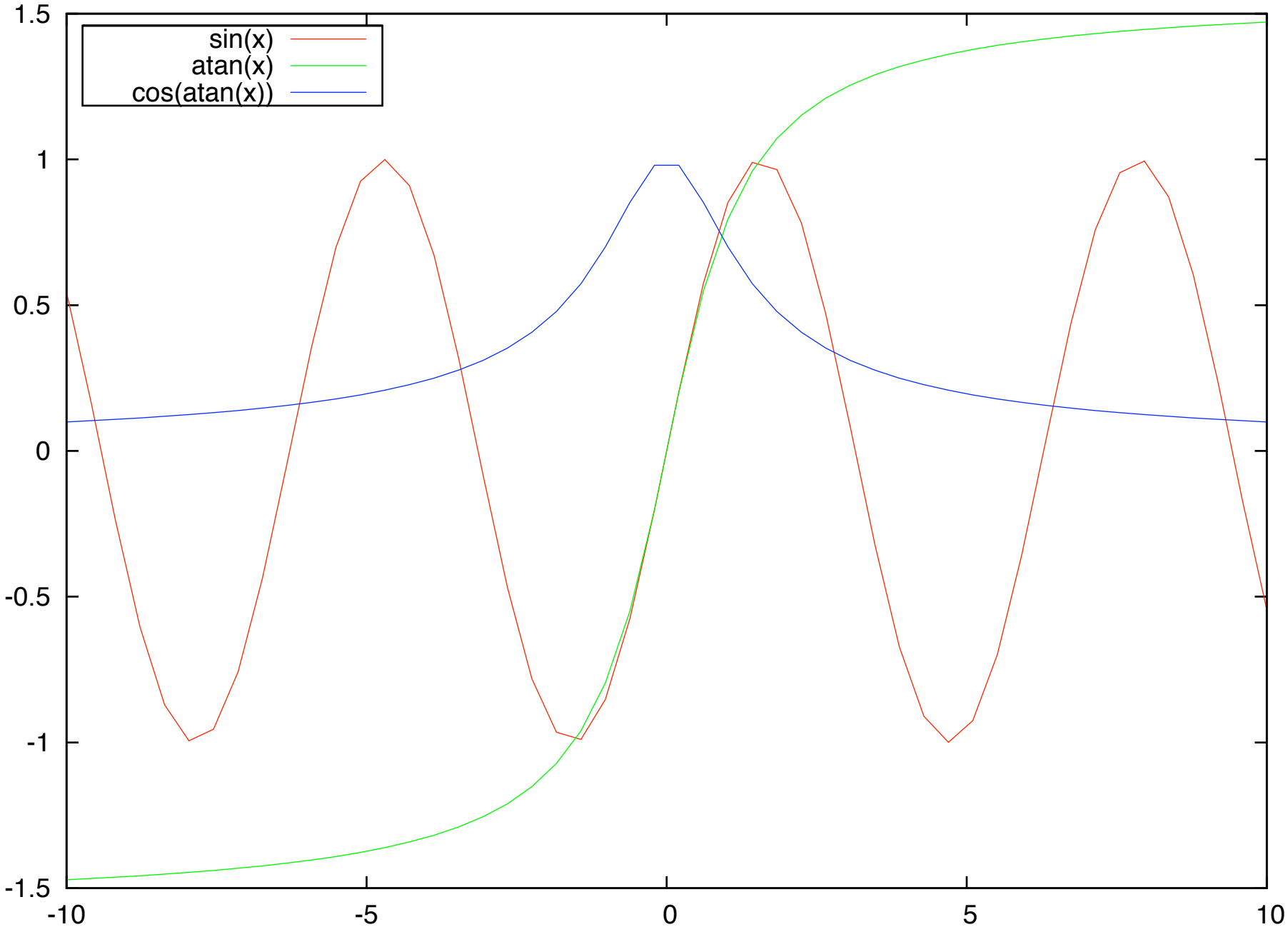
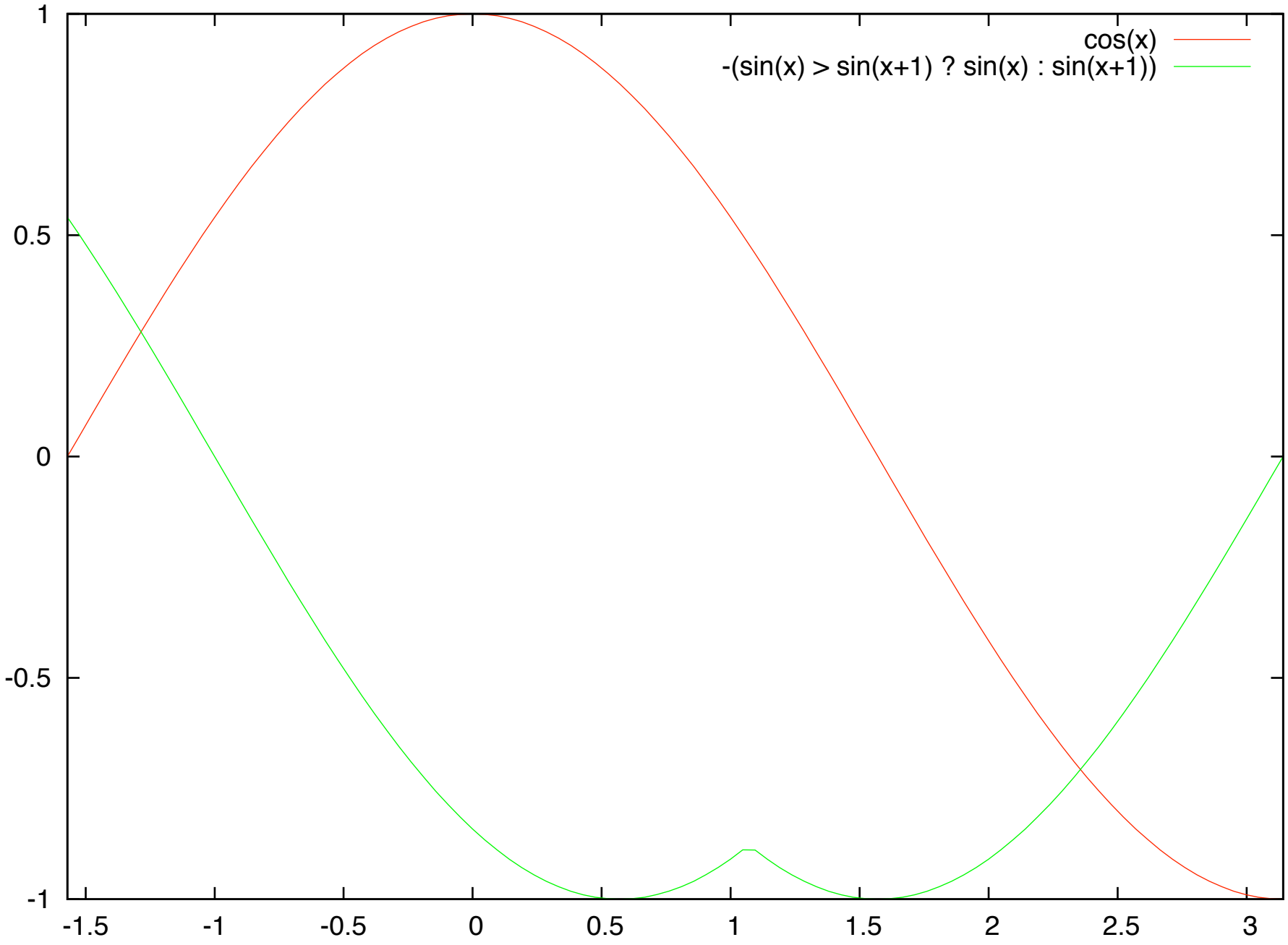
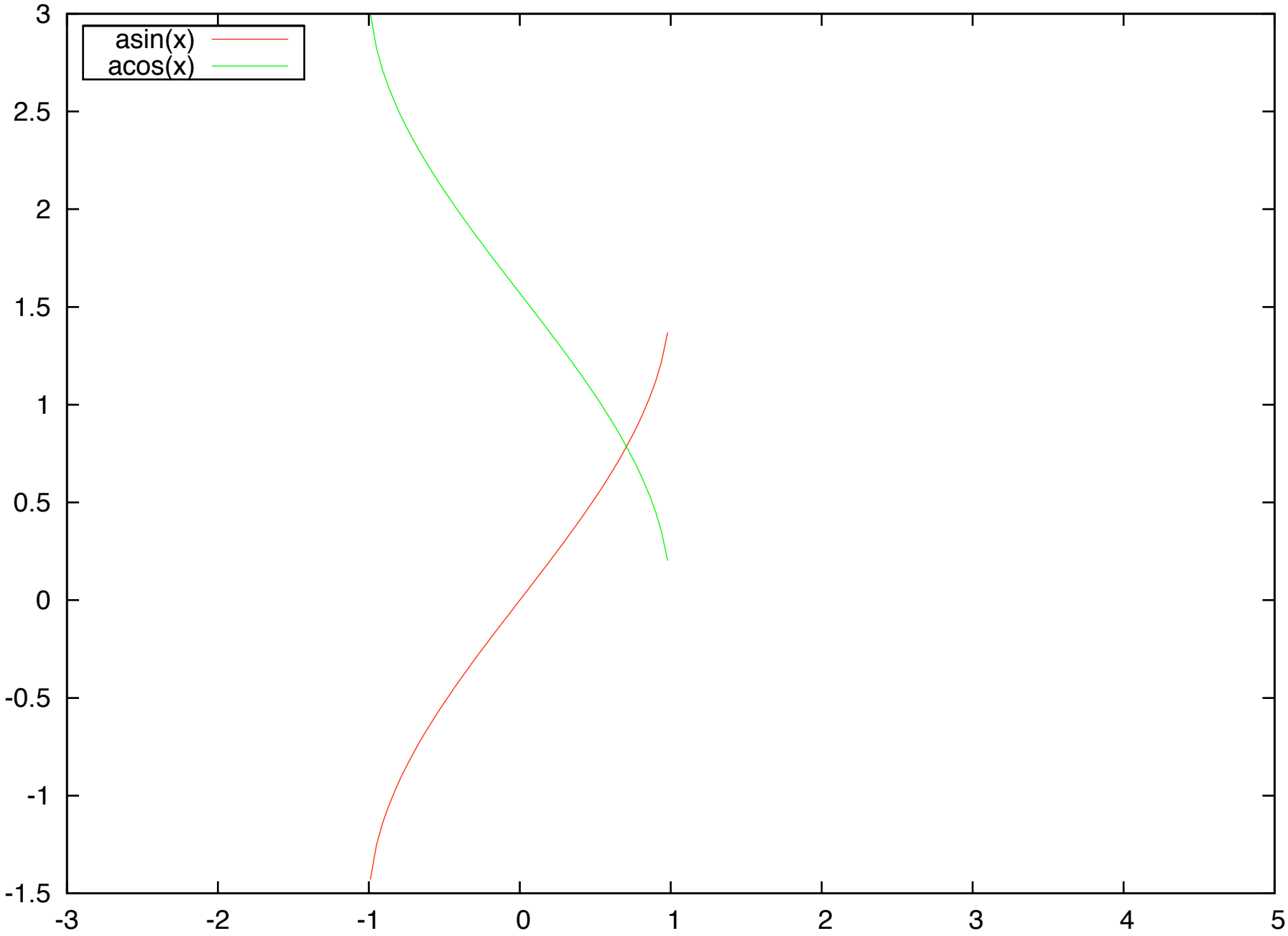
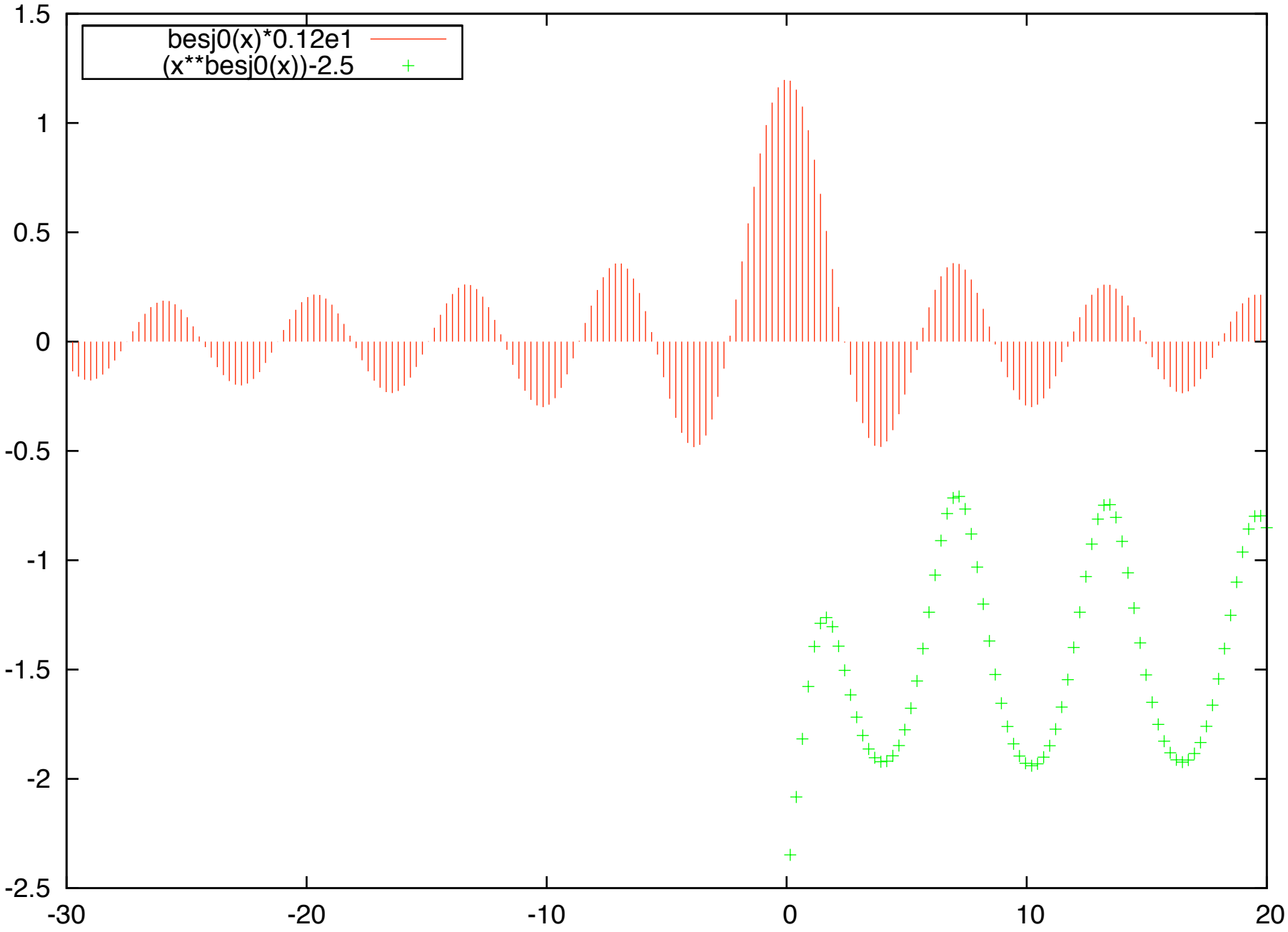
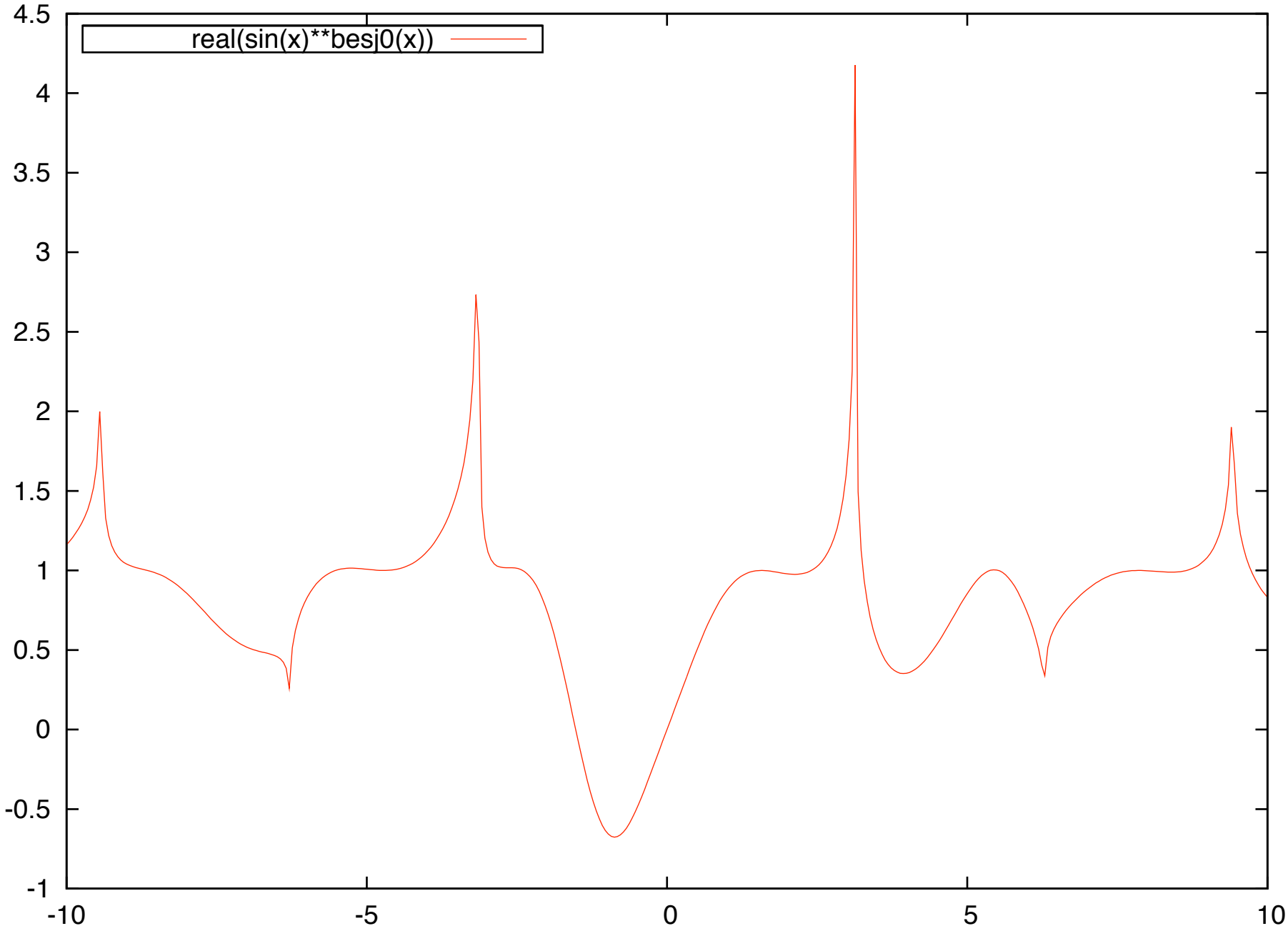
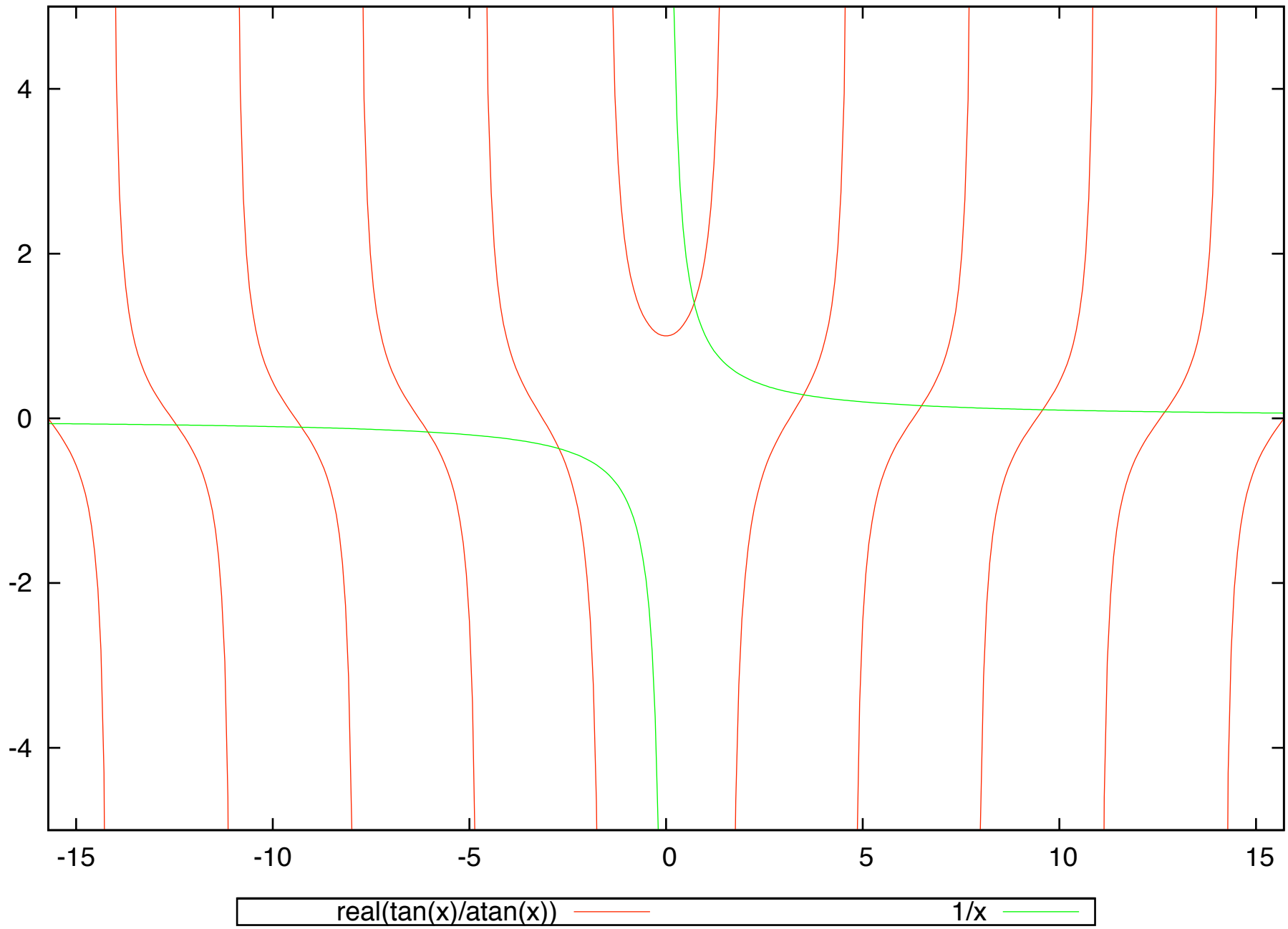

sin(x)
atan(x)
cos(atan(x))
1.5
1
0.5
0
-0.5
-1
-1.5
-10
-5
0
5
10

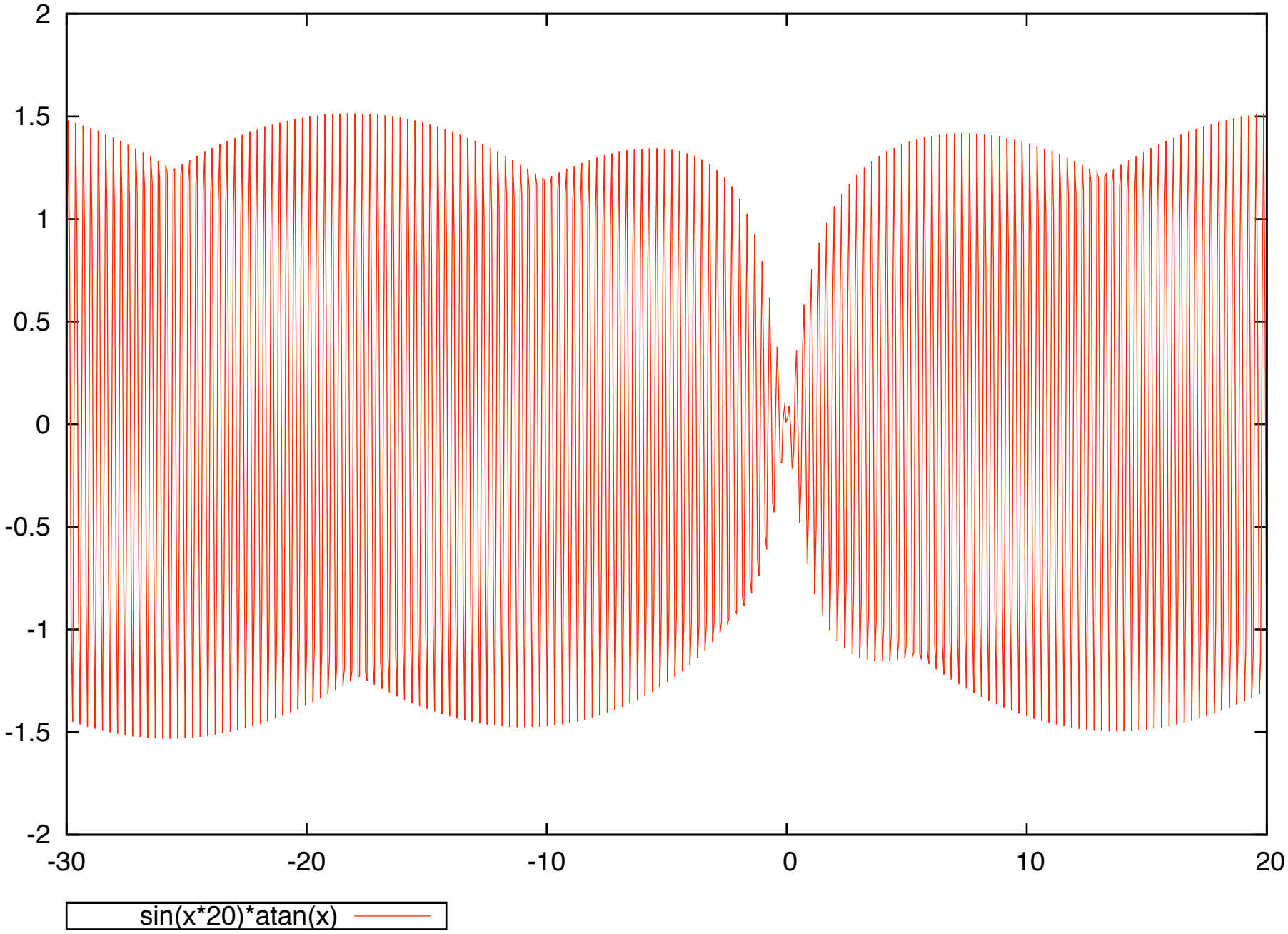

cos(x)
-(sin(x) > sin(x+1) ? sin(x) : sin(x+1))
1
0.5
0
-0.5
-1
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3

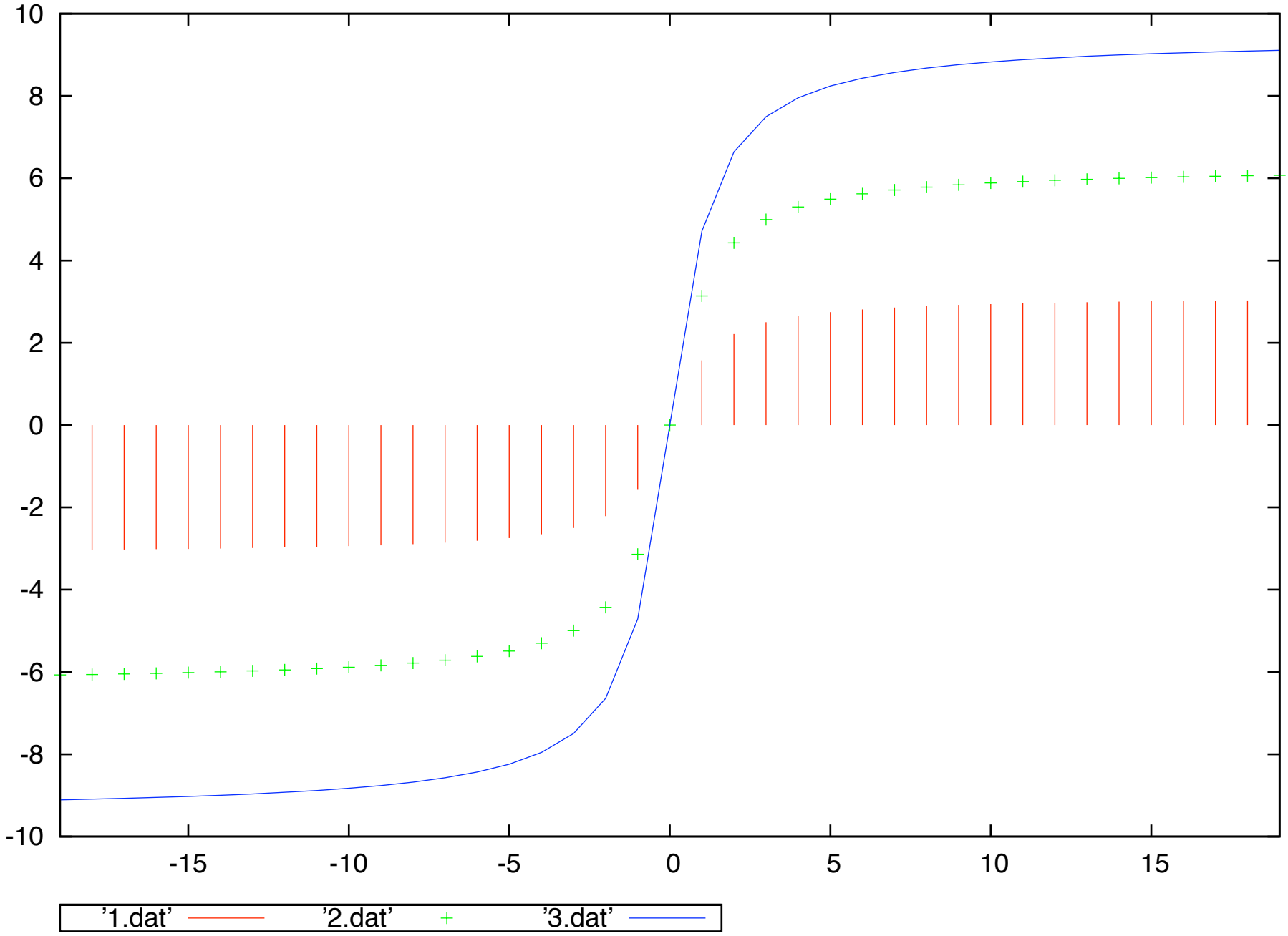

asin(x)
acos(x)
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-3
-2
-1
0
1
2
3
4
5

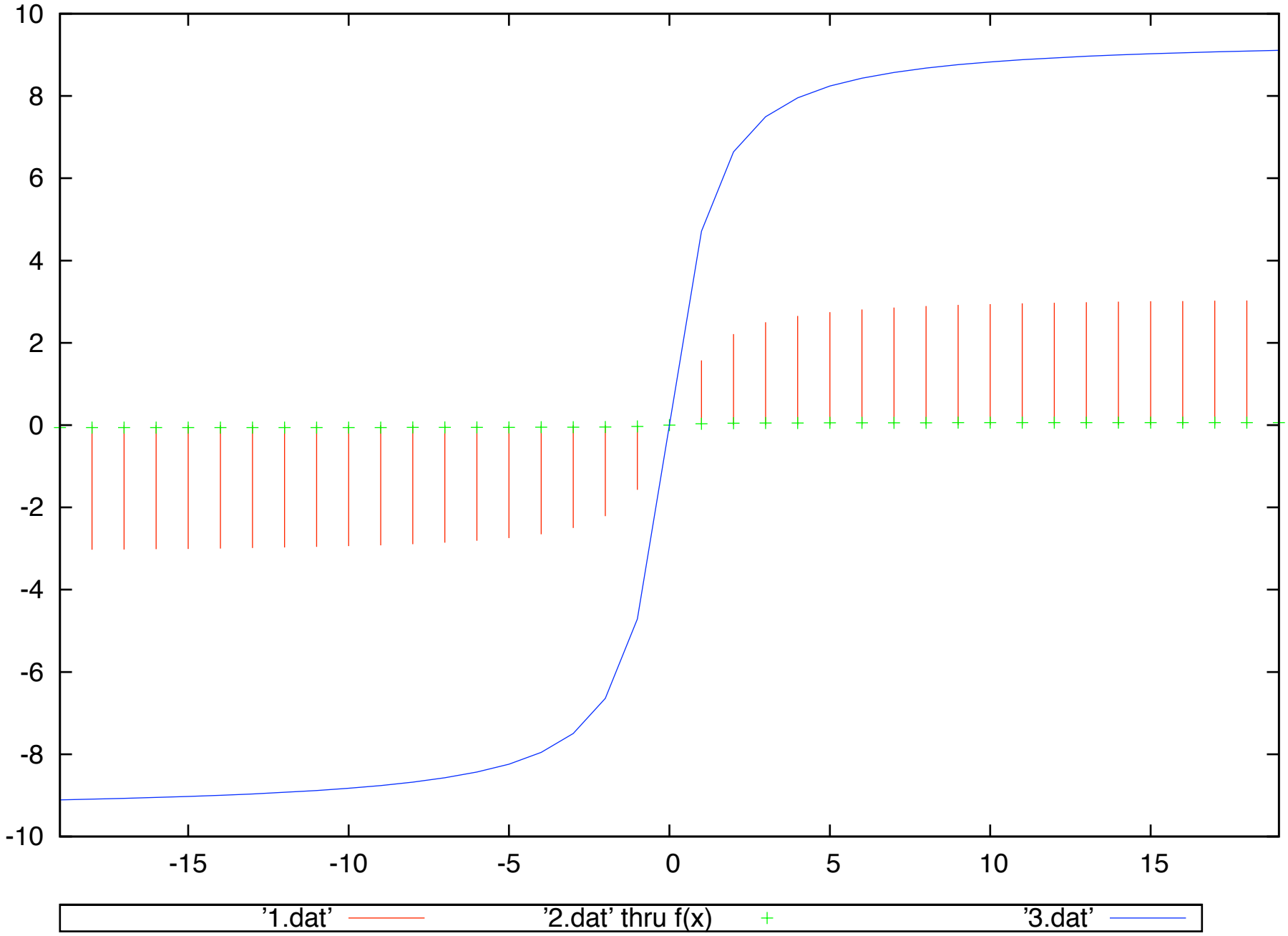

besj0(x)*0.12e1
(x**besj0(x))-2.5
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-30
-20
-10
0
10
20

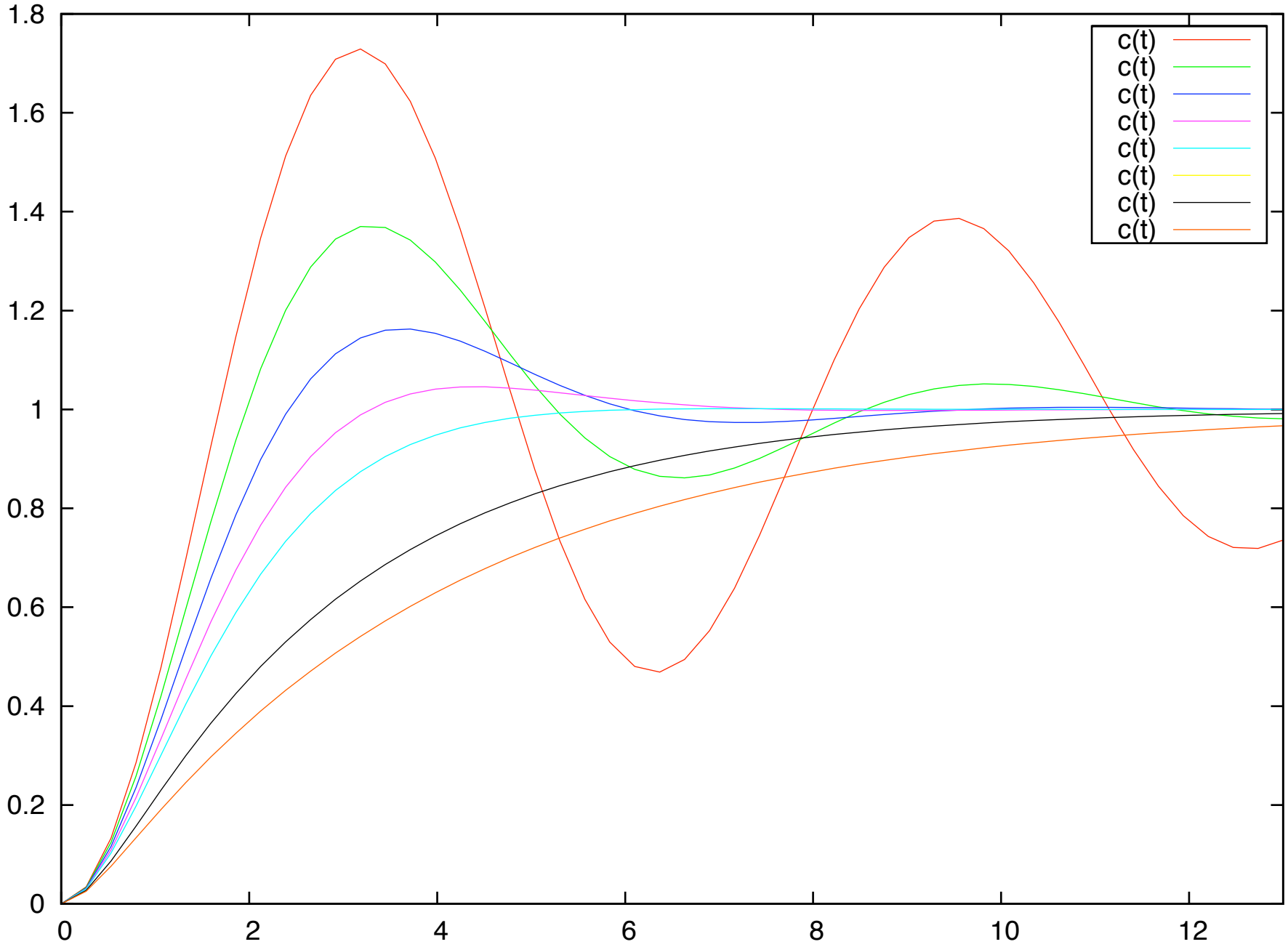

real(sin(x)**besj0(x))
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-10
-5
0
5
10

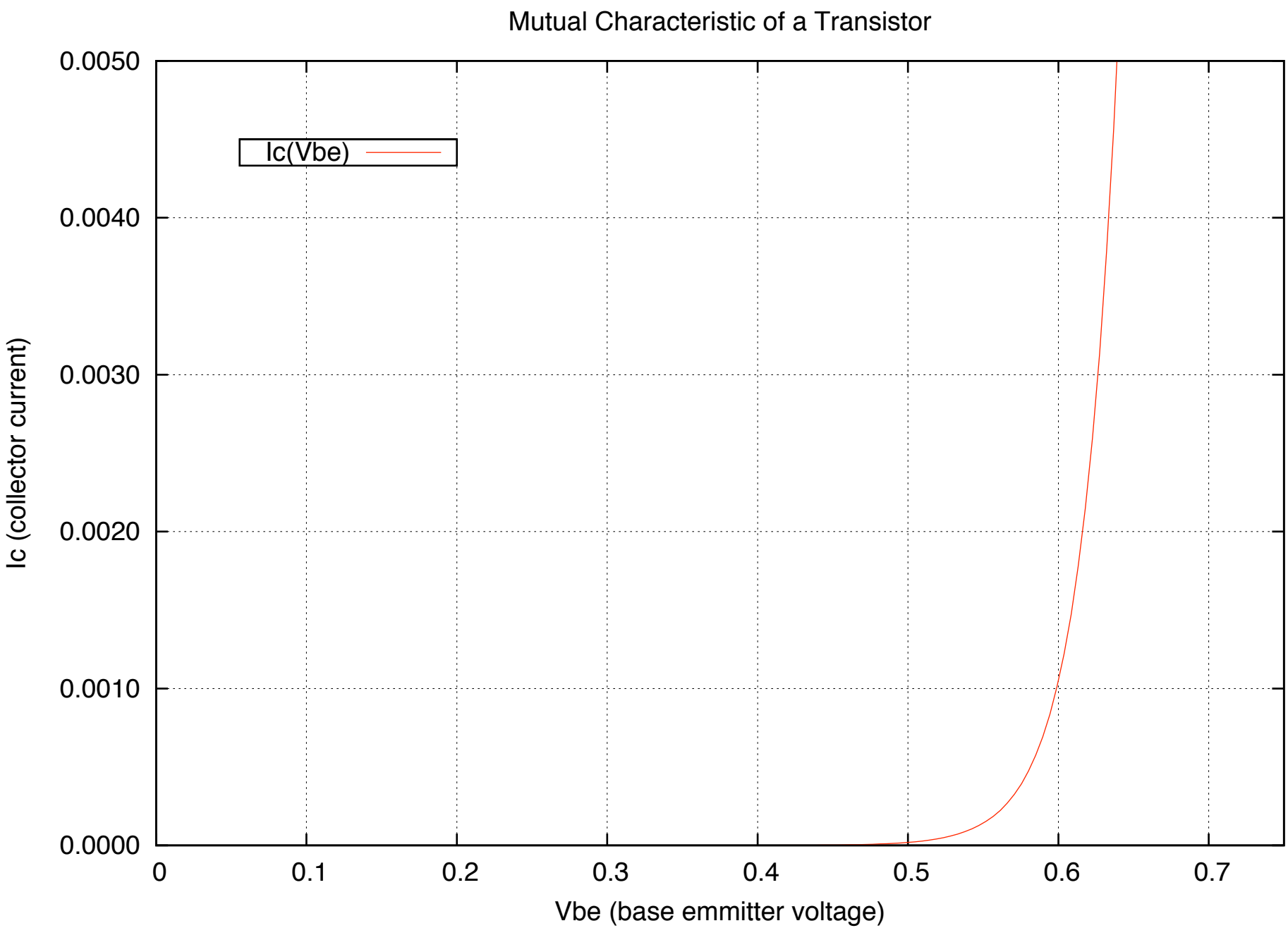

4
2
0
-2
-4
-15
-10
-5
0
5
10
15
real(tan(x)/atan(x))
1/x

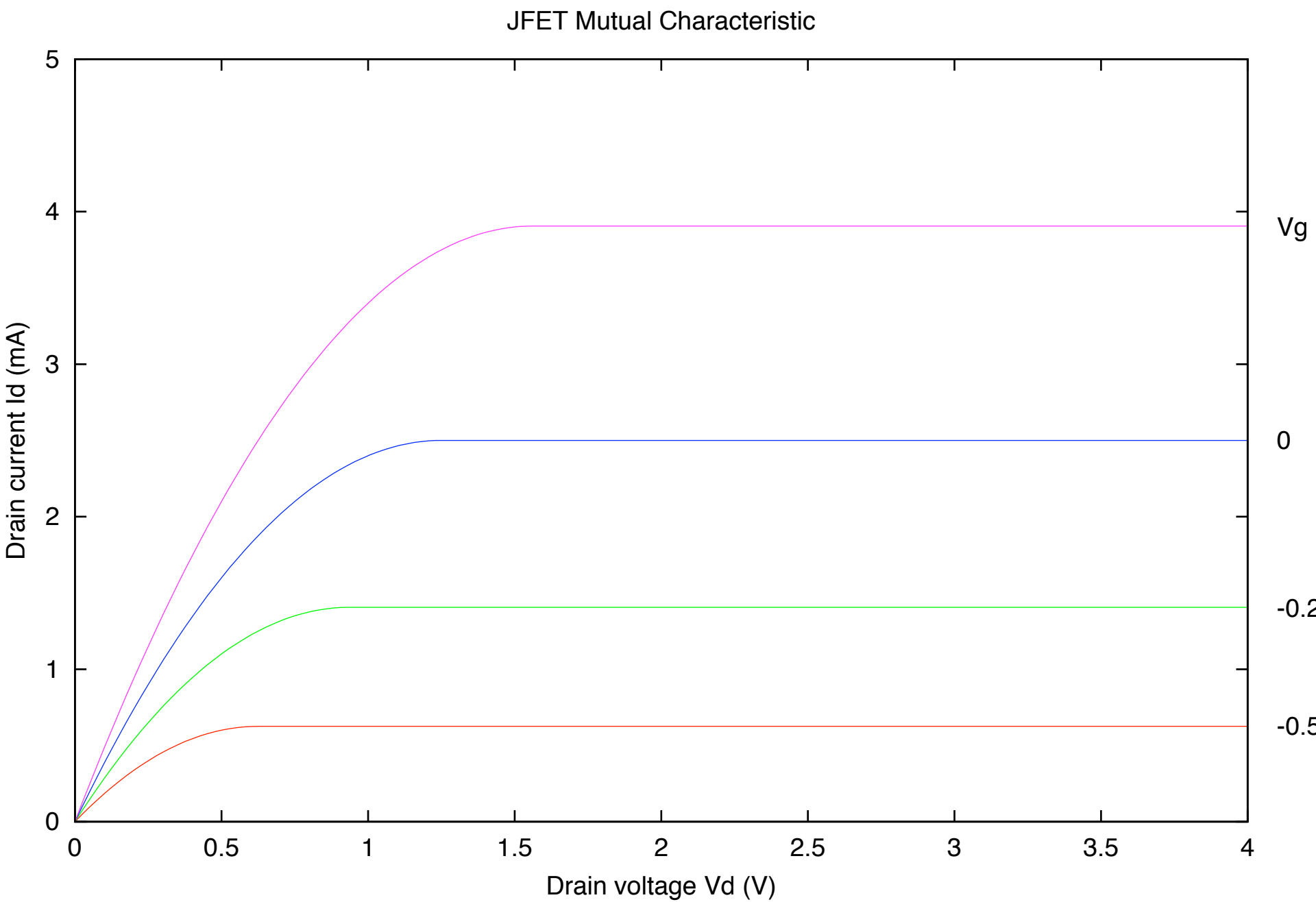

2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-30
-20
-10
0
10
20
sin(x*20)*atan(x)

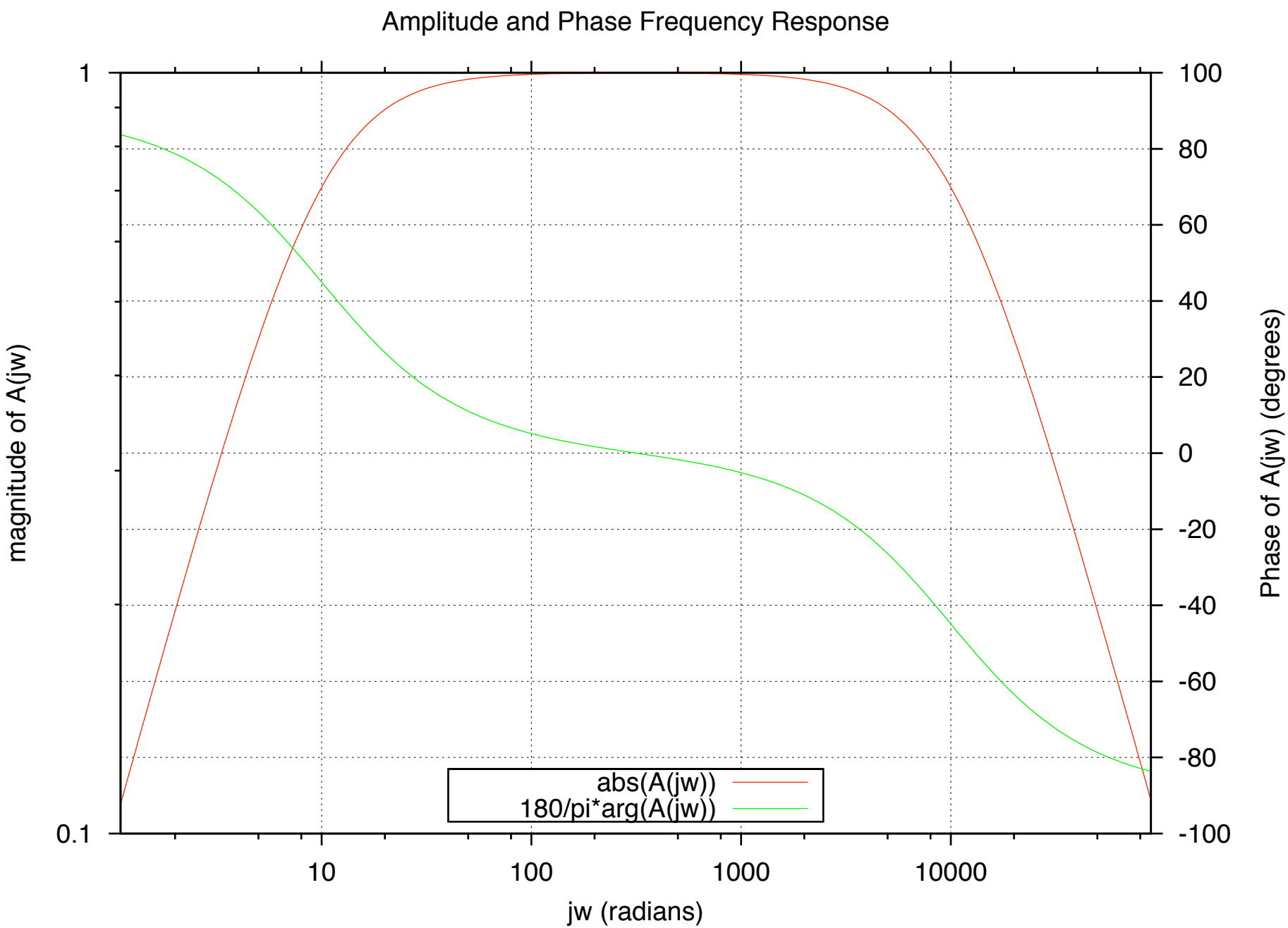

10
8
6
4
2
0
-2
-4
-6
-8
-10
-15
-10
-5
0
5
10
15
'1.dat'
'2.dat'
'3.dat'

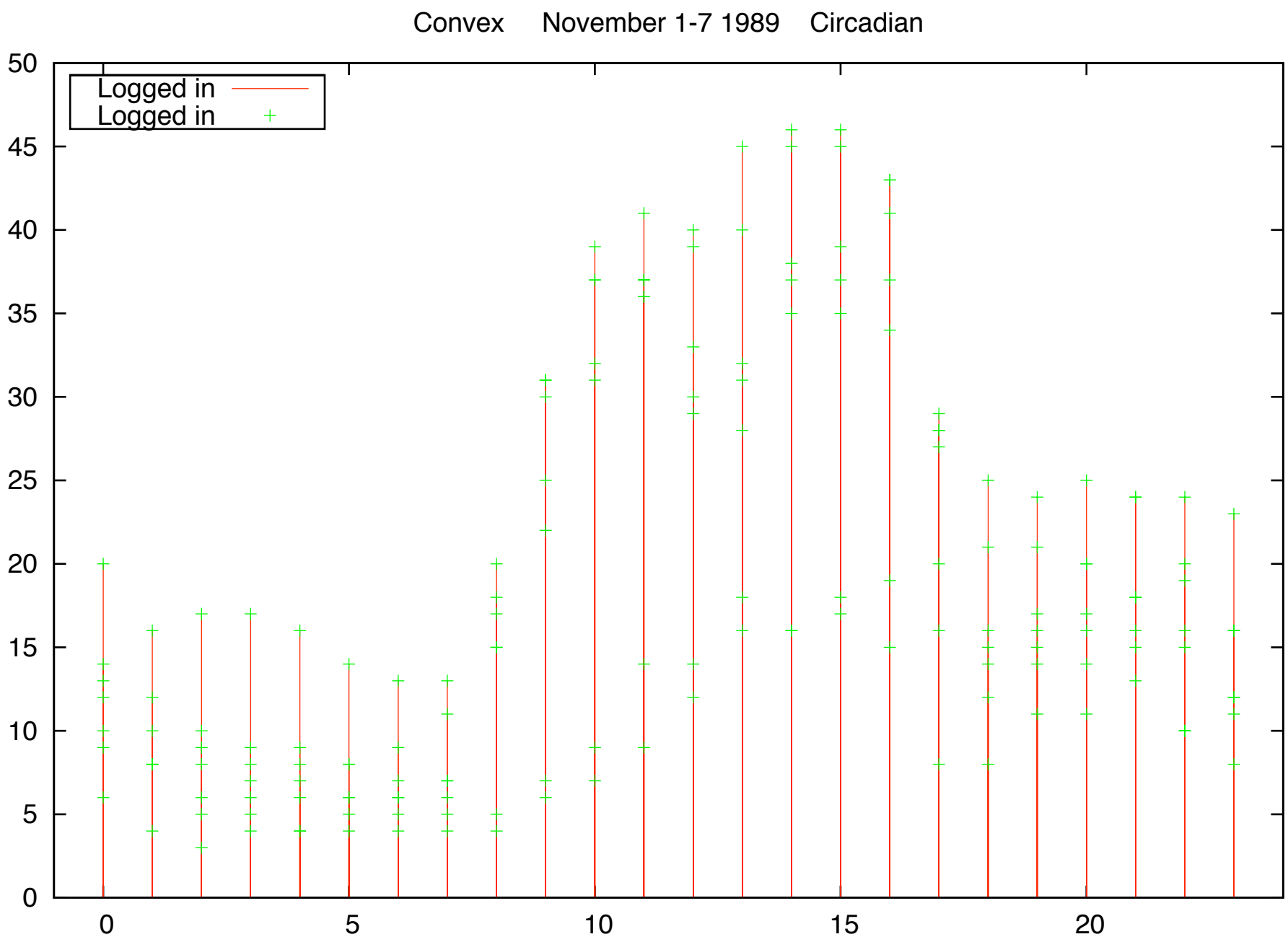

10
8
6
4
2
0
-2
-4
-6
-8
-10
-15
-10
-5
0
5
10
15
'1.dat'
'2.dat' thru f(x)
'3.dat'

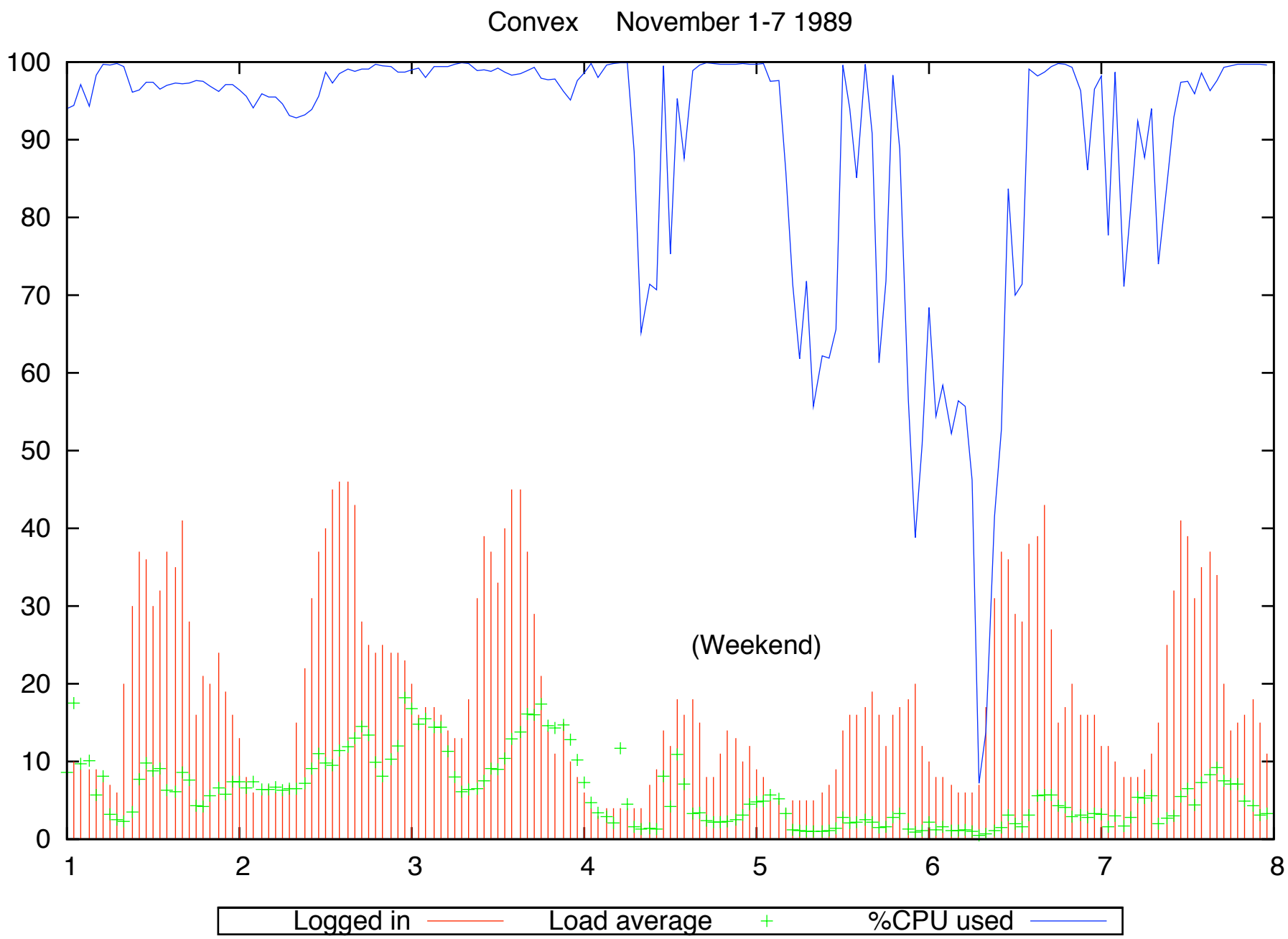

1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
12
c(t)
c(t)
c(t)
c(t)
c(t)
c(t)
c(t)
c(t)

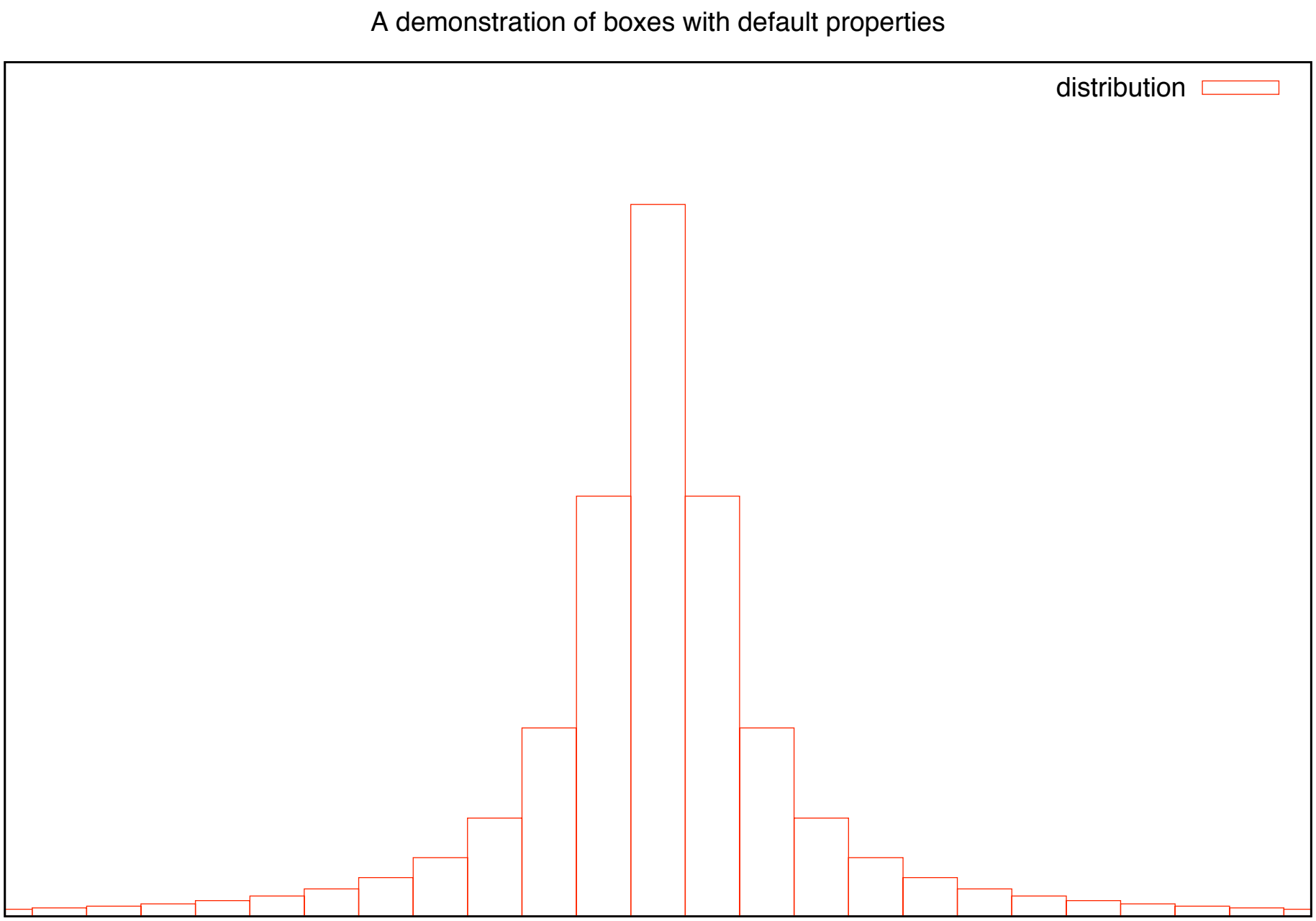

Mutual Characteristic of a Transistor
Ic(Vbe)
Ic (collector current)
Vbe (base emmitter voltage)
0.0050
0.0040
0.0030
0.0020
0.0010
0.0000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

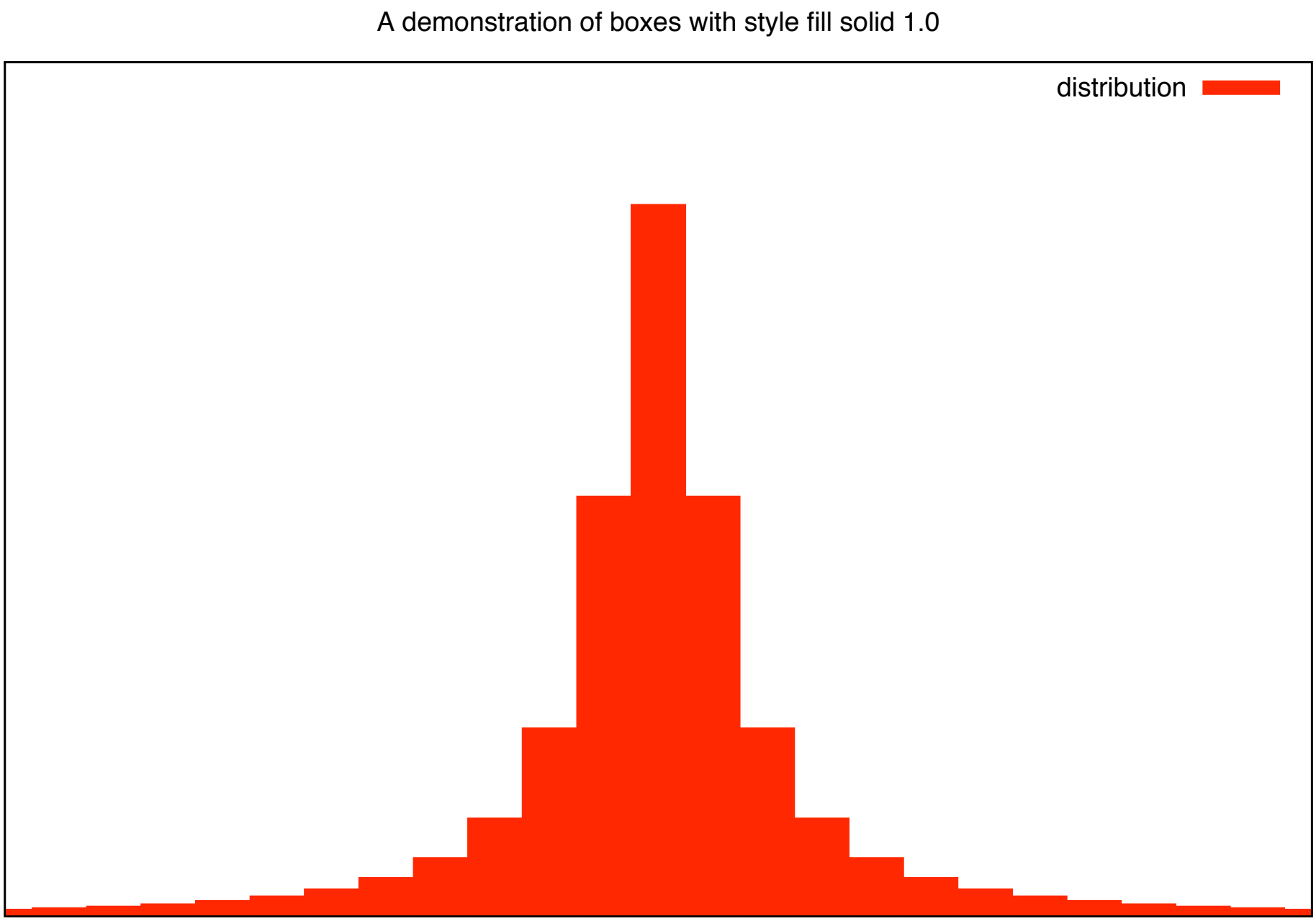

JFET Mutual Characteristic
Drain current Id (mA)
Drain voltage Vd (V)
0
1
2
3
4
5
0
0.5
1
1.5
2
2.5
3
3.5
4
Vg
0
-0.2
-0.5

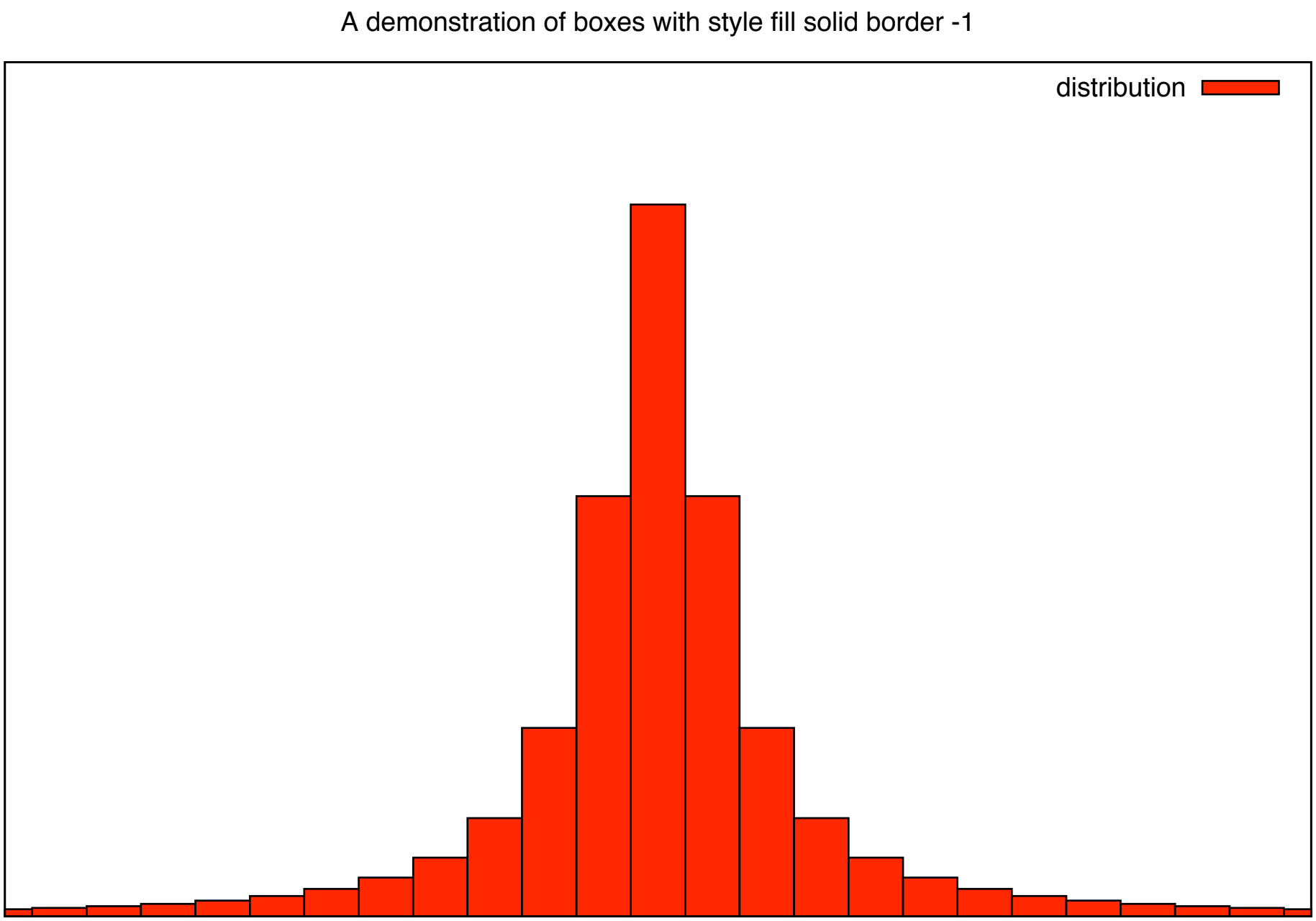

Amplitude and Phase Frequency Response
1
0.1
magnitude of A(jw)
100
80
60
40
20
0
-20
-40
-60
-80
-100
Phase of A(jw) (degrees)
10
100
1000
10000
jw (radians)
abs(A(jw))
180/pi*arg(A(jw))

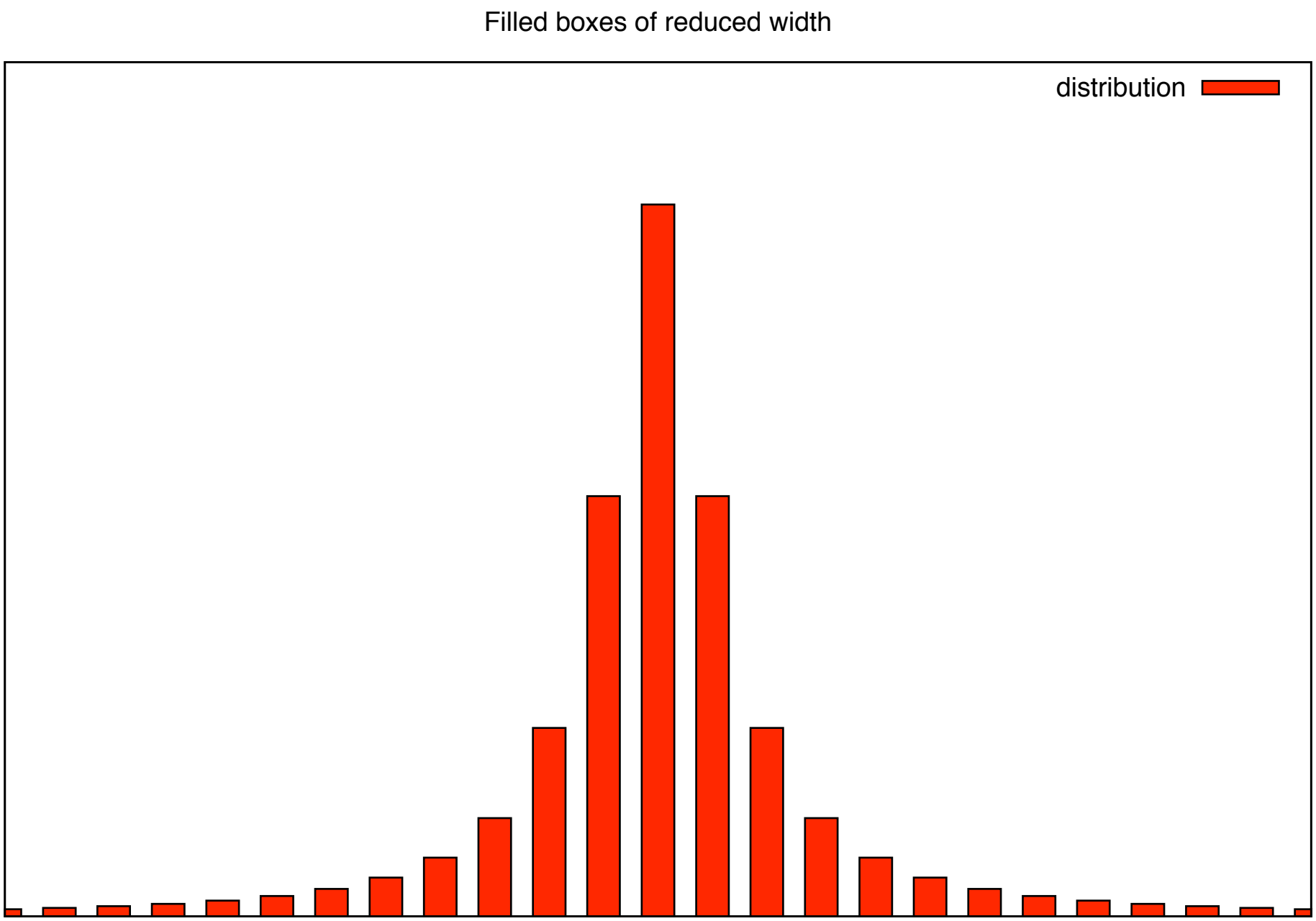

Convex November 1-7 1989 Circadian
Logged in
Logged in
50
45
40
35
30
25
20
15
10
5
0
0
5
10
15
20

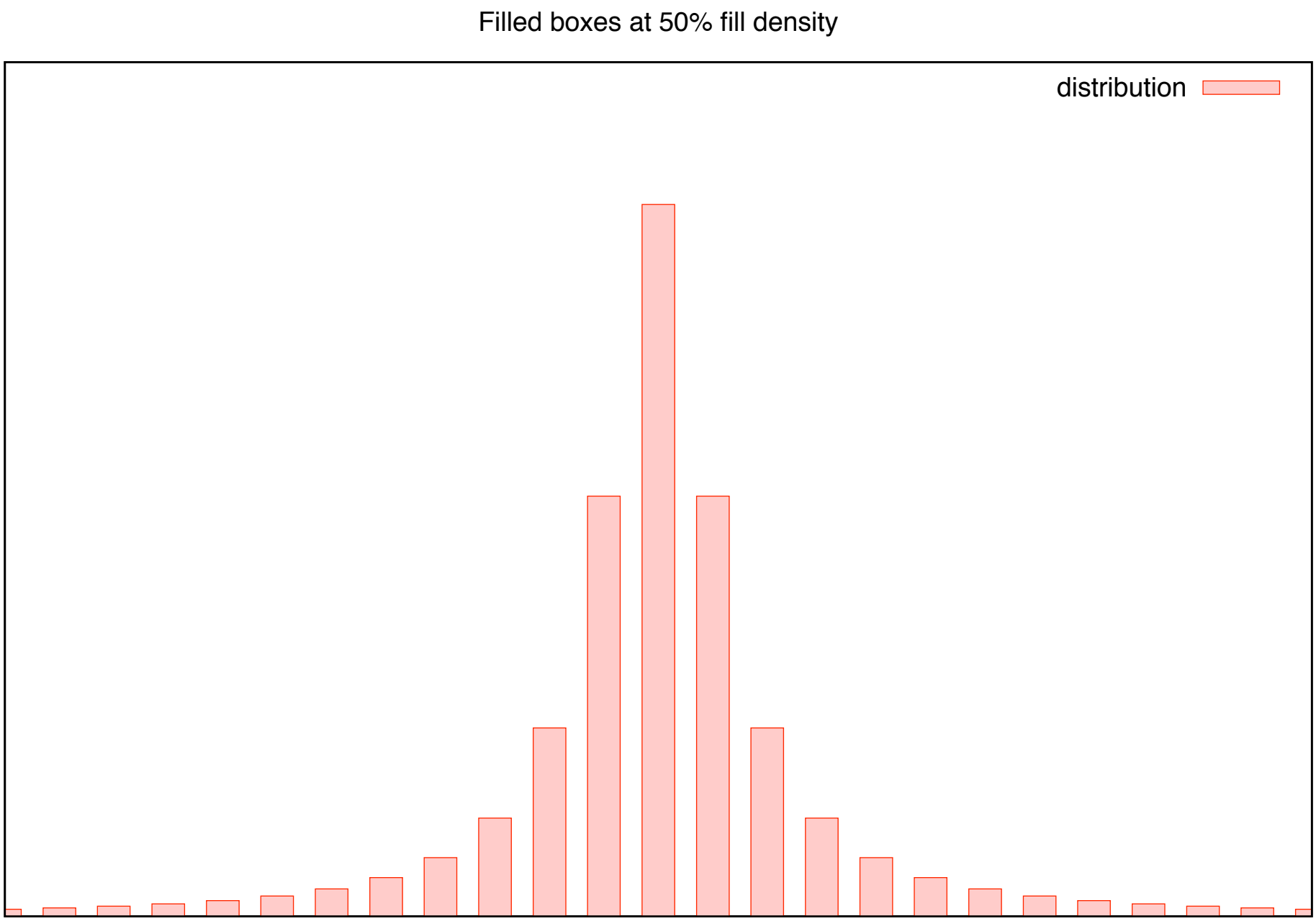

Convex November 1-7 1989
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
(Weekend)
Logged in
Load average
%CPU used


A demonstration of boxes with default properties
distribution

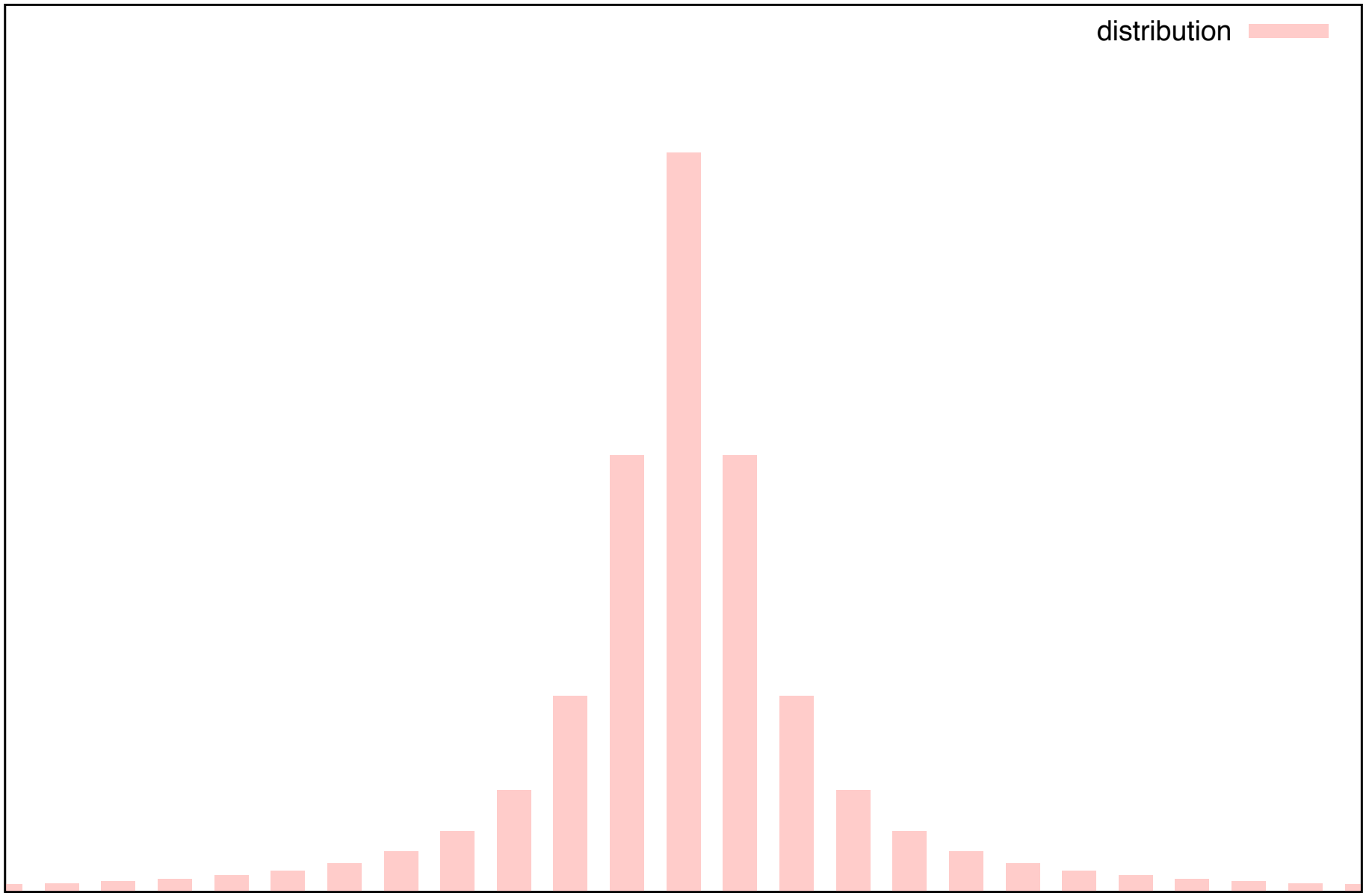

A demonstration of boxes with style fill solid 1.0
distribution


A demonstration of boxes with style fill solid border -1
distribution

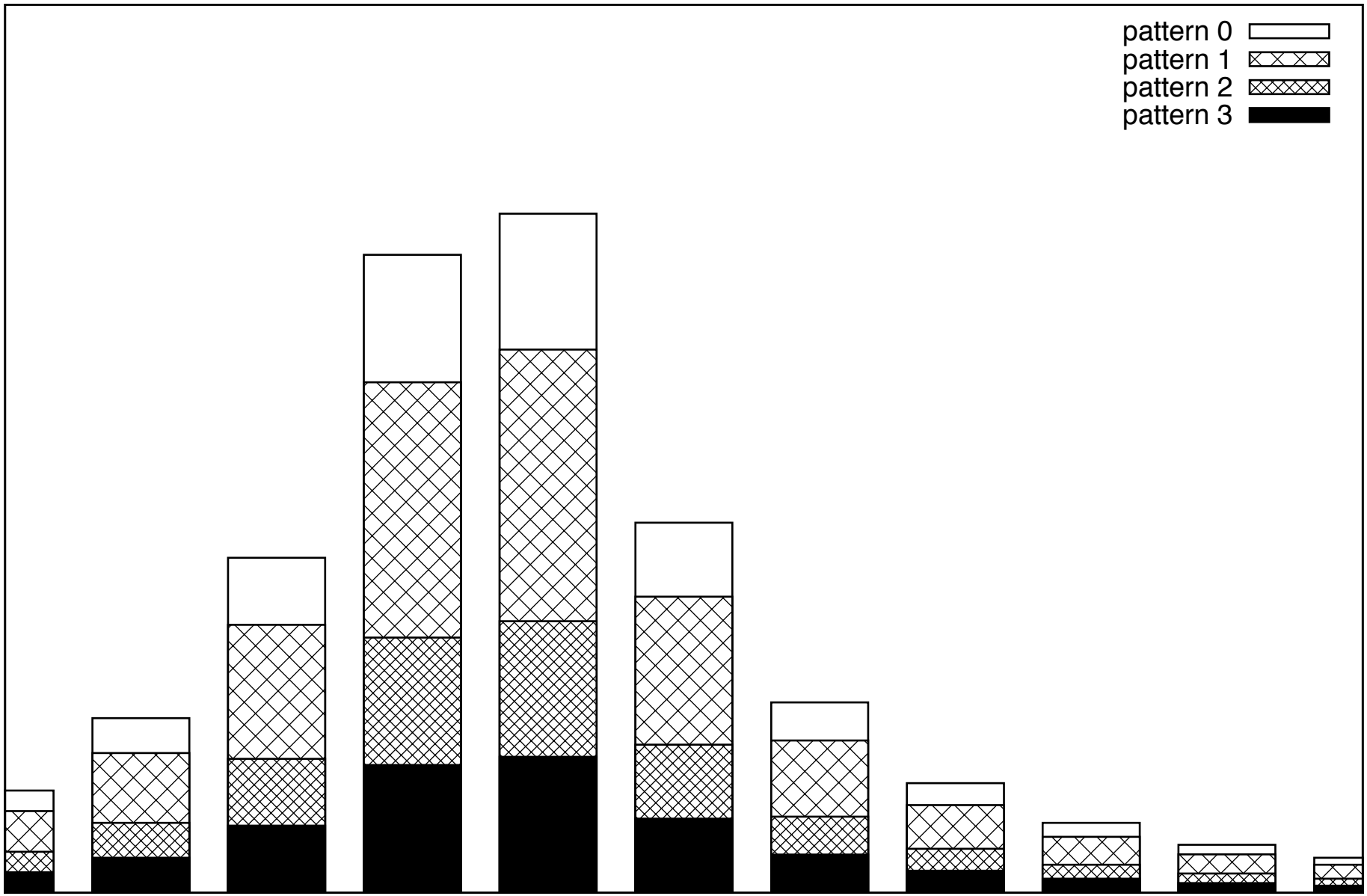

Filled boxes of reduced width
distribution

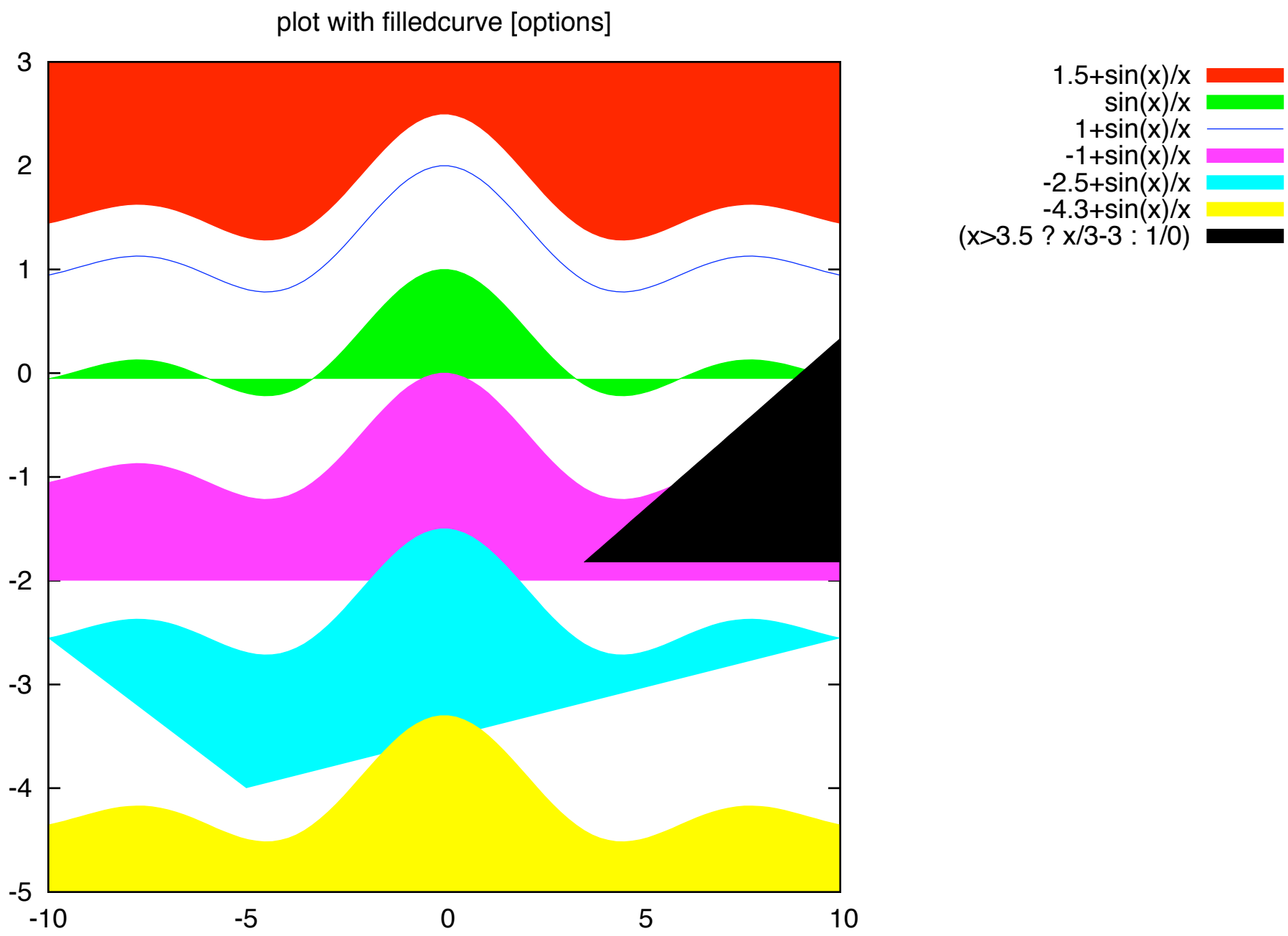

Filled boxes at 50% fill density
distribution

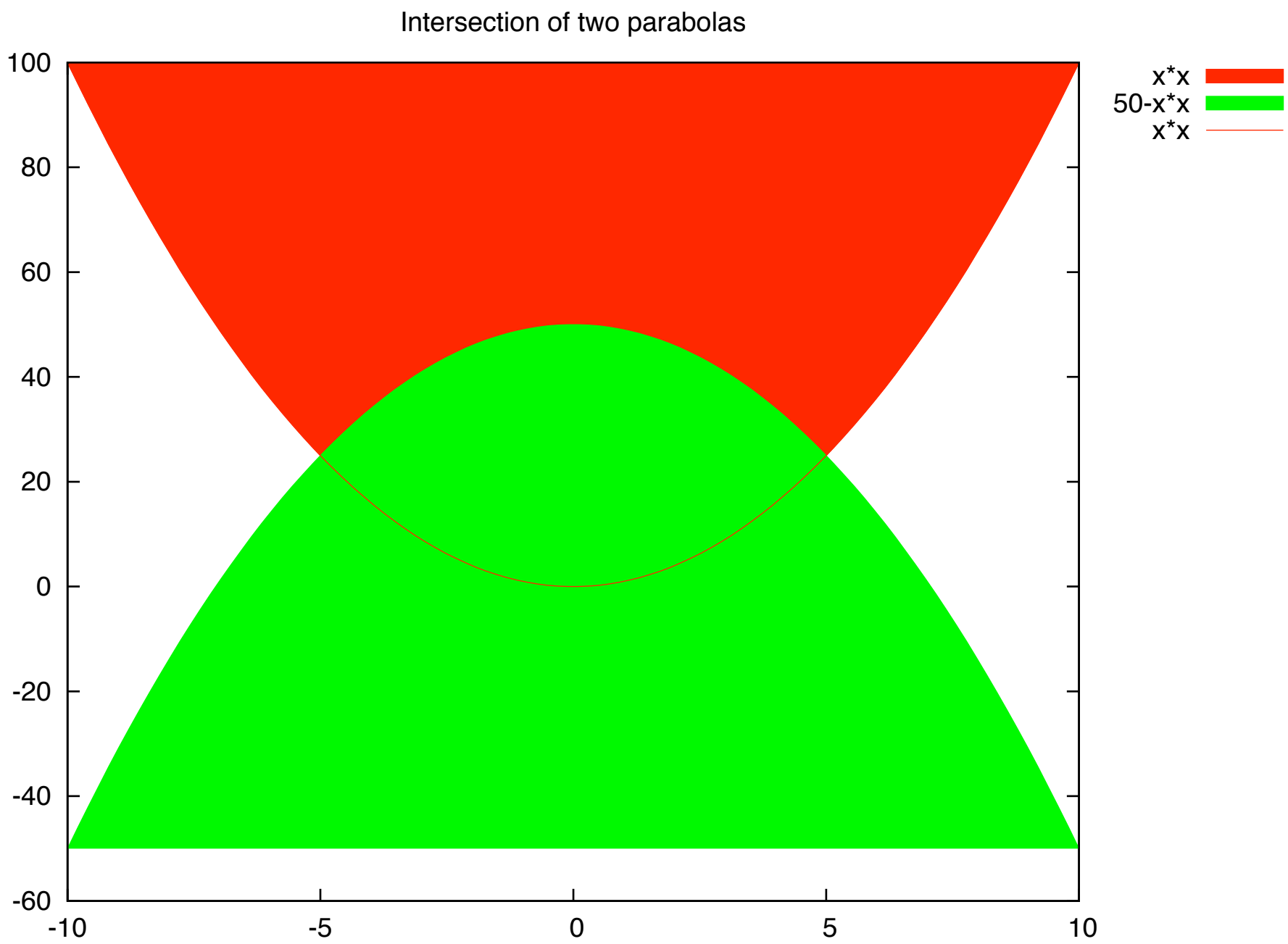
A demonstration of boxes with style fill solid 0.25 noborder

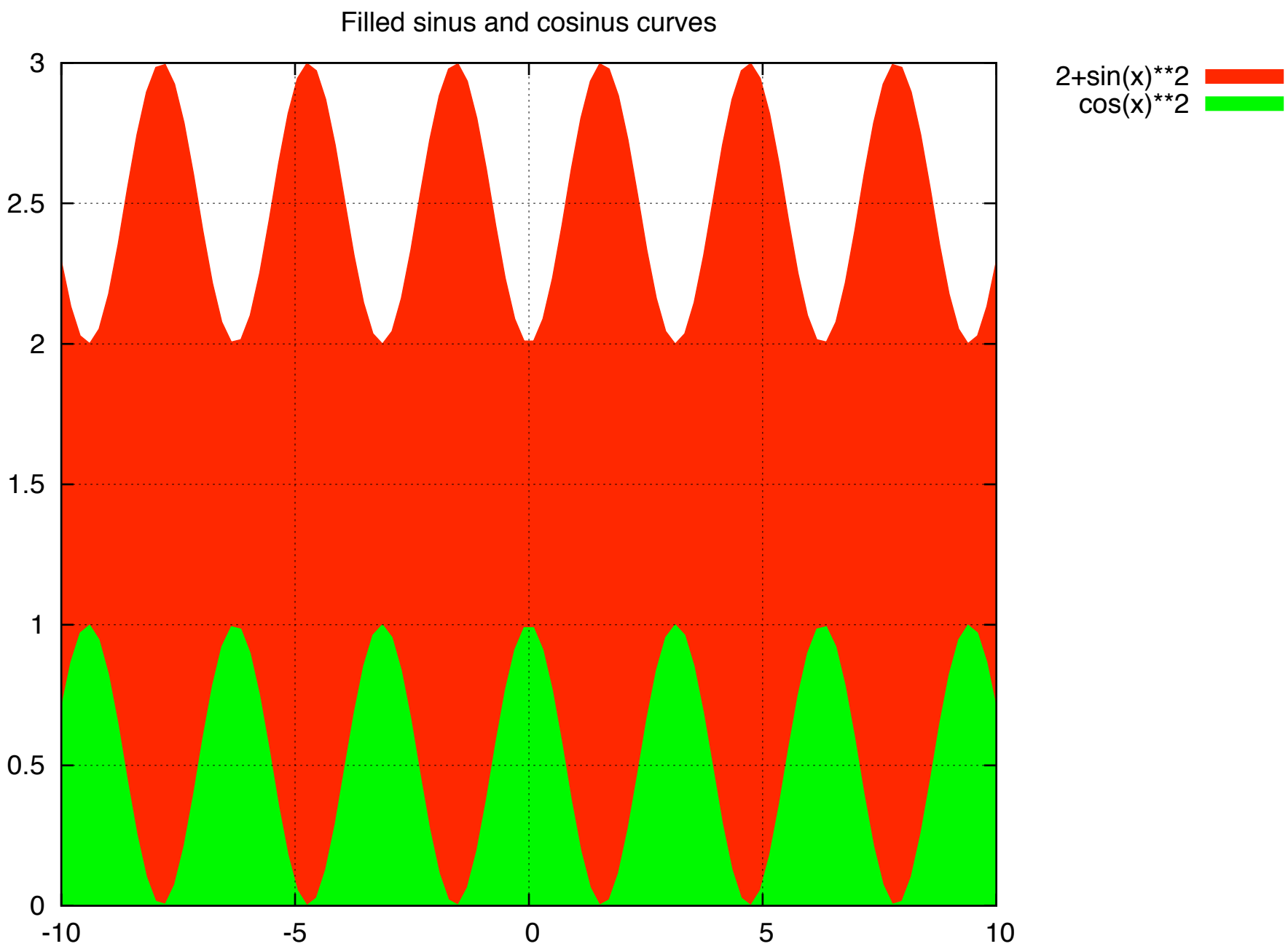


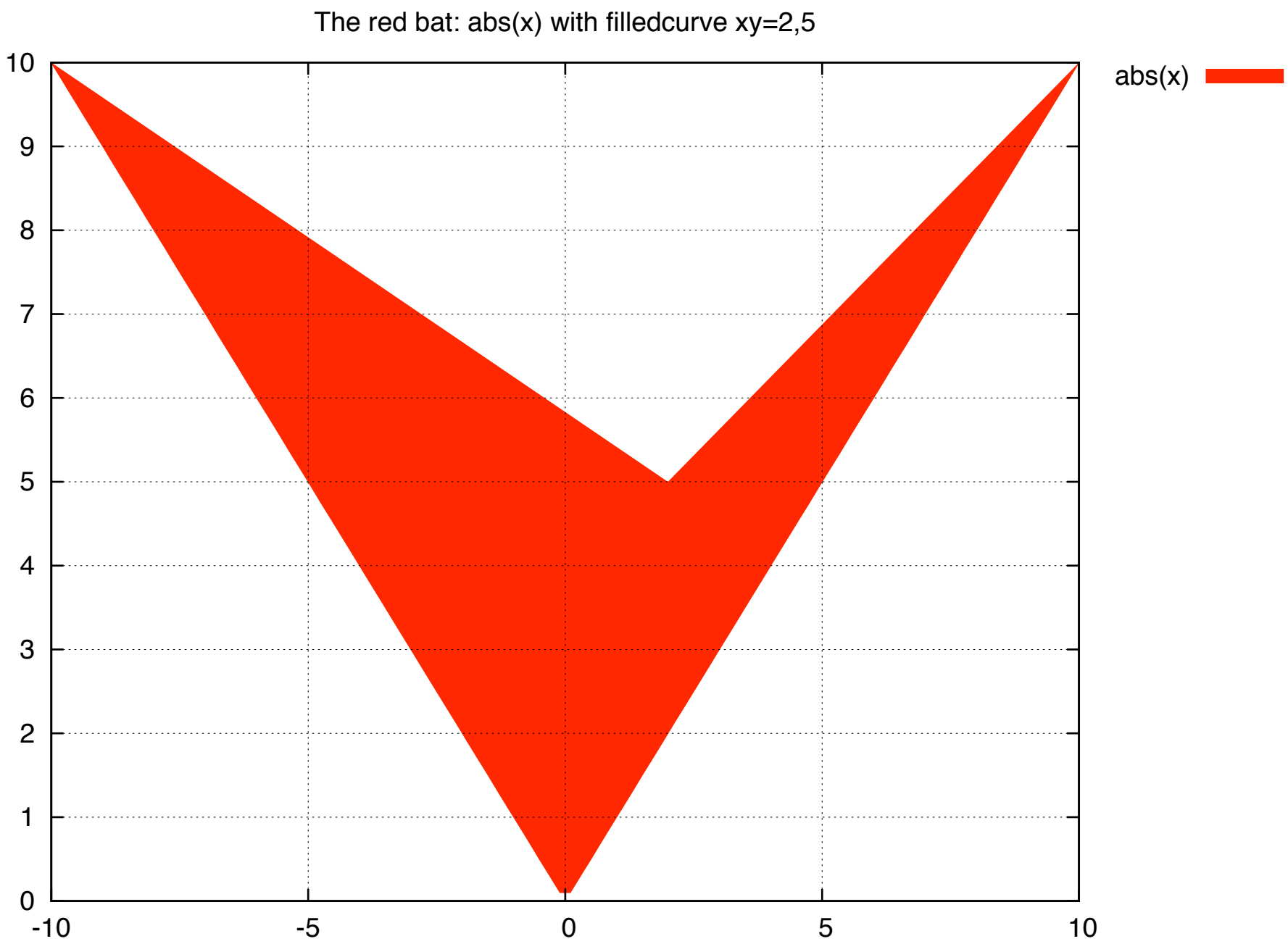
A demonstration of boxes in mono with style fill pattern

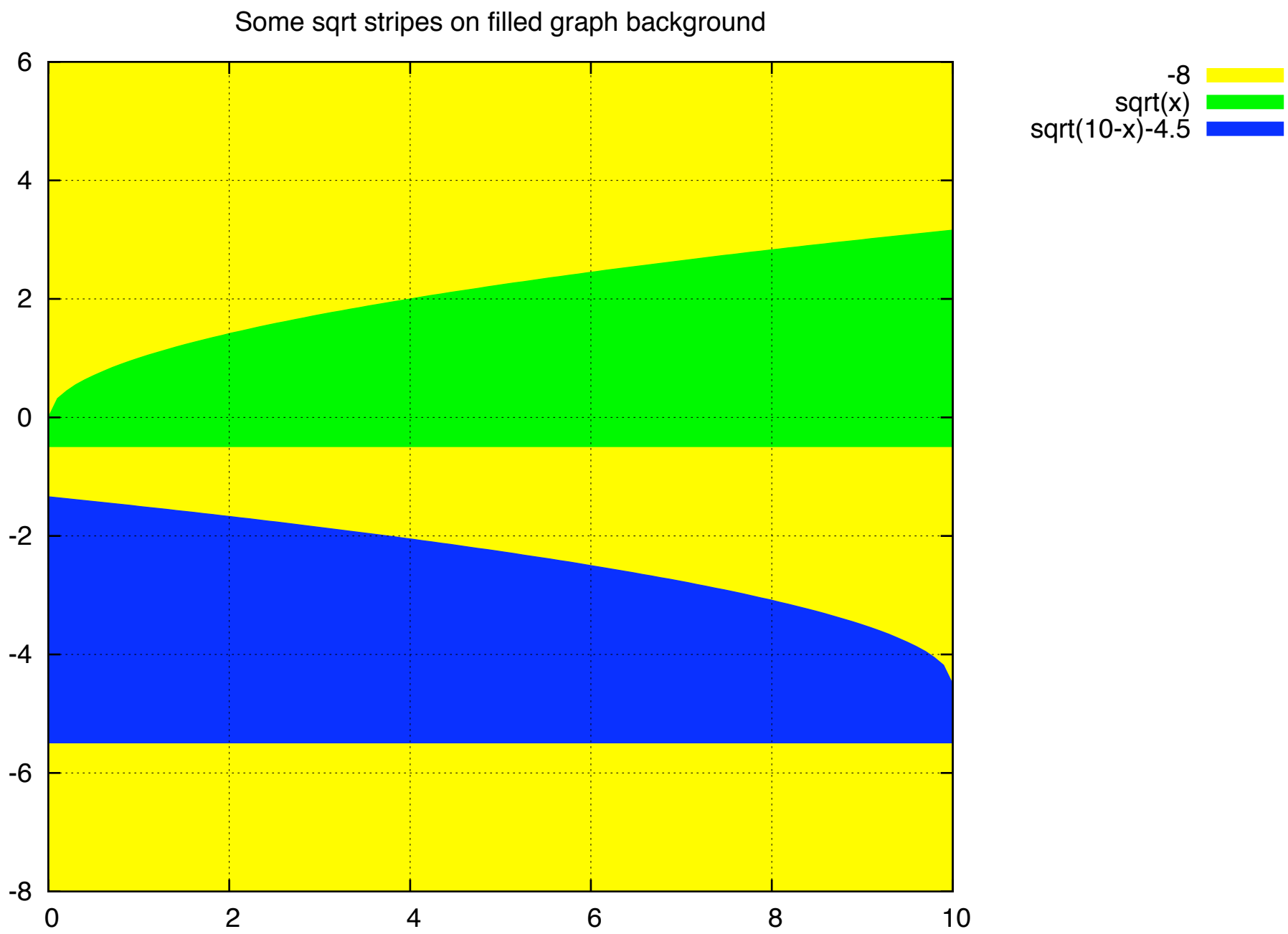



plot with filledcurve [options]
3
2
1
0
-1
-2
-3
-4
-5
-10
-5
0
5
10
1.5+sin(x)/x
sin(x)/x
1+sin(x)/x
-1+sin(x)/x
-2.5+sin(x)/x
-4.3+sin(x)/x
(x>3.5 ? x/3-3 : 1/0)

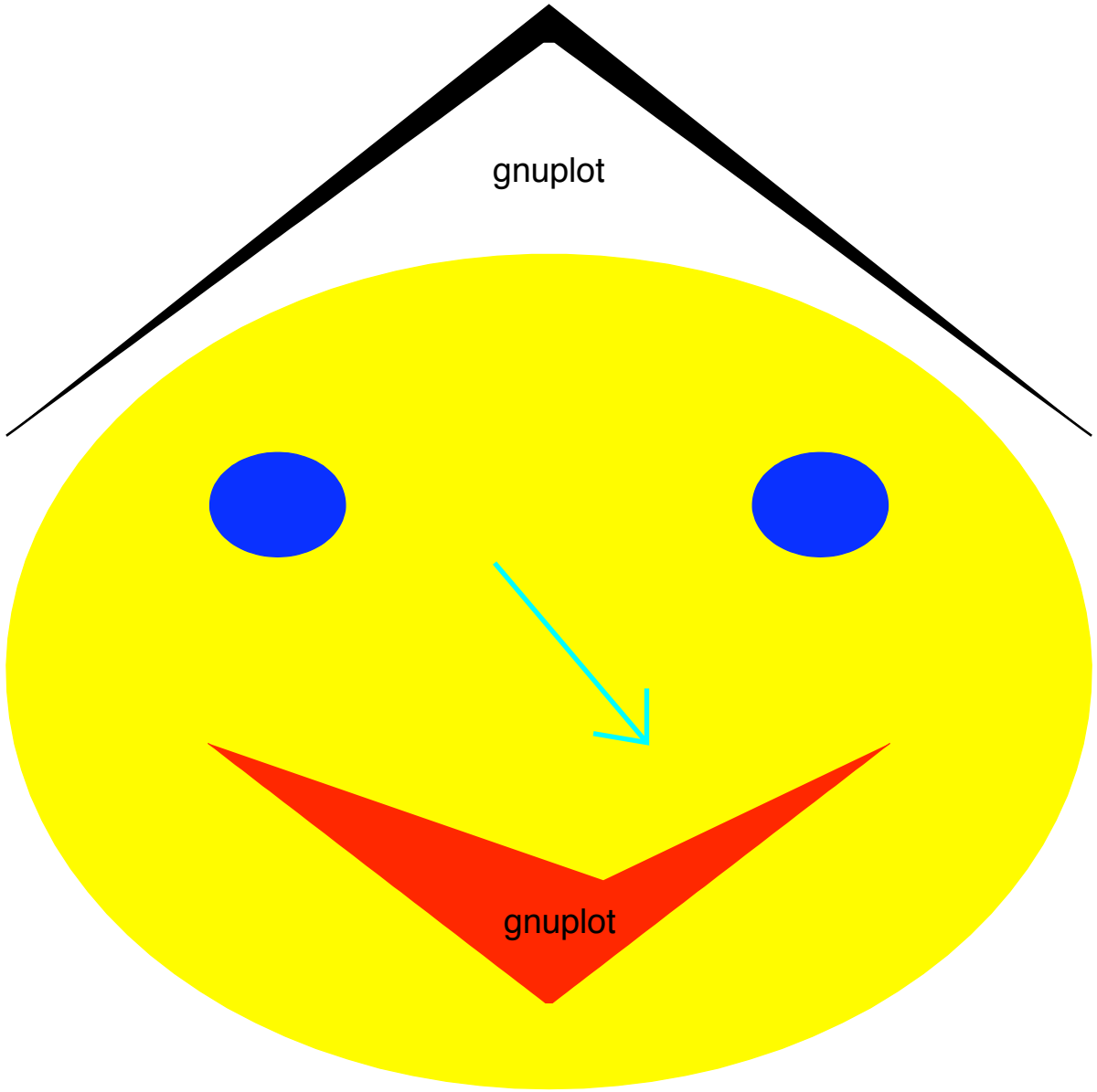

Intersection of two parabolas
100
80
60
40
20
0
-20
-40
-60
-10
-5
0
5
10
x*x
50-x*x
x*x

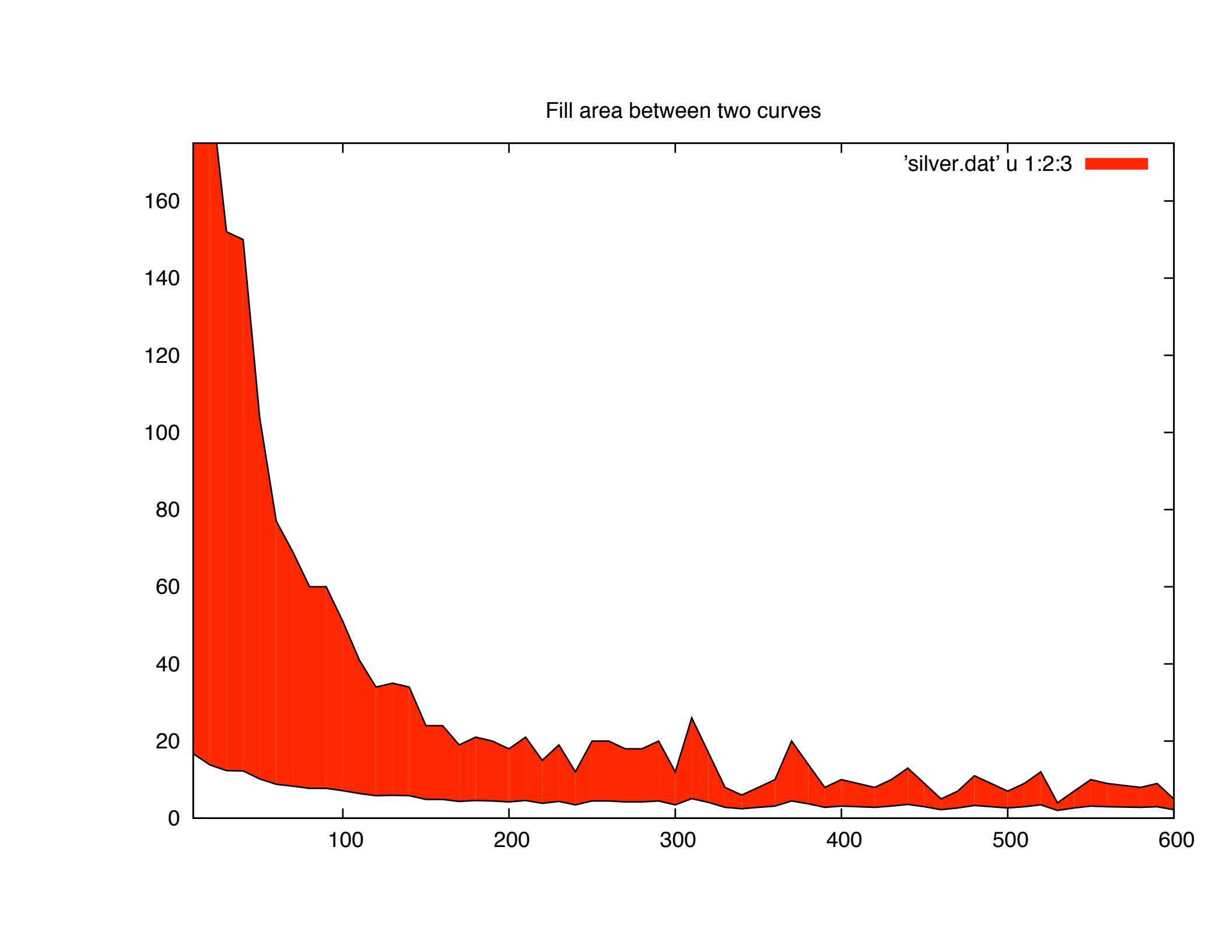

Filled sinus and cosinus curves
2+sin(x)**2
cos(x)**2
3
2.5
2
1.5
1
0.5
0
-10
-5
0
5
10

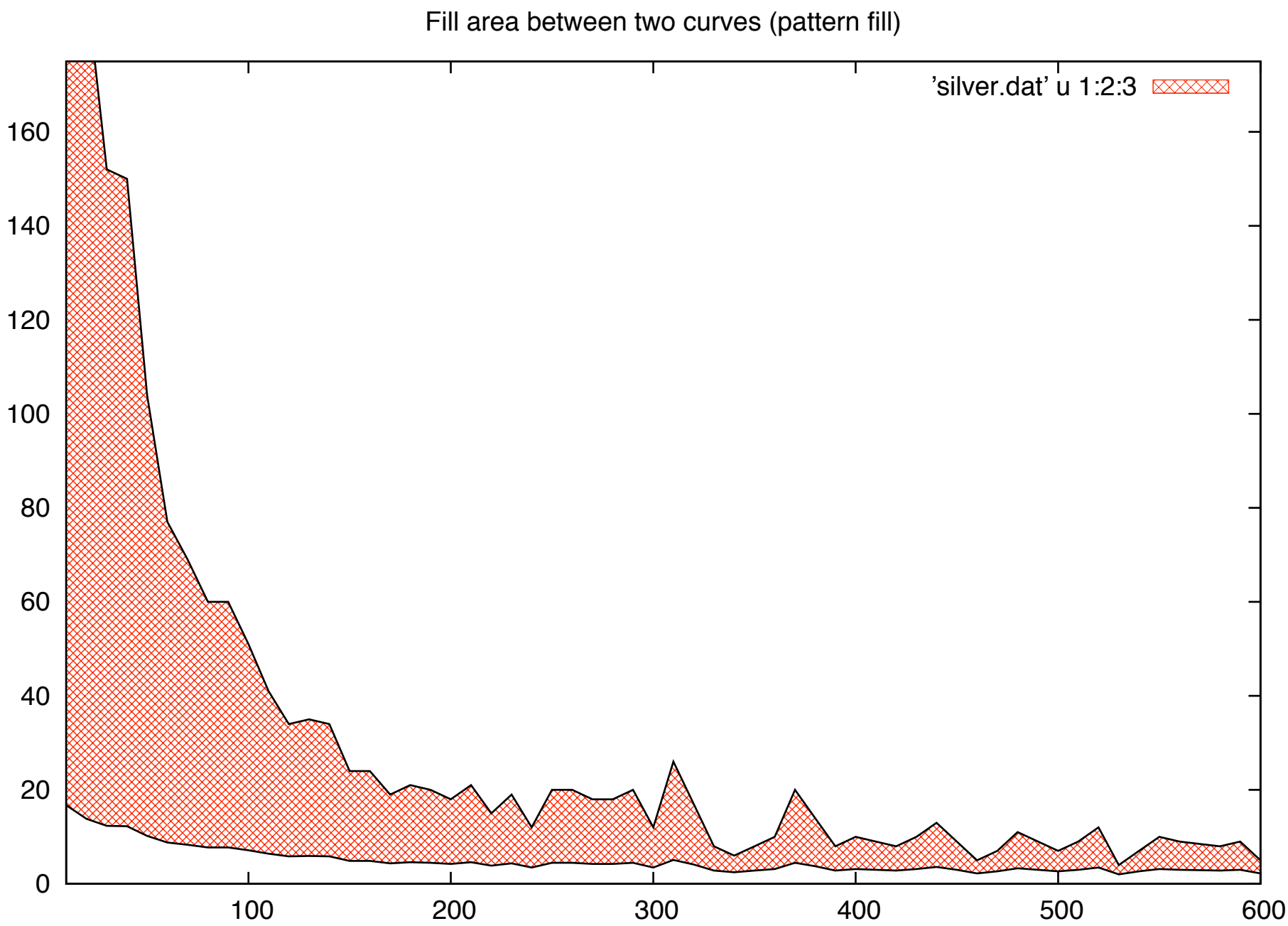

The red bat: abs(x) with filledcurve xy=2,5
abs(x)
10
9
8
7
6
5
4
3
2
1
0
-10
-5
0
5
10

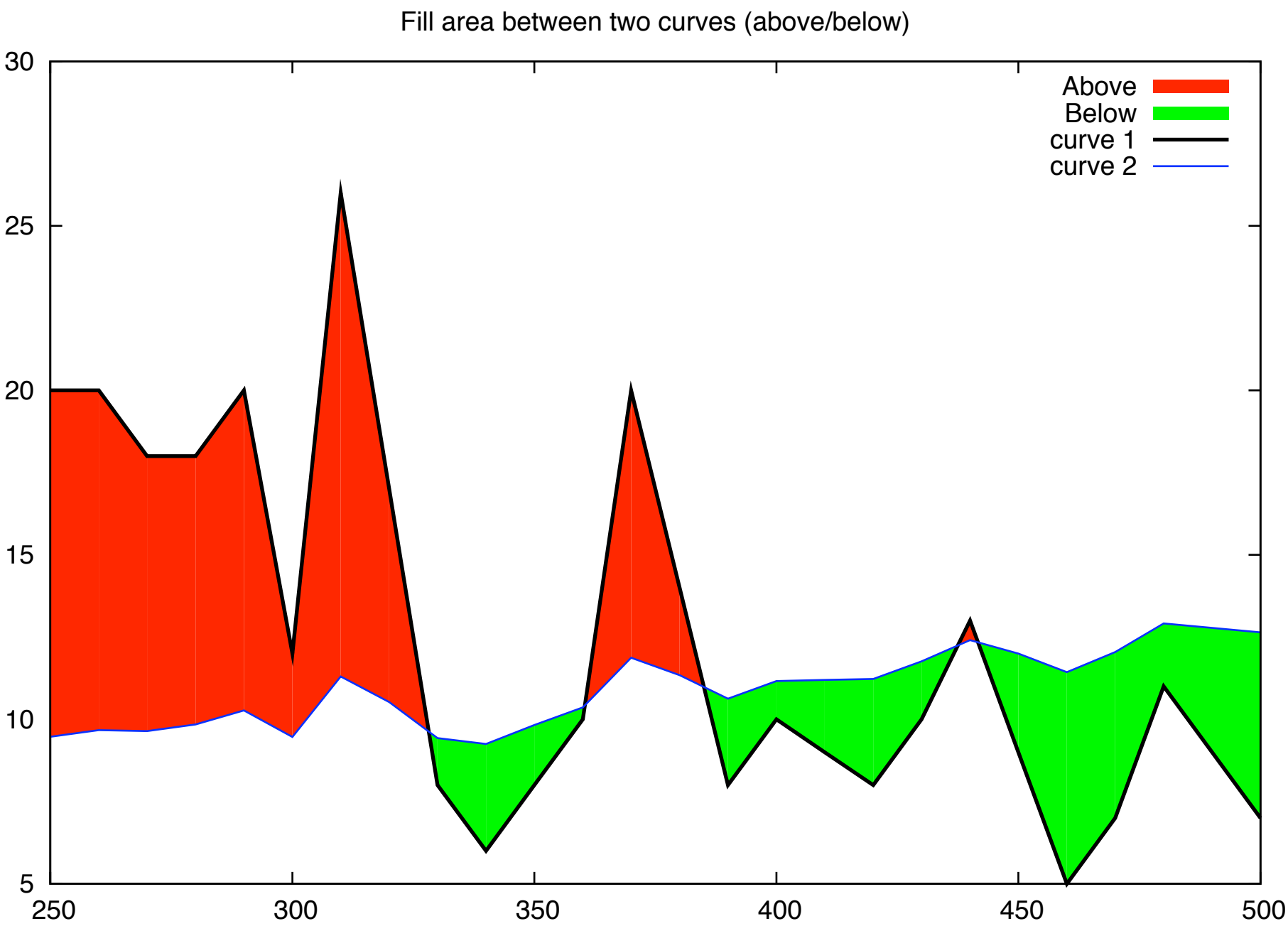

Some sqrt stripes on filled graph background
6
4
2
0
-2
-4
-6
-8
0
2
4
6
8
10
-8
sqrt(x)
sqrt(10-x)-4.5

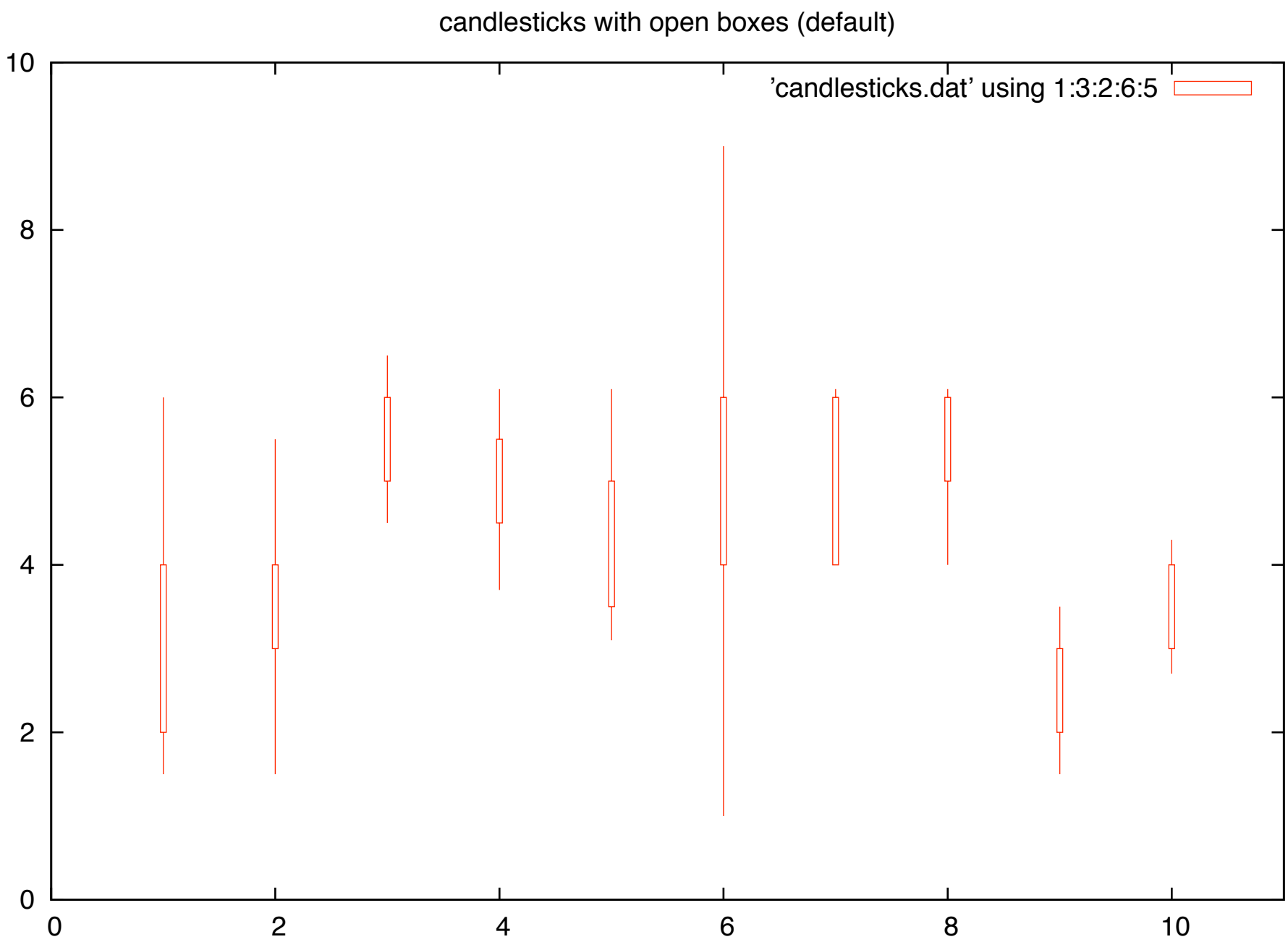
Let’s smile with parametric filled curves

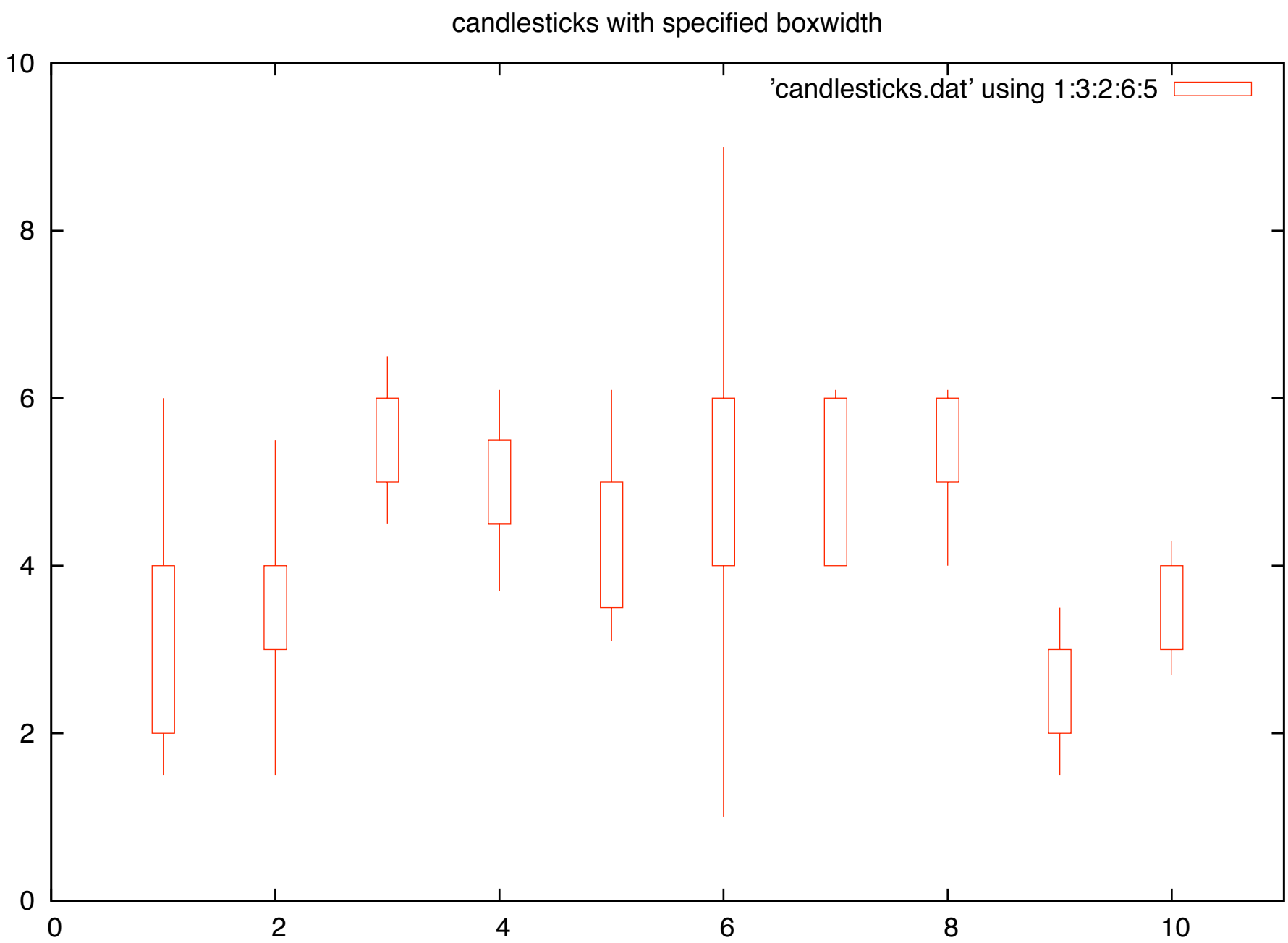


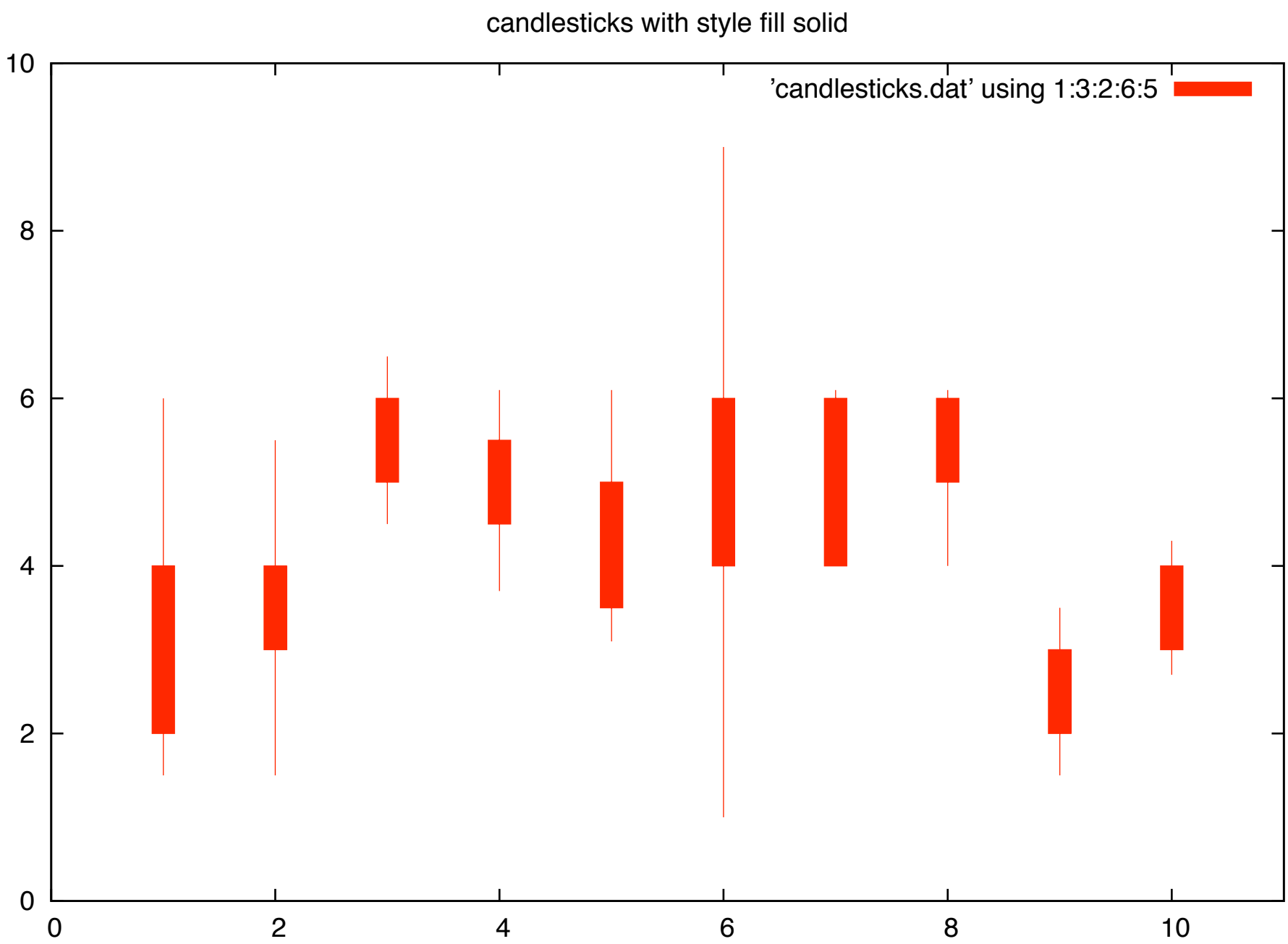

Fill area between two curves
'silver.dat' u 1:2:3
160
140
120
100
80
60
40
20
0
100
200
300
400
500
600

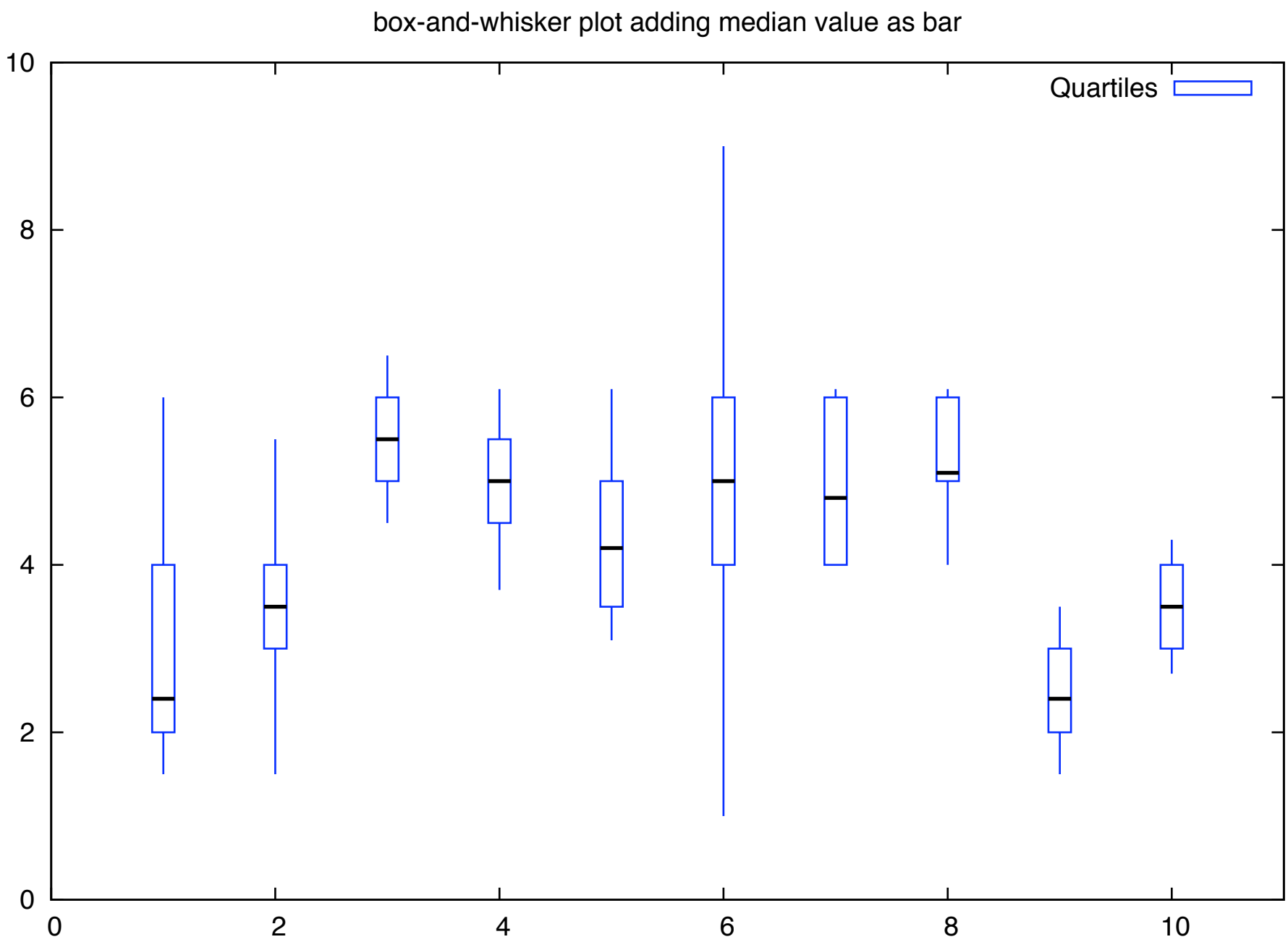

Fill area between two curves (pattern fill)
'silver.dat' u 1:2:3
160
140
120
100
80
60
40
20
0
100
200
300
400
500
600

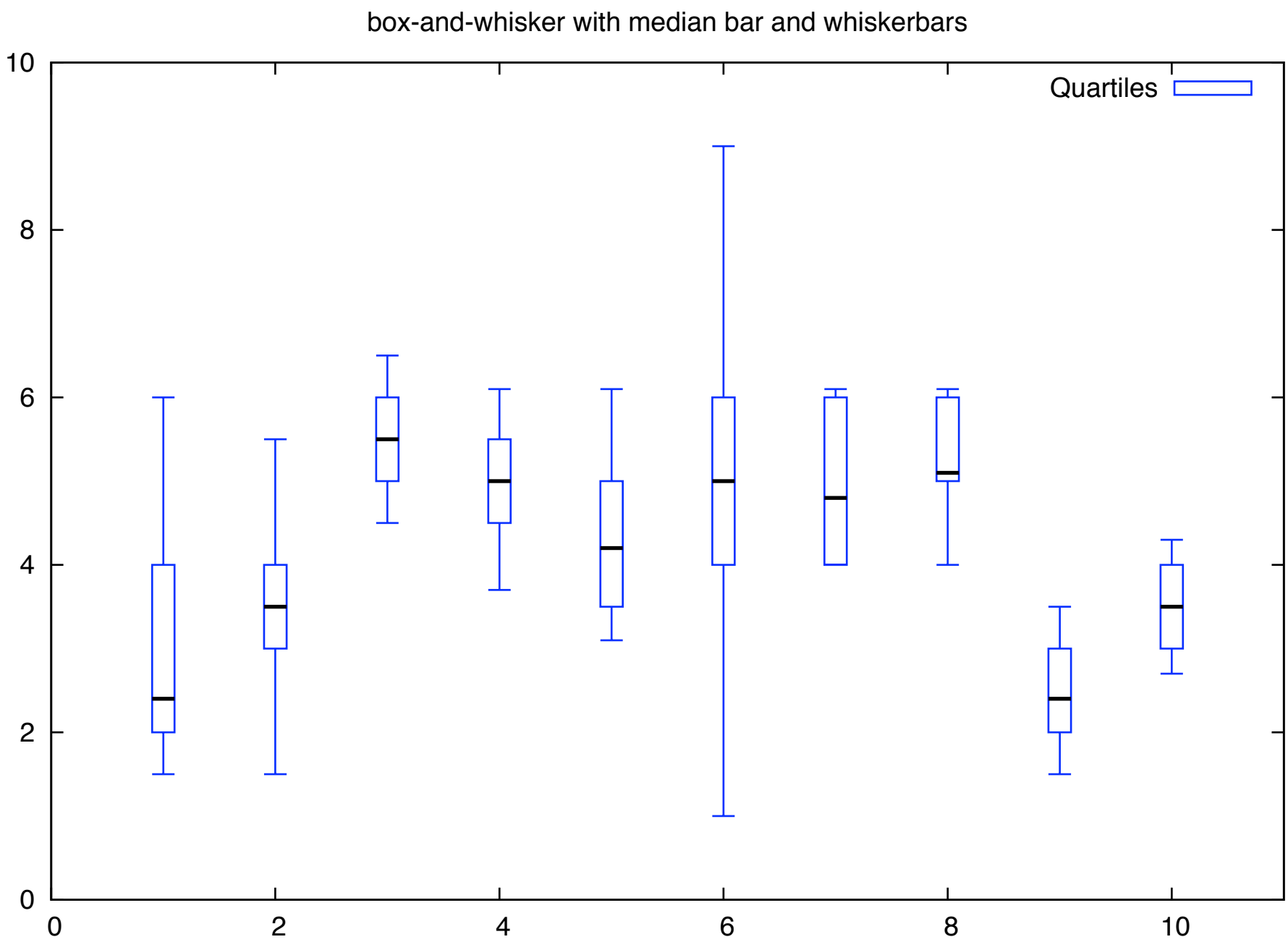

Fill area between two curves (above/below)
Above
Below
curve 1
curve 2
30
25
20
15
10
5
250
300
350
400
450
500

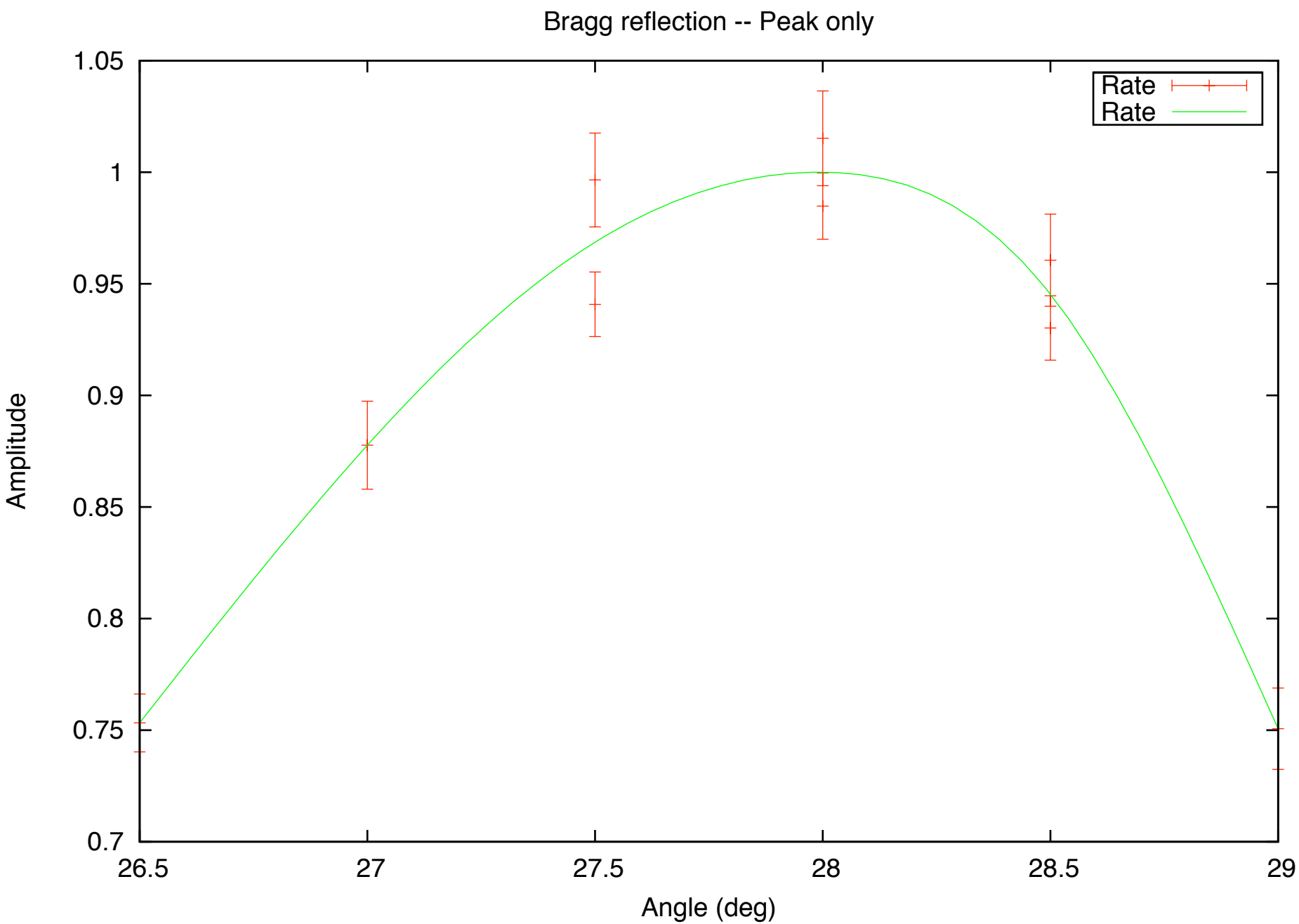

candlesticks with open boxes (default)
'candlesticks.dat' using 1:3:2:6:5
10
8
6
4
2
0
0
2
4
6
8
10

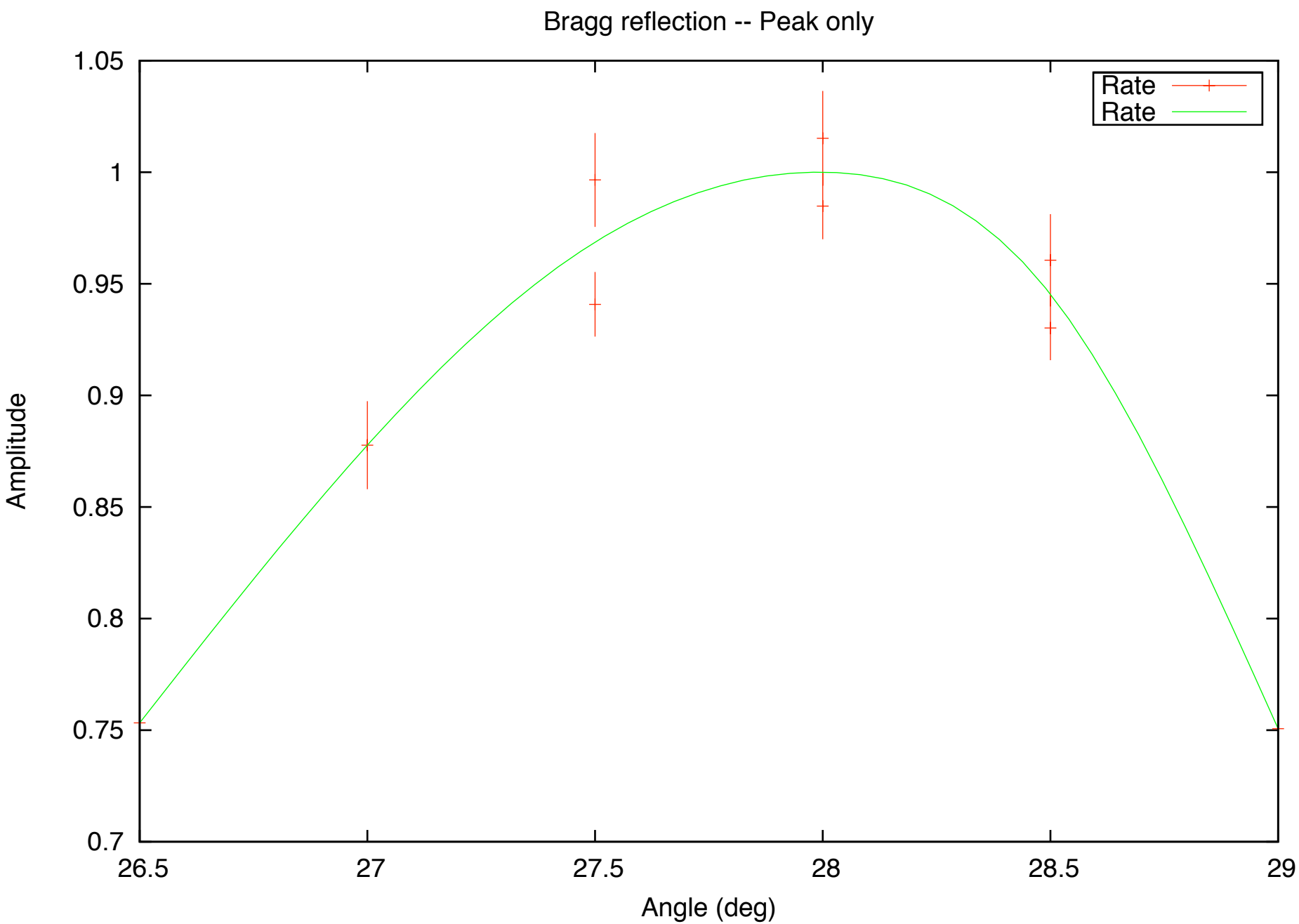

candlesticks with specified boxwidth
'candlesticks.dat' using 1:3:2:6:5
10
8
6
4
2
0
0
2
4
6
8
10

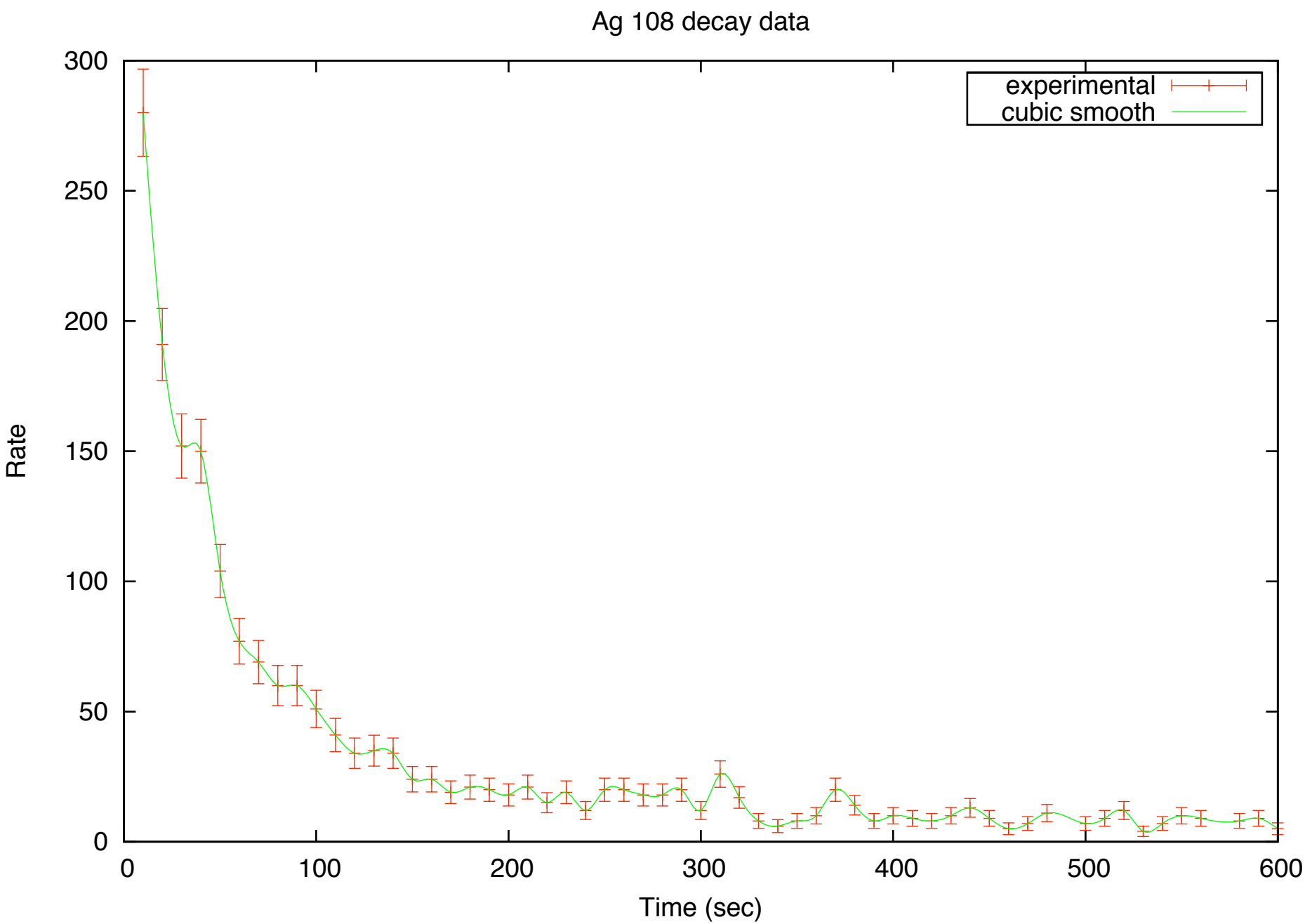

candlesticks with style fill solid
'candlesticks.dat' using 1:3:2:6:5
10
8
6
4
2
0
0
2
4
6
8
10

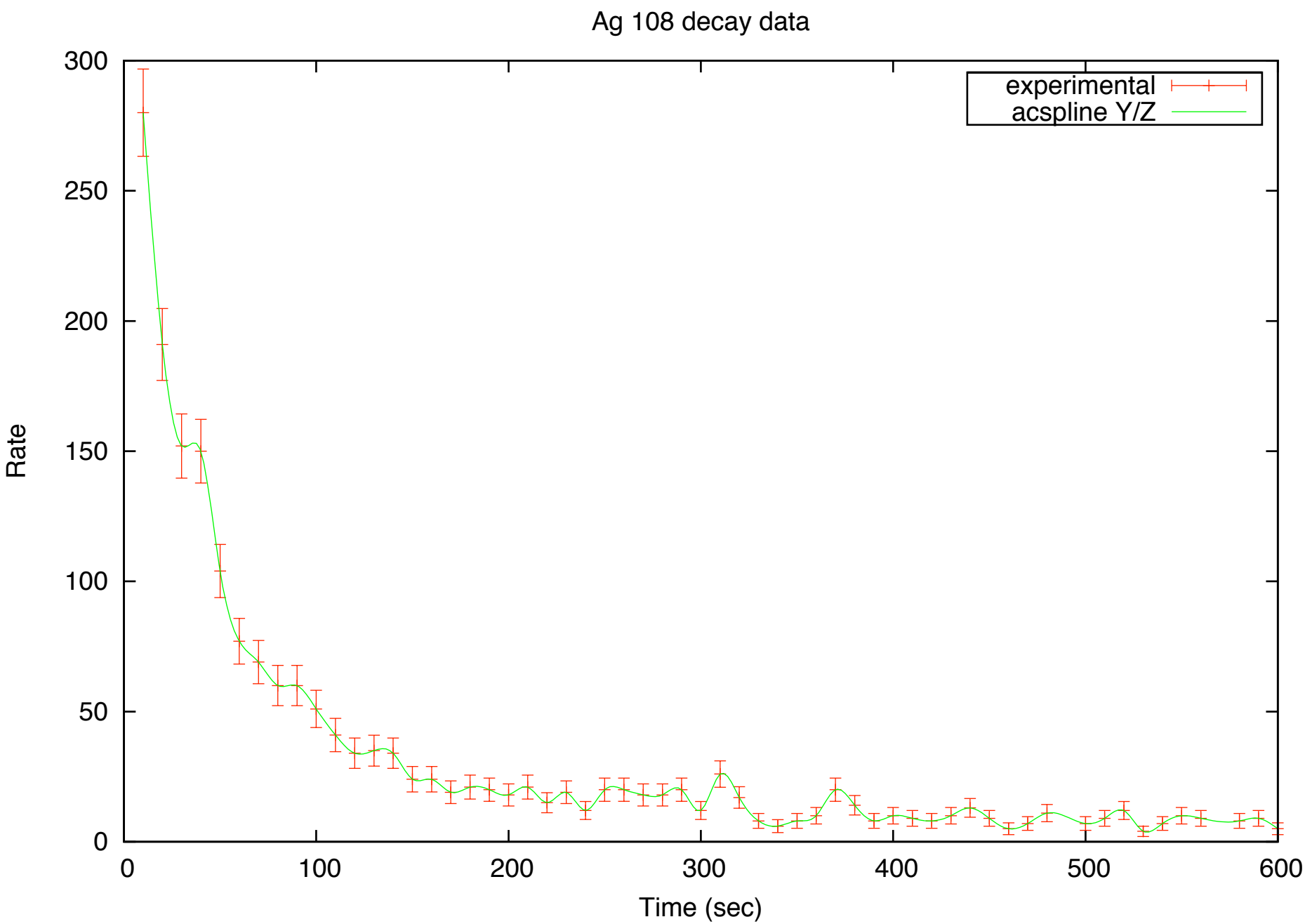

box-and-whisker plot adding median value as bar
Quartiles
10
8
6
4
2
0
0
2
4
6
8
10

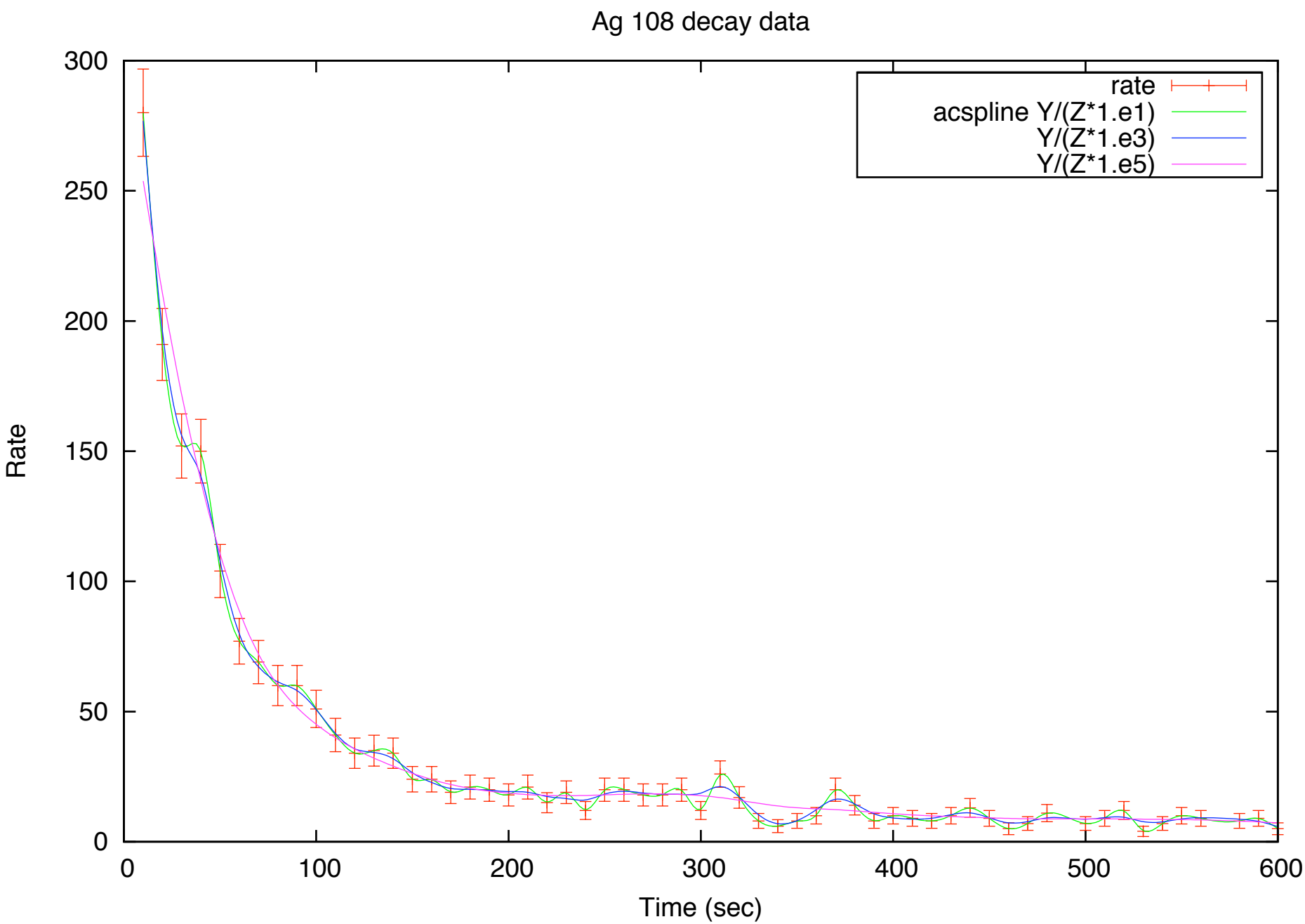

box-and-whisker with median bar and whiskerbars
Quartiles
10
8
6
4
2
0
0
2
4
6
8
10

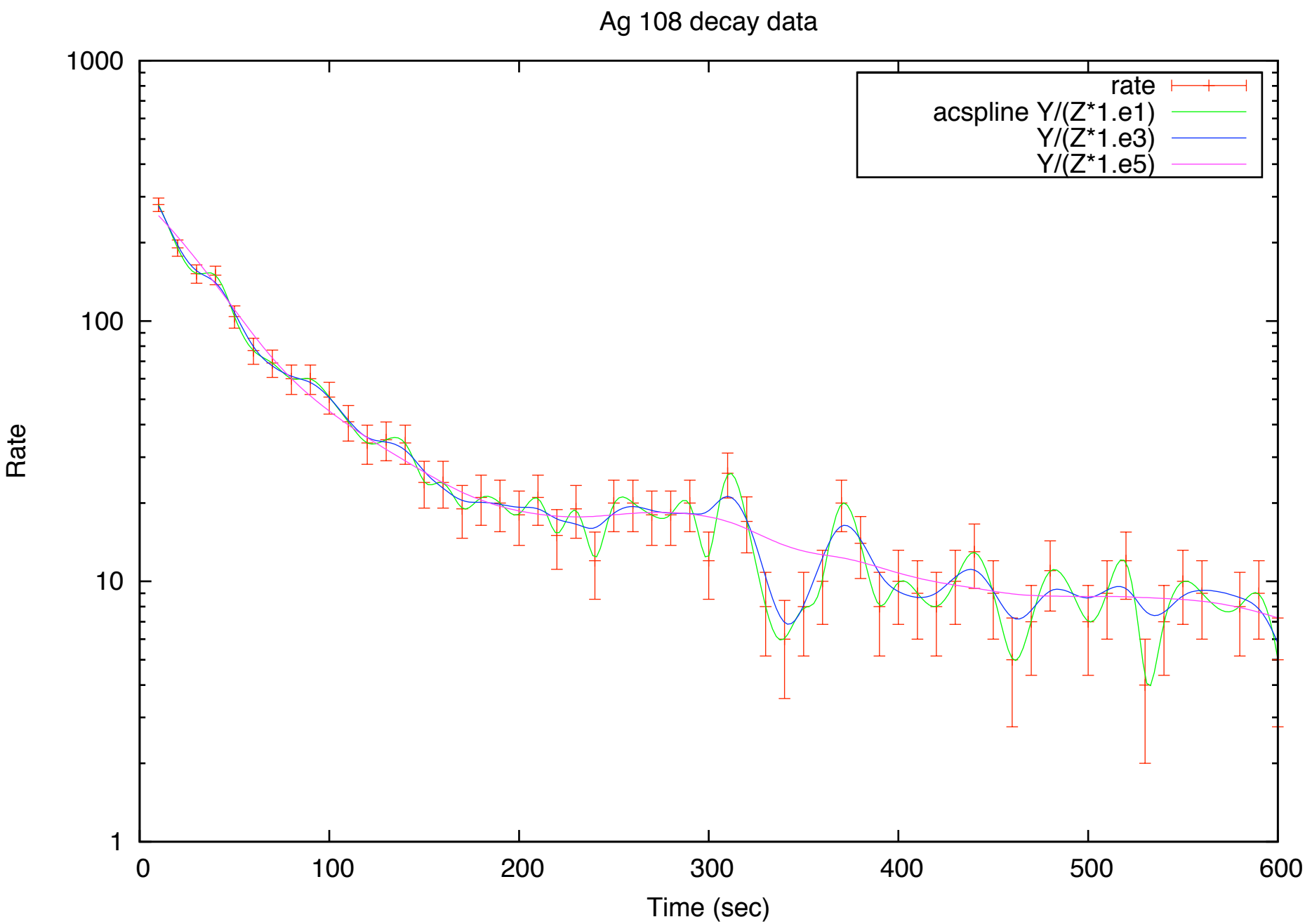

Bragg reflection -- Peak only
Rate
Rate
Amplitude
Angle (deg)
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
26.5
27
27.5
28
28.5
29

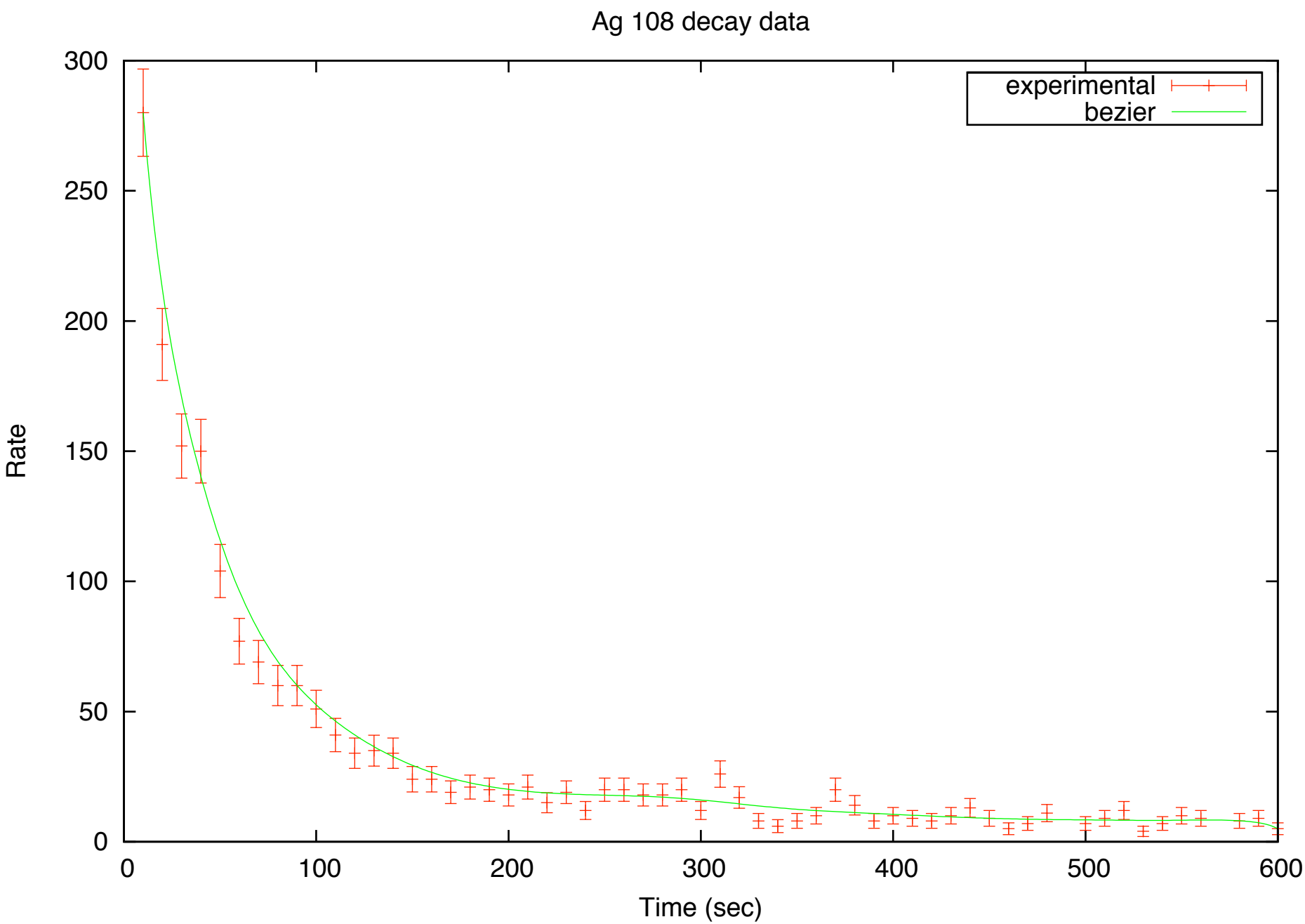

Bragg reflection -- Peak only
Rate
Rate
Amplitude
Angle (deg)
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
26.5
27
27.5
28
28.5
29

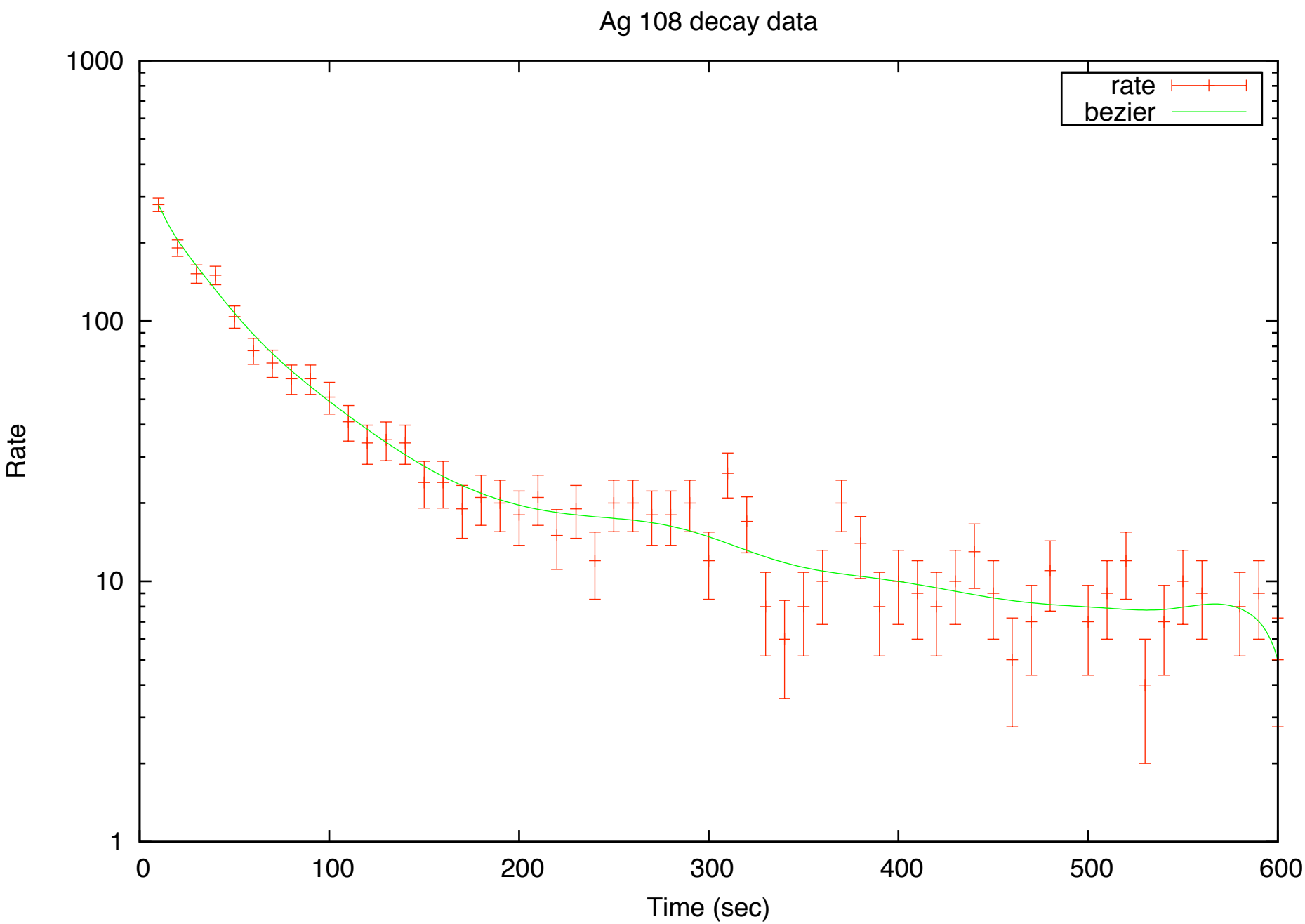

Ag 108 decay data
experimental
cubic smooth
Rate
Time (sec)
0
50
100
150
200
250
300
0
100
200
300
400
500
600

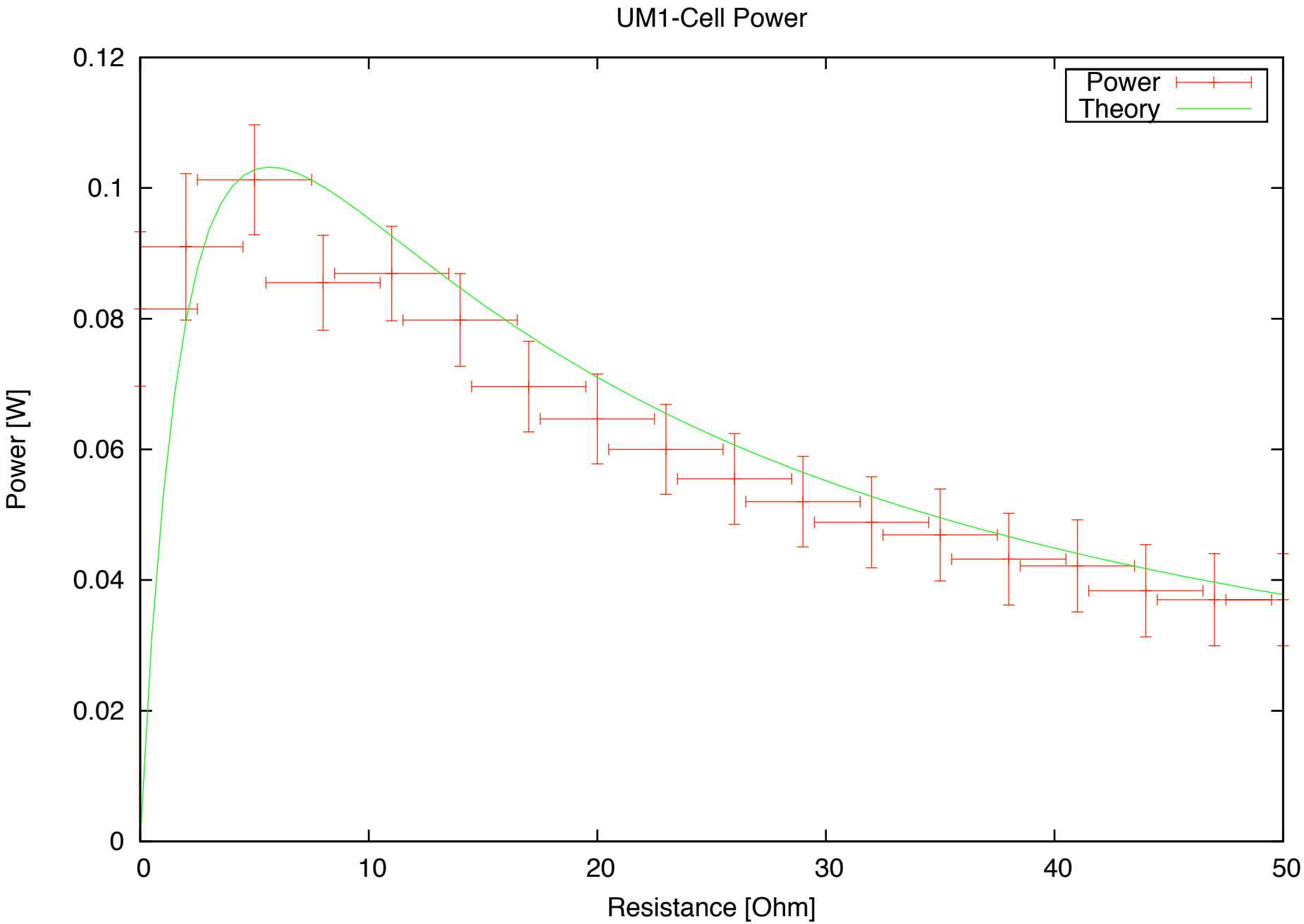

Ag 108 decay data
experimental
acspline Y/Z
Rate
Time (sec)
0
50
100
150
200
250
300
0
100
200
300
400
500
600

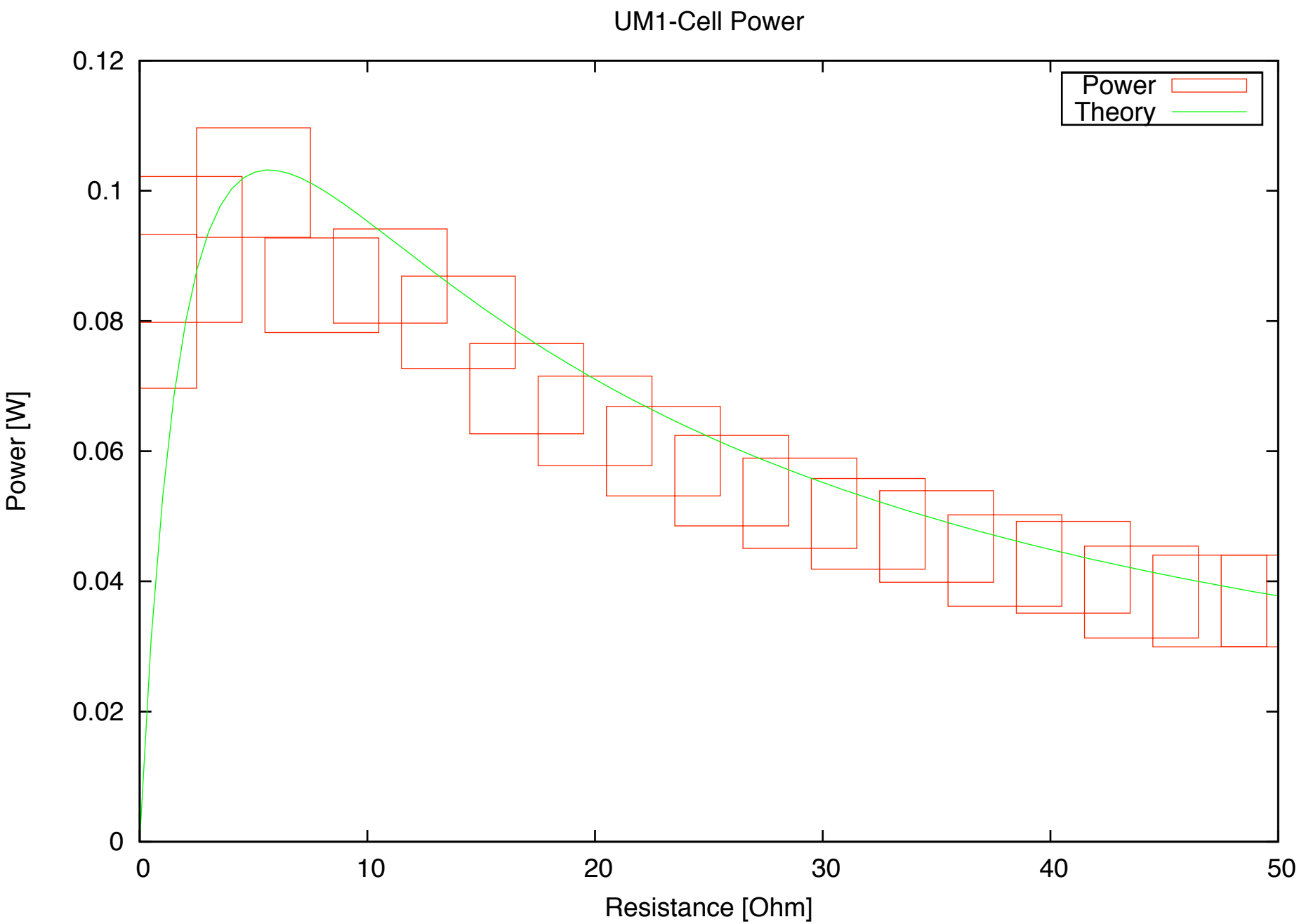

Ag 108 decay data
rate
acspline Y/(Z*1.e1)
Y/(Z*1.e3)
Y/(Z*1.e5)
Rate
Time (sec)
0
50
100
150
200
250
300
0
100
200
300
400
500
600

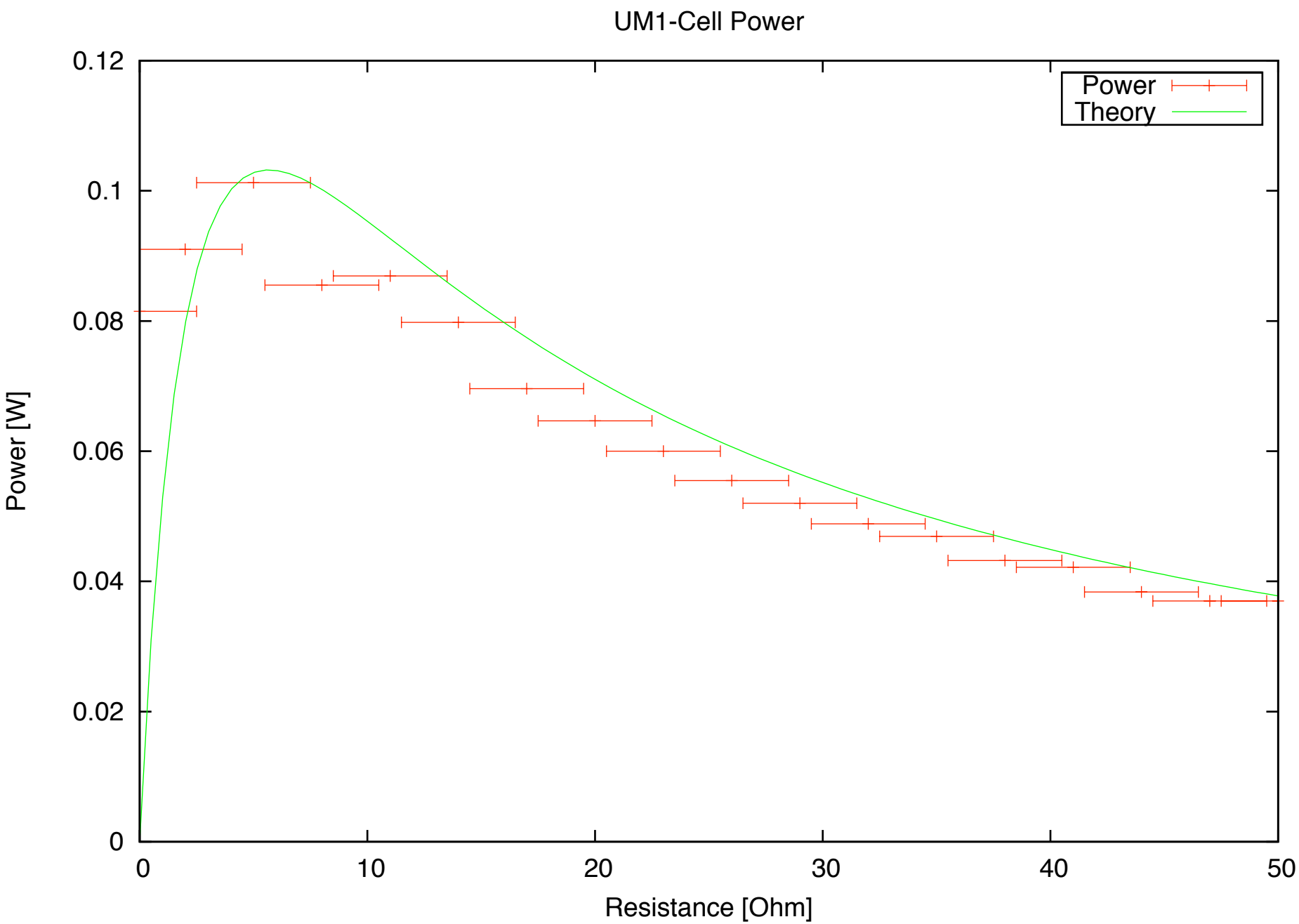

Ag 108 decay data
rate
acspline Y/(Z*1.e1)
Y/(Z*1.e3)
Y/(Z*1.e5)
Rate
Time (sec)
1000
100
10
1
0
100
200
300
400
500
600

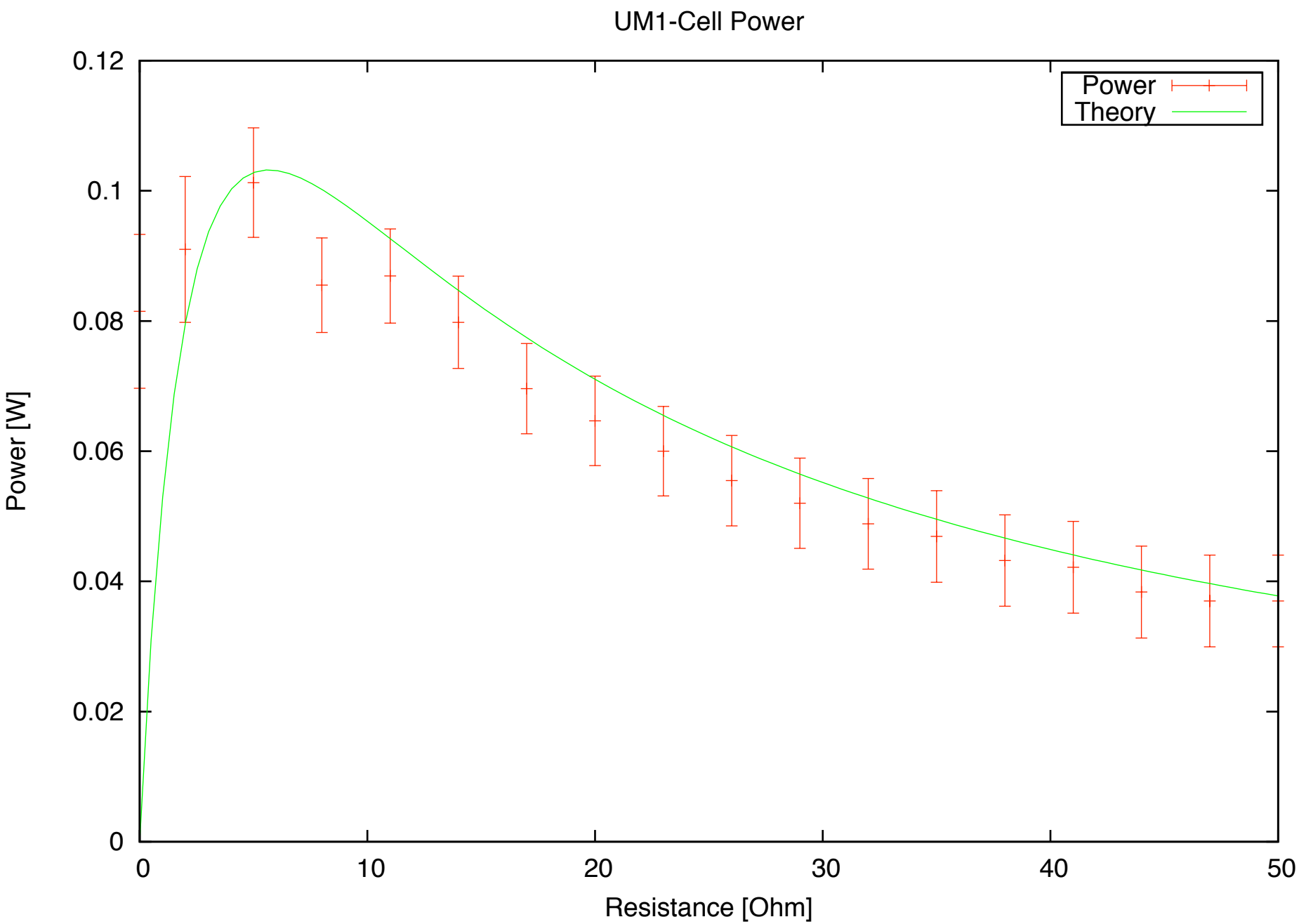

Ag 108 decay data
experimental
bezier
Rate
Time (sec)
0
50
100
150
200
250
300
0
100
200
300
400
500
600

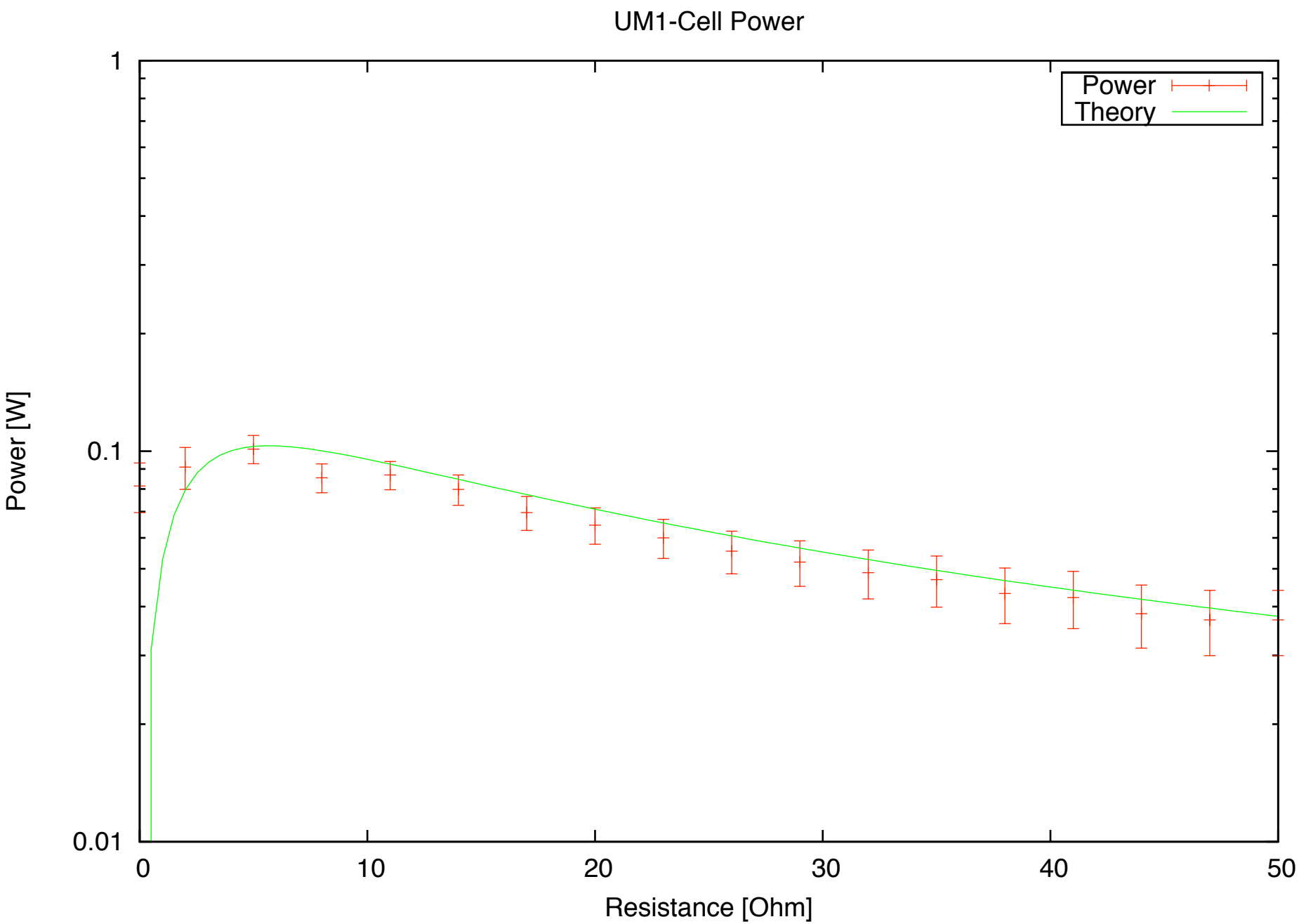

Ag 108 decay data
rate
bezier
1000
100
10
1
Rate
0
100
200
300
400
500
600
Time (sec)

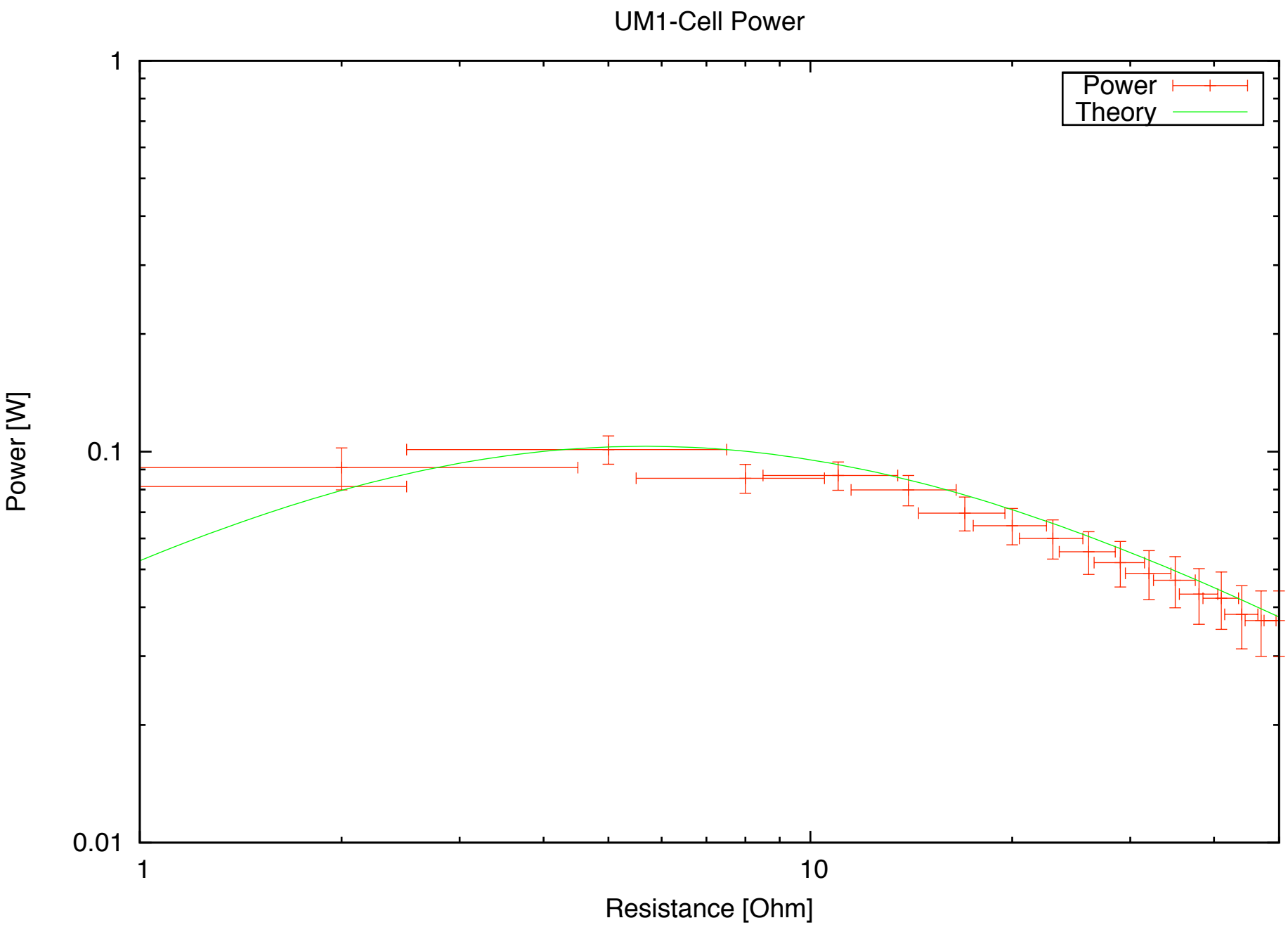

UM1-Cell Power
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

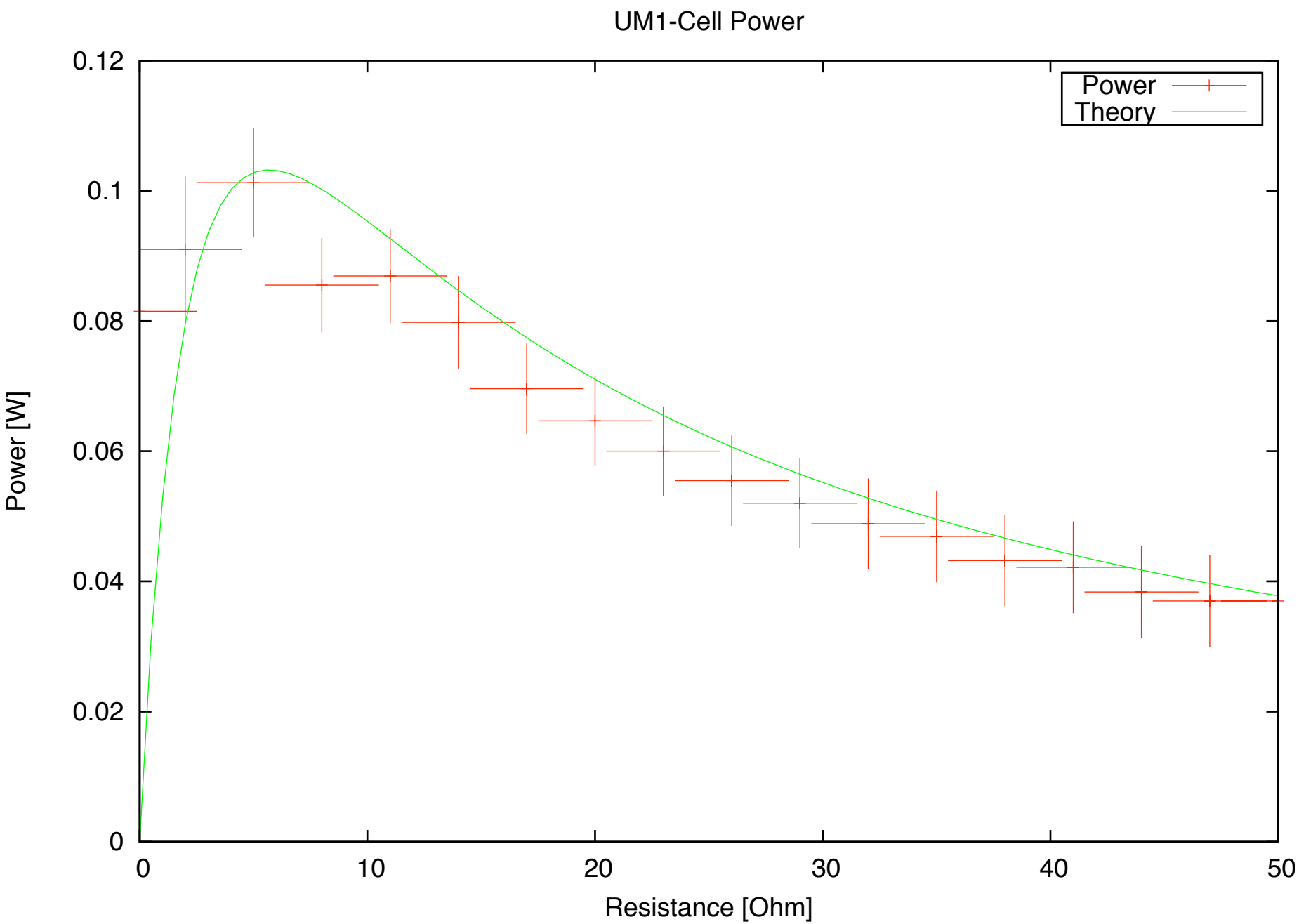

UM1-Cell Power
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

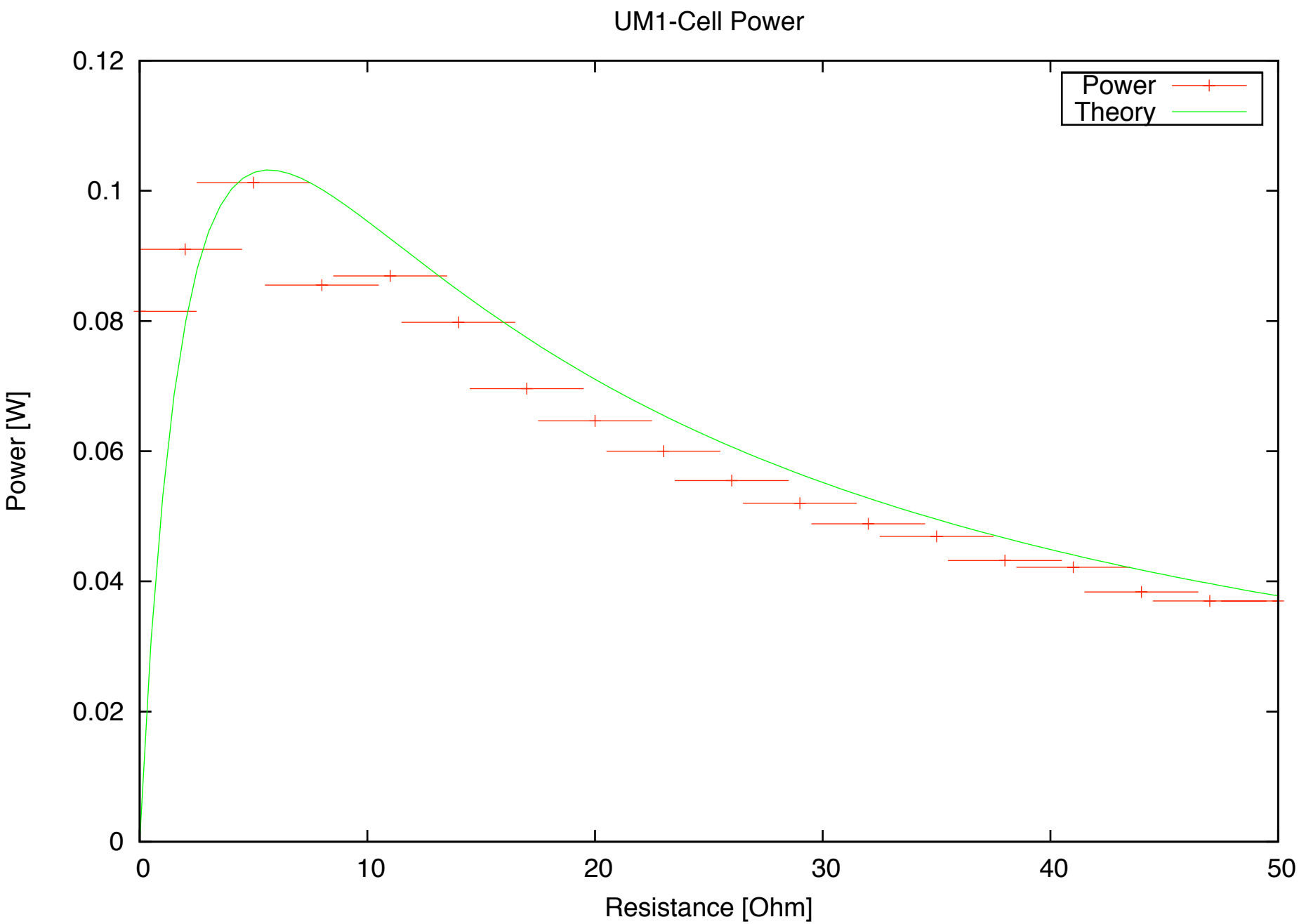

UM1-Cell Power
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

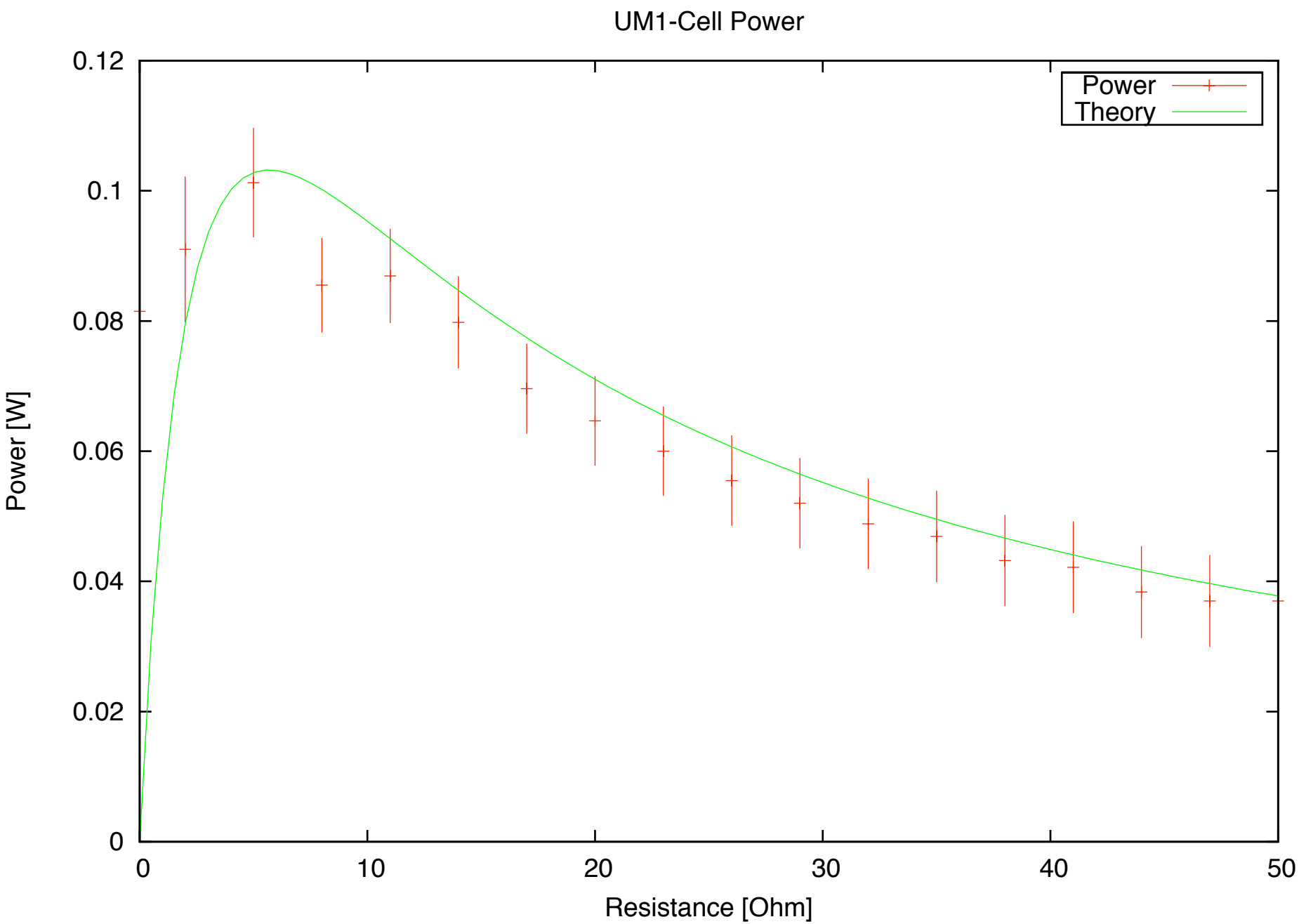

UM1-Cell Power
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

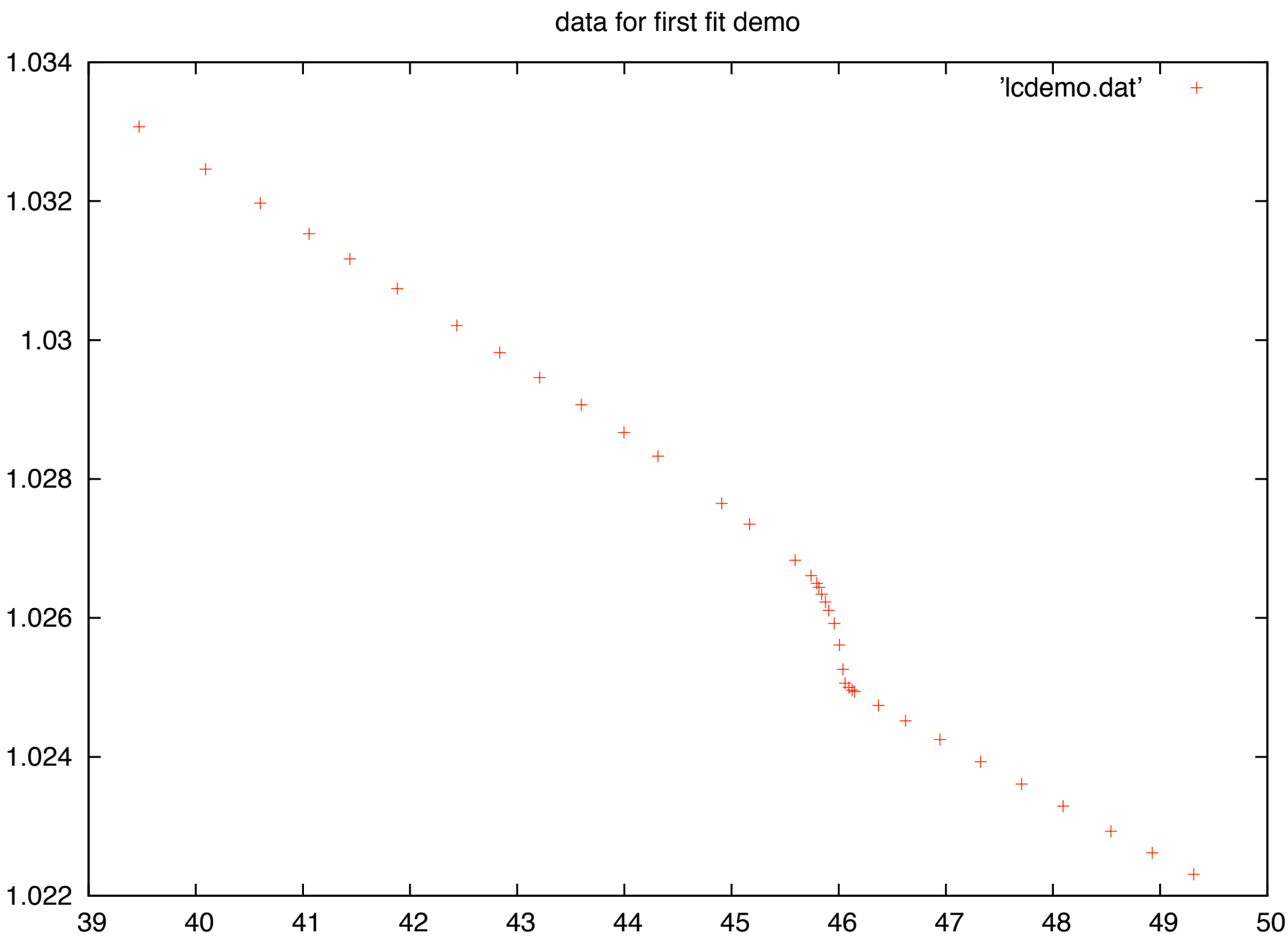

UM1-Cell Power
Power
Theory
Power [W]
1
0.1
0.01
0
10
20
30
40
50
Resistance [Ohm]

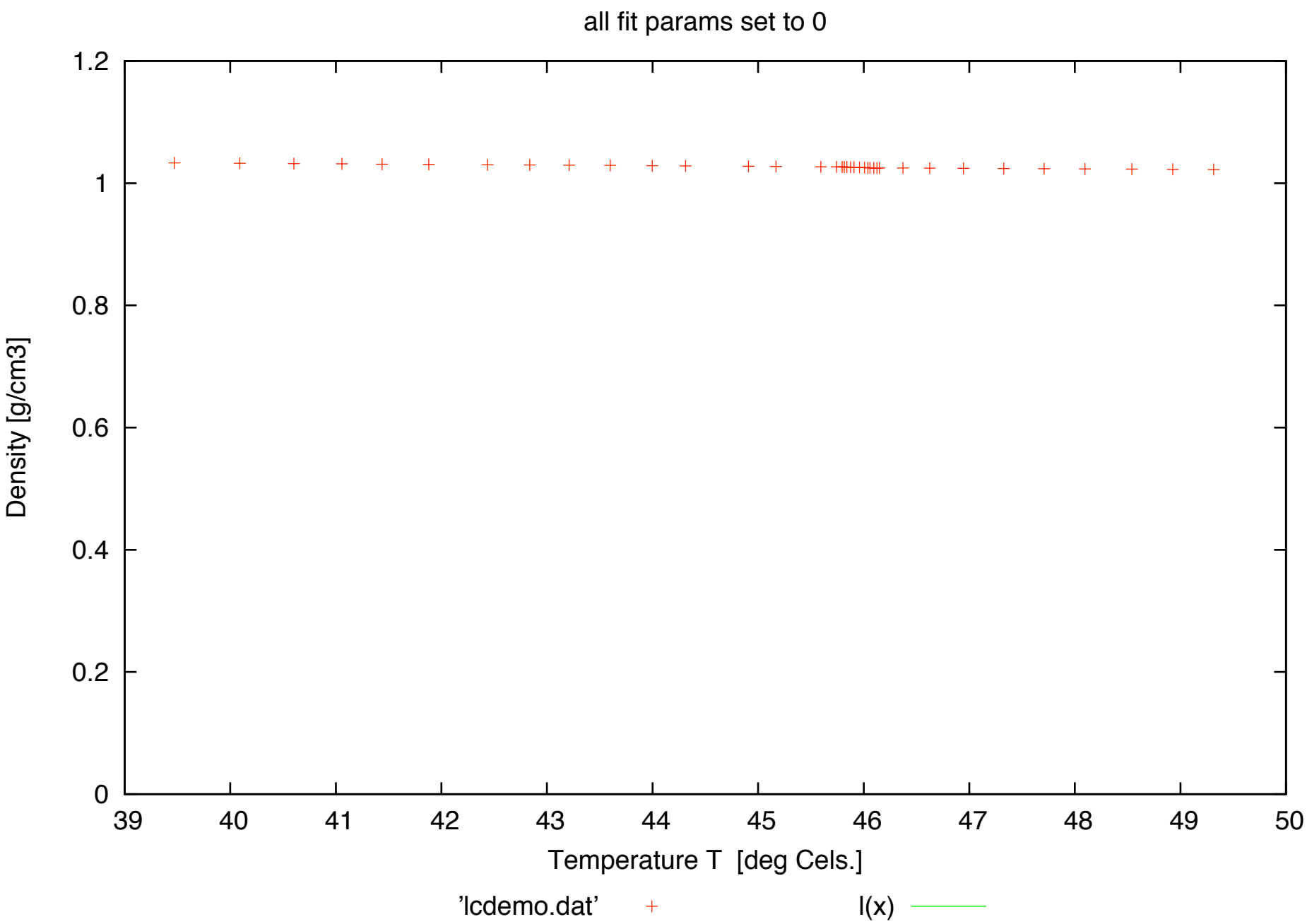

UM1-Cell Power
Power
Theory
Power [W]
1
0.1
0.01
1
10
Resistance [Ohm]

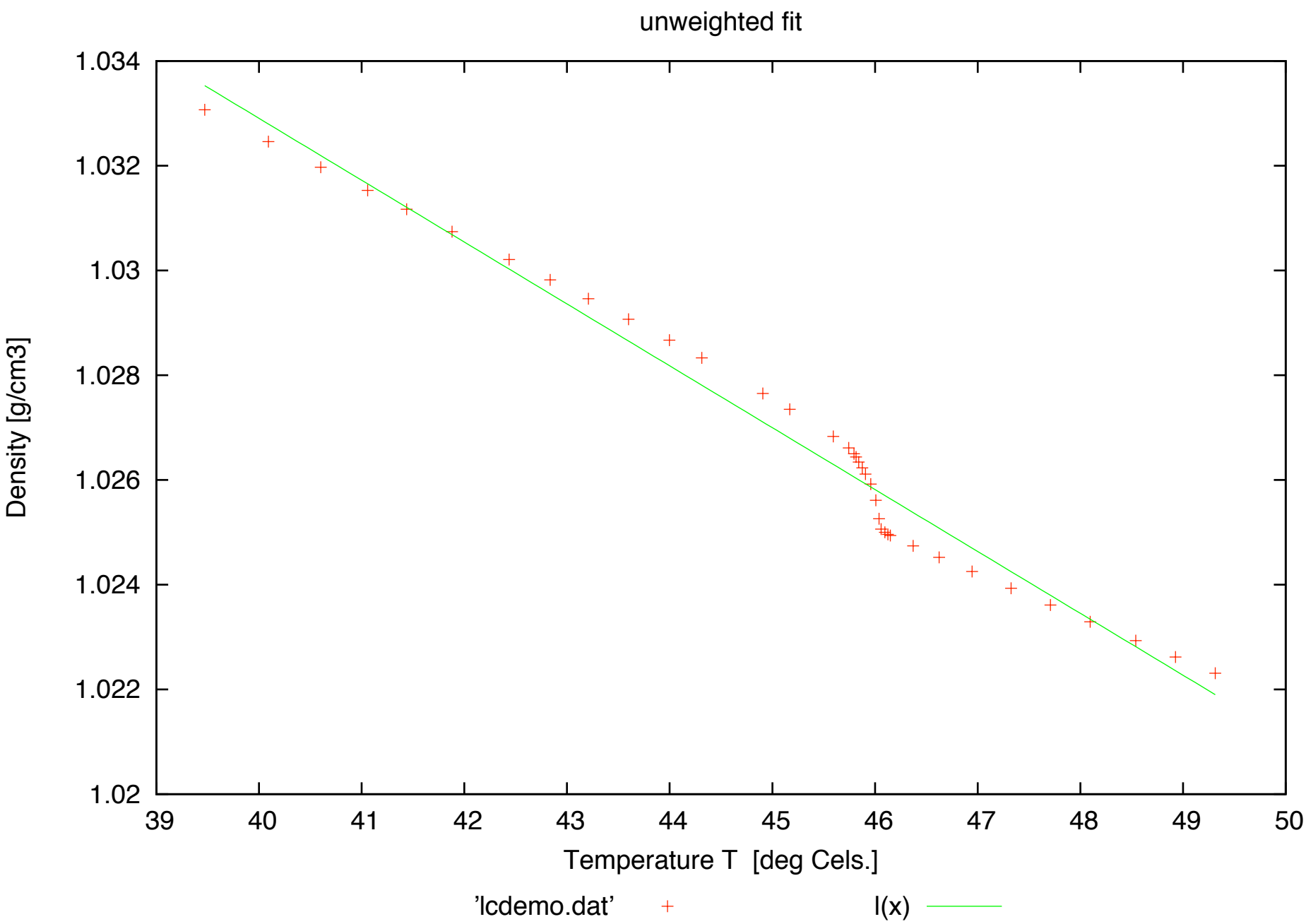

UM1-Cell Power
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

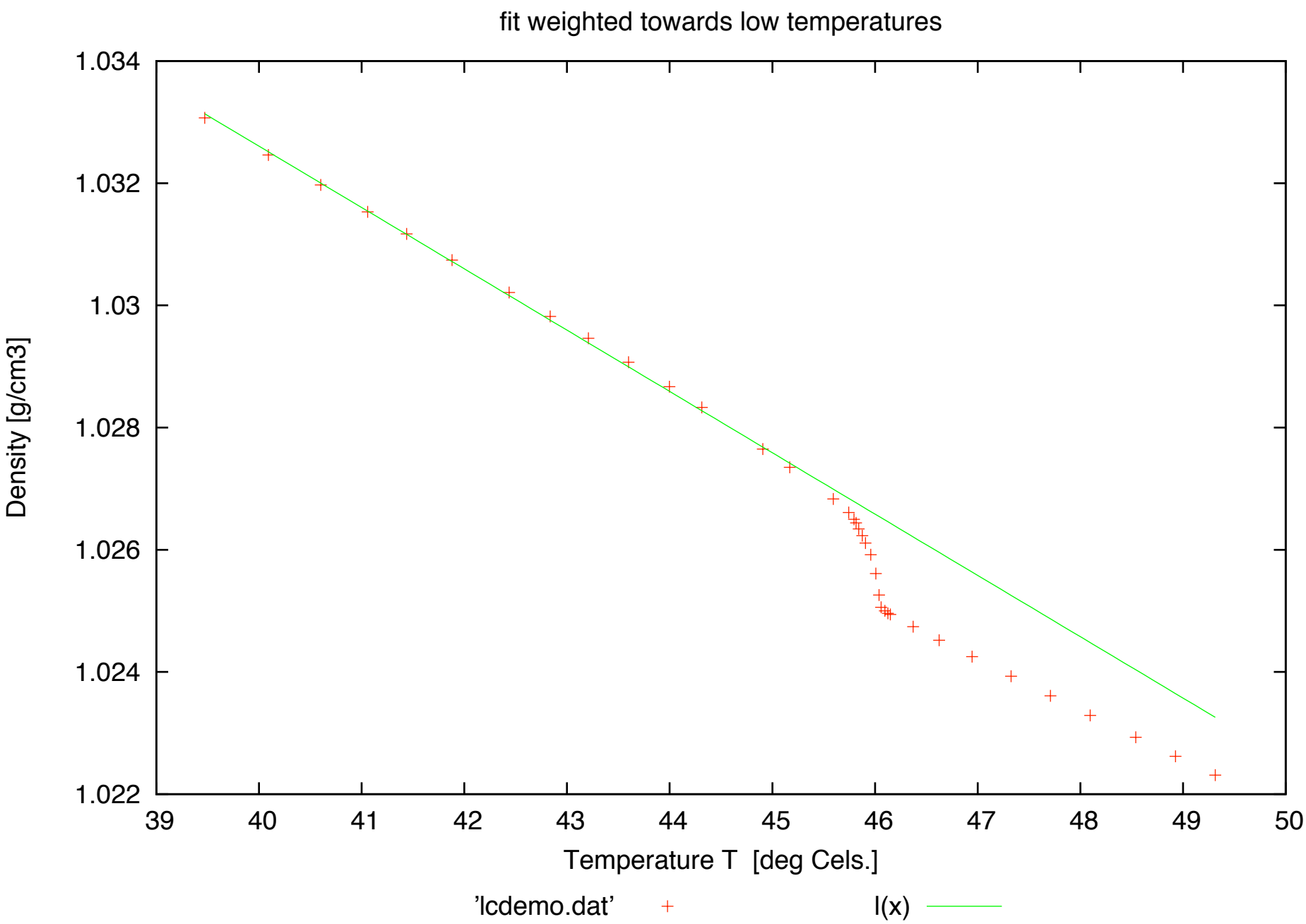

UM1-Cell Power
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

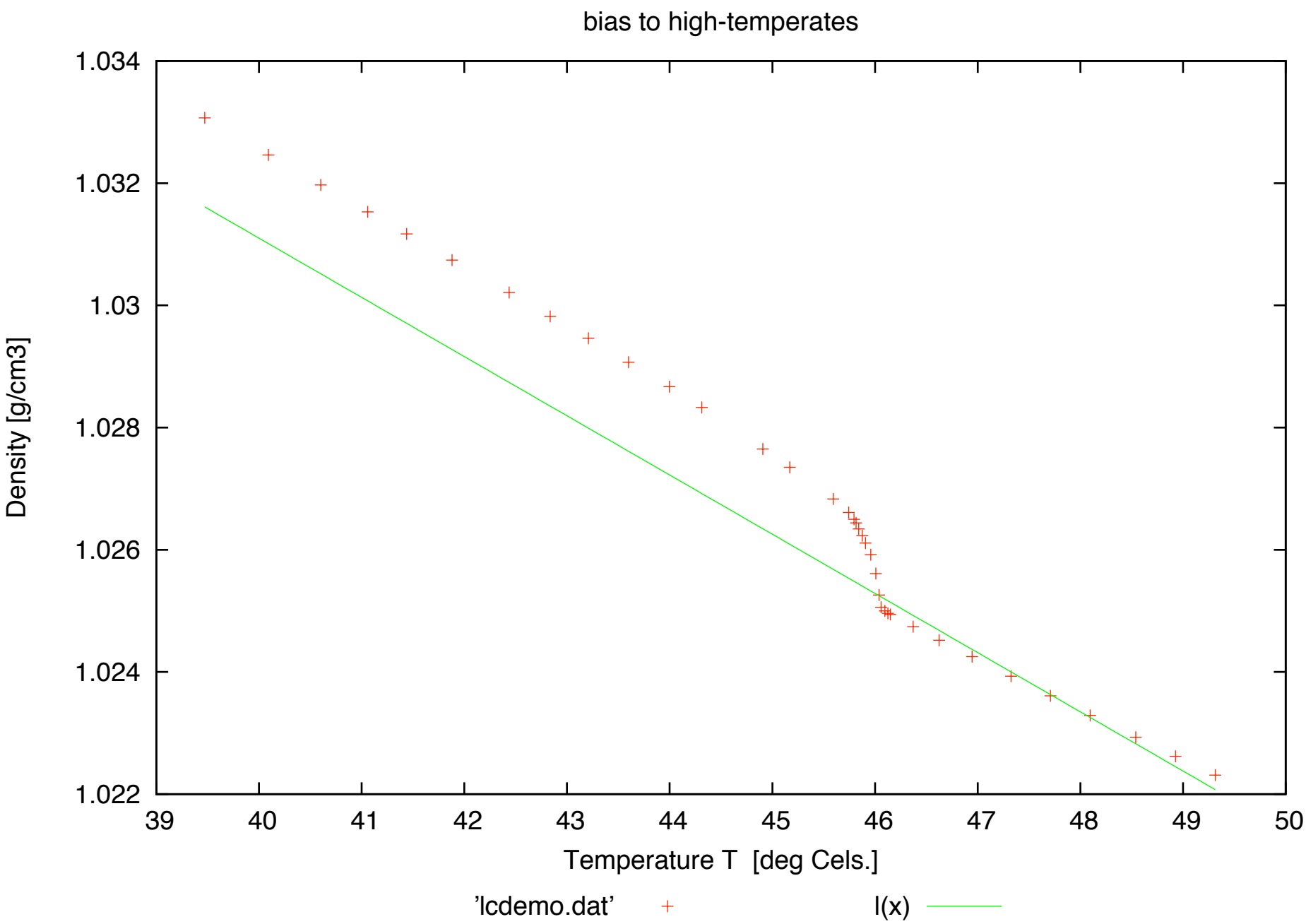

UM1-Cell Power
Power
Theory
Power [W]
Resistance [Ohm]
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50

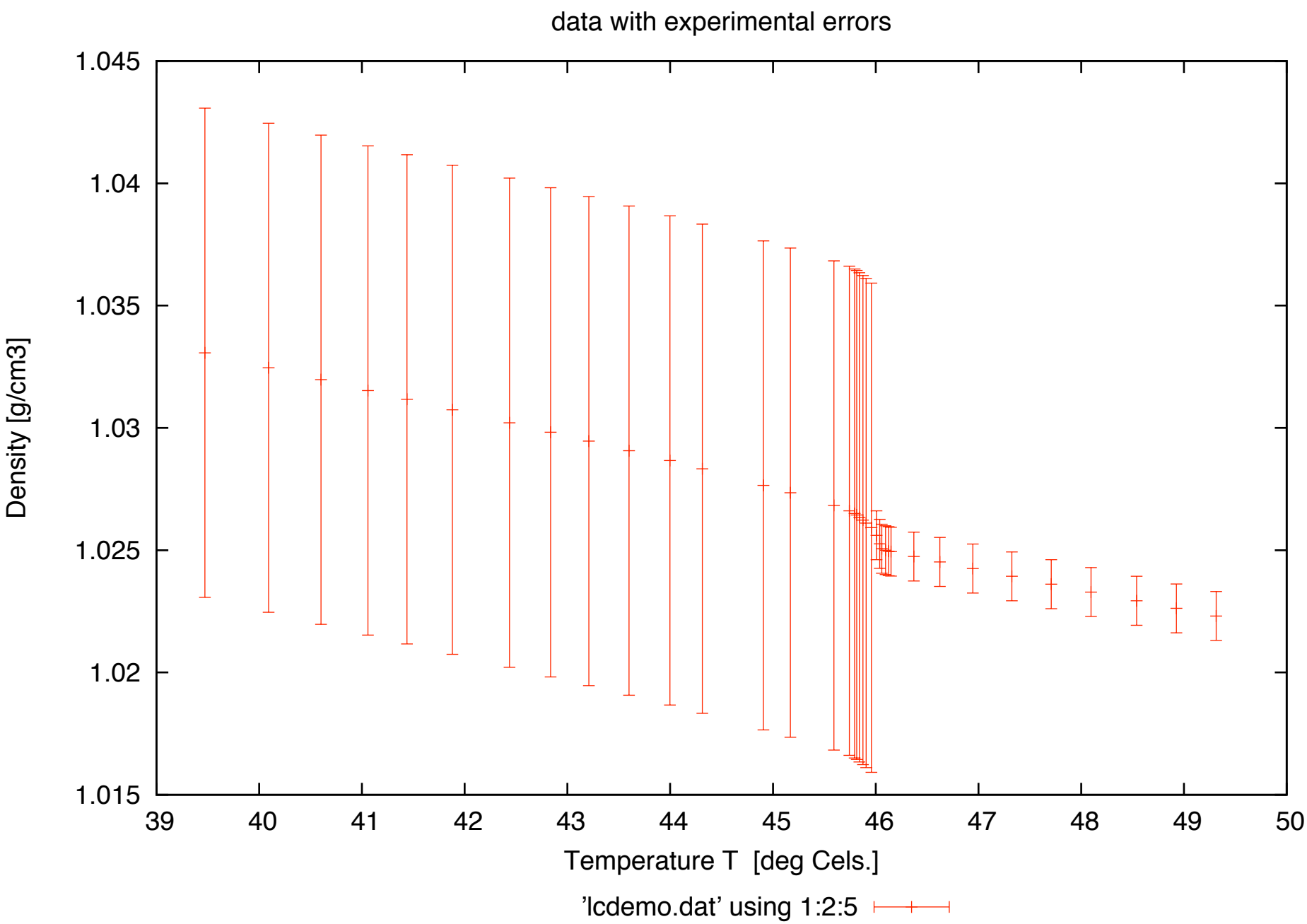

data for first fit demo
'lcdemo.dat'
1.034
1.032
1.03
1.028
1.026
1.024
1.022
39
40
41
42
43
44
45
46
47
48
49
50

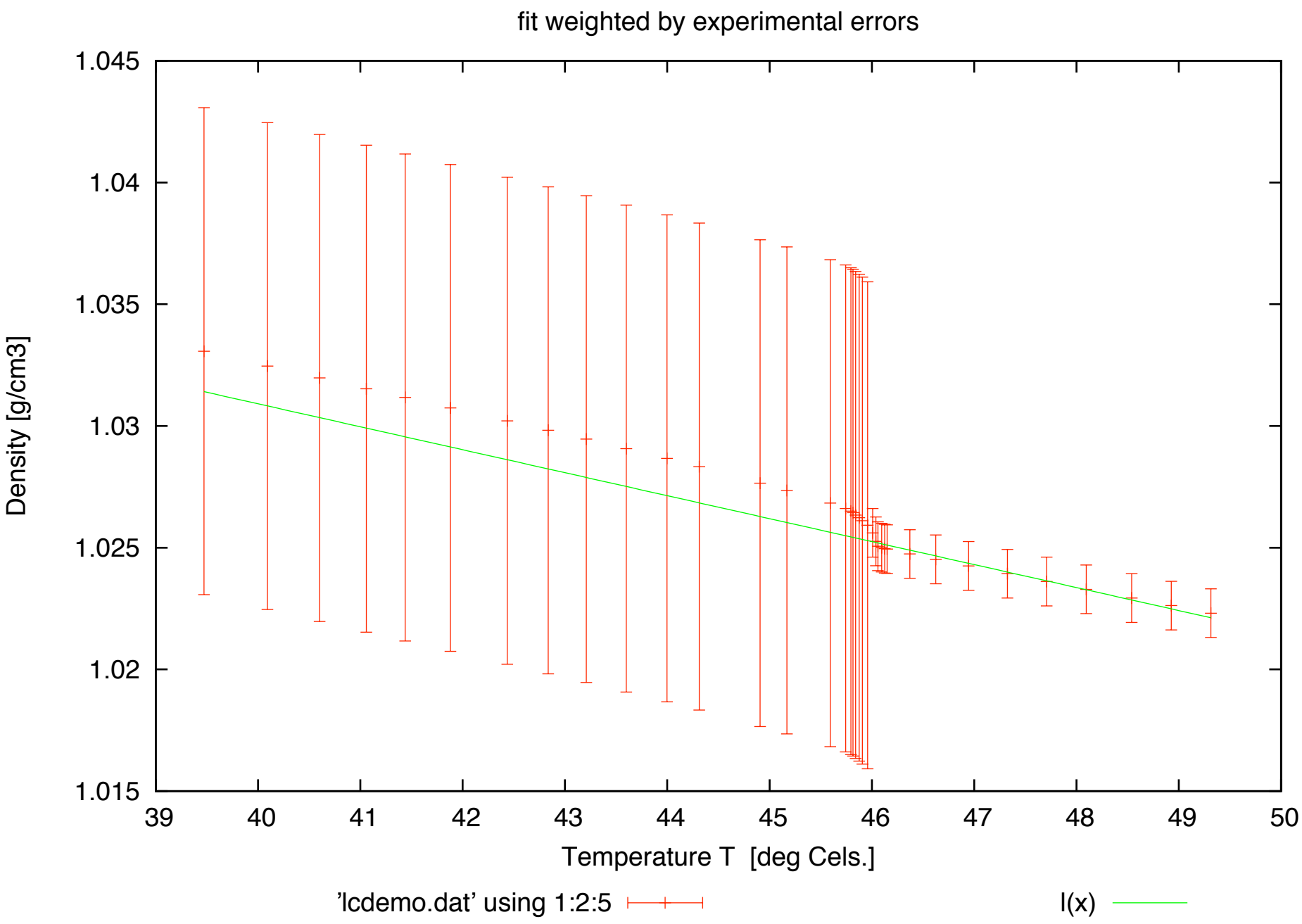

all fit params set to 0
Density [g/cm3]
1.2
1
0.8
0.6
0.4
0.2
0
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T [deg Cels.]
'lcdemo.dat'
l(x)

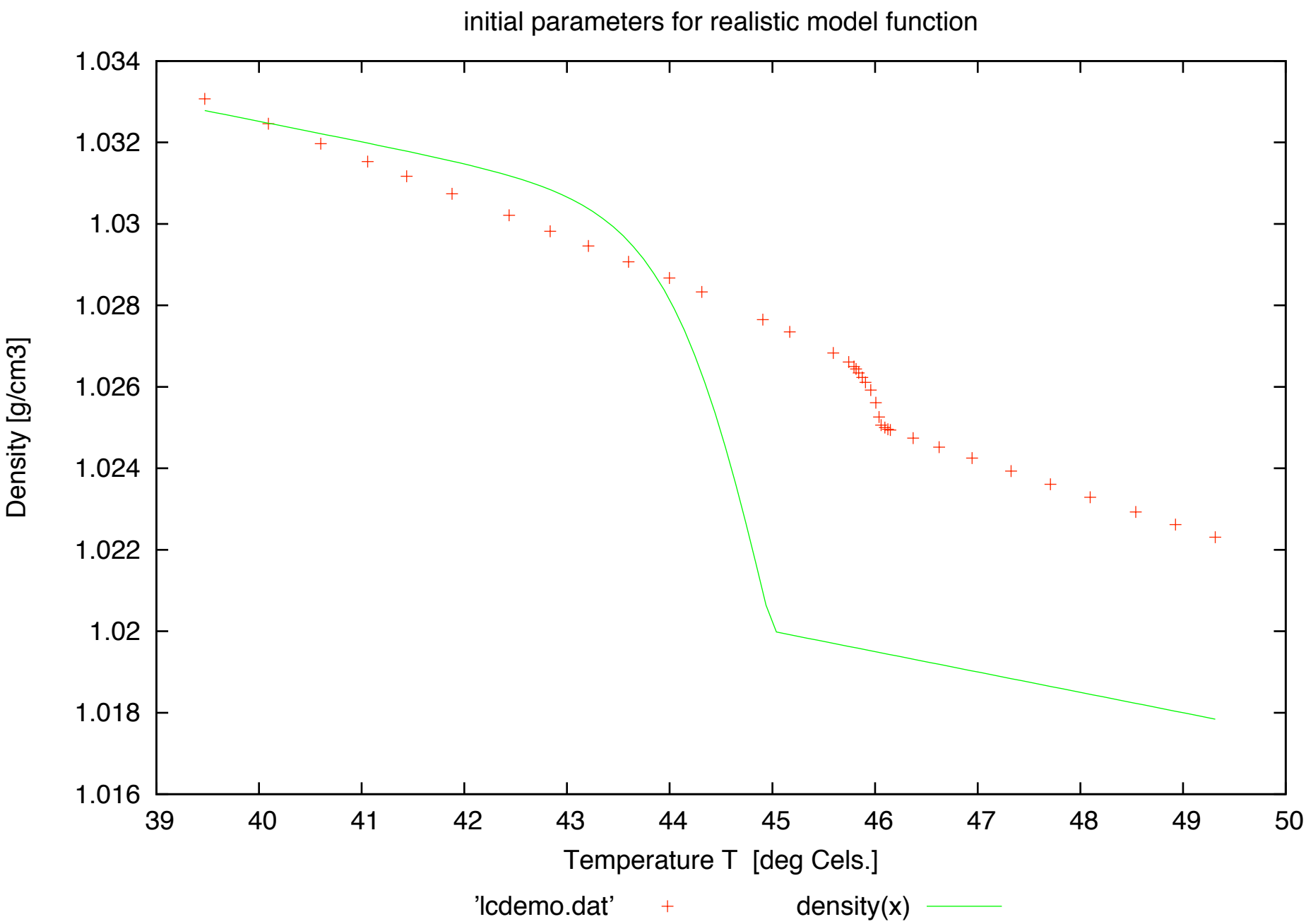

unweighted fit
Density [g/cm3]
1.034
1.032
1.03
1.028
1.026
1.024
1.022
1.02
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T [deg Cels.]
'lcdemo.dat'
f(x)

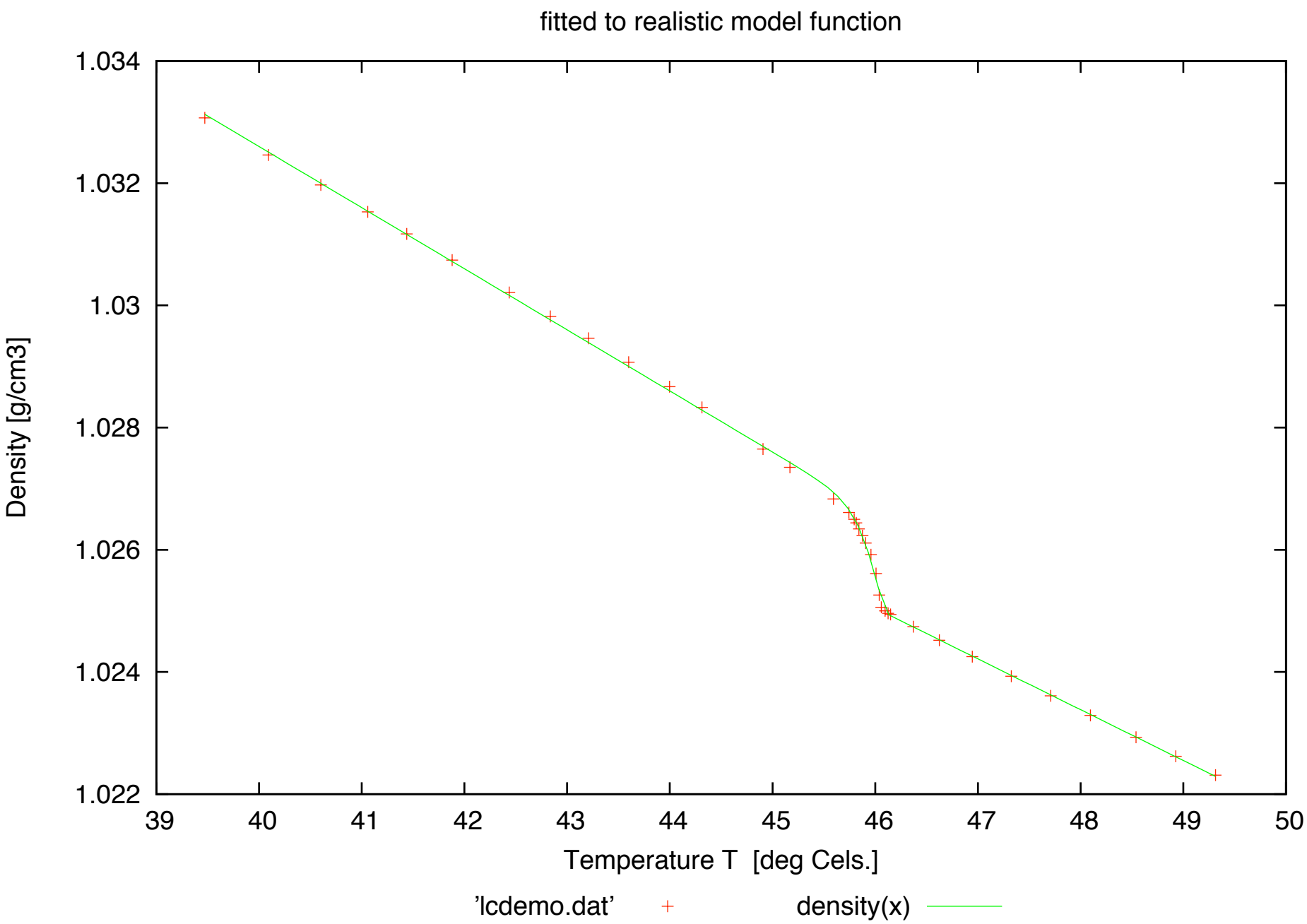

fit weighted towards low temperatures
Density [g/cm3]
1.034
1.032
1.03
1.028
1.026
1.024
1.022
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T [deg Cels.]
'lcdemo.dat'
f(x)

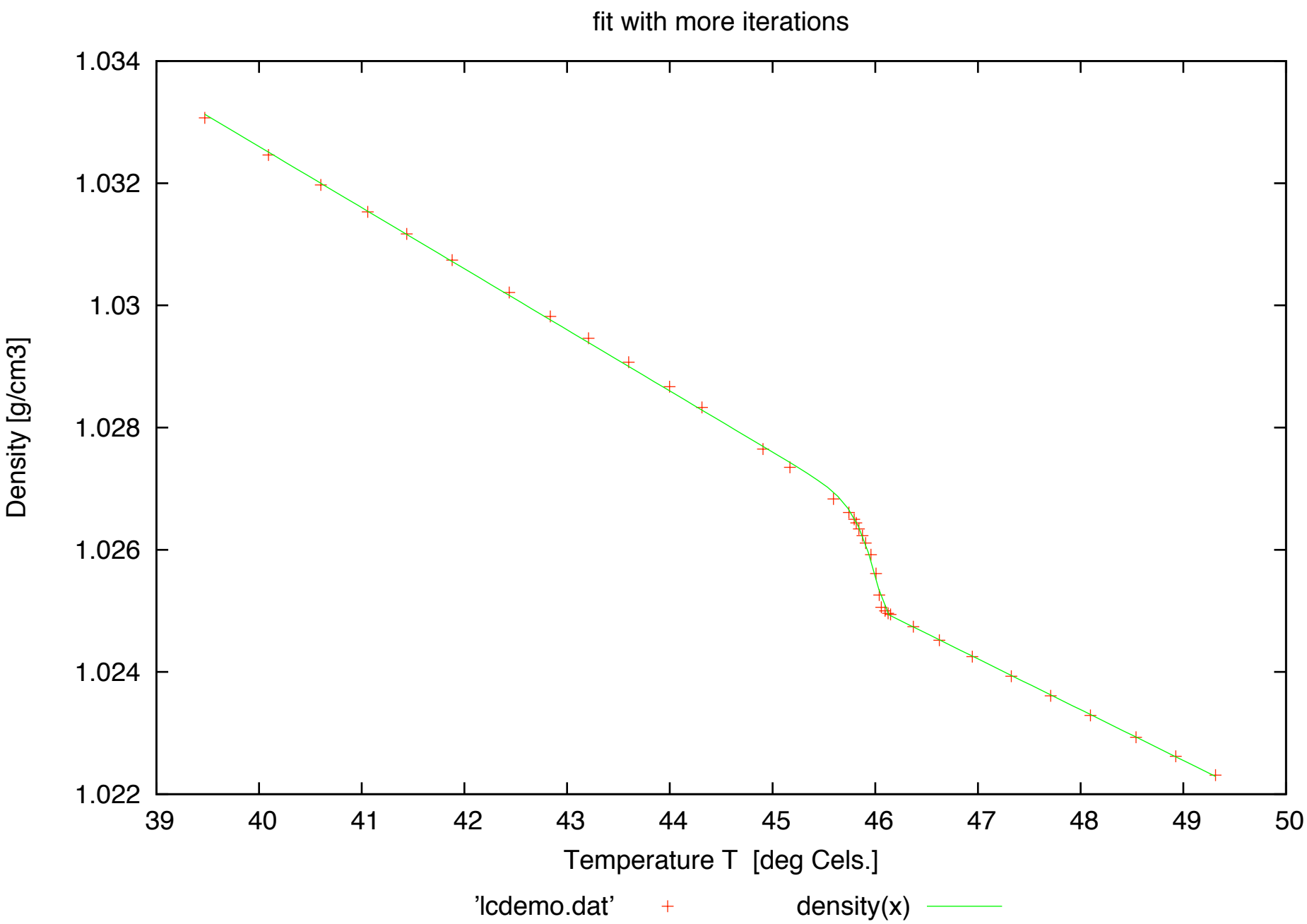

bias to high-temperates
1.034
1.032
1.03
1.028
1.026
1.024
1.022
Density [g/cm3]
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T [deg Cels.]
'lcdemo.dat'
l(x)


