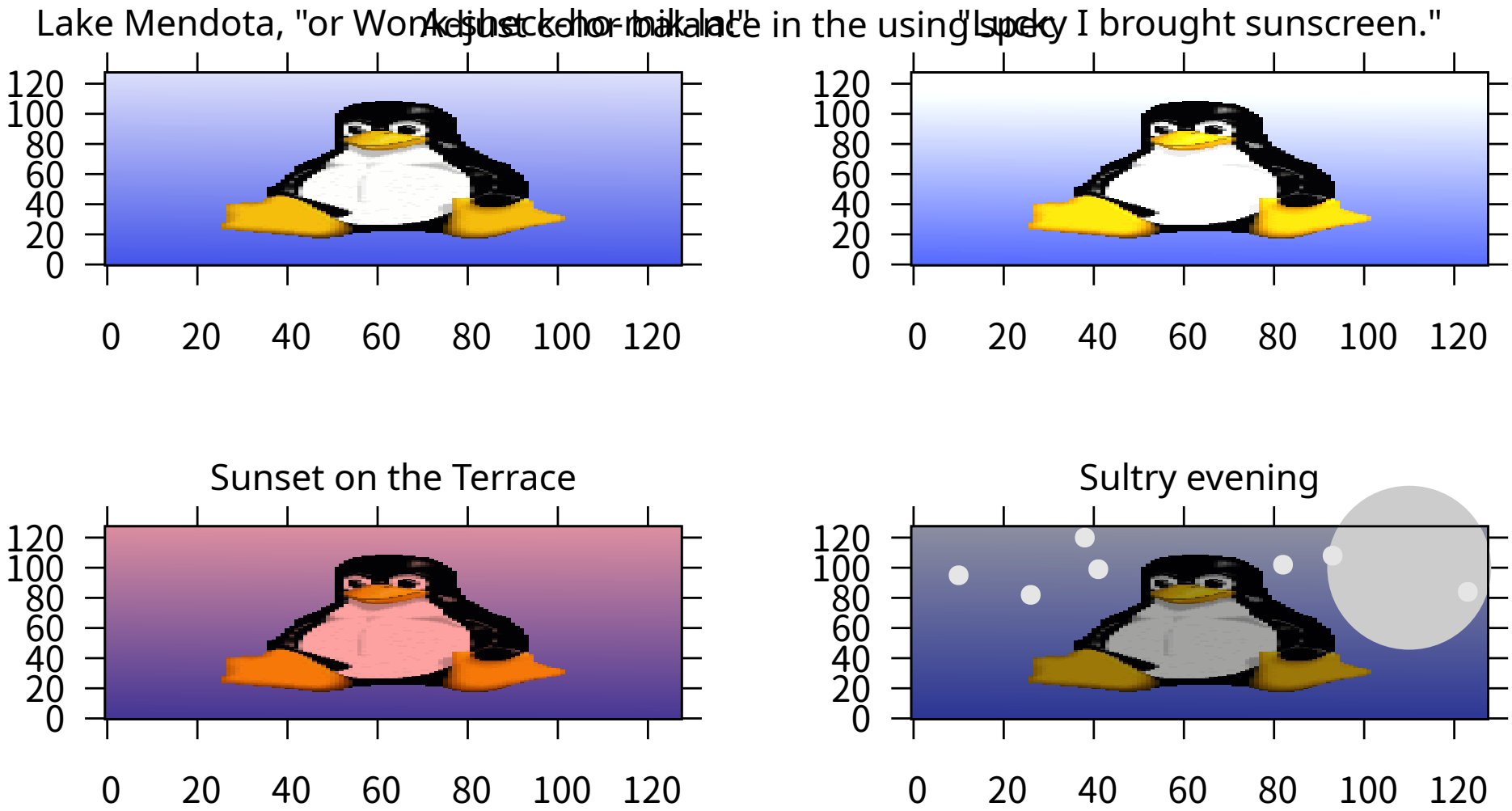


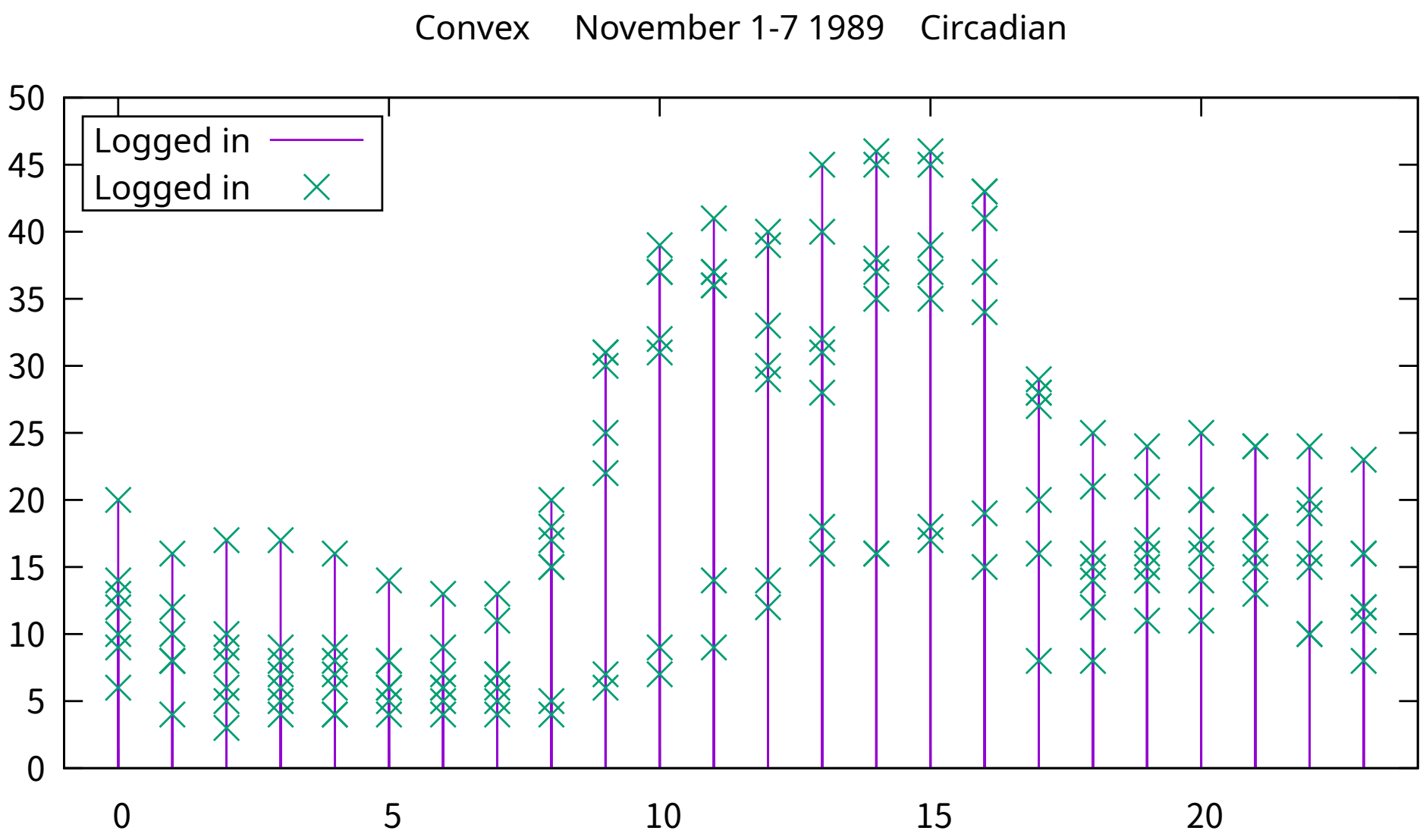
Terminal image routine used to draw plot rotated about origin
Notice the faster refresh rate than for the previous plot

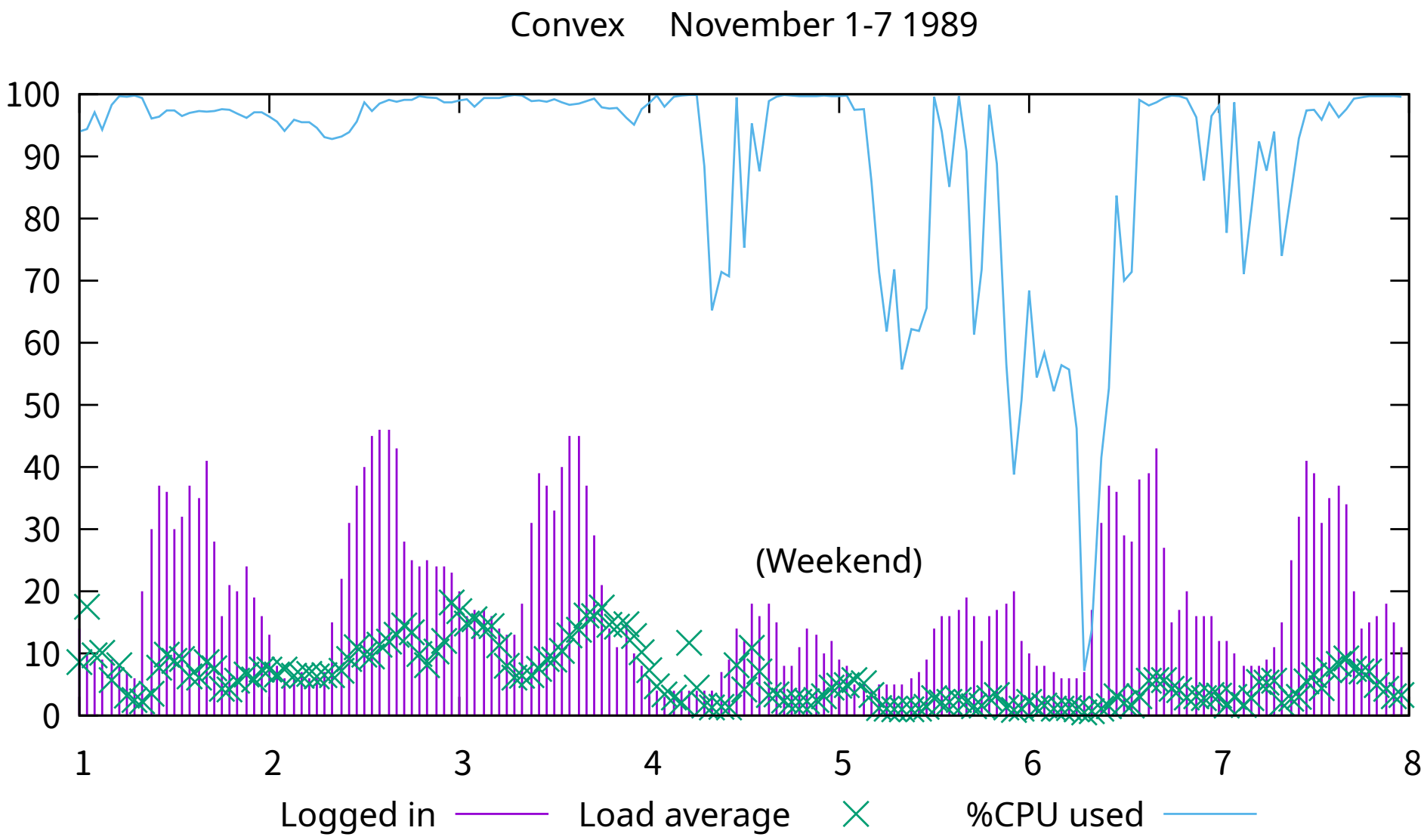


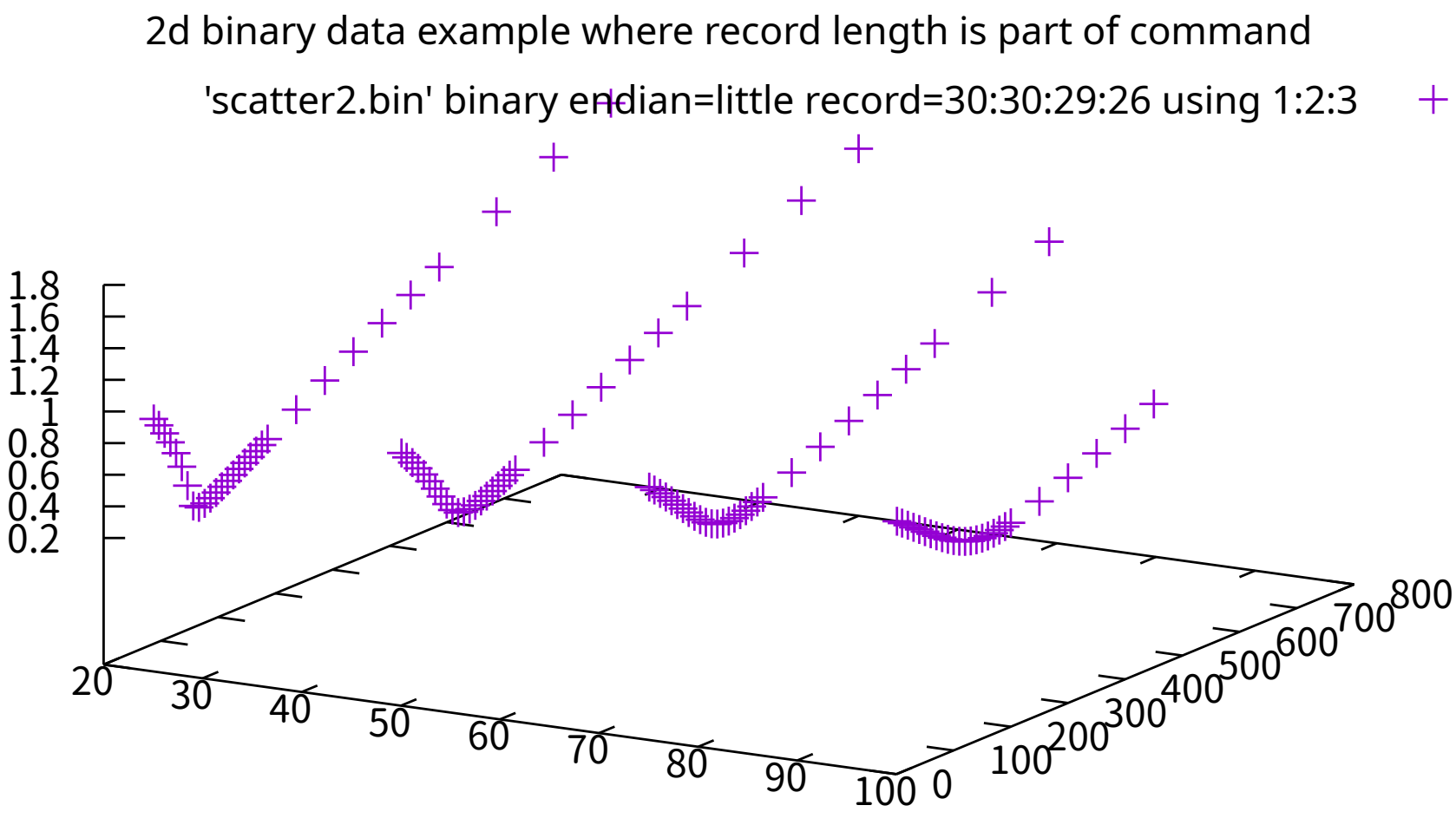
Selections of the input channels via `using` "I do impersonations."



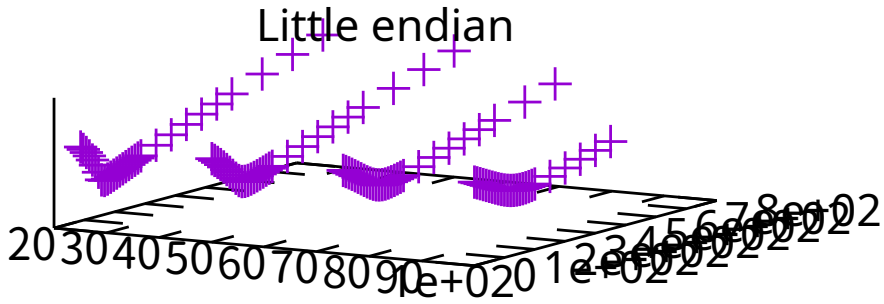

Lake Mendota, "or Wonk-sheck-ho-bik-ah!"
Adjust color balance
"Lucky I brought sunscreen."
Sunset on the Terrace
Sultry evening
0
20
40
60
80
100
120

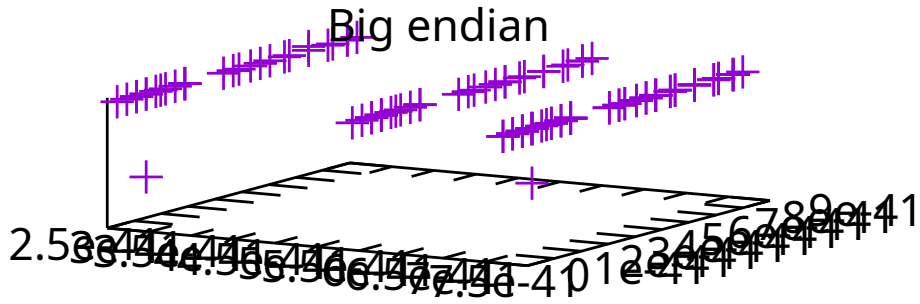

Convex November 1-7 1989 Circadian
Logged in
Logged in
50
45
40
35
30
25
20
15
10
5
0
0
5
10
15
20

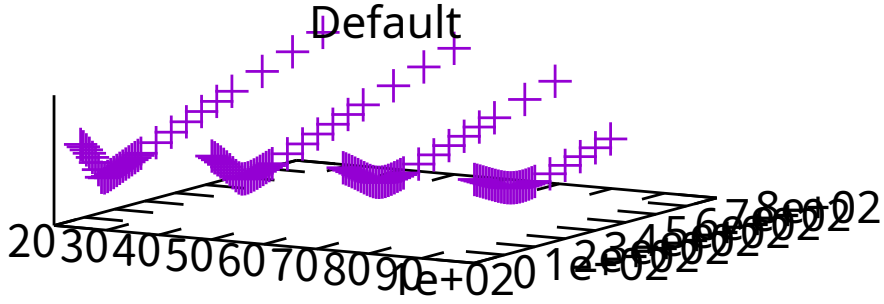

Convex November 1-7 1989
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
(Weekend)
Logged in
Load average
%CPU used

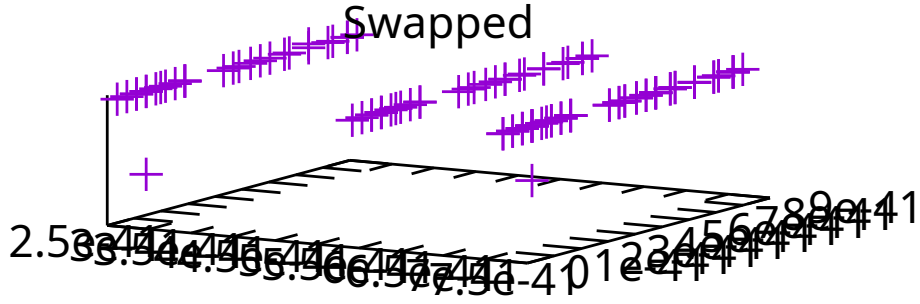

2d binary data example where record length is part of command
'scatter2.bin' binary endian=little record=30:30:29:26 using 1:2:3
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500
600
700
800

If plots in columns match, your compiler is little endian

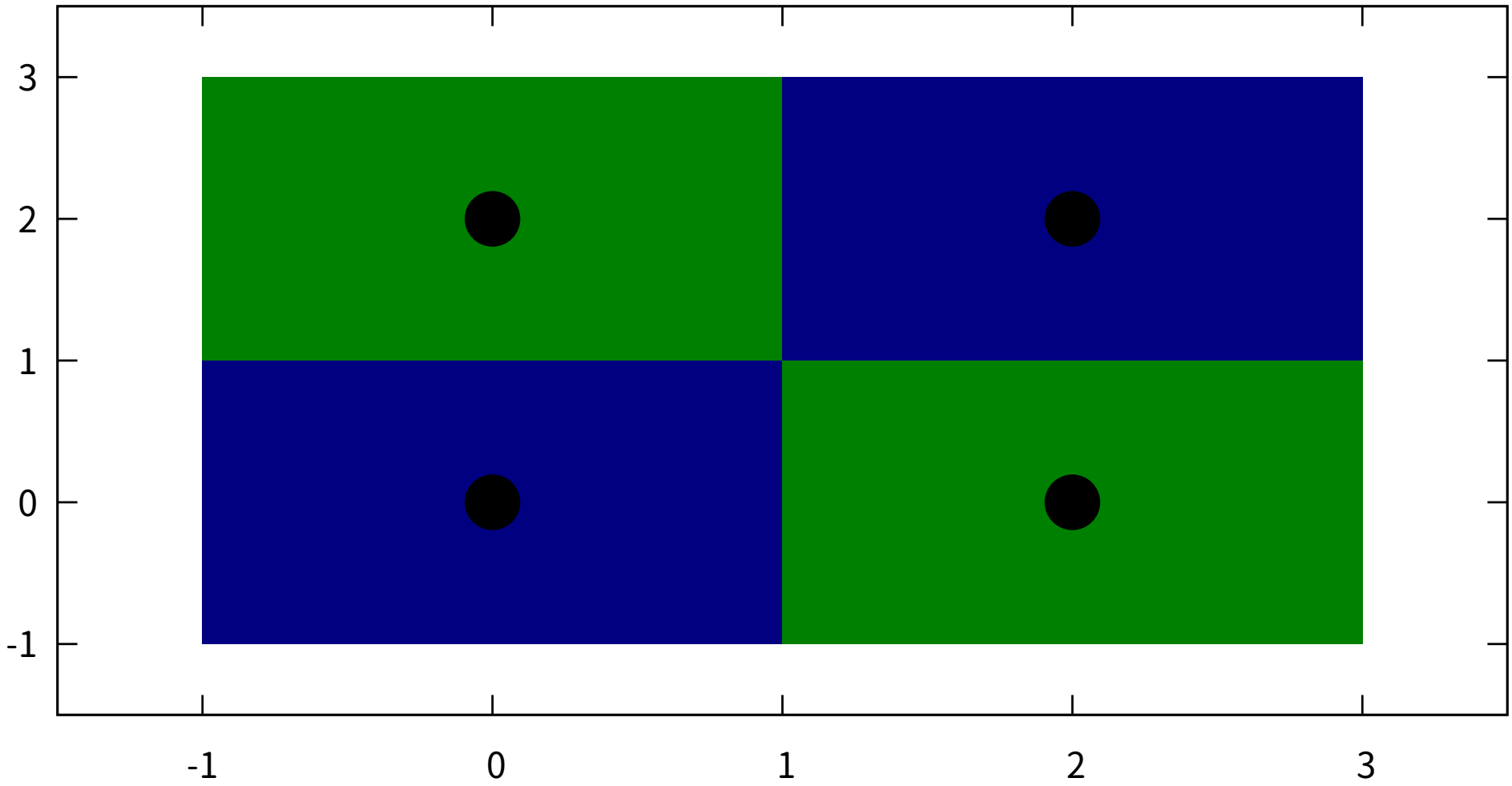






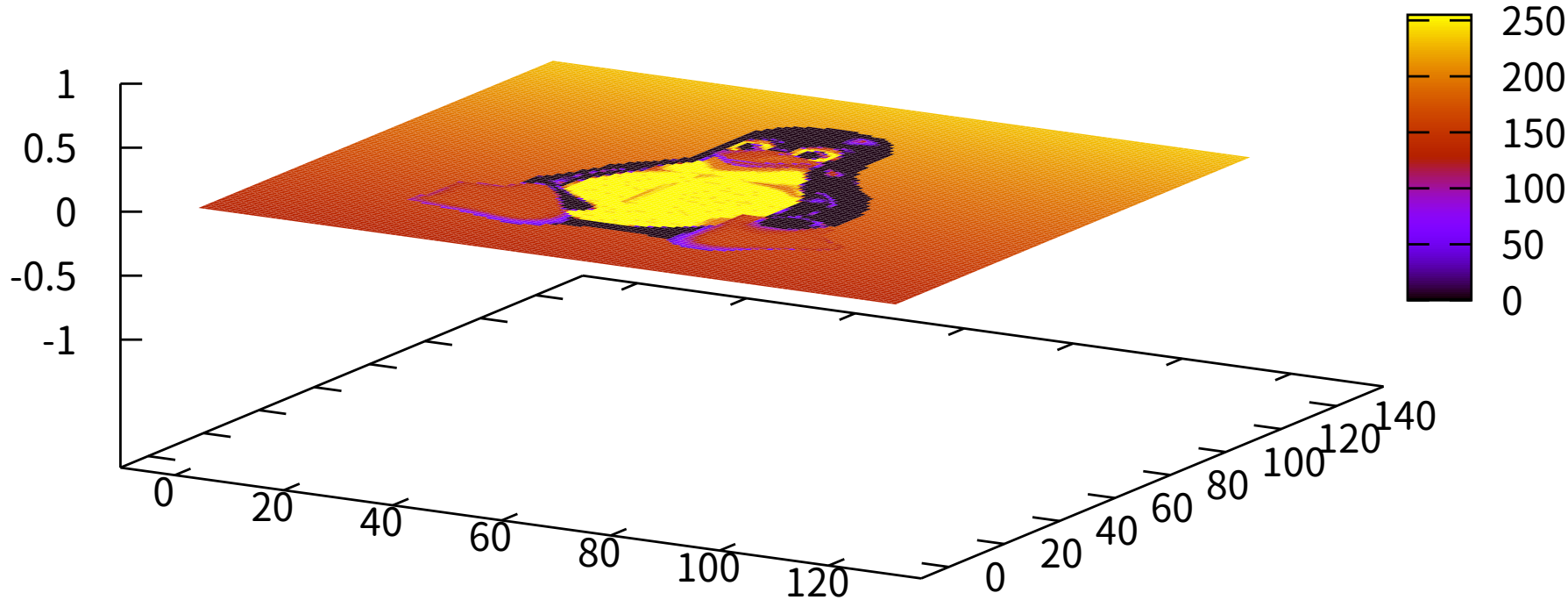


Close up of pixels having grid points (0,0), (0,2), (2,0) and (2,2)



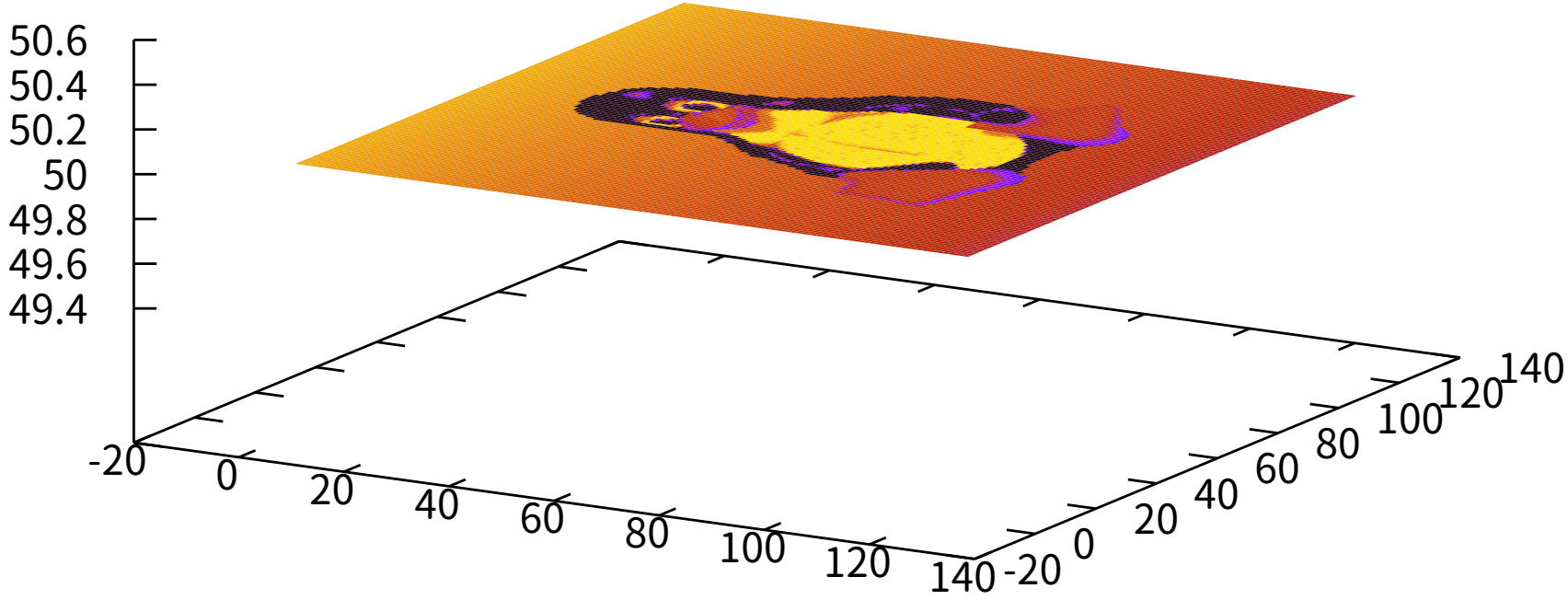
Simple extension of a two dimensional image into three dimensions

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



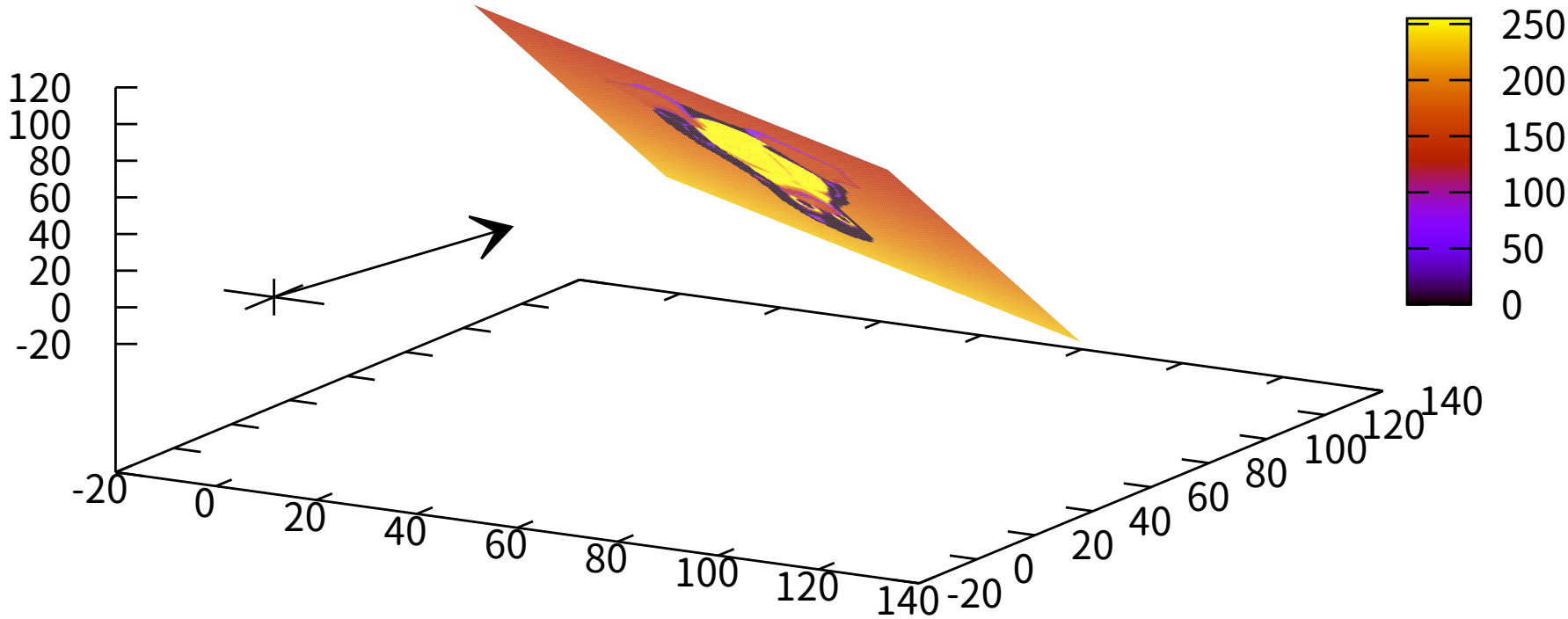
Orientation operations from 'plot' also apply to to 'splot'

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

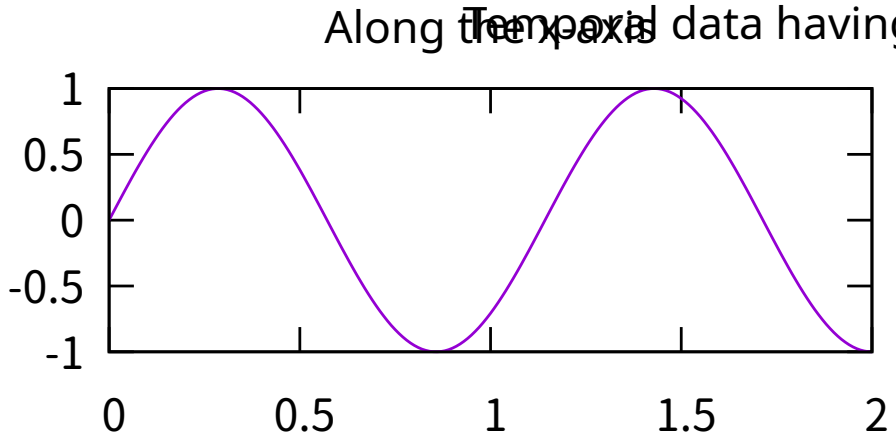


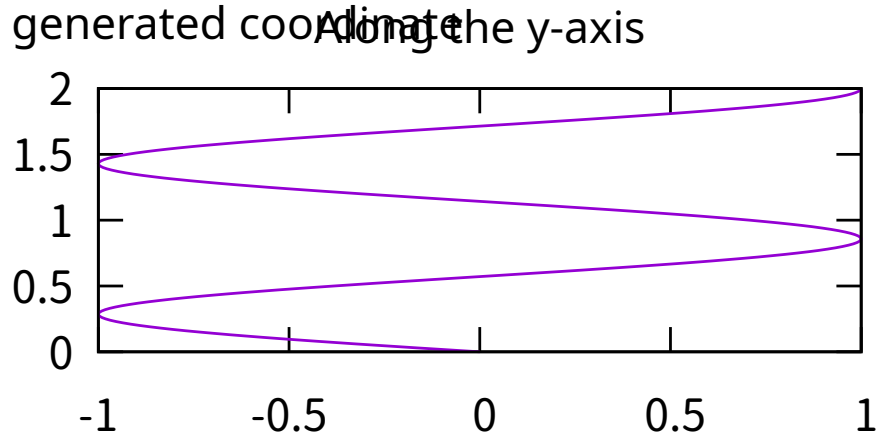
The key word 'perpendicular' applies only to 'splot'

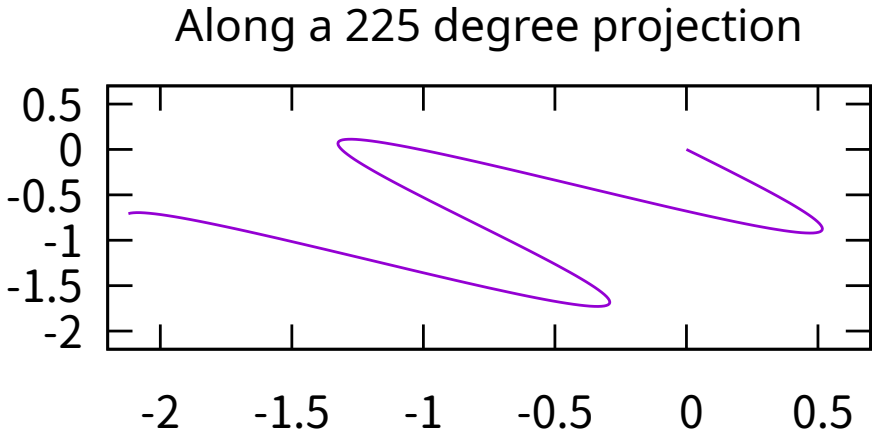
ter = (63.5,63.5,50) perp=(1,1,1) format='%uchar%uchar%uchar' using ($1+$2+$3)/3



Temporal data having one generated coordinate

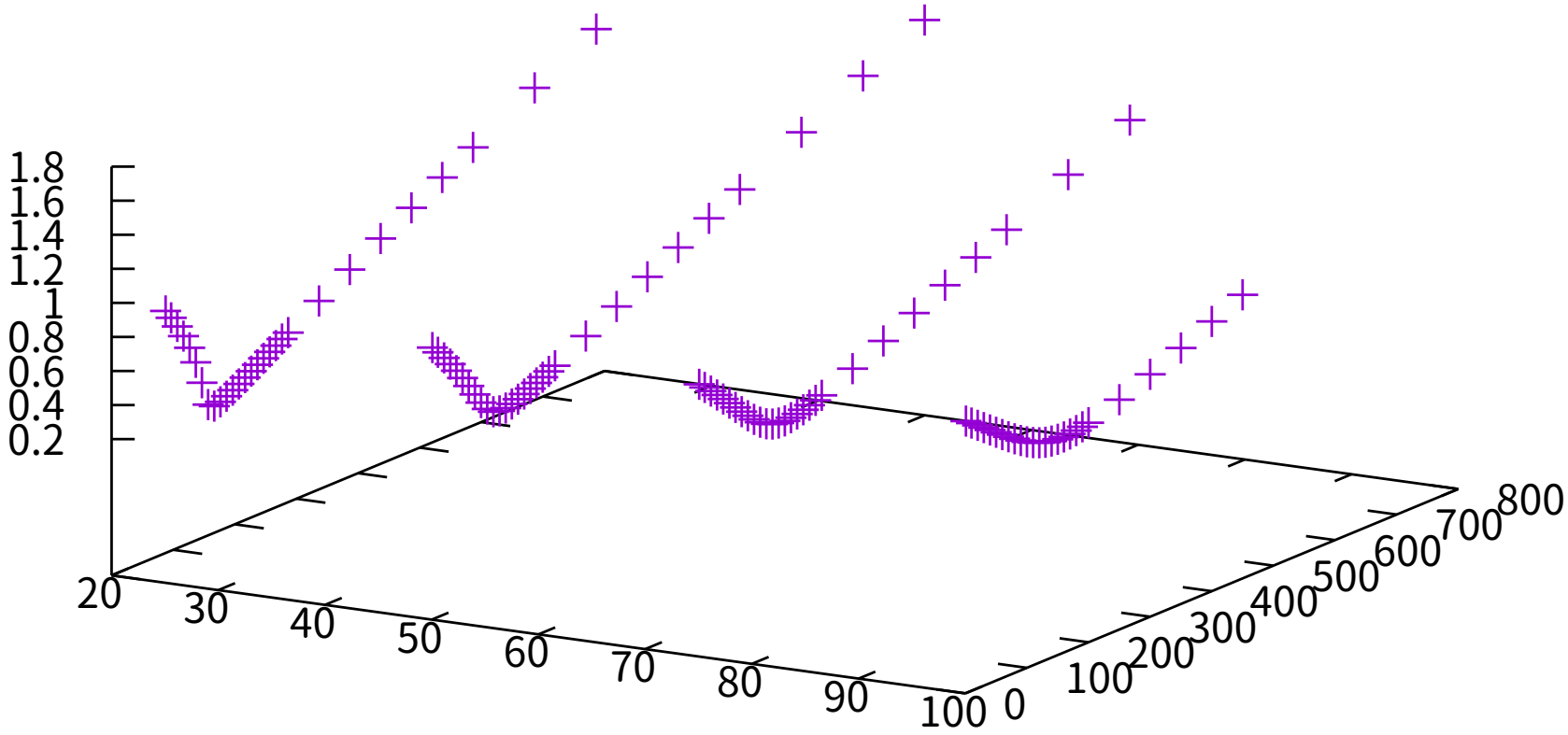




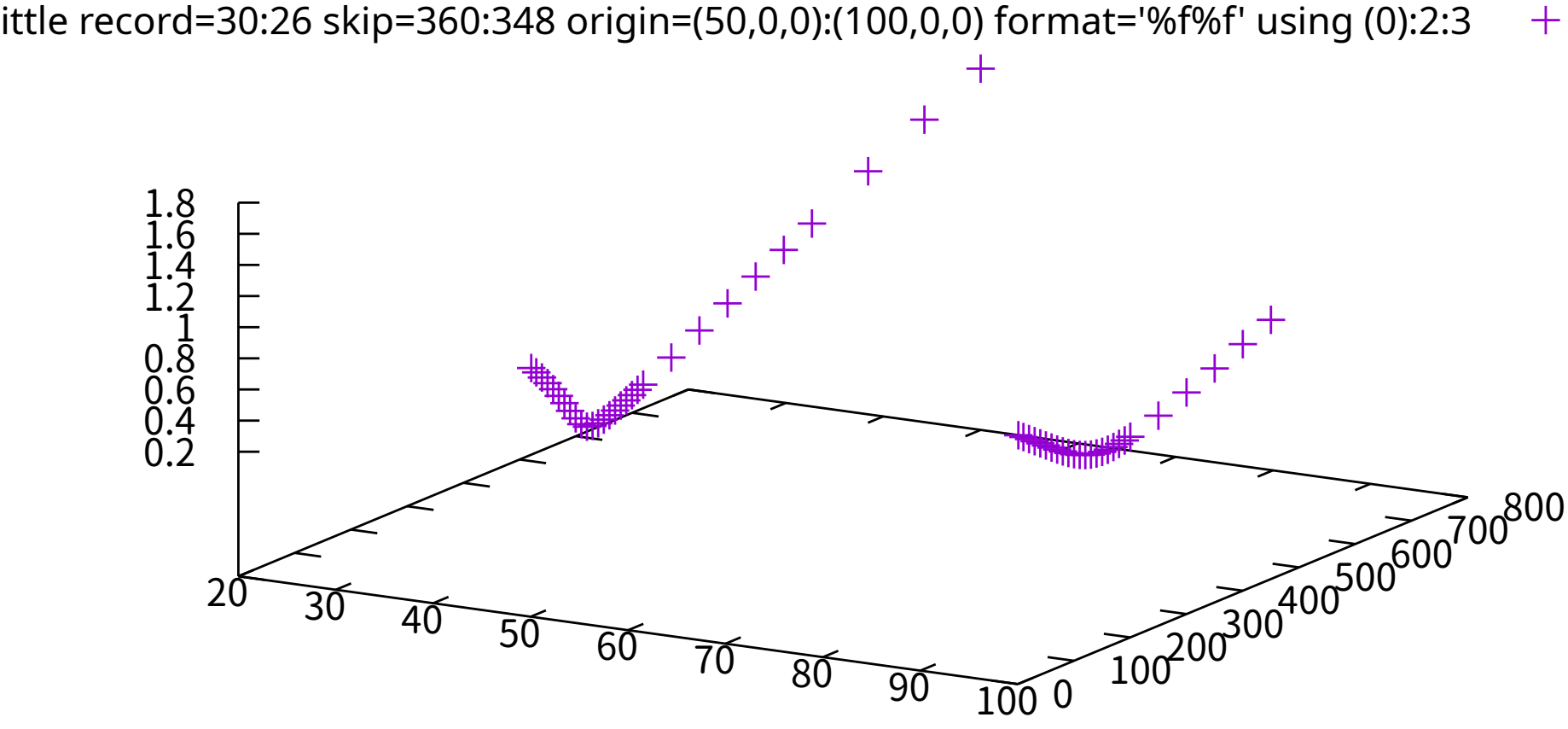


2d binary data example where x coordinate is ignored then generated

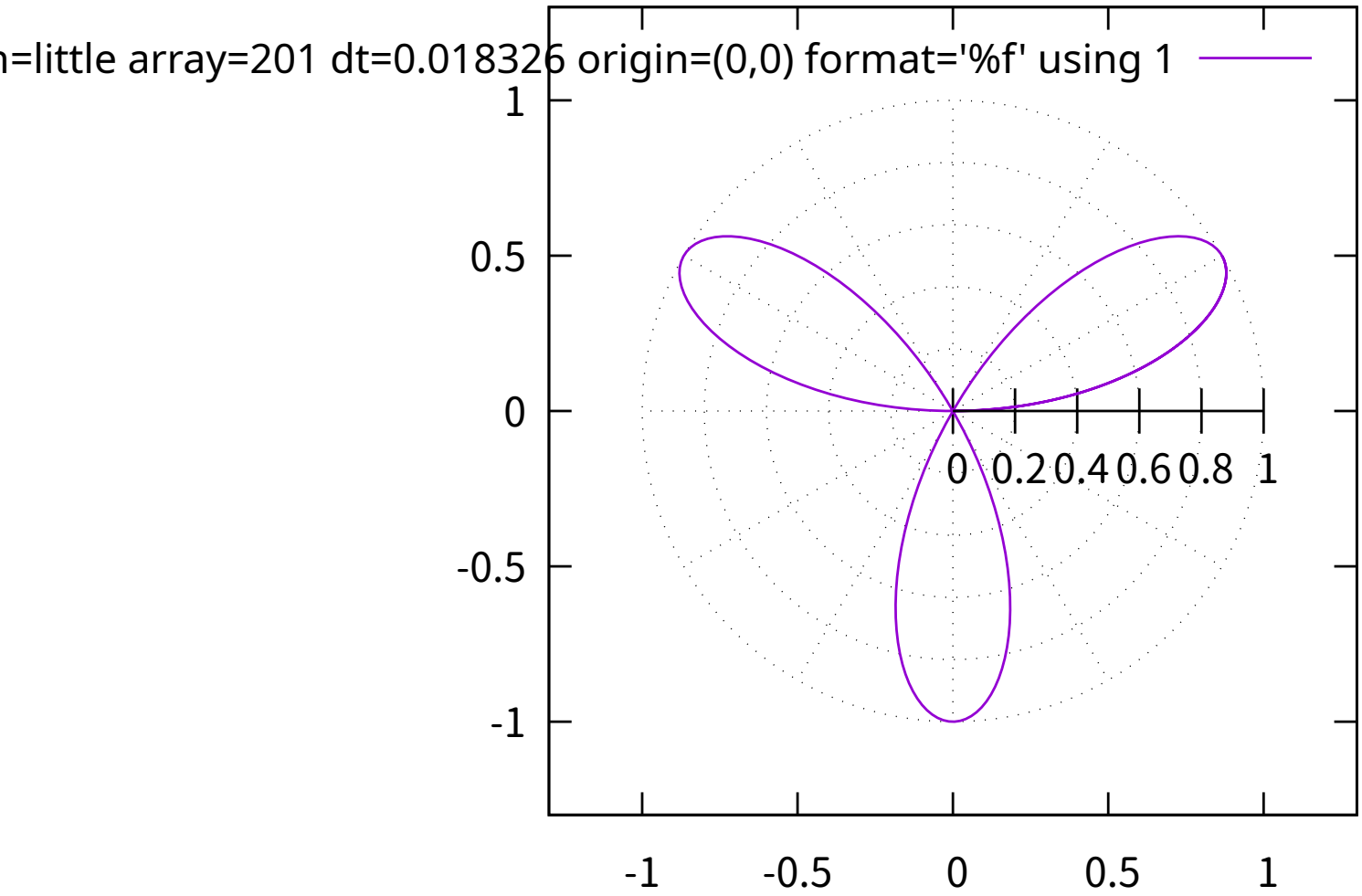
ord=30:30:29:26 origin=(25,0,0):(50,0,0):(75,0,0):(100,0,0) format='%f%f' using (0):2:3



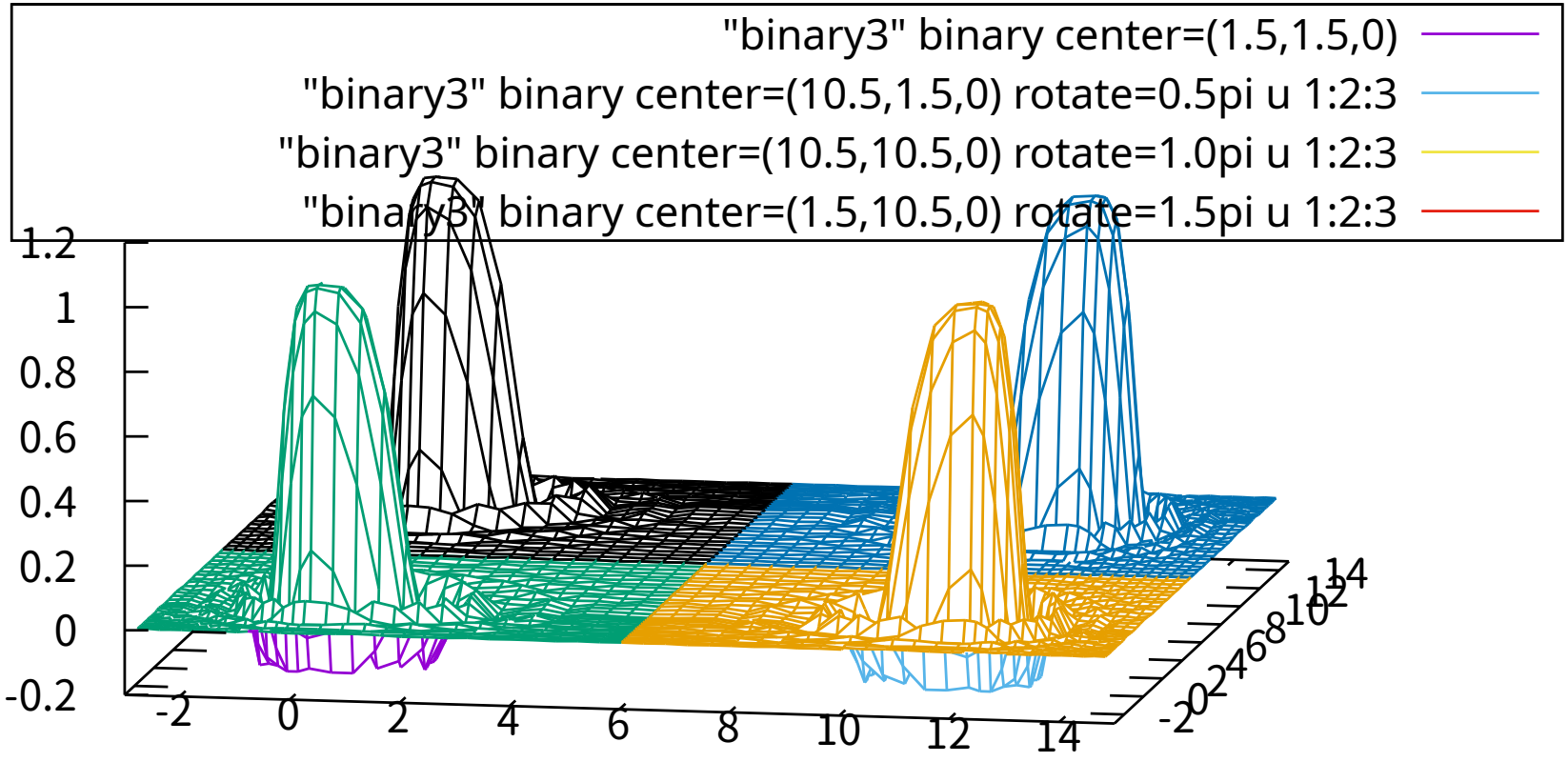
The key word 'skip' used to ignore some data

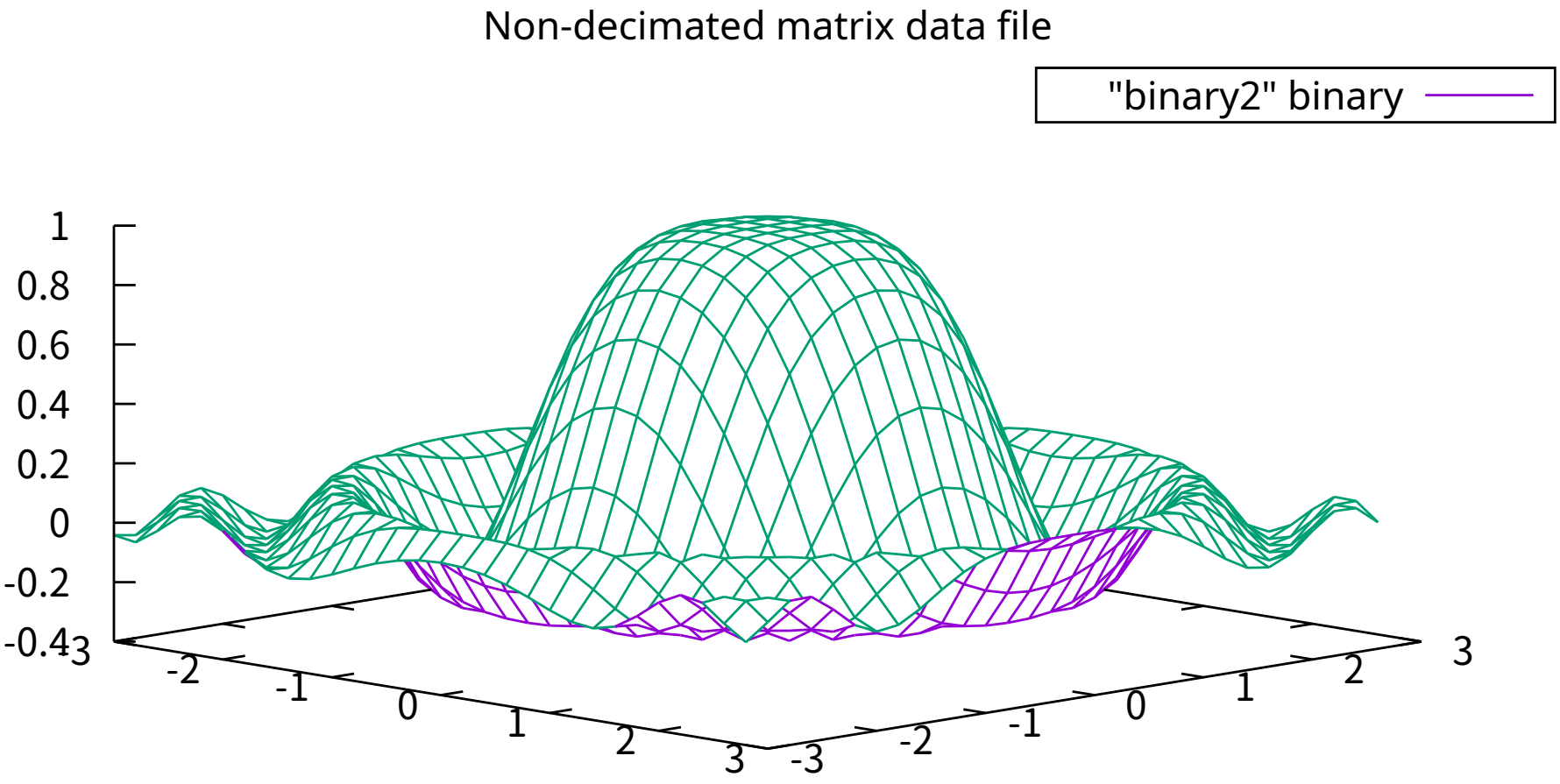


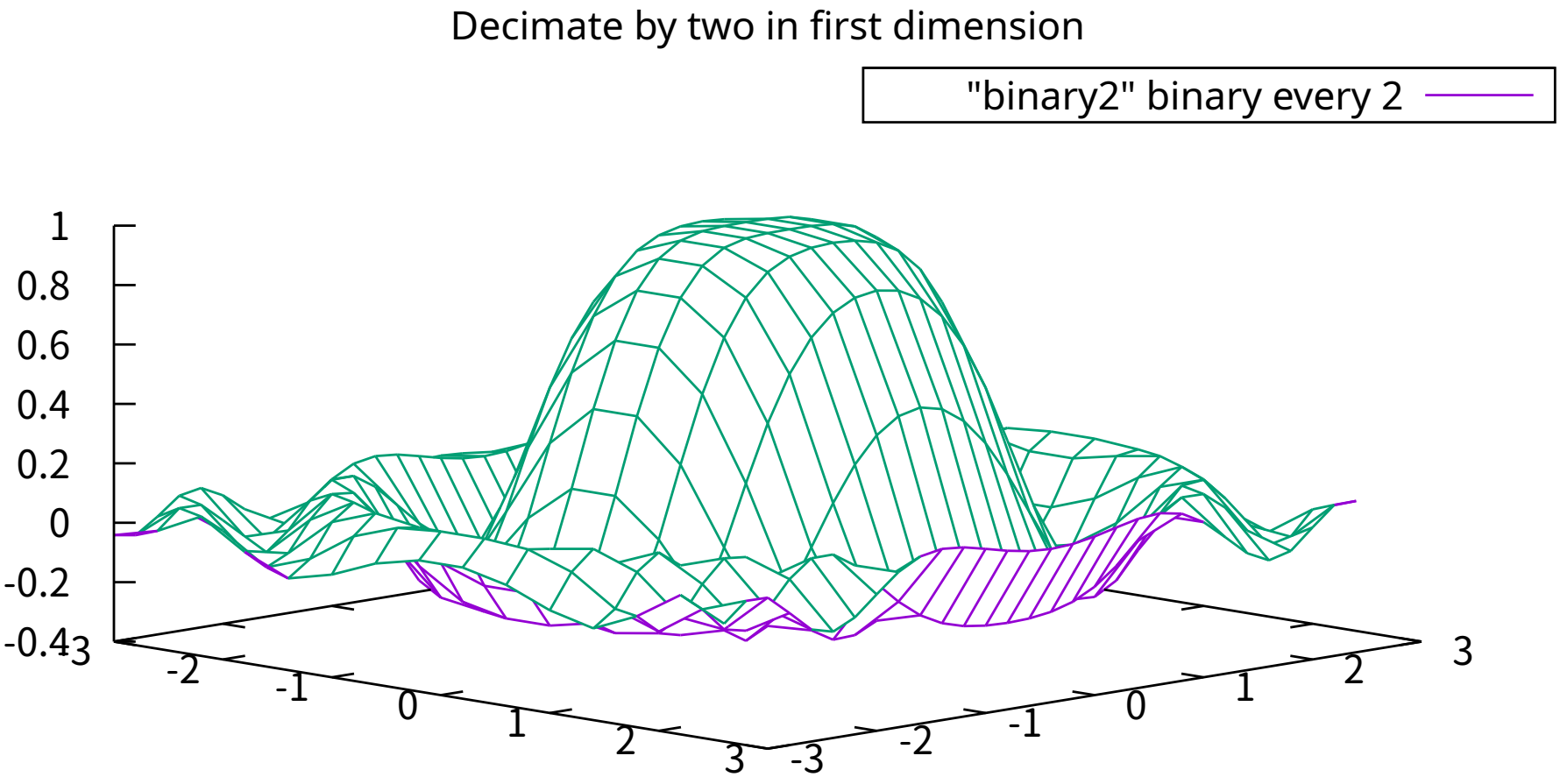
Uniform sampling in the polar coordinate system

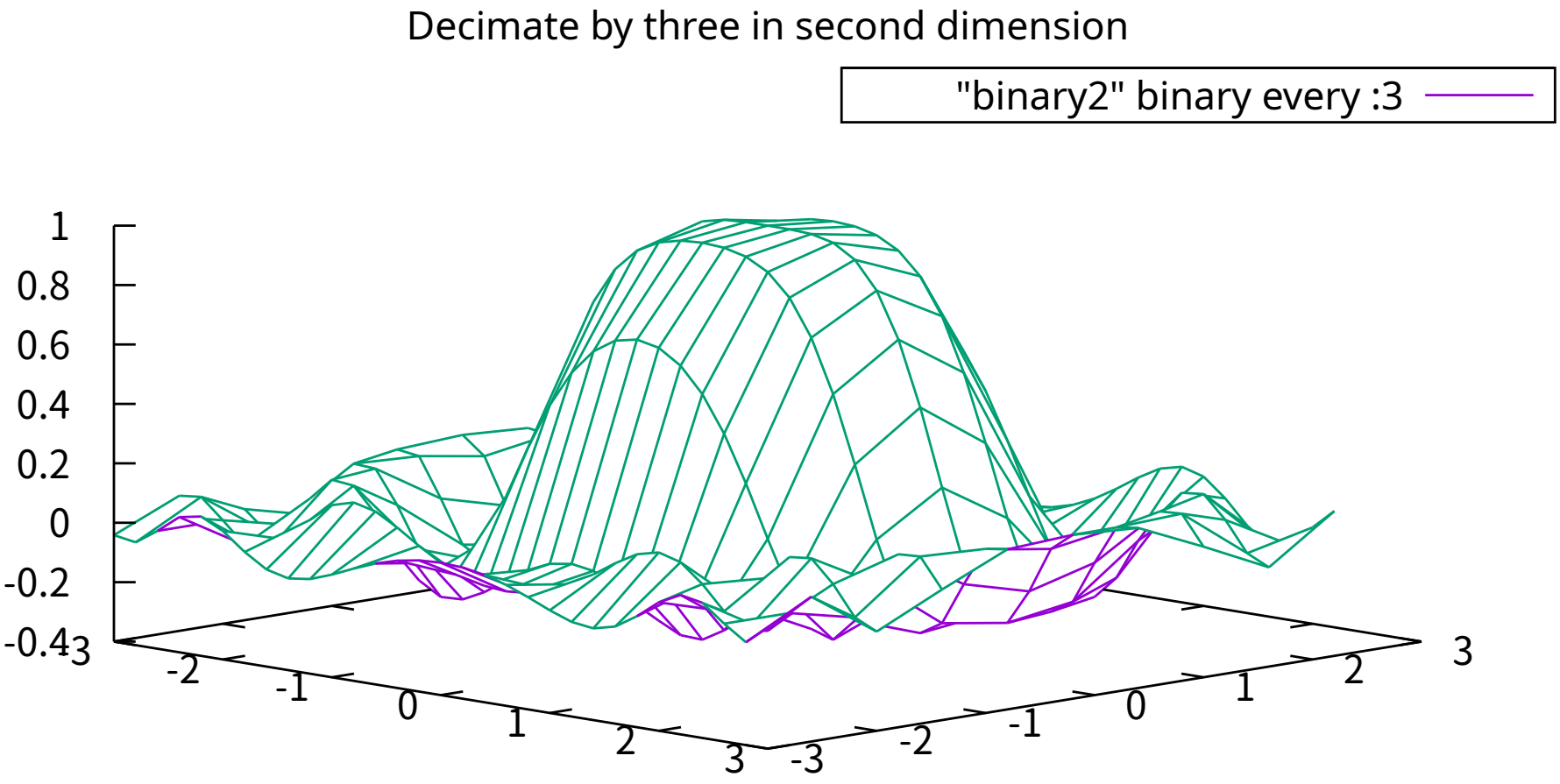


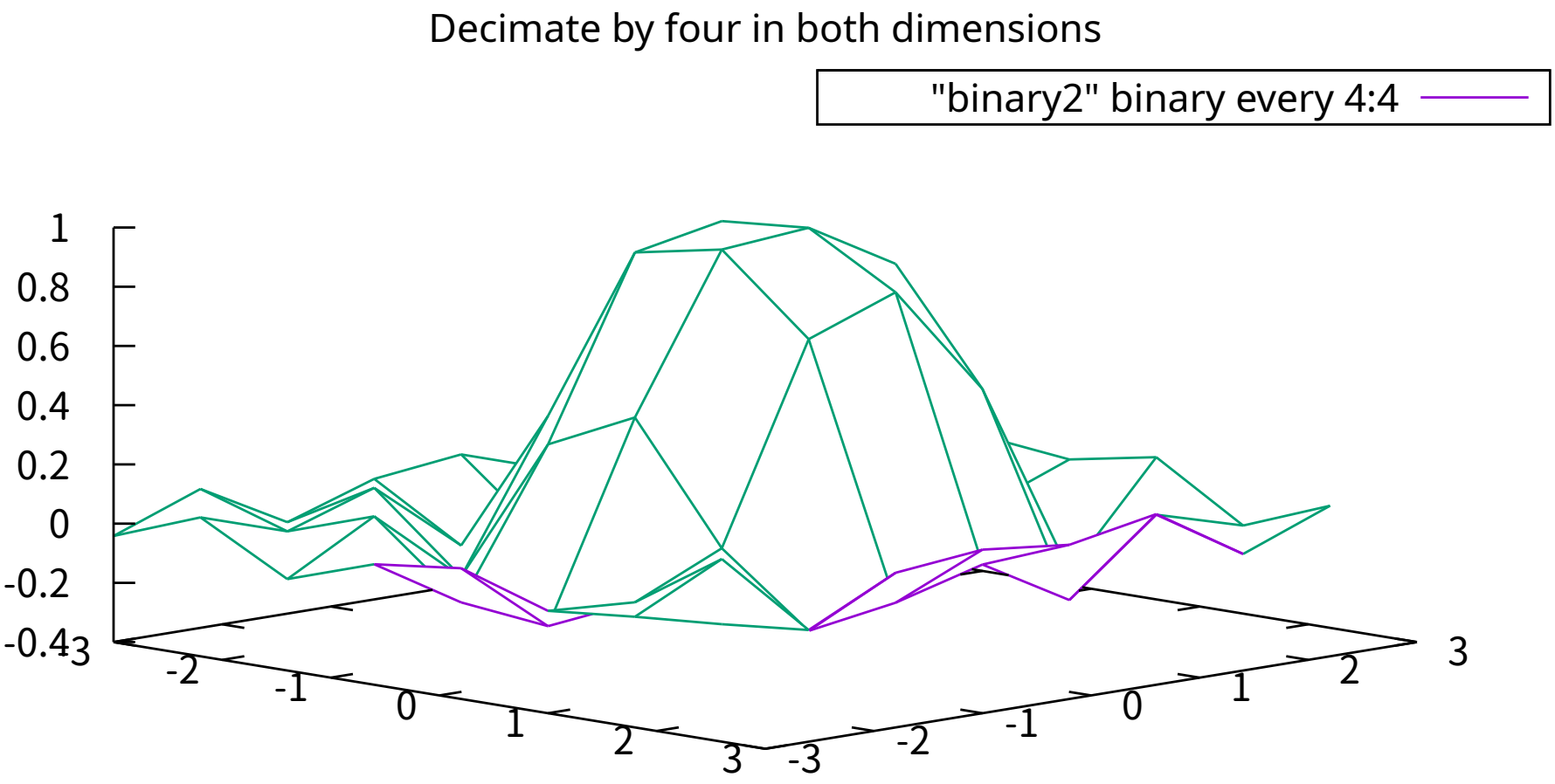
Matrix binary data (gnuplot binary) translated



