$\sin(x)$

$\tan(x)$

$\cos(\tan(x))$
\[-(\sin(x) > \sin(x+1) \ ? \ \sin(x) : \ \sin(x+1))\]
Mutual Characteristic of a Transistor

$I_c (V_{be})$ vs $V_{be}$ (base-emitter voltage)
JFET Mutual Characteristic

Drain current $I_d$ (mA) vs. Drain voltage $V_d$ (V)
Amplitude and Phase Frequency Response

- **Magnitude of A(jw)**
- **Phase of A(jw) (degrees)**

Graph showing the amplitude and phase response of A(jw) as a function of jw (radians). The amplitude is plotted on the y-axis, and the phase is plotted on the right y-axis. The x-axis represents jw (radians). The graph includes two curves: one for the magnitude (abs(A(jw))) and another for the phase (180/π*arg(A(jw))).
A demonstration of boxes with default properties
A demonstration of boxes with style fill solid 1.0
A demonstration of boxes with style fill solid border -1
Filled boxes of reduced width
Filled boxes at 50% fill density
A demonstration of boxes with style fill solid 0.25 noborder
A demonstration of boxes in mono with style fill pattern
plot with filledcurve [options]

1.5+sin(x)/x
sin(x)/x
1+sin(x)/x
-1+sin(x)/x
-2.5+sin(x)/x
-4.3+sin(x)/x
(x>3.5 ? x/3-3 : 1/0)
Intersection of two parabolas

$x^2$

$50-x^2$
Filled sinus and cosinus curves

\[ 2 + \sin(x)^2 \]
\[ \cos(x)^2 \]
The red bat: \( |x| \) with filledcurve \( xy=2,5 \)
Some sqrt stripes on filled graph background

-8
sqrt(x)
sqrt(10-x)-4.5
Let’s smile with parametric filled curves
Fill area between two curves (pattern fill)

'silver.dat' u 1:2:3
Fill area between two curves (above/below)

Above
Below
curve 1
curve 2
candlesticks with specified boxwidth
candlesticks with style fill solid

'candlesticks.dat' using 1:3:2:6:5
box-and-whisker plot adding median value as bar

Quartiles
box-and-whisker with median bar and whiskerbars
Bragg reflection -- Peak only

Amplitude

Angle (deg)

Rate

Rate
Ag 108 decay data

Rate vs. Time (sec)

- Experimental
- Cubic Smooth
Ag 108 decay data

![Graph showing Ag 108 decay data with rate on the y-axis and time (sec) on the x-axis. The graph includes two lines: one for experimental data and another for acspline Y/Z. The experimental data shows a decaying trend with error bars, while the acspline Y/Z line is smoother and follows the trend of the experimental data.](image-url)
Ag 108 decay data

Rate vs. Time (sec)

Data points and curves represent different functions:
- rate
- acspline \( \frac{Y}{Z \times 1.0} \)
- \( \frac{Y}{Z \times 1.0e3} \)
- \( \frac{Y}{Z \times 1.0e5} \)
Ag 108 decay data

Rate vs. Time (sec)

- Rate
- acspline Y/(Z*1.e1)
- Y/(Z*1.e3)
- Y/(Z*1.e5)
Ag 108 decay data
Ag 108 decay data

Rate vs. Time (sec)
UM1-Cell Power

Power [W] vs. Resistance [Ohm]

- **Power**
- **Theory**

The graph shows the relationship between power and resistance for UM1 batteries, with a theoretical line also plotted.
The graph illustrates the relationship between power [W] and resistance [Ohm] for UM1 cells, labeled as 'UM1-Cell Power'. The x-axis represents the resistance in Ohm, ranging from 0 to 50, while the y-axis shows the power in Watts, ranging from 0 to 0.12. The plot includes a theoretical curve and data points for power output at various resistances.
UM1-Cell Power

Power [W] vs. Resistance [Ohm]

- **Power**
- **Theory**

Graph showing the power output in watts as a function of resistance in ohms.
UM1-Cell Power

Power [W] vs. Resistance [Ohm]

The graph shows the relationship between power and resistance for UM1 cells. The data points represent experimental measurements, while the line represents the theoretical prediction.

- **Power** line indicates the experimental data points with error bars.
- **Theory** line represents the theoretical curve fitting the data.
UM1-Cell Power

Power [W] vs. Resistance [Ohm]

- Power
- Theory

The graph shows the relationship between power and resistance for UM1 cells, with a peak power output at a certain resistance value.
data for first fit demo

'lcddemo.dat'
Density [g/cm³]

Temperature T [deg Cels.]

'all fit params set to 0'

'lcdemo.dat' + I(x)
Density [g/cm³]

Temperature T [deg Cels.]

unweighted fit

'lcdemo.dat'

l(x)
fit weighted towards low temperatures
Density \([\text{g/cm}^3]\)

Temperature \(T\) [deg Cels.]

bias to high-temperates

'lcdemo.dat' + I(x)
data with experimental errors
Density [g/cm³] vs. Temperature T [°Cels.]

fit weighted by experimental errors

'lcdemo.dat' using 1:2:5

l(x)
initial parameters for realistic model function

Density [g/cm³]

Temperature T [deg Cels.]

'lcdemo.dat'  +  density(x)
Density [g/cm³]

Temperature T [deg Cels.]

fitted to realistic model function

'dc.demo.dat' + density(x)
Density $[\text{g/cm}^3]$ vs. Temperature $T$ [deg Cels.]