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

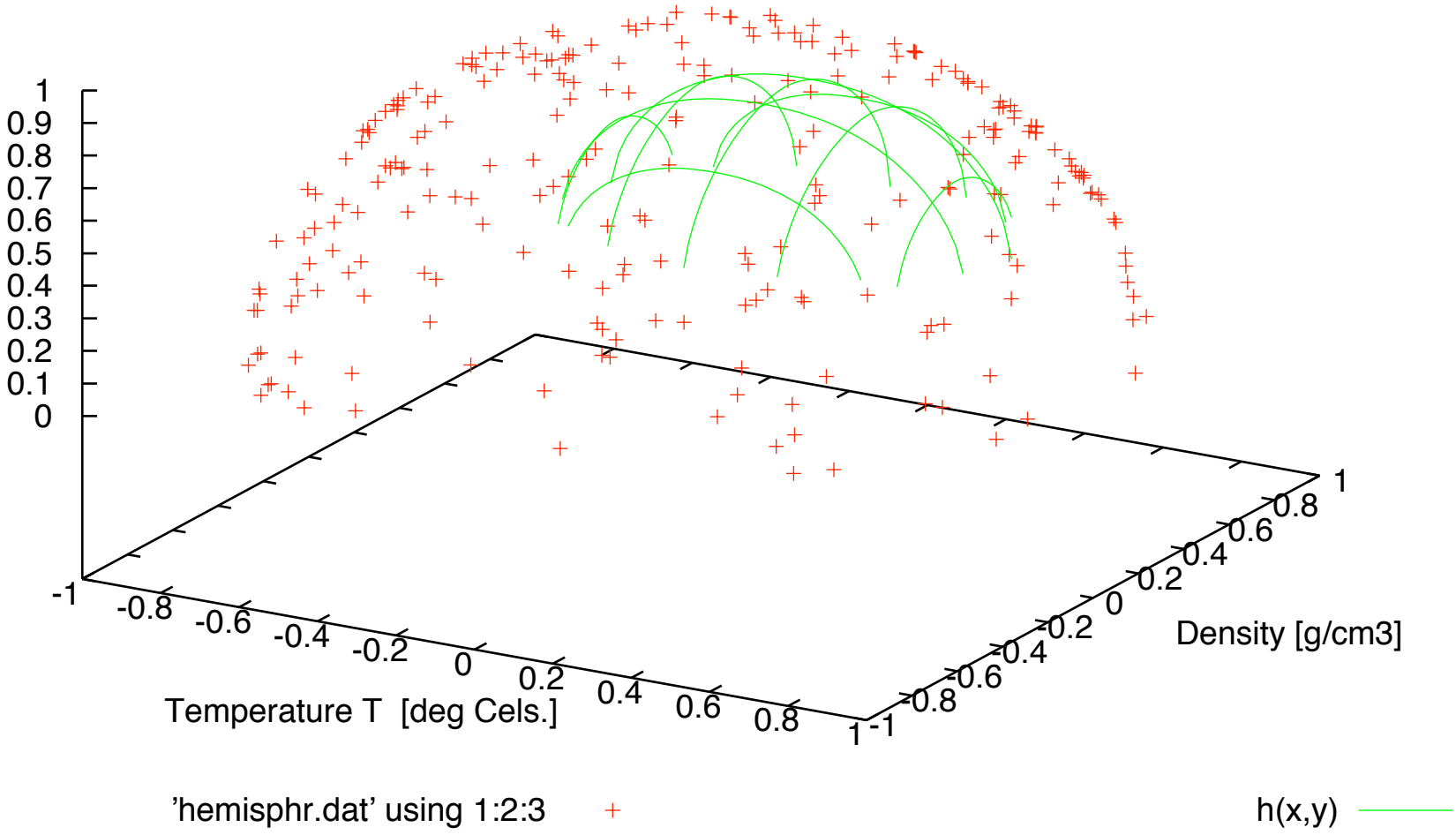

fit weighted by experimental errors
Density [g/cm3]
1.045
1.04
1.035
1.03
1.025
1.02
1.015
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T [deg Cels.]
'lcdemo.dat' using 1:2:5
l(x)

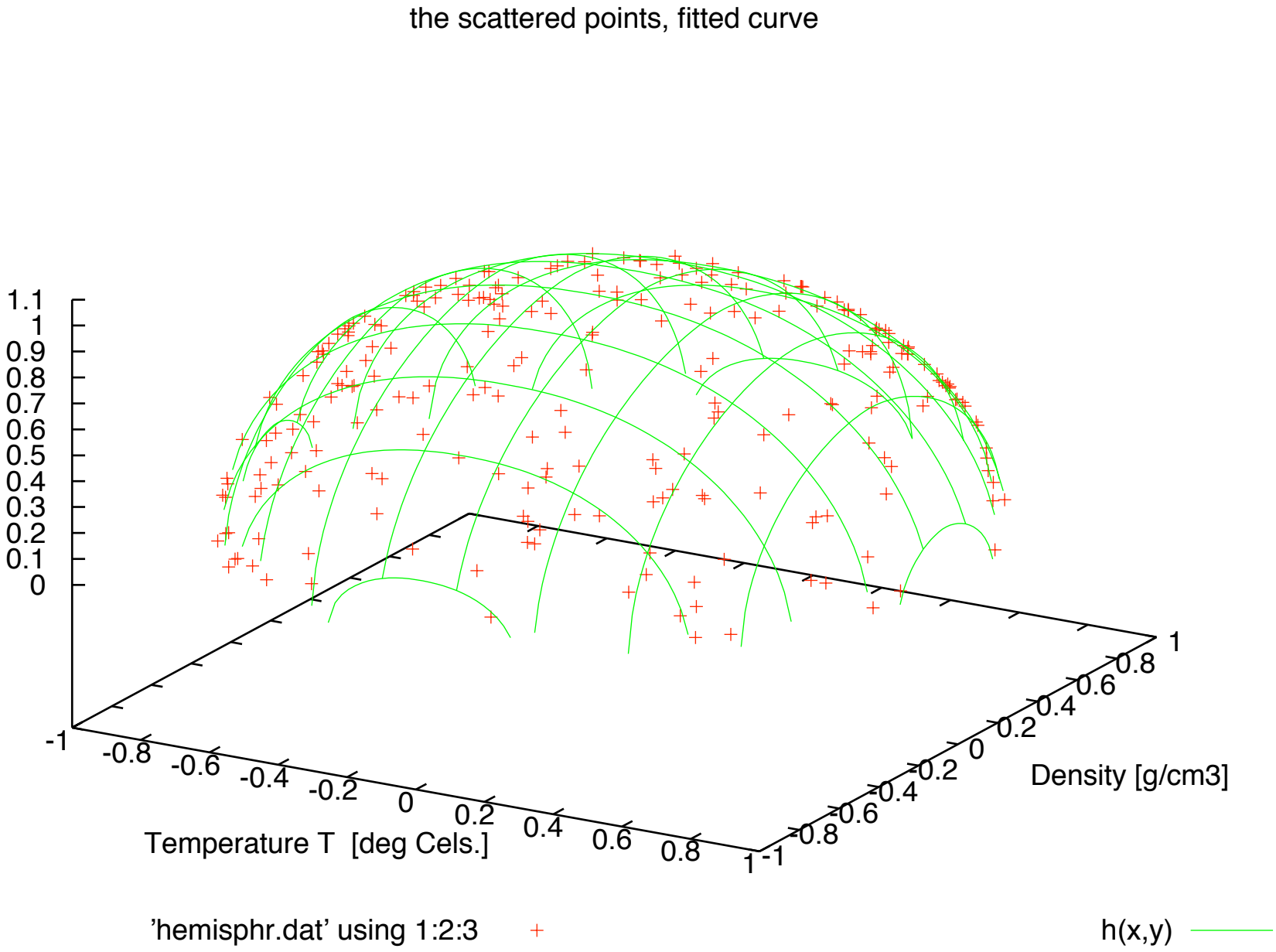

initial parameters for realistic model function
Density [g/cm3]
1.034
1.032
1.03
1.028
1.026
1.024
1.022
1.02
1.018
1.016
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T [deg Cels.]
'lcdemo.dat'
density(x)

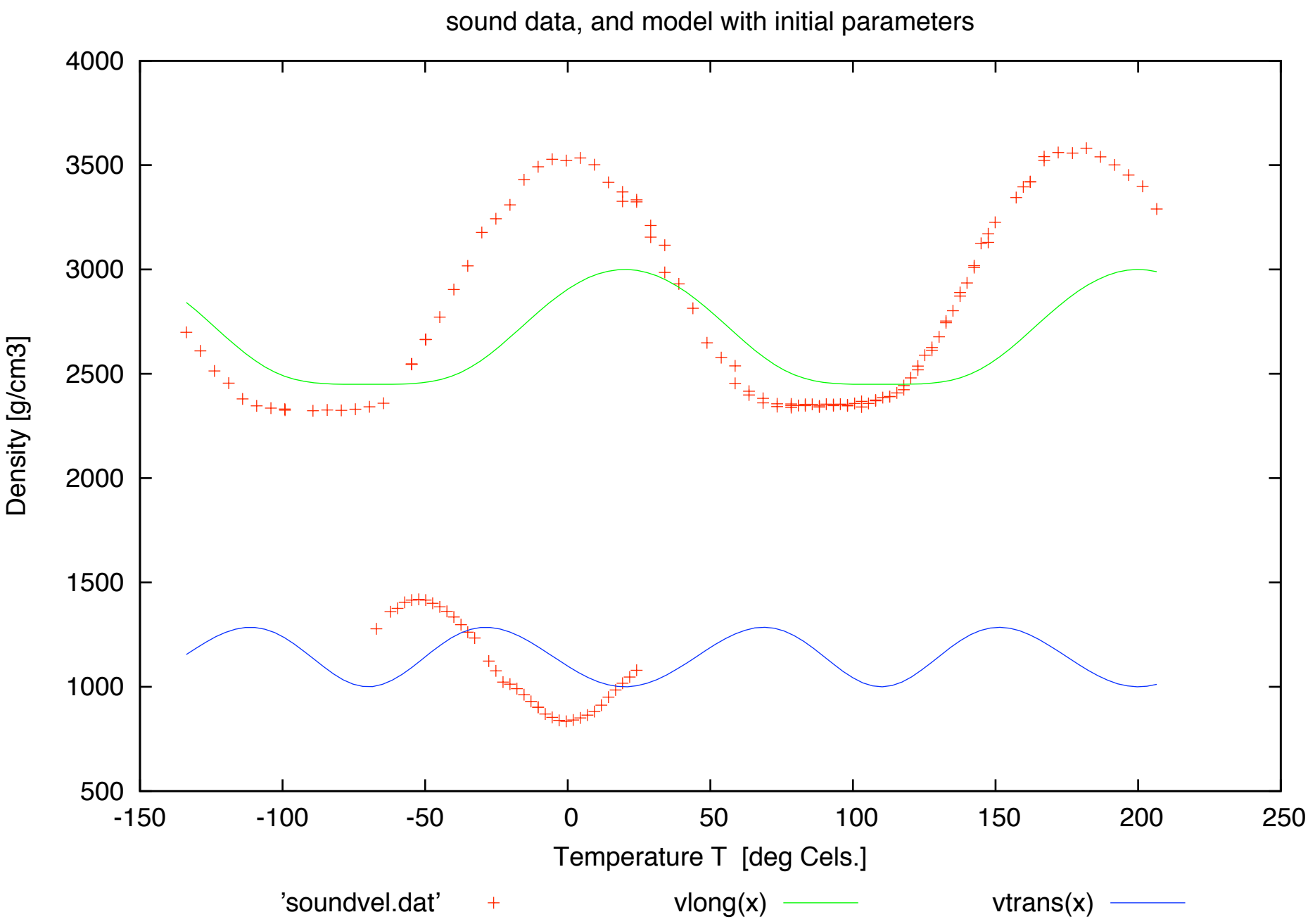

fitted to realistic model function
1.034
1.032
1.03
1.028
1.026
1.024
1.022
Density [g/cm3]
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T [deg Cels.]
'lcdemo.dat'
density(x)

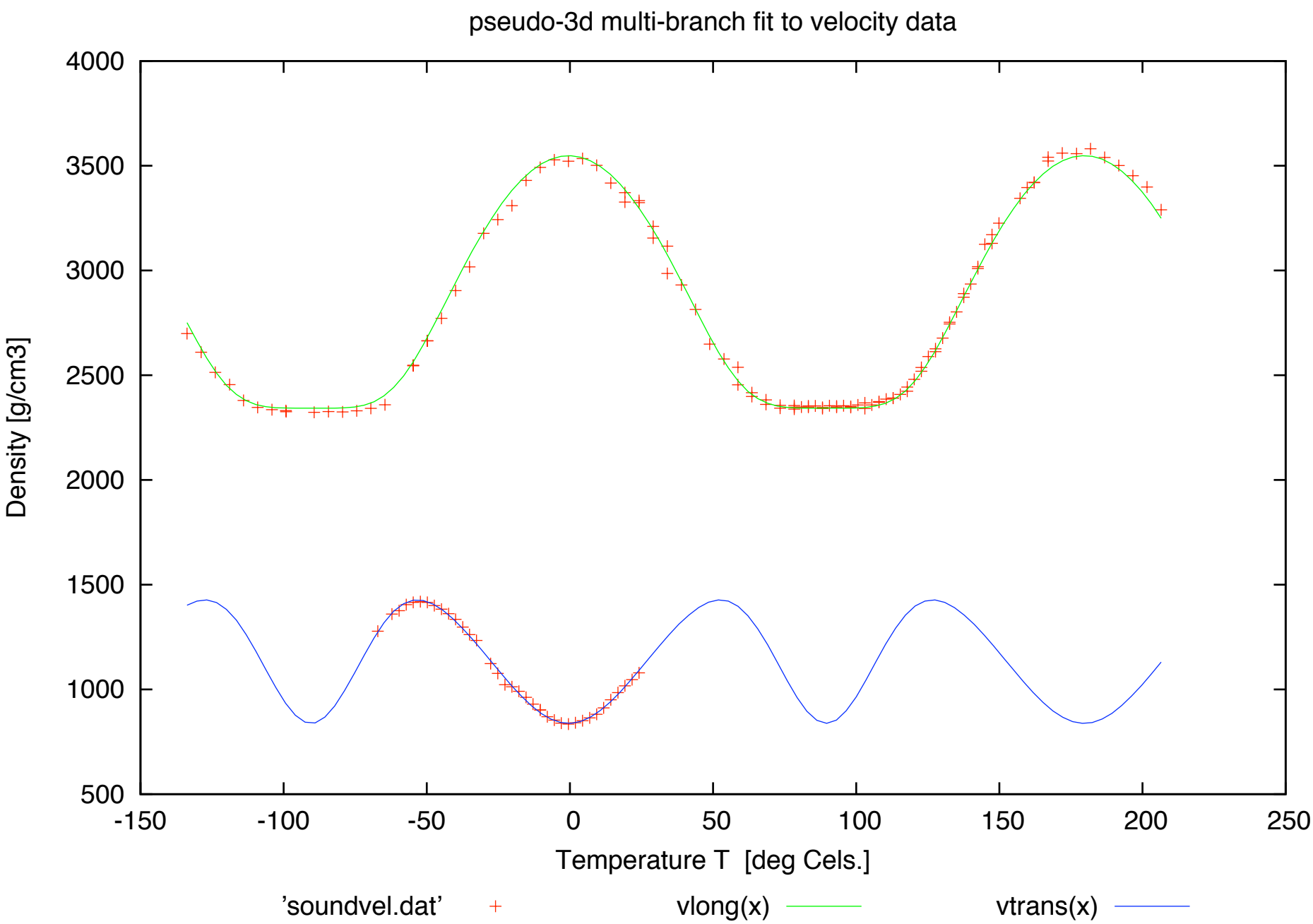

fit with more iterations
1.034
1.032
1.03
1.028
1.026
1.024
1.022
Density [g/cm3]
39
40
41
42
43
44
45
46
47
48
49
50
Temperature T [deg Cels.]
'lcdemo.dat'
density(x)

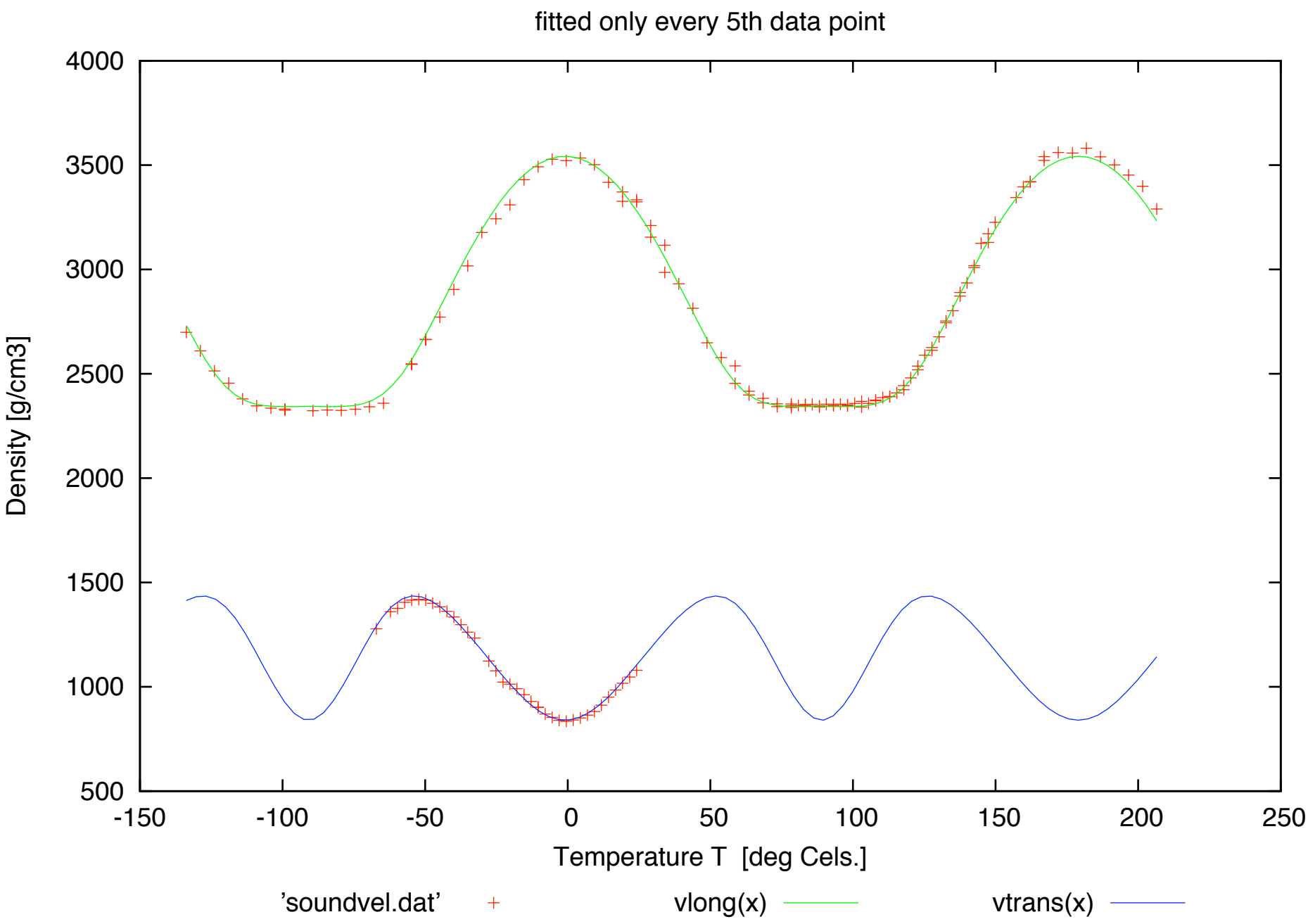
the scattered points, and the initial parameter

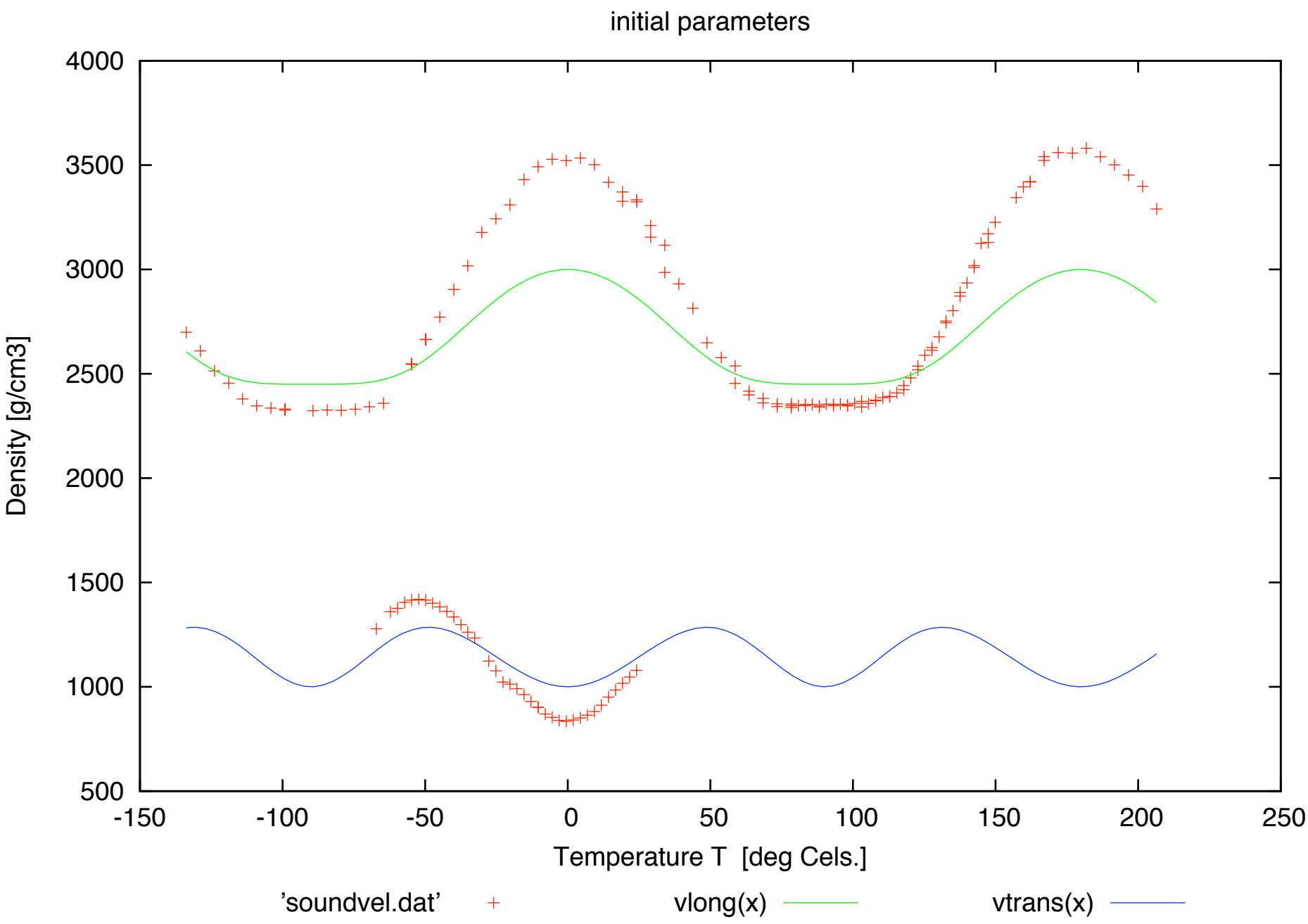


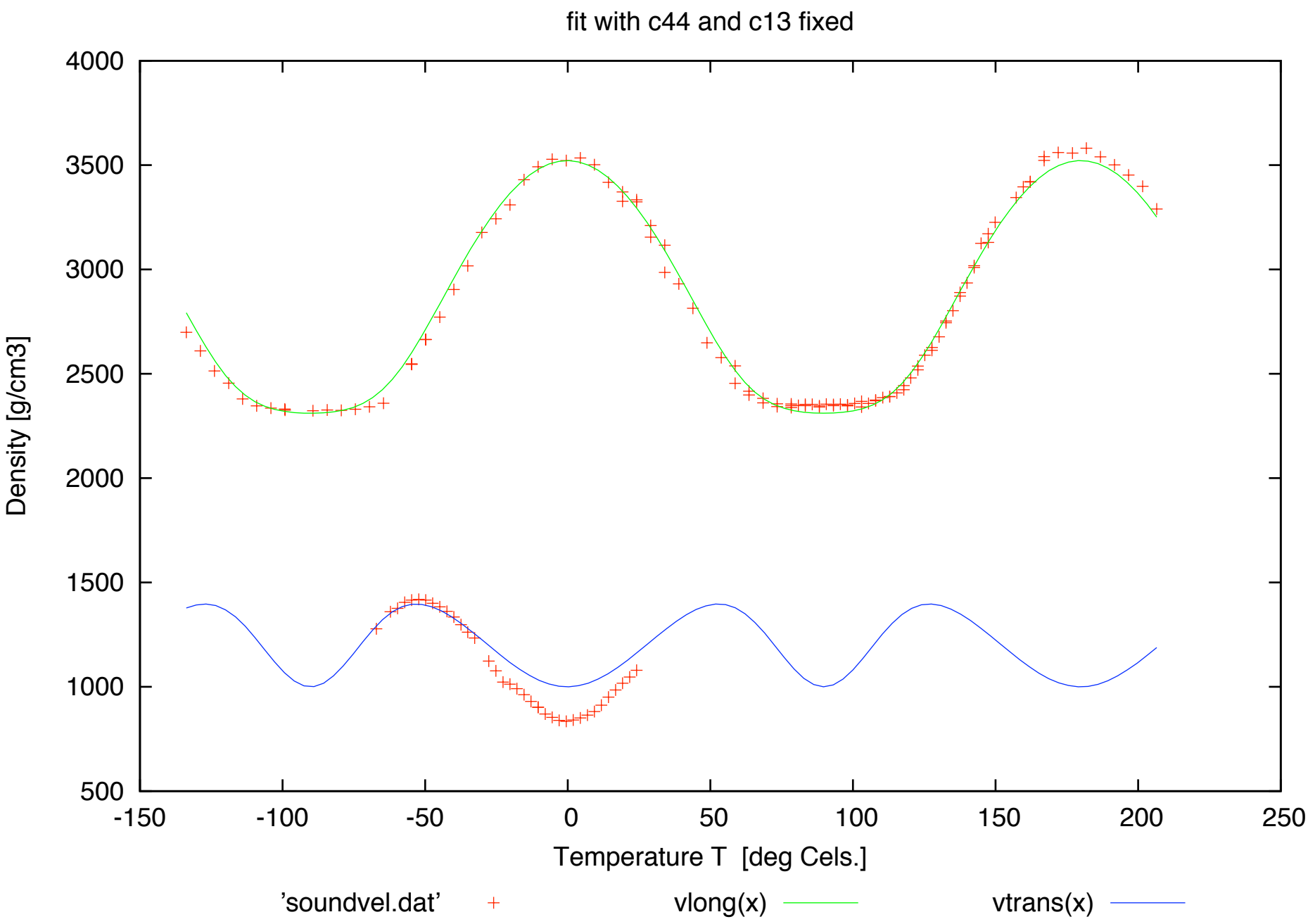

the scattered points, fitted curve
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Temperature T [deg Cels.]
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Density [g/cm3]
'hemisphr.dat' using 1:2:3
h(x,y)

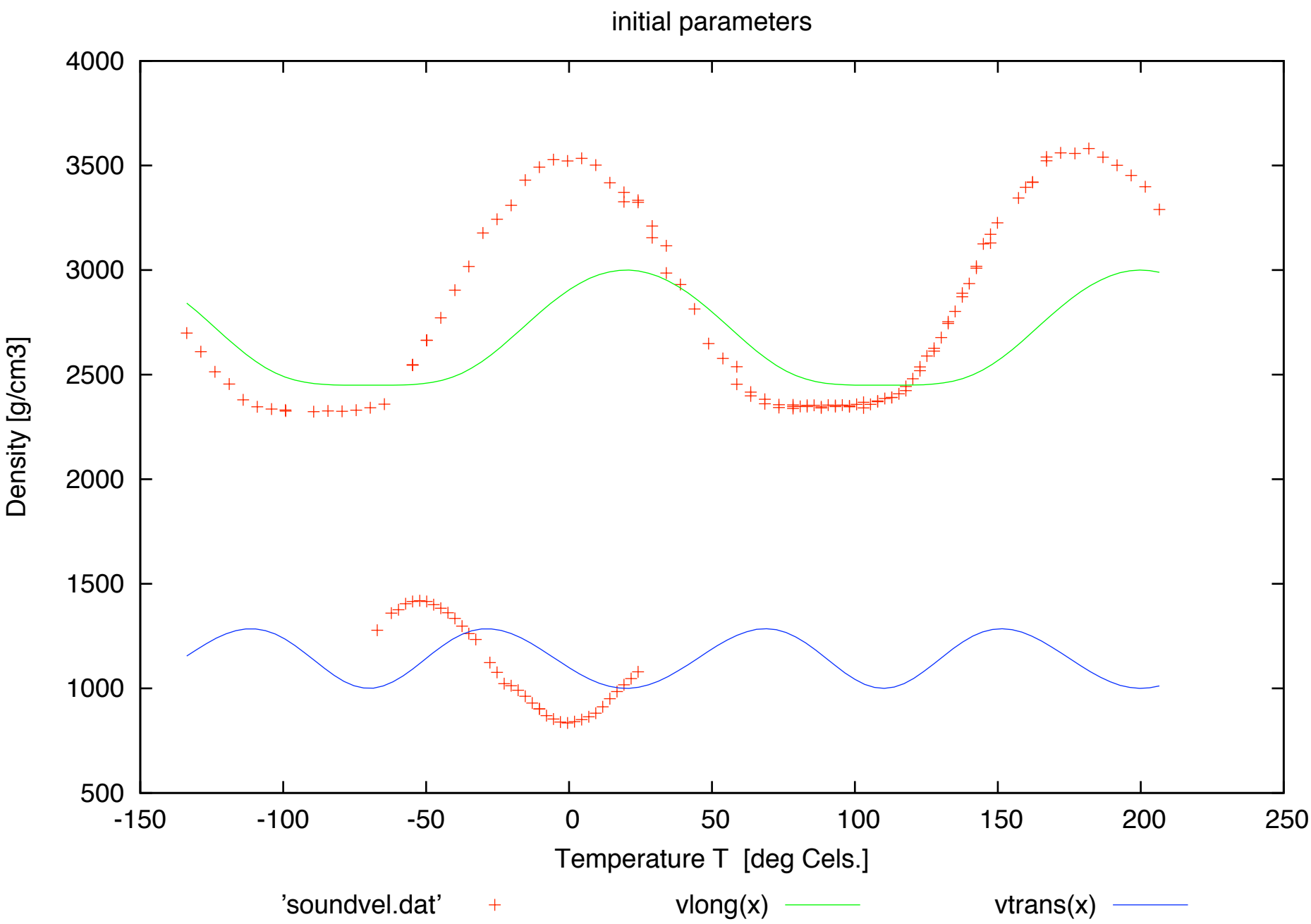

sound data, and model with initial parameters
Density [g/cm3]
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
Temperature T [deg Cels.]
'soundvel.dat'
vlong(x)
vtrans(x)

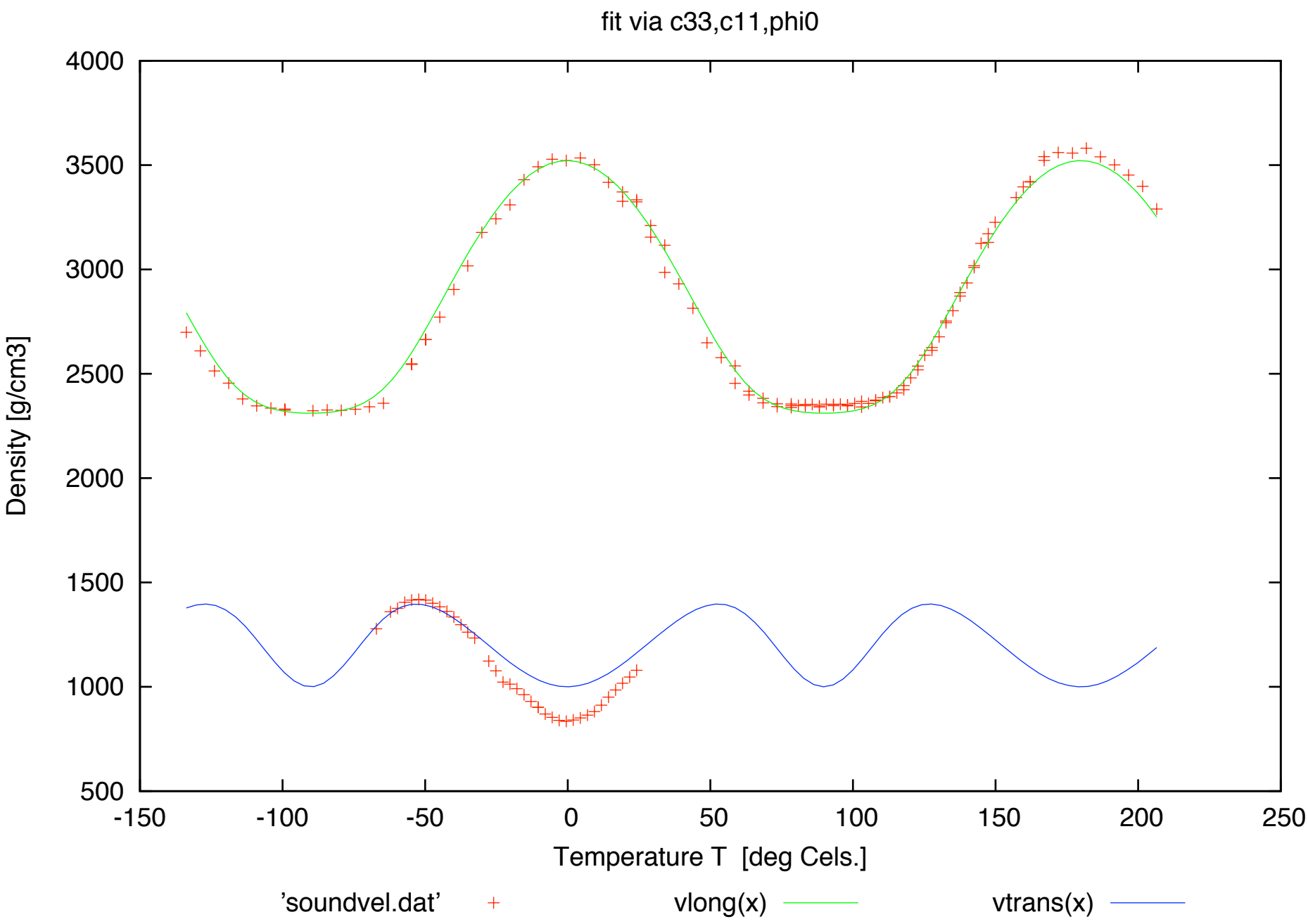

pseudo-3d multi-branch fit to velocity data
Density [g/cm3]
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
Temperature T [deg Cels.]
'soundvel.dat'
vlong(x)
vtrans(x)

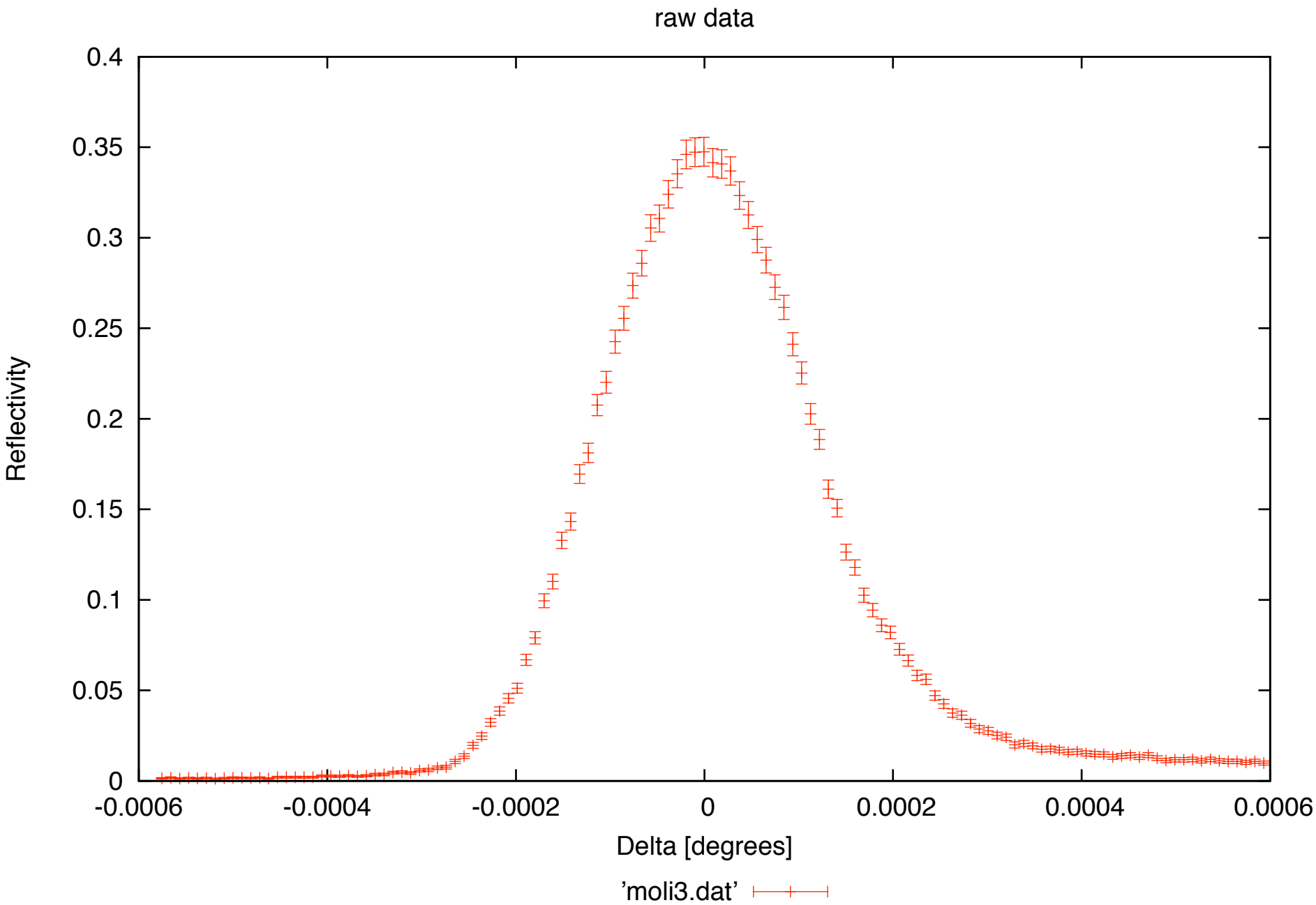

fitted only every 5th data point
Density [g/cm3]
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
Temperature T [deg Cels.]
'soundvel.dat'
vlong(x)
vtrans(x)

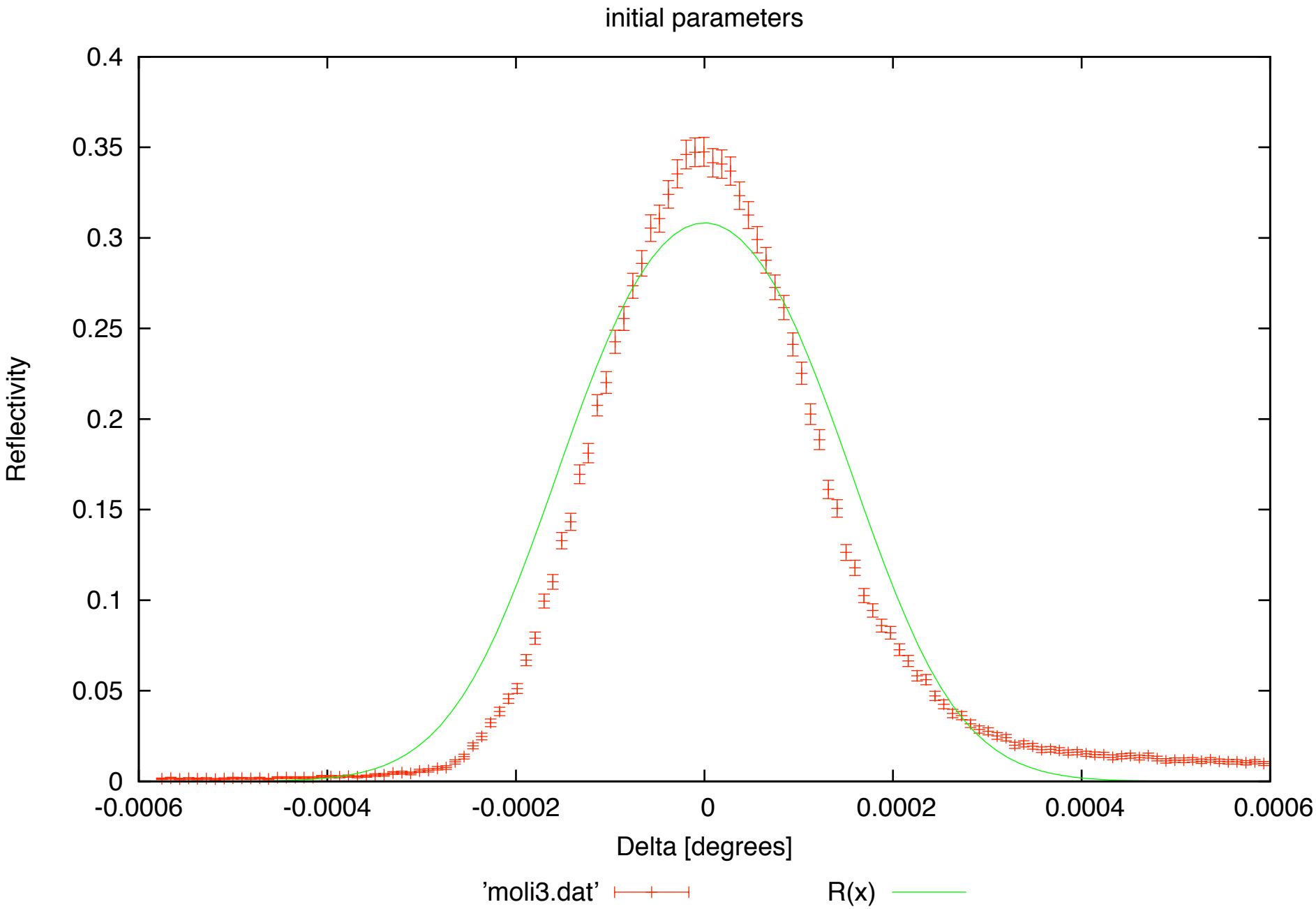

initial parameters
Density [g/cm3]
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
Temperature T [deg Cels.]
'soundvel.dat'
vlong(x)
vtrans(x)

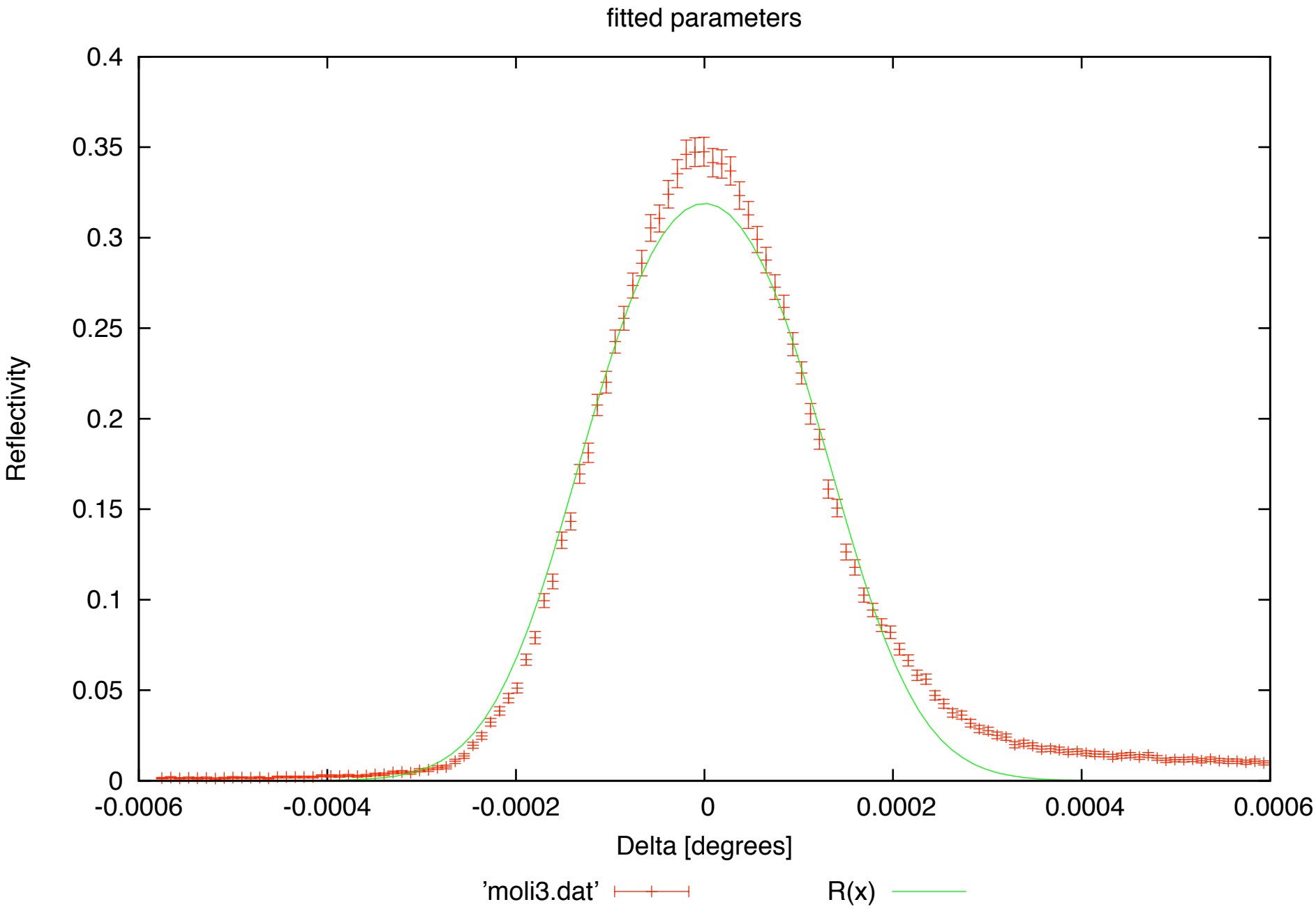

fit with c44 and c13 fixed
Density [g/cm3]
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
Temperature T [deg Cels.]
'soundvel.dat'
vlong(x)
vtrans(x)

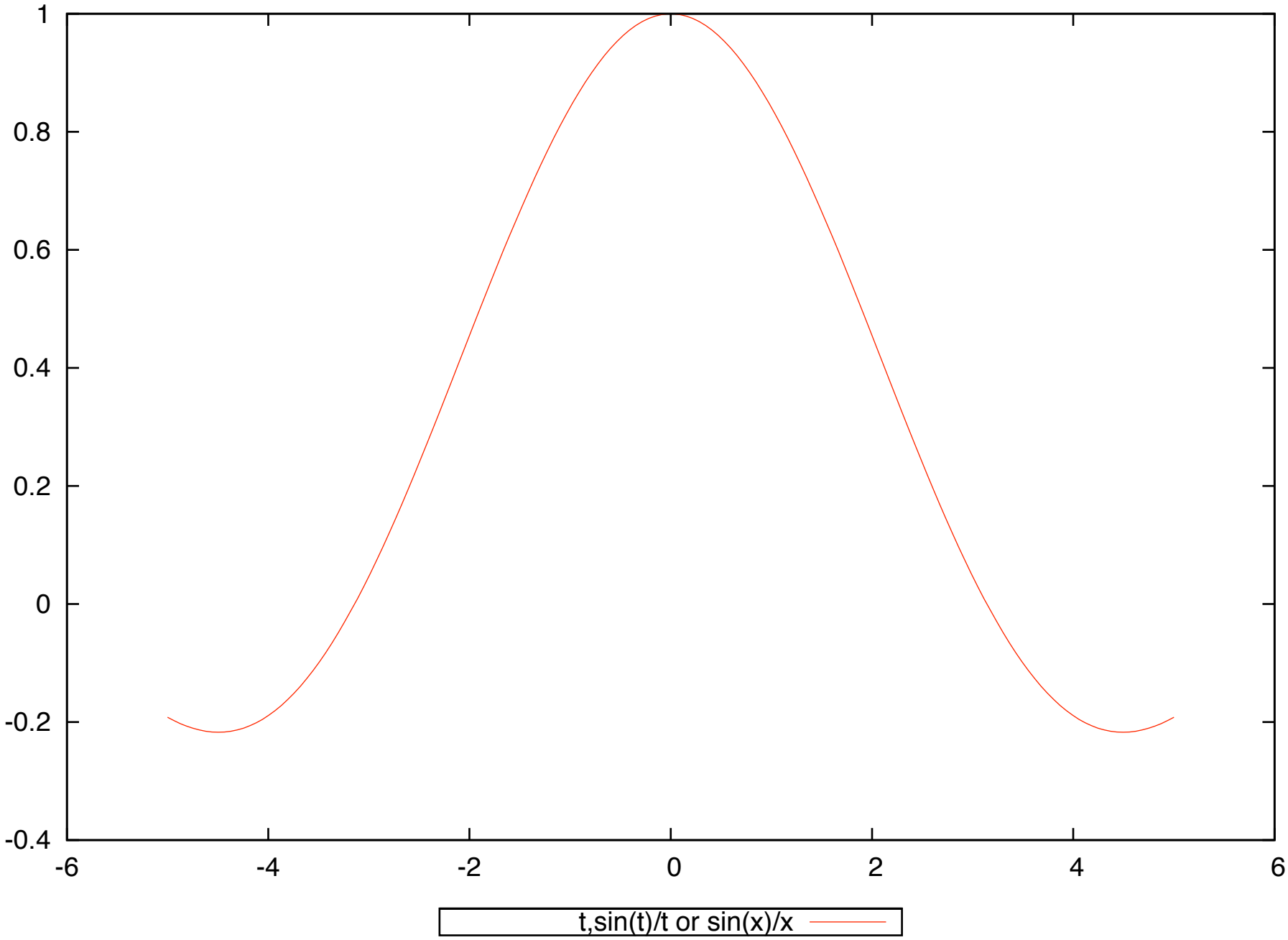

initial parameters
Density [g/cm3]
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
Temperature T [deg Cels.]
'soundvel.dat'
vlong(x)
vtrans(x)

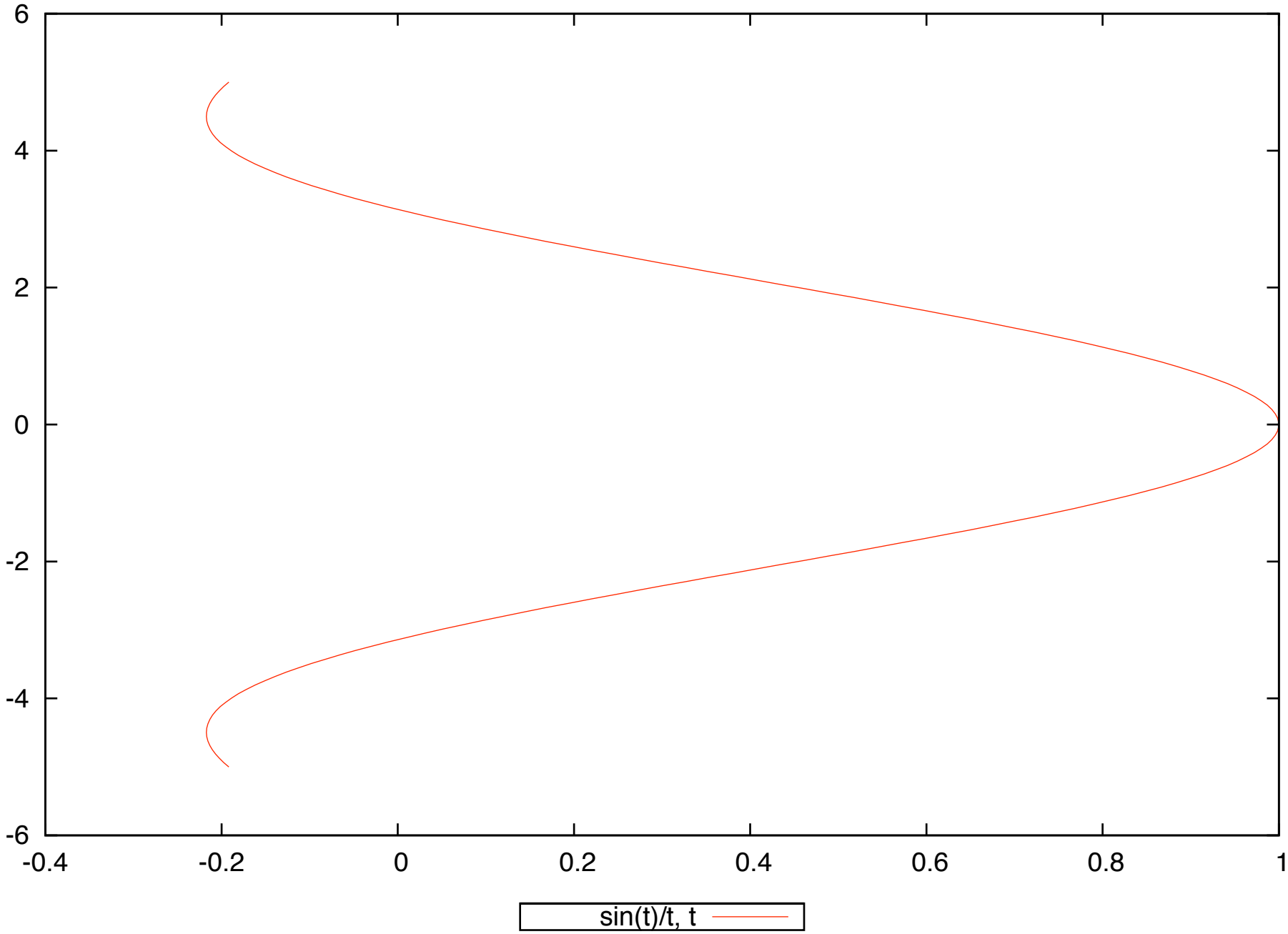

fit via c33,c11,phi0
Density [g/cm3]
4000
3500
3000
2500
2000
1500
1000
500
-150
-100
-50
0
50
100
150
200
250
Temperature T [deg Cels.]
'soundvel.dat'
vlong(x)
vtrans(x)

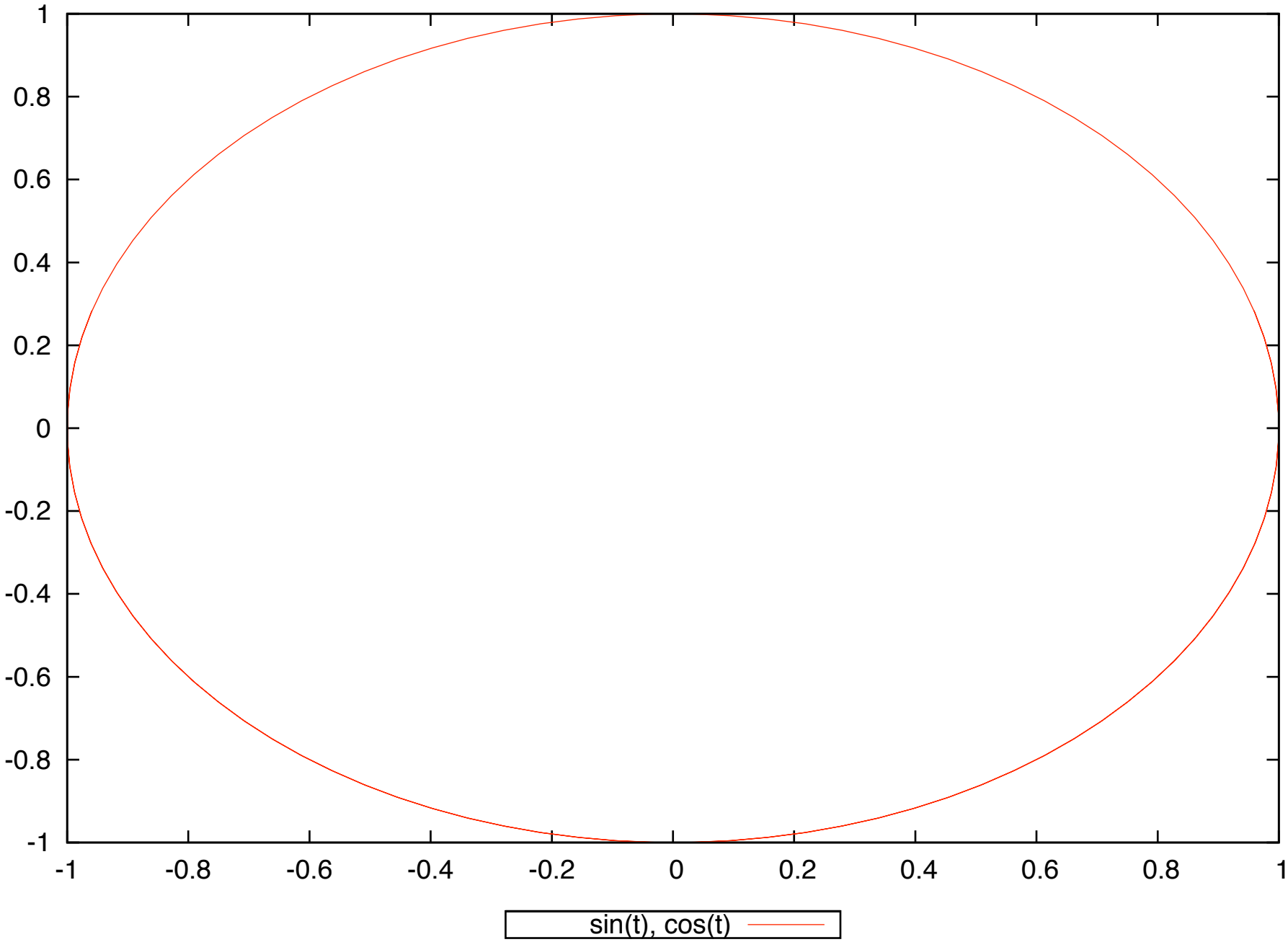

raw data
Reflectivity
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.0006
-0.0004
-0.0002
0
0.0002
0.0004
0.0006
Delta [degrees]
'moli3.dat'

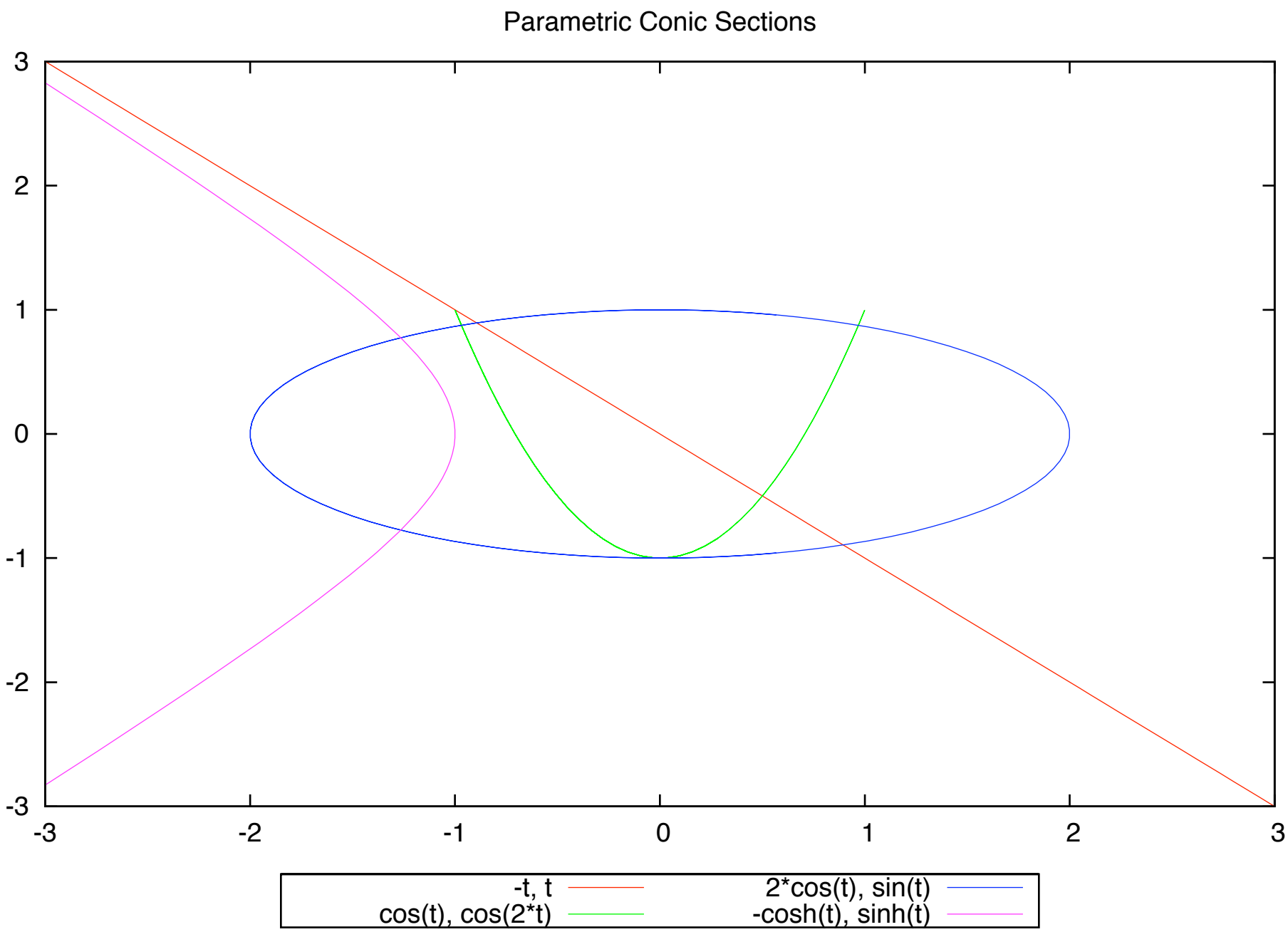

initial parameters
Reflectivity
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.0006
-0.0004
-0.0002
0
0.0002
0.0004
0.0006
Delta [degrees]
'moli3.dat'
R(x)

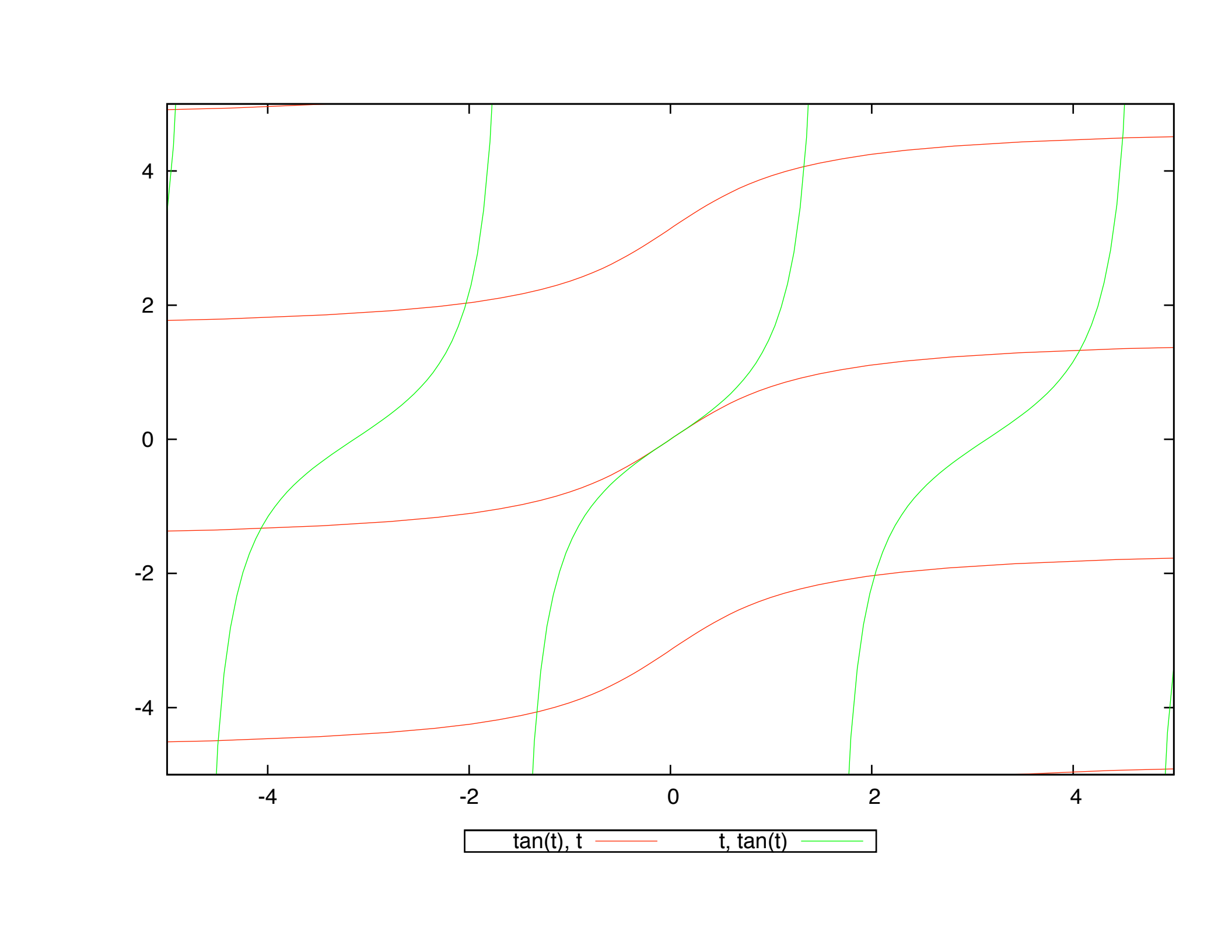

fitted parameters
Reflectivity
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.0006
-0.0004
-0.0002
0
0.0002
0.0004
0.0006
Delta [degrees]
'moli3.dat'
R(x)

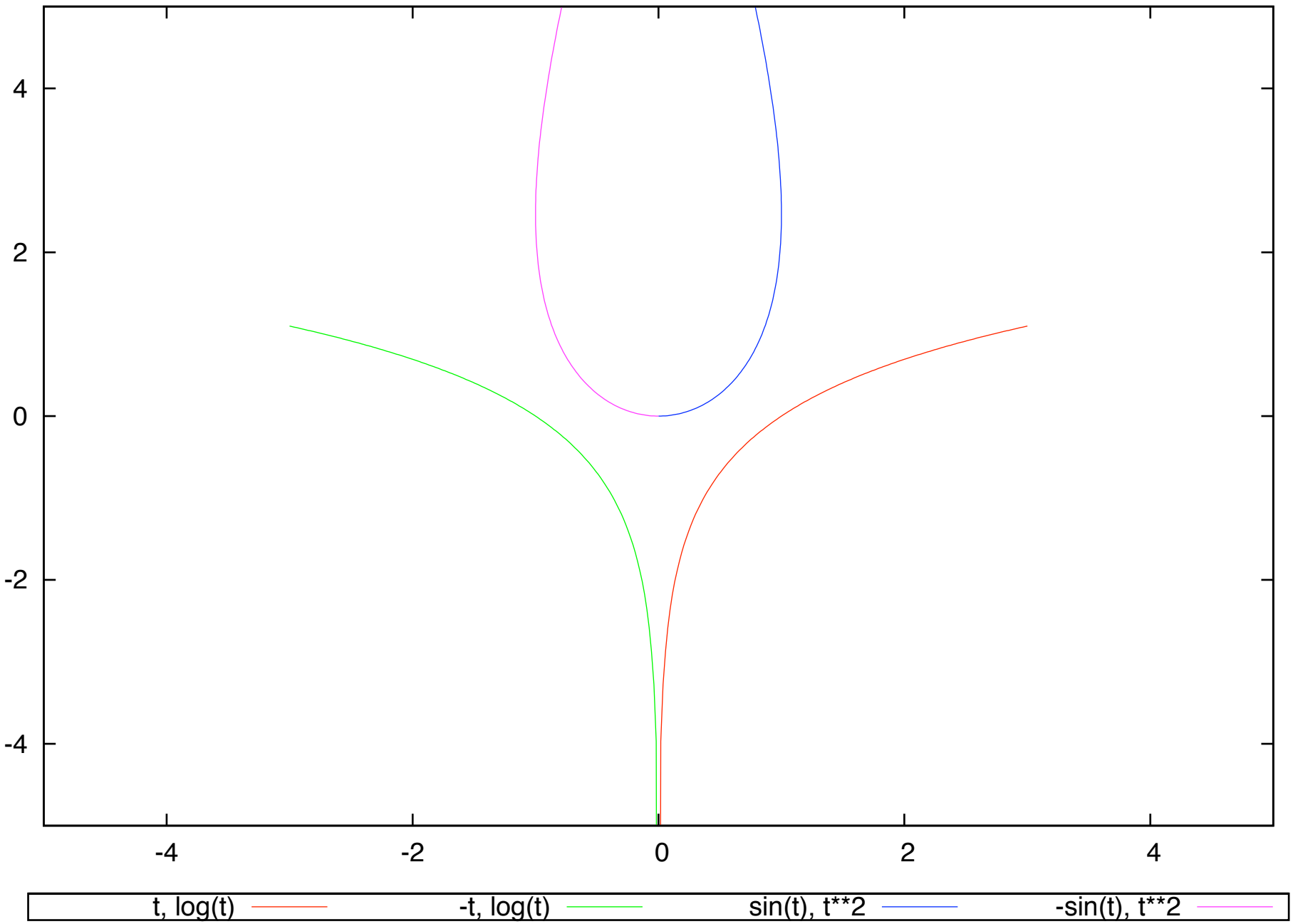

1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-6
-4
-2
0
2
4
6
t,sin(t)/t or sin(x)/x

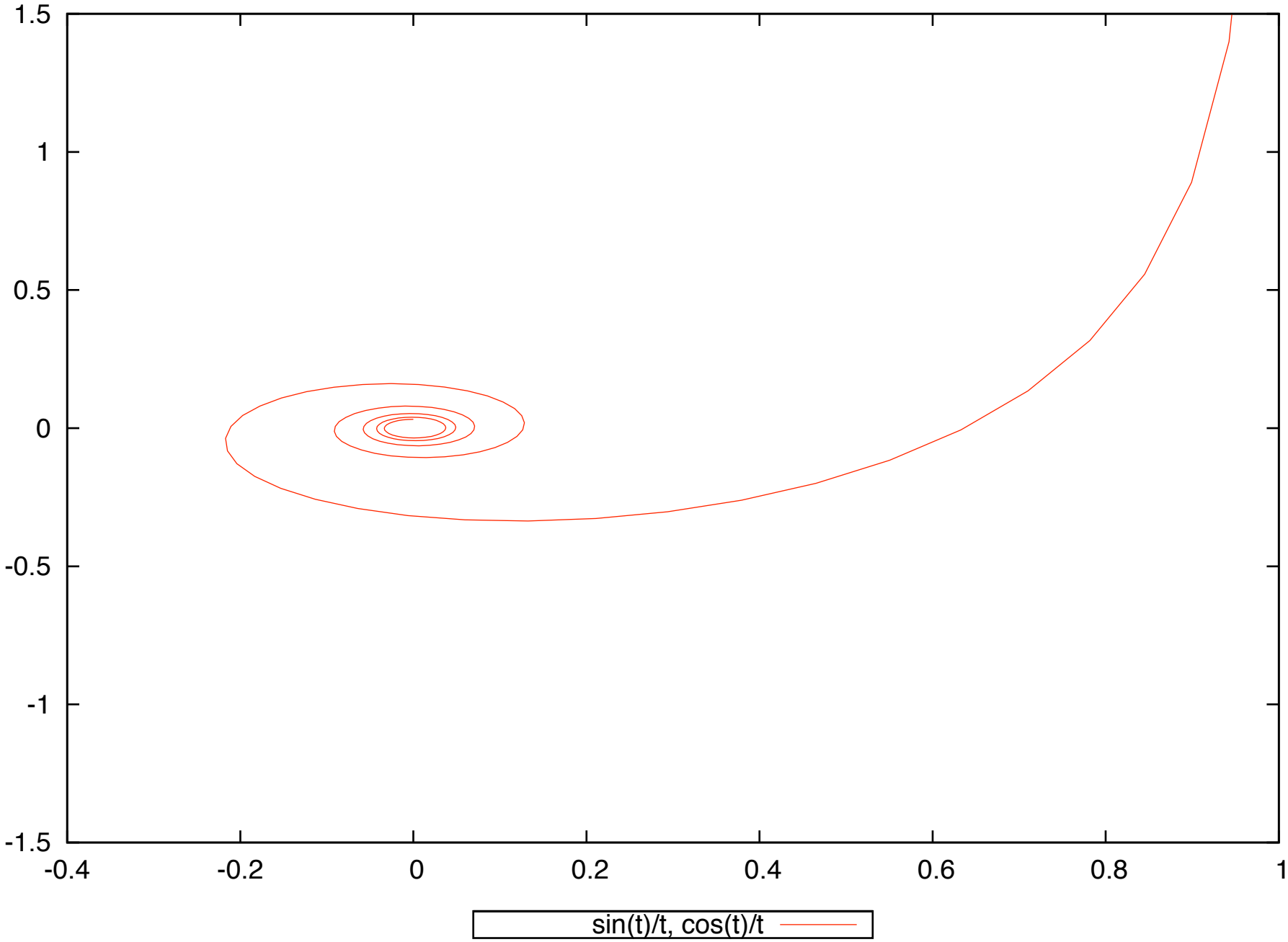

6
4
2
0
-2
-4
-6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
sin(t)/t, t

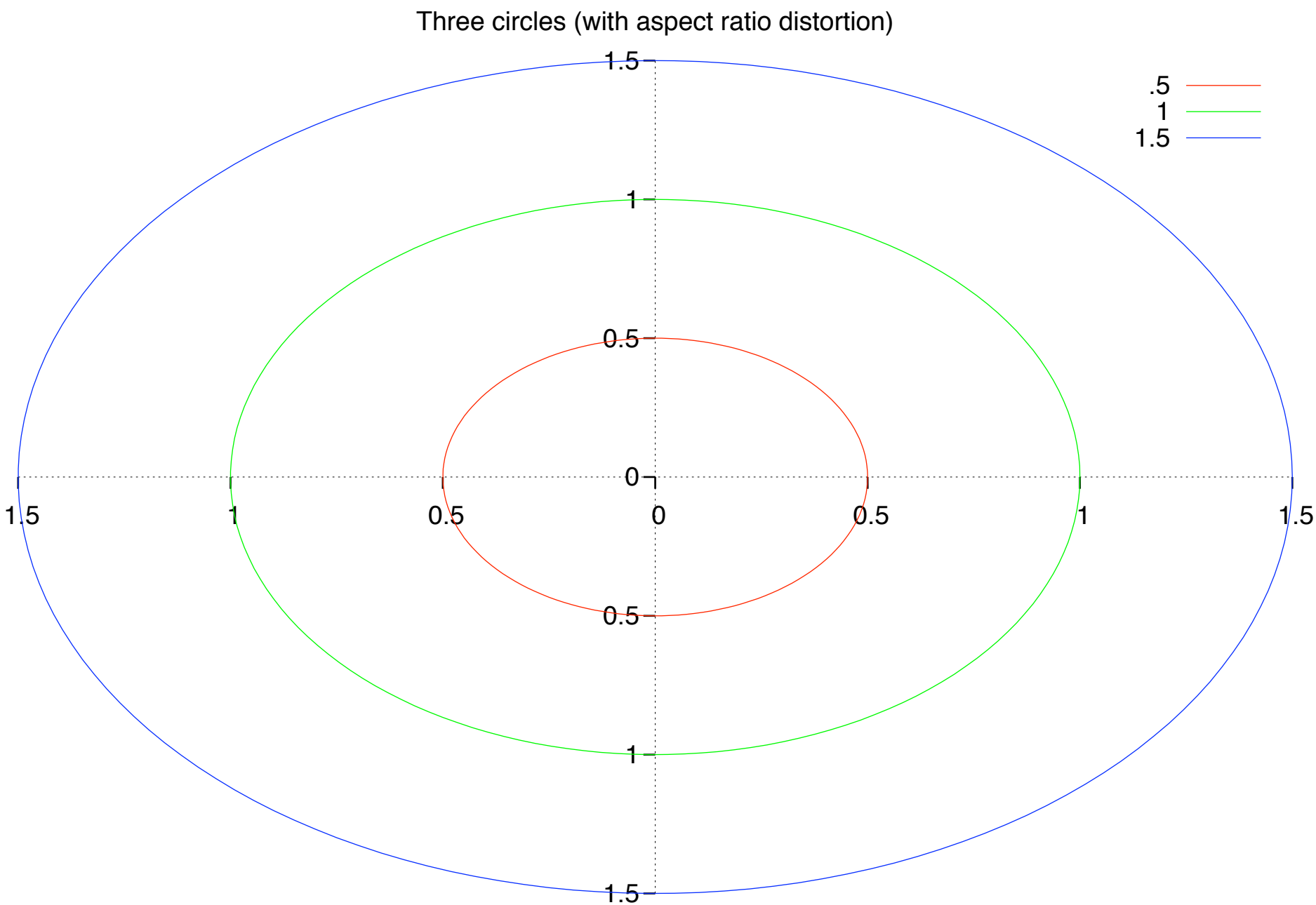

1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
sin(t), cos(t)

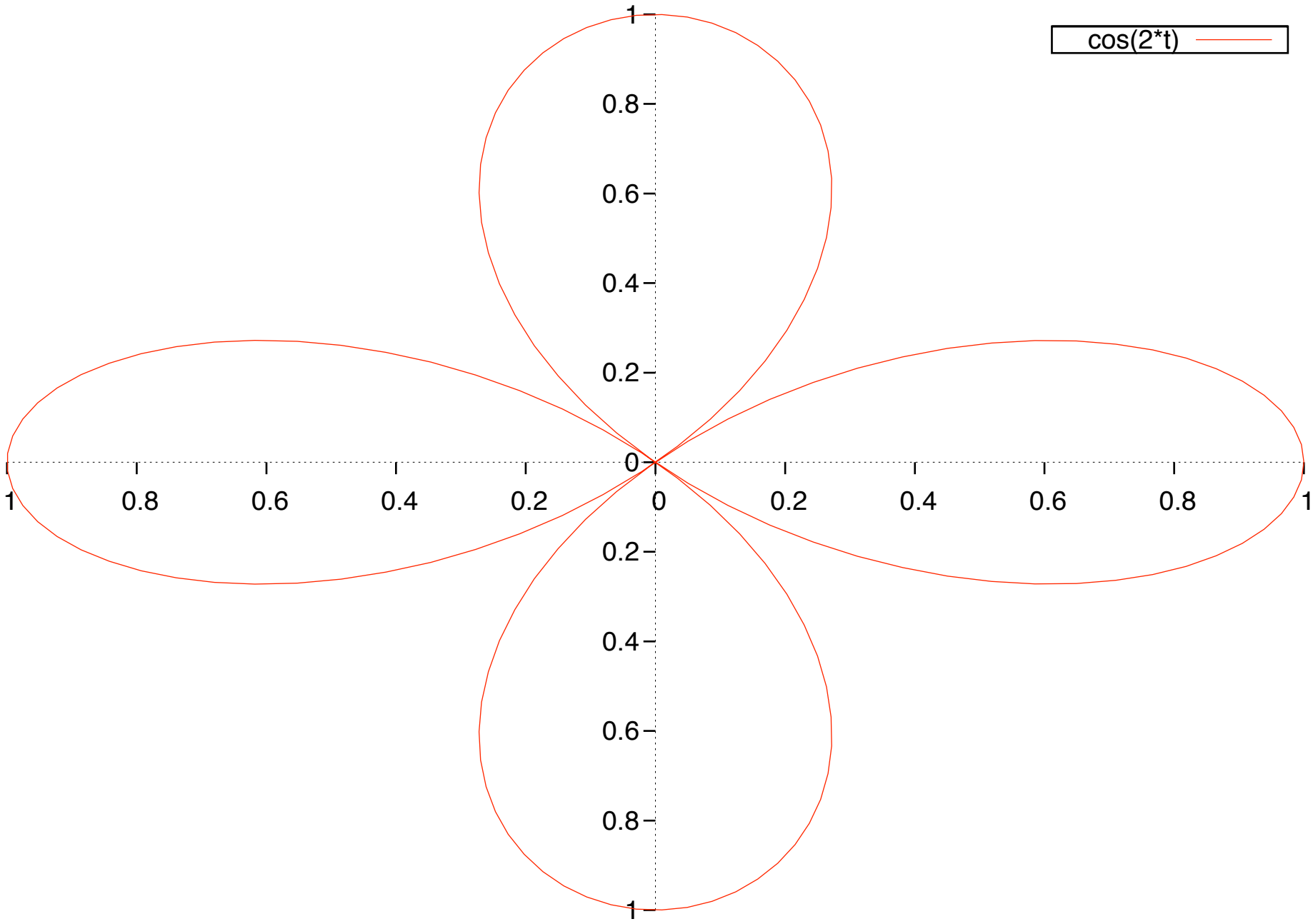

Parametric Conic Sections
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
-t, t
cos(t), cos(2*t)
2*cos(t), sin(t)
-cosh(t), sinh(t)

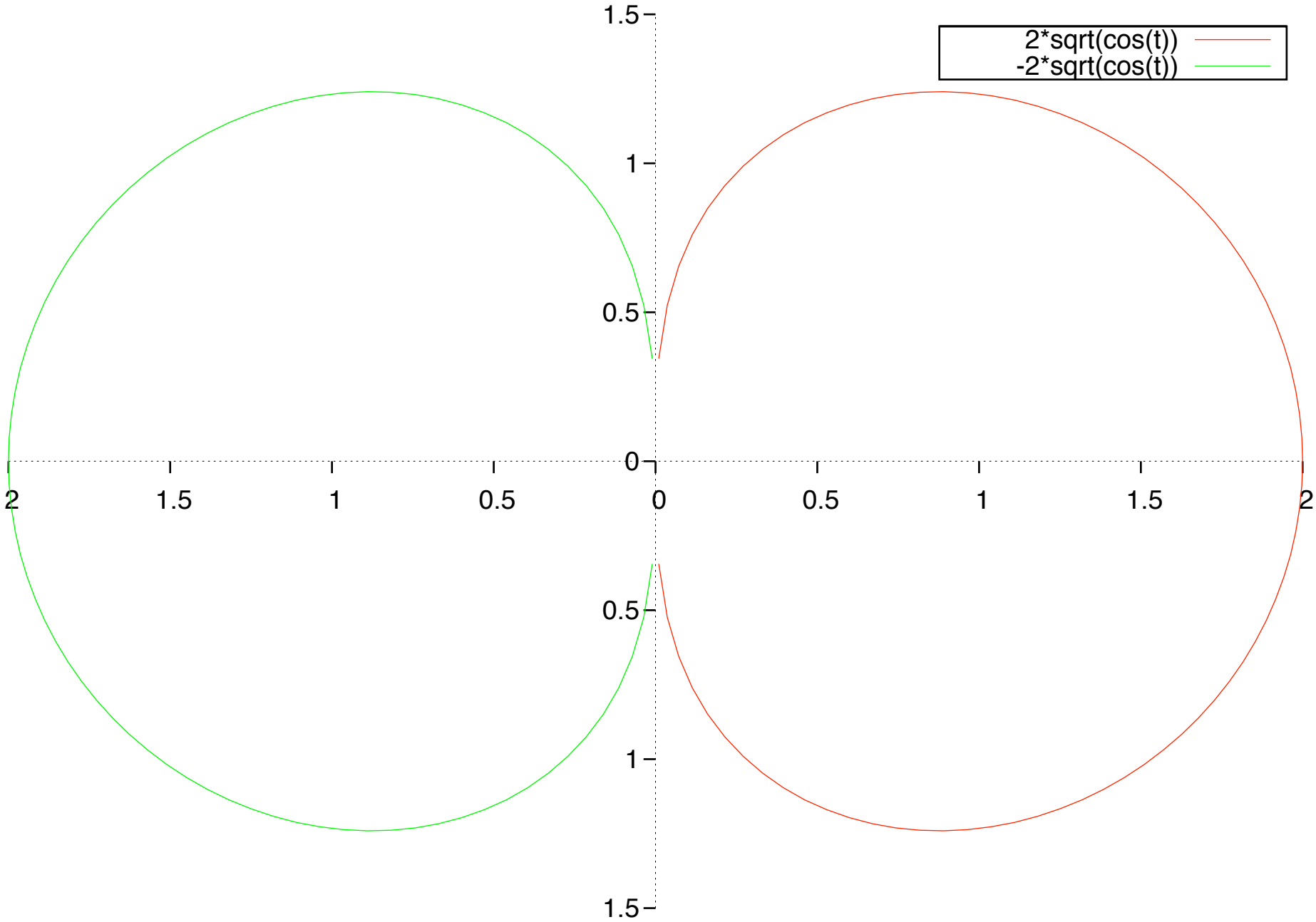

4
2
0
-2
-4
-4
-2
0
2
4
tan(t), t
t, tan(t)

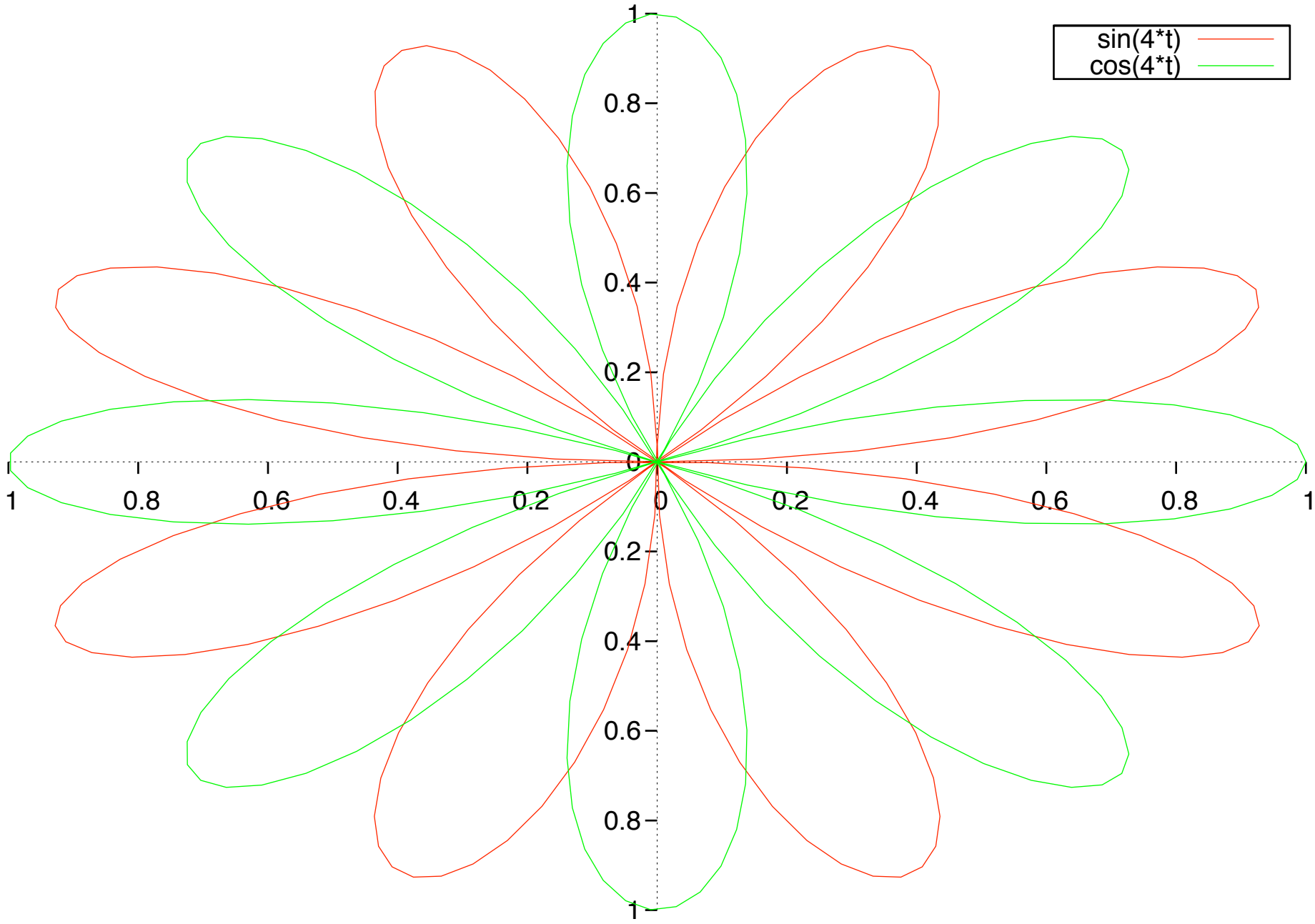

4
2
0
-2
-4
-4
-2
0
2
4
t, log(t)
-t, log(t)
sin(t), t**2
-sin(t), t**2

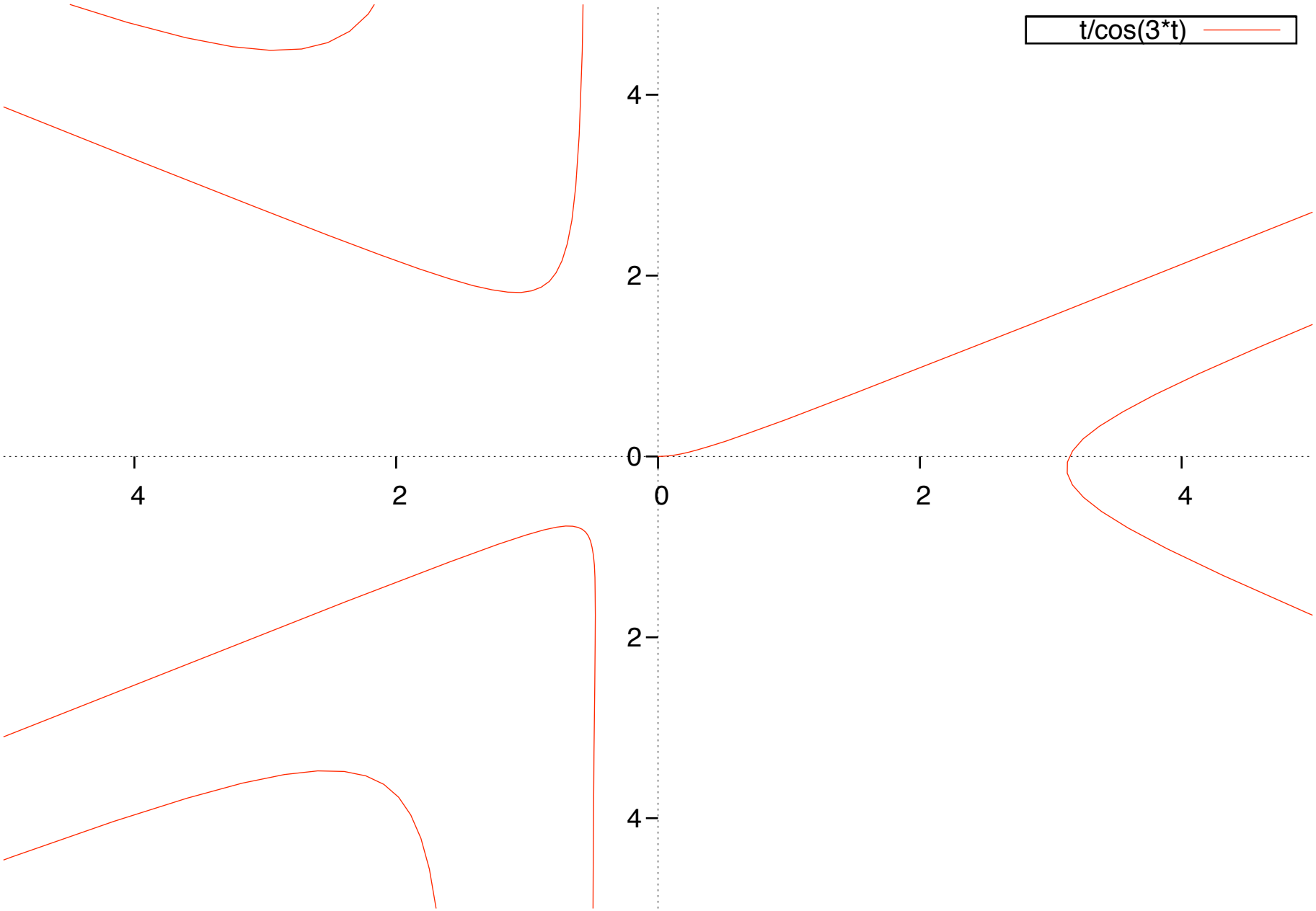

1.5
1
0.5
0
-0.5
-1
-1.5
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
sin(t)/t, cos(t)/t

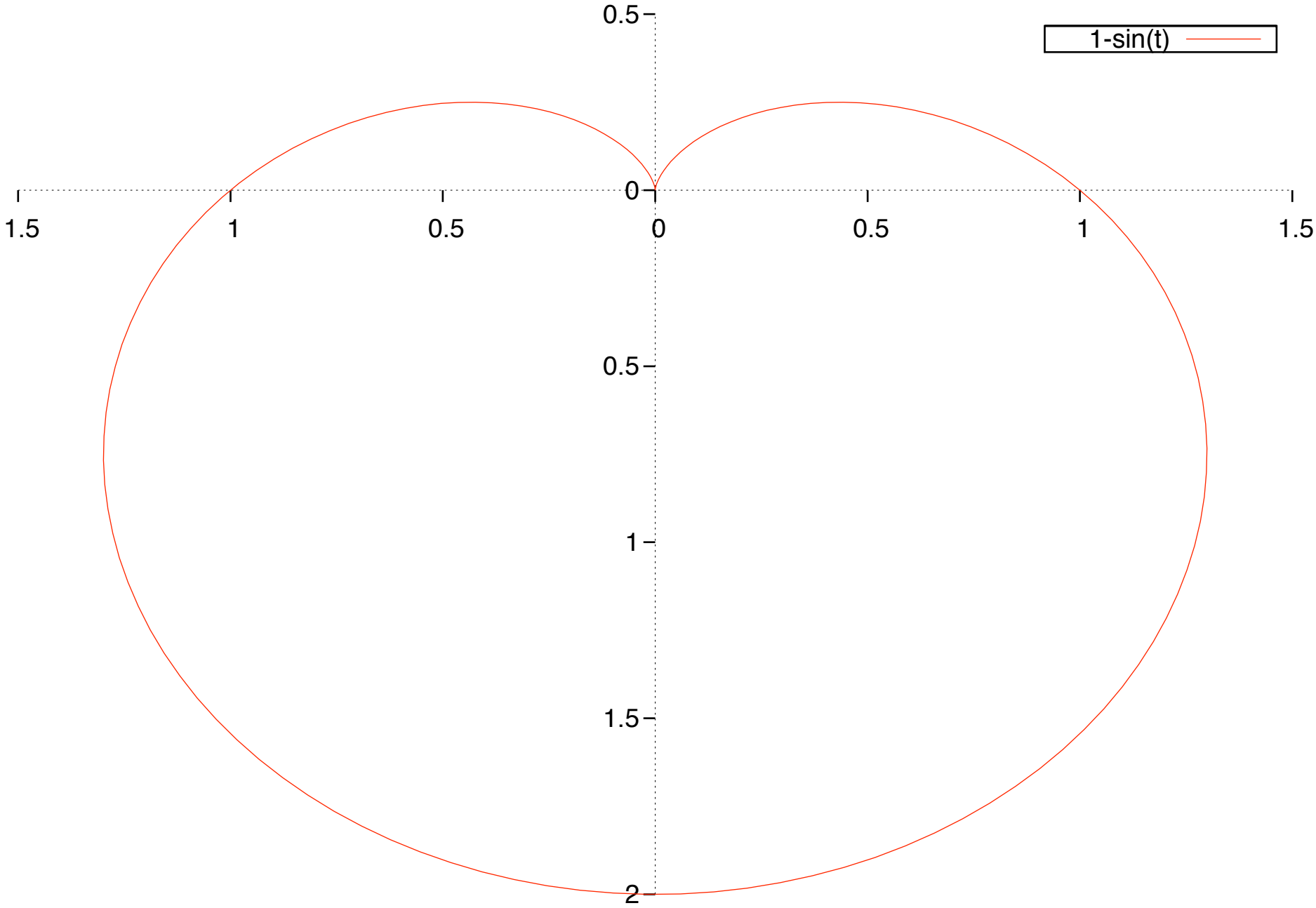

Three circles (with aspect ratio distortion)
.5
1
1.5
1.5
1
0.5
0
0.5
1
1.5
1.5
1
0.5
0
0.5
1
1.5

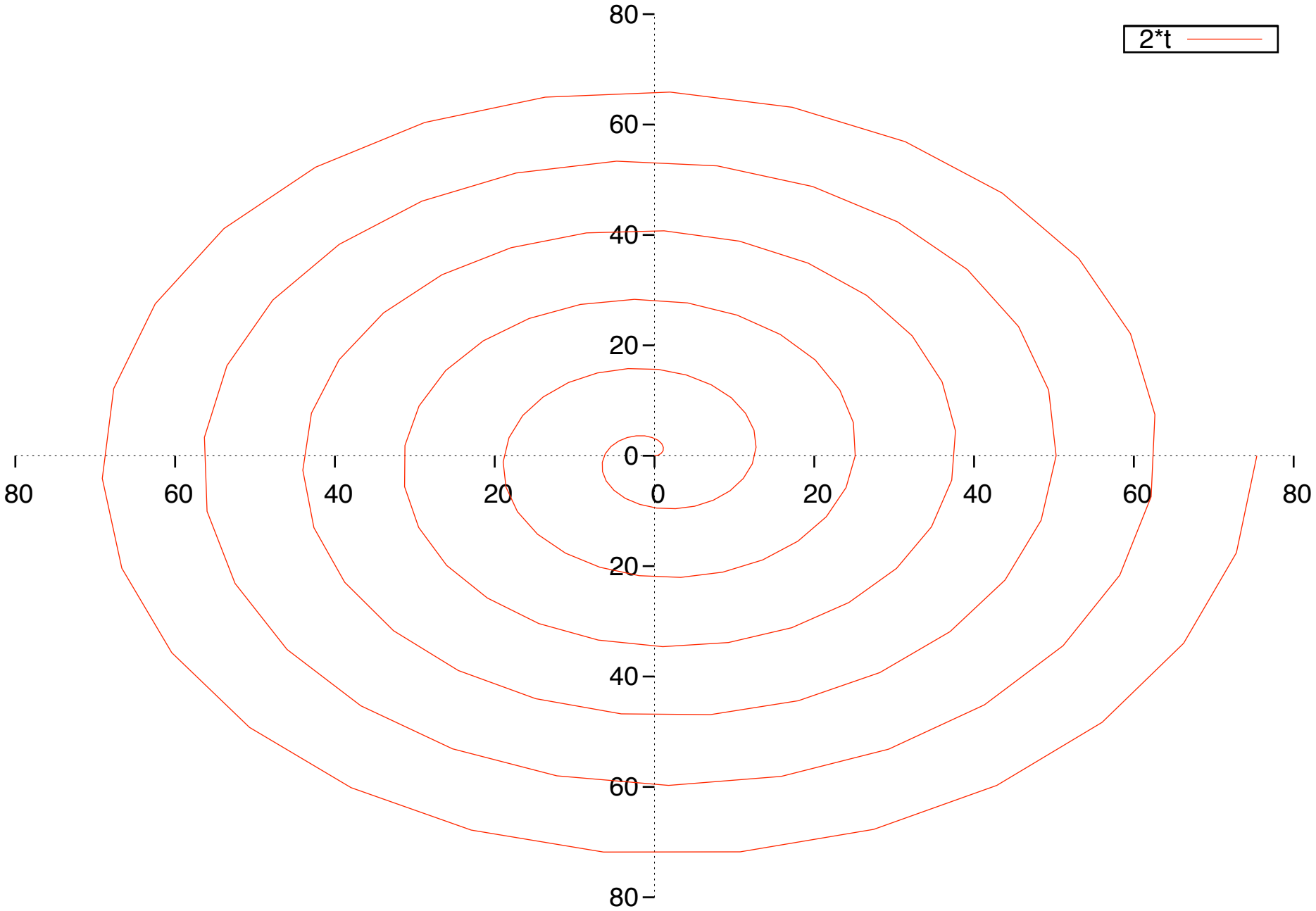

cos(2*t)
1
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1
1
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1

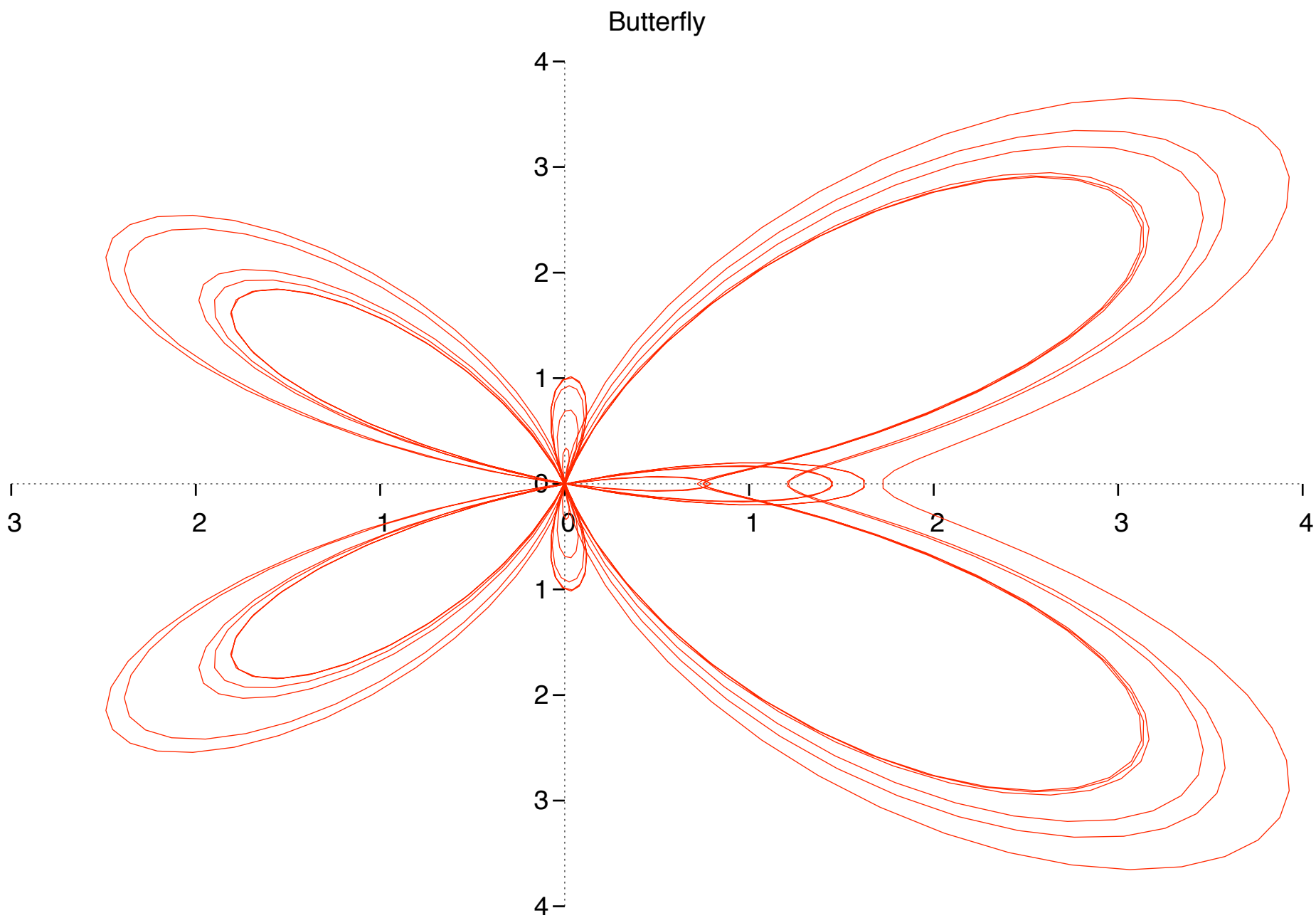

2*sqrt(cos(t))
-2*sqrt(cos(t))
1.5
1
0.5
0
0.5
1
1.5
2
2
1.5
1
0.5
0
0.5
1
1.5

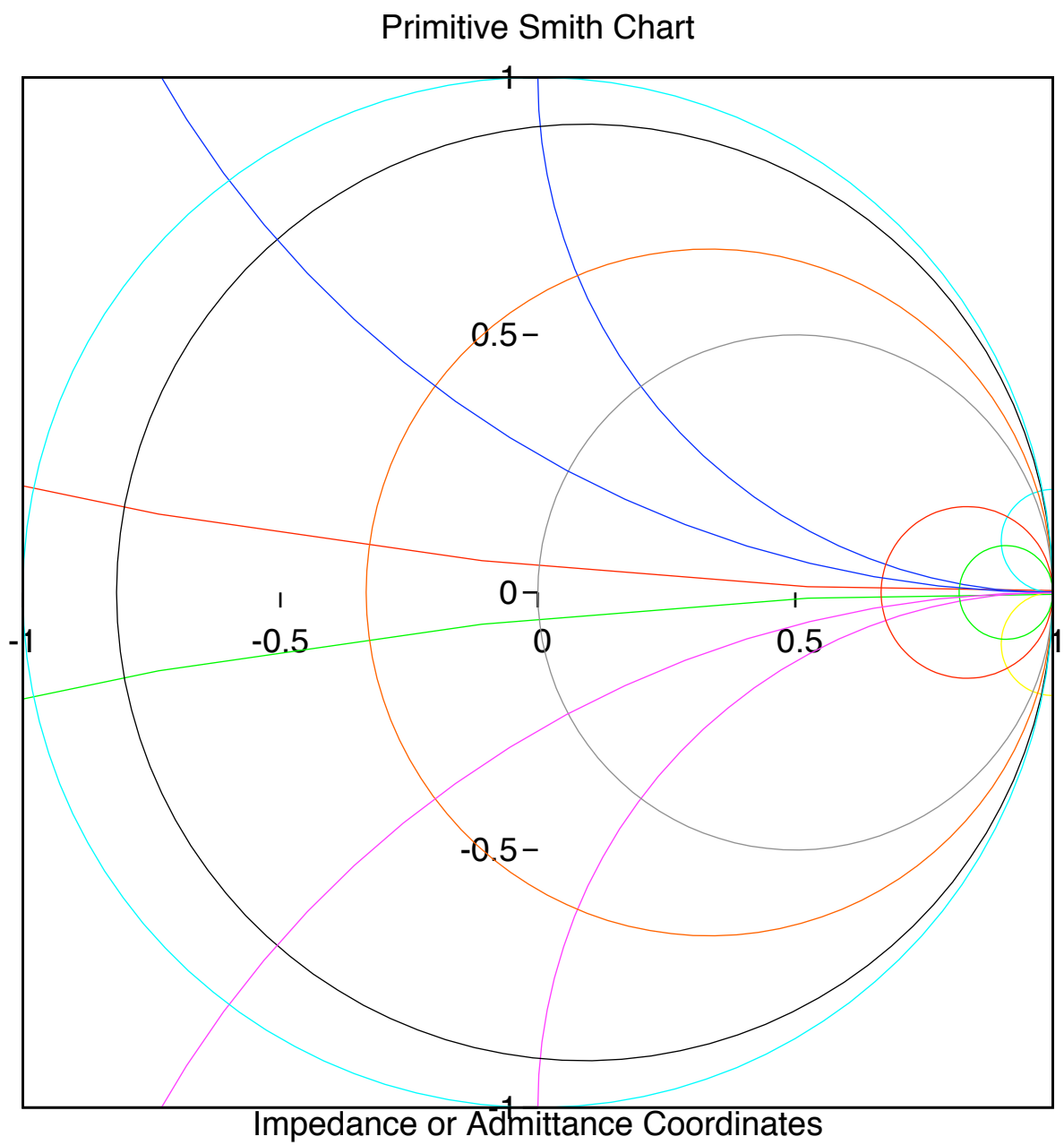

sin(4*t)
cos(4*t)
1
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1
1
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1

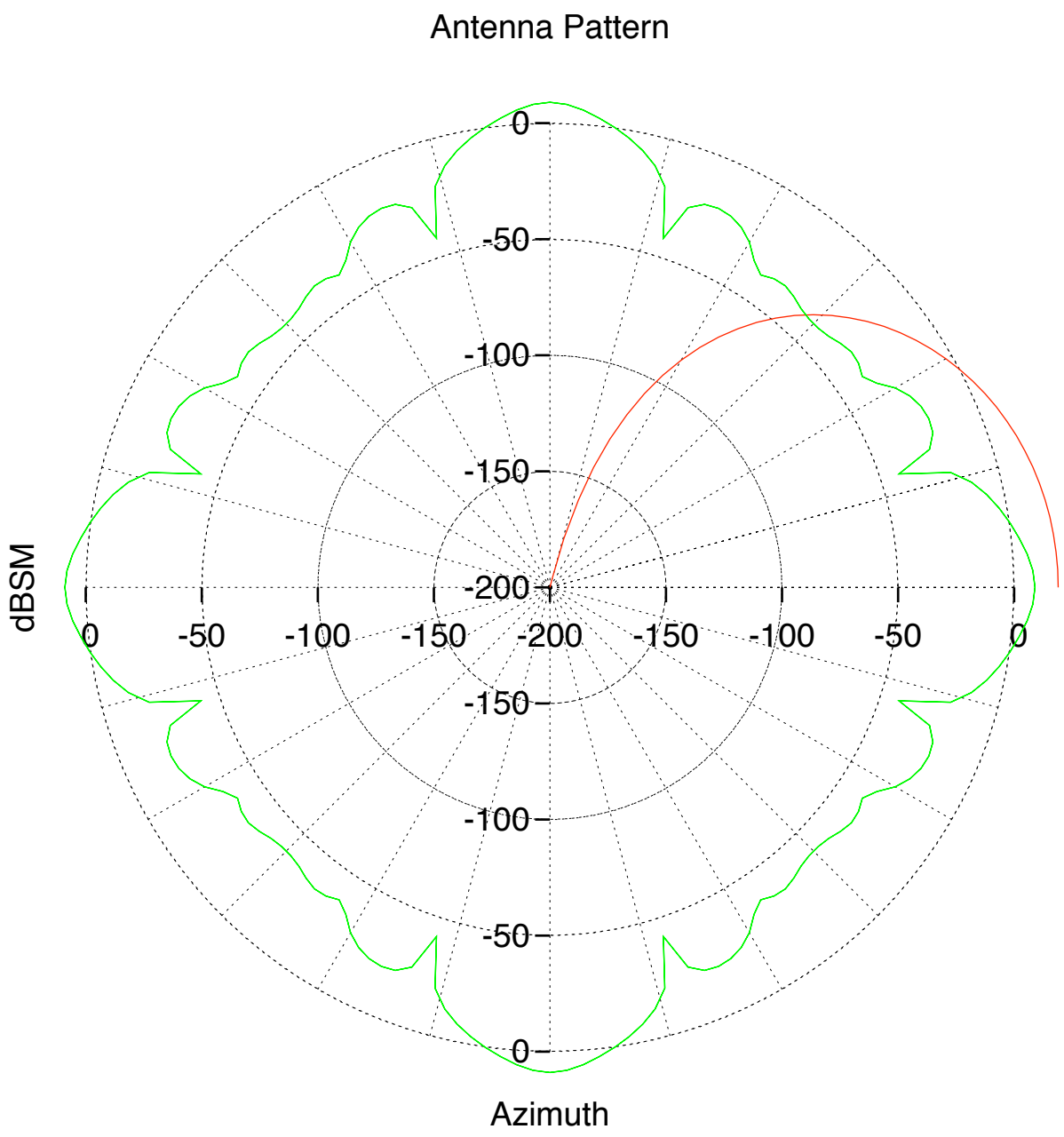

t/cos(3*t)
4
2
0
2
4
4
2
0
2
4

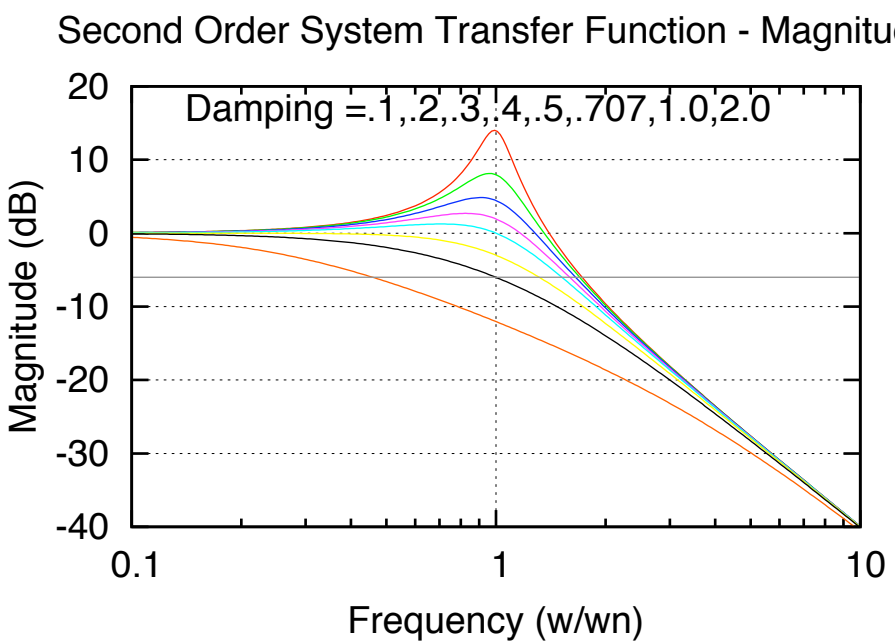

1-sin(t)
0.5
0
0.5
1
1.5
2
1.5
1
0.5
0
0.5
1
1.5

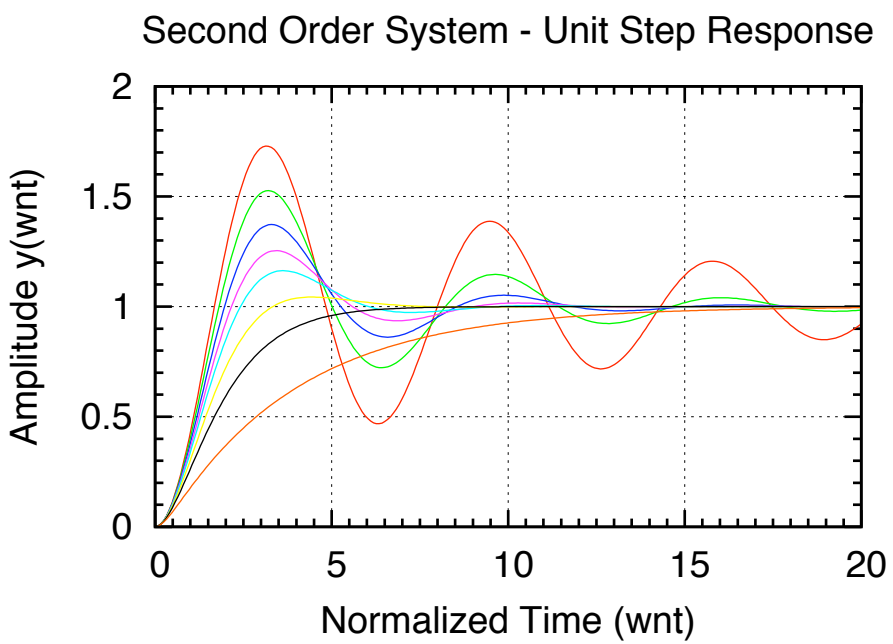

2*t
80
60
40
20
0
20
40
60
80
80
60
40
20
0
20
40
60
80

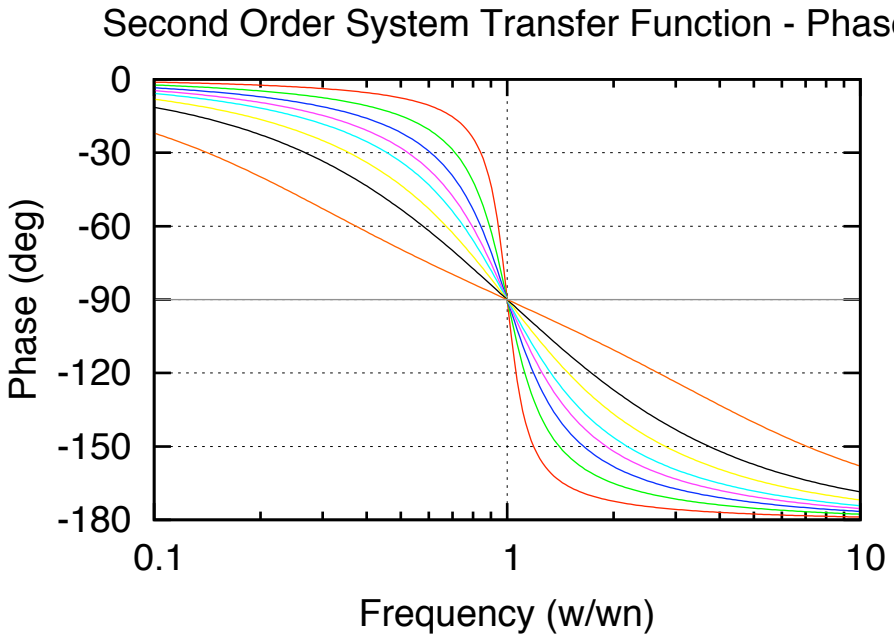

Butterfly
4
3
2
1
0
1
2
3
4
3
2
1
0
1
2
3
4

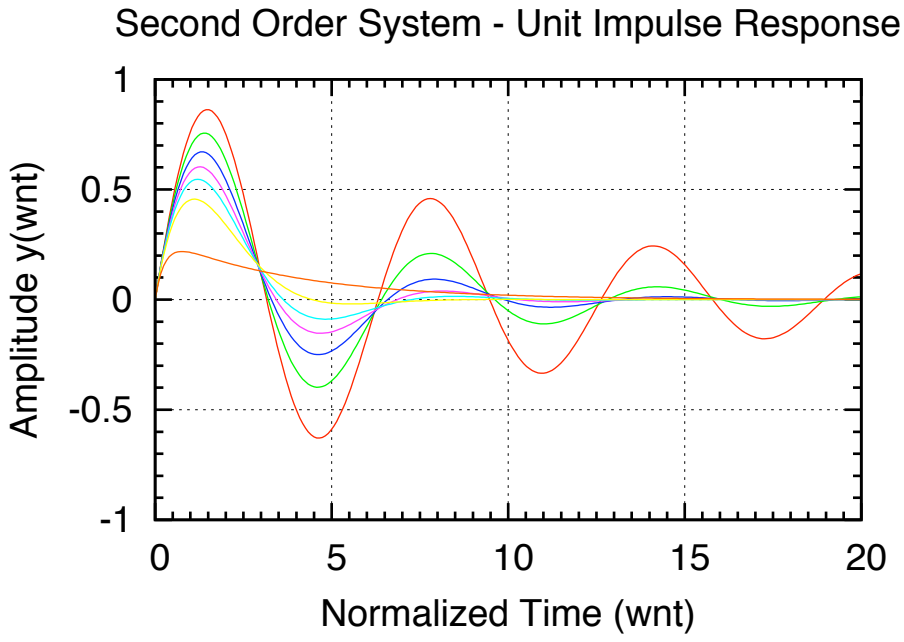

Primitive Smith Chart
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
Impedance or Admittance Coordinates
Wed Feb 04 12:18:46 2009

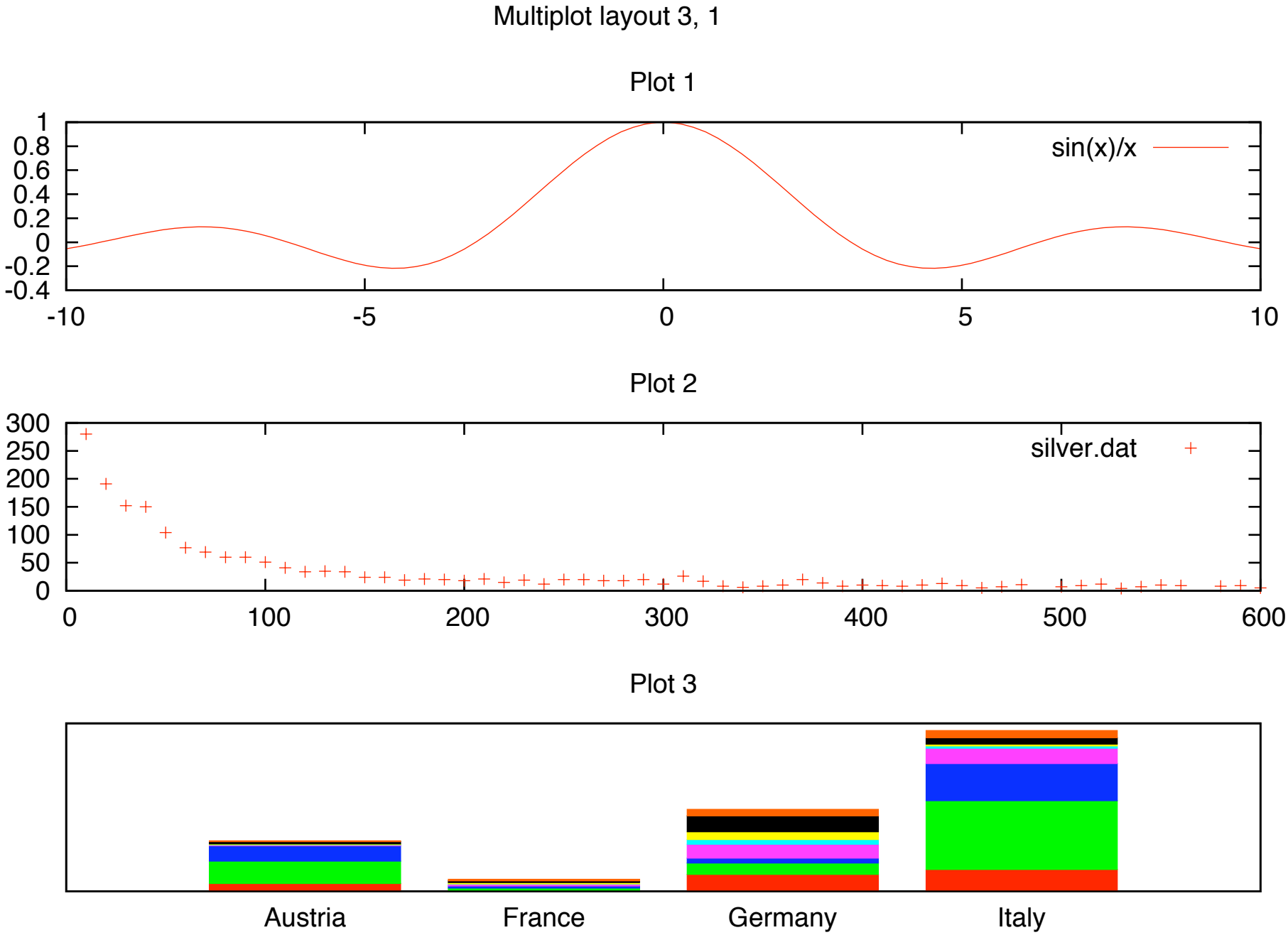

Antenna Pattern
dBSM
0
-50
-100
-150
-200
-150
-100
-50
0
0
-50
-100
-150
-200
-150
-100
-50
0
Azimuth
Wed Feb 04 12:18:46 2009

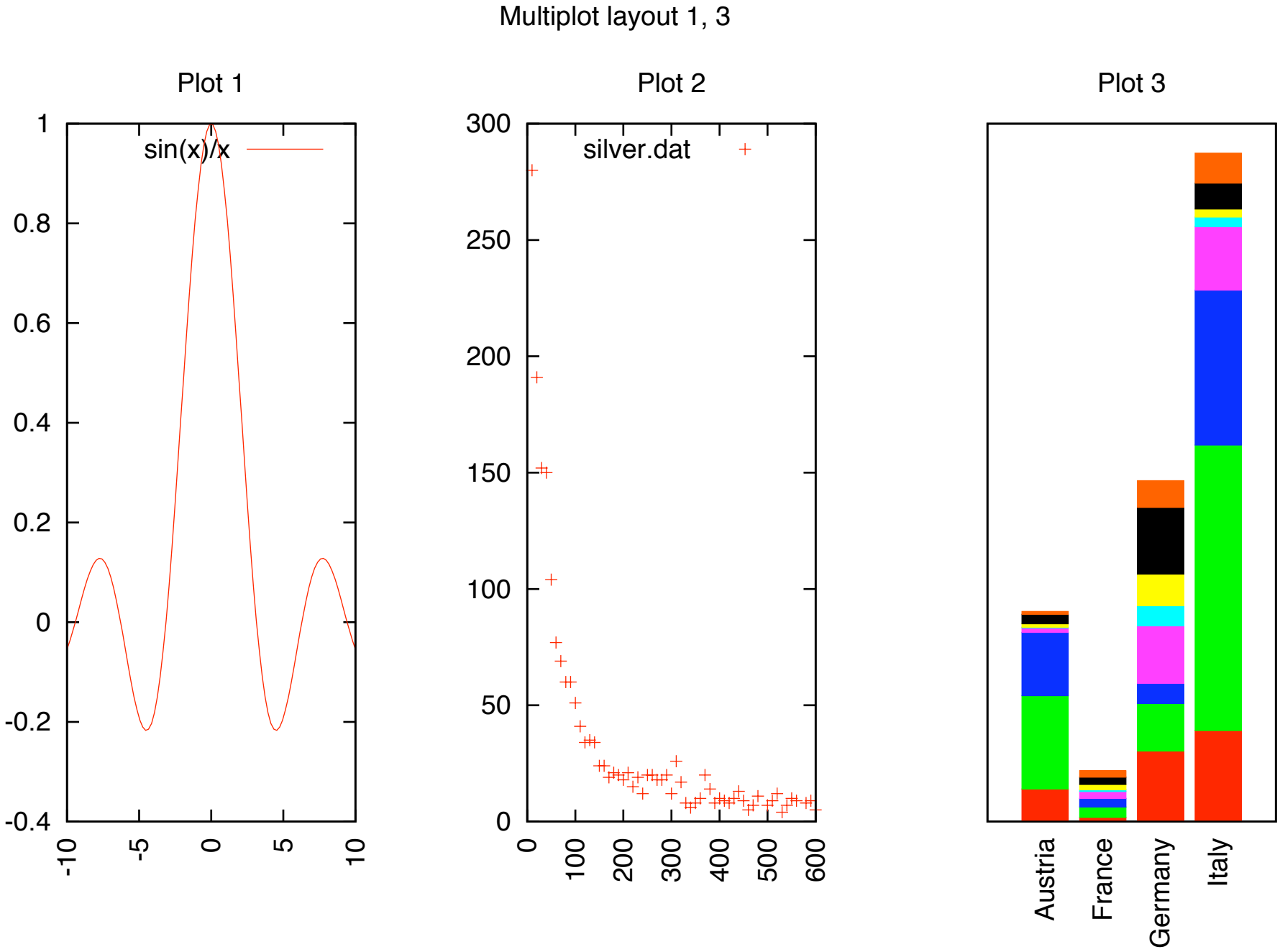

Second Order System Transfer Function - Magnitude
Damping =.1,.2,.3,.4,.5,.707,1.0,2.0
Magnitude (dB)
20
10
0
-10
-20
-30
-40
0.1
1
10
Frequency (w/wn)


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


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

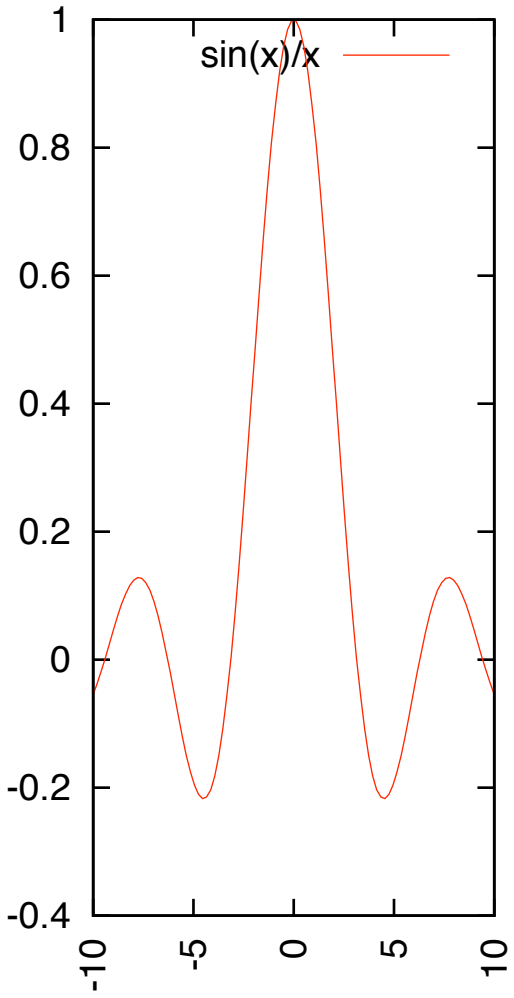

Second Order System - Unit Impulse Response
Amplitude y(wnt)
1
0.5
0
-0.5
-1
0
5
10
15
20
Normalized Time (wnt)


Multiplot layout 3, 1
Plot 1
sin(x)/x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10
Plot 2
silver.dat
300
250
200
150
100
50
0
0
100
200
300
400
500
600
Plot 3
Austria
France
Germany
Italy

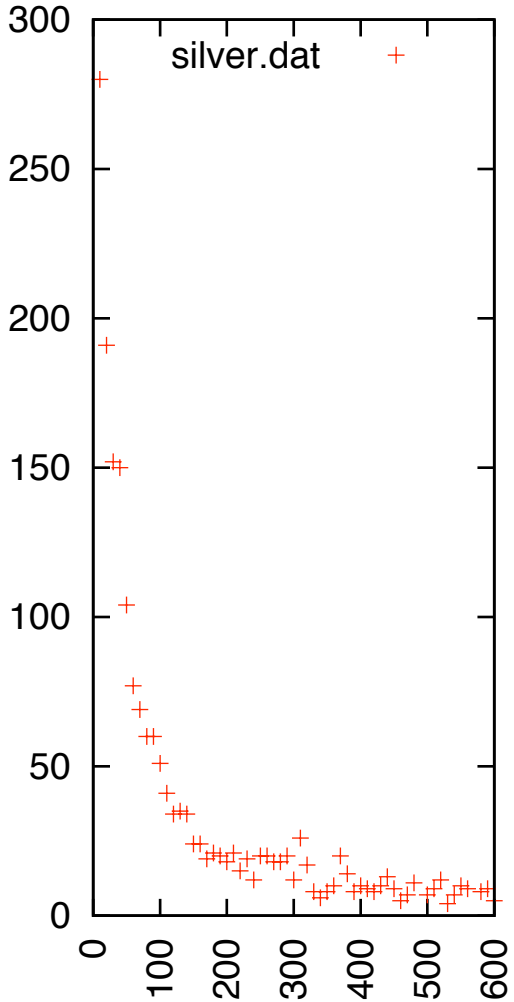

Multiplot layout 1, 3
Plot 1
sin(x)/x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10
Plot 2
silver.dat
300
250
200
150
100
50
0
0
100
200
300
400
500
600
Plot 3
Austria
France
Germany
Italy

Same plot with a multi-line title
showing adjustment of plot area
to accommodate it

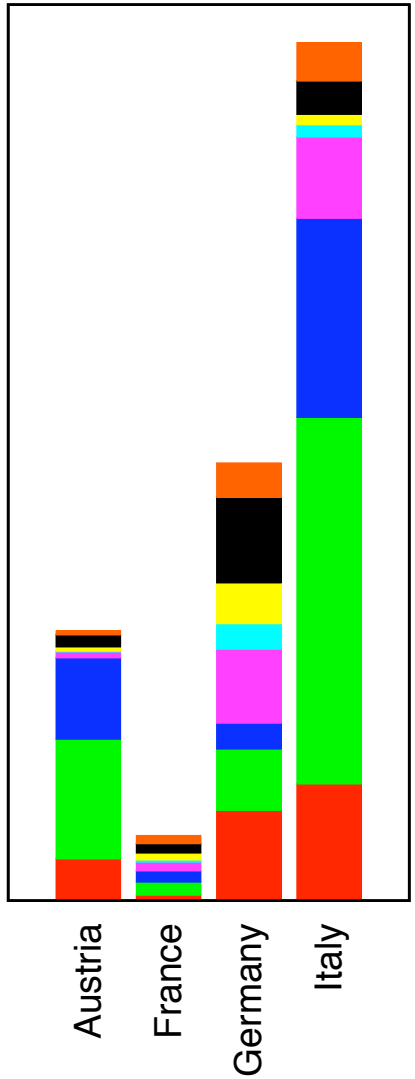
Plot 1

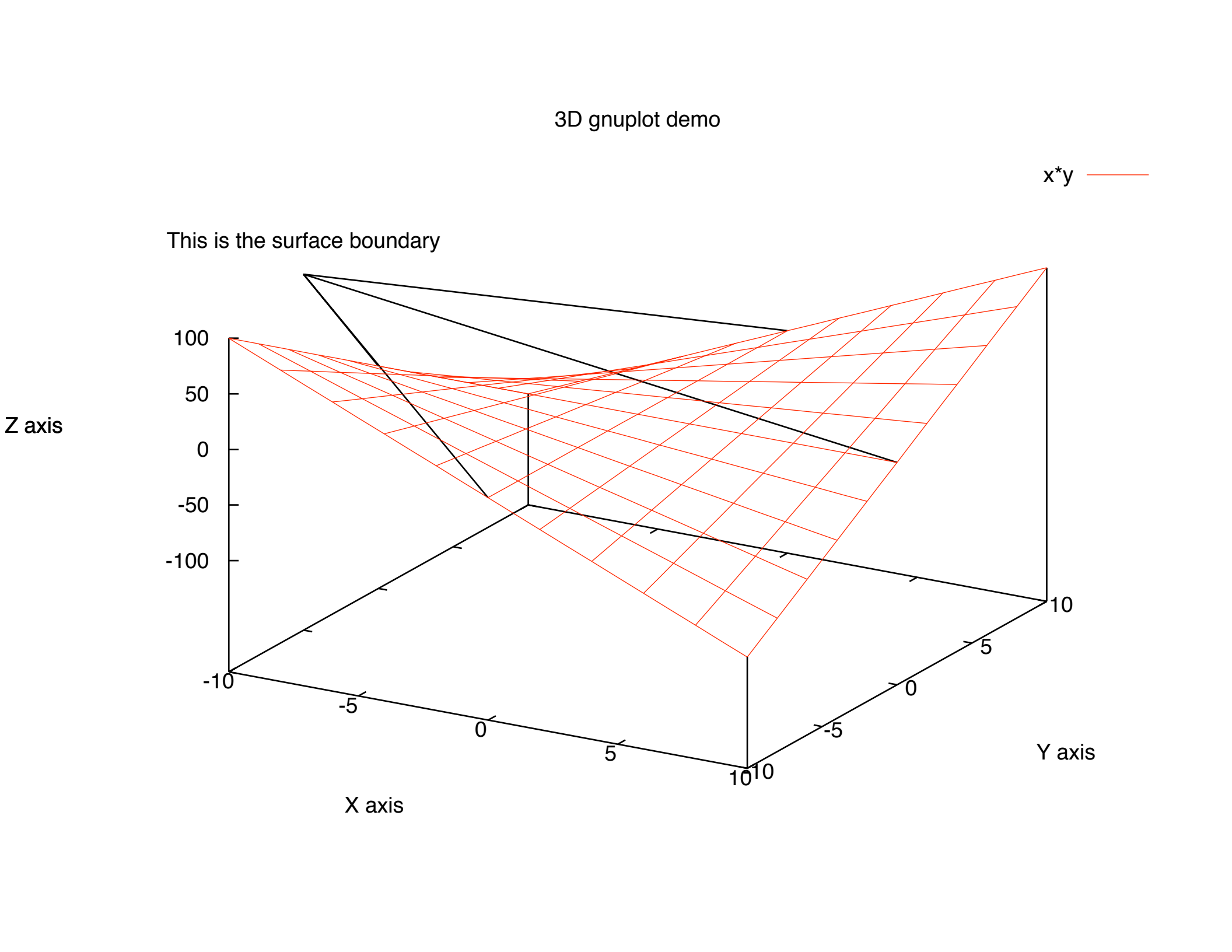


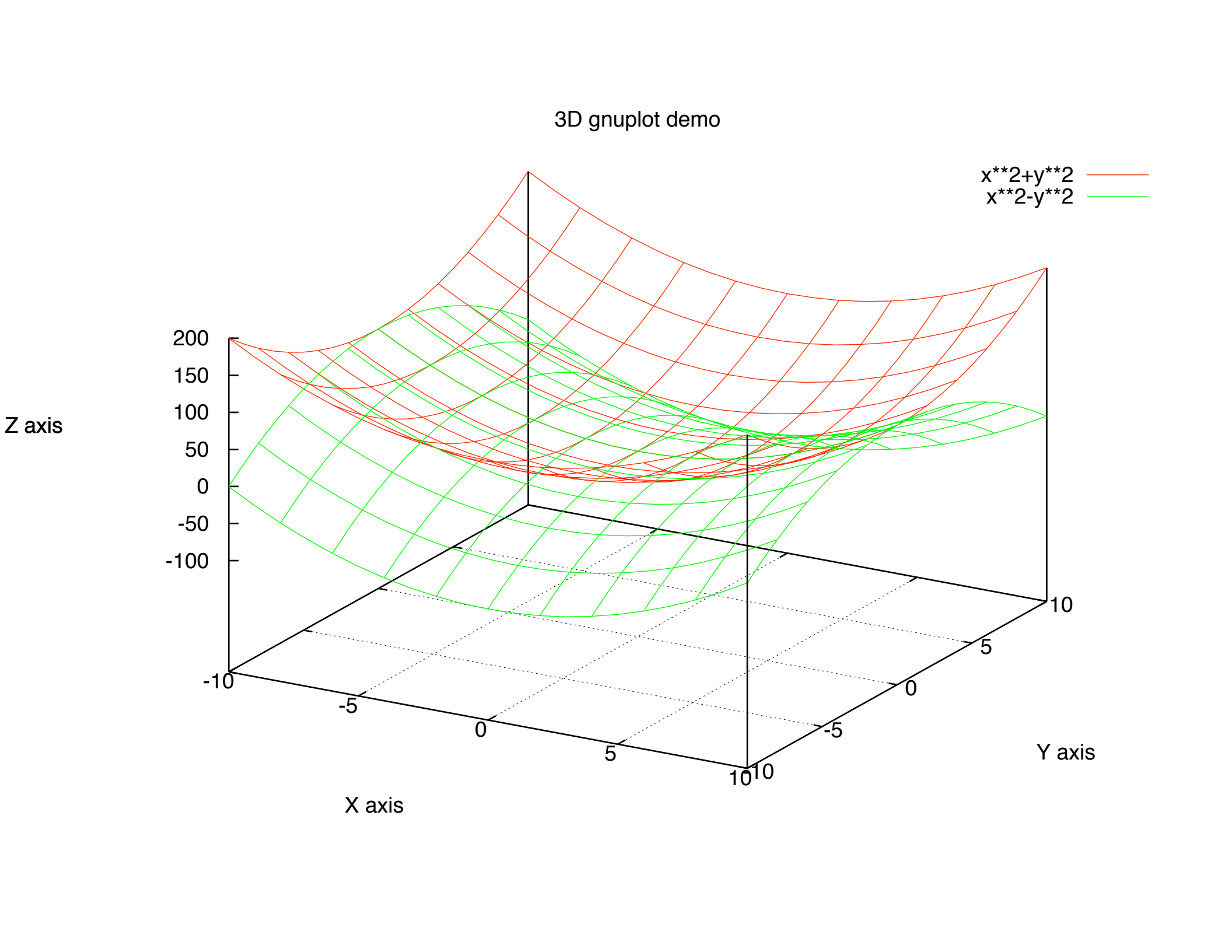
Plot 2

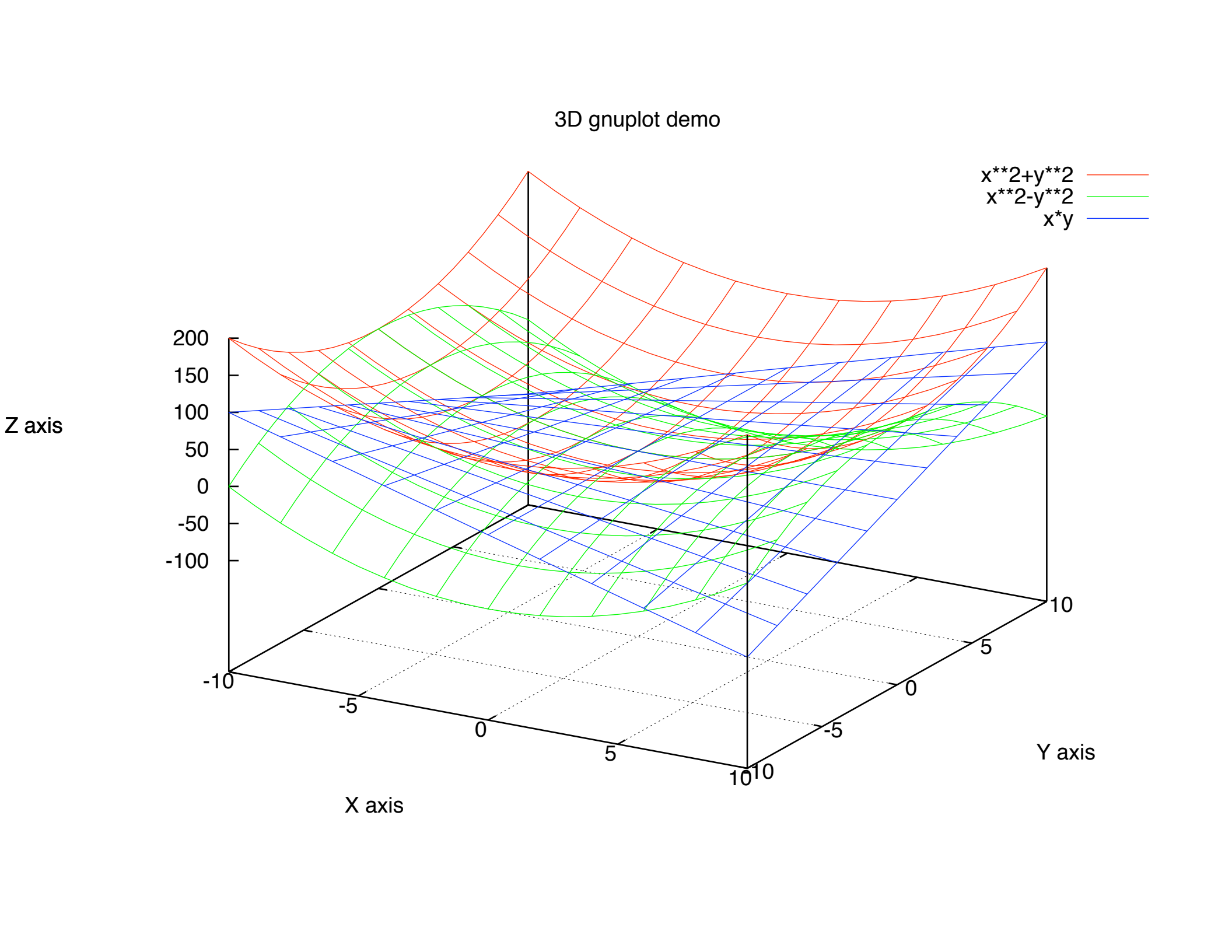


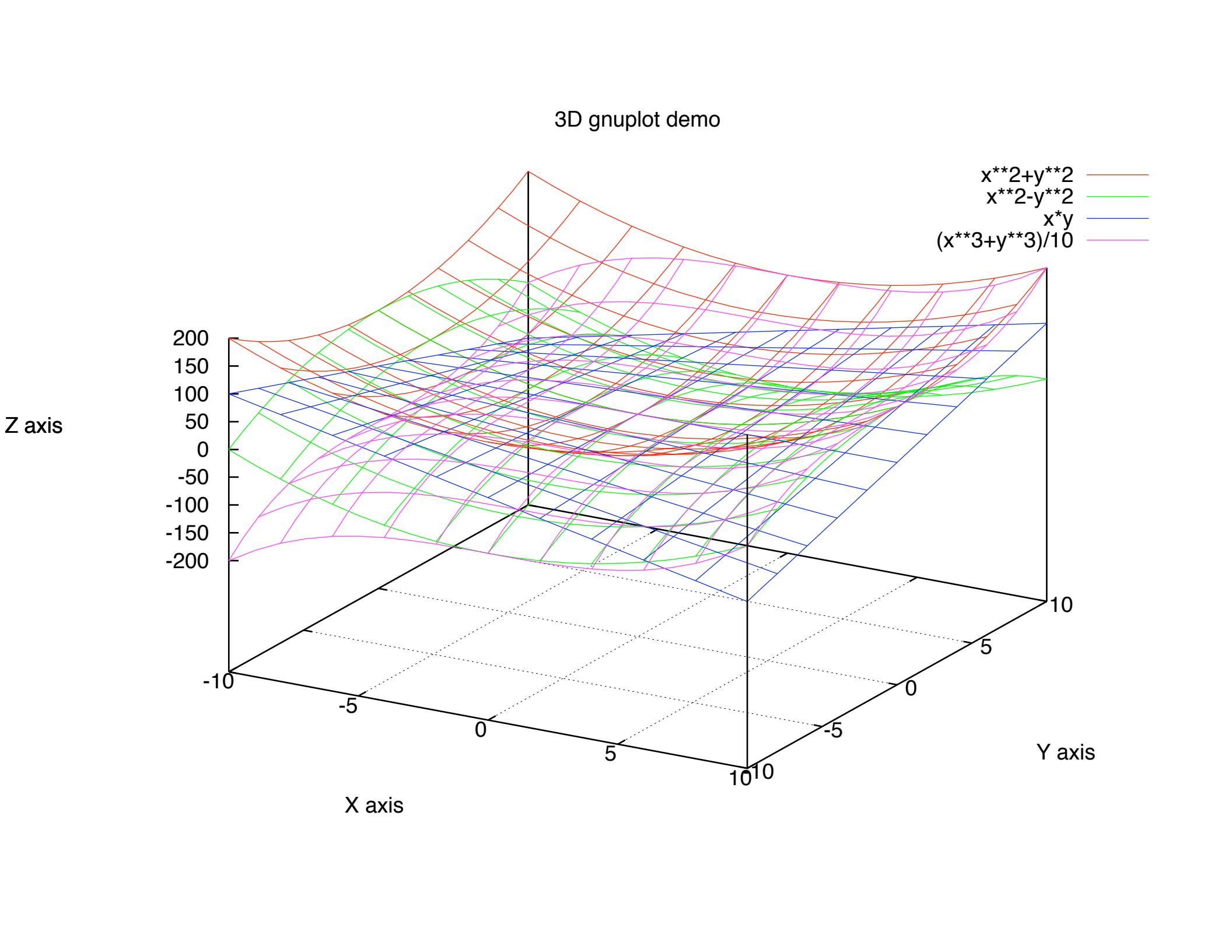
Plot 3

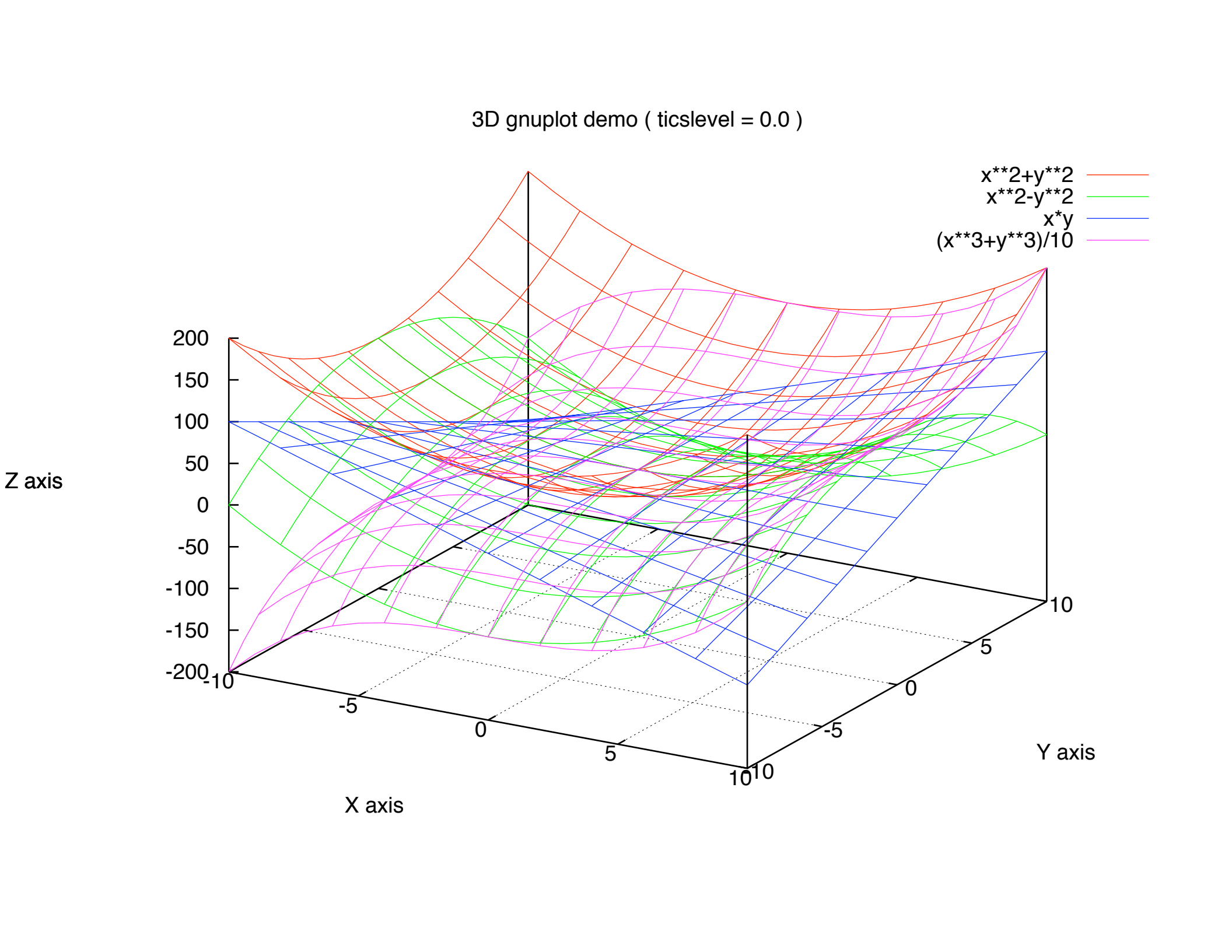


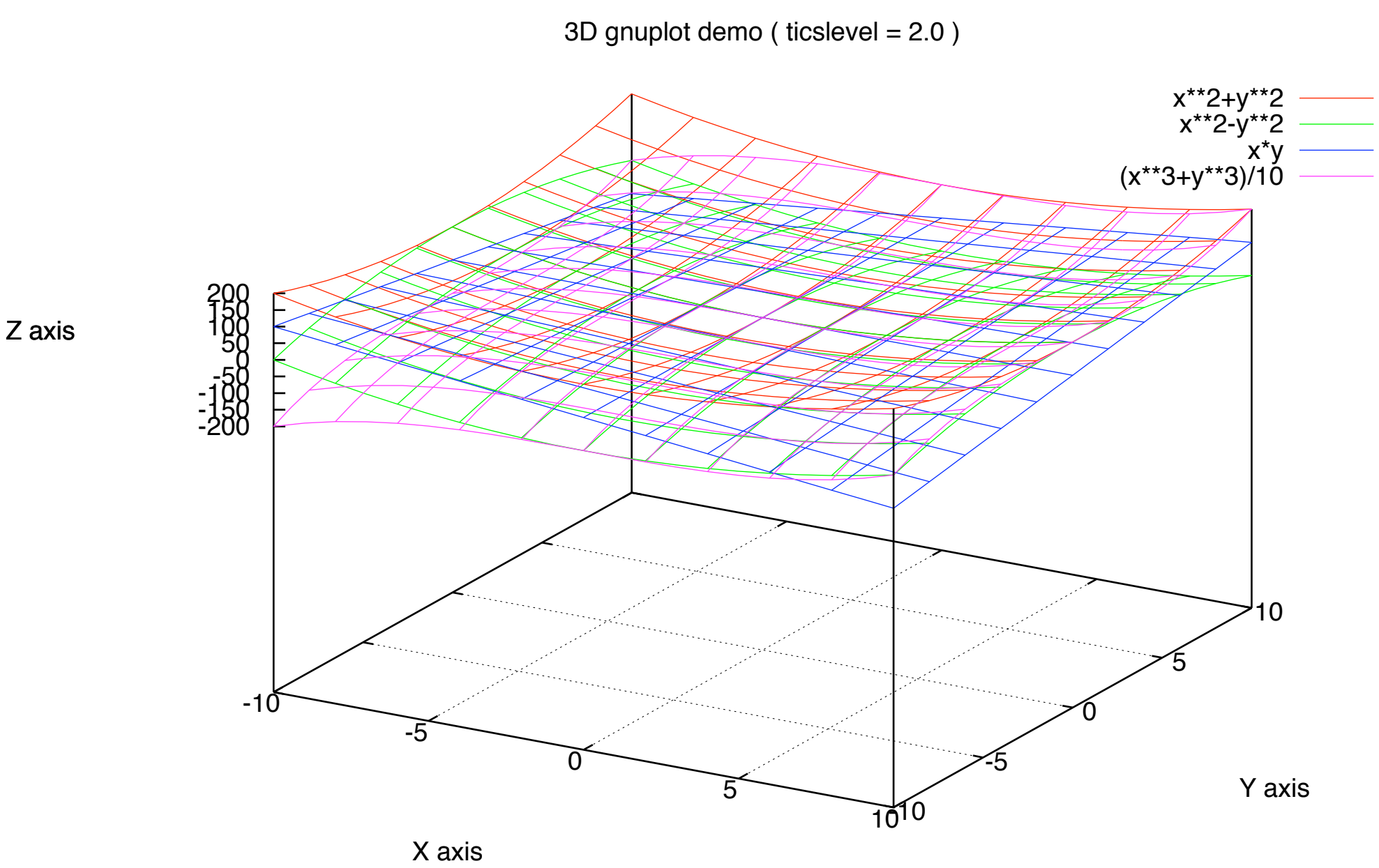

3D gnuplot demo
x*y
This is the surface boundary
Z axis
100
50
0
-50
-100
-10
-5
0
5
10
X axis
10
5
0
-5
-10
Y axis

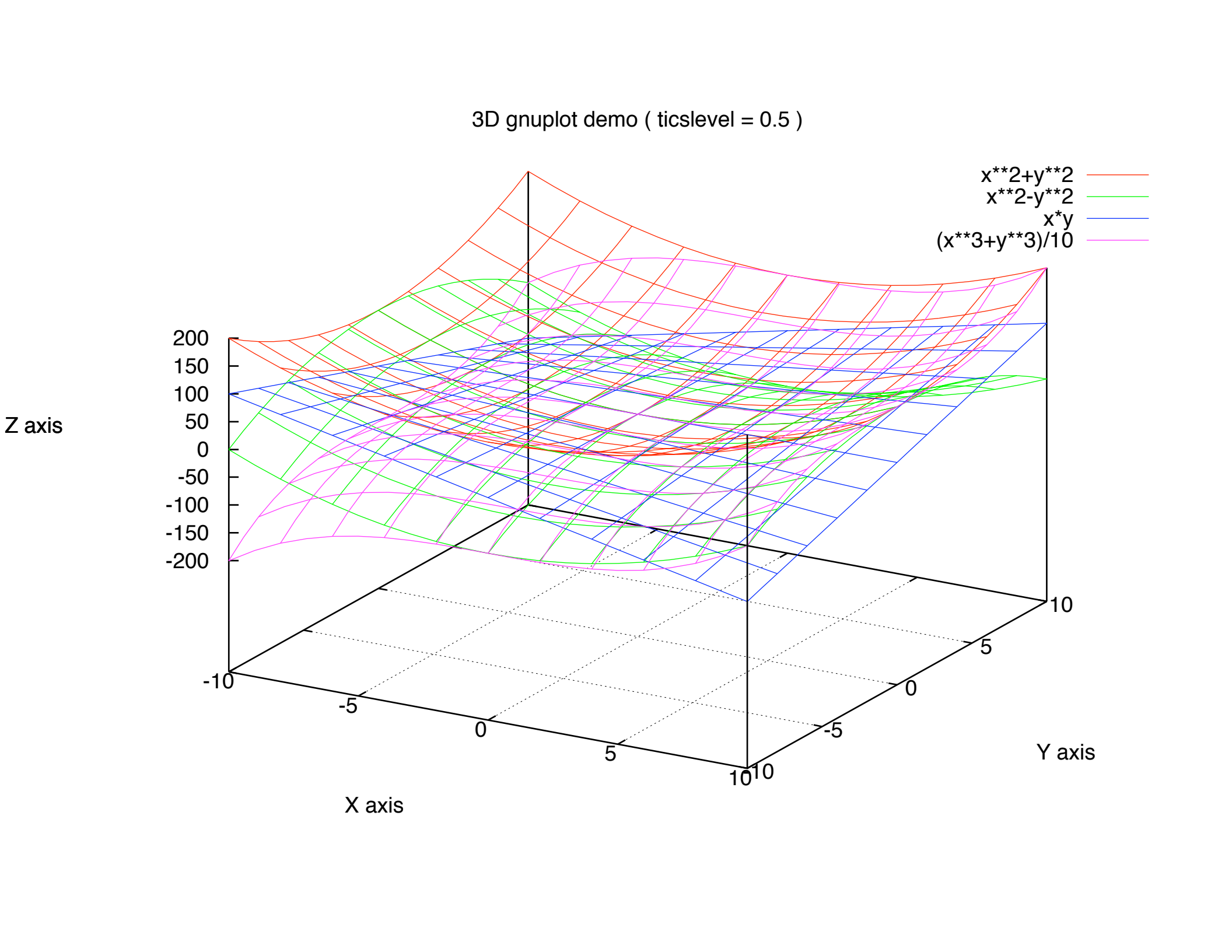

3D gnuplot demo
x**2+y**2
x**2-y**2
Z axis
200
150
100
50
0
-50
-100
-10
-5
0
5
10
X axis
10
5
0
-5
-10
Y axis

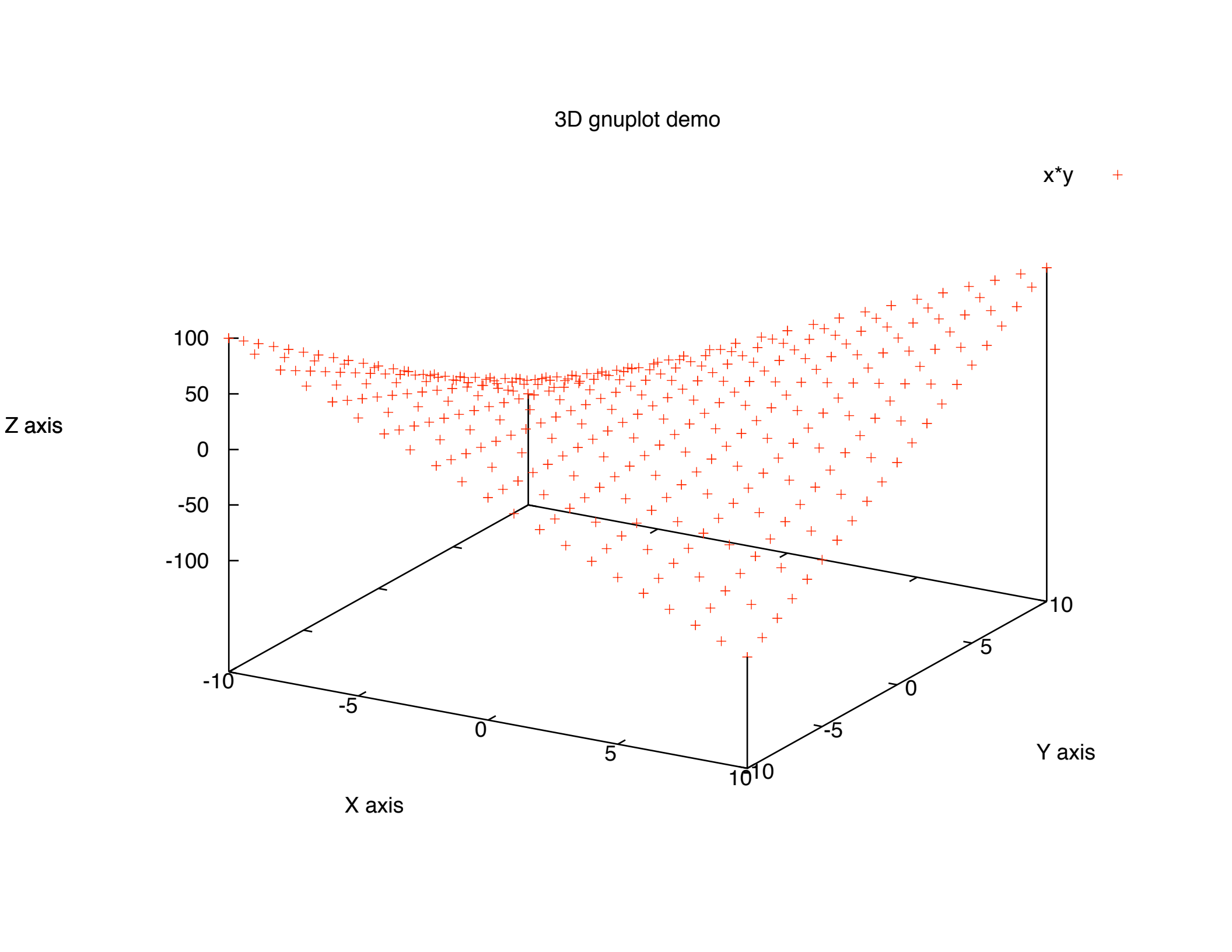

3D gnuplot demo
x**2+y**2
x**2-y**2
x*y
Z axis
200
150
100
50
0
-50
-100
-10
-5
0
5
10
X axis
10
5
0
-5
-10
Y axis

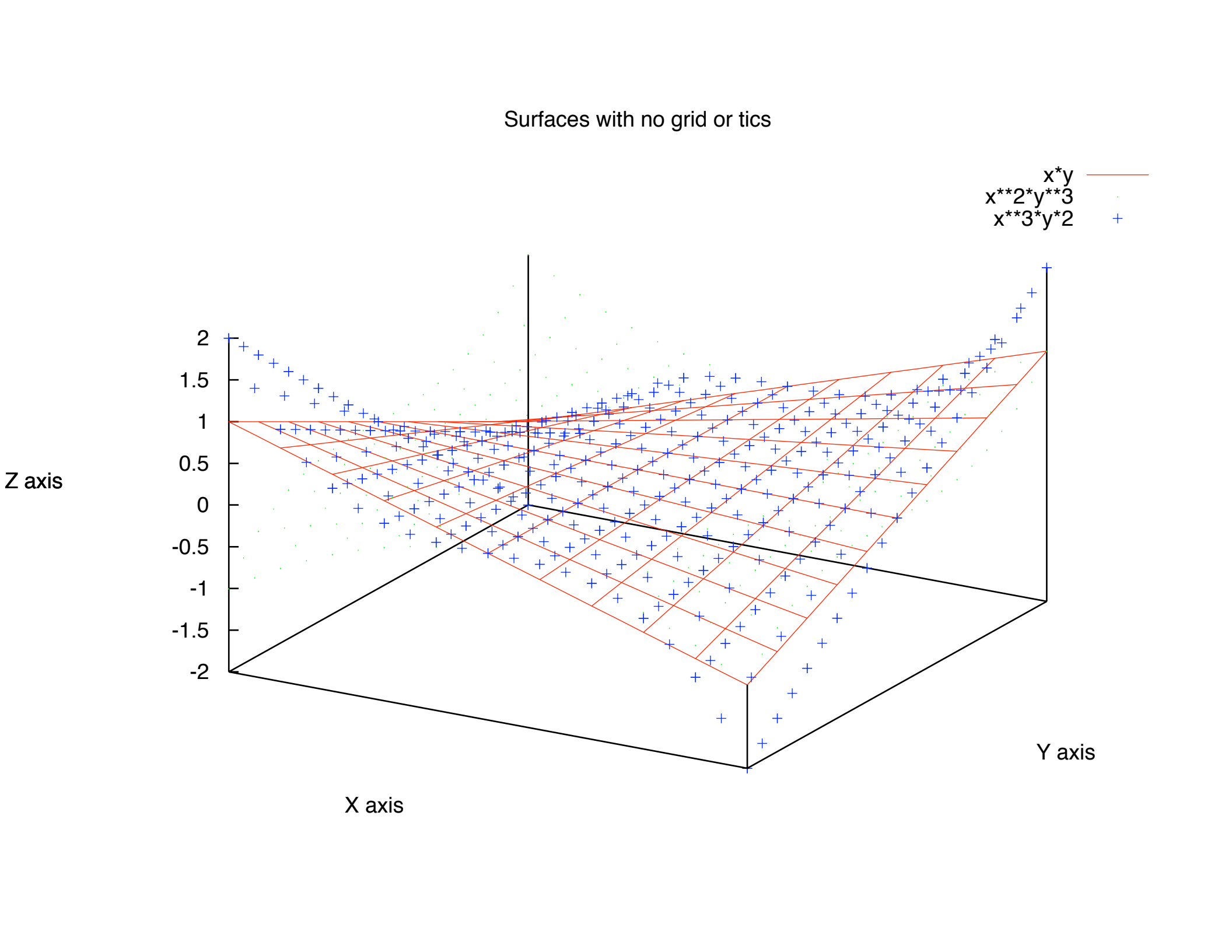

3D gnuplot demo
x**2+y**2
x**2-y**2
x*y
(x**3+y**3)/10
Z axis
200
150
100
50
0
-50
-100
-150
-200
-10
-5
0
5
10
X axis
10
5
0
-5
-10
Y axis

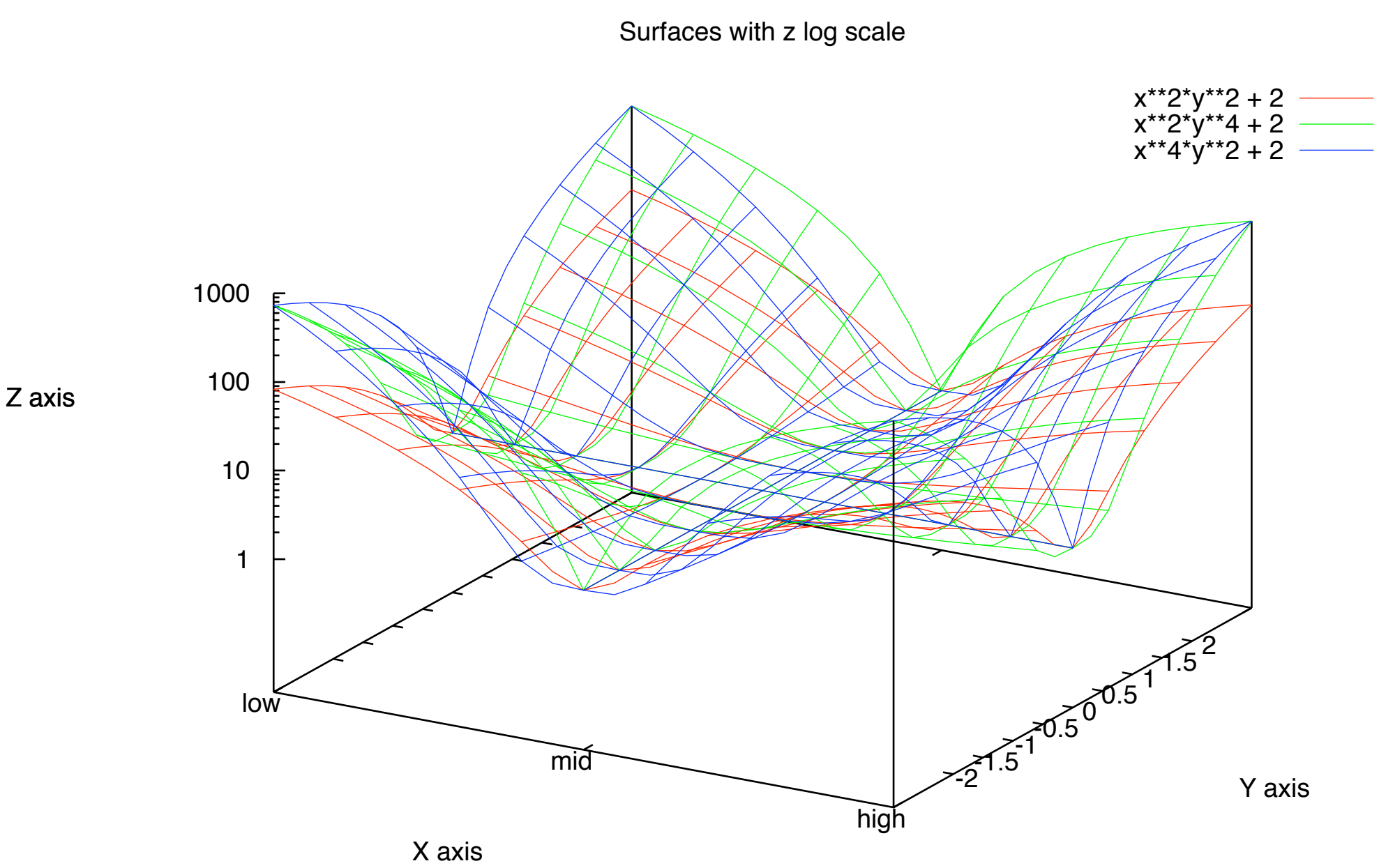

3D gnuplot demo (ticslevel = 0.0)
x**2+y**2
x**2-y**2
x*y
(x**3+y**3)/10
Z axis
200
150
100
50
0
-50
-100
-150
-200
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

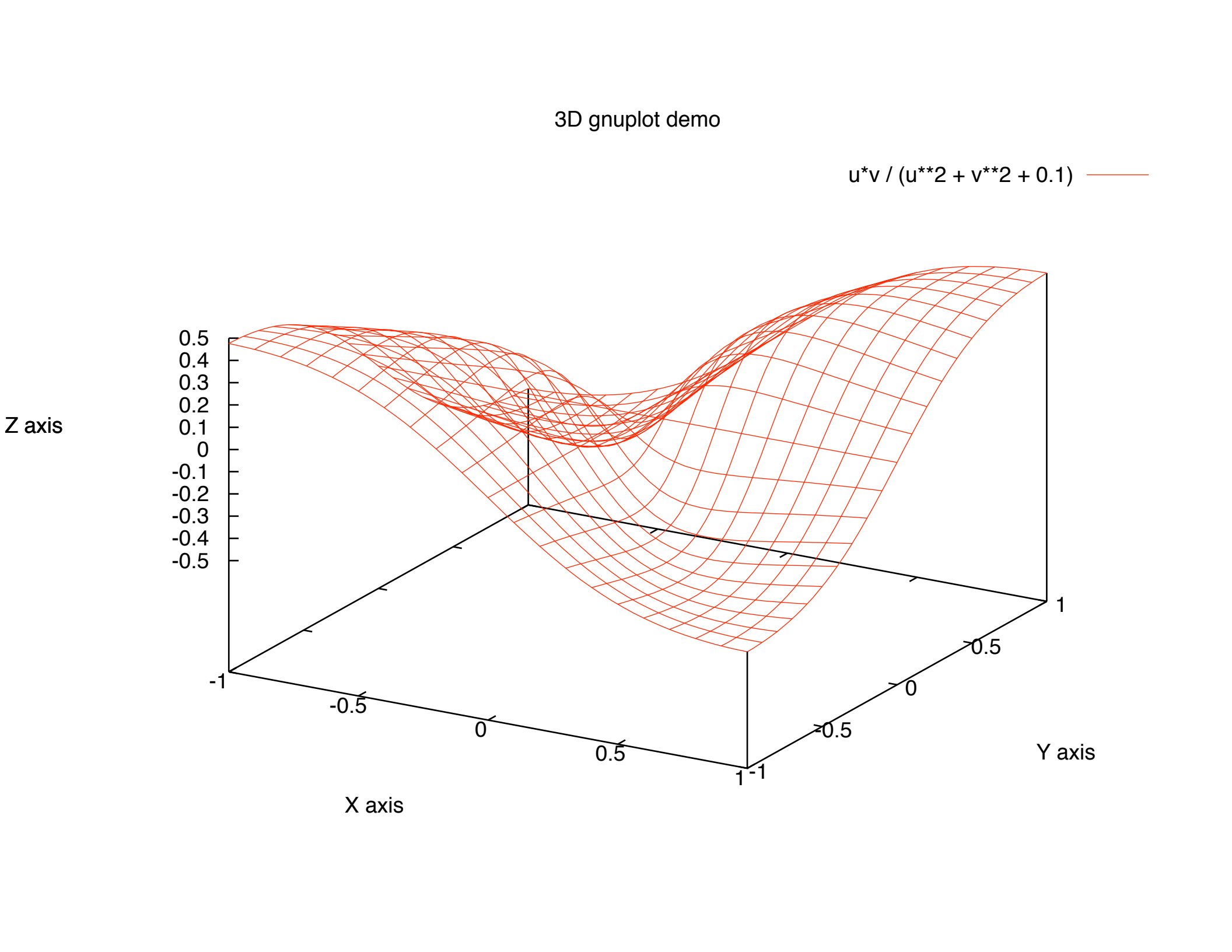

3D gnuplot demo (ticslevel = 2.0)
x**2+y**2
x**2-y**2
x*y
(x**3+y**3)/10
Z axis
200
150
100
50
0
-50
-100
-150
-200
-10
-5
0
5
10
X axis
10
5
0
-5
-10
Y axis

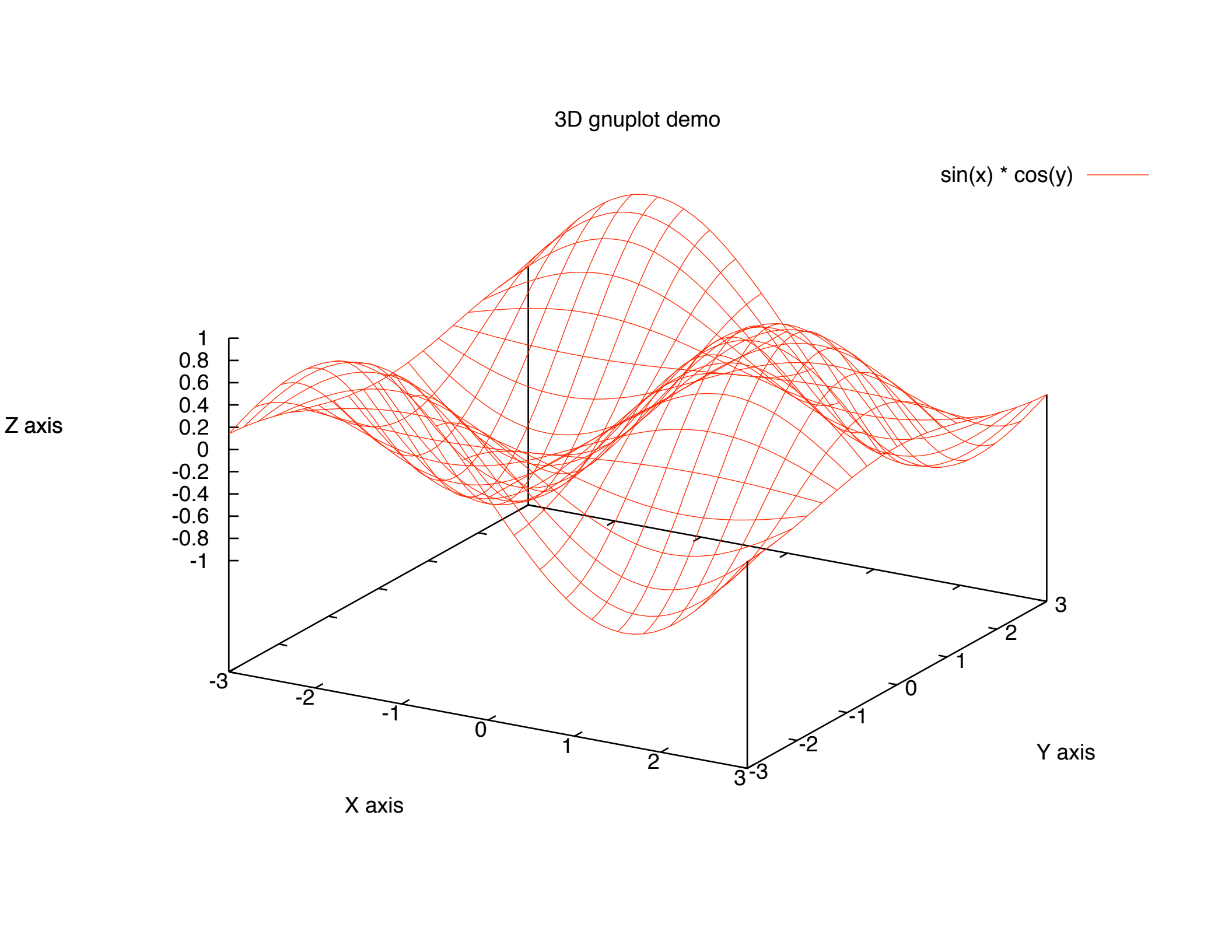

3D gnuplot demo (ticslevel = 0.5)
x**2+y**2
x**2-y**2
x*y
(x**3+y**3)/10
Z axis
200
150
100
50
0
-50
-100
-150
-200
-10
-5
0
5
10
X axis
10
5
0
-5
-10
Y axis

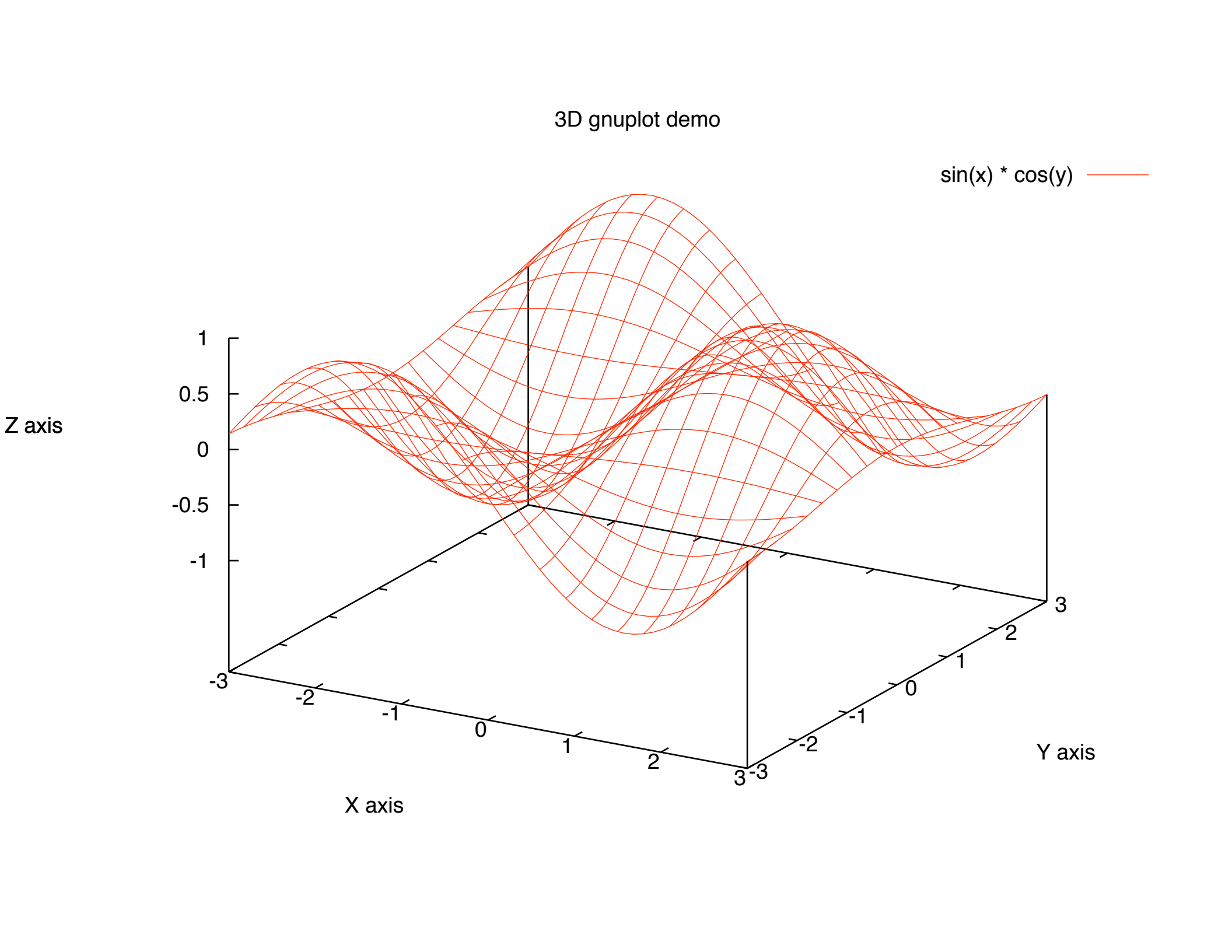

3D gnuplot demo
x*y
Z axis
100
50
0
-50
-100
-10
-5
0
5
10
X axis
10
5
0
-5
-10
Y axis

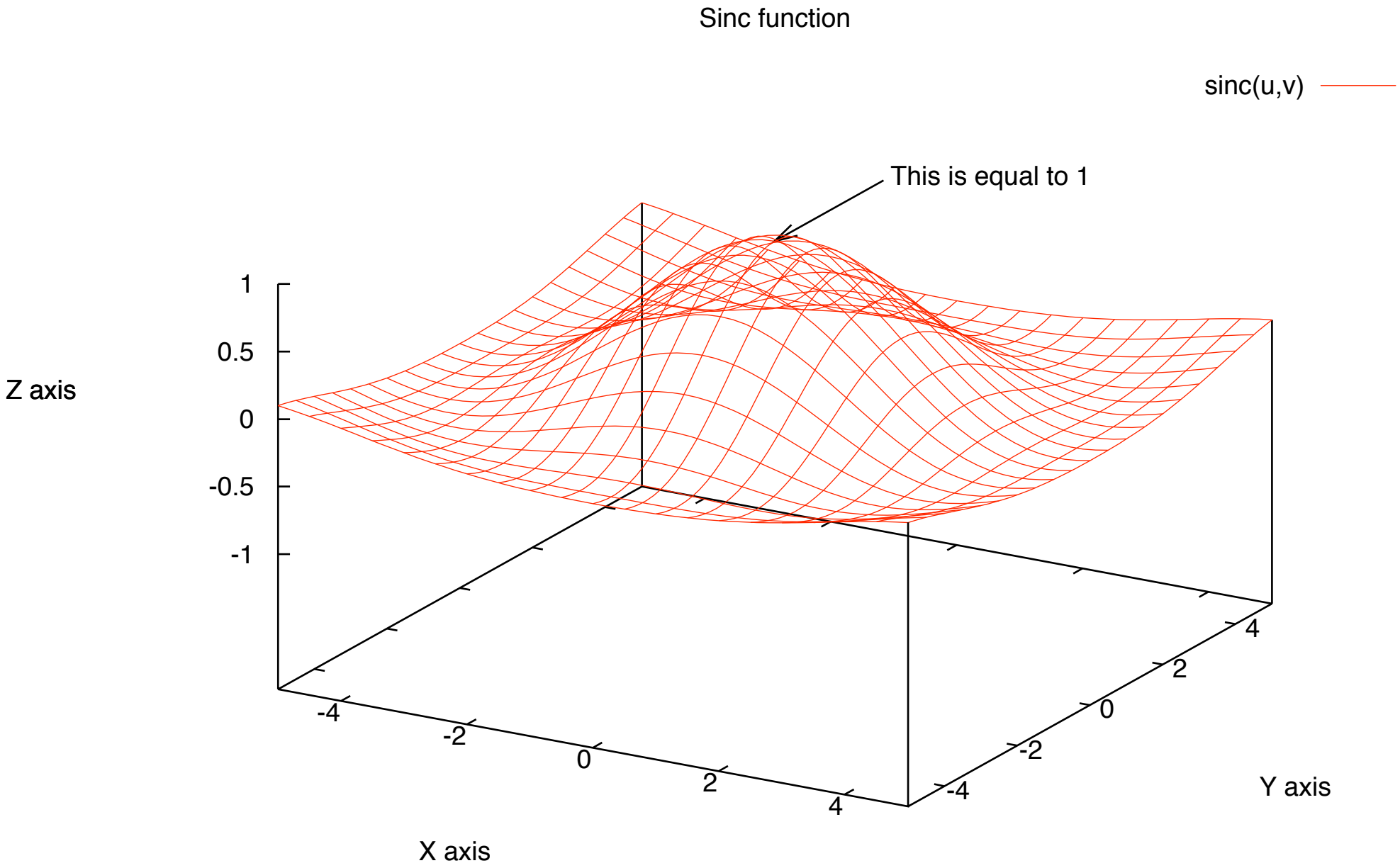

Surfaces with no grid or tics
x*y
x**2*y**3
x**3*y*2
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
Z axis
X axis
Y axis

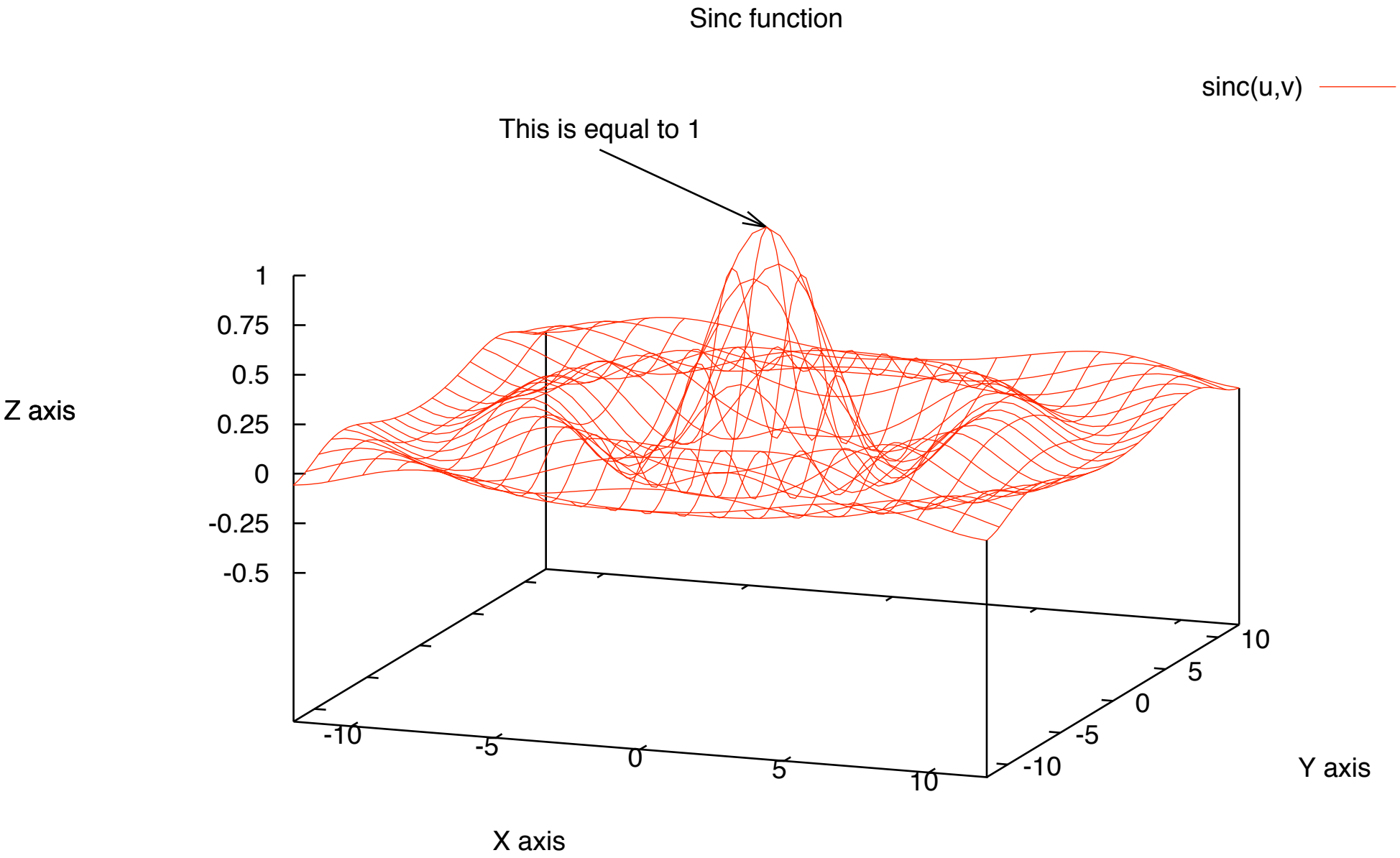

Surfaces with z log scale
x**2*y**2 + 2
x**2*y**4 + 2
x**4*y**2 + 2
1000
100
10
1
Z axis
low
mid
high
X axis
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
Y axis

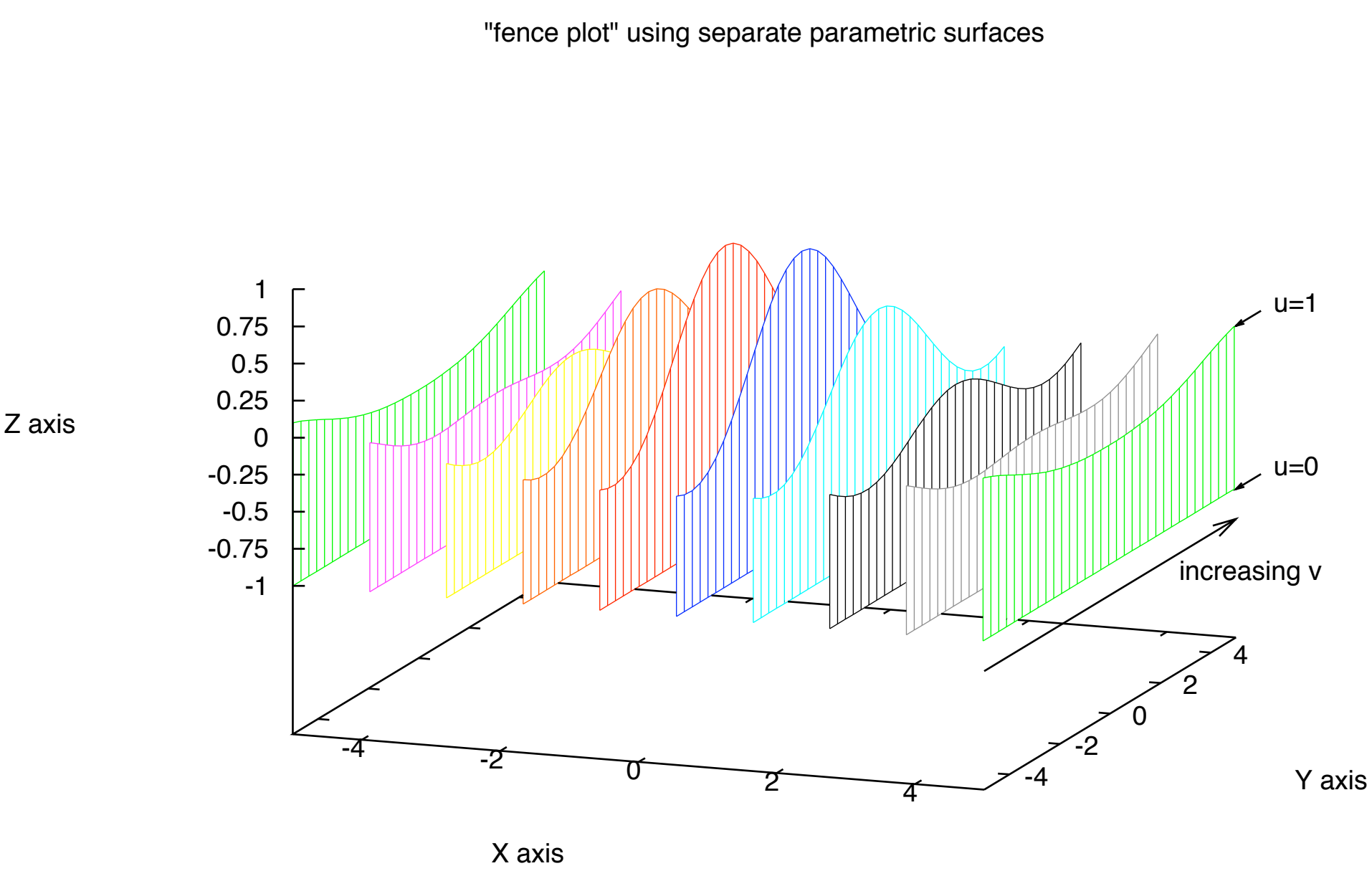

3D gnuplot demo
u*v / (u**2 + v**2 + 0.1)
Z axis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1
-0.5
0
0.5
1
X axis
-1
-0.5
0
0.5
1
Y axis

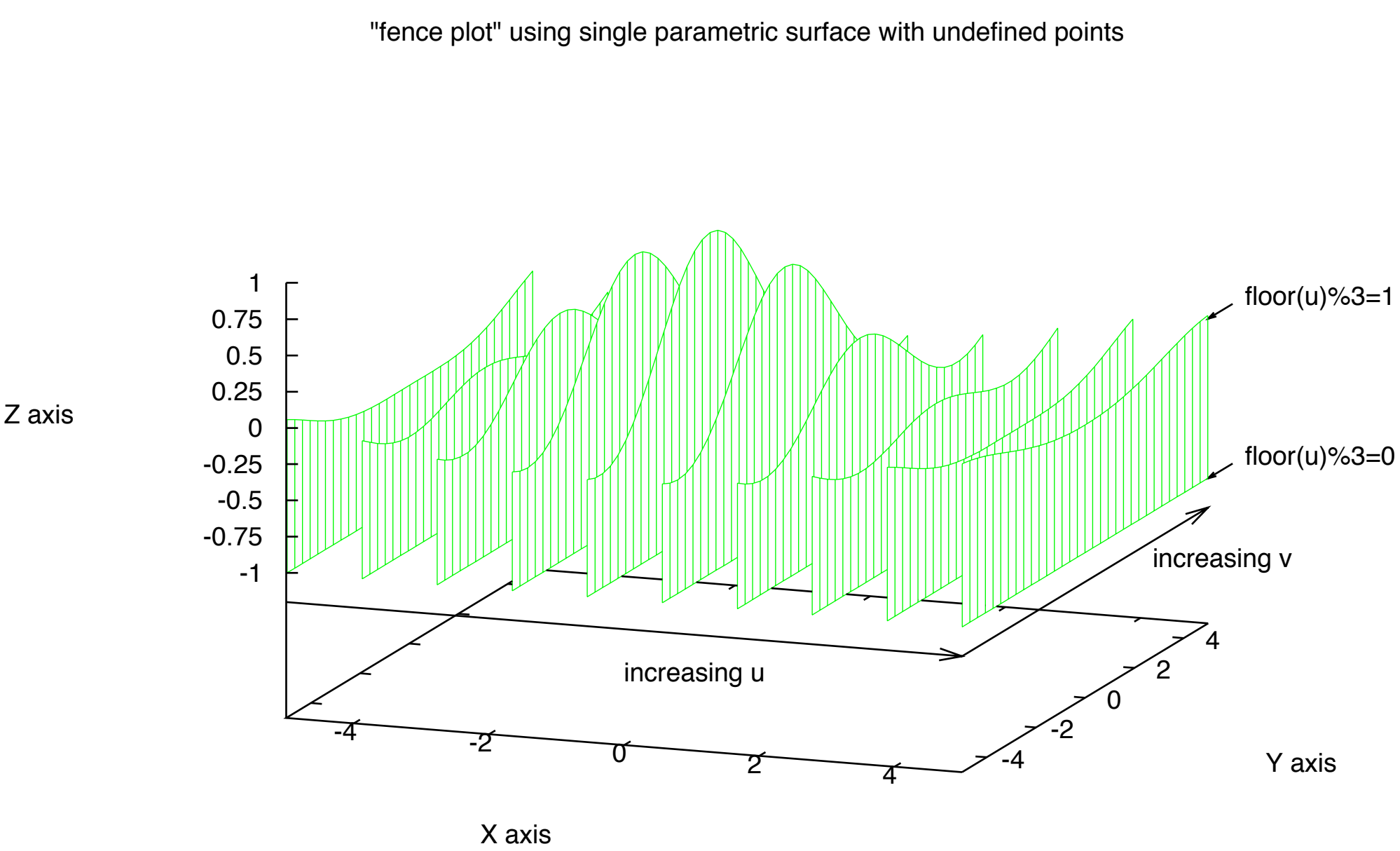

3D gnuplot demo
sin(x) * cos(y)
Z axis
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-3
-2
-1
0
1
2
3
X axis
-3
-2
-1
0
1
2
3
Y axis

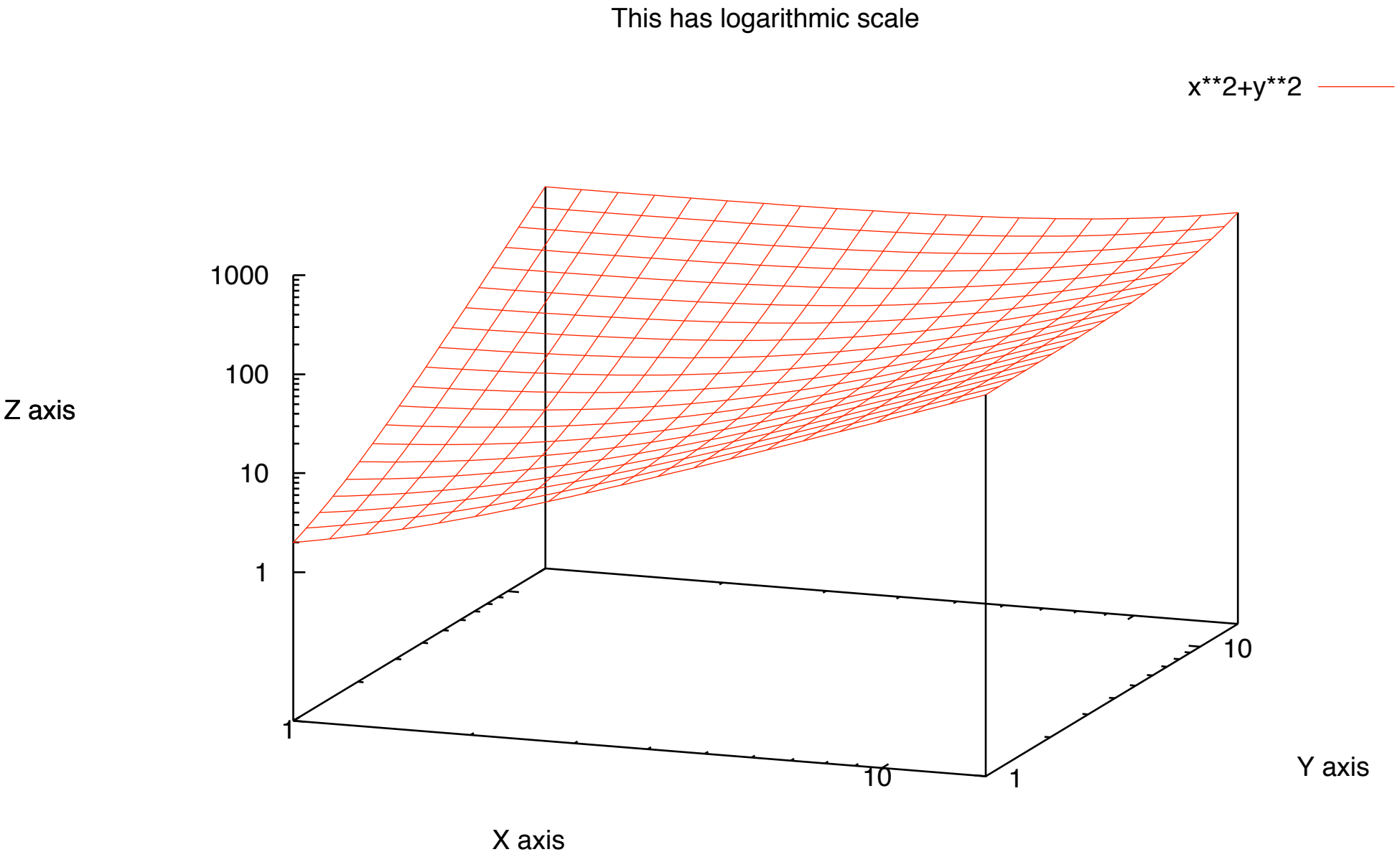

3D gnuplot demo
sin(x) * cos(y)
Z axis
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3
X axis
-3
-2
-1
0
1
2
3
Y axis

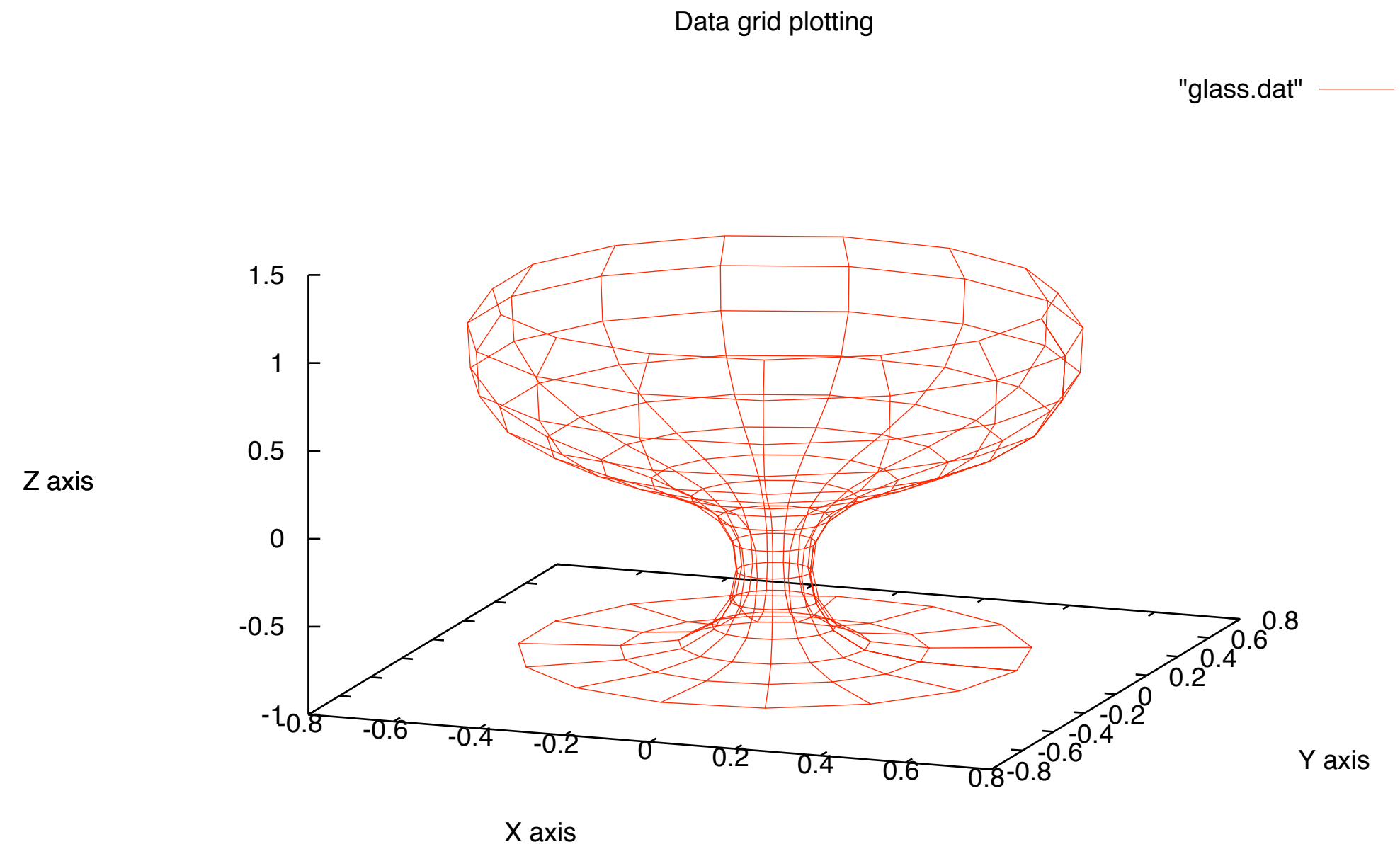

Sinc function
sinc(u,v)
This is equal to 1
Z axis
1
0.5
0
-0.5
-1
-4
-2
0
2
4
X axis
-4
-2
0
2
4
Y axis

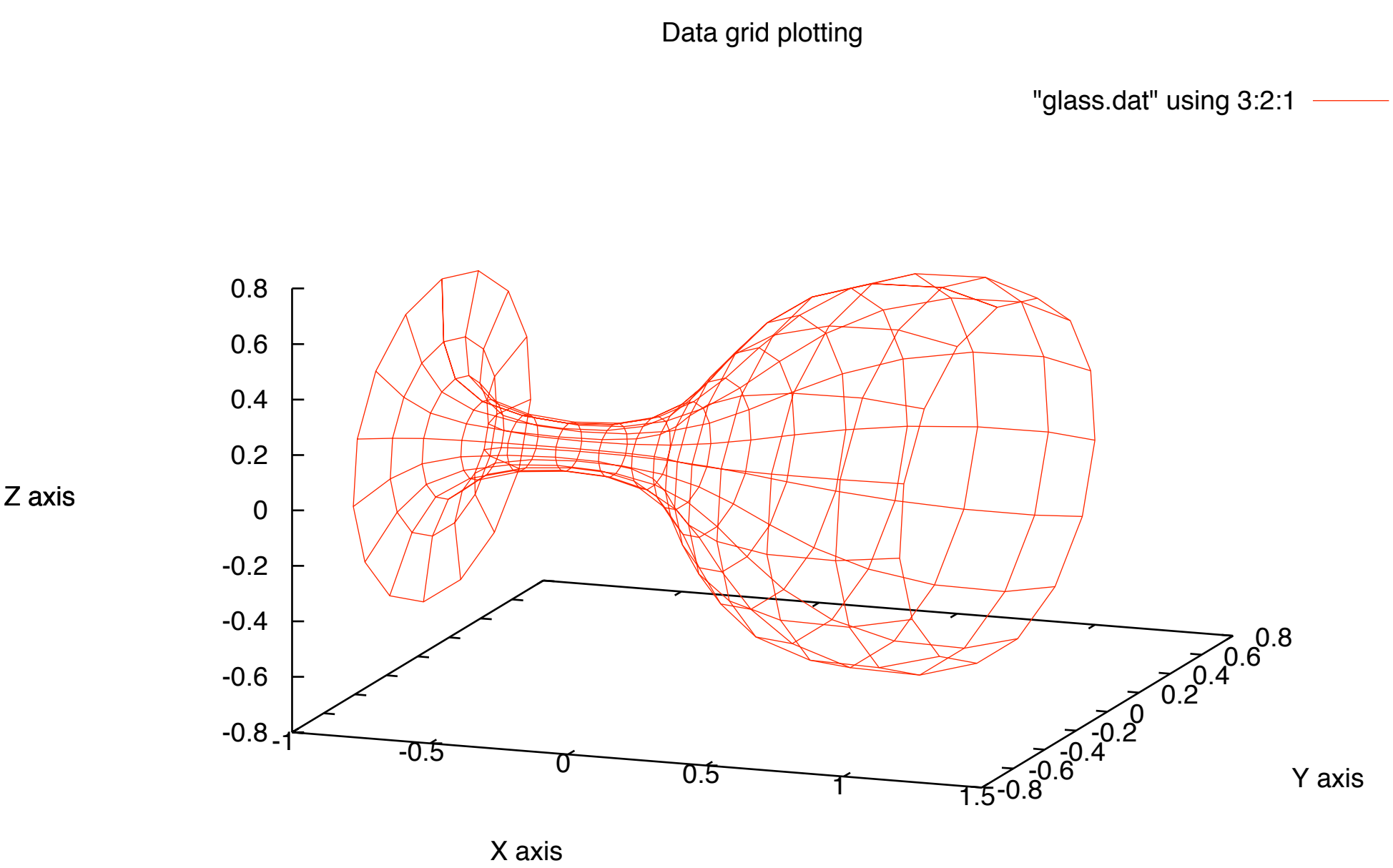

Sinc function
sinc(u,v)
This is equal to 1
Z axis
1
0.75
0.5
0.25
0
-0.25
-0.5
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

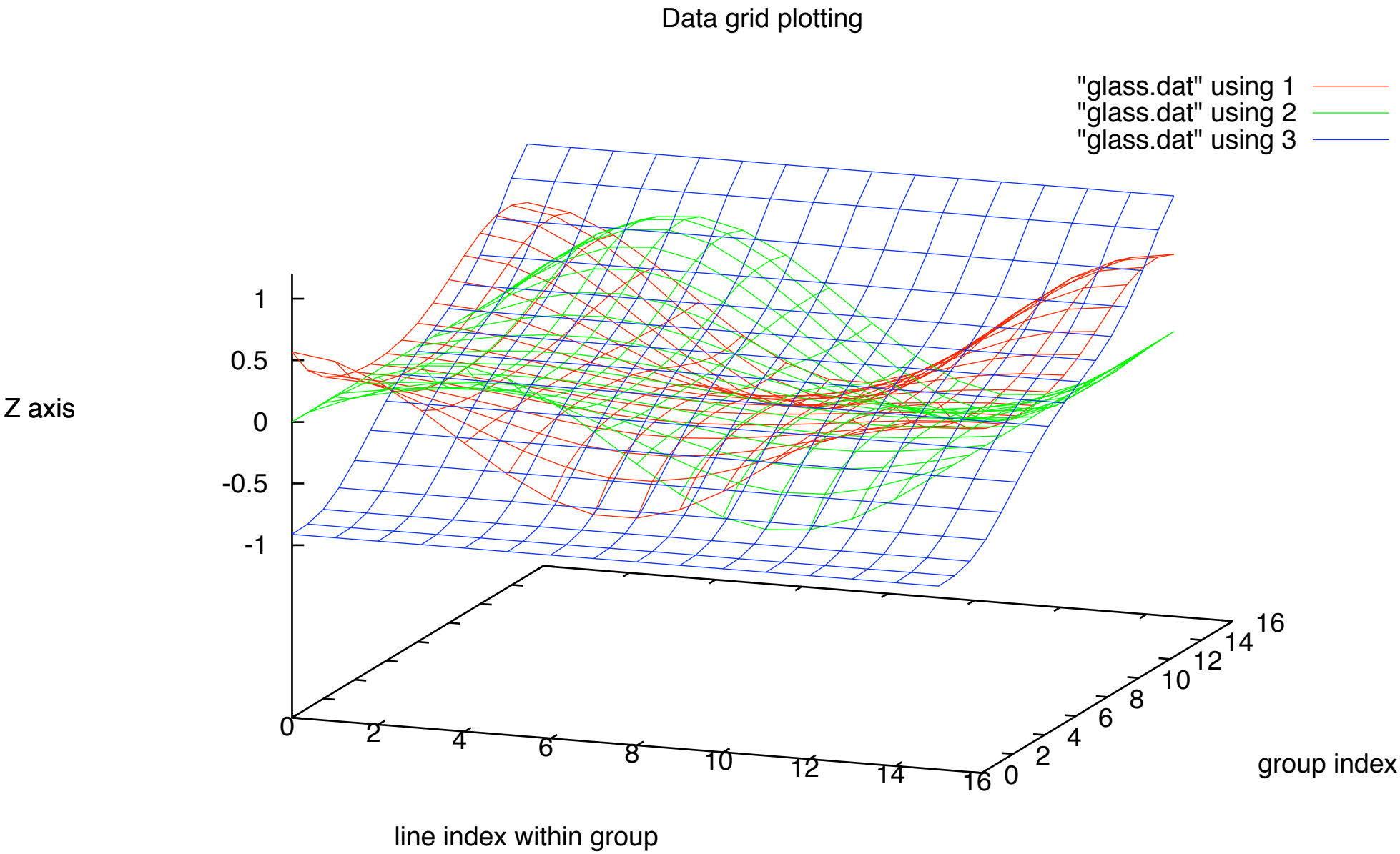

"fence plot" using separate parametric surfaces
Z axis
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1
u=1
u=0
increasing v
-4
-2
0
2
4
X axis
-4
-2
0
2
4
Y axis

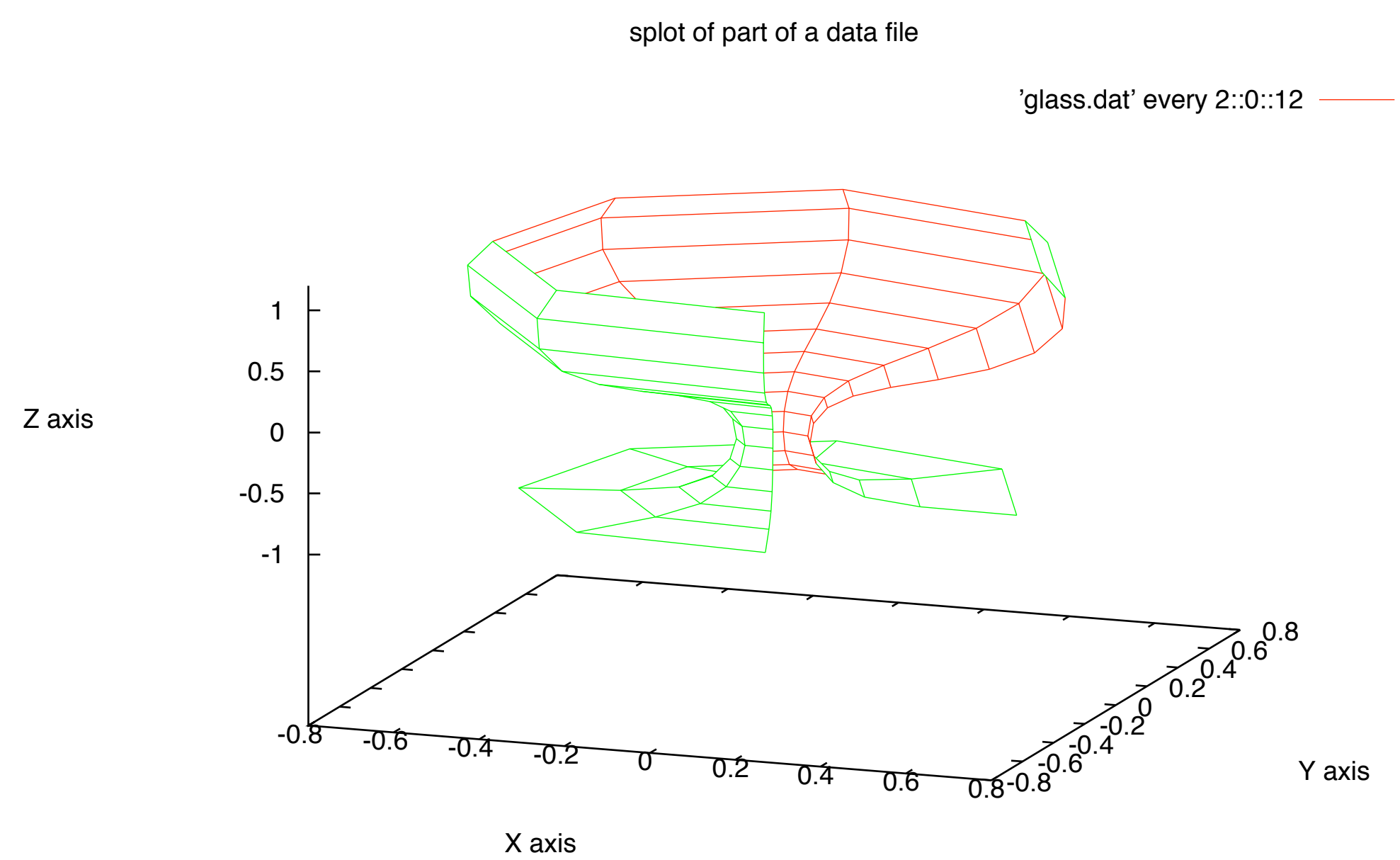

"fence plot" using single parametric surface with undefined points
Z axis
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1
floor(u)%3=1
floor(u)%3=0
increasing v
increasing u
-4
-2
0
2
4
-4
-2
0
2
4
X axis
Y axis

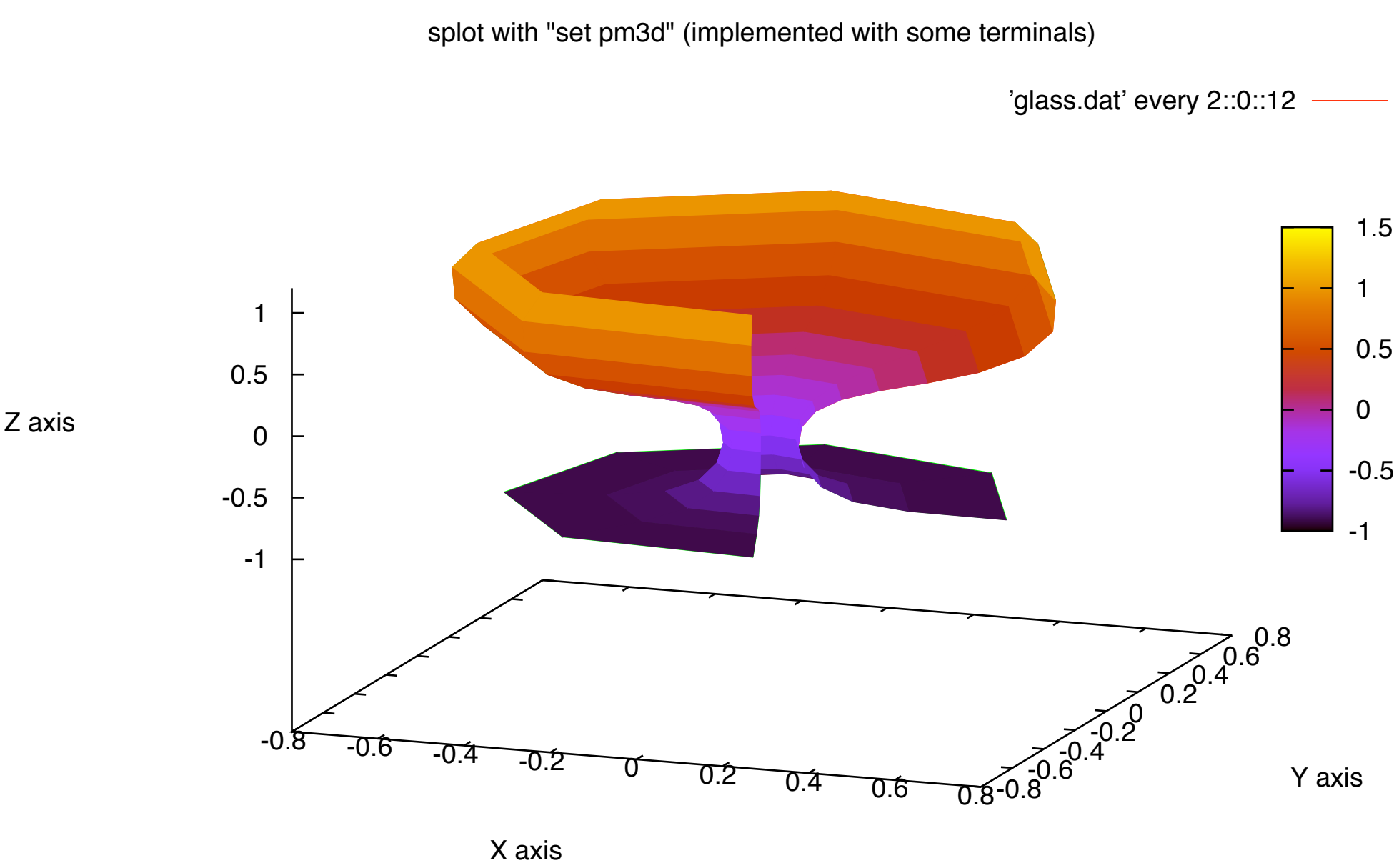

This has logarithmic scale
x**2+y**2
1000
100
10
1
Z axis
1
10
X axis
1
10
Y axis

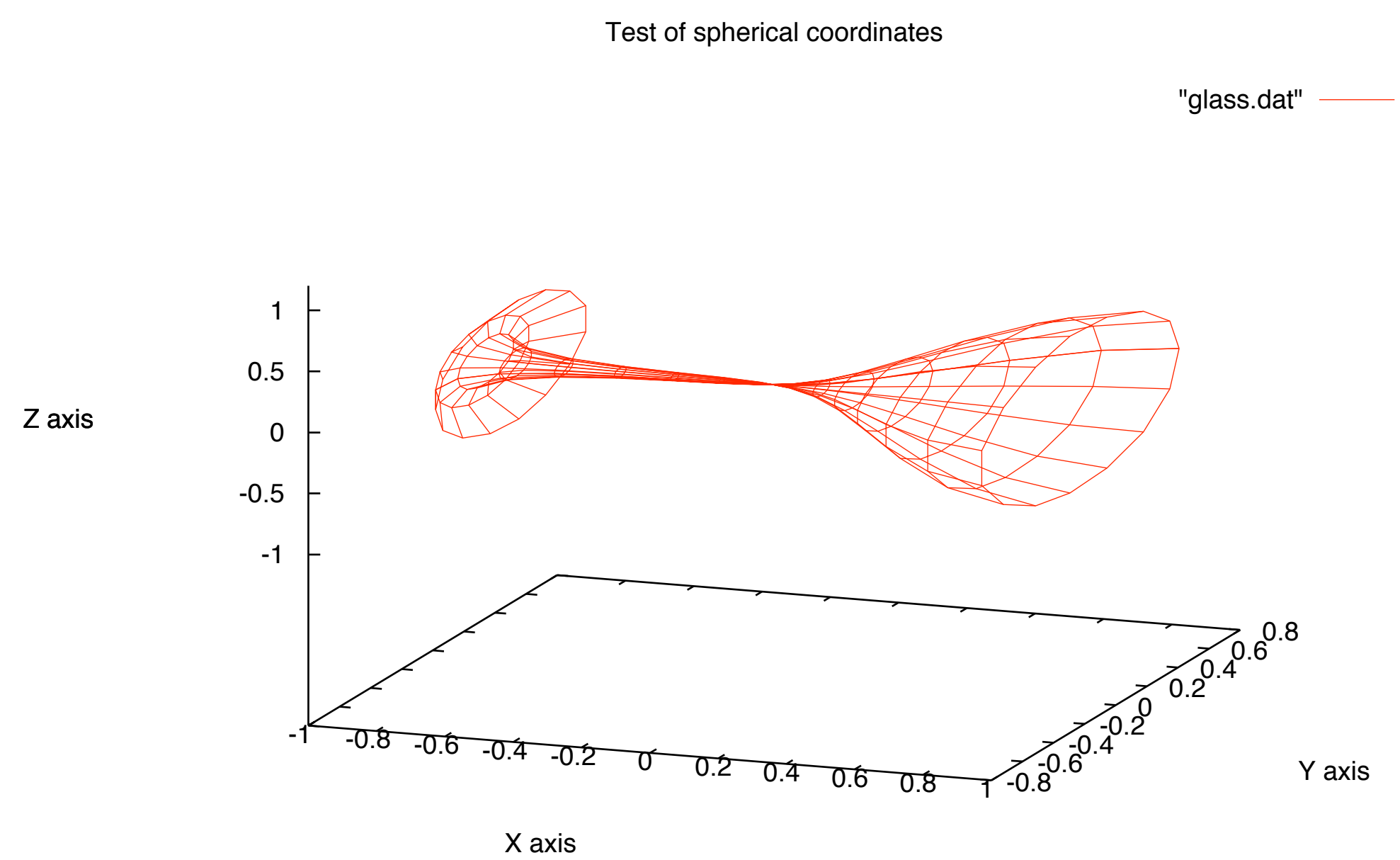

Data grid plotting
"glass.dat"
Z axis
1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
X axis
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
Y axis

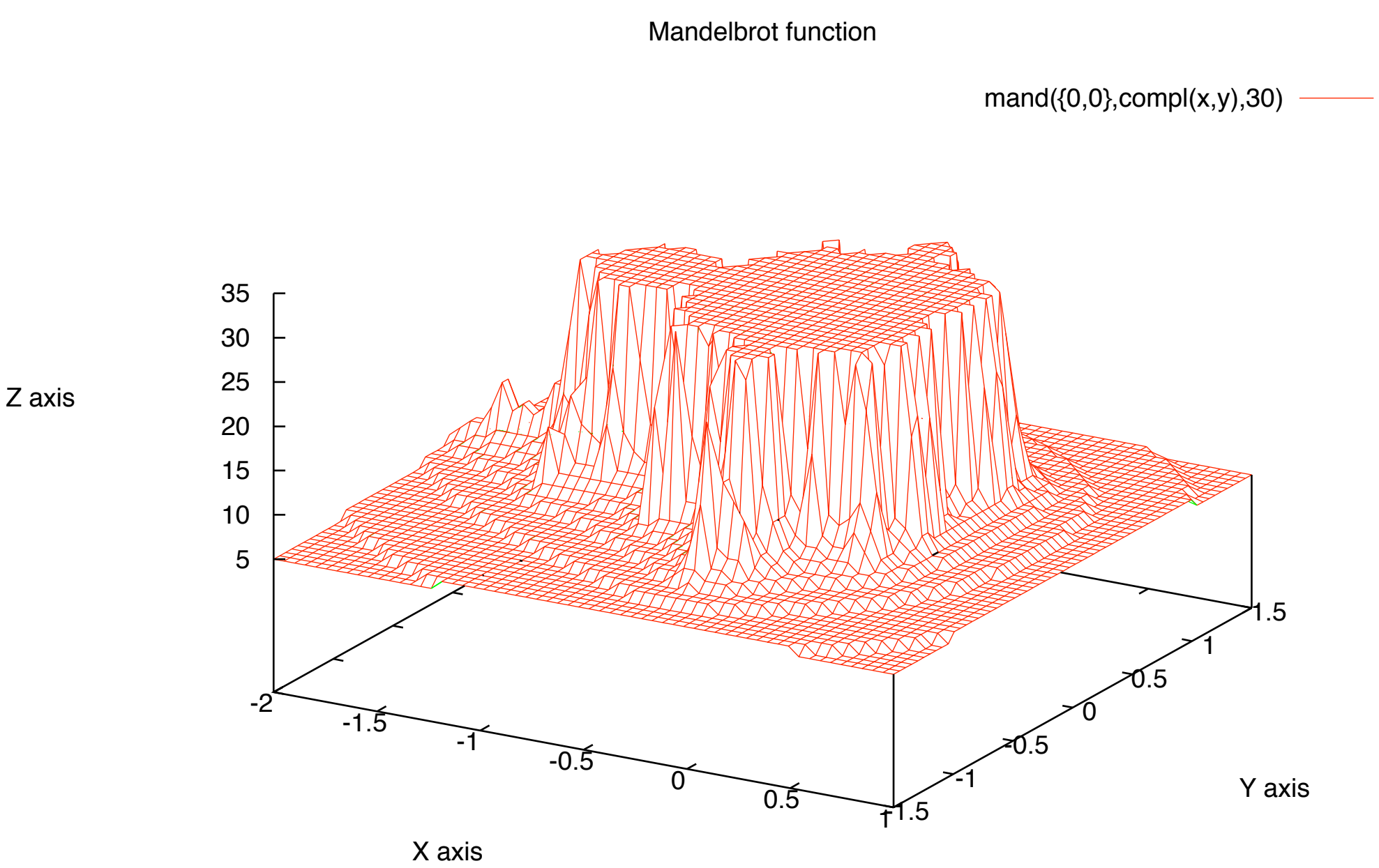

Data grid plotting
"glass.dat" using 3:2:1
Z axis
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0
0.5
1
1.5
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
X axis
Y axis

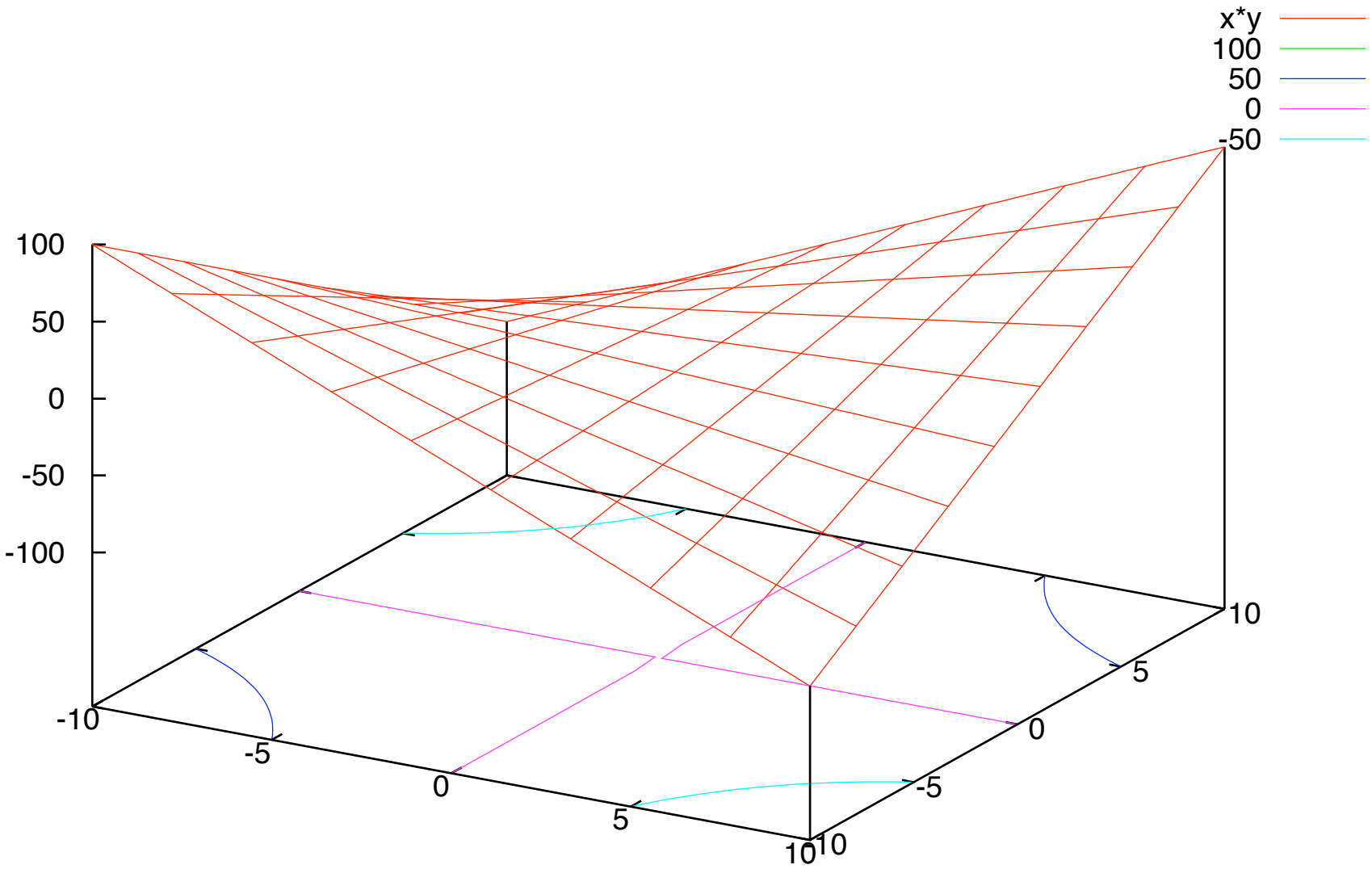

Data grid plotting
"glass.dat" using 1
"glass.dat" using 2
"glass.dat" using 3
Z axis
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
16
line index within group
group index


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

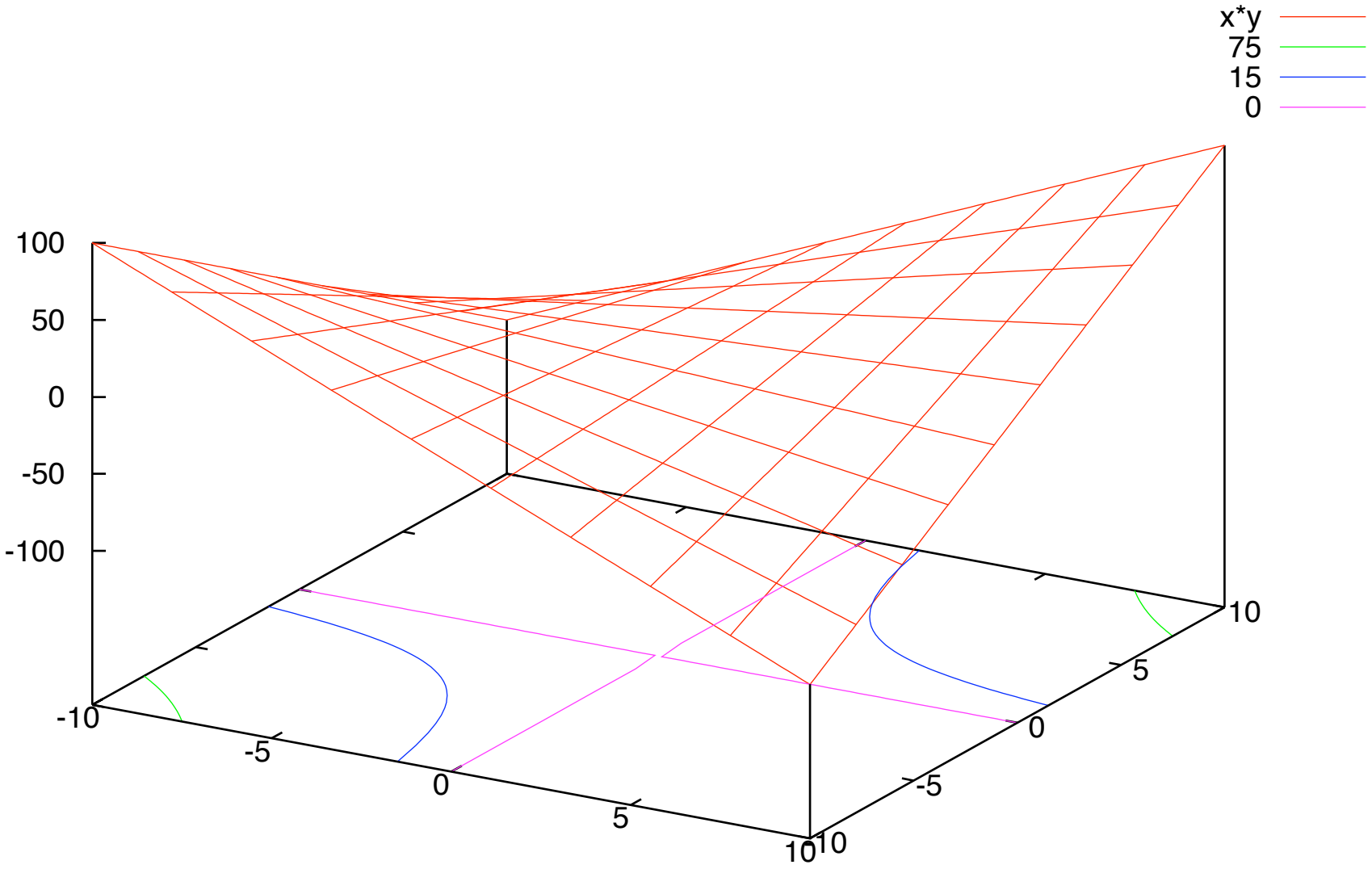

splot with "set pm3d" (implemented with some terminals)
'glass.dat' every 2::0::12
Z axis
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
X axis
Y axis
1.5
1
0.5
0
-0.5
-1

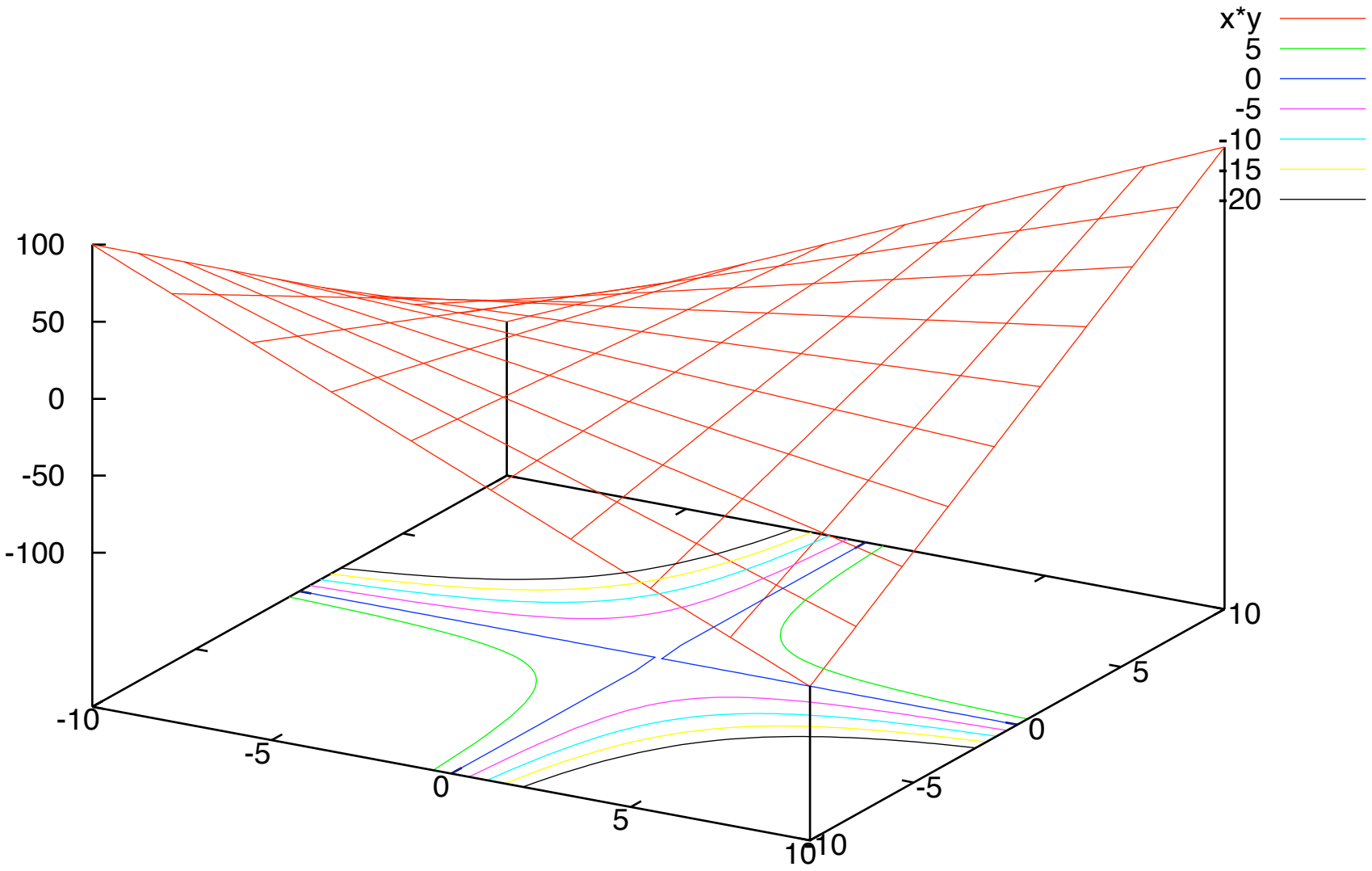

Test of spherical coordinates
"glass.dat"
Z axis
1
0.5
0
-0.5
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
X axis
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
Y axis


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

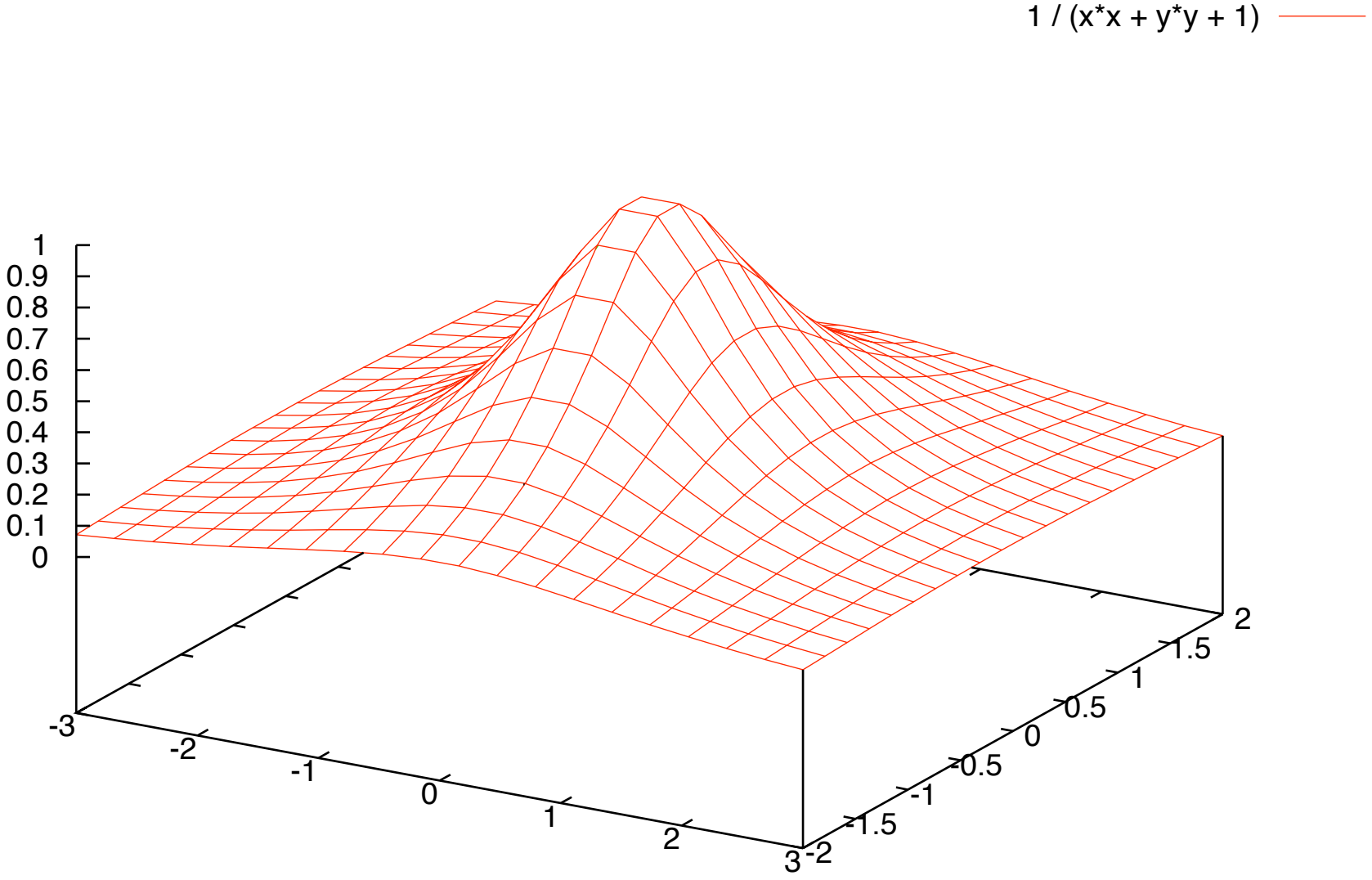

Demo of specifying discrete contour levels - default contours
x*y
100
50
0
-50
100
50
0
-50
-100
-10
-5
0
5
10
10
5
0
-5
-10

3 discrete contours at 0 15 75

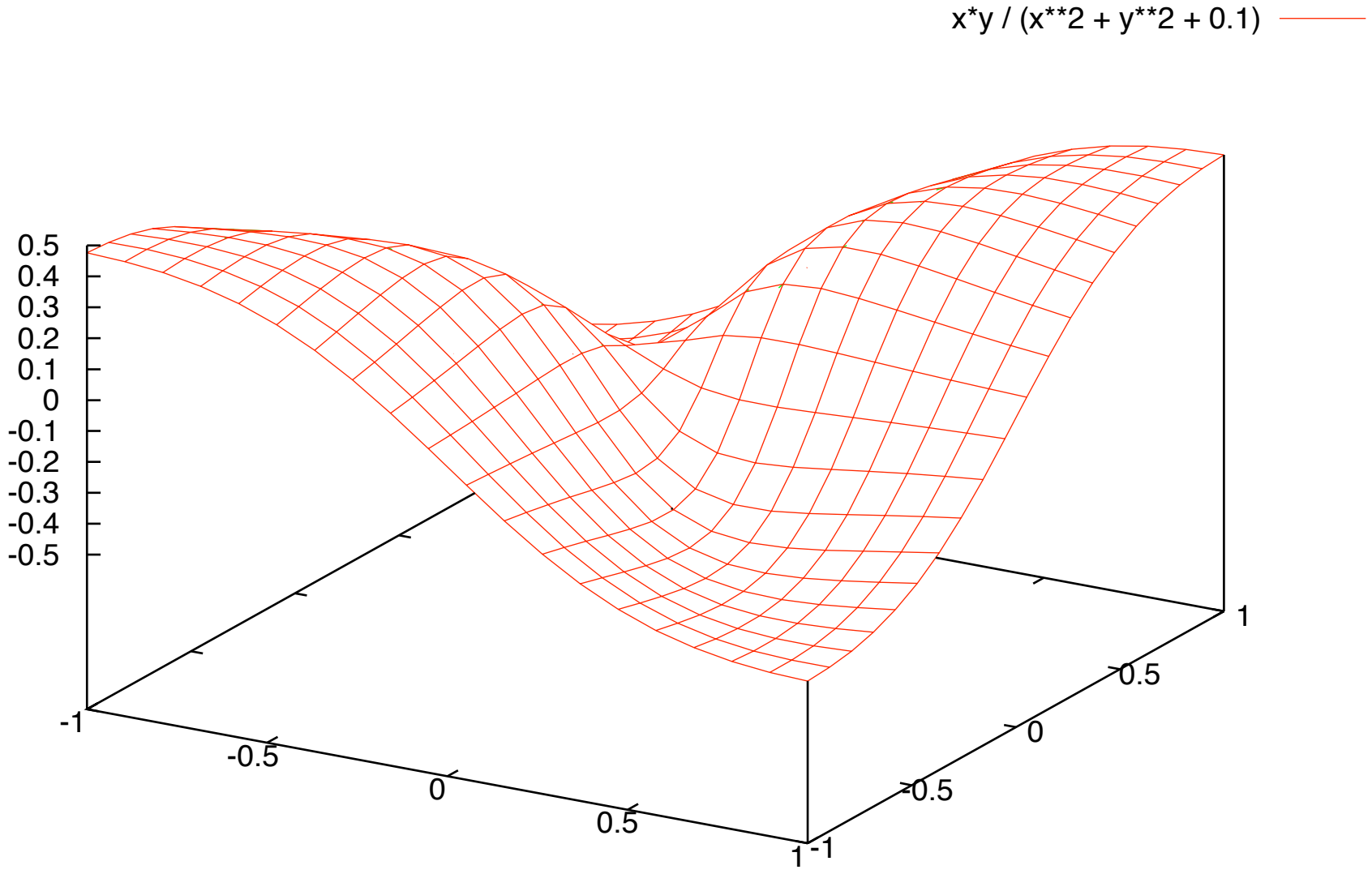



9 incremental contours starting at -20, stepping by 5
x*y
5
0
-5
-10
-15
-20
100
50
0
-50
-100
-10
-5
0
5
10
10
5
0
-5
-10

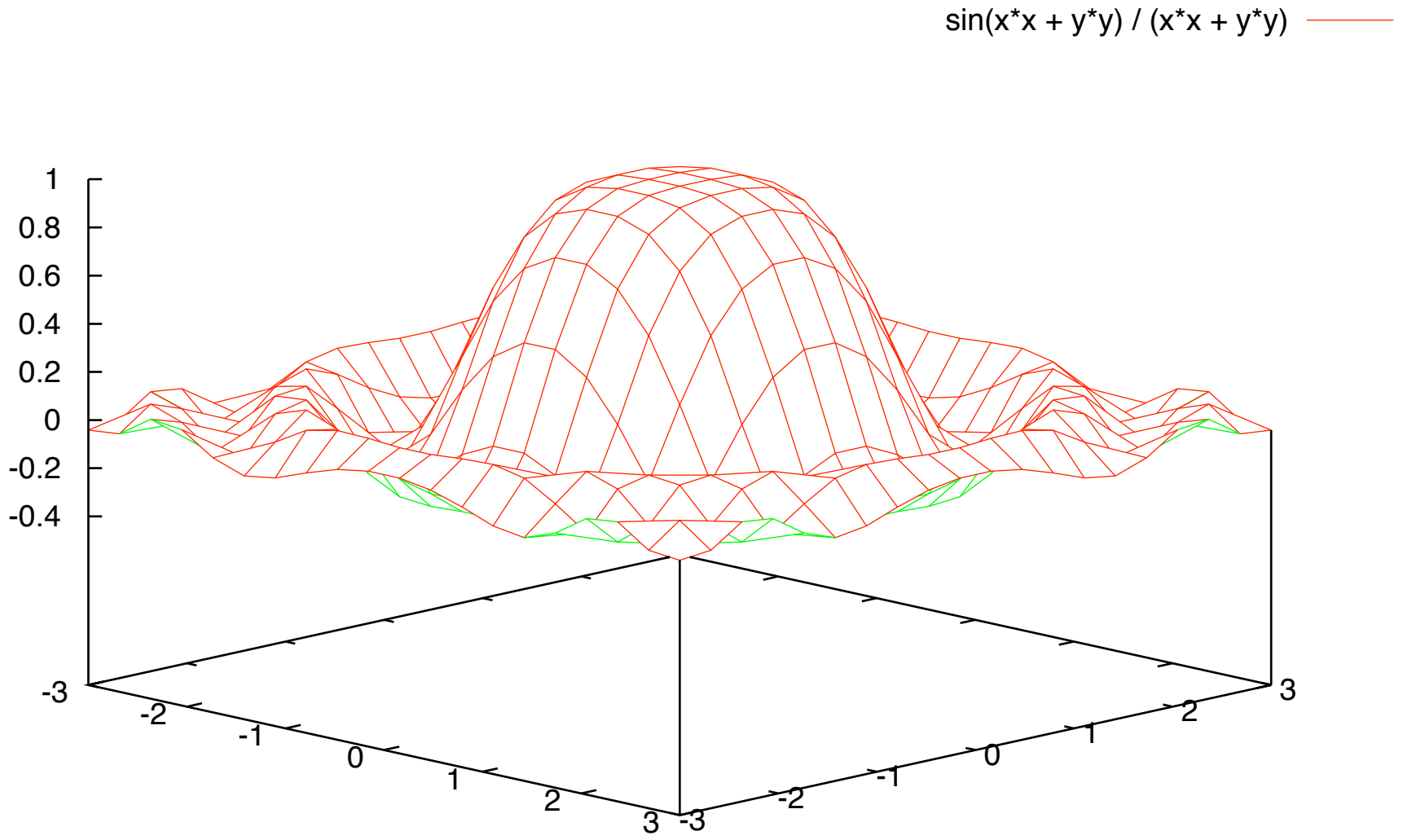
Hidden line removal of explicit surfaces



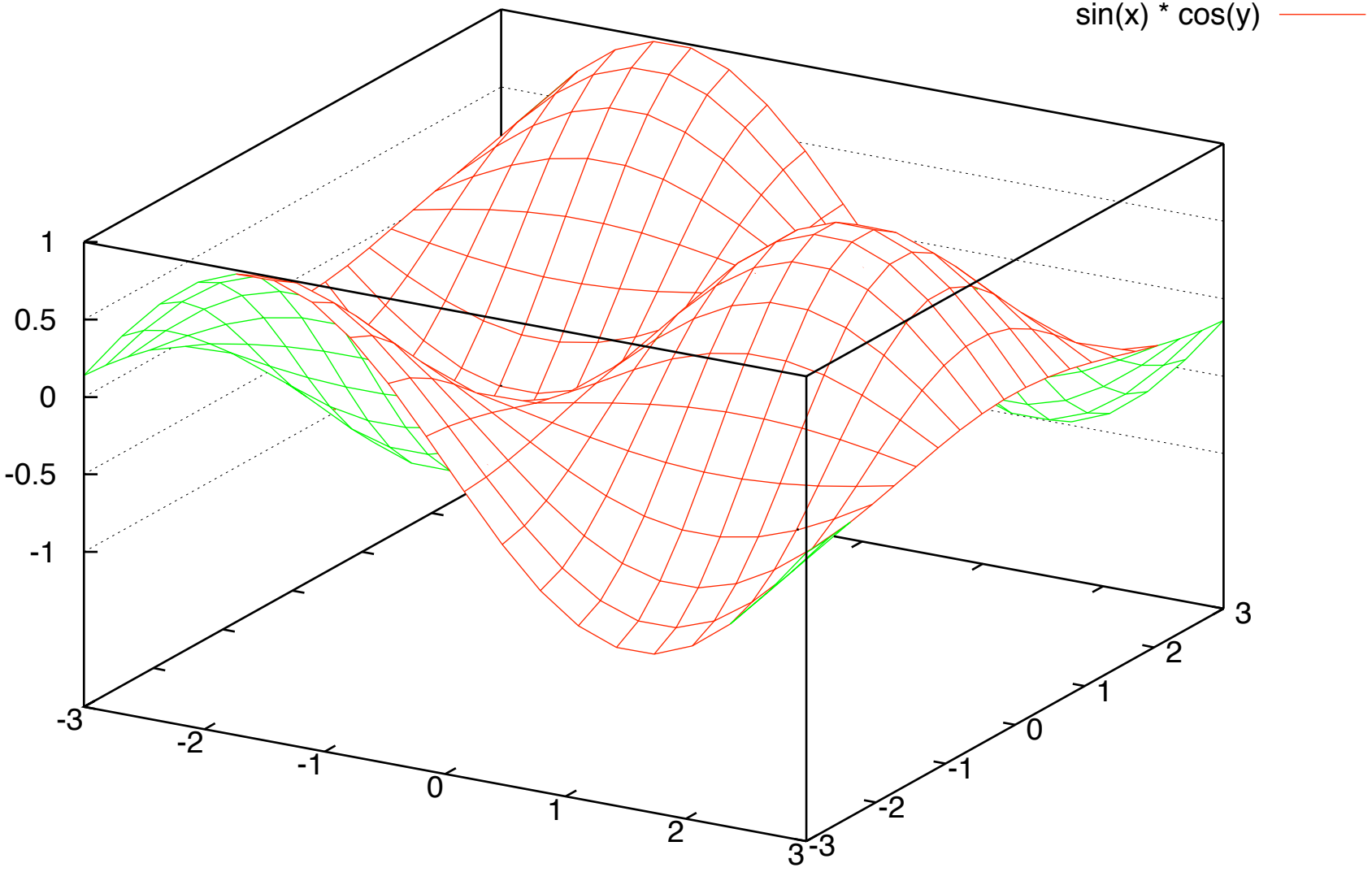
Hidden line removal of explicit surfaces



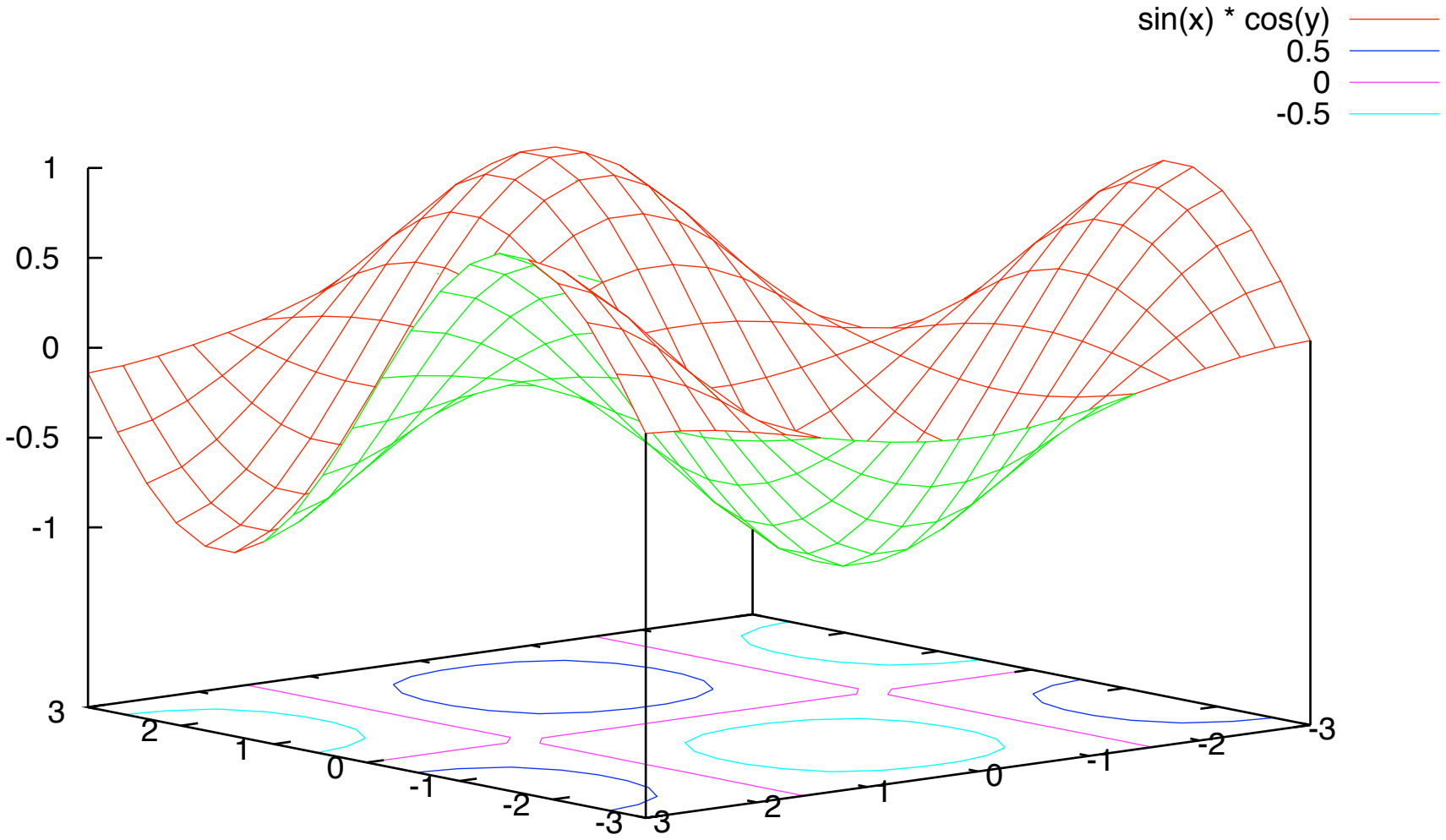
Hidden line removal of explicit surfaces

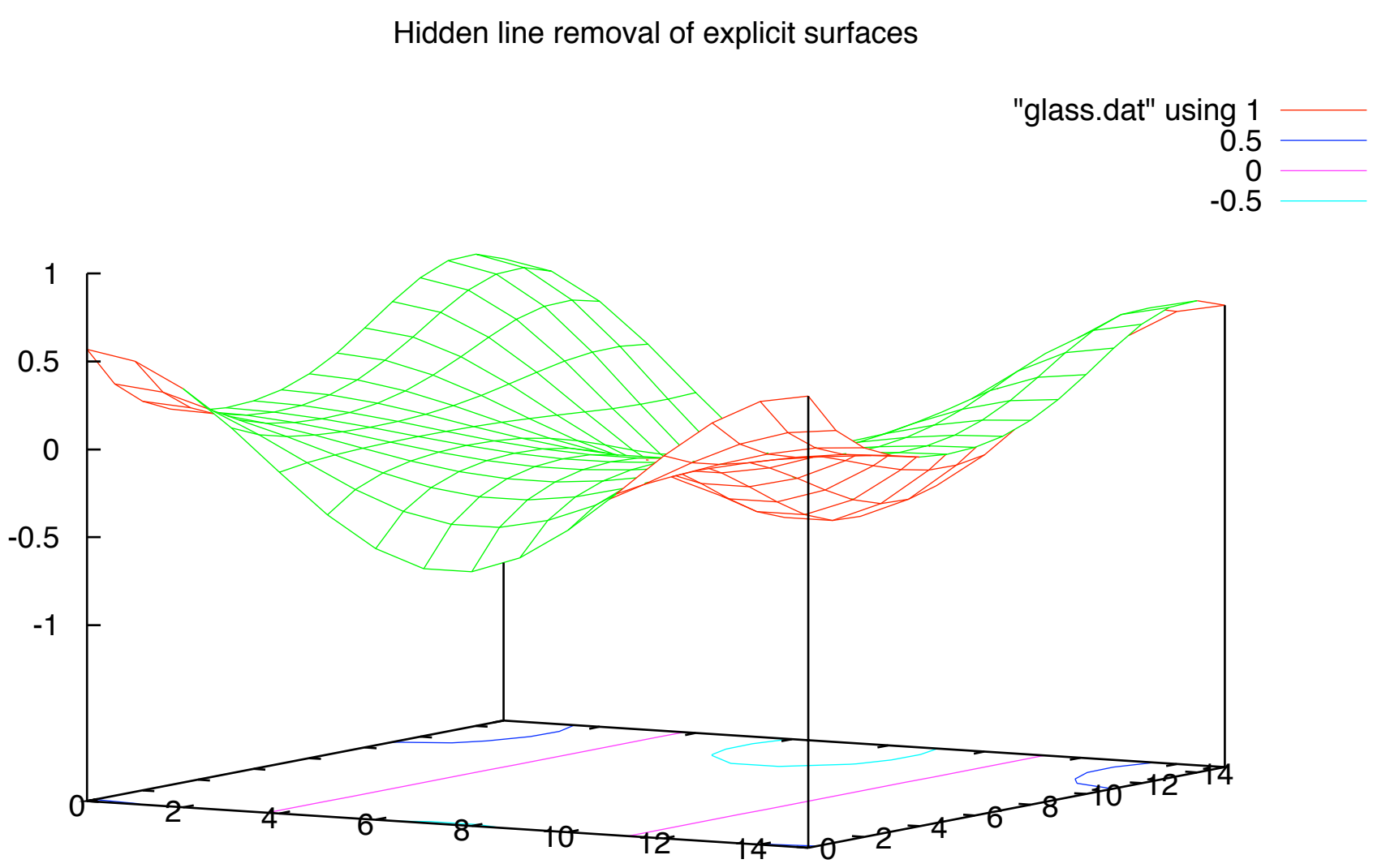


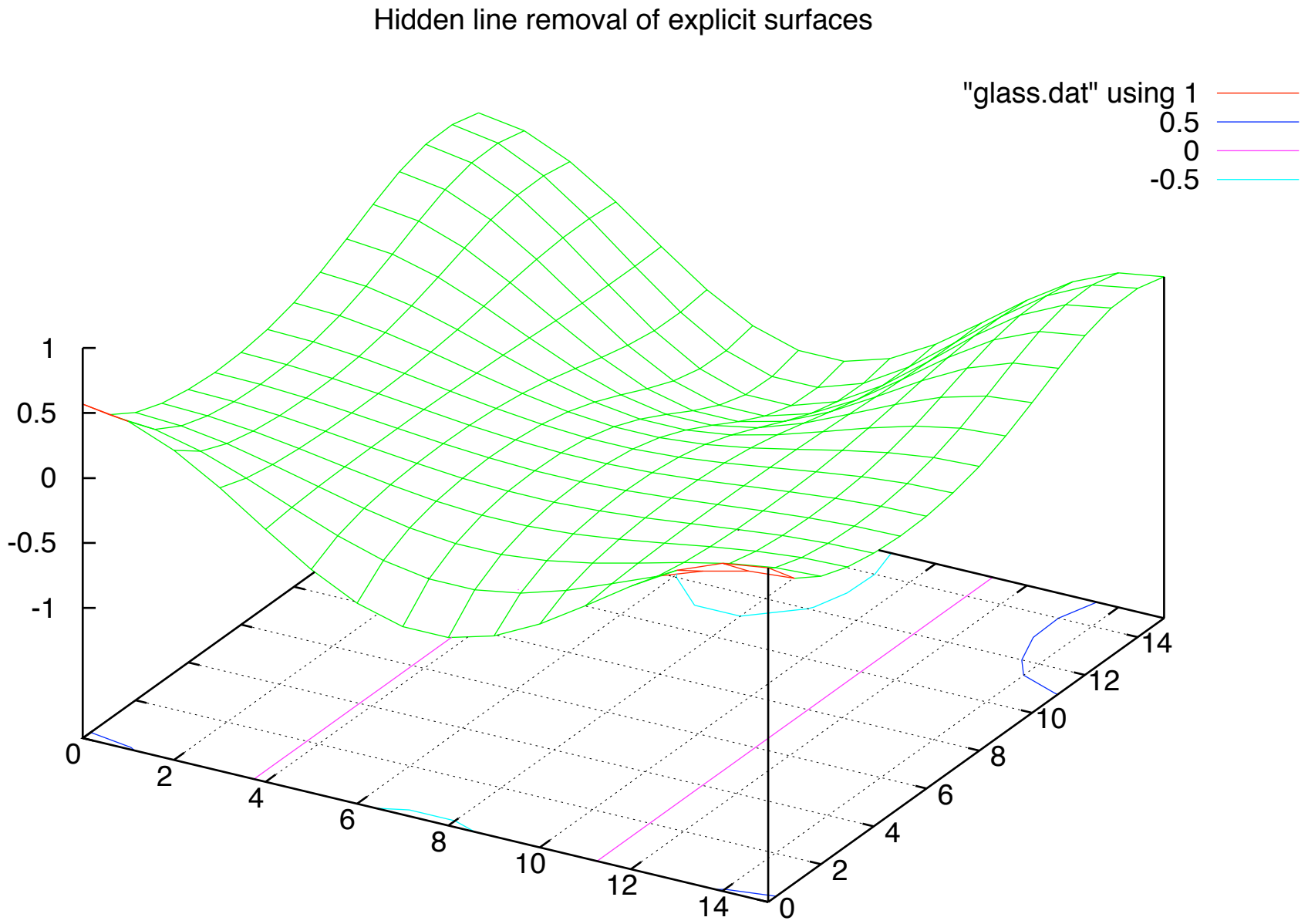
Hidden line removal of explicit surfaces



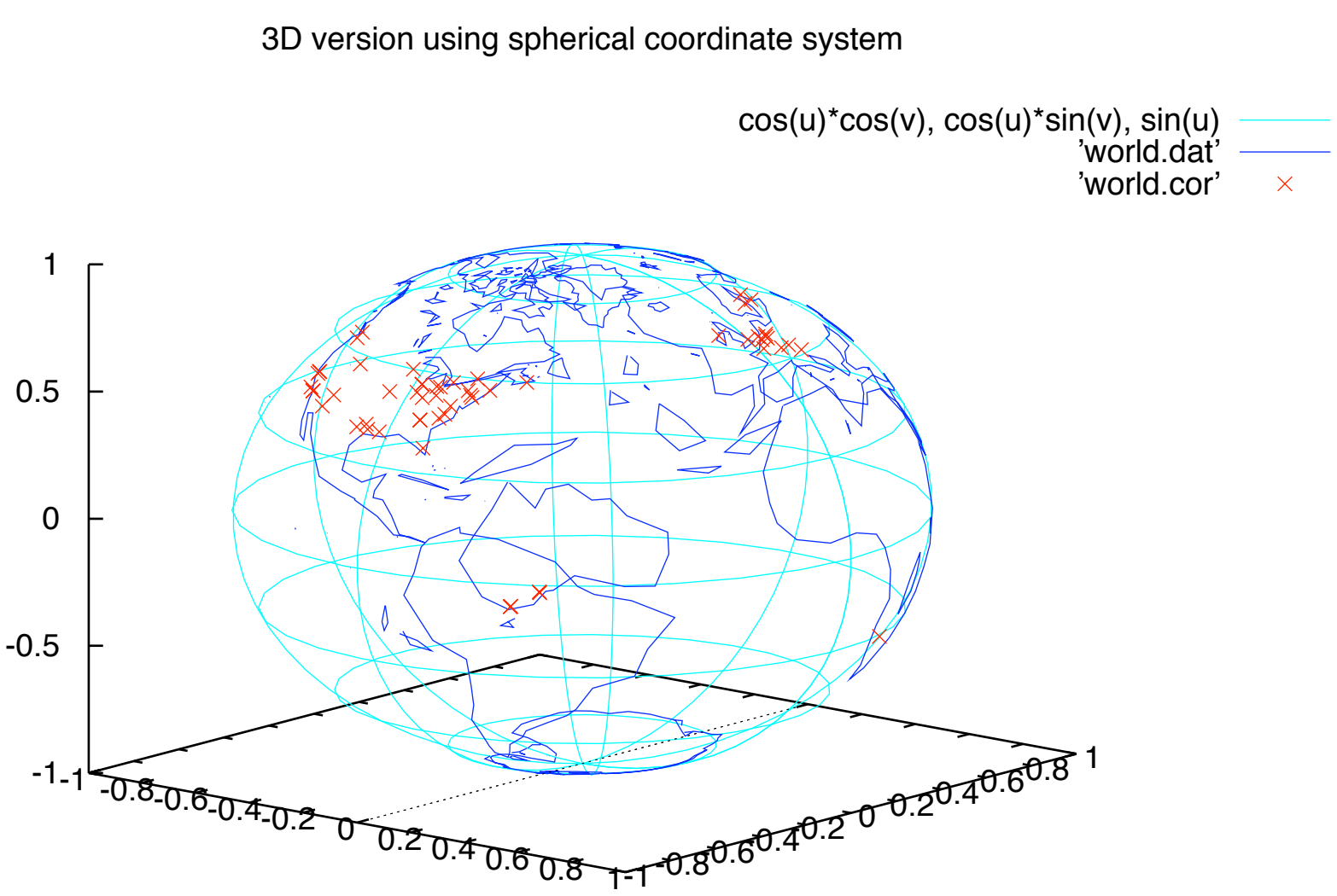
Hidden line removal of explicit surfaces

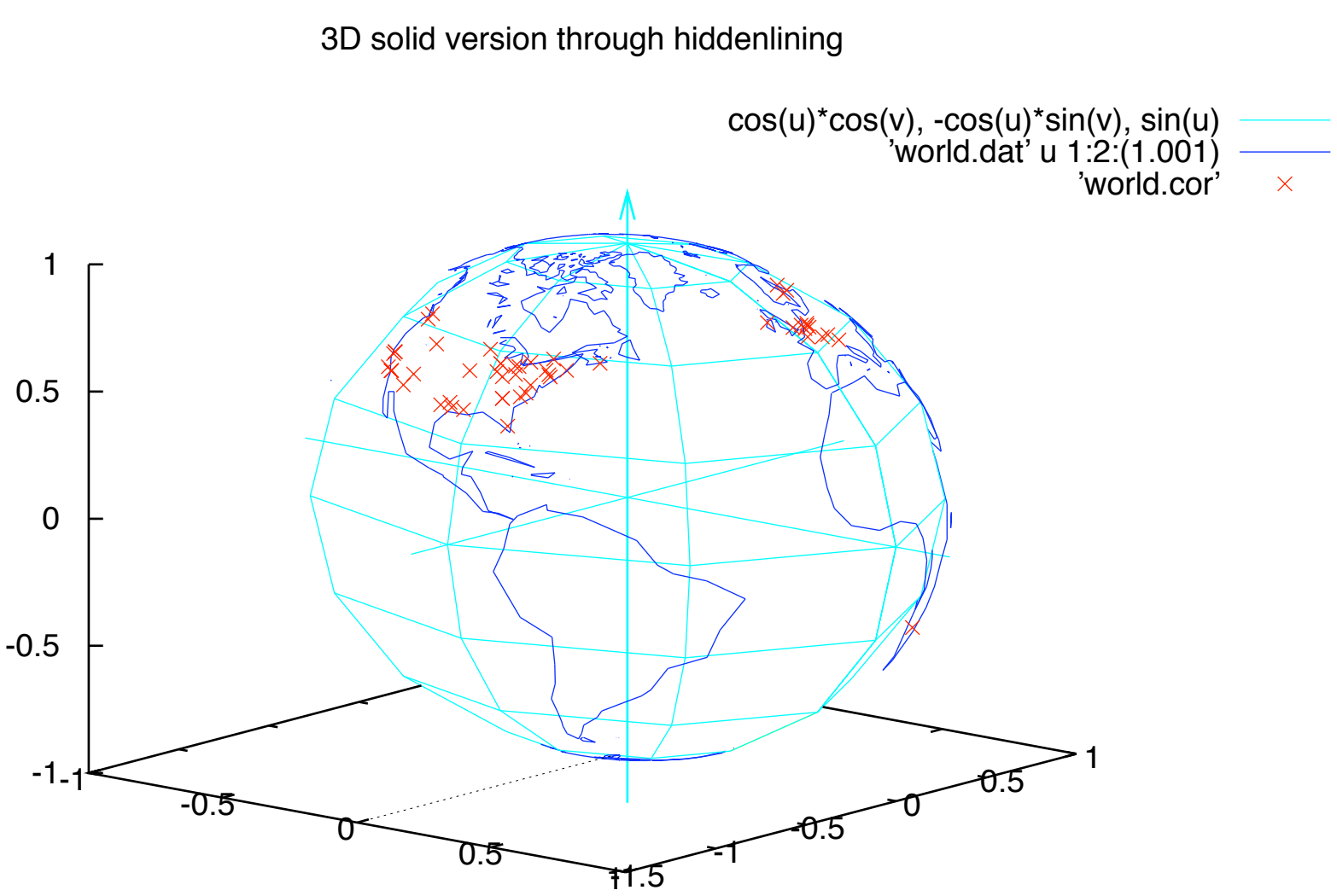


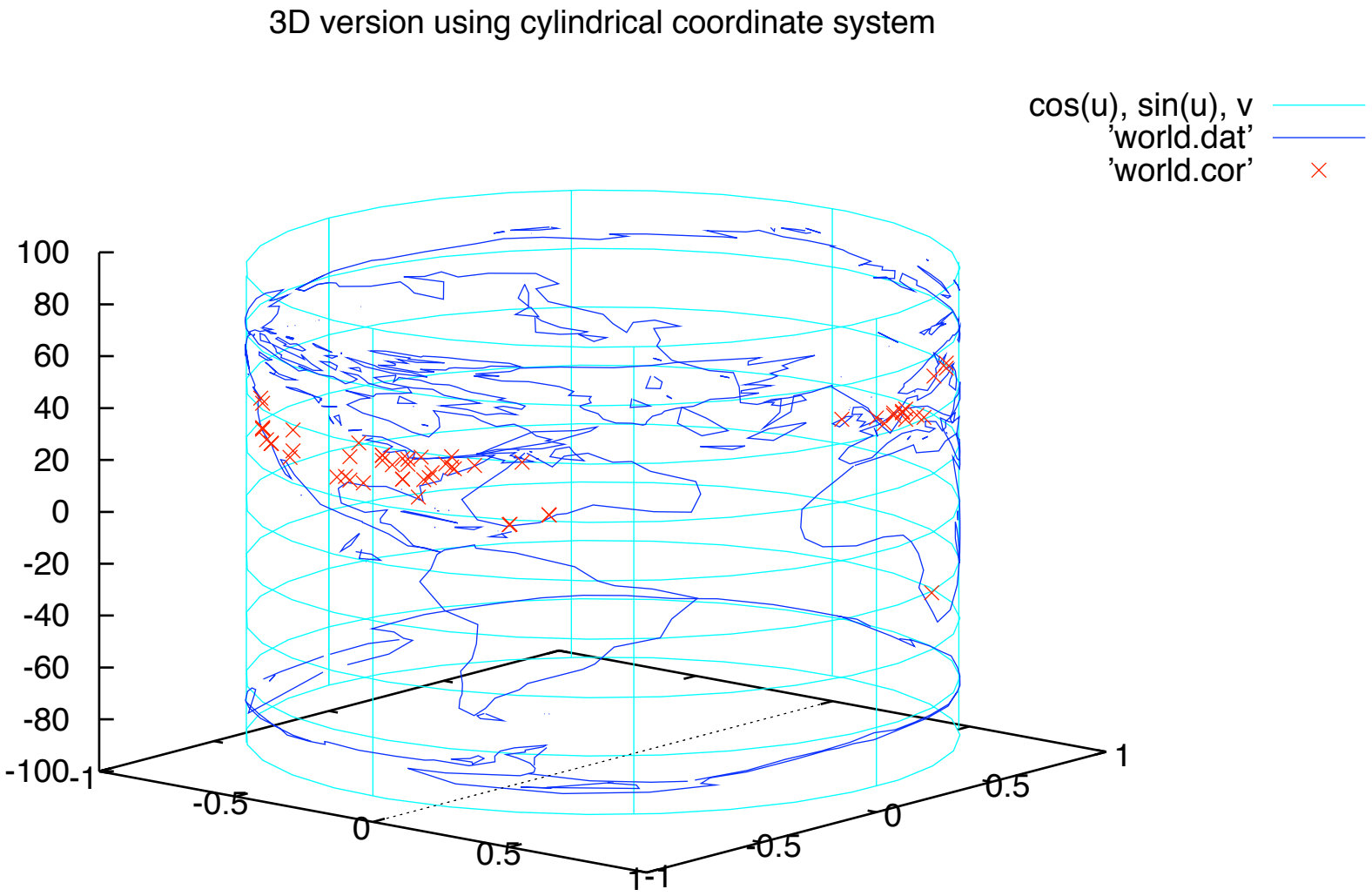

Hidden line removal of explicit surfaces
"glass.dat" using 1
0.5
0
-0.5
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

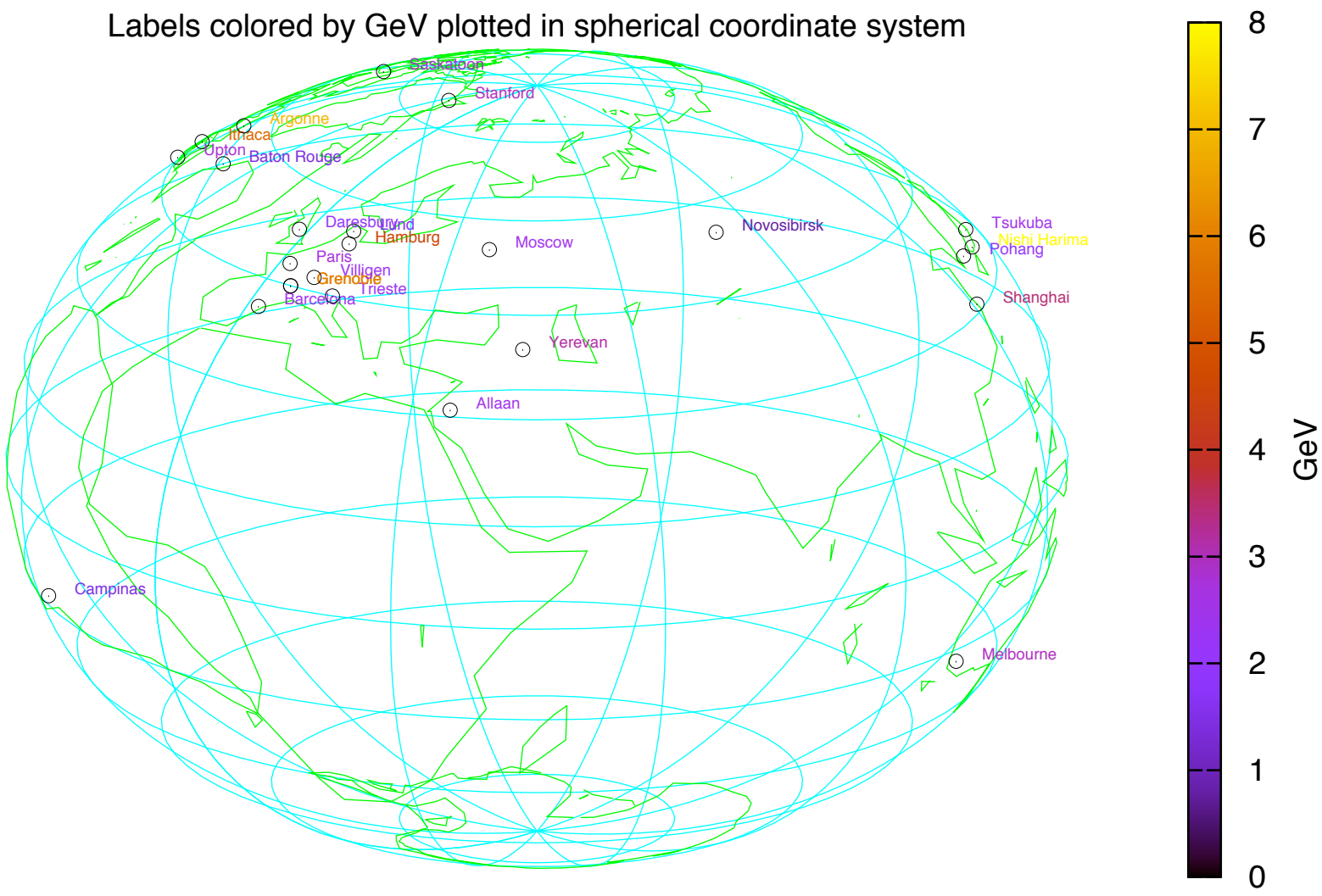

Hidden line removal of explicit surfaces
"glass.dat" using 1
0.5
0
-0.5
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

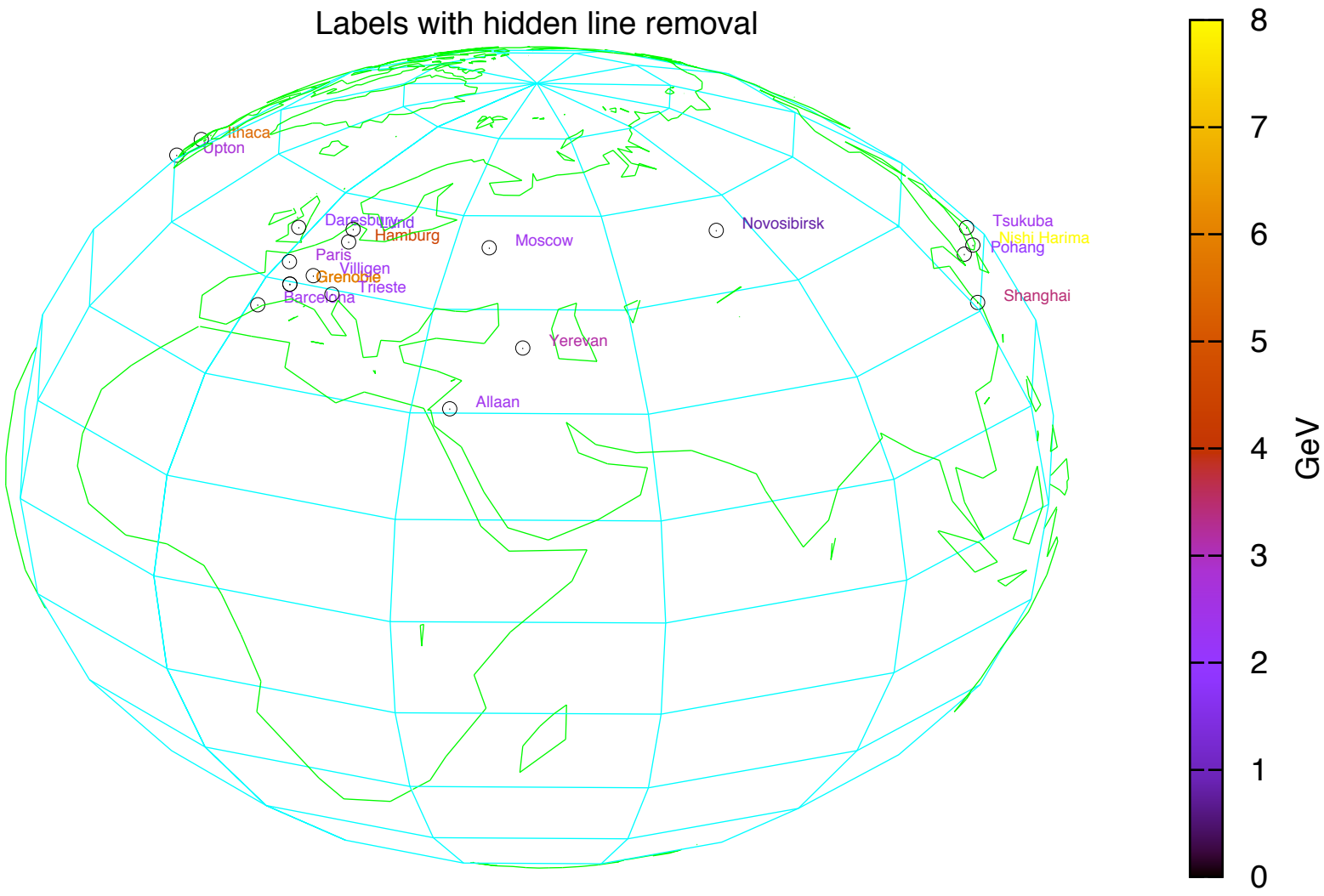
Gnuplot Correspondences

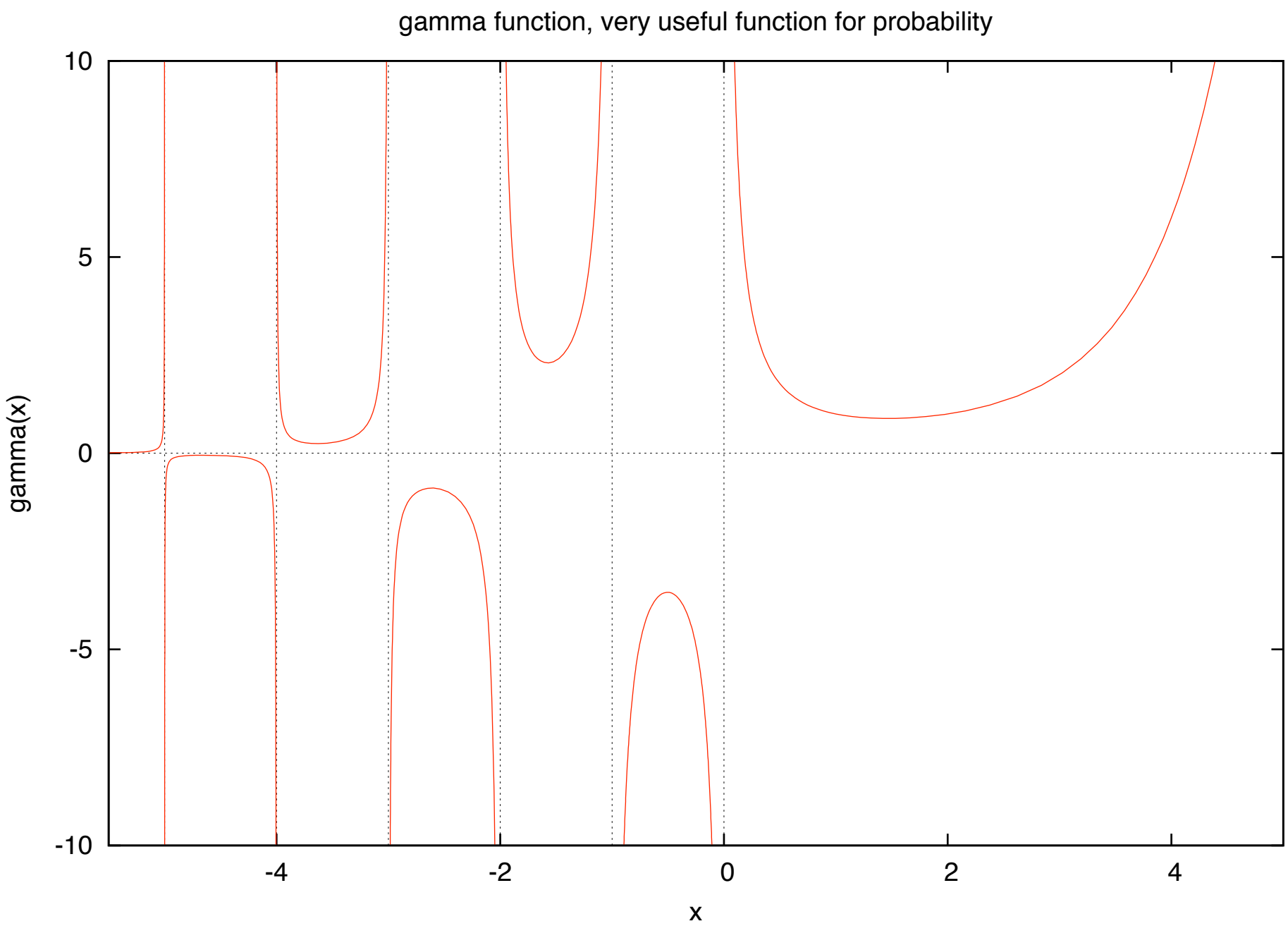

3D version using spherical coordinate system
cos(u)*cos(v), cos(u)*sin(v), sin(u)
'world.dat'
'world.cor'
1
0.5
0
-0.5
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

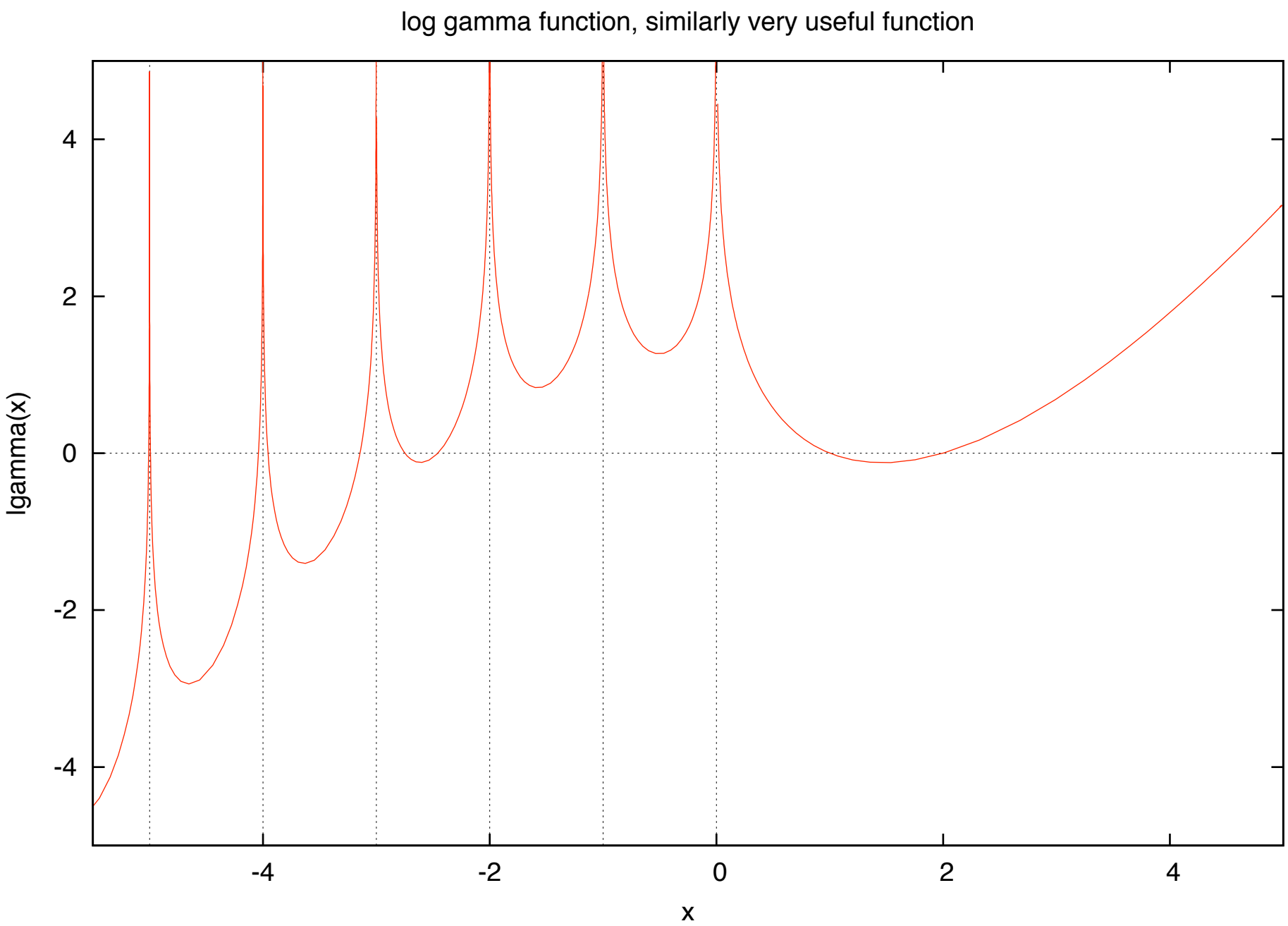

3D solid version through hiddenlining
cos(u)*cos(v), -cos(u)*sin(v), sin(u)
'world.dat' u 1:2:(1.001)
'world.cor'
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
1.5
-1
-0.5
0
0.5
1

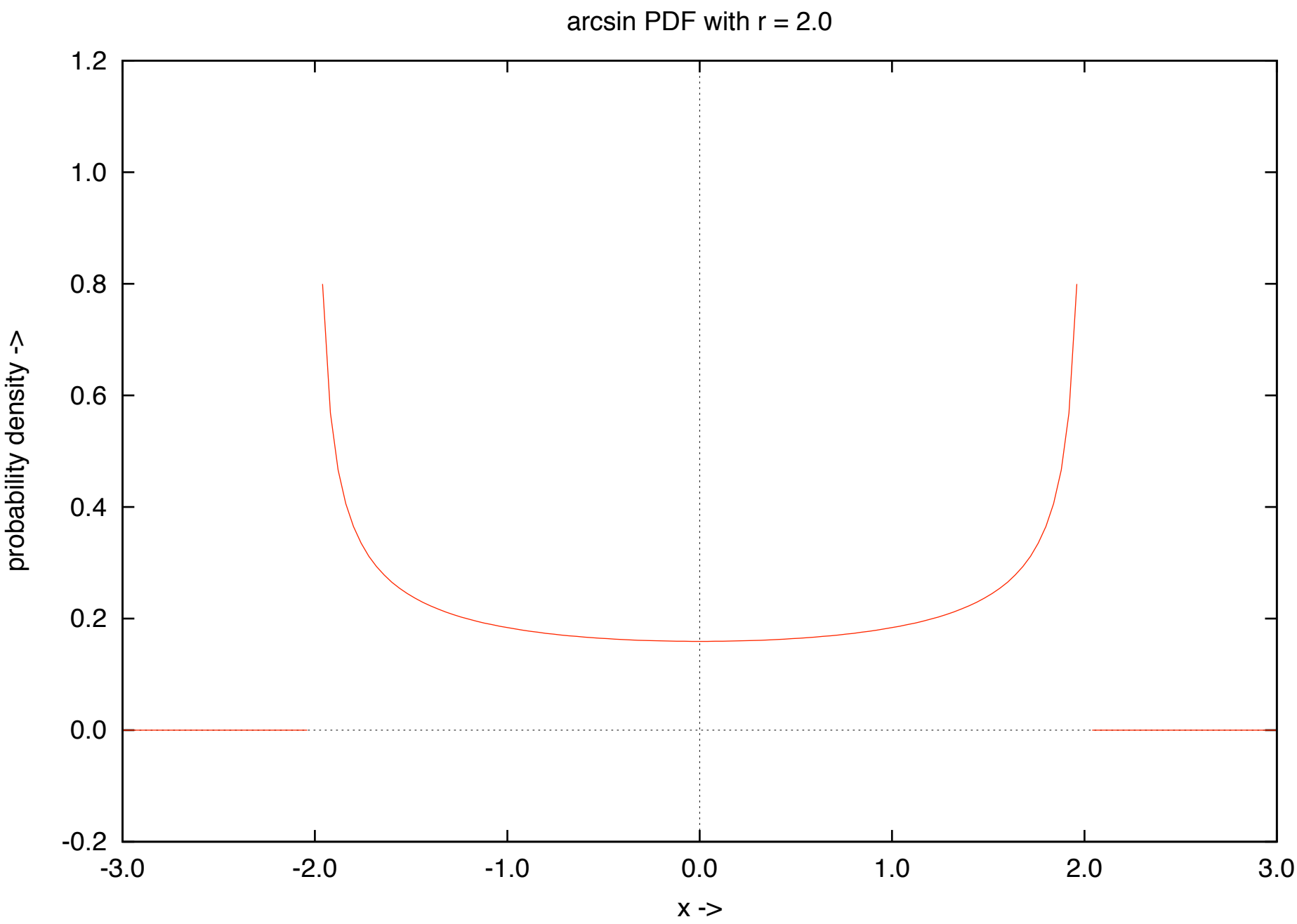

3D version using cylindrical coordinate system
cos(u), sin(u), v
'world.dat'
'world.cor'
100
80
60
40
20
0
-20
-40
-60
-80
-100
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1

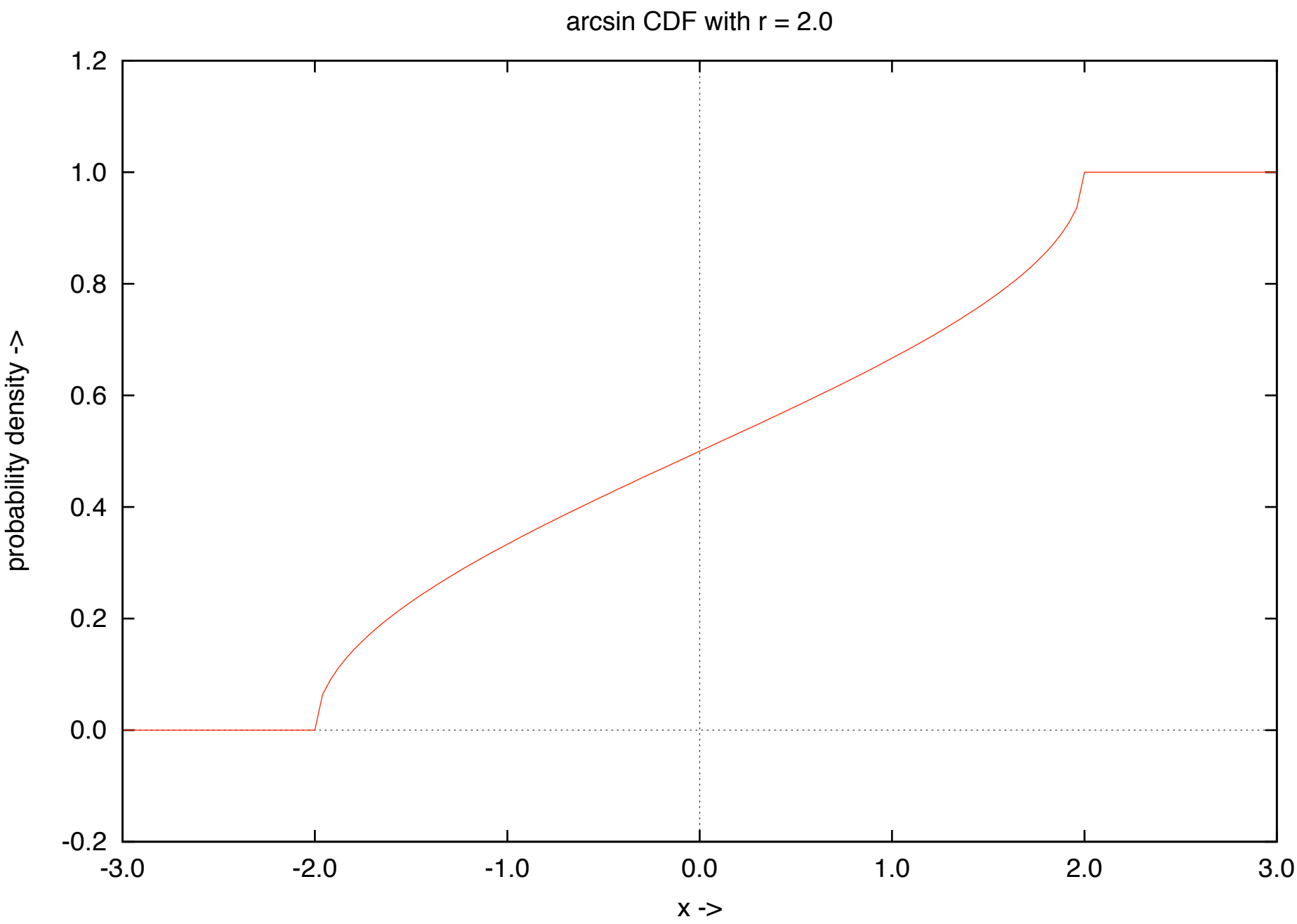

Labels colored by GeV plotted in spherical coordinate system
Stanford
Argonne
Ithaca
Upton
Baton Rouge
Daresbury
Hamburg
Moscow
Novosibirsk
Tsukuba
Nishi Harima
Pohang
Paris
Villigen
Grenoble
Trieste
Barcelona
Shanghai
Yerevan
Allaan
Campinas
Melbourne
GeV
8
7
6
5
4
3
2
1
0

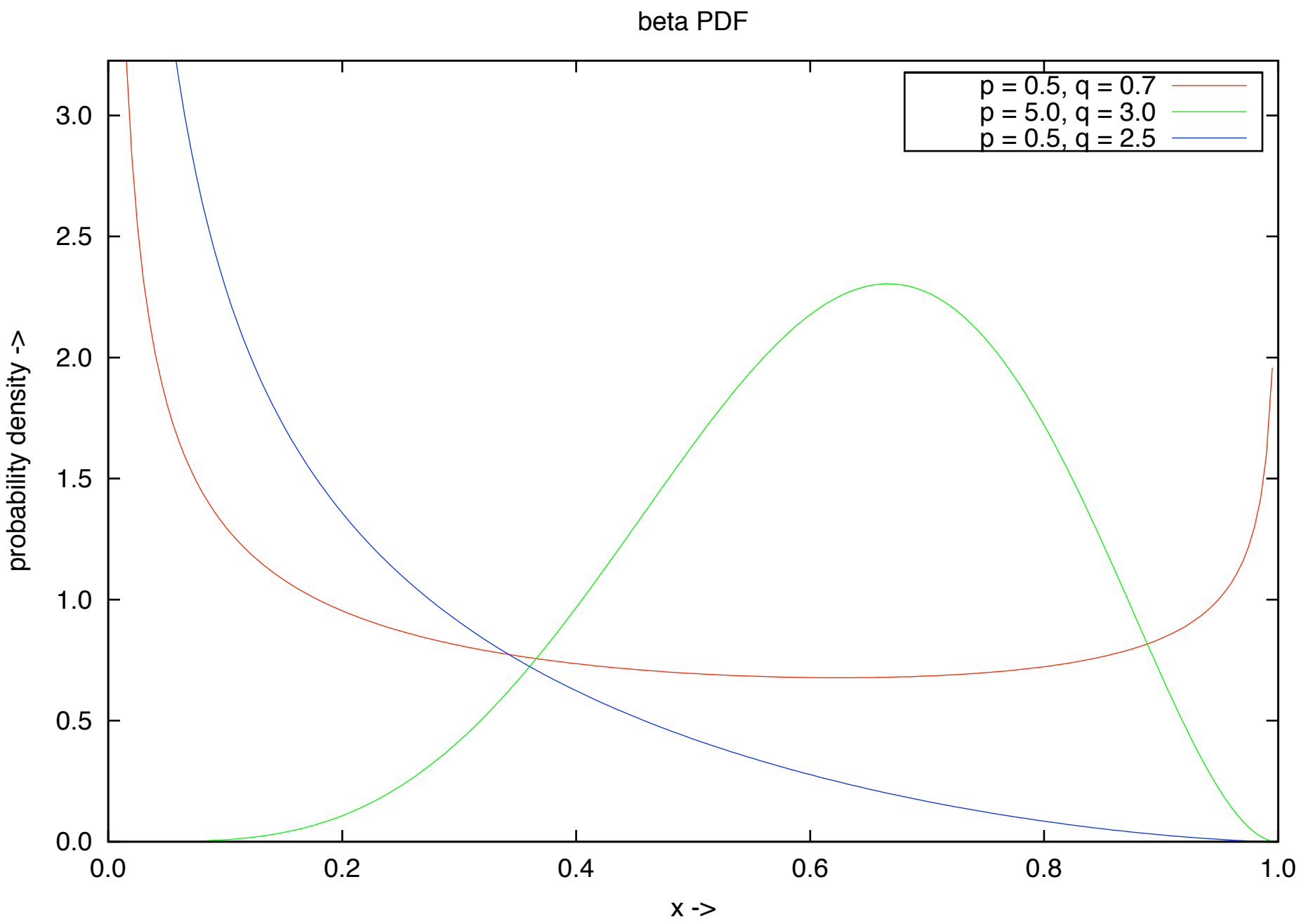

Labels with hidden line removal
Ithaca
Upton
Daresbury
Lund
Hamburg
Moscow
Novosibirsk
Tsukuba
Nishi Harima
Pohang
Paris
Villigen
Grenoble
Trieste
Barcelona
Shanghai
Yerevan
Allaan
8
7
6
5
4
3
2
1
0
GeV

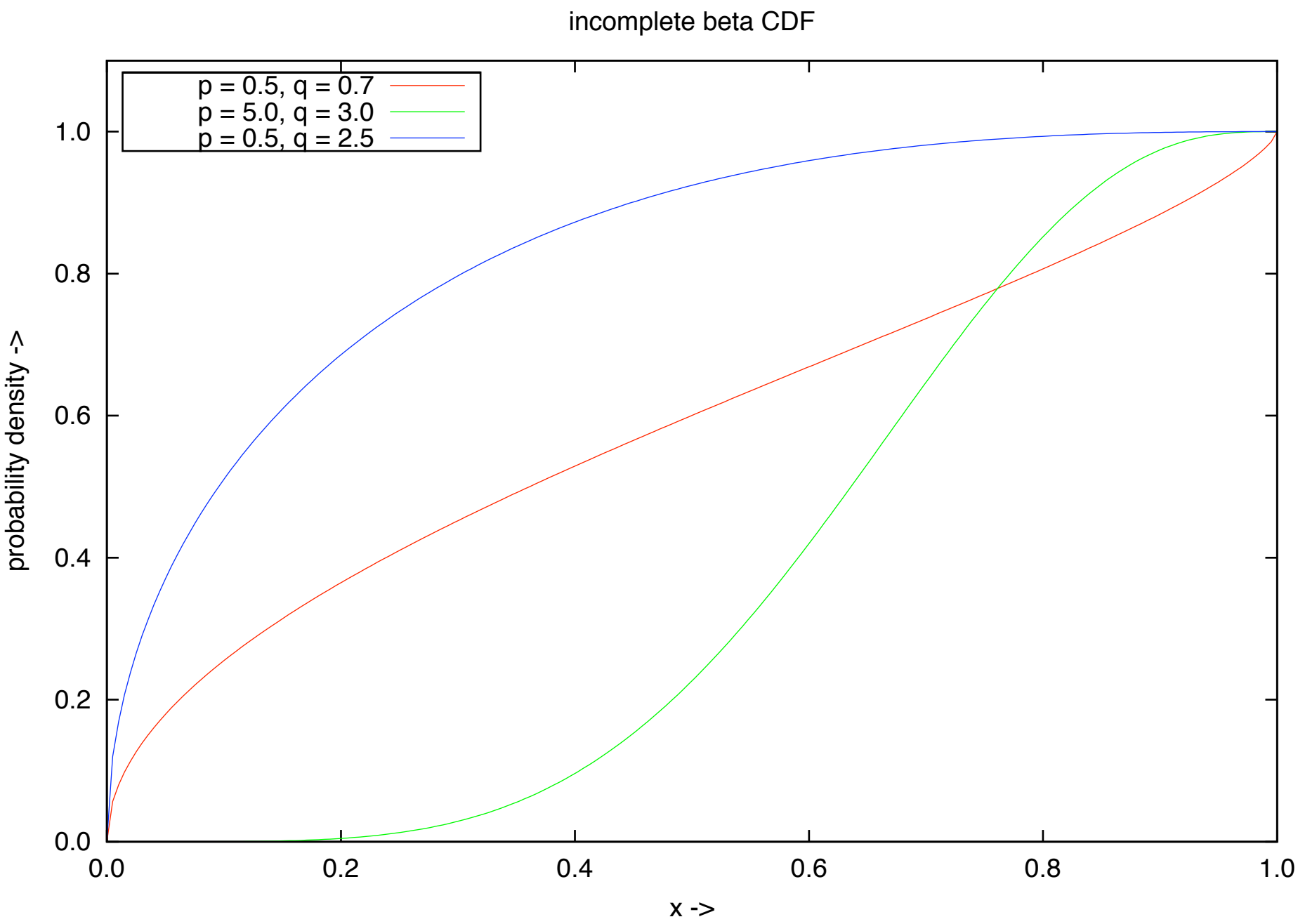

gamma function, very useful function for probability
gamma(x)
x
10
5
0
-5
-10
-4
-2
0
2
4

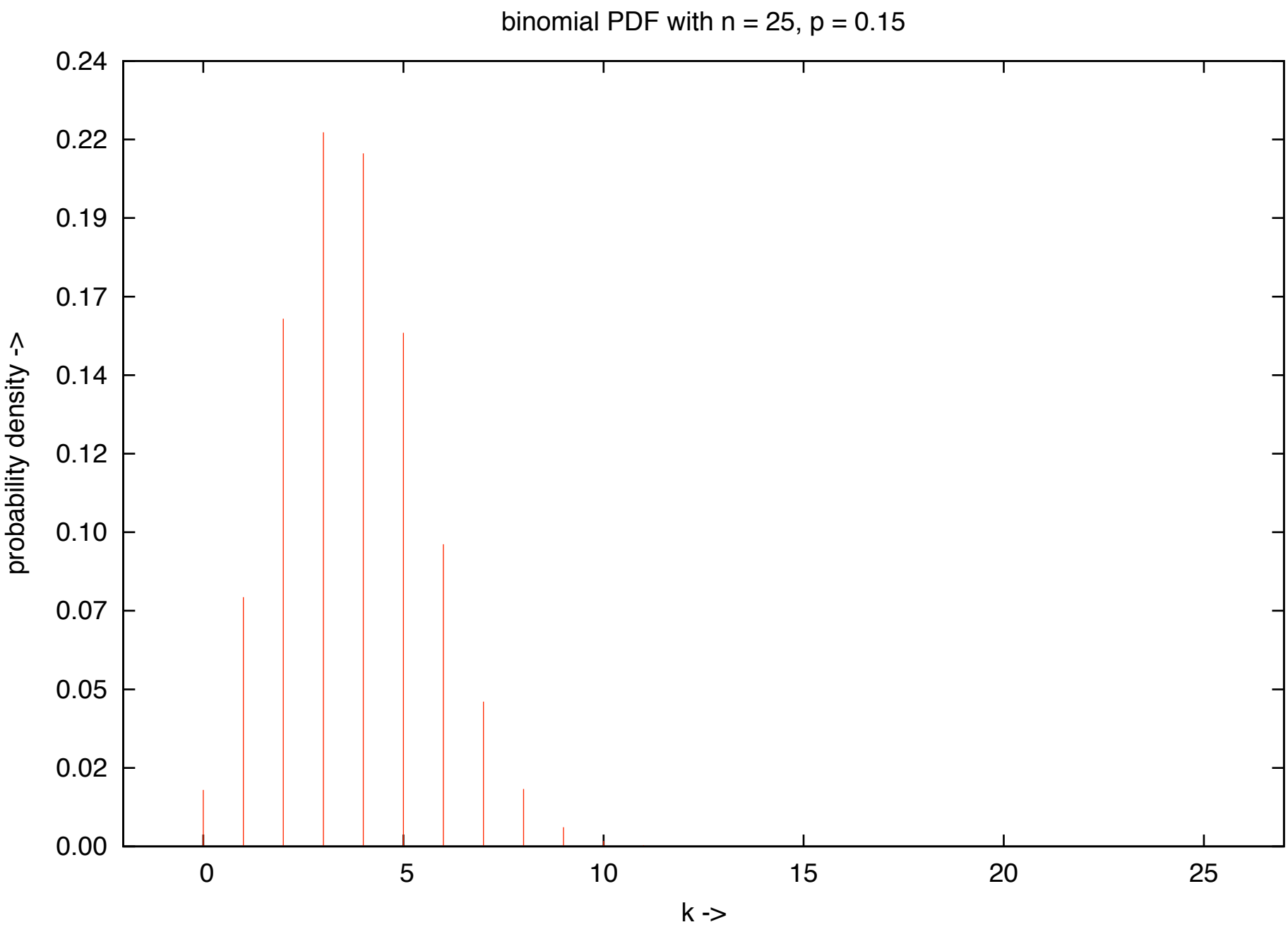

log gamma function, similarly very useful function
lgamma(x)
x
4
2
0
-2
-4
-4
-2
0
2
4

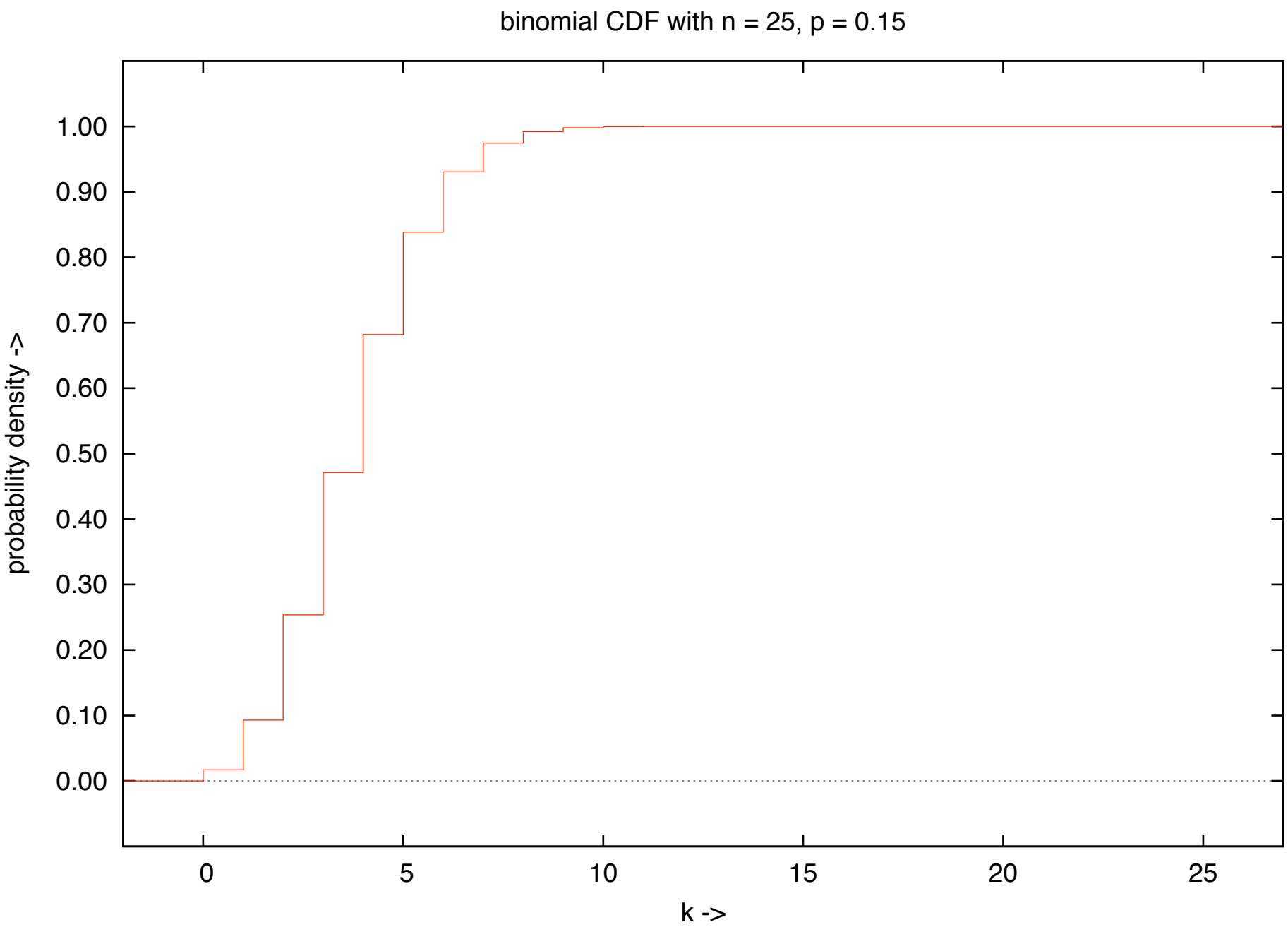

arcsin PDF with r = 2.0
probability density ->
x ->
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

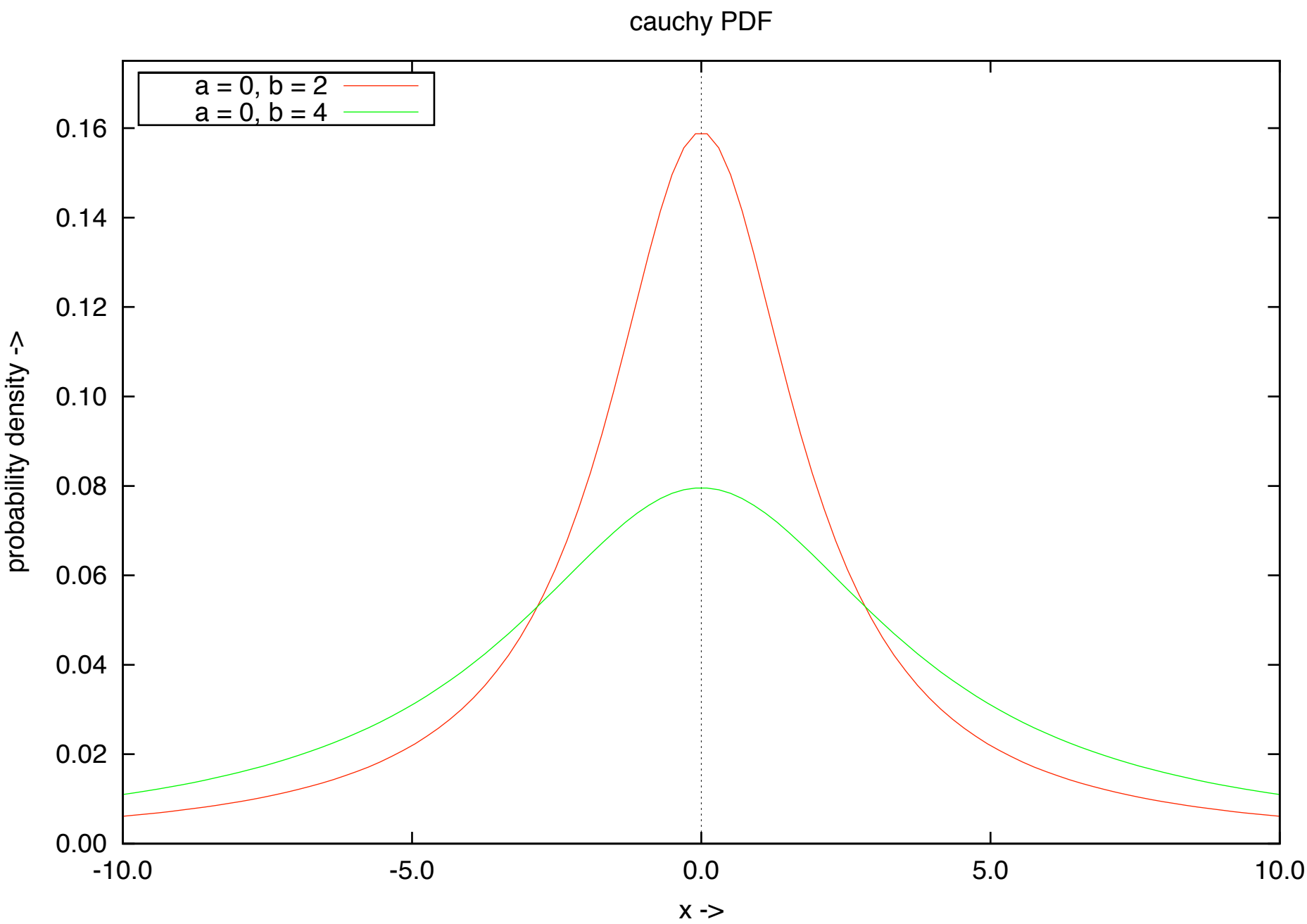

arcsin CDF with r = 2.0
probability density ->
x ->
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

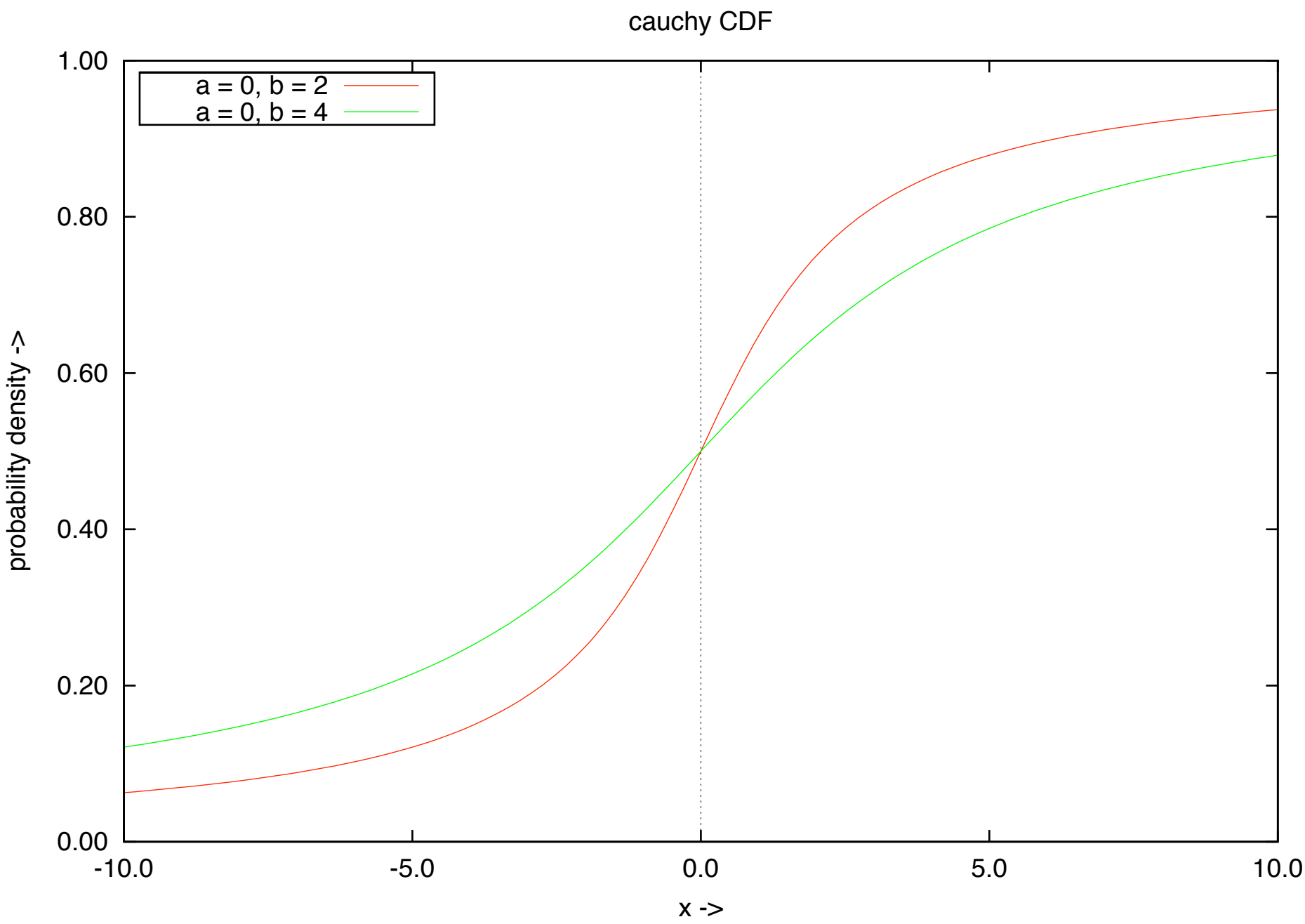

beta PDF
p = 0.5, q = 0.7
p = 5.0, q = 3.0
p = 0.5, q = 2.5
probability density ->
x ->
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
1.0

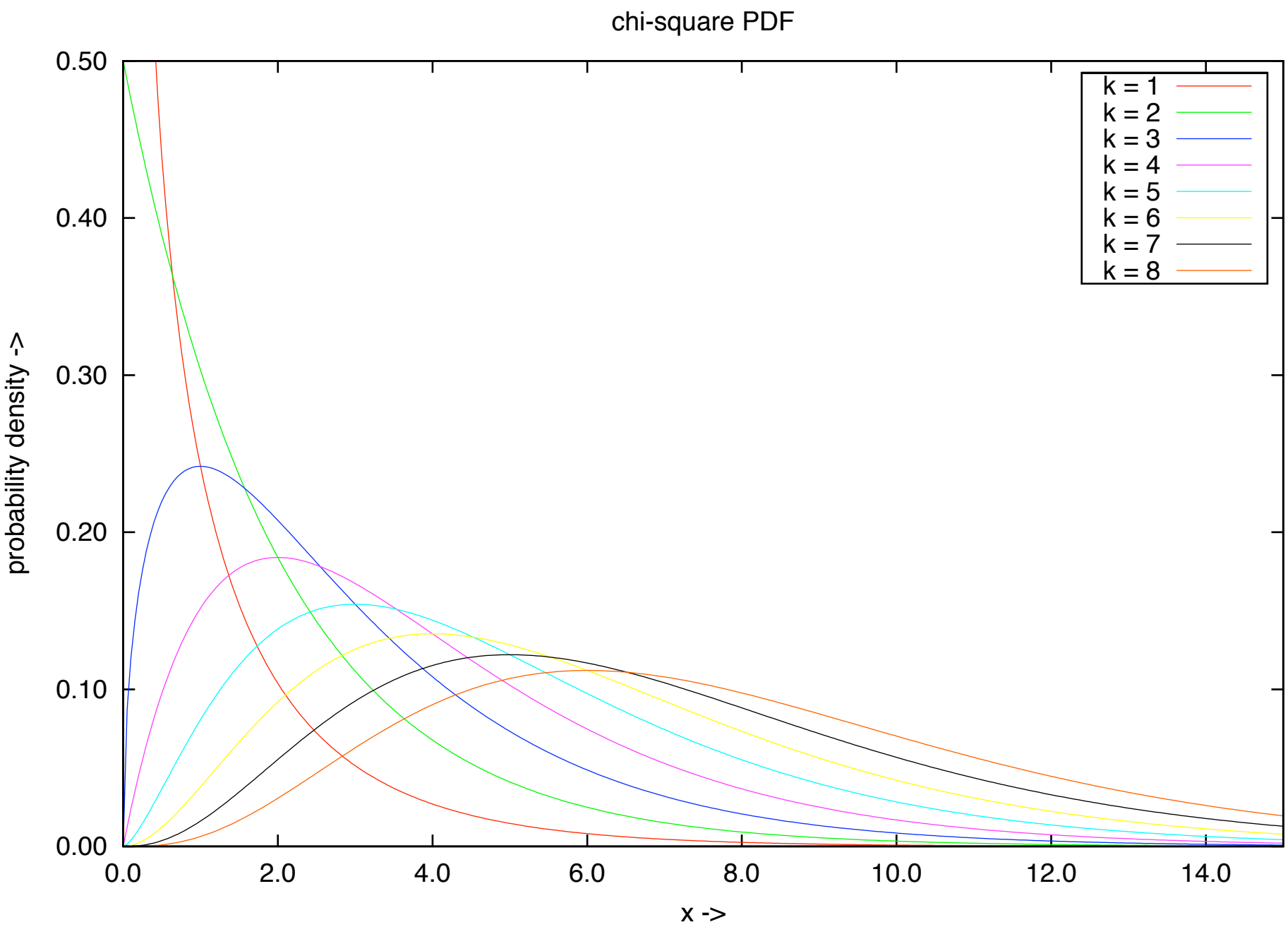

incomplete beta CDF
p = 0.5, q = 0.7
p = 5.0, q = 3.0
p = 0.5, q = 2.5
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0

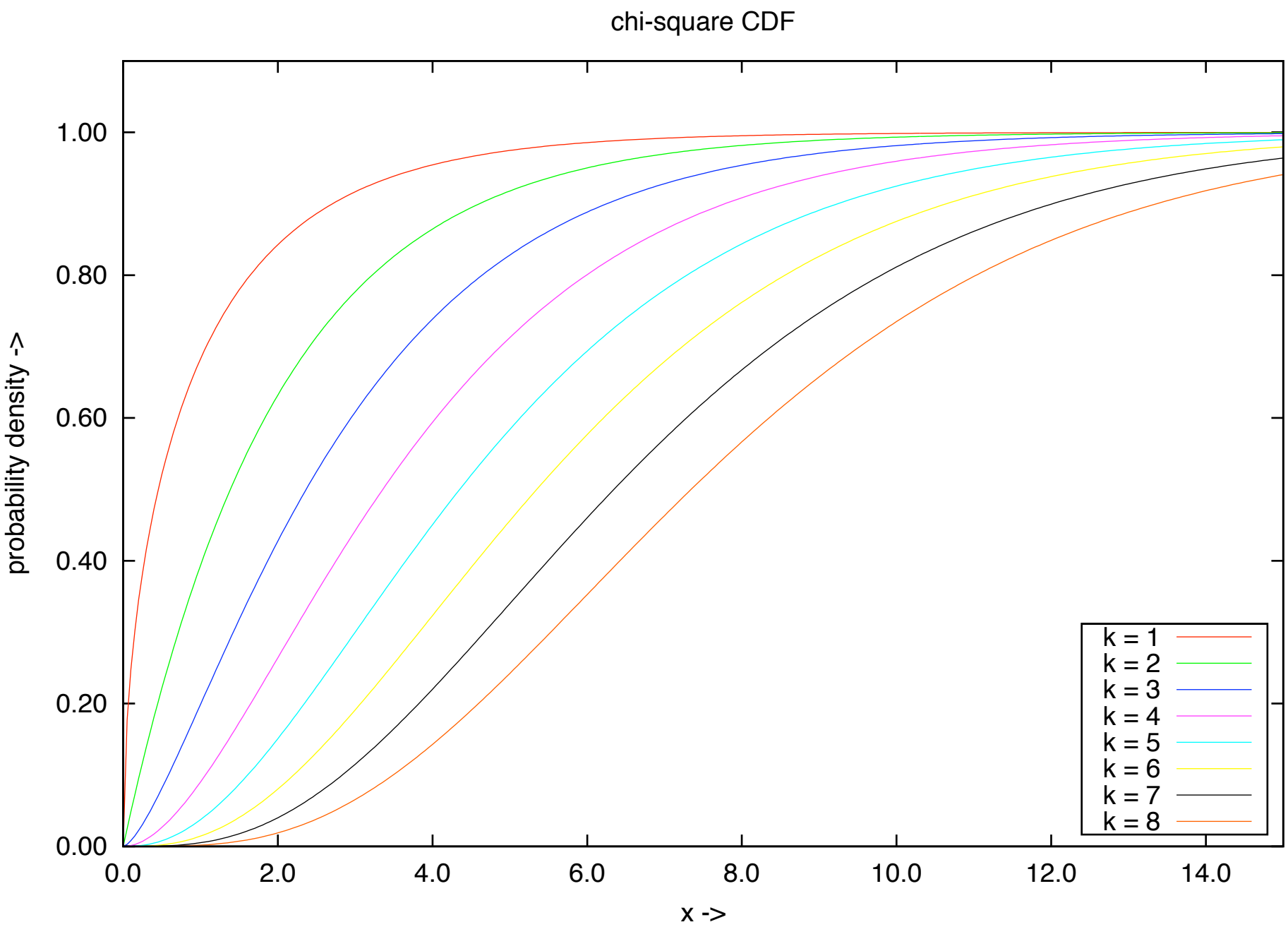

binomial PDF with n = 25, p = 0.15
probability density ->
k ->
0.24
0.22
0.19
0.17
0.14
0.12
0.10
0.07
0.05
0.02
0.00
0
5
10
15
20
25

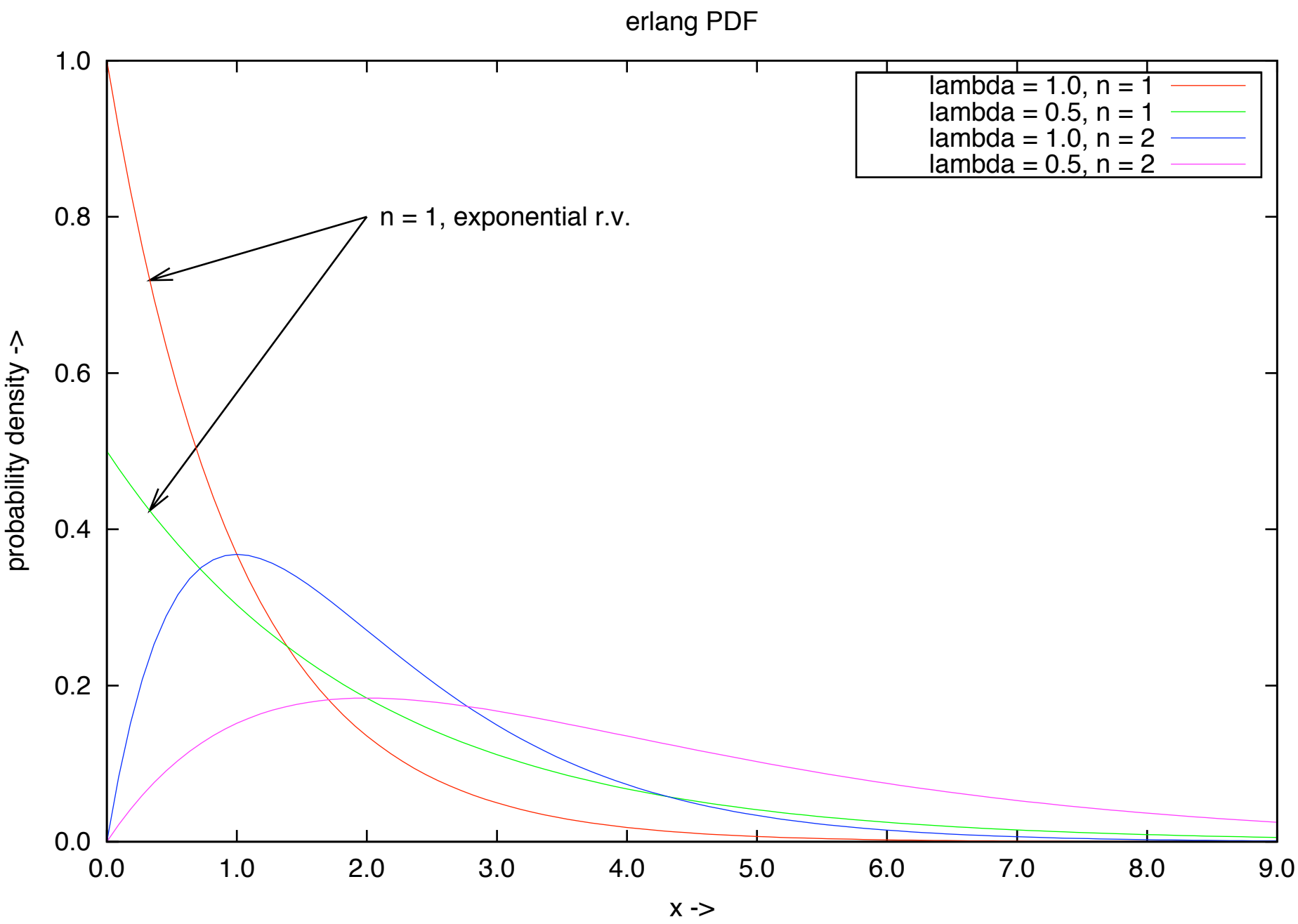

binomial CDF with n = 25, p = 0.15
probability density ->
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0
5
10
15
20
25
k ->

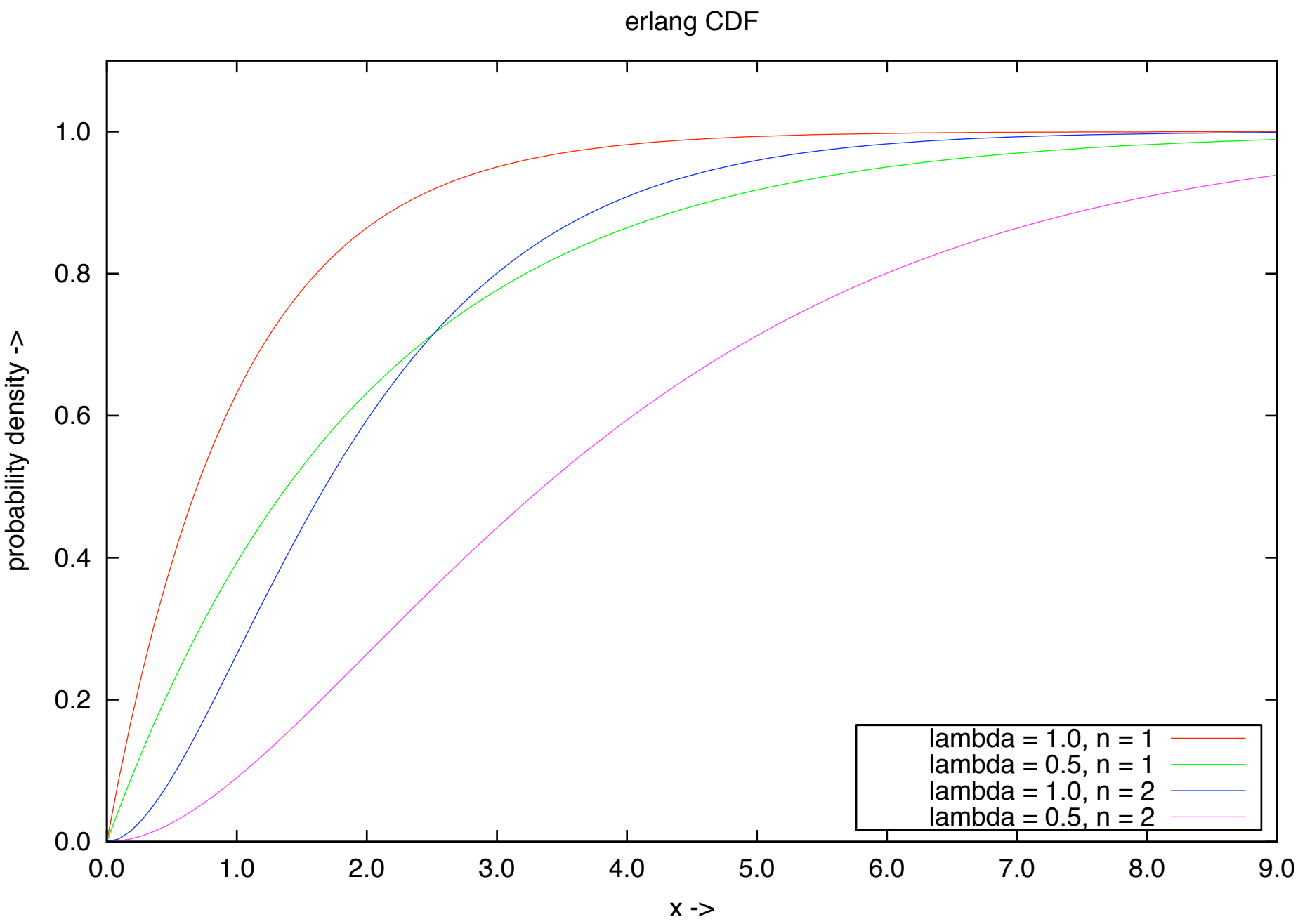

cauchy PDF
a = 0, b = 2
a = 0, b = 4
probability density ->
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
-10.0
-5.0
0.0
5.0
10.0
x ->

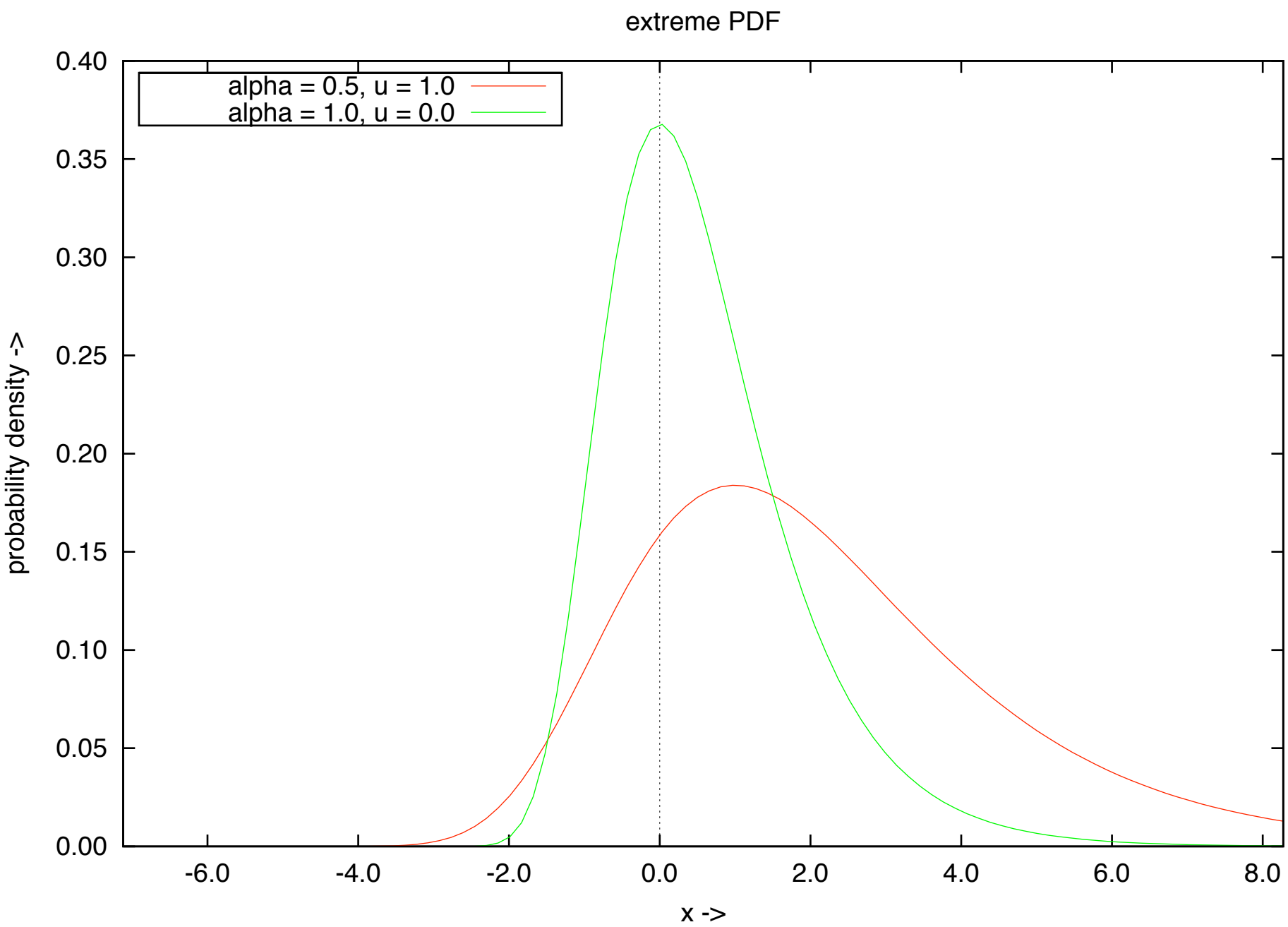

cauchy CDF
a = 0, b = 2
a = 0, b = 4
1.00
0.80
0.60
0.40
0.20
0.00
probability density ->
-10.0
-5.0
0.0
5.0
10.0
x ->

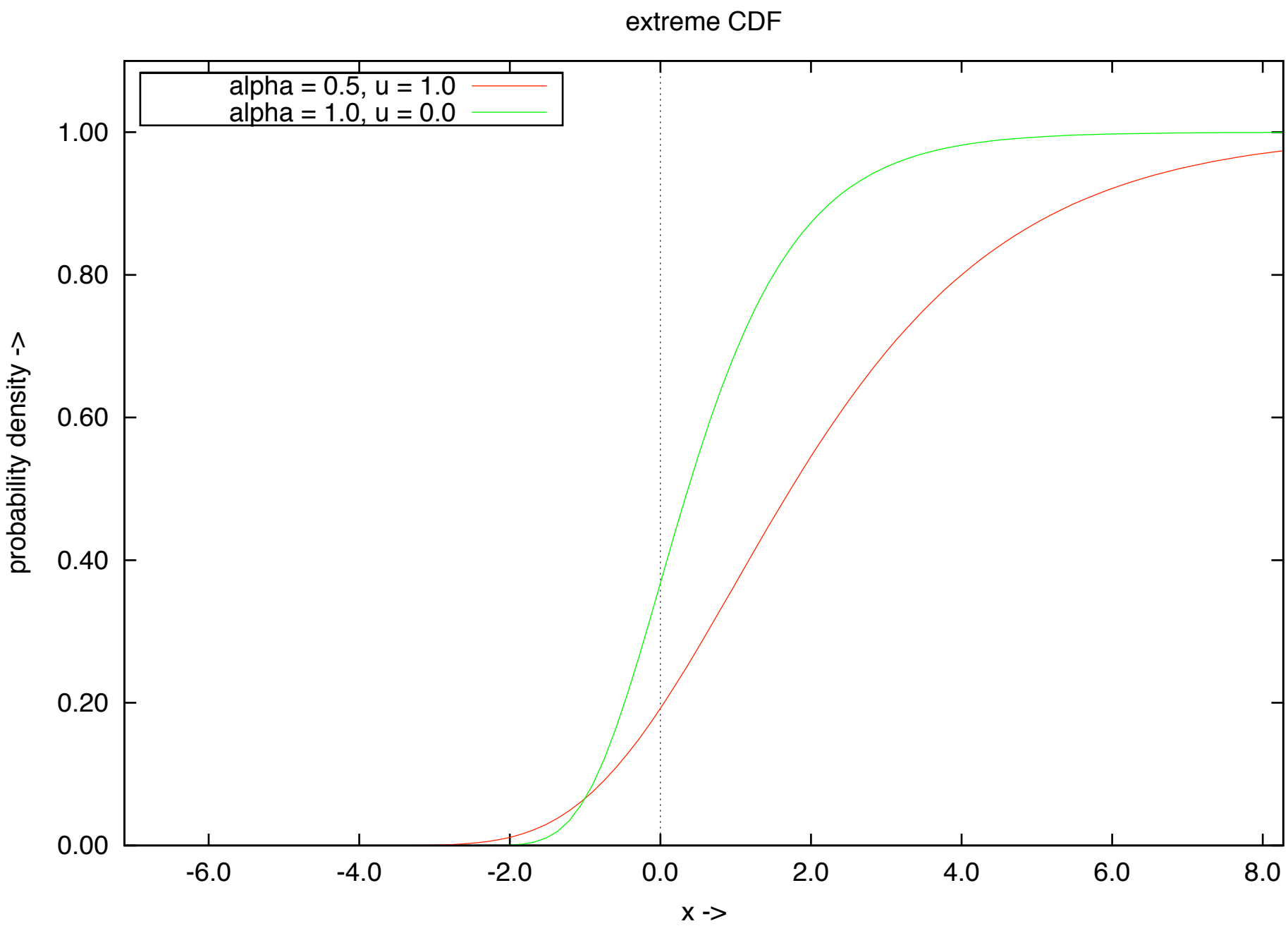

chi-square PDF
probability density ->
x ->
0.00
0.10
0.20
0.30
0.40
0.50
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
k = 1
k = 2
k = 3
k = 4
k = 5
k = 6
k = 7
k = 8

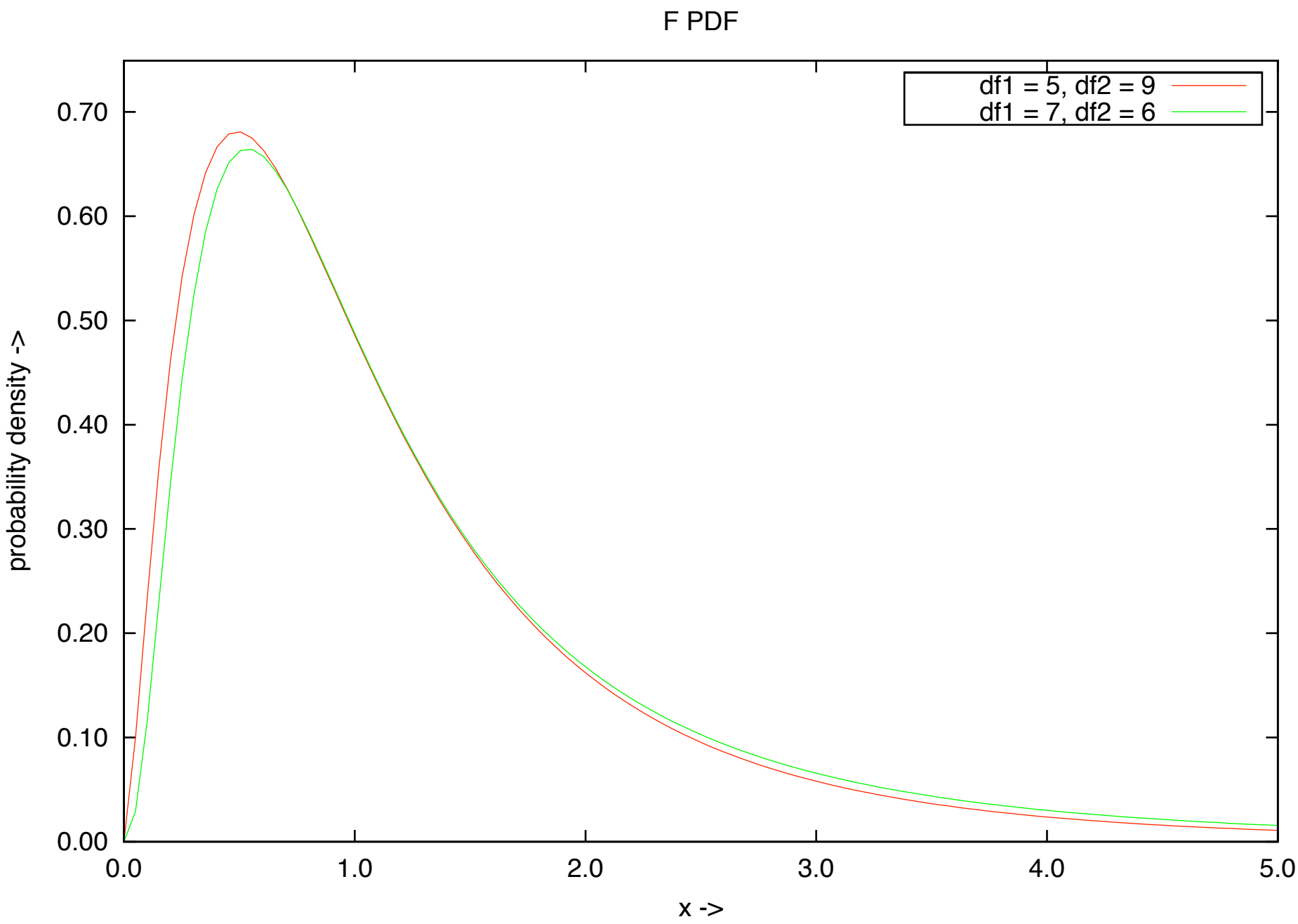

chi-square CDF
probability density ->
x ->
0.00
0.20
0.40
0.60
0.80
1.00
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
k = 1
k = 2
k = 3
k = 4
k = 5
k = 6
k = 7
k = 8

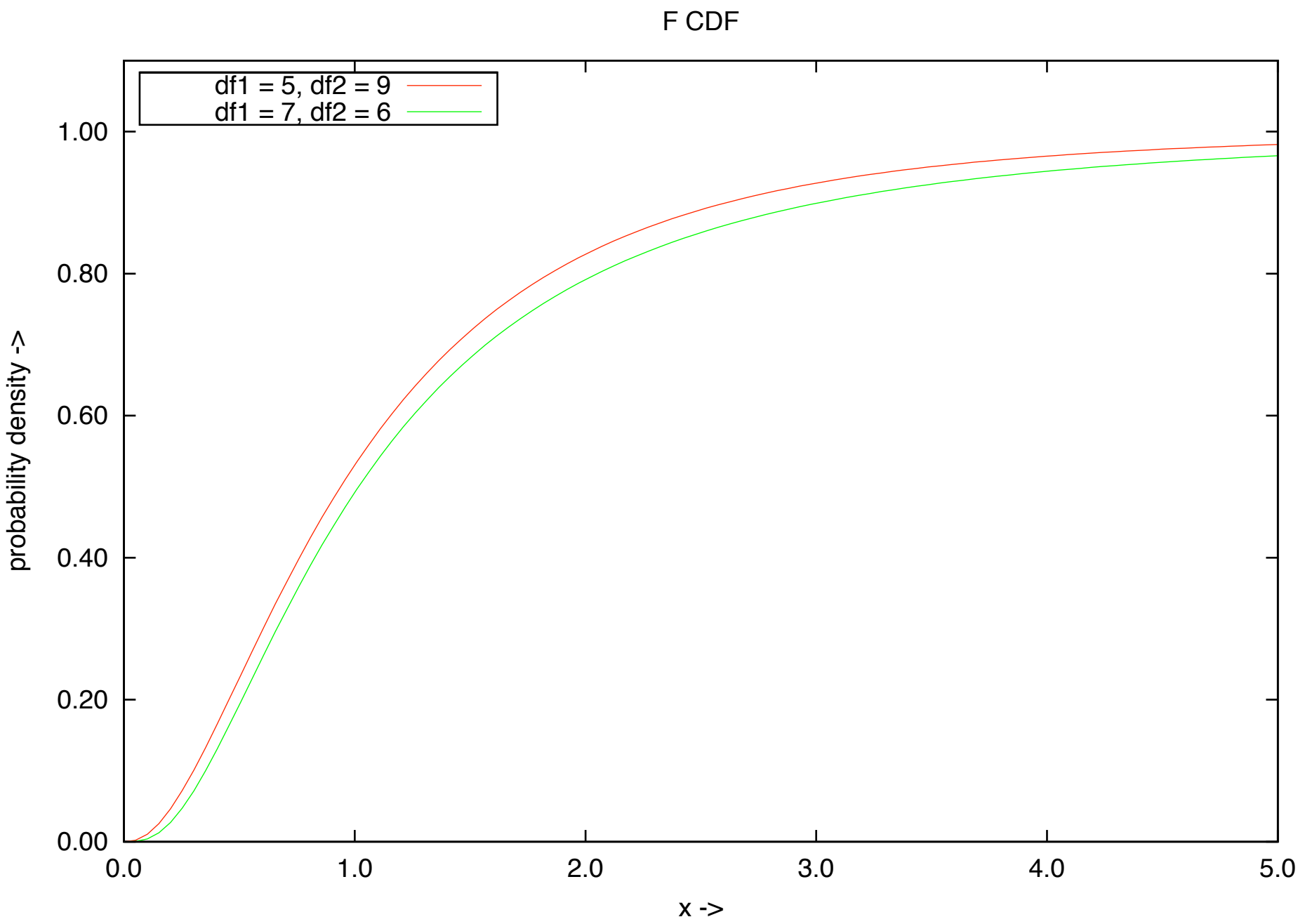

erlang PDF
lambda = 1.0, n = 1
lambda = 0.5, n = 1
lambda = 1.0, n = 2
lambda = 0.5, n = 2
n = 1, exponential r.v.
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0

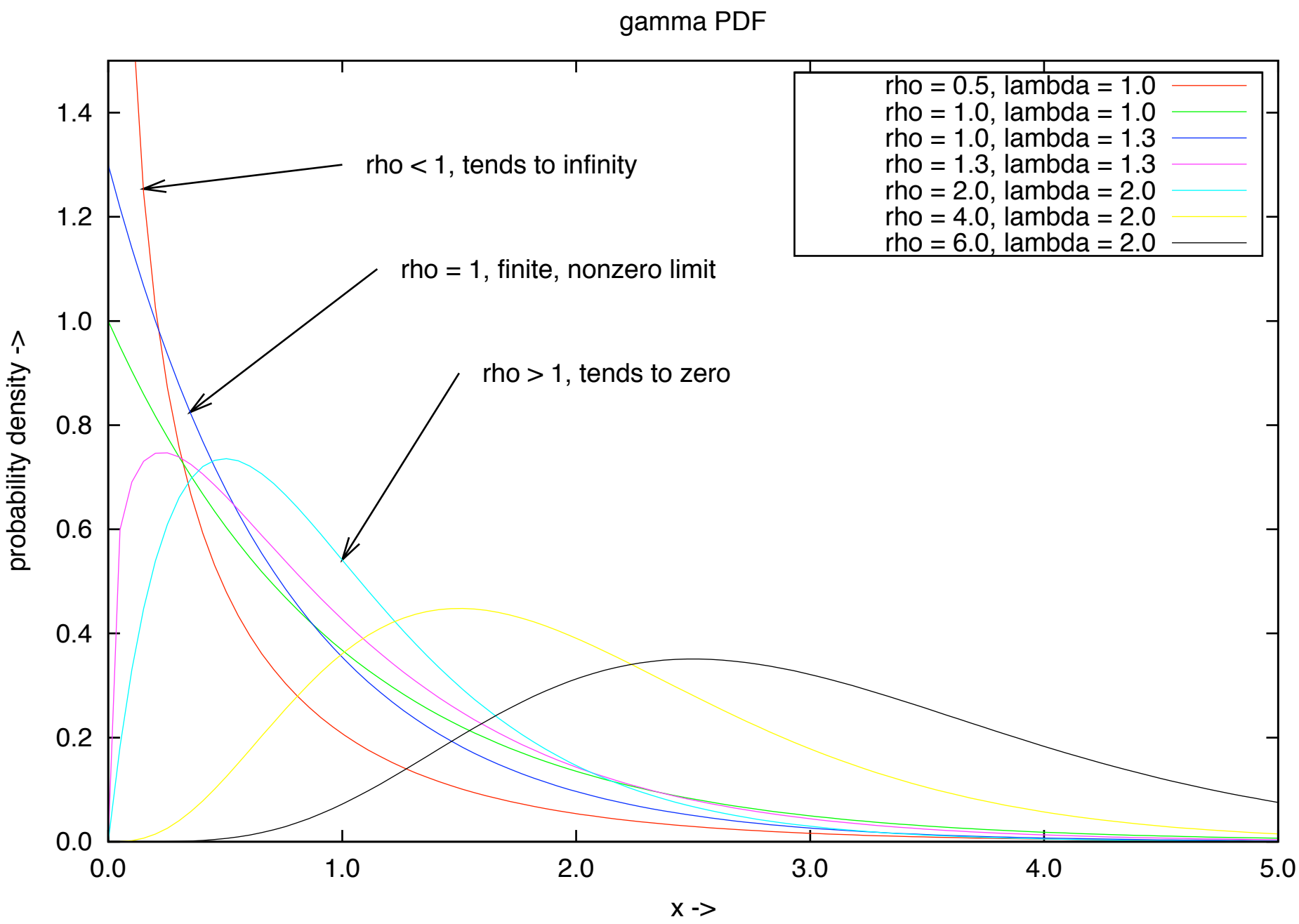

erlang CDF
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
lambda = 1.0, n = 1
lambda = 0.5, n = 1
lambda = 1.0, n = 2
lambda = 0.5, n = 2

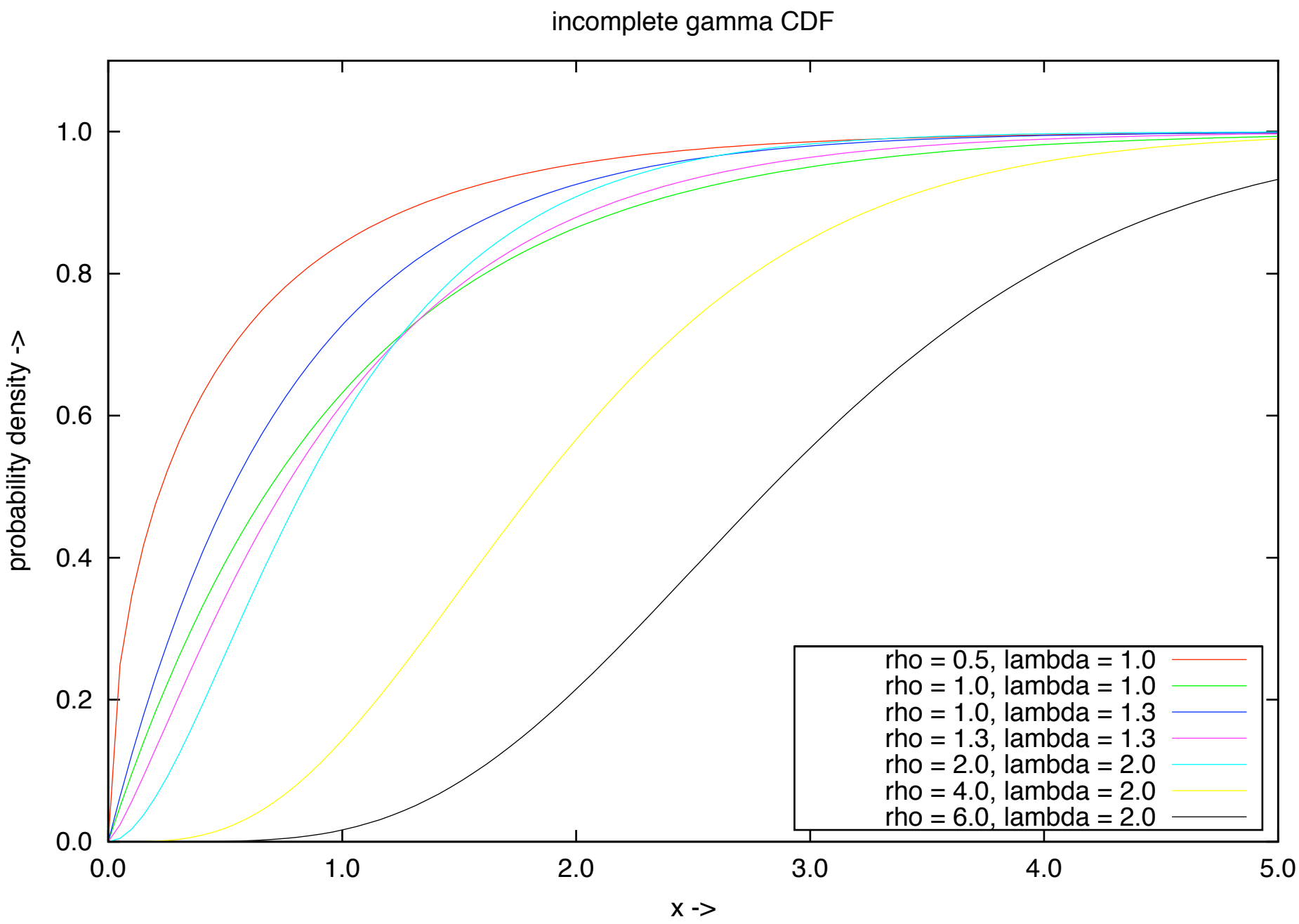

extreme PDF
alpha = 0.5, u = 1.0
alpha = 1.0, u = 0.0
probability density ->
x ->
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-6.0
-4.0
-2.0
0.0
2.0
4.0
6.0
8.0

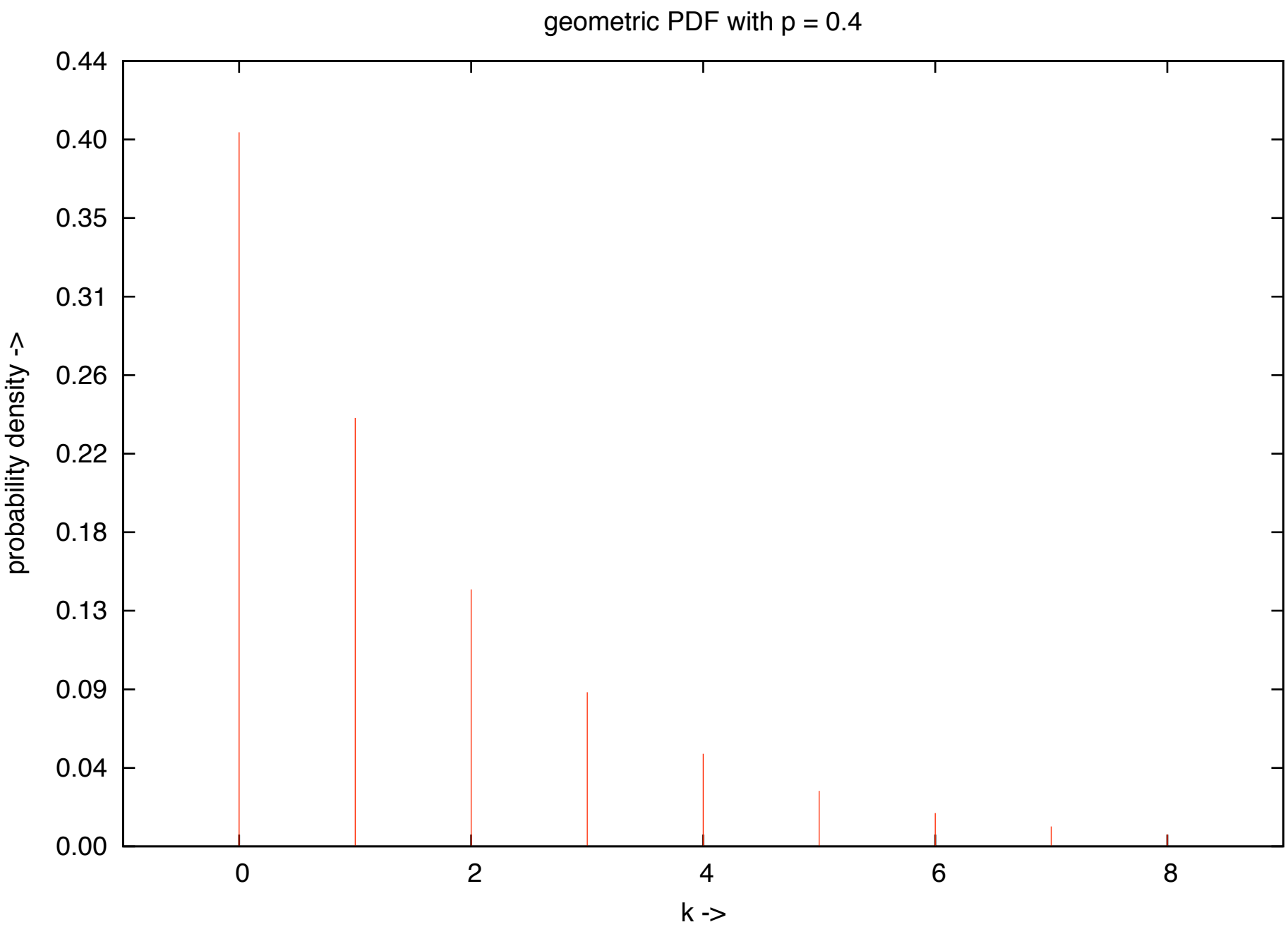

extreme CDF
alpha = 0.5, u = 1.0
alpha = 1.0, u = 0.0
probability density ->
x ->
1.00
0.80
0.60
0.40
0.20
0.00
-6.0
-4.0
-2.0
0.0
2.0
4.0
6.0
8.0

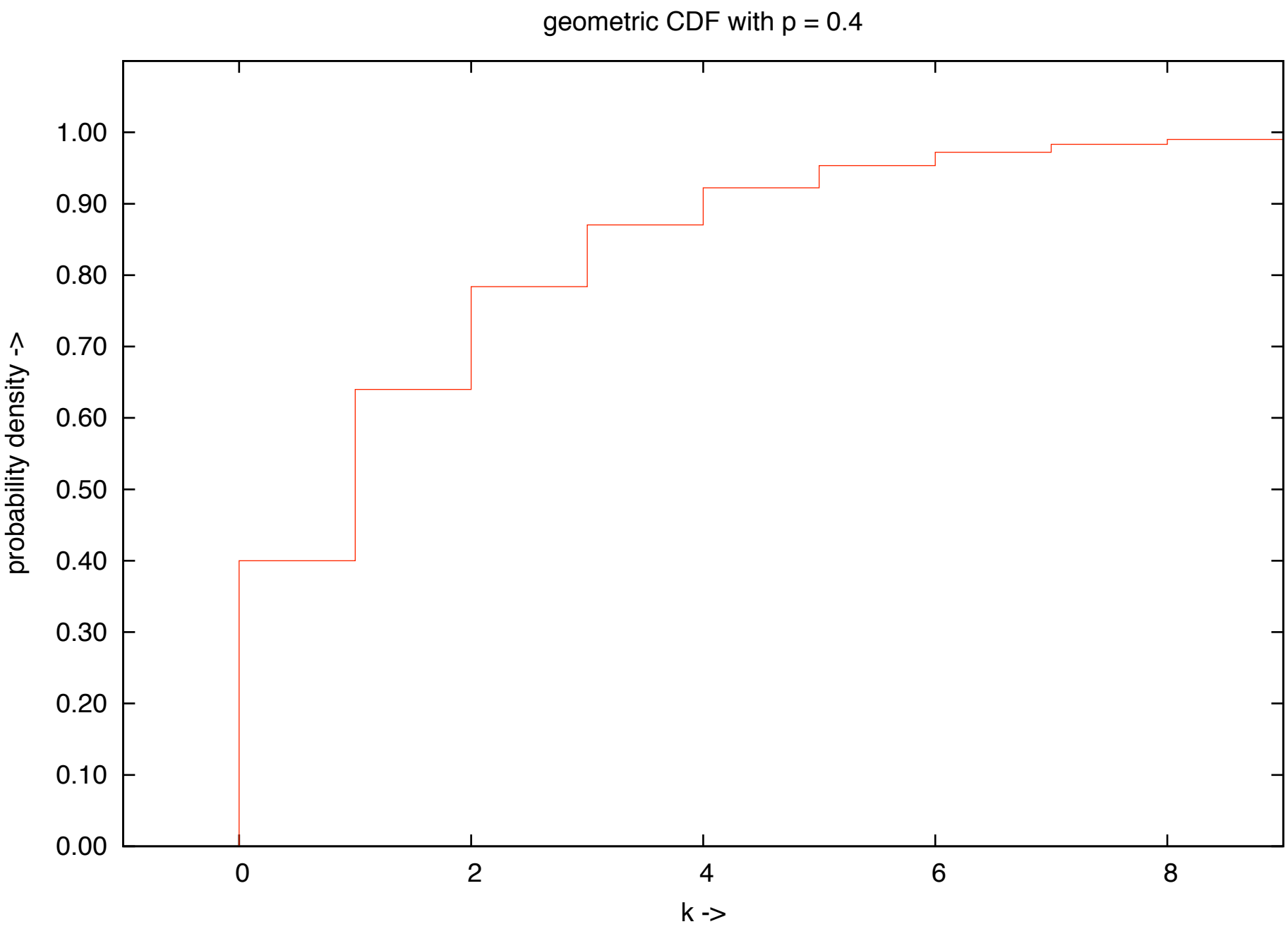

F PDF
df1 = 5, df2 = 9
df1 = 7, df2 = 6
probability density ->
x ->
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.0
1.0
2.0
3.0
4.0
5.0

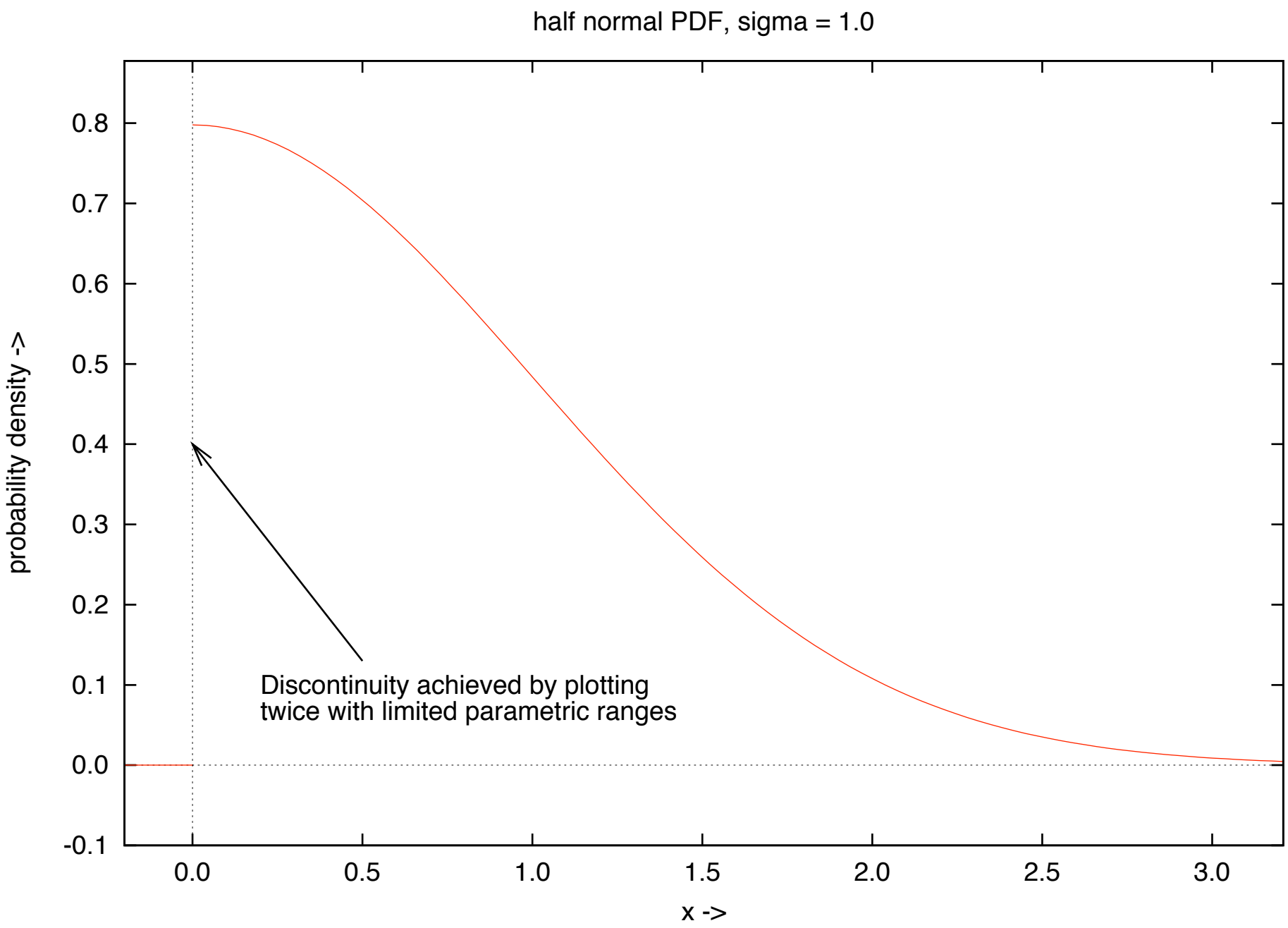

F CDF
df1 = 5, df2 = 9
df1 = 7, df2 = 6
probability density ->
x ->
0.00
0.20
0.40
0.60
0.80
1.00
0.0
1.0
2.0
3.0
4.0
5.0

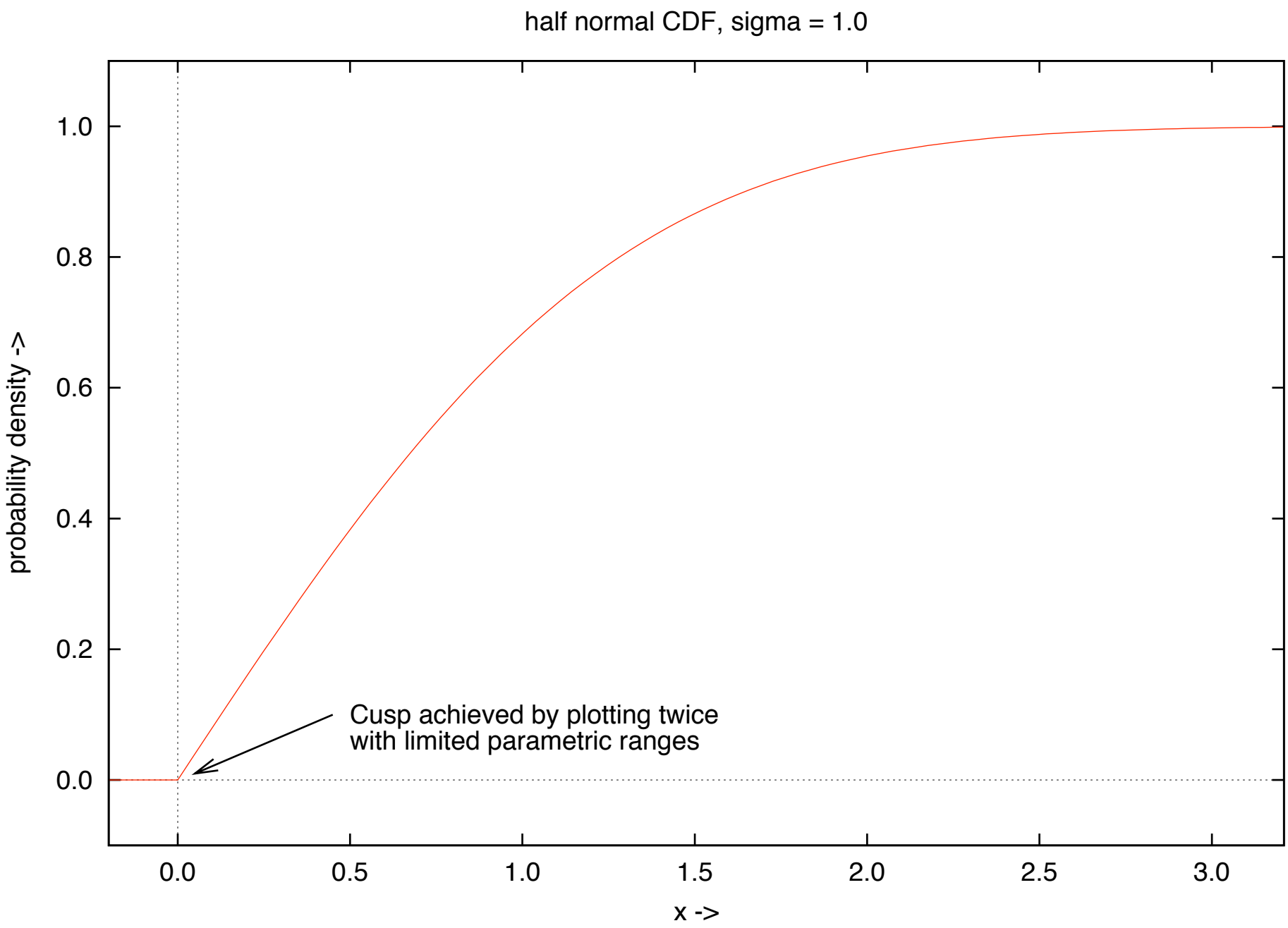

gamma PDF
rho = 0.5, lambda = 1.0
rho = 1.0, lambda = 1.0
rho = 1.0, lambda = 1.3
rho = 1.3, lambda = 1.3
rho = 2.0, lambda = 2.0
rho = 4.0, lambda = 2.0
rho = 6.0, lambda = 2.0
rho < 1, tends to infinity
rho = 1, finite, nonzero limit
rho > 1, tends to zero
probability density ->
x ->
0.0
0.2
0.4
0.6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

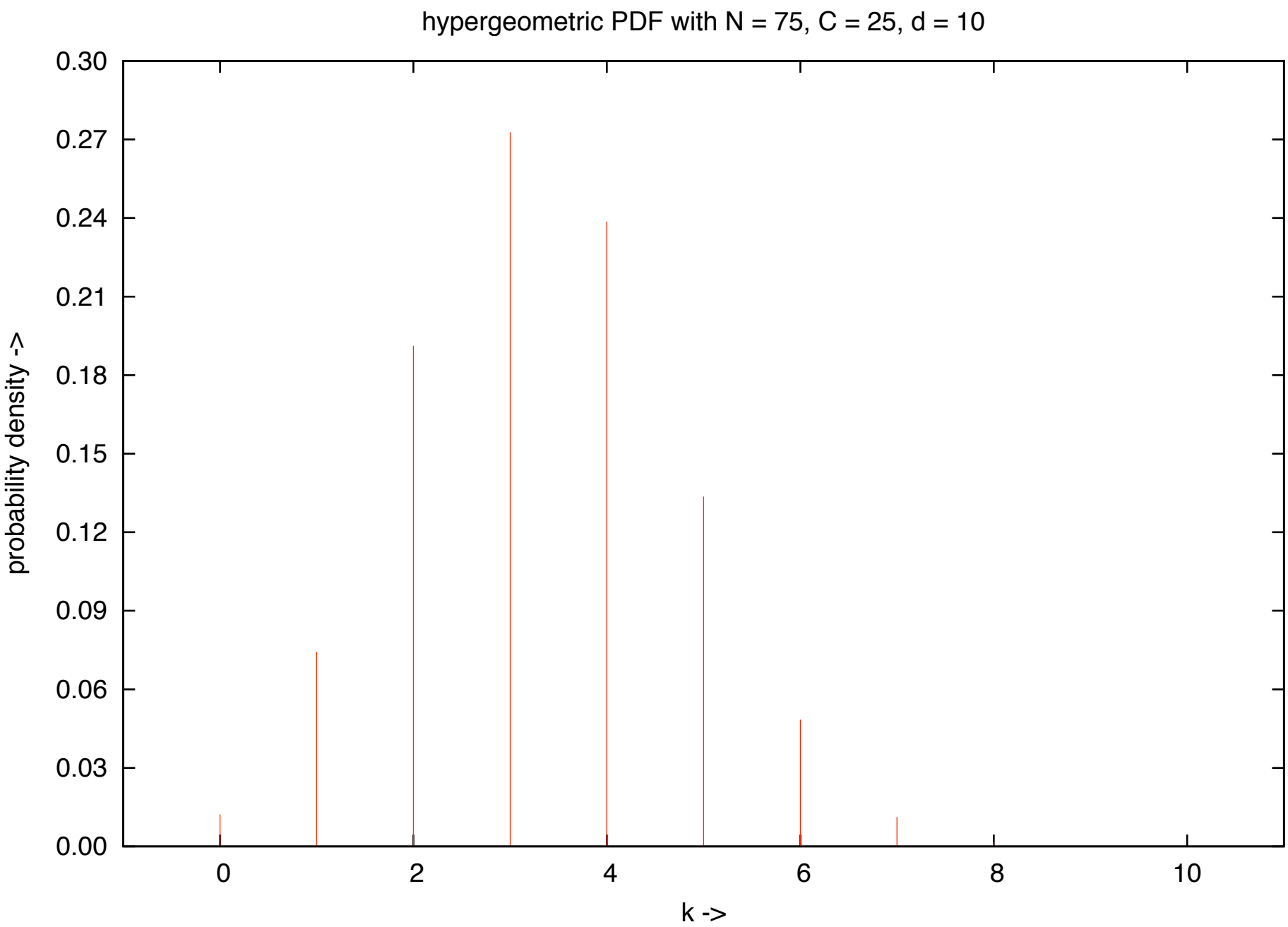

incomplete gamma CDF
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.0
1.0
2.0
3.0
4.0
5.0
rho = 0.5, lambda = 1.0
rho = 1.0, lambda = 1.0
rho = 1.0, lambda = 1.3
rho = 1.3, lambda = 1.3
rho = 2.0, lambda = 2.0
rho = 4.0, lambda = 2.0
rho = 6.0, lambda = 2.0

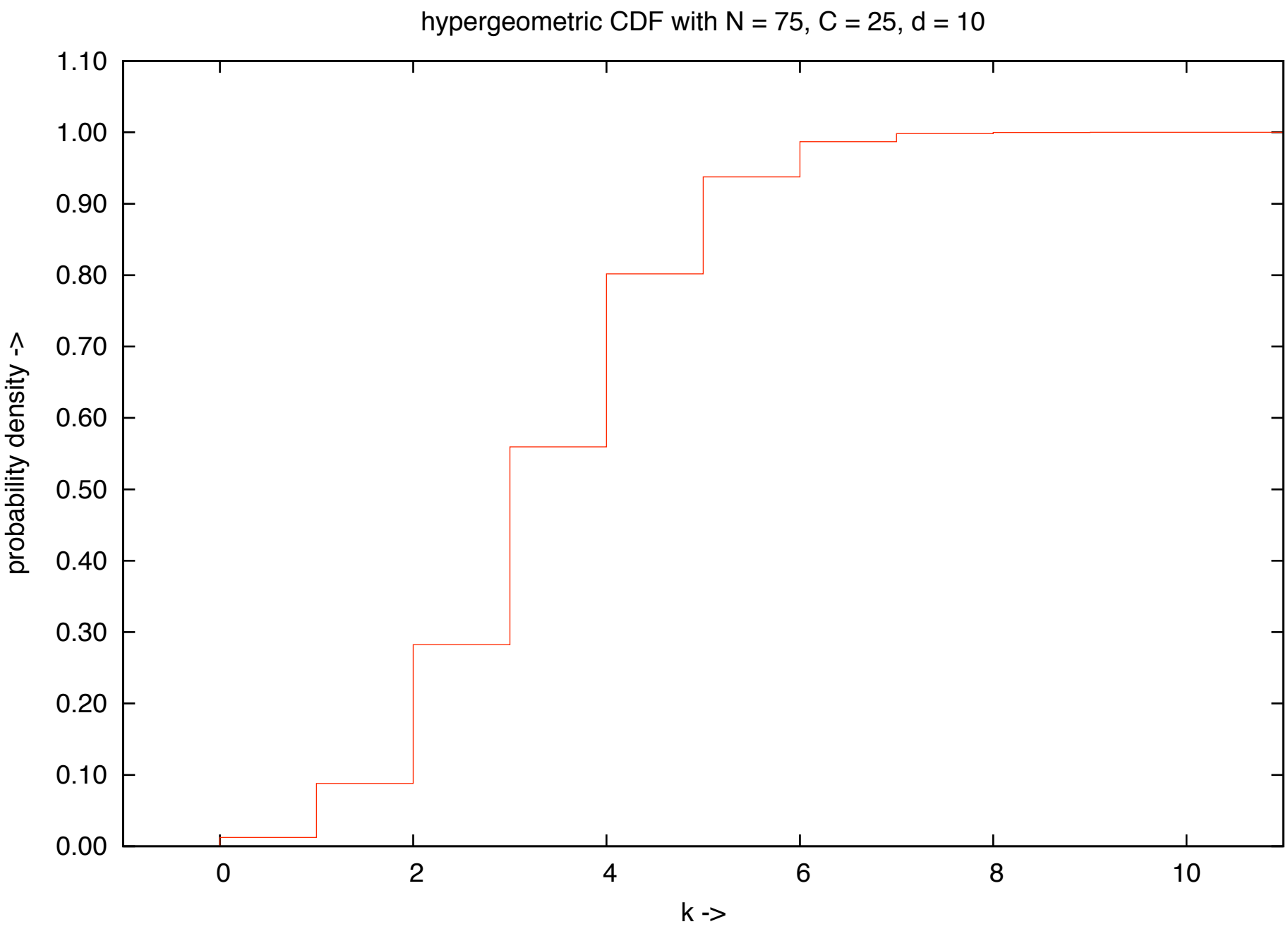

geometric PDF with p = 0.4
probability density ->
k ->
0.44
0.40
0.35
0.31
0.26
0.22
0.18
0.13
0.09
0.04
0.00
0
2
4
6
8

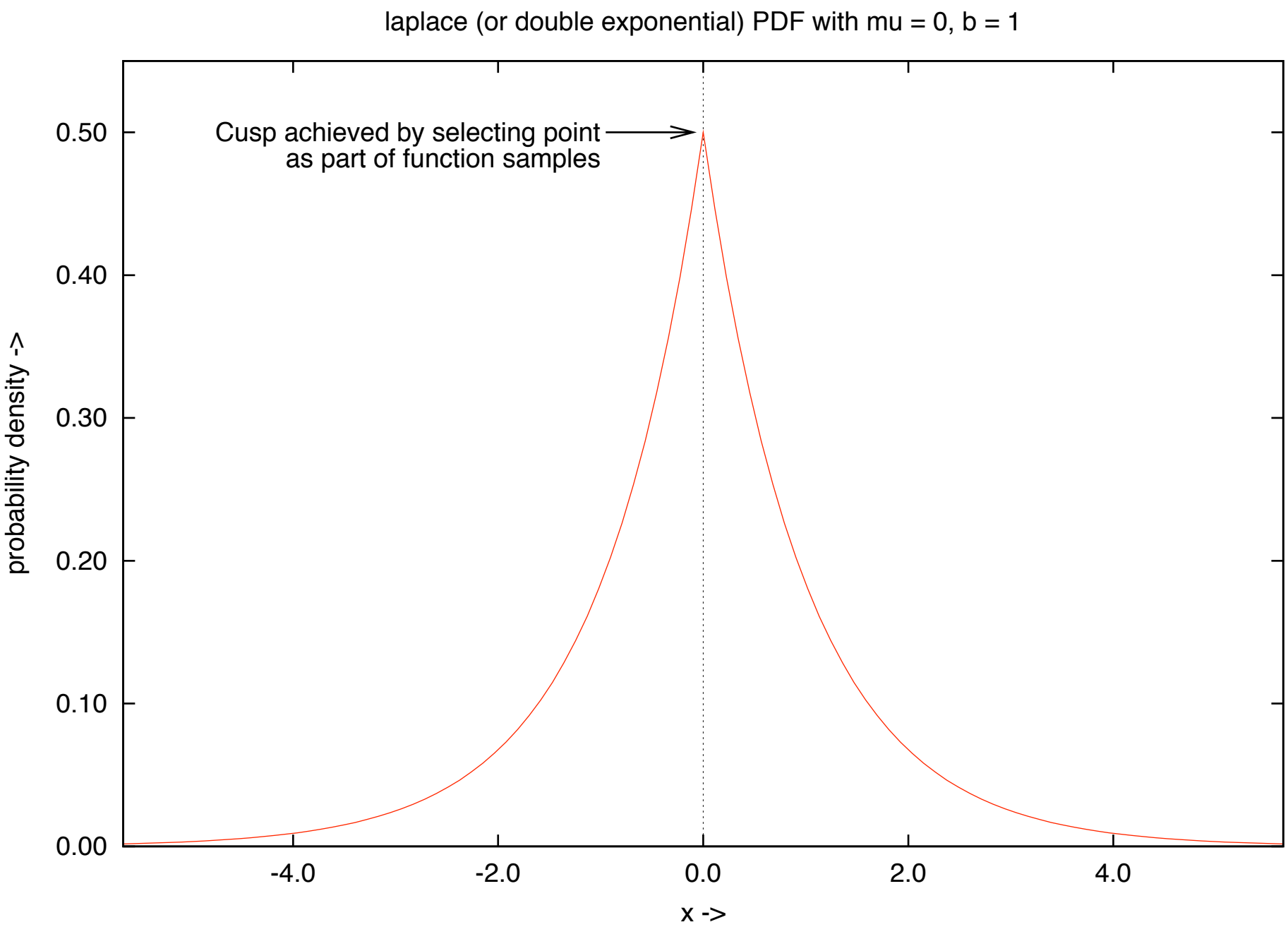

geometric CDF with p = 0.4
probability density ->
k ->
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
0
2
4
6
8

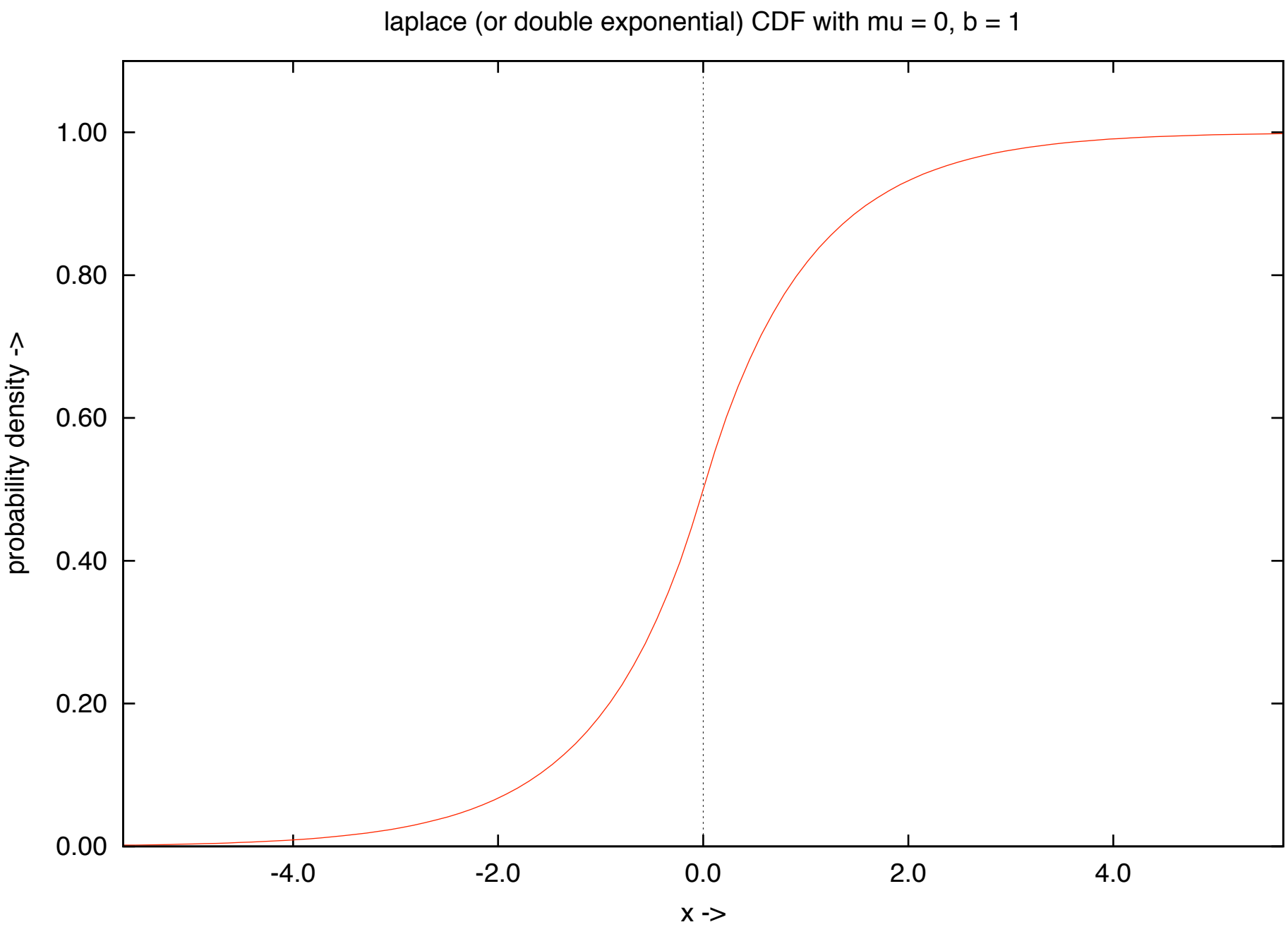

half normal PDF, sigma = 1.0
probability density ->
x ->
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Discontinuity achieved by plotting
twice with limited parametric ranges

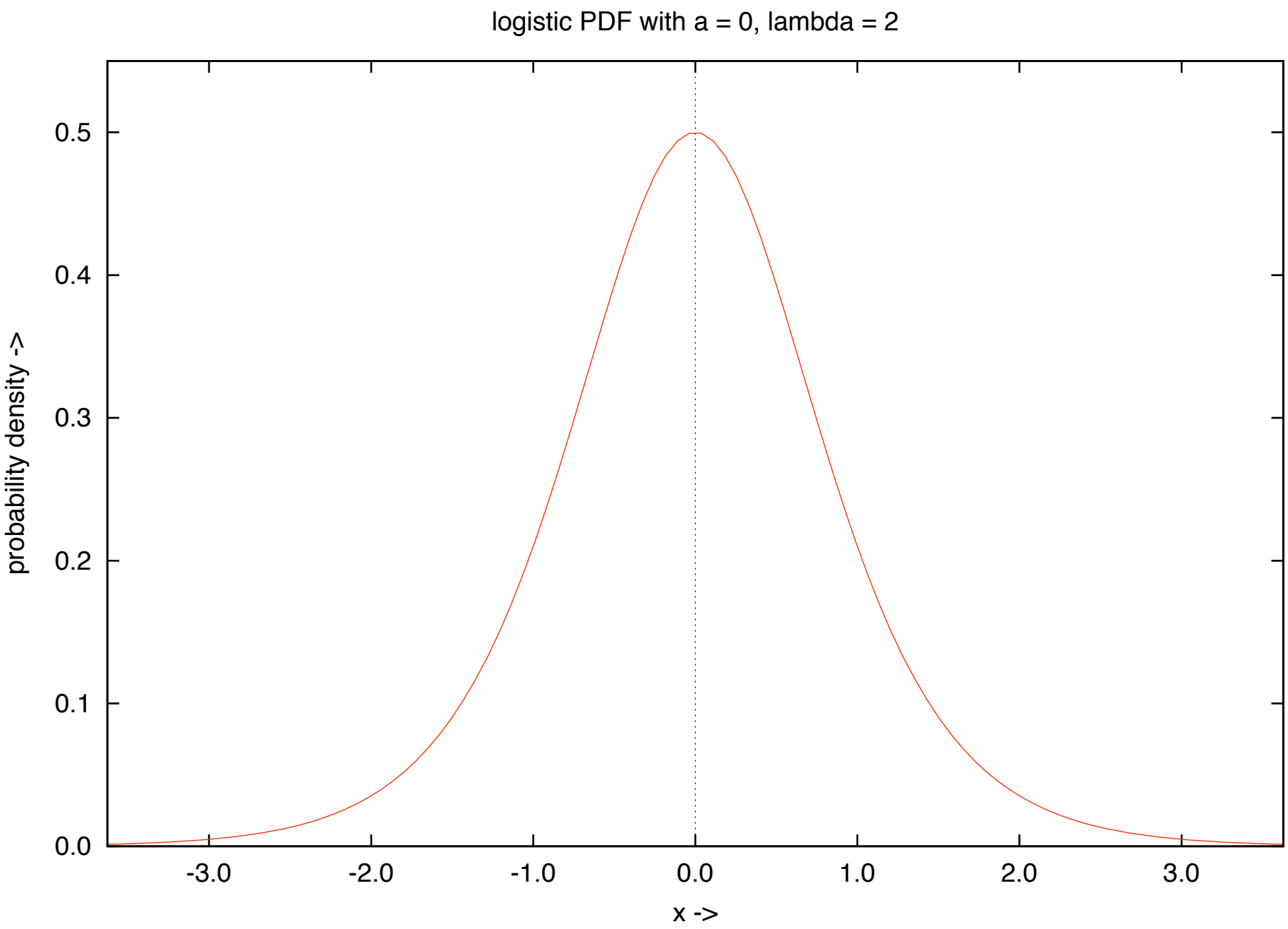

half normal CDF, sigma = 1.0
probability density ->
x ->
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Cusp achieved by plotting twice
with limited parametric ranges

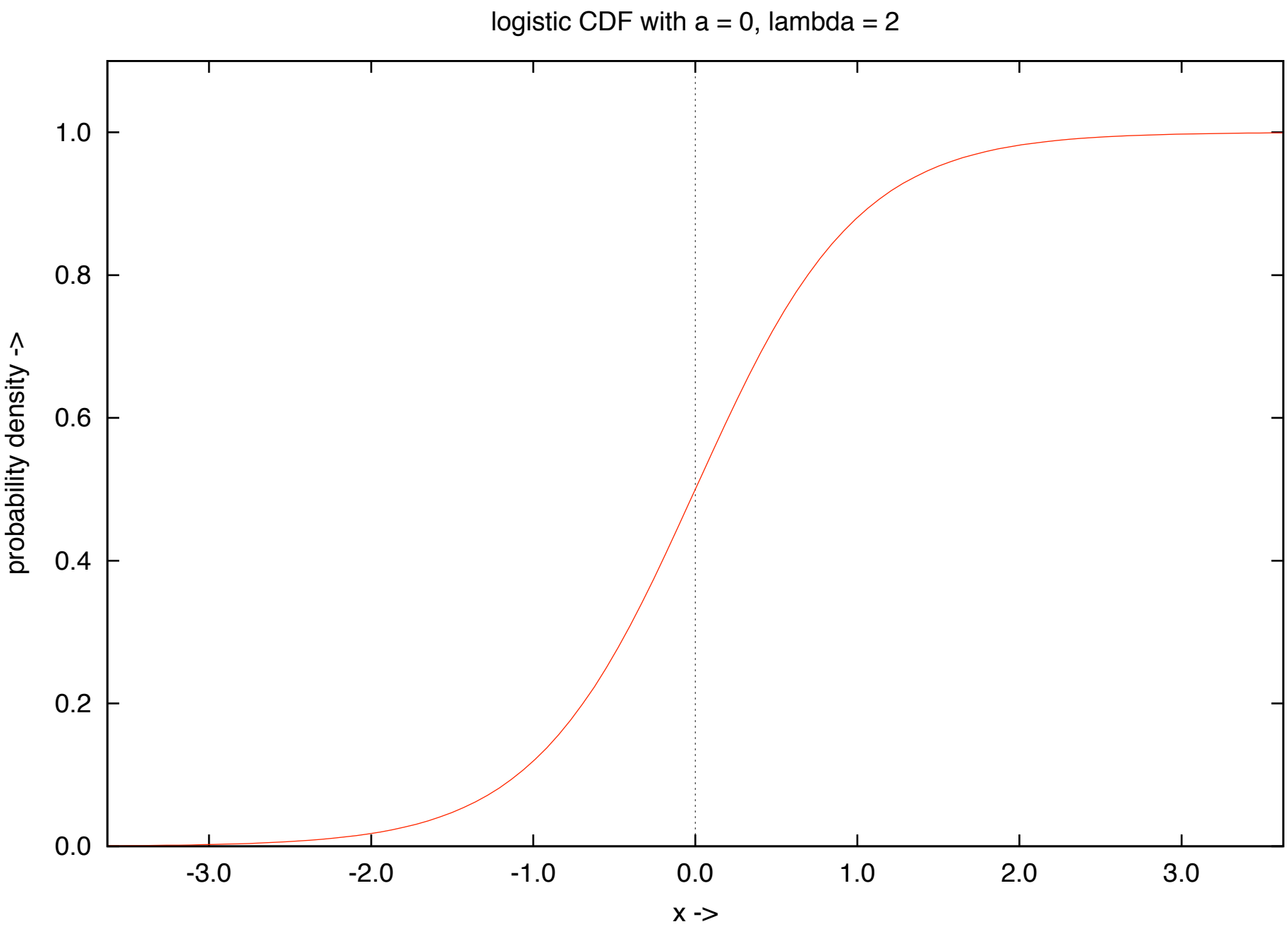

hypergeometric PDF with N = 75, C = 25, d = 10
probability density ->
k ->
0.00
0.03
0.06
0.09
0.12
0.15
0.18
0.21
0.24
0.27
0.30
0
2
4
6
8
10

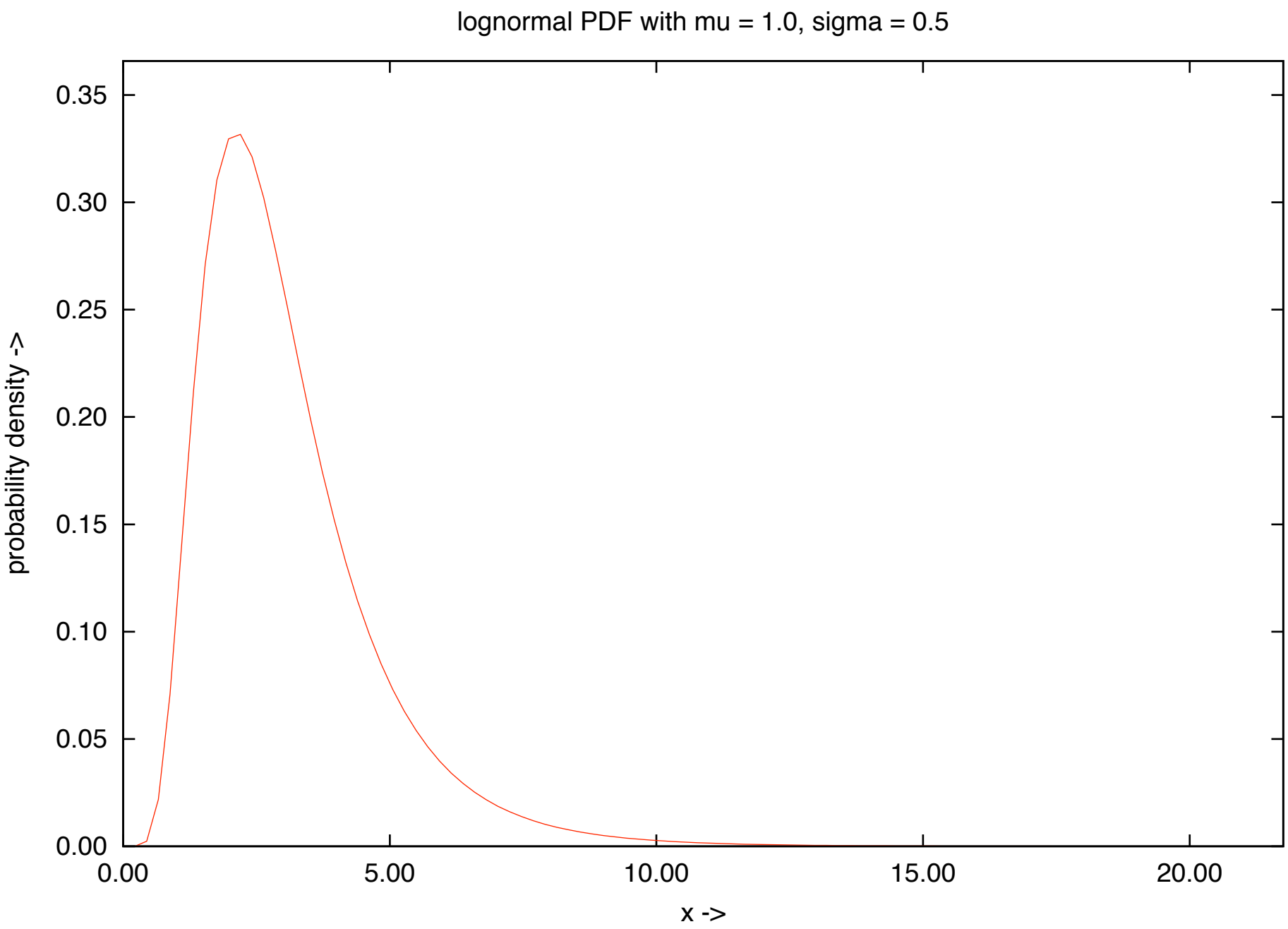

hypergeometric CDF with N = 75, C = 25, d = 10
probability density ->
k ->
1.10
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0
2
4
6
8
10

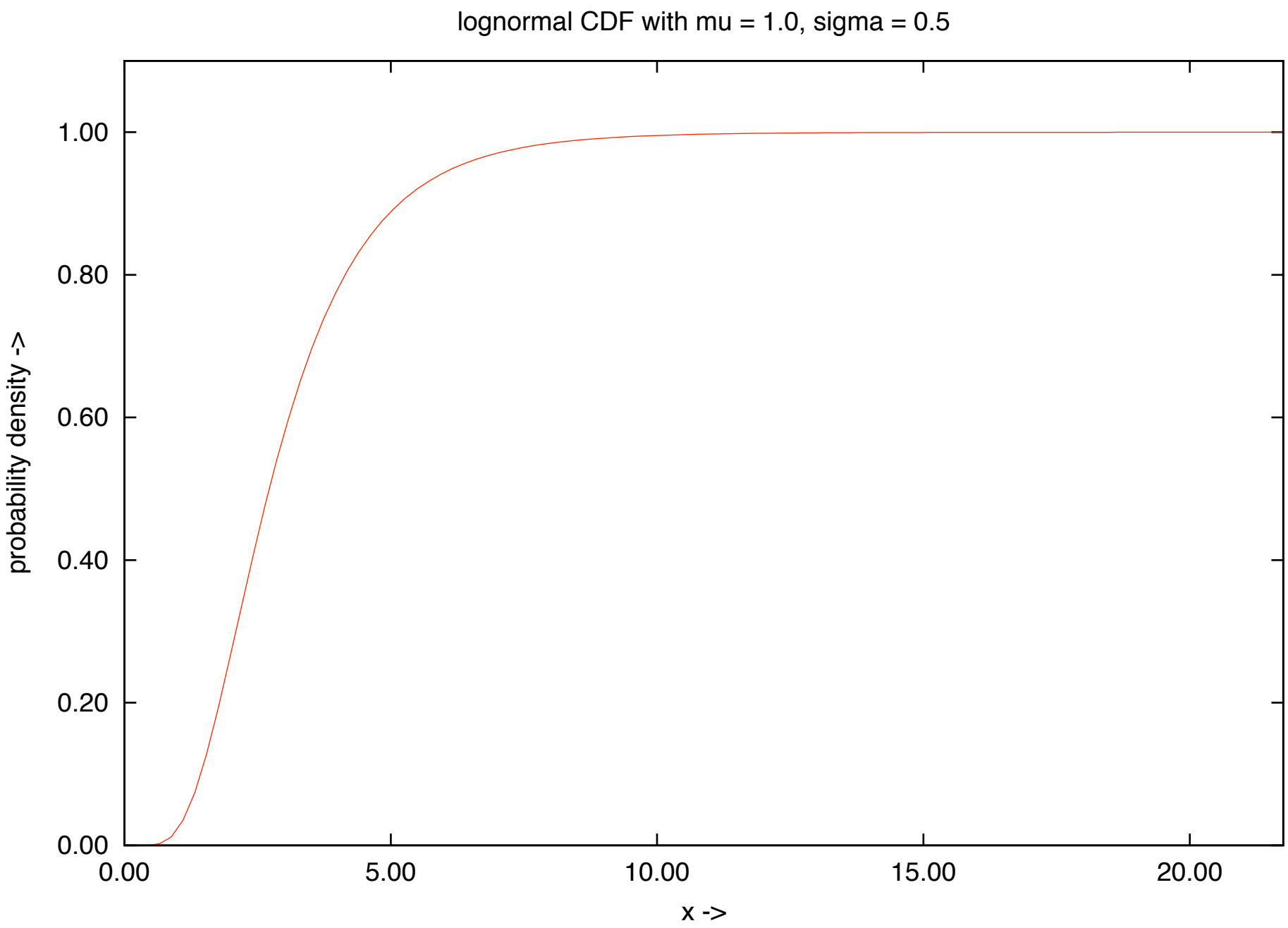

laplace (or double exponential) PDF with mu = 0, b = 1
Cusp achieved by selecting point
as part of function samples
probability density ->
0.50
0.40
0.30
0.20
0.10
0.00
-4.0
-2.0
0.0
2.0
4.0
x ->

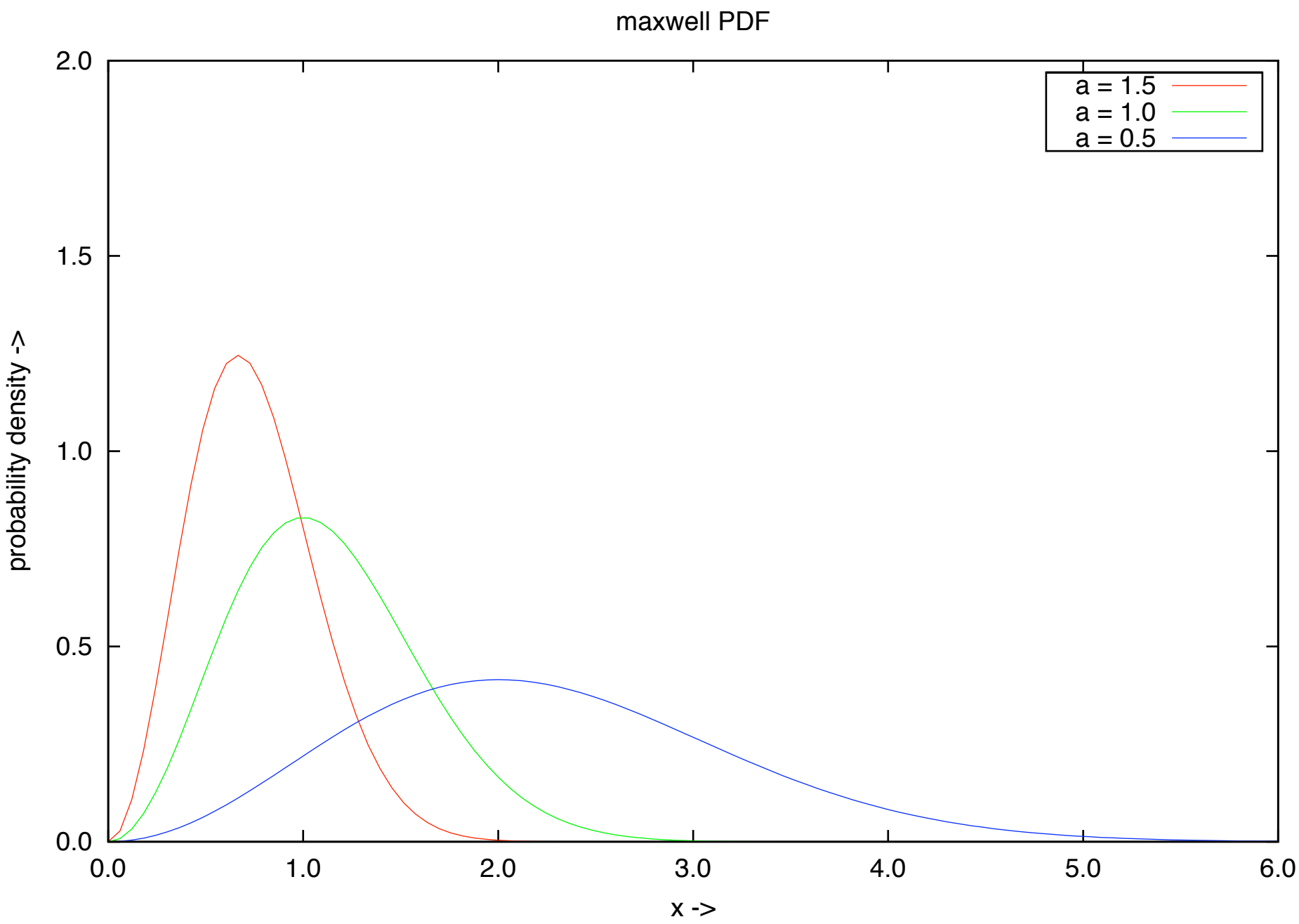

laplace (or double exponential) CDF with mu = 0, b = 1
probability density ->
1.00
0.80
0.60
0.40
0.20
0.00
-4.0
-2.0
0.0
2.0
4.0
x ->

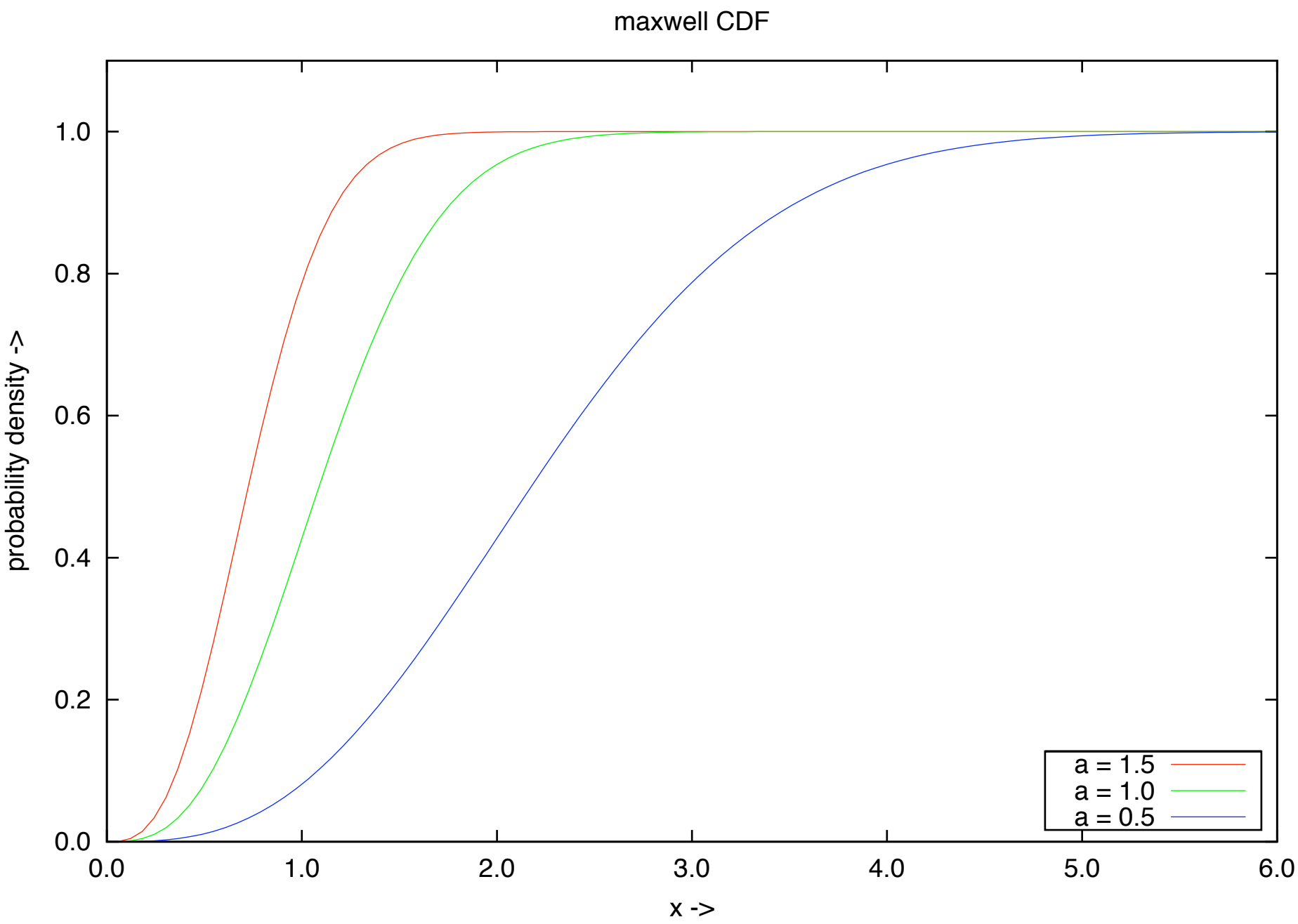

logistic PDF with a = 0, lambda = 2
probability density ->
0.0
0.1
0.2
0.3
0.4
0.5
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
x ->

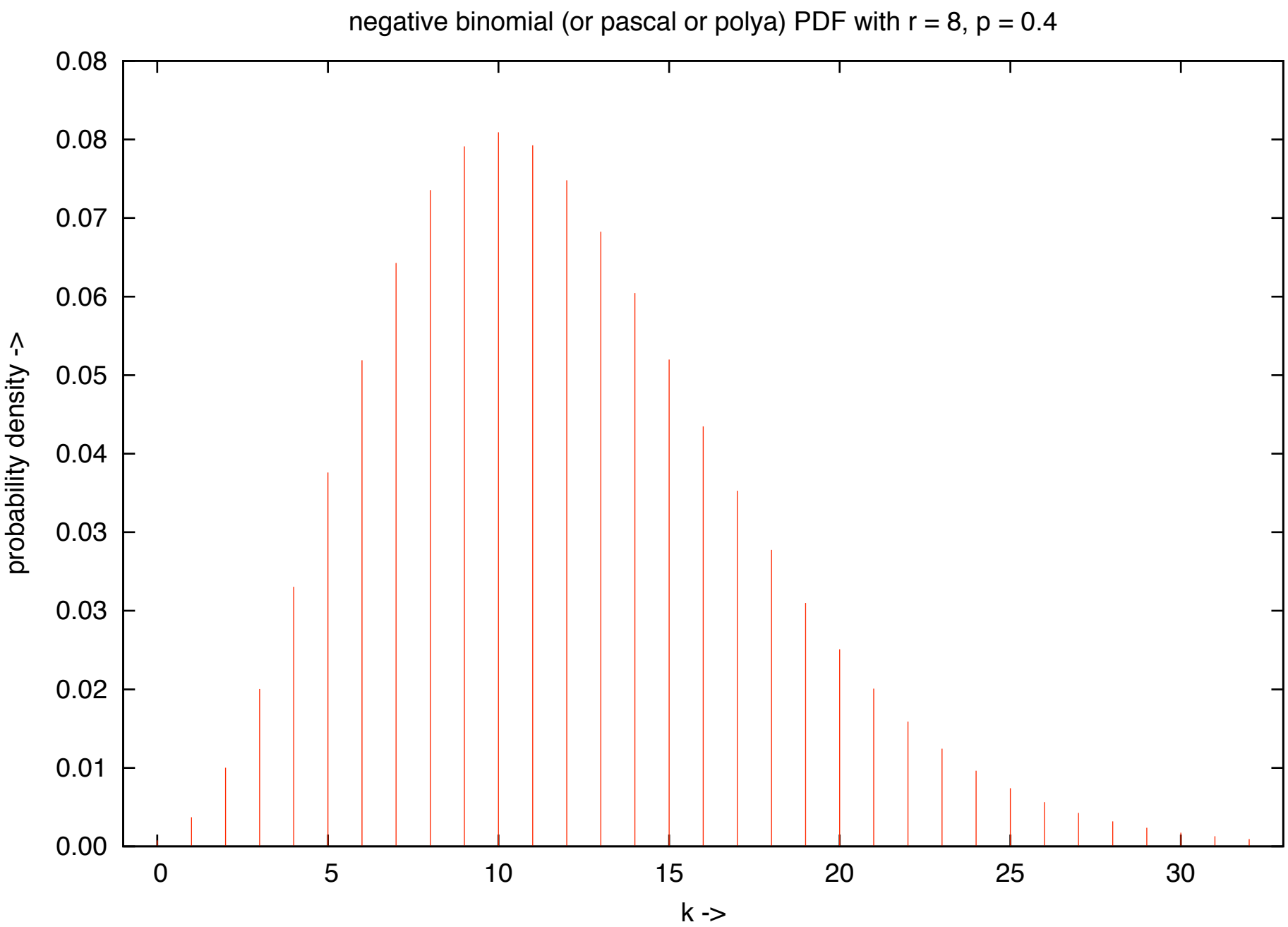

logistic CDF with a = 0, lambda = 2
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

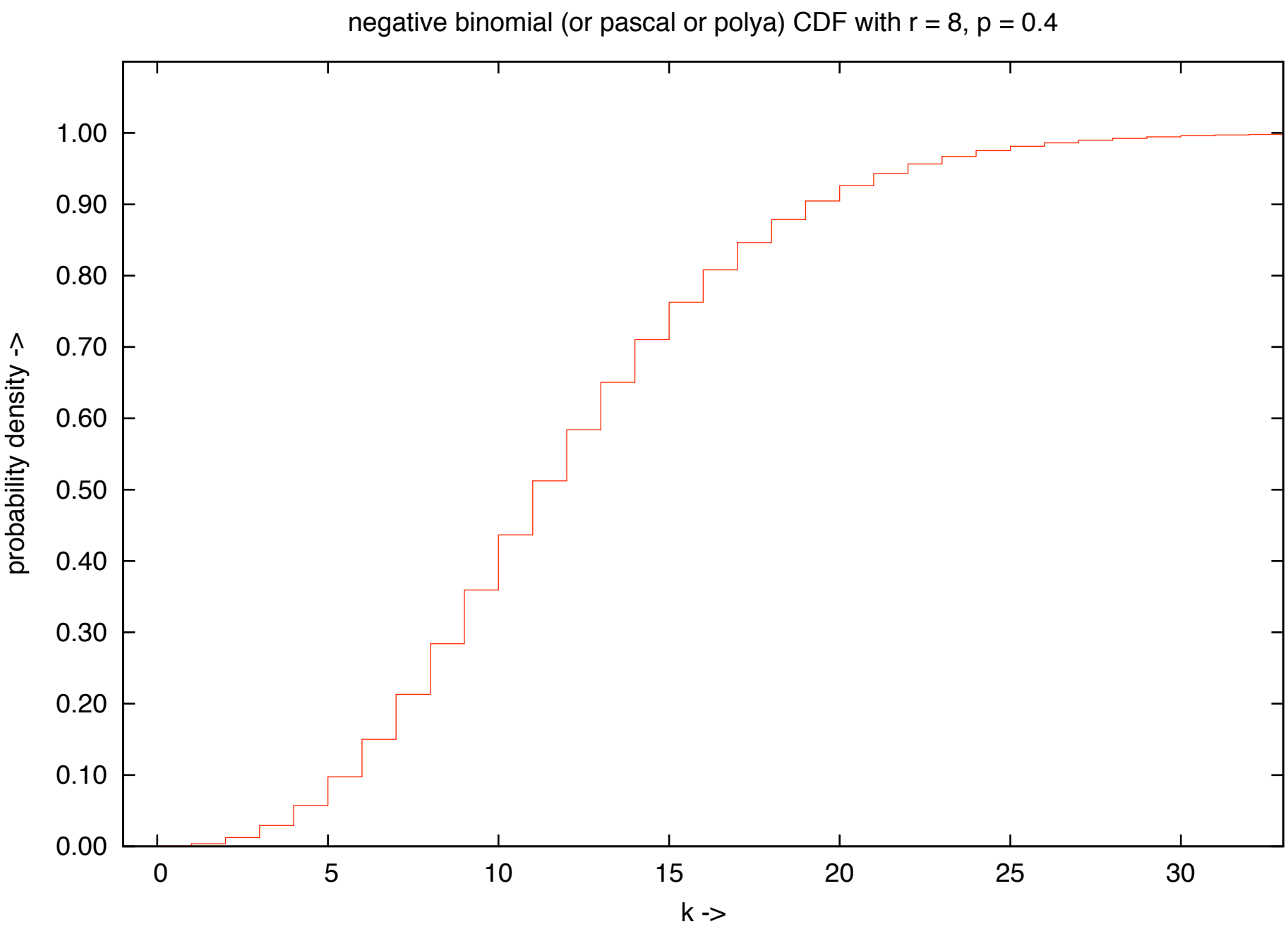

lognormal PDF with mu = 1.0, sigma = 0.5
probability density ->
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.00
5.00
10.00
15.00
20.00
x ->

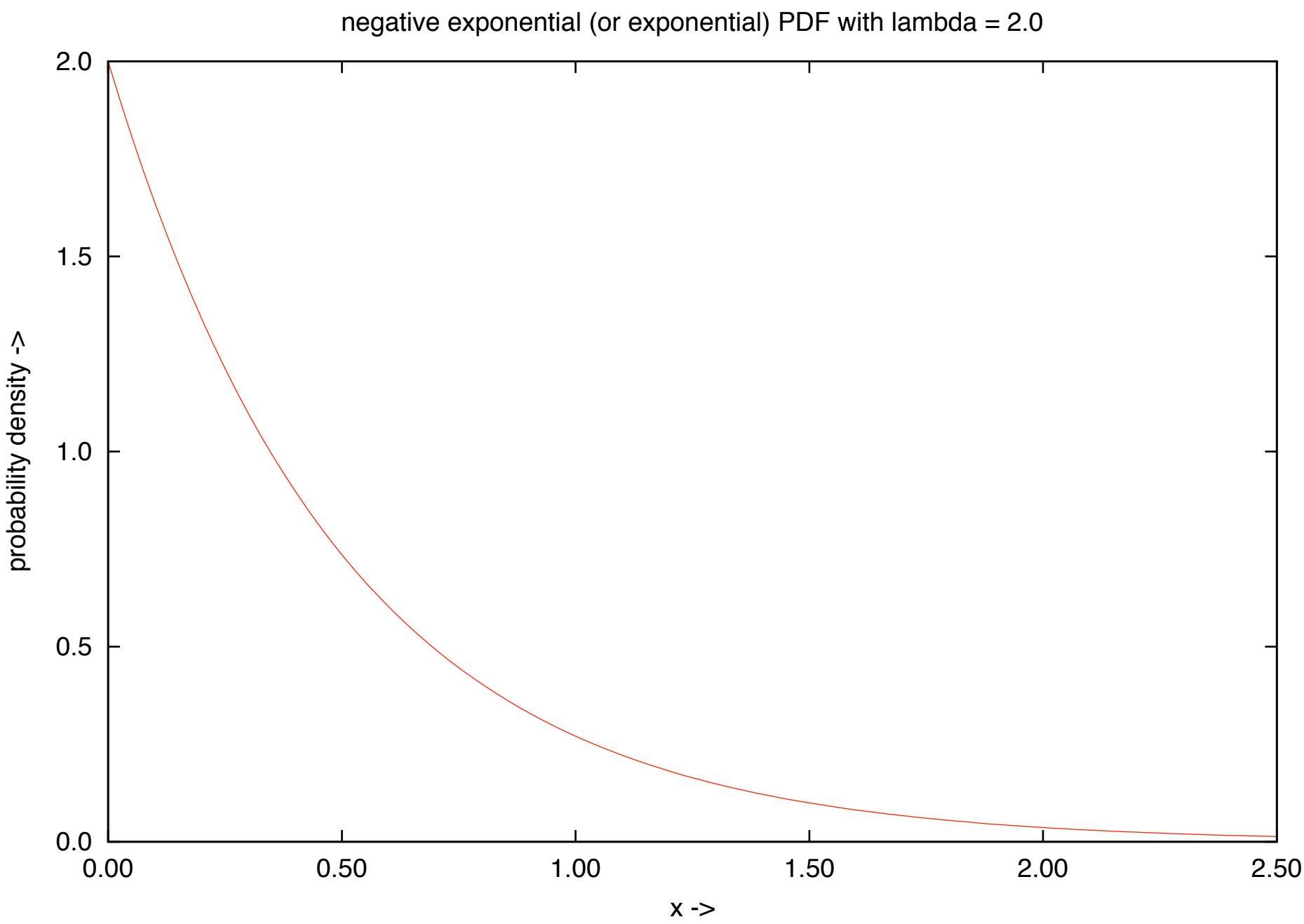

lognormal CDF with mu = 1.0, sigma = 0.5
probability density ->
1.00
0.80
0.60
0.40
0.20
0.00
0.00
5.00
10.00
15.00
20.00
x ->

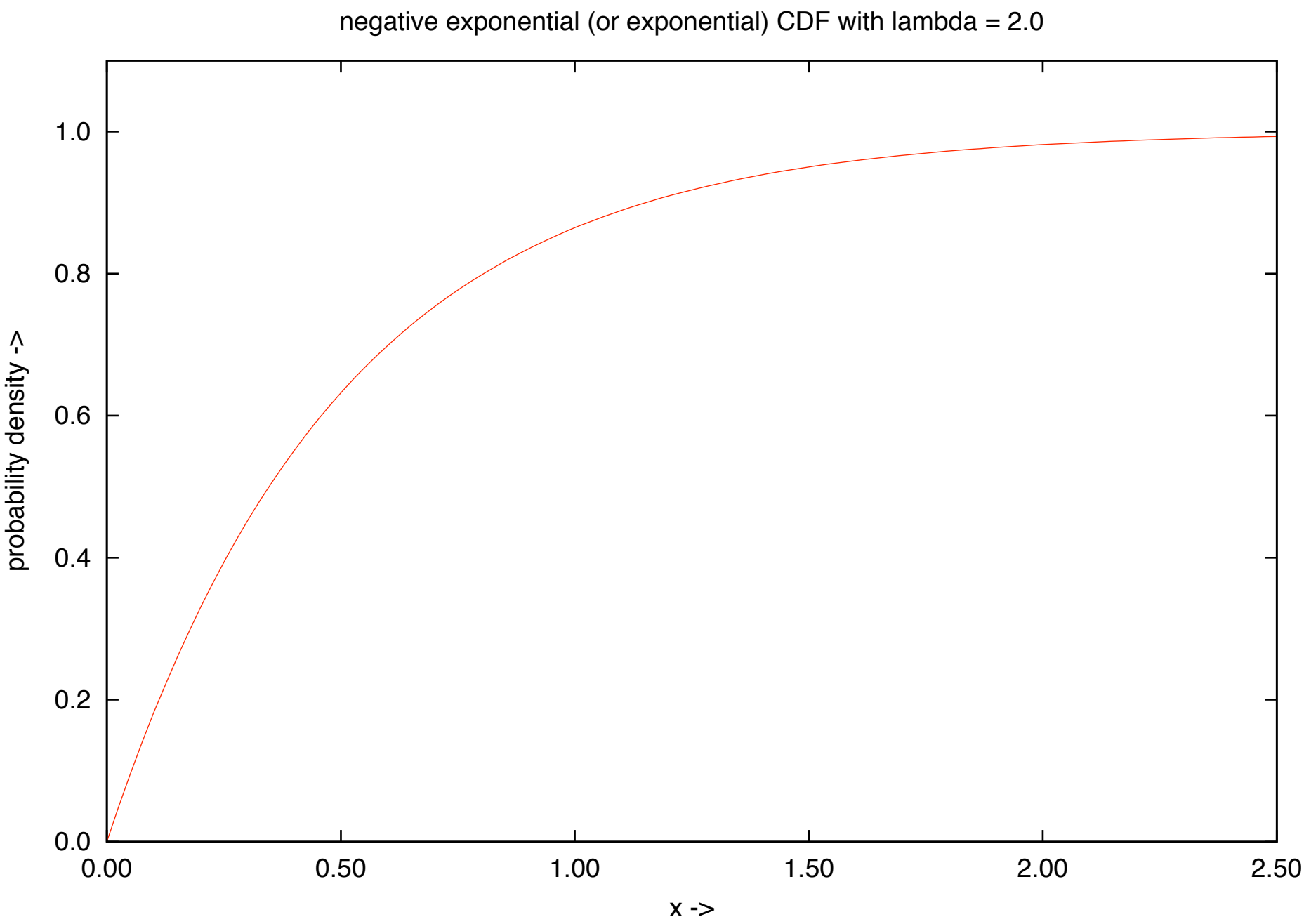

maxwell PDF
a = 1.5
a = 1.0
a = 0.5
probability density ->
x ->
0.0
0.5
1.0
1.5
2.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0

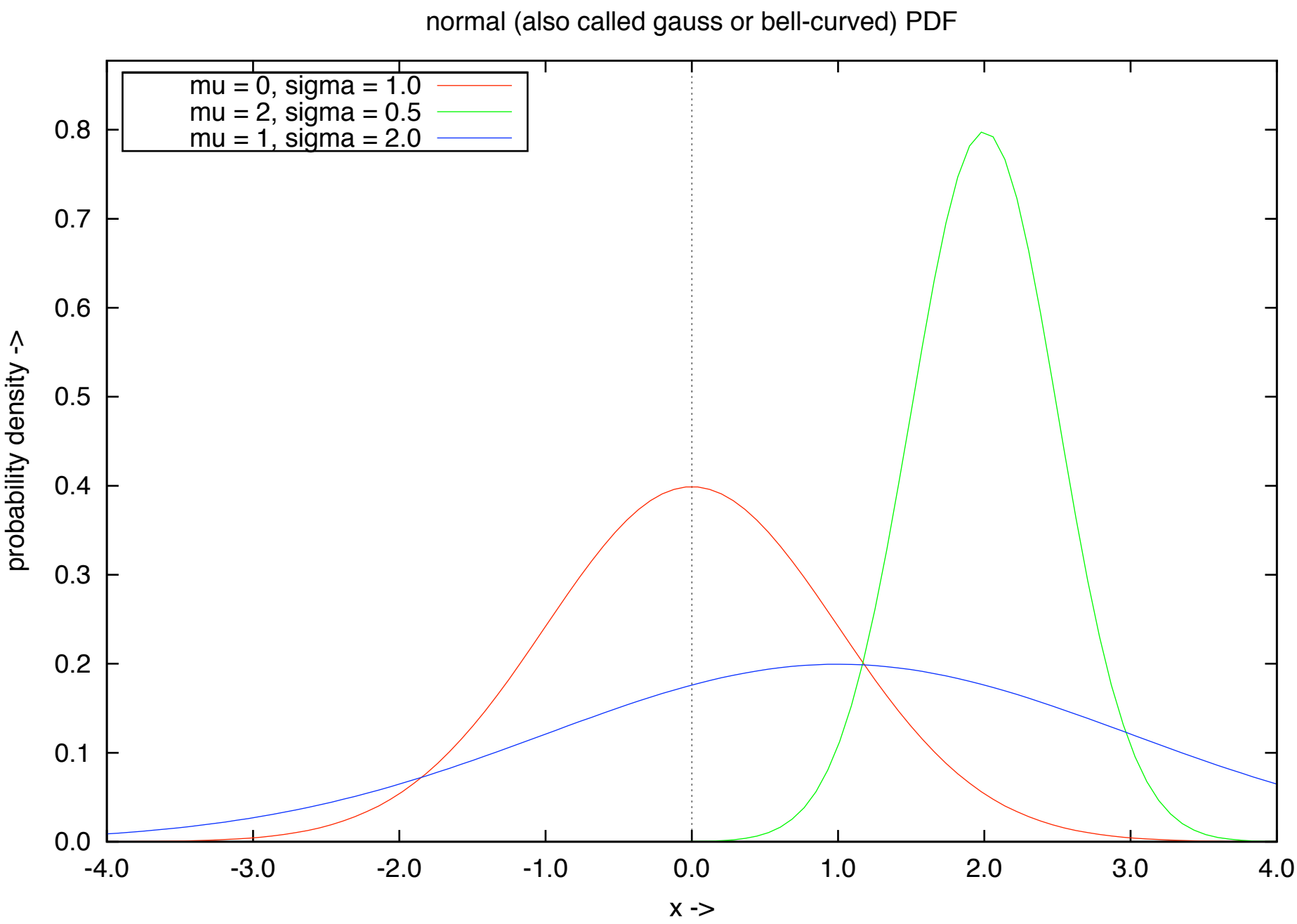

maxwell CDF
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
a = 1.5
a = 1.0
a = 0.5

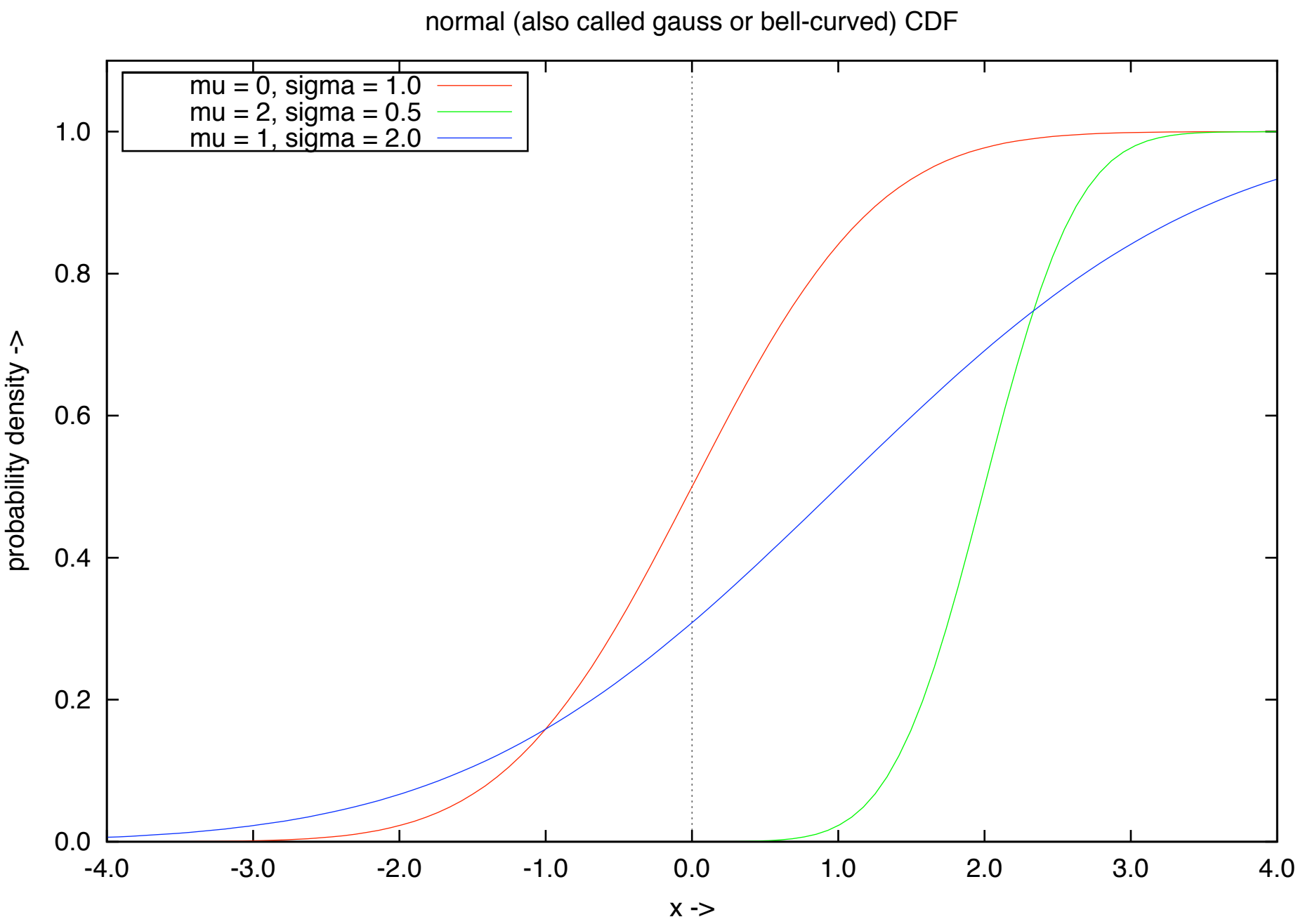

negative binomial (or pascal or polya) PDF with r = 8, p = 0.4
probability density ->
k ->
0.00
0.01
0.02
0.03
0.03
0.04
0.05
0.06
0.07
0.08
0.08
0
5
10
15
20
25
30

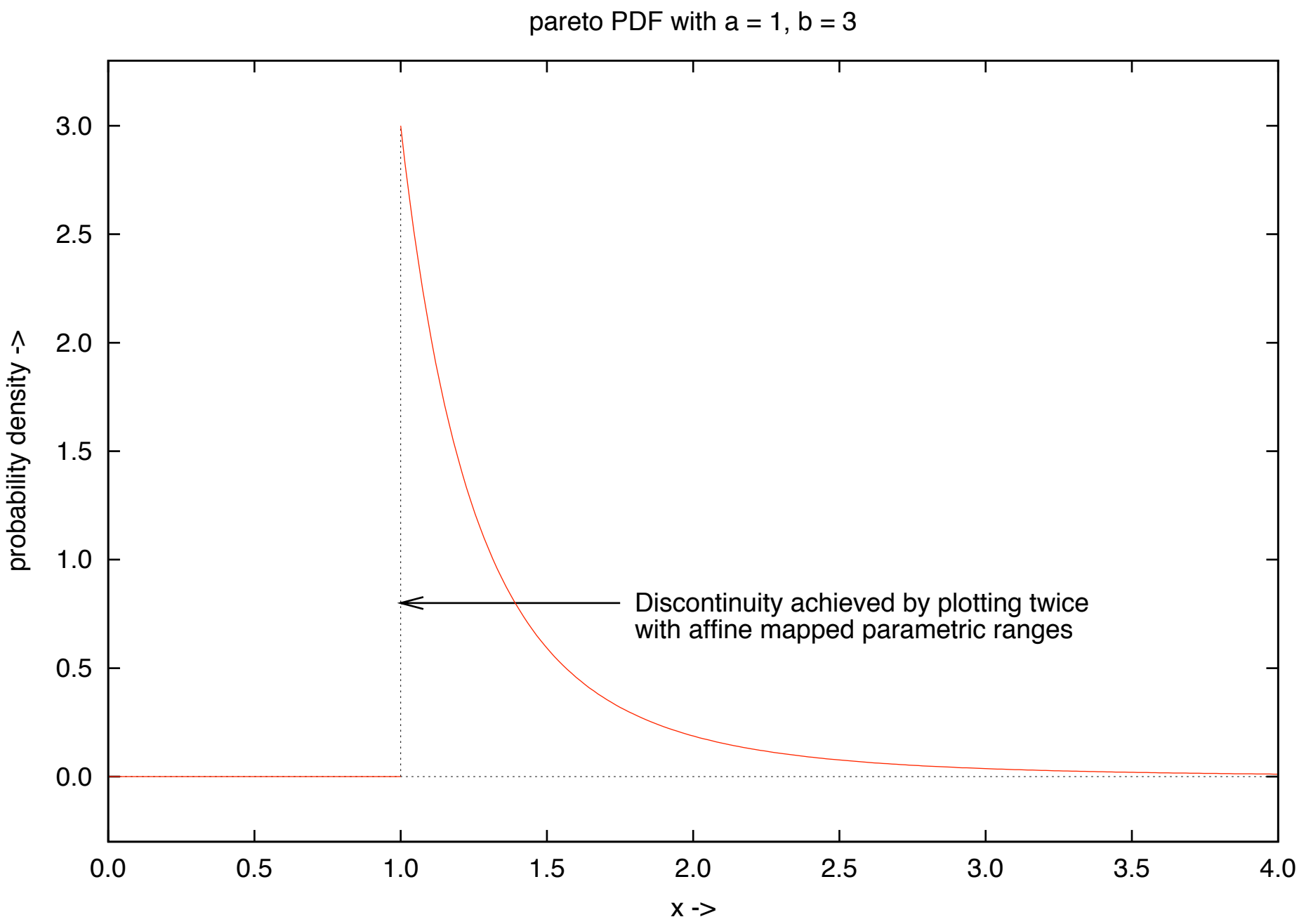

negative binomial (or pascal or polya) CDF with r = 8, p = 0.4
probability density ->
k ->
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
0
5
10
15
20
25
30

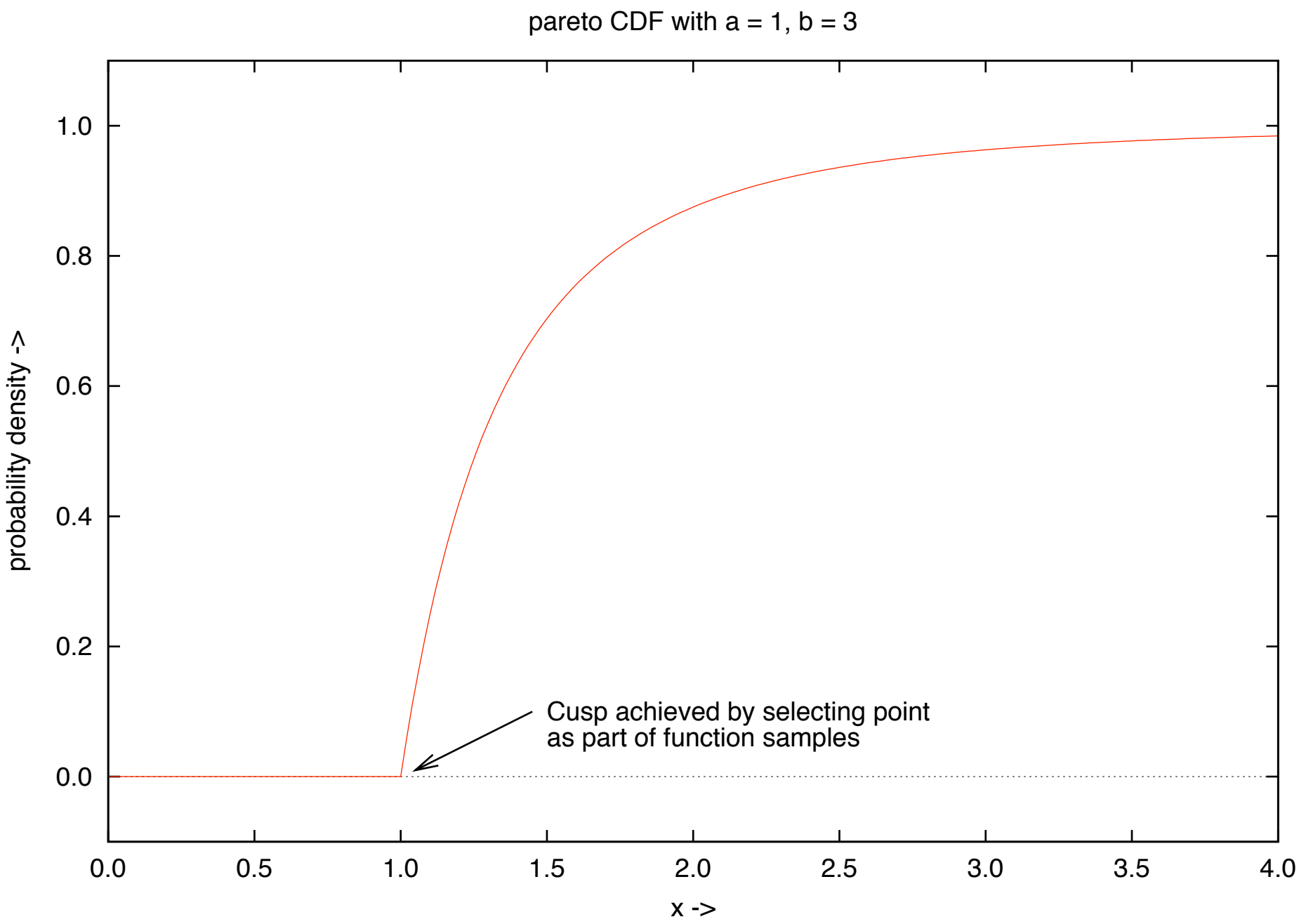

negative exponential (or exponential) PDF with lambda = 2.0
probability density ->
x ->
2.0
1.5
1.0
0.5
0.0
0.00
0.50
1.00
1.50
2.00
2.50

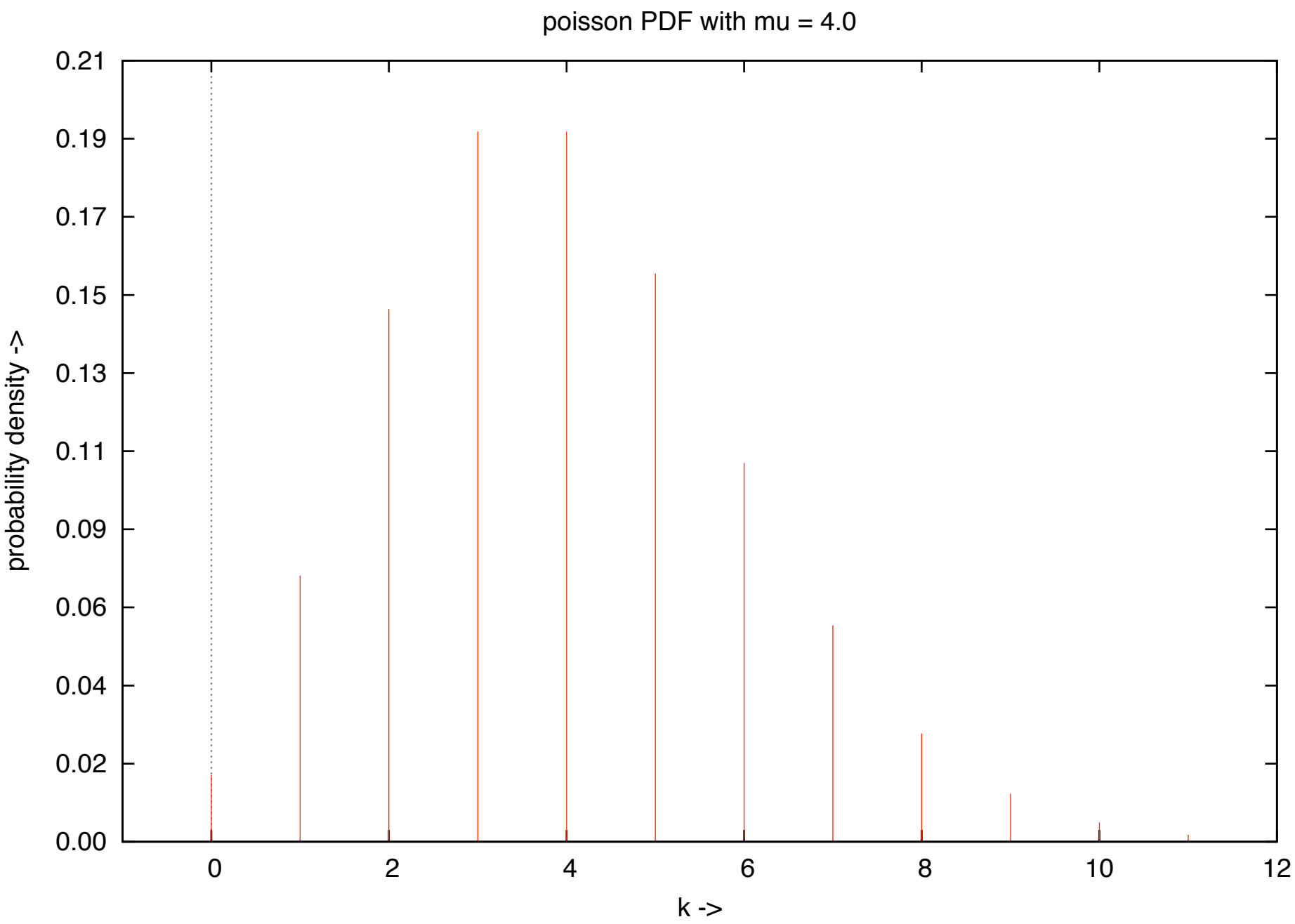

negative exponential (or exponential) CDF with lambda = 2.0
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.00
0.50
1.00
1.50
2.00
2.50

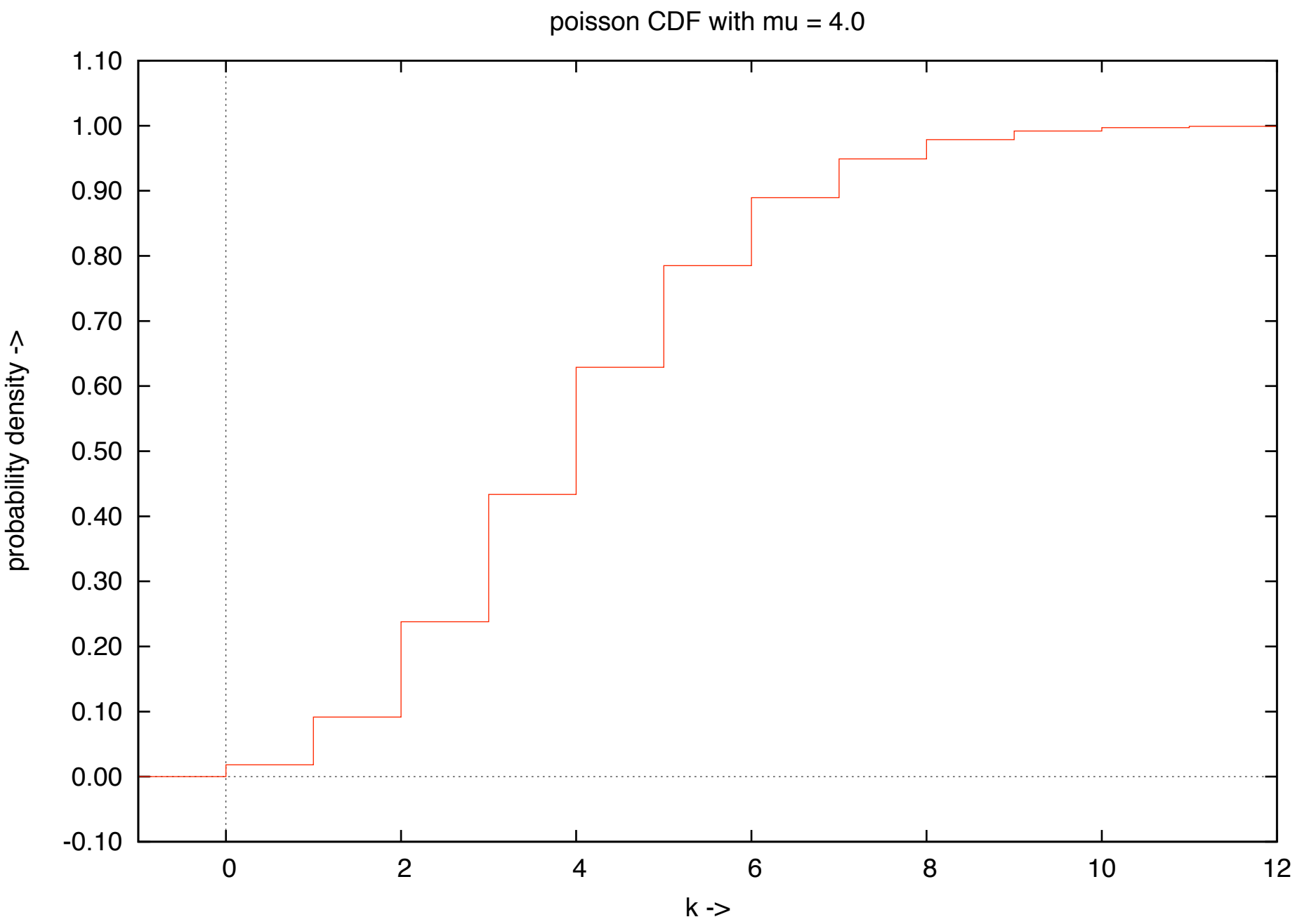

normal (also called gauss or bell-curved) PDF
mu = 0, sigma = 1.0
mu = 2, sigma = 0.5
mu = 1, sigma = 2.0
probability density ->
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
-4.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0
x ->

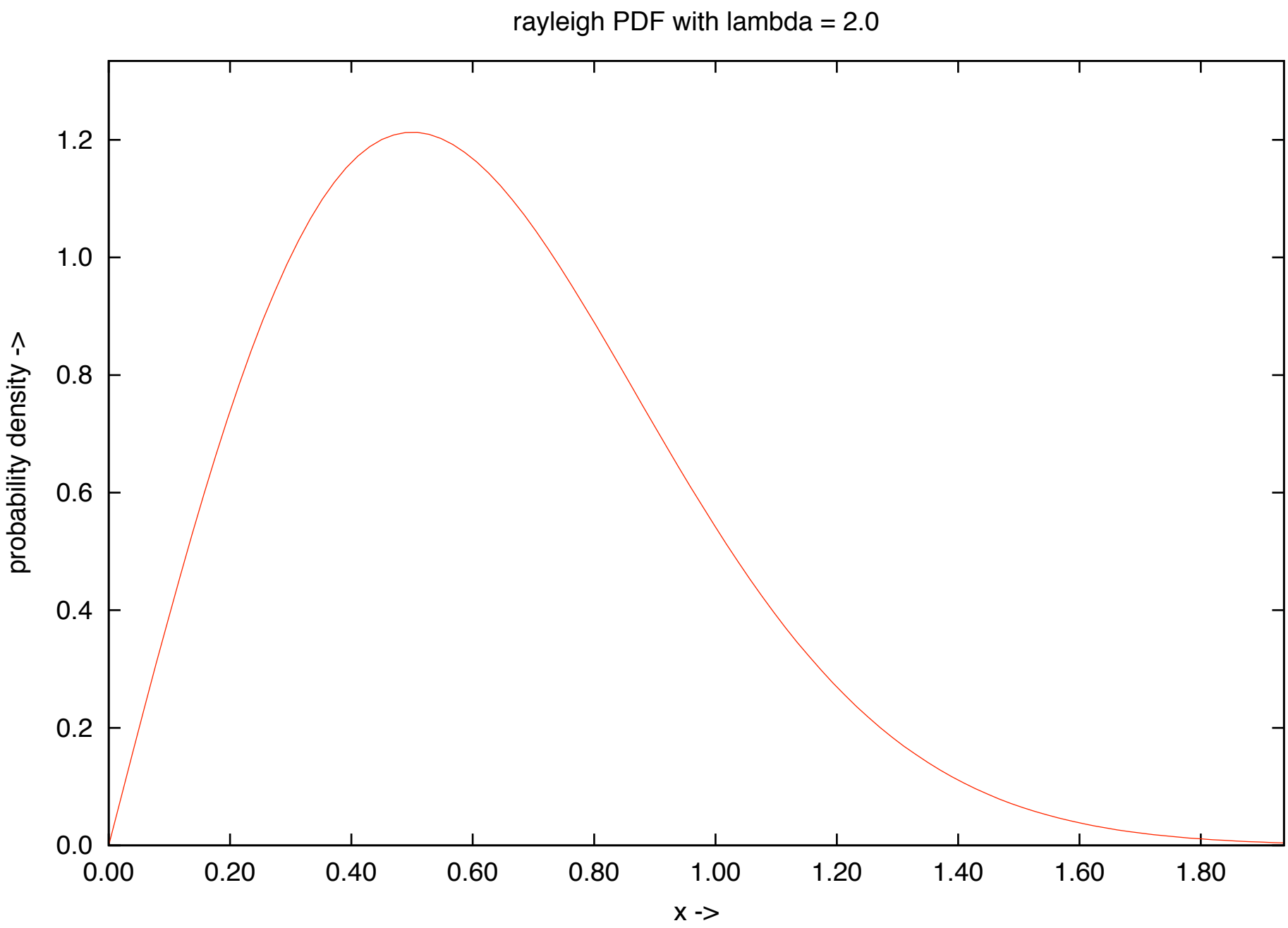

normal (also called gauss or bell-curved) CDF
mu = 0, sigma = 1.0
mu = 2, sigma = 0.5
mu = 1, sigma = 2.0
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
-4.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0

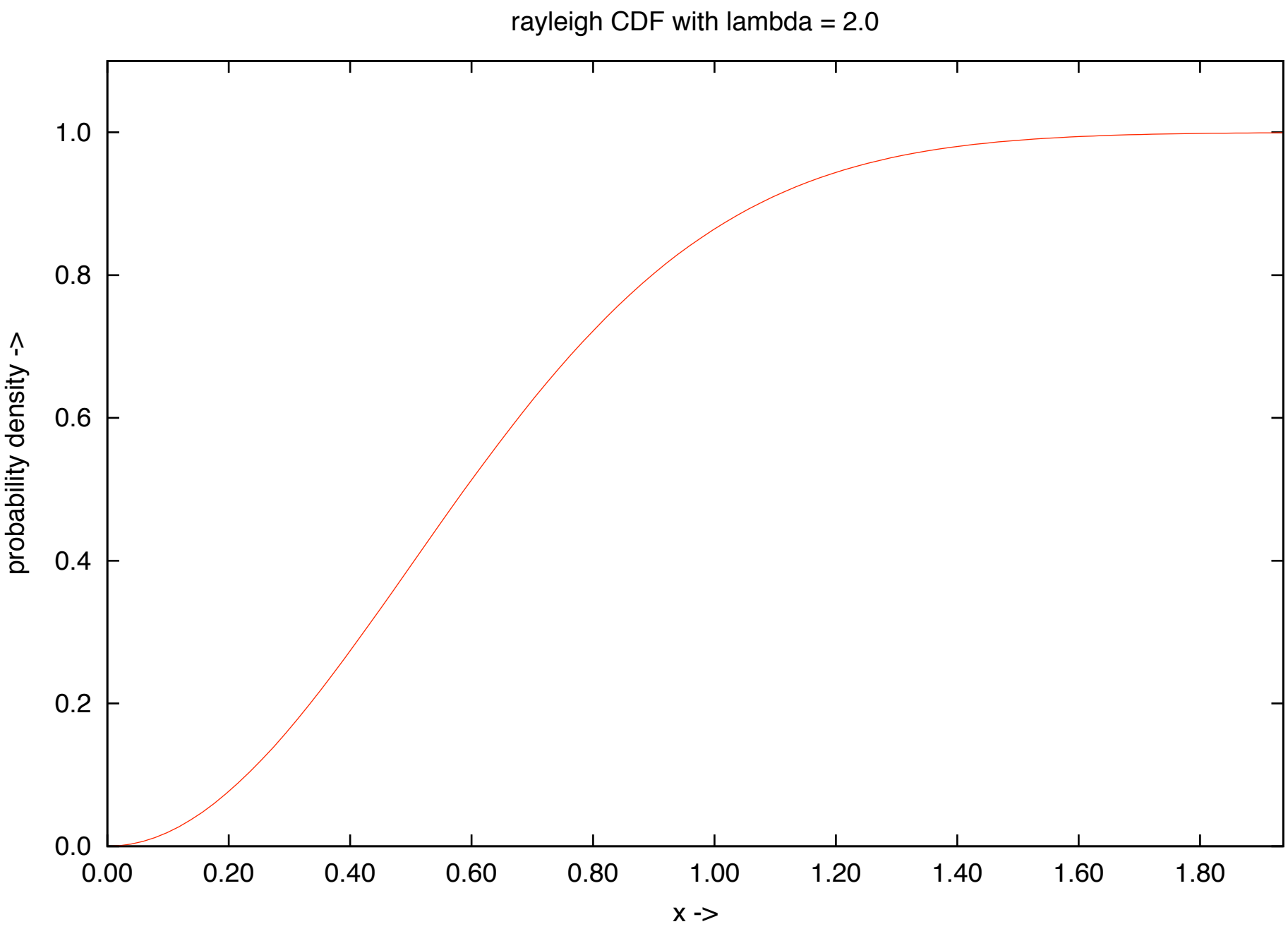

pareto PDF with a = 1, b = 3
probability density ->
x ->
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Discontinuity achieved by plotting twice
with affine mapped parametric ranges

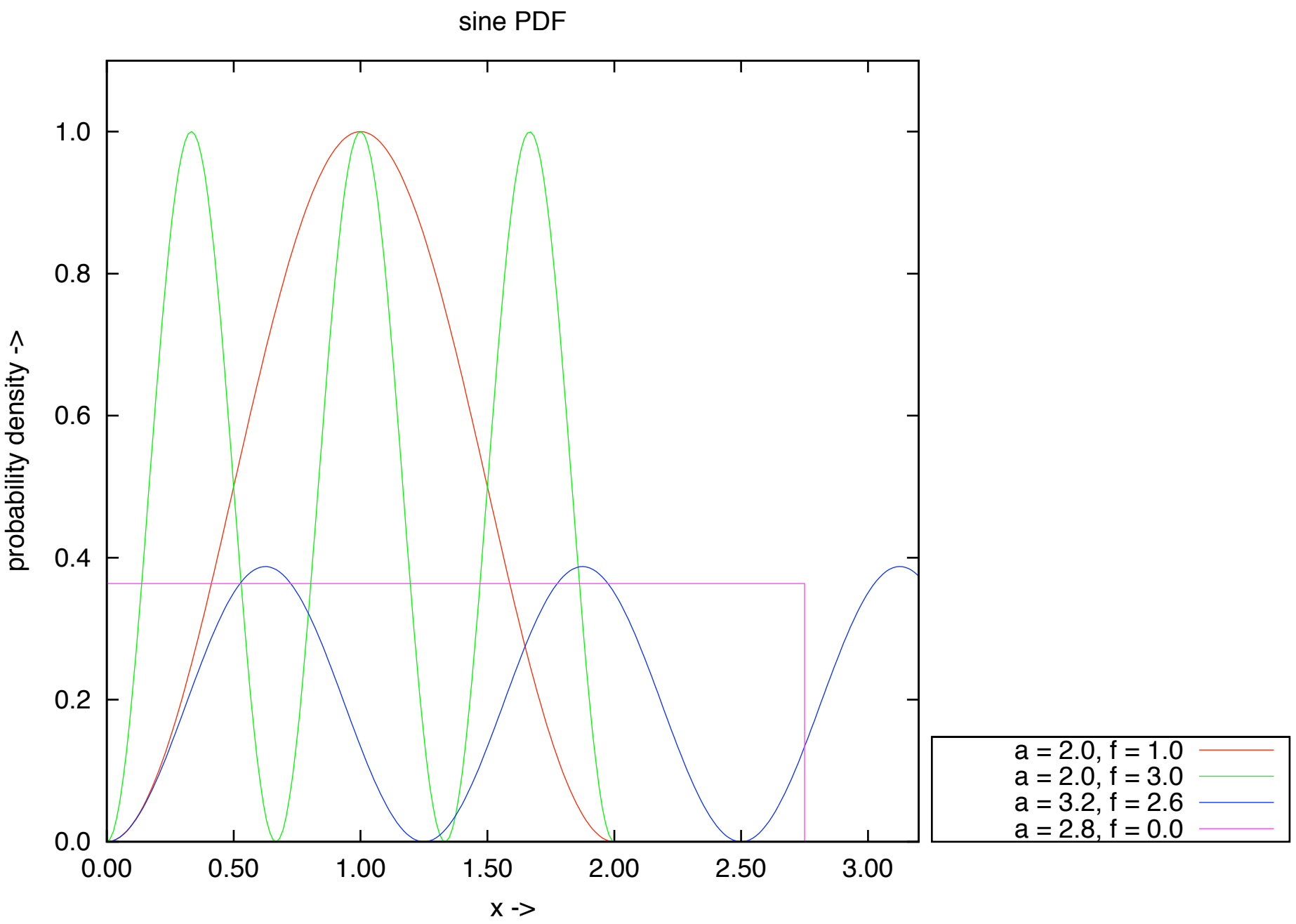

pareto CDF with a = 1, b = 3
probability density ->
x ->
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Cusp achieved by selecting point
as part of function samples

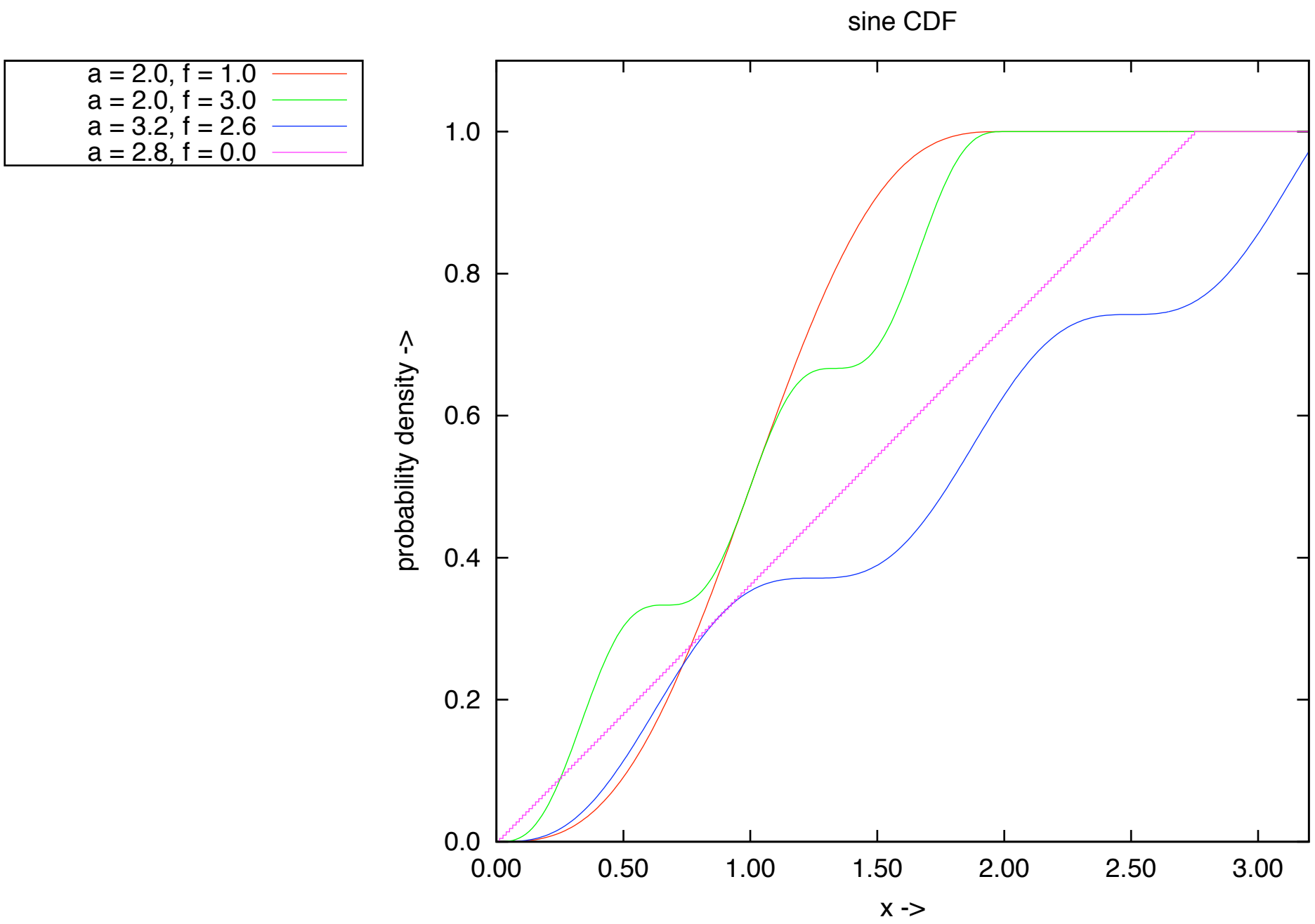

poisson PDF with mu = 4.0
probability density ->
k ->
0.21
0.19
0.17
0.15
0.13
0.11
0.09
0.06
0.04
0.02
0.00
0
2
4
6
8
10
12

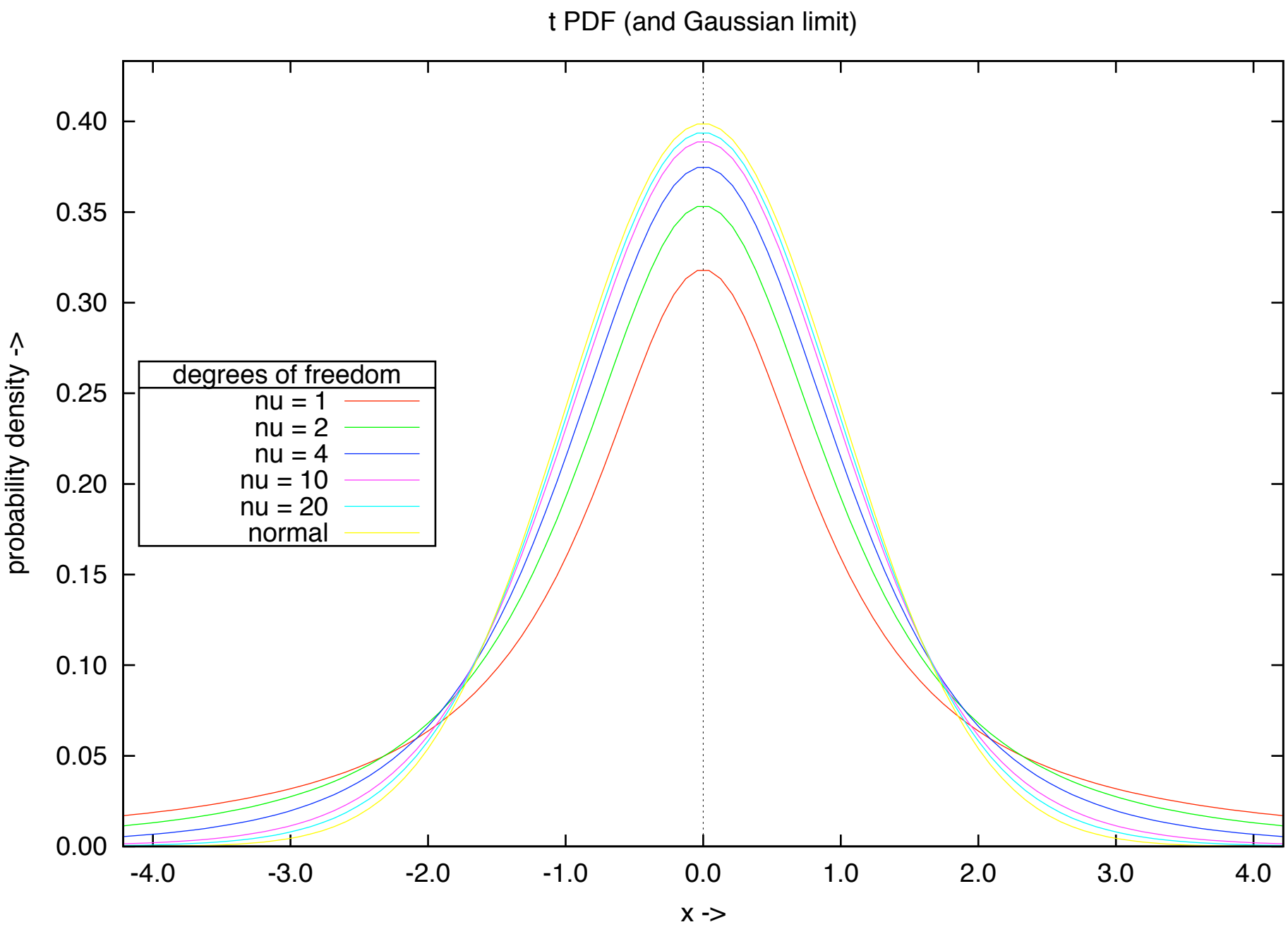

poisson CDF with mu = 4.0
probability density ->
k ->
1.10
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
-0.10
0
2
4
6
8
10
12

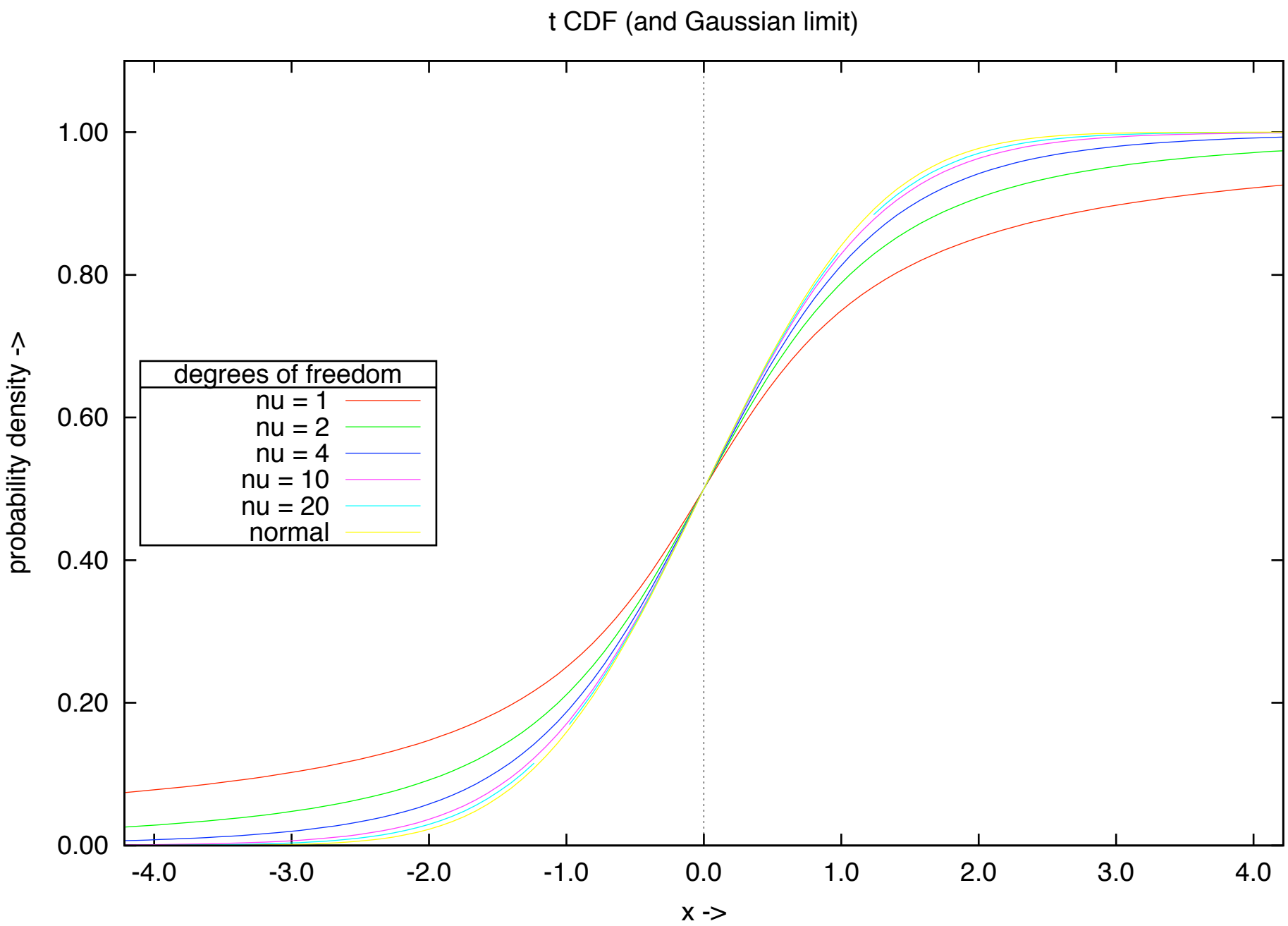

rayleigh PDF with lambda = 2.0
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0.00
0.20
0.40
0.60
0.80
1.00
1.20
1.40
1.60
1.80

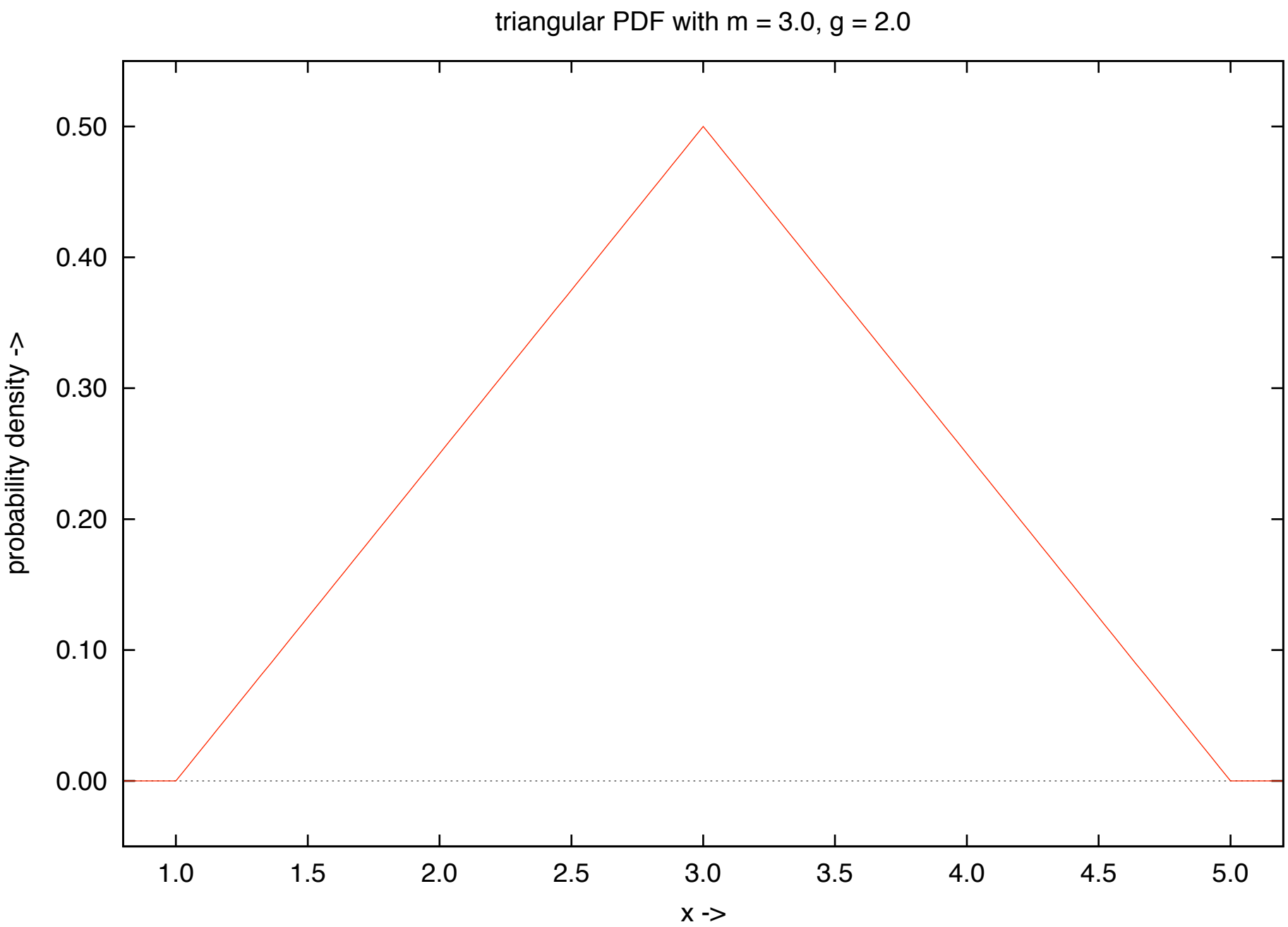

rayleigh CDF with lambda = 2.0
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.00
0.20
0.40
0.60
0.80
1.00
1.20
1.40
1.60
1.80