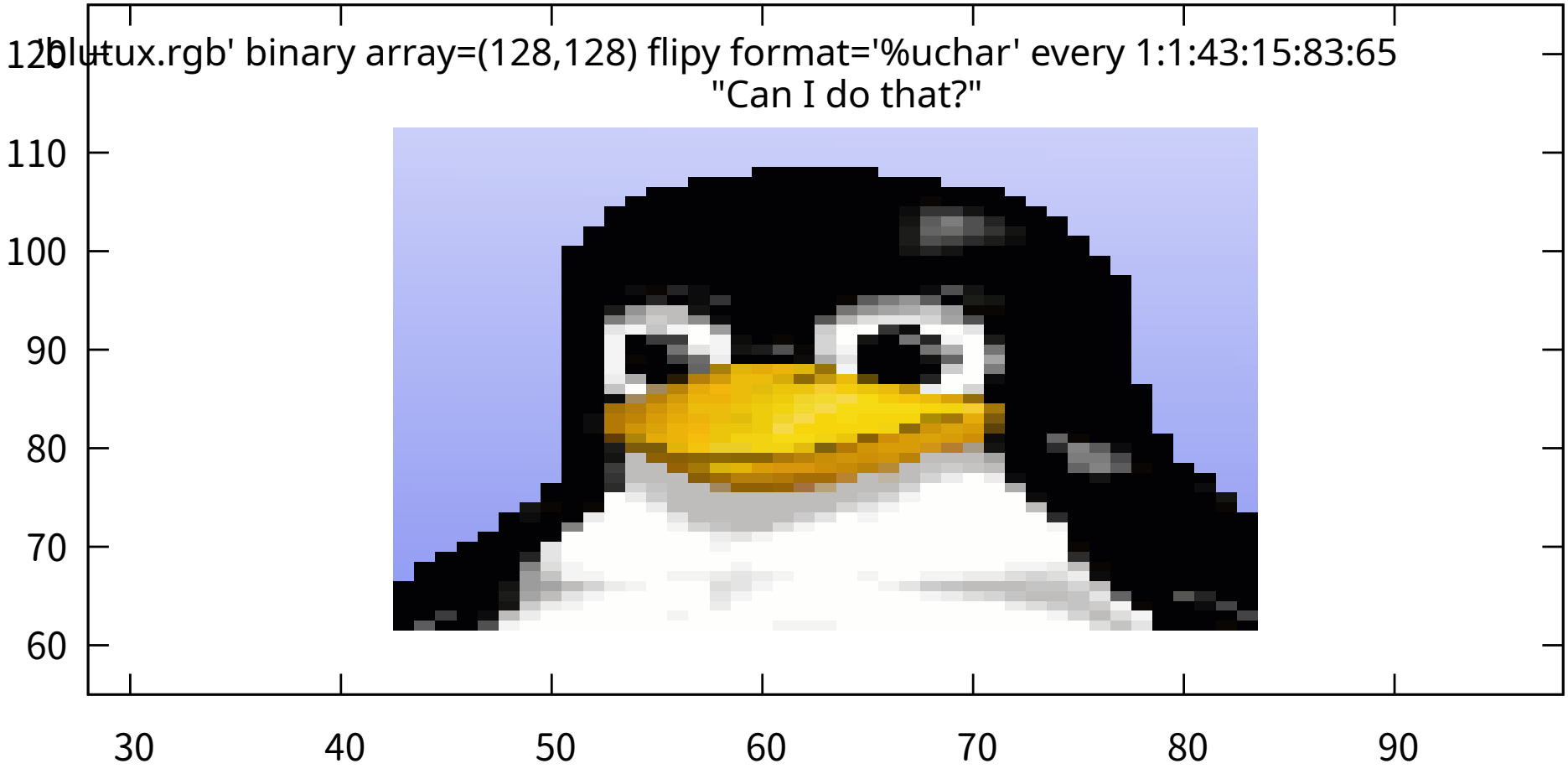

Non-decimated matrix data file
"binary2" binary
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
3
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3


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

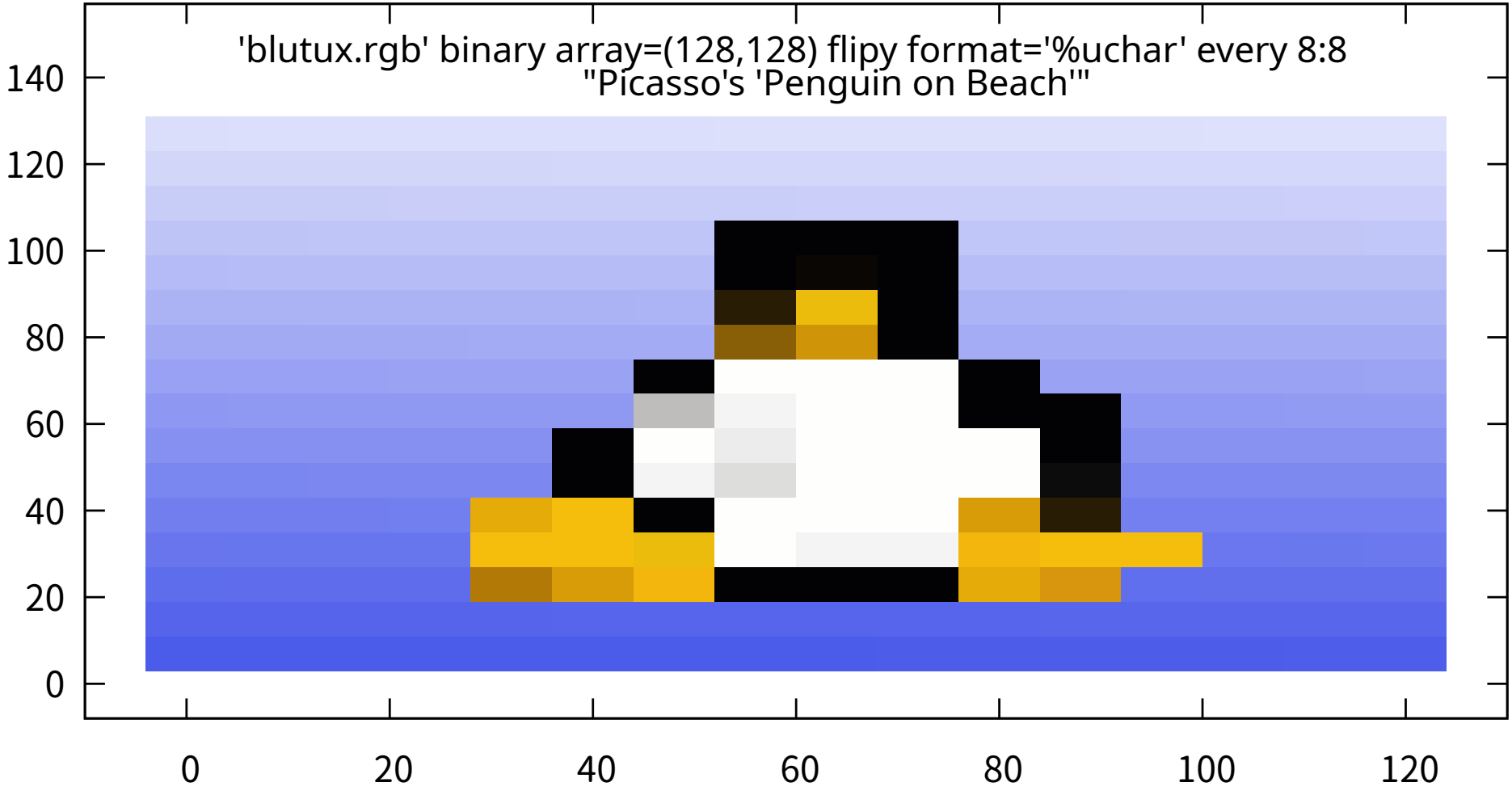

Decimate by three in second dimension
"binary2" binary every :3
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
3
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3


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

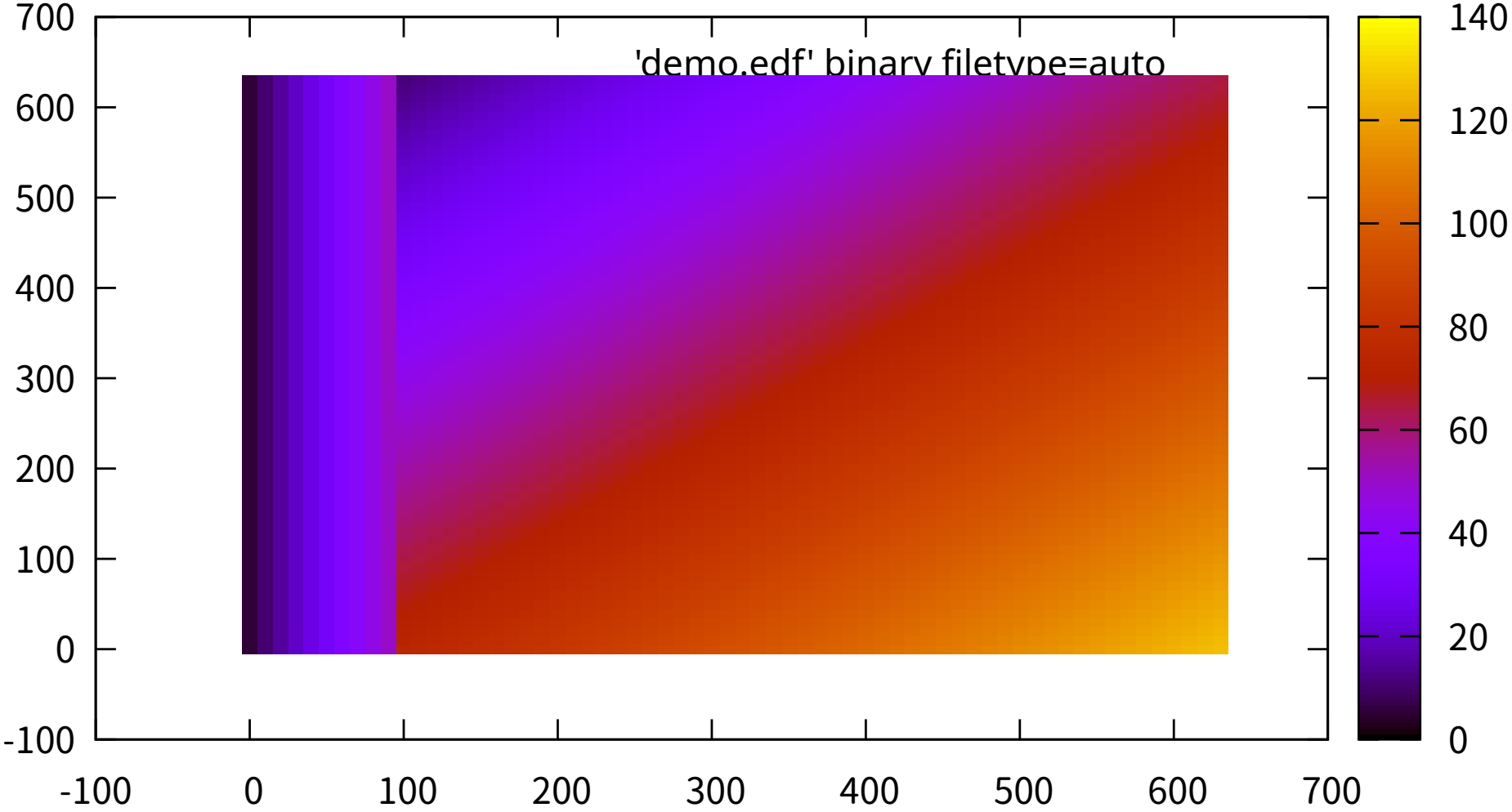
Decimation works on general binary data files as well.
Let Tux have his fun...

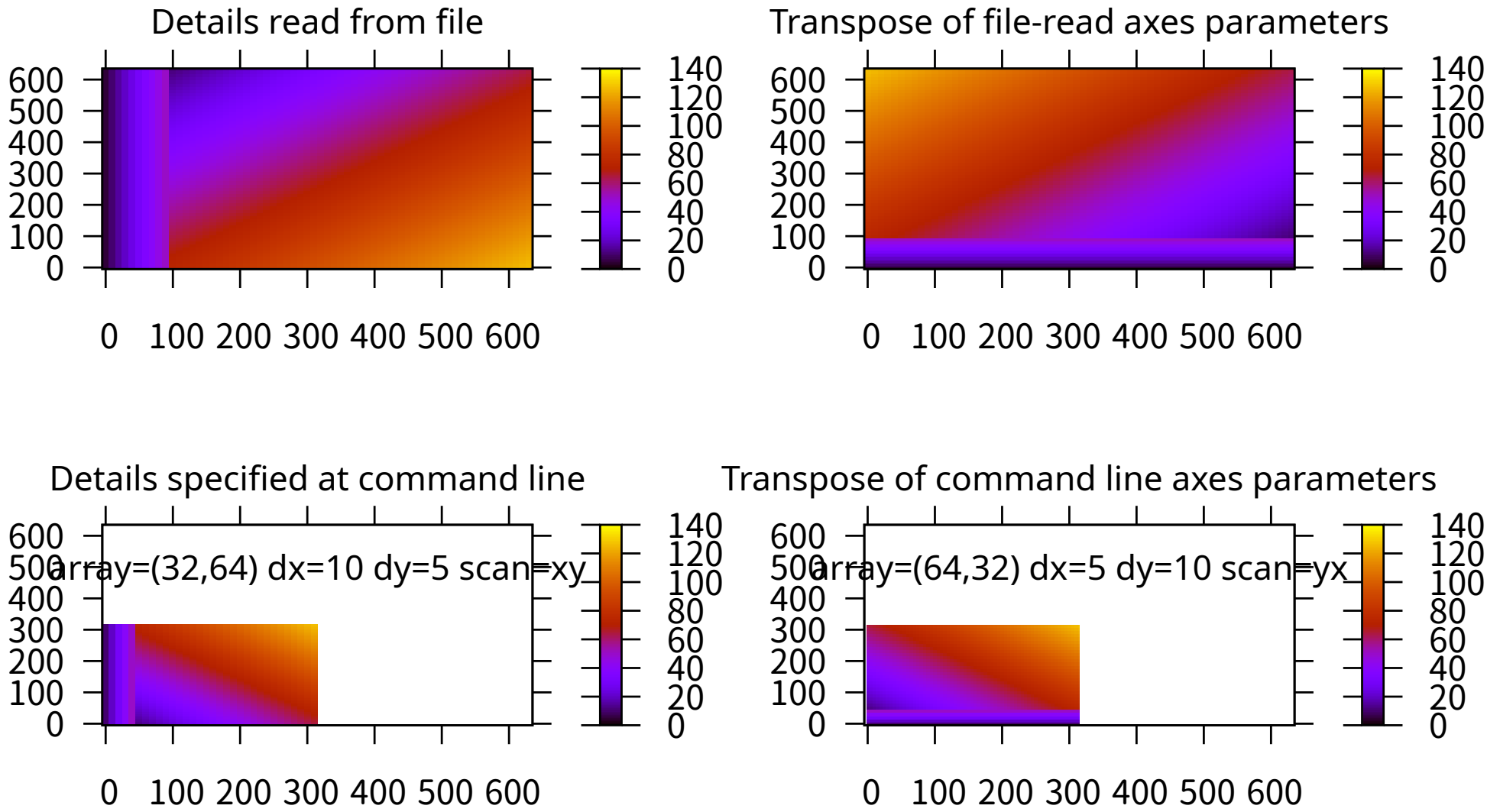


... Sure, go ahead.

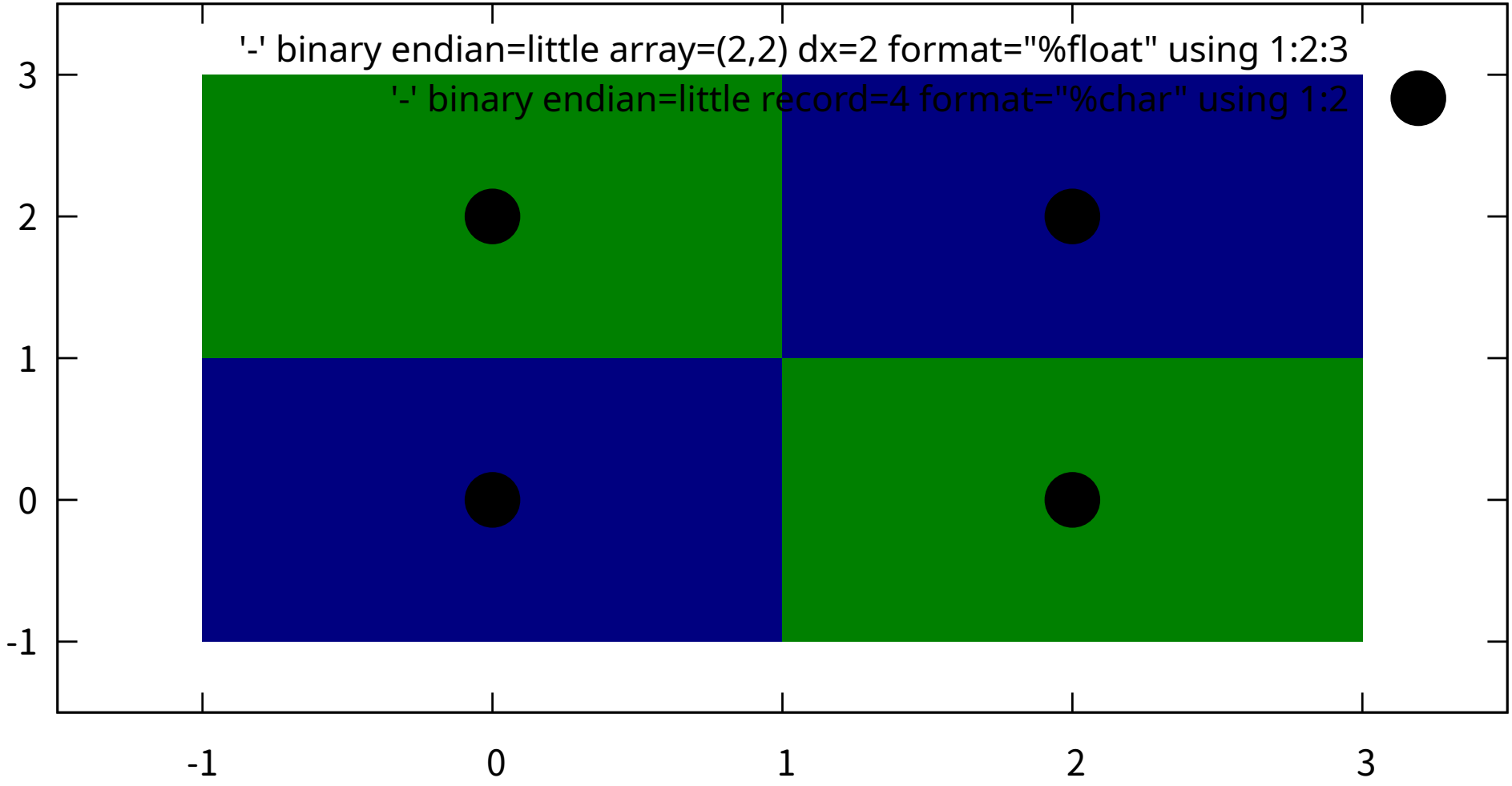


Automatically recognizing file type and extracting file information

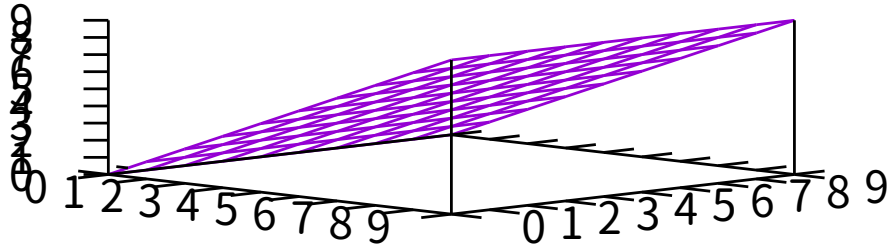



Details read from file
Transpose of file-read axes parameters
Details specified at command line
array=(32,64) dx=10 dy=5 scan=xy
Transpose of command line axes parameters
array=(64,32) dx=5 dy=10 scan=yx
0 100 200 300 400 500 600
0 20 40 60 80 100 120 140

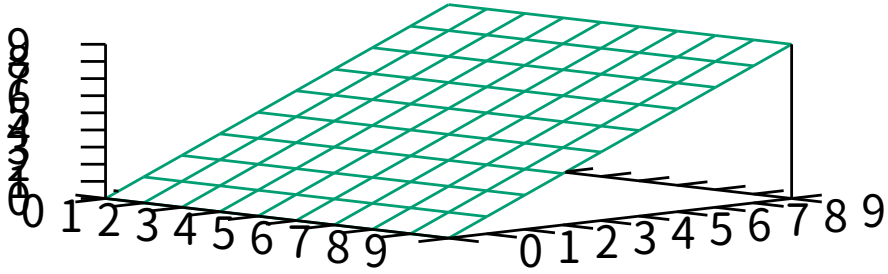
Binary data specified at the command line, intended for use through pipe



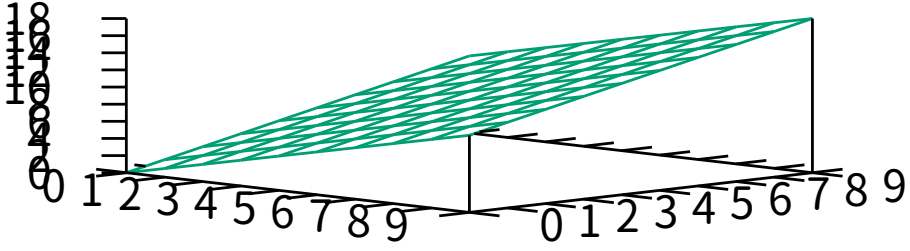
'asciimat.dat' matrix index 0



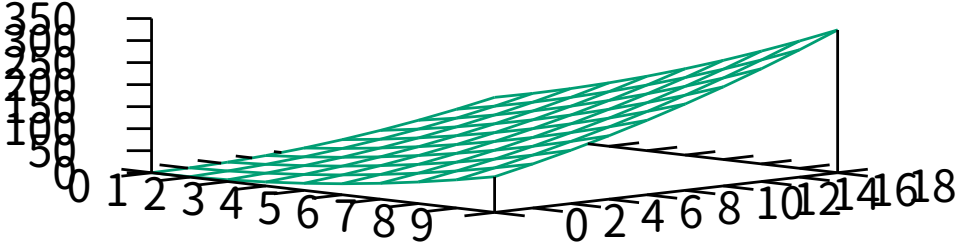
'asciimat.dat' matrix index 1



'asciimat.dat' matrix index 2



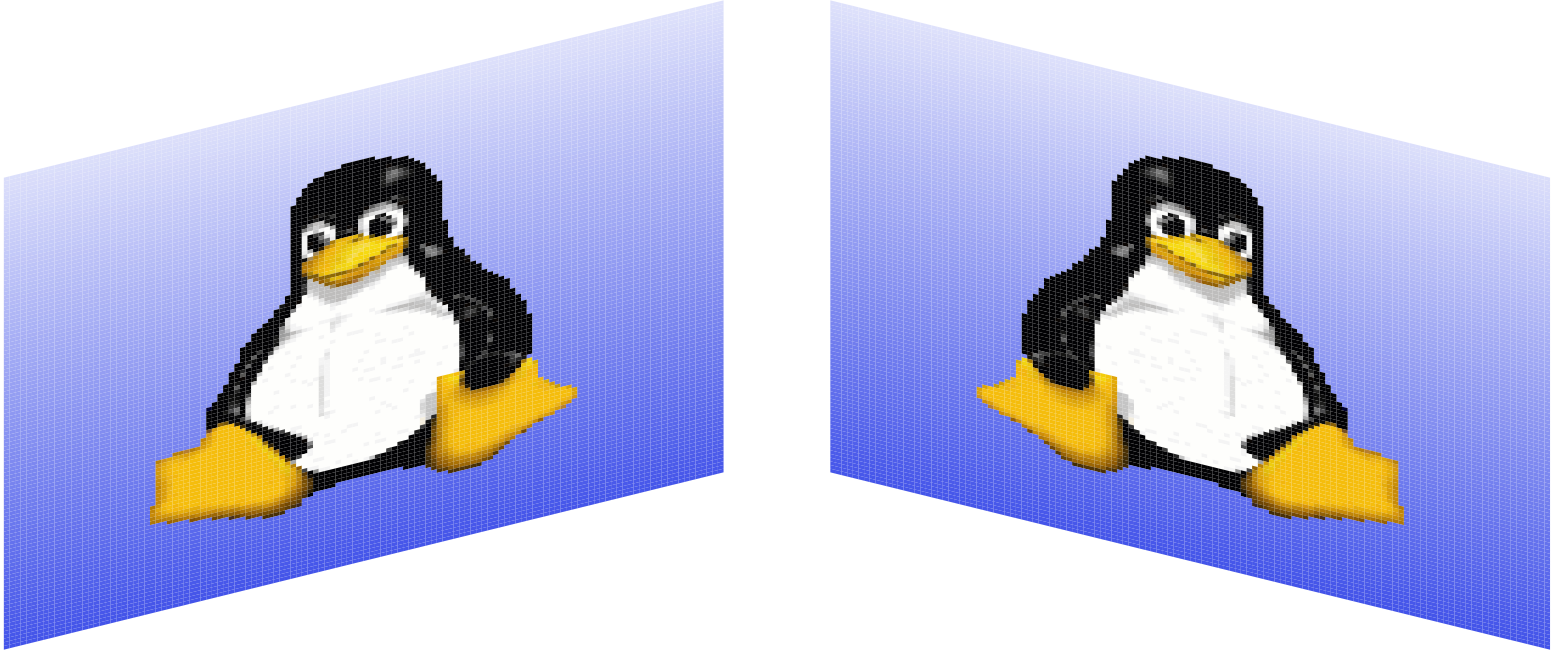
'asciimat.dat' matrix index 2 using 1:(2*$2):($3*$3



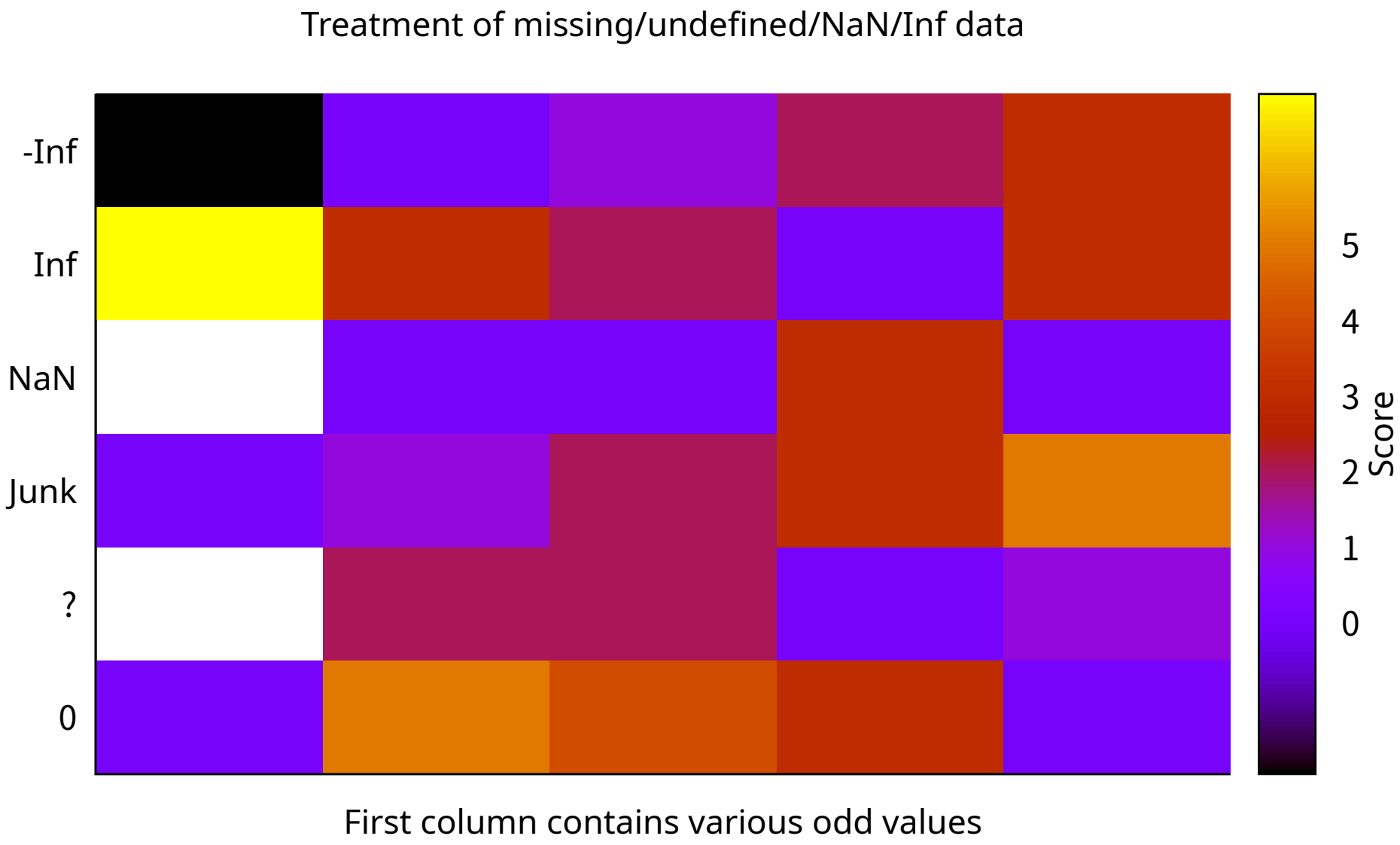
"Eccentric coordinate systems"

Image reverse according to axis orientation

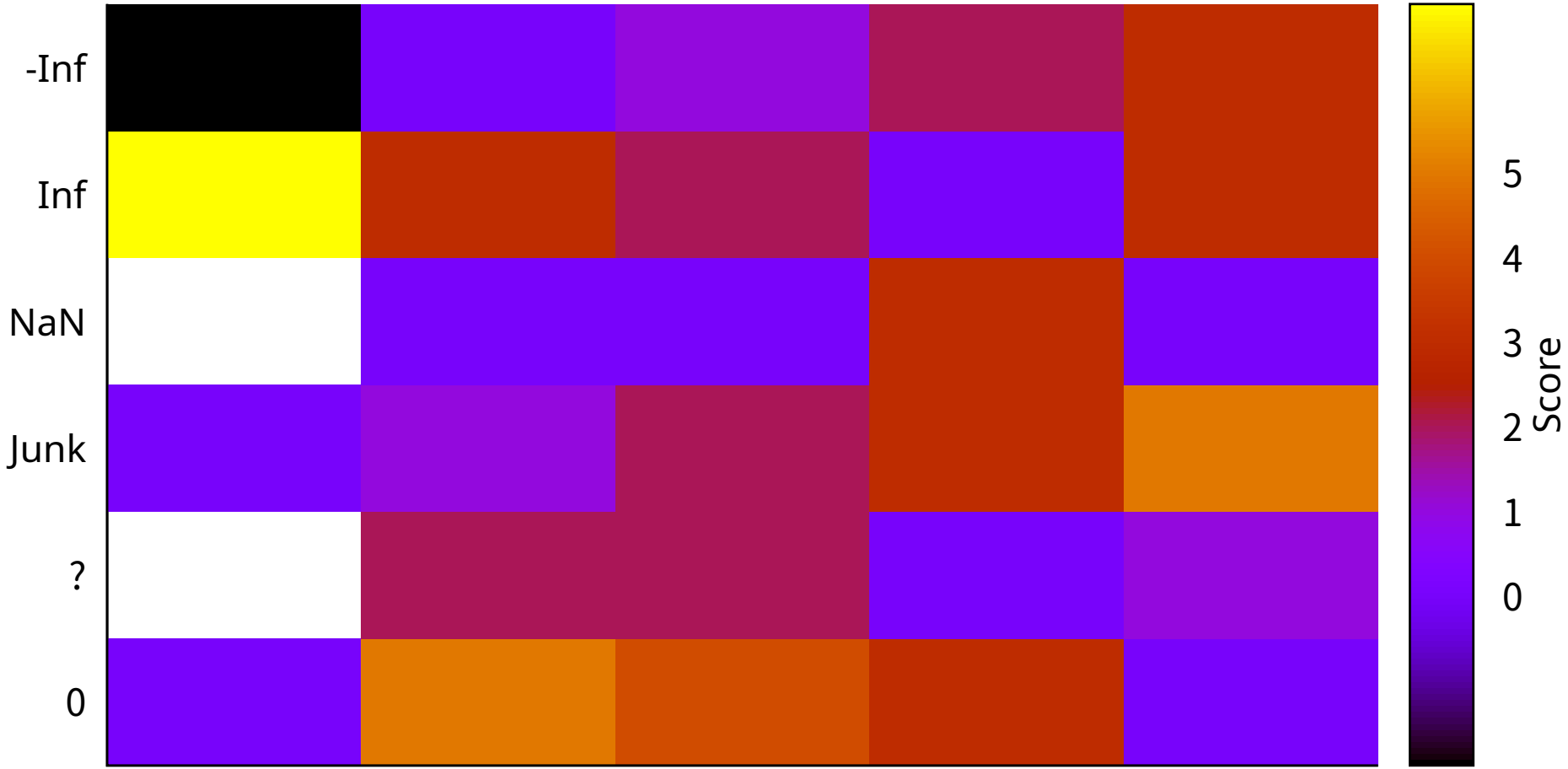
"Cartesian plane!"



Tux in a reflective mood

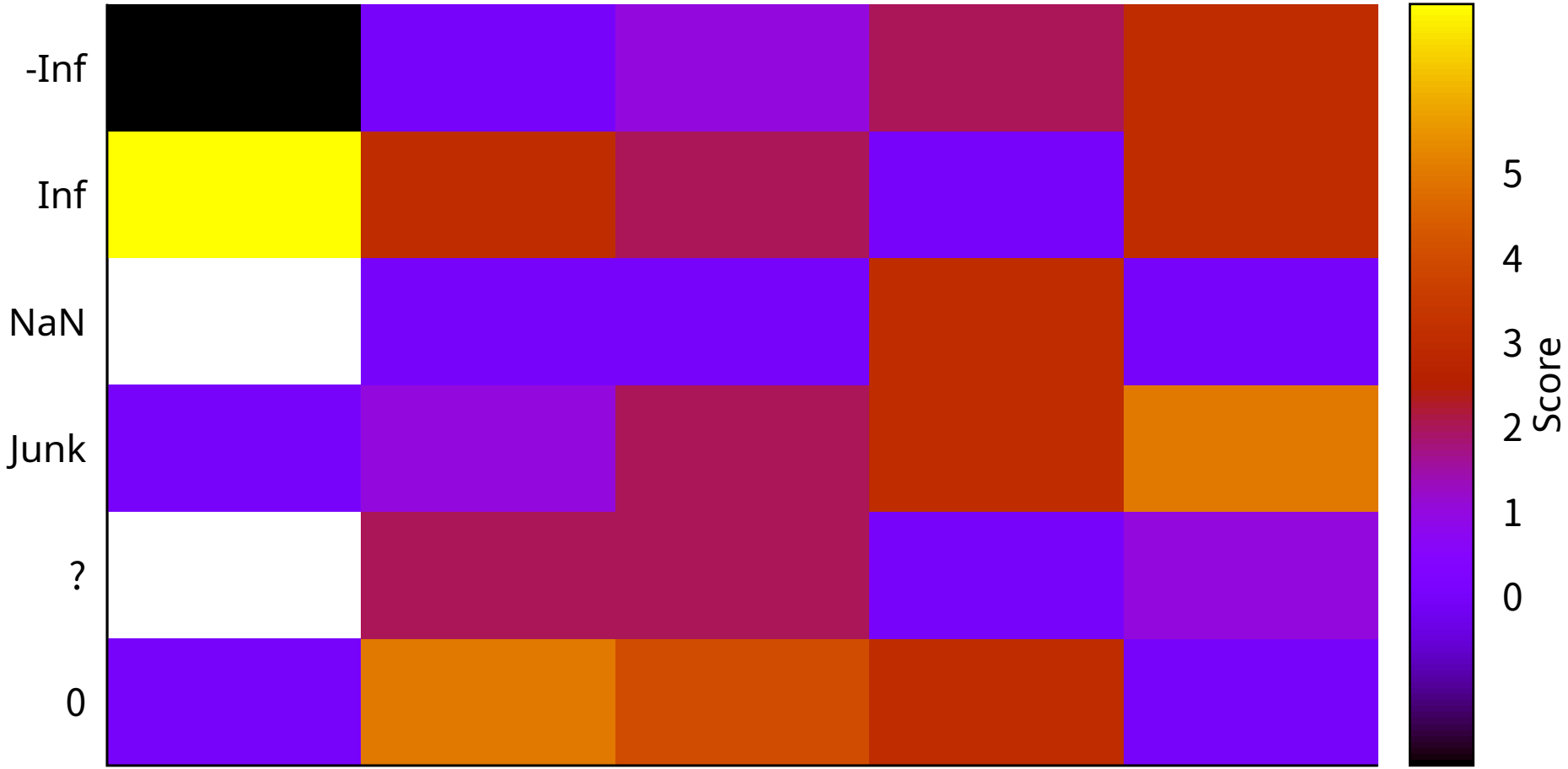

Treatment of missing/undefined/NaN/Inf data
-Inf
Inf
NaN
Junk
?
0
5
4
3
2
1
0
Score
First column contains various odd values

Same thing in 'pixels' mode (2D)



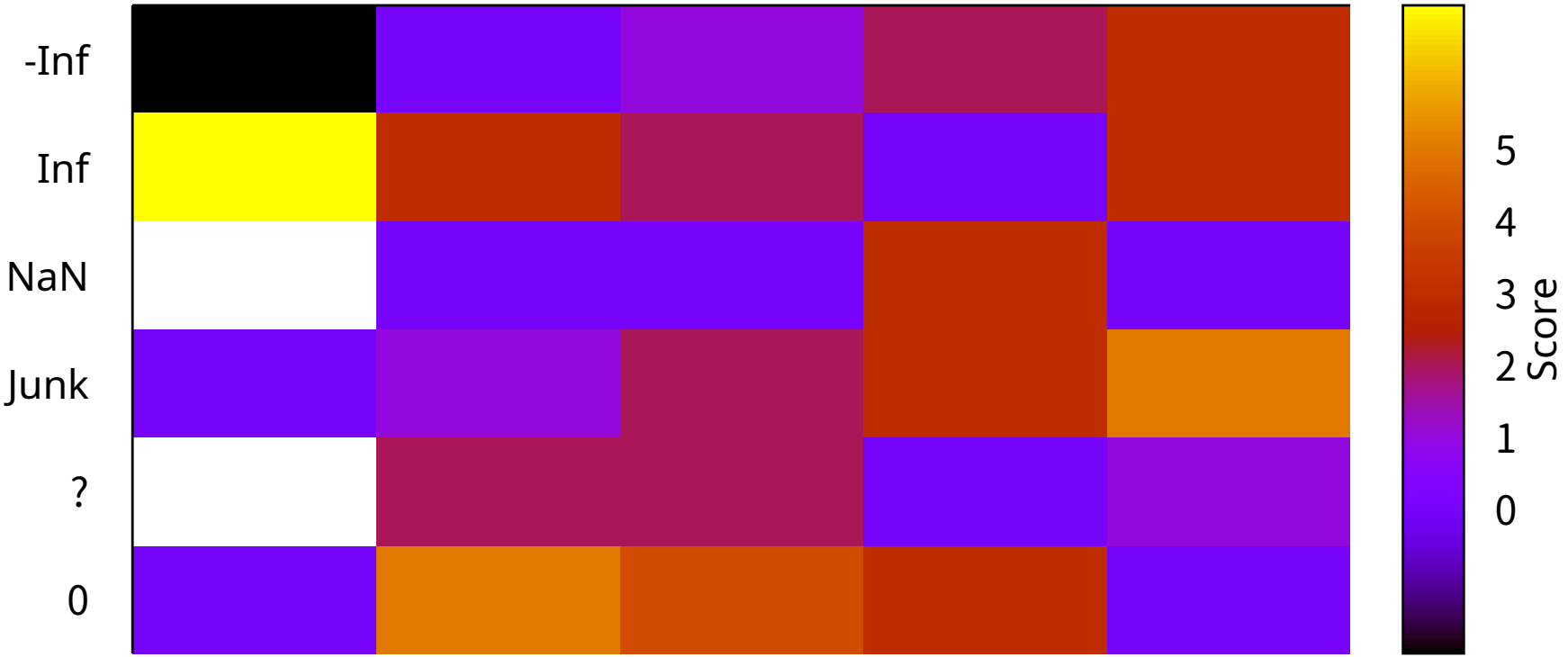
First column contains various odd values

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



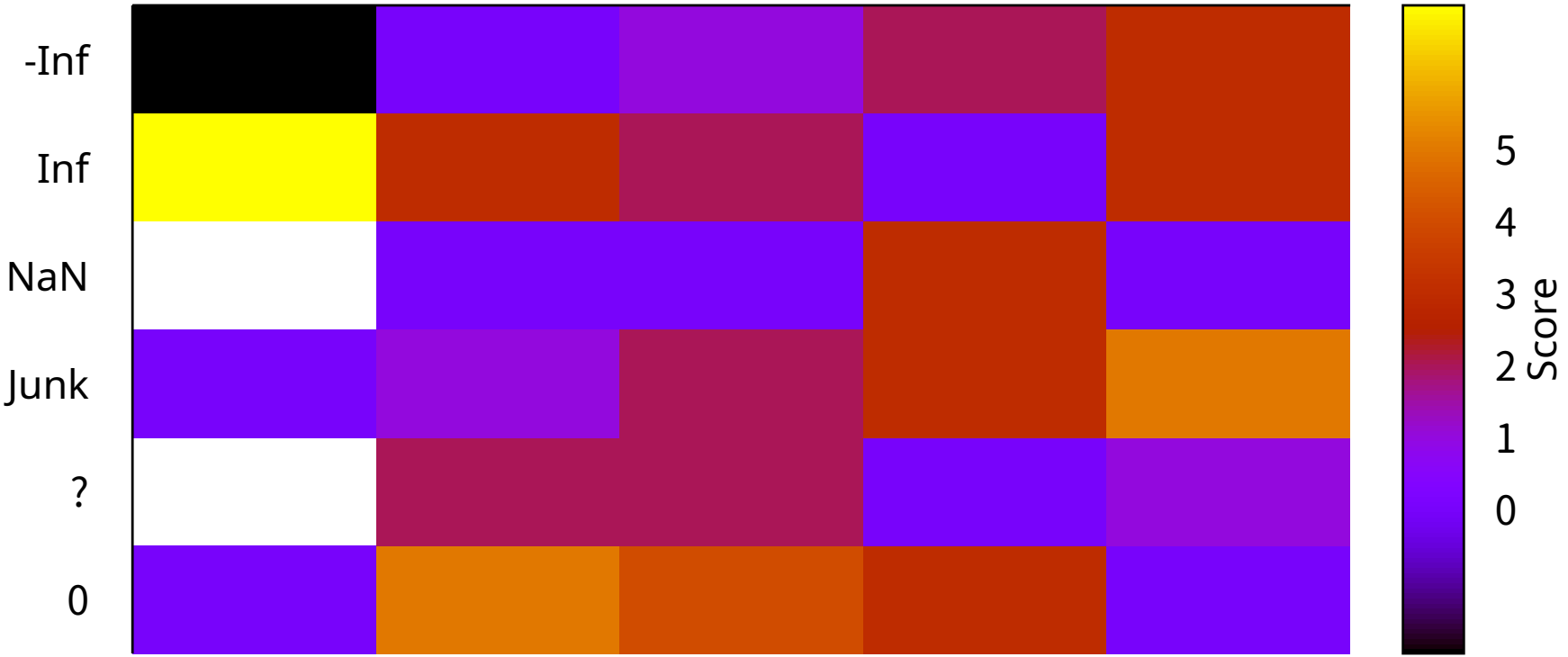
First column contains various odd values

Same thing in 3D mode



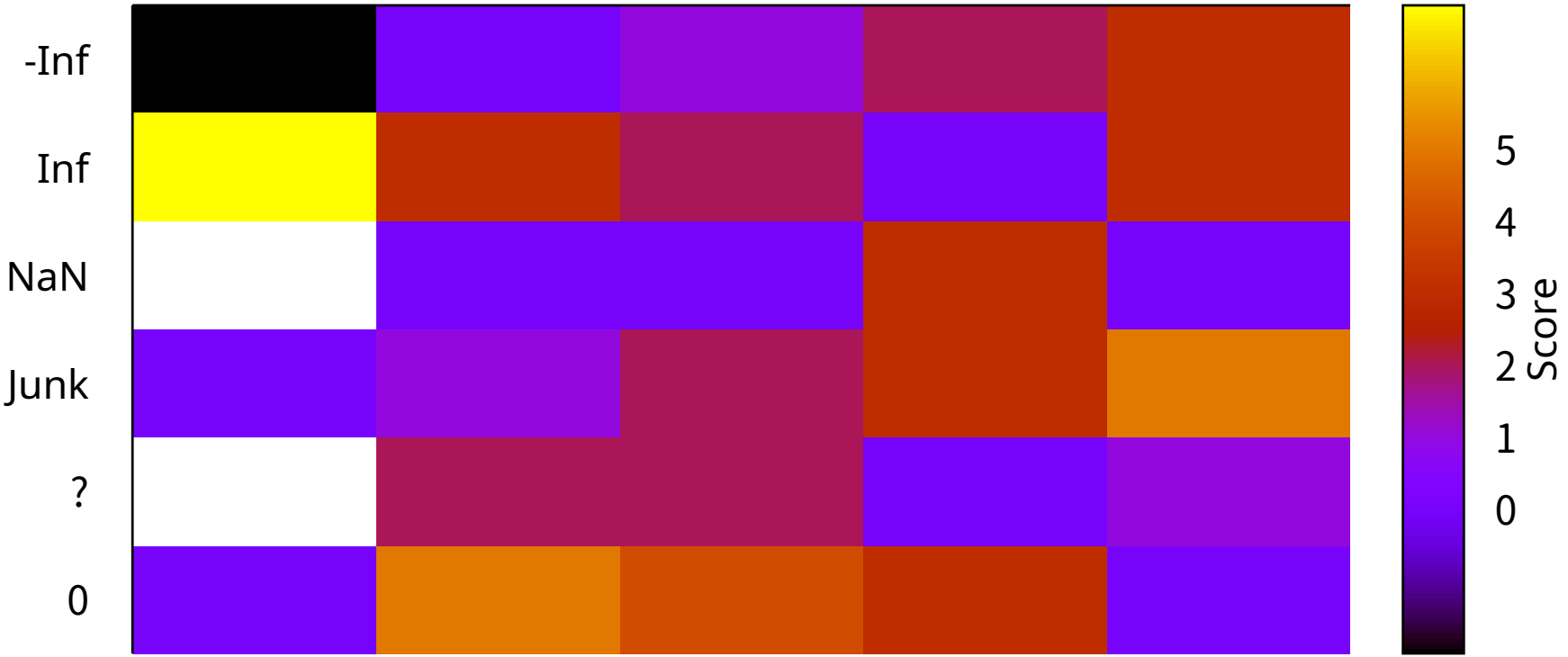
First column contains various odd values

Same thing in 'pixels' mode (3D)



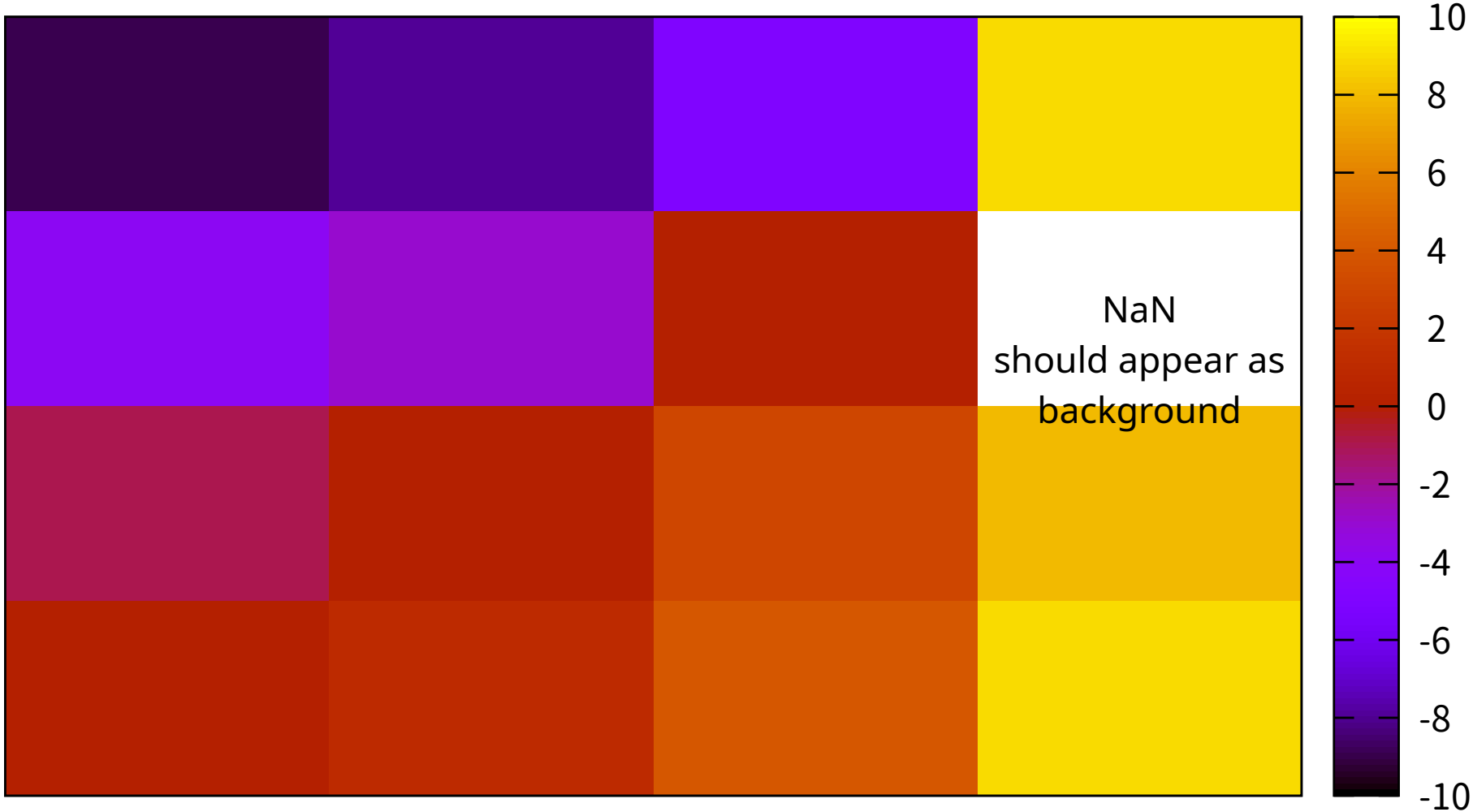
First column contains various odd values

3D image with pixel value in 4th column

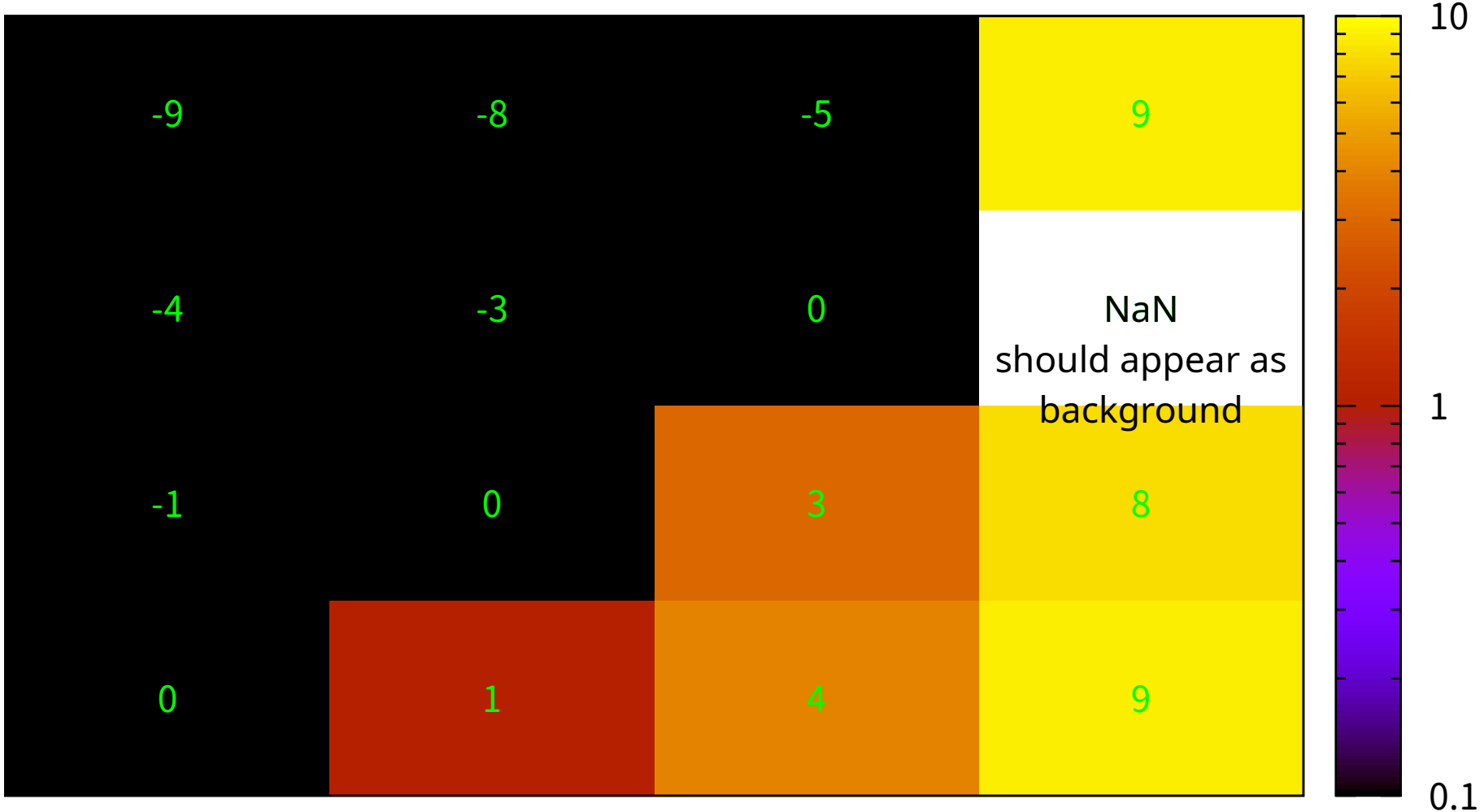


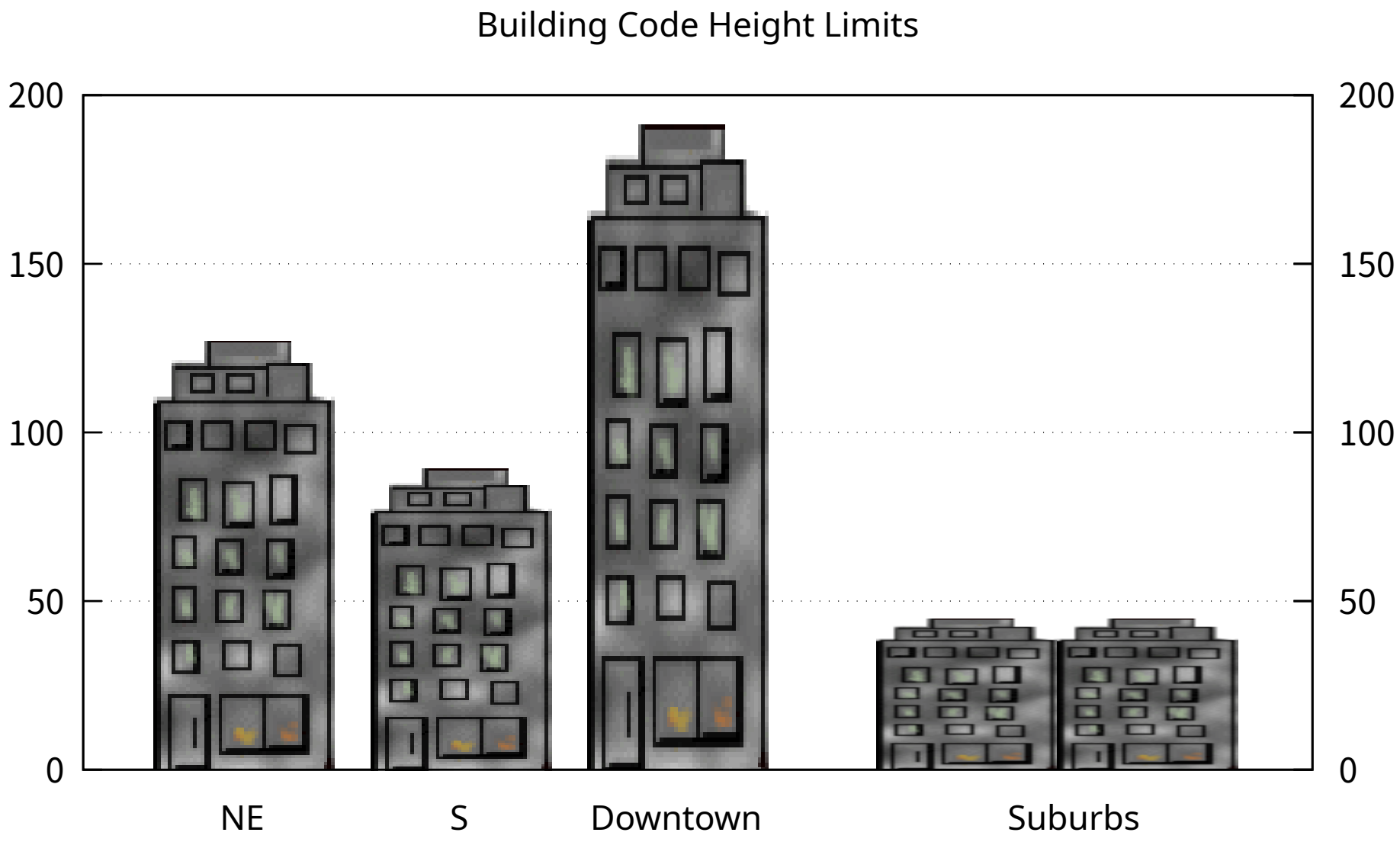
First column contains various odd values

