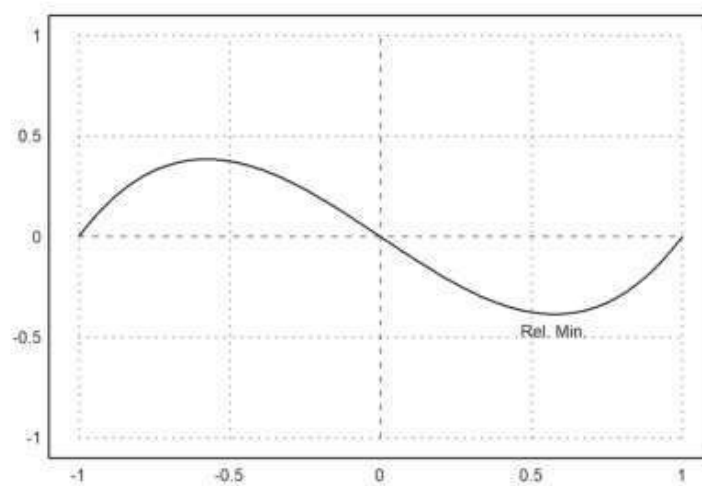
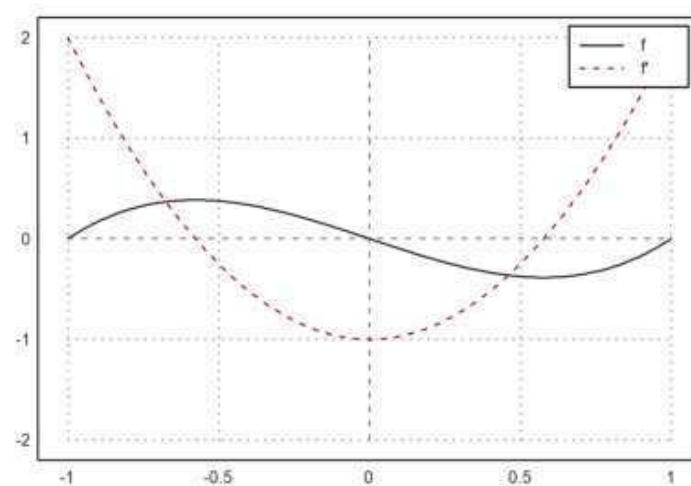


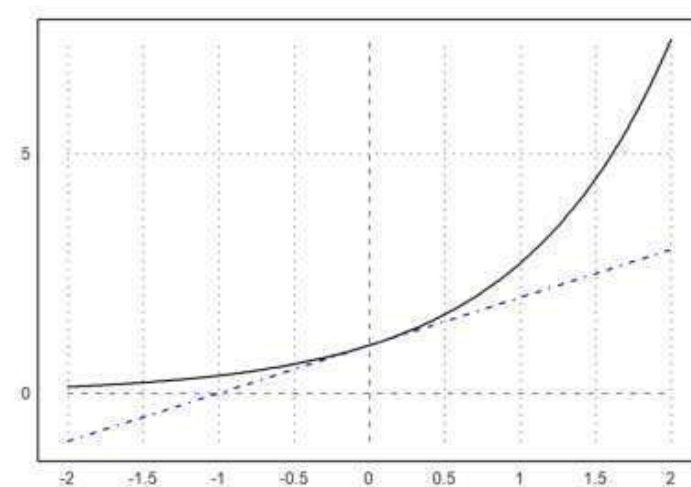


Terdapat juga kotak teks.

```
>plot2d(&f(x),-1,1,-2,2); // fungsi
>plot2d(&diff(f(x),x),>add,style="--",color=red); // turunan
>labelbox(["f","f'"],["-","--"],[black,red]): // kotak label
```

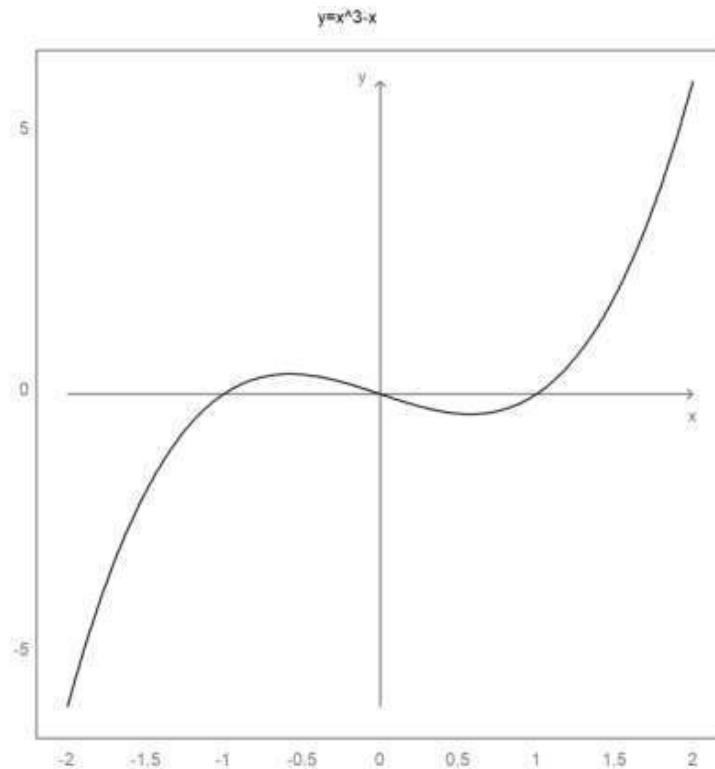


```
>plot2d(["exp(x)","1+x"],color=[black,blue],style=["-","-."]):
```



```
>gridstyle("->",color=gray,textcolor=gray,framecolor=gray); ...
plot2d("x^3-x",grid=1); ...
settitle("y=x^3-x",color=black); ...
label("x",2,0,pos="bc",color=gray); ...
```

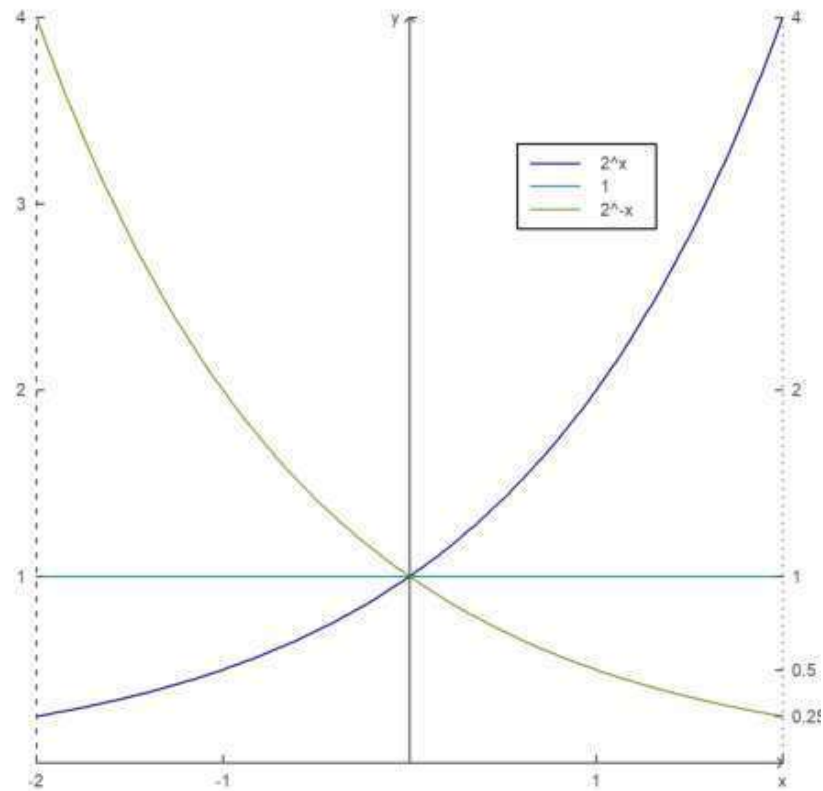
```
label("y",0,6,pos="cl",color=gray); ...
reset();
```



For even more control, the x-axis and the y-axis can be done manually.

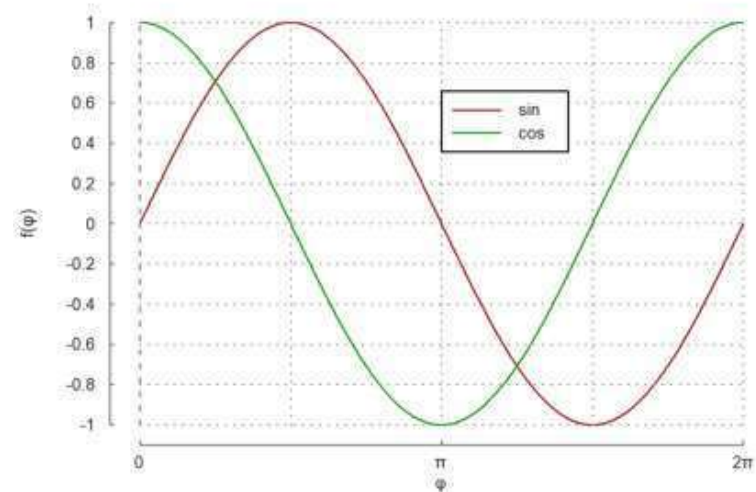
The command `fullwindow()` expands the plot window since we no longer need place for labels outside the plot window. Use `shrinkwindow()` or `reset()` to reset to the defaults.

```
>fullwindow; ...
gridstyle(color=darkgray,textcolor=darkgray); ...
plot2d(["2^x","1","2^(-x)"],a=-2,b=2,c=0,d=4,<grid,color=4:6,<frame); ...
xaxis(0,-2:1,style="->"); xaxis(0,2,"x",<axis); ...
yaxis(0,4,"y",style="->"); ...
yaxis(-2,1:4,>left); ...
yaxis(2,2^(-2:2),style=".",<left); ...
labelbox(["2^x","1","2^-x"],colors=4:6,x=0.8,y=0.2); ...
reset;
```



Berikut contoh lain, di mana string Unicode digunakan dan sumbu berada di luar area plot.

```
>aspect(1.5);
>plot2d(["sin(x)", "cos(x)"], 0, 2pi, color=[red, green], <grid, <frame); ...
  xaxis(-1.1, (0:2)*pi, xt=["0", u"&pi;", u"2&pi;"], style="-", >ticks, >zero); ...
  xgrid((0:0.5:2)*pi, <ticks); ...
  yaxis(-0.1*pi, -1:0.2:1, style="-", >zero, >grid); ...
  labelbox(["sin", "cos"], colors=[red, green], x=0.5, y=0.2, >left); ...
  xlabel(u"&phi;"); ylabel(u"f(&phi;)"):
```



Plot Data 2D

Jika x dan y adalah vektor data, data ini akan digunakan sebagai koordinat x dan y dari sebuah kurva. Dalam hal ini, a , b , c , dan d , atau radius r dapat ditentukan, atau jendela plot akan menyesuaikan secara otomatis dengan data. Sebagai alternatif, `>square` dapat digunakan untuk mempertahankan rasio aspek persegi.

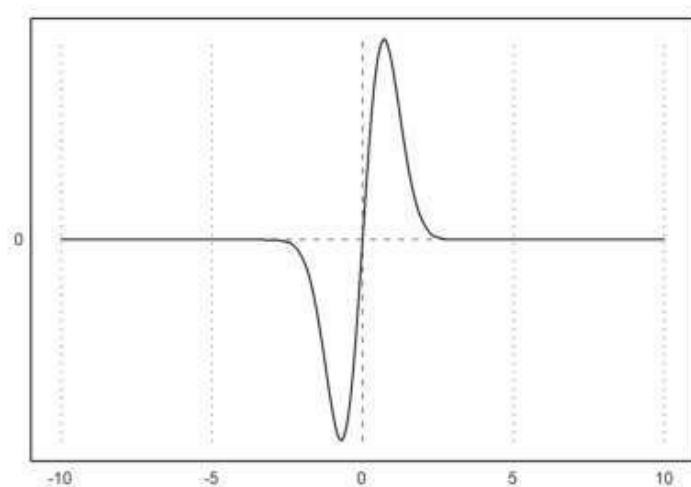
Memplot sebuah ekspresi hanyalah singkatan untuk plot data. Untuk plot data, Anda memerlukan satu atau lebih baris nilai x , dan satu atau lebih baris nilai y . Dari rentang dan nilai x , fungsi `plot2d` akan menghitung data yang akan diplot, secara bawaan dengan evaluasi adaptif dari fungsi. Untuk plot titik gunakan `>points`, untuk gabungan garis dan titik gunakan `>addpoints`.

Namun, Anda juga bisa memasukkan data secara langsung.

- Gunakan vektor baris untuk x dan y untuk satu fungsi.
- Matriks untuk x dan y akan diplot baris demi baris.

Berikut contoh dengan satu baris untuk x dan y.

```
>x=-10:0.1:10; y=exp(-x^2)*x; plot2d(x,y):
```



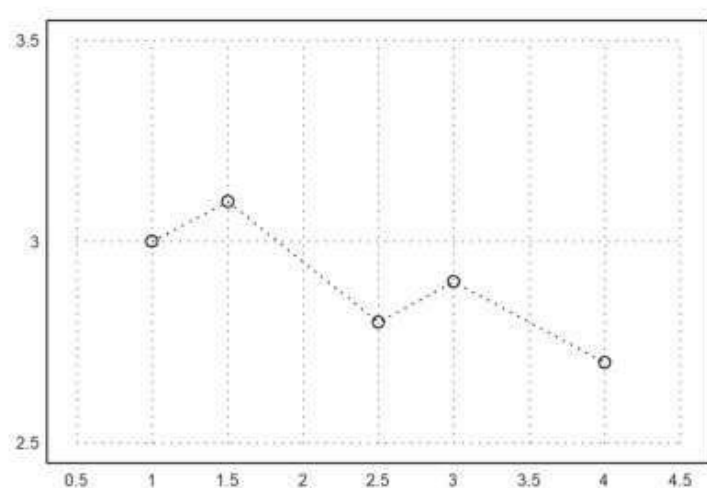
Data juga dapat diplot sebagai titik. Gunakan points=true untuk ini. Plot bekerja seperti poligon, tetapi hanya menggambar sudut-sudutnya.

- style="...": Pilih dari "[", "<>", "o", ".", "..", "+", "*", "\[]#", "<>#", "o#", "..#", "#", "|".

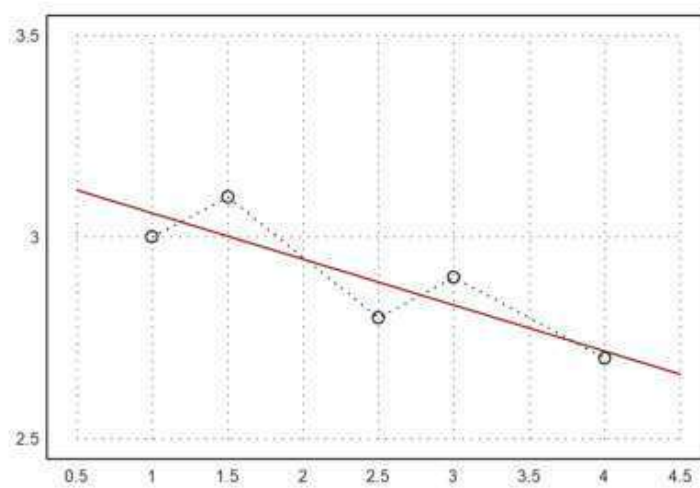
Untuk memplot sekumpulan titik, gunakan >points. Jika color adalah vektor warna, setiap titik akan mendapatkan warna berbeda. Untuk matriks koordinat dan vektor kolom, warna diterapkan pada baris-baris matriks.

Parameter >addpoints menambahkan titik pada segmen garis untuk plot data.

```
>xdata=[1,1.5,2.5,3,4]; ydata=[3,3.1,2.8,2.9,2.7]; // data
>plot2d(xdata,ydata,a=0.5,b=4.5,c=2.5,d=3.5,style="."); // garis
>plot2d(xdata,ydata,>points,>add,style="o"): // tambahkan titik
```



```
>p=polyfit(xdata,ydata,1); // dapatkan garis regresi
>plot2d("polyval(p,x)",>add,color=red): // tambahkan plot garis
```



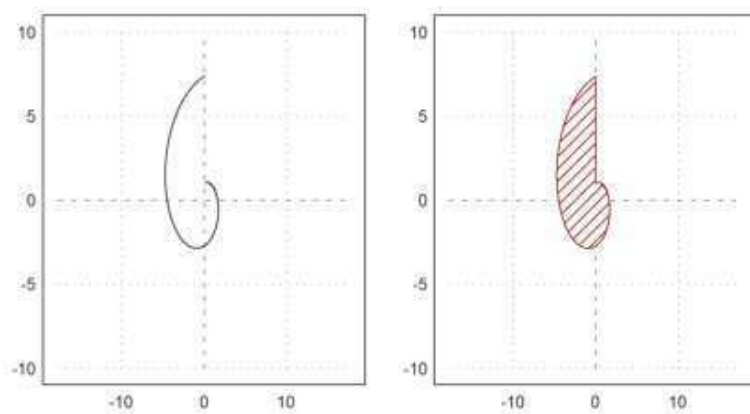
Menggambar Daerah yang Dibatasi Kurva

Plot data sebenarnya berupa poligon. Kita juga dapat memplot kurva atau kurva berisi.

- `filled=true` mengisi plot.
- `style="..."`: Pilih dari "#", "/", "", "/".
- `fillcolor`: Lihat penjelasannya sebelumnya untuk warna yang tersedia.

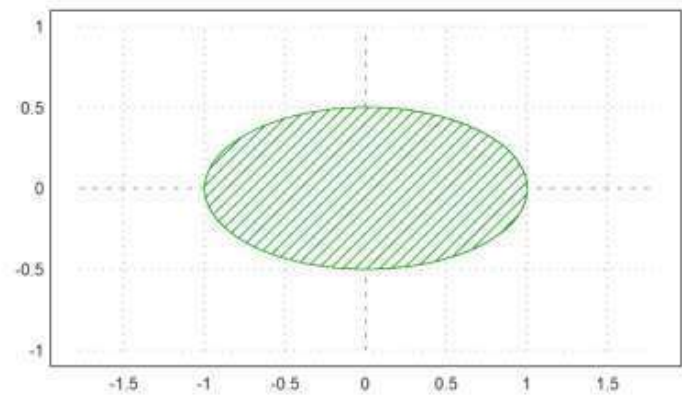
Warna isi ditentukan oleh argumen `"fillcolor"`, dan opsional `\<outline` mencegah penggambaran batas untuk semua gaya kecuali gaya bawaan.

```
>t=linspace(0,2pi,1000); // parameter untuk kurva
>x=sin(t)*exp(t/pi); y=cos(t)*exp(t/pi); // x(t) dan y(t)
>figure(1,2); aspect(16/9)
>figure(1); plot2d(x,y,r=10); // kurva plot
>figure(2); plot2d(x,y,r=10,>filled,style="/",fillcolor=red); // isi kurva
>figure(0):
```

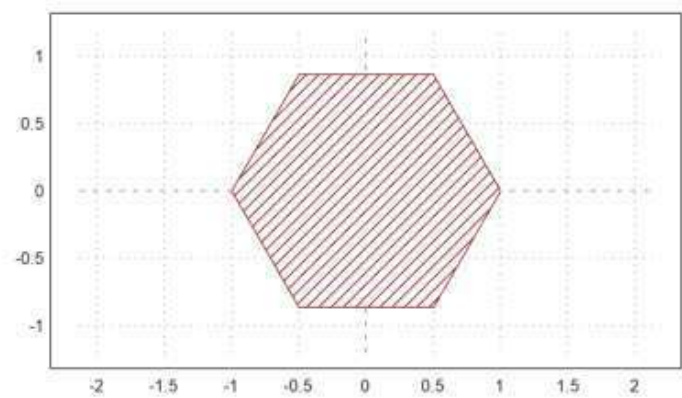


Pada contoh berikut, kita memplot sebuah elips berisi dan dua segi enam berisi menggunakan kurva tertutup dengan 6 titik dan gaya isi yang berbeda.

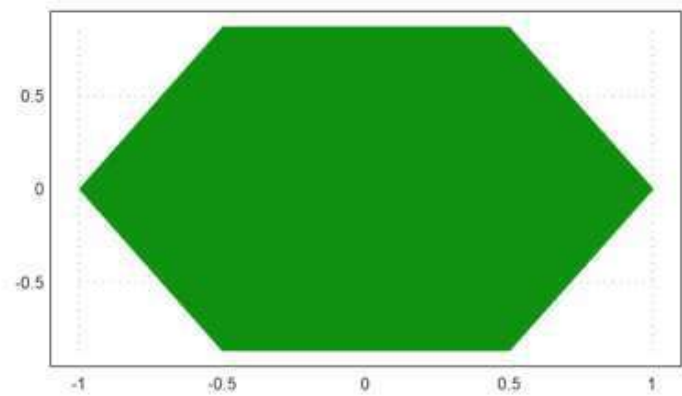
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```



```
>t=linspace(0,2pi,6); ...  
plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1.2):
```

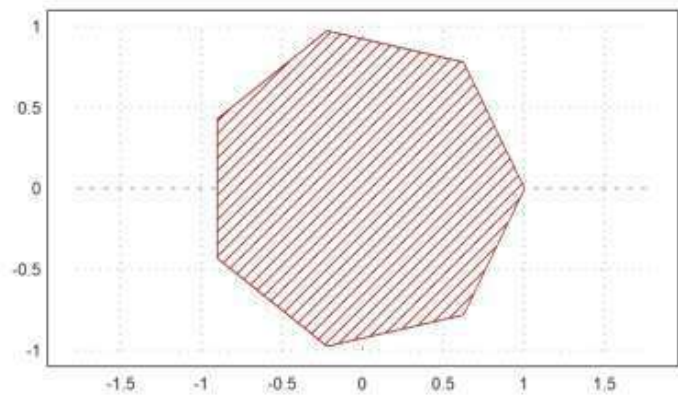


```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



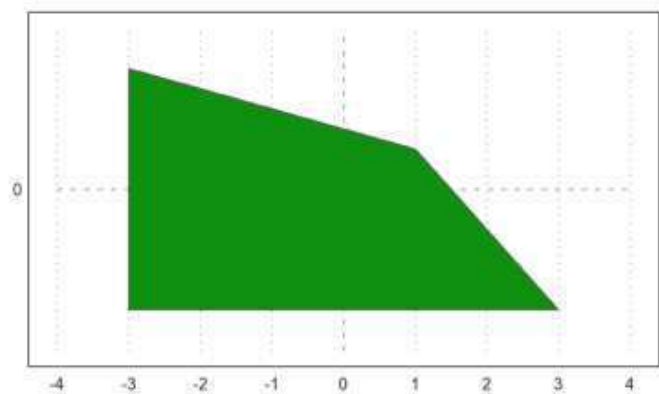
Contoh lain adalah sebuah segi tujuh, yang kita buat dengan 7 titik pada lingkaran satuan.

```
>t=linspace(0,2pi,7); ...  
plot2d(cos(t),sin(t),r=1,>filled,style="/",fillcolor=red):
```



Berikut adalah himpunan nilai maksimum dari empat kondisi linear yang kurang dari atau sama dengan 3. Ini adalah $A[k].v = 3$ untuk semua baris dari A. Untuk mendapatkan sudut yang rapi, kita menggunakan n yang relatif besar.

```
>A=[2,1;1,2;-1,0;0,-1];
>function f(x,y) := max([x,y].A');
>plot2d("f",r=4,level=[0;3],color=green,n=111):
```

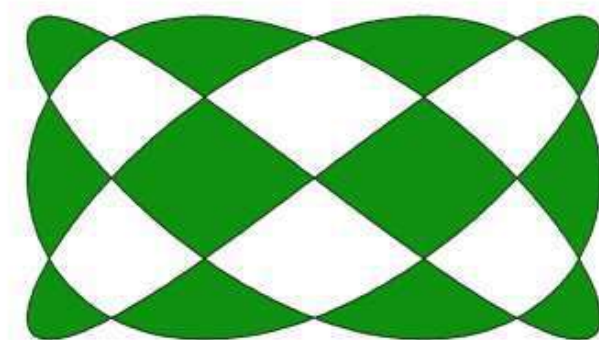


Inti dari bahasa matriks adalah bahwa bahasa ini memungkinkan pembuatan tabel fungsi dengan mudah.

```
>t=linspace(0,2pi,1000); x=cos(3*t); y=sin(4*t);
```

Sekarang kita memiliki vektor x dan y dari nilai-nilai. `plot2d()` dapat memplot nilai-nilai ini sebagai kurva yang menghubungkan titik-titik. Plot ini dapat diisi. Dalam hal ini, hasilnya menjadi rapi berkat aturan winding yang digunakan untuk pengisian.

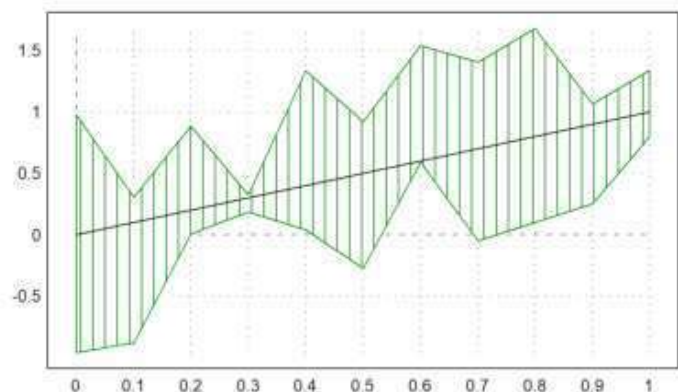
```
>plot2d(x,y,<grid,<frame,>filled):
```



Sebuah vektor interval diplot terhadap nilai x sebagai daerah berisi antara nilai bawah dan atas dari interval tersebut.

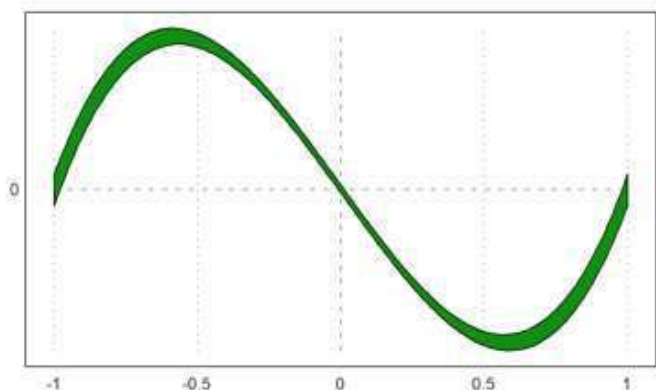
Ini berguna untuk memplot kesalahan perhitungan. Namun, ini juga dapat digunakan untuk memplot kesalahan statistik.

```
>t=0:0.1:1; ...
  plot2d(t,interval(t-random(size(t)),t+random(size(t))),style="|"); ...
  plot2d(t,t,add=true):
```



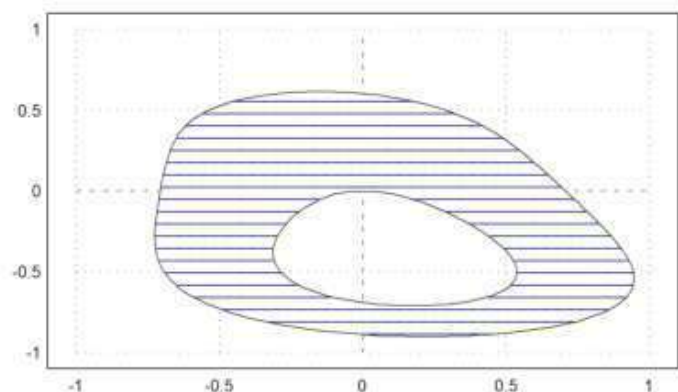
Jika x adalah vektor yang sudah diurutkan, dan y adalah vektor interval, maka plot2d akan memplot rentang isi dari interval tersebut di bidang. Gaya pengisian sama seperti gaya poligon.

```
>t=-1:0.01:1; x=~t-0.01,t+0.01~; y=x^3-x;
>plot2d(t,y):
```



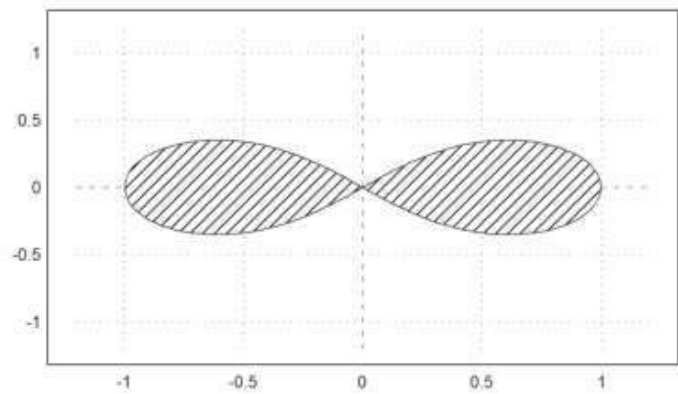
Dimungkinkan untuk mengisi daerah nilai untuk sebuah fungsi tertentu. Untuk ini, level harus berupa matriks 2x1. Baris pertama berisi batas bawah, dan baris kedua berisi batas atas.

```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```

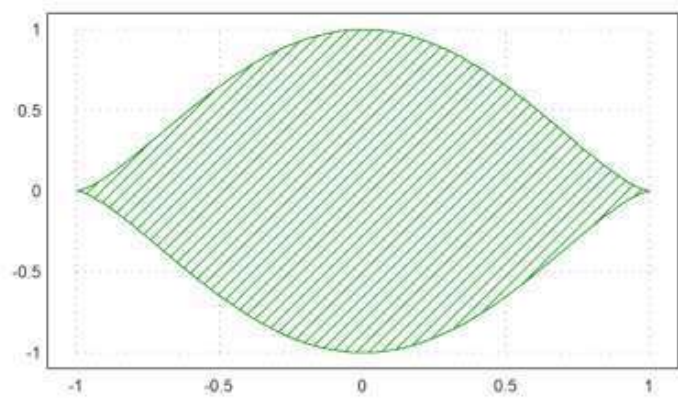


Kita juga dapat mengisi rentang nilai seperti

```
>plot2d("(x^2+y^2)^2-x^2+y^2",r=1.2,level=[-1;0],style="/"):
```



```
>plot2d("cos(x)", "sin(x)^3",xmin=0,xmax=2pi,>filled,style="/"):
```



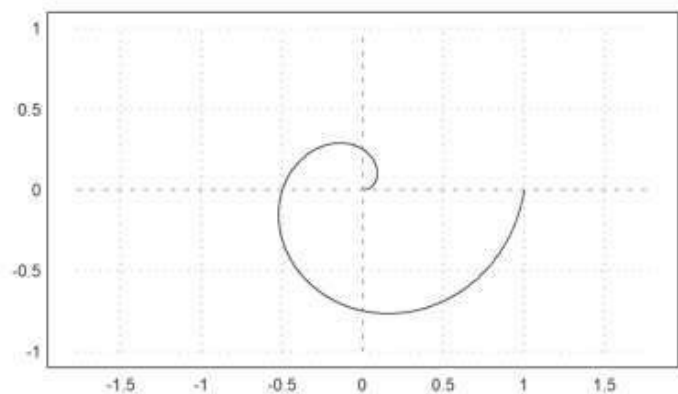
Grafik Fungsi Parametrik

Nilai x tidak perlu diurutkan. (x,y) cukup menggambarkan sebuah kurva. Jika x diurutkan, kurva tersebut merupakan grafik dari sebuah fungsi.

