IDENTITAS

Nama : Cindy Aulia Rosada
NIM : 24030130066
Prodi : S-1 Pendidikan Matematika A 2024

PENGGUNAAN SOFTWARE EMT UNTUK PLOT 2D

Plot Dasar

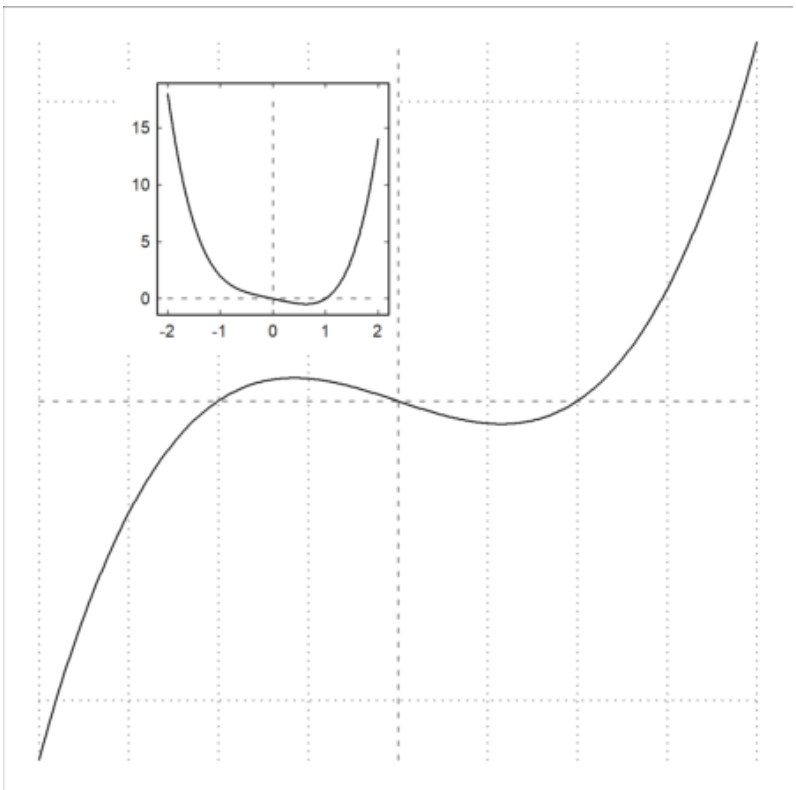
Ada beberapa fungsi dasar untuk menggambar plot. Yang pertama terdapat koordinat layar, yang selalu memiliki rentang dari 0 sampai 1024 pada setiap sumbu(x maupun y), meskipun layarnya tidak berbentuk persegi maupun kotak. Terdapat juga koordinat plot yang bisa diatur dengan setplot() misalnya ingin mengatur x=-2 sampai x=2 . Pemetaan antara koordinat tergantung pada jendela plot yang sedang aktif. Misalnya, fungsi default shrinkwindow() yang menyisakan ruang untuk label sumbu dan juga .
Contoh berikut ini akan digambarkan beberapa garis acak dengan warna yang berbeda.

```
>clg; // clear screen
>window(0,0,1024,1024); // gunakan seluruh jendela
>setplot(0,1,0,1); // atur koordinat plot
>hold on; // aktifkan mode tumpuk
>n=100; X=random(n,2); Y=random(n,2); // buat titik acak
>colors=rgb(random(n),random(n),random(n)); // buat warna acak
>loop 1 to n; color(colors[#]); plot(X[#],Y[#]); end; // plot
>hold off; // matikan mode tumpuk
>insimg; // masukkan gambar ke notebook
```



Untuk menggambar grafik baru, kita perlu menghapus grafik sebelumnya. Karena perintah plot() akan menghapus isi jendela plot, maka kita perlu menggunakan reset() untuk menghapus semua. Untuk menampilkan gambar hasil plot di layar notebook, perintah plot2d() dapat diakhiri dengan titik dua(:). Selain itu perintah plot2d () juga bisa diakhiri dengan titik koma(;), kemudian menggunakan insimg() untuk menampilkan gambar hasil plot.
Contoh lain, kita menggambar plot sebagai sisipan di plot lain. Ini dilakukan dengan mendefinisikan jendela plot yang lebih kecil. Perhatikan bahwa jenderal ini tidak menyediakan ruang untuk label sumbu di luar jendela plot. Kita harus menambahkan beberapa margin yang sesuai dengan kebutuhan. Perhatikan bahwa kita menyimpan dan memulihkan jendela penuh, dan selama membuat insert kita menggunakan hold agar grafik sebelumnya tidak hilang.

```
>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);
```

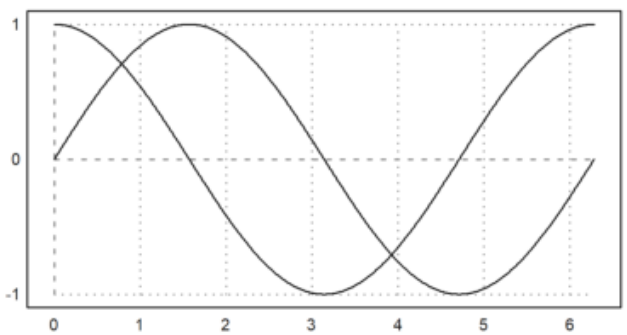
Plot dengan beberapa gambar dapat dibuat dengan cara yang sama. Untuk itu tersedia fungsi bantu yaitu figure().

Plot Aspect

Plot default menggunakan jendela grafik berbentuk persegi. Kita bisa mengubah ini dengan fungsi aspect(). Jangan lupa untuk mereset kembali aspek plot jika sudah selesai. Kita juga bisa mengubah pengaturan default di menu dengan "Set Aspect" ke rasio aspek tertentu, atau menyesuaikan dengan ukuran jendela grafik saat ini.
Namun, kita juga bisa mengubah rasio aspek hanya untuk satu plot. Dalam hal ini, ukuran area plot akan diubah, dan jendela akan diatur agar label tetap punya ruang yang cukup.

Contoh

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




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

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

Plot 2D di Euler

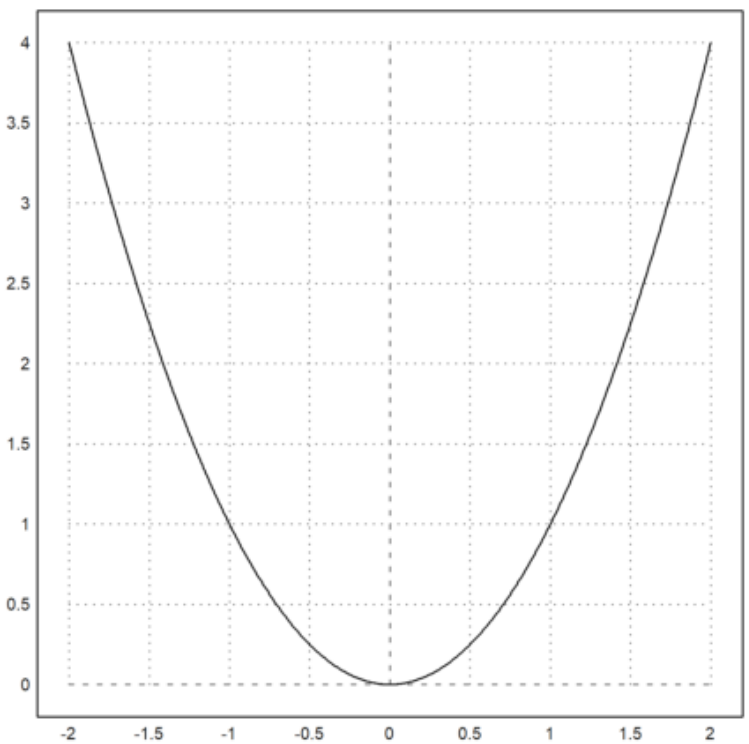
EMT Math Toolbox mempunyai fasilitas untuk membuat plot 2D, baik dari data maupun fungsi. EMT menggunakan perintah plot2d. Perintah ini bisa digunakan untuk 4 fungsi maupun data.
Selain itu, buat ini juga bisa dibuat di Maximal dengan Gnuplot atau di Python menggunakan Math Plot Lib.
Euler dapat memplok 2D dari:
-ekspresi: contohnya x^2+2*x+1
-fungsi, variabel, atau kurva parameter: misalnya f(x)= sin(x)
-vektor nilai(x,y): misalnya dari data pengukuran atau data buatan.
-scatter plot
-kurva implisit
-fungsi kompleks
Gaya plot yang tersedia meliputi berbagai bentuk, seperti garis dan, diagram batang, serta plot dengan arsiran.

Plot Ekspresi/ Variabel

Sebuah ekspresi sederhana dalam variabel x (misalnya "4*x^2) atau nama fungsi (misalnya "f") akan menghasilkan grafik fungsi. Contoh paling sederhana adalah menggunakan range bawaan(nilai x) dan secara otomatis program akan menyesuaikan nilai pada sumbu y supaya grafik terlihat sesuai.
Catatan: jika perintah diakhiri dengan tanda titik dua ":", grafik akan langsung ditampilkan di jendela teks. Jika tidak ada titik dua, tekan tombol TAB untuk menampilkan grafik jika jendela plot tertutup.

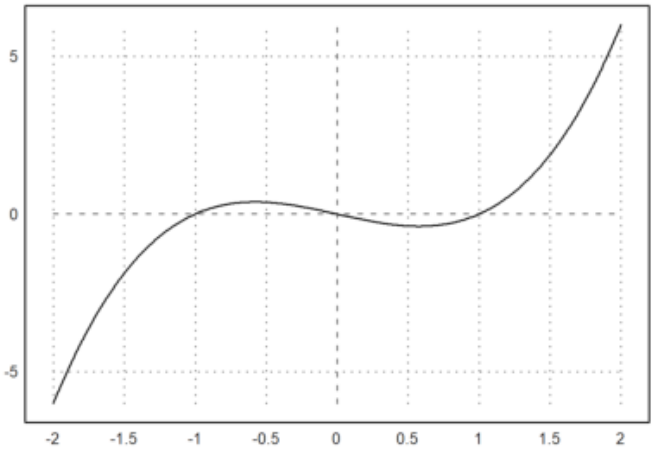
Plot fungsi kuadrat f(x)= x2

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



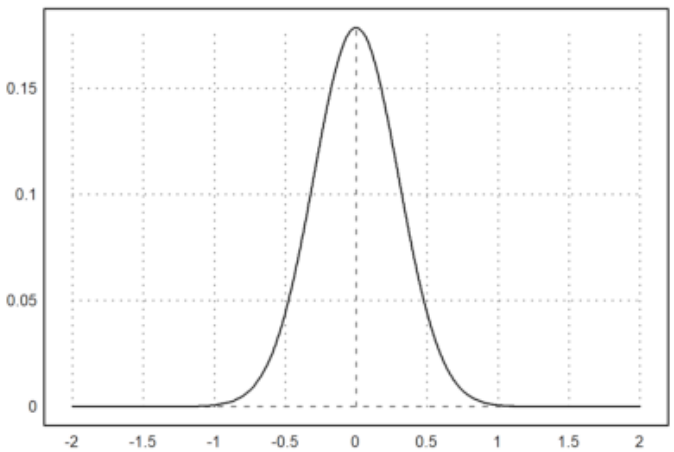
f(x)=x3 ? xdenganrasio1.5

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



Plot fungsi eksponensial f(x)=exp(-a*x2)/a

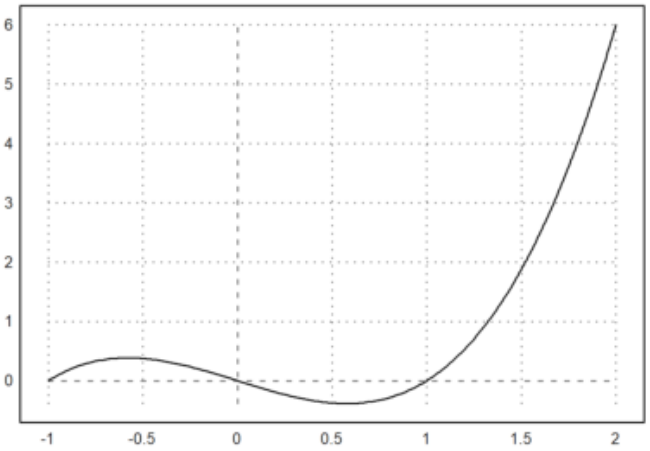
```
>a:=5.6; plot2d("exp(-a*x^2)/a"); insimg(30); // menampilkan gambar hasil plot setinggi 25
```



Dari beberapa contoh sebelumnya kita dapat melihat bahwa aslinya gambar plot menggunakan sumbu x dengan menentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai x dan y, kita dapat menambahkan nilai-nilai batas x dan y di belakang ekspresi yang digambar
Rentang grafik bisa diatur dengan parameter berikut:
-a,b: batas nilai x (default -2,2)
-c,d: batas nilai y (default: skala dengan nilai)
-r: sebagai alternatif, bisa untuk menentukan radius dari titik pusat grafik.
-cx, cy: koordinat titik pusat grafik (default 0,0)

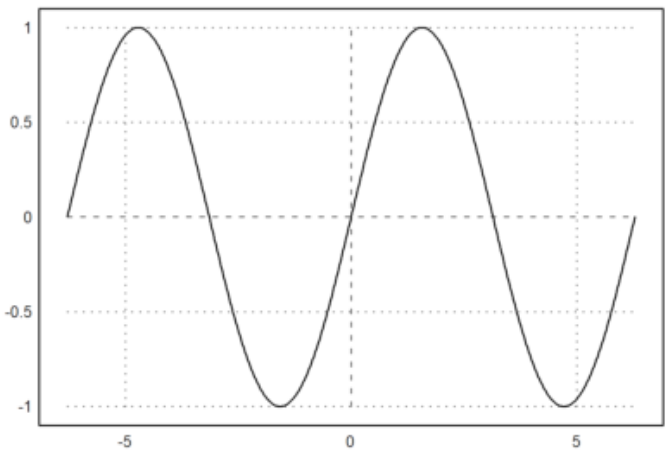
Grafik fungsi f(x)=x3 - xdenganrentangnilai - 1sampaidengan2

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



Grafik fungsi sinus pada interval [-2pi,2pi]

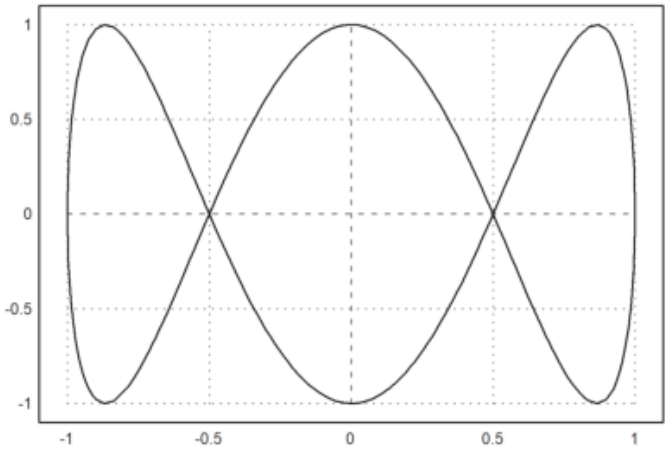
```
>plot2d("sin(x)",-2*pi,2*pi): // plot sin(x) pada interval [-2pi,2pi]
```



Grafik fungsi sinus dan cosinus

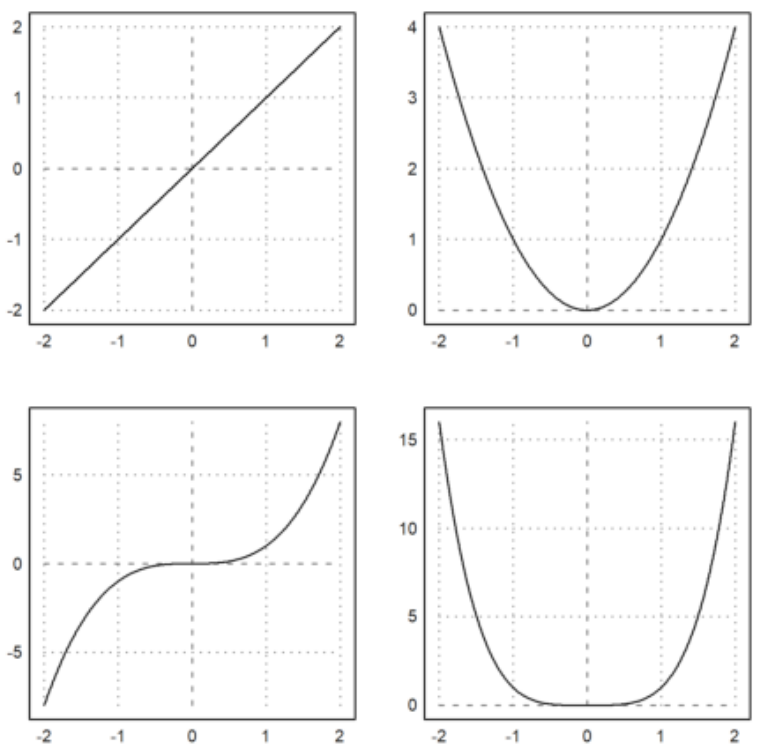
dengan frekuensi berbeda

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



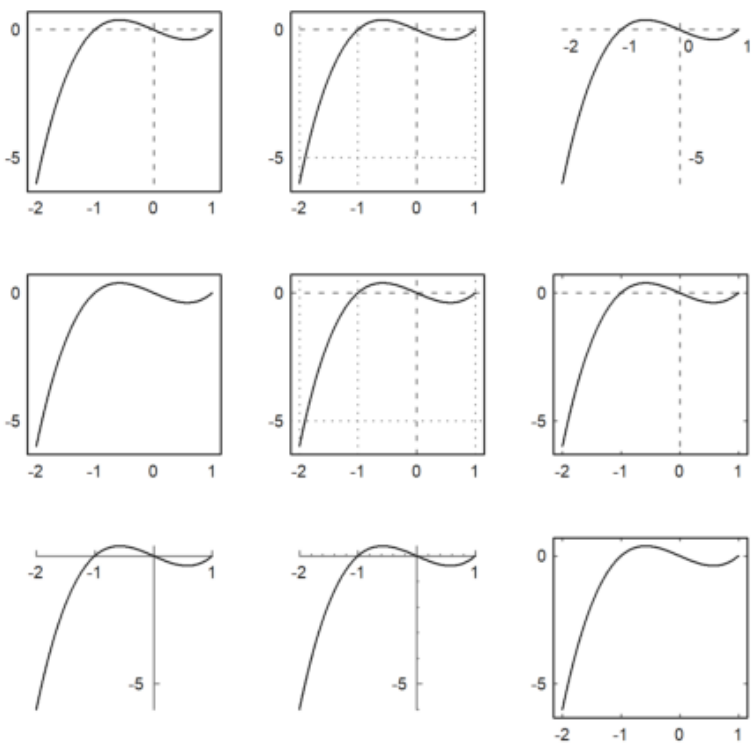
Selain menggunakan tanda titik dua (:), Kita juga bisa menggunakan perintah insimg(baris) untuk menampilkan grafik. Grafik akan muncul di jendela teks dengan ukuran sesuai jumlah baris yang kita tentukan.
Dalam pengaturan grafik bisa ditampilkan:
-di jendela terpisah yang bisa diubah ukurannya
-langsung di jendela notebook
Jika ingin tampilan lebih bervariasi, bisa menggunakan perintah plot khusus lainnya. Jika grafik tidak muncul bisa tekan tombol TAB supaya terlihat. Untuk membagi jendela menjadi beberapa bagian grafik, gunakan perintah figure(). Contohnya grafik fungsi x^1 hingga x^4 bisa ditampilkan dalam 4 bagian dalam satu jendela. Gunakan figure(0) jika ingin kembali ke tampilan awal(default).

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



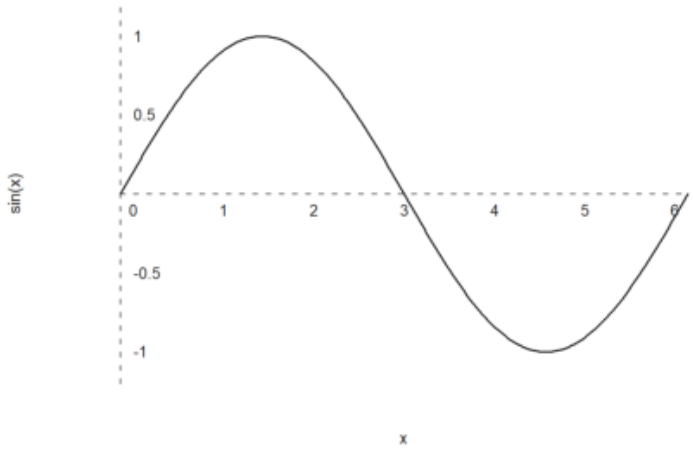
Dalam plot2d(), ada beberapa gaya tampilan alternatif yang bisa dipilih dengan menambahkan grid=x. Sebagai gambaran, kita bisa menampilkan berbagai gaya grid dalam satu jendela (menggunakan figure()). Gaya grid=0 tidak ditampilkan, karena gaya tersebut berarti tanpa grid dan tanpa bingkai.