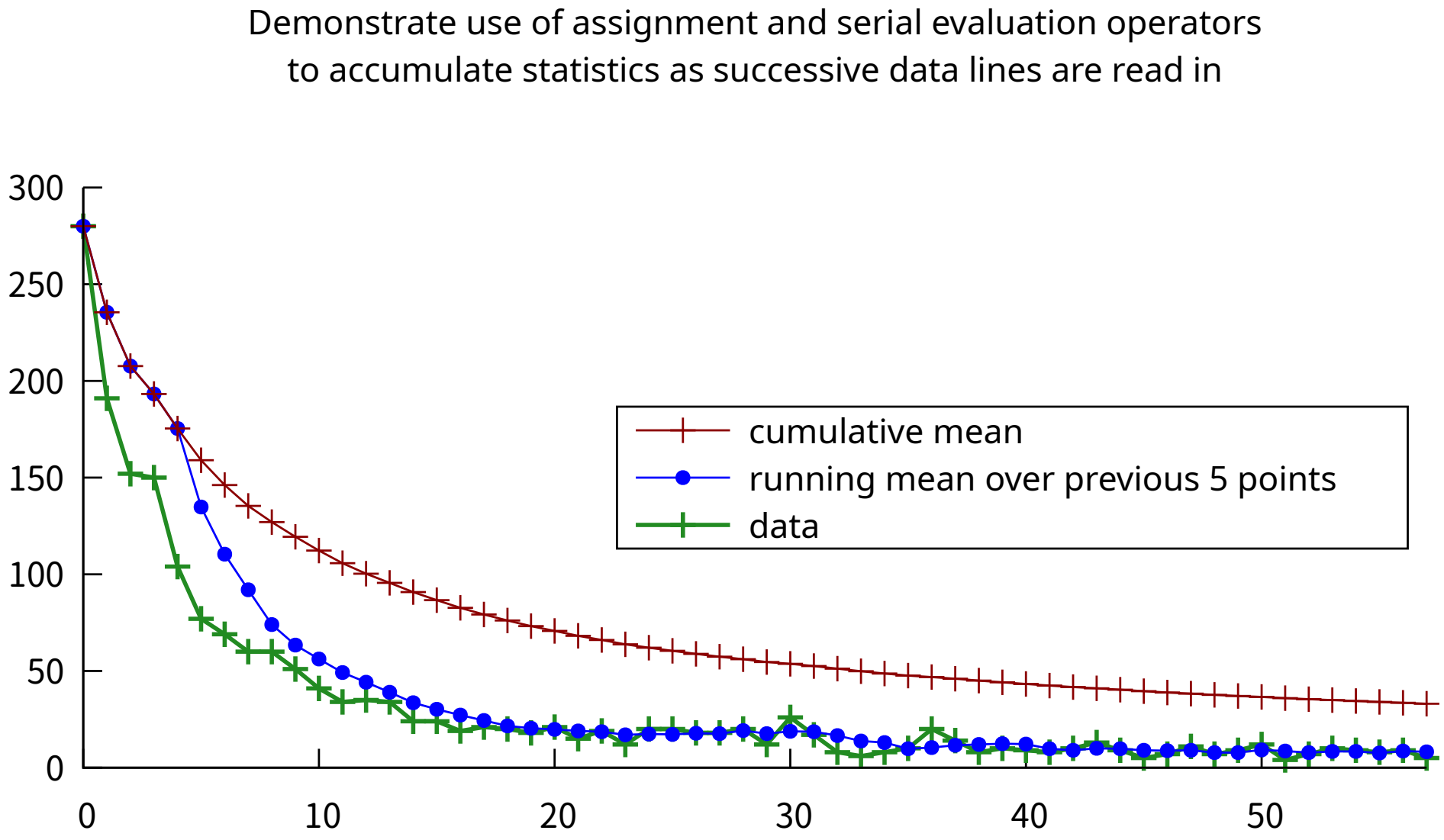
image from non-matrix data



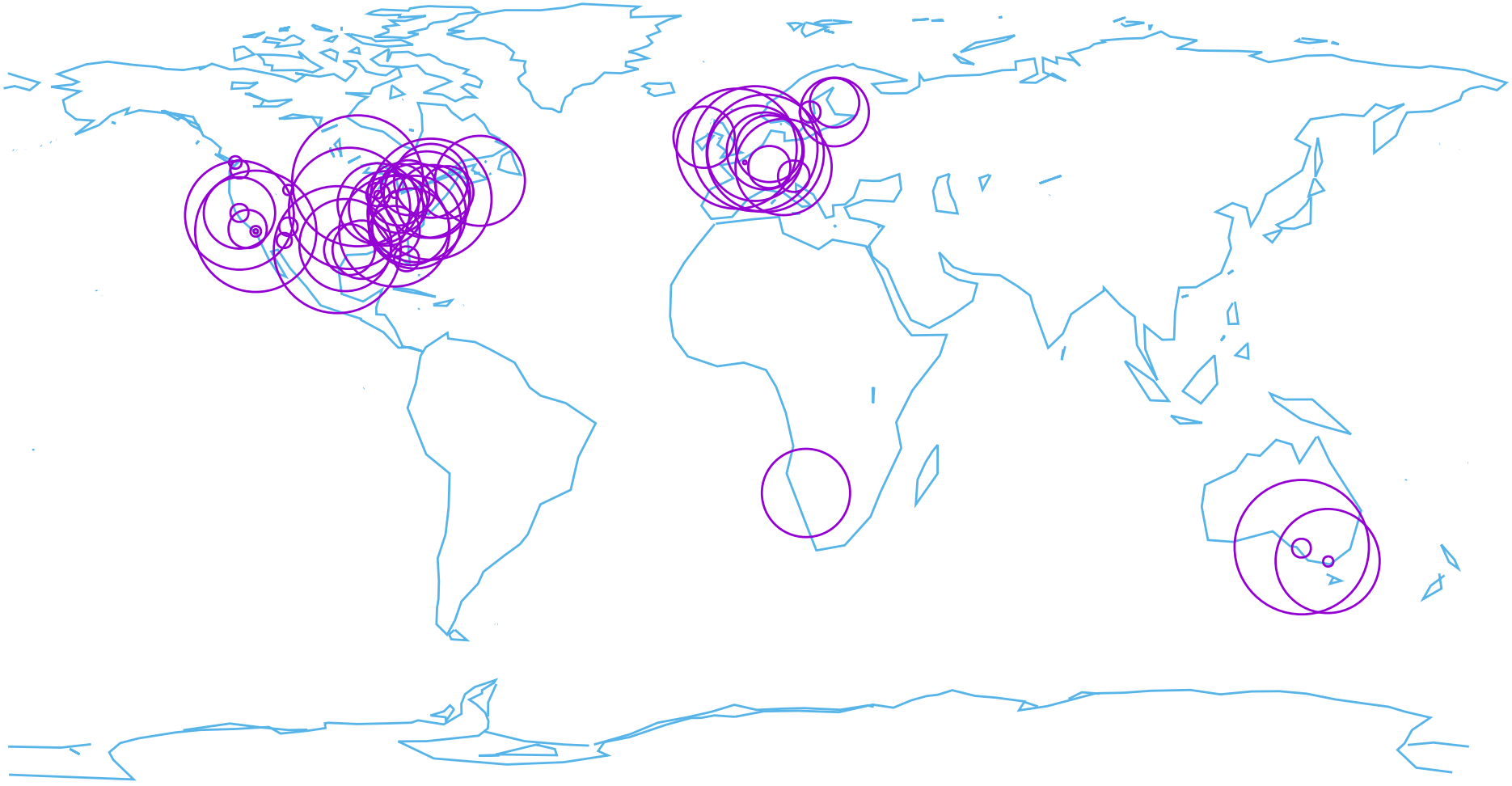
negative values mapped to log-scale colorbar

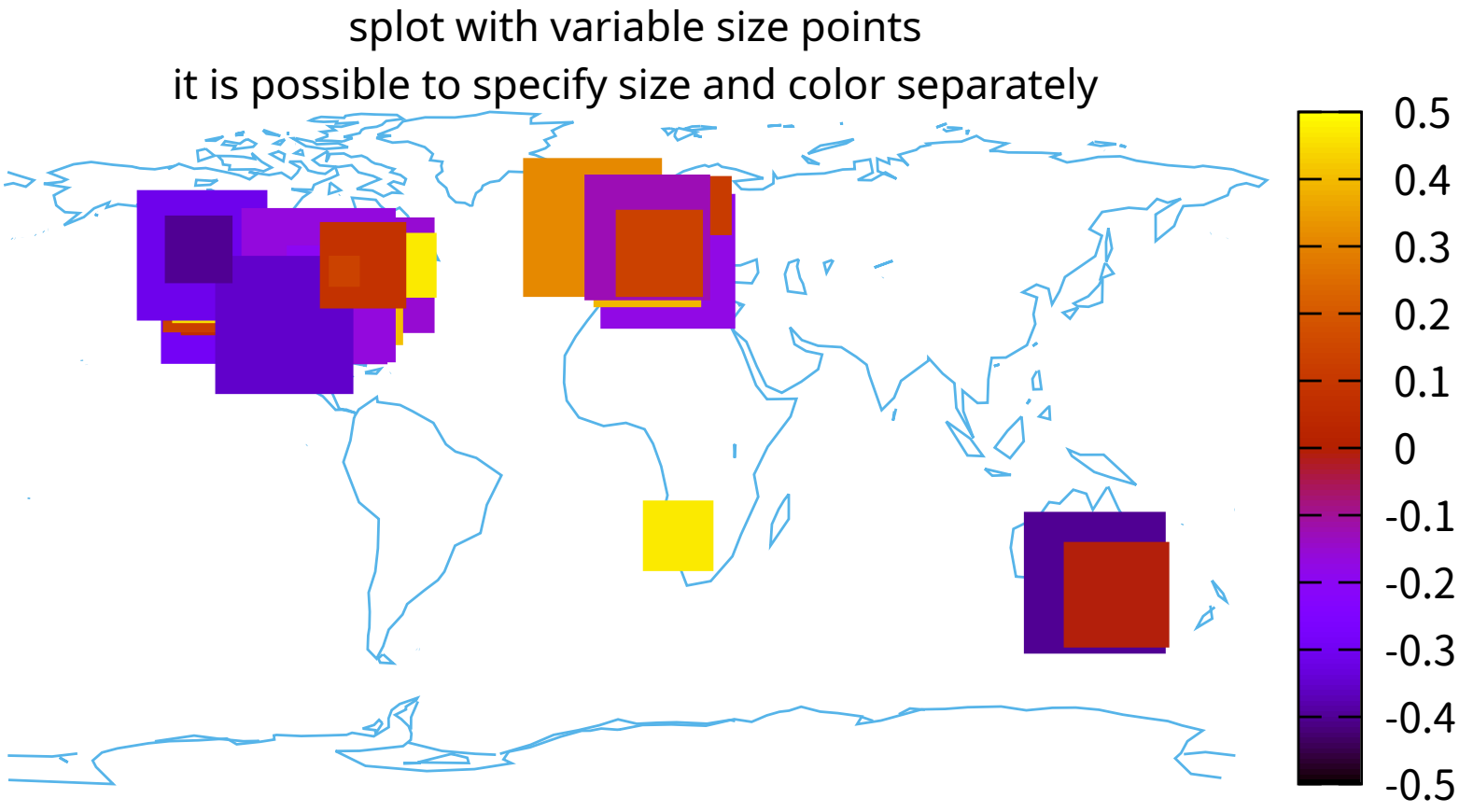



Building Code Height Limits
200
150
100
50
0
NE
S
Downtown
Suburbs

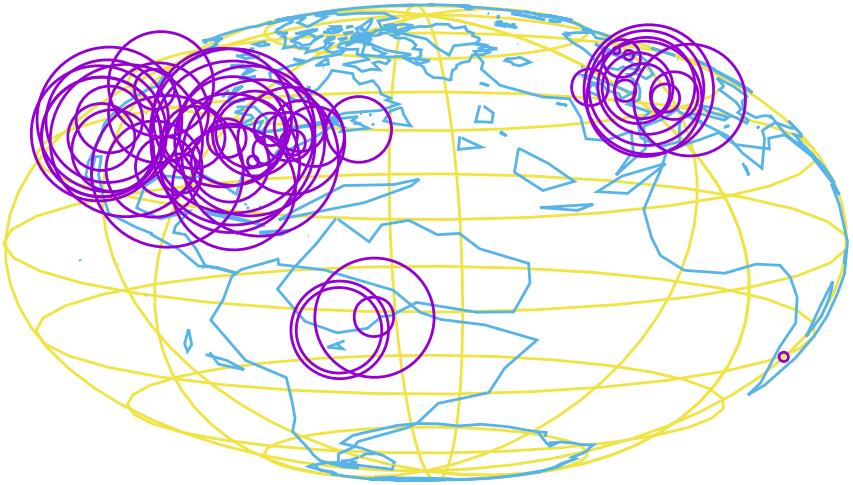

Demonstrate use of assignment and serial evaluation operators
to accumulate statistics as successive data lines are read in
300
250
200
150
100
50
0
0
10
20
30
40
50
cumulative mean
running mean over previous 5 points
data

plot with variable size points

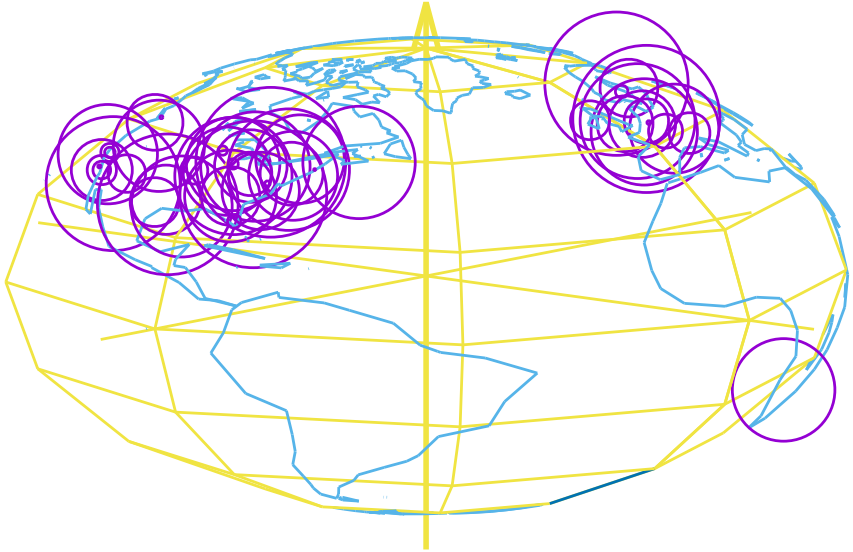



splot with variable size points
it is possible to specify size and color separately
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5

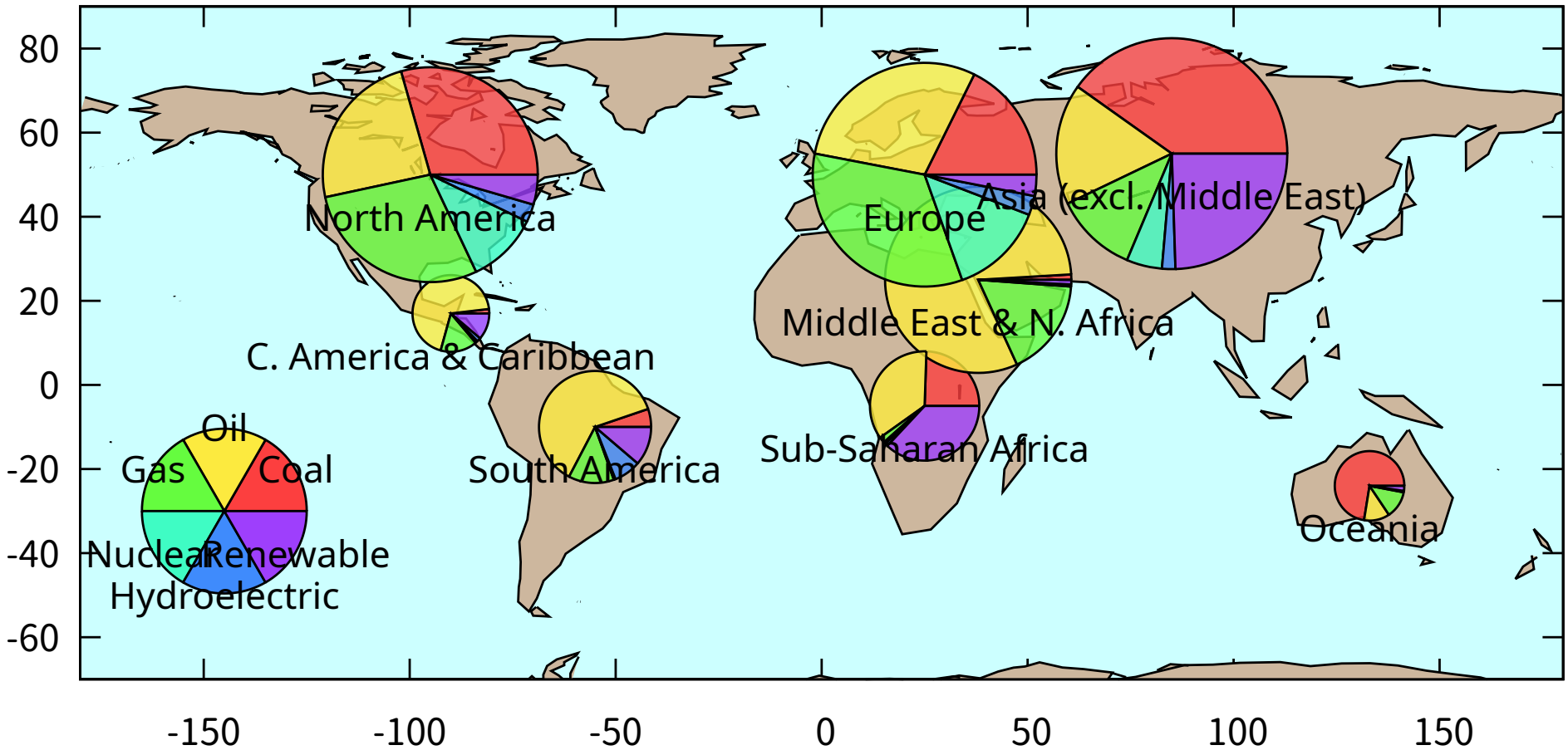
3D version using spherical coordinate system



3D solid version through hiddenlining

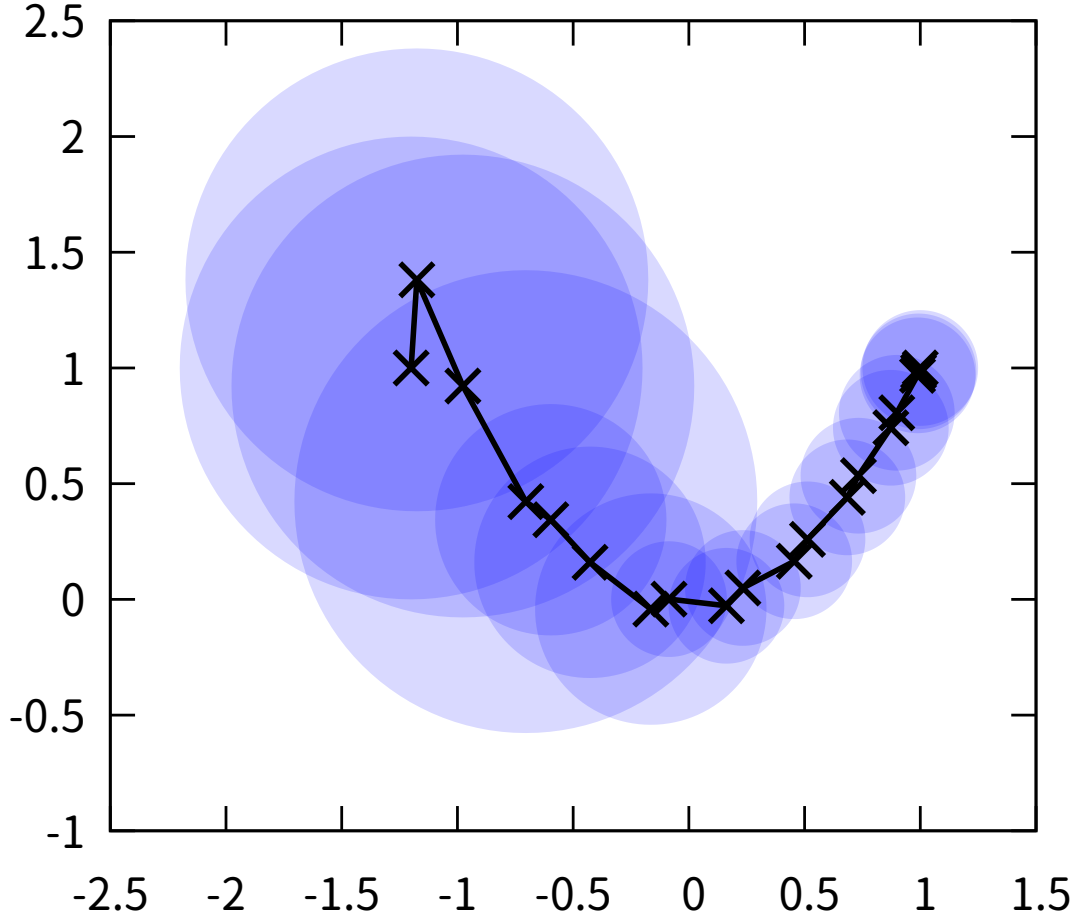


Sources of energy production, plotted for each continent



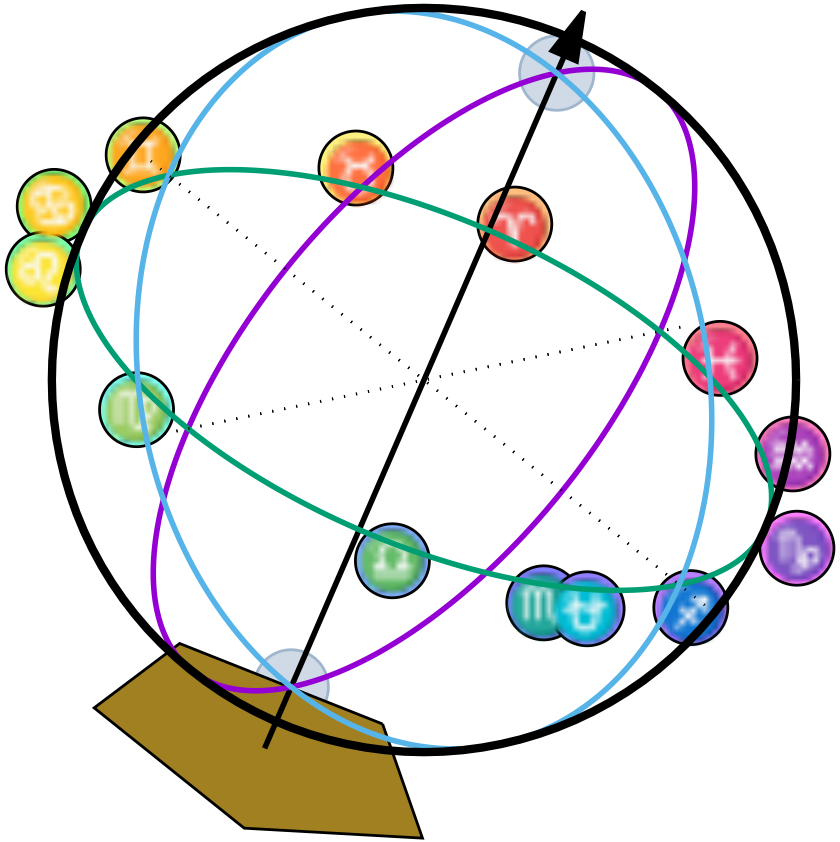
Dynamically generated pie charts

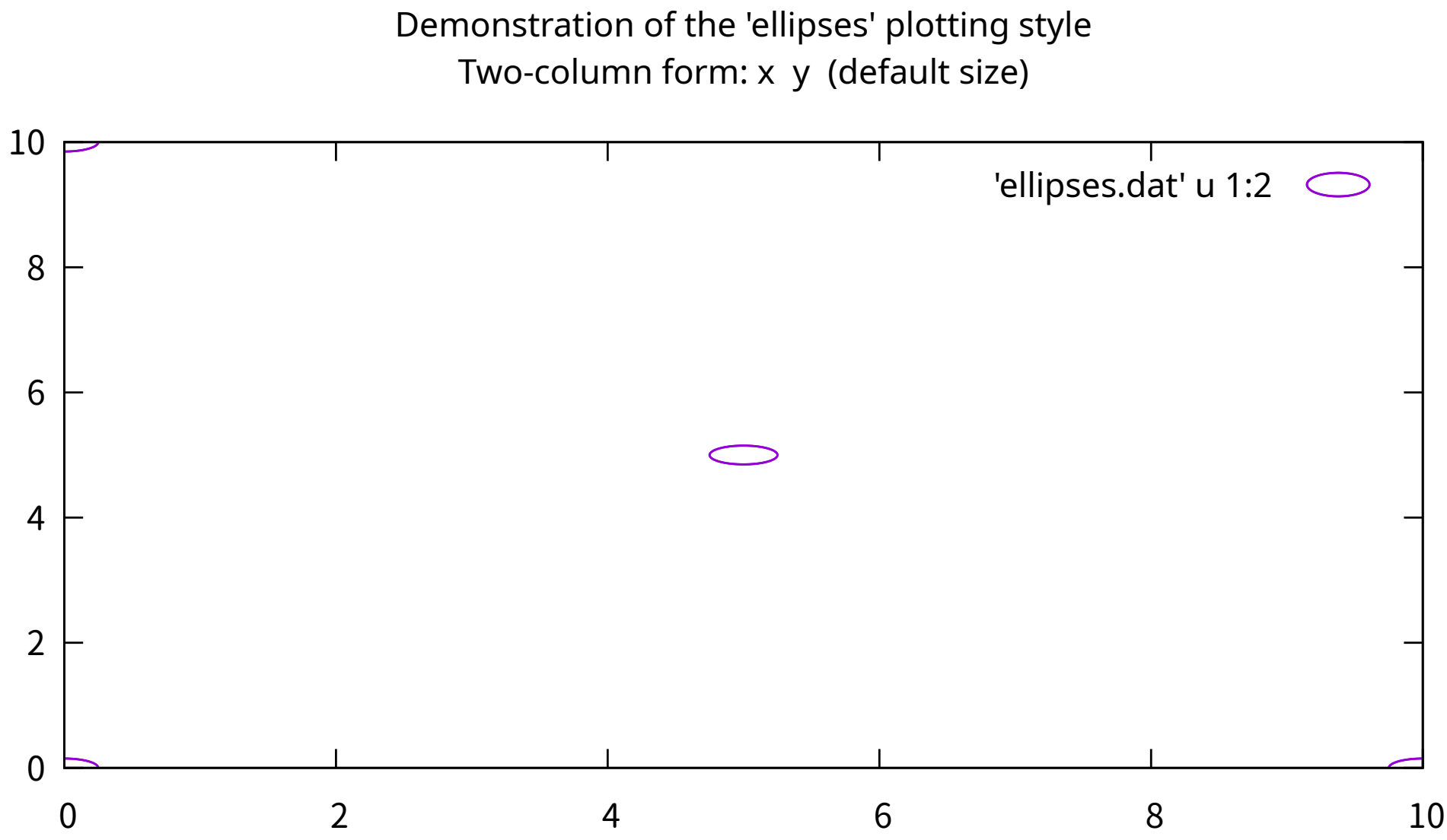
Trace of unconstrained optimization with trust-region method

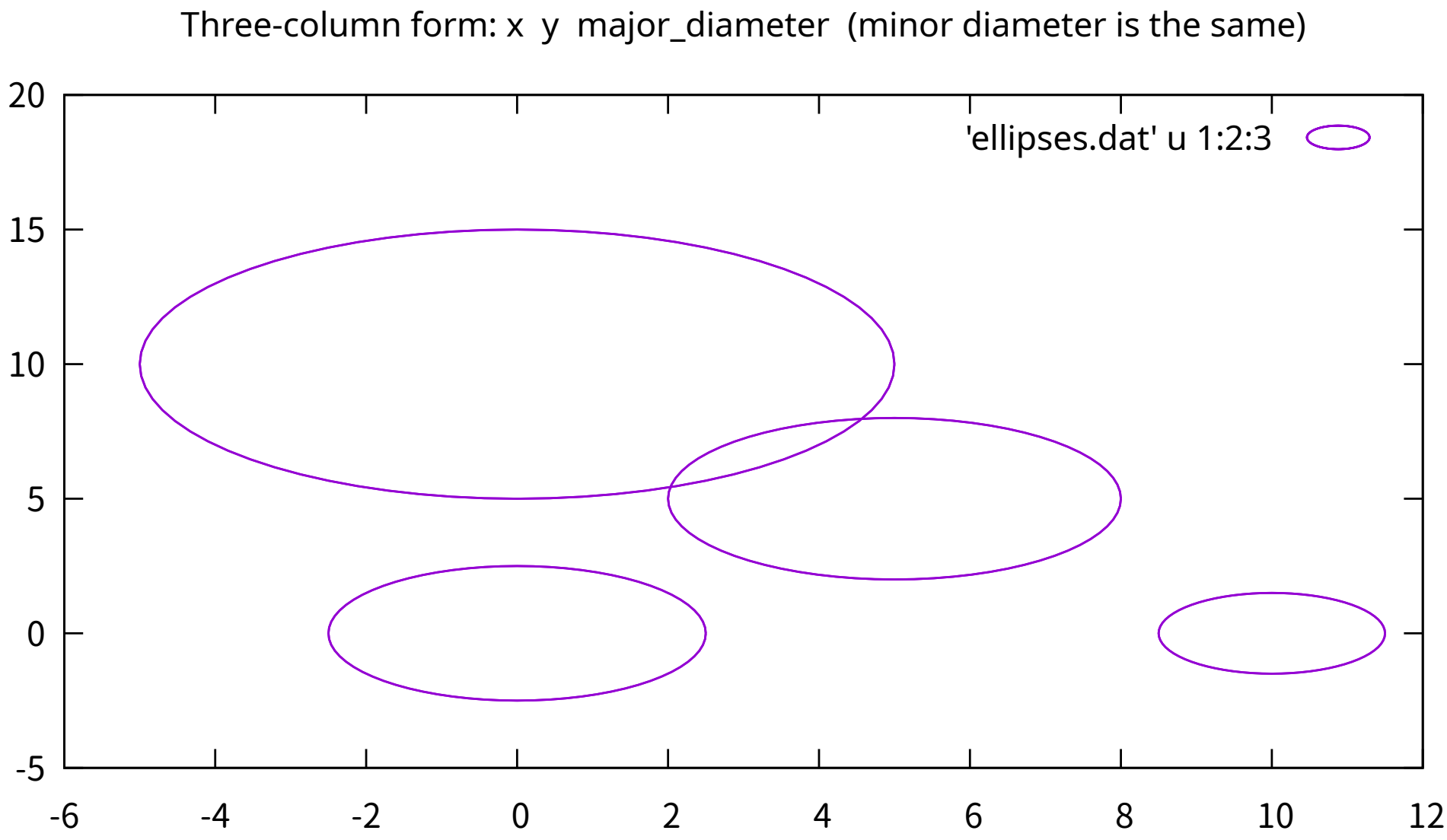


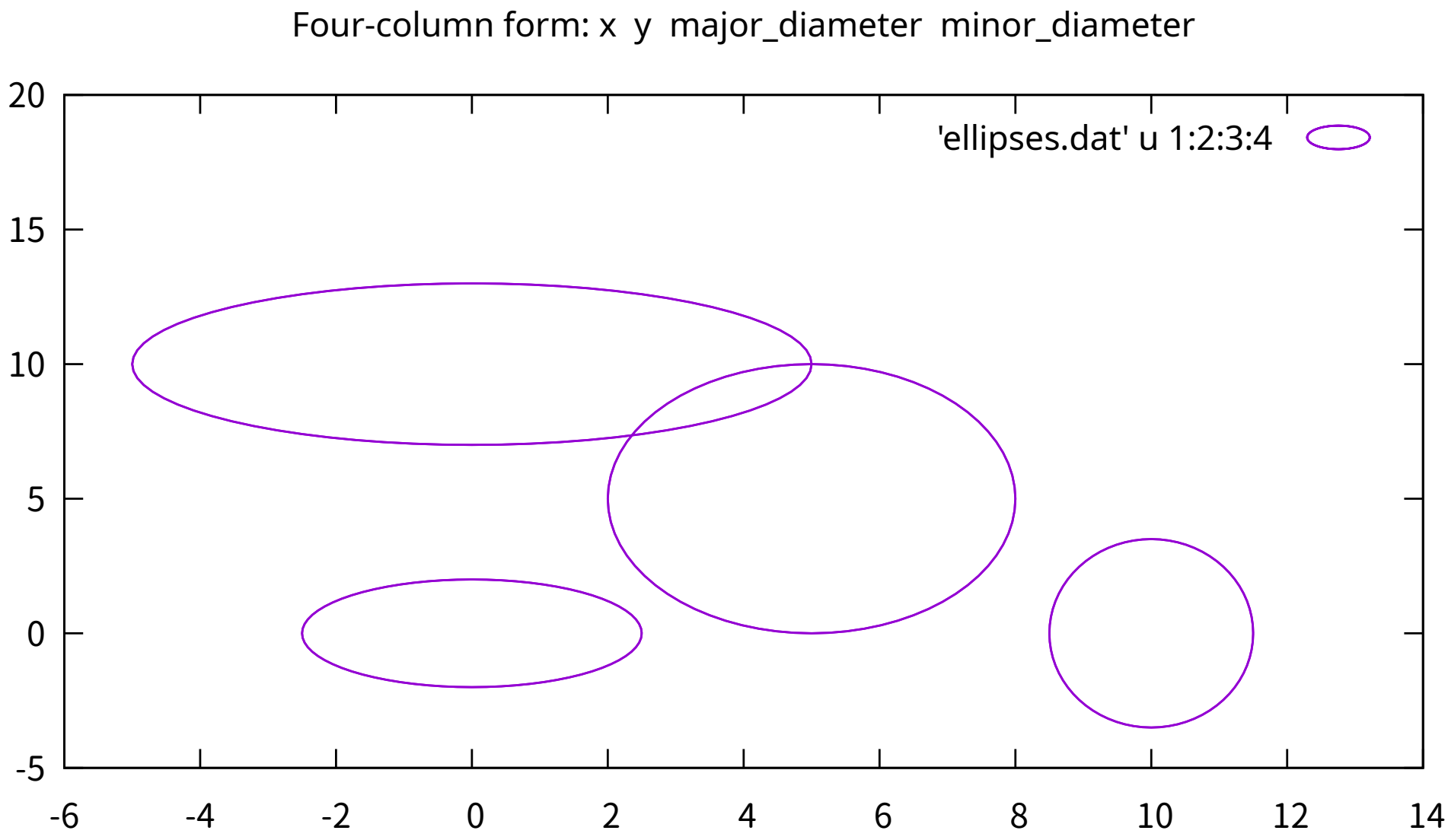
Note that overlapping transparent circles produce a darker area

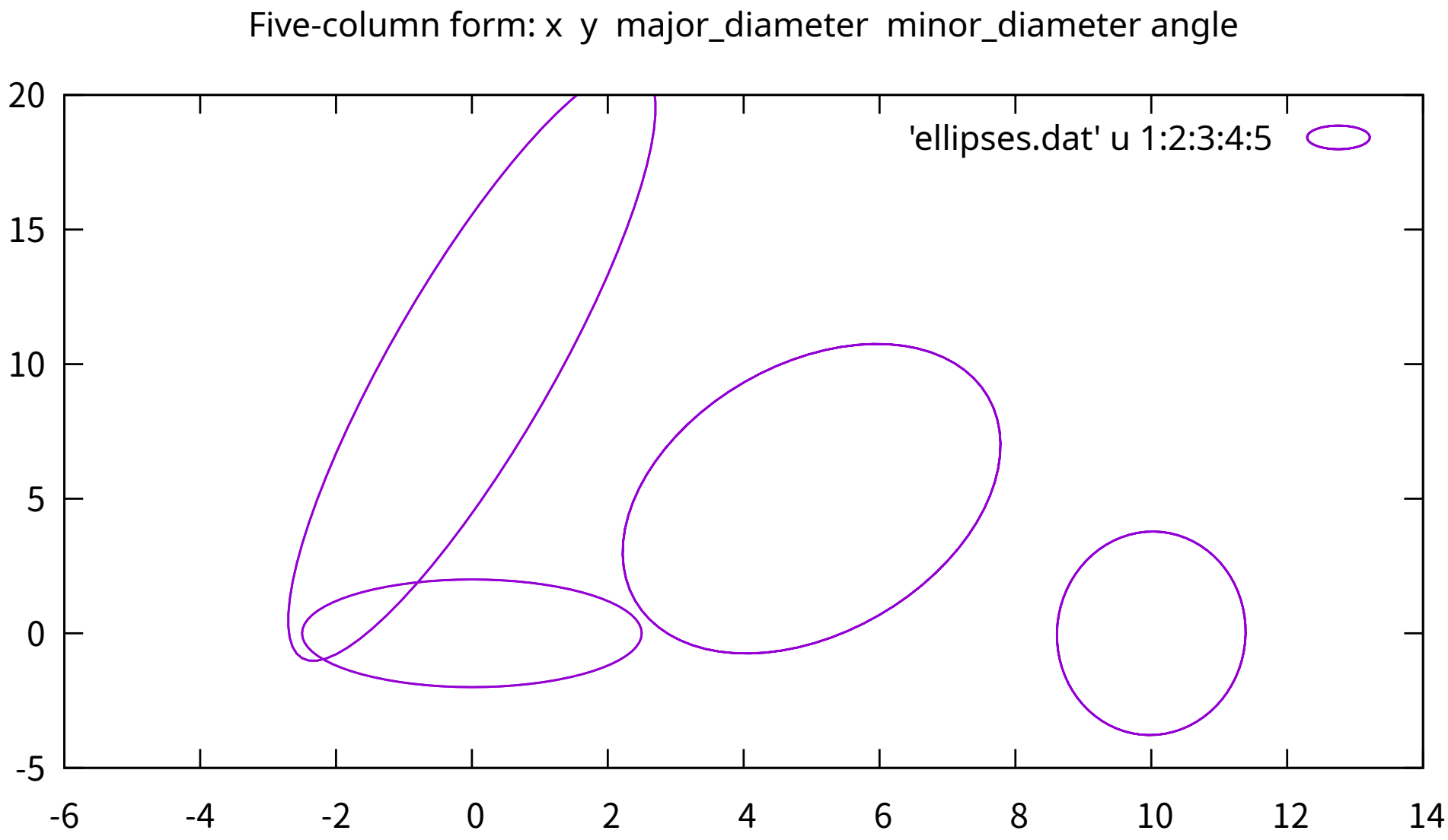
Circle and polygon objects in 3D

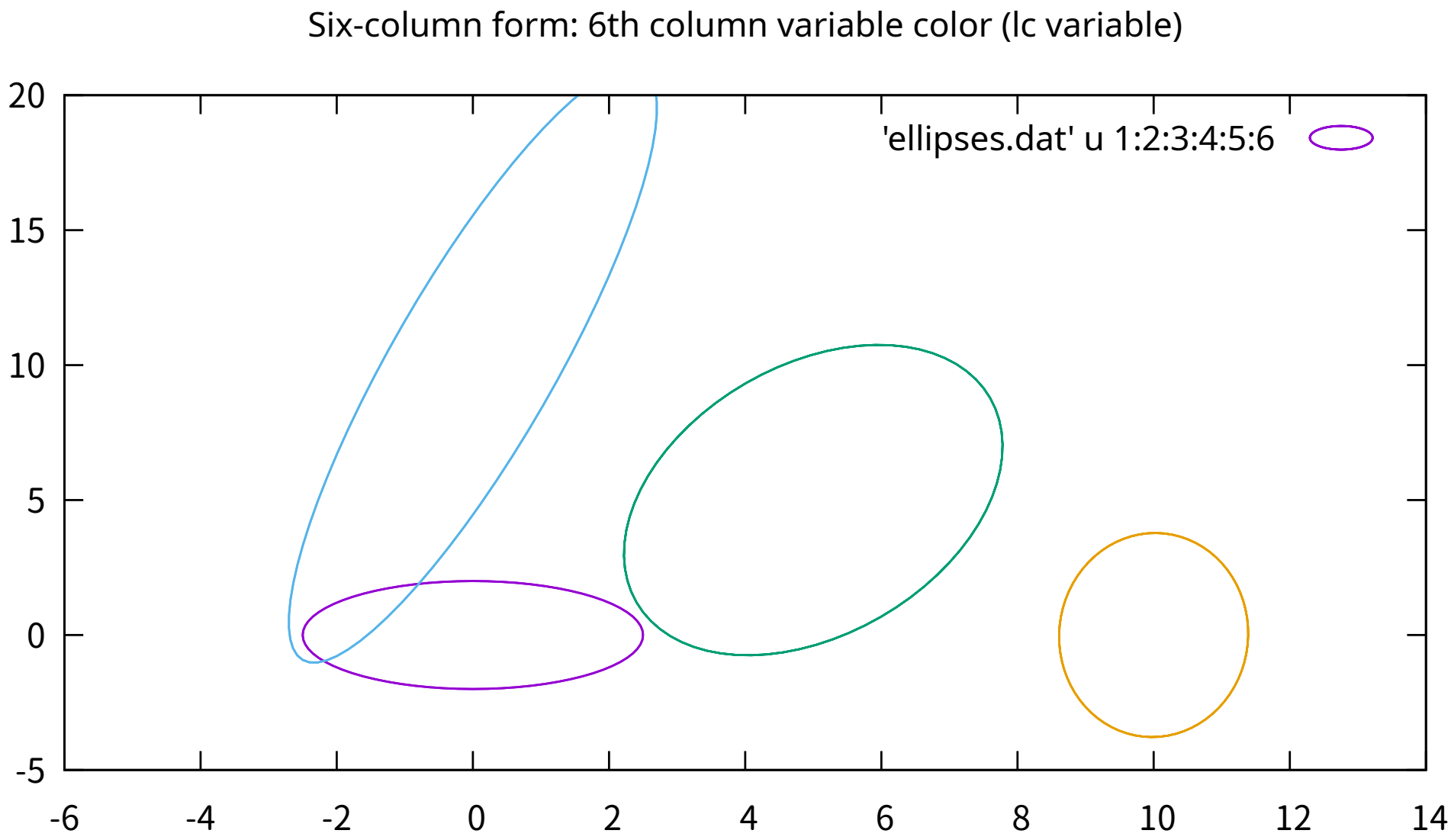


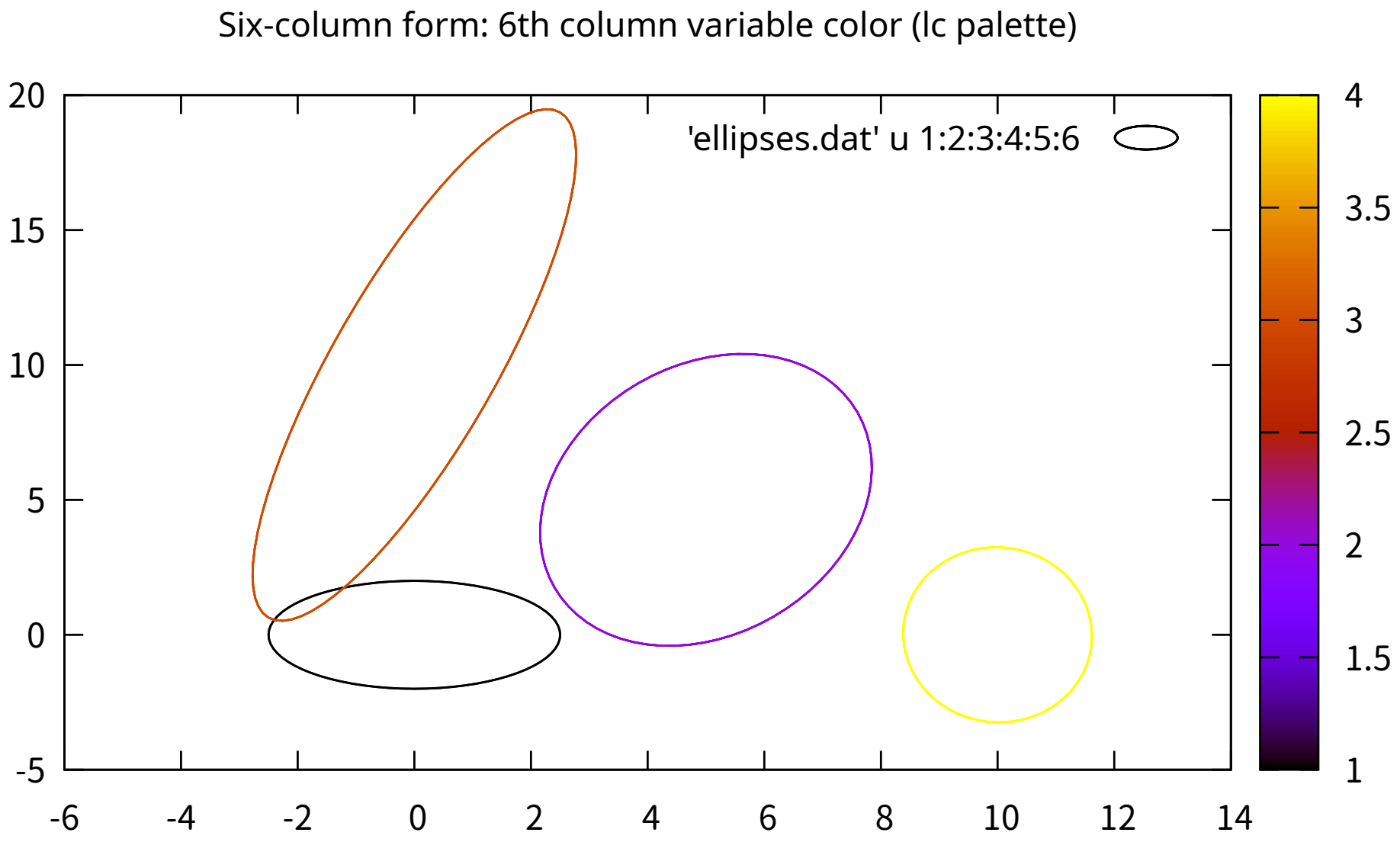

Demonstration of the 'ellipses' plotting style
Two-column form: x y (default size)
'ellipses.dat' u 1:2
0
2
4
6
8
10
0
2
4
6
8
10

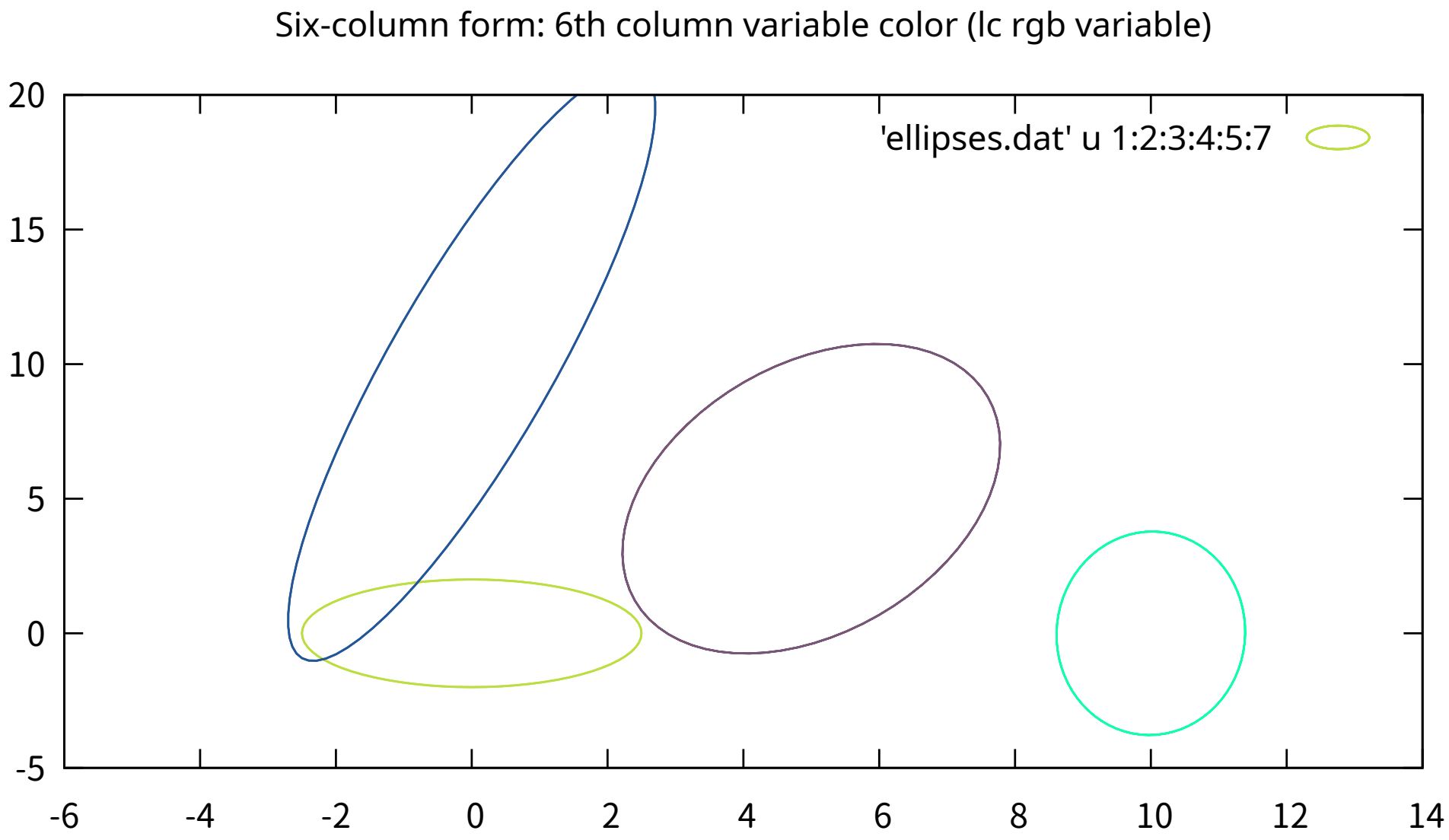

Three-column form: x y major_diameter (minor diameter is the same)
'ellipses.dat' u 1:2:3
20
15
10
5
0
-5
-6
-4
-2
0
2
4
6
8
10
12

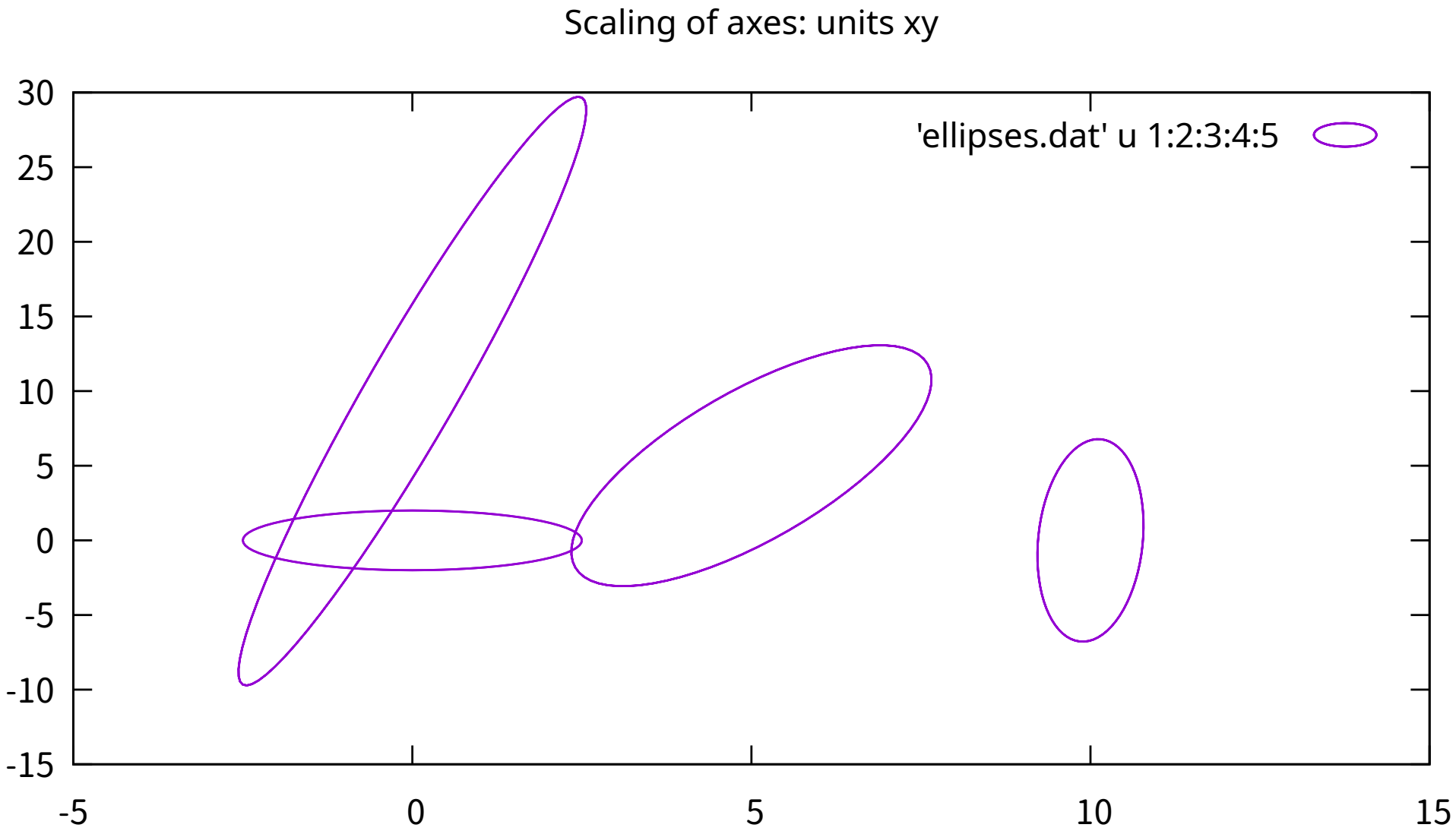

Four-column form: x y major_diameter minor_diameter
'ellipses.dat' u 1:2:3:4
20
15
10
5
0
-5
-6
-4
-2
0
2
4
6
8
10
12
14

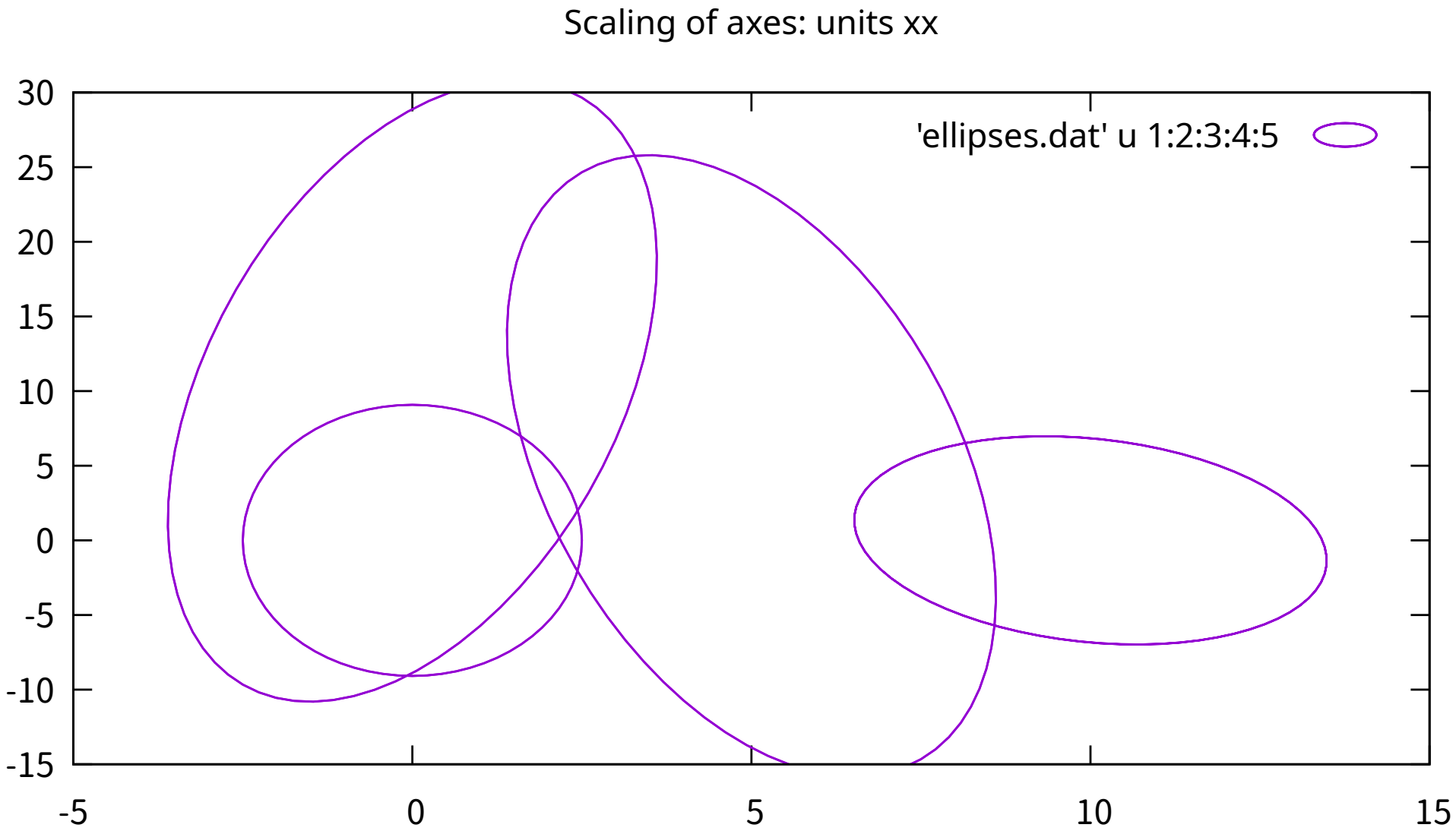

Five-column form: x y major_diameter minor_diameter angle
'ellipses.dat' u 1:2:3:4:5
20
15
10
5
0
-5
-6
-4
-2
0
2
4
6
8
10
12
14

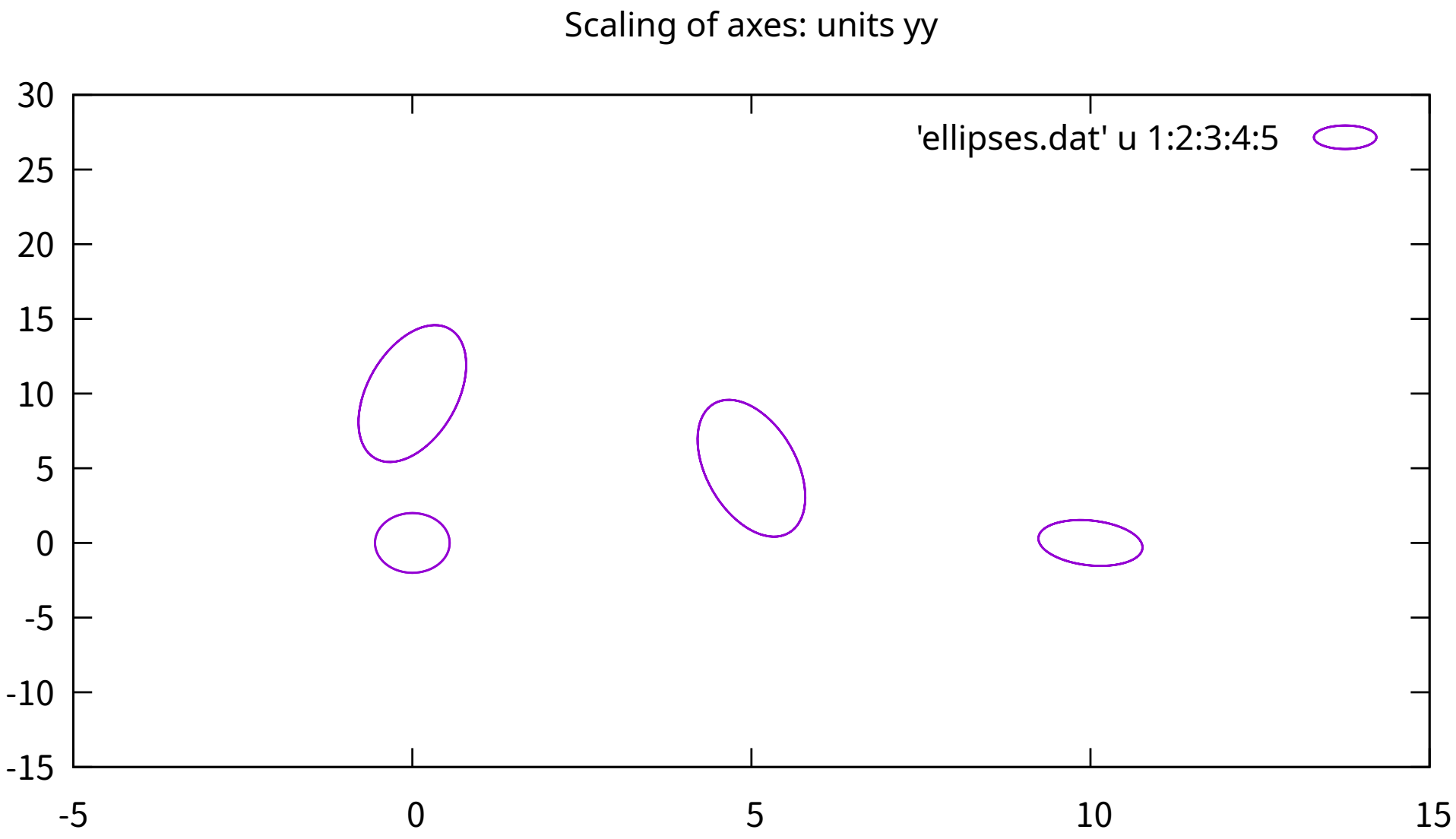

Six-column form: 6th column variable color (lc variable)
'ellipses.dat' u 1:2:3:4:5:6
20
15
10
5
0
-5
-6
-4
-2
0
2
4
6
8
10
12
14