Pada contoh berikut, kita memplot spiral

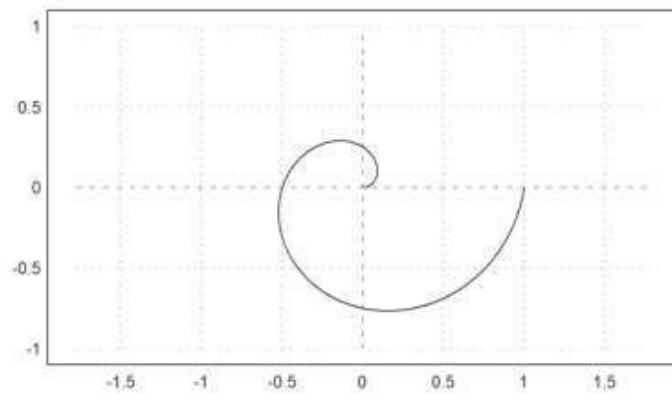
Kita perlu menggunakan sangat banyak titik agar tampilan halus atau menggunakan fungsi `adaptive()` untuk mengevaluasi ekspresi (lihat fungsi `adaptive()` untuk penjelasan lebih lanjut).

```
>t=linspace(0,1,1000); ...
plot2d(t*cos(2*pi*t),t*sin(2*pi*t),r=1):
```

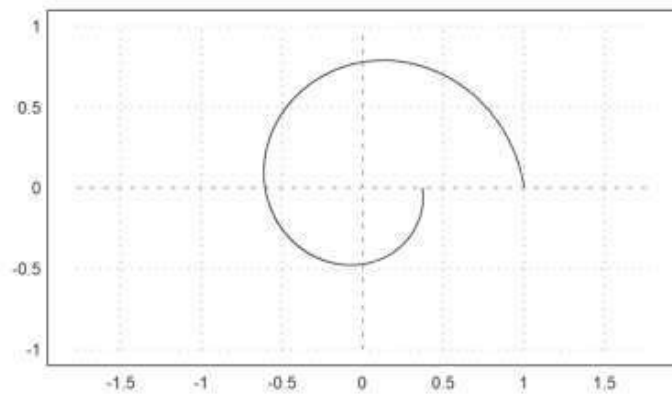


Sebagai alternatif, dimungkinkan menggunakan dua ekspresi untuk kurva. Contoh berikut memplot kurva yang sama seperti sebelumnya.

```
>plot2d("x*cos(2*pi*x)", "x*sin(2*pi*x)", xmin=0, xmax=1, r=1):
```



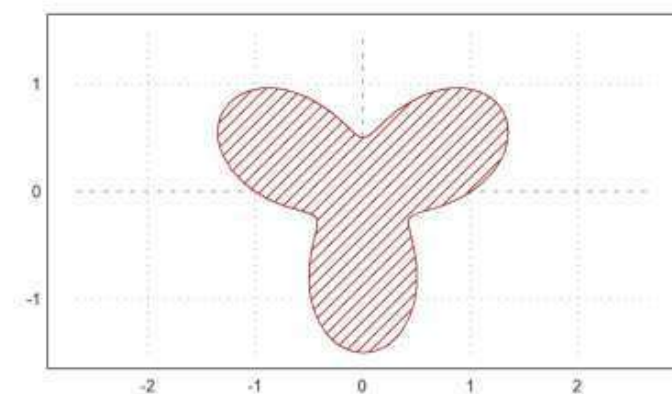
```
>t=linspace(0,1,1000); r=exp(-t); x=r*cos(2pi*t); y=r*sin(2pi*t);
>plot2d(x,y,r=1):
```



Pada contoh berikut, kita memplot kurva

dengan

```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
plot2d(x,y,>filled,fillcolor=red,style="/",r=1.5):
```



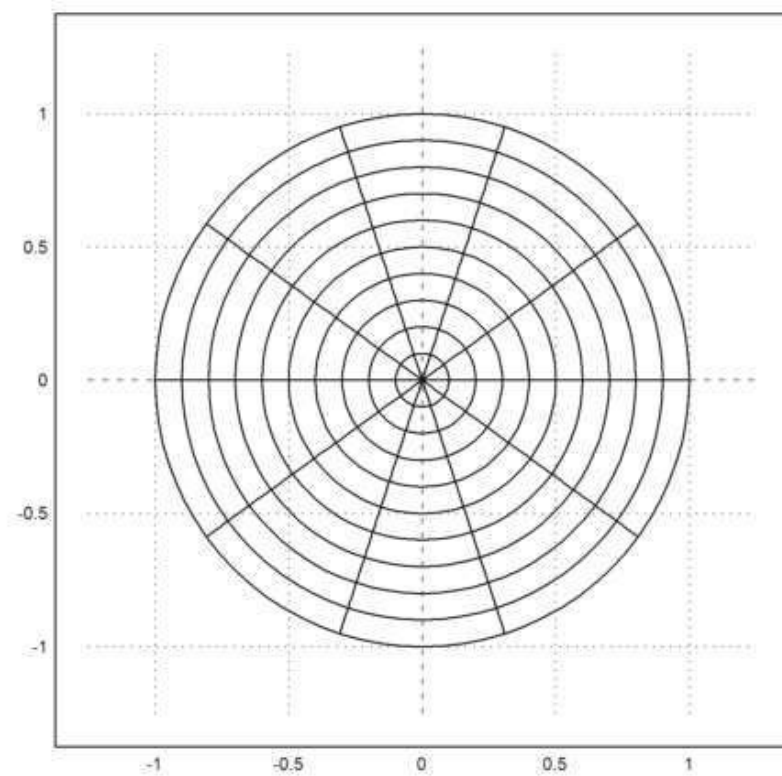
Menggambar Grafik Bilangan Kompleks

Sebuah array bilangan kompleks juga dapat diplot. Titik-titik grid kemudian akan dihubungkan. Jika jumlah garis grid ditentukan (atau vektor 1U2 dari garis grid) pada argumen cgrid, hanya garis-garis grid tersebut yang terlihat.

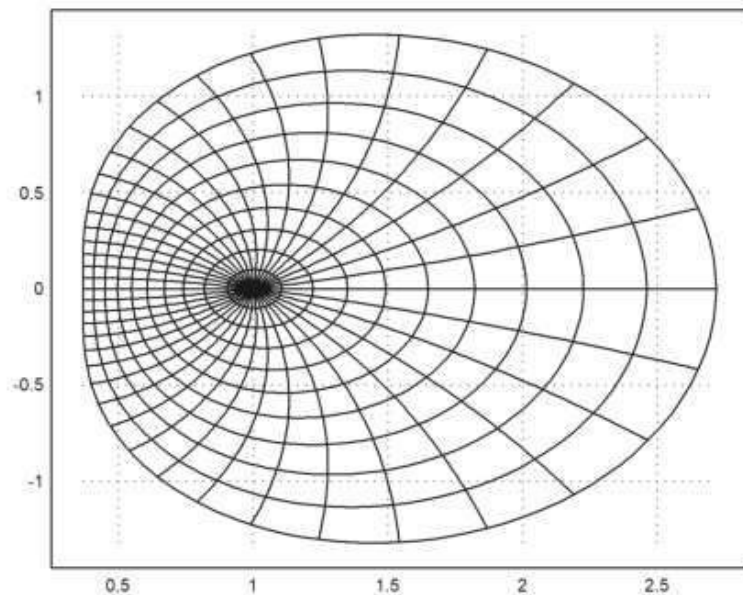
Sebuah matriks bilangan kompleks akan secara otomatis diplot sebagai grid di bidang kompleks.

Pada contoh berikut, kita memplot citra lingkaran satuan di bawah fungsi eksponensial. Parameter `cgrid` menyembunyikan sebagian grid.

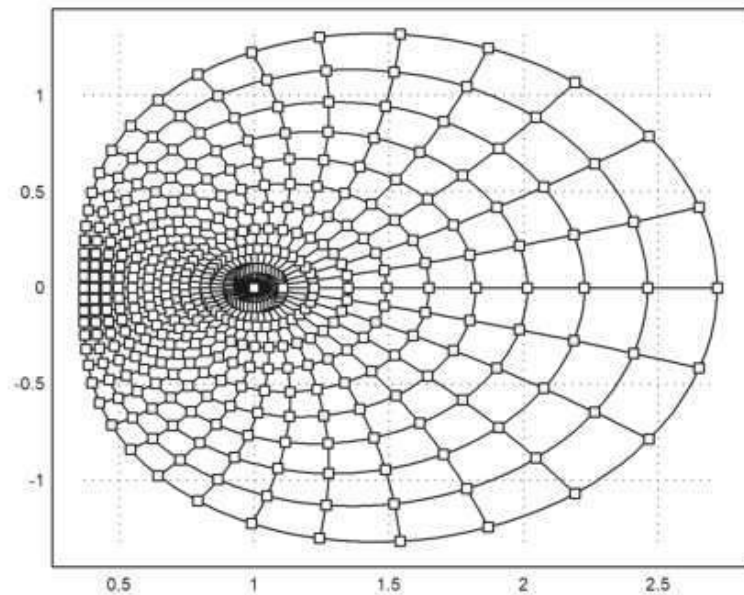
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```



```
>aspect(1.25); r=linspace(0,1,50); a=linspace(0,2pi,200)'; z=r*exp(I*a);
>plot2d(exp(z),cgrid=[40,10]):
```



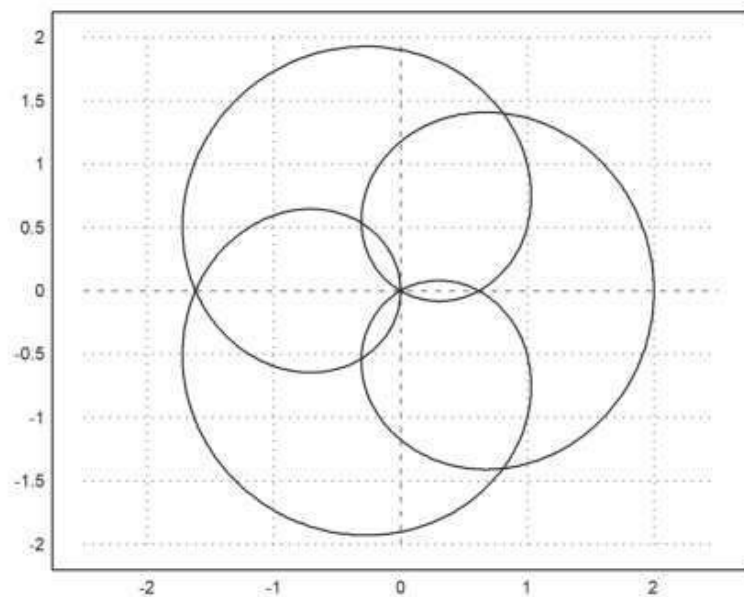
```
>r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=r*exp(I*a);
>plot2d(exp(z),>points,>add):
```



Sebuah vektor bilangan kompleks secara otomatis digambar sebagai kurva pada bidang kompleks dengan bagian real dan bagian imajiner.

Dalam contoh, kita menggambar lingkaran satuan dengan

```
>t=linspace(0,2pi,1000); ...
plot2d(exp(I*t)+exp(4*I*t),r=2):
```



Plot Statistik

Ada banyak fungsi yang dikhususkan untuk plot statistik. Salah satu plot yang sering digunakan adalah plot kolom.

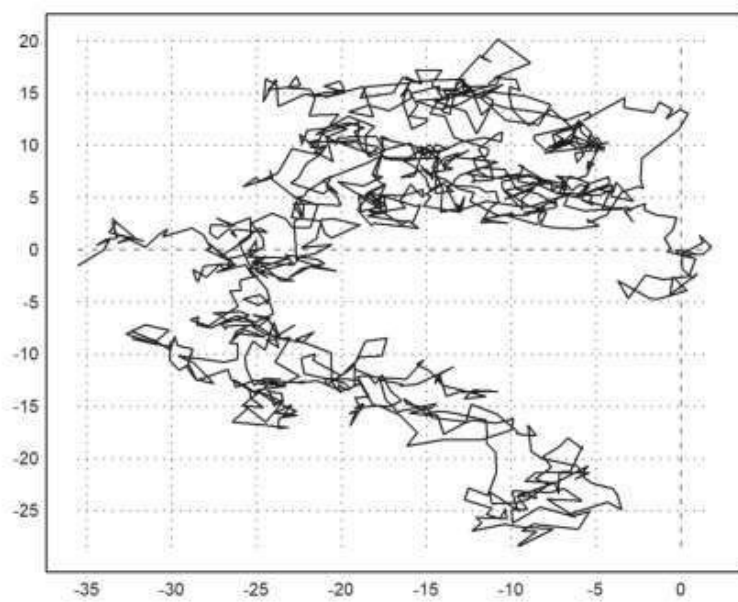
Jumlah kumulatif dari nilai yang terdistribusi normal 0-1 menghasilkan sebuah random walk.

```
>plot2d(cumsum(randnormal(1,1000))):
```

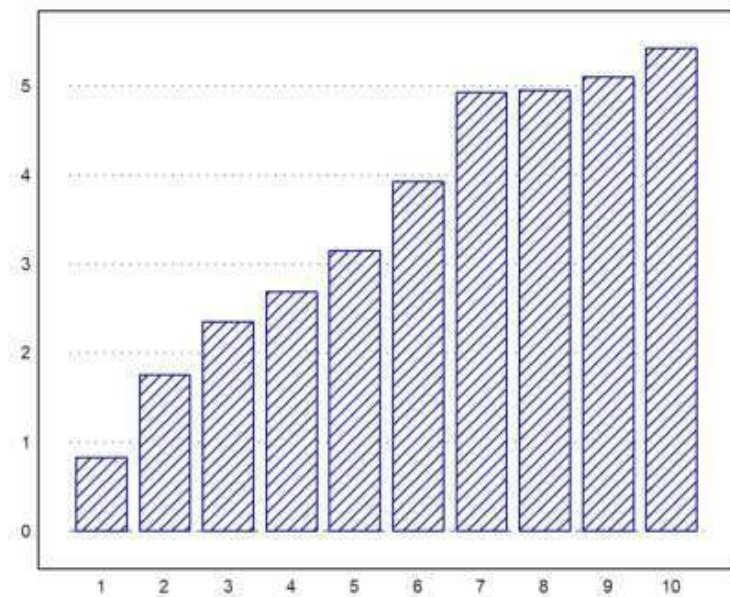


Menggunakan dua baris menunjukkan sebuah lintasan dalam dua dimensi.

```
>X=cumsum(randnormal(2,1000)); plot2d(X[1],X[2]):
```

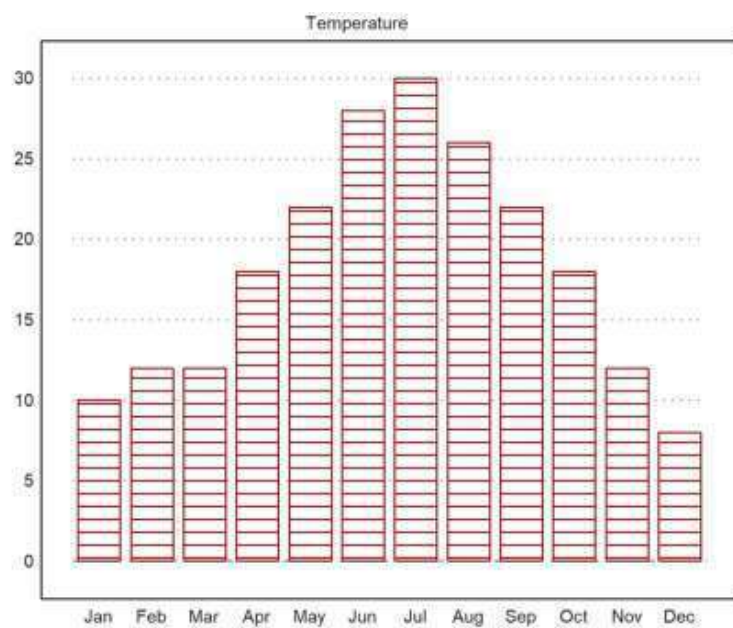


```
>columnspcplot(cumsum(random(10)),style="/",color=blue):
```

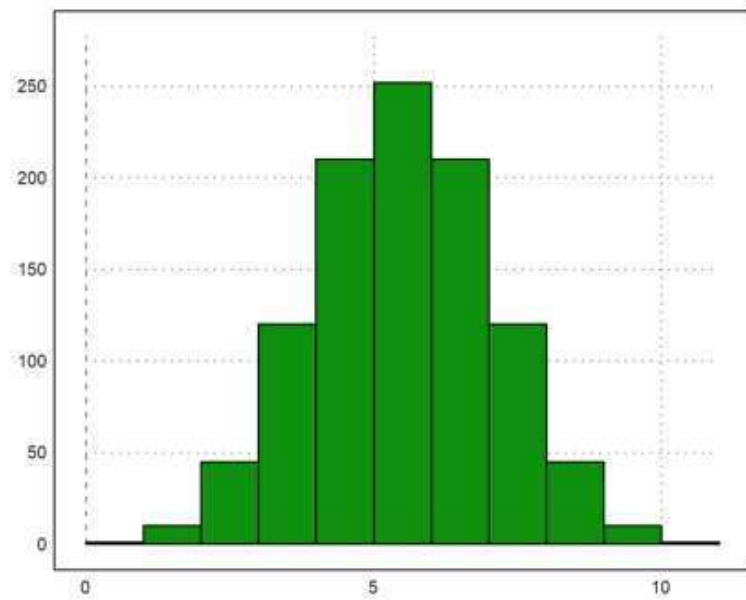


Itu juga dapat menampilkan string sebagai label.

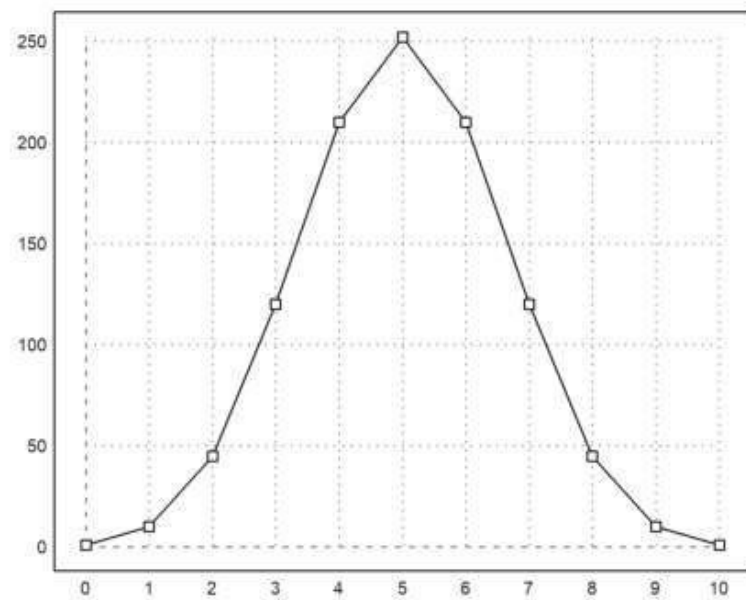
```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
        "Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnsplot(values,lab=months,color=red,style="-");
>title("Temperature"):
```



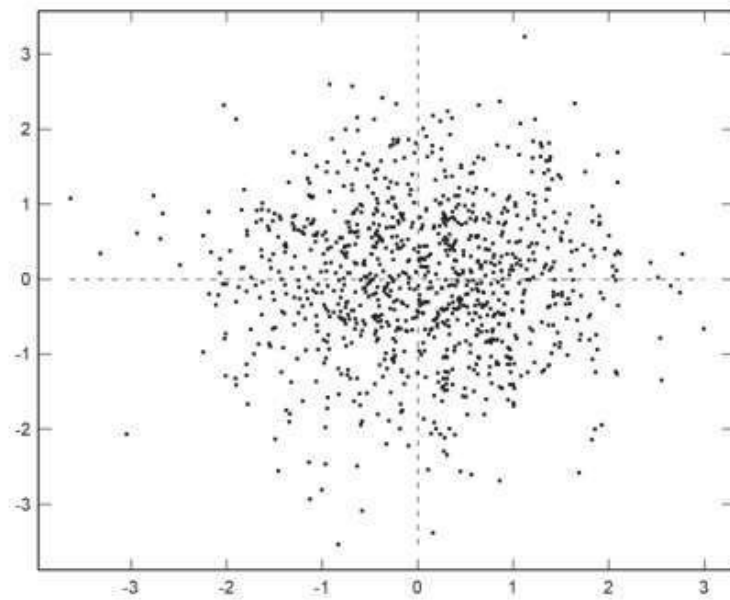
```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```



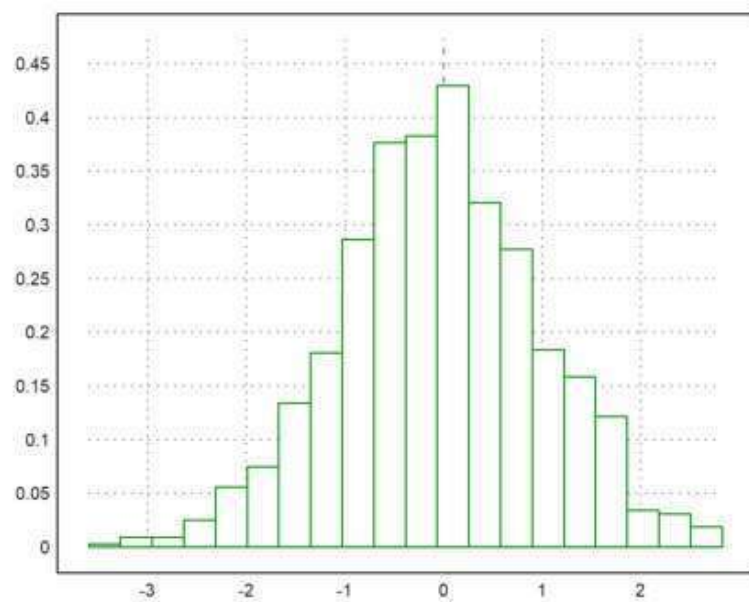
```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



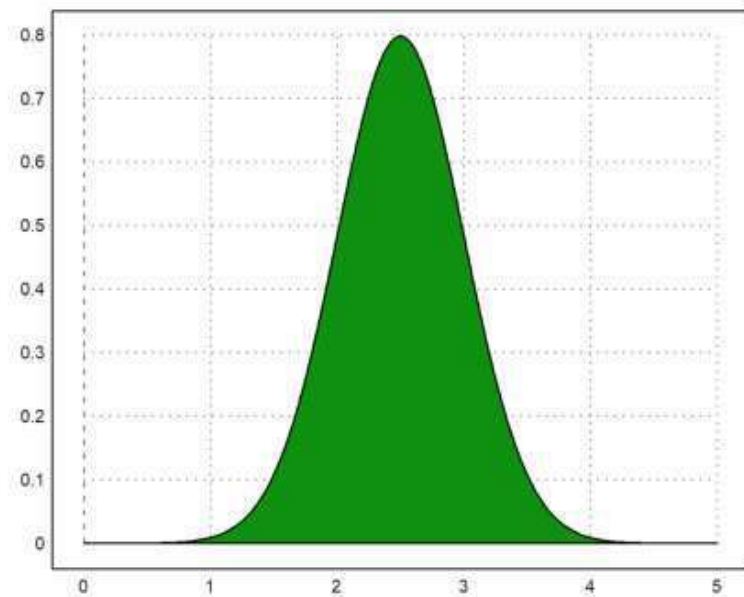
```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```



```
>plot2d(normal(1,1000),>distribution,style="O"):
```

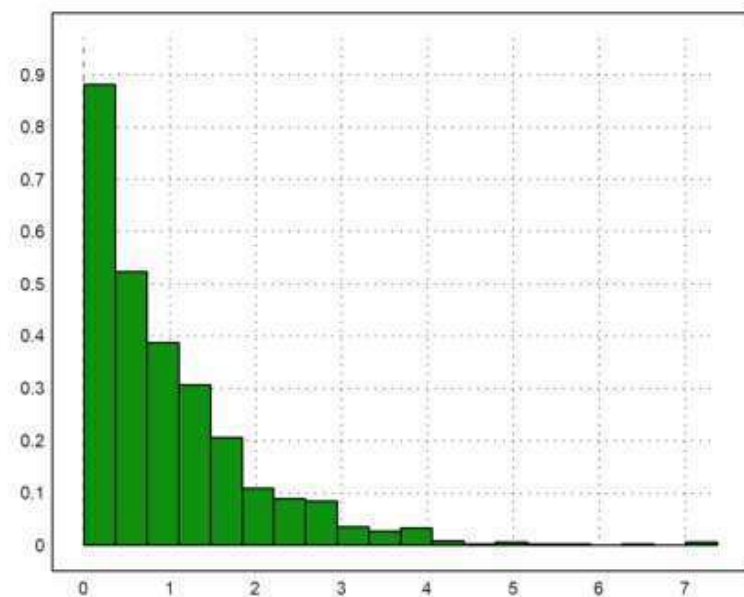


```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



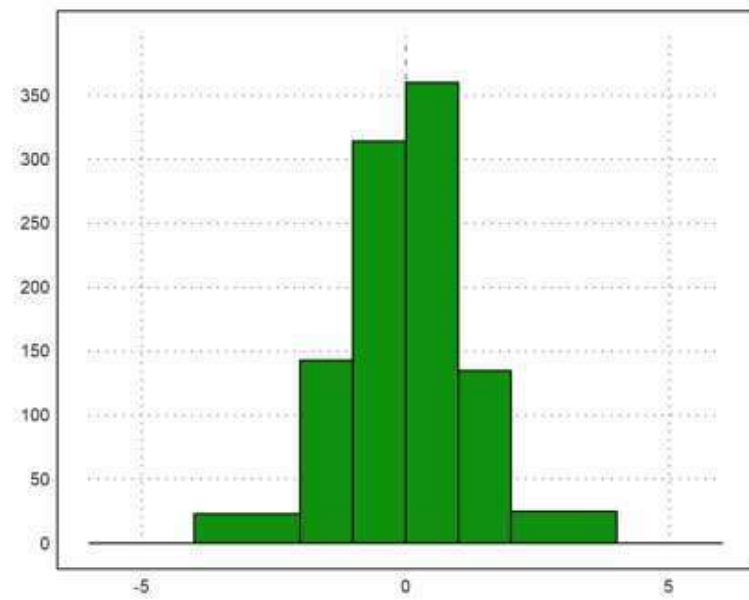
Untuk menggambar distribusi statistik eksperimental, Anda dapat menggunakan `distribution=n` dengan `plot2d`.

```
>w=randexponential(1,1000); // distribusi eksponensial
>plot2d(w,>distribution): // atau distribution=n dengan n interval
```



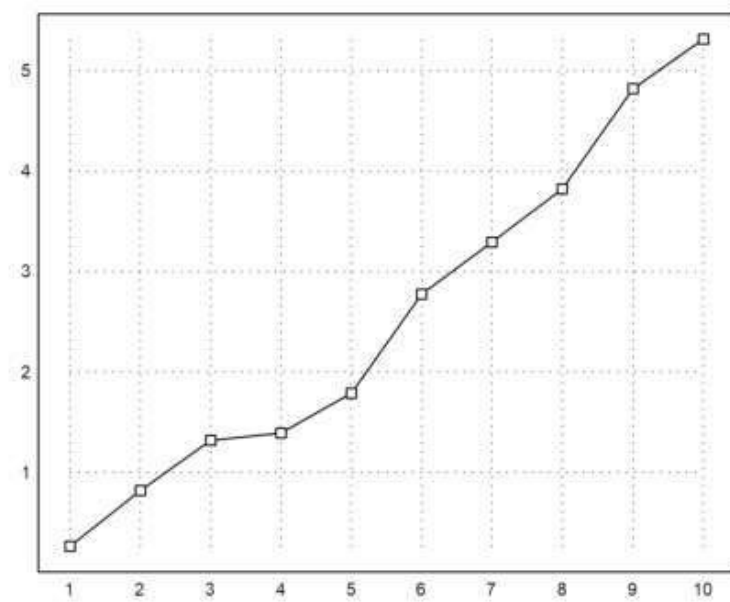
Atau Anda dapat menghitung distribusi dari data dan menggambar hasilnya dengan `>bar` dalam `plot3d`, atau dengan `plot kolom`.

```
>w=normal(1000); // distribusi normal 0-1
>{x,y}=histo(w,10,v=[-6,-4,-2,-1,0,1,2,4,6]); // batas interval v
>plot2d(x,y,>bar):
```

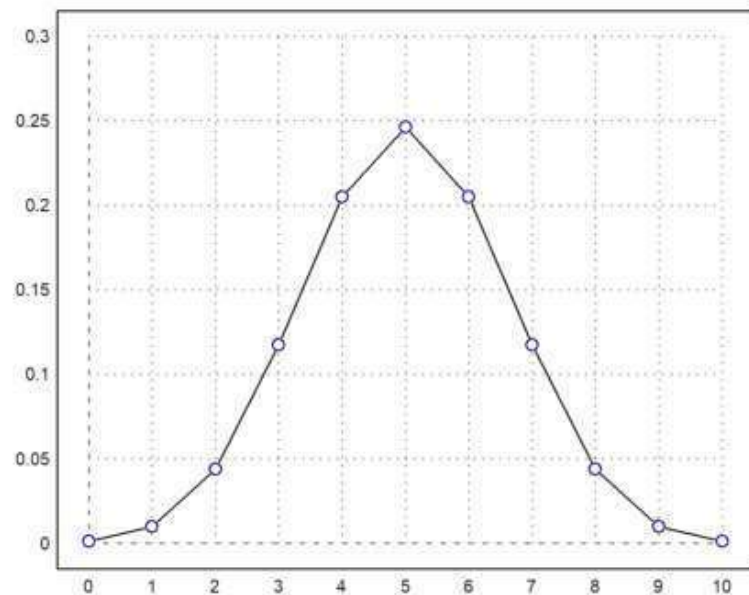


Fungsi `statplot()` mengatur gaya dengan sebuah string sederhana.

```
>statplot(1:10,cumsum(random(10)),"b"):
```



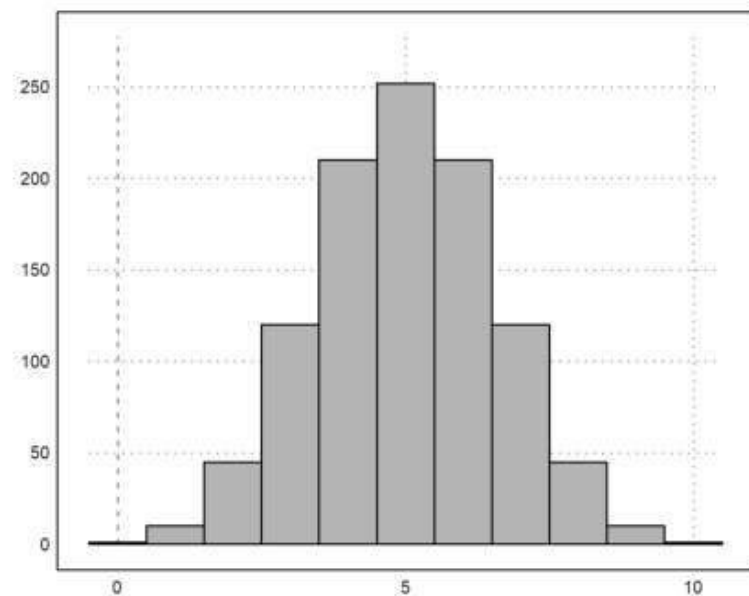
```
>n=10; i=0:n; ...
plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



Selain itu, data dapat digambar sebagai batang. Dalam hal ini, x harus diurutkan dan satu elemen lebih panjang daripada y. Batang akan memanjang dari $x[i]$ hingga $x[i+1]$ dengan nilai $y[i]$. Jika x memiliki ukuran yang sama dengan y, maka akan diperpanjang satu elemen dengan jarak terakhir.

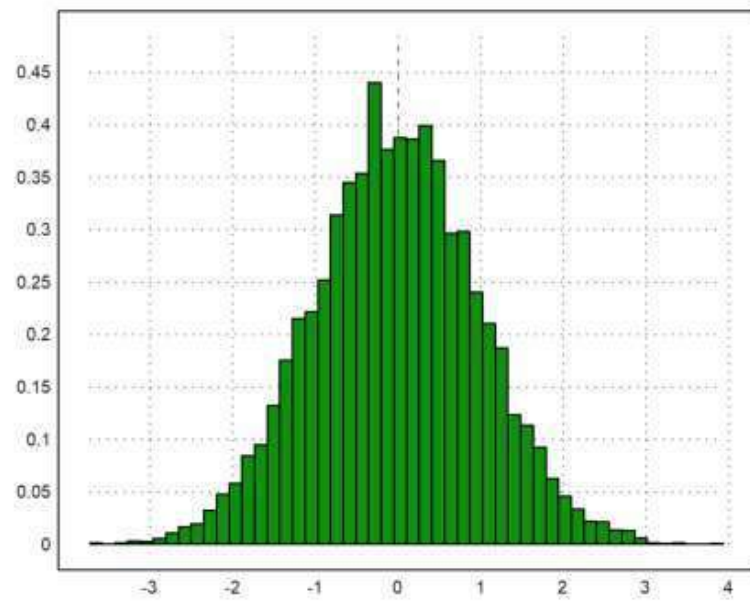
Gaya isian dapat digunakan sama seperti di atas.

```
>n=10; k=bin(n,0:n); ...
plot2d(-0.5:n+0.5,k,bar=true,fillcolor=lightgray):
```

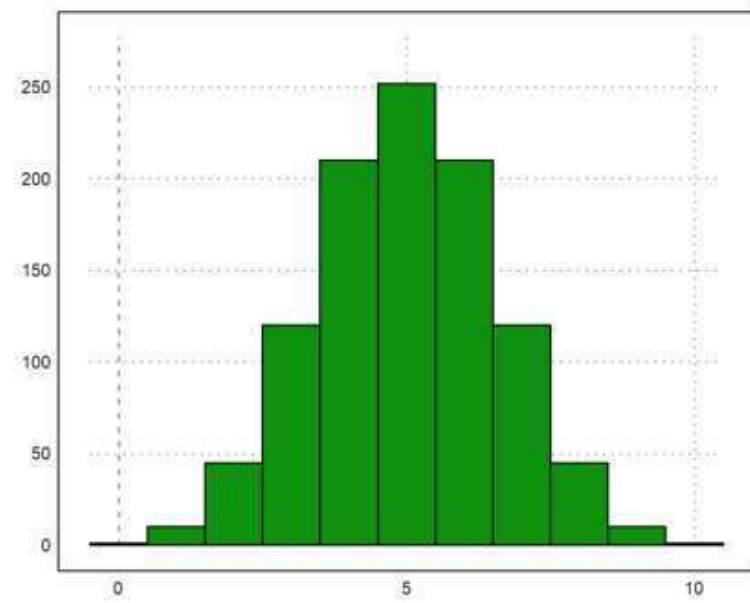


Data untuk plot batang (`bar=1`) dan histogram (`histogram=1`) dapat diberikan secara eksplisit dalam `xv` dan `yv`, atau dapat dihitung dari distribusi empiris dalam `xv` dengan `>distribution` (atau `distribution=n`). Histogram dari nilai `xv` akan dihitung secara otomatis dengan `>histogram`. Jika `>even` ditentukan, nilai `xv` akan dihitung dalam interval bilangan bulat.

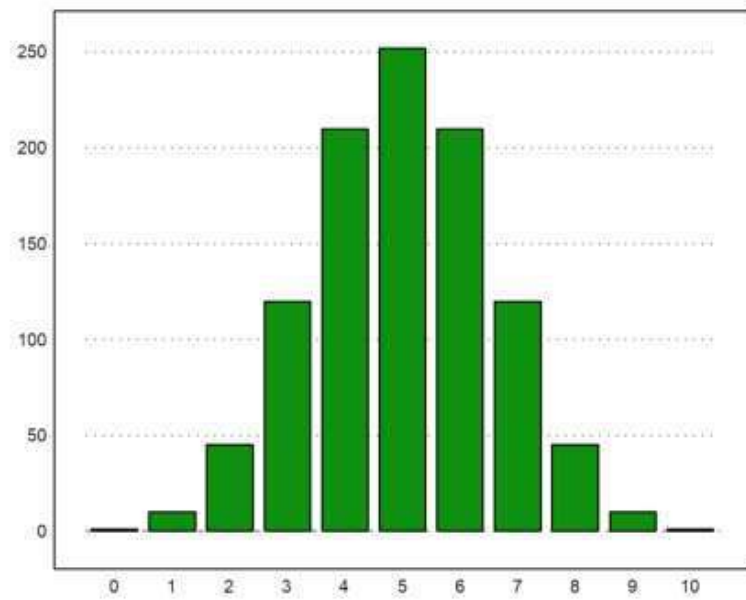
```
>plot2d(normal(10000),distribution=50):
```



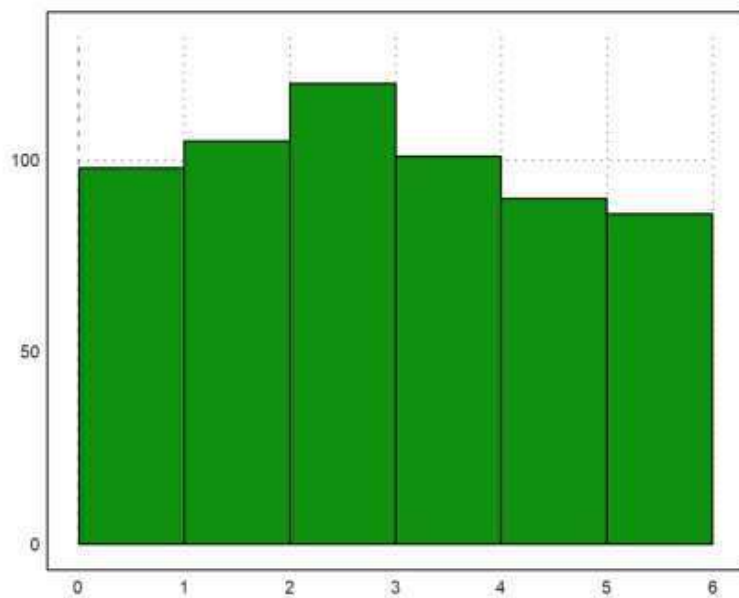
```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```



```
>columnsplo(m,k):
```

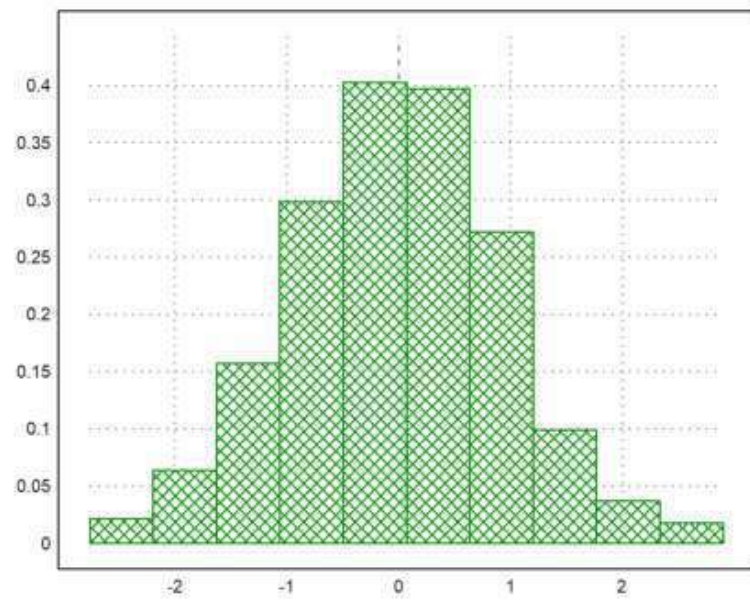


```
>plot2d(random(600)*6,histogram=6):
```



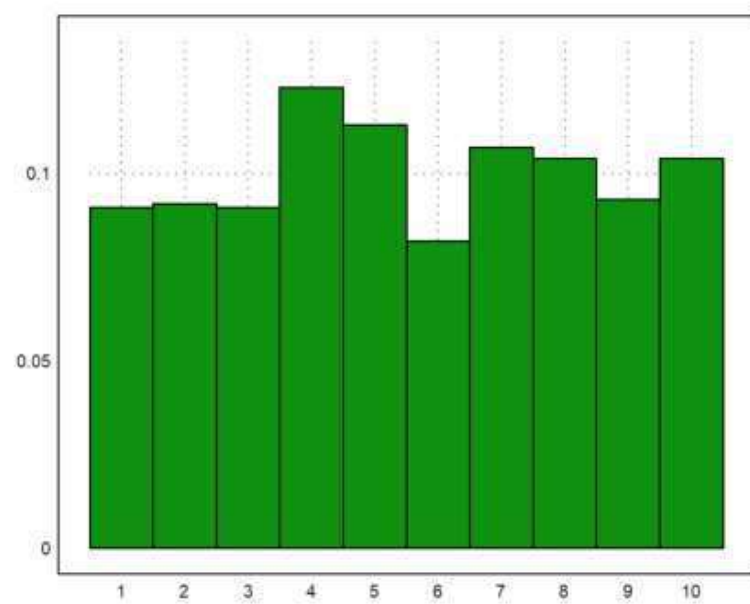
Untuk distribusi, terdapat parameter `distribution=n`, yang menghitung nilai secara otomatis dan menampilkan distribusi relatif dengan `n` sub-interval.

```
>plot2d(normal(1,1000),distribution=10,style="\\/"):
```



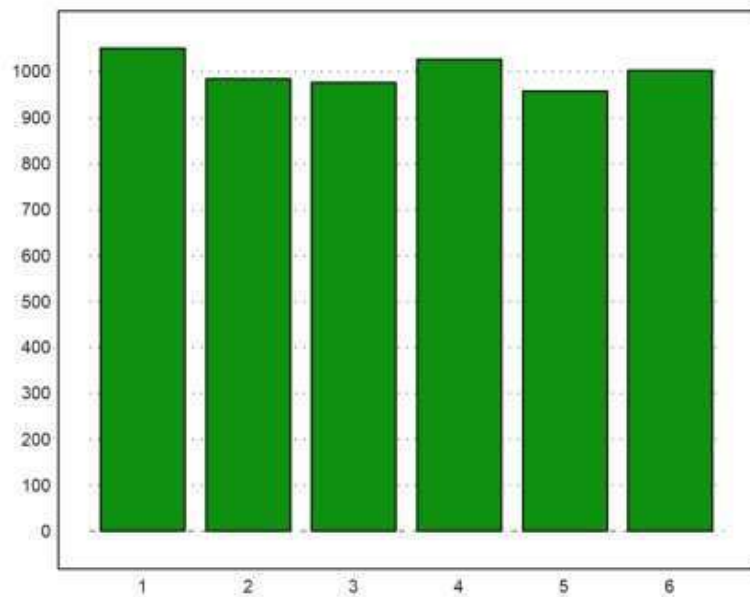
Dengan parameter `even=true`, ini akan menggunakan interval bilangan bulat.

```
>plot2d(intrandom(1,1000,10),distribution=10,even=true):
```

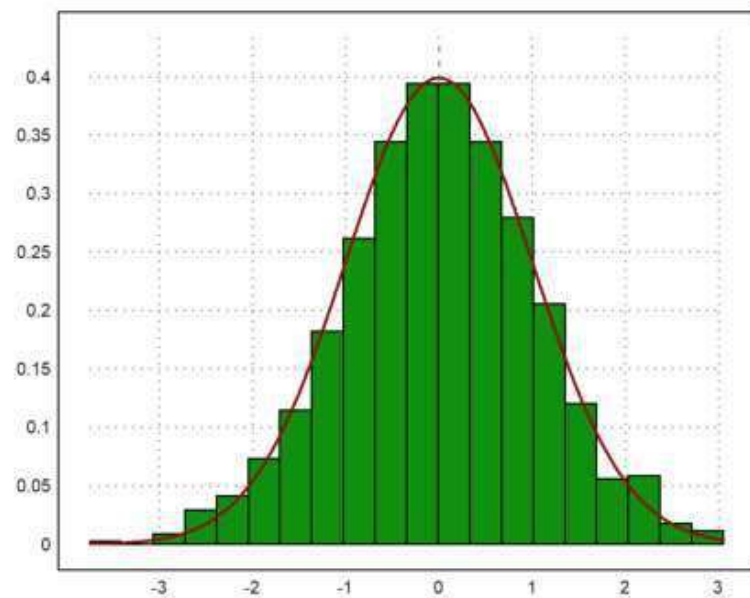


Perlu dicatat bahwa ada banyak plot statistik yang mungkin berguna. Lihatlah tutorial tentang statistik.

```
>columnplot(getmultiplicities(1:6,intrandom(1,6000,6))):
```

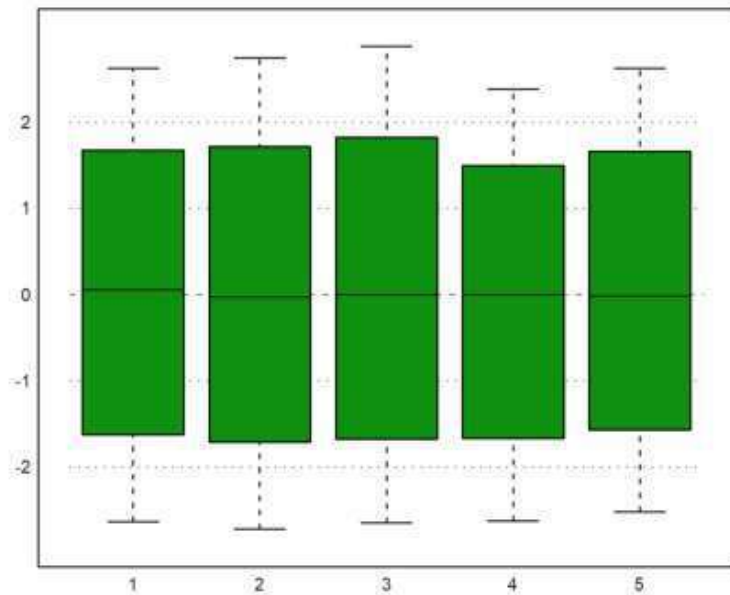


```
>plot2d(normal(1,1000),>distribution); ...
  plot2d("qnormal(x)",color=red,thickness=2,>add):
```



Ada juga banyak plot khusus untuk statistik. Sebuah boxplot menampilkan kuartil dari distribusi ini dan banyak outlier. Menurut definisinya, outlier dalam boxplot adalah data yang melebihi 1,5 kali rentang 50% tengah dari plot.

```
>M=normal(5,1000); boxplot(quartiles(M)):
```



Fungsi Implisit

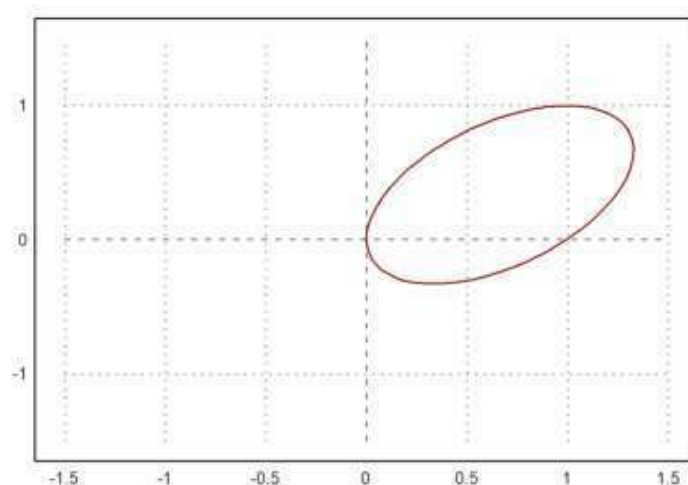
Plot implisit menampilkan garis level yang menyelesaikan $f(x,y)=\text{level}$, di mana "level" dapat berupa satu nilai atau sebuah vektor nilai. Jika $\text{level}=\text{"auto"}$, akan ada n_c garis level, yang tersebar merata antara nilai minimum dan maksimum dari fungsi tersebut. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan $>\text{hue}$ untuk menunjukkan nilai dari fungsi. Untuk fungsi implisit, xv harus berupa sebuah fungsi atau ekspresi dari parameter x dan y , atau sebagai alternatif, xv dapat berupa sebuah matriks nilai.

Euler dapat menandai garis level

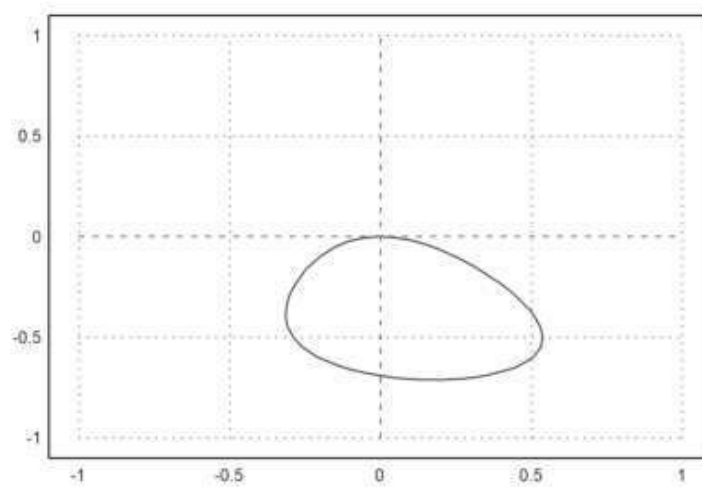
dari fungsi apa pun.

Untuk menggambar himpunan $f(x,y)=c$ untuk satu atau lebih konstanta c , Anda dapat menggunakan `plot2d()` dengan plot implisitnya di bidang. Parameter untuk c adalah $\text{level}=c$, di mana c dapat berupa vektor garis level. Selain itu, sebuah skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi pada setiap titik dalam plot. Parameter " n " menentukan tingkat ketelitian plot.

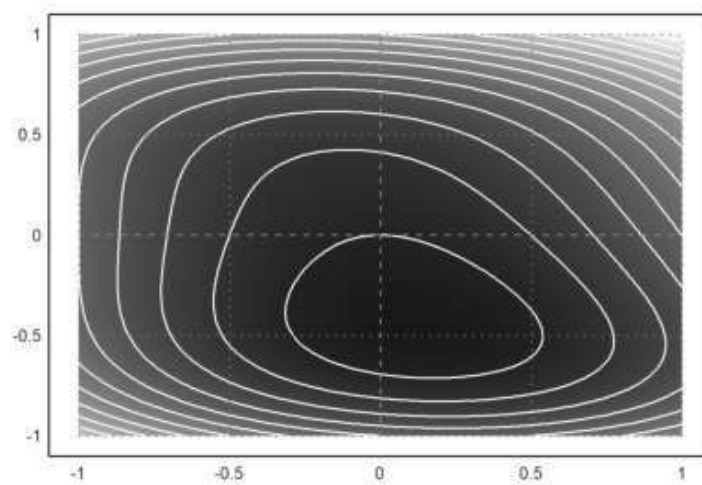
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```



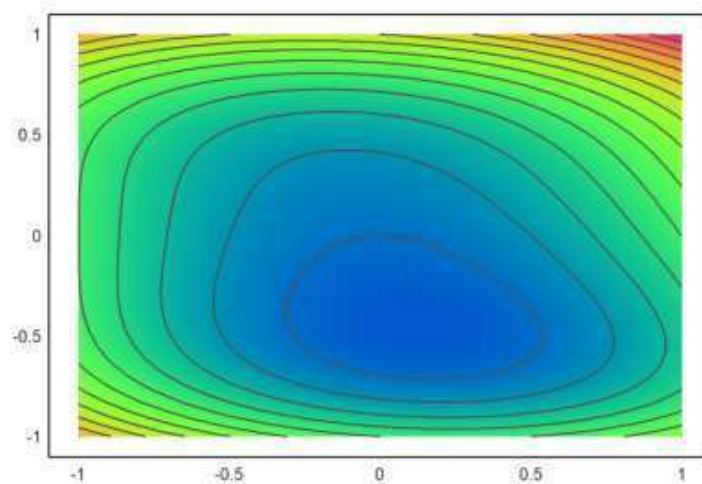
```
>expr := "2*x^2+x*y+3*y^4+y"; // definisikan sebuah ekspresi f(x,y)
>plot2d(expr,level=0): // Solusi dari f(x,y)=0
```



```
>plot2d(expr,level=0:0.5:20,>hue,contourcolor=white,n=200): // bagus
```

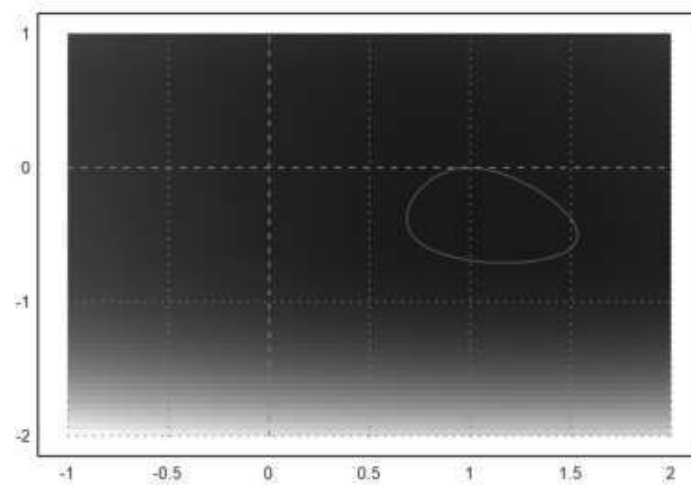


```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // lebih bagus
```

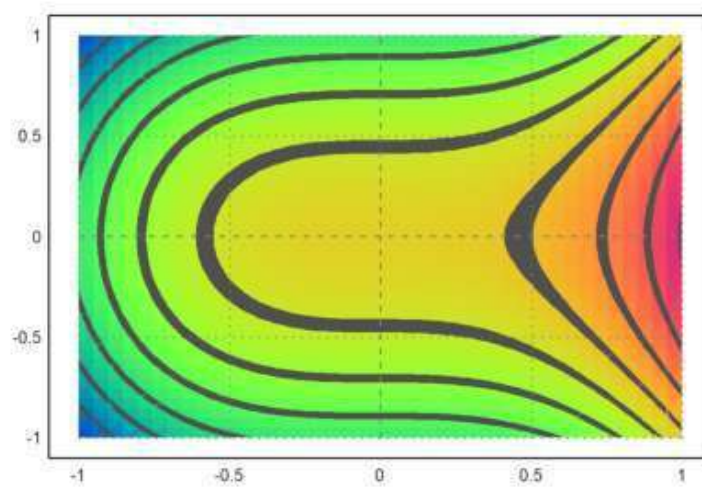


Ini juga berfungsi untuk plot data. Namun, Anda harus menentukan rentang untuk label sumbu.

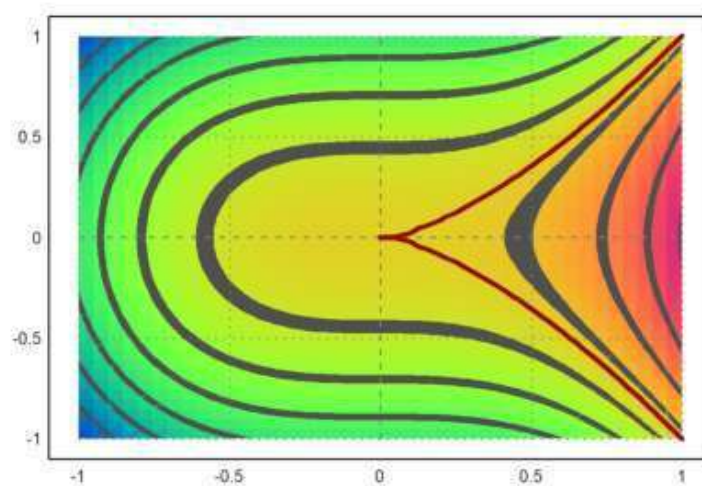
```
>x=-2:0.05:1; y=x'; z=expr(x,y);  
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



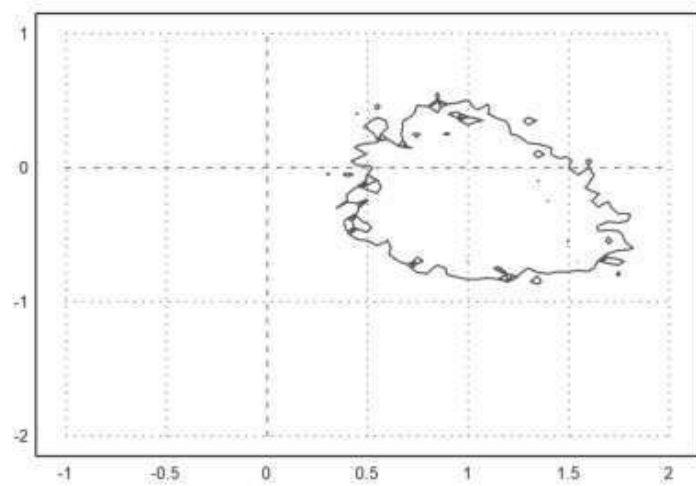
```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



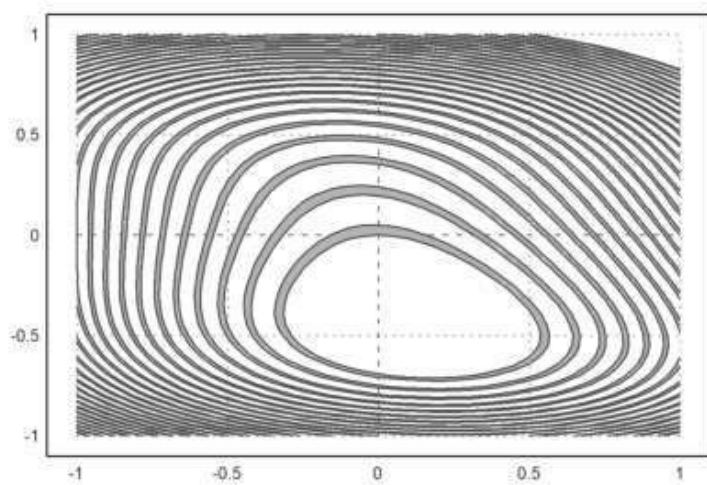
```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



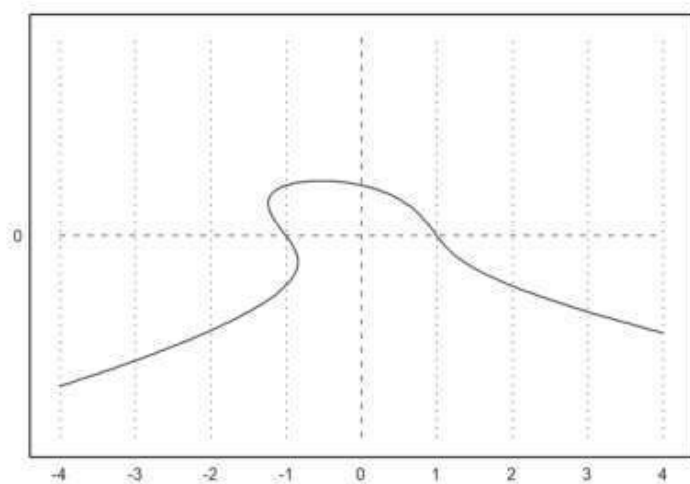
```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



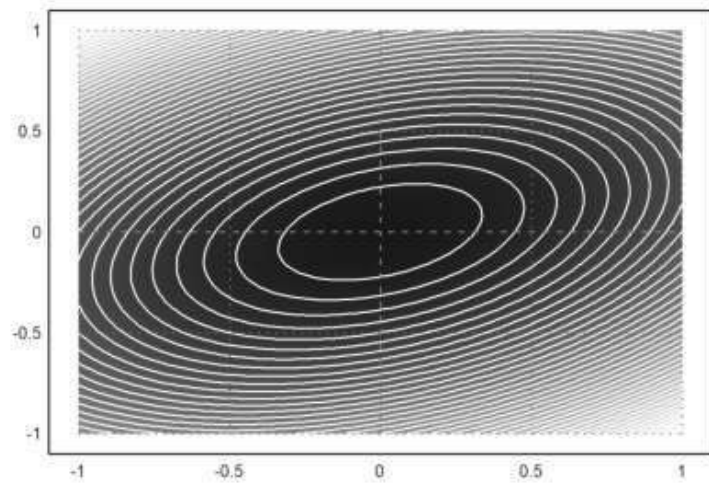
```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```