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

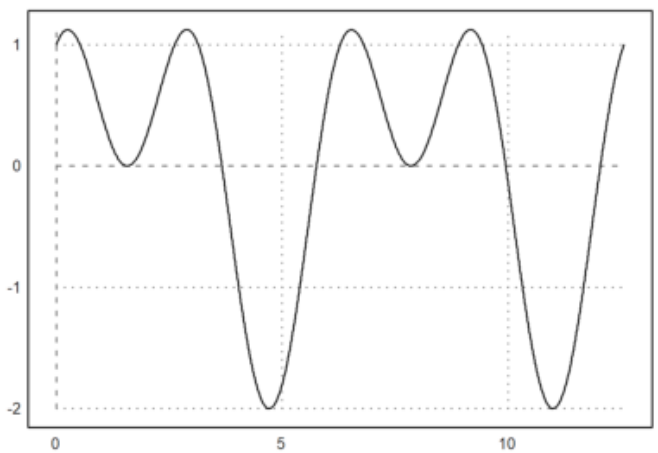


Jika argumen pada plot2d() berupa sebuah ekspresi yang diikuti oleh 4 angka, maka angka-angka tersebut adalah batas rentang x dan y untuk grafik.
Sebagai alternatif, nilai a, b,c,d bisa ditentukan terlebih dahulu sebagai parameter misalnya a=...
Pada contoh ini, kita mengubah gaya grid, menambahkan label pada sumbu, serta 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)"):
```

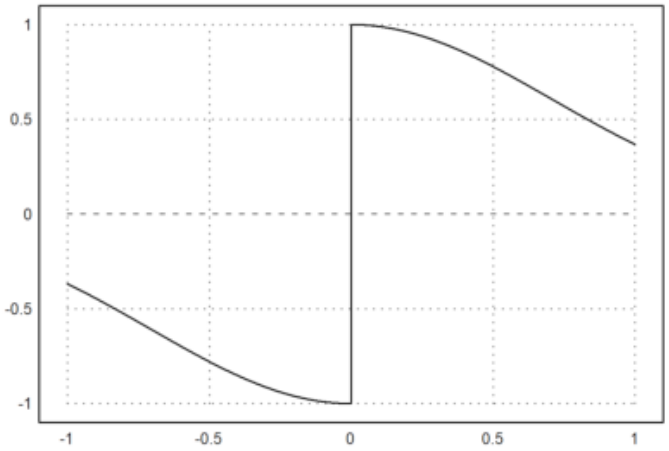



```
>plot2d("sin(x)+cos(2*x)",0,4pi):
```

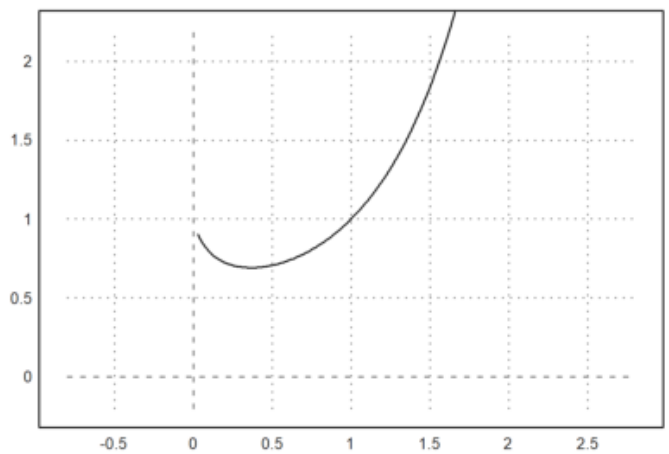


Gambar yang dihasilkan ketika grafik dimasukkan ke jendela teks akan otomatis disimpan di folder yang sama dengan notebook, biasanya dalam sub folder bernama"images". Gambar-gambar ini juga bisa dipakai ketika melakukan ekspor ke HTML.
Kita bisa menyalin gambar dengan mudah: cukup pilih gambarnya, lalu tekan Ctrl+C untuk menyalin ke clipboard. Selain itu grafik juga bisa diekspor lewat menu File.
Fungsi atau ekspresi dalam plot2d hitung secara adaptif. Jika ingin lebih cepat, bisa dimatikan mode adaptif dengan <adaptive dan menentukan jumlah interval perhitungan menggunakan n=... Namun hal tersebut biasanya jarang diperlukan.

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

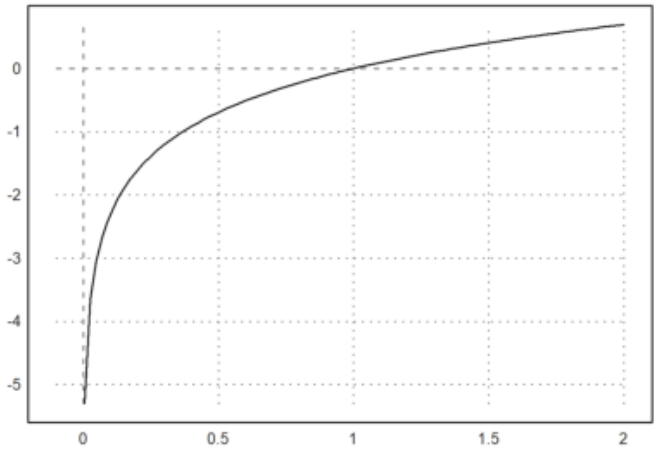



```
>plot2d("x^x",r=1.2,cx=1,cy=1):
```



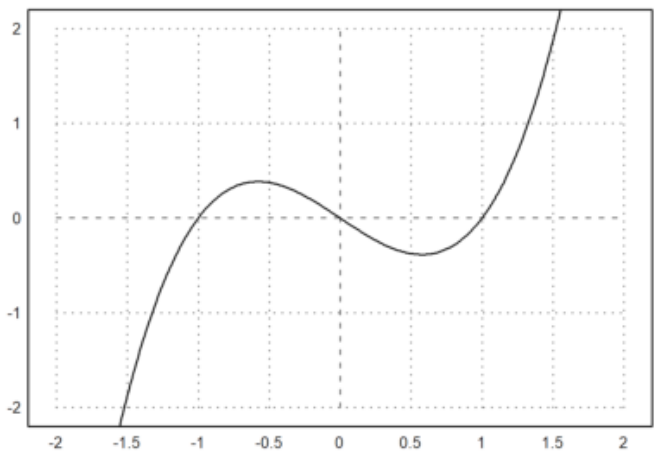
Perhatikan bahwa x^x tidak terdefinisi untuk x=0. Fungsi plot2d akan mengenali kesalahan ini, dan mulai menggambar grafik hanya ketika fungsi tersebut sudah terdefinisi. Hal ini berlaku juga untuk semua fungsi yang menghasilkan nilai NAN jika berada di luar daerah definisinya.

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

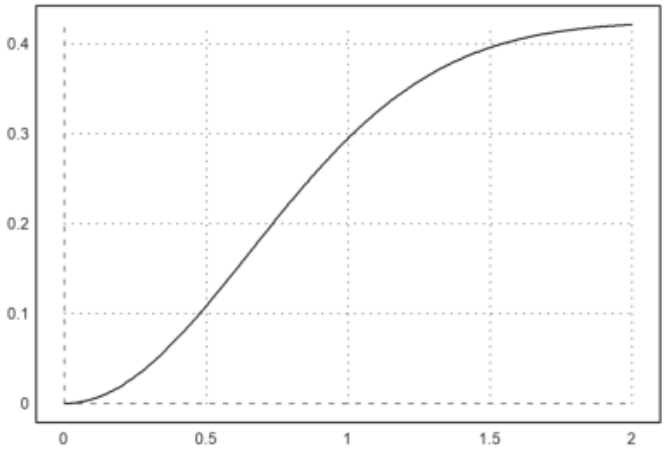


Parameter square=true (atau >square) akan secara otomatis memilih rentang sumbu y sehingga hasilnya berubah jendela grafik berbentuk persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot

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

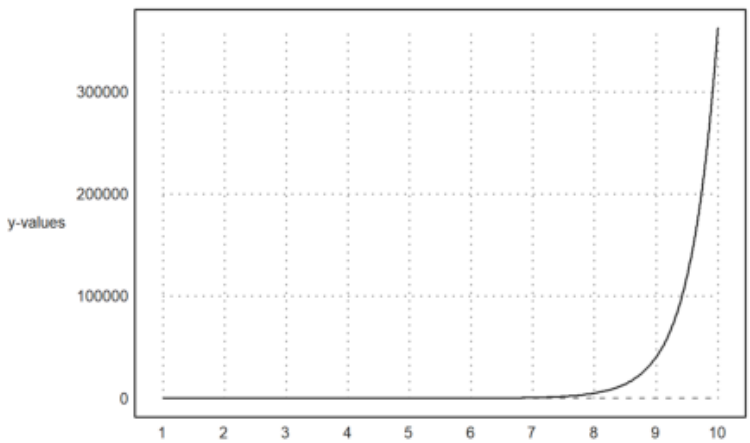



```
>plot2d(''integrate("sin(x)*exp(-x^2)",0,x)'',0,2): // plot integral
```



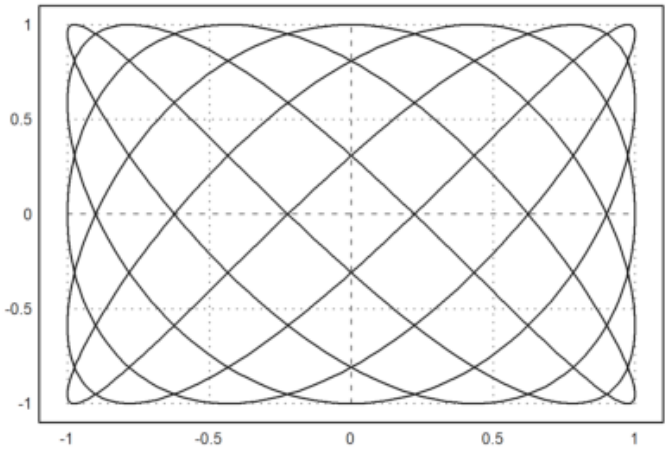
Jika membutuhkan lebih banyak ruang untuk label sumbu y, gunakan perintah shrinkwindow() dengan parameter smaller, atau berikan nilai positif pada opsi smaller di plot2d().

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

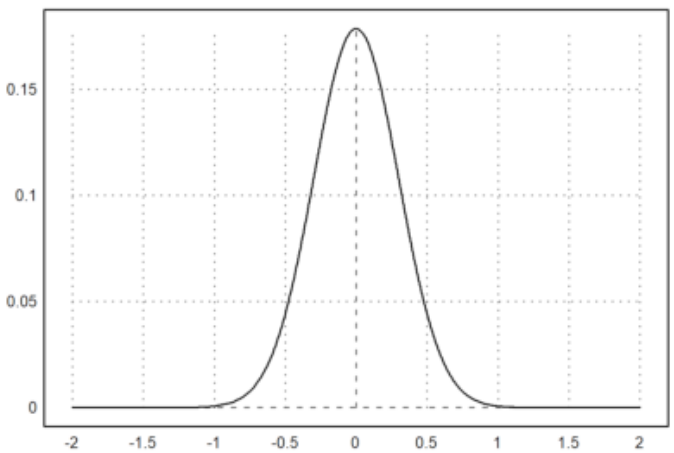


Eskpresi simbolik juga bisa digunakan, karena disimpan sebagai ekspresi string sederhana.

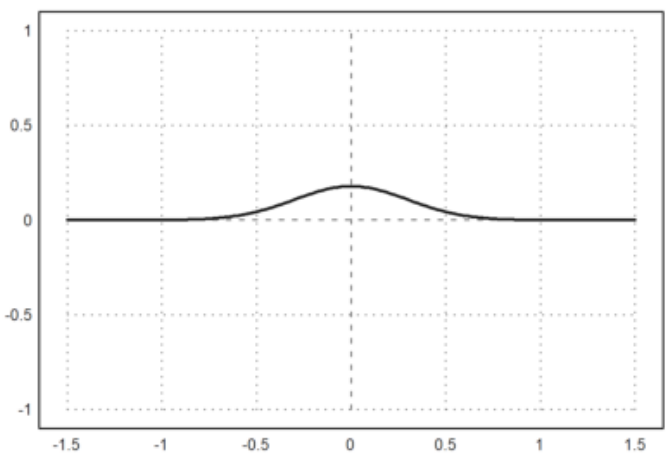
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



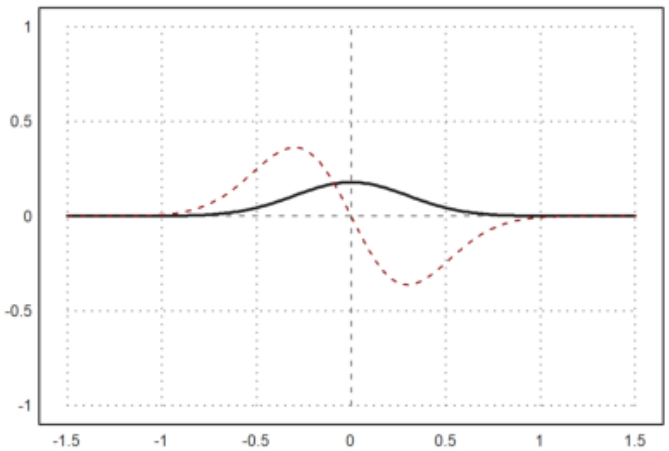

```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression
>plot2d(expr,-2,2): // plot from -2 to 2
```



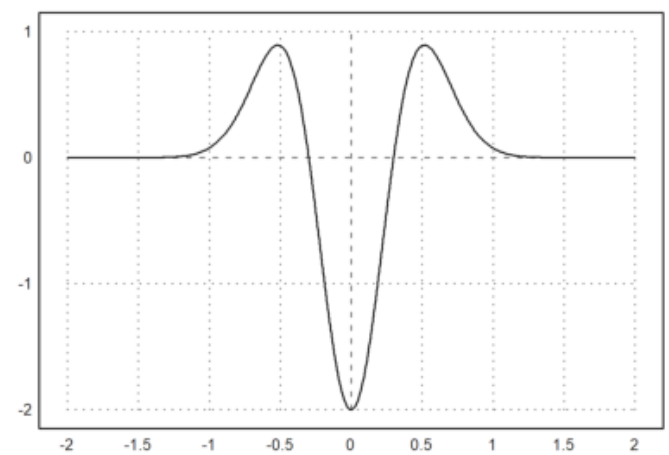

```
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)
```



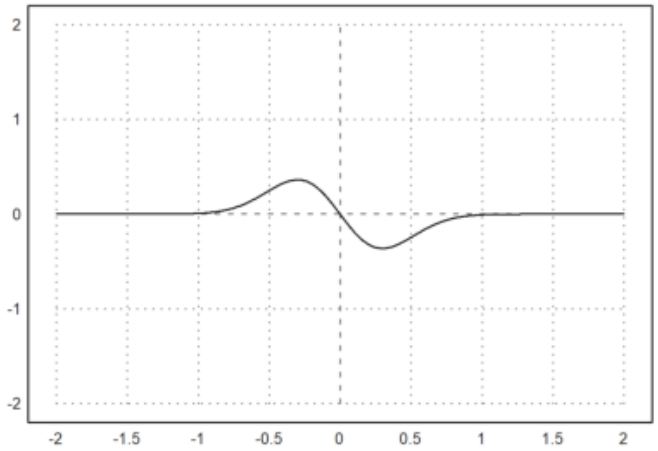
```
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot
```



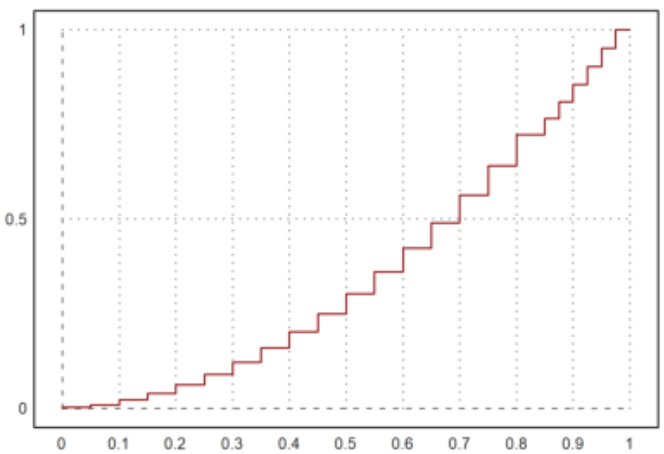

```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle
```



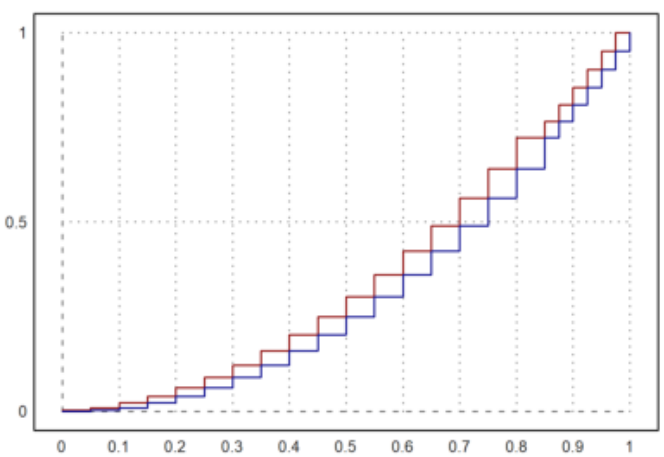

```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square
```




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



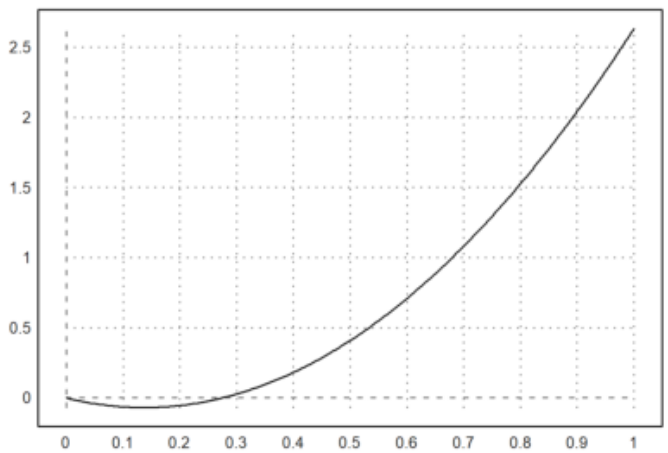

```
>plot2d("x^2",>add,steps=2,color=blue,n=10):
```



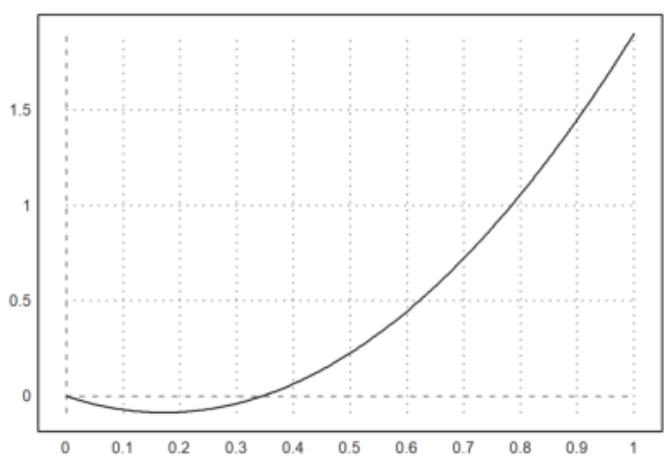
Fungsi dengan Satu Parameter

Fungsi plot yang paling penting untuk grafik bidang datar adalah plot2d(). Fungsi ini diimplementasikan dalam bahasa Euler pada file "plot.e", yang dimuat saat awal program dijalankan.
Berikut beberapa contoh penggunaan fungsi. Seperti biasa dalam EMT, fungsi yang bekerja untuk fungsi lain atau ekspresi dapat diberikan parameter tambahan (selain x), yang bukan merupakan variabel global, ke dalam fungsi dengan parameter titik koma atau dengan call collection.

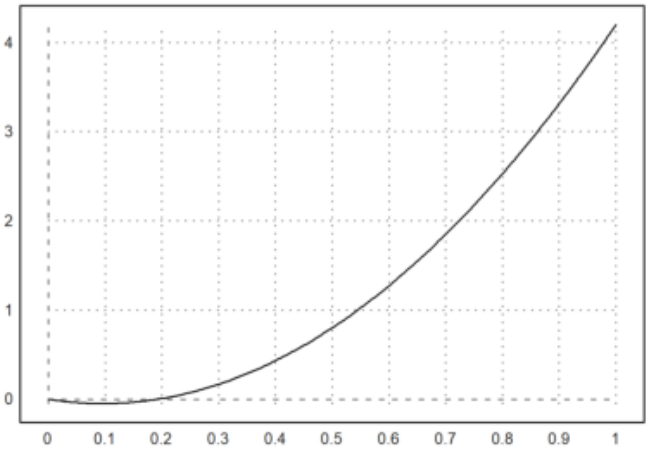
```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
>a=0.3; plot2d("f",0,1;a): // plot with a=0.3
```



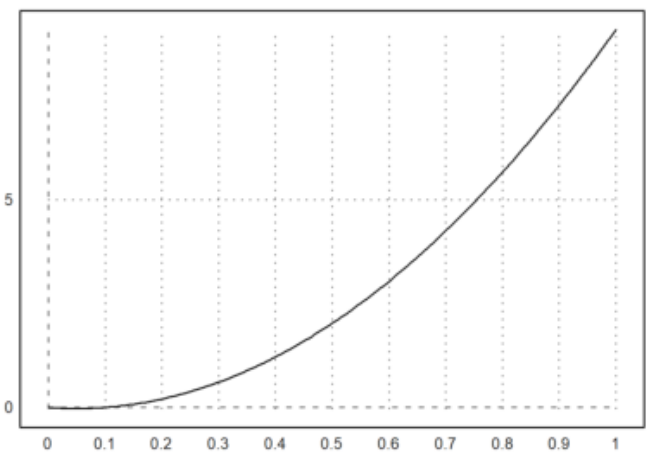

```
>plot2d("f",0,1;0.4): // plot with a=0.4
```




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




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

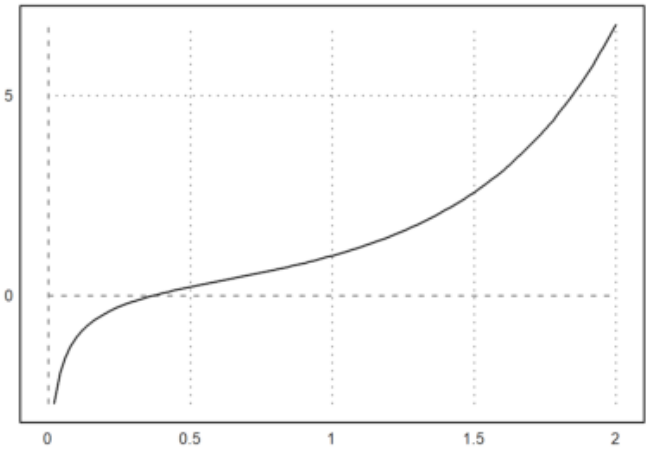


Berikut adalah ringkasan fungsi yang diterima:
-ekspresi simbolik dalam x
-fungsi simbolik dengan nama seperti "f"
-fungsi simbolik hanya dengan menyebutkan nama f
Fungsi plot2d() juga menerima fungsi simbolik. Untuk fungsi simbolik, cukup menggunakan namanya saja.

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

$$x^{x} \left(\log(x) + 1\right)$$

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

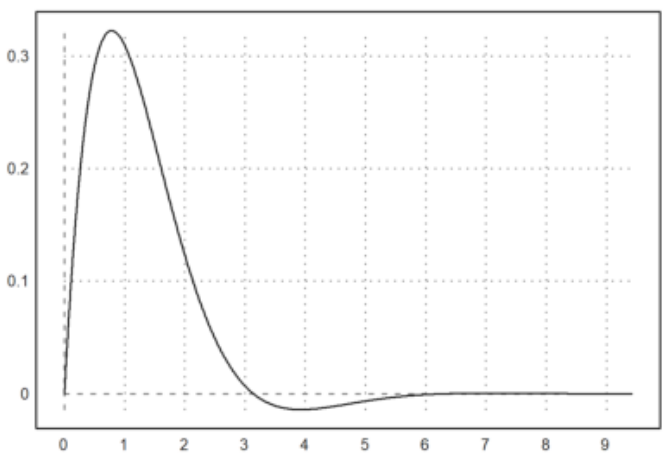


Tentu saja, untuk ekspresi atau ekspresi simbolik, nama variabel saja sudah cukup untuk menggambarkannya.

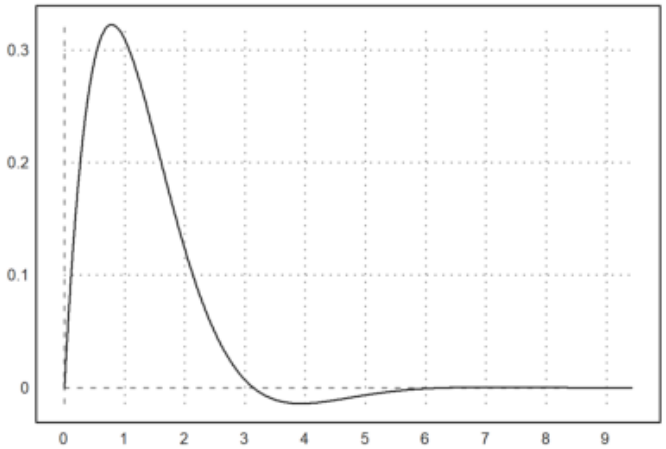
```
>expr &= sin(x)*exp(-x)
```

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

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

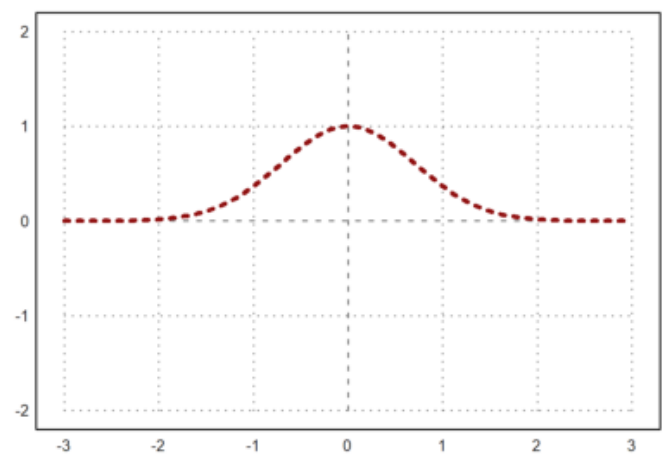