'lcdemo.dat' + density(x)

fit with more iterations
the scattered points, and the initial parameter

'hemisphr.dat' using 1:2:3
the scattered points, fitted curve

'hemisphr.dat' using 1:2:3

h(x,y)
sound data, and model with initial parameters

- Density [g/cm³]
- Temperature T [deg Cels.]

'Soundvel.dat'  + vlong(x)  - vtrans(x)
pseudo-3d multi-branch fit to velocity data

Density [g/cm³]

Temperature T [deg Cels.]

'soundvel.dat'  +  vlong(x)  green  vtrans(x)  blue
Density [g/cm³]

Temperature T [deg Cels.]

fitted only every 5th data point

'soundvel.dat'  +  vlong(x)  -  vtrans(x)
initial parameters

Density [g/cm³]

Temperature T [deg Cels.]

'soundvel.dat' + vlong(x) + vtrans(x)
fit with c44 and c13 fixed

Density [g/cm³] vs Temperature T [deg Cels.]
The graph shows a fit via $c_{33}, c_{11}, \phi_0$.

The graph plots Density [g/cm$^3$] against Temperature $T$ [deg Cels.].

- 'soundvel.dat'
- $v_{long}(x)$
- $v_{trans}(x)$
Reflectivity

raw data

'Delta [degrees]' 'moli3.dat'
$t, \frac{\sin(t)}{t}$ or $\frac{\sin(x)}{x}$
$\sin(t), \cos(t)$
Parametric Conic Sections

- $t, -t$
- $\cos(t), \cos(2t)$
- $2\cos(t), \sin(t)$
- $\cosh(t), \sinh(t)$
$\frac{\sin(t)}{t}, \frac{\cos(t)}{t}$
Three circles (with aspect ratio distortion)
$\cos(2^t)$
\[2\sqrt{\cos(t)}\]

\[-2\sqrt{\cos(t)}\]
\[ \sin(4t) \]
\[ \cos(4t) \]
Impedance or Admittance Coordinates

Primitive Smith Chart
Same plot with a multi-line title showing adjustment of plot area to accommodate it.
This is the surface boundary
3D gnuplot demo

X axis
Y axis
Z axis

$z = x^2 + y^2$

$z = x^2 - y^2$
3D gnuplot demo ( ticslevel = 0.5 )

- $x^2 + y^2$
- $x^2 - y^2$
- $xy$
- $(x^3 + y^3) / 10$

X axis

Y axis

Z axis
3D gnuplot demo

Z axis

X axis

Y axis

x*y

+
Surfaces with no grid or tics

- \( x \cdot y \)
- \( x^2 \cdot y^3 \)
- \( x^3 \cdot y^2 \)
Surfaces with z log scale

- $x^2y^2 + 2$
- $x^2y^4 + 2$
- $x^4y^2 + 2$
3D gnuplot demo

\[ \frac{u \cdot v}{(u^2 + v^2 + 0.1)} \]
3D gnuplot demo

\[ \sin(x) \times \cos(y) \]
3D gnuplot demo

\[ \sin(x) \times \cos(y) \]
Sinc function

\[ \text{sinc}(u,v) \]

This is equal to 1

X axis

Y axis

Z axis
This is equal to 1
"fence plot" using separate parametric surfaces
"fence plot" using single parametric surface with undefined points

In the diagram:
- Increasing $v$
- Increasing $u$
- $\text{floor}(u) \mod 3 = 0$
- $\text{floor}(u) \mod 3 = 1$

The plot shows a surface with defined points along the $u$ and $v$ axes, with $z$ values ranging from -1 to 1 along the $z$ axis.
This has logarithmic scale

$z = x^2 + y^2$
Data grid plotting

"glass.dat"
Data grid plotting

"glass.dat" using 3:2:1
Data grid plotting

"glass.dat" using 1
"glass.dat" using 2
"glass.dat" using 3
splot with "set pm3d" (implemented with some terminals)

'glass.dat' every 2::0::12
Test of spherical coordinates

"glass.dat"
Mandelbrot function

mand({0,0},compl(x,y),30)
3 discrete contours at 0 15 75
9 incremental contours starting at -20, stepping by 5
Hidden line removal of explicit surfaces

\[ 1 / (x^2 + y^2 + 1) \]
Hidden line removal of explicit surfaces

\[
x \cdot y / (x^2 + y^2 + 0.1)
\]
Hidden line removal of explicit surfaces

\[ \frac{\sin(x^2 + y^2)}{(x^2 + y^2)} \]
Hidden line removal of explicit surfaces

\[ \sin(x) \times \cos(y) \]
Hidden line removal of explicit surfaces

\[ \sin(x) \times \cos(y) \]

-3 -2 -1 0 1 2 3

-3 -2 -1 0 1 2 3

-1

0

1
Hidden line removal of explicit surfaces

"glass.dat" using 1

0.5
0
-0.5
Hidden line removal of explicit surfaces

"glass.dat" using 1 0.5 0 -0.5

0 2 4 6 8 10 12 14
-1 -0.5 0 0.5 1

0 2 4 6 8 10 12 14
-1 -0.5 0 0.5 1
3D version using spherical coordinate system

\[
\begin{align*}
\cos(u)\cos(v), \cos(u)\sin(v), \sin(u)
\end{align*}
\]