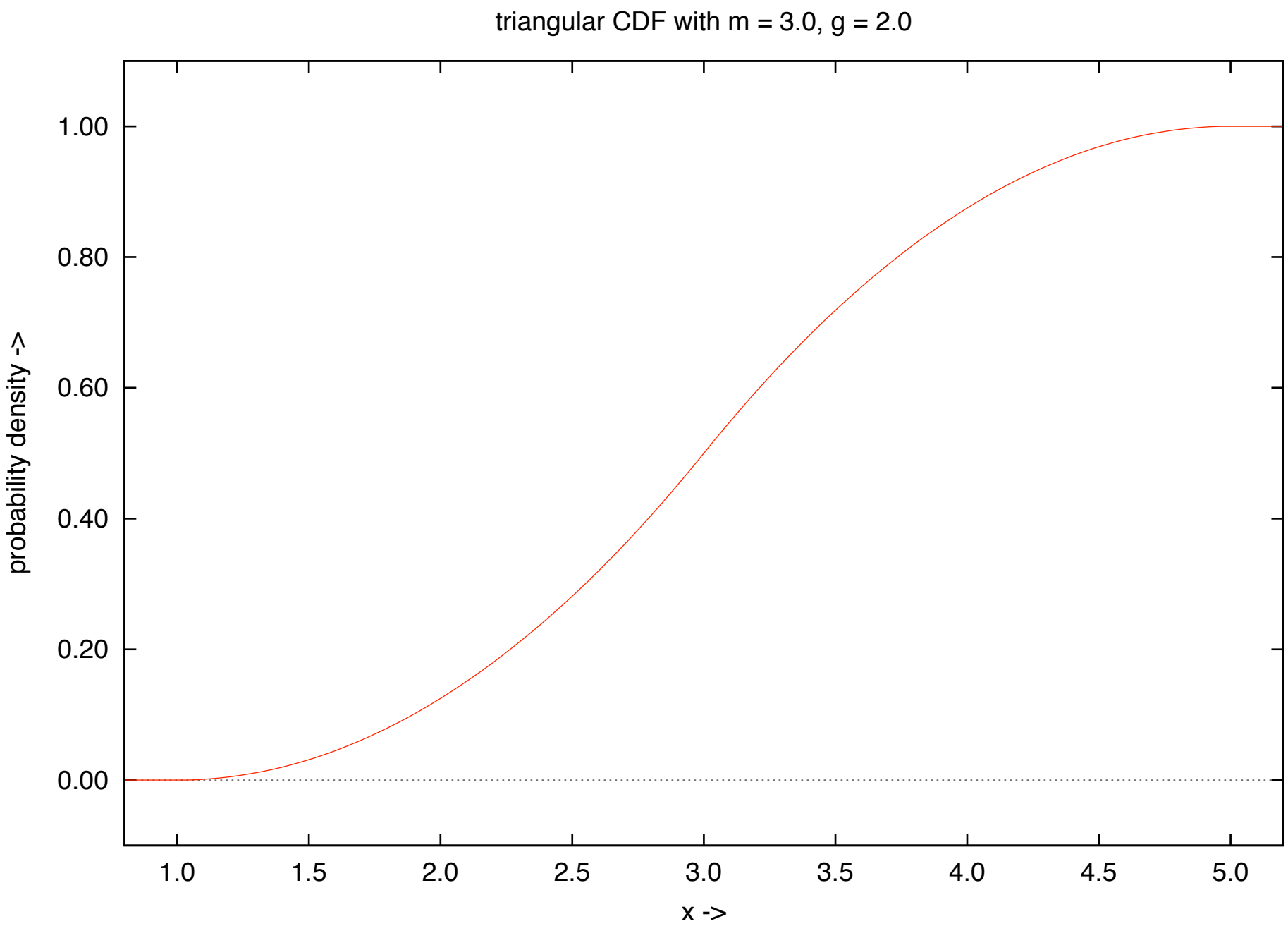

sine PDF
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.00
0.50
1.00
1.50
2.00
2.50
3.00
a = 2.0, f = 1.0
a = 2.0, f = 3.0
a = 3.2, f = 2.6
a = 2.8, f = 0.0

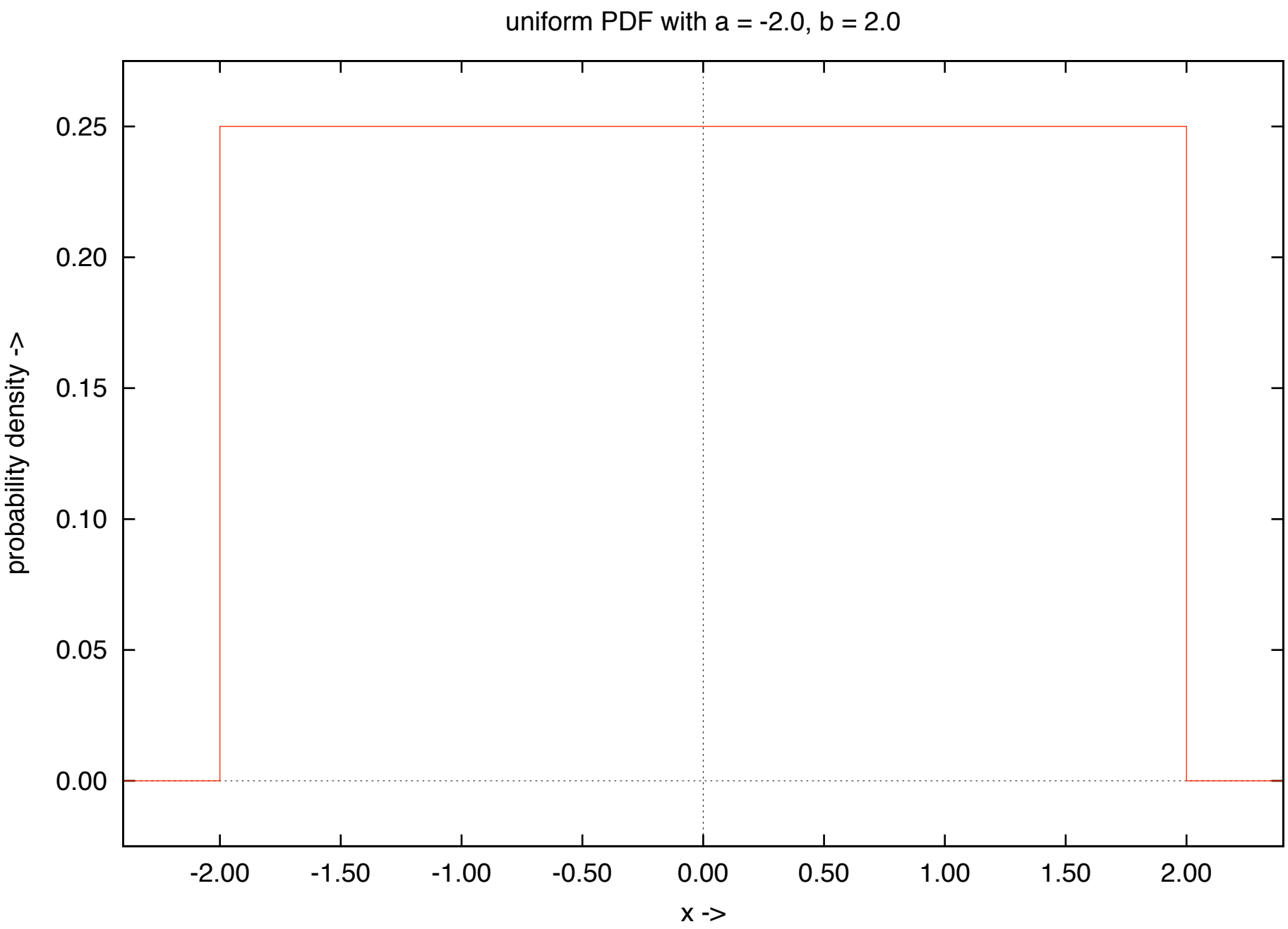

sine CDF
a = 2.0, f = 1.0
a = 2.0, f = 3.0
a = 3.2, f = 2.6
a = 2.8, f = 0.0
probability density ->
x ->
0.0
0.2
0.4
0.6
0.8
1.0
0.00
0.50
1.00
1.50
2.00
2.50
3.00

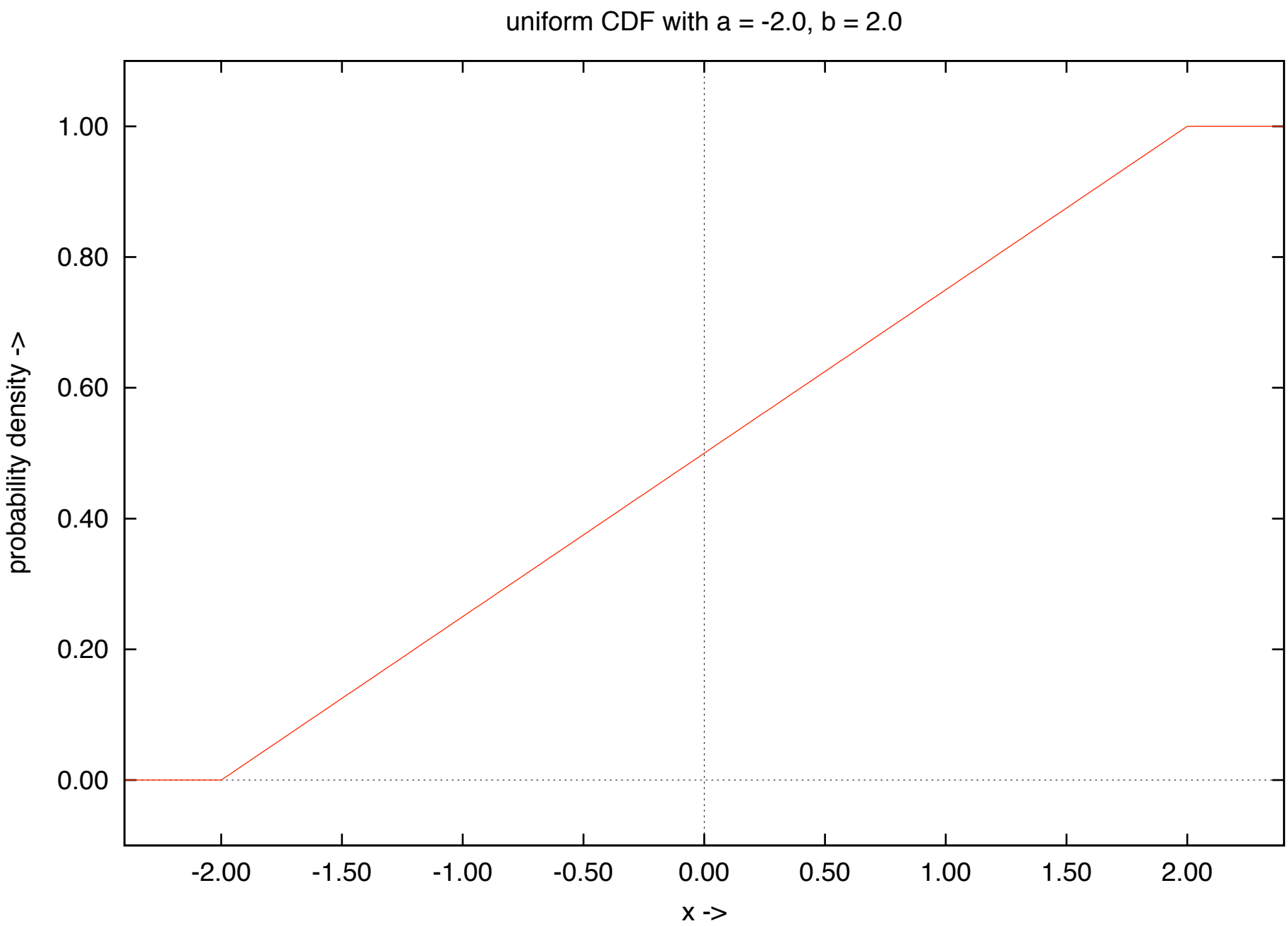

t PDF (and Gaussian limit)
probability density ->
x ->
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
-4.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0
degrees of freedom
nu = 1
nu = 2
nu = 4
nu = 10
nu = 20
normal

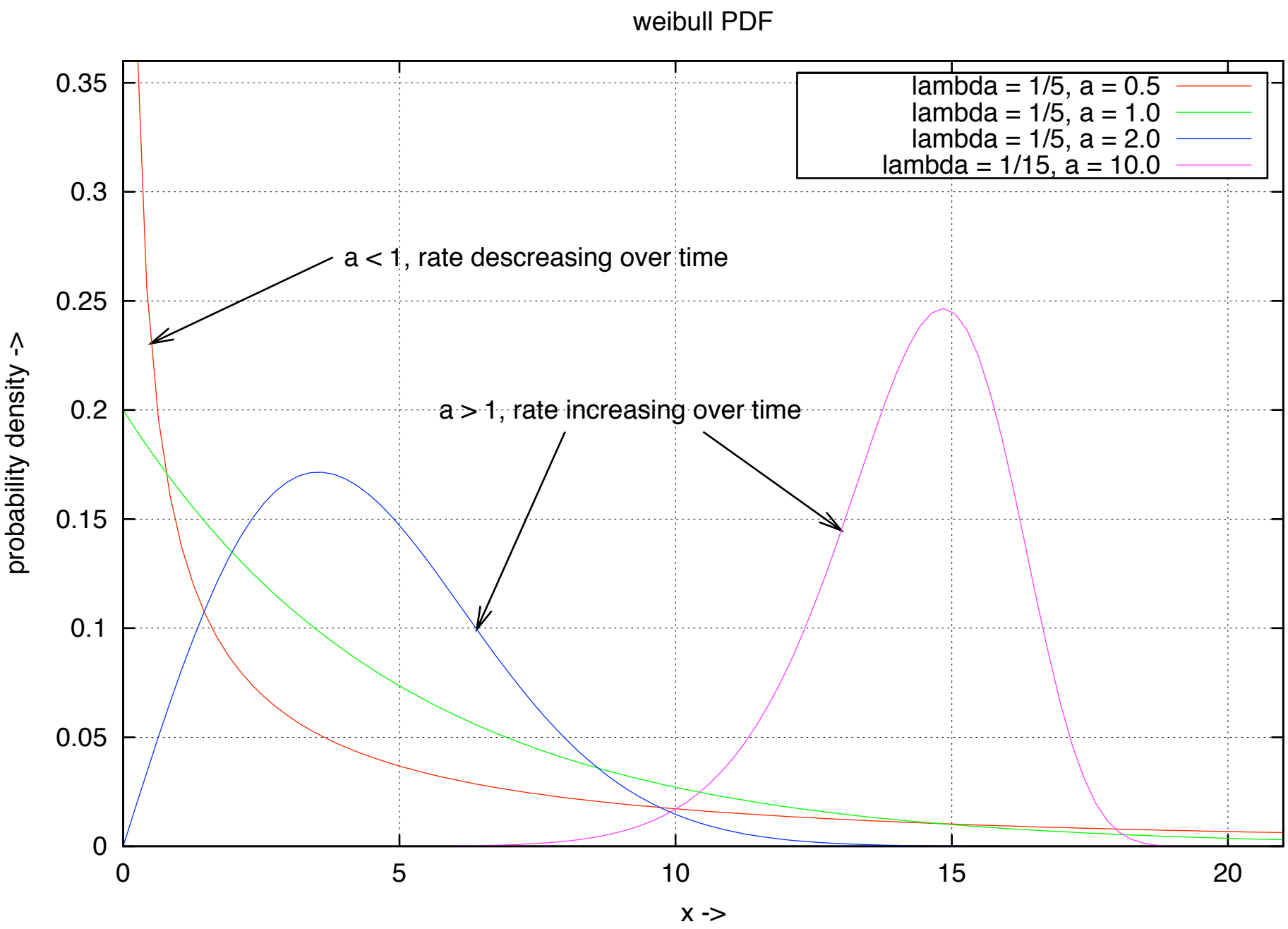

t CDF (and Gaussian limit)
probability density ->
x ->
1.00
0.80
0.60
0.40
0.20
0.00
-4.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0
degrees of freedom
nu = 1
nu = 2
nu = 4
nu = 10
nu = 20
normal

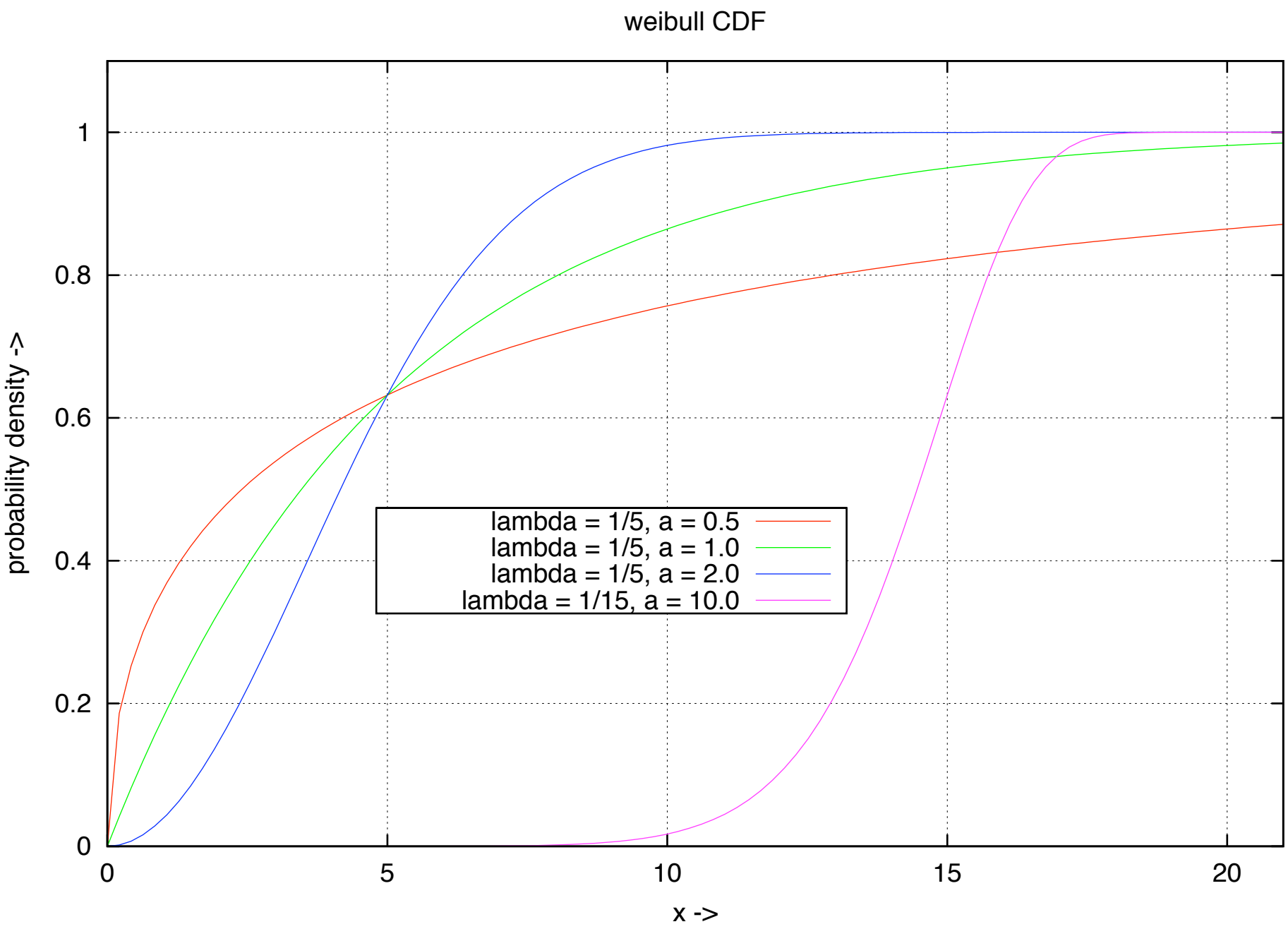

triangular PDF with m = 3.0, g = 2.0
probability density ->
0.50
0.40
0.30
0.20
0.10
0.00
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
x ->

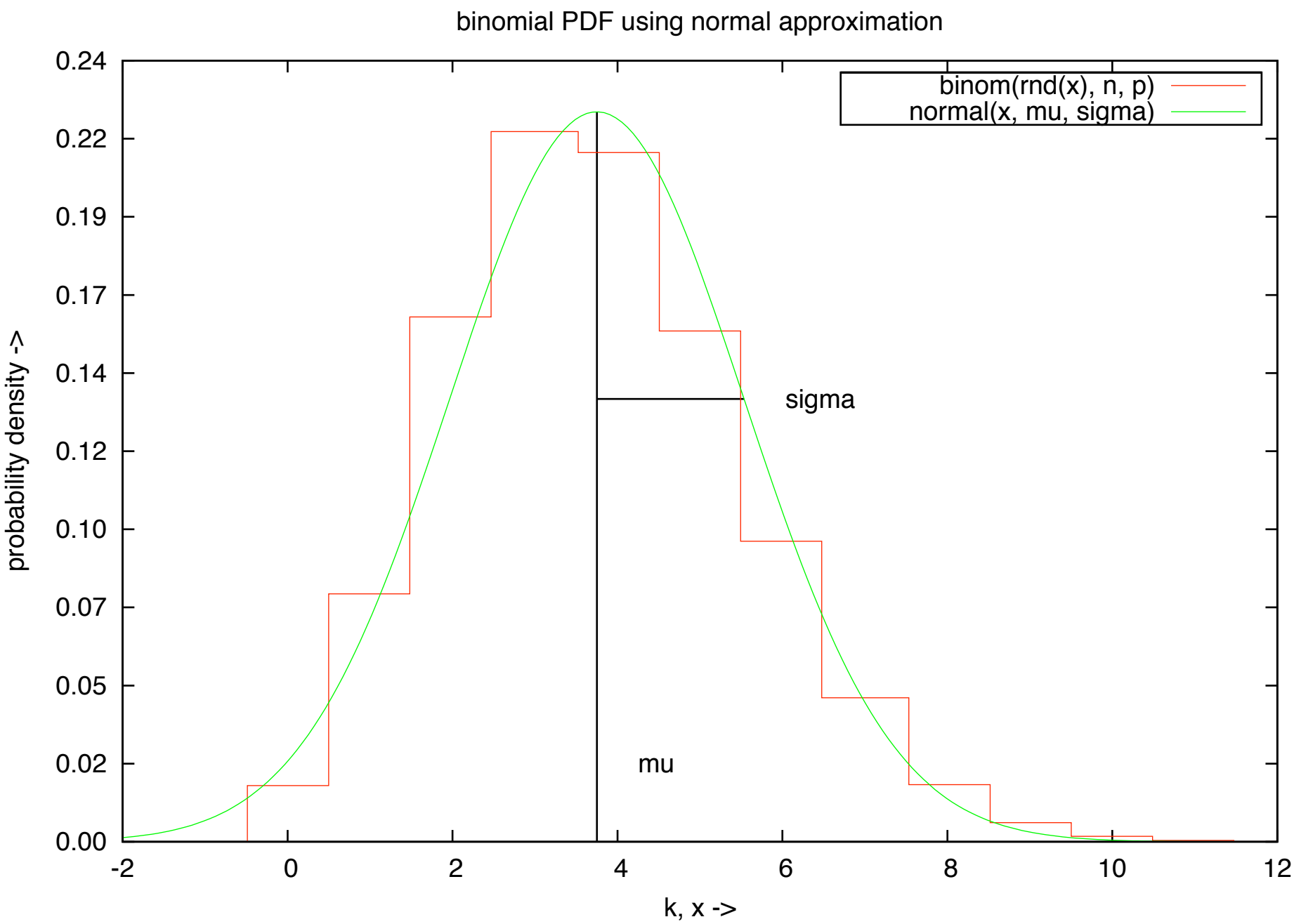

triangular CDF with m = 3.0, g = 2.0
probability density ->
x ->
1.00
0.80
0.60
0.40
0.20
0.00
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

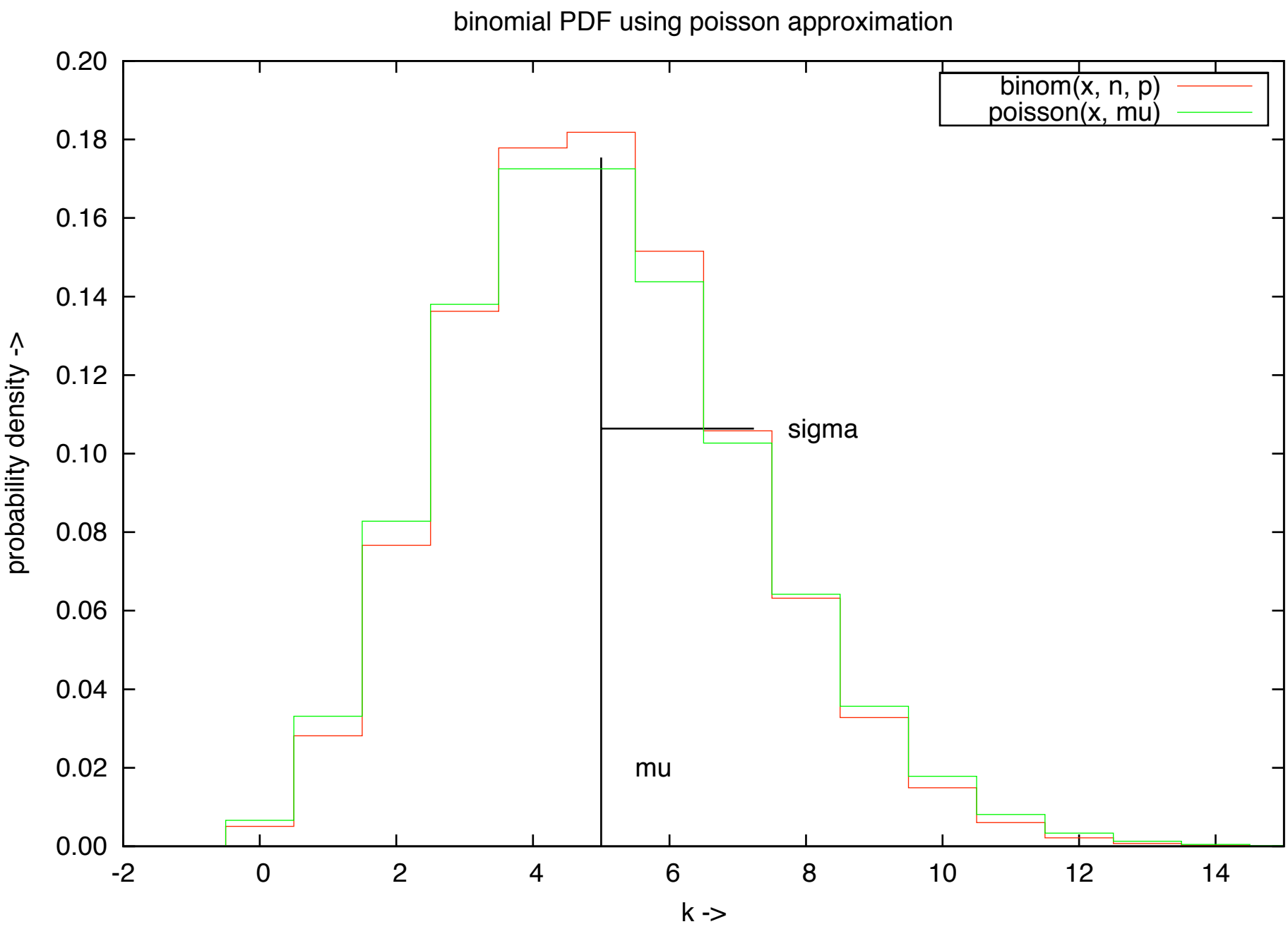

uniform PDF with a = -2.0, b = 2.0
probability density ->
0.25
0.20
0.15
0.10
0.05
0.00
-2.00
-1.50
-1.00
-0.50
0.00
0.50
1.00
1.50
2.00
x ->

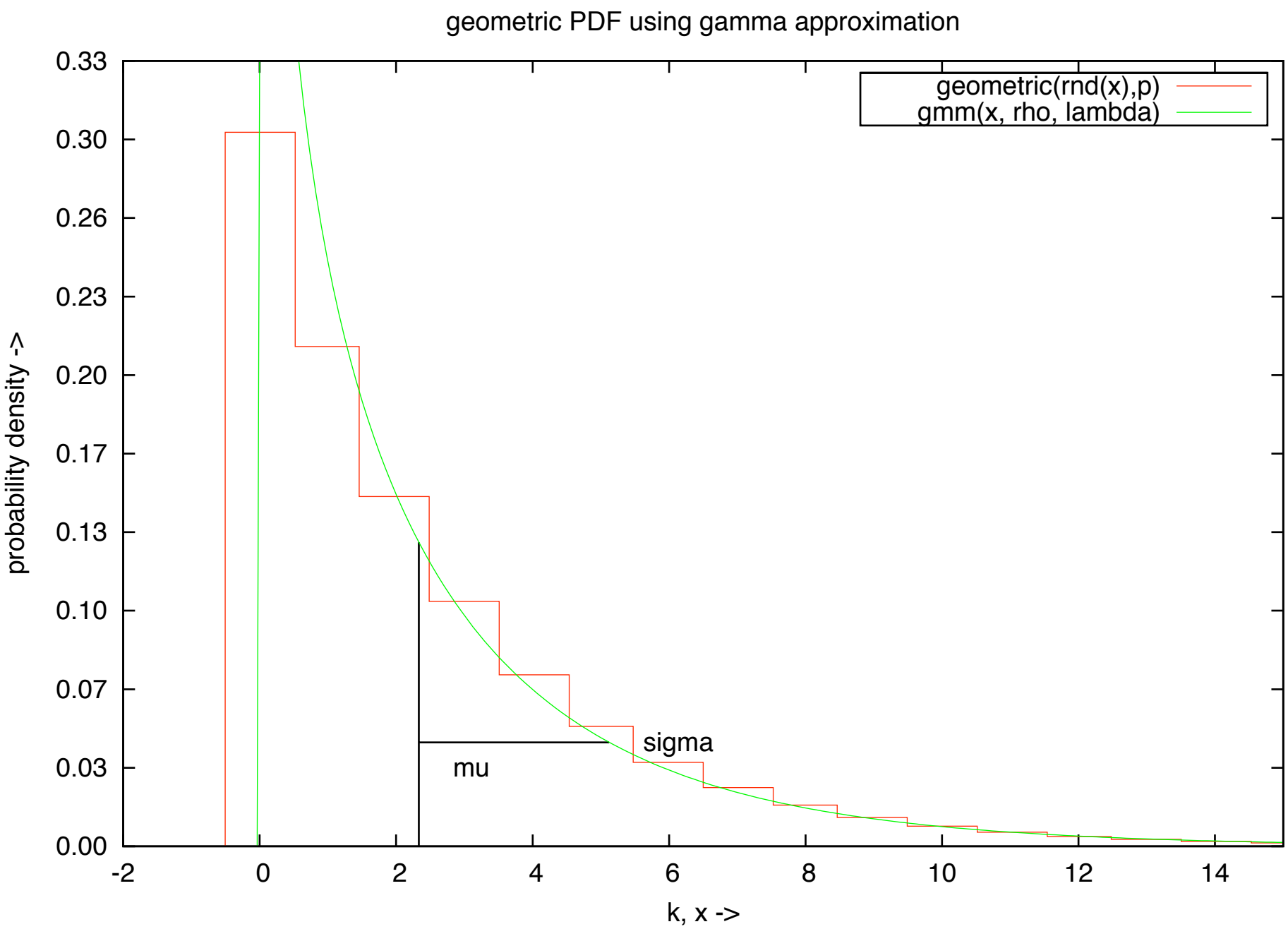

uniform CDF with a = -2.0, b = 2.0
probability density ->
1.00
0.80
0.60
0.40
0.20
0.00
-2.00
-1.50
-1.00
-0.50
0.00
0.50
1.00
1.50
2.00
x ->

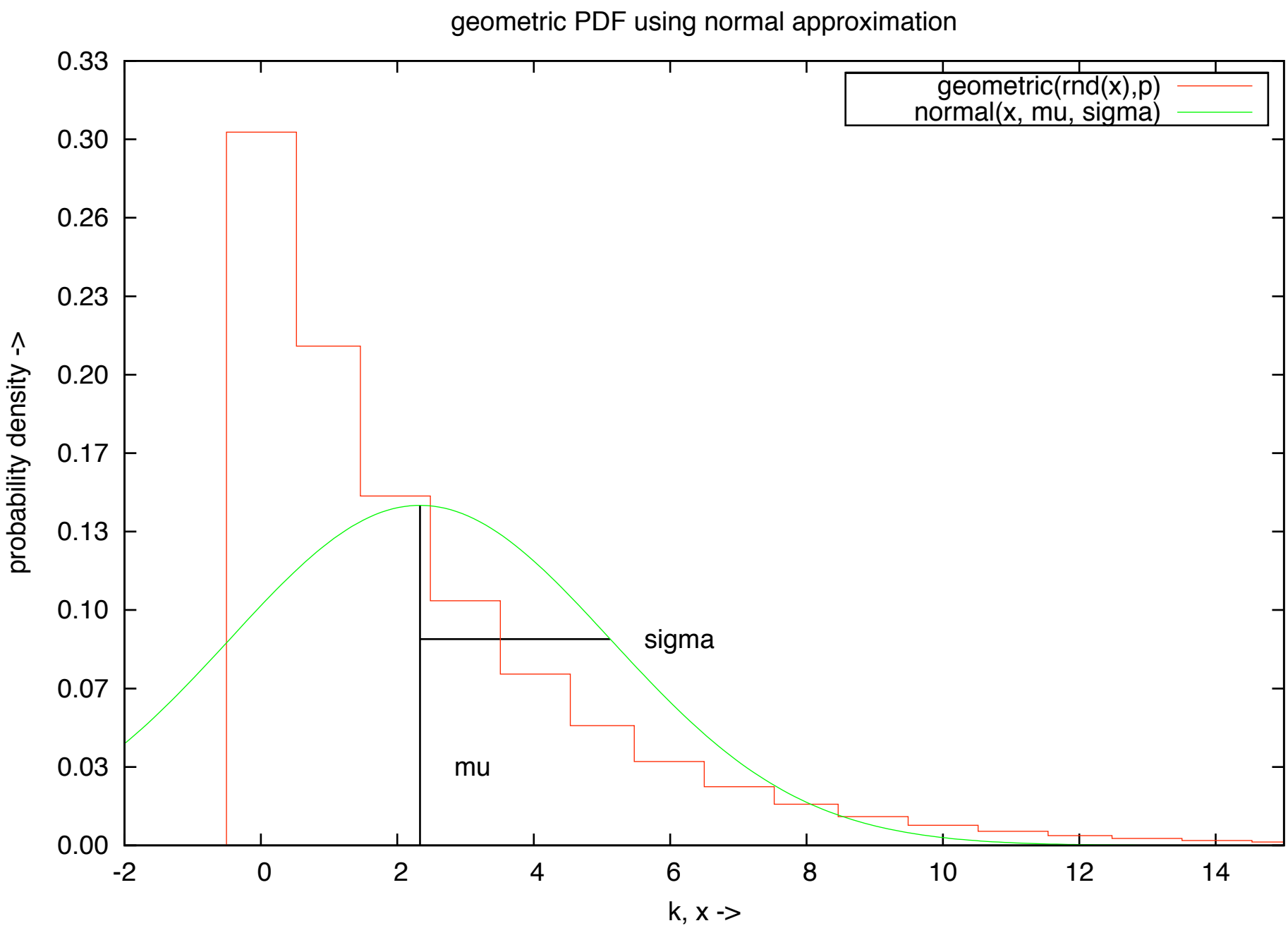

weibull PDF
lambda = 1/5, a = 0.5
lambda = 1/5, a = 1.0
lambda = 1/5, a = 2.0
lambda = 1/15, a = 10.0
a < 1, rate descreasing over time
a > 1, rate increasing over time
probability density ->
x ->
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0
5
10
15
20

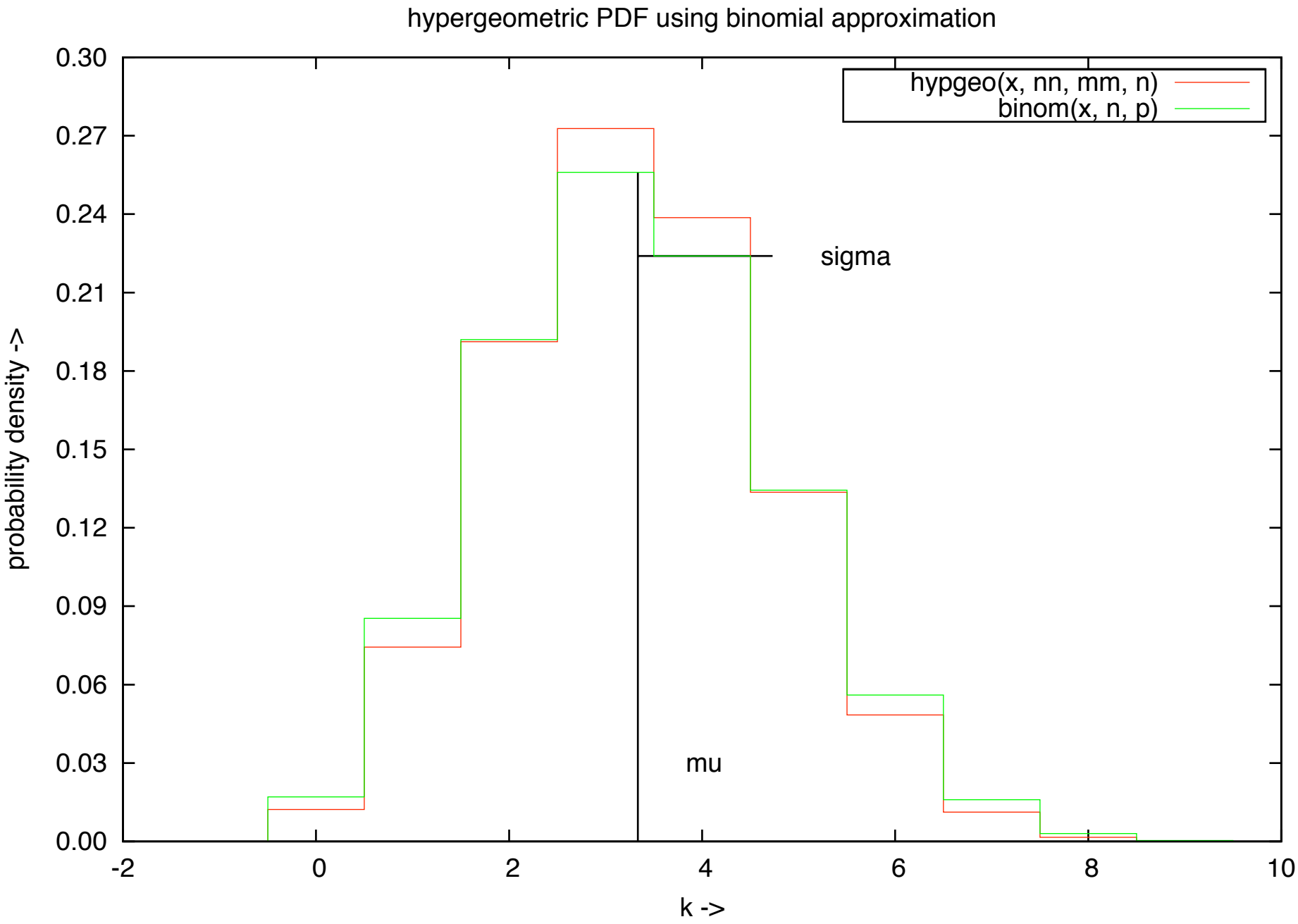

weibull CDF
probability density ->
x ->
0
0.2
0.4
0.6
0.8
1
0
5
10
15
20
lambda = 1/5, a = 0.5
lambda = 1/5, a = 1.0
lambda = 1/5, a = 2.0
lambda = 1/15, a = 10.0

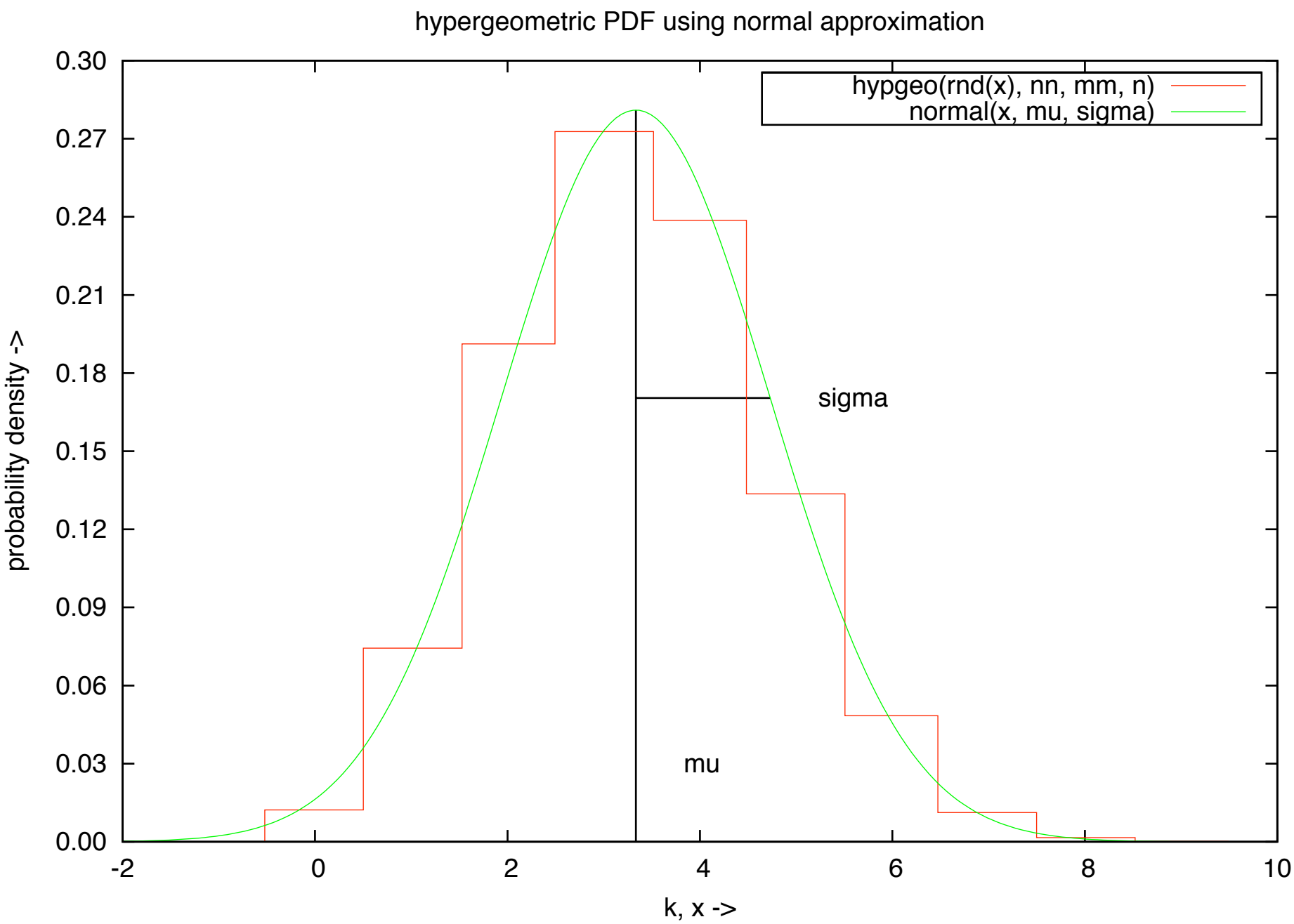

binomial PDF using normal approximation
binom(rnd(x), n, p)
normal(x, mu, sigma)
sigma
mu
probability density ->
k, x ->
0.24
0.22
0.19
0.17
0.14
0.12
0.10
0.07
0.05
0.02
0.00
-2
0
2
4
6
8
10
12

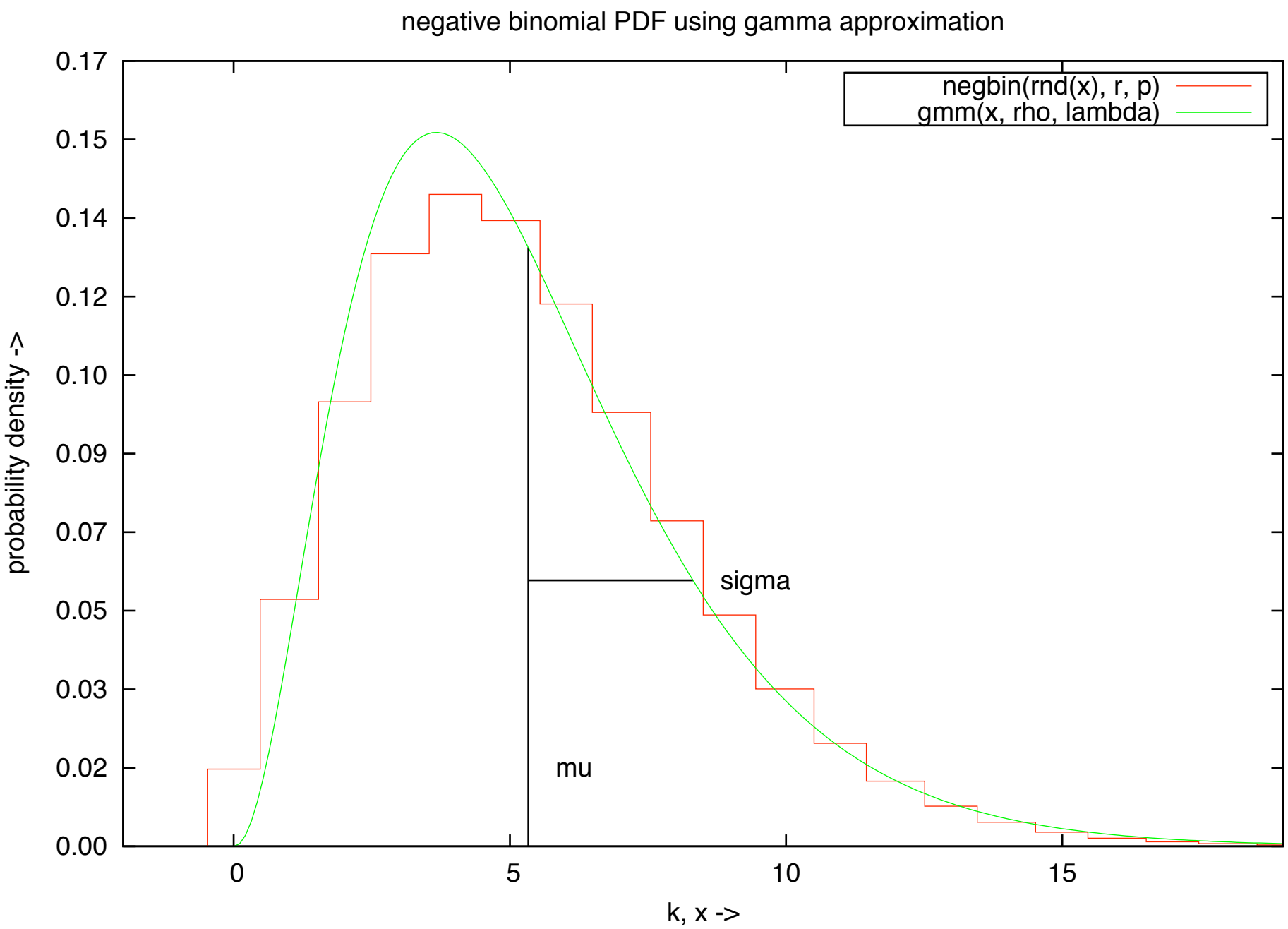

binomial PDF using poisson approximation
binom(x, n, p)
poisson(x, mu)
sigma
mu
probability density ->
k ->
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
-2
0
2
4
6
8
10
12
14

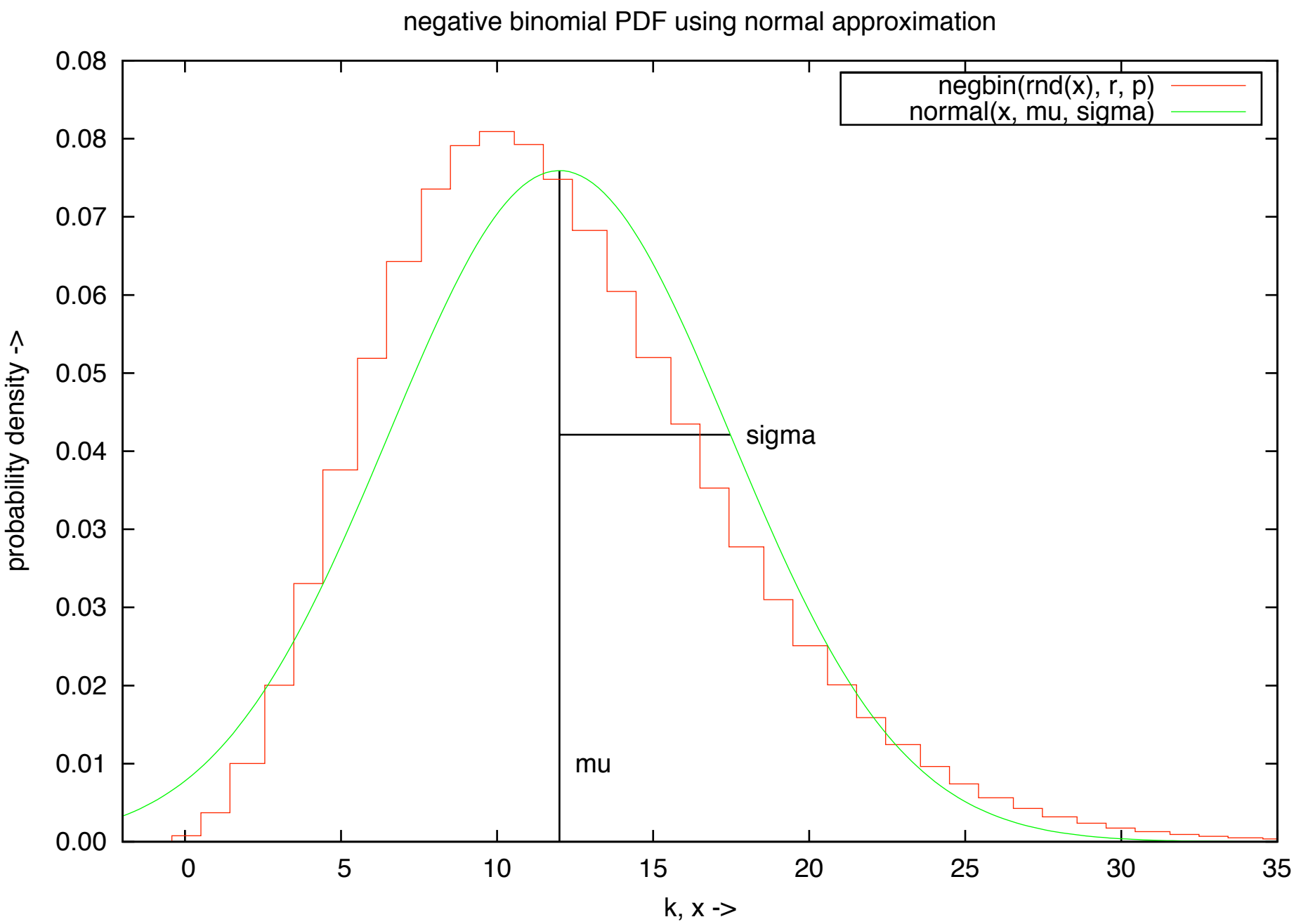

geometric PDF using gamma approximation
geometric(rnd(x),p)
gmm(x, rho, lambda)
probability density ->
k, x ->
0.33
0.30
0.26
0.23
0.20
0.17
0.13
0.10
0.07
0.03
0.00
-2
0
2
4
6
8
10
12
14
sigma
mu

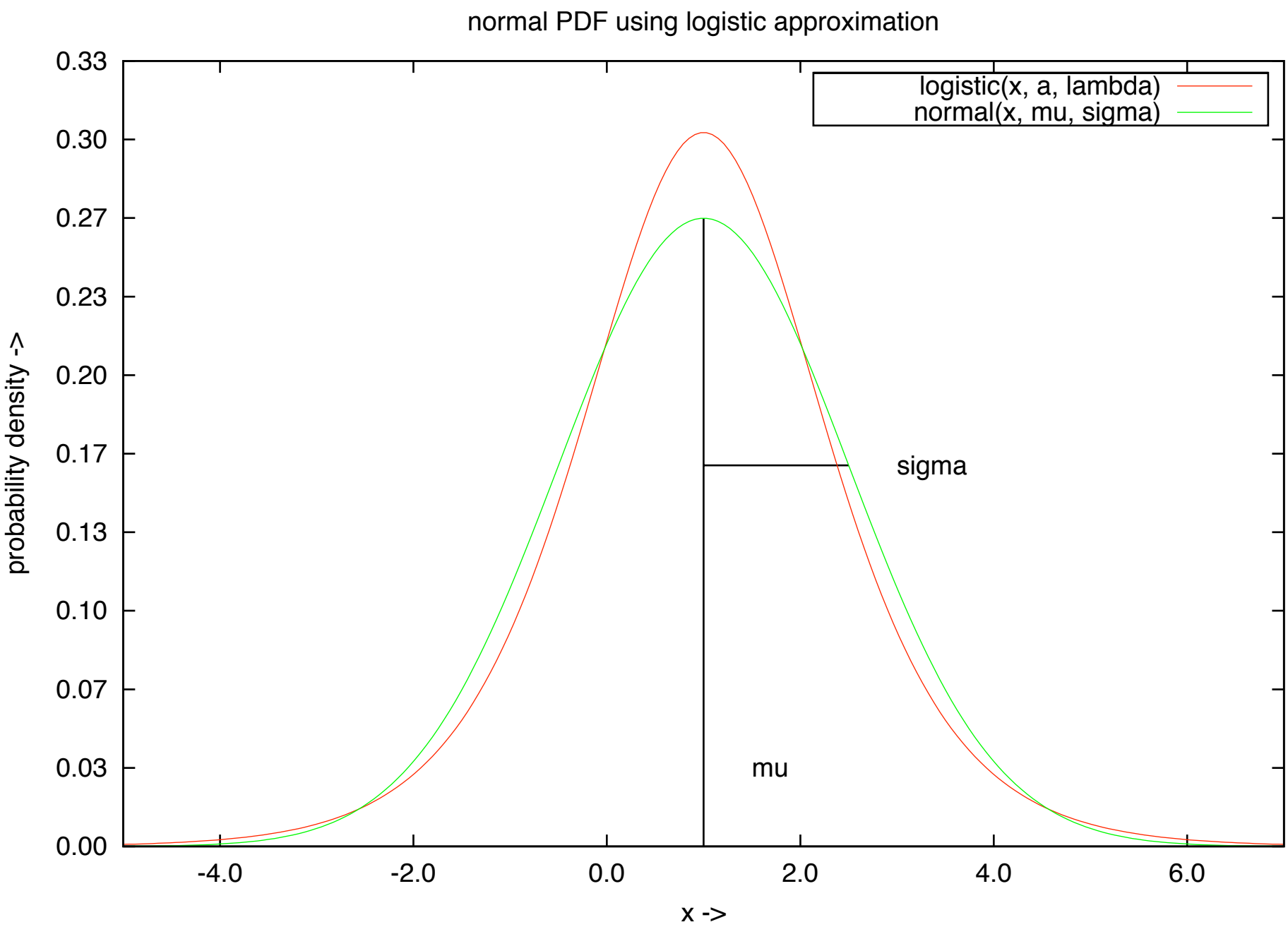

geometric PDF using normal approximation
geometric(rnd(x),p)
normal(x, mu, sigma)
probability density ->
k, x ->
0.33
0.30
0.26
0.23
0.20
0.17
0.13
0.10
0.07
0.03
0.00
-2
0
2
4
6
8
10
12
14
sigma
mu

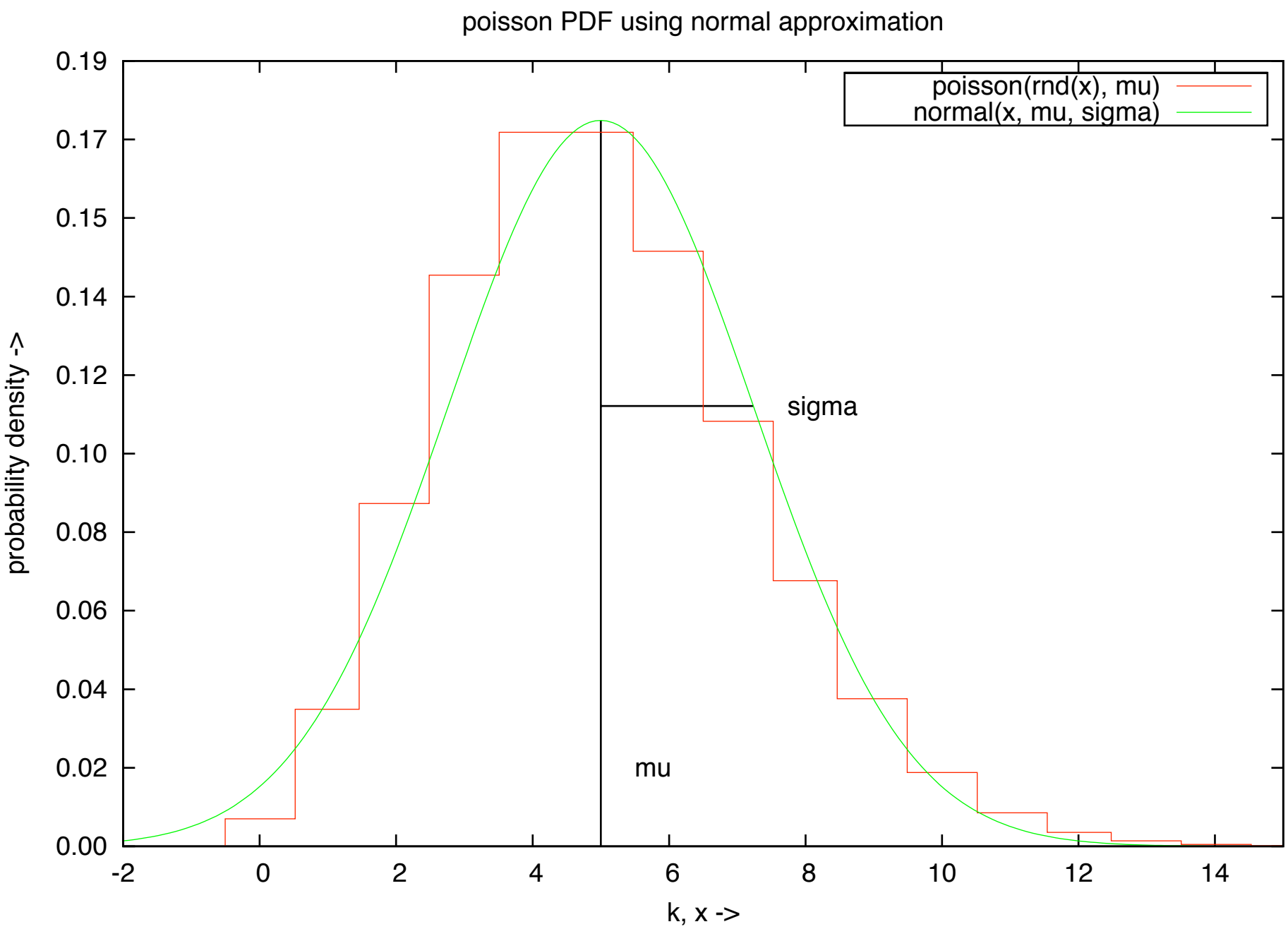

hypergeometric PDF using binomial approximation
hypgeo(x, nn, mm, n)
binom(x, n, p)
sigma
mu
probability density ->
k ->
0.30
0.27
0.24
0.21
0.18
0.15
0.12
0.09
0.06
0.03
0.00
-2
0
2
4
6
8
10

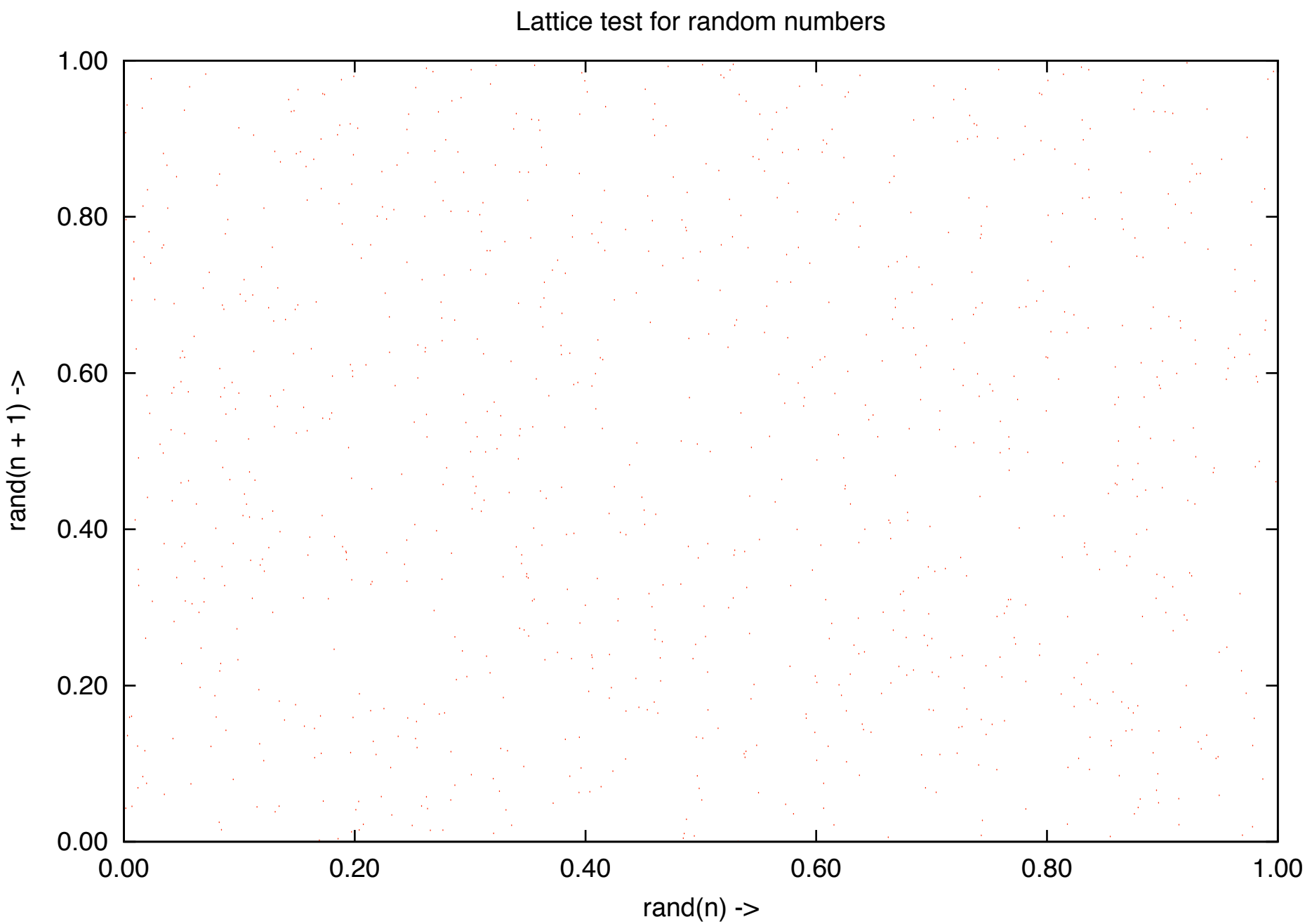

hypergeometric PDF using normal approximation
hypgeo(rnd(x), nn, mm, n)
normal(x, mu, sigma)
sigma
mu
probability density ->
k, x ->
0.30
0.27
0.24
0.21
0.18
0.15
0.12
0.09
0.06
0.03
0.00
-2
0
2
4
6
8
10

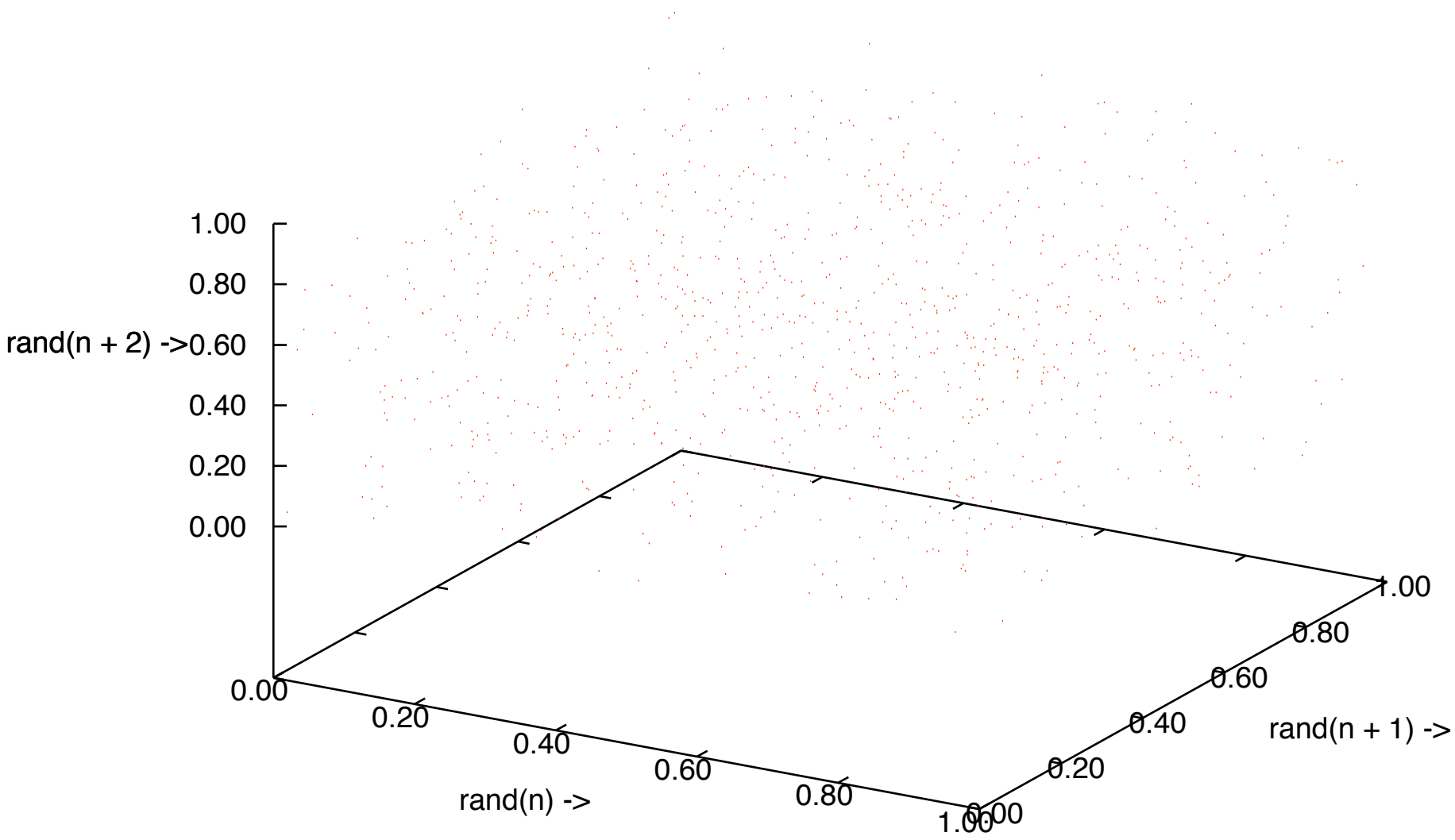

negative binomial PDF using gamma approximation
negbin(rnd(x), r, p)
gmm(x, rho, lambda)
probability density ->
k, x ->
0.17
0.15
0.14
0.12
0.10
0.09
0.07
0.05
0.03
0.02
0.00
0
5
10
15
sigma
mu


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

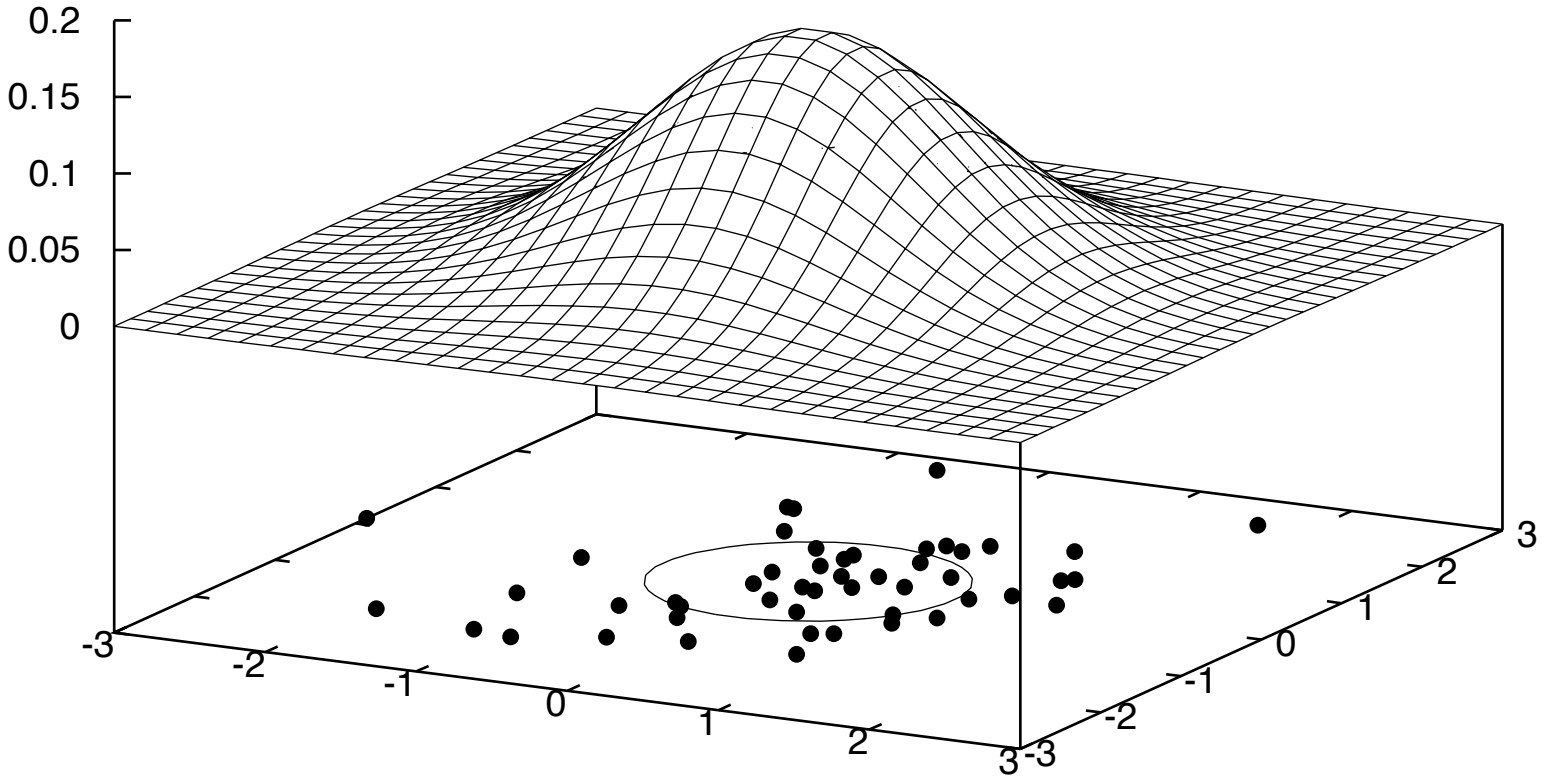

normal PDF using logistic approximation
logistic(x, a, lambda)
normal(x, mu, sigma)
sigma
mu
probability density ->
x ->
0.33
0.30
0.27
0.23
0.20
0.17
0.13
0.10
0.07
0.03
0.00
-4.0
-2.0
0.0
2.0
4.0
6.0

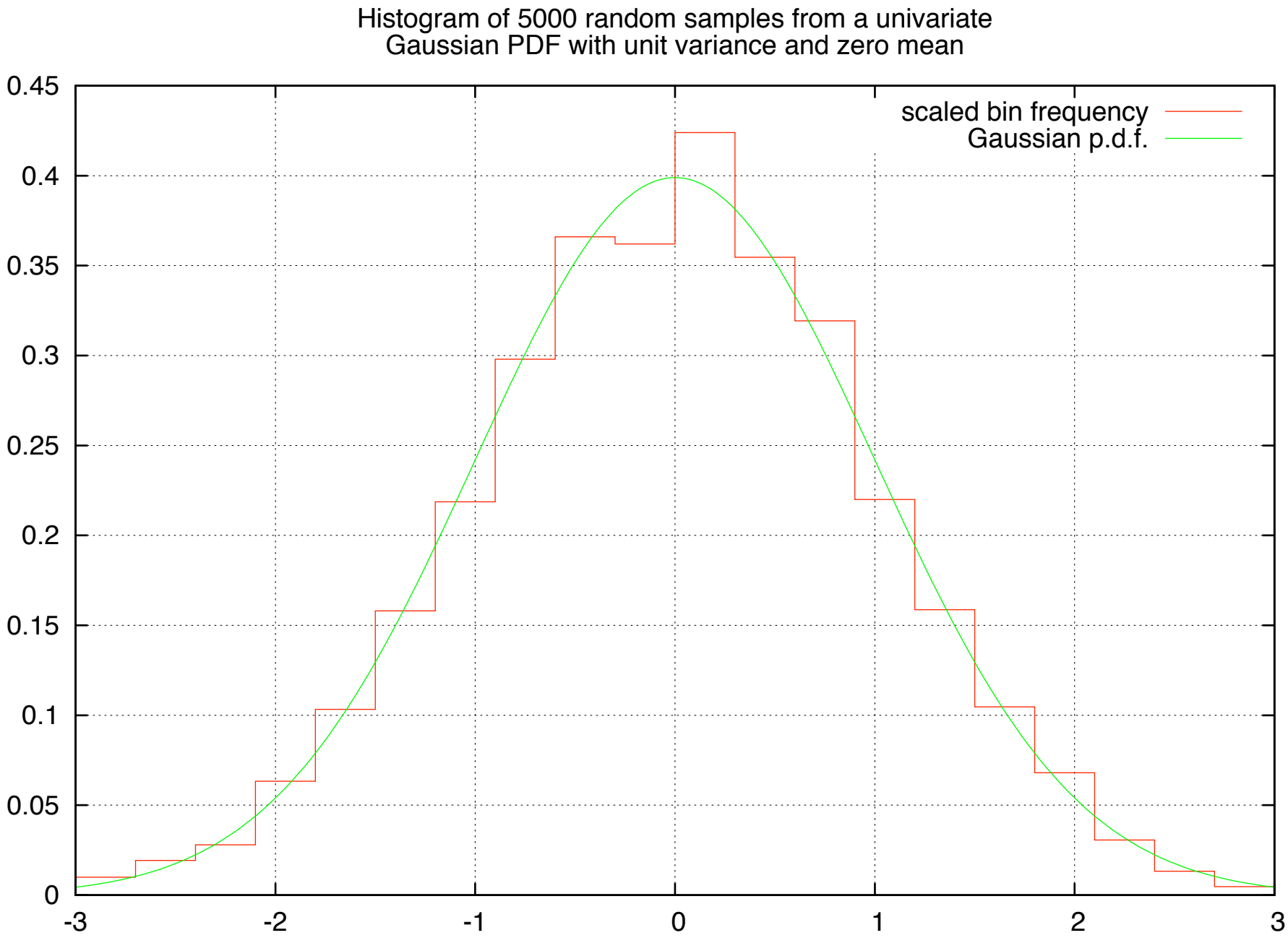

poisson PDF using normal approximation
poisson(rnd(x), mu)
normal(x, mu, sigma)
sigma
mu
probability density ->
k, x ->
0.19
0.17
0.15
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
-2
0
2
4
6
8
10
12
14


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

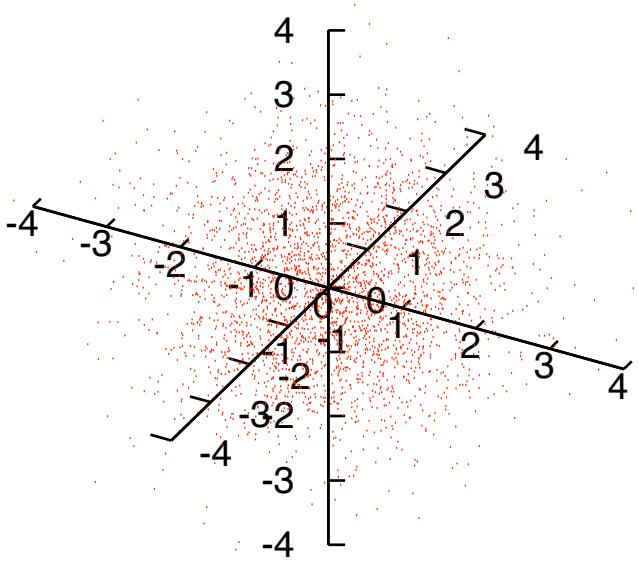

Lattice test for random numbers
rand(n + 2) ->
1.00
0.80
0.60
0.40
0.20
0.00
0.00
0.20
0.40
0.60
0.80
1.00
rand(n) ->
0.00
0.20
0.40
0.60
0.80
1.00
rand(n + 1) ->

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

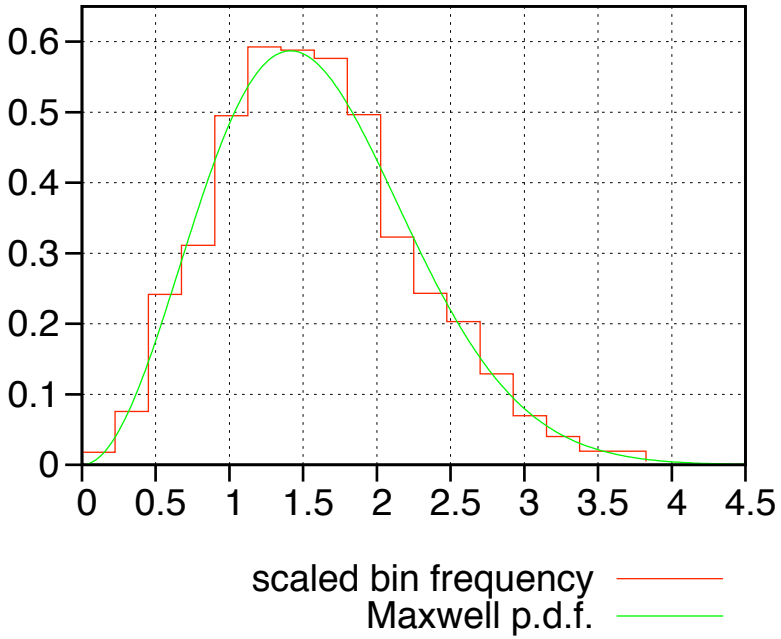


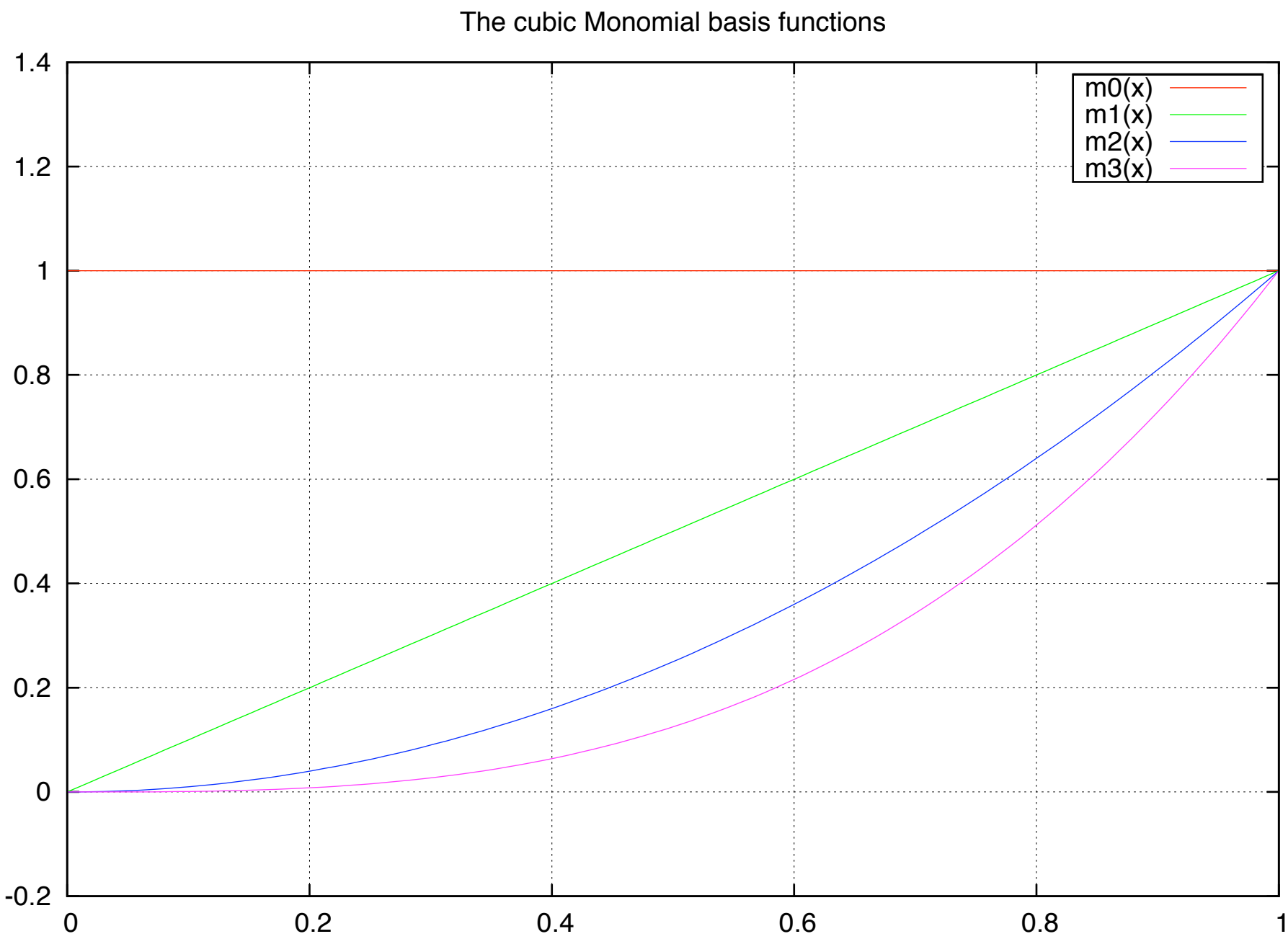

Histogram of 5000 random samples from a univariate
Gaussian PDF with unit variance and zero mean
scaled bin frequency
Gaussian p.d.f.
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-3
-2
-1
0
1
2
3

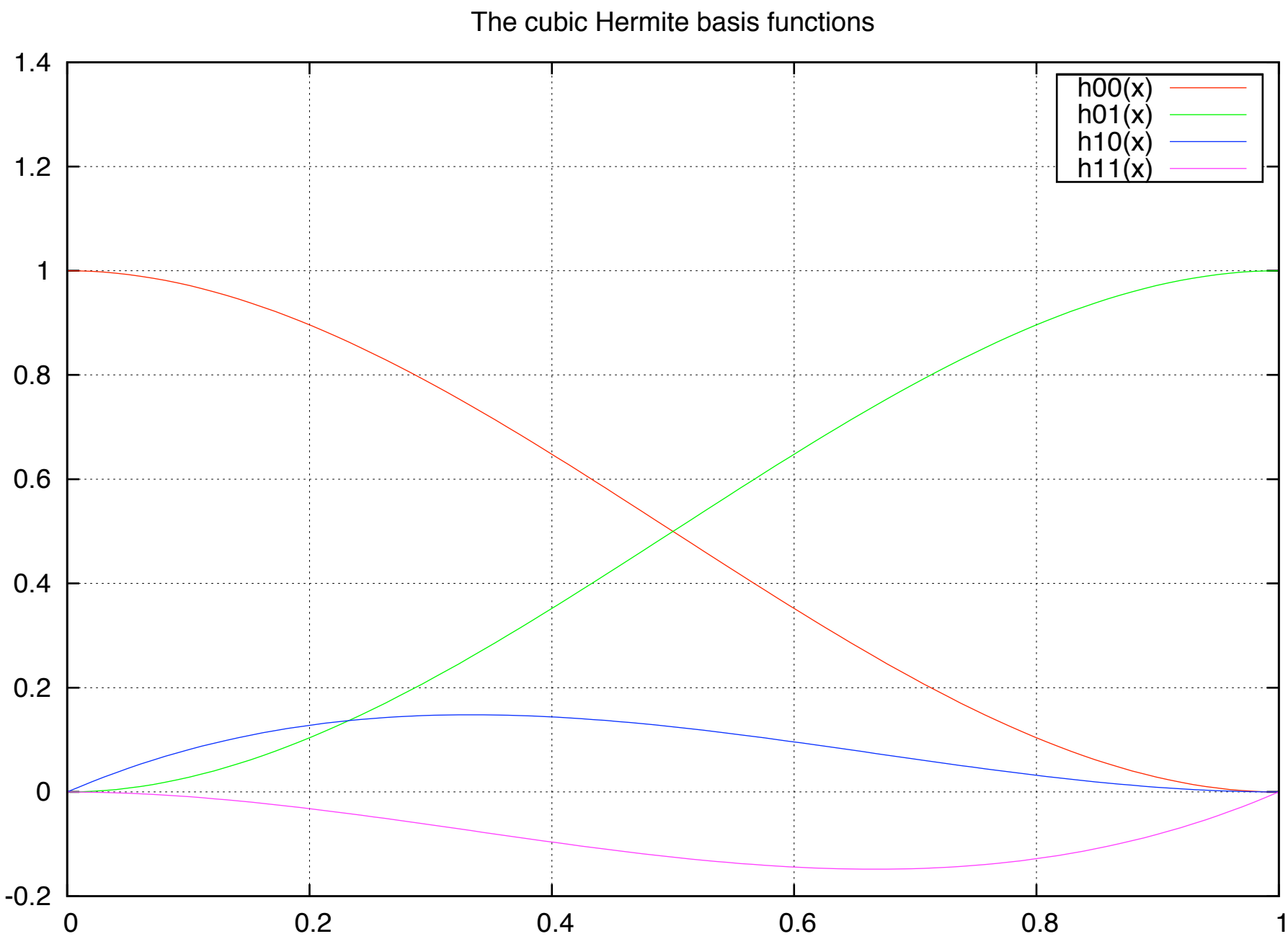
Gaussian 3D cloud of 3000 random samples

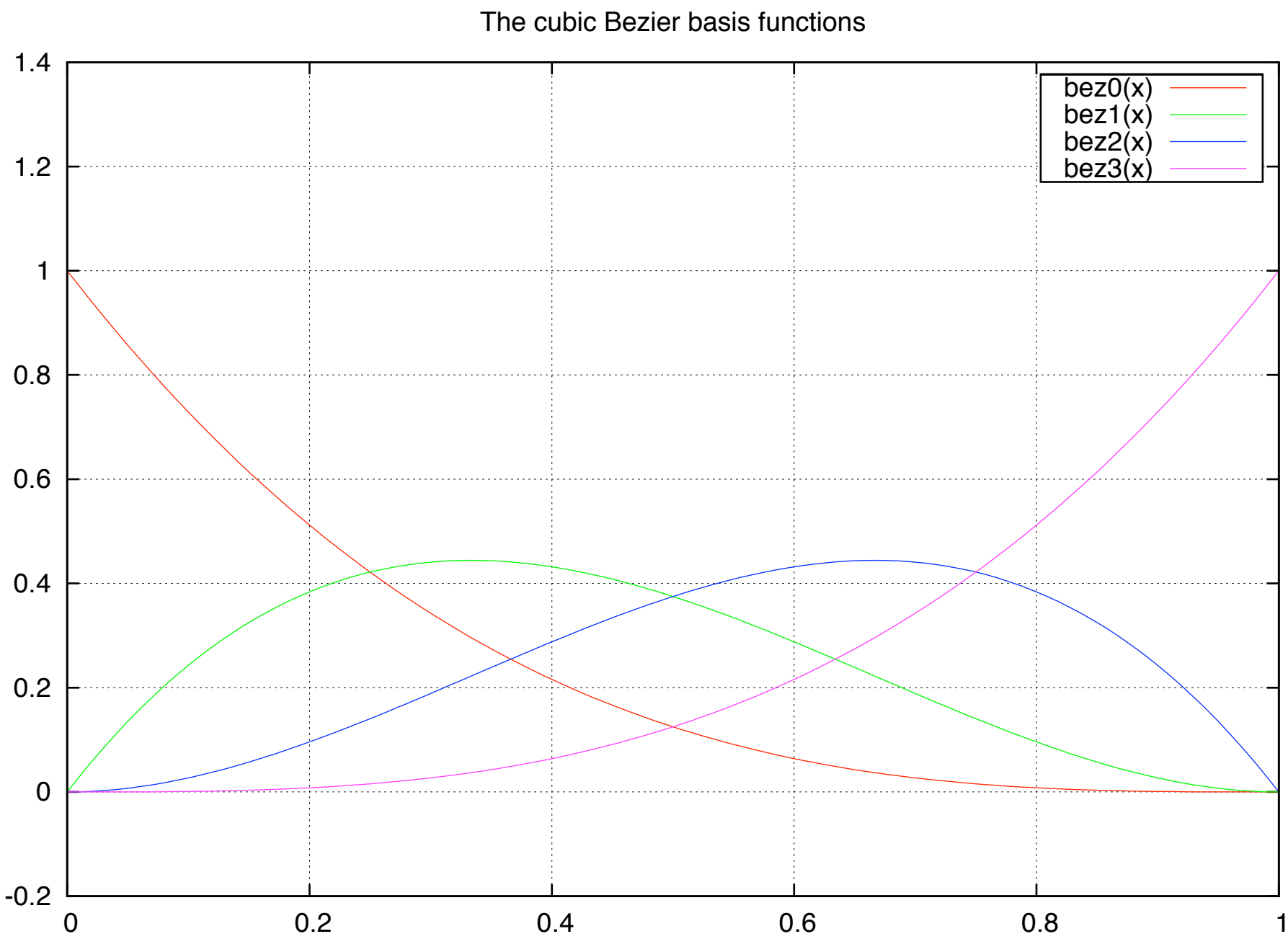


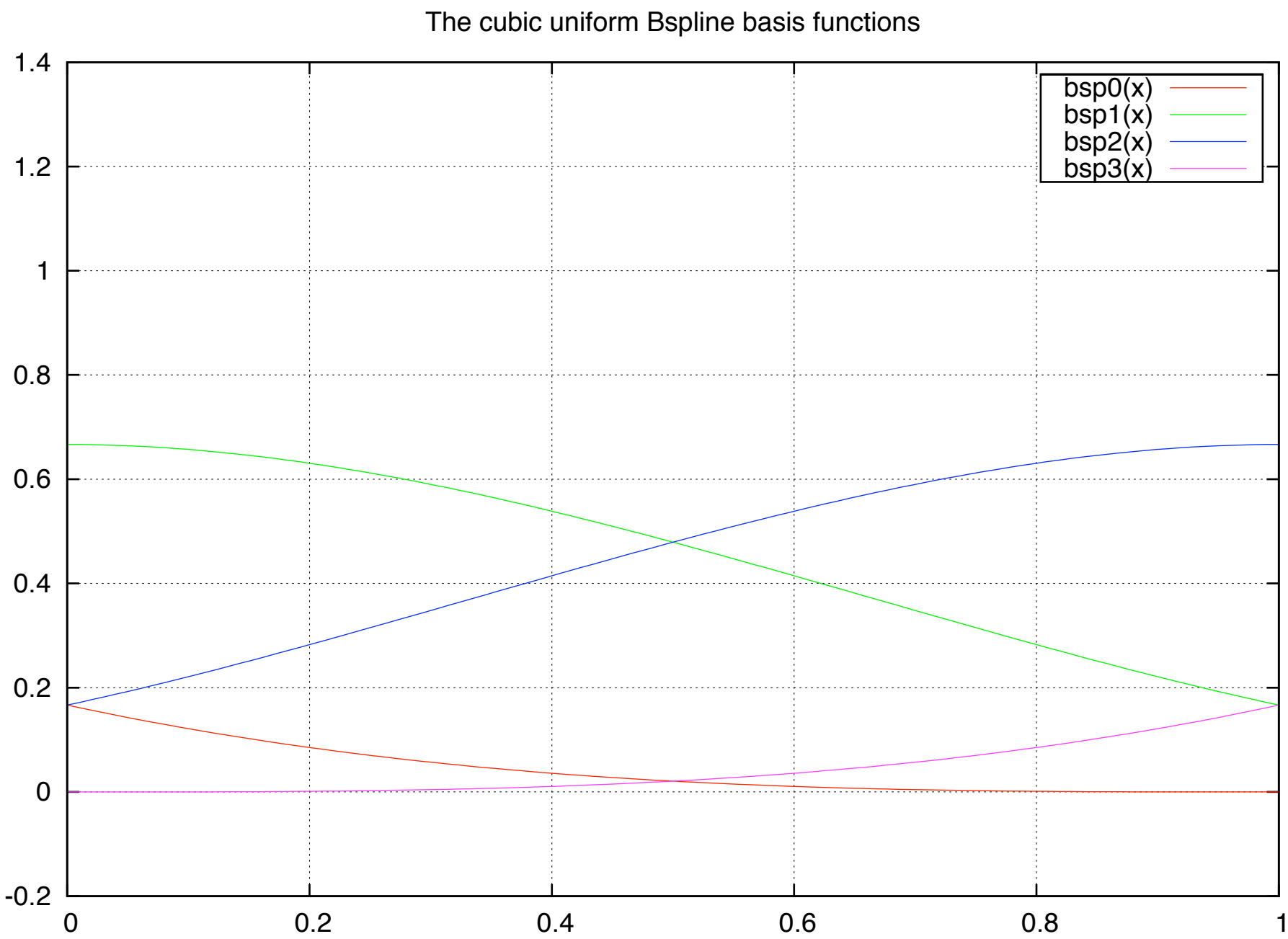
Histogram of distance from origin of
3000 multivariate unit variance samples

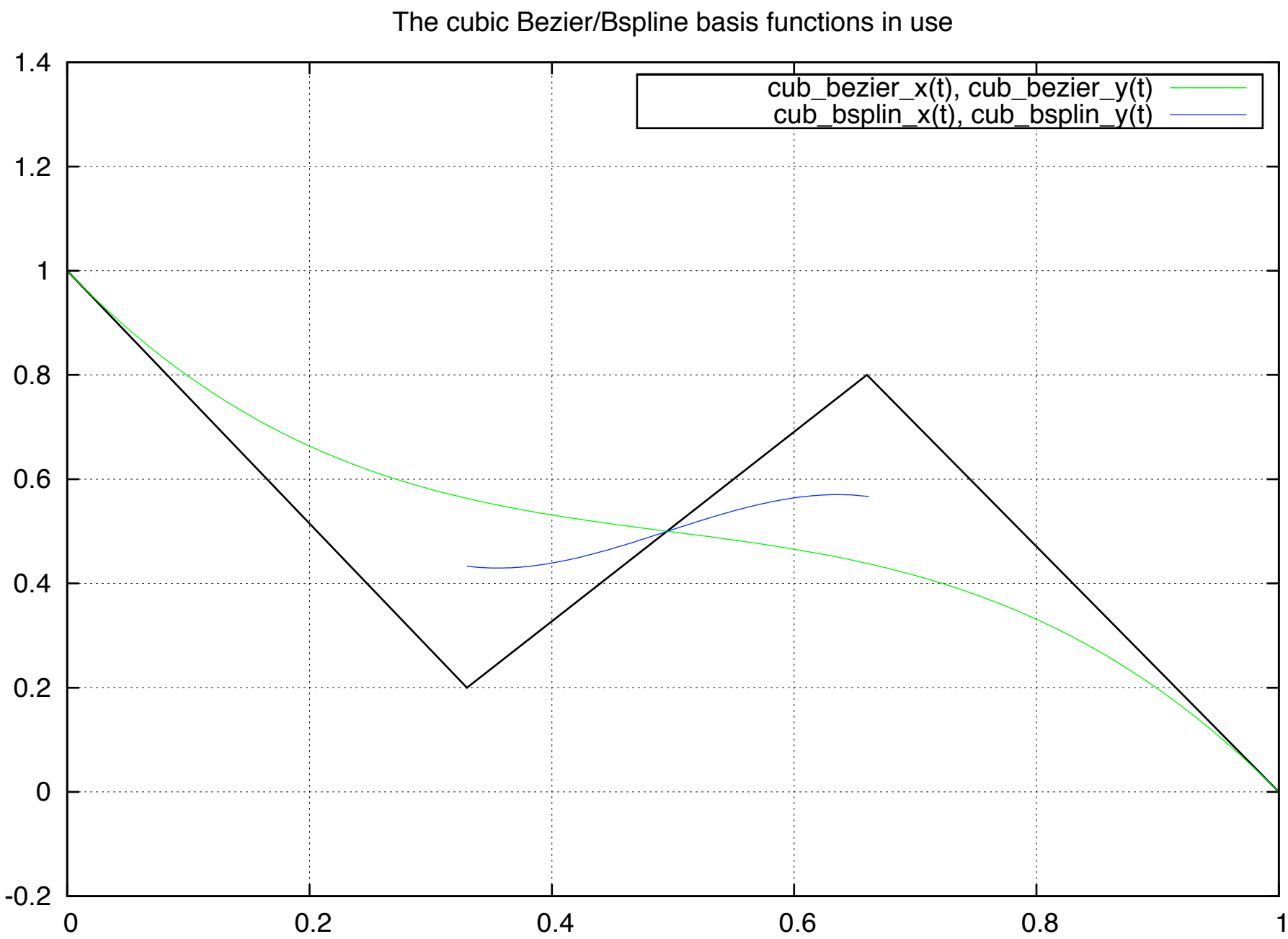


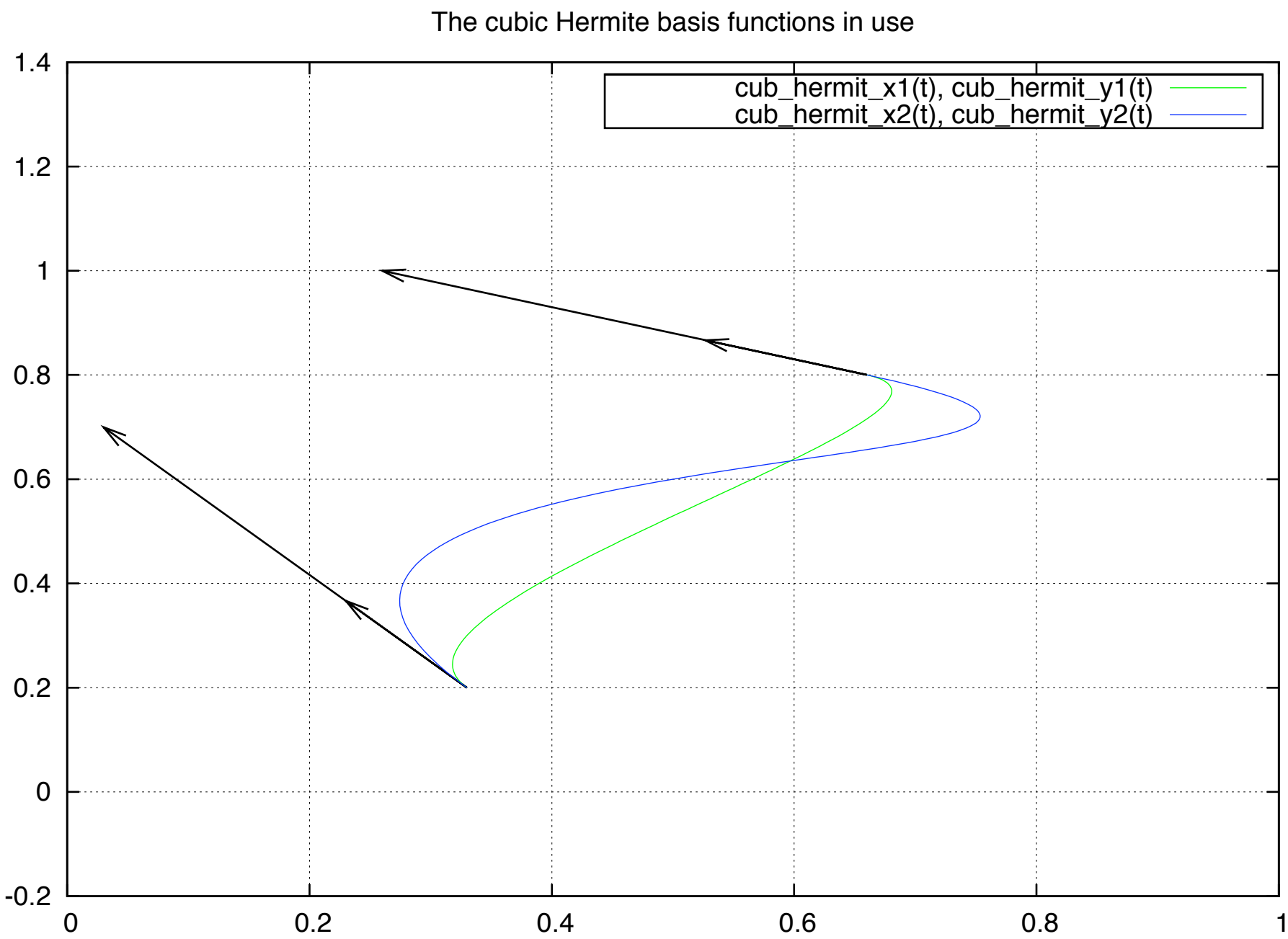

The cubic Monomial basis functions
m0(x)
m1(x)
m2(x)
m3(x)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1

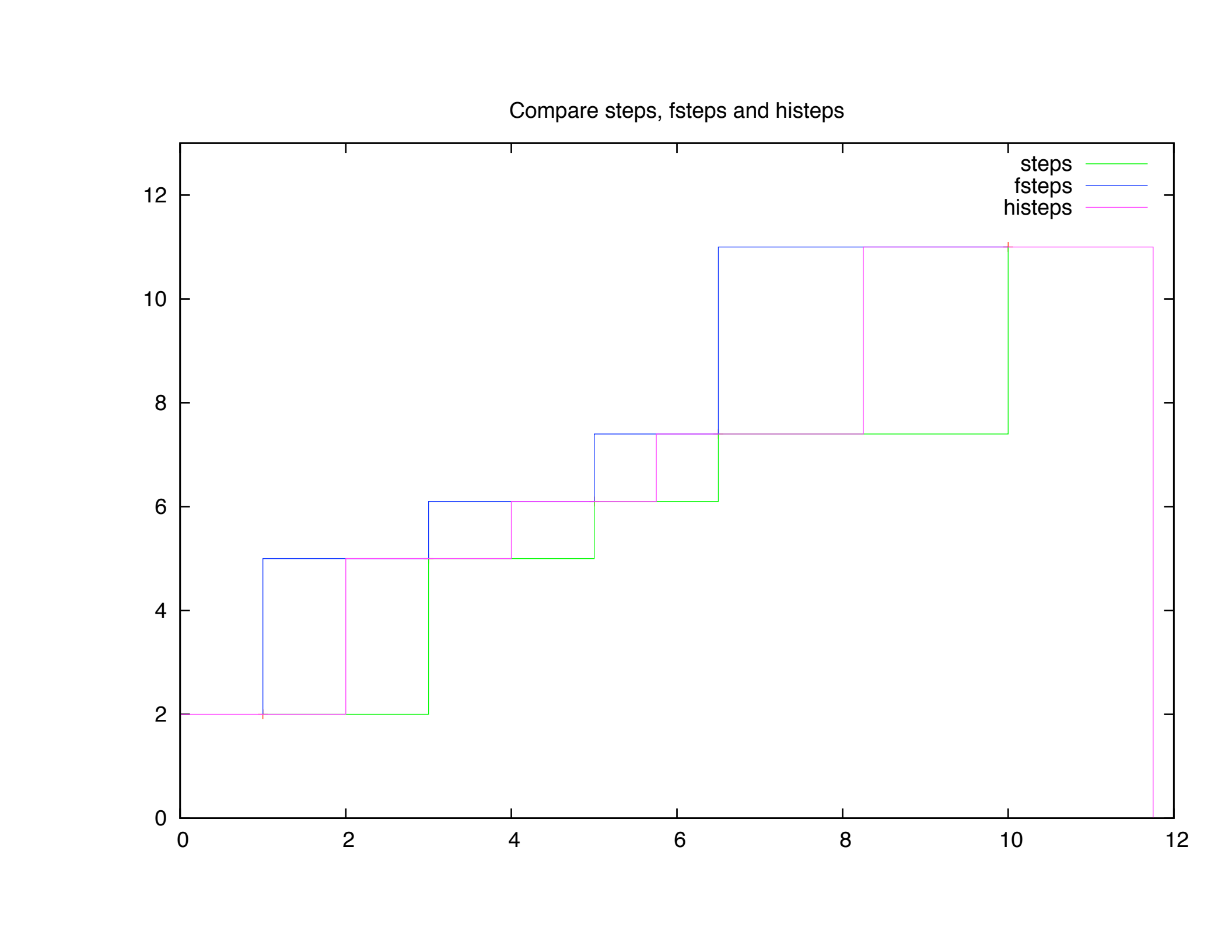

The cubic Hermite basis functions
h00(x)
h01(x)
h10(x)
h11(x)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1

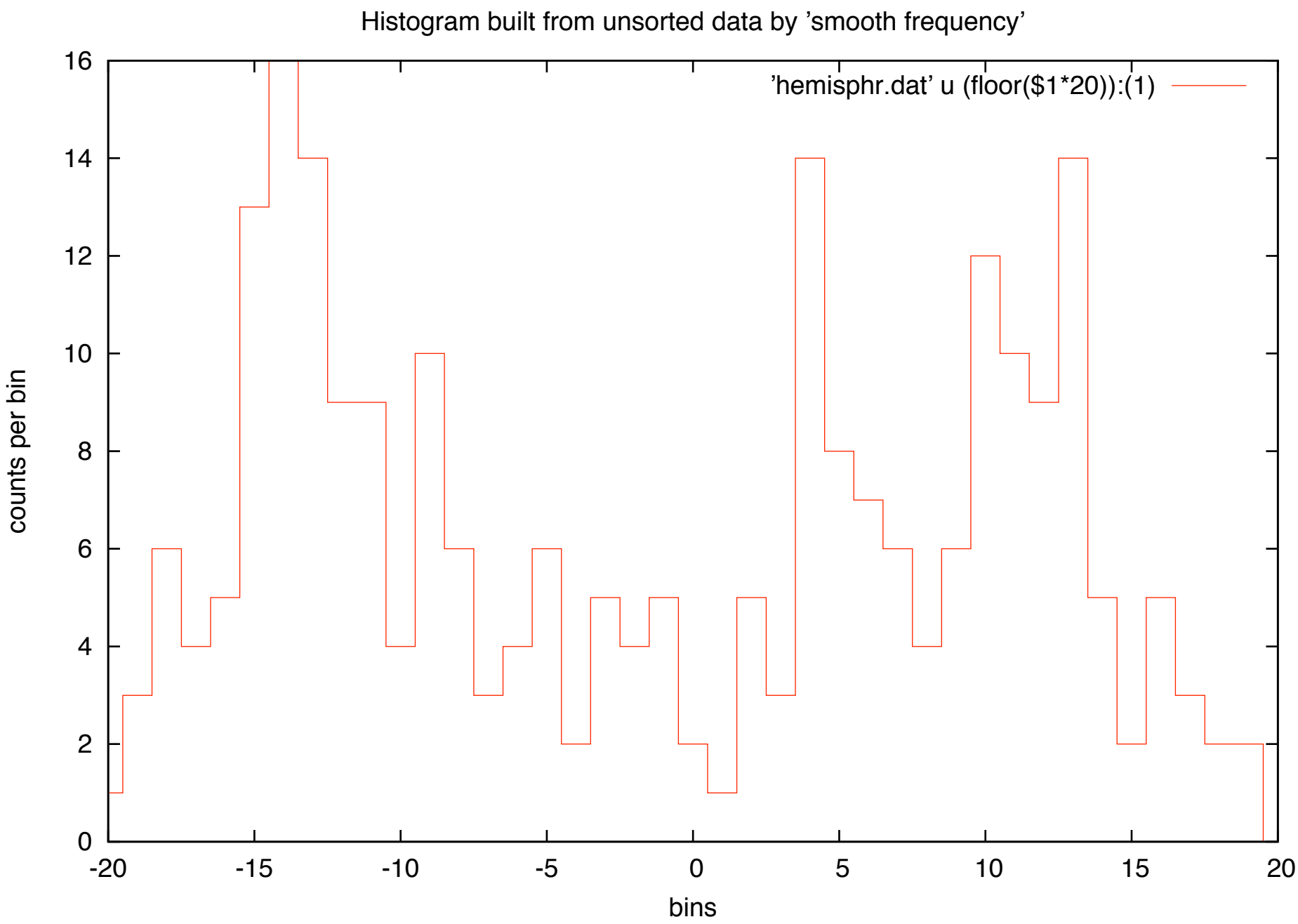

The cubic Bezier basis functions
bez0(x)
bez1(x)
bez2(x)
bez3(x)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1

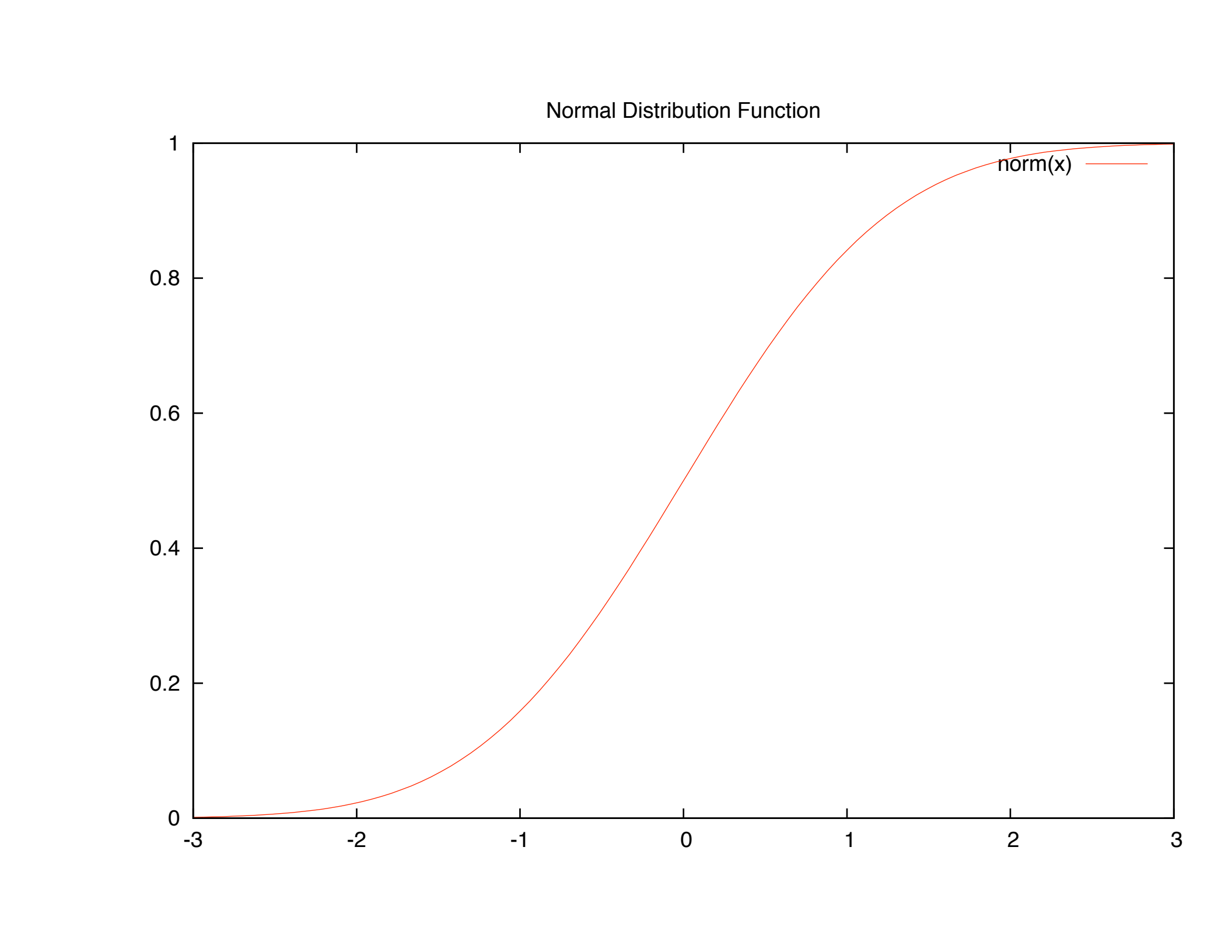

The cubic uniform Bspline basis functions
bsp0(x)
bsp1(x)
bsp2(x)
bsp3(x)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1

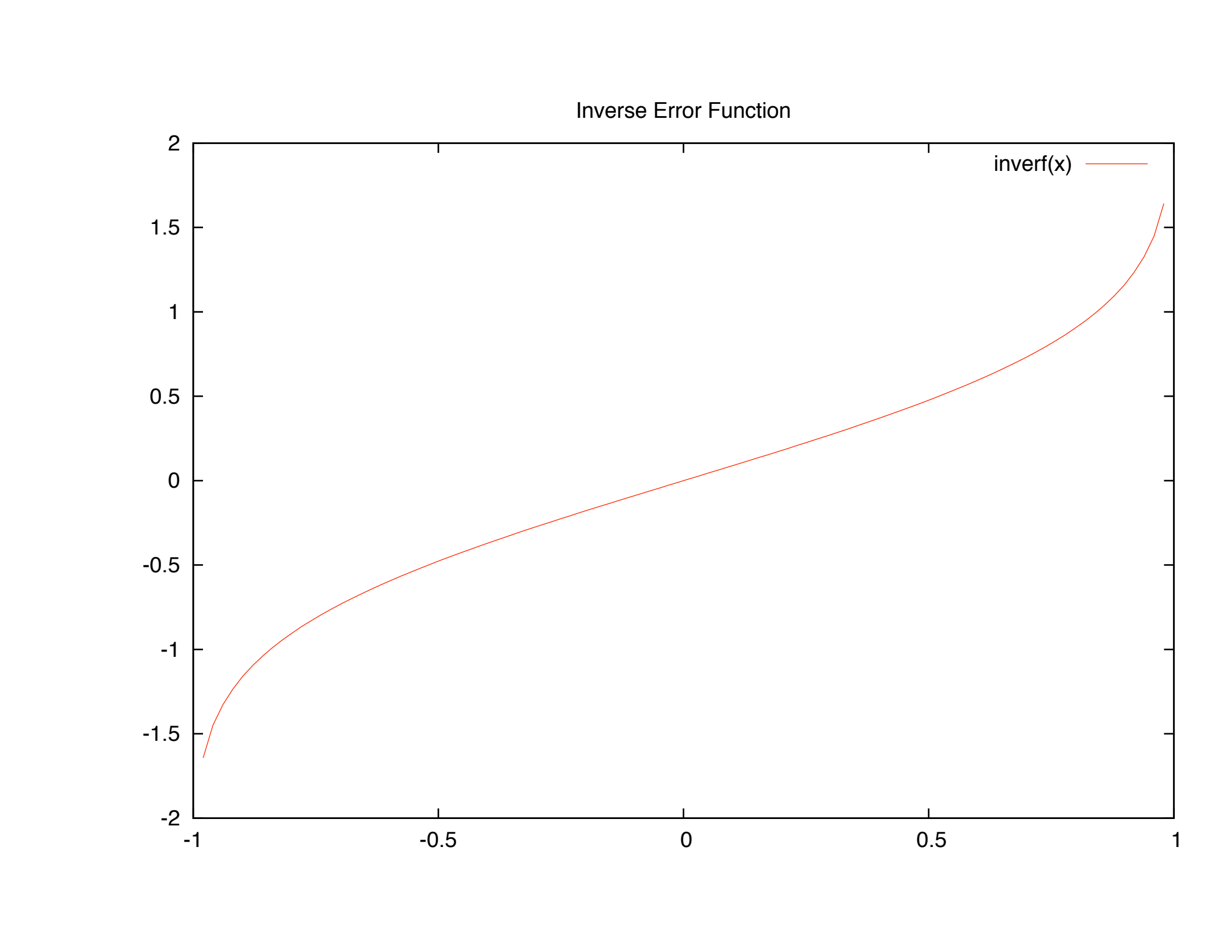

The cubic Bezier/Bspline basis functions in use
cub_bezier_x(t), cub_bezier_y(t)
cub_bsplin_x(t), cub_bsplin_y(t)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1

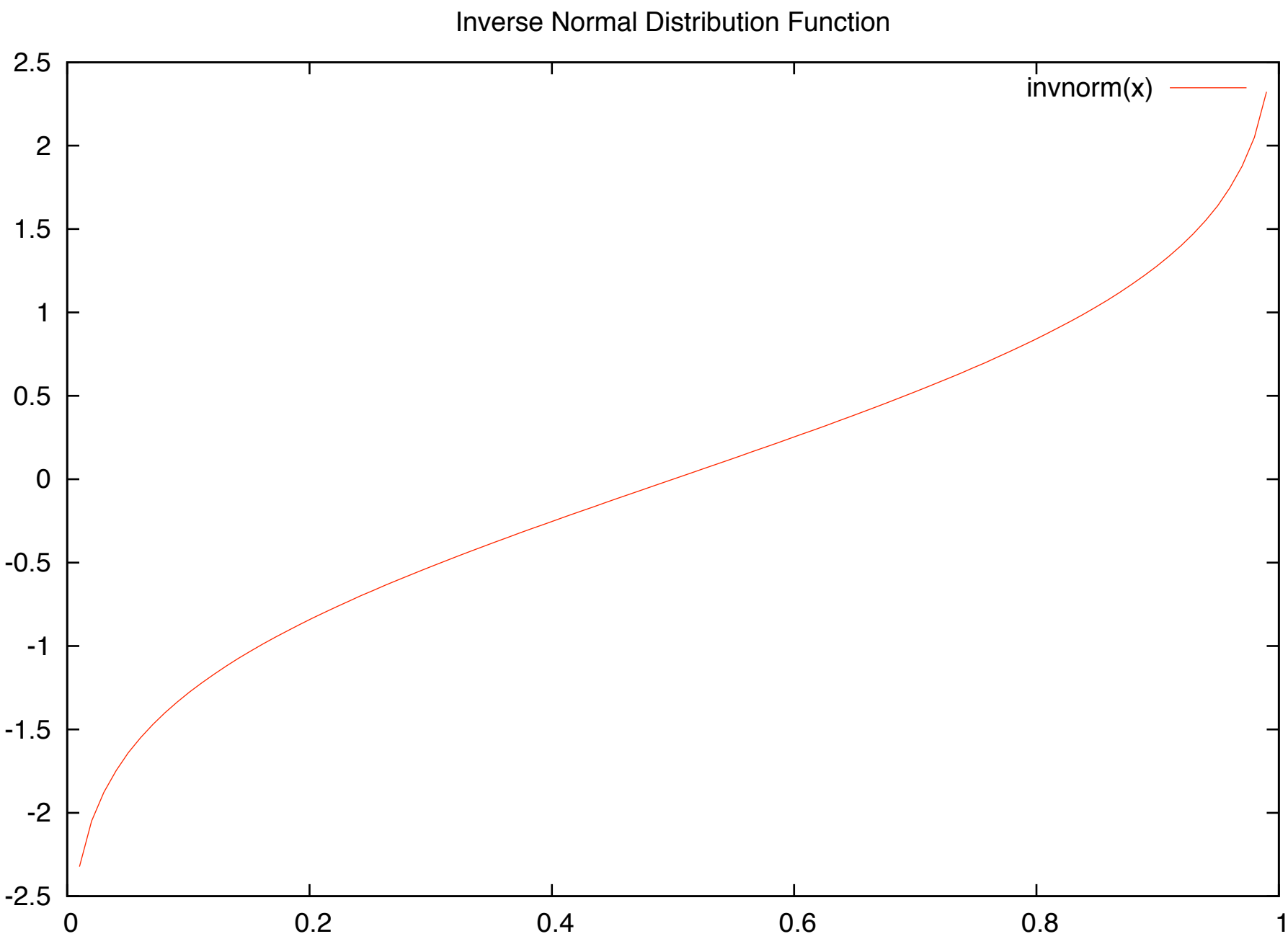

The cubic Hermite basis functions in use
cub_hermit_x1(t), cub_hermit_y1(t)
cub_hermit_x2(t), cub_hermit_y2(t)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1

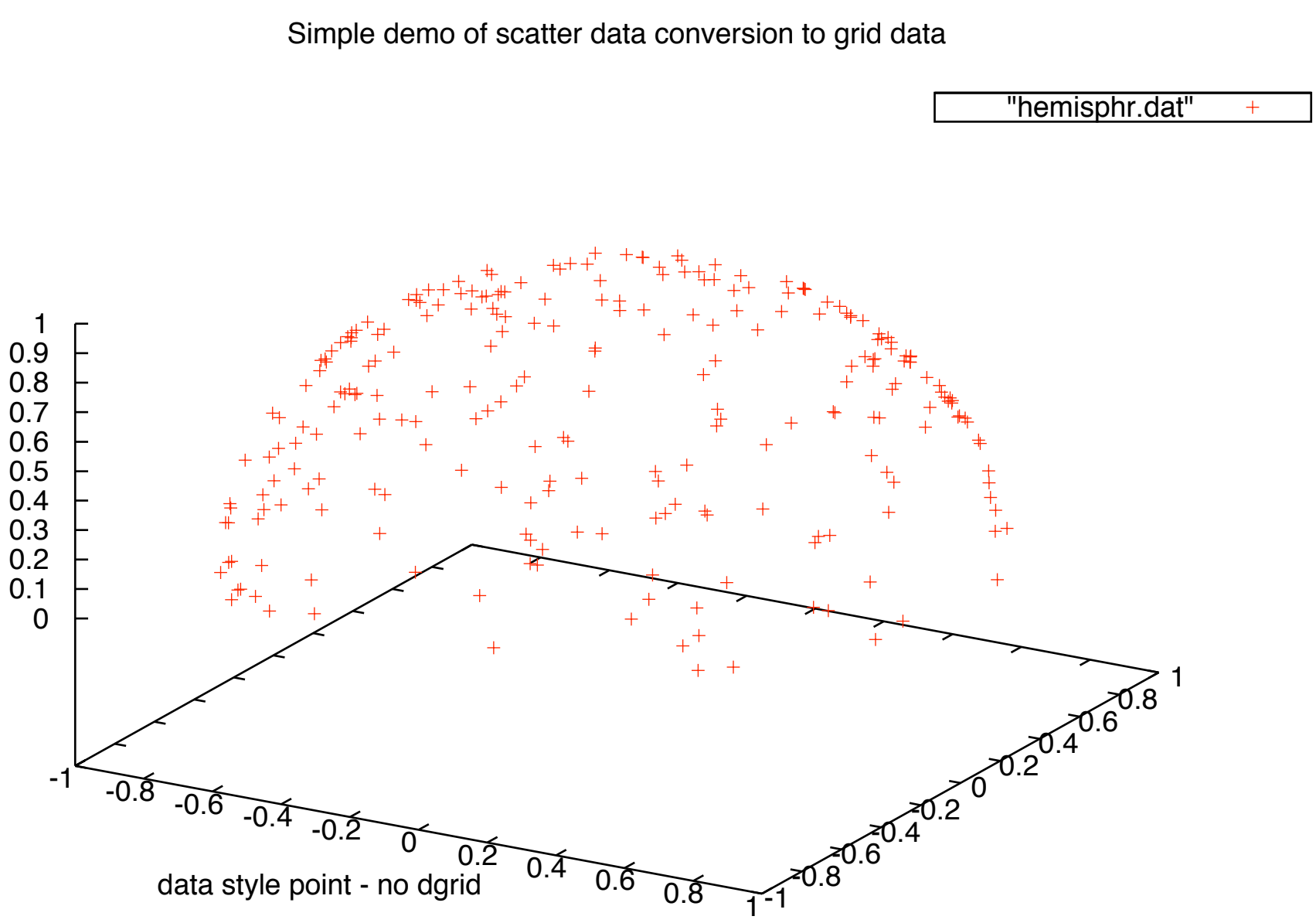

Compare steps, fsteps and histeps
steps
fsteps
histeps
0
2
4
6
8
10
12

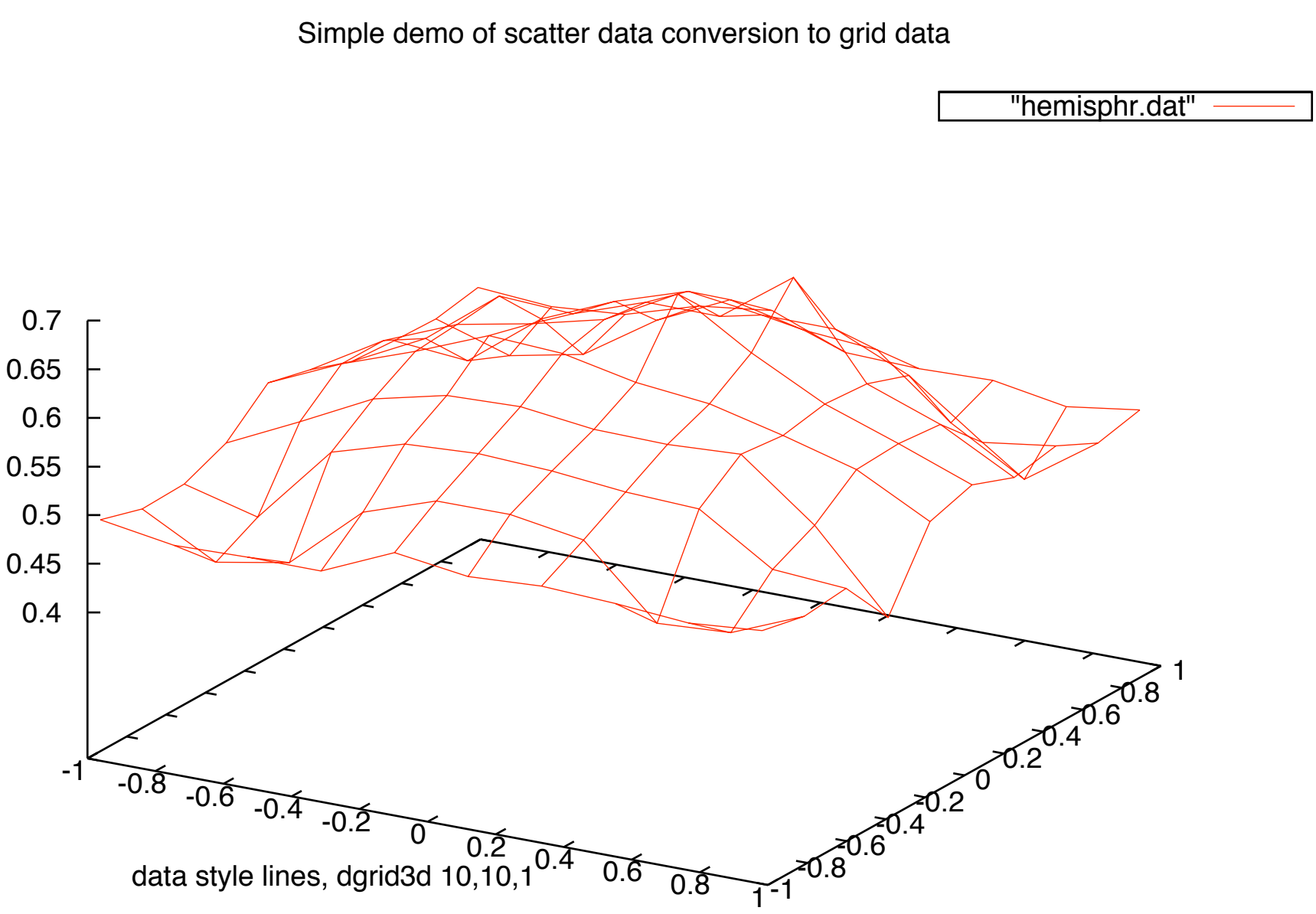

Histogram built from unsorted data by 'smooth frequency'
'hemisphr.dat' u (floor($1*20)):(1)
counts per bin
bins
0
2
4
6
8
10
12
14
16
-20
-15
-10
-5
0
5
10
15
20

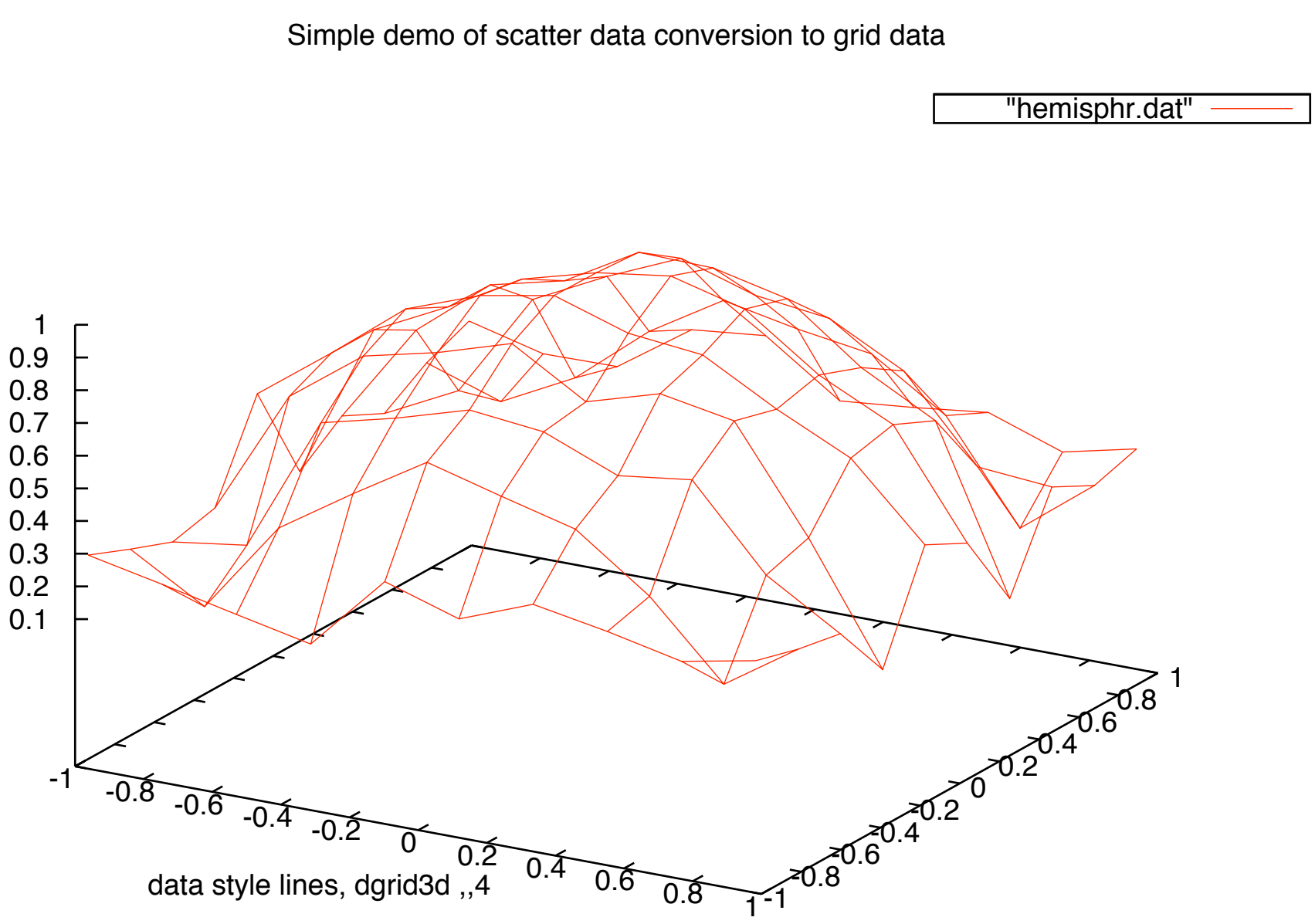

Normal Distribution Function
norm(x)
1
0.8
0.6
0.4
0.2
0
-3
-2
-1
0
1
2
3

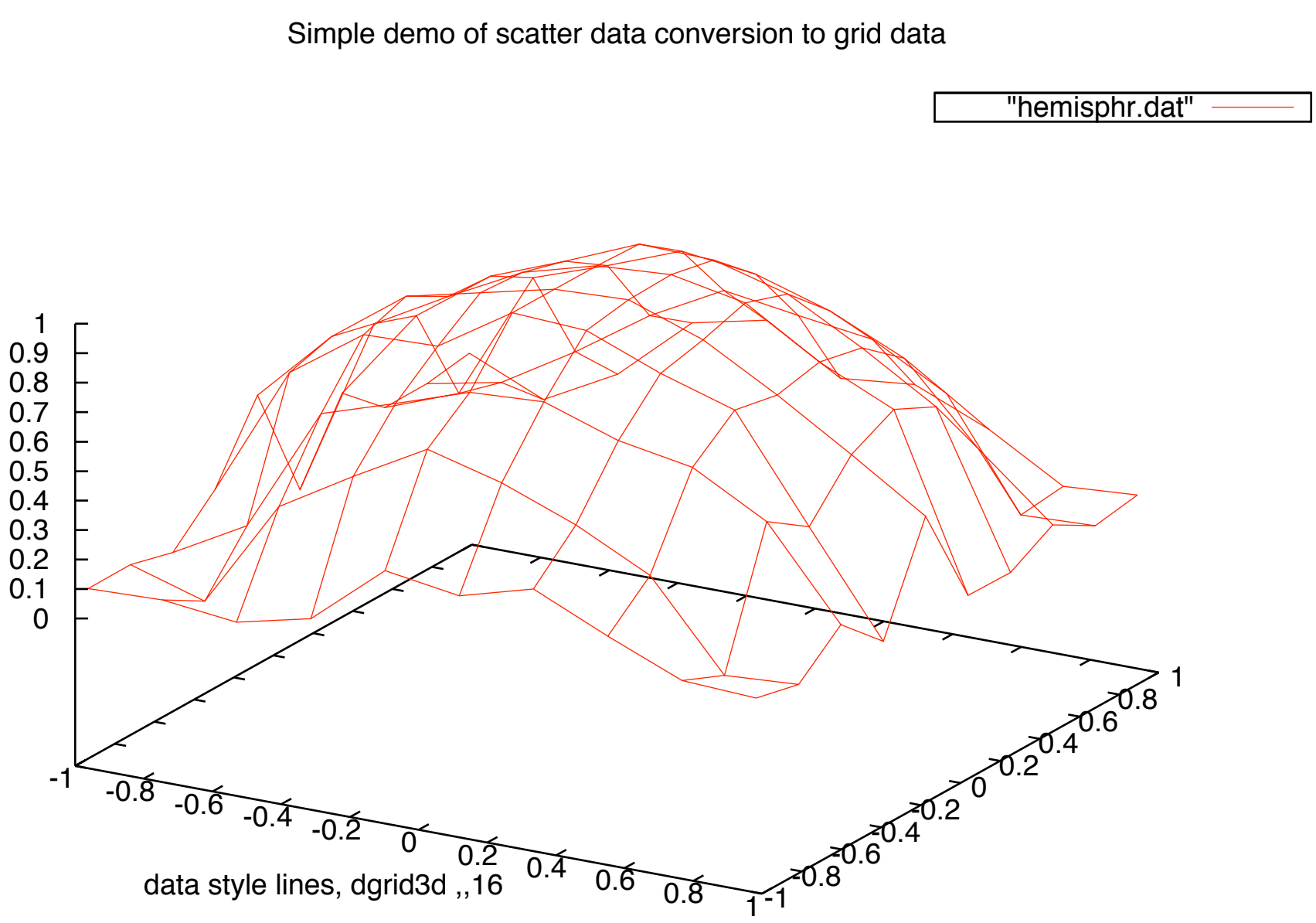

Inverse Error Function
inverf(x)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1
-0.5
0
0.5
1

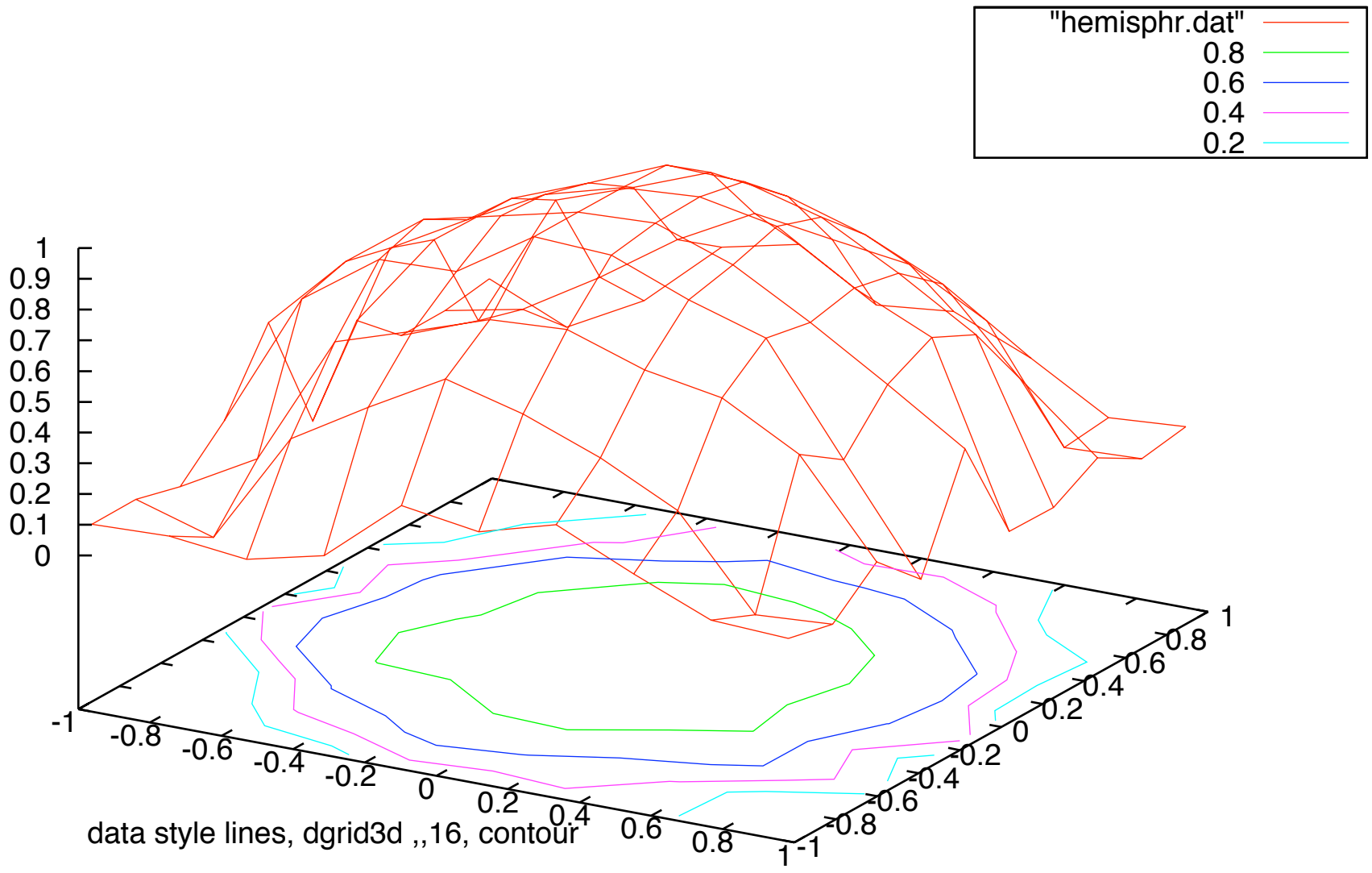

Inverse Normal Distribution Function
invnorm(x)
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
0.2
0.4
0.6
0.8
1

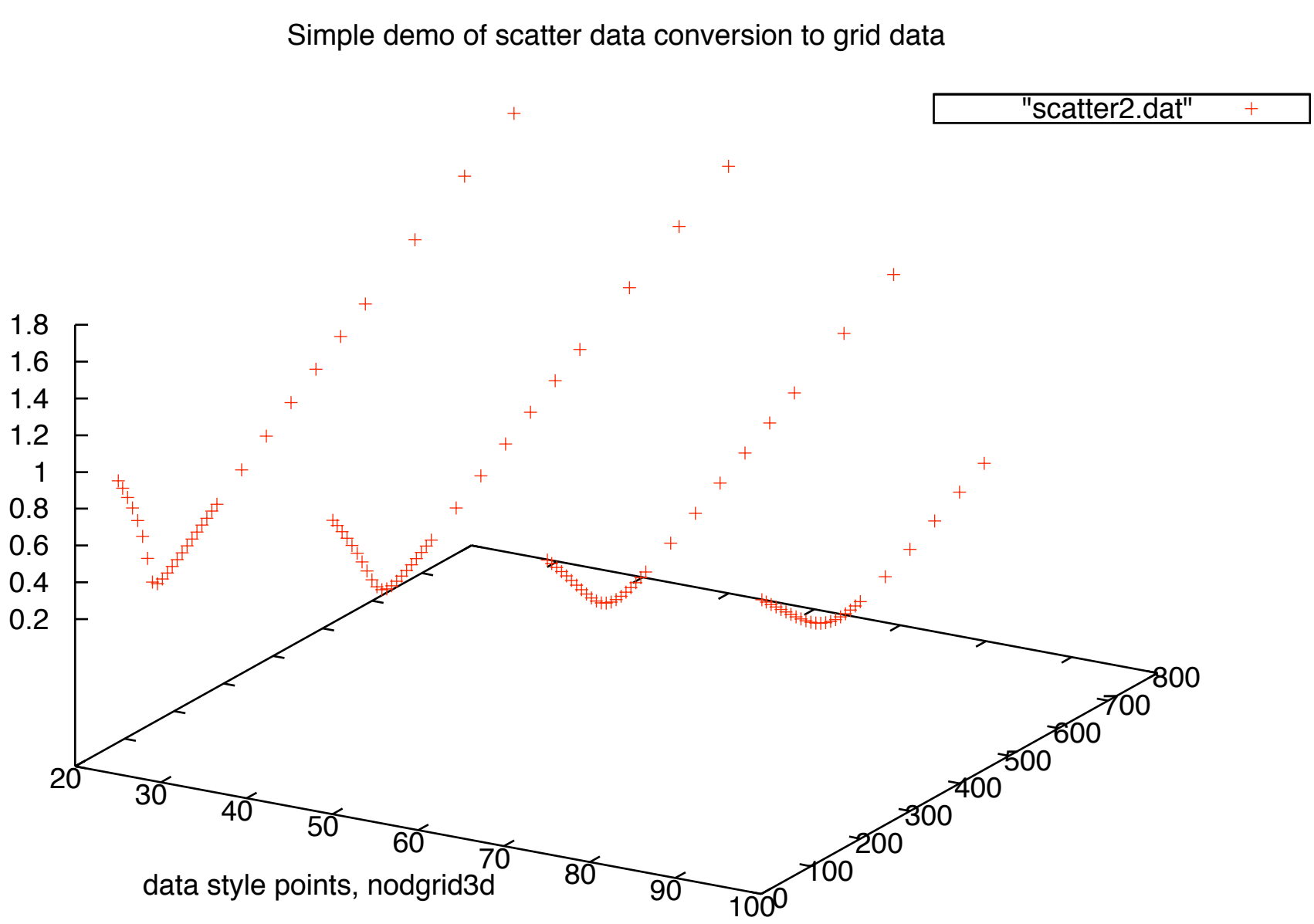

Simple demo of scatter data conversion to grid data
"hemisphr.dat"
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
data style point - no dgrid

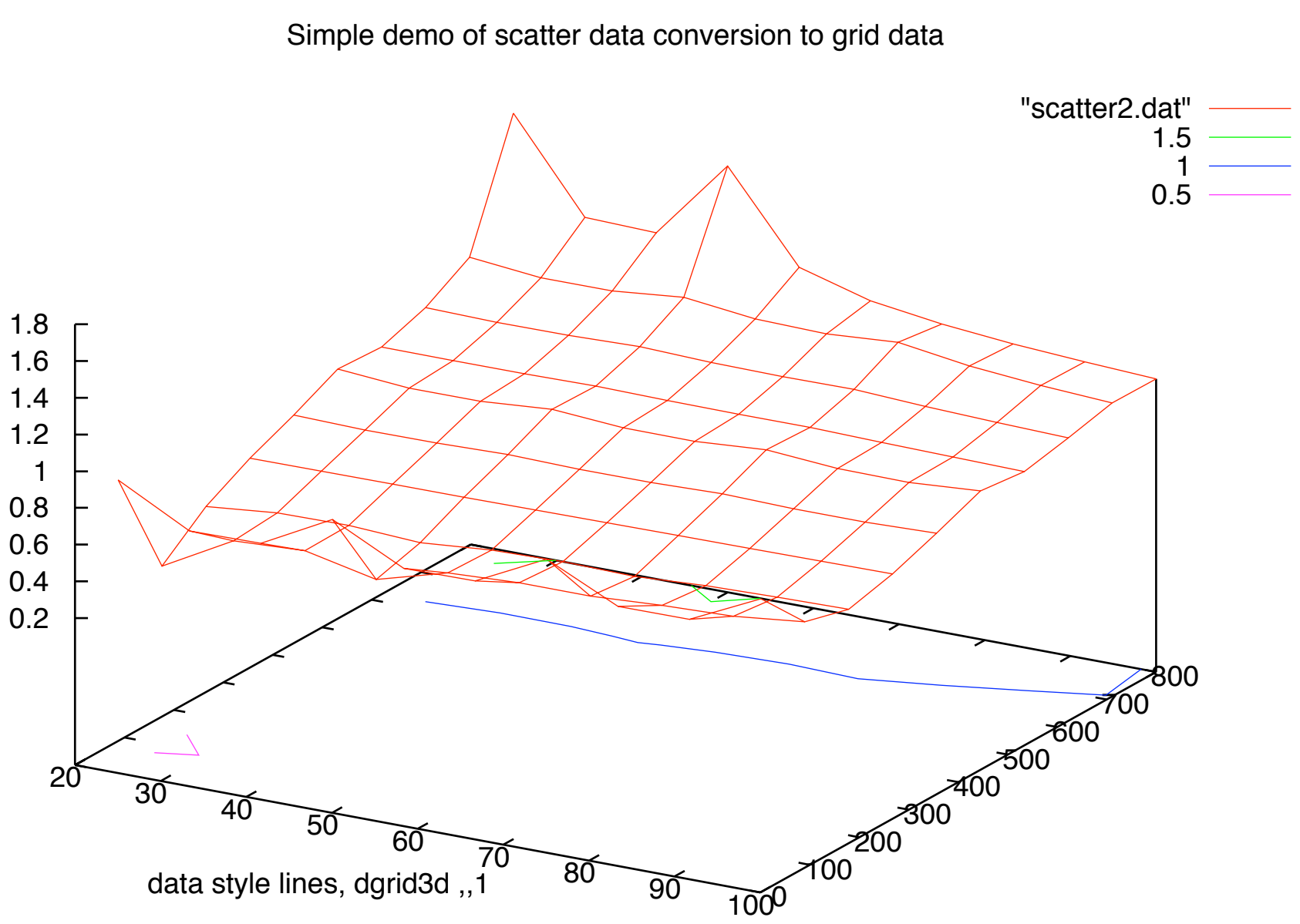

Simple demo of scatter data conversion to grid data
"hemisphr.dat"
0.7
0.65
0.6
0.55
0.5
0.45
0.4
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
data style lines, dgrid3d 10,10,1

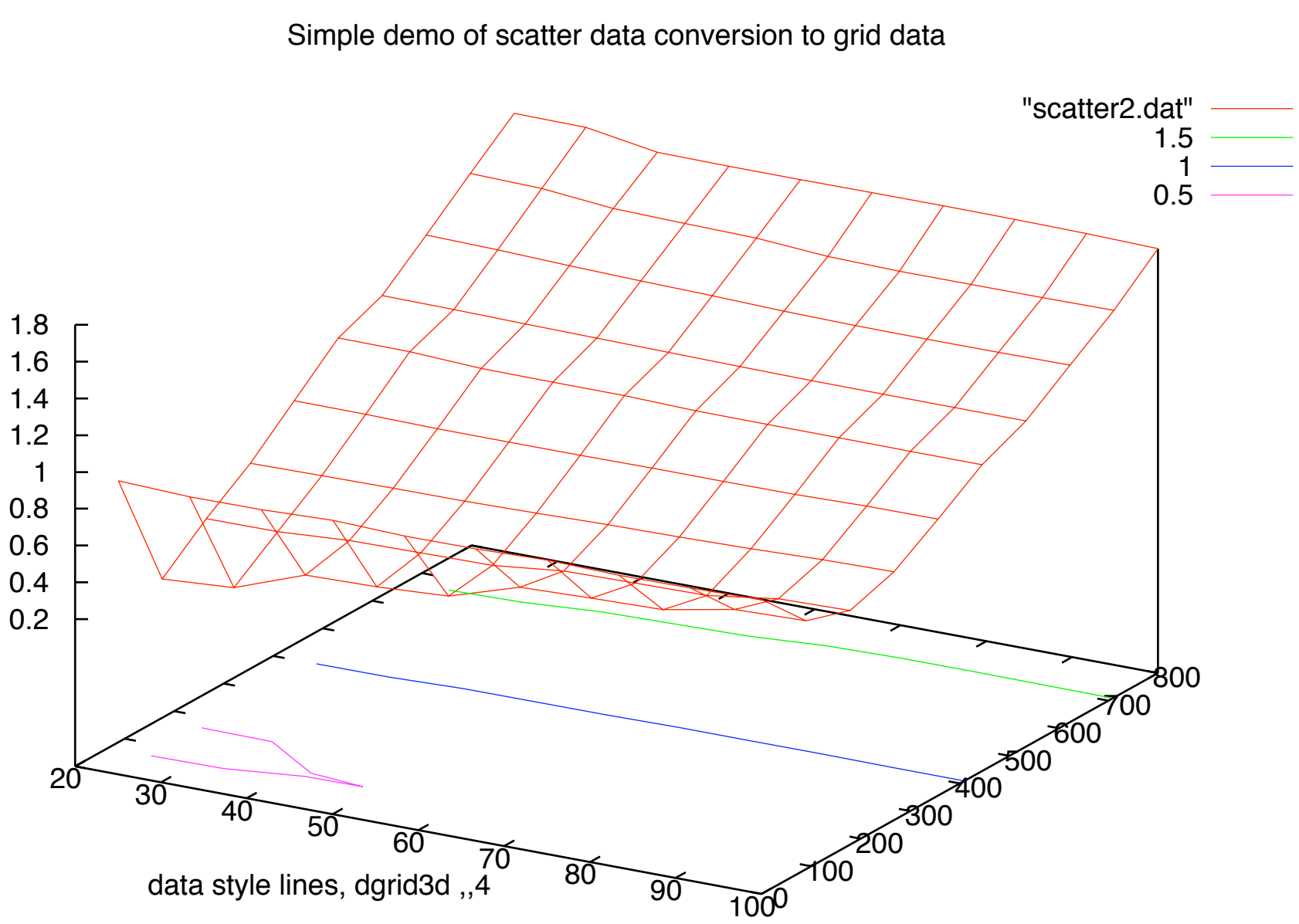

Simple demo of scatter data conversion to grid data
"hemisphr.dat"
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
data style lines, dgrid3d ,,4

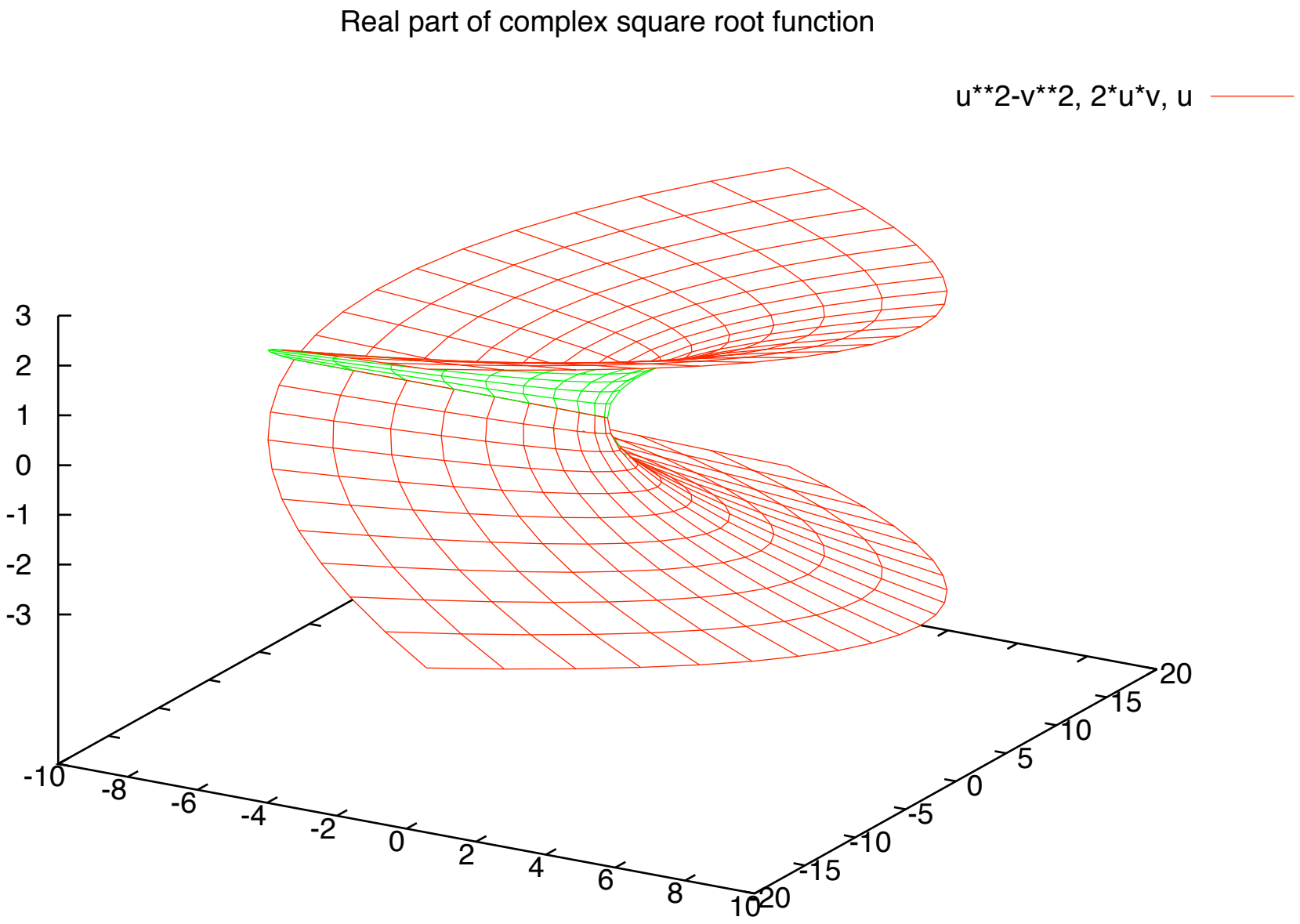

Simple demo of scatter data conversion to grid data
"hemisphr.dat"
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
data style lines, dgrid3d ,,16


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

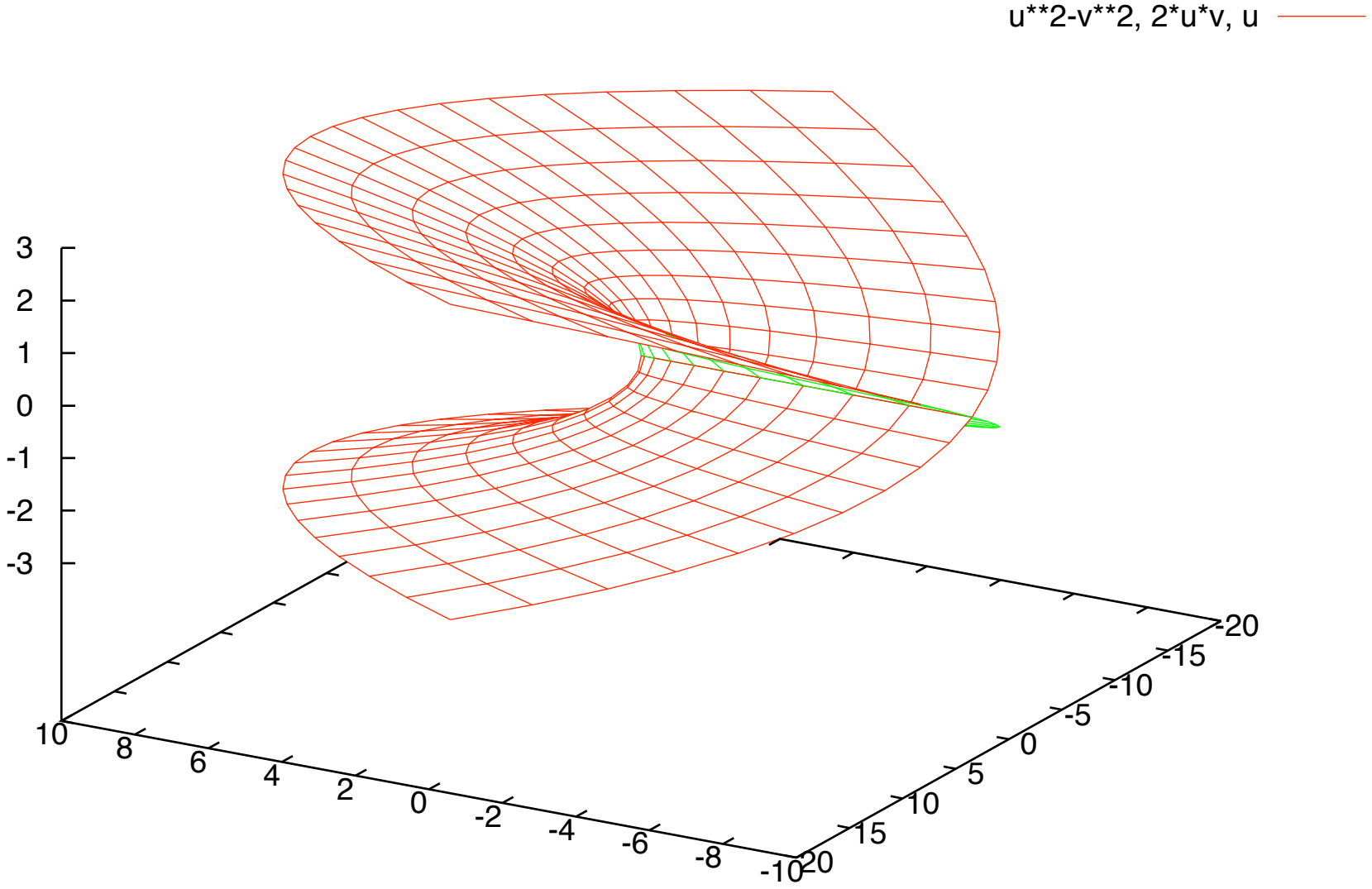

Simple demo of scatter data conversion to grid data
"scatter2.dat"
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500
600
700
800
data style points, nodgrid3d

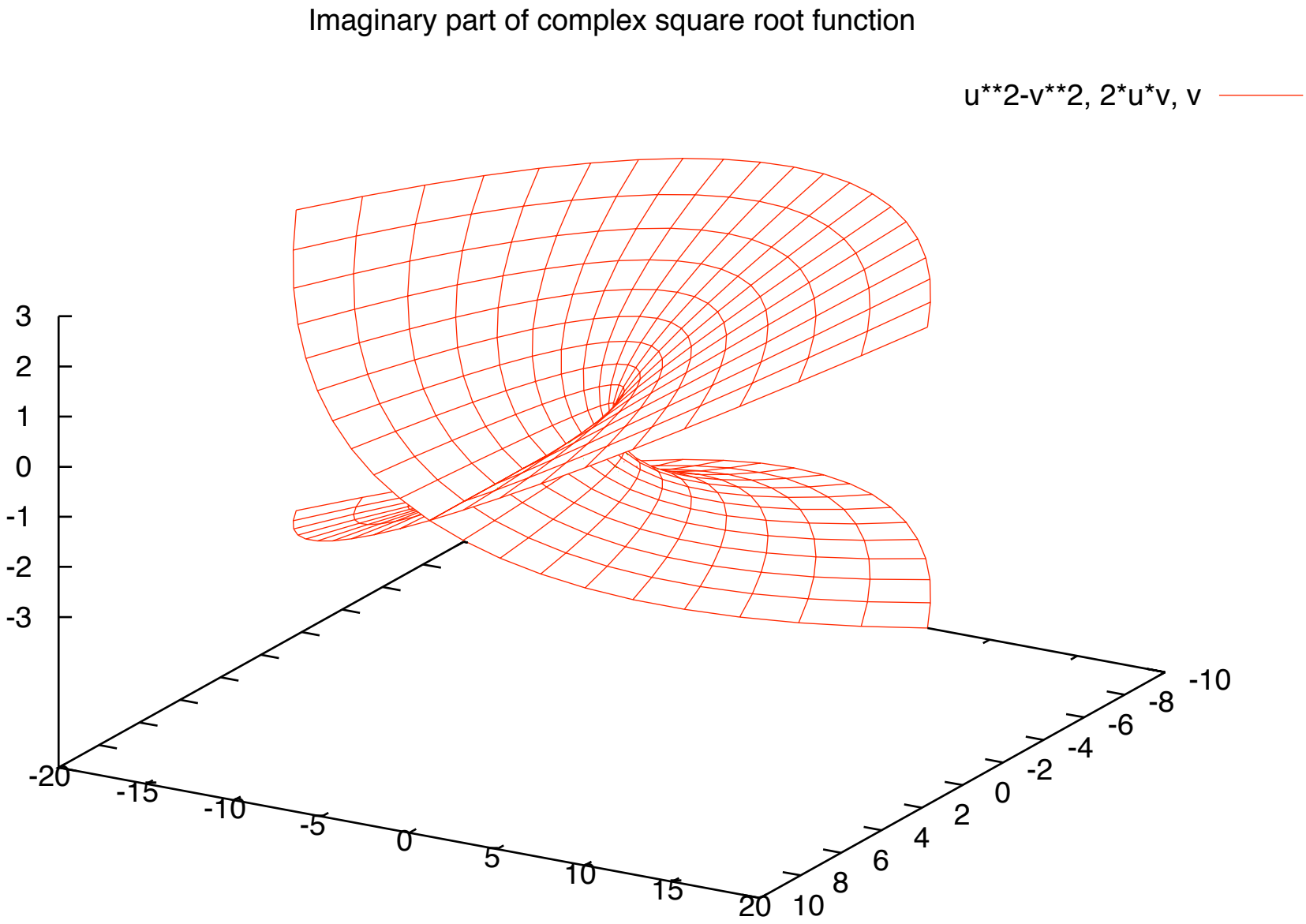

Simple demo of scatter data conversion to grid data
"scatter2.dat"
1.5
1
0.5
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500
600
700
800
data style lines, dgrid3d ,,1


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

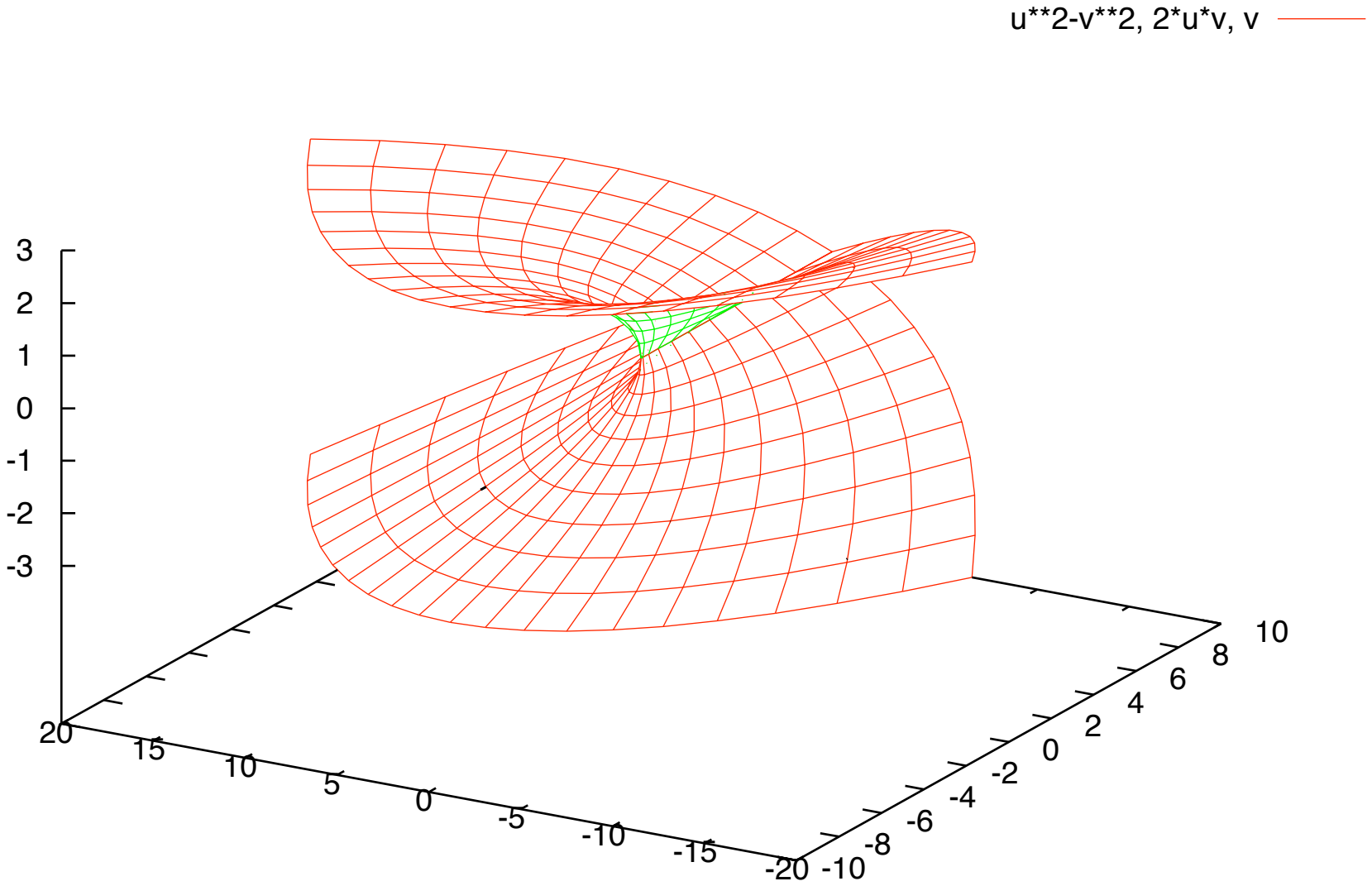

Real part of complex square root function
u**2-v**2, 2*u*v, u
3
2
1
0
-1
-2
-3
-10
-8
-6
-4
-2
0
2
4
6
8
10
20
15
10
5
0
-5
-10
-15
-20

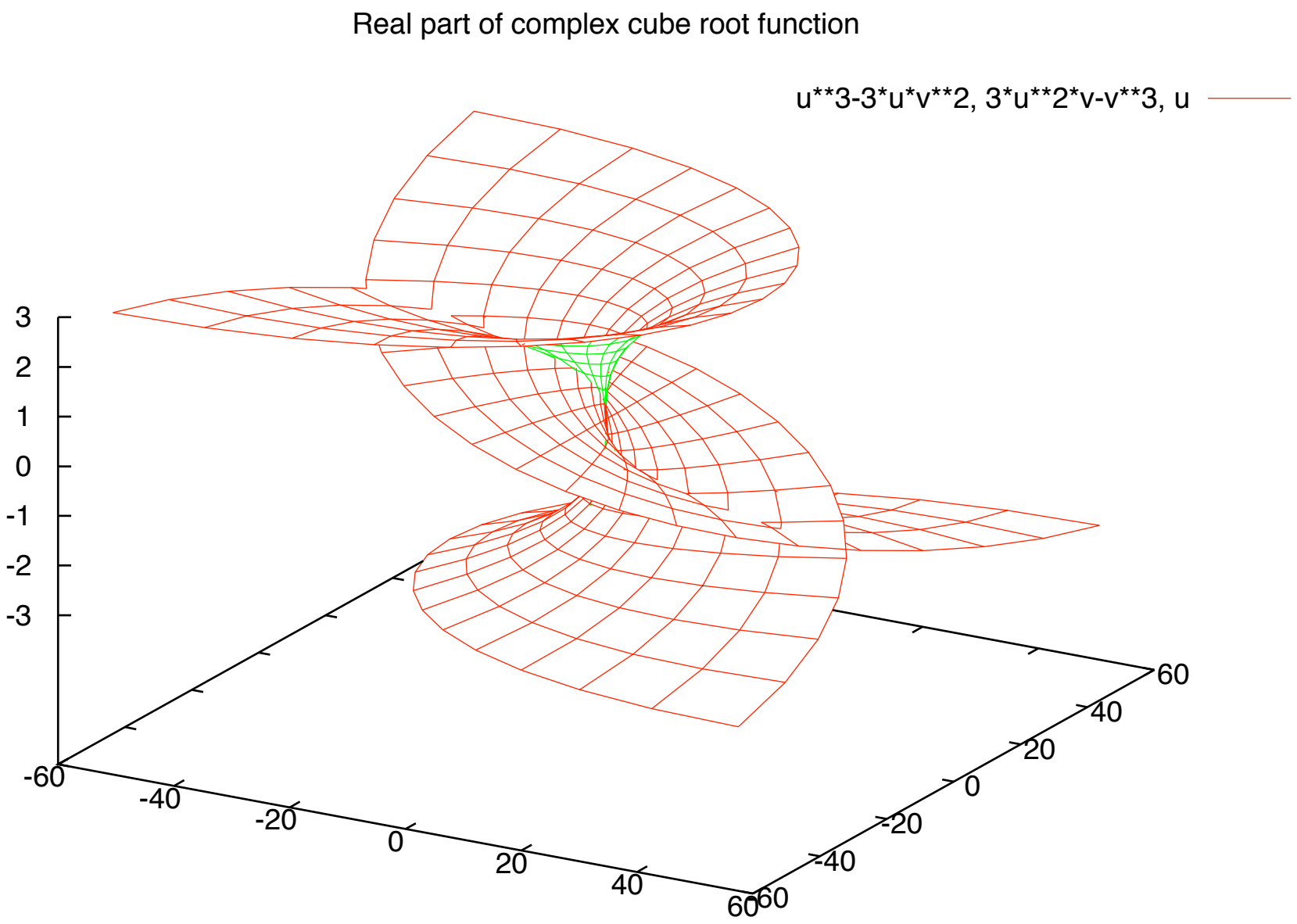
Real part of complex square root function (different view)



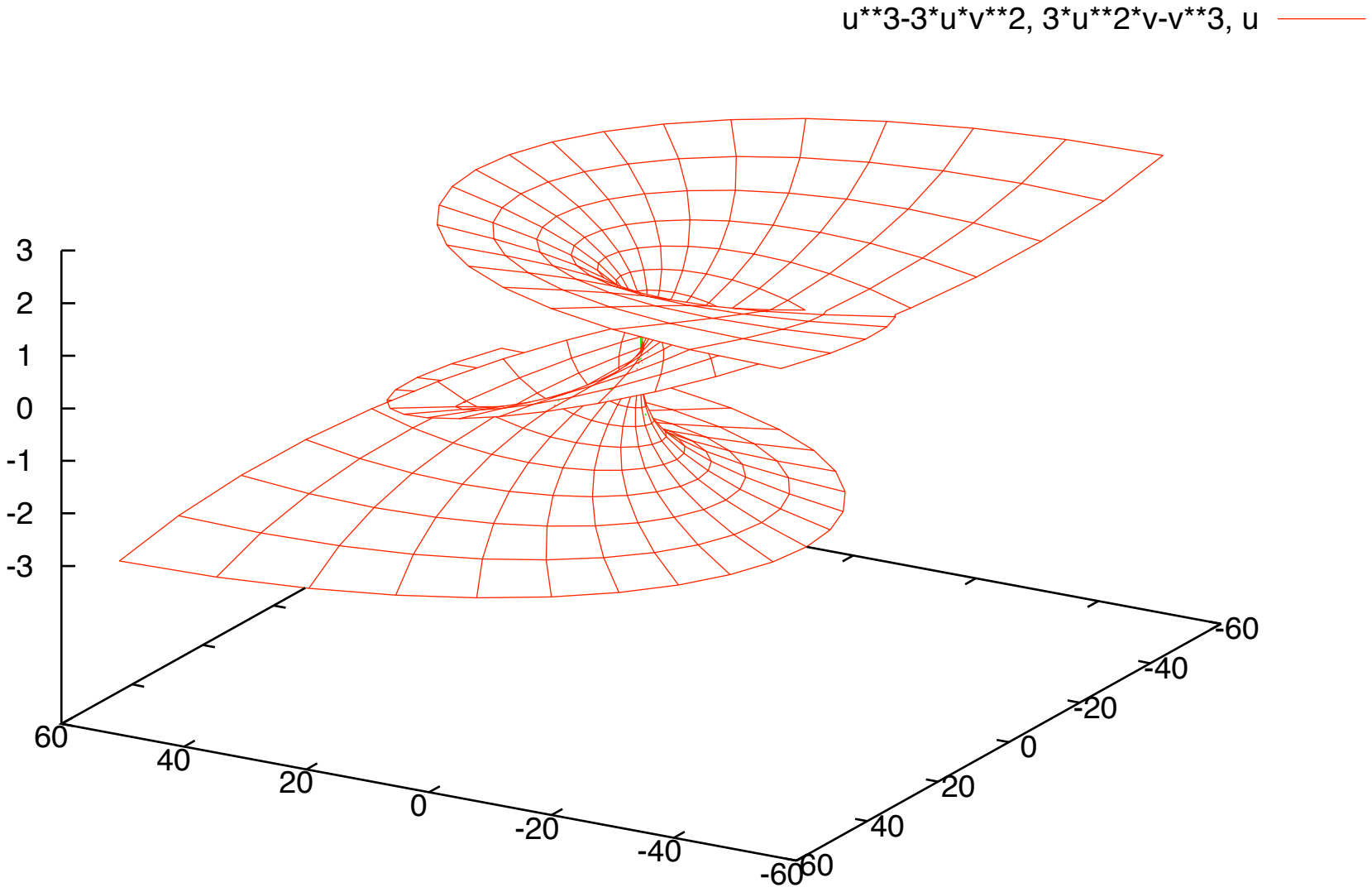

Imaginary part of complex square root function
u**2-v**2, 2*u*v, v
3
2
1
0
-1
-2
-3
-20
-15
-10
-5
0
5
10
15
20
10
8
6
4
2
0
-2
-4
-6
-8
-10

Imaginary part of complex square root function (different view)

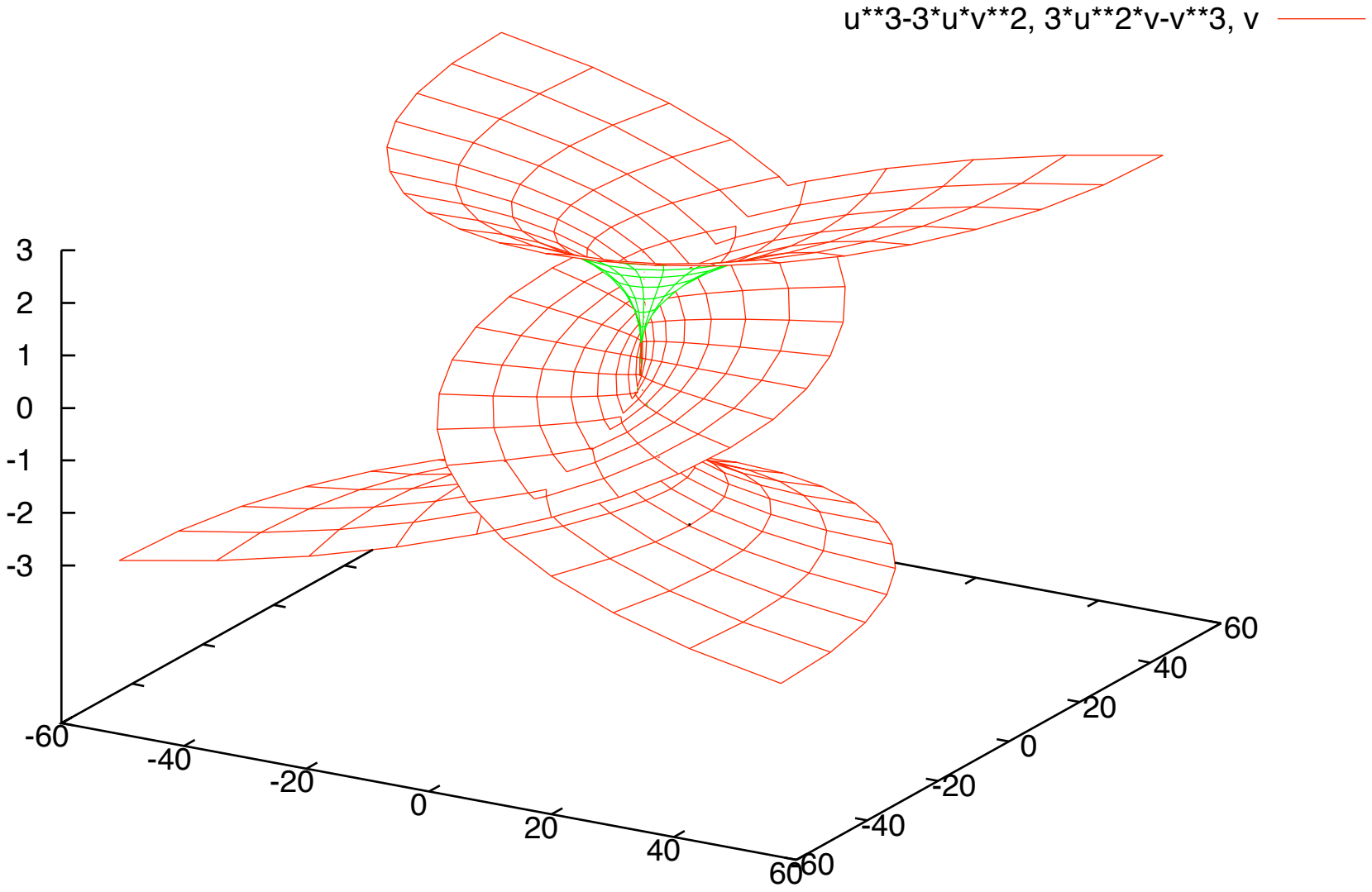



Real part of complex cube root function
u**3-3*u*v**2, 3*u**2*v-v**3, u
3
2
1
0
-1
-2
-3
-60
-40
-20
0
20
40
60
60
40
20
0
-20
-40
-60

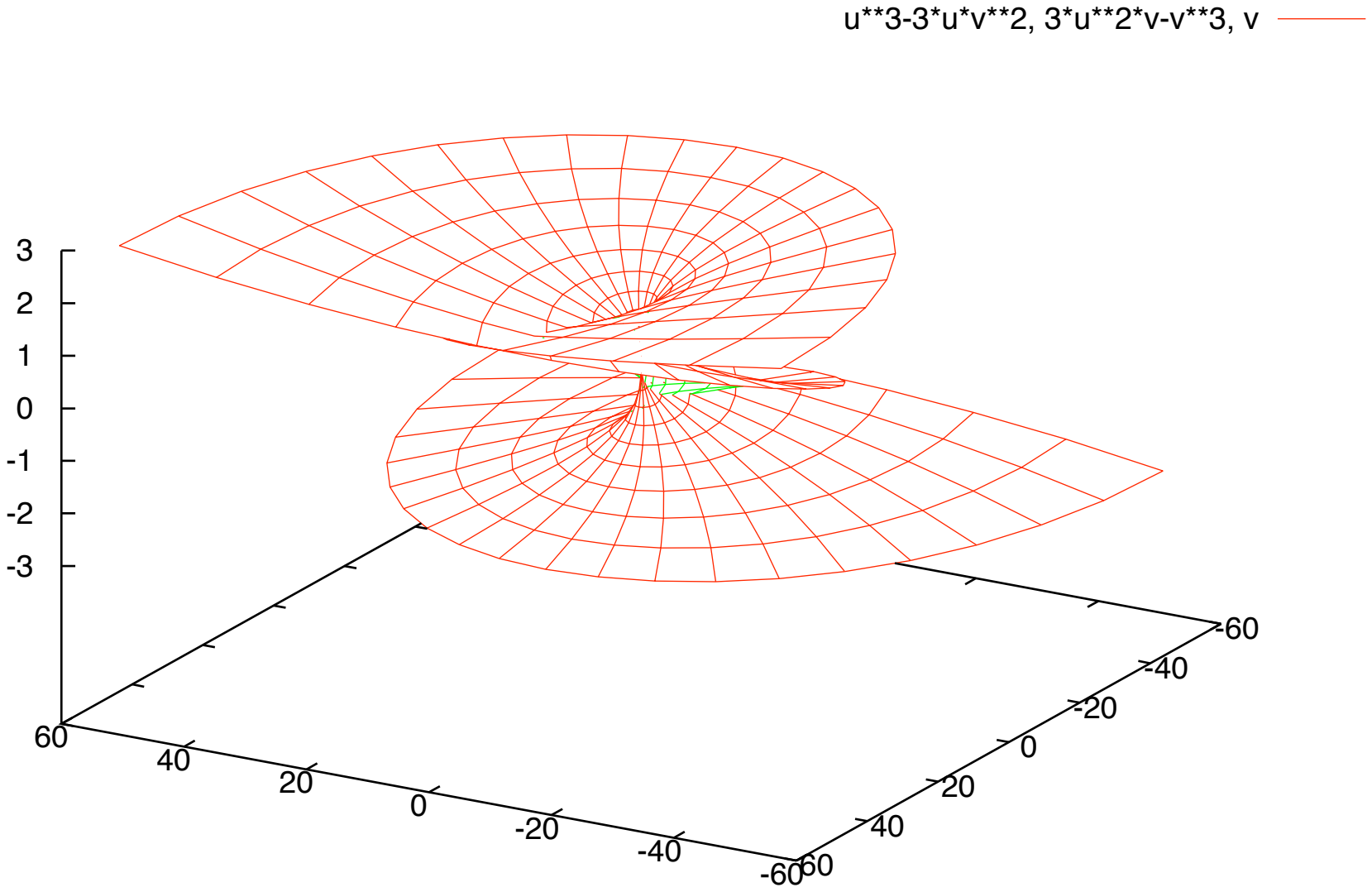
Real part of complex cube root function (different view)

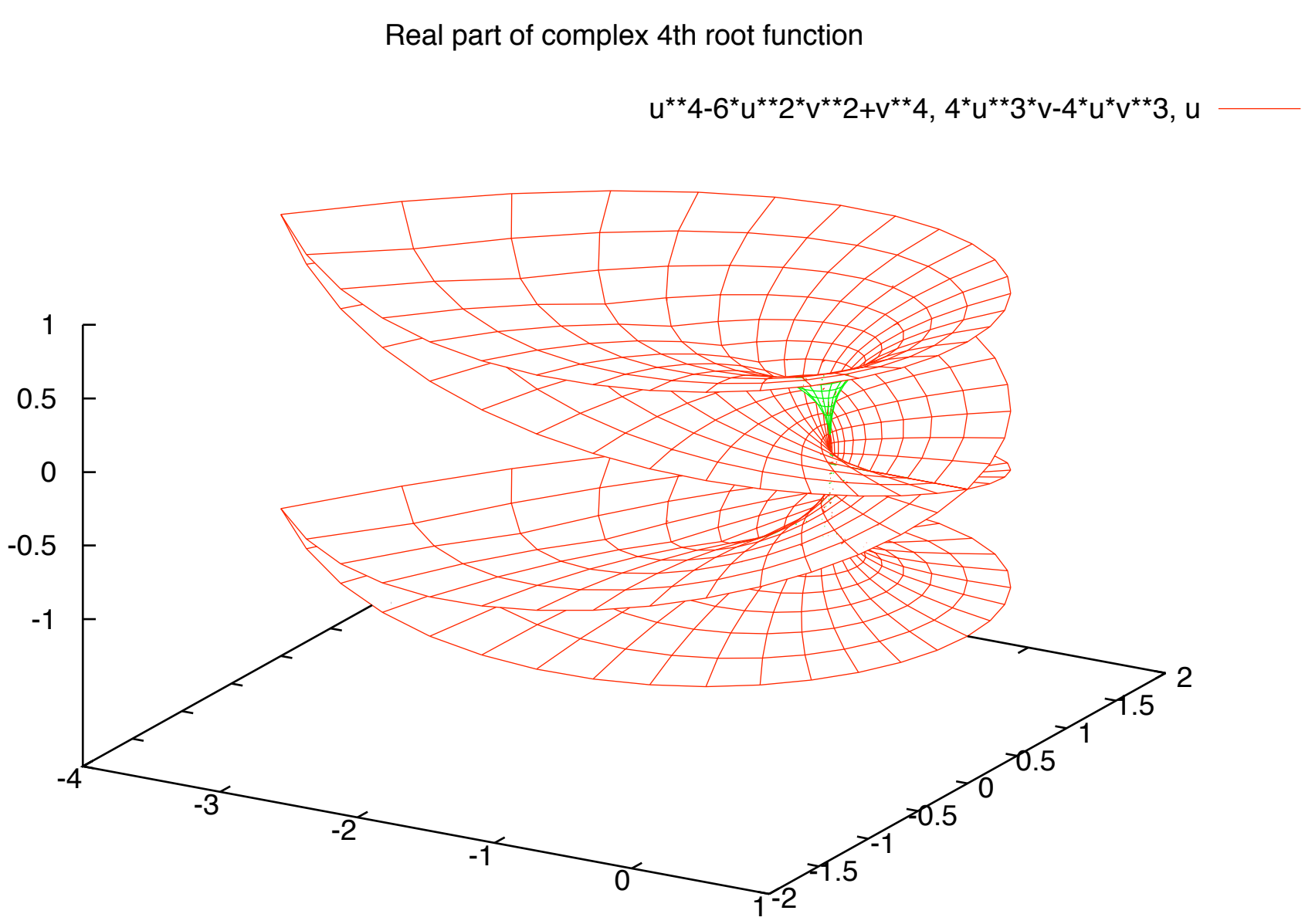


Imaginary part of complex cube root function

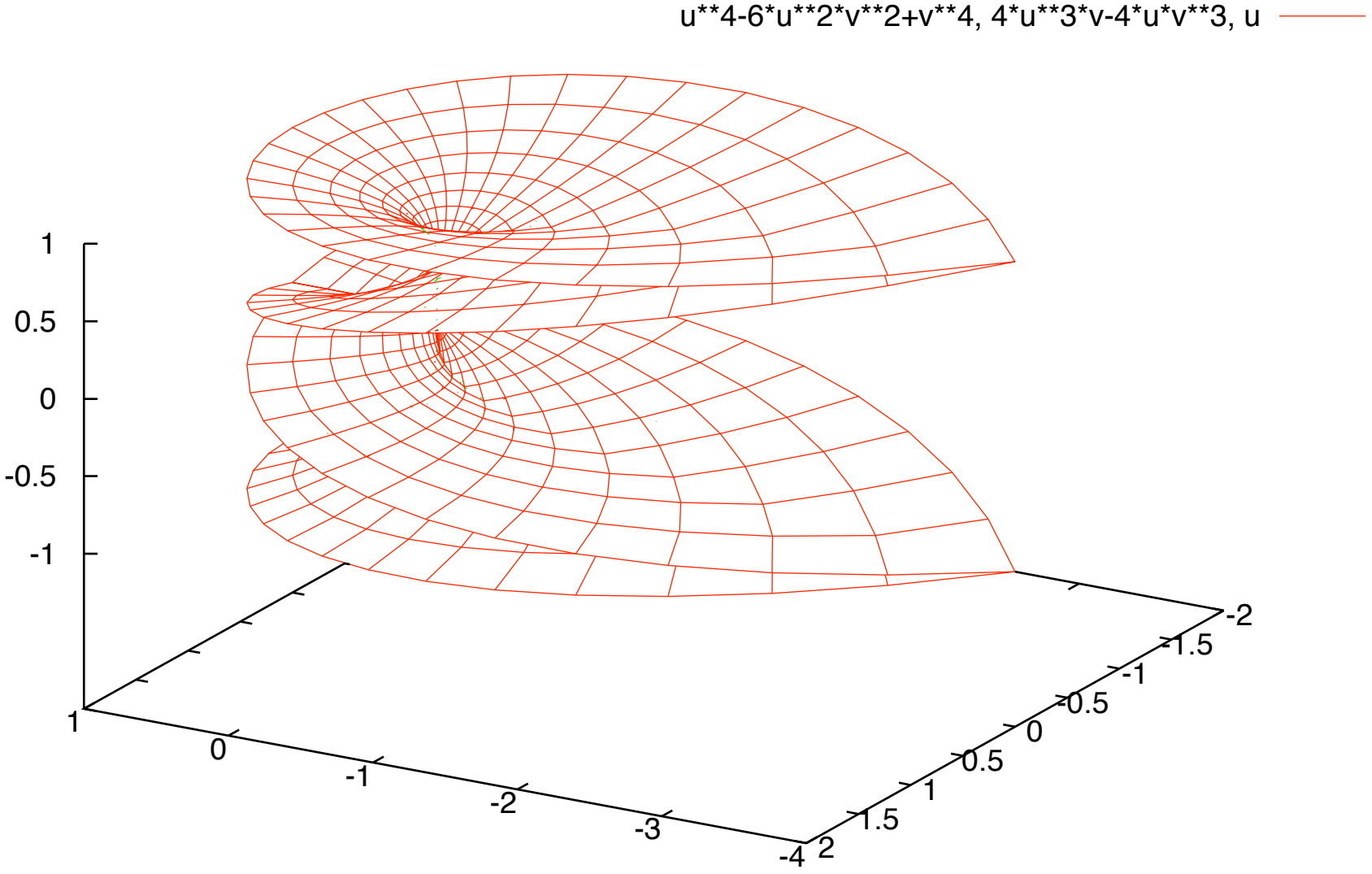


Imaginary part of complex cube root function (different view)

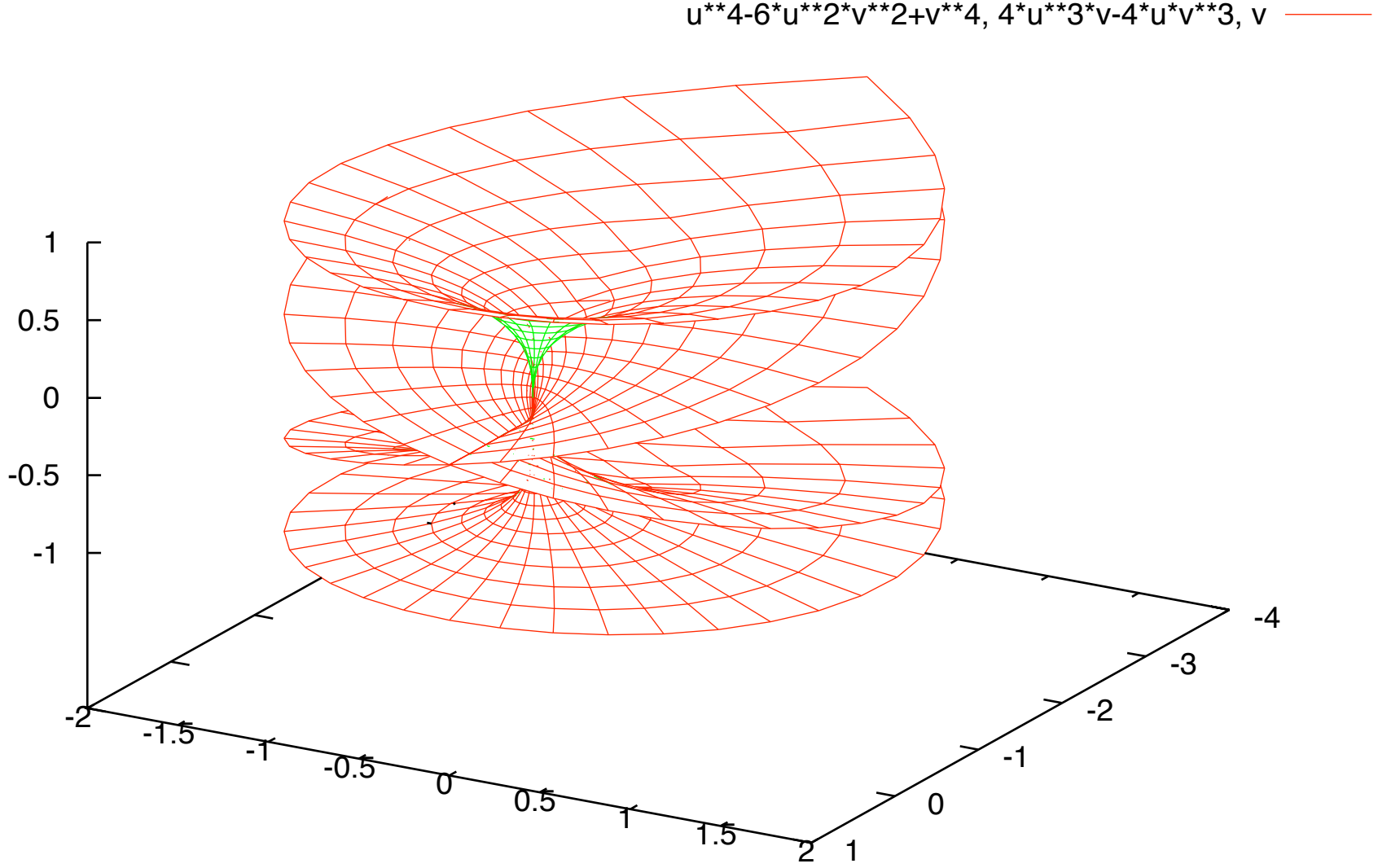


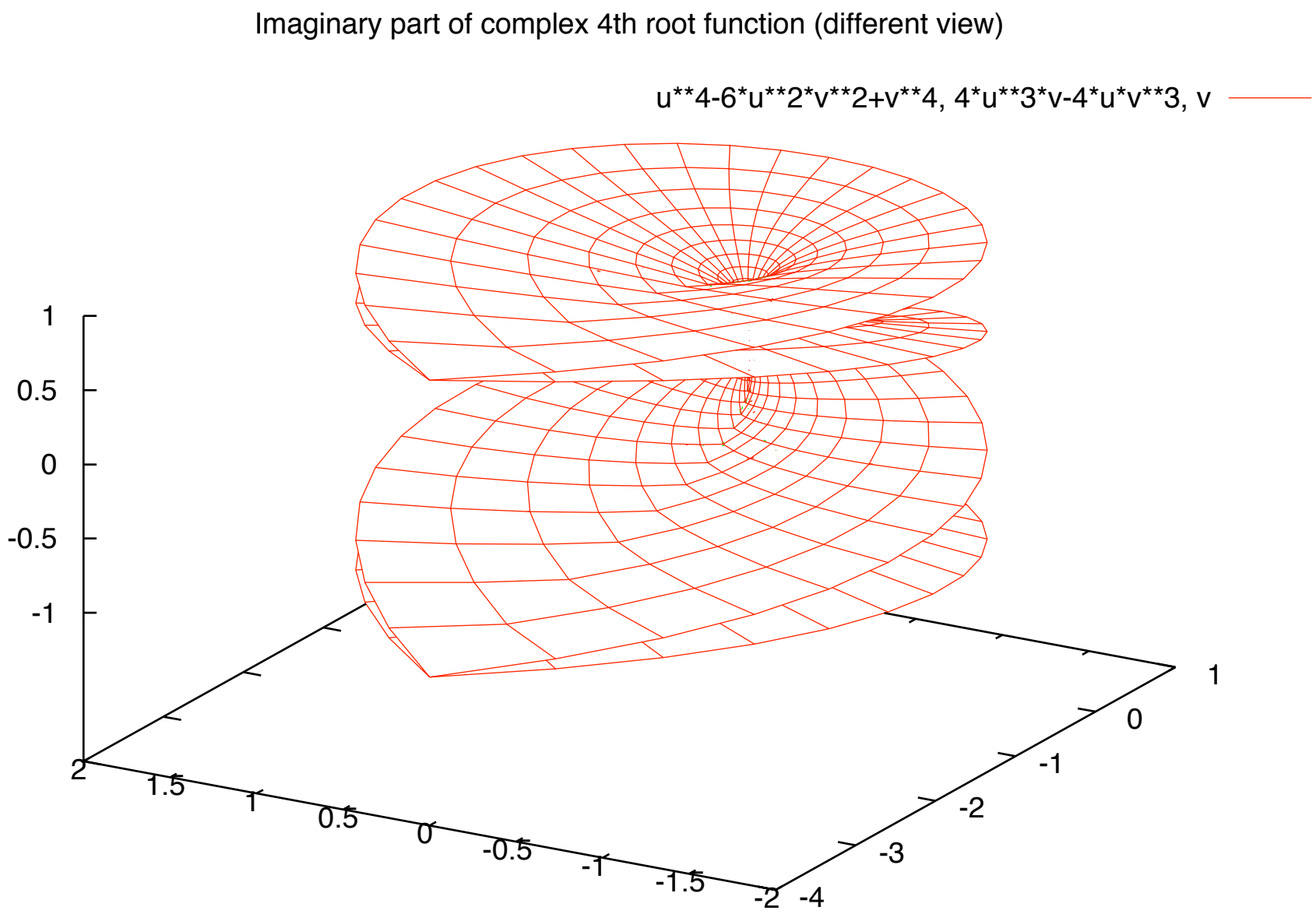

Real part of complex 4th root function
u**4-6*u**2*v**2+v**4, 4*u**3*v-4*u*v**3, u
1
0.5
0
-0.5
-1
-4
-3
-2
-1
0
1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

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

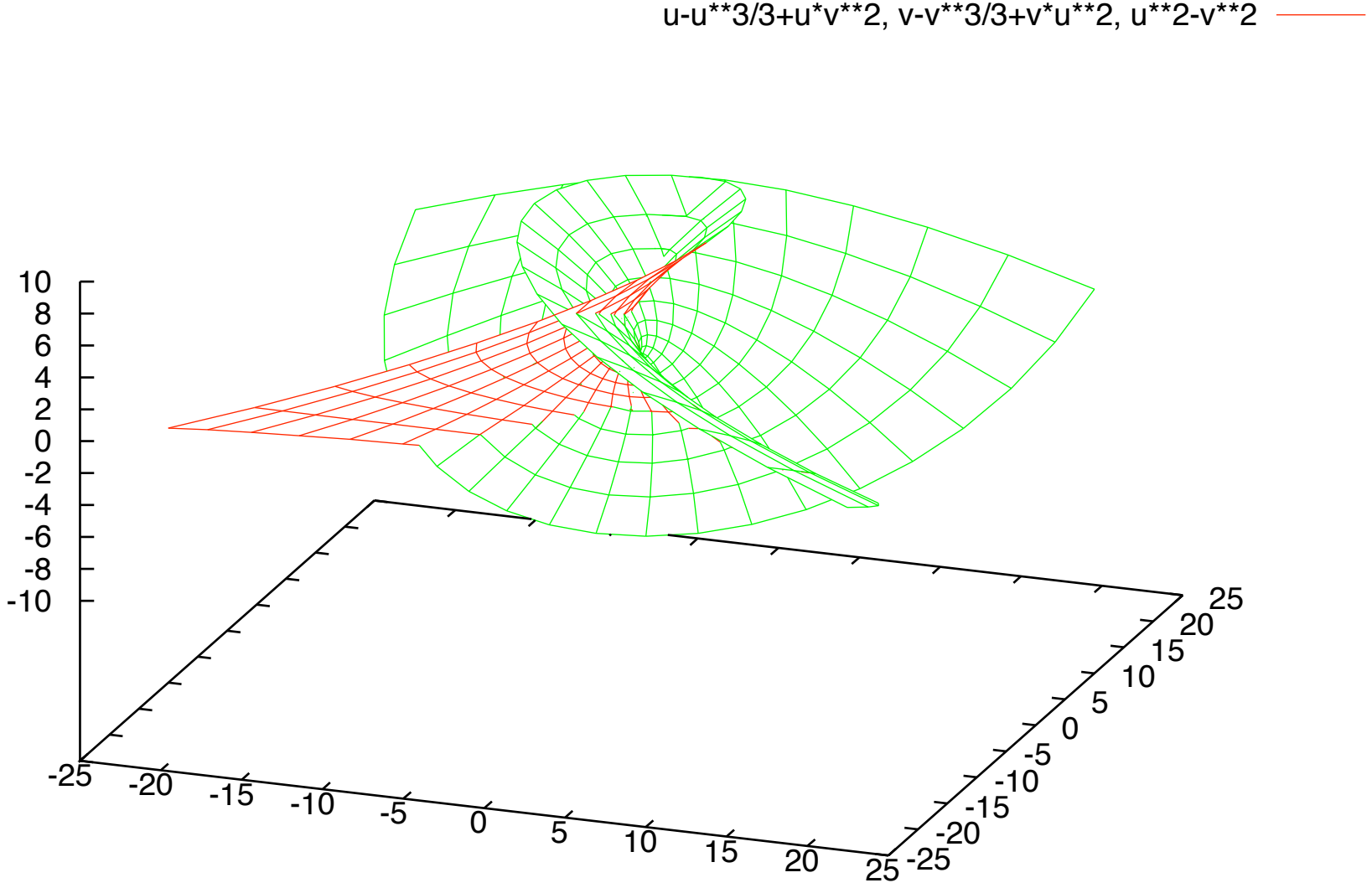


Imaginary part of complex 4th root function

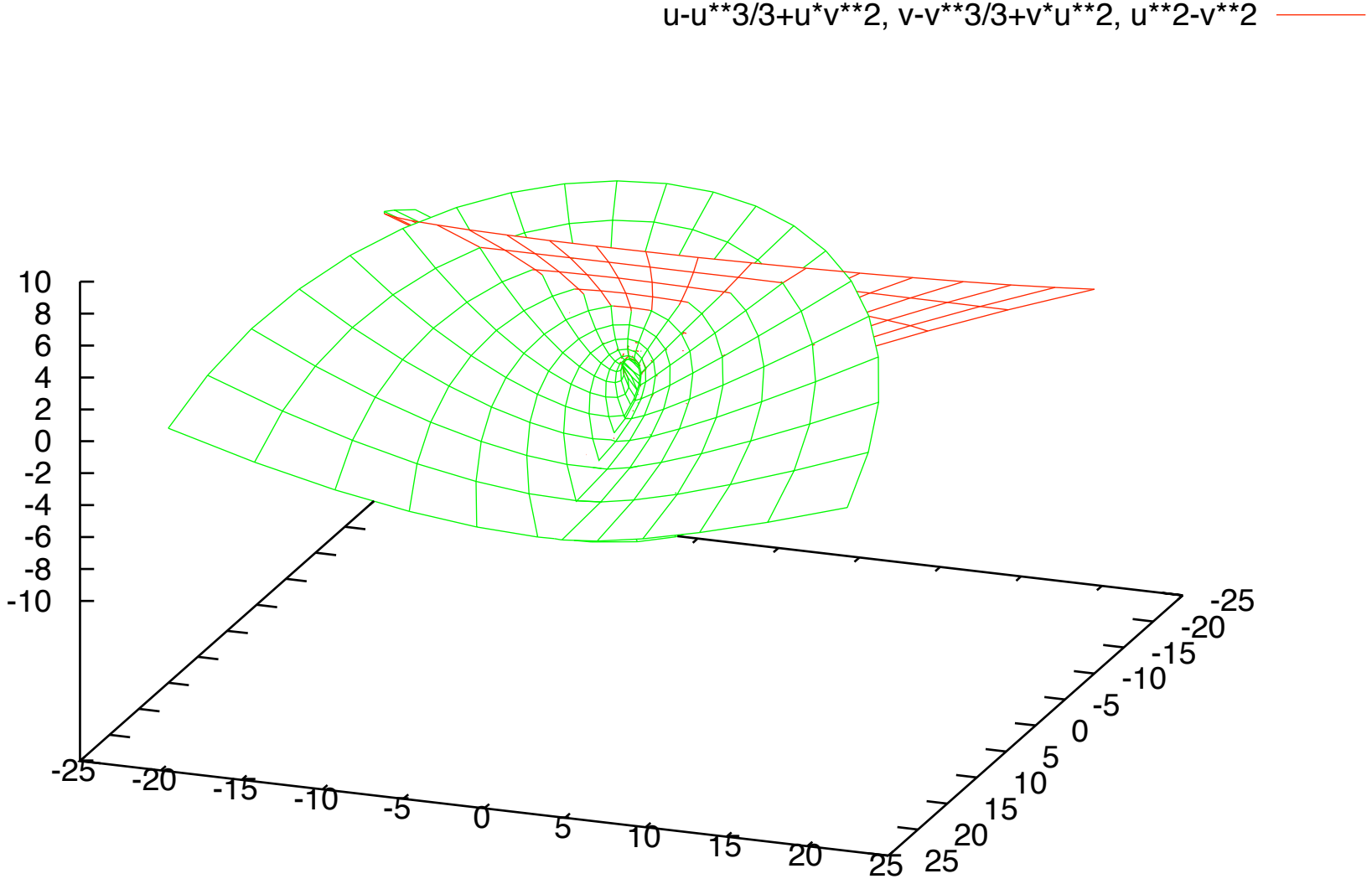


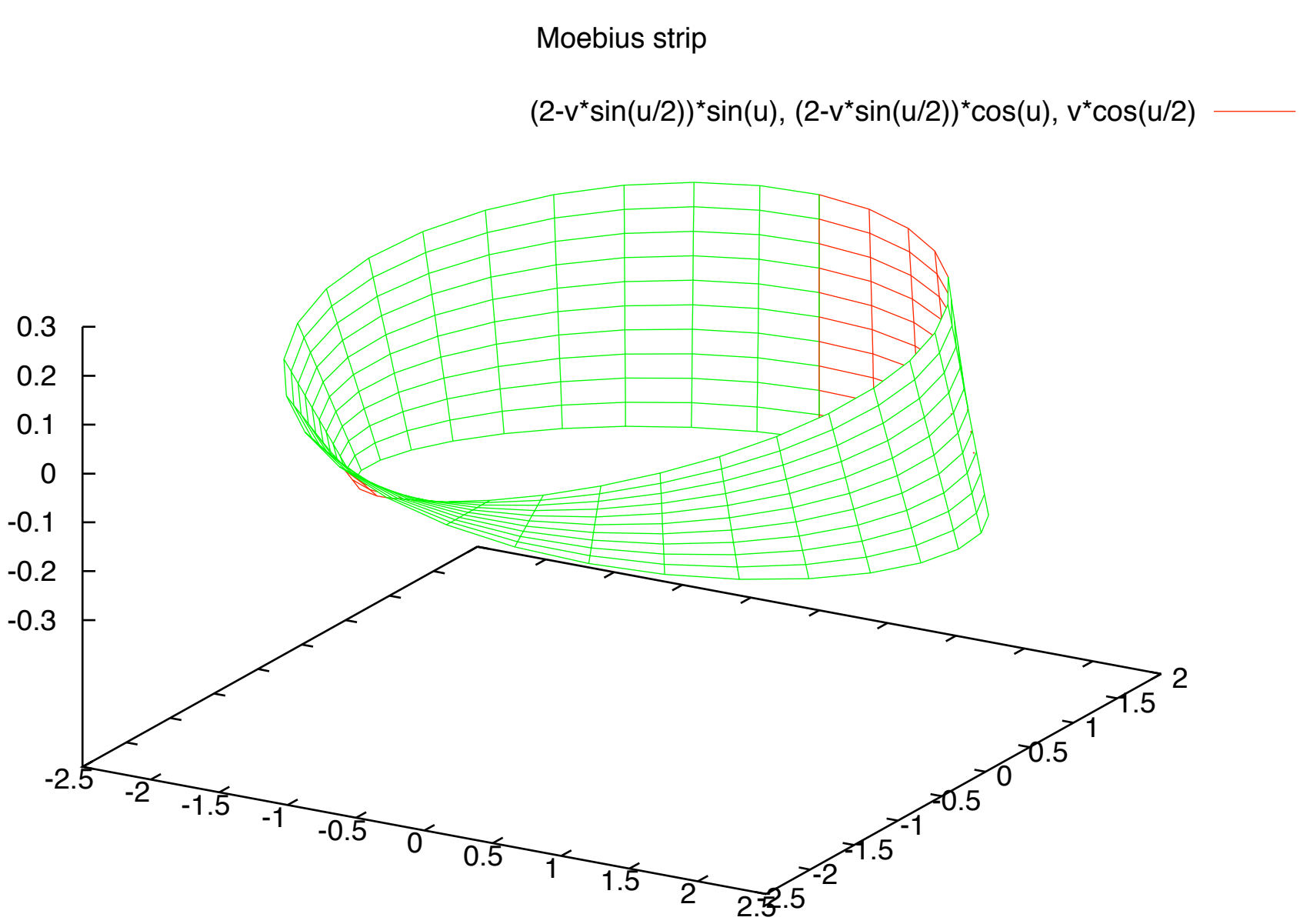

Imaginary part of complex 4th root function (different view)
u**4-6*u**2*v**2+v**4, 4*u**3*v-4*u*v**3, v
1
0.5
0
-0.5
-1
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-4
-3
-2
-1
0
1

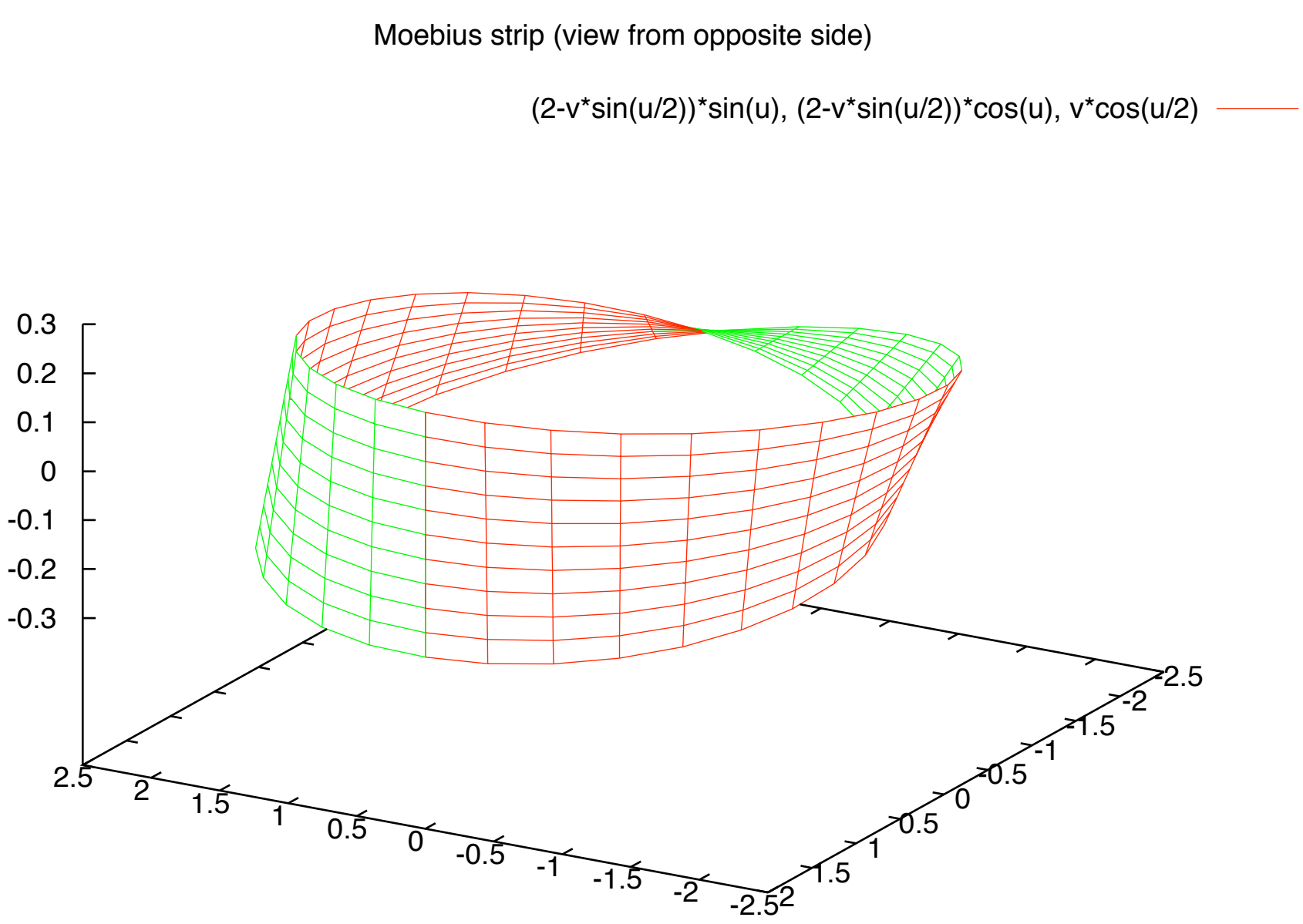
Enneper's surface

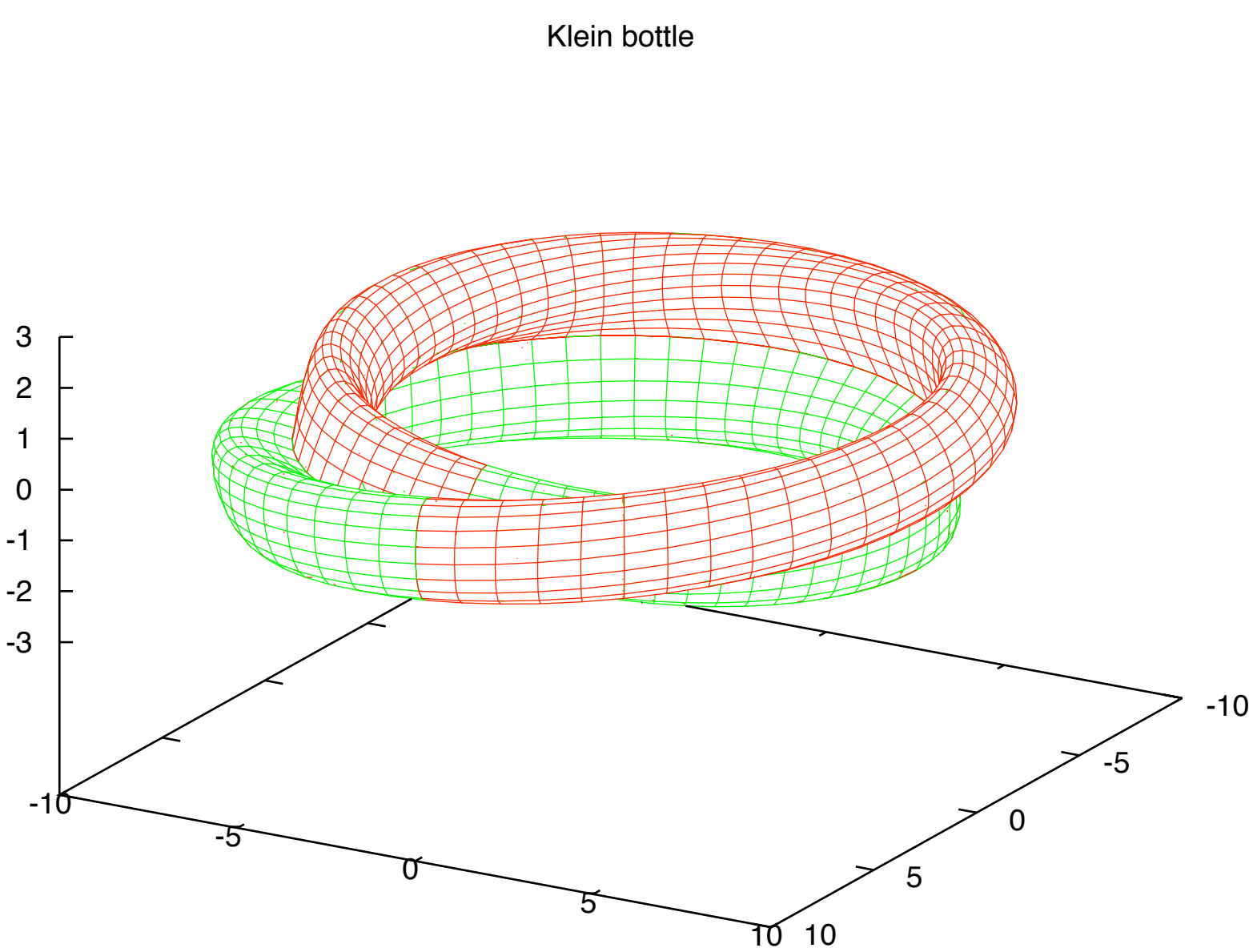


Enneper’s surface (different view)

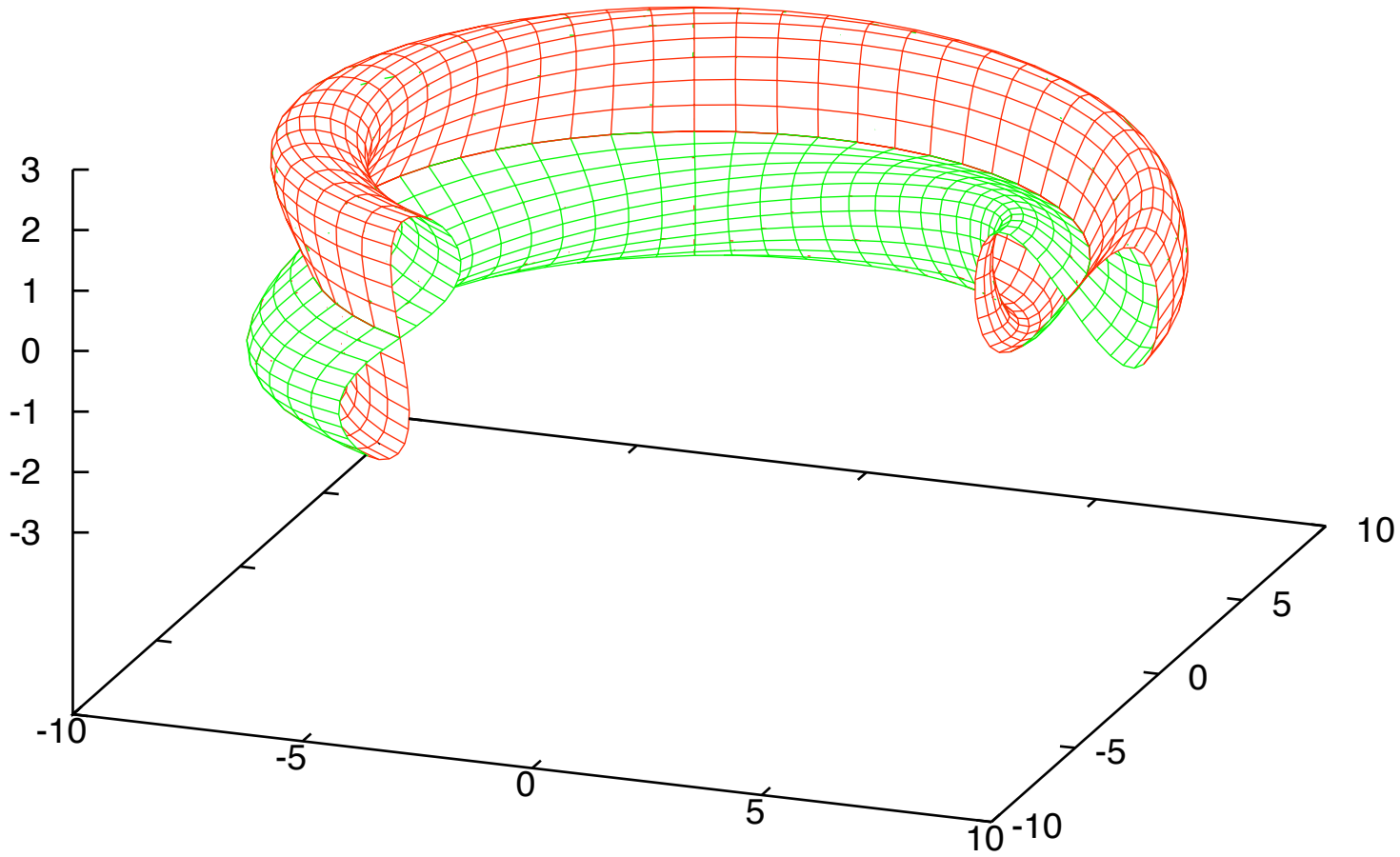



Moebius strip
(2-v*sin(u/2))*sin(u), (2-v*sin(u/2))*cos(u), v*cos(u/2)
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5

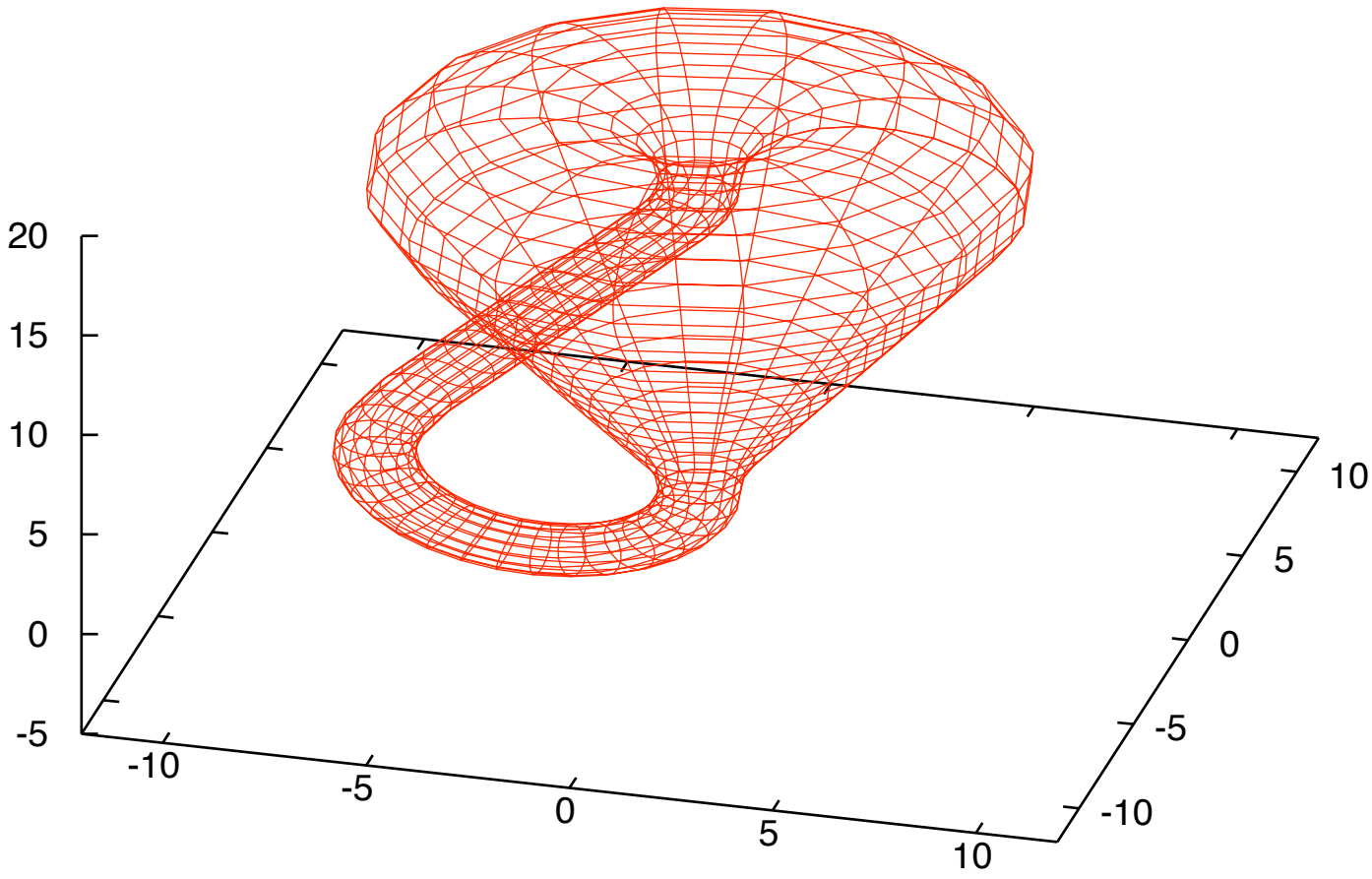

Moebius strip (view from opposite side)
(2-v*sin(u/2))*sin(u), (2-v*sin(u/2))*cos(u), v*cos(u/2)
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
2.5


Klein bottle
3
2
1
0
-1
-2
-3
-10
-5
0
5
10
-10
-5
0
5
10

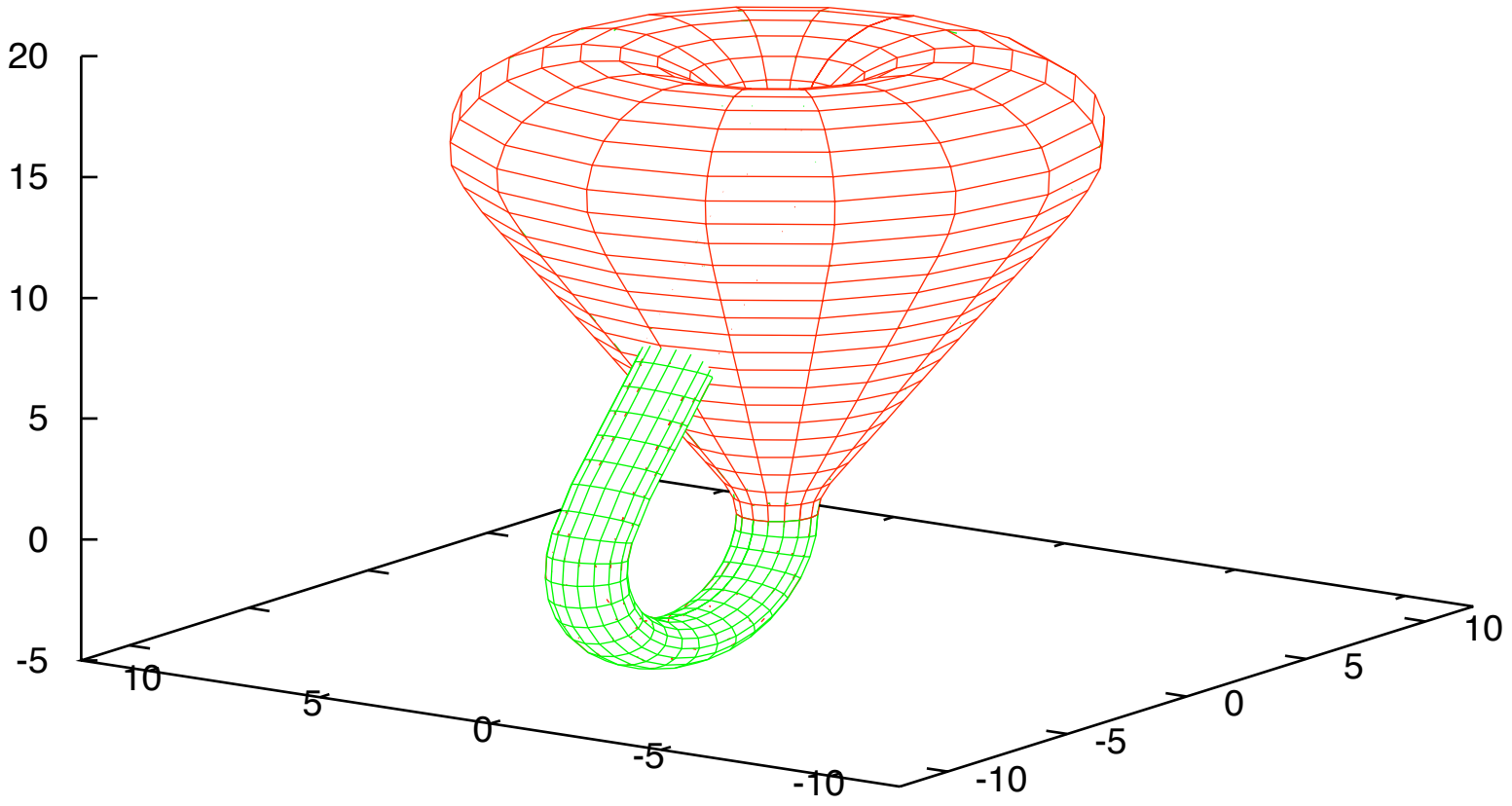
Klein bottle with look at the ’inside’

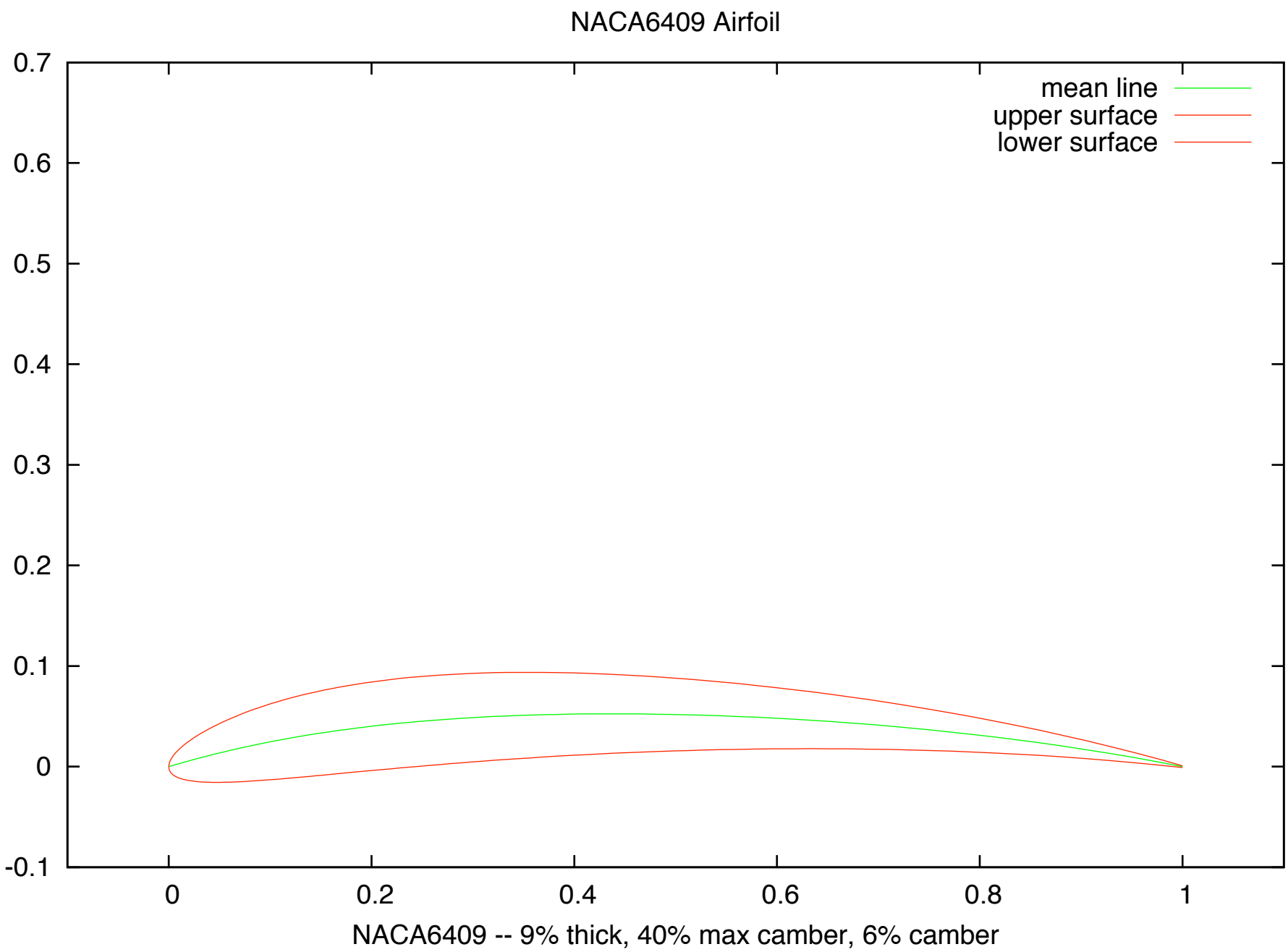


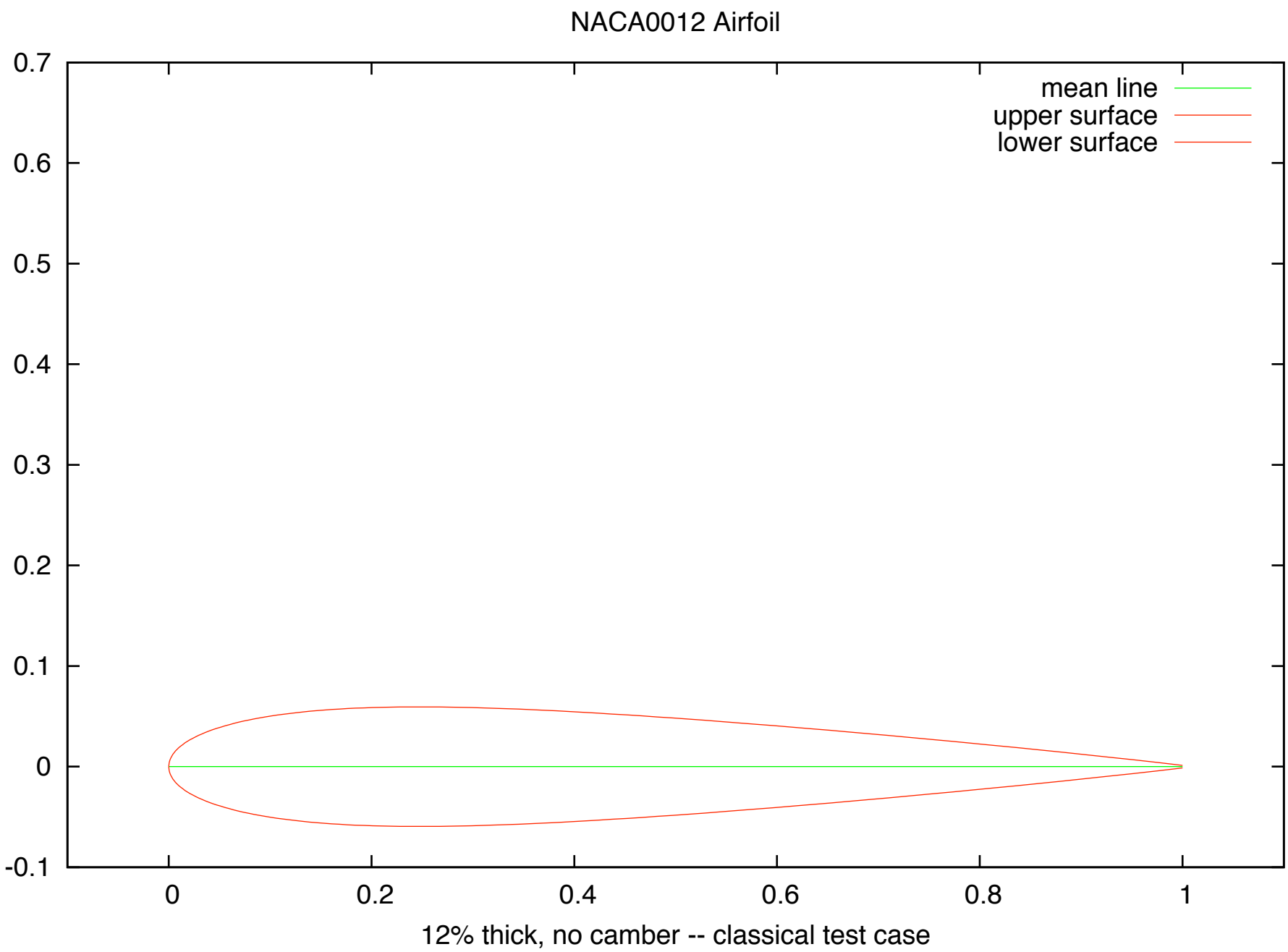
Klein bottle, glassblowers' version (look-through)

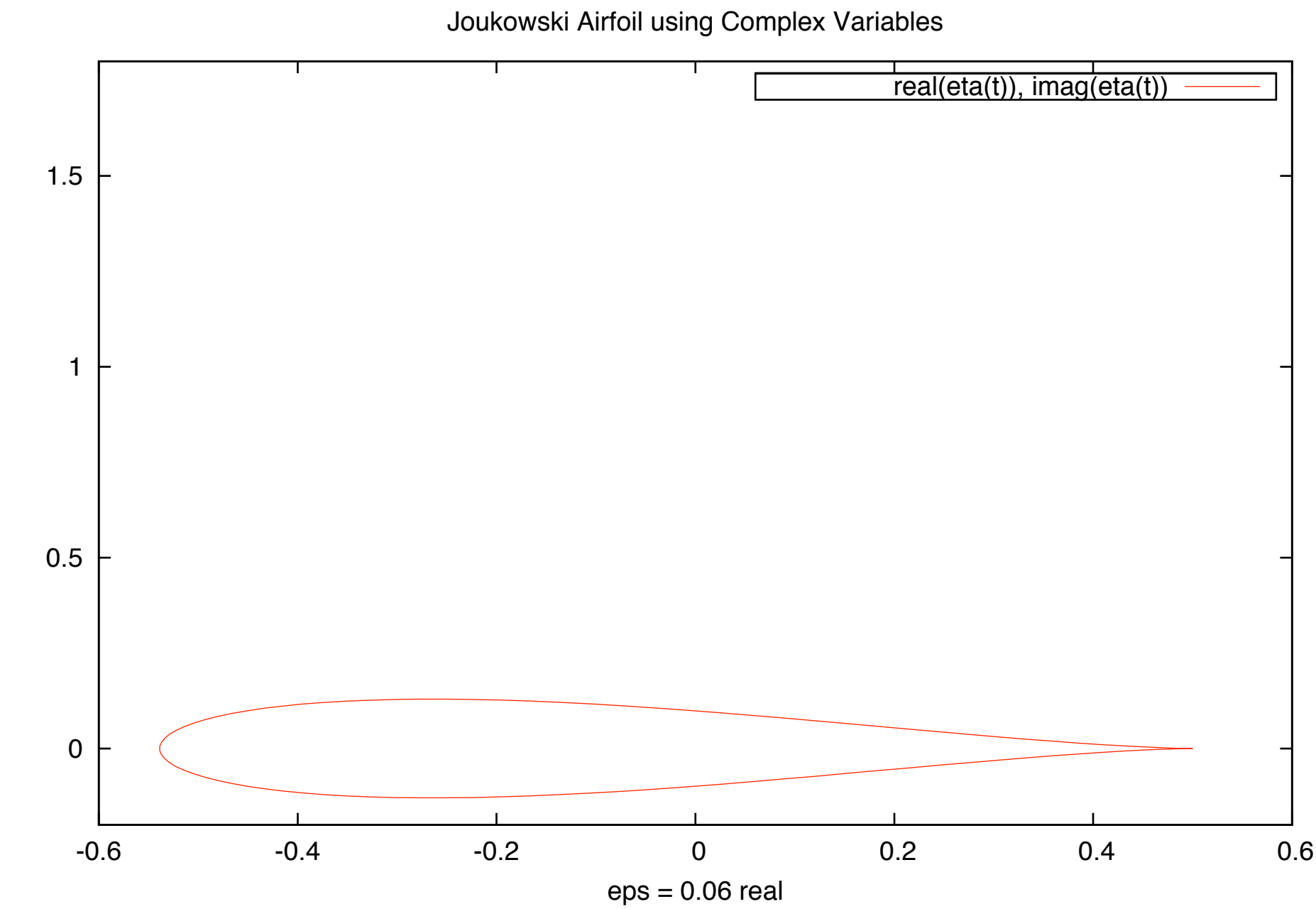


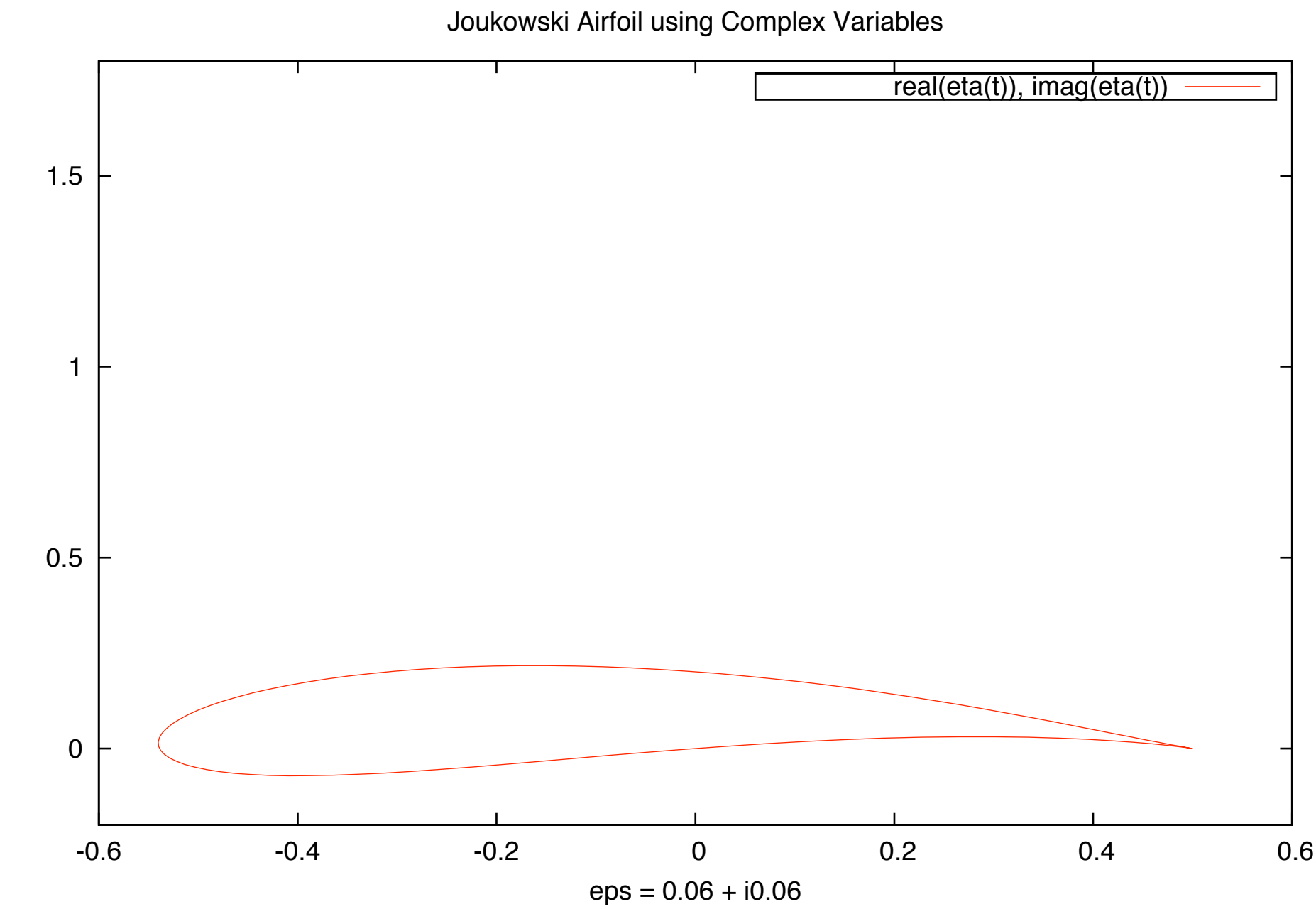
Klein bottle, glassblowers' version (solid)

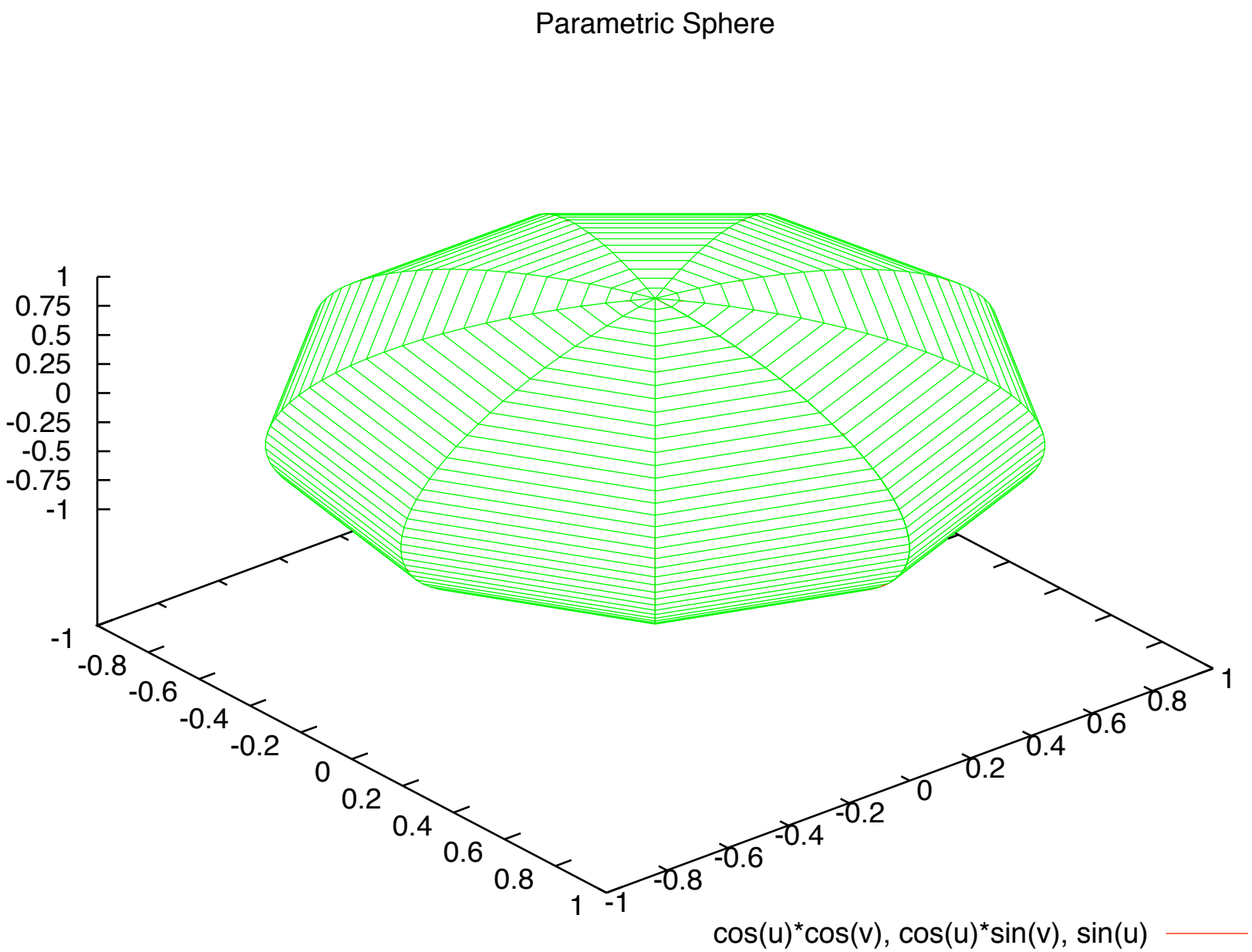


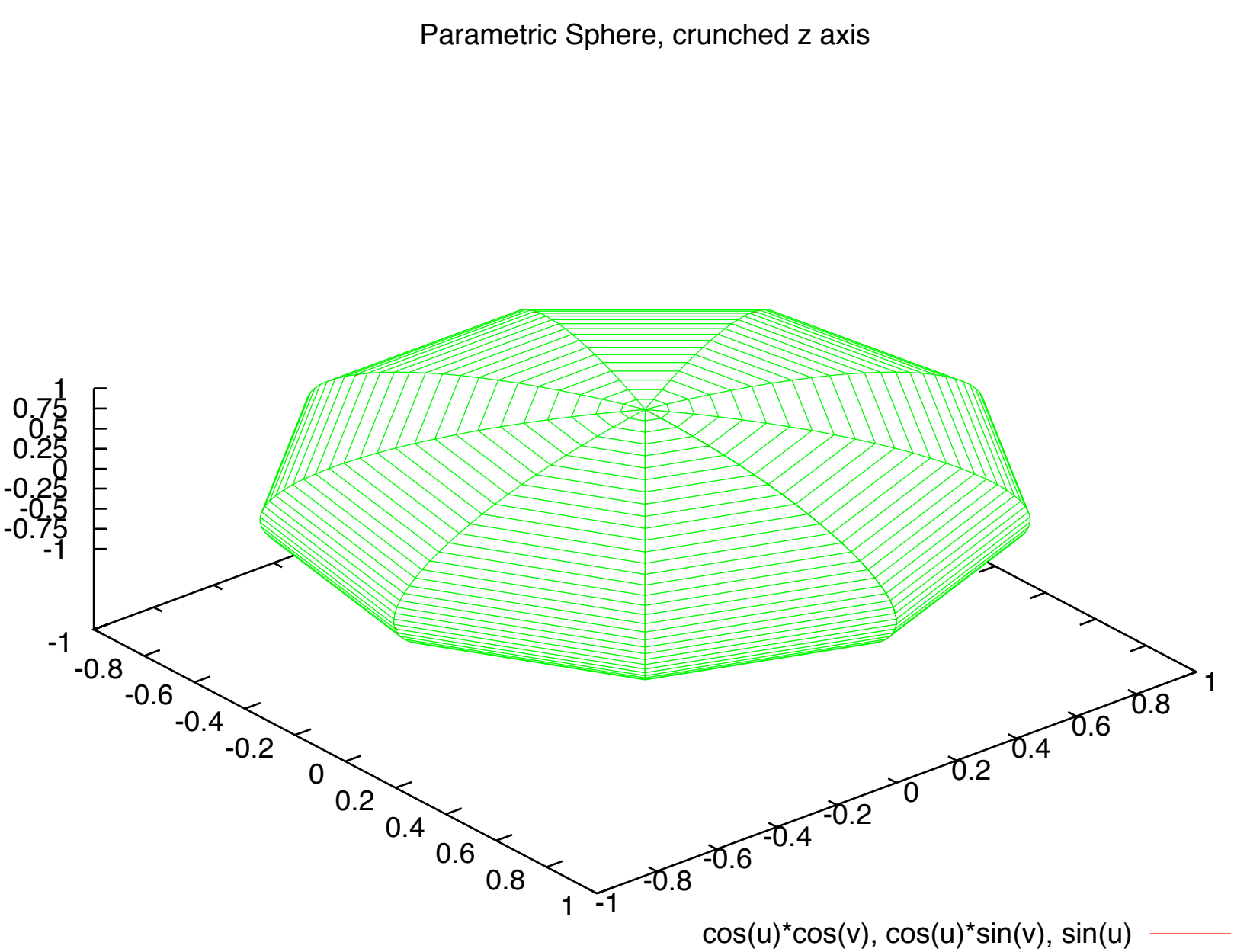

NACA6409 Airfoil
mean line
upper surface
lower surface
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
0
0.2
0.4
0.6
0.8
1
NACA6409 -- 9% thick, 40% max camber, 6% camber

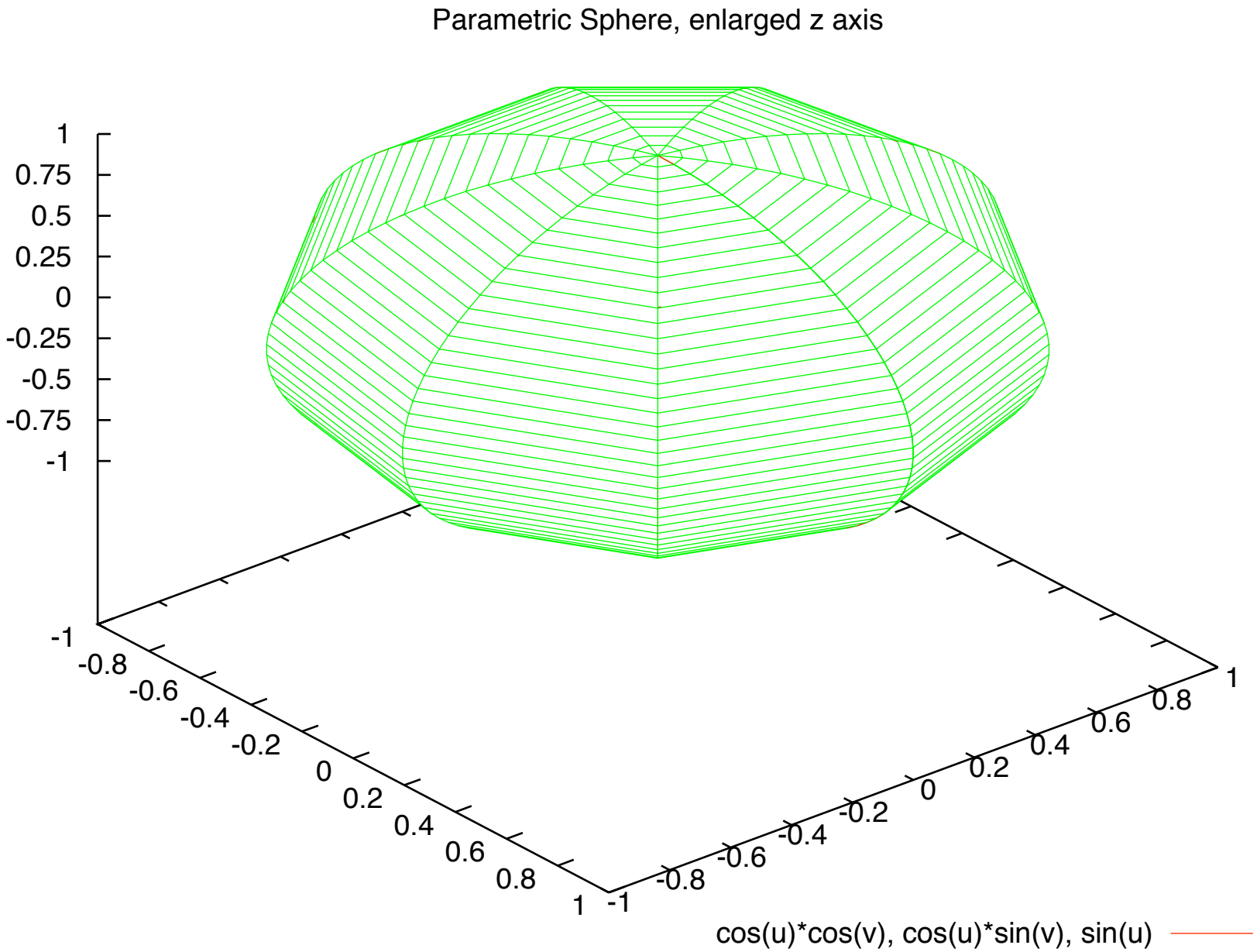

NACA0012 Airfoil
mean line
upper surface
lower surface
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
0
0.2
0.4
0.6
0.8
1
12% thick, no camber -- classical test case

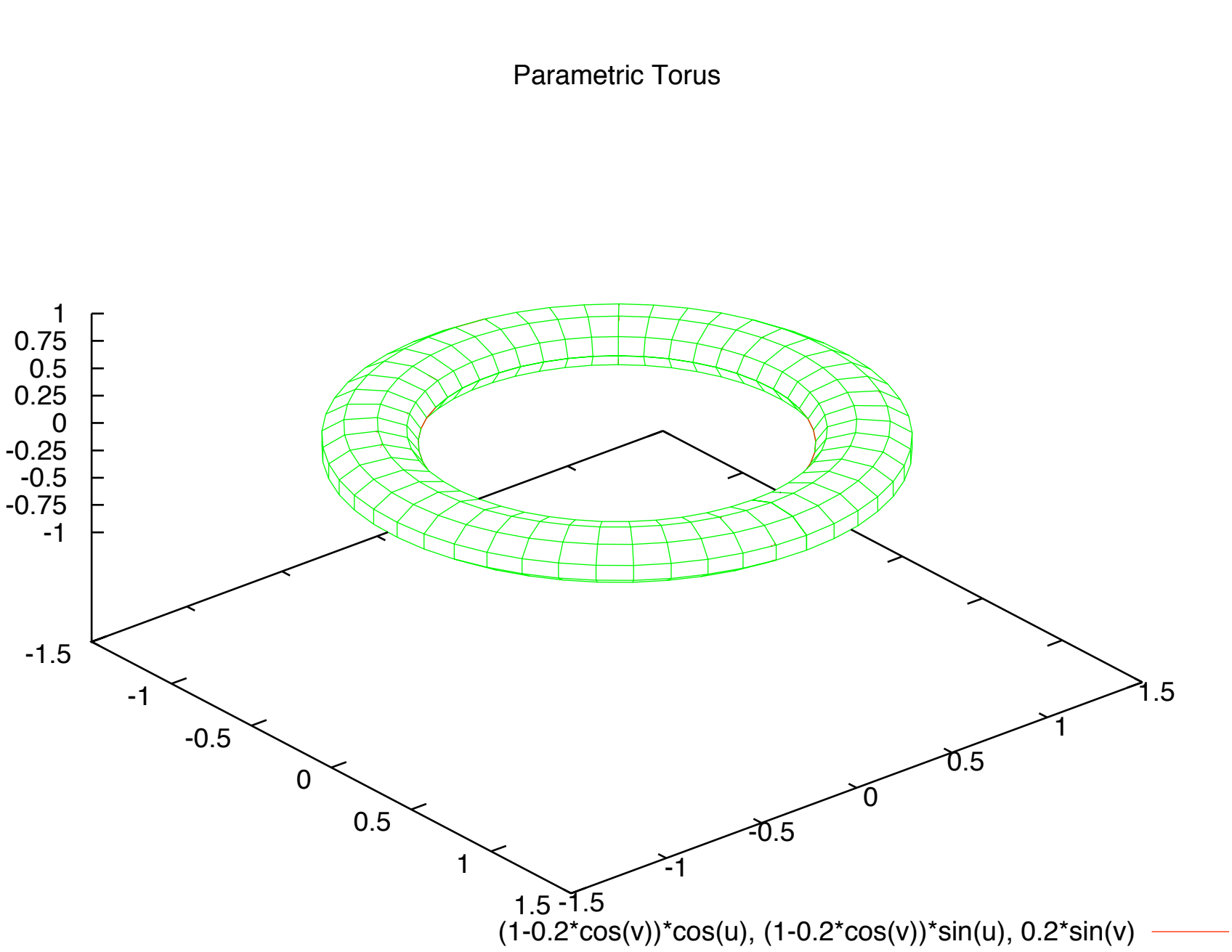

Joukowski Airfoil using Complex Variables
real(eta(t)), imag(eta(t))
1.5
1
0.5
0
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
eps = 0.06 real
Wed Feb 04 12:18:51 2009


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
Wed Feb 04 12:18:51 2009

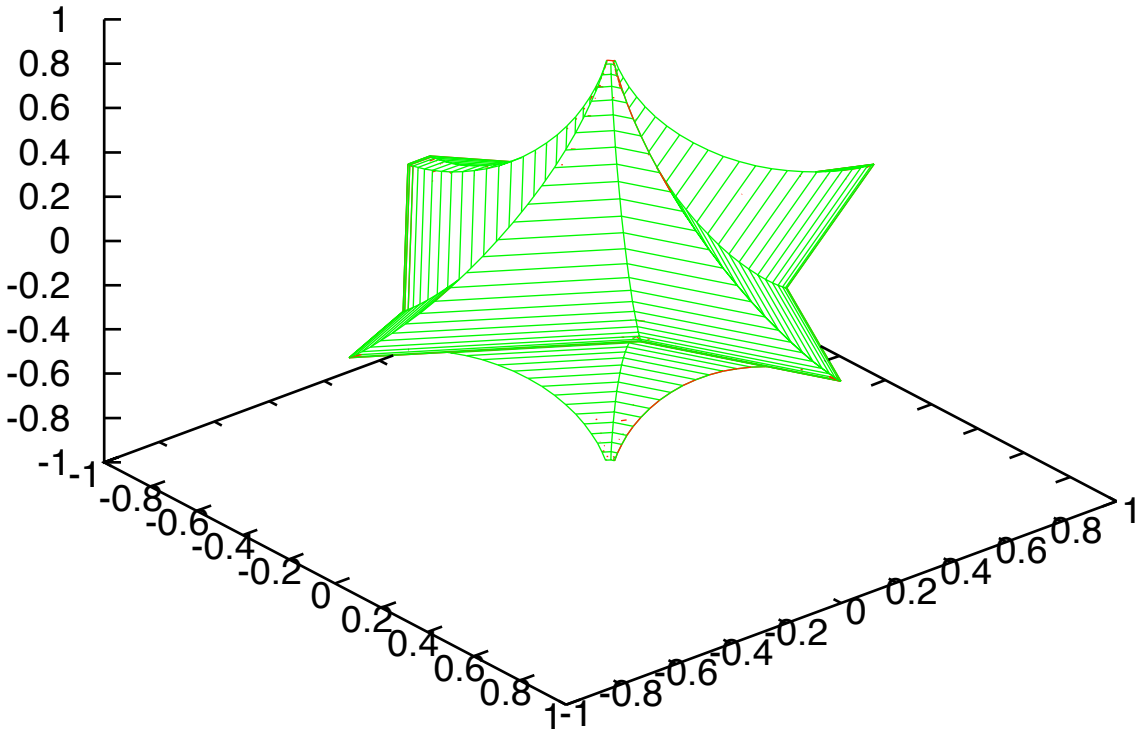

Parametric Sphere
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
cos(u)*cos(v), cos(u)*sin(v), sin(u)


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

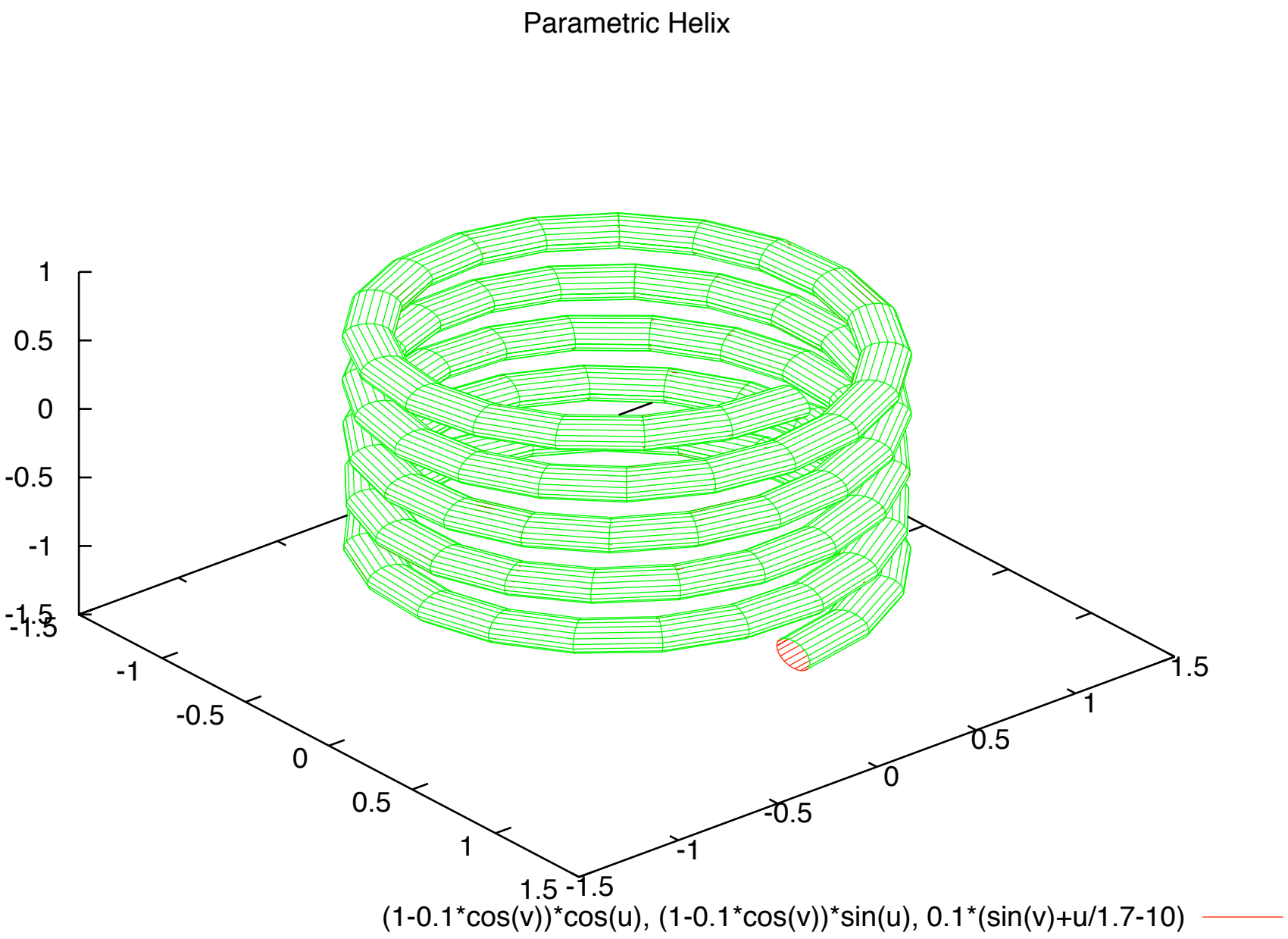

Parametric Sphere, enlarged z axis
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
cos(u)*cos(v), cos(u)*sin(v), sin(u)

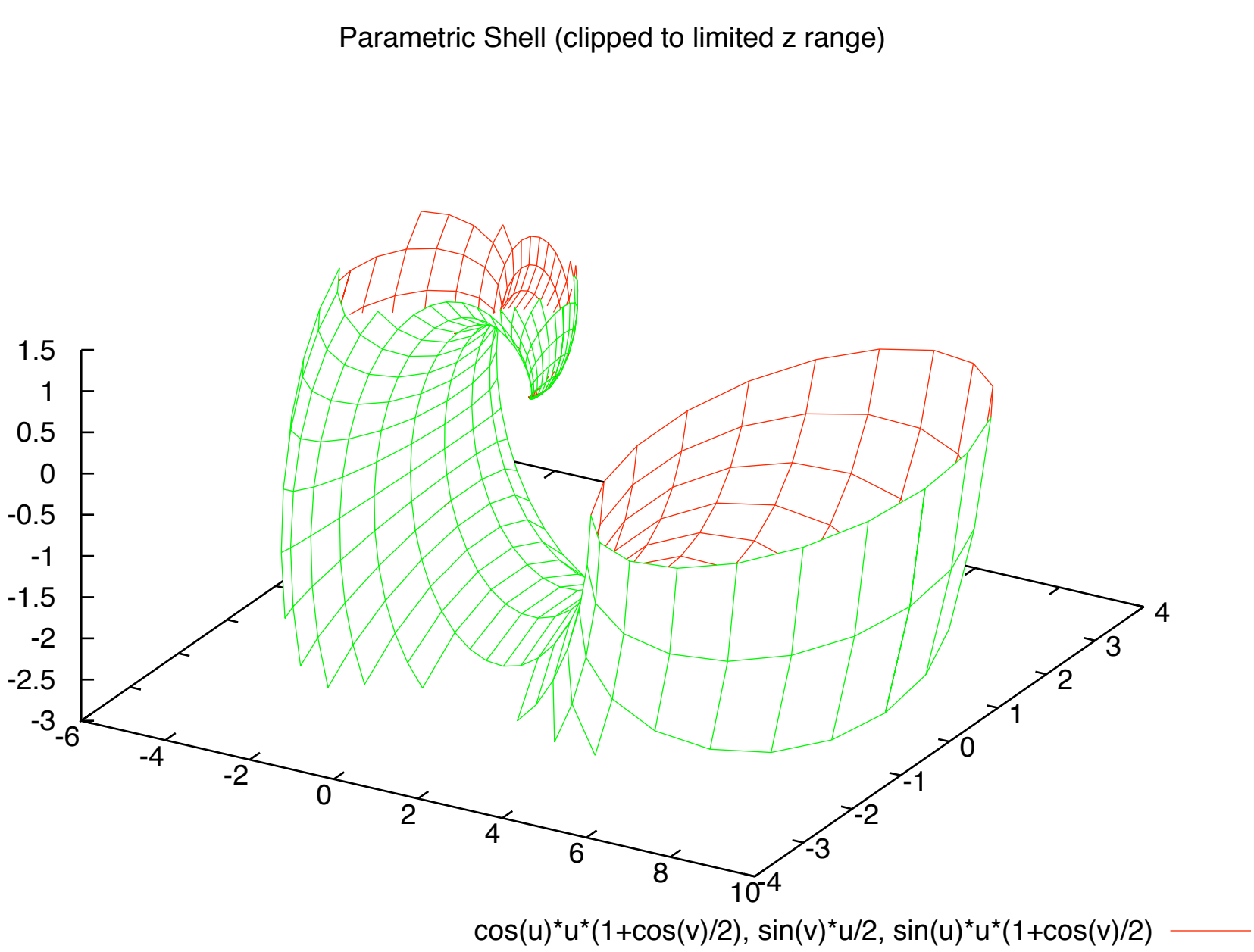

Parametric Torus
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1
-1.5
-1
-0.5
0
0.5
1
1.5
-1.5
-1
-0.5
0
0.5
1
1.5
(1-0.2*cos(v))*cos(u), (1-0.2*cos(v))*sin(u), 0.2*sin(v)

Parametric Hexagon

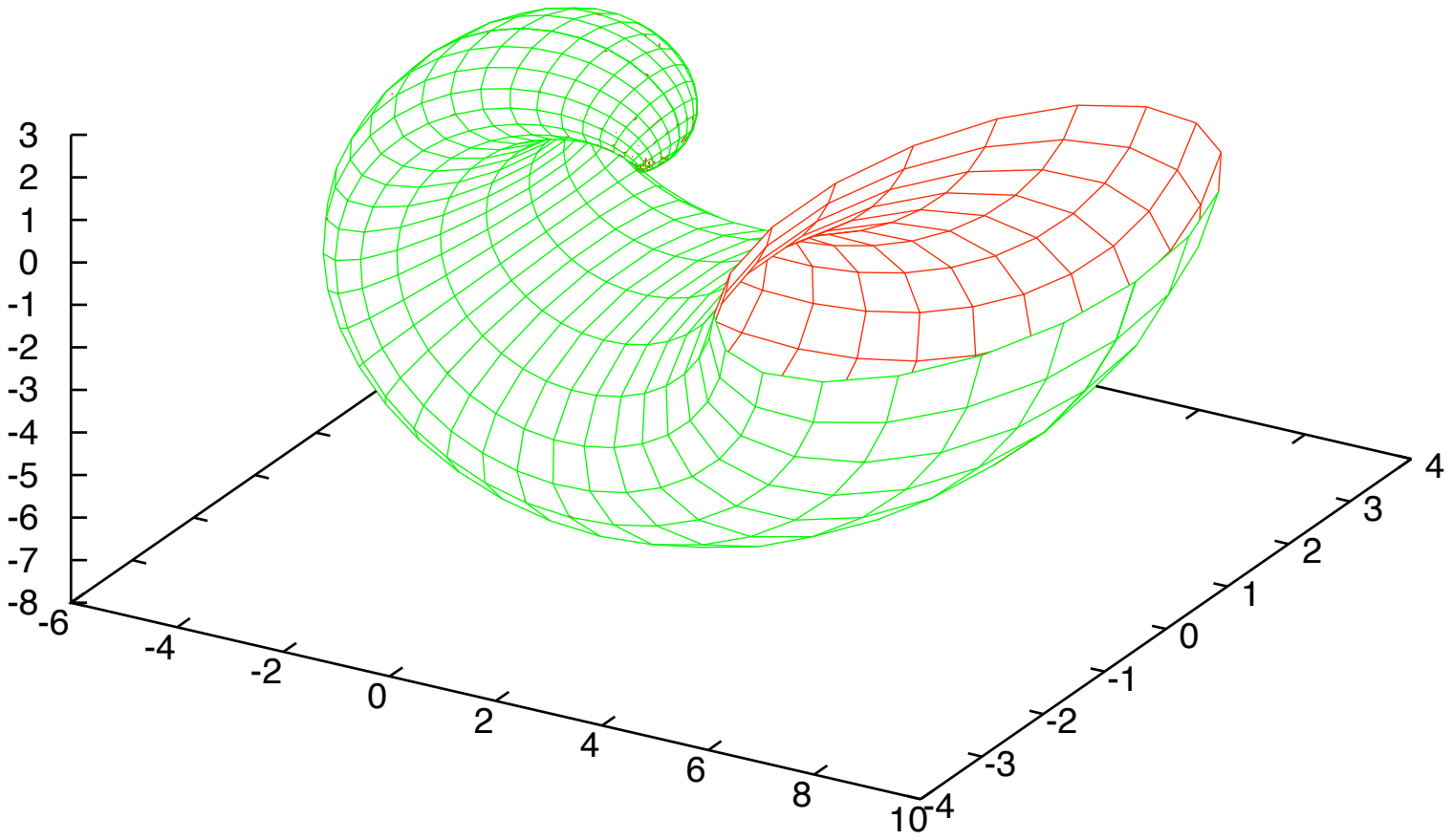


cos(v)**3*cos(u)**3, sin(v)**3*cos(u)**3, sin(u)**3

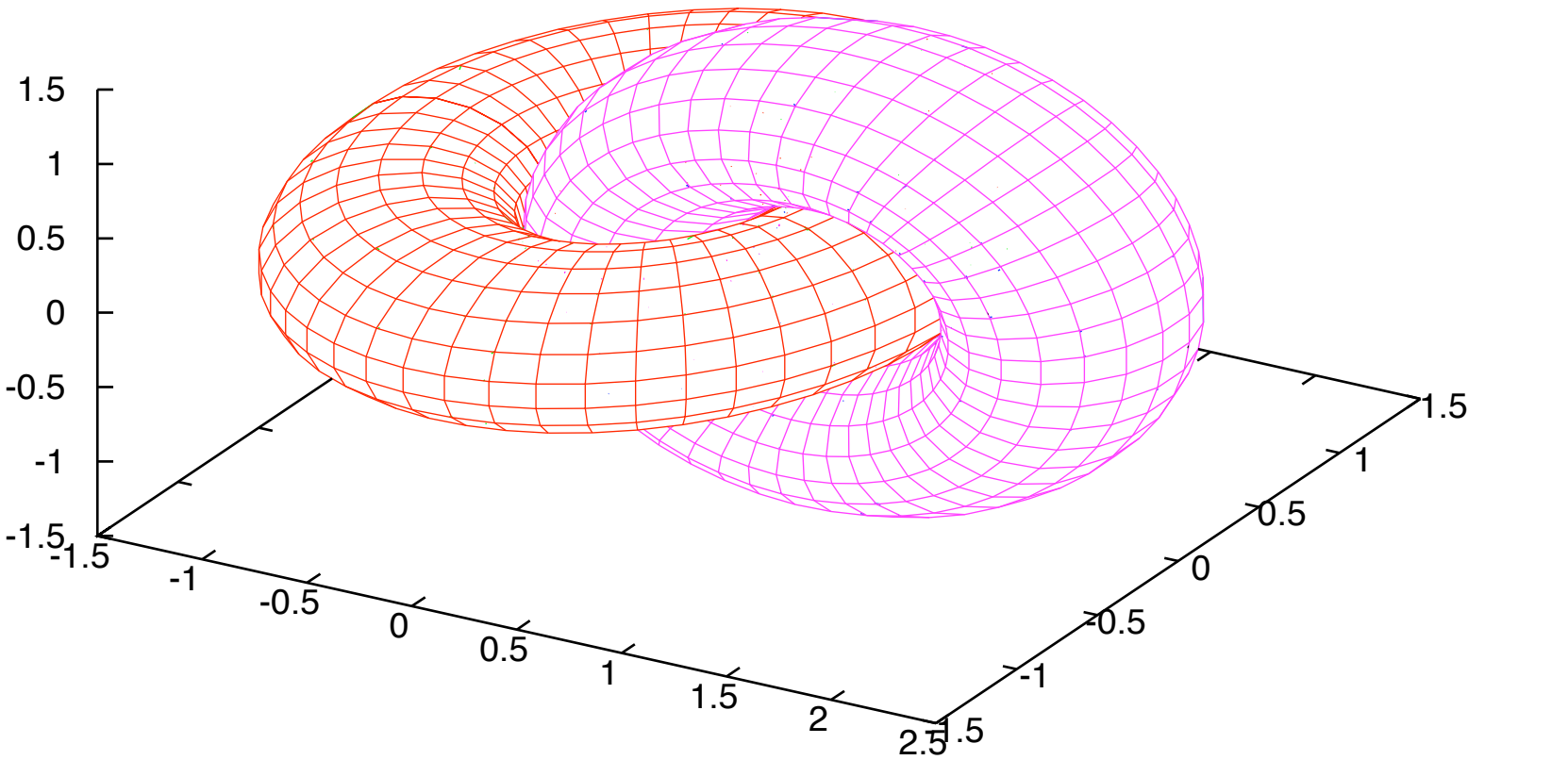

Parametric Helix
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5
-1.5
-1
-0.5
0
0.5
1
1.5
(1-0.1*cos(v))*cos(u), (1-0.1*cos(v))*sin(u), 0.1*(sin(v)+u/1.7-10)

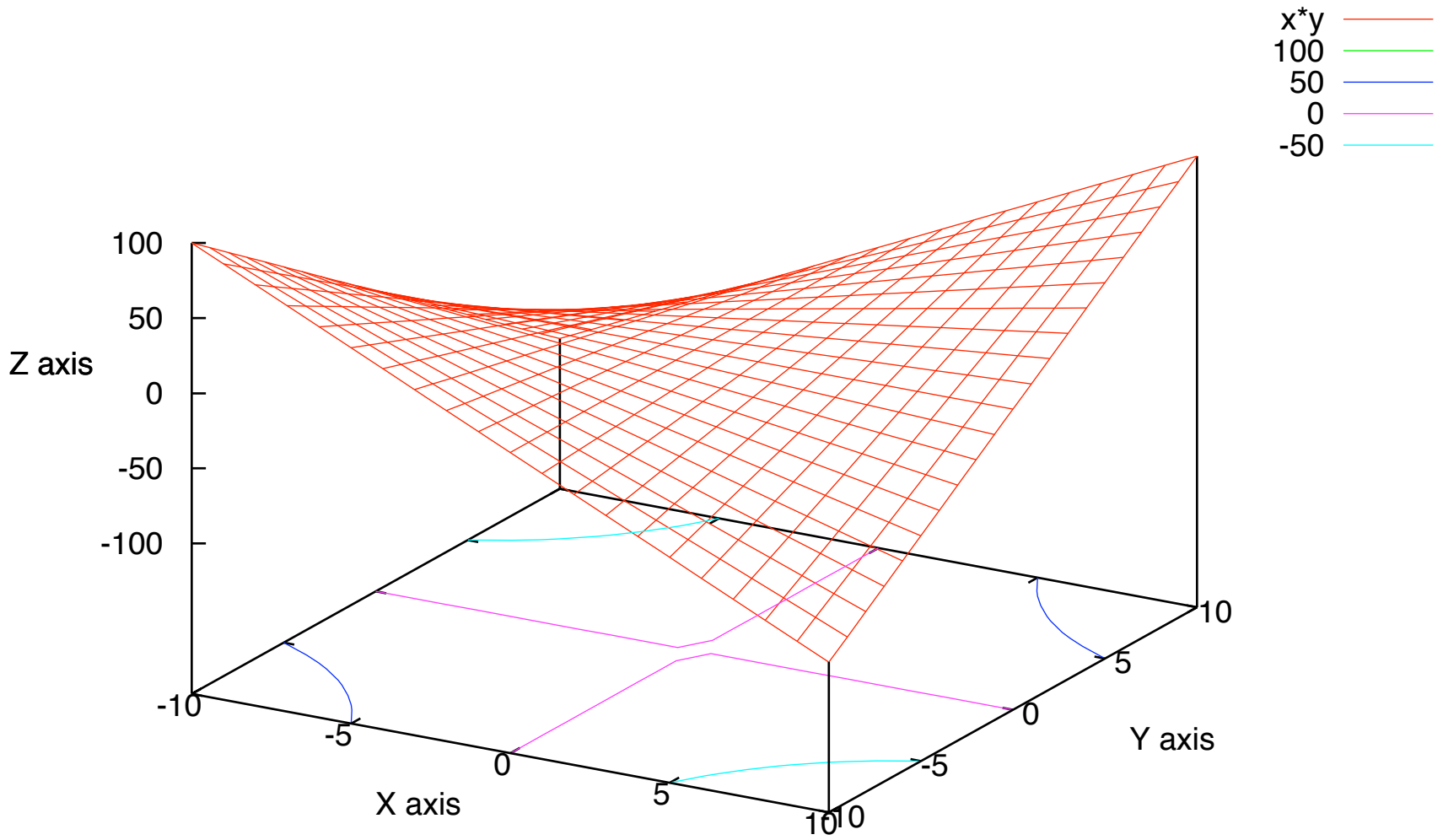

Parametric Shell (clipped to limited z range)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-6
-4
-2
0
2
4
6
8
10
-4
-3
-2
-1
0
1
2
3
4
cos(u)*u*(1+cos(v)/2), sin(v)*u/2, sin(u)*u*(1+cos(v)/2)

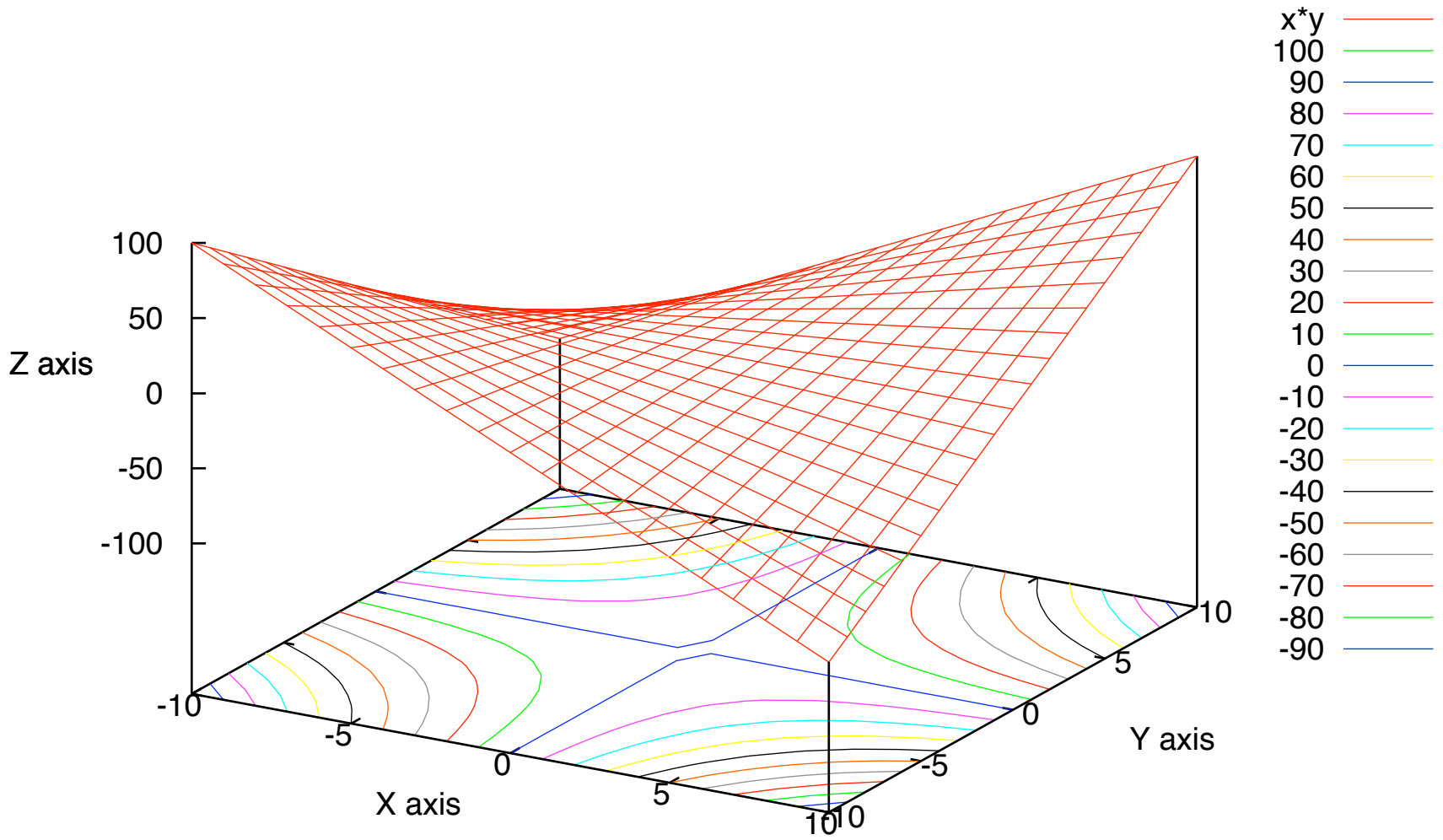
Parametric Shell (automatic z range)

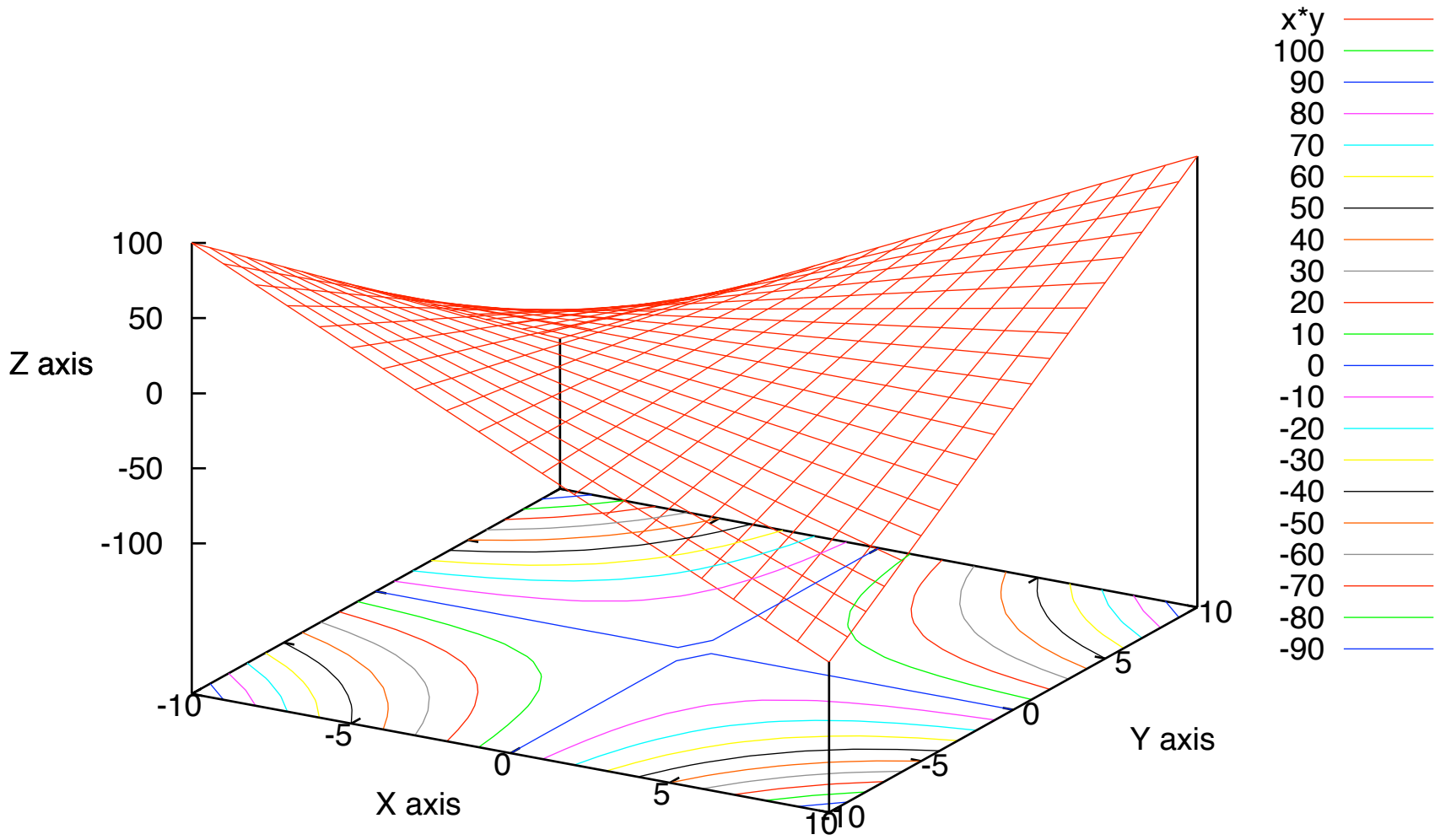


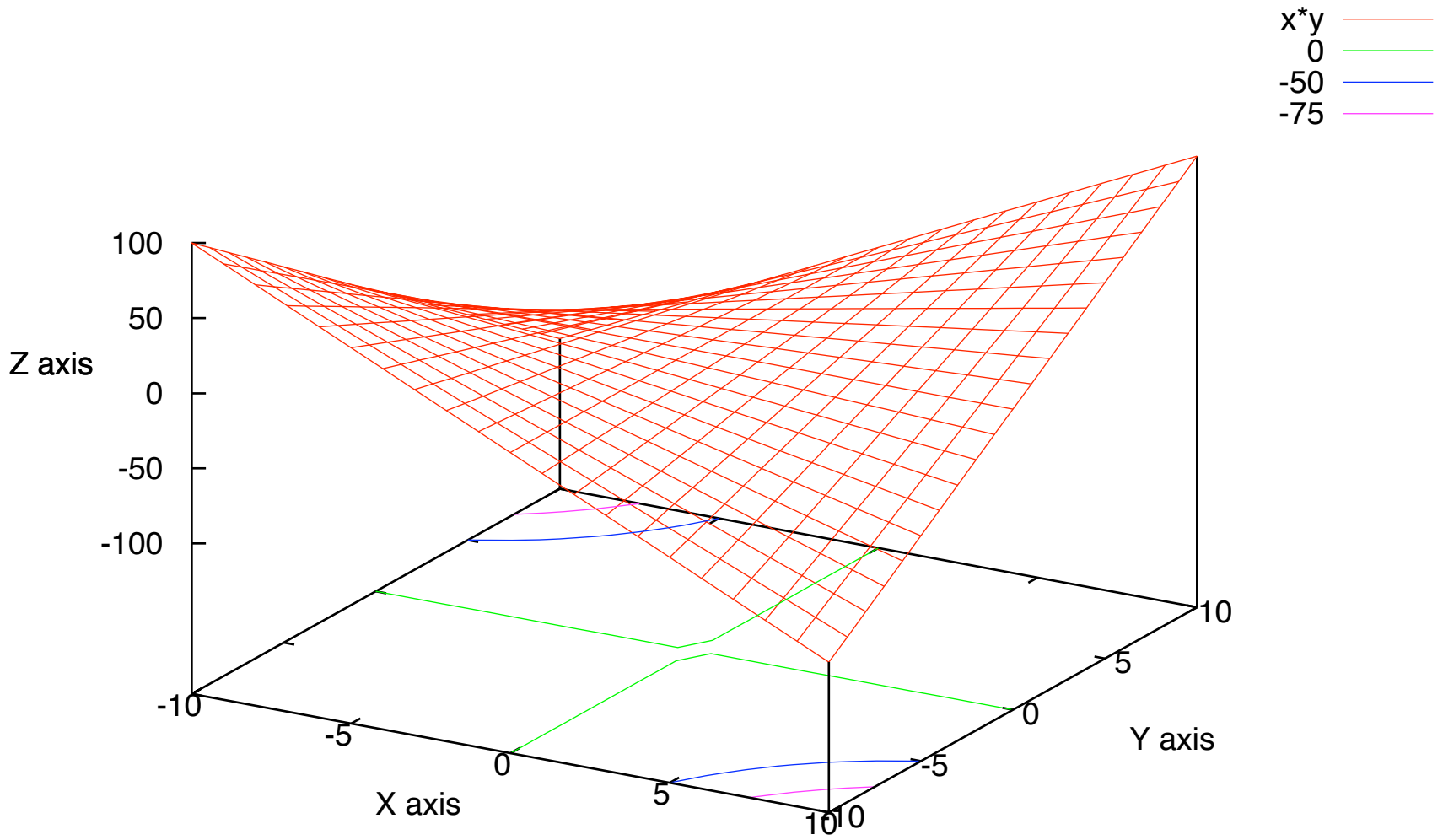
Interlocking Tori

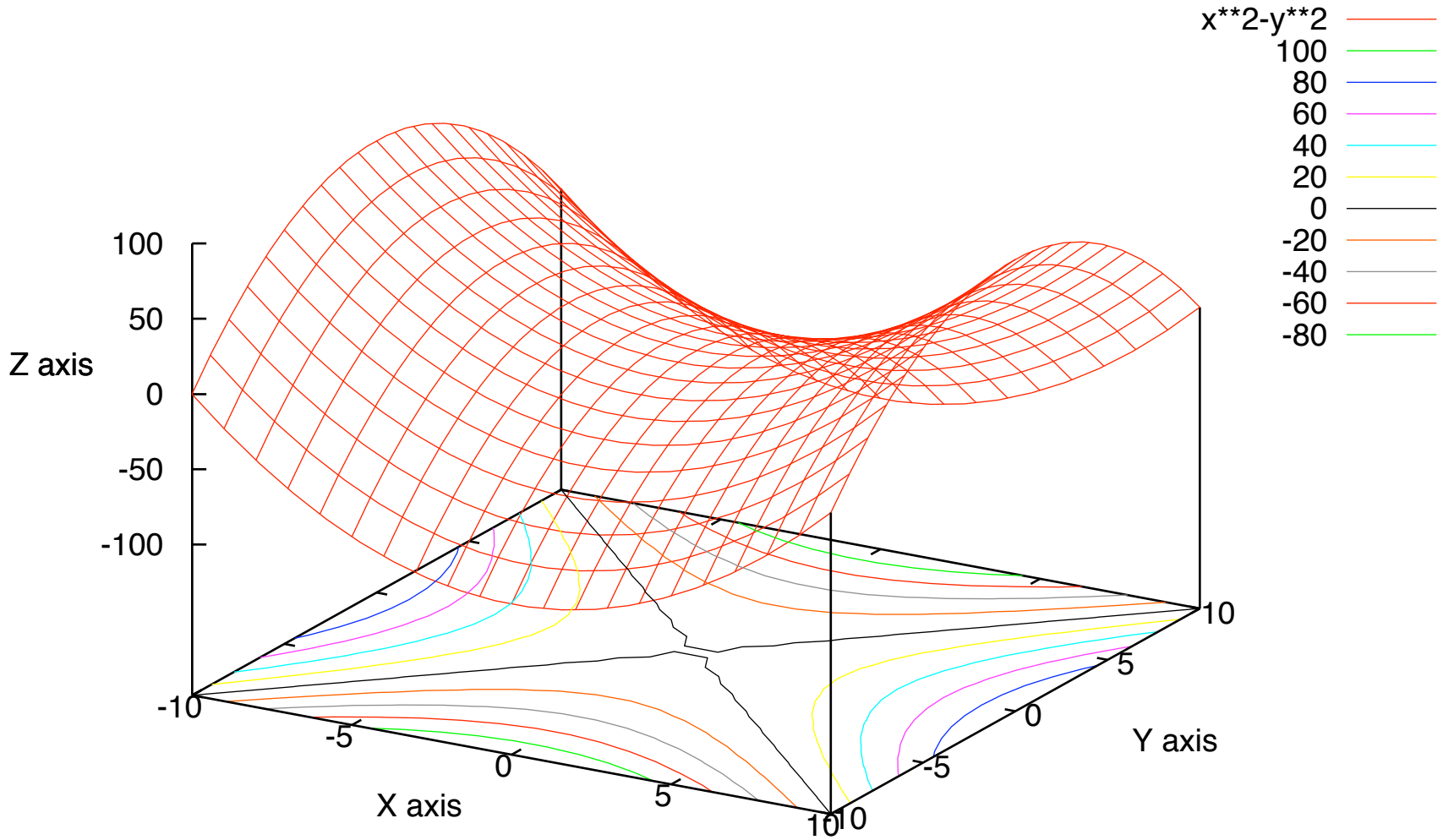


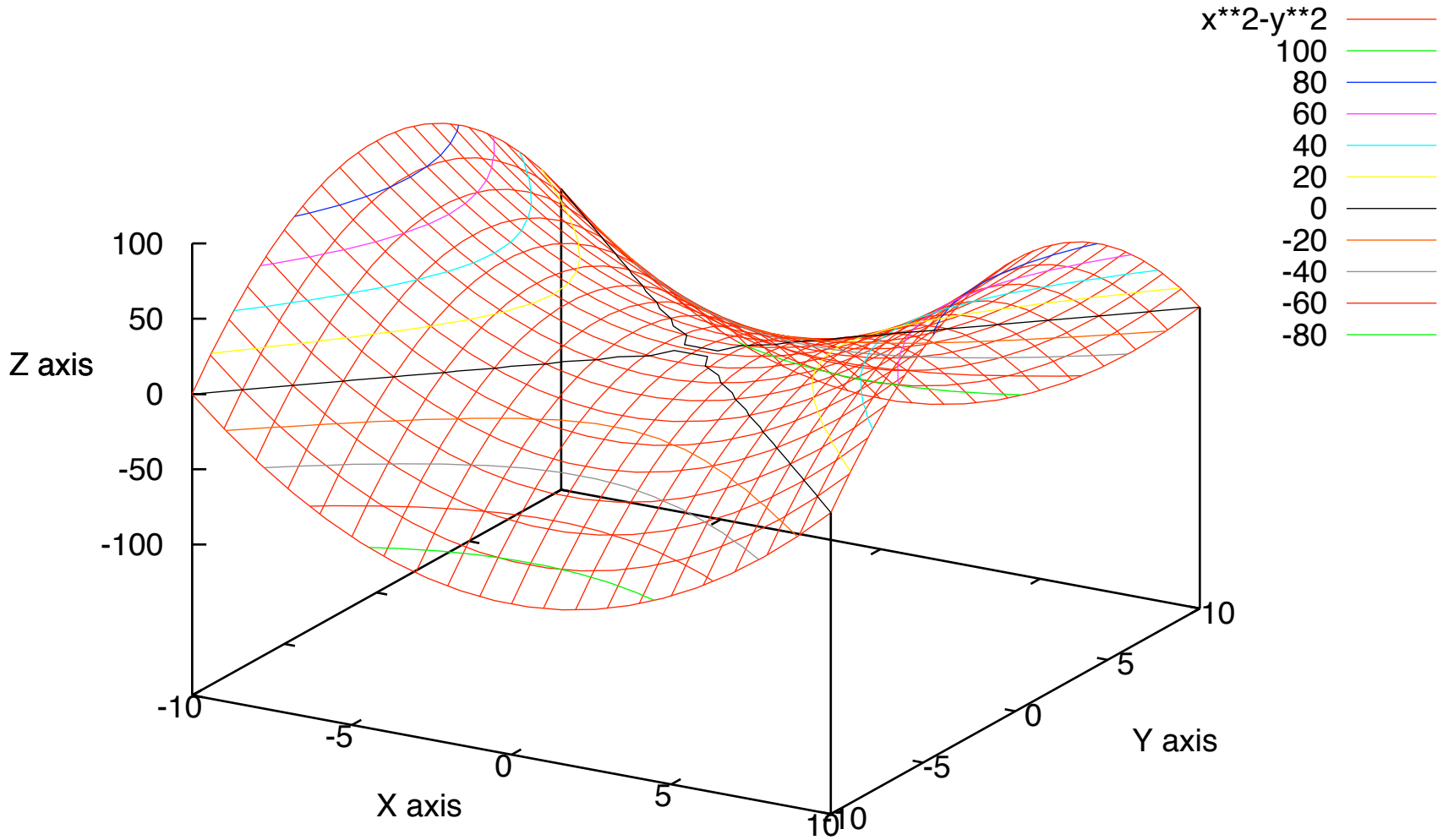

3D gnuplot demo - contour plot
x*y
100
50
0
-50
Z axis
100
50
0
-50
-100
-10
-5
0
5
10
X axis
10
5
0
-5
-10
Y axis

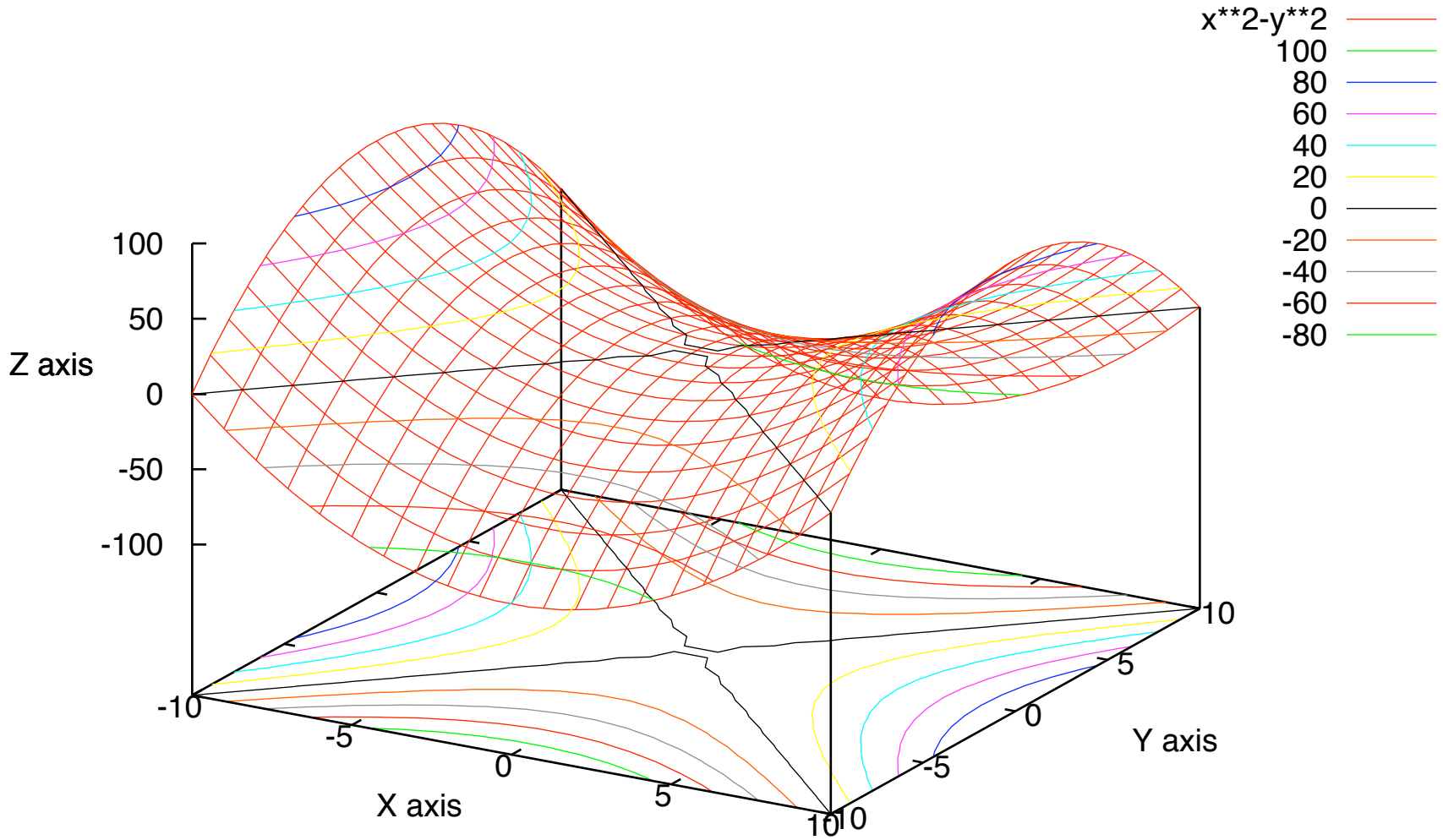

3D gnuplot demo - contour plot (more contours)
Z axis
100
50
0
-50
-100
X axis
-10
-5
0
5
10
Y axis
-10
-5
0
5
10
x*y
100
90
80
70
60
50
40
30
20
10
0
-10
-20
-30
-40
-50
-60
-70
-80
-90

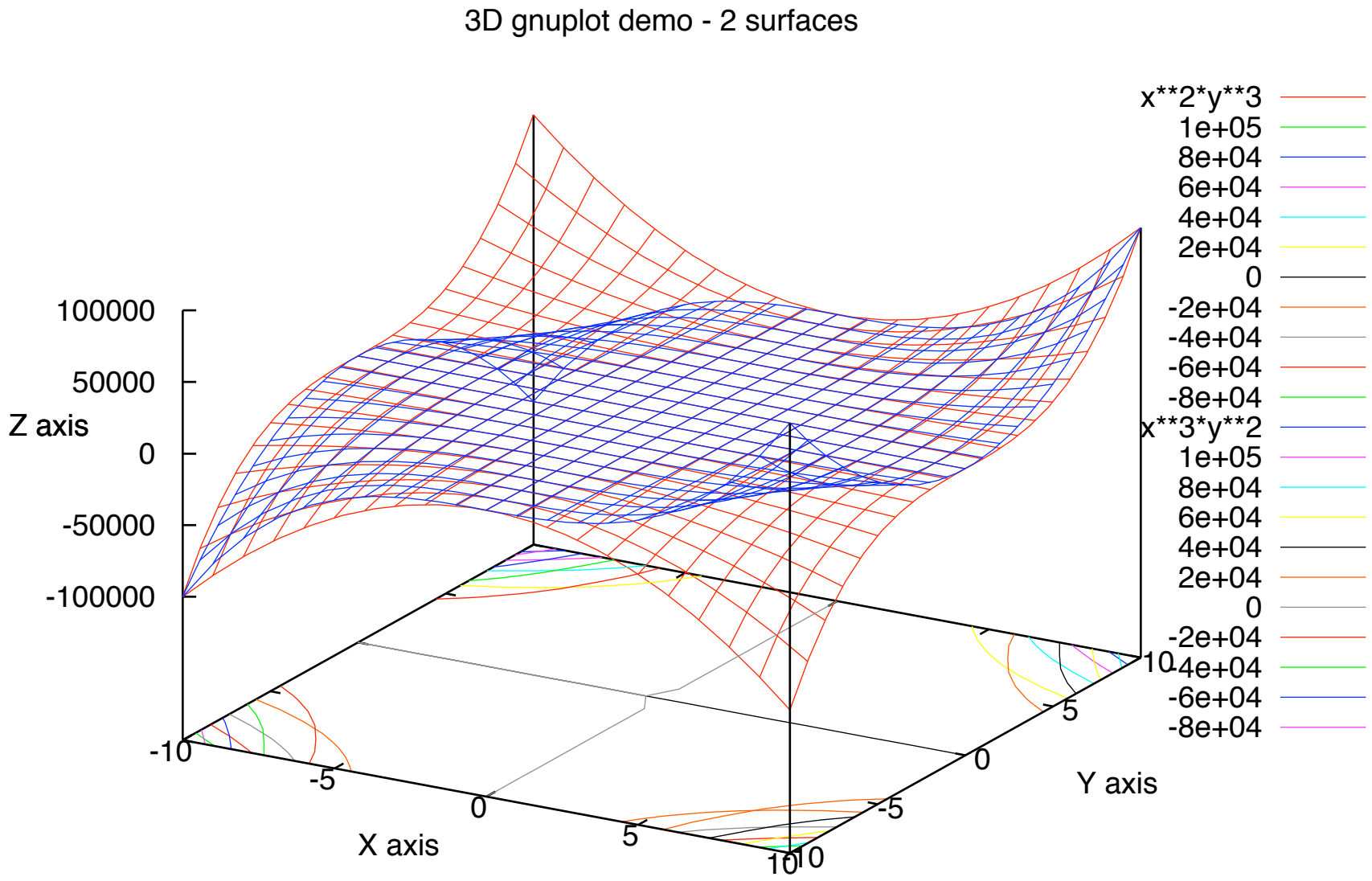

3D gnuplot demo - contour plot (every 10, starting at -100)
x*y
100
90
80
70
60
50
40
30
20
10
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
Z axis
100
50
0
-50
-100
X axis
-10
-5
0
5
10
Y axis
-10
-5
0
5
10

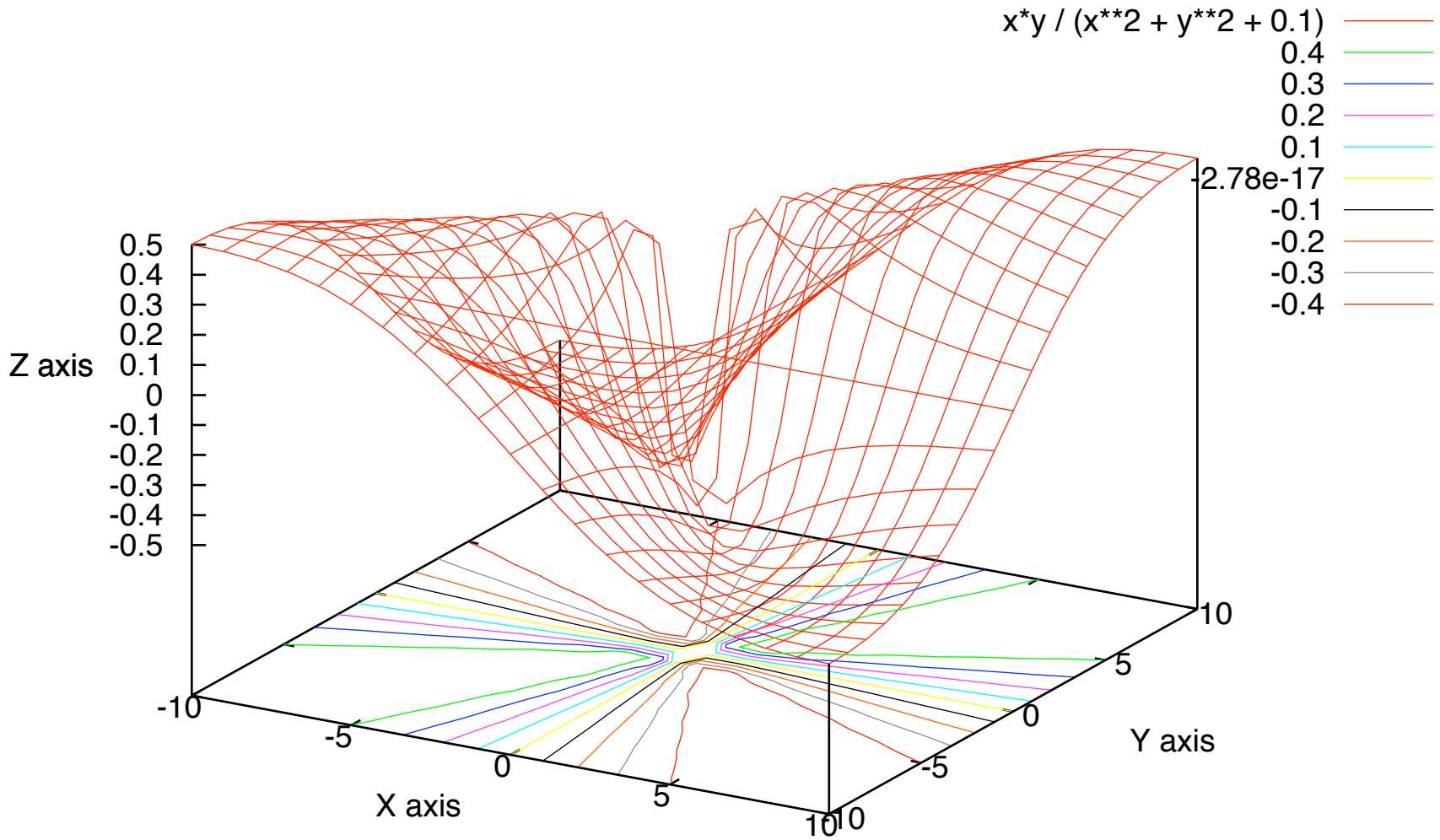

3D gnuplot demo - contour plot (at -75, -50, 0)
x*y
0
-50
-75
Z axis
100
50
0
-50
-100
X axis
-10
-5
0
5
10
Y axis
-10
-5
0
5
10

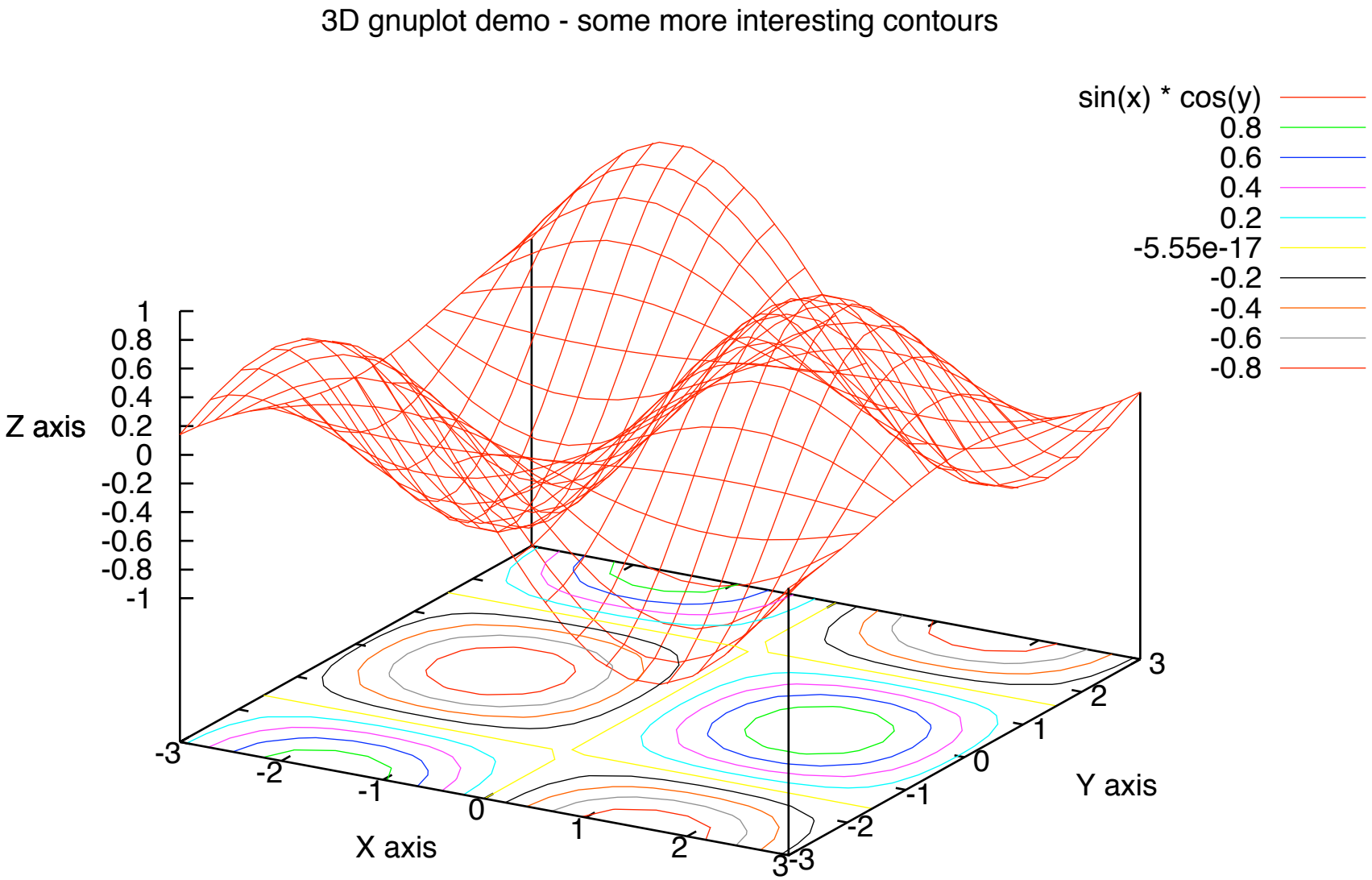

3D gnuplot demo - contour plot on base grid
x**2-y**2
100
80
60
40
20
0
-20
-40
-60
-80
Z axis
100
50
0
-50
-100
X axis
-10
-5
0
5
10
Y axis
-10
-5
0
5
10

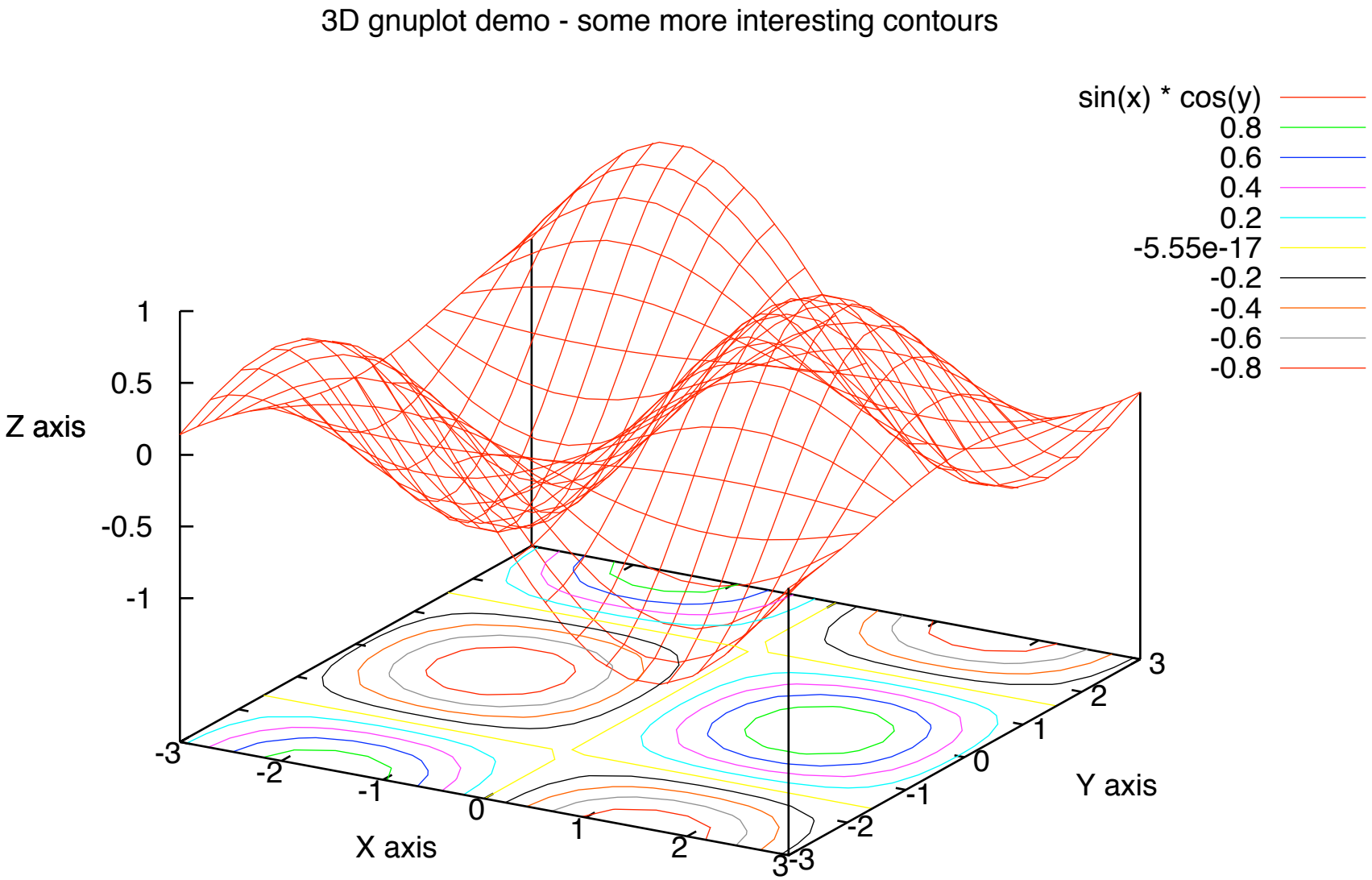

3D gnuplot demo - contour plot on surface
x**2-y**2
100
80
60
40
20
0
-20
-40
-60
-80
Z axis
100
50
0
-50
-100
X axis
-10
-5
0
5
10
Y axis
-10
-5
0
5
10

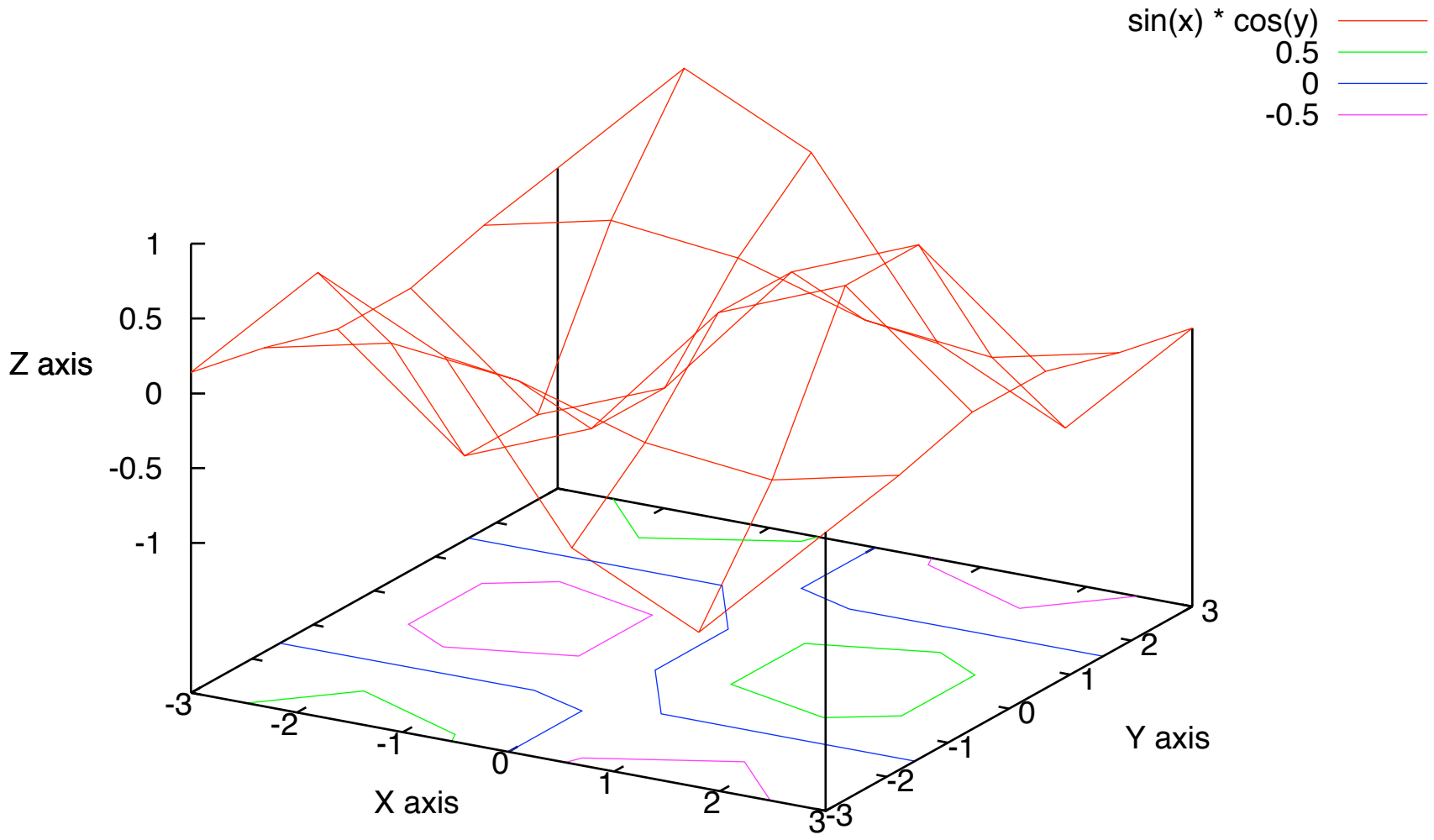

3D gnuplot demo - contour plot on both
x**2-y**2
100
80
60
40
20
0
-20
-40
-60
-80
Z axis
100
50
0
-50
-100
X axis
-10
-5
0
5
10
Y axis
-10
-5
0
5
10

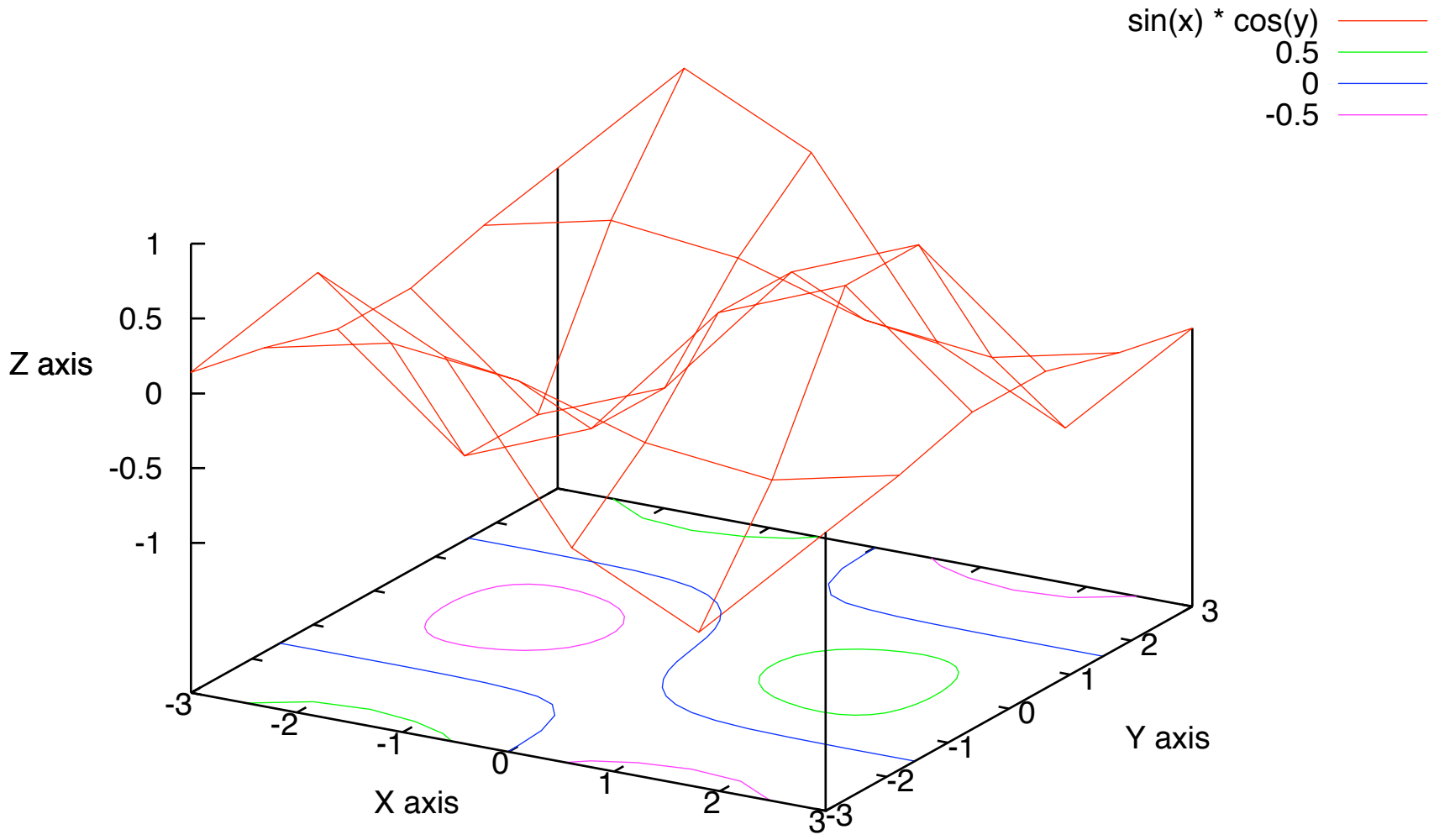

3D gnuplot demo - 2 surfaces
x**2*y**3
1e+05
8e+04
6e+04
4e+04
2e+04
0
-2e+04
-4e+04
-6e+04
-8e+04
x**3*y**2
1e+05
8e+04
6e+04
4e+04
2e+04
0
-2e+04
-4e+04
-6e+04
-8e+04
100000
50000
0
-50000
-100000
Z axis
-10
-5
0
5
10
X axis
-10
-5
0
5
10
Y axis

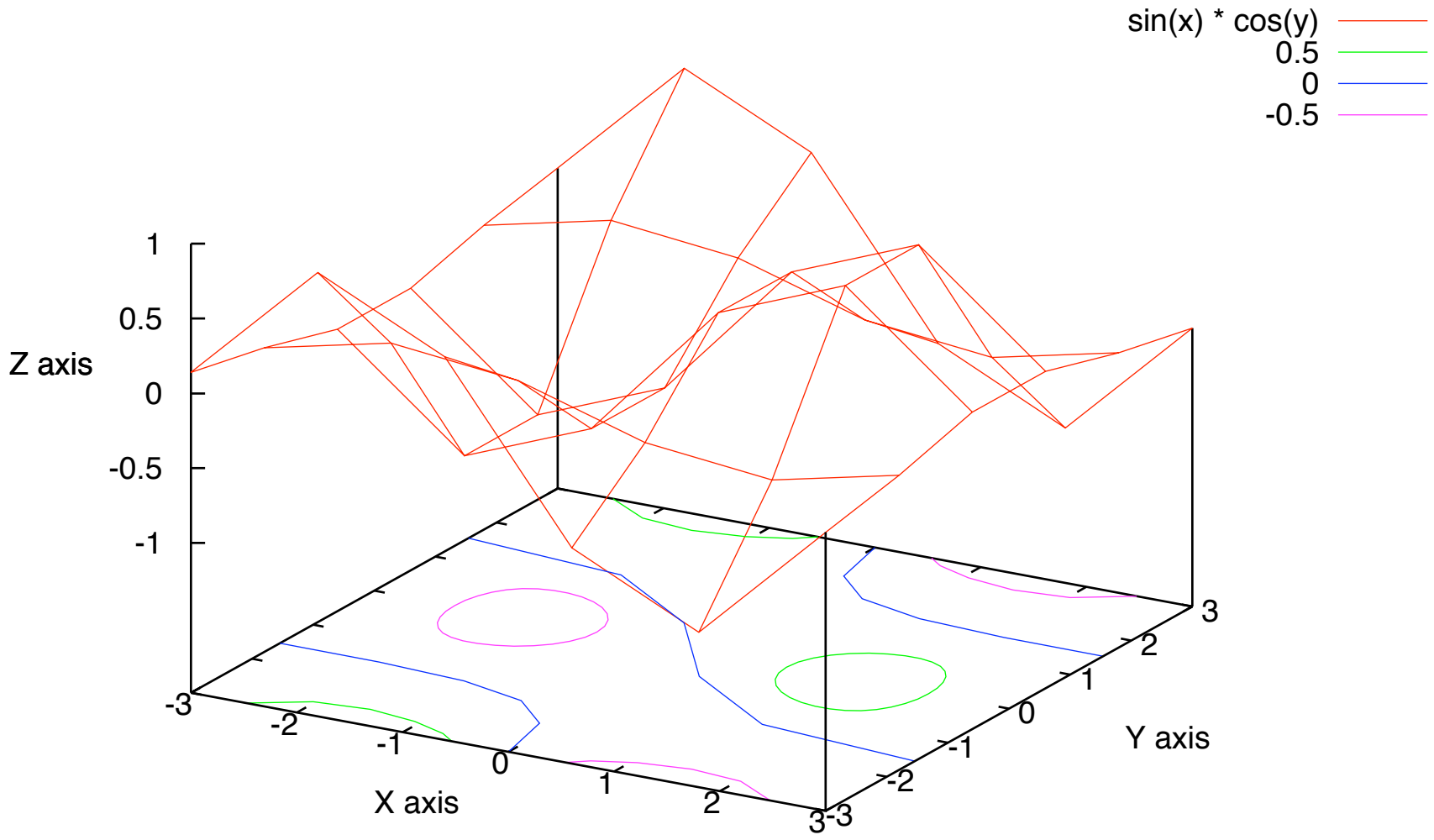

3D gnuplot demo - some more interesting contours
x*y / (x**2 + y**2 + 0.1)
0.4
0.3
0.2
0.1
2.78e-17
-0.1
-0.2
-0.3
-0.4
Z axis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-10
-5
0
5
10
X axis
10
5
0
-5
-10
Y axis

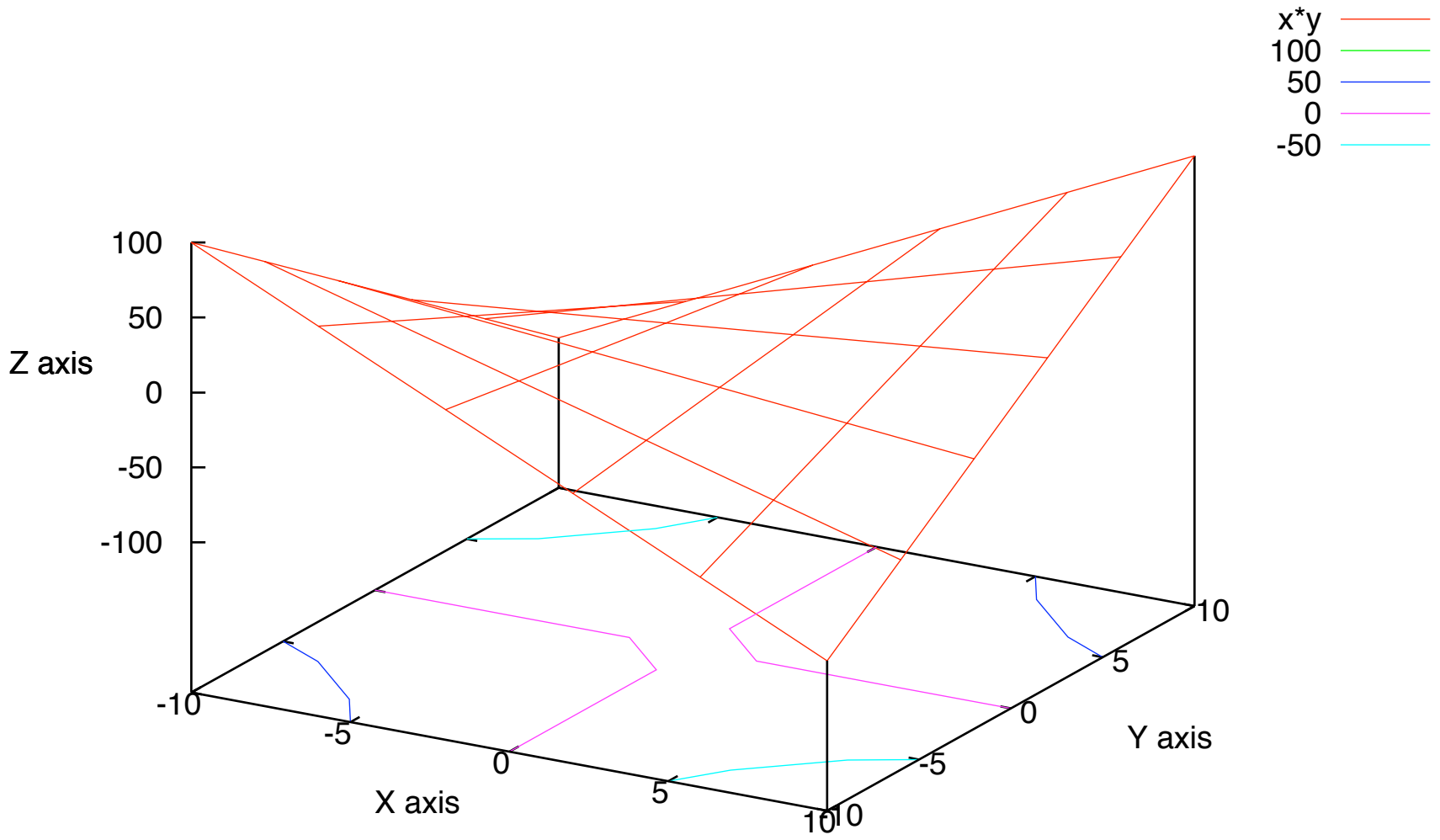

3D gnuplot demo - some more interesting contours
sin(x) * cos(y)
0.8
0.6
0.4
0.2
-5.55e-17
-0.2
-0.4
-0.6
-0.8
Z axis
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
X axis
-3
-2
-1
0
1
2
3
Y axis
-3
-2
-1
0
1
2
3

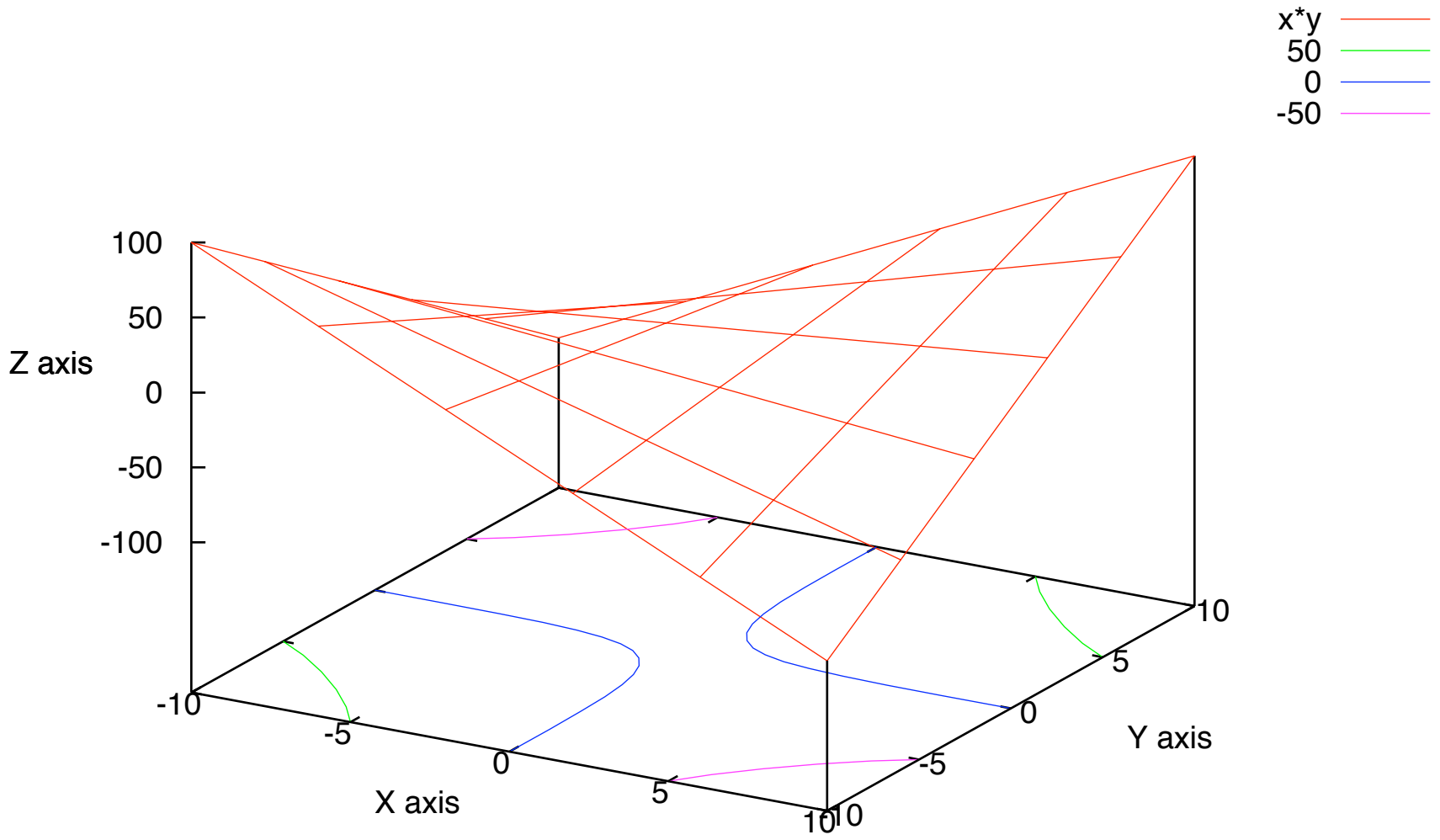

3D gnuplot demo - some more interesting contours
sin(x) * cos(y)
0.8
0.6
0.4
0.2
-5.55e-17
-0.2
-0.4
-0.6
-0.8
Z axis
1
0.5
0
-0.5
-1
X axis
-3
-2
-1
0
1
2
3
Y axis
-3
-2
-1
0
1
2
3

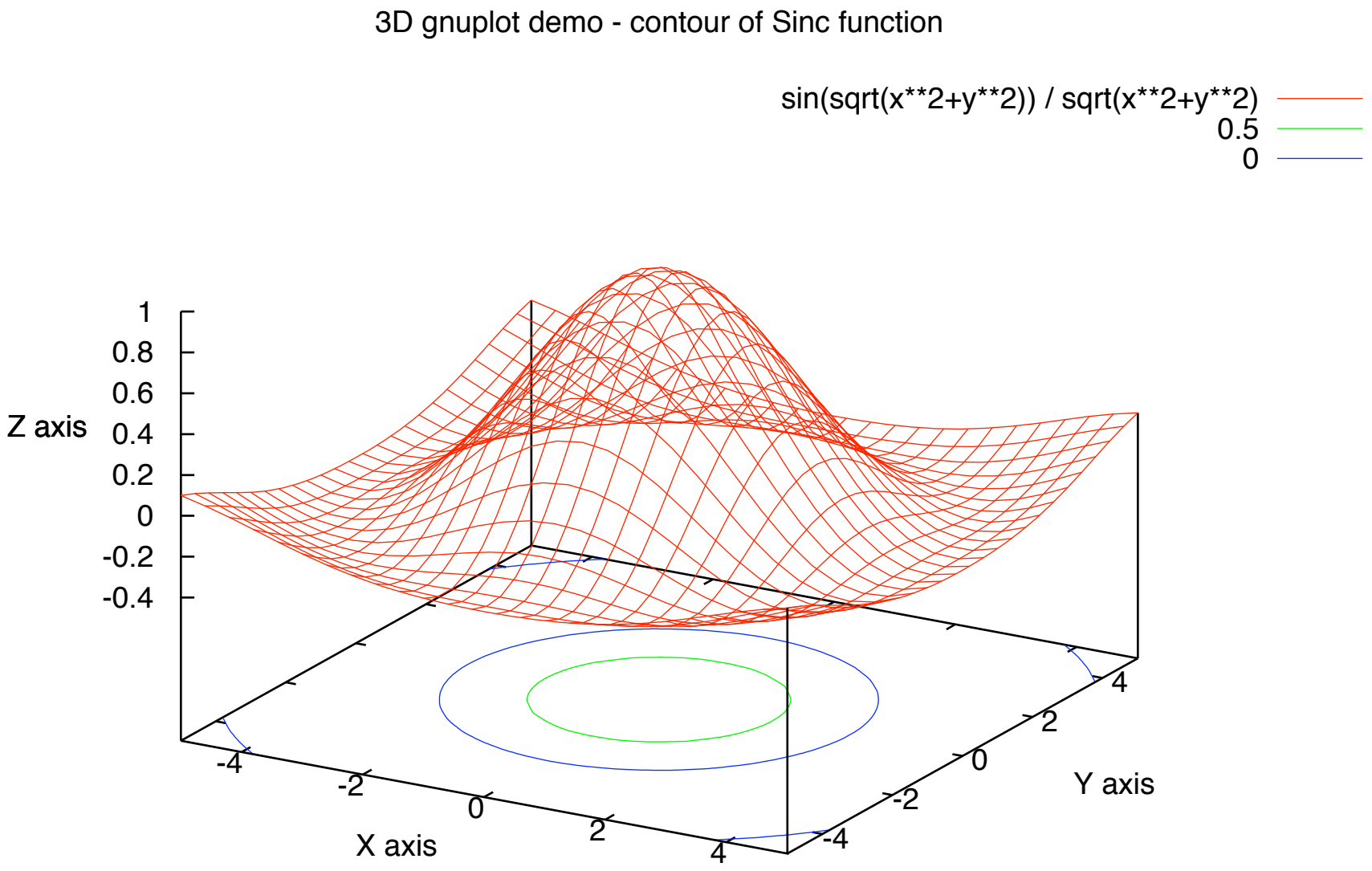

3D gnuplot demo - low resolution (6x6)
sin(x) * cos(y)
0.5
0
-0.5
Z axis
1
0.5
0
-0.5
-1
X axis
-3
-2
-1
0
1
2
3
Y axis
-3
-2
-1
0
1
2
3

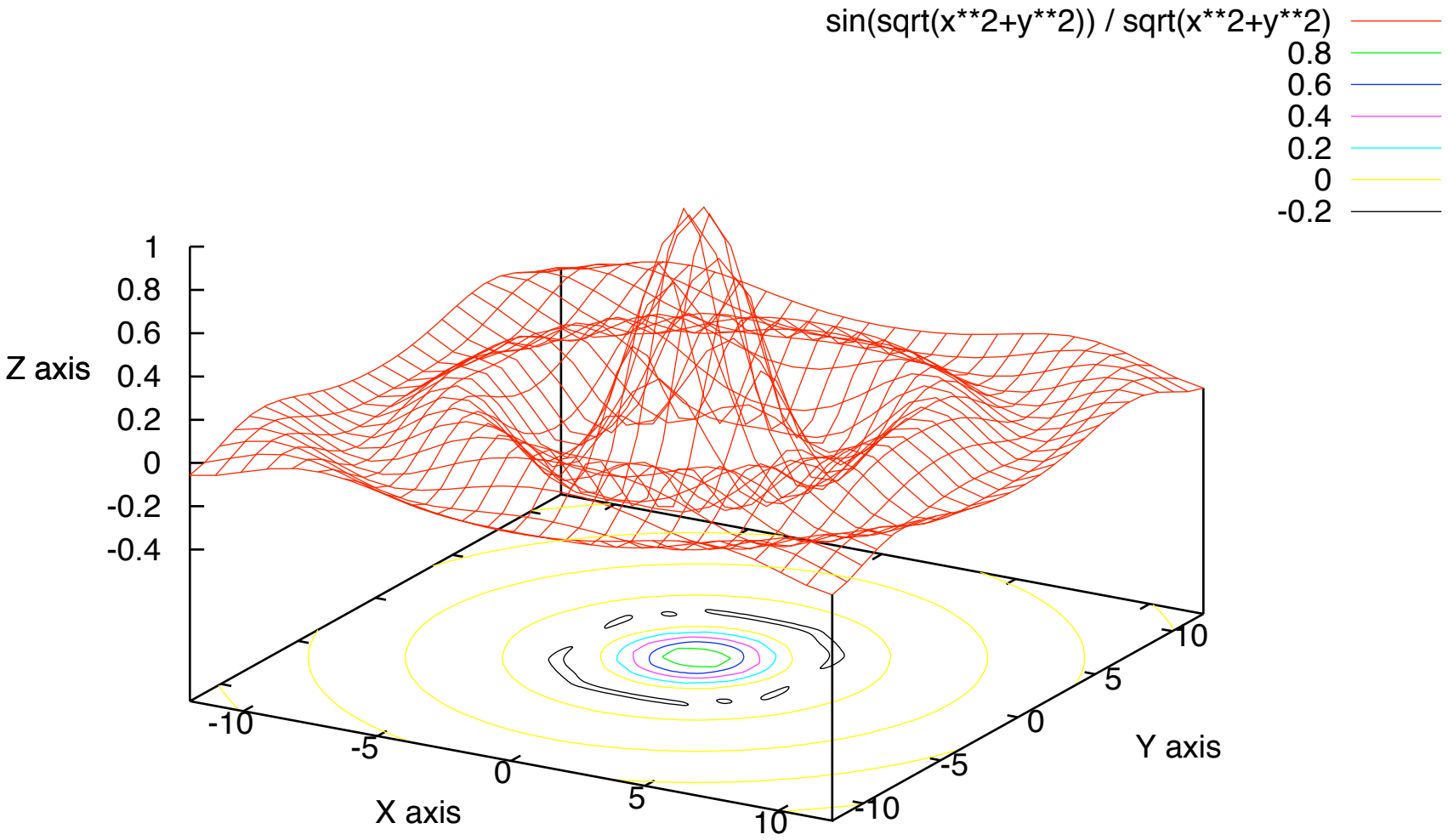

3D gnuplot demo - low resolution (6x6) using bspline approx.
sin(x) * cos(y)
0.5
0
-0.5
Z axis
1
0.5
0
-0.5
-1
X axis
-3
-2
-1
0
1
2
3
Y axis
-3
-2
-1
0
1
2
3

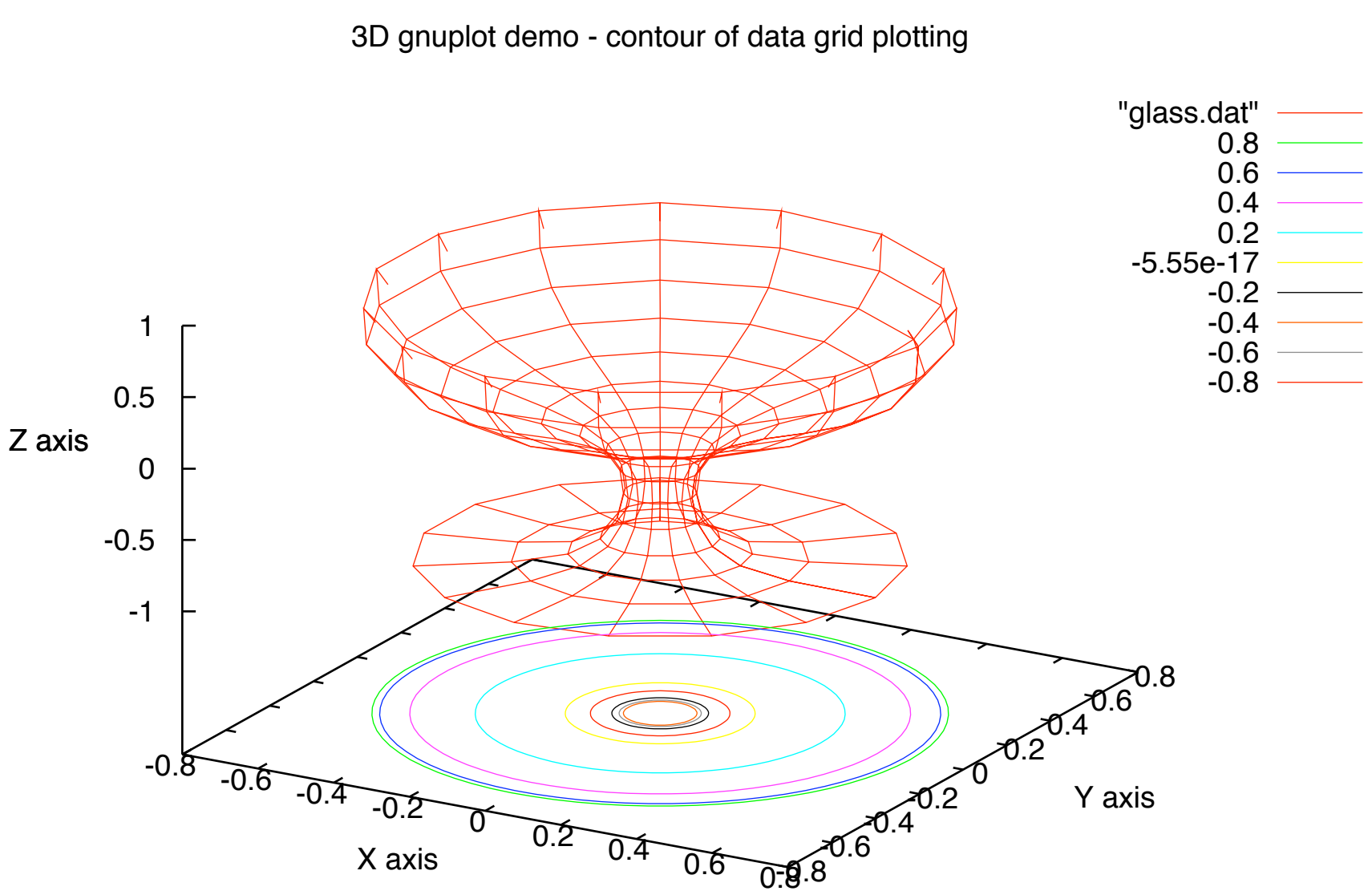

3D gnuplot demo - low resolution (6x6) raise bspline order.
sin(x) * cos(y)
0.5
0
-0.5
Z axis
1
0.5
0
-0.5
-1
X axis
-3
-2
-1
0
1
2
3
Y axis
-3
-2
-1
0
1
2
3

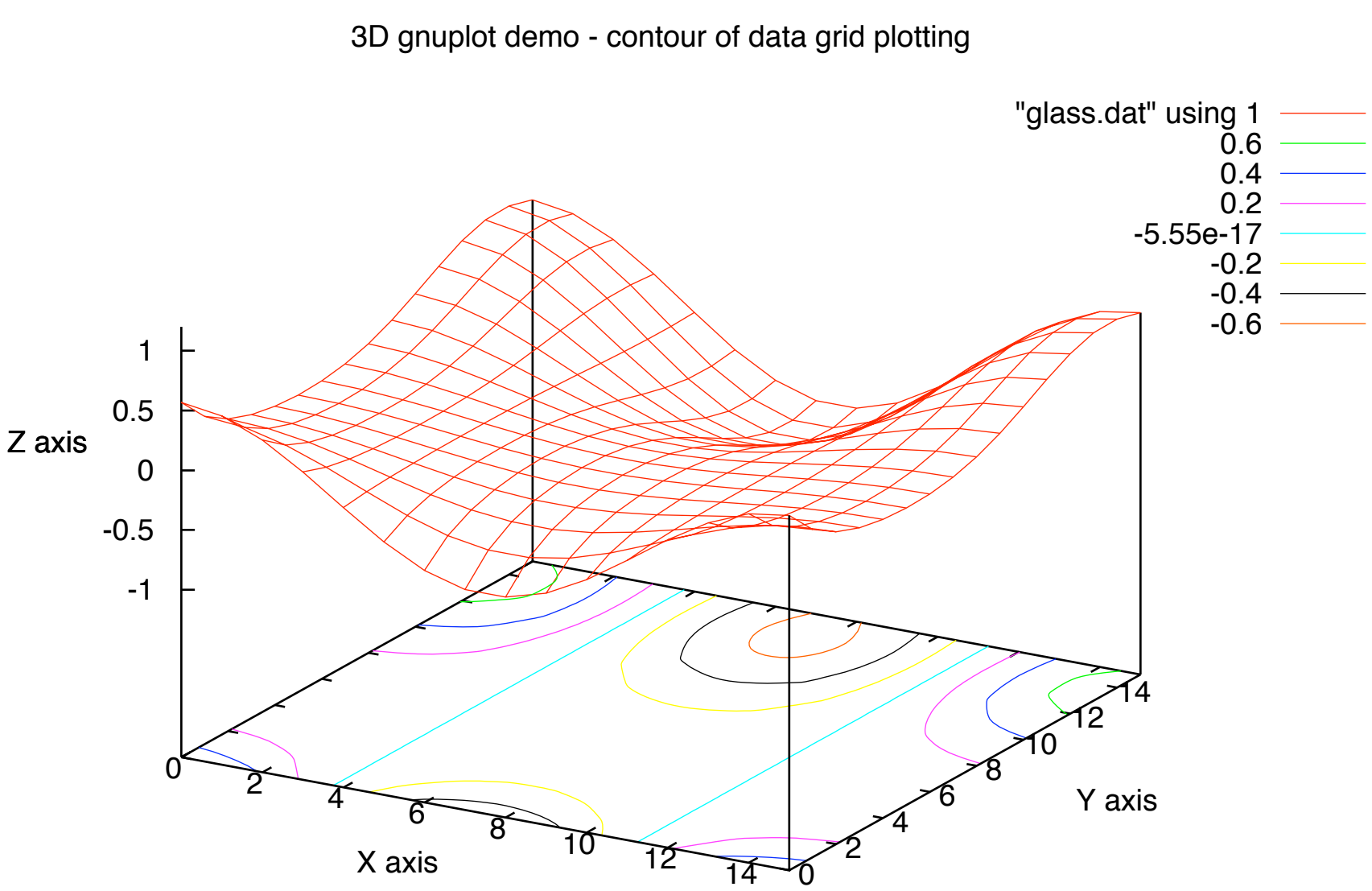

3D gnuplot demo - low resolution (6x6) using linear contours.
x*y
100
50
0
-50
Z axis
100
50
0
-50
-100
-10
-5
0
5
10
X axis
10
5
0
-5
-10
Y axis

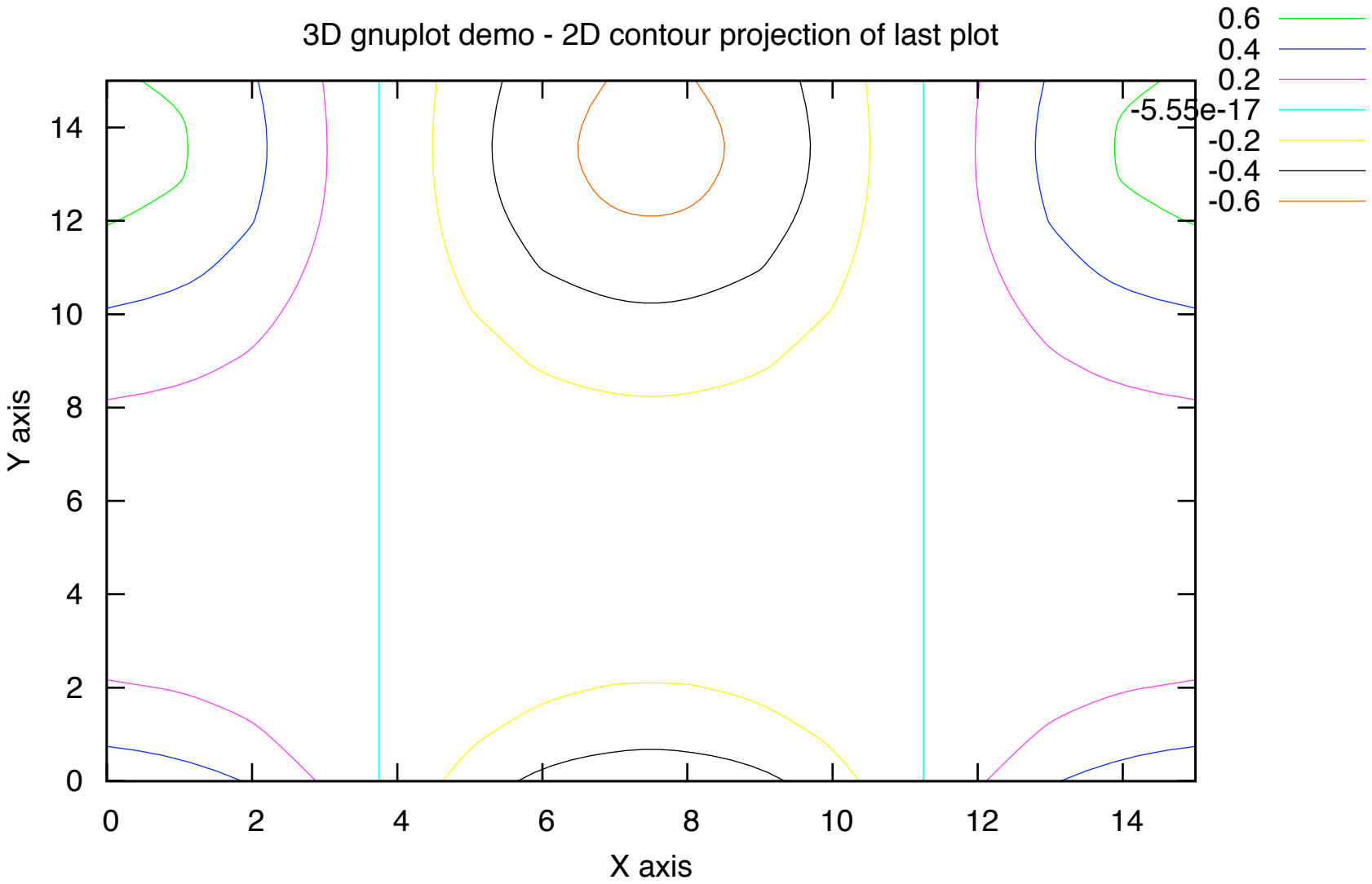

3D gnuplot demo - low resolution (6x6) using bspline approx.
x*y
50
0
-50
Z axis
100
50
0
-50
-100
X axis
-10
-5
0
5
10
Y axis
10
5
0
-5
10

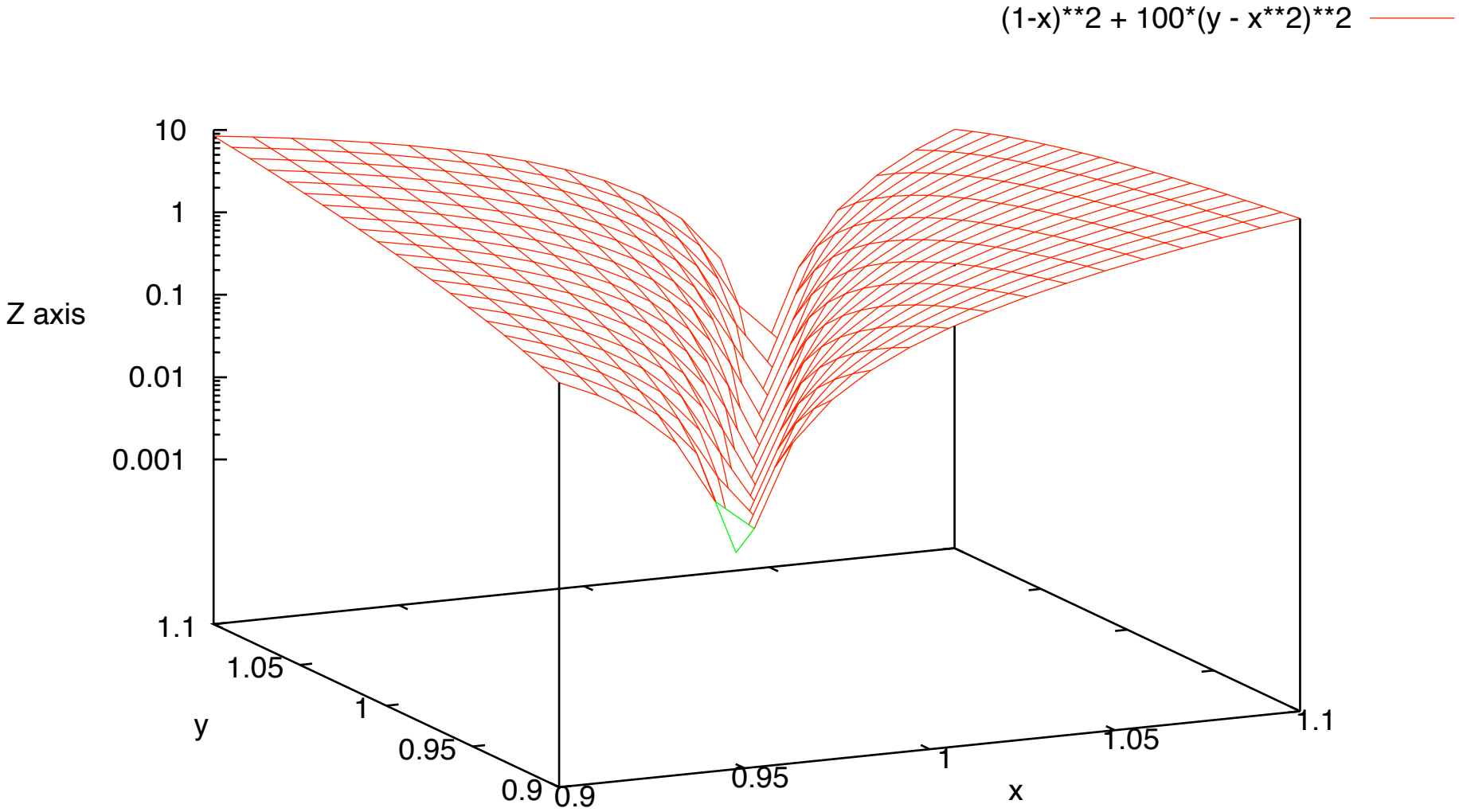

3D gnuplot demo - contour of Sinc function
sin(sqrt(x**2+y**2)) / sqrt(x**2+y**2)
0.5
0
Z axis
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
X axis
-4
-2
0
2
4
Y axis
-4
-2
0
2
4

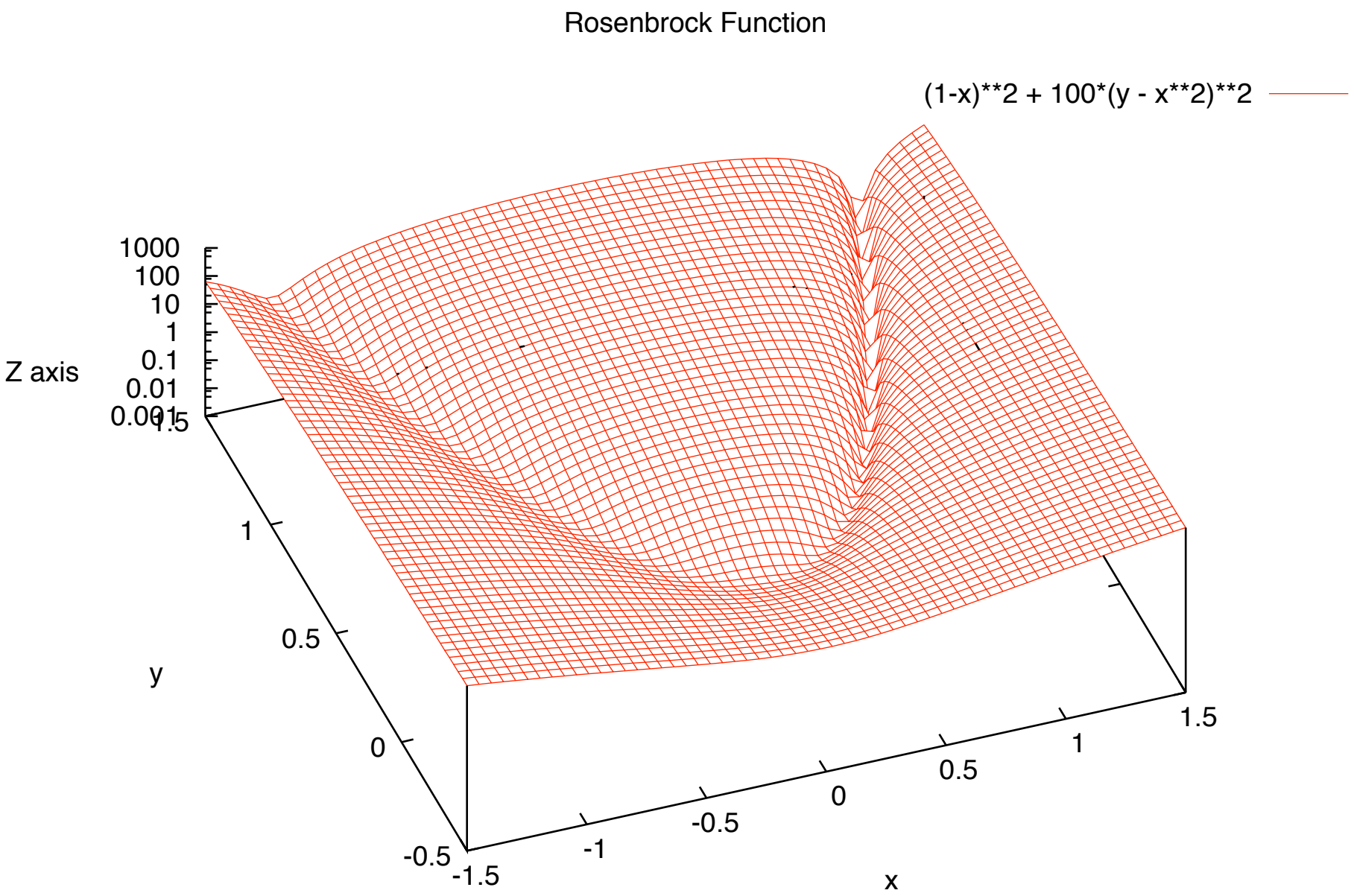

3D gnuplot demo - contour of Sinc function
sin(sqrt(x**2+y**2)) / sqrt(x**2+y**2)
0.8
0.6
0.4
0.2
0
-0.2
Z axis
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
X axis
-10
-5
0
5
10
Y axis
-10
-5
0
5
10

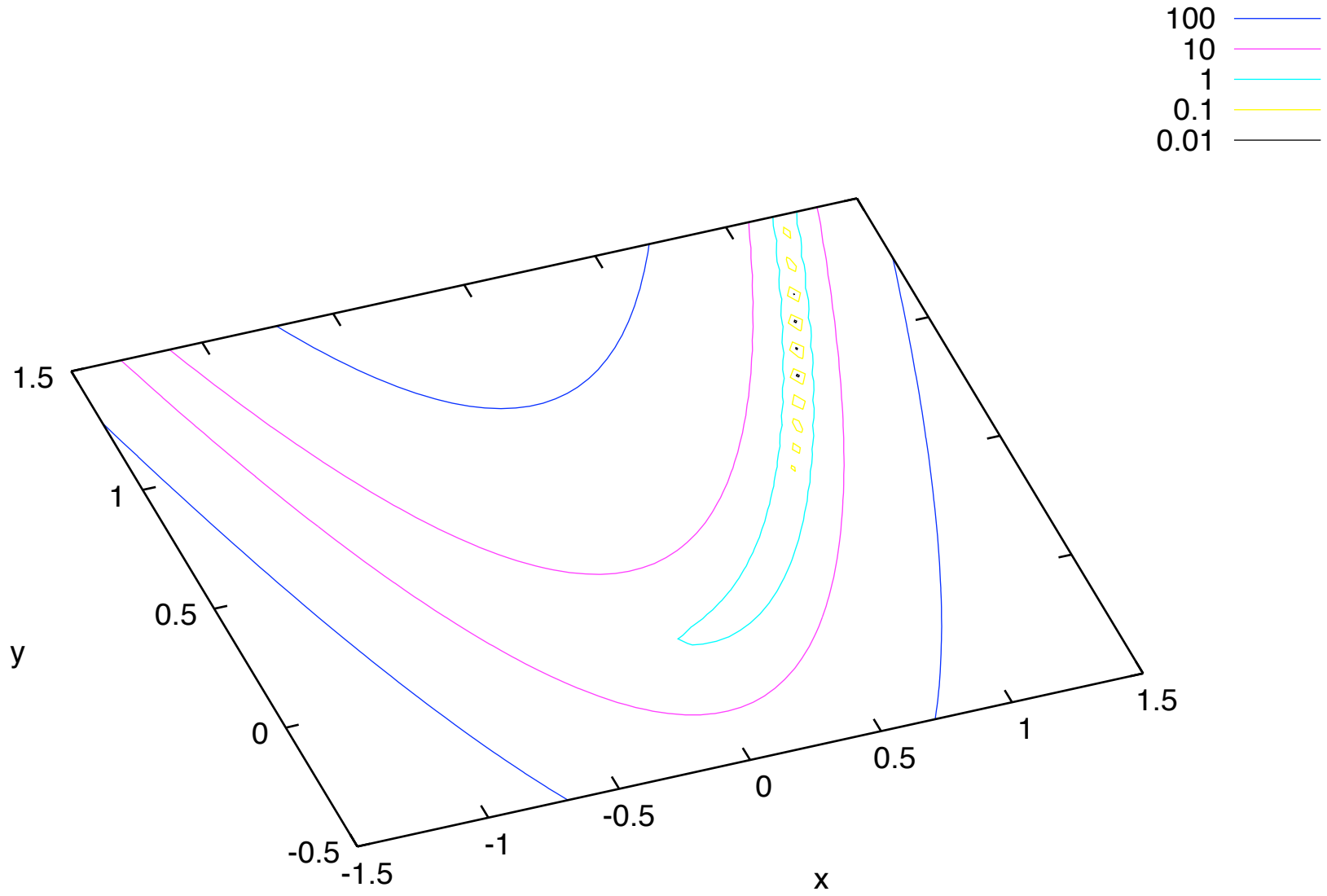

3D gnuplot demo - contour of data grid plotting
"glass.dat"
0.8
0.6
0.4
0.2
-5.55e-17
-0.2
-0.4
-0.6
-0.8
Z axis
1
0.5
0
-0.5
-1
X axis
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
Y axis
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8


3D gnuplot demo - contour of data grid plotting
"glass.dat" using 1
0.6
0.4
0.2
-5.55e-17
-0.2
-0.4
-0.6
Z axis
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

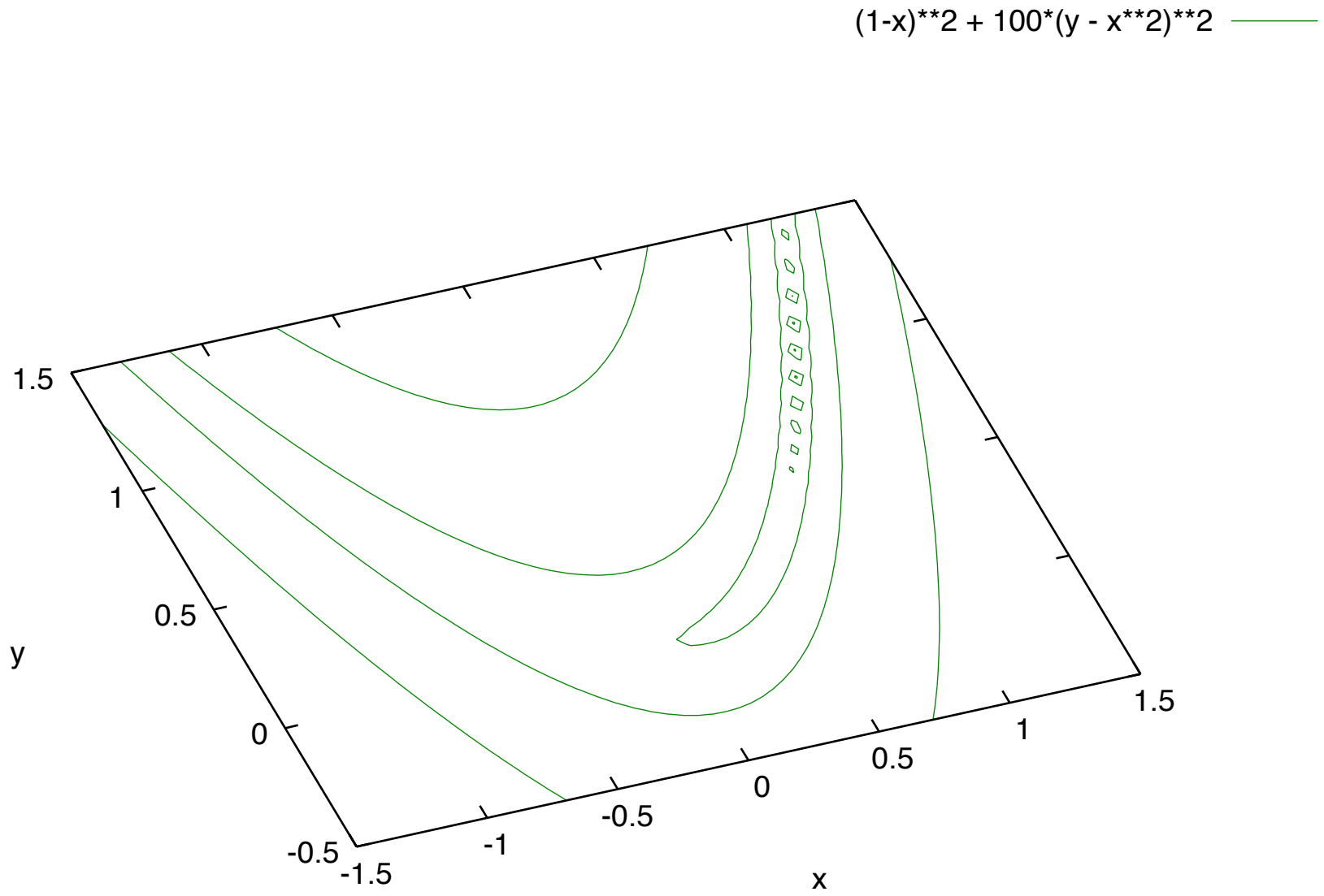

3D gnuplot demo - 2D contour projection of last plot
0.6
0.4
0.2
-5.55e-17
-0.2
-0.4
-0.6
Y axis
X axis
0
2
4
6
8
10
12
14


Rosenbrock Function
(1-x)**2 + 100*(y - x**2)**2
Z axis
10
1
0.1
0.01
0.001
1.1
1.05
1
0.95
0.9
y
0.9
0.95
1
1.05
1.1
x

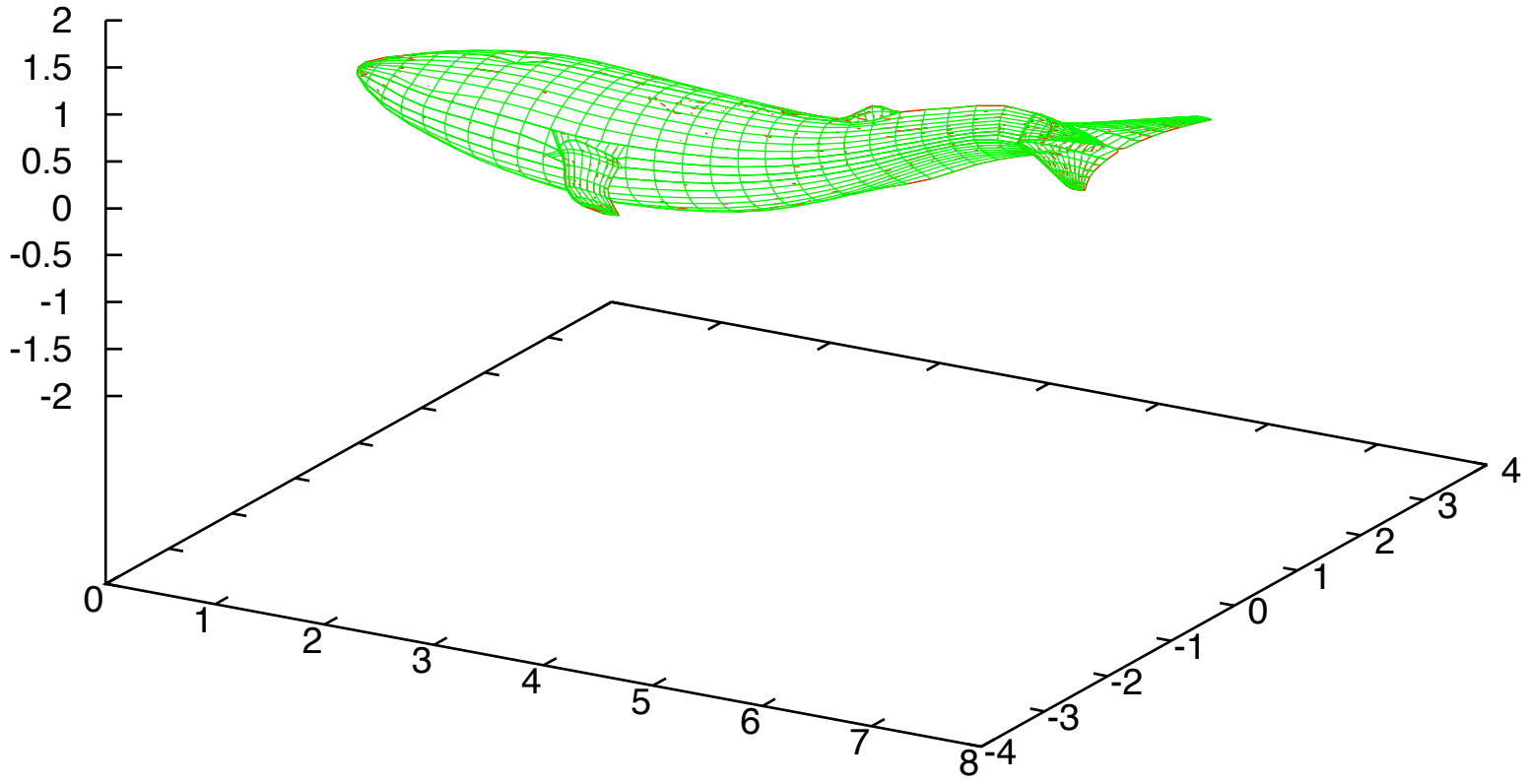

Rosenbrock Function
(1-x)**2 + 100*(y - x**2)**2
Z axis
1000
100
10
1
0.1
0.01
0.001
y
1.5
1
0.5
0
-0.5
-1.5
-1
-0.5
0
0.5
1
1.5
x


Rosenbrock Function
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

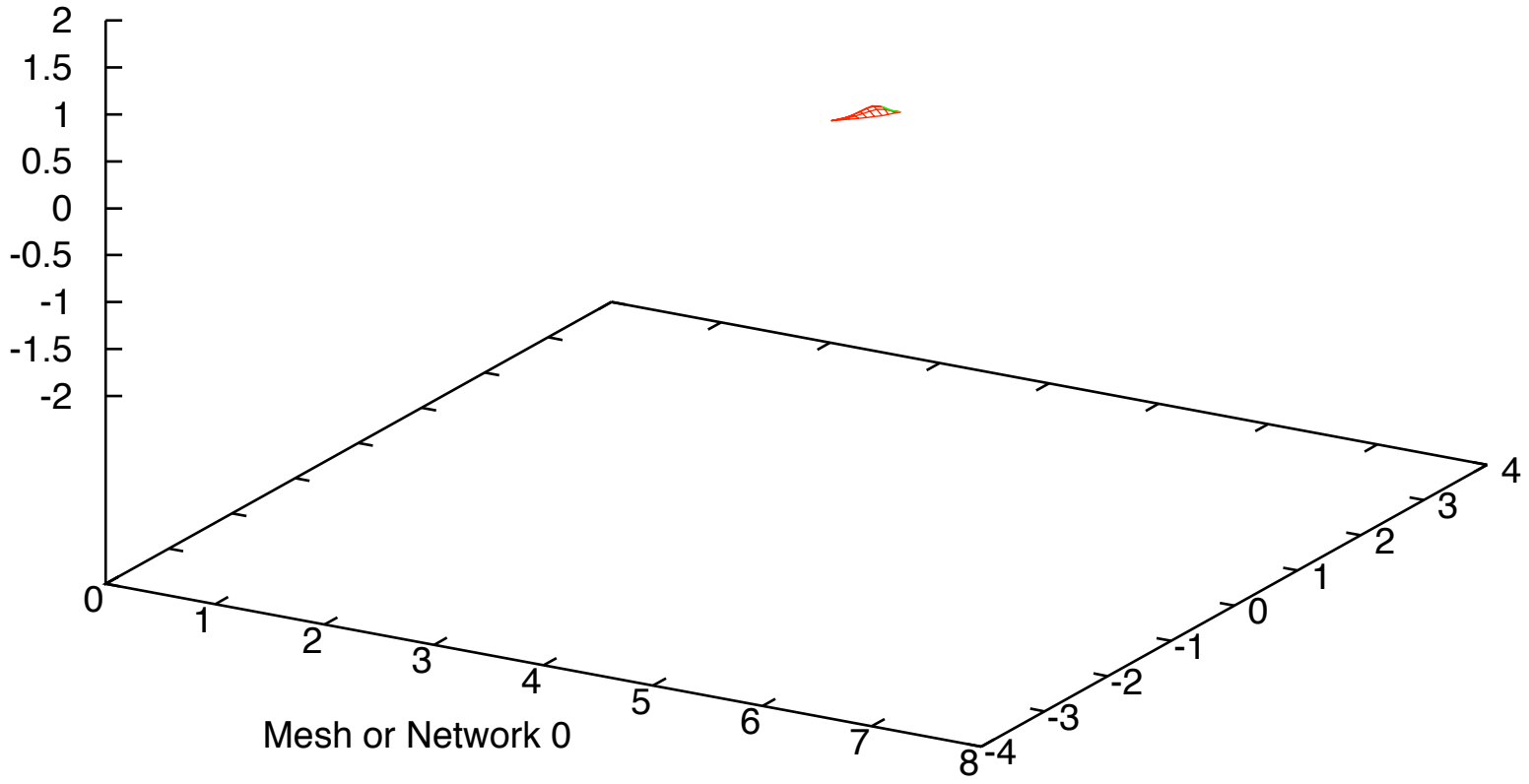
All contours drawn in a single color



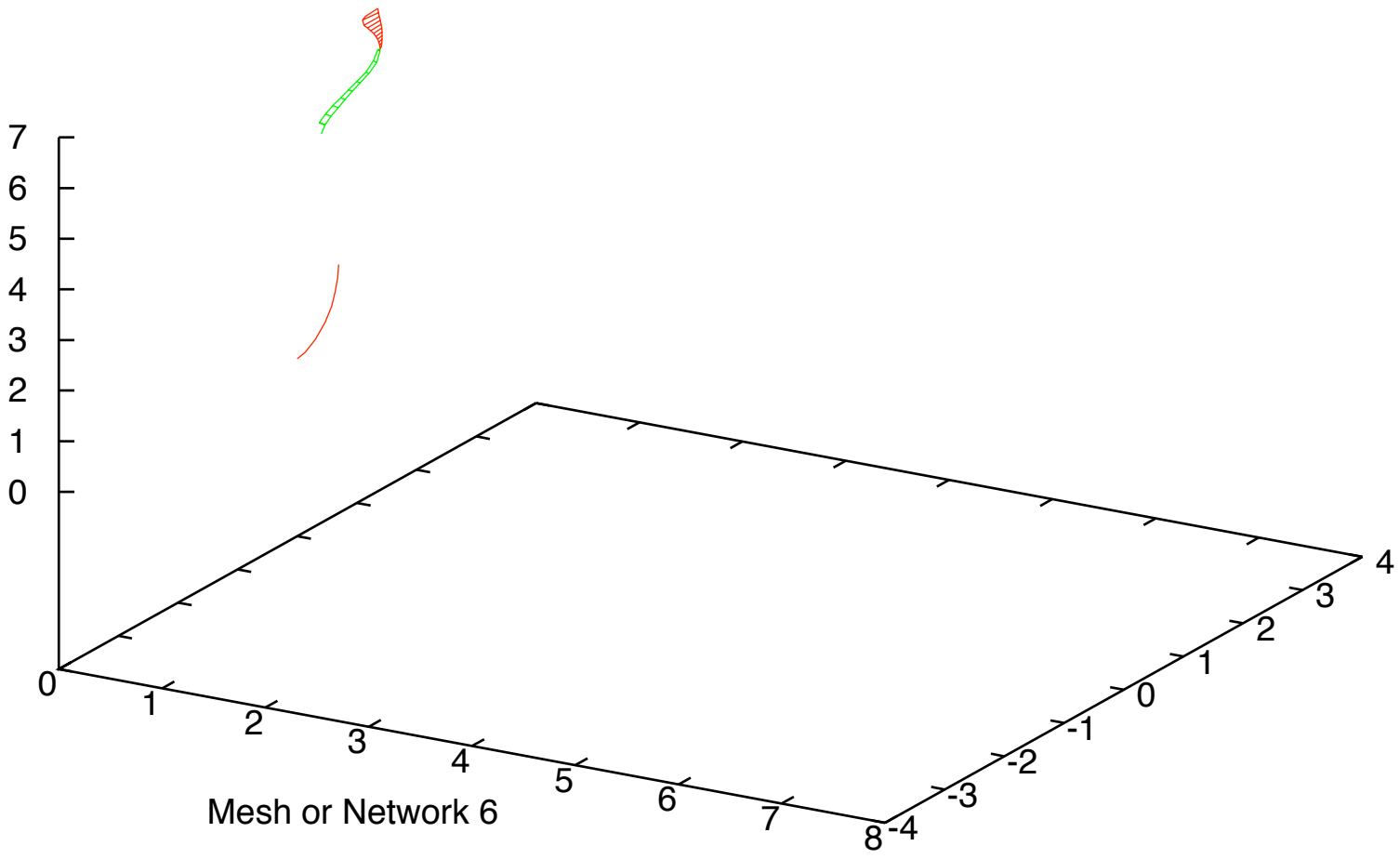
Demo of multiple mesh per file capability - Digitized Blue Whale



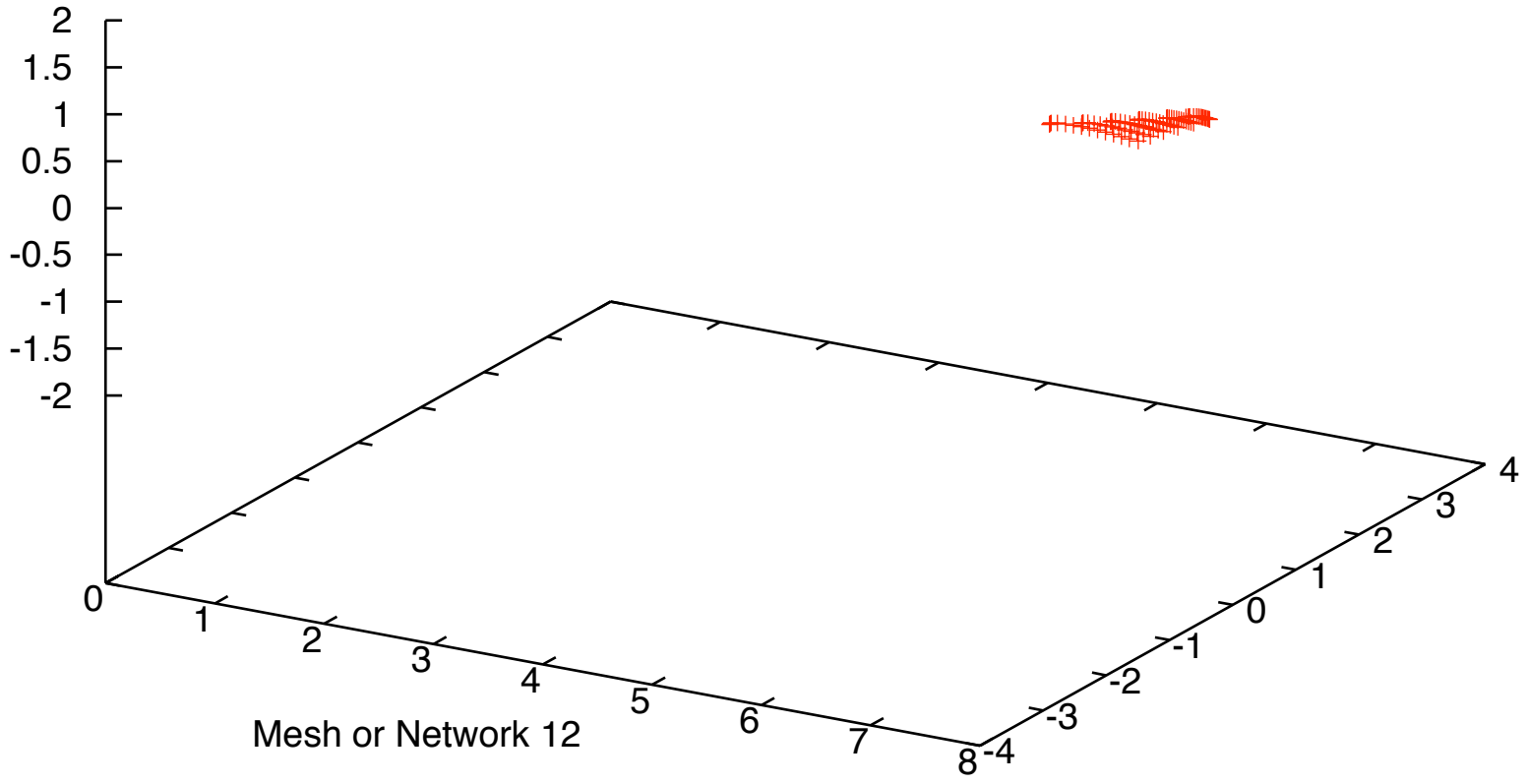
Demo of multiple mesh per file capability - Digitized Blue Whale



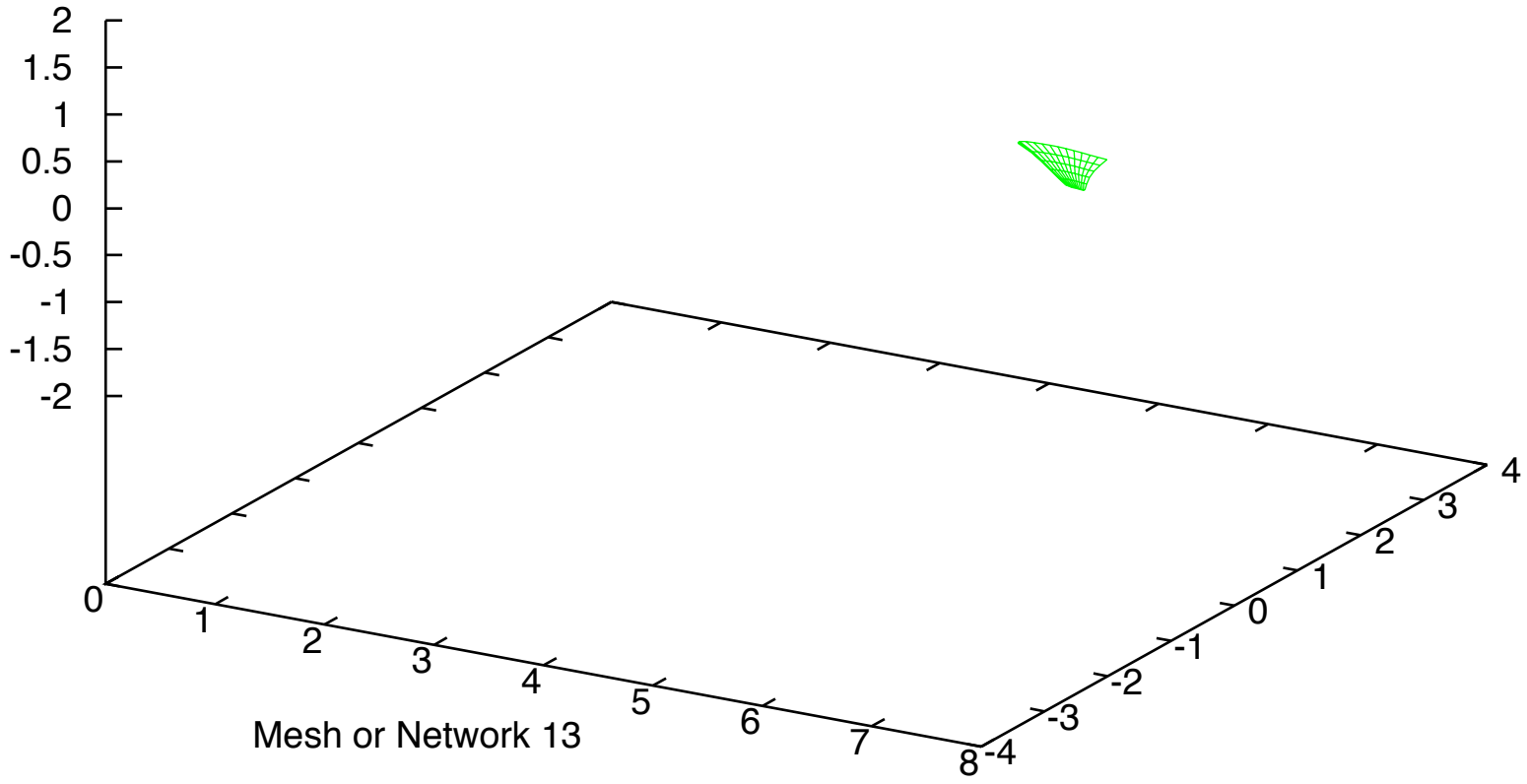
Demo of multiple mesh per file capability - Digitized Blue Whale



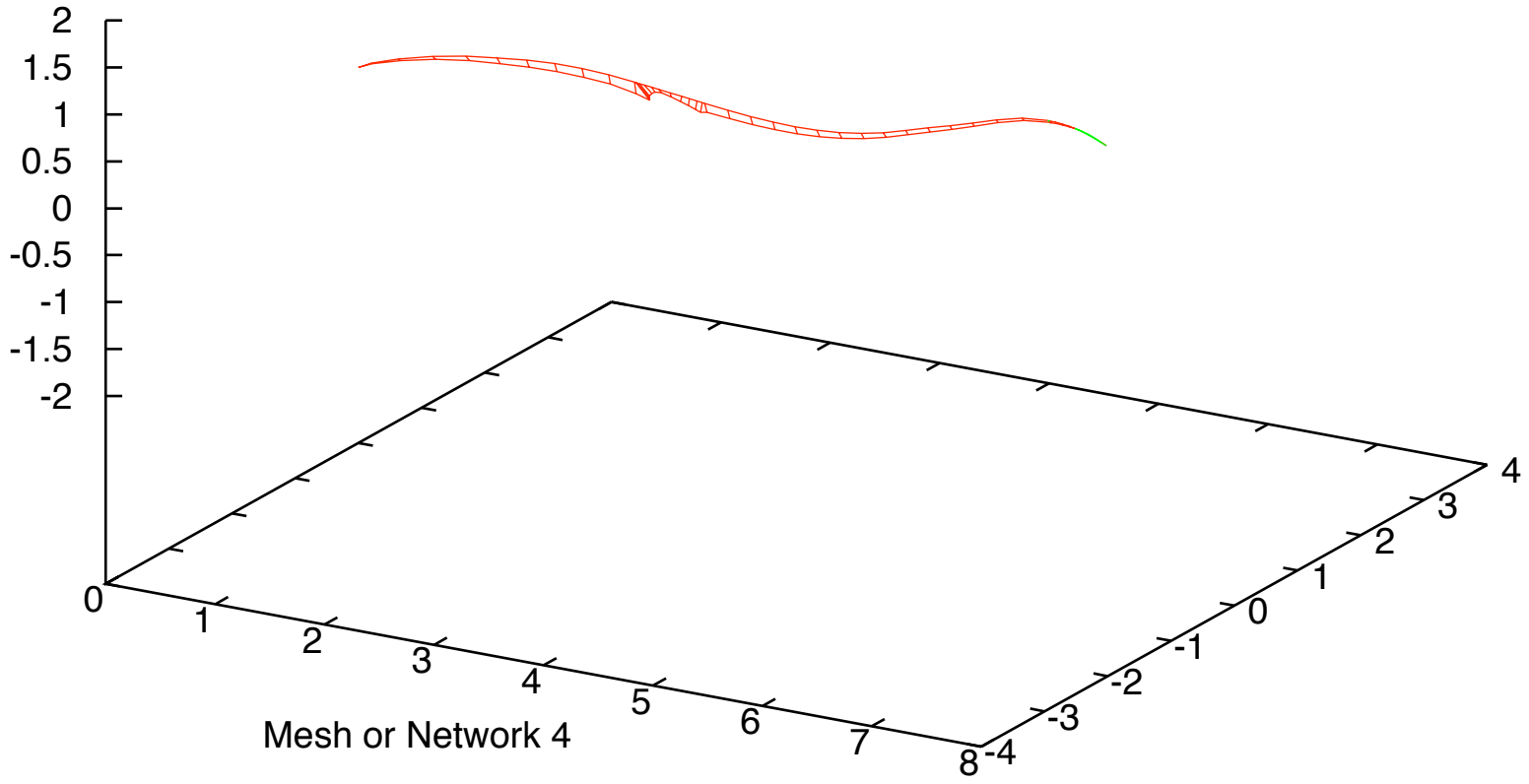
Demo of multiple mesh per file capability - Digitized Blue Whale



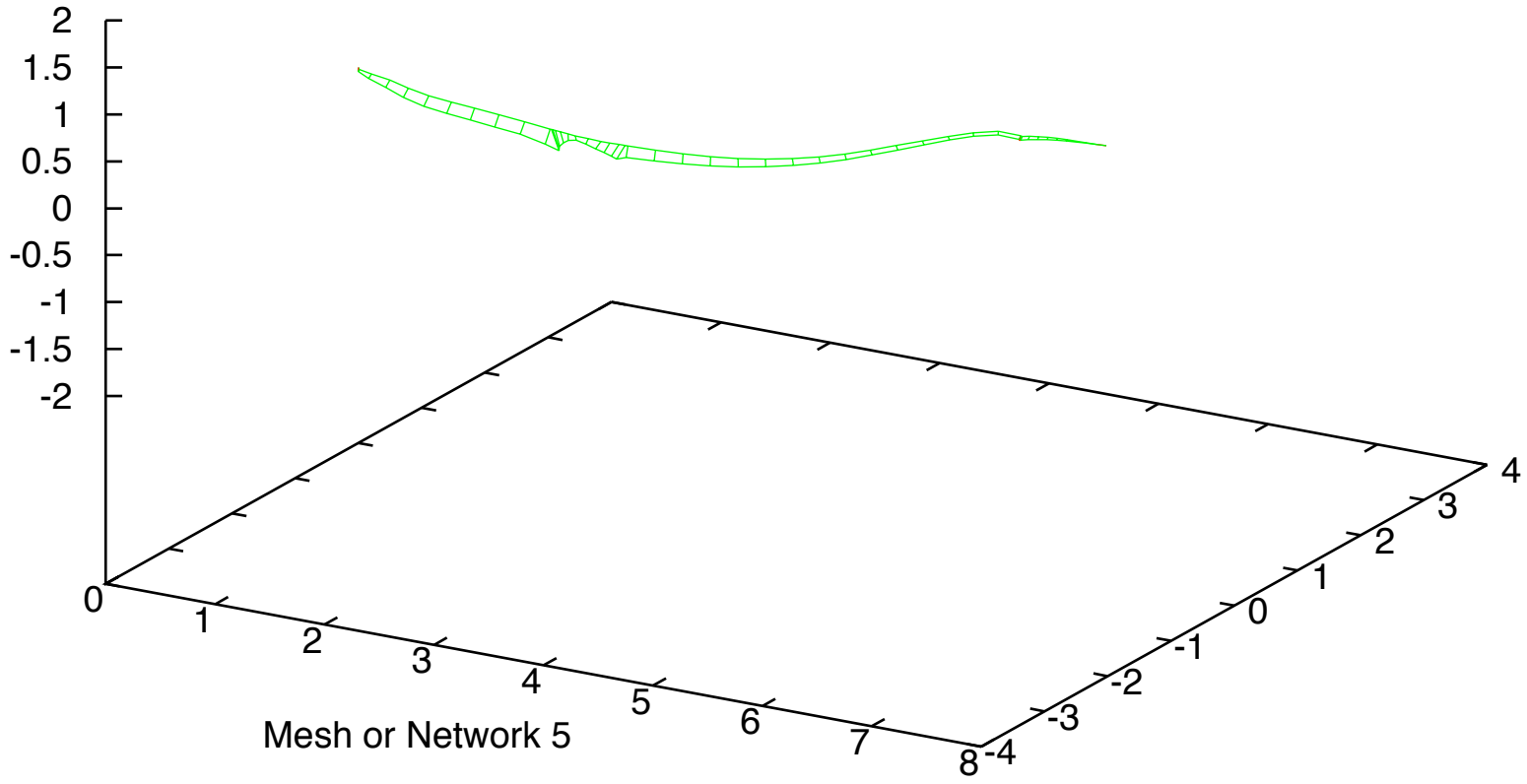
Demo of multiple mesh per file capability - Digitized Blue Whale

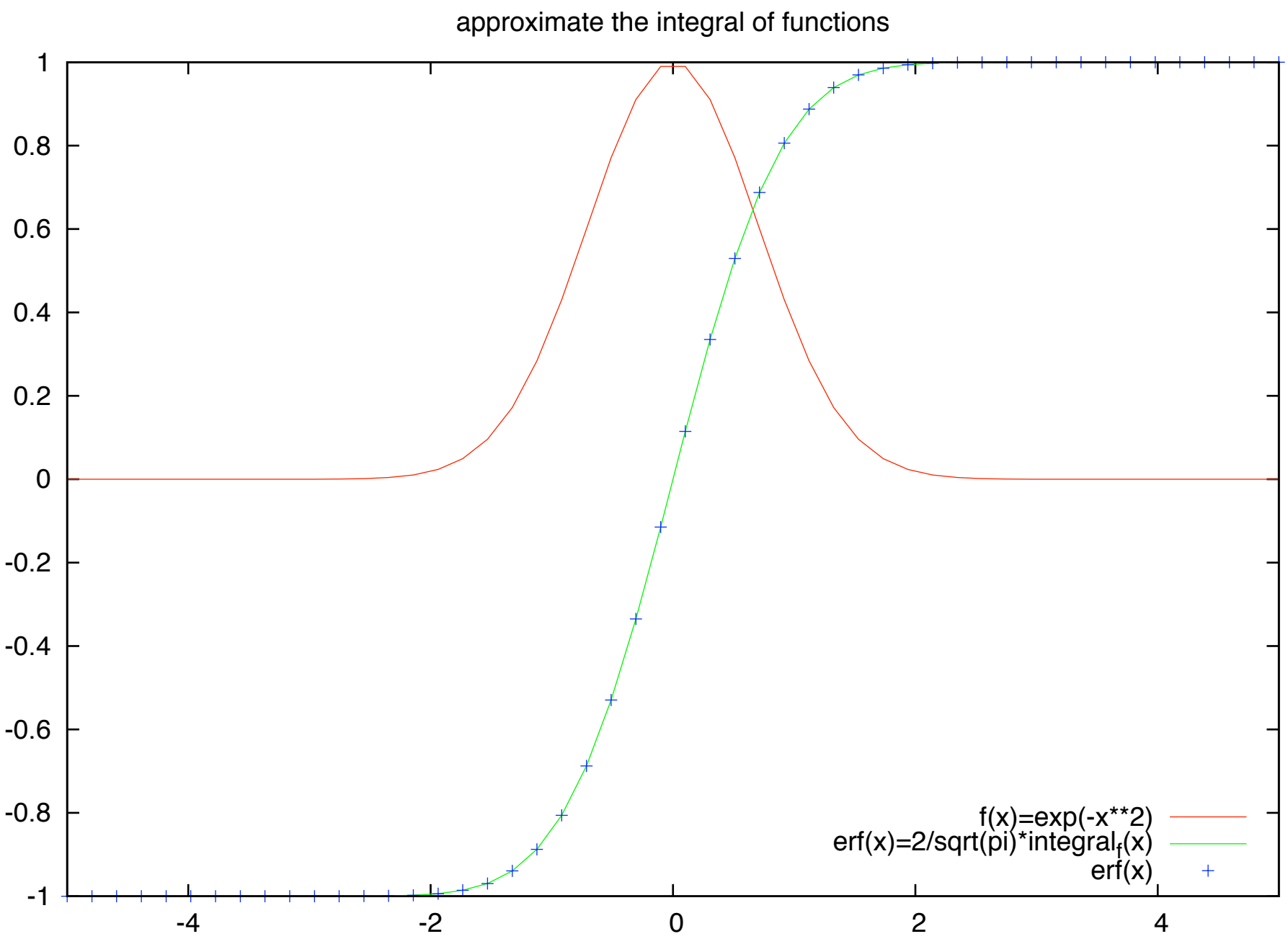


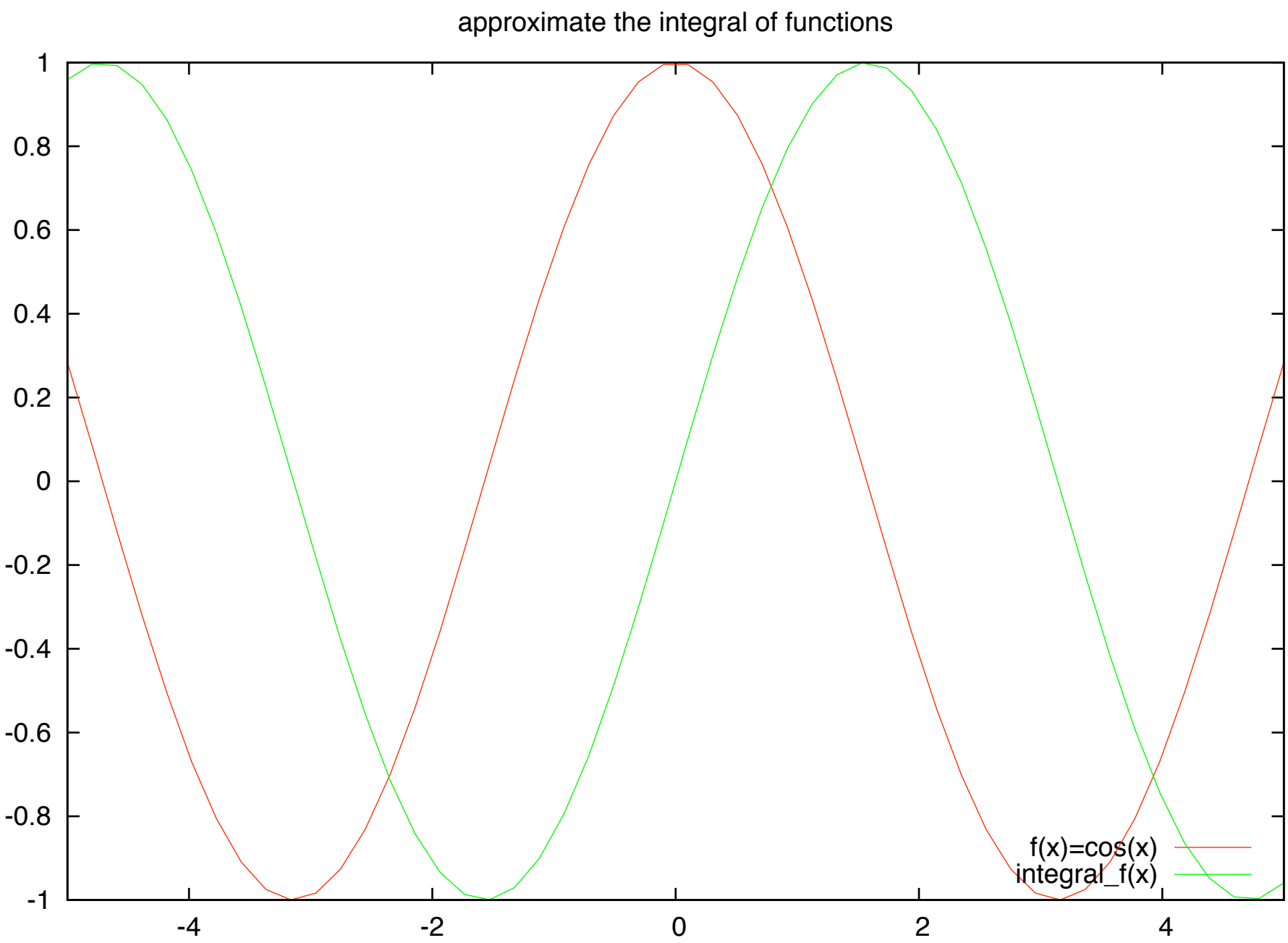
Demo of multiple mesh per file capability - Digitized Blue Whale

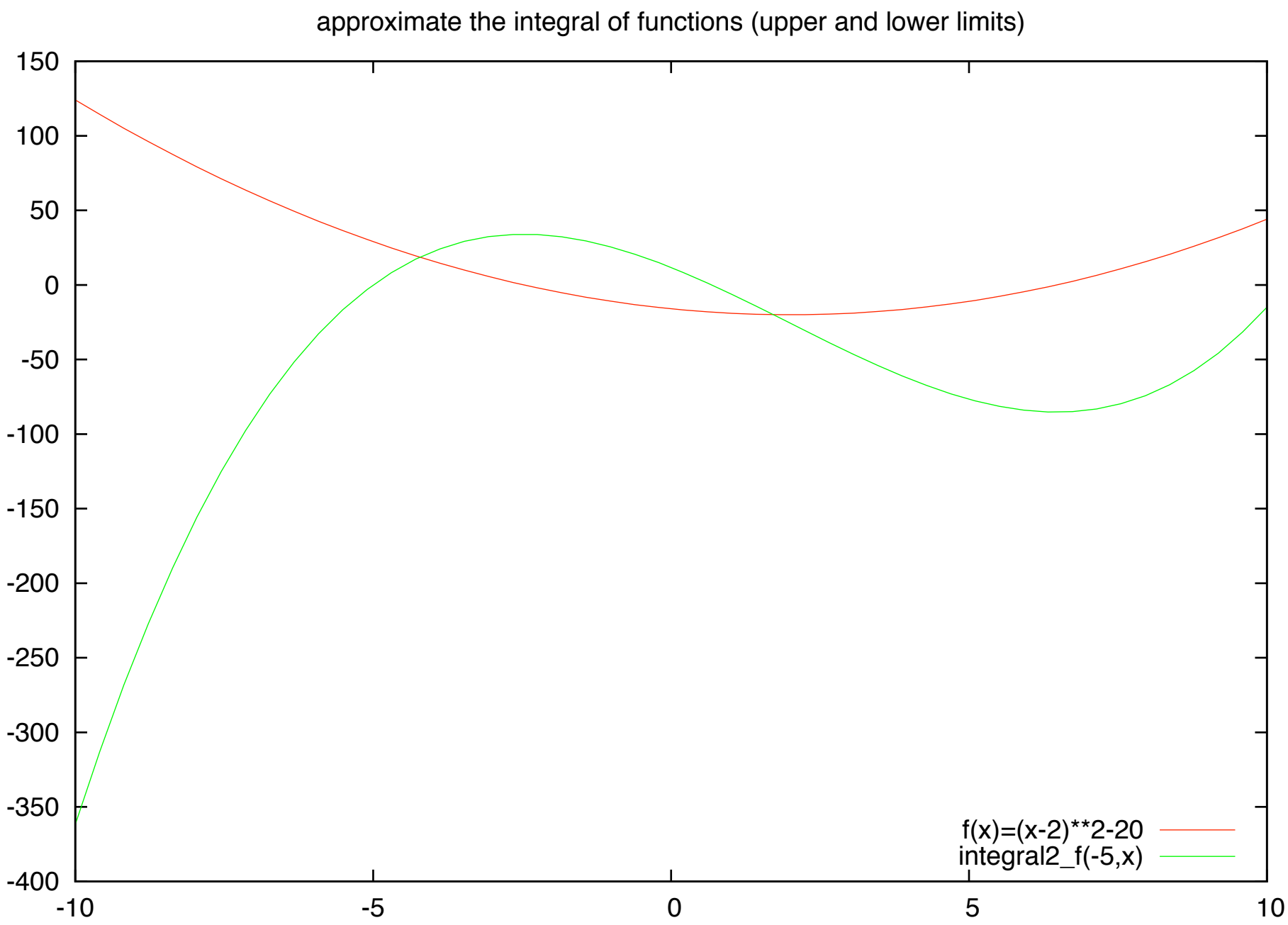


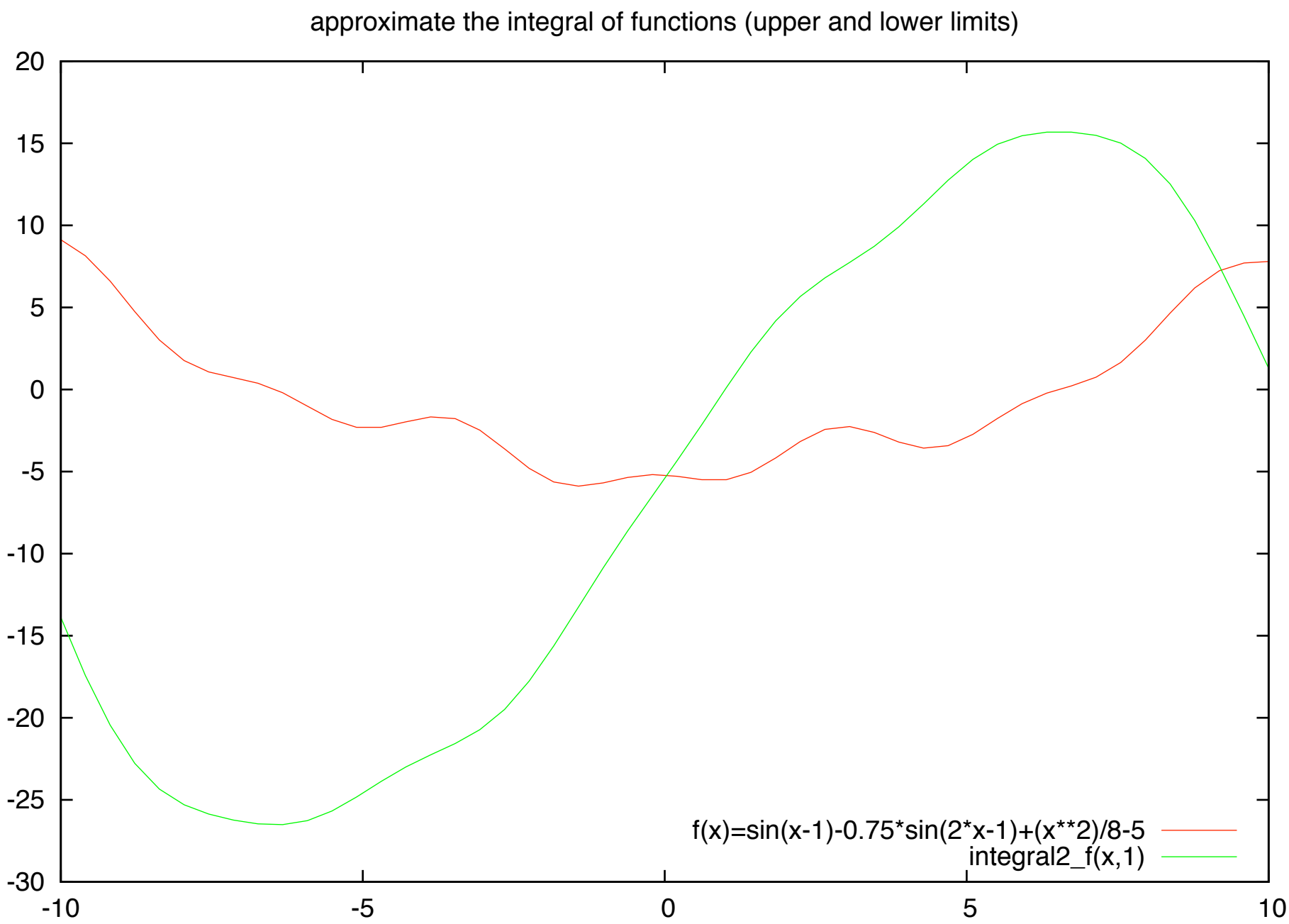
Demo of multiple mesh per file capability - Digitized Blue Whale



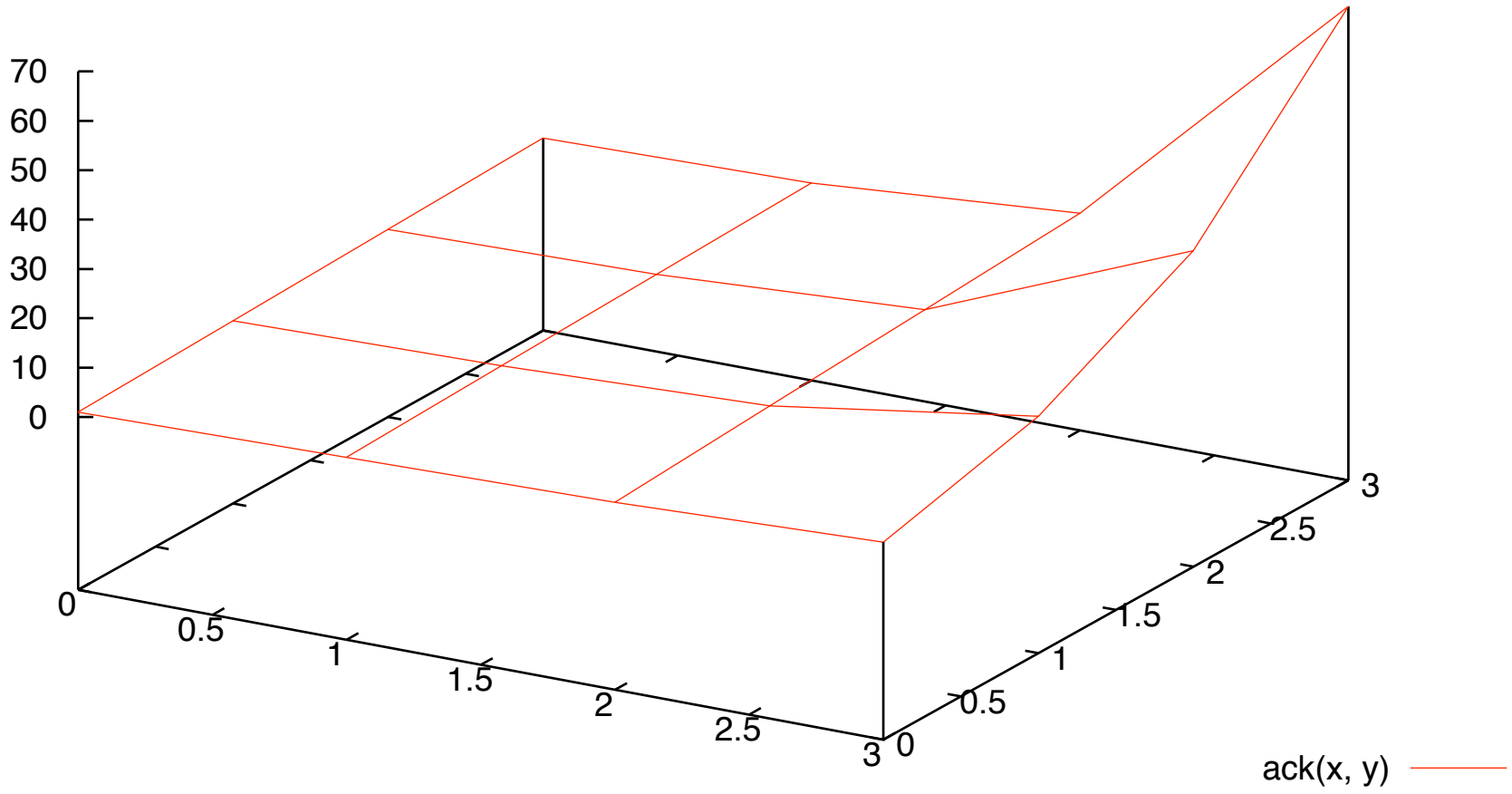

approximate the integral of functions
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-4
-2
0
2
4
f(x)=exp(-x**2)
erf(x)=2/sqrt(pi)*integral_f(x)
erf(x)

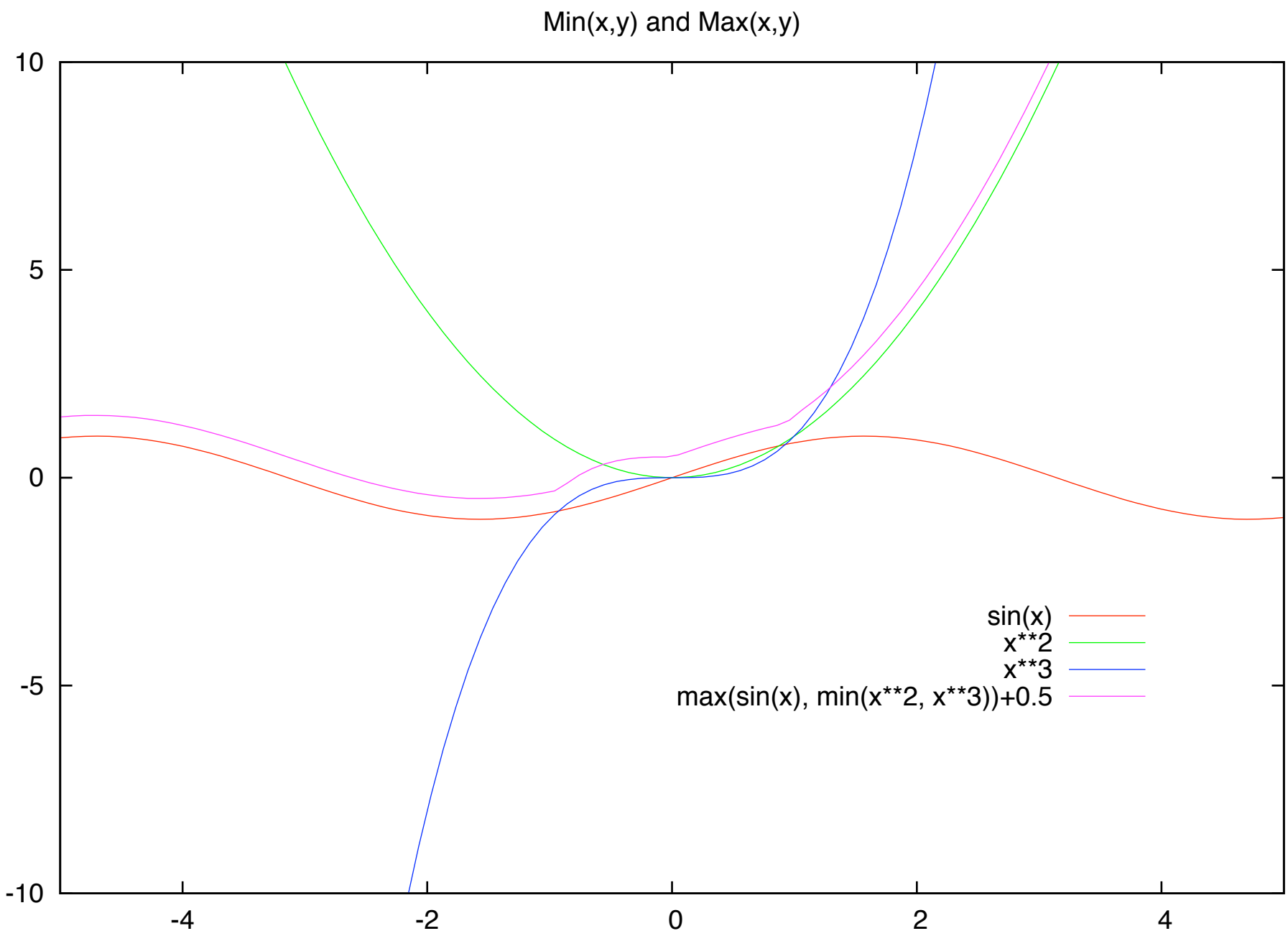

approximate the integral of functions
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-4
-2
0
2
4
f(x)=cos(x)
integral_f(x)

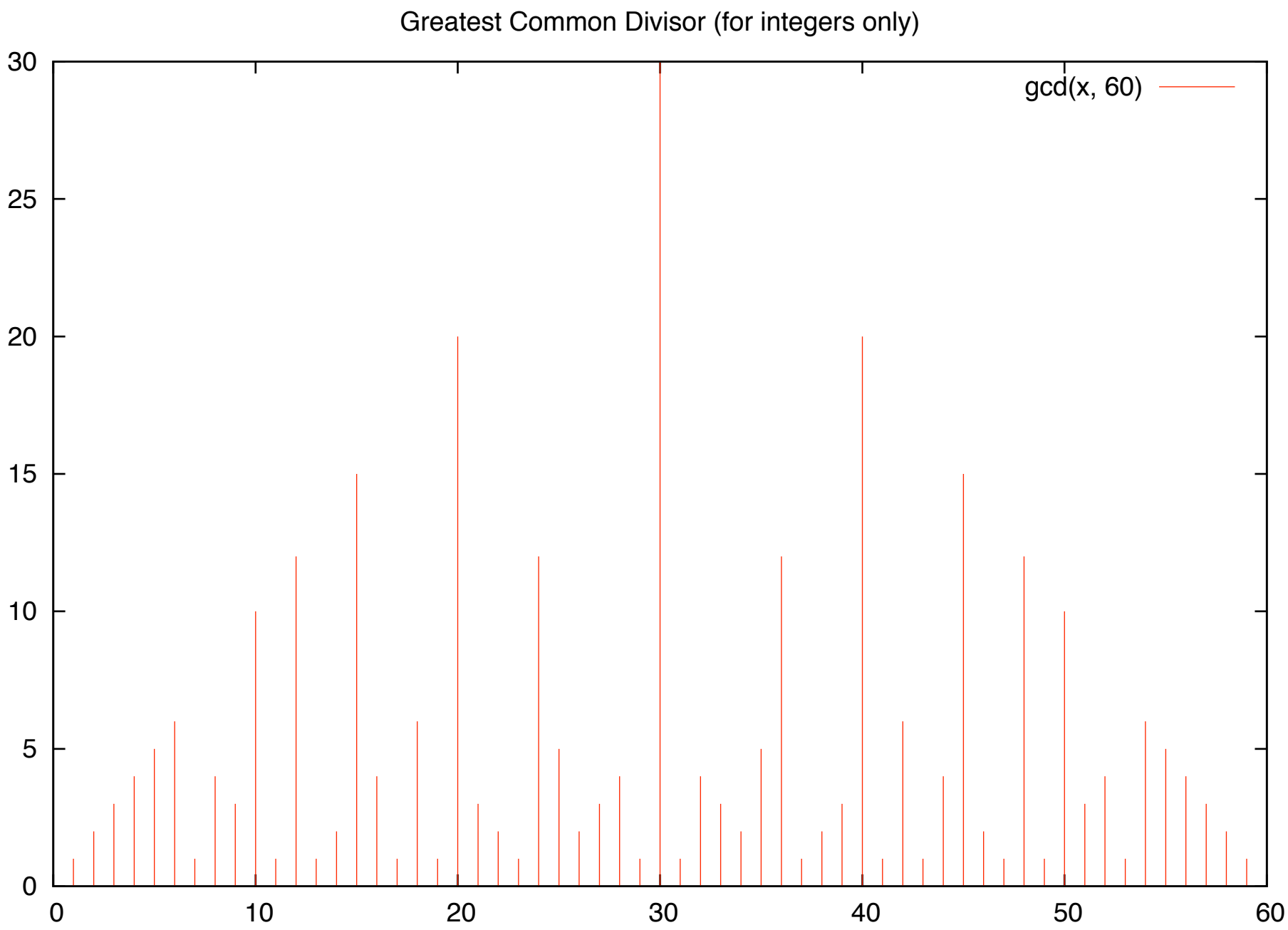

approximate the integral of functions (upper and lower limits)
150
100
50
0
-50
-100
-150
-200
-250
-300
-350
-400
-10
-5
0
5
10
f(x)=(x-2)**2-20
integral2_f(-5,x)

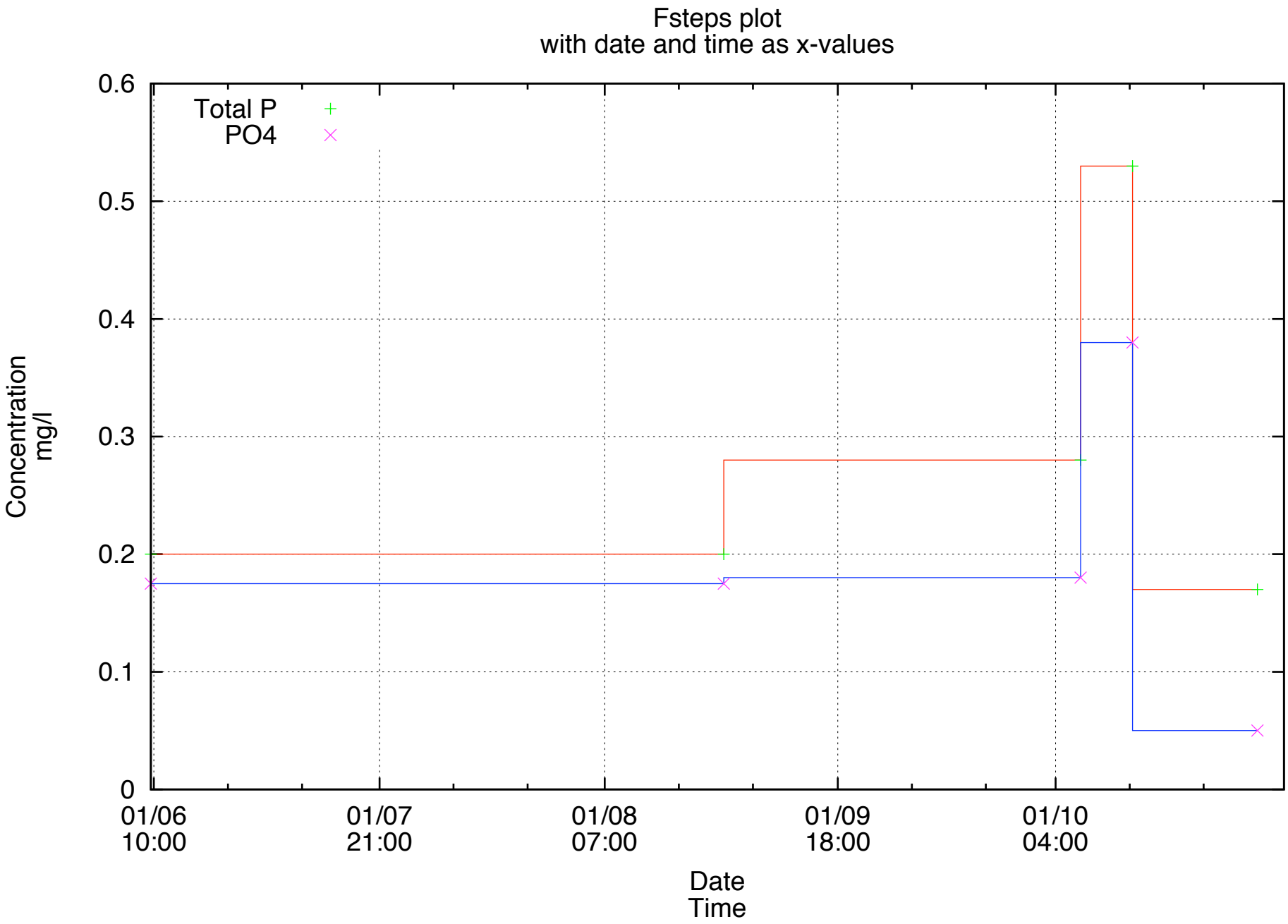

approximate the integral of functions (upper and lower limits)
20
15
10
5
0
-5
-10
-15
-20
-25
-30
-10
-5
0
5
10
f(x)=sin(x-1)-0.75*sin(2*x-1)+(x**2)/8-5
integral2_f(x,1)

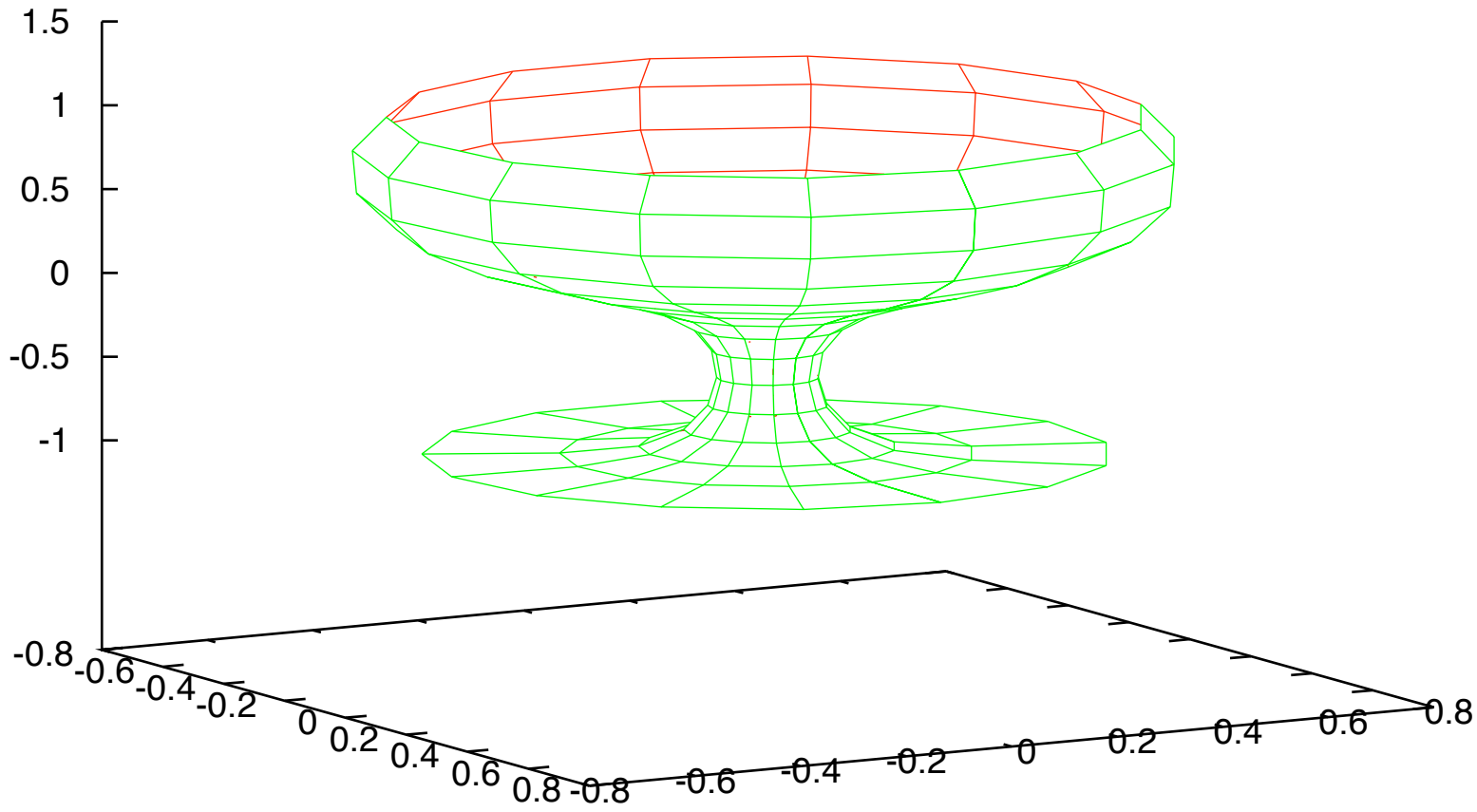
Plot of the ackermann function

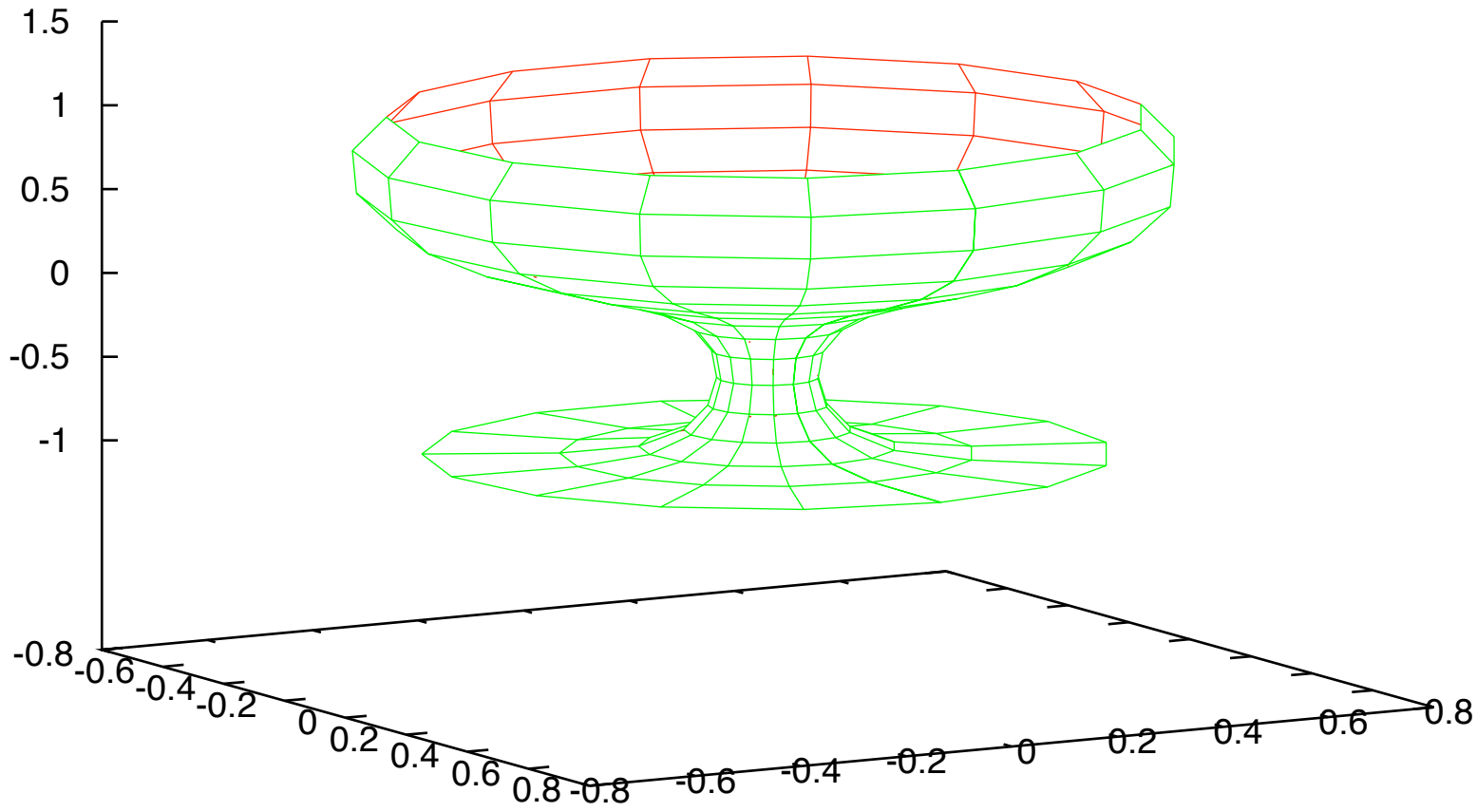


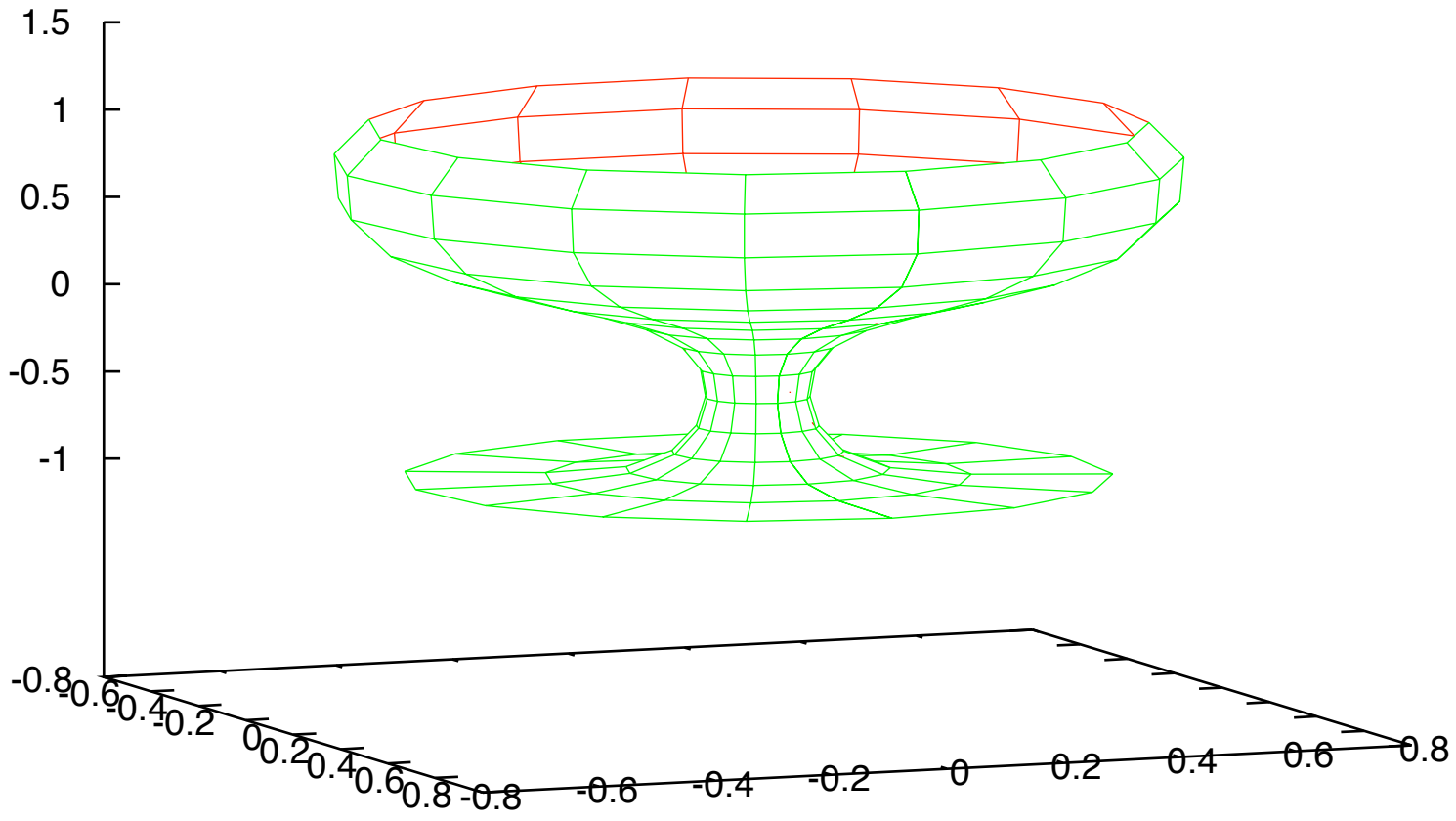

Min(x,y) and Max(x,y)
10
5
0
-5
-10
-4
-2
0
2
4
sin(x)
x**2
x**3
max(sin(x), min(x**2, x**3))+0.5

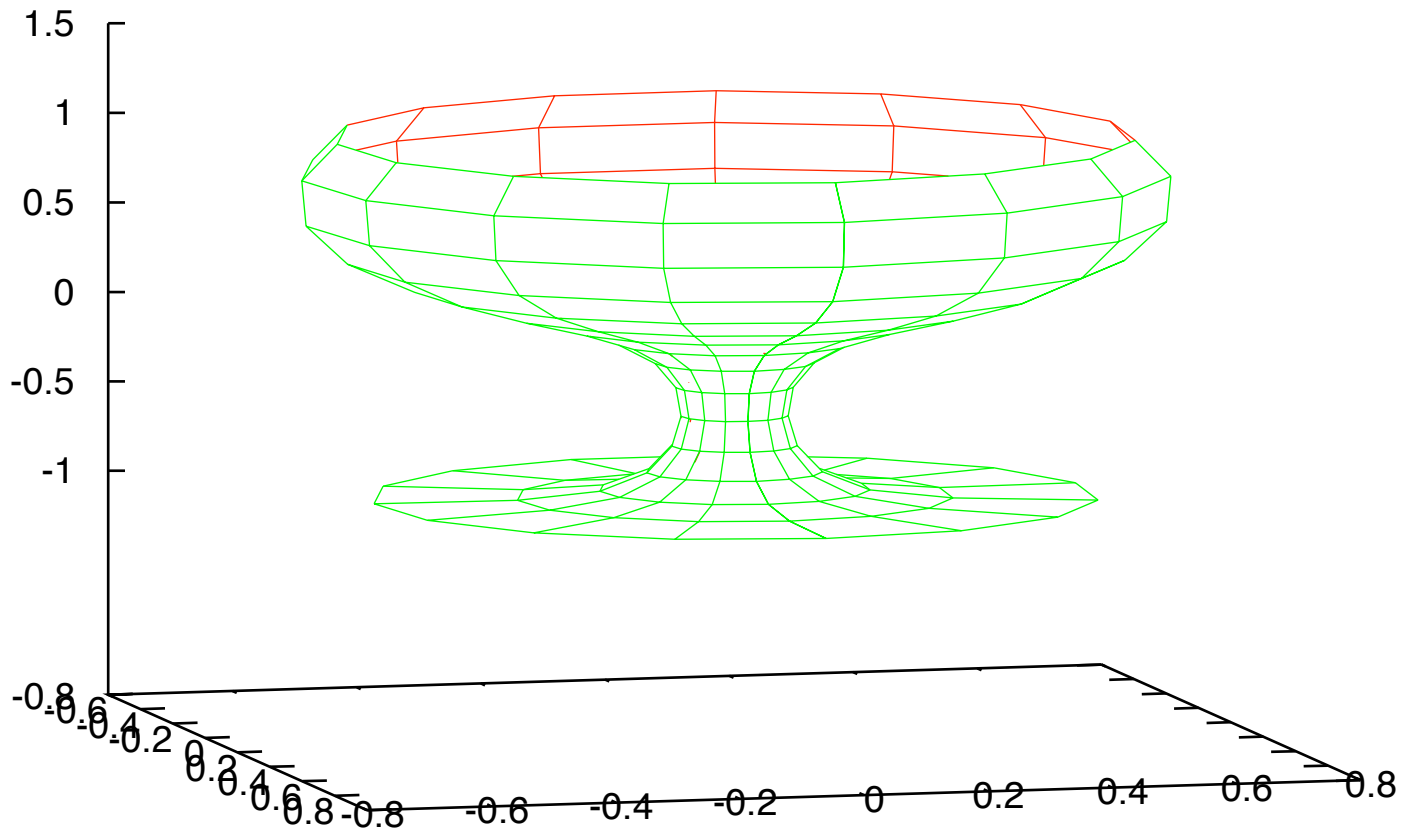

Greatest Common Divisor (for integers only)
gcd(x, 60)
30
25
20
15
10
5
0
0
10
20
30
40
50
60

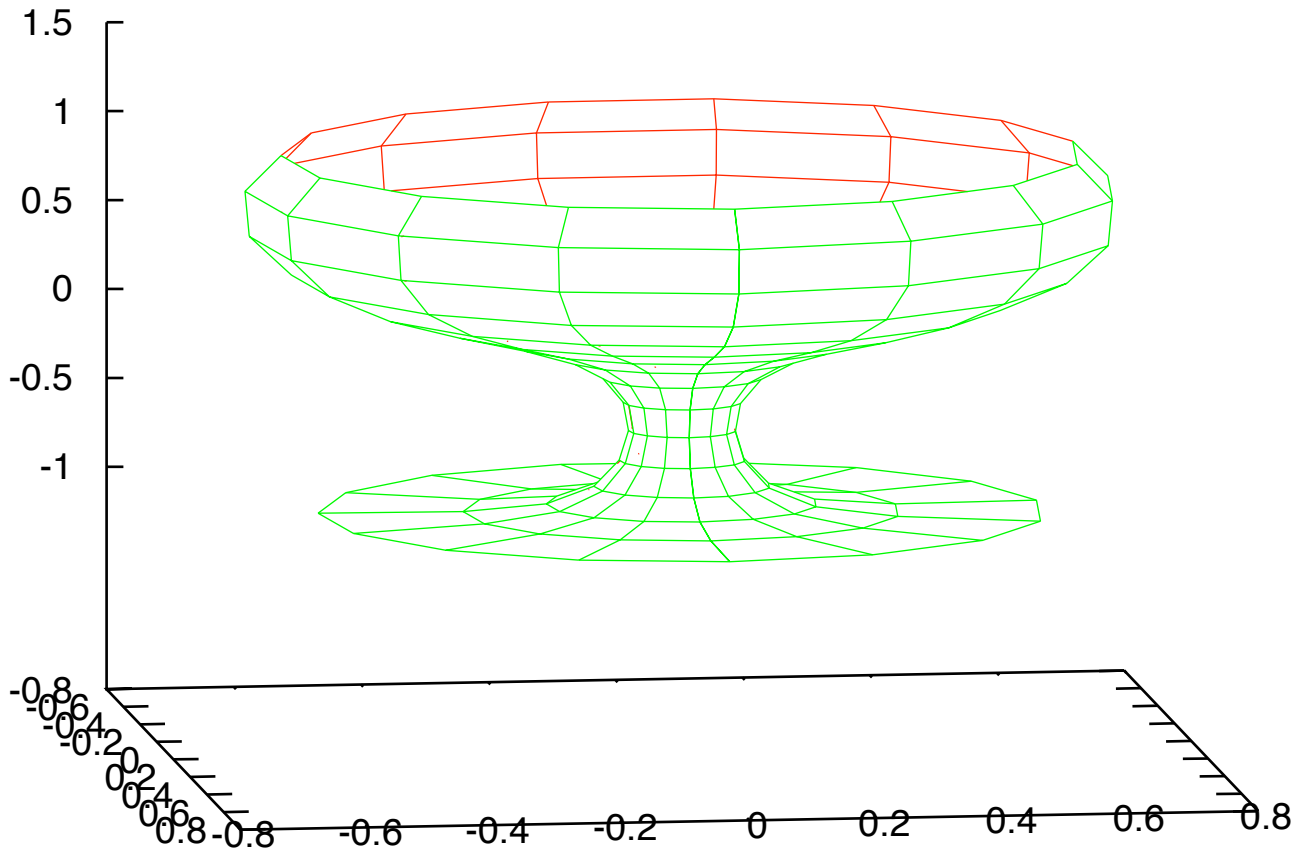

Fsteps plot
with date and time as x-values
Total P
PO4
Concentration
mg/l
0.6
0.5
0.4
0.3
0.2
0.1
0
01/06
10:00
01/07
21:00
01/08
07:00
01/09
18:00
01/10
04:00
Date
Time

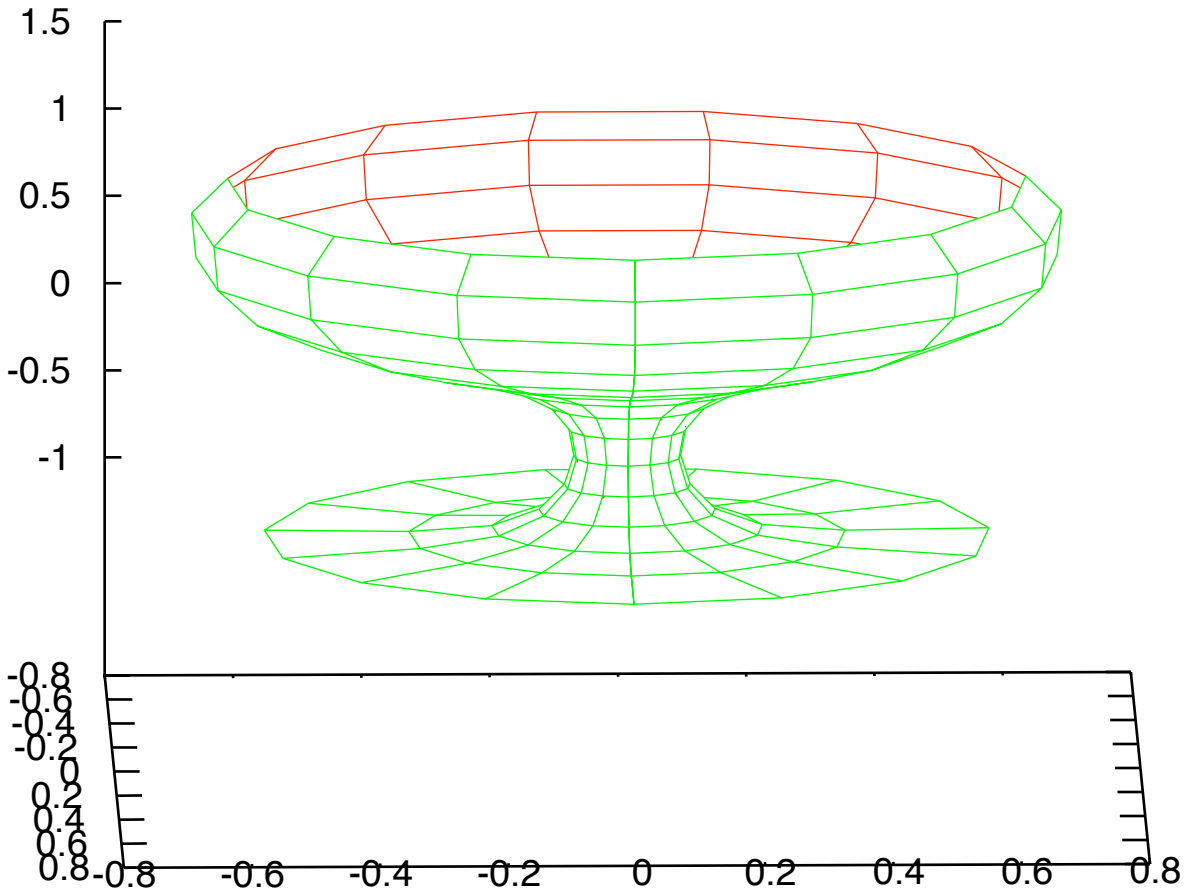

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

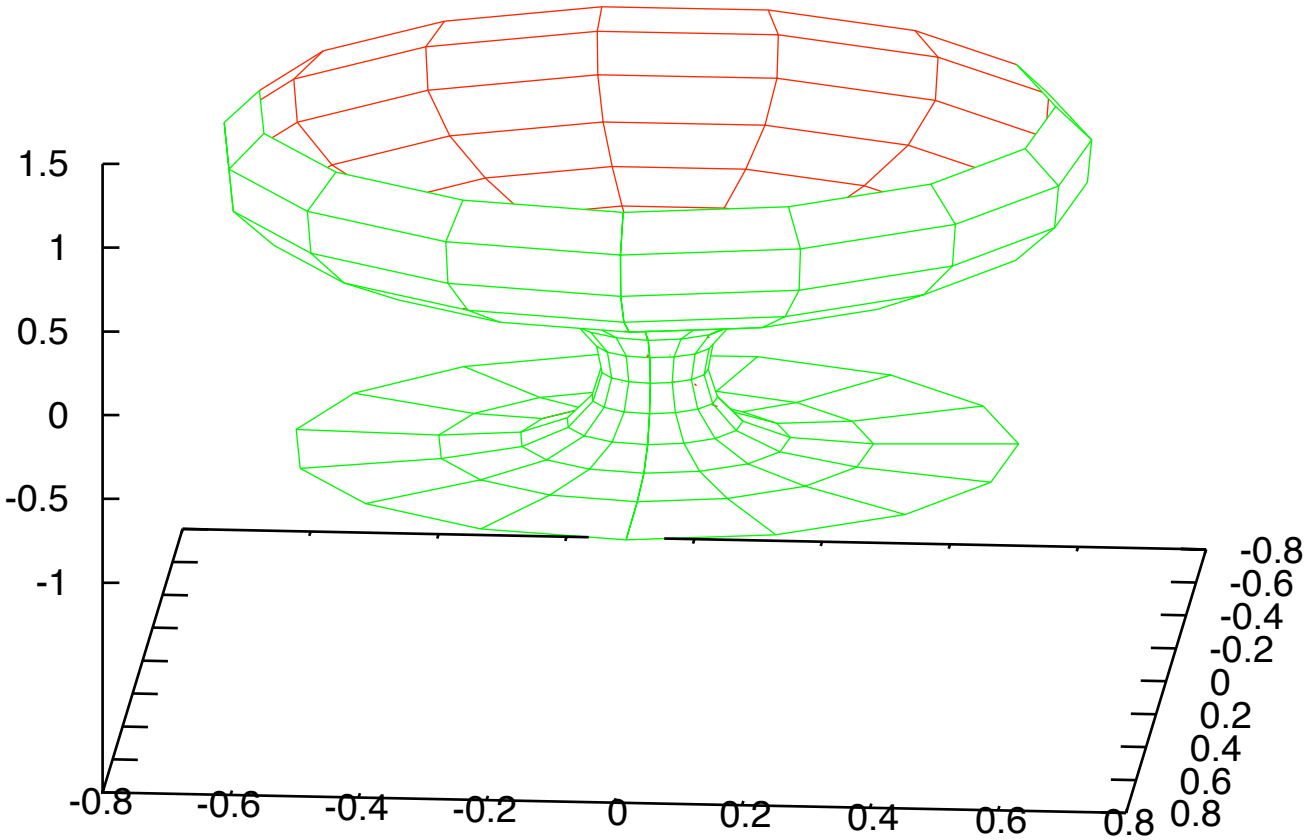

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

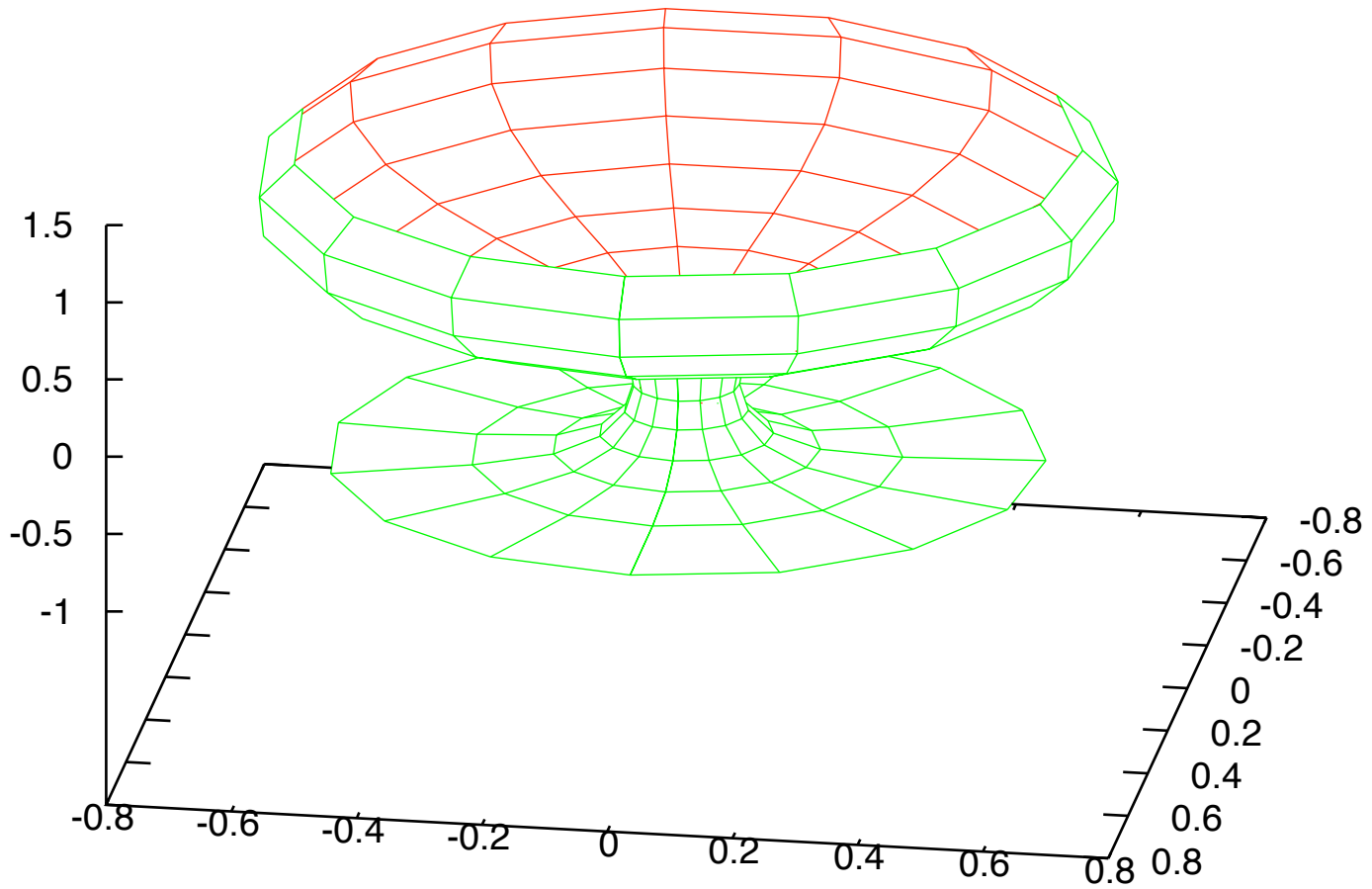

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

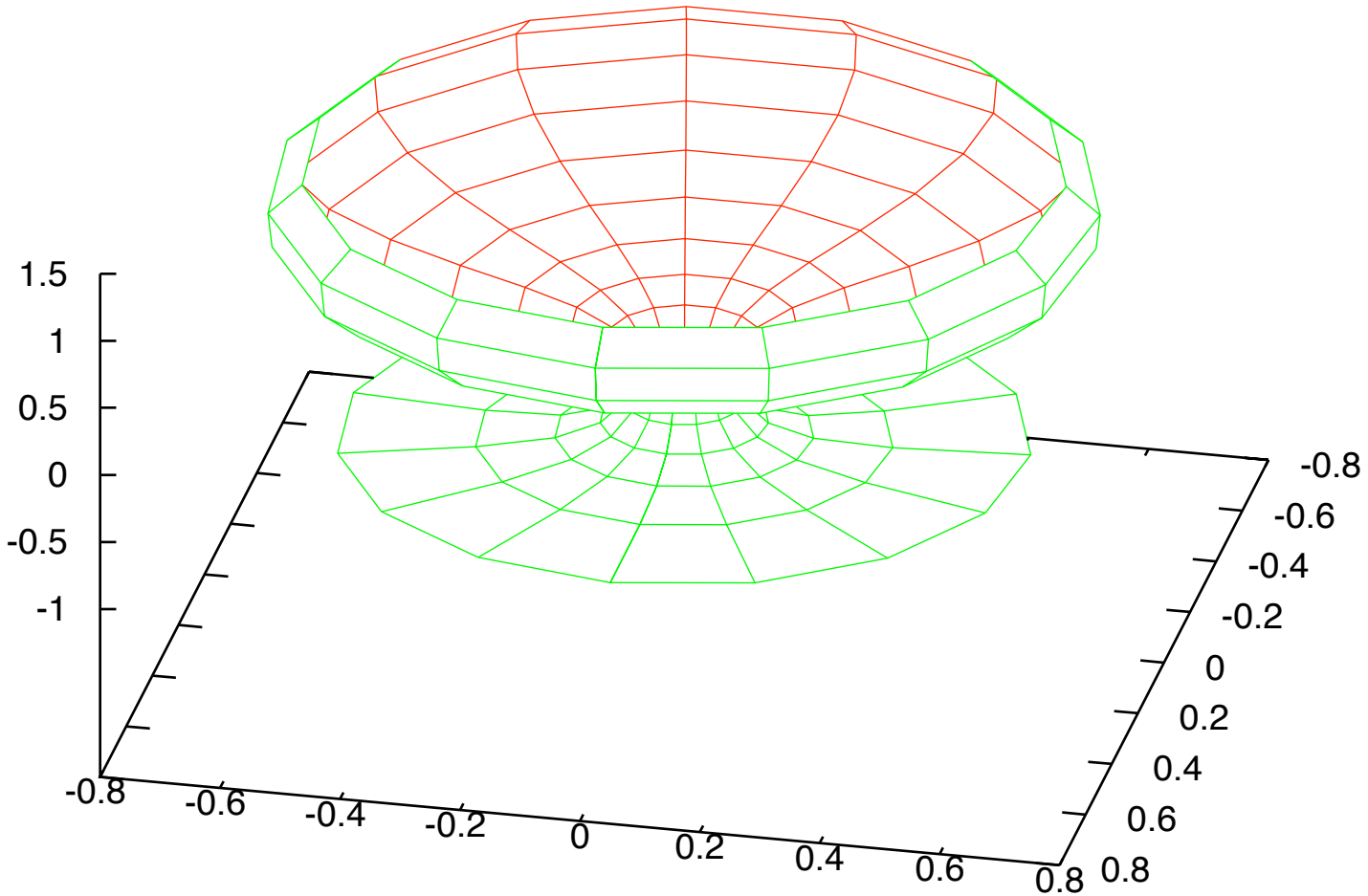

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

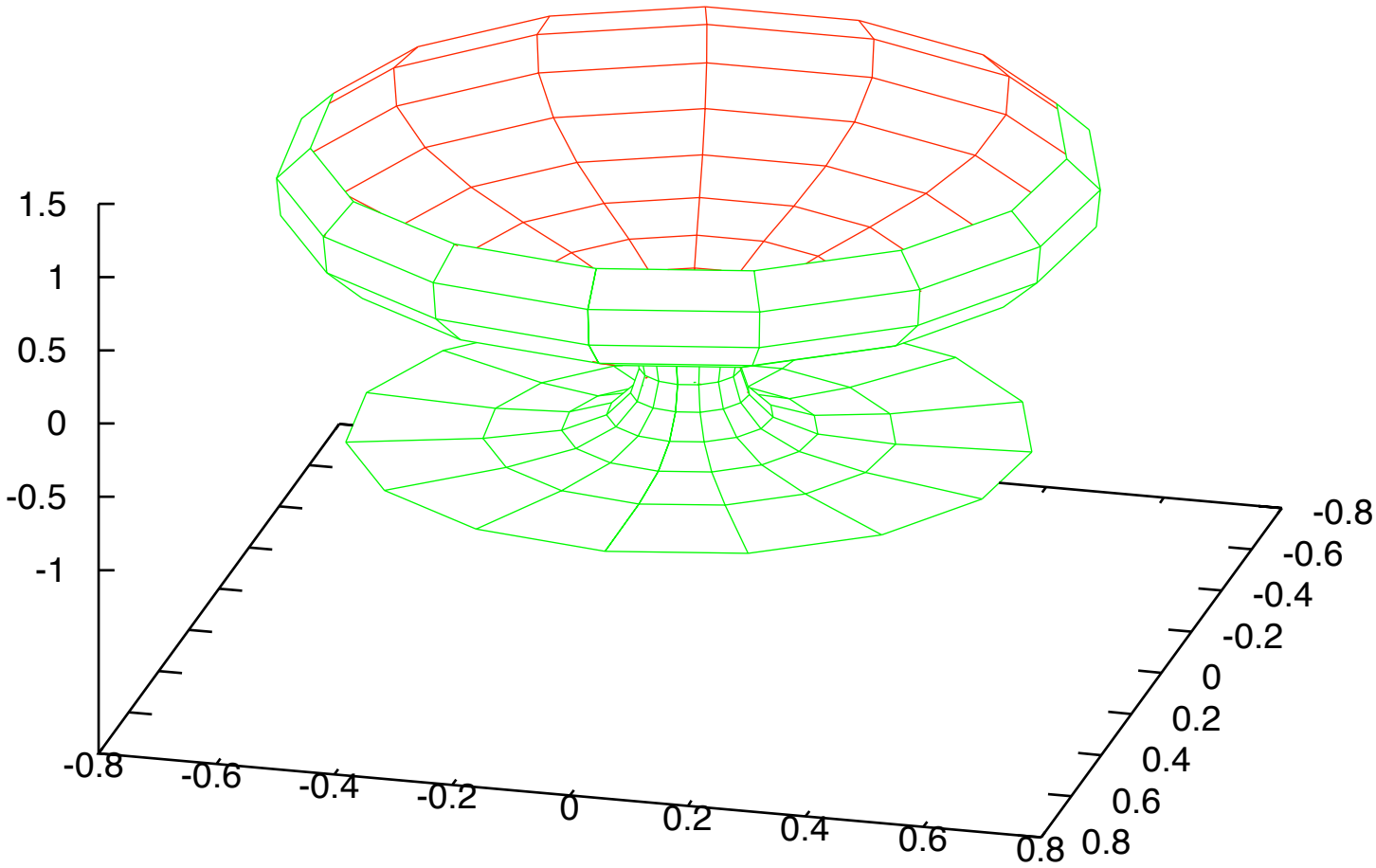

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

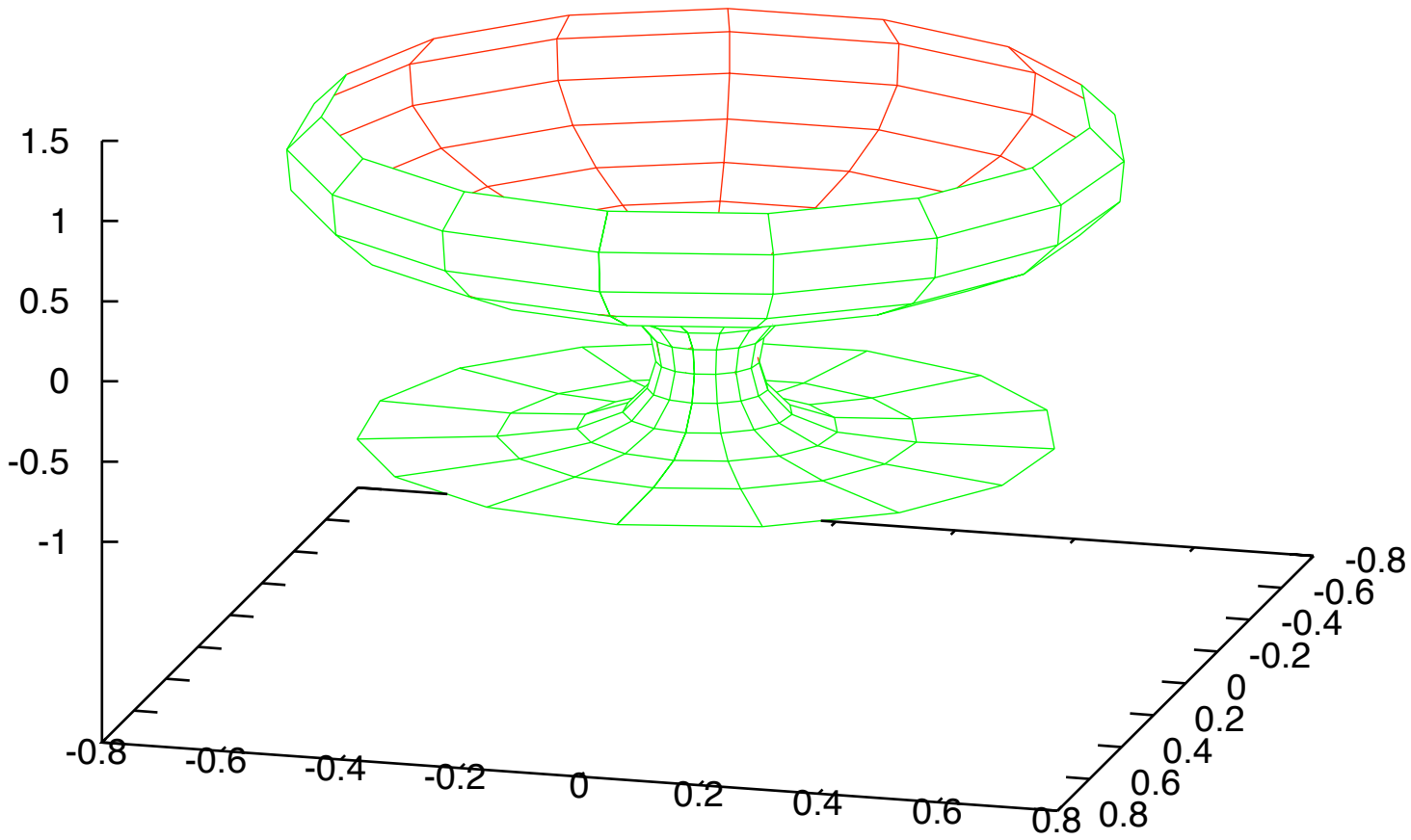

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

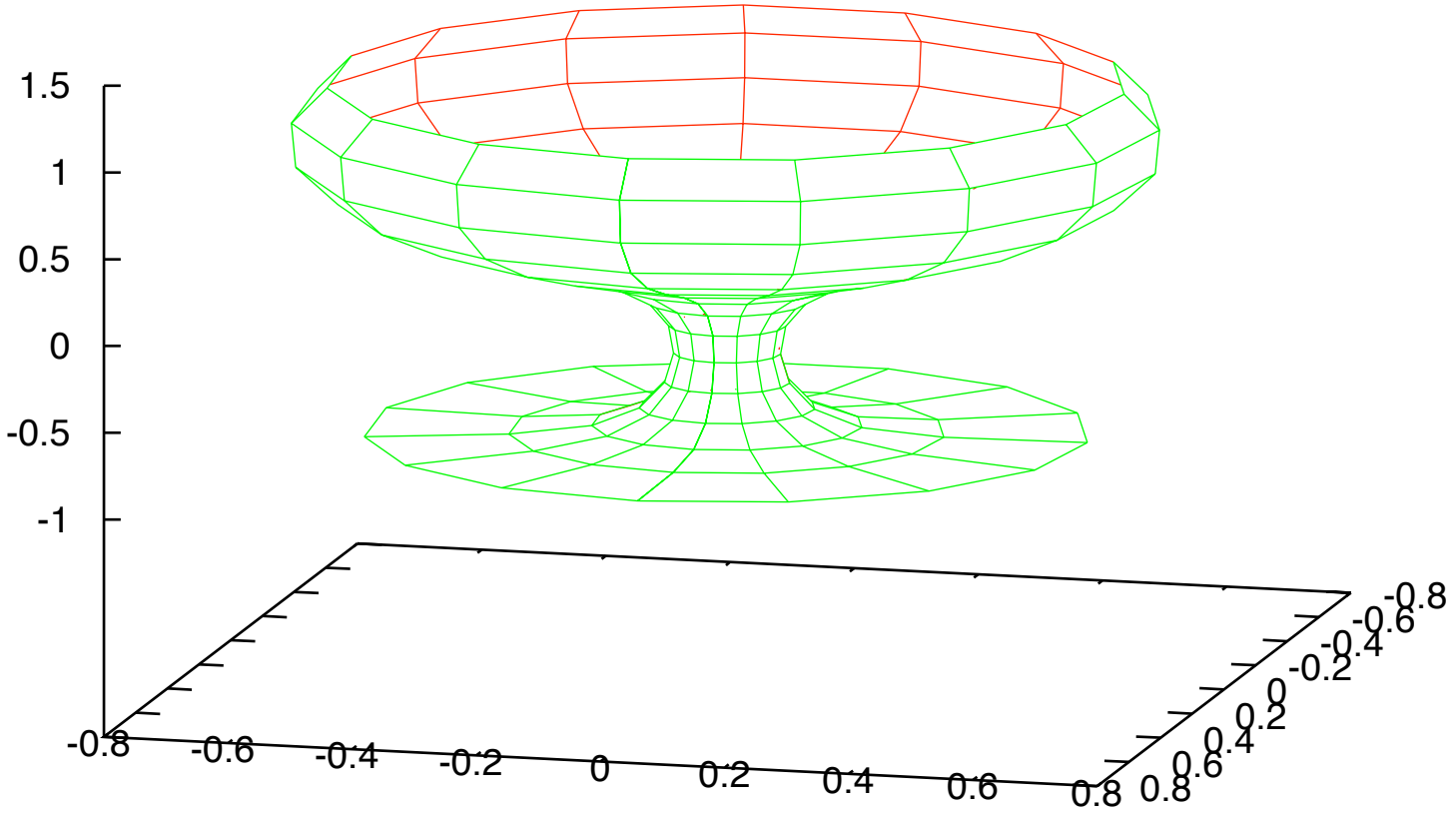

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

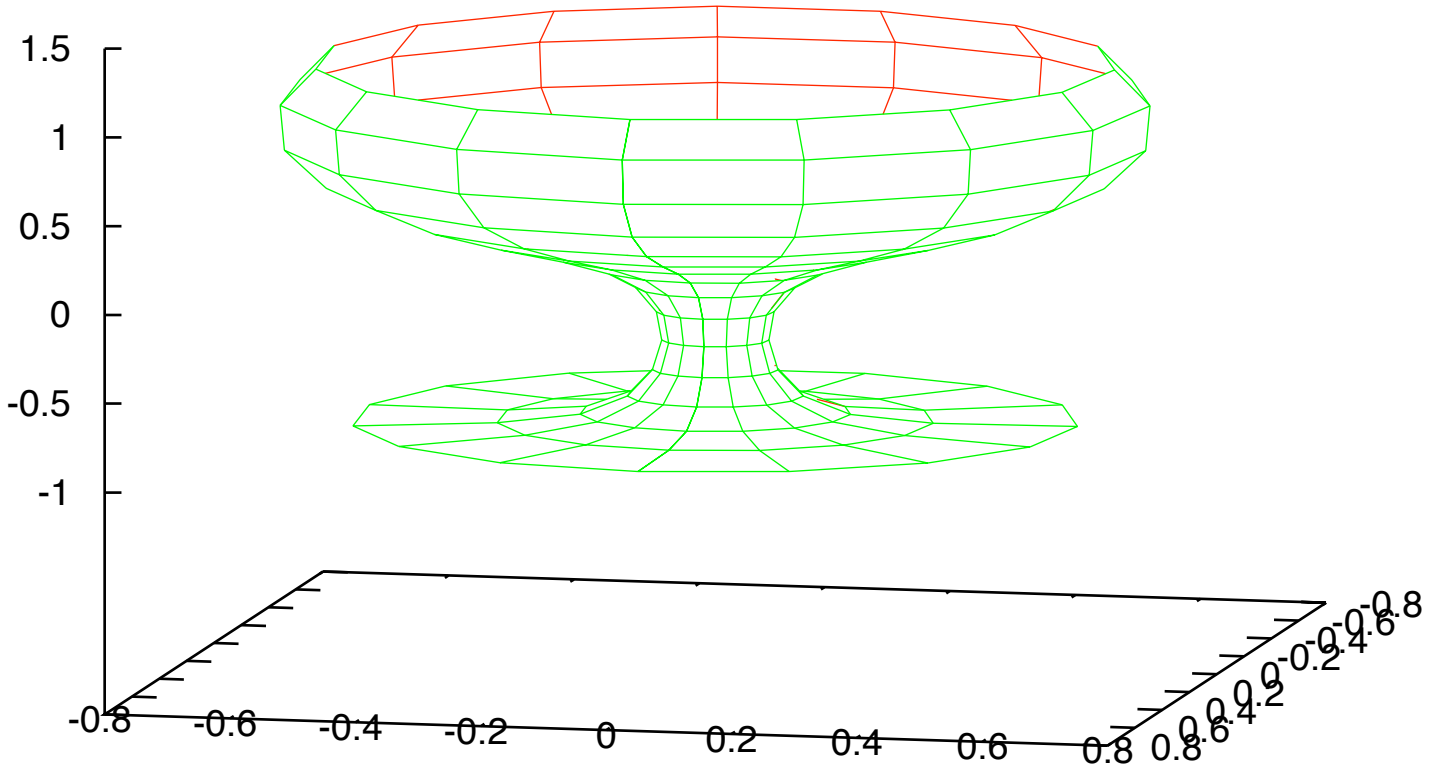

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

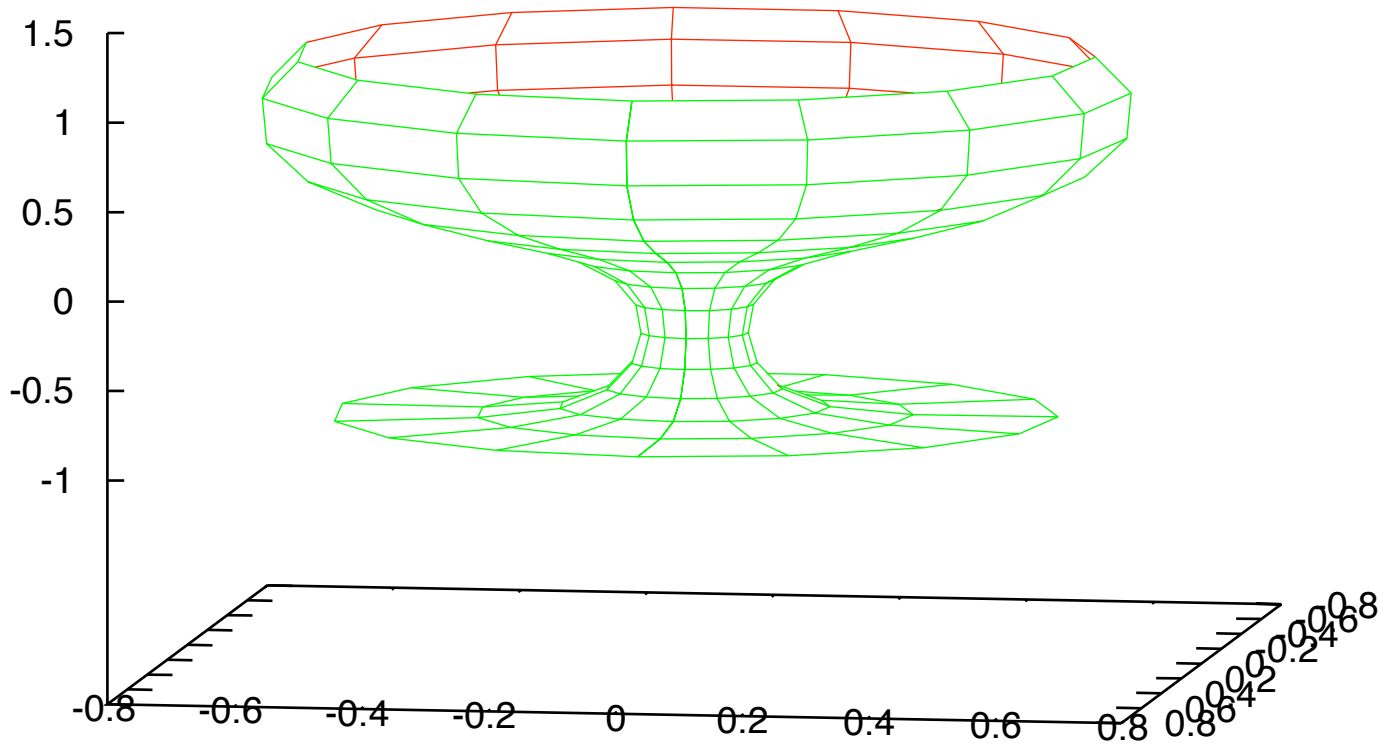

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

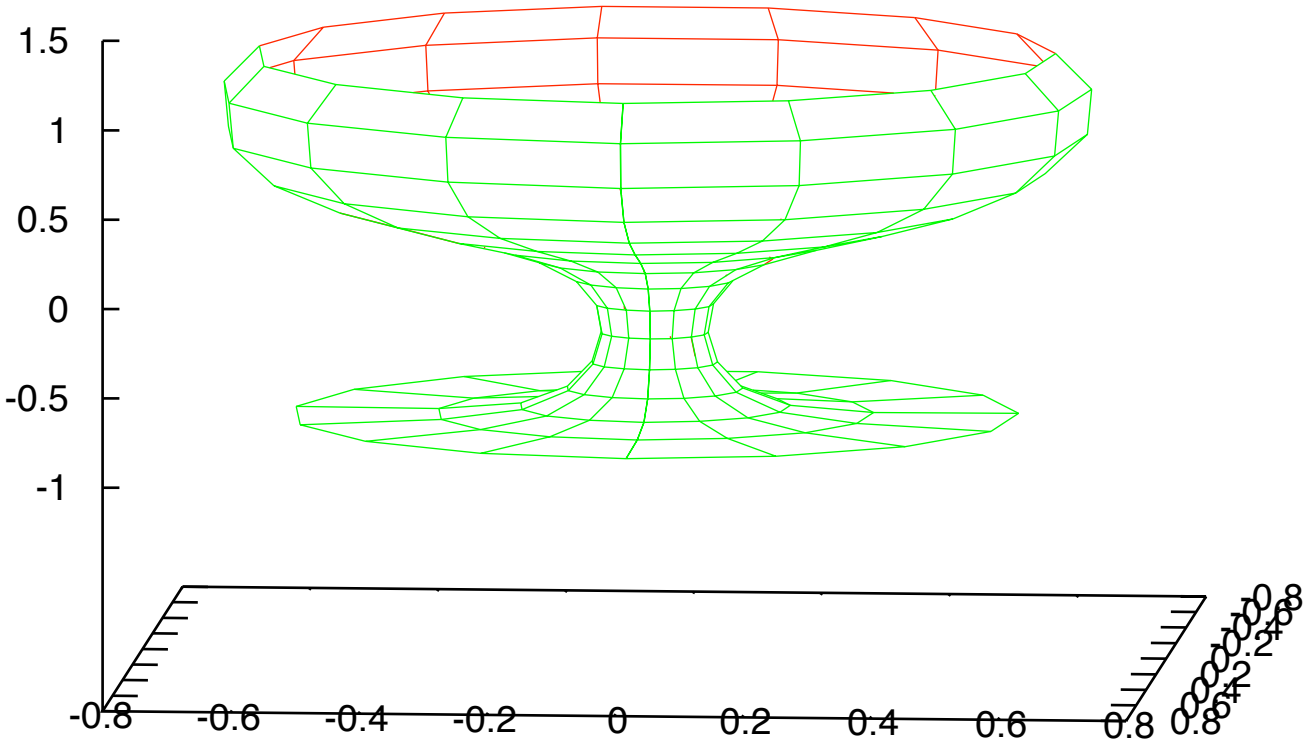

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

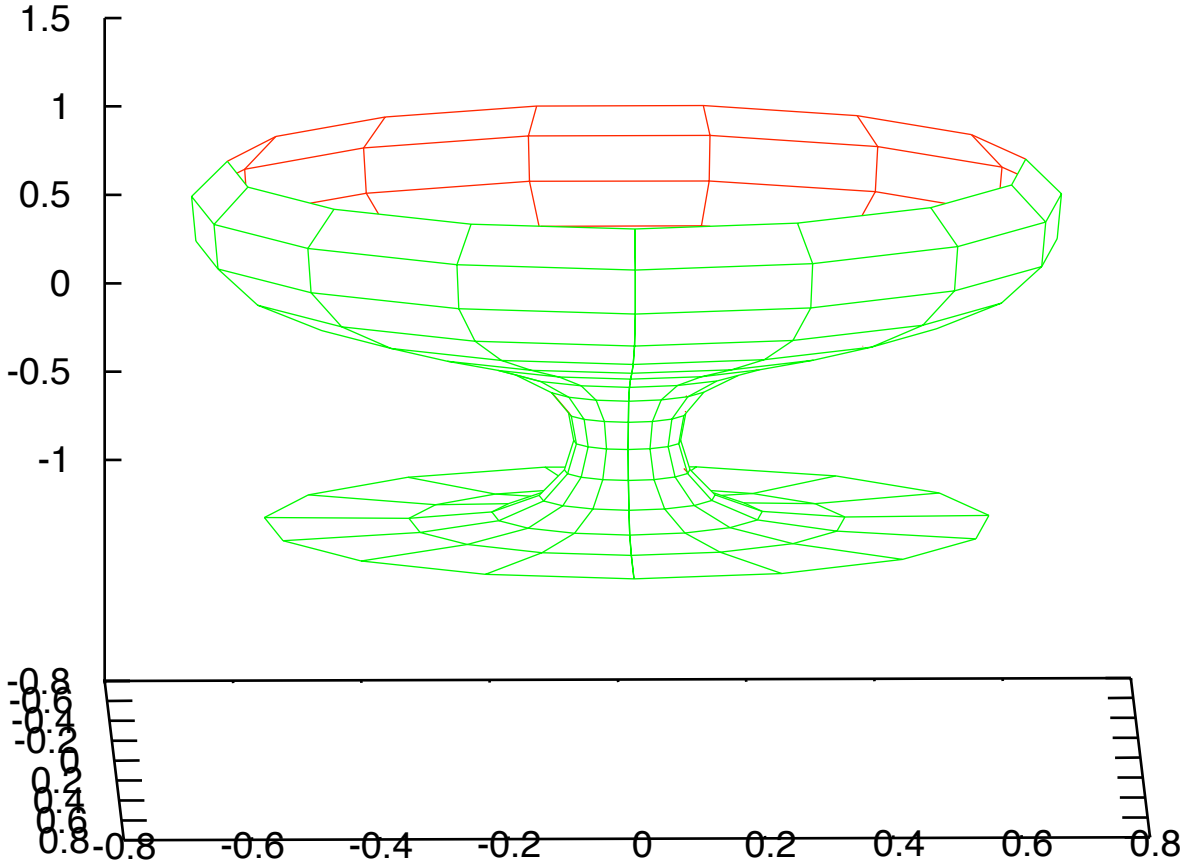

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

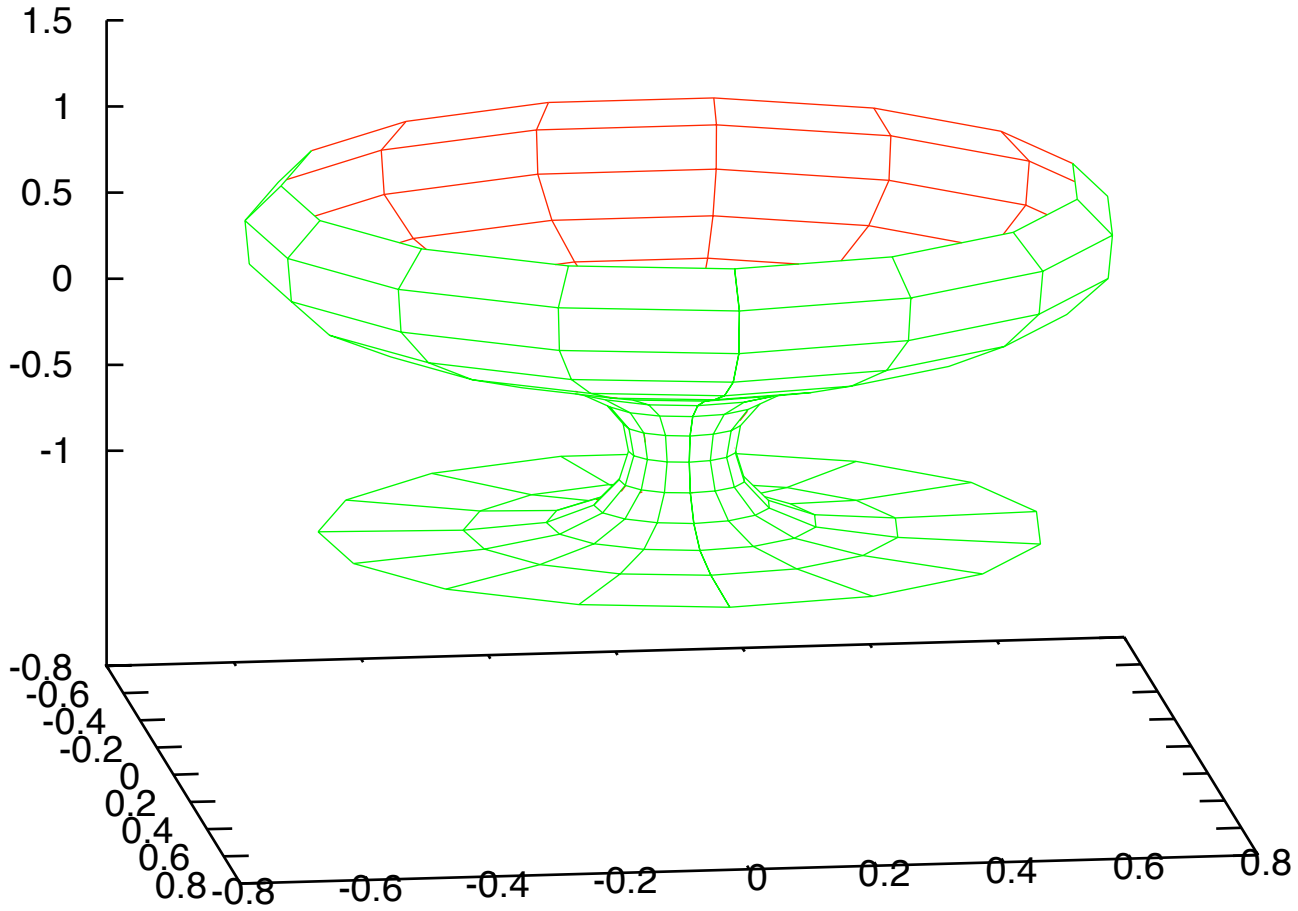

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

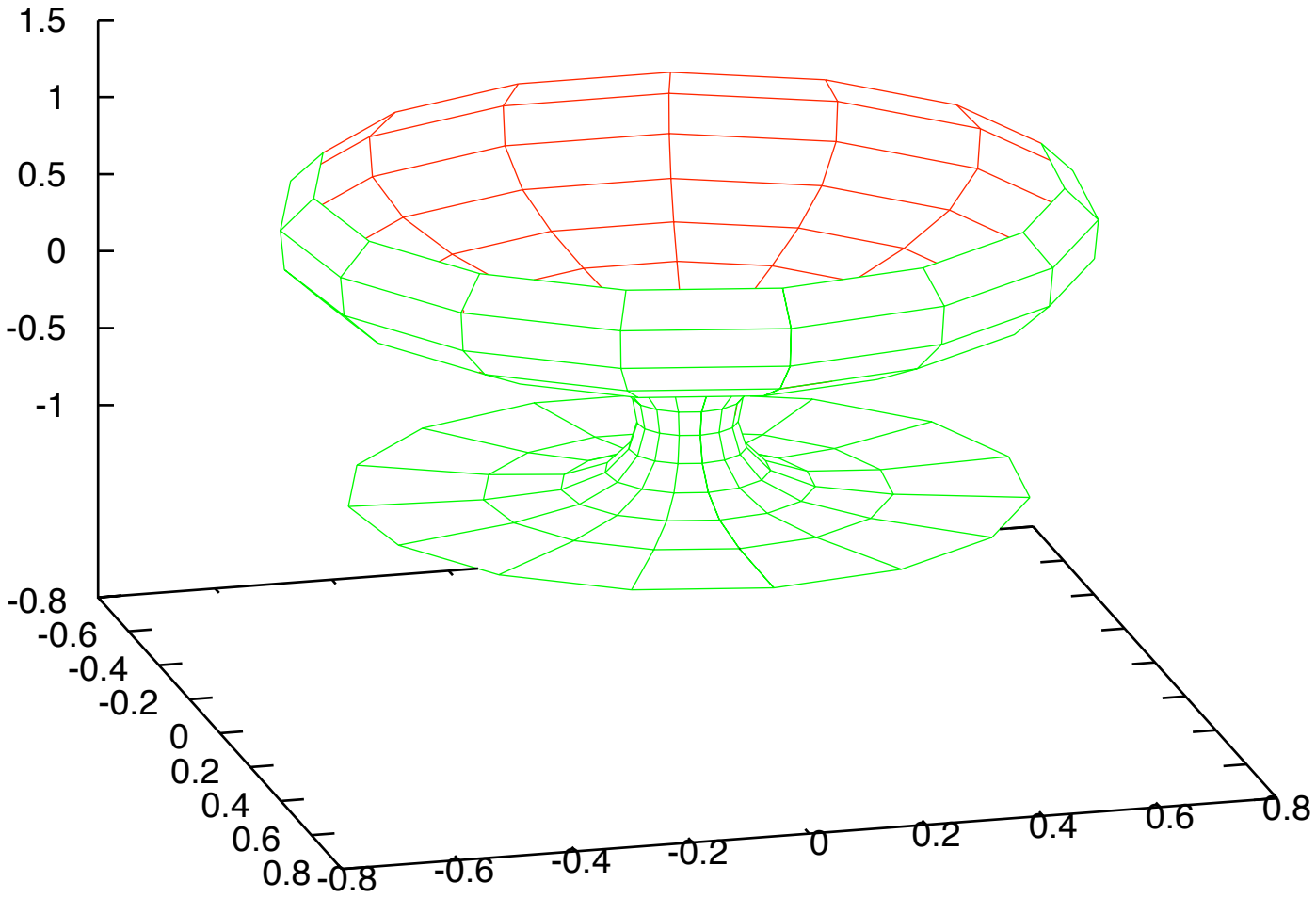

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8

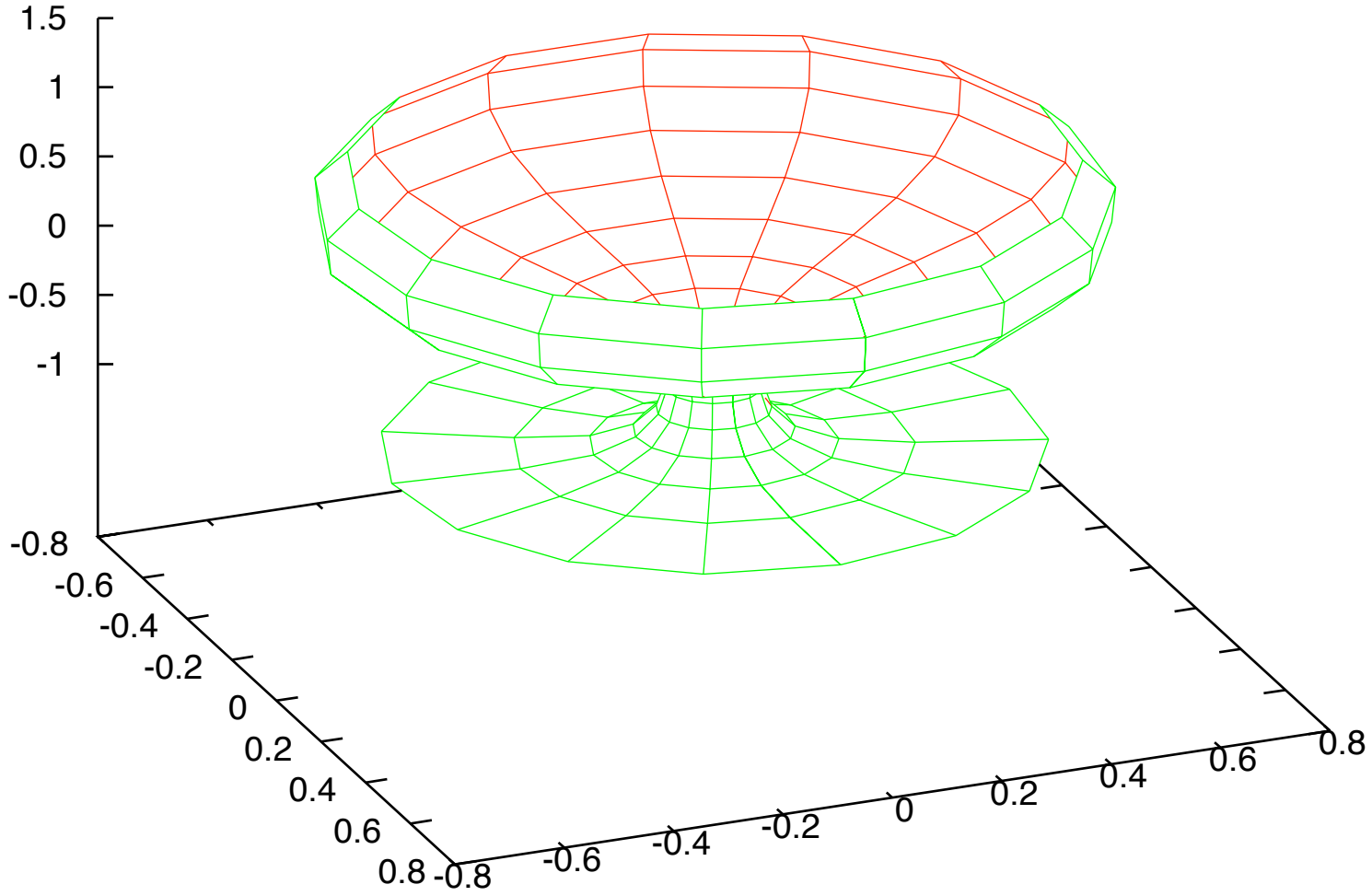

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0.8
0.6
0.4
0.2

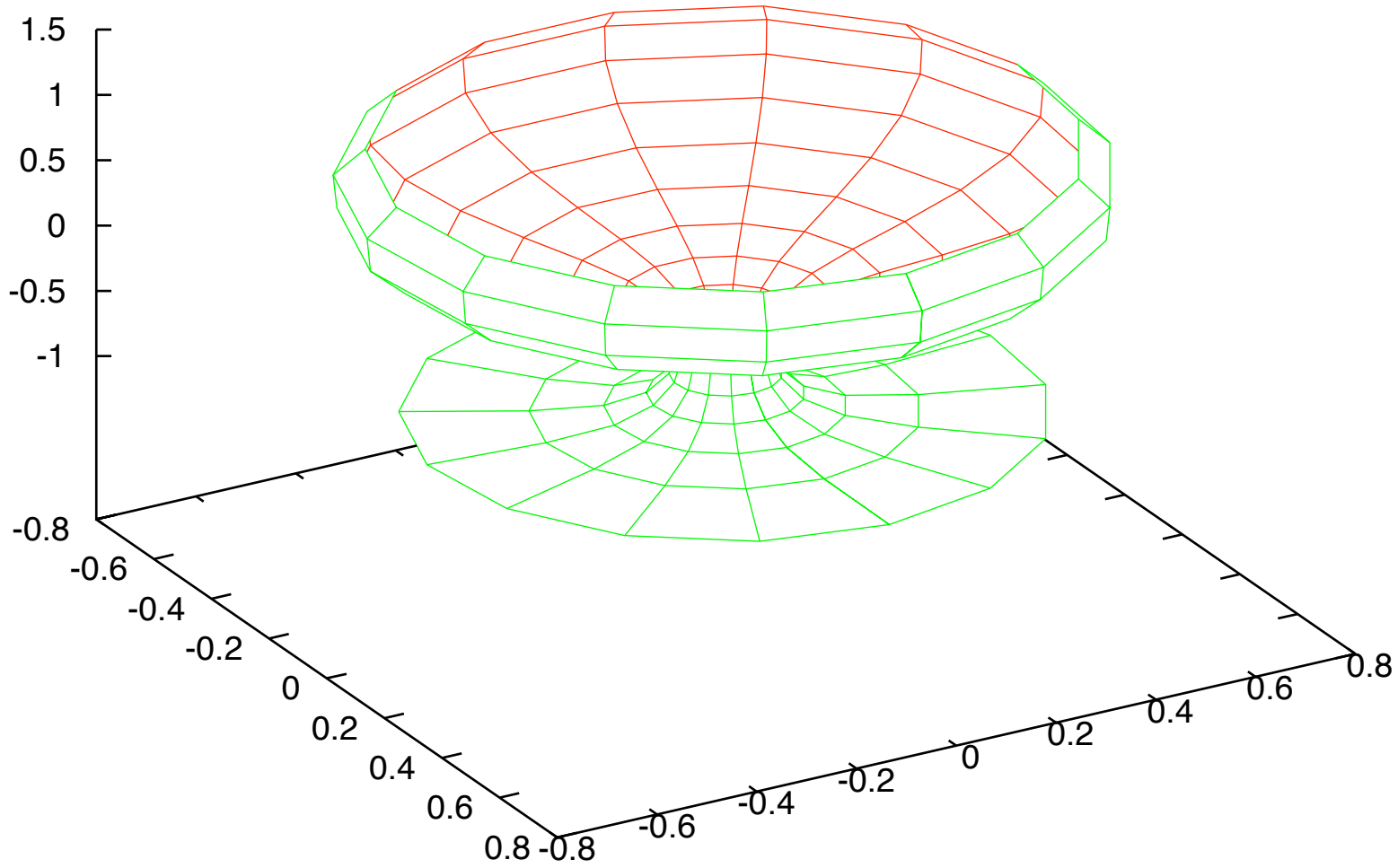

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0.8

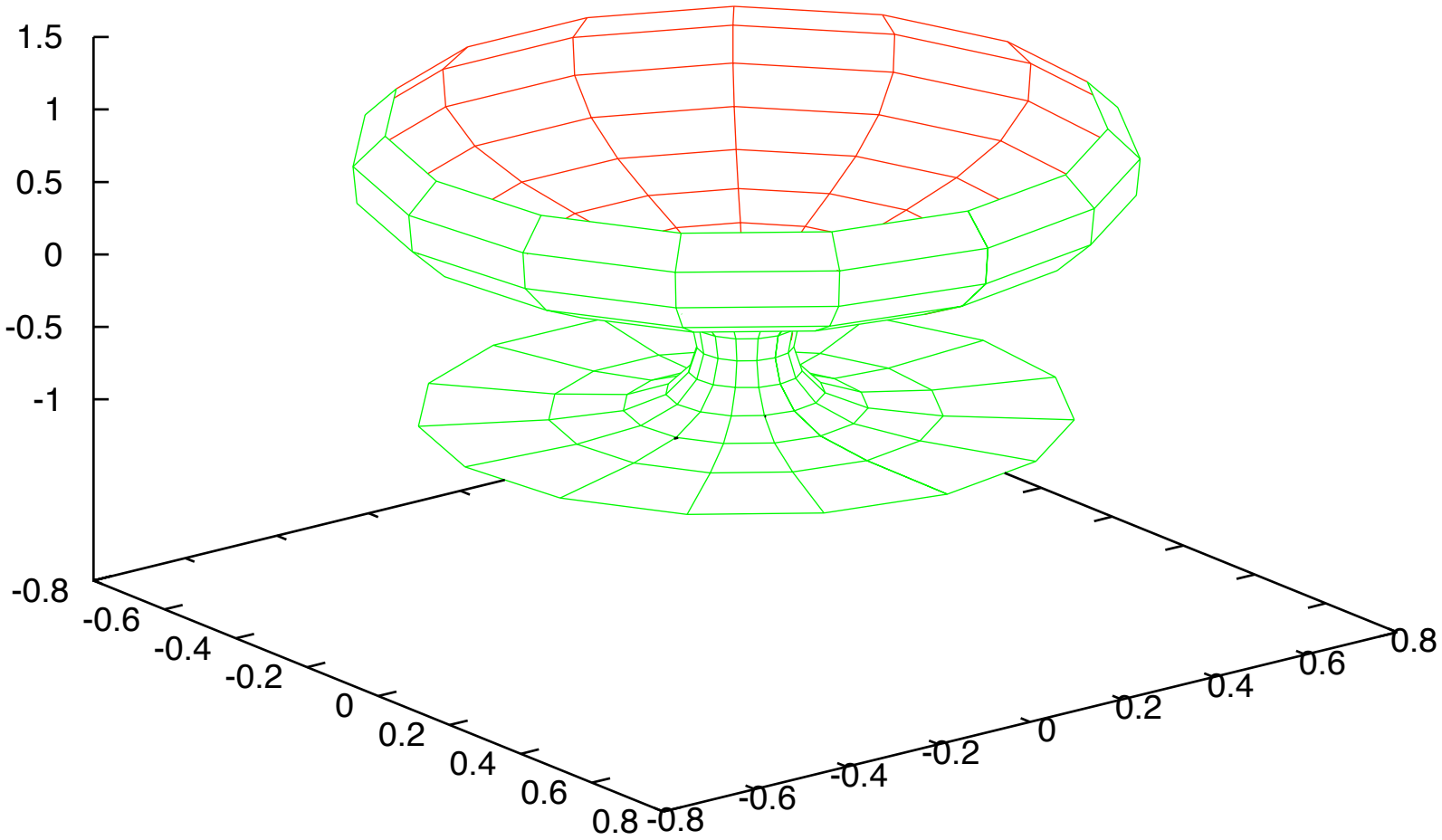

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

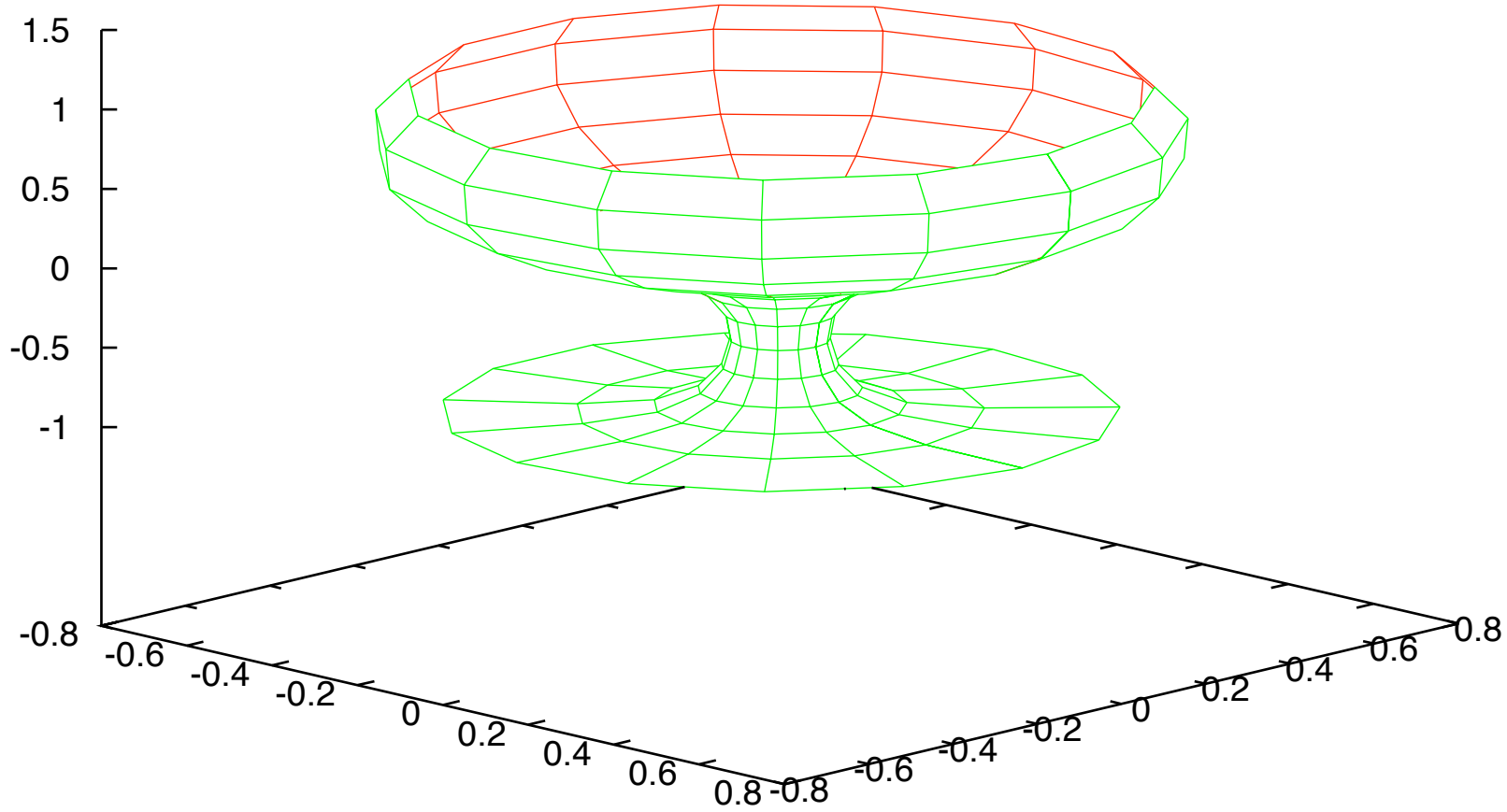

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

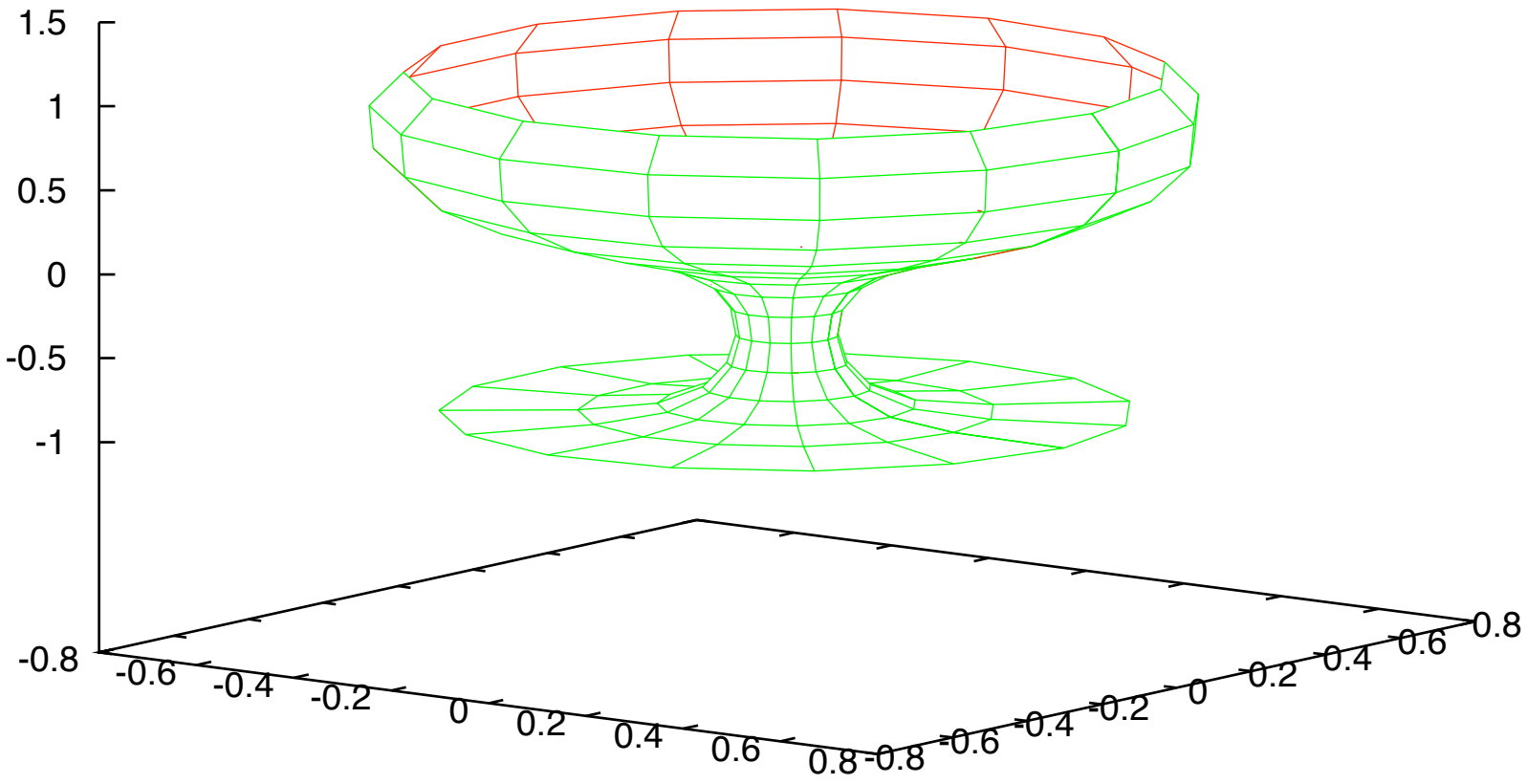

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

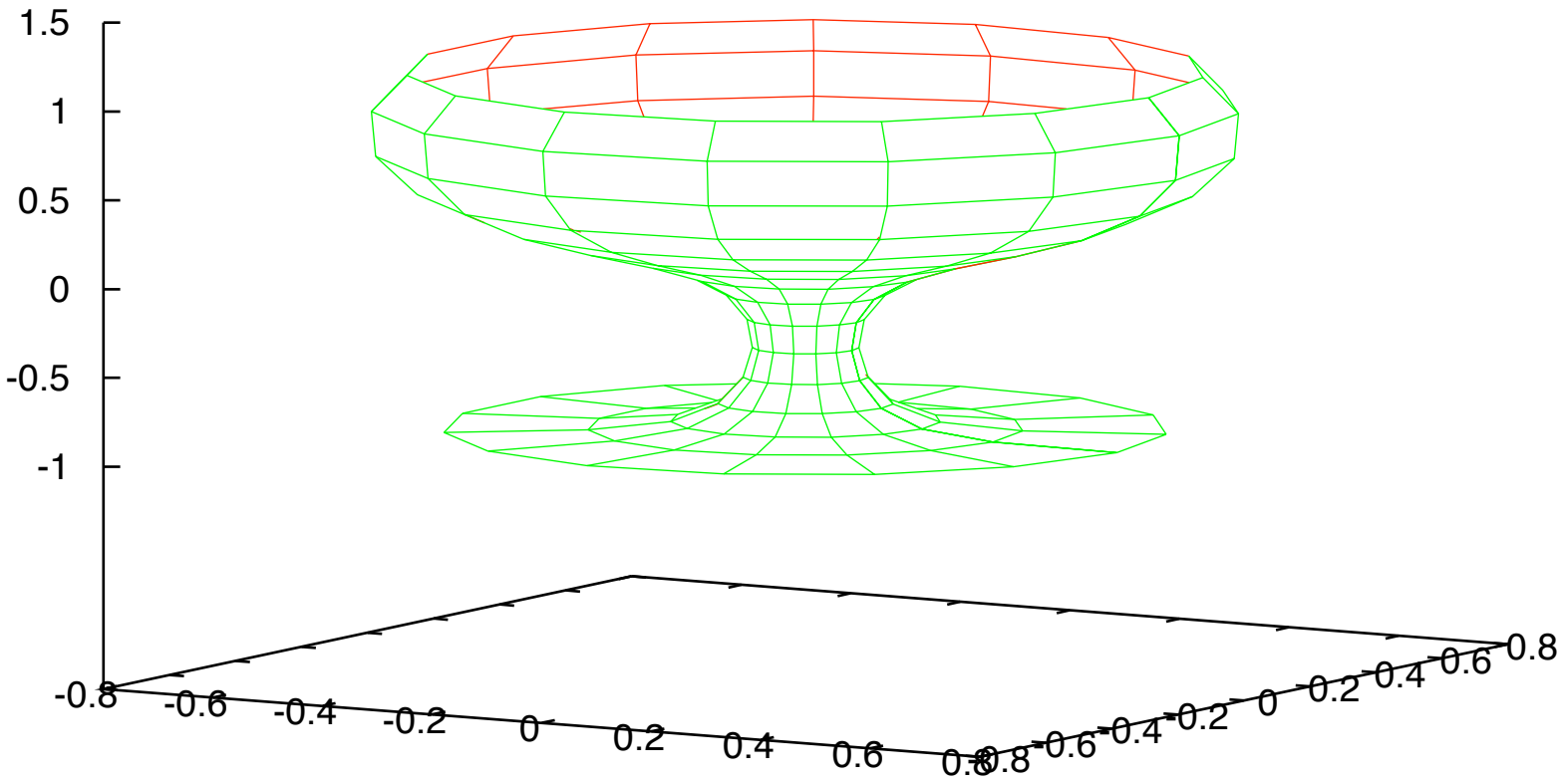

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

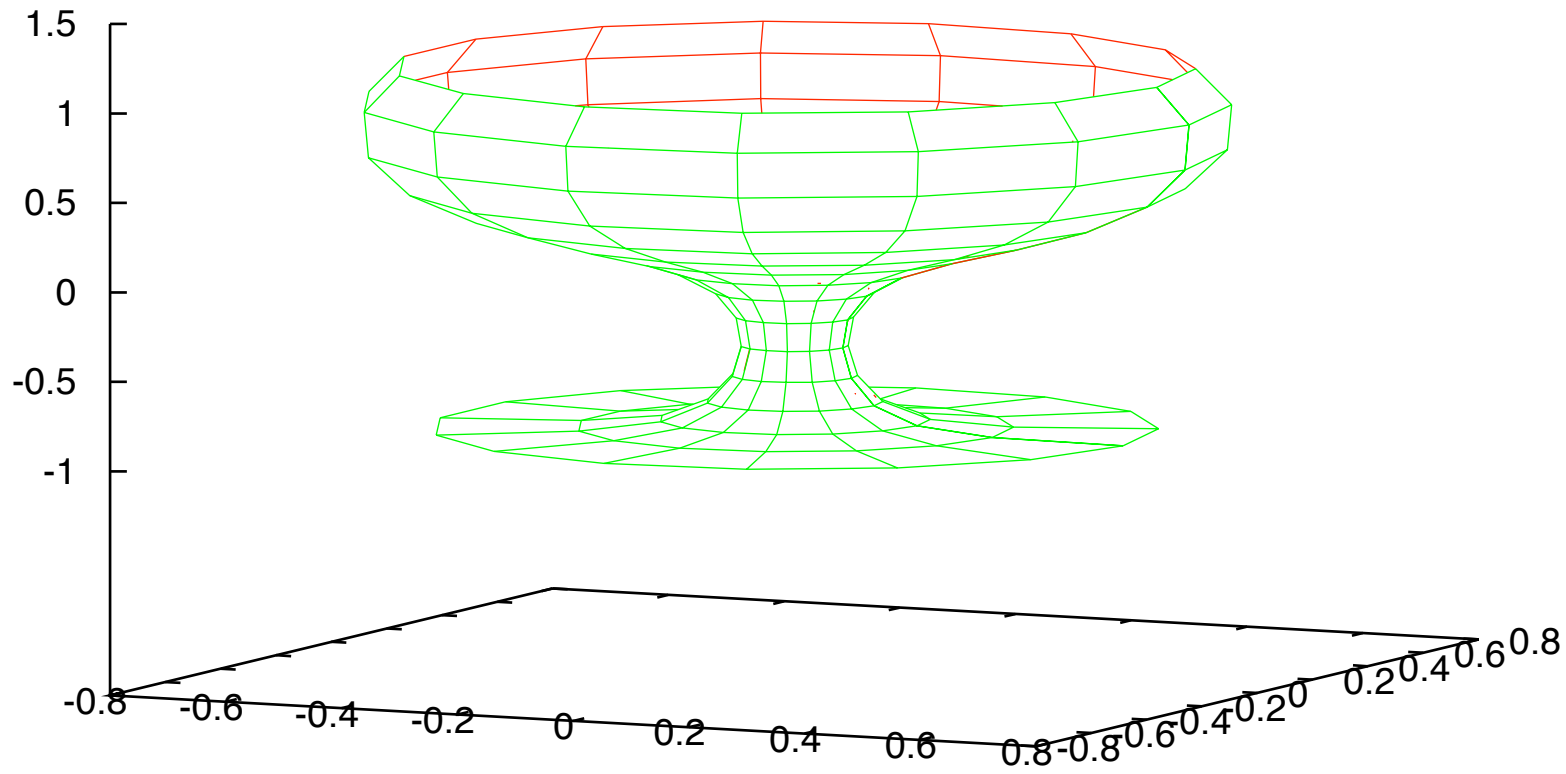

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

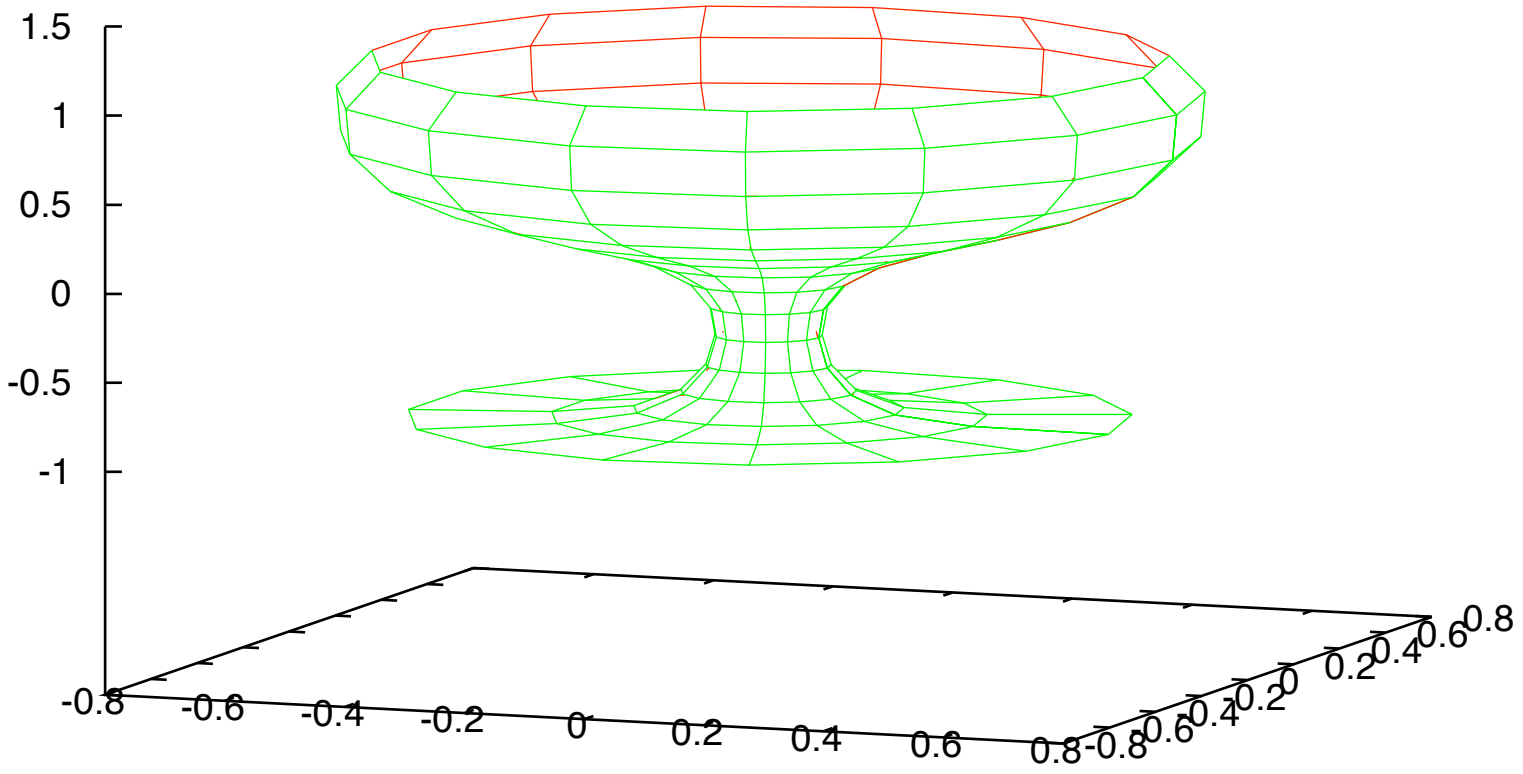

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

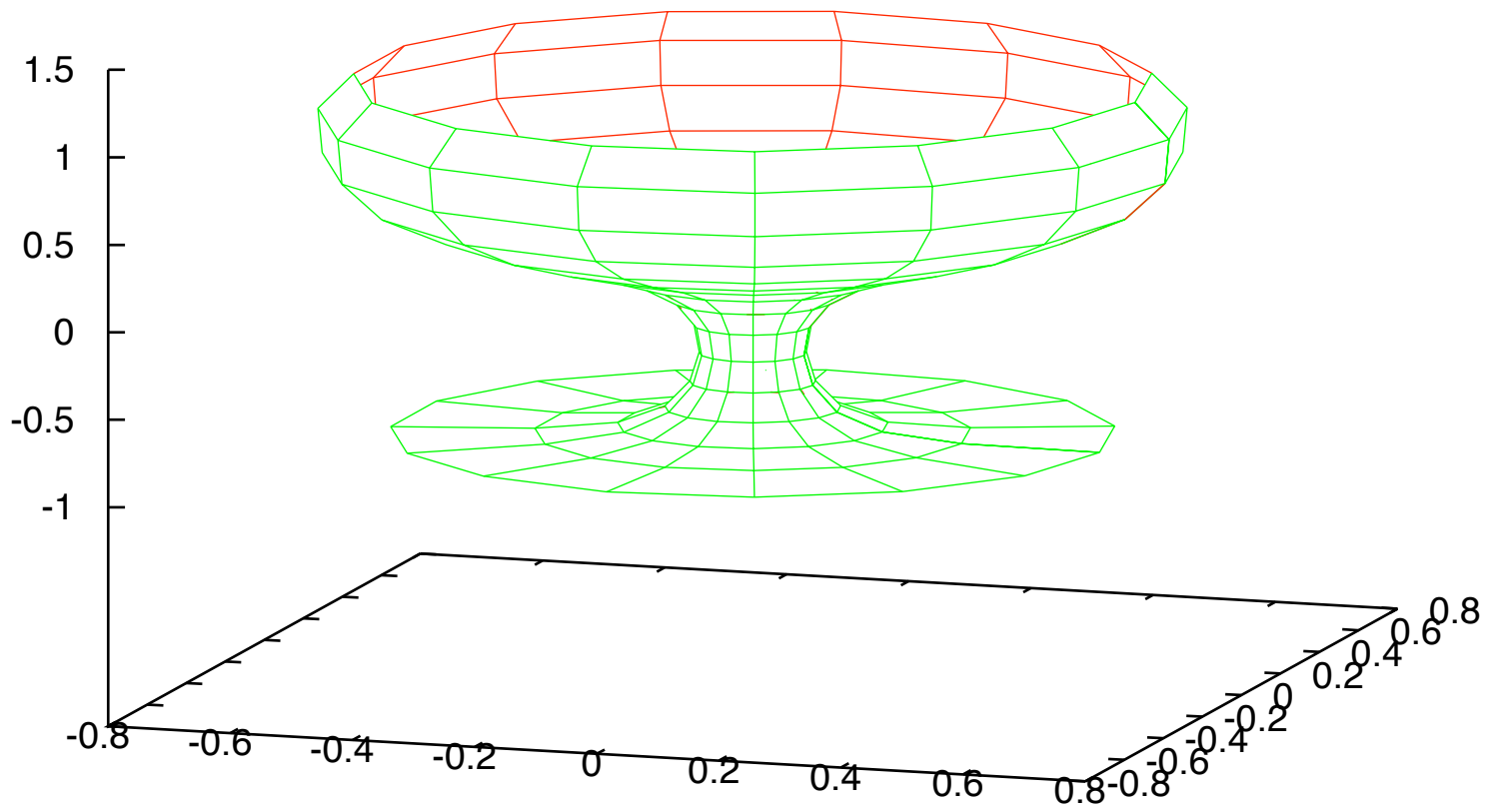

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

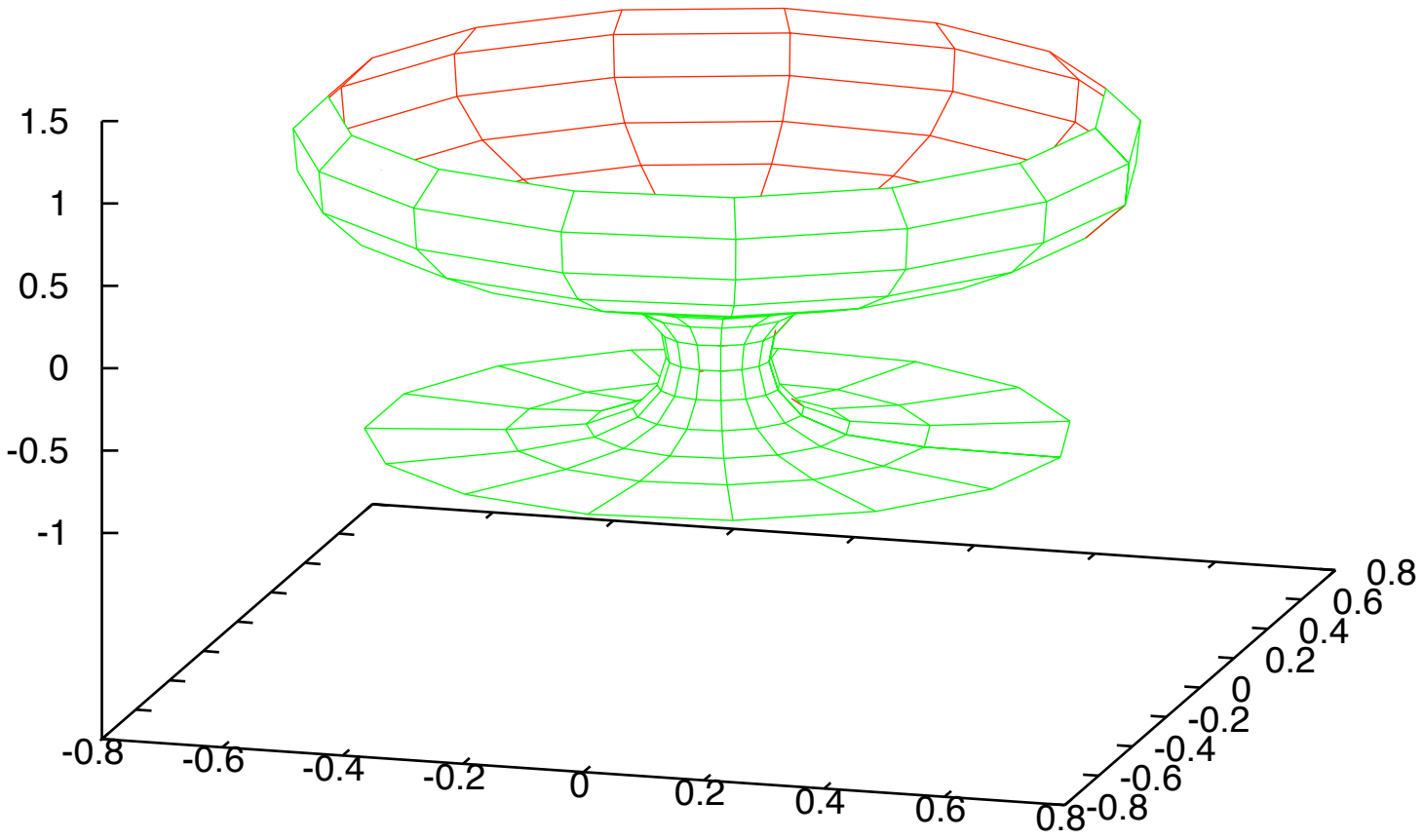

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

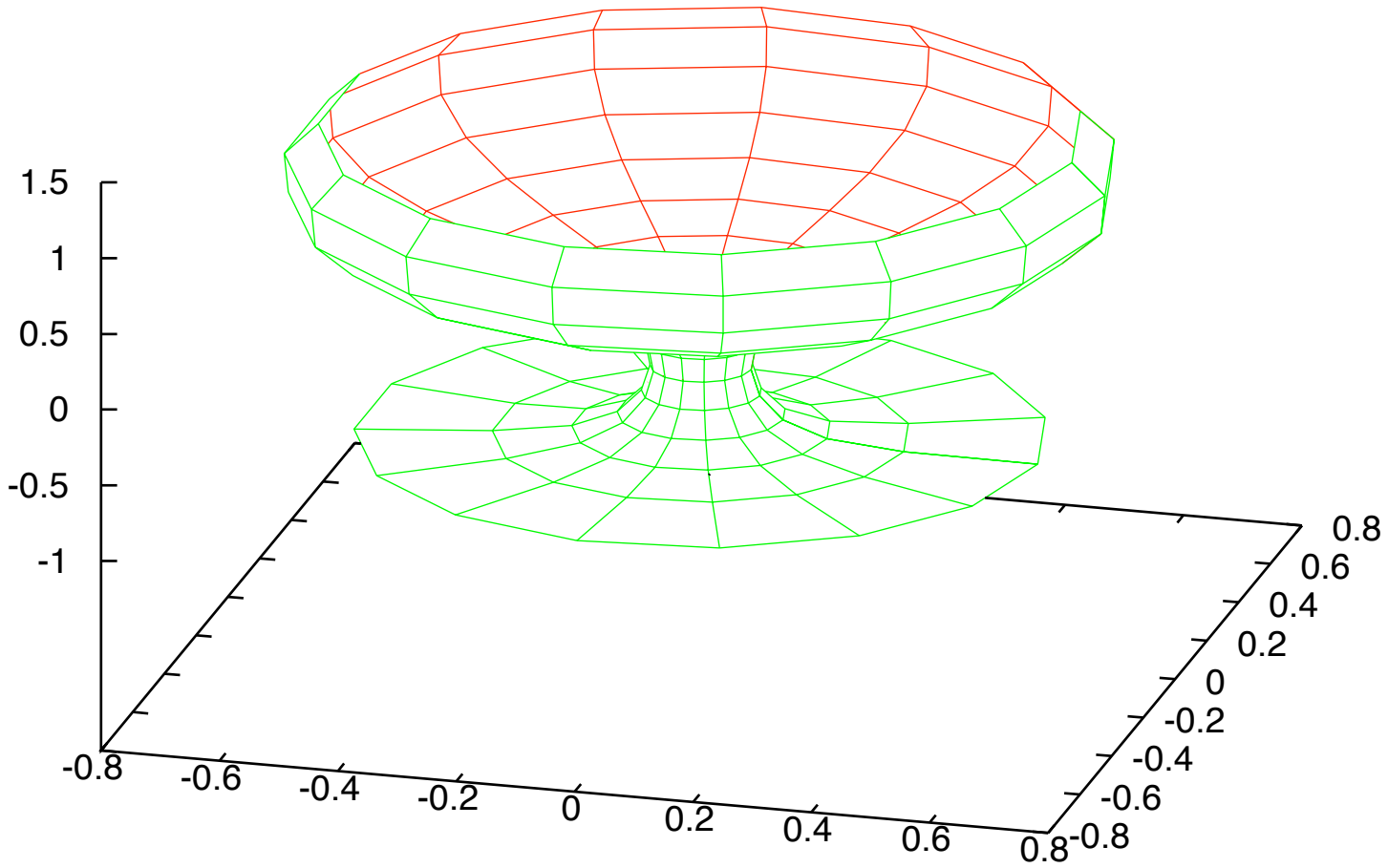

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8

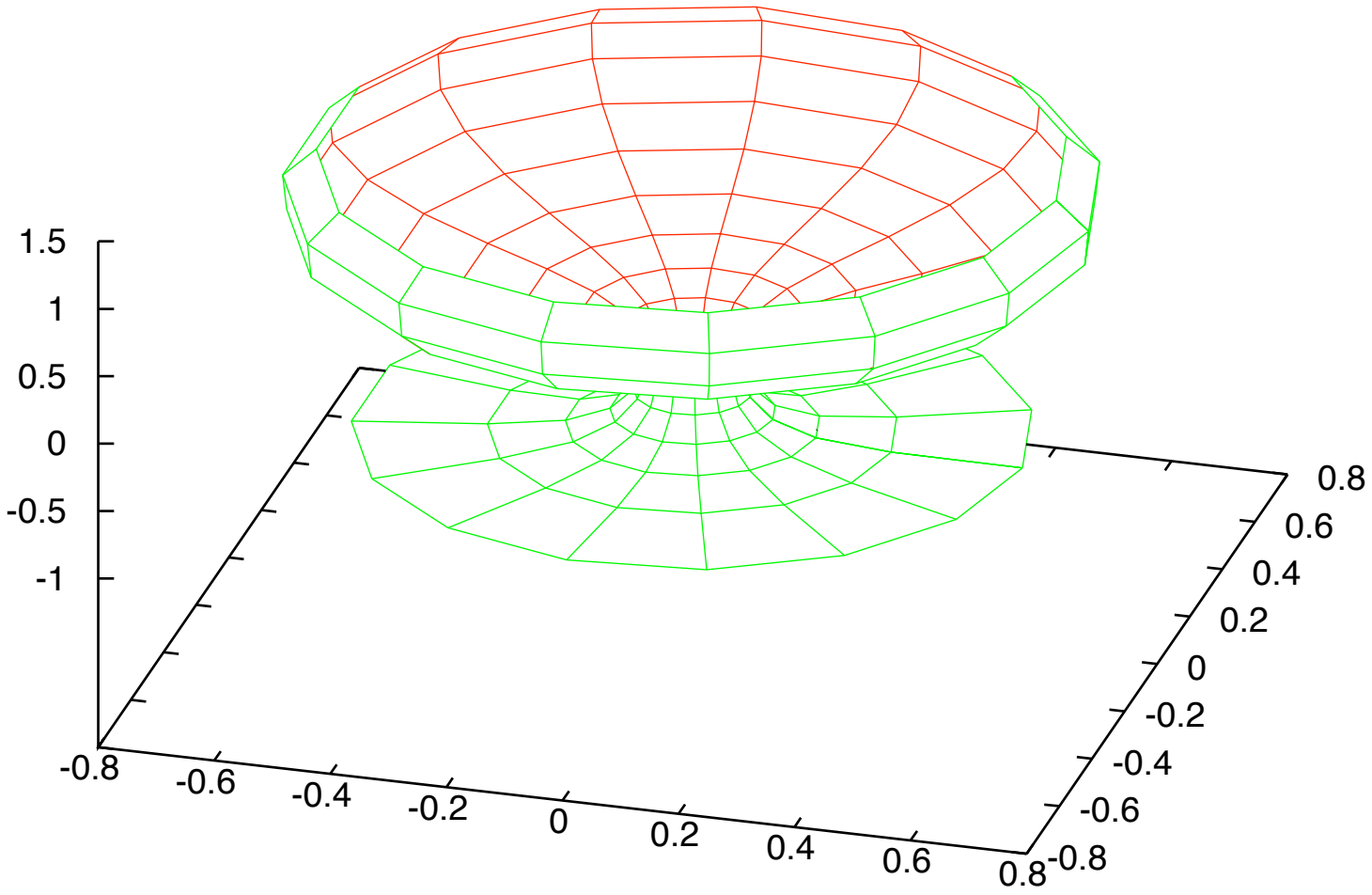

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

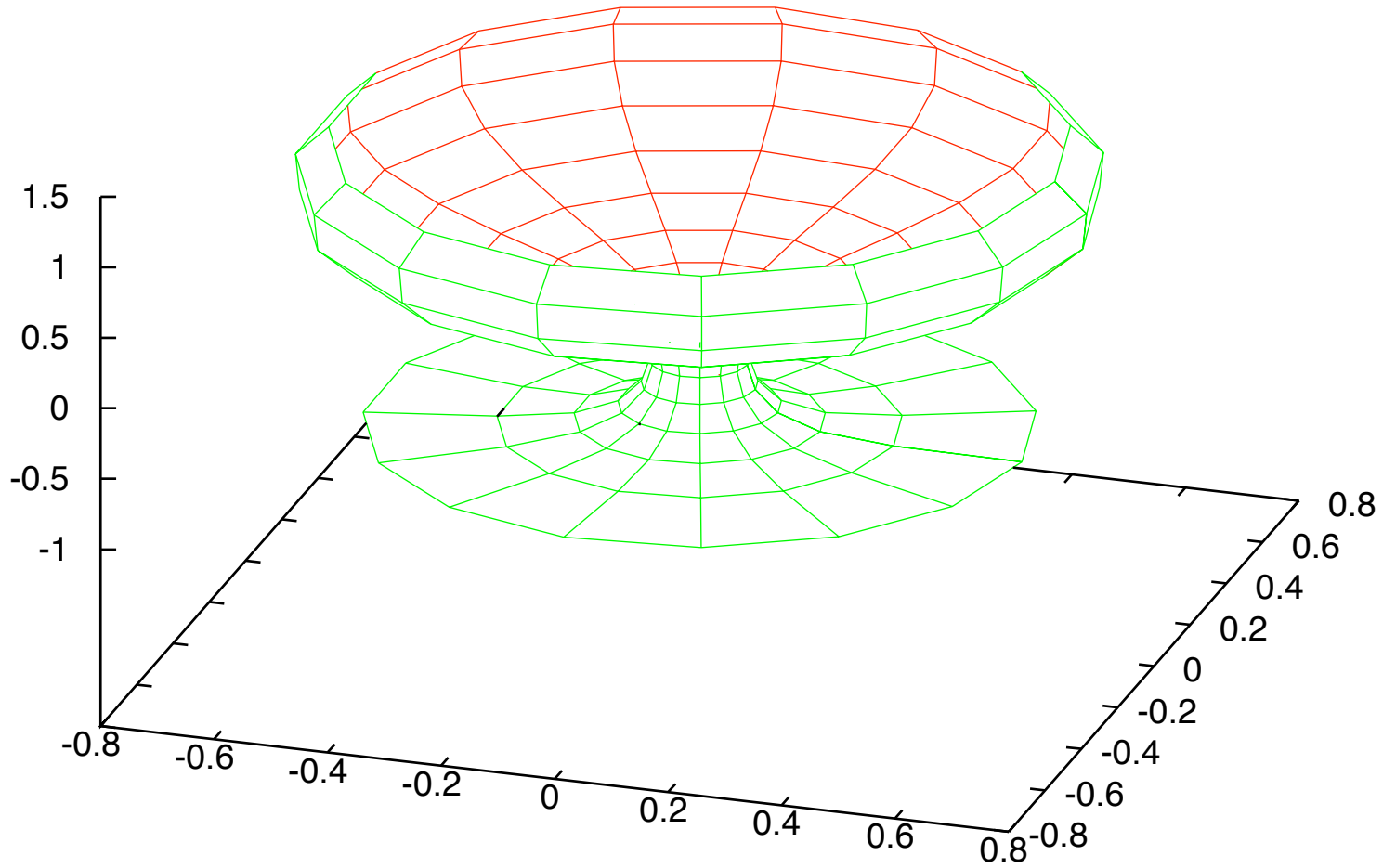

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

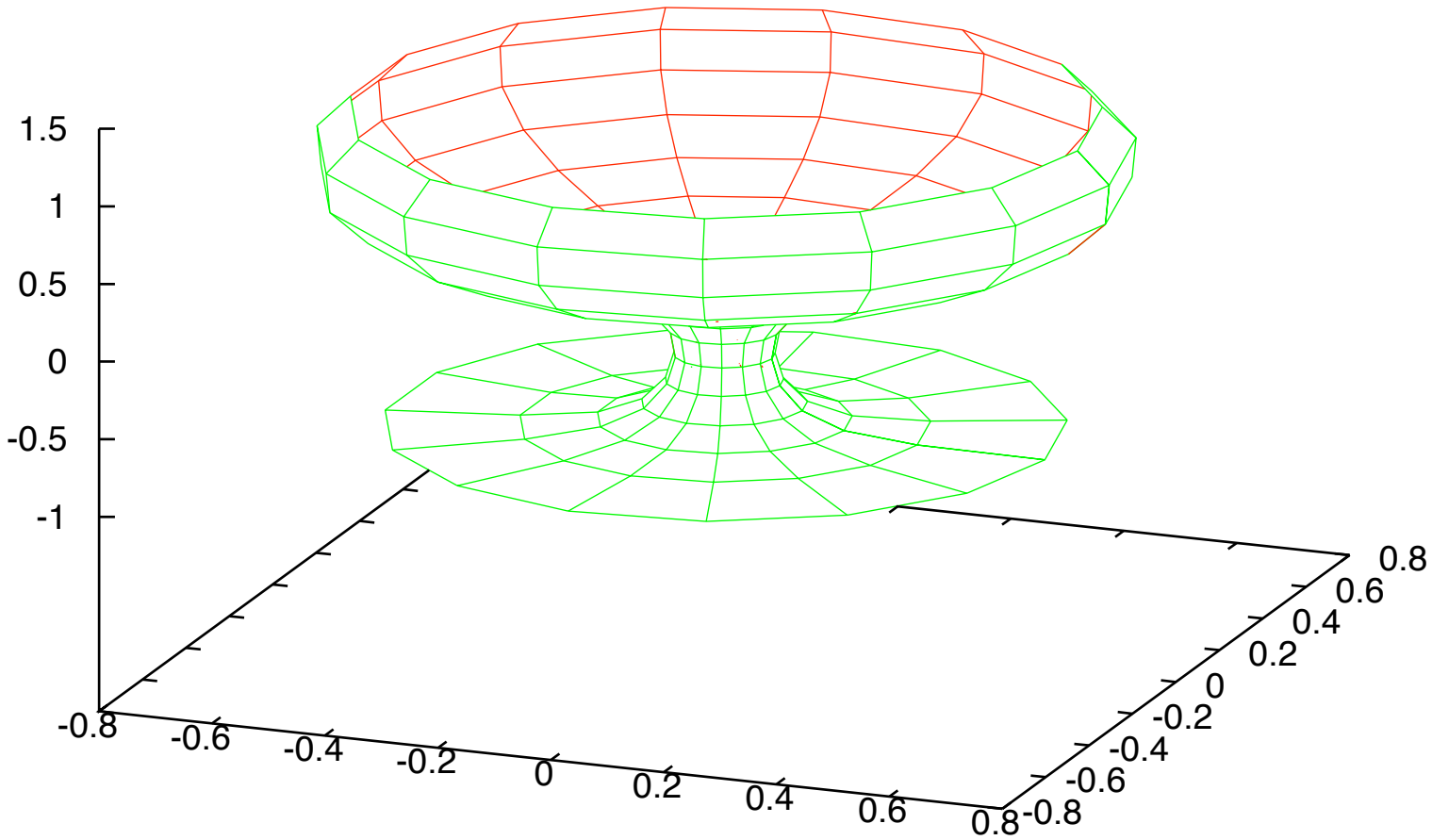

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

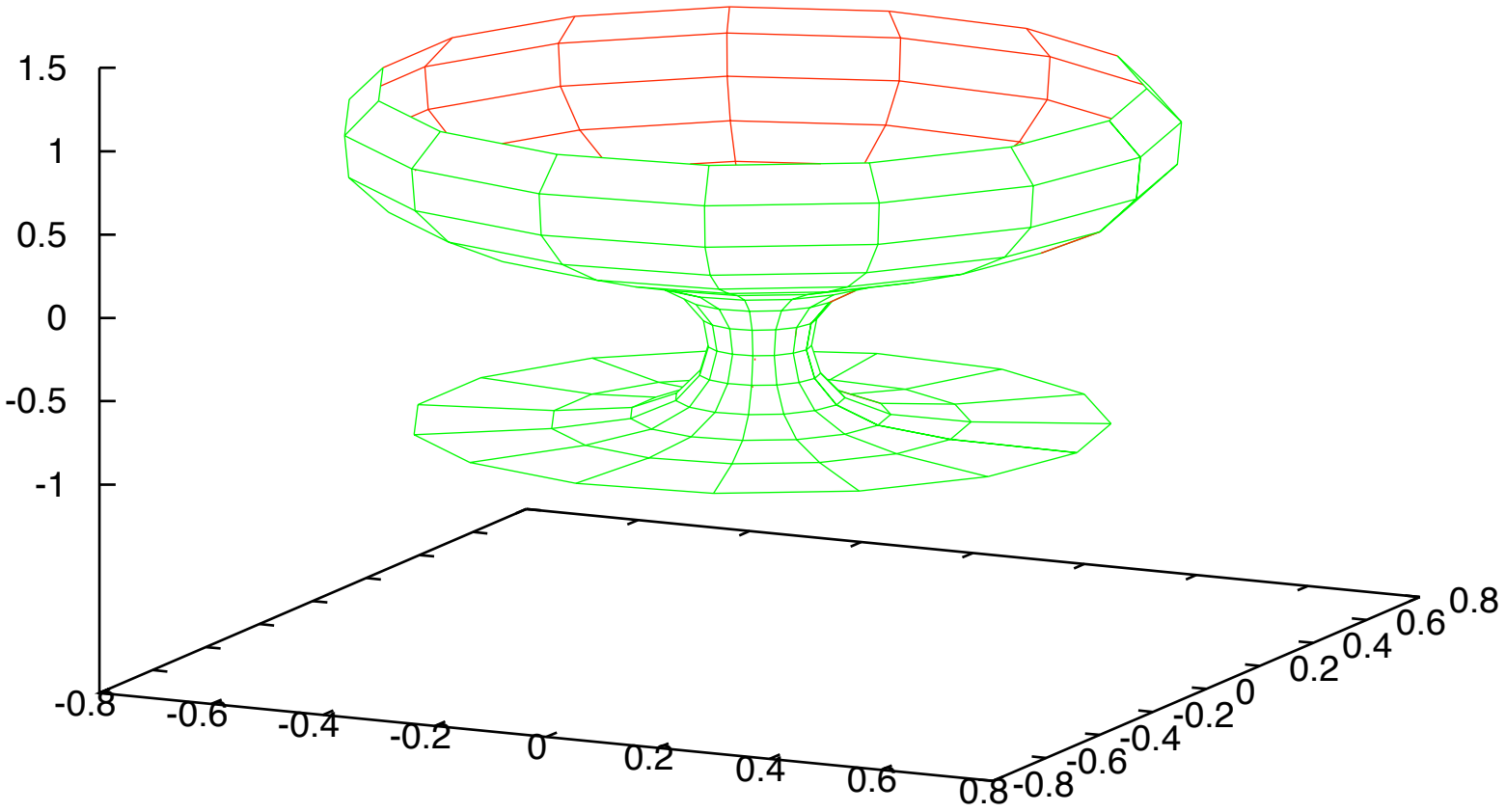

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8

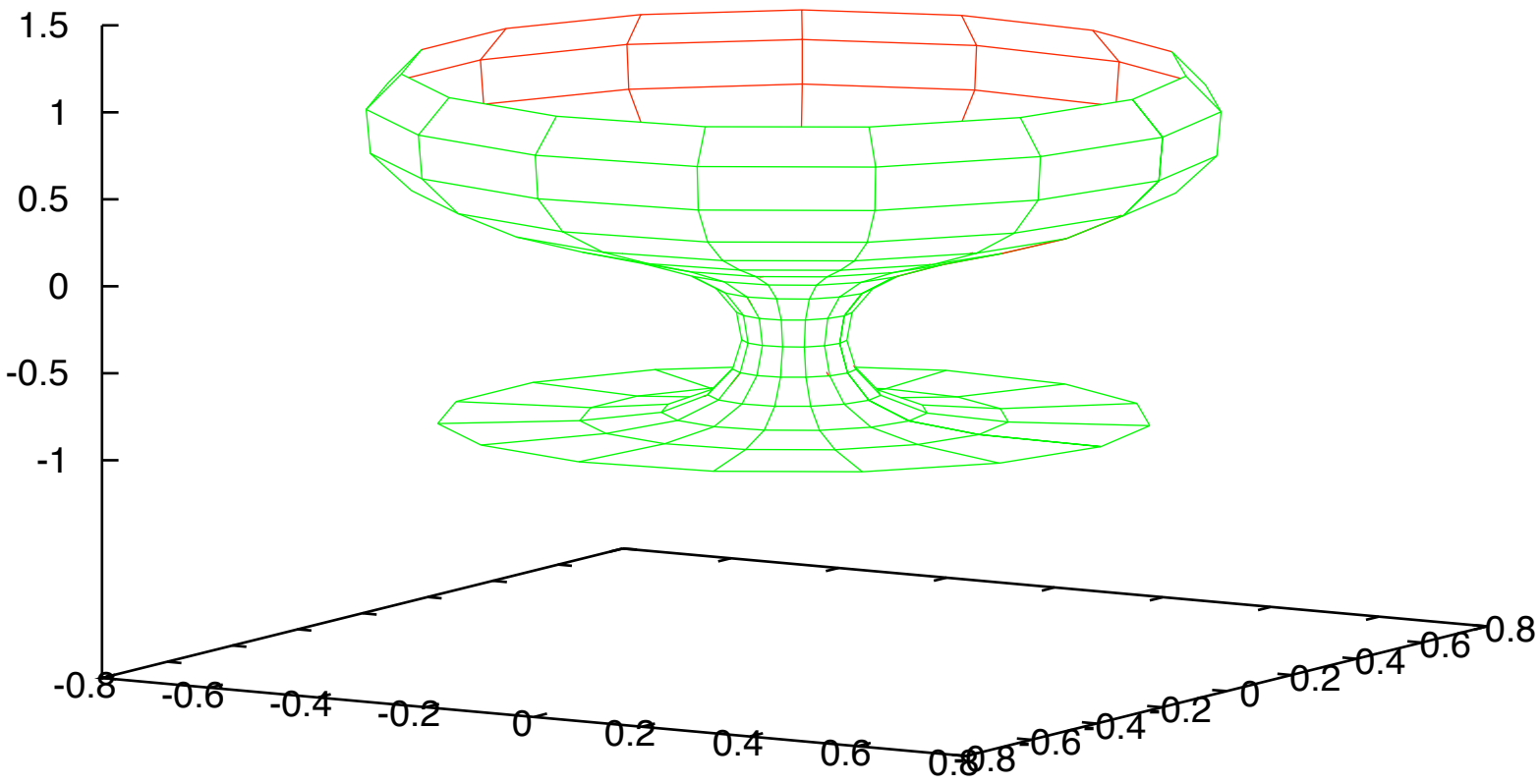

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8

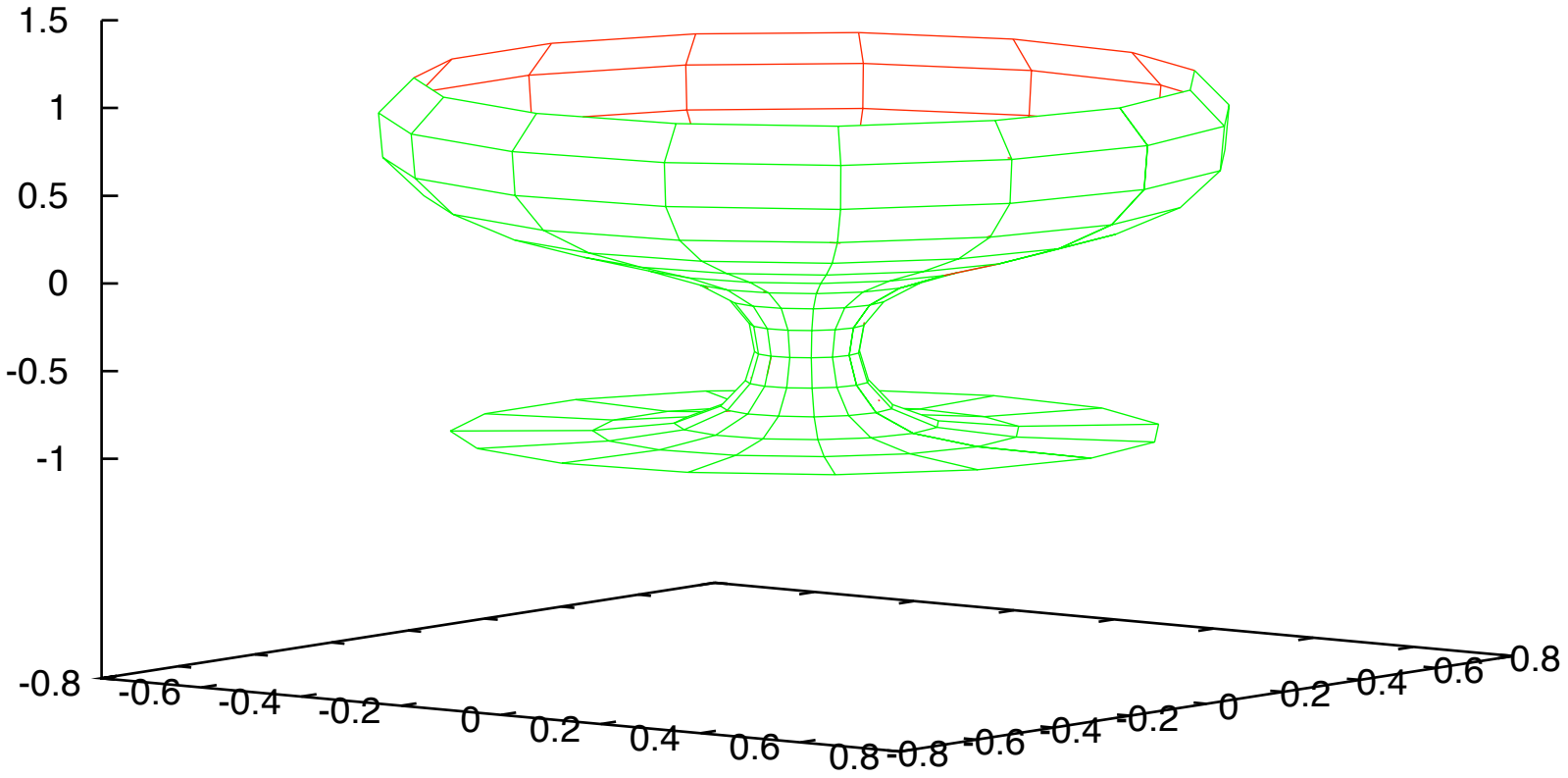

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8

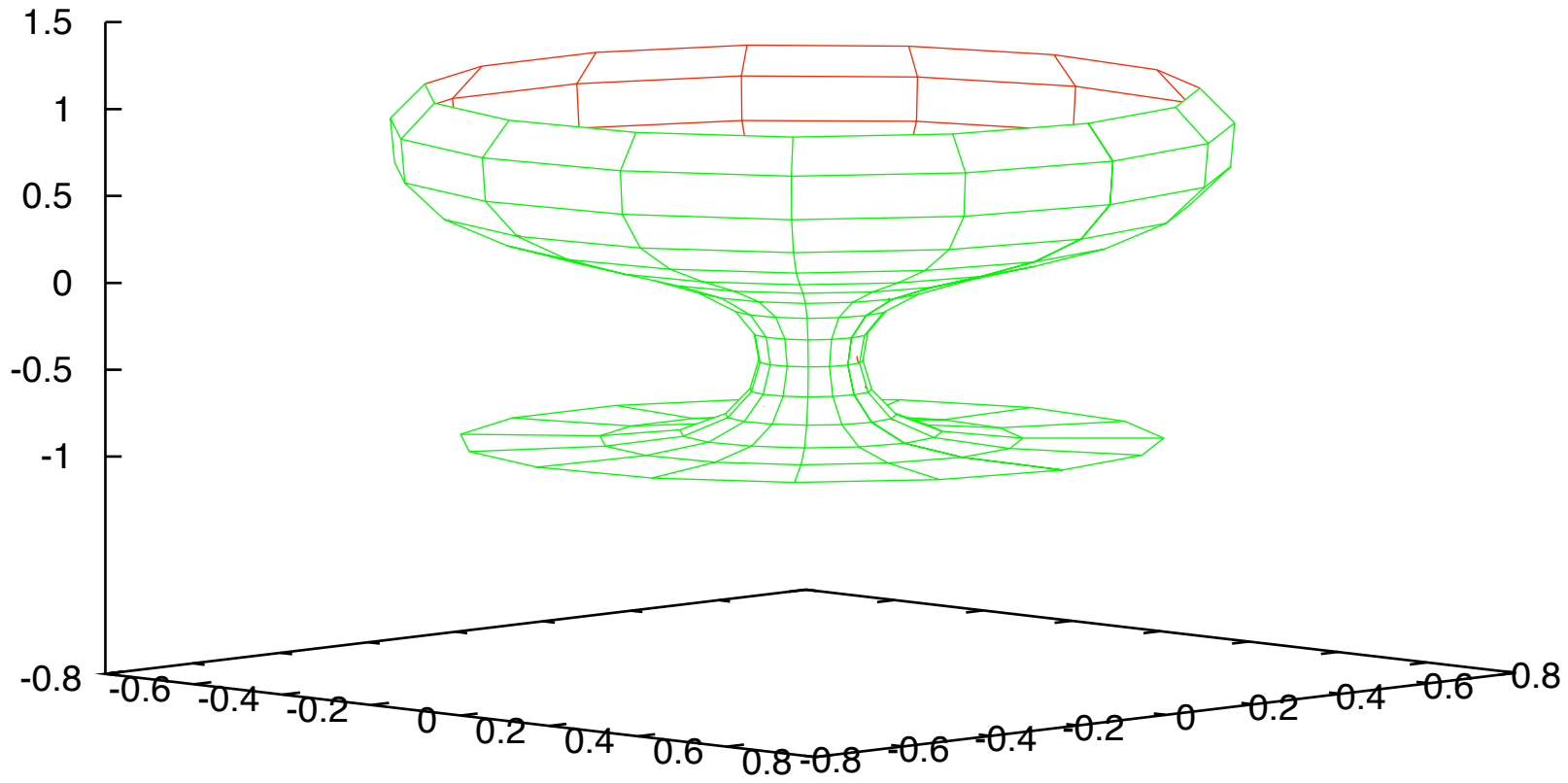

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8

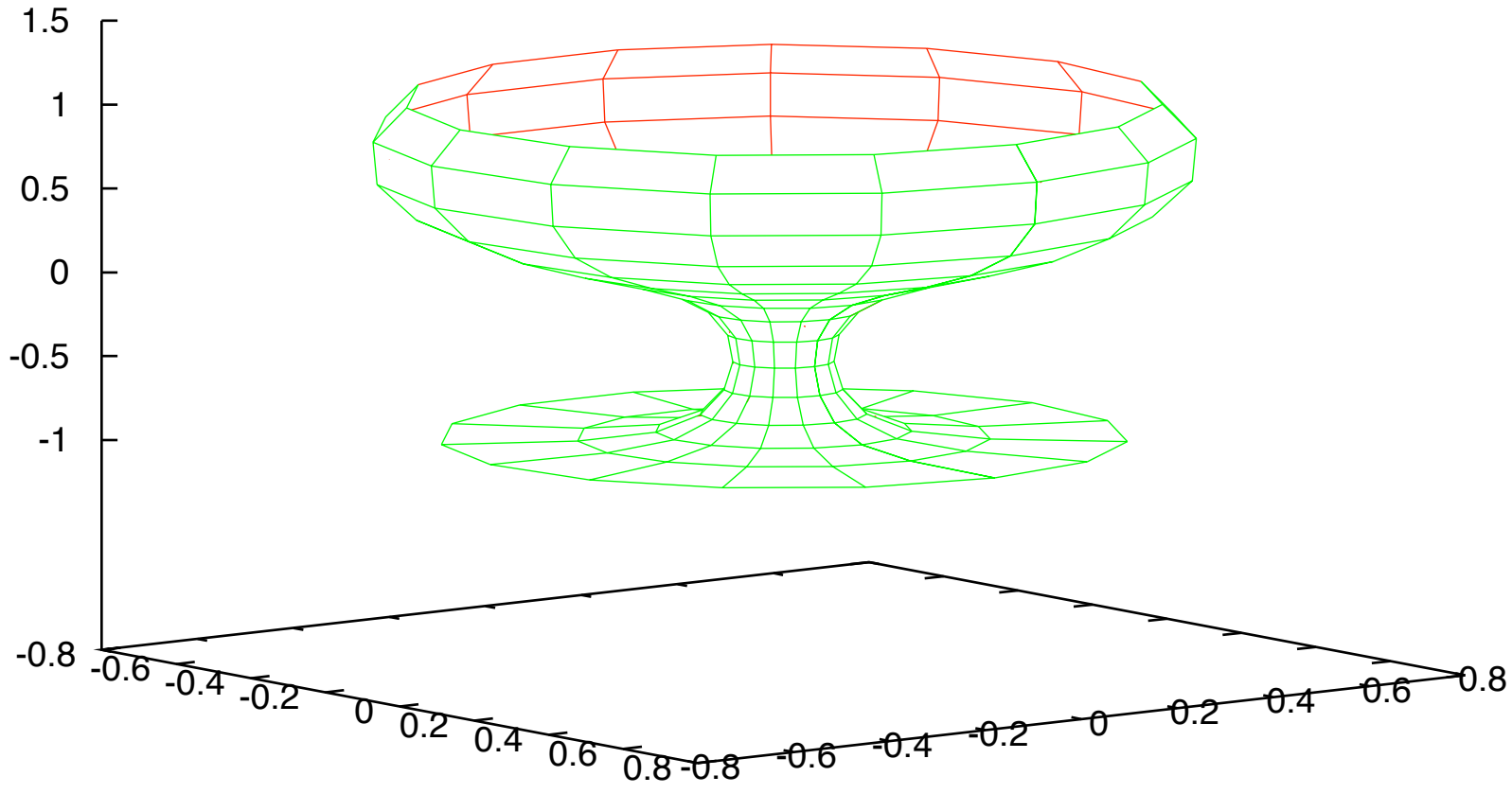

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

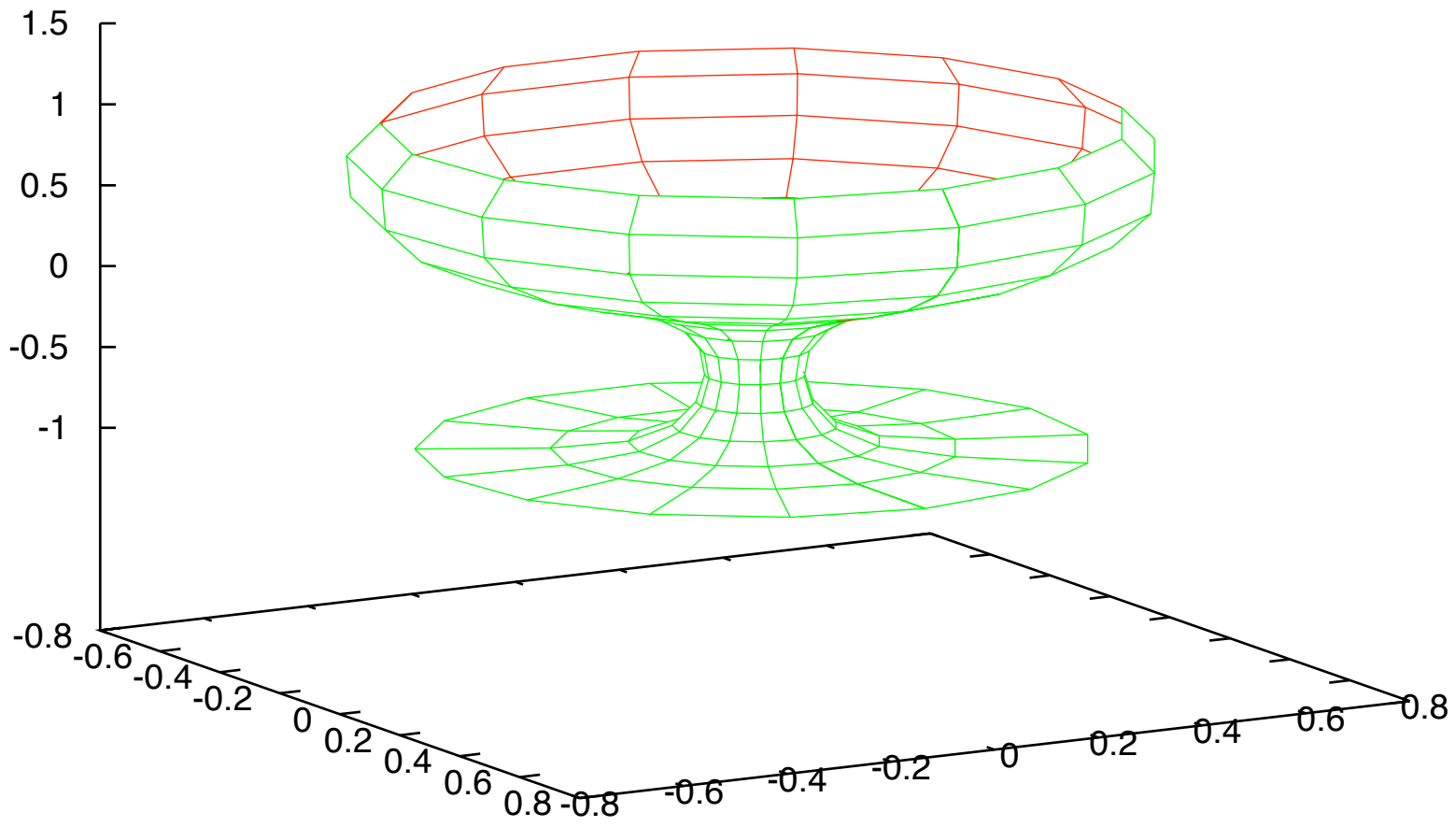

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

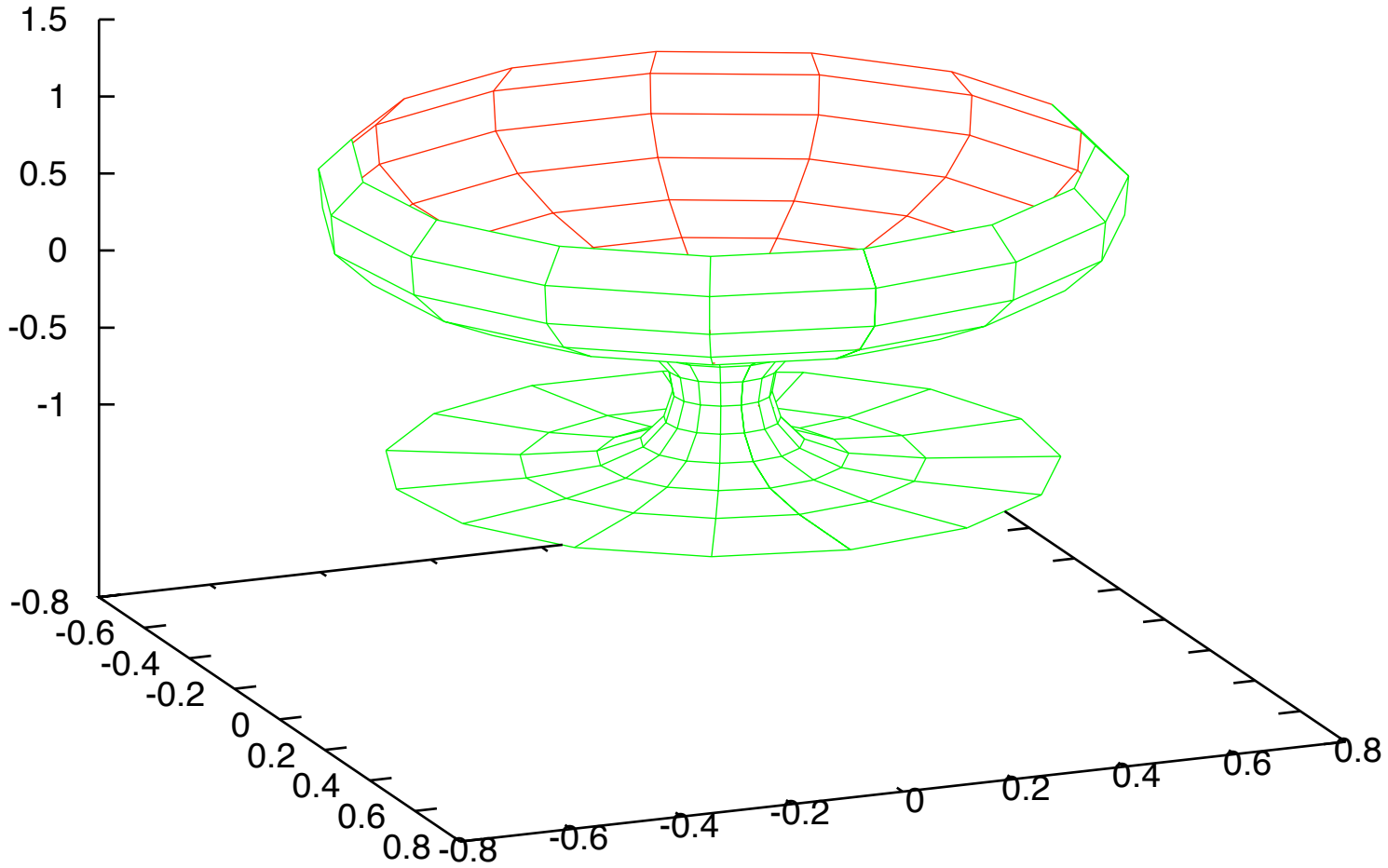

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

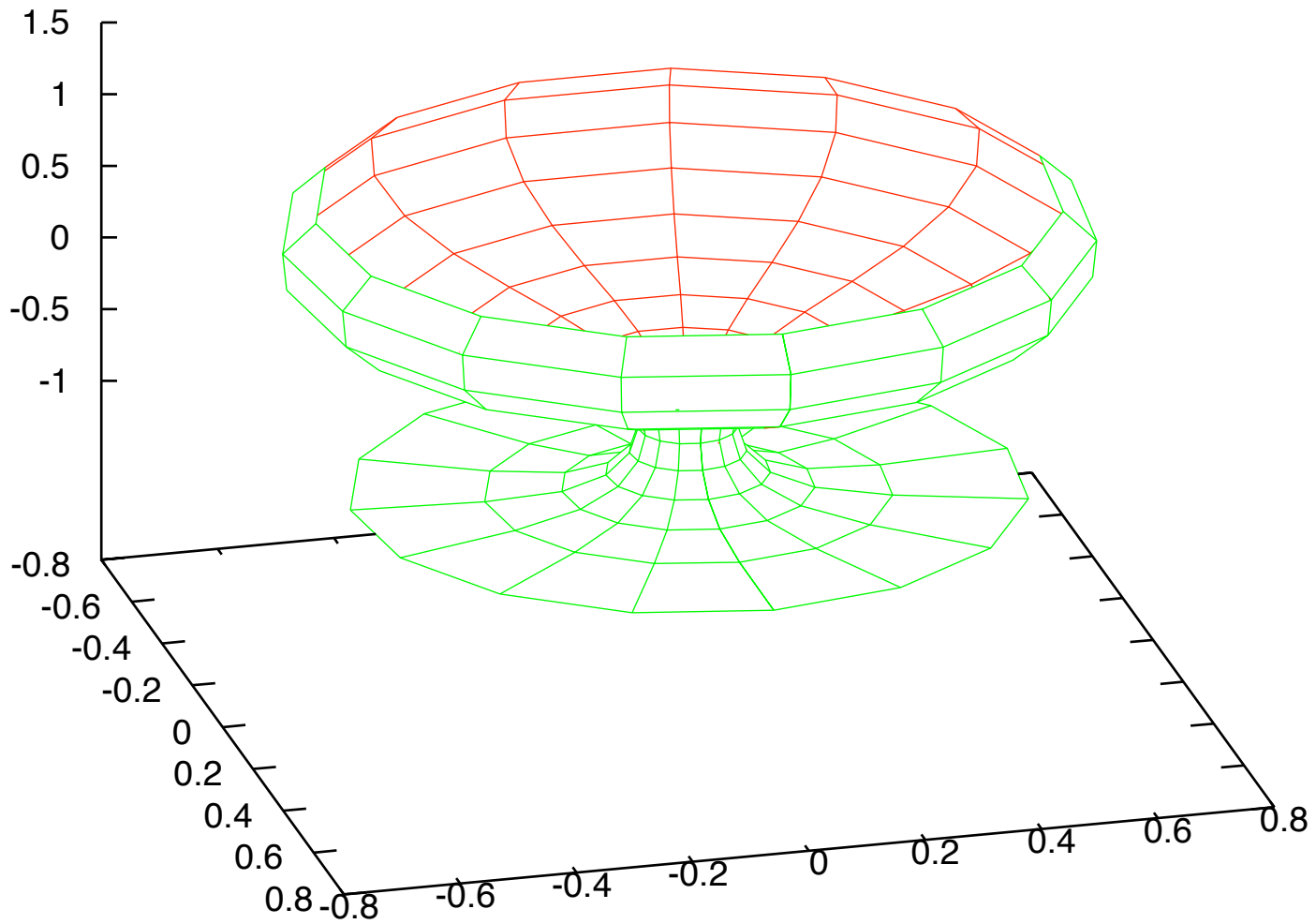

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

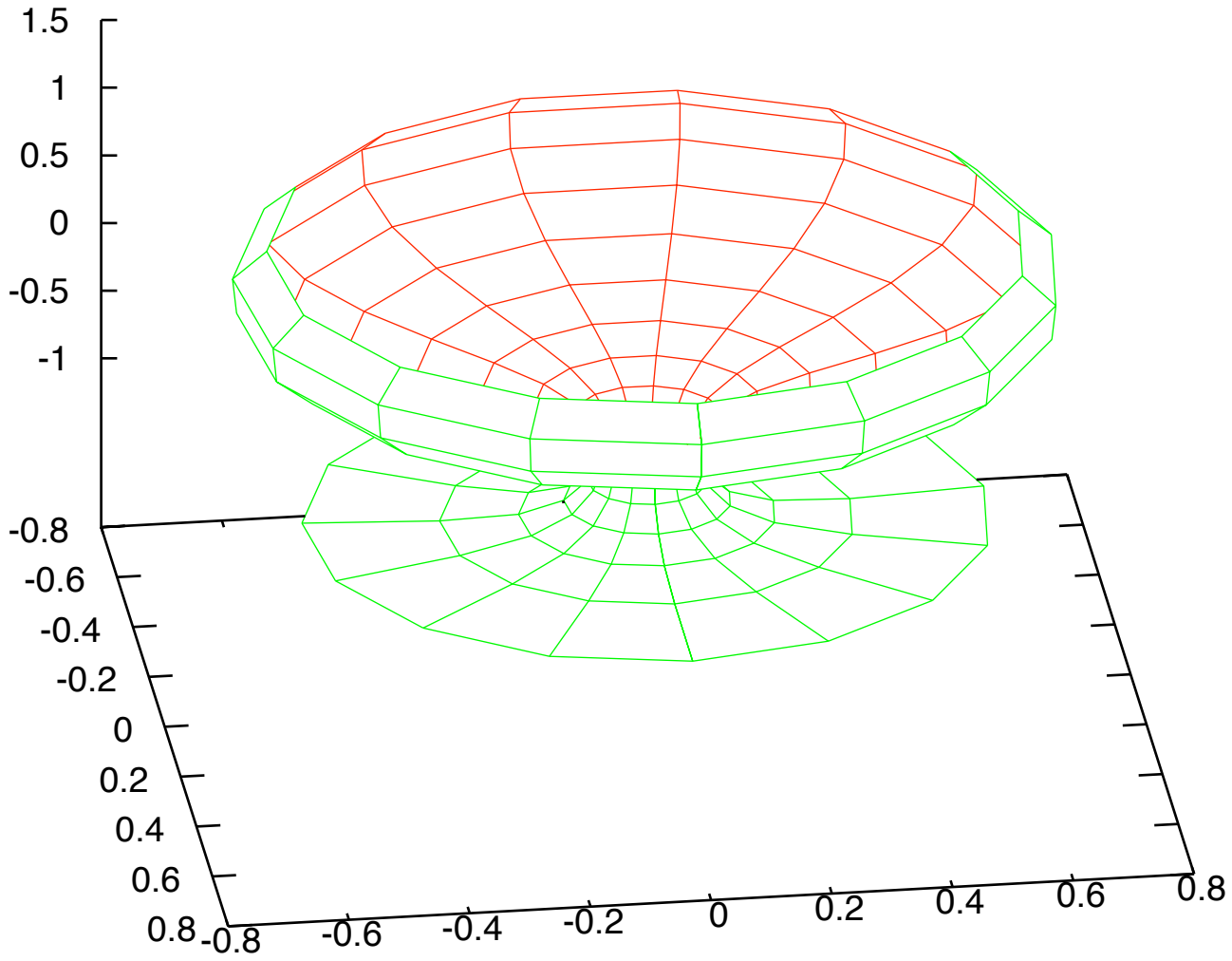

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8


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

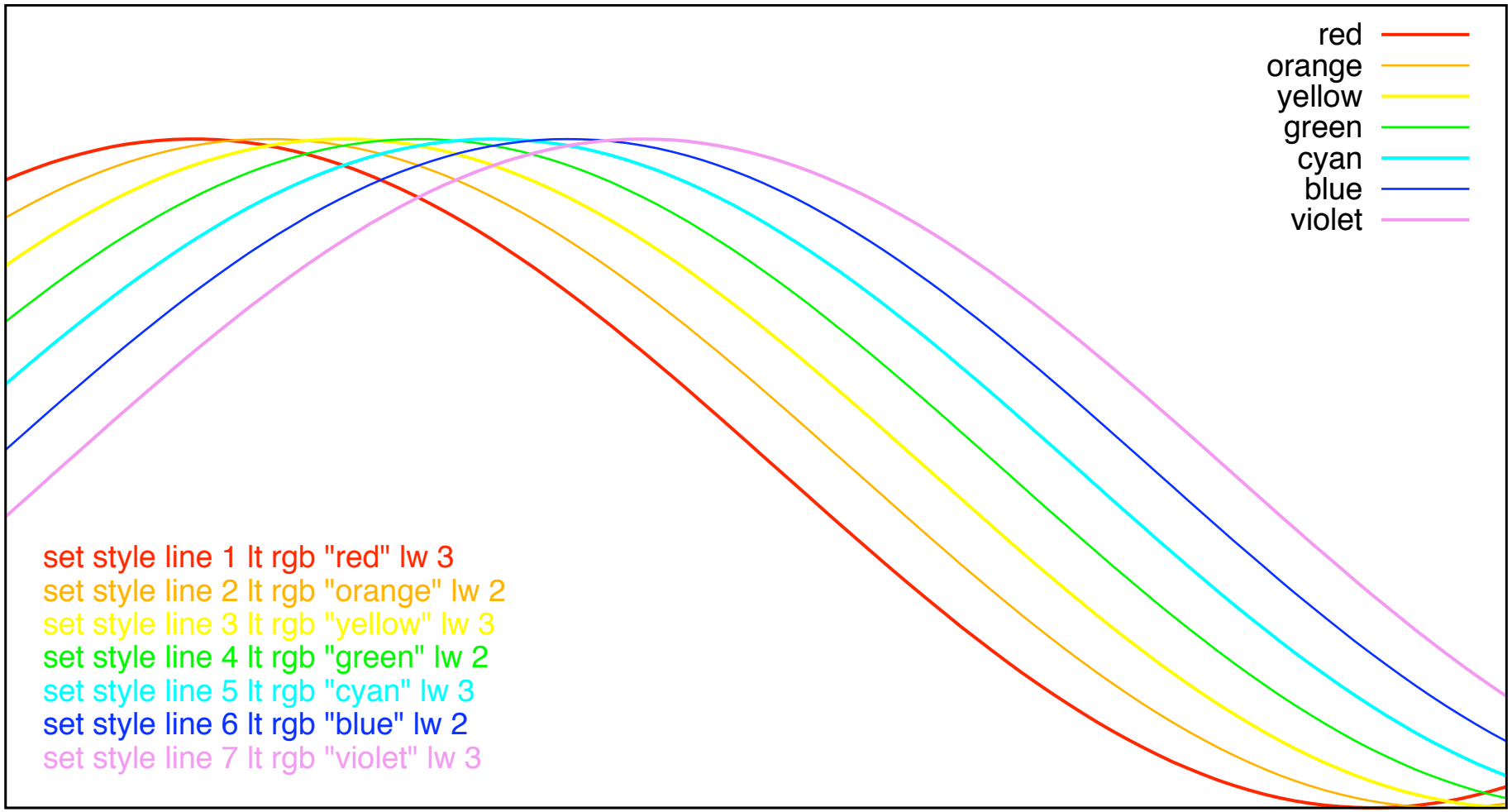

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8


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

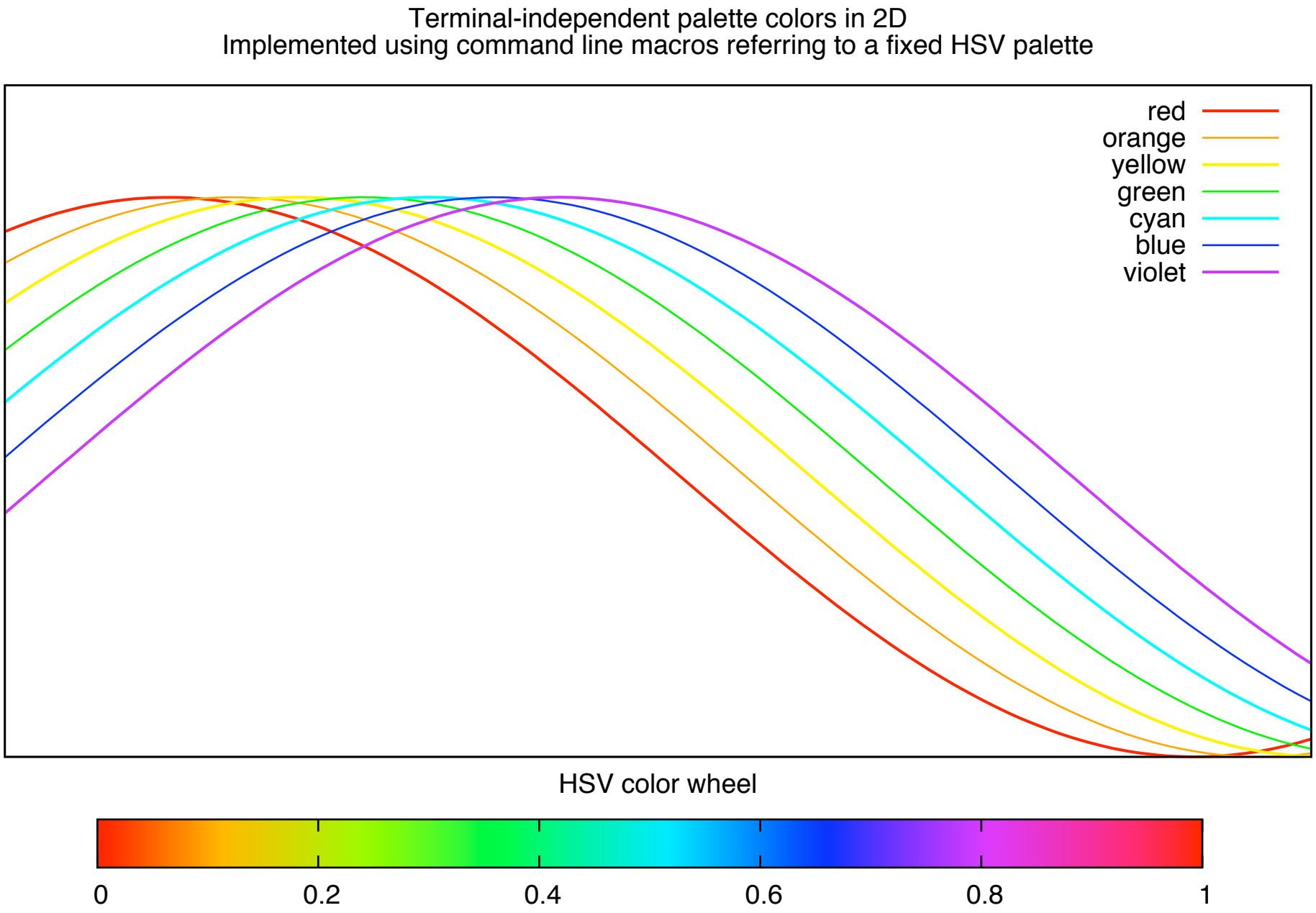

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

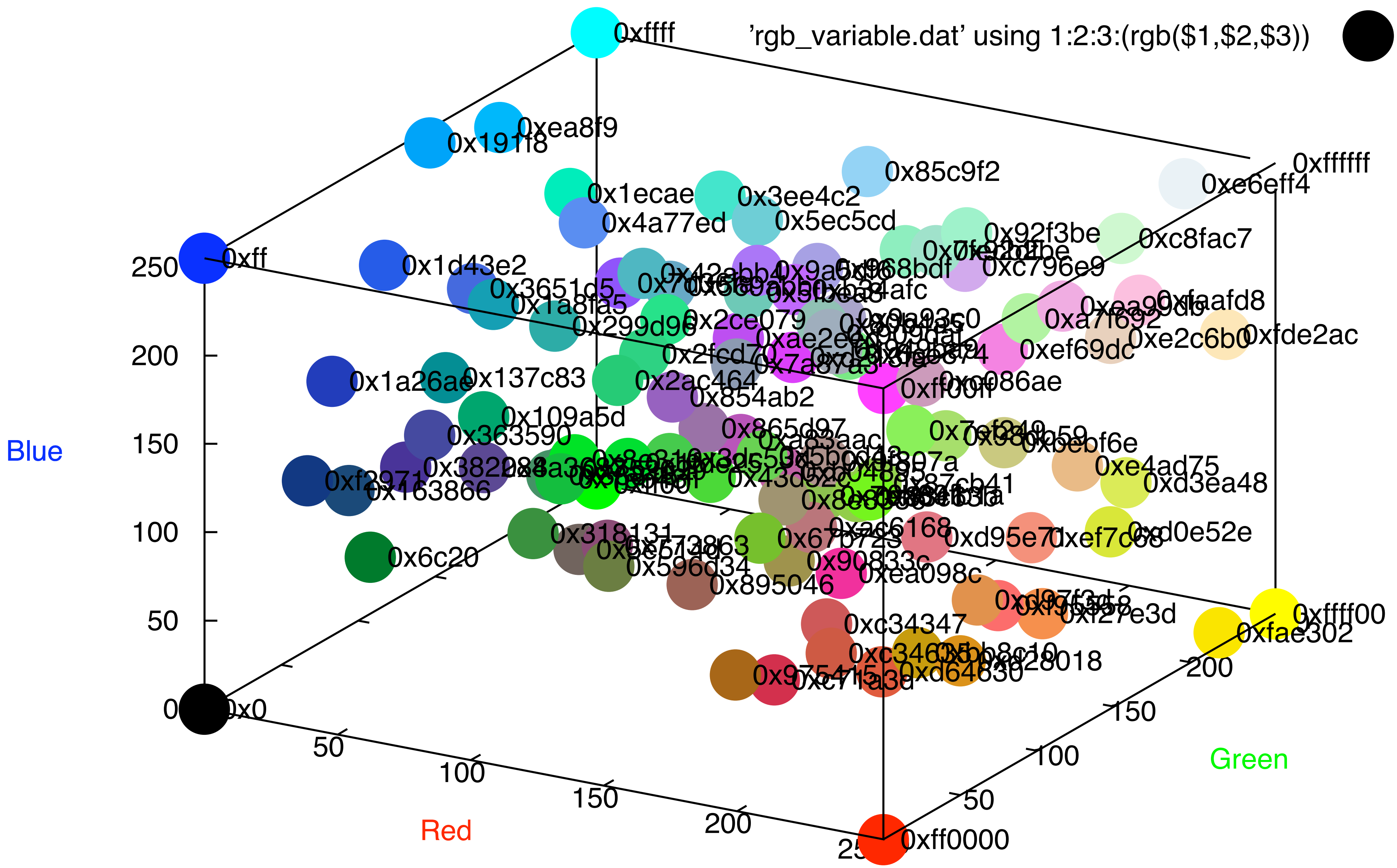

1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

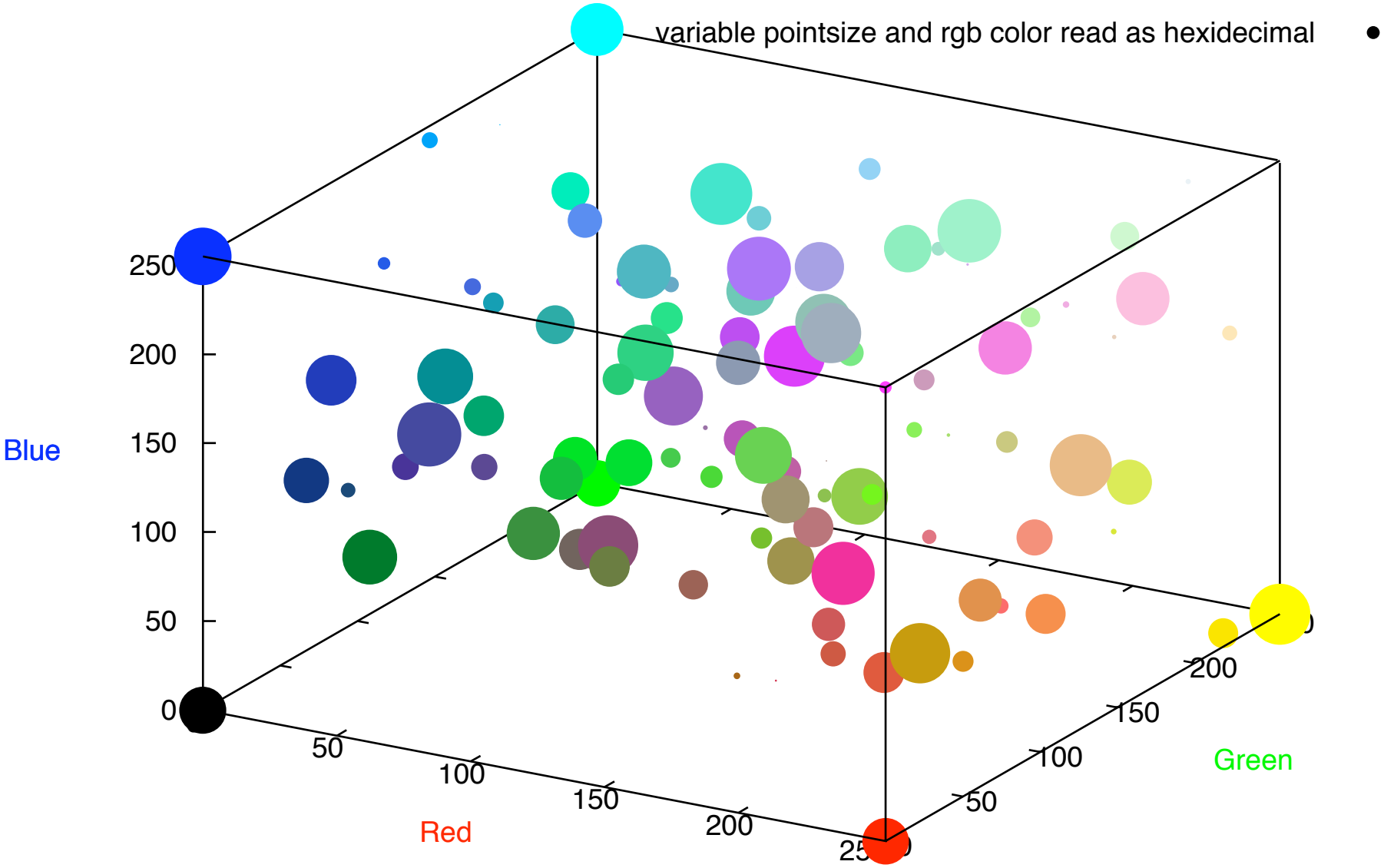
Terminal-independent RGB colors in 2D

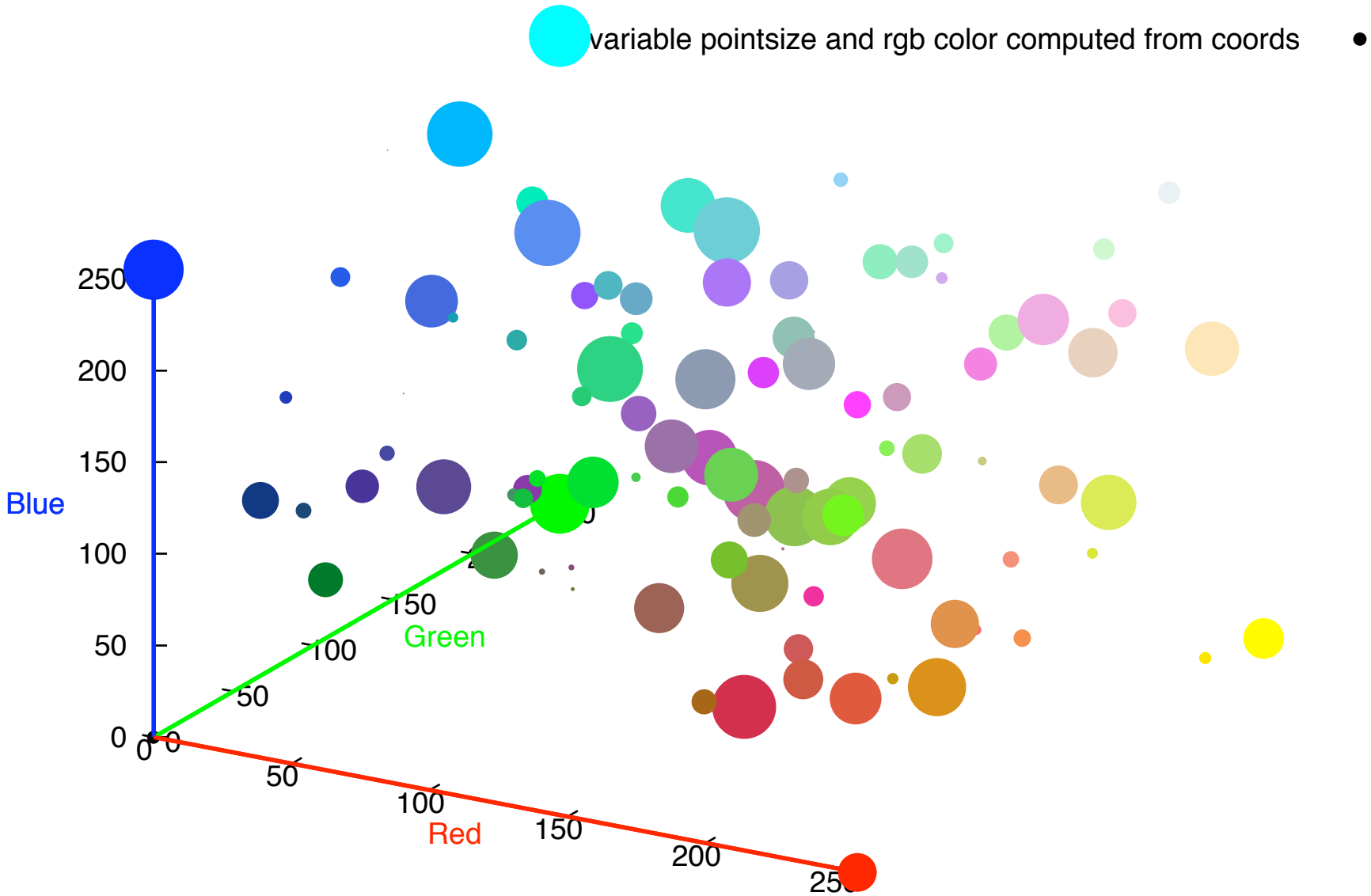


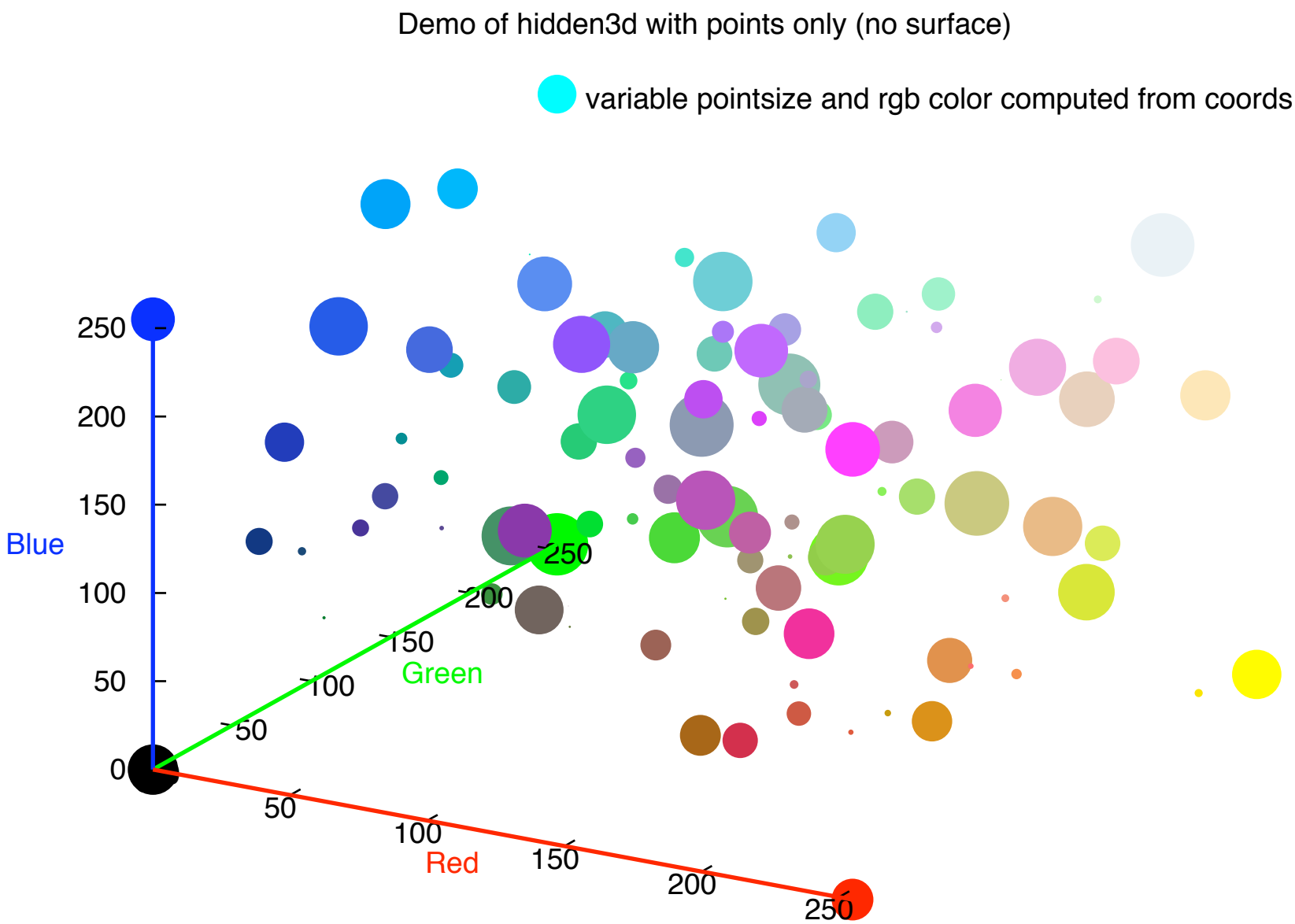
Implemented using built-in rgb color names
(only works for terminals that can do full rgb color)

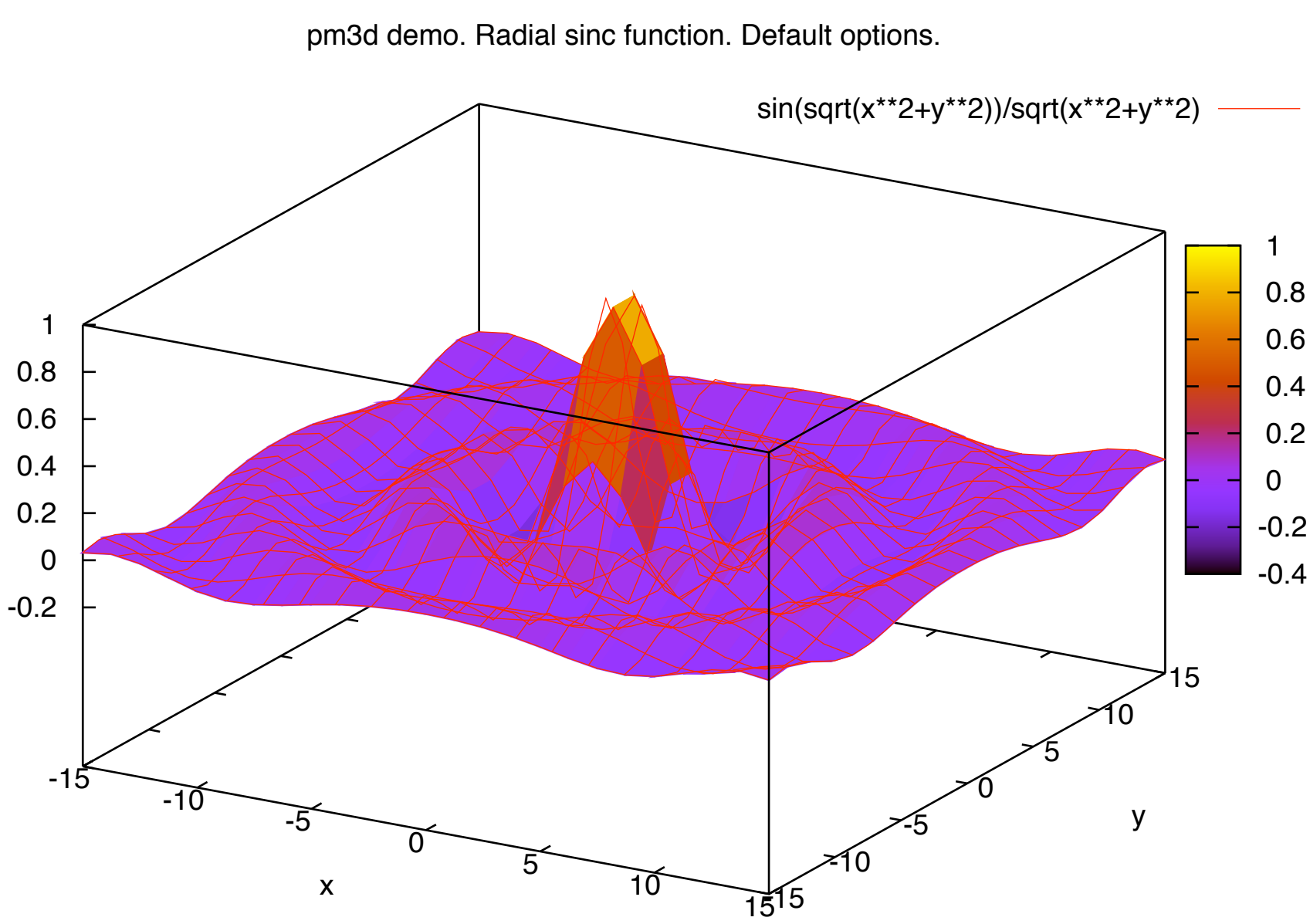

Terminal-independent palette colors in 2D
Implemented using command line macros referring to a fixed HSV palette
red
orange
yellow
green
cyan
blue
violet
HSV color wheel
0
0.2
0.4
0.6
0.8
1

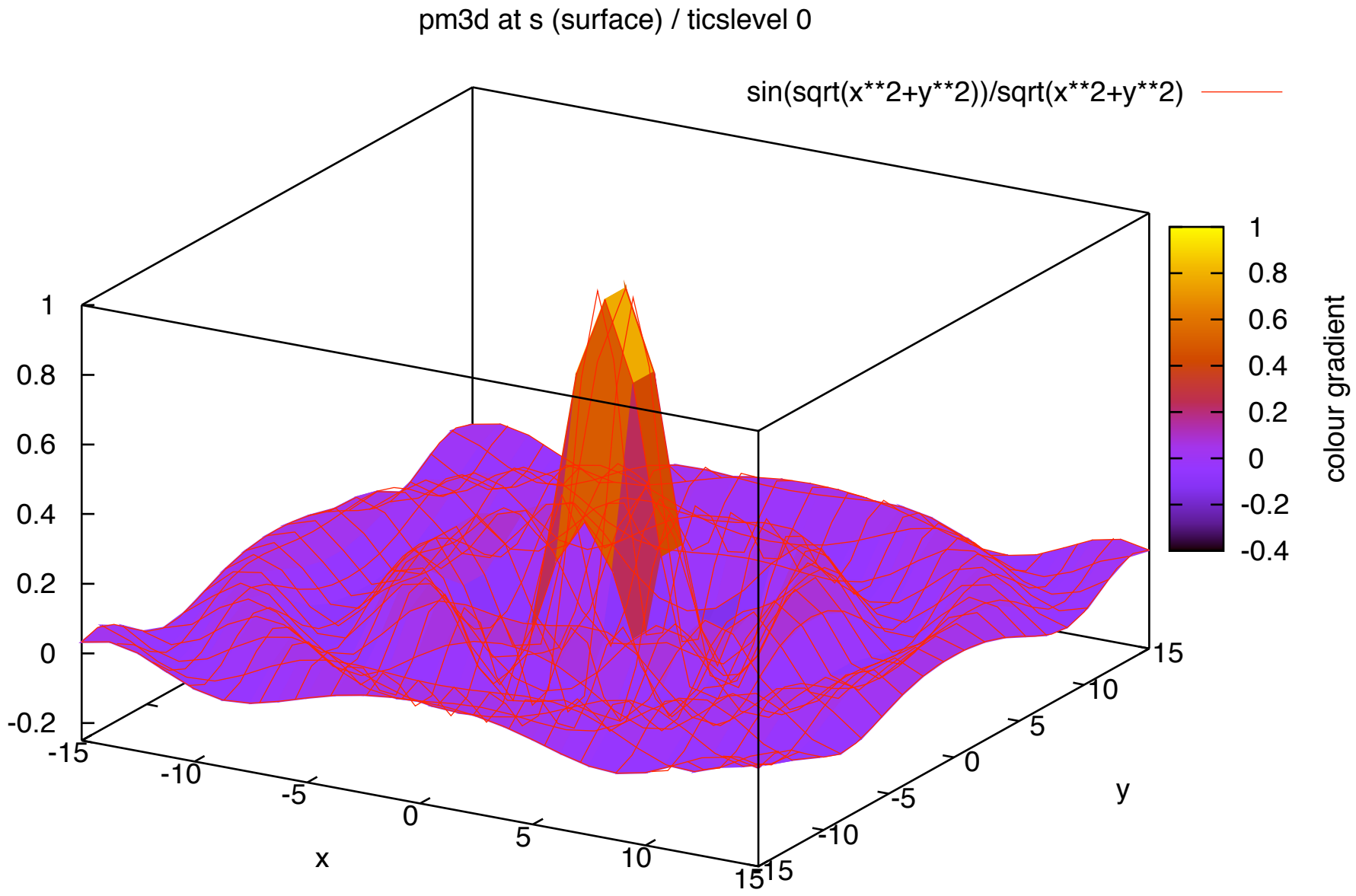

'rgb_variable.dat' using 1:2:3:(rgb($1,$2,$3))
Blue
Red
Green
0xffff
0xff
0x0
0xff0000
0xffff00
0xffffff
0x191ff8
0xea8f9
0x1ecae
0x4a77ed
0x3ee4c2
0x5ec5cd
0x85c9f2
0xe6eff4
0x92f3be
0xc8fac7
0xc796e9
0x1d43e2
0x3651c5
0x1a8fa5
0x299d96
0x2ce079
0xfaafd8
0xa7f692
0xe2c6b0
0xfde2ac
0xef69dc
0x1a26ae
0x137c83
0x2ac464
0x854ab2
0xc086ae
0x109a5d
0x363590
0x865d97
0xe4ad75
0xd3ea48
0x163866
0x6c20
0x318131
0x596d34
0x895046
0xd95e71
0xd0e52e
0x90833c
0xea098c
0xc34347
0xfae302
0xd64830
0

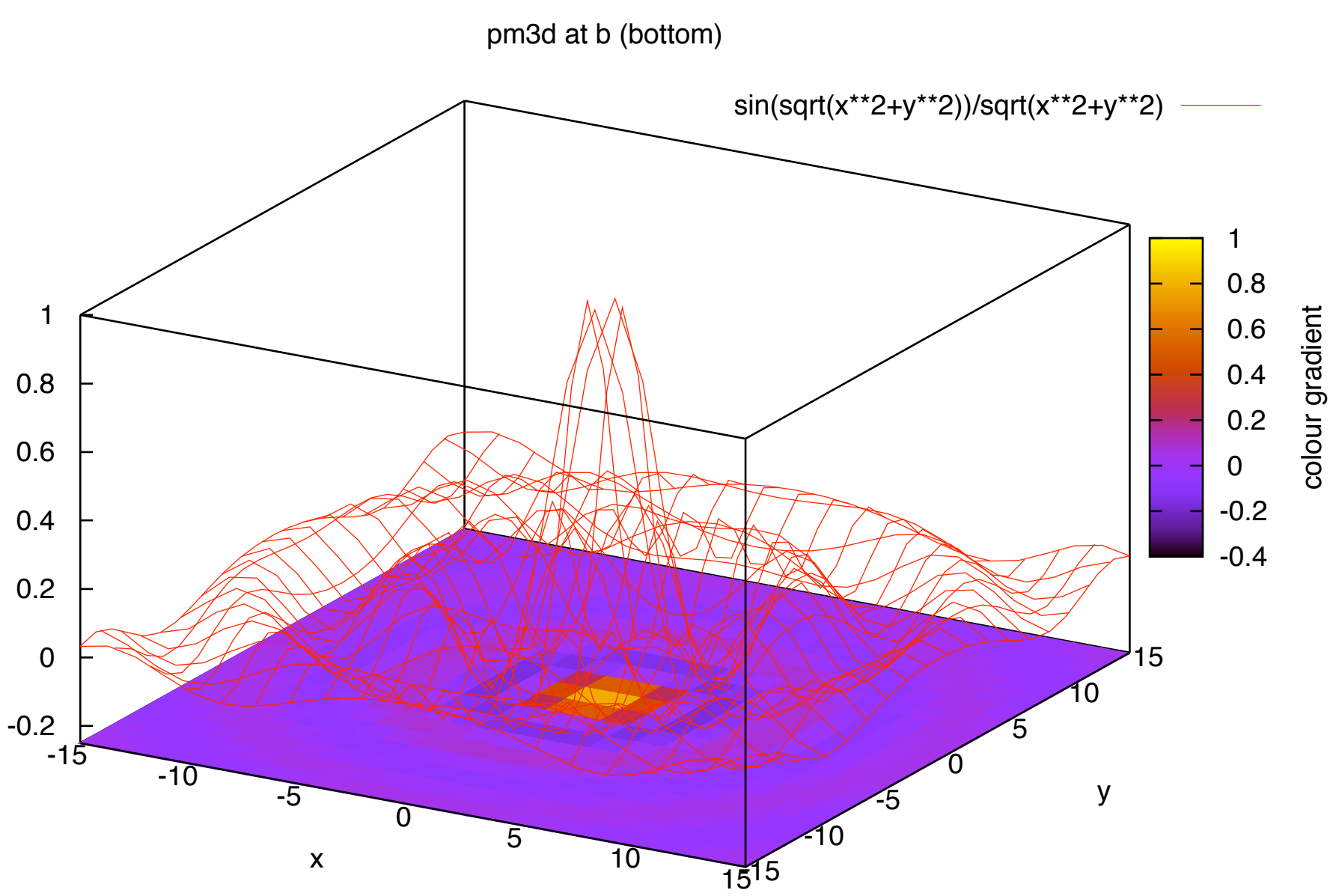

variable pointsize and rgb color read as hexidecimal
Blue
250
200
150
100
50
0
50
100
150
200
Red
50
100
150
200
Green

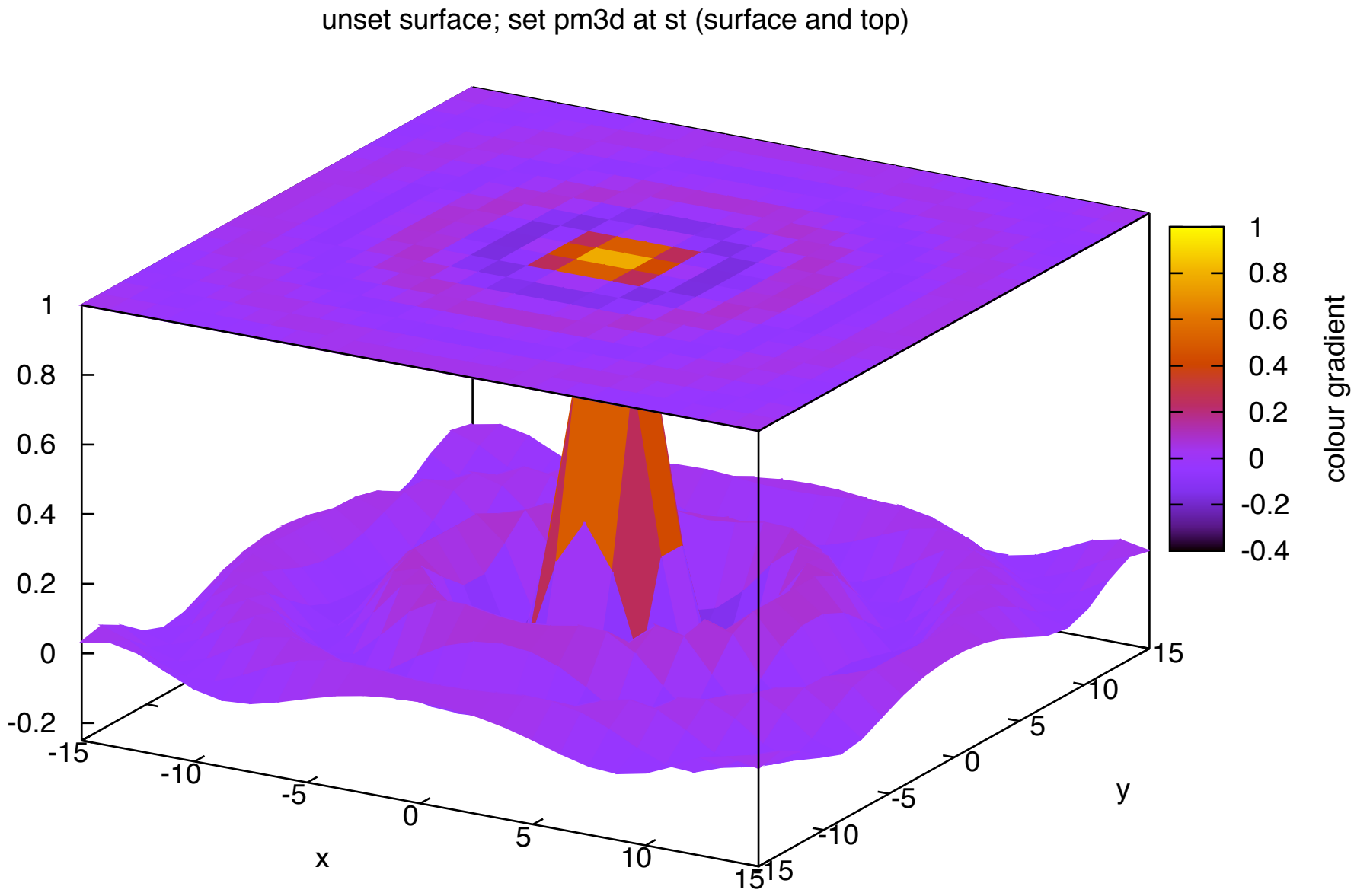

variable pointsize and rgb color computed from coords
Blue
250
200
150
100
50
0
Green
150
100
50
0
Red
50
100
150
200

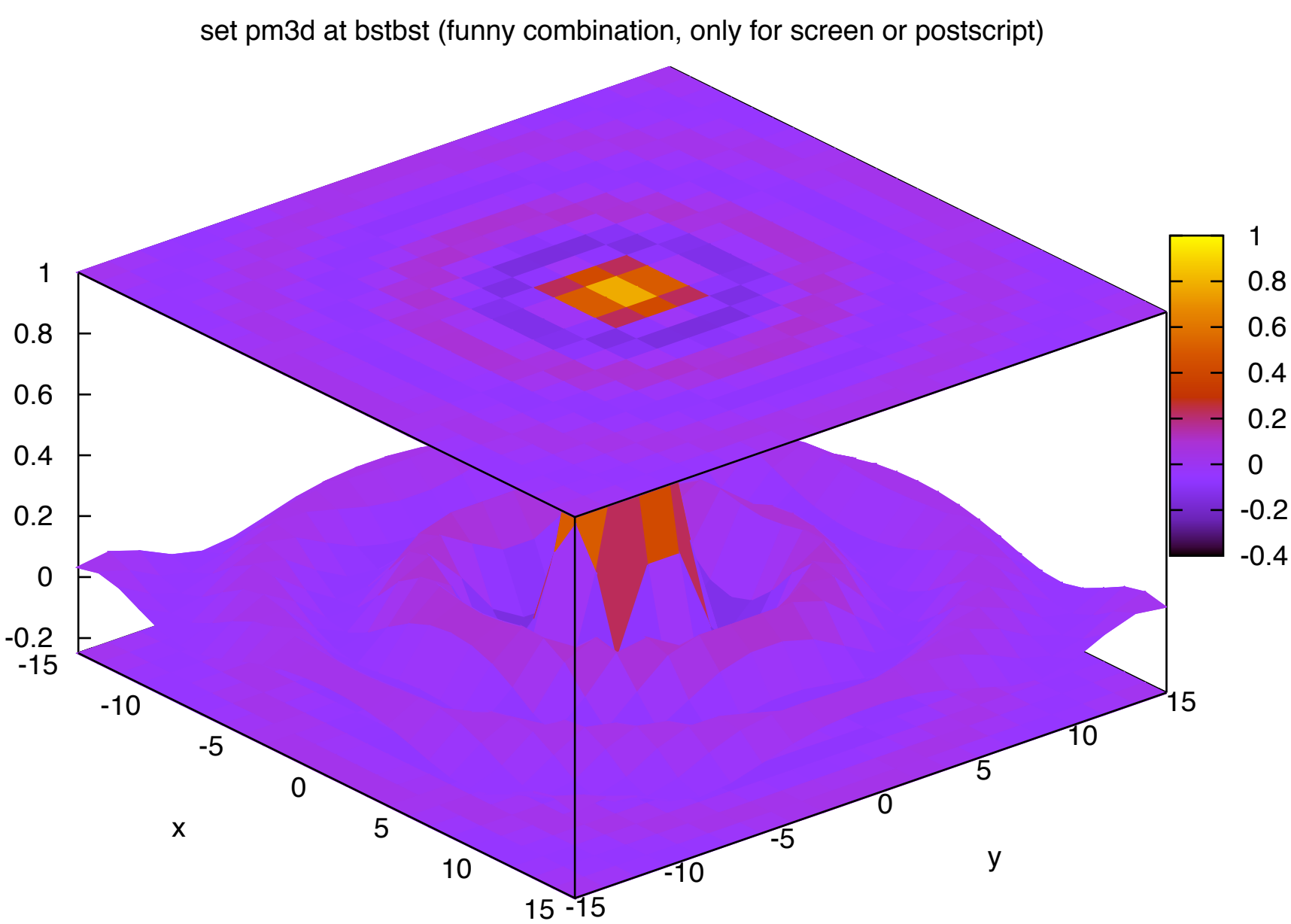

Demo of hidden3d with points only (no surface)
variable pointsize and rgb color computed from coords
Blue
Green
Red
0
50
100
150
200
250

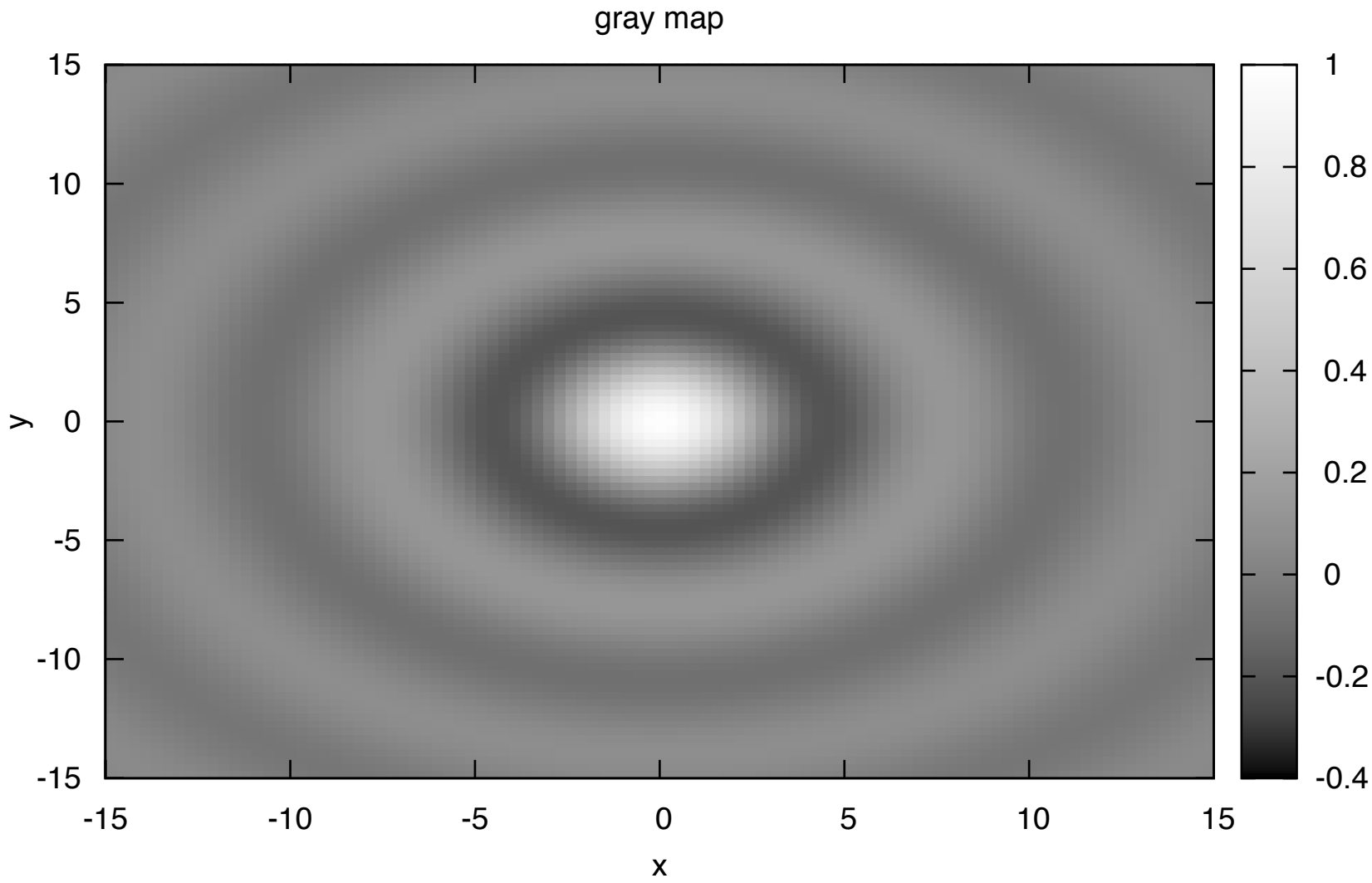

pm3d demo. Radial sinc function. Default options.
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x
y

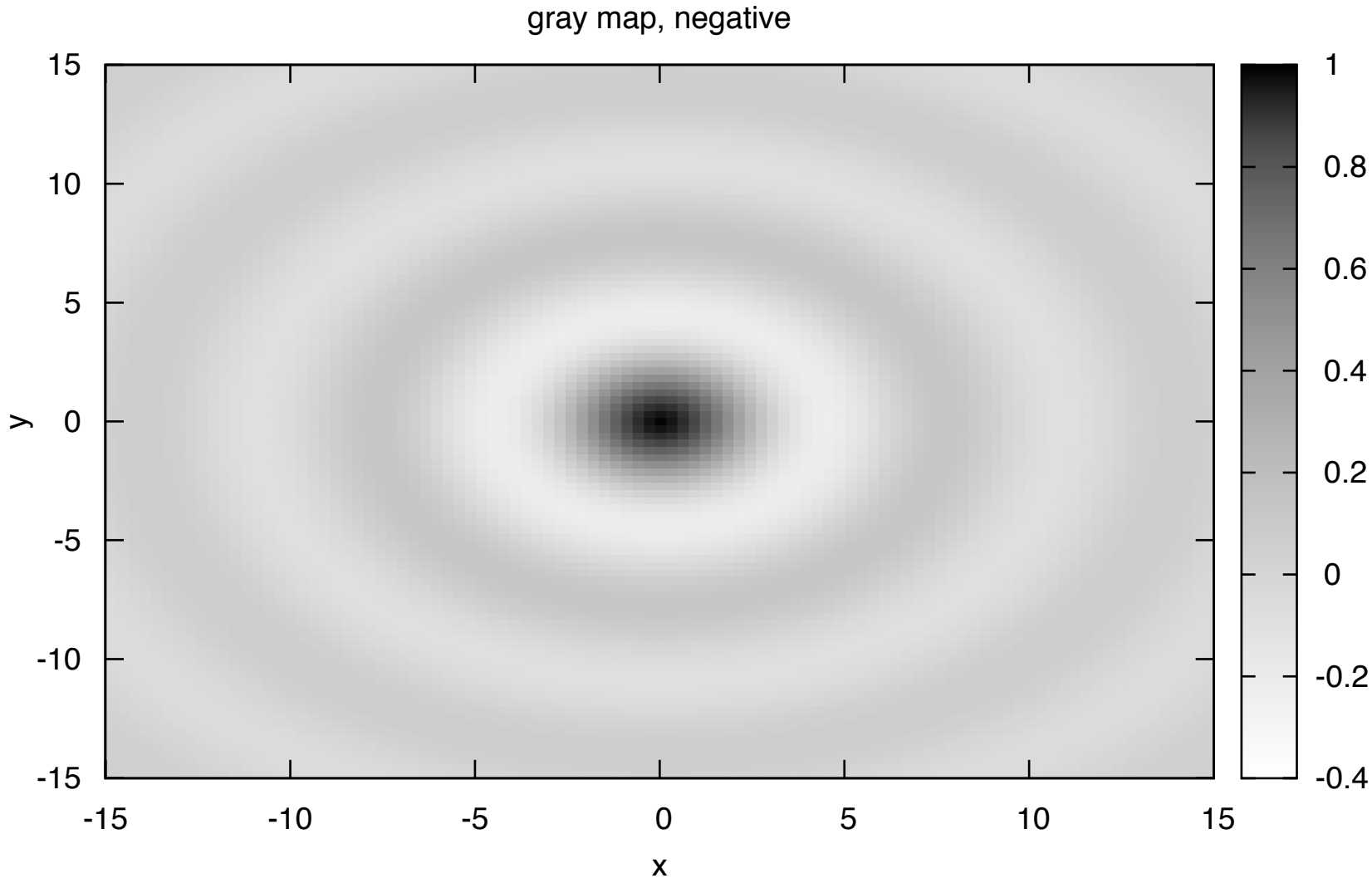

pm3d at s (surface) / ticslevel 0
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x
y
colour gradient

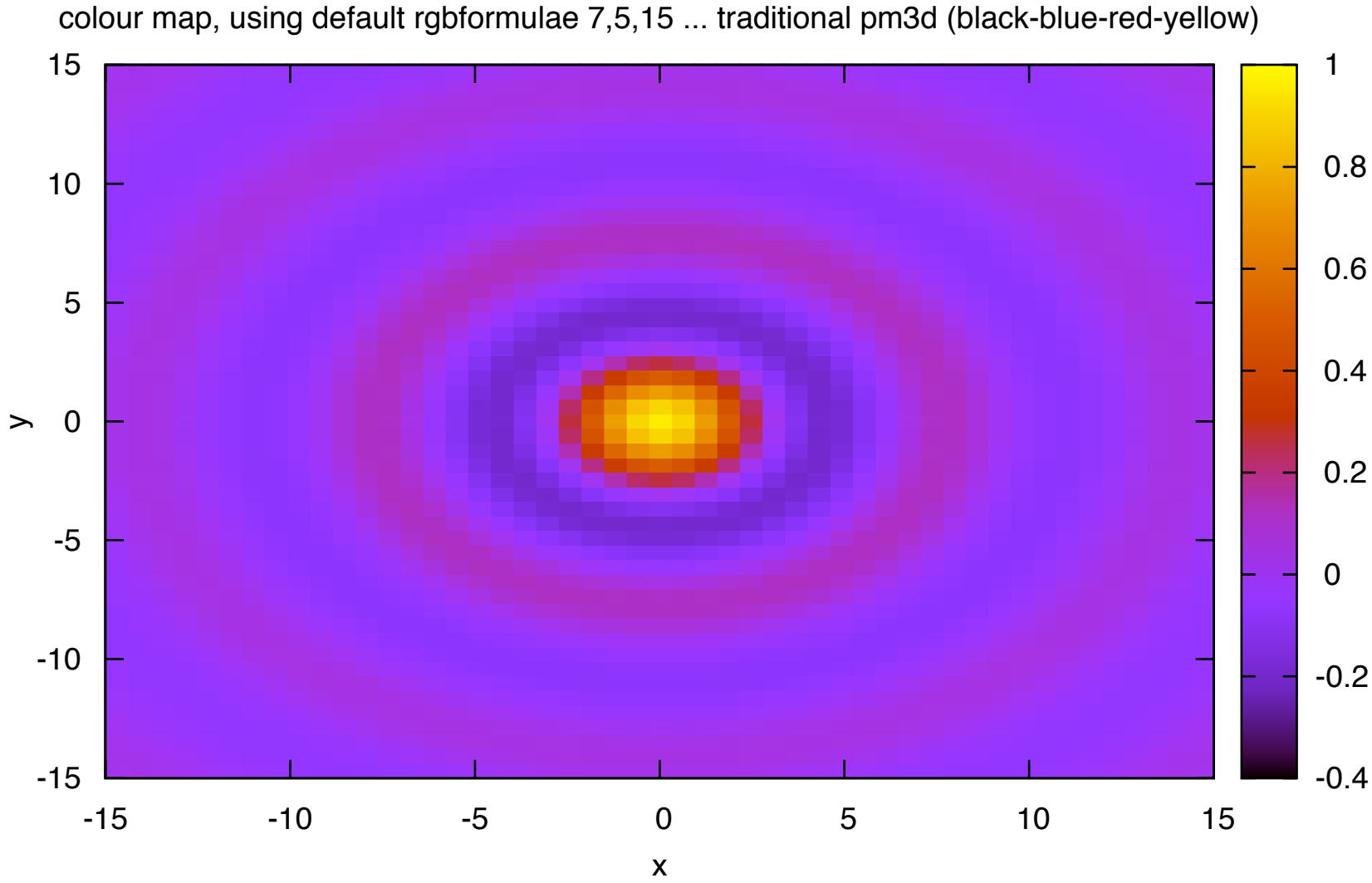

pm3d at b (bottom)
sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x
y
colour gradient

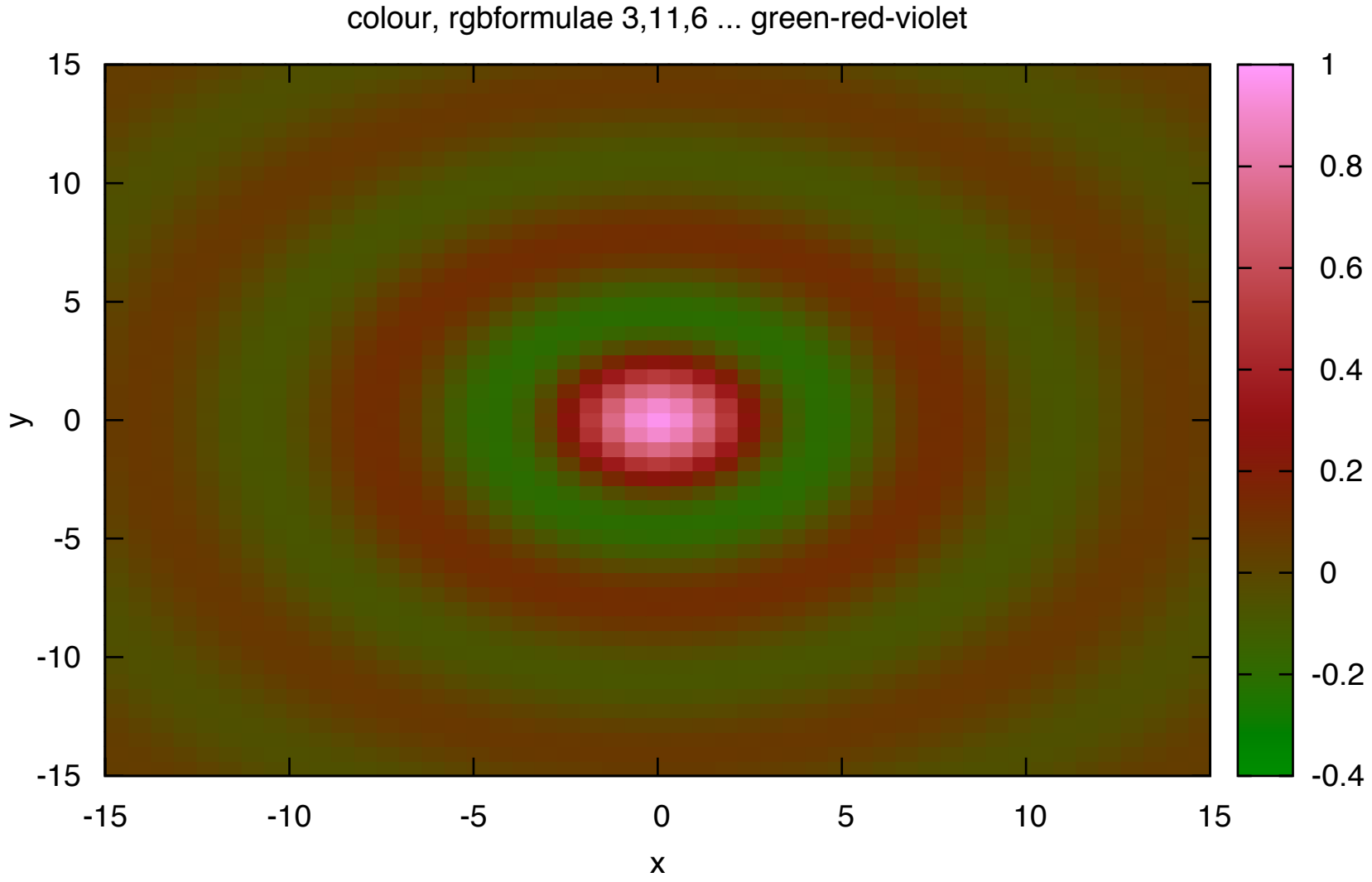

unset surface; set pm3d at st (surface and top)
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x
y
colour gradient

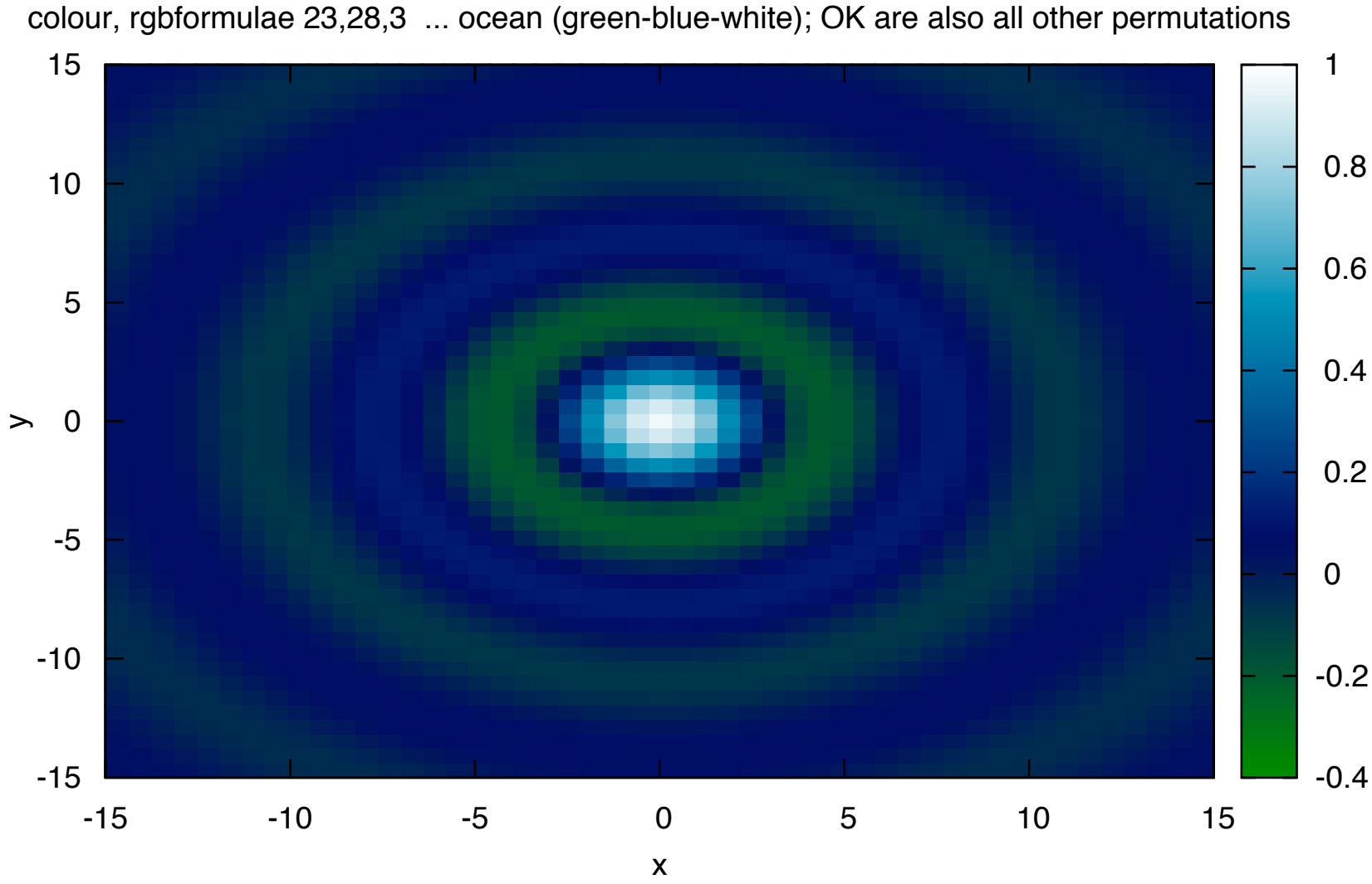

set pm3d at bstbst (funny combination, only for screen or postscript)
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x
y
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

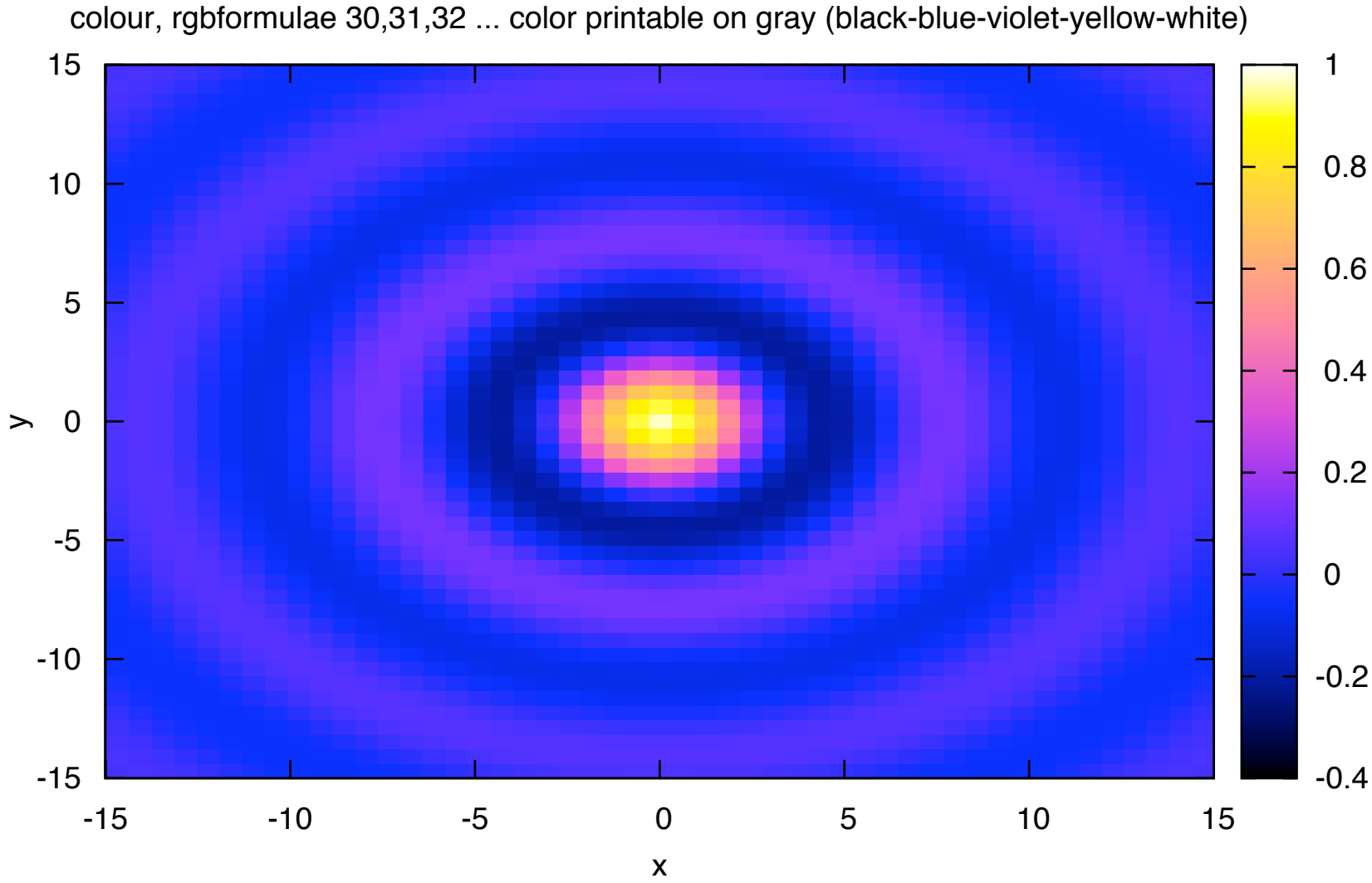

gray map
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

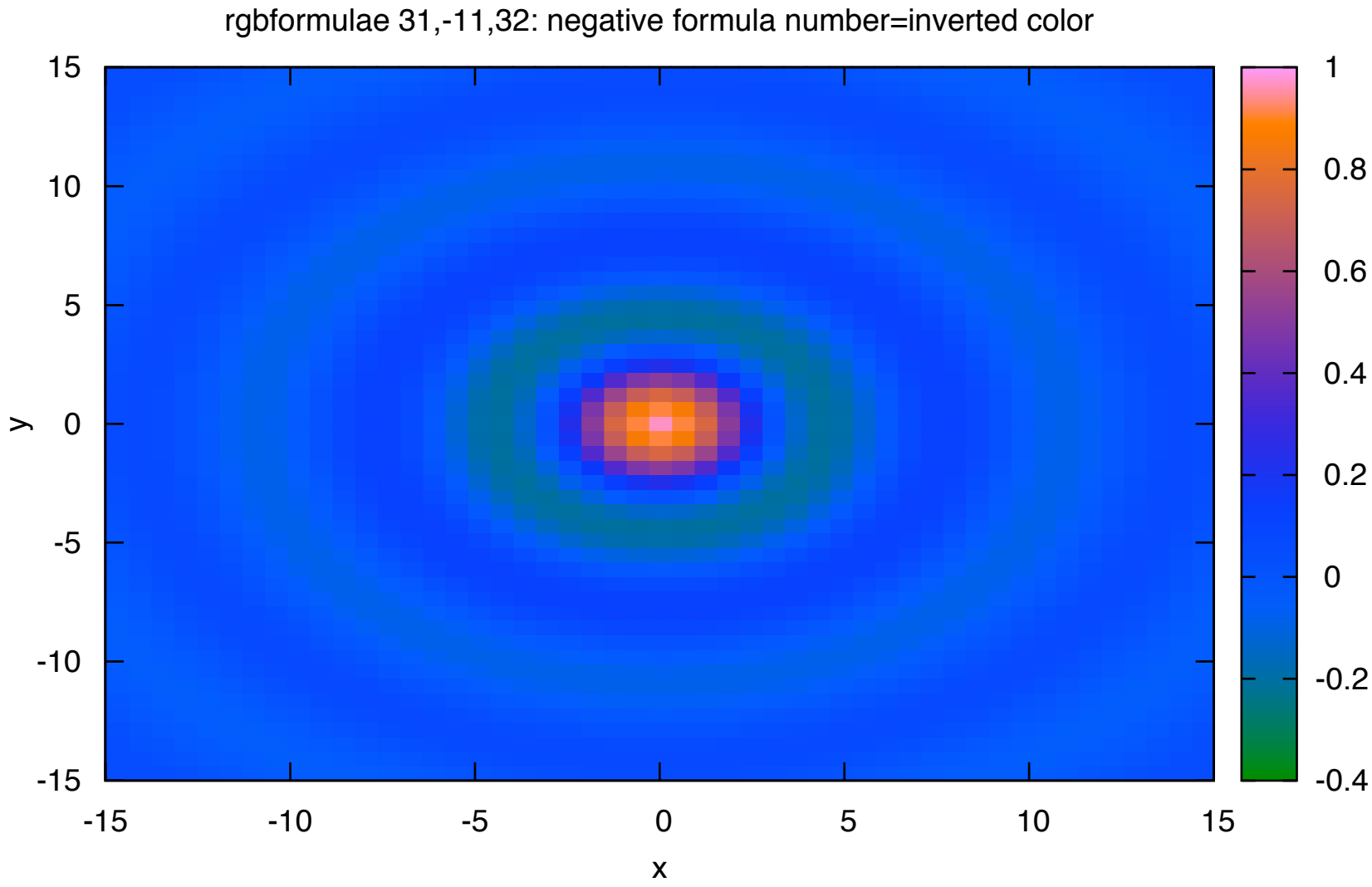

gray map, negative
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

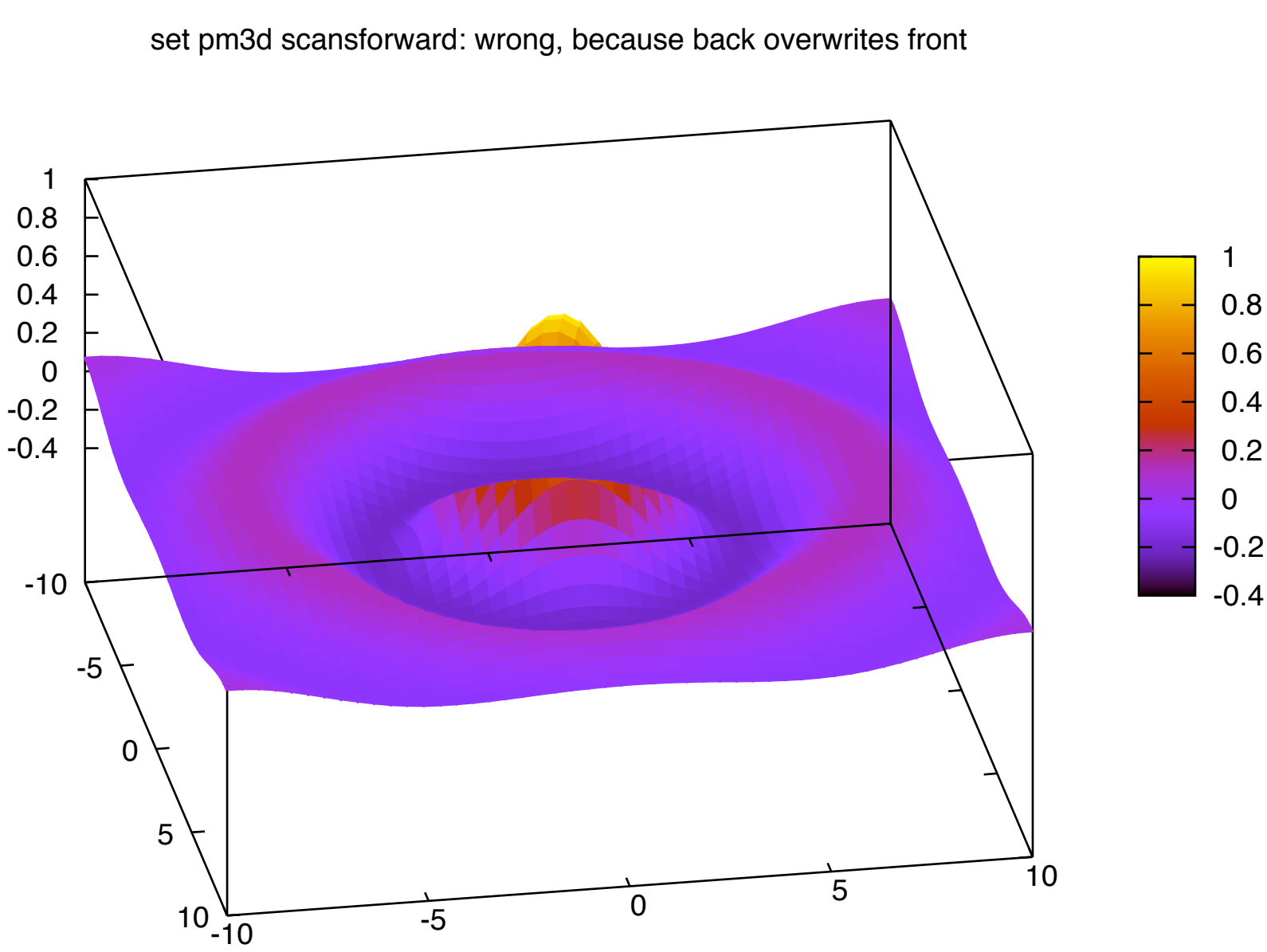

colour map, using default rgbformulae 7,5,15 ... traditional pm3d (black-blue-red-yellow)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

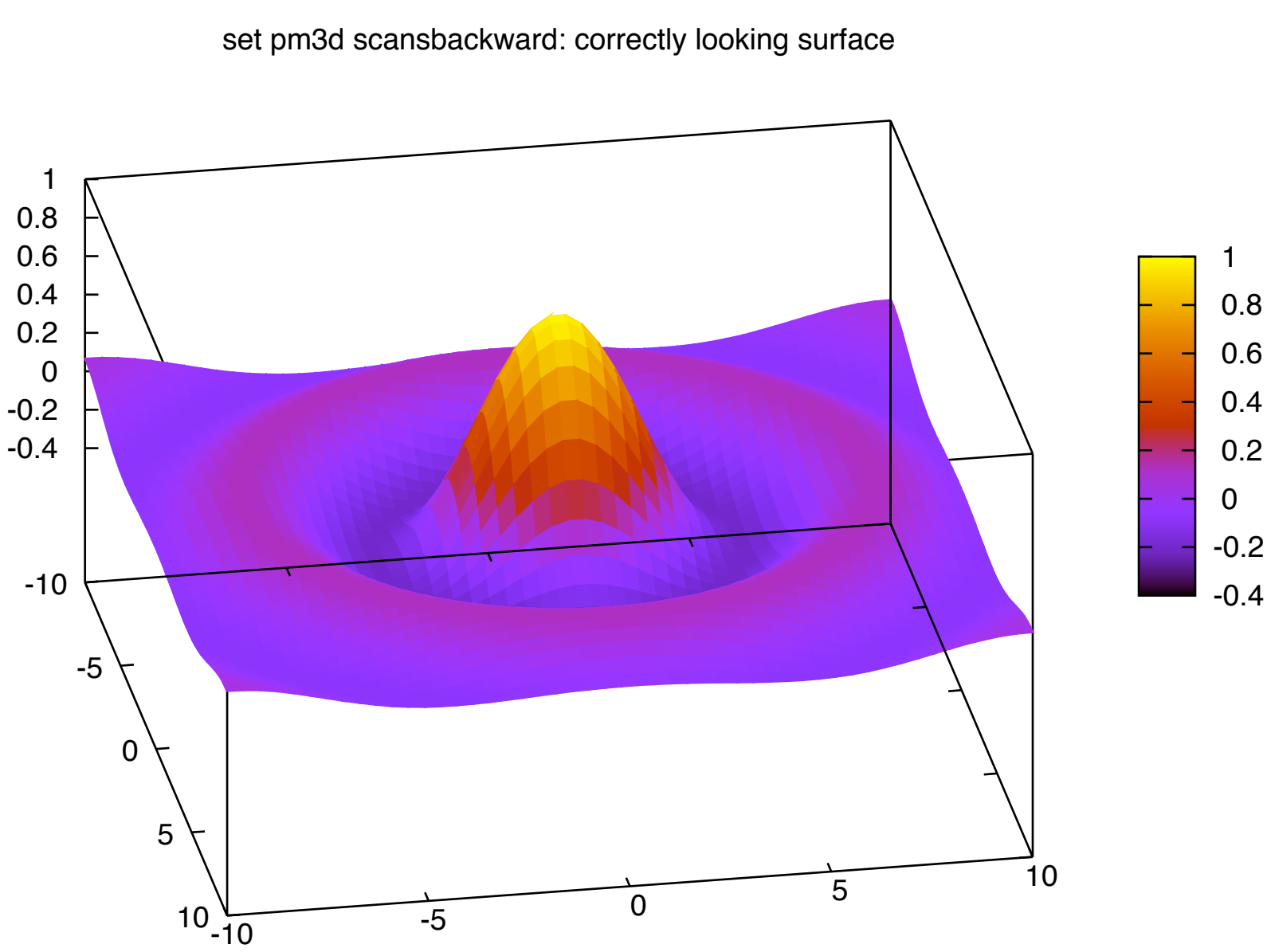

colour, rgbformulae 3,11,6 ... green-red-violet
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

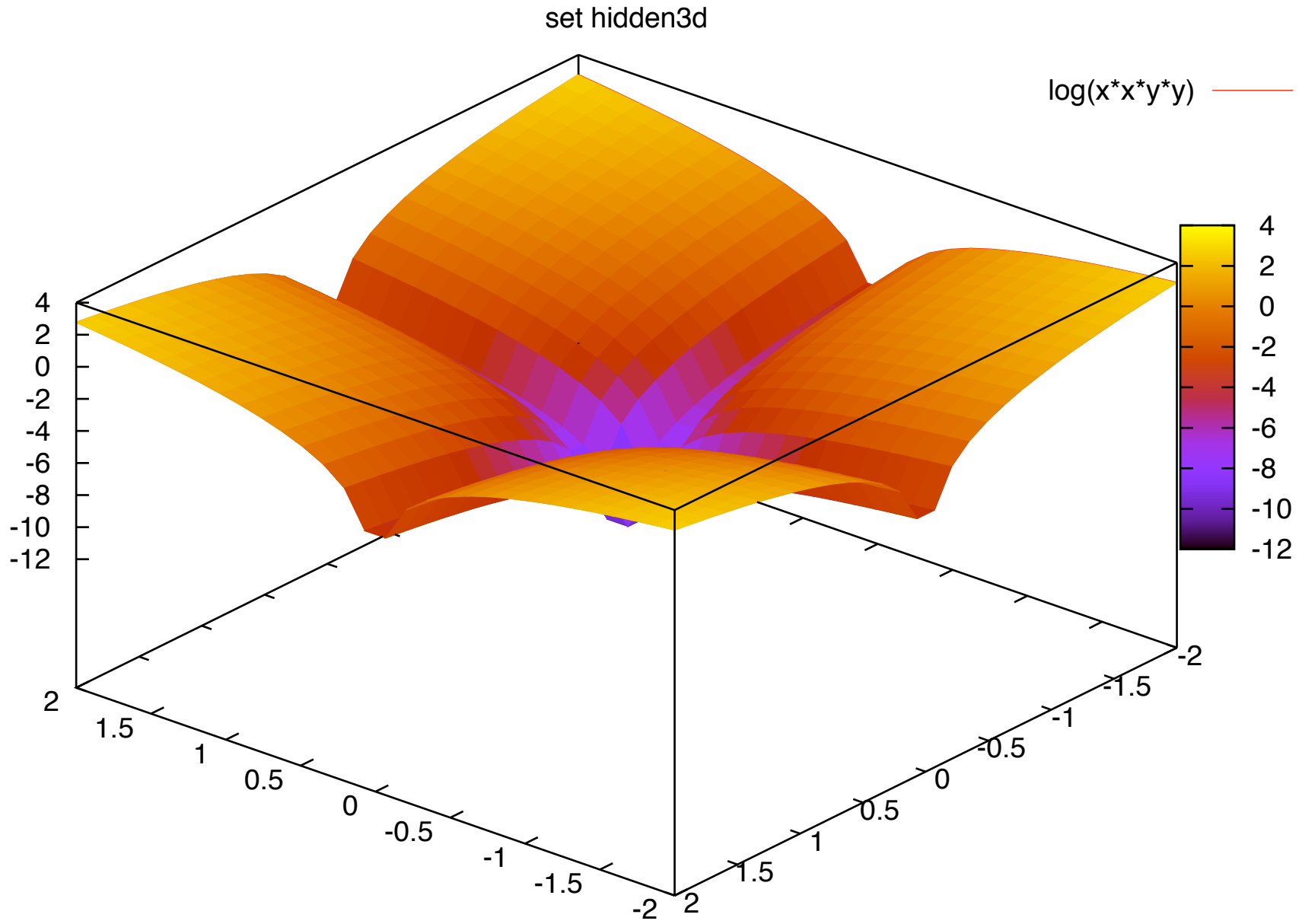

colour, rgbformulae 23,28,3 ... ocean (green-blue-white); OK are also all other permutations
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

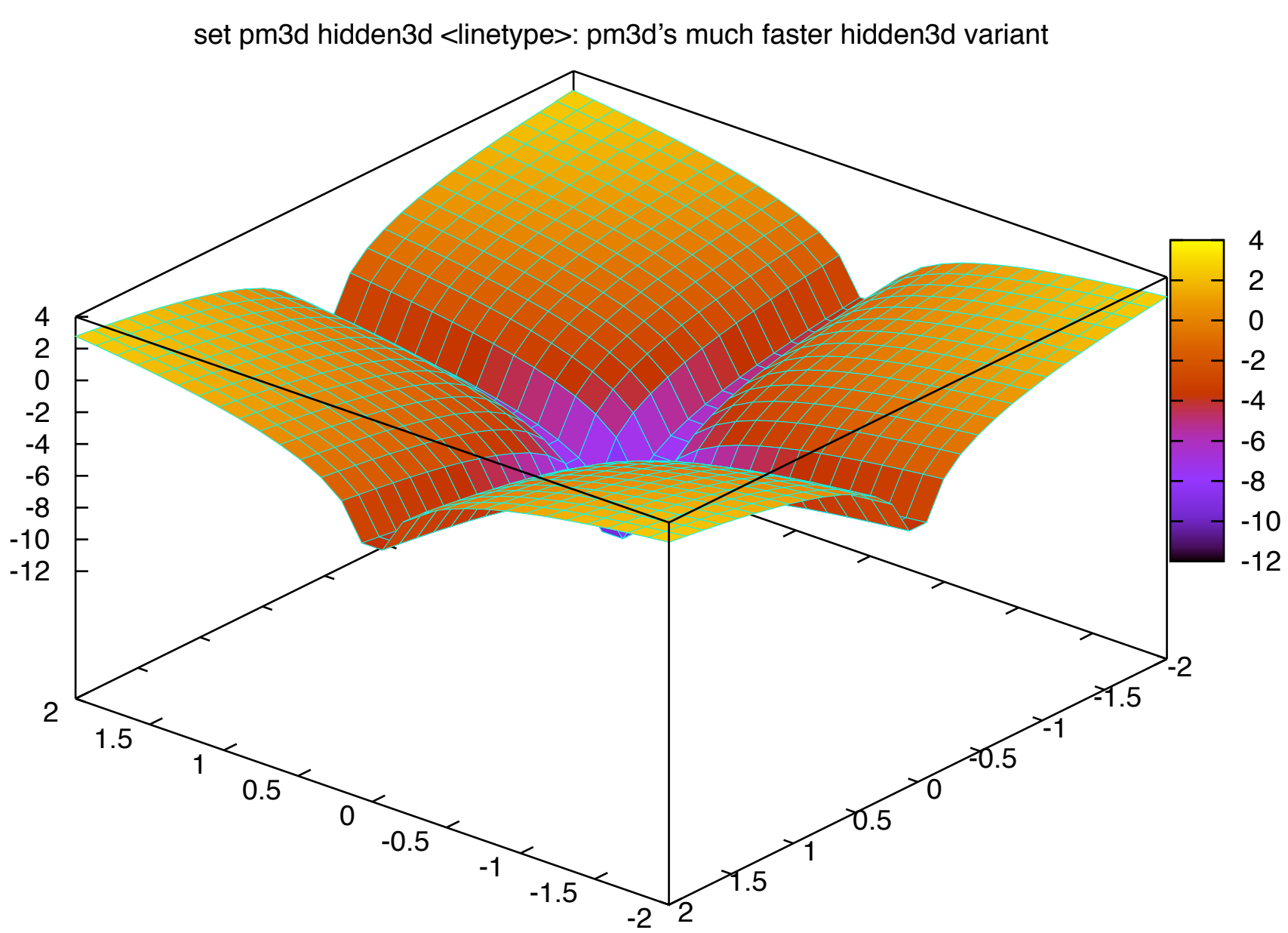

colour, rgbformulae 30,31,32 ... color printable on gray (black-blue-violet-yellow-white)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

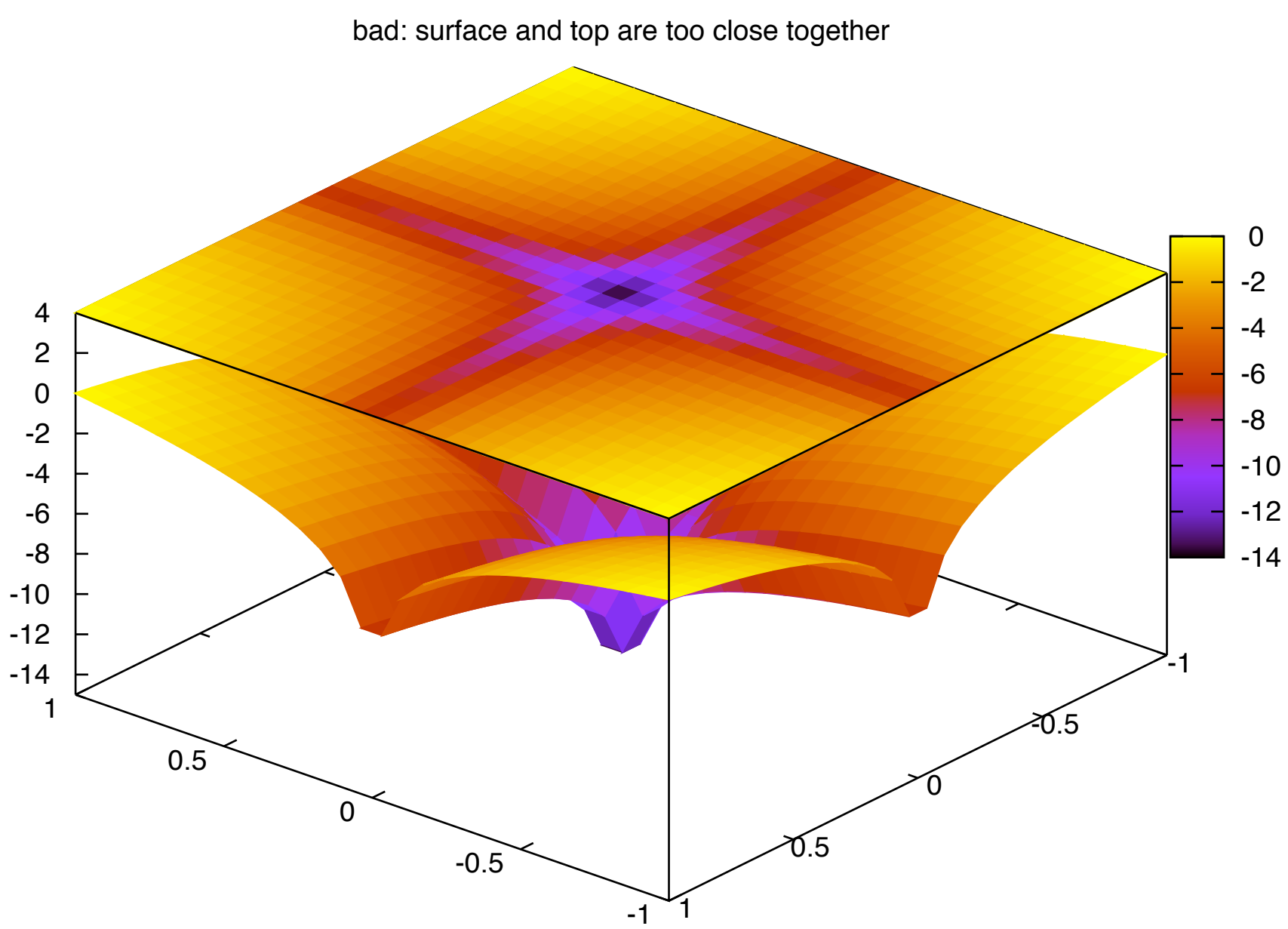

rgbformulae 31,-11,32: negative formula number=inverted color
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

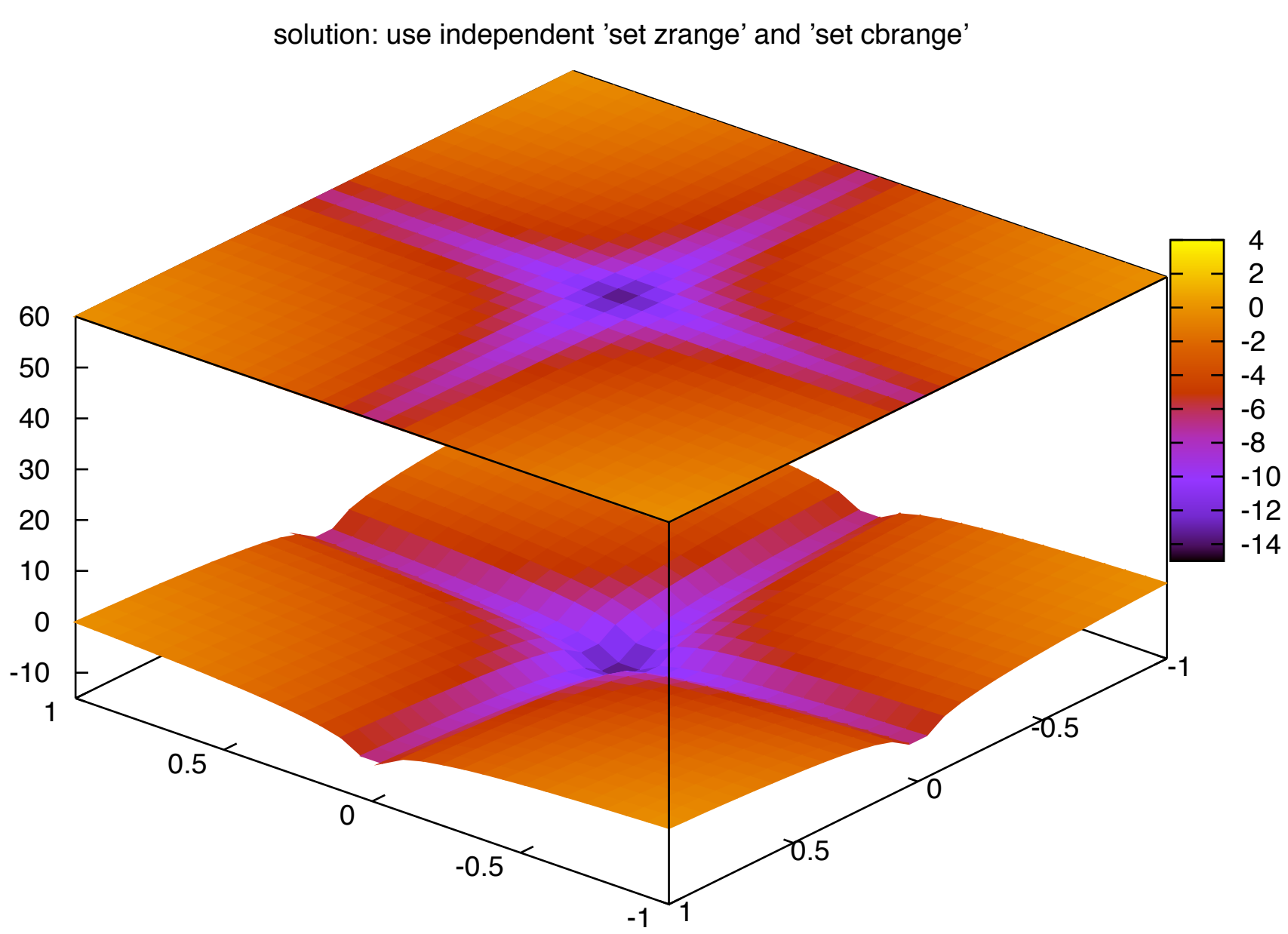

set pm3d scansforward: wrong, because back overwrites front
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10
-10
-5
0
5
10
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

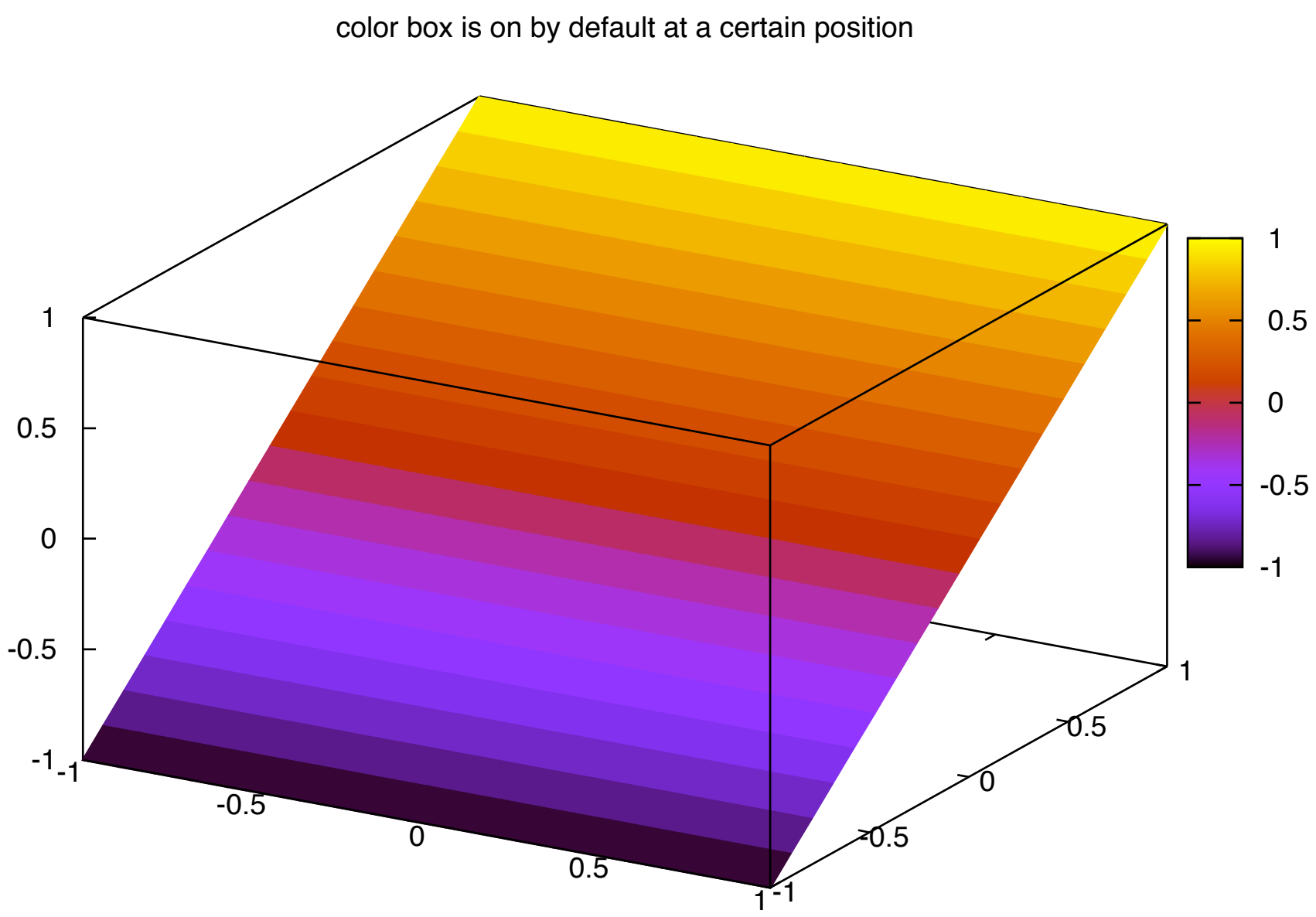

set pm3d scansbackward: correctly looking surface
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10
-10
-5
0
5
10
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4


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

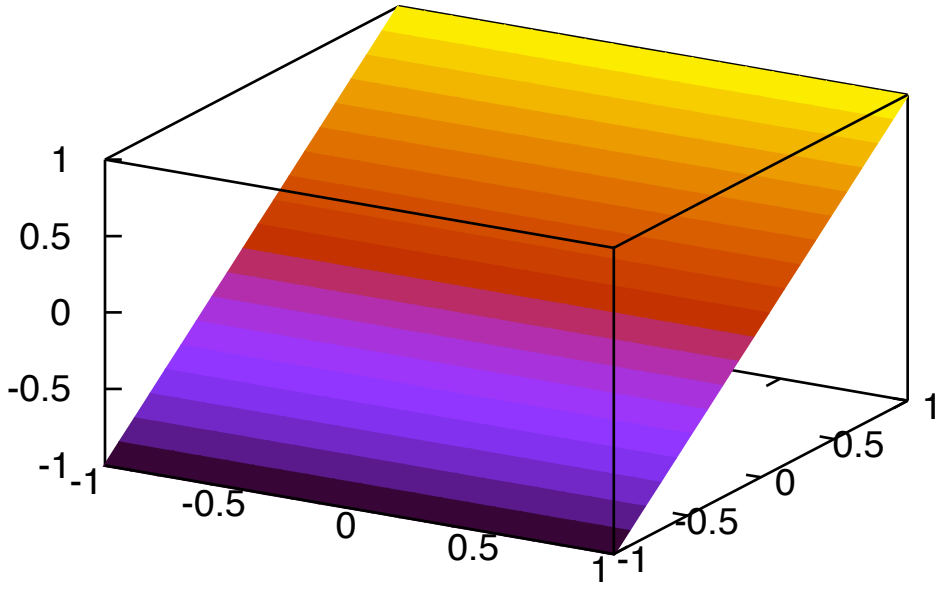

set pm3d hidden3d <linetype>: pm3d's much faster hidden3d variant
4
2
0
-2
-4
-6
-8
-10
-12
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
4
2
0
-2
-4
-6
-8
-10
-12

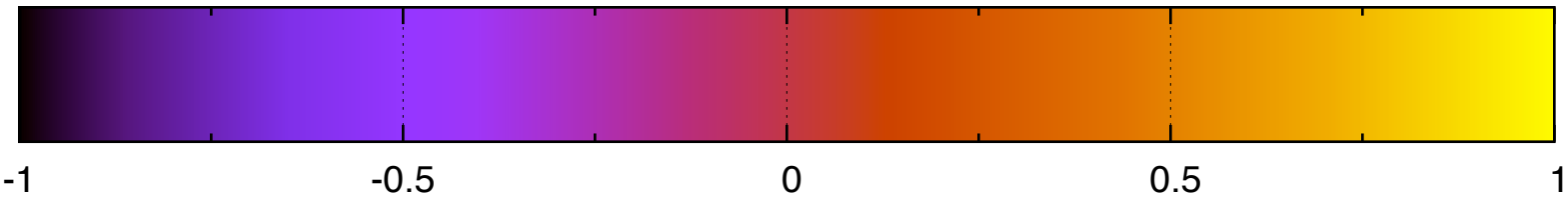

bad: surface and top are too close together
4
2
0
-2
-4
-6
-8
-10
-12
-14
1
0.5
0
-0.5
-1
1
0.5
0
-0.5
-1
0
-2
-4
-6
-8
-10
-12
-14


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

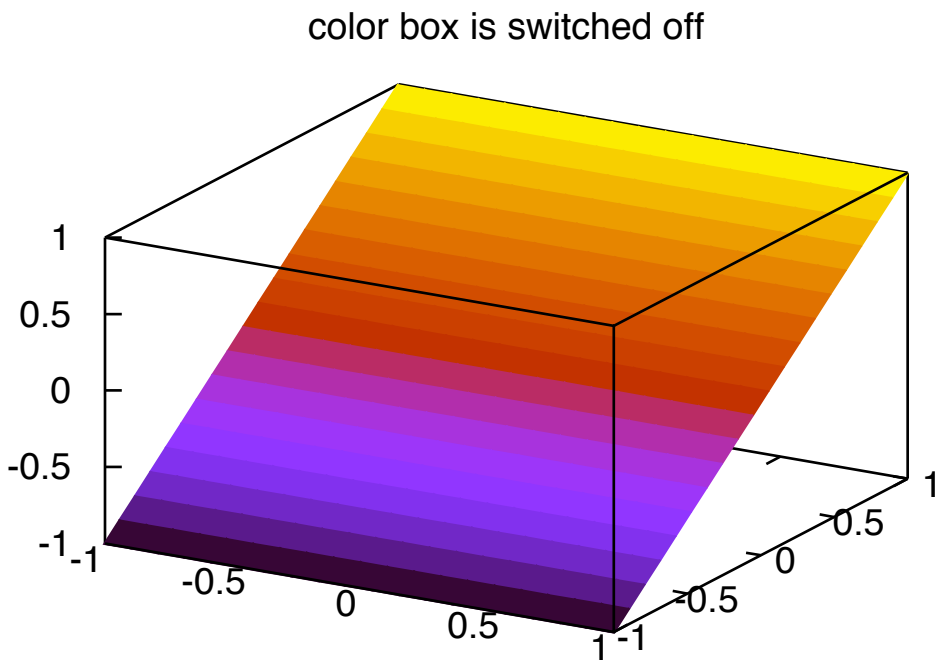

color box is on by default at a certain position
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1
1
0.5
0
-0.5
-1

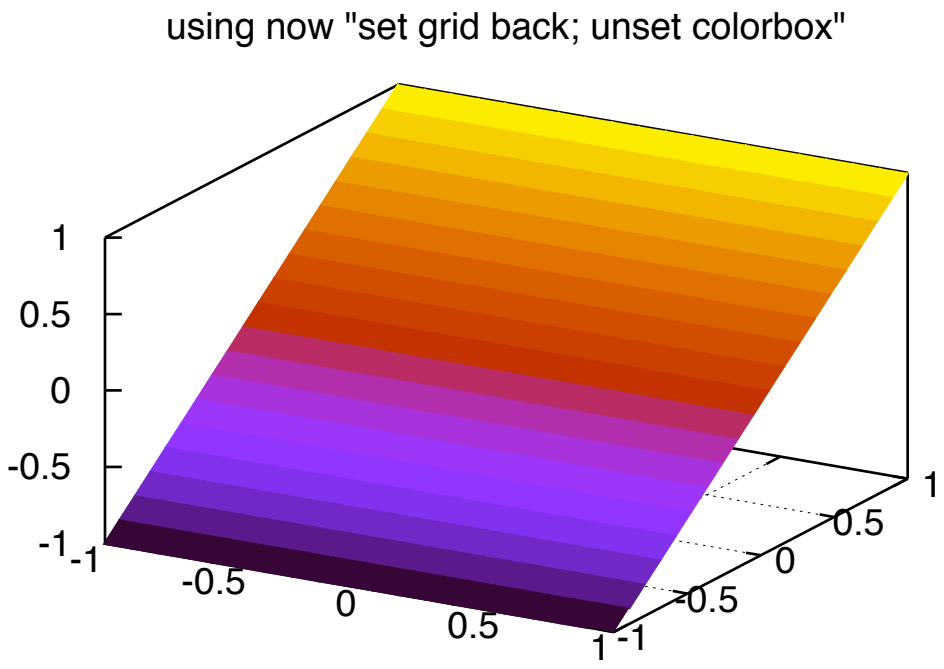
color box is on again, now with horizontal gradient

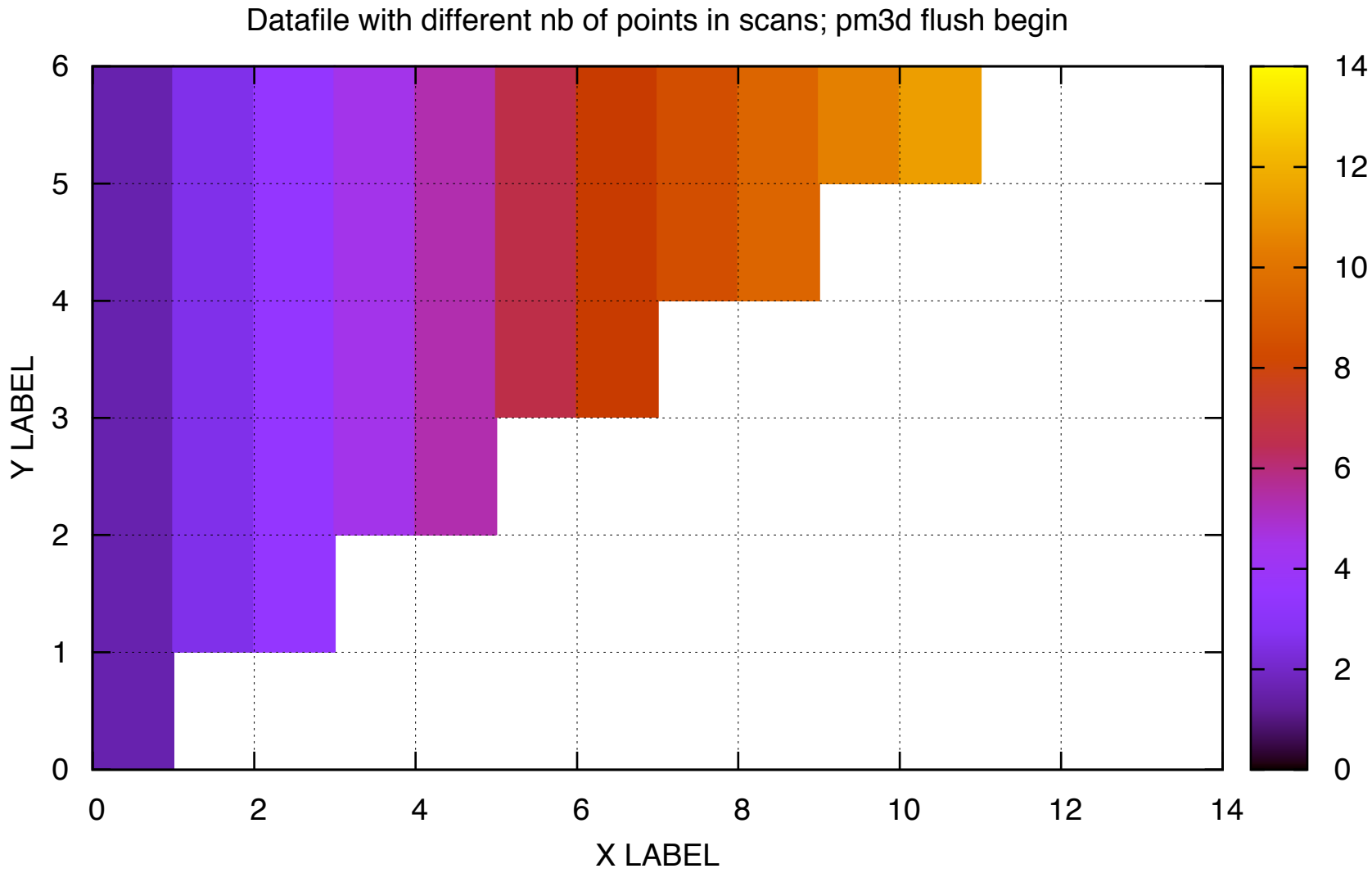


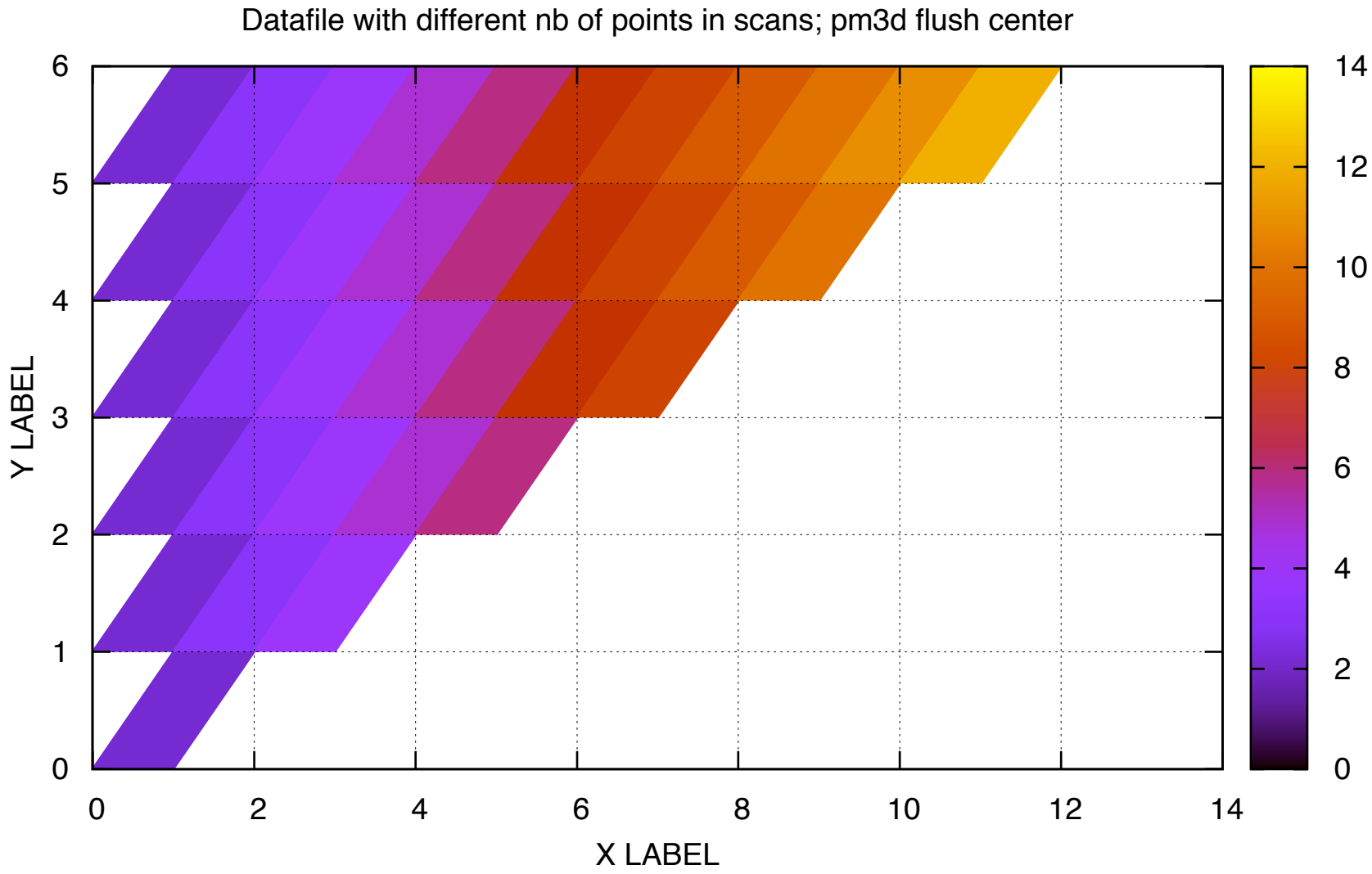


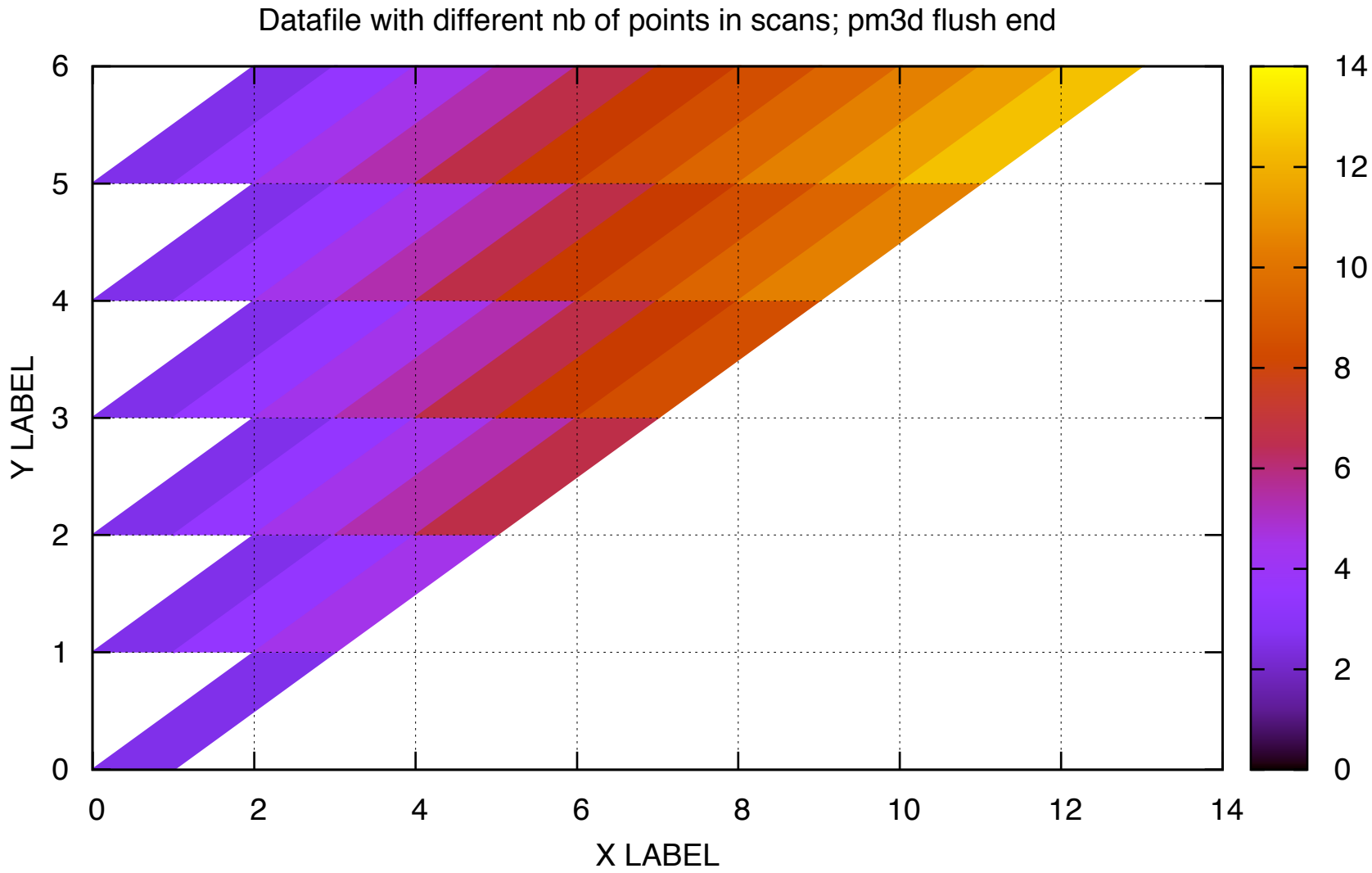
see cblabel, grid cb, mcbtics, ...

