

Nama: Alya Putri Pertiwi
NIM: 24030130027
Kelas: Pendidikan Matematika C

Membuat Grafik 2D dengan EMT

Buku panduan ini menjelaskan cara menggambar berbagai kurva dan grafik 2D menggunakan perangkat lunak EMT. EMT menyediakan fungsi `plot2d()` untuk menggambar berbagai kurva dan grafik dua dimensi (2D).

Grafik Dasar

Ada fungsi dasar untuk menggambar grafik. Ada koordinat layar, yang selalu berkisar antara 0 hingga 1024 pada setiap sumbu, terlepas dari apakah layar berbentuk persegi atau tidak. Ada juga koordinat grafik, yang dapat diatur dengan fungsi `setplot()`. Pemetaan antara koordinat bergantung pada jendela grafik saat ini. Misalnya, fungsi default `shrinkwindow()` menyisakan ruang untuk label sumbu dan judul grafik.

Dalam contoh ini, kita hanya menggambar beberapa garis acak dengan berbagai warna. Untuk detail tentang fungsi-fungsi ini, pelajari fungsi inti EMT.

```
>clg; // clear screen
>window(0,0,1024,1024); // use all of the window
>setplot(0,1,0,1); // set plot coordinates
>hold on; // start overwrite mode
>n=100; X=random(n,2); Y=random(n,2); // get random points
>colors=rgb(random(n),random(n),random(n)); // get random colors
>loop 1 to n; color(colors[#]); plot(X[#],Y[#]); end; // plot
>hold off; // end overwrite mode
>insimg; // insert to notebook
```



```
>reset;
```

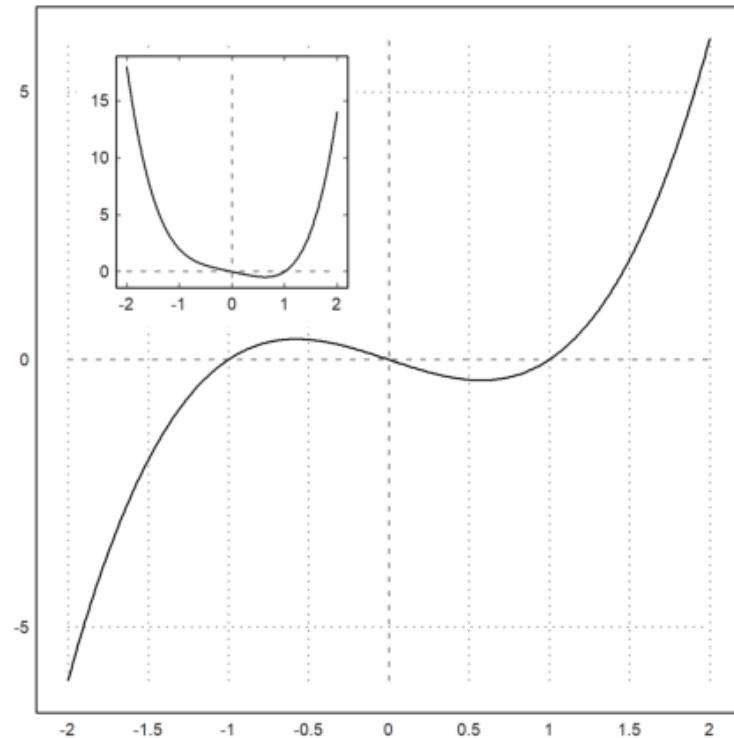
Perlu mempertahankan grafik, karena perintah `plot()` akan menghapus jendela plot.

Untuk menghapus semua yang telah kita lakukan, kita menggunakan `reset()`.

Untuk menampilkan gambar hasil plot di layar notebook, perintah `plot2d()` dapat diakhiri dengan titik dua (:). Cara lain adalah perintah `plot2d()` diakhiri dengan titik koma (;), kemudian menggunakan perintah `insimg()` untuk menampilkan gambar hasil plot.

For another example, we draw a plot as an inset in another plot. This is done by defining a smaller plot window. Note that this window does not provide room for the axis labels outside the plot window. We have to add some margin for this as needed. Note that we save and restore the full window, and hold the current plot while we plot the inset.

```
>plot2d("x^3-x");
>xw=200; yw=100; ww=300; hw=300;
>ow=window();
>window(xw,yw,xw+ww,yw+hw);
>hold on;
>barclear(xw-50,yw-10,ww+60,ww+60);
>plot2d("x^4-x",grid=6):
```



```
>hold off;
>window(ow);
```

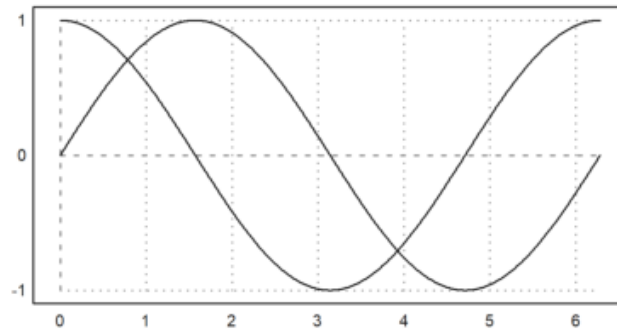
Grafik dengan beberapa gambar dapat dibuat dengan cara yang sama. Fungsi utility figure() digunakan untuk ini.

Aspek Grafik

Grafik default menggunakan jendela grafik berbentuk persegi. Anda dapat mengubah ini dengan fungsi aspect(). Jangan lupa untuk mengatur ulang aspeknya nanti. Anda juga dapat mengubah pengaturan default ini di menu dengan "Set Aspect" ke rasio aspek tertentu atau ke ukuran jendela grafik saat ini.

Namun, Anda juga dapat mengubahnya untuk satu plot. Untuk ini, ukuran area plot saat ini diubah, dan jendela disesuaikan agar label memiliki ruang yang cukup.

```
>aspect(2); // rasio panjang dan lebar 2:1
>plot2d(["sin(x)", "cos(x)"], 0, 2pi):
```



```
>aspect ( ) ;
>reset ;
```

Fungsi reset() mengembalikan pengaturan default plot, termasuk rasio aspek.

Plot 2D di Euler

EMT Math Toolbox memiliki plot 2D, baik untuk data maupun fungsi. EMT menggunakan fungsi plot2d. Fungsi ini dapat menggambar fungsi dan data.

Dimungkinkan untuk menggambar plot di Maxima menggunakan Gnuplot atau di Python menggunakan Math Plot Lib.

Euler dapat memplot grafik 2D dari

- ekspresi
- fungsi, variabel, atau kurva yang diparameterkan,
- vektor nilai x-y,
- awan titik di bidang,
- kurva implisit dengan tingkat atau wilayah tingkat.
- Fungsi kompleks

Gaya plot mencakup berbagai gaya untuk garis dan titik, grafik batang, dan grafik bergradasi.

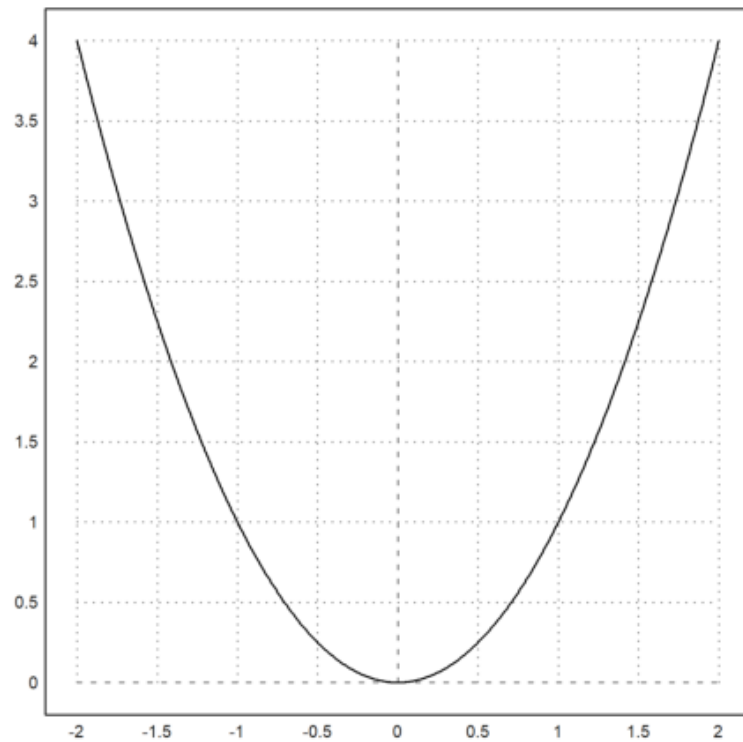
Membuat Grafik Ekspresi atau Variabel

Sebuah ekspresi tunggal dalam "x" (misalnya " $4x^2$ ") atau nama fungsi (misalnya "f") akan menghasilkan grafik fungsi tersebut.

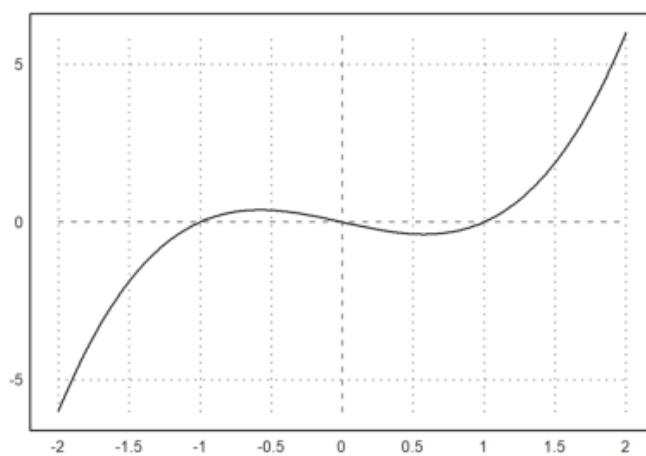
Berikut adalah contoh paling dasar, yang menggunakan rentang default dan mengatur rentang y yang tepat untuk menyesuaikan grafik fungsi.

Catatan: Jika Anda mengakhiri baris perintah dengan tanda titik dua ":", grafik akan disisipkan ke jendela teks. Jika tidak, tekan TAB untuk melihat grafik jika jendela grafik tertutup.

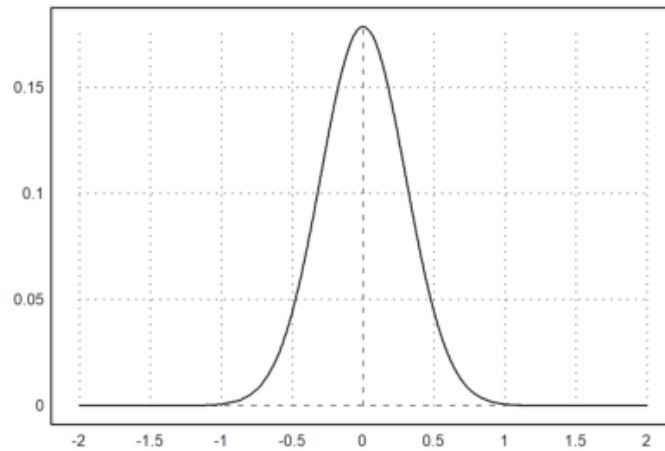
```
>plot2d("x^2") :
```



```
>aspect(1.5); plot2d("x^3-x"):
```



```
>a:=5.6; plot2d("exp(-a*x^2)/a"); insimg(30); // menampilkan gambar hasil plot setinggi 25
```

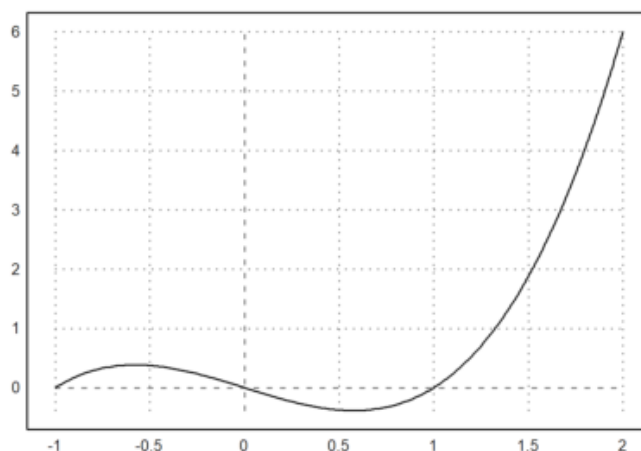


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

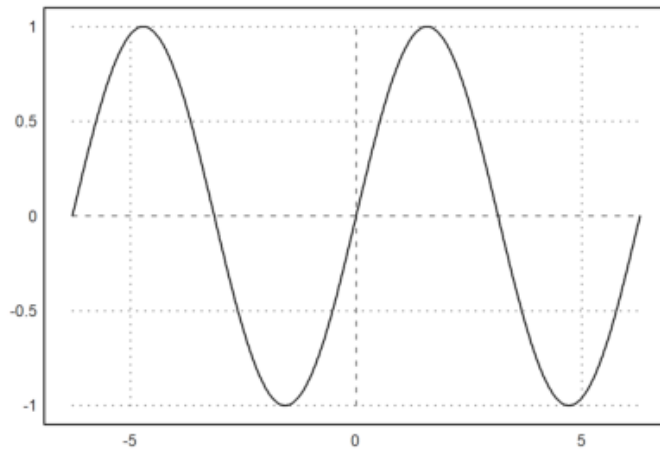
Rentang plot diatur dengan parameter yang telah ditetapkan sebagai berikut

- a,b: rentang x (default -2,2)
- c,d: rentang y (default: disesuaikan dengan nilai)
- r: jari-jari di sekitar pusat plot
- cx,cy: koordinat pusat plot (default 0,0)

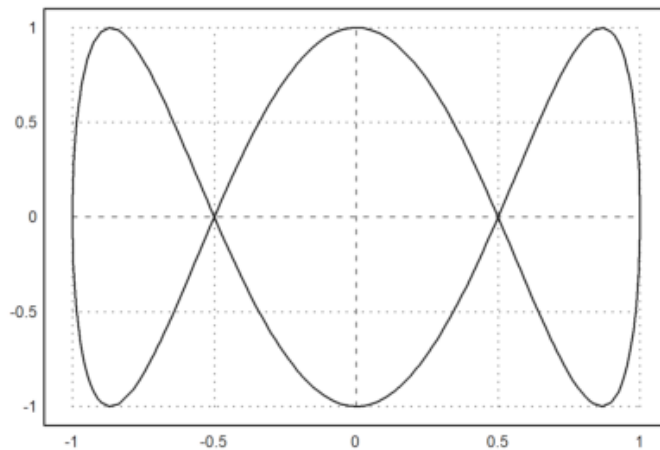
```
>plot2d("x^3-x",-1,2):
```



```
>plot2d("sin(x)",-2*pi,2*pi): // plot sin(x) pada interval [-2pi, 2pi]
```



```
>plot2d("cos(x)", "sin(3*x)", xmin=0, xmax=2pi):
```



Alternatif untuk simbol titik dua adalah perintah `insimg(lines)`, yang menyisipkan grafik yang menempati jumlah baris teks yang ditentukan.

Dalam opsi, grafik dapat diatur untuk ditampilkan

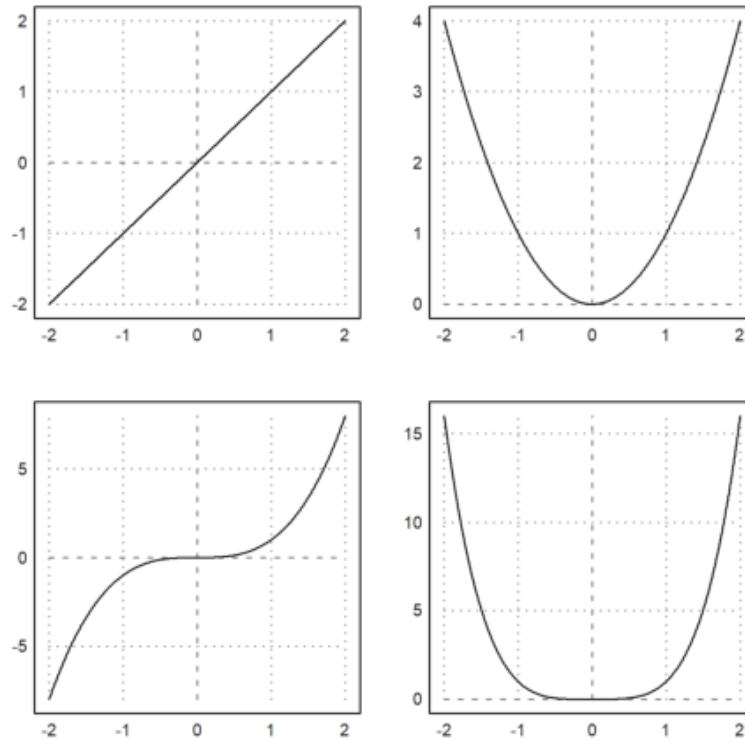
- di jendela terpisah yang dapat diubah ukurannya,
- di jendela notebook.

Gaya lain dapat dicapai dengan perintah grafik spesifik.

Dalam hal apa pun, tekan tombol tabulator untuk melihat grafik jika tersembunyi.

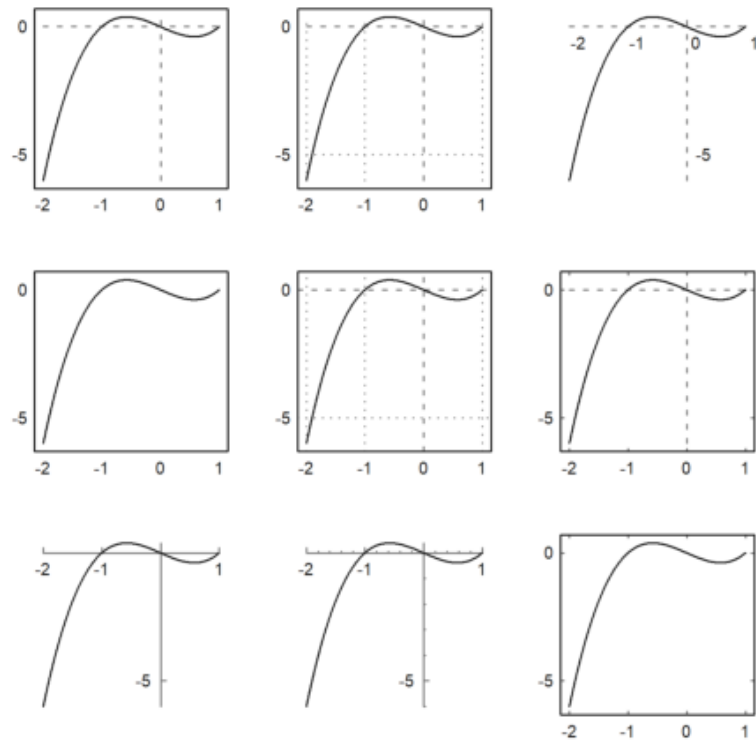
Untuk membagi jendela menjadi beberapa plot, gunakan perintah `figure()`. Dalam contoh ini, kita memplot x^1 hingga x^4 ke dalam 4 bagian jendela. `figure(0)` mengembalikan jendela ke pengaturan default.

```
>reset;
>figure(2,2); ...
>for n=1 to 4; figure(n); plot2d("x^"+n); end; ...
>figure(0):
```



Dalam fungsi `plot2d()`, terdapat gaya alternatif yang tersedia dengan opsi `grid=x`. Untuk gambaran umum, kami menampilkan berbagai gaya grid dalam satu gambar (lihat perintah `figure()` di bawah). Gaya `grid=0` tidak termasuk. Gaya ini tidak menampilkan grid dan bingkai.

```
>figure(3,3); ...
>for k=1:9; figure(k); plot2d("x^3-x",-2,1,grid=k); end; ...
>figure(0) :
```

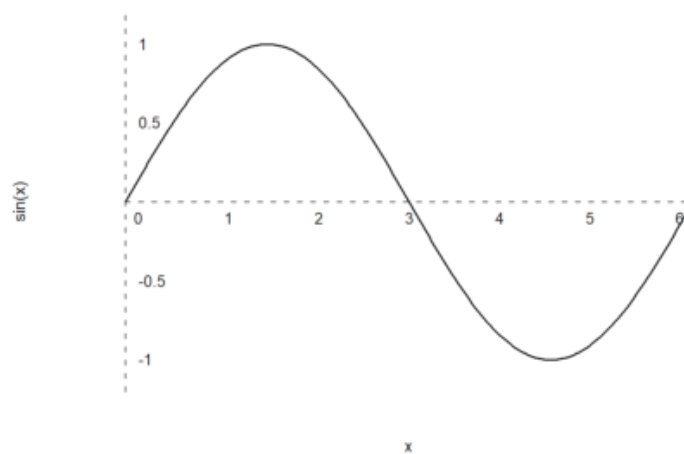



Jika argumen yang diberikan ke `plot2d()` berupa ekspresi diikuti oleh empat angka, angka-angka tersebut merupakan rentang x dan y untuk grafik.

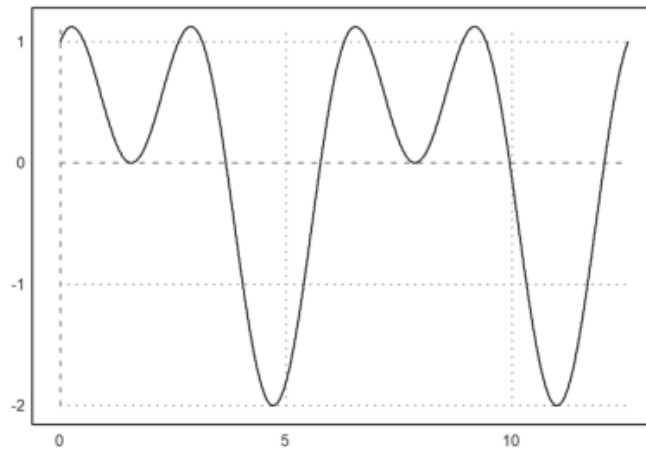
Sebagai alternatif, a , b , c , d dapat ditentukan sebagai parameter yang ditetapkan, misalnya $a=...$ dan seterusnya.

Dalam contoh berikut, kita mengubah gaya grid, menambahkan label, dan menggunakan label vertikal untuk sumbu y .

```
>aspect(1.5); plot2d("sin(x)",0,2pi,-1.2,1.2,grid=3,xl="x",yl="sin(x)");
```



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

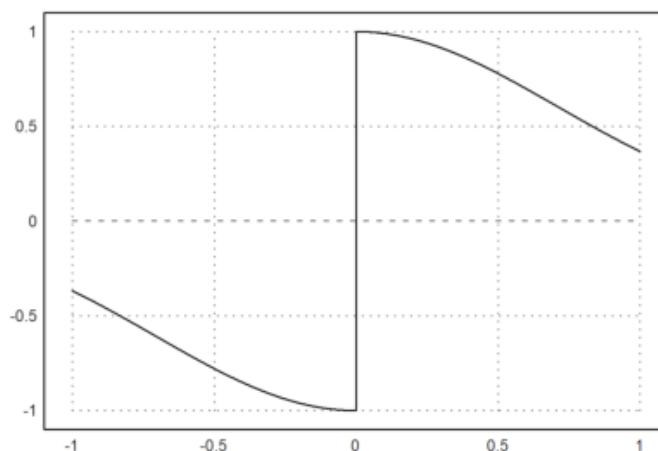


Gambar yang dihasilkan dengan menyisipkan plot ke dalam jendela teks disimpan di direktori yang sama dengan notebook, secara default di subdirektori bernama "images". Gambar-gambar ini juga digunakan oleh ekspor HTML.

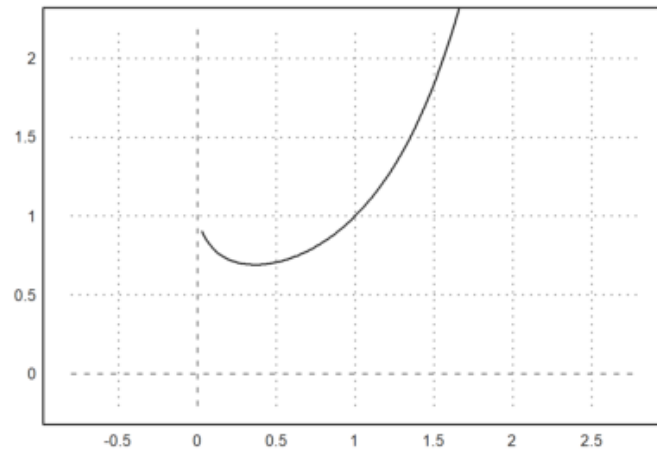
Anda dapat menandai gambar apa pun dan menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, Anda juga dapat mengeksport grafik saat ini menggunakan fungsi di menu File.

Fungsi atau ekspresi dalam plot2d dievaluasi secara adaptif. Untuk kecepatan yang lebih tinggi, nonaktifkan plot adaptif dengan <adaptive dan tentukan jumlah subinterval dengan n=... Hal ini hanya diperlukan dalam kasus yang jarang terjadi.

```
>plot2d("sign(x)*exp(-x^2)",-1,1,<adaptive,n=10000):
```

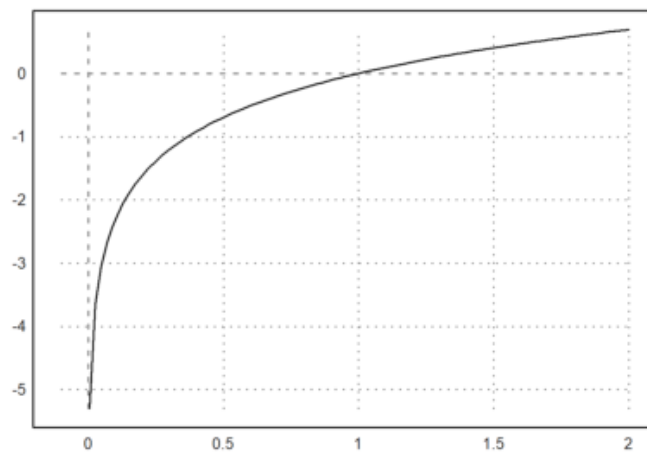


```
>plot2d("x^x",r=1.2,cx=1,cy=1):
```



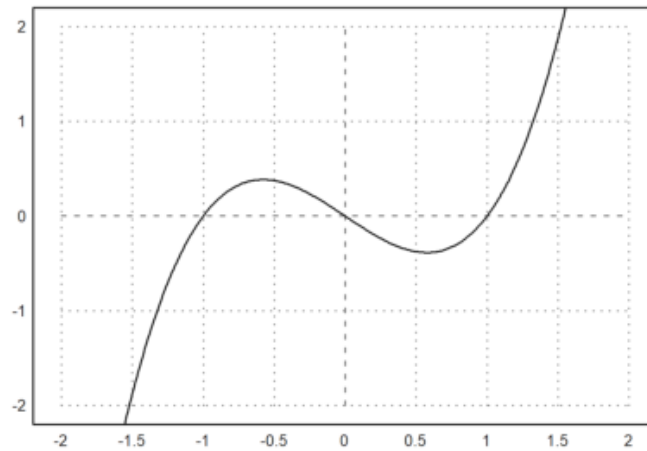
Perhatikan bahwa x^x tidak terdefinisi untuk $x = 0$. Fungsi `plot2d` mendeteksi kesalahan ini dan mulai menggambar grafik begitu fungsi tersebut terdefinisi. Hal ini berlaku untuk semua fungsi yang mengembalikan NAN di luar rentang definisinya.

```
>plot2d("log(x)",-0.1,2):
```

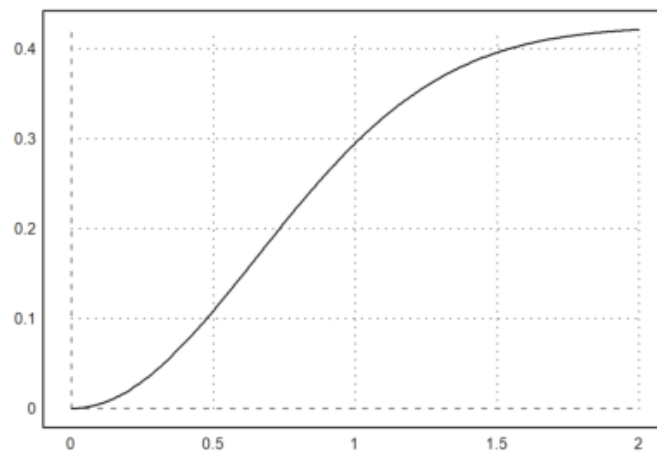


Parameter `square=true` (atau `>square`) secara otomatis memilih rentang y sehingga hasilnya adalah jendela plot berbentuk persegi. Perhatikan bahwa secara default, Euler menggunakan ruang berbentuk persegi di dalam jendela plot.

```
>plot2d("x^3-x",>square):
```

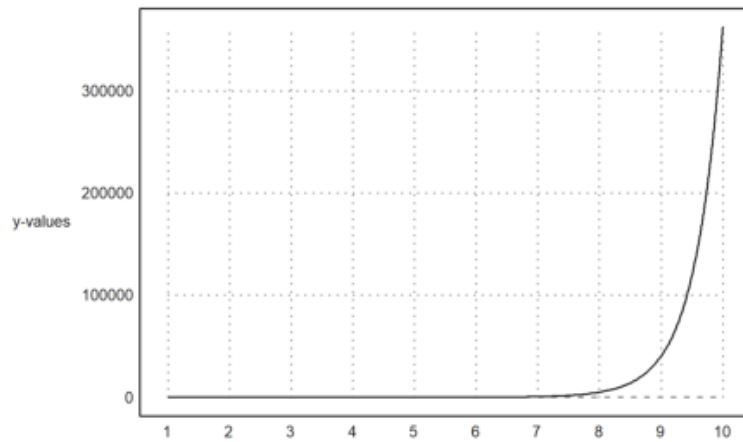


```
>plot2d(''integrate("sin(x)*exp(-x^2)","0,x)','',0,2): // plot integral
```



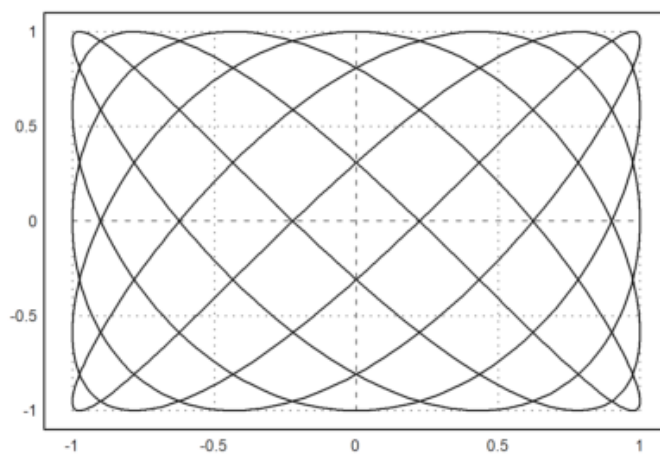
Jika Anda membutuhkan ruang lebih untuk label sumbu y, panggil fungsi `shrinkwindow()` dengan parameter yang lebih kecil, atau atur nilai positif untuk “smaller” dalam fungsi `plot2d()`.

```
>plot2d("gamma(x)",1,10,yl="y-values",smaller=6,<vertical):
```

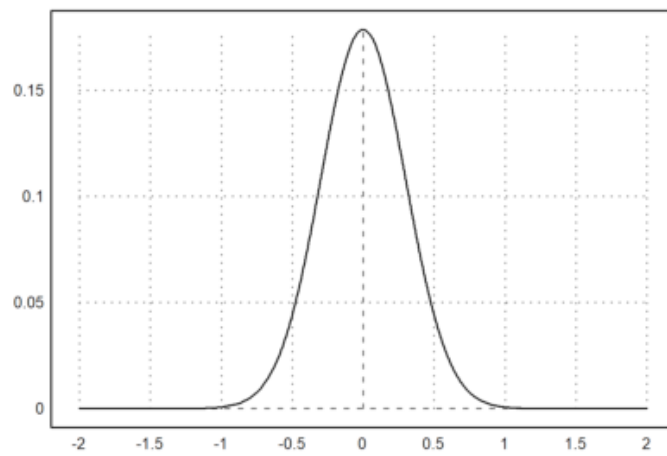


Ekspresi simbolik juga dapat digunakan, karena disimpan sebagai ekspresi string sederhana.

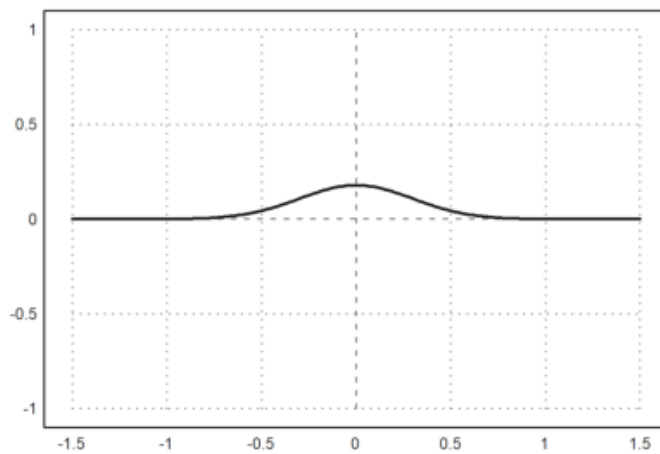
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



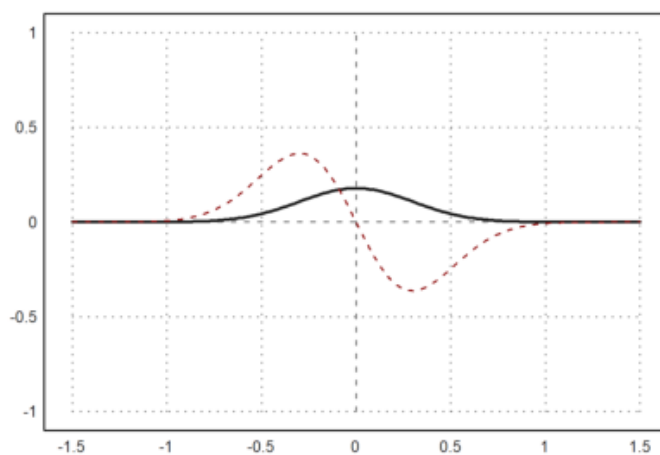
```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression
>plot2d(expr,-2,2): // plot from -2 to 2
```



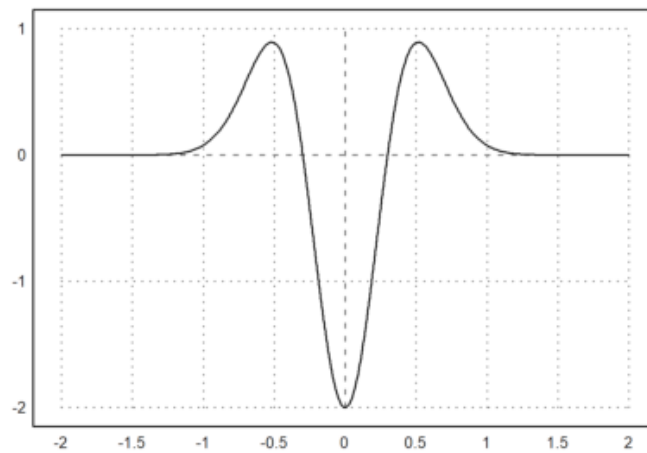
```
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)
```



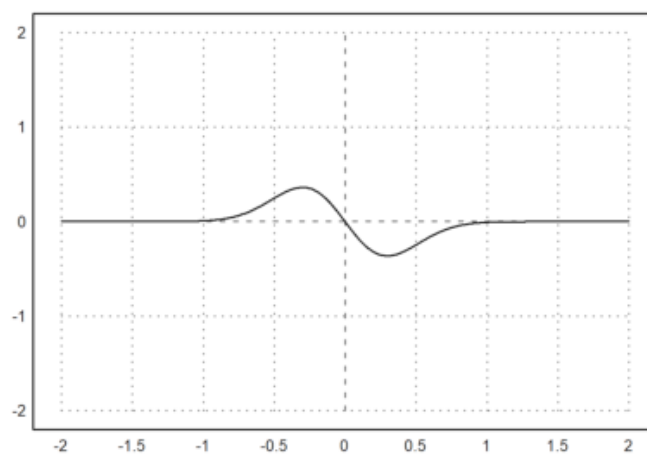
```
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot
```



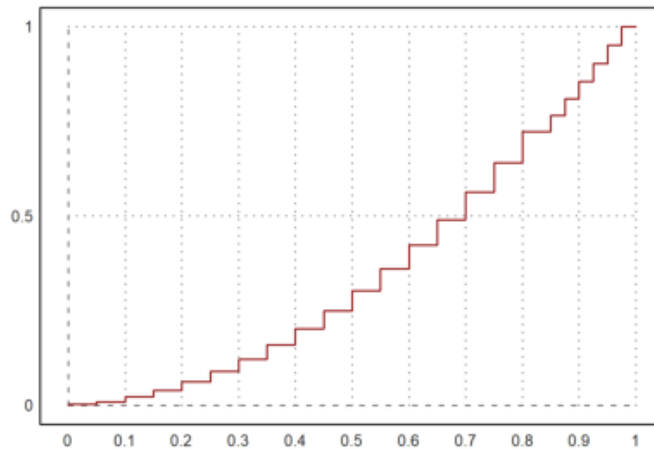
```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle
```



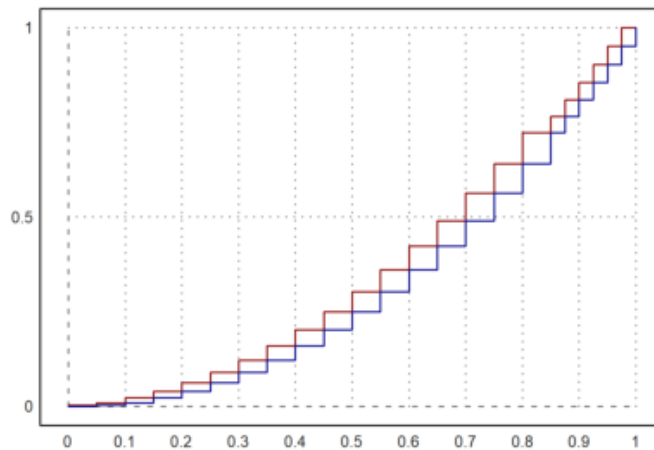
```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square
```



```
>plot2d("x^2",0,1,steps=1,color=red,n=10):
```



```
>plot2d("x^2",>add,steps=2,color=blue,n=10):
```

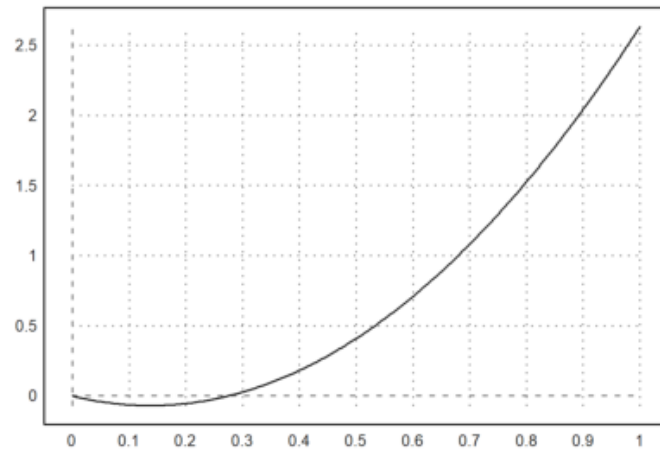


Fungsi dengan Satu Parameter

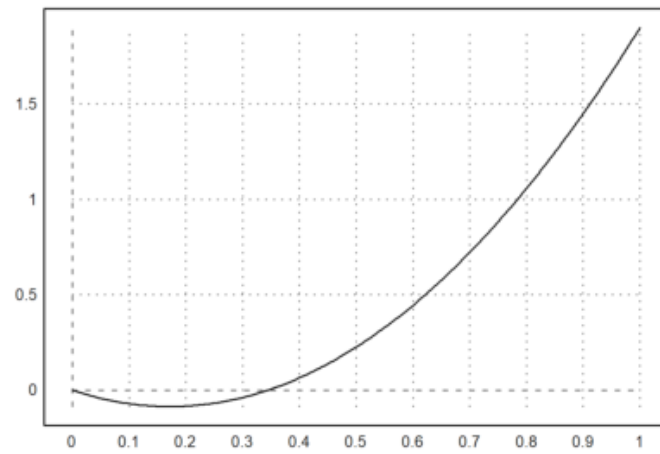
Fungsi plot terpenting untuk grafik dua dimensi adalah `plot2d()`. Fungsi ini diimplementasikan dalam bahasa Euler pada berkas “plot.e”, yang dimuat pada awal program.

Berikut adalah beberapa contoh penggunaan fungsi. Seperti biasa dalam EMT, fungsi yang bekerja untuk fungsi atau ekspresi lain, Anda dapat meneruskan parameter tambahan (selain x) yang bukan variabel global ke fungsi menggunakan parameter titik koma atau kumpulan panggilan.

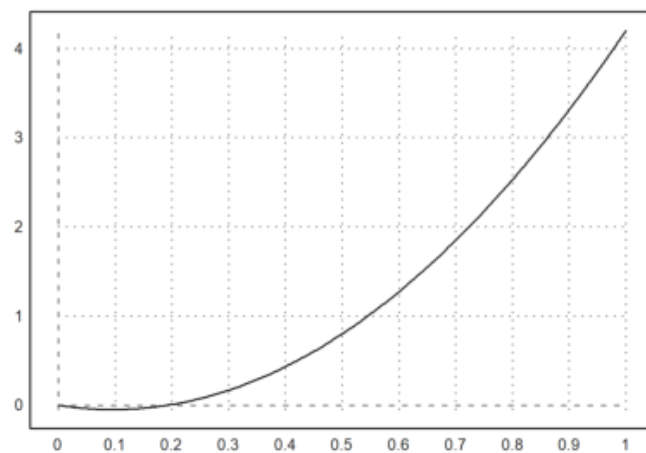
```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
>a=0.3; plot2d("f",0,1;a): // plot with a=0.3
```

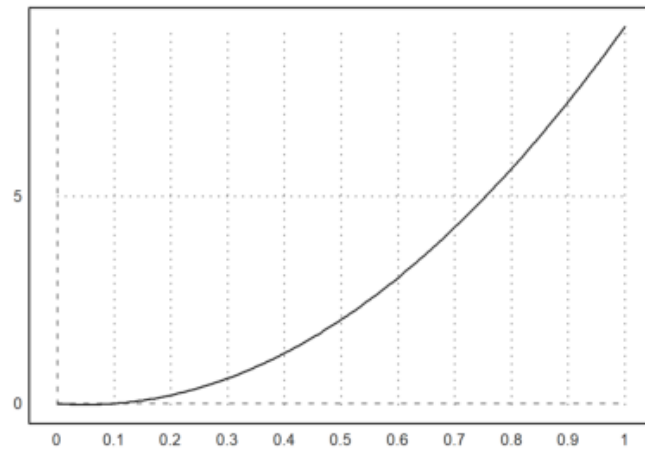
```
>plot2d("f",0,1;0.4): // plot with a=0.4
```



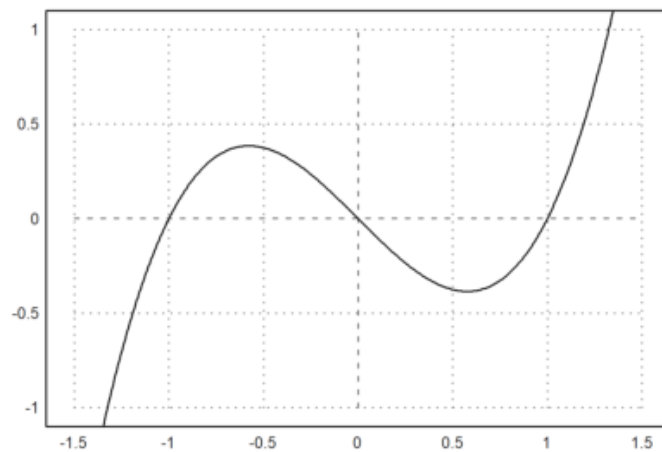
```
>plot2d({"f",0.2},0,1): // plot with a=0.2
```



```
>plot2d({{"f(x,b)",b=0.1}},0,1): // plot with 0.1
```



```
>function f(x) := x^3-x; ...
>plot2d("f",r=1):
```



Berikut adalah ringkasan fungsi yang diterima

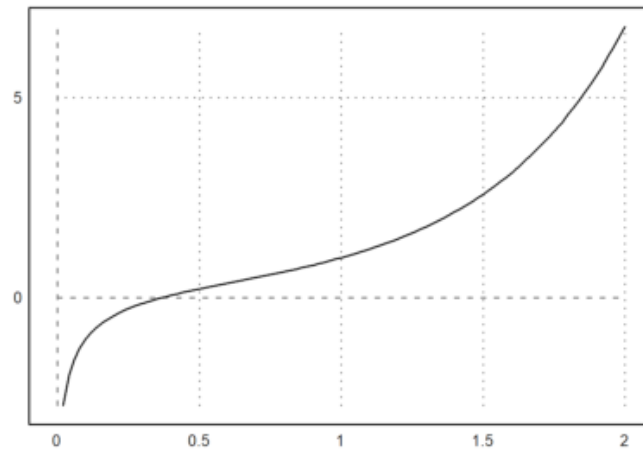
- ekspresi atau ekspresi simbolik dalam x
- fungsi atau fungsi simbolik dengan nama "f"
- fungsi simbolik hanya dengan nama f

Fungsi plot2d() juga menerima fungsi simbolik. Untuk fungsi simbolik, nama saja sudah cukup.

```
>function f(x) &= diff(x^x,x)
```

$$x^x (\log(x) + 1)$$

```
>plot2d(f,0,2):
```

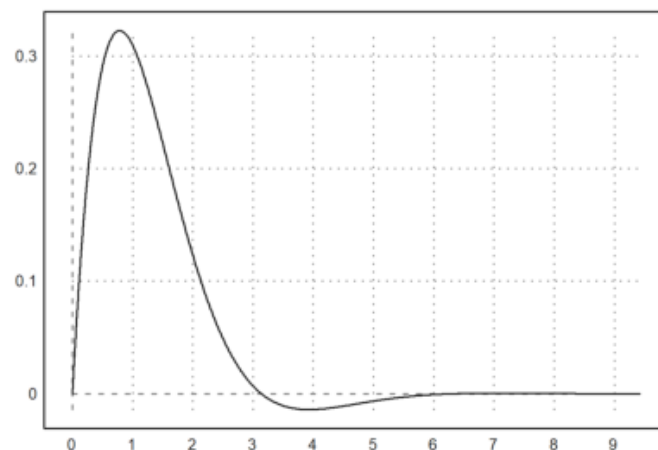


Tentu saja, untuk ekspresi atau ekspresi simbolik, nama variabel saja sudah cukup untuk menggambarannya.

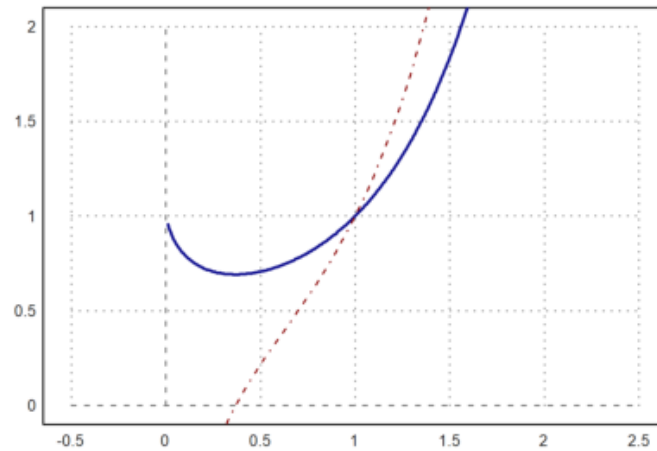
```
>expr &= sin(x)*exp(-x)
```

$$E^{-x} \sin(x)$$

```
>plot2d(expr,0,3pi):
```



```
>function f(x) &= x^x;
>plot2d(f,r=1,cx=1,cy=1,color=blue,thickness=2);
>plot2d(&diff(f(x),x),>add,color=red,style="-.-"):
```



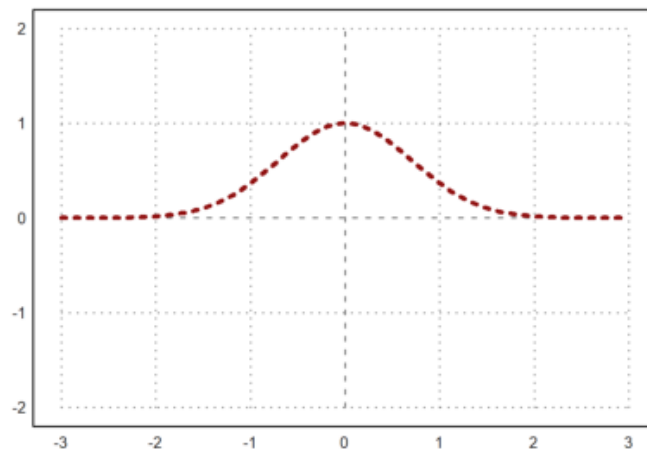
Untuk gaya garis, terdapat berbagai opsi.

- style="...". Pilih dari "-", "_", "-.", ".", "-.-", "-.-".
- warna: Lihat di bawah untuk pilihan warna.
- ketebalan: Default adalah 1.

Warna dapat dipilih sebagai salah satu warna default, atau sebagai warna RGB.

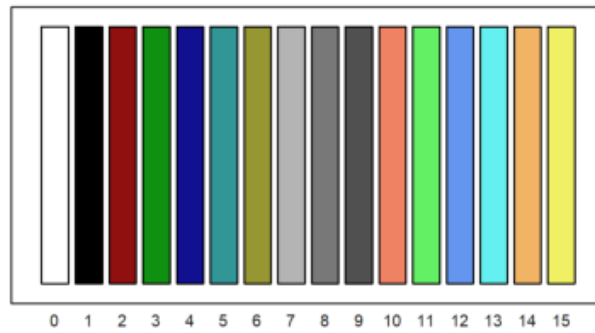
- 0..15: indeks warna default.
- konstanta warna: putih, hitam, merah, hijau, biru, cyan, zaitun, abu-abu terang, abu-abu, abu-abu gelap, oranye, hijau terang, turquoise, biru terang, oranye terang, kuning
- rgb(merah, hijau, biru): parameter adalah bilangan real dalam [0,1].

```
>plot2d("exp(-x^2)",r=2,color=red,thickness=3,style="--") :
```



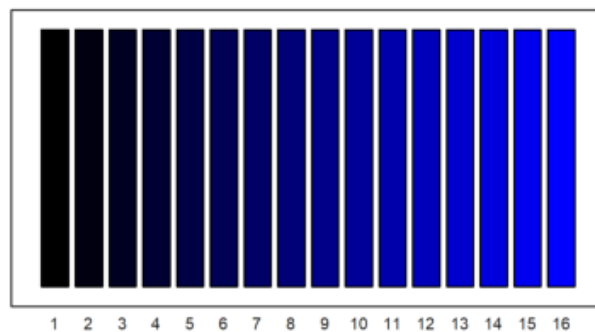
Berikut ini adalah tampilan warna-warna bawaan EMT.

```
>aspect(2); columnsplot(ones(1,16),lab=0:15,grid=0,color=0:15) :
```



Tetapi Anda dapat menggunakan warna apa pun.

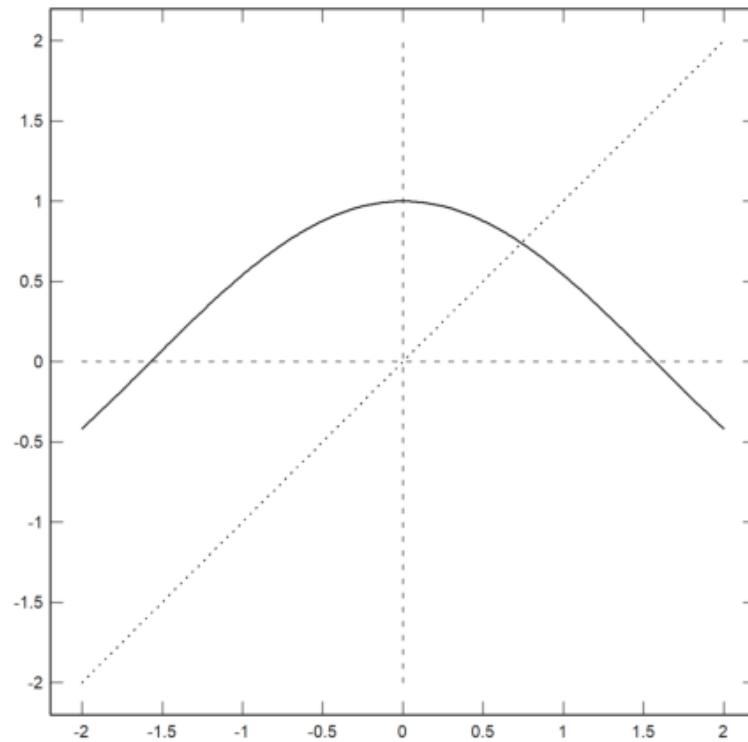
```
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```



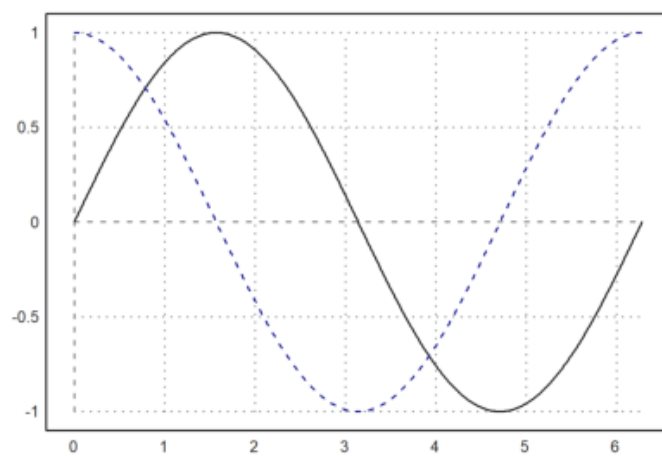
Menggambar Beberapa Kurva pada bidang koordinat yang sama

Menampilkan lebih dari satu fungsi (fungsi ganda) dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu metodenya adalah menggunakan `>add` untuk beberapa panggilan ke `plot2d`, kecuali panggilan pertama. Fitur ini telah kami gunakan dalam contoh-contoh di atas.

```
>aspect(); plot2d("cos(x)",r=2,grid=6); plot2d("x",style=".",>add):
```

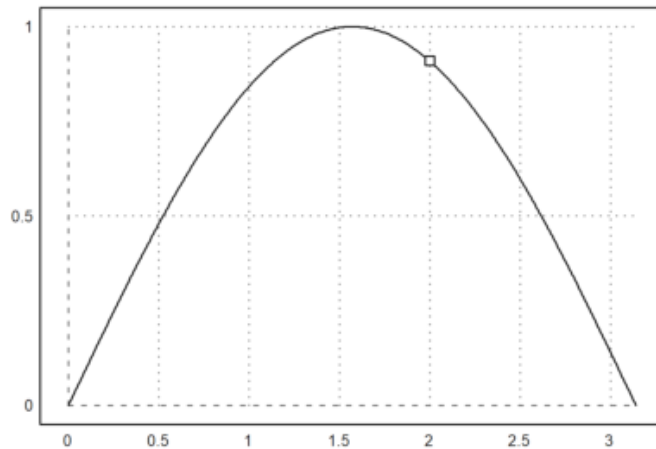


```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



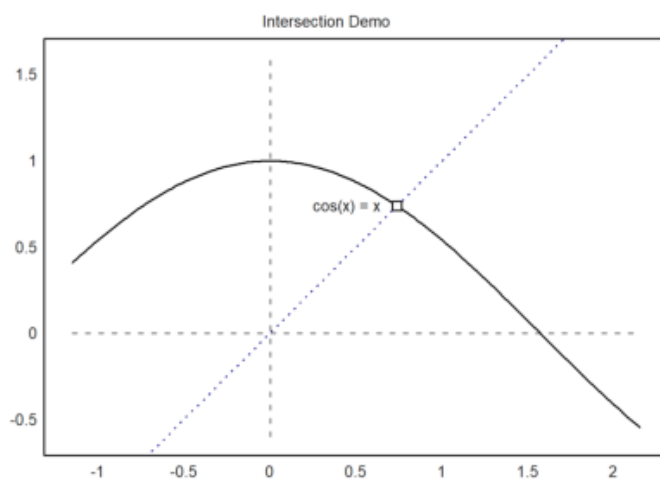
Salah satu kegunaan `>add` adalah untuk menambahkan titik pada kurva.

```
>plot2d("sin(x)",0,pi); plot2d(2,sin(2),>points,>add):
```



Kami menambahkan titik potong dengan label (pada posisi "cl" untuk tengah kiri), dan menyisipkan hasilnya ke dalam notebook. Kami juga menambahkan judul ke grafik.

```
>plot2d(["cos(x)", "x"], r=1.1, cx=0.5, cy=0.5, ...
> color=[black, blue], style=["-", "."], ...
> grid=1);
>x0=solve("cos(x)-x", 1); ...
> plot2d(x0, x0, >points, >add, title="Intersection Demo"); ...
> label("cos(x) = x", x0, x0, pos="cl", offset=20):
```



Dalam demo berikut, kami menggambar fungsi $\text{sinc}(x) = \sin(x)/x$ dan ekspansi Taylor ke-8 dan ke-16-nya. Ekspansi ini dihitung menggunakan Maxima melalui ekspresi simbolik.

Grafik ini dibuat dengan perintah multi-baris berikut yang mencakup tiga panggilan ke `plot2d()`. Panggilan kedua dan ketiga memiliki bendera `>add` yang diaktifkan, sehingga grafik menggunakan rentang sebelumnya. Kami menambahkan kotak label yang menjelaskan fungsi-fungsi tersebut.

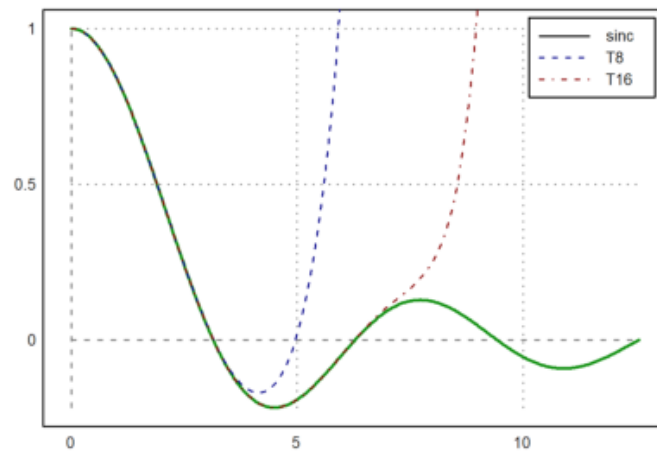
```
>$taylor(sin(x)/x, x, 0, 4)
```

$$\frac{x^4}{120} - \frac{x^2}{6} + 1$$

```

>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
> plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
> plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-.-"); ...
> labelbox(["sinc","T8","T16"],styles=["-","--","-.-"], ...
> colors=[black,blue,red]):

```



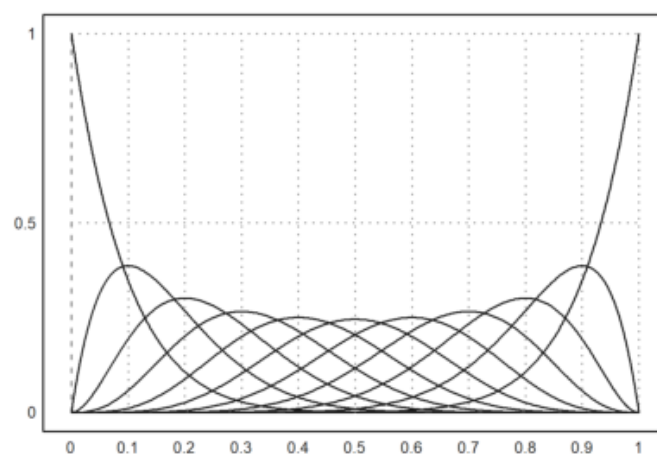
Dalam contoh berikut, kita menghitung Polinomial Bernstein.

$$B_i(x) = \binom{n}{i} x^i (1-x)^{n-i}$$

```

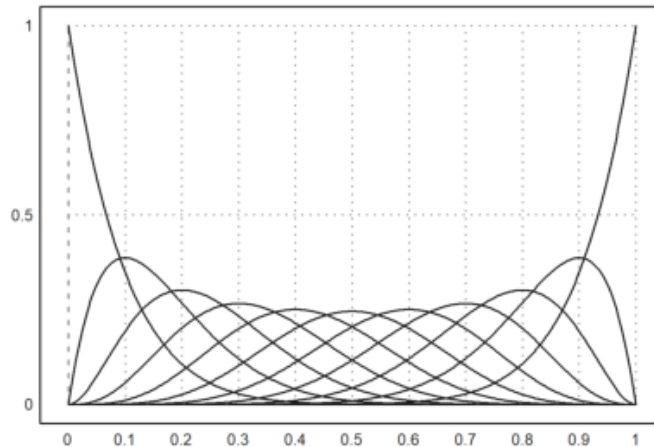
>plot2d("(1-x)^10",0,1); // plot first function
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;
>insimg;

```



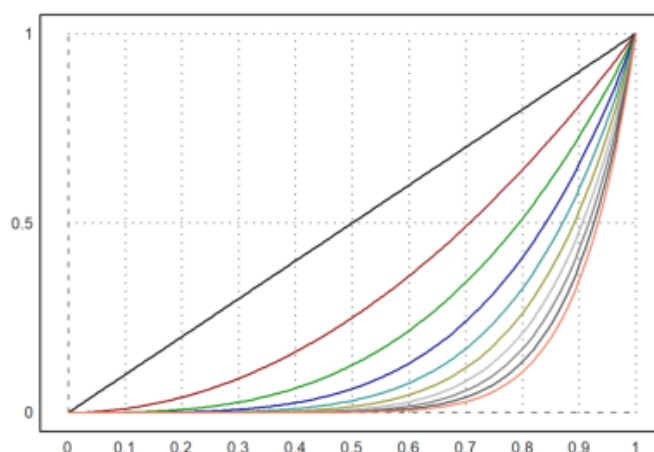
Metode kedua adalah menggunakan pasangan matriks nilai x dan matriks nilai y dengan ukuran yang sama. Kami menghasilkan matriks nilai dengan satu polinomial Bernstein di setiap baris. Untuk ini, kami cukup menggunakan vektor kolom i . Lihat bagian pengenalan tentang bahasa matriks untuk detail lebih lanjut.

```
>x=linspace(0,1,500);
>n=10; k=(0:n)'; // n is row vector, k is column vector
>y=bin(n,k)*x^k*(1-x)^(n-k); // y is a matrix then
>plot2d(x,y):
```



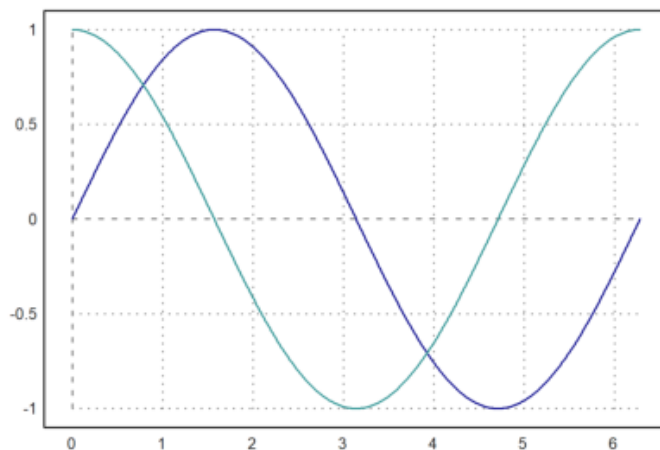
Perhatikan bahwa parameter warna dapat berupa vektor. Kemudian, setiap warna digunakan untuk setiap baris matriks.

```
>x=linspace(0,1,200); y=x^(1:10)'; plot2d(x,y,color=1:10):
```

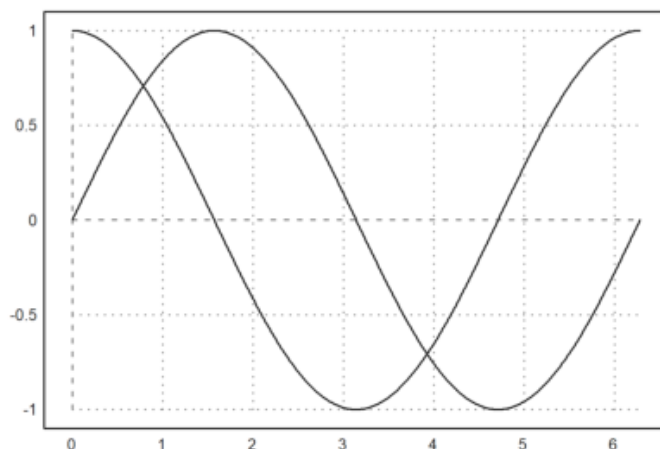


Metode lain adalah menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan array warna, array gaya, dan array ketebalan dengan panjang yang sama.

```
>plot2d(["sin(x)", "cos(x)"], 0, 2pi, color=4:5):
```



```
>plot2d(["sin(x)", "cos(x)"], 0, 2pi): // plot vector of expressions
```



Kita dapat memperoleh vektor tersebut dari Maxima menggunakan makelist() dan mxm2str().

```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // make list
```

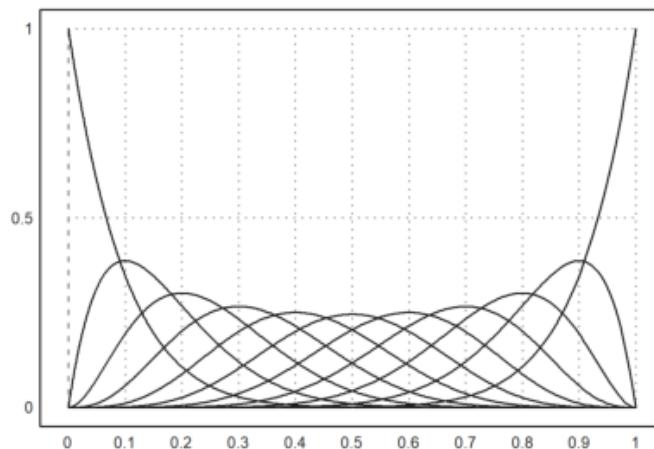
```

      10      9      8 2      7 3
[(1 - x) , 10 (1 - x) x, 45 (1 - x) x , 120 (1 - x) x ,
 6 4      5 5      4 6      3 7
210 (1 - x) x , 252 (1 - x) x , 210 (1 - x) x , 120 (1 - x) x ,
 2 8      9 10
45 (1 - x) x , 10 (1 - x) x , x ]
```

```
>mxm2str(v) // get a vector of strings from the symbolic vector
```

```
(1-x)^10  
10*(1-x)^9*x  
45*(1-x)^8*x^2  
120*(1-x)^7*x^3  
210*(1-x)^6*x^4  
252*(1-x)^5*x^5  
210*(1-x)^4*x^6  
120*(1-x)^3*x^7  
45*(1-x)^2*x^8  
10*(1-x)*x^9  
x^10
```

```
>plot2d(mxm2str(v),0,1): // plot functions
```

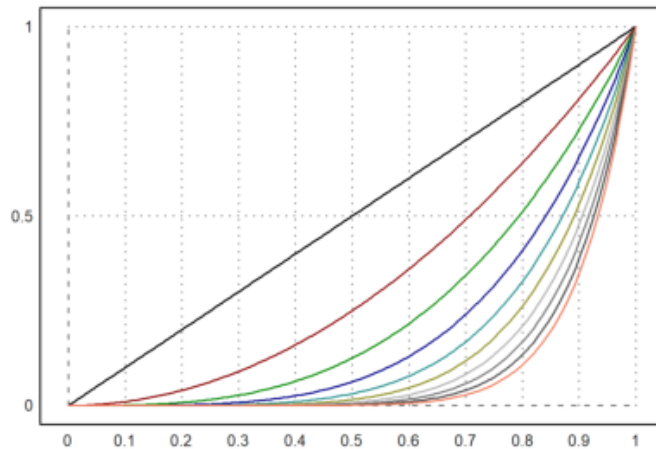


Alternatif lain adalah menggunakan bahasa matriks Euler.

Jika suatu ekspresi menghasilkan matriks fungsi, dengan satu fungsi di setiap baris, semua fungsi tersebut akan digambar dalam satu grafik.

Untuk ini, gunakan vektor parameter dalam bentuk vektor kolom. Jika ditambahkan array warna, array tersebut akan digunakan untuk setiap baris grafik.

```
>n=(1:10)'; plot2d("x^n",0,1,color=1:10):
```

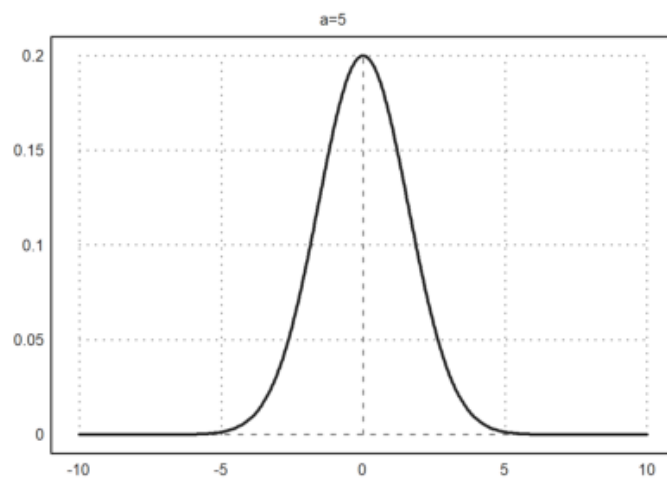


Ekspresi dan fungsi satu baris dapat mengakses variabel global.

Jika Anda tidak dapat menggunakan variabel global, Anda perlu menggunakan fungsi dengan parameter tambahan, dan meneruskan parameter ini sebagai parameter titik koma.

Perhatikan, untuk menempatkan semua parameter yang ditugaskan di akhir perintah plot2d. Dalam contoh ini, kami meneruskan $a=5$ ke fungsi f , yang kami plot dari -10 hingga 10.

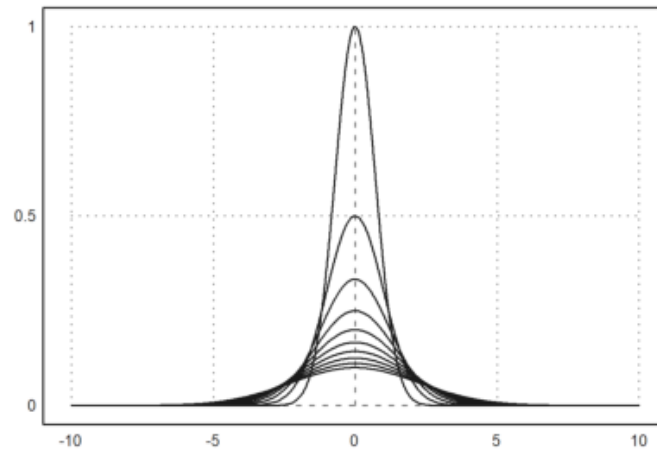
```
>function f(x,a) := 1/a*exp(-x^2/a); ...
>plot2d("f",-10,10;5,thickness=2,title="a=5"):
```



Sebagai alternatif, gunakan koleksi yang berisi nama fungsi dan semua parameter tambahan. Daftar khusus ini disebut koleksi panggilan, dan ini adalah cara yang disarankan untuk meneruskan argumen ke fungsi yang sendiri diteruskan sebagai argumen ke fungsi lain.

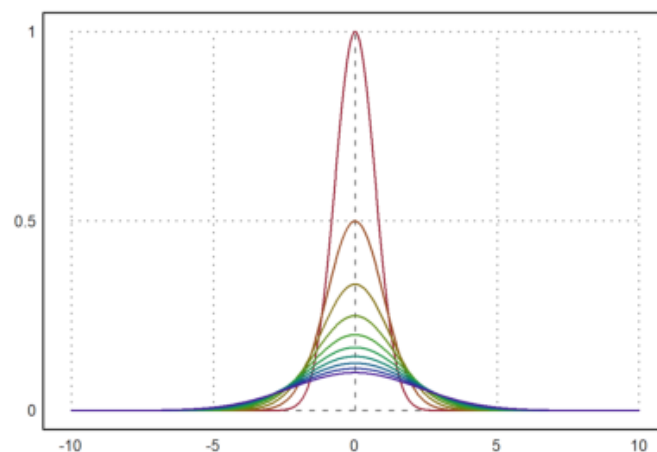
Dalam contoh berikut, kita menggunakan loop untuk menggambar beberapa fungsi (lihat tutorial tentang pemrograman loop).

```
>plot2d({{"f",1}},-10,10); ...
>for a=2:10; plot2d({{"f",a}},>add); end:
```



Kita dapat mencapai hasil yang sama dengan cara berikut menggunakan bahasa matriks EMT. Setiap baris matriks $f(x,a)$ mewakili satu fungsi. Selain itu, kita dapat menetapkan warna untuk setiap baris matriks. Klik ganda pada fungsi `getspectral()` untuk penjelasan.

```
>x=-10:0.01:10; a=(1:10)'; plot2d(x,f(x,a),color=getspectral(a/10)):
```



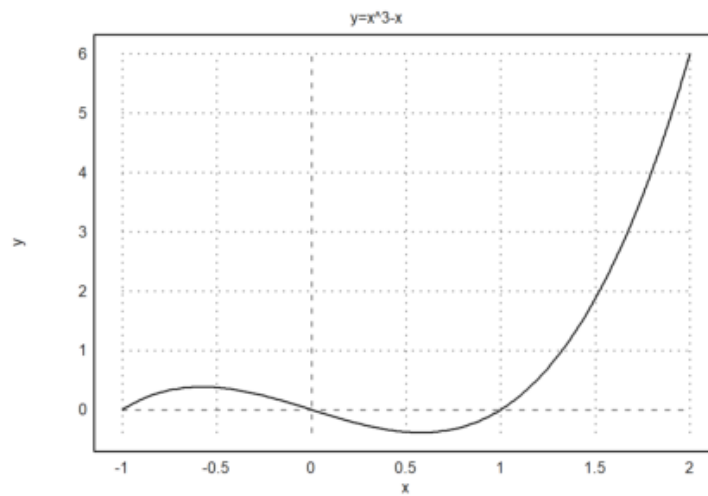
Label Teks

Dekorasi sederhana dapat berupa

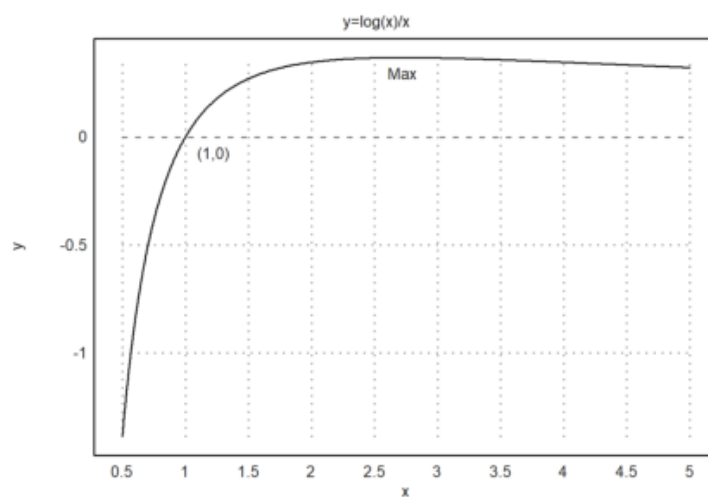
- judul dengan `title="..."`
- label x dan y dengan `xl="...", yl="..."`
- label teks lain dengan `label("...",x,y)`

Perintah label akan menggambar ke dalam plot saat ini pada koordinat plot (x,y). Perintah ini dapat menerima argumen posisional.

```
>plot2d("x^3-x",-1,2,title="y=x^3-x",yl="y",xl="x"):
```

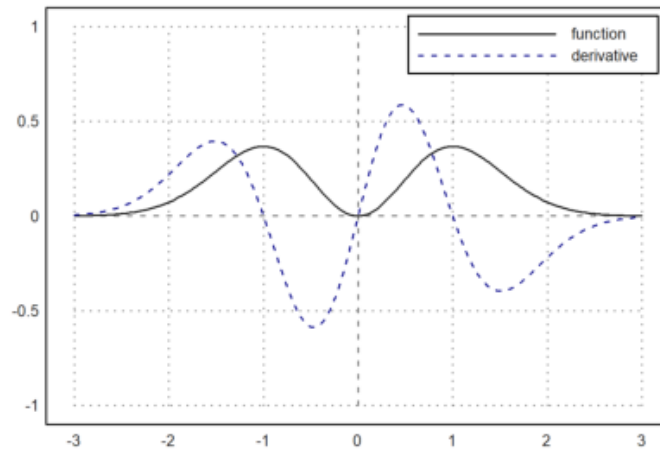


```
>expr := "log(x)/x"; ...
> plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
> label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```



Ada juga fungsi `labelbox()`, yang dapat menampilkan fungsi-fungsi dan teks. Fungsi ini menerima vektor string dan warna, dengan satu item untuk setiap fungsi.

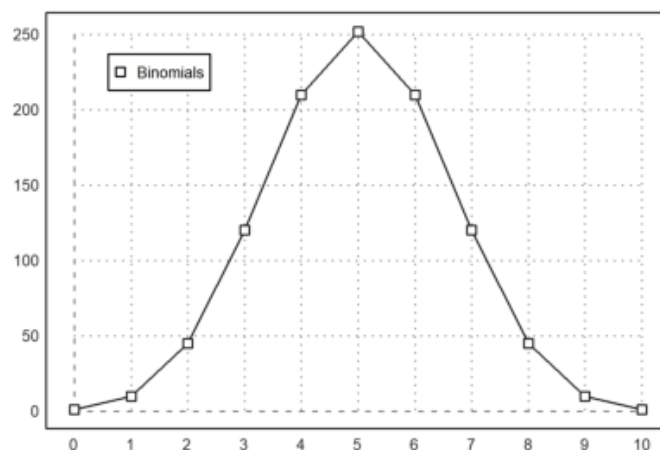
```
>function f(x) &= x^2*exp(-x^2); ...
>plot2d(&f(x),a=-3,b=3,c=-1,d=1); ...
>plot2d(&diff(f(x),x),>add,color=blue,style="--"); ...
>labelbox(["function","derivative"],styles=["-", "--"], ...
> colors=[black,blue],w=0.4):
```



Kotak secara default diposisikan di pojok kanan atas, tetapi `>left` akan memposisikannya di pojok kiri atas. Anda dapat memindahkannya ke mana saja sesuai keinginan. Posisi jangkar adalah pojok kanan atas kotak, dan angka-angka tersebut merupakan pecahan dari ukuran jendela grafik. Lebar kotak diatur secara otomatis. Untuk grafik titik, kotak label juga berfungsi. Tambahkan parameter `>points`, atau vektor bendera, satu untuk setiap label.

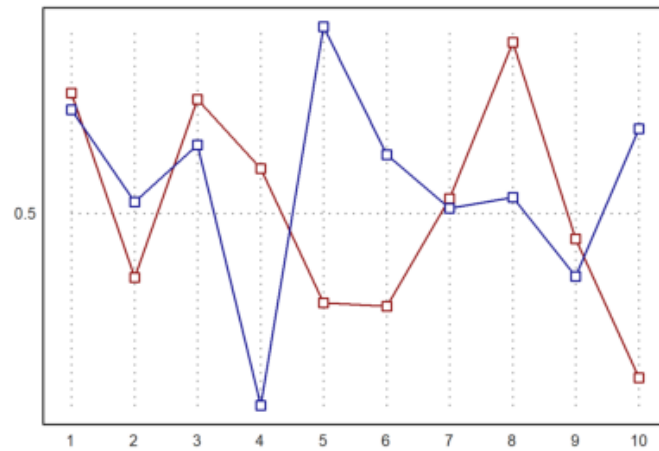
Dalam contoh berikut, hanya ada satu fungsi. Jadi kita dapat menggunakan string alih-alih vektor string. Kita mengatur warna teks menjadi hitam untuk contoh ini.