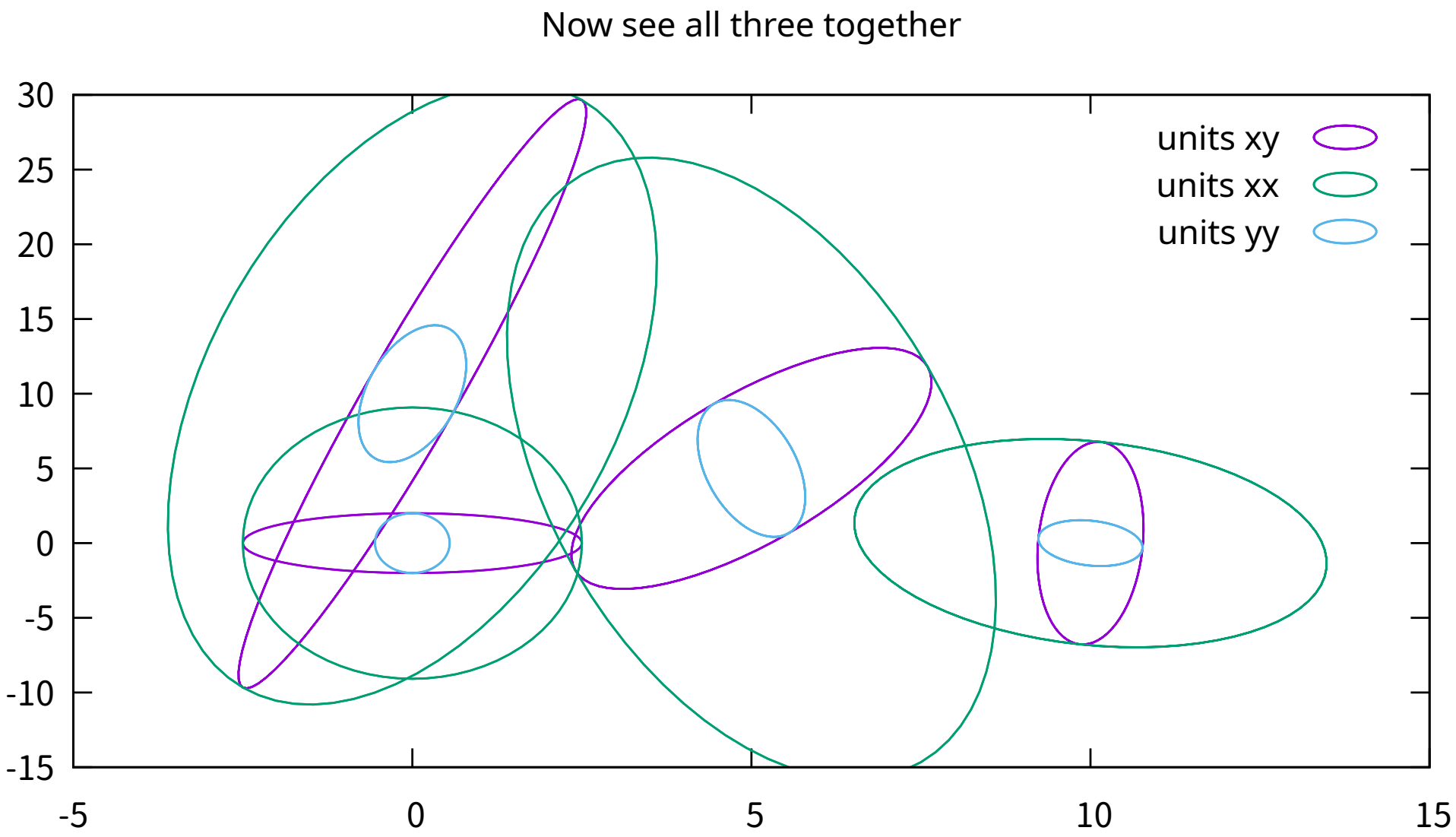

Six-column form: 6th column variable color (lc palette)
'ellipses.dat' u 1:2:3:4:5:6
20
15
10
5
0
-5
-6
-4
-2
0
2
4
6
8
10
12
14
4
3.5
3
2.5
2
1.5
1


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

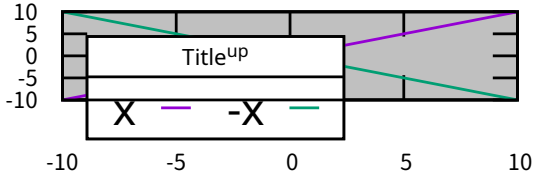

Scaling of axes: units xy
'ellipses.dat' u 1:2:3:4:5
30
25
20
15
10
5
0
-5
-10
-15
-5
0
5
10
15


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

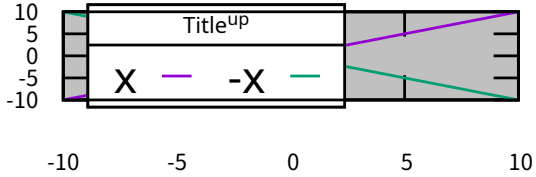

Scaling of axes: units yy
'ellipses.dat' u 1:2:3:4:5
30
25
20
15
10
5
0
-5
-10
-15
-5
0
5
10
15


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

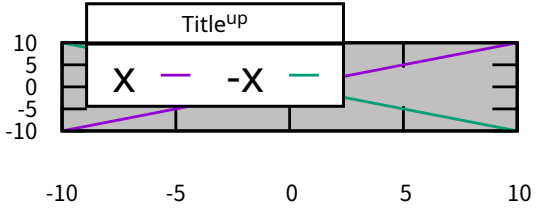
Key (ins vert left top)



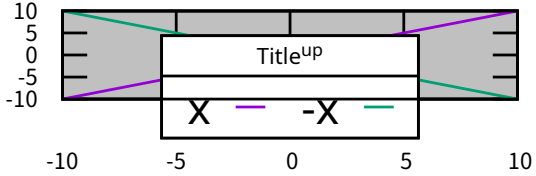
Key (ins vert center left)



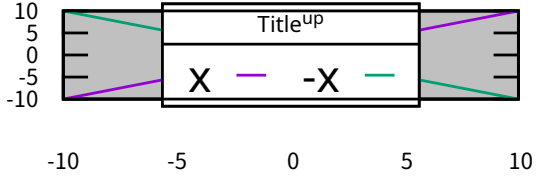
Key (ins vert bot left)



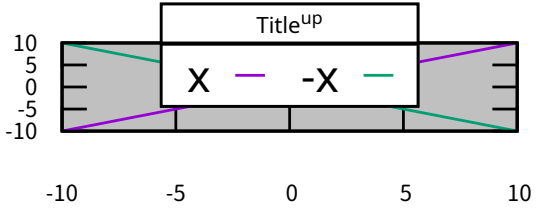
Key (ins vert center top)



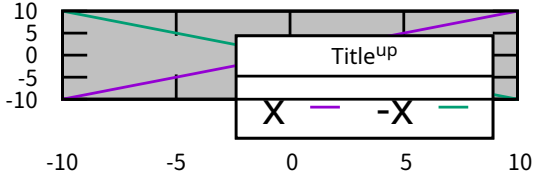
Key (inside vertical center)



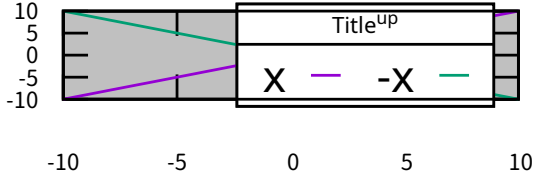
Key (ins vert bot center)



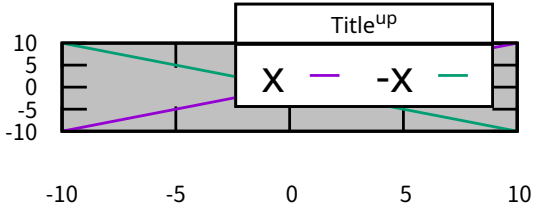
Key (ins vert right top)



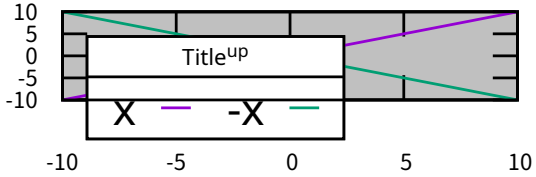
Key (ins vert cent right)



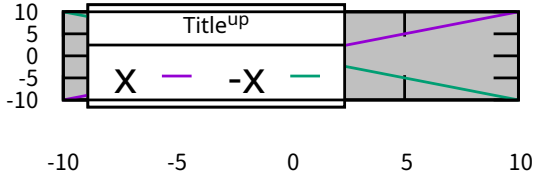
Key (ins vert bot right)



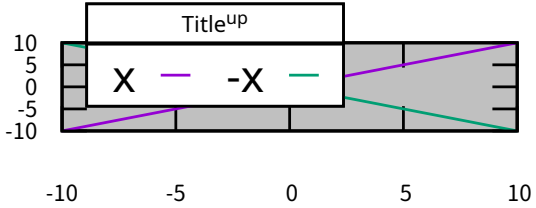
Key (ins horiz left top)



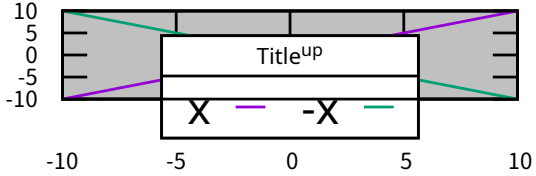
Key (ins horiz center left)



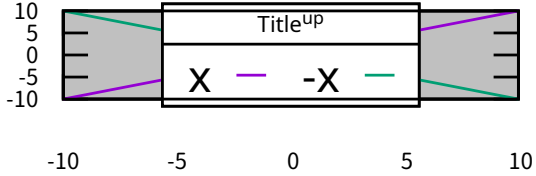
Key (ins horiz bot left)



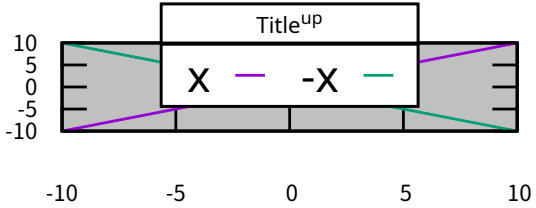
Key (ins horiz center top)



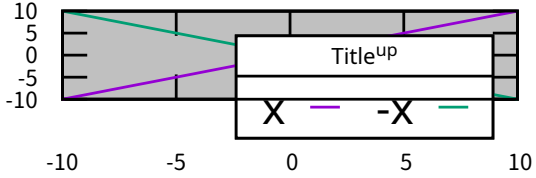
Key (inside horizontal center)



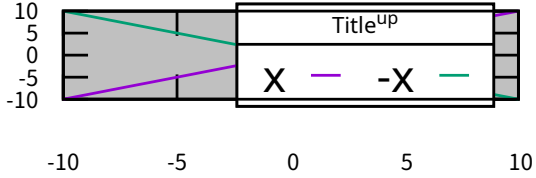
Key (ins horiz bot center)



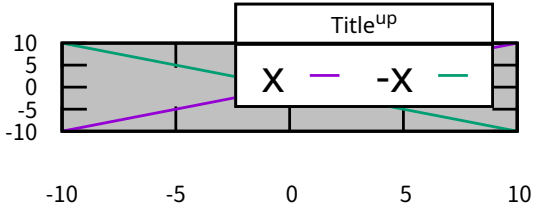
Key (ins horiz right top)



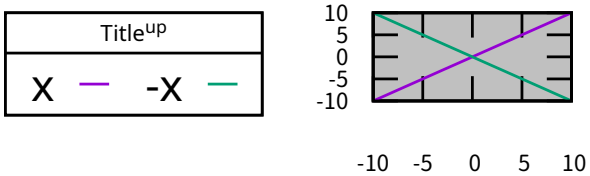
Key (ins horiz cent right)



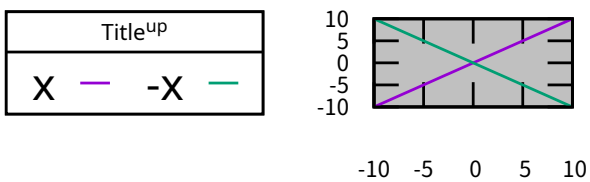
Key (ins horiz bot right)



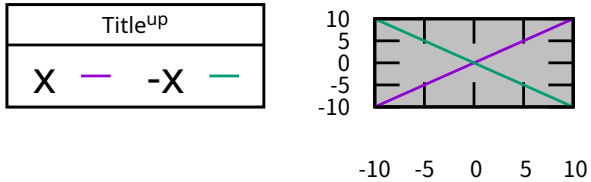
Key (out vert left top)



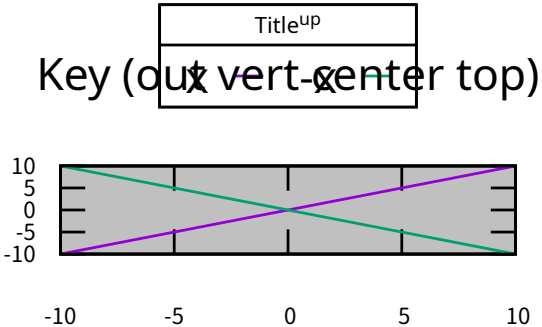
Key (out vert center left)



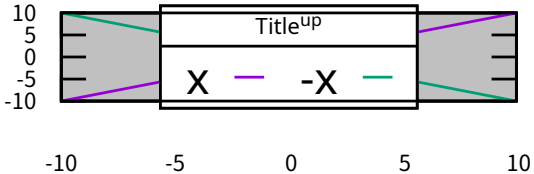
Key (out vert bot le



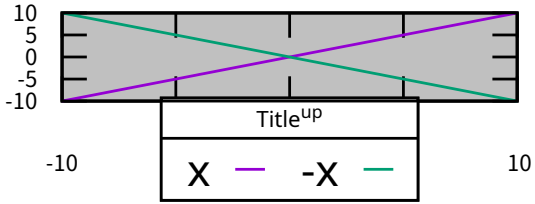
Key (out vert center top)



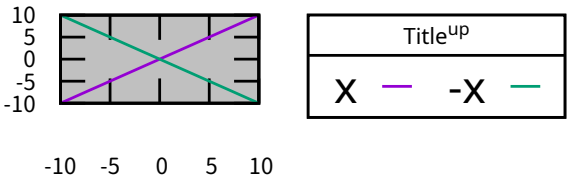
Key (outside vertical center)



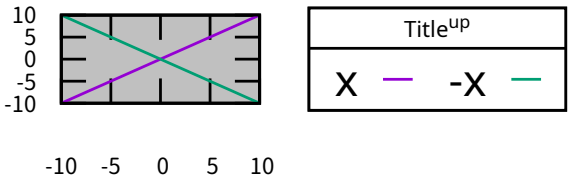
Key (out vert bot center)



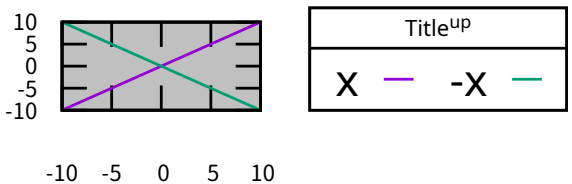
Key (out vert right top)



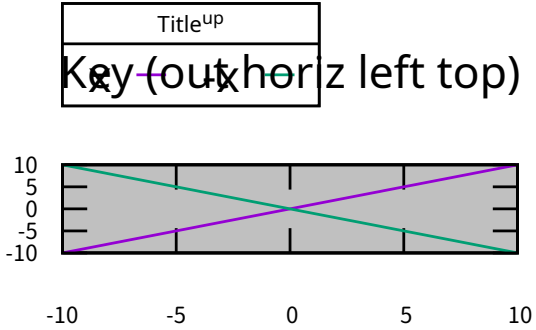
Key (out vert cent right)



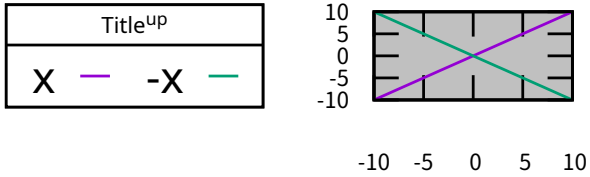
Key (out vert bot right)



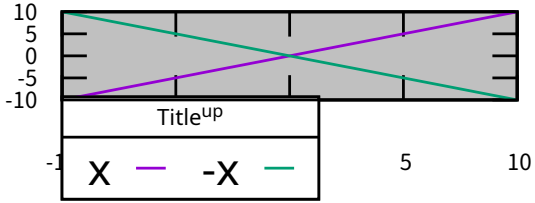
Key (out horiz left top)



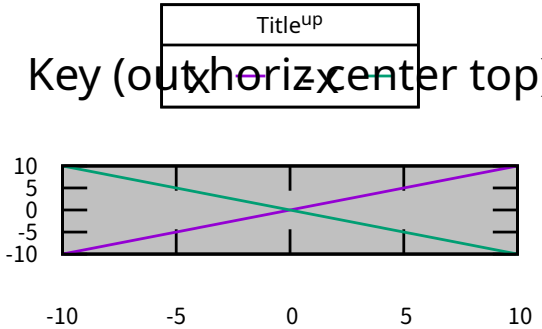
Key (out horiz center left)



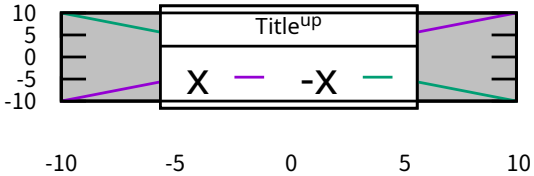
Key (out horiz bot left)



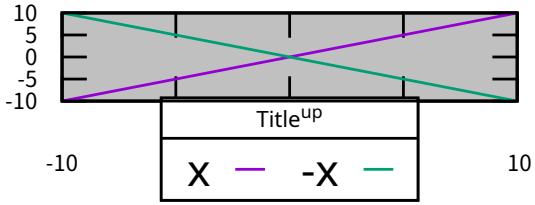
Key (out horiz center top)



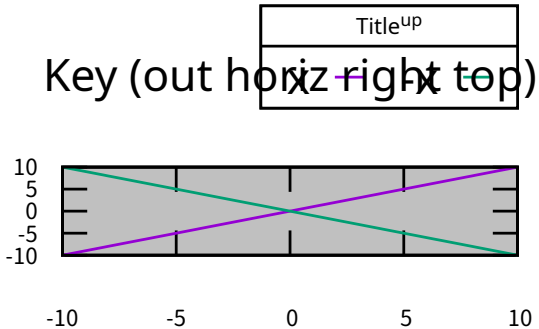
Key (outside horizontal center)



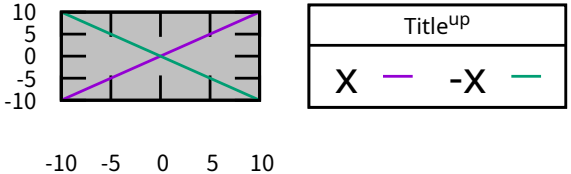
Key (out horiz bot center)



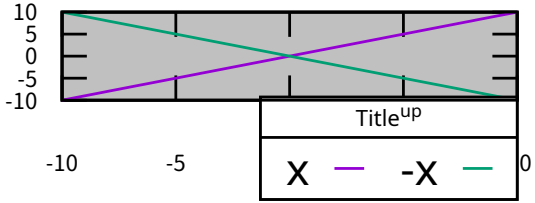
Key (out horiz right top)



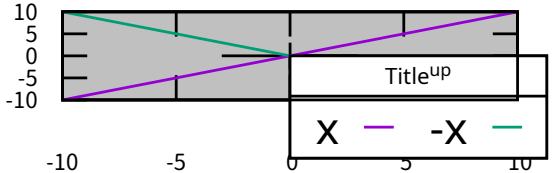
Key (out horiz cent right)



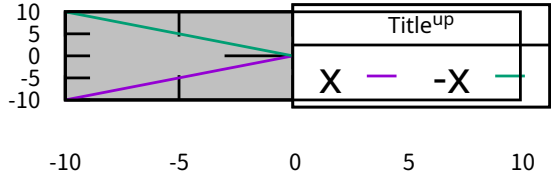
Key (out horiz bot right)



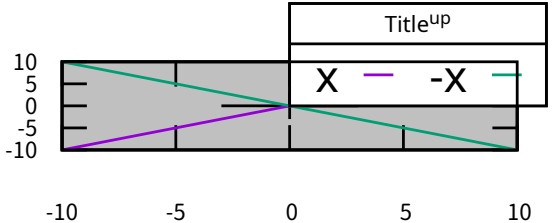
Key (<manual> vert left top)



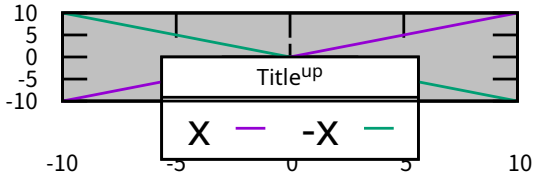
Key (<manual> vert center left)



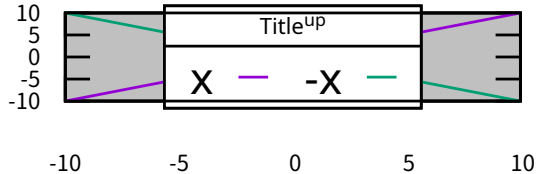
Key (<manual> vert bot left)



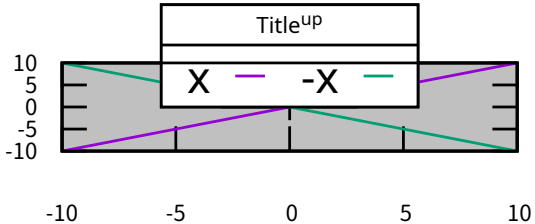
Key (<manual> vert center top)



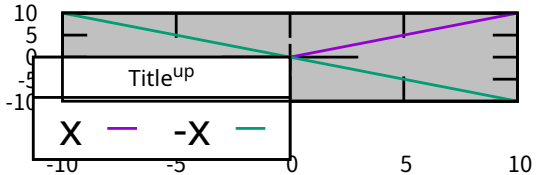
Key (<manual> vertical center)



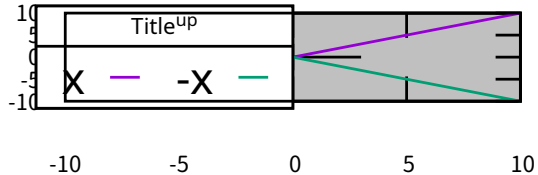
Key (<manual> vert bot center



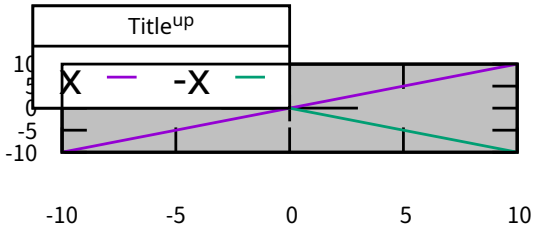
Key (<manual> vert right top)



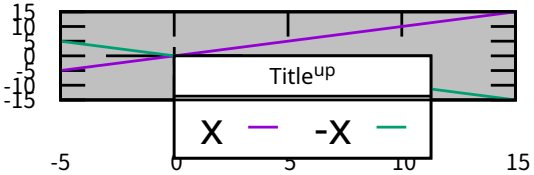
Key (<manual> vert cent right)



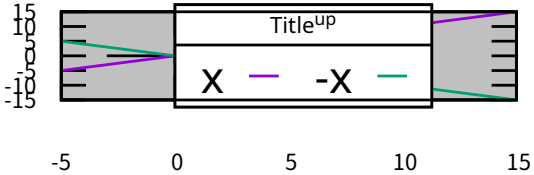
Key (<manual> vert bot right)



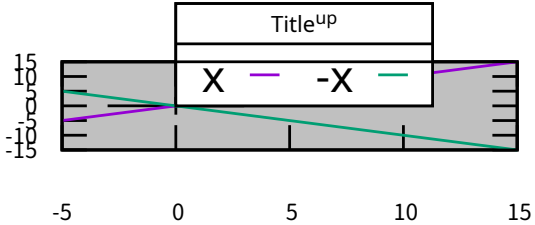
Key (<manual> horiz left top)



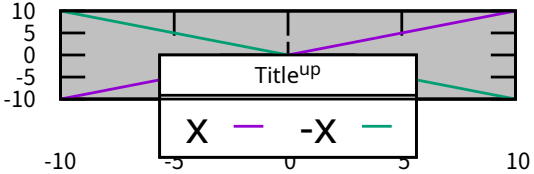
Key (<manual> horiz center left)



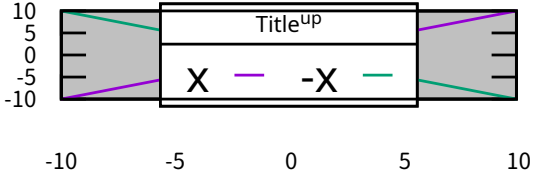
Key (<manual> horiz bot left)



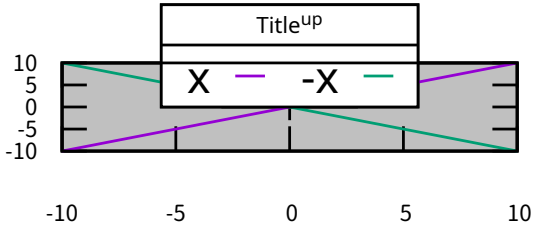
Key (<manual> horiz center top)



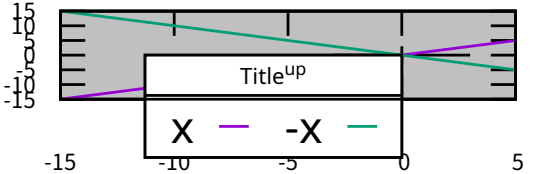
Key (<manual> horizontal center)



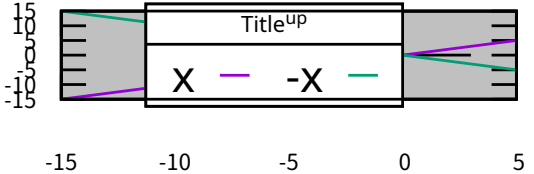
Key (<manual> horiz bot cente



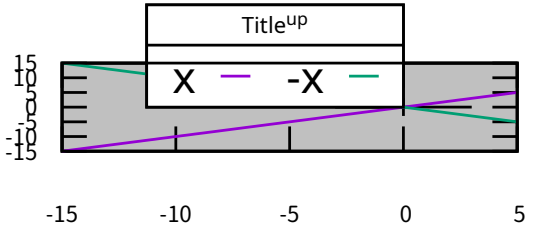
Key (<manual> horiz right top)



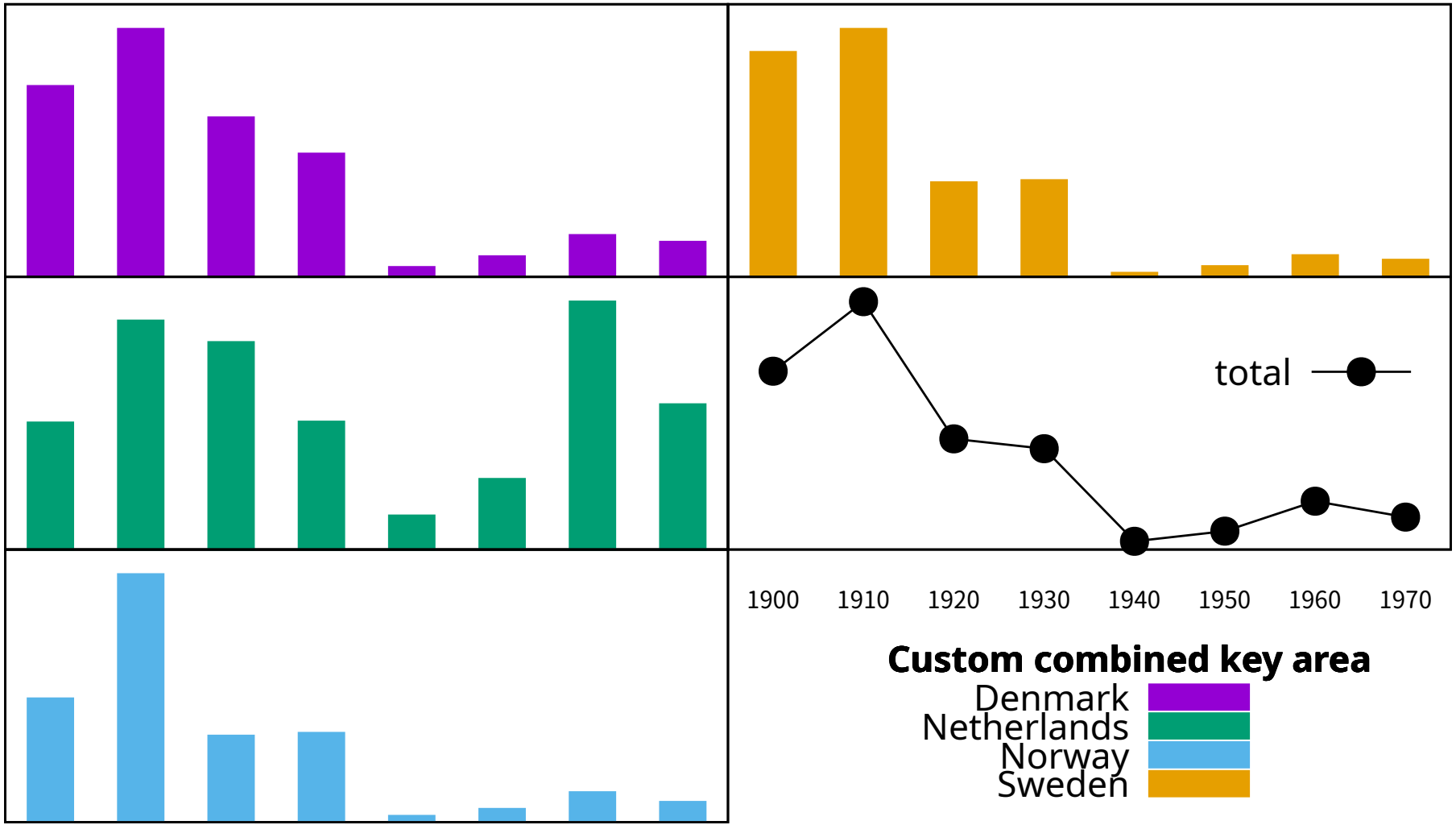
Key (<manual> horiz cent right)

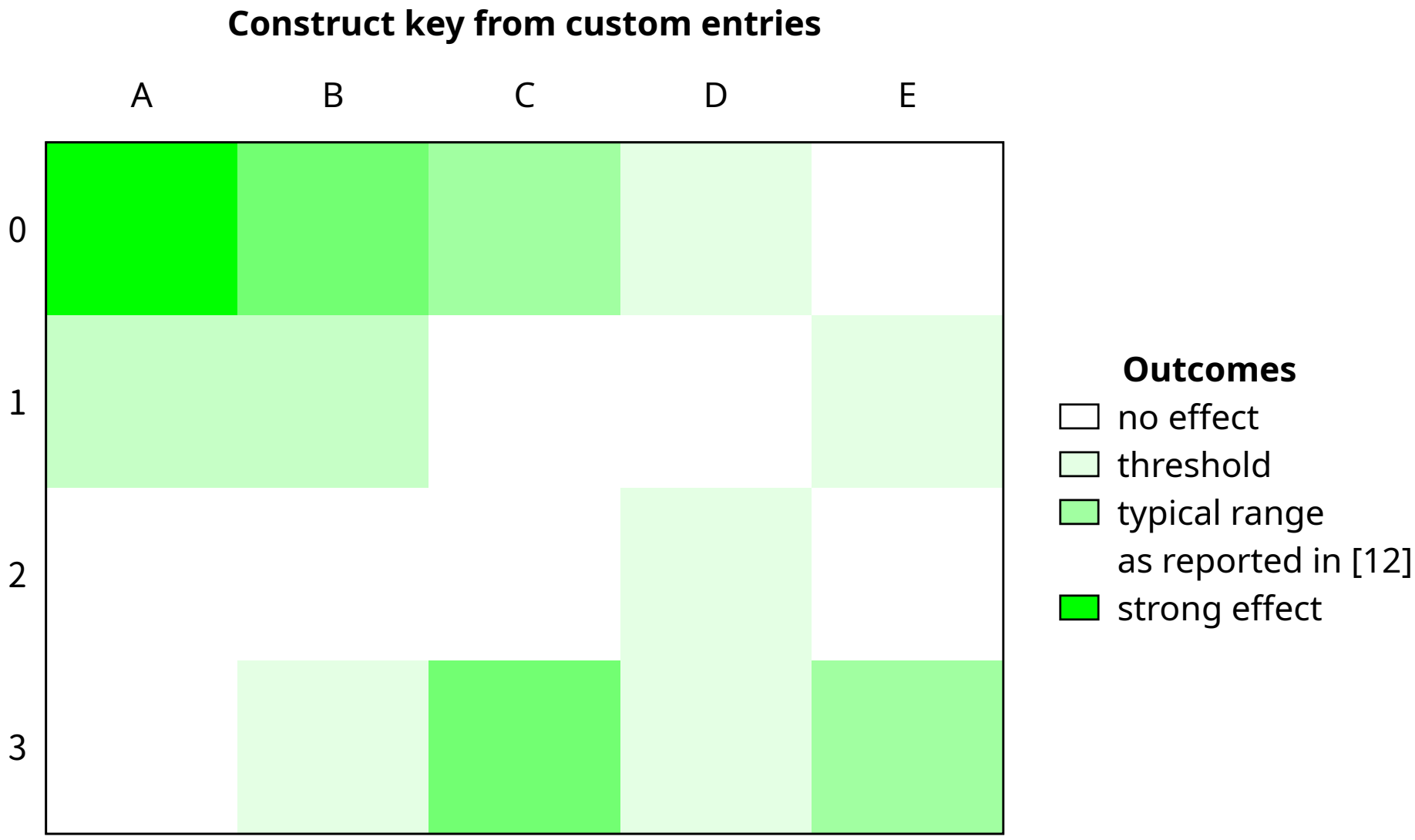


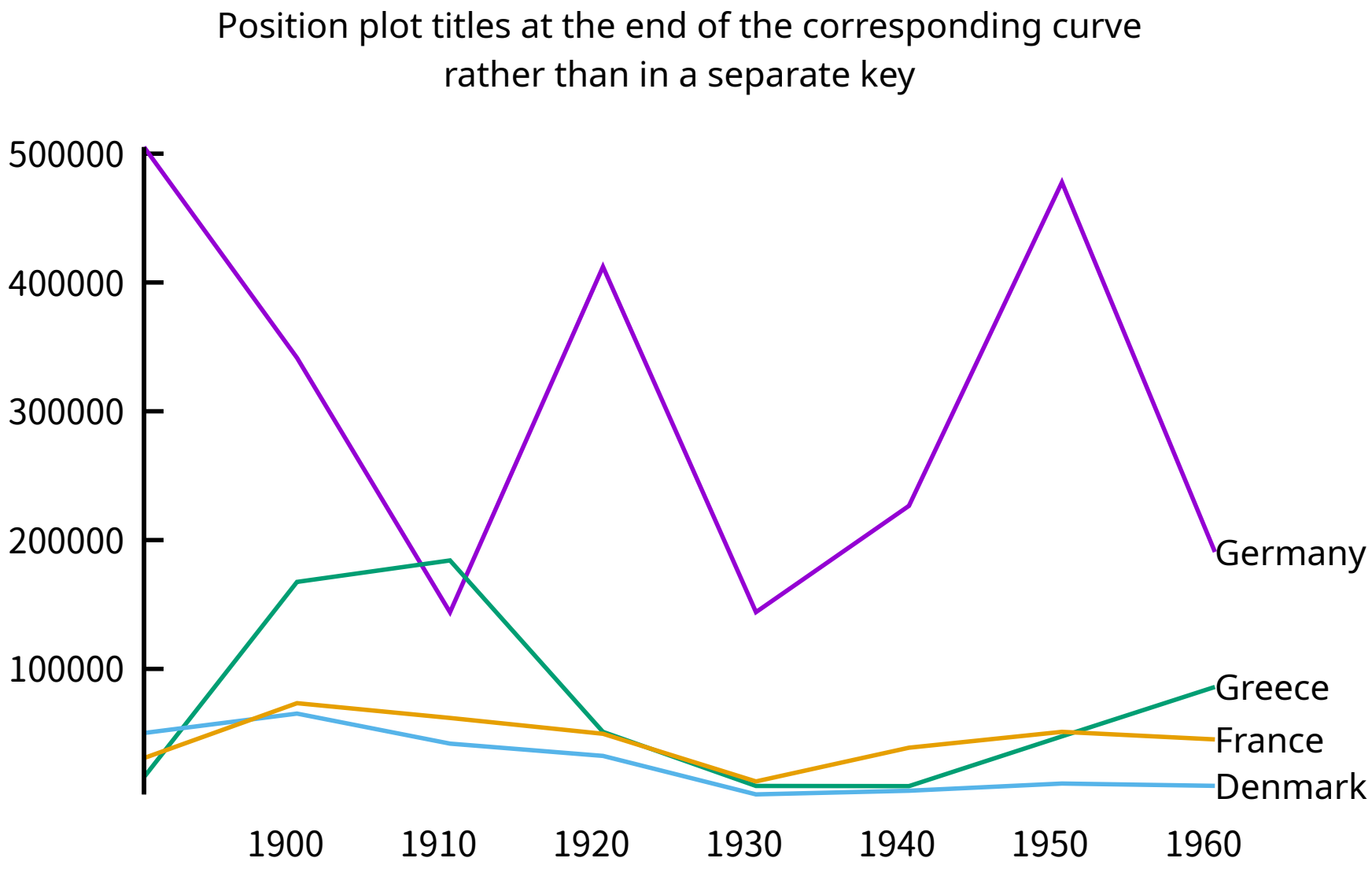
Key (<manual> horiz bot right

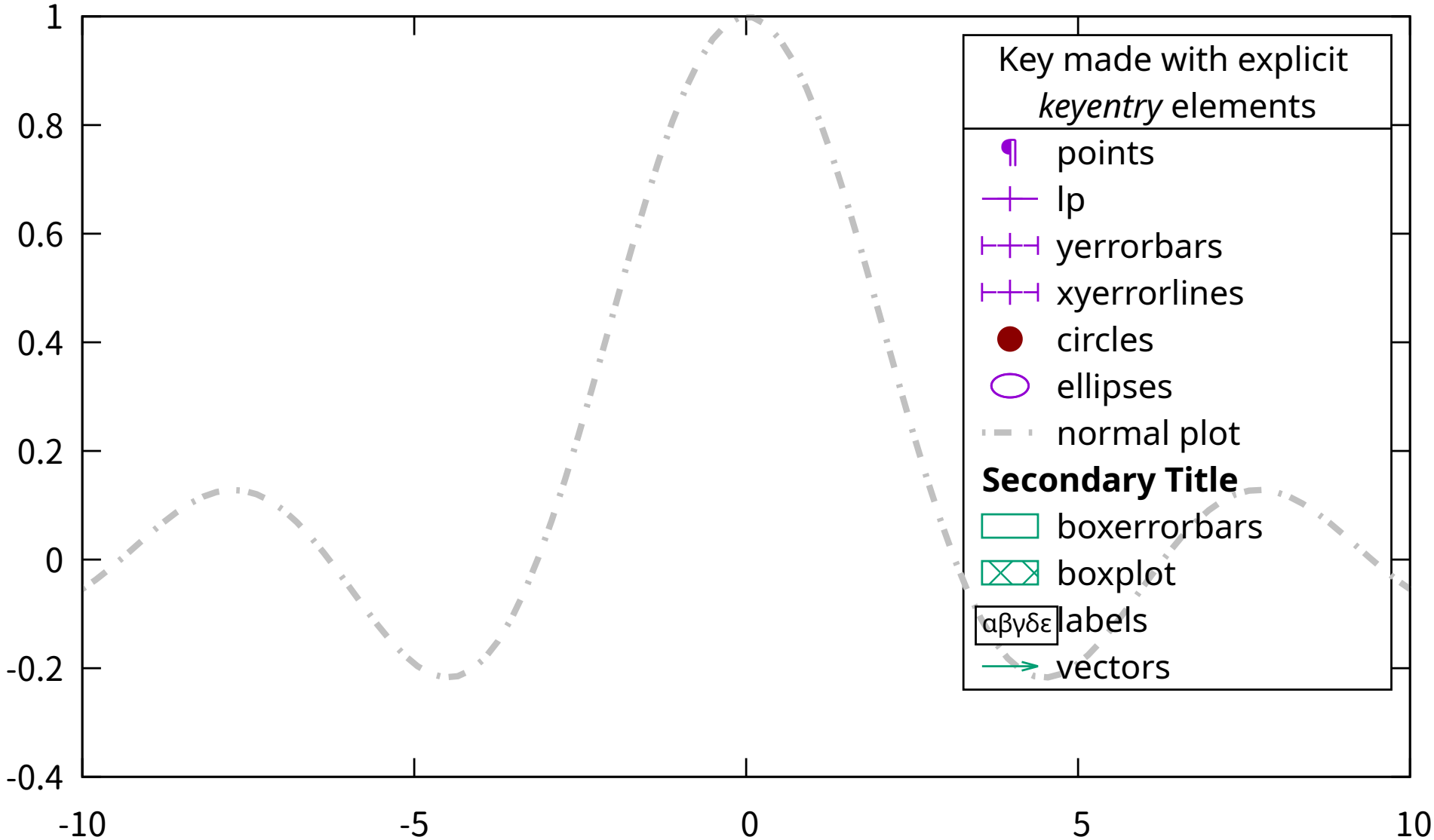


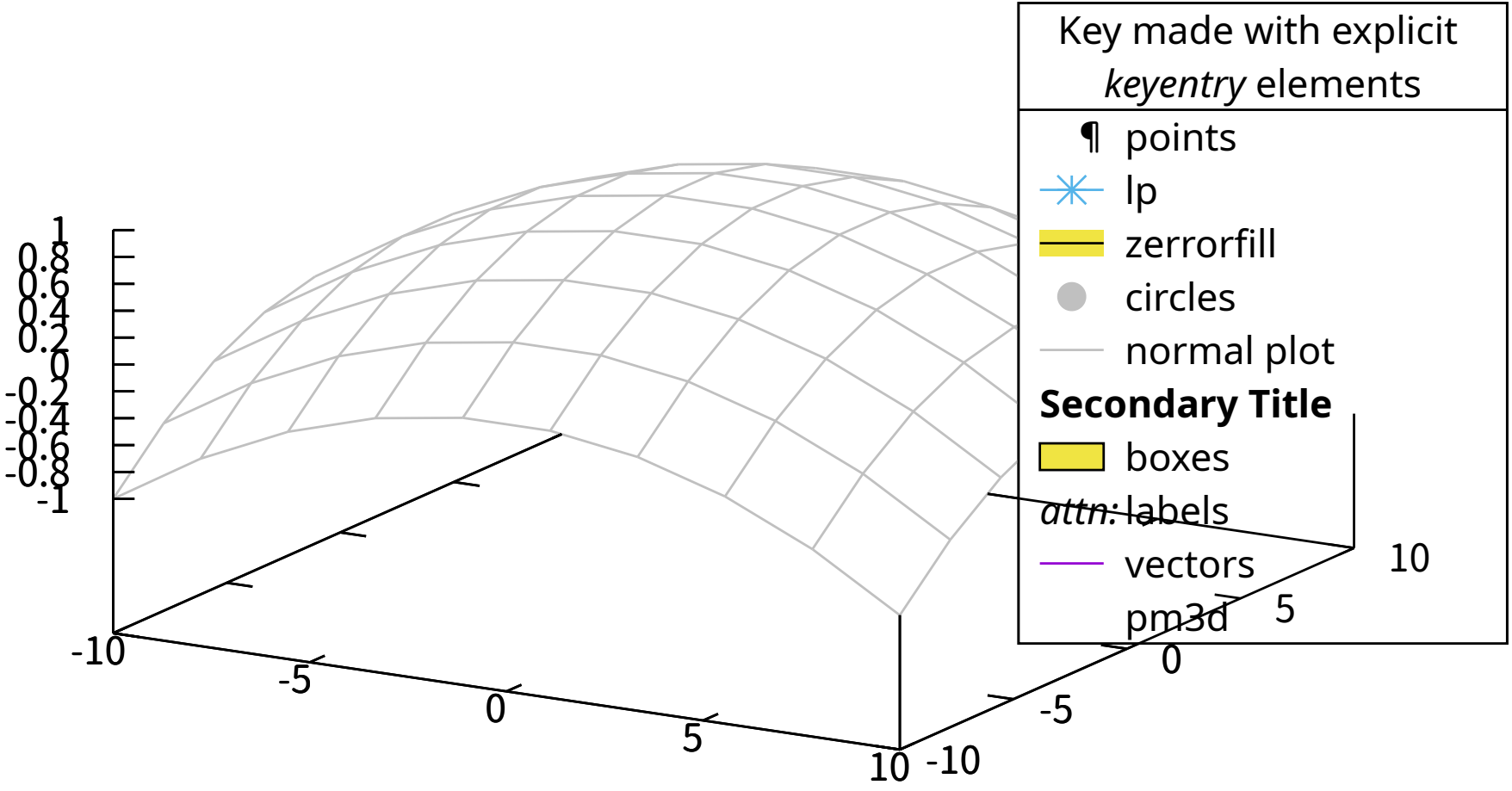
Illustrate use of a custom key area



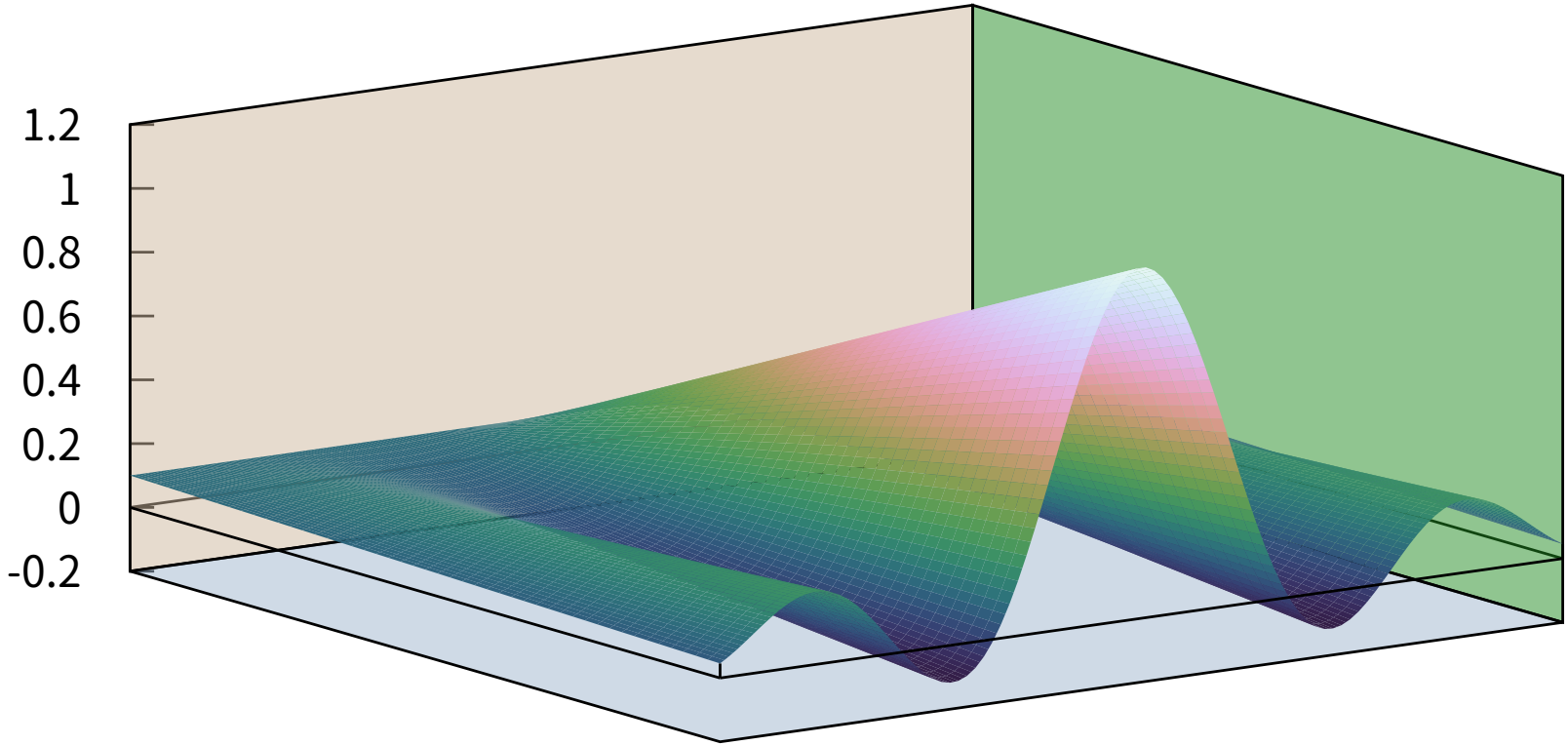

Construct key from custom entries
A
B
C
D
E
0
1
2
3
Outcomes
no effect
threshold
typical range
as reported in [12]
strong effect

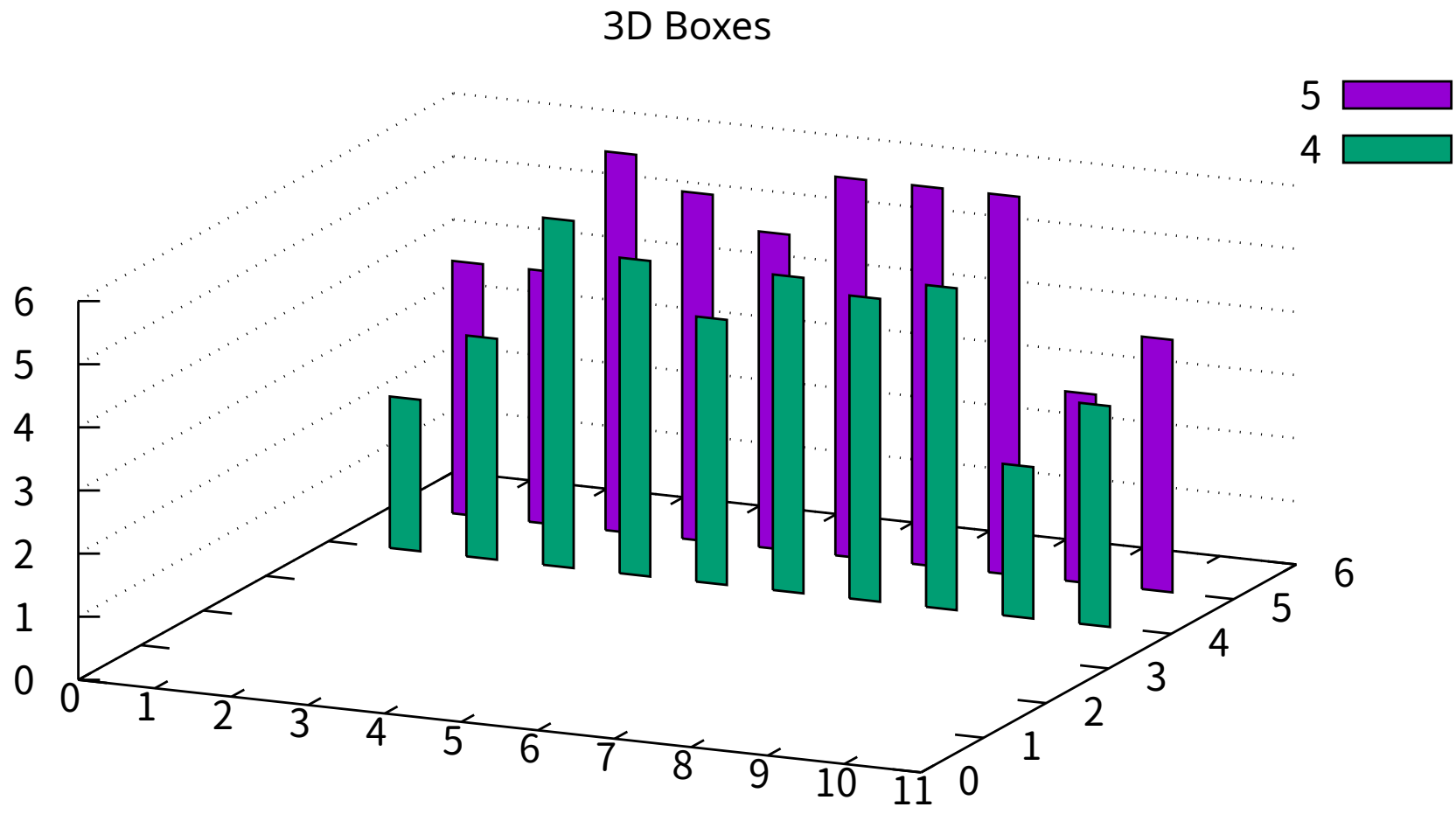

Position plot titles at the end of the corresponding curve
rather than in a separate key
500000
400000
300000
200000
100000
1900
1910
1920
1930
1940
1950
1960
Germany
Greece
France
Denmark

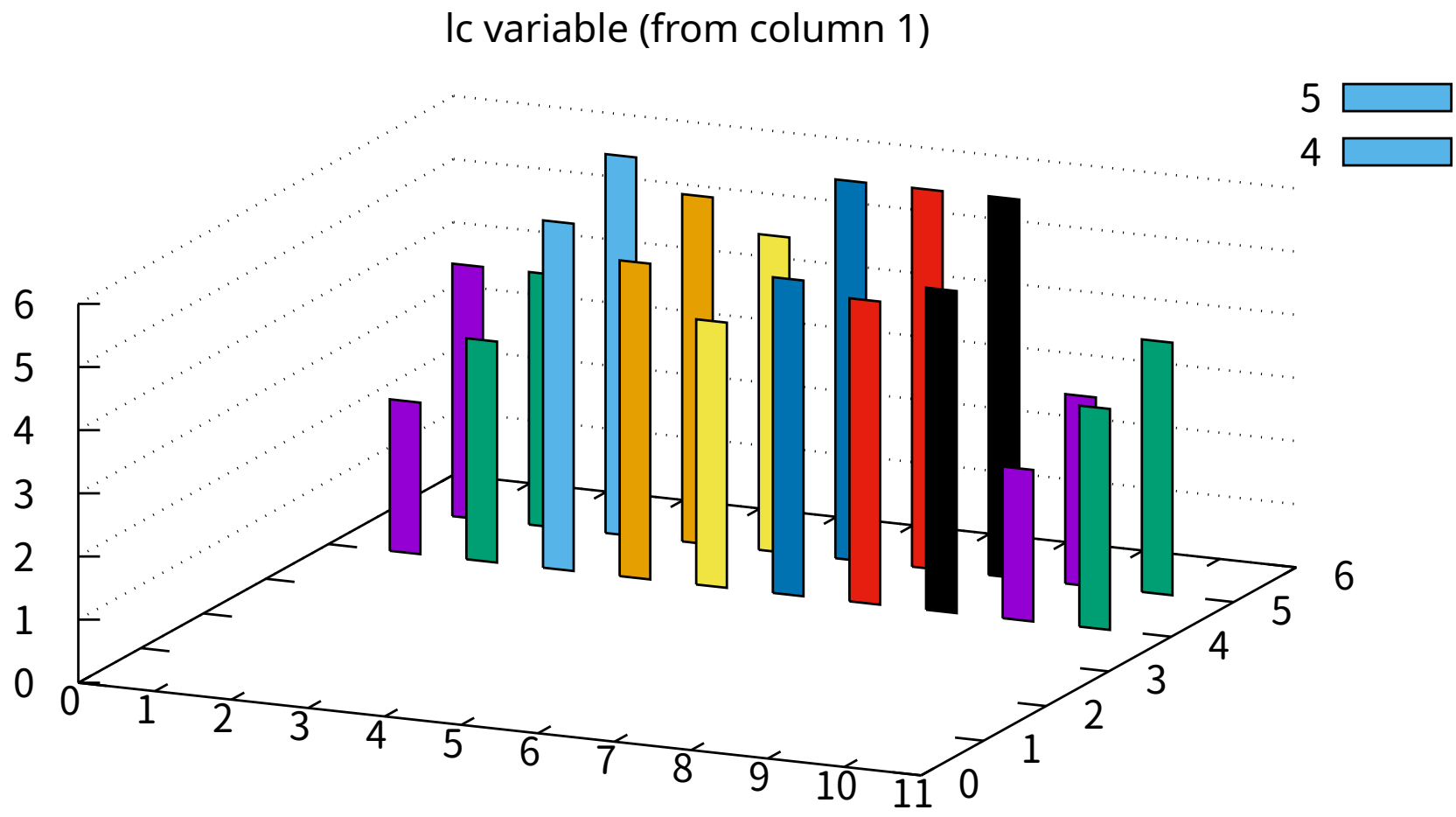

Key made with explicit
keyentry elements
points
lp
yerrorbars
xyerrorlines
circles
ellipses
normal plot
Secondary Title
boxerrorbars
boxplot
αβγδε
labels
vectors
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10

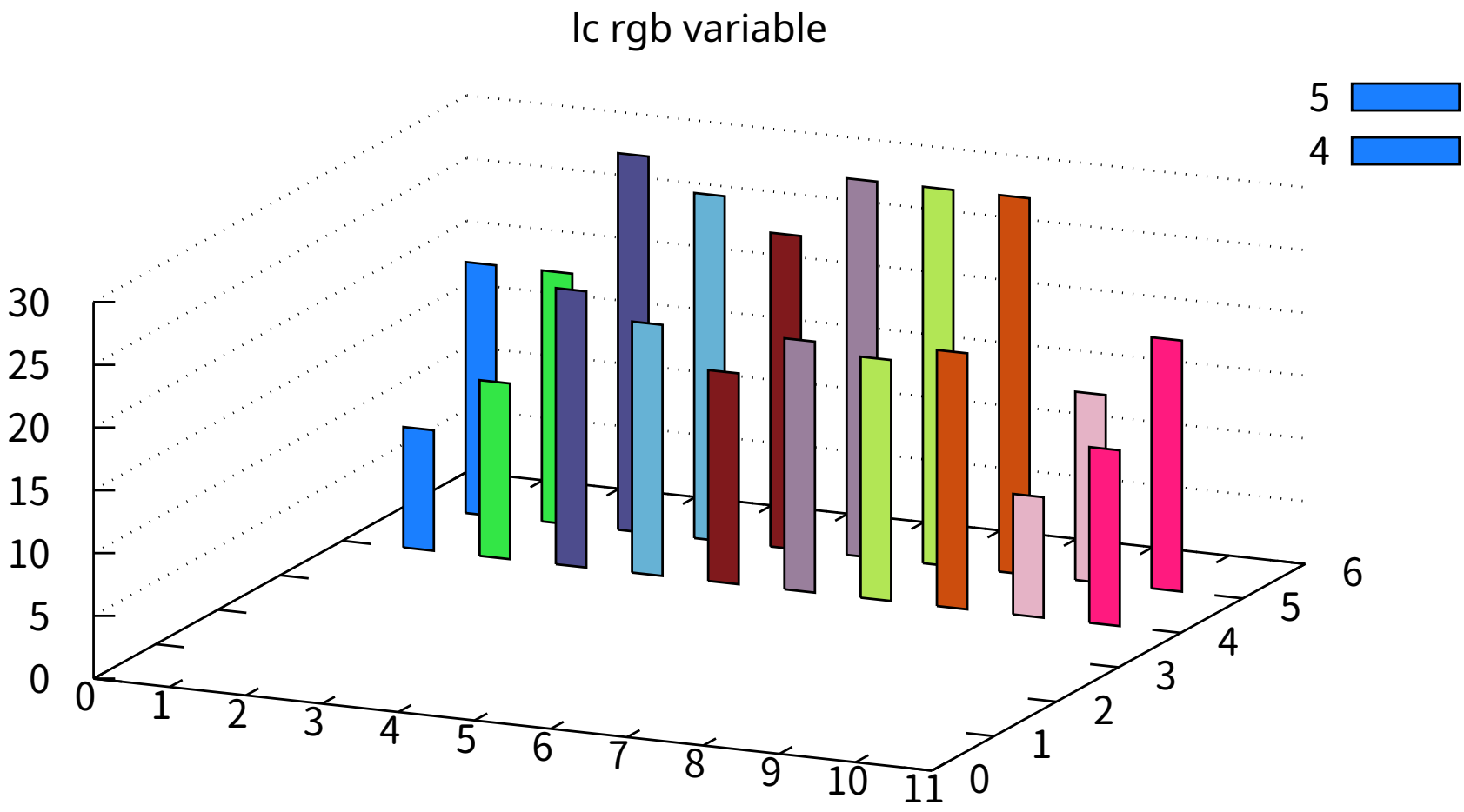

Key made with explicit
keyentry elements
points
lp
zerrorfill
circles
normal plot
Secondary Title
boxes
attn: labels
vectors
pm3d
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-10
-5
0
5
10
-10
-5
0
5
10

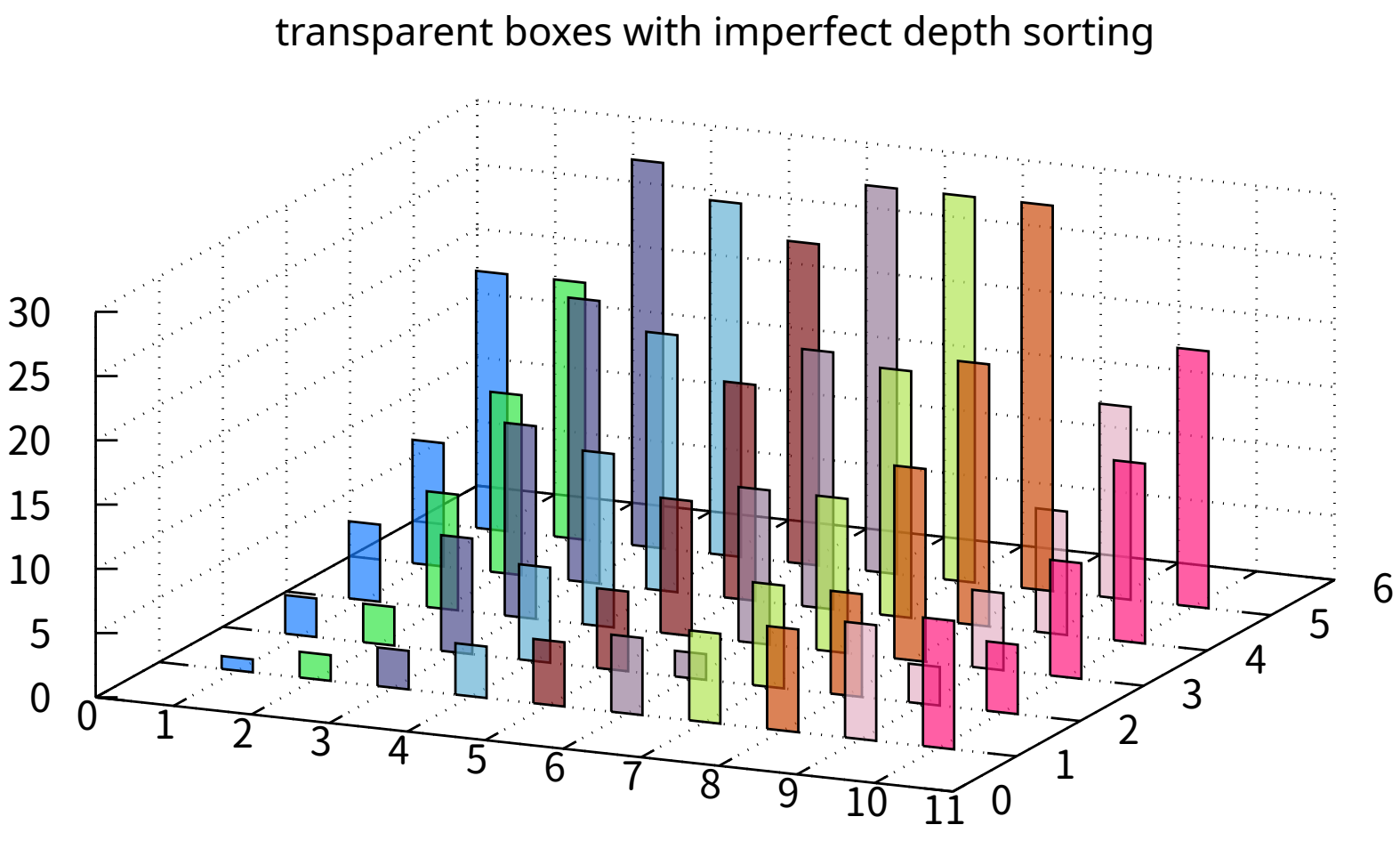
Test/demo of new feature 'grid walls'