'world.dat'

'world.cor'
3D solid version through hiddenlining

$cos(u)\cdot cos(v)$, $-cos(u)\cdot sin(v)$, $sin(u)$

'world.dat' u 1:2:(1.001)

'world.cor'
3D version using cylindrical coordinate system

\[ \cos(u), \sin(u), v \]

'world.dat'

'world.cor'
Labels colored by GeV plotted in spherical coordinate system
gamma function, very useful function for probability
log gamma function, similarly very useful function
arcsin PDF with $r = 2.0$
arcsin CDF with $r = 2.0$
beta PDF

- $p = 0.5, q = 0.7$
- $p = 5.0, q = 3.0$
- $p = 0.5, q = 2.5$
probability density ->

incomplete beta CDF

- $p = 0.5, q = 0.7$
- $p = 5.0, q = 3.0$
- $p = 0.5, q = 2.5$
binomial PDF with n = 25, p = 0.15
binomial CDF with $n = 25$, $p = 0.15$
cauchy PDF

probability density ->

x ->

a = 0, b = 2
a = 0, b = 4
cauchy CDF

- $a = 0, b = 2$
- $a = 0, b = 4$
chi-square PDF

probability density $\rightarrow$

$x \rightarrow$

$k = 1$
$k = 2$
$k = 3$
$k = 4$
$k = 5$
$k = 6$
$k = 7$
$k = 8$
erlang PDF

probability density ->
x ->
n = 1, exponential r.v.

lambda = 1.0, n = 1
lambda = 0.5, n = 1
lambda = 1.0, n = 2
lambda = 0.5, n = 2
probability density $\rightarrow$

$x \rightarrow$

extreme CDF

- $\alpha = 0.5, u = 1.0$
- $\alpha = 1.0, u = 0.0$
The F PDF (F-distribution Probability Density Function) with different degrees of freedom (df) is shown in the graph. Two distributions are depicted:

- df1 = 5, df2 = 9 (red line)
- df1 = 7, df2 = 6 (green line)

The x-axis represents the value of x, and the y-axis represents the probability density. The graph illustrates how the shape of the distribution changes with different degrees of freedom.
F CDF

- df1 = 5, df2 = 9 (red line)
- df1 = 7, df2 = 6 (green line)
gamma PDF

rho < 1, tends to infinity
rho = 1, finite, nonzero limit
rho > 1, tends to zero

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
Graph showing the incomplete gamma CDF with different parameter values:

- \( \rho = 0.5, \lambda = 1.0 \)
- \( \rho = 1.0, \lambda = 1.0 \)
- \( \rho = 1.0, \lambda = 1.3 \)
- \( \rho = 1.3, \lambda = 1.3 \)
- \( \rho = 2.0, \lambda = 2.0 \)
- \( \rho = 4.0, \lambda = 2.0 \)
- \( \rho = 6.0, \lambda = 2.0 \)
geometric PDF with $p = 0.4$
geometric CDF with $p = 0.4$
half normal PDF, sigma = 1.0

Discontinuity achieved by plotting twice with limited parametric ranges
half normal CDF, sigma = 1.0

Cusp achieved by plotting twice with limited parametric ranges
hypergeometric PDF with $N = 75$, $C = 25$, $d = 10$
hypergeometric CDF with N = 75, C = 25, d = 10
laplace (or double exponential) PDF with $\mu = 0, b = 1$

Cusp achieved by selecting point as part of function samples
laplace (or double exponential) CDF with $\mu = 0$, $b = 1$
logistic PDF with $a = 0$, $\lambda = 2$
logistic CDF with $a = 0$, $\lambda = 2$
lognormal PDF with $\mu = 1.0$, $\sigma = 0.5$
lognormal CDF with mu = 1.0, sigma = 0.5
negative binomial (or pascal or polya) PDF with $r = 8$, $p = 0.4$
negative binomial (or pascal or polya) CDF with $r = 8$, $p = 0.4$
negative exponential (or exponential) PDF with \( \lambda = 2.0 \)
negative exponential (or exponential) CDF with $\lambda = 2.0$
normal (also called gauss or bell-curved) PDF

mu = 0, sigma = 1.0
mu = 2, sigma = 0.5
mu = 1, sigma = 2.0
normal (also called gauss or bell-curved) CDF

- $\mu = 0, \sigma = 1.0$ (red)
- $\mu = 2, \sigma = 0.5$ (green)
- $\mu = 1, \sigma = 2.0$ (blue)
Discontinuity achieved by plotting twice with affine mapped parametric ranges.
Cusp achieved by selecting point as part of function samples

pareto CDF with $a = 1$, $b = 3$
poisson PDF with $\mu = 4.0$
poisson CDF with $\mu = 4.0$
rayleigh PDF with $\lambda = 2.0$
rayleigh CDF with lambda = 2.0
sine PDF

probability density $\rightarrow$

$x \rightarrow$

$a = 2.0, f = 1.0$

$a = 2.0, f = 3.0$

$a = 3.2, f = 2.6$

$a = 2.8, f = 0.0$
a = 2.0, f = 1.0
a = 2.0, f = 3.0
a = 3.2, f = 2.6
a = 2.8, f = 0.0
t PDF (and Gaussian limit)