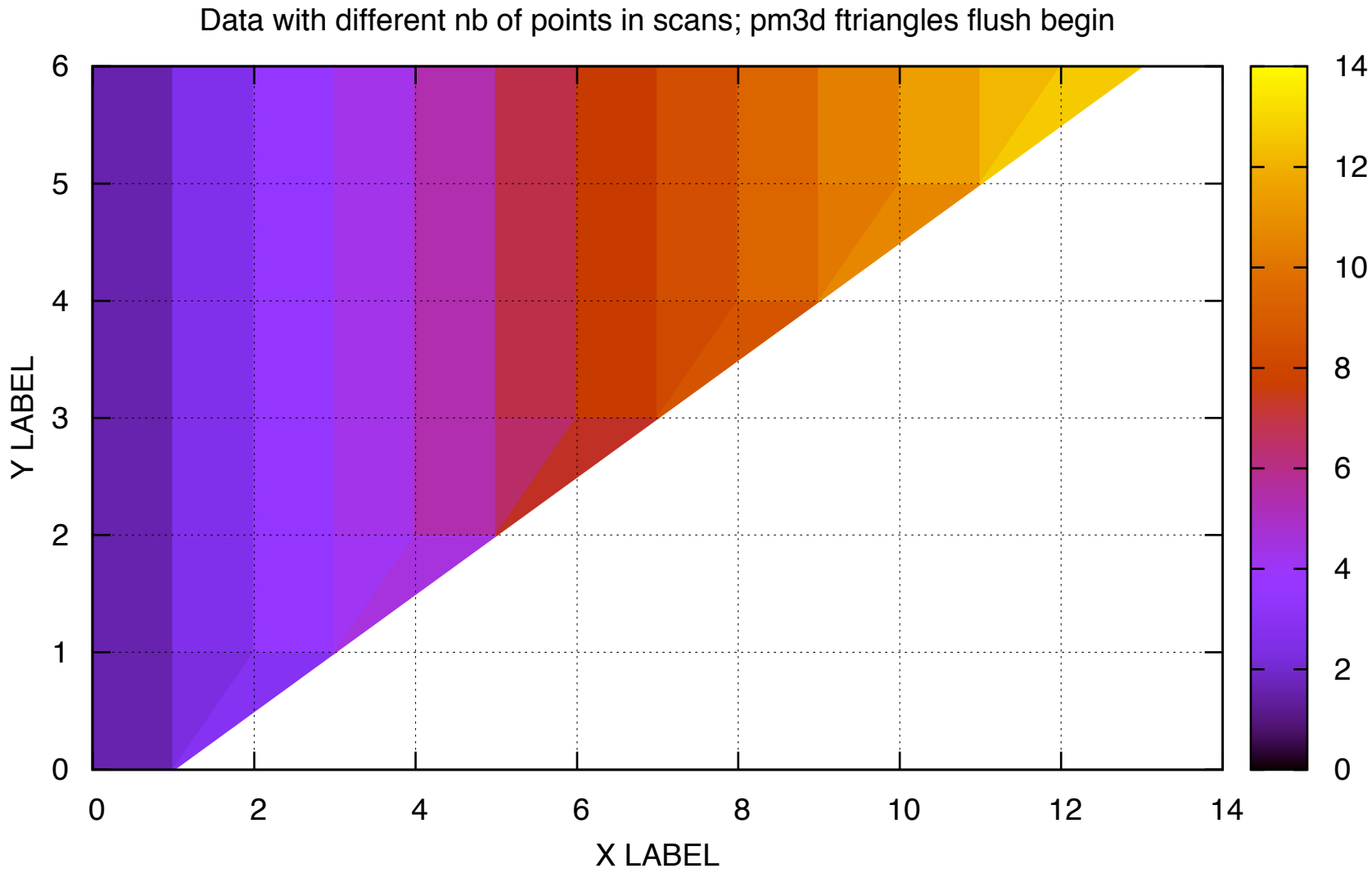

color box is switched off
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1

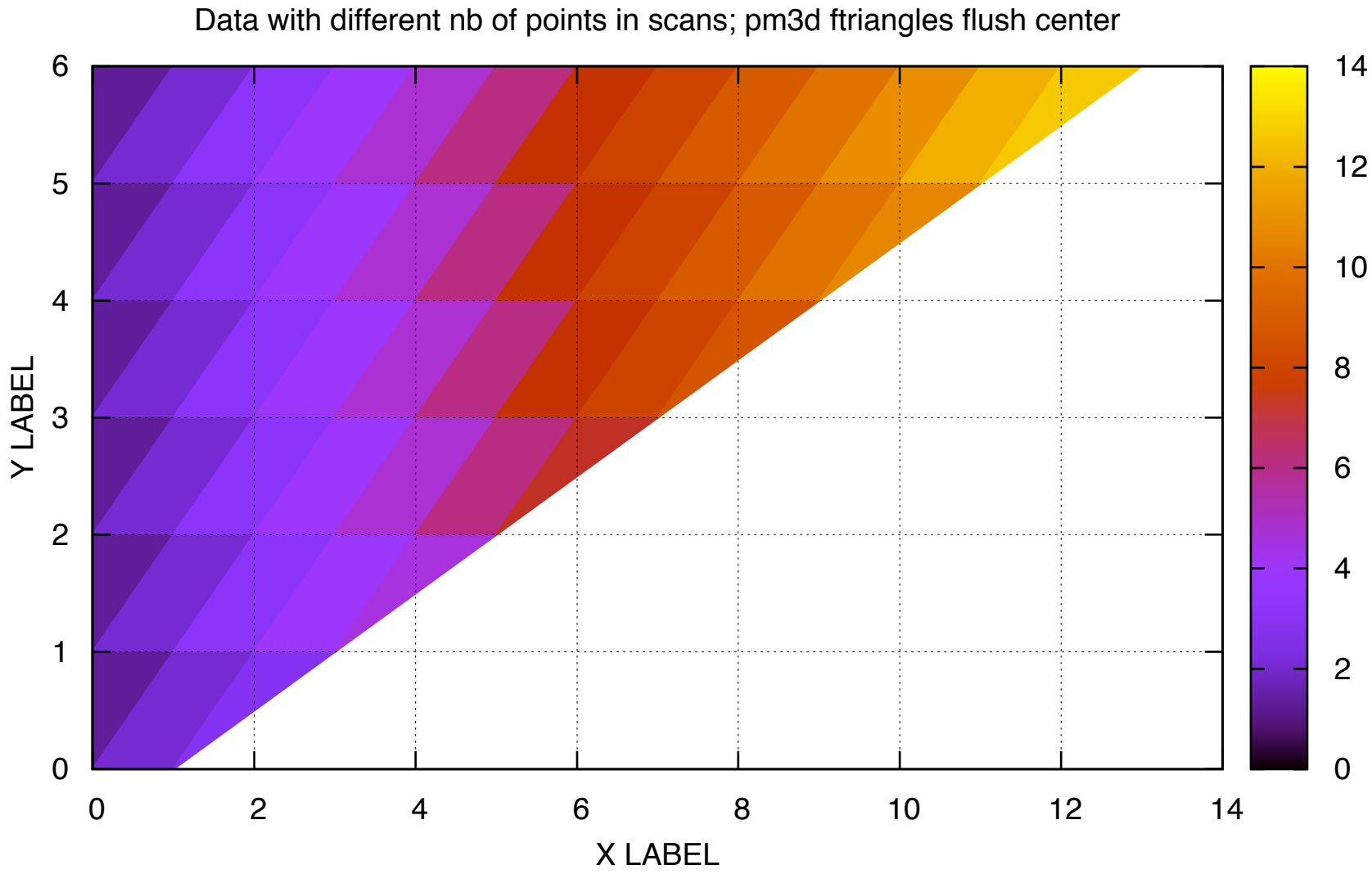

using now "set grid back; unset colorbox"
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1

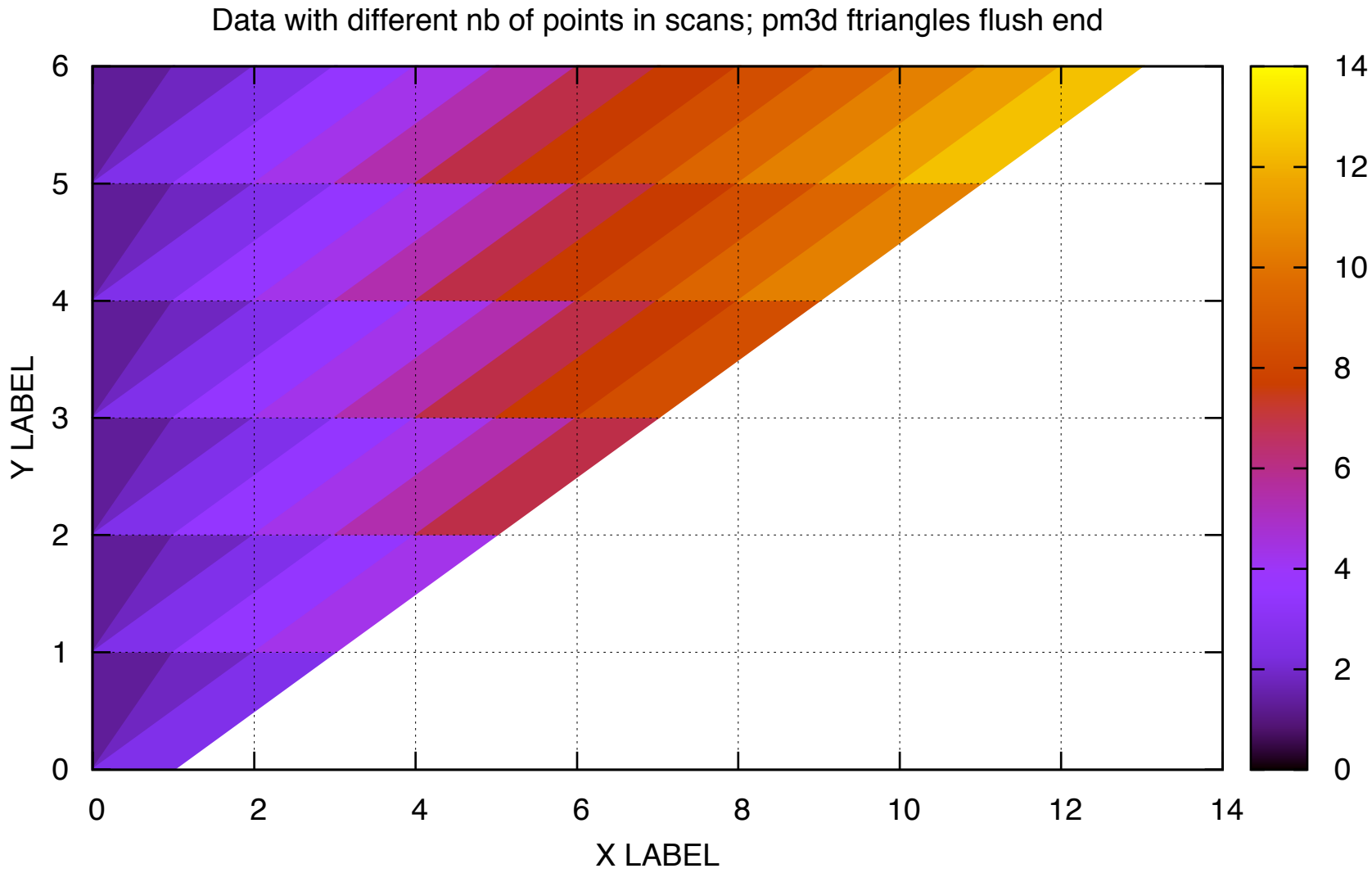

Datafile with different nb of points in scans; pm3d flush begin
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

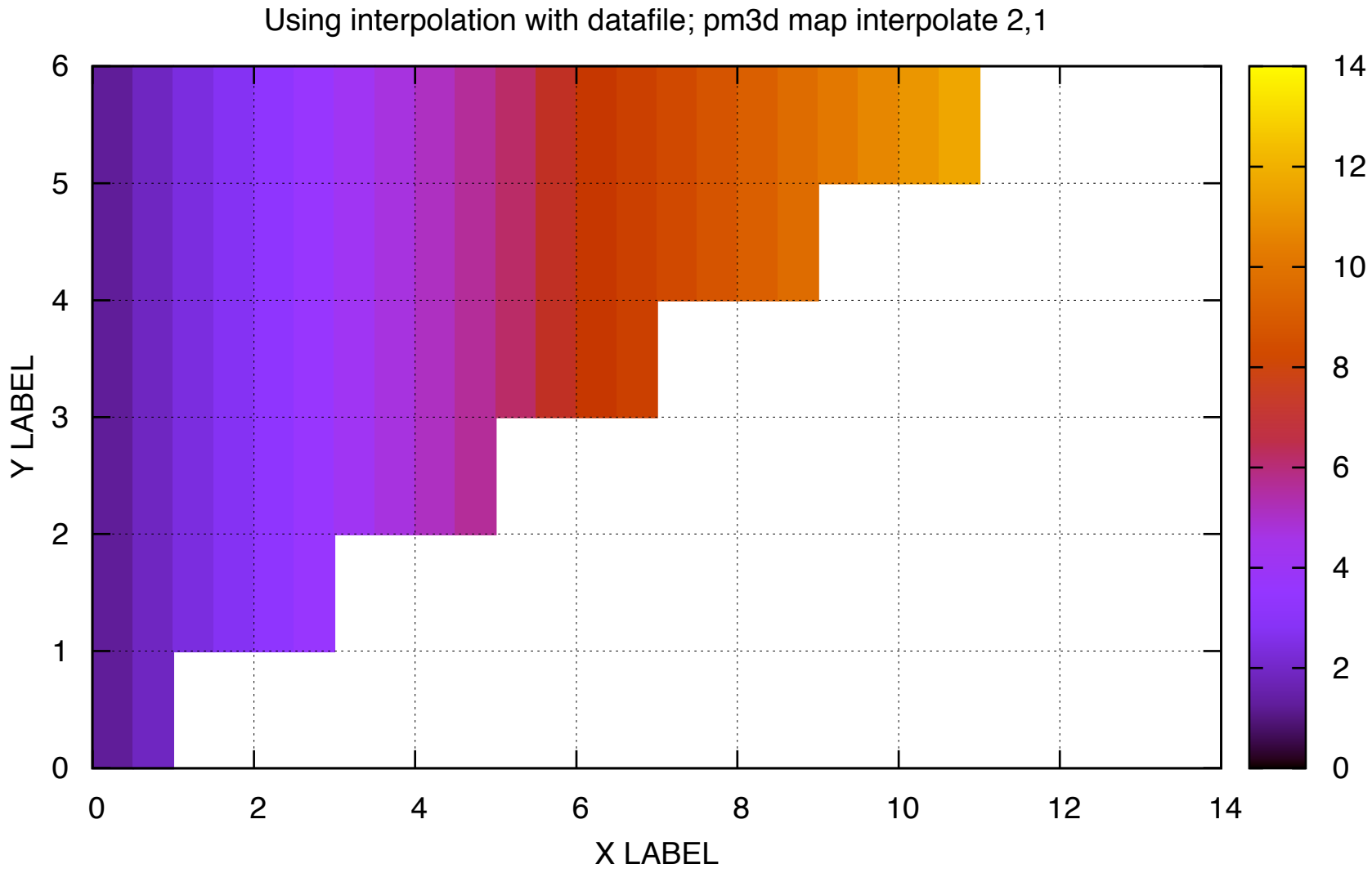

Datafile with different nb of points in scans; pm3d flush center
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

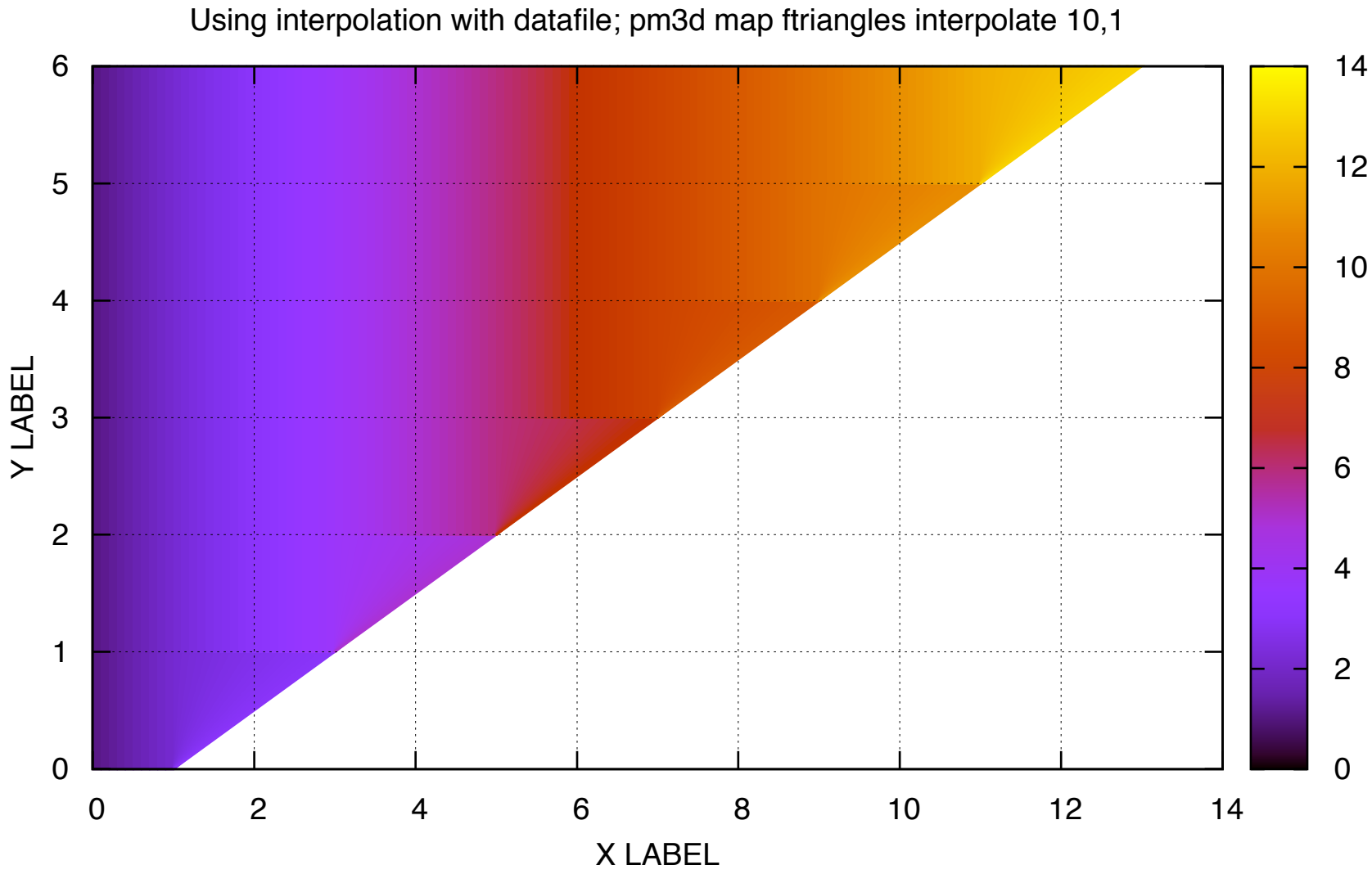

Datafile with different nb of points in scans; pm3d flush end
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

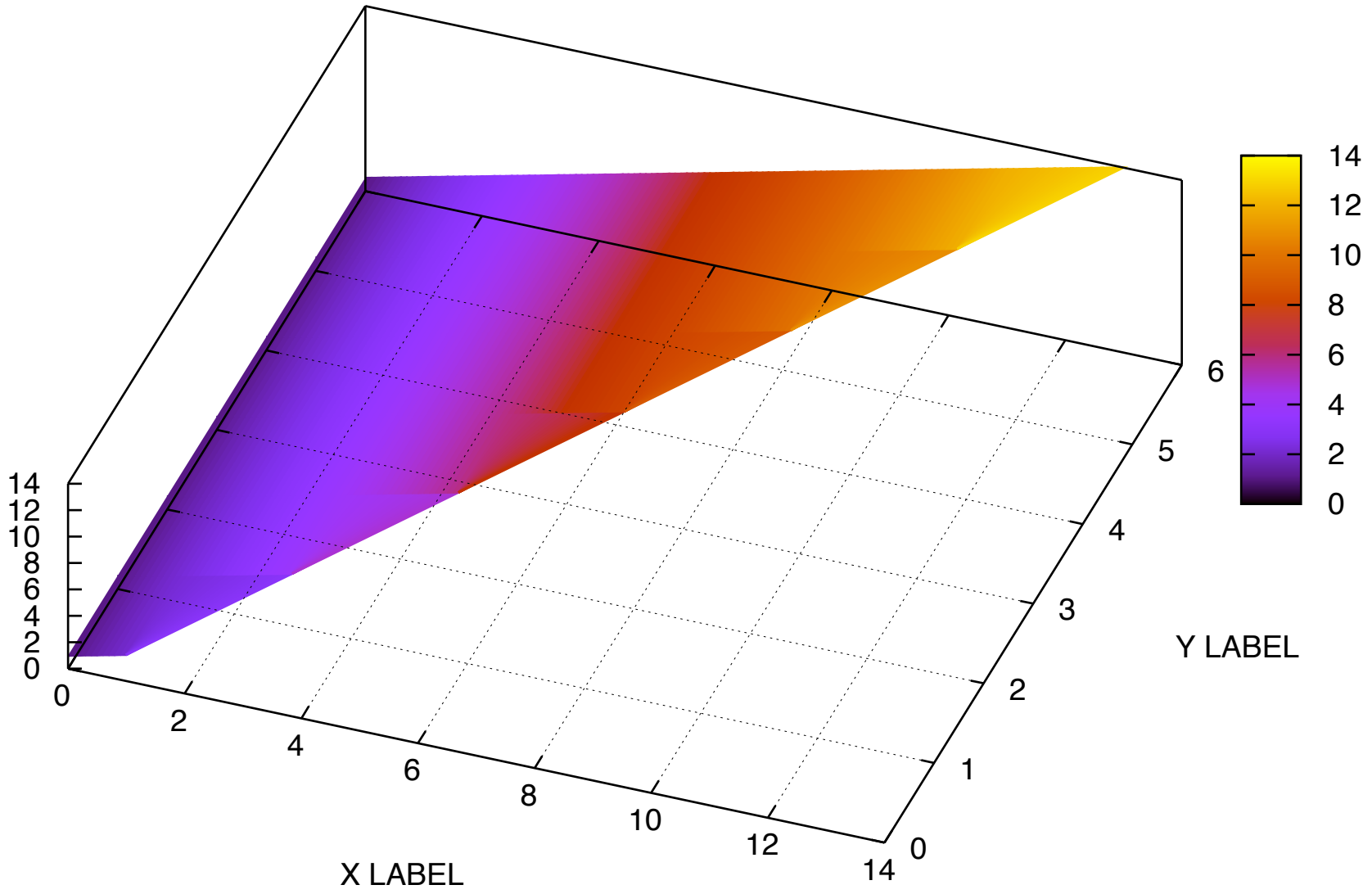

Data with different nb of points in scans; pm3d ftriangles flush begin
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

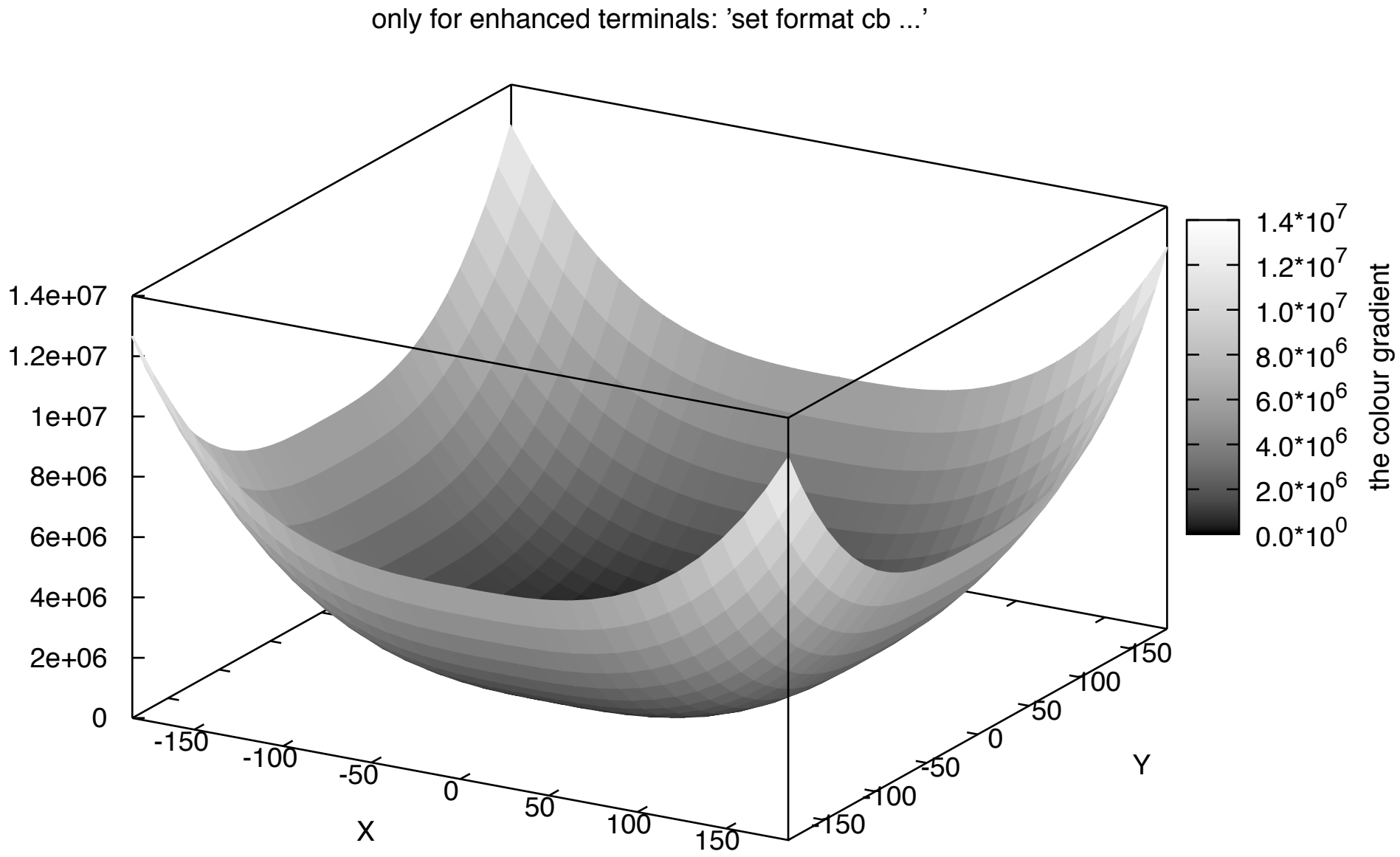

Data with different nb of points in scans; pm3d ftriangles flush center
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

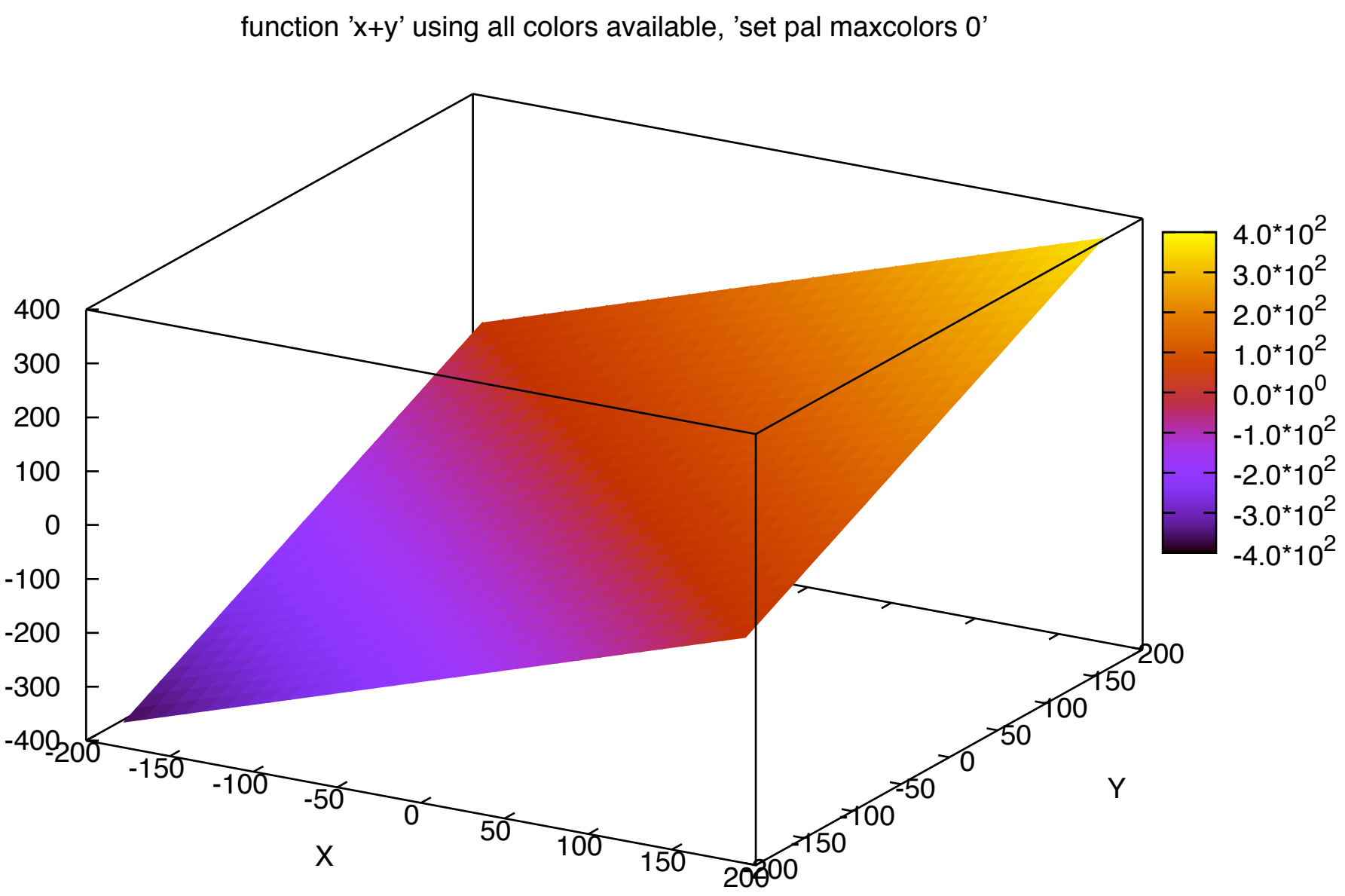

Data with different nb of points in scans; pm3d ftriangles flush end
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

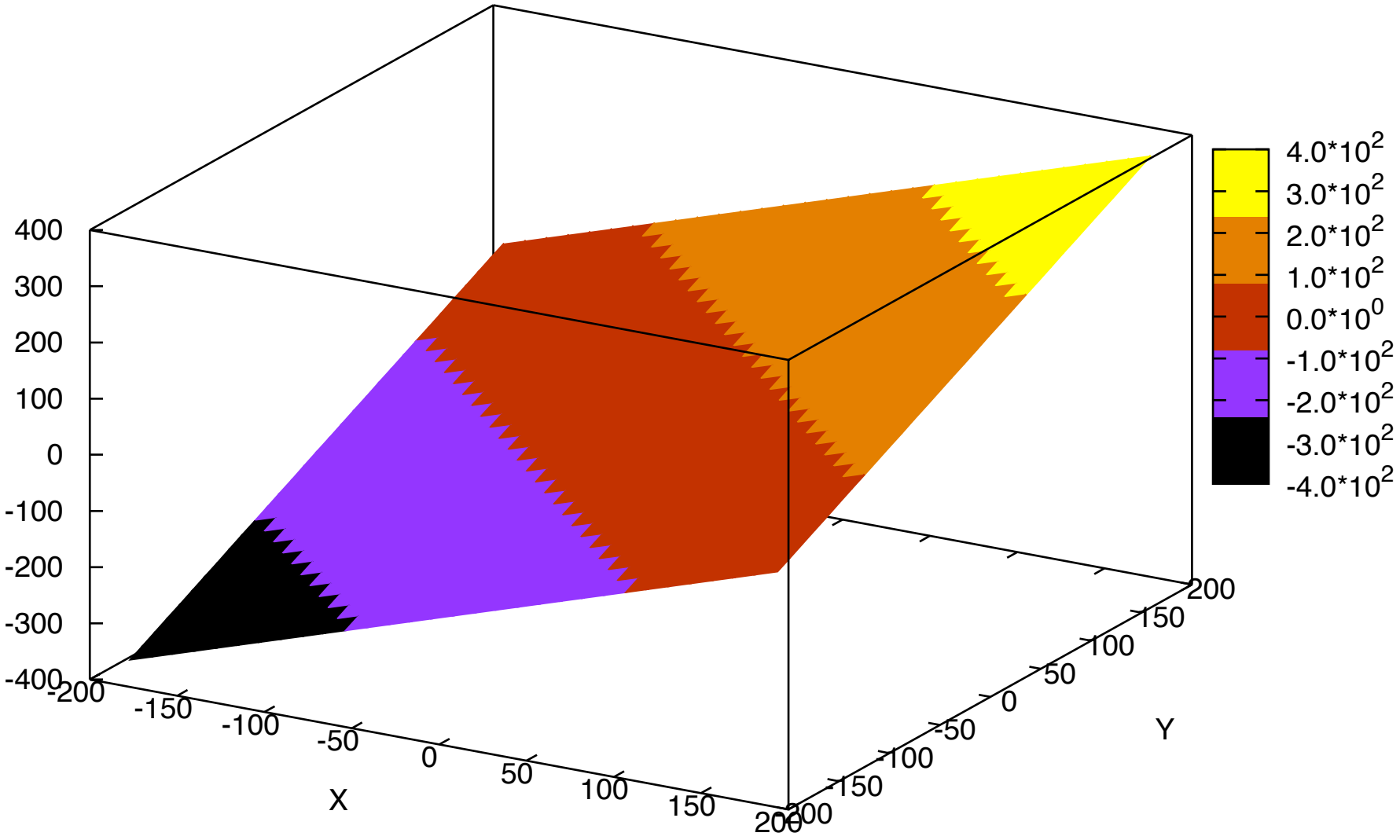

Using interpolation with datafile; pm3d map interpolate 2,1
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

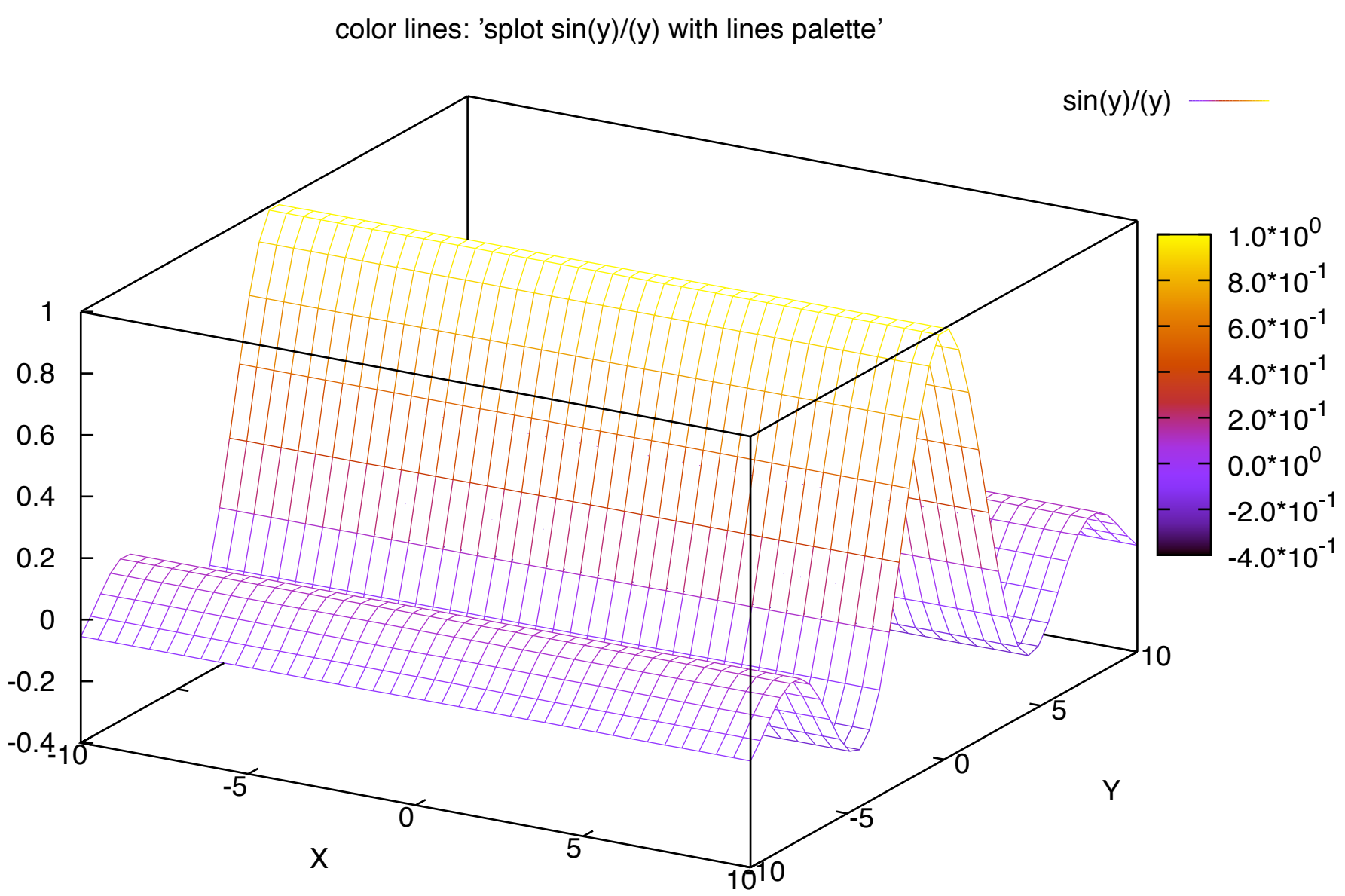

Using interpolation with datafile; pm3d map ftriangles interpolate 10,1
Y LABEL
X LABEL
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14

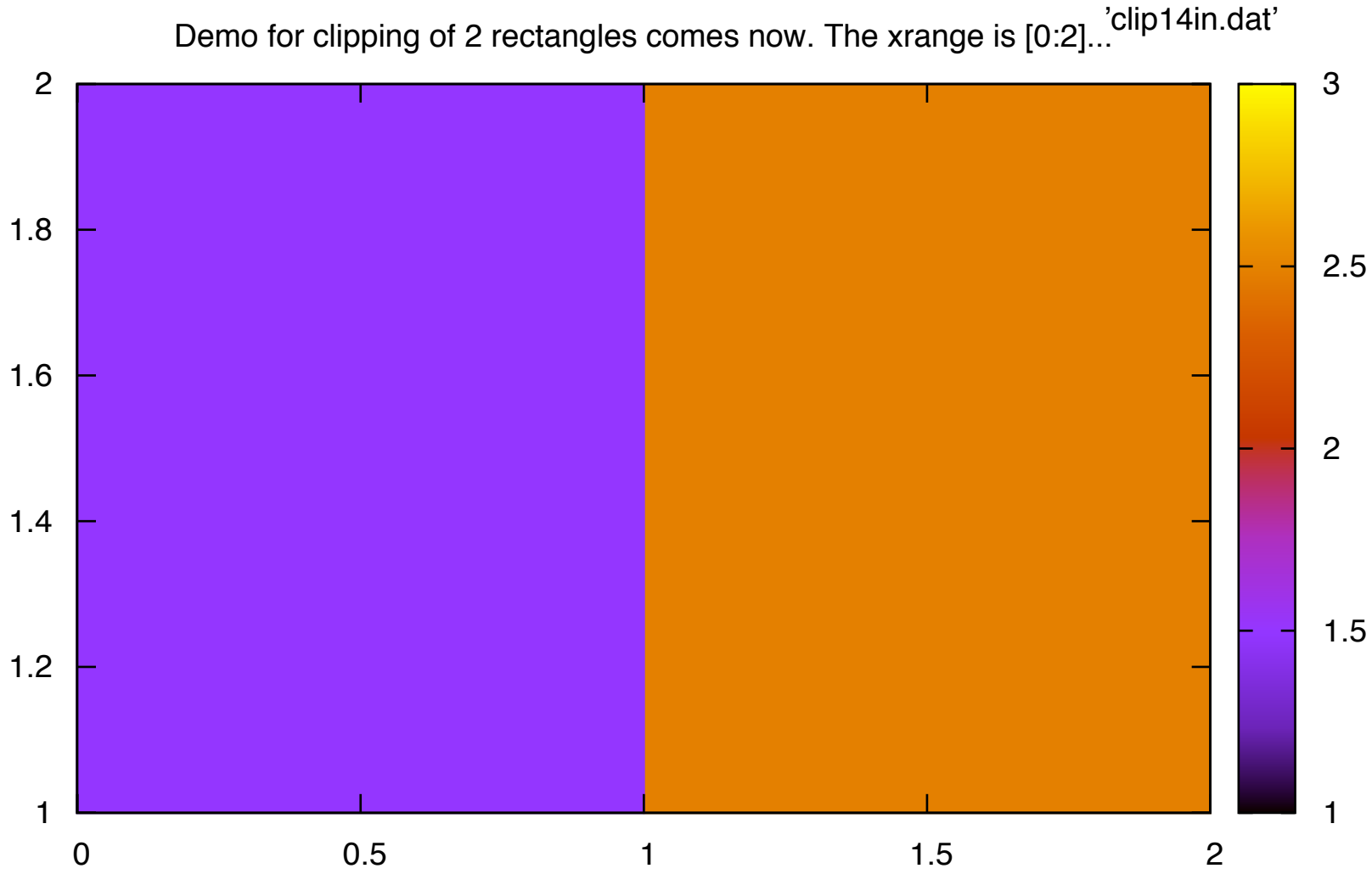

Using interpolation with datafile; pm3d at s ftriangles interpolate 10,1
14
12
10
8
6
4
2
0
0
2
4
6
8
10
12
14
X LABEL
0
1
2
3
4
5
6
Y LABEL
14
12
10
8
6
4
2
0

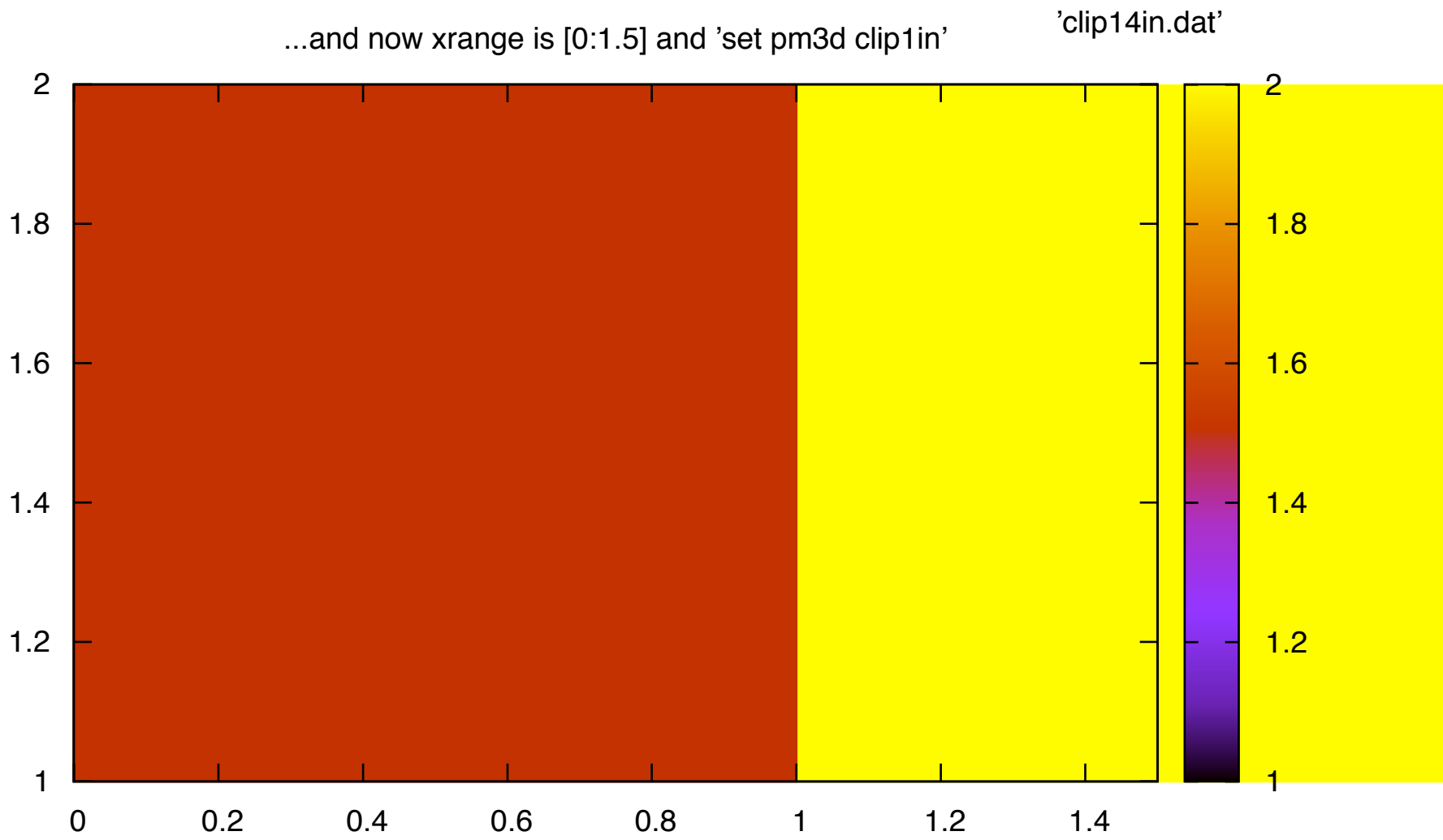

only for enhanced terminals: 'set format cb ...'
1.4e+07
1.2e+07
1e+07
8e+06
6e+06
4e+06
2e+06
0
-150
-100
-50
0
50
100
150
X
Y
1.4*10^7
1.2*10^7
1.0*10^7
8.0*10^6
6.0*10^6
4.0*10^6
2.0*10^6
0.0*10^0
the colour gradient

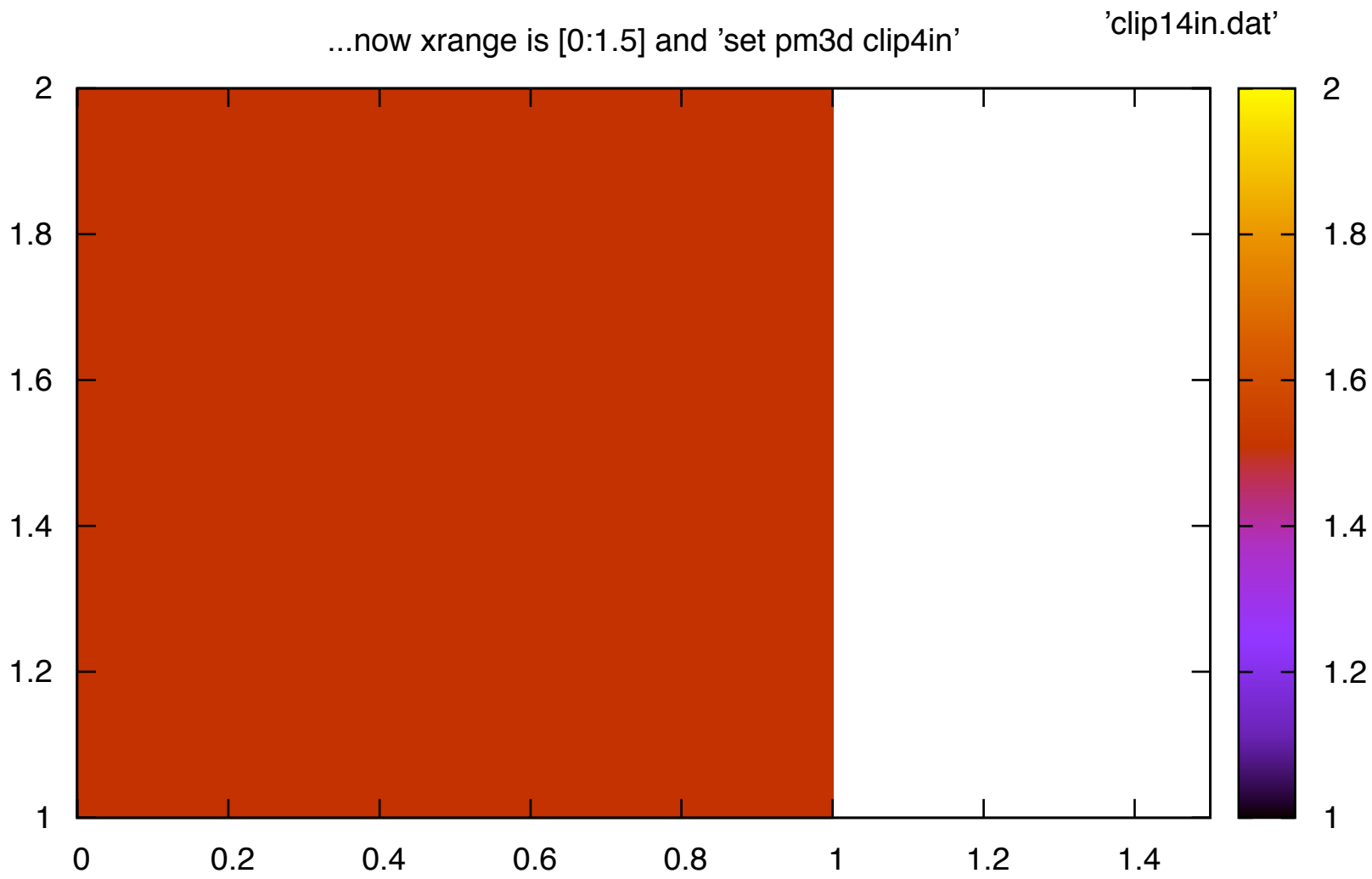

function ’x+y’ using all colors available, ’set pal maxcolors 0’
400
300
200
100
0
-100
-200
-300
-400
-200
-150
-100
-50
0
50
100
150
200
X
200
150
100
50
0
-50
-100
-150
-200
Y
4.0*10^2
3.0*10^2
2.0*10^2
1.0*10^2
0.0*10^0
-1.0*10^2
-2.0*10^2
-3.0*10^2
-4.0*10^2

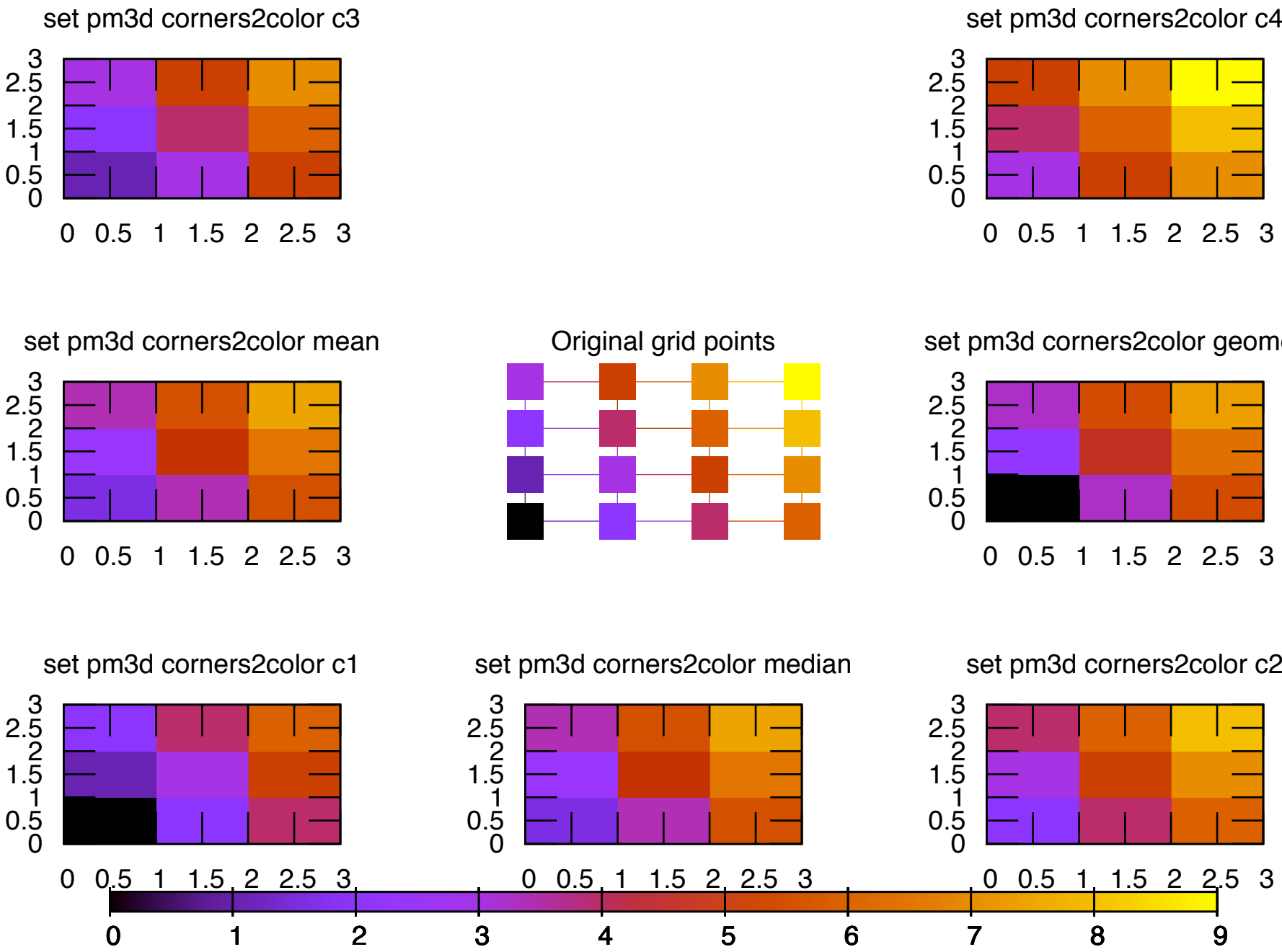

function ’x+y’ using only 5 colors, ’set pal maxcolors 5’
400
300
200
100
0
-100
-200
-300
-400
-200
-150
-100
-50
0
50
100
150
200
X
200
150
100
50
0
-50
-100
-150
-200
Y
4.0*10^2
3.0*10^2
2.0*10^2
1.0*10^2
0.0*10^0
-1.0*10^2
-2.0*10^2
-3.0*10^2
-4.0*10^2

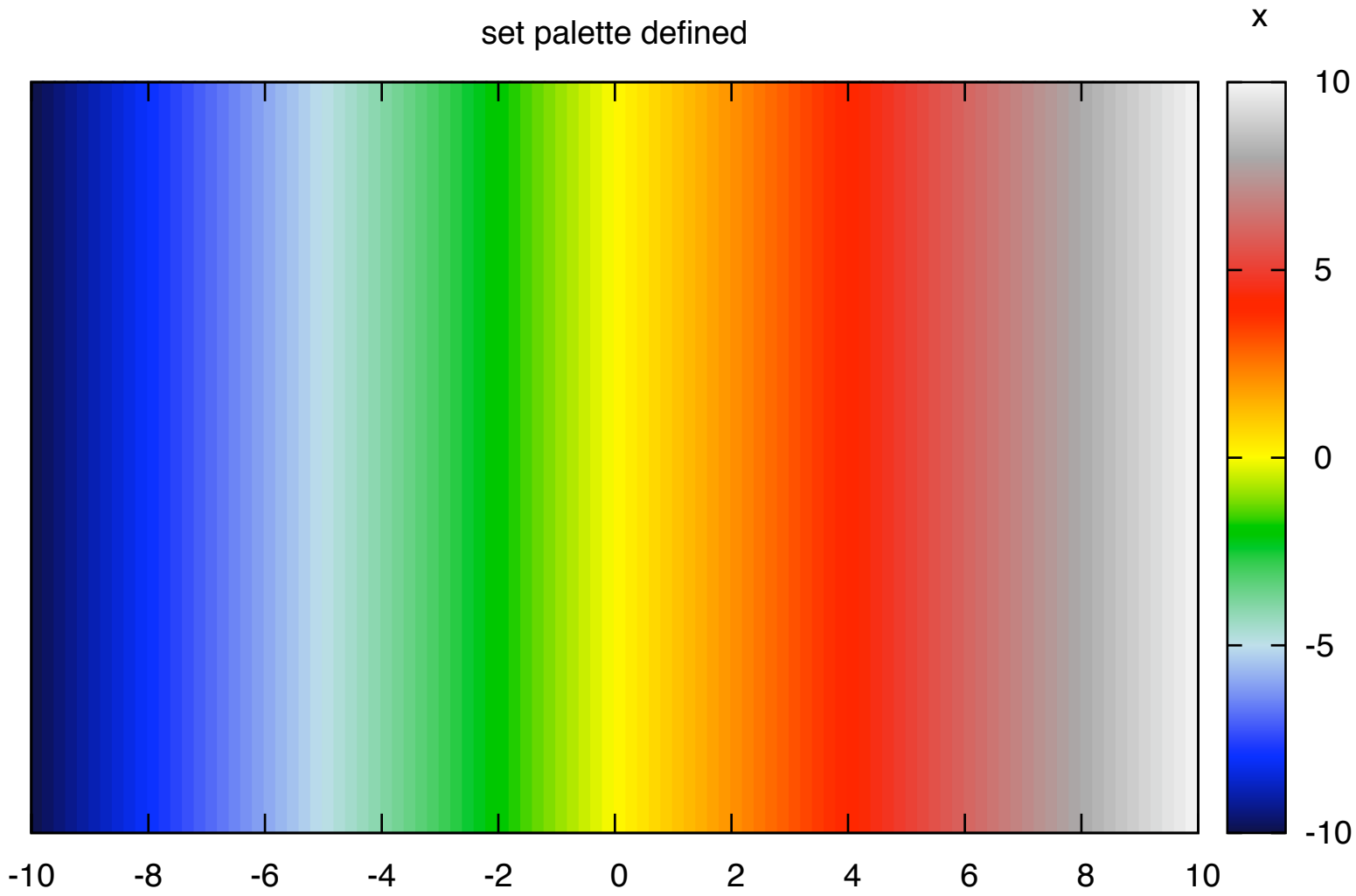

color lines: 'splot sin(y)/(y) with lines palette'
sin(y)/(y)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10
X
10
5
0
-5
-10
Y
1.0*10^0
8.0*10^-1
6.0*10^-1
4.0*10^-1
2.0*10^-1
0.0*10^0
-2.0*10^-1
-4.0*10^-1

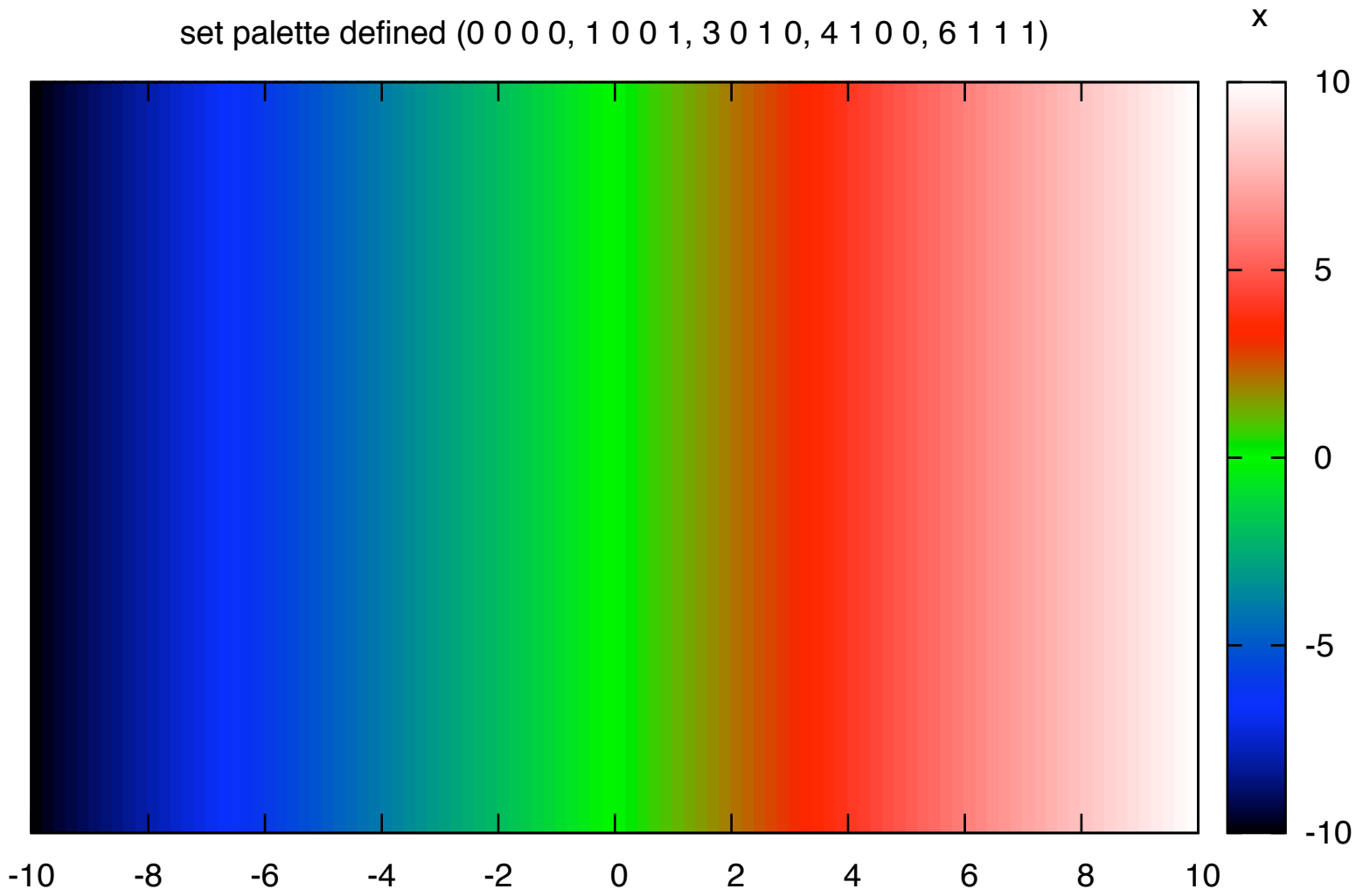

Demo for clipping of 2 rectangles comes now. The xrange is [0:2]... 'clip14in.dat'
2
1.8
1.6
1.4
1.2
1
0
0.5
1
1.5
2
3
2.5
2
1.5
1

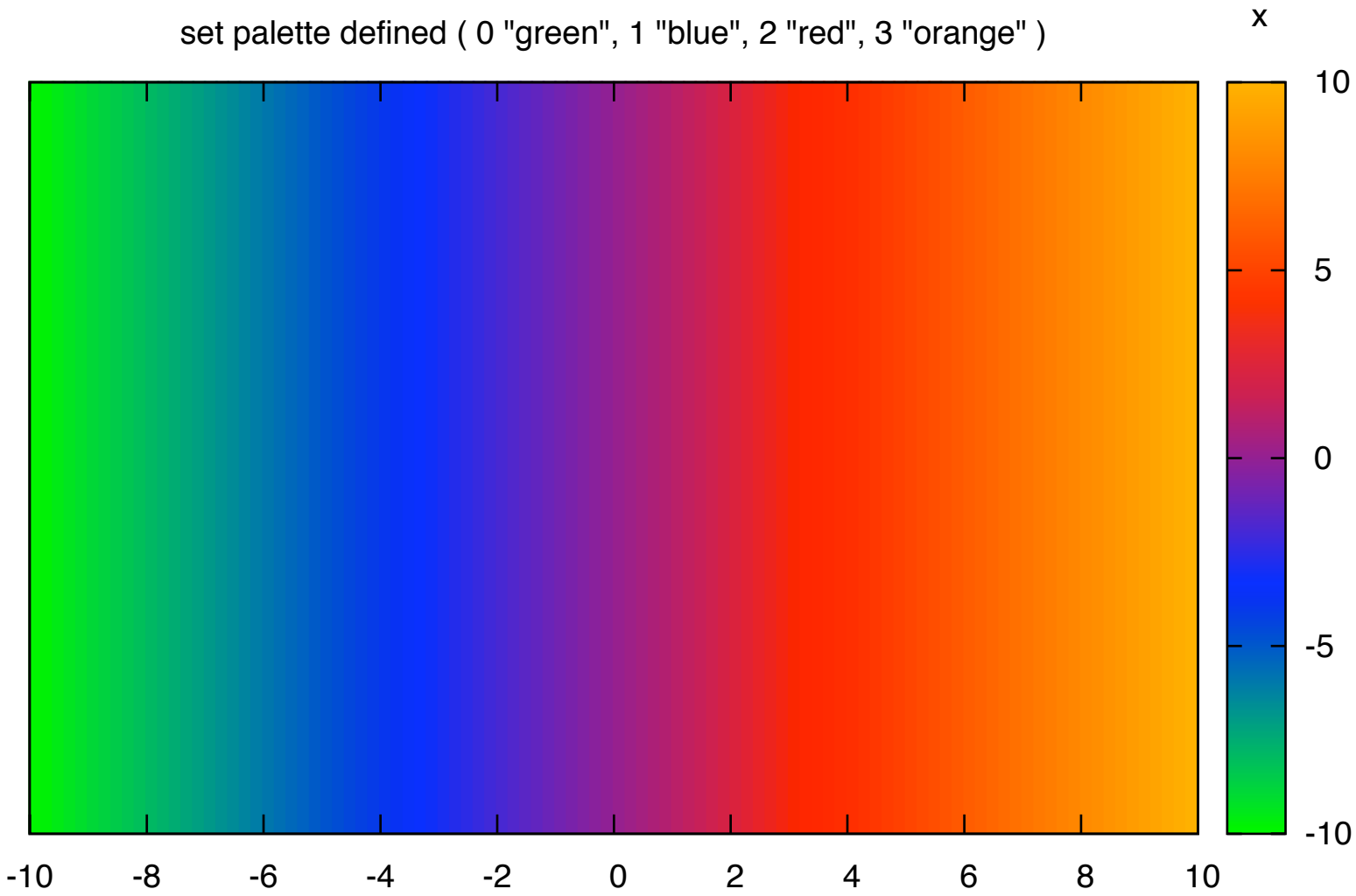

...and now xrange is [0:1.5] and 'set pm3d clip1in'
'clip14in.dat'
2
1.8
1.6
1.4
1.2
1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
2
1.8
1.6
1.4
1.2
1

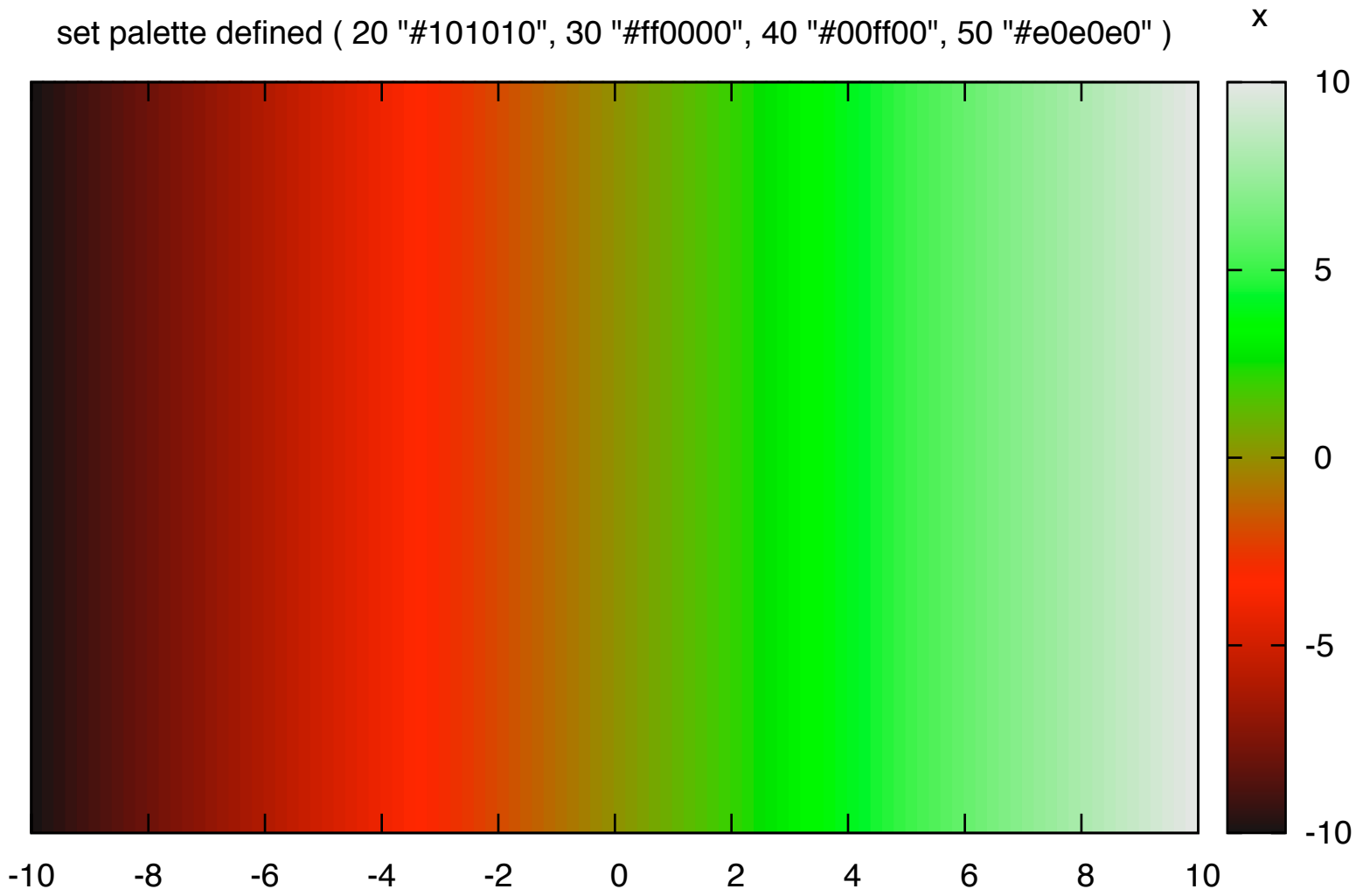

...now xrange is [0:1.5] and 'set pm3d clip4in'
'clip14in.dat'
2
1.8
1.6
1.4
1.2
1
0
0.2
0.4
0.6
0.8
1
1.2
1.4

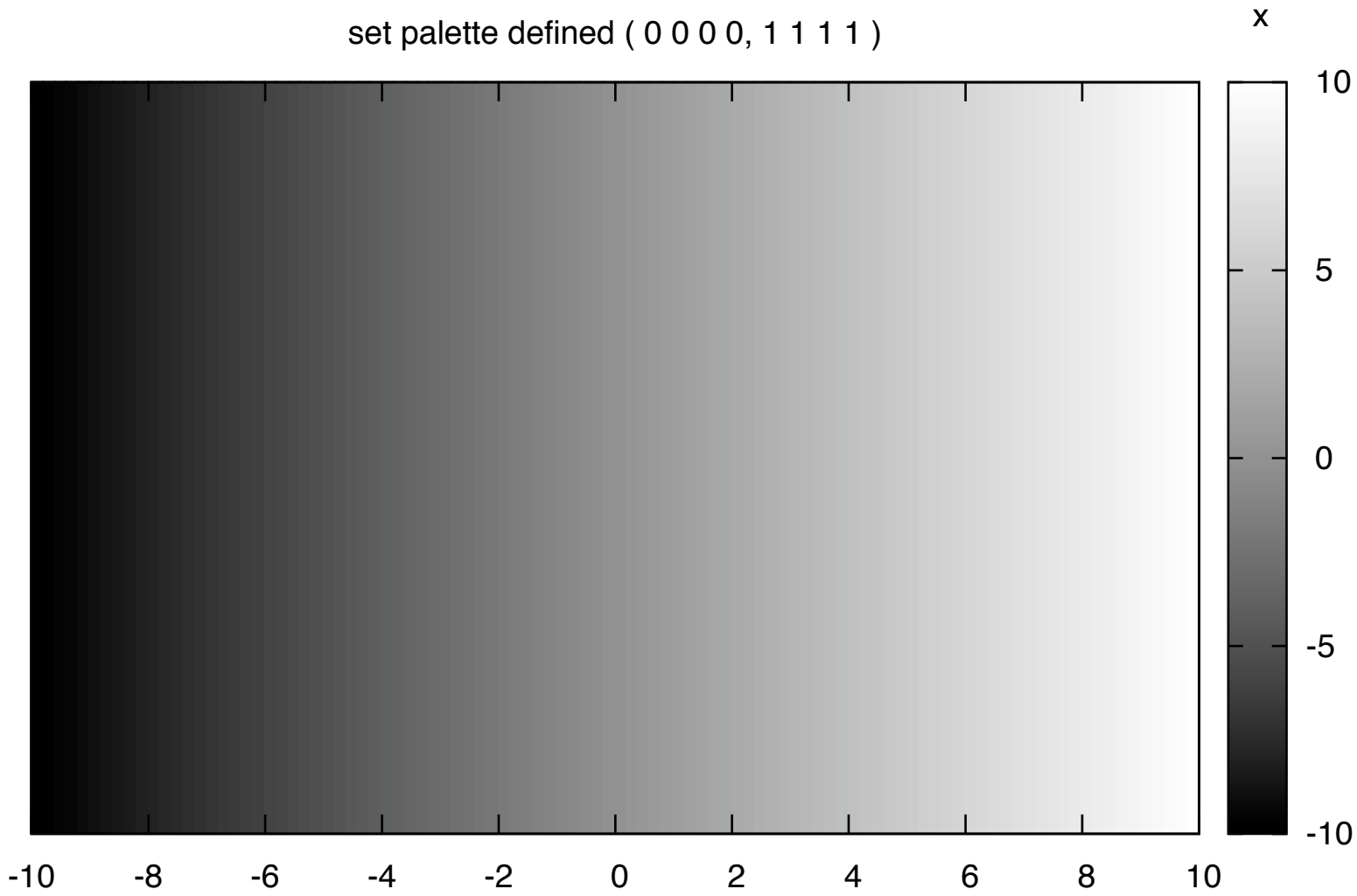

set pm3d corners2color c3
set pm3d corners2color c4
set pm3d corners2color mean
Original grid points
set pm3d corners2color geomean
set pm3d corners2color c1
set pm3d corners2color median
set pm3d corners2color c2
0 0.5 1 1.5 2 2.5 3
0 1 2 3 4 5 6 7 8 9

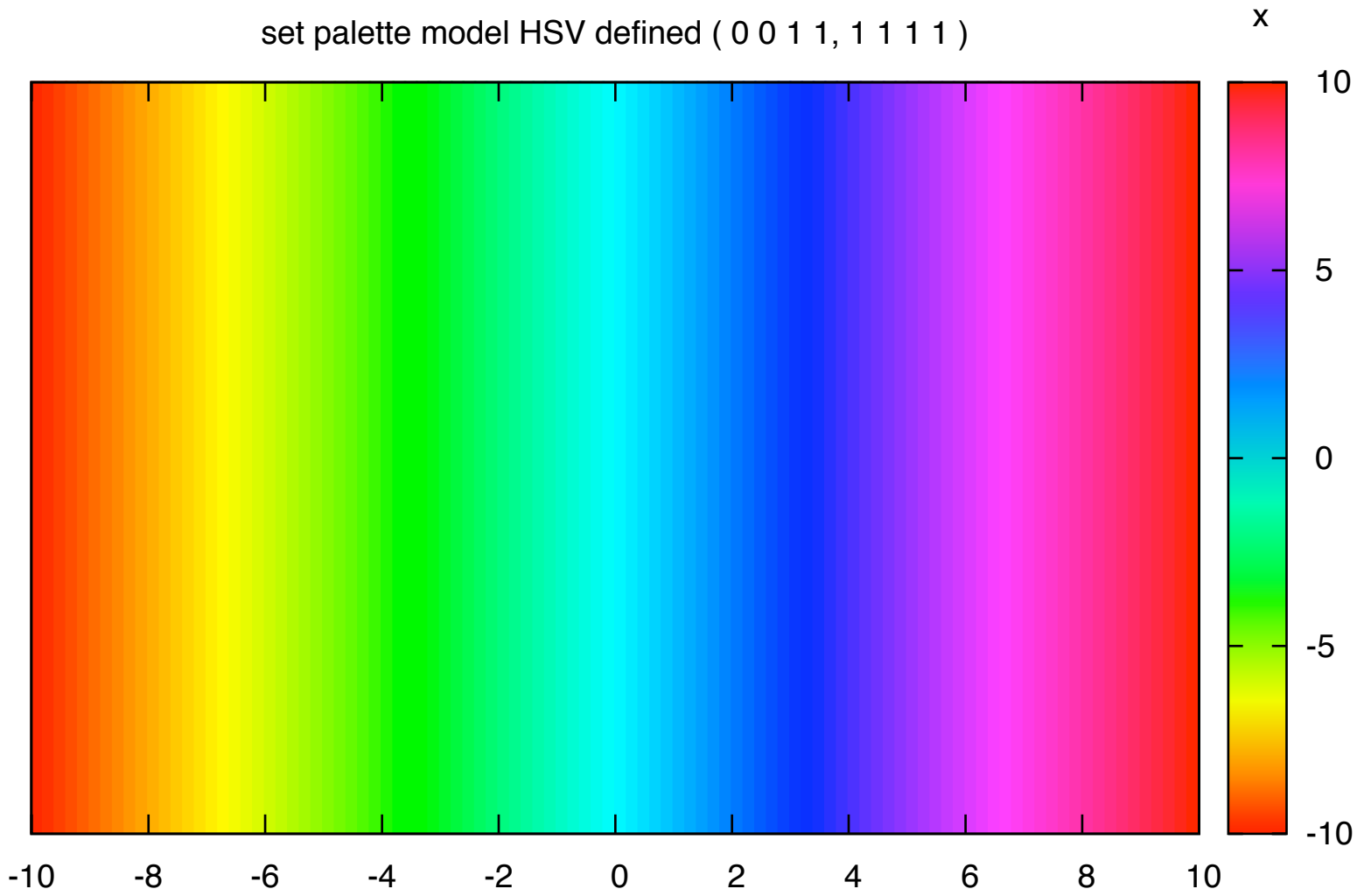

set palette defined
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

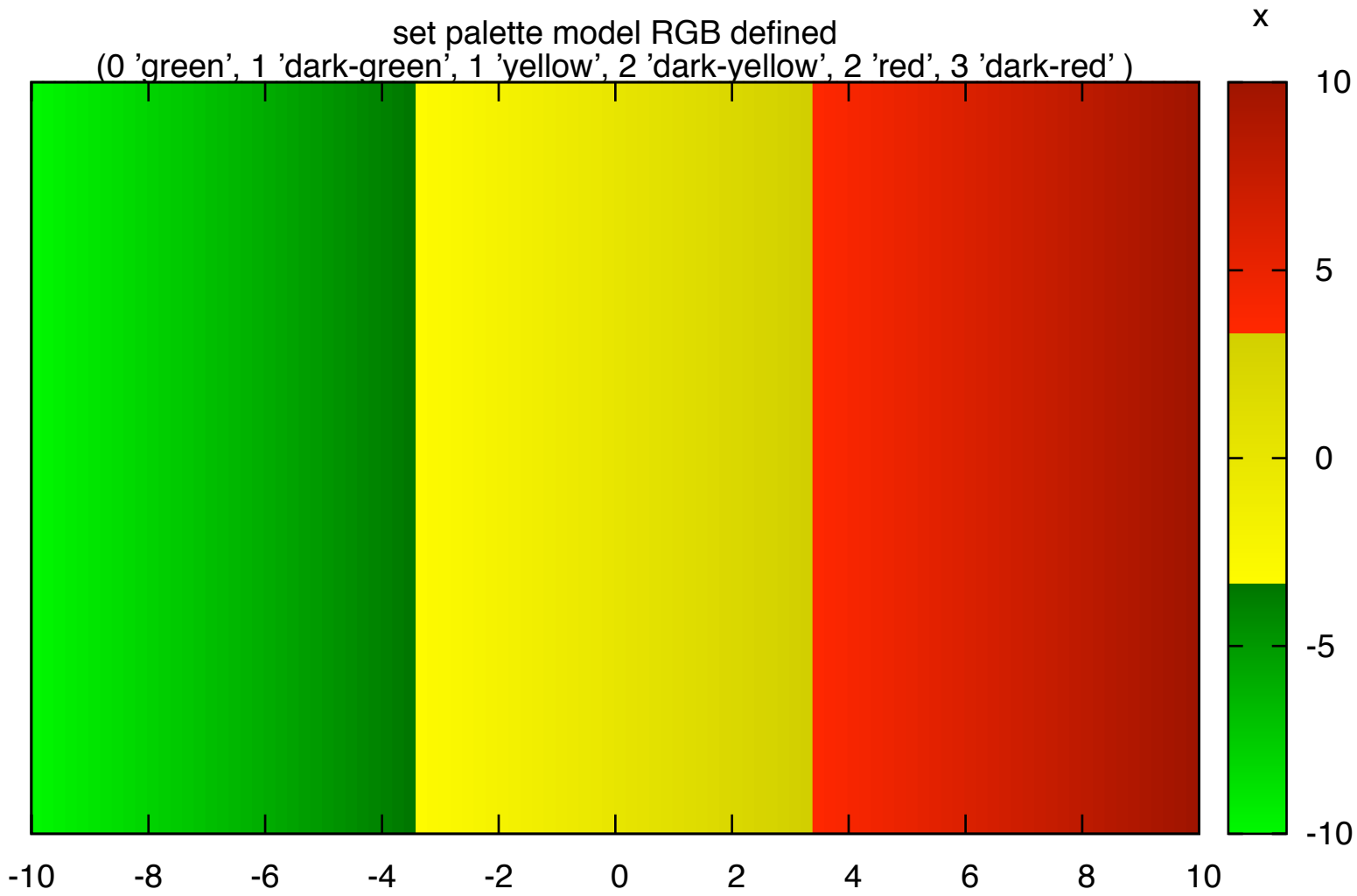

set palette defined (0 0 0 0, 1 0 0 1, 3 0 1 0, 4 1 0 0, 6 1 1 1)
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

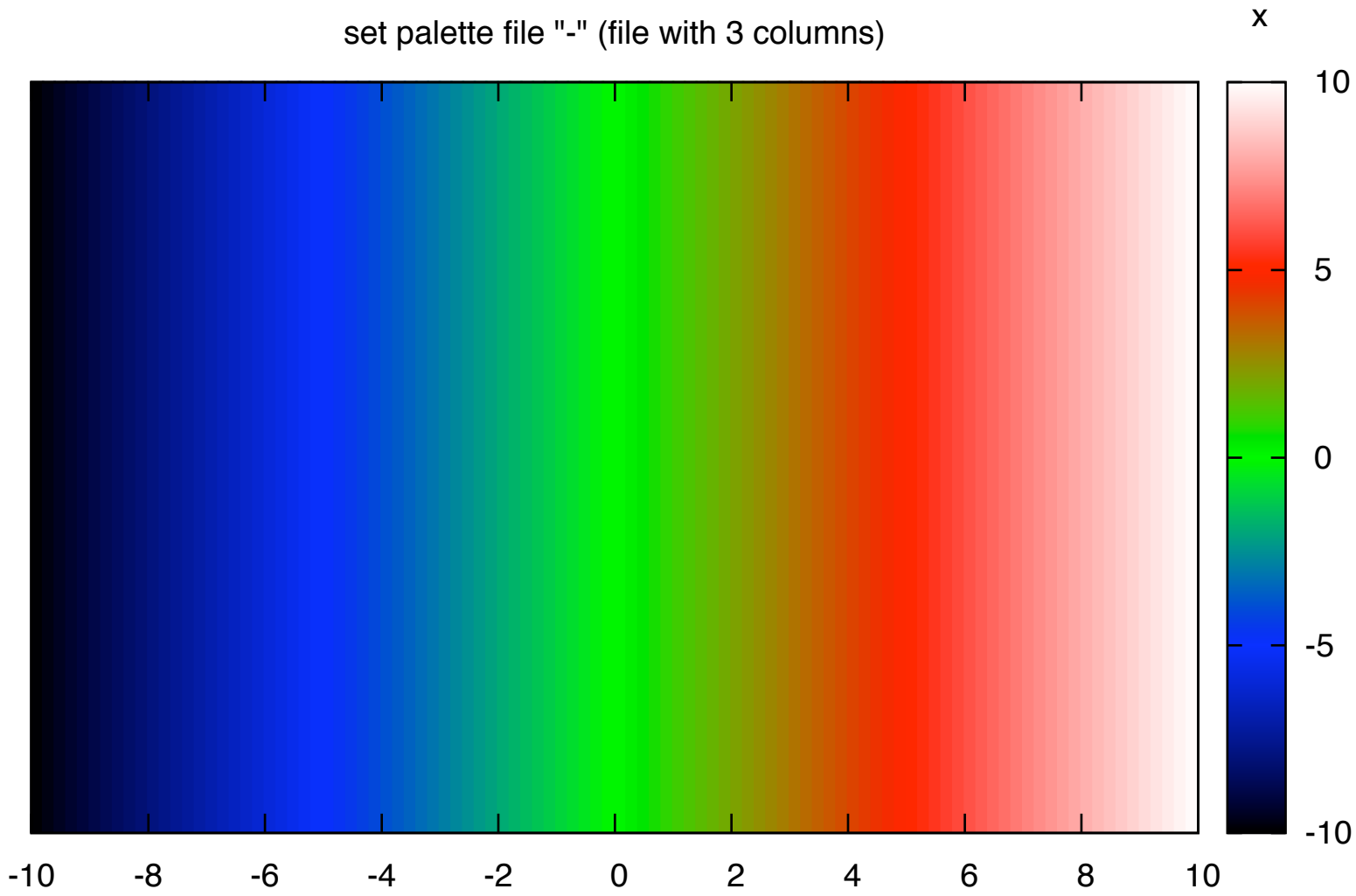

set palette defined (0 "green", 1 "blue", 2 "red", 3 "orange")
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

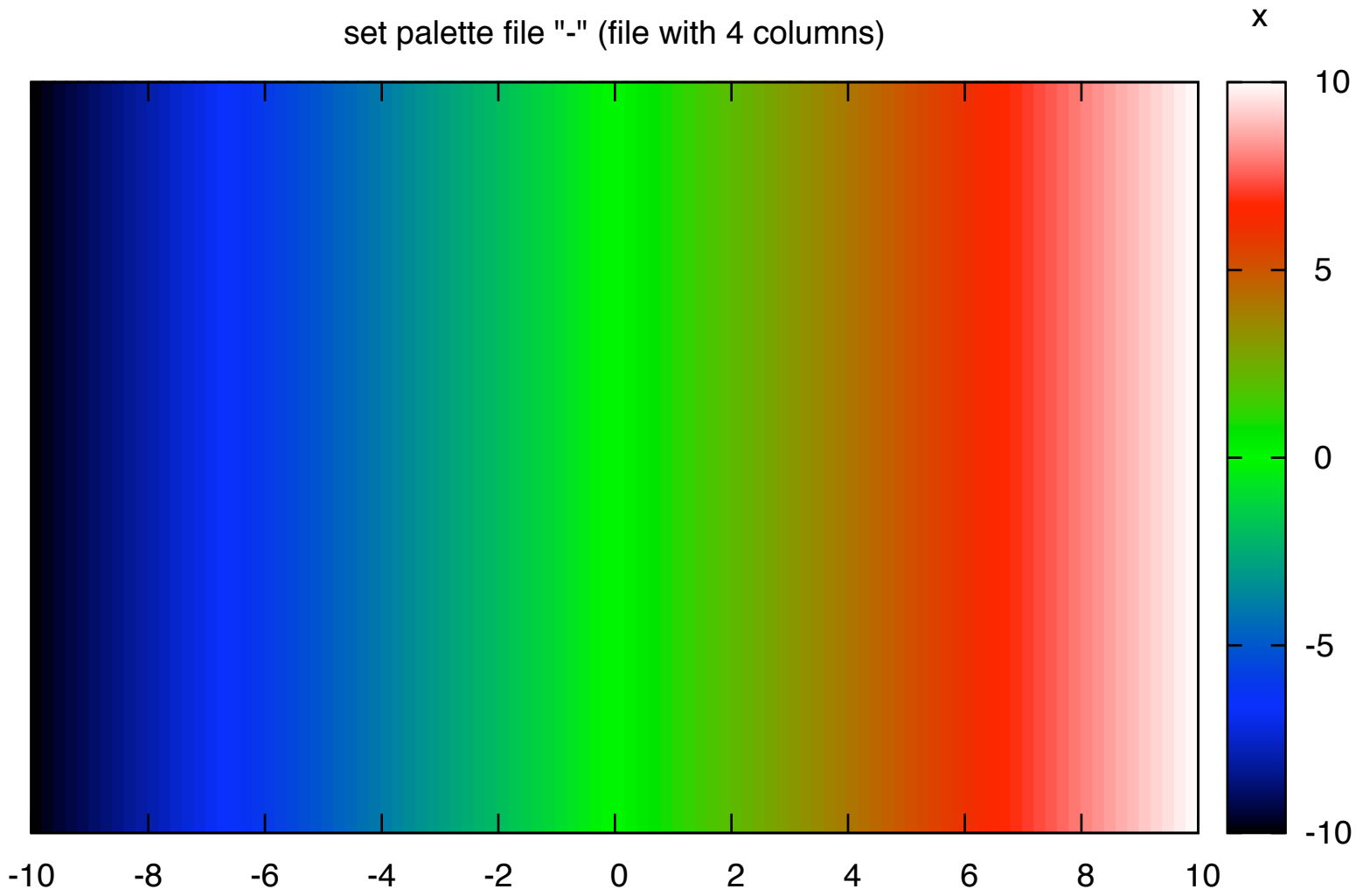

set palette defined (20 "#101010", 30 "#ff0000", 40 "#00ff00", 50 "#e0e0e0")
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

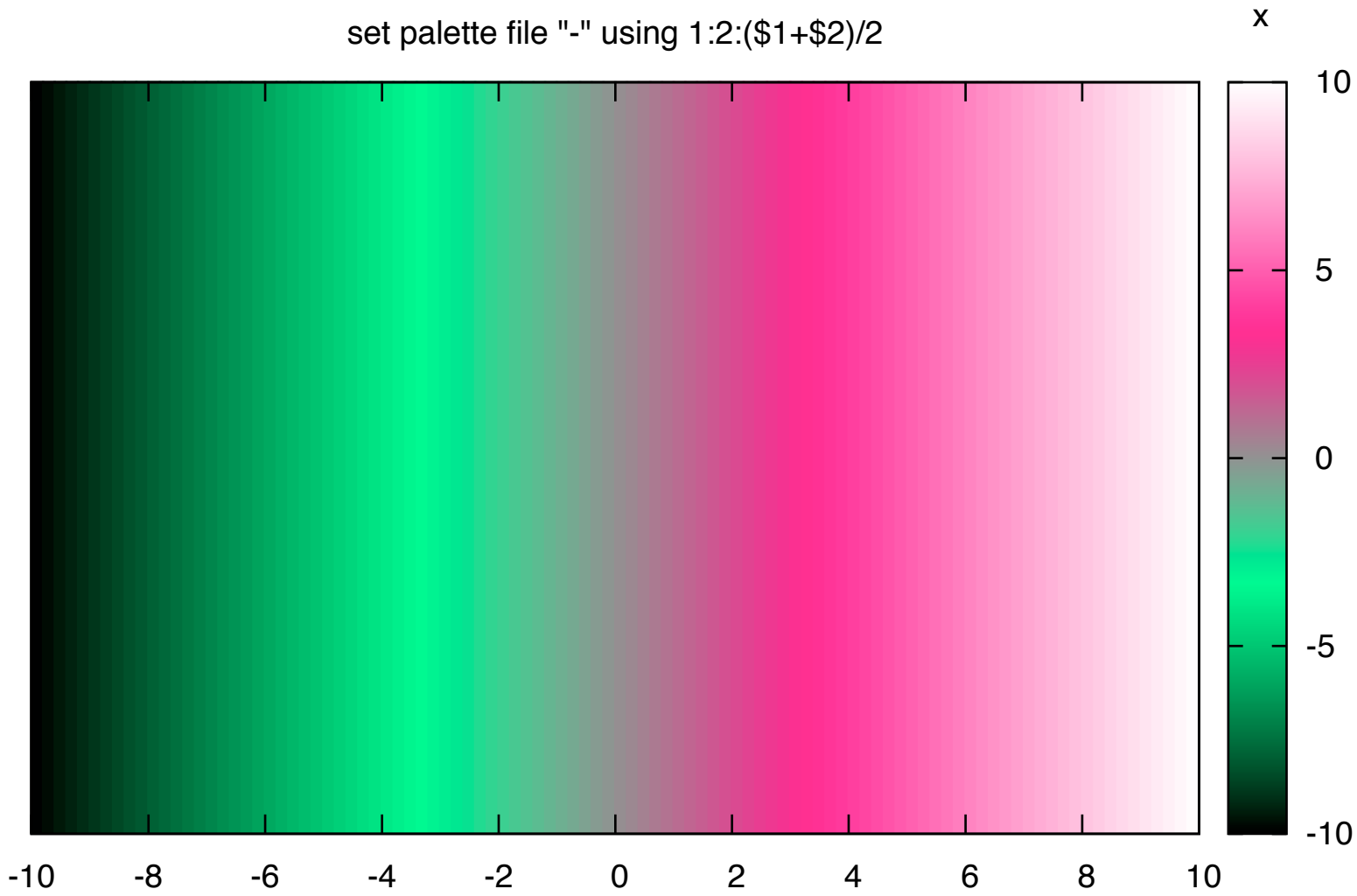

set palette defined (0 0 0 0, 1 1 1 1)
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

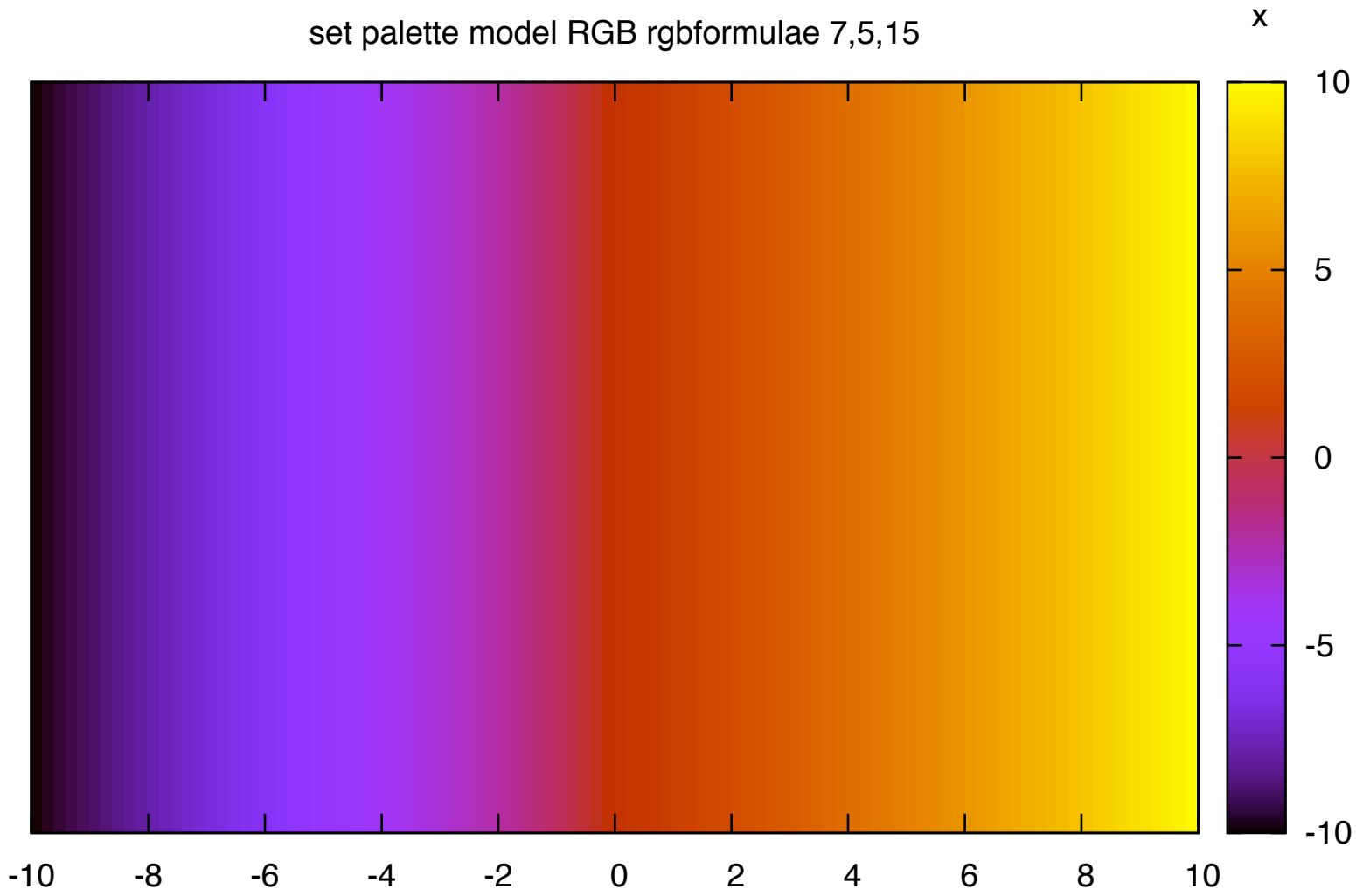

set palette model HSV defined (0 0 1 1, 1 1 1 1)
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

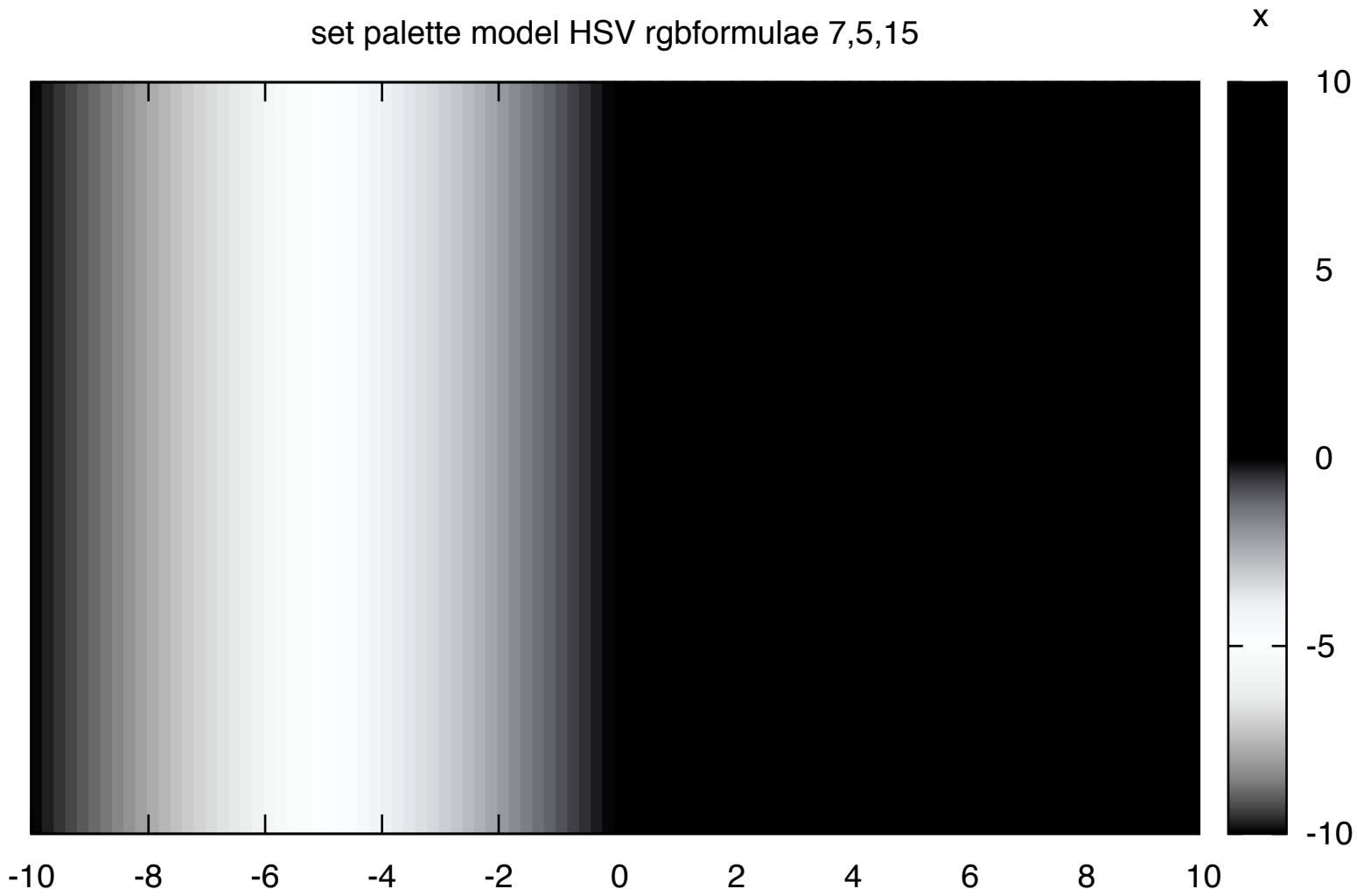

set palette model RGB defined
(0 'green', 1 'dark-green', 1 'yellow', 2 'dark-yellow', 2 'red', 3 'dark-red')
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

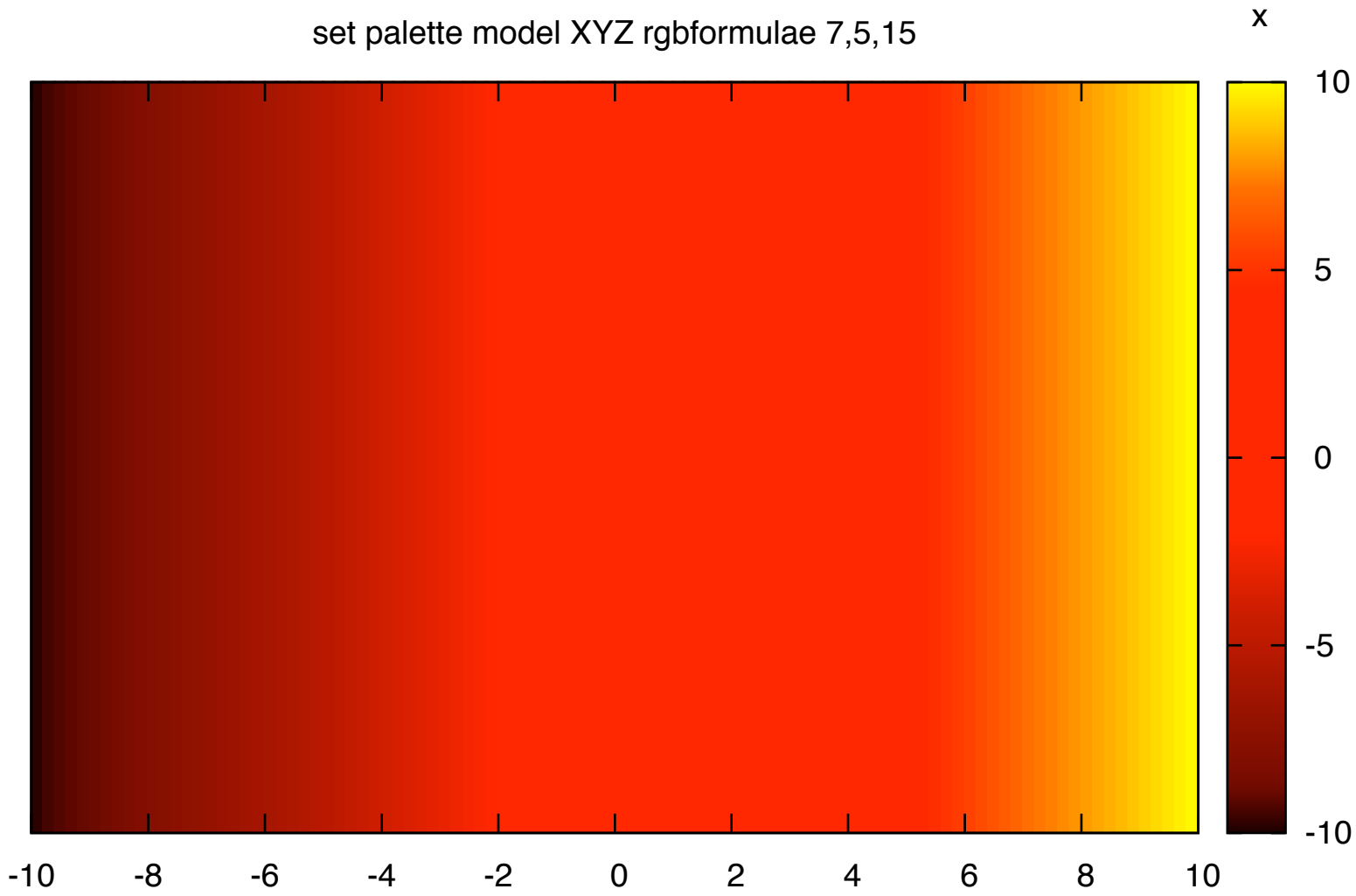

set palette file "-" (file with 3 columns)
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

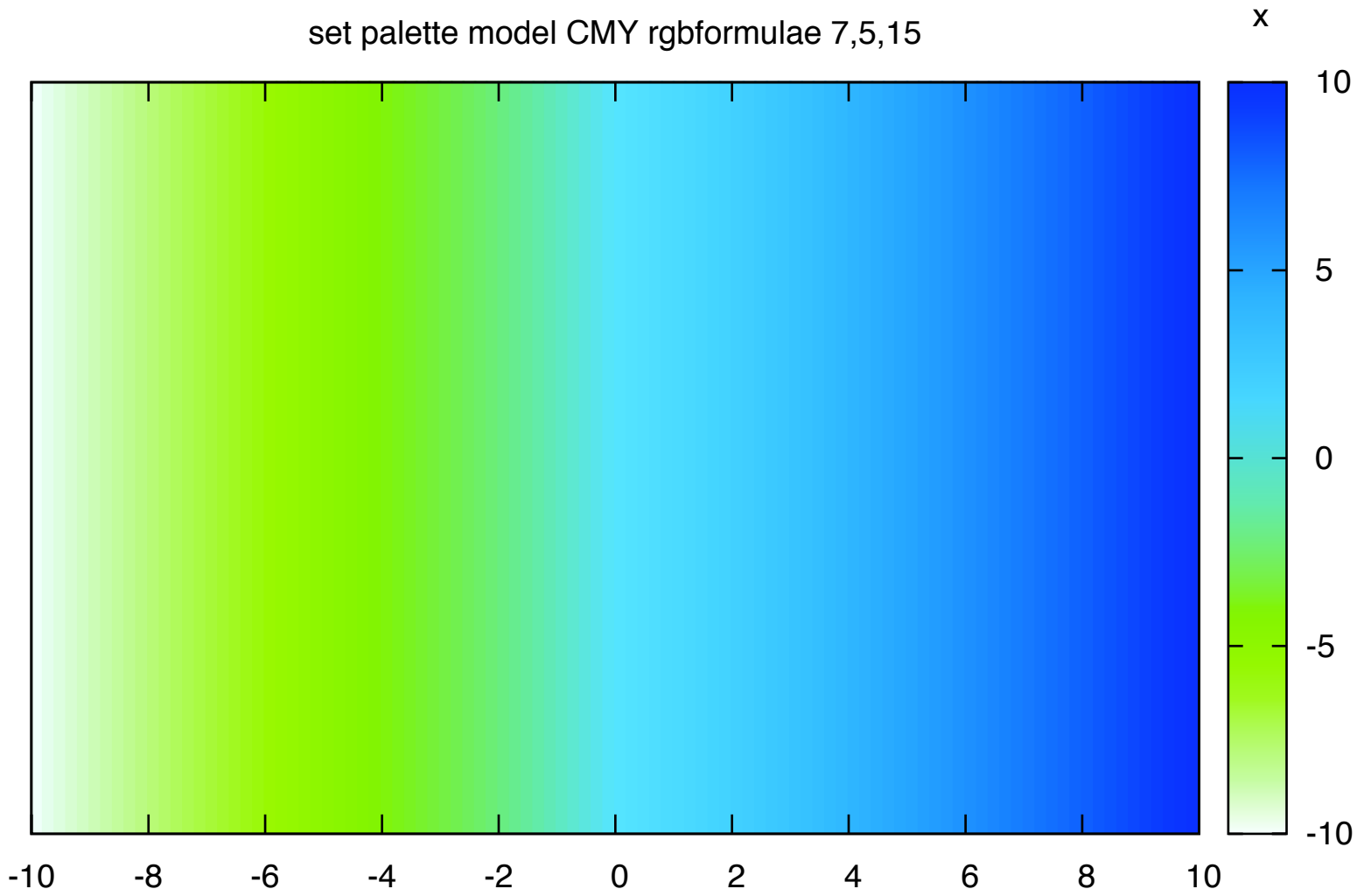

set palette file "-" (file with 4 columns)
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

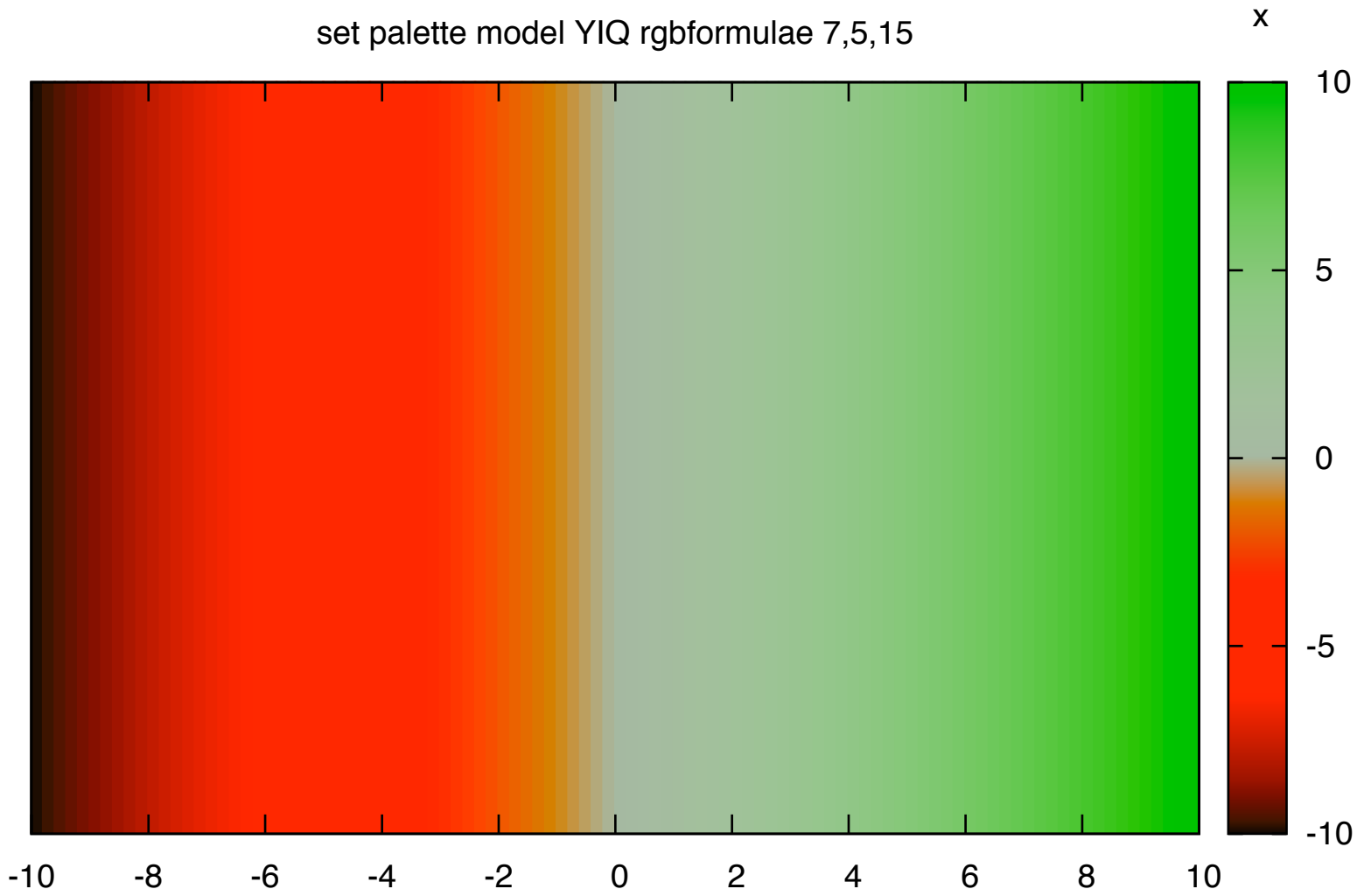

set palette file "-" using 1:2:($1+$2)/2
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

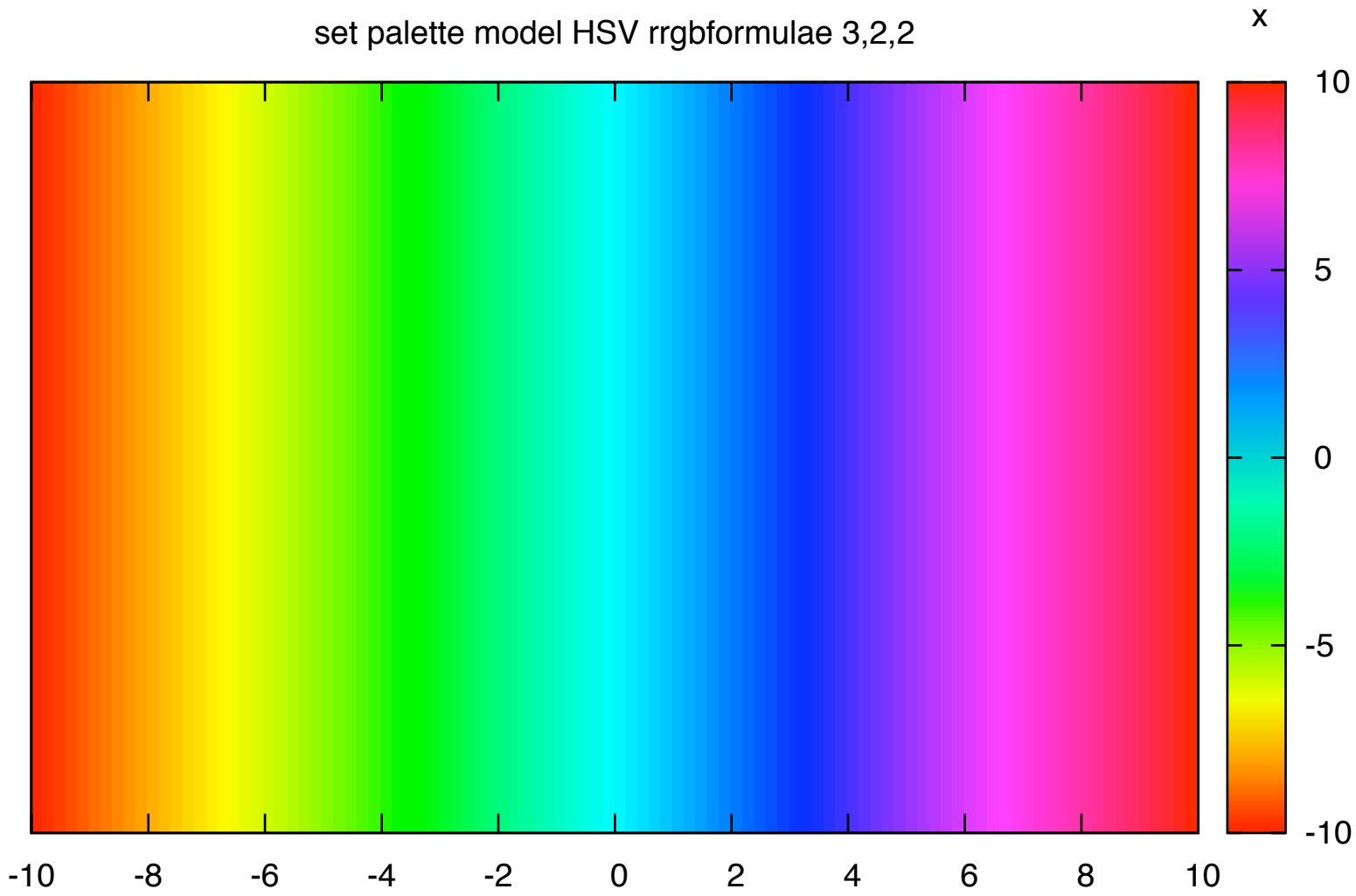

set palette model RGB rgbformulae 7,5,15
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

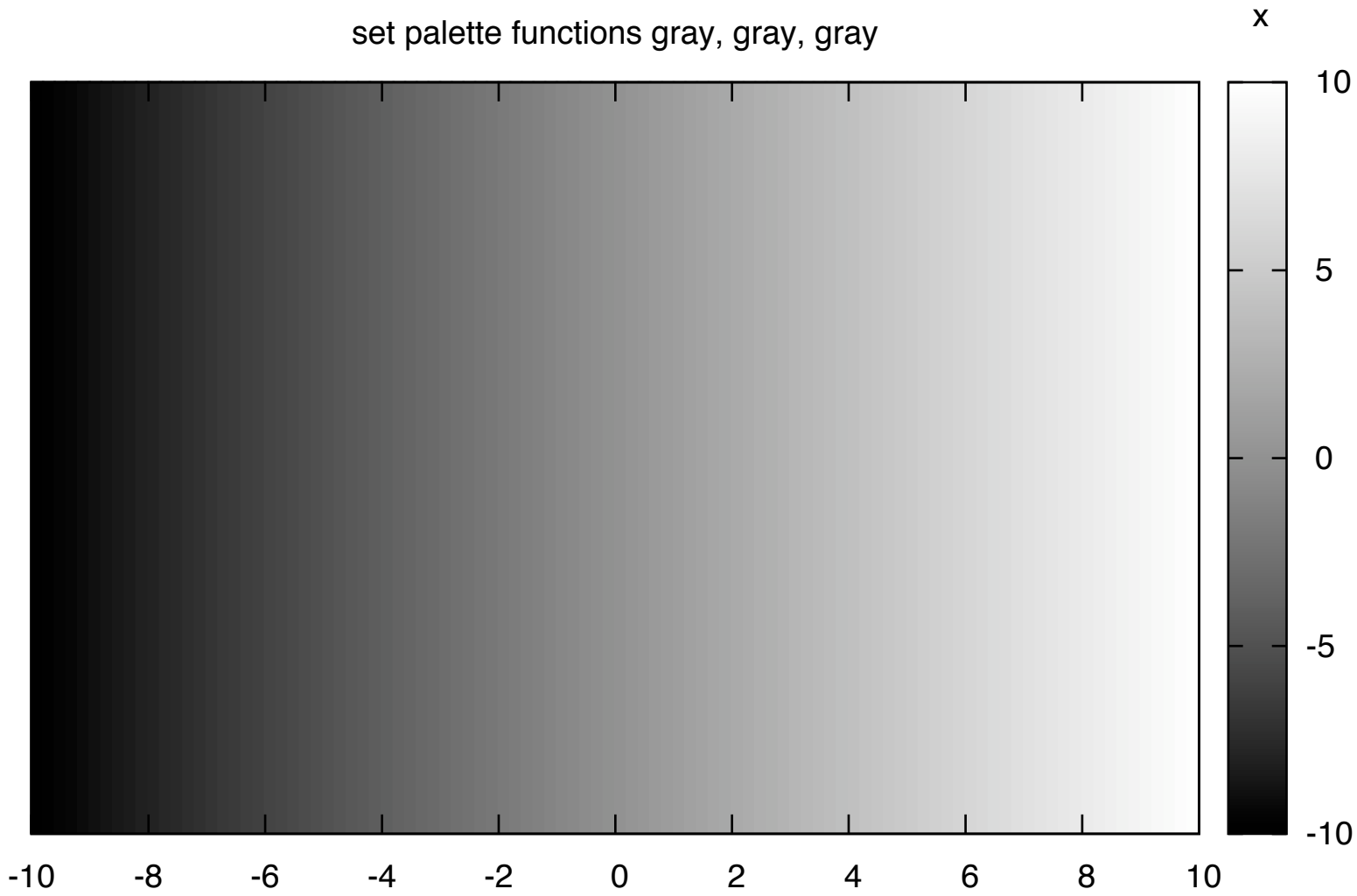

set palette model HSV rgbformulae 7,5,15
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

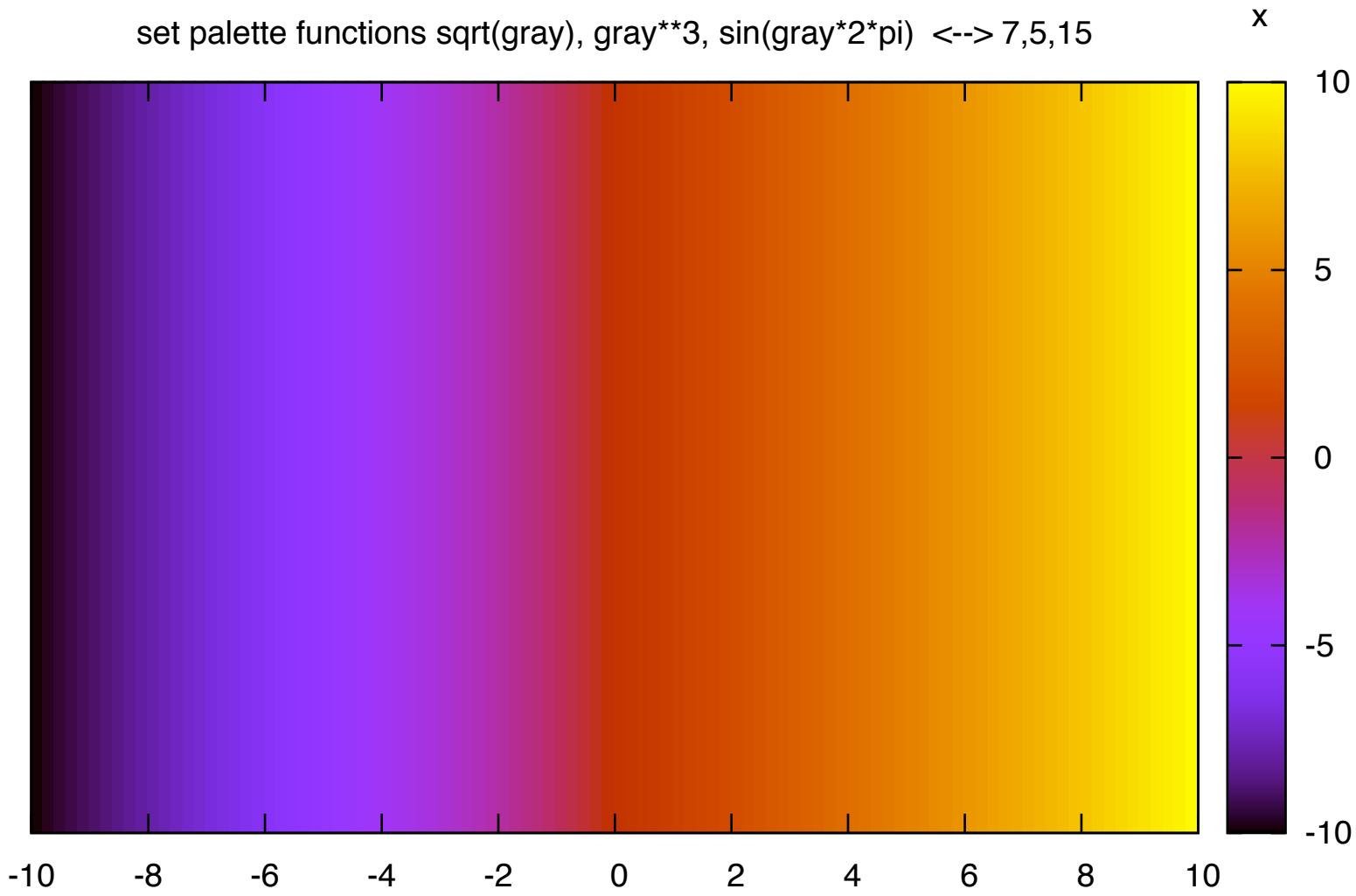

set palette model XYZ rgbformulae 7,5,15
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

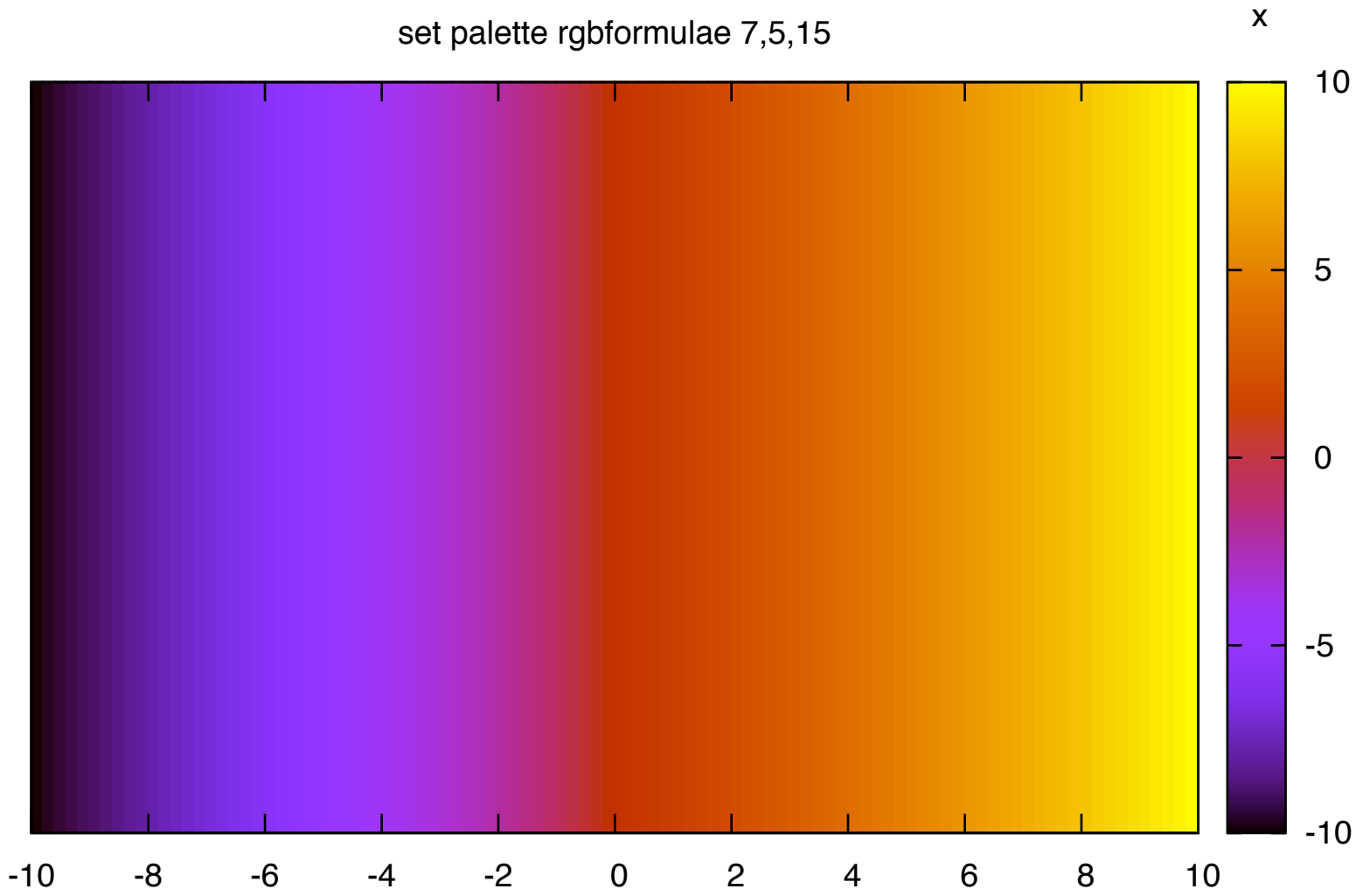

set palette model CMY rgbformulae 7,5,15
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

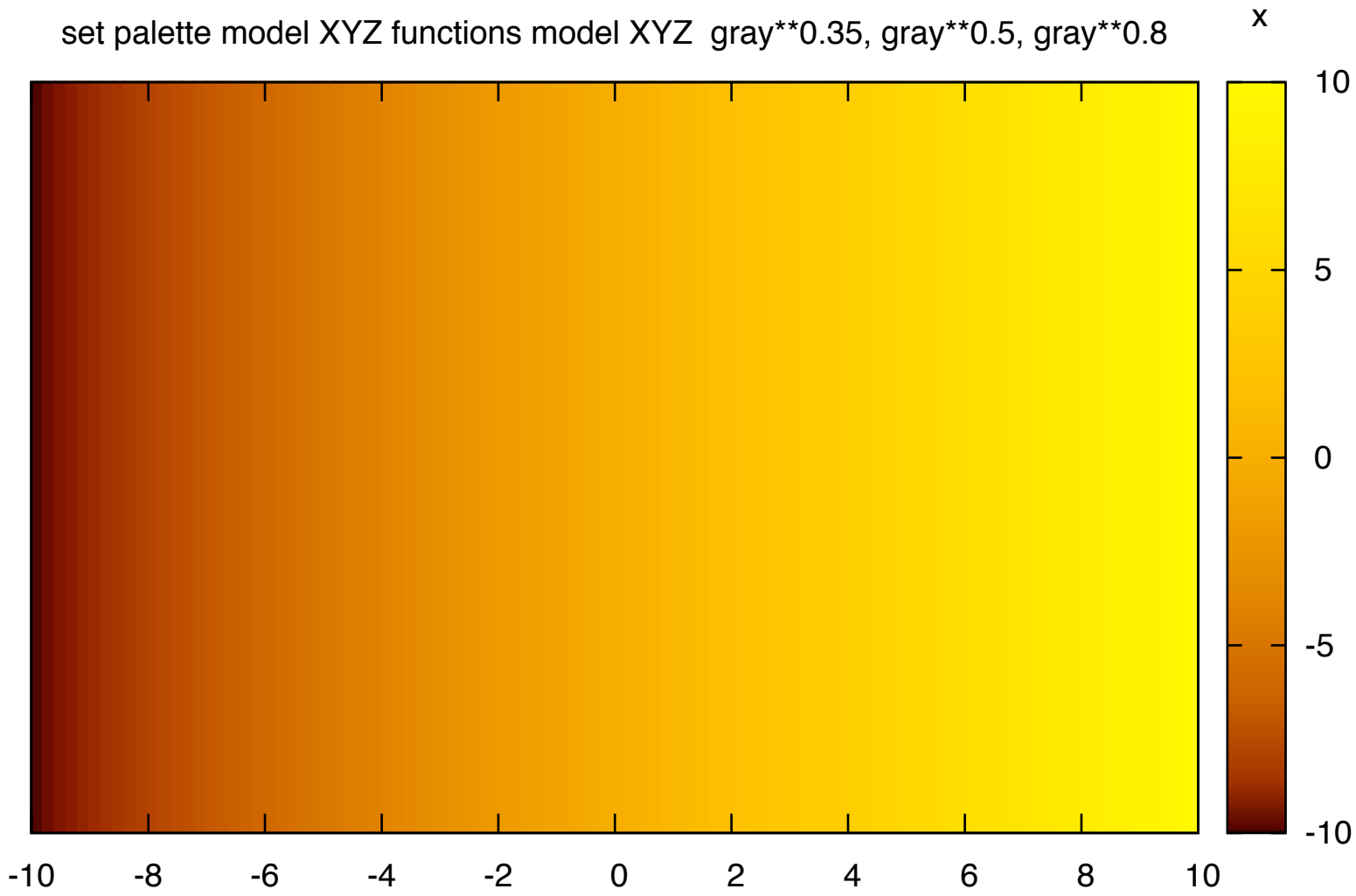

set palette model YIQ rgbformulae 7,5,15
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

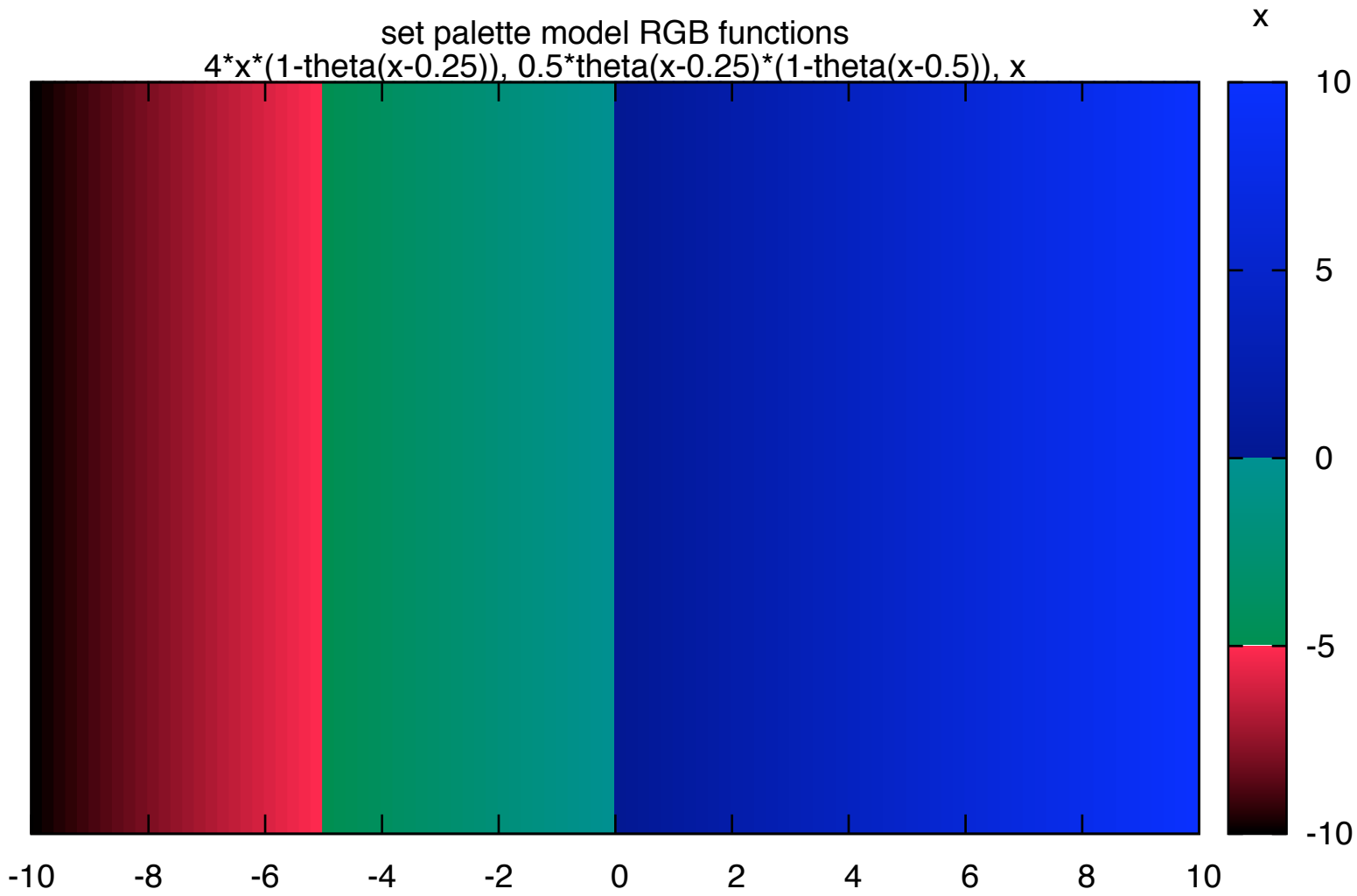

set palette model HSV rrgbformulae 3,2,2
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

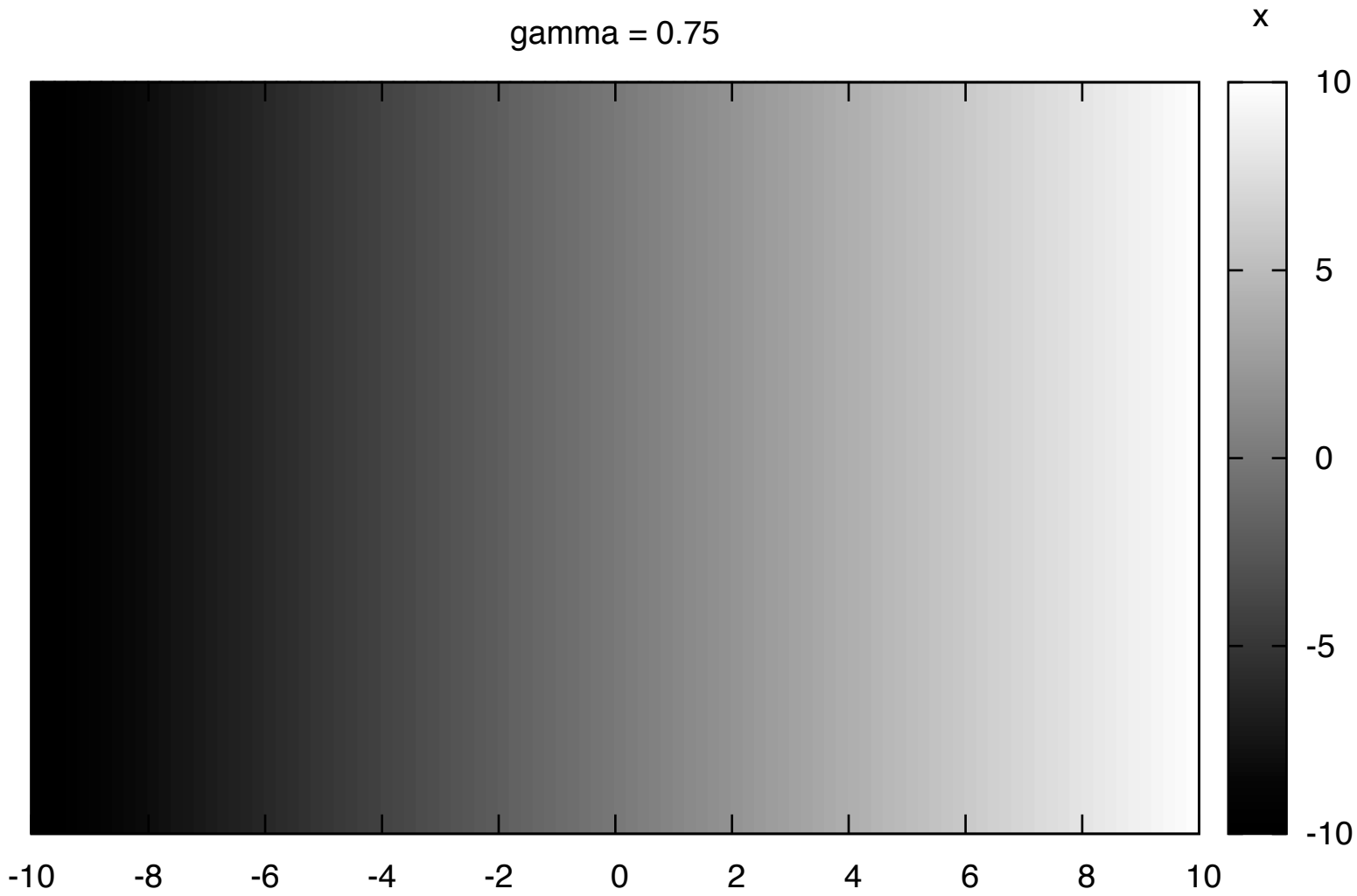

set palette functions gray, gray, gray
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

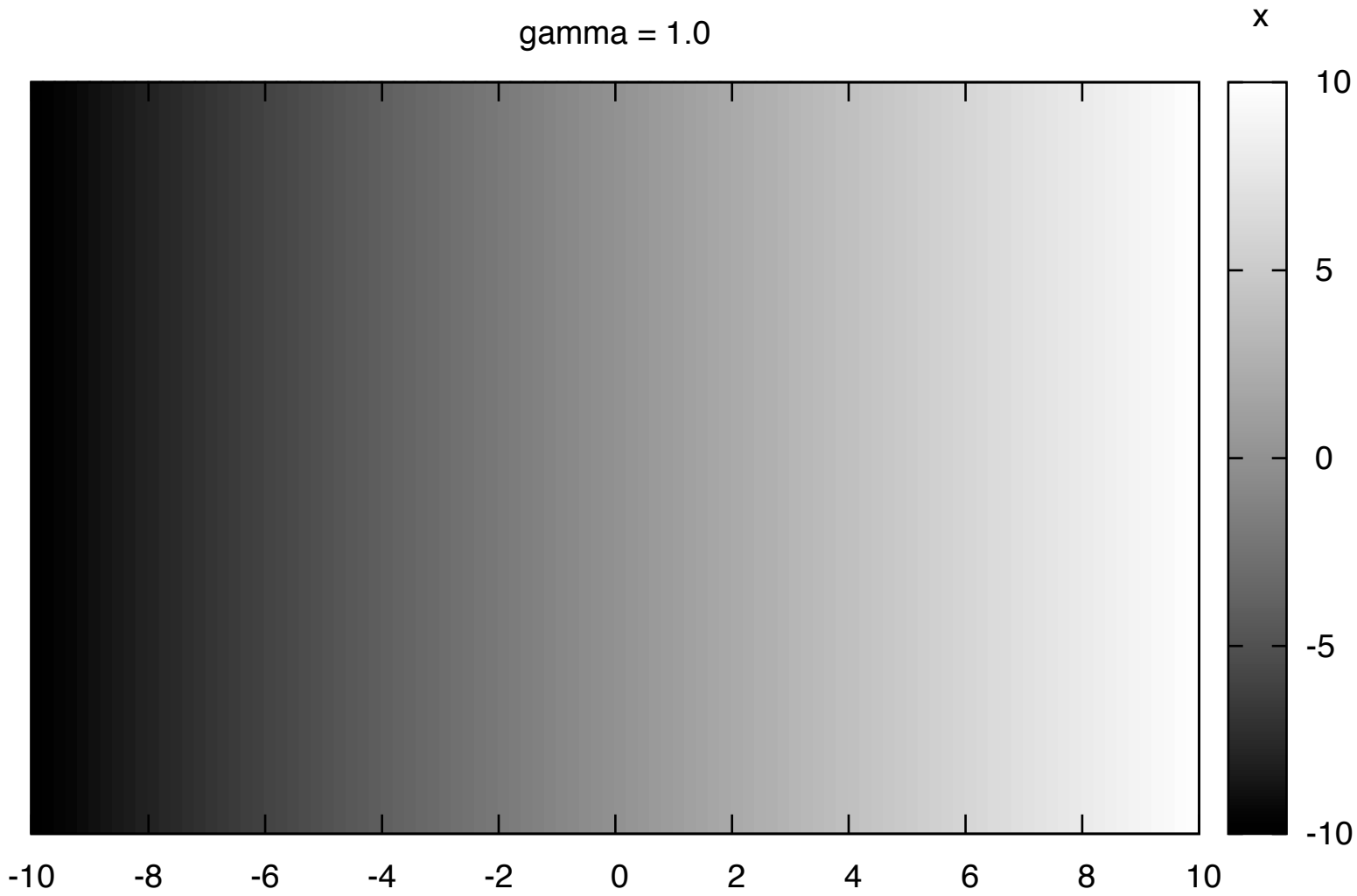

set palette functions sqrt(gray), gray**3, sin(gray*2*pi) <--> 7,5,15
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

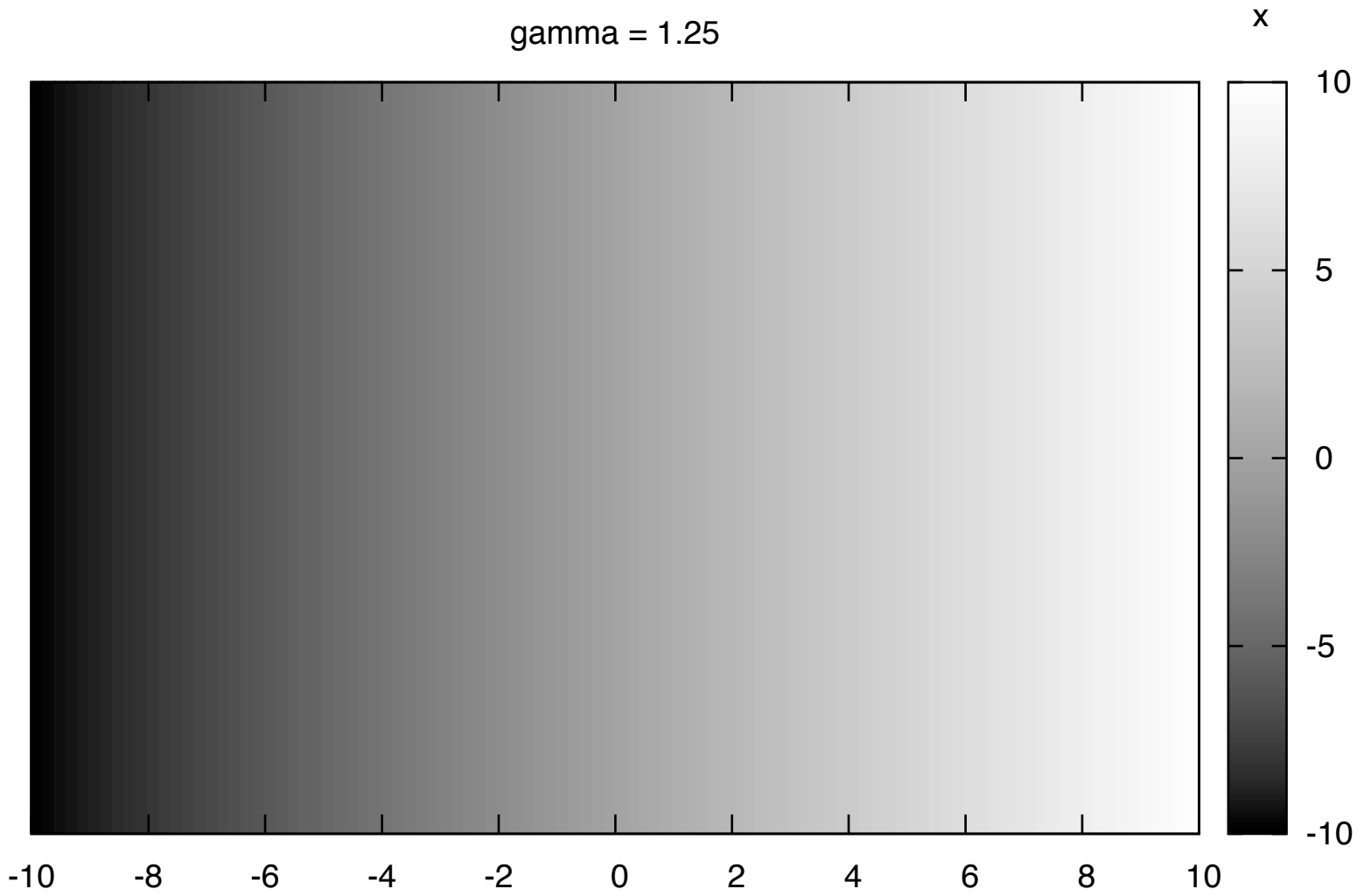

set palette rgbformulae 7,5,15
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

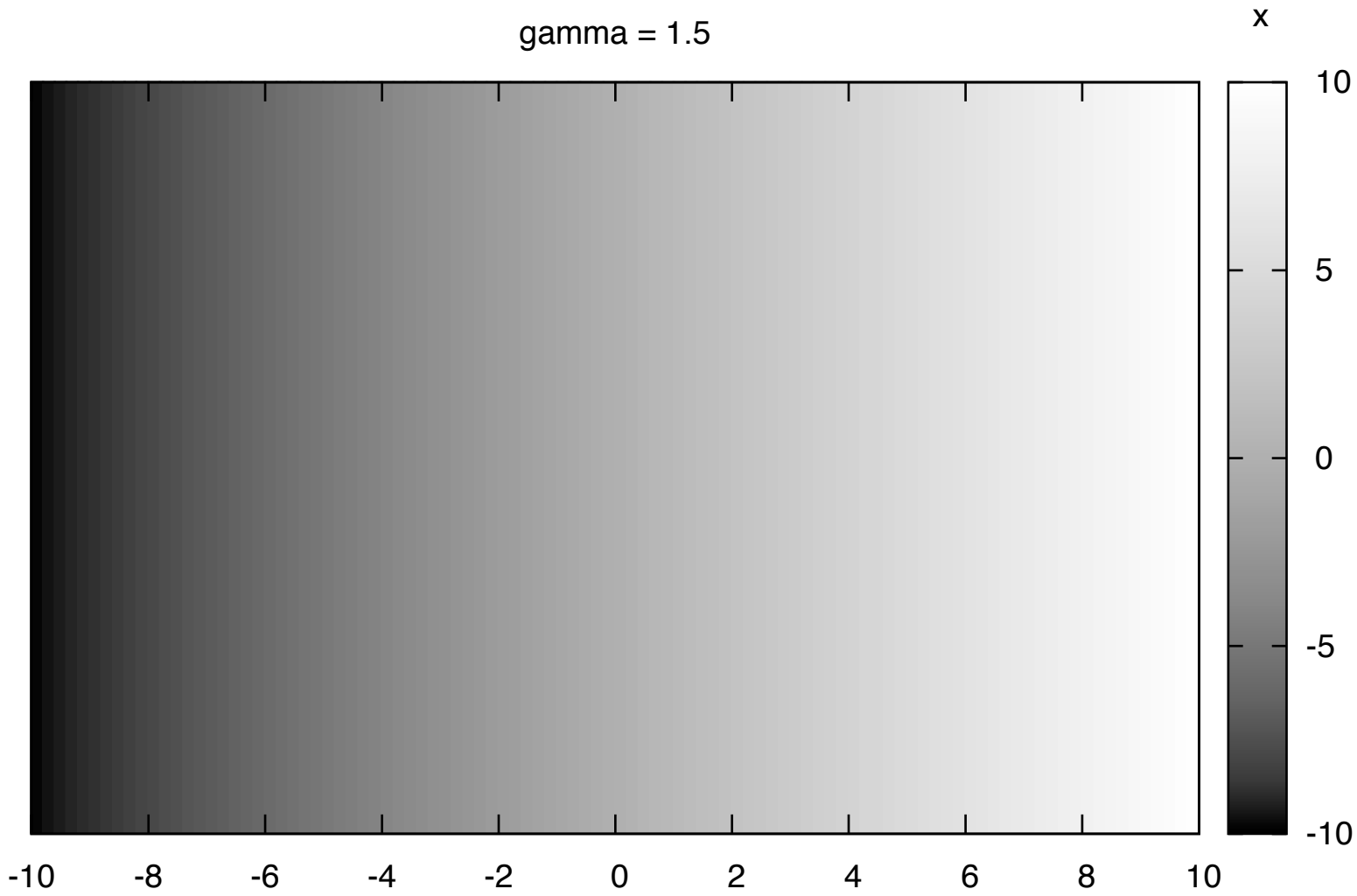

set palette model XYZ functions model XYZ gray**0.35, gray**0.5, gray**0.8
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

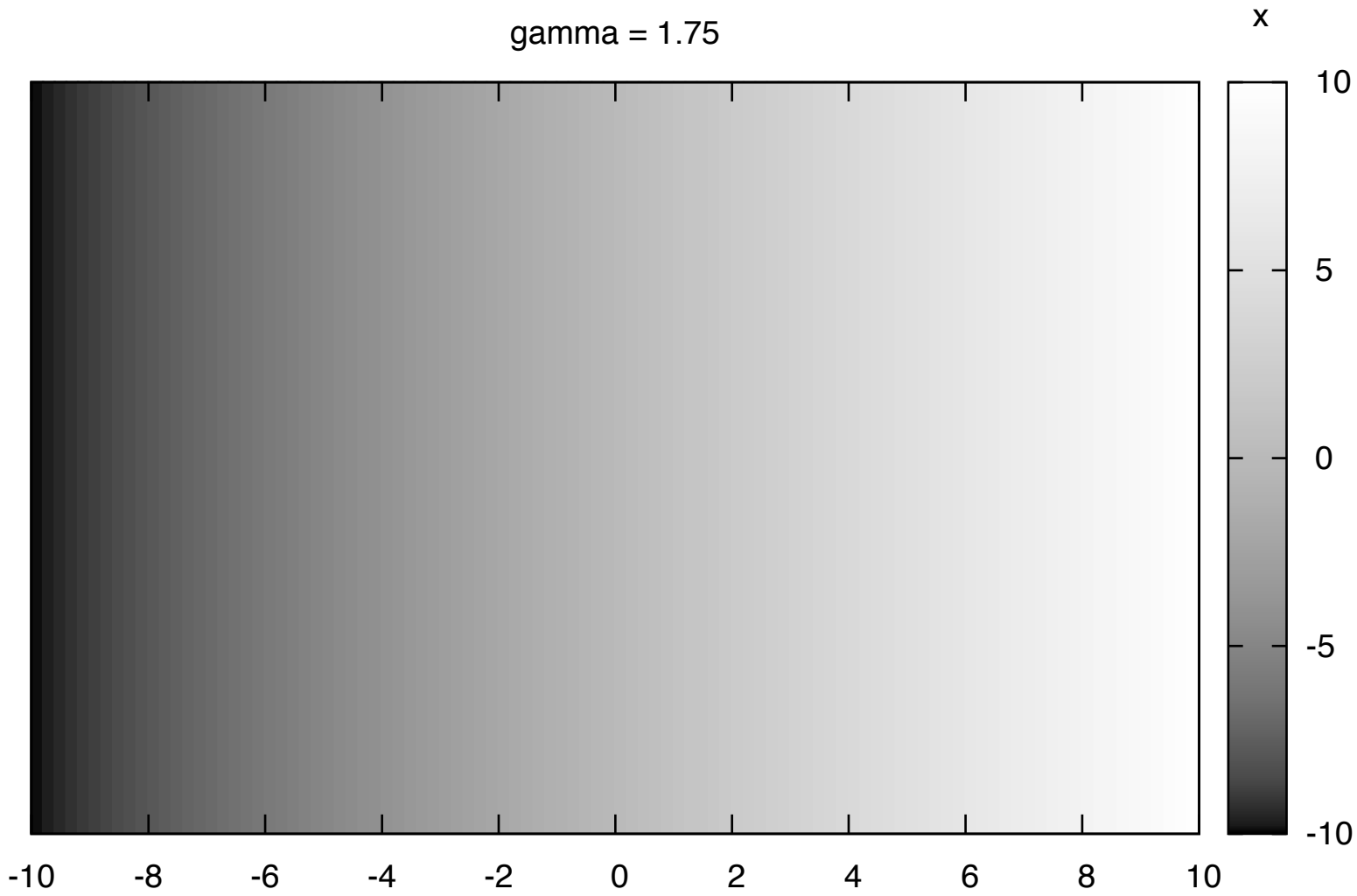

set palette model RGB functions
4*x*(1-theta(x-0.25)), 0.5*theta(x-0.25)*(1-theta(x-0.5)), x
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

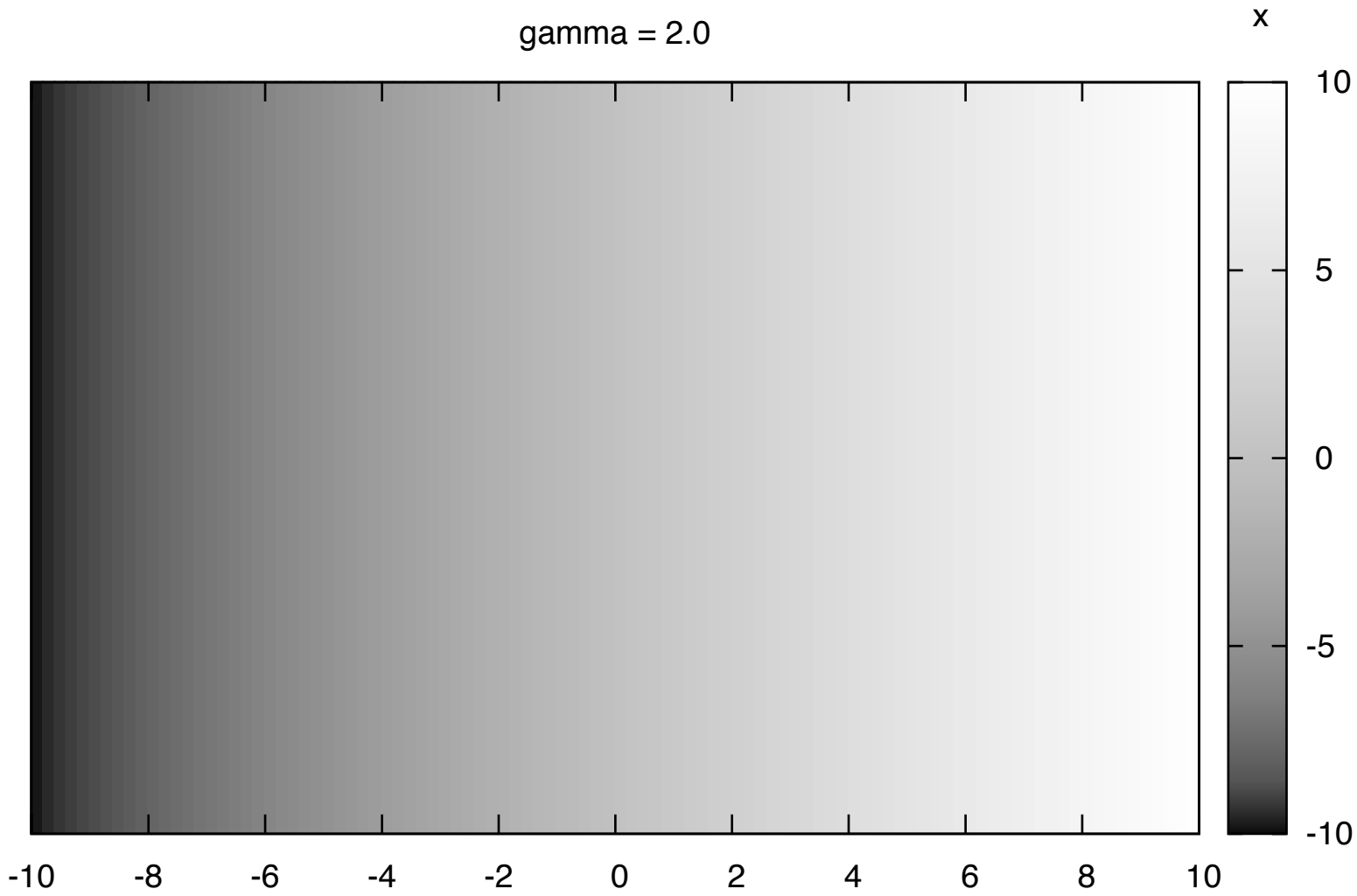

gamma = 0.75
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

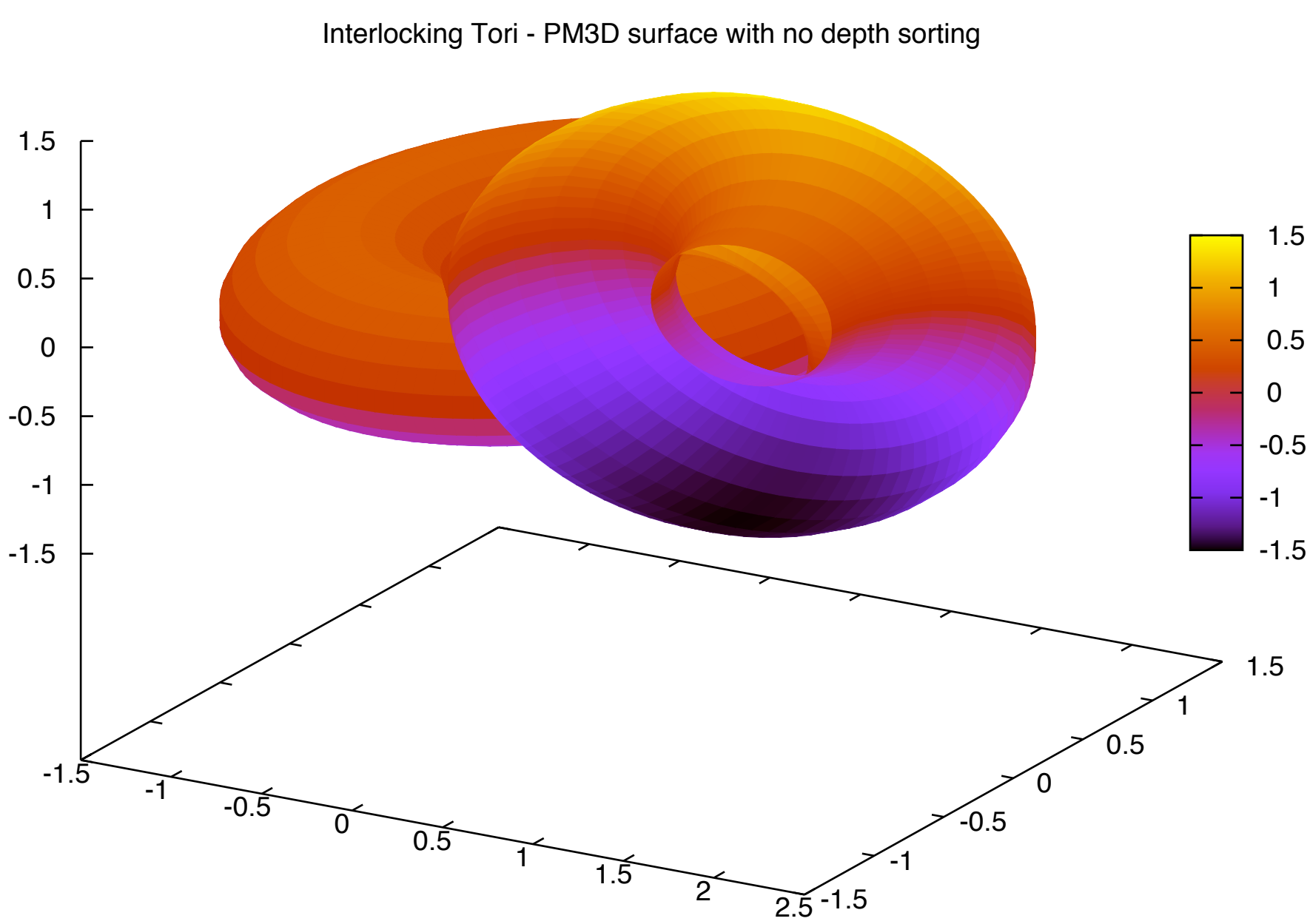

gamma = 1.0
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

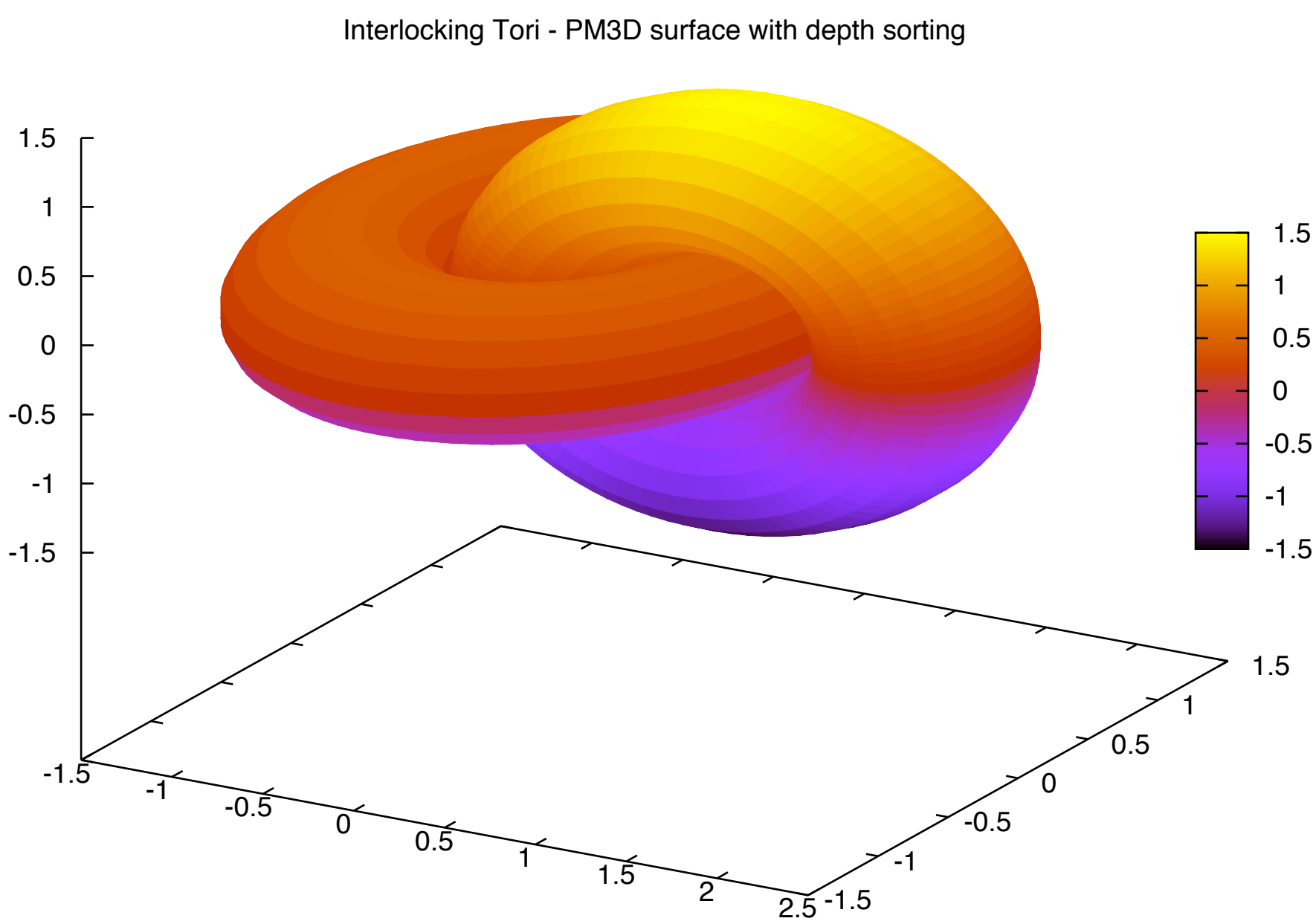

gamma = 1.25
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

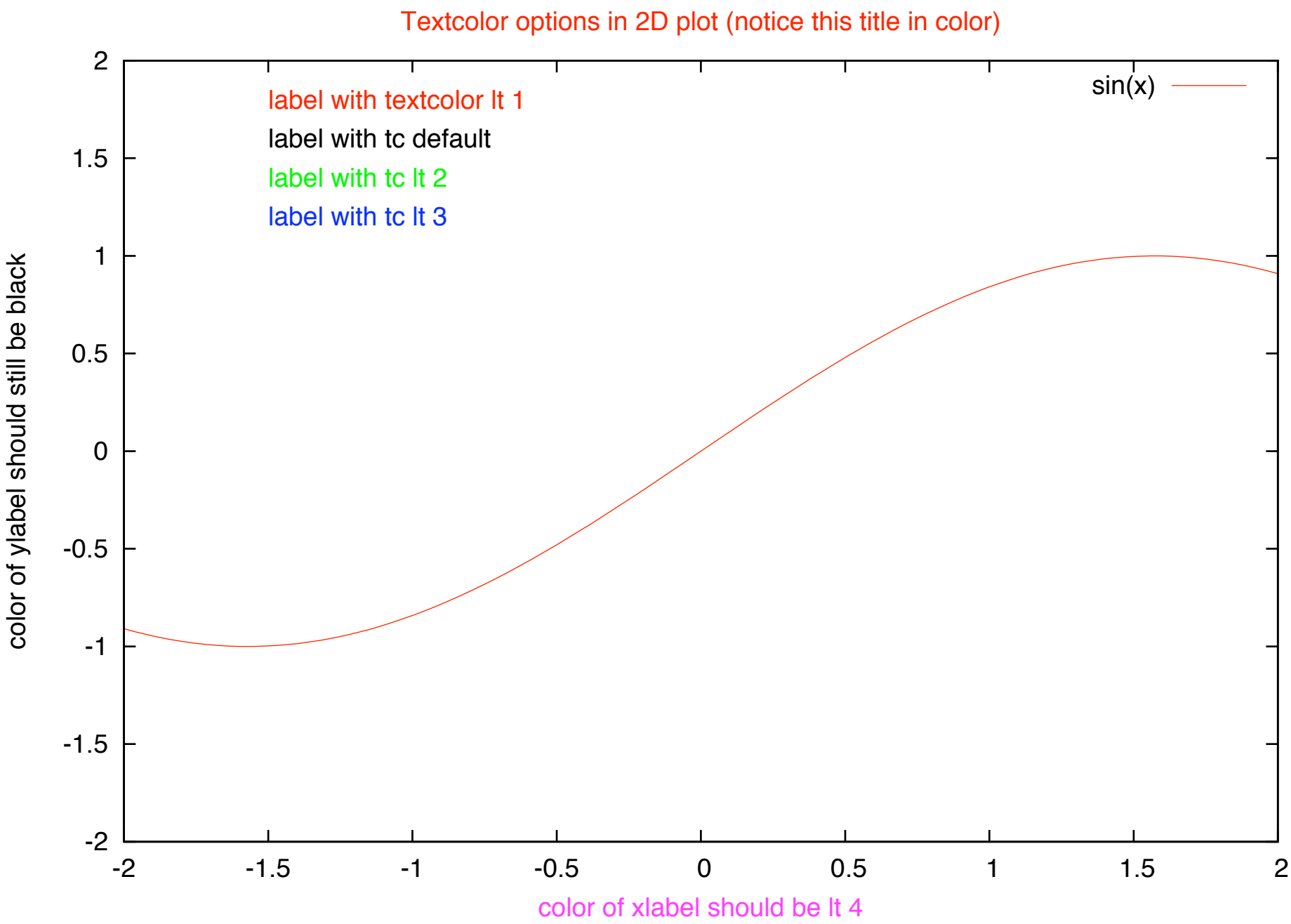

gamma = 1.5
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

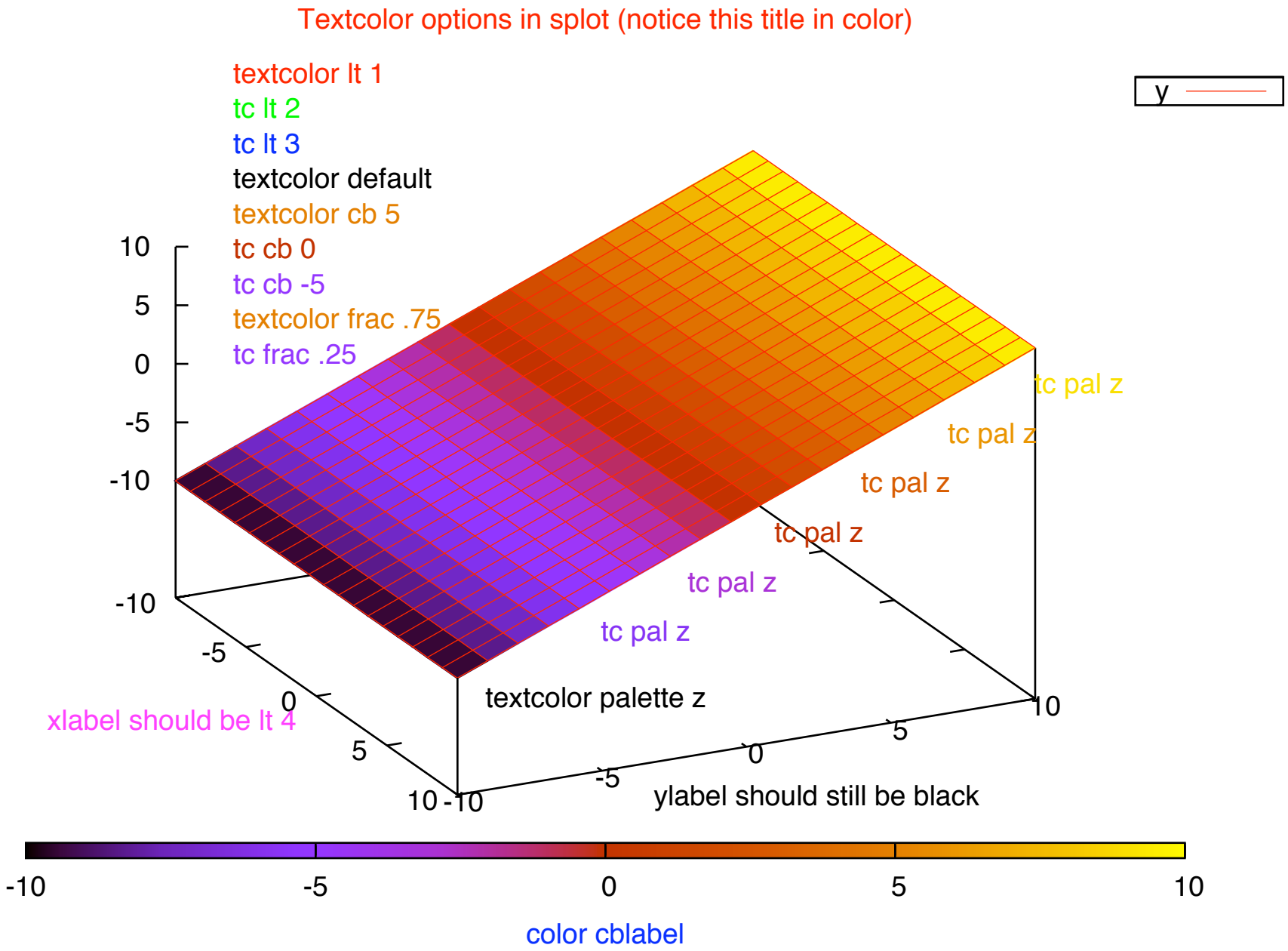

gamma = 1.75
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

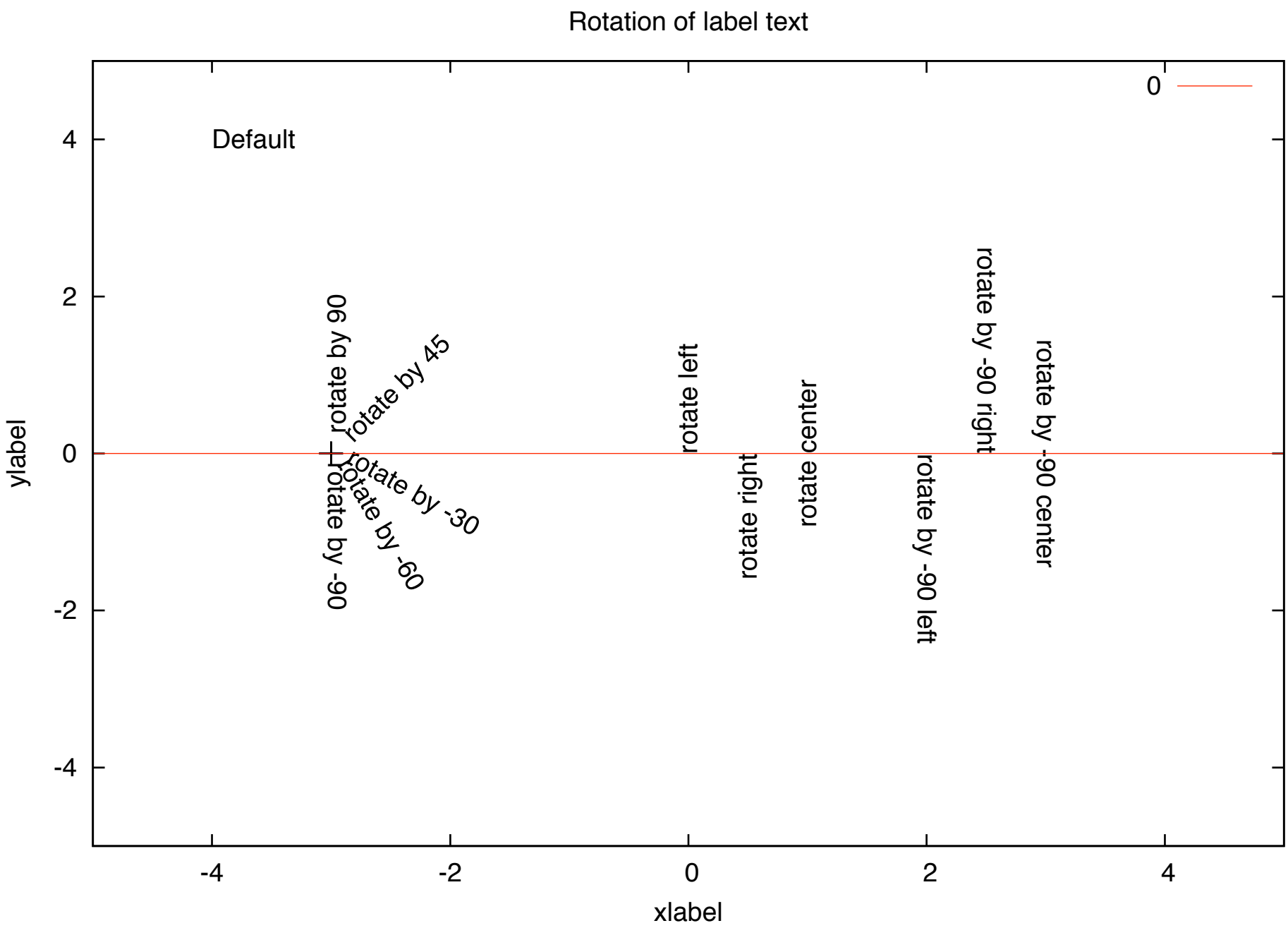

gamma = 2.0
x
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

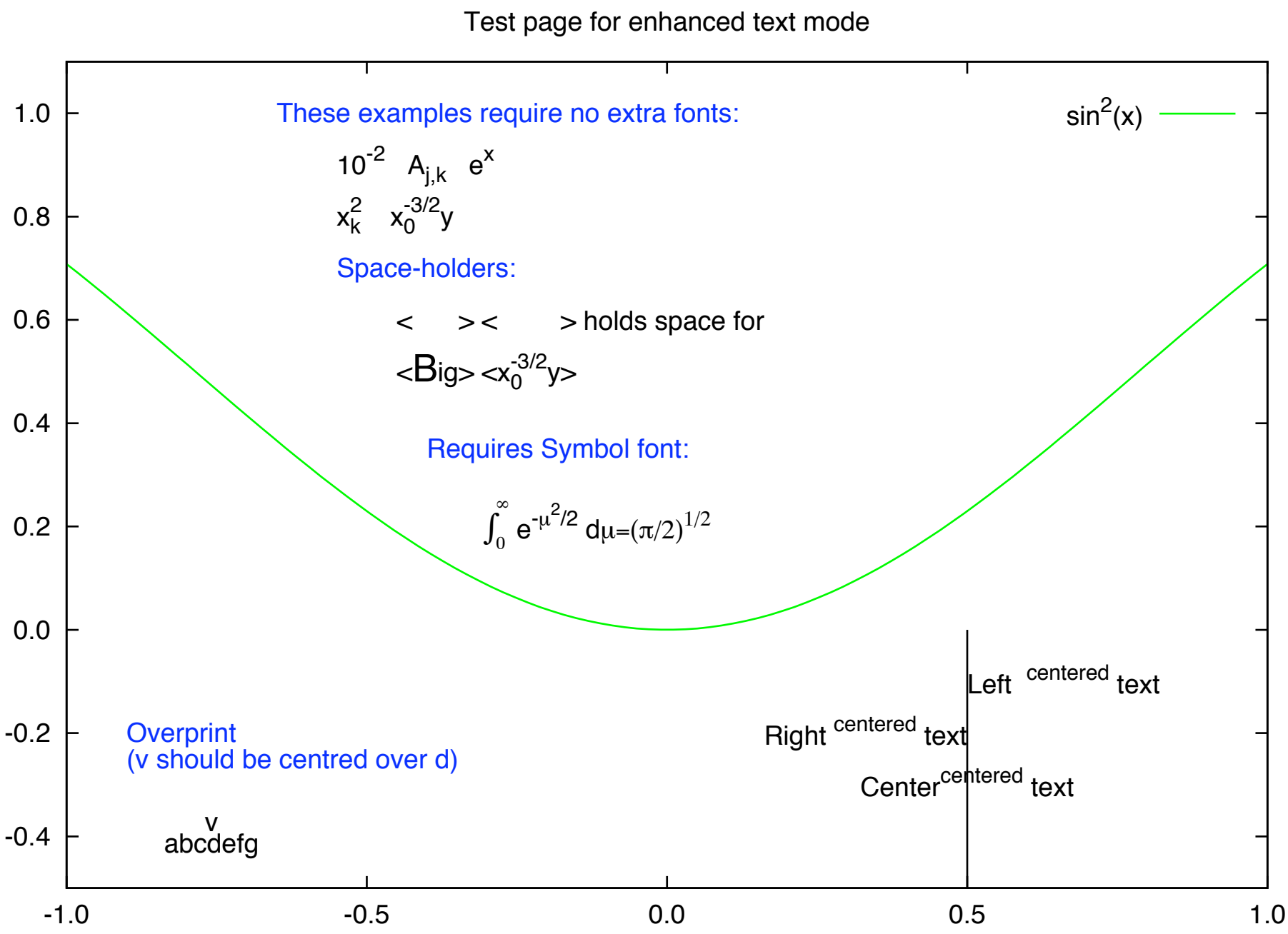

Interlocking Tori - PM3D surface with no depth sorting
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
-1.5
-1
-0.5
0
0.5
1
1.5
1.5
1
0.5
0
-0.5
-1
-1.5

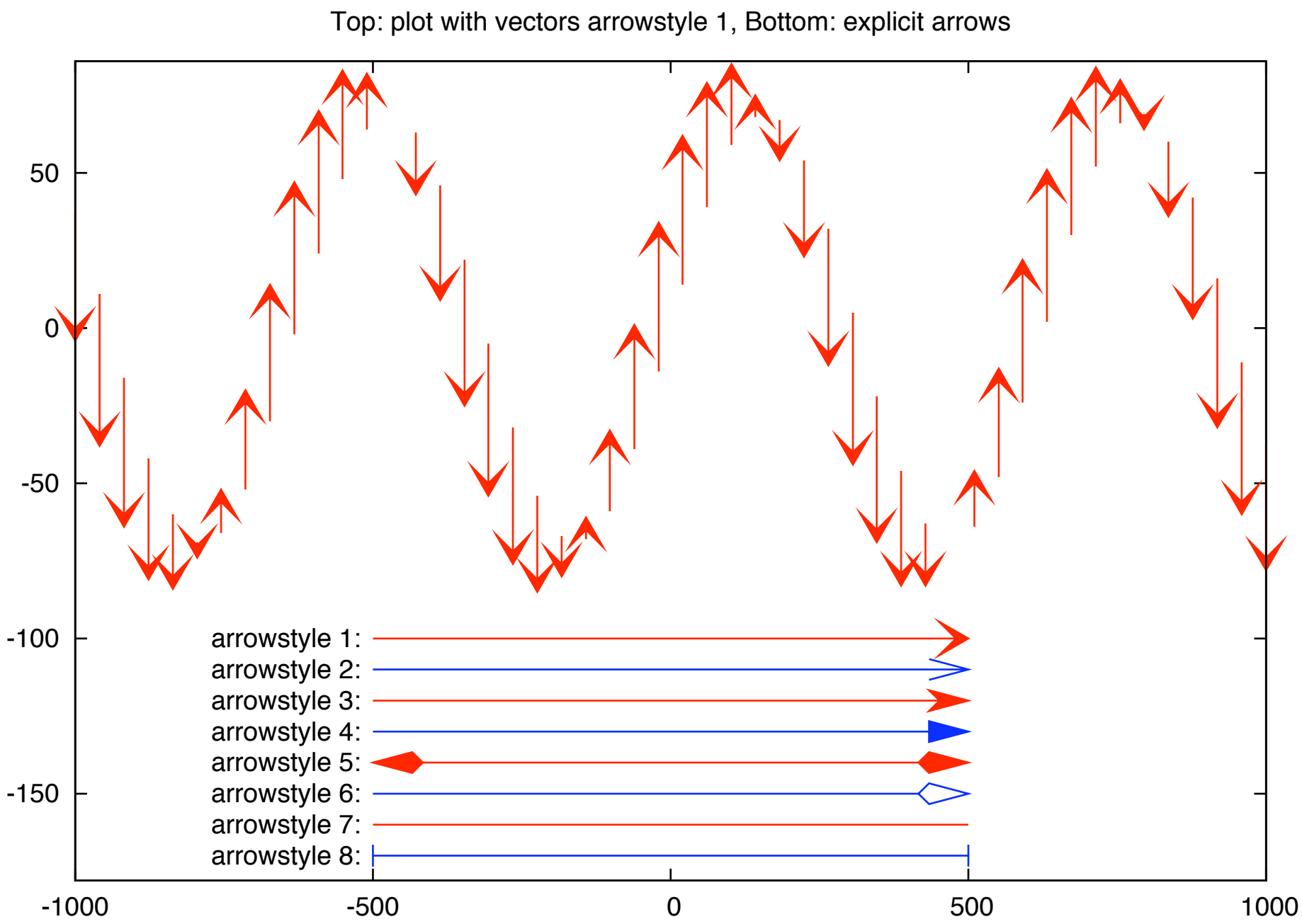

Interlocking Tori - PM3D surface with depth sorting
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
-1.5
-1
-0.5
0
0.5
1
1.5
1.5
1
0.5
0
-0.5
-1
-1.5

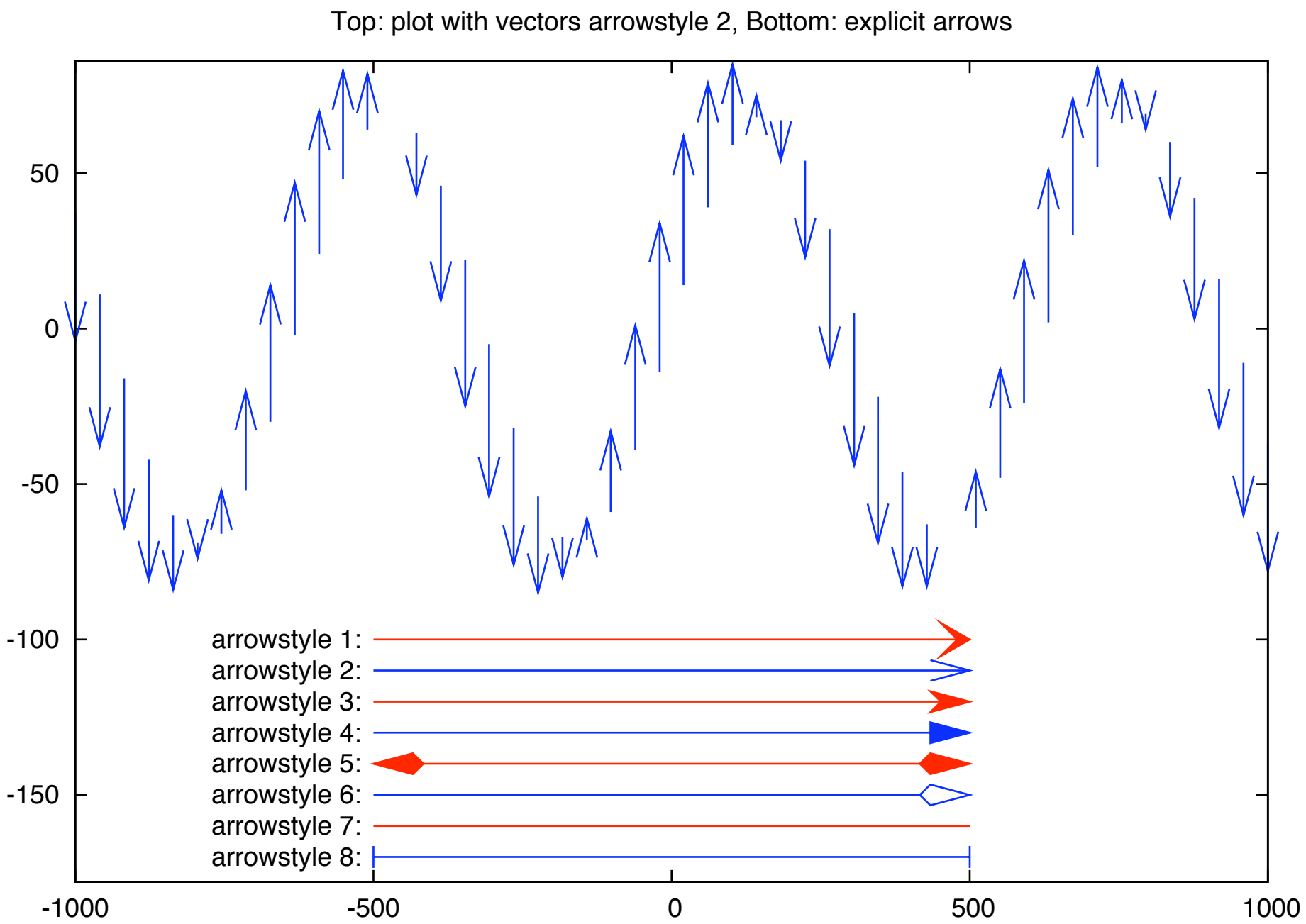

Textcolor options in 2D plot (notice this title in color)
label with textcolor lt 1
label with tc default
label with tc lt 2
label with tc lt 3
sin(x)
color of ylabel should still be black
color of xlabel should be lt 4
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

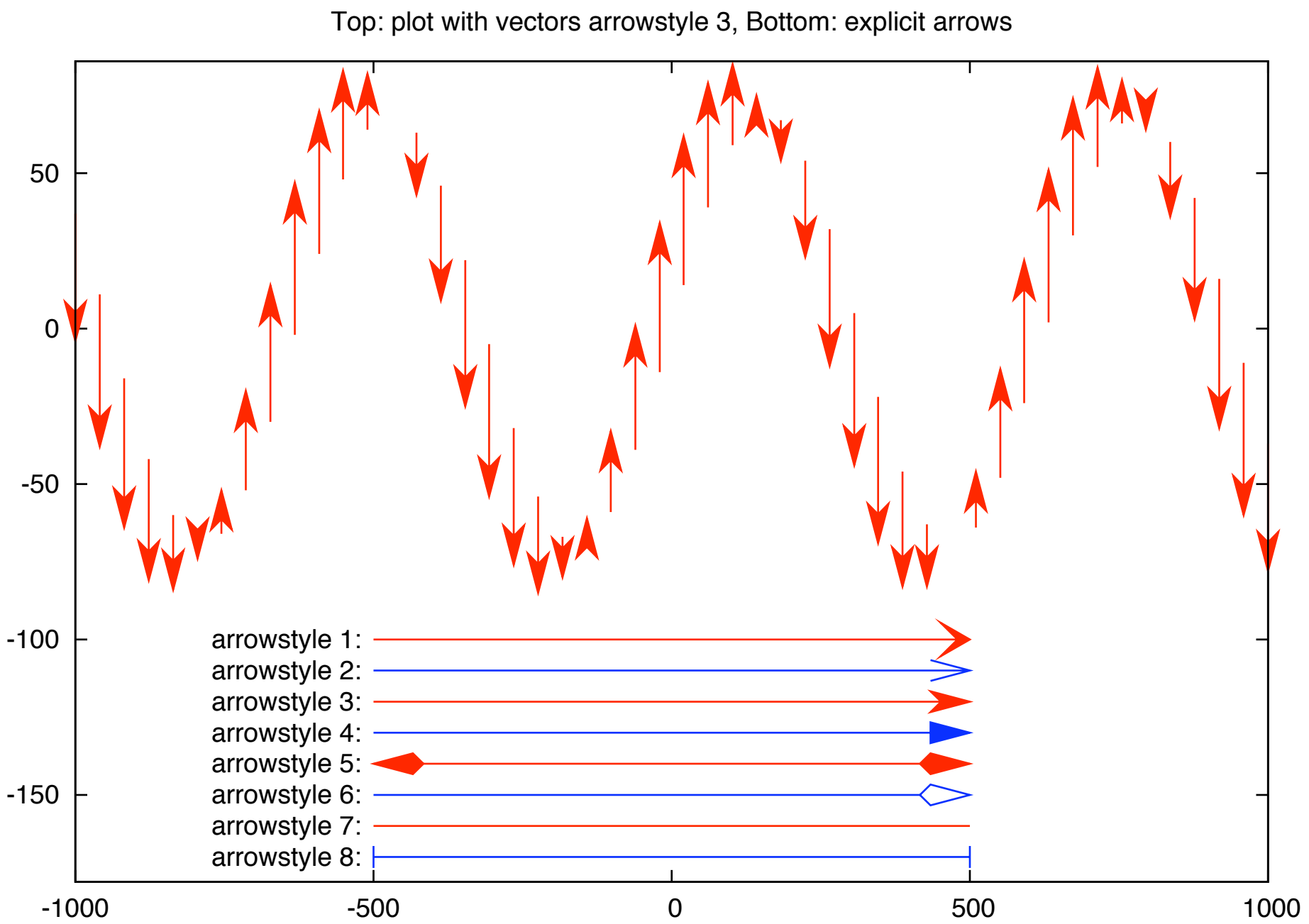

Textcolor options in splot (notice this title in color)
textcolor lt 1
tc lt 2
tc lt 3
textcolor default
textcolor cb 5
tc cb 0
tc cb -5
textcolor frac .75
tc frac .25
y
10
5
0
-5
-10
tc pal z
tc pal z
tc pal z
tc pal z
tc pal z
tc pal z
-10
-5
0
5
10
xlabel should be lt 4
textcolor palette z
10
5
0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

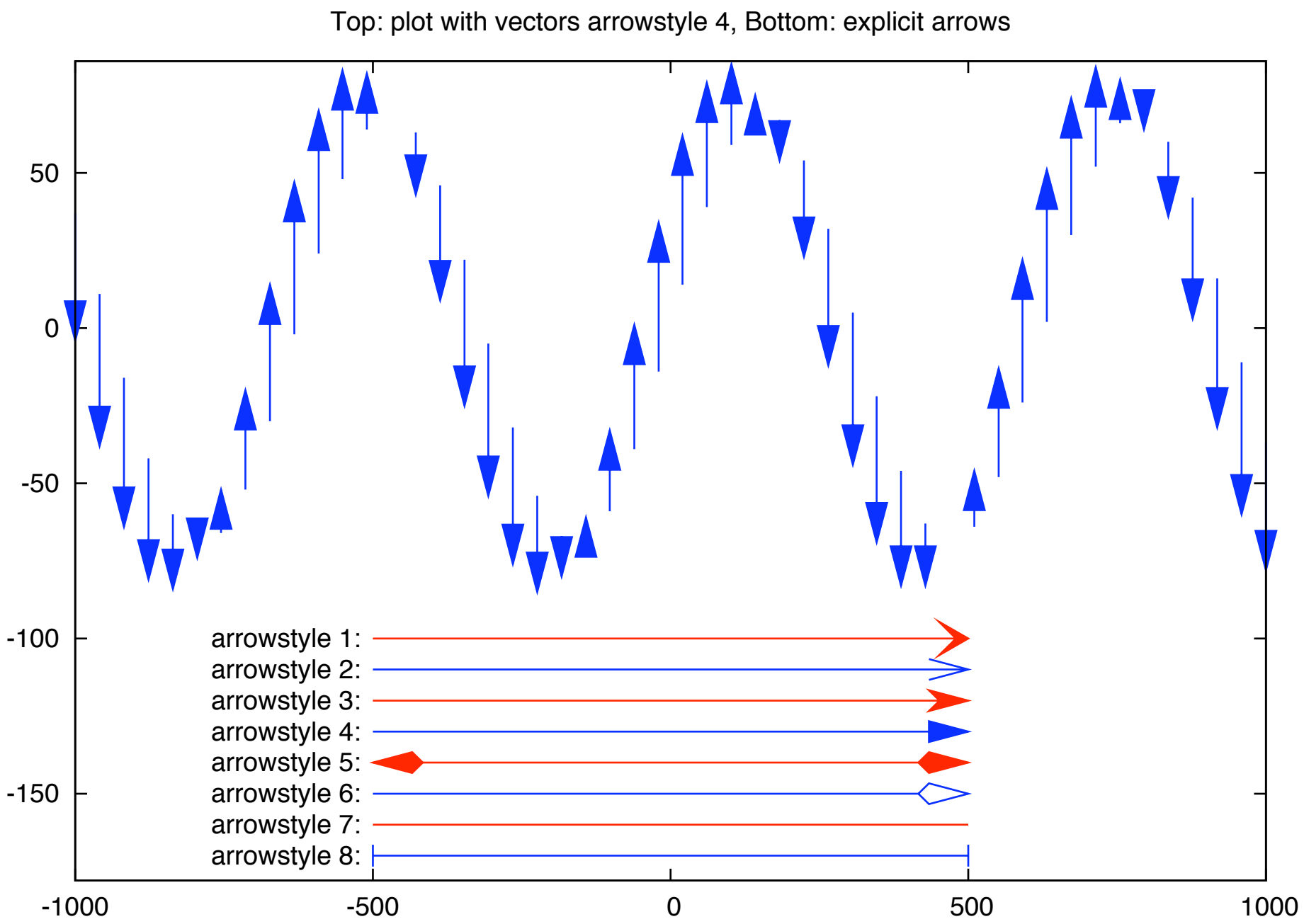

Rotation of label text
0
Default
rotate by 90
rotate by 45
rotate by -30
rotate by -60
rotate by -90
rotate left
rotate right
rotate center
rotate by -90 left
rotate by -90 right
rotate by -90 center
ylabel
xlabel
4
2
0
-2
-4
-4
-2
0
2
4

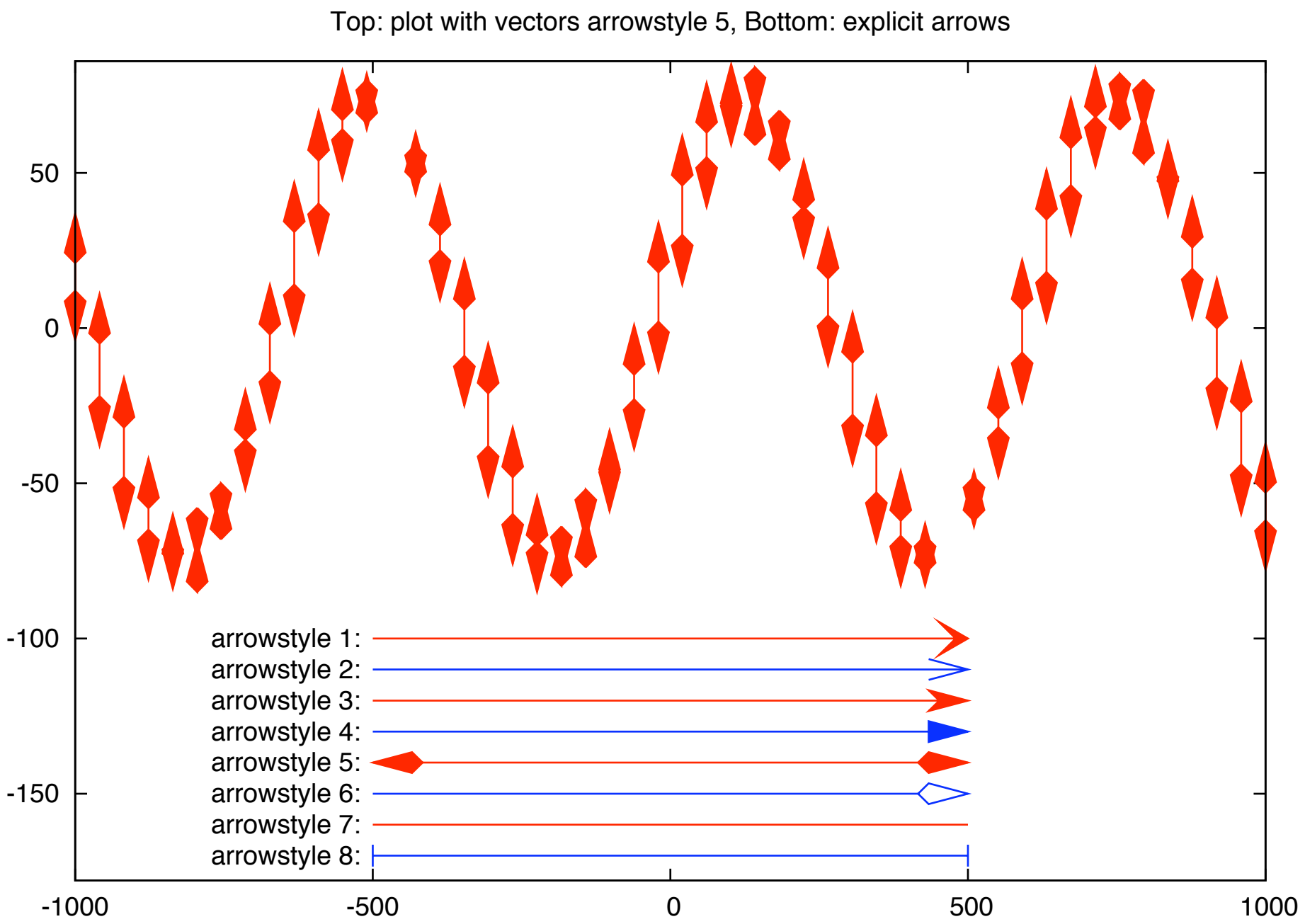

Test page for enhanced text mode
These examples require no extra fonts:
10^-2 A_j,k e^x
x_k^2 x_0^-3/2 y
Space-holders:
< >< > holds space for
<Big> <x_0^-3/2 y>
Requires Symbol font:
∫_0^∞ e^-μ^2/2 dμ=(π/2)^1/2
sin^2(x)
Overprint
(v should be centred over d)
v
abcdefg
Left centered text
Right centered text
Center centered text
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-1.0
-0.5
0.0
0.5
1.0

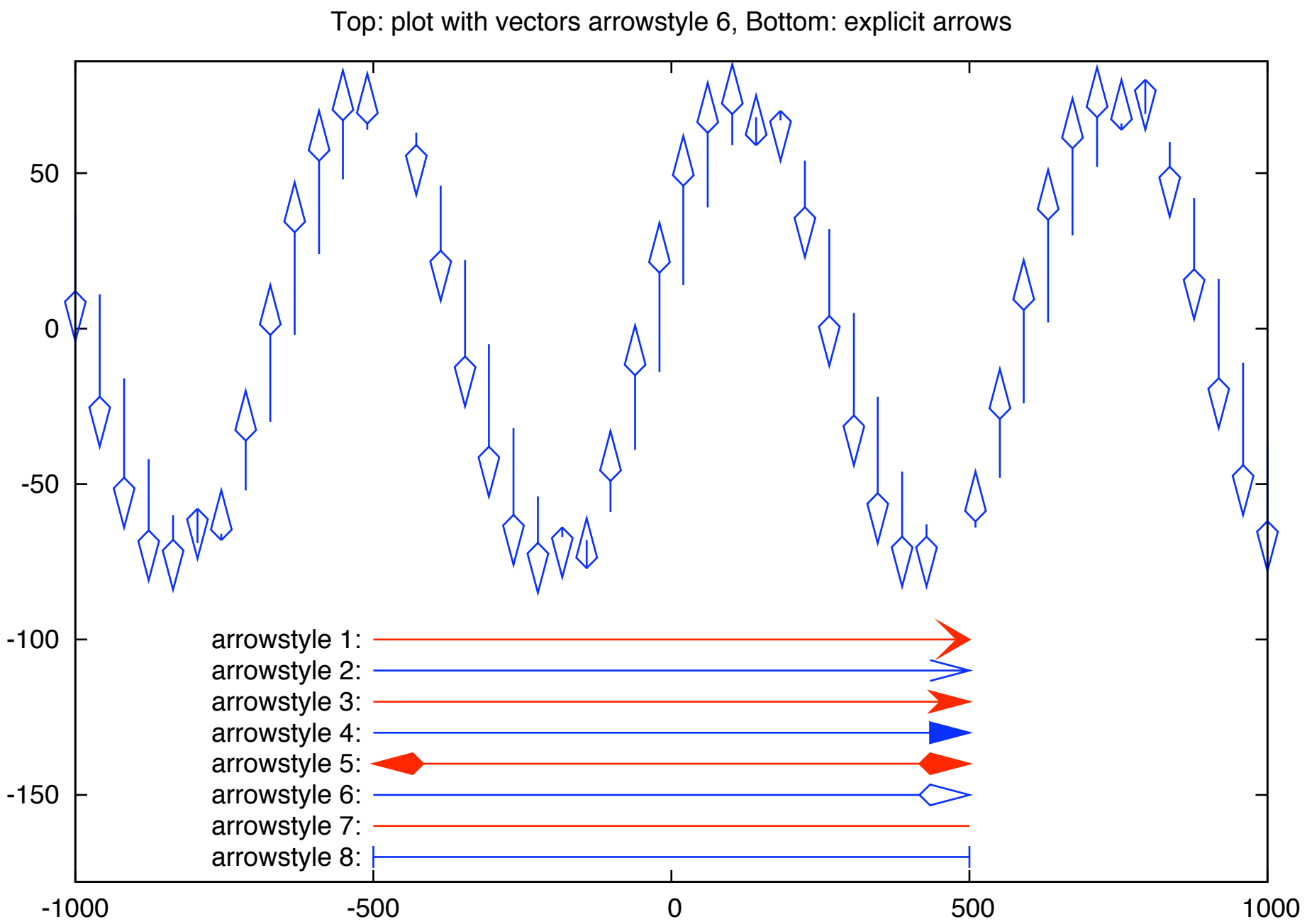

Top: plot with vectors arrowstyle 1, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:

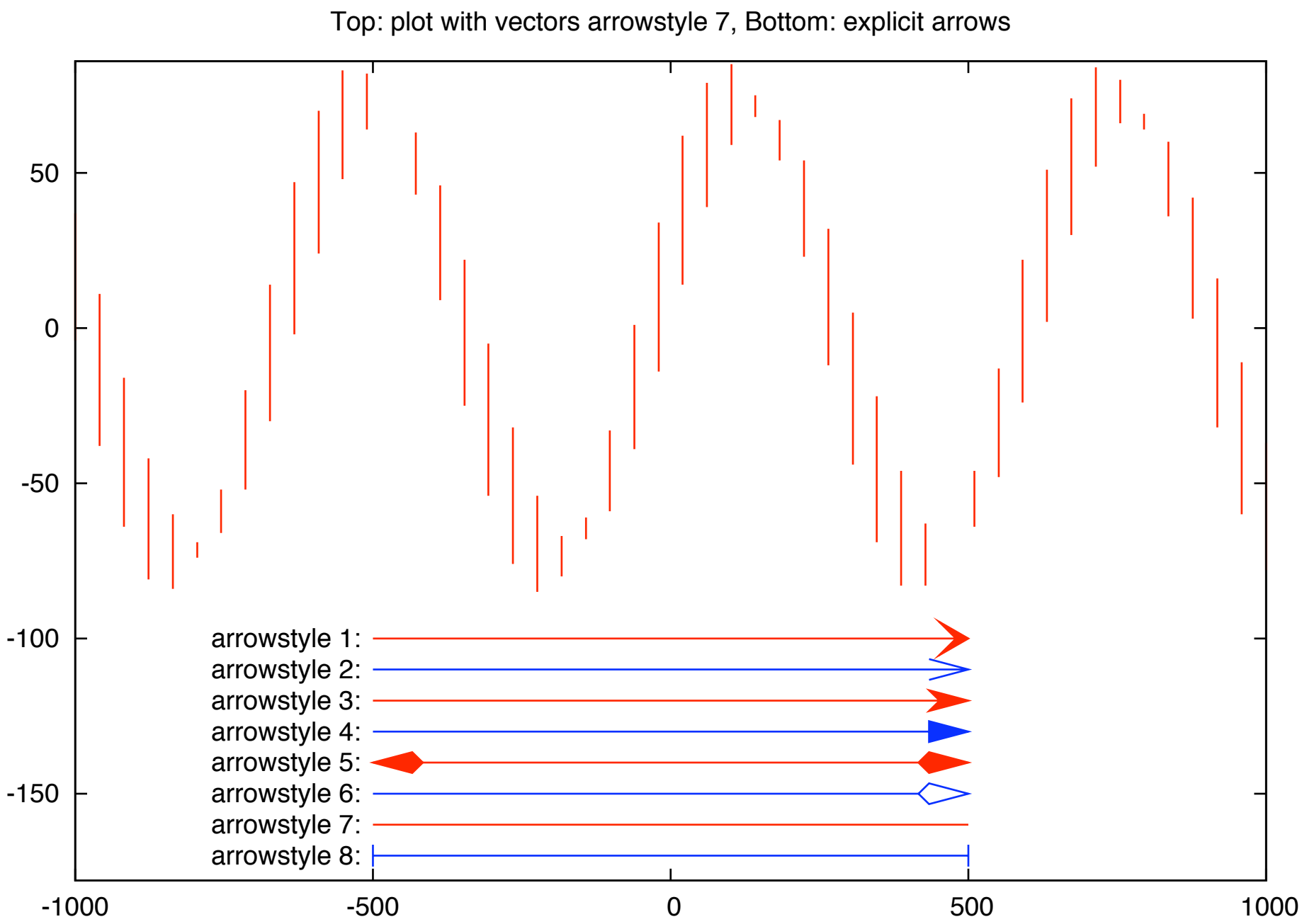

Top: plot with vectors arrowstyle 2, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:

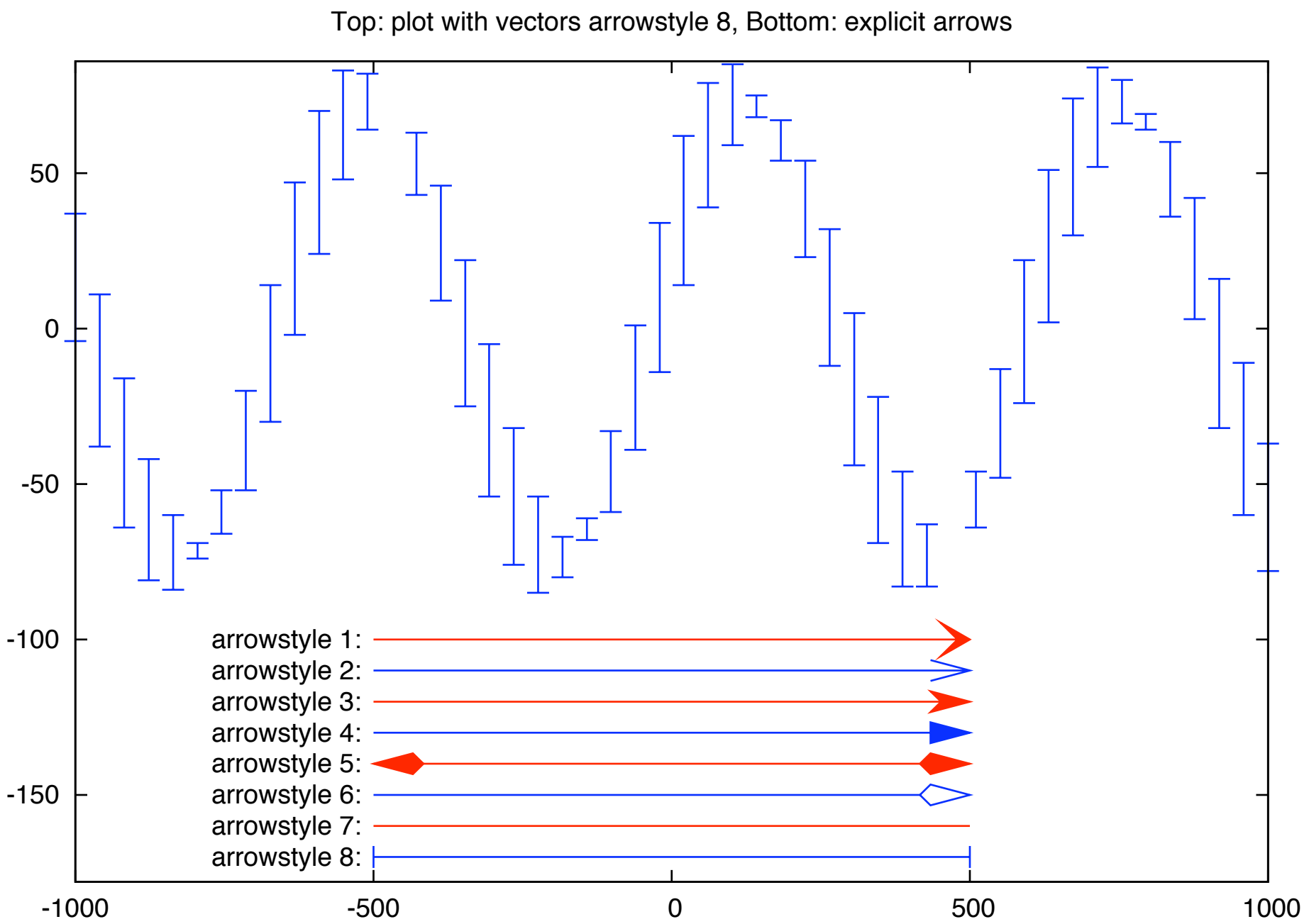

Top: plot with vectors arrowstyle 3, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:

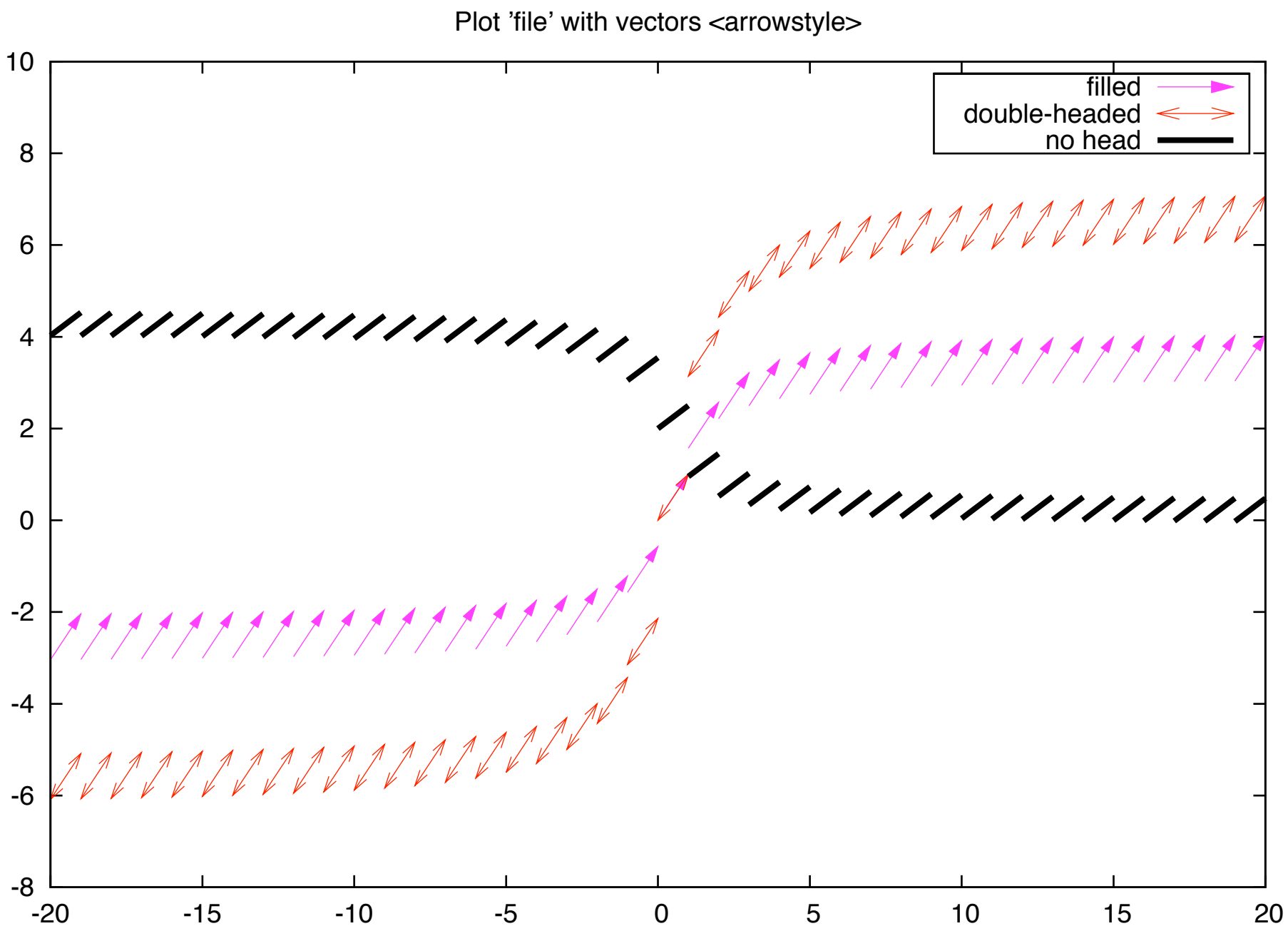

Top: plot with vectors arrowstyle 4, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:

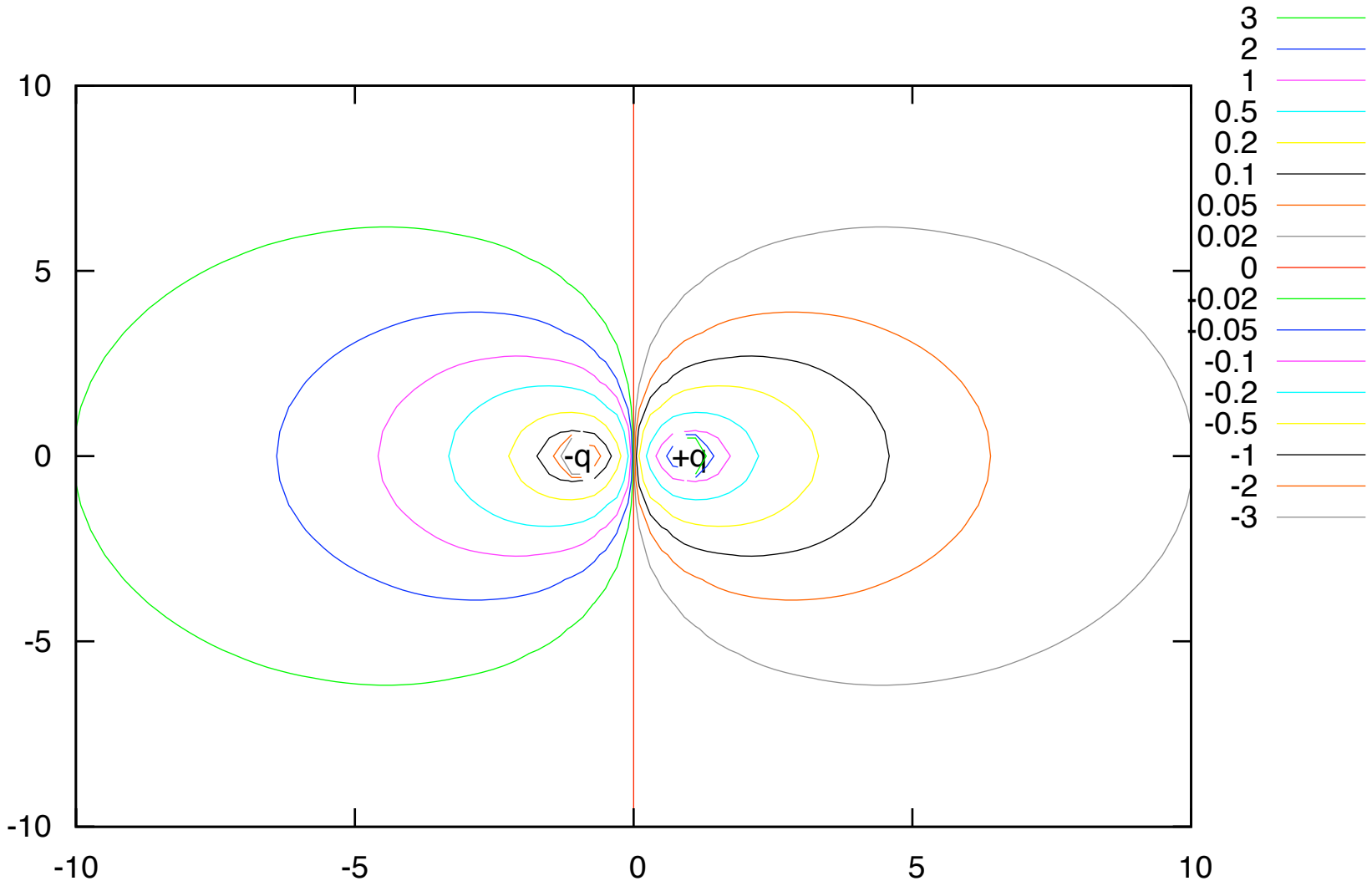

Top: plot with vectors arrowstyle 5, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:

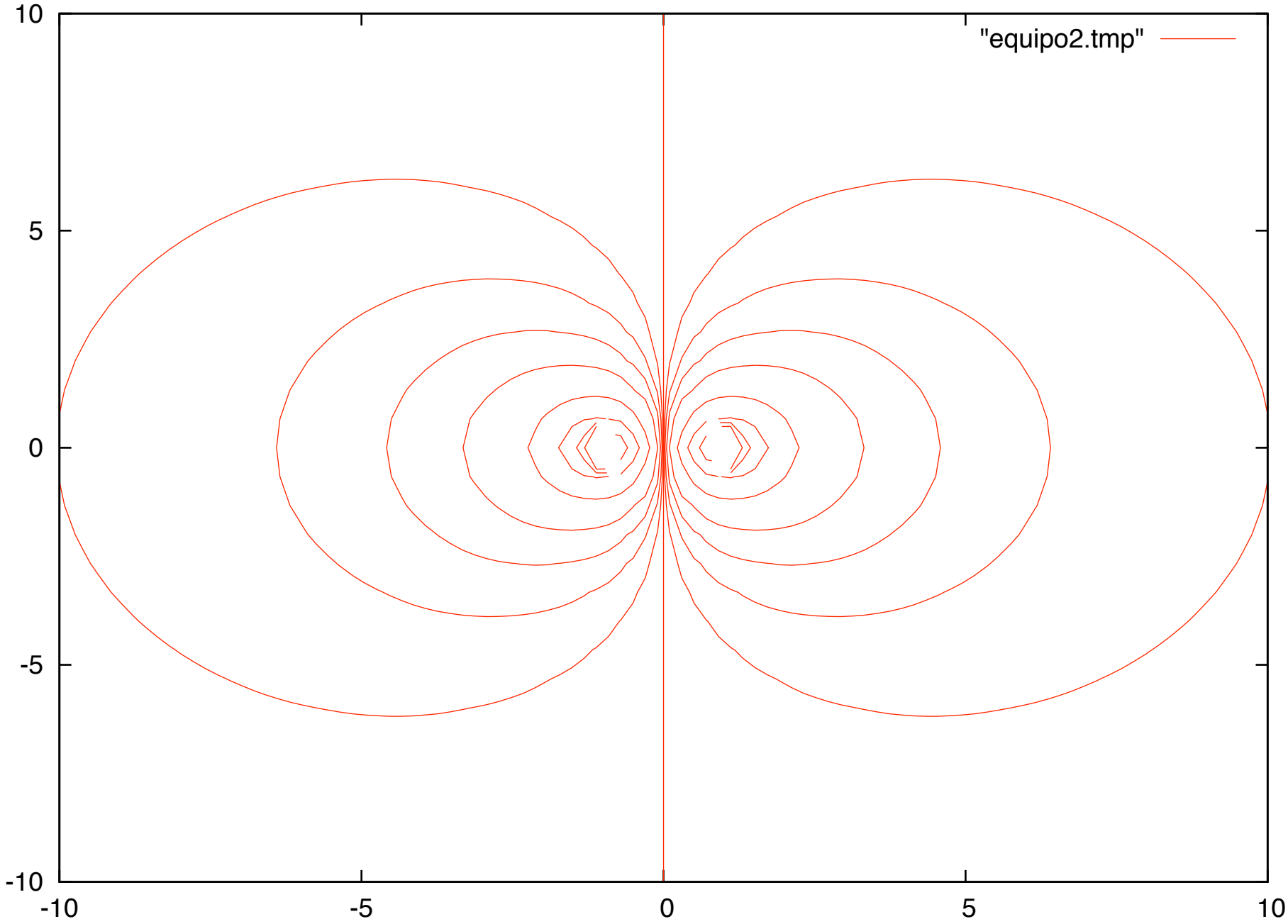

Top: plot with vectors arrowstyle 6, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:

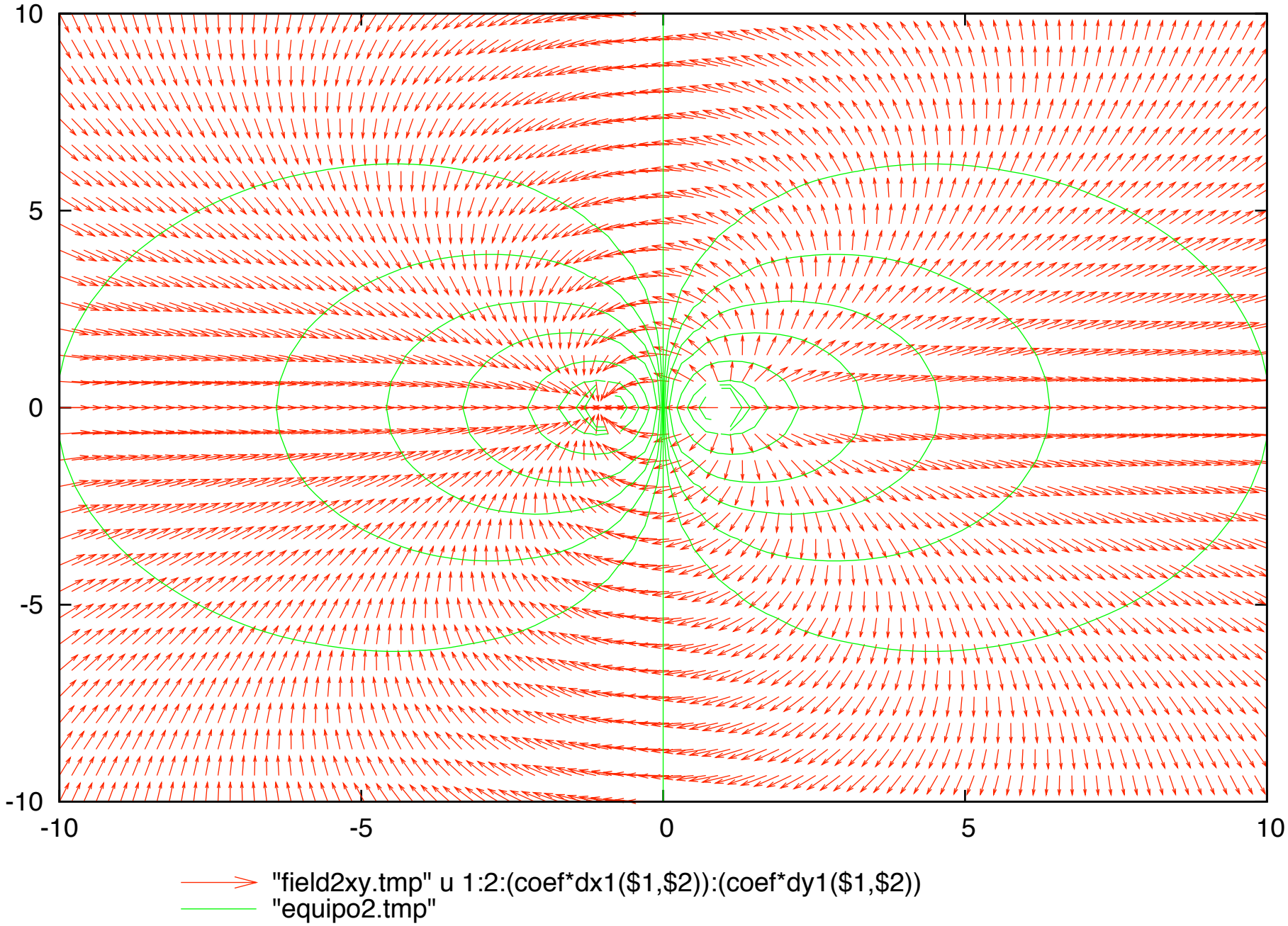

Top: plot with vectors arrowstyle 7, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:

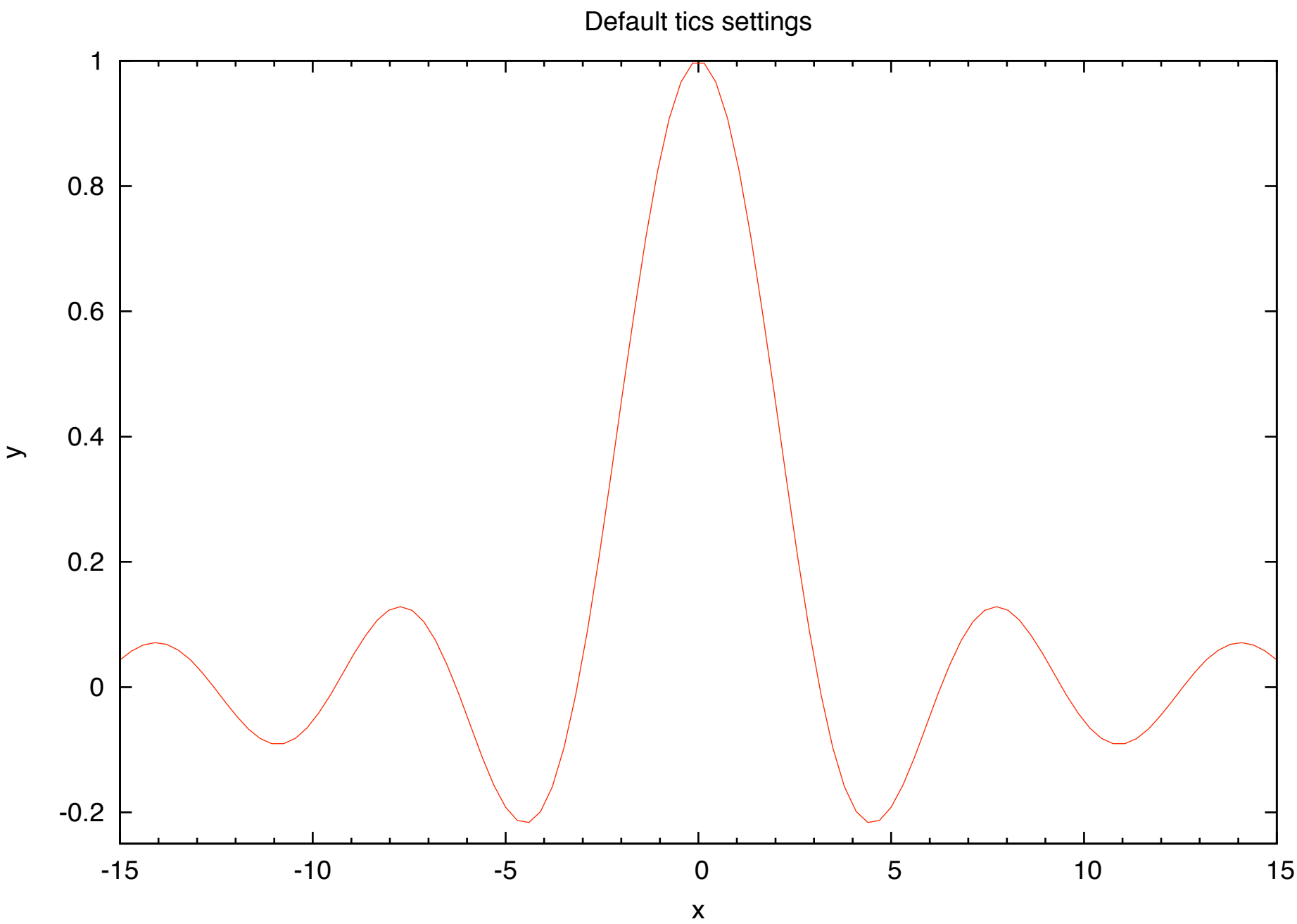

Top: plot with vectors arrowstyle 8, Bottom: explicit arrows
50
0
-50
-100
-150
-1000
-500
0
500
1000
arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:

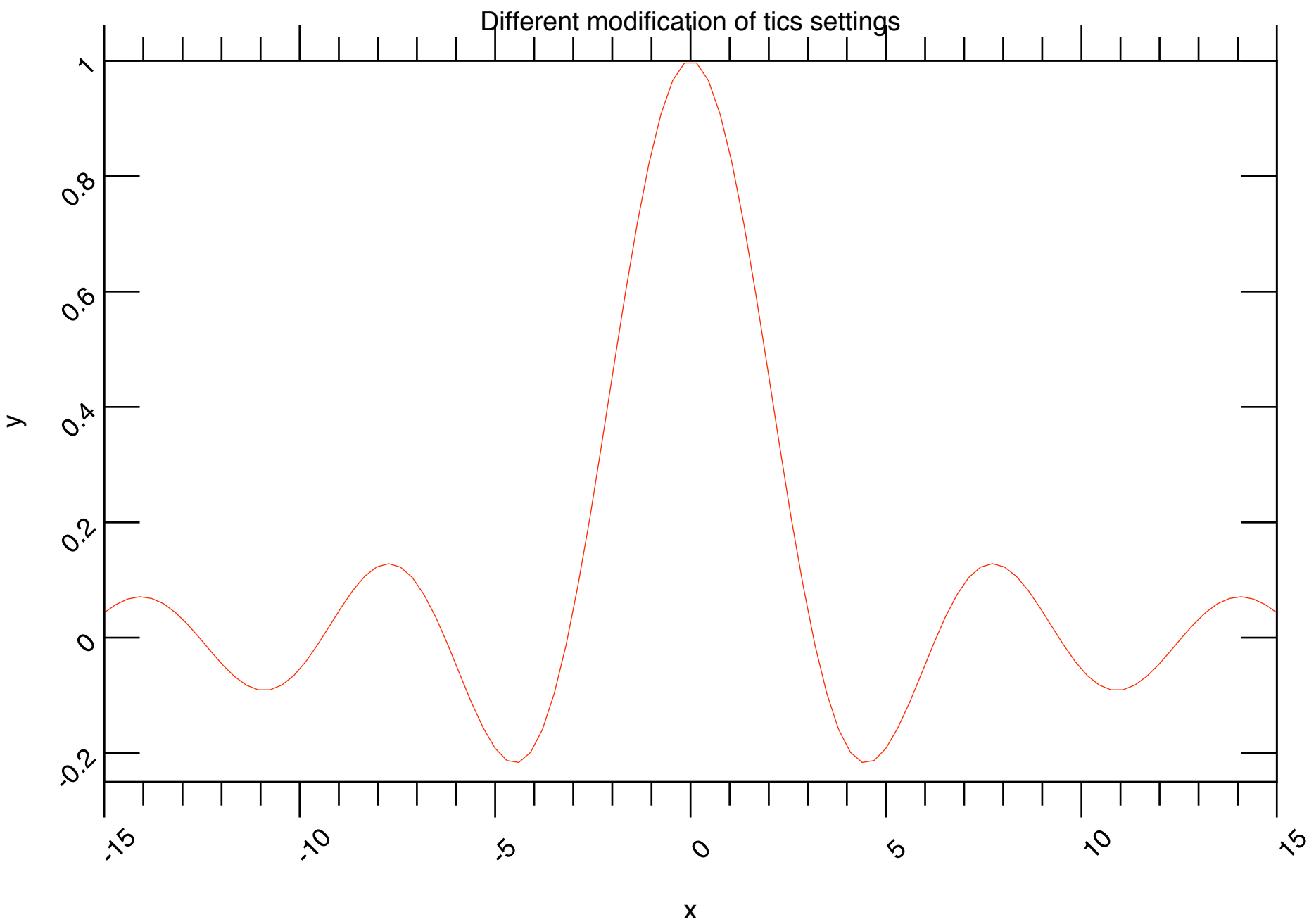

Plot 'file' with vectors <arrowstyle>
filled
double-headed
no head
10
8
6
4
2
0
-2
-4
-6
-8
-20
-15
-10
-5
0
5
10
15
20

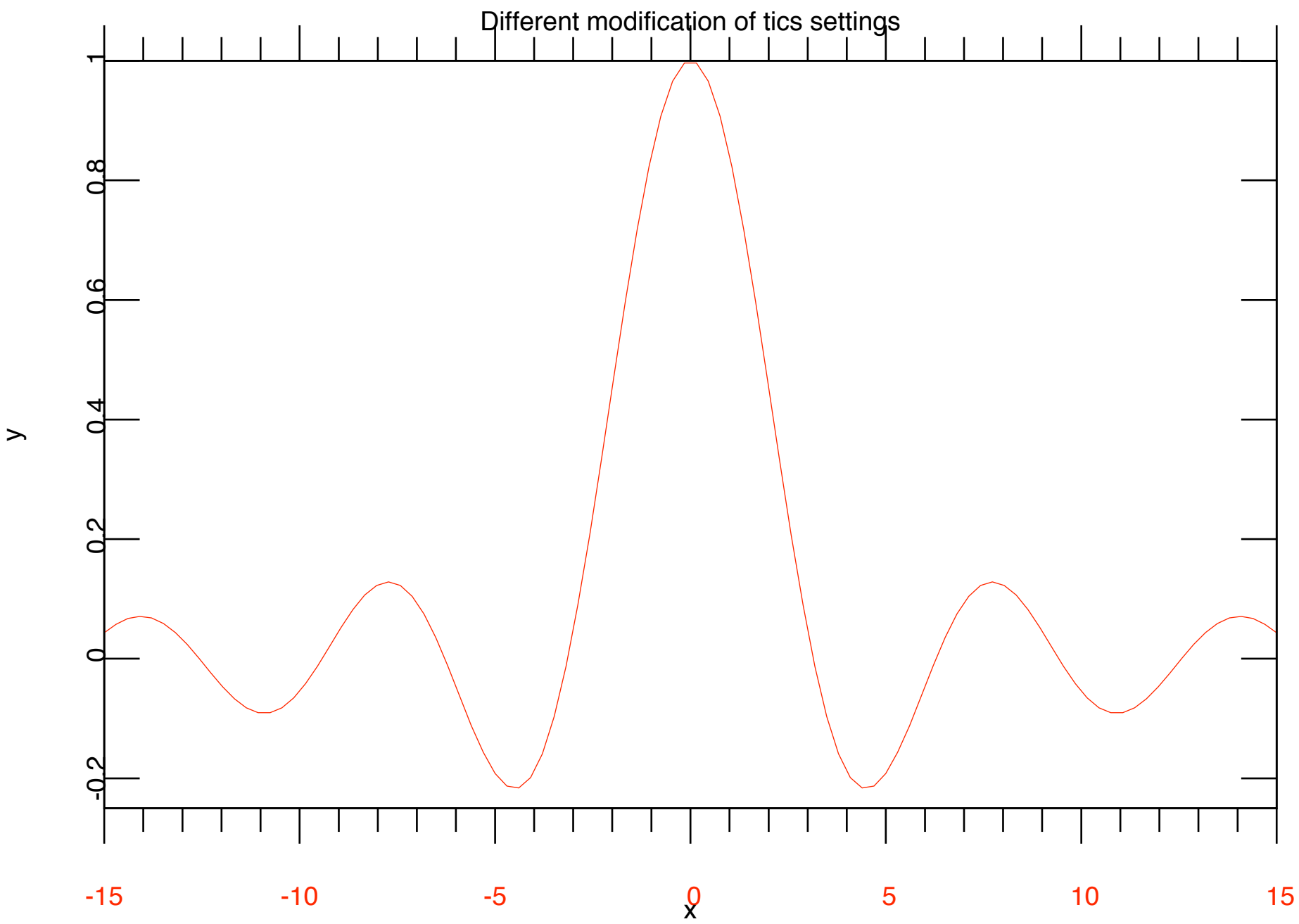

10
5
0
-5
-10
-10
-5
0
5
10
-q
+q
3
2
1
0.5
0.2
0.1
0.05
0.02
0
-0.02
-0.05
-0.1
-0.2
-0.5
-1
-2
-3

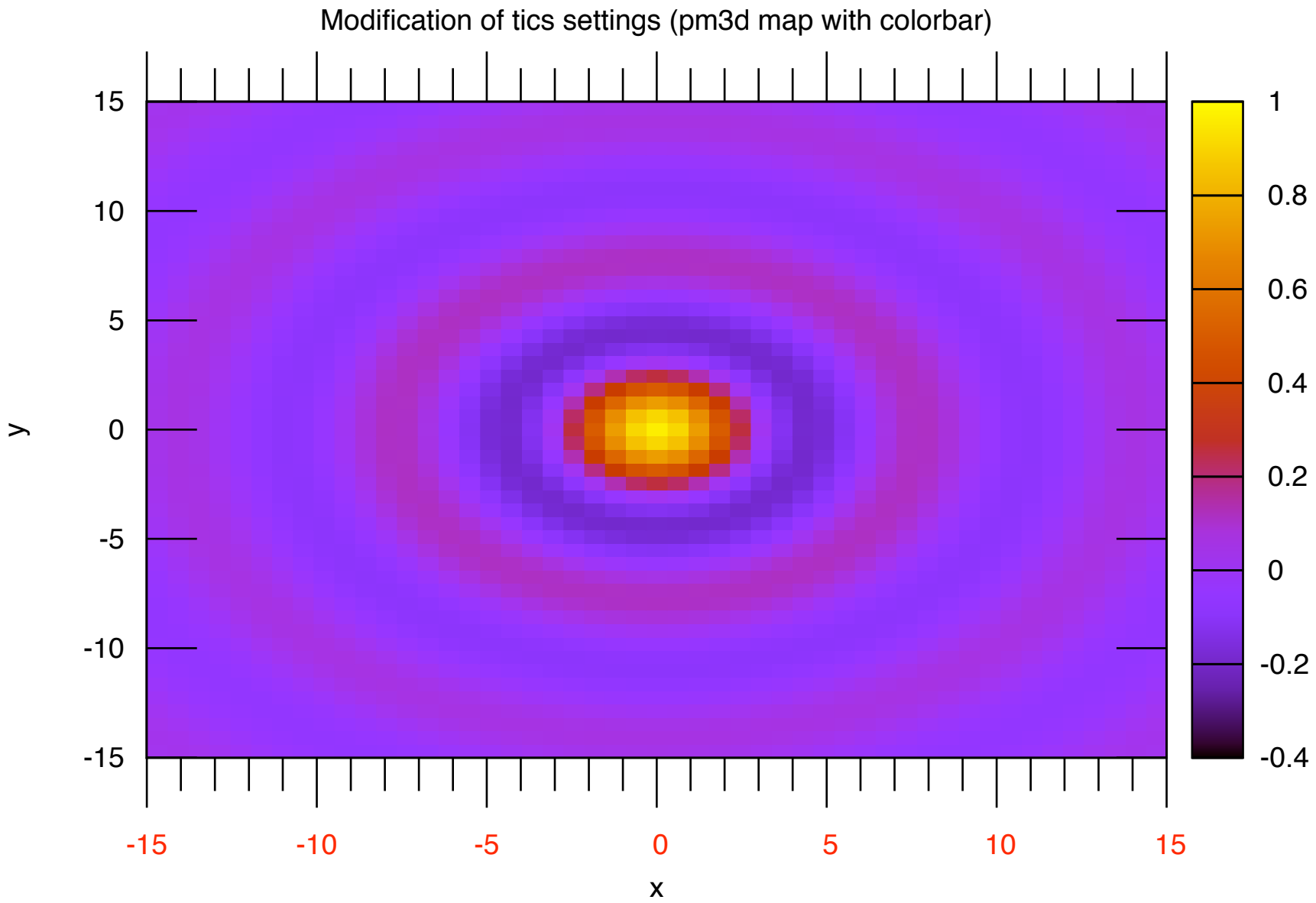

"equipo2.tmp"
10
5
0
-5
-10
-10
-5
0
5
10

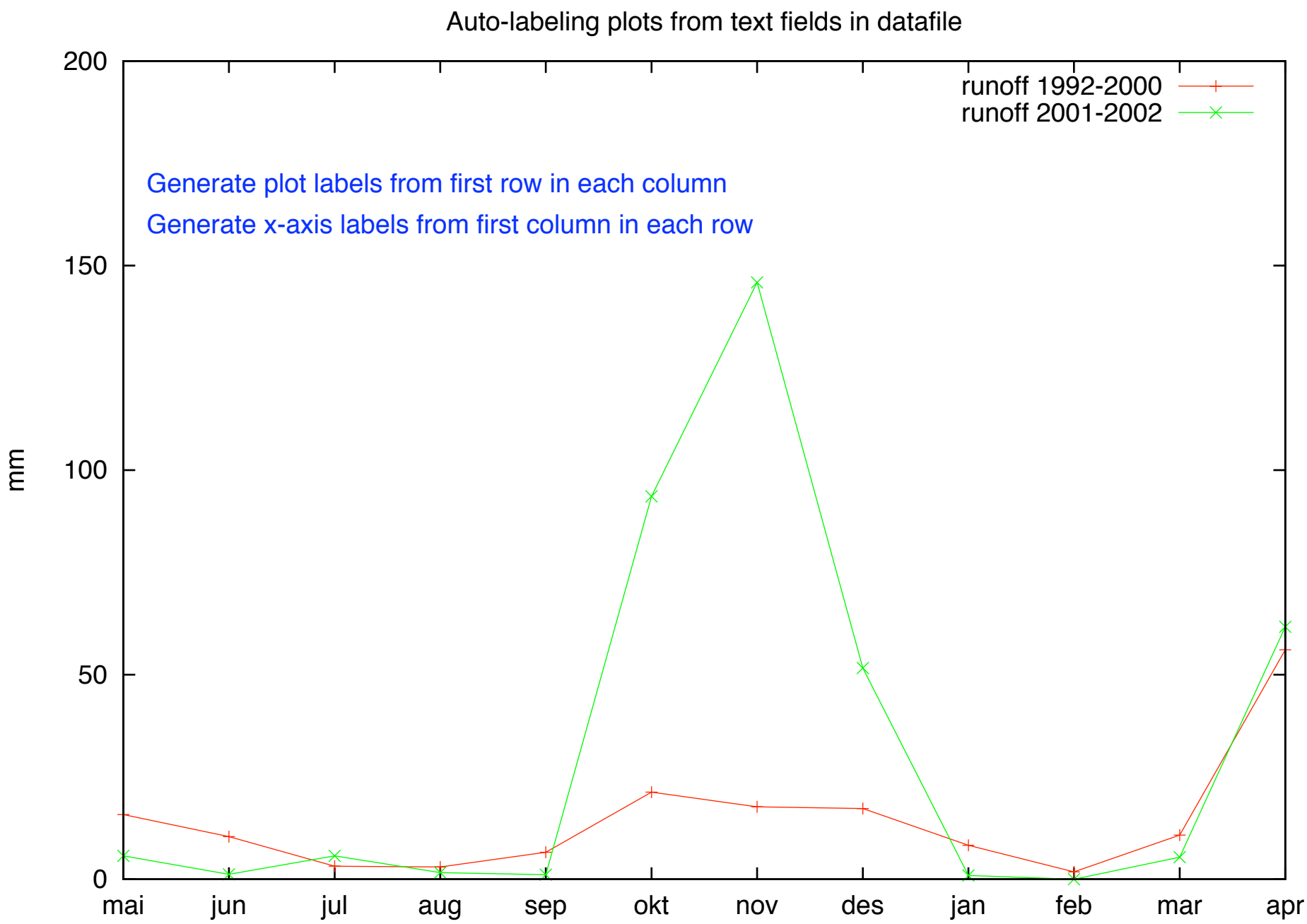

10
5
0
-5
-10
-10
-5
0
5
10
"field2xy.tmp" u 1:2:(coef*dx1($1,$2)):(coef*dy1($1,$2))
"equipo2.tmp"

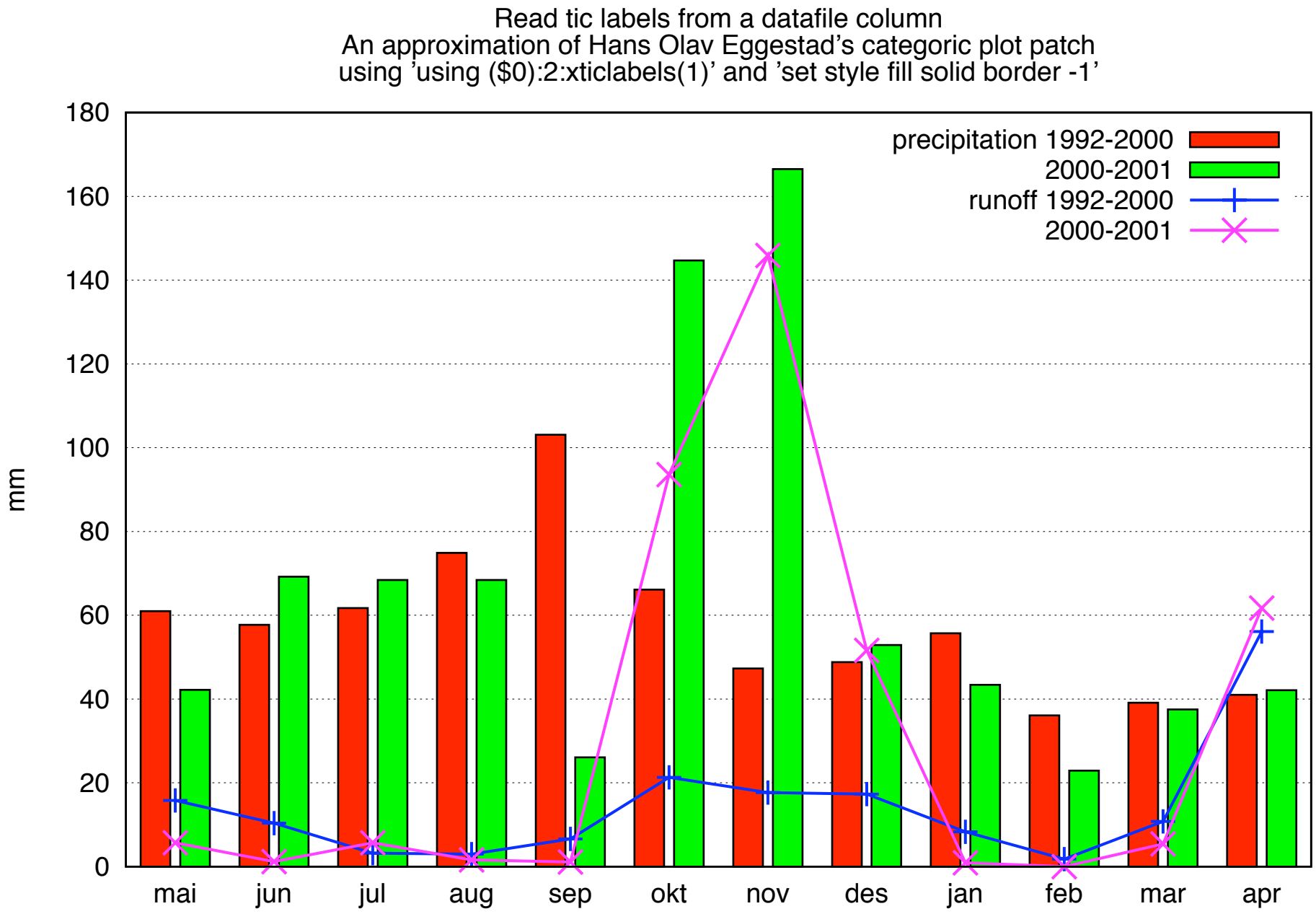

Default tics settings
1
0.8
0.6
0.4
0.2
0
-0.2
y
-15
-10
-5
0
5
10
15
x

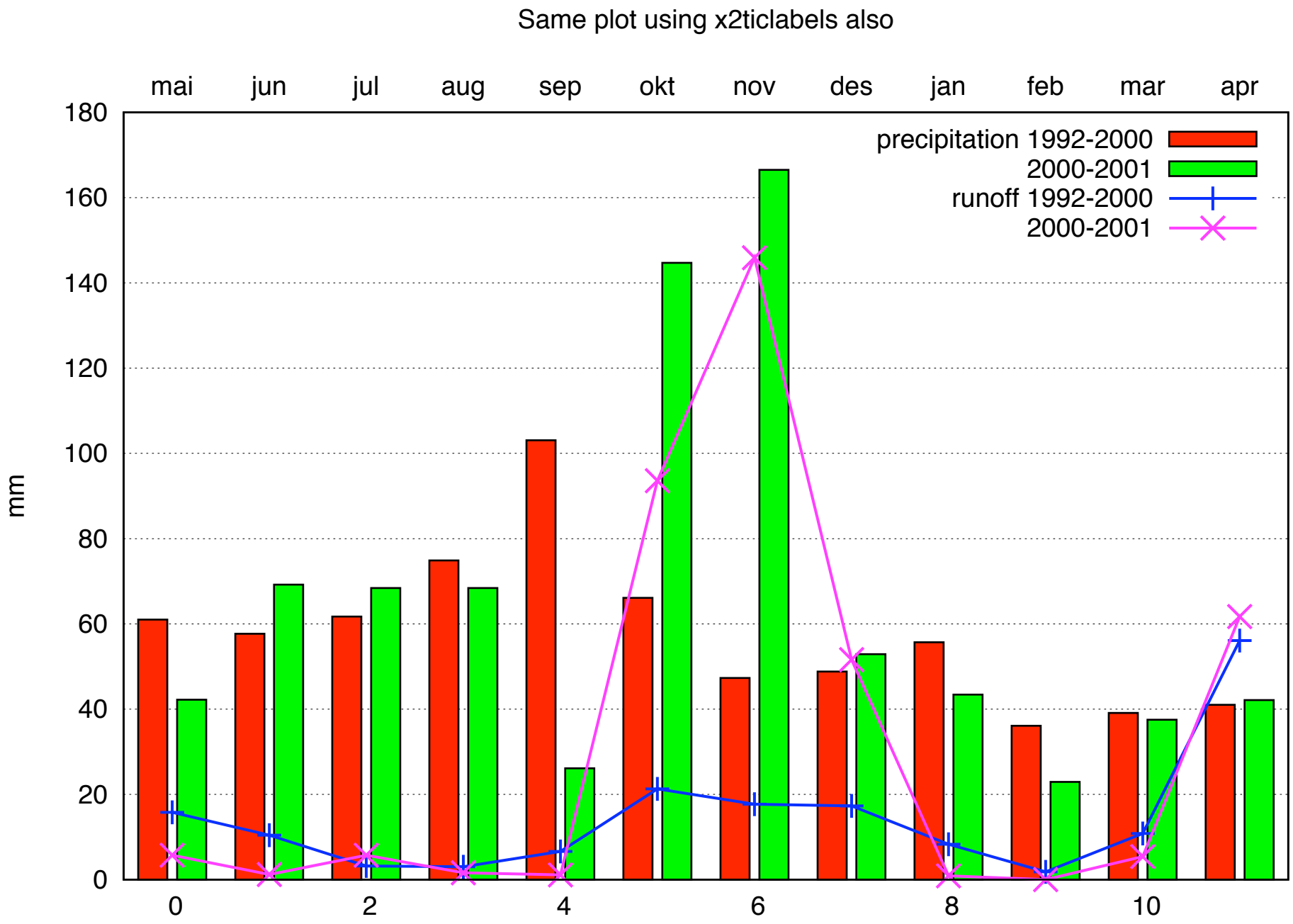

Different modification of tics settings
y
x
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15

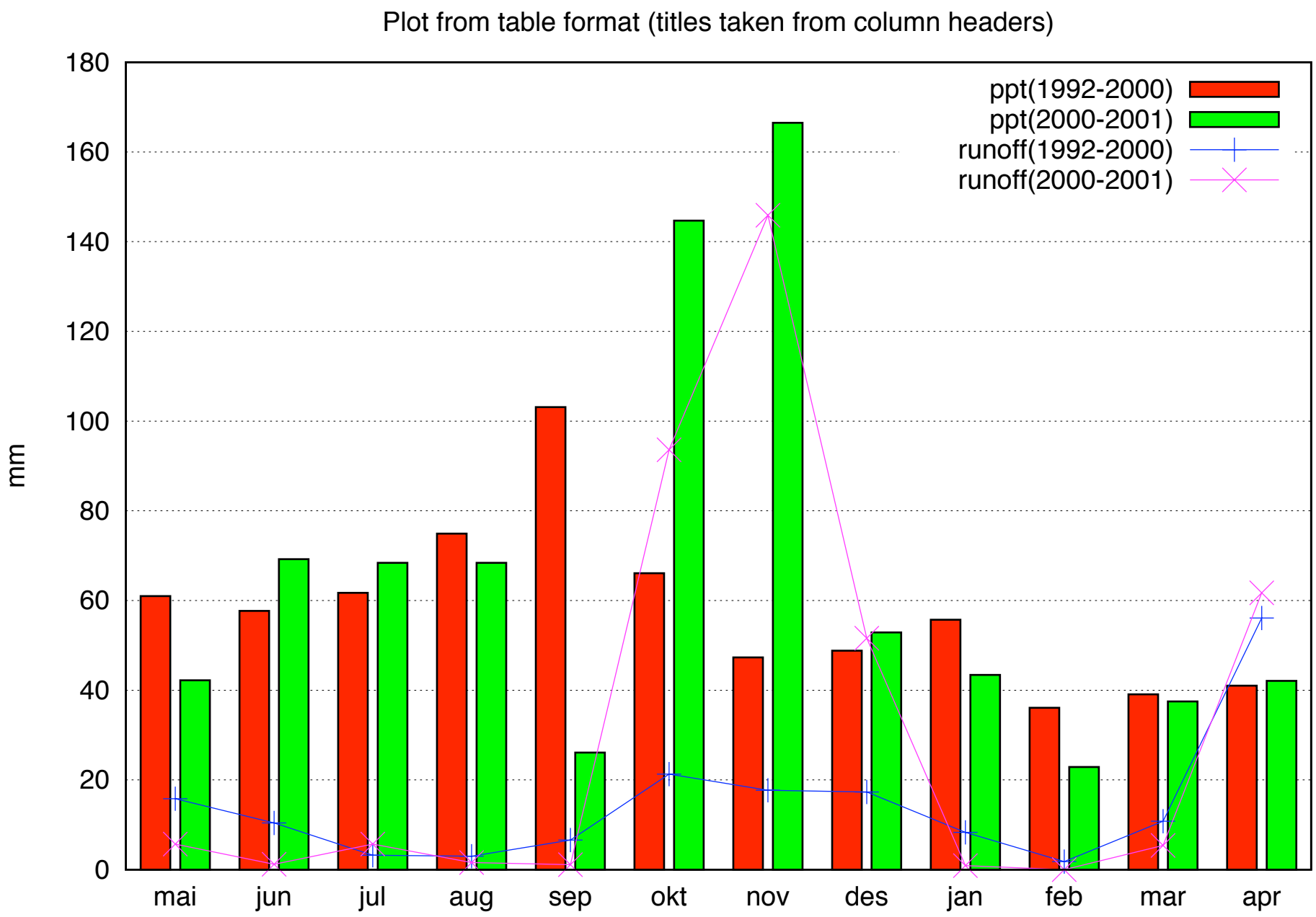

Different modification of tics settings
y
1
0.8
0.6
0.4
0.2
0
-0.2
-15
-10
-5
0
5
10
15
x

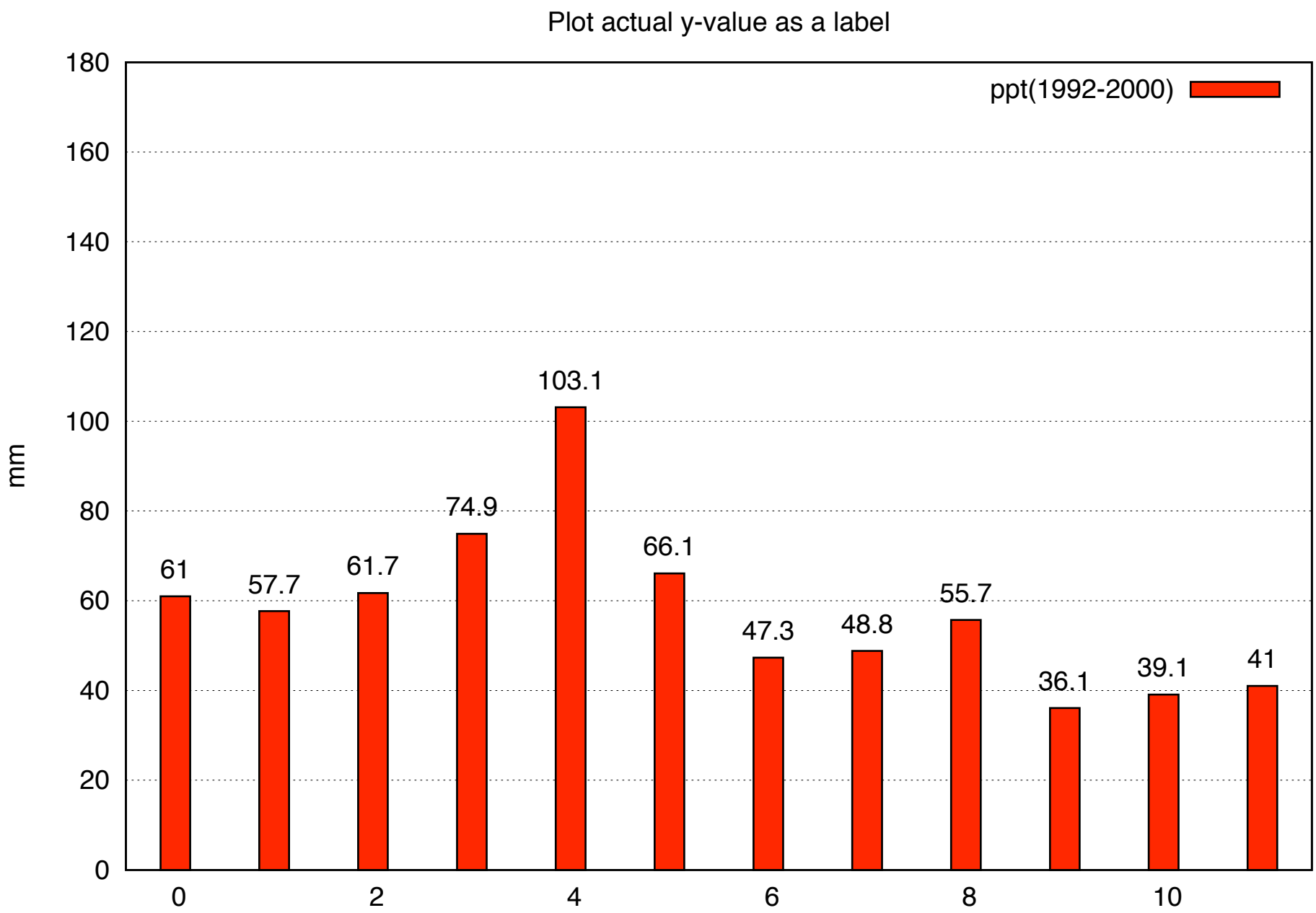

Modification of tics settings (pm3d map with colorbar)
15
10
5
0
-5
-10
-15
y
-15
-10
-5
0
5
10
15
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4


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

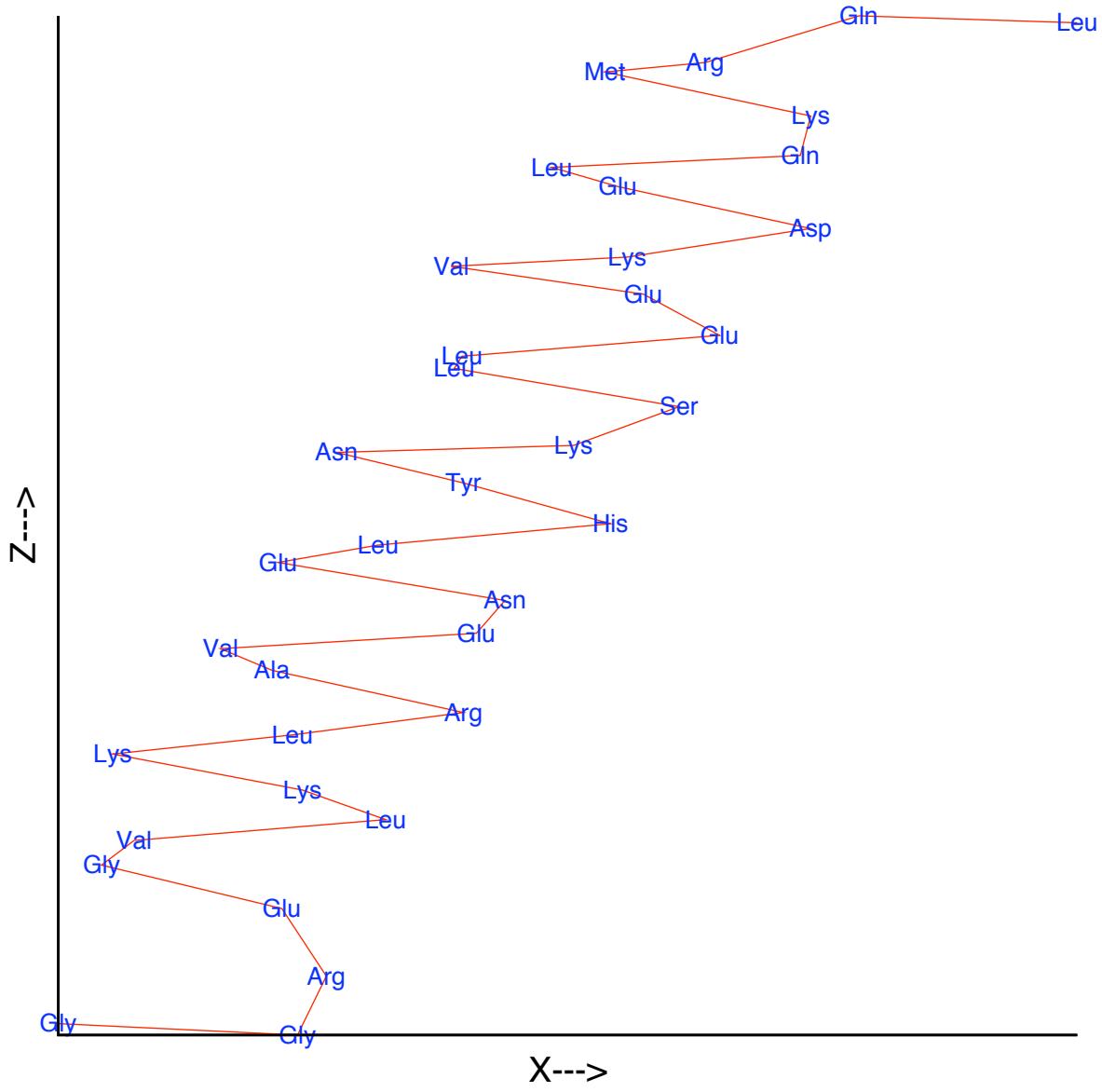

Read tic labels from a datafile column
An approximation of Hans Olav Eggestad's categoric plot patch
using 'using ($0):2:xticlabels(1)' and 'set style fill solid border -1'
precipitation 1992-2000
2000-2001
runoff 1992-2000
2000-2001
mm
180
160
140
120
100
80
60
40
20
0
mai
jun
jul
aug
sep
okt
nov
des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

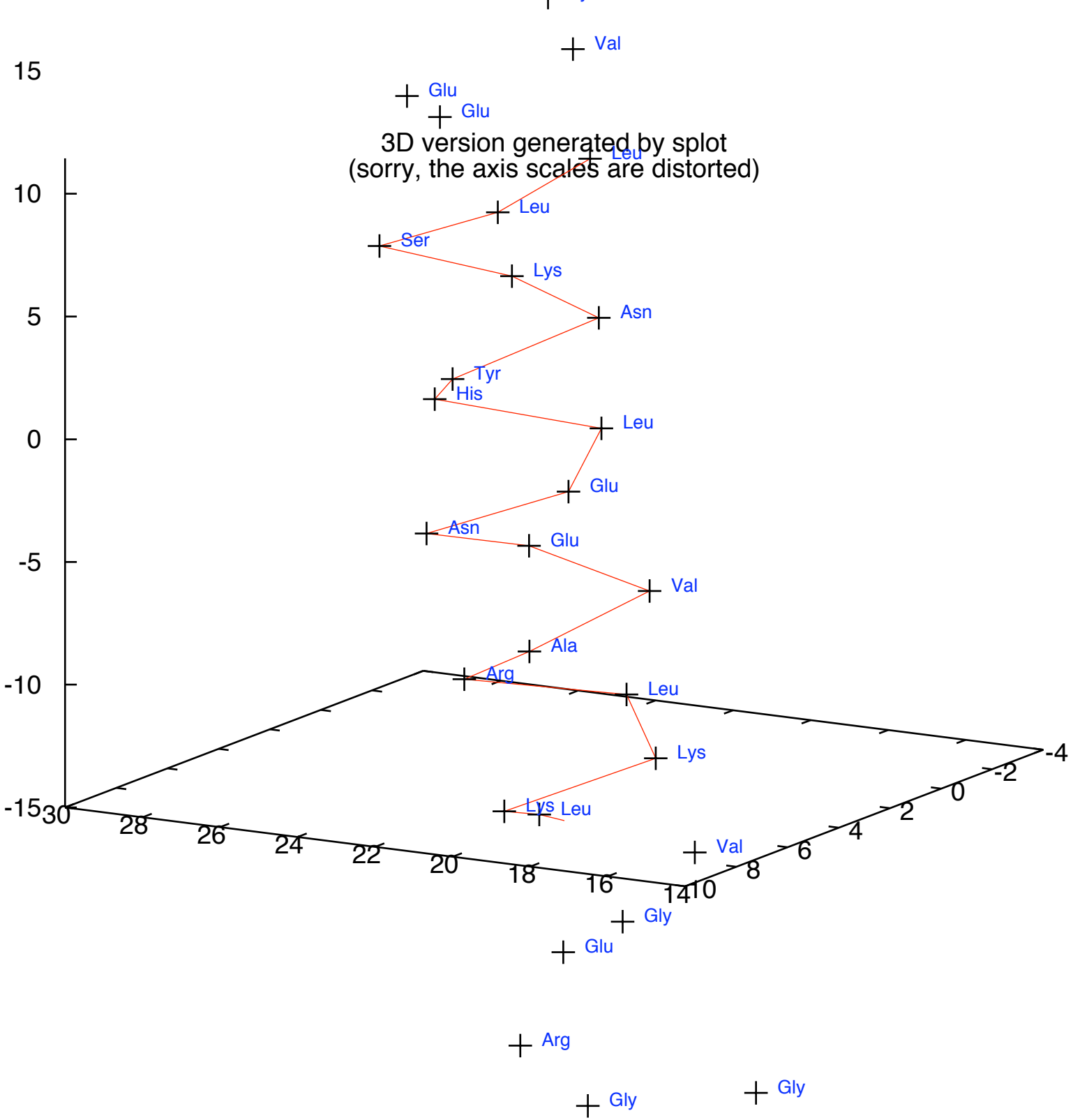

Same plot using x2ticlabels also
mai
jun
jul
aug
sep
okt
nov
des
jan
feb
mar
apr
180
160
140
120
100
80
60
40
20
0
mm
0
2
4
6
8
10
precipitation 1992-2000
2000-2001
runoff 1992-2000
2000-2001

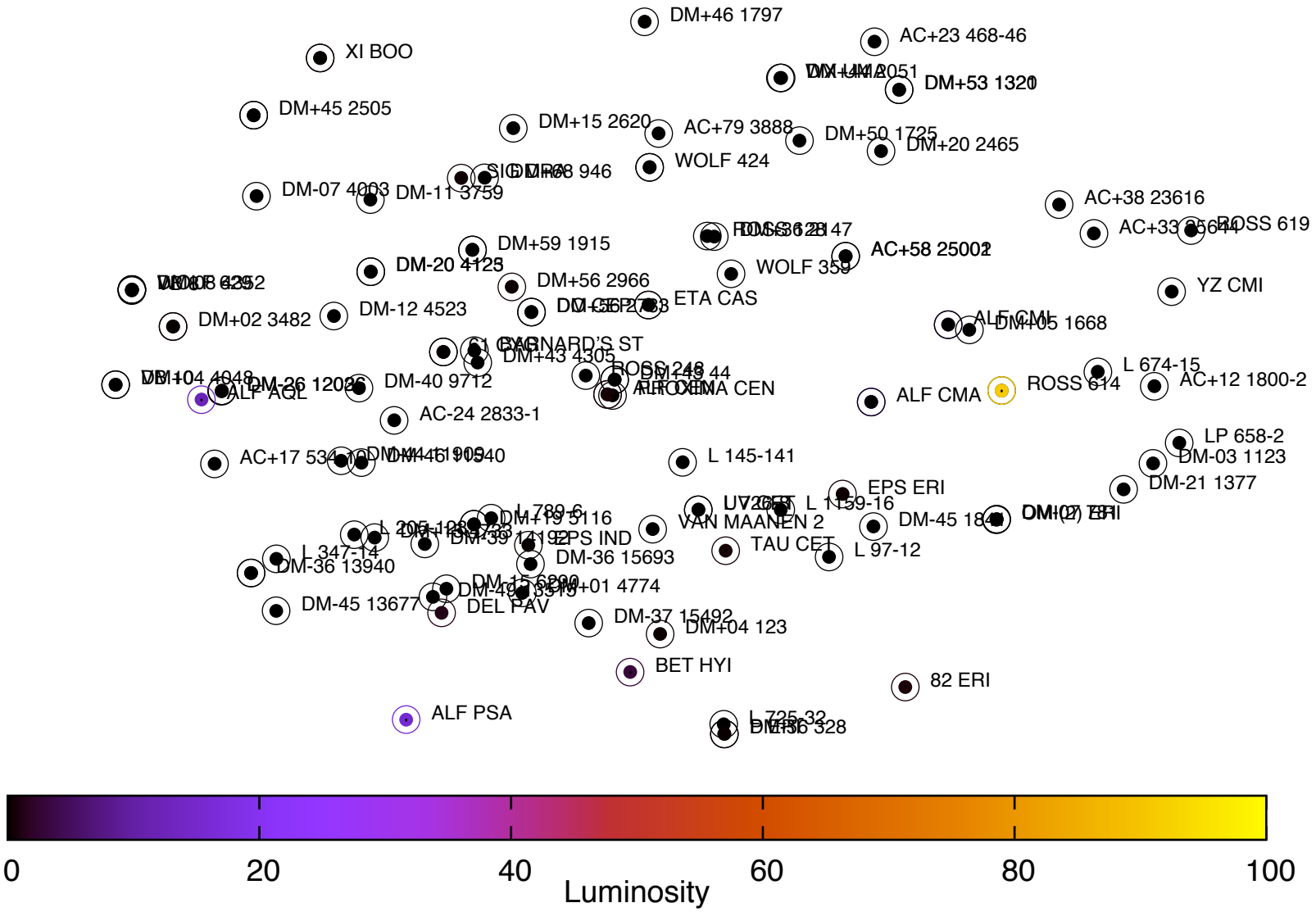

Plot from table format (titles taken from column headers)
mm
180
160
140
120
100
80
60
40
20
0
mai
jun
jul
aug
sep
okt
nov
des
jan
feb
mar
apr
ppt(1992-2000)
ppt(2000-2001)
runoff(1992-2000)
runoff(2000-2001)

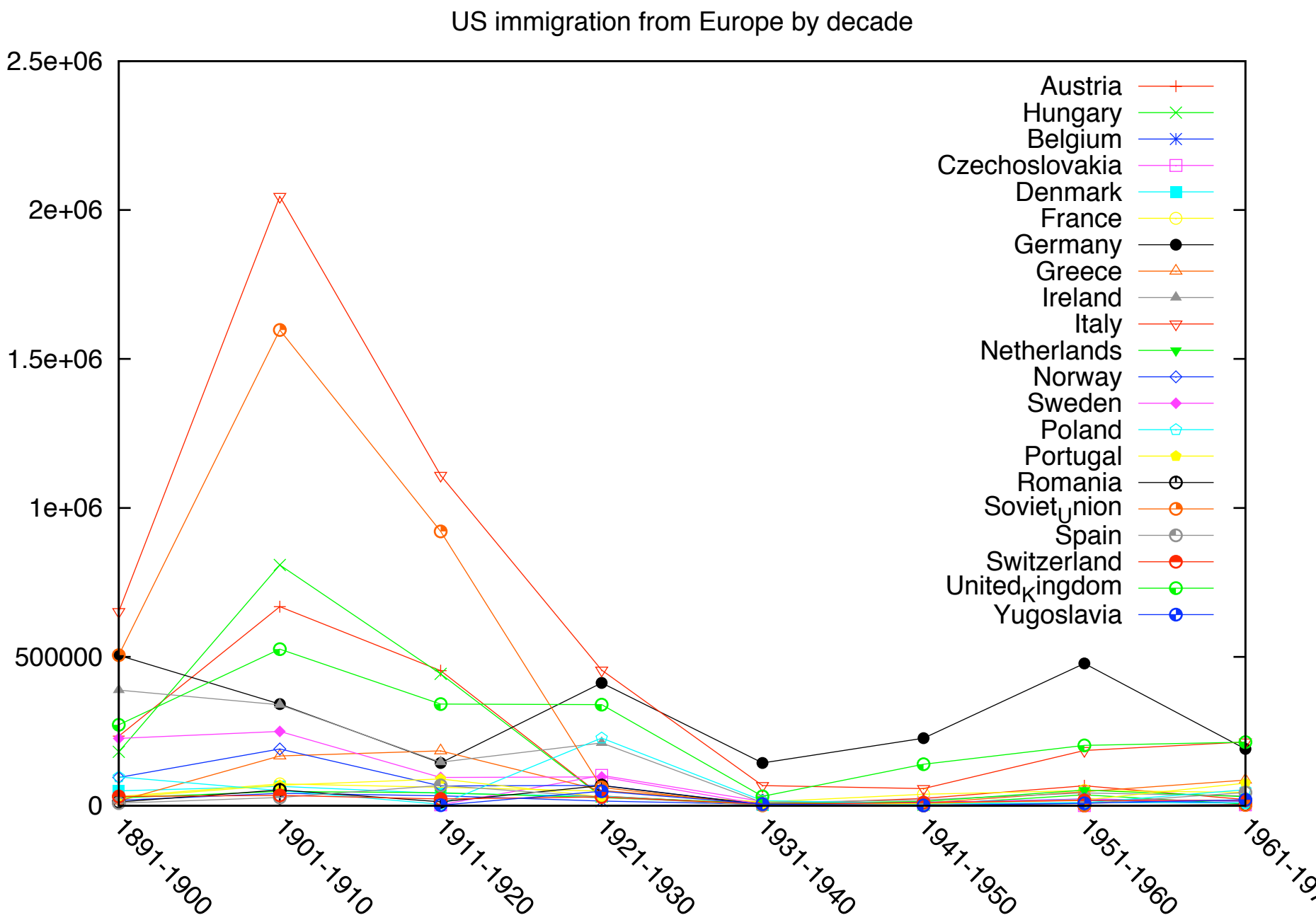

Plot actual y-value as a label
ppt(1992-2000)
mm
180
160
140
120
100
80
60
40
20
0
0
2
4
6
8
10
61
57.7
61.7
74.9
103.1
66.1
47.3
48.8
55.7
36.1
39.1
41

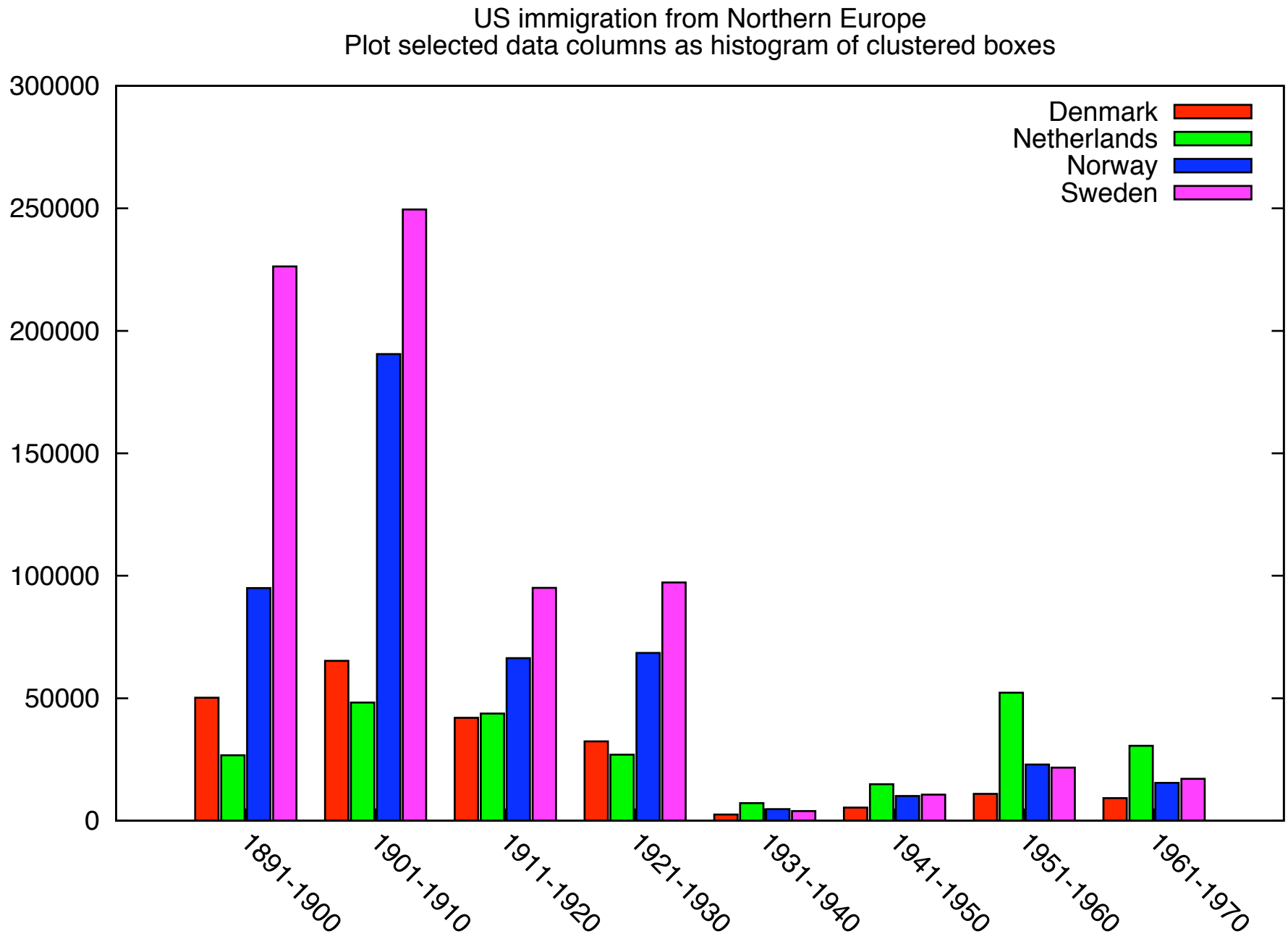
Read labels from a datafile column
Here the 'plot with labels' command generates a
C-alpha trace of retro-GCN4 peptide

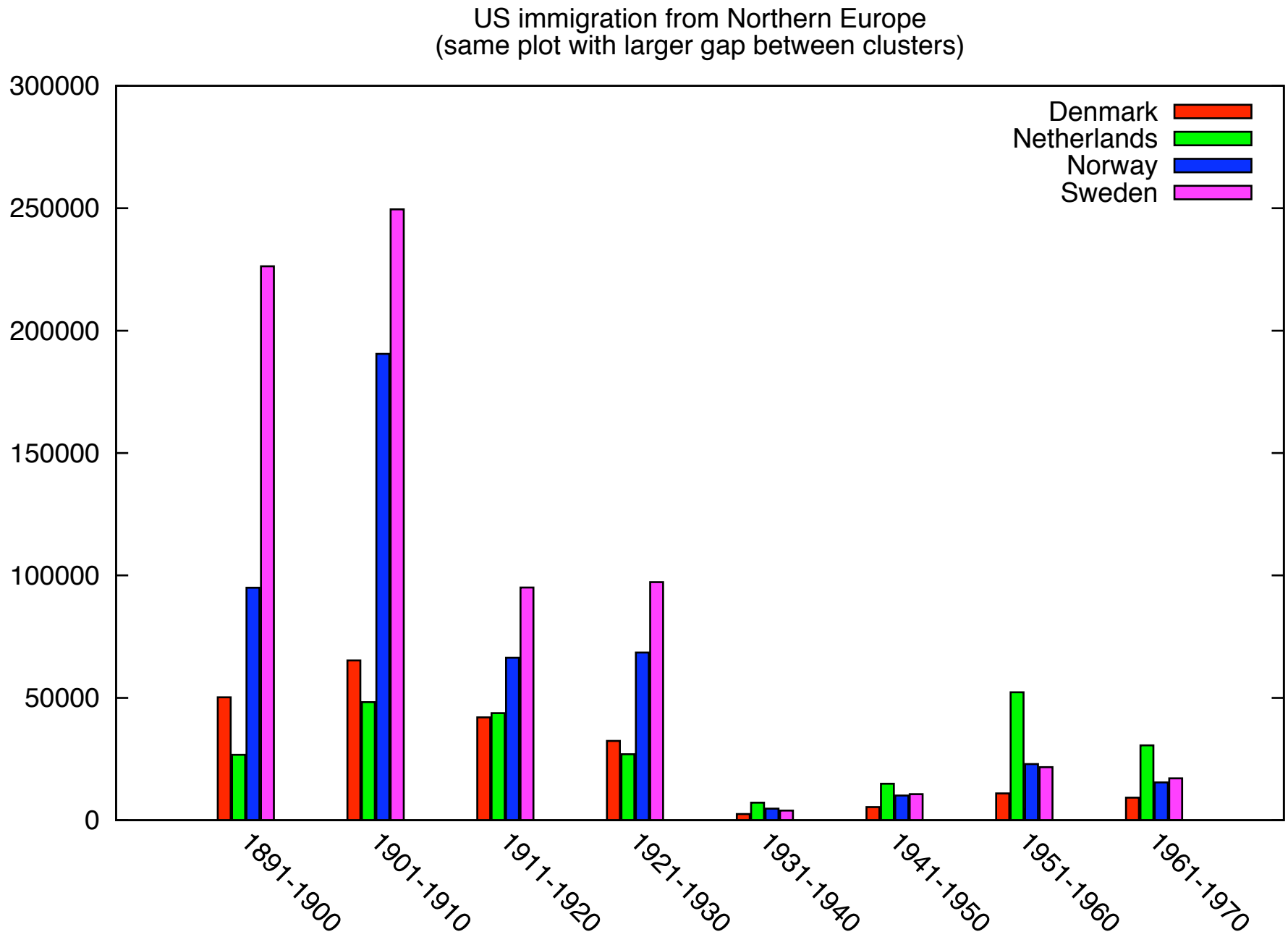


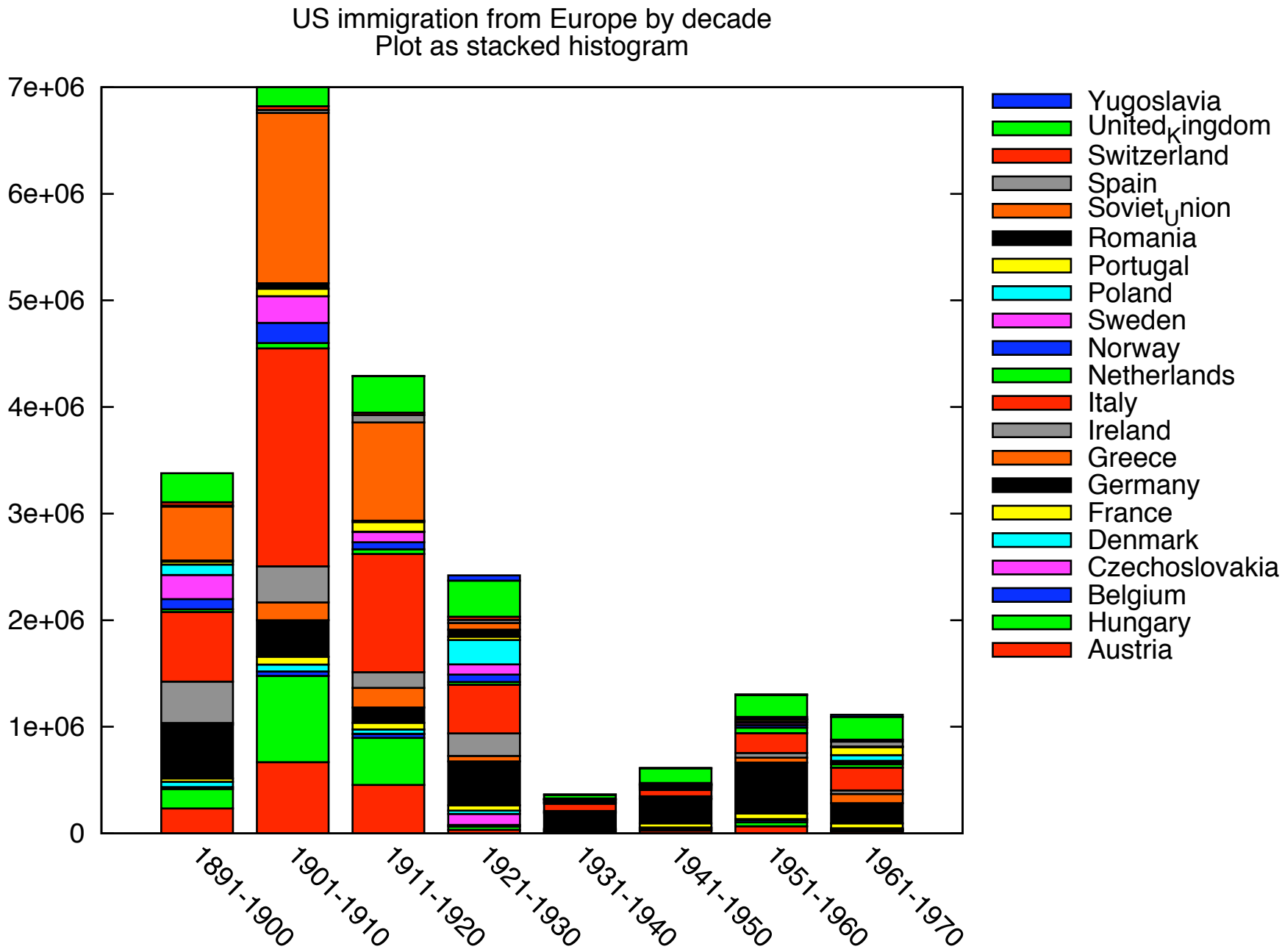

3D version generated by splot
(sorry, the axis scales are distorted)
15
10
5
0
-5
-10
-15
30
28
26
24
22
20
18
16
14
-4
-2
0
2
4
6
8
10
Val
Glu
Glu
Leu
Leu
Ser
Lys
Asn
Tyr
His
Leu
Glu
Asn
Glu
Val
Ala
Arg
Leu
Lys
Lys
Leu
Val
Gly
Glu
Arg
Gly
Gly

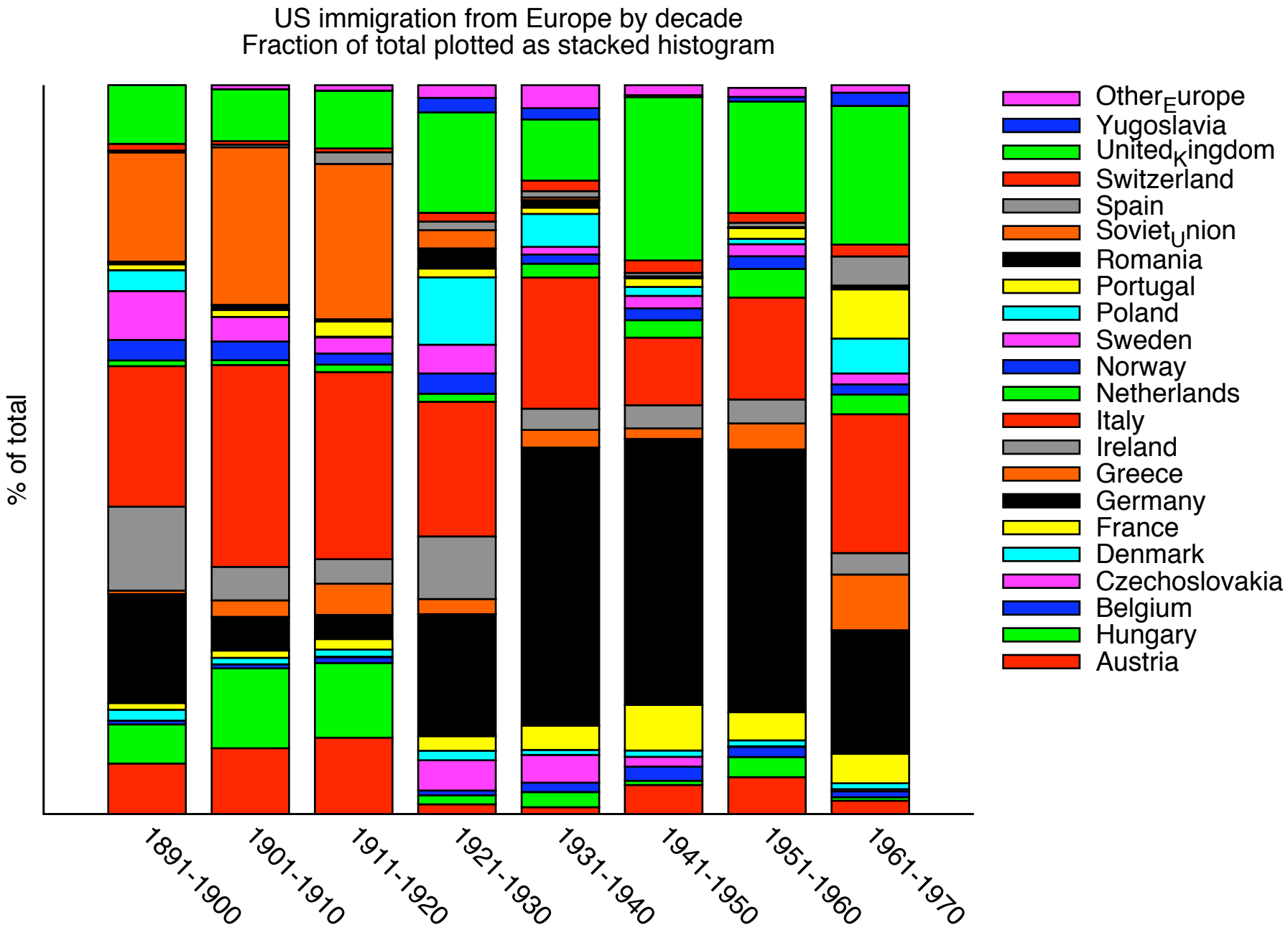

Gliese star catalog - 7 parsec neighborhood centered on Earth
DM+46 1797
AC+23 468-46
XI BOO
DM+53 1320
DM+45 2505
DM+15 2620
AC+79 3888
DM+50 1725
DM+20 2465
WOLF 424
DM-07 4003
DM-11 3759
AC+38 23616
ROSS 619
DM+59 1915
AC+58 25002
WOLF 359
DM-20 4125
DM+56 2966
YZ CMI
ETA CAS
DM-12 4523
DM+02 3482
DM+05 1668
BARNARD'S ST
DM+43 4305
L 674-15
DM-40 9712
ROSS 614
AC+12 1800-2
ALF AQL
ALF CMA
AC-24 2833-1
LP 658-2
DM-03 1123
L 145-141
DM-21 1377
EPS ERI
L 1159-16
VAN MAANEN 2
DM-45 184
EPS IND
TAU CET
L 97-12
L 347-14
DM-36 15693
DM-36 13940
DM+01 4774
DM-45 13677
DEL PAV
DM-37 15492
DM+04 123
BET HYI
82 ERI
ALF PSA
0
20
40
60
80
100
Luminosity

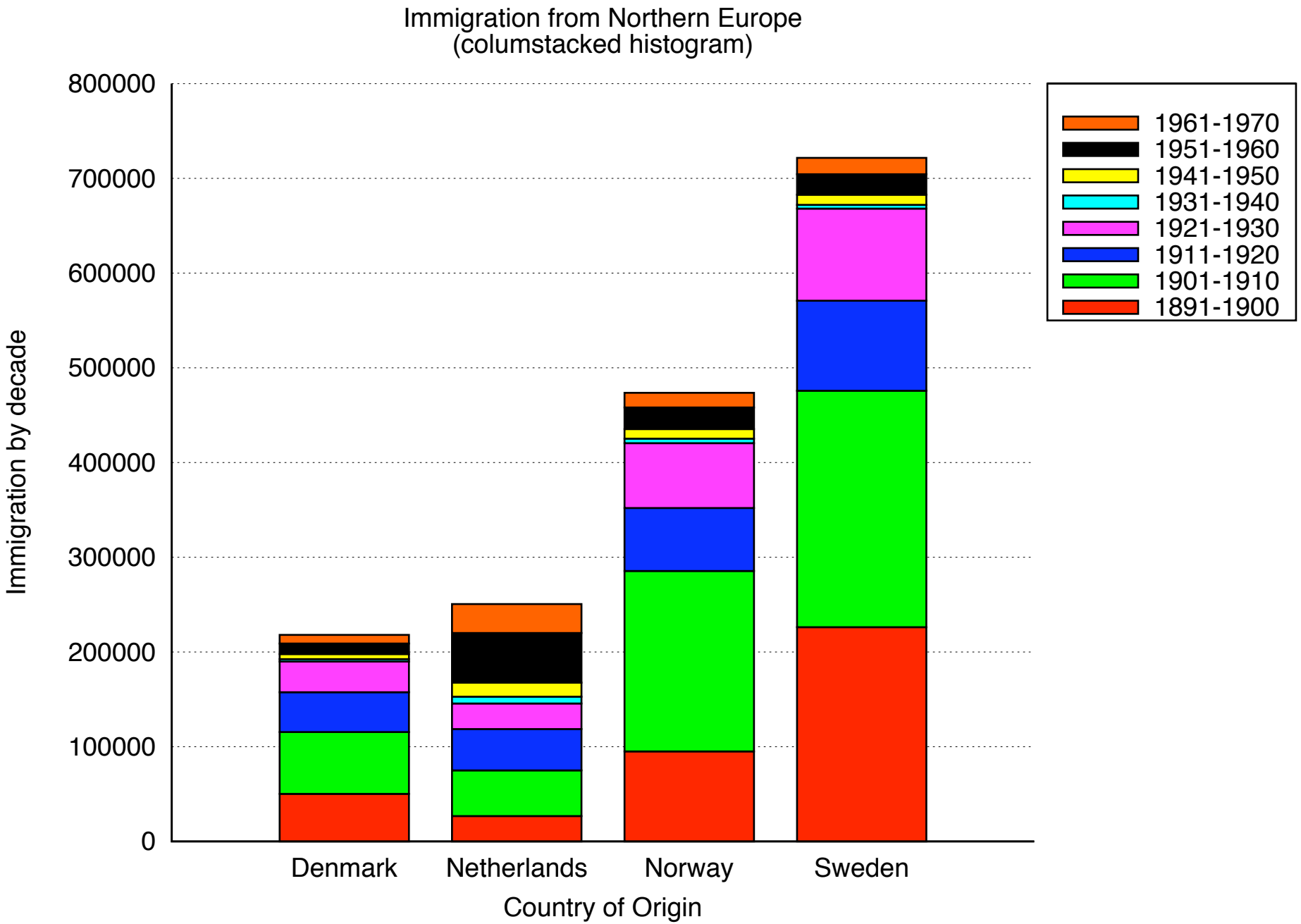

US immigration from Europe by decade
2.5e+06
2e+06
1.5e+06
1e+06
500000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
Austria
Hungary
Belgium
Czechoslovakia
Denmark
France
Germany
Greece
Ireland
Italy
Netherlands
Norway
Sweden
Poland
Portugal
Romania
SovietUnion
Spain
Switzerland
UnitedKingdom
Yugoslavia

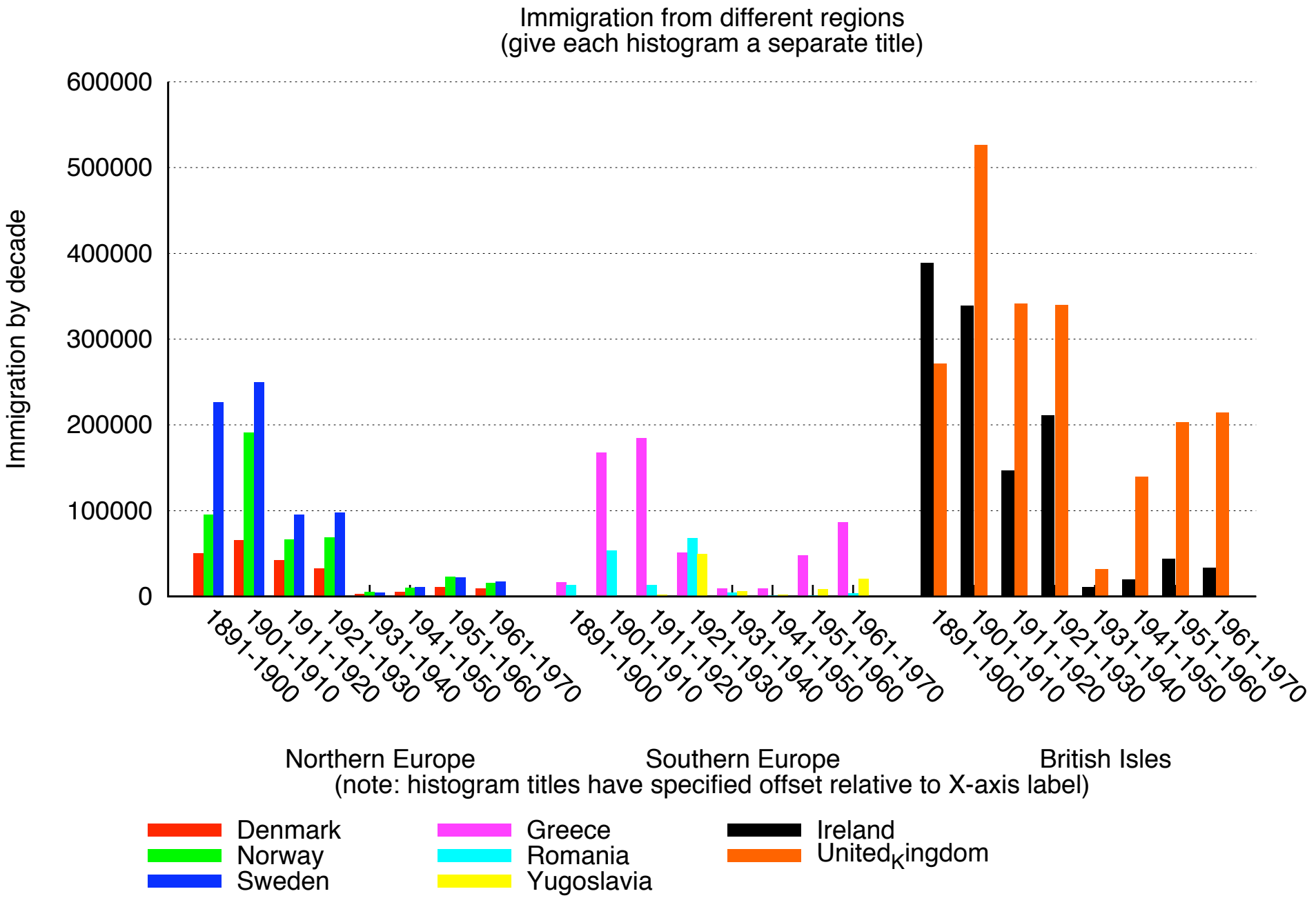

US immigration from Northern Europe
Plot selected data columns as histogram of clustered boxes
300000
250000
200000
150000
100000
50000
0
Denmark
Netherlands
Norway
Sweden
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970

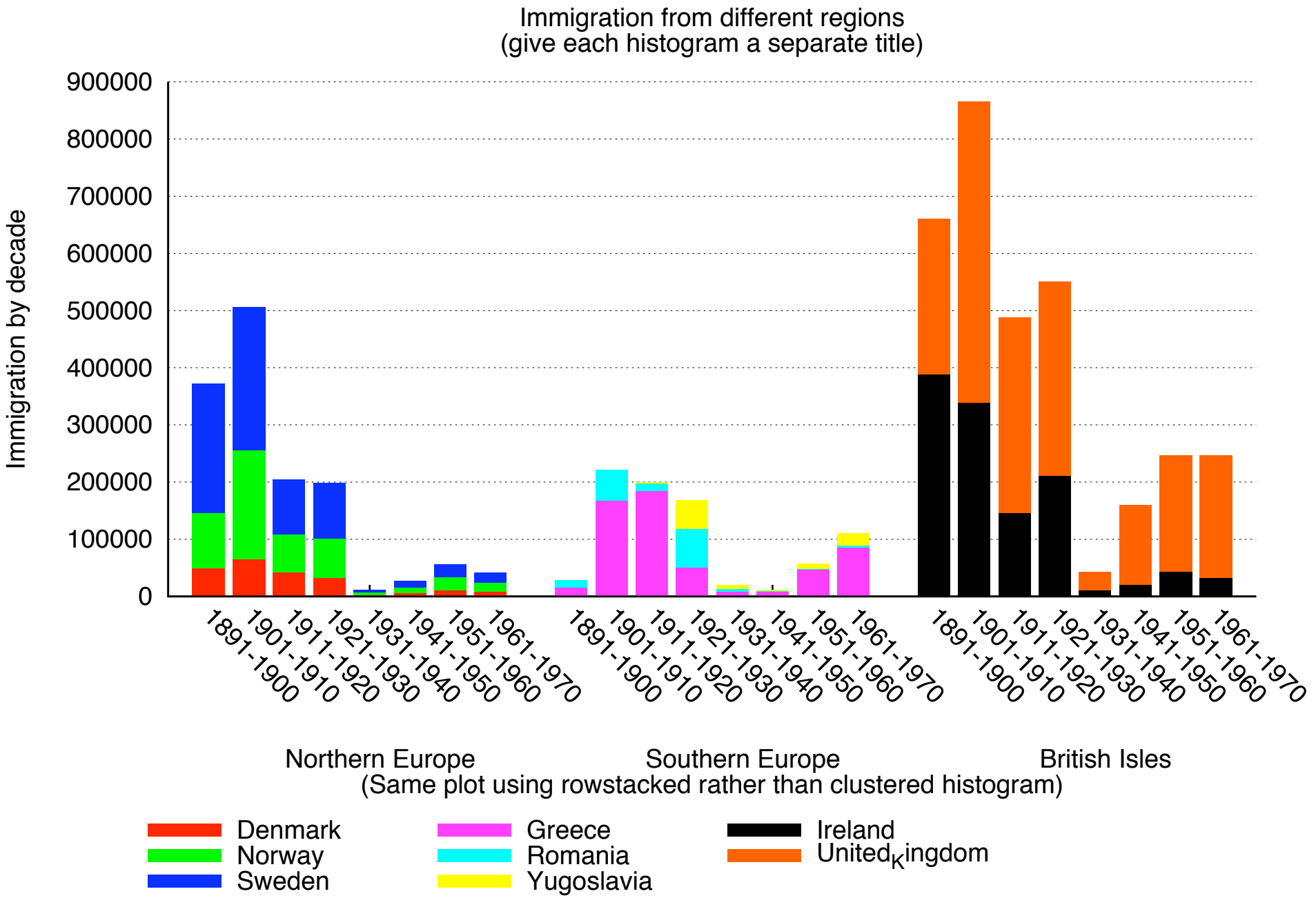

US immigration from Northern Europe
(same plot with larger gap between clusters)
300000
250000
200000
150000
100000
50000
0
Denmark
Netherlands
Norway
Sweden
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970

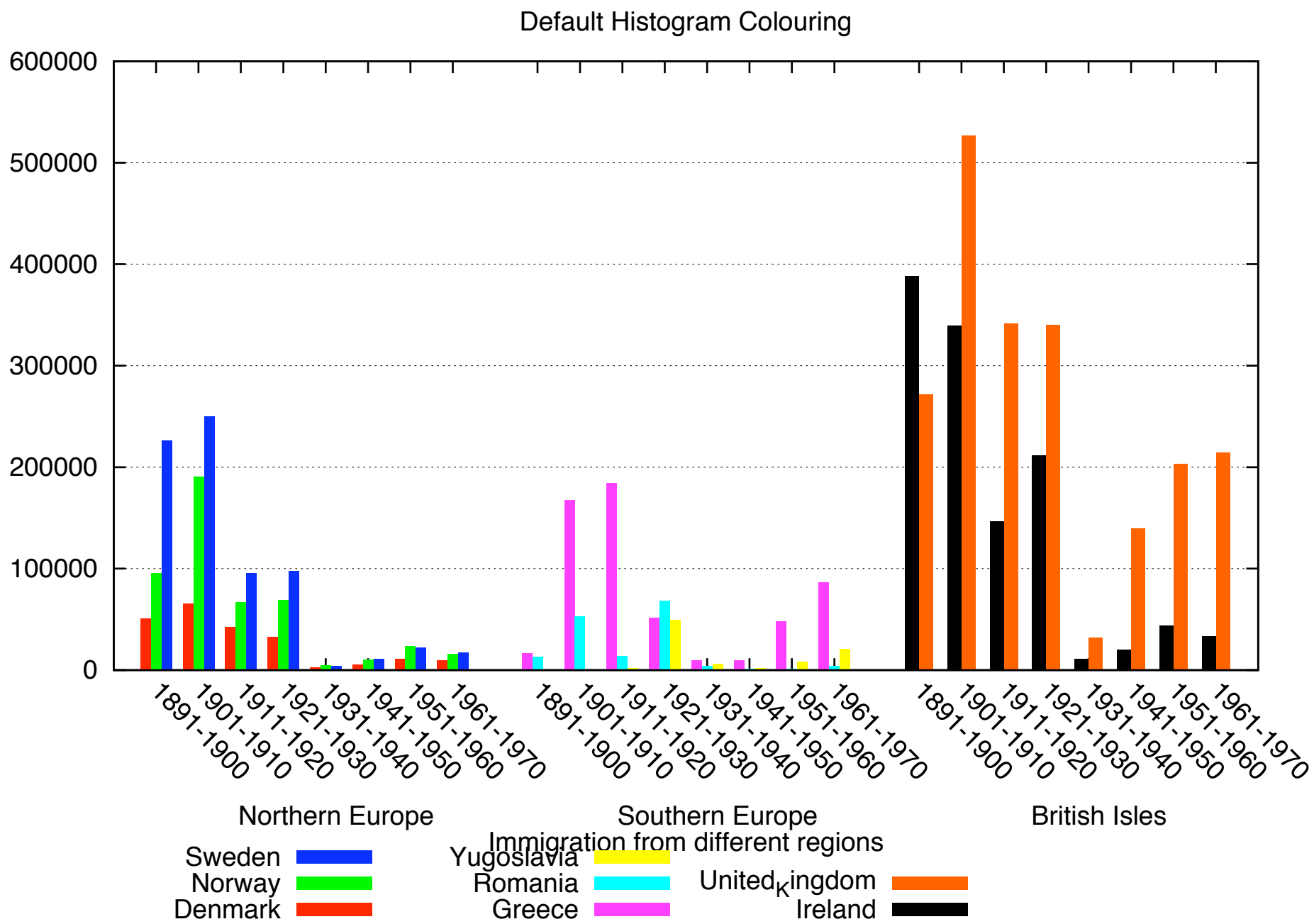

US immigration from Europe by decade
Plot as stacked histogram
7e+06
6e+06
5e+06
4e+06
3e+06
2e+06
1e+06
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Yugoslavia
United_Kingdom
Switzerland
Spain
Soviet_Union
Romania
Portugal
Poland
Sweden
Norway
Netherlands
Italy
Ireland
Greece
Germany
France
Denmark
Czechoslovakia
Belgium
Hungary
Austria

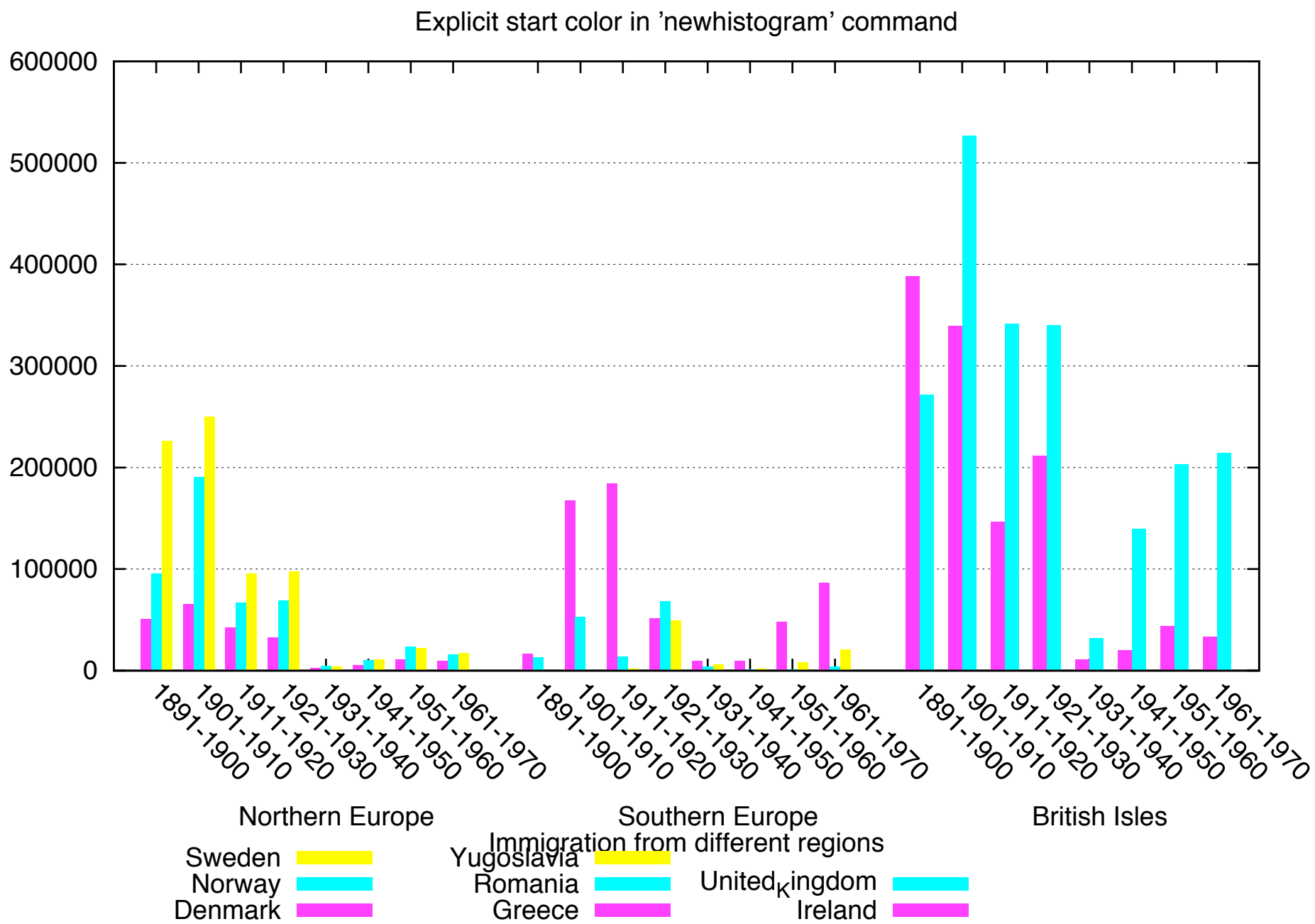

US immigration from Europe by decade
Fraction of total plotted as stacked histogram
% of total
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Other_Europe
Yugoslavia
United_Kingdom
Switzerland
Spain
Soviet_Union
Romania
Portugal
Poland
Sweden
Norway
Netherlands
Italy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

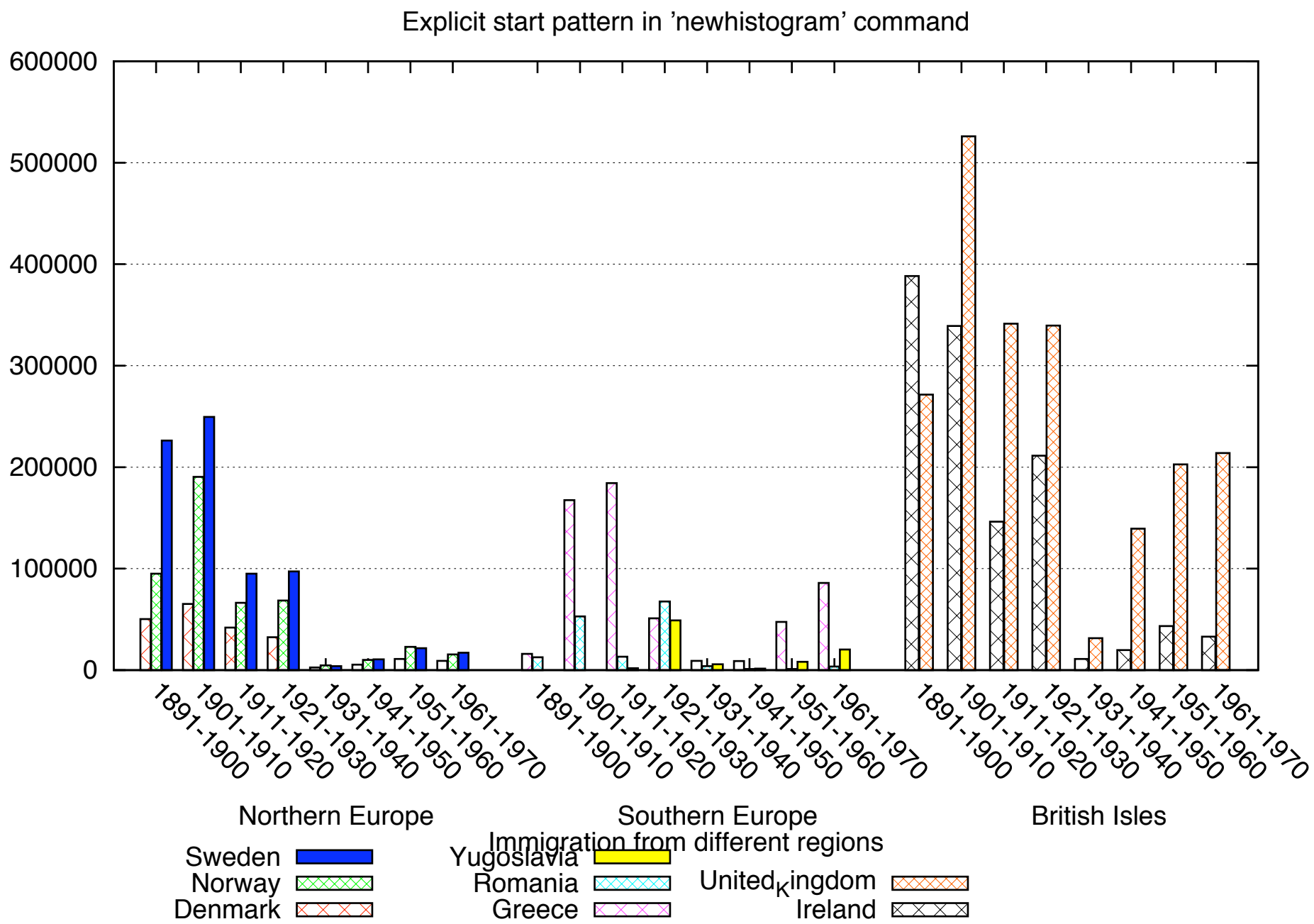

Immigration from Northern Europe
(columstacked histogram)
800000
700000
600000
500000
400000
300000
200000
100000
0
Immigration by decade
Denmark
Netherlands
Norway
Sweden
Country of Origin
1961-1970
1951-1960
1941-1950
1931-1940
1921-1930
1911-1920
1901-1910
1891-1900

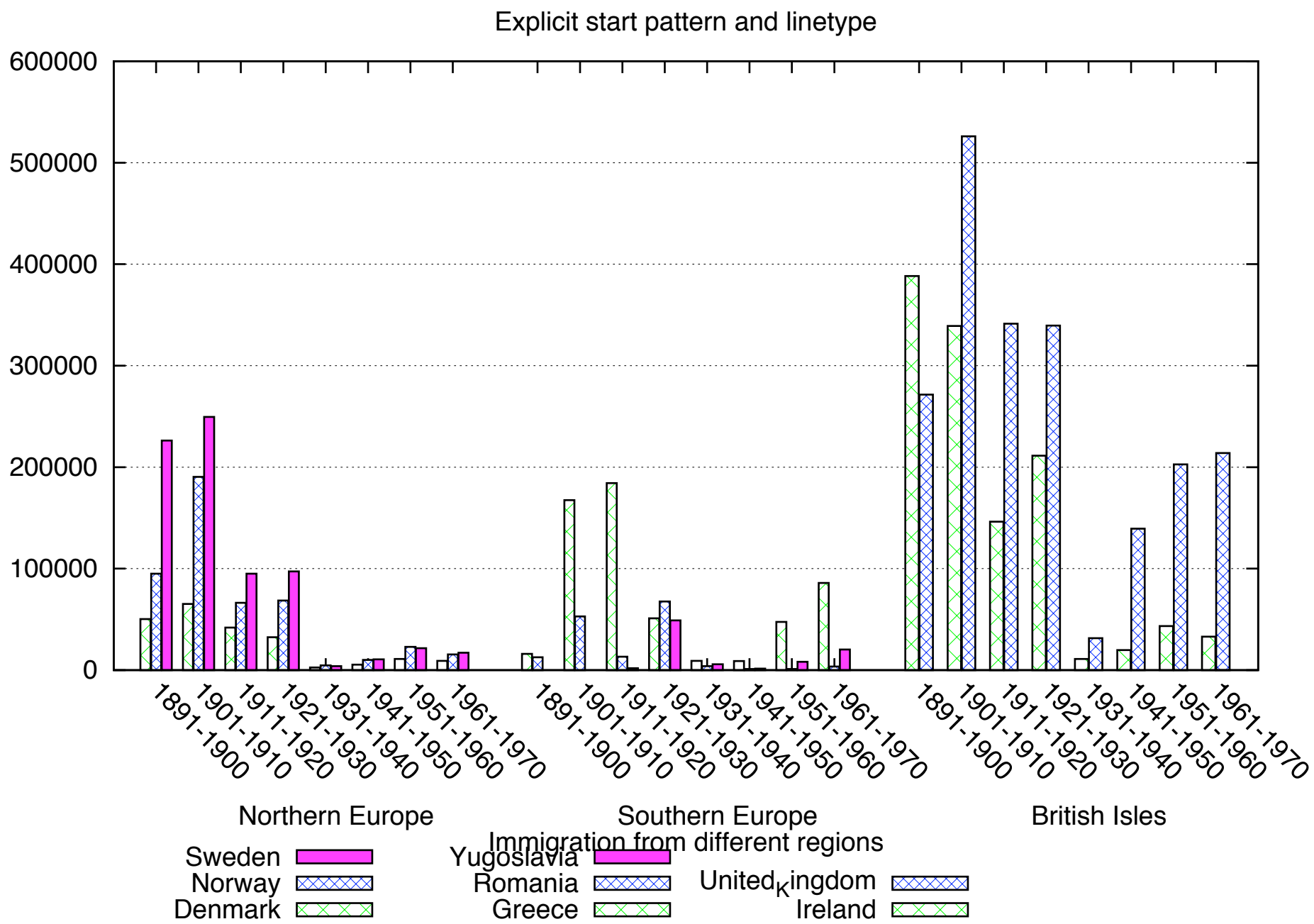

Immigration from different regions
(give each histogram a separate title)
Immigration by decade
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Northern Europe
Southern Europe
British Isles
(note: histogram titles have specified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

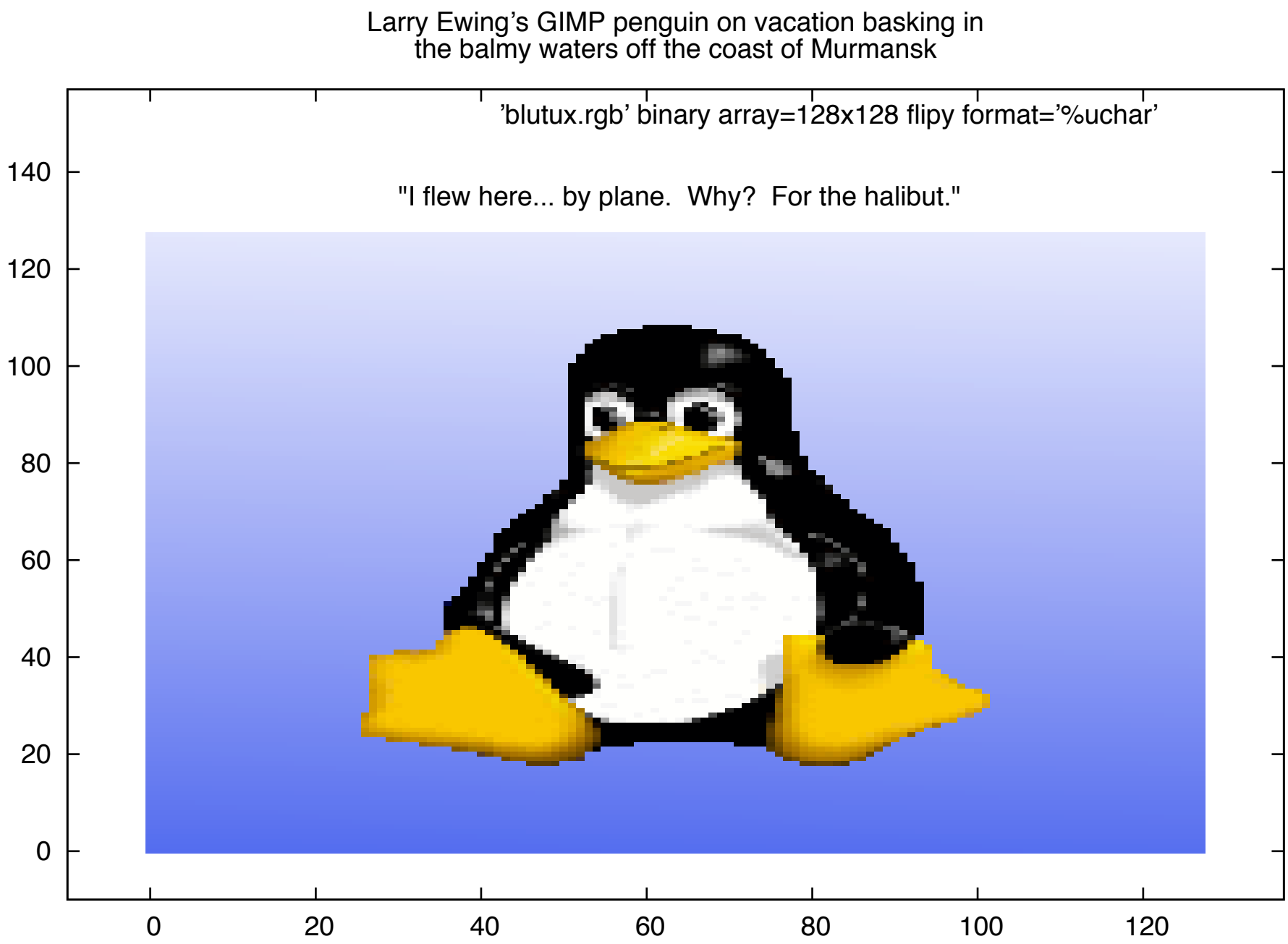

Immigration from different regions
(give each histogram a separate title)
Immigration by decade
900000
800000
700000
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Northern Europe
Southern Europe
British Isles
(Same plot using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

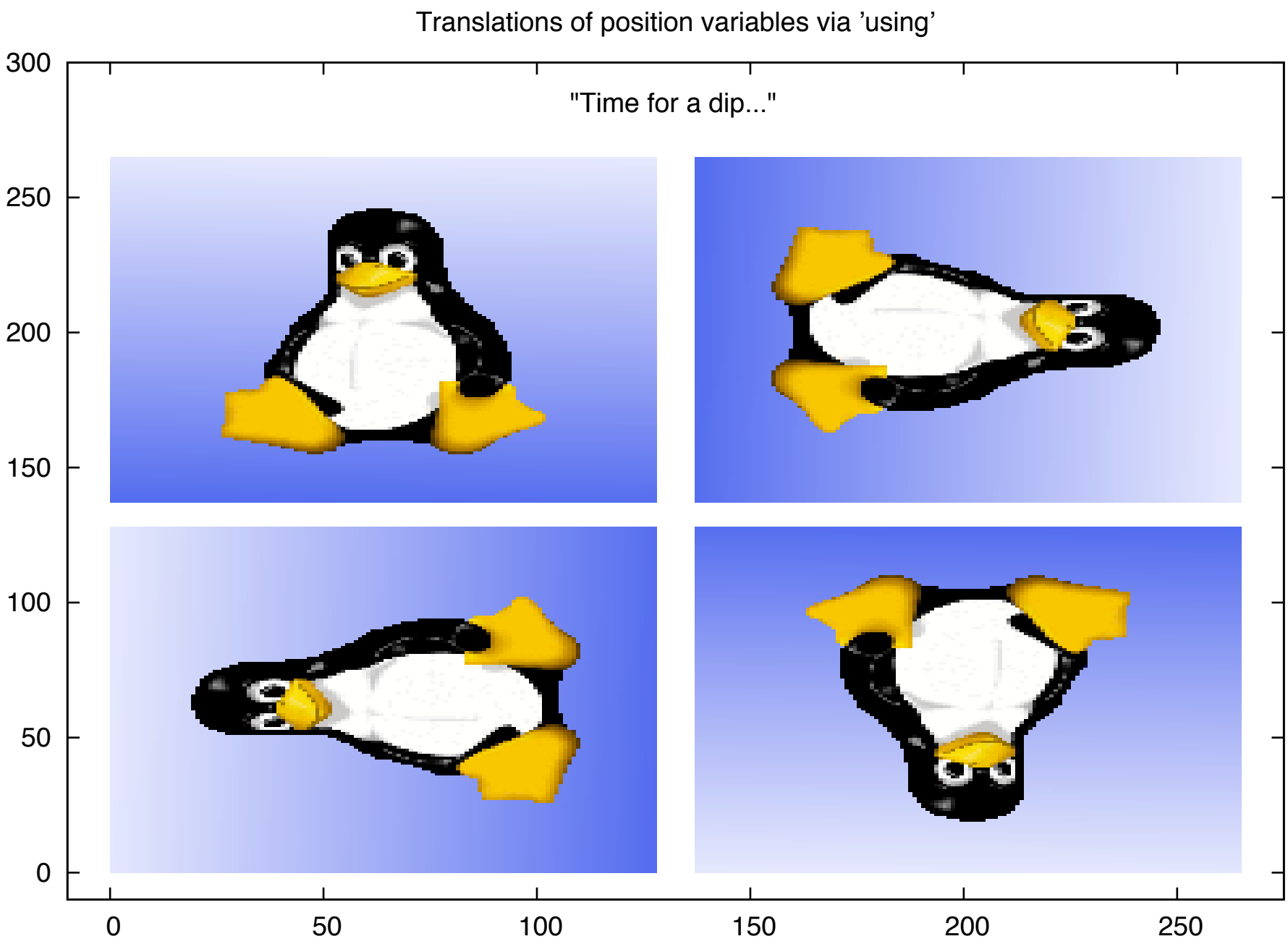

Default Histogram Colouring
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Northern Europe
Southern Europe
British Isles
Immigration from different regions
Sweden
Norway
Denmark
Yugoslavia
Romania
Greece
United Kingdom
Ireland

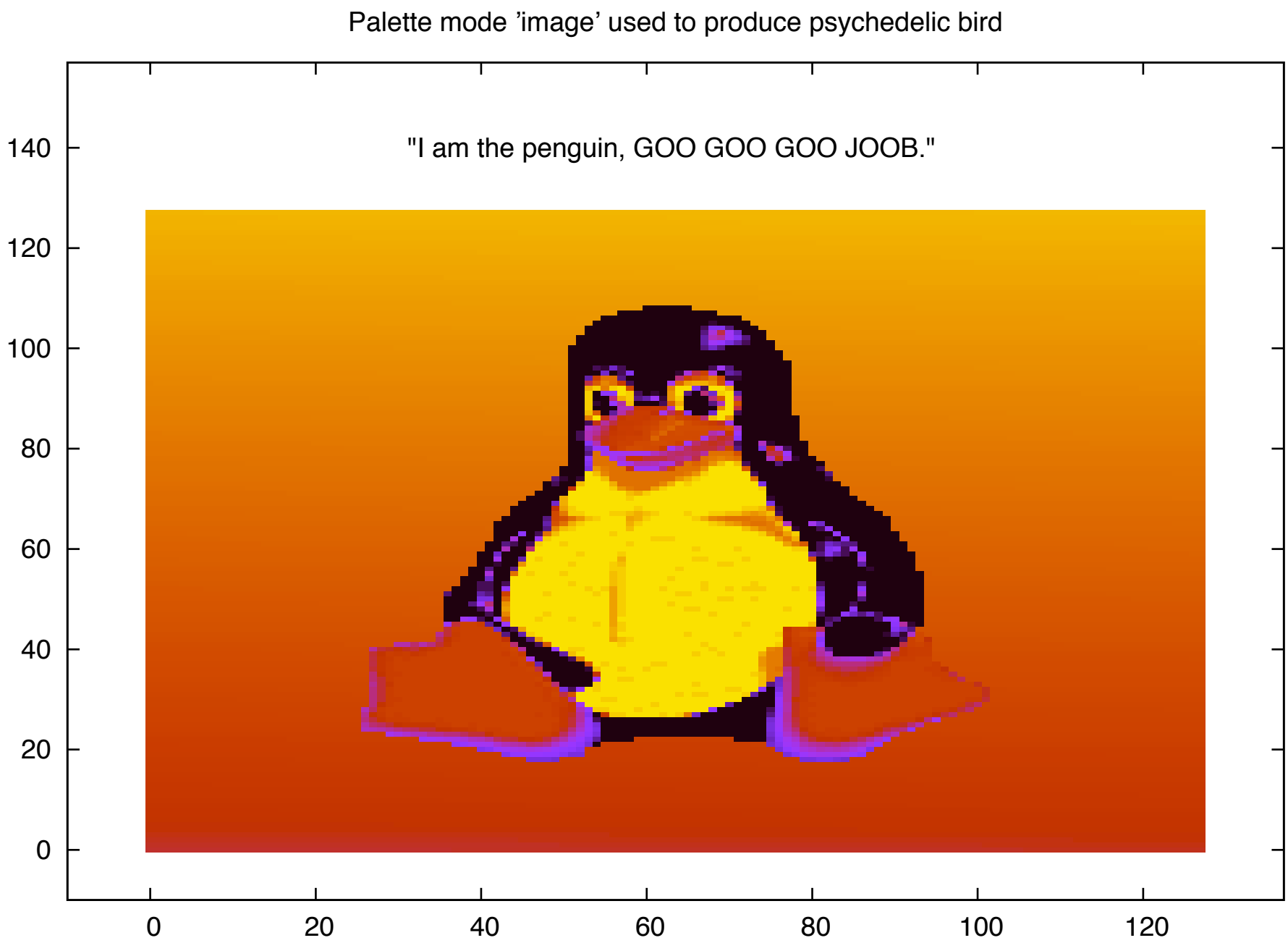

Explicit start color in 'newhistogram' command
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Northern Europe
Southern Europe
British Isles
Immigration from different regions
Sweden
Norway
Denmark
Yugoslavia
Romania
Greece
United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

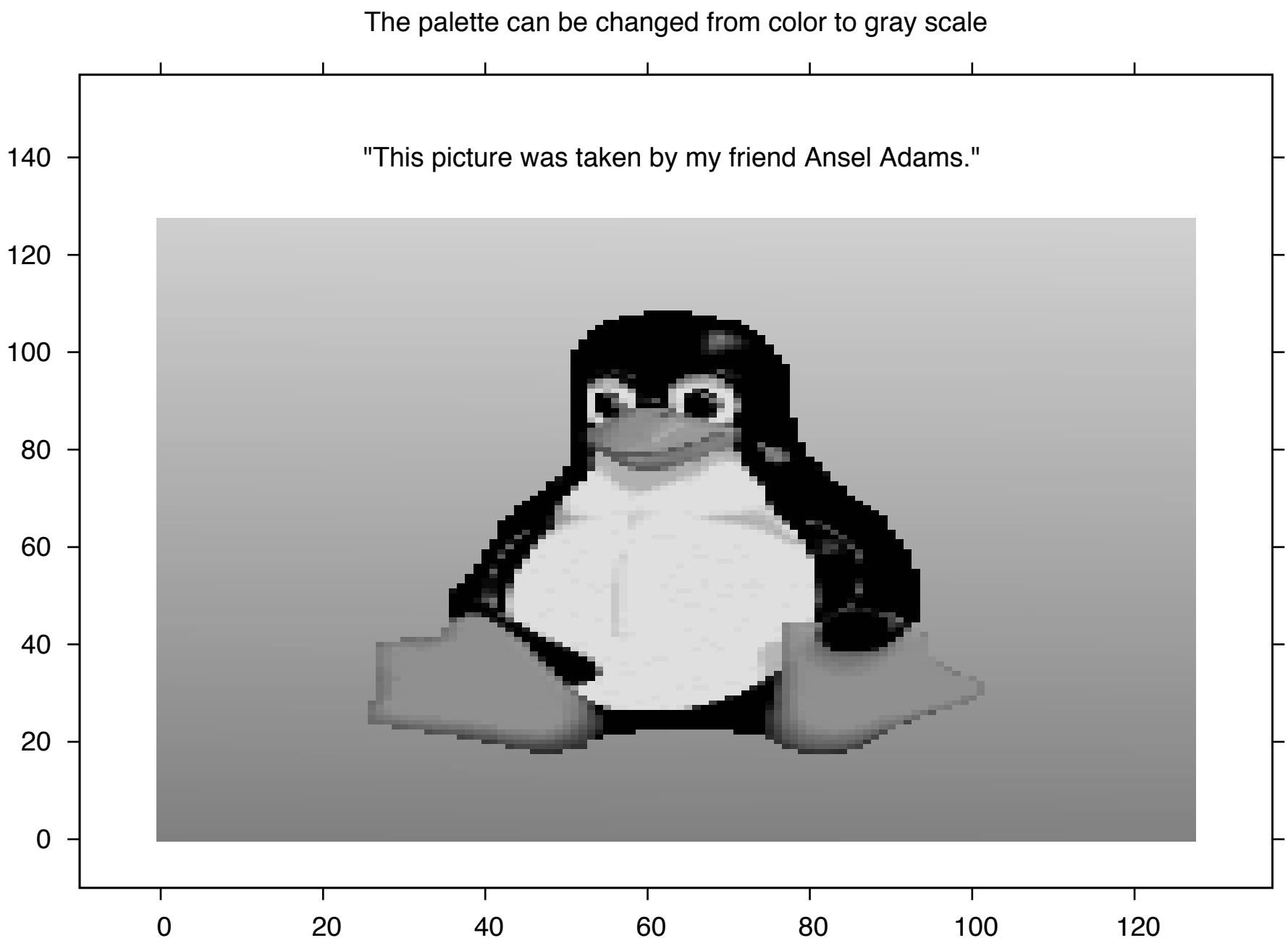

Explicit start pattern in 'newhistogram' command
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Northern Europe
Southern Europe
British Isles
Immigration from different regions
Sweden
Norway
Denmark
Yugoslavia
Romania
Greece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

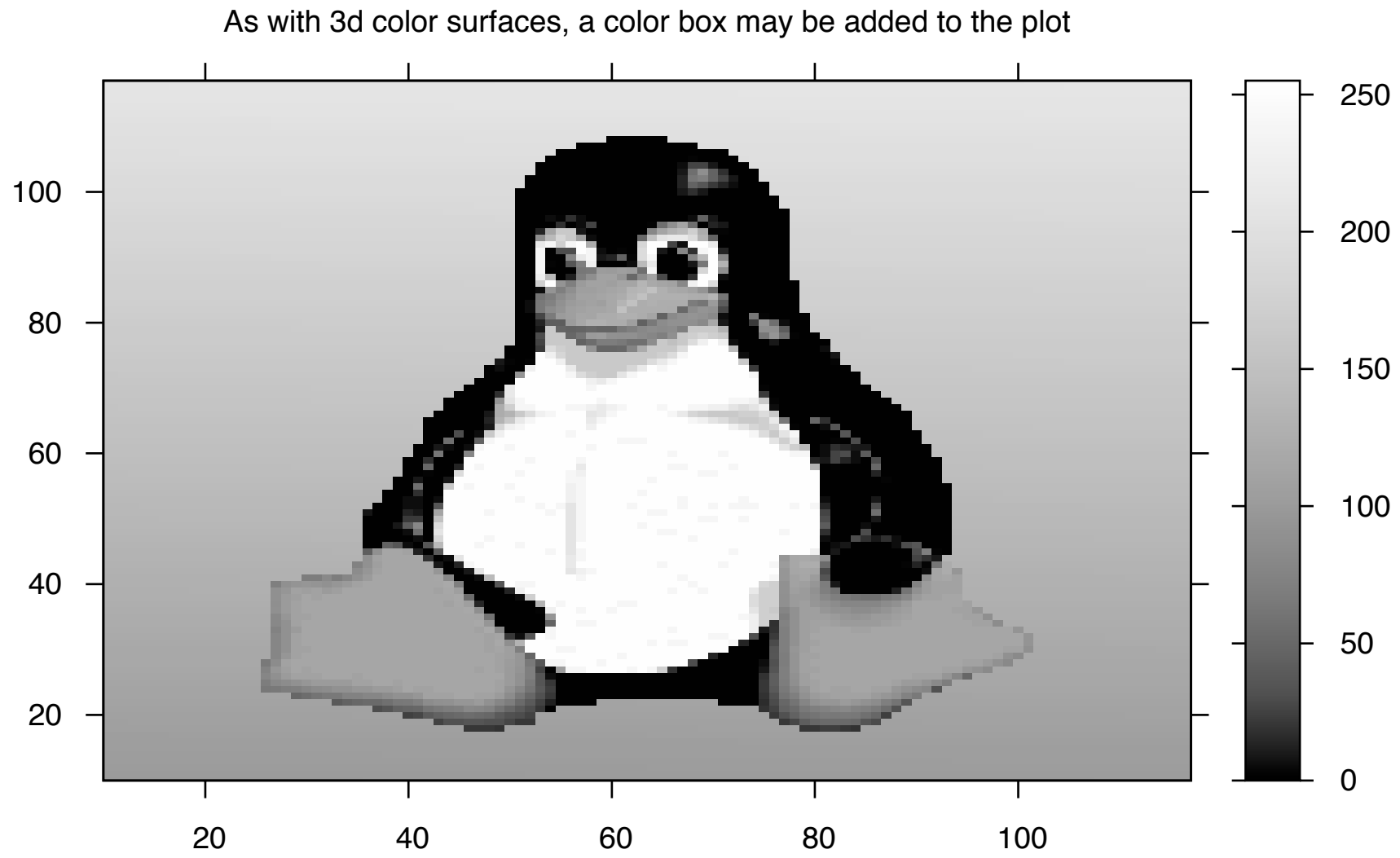

Explicit start pattern and linetype
600000
500000
400000
300000
200000
100000
0
1891-1900
1901-1910
1911-1920
1921-1930
1931-1940
1941-1950
1951-1960
1961-1970
Northern Europe
Southern Europe
British Isles
Immigration from different regions
Sweden
Norway
Denmark
Yugoslavia
Romania
Greece
United Kingdom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

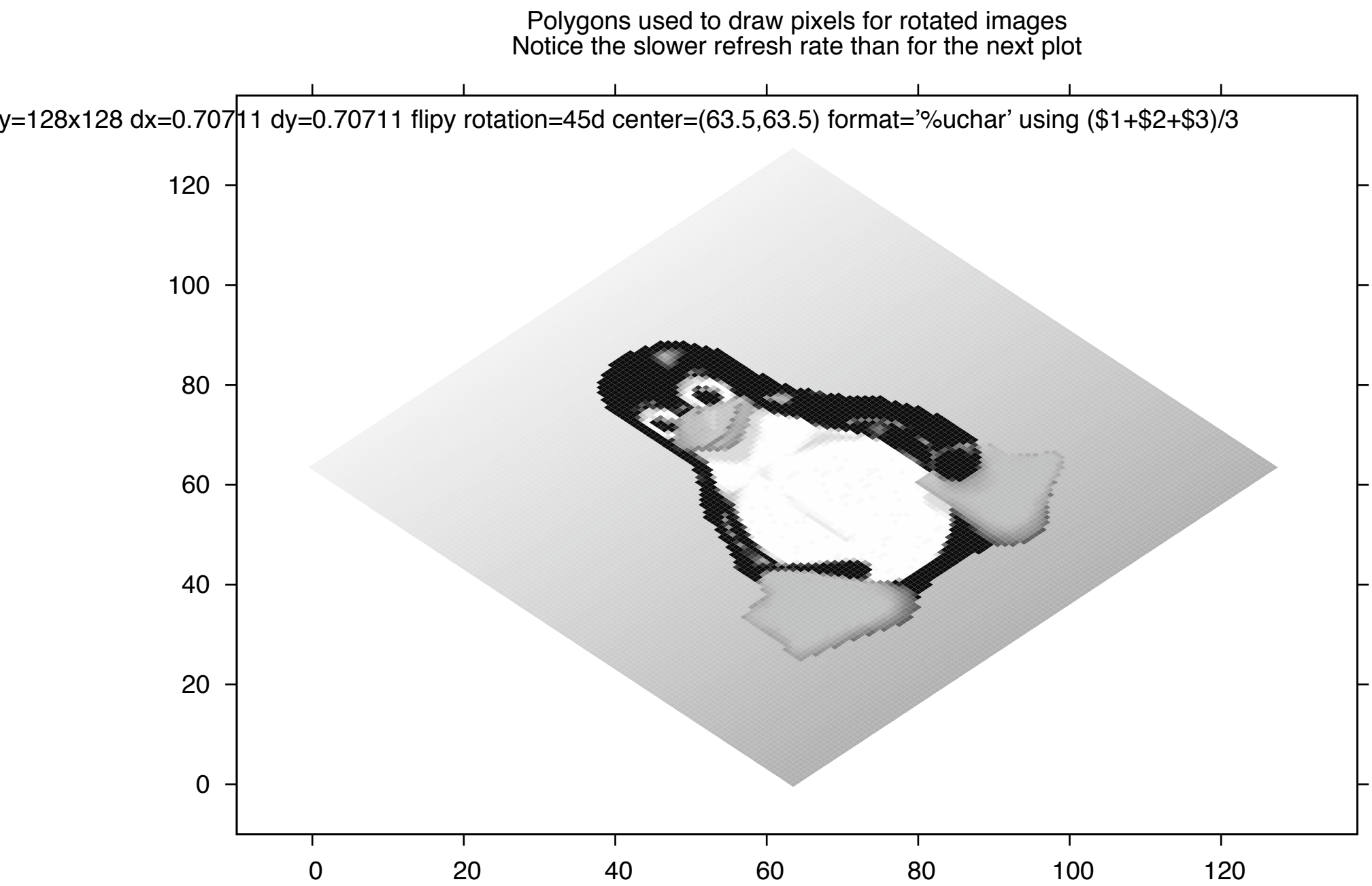

Larry Ewing’s GIMP penguin on vacation basking in
the balmy waters off the coast of Murmansk
’blutux.rgb’ binary array=128x128 flipy format=’%uchar’
"I flew here... by plane. Why? For the halibut."
140
120
100
80
60
40
20
0
0
20
40
60
80
100
120

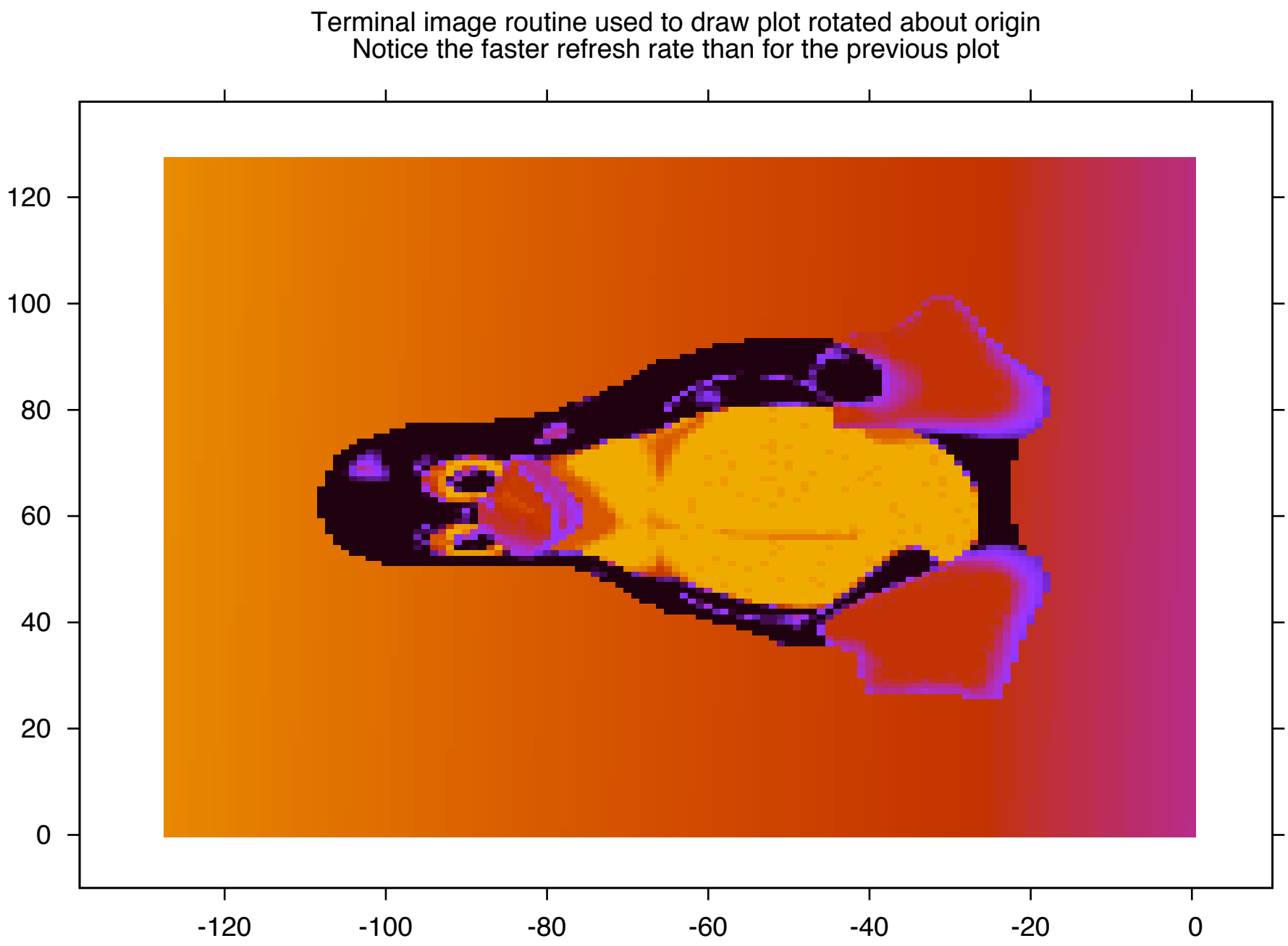

Translations of position variables via 'using'
"Time for a dip..."
300
250
200
150
100
50
0
0
50
100
150
200
250

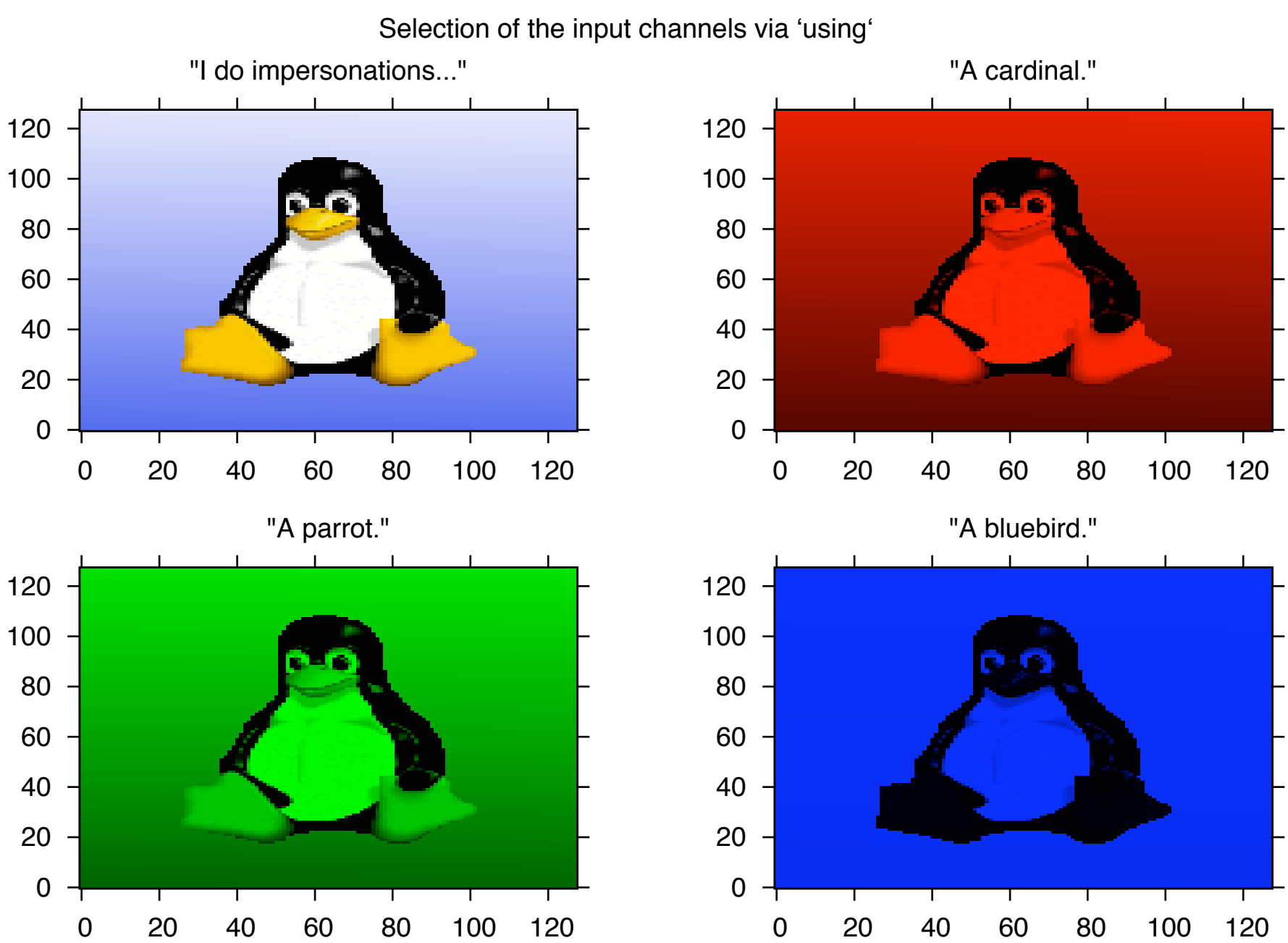

Palette mode ’image’ used to produce psychedelic bird
"I am the penguin, GOO GOO GOO JOOB."
140
120
100
80
60
40
20
0
0
20
40
60
80
100
120

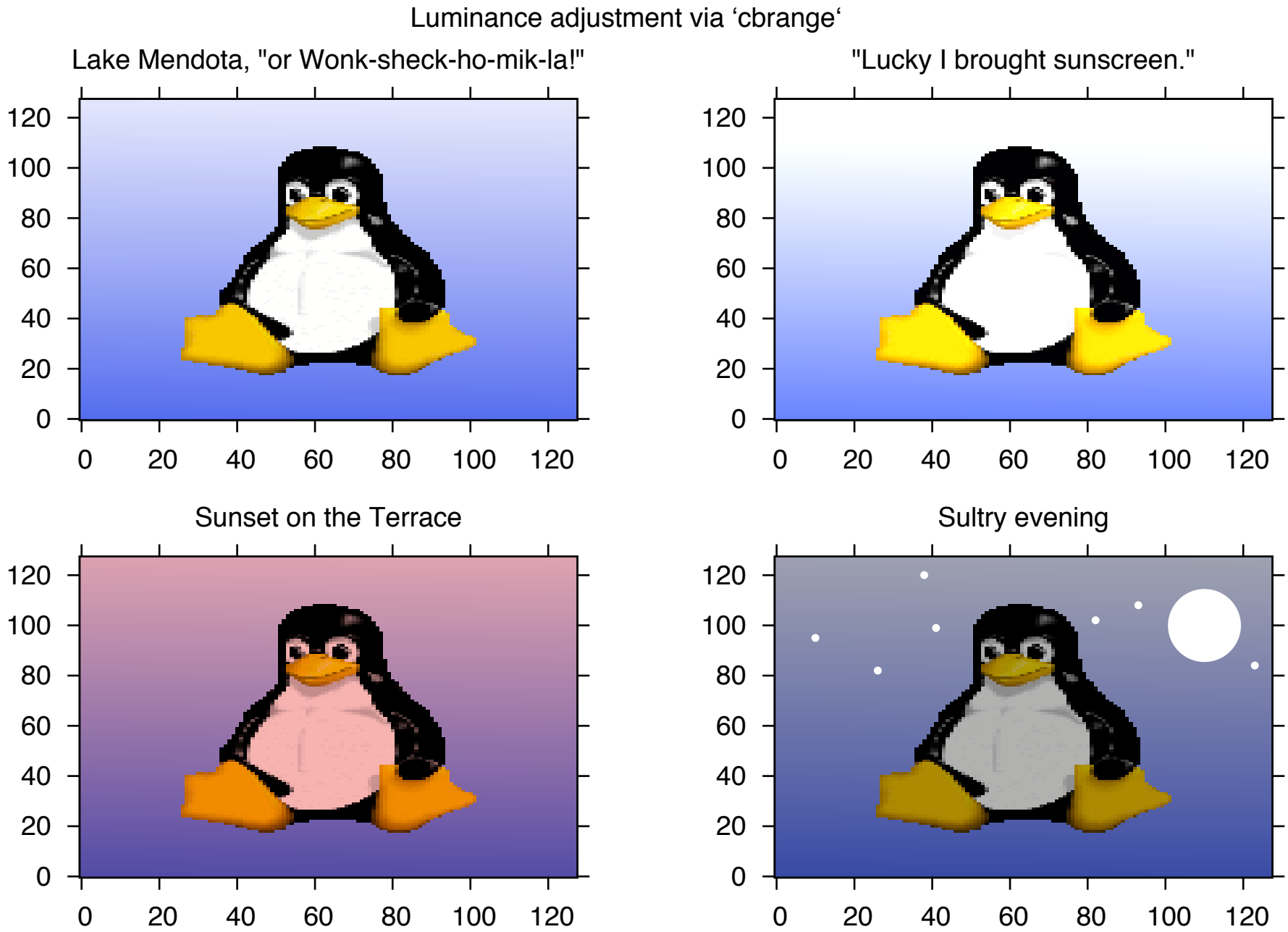

The palette can be changed from color to gray scale
"This picture was taken by my friend Ansel Adams."
140
120
100
80
60
40
20
0
0
20
40
60
80
100
120

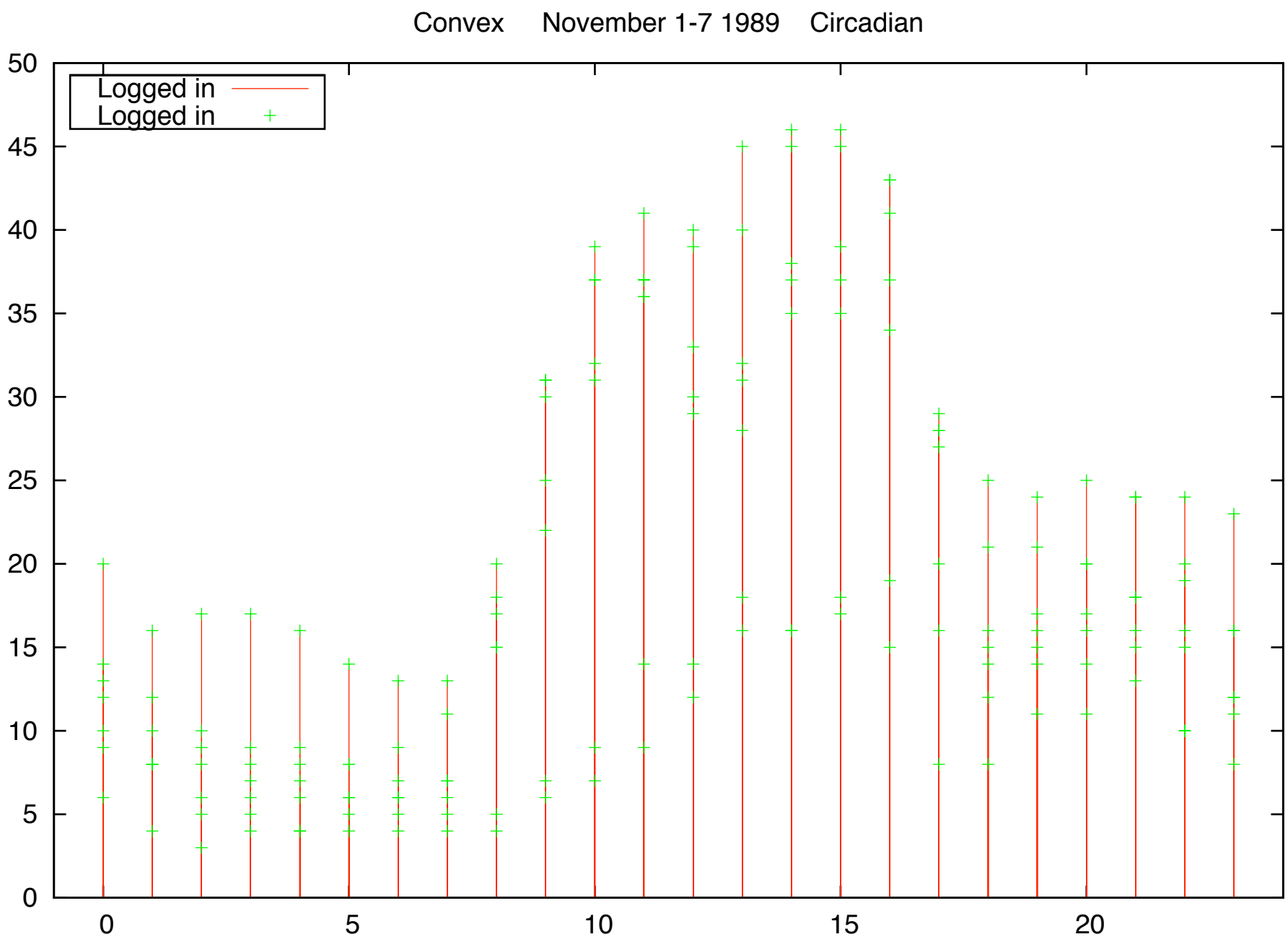

As with 3d color surfaces, a color box may be added to the plot
100
80
60
40
20
20
40
60
80
100
250
200
150
100
50
0

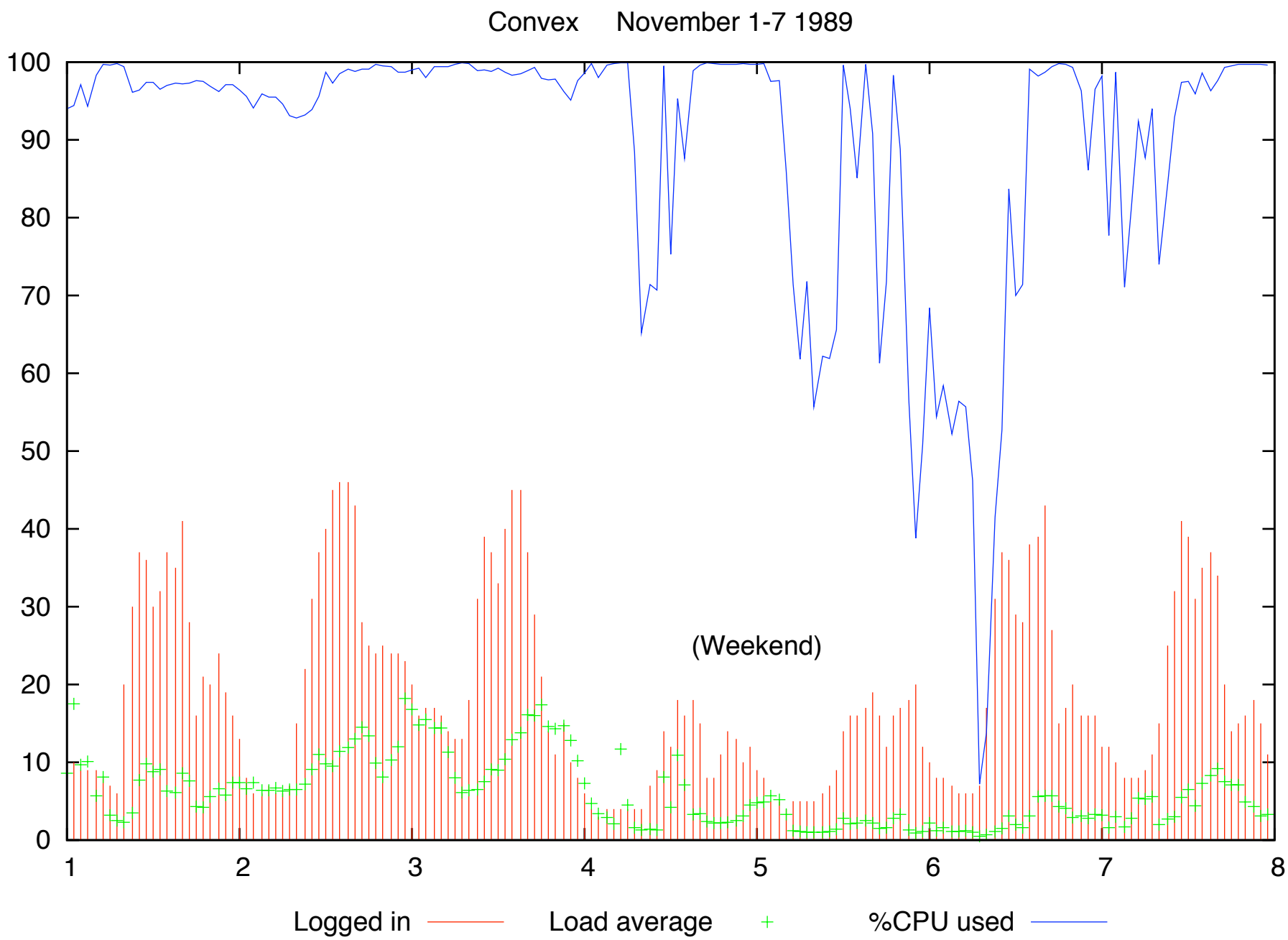

Polygons used to draw pixels for rotated images
Notice the slower refresh rate than for the next plot
y=128x128 dx=0.70711 dy=0.70711 flipy rotation=45d center=(63.5,63.5) format='%uchar' using ($1+$2+$3)/3
120
100
80
60
40
20
0
0
20
40
60
80
100
120

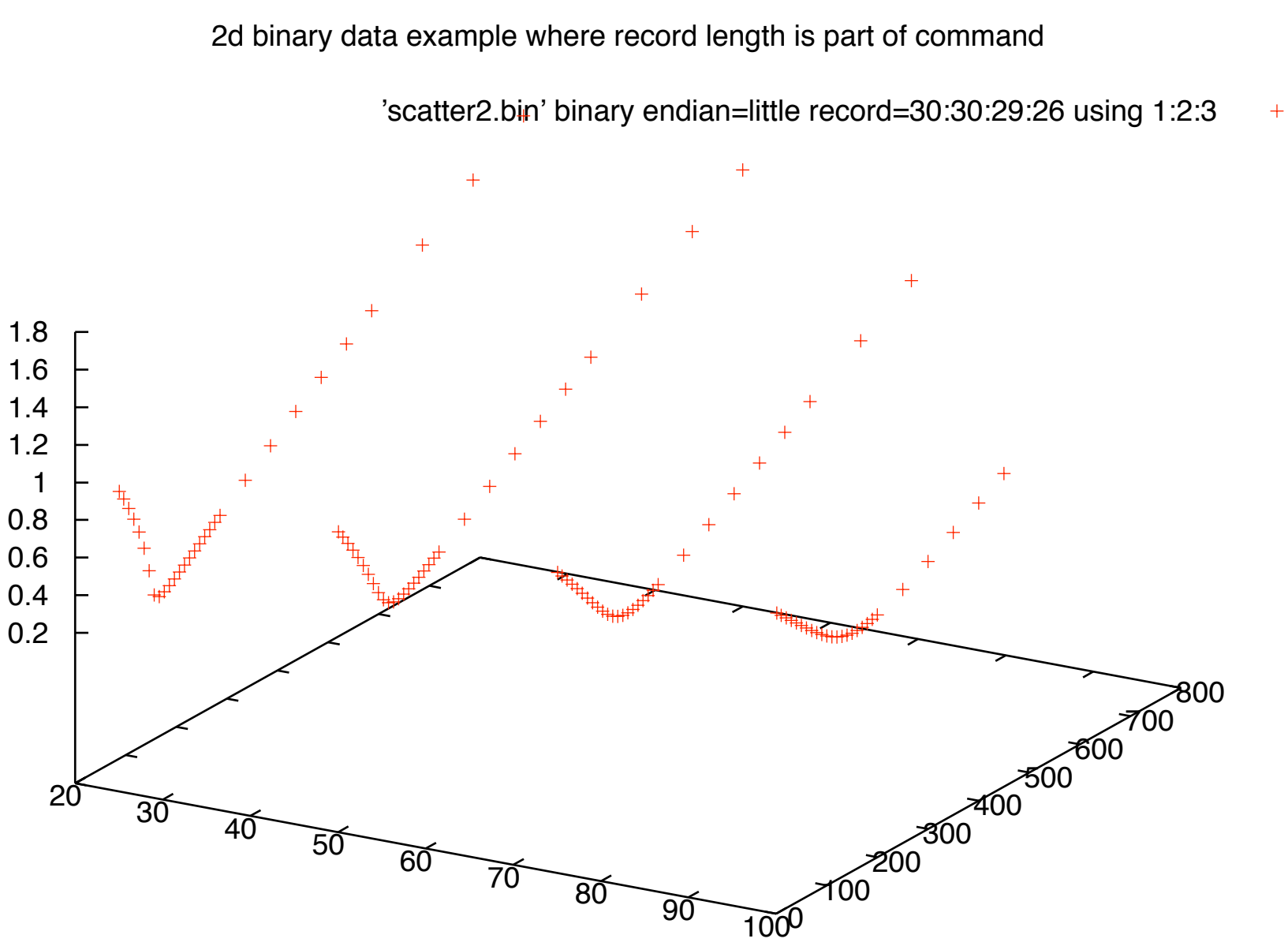

Terminal image routine used to draw plot rotated about origin
Notice the faster refresh rate than for the previous plot
120
100
80
60
40
20
0
-120
-100
-80
-60
-40
-20
0


Selection of the input channels via ‘using‘
"I do impersonations..."
"A cardinal."
"A parrot."
"A bluebird."
0
20
40
60
80
100
120


Luminance adjustment via ‘cbrange‘
Lake Mendota, "or Wonk-sheck-ho-mik-la!"
"Lucky I brought sunscreen."
Sunset on the Terrace
Sultry evening
0
20
40
60
80
100
120

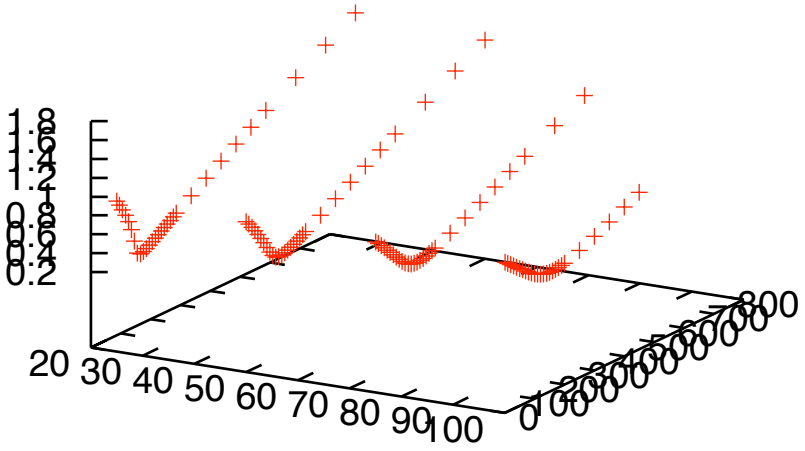

Convex November 1-7 1989 Circadian
Logged in
Logged in
50
45
40
35
30
25
20
15
10
5
0
0
5
10
15
20


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

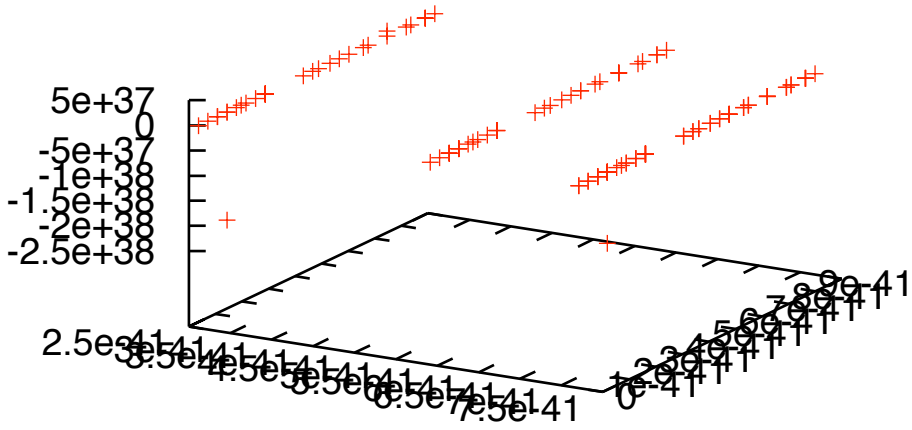

2d binary data example where record length is part of command
'scatter2.bin' binary endian=little record=30:30:29:26 using 1:2:3
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500
600
700
800

If plots in columns match, your compiler is little endian

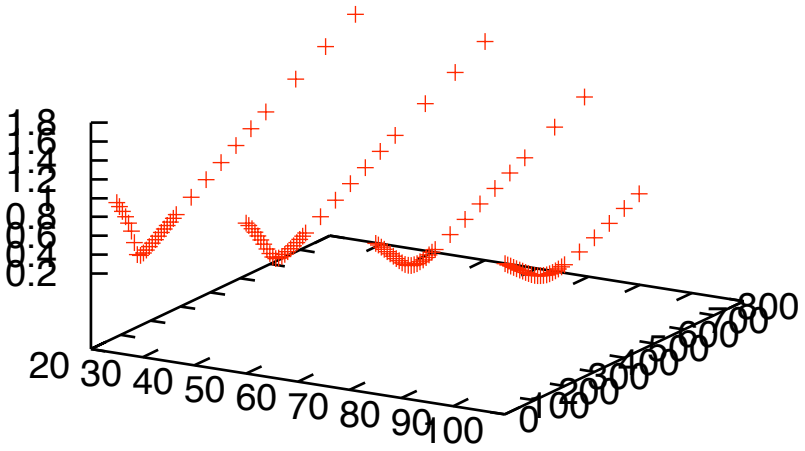
Little endian



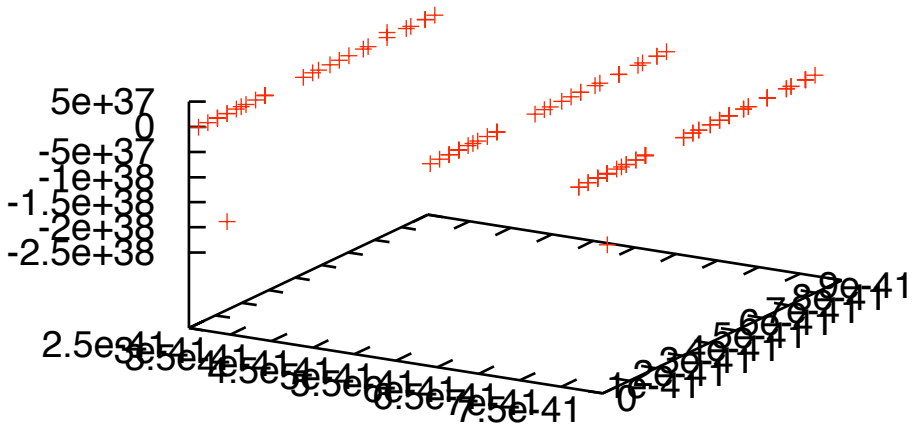
Big endian

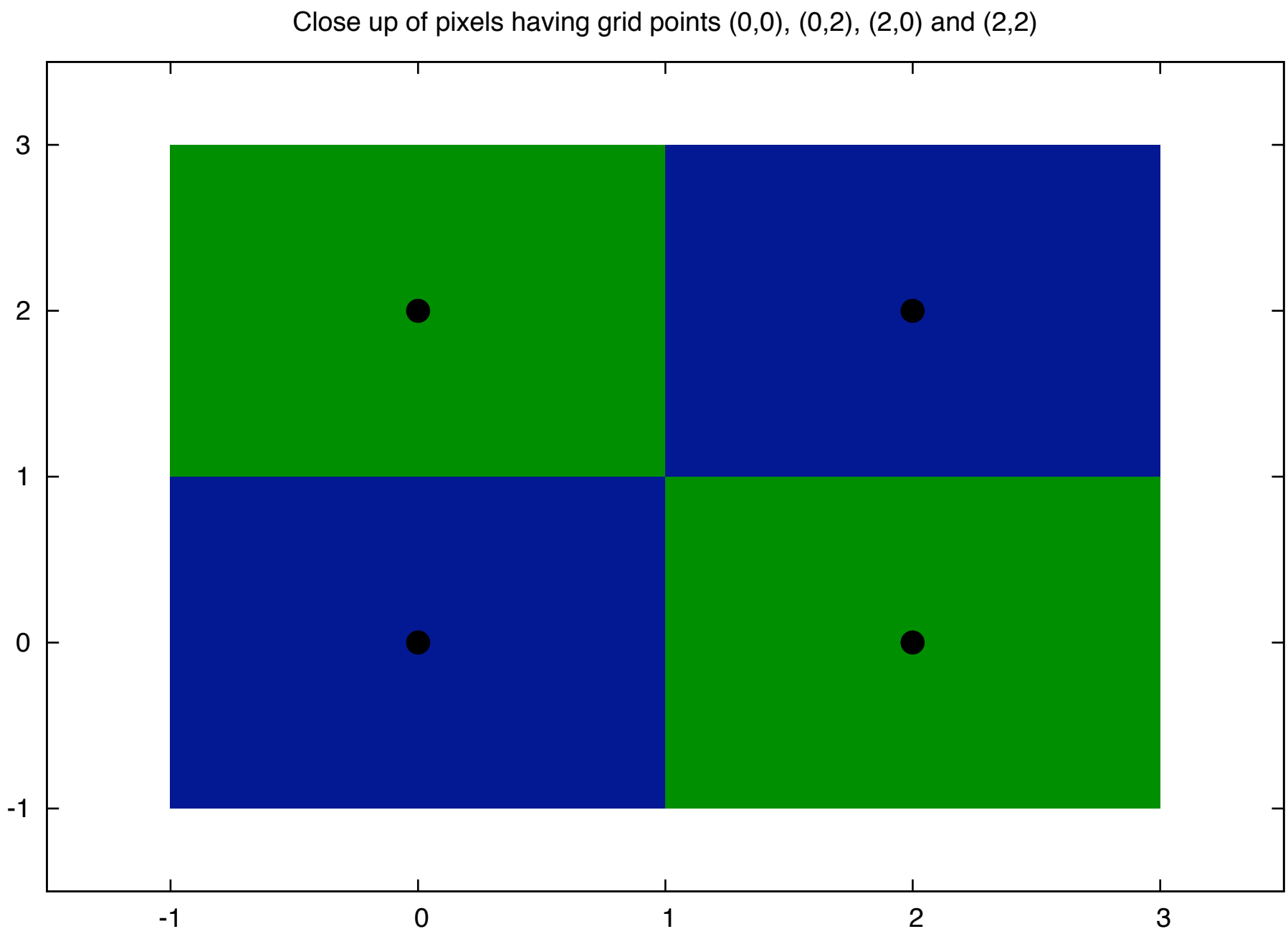


Default



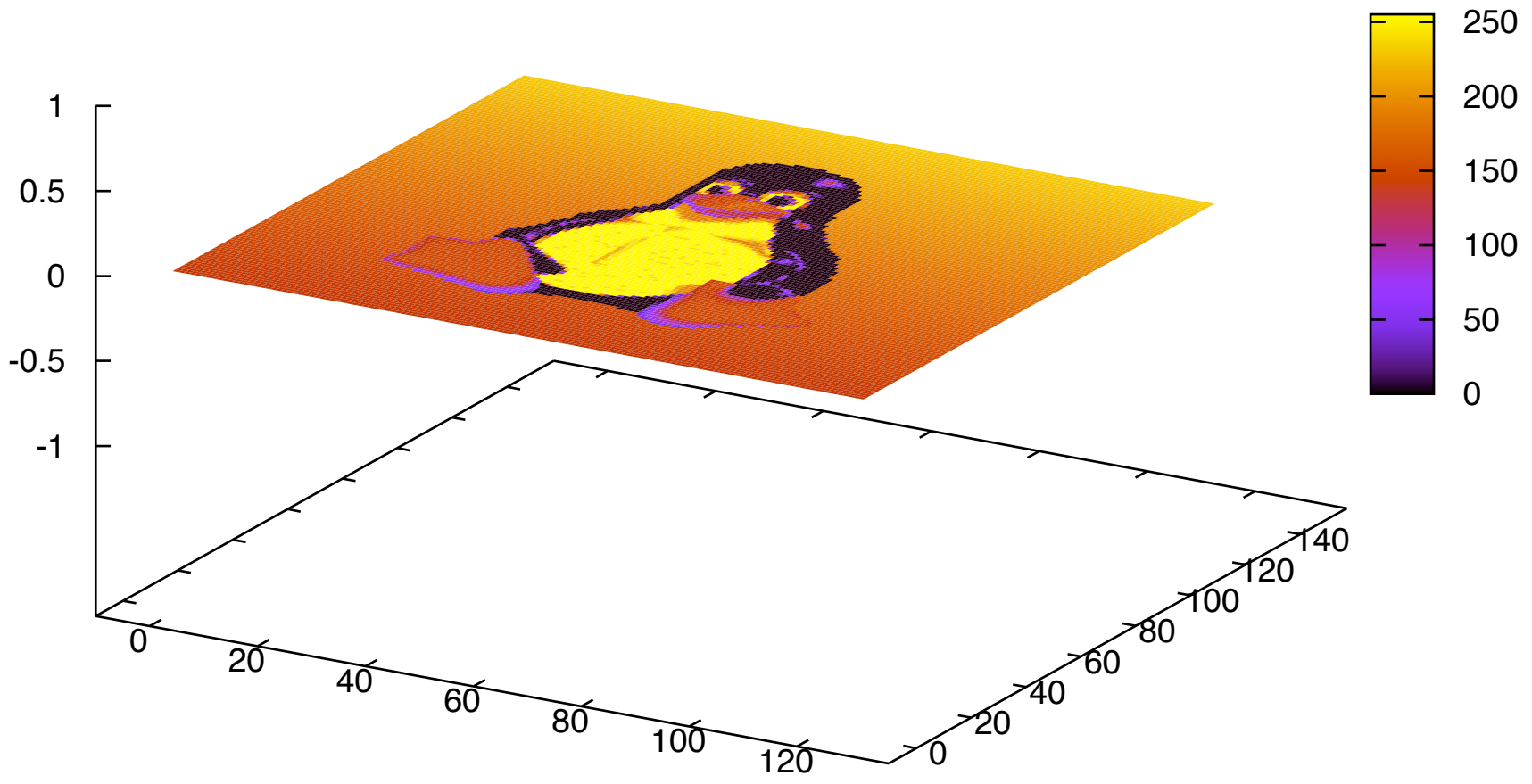
Swapped




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

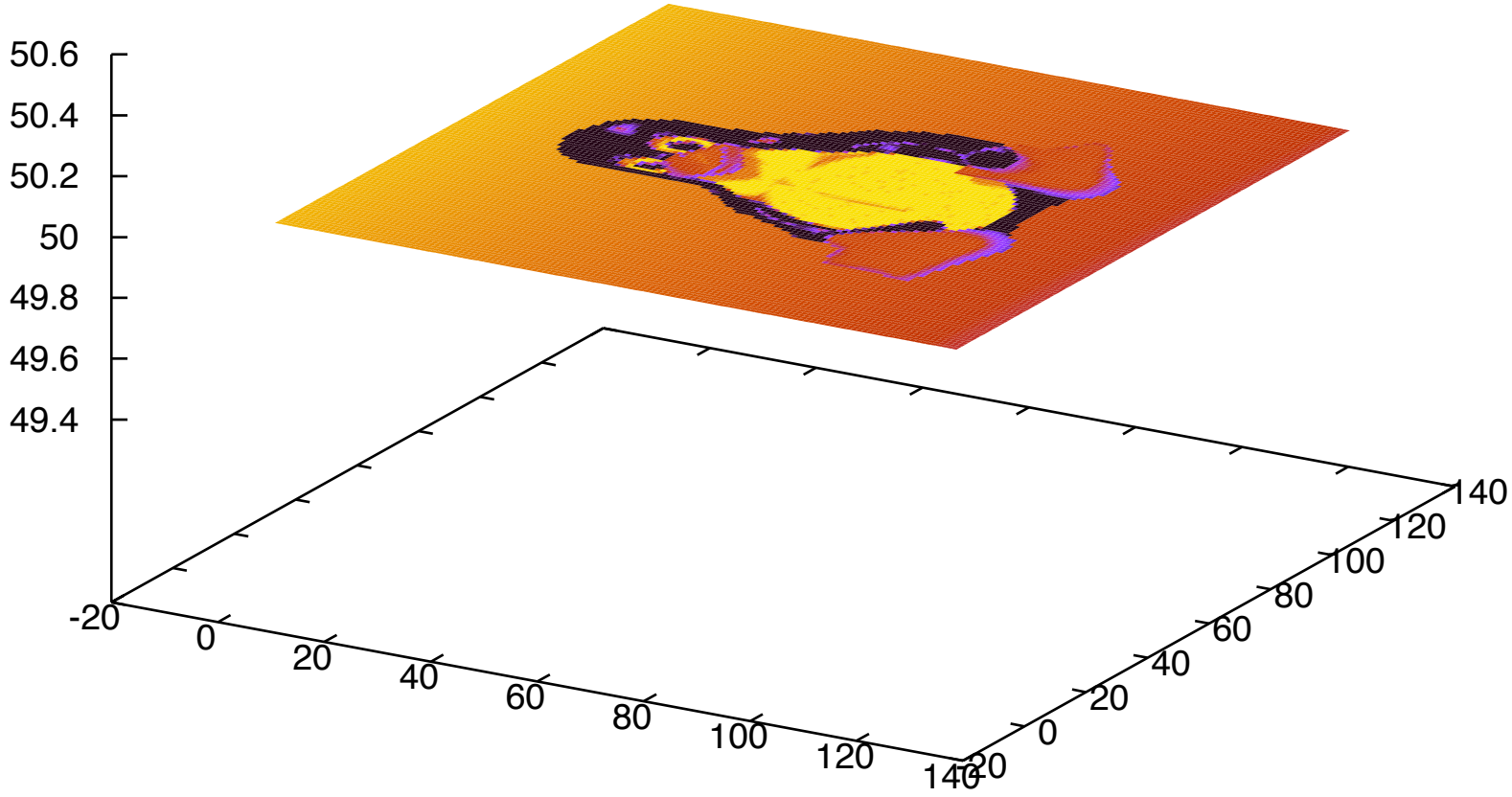
Simple extension of a two dimensional image into three dimensions

'blutux.rgb' binary array=128x128 flip=y format='%uchar%uchar%uchar' using ($1+$2+$3)/3



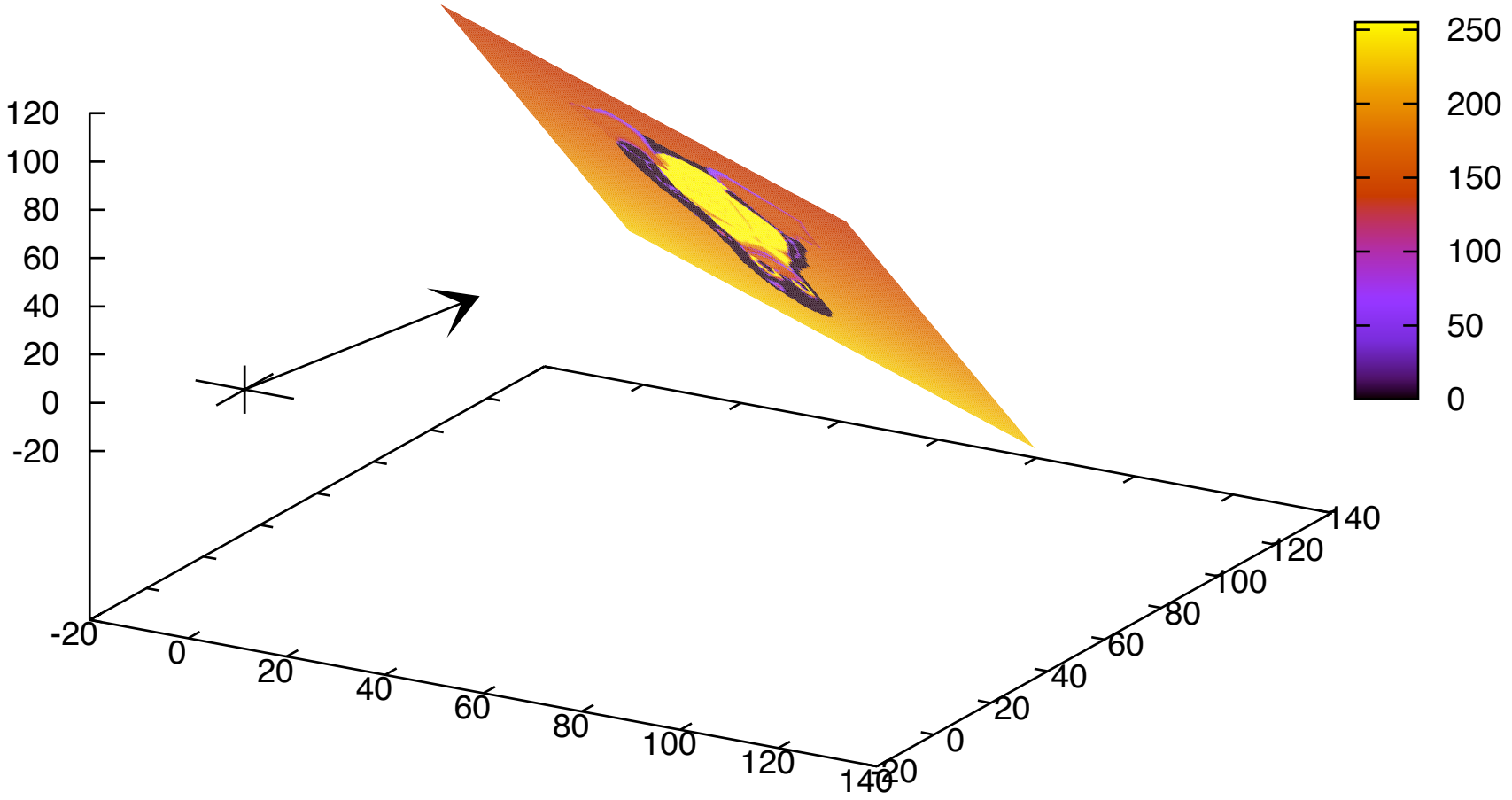
Orientation operations from ’plot’ also apply to to ’splot’

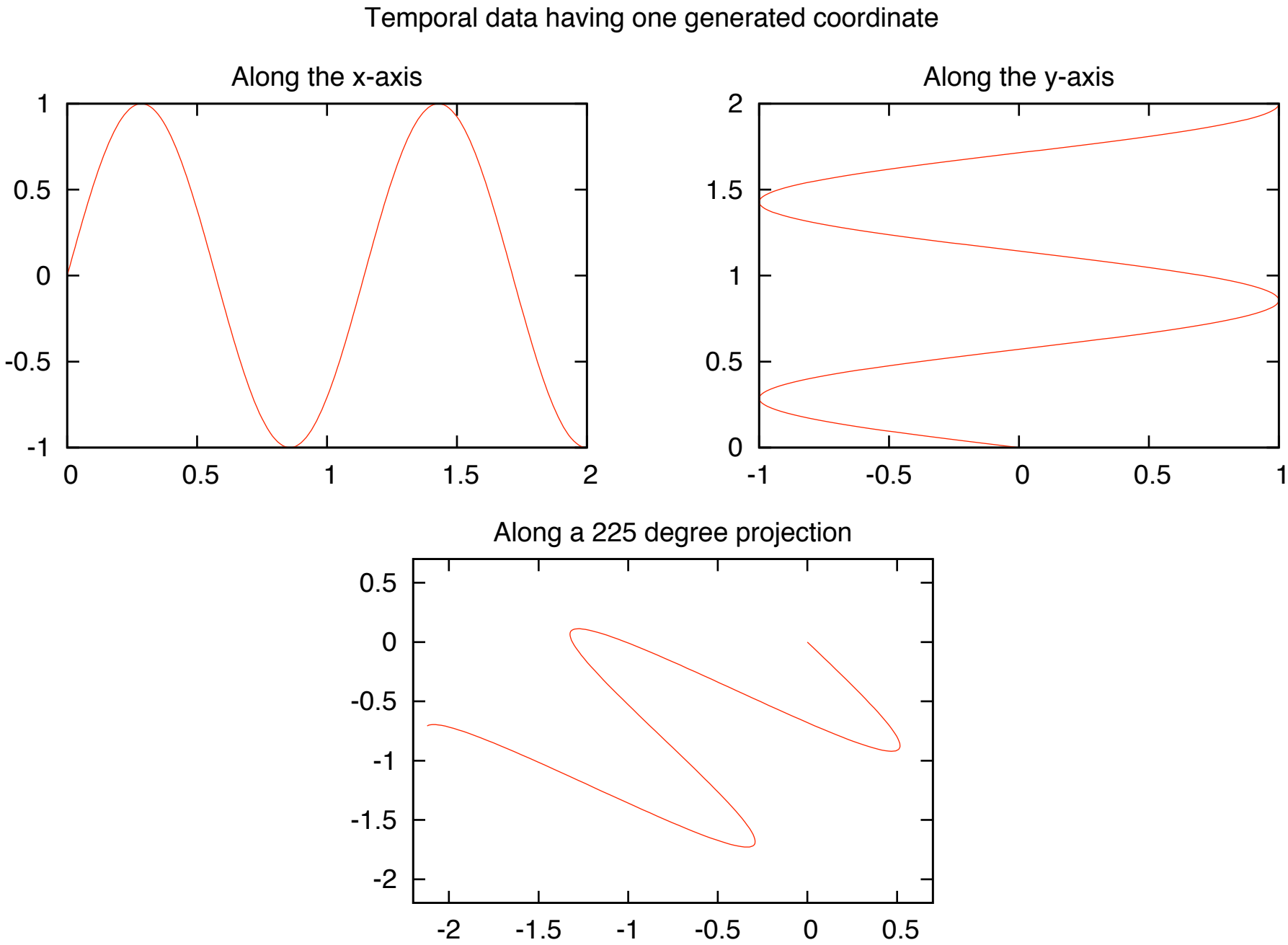
inary array=128x128 flipy rotate=90d center = (63.5,63.5,50) format=’%uchar%uchar%uchar’ using ($1+$2+$3)



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

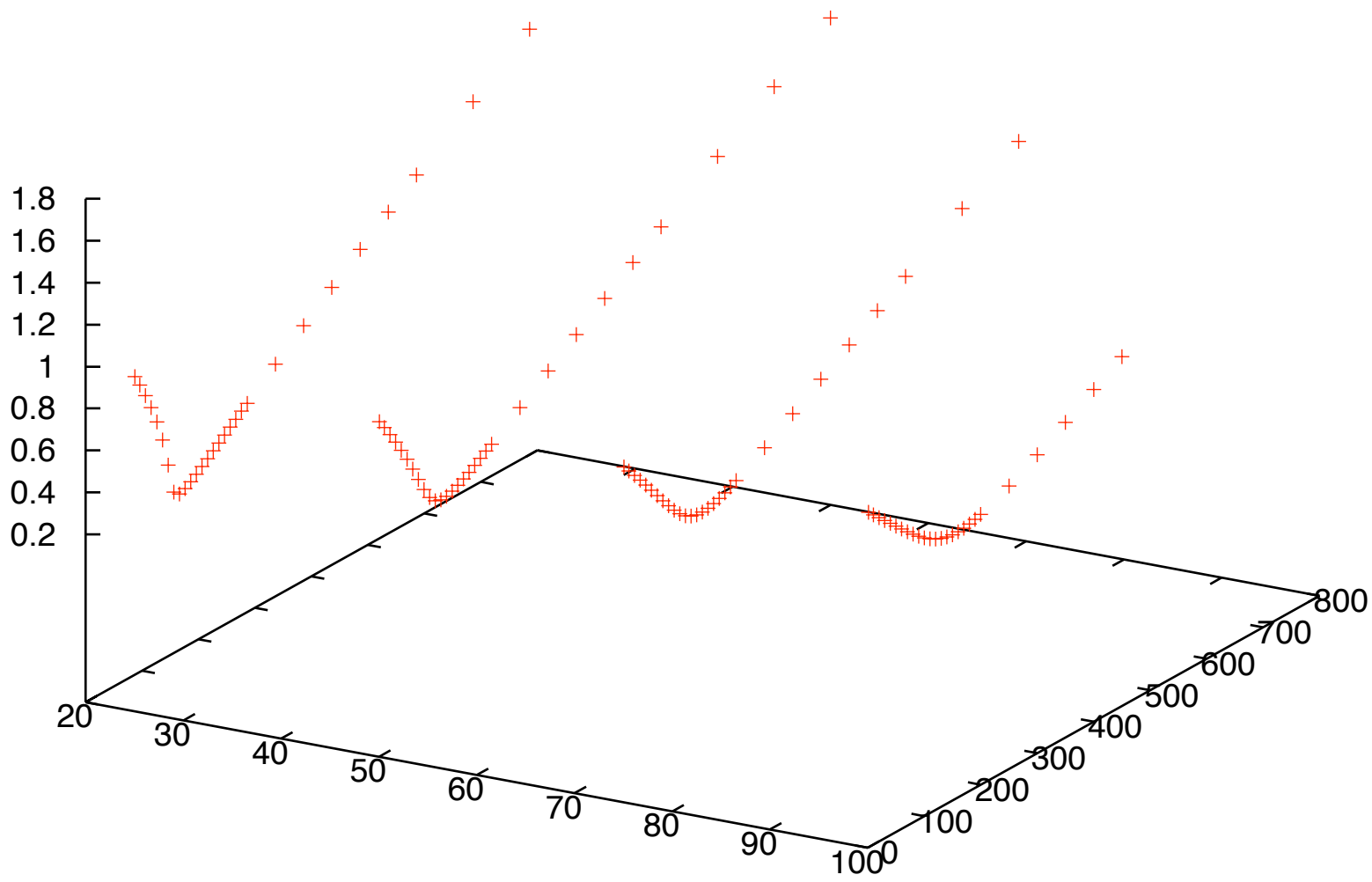
128x128 flipy rot=1.0pi center = (63.5,63.5,50) perp=(1,1,1) format=’%uchar%uchar%uchar’ using ($1+$2+$3)/3




Temporal data having one generated coordinate
Along the x-axis
1
0.5
0
-0.5
-1
0
0.5
1
1.5
2
Along the y-axis
2
1.5
1
0.5
0
-1
-0.5
0
0.5
1
Along a 225 degree projection
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5

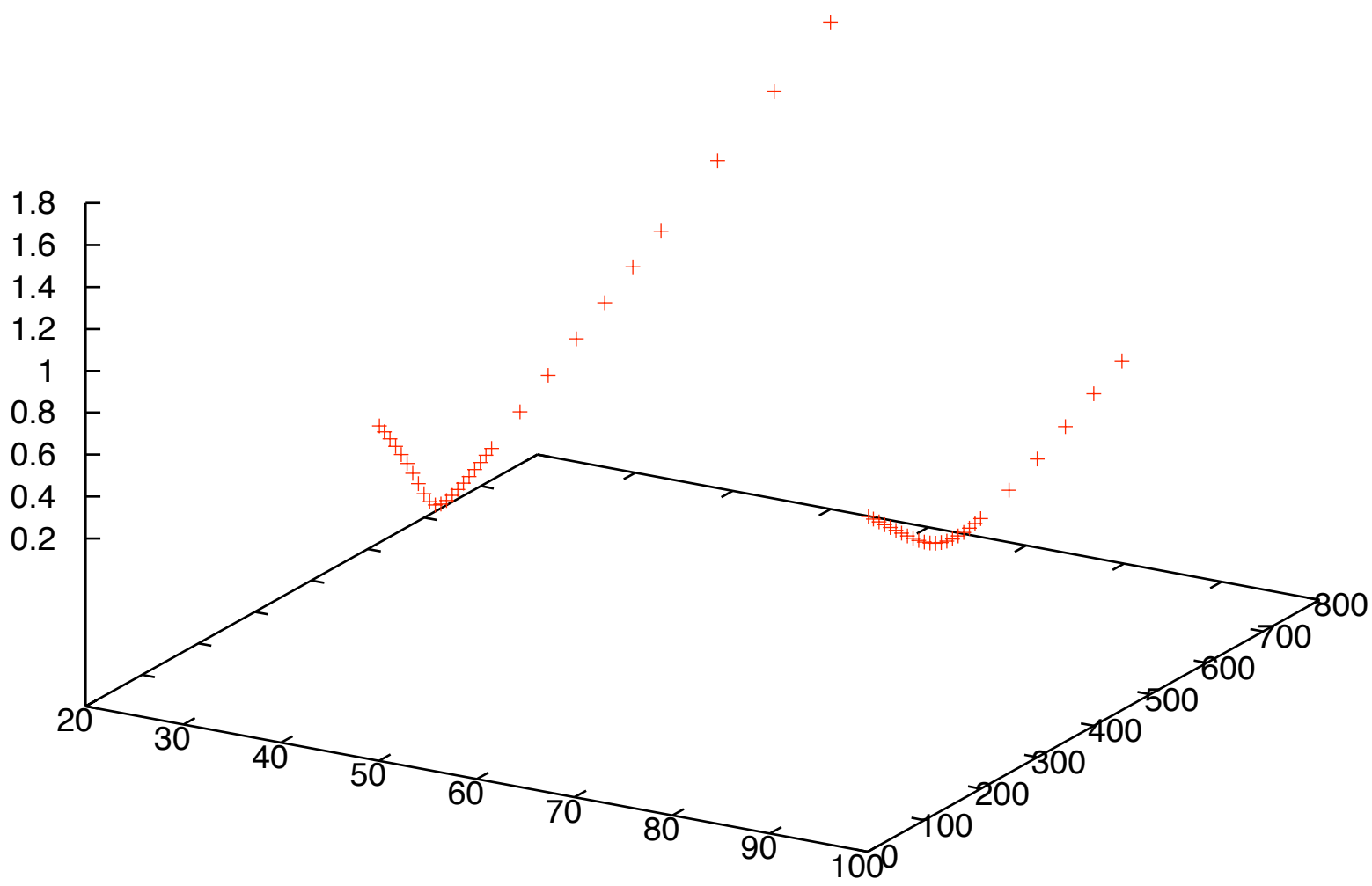
2d binary data example where x coordinate is ignored then generated

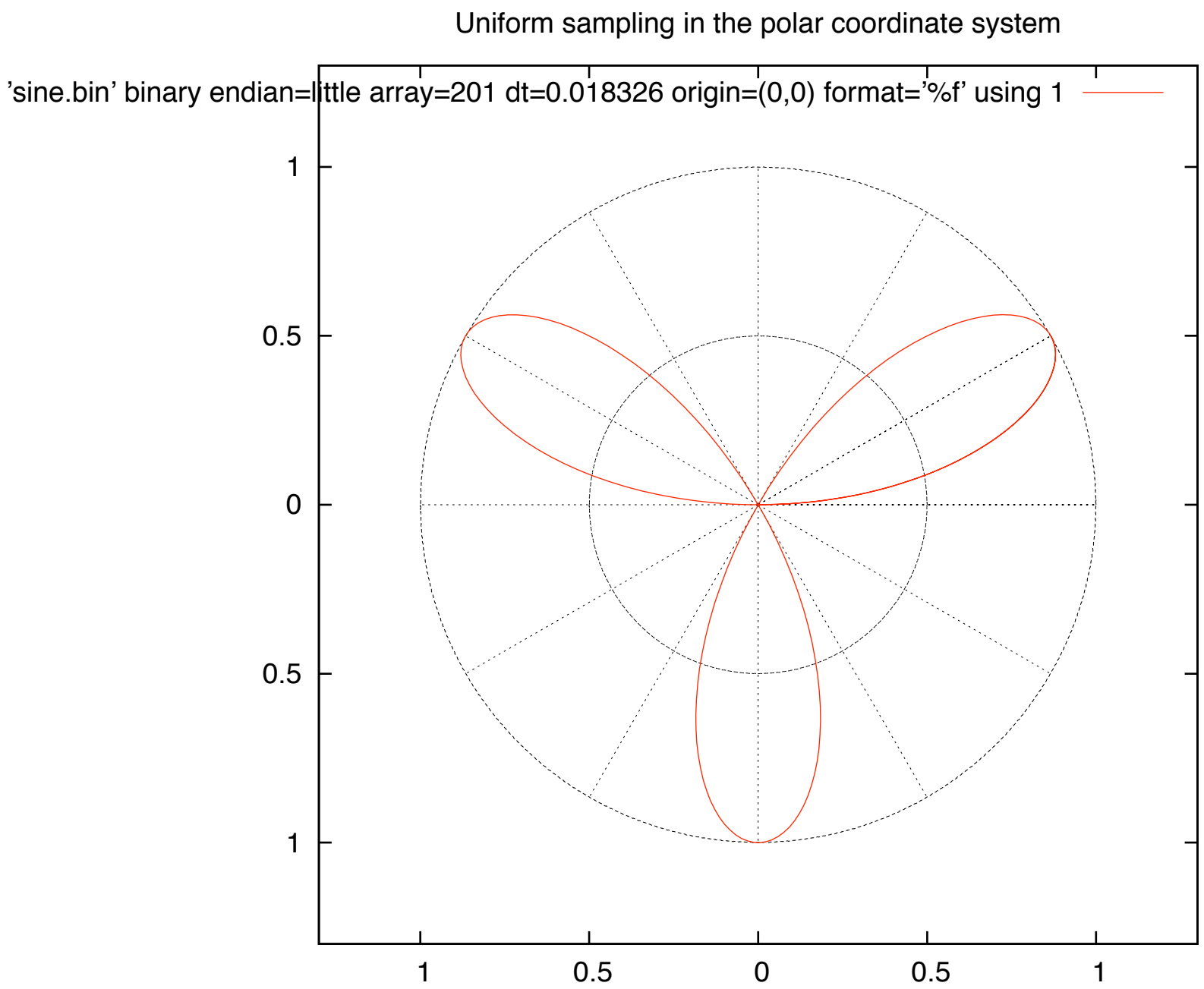
in' binary endian=little record=30:30:29:26 origin=(25,0,0):(50,0,0):(75,0,0):(100,0,0) format='%f%f' using (0):2:3

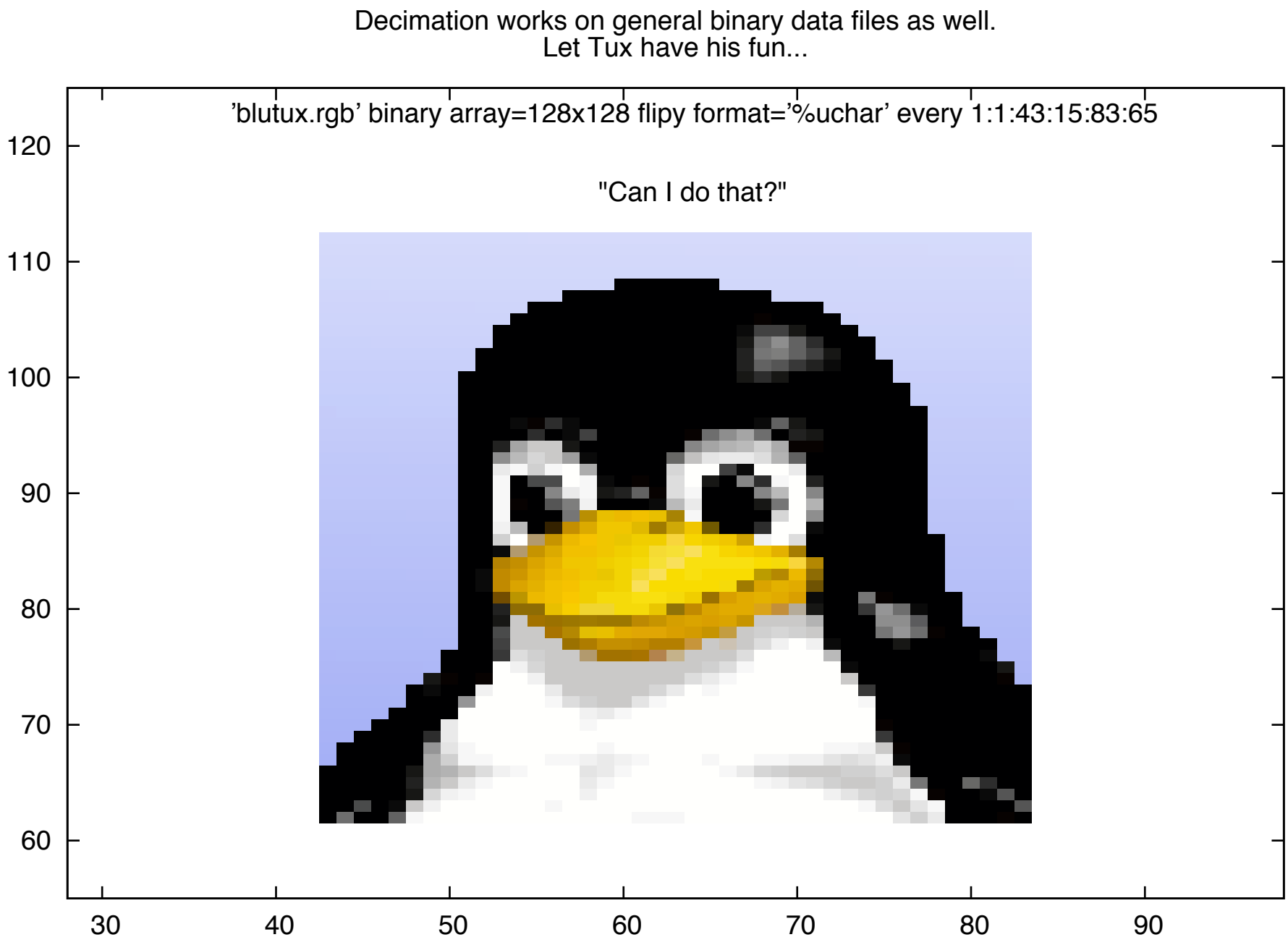


The key word ’skip’ used to ignore some data

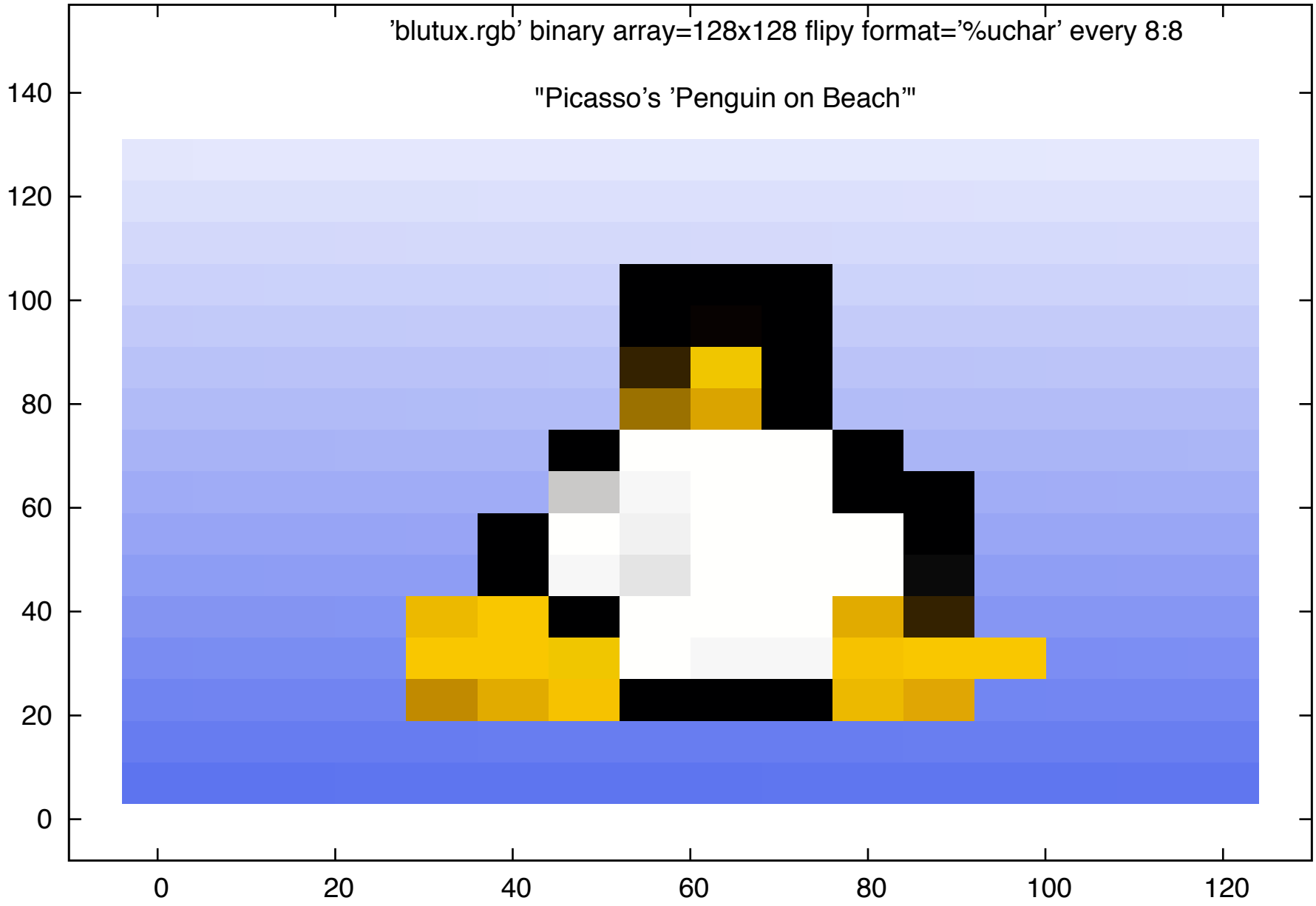
catter2.bin’ binary endian=little record=30:26 skip=360:348 origin=(50,0,0):(100,0,0) format=’%f%f’ using (0):2:3

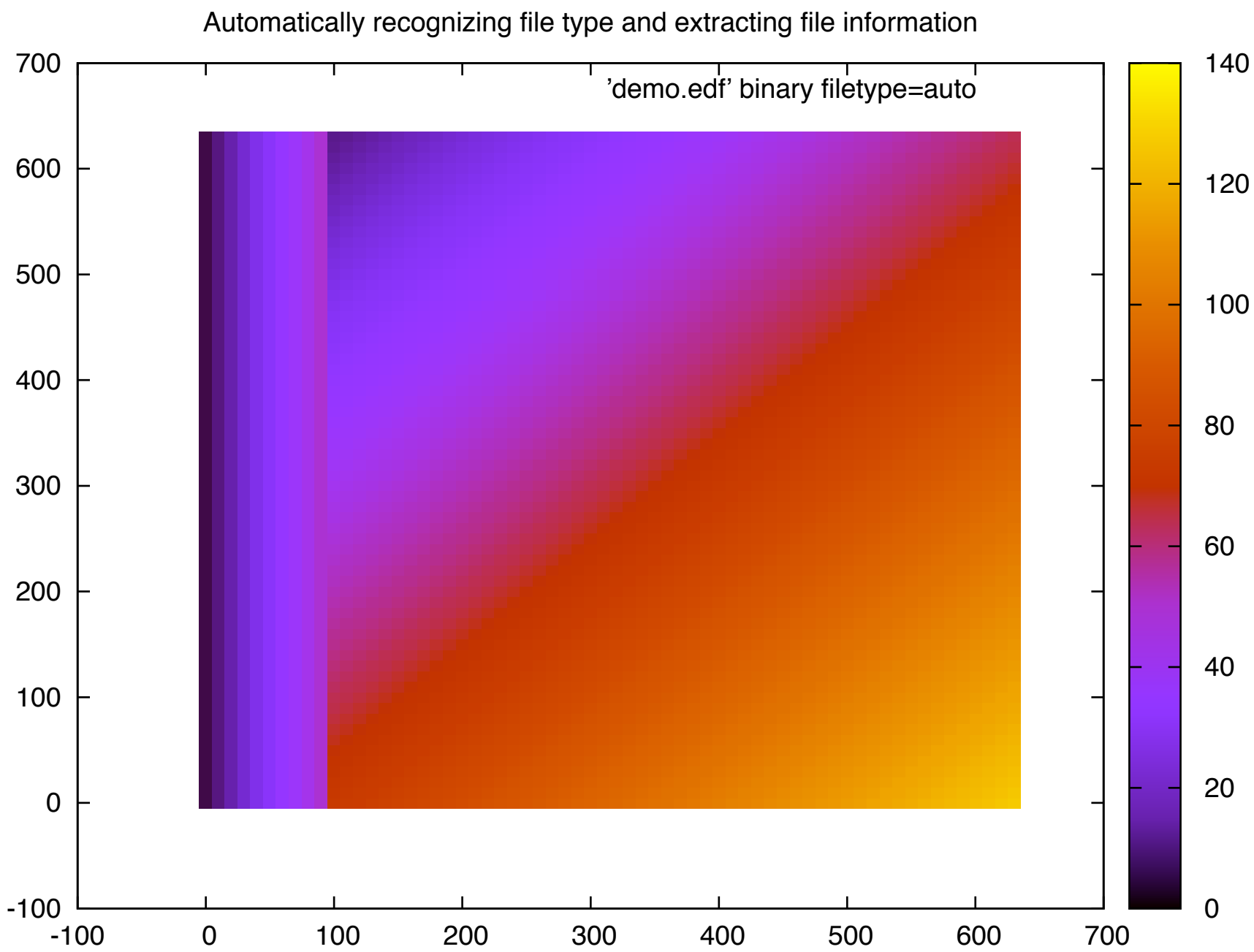


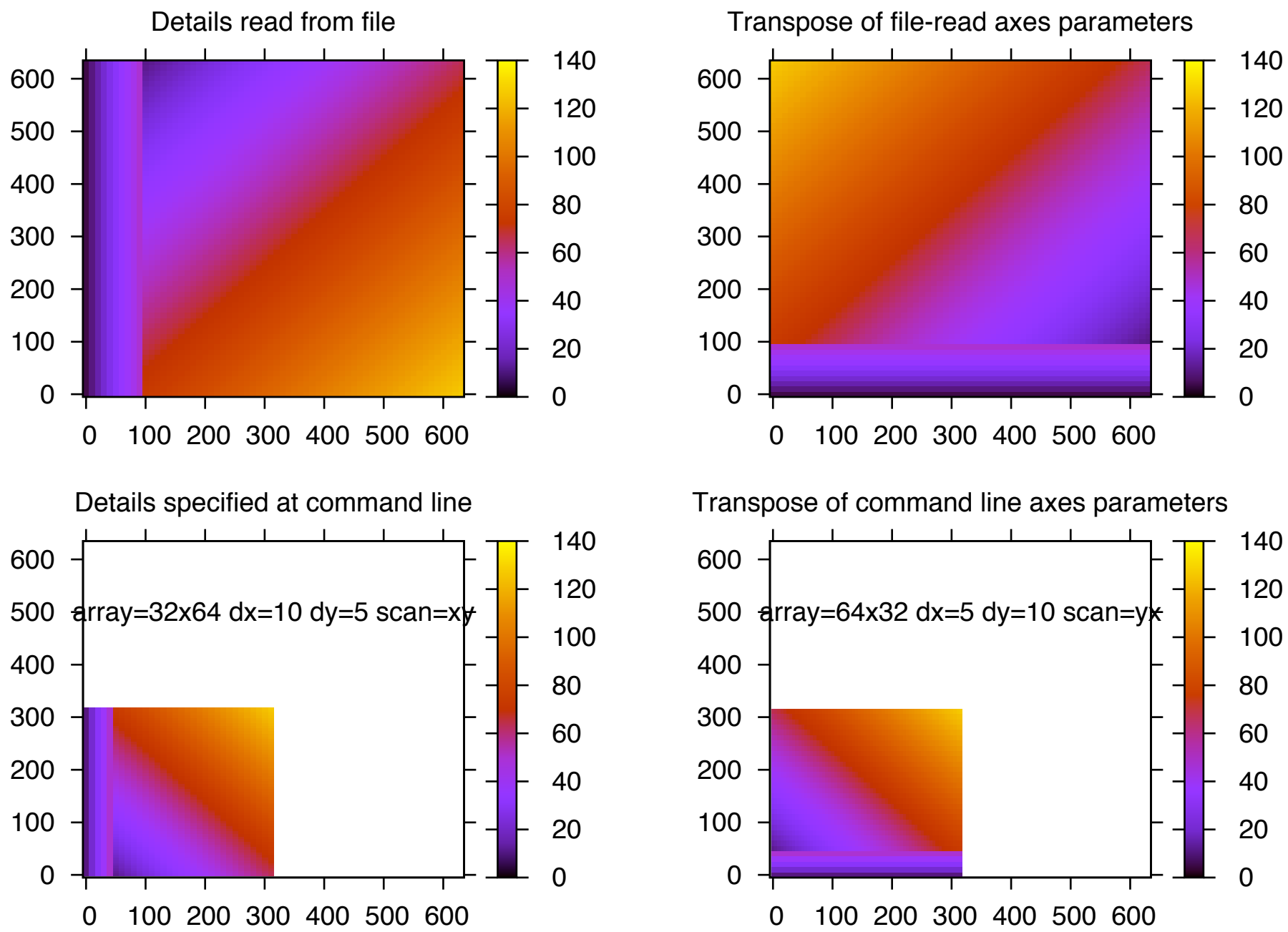

Uniform sampling in the polar coordinate system
'sine.bin' binary endian=little array=201 dt=0.018326 origin=(0,0) format='%f' using 1
1
0.5
0
0.5
1
1
0.5
0
0.5
1

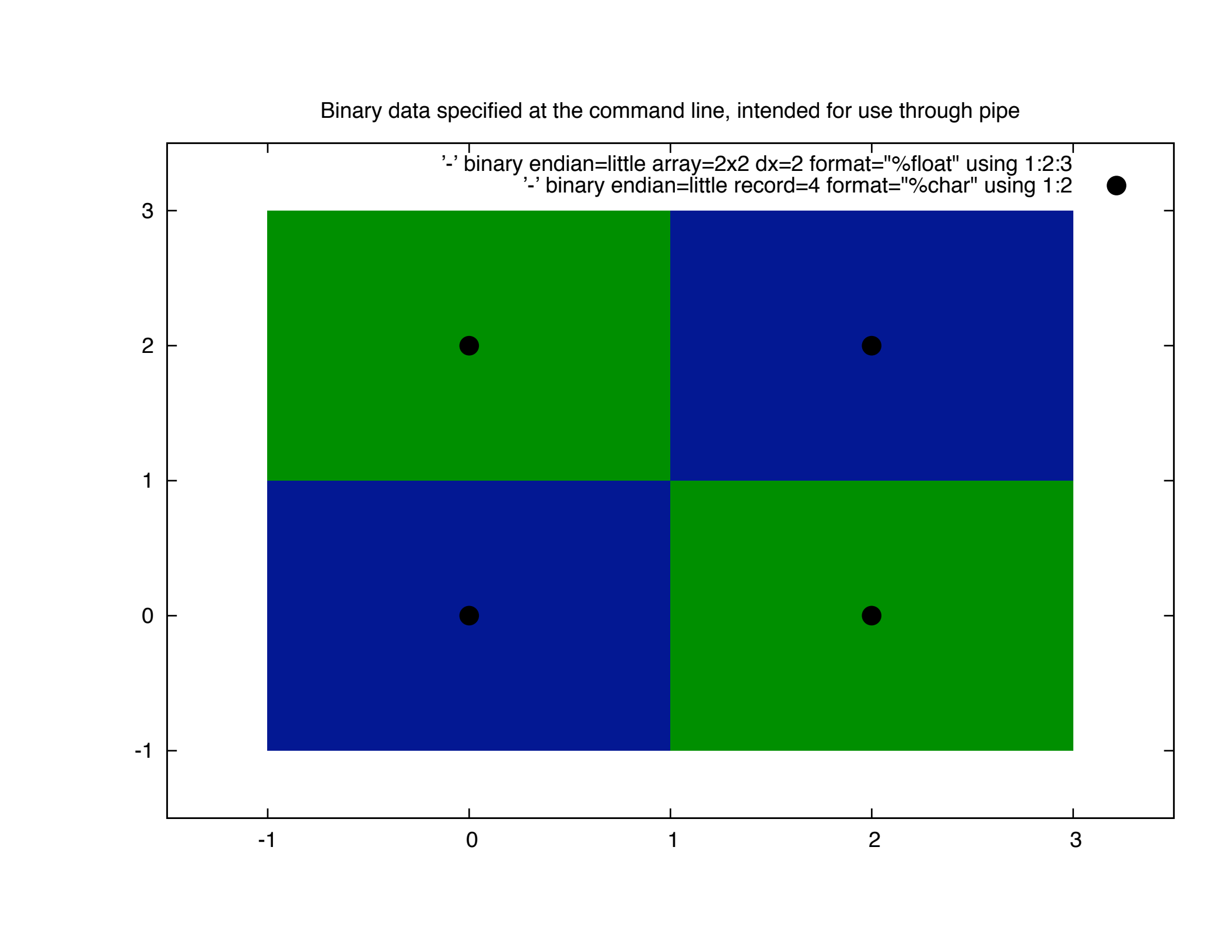

Decimation works on general binary data files as well.
Let Tux have his fun...
'blutux.rgb' binary array=128x128 flipy format='%uchar' every 1:1:43:15:83:65
"Can I do that?"
120
110
100
90
80
70
60
30
40
50
60
70
80
90

... Sure, go ahead.

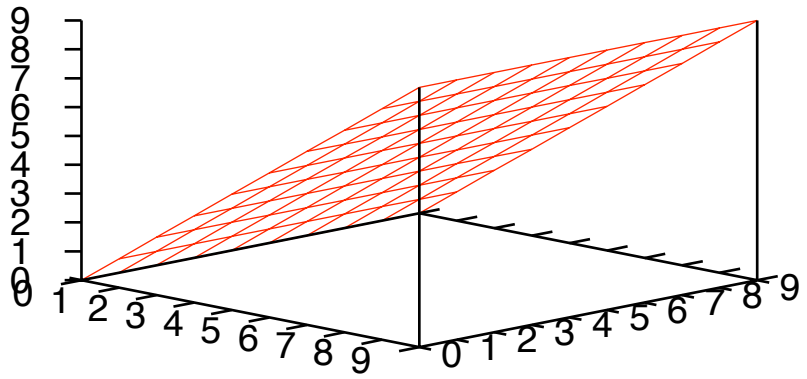



Automatically recognizing file type and extracting file information
'demo.edf' binary filetype=auto
700
600
500
400
300
200
100
0
-100
-100
0
100
200
300
400
500
600
700
140
120
100
80
60
40
20
0

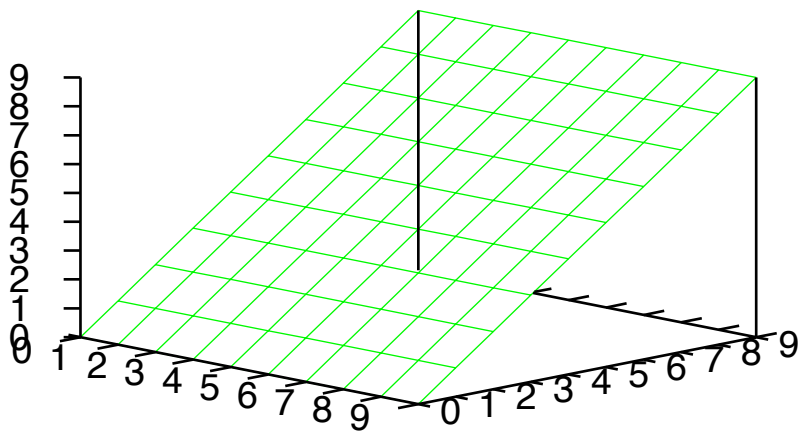

Details read from file
Transpose of file-read axes parameters
Details specified at command line
array=32x64 dx=10 dy=5 scan=xy
Transpose of command line axes parameters
array=64x32 dx=5 dy=10 scan=yx
0
100
200
300
400
500
600
20
40
60
80
120
140


Binary data specified at the command line, intended for use through pipe
'-' binary endian=little array=2x2 dx=2 format="%float" using 1:2:3
'-' binary endian=little record=4 format="%char" using 1:2
3
2
1
0
-1
-1
0
1
2
3

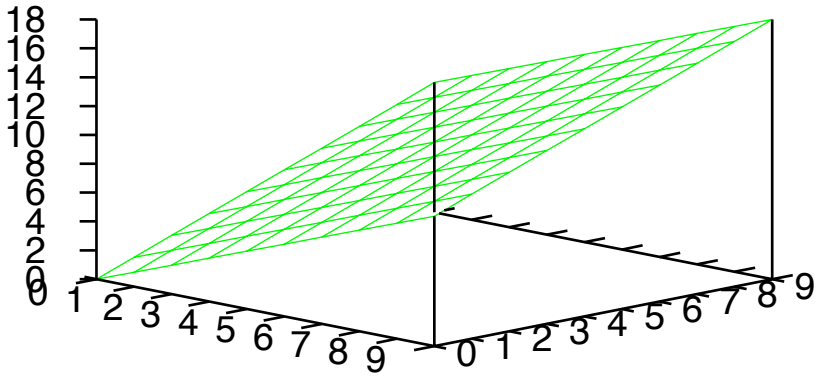
'asciimat.dat' matrix index 0



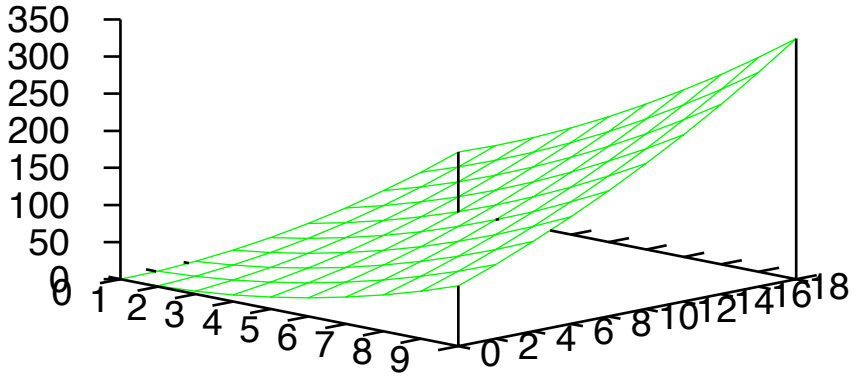
'asciimat.dat' matrix index 1

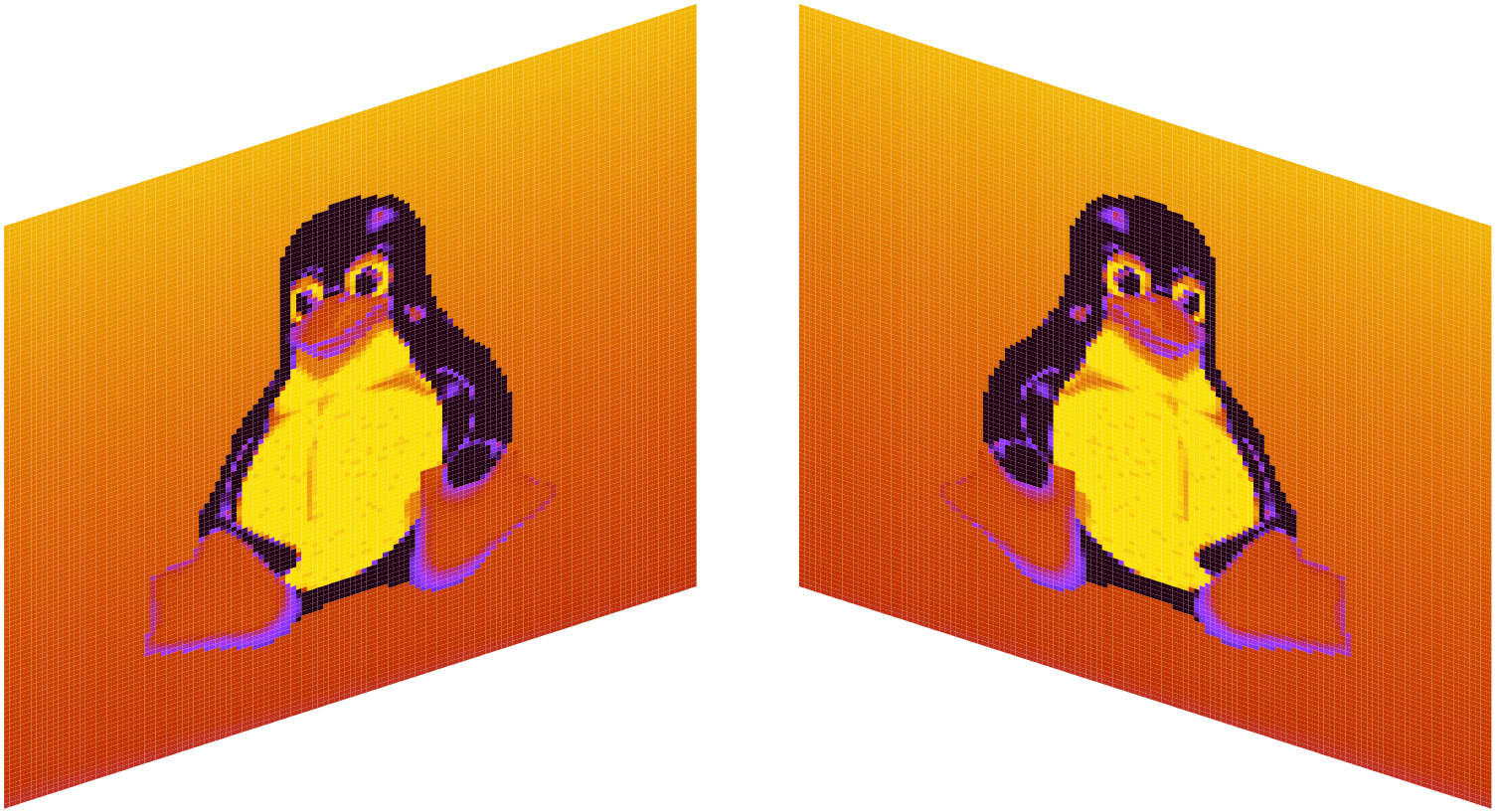


'asciimat.dat' matrix index 2



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





"Mirror mirror on the wall,
who’s the GNUiest penguin of all?"

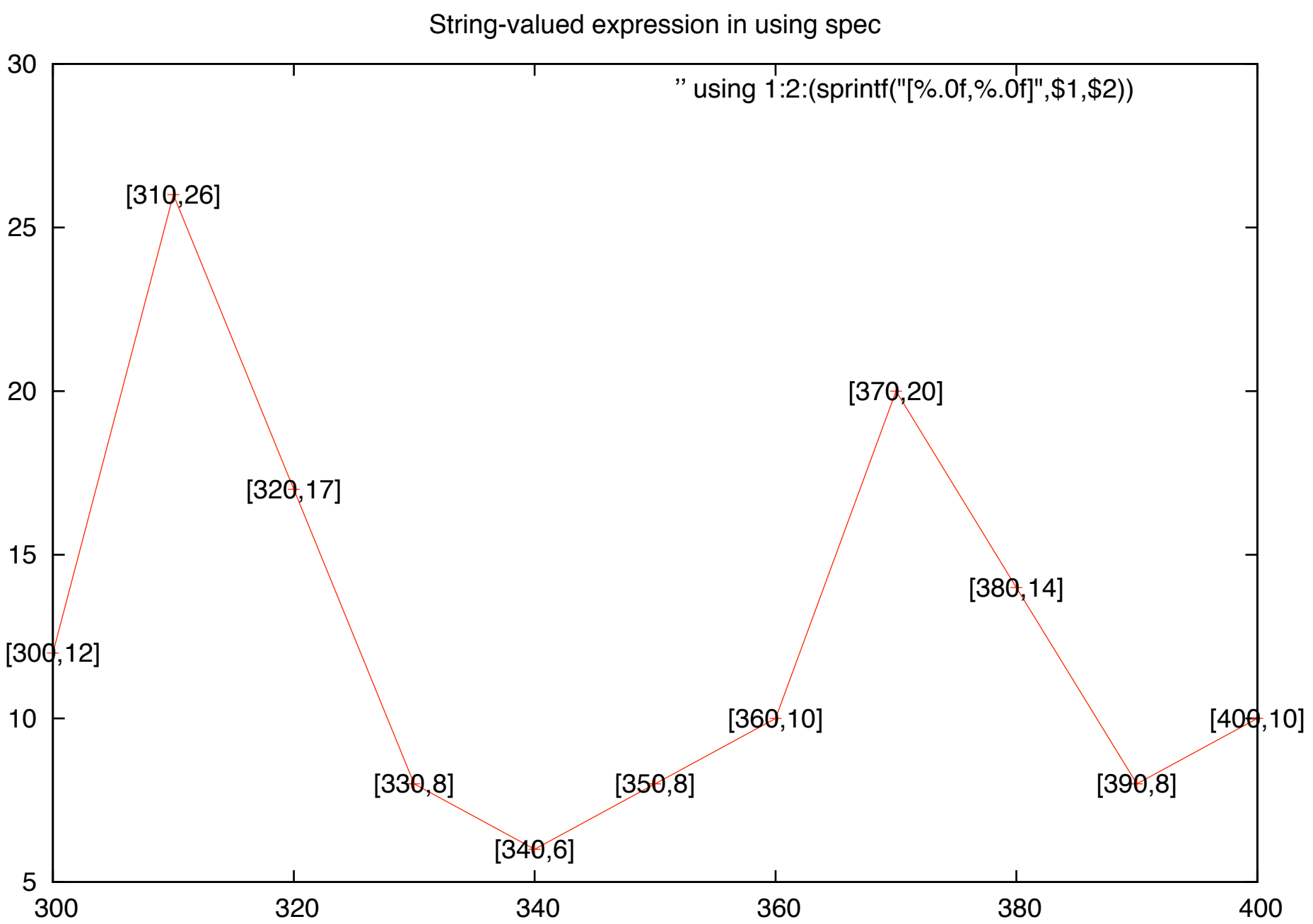
Exercise substring handling

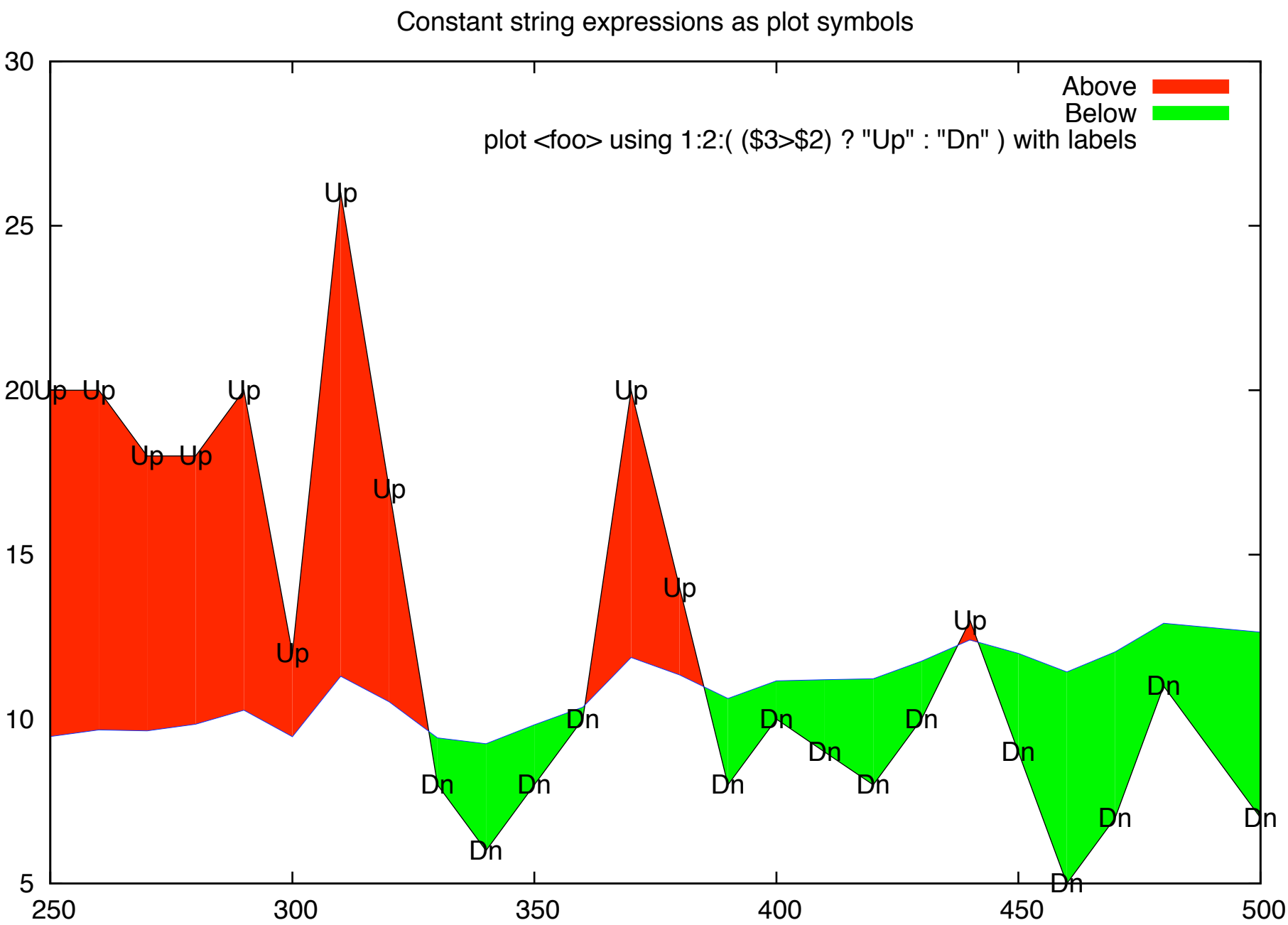
beg = 2 end = 4
foo = ABCDEF
foo[3:5] = CDE
foo[1:1] = A
foo[5:3] =
foo[beg:end] = BCD
foo[end:beg] =
foo[5:] = EF
foo[5:*] = EF
foo[:] = ABCDEF
foo[*:*] = ABCDEF
foo.foo[2:2] = ABCDEFB
(foo.foo)[2:2]= B

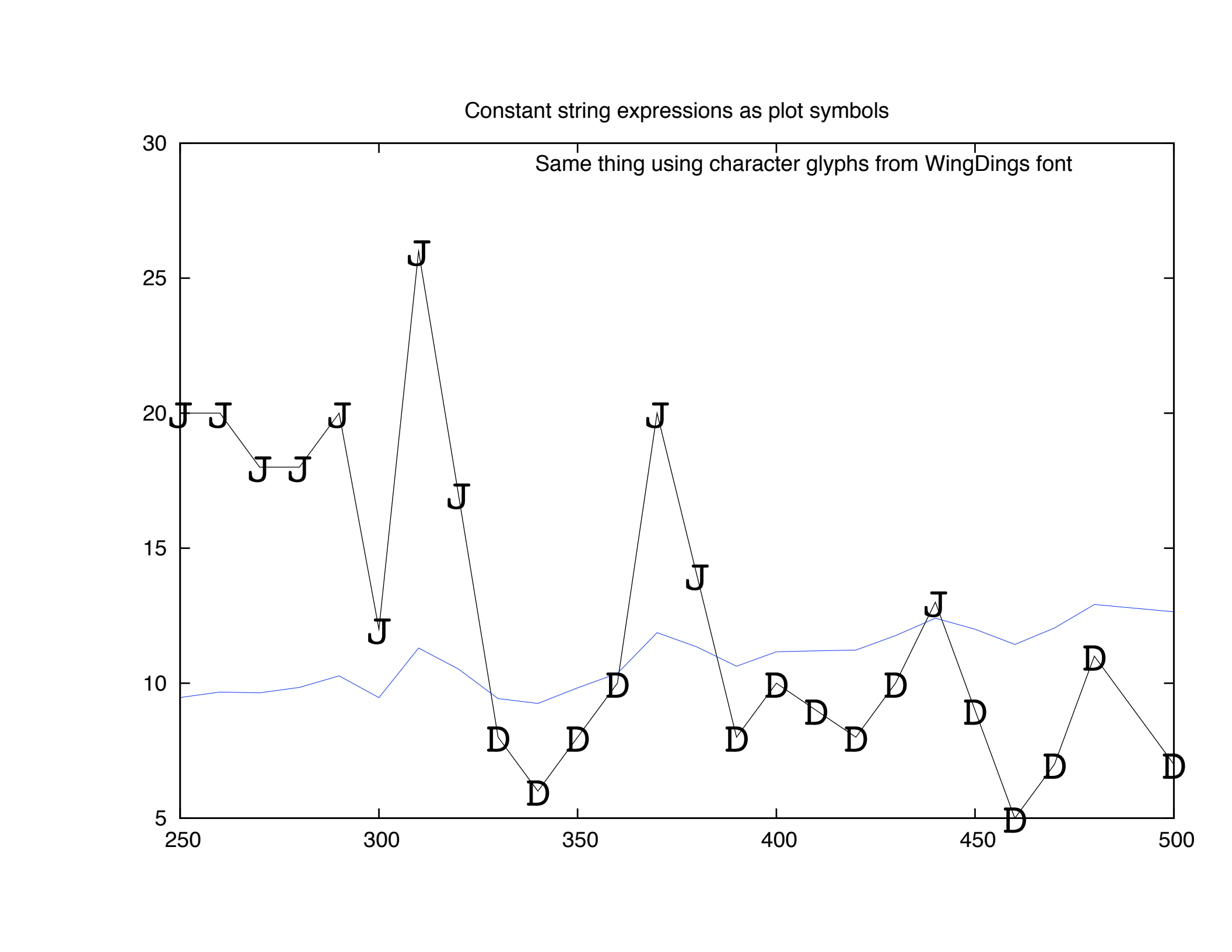
Exercise string handling functions

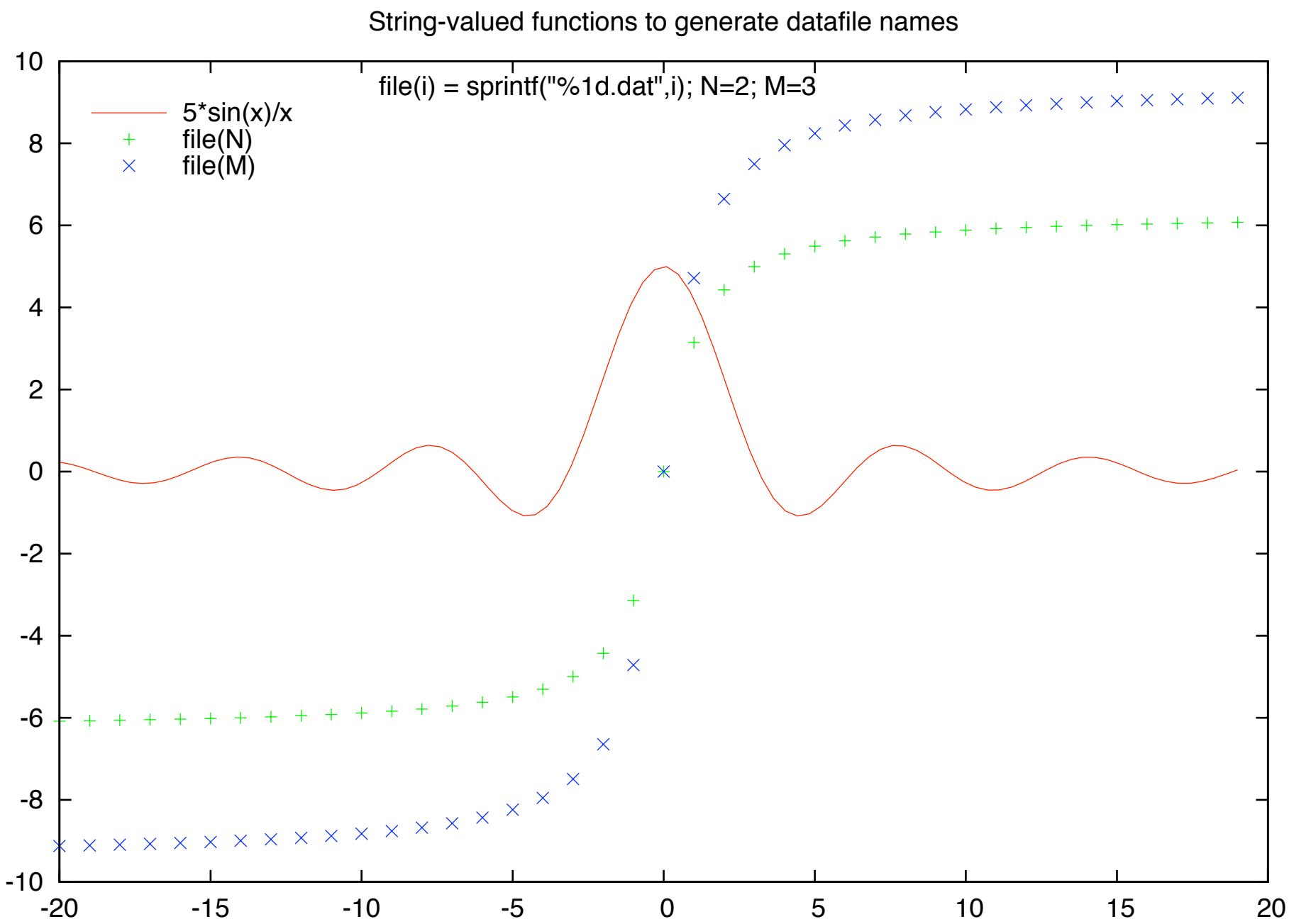
```
foo          = ABCDEF
strlen(foo)  = 6
substr(foo,3,4) = CD

haystack     = ‘date‘
haystack     = Wed Feb  4 12:18:59 EST 2009
needle       = :
S = strstrt(haystack,needle) = 14
haystack[S-2:S+2] = 12:18
It is now 12:18

words(haystack)   = 6
word(haystack,5)  = EST
```

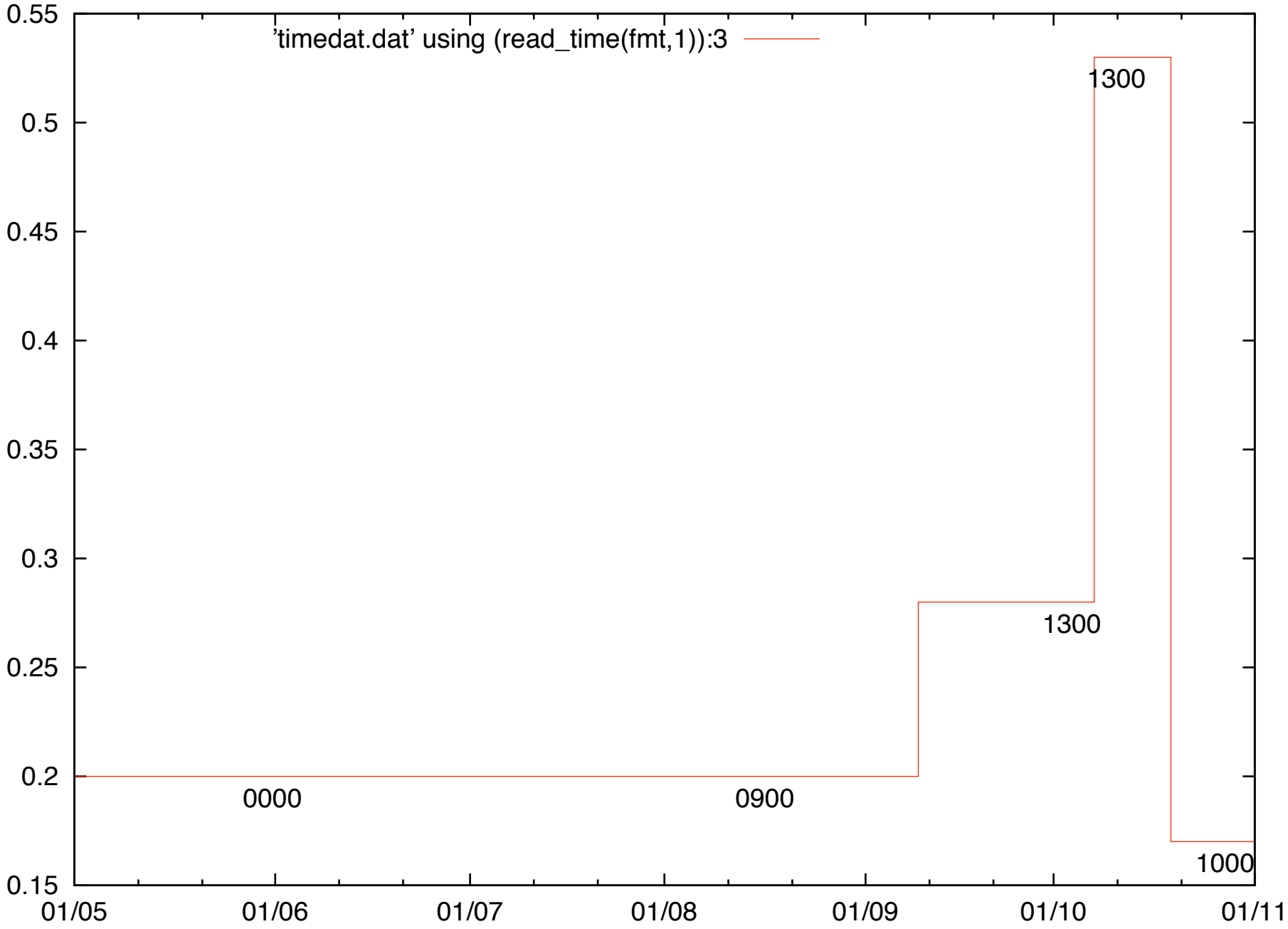
sprintf output of long strings works OK


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

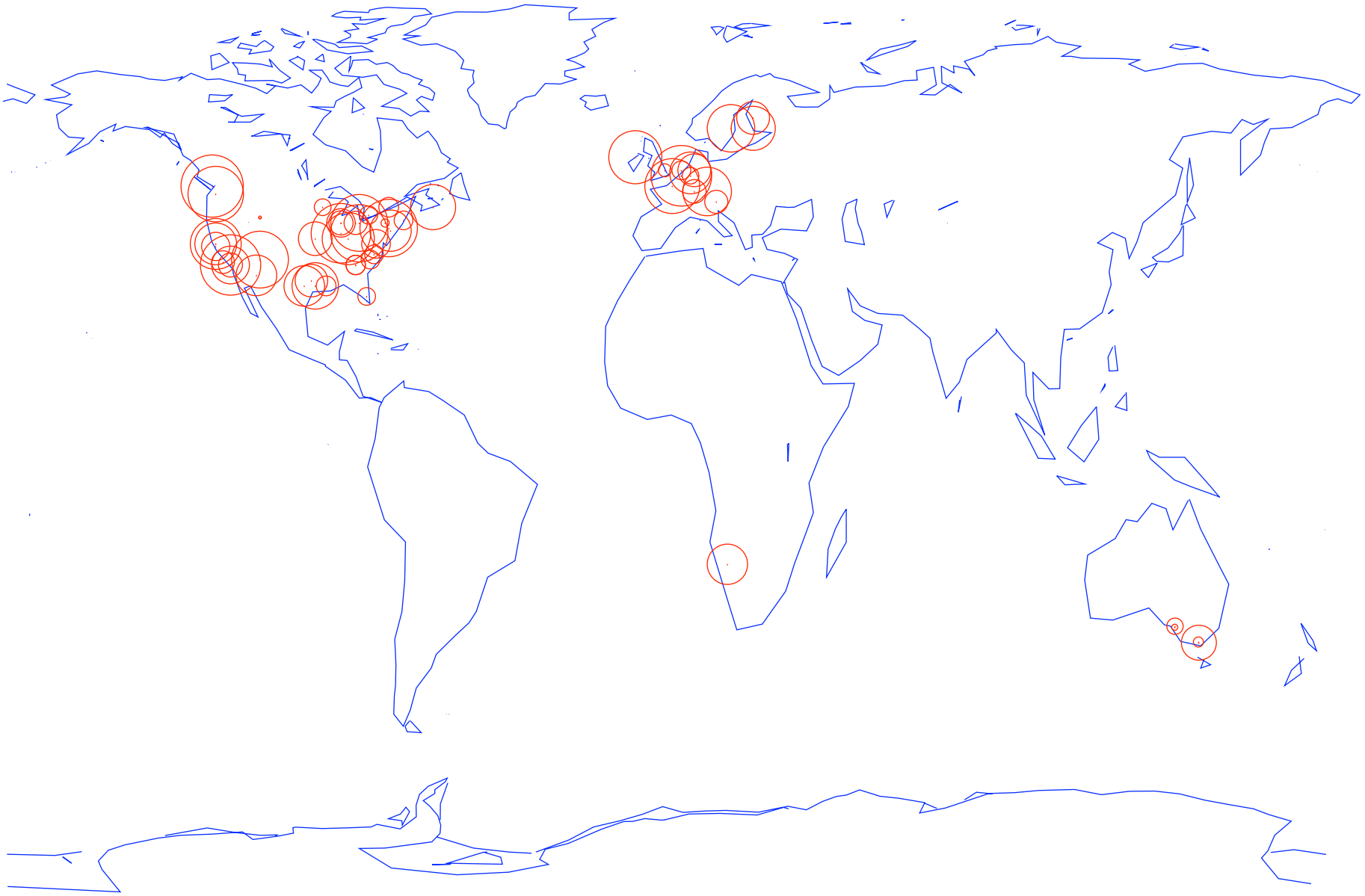

Constant string expressions as plot symbols
plot <foo> using 1:2:(($3>$2) ? "Up" : "Dn") with labels
Above
Below
30
25
20
15
10
5
250
300
350
400
450
500
Up
Up
Up
Up
Up
Up
Up
Up
Up
Up
Dn
Dn
Dn
Dn
Dn
Dn
Dn
Dn
Dn
Dn
Dn
Dn

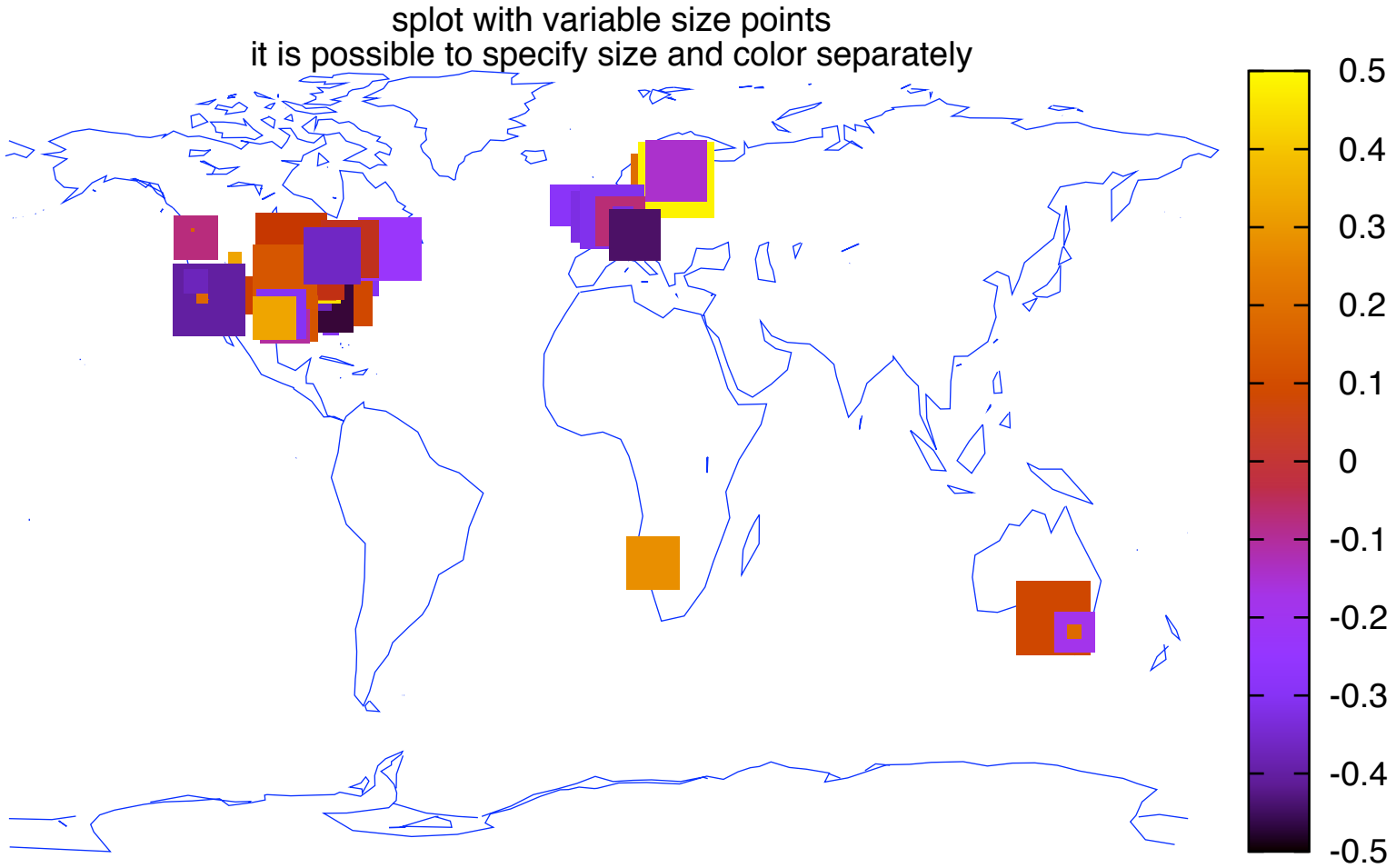

Constant string expressions as plot symbols
Same thing using character glyphs from WingDings font
30
25
20
15
10
5
250
300
350
400
450
500


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

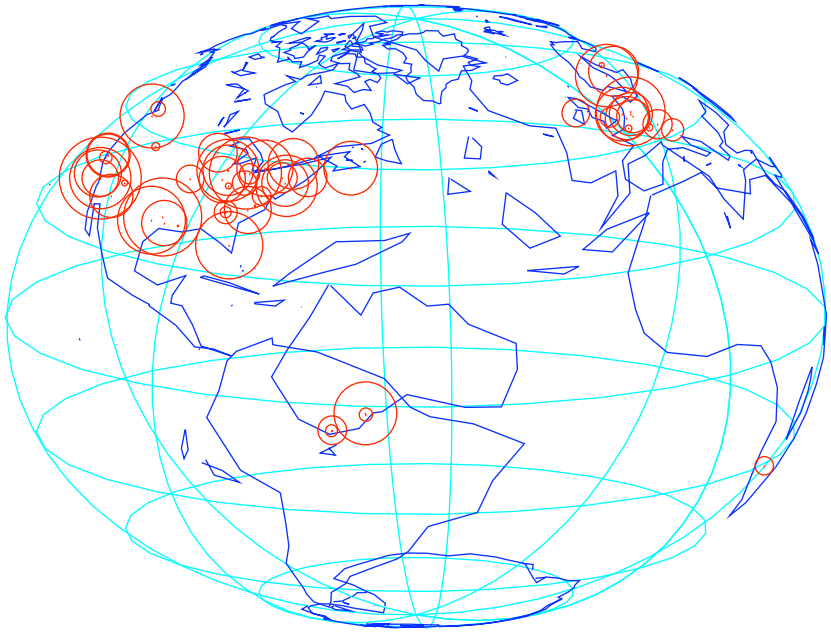

'timedat.dat' using (read_time(fmt,1)):3
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
01/05
01/06
01/07
01/08
01/09
01/10
01/11
0000
0900
1300
1300
1000

plot with variable size points

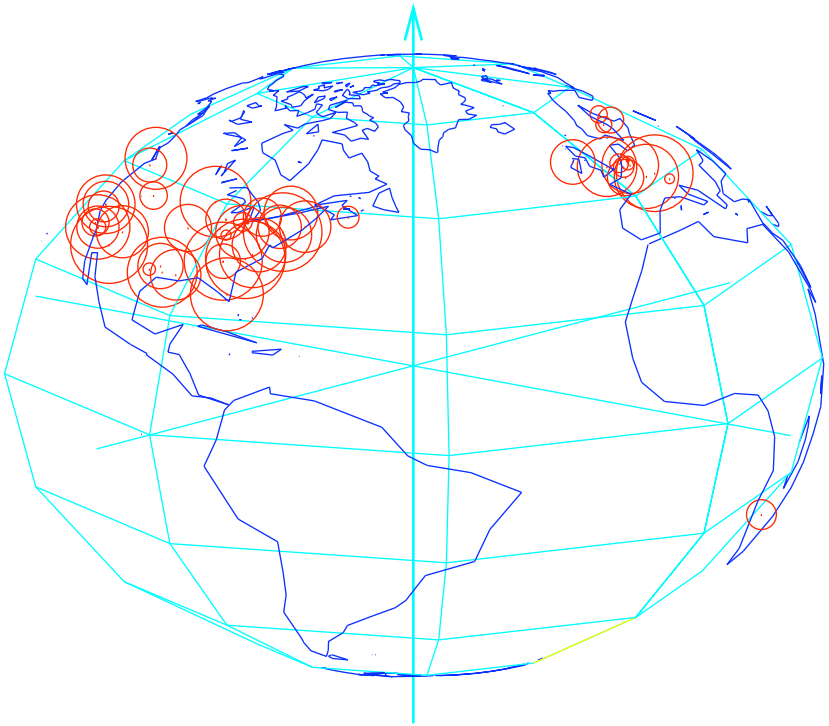


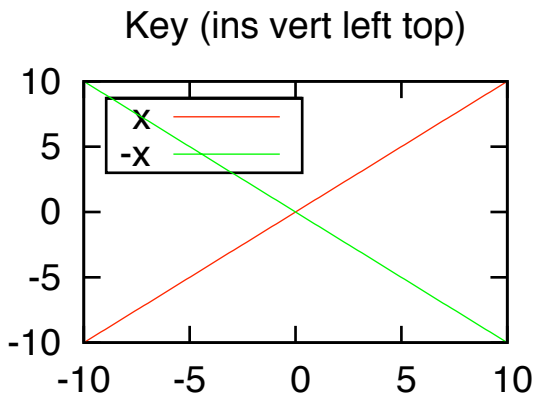

splot with variable size points
it is possible to specify size and color separately
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5

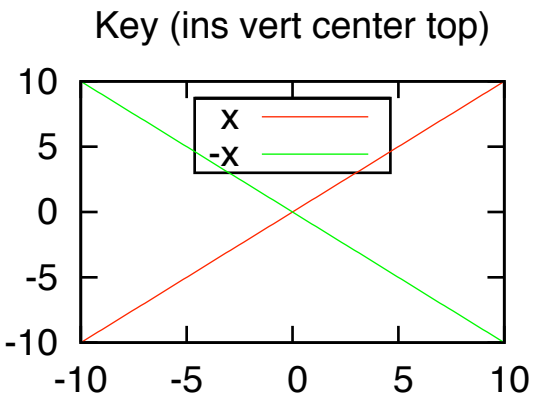
3D version using spherical coordinate system

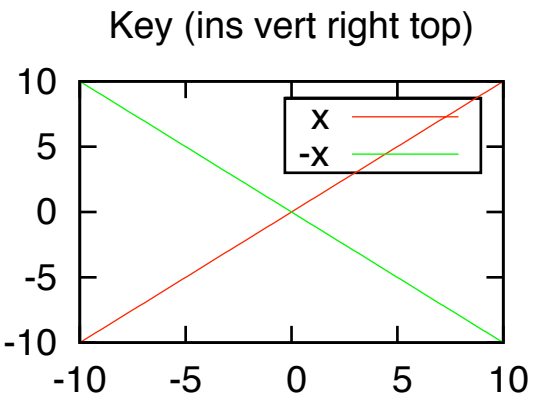


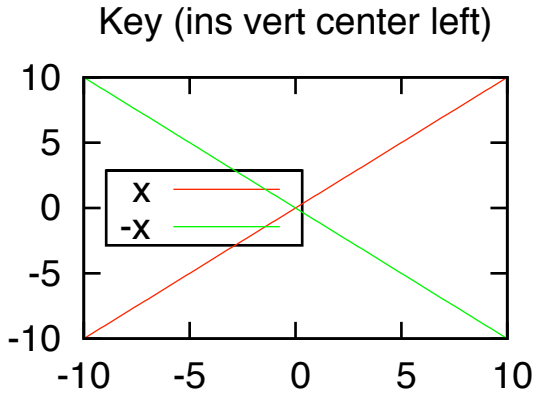
3D solid version through hiddenlining

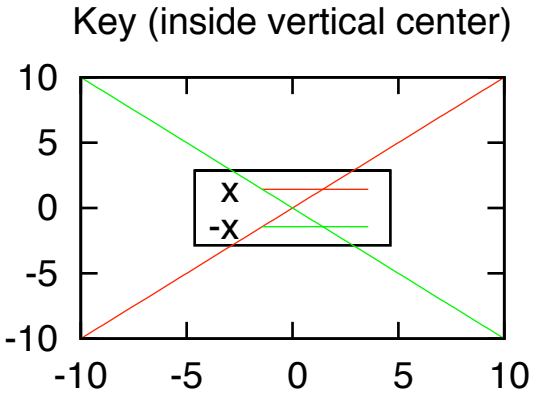


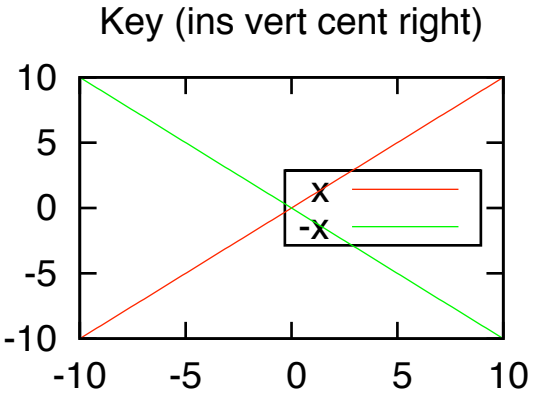

Key (ins vert left top)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

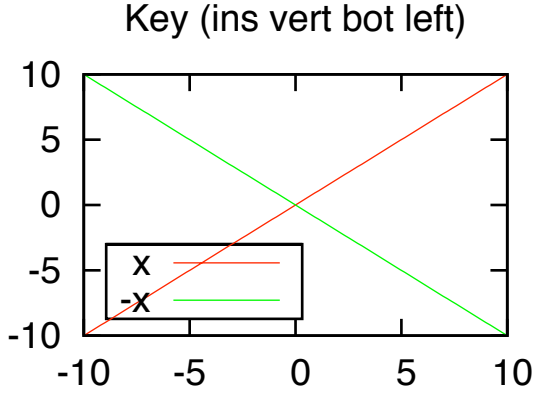

Key (ins vert center top)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

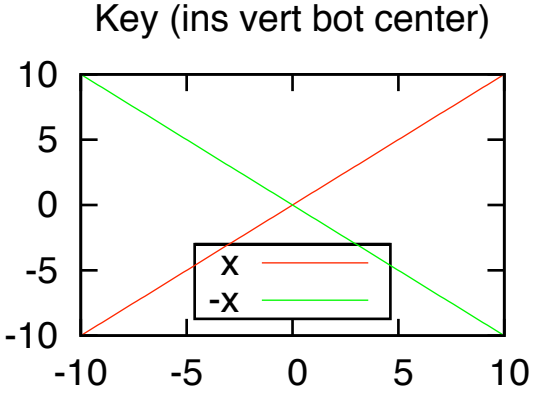

Key (ins vert right top)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

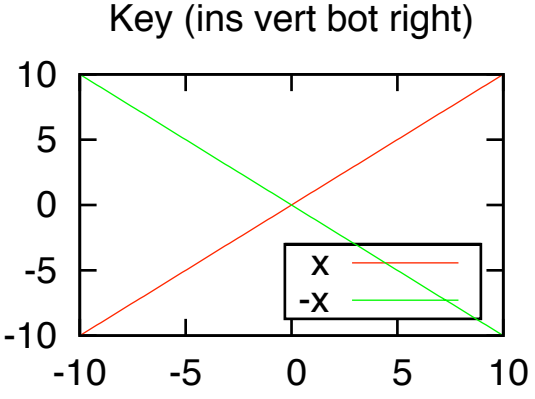

Key (ins vert center left)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

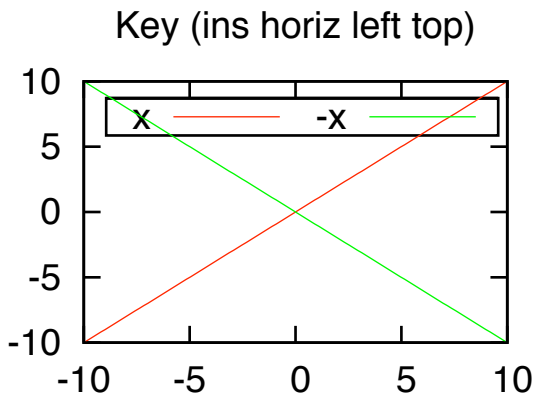

Key (inside vertical center)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

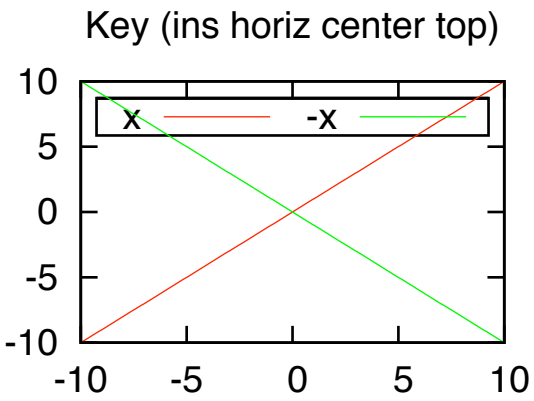

Key (ins vert cent right)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

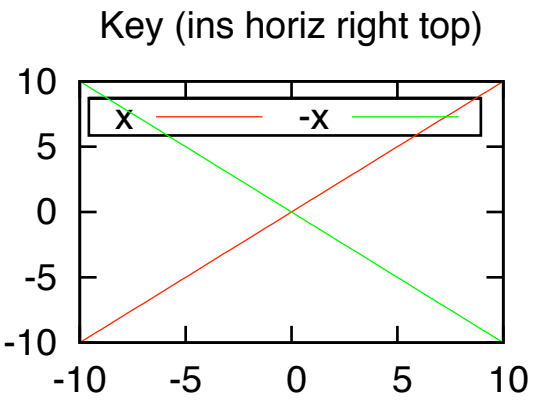

Key (ins vert bot left)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

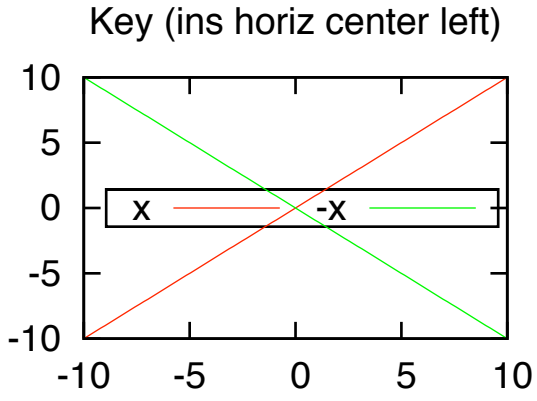

Key (ins vert bot center)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

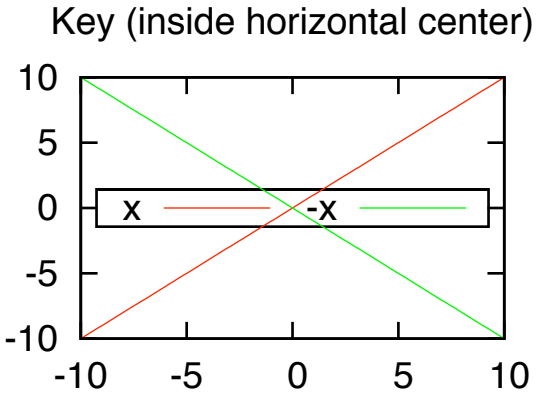

Key (ins vert bot right)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

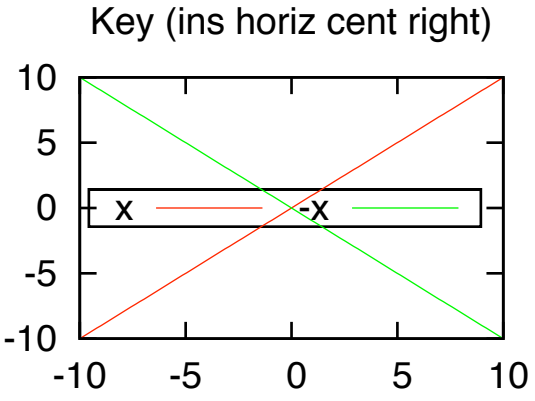

Key (ins horiz left top)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

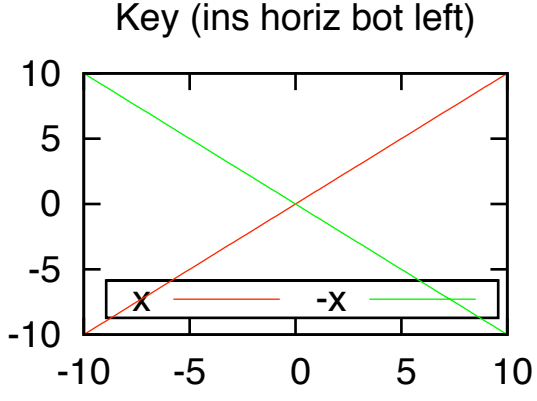

Key (ins horiz center top)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

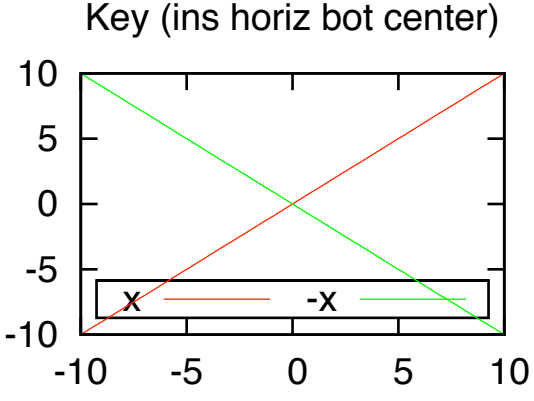

Key (ins horiz right top)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

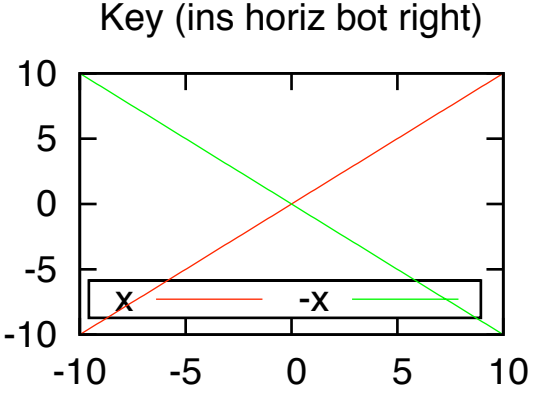

Key (ins horiz center left)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

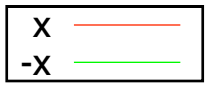

Key (inside horizontal center)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10


Key (ins horiz cent right)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

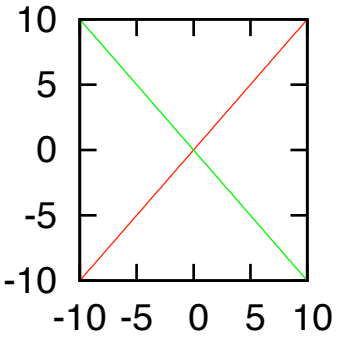

Key (ins horiz bot left)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

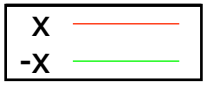

Key (ins horiz bot center)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10


Key (ins horiz bot right)
x
-x
10
5
0
-5
-10
-10
-5
0
5
10

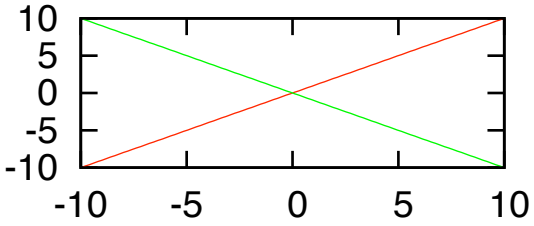


Key (out vert left top)

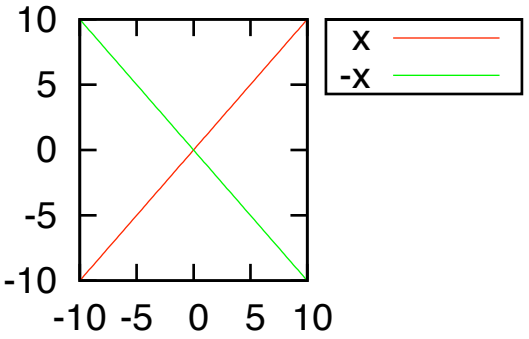


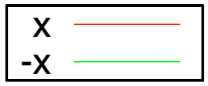


Key (out vert center top)

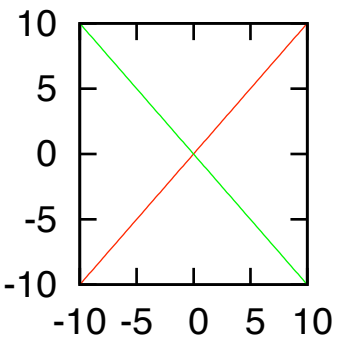


Key (out vert right top)

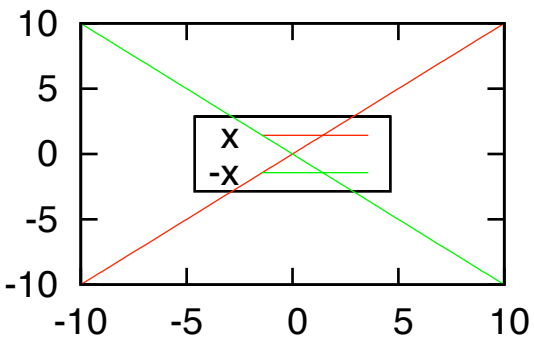




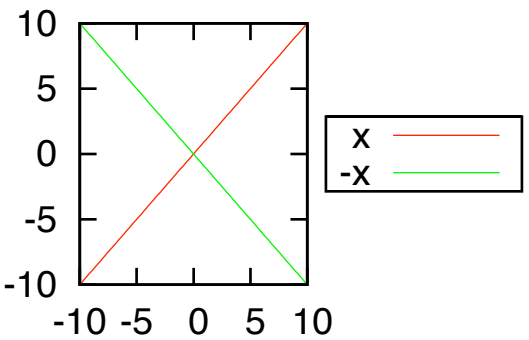
Key (out vert center left)

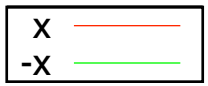


Key (outside vertical center)

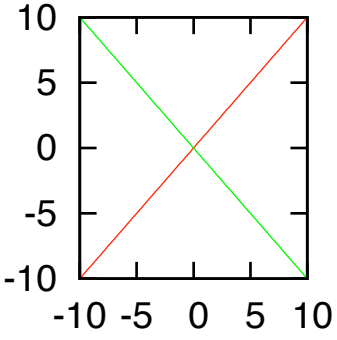


Key (out vert cent right)

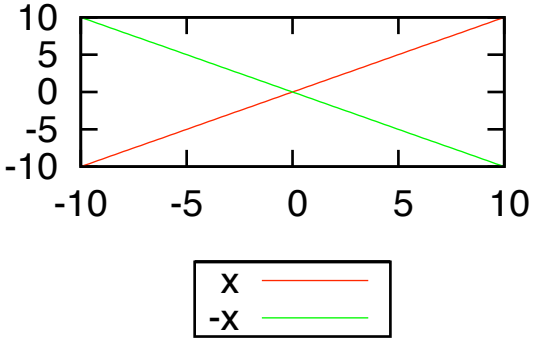




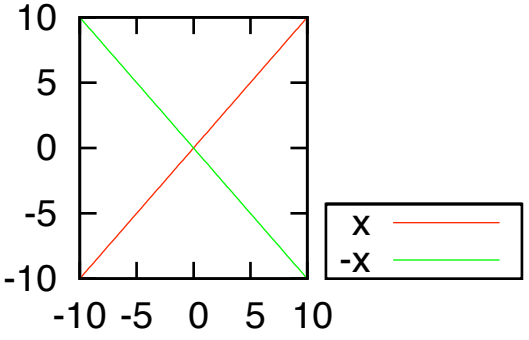
Key (out vert bot left)

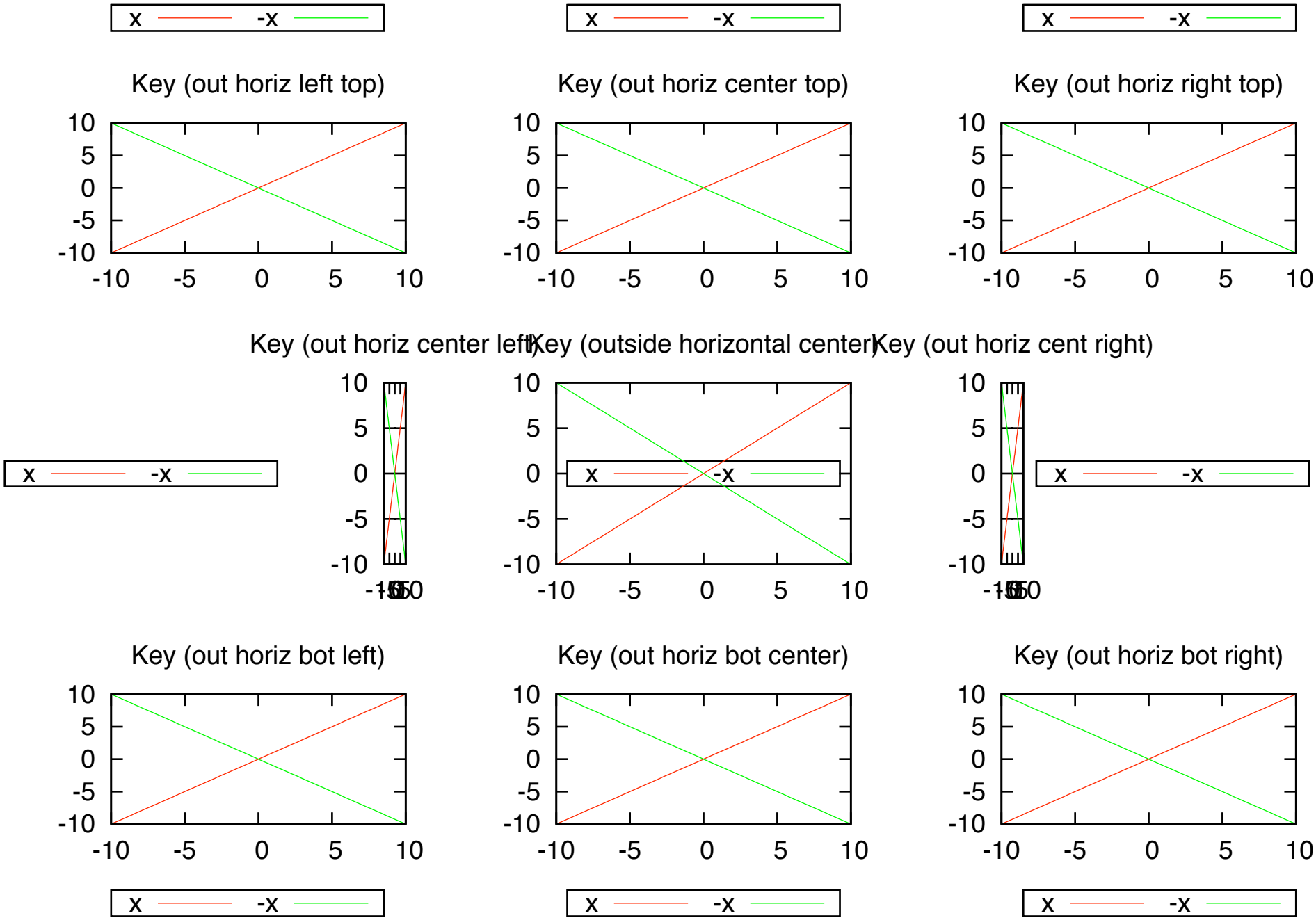


Key (out vert bot center)

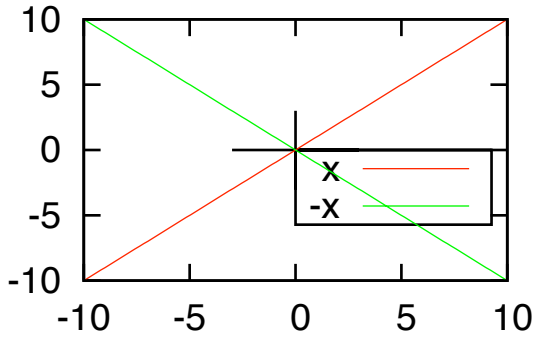


Key (out vert bot right)

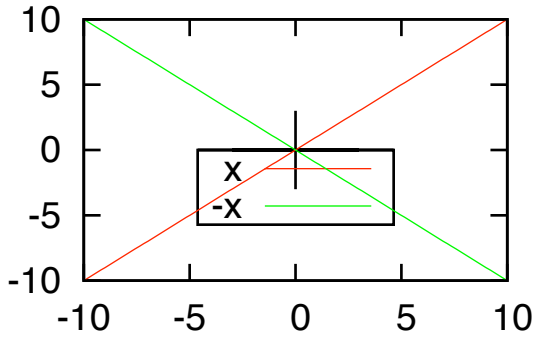



x
-x
Key (out horiz left top)
Key (out horiz center top)
Key (out horiz right top)
Key (out horiz center left)
Key (outside horizontal center)
Key (out horiz cent right)
Key (out horiz bot left)
Key (out horiz bot center)
Key (out horiz bot right)
10
5
0
-5
-10

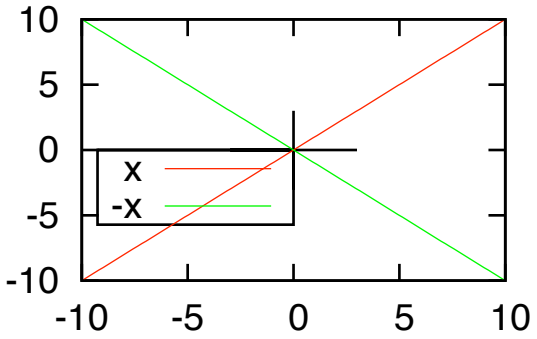
Key (<manual> vert left top)



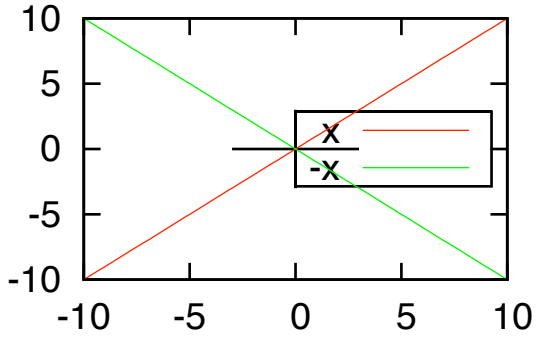
Key (<manual> vert center top)



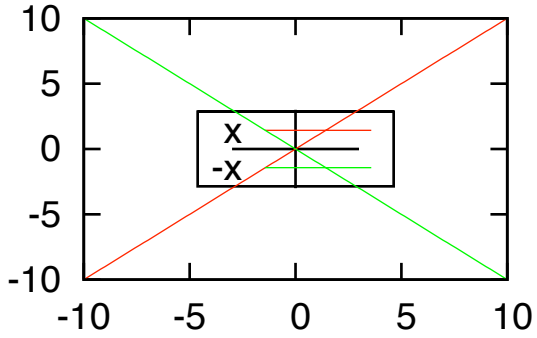
Key (<manual> vert right top)



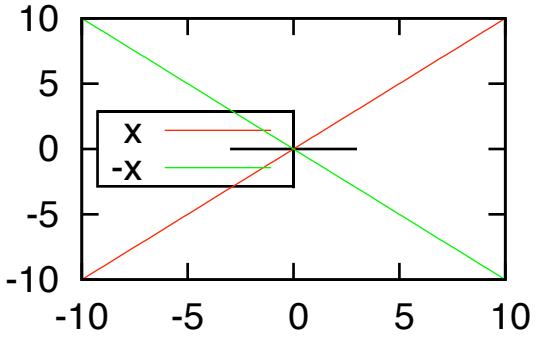
Key (<manual> vert center left)



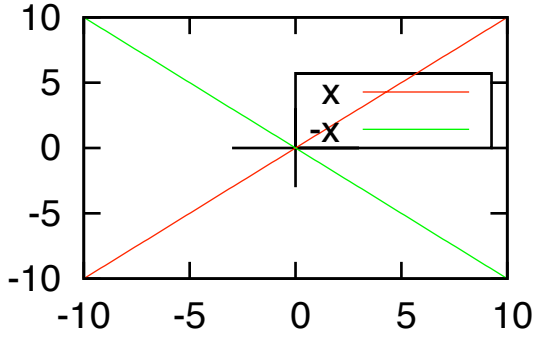
Key (<manual> vertical center)



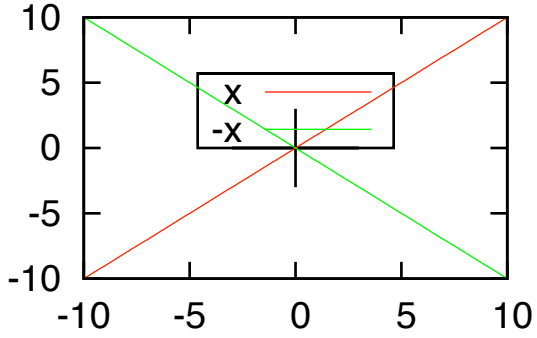
Key (<manual> vert cent right)



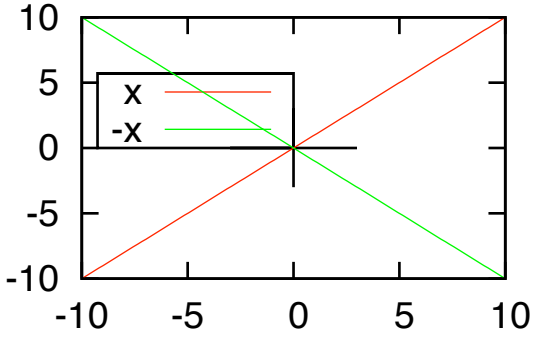
Key (<manual> vert bot left)



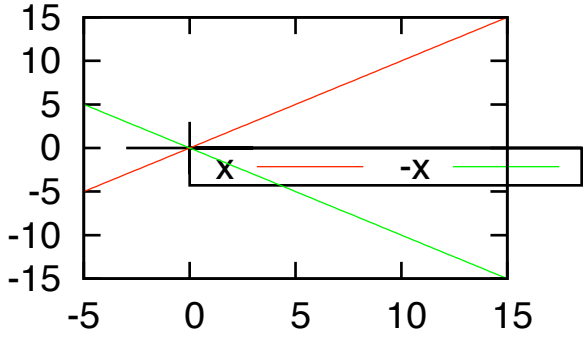
Key (<manual> vert bot center)



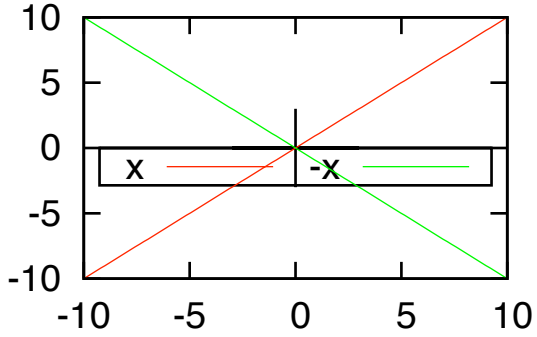
Key (<manual> vert bot right)



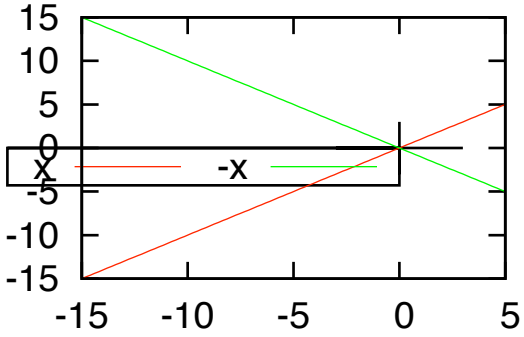
Key (<manual> horiz left top)



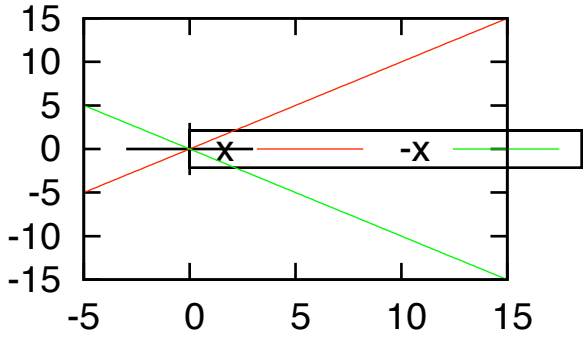
Key (<manual> horiz center top)



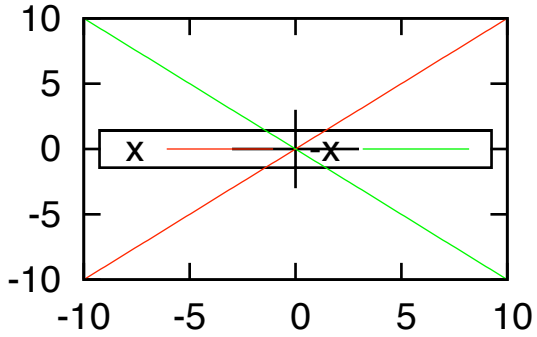
Key (<manual> horiz right top)



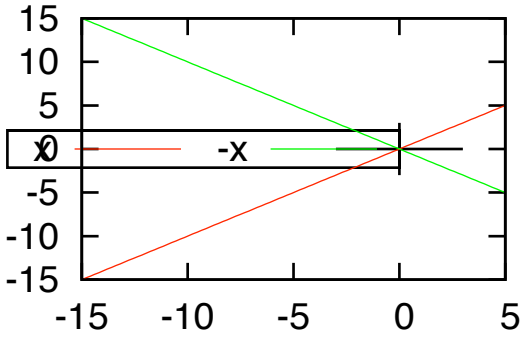
Key (<manual> horiz center left)



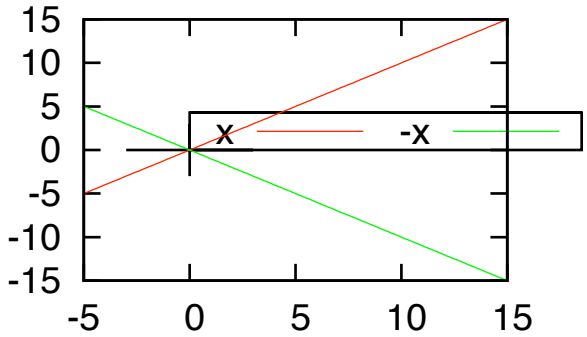
Key (<manual> horizontal center)



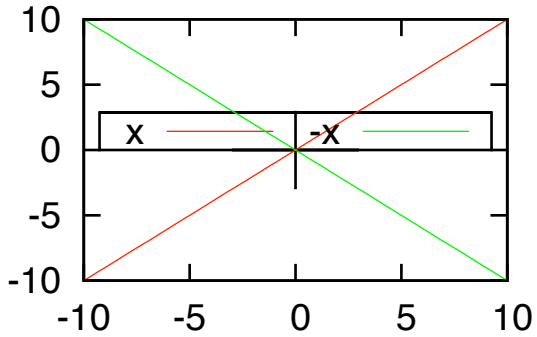
Key (<manual> horiz cent right)



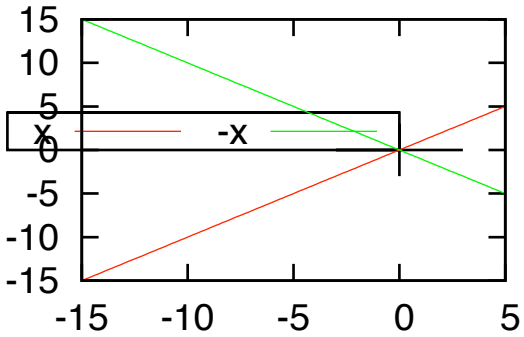
Key (<manual> horiz bot left)

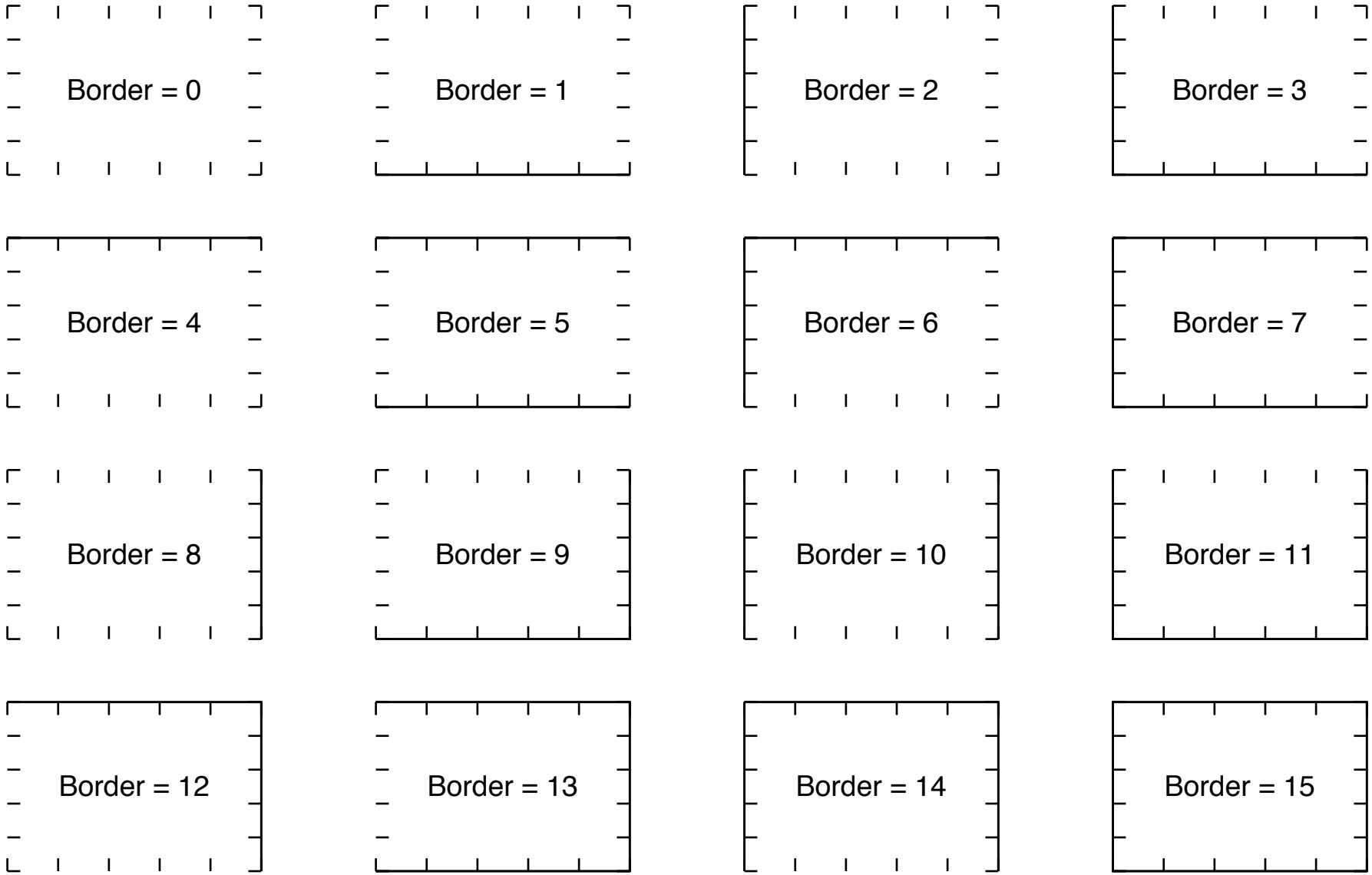


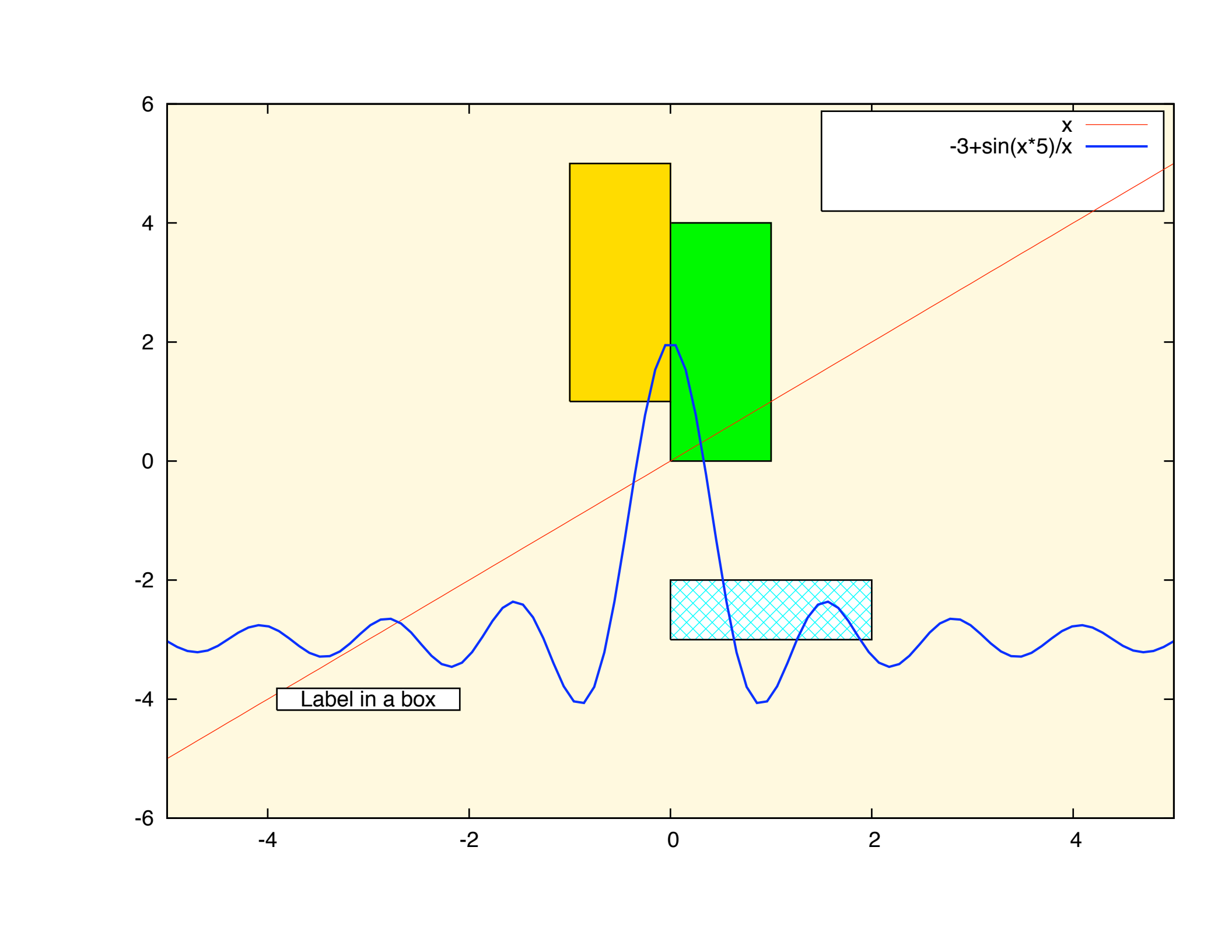
Key (<manual> horiz bot center)

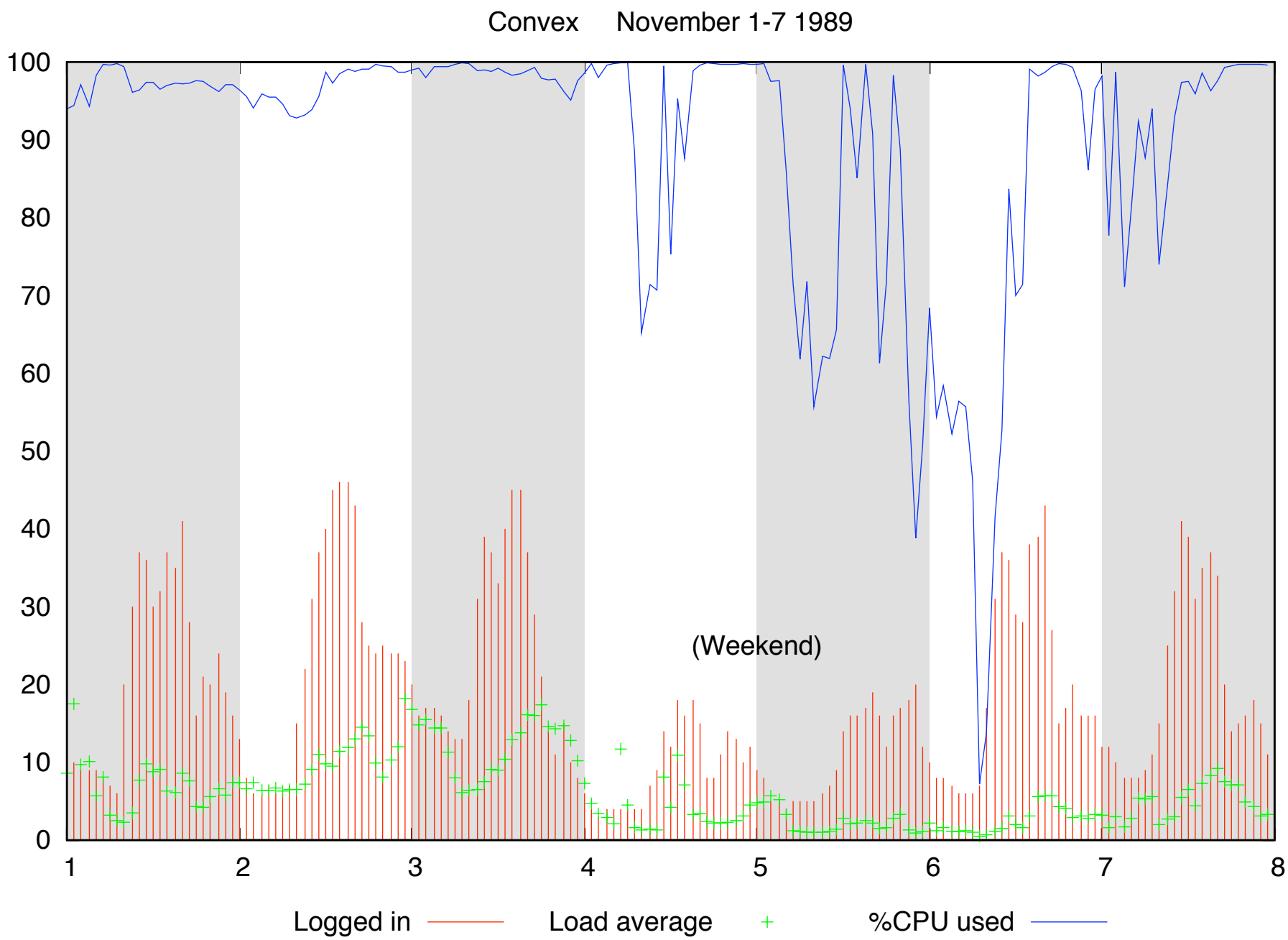


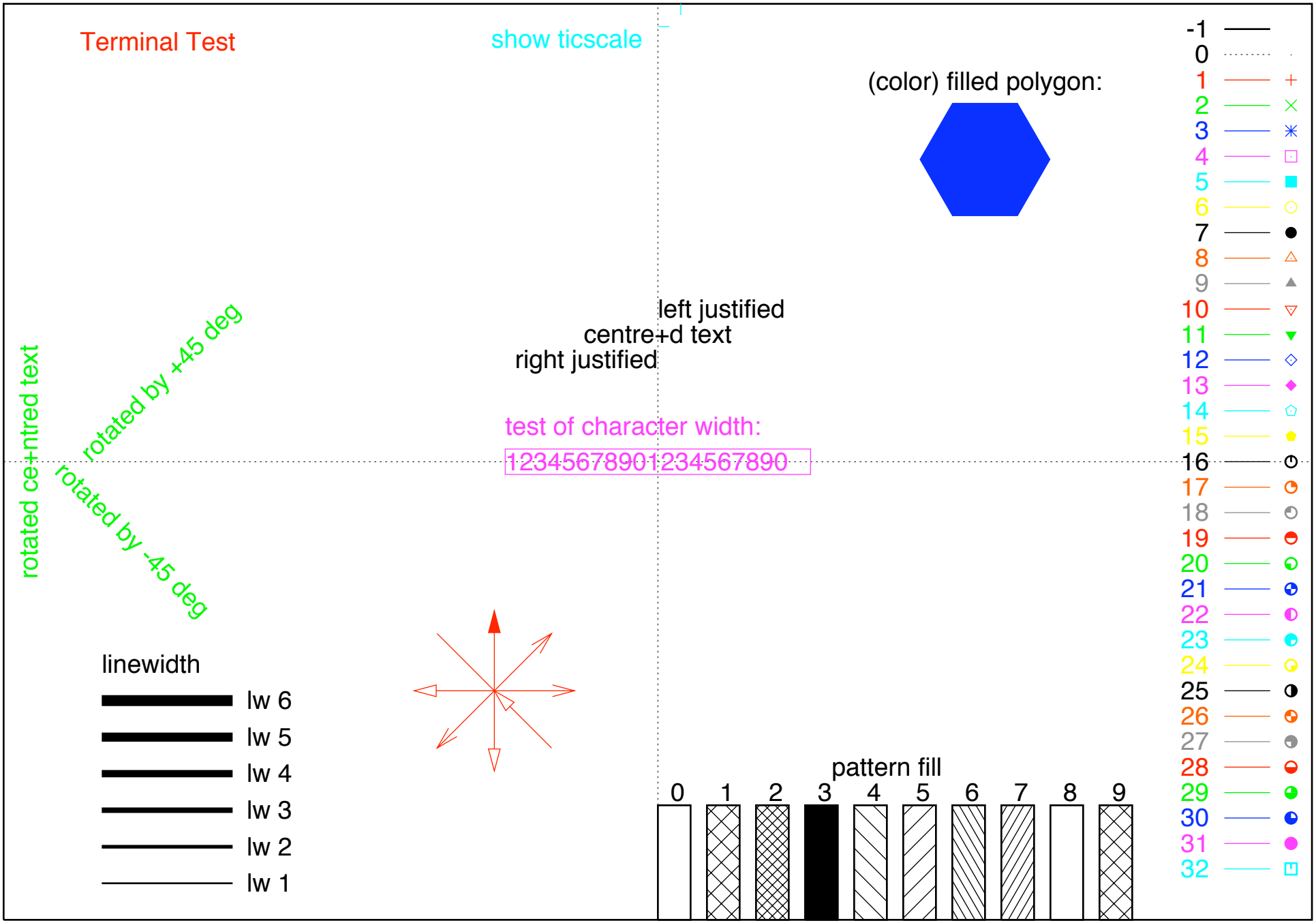
Key (<manual> horiz bot right)




Demonstration of different border settings
and also of a reread loop
Border = 0
Border = 1
Border = 2
Border = 3
Border = 4
Border = 5
Border = 6
Border = 7
Border = 8
Border = 9
Border = 10
Border = 11
Border = 12
Border = 13
Border = 14
Border = 15


x
-3+sin(x*5)/x
Label in a box
6
4
2
0
-2
-4
-6
-4
-2
0
2
4


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


Terminal Test
show ticscale
(color) filled polygon:
left justified
centre+d text
right justified
test of character width:
12345678901234567890
rotated by +45 deg
rotated by -45 deg
rotated ce+ntred text
linewidth
lw 6
lw 5
lw 4
lw 3
lw 2
lw 1
pattern fill
0 1 2 3 4 5 6 7 8 9
-1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32