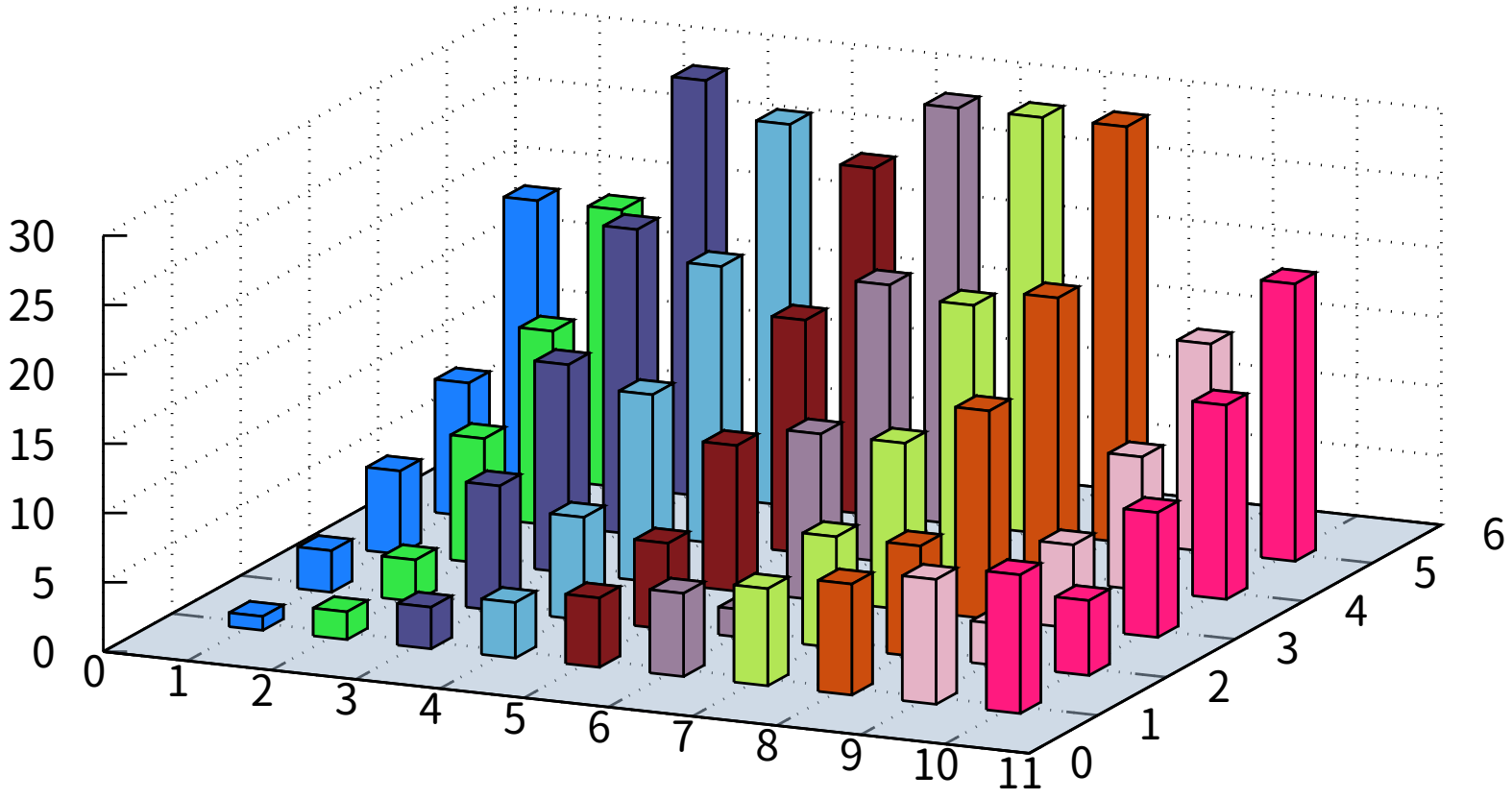

3D Boxes
5
4
0
1
2
3
4
5
6
0
1
2
3
4
5
6
7
8
9
10
11
0
1
2
3
4
5
6


lc variable (from column 1)
5
4
0
1
2
3
4
5
6
0
1
2
3
4
5
6
7
8
9
10
11
0
1
2
3
4
5
6

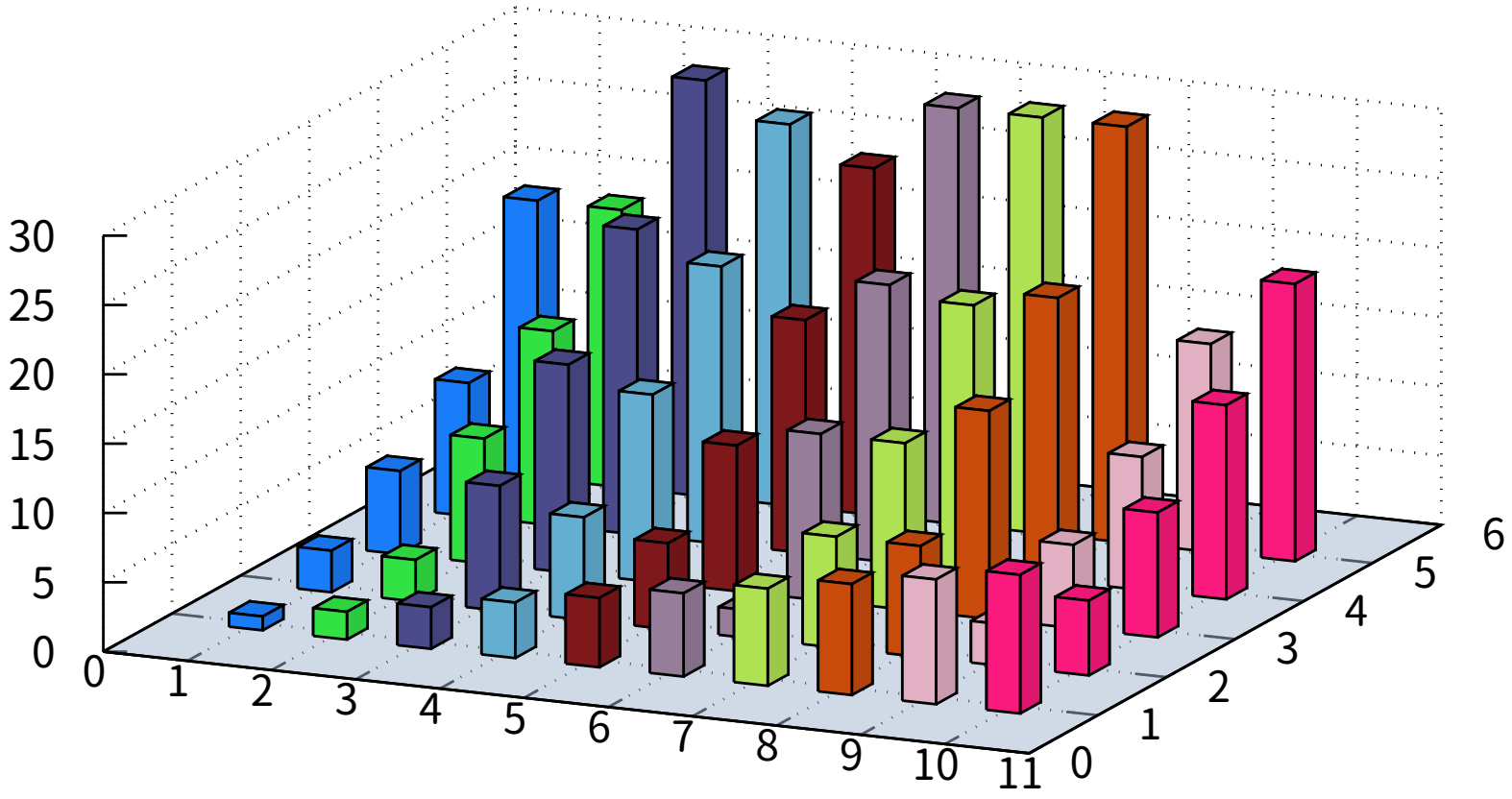

lc rgb variable
5
4
30
25
20
15
10
5
0
0
1
2
3
4
5
6
7
8
9
10
11
0
1
2
3
4
5
6


transparent boxes with imperfect depth sorting
30
25
20
15
10
5
0
0
1
2
3
4
5
6
7
8
9
10
11
0
1
2
3
4
5
6

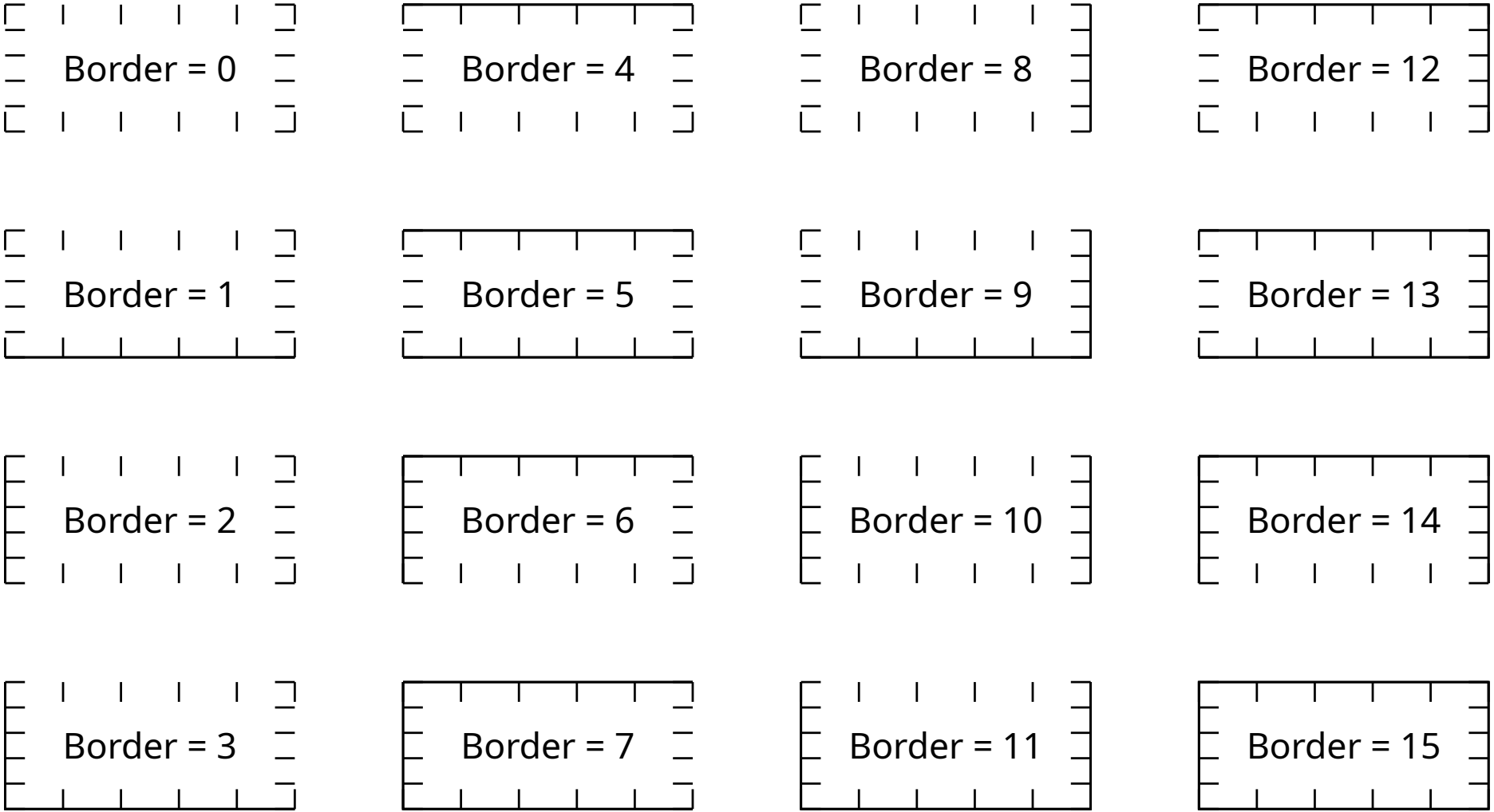
give the boxes a 3D depth and correct depth sorting

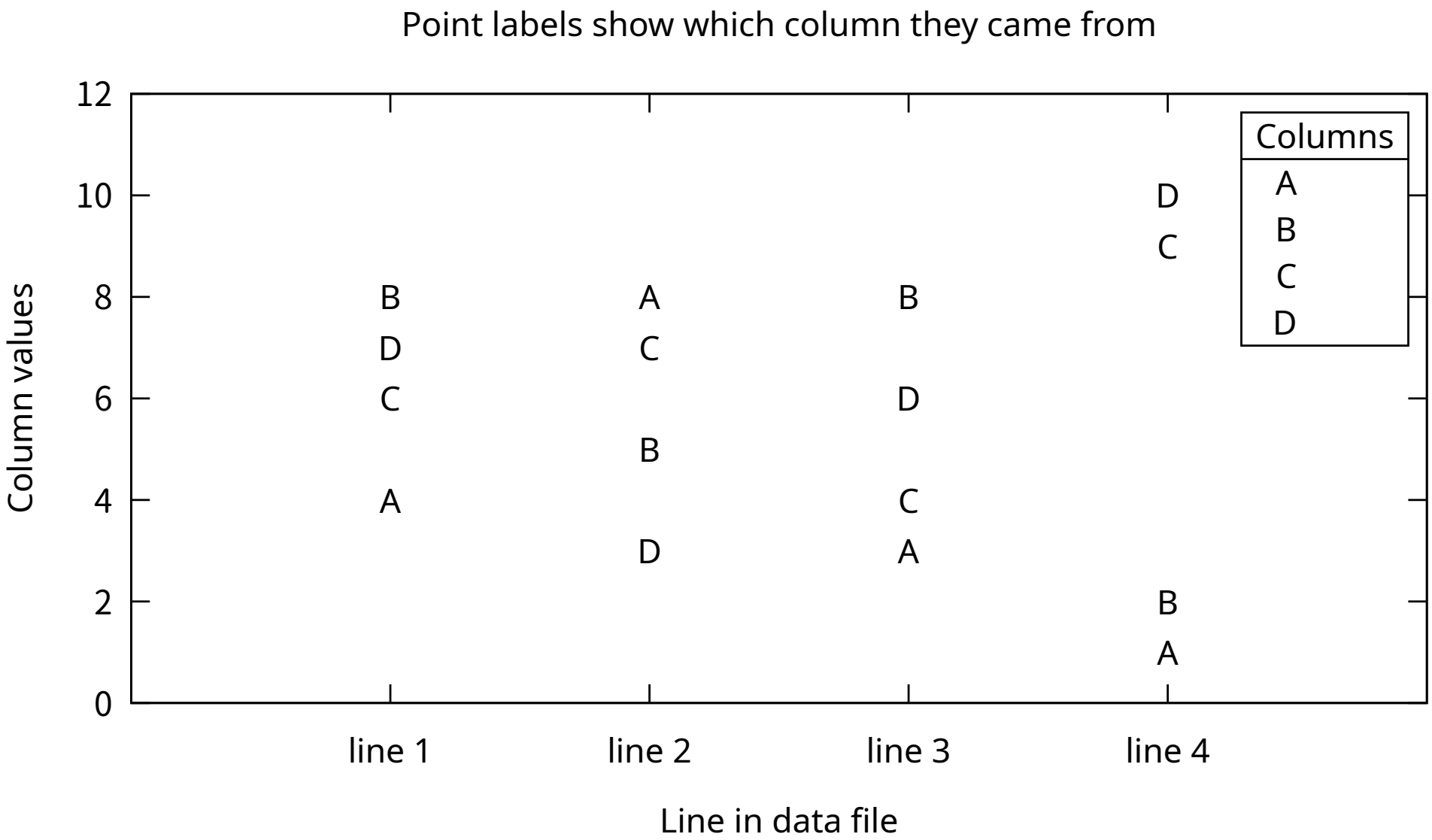


Full treatment: 3D boxes with pm3d depth sorting and lighting

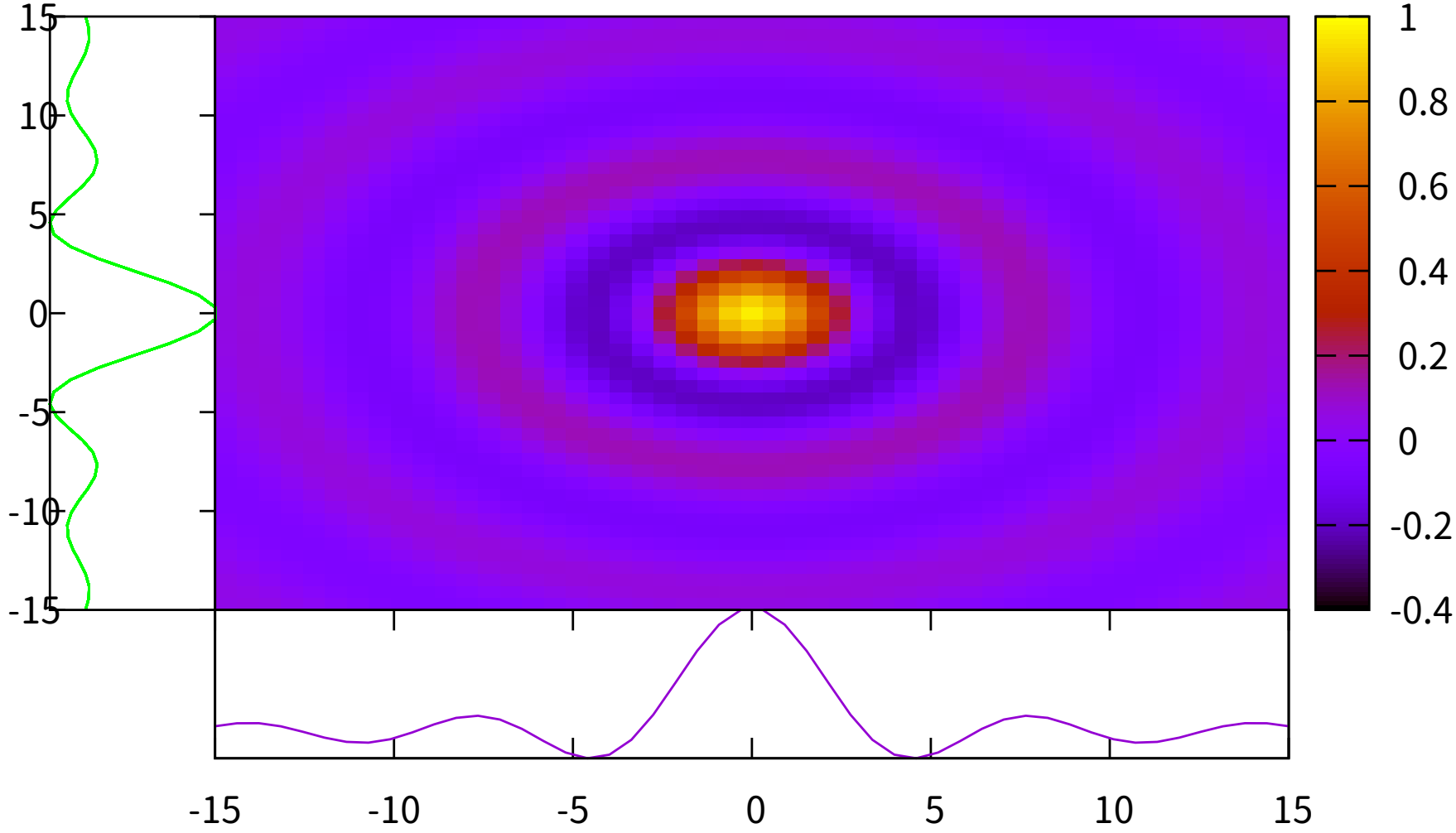


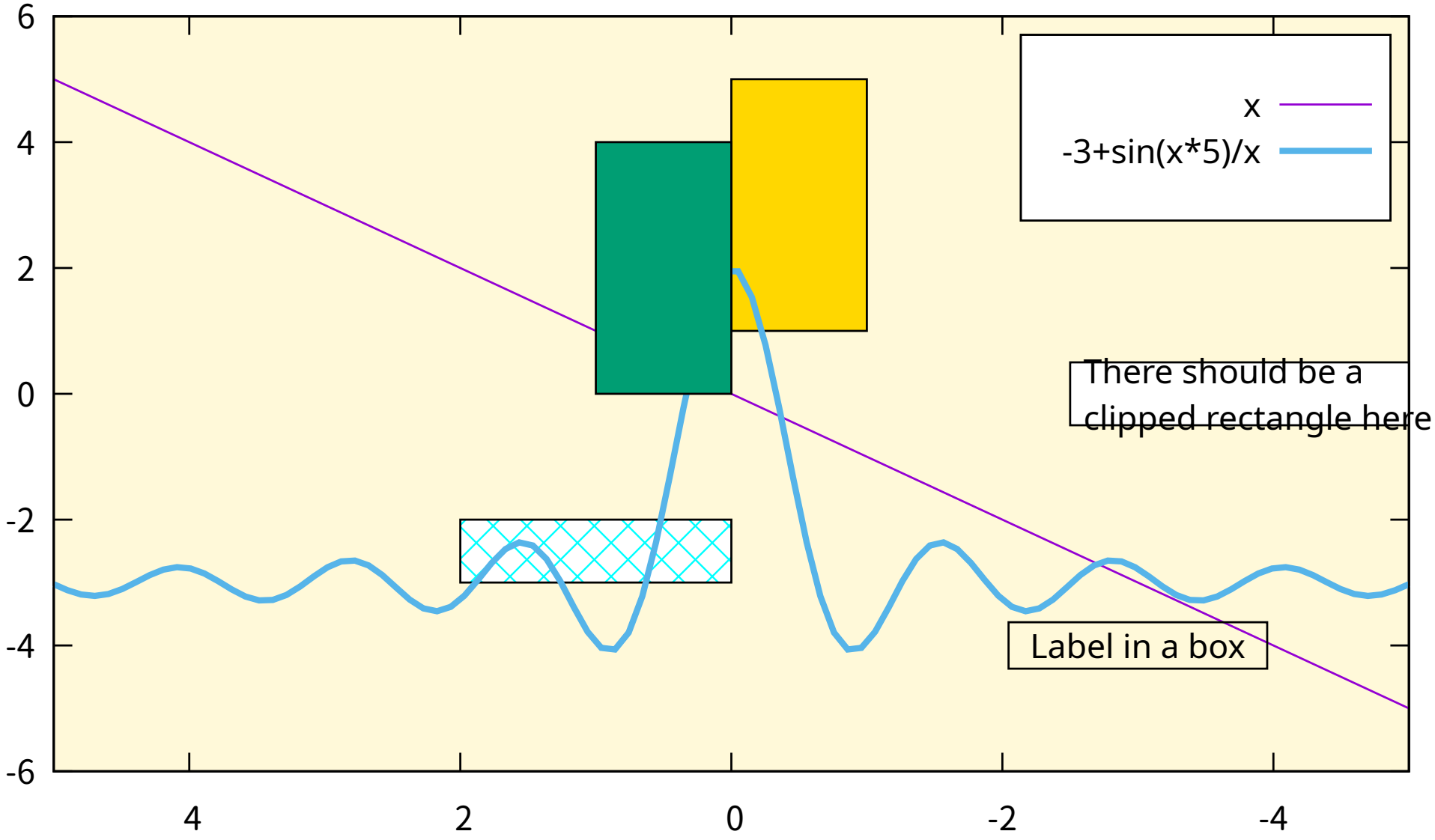
Demonstration of different border settings

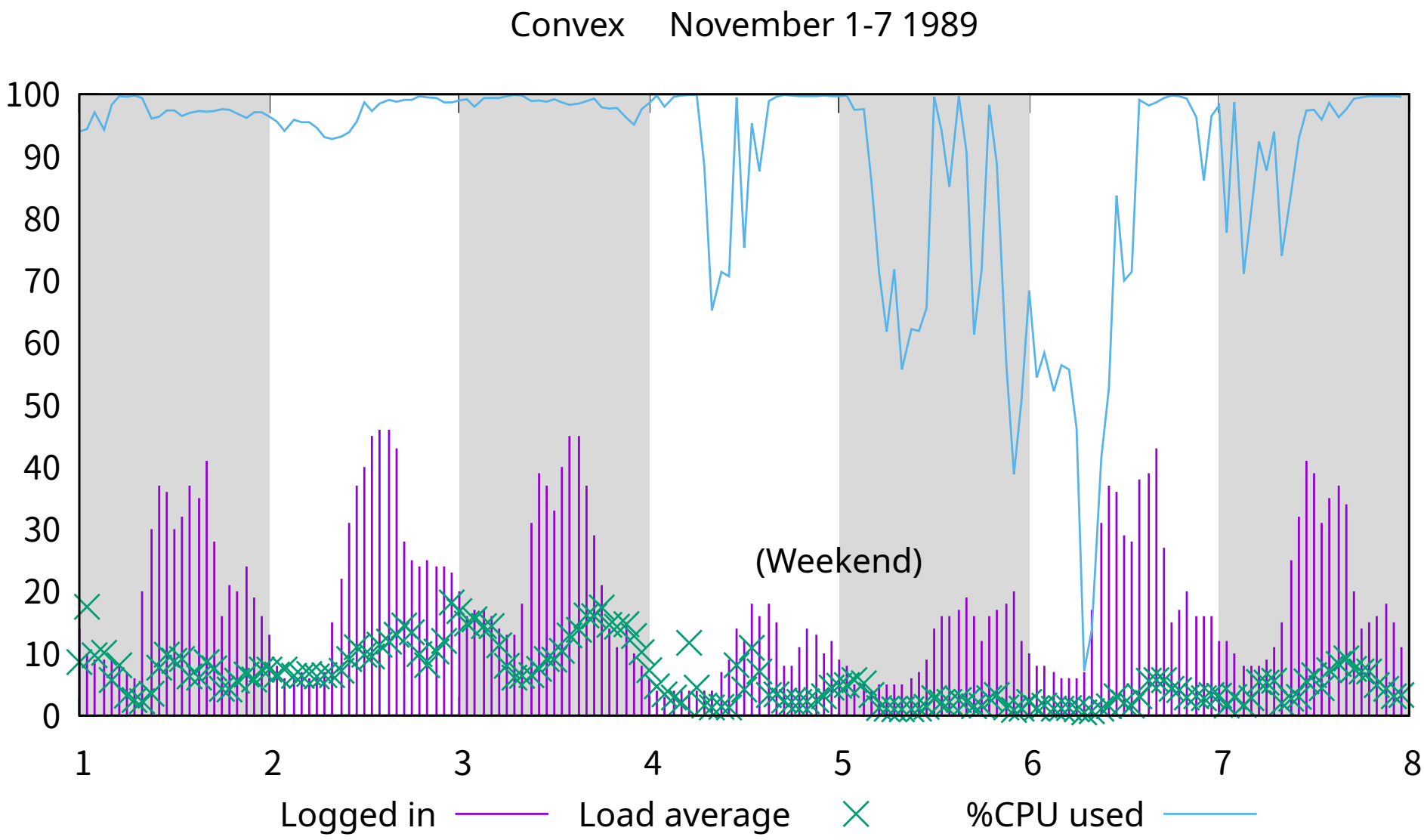


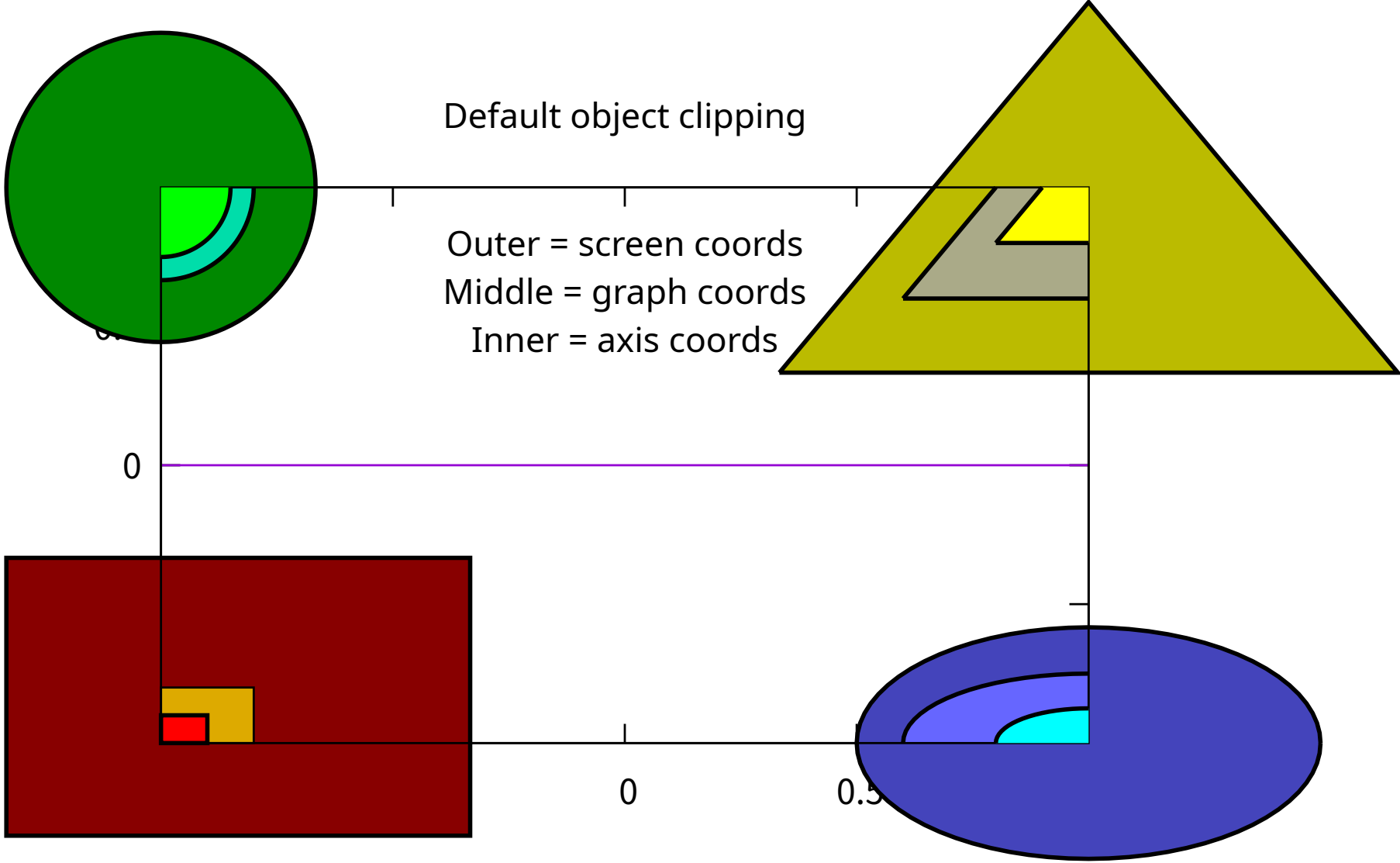

Point labels show which column they came from
Column values
Line in data file
line 1
line 2
line 3
line 4
0
2
4
6
8
10
12
Columns
A
B
C
D

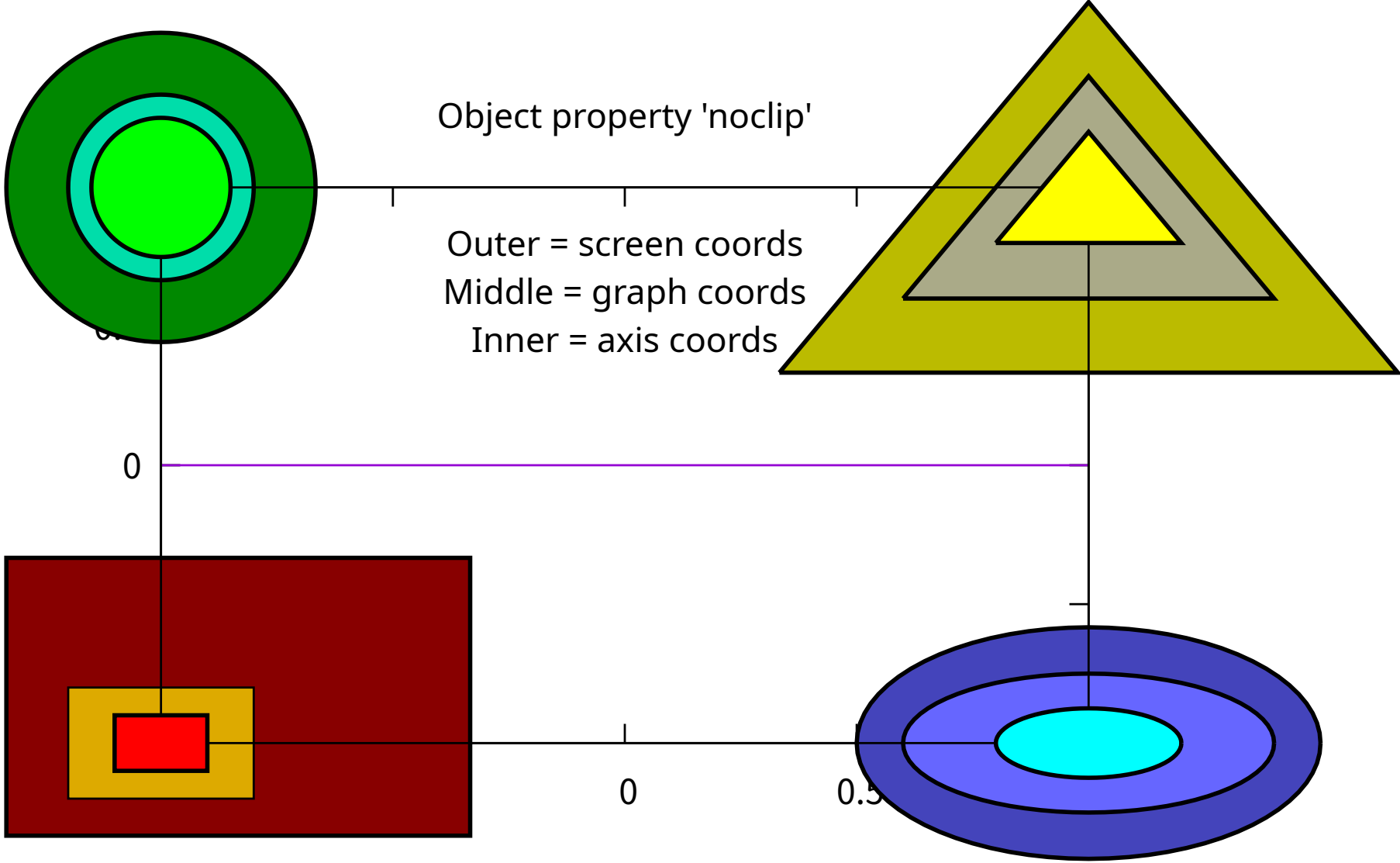
Demo of placing multiple plots (2D and 3D)
with explicit alignment of plot borders

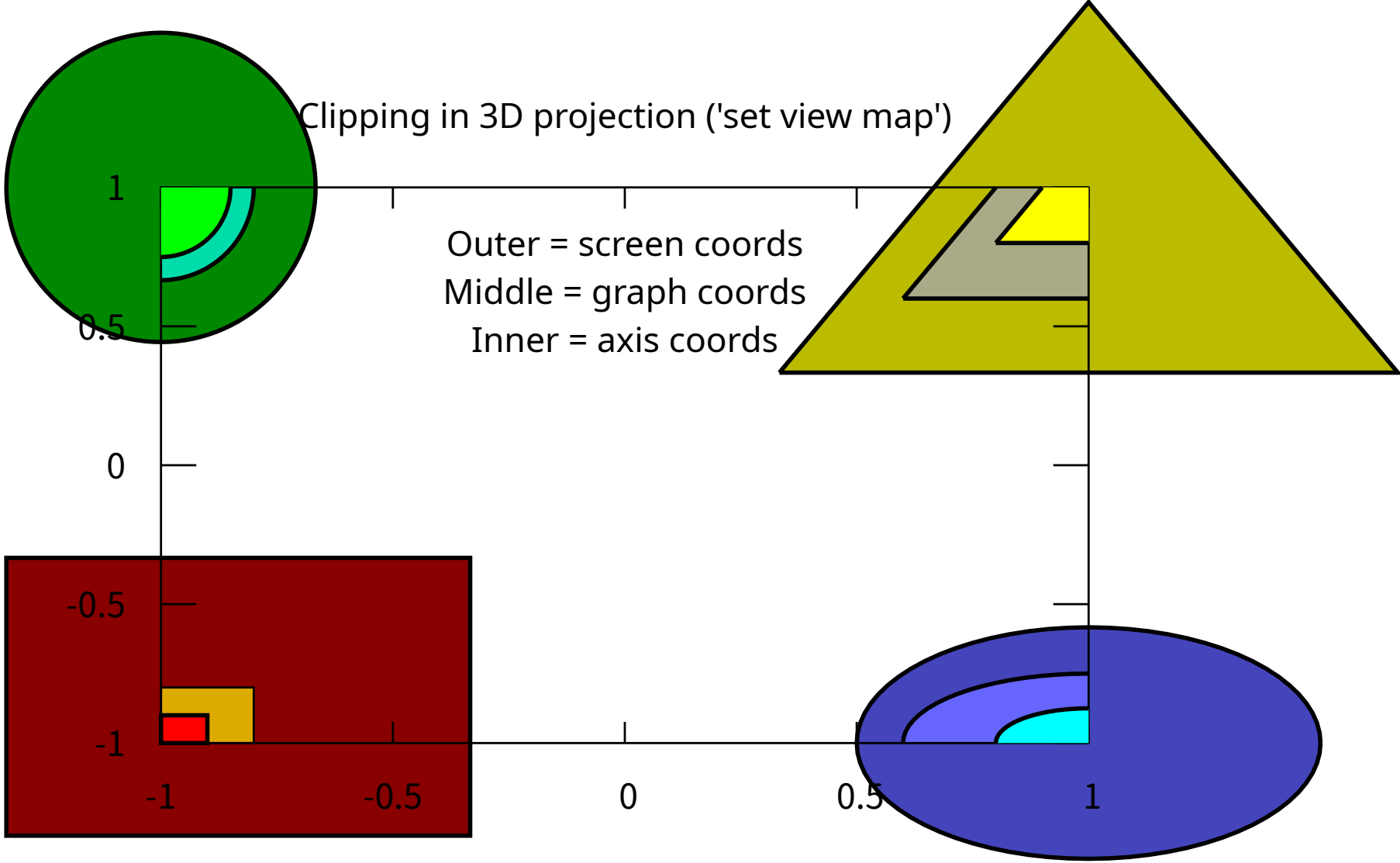


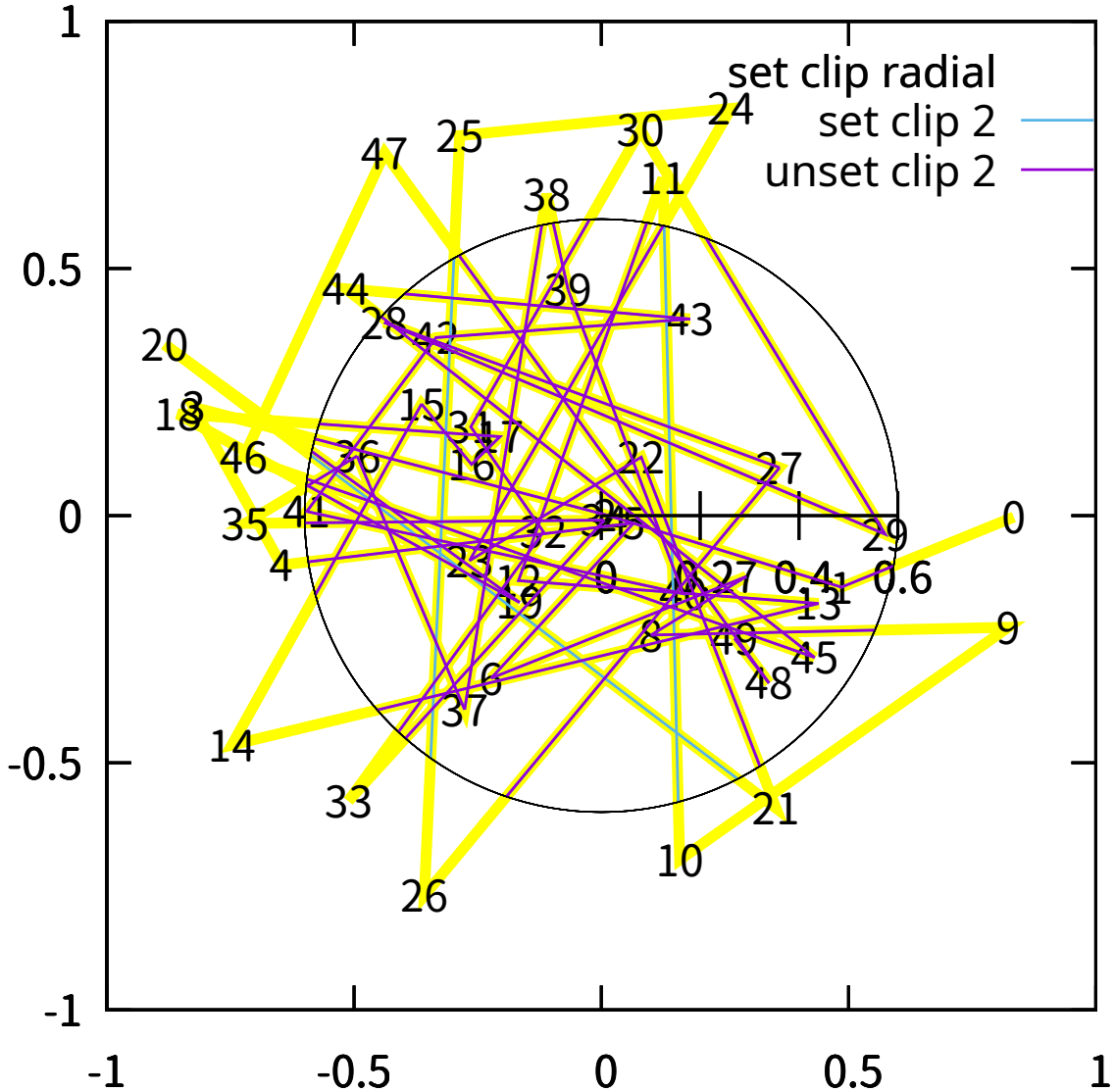

6
4
2
0
-2
-4
-6
4
2
0
-2
-4
x
-3+sin(x*5)/x
There should be a
clipped rectangle here
Label in a box

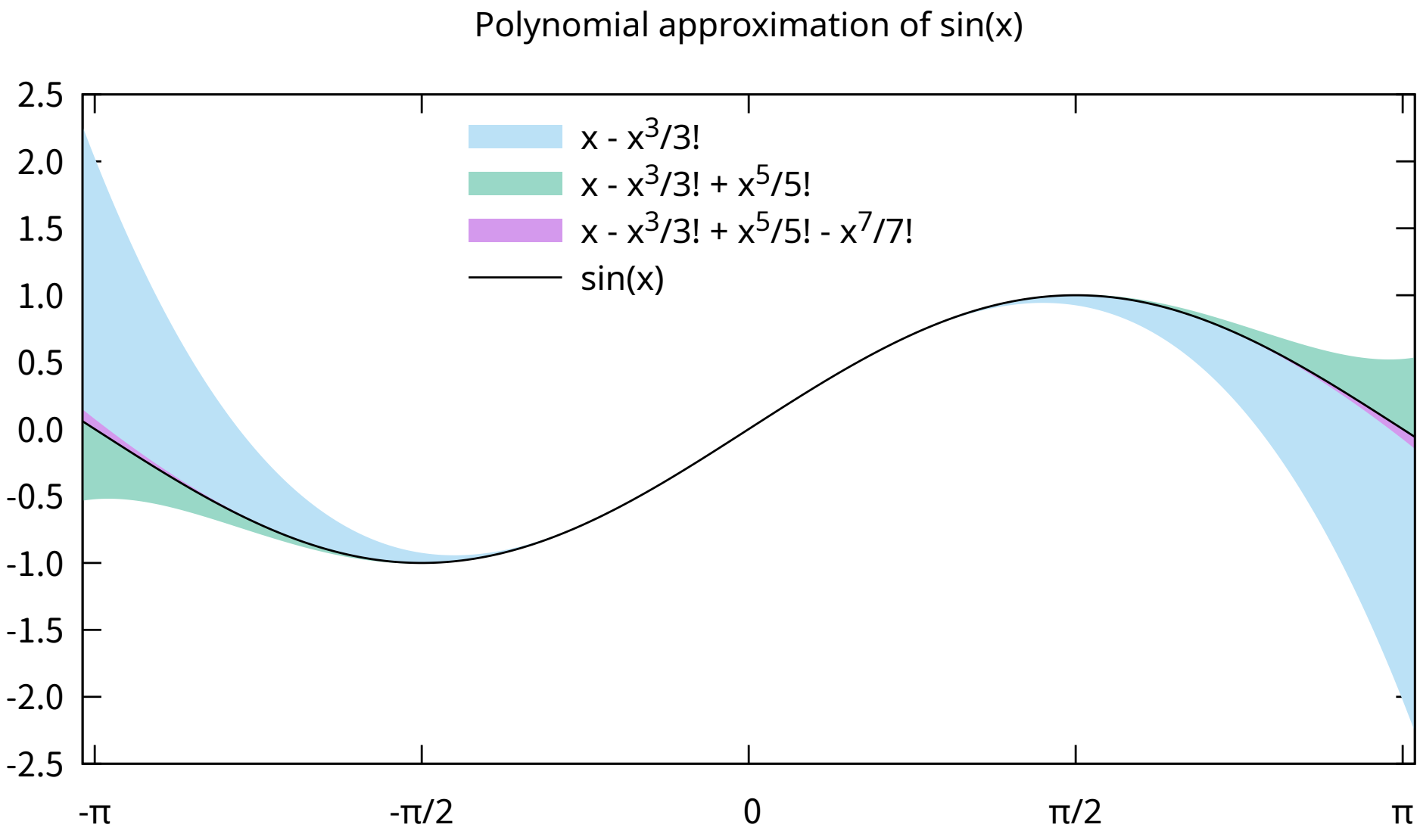

Convex November 1-7 1989
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
(Weekend)
Logged in
Load average
%CPU used


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

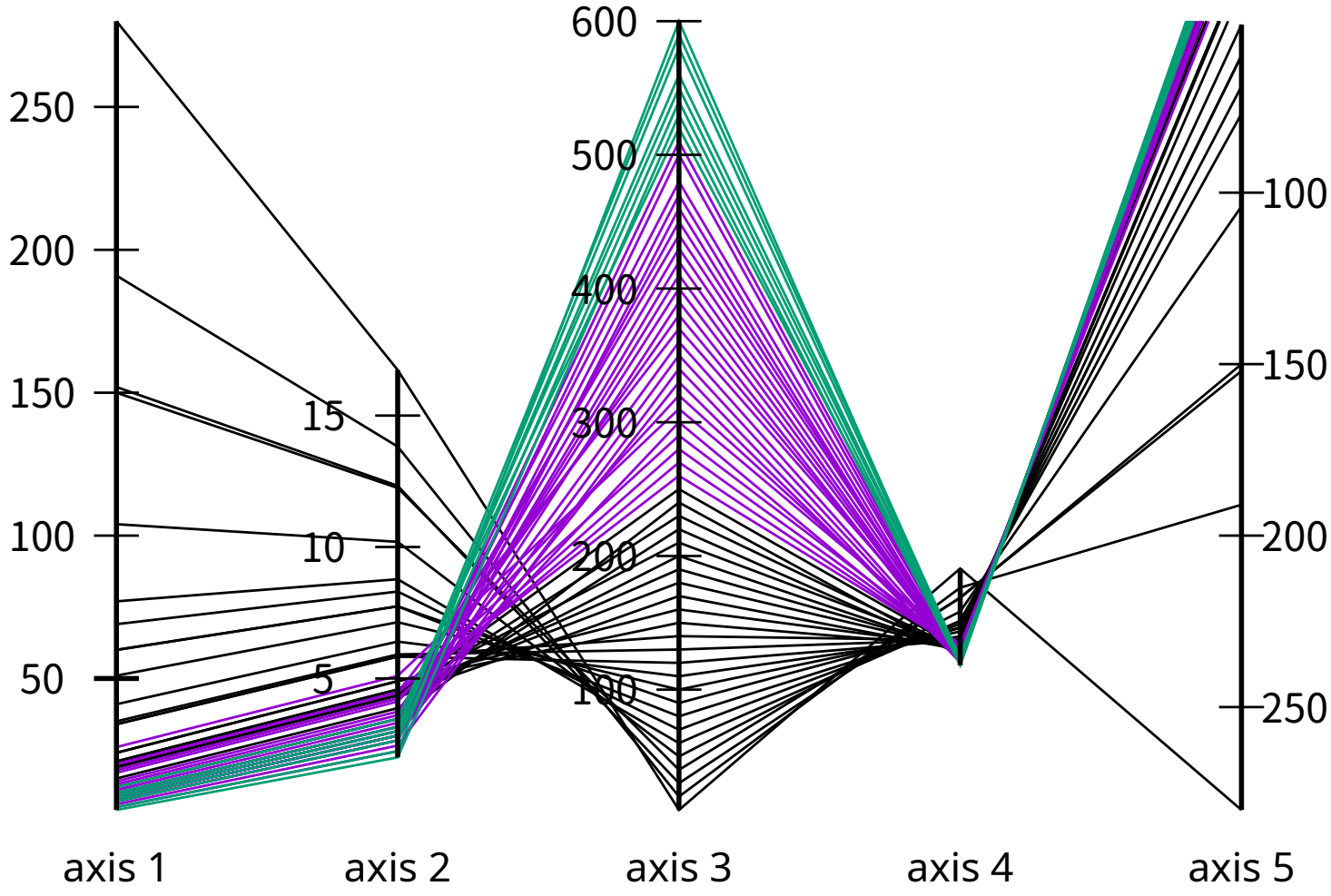

Object property 'noclip'
Outer = screen coords
Middle = graph coords
Inner = axis coords
0
0
0.5


Clipping in 3D projection ('set view map')
Outer = screen coords
Middle = graph coords
Inner = axis coords
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

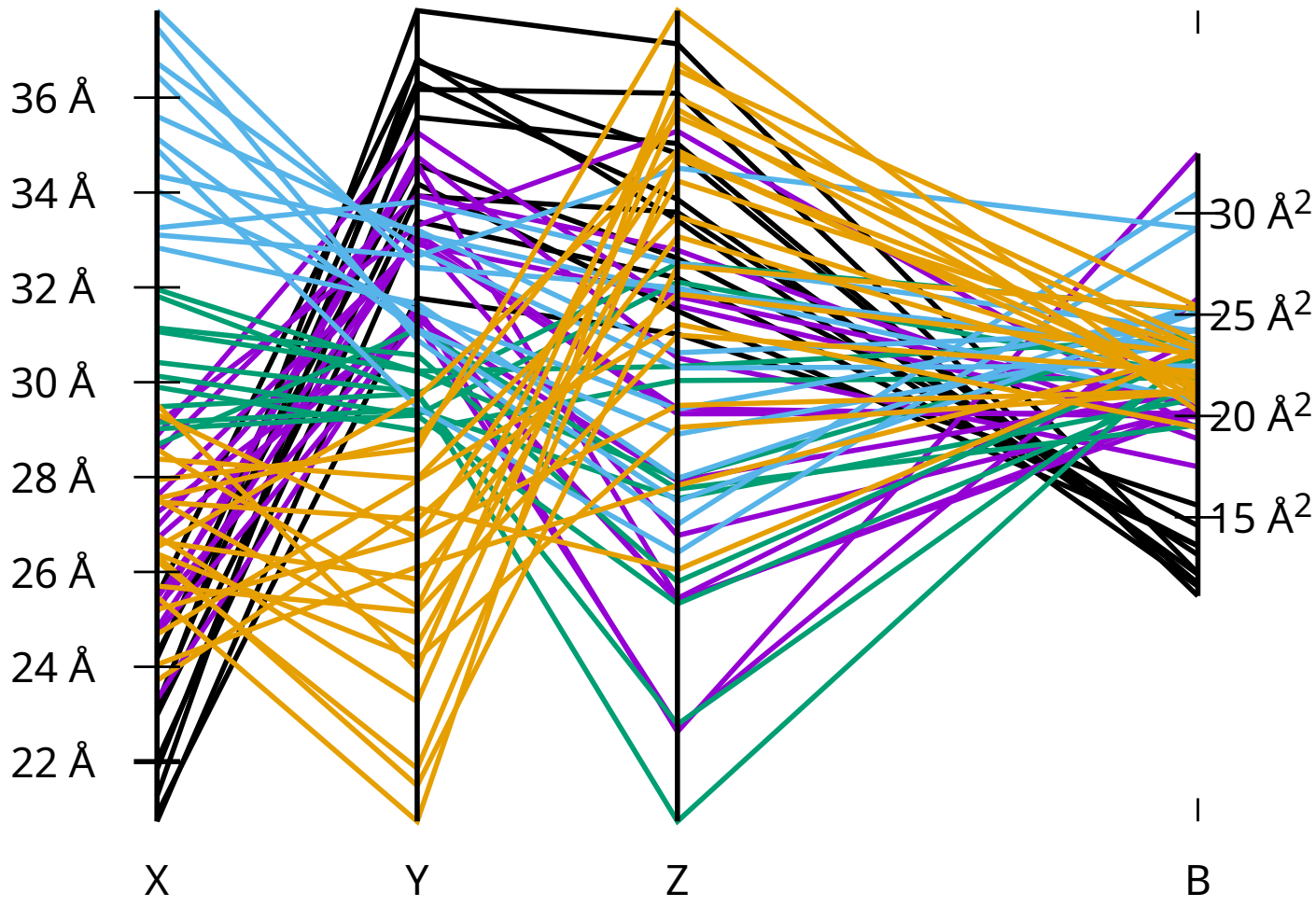

set clip radial
set clip 2
unset clip 2
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1