degrees of freedom

<table>
<thead>
<tr>
<th>nu = 1</th>
<th>nu = 2</th>
<th>nu = 4</th>
<th>nu = 10</th>
<th>nu = 20</th>
<th>normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>green</td>
<td>blue</td>
<td>pink</td>
<td>cyan</td>
<td>yellow</td>
</tr>
</tbody>
</table>

probability density \( \rightarrow \)

x \( \rightarrow \)
probability density $\rightarrow$

$x \rightarrow$

t CDF (and Gaussian limit)

degrees of freedom

<table>
<thead>
<tr>
<th>$\nu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>normal</td>
</tr>
</tbody>
</table>
triangular PDF with $m = 3.0$, $g = 2.0$
triangular CDF with $m = 3.0$, $g = 2.0$
uniform PDF with \( a = -2.0, b = 2.0 \)
uniform CDF with $a = -2.0$, $b = 2.0$
The Weibull probability density function (PDF) is shown with different parameter values:

- \( \lambda = \frac{1}{5}, a = 0.5 \)
- \( \lambda = \frac{1}{5}, a = 1.0 \)
- \( \lambda = \frac{1}{5}, a = 2.0 \)
- \( \lambda = \frac{1}{15}, a = 10.0 \)

The parameter \( a \) affects the shape of the distribution:

- \( a < 1 \); rate decreasing over time
- \( a > 1 \); rate increasing over time
weibull CDF

- $\lambda = \frac{1}{5}, a = 0.5$
- $\lambda = \frac{1}{5}, a = 1.0$
- $\lambda = \frac{1}{5}, a = 2.0$
- $\lambda = \frac{1}{15}, a = 10.0$
probability density ->

binomial PDF using normal approximation

mu

sigma

binom(rnd(x), n, p)
normal(x, mu, sigma)
binomial PDF using poisson approximation

probability density ->

k ->

binom(x, n, p)

poisson(x, mu)

mu

sigma

-2 0 2 4 6 8 10 12 14

k ->
geometric PDF using gamma approximation

\[ \text{geometric(rnd}(x), p) \]
\[ \text{gmm}(x, \rho, \lambda) \]
probability density $\rightarrow$

$\text{geometric PDF using normal approximation}$

$\text{geometric}(\text{rnd}(x), p)$

$\text{normal}(x, \mu, \sigma)$
hypergeometric PDF using binomial approximation

probability density $\rightarrow$

$k \rightarrow$

$$\text{hypgeo}(x, nn, mm, n)$$

$$\text{binom}(x, n, p)$$

$\mu$

$\sigma$
hypergeometric PDF using normal approximation

$\text{normal}(x, \mu, \sigma)$
negative binomial PDF using gamma approximation

probability density ->

k, x ->

mu

sigma

gmm(x, rho, lambda)

negbin(rnd(x), r, p)
negative binomial PDF using normal approximation

probability density -> k, x ->

mu

sigma

negbin(rnd(x), r, p)  
normal(x, mu, sigma)
normal PDF using logistic approximation

\[ \text{logistic}(x, a, \lambda) \]
\[ \text{normal}(x, \mu, \sigma) \]
poisson PDF using normal approximation

poisson(rnd(x), mu)  
normal(x, mu, sigma)
Lattice test for random numbers
Lattice test for random numbers
50 random samples from a 2D Gaussian PDF with unit variance, zero mean and no dependence
Histogram of 5000 random samples from a univariate Gaussian PDF with unit variance and zero mean

Scaled bin frequency

Gaussian p.d.f.
Gaussian 3D cloud of 3000 random samples

Histogram of distance from origin of 3000 multivariate unit variance samples

scaled bin frequency
Maxwell p.d.f.
The cubic Monomial basis functions

- $m_0(x)$
- $m_1(x)$
- $m_2(x)$
- $m_3(x)$
The cubic Hermite basis functions

- $h_{00}(x)$
- $h_{01}(x)$
- $h_{10}(x)$
- $h_{11}(x)$
The cubic Bezier basis functions

$bez_0(x)$
$bez_1(x)$
$bez_2(x)$
$bez_3(x)$
The cubic uniform Bspline basis functions

- $bsp_0(x)$
- $bsp_1(x)$
- $bsp_2(x)$
- $bsp_3(x)$
The cubic Bezier/Bspline basis functions in use

- $\text{cub\_bezier\_x}(t)$, $\text{cub\_bezier\_y}(t)$
- $\text{cub\_bsplin\_x}(t)$, $\text{cub\_bsplin\_y}(t)$
The cubic Hermite basis functions in use

cub_hermit_x1(t), cub_hermit_y1(t)
cub_hermit_x2(t), cub_hermit_y2(t)
Histogram built from unsorted data by 'smooth frequency'

'hemisphr.dat' u (floor($1*20)):(1)
Normal Distribution Function

$\text{norm}(x)$
Inverse Error Function

inverf(x)
Inverse Normal Distribution Function

Invnorm(x)
Simple demo of scatter data conversion to grid data

"hemisphr.dat"
Simple demo of scatter data conversion to grid data

data style lines, dgrid3d 10,10,1

"hemisphr.dat"
Simple demo of scatter data conversion to grid data

"hemisphr.dat"
Simple demo of scatter data conversion to grid data

"hemisphr.dat"
data style lines, dgrid3d ,,16
Simple demo of scatter data conversion to grid data