```
>plot2d("x^2+y^3+x*y",level=1,r=4,n=100):
```



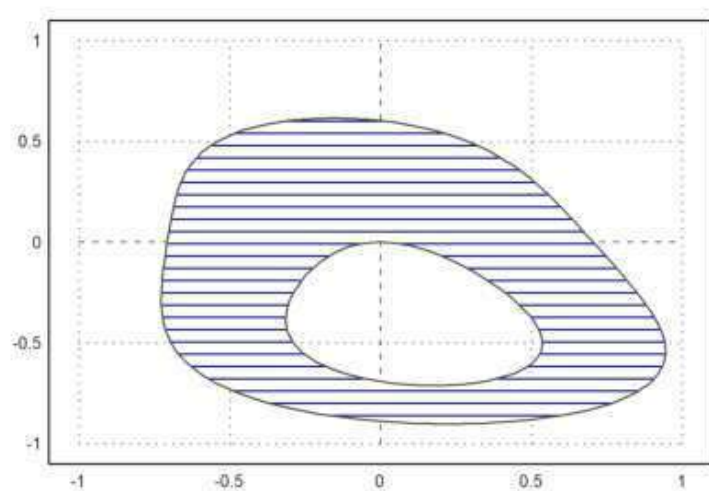
```
>plot2d("x^2+2*y^2-x*y",level=0:0.1:10,n=100,contourcolor=white,>hue):
```



Juga memungkinkan untuk mengisi himpunan dengan rentang level.

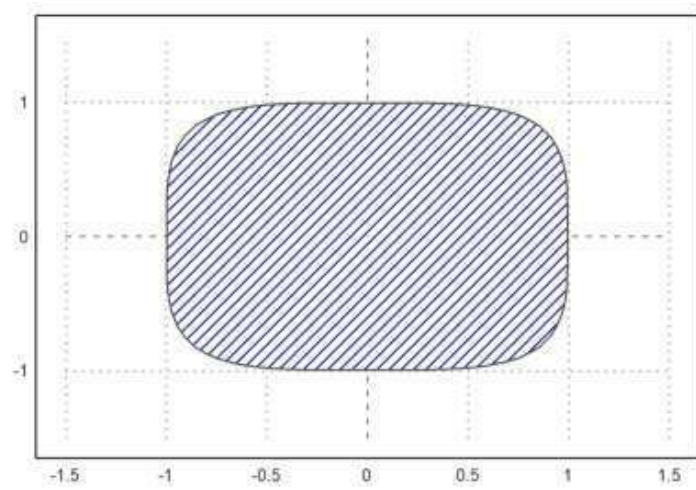
Dimungkinkan untuk mengisi daerah nilai untuk suatu fungsi tertentu. Untuk ini, level harus berupa matriks $2 \times n$. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

```
>plot2d(expr,level=[0;1],style="-",color=blue): //  $0 \leq f(x,y) \leq 1$ 
```

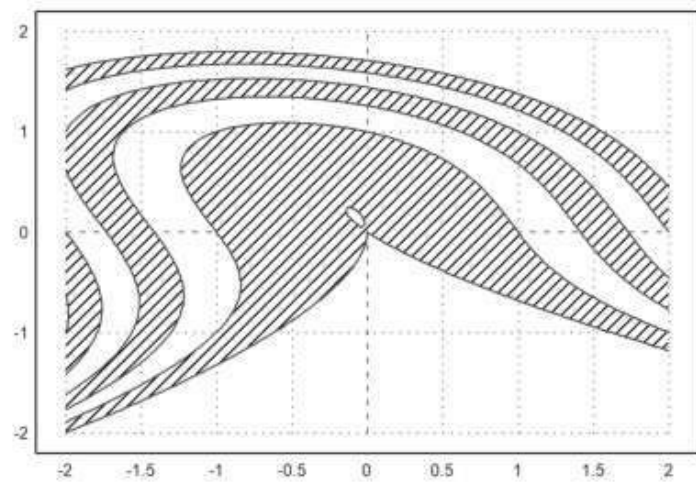


Plot implisit juga dapat menampilkan rentang level. Dalam hal ini, level harus berupa matriks $2 \times n$ dari interval level, di mana baris pertama berisi awal dan baris kedua berisi akhir setiap interval. Sebagai alternatif, sebuah vektor baris sederhana dapat digunakan untuk level, dan parameter `dl` memperluas nilai level menjadi interval.

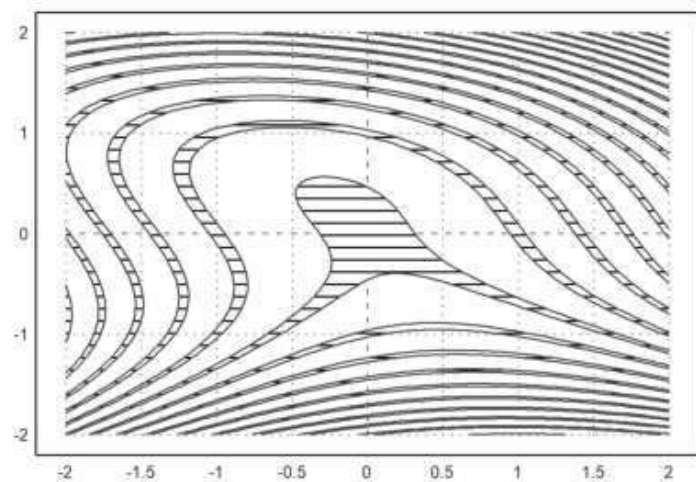
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



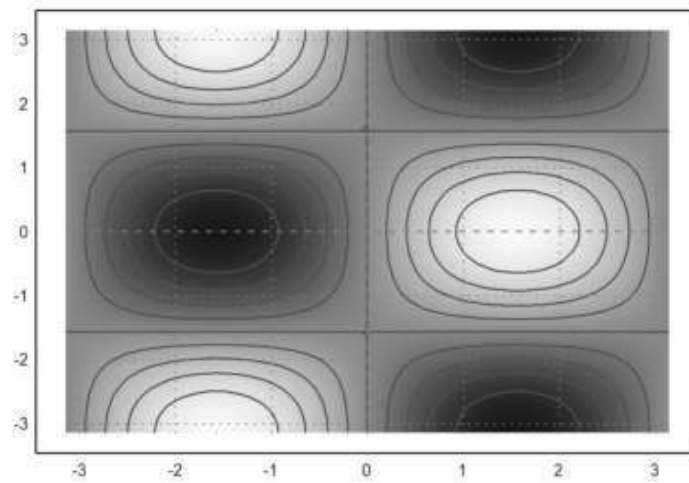
```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```



```
>plot2d("x^2+y^3+x*y",level=-10:20,r=2,style="-",dl=0.1,n=100):
```



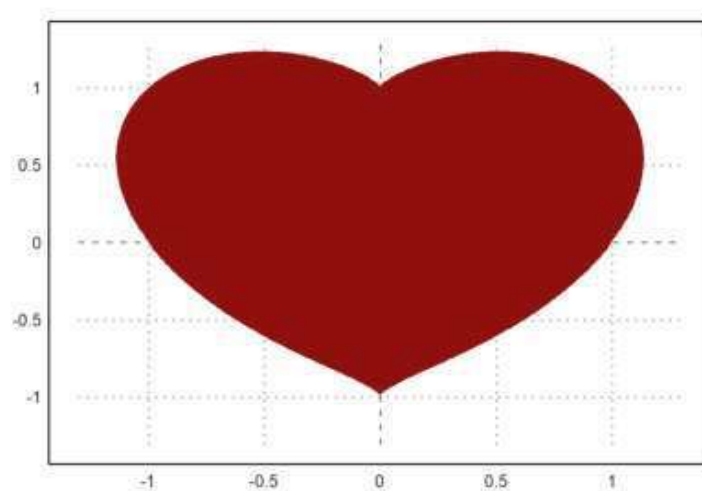
```
>plot2d("sin(x)*cos(y)",r=pi,>hue,>levels,n=100):
```



Juga dimungkinkan untuk menandai sebuah daerah

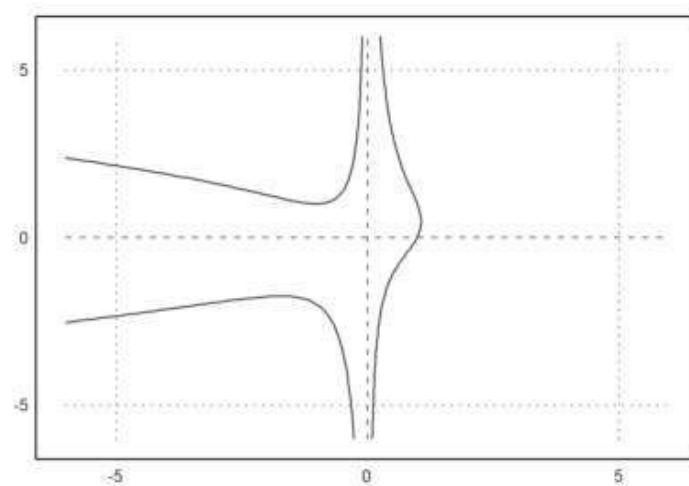
Hal ini dilakukan dengan menambahkan level dengan dua baris.

```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
  style="#",color=red,<outline, ...
  level=[-2;0],n=100):
```



Dimungkinkan untuk menentukan level tertentu. Misalnya, kita dapat menggambar solusi dari sebuah persamaan seperti

```
>plot2d("x^3-x*y+x^2*y^2",r=6,level=1,n=100):
```



```

>function starplot1 (v, style="/", color=green, lab=none) ...
    if !holding() then clg; endif;
    w=window(); window(0,0,1024,1024);
    h=holding(1);
    r=max(abs(v))*1.2;
    setplot(-r,r,-r,r);
    n=cols(v); t=linspace(0,2pi,n);
    v=v|v[1]; c=v*cos(t); s=v*sin(t);
    cl=barcolor(color); st=barstyle(style);
    loop 1 to n
        polygon([0,c[#],c[#+1]], [0,s[#],s[#+1]],1);
        if lab!=none then
            rlab=v[#]+r*0.1;
            {col,row}=toscreen(cos(t[#])*rlab,sin(t[#])*rlab);
            ctext(""+lab[#],col,row-textheight()/2);
        endif;
    end;
    barcolor(cl); barstyle(st);
    holding(h);
    window(w);
endfunction

```

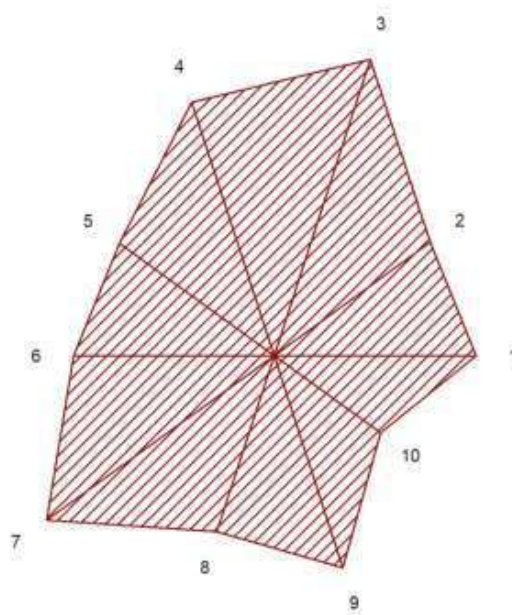
Di sini tidak ada grid atau tanda sumbu. Selain itu, kita menggunakan seluruh jendela untuk plot.

Kita memanggil reset sebelum menguji plot ini untuk mengembalikan pengaturan default grafik. Hal ini tidak diperlukan jika Anda yakin bahwa plot Anda berfungsi.

```

>reset; starplot1(normal(1,10)+5,color=red,lab=1:10):

```



Terkadang, Anda mungkin ingin menggambar sesuatu yang hampir bisa dilakukan oleh plot2d, tetapi tidak sepenuhnya.

Dalam fungsi berikut, kita membuat plot impuls logaritmik. plot2d dapat membuat plot logaritmik, tetapi tidak untuk batang impuls.

```

>function logimpulseplot1 (x,y) ...
    {x0,y0}=makeimpulse(x,log(y)/log(10));
    plot2d(x0,y0,>bar,grid=0);
    h=holding(1);
    frame();
    xgrid(ticks(x));
    p=plot();

```

```

for i=-10 to 10;
    if i<=p[4] and i>=p[3] then
        ygrid(i,yt="10^"+i);
    endif;
end;
holding(h);
endfunction

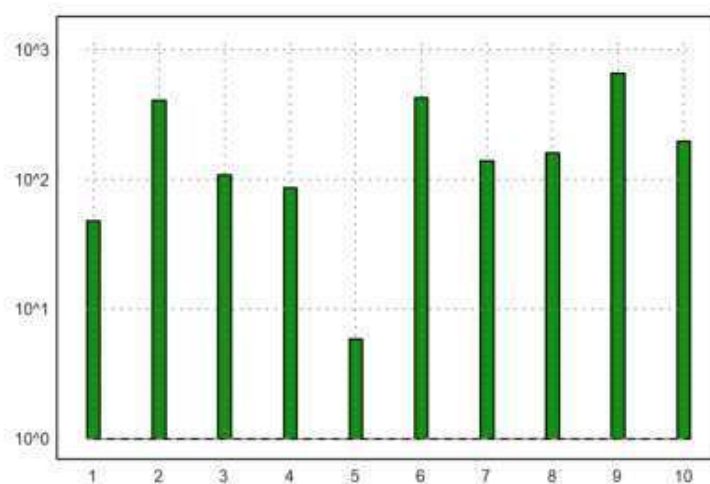
```

Mari kita uji dengan nilai yang terdistribusi secara eksponensial.

```

>aspect(1.5); x=1:10; y=-log(random(size(x)))*200; ...
logimpulseplot1(x,y):

```



Mari kita animasikan sebuah kurva 2D menggunakan plot langsung. Perintah plot(x,y) hanya menggambar sebuah kurva ke dalam jendela plot. setplot(a,b,c,d) mengatur jendela ini.

Fungsi wait(0) memaksa plot muncul di jendela grafik. Jika tidak, pembaruan ulang terjadi pada interval waktu yang jarang.

```

>function animliss (n,m) ...
t=linspace(0,2pi,500);
f=0;
c=framecolor(0);
l=linewidth(2);
setplot(-1,1,-1,1);
repeat
    clg;
    plot(sin(n*t),cos(m*t+f));
    wait(0);
    if testkey() then break; endif;
    f=f+0.02;
end;
framecolor(c);
linewidth(l);
endfunction

```

Tekan sembarang tombol untuk menghentikan animasi ini.

```

>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER

```

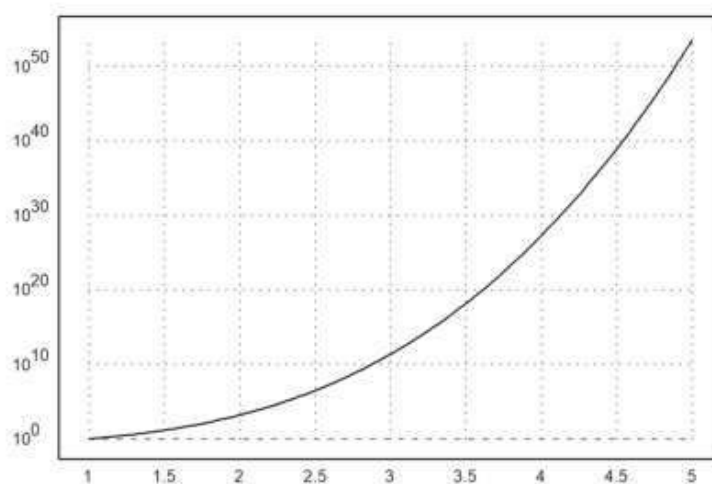
Plot Logaritmik

EMT menggunakan parameter "logplot" untuk skala logaritmik.

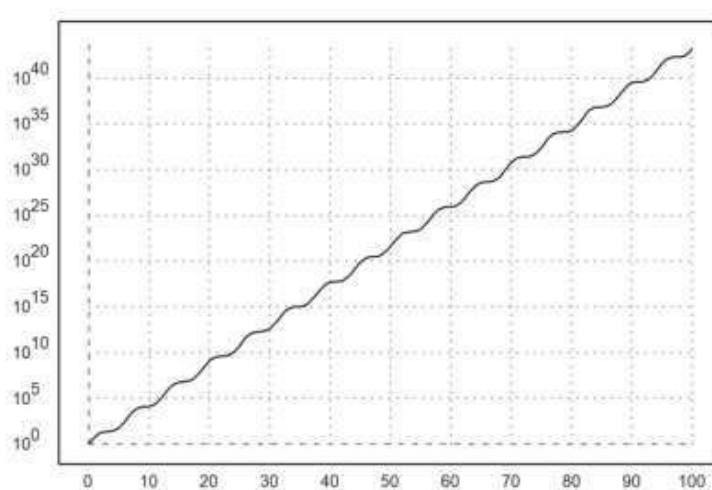
Plot logaritmik dapat digambar dengan menggunakan skala logaritmik pada y dengan logplot=1, atau menggunakan skala logaritmik pada x dan y dengan logplot=2, atau pada x dengan logplot=3.

- logplot=1: y-logaritmik
- logplot=2: x-y-logaritmik
- logplot=3: x-logaritmik

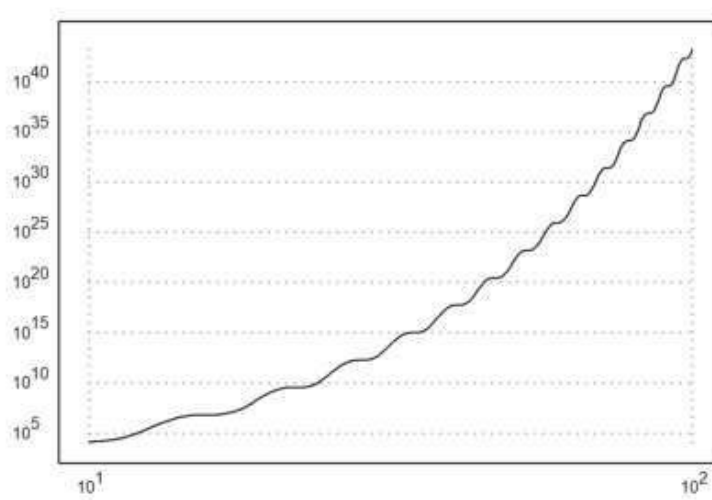
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



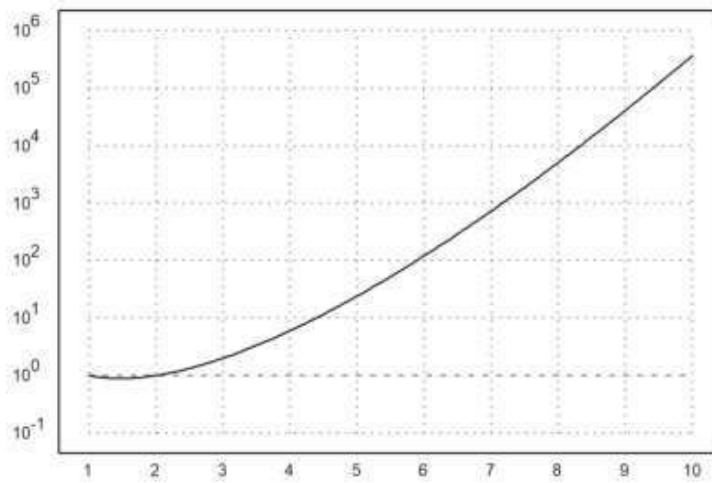
```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



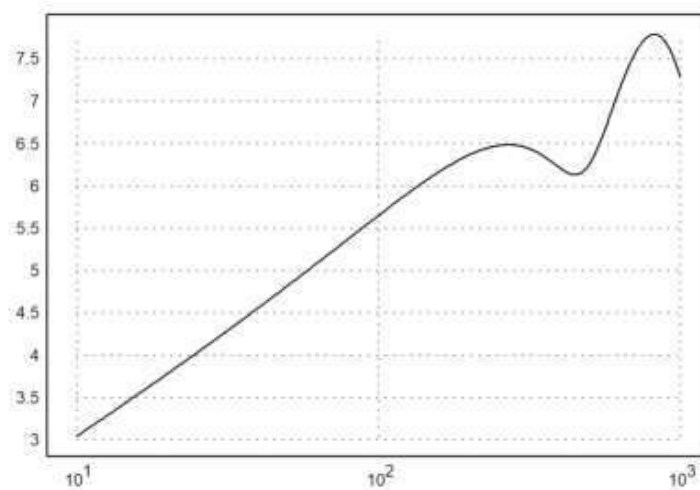
```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```



```
>plot2d("gamma(x)",1,10,logplot=1):
```

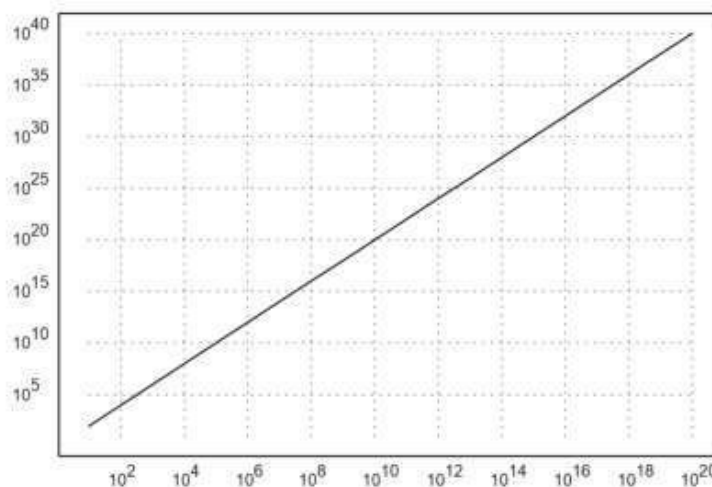


```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Ini juga berfungsi untuk plot data.

```
>x=10^(1:20); y=x^2-x;
>plot2d(x,y,logplot=2):
```



Rujukan Lengkap Fungsi plot2d()

```
function plot2d (xv, yv, btest, a, b, c, d, xmin, xmax, r, n, ..
logplot, grid, frame, framecolor, square, color, thickness, style, ..
auto, add, user, delta, points, addpoints, pointstyle, bar, histogram, ..
```

distribution, even, steps, own, adaptive, hue, level, contour, ..
nc, filled, fillcolor, outline, title, xl, yl, maps, contourcolor, ..
contourwidth, ticks, margin, clipping, cx, cy, insimg, spectral, ..
cgrid, vertical, smaller, dl, niveau, levels)

Fungsi plot serbaguna untuk plot dalam bidang (plot 2D). Fungsi ini dapat melakukan plot fungsi dari satu variabel, plot data, kurva dalam bidang, plot batang, grid bilangan kompleks, dan plot implisit dari fungsi dua variabel.

Parameter

x,y : persamaan, fungsi, atau vektor data
a,b,c,d : area plot (default a=-2,b=2)
r : jika r diatur, maka a=cx-r, b=cx+r, c=cy-r, d=cy+r
r bisa berupa vektor [rx,ry] atau vektor [rx1,rx2,ry1,ry2].
xmin,xmax : rentang parameter untuk kurva
auto : Tentukan rentang-y secara otomatis (default)
square : jika true, coba pertahankan rentang x-y berbentuk persegi
n : jumlah interval (default adaptif)
grid : 0 = tanpa grid dan label,
1 = hanya sumbu,
2 = grid normal (lihat di bawah untuk jumlah garis grid)
3 = sumbu di dalam
4 = tanpa grid
5 = grid penuh termasuk margin
6 = ticks di bingkai
7 = hanya sumbu
8 = hanya sumbu, sub-ticks
frame : 0 = tanpa bingkai
framecolor: warna bingkai dan grid
margin : angka antara 0 dan 0.4 untuk margin di sekitar plot
color : Warna kurva. Jika ini adalah vektor warna,
akan digunakan untuk setiap baris matriks plot. Dalam kasus
plot titik, sebaiknya berupa vektor kolom. Jika vektor baris atau matriks penuh warna digunakan untuk plot
titik, akan dipakai untuk setiap titik data.
thickness : ketebalan garis untuk kurva
Nilai ini bisa lebih kecil dari 1 untuk garis yang sangat tipis.
style : Gaya plot untuk garis, marker, dan isian.
Untuk titik gunakan
"\[", "<>", ":", ":", "...",
"*", "+", "|", ":", "O"
"\[#", "<>#", "o#" (bentuk terisi)
"\[w", "<>w", "ow" (tidak transparan)
Untuk garis gunakan
"-", "--", "-.", ":", "-.-", "-.-", "->"
Untuk poligon terisi atau plot batang gunakan
"#", "#O", "O", "/", "", "/",
"+", "|", ":", "t"
points : plot titik tunggal alih-alih segmen garis
addpoints : jika true, plot segmen garis dan titik
add : tambahkan plot ke plot yang ada
user : aktifkan interaksi pengguna untuk fungsi
delta : ukuran langkah untuk interaksi pengguna
bar : plot batang (x adalah batas interval, y adalah nilai interval)
histogram : plot frekuensi x dalam n subinterval
distribution=n : plot distribusi x dengan n subinterval
even : gunakan nilai antar untuk histogram otomatis
steps : plot fungsi sebagai fungsi langkah (steps=1,2)
adaptive : gunakan plot adaptif (n adalah jumlah langkah minimal)
level : plot garis level dari fungsi implisit dua variabel
outline : menggambar batas dari rentang level.

Jika nilai level berupa matriks 2xn, rentang level akan digambar dalam warna menggunakan gaya isian yang diberikan. Jika outline true, akan digambar dalam warna kontur. Dengan fitur ini, daerah f(x,y) antara batas bisa ditandai.

hue : tambahkan warna hue ke plot level untuk menunjukkan nilai fungsi

contour : Gunakan plot level dengan level otomatis

nc : jumlah garis level otomatis

title : judul plot (default "")

xl, yl : label untuk sumbu x dan y

smaller : jika >0, akan ada lebih banyak ruang di kiri untuk label.

vertical :

Mengaktifkan atau menonaktifkan label vertikal. Ini mengubah variabel global verticallabels secara lokal untuk satu plot. Nilai 1 hanya menggunakan teks vertikal, nilai 2 menggunakan label numerik vertikal pada sumbu y.

filled : isi plot kurva

fillcolor : warna isi untuk batang dan kurva terisi

outline : batas untuk poligon terisi

logplot : atur plot logaritmik

1 = logplot di y,

2 = logplot di xy,

3 = logplot di x

own :

String yang menunjuk ke rutinitas plot sendiri. Dengan >user, Anda mendapatkan interaksi pengguna yang sama seperti di plot2d. Rentang akan diatur sebelum setiap pemanggilan fungsi Anda.

maps : map ekspresi (0 lebih cepat), fungsi selalu dimap.

contourcolor : warna garis kontur

contourwidth : lebar garis kontur

clipping : nyalakan/matikan clipping (default true)

title :

Ini bisa digunakan untuk menjelaskan plot. Judul akan muncul di atas

plot. Selain itu, label untuk sumbu x dan y bisa ditambahkan dengan

xl="string" atau yl="string". Label lain bisa ditambahkan dengan

fungsi label() atau labelbox(). Judul bisa berupa string unicode

atau gambar formula LaTeX.

cgrid :

Menentukan jumlah garis grid untuk plot grid kompleks.

Harus merupakan pembagi dari ukuran matriks dikurangi 1 (jumlah subinterval). cgrid bisa berupa vektor [cx,cy].

Ikhtisar

Fungsi ini dapat memplot

- ekspresi, koleksi pemanggilan atau fungsi dari satu variabel,
- kurva parametrik,
- data x terhadap data y,
- fungsi implisit,
- plot batang,
- grid kompleks,
- poligon.

Jika fungsi atau ekspresi untuk xv diberikan, plot2d() akan

menghitung

nilai dalam rentang yang diberikan menggunakan fungsi atau ekspresi tersebut.

Ekspresi harus berupa ekspresi dalam variabel x. Rentang harus didefinisikan dalam parameter a dan b kecuali jika rentang default [-2,2] digunakan. Rentang y akan dihitung secara otomatis, kecuali c dan d ditentukan, atau radius r, yang menghasilkan rentang [-r,r] untuk x dan y. Untuk plot fungsi, plot2d secara default akan

menggunakan evaluasi adaptif dari fungsi. Untuk mempercepat plot fungsi rumit, matikan dengan <adaptive, dan opsional kurangi jumlah interval n. Selain itu, plot2d() secara default akan menggunakan mapping. Artinya, ia akan menghitung plot elemen demi elemen.

Jika ekspresi atau fungsi Anda bisa menangani vektor x,

Anda bisa mematakannya dengan <maps untuk evaluasi lebih cepat.

Perhatikan bahwa plot adaptif selalu dihitung elemen demi elemen. Jika fungsi atau ekspresi untuk xv dan yv ditentukan, plot2d() akan menghitung kurva dengan nilai xv sebagai koordinat x dan nilai yv sebagai koordinat y. Dalam hal ini, rentang harus didefinisikan untuk parameter menggunakan xmin, xmax. Ekspresi yang

ada

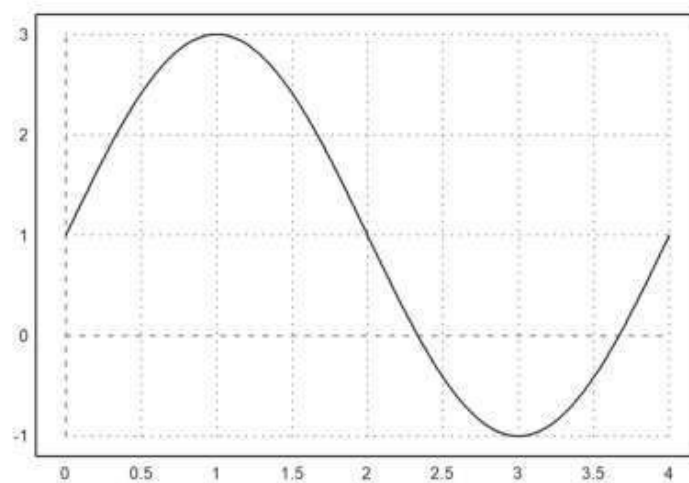
dalam string harus selalu berupa ekspresi dalam variabel parameter x.

Contoh Soal

1. Sketsakan grafik dari

untuk

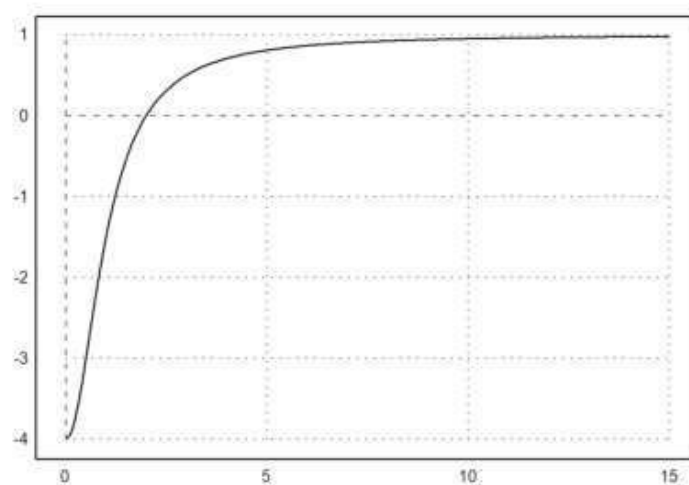
```
>plot2d("2*sin(pi*x/2)+1",0,4):
```



2. Sketsakan grafik fungsi dari

di rentang $[0, 15]$

```
>plot2d("(x^2-4)/(x^2+1)",0,15):
```



3. Gambarkan grafik

di rentang $[-10, 10]$

```
>plot2d("x^3-6*x^2+9*x+1",-10,10):
```