```
>function f(x) &= x^x;
>plot2d(f,r=1,cx=1,cy=1,color=blue,thickness=2);
>plot2d(&diff(f(x),x),>add,color=red,style="-.-"):
```



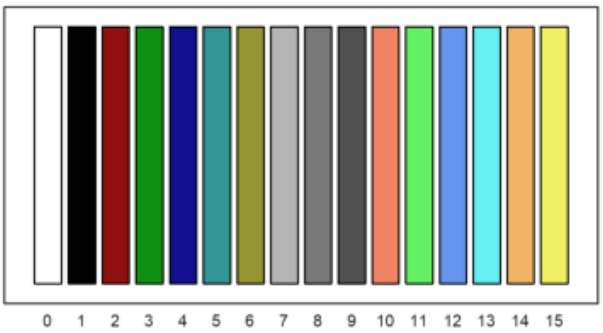
Untuk gaya garis terdapat berbagai pilihan.
-style="...". Pilihan: "-", "-", "-.", ".", ".-.", "-.-".
-color: lihat bagian di bawah untuk warna.
-thickness: default bernilai 1.
Warna dapat dipilih dari salah satu warna bawaan, atau menggunakan warna RGB.
-0..15: indeks warna bawaan.
-konstanta warna: white, black, red, green, blue, cyan, olive, lightgray, gray, darkgray, orange, lightgreen, turquoise, lightblue, lightorange, yellow
-rgb(red,green,blue): parameter berupa bilangan riil dalam [0,1].

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



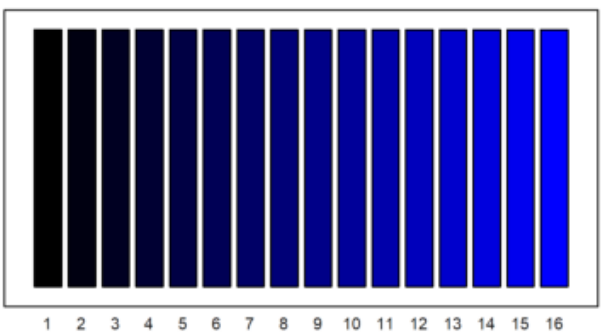
Berikut ini adalah tampilan warna-warna bawaan EMT.

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



Tetapi Anda dapat menggunakan warna apa pun.

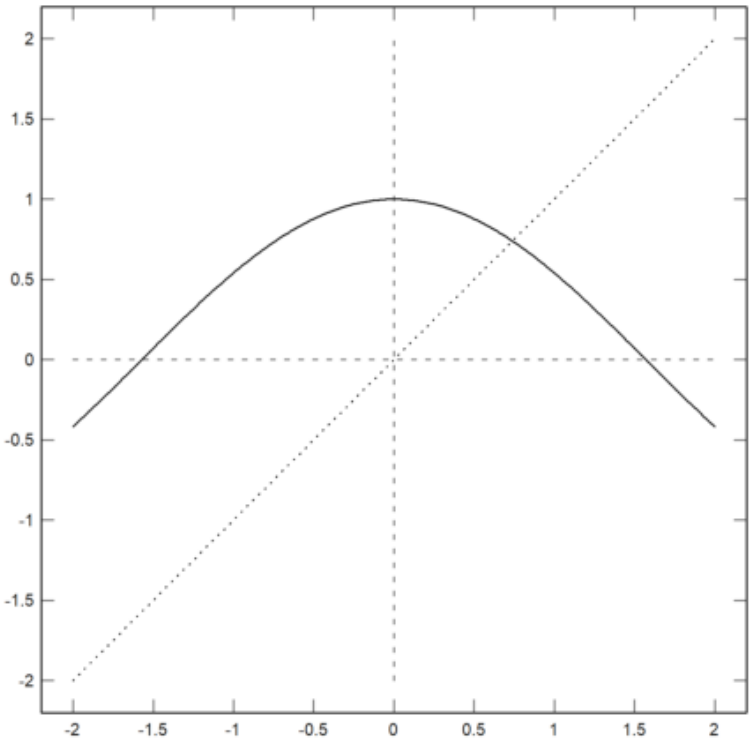
```
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```



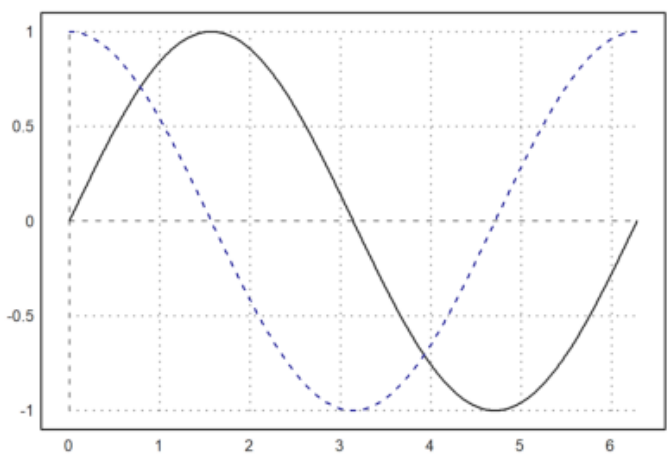
Menggambar Beberapa Kurva pada Bidang Koordinat yang Sama

Untuk memplot lebih dari satu fungsi (banyak fungsi) dalam satu jendela dapat dilakukan dengan beberapa cara. Salah satu metode adalah dengan menggunakan >add pada beberapa pemanggilan plot2d, kecuali pada pemanggilan pertama. Fitur ini sudah kita gunakan pada contoh-contoh sebelumnya.

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

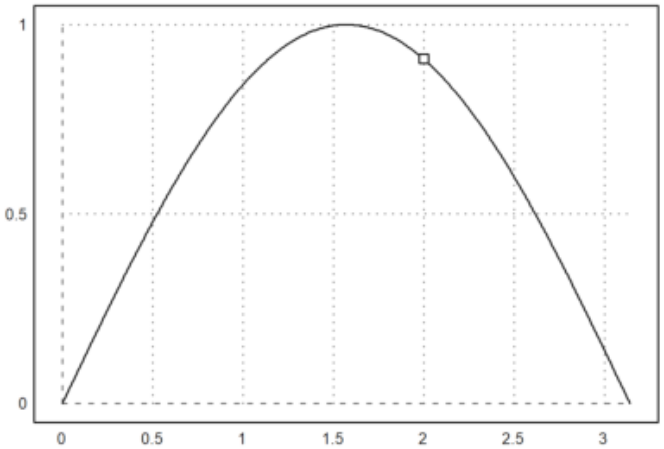



```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



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

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

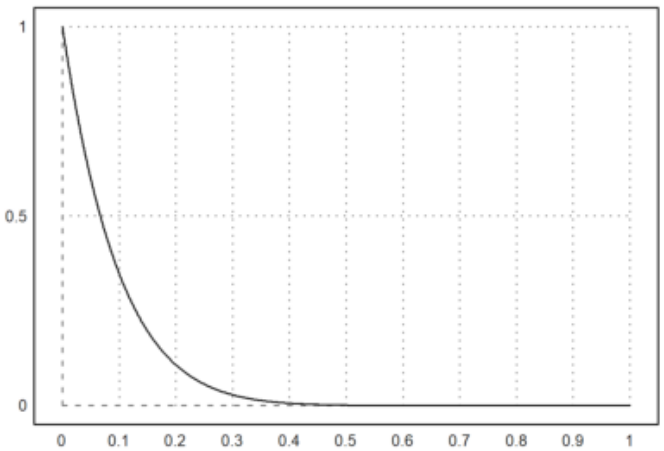


Kita menambahkan titik potong dengan sebuah label (pada posisi "cl" untuk center left), dan menyisipkan hasilnya ke dalam notebook. Kita juga menambahkan judul pada plot tersebut.

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

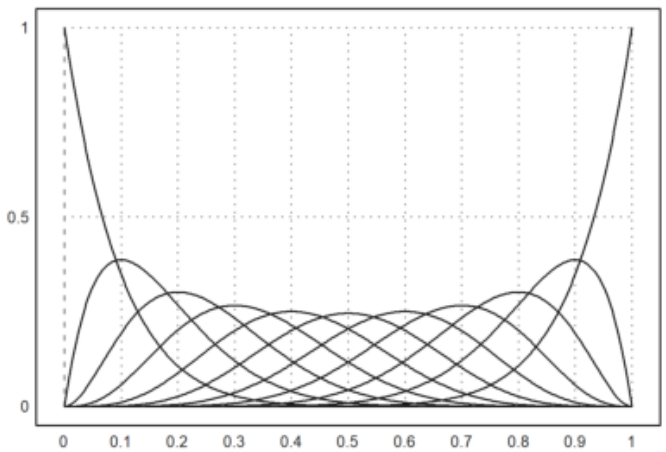
$$\frac{x^4}{120}-\frac{x^2}{6}+1$$

```
>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;
```



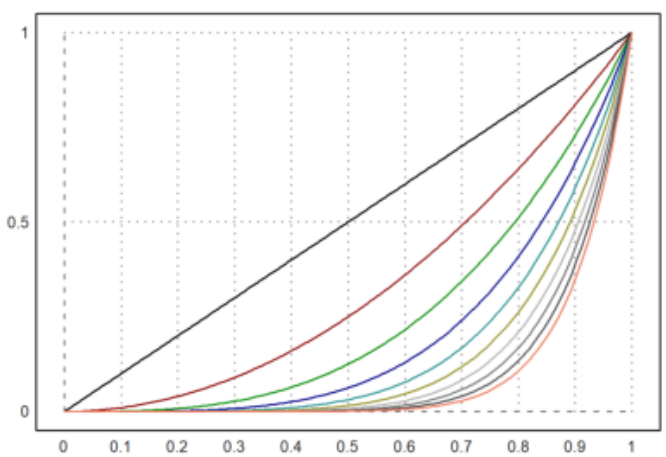
Metode kedua adalah dengan menggunakan sepasang matriks berupa nilai-nilai x dan nilai-nilai y dengan ukuran yang sama.
Kita menghasilkan sebuah matriks nilai dengan satu Polinomial Bernstein pada setiap baris. Untuk itu, kita cukup menggunakan sebuah vektor kolom dari i. Silakan lihat bagian pengantar tentang bahasa matriks untuk mempelajari lebih detail.

```
>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):
```



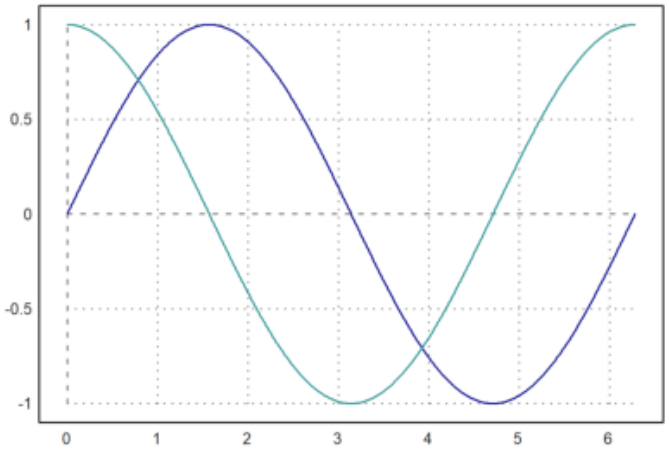
Perlu dicatat bahwa parameter color dapat berupa sebuah vektor. Dengan demikian, setiap warna akan digunakan untuk setiap baris dari matriks.

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

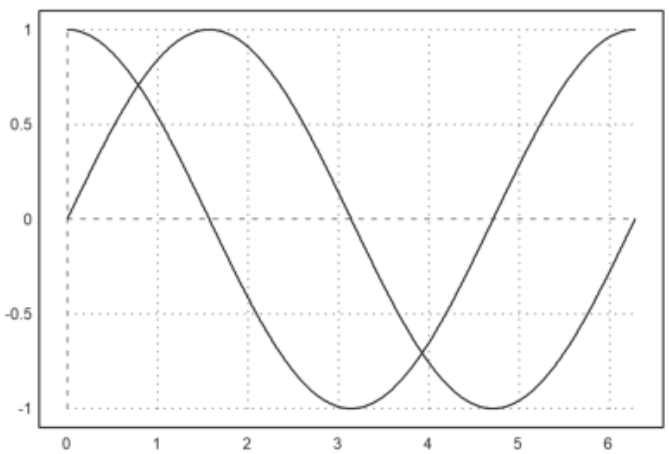


Metode lain adalah dengan menggunakan sebuah vektor berisi ekspresi (string). Dengan cara ini, Anda dapat menggunakan array warna, array gaya, dan array ketebalan dengan panjang yang sama.

```
>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 seperti itu dari Maxima dengan menggunakan makelist() dan mxm2str().

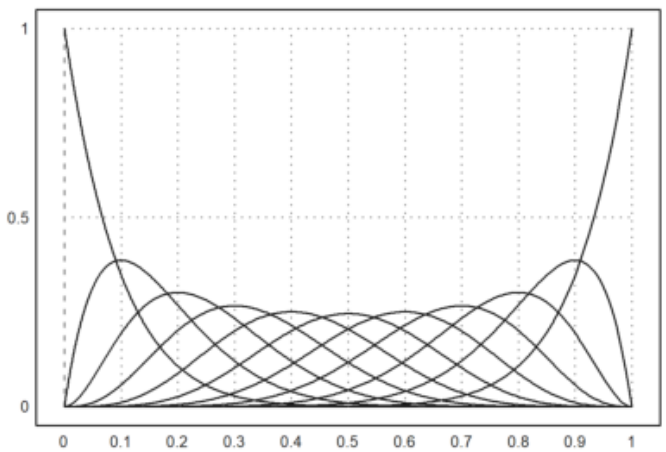
```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // make list
```

$$\left[(1-x)^{10},\ 10\,(1-x)^{9}\,x,\ 45\,(1-x)^{8}\,x^{2},\ 120\,(1-x)^{7}\,x^{3},\ 210\,(1-x)^{6}\,x^{4},\ 252\,(1-x)^{5}\,x^{5},\ 210\,(1-x)^{4}\,x^{6},\ 120\,(1-x)^{3}\,x^{7},\ 45\,(1-x)^{2}\,x^{8},\ 10\,(1-x)\,x^{9},\ x^{10}\right]$$

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

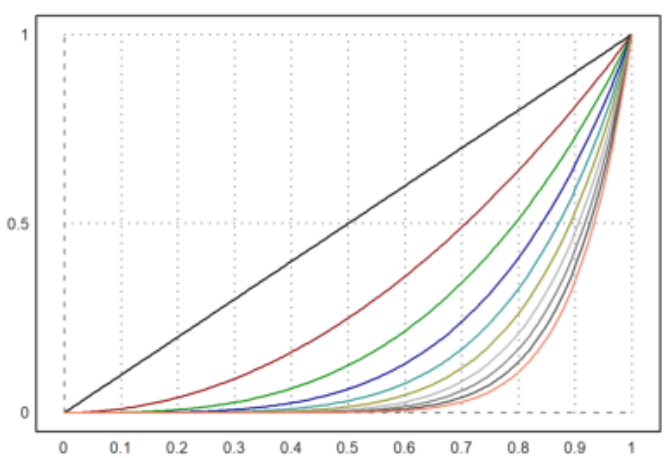
```
(1-x)^10
10*(1-x)^9*x
45*(1-x)^8*x^2
120*(1-x)^7*x^3
210*(1-x)^6*x^4
252*(1-x)^5*x^5
210*(1-x)^4*x^6
120*(1-x)^3*x^7
45*(1-x)^2*x^8
10*(1-x)*x^9
x^10
```

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



Alternatif lain adalah menggunakan bahasa matriks dari Euler.
Jika suatu ekspresi menghasilkan sebuah matriks fungsi, dengan satu fungsi pada setiap baris, maka semua fungsi tersebut akan dipetakan ke dalam satu plot.
Untuk itu, gunakan vektor parameter dalam bentuk vektor kolom. Jika ditambahkan array warna, maka array tersebut akan digunakan untuk setiap baris pada plot.

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

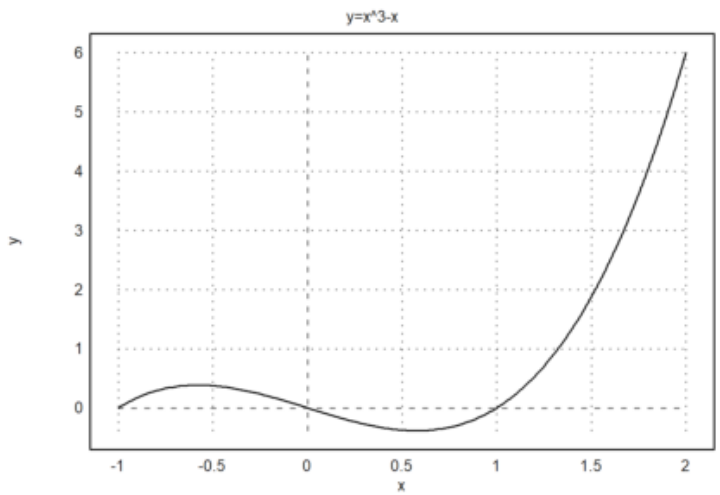


Ekspresi dan fungsi satu baris dapat melihat variabel global.
Jika Anda tidak dapat menggunakan variabel global, Anda perlu menggunakan fungsi dengan parameter tambahan, dan meneruskan parameter tersebut sebagai parameter titik koma.
Perhatikan untuk meletakkan semua parameter yang diberikan di bagian akhir perintah plot2d. Pada contoh, kita meneruskan a=5 ke fungsi f, yang kemudian kita plot dari -10 hingga 10.

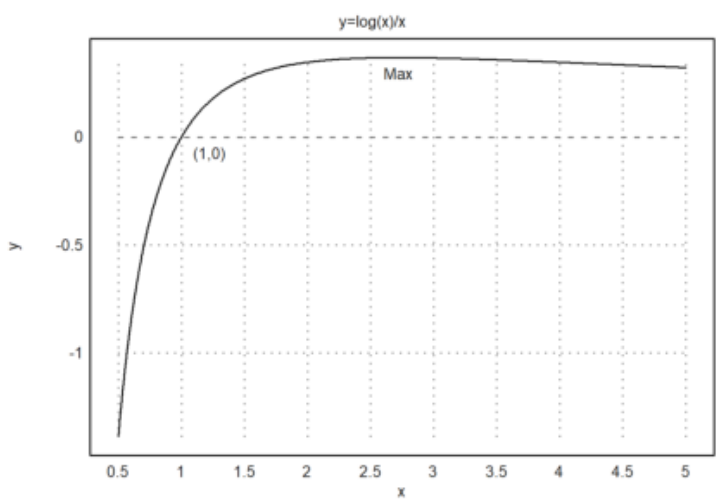
Label Teks

Dekorasi dalam grafik bisa berupa:
judul menggunakan title="..."
label sumbu x dan y dengan xl="..." serta yl="..." label teks lain memakai label("...",x,y)
Perintah label akan menambahkan teks ke grafik pada titik koordinat (x,y). Selain itu, perintah ini juga dapat diberi tambahan argumen untuk menentukan posisinya.

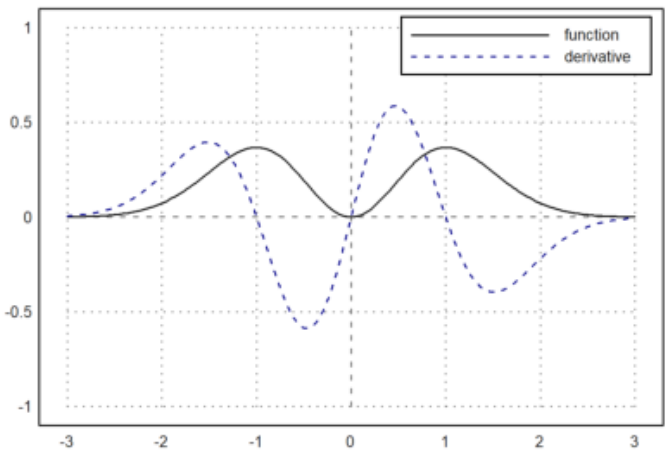
```
>plot2d("x^3-x",-1,2,title="y=x^3-x",yl="y",xl="x"):
```




```
>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"):
```



```
>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):
```

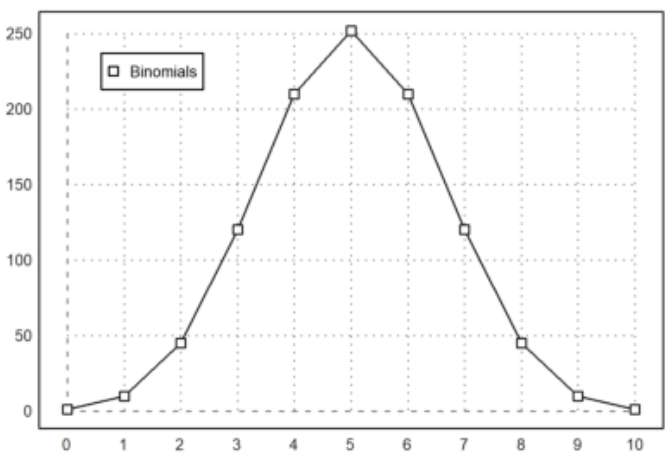


Secara bawaan, kotak label ditempatkan di bagian kanan atas, namun dengan opsi >left, posisinya akan berpindah ke kiri atas. Kotak ini bisa dipindahkan ke lokasi lain sesuai kebutuhan. Titik acuannya berada di sudut kanan atas kotak, sedangkan nilai koordinat ditentukan sebagai pecahan dari ukuran jendela grafik. Lebar kotak akan menyesuaikan secara otomatis.

Pada grafik berupa titik, kotak label tetap dapat digunakan. Untuk itu, tambahkan parameter >points atau gunakan vektor penanda, masing-masing sesuai dengan jumlah label yang ada.

Pada contoh berikut hanya digunakan satu fungsi, sehingga cukup memakai string biasa, bukan vektor string. Warna teks diatur menjadi hitam untuk ilustrasi ini.

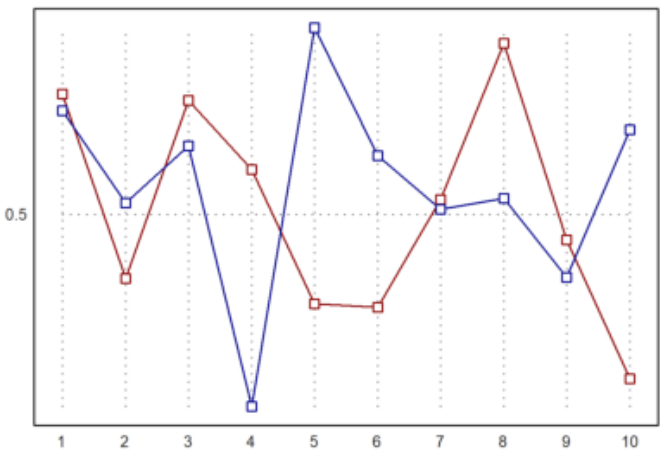
```
>n=10; plot2d(0:n,bin(n,0:n),>addpoints); ...
>labelbox("Binomials",styles="[]",>points,x=0.1,y=0.1, ...
>tcolor=black,>left):
```



```
>
```

Selain itu, gaya plot yang dipakai juga tersedia dalam statplot(). Sama seperti pada plot2d(), pengguna bisa mengatur warna tiap garis plot. Ada pula jenis plot khusus yang ditujukan untuk analisis statistik (lihat panduan statistik untuk detailnya).

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

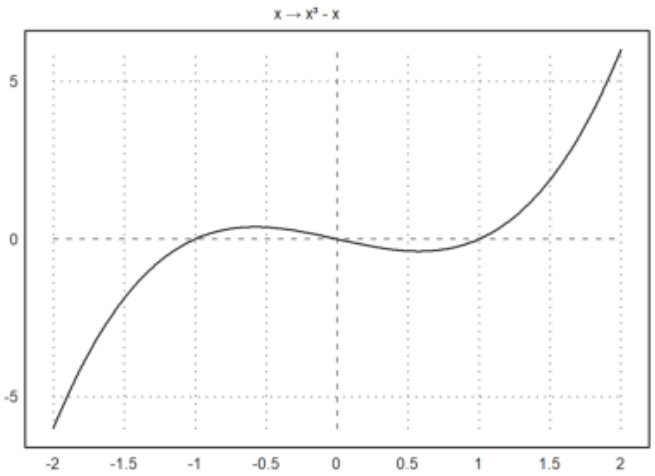



```
>function f(x) &= exp(-x)*sin(2*pi*x); ...
>plot2d("f(x)",0,2pi); ...
>textbox(latex("\text{Example of a damped oscillation}\ f(x)=e^{-x}sin(2\pi x)"),w=0.
```

```
  Closing bracket missing in function call!
  Error in:
  ... amped oscillation}\ f(x)=e^{-x}sin(2\pi x)"),w=0. ...
                                                      ^
```

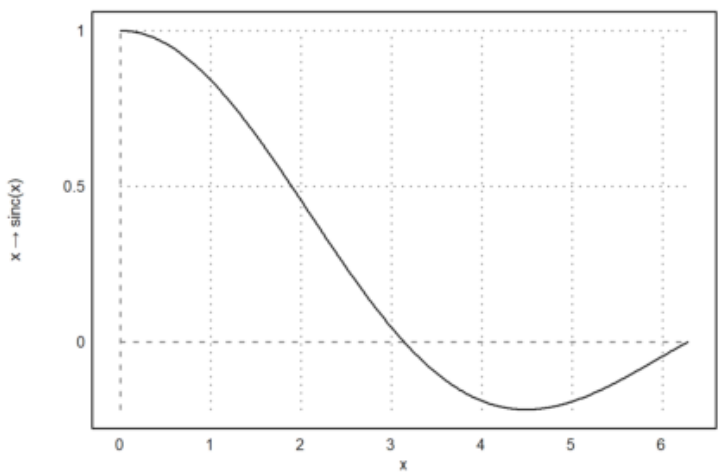
Fungsi lain yang serupa adalah textbox(). Lebar kotak teks secara default mengikuti panjang baris terpanjang, tetapi pengguna juga bisa menentukannya sendiri.

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



Label teks, judul, kotak label, maupun teks lainnya dapat dituliskan dengan string Unicode (lihat dokumentasi EMT untuk detail penggunaannya).
Selain itu, label sumbu x maupun y dapat dibuat vertikal, begitu juga orientasi sumbunya.

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



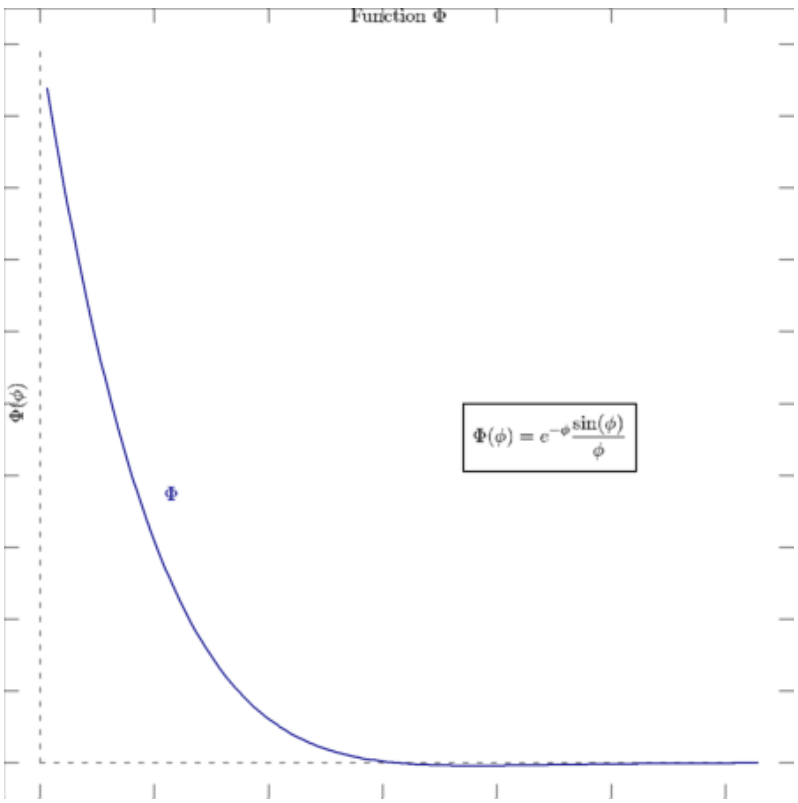

```
>
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
```

PENGGUNAAN SOFTWARE EMT UNTUK PLOT 2D

LaTeX

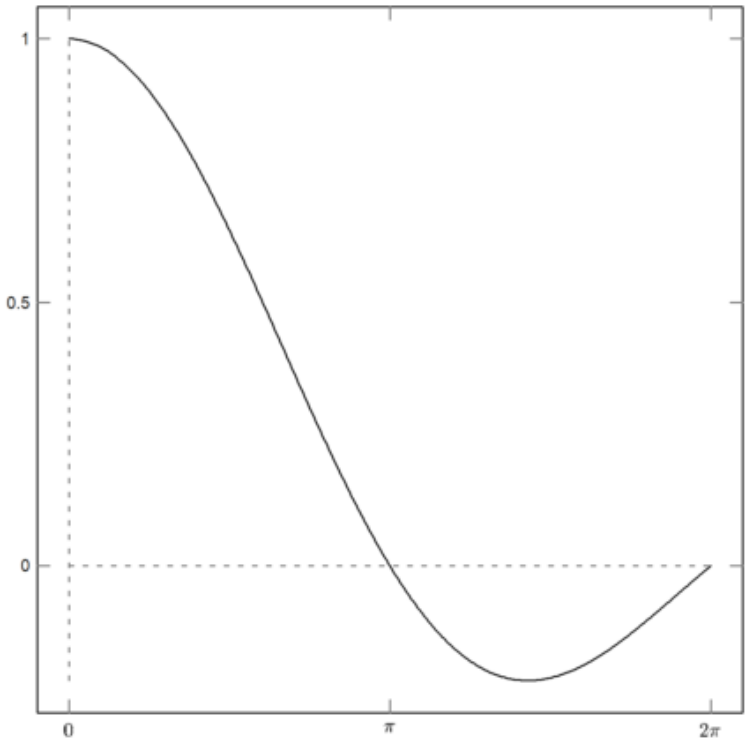
Anda juga dapat memplot rumus LaTeX jika Anda telah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Jalur ke biner "lateks" dan "dvipng" harus berada di jalur sistem, atau Anda harus mengatur LaTeX di menu opsi.
Perhatikan, bahwa penguraian LaTeXlambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, Anda harus memanggil latex() sebelum loop sekali dan menggunakan hasilnya (gambar dalam matriks RGB).
Dalam plot berikut, kami menggunakan LaTeX untuk label x dan y, label, kotak label, dan judul plot.

```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...  title=latex("\text{Fu
>  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, kami menginginkan spasi dan label teks non-konformal pada sumbu x. Kita dapat menggunakan xaxis() dan yaxis() seperti yang akan kita tunjukkan nanti.
Cara termudah adalah dengan membuat plot kosong dengan bingkai menggunakan grid=4, lalu menambahkan grid dengan ygrid() dan xgrid(). Dalam contoh berikut, kami 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):
```



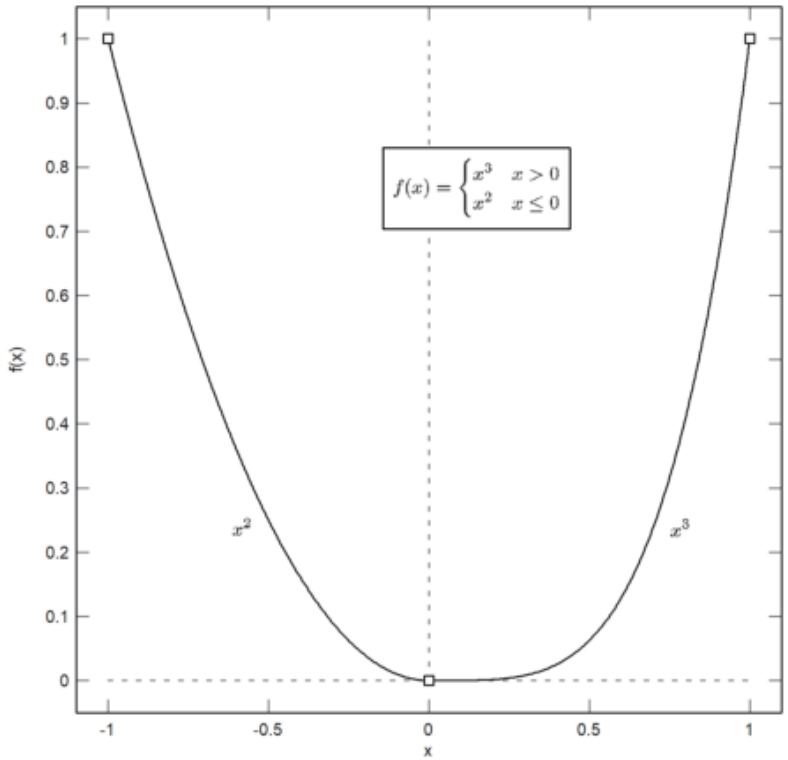
Tentu saja, fungsi juga dapat digunakan.

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

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

Parameter "peta" membantu menggunakan fungsi untuk vektor. Untuk plot, itu tidak perlu. Tetapi untuk menunjukkan vektorisasi itu
berguna, kami menambahkan beberapa poin kunci ke plot di 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 akan dapat menggunakan
LaTeX jika Anda telah menginstal LaTeX 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 \le 0\end{cases}"), ...
>  x=0.7,y=0.2):
```

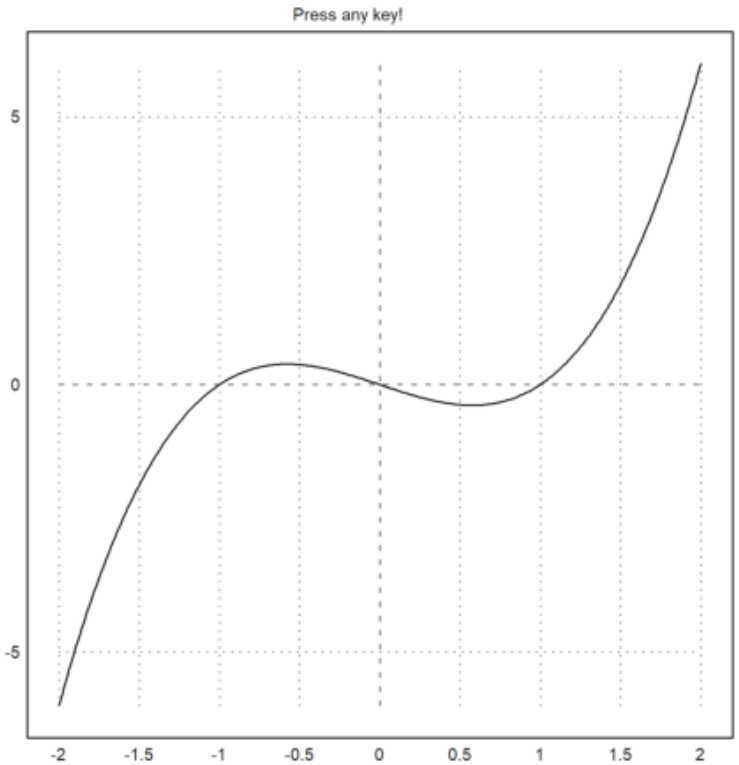


Interaksi pengguna

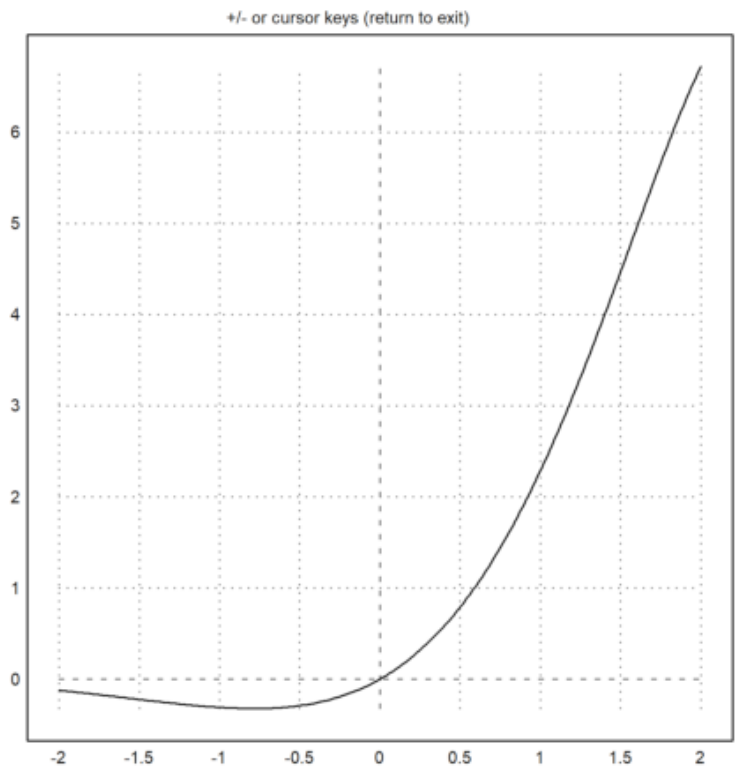
Saat memplot fungsi atau ekspresi, parameter >user memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kursor atau mouse. Pengguna dapat
- perbesar dengan + atau -
- pindahkan plot dengan tombol kursor
- pilih jendela plot dengan mouse
- atur ulang tampilan dengan spasi
- keluar dengan kembali

Tombol spasi akan mengatur ulang plot ke jendela plot asli.
Saat memplot data, flag >user hanya akan menunggu penekanan 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)"):
```



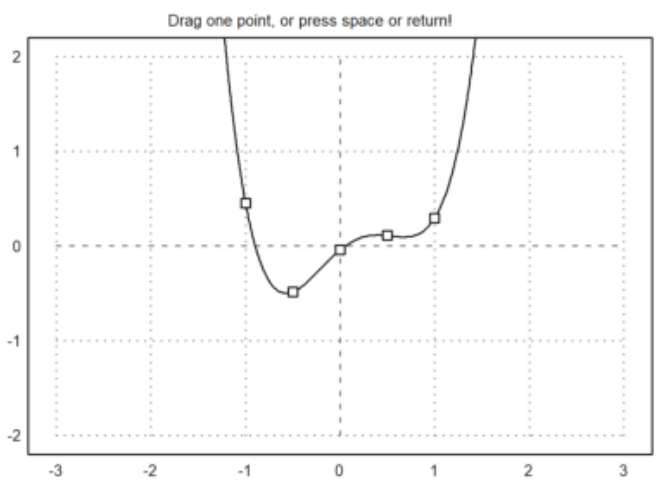
Berikut ini menunjukkan cara interaksi pengguna tingkat lanjut (lihat tutorial tentang pemrograman untuk detailnya).
Fungsi bawaan mousedrag() menunggu event mouse atau keyboard. Ini melaporkan mouse ke bawah, mouse dipindahkan atau mouse ke atas, dan penekanan tombol. Fungsi dragpoints() memanfaatkan ini, dan memungkinkan pengguna menyeret titik mana pun dalam plot.
Kita membutuhkan fungsi plot terlebih dahulu. Sebagai contoh, kita interpolasi dalam 5 titik dengan polinomial. Fungsi harus diplot ke area plot 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 di plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp(). Tanpa ini, kita harus menulis fungsi plotinterp() terlebih dahulu, mengakses nilai secara global.
Sekarang kita menghasilkan beberapa nilai acak, dan membiarkan pengguna menyeret poin.

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

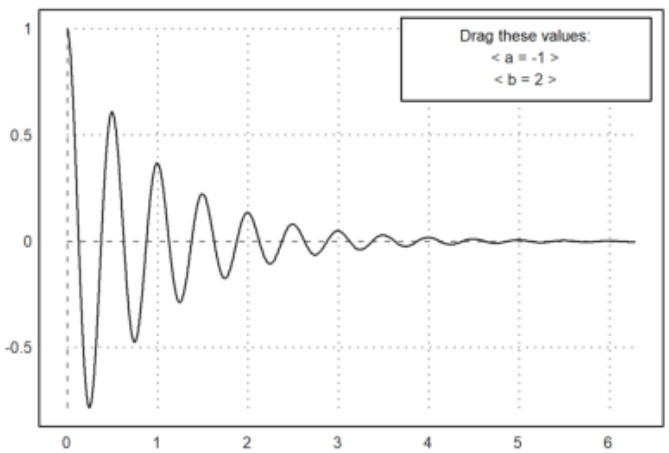


Ada juga fungsi, yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.
Pertama kita membutuhkan fungsi plot.

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

Kemudian kita membutuhkan nama untuk parameter, nilai awal dan matriks rentang nx2, opsional baris judul.
Ada slider interaktif, yang dapat mengatur nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

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



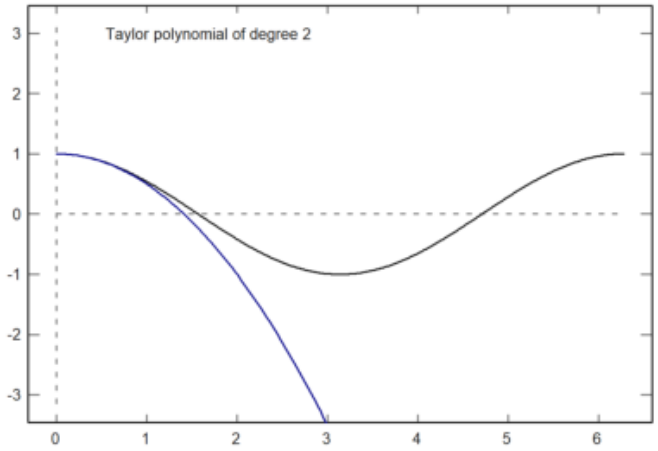
Dimungkinkan untuk membatasi nilai yang diseret ke bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor derajat n ke fungsi kosinus.

```
>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 kami mengizinkan derajat n bervariasi dari 0 hingga 20 dalam 20 pemberhentian. Hasil dragvalues() digunakan untuk memplot sketsa dengan n ini, dan untuk memasukkan plot ke dalam buku catatan.

```
>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 atas jendela plot, meninggalkan jejak poin.

```
>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
```

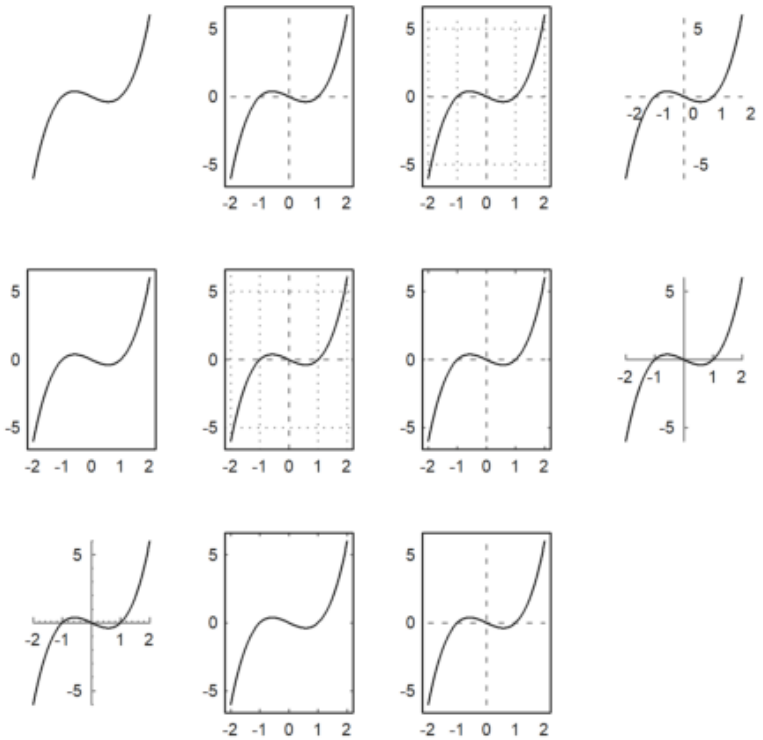
```
>dragtest // lihat hasilnya dan cobalah lakukan!
```

Gaya Plot 2D

Secara default, EMT menghitung tick sumbu otomatis dan menambahkan label ke setiap tick. Ini dapat diubah dengan parameter grid. Gaya default sumbu dan label dapat dimodifikasi. Selain itu, label dan judul dapat ditambahkan secara manual. Untuk mengatur ulang ke gaya default, gunakan reset().

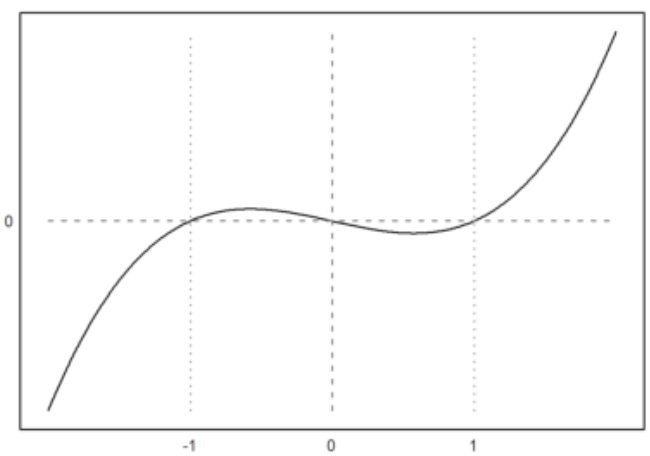
```
>aspect();
>figure(3,4); ...
> figure(1); plot2d("x^3-x",grid=0); ... // no grid, frame or axis
> figure(2); plot2d("x^3-x",grid=1); ... // x-y-axis
> figure(3); plot2d("x^3-x",grid=2); ... // default ticks
> figure(4); plot2d("x^3-x",grid=3); ... // x-y- axis with labels inside
> figure(5); plot2d("x^3-x",grid=4); ... // 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):
```



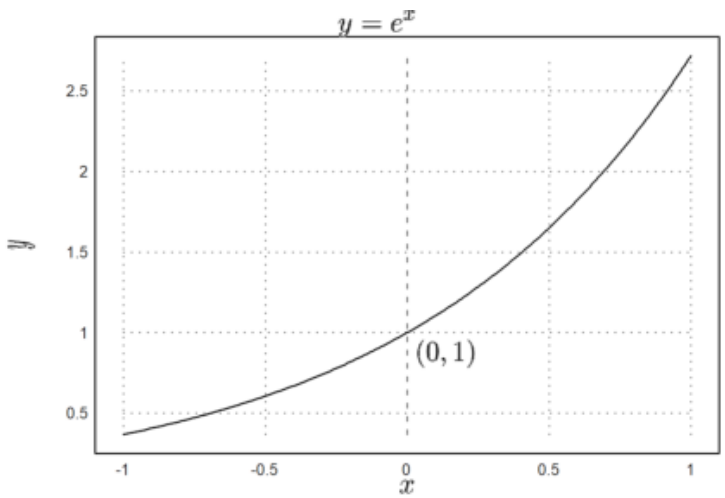
Parameter <frame mematikan frame, dan framecolor=blue mengatur frame ke warna biru.
Jika Anda ingin centang sendiri, Anda dapat menggunakan style=0, dan menambahkan semuanya nanti.

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



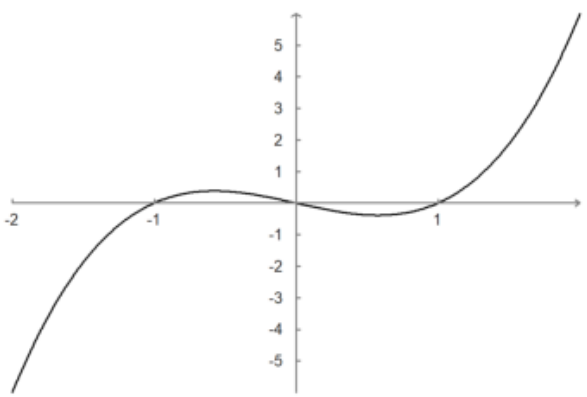
Untuk judul plot dan label sumbu, lihat contoh berikut.

```
>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 dengan xaxis() dan yaxis().

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

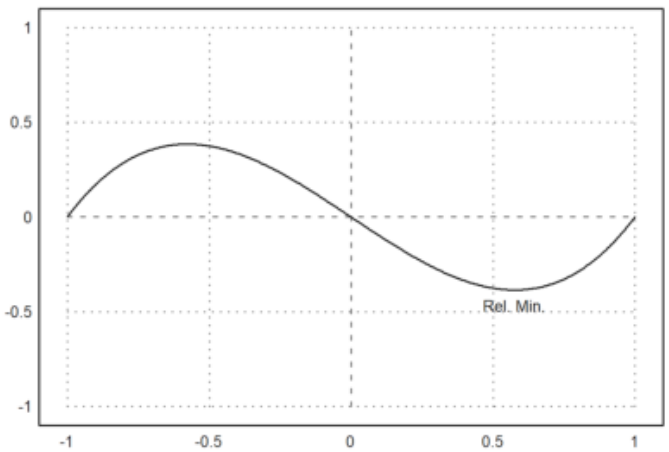


Teks pada plot dapat diatur dengan label(). Dalam contoh berikut, "lc" berarti tengah bawah. Ini mengatur posisi label relatif terhadap koordinat plot

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

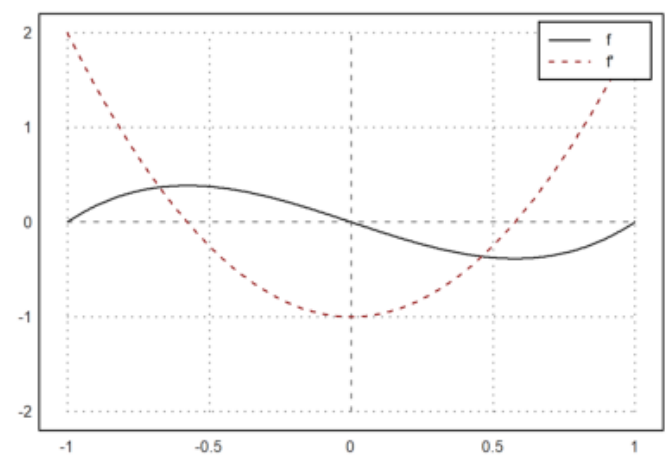
$$x^3 - x$$

```
>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
```

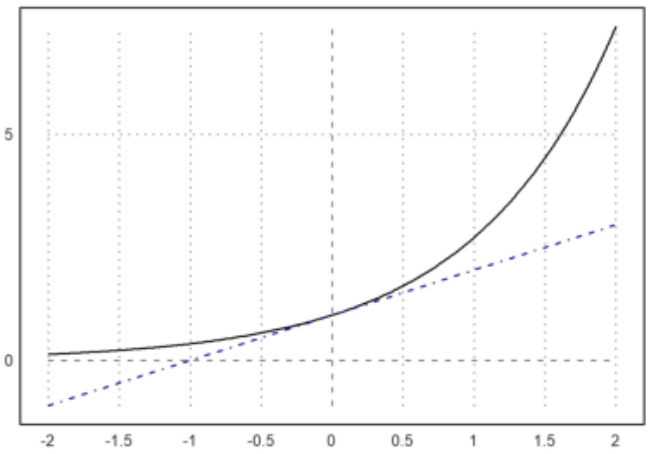


Ada juga kotak teks.

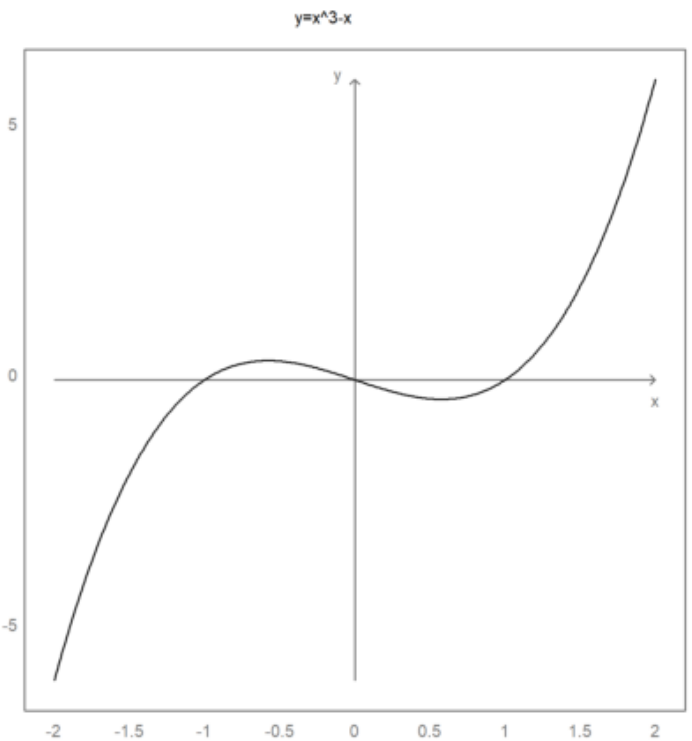
```
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // derivative
>labelbox(["f","f'"],["-","--"],[black,red]): // label box
```



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

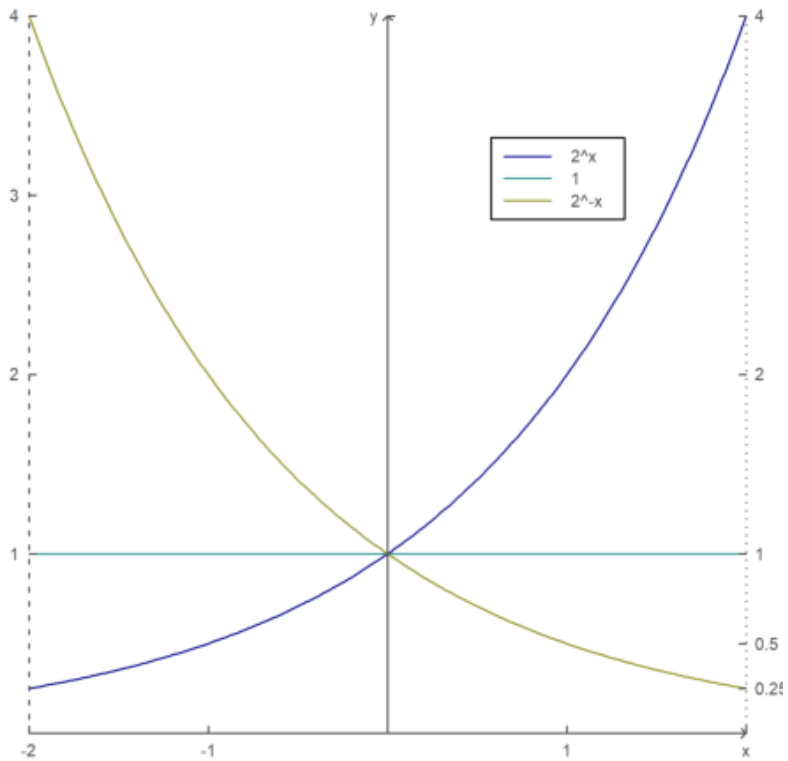



```
>gridstyle("->",color=gray,textcolor=gray,framecolor=gray);  ...
> plot2d("x^3-x",grid=1);   ...
> settitle("y=x^3-x",color=black); ...
> label("x",2,0,pos="bc",color=gray);  ...
> label("y",0,6,pos="cl",color=gray); ...
> reset():
```



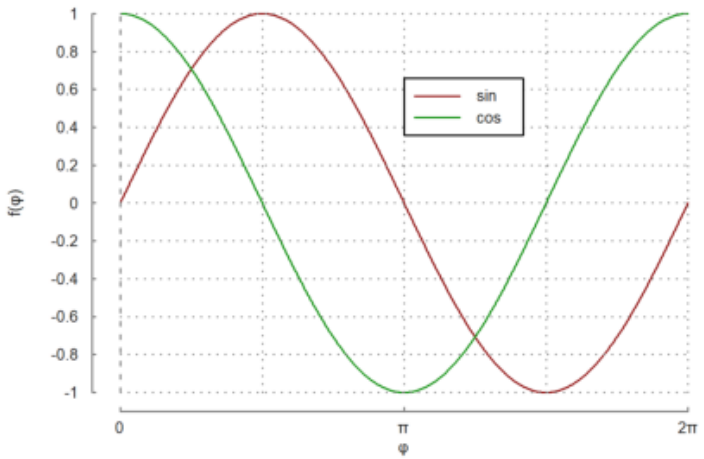
Untuk kontrol lebih, sumbu x dan sumbu y dapat dilakukan secara manual.
Perintah fullwindow() memperluas jendela plot karena kita tidak lagi membutuhkan tempat untuk label di luar jendela plot. Gunakan shrinkwindow() atau reset() untuk mengatur ulang ke default.

```
>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 di luar area plot.

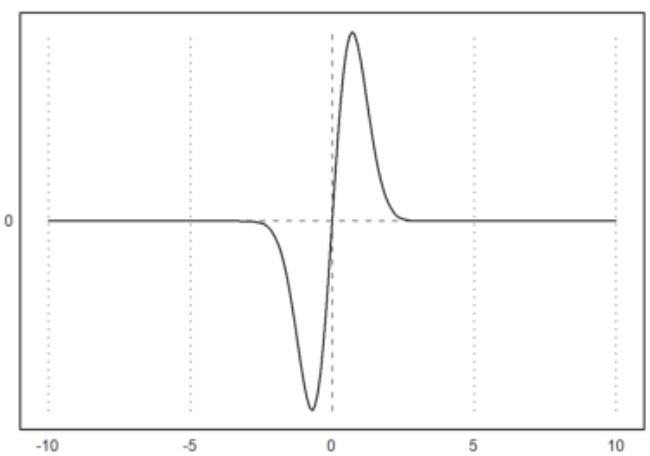
```
>aspect(1.5);
>plot2d(["sin(x)","cos(x)"],0,2pi,color=[red,green],<grid,<frame); ...
> xaxis(-1.1,(0:2)*pi,xt=["0",u"&pi;",u"2&pi;"],style="-",>ticks,>zero);  ...
> xgrid((0:0.5:2)*pi,<ticks); ...
> yaxis(-0.1*pi,-1:0.2:1,style="-",>zero,>grid); ...
> labelbox(["sin","cos"],colors=[red,green],x=0.5,y=0.2,>left);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



Merencanakan Data 2D

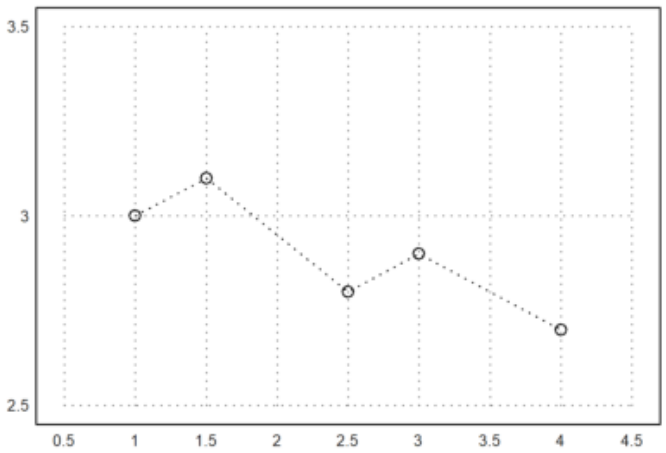
Jika x dan y adalah vektor data, data ini akan digunakan sebagai koordinat x dan y dari suatu kurva. Dalam hal ini, a, b, c, dan d, atau radius r dapat ditentukan, atau jendela plot akan menyesuaikan secara otomatis dengan data. Atau, >persegi dapat diatur untuk menjaga rasio aspek persegi.
Memplot ekspresi hanyalah singkatan untuk plot data. Untuk plot data, Anda memerlukan satu atau beberapa baris nilai x, dan satu atau beberapa baris nilai y. Dari rentang dan nilai-x, fungsi plot2d akan menghitung data yang akan diplot, secara default dengan evaluasi fungsi yang adaptif. Untuk plot titik gunakan ">titik", untuk garis campuran dan titik gunakan ">tambahan".
Tapi Anda bisa memasukkan data secara langsung.
- Gunakan vektor baris untuk x dan y untuk satu fungsi.
- Matriks untuk x dan y diplot baris demi baris. Berikut adalah contoh dengan satu baris untuk x dan y.

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

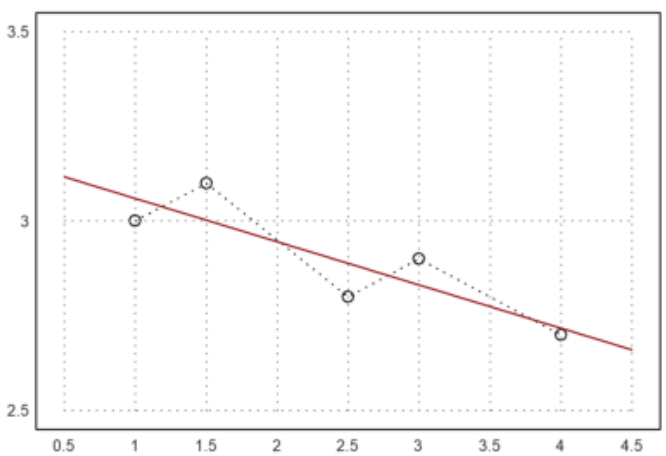


Data juga dapat diplot sebagai titik. Gunakan poin=true untuk ini. Plotnya bekerja seperti poligon, tetapi hanya menggambar sudut-sudutnya.
- style="...": Pilih dari "[]", "<>", "o", ".", "..", "+", "*", "[]", "< >", "o", "..", "", "|".
Untuk memplot set poin gunakan >points. Jika warna adalah vektor warna, setiap titik
mendapat warna yang berbeda. Untuk matriks koordinat dan vektor kolom, warna berlaku untuk 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
```

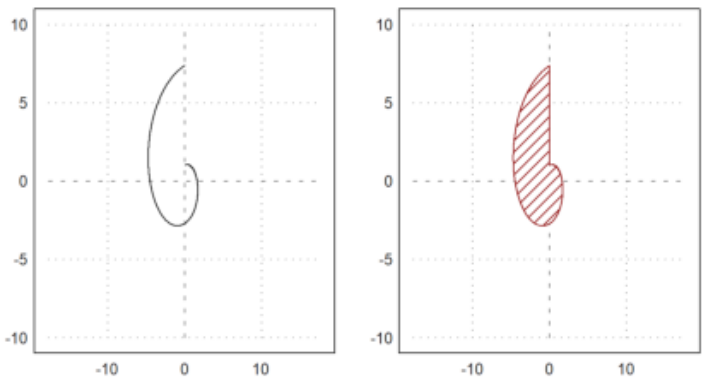


Menggambar Daerah Yang Dibatasi Kurva

Plot data benar-benar poligon. Kita juga dapat memplot kurva atau kurva terisi.
- terisi=benar mengisi plot.
- style="...": Pilih dari "", "/", "\", "\/".
- fillcolor: Lihat di atas untuk warna yang tersedia.

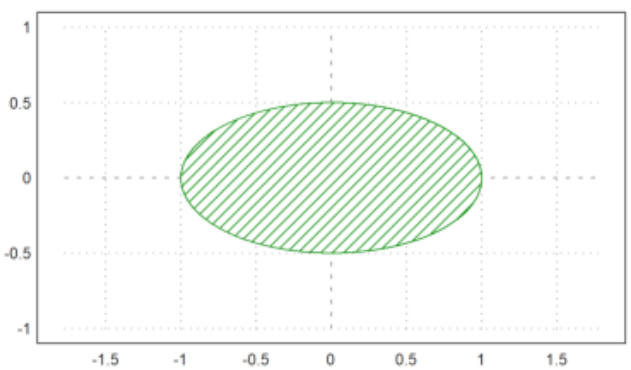
Warna isian ditentukan oleh argumen "fillcolor", dan pada <outline opsional mencegah menggambar batas untuk semua gaya kecuali yang default.

```
>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):
```

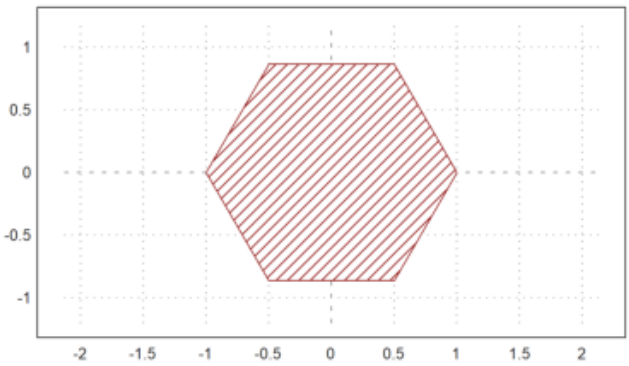


Dalam contoh berikut kami memplot elips terisi dan dua segi enam terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian berbeda.

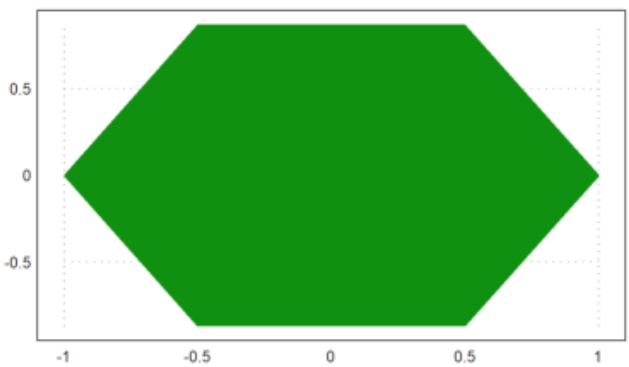
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```




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

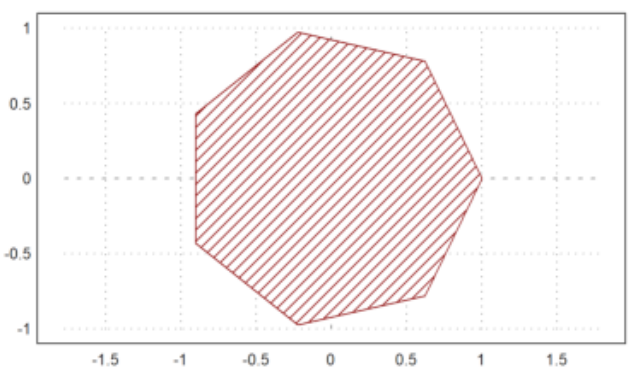



```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



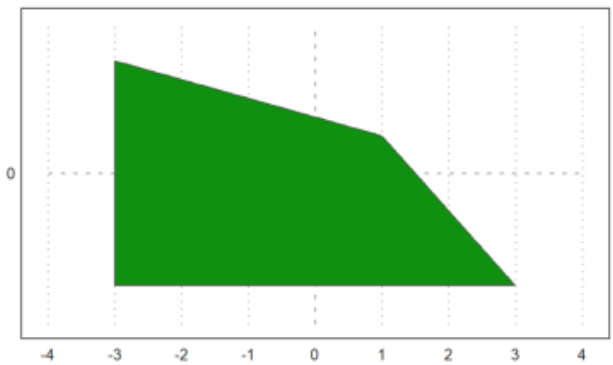
Contoh lainnya adalah segi empat, yang kita buat dengan 7 titik pada lingkaran satuan.

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



Berikut ini adalah himpunan nilai maksimal dari empat kondisi linier yang kurang dari atau sama dengan 3. Ini adalah A[k].v<=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):
```

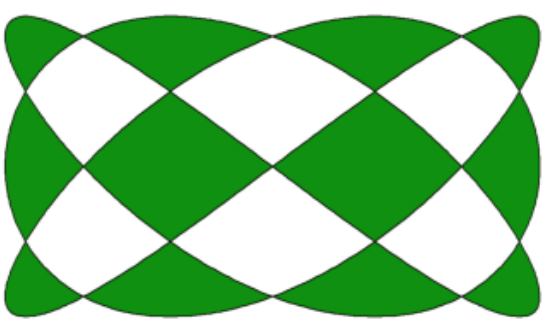


Poin utama dari bahasa matriks adalah memungkinkan untuk menghasilkan tabel fungsi dengan mudah.

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

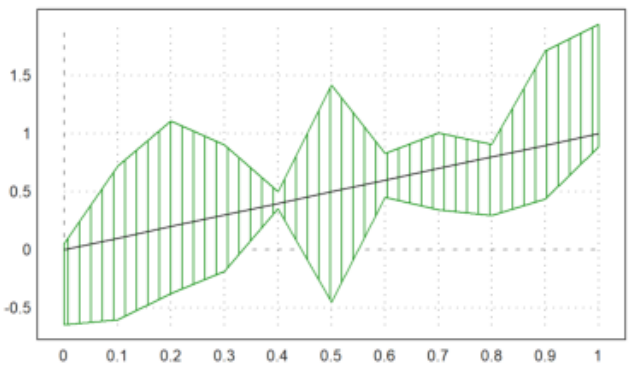
Kami sekarang memiliki vektor x dan y nilai. plot2d() dapat memplot nilai-nilai ini sebagai kurva yang menghubungkan titik-titik. Plotnya bisa diisi. Pada kasus ini ini menghasilkan hasil yang bagus karena aturan lilitan, yang digunakan untuk
isi.

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



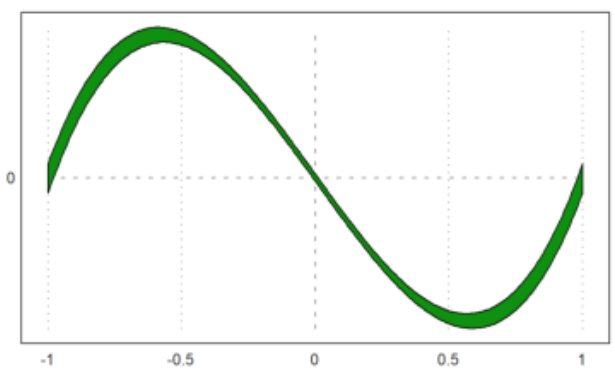
Sebuah vektor interval diplot terhadap nilai x sebagai daerah terisi antara nilai interval bawah dan atas.
Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga digunakan untuk memplot kesalahan statistik.

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



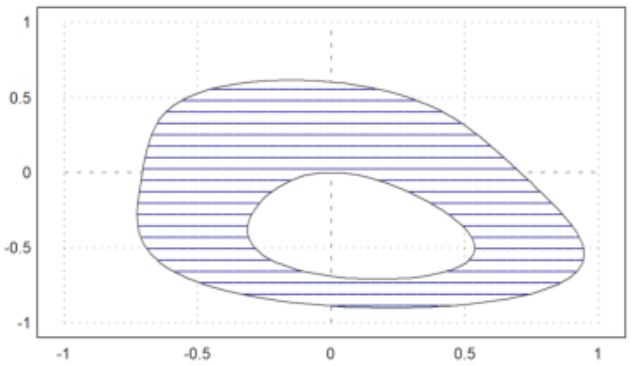
Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka plot2d akan memplot rentang interval yang terisi dalam bidang. Gaya isian sama dengan gaya poligon.

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



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

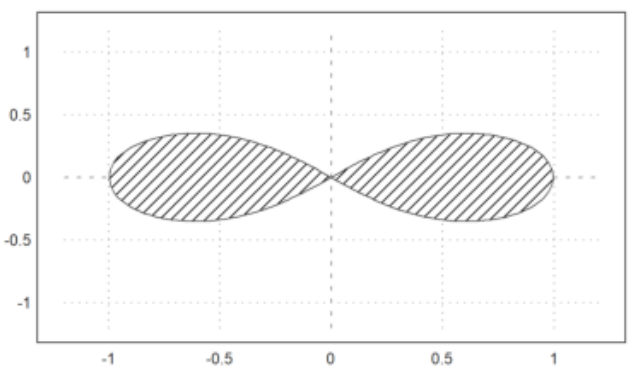
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



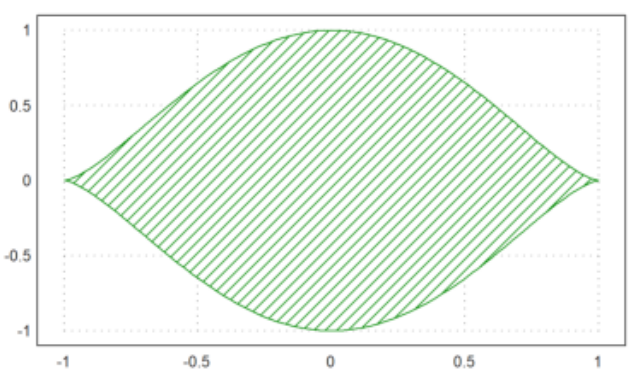
Kami juga dapat mengisi rentang nilai seperti

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

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

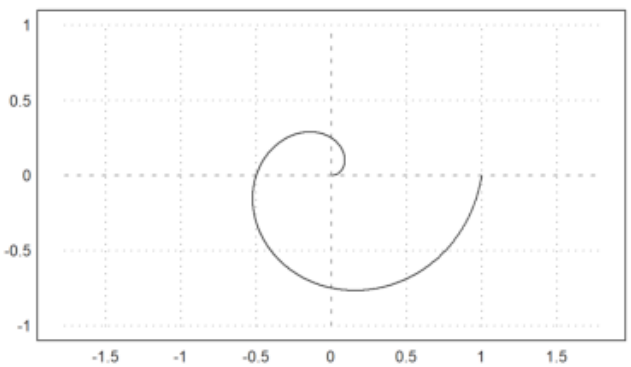



```
>plot2d("cos(x)","sin(x)^3",xmin=0,xmax=2pi,>filled,style="/"):
```



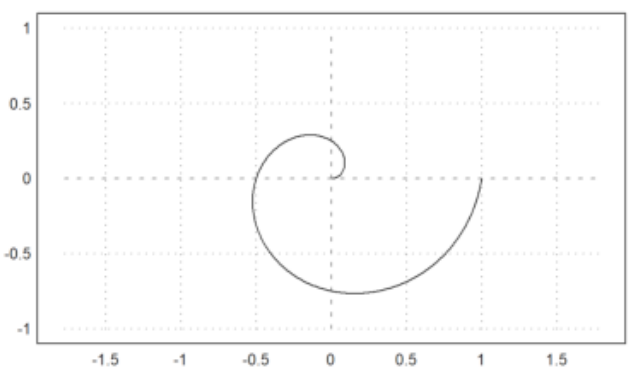
Nilai-x tidak perlu diurutkan. (x,y) hanya menggambarkan kurva. Jika x diurutkan, kurva tersebut merupakan grafik fungsi.
Dalam contoh berikut, kami memplot spiral
lateks: \gamma(t) = t \cdot (\cos(2\pi t),\sin(2\pi t))
Kita perlu menggunakan banyak titik untuk tampilan yang halus atau fungsi adaptif() untuk mengevaluasi ekspresi (lihat fungsi adaptif() untuk lebih jelasnya).

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

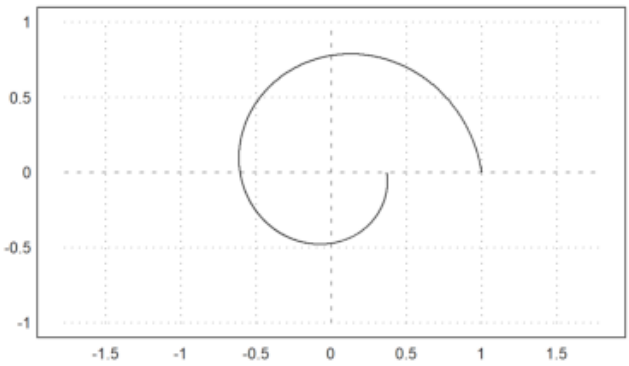


Atau, dimungkinkan untuk menggunakan dua ekspresi untuk kurva. Berikut ini plot kurva yang sama seperti di atas.

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



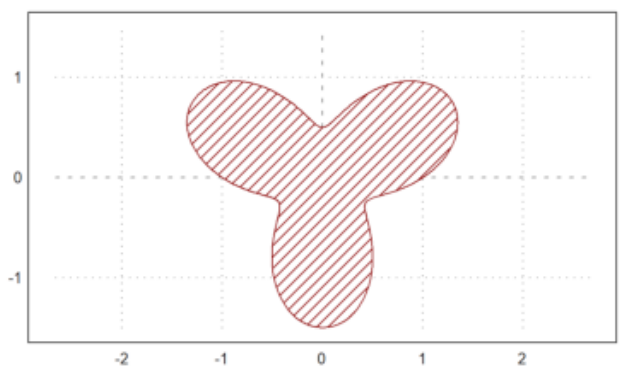
Dalam contoh berikutnya, kami memplot kurva

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

dengan

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

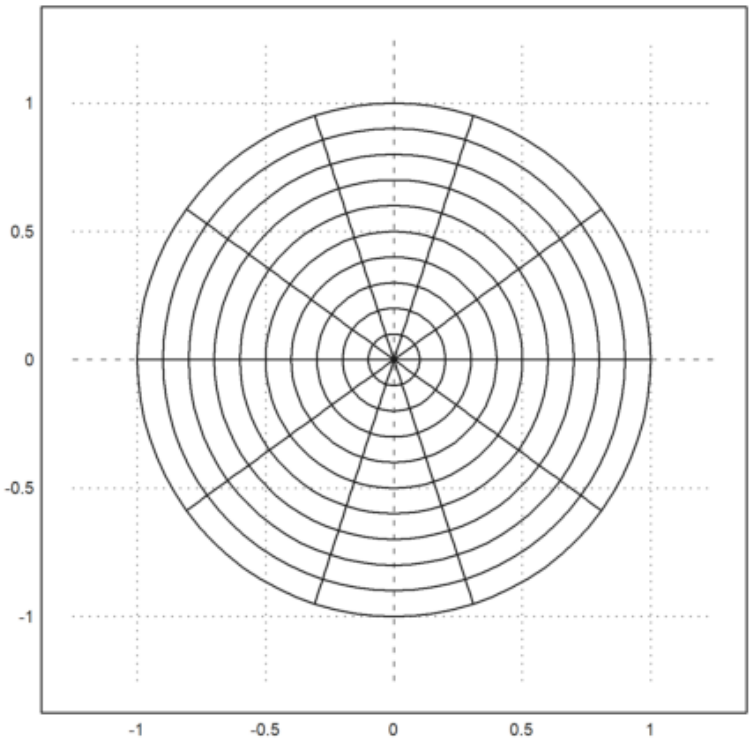
```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
>plot2d(x,y,>filled,fillcolor=red,style="/",r=1.5):
```



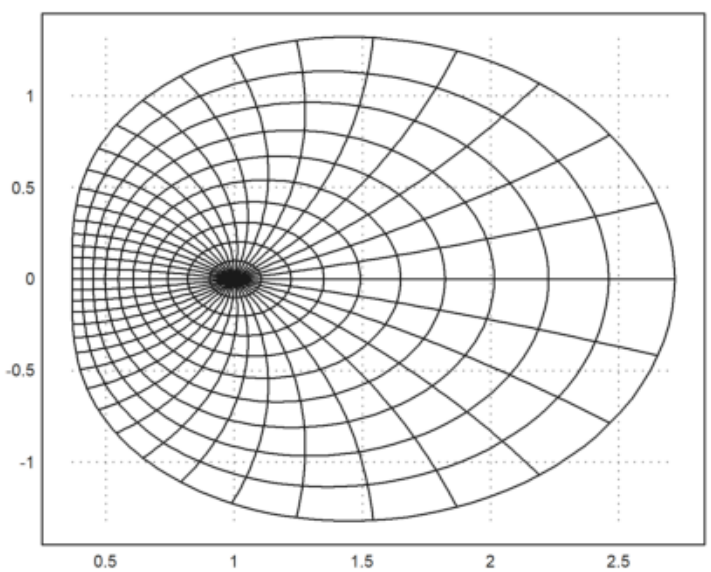
Menggambar Grafik Bilangan Kompleks

Array bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan terhubung. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi 1x2) dalam argumen cgrid, hanya garis kisi tersebut yang terlihat.
Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi di bidang kompleks.
Dalam contoh berikut, kami memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

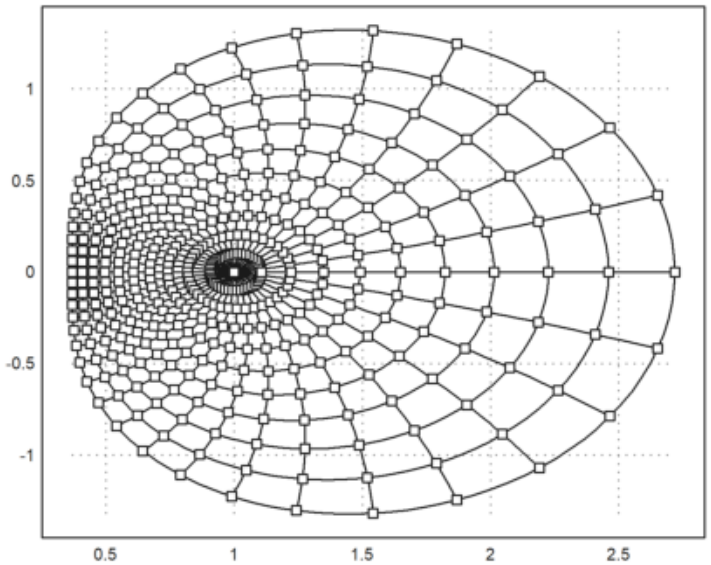
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```




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



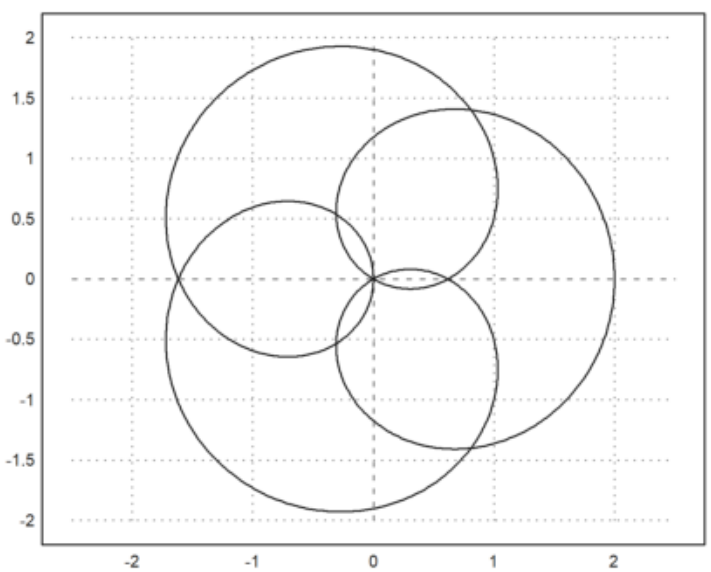

```
>r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=r*exp(I*a);
>plot2d(exp(z),>points,>add):
```



Sebuah vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian real dan bagian imajiner.
Dalam contoh, kami memplot lingkaran satuan dengan lateks:

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

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



Plot Statistik

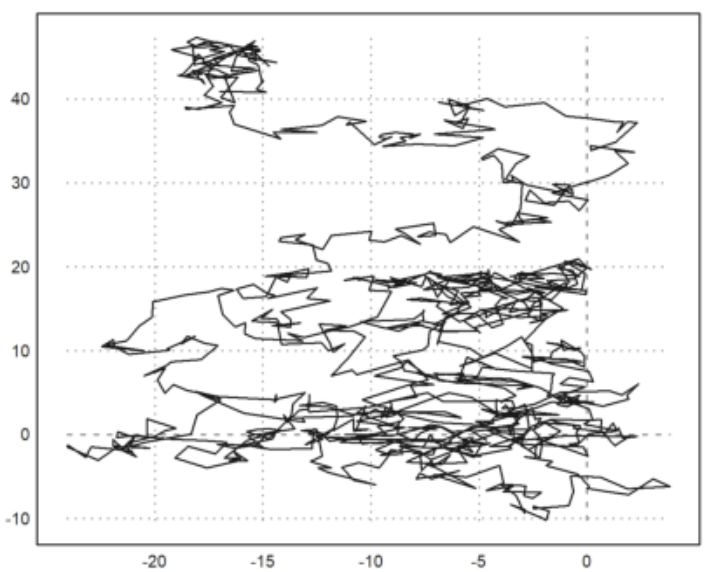
Ada banyak fungsi yang dikhususkan pada plot statistik. Salah satu plot yang sering digunakan adalah plot kolom.
Jumlah kumulatif dari nilai terdistribusi 0-1-normal menghasilkan jalan acak.

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

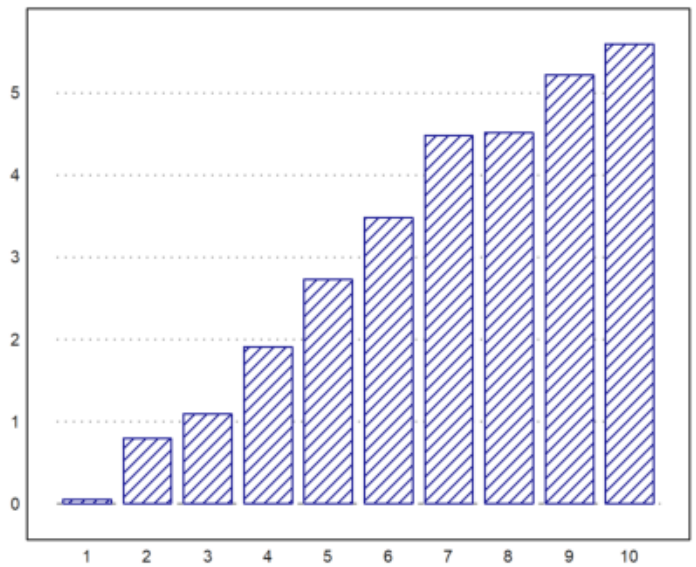


Menggunakan dua baris menunjukkan jalan dalam dua dimensi.

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

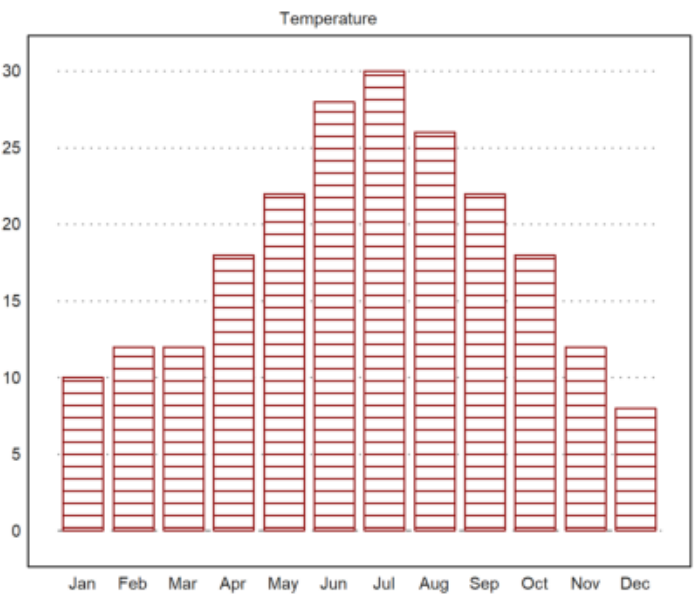


```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

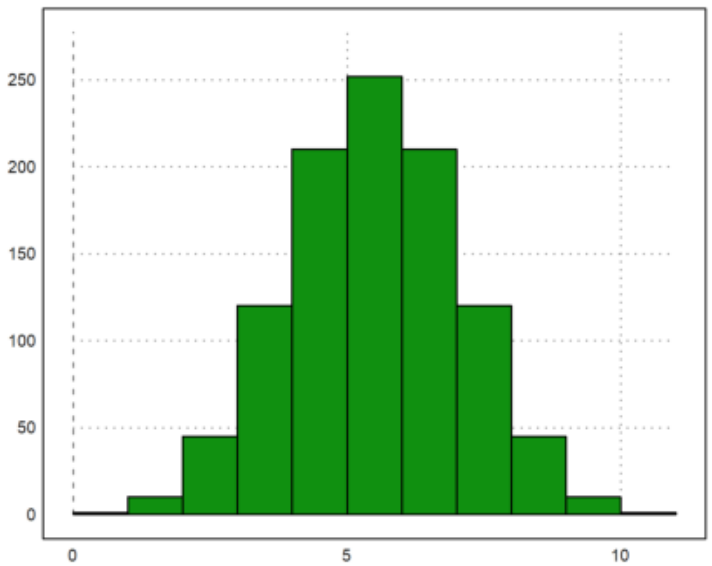


Itu juga dapat menampilkan string sebagai label.

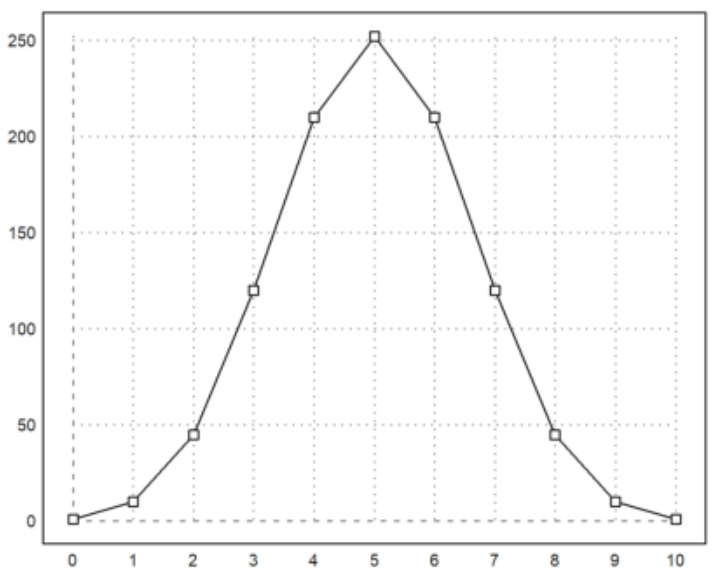
```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
>  "Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnsplot(values,lab=months,color=red,style="-");
>title("Temperature"):
```



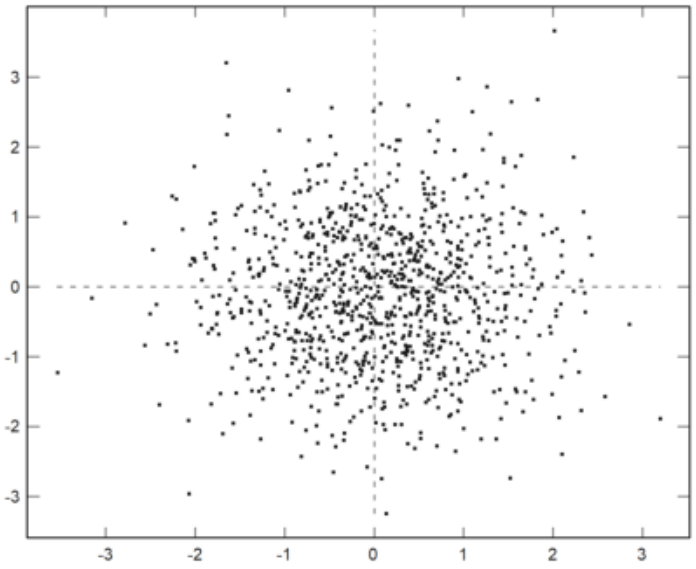
```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```



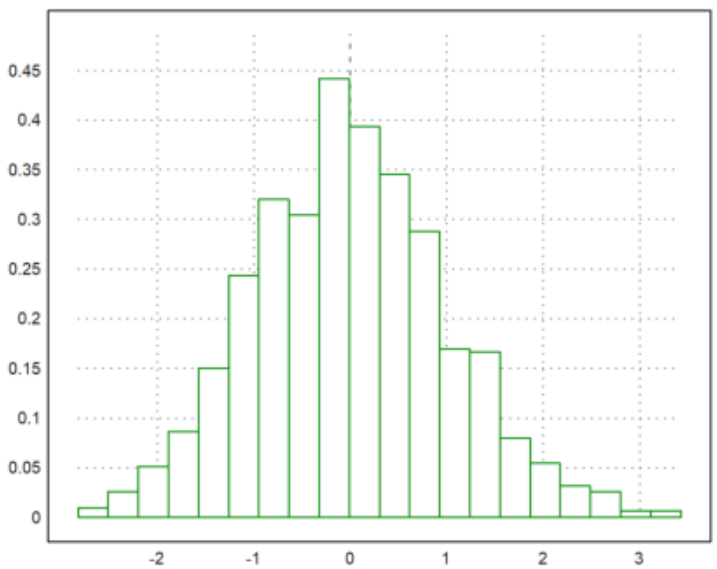

```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



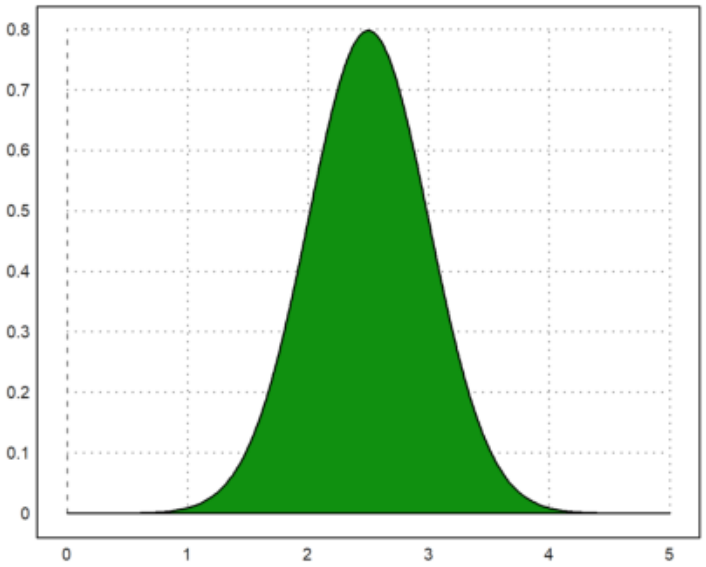

```
>plot2d(normal(1000),normal(1000),>points,grid=6,style="..");
```




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

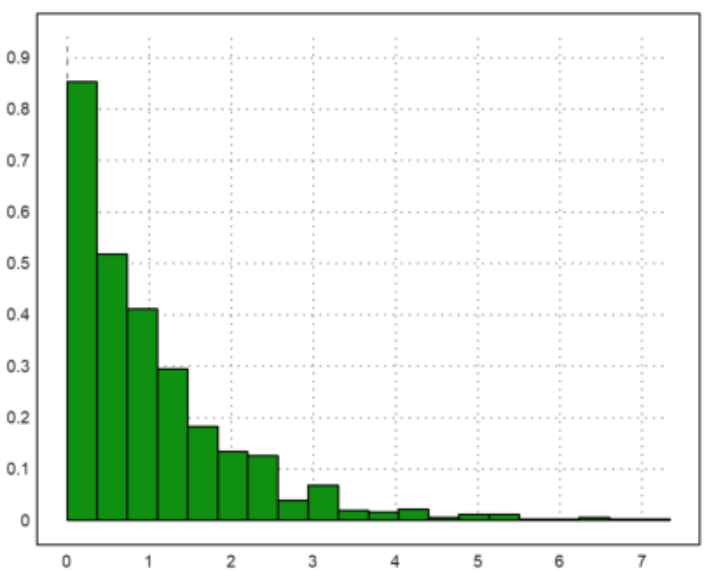



```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



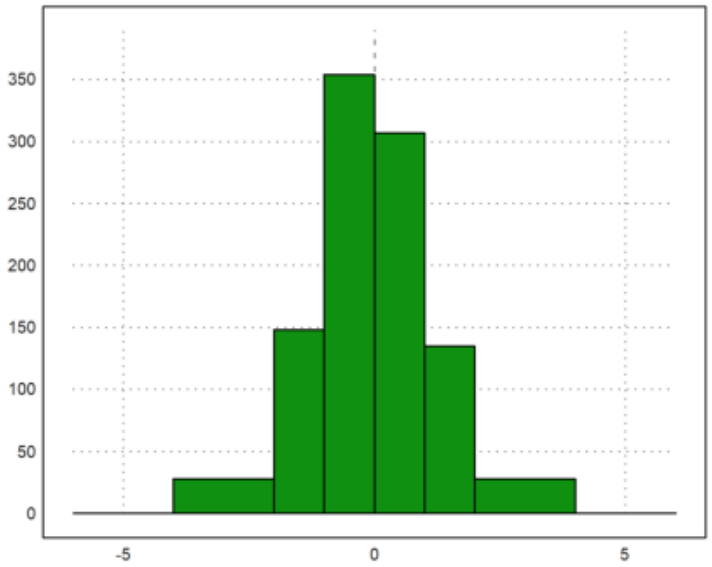
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan distribution=n dengan plot2d.

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



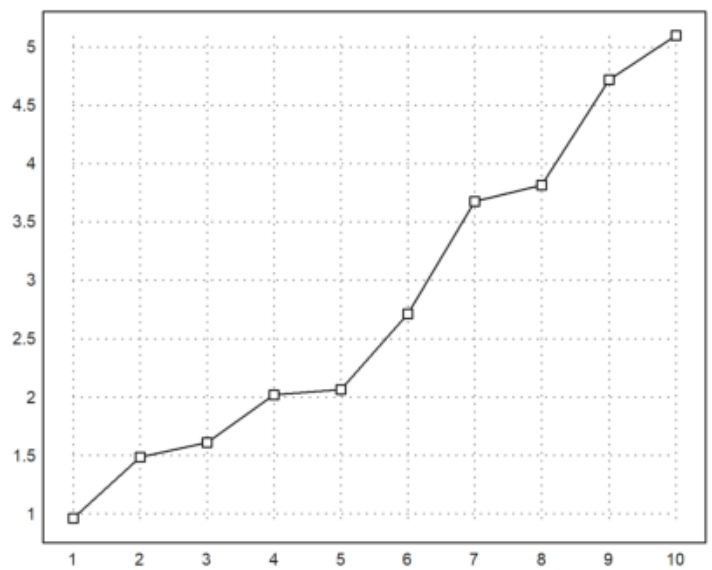
Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan >bar di plot3d, atau dengan plot kolom.

```
>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):
```

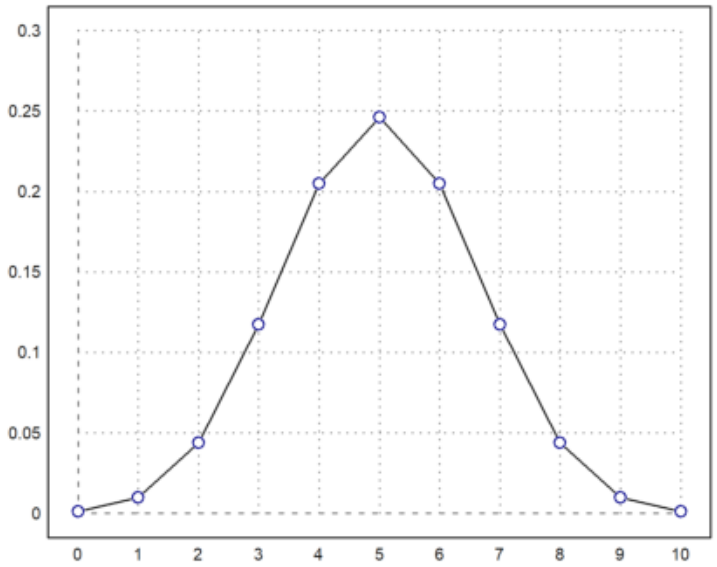


Fungsi statplot() menyetel gaya dengan string sederhana.

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

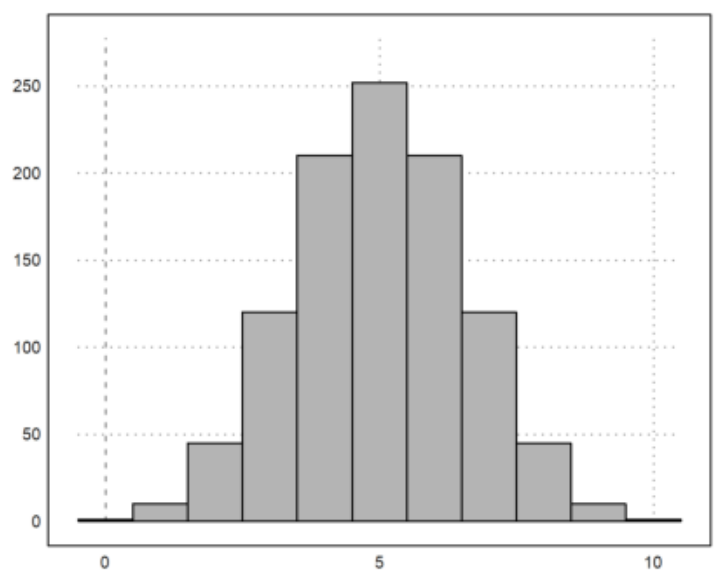



```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



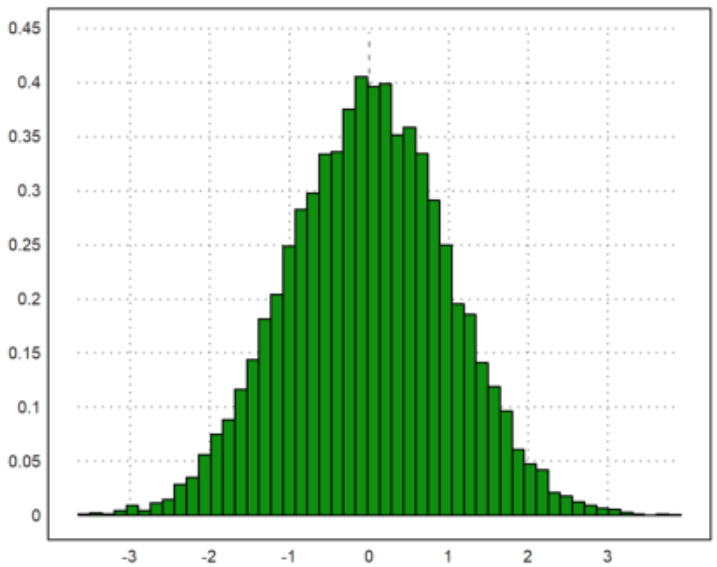
Selain itu, data dapat diplot sebagai batang. Dalam hal ini, x harus diurutkan dan satu elemen lebih panjang dari y. Bilah akan memanjang dari x[i] ke x[i+1] dengan nilai y[i]. Jika x memiliki ukuran yang sama dengan y, maka akan diperpanjang satu elemen dengan spasi terakhir.
Gaya isian dapat digunakan seperti di atas.

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

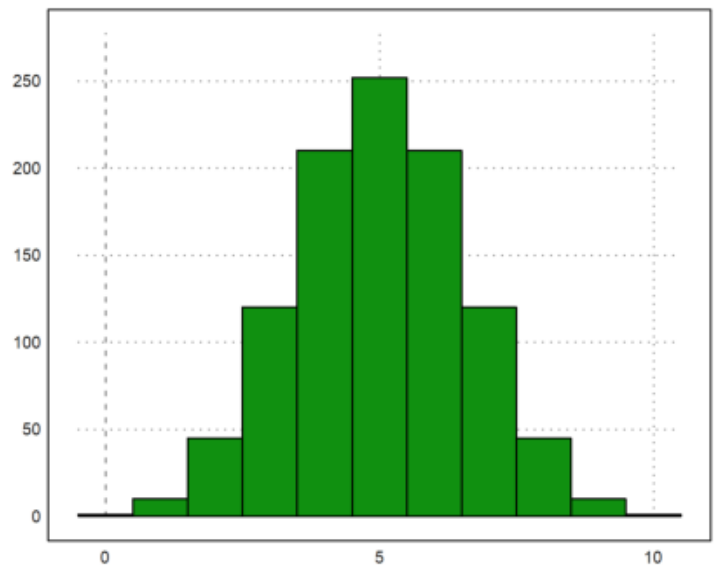


Data untuk plot batang (bar=1) dan histogram (histogram=1) dapat dinyatakan secara eksplisit dalam xv dan yv, atau dapat dihitung dari distribusi empiris dalam xv dengan >distribusi (atau distribusi=n). Histogram nilai xv akan dihitung secara otomatis dengan >histogram. Jika >genap ditentukan, nilai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

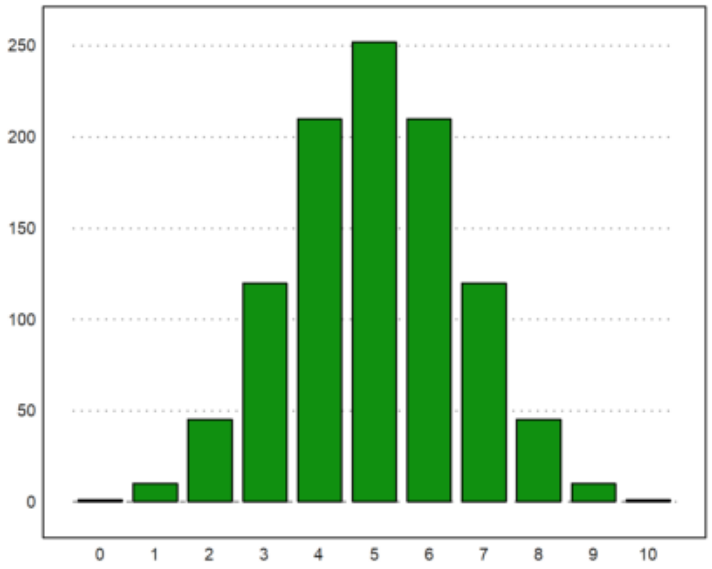
```
>plot2d(normal(10000),distribution=50):
```



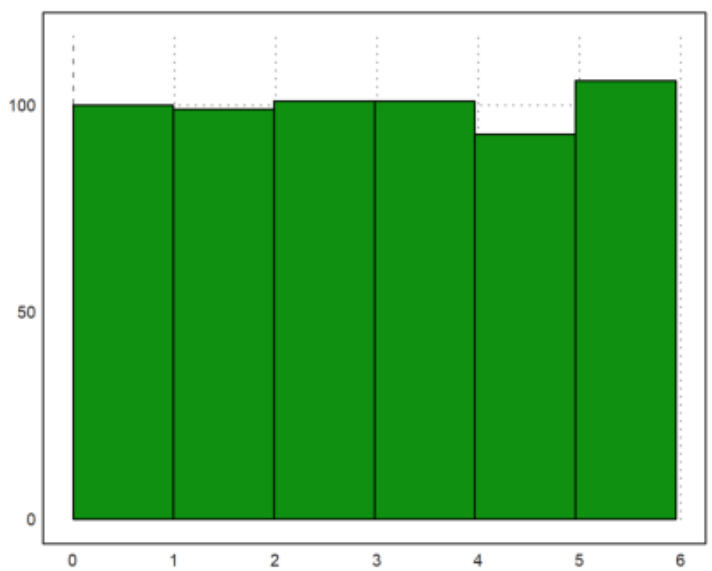

```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```




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

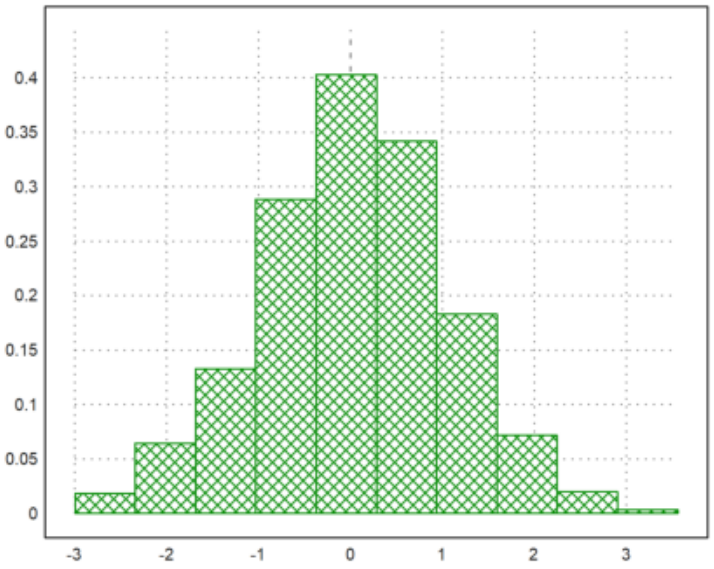



```
>plot2d(random(600)*6,histogram=6):
```



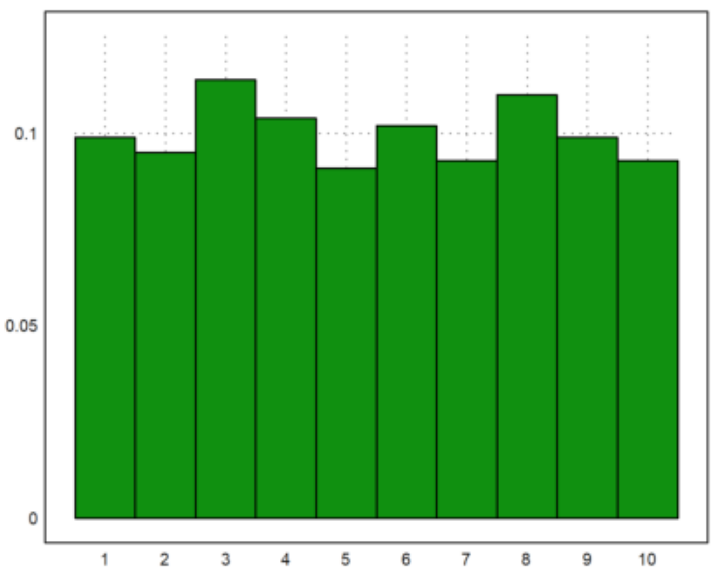
Untuk distribusi, ada parameter distribusi=n, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.

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



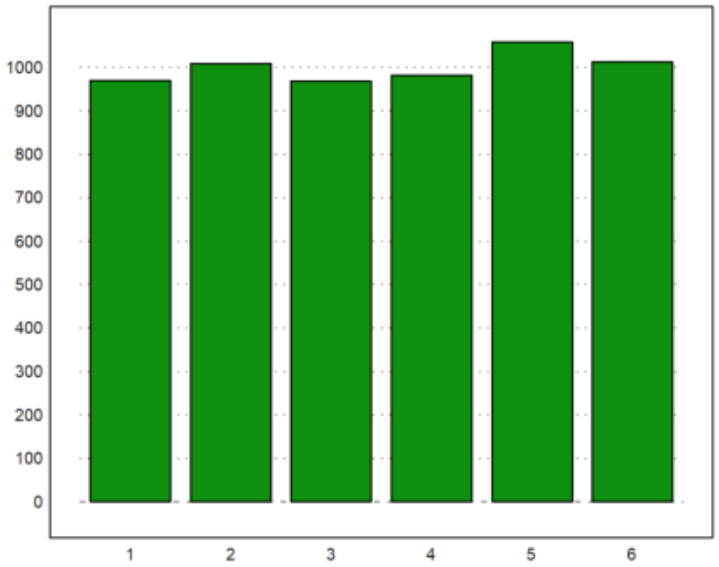
Dengan parameter even=true, ini akan menggunakan interval integer.

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

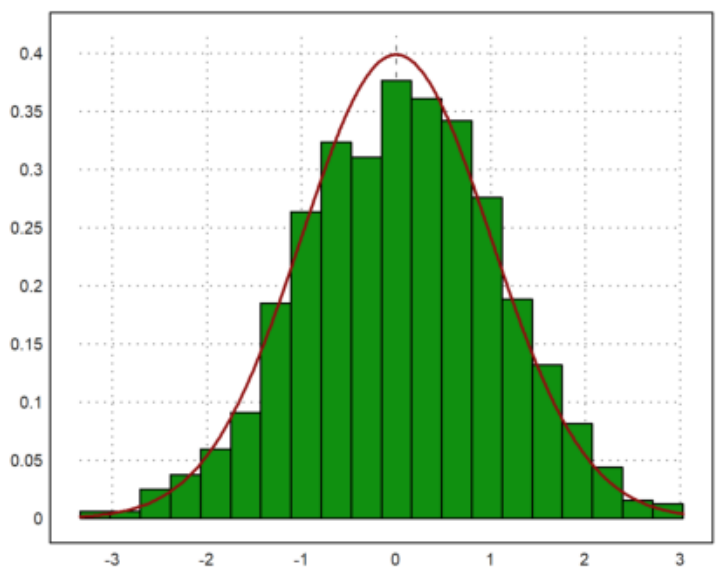


Perhatikan bahwa ada banyak plot statistik, yang mungkin berguna. Silahkan lihat tutorial tentang statistik.

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

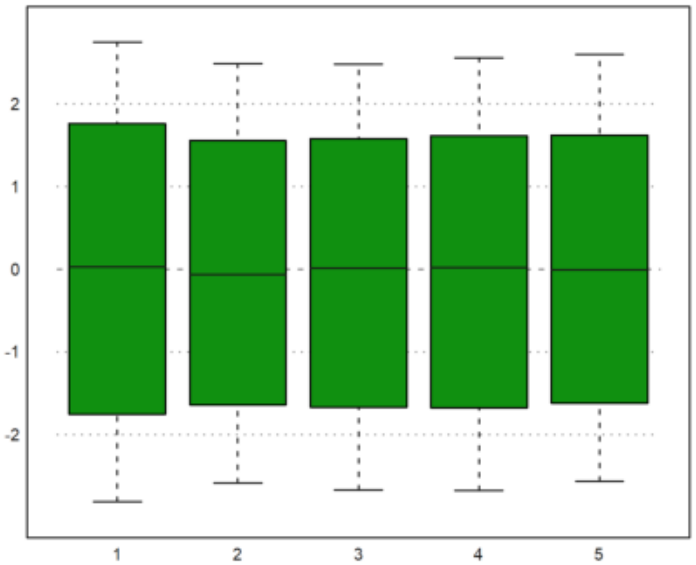



```
>plot2d(normal(1,1000),>distribution); ...
>  plot2d("qnormal(x)",color=red,thickness=2,>add):
```



Ada juga banyak plot khusus untuk statistik. Boxplot menunjukkan kuartil dari distribusi ini dan banyak outlier. Menurut definisi, outlier dalam boxplot adalah data yang melebihi 1,5 kali kisaran 50% tengah plot.

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



Fungsi Implisit

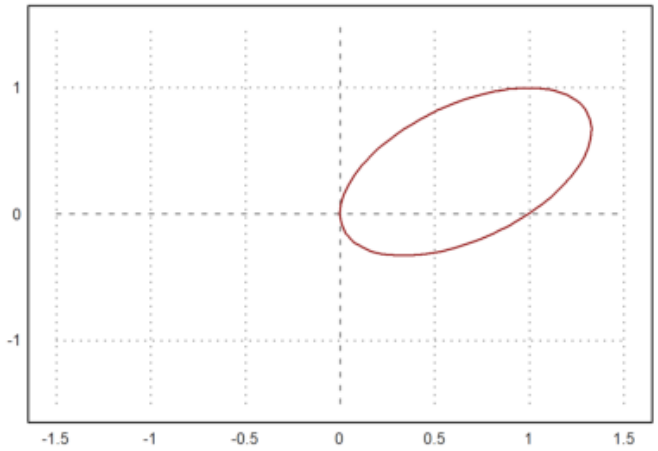
Plot implisit menunjukkan garis level yang menyelesaikan f(x,y)=level, di mana "level" dapat berupa nilai tunggal atau vektor nilai. Jika level="auto", akan ada garis level nc, yang akan menyebar antara fungsi minimum dan maksimum secara merata. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan >hue untuk menunjukkan nilai fungsi. Untuk fungsi implisit, xv harus berupa fungsi atau ekspresi dari parameter x dan y, atau, sebagai alternatif, xv dapat berupa matriks nilai.
Euler dapat menandai garis level

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

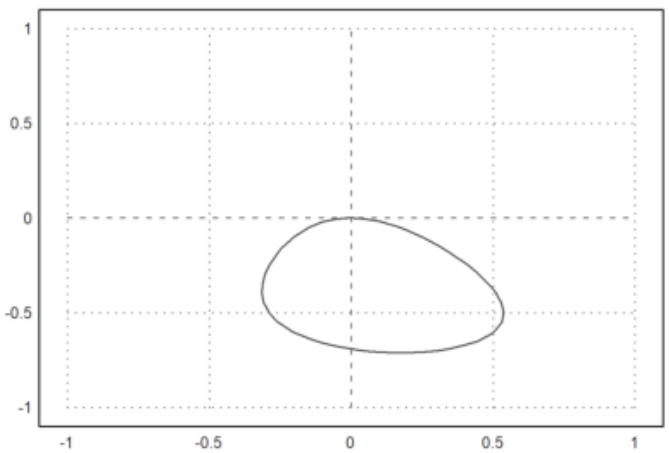
dari fungsi apapun.

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

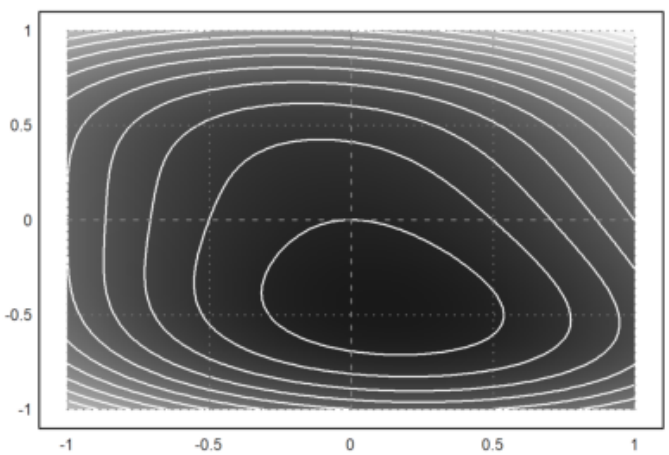
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```



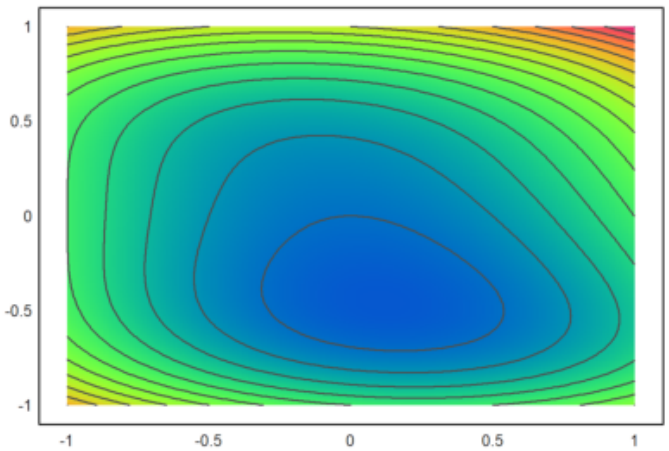

```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0): // Solutions of f(x,y)=0
```




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

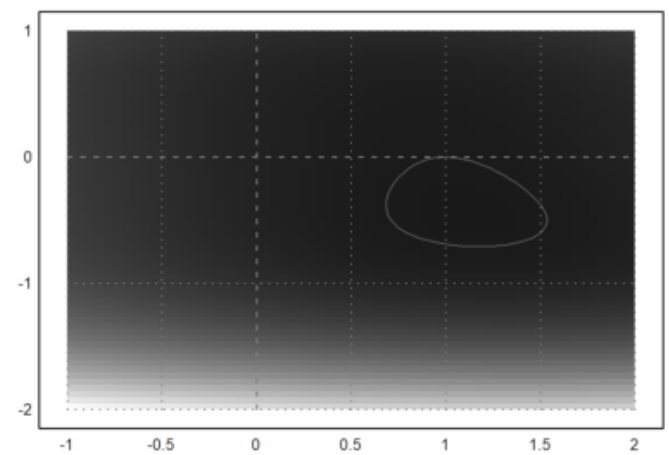


```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
```

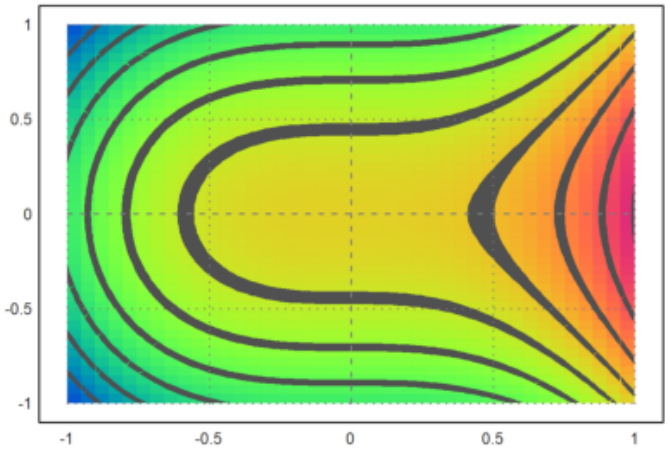


Ini berfungsi untuk plot data juga. Tetapi Anda harus menentukan rentangnya untuk label sumbu.

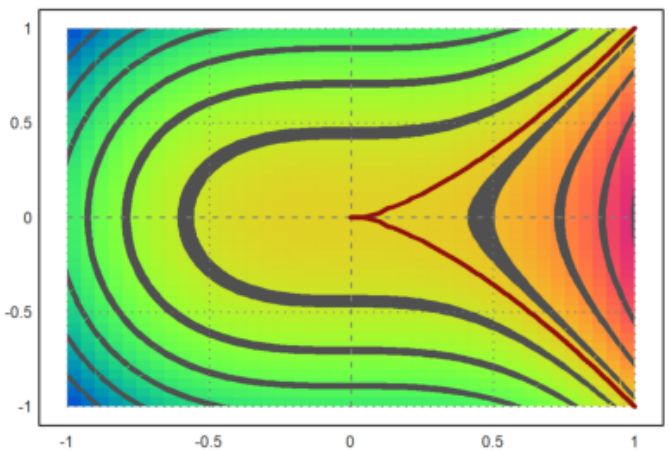
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



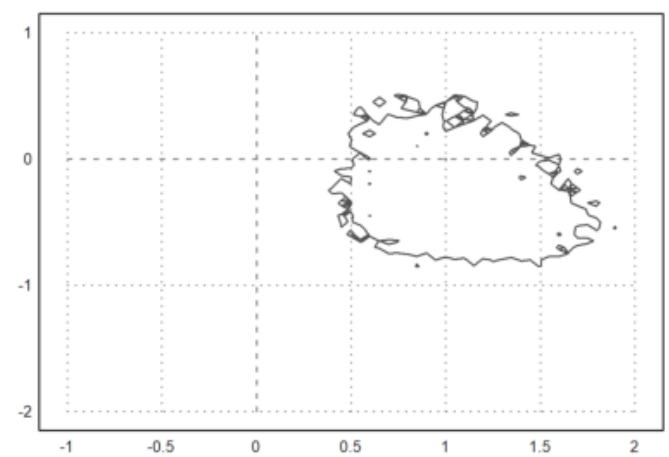

```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



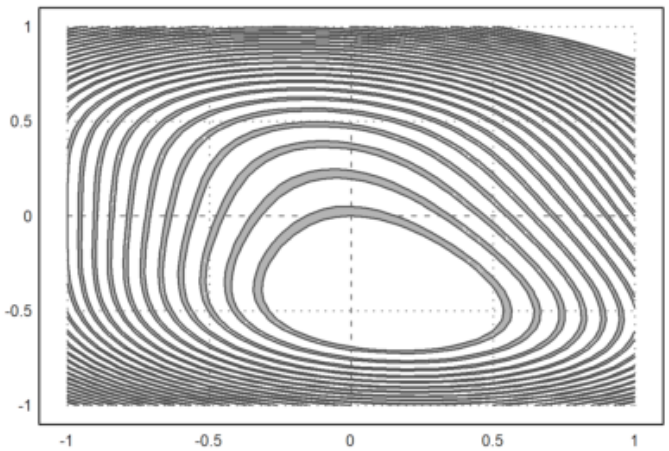

```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



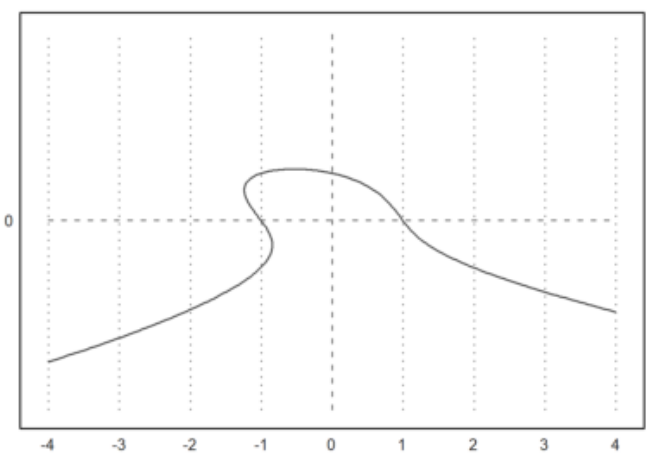

```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



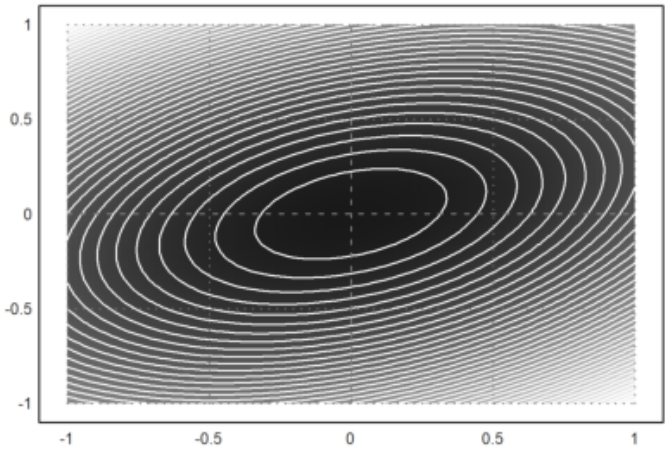
```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```




```
>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):
```

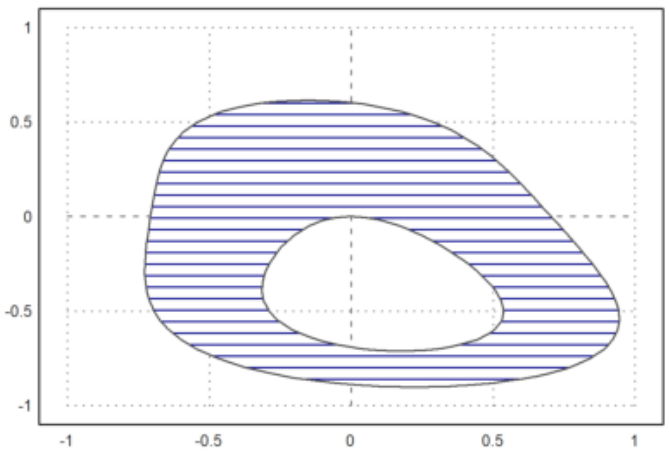


Juga dimungkinkan untuk mengisi set

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

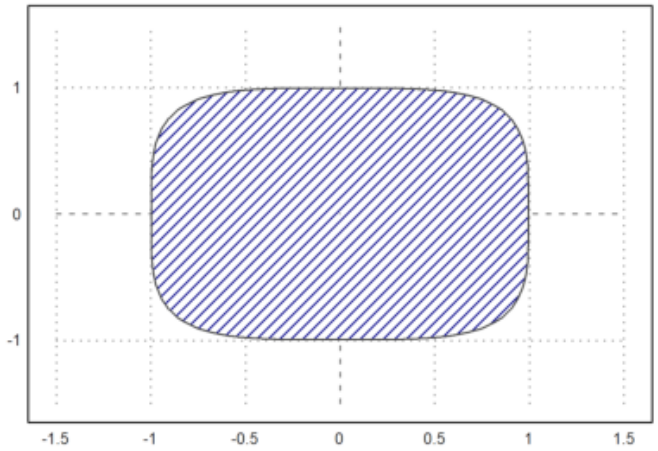
dengan rentang tingkat.
Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

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

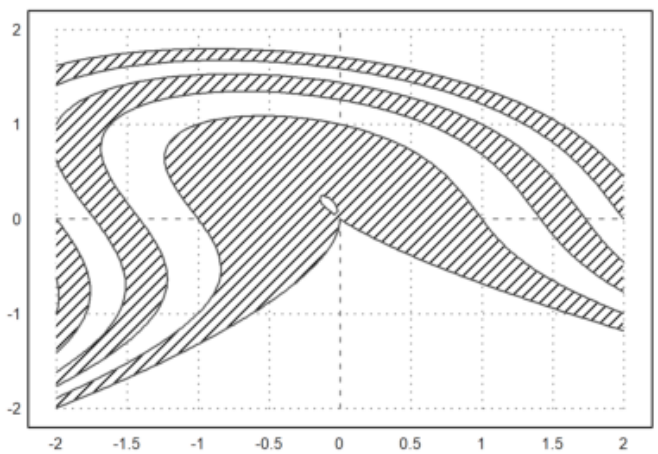


Plot implisit juga dapat menunjukkan rentang level. Kemudian level harus berupa matriks 2xn dari interval level, di mana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval. Atau, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level ke interval.

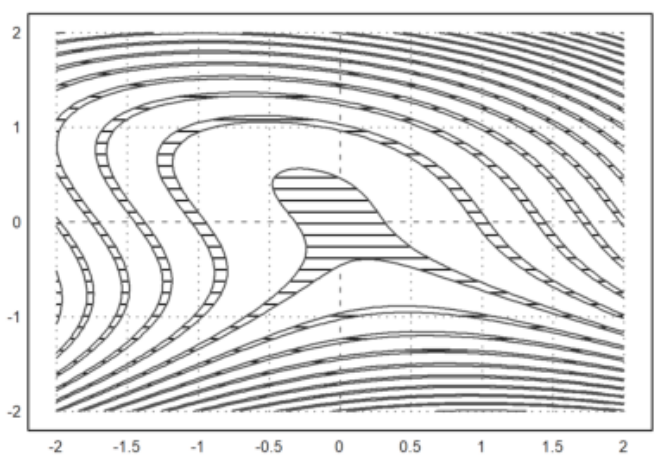
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



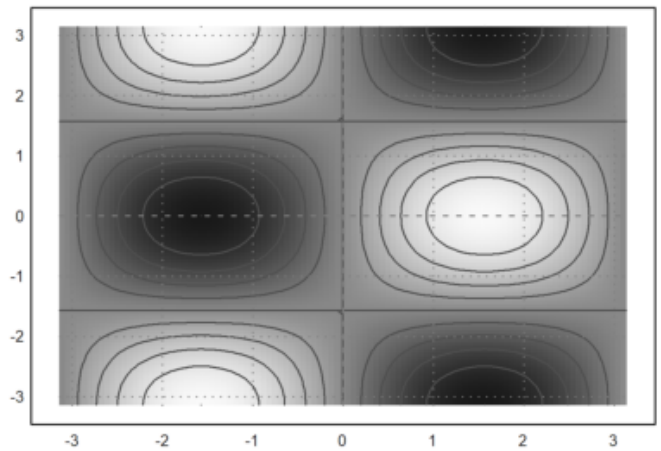

```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```




```
>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):
```

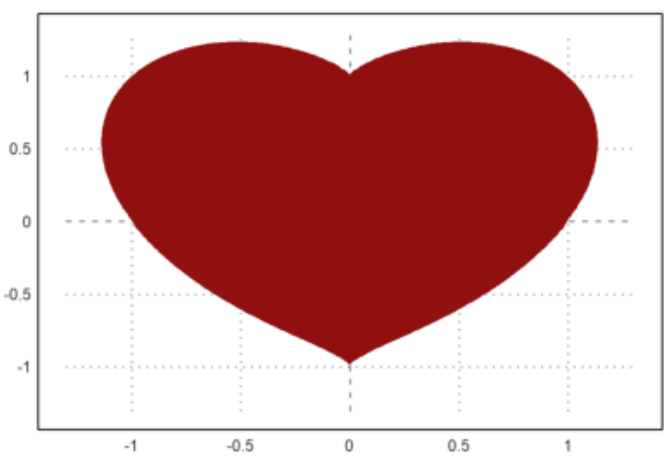


Dimungkinkan juga untuk menandai suatu wilayah

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

Ini dilakukan dengan menambahkan level dengan dua baris.

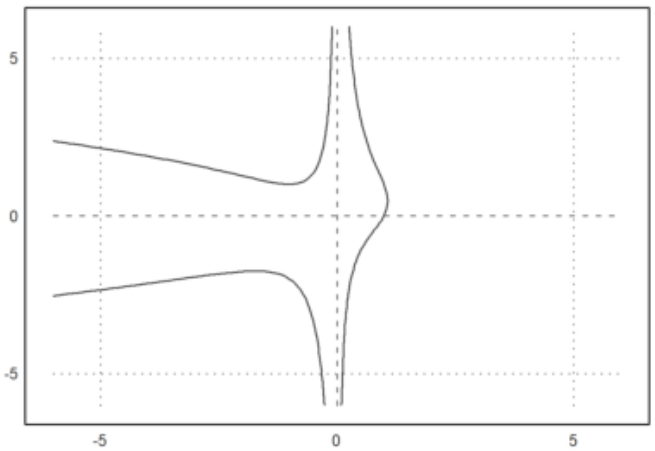
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
>  style="#",color=red,<outline, ...
>  level=[-2;0],n=100):
```



Dimungkinkan untuk menentukan level tertentu. Misalnya, kita dapat memplot 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):
```

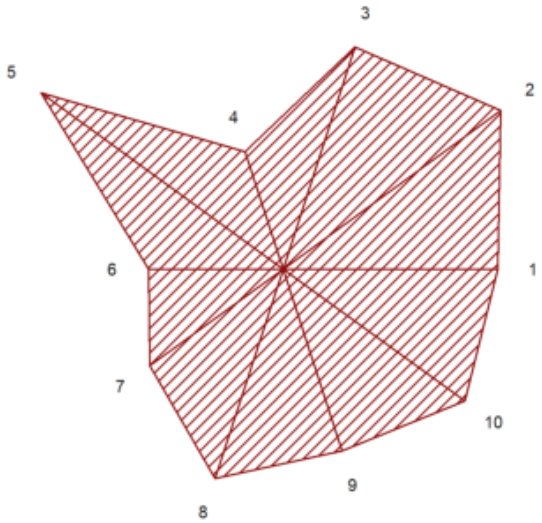



```
>function starplot1 (v, style="/", color=green, lab=none) ...
```

```
    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
```

Tidak ada kotak atau sumbu kutu di sini. Selain itu, kami menggunakan jendela penuh untuk plot.
Kami memanggil reset sebelum kami menguji plot ini untuk mengembalikan default grafis. Ini tidak perlu, jika Anda yakin plot Anda berhasil.

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



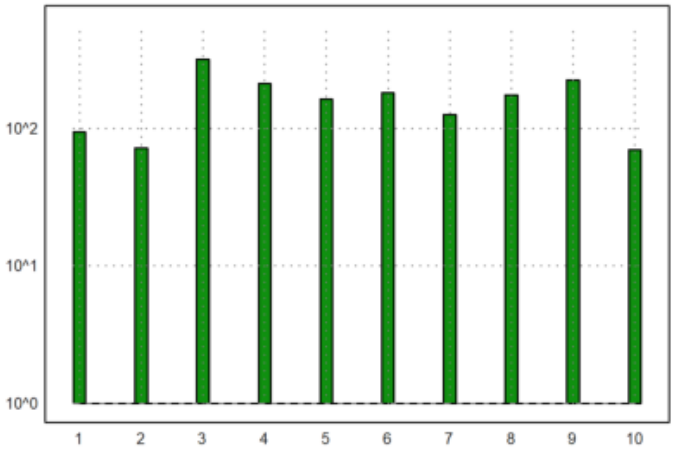
Terkadang, Anda mungkin ingin merencanakan sesuatu yang tidak dapat dilakukan plot2d, tetapi hampir. Dalam fungsi berikut, kami melakukan plot impuls logaritmik. plot2d dapat melakukan plot logaritmik, tetapi tidak untuk batang impuls.

```
>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 menganimasikan kurva 2D menggunakan plot langsung. Perintah plot(x,y) hanya memplot kurva ke jendela plot. setplot(a,b,c,d) mengatur jendela ini.
Fungsi wait(0) memaksa plot untuk muncul di jendela grafik. Jika tidak, menggambar ulang terjadi dalam 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

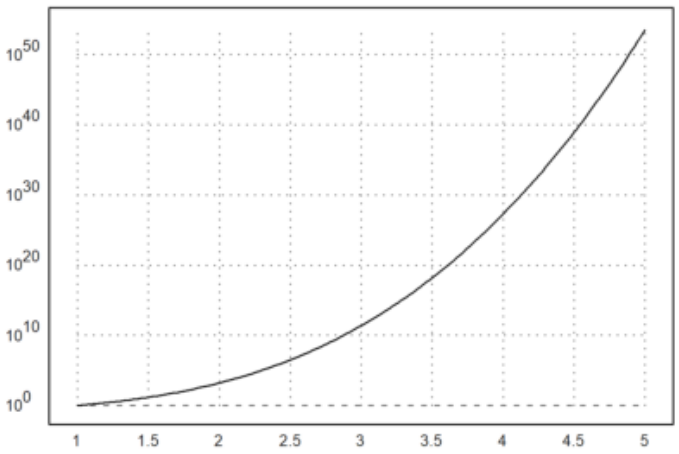
```
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

Plot Logaritmik

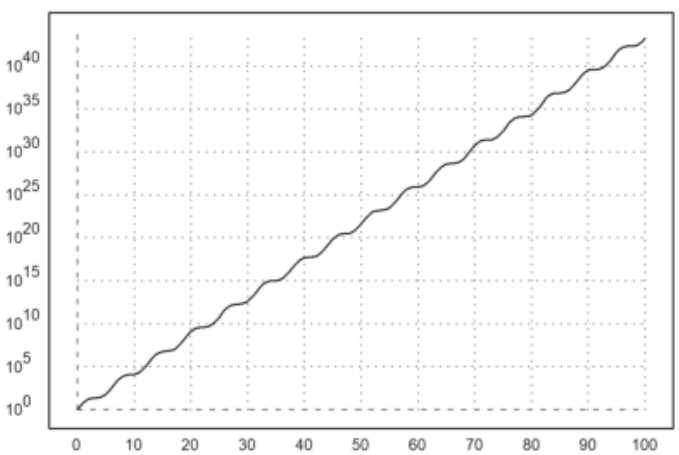
EMT menggunakan parameter "logplot" untuk skala logaritmik.
Plot logaritma dapat diplot baik menggunakan skala logaritma dalam y dengan logplot=1, atau menggunakan skala logaritma dalam x dan y dengan logplot=2, atau dalam x dengan logplot=3.

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

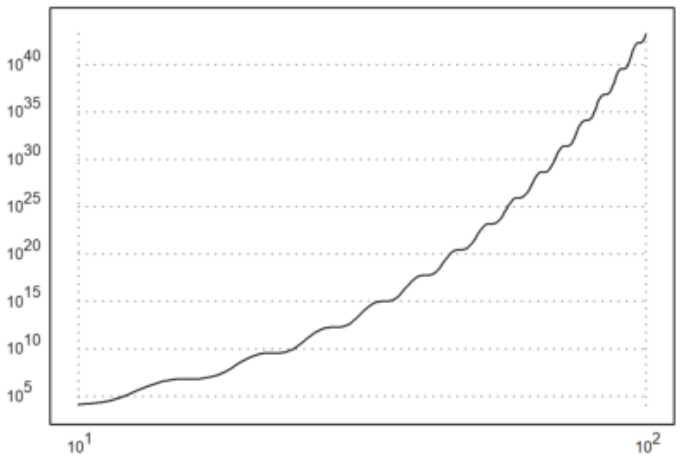
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



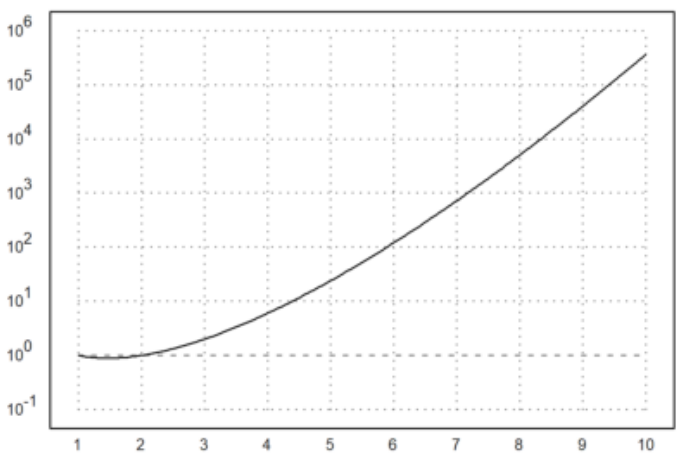

```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



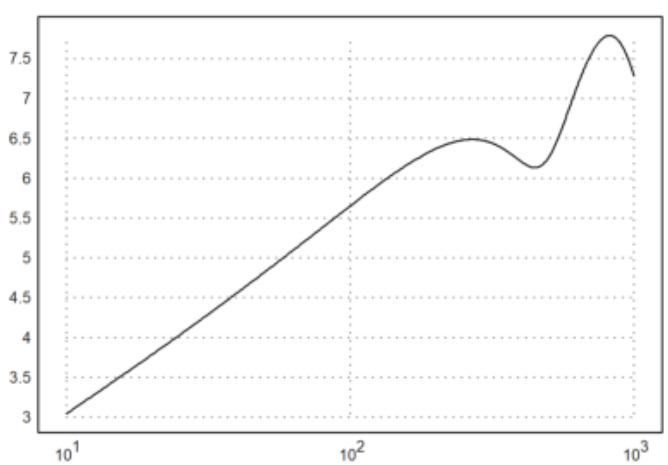

```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```




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

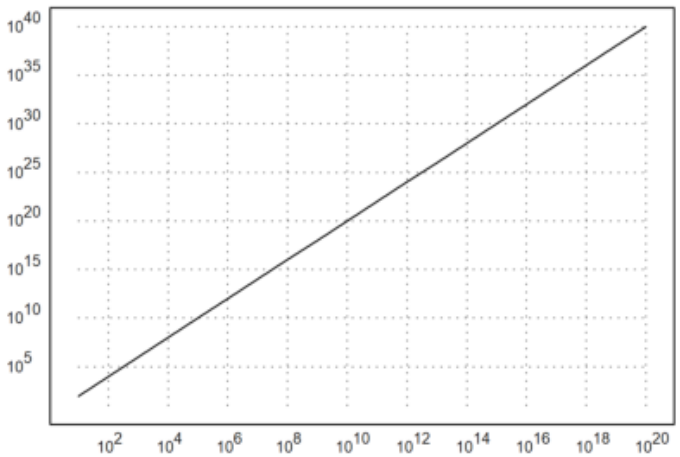



```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Ini juga berfungsi dengan 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 serbaguna untuk menggambar grafik pada bidang (grafik 2D).
ungsi ini dapat digunakan untuk:
grafik fungsi satu variabel,
plot data,
kurva di bidang,
diagram batang (bar plot),
grid bilangan kompleks,
serta plot implisit untuk fungsi dua variabel.

```
Parameters
```

x,y : persamaan, fungsi, atau vektor data
a,b,c,d : area plot (default a=-2, b=2)
r : jika r ditentukan, maka a=cx-r, b=cx+r, c=cy-r, d=cy+r
r bisa berupa vektor [rx,ry] atau [rx1,rx2,ry1,ry2].
xmin, xmax : rentang parameter untuk kurva
auto : menentukan rentang-y secara otomatis (default)
square : jika true, mencoba menjaga skala sumbu x-y tetap sama (persegi)

```
: jumlah interval (default: adaptif)
```

rid :
0 = tanpa grid dan label
1 = hanya sumbu
2 = grid normal
3 = sumbu di dalam
4 = tanpa grid
5 = grid penuh termasuk margin
6 = tanda pada bingkai
7 = hanya sumbu
8 = sumbu dengan sub-ticks
frame :
0 = tanpa bingkai
framecolor : warna bingkai dan grid
margin : nilai antara 0–0.4 untuk margin sekitar plot
color : warna kurva
jika berupa vektor, digunakan untuk tiap baris matriks plot
untuk plot titik, sebaiknya vektor kolom
thickness : ketebalan garis (bisa <1 untuk garis sangat tipis)
style : gaya plot untuk garis, marker, atau isian
titik : "[]", " <>", ".", "..", "...", "*", "+", "|", "-", "o", "" (isian), "w" (non-transparan)
garis : "-", "–", "-.", ".", ".-.", "-.-", "->"
poligon isi/bar : "", "O", "/", "\", "\ /", "+", "|", "-", "t"
points : menampilkan titik tunggal, bukan segmen garis
addpoints : jika true, tampilkan titik dan garis
add : menambahkan plot pada plot yang sudah ada
user : mengaktifkan interaksi pengguna untuk fungsi
delta : ukuran langkah untuk interaksi pengguna
bar : diagram batang (x = batas interval, y = nilai)
histogram : frekuensi data x dalam n subinterval
distribution=n : distribusi x dengan n subinterval
even : gunakan nilai antar interval untuk histogram otomatis
steps : menampilkan fungsi sebagai fungsi tangga (steps=1,2)
adaptive : plot adaptif (n = jumlah langkah minimum)
level : menampilkan garis level fungsi implisit 2 variabel
outline : menggambar batas dari area level

Jika level berupa matriks 2xn, maka rentang level akan digambar dengan warna dan gaya isi yang diberikan. Jika outline=true, maka batas digambar dengan warna kontur. Bisa dipakai untuk menandai daerah f(x,y) antara dua batas nilai.

hue : memberi warna gradien (hue) pada plot level sesuai nilai fungsi

ontour : menampilkan garis level otomatis

nc : jumlah garis level otomatis

title : judul plot (default "")

xl, yl : label sumbu x dan y

smaller : jika >0, memberi ruang lebih untuk label di sebelah kiri

ertical :
mengatur label vertikal
1 = teks vertikal
2 = label angka vertikal pada sumbu-y

filled : mengisi area plot kurva

fillcolor : warna isi untuk bar/kurva isi

logplot : menentukan jenis skala logaritmik
1 = logaritmik pada sumbu y
2 = logaritmik pada sumbu x dan y
3 = logaritmik pada sumbu x

own : string yang menunjuk ke rutin plot buatan sendiri

maps : memetakan ekspresi (0 lebih cepat), fungsi selalu dipetakan
contourcolor : warna garis kontur
contourwidth : ketebalan garis kontur
clipping : menyalakan/mematikan pemotongan (default aktif)

cgrid : jumlah garis grid untuk plot bilangan kompleks
sebaiknya kelipatan ukuran matriks - 1
bisa berupa [cx,cy]

```
Overview


Fungsi plot2d() dapat digunakan untuk menggambbar:


- ekspresi, koleksi panggilan, atau fungsi satu variabel,
- kurva parametik,
- data x terhadap y,
- fungsi implisit,
- diagram batang,
- grid kompleks,
- poligon.


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

menghitung nilainya dalam rentang yang ditentukan (a dan b, default [-2,2]). Rentang-y dihitung otomatis kecuali c dan d ditentukan, atau r digunakan (memberikan area [-r,r]).
Secara default, plot2d() memakai evaluasi adaptif. Untuk mempercepat fungsi kompleks, bisa dimatikan dengan <adaptive dan jumlah interval n dikurangi.

Selain itu, plot2d() secara default menggunakan mapping (menghitung elemen per elemen). Jika fungsi bisa menerima vektor x sekaligus, gunakan <maps agar lebih cepat.
Jika xv dan yv diberikan, maka plot2d() akan menggambar kurva dengan koordinat (xv,yv). Dalam hal ini, perlu didefinisikan rentang parameter dengan xmin, xmax. Ekspresi string harus berbentuk fungsi dari variabel parameter x.