"hemisphr.dat"

0.8
0.6
0.4
0.2

data style lines, dgrid3d ,,16, contour
Simple demo of scatter data conversion to grid data

"scatter2.dat"
Simple demo of scatter data conversion to grid data

"scatter2.dat"

1.5
1
0.5

data style lines, dgrid3d ,,1
Simple demo of scatter data conversion to grid data
Real part of complex square root function

\[ u^{\ast} \cdot v^{\ast}, 2 \cdot u \cdot v, u \]
Real part of complex square root function (different view)

\[ u^{**2} - v^{**2}, 2*u*v, u \]
Imaginary part of complex square root function

$u^2 - v^2$, $2uv$, $v$
Imaginary part of complex square root function (different view)

$u^2 - v^2, 2uv, v$
Real part of complex cube root function

\[ u^{**3} - 3u^*v^{**2}, 3u^{**2}v-v^{**3}, u \]
Real part of complex cube root function (different view)

$u^{**3}-3*u*v^{**2}, 3*u^{**2}*v-v^{**3}, u$
Imaginary part of complex cube root function

\( u^3 - 3u^2v, 3u^2v - v^3, v \)
Imaginary part of complex cube root function (different view)

\[ u^{**3-3*u*v**2}, 3*u**2*v-v**3, v \]
Real part of complex 4th root function

\[ u^{**4} - 6u^{**2}v^{**2} + v^{**4}, 4u^{**3}v - 4u^v3v^3, u \]
Real part of complex 4th root function (different view)

$$u^{*4}-6u^{*2}v^{*2}+v^{*4}, \quad 4u^{*3}v-4u^{*}v^{*3}, \quad u$$
Imaginary part of complex 4th root function

\[ u^{*4-6u^{**2}v^{**2}+v^{**4}}, 4u^{**3}v-4u^{*}v^{**3}, v \]
Imaginary part of complex 4th root function (different view)

\[ u^{**4} - 6u^{**2}v^{**2} + v^{**4}, \quad 4u^{**3}v - 4u^v v^{**3}, \quad v \]
Enneper’s surface

\[ u - u^{**3}/3 + u^**2, v - v^{**3}/3 + v^**2, u^{**2} - v^{**2} \]
Enneper’s surface (different view)

\[ u-u^{**3/3}+u^v^{**2}, \ v-v^{**3/3}+v^u^{**2}, \ u^{**2}-v^{**2} \]
Moebius strip

\[ (2-v\sin(u/2))\sin(u), (2-v\sin(u/2))\cos(u), v\cos(u/2) \]
Moebius strip (view from opposite side)

\[(2-v\sin(u/2))\sin(u), (2-v\sin(u/2))\cos(u), v\cos(u/2)\]
Klein bottle
Klein bottle with look at the 'inside'
Klein bottle, glassblowers’ version (look-through)
Klein bottle, glassblowers’ version (solid)
NACA6409 -- 9% thick, 40% max camber, 6% camber
12% thick, no camber -- classical test case

NACA0012 Airfoil

mean line
upper surface
lower surface
Joukowski Airfoil using Complex Variables

eps = 0.06 + i0.06

real(\(\eta(t)\)), imag(\(\eta(t)\))
Parametric Sphere

\[
\begin{align*}
\text{cos}(u) \cdot \text{cos}(v), & \quad \text{cos}(u) \cdot \text{sin}(v), & \quad \text{sin}(u)
\end{align*}
\]
Parametric Sphere, crunched z axis

\[ x = \cos(u) \cos(v), \quad y = \cos(u) \sin(v), \quad z = \sin(u) \]
Parametric Sphere, enlarged z axis

\[ \cos(u)\cos(v), \cos(u)\sin(v), \sin(u) \]
Parametric Torus

\((1-0.2\cos(v))\cos(u), (1-0.2\cos(v))\sin(u), 0.2\sin(v)\)
Parametric Hexagon

\[ \cos(v)^3 \cos(u)^3, \sin(v)^3 \cos(u)^3, \sin(u)^3 \]
Parametric Helix

\[(1-0.1\cos(v))\cos(u), (1-0.1\cos(v))\sin(u), 0.1(\sin(v)+u/1.7-10)\]
Parametric Shell (clipped to limited z range)

\[
\begin{align*}
\cos(u)u^*(1+\cos(v)/2), \sin(v)/2, \sin(u)u^*(1+\cos(v)/2)
\end{align*}
\]
Parametric Shell (automatic z range)

\[ \cos(u)u^{1+\cos(v)/2}, \sin(v)u/2, \sin(u)u^{1+\cos(v)/2} \]
Interlocking Tori

\[\begin{align*}
\text{First Torus:} & \quad \cos(u) + 0.5\cos(u)\cos(v), \sin(u) + 0.5\sin(u)\cos(v), 0.5\sin(v) \\
\text{Second Torus:} & \quad 1 + \cos(u) + 0.5\cos(u)\cos(v), 0.5\sin(v), \sin(u) + 0.5\sin(u)\cos(v)
\end{align*}\]
3D gnuplot demo - contour plot
3D gnuplot demo - contour plot (every 10, starting at -100)
3D gnuplot demo - contour plot (at -75, -50, 0)
3D gnuplot demo - contour plot on both $x^2-y^2$.
3D gnuplot demo - some more interesting contours

\[ \frac{xy}{x^2 + y^2 + 0.1} \]
3D gnuplot demo - some more interesting contours

$\sin(x) \times \cos(y)$
3D gnuplot demo - some more interesting contours

$\sin(x) \cdot \cos(y)$

- $0.8$
- $0.6$
- $0.4$
- $0.2$
- $-5.55e-17$
- $-0.2$
- $-0.4$
- $-0.6$
- $-0.8$
3D gnuplot demo - low resolution (6x6)

sin(x) * cos(y)
-0.5
0
0.5
3D gnuplot demo - low resolution (6x6) using bspline approx.