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
-π
-π/2
0
π/2
π

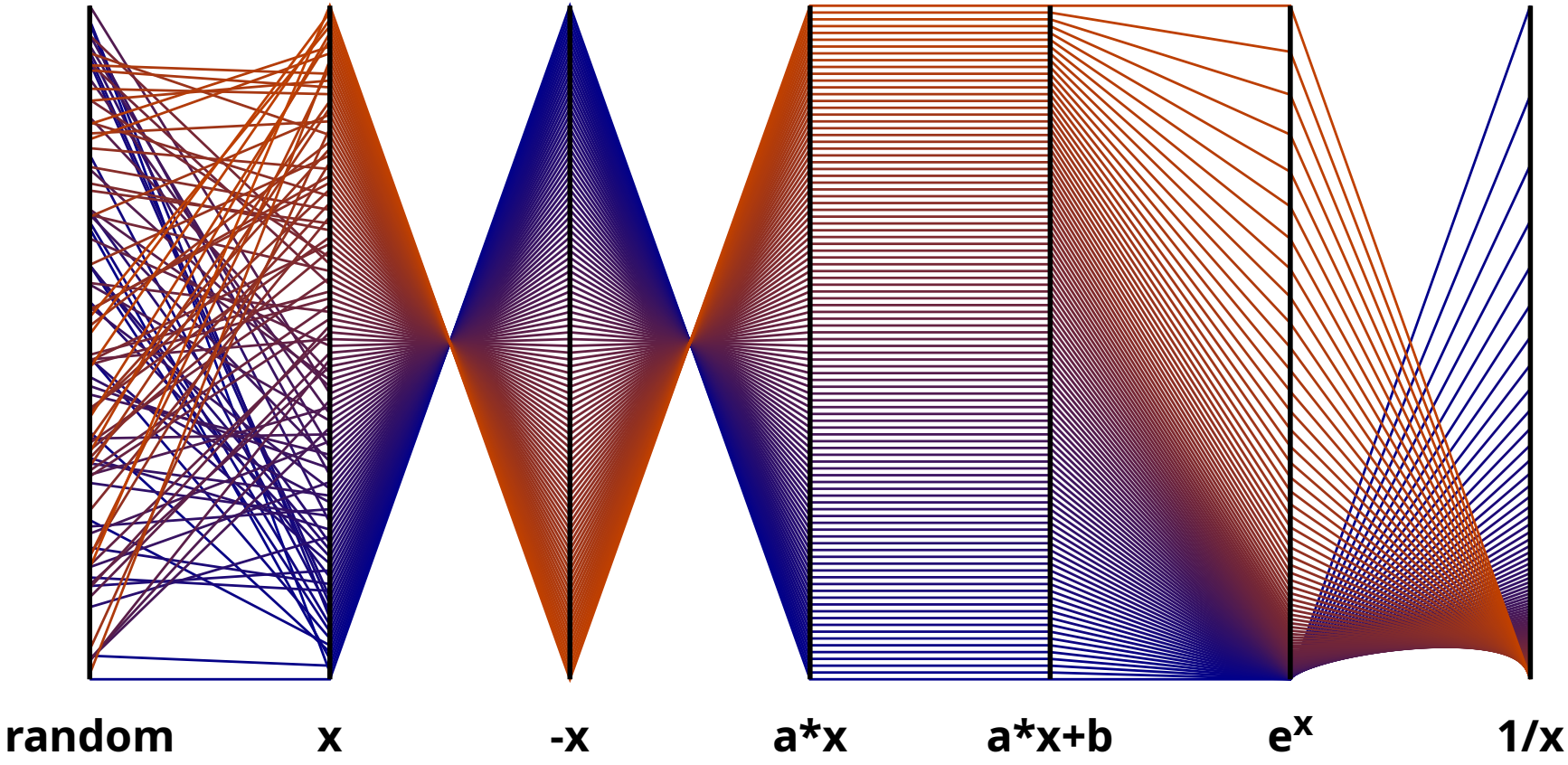
Parallel Axis Plot

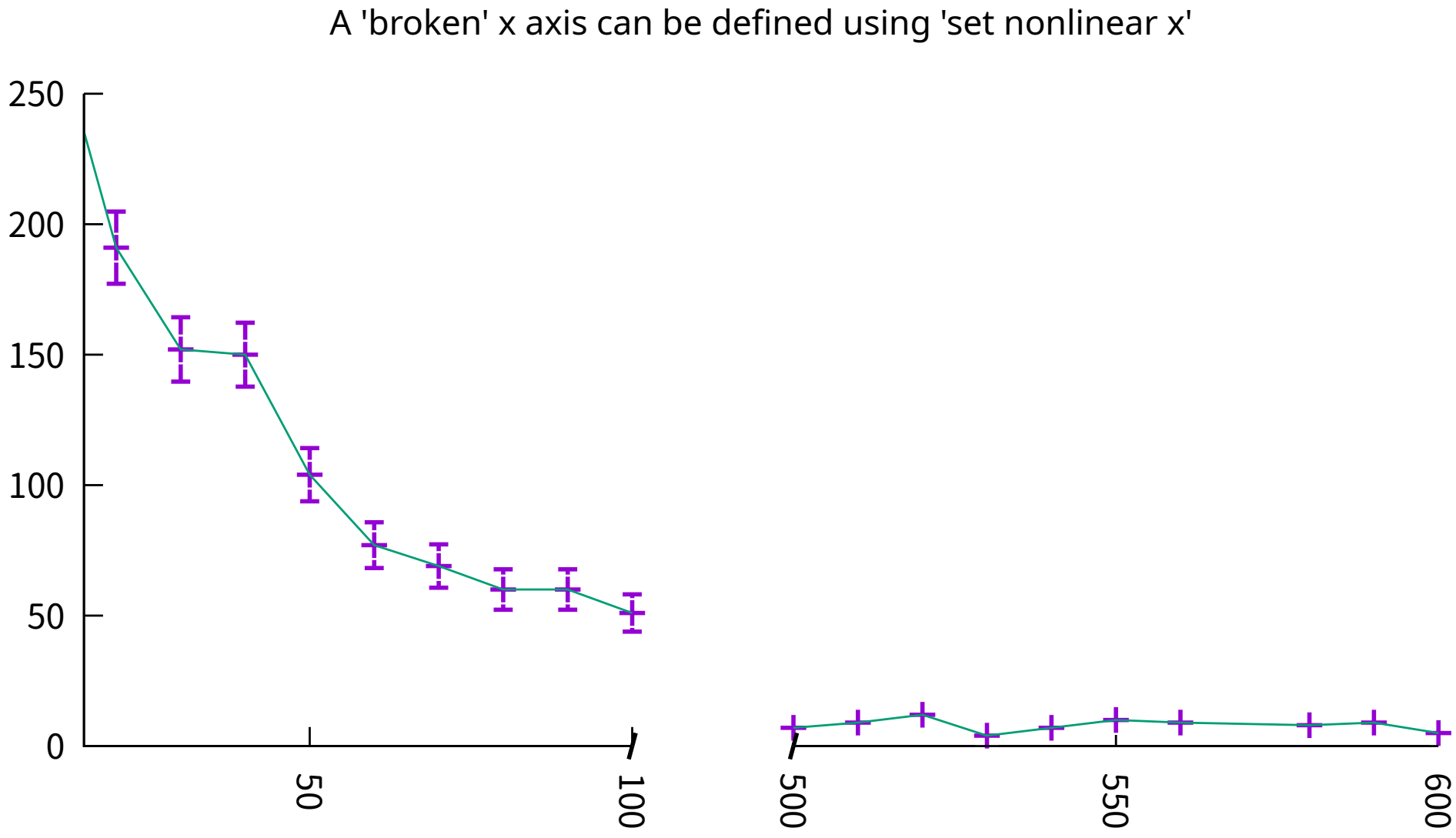


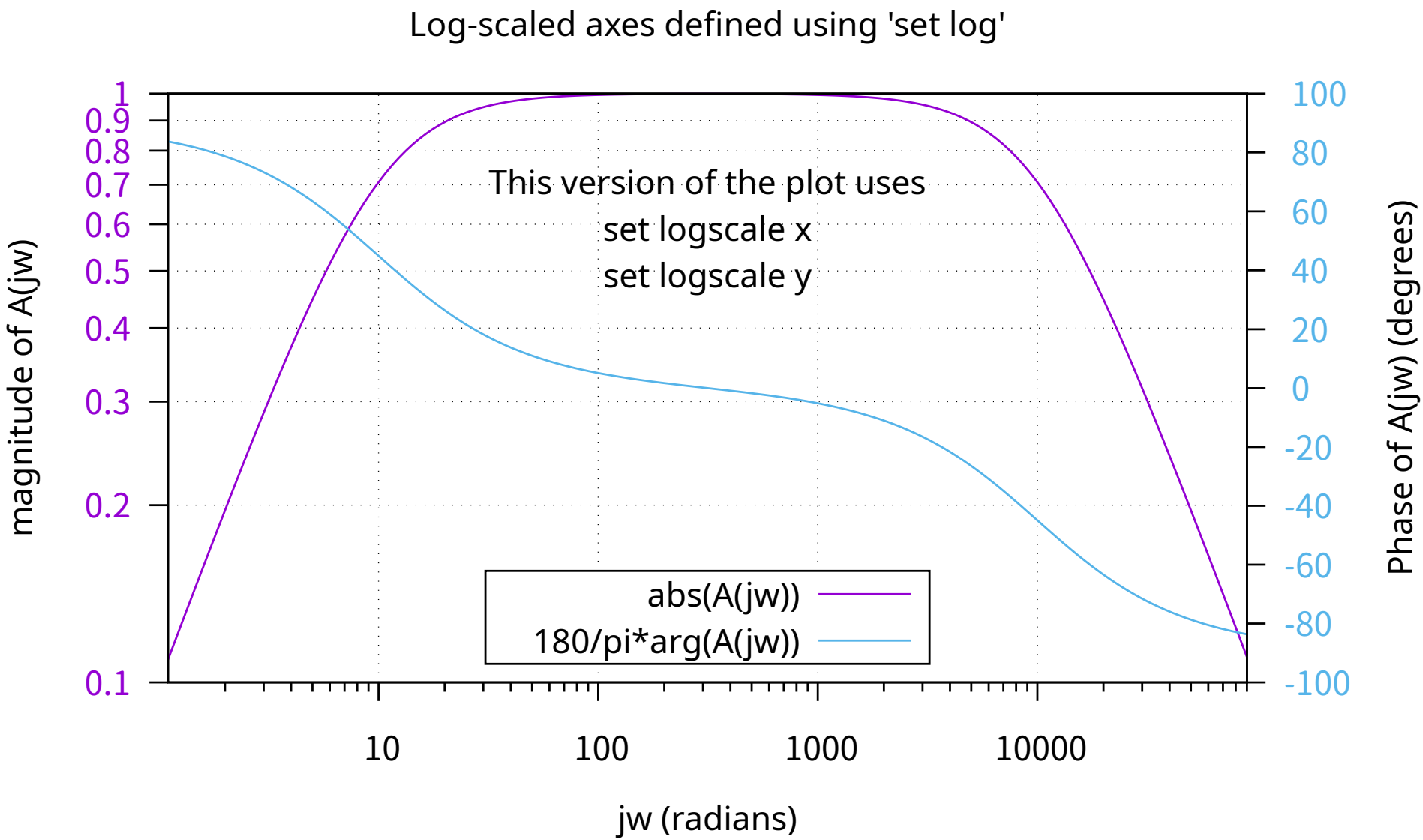
Parallel Axis Plot

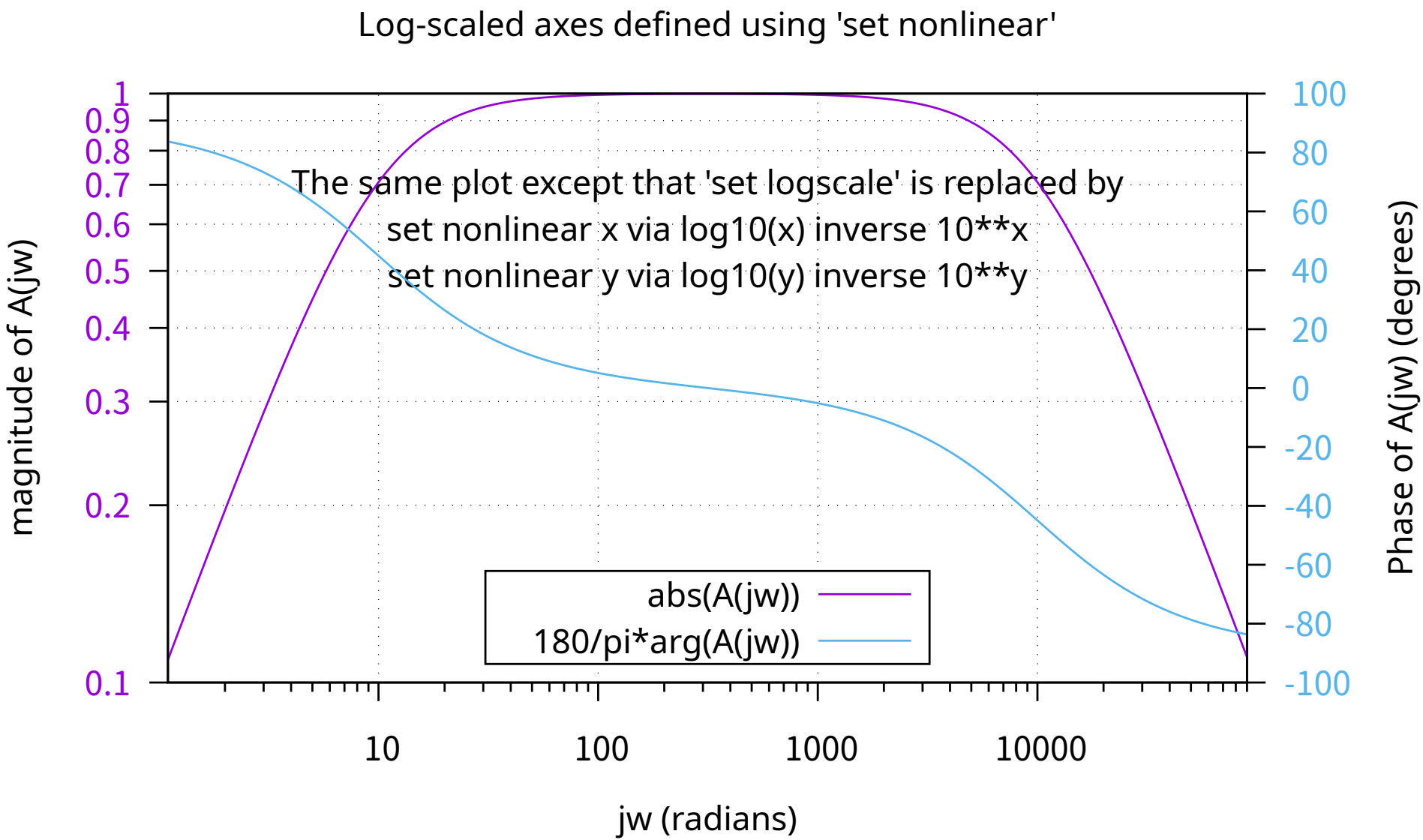


Parallel Axis Plot

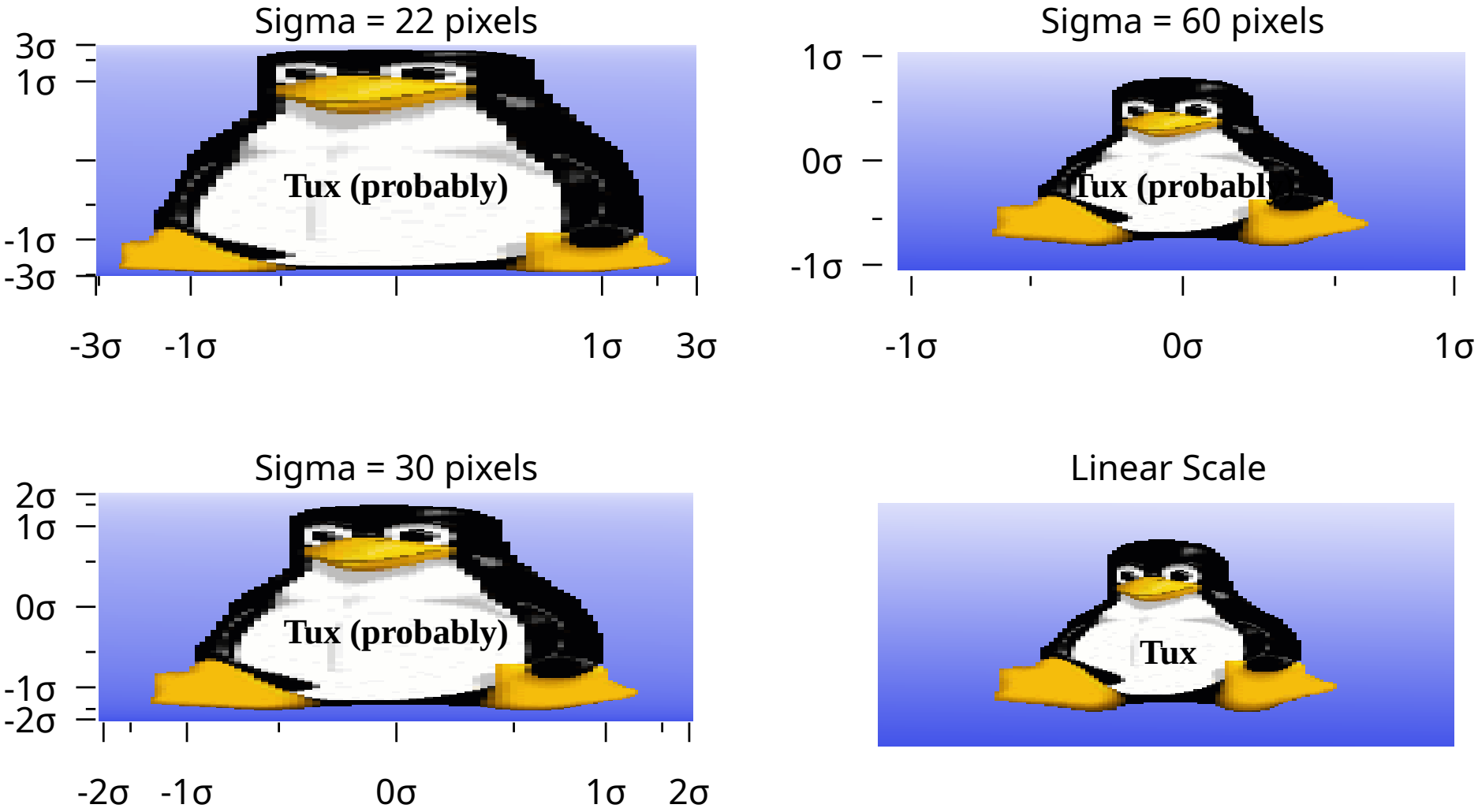


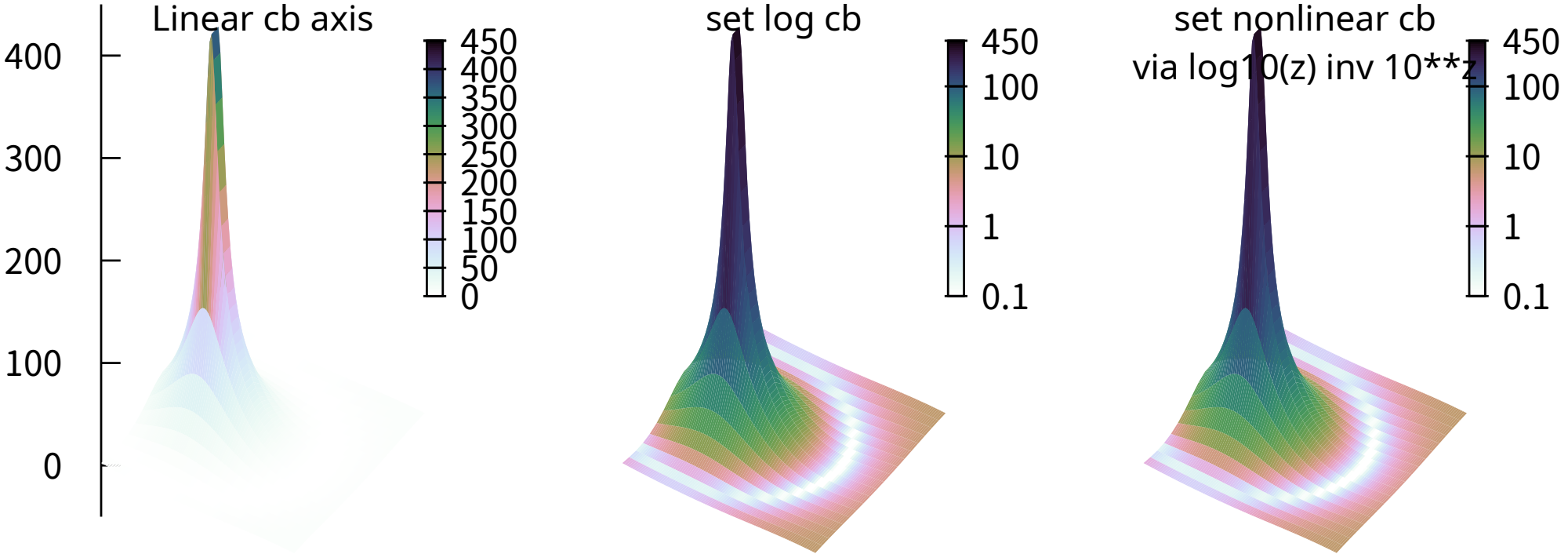

A 'broken' x axis can be defined using 'set nonlinear x'
250
200
150
100
50
0
50
100
500
550
600

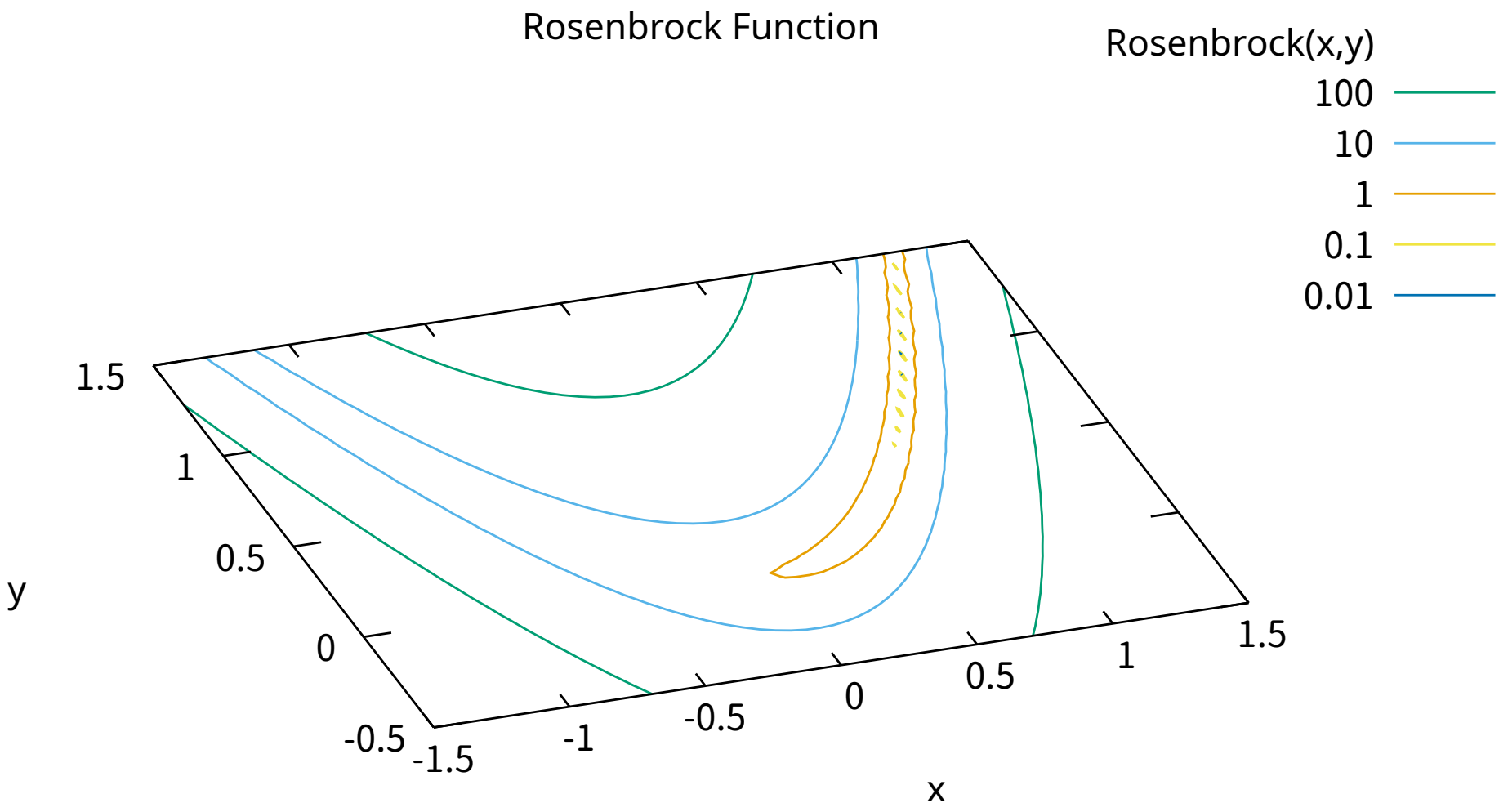

Log-scaled axes defined using 'set log'
This version of the plot uses
set logscale x
set logscale y
abs(A(jw))
180/pi*arg(A(jw))
magnitude of A(jw)
Phase of A(jw) (degrees)
jw (radians)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
100
80
60
40
20
0
-20
-40
-60
-80
-100
10
100
1000
10000

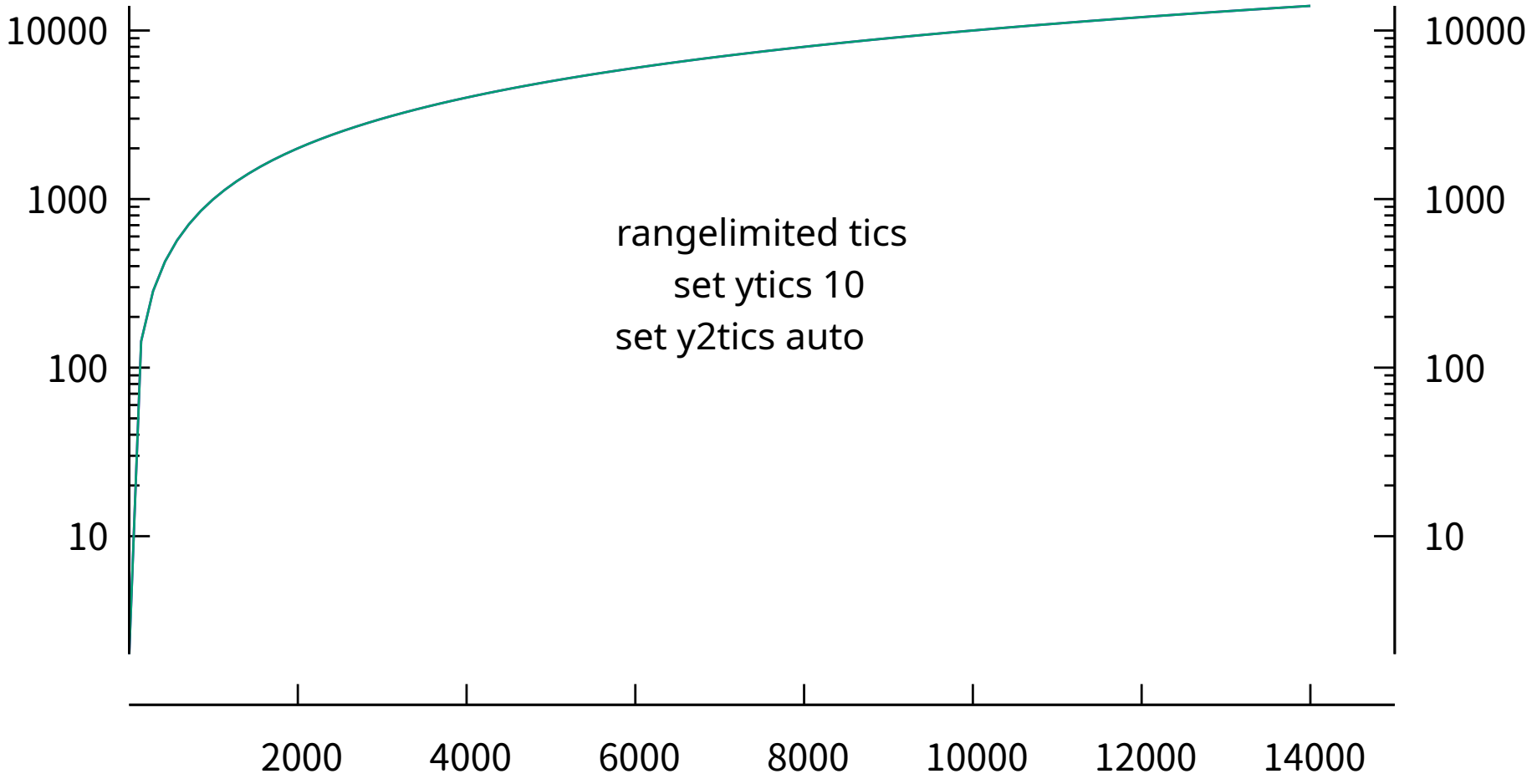

Log-scaled axes defined using 'set nonlinear'
The same plot except that 'set logscale' is replaced by
set nonlinear x via log10(x) inverse 10**x
set nonlinear y via log10(y) inverse 10**y
abs(A(jw))
180/pi*arg(A(jw))
magnitude of A(jw)
Phase of A(jw) (degrees)
jw (radians)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
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

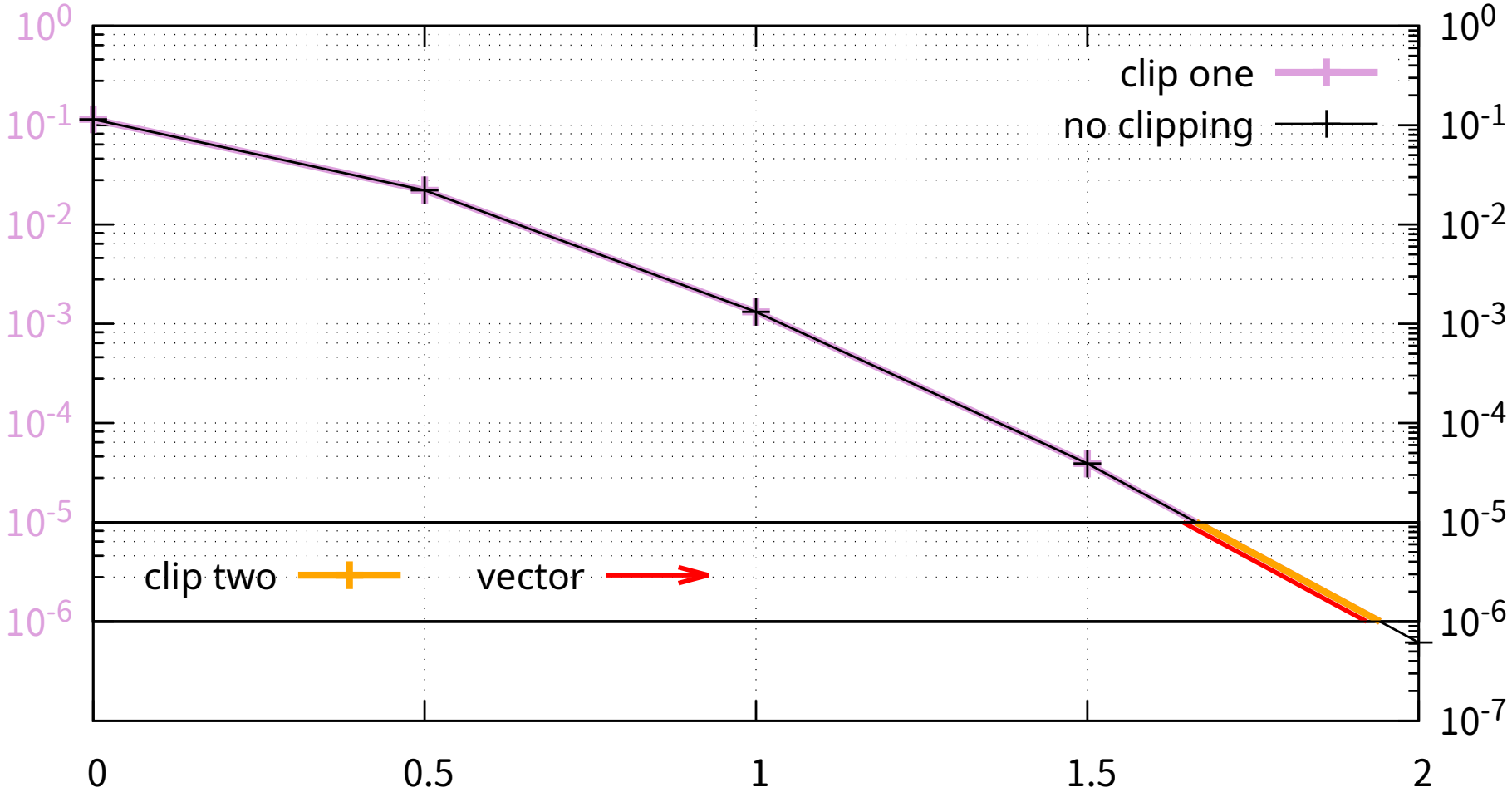


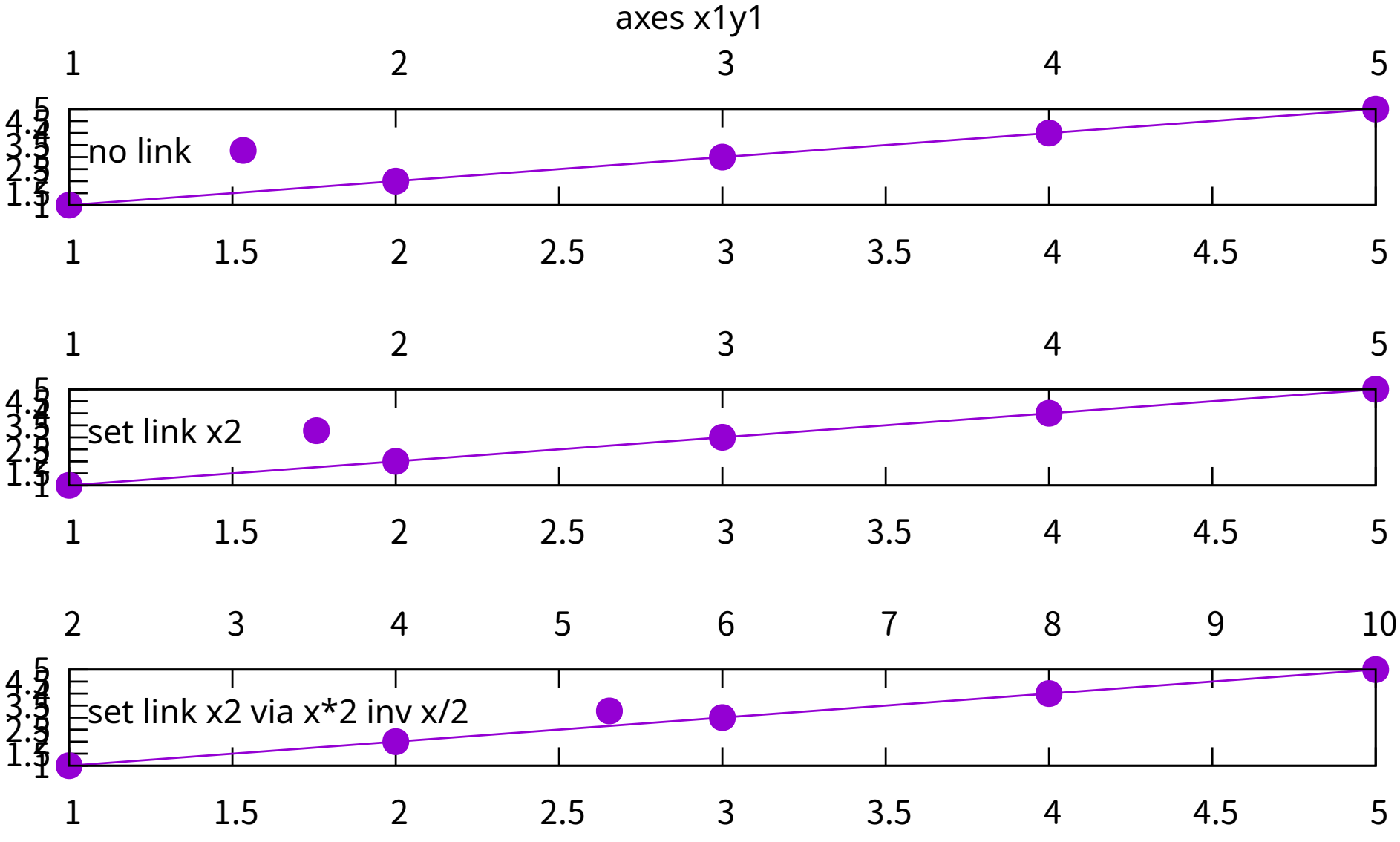

Linear cb axis
400
300
200
100
0
450
400
350
300
250
200
150
100
50
0
set log cb
450
100
10
1
0.1
set nonlinear cb
via log10(z) inv 10**z
450
100
10
1
0.1

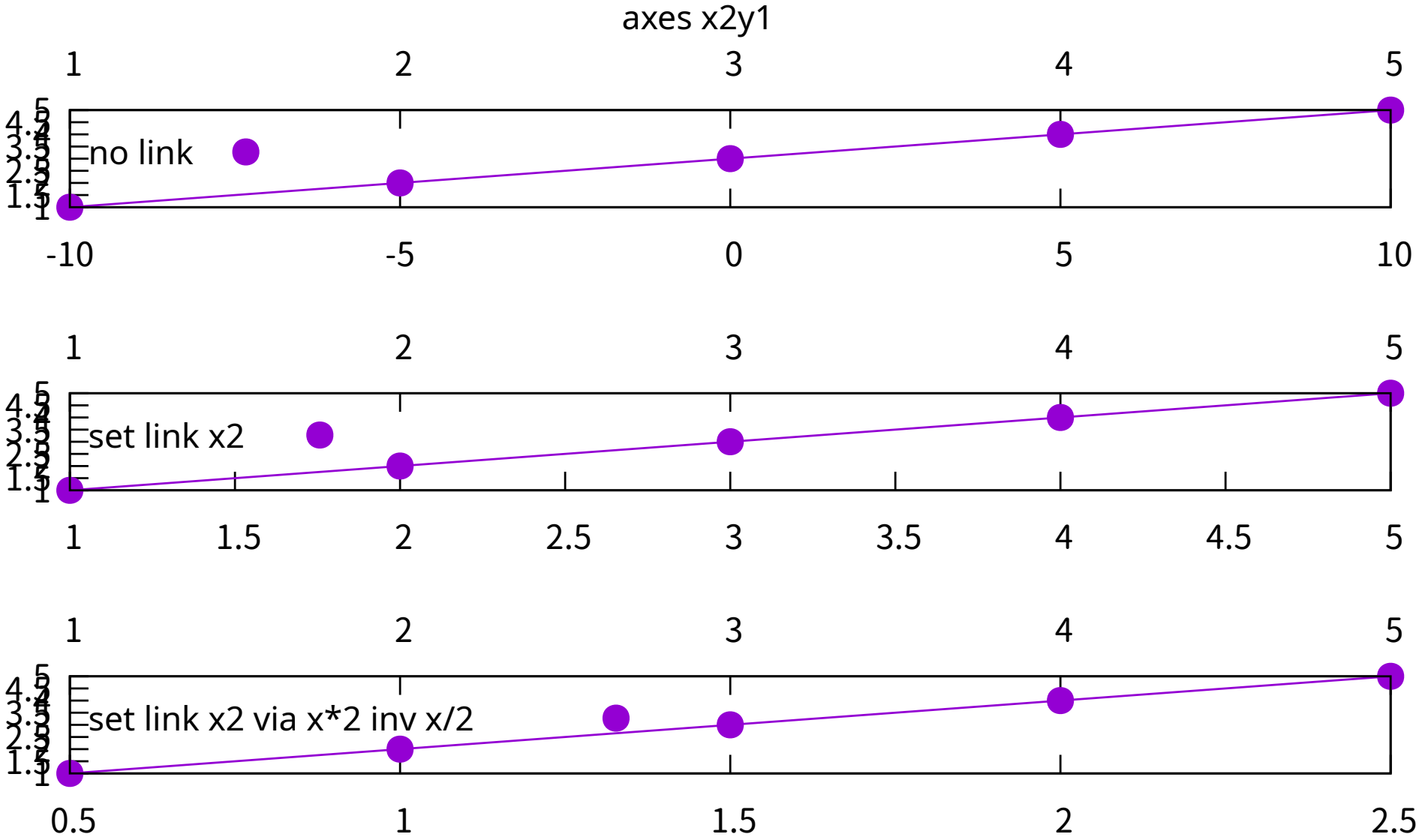

Rosenbrock Function
Rosenbrock(x,y)
100
10
1
0.1
0.01
1.5
1
0.5
0
-0.5
y
-1.5
-1
-0.5
0
0.5
1
1.5
x


10000
1000
100
10
rangelimited tics
set ytics 10
set y2tics auto
2000
4000
6000
8000
10000
12000
14000

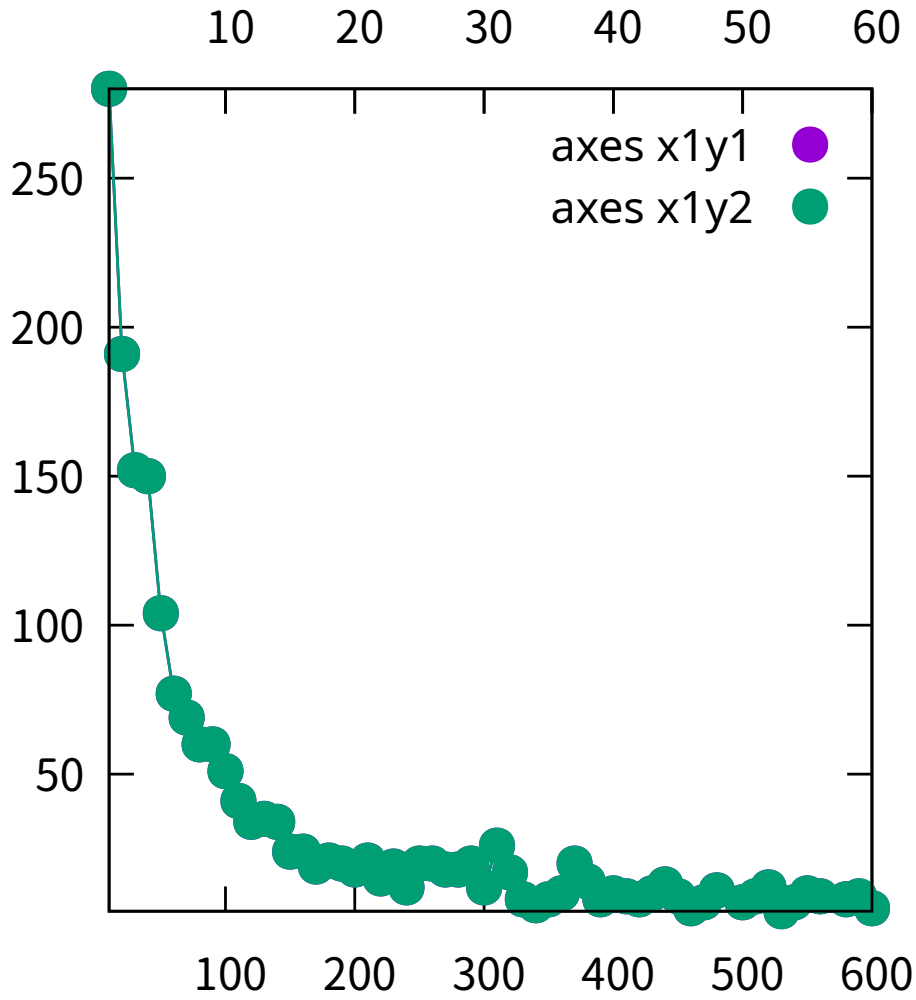
Bug #2046 - incorrect clipped line segments for logscale coordinates

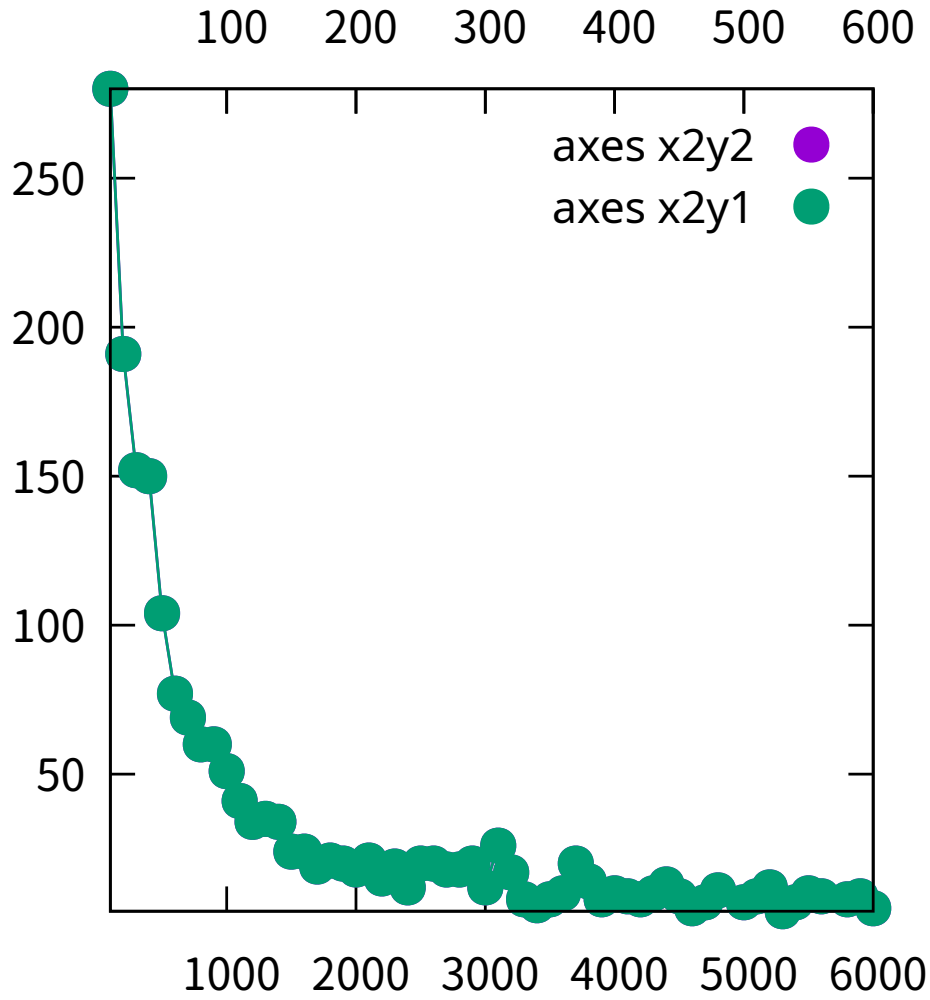



axes x1y1
no link
set link x2
set link x2 via x*2 inv x/2

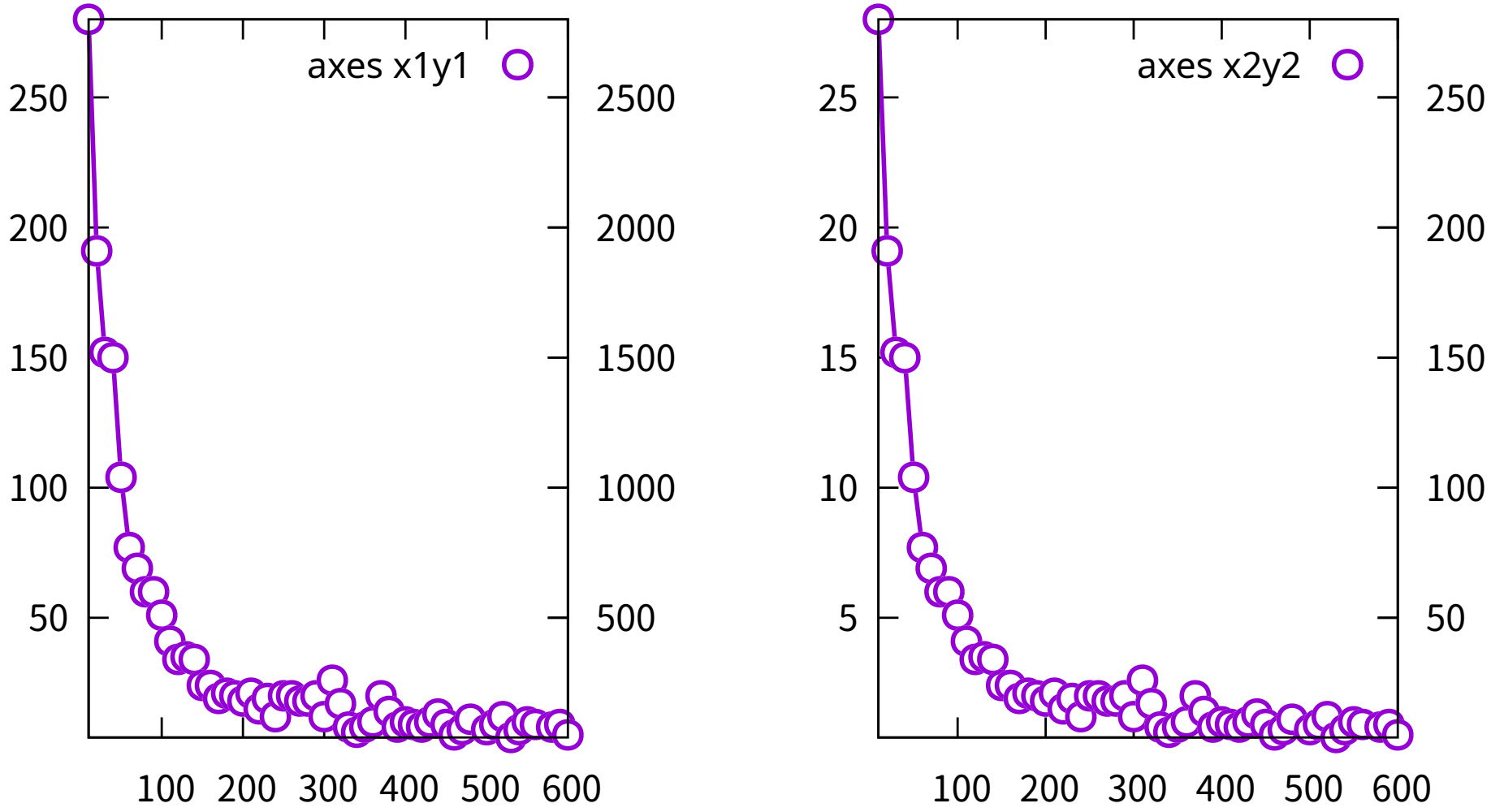

axes x2y1
no link
set link x2
set link x2 via x*2 inv x/2

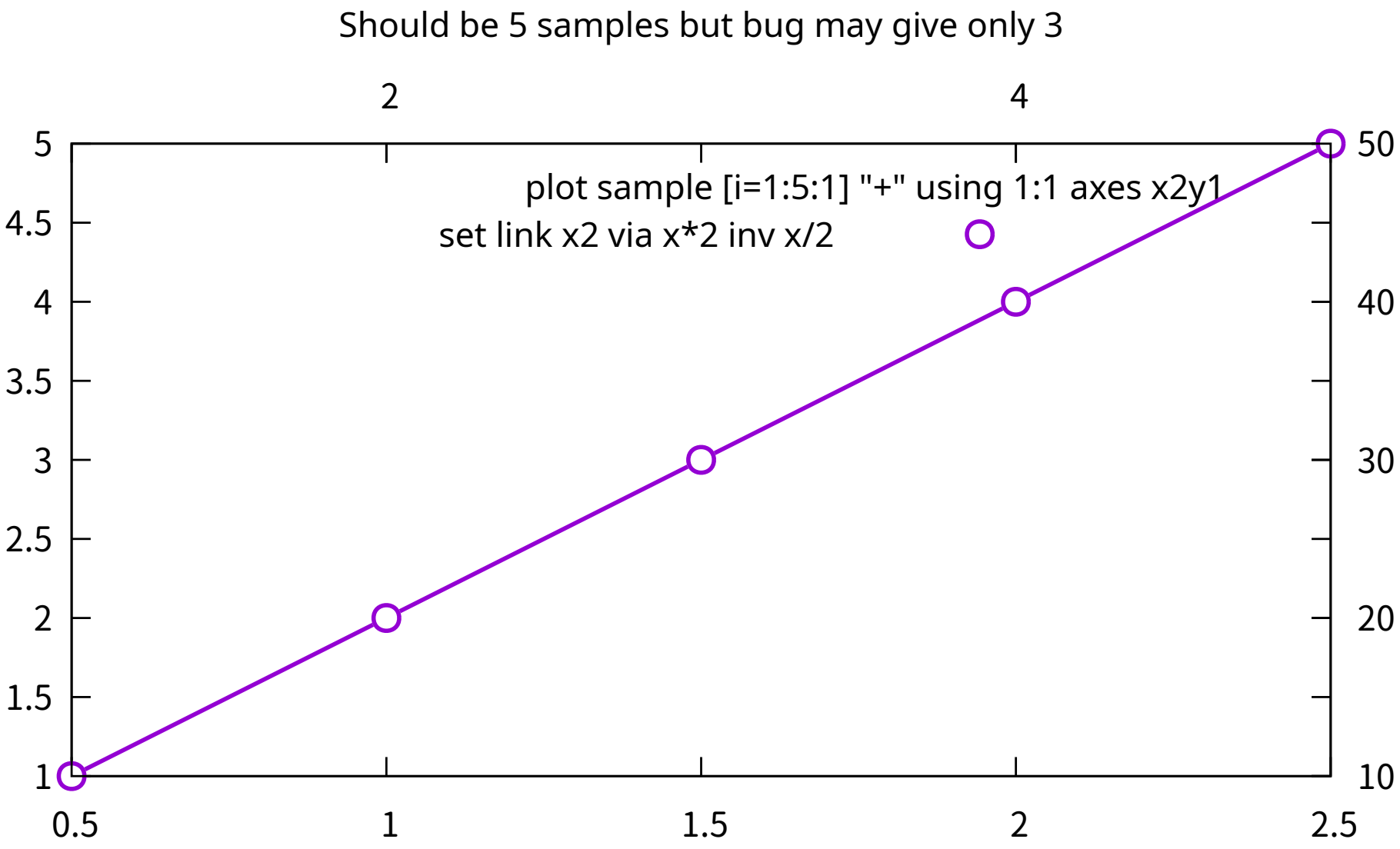
set link x2 via x/10. inv x*10
set link y2

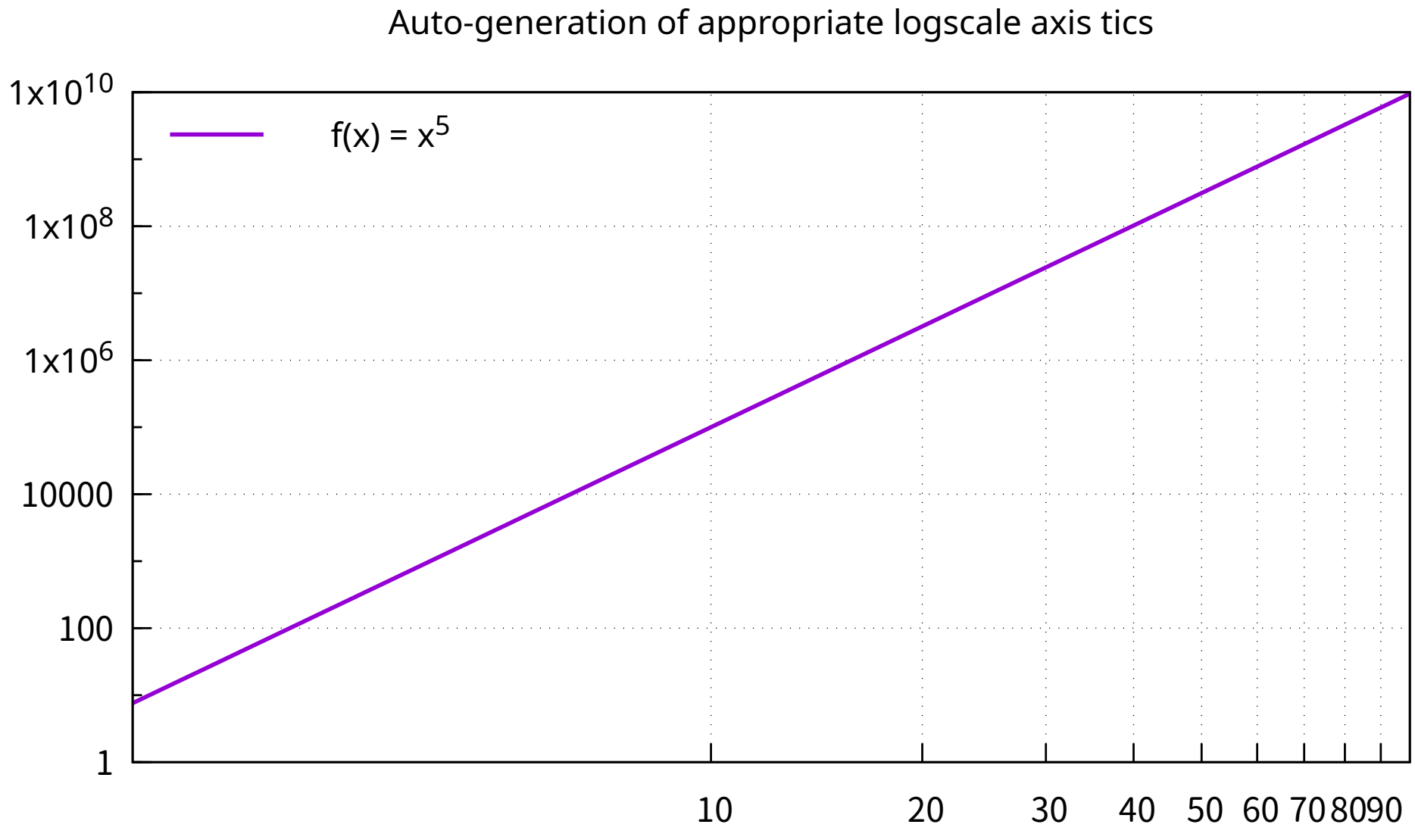




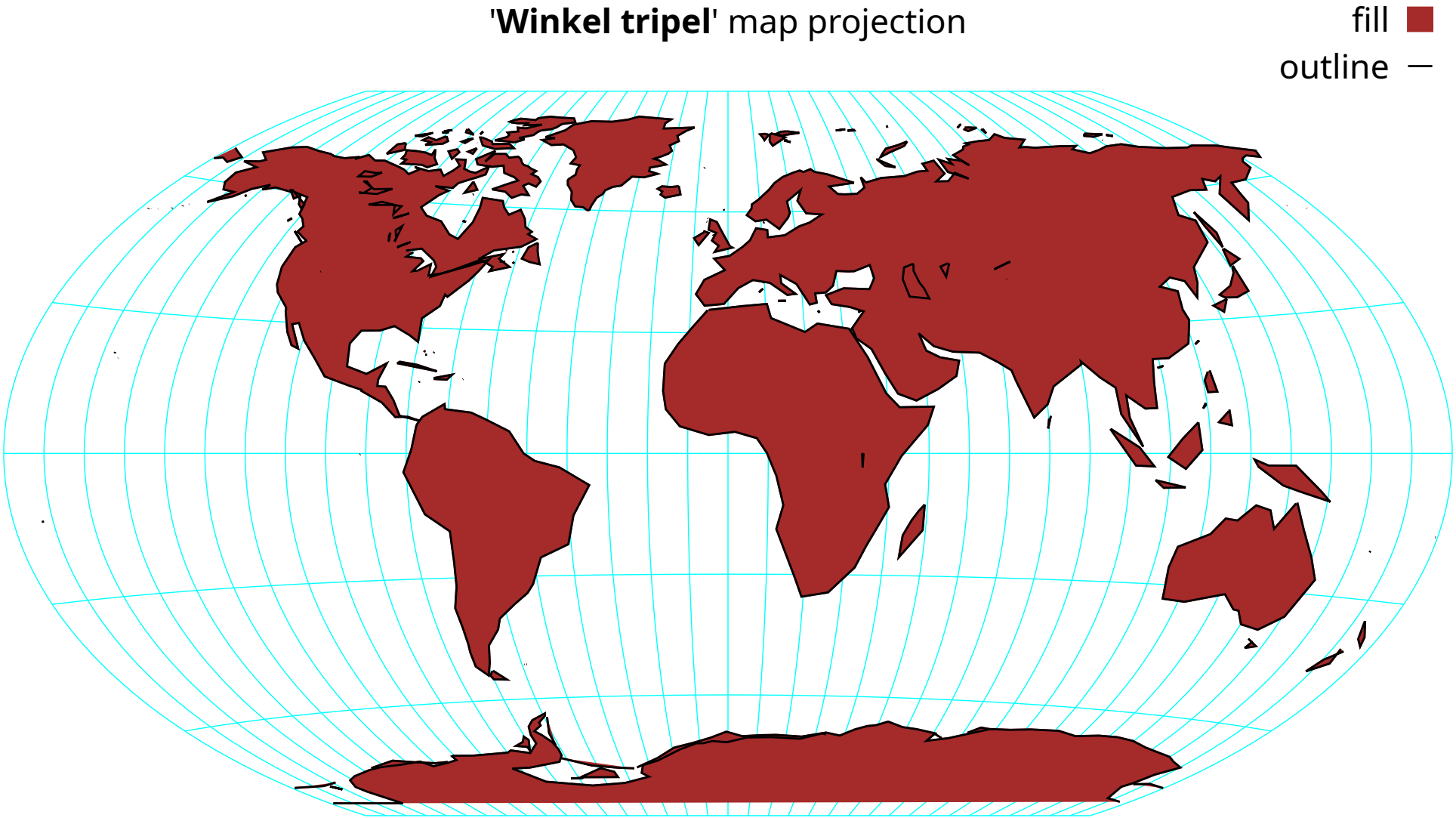
set link x2
set link y2 via y*10. inv y/10.




Should be 5 samples but bug may give only 3
plot sample [i=1:5:1] "+" using 1:1 axes x2y1
set link x2 via x*2 inv x/2
2
4
5
4.5
4
3.5
3
2.5
2
1.5
1
50
40
30
20
10
0.5
1
1.5
2
2.5

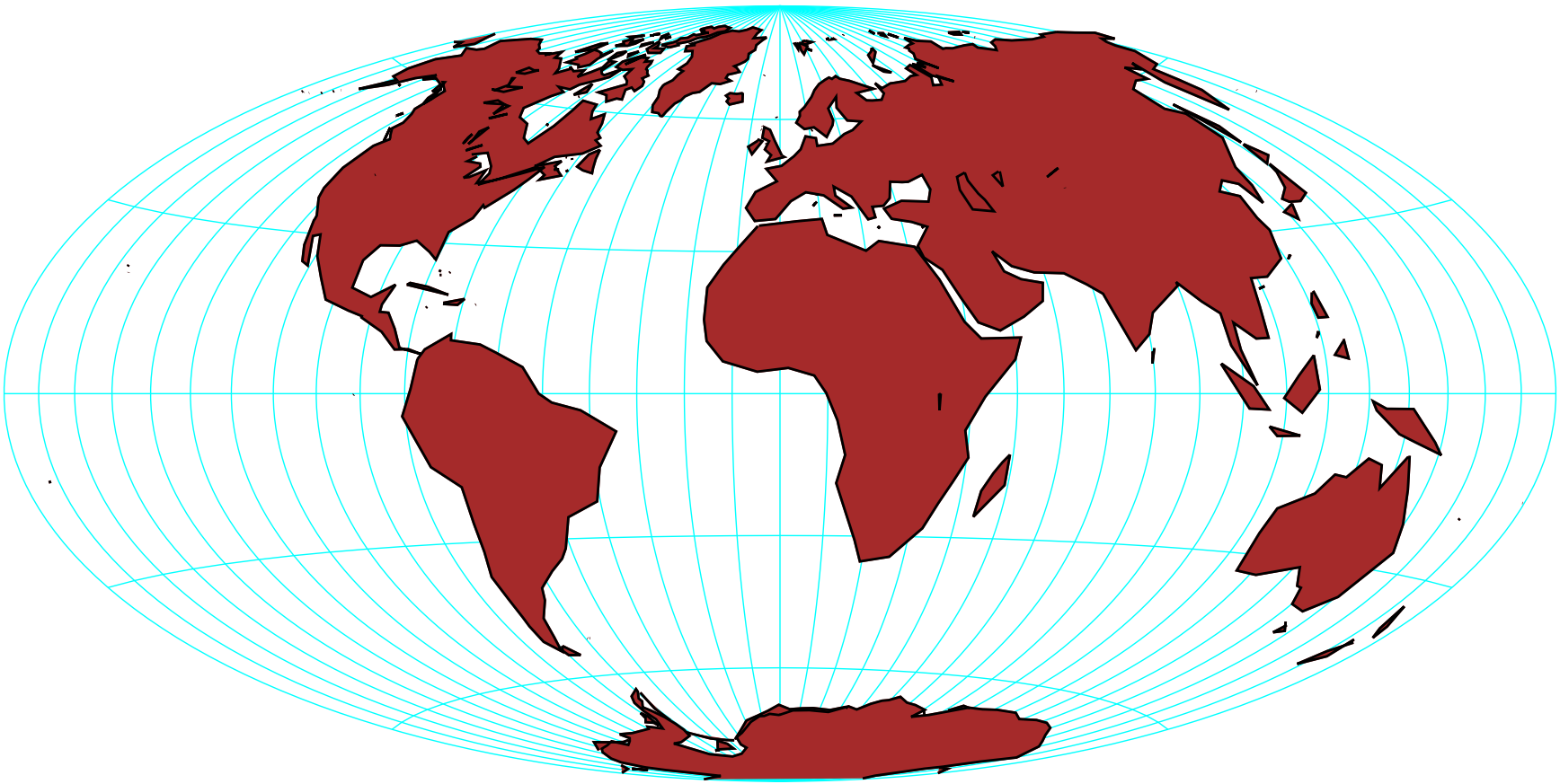

Auto-generation of appropriate logscale axis tics
f(x) = x^5
1x10^10
1x10^8
1x10^6
10000
100
1
10
20
30
40
50
60
70
80
90