```
>n=10; plot2d(0:n,bin(n,0:n),>addpoints); ...
>labelbox("Binomials",styles="[]",>points,x=0.1,y=0.1, ...
>tcolor=black,>left):
```



Gaya plot ini juga tersedia dalam fungsi `statplot()`. Sama seperti dalam `plot2d()`, warna dapat diatur untuk setiap baris plot. Terdapat lebih banyak plot khusus untuk tujuan statistik (lihat tutorial tentang statistik).

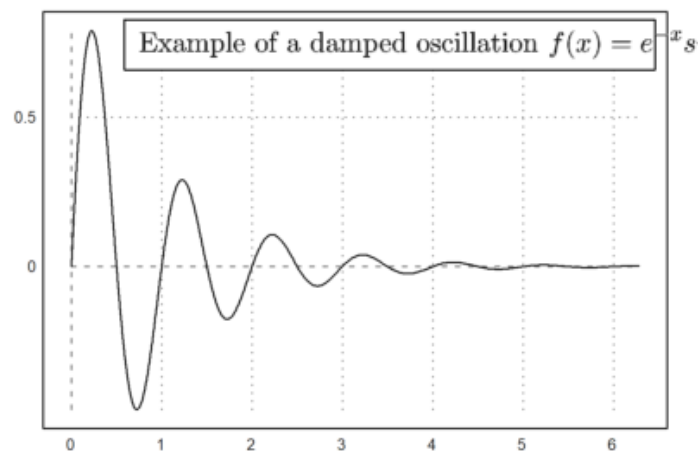
```
>statplot(1:10,random(2,10),color=[red,blue]):
```



Fitur serupa adalah fungsi `textbox()`.

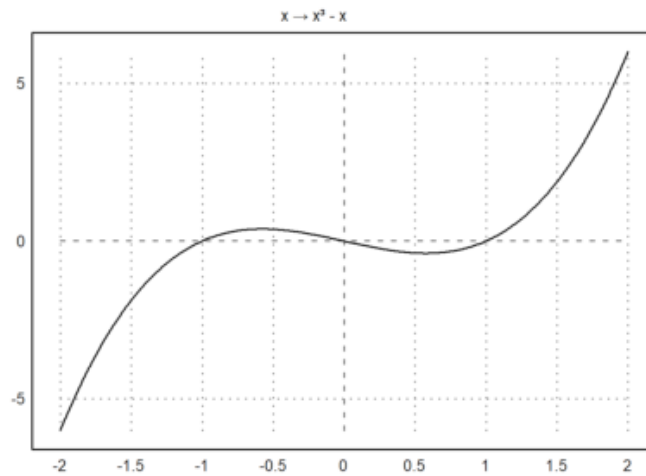
Lebar secara default adalah lebar maksimal dari baris teks. Namun, pengguna juga dapat mengaturnya.

```
>function f(x) &= exp(-x)*sin(2*pi*x); ...
>plot2d("f(x)",0,2pi); ...
>textbox(latex("\text{Example of a damped oscillation}\ f(x)=e^{-x}\sin(2\pi x)"),w=0.85):
```



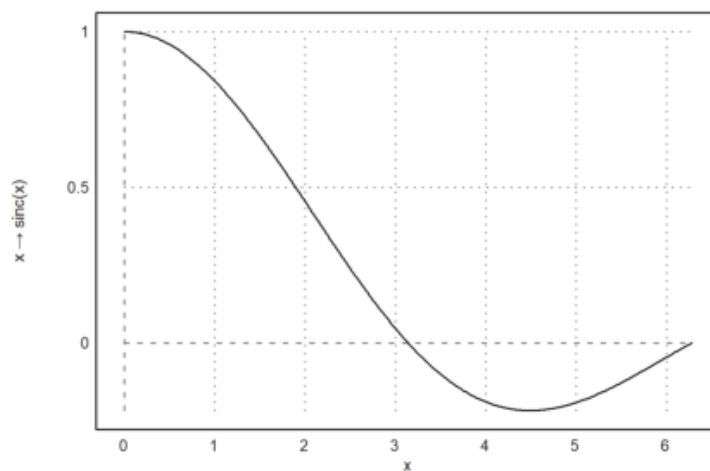
Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaks EMT untuk informasi lebih lanjut tentang string Unicode).

```
>plot2d("x^3-x",title=u"x \rarr; x&sup3; - x"):
```

Label pada sumbu x dan y dapat ditempatkan secara vertikal, begitu pula dengan sumbu itu sendiri.

```
>plot2d("sinc(x)",0,2pi,xl="x",yl="x \rarr; sinc(x)",>vertical):
```



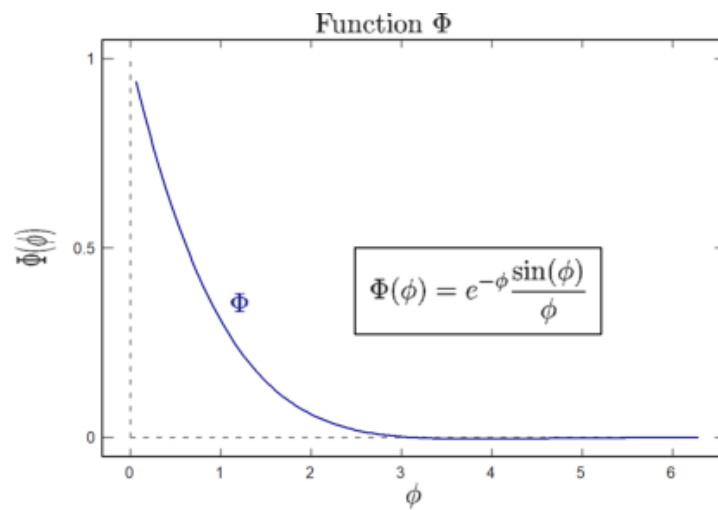
LaTeX

Anda juga dapat menampilkan rumus LaTeX jika sistem LaTeX telah terinstal. Saya merekomendasikan MiKTeX. Jalan menuju berkas biner “latex” dan “dvi2ps” harus ada di jalur sistem, atau Anda harus mengatur LaTeX di menu opsi.

Perhatikan bahwa pemrosesan LaTeX lambat. Jika Anda ingin menggunakan LaTeX dalam grafik animasi, panggil fungsi latex() sekali sebelum loop dan gunakan hasilnya (gambar dalam matriks RGB).

Dalam grafik berikut, kami menggunakan LaTeX untuk label x dan y, label, kotak label, dan judul grafik.

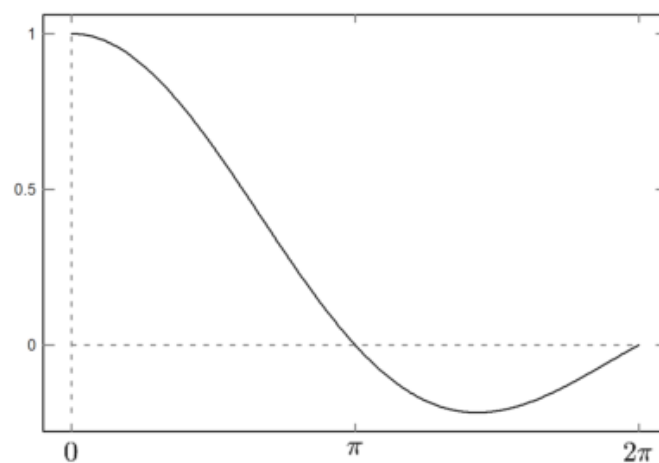
```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
> title=latex("\text{Function } \Phi"), ...
> xl=latex("\phi"),yl=latex("\Phi(\phi)")); ...
>textbox( ...
> latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}",x=0.8,y=0.5); ...
>label(latex("\Phi",color=blue),1,0.4):
```



Seringkali, kita menginginkan spasi yang tidak konvensional dan label teks pada sumbu x. Kita dapat menggunakan `xaxis()` dan `yaxis()` seperti yang akan kita tunjukkan nanti.

Cara termudah adalah membuat plot kosong dengan bingkai menggunakan `grid=4`, lalu menambahkan grid dengan `ygrid()` dan `xgrid()`. Dalam contoh berikut, kita menggunakan tiga string LaTeX untuk label pada sumbu x dengan `xtick()`.

```
>plot2d("sinc(x)",0,2pi,grid=4,<ticks); ...
>ygrid(-2:0.5:2,grid=6); ...
>xgrid([0:2]*pi,<ticks,grid=6); ...
>xtick([0,pi,2pi],["0","\pi","2\pi"],>latex):
```



Tentu saja, fungsi juga dapat digunakan.

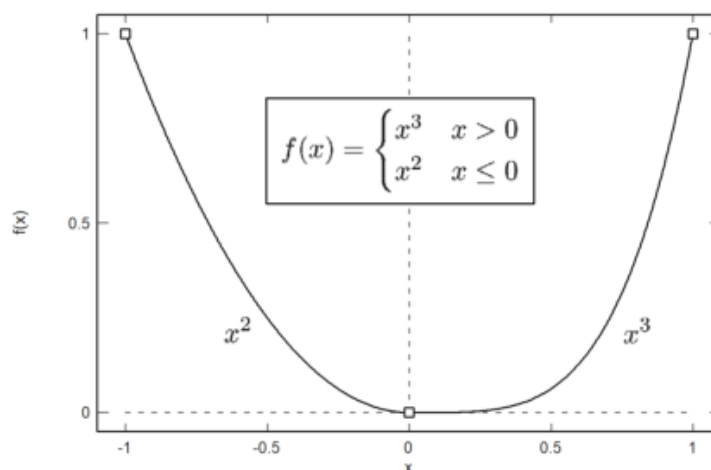
```
>function map f(x) ...

  if x>0 then return x^4
  else return x^2
  endif
endfunction
```

Parameter “map” membantu menggunakan fungsi untuk vektor. Untuk grafik, hal ini tidak diperlukan. Namun, untuk menunjukkan bahwa vektorisasi berguna, kami menambahkan beberapa titik kunci ke grafik pada $x=-1$, $x=0$, dan $x=1$.

Pada plot berikut, kami juga memasukkan beberapa kode LaTeX. Kami menggunakannya untuk dua label dan kotak teks. Tentu saja, Anda hanya dapat menggunakan LaTeX jika LaTeX telah diinstal dengan benar.

```
>plot2d("f",-1,1,xl="x",yl="f(x)",grid=6); ...
>plot2d([-1,0,1],f([-1,0,1]),>points,>add); ...
>label(latex("x^3"),0.72,f(0.72)); ...
>label(latex("x^2"),-0.52,f(-0.52),pos="ll"); ...
>textbox( ...
> latex("f(x)=\begin{cases} x^3 & x>0 \\ x^2 & x \leq 0 \end{cases}"), ...
> x=0.7,y=0.2):
```



Interaksi Pengguna

Saat menggambar fungsi atau ekspresi, parameter `>user` memungkinkan pengguna untuk memperbesar/memperkecil dan memindahkan grafik menggunakan tombol arah atau mouse. Pengguna dapat

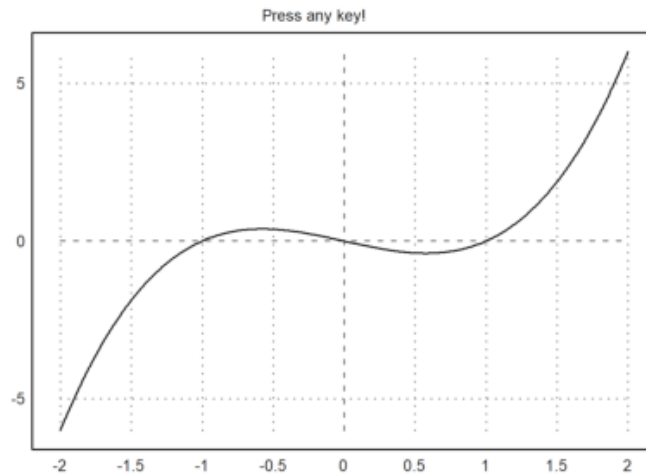
- memperbesar/memperkecil dengan + atau -
- memindahkan grafik dengan tombol arah
- memilih jendela grafik dengan mouse
- mereset tampilan dengan tombol spasi
- keluar dengan tombol enter

Tombol spasi akan mereset grafik ke jendela grafik asli.

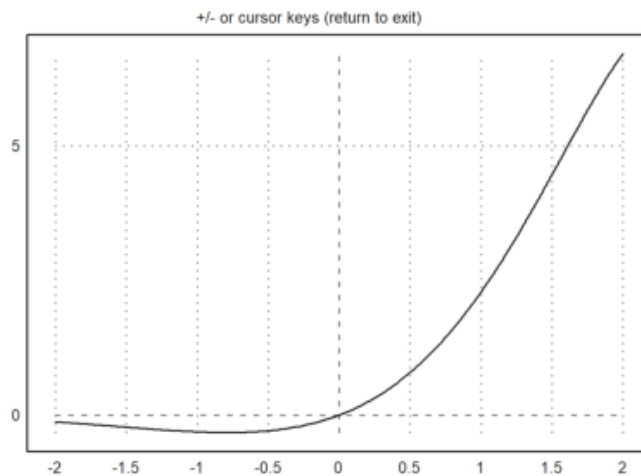
Saat menggambar data, bendera `>user` hanya akan menunggu input tombol.

```
>plot2d({{"x^3-a*x",a=1}},>user,title="Press any key!"):

```



```
>plot2d("exp(x)*sin(x)",user=true, ...
> title="+/- or cursor keys (return to exit)":
```



Berikut ini menunjukkan cara interaksi pengguna yang lebih canggih (lihat tutorial tentang pemrograman untuk detailnya).

Fungsi bawaan `mousedrag()` menunggu peristiwa mouse atau keyboard. Fungsi ini melaporkan peristiwa mouse ditekan, mouse dipindahkan, atau mouse dilepaskan, serta penekanan tombol keyboard. Fungsi `drag-points()` memanfaatkan ini dan memungkinkan pengguna untuk menyeret titik mana pun dalam grafik.

Kita memerlukan fungsi plot terlebih dahulu. Sebagai contoh, kita melakukan interpolasi pada 5 titik menggunakan polinomial. Fungsi ini harus menggambar grafik pada area plot yang tetap.

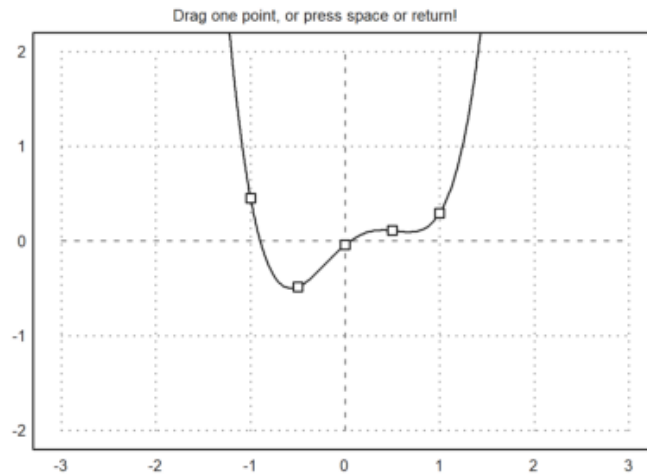
```
>function plotf(xp,yp,select) ...

    d=interp(xp,yp);
    plot2d("interpval(xp,d,x)";d,xp,r=2);
    plot2d(xp,yp,>points,>add);
    if select>0 then
        plot2d(xp[select],yp[select],color=red,>points,>add);
    endif;
    title("Drag one point, or press space or return!");
endfunction
```

Perhatikan parameter titik koma dalam plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp(). Tanpa ini, kita harus membuat fungsi plotinterp() terlebih dahulu, mengakses nilai-nilai secara global.

Sekarang kita menghasilkan beberapa nilai acak, dan membiarkan pengguna menyeret titik-titik tersebut.

```
>t=-1:0.5:1; dragpoints("plotf",t,random(size(t))-0.5):
```



Ada juga fungsi yang menggambar grafik suatu fungsi berdasarkan vektor parameter, dan memungkinkan pengguna untuk menyesuaikan parameter-parameter tersebut.

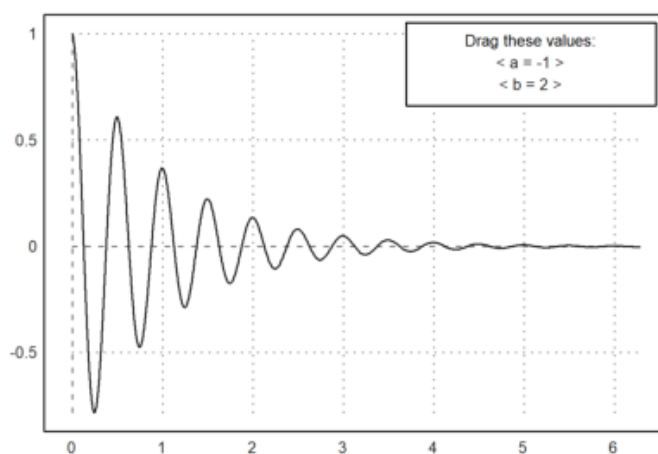
Pertama, kita membutuhkan fungsi plot.

```
>function plotf([a,b]) := plot2d("exp(a*x)*cos(2pi*b*x)",0,2pi;a,b);
```

Kemudian kita memerlukan nama untuk parameter, nilai awal, dan matriks nx2 dari rentang, serta baris judul (jika diperlukan).

Terdapat penggeser interaktif yang memungkinkan pengguna untuk mengatur nilai. Fungsi dragvalues() menyediakan fitur ini.

```
>dragvalues("plotf",["a","b],[-1,2],[[-2,2];[1,10]], ...  
> heading="Drag these values:",hcolor=black):
```



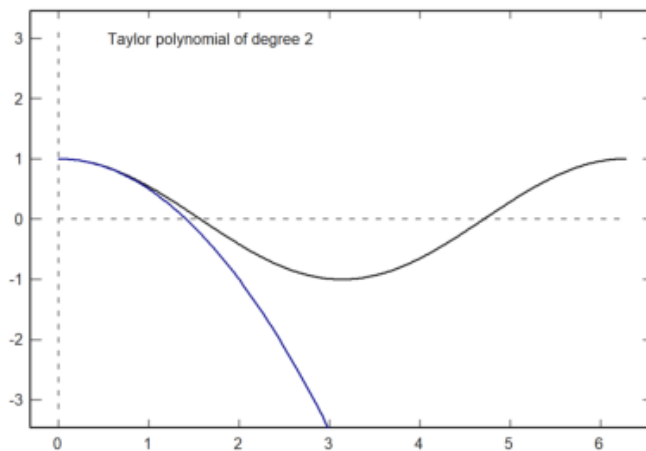
Dimungkinkan untuk membatasi nilai yang ditarik menjadi bilangan bulat. Sebagai contoh, kita menulis fungsi plot yang menggambar polinomial Taylor derajat n dari fungsi cosinus.

```
>function plotf(n) ...
```

```
    plot2d("cos(x)",0,2pi,>square,grid=6);  
    plot2d("&taylor(cos(x),x,0,@n)",color=blue,>add);  
    textbox("Taylor polynomial of degree "+n,0.1,0.02,style="t",>left);  
endfunction
```

Sekarang kita membiarkan nilai n bervariasi dari 0 hingga 20 dalam 20 langkah. Hasil dari `dragvalues()` digunakan untuk menggambar sketsa dengan nilai n ini, dan untuk menyisipkan grafik ke dalam notebook.

```
>nd=dragvalues("plotf","degree",2,[0,20],20,y=0.8, ...  
> heading="Drag the value:"); ...  
>plotf(nd):
```



Berikut ini adalah demonstrasi sederhana dari fungsi tersebut. Pengguna dapat menggambar di jendela plot, meninggalkan jejak titik-titik.

```
>function dragtest ...
```

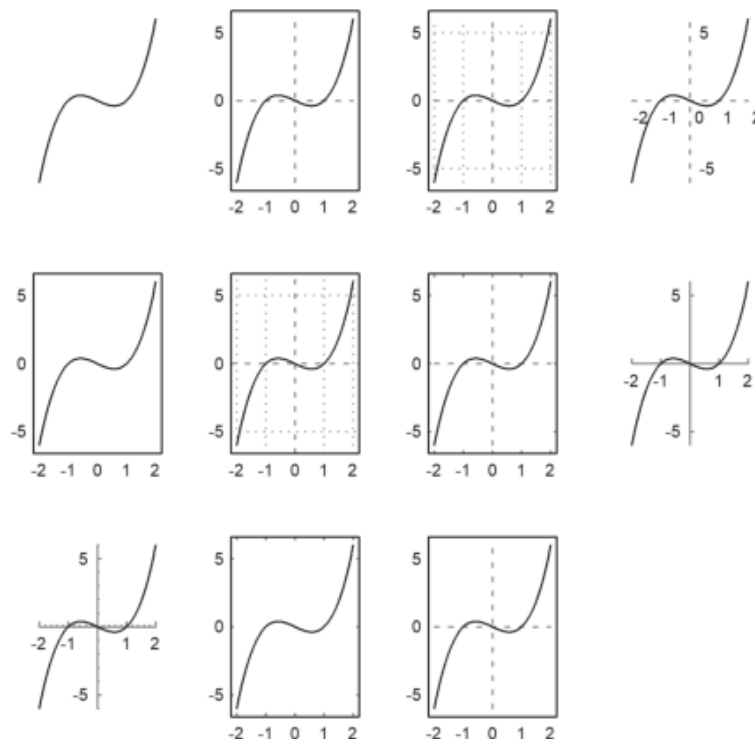
```
    plot2d(none,r=1,title="Drag with the mouse, or press any key!");  
    start=0;  
    repeat  
        {flag,m,time}=mousedrag();  
        if flag==0 then return; endif;  
        if flag==2 then  
            hold on; mark(m[1],m[2]); hold off;  
        endif;  
    end  
endfunction
```

```
>dragtest // lihat hasilnya dan cobalah lakukan!
```

Gaya Grafik 2D

Secara default, EMT menghitung tanda sumbu otomatis dan menambahkan label pada setiap tanda. Hal ini dapat diubah dengan parameter grid. Gaya default sumbu dan label dapat dimodifikasi. Selain itu, label dan judul dapat ditambahkan secara manual. Untuk mengembalikan ke gaya default, gunakan fungsi reset().

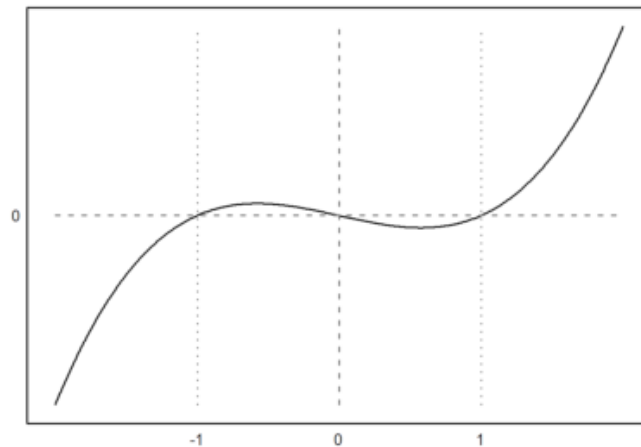
```
>aspect();  
>figure(3,4); ...  
> figure(1); plot2d("x^3-x",grid=0); ... // no grid, frame or axis  
> figure(2); plot2d("x^3-x",grid=1); ... // x-y-axis  
> figure(3); plot2d("x^3-x",grid=2); ... // default ticks  
> figure(4); plot2d("x^3-x",grid=3); ... // x-y- axis with labels inside  
> figure(5); plot2d("x^3-x",grid=4); ... // no ticks, only labels  
> figure(6); plot2d("x^3-x",grid=5); ... // default, but no margin  
> figure(7); plot2d("x^3-x",grid=6); ... // axes only  
> figure(8); plot2d("x^3-x",grid=7); ... // axes only, ticks at axis  
> figure(9); plot2d("x^3-x",grid=8); ... // axes only, finer ticks at axis  
> figure(10); plot2d("x^3-x",grid=9); ... // default, small ticks inside  
> figure(11); plot2d("x^3-x",grid=10); ...// no ticks, axes only  
> figure(0):
```



Parameter <frame> mematikan bingkai, dan framecolor=blue mengatur warna bingkai menjadi biru.

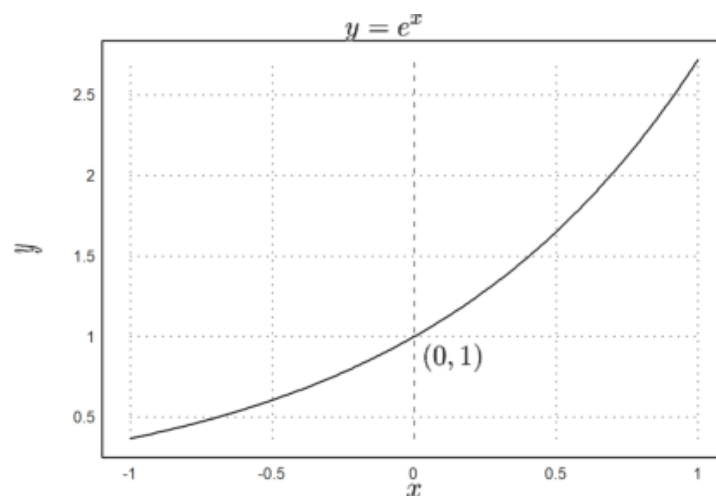
Jika Anda ingin menggunakan tanda sendiri, Anda dapat menggunakan style=0, dan menambahkan semuanya nanti.

```
>aspect(1.5);  
>plot2d("x^3-x",grid=0); // plot  
>frame; xgrid([-1,0,1]); ygrid(0): // add frame and grid
```



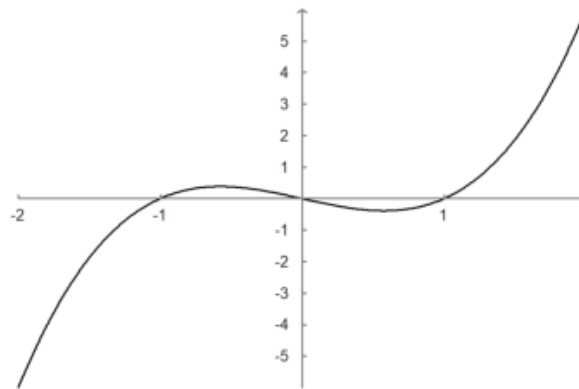
Untuk judul grafik dan label sumbu, lihat contoh berikut.

```
>plot2d("exp(x)",-1,1);  
>textcolor(black); // set the text color to black  
>title(latex("y=e^x")); // title above the plot  
>xlabel(latex("x")); // "x" for x-axis  
>ylabel(latex("y"),>vertical); // vertical "y" for y-axis  
>label(latex("(0,1)"),0,1,color=blue): // label a point
```



Sumbu dapat digambar secara terpisah menggunakan fungsi `xaxis()` dan `yaxis()`.

```
>plot2d("x^3-x",<grid,<frame);  
>xaxis(0,xx=-2:1,style="->"); yaxis(0,yy=-5:5,style="->");
```

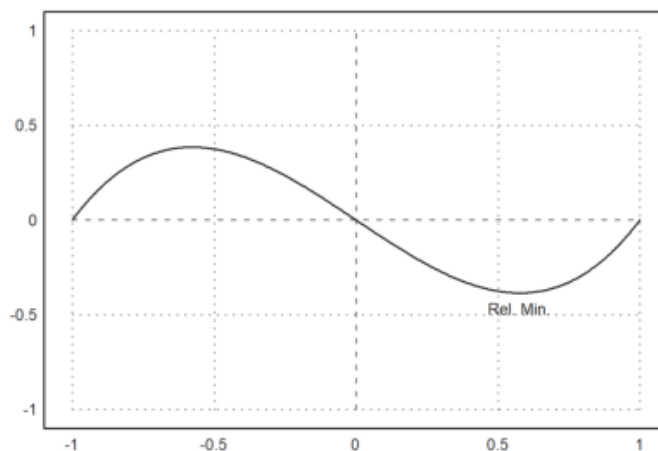


Teks pada grafik dapat diatur menggunakan fungsi `label()`. Dalam contoh berikut, "lc" berarti bagian tengah bawah. Fungsi ini mengatur posisi label relatif terhadap koordinat grafik.

```
>function f(x) &= x^3-x
```

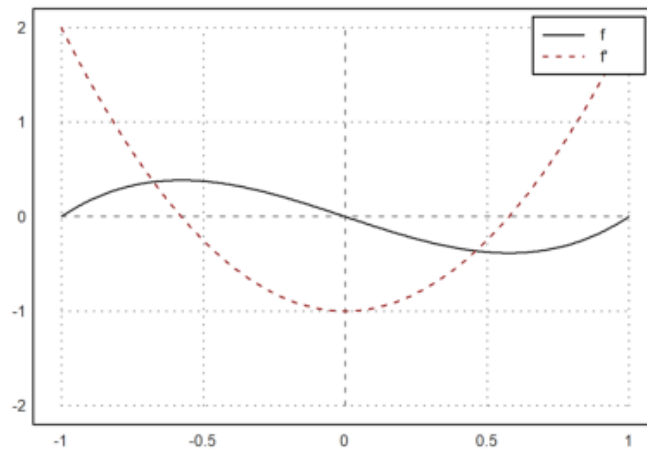
$$x^3 - x$$

```
>plot2d(f,-1,1,>square);  
>x0=fmin(f,0,1); // compute point of minimum  
>label("Rel. Min.",x0,f(x0),pos="lc"): // add a label there
```

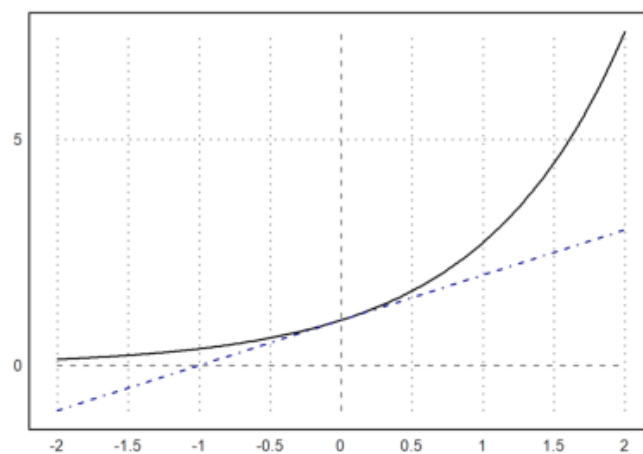


Ada juga kotak teks.

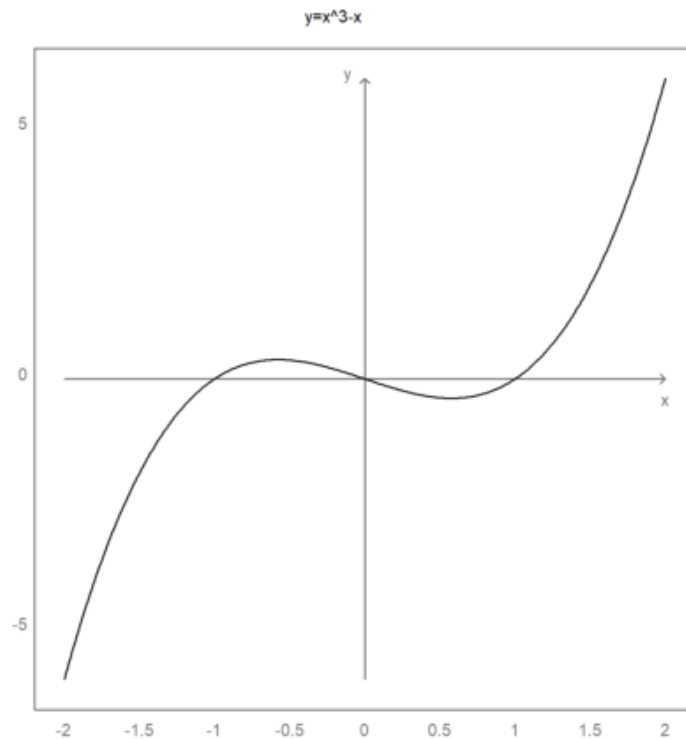
```
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // derivative
>labelbox(["f","f'"],["-","--"],[black,red]): // label box
```



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



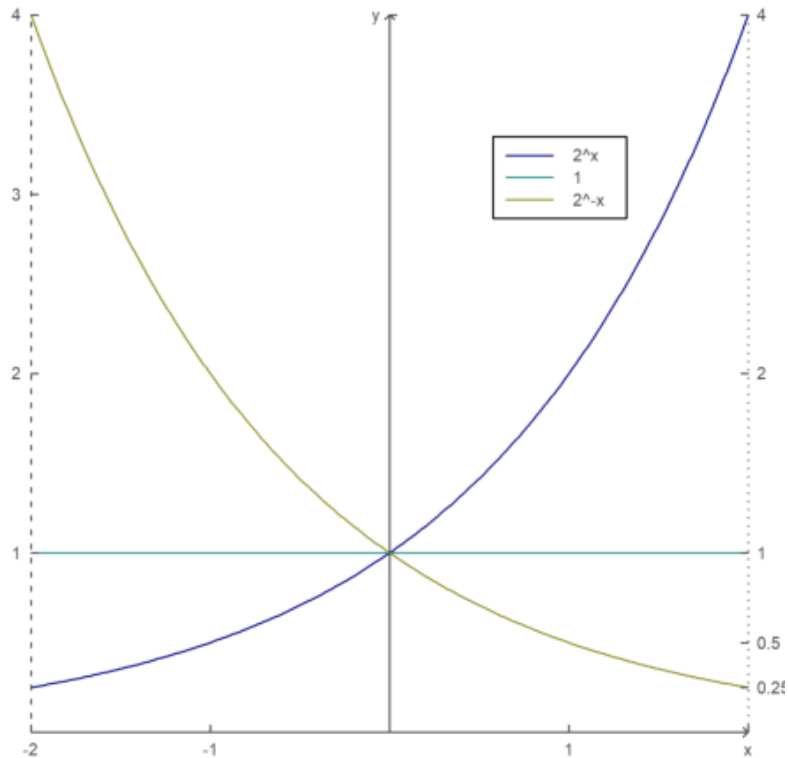
```
>gridstyle("->",color=gray,textcolor=gray,framecolor=gray); ...
> plot2d("x^3-x",grid=1); ...
> setttitle("y=x^3-x",color=black); ...
> label("x",2,0,pos="bc",color=gray); ...
> label("y",0,6,pos="cl",color=gray); ...
> reset():
```



Untuk kontrol yang lebih besar, sumbu x dan sumbu y dapat diatur secara manual.

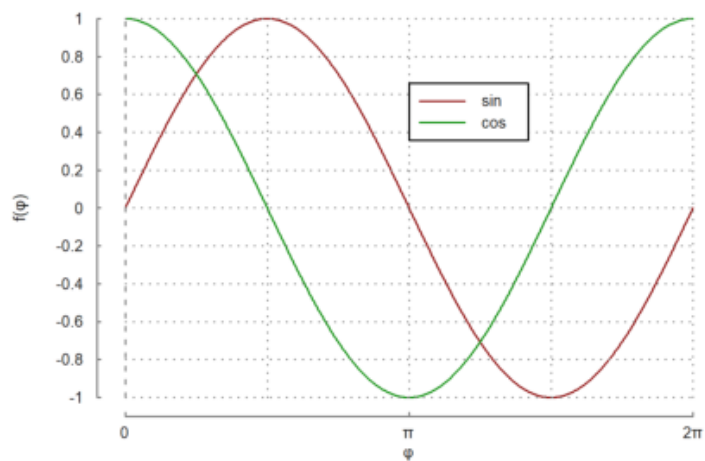
Perintah `fullwindow()` memperluas jendela plot karena kita tidak lagi memerlukan ruang untuk label di luar jendela plot. Gunakan `shrinkwindow()` atau `reset()` untuk mengembalikan ke pengaturan default.

```
>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 adalah 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;)"):
```



Membuat Grafik 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 jari-jari r dapat ditentukan, atau jendela grafik akan menyesuaikan secara otomatis dengan data. Sebagai alternatif, `>square` dapat diatur untuk mempertahankan rasio aspek persegi.

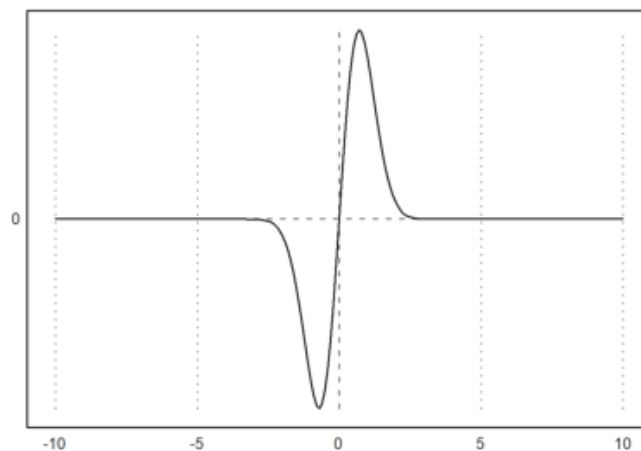
Membuat grafik ekspresi hanyalah singkatan untuk grafik data. Untuk grafik data, Anda memerlukan satu atau lebih baris nilai x , dan satu atau lebih baris nilai y . Dari rentang dan nilai x , fungsi `plot2d` akan menghitung data yang akan digambarkan, secara default dengan evaluasi adaptif fungsi. Untuk grafik titik gunakan `>points`, untuk grafik campuran garis dan titik gunakan `>addpoints`.

Namun, Anda dapat memasukkan data secara langsung.

- Gunakan vektor baris untuk x dan y untuk satu fungsi.
- Matriks untuk x dan y akan digambar baris per baris.

Berikut adalah contoh dengan satu baris untuk x dan y .

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



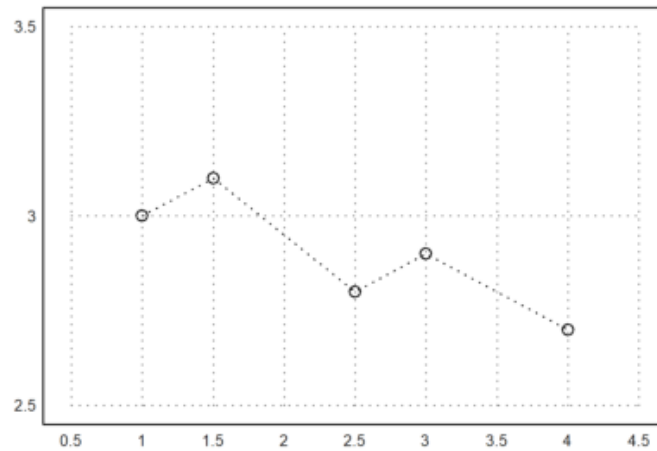
Data juga dapat ditampilkan sebagai titik. Gunakan `points=true` untuk ini. Grafik ini berfungsi seperti poligon, tetapi hanya menggambar sudut-sudutnya.

- `style="..."`: Pilih dari `"["`, `"<"`, `"o"`, `"."`, `".."`, `"+"`, `"*"`, `"["`, `"<"`, `"o"`, `"."`, `"|"`.

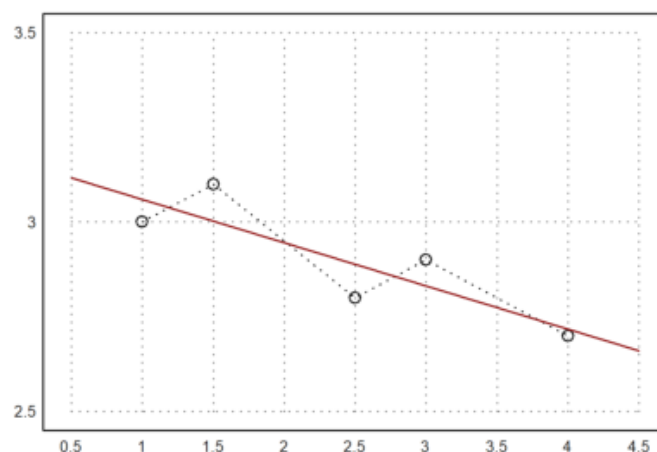
Untuk menggambar kumpulan titik, gunakan `>points`. Jika warna adalah vektor warna, setiap titik akan mendapatkan warna yang berbeda. Untuk matriks koordinat dan vektor kolom, warna diterapkan pada baris matriks.

Parameter `>addpoints` menambahkan titik ke 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="."); // lines
>plot2d(xdata,ydata,>points,>add,style="o"): // add points
```



```
>p=polyfit(xdata,ydata,1); // get regression line
>plot2d("polyval(p,x)",>add,color=red): // add plot of line
```



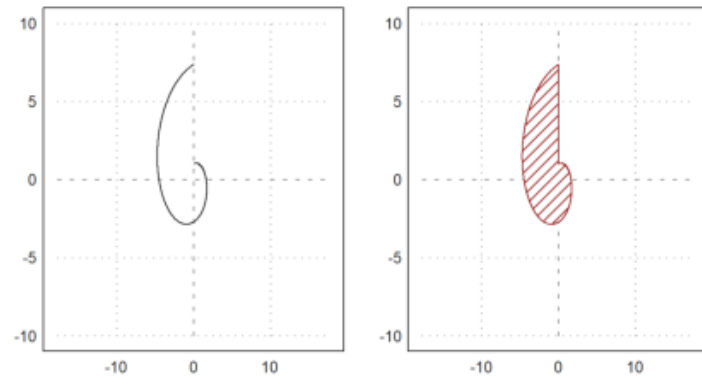
Menggambar Daerah yang Dibatasi Kurva

Grafik data sebenarnya adalah poligon. Kita juga dapat menggambar kurva atau kurva yang diisi.

- filled=true mengisi grafik.
- style="...": Pilih dari " ", " / ", " \ ", " \ / ".
- fillcolor: Lihat di atas untuk warna yang tersedia.

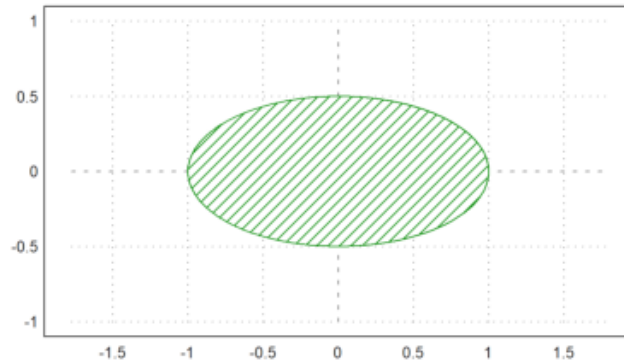
Warna isi ditentukan oleh argumen "fillcolor", dan opsi <outline mencegah penggambaran batas untuk semua gaya kecuali gaya default.

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

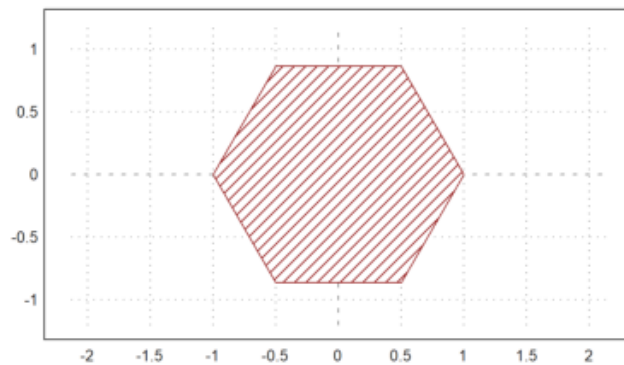


Dalam contoh-contoh berikut, kami menggambar elips terisi dan dua heksagon terisi menggunakan kurva tertutup dengan 6 titik dan gaya pengisian yang berbeda.

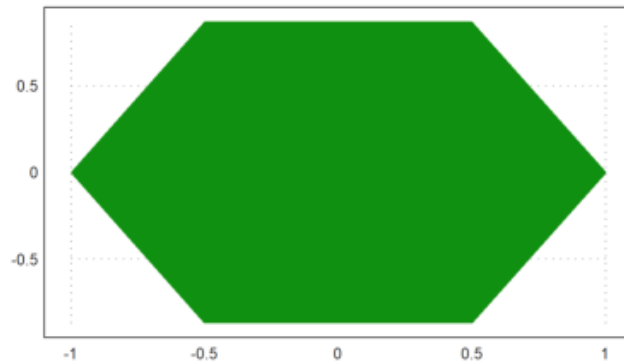
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```



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

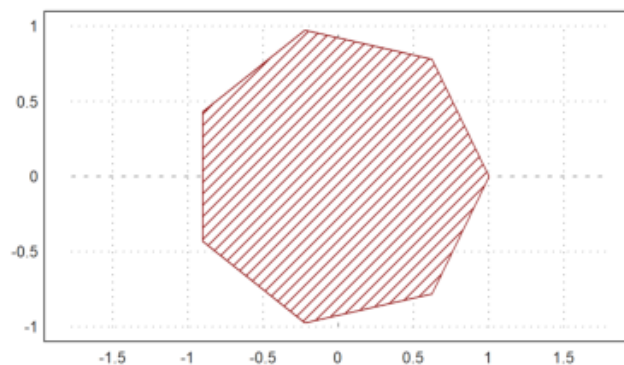


```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



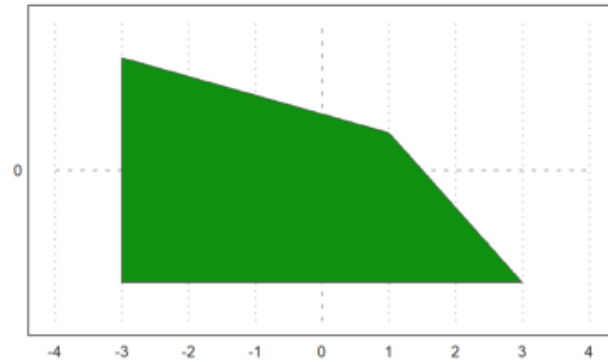
Contoh lain adalah heptagon, yang kita buat dengan 7 titik pada lingkaran satuan.

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



Berikut adalah himpunan nilai maksimal dari empat kondisi linier yang kurang dari atau sama dengan 3. Ini adalah $A[k].v \leq 3$ untuk semua baris A . Untuk mendapatkan sudut yang bagus, 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):
```

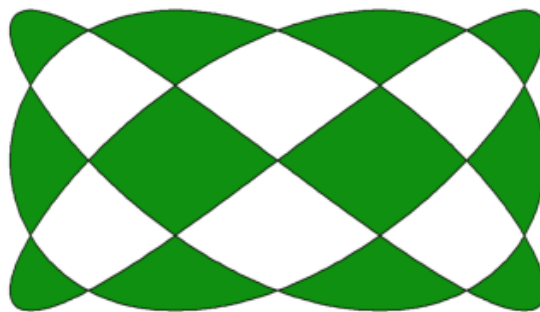



Inti dari bahasa matriks adalah bahwa ia memungkinkan untuk menghasilkan tabel fungsi dengan mudah.

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

Kami sekarang memiliki vektor x dan y yang berisi nilai-nilai. Fungsi plot2d() dapat menggambar nilai-nilai ini sebagai kurva yang menghubungkan titik-titik tersebut. Grafik dapat diisi. Dalam hal ini hal ini menghasilkan hasil yang bagus berkat aturan pembalikan, yang digunakan untuk mengisi grafik.

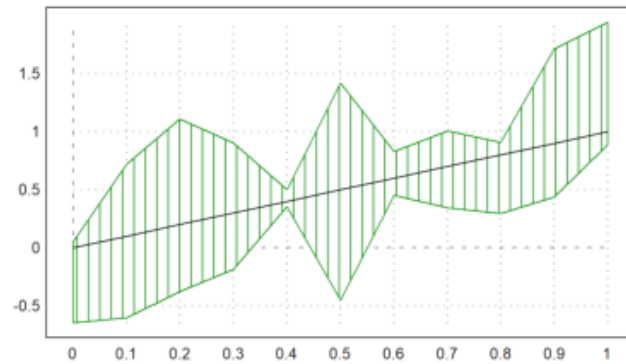
```
>plot2d(x,y,<grid,<frame,>filled):
```



Sebuah vektor interval digambar terhadap nilai x sebagai daerah terisi antara nilai bawah dan atas interval.

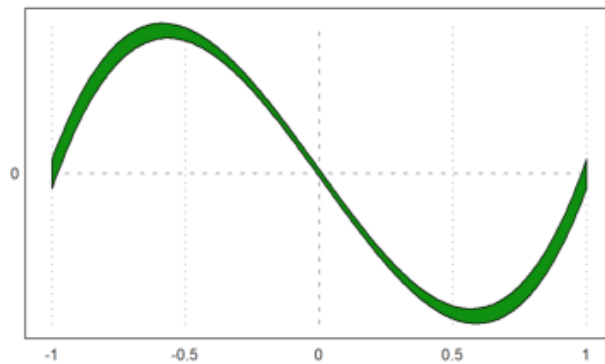
Ini dapat berguna untuk menggambar kesalahan perhitungan. Namun, hal ini juga dapat digunakan untuk menggambar kesalahan statistik.

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



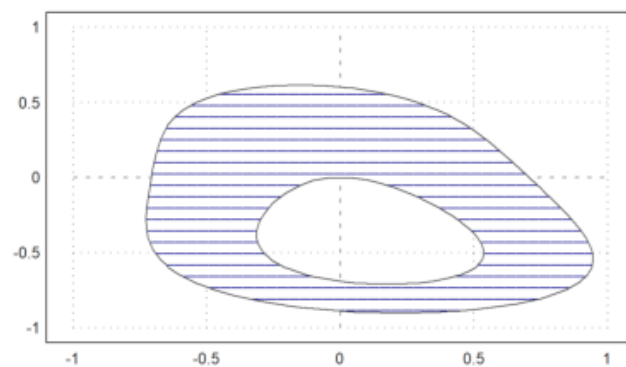
Jika x adalah vektor yang terurut, dan y adalah vektor interval, maka `plot2d` akan menggambar rentang terisi dari interval-interval tersebut di bidang datar. Gaya pengisiannya sama dengan gaya poligon..

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



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

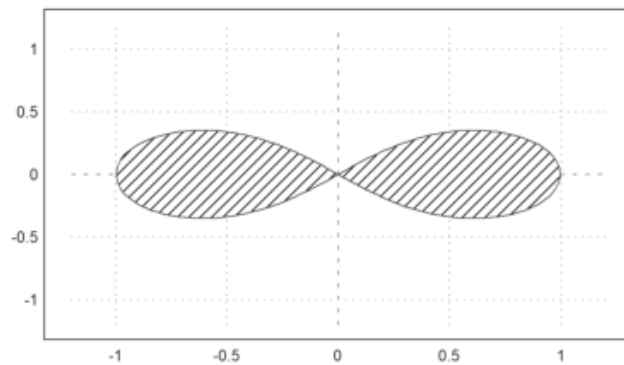
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



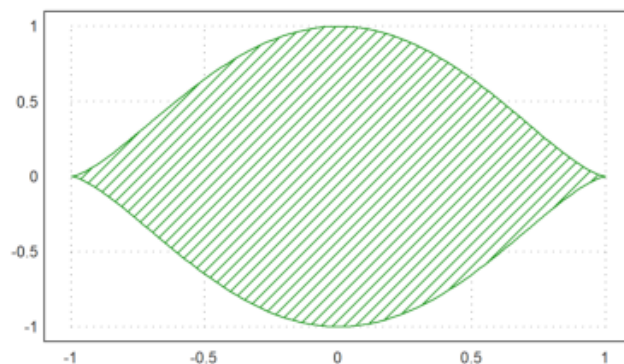
Kita juga dapat mengisi rentang nilai seperti

$$-1 \leq (x^2 + y^2)^2 - x^2 + y^2 \leq 0.$$

```
>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

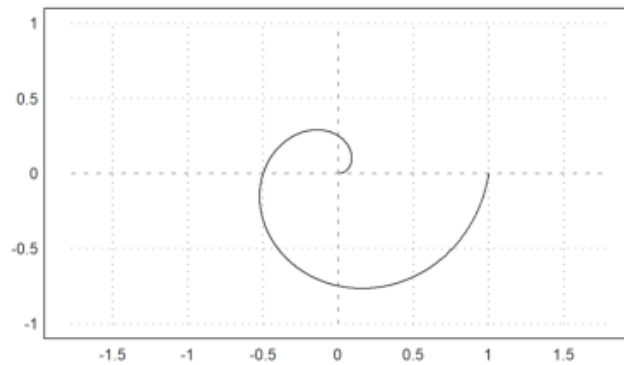
Nilai x tidak perlu diurutkan. (x,y) hanya menggambarkan sebuah kurva. Jika x diurutkan, kurva tersebut merupakan grafik dari suatu fungsi.

Dalam contoh berikut, kita menggambar spiral

$$\gamma(t) = t \cdot (\cos(2\pi t), \sin(2\pi t))$$

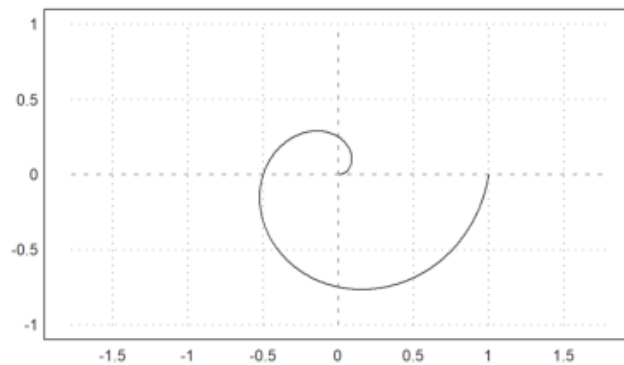
Kita perlu menggunakan banyak titik untuk tampilan yang halus atau fungsi `adaptive()` untuk mengevaluasi ekspresi (lihat fungsi `adaptive()` untuk detail lebih lanjut).

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

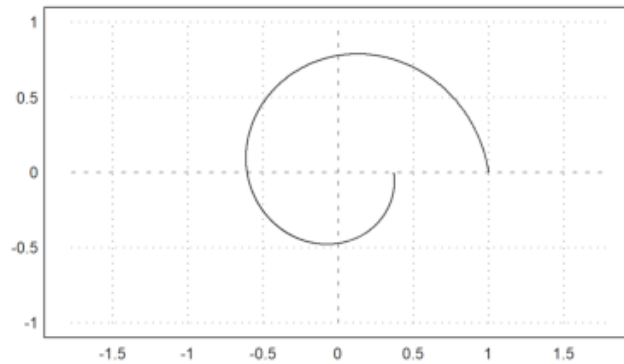


Sebagai alternatif, dimungkinkan untuk menggunakan dua ekspresi untuk kurva. Grafik berikut ini menggambarkan kurva yang sama seperti di atas.

```
>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(2*pi*t); y=r*sin(2*pi*t);  
>plot2d(x,y,r=1):
```



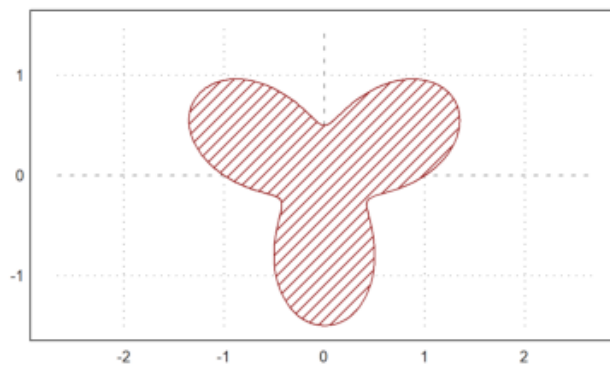
Dalam contoh berikut, kita menggambar kurva

$$\gamma(t) = (r(t) \cos(t), r(t) \sin(t))$$

dengan

$$r(t) = 1 + \frac{\sin(3t)}{2}.$$

```
>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

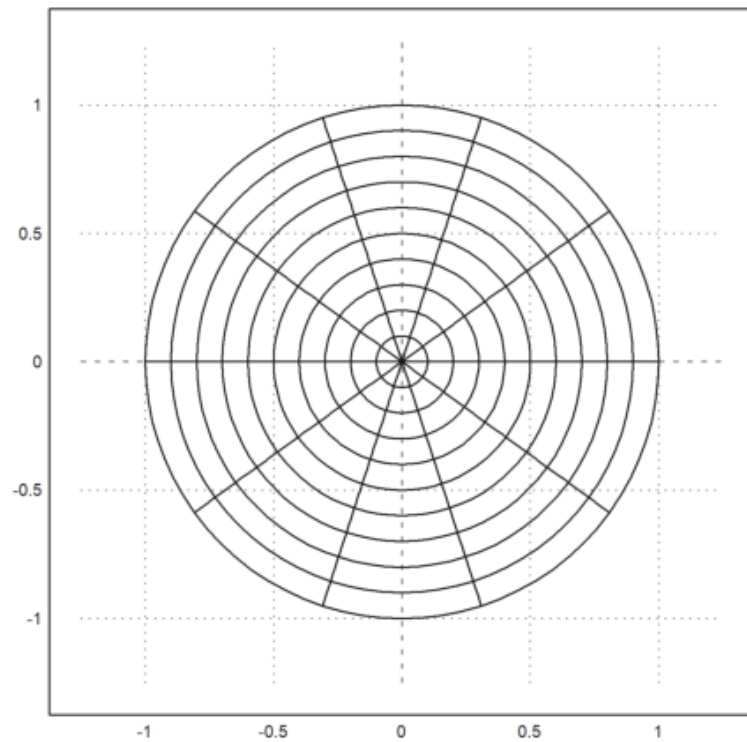
* Menggambar Grafik Bilangan Kompleks

An array of complex numbers can also be plotted. Then the grid points will be connected. If a number of grid lines is specified (or a 1x2 vector of grid lines) in the argument `cgrid` only those grid lines are visible.

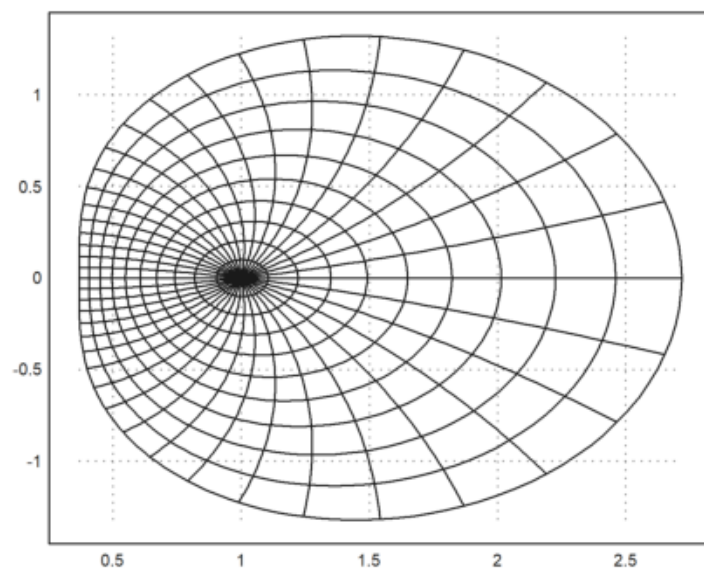
A matrix of complex numbers will automatically plot as a grid in the complex plane.

In the following example, we plot the image of the unit circle under the exponential function. The `cgrid` parameter hides some of the grid curves.

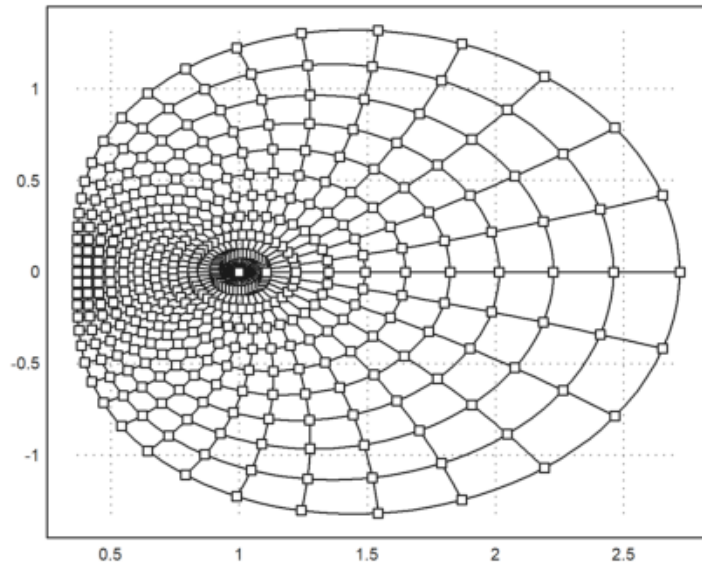
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```



```
>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):
```

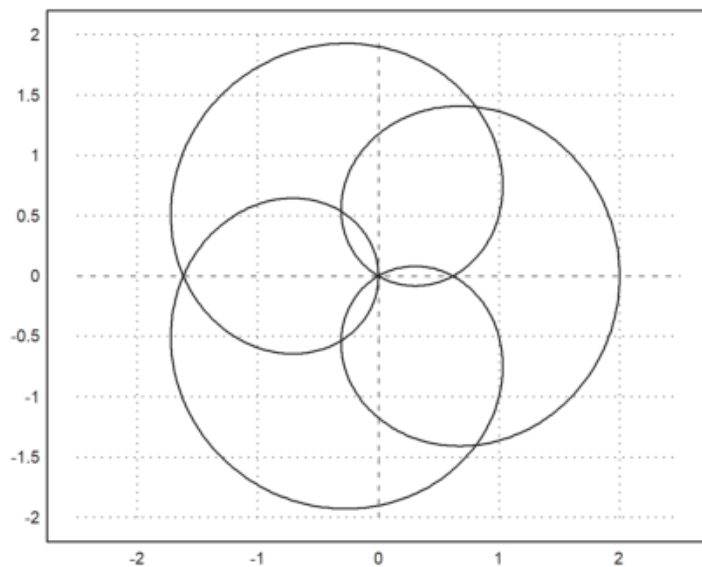


Vektor bilangan kompleks secara otomatis digambar sebagai kurva di bidang kompleks dengan bagian nyata dan bagian imajiner.

Dalam contoh ini, kita menggambar lingkaran satuan dengan

$$\gamma(t) = e^{it}$$

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



Ada banyak fungsi yang khusus digunakan untuk grafik statistik. Salah satu grafik yang sering digunakan adalah grafik kolom.

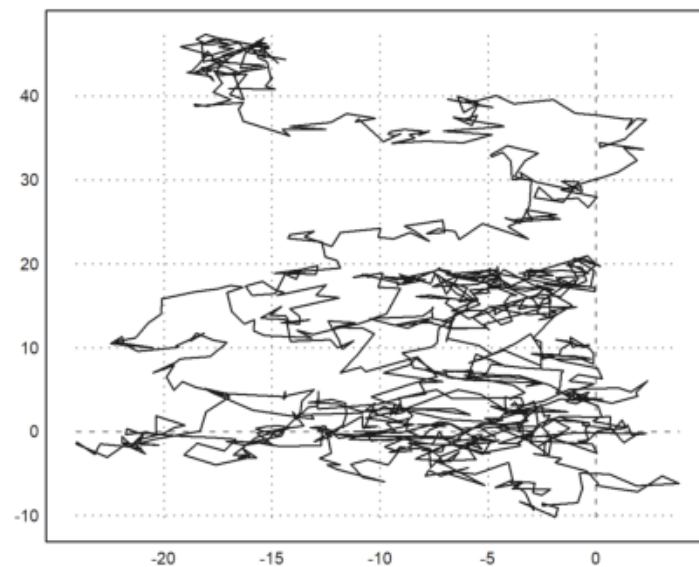
Jumlah kumulatif dari nilai-nilai yang terdistribusi normal 0-1 menghasilkan pergerakan acak.

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

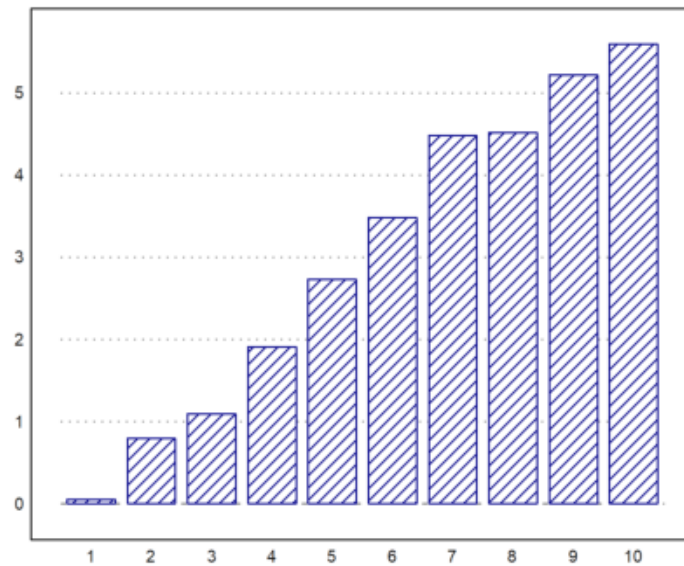


Menggunakan dua baris menunjukkan pergerakan dalam dua dimensi.

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

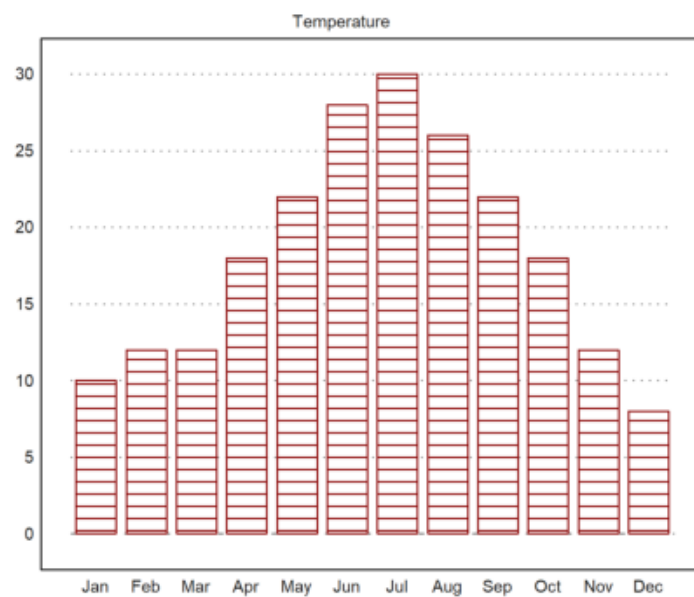



```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

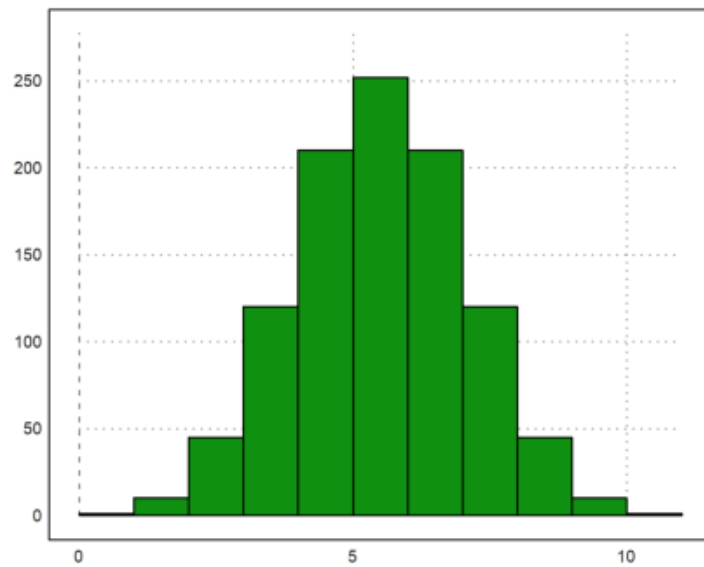


Ini juga dapat menampilkan string sebagai label.

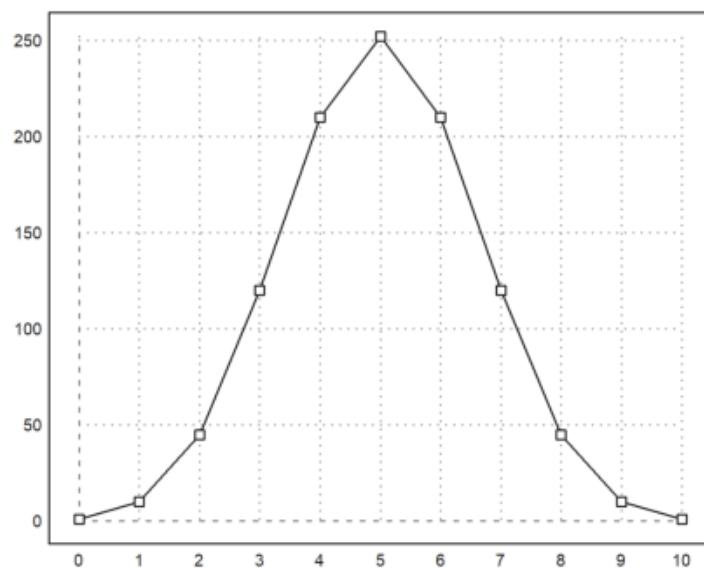
```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...  
> "Jul","Aug","Sep","Oct","Nov","Dec"];  
>values=[10,12,12,18,22,28,30,26,22,18,12,8];  
>columnsplot(values,lab=months,color=red,style="-");  
>title("Temperature"):
```



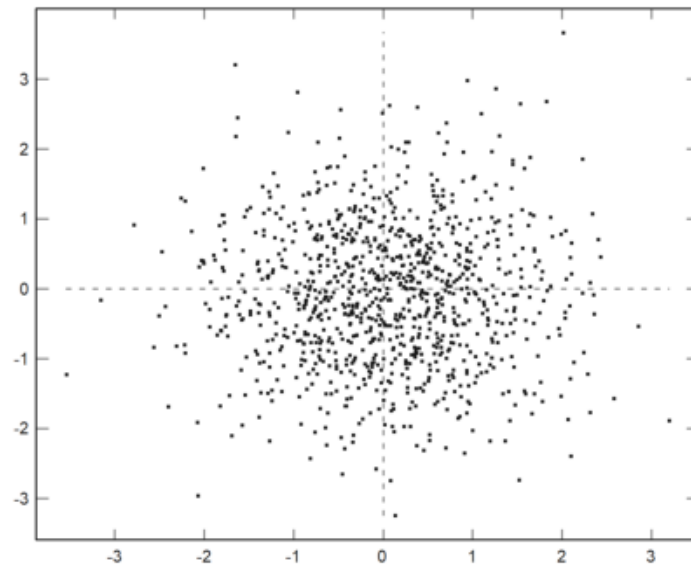
```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```



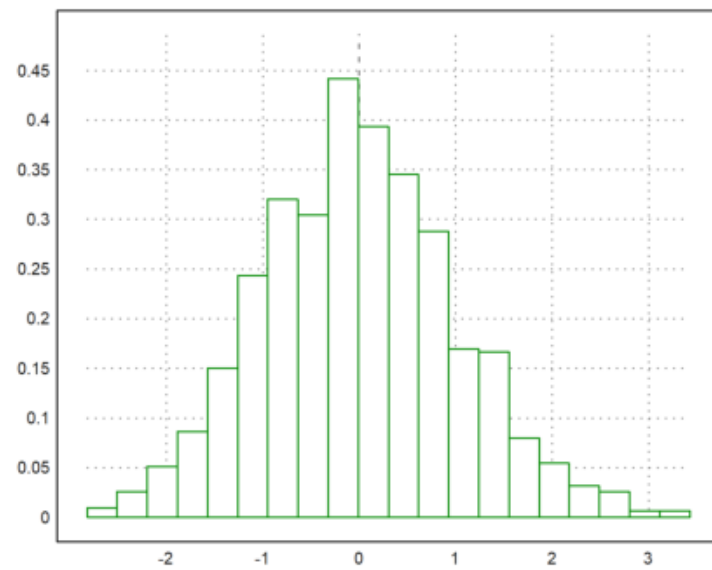
```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



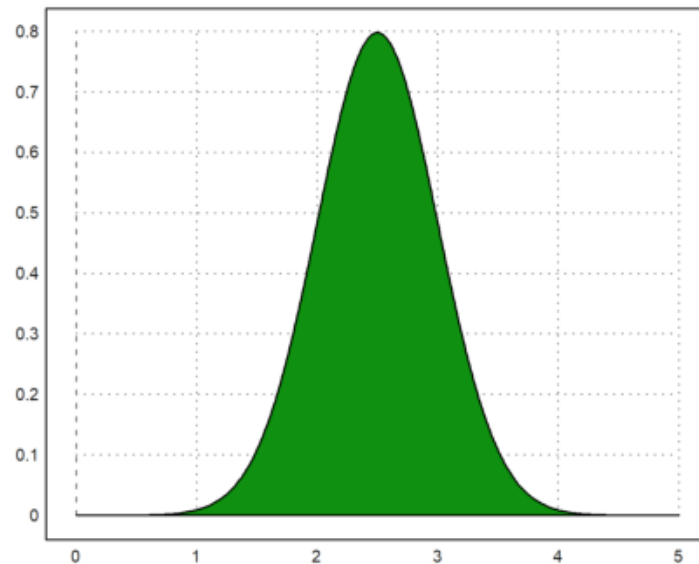
```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```



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

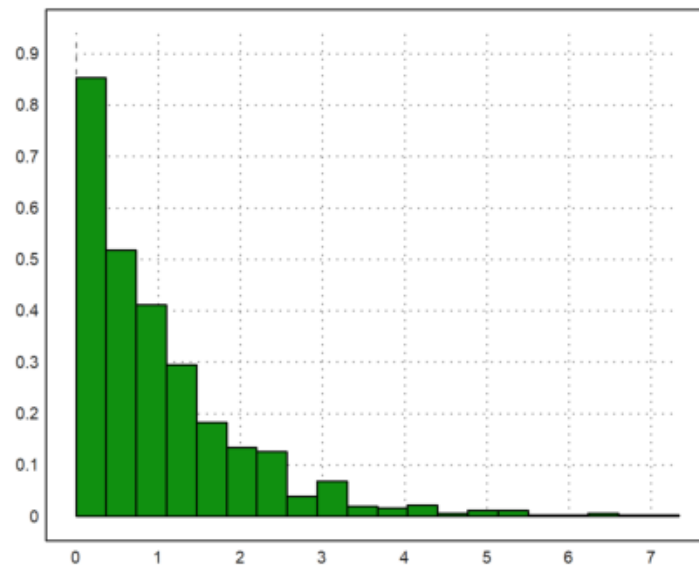


```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



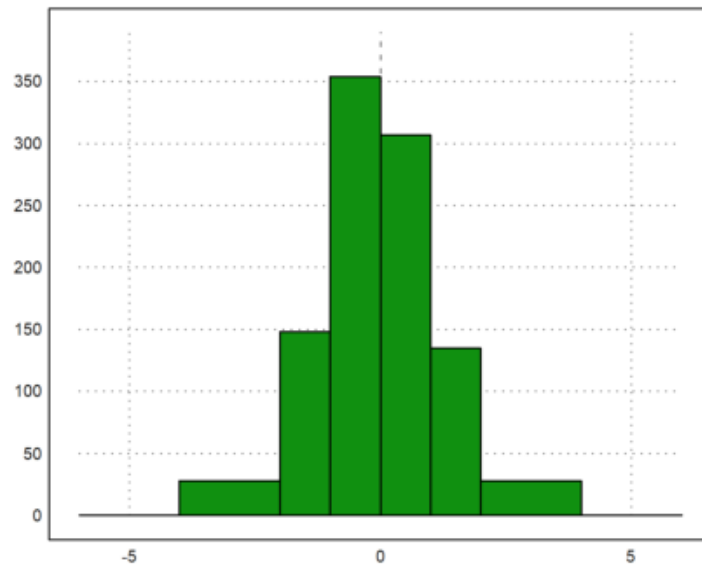
Untuk menggambar distribusi statistik eksperimental, Anda dapat menggunakan `distribution=n` dengan `plot2d`.

```
>w=randexponential(1,1000); // exponential distribution
>plot2d(w,>distribution): // or distribution=n with n intervals
```



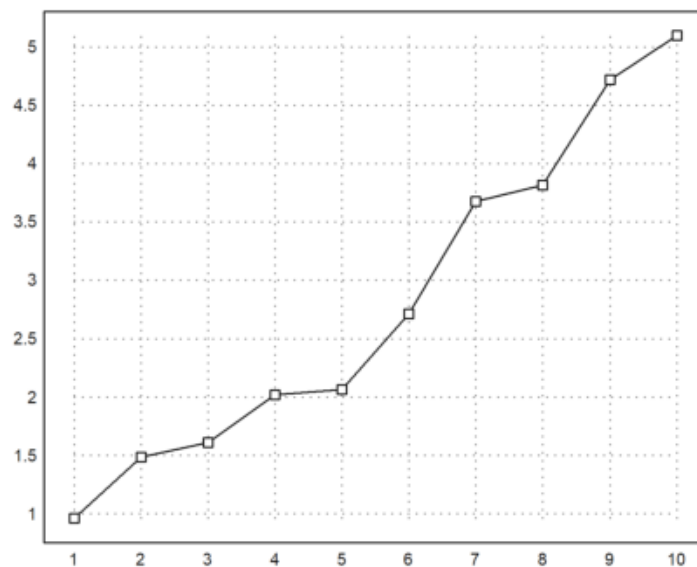
Atau Anda dapat menghitung distribusi dari data dan menggambar hasilnya dengan `>bar` di `plot3d`, atau dengan grafik kolom.

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

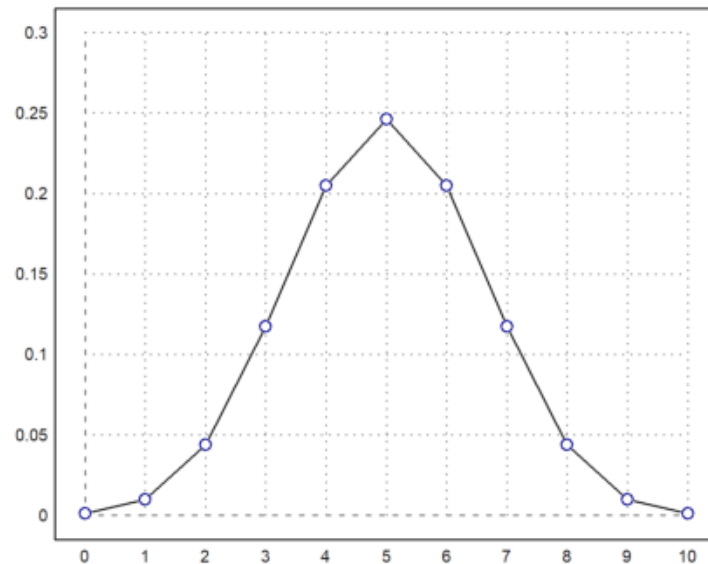


Fungsi `statplot()` menetapkan gaya dengan string sederhana.

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

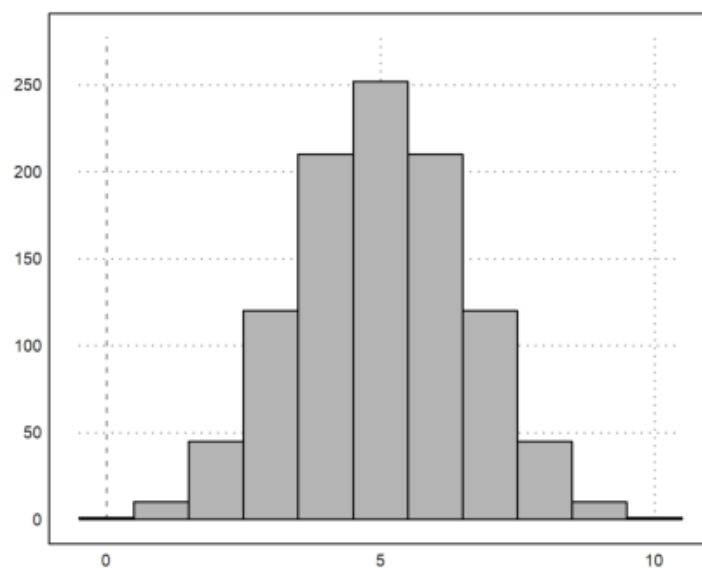


```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



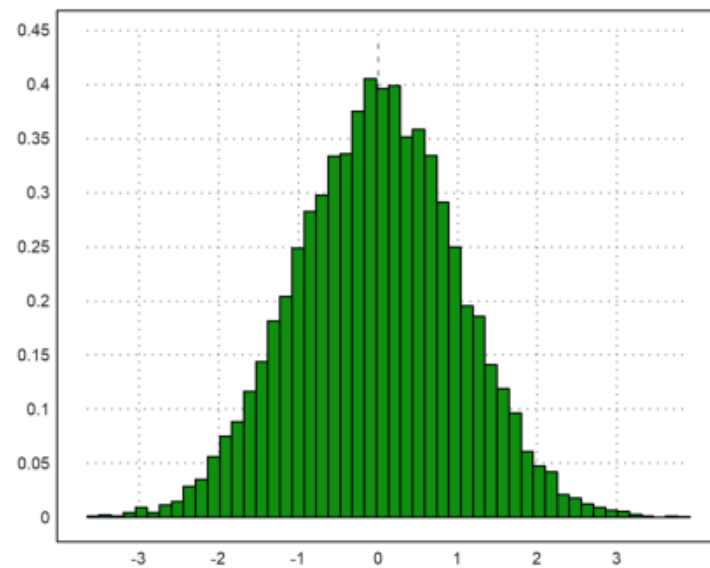
Selain itu, data dapat ditampilkan dalam bentuk batang. Dalam hal ini, x harus diurutkan dan memiliki satu elemen lebih panjang dari y . Batang-batang tersebut akan membentang dari $x[i]$ hingga $x[i+1]$ dengan nilai $y[i]$. Jika x memiliki ukuran yang sama dengan y , maka akan ditambahkan satu elemen dengan spasi terakhir. Gaya pengisian dapat digunakan seperti di atas.

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

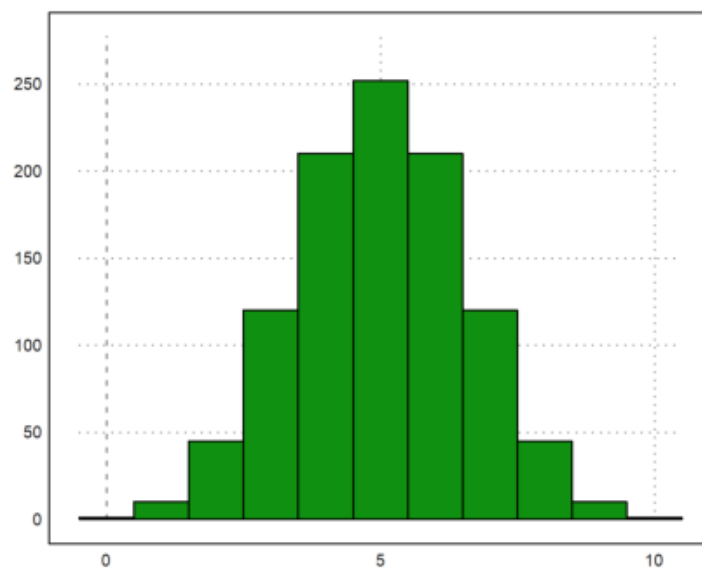


Data untuk diagram batang (`bar=1`) dan histogram (`histogram=1`) dapat diberikan secara eksplisit dalam xv dan yv , atau dihitung dari distribusi empiris dalam xv dengan `>distribution` (atau `distribution=n`). Histogram nilai xv akan dihitung secara otomatis dengan `>histogram`. Jika `>even` ditentukan,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

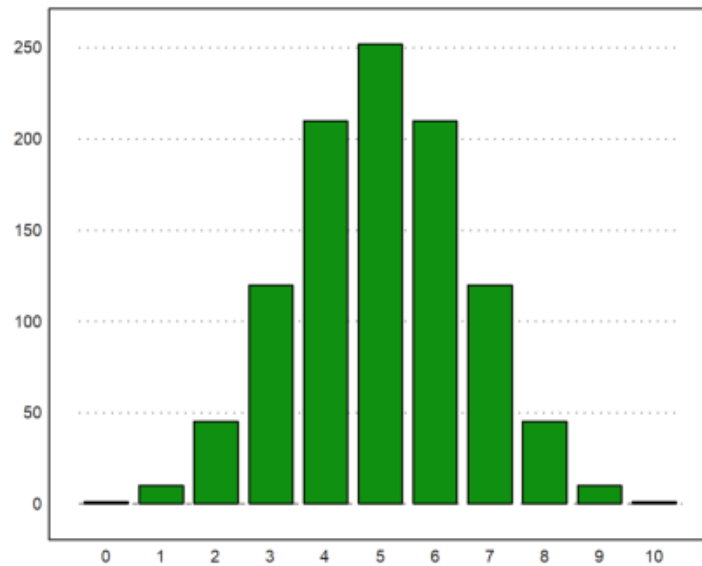
```
>plot2d(normal(10000),distribution=50):
```



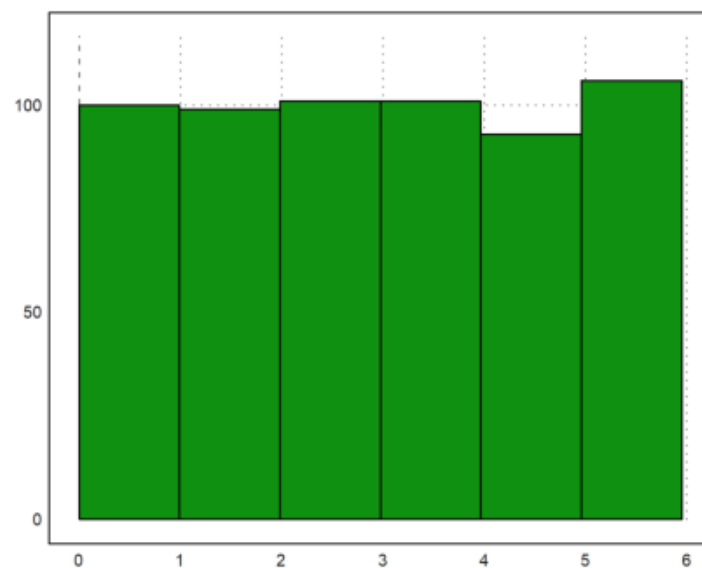
```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```



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

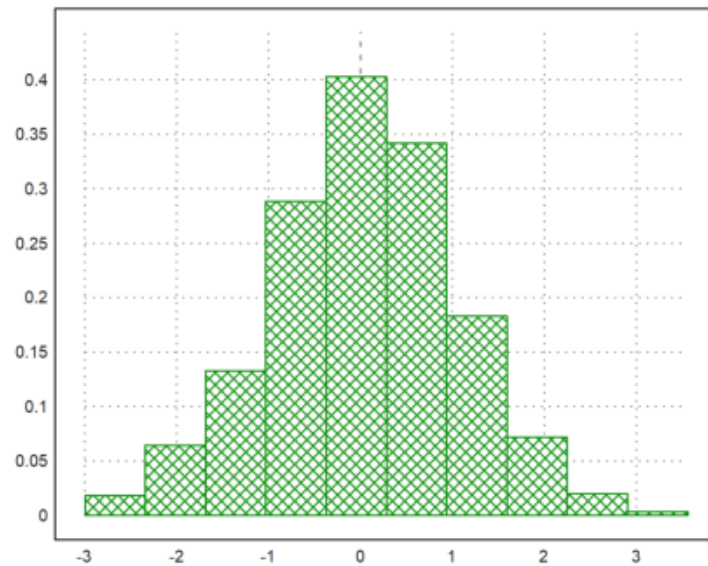


```
>plot2d(random(600)*6,histogram=6):
```



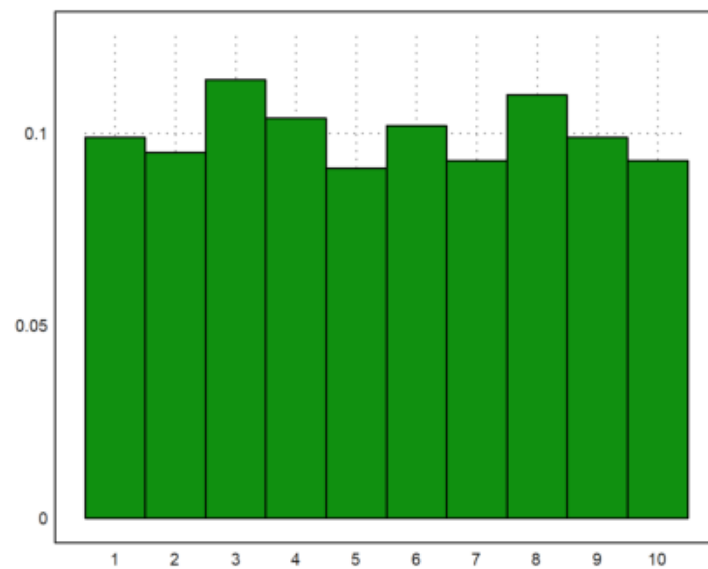
Untuk distribusi, terdapat parameter `distribution=n`, yang menghitung nilai secara otomatis dan menampilkan distribusi relatif dengan `n` sub-interval.

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

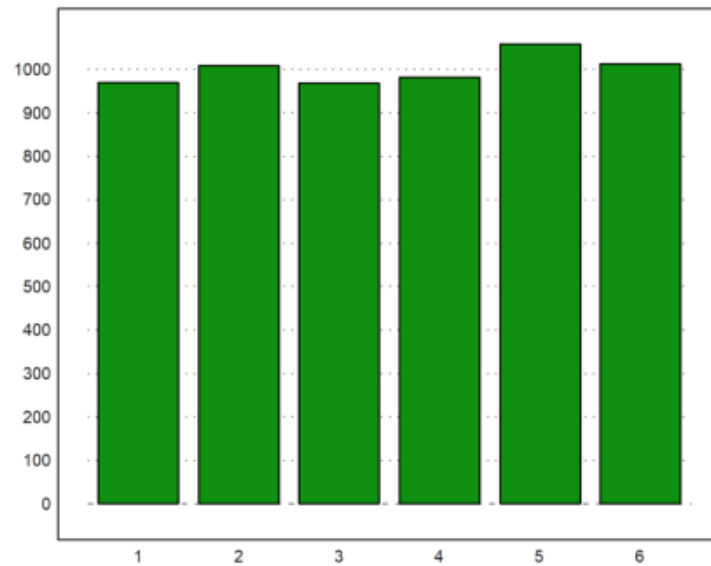
Dengan parameter `even=true`, ini akan menggunakan interval bilangan bulat.

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

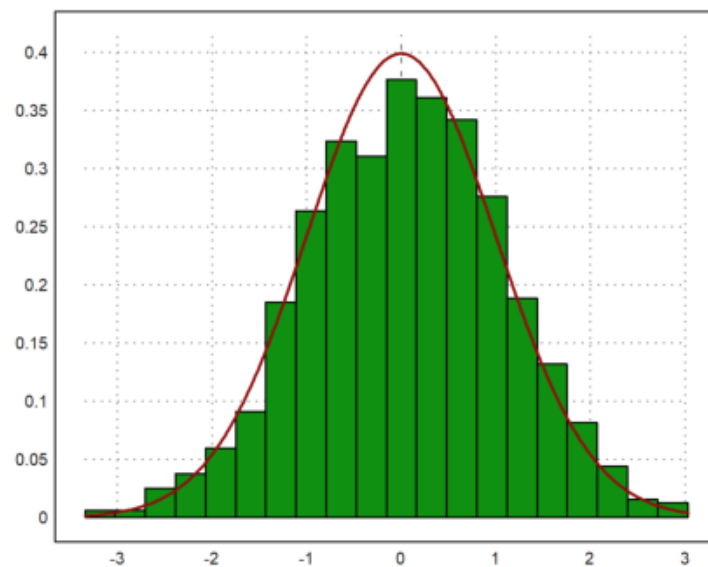


Perhatikan bahwa terdapat banyak grafik statistik yang mungkin berguna. Silakan lihat tutorial tentang statistik.

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

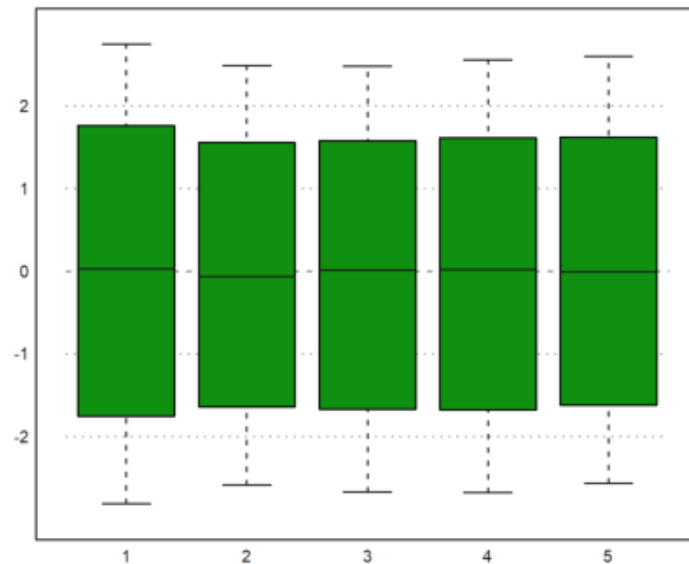


```
>plot2d(normal(1,1000),>distribution); ...
> plot2d("qnormal(x)",color=red,thickness=2,>add):
```



Ada juga banyak diagram khusus untuk statistik. Diagram kotak (boxplot) menampilkan kuartil dari distribusi ini dan banyak nilai penyimpangan (outlier). Secara definisi, nilai penyimpangan dalam diagram kotak adalah data yang melebihi 1,5 kali rentang 50% tengah dari diagram tersebut.

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



Fungsi Implisit

Grafik implisit menampilkan garis kontur yang memecahkan persamaan $f(x,y)=\text{level}$, di mana “level” dapat berupa nilai tunggal atau vektor nilai. Jika `level="auto"`, akan ada `nc` garis kontur yang tersebar secara merata antara nilai minimum dan maksimum fungsi. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan `>hue` untuk menunjukkan nilai fungsi. Untuk fungsi implisit, `xv` harus berupa fungsi atau ekspresi dari parameter `x` dan `y`, atau, alternatifnya, `xv` dapat berupa matriks nilai.

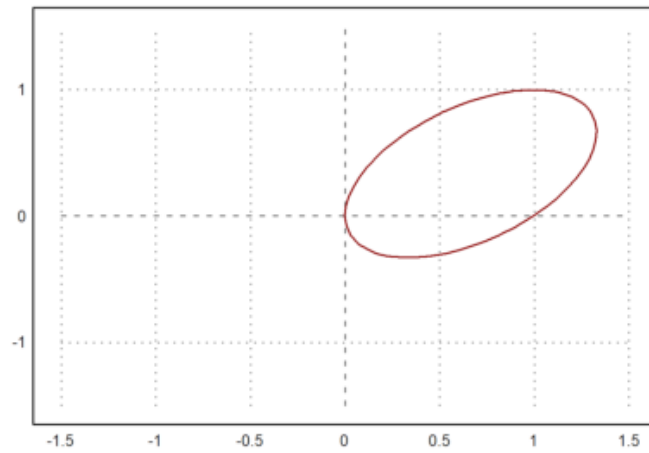
Euler dapat menandai garis tingkat

$$f(x,y) = c$$

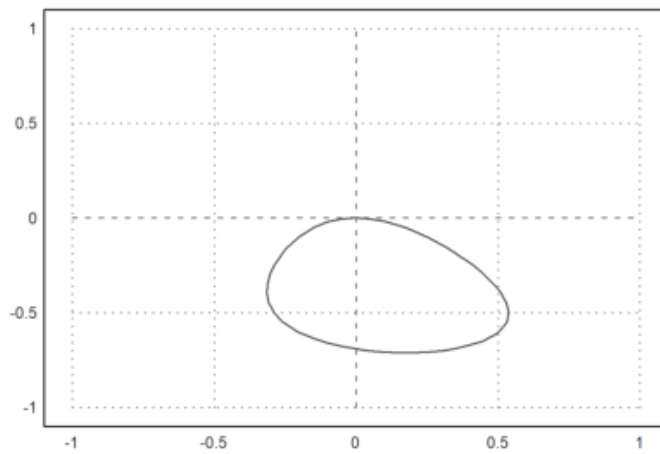
dari fungsi apa pun.

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 `level=c`, di mana `c` dapat berupa vektor garis tingkat. Selain itu, skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi di setiap titik pada plot. Parameter “`n`” menentukan ketelitian plot.

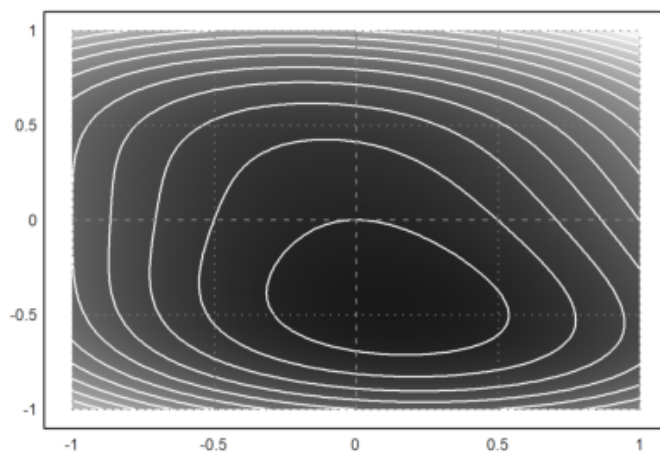
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```



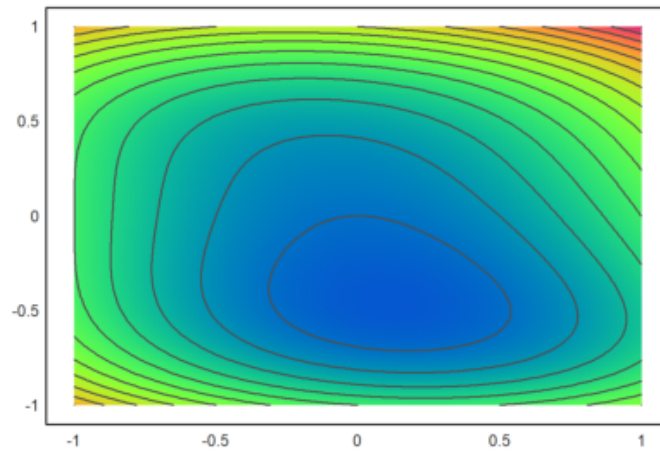
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0): // Solutions of f(x,y)=0
```



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

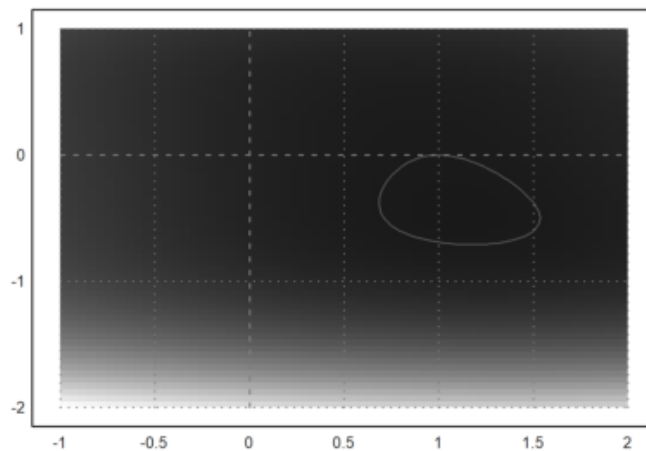


```
>plot2d(expr, level=0:0.5:20, >hue, >spectral, n=200, grid=4): // nicer
```

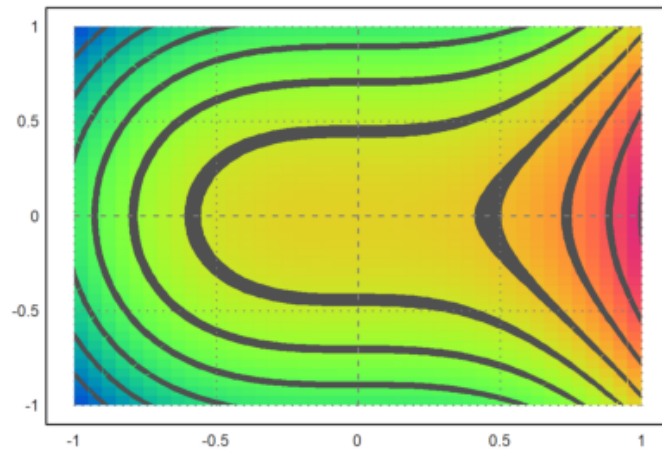


Ini juga berlaku untuk grafik data. Namun, Anda harus menentukan rentang untuk label sumbu.

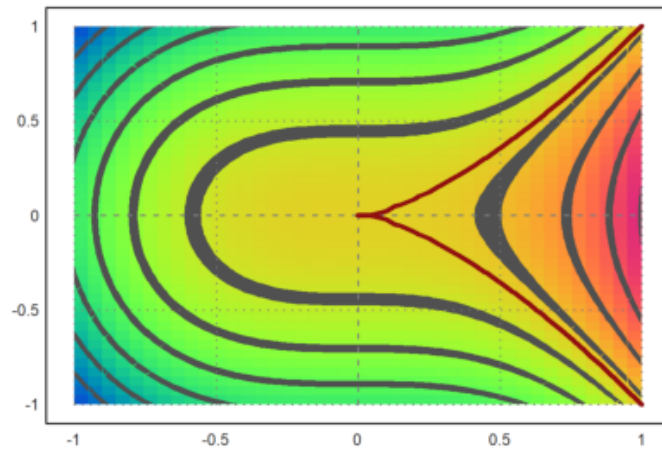
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z, level=0, a=-1, b=2, c=-2, d=1, >hue):
```



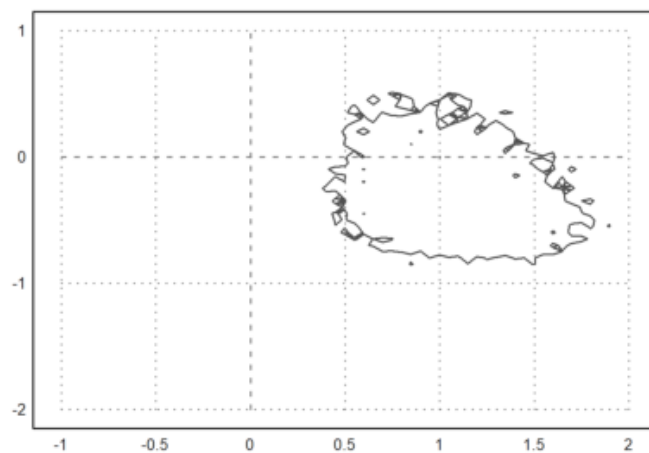
```
>plot2d("x^3-y^2", >contour, >hue, >spectral):
```



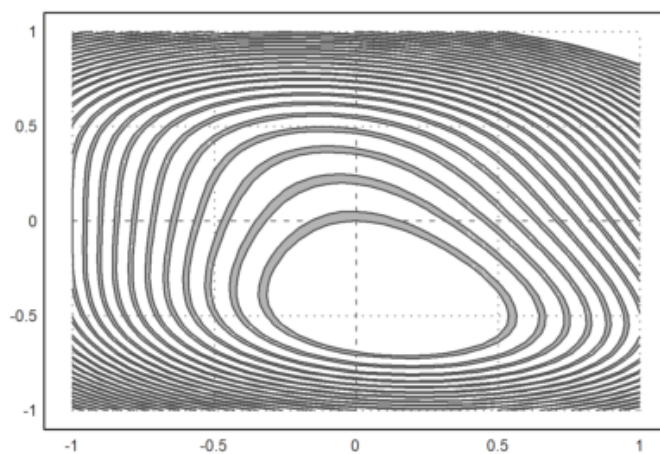
```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



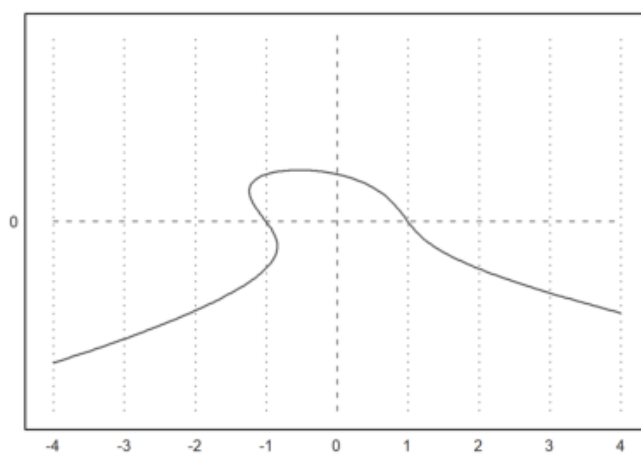
```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



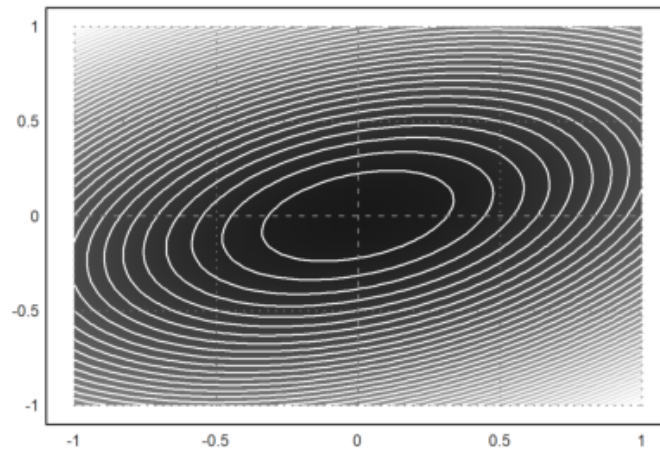
```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```



```
>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):
```



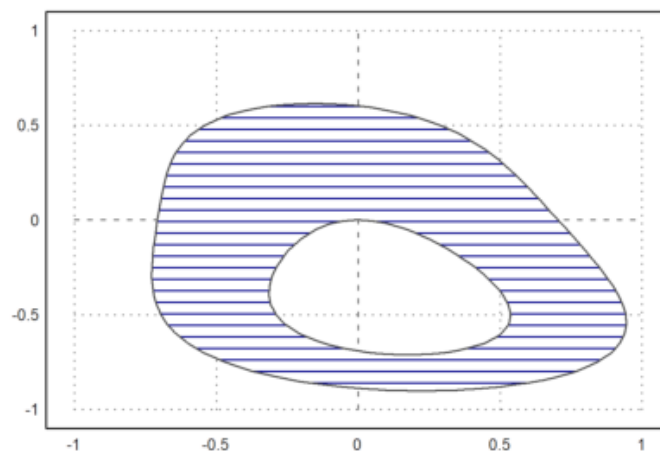
Juga dimungkinkan untuk mengisi himpunan

$$a \leq f(x, y) \leq b$$

dengan rentang level.

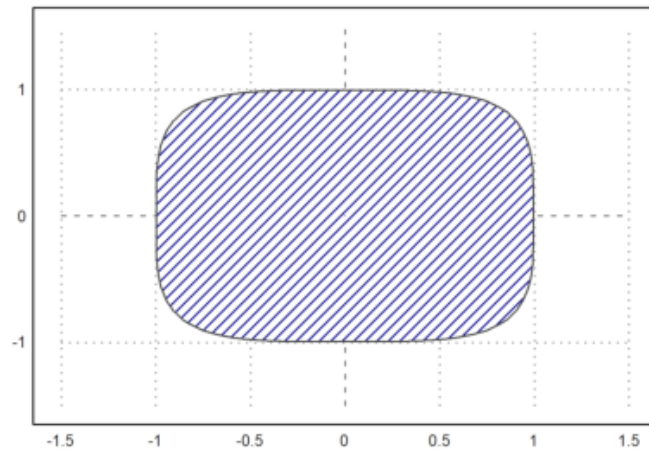
Dimungkinkan untuk mengisi daerah nilai untuk suatu fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama berisi batas bawah dan baris kedua berisi batas atas.

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

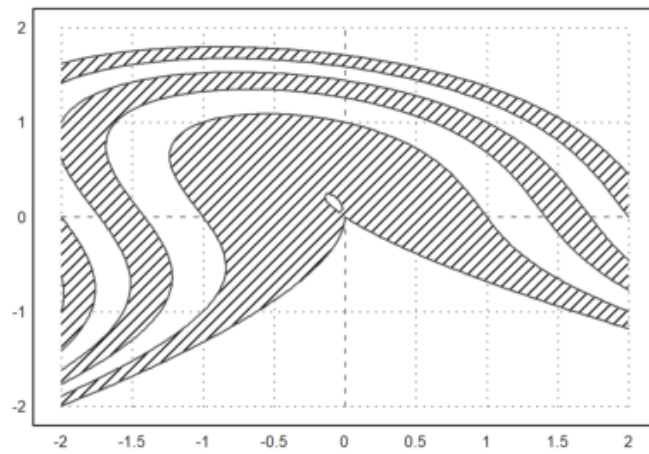


Grafik implisit juga dapat menampilkan rentang tingkat. Maka, tingkat harus berupa matriks 2xn yang berisi interval tingkat, di mana baris pertama berisi awal dan baris kedua berisi akhir dari setiap interval. Sebagai alternatif, vektor baris sederhana dapat digunakan untuk tingkat, dan parameter dl memperluas nilai tingkat menjadi interval.

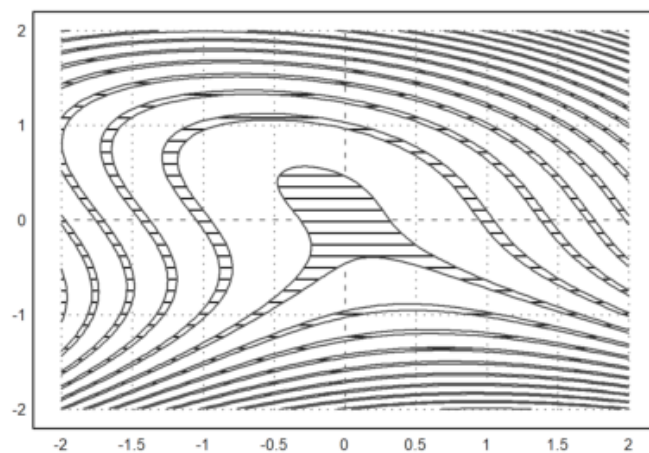
```
>plot2d("x^4+y^4", r=1.5, level=[0;1], color=blue, style="/") :
```

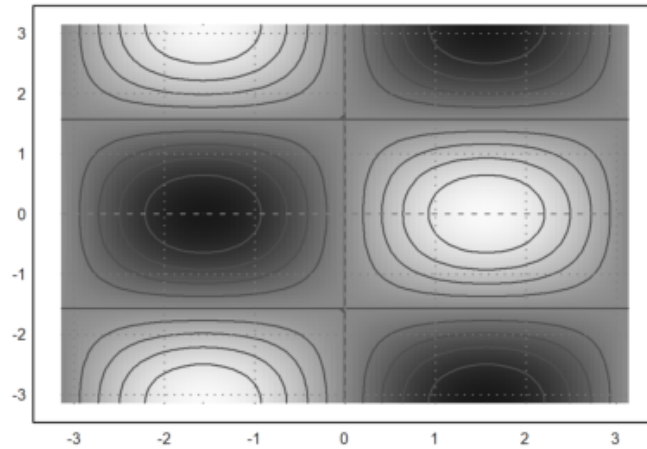
```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```



```
>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):
```

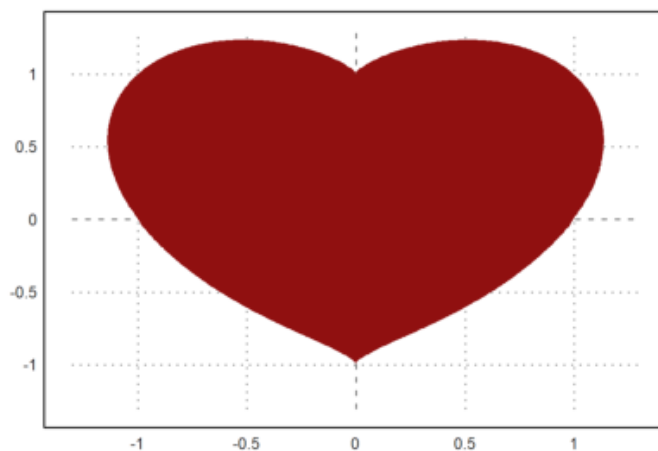


Juga dimungkinkan untuk menandai suatu wilayah

$$a \leq f(x, y) \leq b.$$

Hal ini dilakukan dengan menambahkan tingkat dengan dua baris.

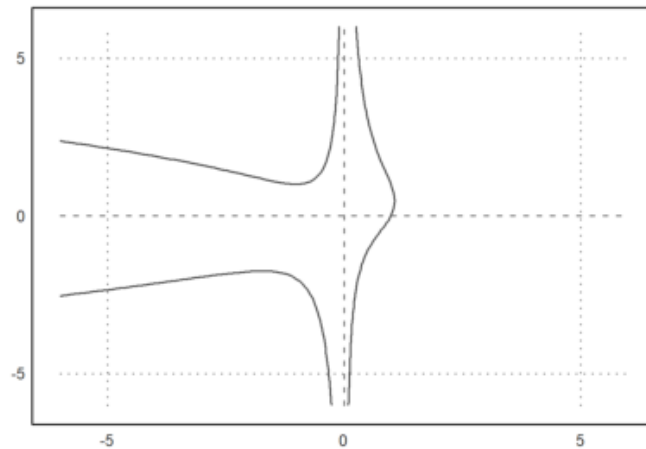
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
> style="#",color=red,<outline, ...
> level=[-2;0],n=100):
```



Dimungkinkan untuk menentukan tingkat tertentu. Misalnya, kita dapat menggambar solusi persamaan seperti

$$x^3 - xy + x^2y^2 = 6$$

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



```
>function starplot1 (v, style="/", color=green, lab=none) ...
```

```

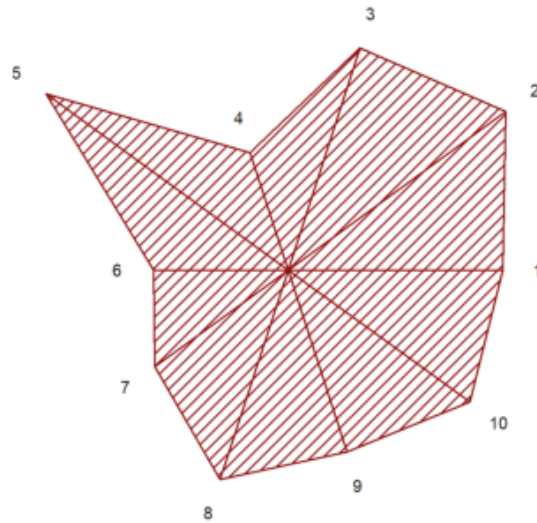
if !holding() then clg; endif;
w=window(); window(0,0,1024,1024);
h=holding(1);
r=max(abs(v))*1.2;
setplot(-r,r,-r,r);
n=cols(v); t=linspace(0,2pi,n);
v=v|v[1]; c=v*cos(t); s=v*sin(t);
cl=barcolor(color); st=barstyle(style);
loop 1 to n
  polygon([0,c[#],c[#+1]], [0,s[#],s[#+1]],1);
  if lab!=none then
    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 skala pada sumbu. Selain itu, kami menggunakan seluruh jendela untuk grafik.

Kami memanggil fungsi reset sebelum menguji grafik ini untuk mengembalikan pengaturan default grafik. Hal ini tidak diperlukan jika Anda yakin grafik Anda berfungsi dengan baik.

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



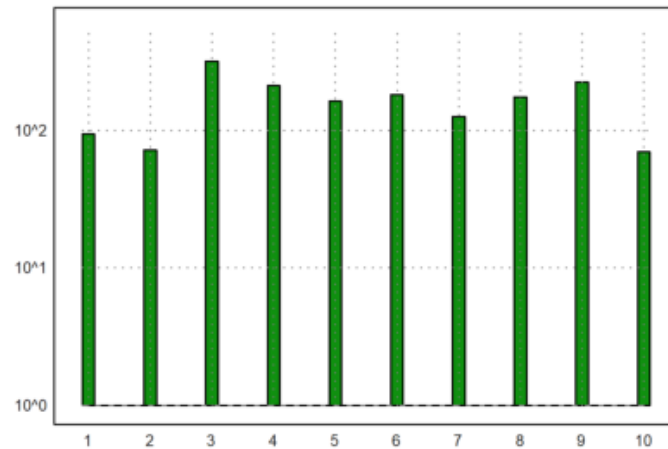
Terkadang, Anda mungkin ingin membuat grafik yang tidak dapat dilakukan oleh plot2d, tetapi hampir bisa. Dalam fungsi berikut, kita membuat grafik impuls logaritmik. plot2d dapat membuat grafik logaritmik, tetapi tidak untuk batang impuls.

```
>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-nilai yang terdistribusi secara eksponensial.

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



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

Fungsi `wait(0)` memaksa plot untuk muncul di jendela grafis. Jika tidak, proses redraw 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 tombol apa saja untuk menghentikan animasi ini.

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

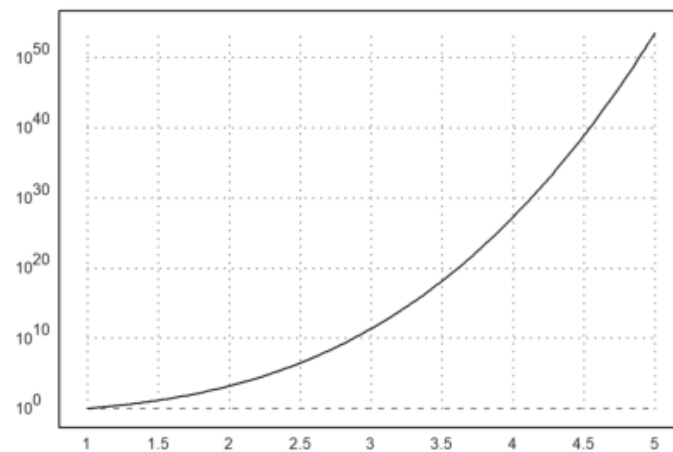
Grafik Logaritmik

EMT menggunakan parameter “logplot” untuk skala logaritmik.

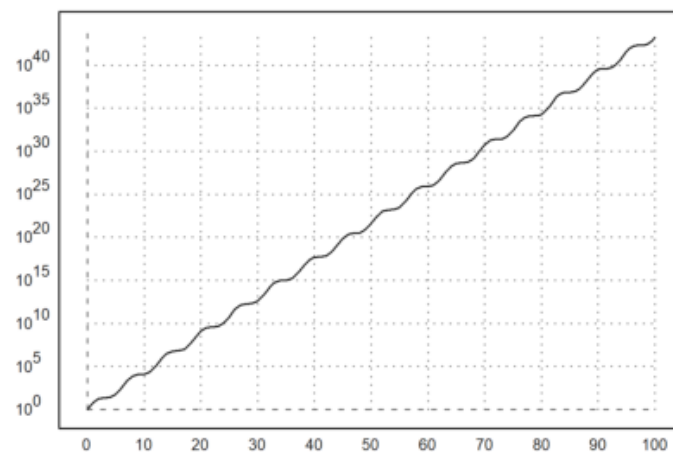
Grafik logaritmik dapat digambar menggunakan skala logaritmik pada sumbu y dengan `logplot=1`, atau menggunakan skala logaritmik pada sumbu x dan y dengan `logplot=2`, atau pada sumbu x dengan `logplot=3`.

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

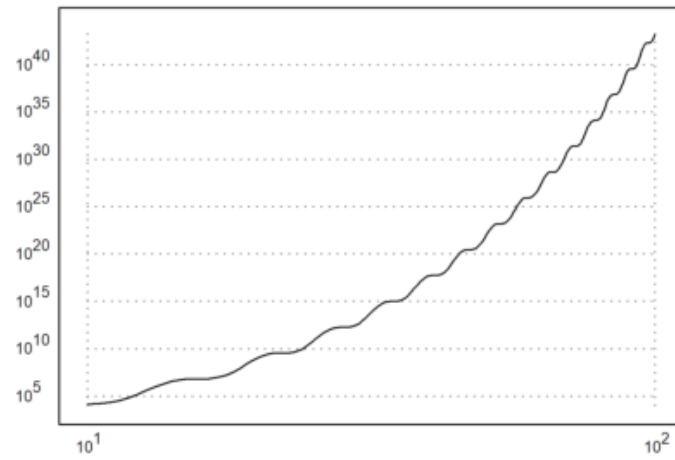
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



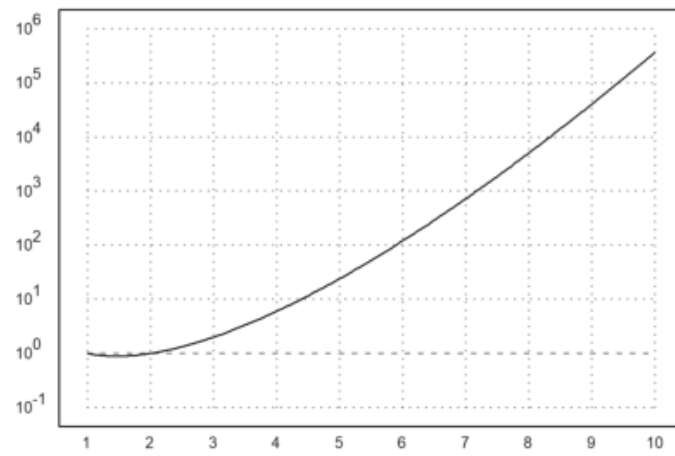
```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



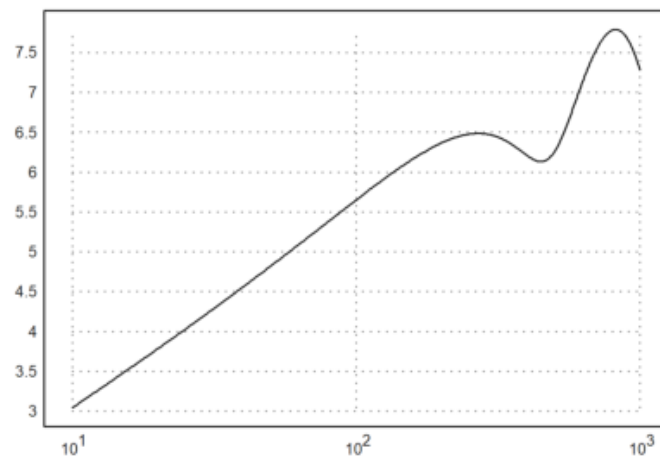
```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```



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

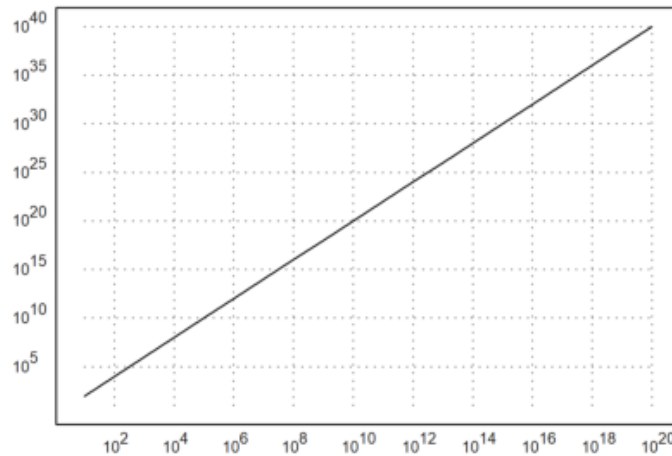


```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Hal ini juga berlaku untuk grafik data.

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



Rujukan Fungsi Lengkap Plot2D()

fungsi 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, ..
distribusi, merata, langkah, sendiri, adaptif, warna, tingkat, kontur, ..
nc, terisi, warna isi, garis luar, judul, xl, yl, peta, warna kontur, ..
lebar kontur, tanda skala, margin, pemotongan, cx, cy, insimg, spektral, ..
cgrid, vertikal, lebih kecil, dl, niveau, tingkat)

Fungsi plot serbaguna untuk grafik di bidang datar (grafik 2D). Fungsi ini dapat digunakan untuk menggambar grafik fungsi satu variabel, grafik data, kurva di bidang datar, grafik batang, kisi bilangan kompleks, dan grafik implisit 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 ditetapkan, maka a=cx-r, b=cx+r, c=cy-r, d=cy+r

r dapat 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 benar, coba pertahankan rentang x-y yang persegi

n : jumlah interval (default adaptif)

grid : 0 = tanpa grid dan label,

- 1 = sumbu saja,
- 2 = grid normal (lihat di bawah untuk jumlah garis grid)
- 3 = di dalam sumbu
- 4 = tanpa grid
- 5 = grid penuh termasuk margin
- 6 = tanda pada bingkai
- 7 = sumbu saja
- 8 = sumbu saja, sub-tanda

frame : 0 = tanpa bingkai

framecolor: Warna bingkai dan grid

margin : Nilai antara 0 dan 0,4 untuk margin di sekitar plot

color : Warna kurva. Jika ini adalah vektor warna,

maka akan digunakan untuk setiap baris matriks plot. Dalam

kasus

grafik titik, harus berupa vektor kolom. Jika vektor baris

atau

matriks warna penuh digunakan untuk grafik titik, maka

akan digunakan

untuk setiap titik data.

thickness : ketebalan garis untuk kurva

Nilai ini dapat lebih kecil dari 1 untuk garis yang sangat

tipis.

style : Gaya grafik untuk garis, penanda, dan pengisian.

```

Untuk titik gunakan
"[", "<>", ".", "..", "...",
"*", "+", "|", "-", 'o'
>[]#", "<>#", "o#" (bentuk terisi)
>[]w", "<>w", "ow" (tidak transparan)
Untuk garis gunakan
"--", "--", "-.", ".", "-.", "-.-", "->"
Untuk poligon terisi atau diagram batang gunakan
"#", "#o", 'o', "/", "\", "\/",
"+", "|", "-", "t"

```

titik : menggambar titik tunggal alih-alih segmen garis
 tambah_titik : jika benar, menggambar segmen garis dan titik
 tambah : menambahkan plot ke plot yang sudah ada
 pengguna : mengaktifkan interaksi pengguna untuk fungsi
 delta : ukuran langkah untuk interaksi pengguna
 batang : grafik batang (x adalah batas interval, y adalah nilai interval)
 histogram : menggambar frekuensi x dalam n subinterval
 distribution=n : menggambar distribusi x dengan n subinterval
 even : gunakan nilai antar untuk histogram otomatis.
 langkah : menggambar fungsi sebagai fungsi langkah (langkah=1,2)
 adaptif : gunakan grafik adaptif (n adalah jumlah langkah minimal)
 tingkat : menggambar garis tingkat fungsi implisit dua variabel
 garis batas : menggambar batas rentang tingkat.
 Jika nilai level berupa matriks $2 \times n$, rentang level akan digambar
 dengan warna menggunakan gaya pengisian yang diberikan. Jika outline benar,
 garis kontur akan digambar dengan warna kontur. Dengan fitur ini, daerah
 $f(x,y)$ antara batas dapat 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 sebelah kiri untuk label.
 vertical :

Mengaktifkan atau menonaktifkan label vertikal. Ini mengubah

variabel global

verticallabels secara lokal untuk satu plot. Nilai 1 hanya

menampilkan teks vertikal,

nilai 2 menggunakan label numerik vertikal pada sumbu y.

filled : mengisi plot kurva

fillcolor : warna pengisian untuk batang dan kurva terisi

outline : batas untuk poligon terisi

logplot : mengatur plot logaritmik

```
1 = plot logaritmik pada y,  
2 = plot logaritmik pada xy,  
3 = plot logaritmik pada x
```

sendiri :

Sebuah string yang mengacu pada rutin plot sendiri. Dengan >user,

Anda akan

mendapatkan interaksi pengguna yang sama seperti di plot2d. Rentang

akan

ditetapkan sebelum setiap panggilan ke fungsi Anda.

peta : ekspresi peta (0 lebih cepat), fungsi selalu dipetakan.

warna_kontur : warna garis kontur

lebar_kontur : lebar garis kontur

pemotongan : mengaktifkan/menonaktifkan pemotongan (default adalah true)

title :

Ini dapat digunakan untuk mendeskripsikan plot. Judul akan muncul di

atas

plot. Selain itu, label untuk sumbu x dan y dapat ditambahkan dengan `xl="string"` atau `yl="string"`. Label lain dapat ditambahkan dengan

fungsi

`label()` atau `labelbox()`. Judul dapat berupa string Unicode atau gambar rumus LaTeX.

cgrid :

Menentukan jumlah garis kisi untuk grafik kisi kompleks.

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

Ringkasan

Fungsi ini dapat menggambar

- ekspresi, kumpulan panggilan, atau fungsi satu variabel,
- kurva parametrik,
- data x terhadap data y ,
- fungsi implisit,
- diagram 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 tersebut harus berupa ekspresi dalam variabel x . Rentang harus didefinisikan dalam parameter a dan b kecuali rentang default ingin digunakan. Rentang y akan dihitung secara otomatis, kecuali c dan d ditentukan, atau jari-jari r , yang menghasilkan rentang r, r

untuk x dan y . Untuk plot fungsi, `plot2d` akan menggunakan evaluasi adaptif fungsi secara default. Untuk mempercepat grafik fungsi yang rumit, matikan opsi ini dengan `<adaptive`, dan opsionalnya kurangi jumlah interval n . Selain itu, `plot2d()` secara default akan menggunakan pemetaan. Artinya, ia akan menghitung grafik elemen per elemen. Jika ekspresi atau fungsi Anda dapat menangani vektor x , Anda dapat mematikan opsi ini dengan `<maps` untuk evaluasi yang lebih cepat.

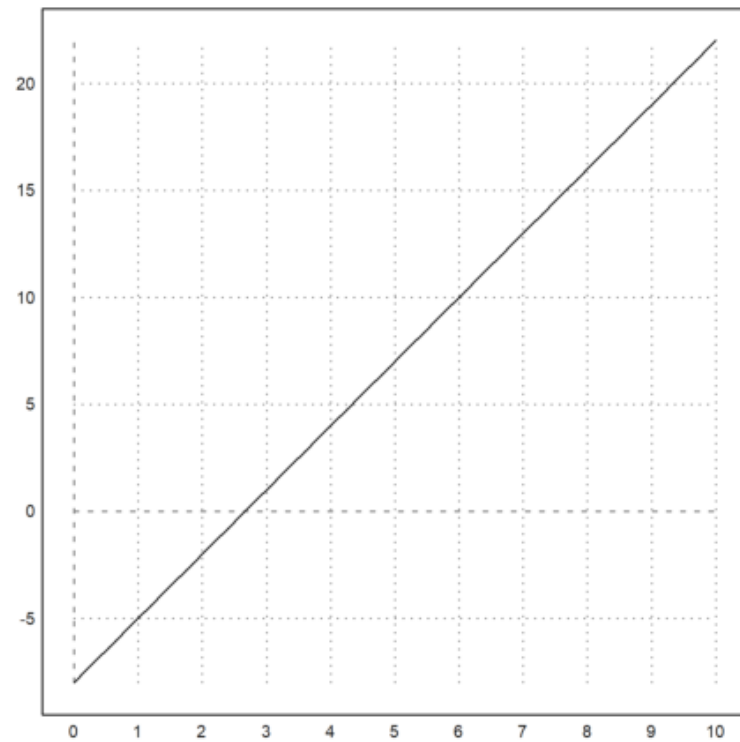
Perhatikan bahwa plot adaptif selalu dihitung elemen demi elemen.

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 terdapat dalam string harus selalu berupa ekspresi dalam variabel parameter x .

Latihan

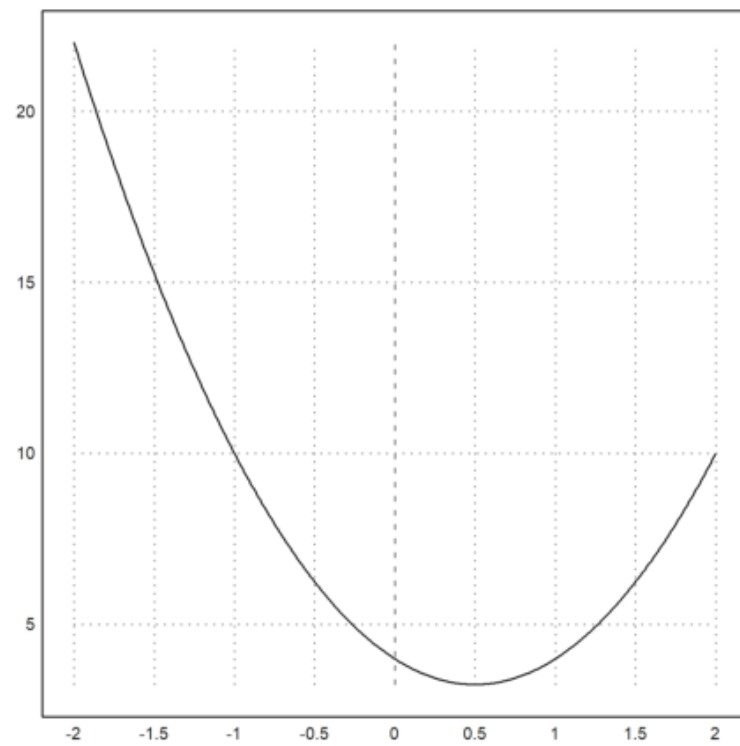
Plot 2D pada fungsi linear:

```
>plot2d("3x-8",0,10):
```



Plot 2D pada fungsi kuadrat:

```
>plot2d("3*x^2-3*x+4",-2,2):
```



Plot 2D pada fungsi trigonometri:

```
>plot2d("cos (x)", 0, 2*pi):
```