$sin(x) \times cos(y)$
3D gnuplot demo - low resolution (6x6) raise bspline order.

\[
\sin(x) \times \cos(y)
\]

-0.5
0
0.5
1

X axis
Y axis
Z axis
3D gnuplot demo - low resolution (6x6) using linear contours.
3D gnuplot demo - low resolution (6x6) using bspline approx.
3D gnuplot demo - contour of Sinc function

\[ \frac{\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}} \]
3D gnuplot demo - contour of Sinc function

\[ \sin(\sqrt{x^2+y^2}) / \sqrt{x^2+y^2} \]
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
Rosenbrock Function

\[(1-x)^2 + 100(y - x^2)^2\]
Rosenbrock Function

\[(1-x)^2 + 100(y - x^2)^2\]
Rosenbrock Function
All contours drawn in a single color

$$(1-x)^2 + 100(y-x^2)^2$$
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

\[ f(x) = \exp(-x^2) \]

\[ \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int f(x) \]
approximate the integral of functions

\[ f(x) = \cos(x) \]

\[ \text{integral}_f(x) \]
approximate the integral of functions (upper and lower limits)

\[ f(x) = (x-2)^2 - 20 \]

\[ \int_{-5}^{x} f(x) \, dx \]
approximate the integral of functions (upper and lower limits)

\[ f(x) = \sin(x-1) - 0.75\sin(2x-1) + \frac{x^2}{8} - 5 \]

integral2_f(x,1)
Plot of the ackermann function

$ack(x, y)$
\[\text{max}(\sin(x), \min(x^2, x^3)) + 0.5\]
Greatest Common Divisor (for integers only)

gcd(x, 60)
Implemented using built-in rgb color names

Terminal-independent RGB colors in 2D

- red
- orange
- yellow
- green
- cyan
- blue
- violet

```
set style line 1 lt rgb "red" lw 3
set style line 2 lt rgb "orange" lw 2
set style line 3 lt rgb "yellow" lw 3
set style line 4 lt rgb "green" lw 2
set style line 5 lt rgb "cyan" lw 3
set style line 6 lt rgb "blue" lw 2
set style line 7 lt rgb "violet" lw 3
```

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
Demo of hidden3d with points only (no surface)

variable pointsize and rgb color computed from coords
pm3d demo. Radial sinc function. Default options.

\[ \sin(\sqrt{x^2+y^2})/\sqrt{x^2+y^2} \]
pm3d at s (surface) / ticslevel 0

\sin(\sqrt{x^2+y^2})/\sqrt{x^2+y^2}
pm3d at b (bottom)

$\sin(\sqrt{x^2+y^2})/\sqrt{x^2+y^2}$
unset surface; set pm3d at st (surface and top)
set pm3d at bstbst (funny combination, only for screen or postscript)
colour map, using default rgbformulae 7,5,15 ... traditional pm3d (black-blue-red-yellow)
colour, rgbformulae 23,28,3  ... ocean (green-blue-white); OK are also all other permutations
rgbformulae 31,-11,32: negative formula number=inverted color
set pm3d scansforward: wrong, because back overwrites front
set pm3d scansbackward: correctly looking surface
set hidden3d

log(x*x*y*y)

-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

-12
-10
-8
-6
-4
-2
0
2
4
set pm3d hidden3d <linetype>: pm3d’s much faster hidden3d variant
bad: surface and top are too close together
solution: use independent 'set zrange' and 'set cbrange'
color box is on by default at a certain position
color box is on again, now with horizontal gradient

see cblabel, grid cb, mcbtics, ...
color box is switched off
using now "set grid back; unset colorbox"
Datafile with different nb of points in scans; pm3d flush begin
Datafile with different nb of points in scans; pm3d flush center
Datafile with different nb of points in scans; pm3d flush end
Data with different nb of points in scans; pm3d ftriangles flush begin
Data with different nb of points in scans; pm3d ftriangles flush center
Data with different nb of points in scans; pm3d ftriangles flush end
Using interpolation with datafile; pm3d map interpolate 2,1
Using interpolation with datafile; pm3d map ftriangles interpolate 10,1
Using interpolation with datafile; pm3d at s ftriangles interpolate 10,1
only for enhanced terminals: 'set format cb ...'
function 'x+y' using all colors available, 'set pal maxcolors 0'
function ‘x+y’ using only 5 colors, ‘set pal maxcolors 5’
color lines: 'splot sin(y)/(y) with lines palette’
Demo for clipping of 2 rectangles comes now. The xrange is [0:2].
...and now xrange is [0:1.5] and 'set pm3d clip1in'

'clip14in.dat'
...now xrange is [0:1.5] and 'set pm3d clip4in' 