'**Winkel tripel**' map projection



Hammer equal-area map projection

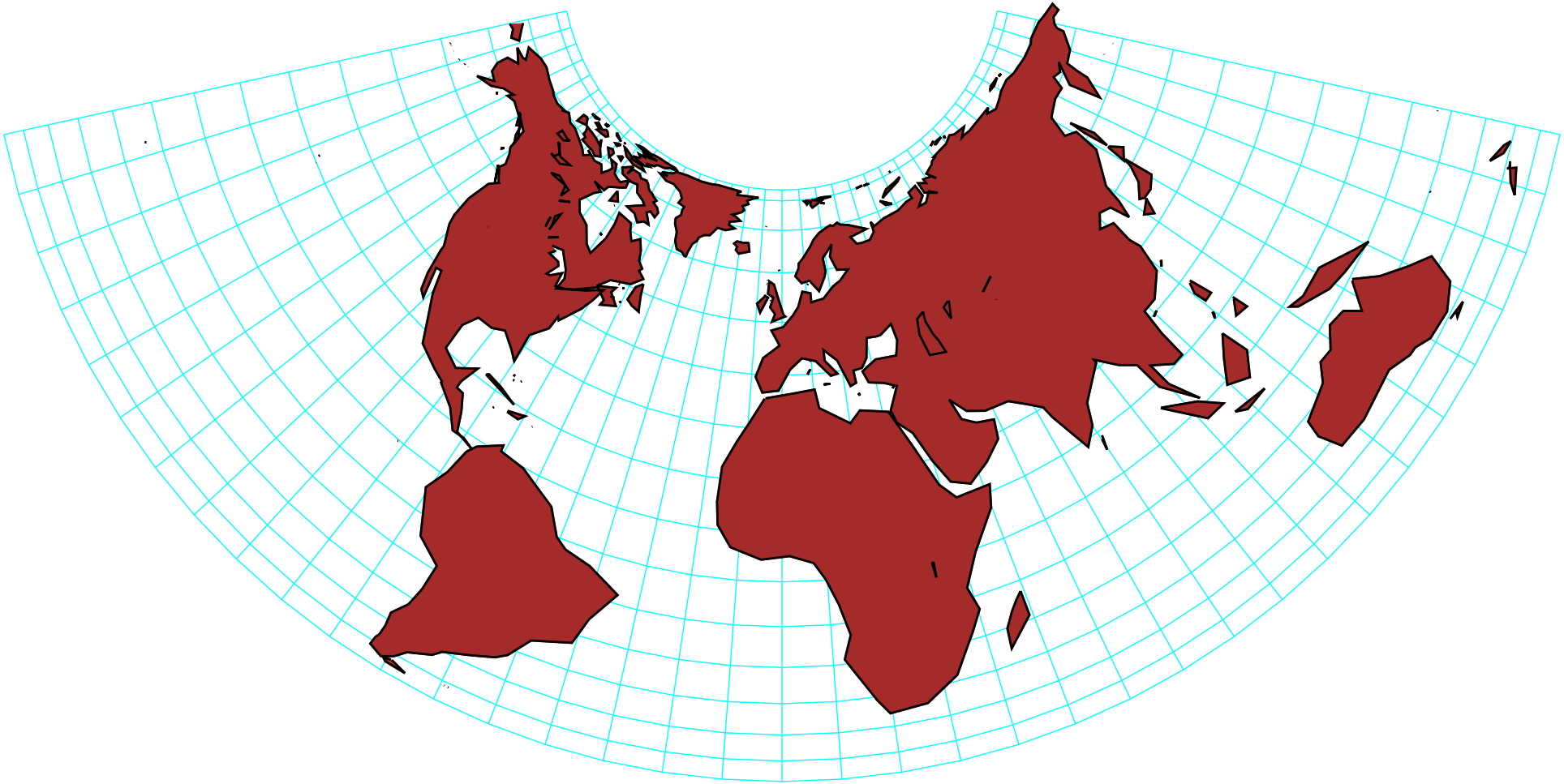
fill
outline

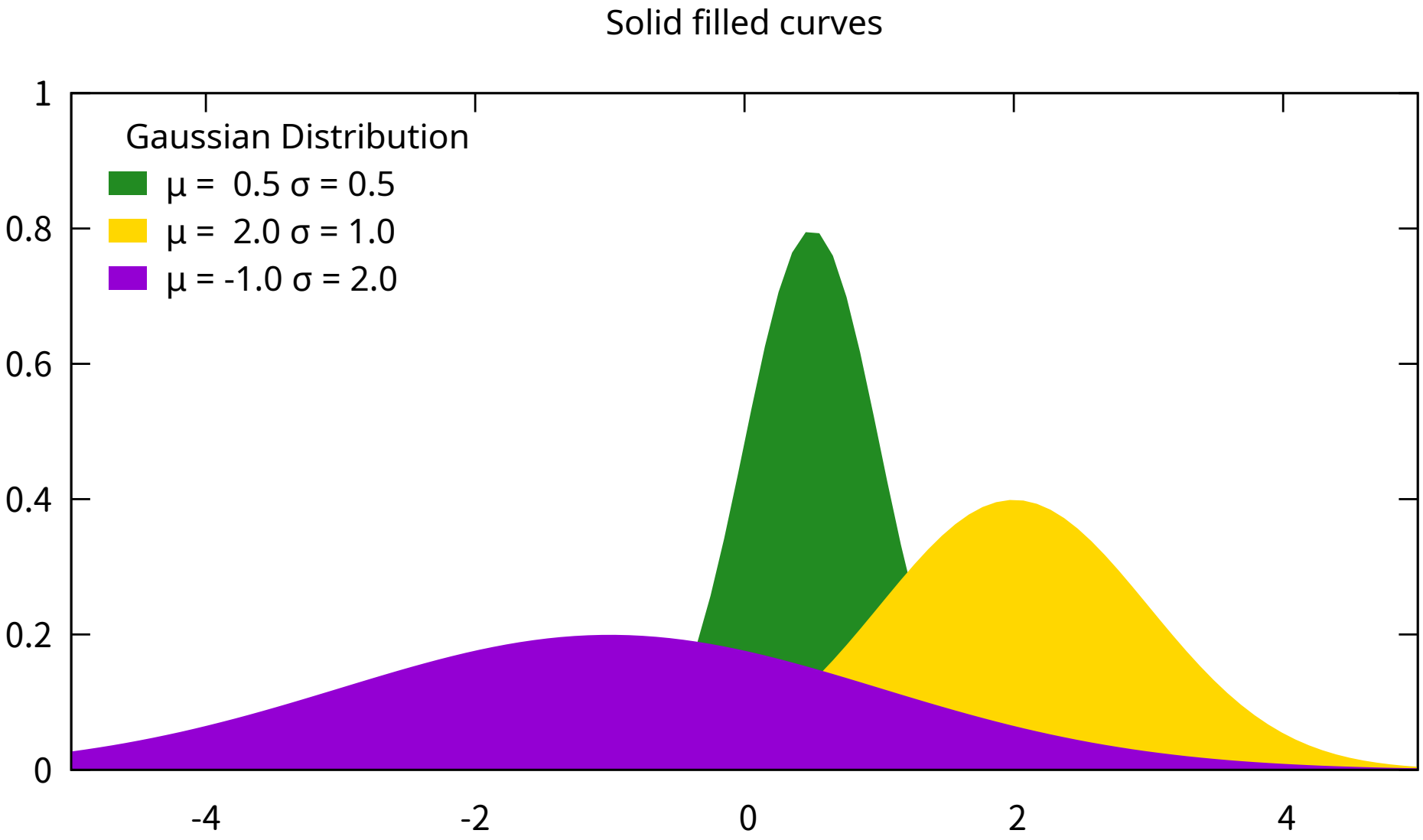


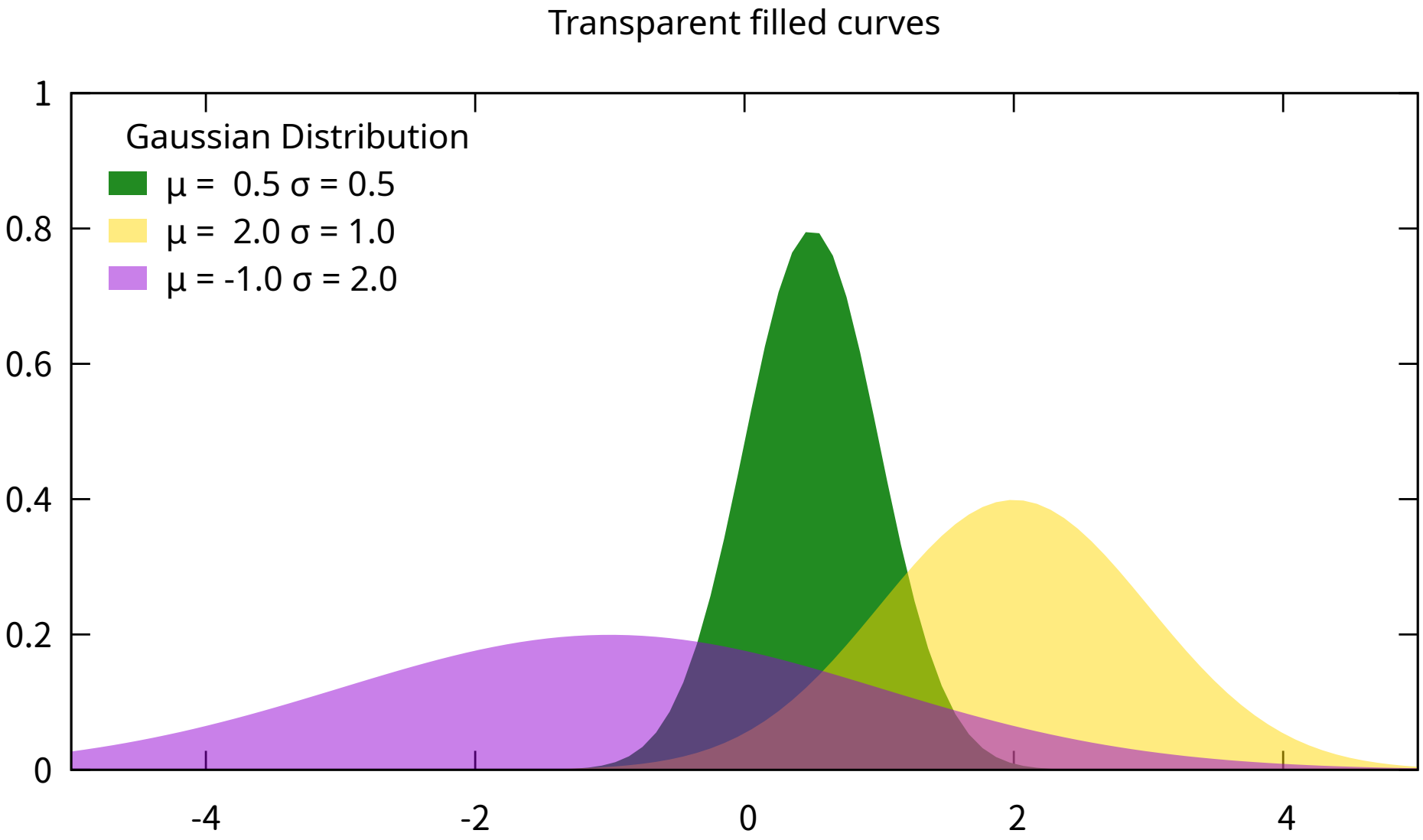
Albers equal-area conic projection

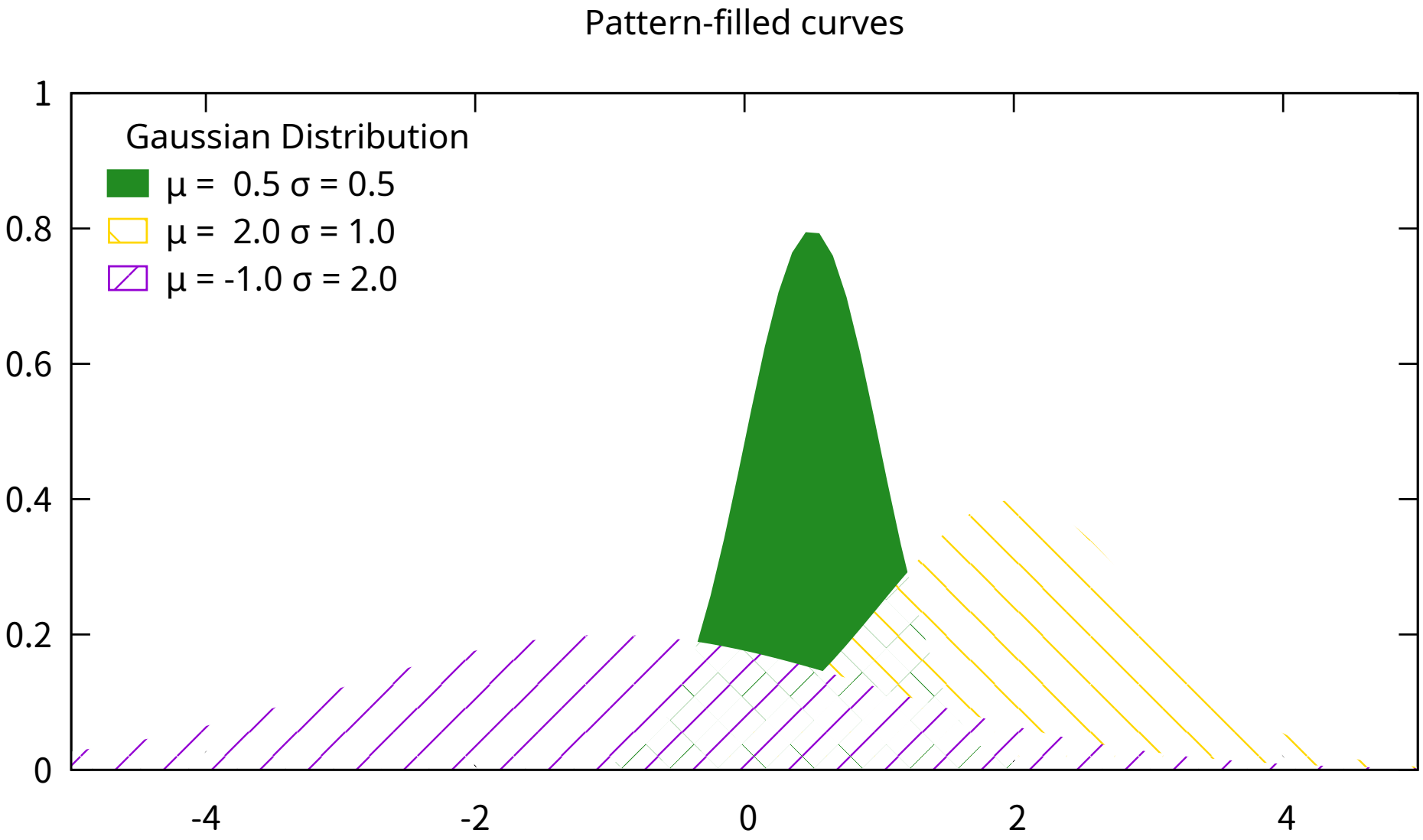
fill

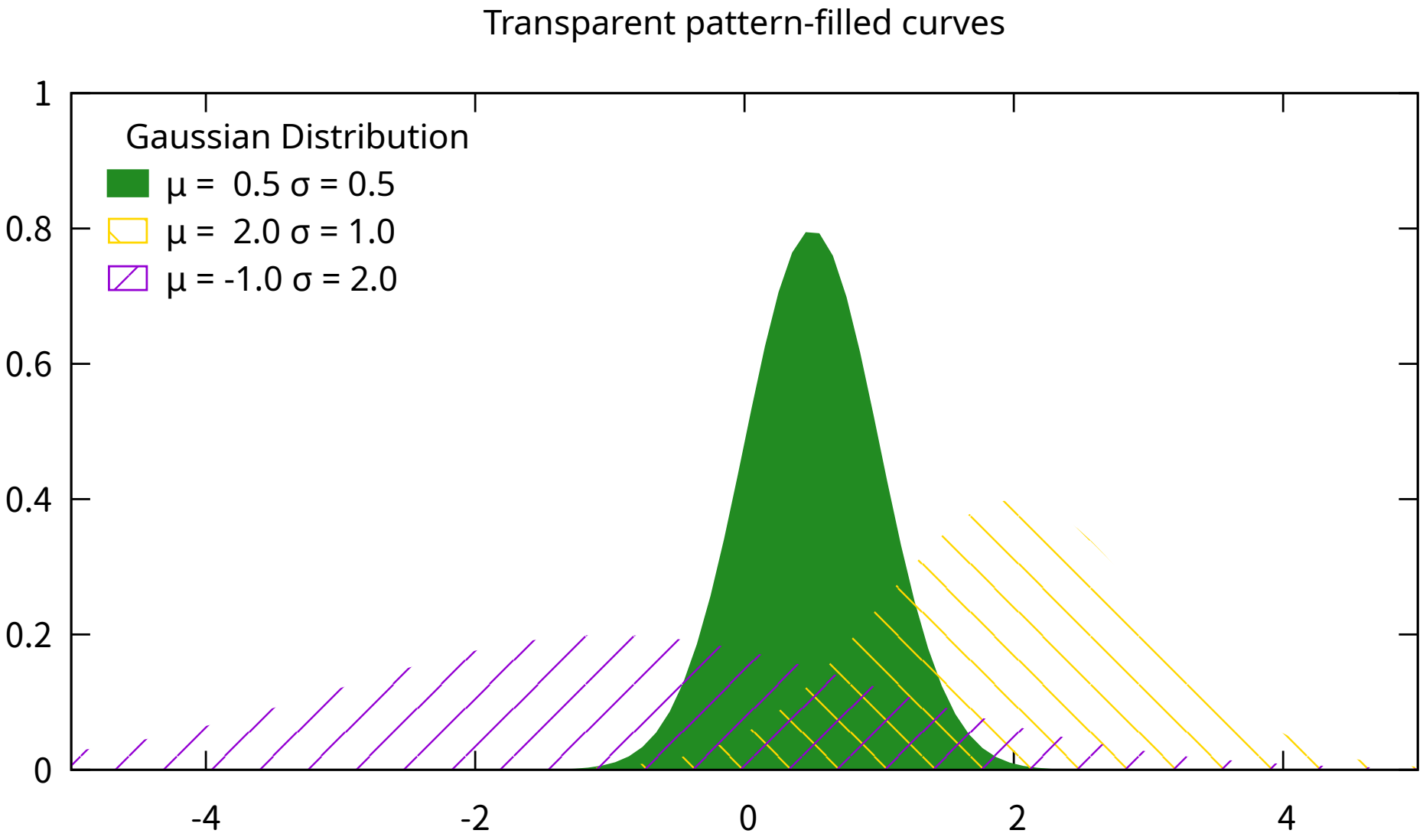
outline



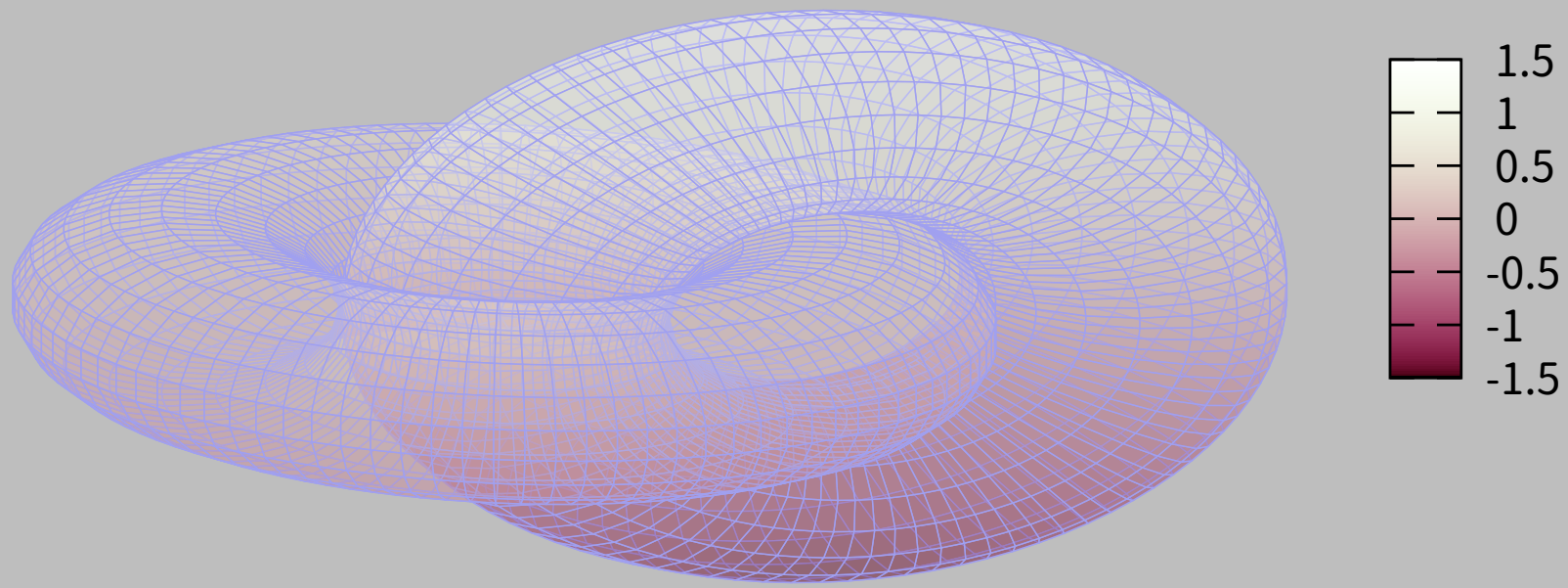

Solid filled curves
Gaussian Distribution
μ = 0.5 σ = 0.5
μ = 2.0 σ = 1.0
μ = -1.0 σ = 2.0
1
0.8
0.6
0.4
0.2
0
-4
-2
0
2
4

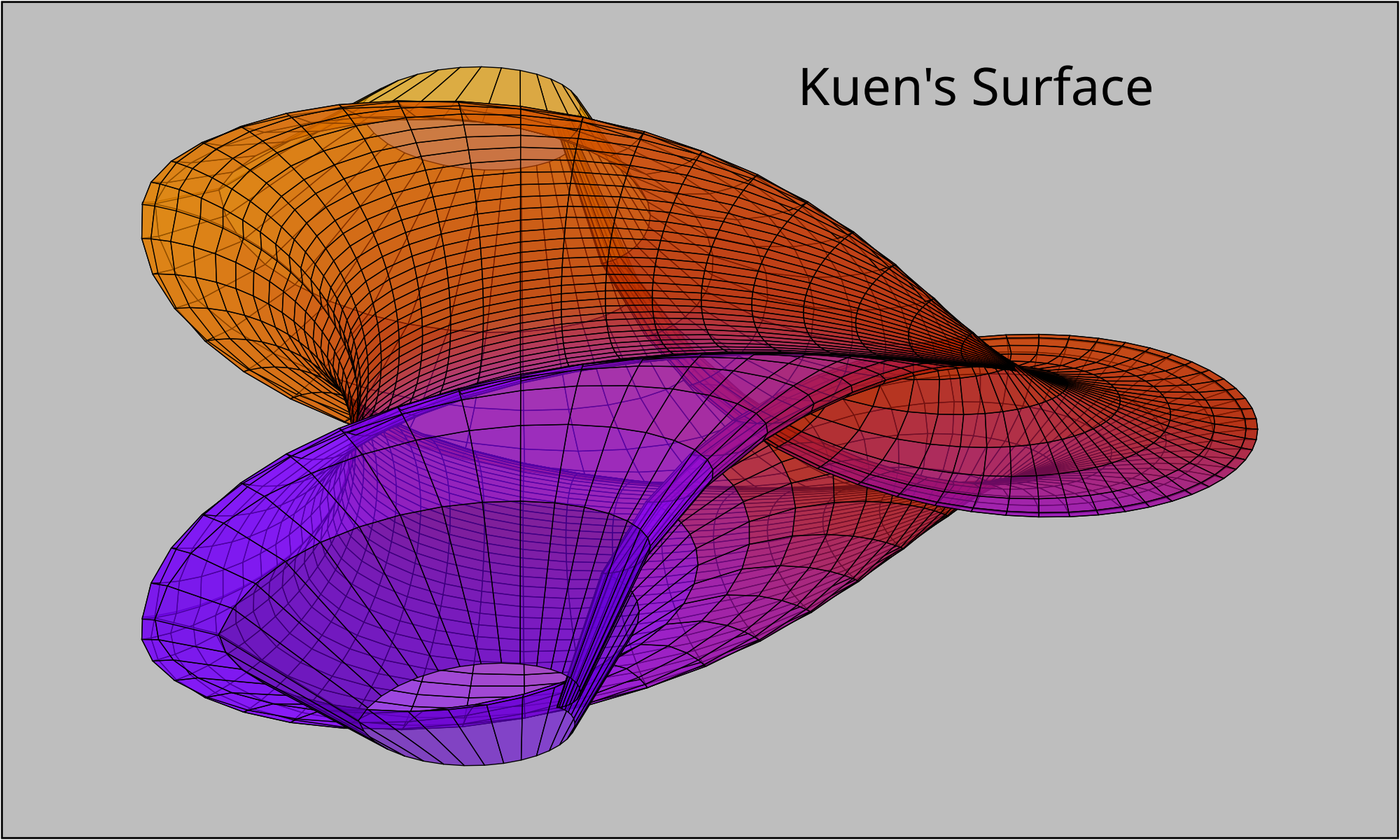

Transparent filled curves
Gaussian Distribution
μ = 0.5 σ = 0.5
μ = 2.0 σ = 1.0
μ = -1.0 σ = 2.0
1
0.8
0.6
0.4
0.2
0
-4
-2
0
2
4


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

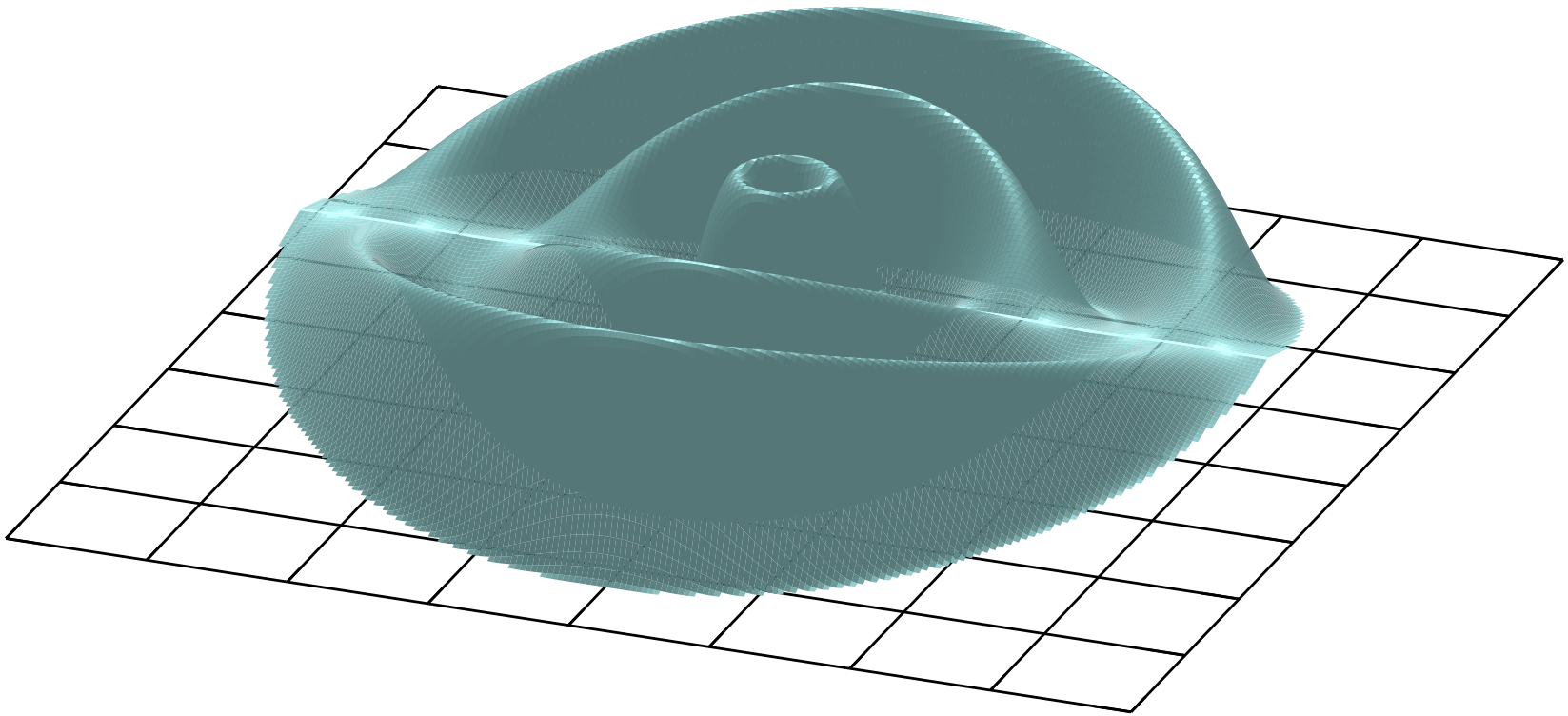

Transparent pattern-filled curves
Gaussian Distribution
μ = 0.5 σ = 0.5
μ = 2.0 σ = 1.0
μ = -1.0 σ = 2.0
1
0.8
0.6
0.4
0.2
0
-4
-2
0
2
4

Interlocking Tori - PM3D surface with depth sorting and transparency

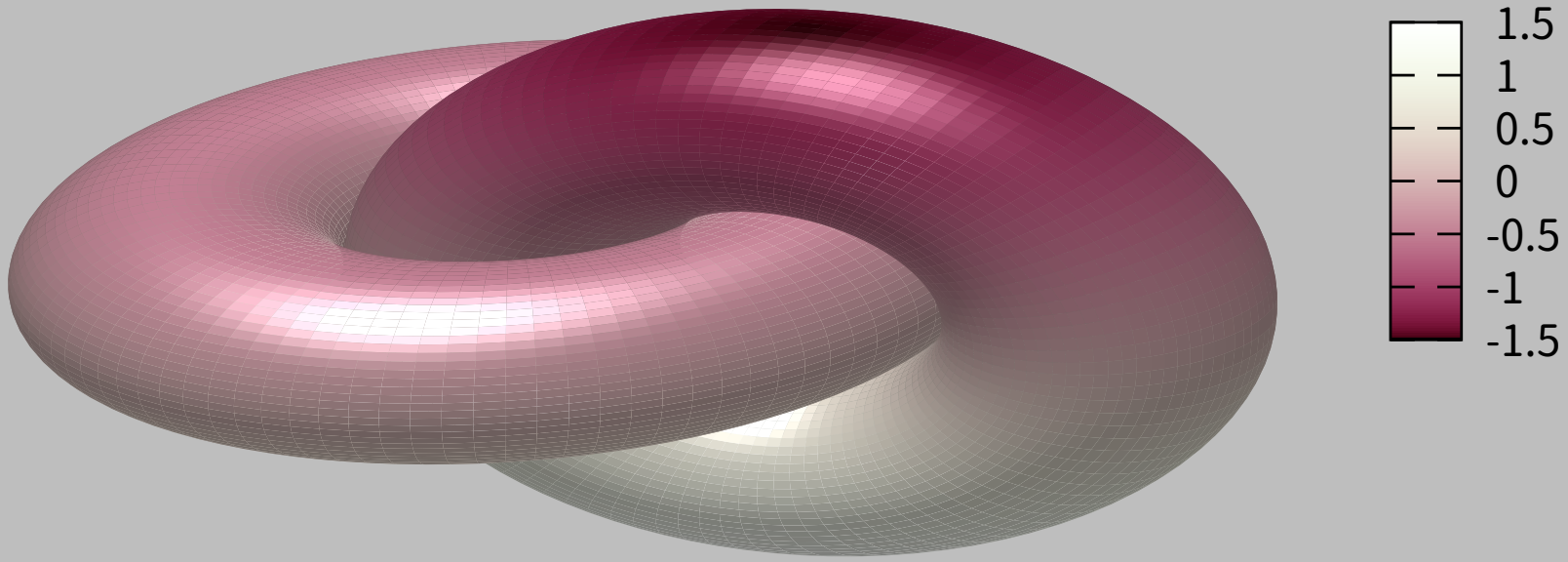



Kuen's Surface

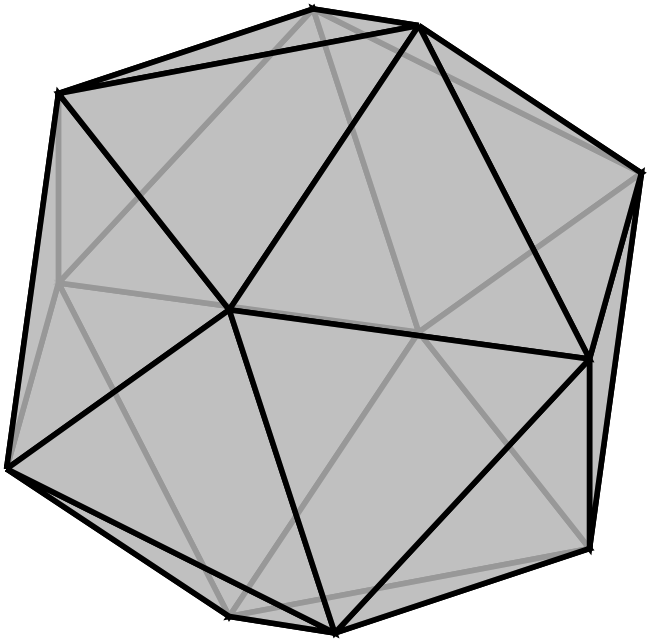
pm3d lighting model with specular highlighting



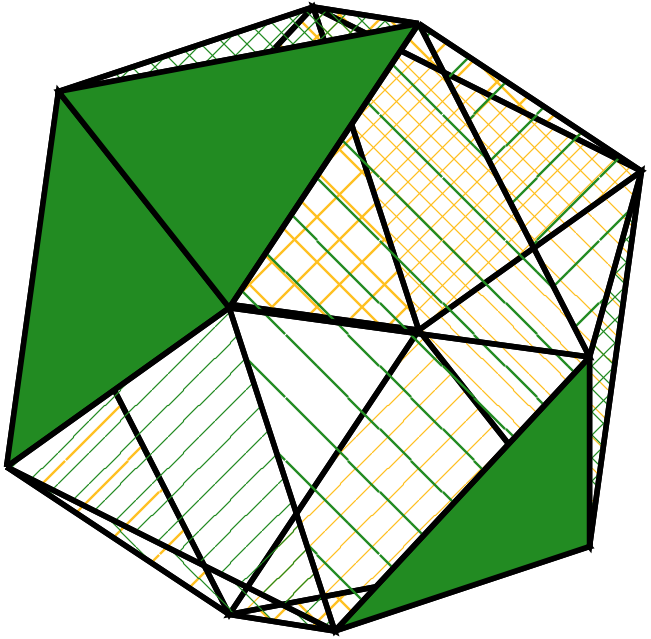
PM3D surfaces with specular highlighting



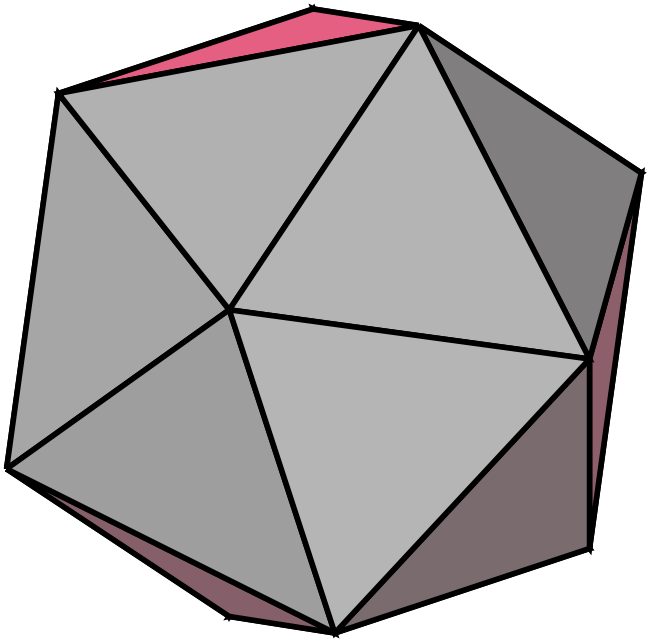
Faces of an icosahedron drawn as 20 individual objects



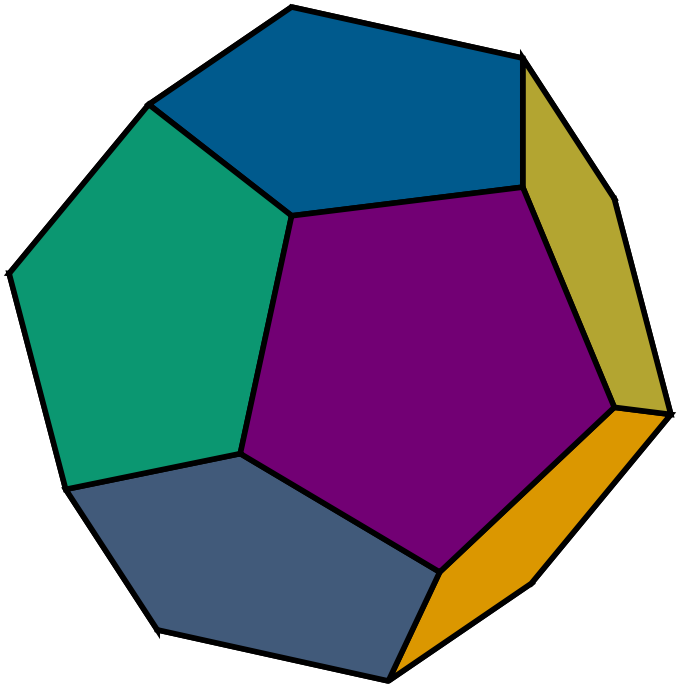
2-sided coloring
green outside, yellow inside



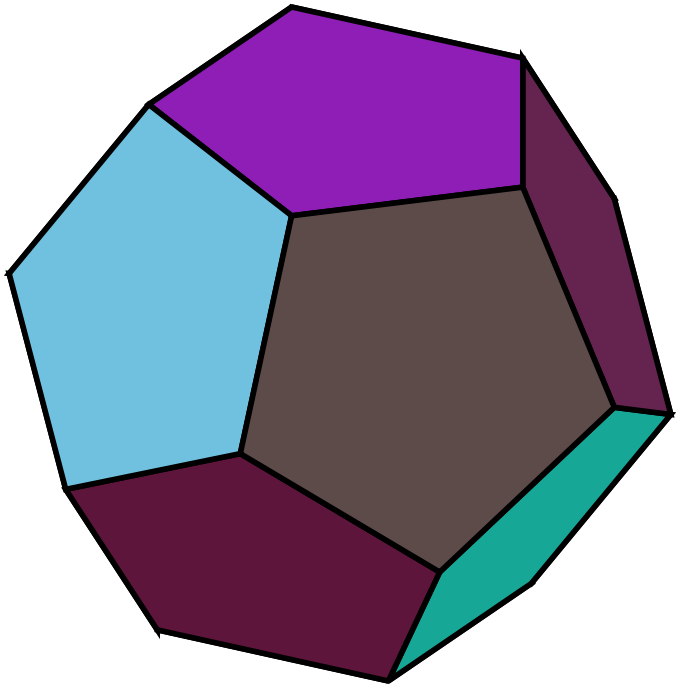
splot icosahedron.dat with polygons



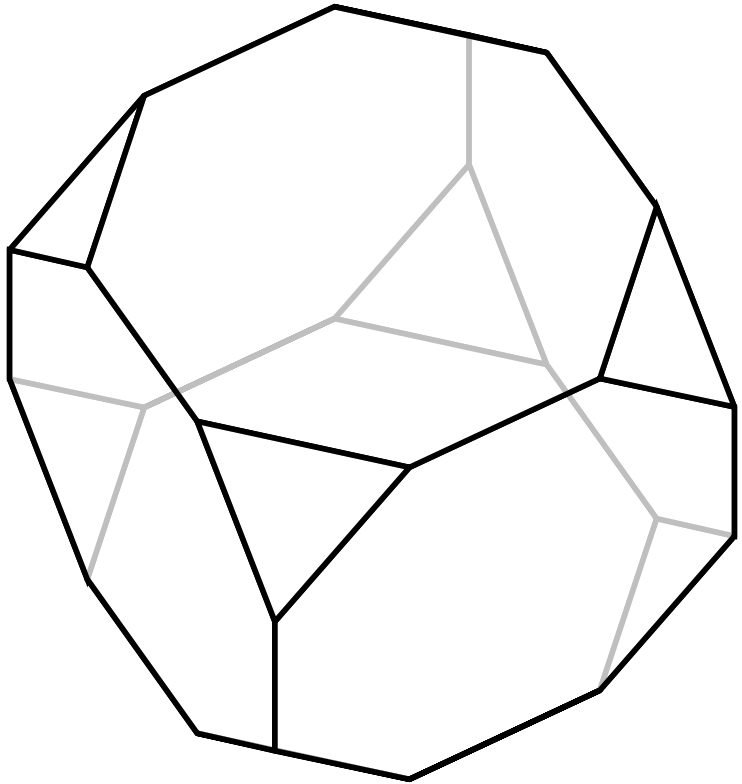
splot dodecahedron.dat with polygons fc variable



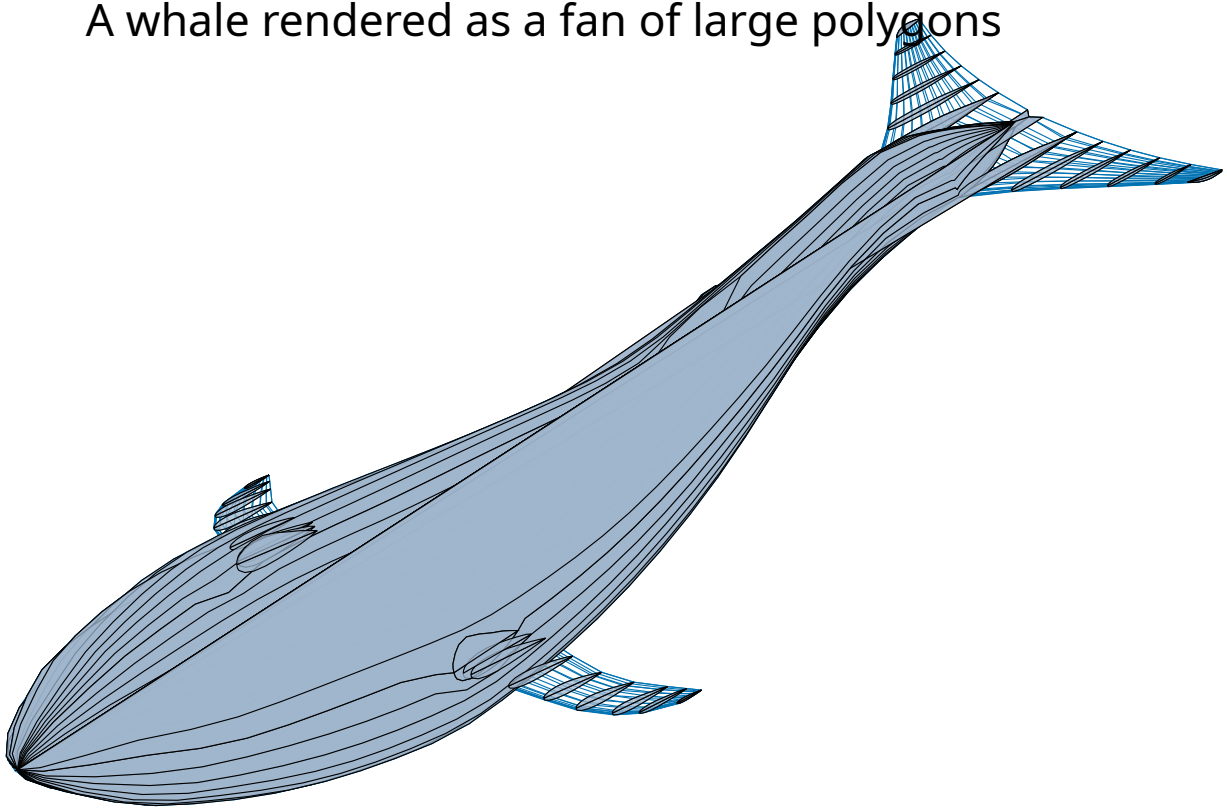
splot dodecahedron with polygons fc rgb variable



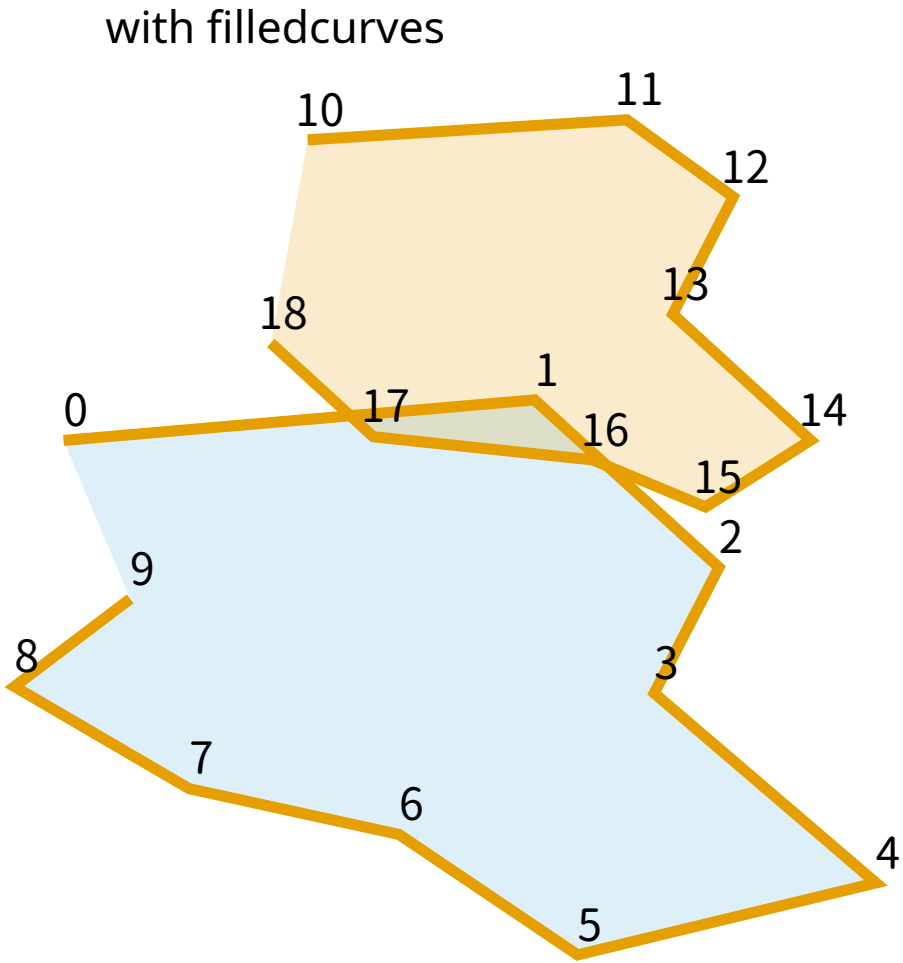
splot truncated_cube with polygons

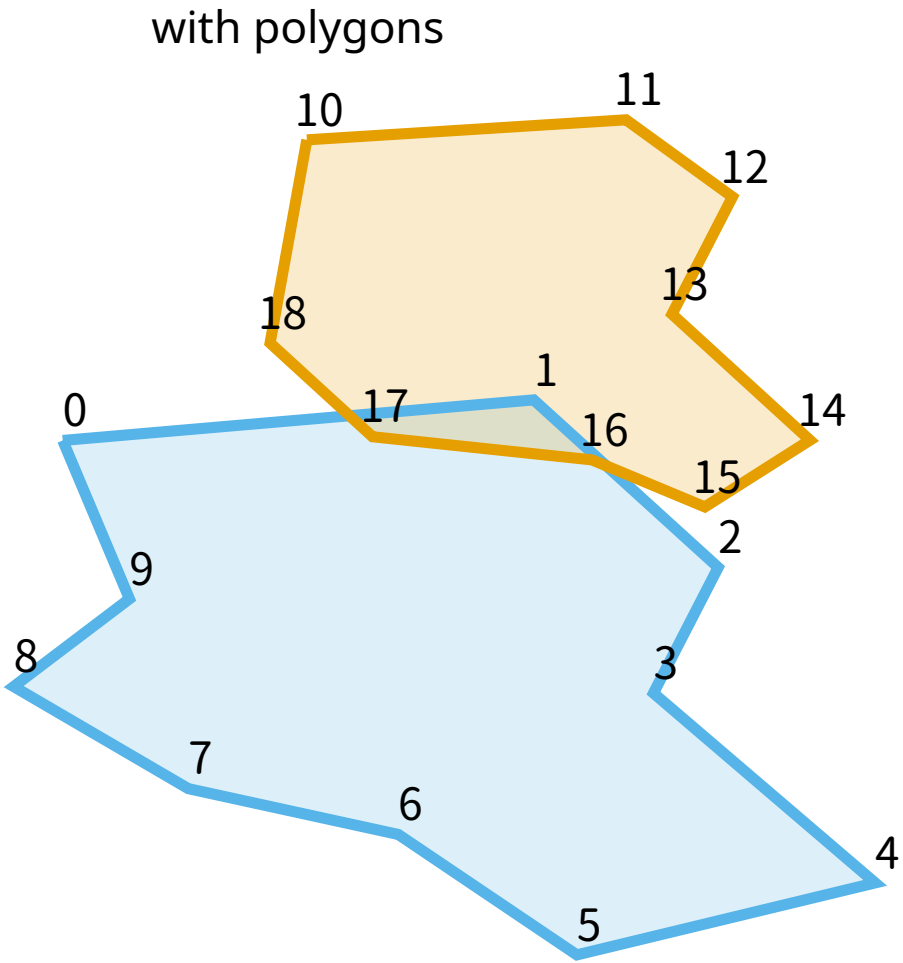


A whale rendered as a fan of large polygons



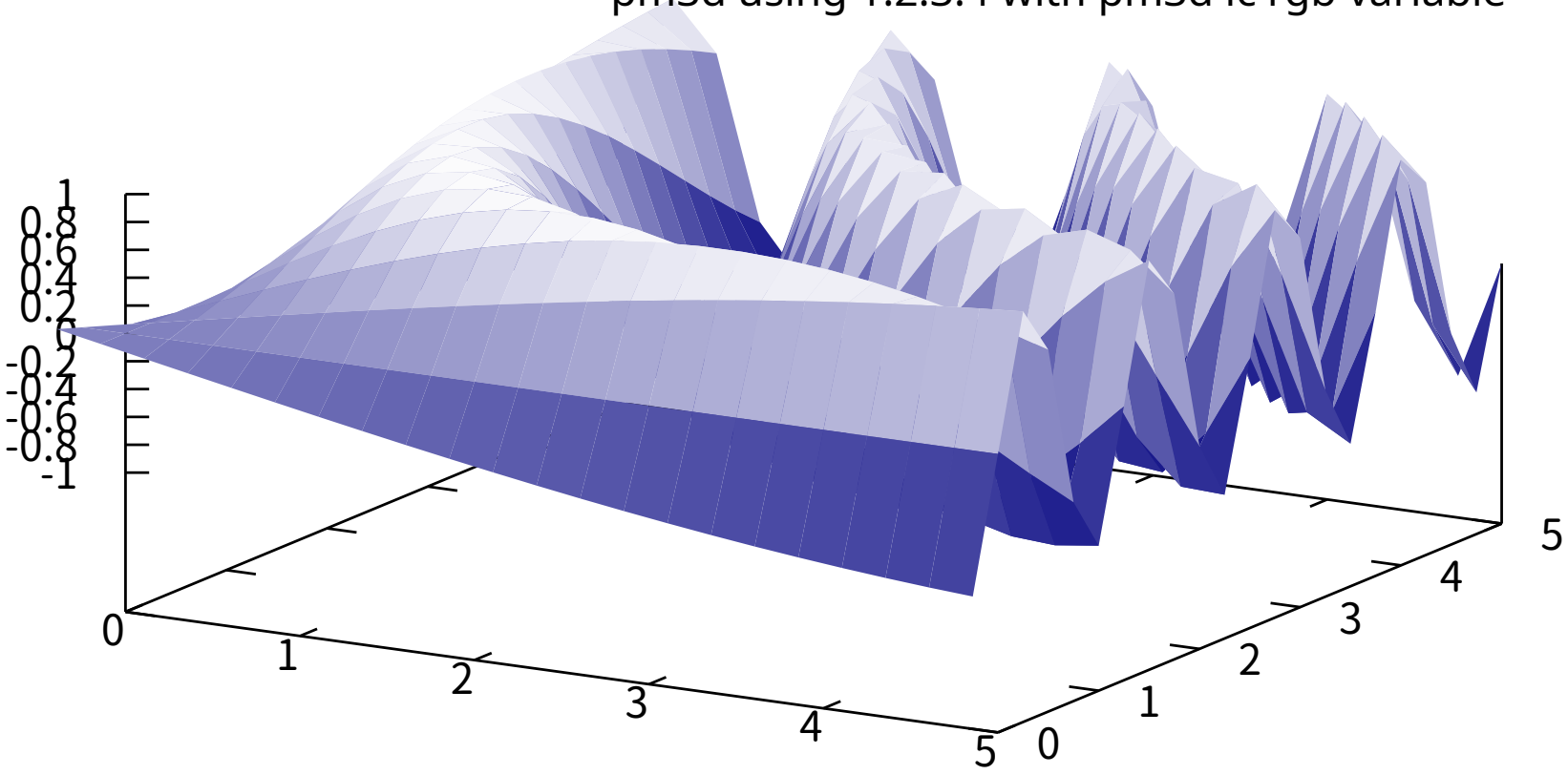
Compare closure of perimeter





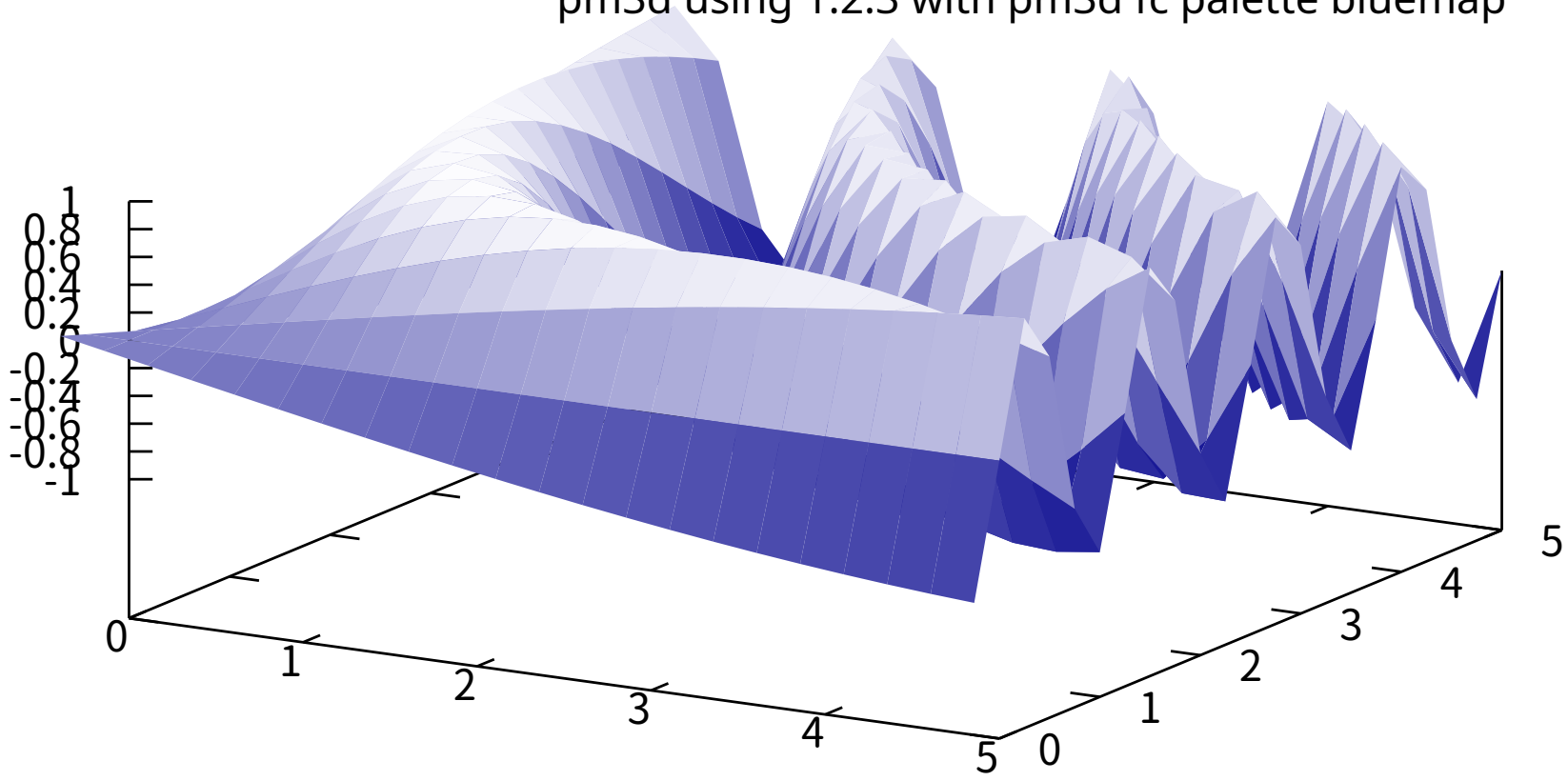
Use hand-constructed 'blues' palette via rgb variable

pm3d using 1:2:3:4 with pm3d lc rgb variable



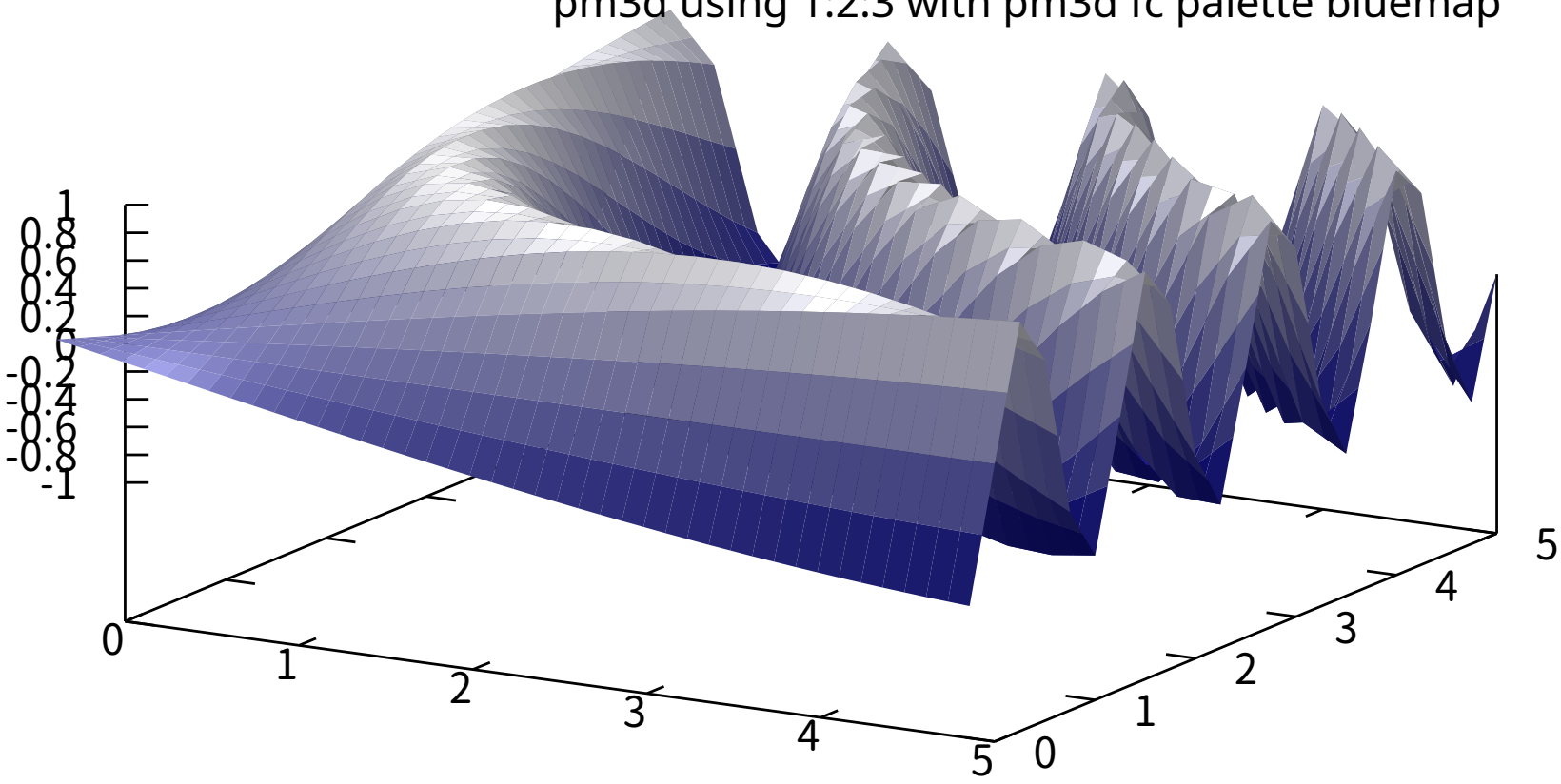
Version 6 offers a new keyword to access this palette

pm3d using 1:2:3 with pm3d fc palette bluemap

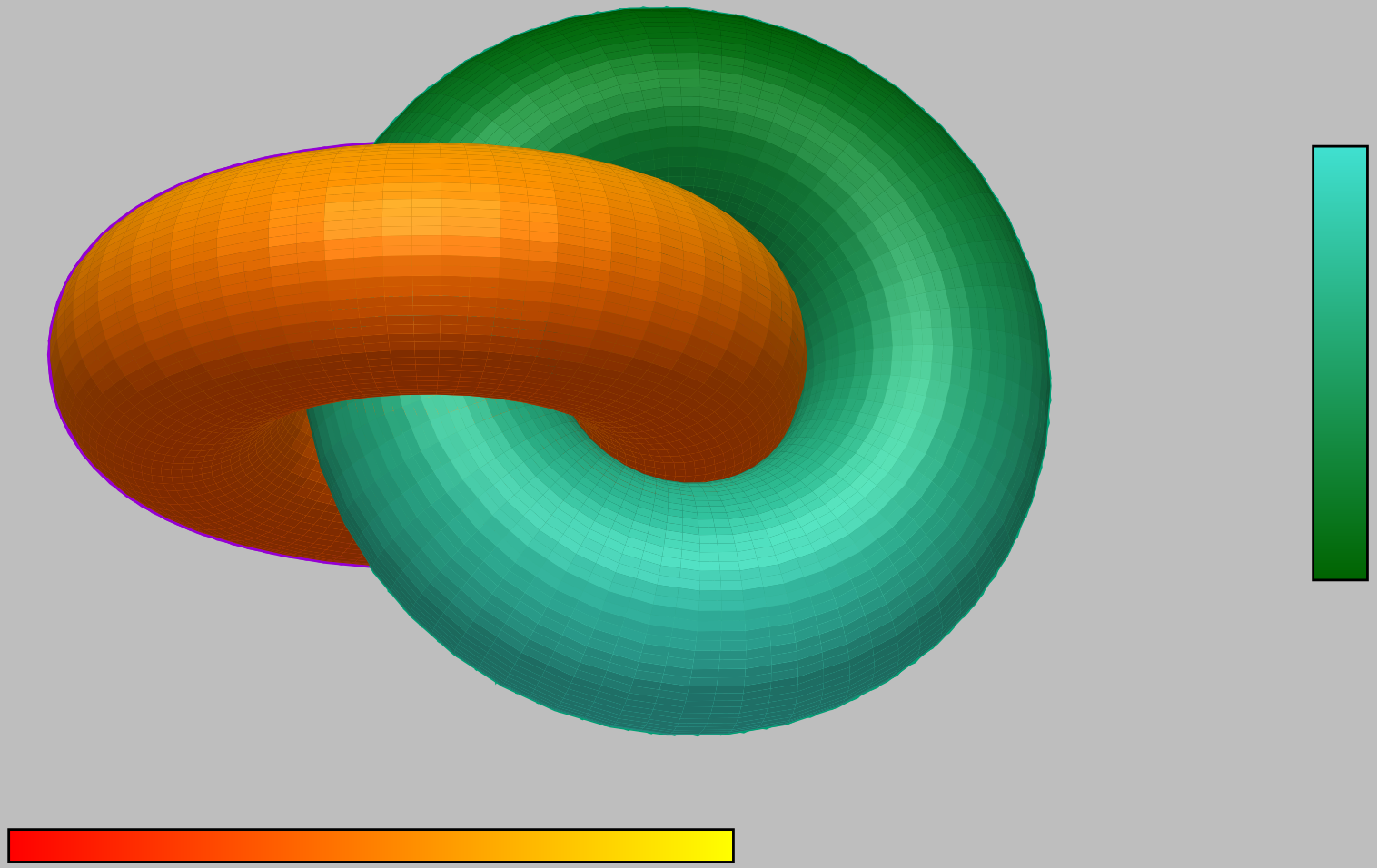


Named colormaps allow pm3d interpolation and lighting

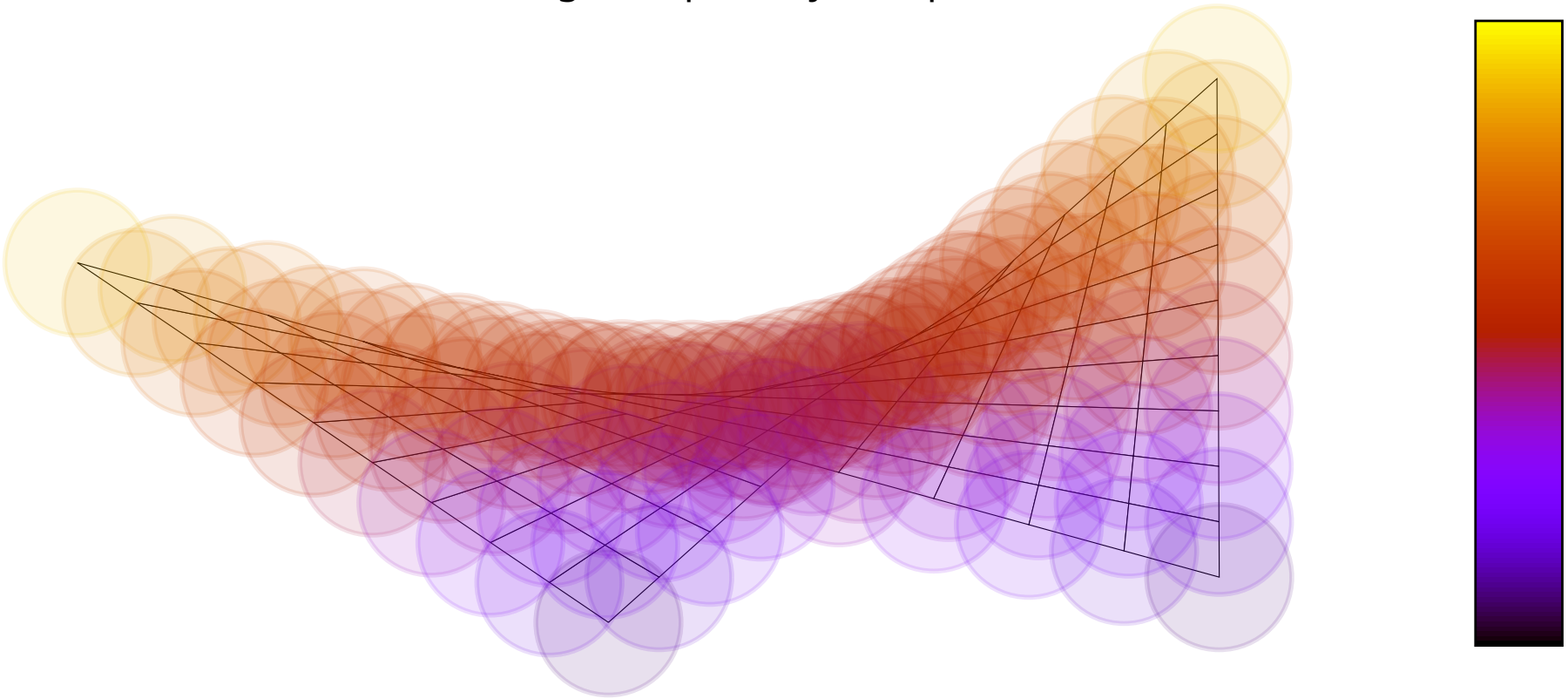
pm3d using 1:2:3 with pm3d fc palette bluemap



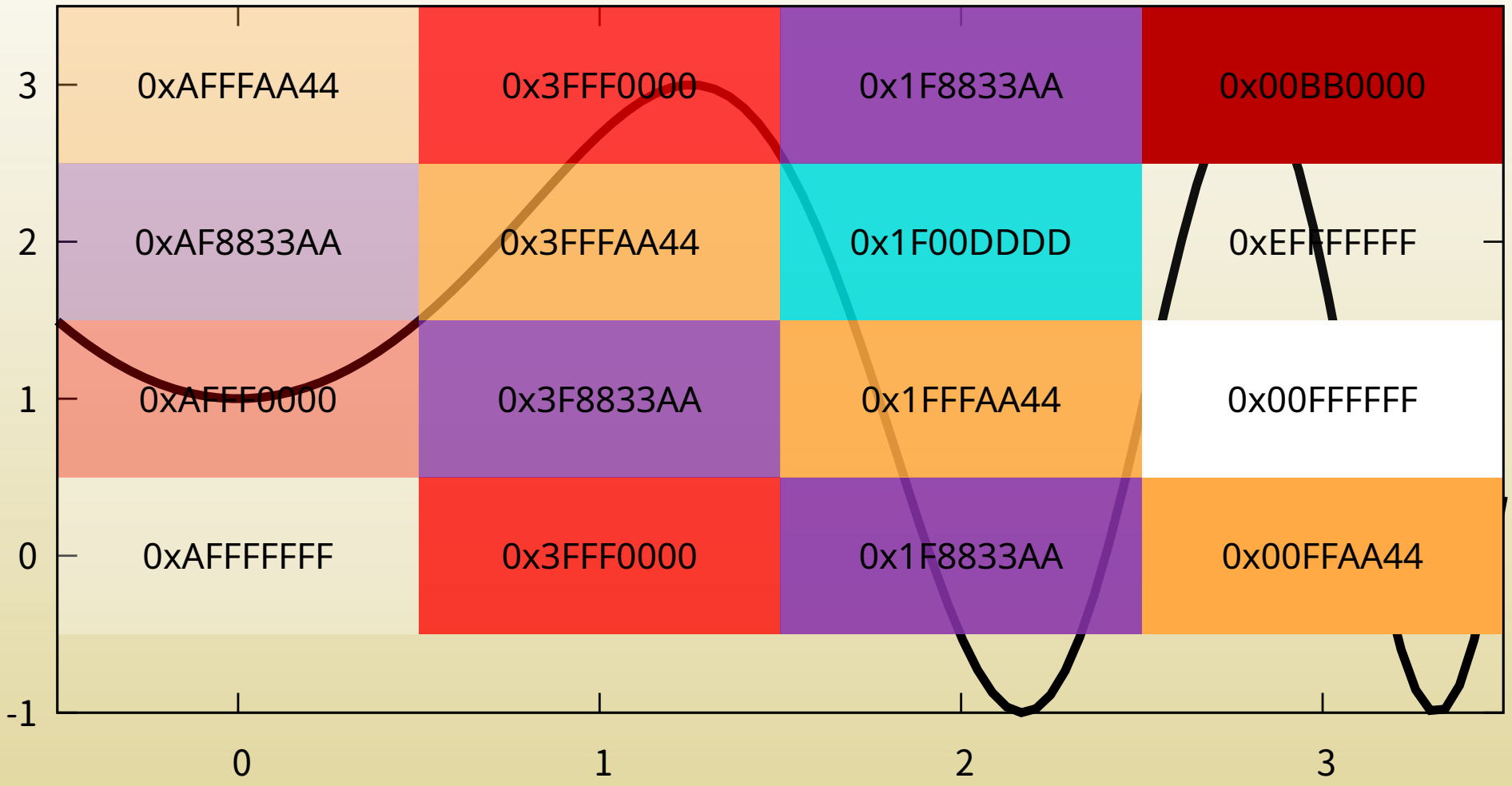
pm3d coloring using two named colormap palettes



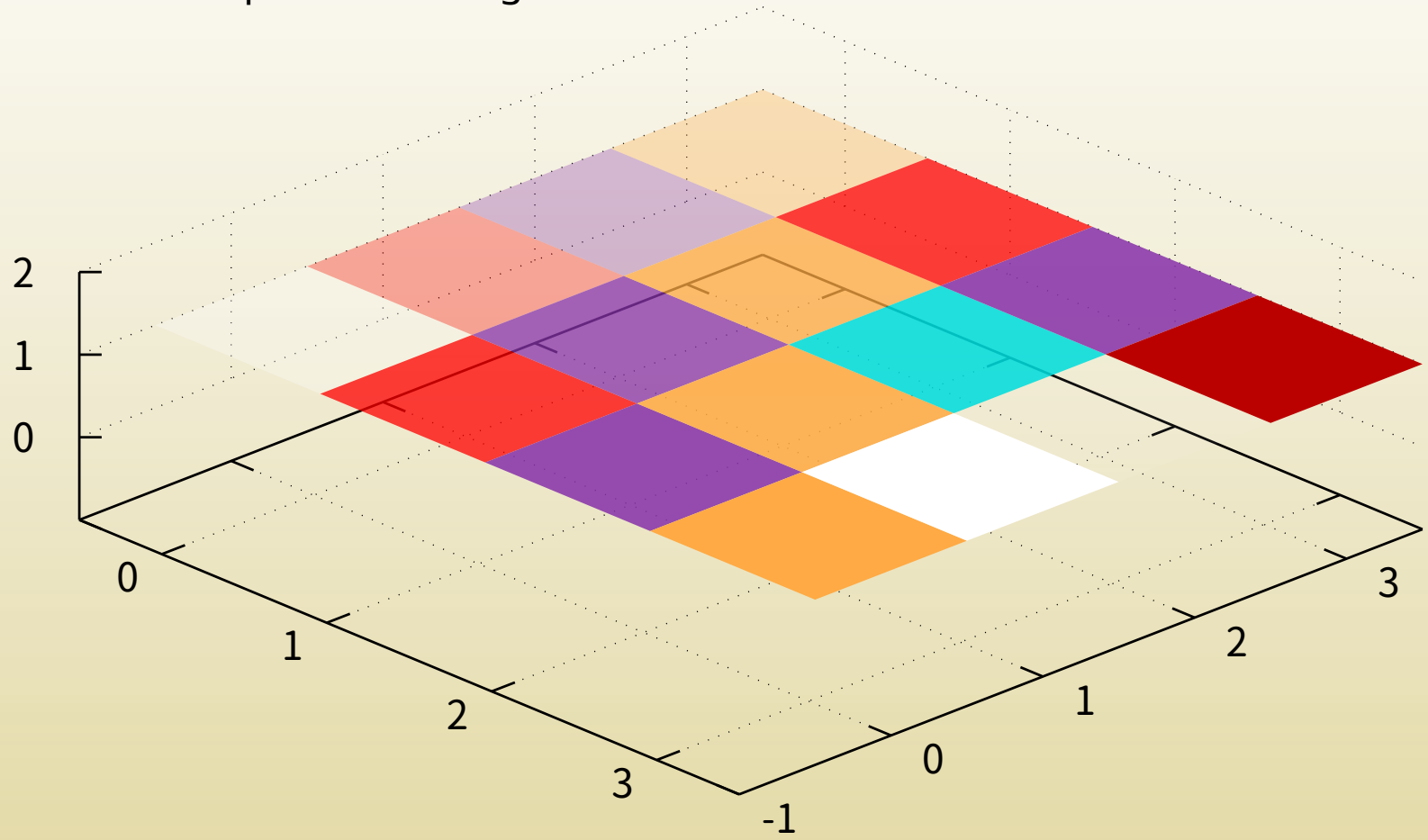
combining transparency with palette colors

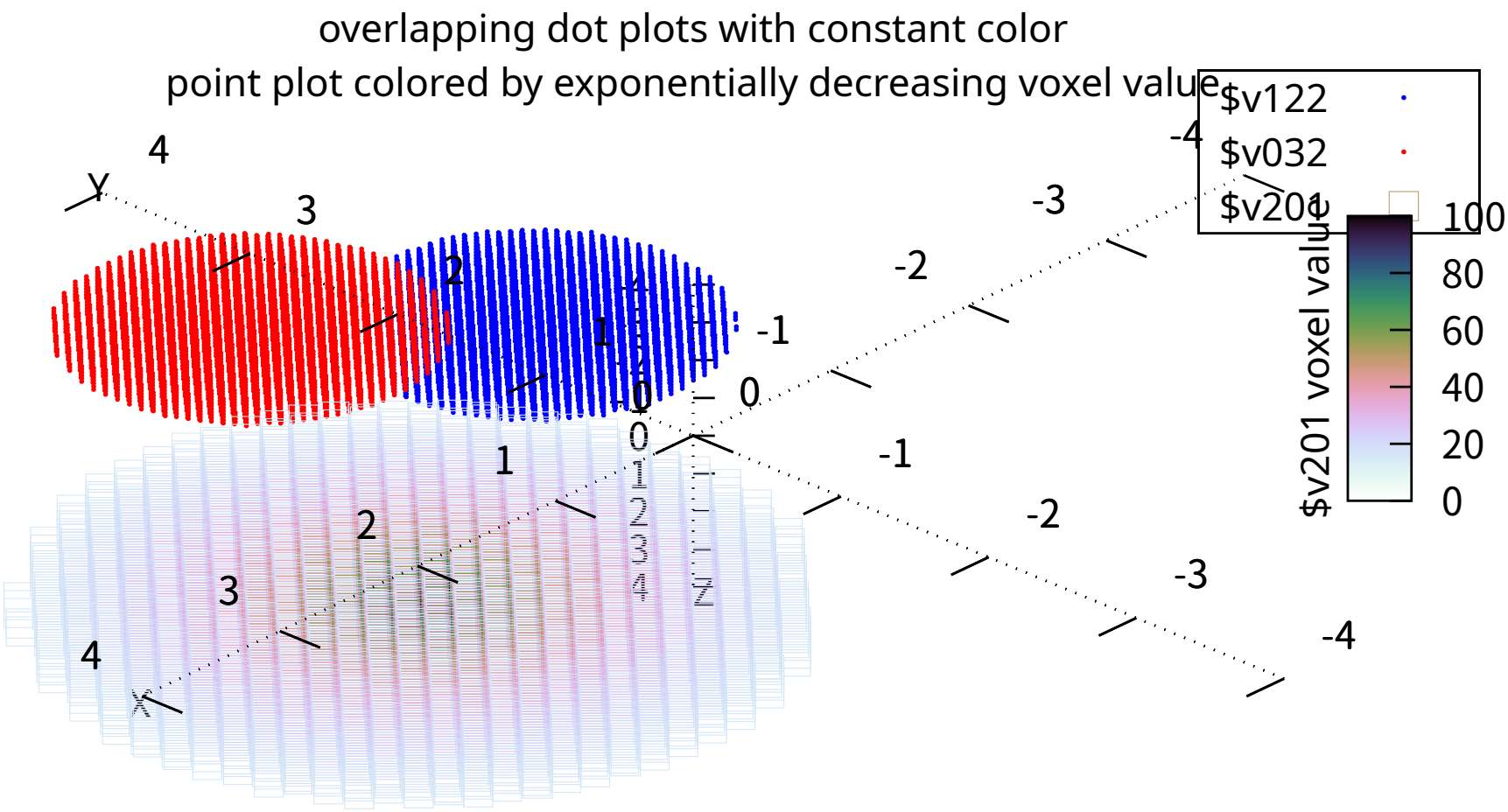


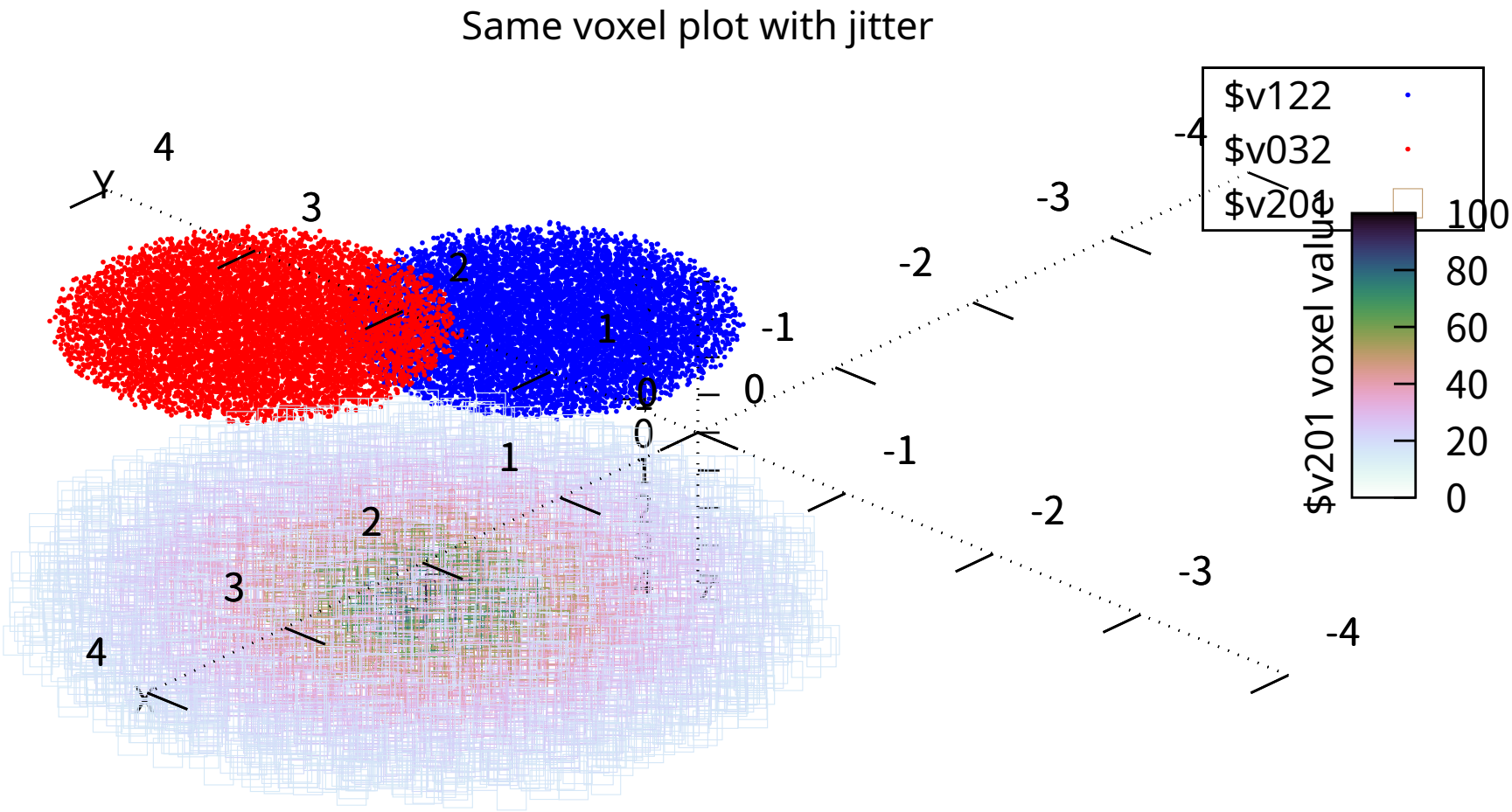
Alpha+RGB image data loaded from a hexadecimal matrix

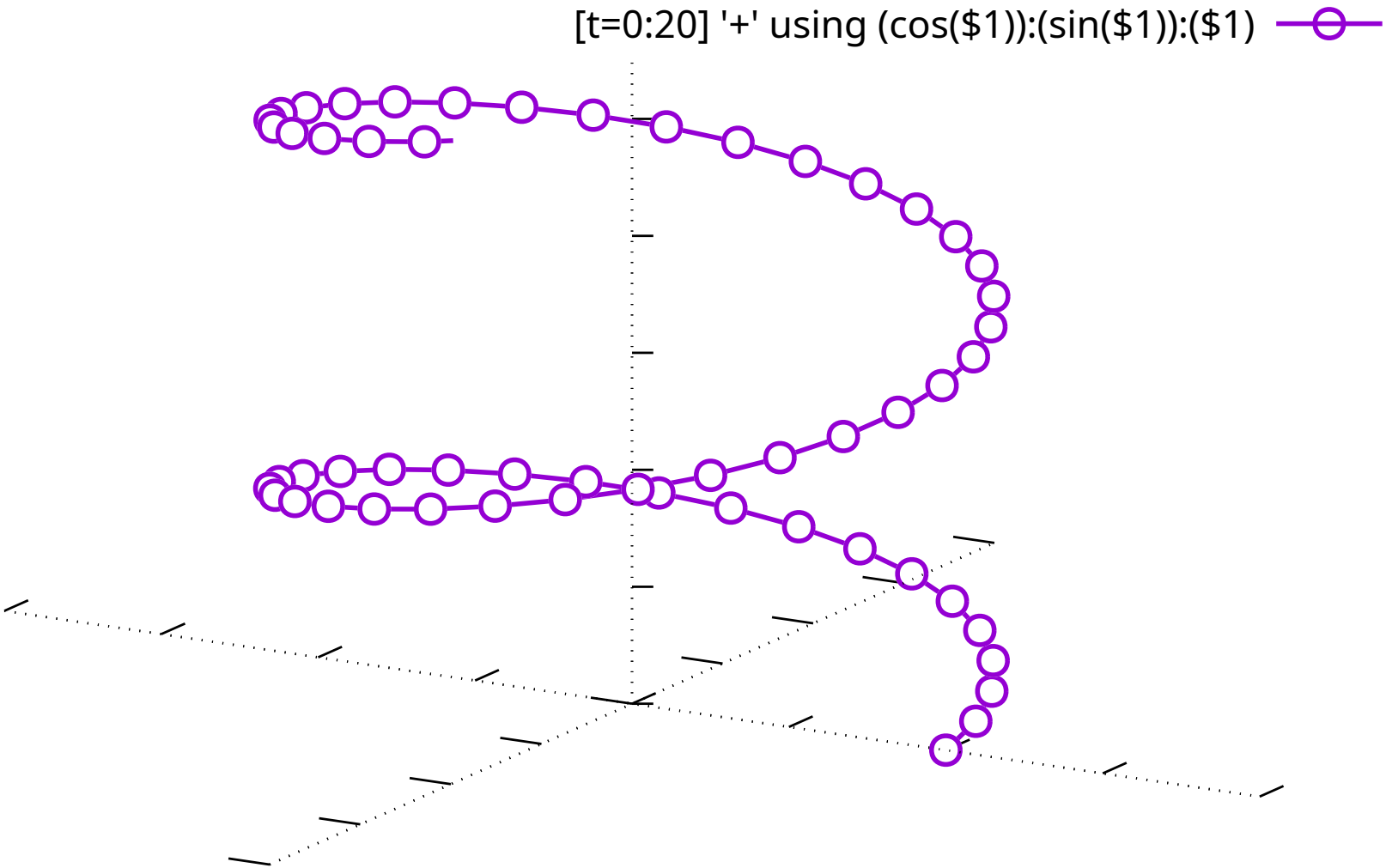


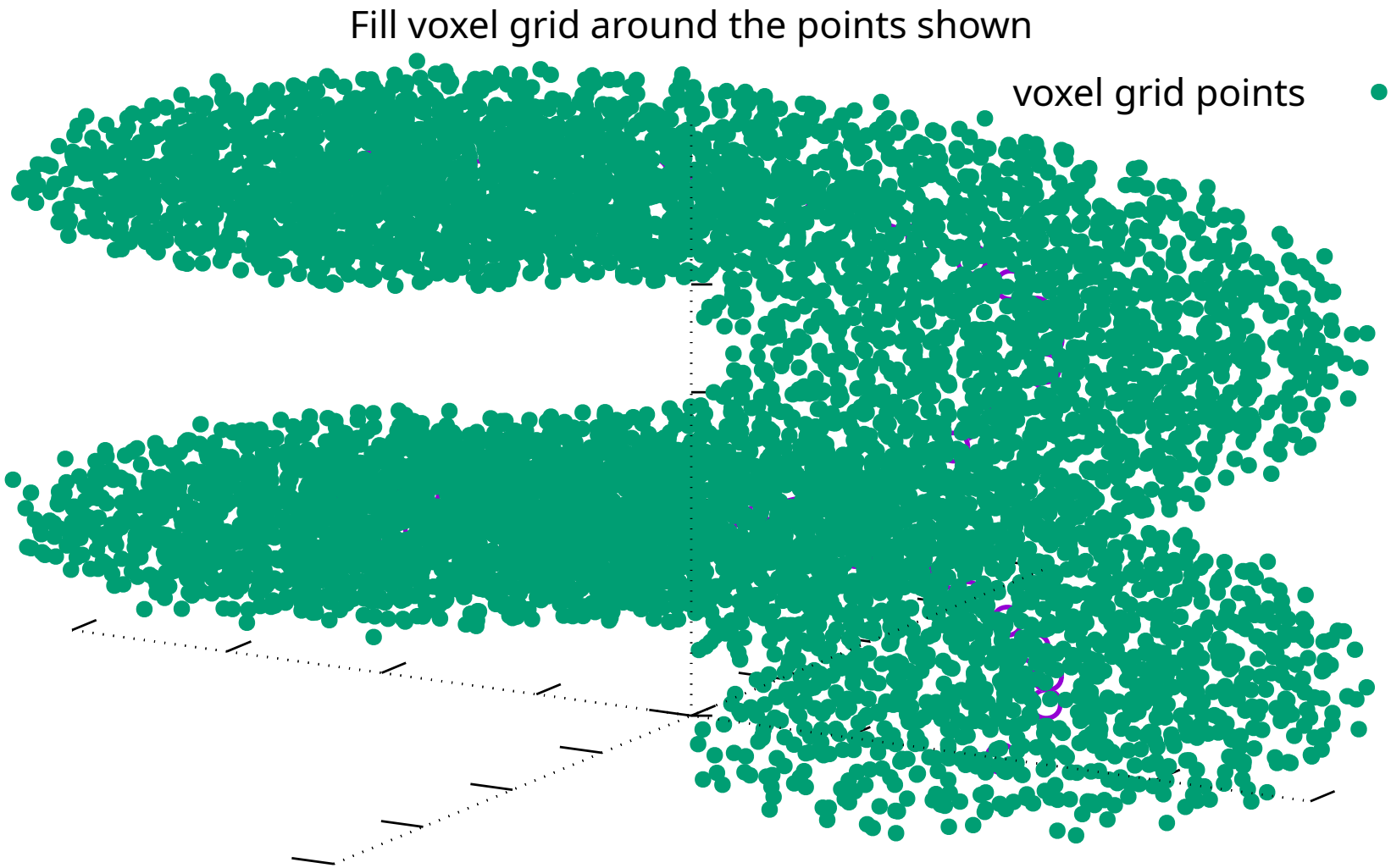
Alpha+RGB image data loaded from a hexadecimal matrix



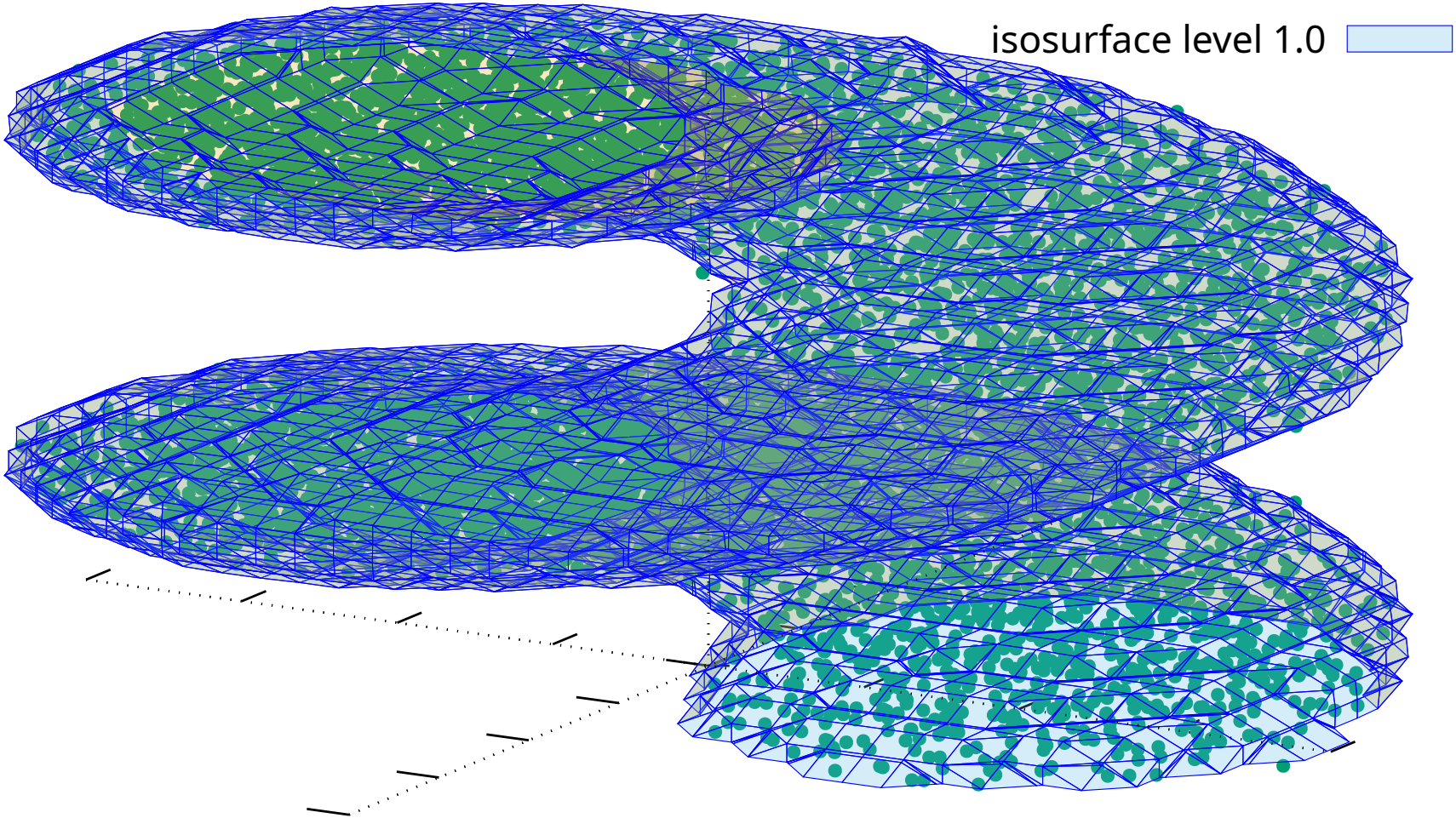

overlapping dot plots with constant color
point plot colored by exponentially decreasing voxel value
$v122
$v032
$v204
$v201 voxel value
100
80
60
40
20
0
Y
X
Z
4
3
2
1
0
-1
-2
-3
-4

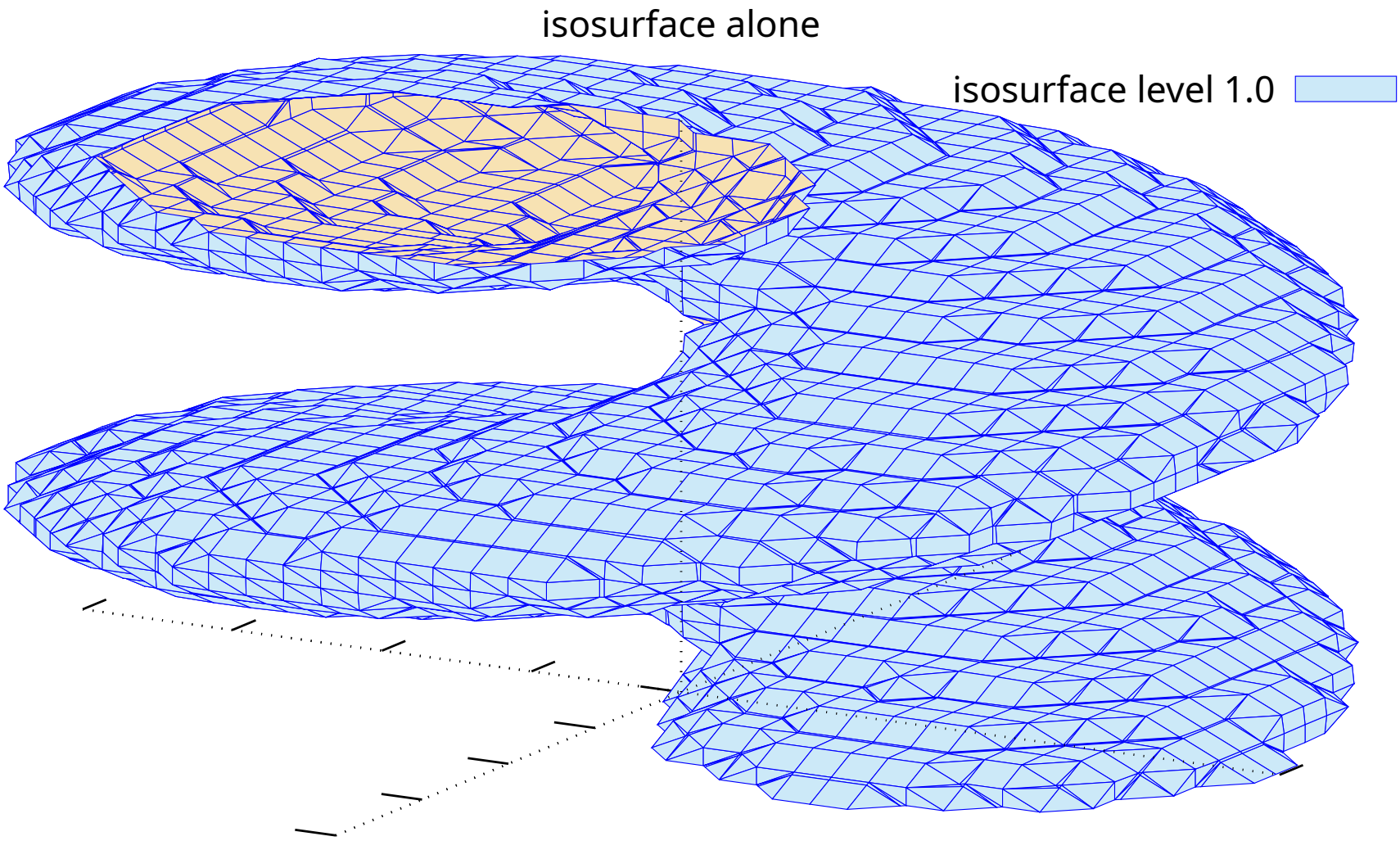

Same voxel plot with jitter
$v122
$v032
$v201
$v201 voxel value
100
80
60
40
20
0
Y
4
3
2
1
0
-1
-2
-3
-4

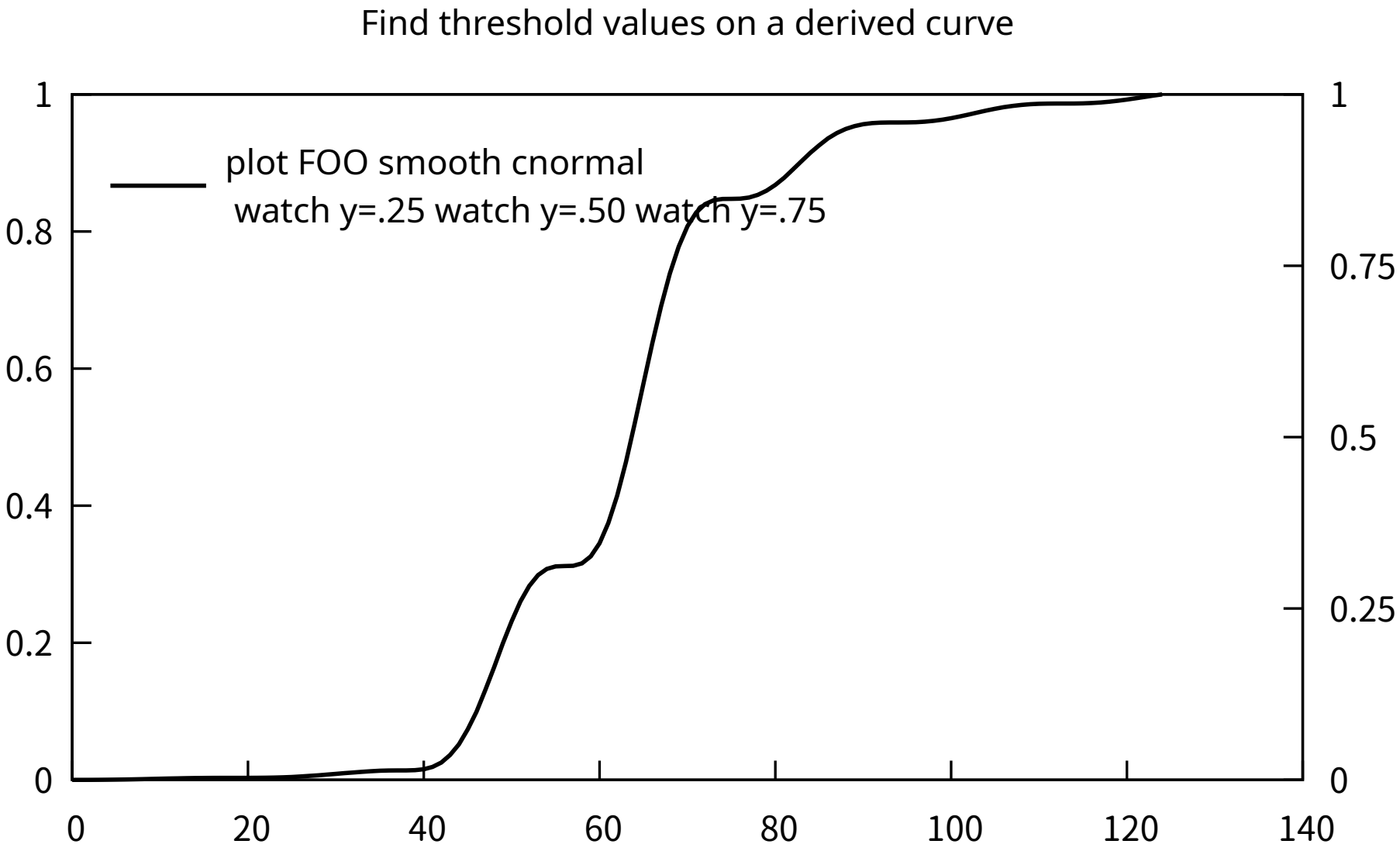

[t=0:20] '+' using (cos($1)):(sin($1)):($1)

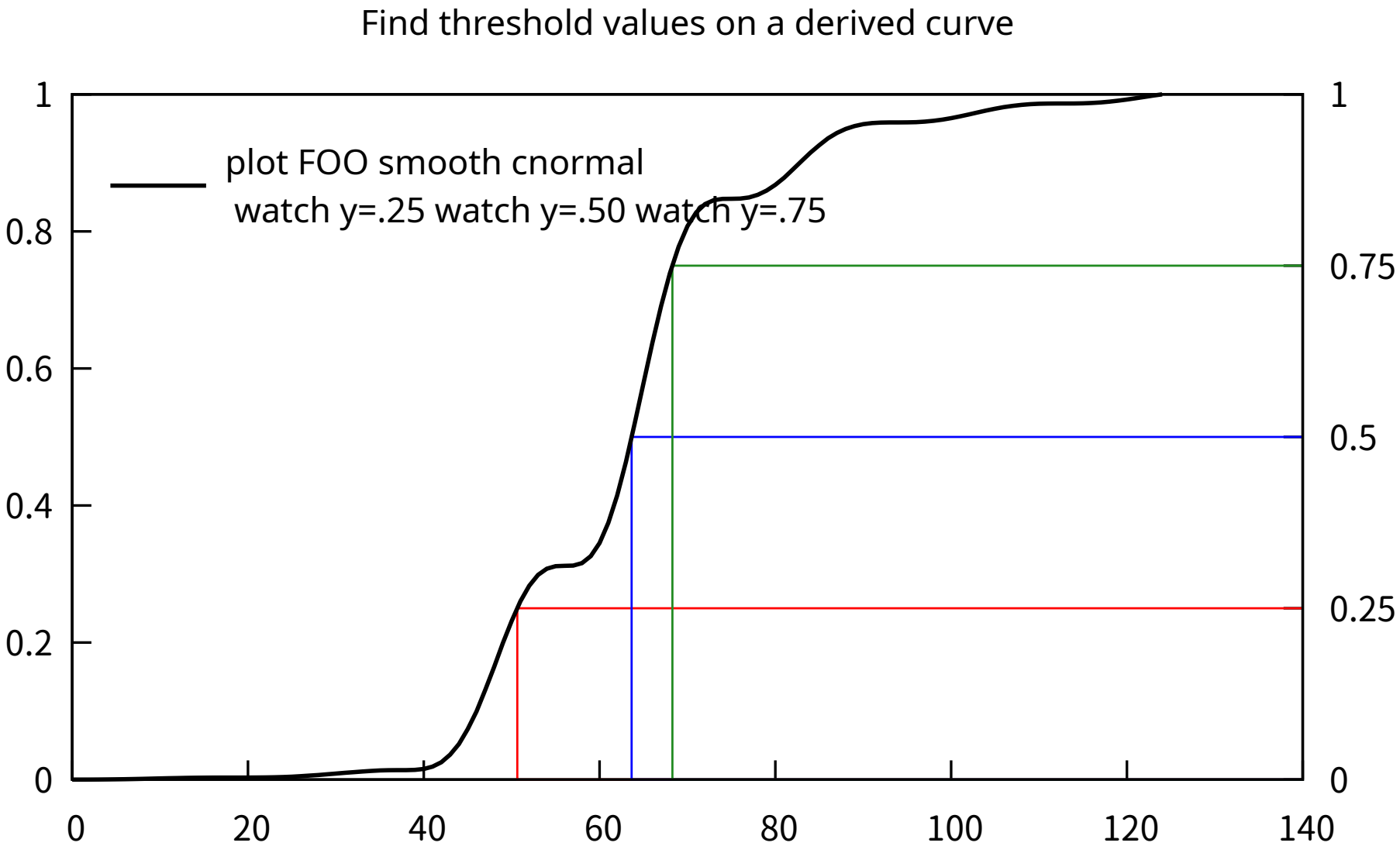

Fill voxel grid around the points shown
voxel grid points

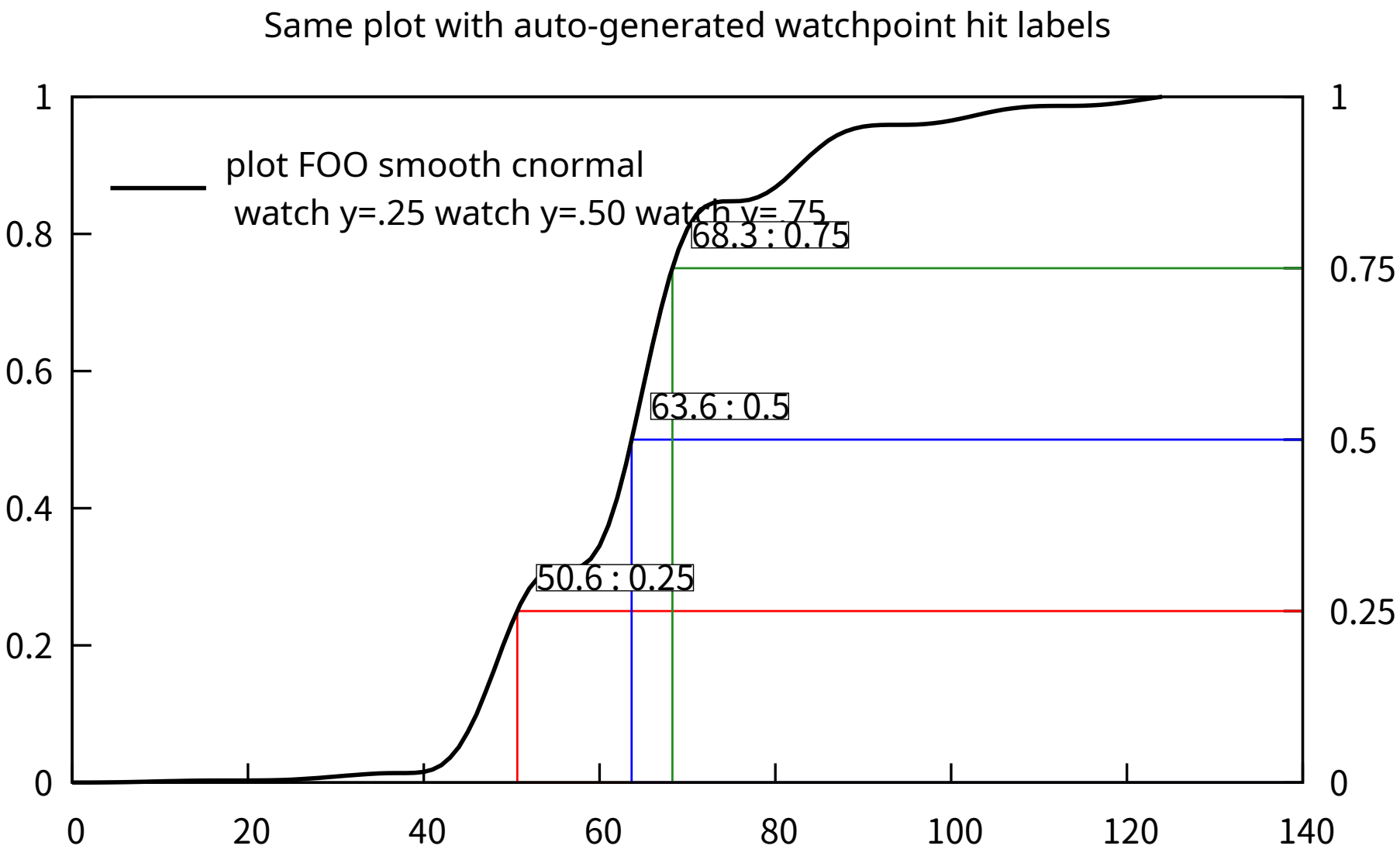
Draw isosurface enclosing all points

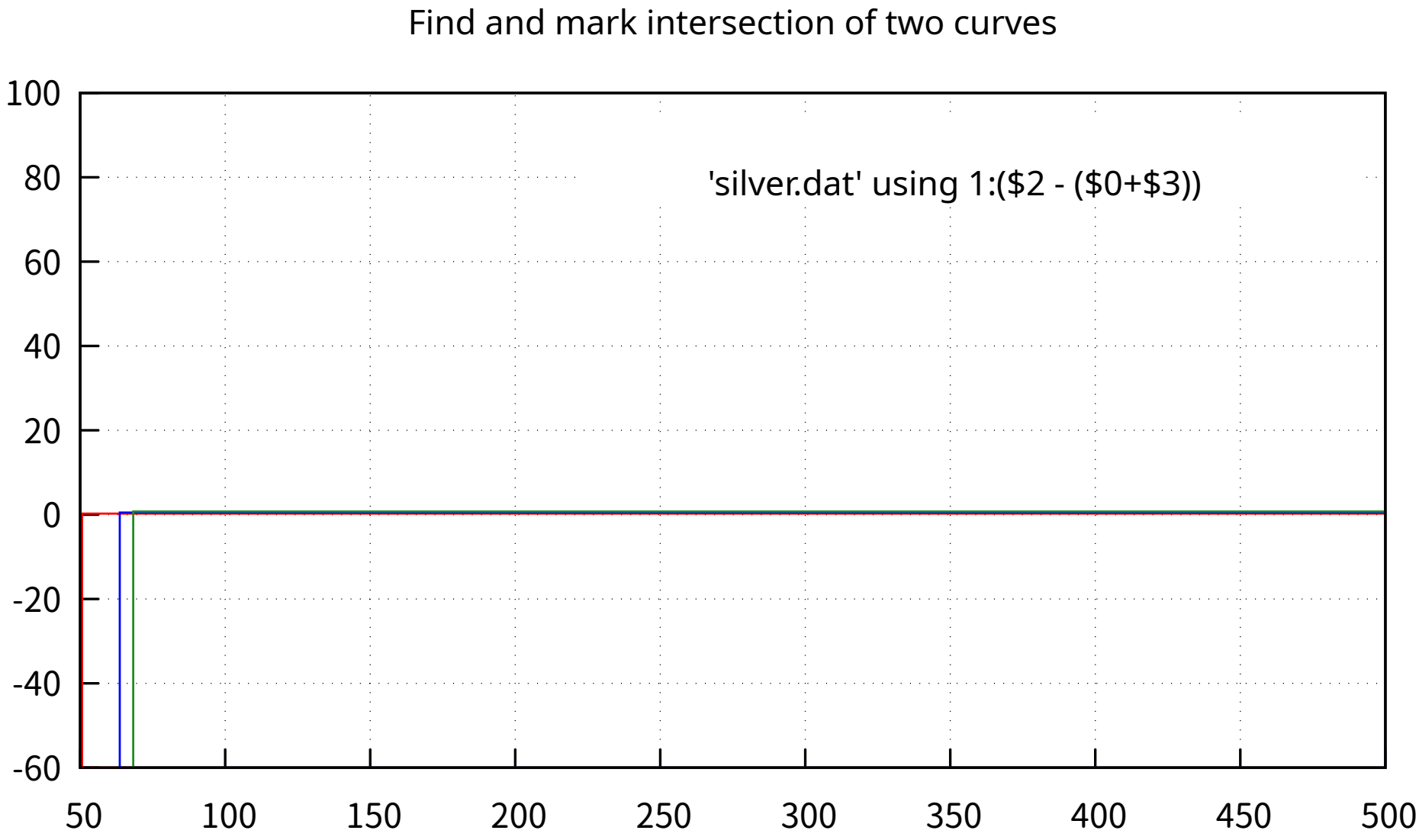


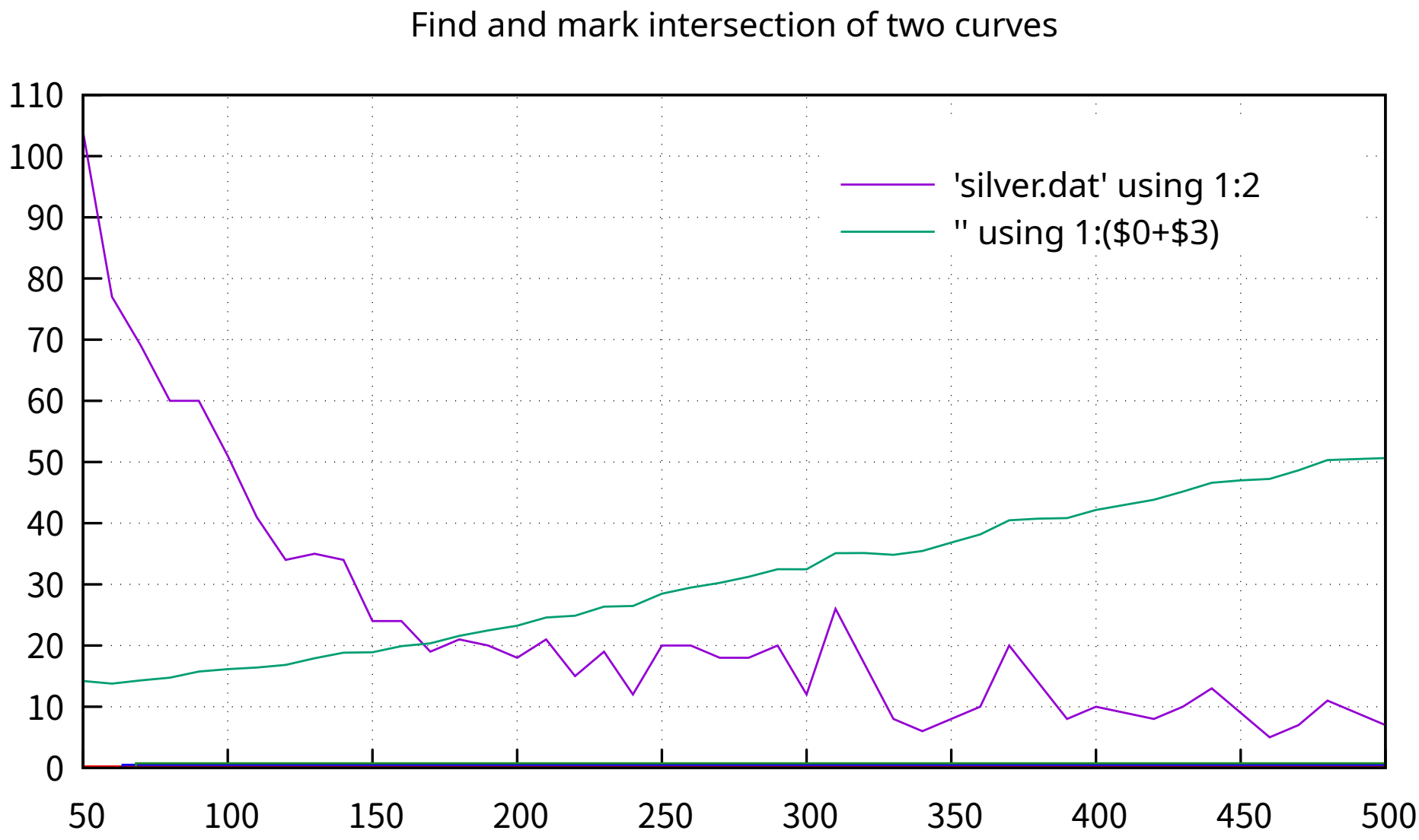

isosurface alone
isosurface level 1.0

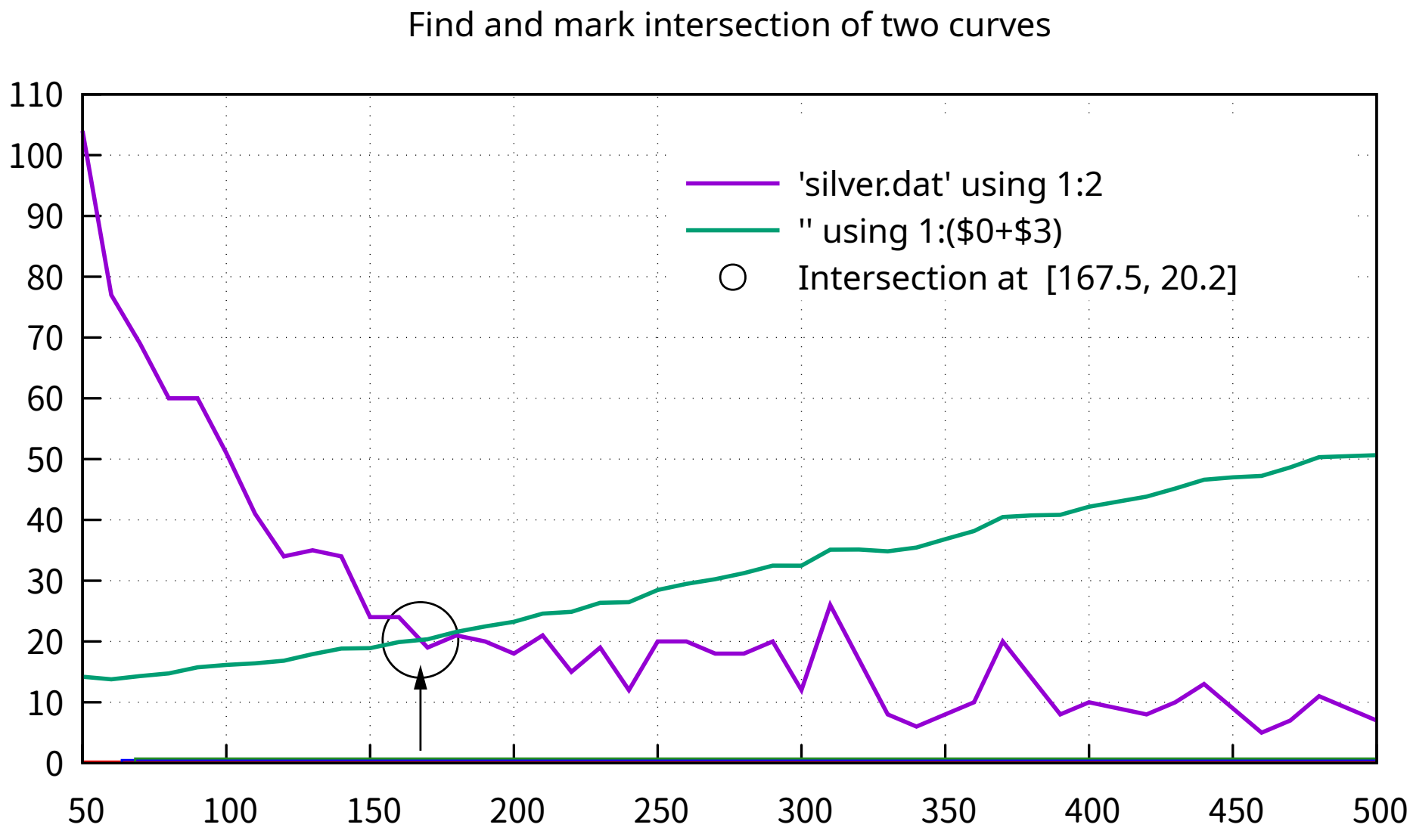

Find threshold values on a derived curve
plot FOO smooth cnormal
watch y=.25 watch y=.50 watch y=.75
1
0.8
0.6
0.4
0.2
0
1
0.75
0.5
0.25
0
0
20
40
60
80
100
120
140


Find threshold values on a derived curve
plot FOO smooth cnormal
watch y=.25 watch y=.50 watch y=.75
1
0.8
0.6
0.4
0.2
0
1
0.75
0.5
0.25
0
0
20
40
60
80
100
120
140

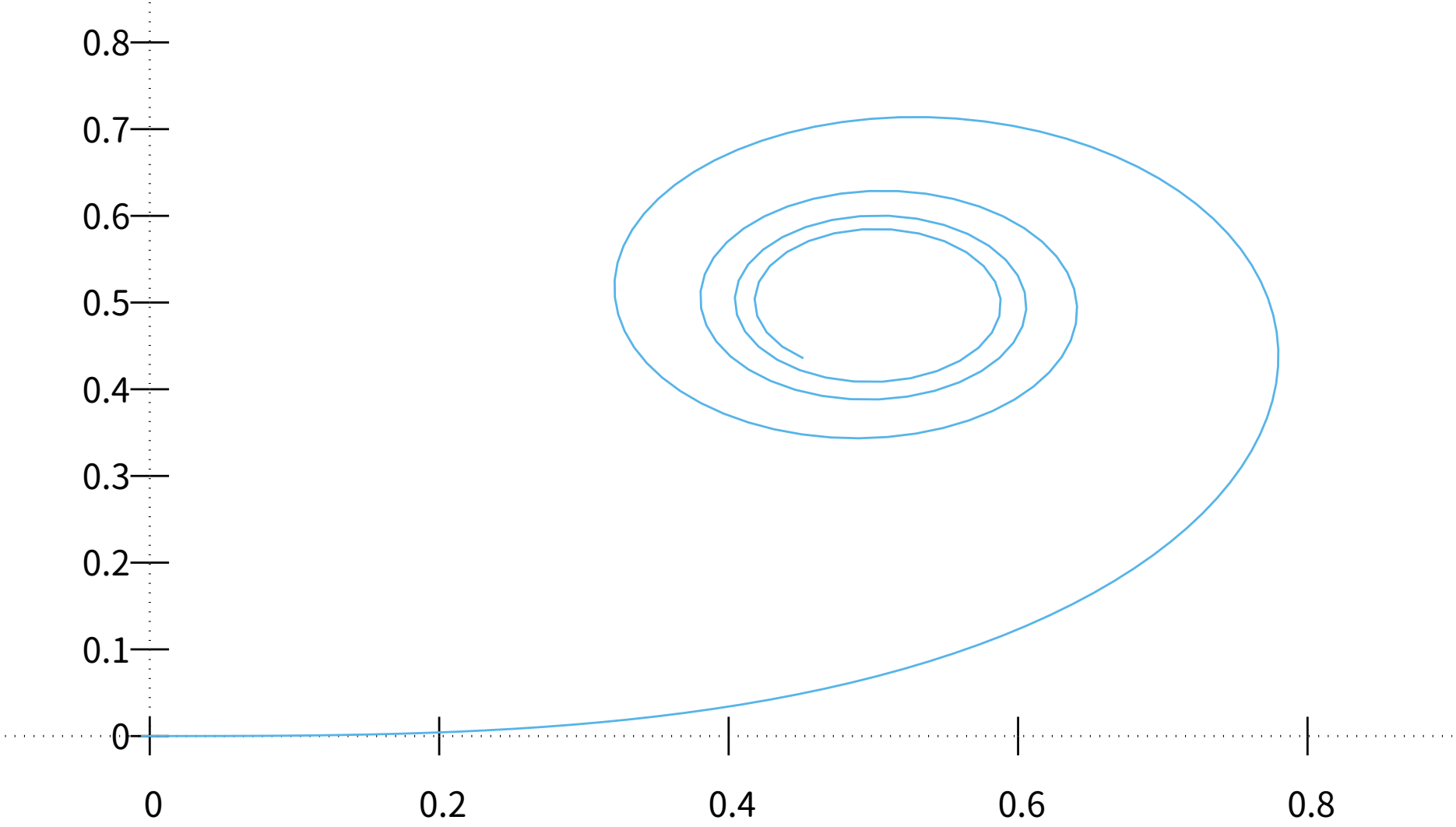

Same plot with auto-generated watchpoint hit labels
plot FOO smooth cnormal
watch y=.25 watch y=.50 watch y=.75
68.3 : 0.75
63.6 : 0.5
50.6 : 0.25
1
0.8
0.6
0.4
0.2
0
1
0.75
0.5
0.25
0
0
20
40
60
80
100
120
140

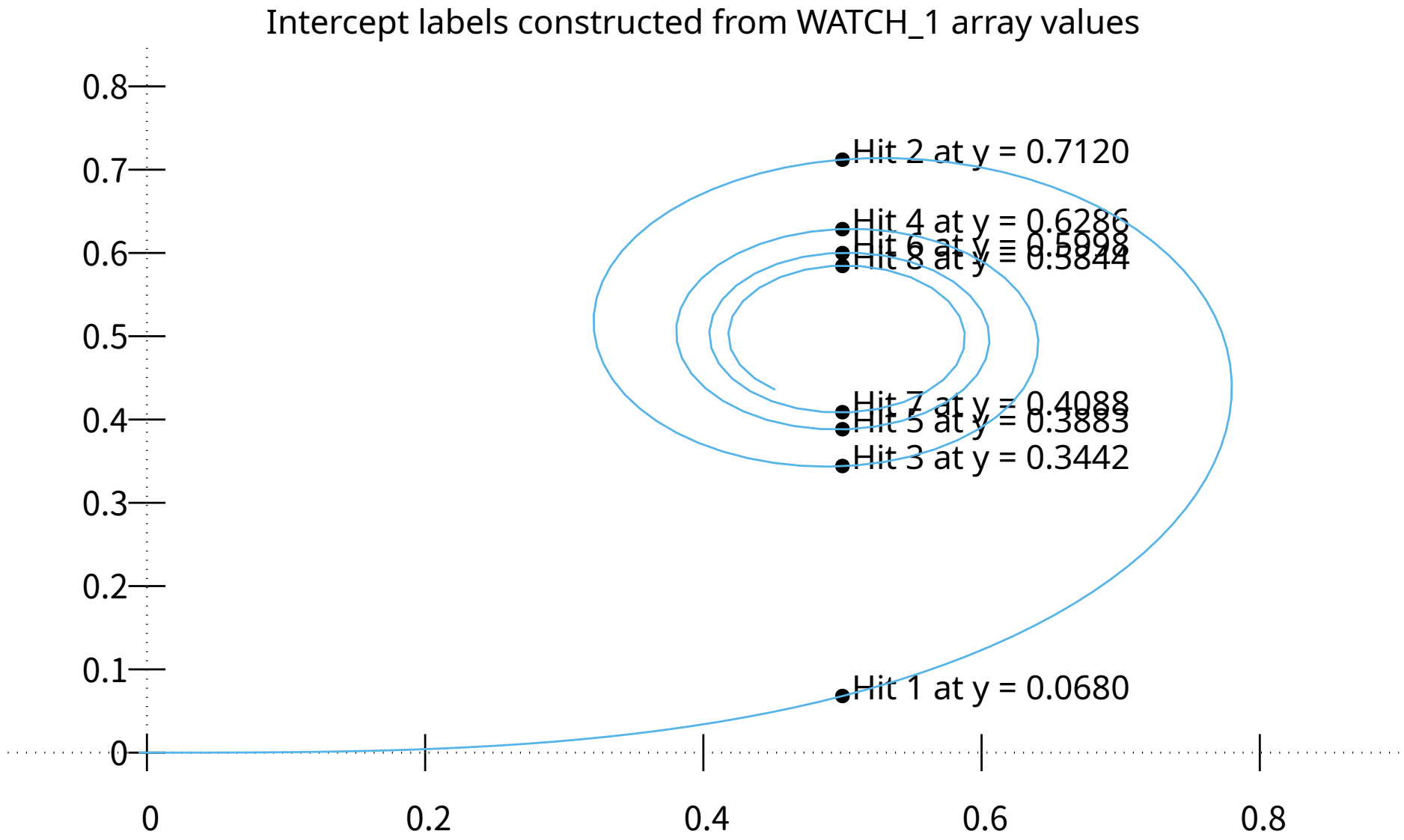

Find and mark intersection of two curves
'silver.dat' using 1:($2 - ($0+$3))
100
80
60
40
20
0
-20
-40
-60
50
100
150
200
250
300
350
400
450
500


Find and mark intersection of two curves
'silver.dat' using 1:2
'' using 1:($0+$3)
110
100
90
80
70
60
50
40
30
20
10
0
50
100
150
200
250
300
350
400
450
500


Find and mark intersection of two curves
'silver.dat' using 1:2
'' using 1:($0+$3)
Intersection at [167.5, 20.2]
110
100
90
80
70
60
50
40
30
20
10
0
50
100
150
200
250
300
350
400
450
500

Find y intercepts of a parametric function




Intercept labels constructed from WATCH_1 array values
Hit 2 at y = 0.7120
Hit 4 at y = 0.6286
Hit 6 at y = 0.5998
Hit 8 at y = 0.5844
Hit 7 at y = 0.4088
Hit 5 at y = 0.3883
Hit 3 at y = 0.3442
Hit 1 at y = 0.0680
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8