'clip14in.dat'
set pm3d corners2color c1

set pm3d corners2color c2

set pm3d corners2color c3

set pm3d corners2color c4

set pm3d corners2color mean

set pm3d corners2color geomean

set pm3d corners2color median

Original grid points
set palette defined

![Graph showing a color gradient with labeled axes and scale.]
set palette defined (0 0 0 0, 1 0 0 1, 3 0 1 0, 4 1 0 0, 6 1 1 1)
set palette defined ( 0 "green", 1 "blue", 2 "red", 3 "orange" )
set palette defined ( 20 "#101010", 30 "#ff0000", 40 "#00ff00", 50 "#e0e0e0" )
set palette model HSV defined (0 0 1 1, 1 1 1 1)
set palette model RGB defined
(0 'green', 1 'dark-green', 1 'yellow', 2 'dark-yellow', 2 'red', 3 'dark-red')
set palette file "-" (file with 3 columns)
set palette file "-" (file with 4 columns)
set palette model XYZ rgbformulae 7,5,15
set palette functions gray, gray, gray
set palette functions sqrt(gray), gray**3, sin(gray*2*pi) <- 7,5,15
set palette rgbformulae 7,5,15
set palette model XYZ functions model XYZ  gray**0.35, gray**0.5, gray**0.8
set palette model RGB functions
$4x(1-\theta(x-0.25)), 0.5\theta(x-0.25)(1-\theta(x-0.5)), x$
gamma = 0.75
$\gamma = 1.0$
gamma = 1.25

The graph shows a color gradient with darker shades indicating lower values and lighter shades indicating higher values. The x-axis ranges from -10 to 10, and the y-axis ranges from -10 to 10.
gamma = 1.5
Interlocking Tori - PM3D surface with no depth sorting
Interlocking Tori - PM3D surface with depth sorting
Textcolor options in 2D plot (notice this title in color)

- label with textcolor lt 1
- label with tc default
- label with tc lt 2
- label with tc lt 3

- color of ylabel should still be black
- color of xlabel should be lt 4
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
textcolor default

textcolor cb 5
tc cb 0

tc cb -5
textcolor frac .75
tc frac .25

xlabel should be lt 4
ylabel should still be black
These examples require no extra fonts:

\[ 10^{-2} \quad A_{j,k} \quad e^x \]
\[ x_k^2 \quad x_0^{-3/2}y \]

Space-holders:

\[
\langle \text{Big} \rangle \langle x_0^{-3/2}y \rangle
\]

Requires Symbol font:

\[
\int_0^\infty e^{-u^{2/2}} \, d\mu = (\pi/2)^{1/2}
\]

Overprint (v should be centred over d)

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

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

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
Top: plot with vectors arrowstyle 5, Bottom: explicit arrows
Top: plot with vectors arrowstyle 6, Bottom: explicit arrows

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

arrowstyle 1:
arrowstyle 2:
arrowstyle 3:
arrowstyle 4:
arrowstyle 5:
arrowstyle 6:
arrowstyle 7:
arrowstyle 8:
Plot 'file' with vectors <arrowstyle>
"field2xy.tmp" u 1:2:(coef*dx1($1,$2)):coef*dy1($1,$2))
"equipo2.tmp"
Default tics settings
Different modification of tics settings
Different modification of tic settings
Auto-labeling plots from text fields in datafile

Generate plot labels from first row in each column
Generate x-axis labels from first column in each row

Runoff 1992-2000
Runoff 2001-2002
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
Same plot using x2ticlabels also

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mai</td>
<td>60</td>
<td>40</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>jun</td>
<td>40</td>
<td>30</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>jul</td>
<td>20</td>
<td>10</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>aug</td>
<td>30</td>
<td>20</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>sep</td>
<td>40</td>
<td>30</td>
<td>25</td>
<td>20</td>
</tr>
<tr>
<td>okt</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>35</td>
</tr>
<tr>
<td>nov</td>
<td>80</td>
<td>70</td>
<td>60</td>
<td>55</td>
</tr>
<tr>
<td>des</td>
<td>100</td>
<td>90</td>
<td>80</td>
<td>75</td>
</tr>
<tr>
<td>jan</td>
<td>120</td>
<td>110</td>
<td>100</td>
<td>95</td>
</tr>
<tr>
<td>feb</td>
<td>140</td>
<td>130</td>
<td>120</td>
<td>115</td>
</tr>
<tr>
<td>mar</td>
<td>160</td>
<td>150</td>
<td>140</td>
<td>135</td>
</tr>
<tr>
<td>apr</td>
<td>180</td>
<td>170</td>
<td>160</td>
<td>155</td>
</tr>
</tbody>
</table>
Plot from table format (titles taken from column headers)
Read labels from a datafile column
Here the 'plot with labels' command generates a C-alpha trace of retro-GCN4 peptide

Gly Gly Arg Glu Gly Val Leu Lys Lys Leu Arg Ala Val Glu Asn Glu Leu His Tyr Asn Lys Leu Leu Glu Glu Val Lys Asp Glu Leu Gln Lys Met Arg Gln Leu
3D version generated by splot
(sorry, the axis scales are distorted)
US immigration from Europe by decade

- Austria
- Hungary
- Belgium
- Czechoslovakia
- Denmark
- France
- Germany
- Greece
- Ireland
- Italy
- Netherlands
- Norway
- Sweden
- Poland
- Portugal
- Romania
- Soviet Union
- Spain
- Switzerland
- United Kingdom
- Yugoslavia
US immigration from Northern Europe
Plot selected data columns as histogram of clustered boxes

Denmark
Netherlands
Norway
Sweden
US immigration from Northern Europe
(same plot with larger gap between clusters)
US immigration from Europe by decade
Plot as stacked histogram

- Austria
- Hungary
- Belgium
- Czechoslovakia
- United Kingdom
- Sweden
- Norway
- Poland
- Portugal
- Spain
- Switzerland
- Germany
- Greece
- Ireland
- Italy
- Netherlands
- France
- United States
- Yugoslavia
- Soviet Union
- Yugoslavia
- Spain
- Portugal
- Poland
- Sweden
- Norway
- Poland
- Portugal
- Spain
- Switzerland
- United Kingdom
- Yugoslavia
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
- Austria
- Belgium
- Czechoslovakia
- Hungary
- Denmark
Immigration from Northern Europe
(columstacked histogram)

Country of Origin

Denmark  Netherlands  Norway  Sweden

Immigration by decade

- 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)

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 by decade

(1891-1900)

1901-1910

1911-1920

1921-1930

1931-1940

1941-1950

1951-1960

1961-1970

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

Denmark
Norway
Sweden
Greece
Romania
Yugoslavia
Ireland
United Kingdom

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

Southern Europe

British Isles

Denmark
Norway
Sweden
Greece
Romania
Yugoslavia
Ireland
United Kingdom
Immigration from different regions

Denmark
Norway
Sweden
Greece
Romania
Yugoslavia
Ireland
United Kingdom
Southern Europe
Northern Europe
British Isles

Explicit start color in 'newhistogram' command
Immigration from different regions

- Denmark
- Norway
- Sweden
- Greece
- Romania
- Yugoslavia
- Ireland
- United Kingdom
- British Isles

Explicit start pattern in 'newhistogram' command
Immigration from different regions

- Northern Europe
  - Sweden
  - Norway
  - Denmark

- Southern Europe
  - Yugoslavia
  - Romania
  - Greece

- British Isles
  - United Kingdom
  - Ireland

Explicit start pattern and linetype
Larry Ewing’s GIMP penguin on vacation basking in the balmy waters off the coast of Murmansk

"I flew here... by plane. Why? For the halibut."
Translations of position variables via 'using'

"Time for a dip..."
Palette mode 'image' used to produce psychedelic bird

"I am the penguin, GOO GOO GOO JOOB."
The palette can be changed from color to gray scale

"This picture was taken by my friend Ansel Adams."
As with 3d color surfaces, a color box may be added to the plot.
Polygons used to draw pixels for rotated images
Notice the slower refresh rate than for the next plot

\( y = 128 \times 128 \) \( dx = 0.70711 \) \( dy = 0.70711 \) flipy rotation=45d center=(63.5,63.5) format='uchar' using \((1+2+3)/3\)
Terminal image routine used to draw plot rotated about origin
Notice the faster refresh rate than for the previous plot
Selection of the input channels via 'using'

"I do impersonations..."

"A cardinal."

"A parrot."

"A bluebird."
Luminance adjustment via ‘cbrange’

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

"Lucky I brought sunscreen."

Sunset on the Terrace

Sultry evening
2d binary data example where record length is part of command

'scatter2.bin' binary endian=little record=30:30:29:26 using 1:2:3
If plots in columns match, your compiler is little endian.
Close up of pixels having grid points (0,0), (0,2), (2,0) and (2,2)
Simple extension of a two dimensional image into three dimensions

'blutux.rgb' binary array=128x128 flip=y format='%uchar%uchar%uchar' using ($1+$2+$3)/3
Orientation operations from 'plot' also apply to 'splot'

binary 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

Along the y-axis

Along a 225 degree projection
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.
Uniform sampling in the polar coordinate system

'sine.bin' binary endian=little array=201 dt=0.018326 origin=(0,0) format='%f' using 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?"
... Sure, go ahead.

'blutux.rgb' binary array=128x128 flipy format='%uchar' every 8:8

"Picasso's 'Penguin on Beach'"
Automatically recognizing file type and extracting file information

'demo.edf' binary filetype=auto
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
"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 = strstr(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))

\[300,12\]
\[310,26\]
\[320,17\]
\[330,8\]
\[350,8\]
\[360,10\]
\[370,20\]
\[380,14\]
\[390,8\]
\[400,10\]
plot <foo> using 1:2:( ($3>$2) ? "Up" : "Dn" ) with labels
Constant string expressions as plot symbols

Same thing using character glyphs from WingDings font
String-valued functions to generate datafile names

\[ 5\sin(x)/x \]

\text{file}(N)

\text{file}(M)
plot with variable size points
splot with variable size points
it is possible to specify size and color separately
3D version using spherical coordinate system
3D solid version through hiddenlining
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
$-3 + \sin(x^5)/x$
Terminal Test

- Test of character width:
  - Left justified
  - Centre justified
  - Right justified

- Rotated text:
  - Rotated by +45 deg
  - Rotated by -45 deg

- Linewidth:
  - lw 1
  - lw 2
  - lw 3
  - lw 4
  - lw 5
  - lw 6

- Pattern fill:
  - 0
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
  - 8
  - 9

- Show ticscale:
  - -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

- (Color) filled polygon: