Menggambar Grafik 2D dengan EMT

Notebook ini menjelaskan tentang cara menggambar berbagaikurva dan grafik 2D dengan software EMT. EMT menyediakan fungsi plot2d() untuk menggambar berbagai kurva dan grafik dua dimensi (2D).

Plot Dasar

Terdapat fungsi dasar untuk membuat plot. Ada koordinat layar, yang selalu berada dalam rentang 0 hingga 1024 pada setiap sumbu, tidak peduli apakah layar tersebut berbentuk persegi atau tidak. Dan ada koordinat plot, yang bisa diatur dengan setplot(). Pemetaan antara koordinat-koordinat ini bergantung pada jendela plot yang aktif saat ini. Misalnya, fungsi default shrinkwindow() memberikan ruang untuk label sumbu dan judul plot.

Pada contoh ini, kita hanya menggambar beberapa garis acak dengan berbagai warna. Untuk rincian lebih lanjut tentang fungsi-fungsi ini, pelajari fungsi-fungsi inti dari 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;
```

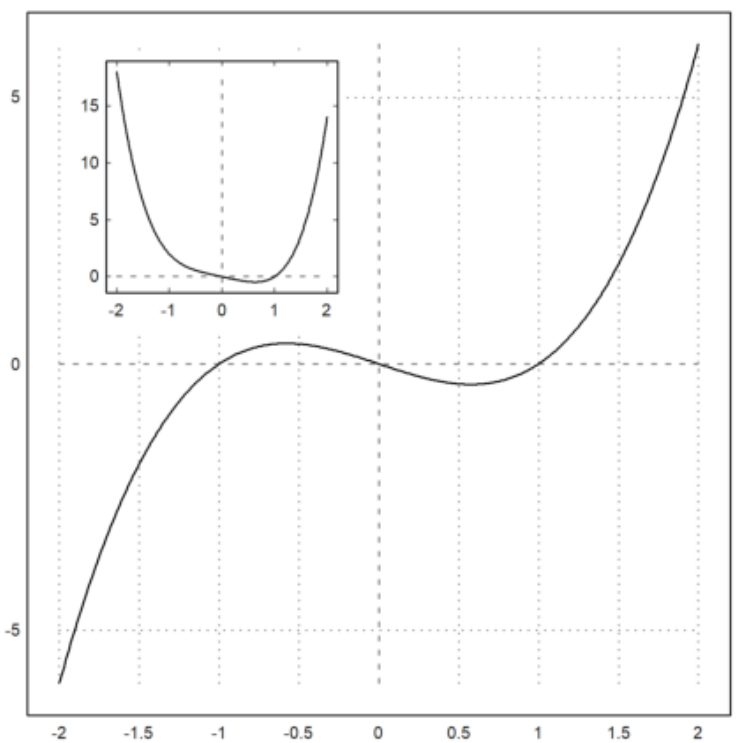
Kita perlu menahan tampilan grafik agar tetap muncul, karena perintah plot() akan menghapus grafik yang sudah ada di jendela tampilan.

Untuk menghapus semua hasil yang sudah kita buat, kita bisa menggunakan perintah 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.

Untuk contoh lainnya, kita membuat grafik kecil (inset) di dalam grafik utama. Ini dilakukan dengan menentukan jendela grafik yang lebih kecil. Perlu diingat, jendela ini tidak menyediakan ruang untuk label sumbu di luar area grafik, jadi kita perlu menambahkan margin sesuai kebutuhan. Kita juga menyimpan dan mengembalikan pengaturan jendela penuh, serta menahan grafik yang sedang aktif saat menambahkan grafik 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);
```

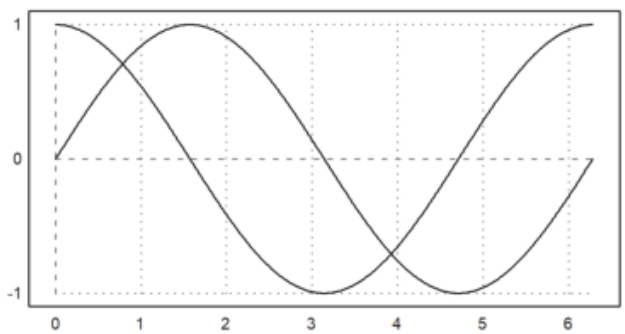
Grafik dengan beberapa gambar (multiple figures) dibuat dengan cara yang sama. Untuk itu tersedia fungsi 'figure()'.

Aspek Grafik

Secara bawaan, grafik ditampilkan dalam jendela berbentuk persegi. Kamu bisa mengubahnya dengan fungsi 'aspect()'. Jangan lupa mengembalikan pengaturan aspek ke semula setelahnya. Pengaturan bawaan ini juga bisa diubah lewat menu "Set Aspect" agar sesuai rasio tertentu atau ukuran jendela grafik saat ini.

Perubahan aspek ini juga bisa dilakukan hanya untuk satu grafik saja. Caranya dengan mengubah ukuran area grafik yang sedang aktif, lalu mengatur jendelanya supaya label-label punya cukup ruang.

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




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

Fungsi reset() mengembalikan semua pengaturan grafik ke nilai awal, termasuk rasio aspeknya.

Plot 2D di Euler

EMT Math Toolbox dapat membuat plot 2D, baik untuk data maupun fungsi, menggunakan fungsi 'plot2d'. Fungsi ini bisa menampilkan fungsi maupun data.

Kamu juga bisa memplot di Maxima menggunakan Gnuplot, atau di Python menggunakan Matplotlib.

Euler dapat membuat plot 2D untuk:

- ekspresi,
- fungsi, variabel, atau kurva parametrik,
- vektor nilai x–y,
- sebaran titik pada bidang,
- kurva implisit dengan level atau wilayah level,
- fungsi kompleks.

Gaya plot yang tersedia mencakup berbagai gaya garis dan titik, plot batang, dan plot berarsir.

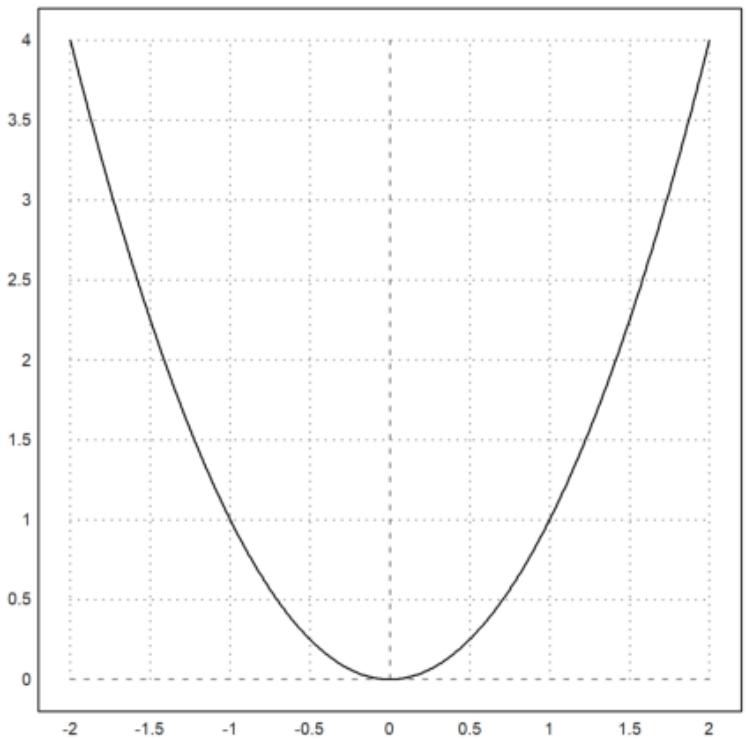
Plot Ekspresi atau Variabel

Satu ekspresi dalam “x” (misalnya “4*x^2”) atau nama fungsi (misalnya “f”) akan menghasilkan grafik fungsi tersebut.

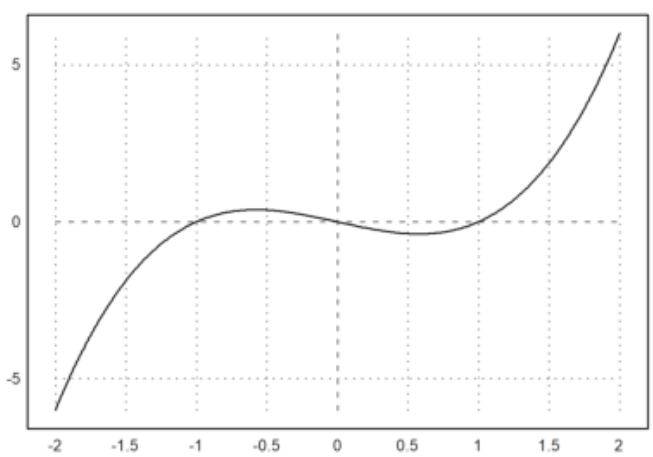
Contoh paling dasar memakai rentang default dan secara otomatis menyesuaikan rentang sumbu-y agar grafik fungsi pas terlihat.

Catatan: Jika perintah diakhiri dengan titik dua “:”, grafik akan dimasukkan ke jendela teks. Jika tidak, tekan TAB untuk melihat grafik bila jendela plot tertutup.

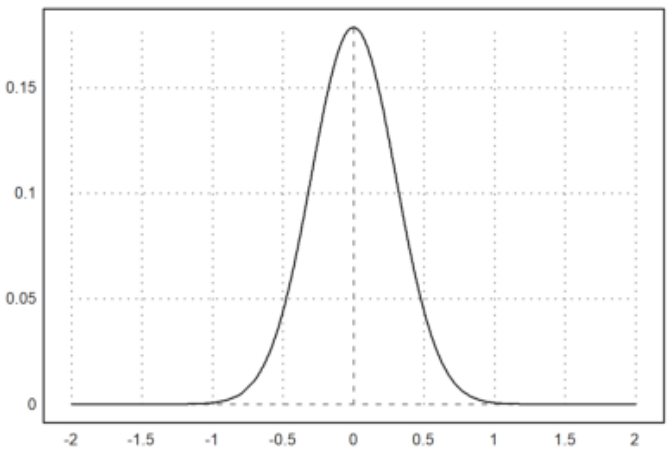
```
>plot2d("x^2"):
```




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



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

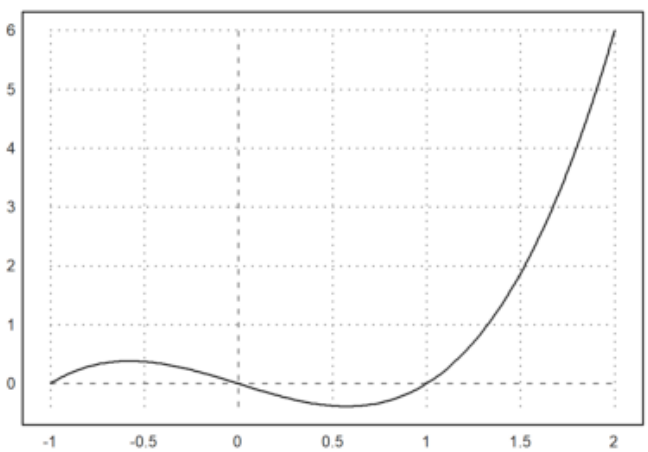


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

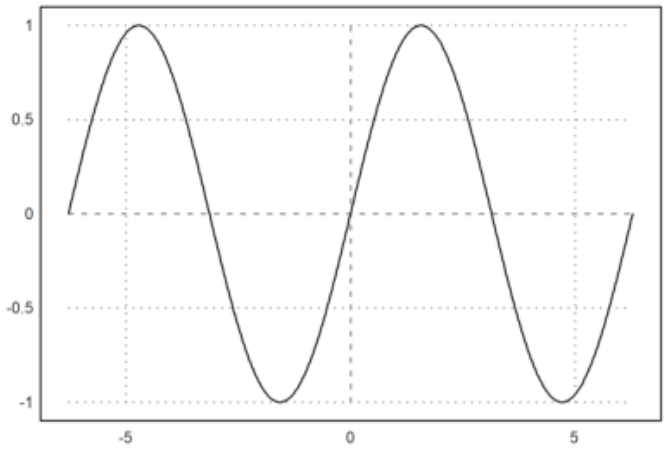
Rentang plot diatur dengan parameter yang ditentukan berikut ini:

- a,b: rentang x (default -2,2)
- c,d: rentang y (default: mengikuti skala nilai)
- r: sebagai alternatif berupa radius di sekitar pusat plot
- cx,cy: koordinat pusat plot (default 0,0)

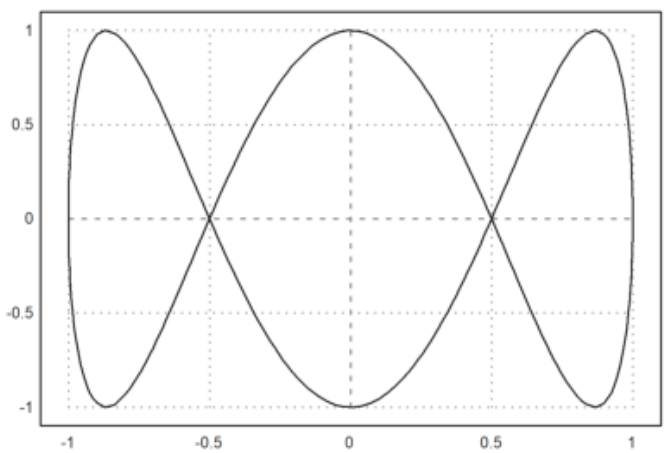
```
>plot2d("x^3-x",-1,2):
```




```
>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 lain dari penggunaan titik dua adalah perintah insimg(lines), yang akan menyisipkan grafik dengan ukuran sesuai jumlah baris teks yang ditentukan.

Dalam opsi, grafik dapat ditampilkan:

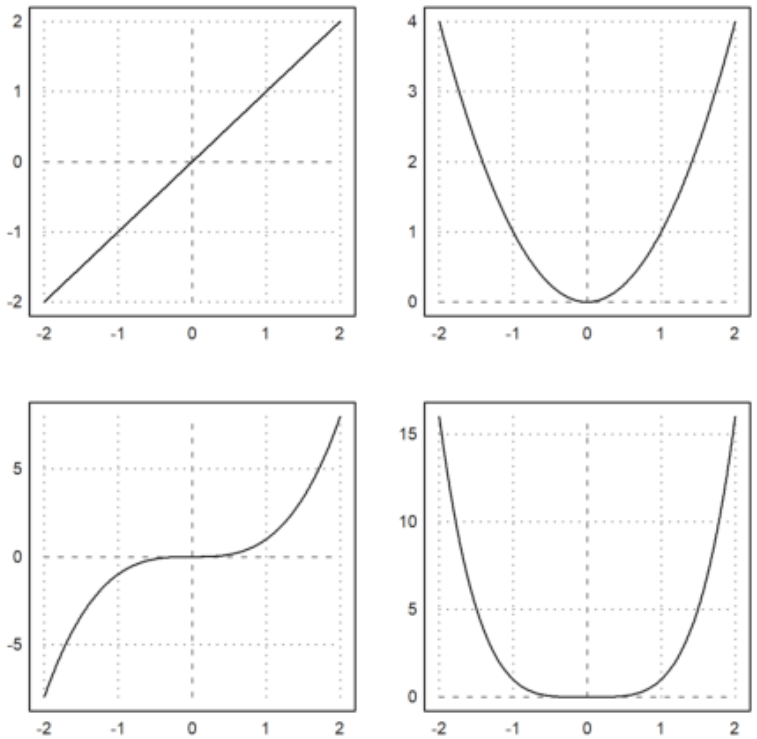
- di jendela terpisah yang bisa diubah ukurannya,
- di dalam jendela notebook.

Berbagai gaya tambahan bisa digunakan dengan perintah plot tertentu.

Jika grafik tersembunyi, tekan tombol Tab untuk menampilkannya.

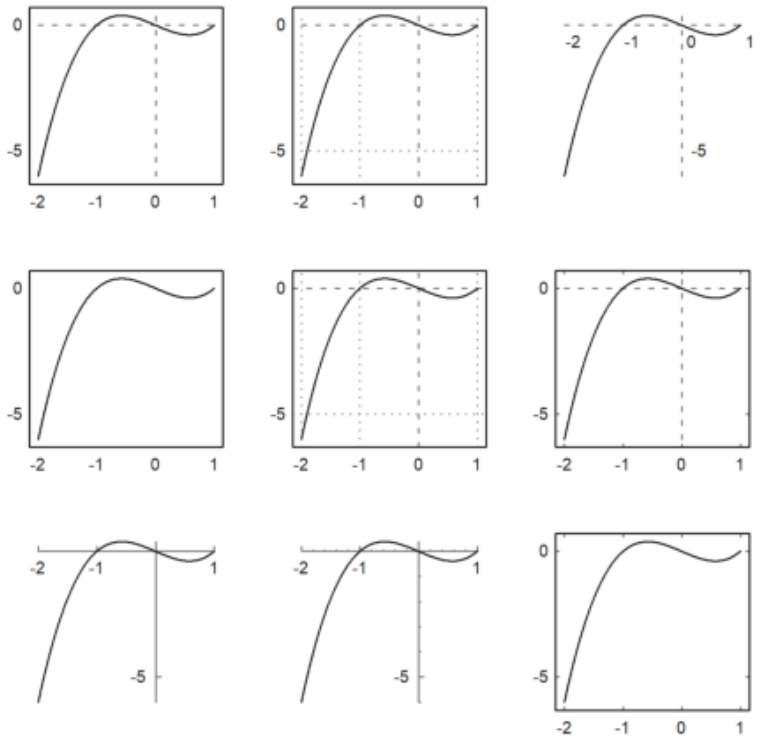
Untuk membagi jendela menjadi beberapa bagian grafik, gunakan perintah figure(). Pada contoh, grafik x^1 sampai x^4 dibagi ke dalam 4 bagian jendela. Perintah figure(0) mengembalikan jendela ke pengaturan default.

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



Dalam plot2d(), tersedia gaya alternatif dengan grid=x. Untuk gambaran umumnya, berbagai gaya grid ditampilkan dalam satu gambar (lihat bagian figure() di bawah untuk perintahnya). Gaya grid=0 tidak disertakan karena tidak menampilkan grid maupun bingkai.

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

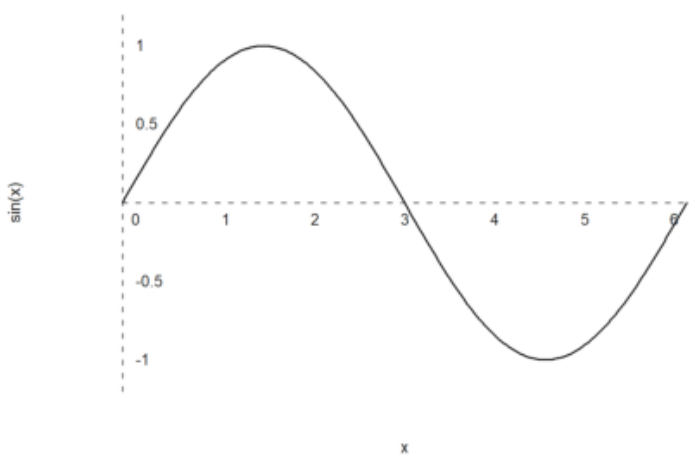


Jika argumen untuk plot2d() berupa sebuah ekspresi diikuti empat angka, maka angka-angka tersebut adalah rentang x dan y untuk plot.

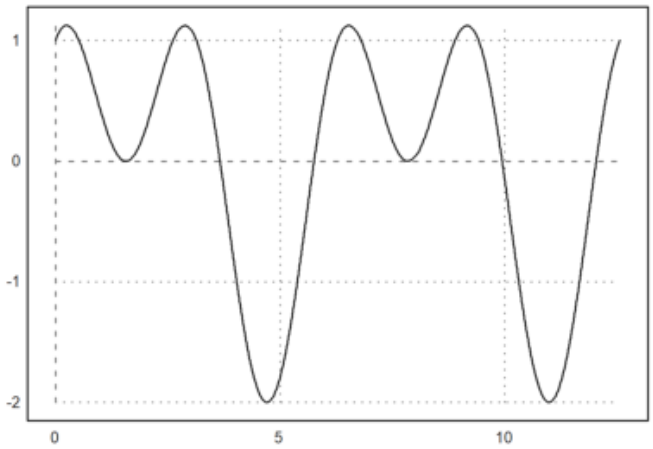
Sebagai alternatif, a, b, c, d dapat ditentukan sebagai parameter yang diberi nilai, misalnya a=… dan seterusnya.

Pada 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):
```

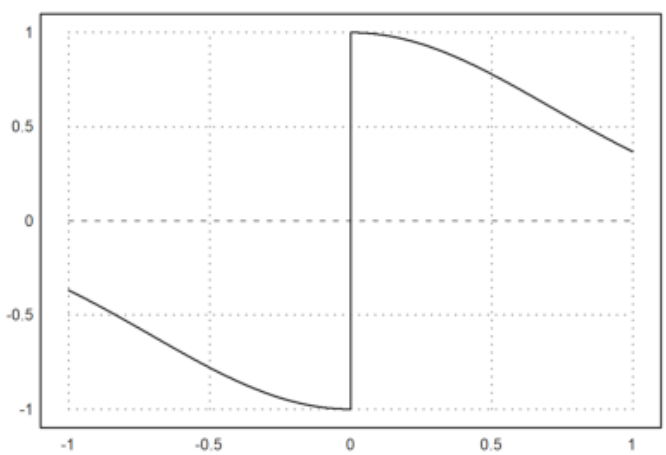


Gambar yang dihasilkan dengan menyisipkan plot ke dalam jendela teks secara otomatis disimpan di direktori yang sama dengan notebook, secara default di subdirektori bernama “images”. Gambar-gambar ini juga digunakan saat ekspor ke HTML.

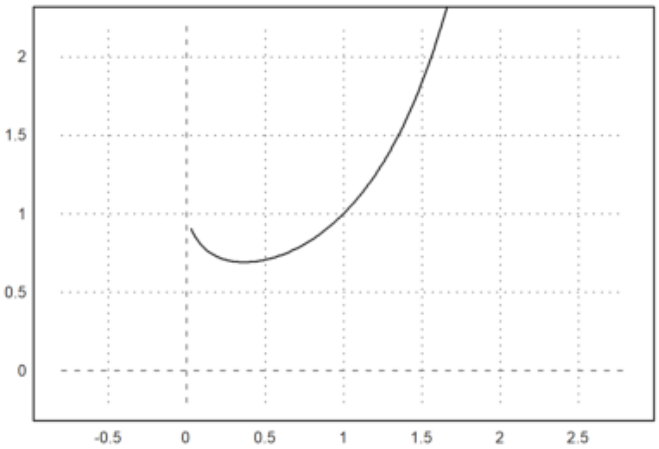
Kamu bisa menandai gambar apa pun lalu menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, grafik yang sedang aktif juga bisa diekspor menggunakan fungsi yang ada di menu File.

Fungsi atau ekspresi dalam plot2d dievaluasi secara adaptif. Untuk kecepatan lebih tinggi, mode adaptive plot bisa dimatikan dengan \<adaptive dan jumlah subinterval ditentukan dengan n=... Langkah ini biasanya hanya diperlukan pada kasus-kasus tertentu saja.

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

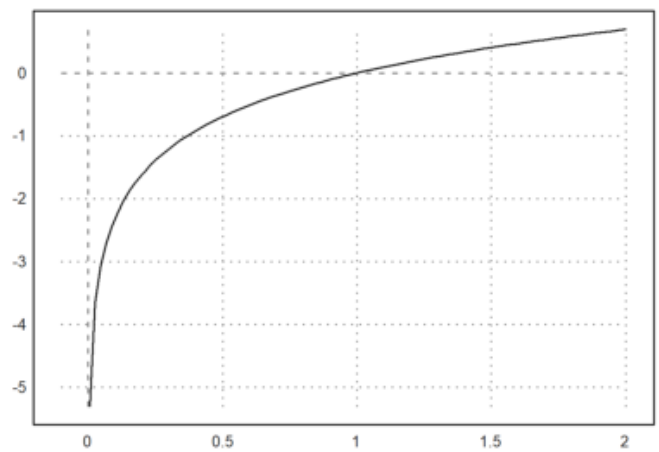



```
>plot2d("x^x",r=1.2,cx=1,cy=1):
```



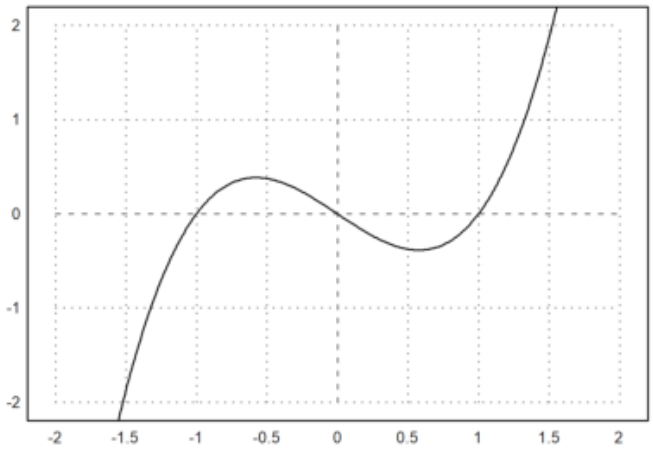
Perlu diperhatikan bahwa x^x tidak terdefinisi untuk x <= 0. Fungsi plot2d akan menangkap kesalahan ini dan mulai memplot begitu fungsi tersebut sudah terdefinisi. Mekanisme ini berlaku untuk semua fungsi yang menghasilkan NAN di luar daerah definisinya.

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

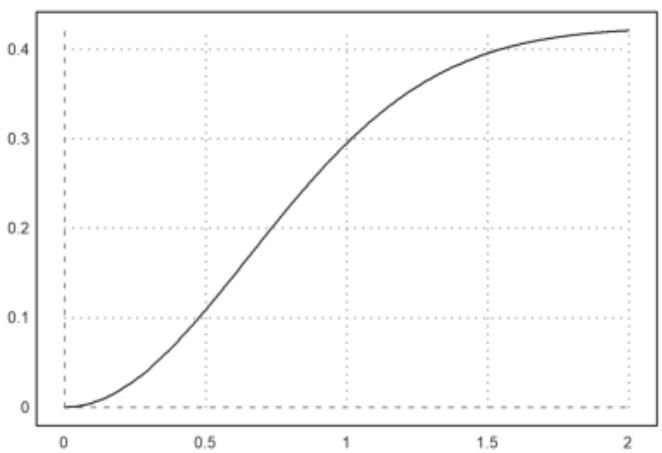


Parameter square=true (atau >square) akan otomatis memilih rentang y sehingga hasilnya berupa jendela plot berbentuk persegi. Perlu dicatat bahwa secara default Euler memang menggunakan ruang berbentuk persegi di dalam jendela plot.

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

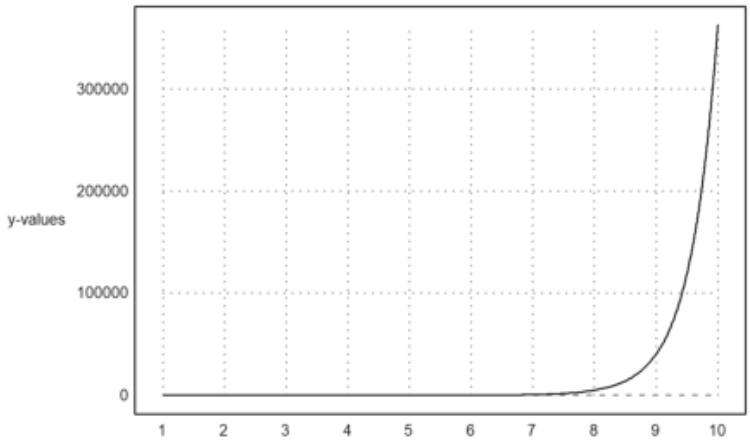



```
>plot2d(''integrate("sin(x)*exp(-x^2)",0,x)'',0,2): // plot integral
```



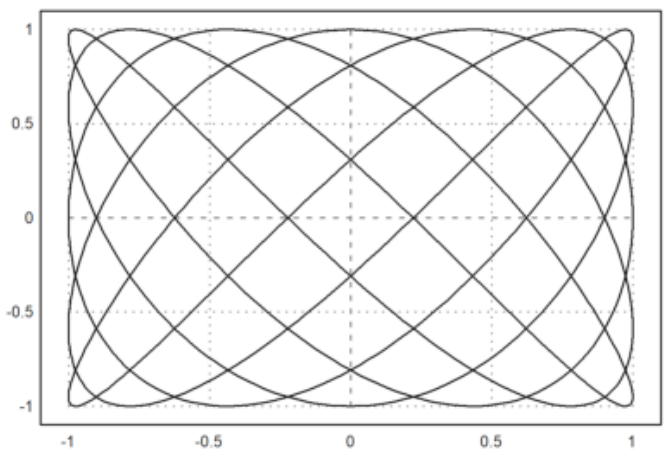
Jika kamu memerlukan lebih banyak ruang untuk label sumbu-y, panggil shrinkwindow() dengan parameter yang lebih kecil, atau atur nilai positif untuk “smaller” di plot2d().

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

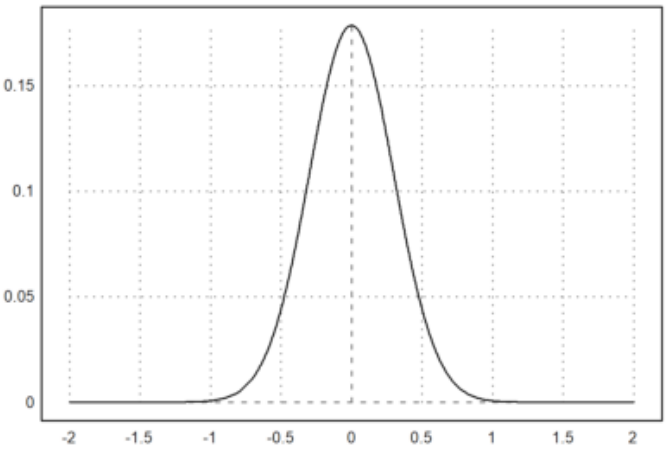


Ekspresi simbolik juga bisa digunakan, karena disimpan sebagai ekspresi string sederhana.

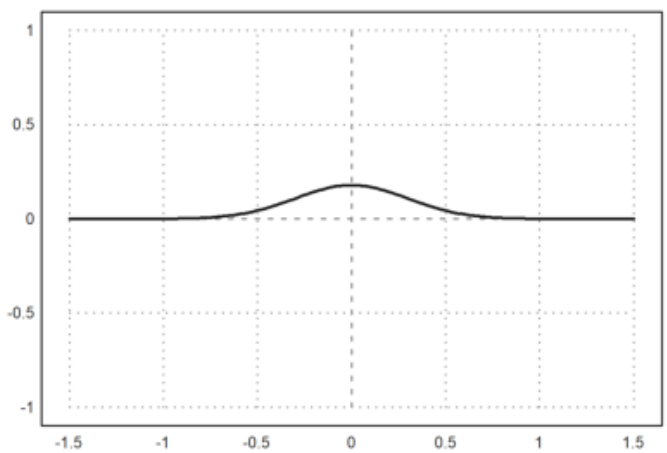
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



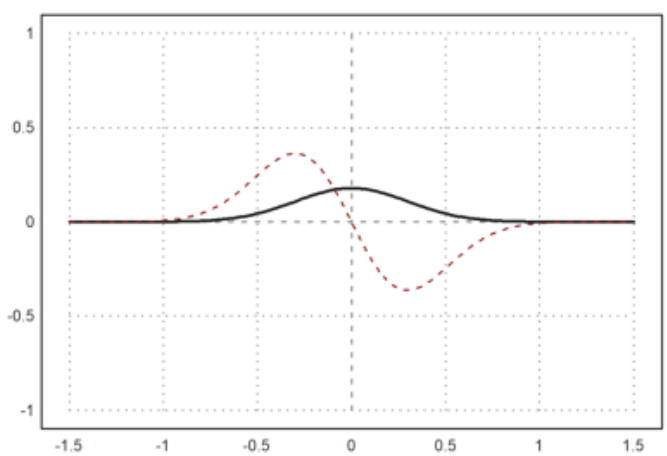

```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression
>plot2d(expr,-2,2): // plot from -2 to 2
```



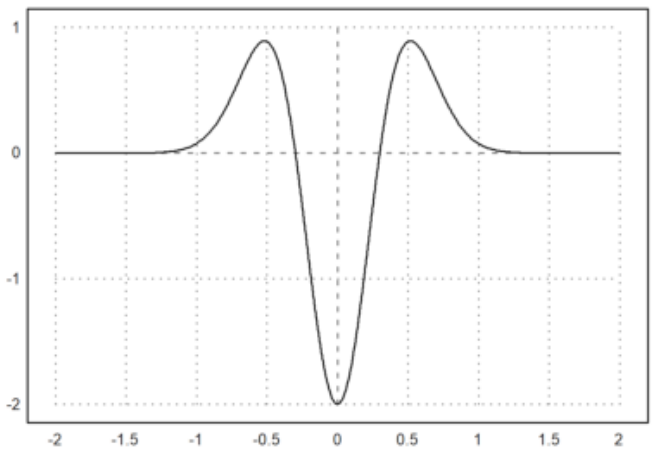

```
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)
```



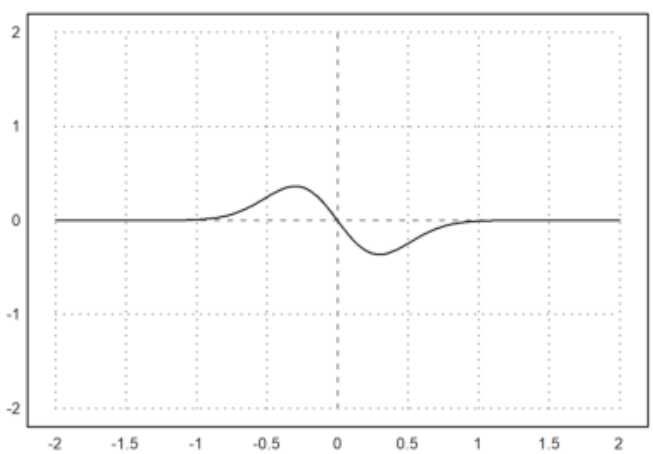

```
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot
```



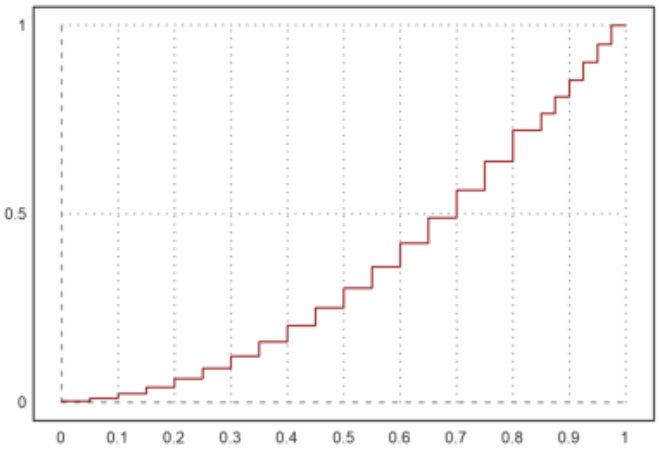
```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle
```



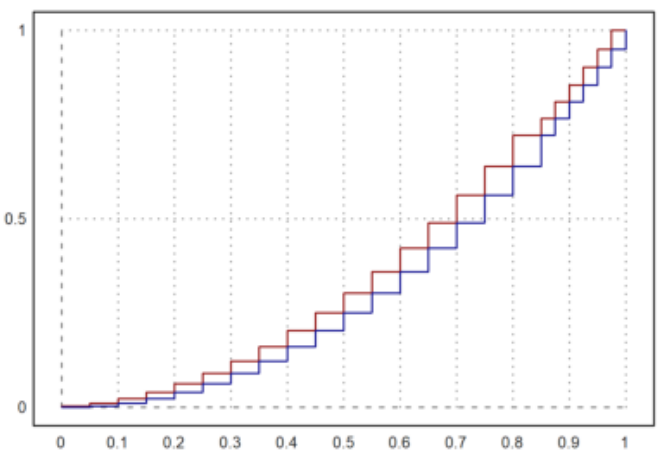

```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square
```




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




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

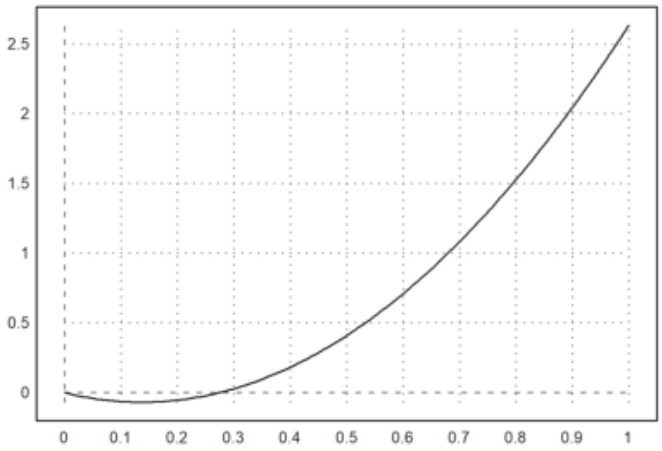


Fungsi dengan Satu Parameter

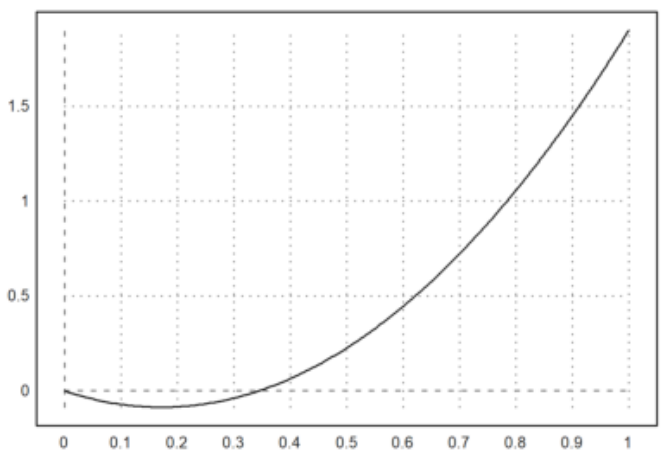
Fungsi plotting terpenting untuk plot bidang datar adalah plot2d(). Fungsi ini diimplementasikan dalam bahasa Euler di file "plot.e", yang dimuat saat program dijalankan.

Berikut beberapa contoh penggunaan fungsi. Seperti biasa di EMT, untuk fungsi yang bekerja pada fungsi lain atau ekspresi, kamu bisa meneruskan parameter tambahan (selain x) yang bukan variabel global ke fungsi tersebut dengan parameter yang dipisahkan titik koma atau dengan call collection.

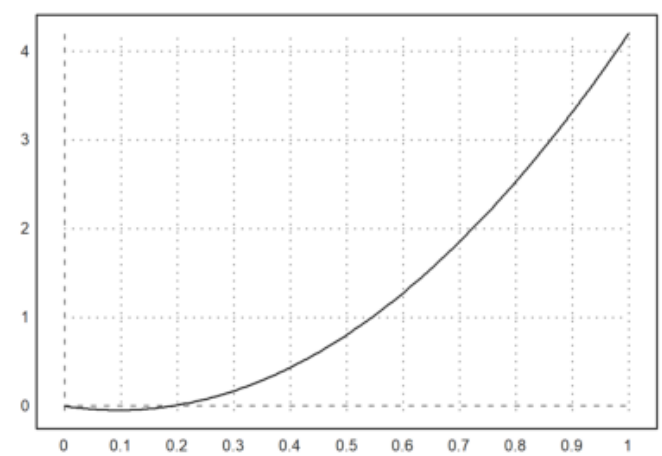
```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
>a=0.3; plot2d("f",0,1;a): // plot with a=0.3
```



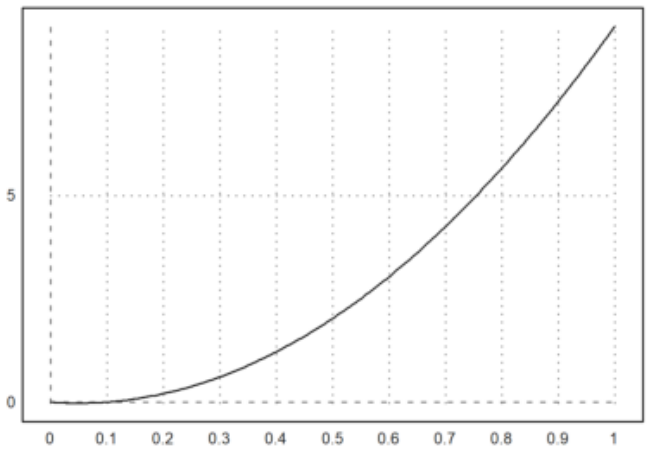

```
>plot2d("f",0,1;0.4): // plot with a=0.4
```



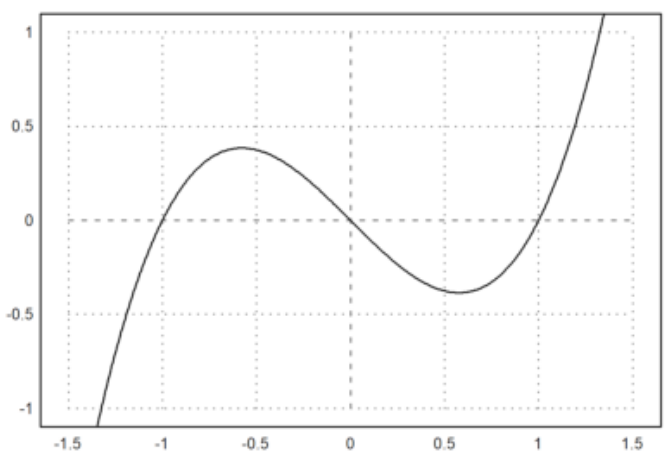

```
>plot2d({{"f",0.2}},0,1): // plot with a=0.2
```



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




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



Berikut ringkasan fungsi yang diterima:

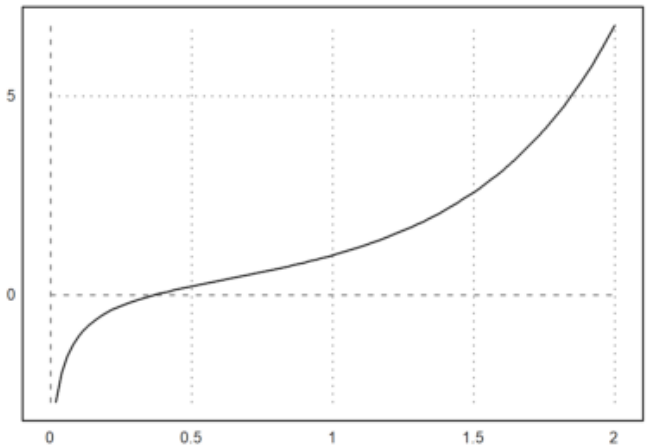
- ekspresi atau ekspresi simbolik dalam x
- fungsi atau fungsi simbolik dengan nama seperti "f"
- fungsi simbolik cukup dengan menyebut namanya saja, misalnya f

Fungsi plot2d() juga menerima fungsi simbolik. Untuk fungsi simbolik, cukup dengan menuliskan namanya saja.

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

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

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

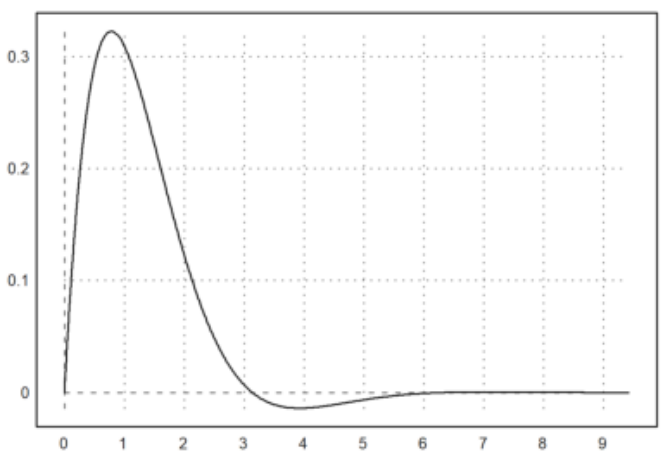


Untuk ekspresi atau ekspresi simbolik, cukup menuliskan nama variabelnya saja untuk memplotnya.

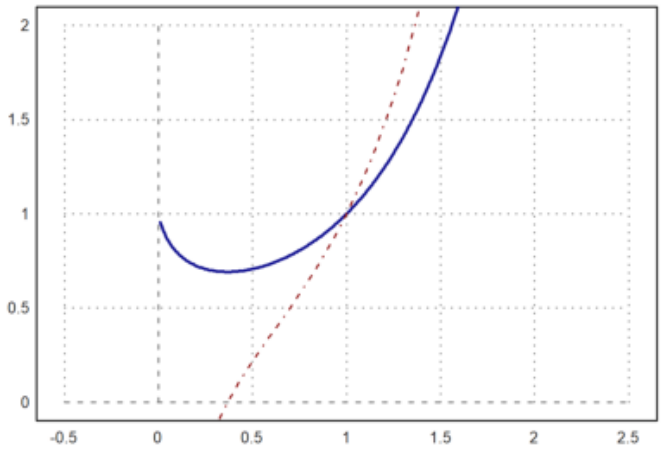
```
>expr &= sin(x)*exp(-x)
```

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

```
>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="-.-"):
```



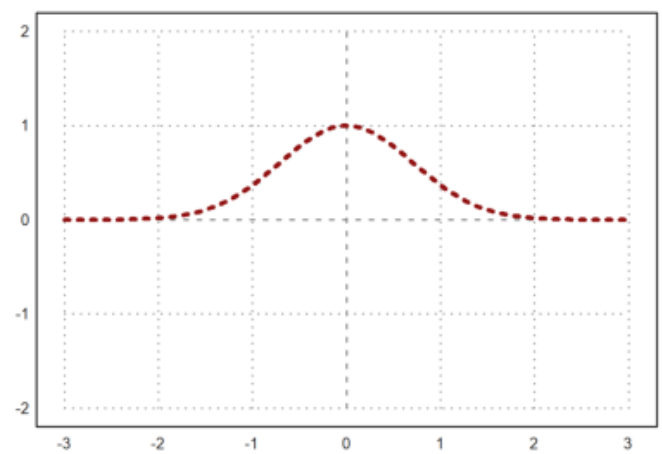
Untuk gaya garis tersedia berbagai pilihan:

- style="...". Pilih dari "-", "–", "-.", ".", ".-.", "-.-".
- color: Lihat bagian warna di bawah.
- thickness: Ketebalan garis, default = 1.

Warna bisa dipilih dari warna bawaan atau dengan format RGB.

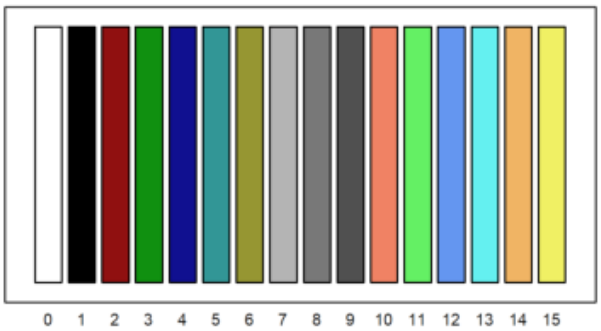
- 0..15: indeks warna default.
- konstanta warna: white, black, red, green, blue, cyan, olive, lightgray, gray, darkgray, orange, lightgreen, turquoise, lightblue, lightorange, yellow
- rgb(red,green,blue): parameternya berupa bilangan riil antara \[0,1].

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



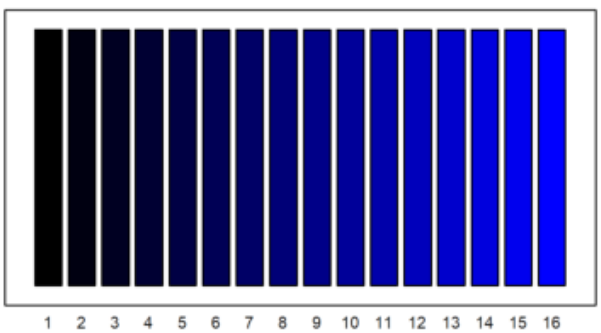
Berikut ini adalah tampilan warna bawaan (predefined colors) yang sudah tersedia di EMT.

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



Tapi, kamu bisa menggunakan warna apapun.

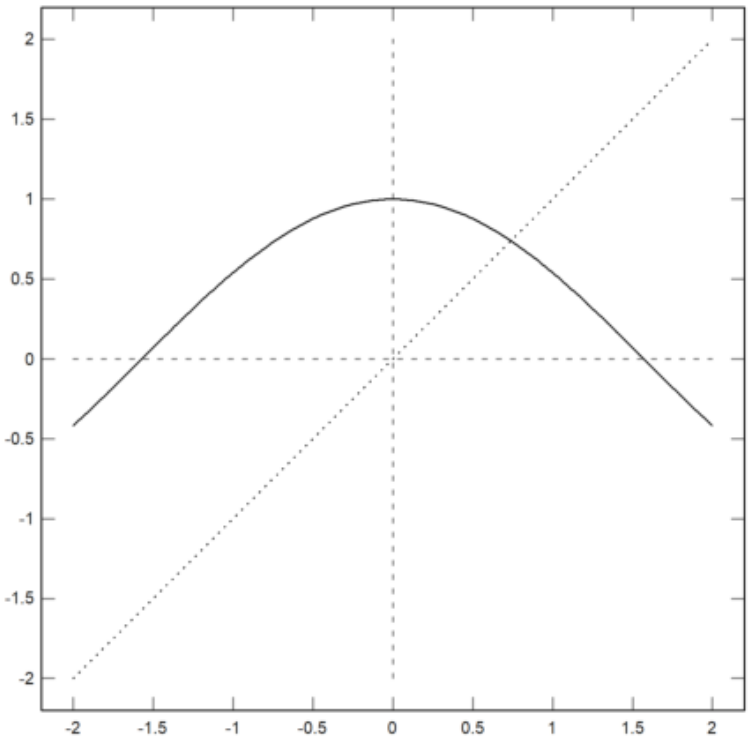
```
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```



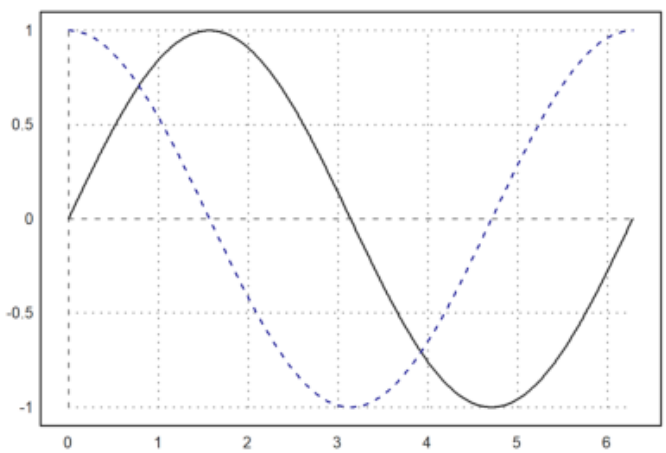
Menggambar Beberapa Kurva pada bidang koordinat yang sama

Memplot lebih dari satu fungsi (multiple functions) dalam satu jendela dapat dilakukan dengan beberapa cara. Salah satu caranya adalah menggunakan >add untuk beberapa pemanggilan plot2d pada semua pemanggilan kecuali pemanggilan pertama. Fitur ini sudah digunakan pada contoh-contoh di atas.

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

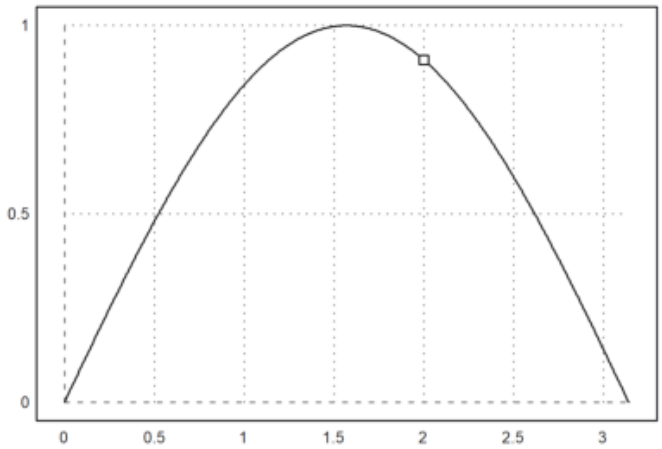



```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



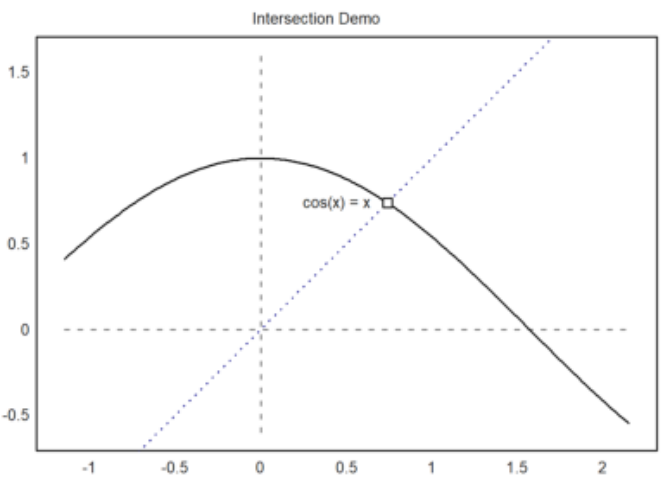
Salah satu kegunaan >add adalah untuk menambahkan titik pada kurva.

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



Kita menambahkan titik potong beserta labelnya (pada posisi "cl" untuk center left), lalu menyisipkan hasilnya ke dalam notebook. Kita juga menambahkan judul pada plot tersebut.

```
>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):
```

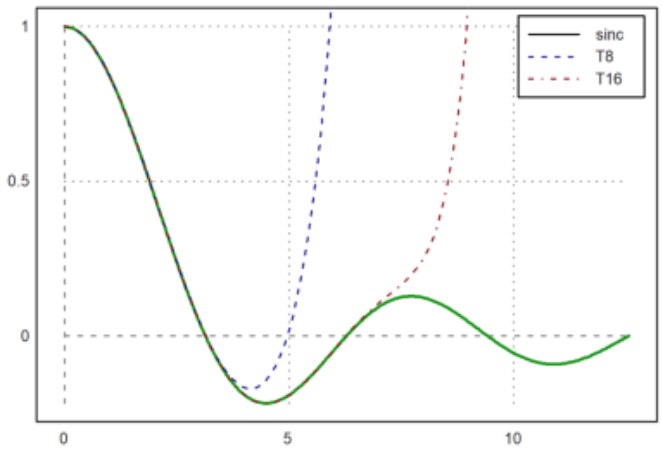


Dalam demo berikut, kita memplot fungsi sinc(x)=sin(x)/x beserta ekspansi Taylor orde ke-8 dan ke-16. Ekspansi ini dihitung menggunakan Maxima melalui ekspresi simbolik.
Plot ini dibuat dengan perintah multi-baris yang berisi tiga pemanggilan plot2d(). Pemanggilan kedua dan ketiga menggunakan flag >add, sehingga grafiknya memakai rentang dari pemanggilan sebelumnya.
Kita juga menambahkan kotak label yang menjelaskan fungsi-fungsi tersebut.

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

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

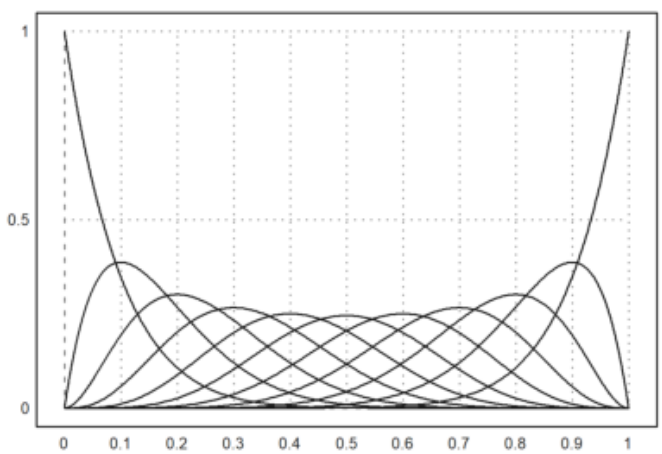
```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
>  plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
>  plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-.-"); ...
>  labelbox(["sinc","T8","T16"],styles=["-","--","-.-"], ...
>    colors=[black,blue,red]):
```



Dalam contoh berikut, kita membangkitkan Polinomial Bernstein.

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

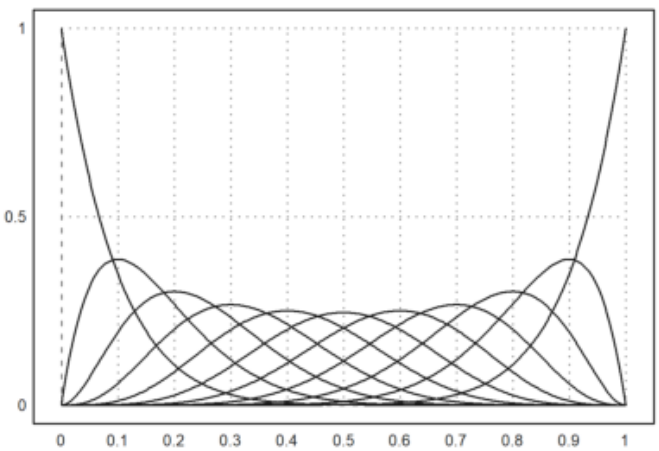
```
>plot2d("(1-x)^10",0,1); // plot first function
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;
>insimg;
```



Metode kedua adalah menggunakan sepasang matriks x-values dan matriks y-values dengan ukuran yang sama.

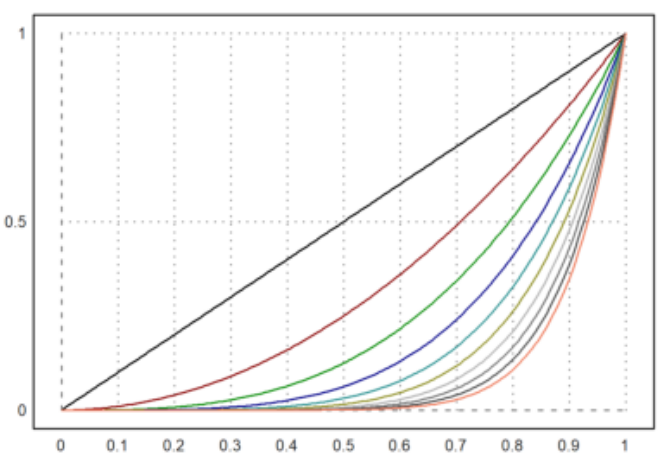
Kita membangkitkan sebuah matriks nilai yang setiap barisnya berisi satu Polinomial Bernstein. Untuk ini, kita cukup memakai vektor kolom i. Lihat kembali bagian pengantar tentang bahasa matriks untuk mempelajari detail lebih lanjut.

```
>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):
```



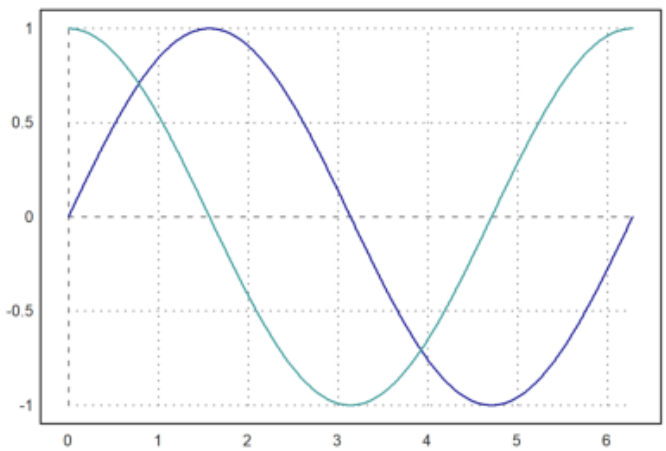
Perlu diperhatikan bahwa parameter color bisa berupa sebuah vektor. Dengan begitu, setiap warna akan digunakan untuk masing-masing baris pada matriks.

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

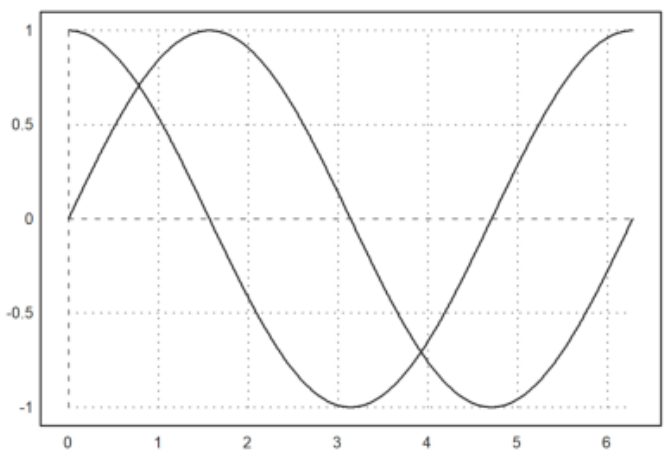


Metode lain adalah menggunakan sebuah vektor berisi ekspresi (string). Dengan cara ini, kamu bisa memakai array warna, array gaya, dan array ketebalan yang panjangnya sama.

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




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



Kita bisa mendapatkan vektor seperti itu dari Maxima dengan menggunakan makelist() dan mxm2str().

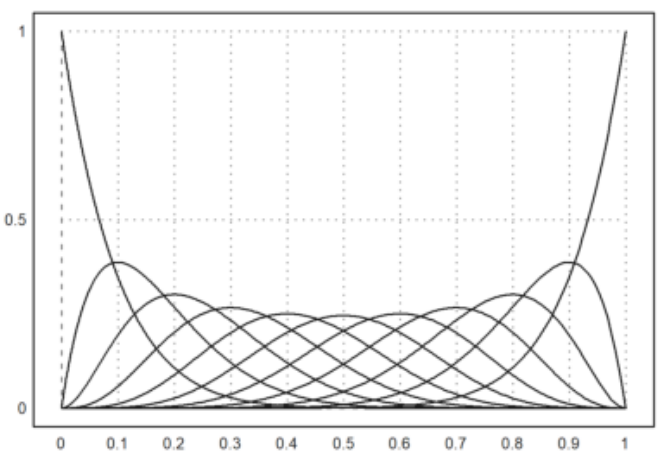
```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // make list
```

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

```
>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
```

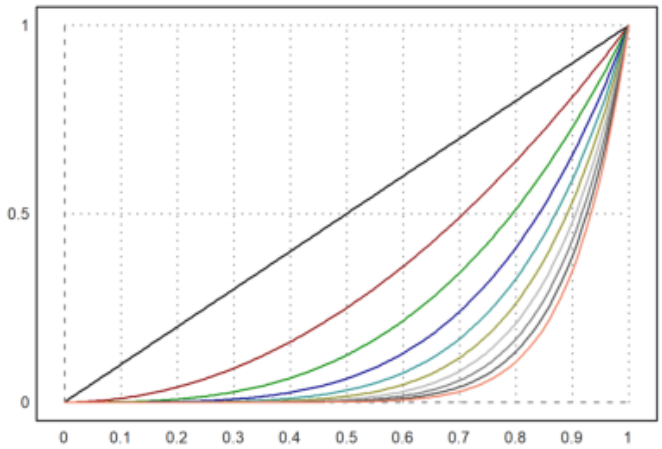


Alternatif lain adalah menggunakan bahasa matriks Euler.

Jika sebuah ekspresi menghasilkan matriks fungsi dengan satu fungsi di setiap baris, maka semua fungsi tersebut akan diplot dalam satu grafik.

Untuk ini, gunakan vektor parameter dalam bentuk vektor kolom. Jika ditambahkan array warna, maka warna tersebut akan dipakai untuk masing-masing baris pada plot.

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

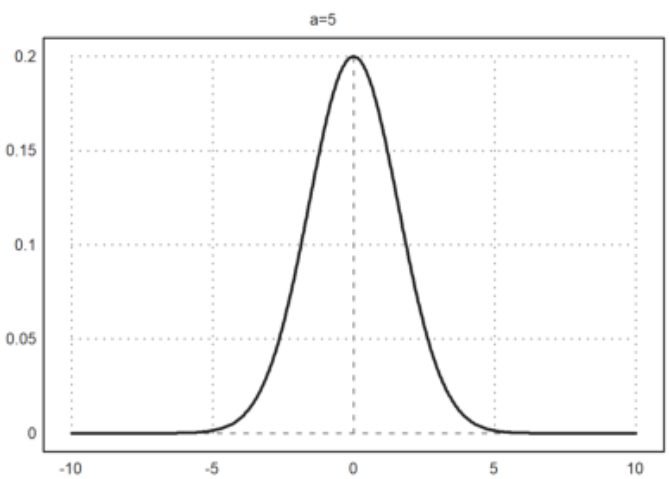


Ekspresi dan fungsi satu baris dapat melihat variabel global.

Jika kamu tidak bisa menggunakan variabel global, maka kamu perlu memakai fungsi dengan parameter tambahan, dan meneruskan parameter ini sebagai parameter setelah titik koma.

Perhatikan untuk meletakkan semua parameter yang diberi nilai di bagian akhir perintah plot2d. Pada contoh, kita meneruskan a=5 ke fungsi f yang kita plot dari -10 sampai 10.

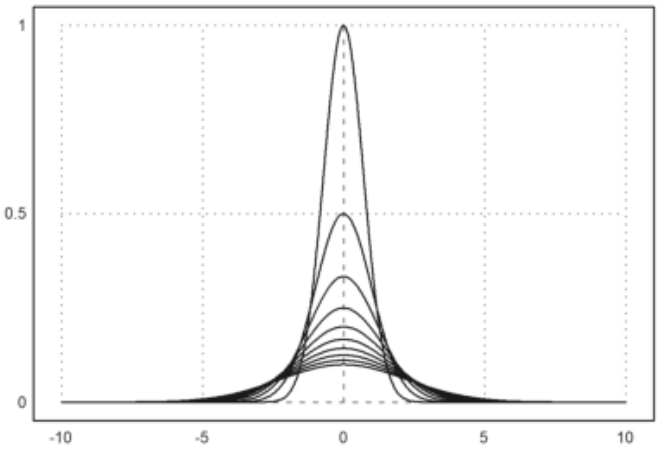
```
>function f(x,a) := 1/a*exp(-x^2/a); ...
>plot2d("f",-10,10;5,thickness=2,title="a=5"):
```



Sebagai alternatif, gunakan sebuah koleksi berisi nama fungsi dan semua parameter tambahan. Daftar khusus ini disebut call collections, dan inilah cara yang lebih disarankan untuk meneruskan argumen ke sebuah fungsi yang sendiri akan diteruskan sebagai argumen ke fungsi lain.

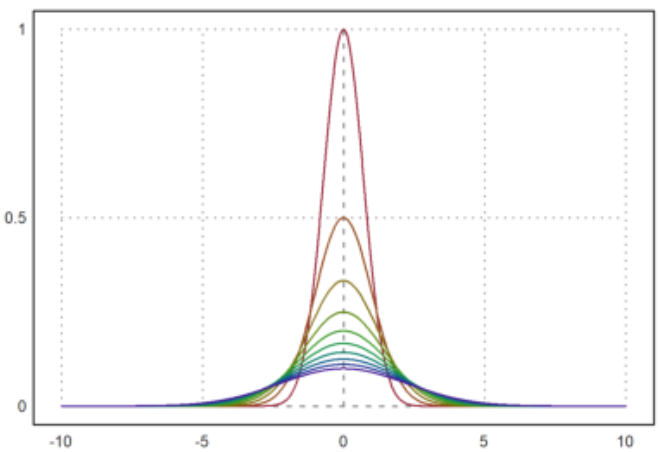
Pada contoh berikut, kita menggunakan sebuah loop untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman untuk loop).

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



Kita bisa mendapatkan hasil yang sama dengan cara berikut menggunakan bahasa matriks EMT. Setiap baris dari matriks f(x,a) merupakan satu fungsi. Selain itu, kita juga bisa mengatur warna untuk setiap baris pada matriks tersebut. Klik ganda pada fungsi getspectral() untuk penjelasannya.

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



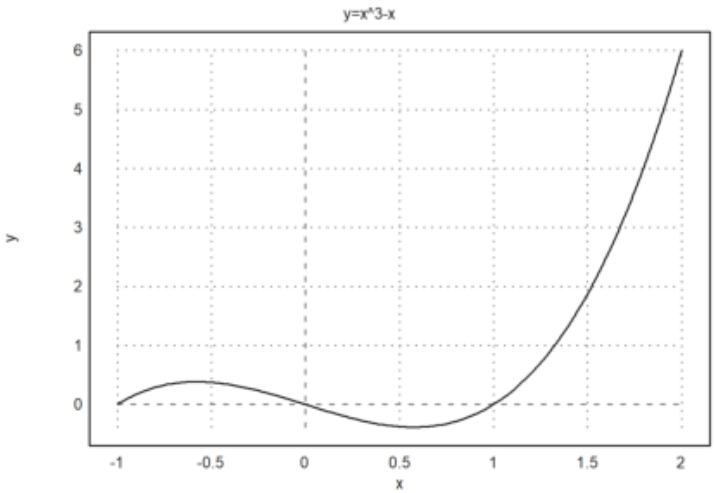
Teks Label

Hiasan sederhana yang bisa ditambahkan:

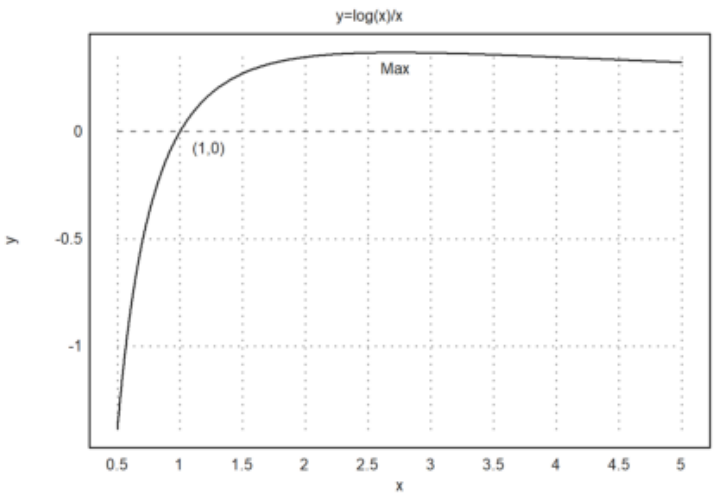
- judul dengan title="..."
- label sumbu x dan y dengan xl="...", yl="..."
- teks lain dengan label("...",x,y)

Perintah label akan menampilkan teks ke plot yang sedang aktif pada koordinat plot (x,y). Perintah ini juga bisa menerima argumen posisi.

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

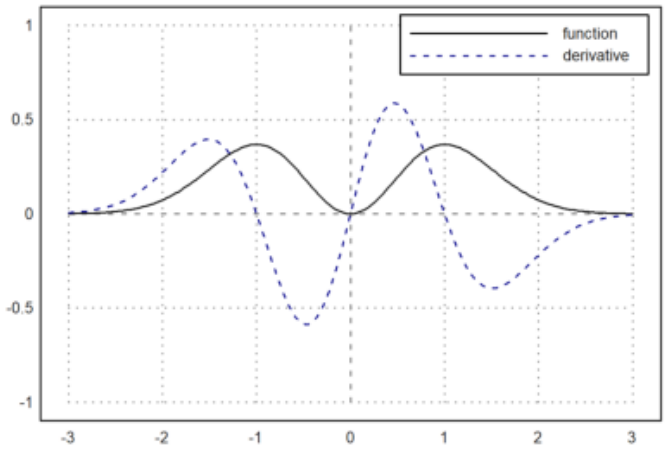



```
>expr := "log(x)/x"; ...
>  plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
>  label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```



Ada juga fungsi labelbox(), yang bisa menampilkan fungsi-fungsi dan teks. Fungsi ini menerima vektor string dan vektor warna, masing-masing 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):
```

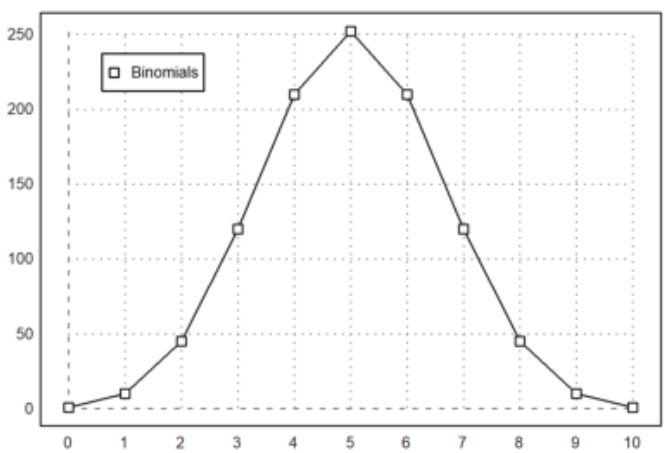


Kotak label secara default ditempatkan di sudut kanan atas, tetapi >left akan menempatkannya di sudut kiri atas. Kamu bisa memindahkannya ke posisi mana saja yang kamu inginkan. Posisi jangkar adalah sudut kanan atas kotak, dan angkanya merupakan pecahan dari ukuran jendela grafik. Lebarnya diatur otomatis.

Untuk plot titik, kotak label juga dapat digunakan. Tambahkan parameter >points, atau sebuah vektor berisi flag, masing-masing untuk setiap label.

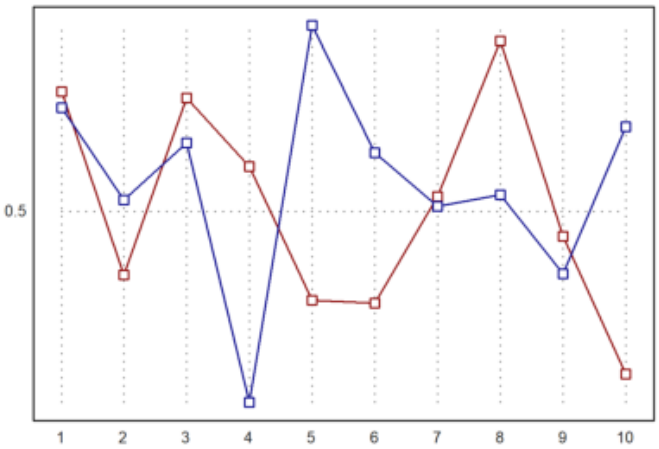
Pada contoh berikut hanya ada satu fungsi. Jadi kita bisa menggunakan string tunggal, bukan vektor string. Kita atur warna teks menjadi hitam untuk contoh ini.

```
>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 seperti ini juga tersedia di statplot(). Sama seperti di plot2d(), warna bisa diatur untuk setiap baris pada plot. Ada lebih banyak plot khusus untuk keperluan statistik (lihat tutorial tentang statistik).

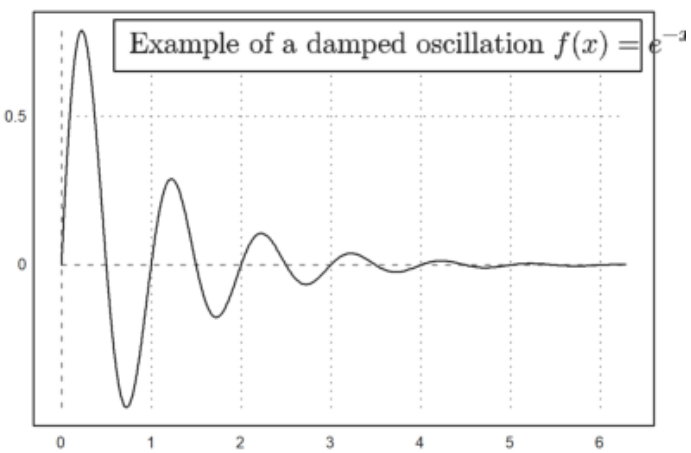
```
>statplot(1:10,random(2,10),color=[red,blue]):
```



Fitur serupa adalah fungsi textbox().

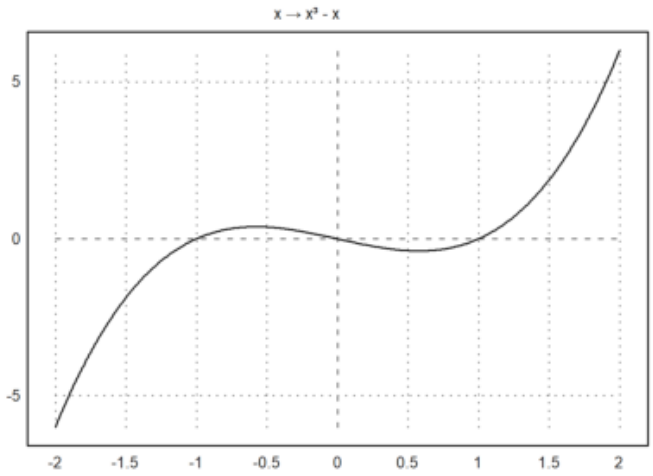
Lebarnya secara default mengikuti lebar maksimal dari baris-baris teks. Namun, lebar ini juga bisa diatur sendiri oleh pengguna.

```
>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):
```



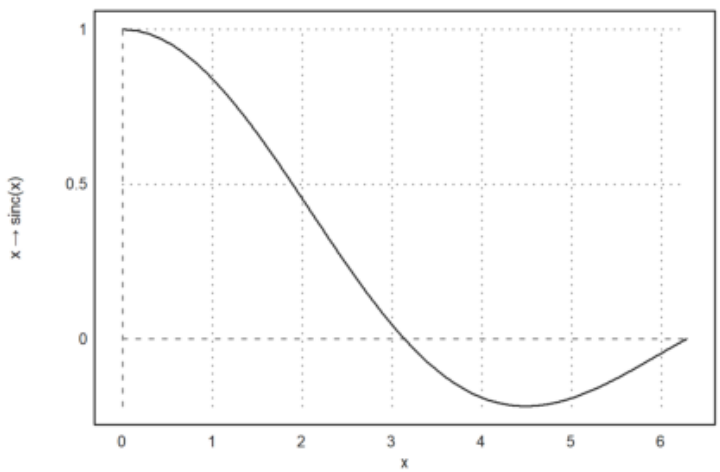
Label teks, judul, kotak label, dan teks lainnya bisa berisi string Unicode (lihat sintaks EMT untuk informasi lebih lanjut tentang string Unicode).

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



Label pada sumbu x dan y bisa ditampilkan secara vertikal, begitu juga dengan sumbunya sendiri.

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



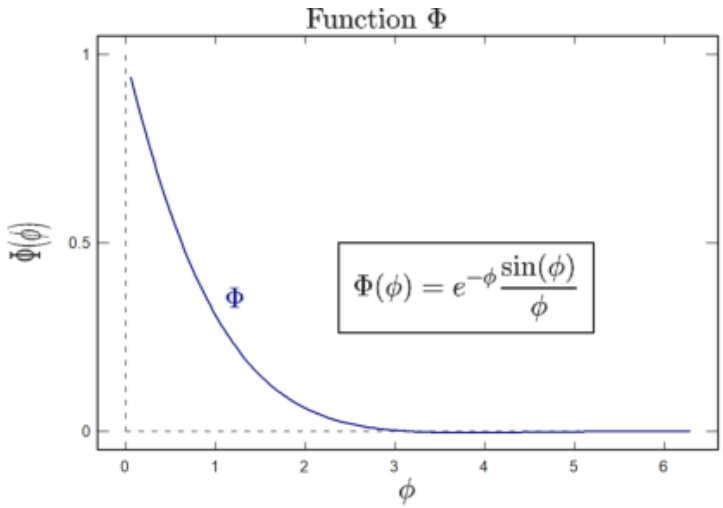
LaTeX

Kamu juga bisa memplot rumus LaTeX jika sistem LaTeX sudah terpasang. Saya merekomendasikan MiKTeX. Jalur ke file biner "latex" dan "dvipng" harus ada di path sistem, atau kamu perlu mengatur LaTeX di menu opsi.

Perlu diperhatikan bahwa pemrosesan LaTeX itu lambat. Jika kamu ingin memakai LaTeX di plot animasi, sebaiknya panggil latex() sekali sebelum loop dan gunakan hasilnya (sebuah gambar dalam matriks RGB).

Pada plot berikut, kita menggunakan LaTeX untuk label sumbu x dan y, label, kotak label, dan judul plot.

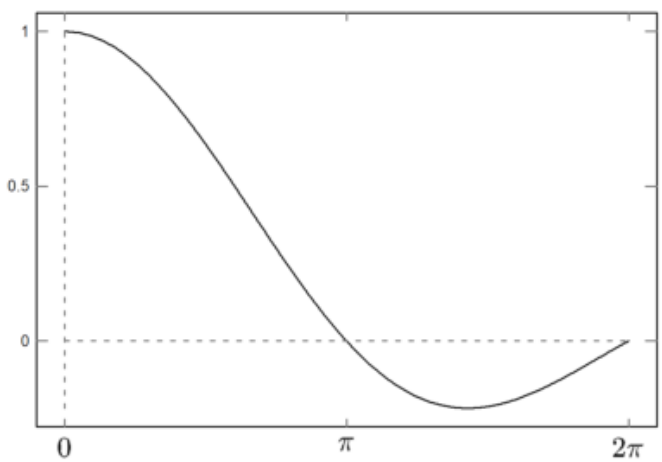
```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
>  title=latex("\text{Function $\Phi$}"), ...
>  xl=latex("\phi"),yl=latex("\Phi(\phi)")); ...
>textbox( ...
>  latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}"),x=0.8,y=0.5); ...
>label(latex("\Phi",color=blue),1,0.4):
```



Sering kali kita menginginkan jarak yang tidak seragam dan label teks pada sumbu x. Kita bisa menggunakan xaxis() dan yaxis() seperti yang akan ditunjukkan nanti.

Cara termudah adalah membuat plot kosong dengan bingkai menggunakan grid=4, lalu menambahkan garis bantu dengan ygrid() dan xgrid(). Pada contoh berikut, kita memakai 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):
```



Fungsi juga dapat digunakan.

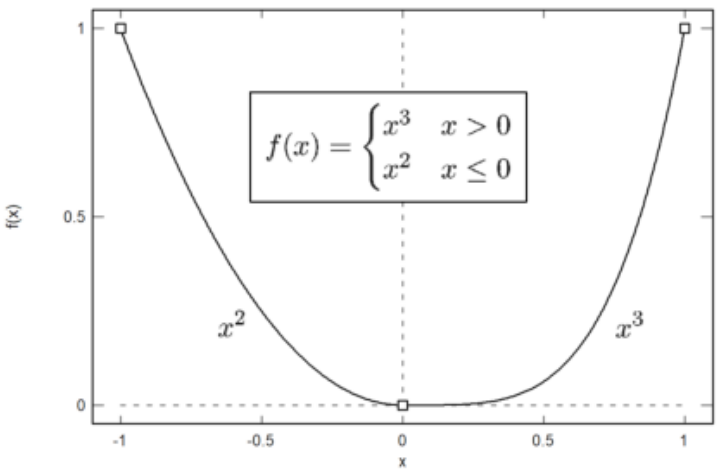
```
>function map f(x) ...
```

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

Parameter “map” membantu agar fungsi bisa digunakan untuk vektor. Untuk plot, sebenarnya ini tidak wajib. Namun untuk menunjukkan bahwa vektorisasi itu berguna, kita menambahkan beberapa titik penting pada plot di x=-1, x=0, dan x=1.

Pada plot berikut, kita juga memasukkan beberapa kode LaTeX. Kita menggunakannya untuk dua label dan sebuah kotak teks. Tentu saja, kamu hanya bisa memakai LaTeX jika sudah terpasang 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

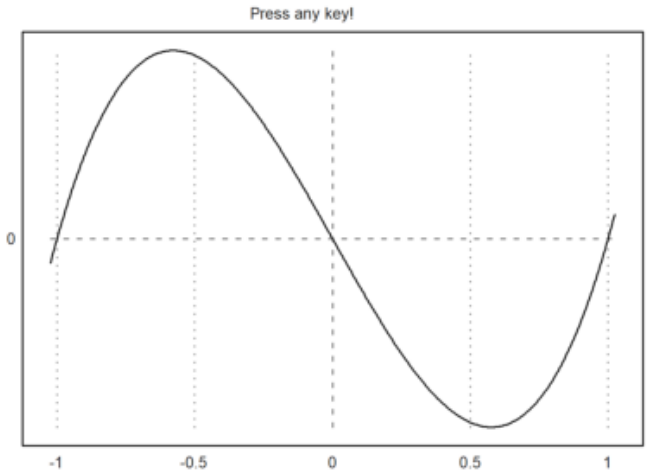
Saat memplot sebuah fungsi atau ekspresi, parameter >user memungkinkan pengguna untuk memperbesar dan menggeser plot menggunakan tombol panah atau mouse. Pengguna bisa:

- memperbesar atau memperkecil dengan tombol + atau –
- menggeser plot dengan tombol panah
- memilih jendela plot dengan mouse
- mengembalikan tampilan ke awal dengan tombol spasi
- keluar dengan tombol return

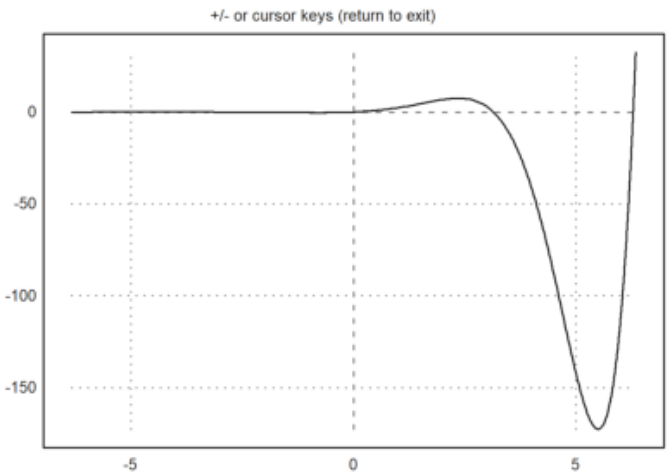
Tombol spasi akan mengembalikan plot ke jendela awalnya.

Saat memplot data, flag >user hanya akan menunggu penekanan tombol dari pengguna.

```
>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 lanjut (lihat tutorial tentang pemrograman untuk detailnya).

Fungsi bawaan mousedrag() menunggu peristiwa dari mouse atau keyboard. Fungsi ini melaporkan mouse ditekan, mouse digerakkan, atau mouse dilepas, serta penekanan tombol. Fungsi dragpoints() memanfaatkan hal ini, dan memungkinkan pengguna untuk menyeret titik mana pun dalam plot.

Kita memerlukan sebuah fungsi plot terlebih dahulu. Sebagai contoh, kita melakukan interpolasi pada 5 titik dengan sebuah polinomial. Fungsi ini harus memplot ke area plot yang tetap.

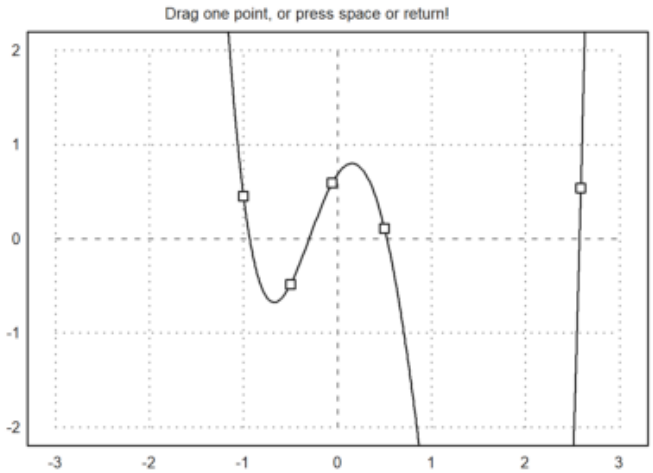
```
>function plotf(xp,yp,select) ...
```

```
    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 setelah titik koma pada plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp(). Tanpa cara ini, kita harus menulis fungsi plotinterp() terlebih dahulu dan mengakses nilainya secara global.

Sekarang kita membangkitkan beberapa nilai acak, lalu membiarkan pengguna menyeret titik-titik tersebut.

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



Ada juga sebuah fungsi yang memplot fungsi lain yang bergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter-parameter tersebut.

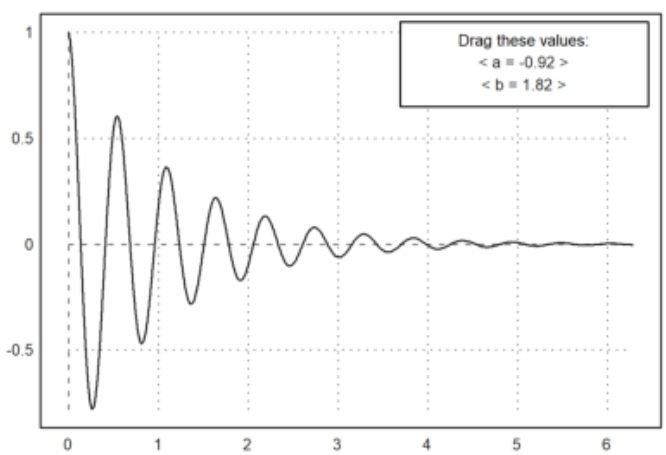
Pertama kita memerlukan fungsi plotnya.

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

Lalu kita memerlukan nama-nama parameter, nilai awal, dan sebuah matriks nx2 yang berisi rentang nilai, serta (opsional) satu baris judul.

Tersedia slider interaktif yang memungkinkan pengguna mengatur nilainya. Fungsi dragvalues() menyediakan fasilitas ini.

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



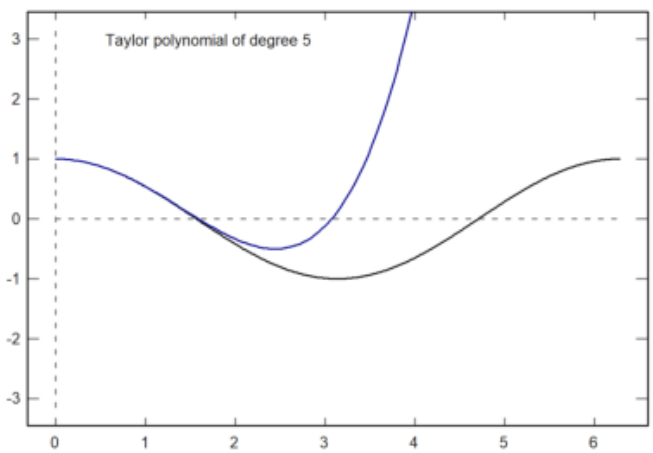
Nilai yang diseret juga bisa dibatasi hanya ke bilangan bulat. Sebagai contoh, kita menulis sebuah fungsi plot yang memplot polinomial Taylor berderajat n untuk fungsi cosinus.

```
>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 derajat n bervariasi dari 0 sampai 20 dalam 20 langkah. Hasil dari dragvalues() digunakan untuk memplot sketsa dengan nilai n tersebut, lalu menyisipkan plotnya ke dalam notebook.

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



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

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

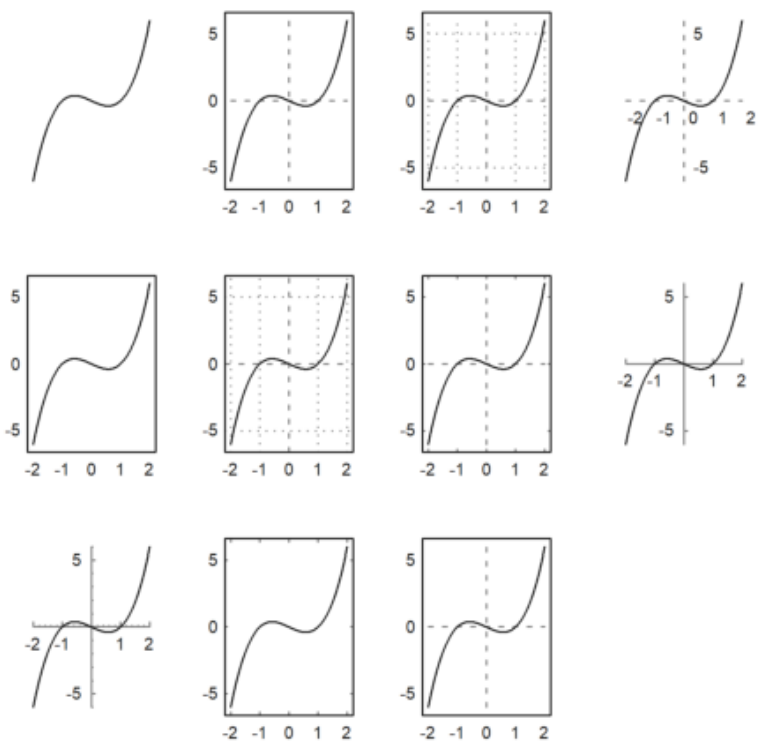
```
    plot2d(none,r=1,title="Drag with the mouse, or press any key!");
    start=0;
    repeat
      {flag,m,time}=mousedrag();
      if flag==0 then return; endif;
      if flag==2 then
        hold on; mark(m[1],m[2]); hold off;
      endif;
    end
  endfunction
```

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

Gaya Plot 2D

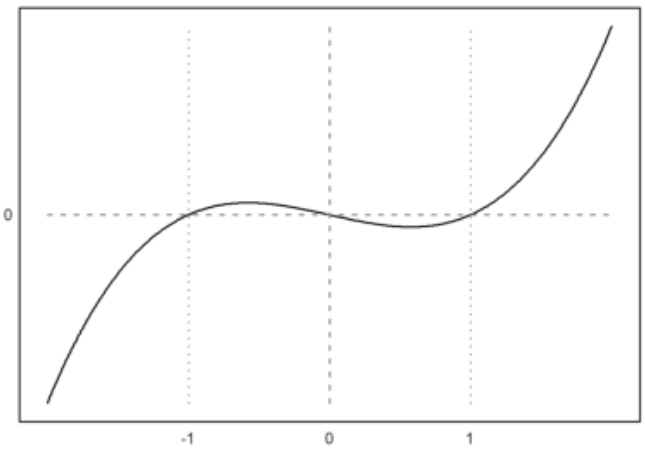
Secara bawaan, EMT menghitung posisi tick pada sumbu secara otomatis dan menambahkan label pada setiap tick. Hal ini bisa diubah menggunakan parameter grid. Gaya default sumbu dan label juga bisa dimodifikasi. Selain itu, kamu bisa menambahkan label dan judul secara manual. Untuk mengembalikan gaya ke pengaturan 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):
```



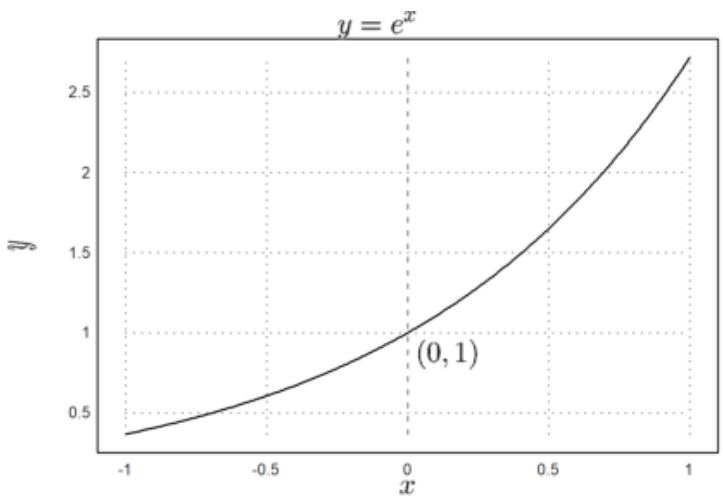
Parameter \<frame mematikan bingkai, dan framecolor=blue mengatur warna bingkai menjadi biru. Jika kamu ingin membuat tick sendiri, gunakan style=0, lalu tambahkan semuanya secara manual.

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



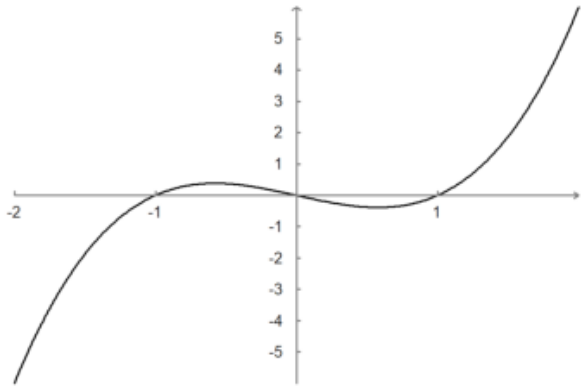
Untuk judul plot dan label sumbu, lihat contoh berikut.

```
>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 bisa digambar secara terpisah menggunakan xaxis() dan yaxis().

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

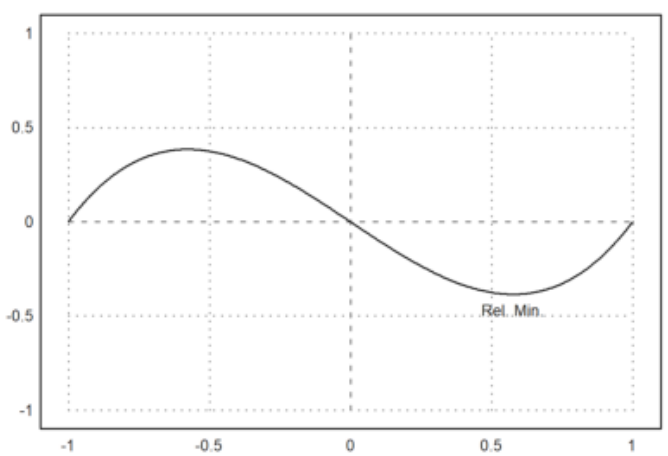


Teks pada plot bisa ditambahkan dengan label(). Pada contoh berikut, "lc" berarti lower center, yang menentukan posisi label relatif terhadap koordinat plot.

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

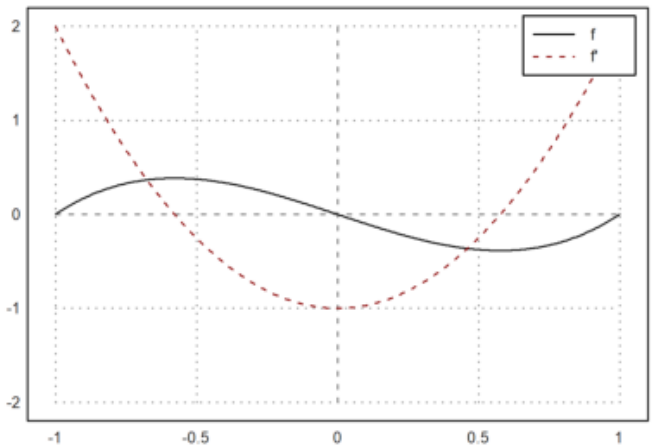
$$x^3 - x$$

```
>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
```

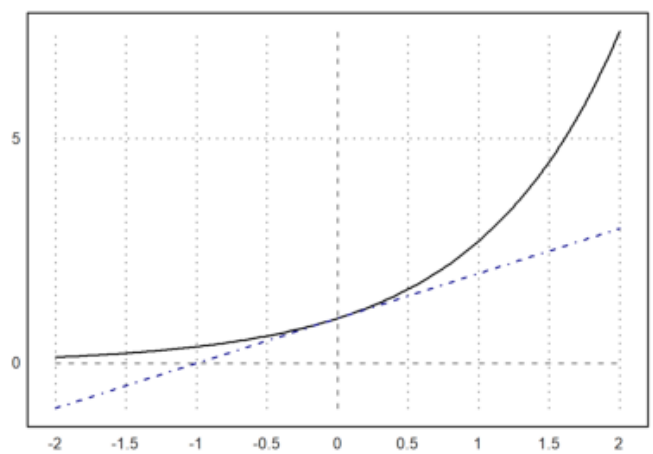


Selain itu, tersedia juga kotak teks (text boxes).

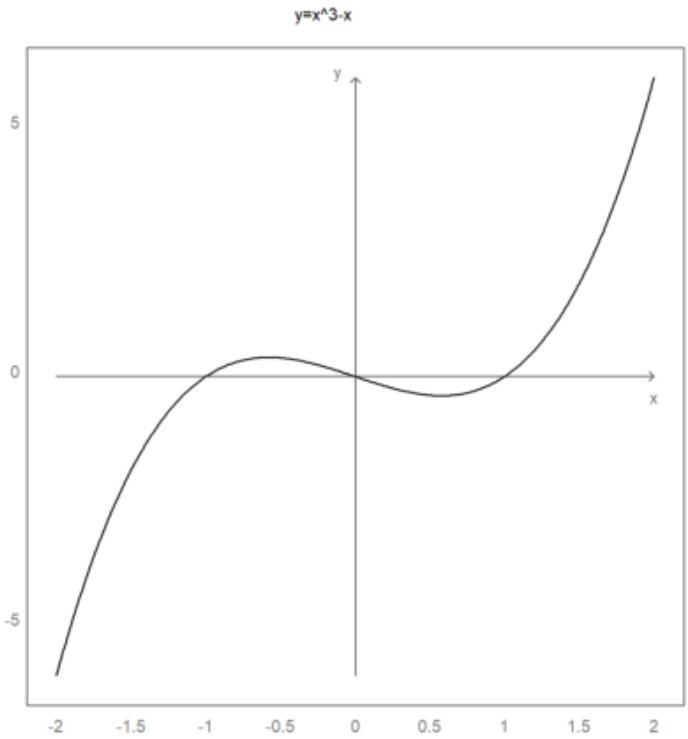
```
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // derivative
>labelbox(["f","f'"],["-","--"],[black,red]): // label box
```




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



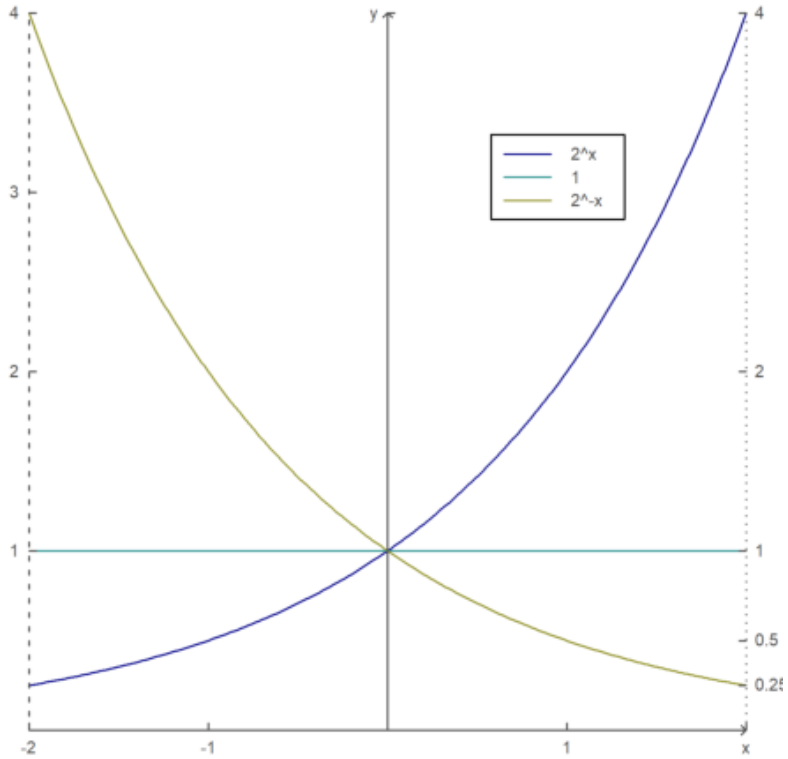

```
>gridstyle("->",color=gray,textcolor=gray,framecolor=gray);  ...
> plot2d("x^3-x",grid=1);   ...
> settitle("y=x^3-x",color=black); ...
> label("x",2,0,pos="bc",color=gray);  ...
> label("y",0,6,pos="cl",color=gray); ...
> reset():
```



Untuk kontrol yang lebih lengkap, sumbu x dan y bisa digambar secara manual.

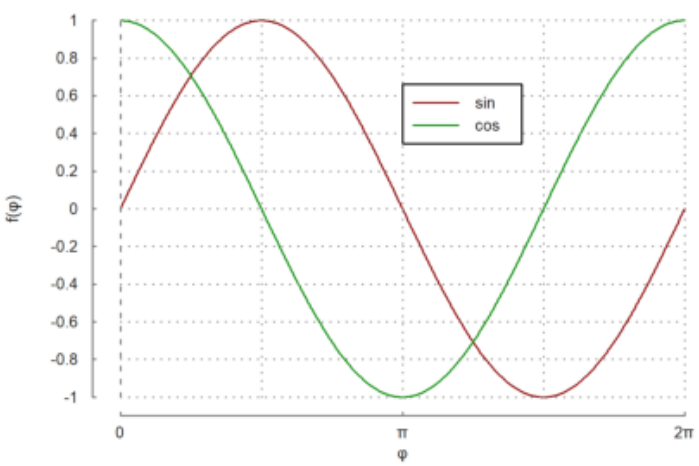
Perintah fullwindow() memperluas jendela plot karena kita tidak lagi membutuhkan ruang untuk label di luar plot. Gunakan shrinkwindow() atau reset() untuk mengembalikan pengaturan ke default.

```
>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 contoh lain, di mana string Unicode digunakan dan sumbu ditempatkan di luar area plot.

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



Plot Data 2D

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

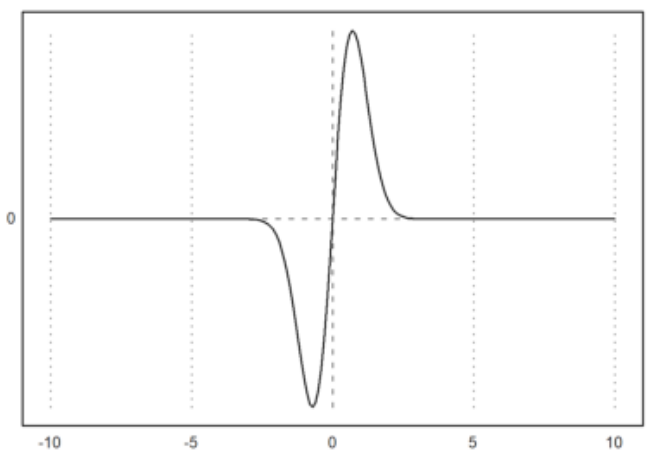
Memplot sebuah ekspresi sebenarnya hanyalah singkatan dari plot data. Untuk plot data, kamu memerlukan satu atau lebih baris nilai x, dan satu atau lebih baris nilai y. Dari rentang dan nilai x, fungsi plot2d akan menghitung data yang akan diplot, secara default dengan evaluasi adaptif dari fungsi. Untuk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

Namun, kamu juga bisa memasukkan data secara langsung:

- Gunakan vektor baris untuk x dan y jika hanya ada satu fungsi.
- Gunakan matriks untuk x dan y untuk diplot baris demi baris.

Berikut contoh dengan satu baris untuk x dan y.

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



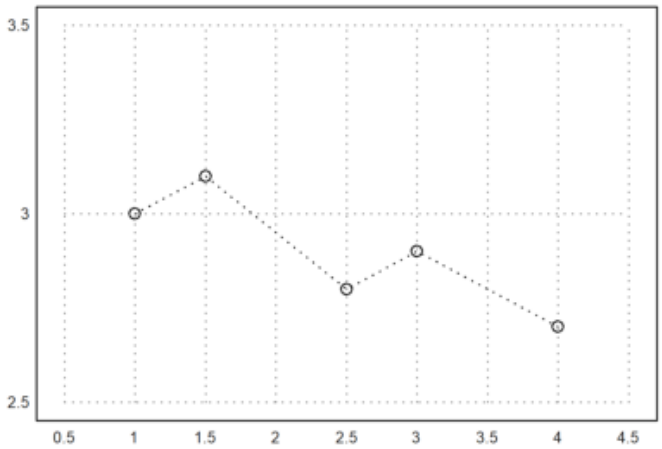
Data juga bisa diplot sebagai titik. Gunakan points=true untuk ini. Plot bekerja seperti poligon, tetapi hanya menggambar titik sudutnya.

- style="...": Select from "[]", "<>", "o", ".", "..", "+", "*", "[]", "<>", "o", "..", "", "|".

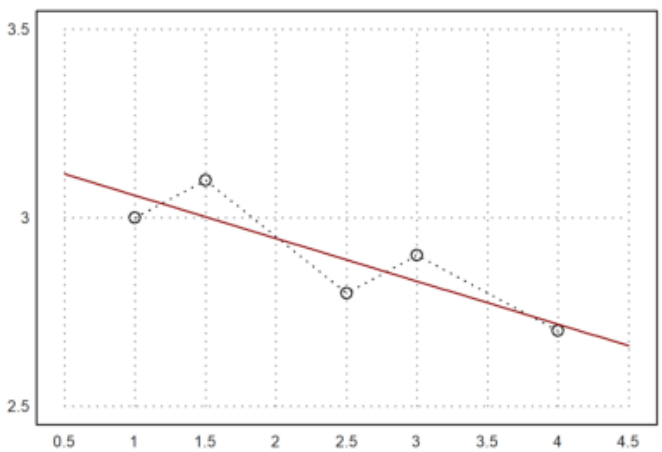
Untuk memplot sekumpulan titik, gunakan >points. Jika color berupa vektor warna, setiap titik akan memiliki warna berbeda. Untuk matriks koordinat dan vektor kolom, warna diterapkan ke tiap baris matriks.

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

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



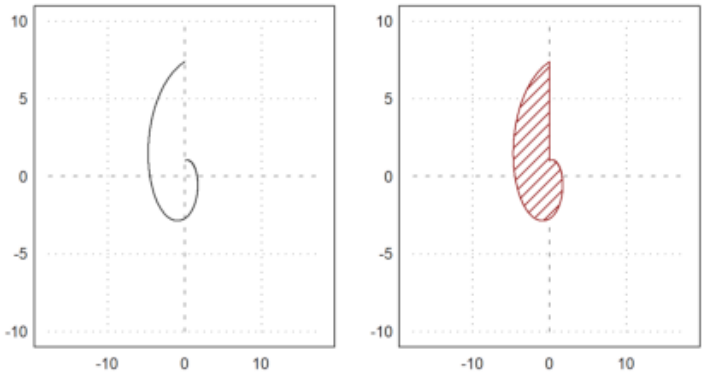
Menggambar Daerah Yang Dibatasi Kurva

Plot data sebenarnya adalah poligon. Kita juga dapat memplot kurva atau kurva terisi.

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

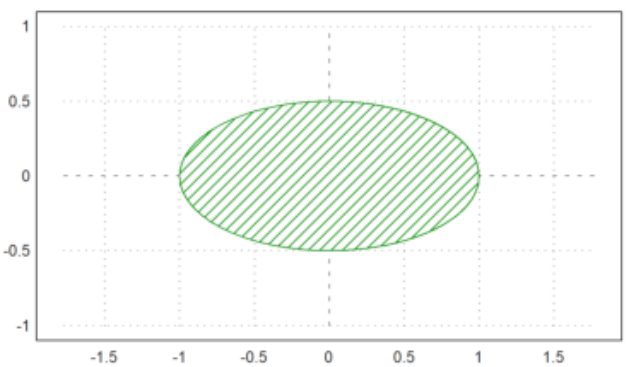
Warna isi ditentukan oleh argumen "fillcolor", dan opsional <outline mencegah penggambaran batas untuk semua gaya kecuali gaya default.

```
>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):
```

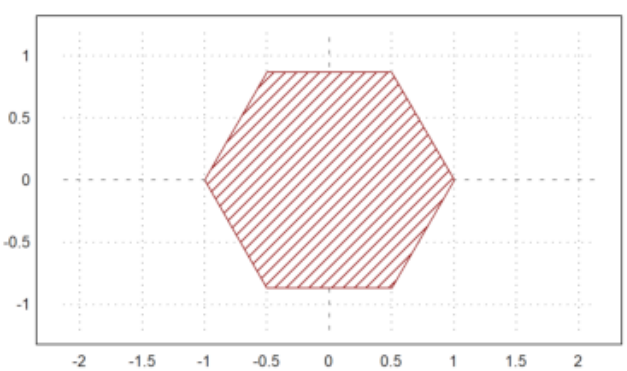


Dalam contoh berikut, kita memplot sebuah ellips terisi dan dua heksagon terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isi yang berbeda.

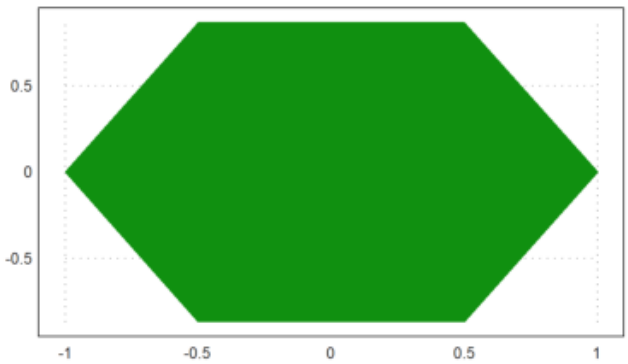
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```




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

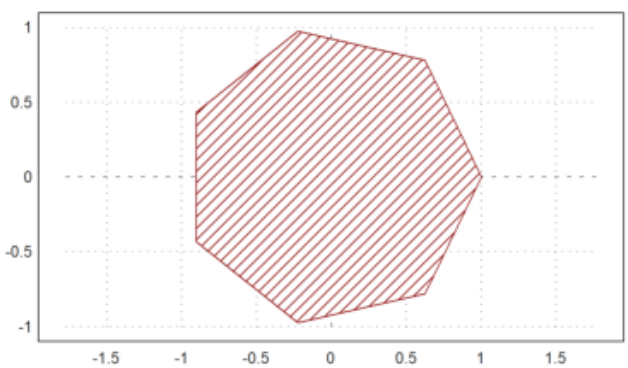


```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



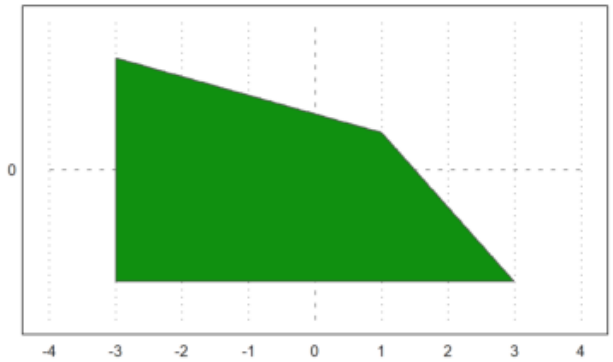
Contoh lain adalah septagon, yang kita buat dengan 7 titik pada lingkaran satuan.

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



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

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

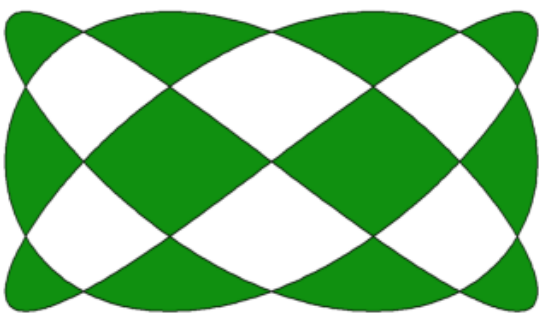


Poin utama dari bahasa matriks adalah bahwa bahasa ini memungkinkan pembuatan tabel fungsi dengan mudah.

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

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

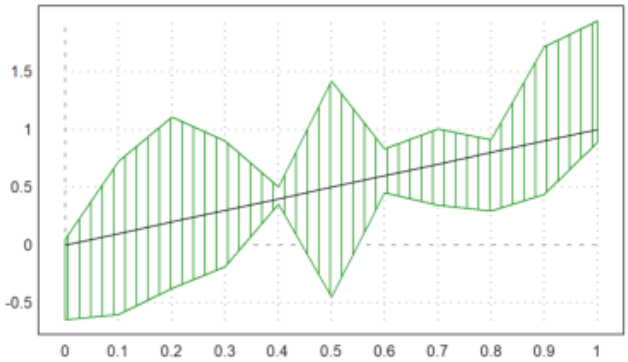
```
>plot2d(x,y,<grid,<frame,>filled):
```



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

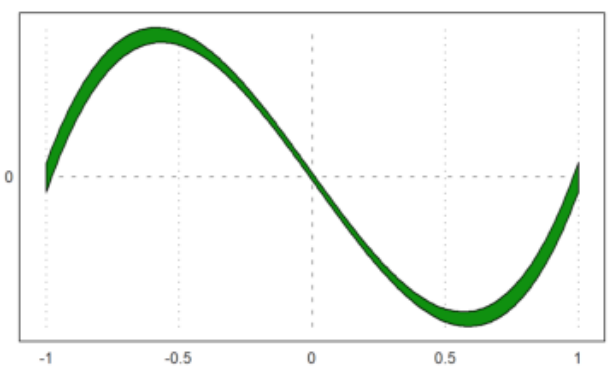
Ini bisa berguna untuk memplot kesalahan perhitungan. Namun, ini juga dapat digunakan untuk memplot kesalahan statistik.

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



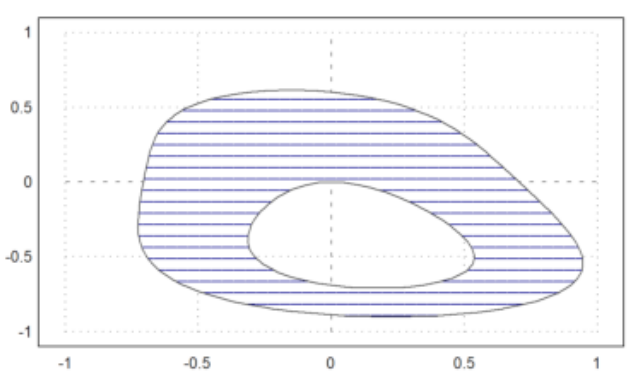
Jika x adalah vektor yang sudah diurutkan, dan y adalah vektor interval, maka plot2d akan memplot rentang terisi dari interval tersebut di bidang. Gaya pengisian sama dengan gaya pada poligon.

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



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

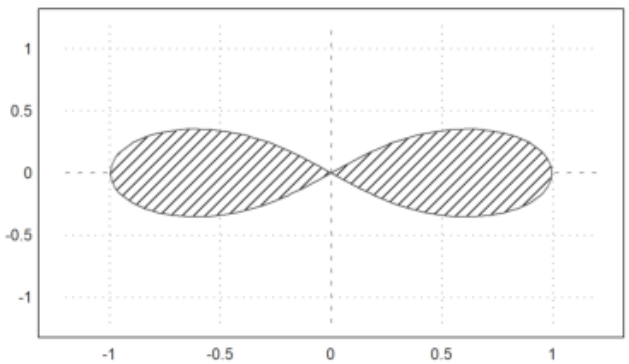
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



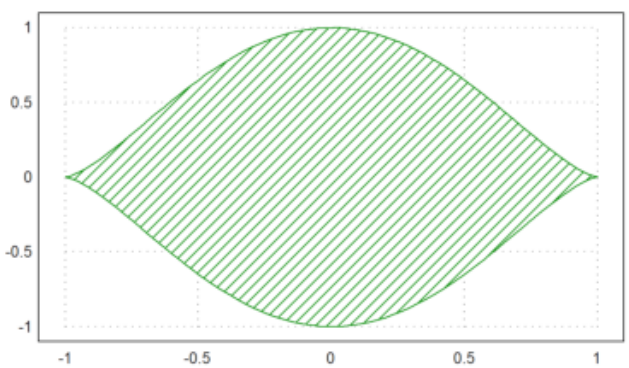
Kita juga dapat mengisi rentang nilai seperti

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

```
>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="/"):
```



Grafik Fungsi Parametrik

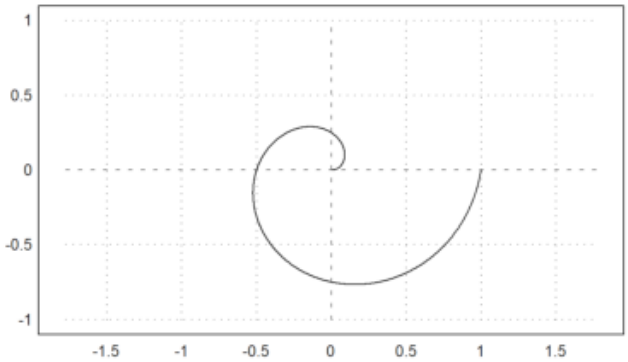
Nilai x tidak harus diurutkan. (x,y) hanya menggambarkan sebuah kurva. Jika x diurutkan, kurva tersebut merupakan grafik dari sebuah fungsi.

Dalam contoh berikut, kita memplot spiral

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

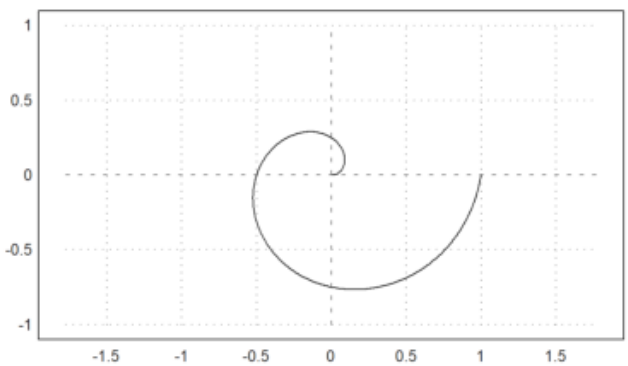
Kita perlu menggunakan sangat banyak titik agar terlihat halus atau menggunakan fungsi adaptive() untuk mengevaluasi ekspresi (lihat fungsi adaptive() untuk detail lebih lanjut).

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

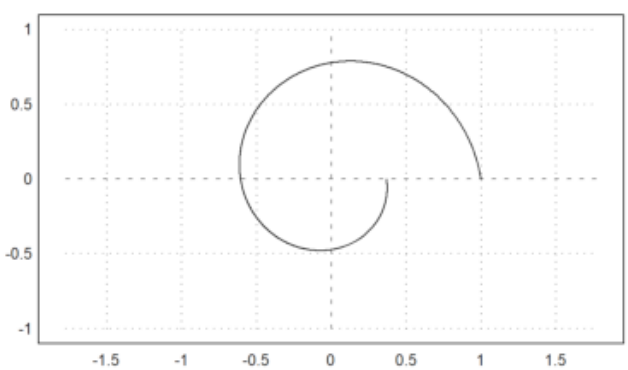


Alternatifnya, dimungkinkan menggunakan dua ekspresi untuk kurva. Berikut memplot kurva yang sama seperti di atas.

```
>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):
```



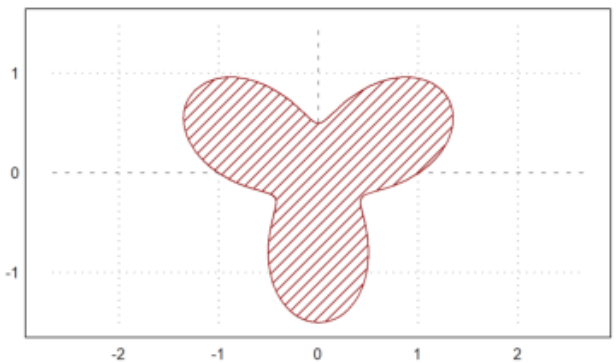
Dalam contoh berikut, kita memplot kurva

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

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):
```



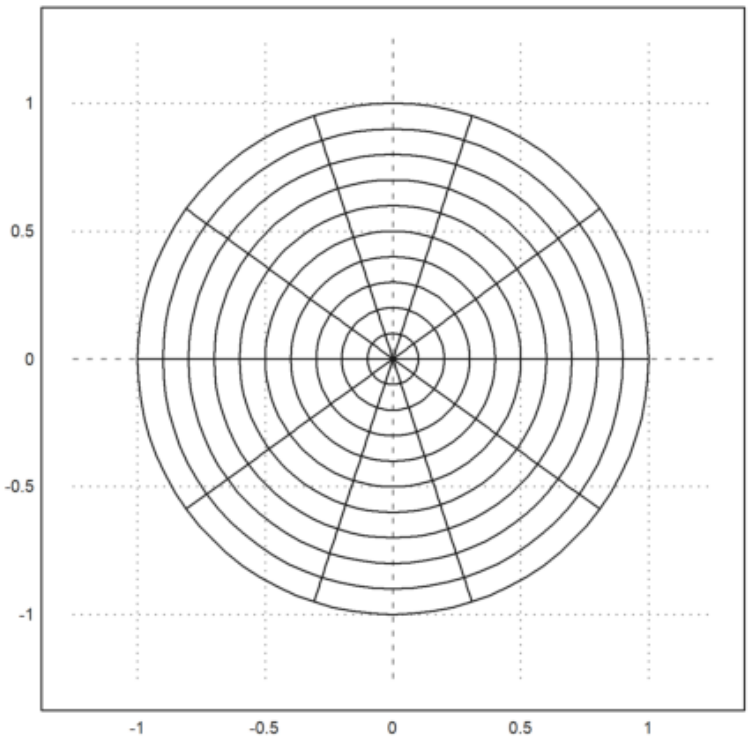
Menggambar Grafik Bilangan Kompleks

Sebuah array bilangan kompleks juga dapat diplot. Titik-titik grid kemudian akan dihubungkan. Jika jumlah garis grid ditentukan (atau vektor 1x2 dari garis grid) pada argumen cgrid, hanya garis-garis grid tersebut yang terlihat.

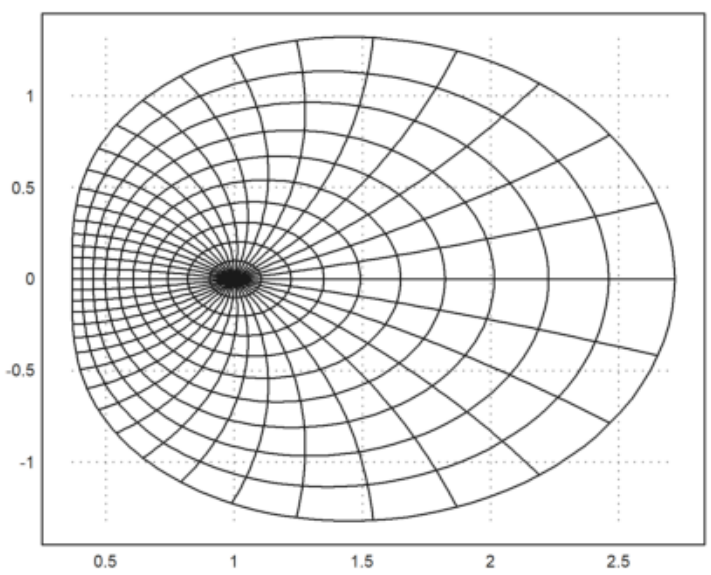
Matriks bilangan kompleks secara otomatis akan diplot sebagai grid di bidang kompleks.

Dalam contoh berikut, kita memplot citra lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

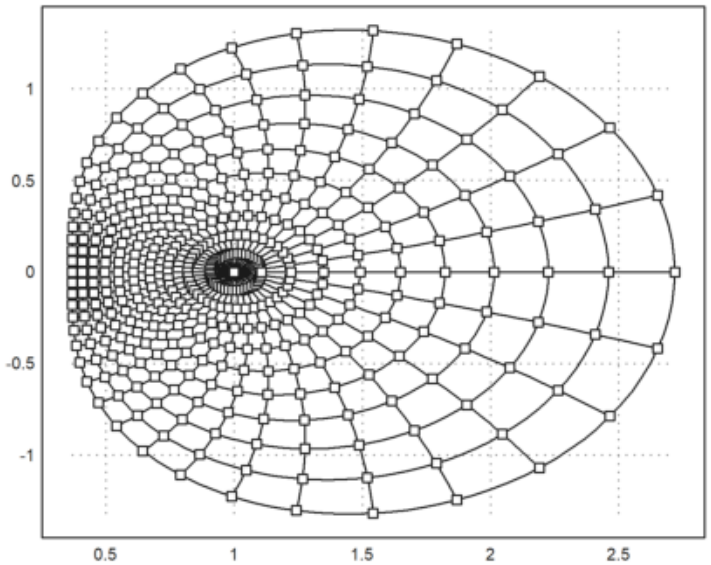
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```




```
>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):
```

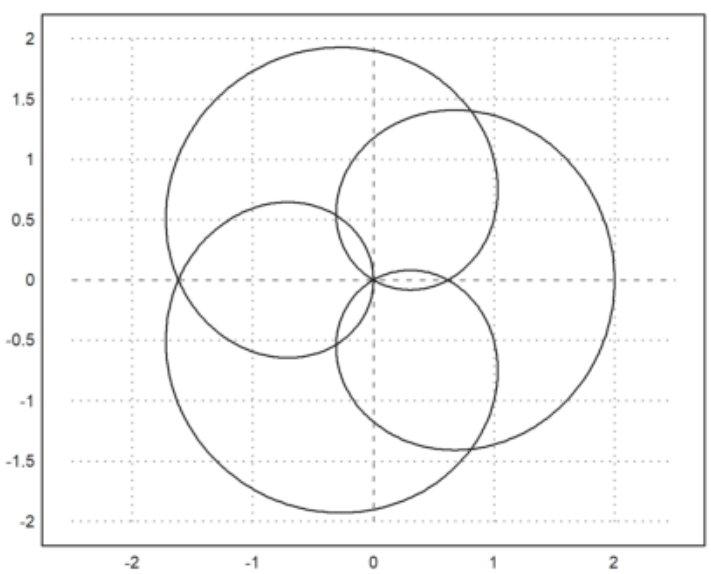


Sebuah vektor bilangan kompleks secara otomatis diplot sebagai kurva di bidang kompleks dengan bagian real dan bagian imajiner.

Dalam contoh ini, kita memplot lingkaran satuan dengan

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

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



Plot Statistik

Terdapat banyak fungsi yang khusus untuk plot statistik. Salah satu plot yang sering digunakan adalah plot kolom.

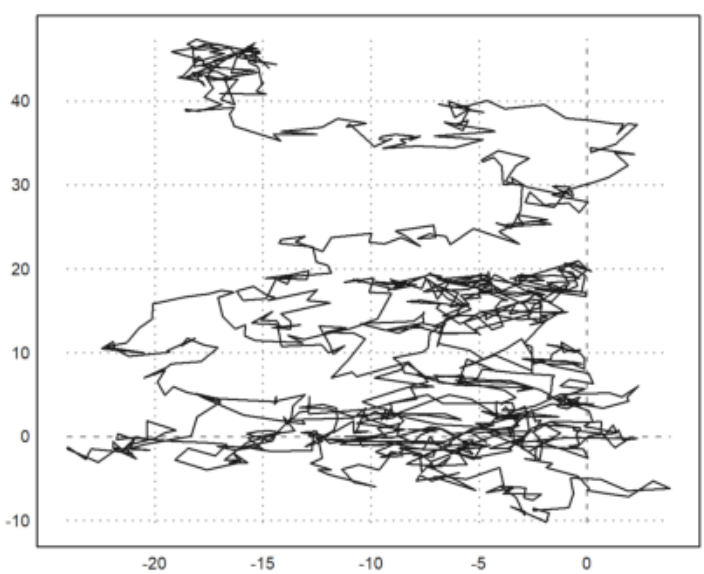
Jumlah kumulatif dari nilai yang terdistribusi normal 0-1 menghasilkan jalur acak (random walk).

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

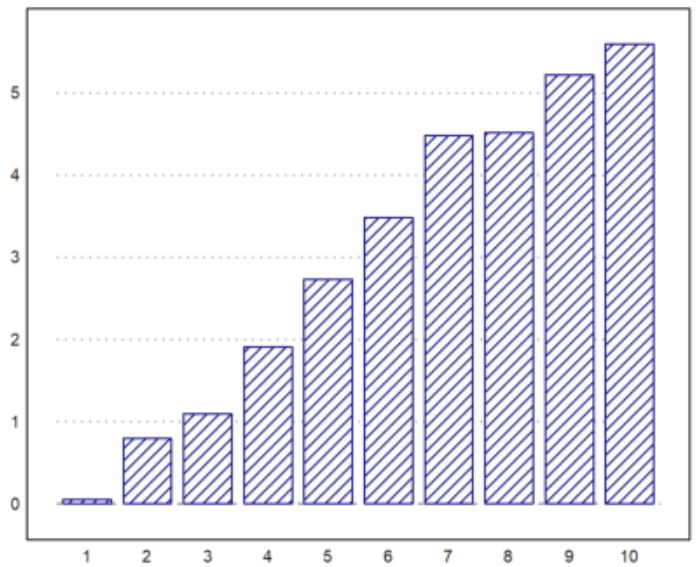


Menggunakan dua baris menunjukkan jalur acak dalam dua dimensi.

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

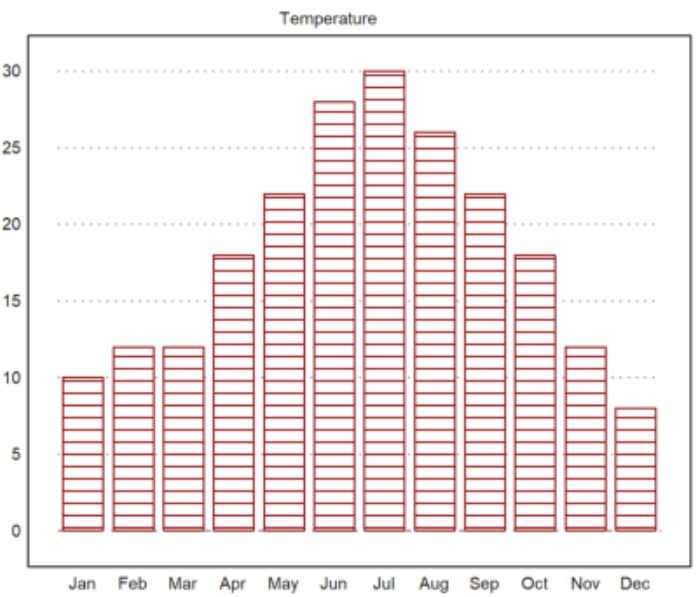


```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

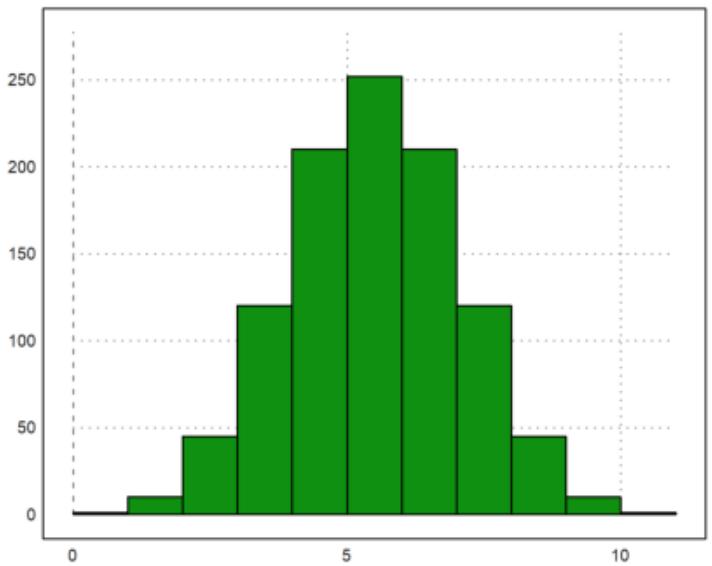


Ini juga dapat menampilkan string sebagai label.

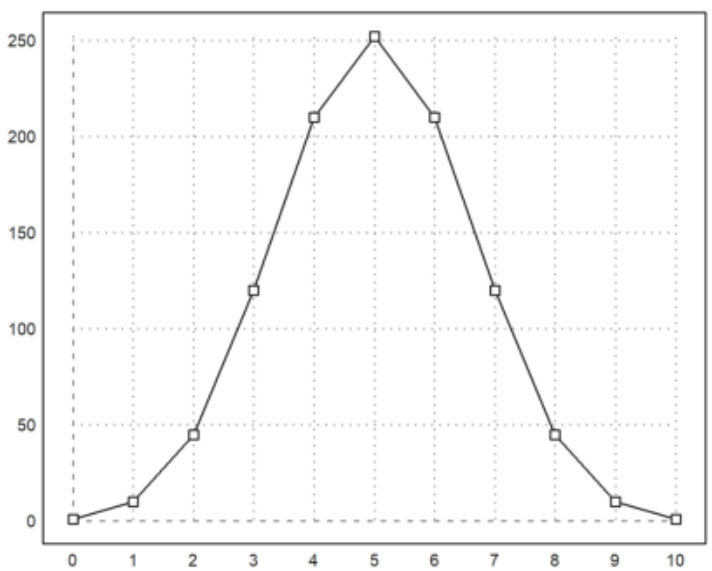
```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
>  "Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnsplot(values,lab=months,color=red,style="-");
>title("Temperature"):
```



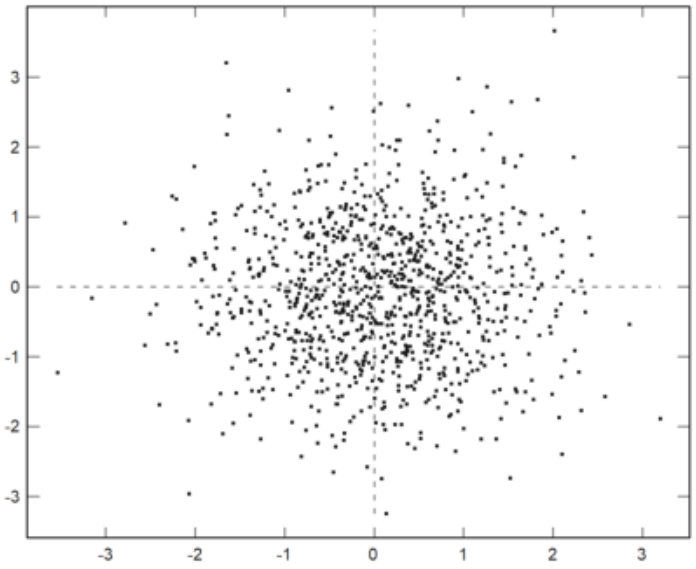
```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```



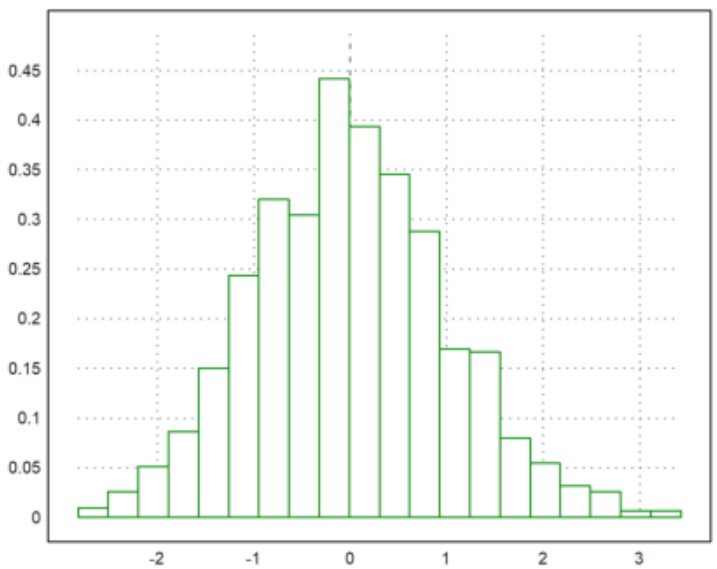

```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



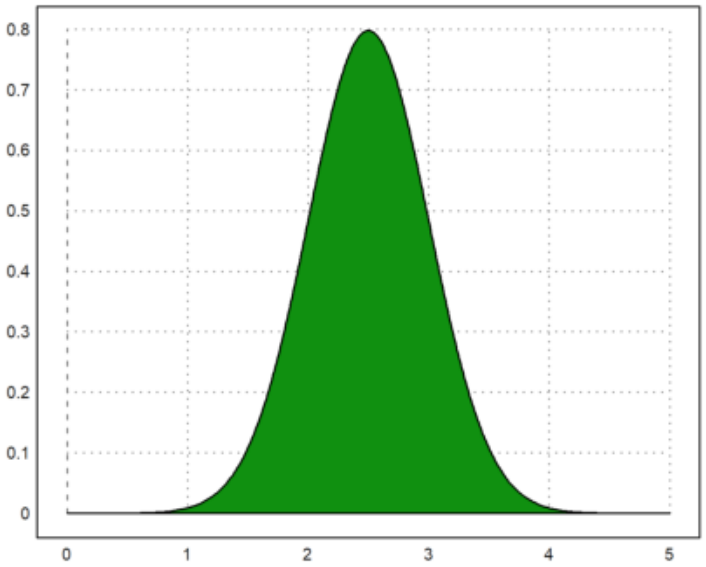

```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```




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

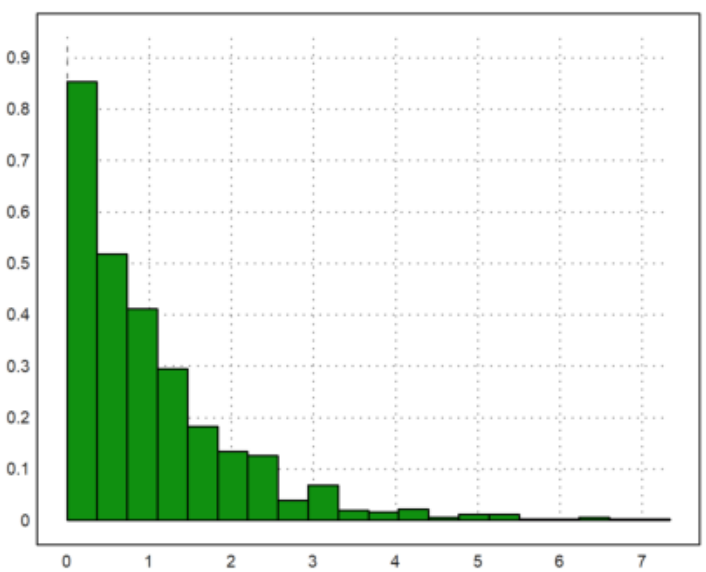



```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



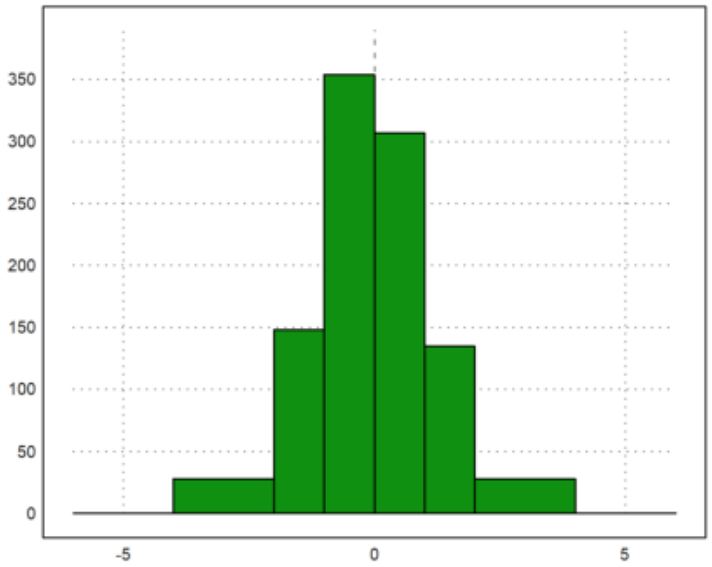
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan distribution=n dengan plot2d.

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



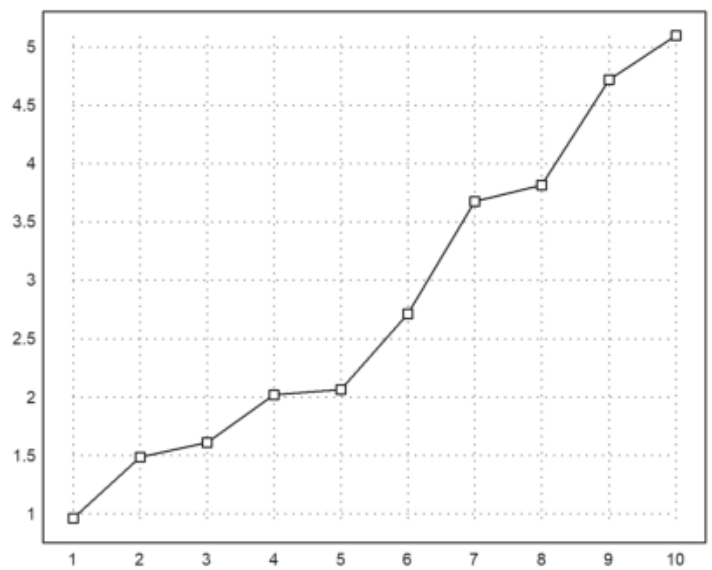
Atau, Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan >bar di plot3d, atau dengan plot kolom.

```
>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):
```

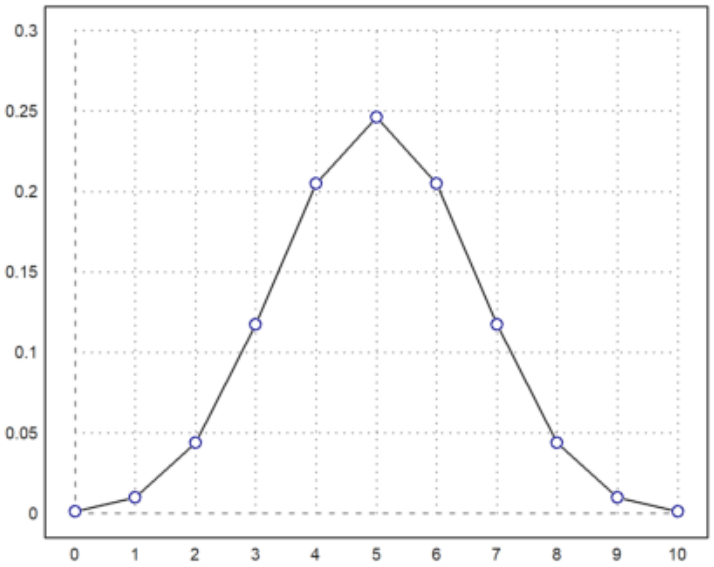


Fungsi statplot() mengatur gaya dengan string sederhana.

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



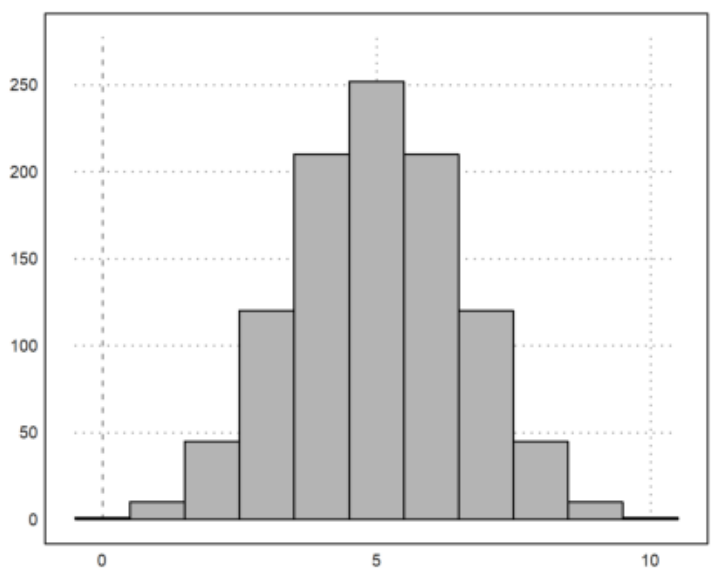

```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



Selain itu, data dapat diplot sebagai batang. Dalam hal ini, x harus diurutkan dan memiliki satu elemen lebih banyak daripada y. Batang akan memanjang dari x[i] ke x[i+1] dengan nilai y[i]. Jika x memiliki ukuran sama dengan y, x akan diperluas dengan satu elemen menggunakan jarak terakhir.

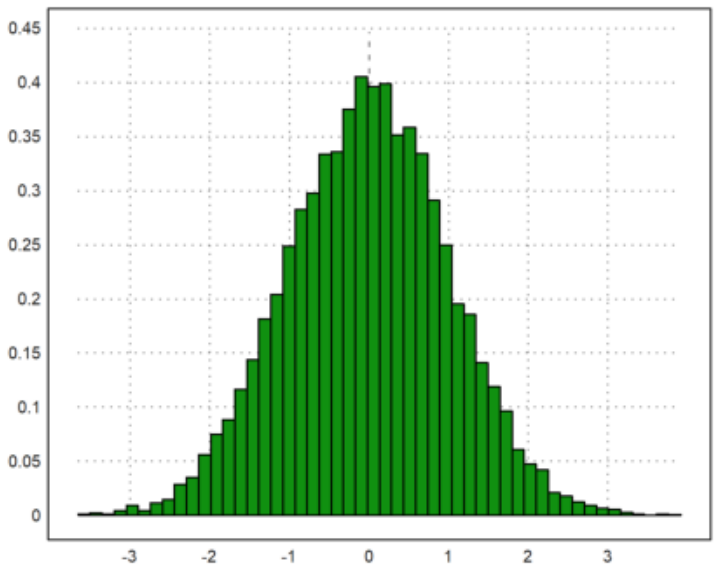
Gaya pengisian dapat digunakan seperti sebelumnya.

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

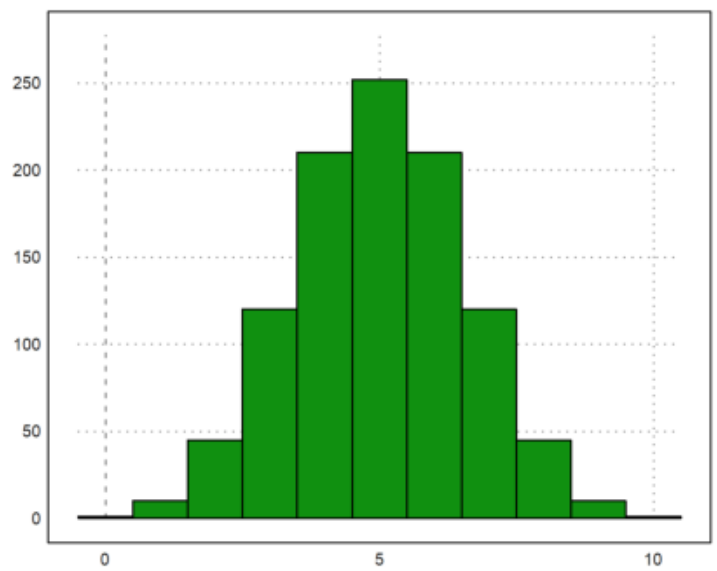


Data untuk plot batang (bar=1) dan histogram (histogram=1) dapat diberikan secara eksplisit di xv dan yv, atau dapat dihitung dari distribusi empiris di xv dengan >distribution (atau distribution=n). Histogram dari nilai xv akan dihitung secara otomatis dengan >histogram. Jika >even ditentukan, nilai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

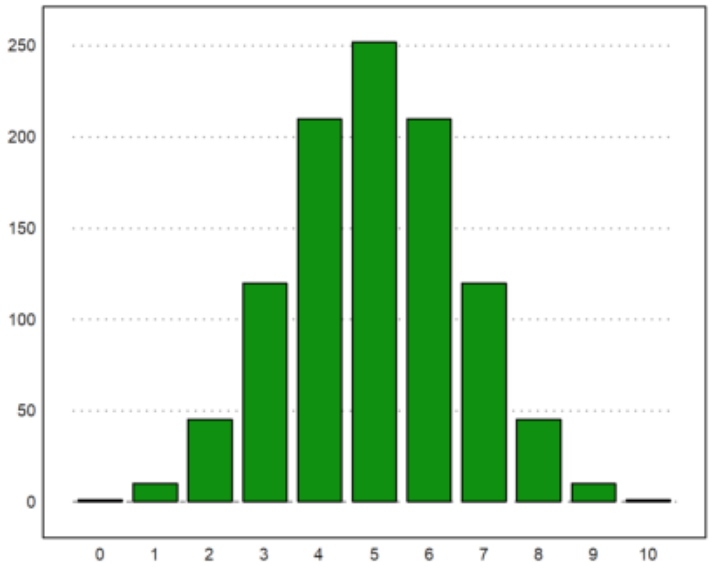
```
>plot2d(normal(10000),distribution=50):
```



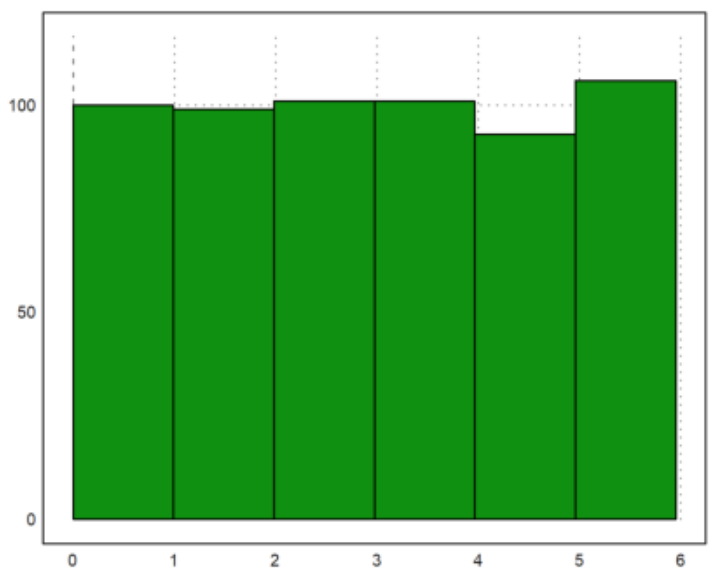

```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```




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

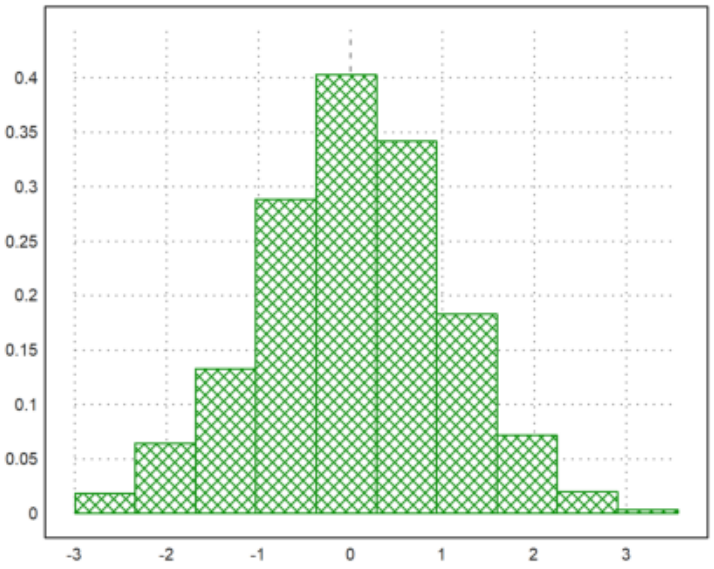



```
>plot2d(random(600)*6,histogram=6):
```



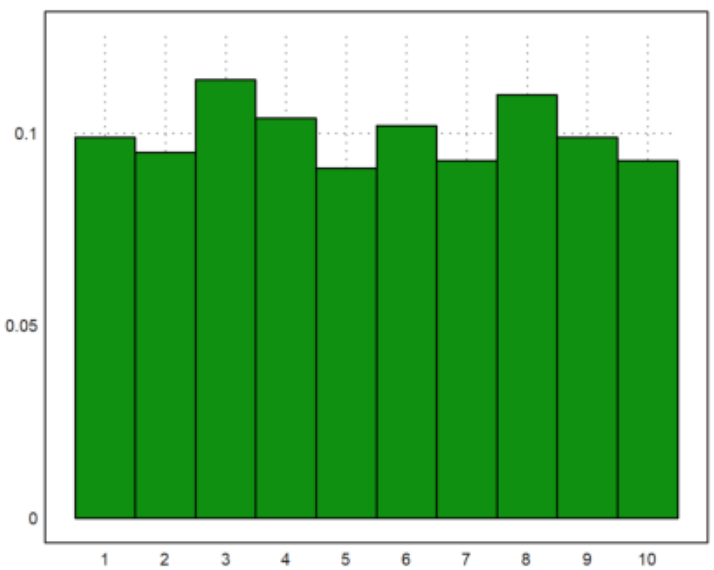
Untuk distribusi, terdapat parameter distribution=n, yang menghitung nilai secara otomatis dan menampilkan distribusi relatif dengan n sub-interval.

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



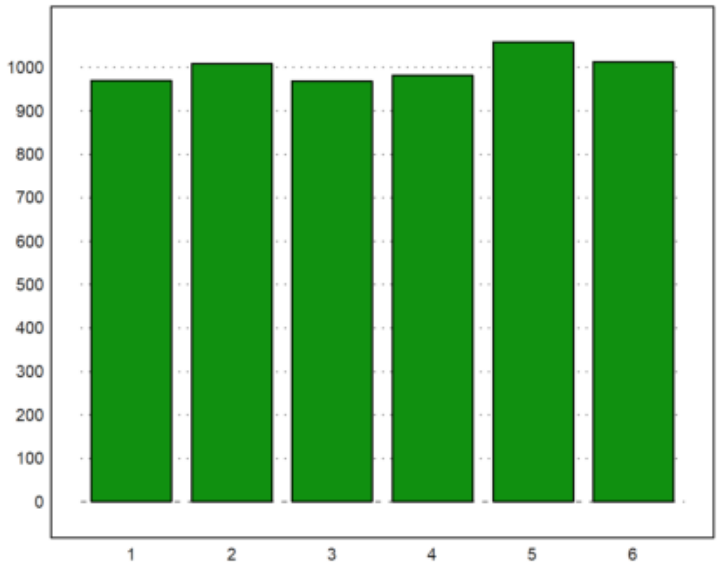
Dengan parameter even=true, ini akan menggunakan interval bilangan bulat.

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

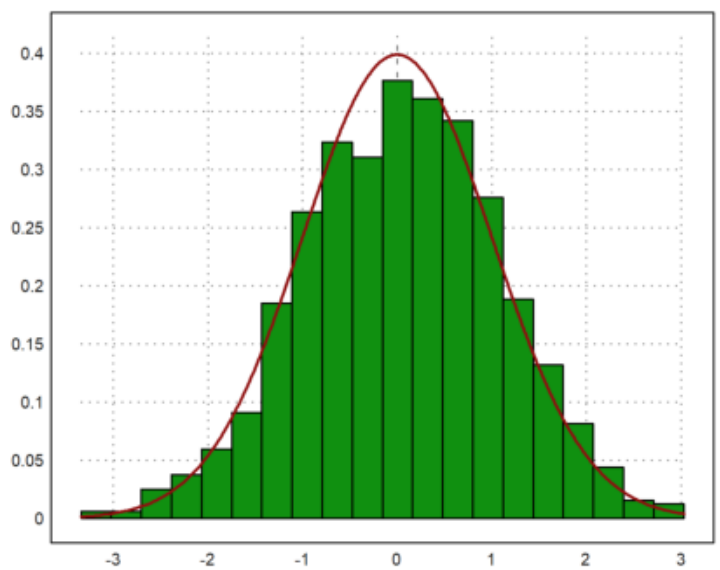


Perlu dicatat bahwa ada banyak plot statistik yang mungkin berguna. Lihatlah tutorial tentang statistik.

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

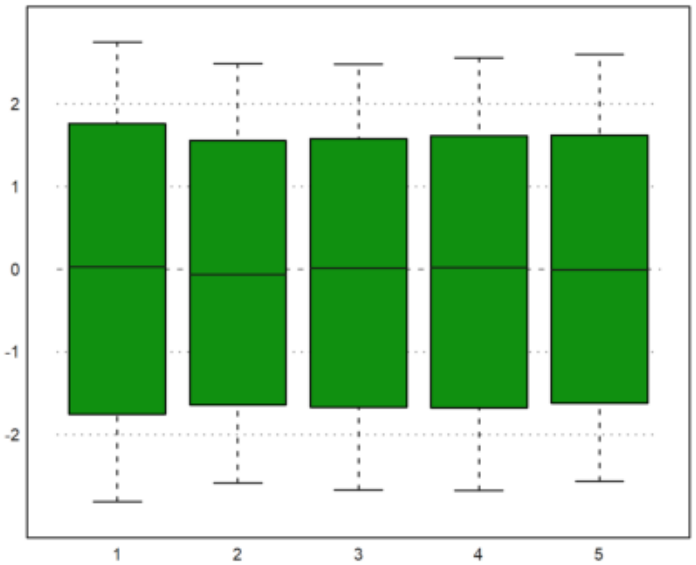



```
>plot2d(normal(1,1000),>distribution); ...
>  plot2d("qnormal(x)",color=red,thickness=2,>add):
```



Terdapat juga banyak plot khusus untuk statistik. Boxplot menunjukkan kuartil dari distribusi ini dan banyak outlier. Menurut definisi, outlier pada boxplot adalah data yang melebihi 1,5 kali rentang 50% tengah dari 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 satu nilai atau vektor nilai. Jika level="auto", akan ada nc garis level yang tersebar merata antara nilai minimum dan maksimum fungsi. Warna 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 xv dapat berupa matriks nilai.

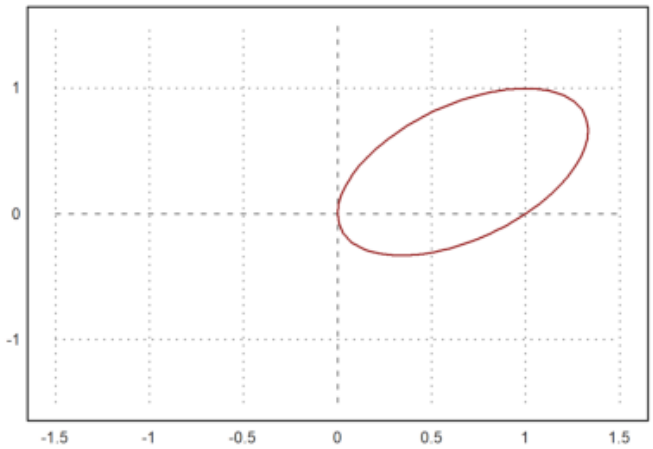
Euler dapat menandai garis level

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

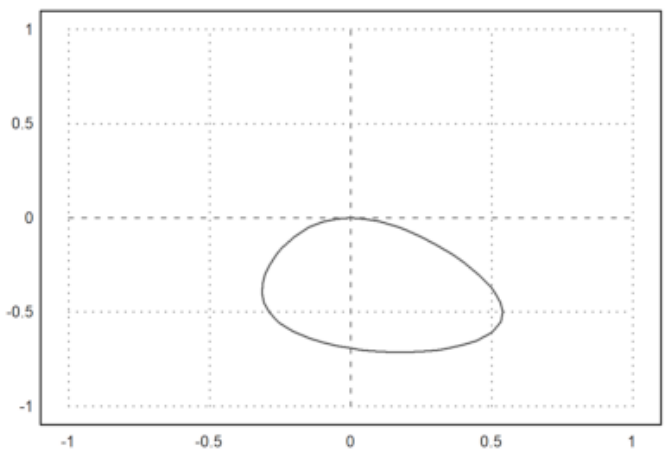
dari fungsi apa pun.

Untuk menggambar himpunan f(x,y)=c untuk satu atau lebih konstanta c, Anda dapat menggunakan plot2d() dengan plot implisit di 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 pada setiap titik dalam plot. Parameter "n" menentukan ketelitian plot.

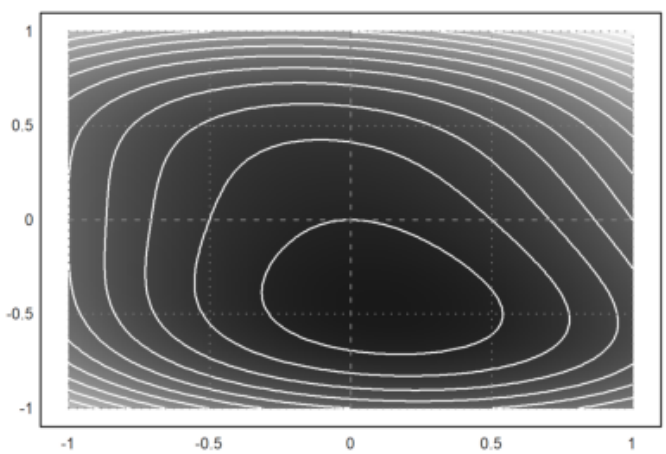
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```



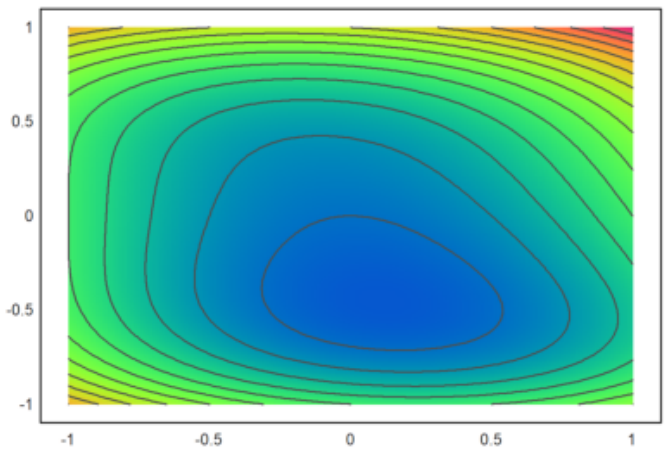

```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0): // Solutions of f(x,y)=0
```




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

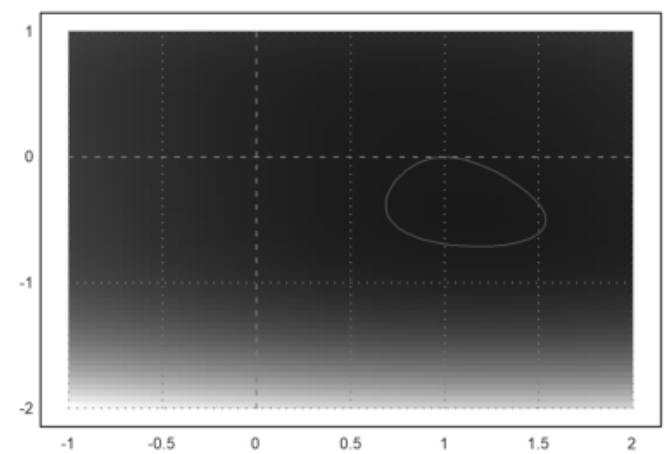


```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
```

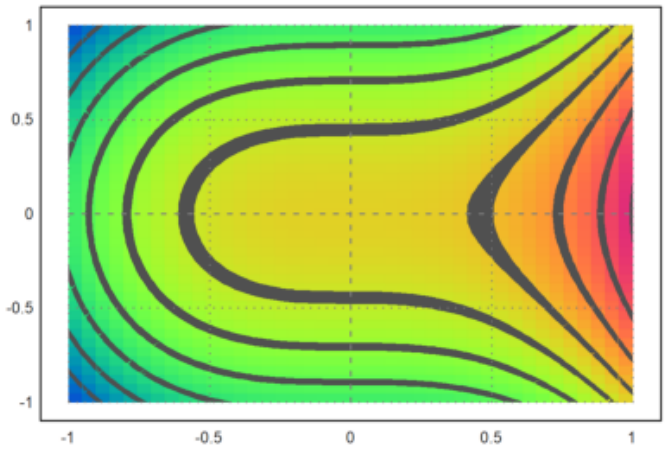


Ini juga berlaku untuk plot data. Namun, Anda harus menentukan rentang untuk label sumbu.

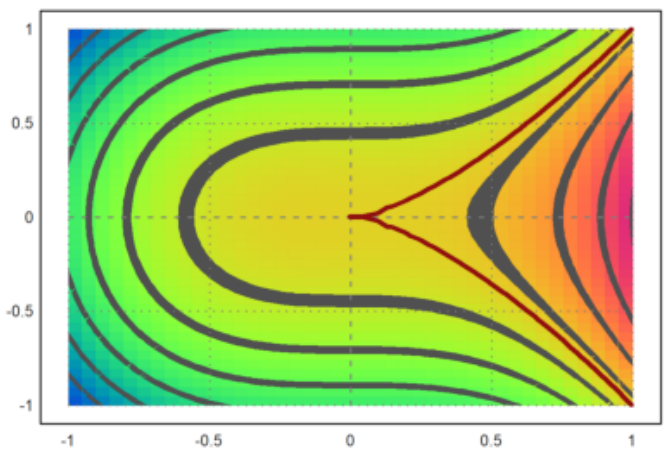
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



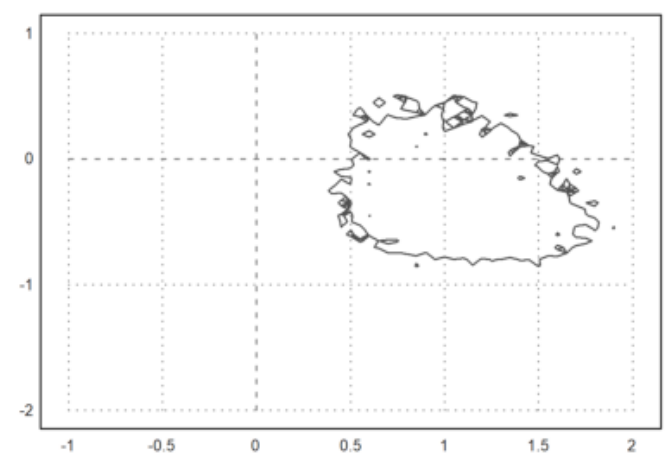

```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



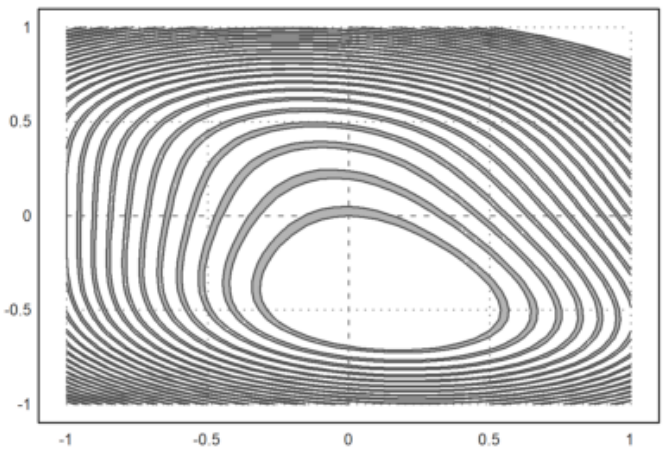

```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



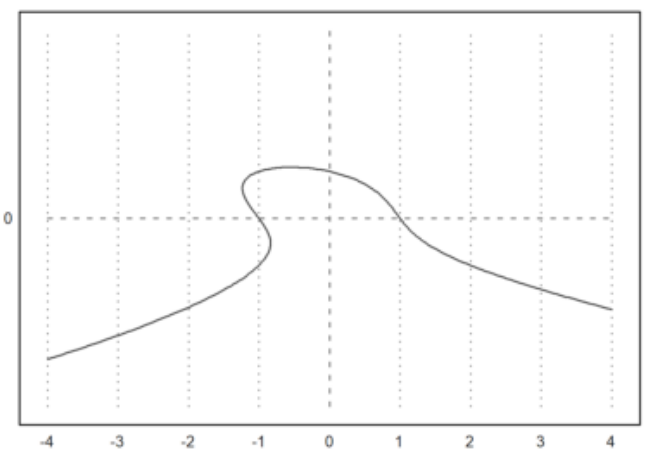

```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



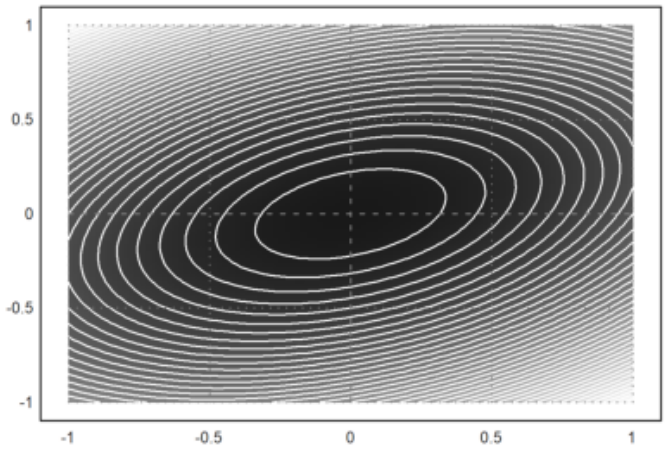
```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```




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




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

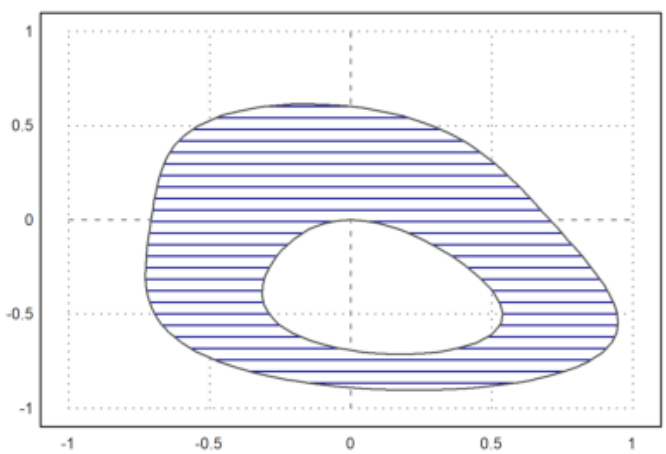


Dimungkinkan juga untuk mengisi himpunan

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

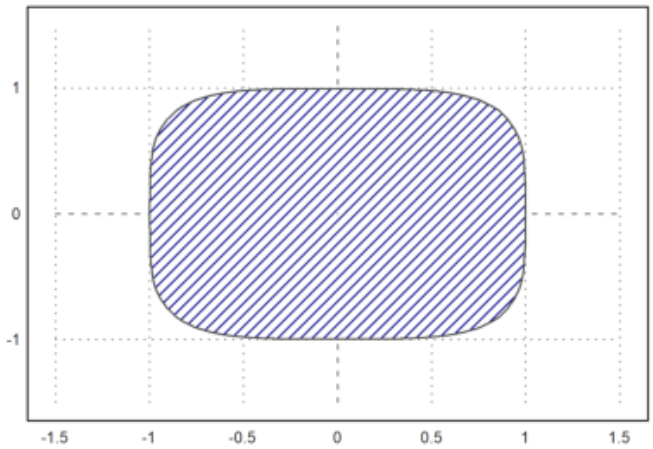
dengan rentang level.
Dimungkinkan untuk mengisi daerah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama berisi batas bawah dan baris kedua berisi batas atas.

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

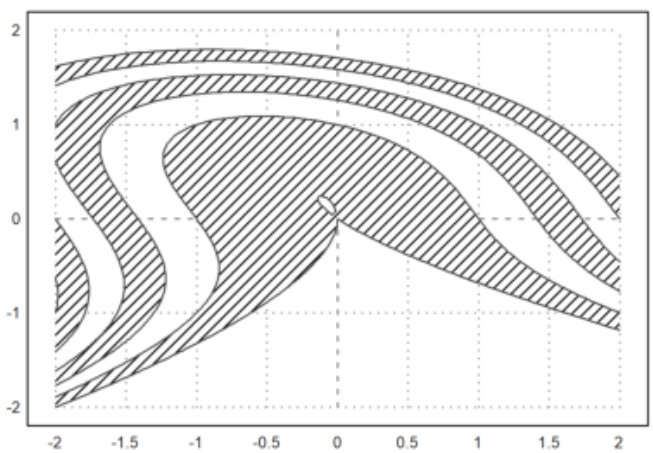


Plot implisit juga dapat menunjukkan rentang level. Maka level harus berupa matriks 2xn dari interval level, di mana baris pertama berisi awal dan baris kedua berisi akhir dari setiap interval. Sebagai alternatif, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level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

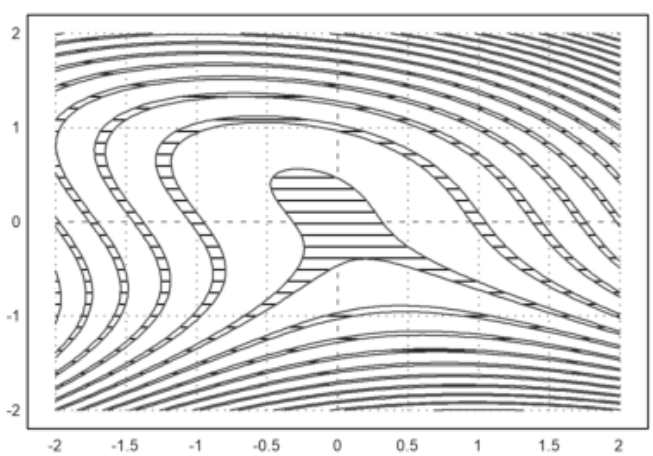
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



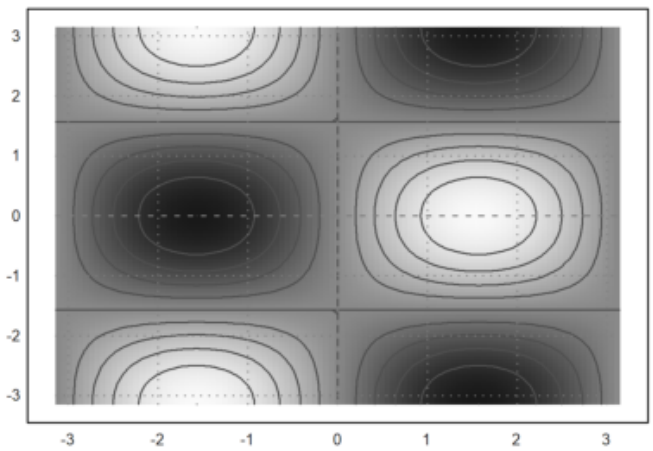

```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```




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



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

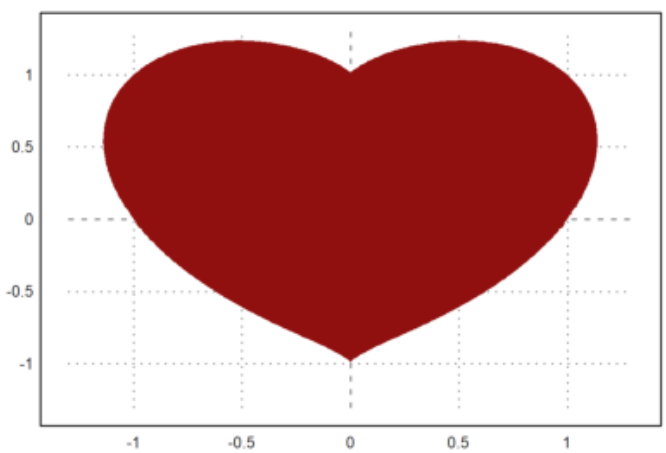


Dimungkinkan juga untuk menandai sebuah daerah

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

Ini dilakukan dengan menambahkan level dengan dua baris.

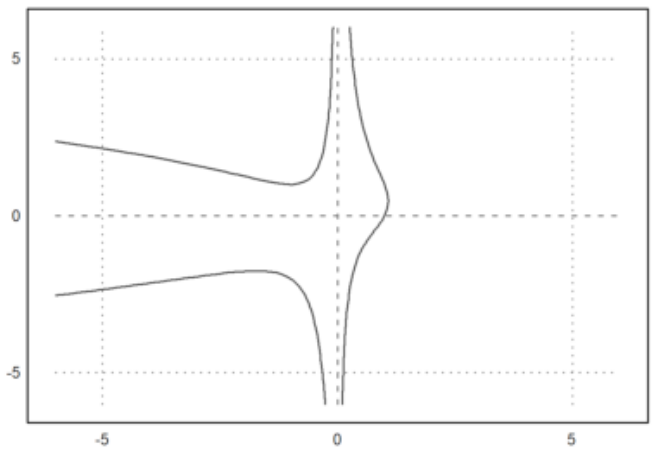
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
>  style="#",color=red,<outline, ...
>  level=[-2;0],n=100):
```



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

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

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



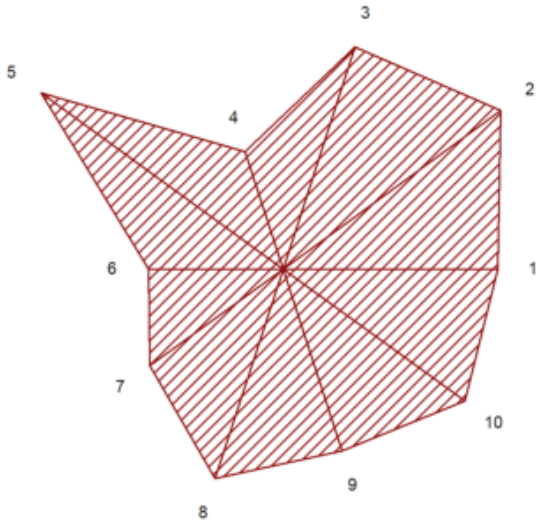

```
>function starplot1 (v, style="/", color=green, lab=none) ...
```

```
    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 sumbu. Selain itu, kita menggunakan seluruh jendela untuk plot.

Kita memanggil reset sebelum menguji plot ini untuk mengembalikan pengaturan grafik ke default. Ini tidak diperlukan jika Anda yakin plot Anda sudah benar.

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



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

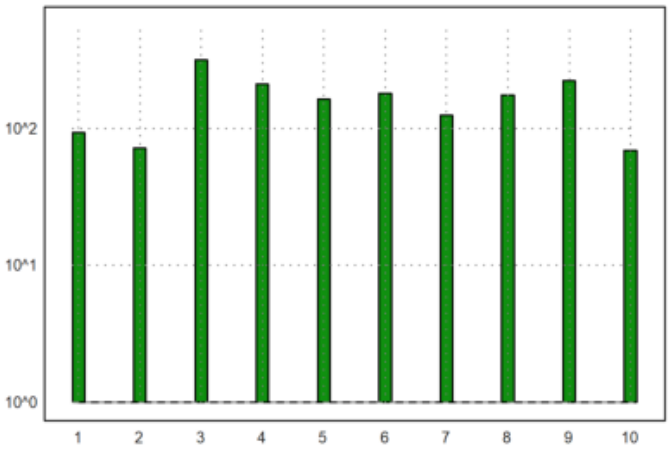
Dalam fungsi berikut, kita membuat plot impuls logaritmik. plot2d dapat membuat plot logaritmik, tetapi tidak untuk batang impuls.

```
>function logimpulseplot1 (x,y) ...
```

```
    {x0,y0}=makeimpulse(x,log(y)/log(10));
    plot2d(x0,y0,>bar,grid=0);
    h=holding(1);
    frame();
    xgrid(ticks(x));
    p=plot();
    for i=-10 to 10;
      if i<=p[4] and i>=p[3] then
         ygrid(i,yt="10^"+i);
      endif;
    end;
    holding(h);
  endfunction
```

Mari kita uji dengan nilai yang terdistribusi eksponensial.

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



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

Fungsi wait(0) memaksa plot muncul di jendela grafik. Jika tidak, 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 pun untuk menghentikan animasi ini.

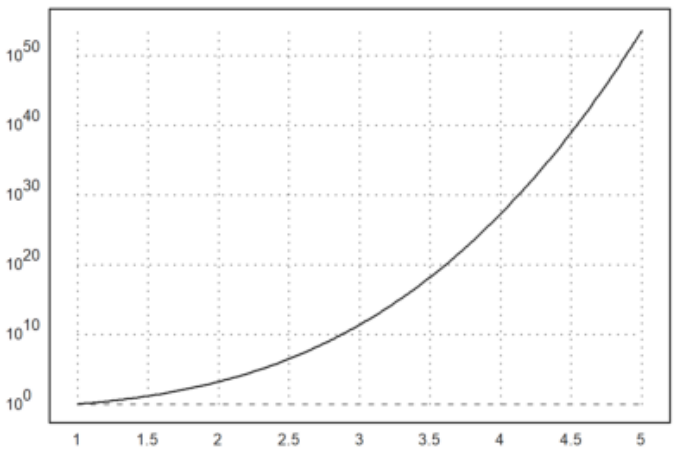
```
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

Plot Logaritmik

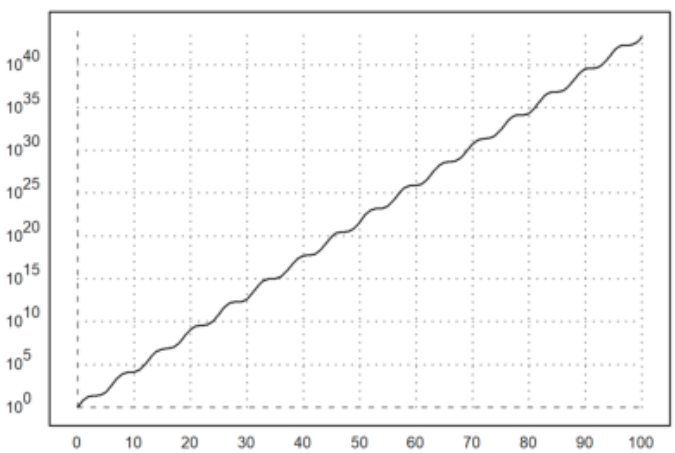
EMT menggunakan parameter "logplot" untuk skala logaritmik. Plot logaritmik dapat dibuat menggunakan skala logaritmik pada y dengan logplot=1, atau skala logaritmik pada x dan y dengan logplot=2, atau pada x dengan logplot=3.

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

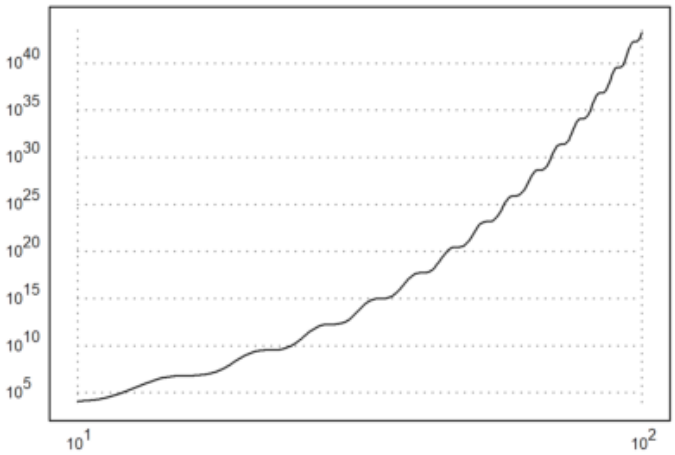
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



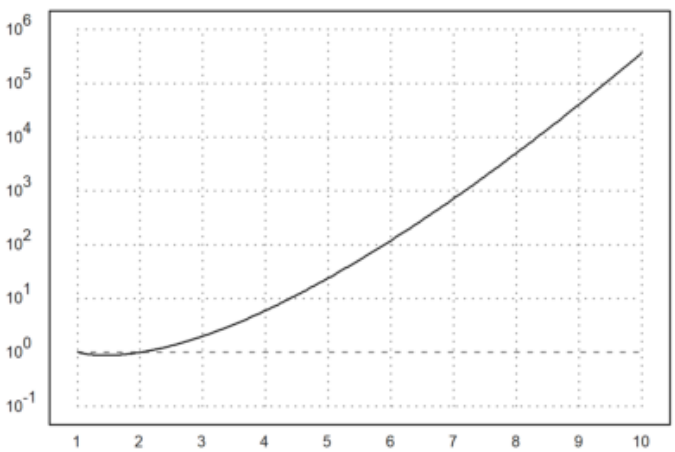

```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



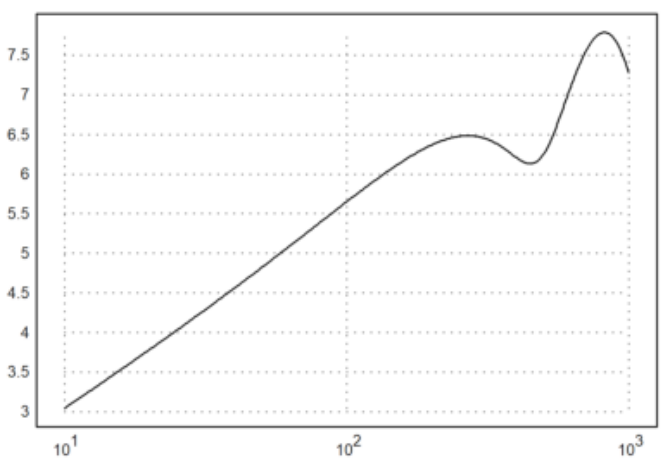

```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```




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




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



Ini juga berlaku untuk plot data.

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

```
  Function xylogplot needs positive values for x
  Try "trace errors" to inspect local variables after errors.
  plot2d:
      xylogplot(xx,yy,=points,=color,=frame,=grid,=ticks);
```

Rujukan Lengkap Fungsi plot2d()

function plot2d (xv, yv, btest, a, b, c, d, xmin, xmax, r, n, ..
logplot, grid, frame, framecolor, square, color, thickness, style, ..
auto, add, user, delta, points, addpoints, pointstyle, bar, histogram, ..
distribution, even, steps, own, adaptive, hue, level, contour, ..
nc, filled, fillcolor, outline, title, xl, yl, maps, contourcolor, ..
contourwidth, ticks, margin, clipping, cx, cy, insimg, spectral, ..
cgrid, vertical, smaller, dl, niveau, levels)

Fungsi plot serbaguna untuk membuat grafik di bidang 2D. Fungsi ini dapat memplot fungsi satu variabel, data, kurva di bidang, plot batang, grid bilangan kompleks, dan plot implisit dari fungsi dua variabel.

Parameter

x,y : persamaan, fungsi, atau vektor data
a,b,c,d : Area plot (default a=-2, b=2)
r : jika r ditetapkan, 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 : Tentukan rentang-y secara otomatis (default)
square : jika true, coba pertahankan rentang x-y berbentuk kotak
n : jumlah interval (default adalah adaptive)
grid : 0 = tanpa grid dan label,

```
            1 = hanya sumbu,
            2 = grid normal (lihat jumlah garis grid di bawah)
            3 = di dalam sumbu
            4 = tanpa grid
            5 = grid penuh termasuk margin
            6 = ticks di frame
            7 = hanya sumbu
            8 = hanya sumbu, sub-ticks
```

frame : 0 = tanpa frame
framecolor: warna frame dan grid
margin : angka antara 0 dan 0.4 untuk margin di sekitar plot
color : Warna kurva. Jika ini vektor warna,

```
            digunakan untuk setiap baris matriks plot. Untuk plot
```

titik,

```
sebaiknya vektor kolom. Jika vektor baris atau matriks
```

penuh

```
digunakan, akan diterapkan pada setiap titik.
```

thickness : ketebalan garis untuk kurva

```
Nilai ini bisa <1 untuk garis sangat tipis.
```

style : Gaya plot untuk garis, marker, dan isi.

```
Untuk titik gunakan:
"[]", "<>", ".", "..", "...",
"*", "+", "|", "-", "o"
"[]#", "<>#", "o#" (bentuk terisi)
"[]w", "<>w", "ow" (non-transparan)
Untuk garis gunakan:
"-", "--", "-.", ".", ".-.", "-.-", "->"
Untuk poligon terisi atau plot batang gunakan:
"#", "#O", "O", "/", "\", "\/",
"+", "|", "-", "t"
```

points : plot titik tunggal bukan segmen garis
addpoints : jika true, plot segmen garis dan titik
add : tambahkan plot ke plot yang sudah ada
user : aktifkan interaksi user untuk fungsi
delta : ukuran langkah untuk interaksi user
bar : plot batang (x = batas interval, y = nilai interval)
histogram : plot frekuensi x dalam n subinterval
distribution=n : plot distribusi x dengan n subinterval
even : gunakan nilai antar untuk histogram otomatis
steps : plot fungsi sebagai fungsi tangga (steps=1,2)
adaptive : gunakan plot adaptif (n = jumlah minimal langkah)
level : plot garis level dari fungsi implisit dua variabel
outline : menggambar batas level.

Jika nilai level berupa matriks 2xn, rentang level akan digambar
dengan warna menggunakan gaya isi yang diberikan. Jika outline true,
akan digambar dengan warna kontur. Fitur ini menandai area f(x,y)
antara batas.

hue : menambahkan warna hue pada 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, memberi lebih banyak ruang di kiri untuk label
vertical :

```
  Aktif/nonaktif label vertikal. Mengubah variabel global
  verticallabels secara lokal untuk satu plot. Nilai 1 = hanya teks
```

vertikal,

```
  nilai 2 = label numerik vertikal pada sumbu y.
```

filled : isi plot kurva
fillcolor : warna isi untuk kurva dan poligon terisi
outline : batas untuk poligon terisi
logplot : atur plot logaritmik

```
          1 = logplot di y,
          2 = logplot di xy,
          3 = logplot di x
```

own :

```
  String menunjuk ke rutinitas plot sendiri. Dengan >user, dapat
```

interaksi sama seperti plot2d. Rentang akan ditetapkan sebelum pemanggilan fungsi.
maps : ekspresi peta (0 lebih cepat), fungsi selalu dipetakan
contourcolor : warna garis kontur
contourwidth : lebar garis kontur
clipping : aktif/nonaktif clipping (default true)
title :

```
  Dapat digunakan untuk mendeskripsikan plot. Judul muncul di atas
```

plot.

```
  Label x dan y dapat ditambahkan dengan xl="string" atau yl="string".
  Label lain dapat ditambahkan dengan fungsi label() atau labelbox().
  Judul bisa berupa string unicode atau gambar formula Latex.
```

cgrid :

```
  Menentukan jumlah garis grid untuk plot grid kompleks.
  Harus pembagi dari ukuran matriks -1 (jumlah subinterval). cgrid =
```

[cx,cy].

Overview

Fungsi dapat memplot:

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

Jika fungsi atau ekspresi untuk xv diberikan, plot2d() menghitung
nilai dalam rentang yang diberikan menggunakan fungsi/ekspresi.
Ekspresi harus dalam variabel x. Rentang ditentukan melalui a dan b kecuali menggunakan default [-2,2].
Rentang-y dihitung otomatis,
kecuali c dan d ditentukan, atau radius r, menghasilkan [-r,r] untuk x dan y.
Untuk fungsi, plot2d menggunakan evaluasi adaptif secara default.
Untuk mempercepat plot fungsi kompleks, matikan <adaptive dan
opsional kurangi jumlah interval n. plot2d() menggunakan mapping
default. Elemen dihitung satu per satu. Jika fungsi mendukung
vektor x, gunakan <maps untuk evaluasi lebih cepat.

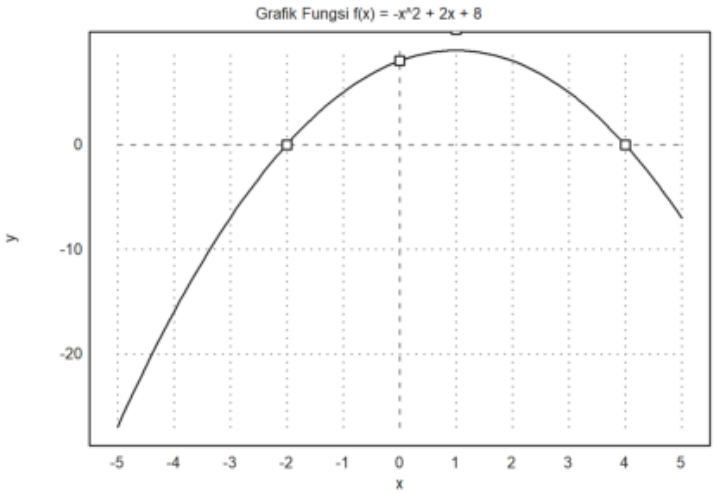
Plot adaptif selalu dihitung elemen per elemen.
Jika fungsi/ekspresi untuk xv dan yv ditentukan, plot2d() menghasilkan
kurva dengan nilai xv sebagai koordinat x dan yv sebagai koordinat y.
Rentang parameter ditentukan dengan xmin, xmax. Ekspresi dalam string
selalu harus dalam variabel parameter x.

Mengerjakan Soal

1. Gambarkan dan tentukan titik potong grafik fungsi dengan sumbu-x dan sumbu-y.

$$f(x) = -x^2 + 2x + 8$$

```
>f = -x.^2 + 2*x + 8;
>x = -5:0.1:5;
>y = f;
>plot2d(x, y, style="-");
>xlabel("x");
>ylabel("y")
>title("Grafik Fungsi f(x) = -x^2 + 2x + 8");
>plot2d([-2, 4, 0, 1], [0, 0, 8, 11],>points,>add)
>insimg
```



2. Gambarlah grafik fungsi

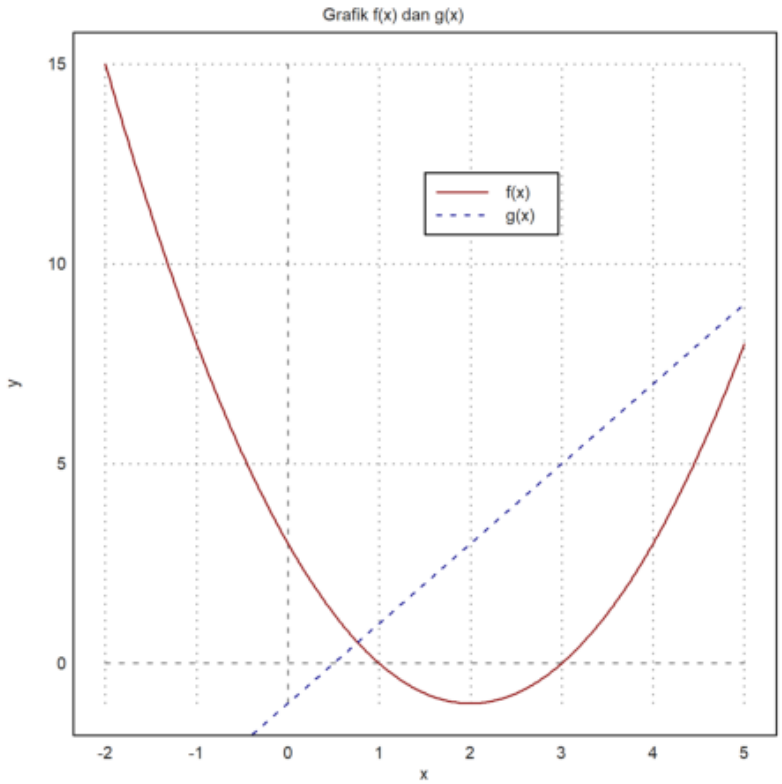
$$f(x) = x^2 - 4x + 3$$

$$g(x) = 2x - 1$$

pada interval

$$x \in [-2, 5]$$

```
>x = linspace(-2,5,400);
>f = x^2 - 4*x + 3;
>g = 2*x - 1;
>plot2d(x, f, color=red, style="-");
>plot2d(x, g, color=blue, style="--", add=true);
>title("Grafik f(x) dan g(x)");
>xlabel("x"); ylabel("y");
>labelbox(["f(x)","g(x)"],colors=[red,blue],styles=["-","--"],x=0.5,y=0.2,>left);
>insimg
```



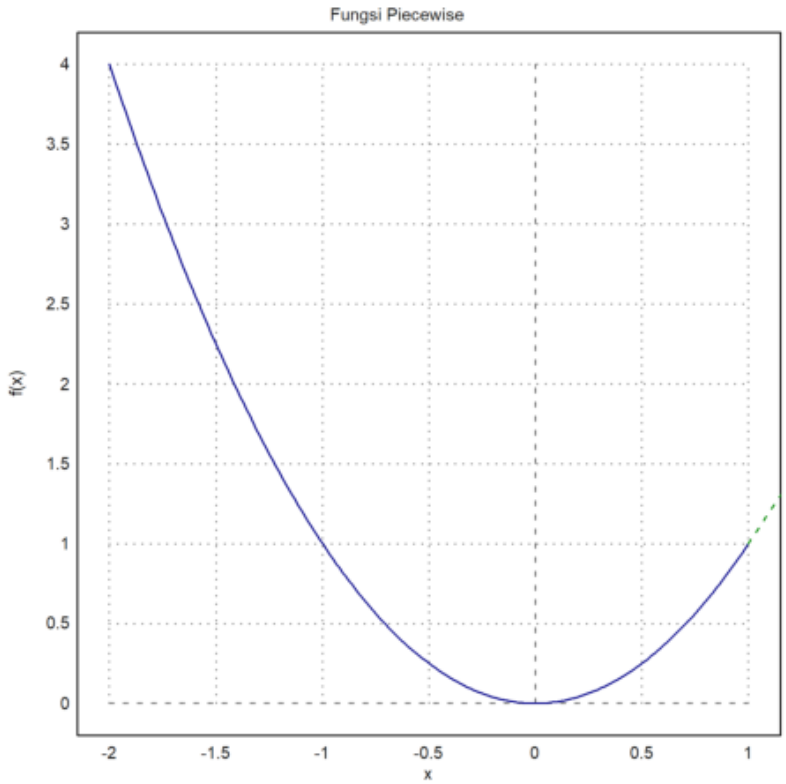
3. Gambarlah fungsi piecewise

$$f(x) = x^2 \text{jika} x < 1, 2x - 1 \text{jika} x \geq 1$$

untuk

$$x \in [-2, 3]$$

```
>x1 = linspace(-2,1,200);
>x2 = linspace(1,3,200);
>f1 = x1^2;
>f2 = 2*x2 - 1;
>plot2d(x1, f1, color=blue, style="-");
>plot2d(x2, f2, color=green, style="--", add=true);
>title("Fungsi Piecewise");
>xlabel("x"); ylabel("f(x)");
>insimg
```


Fungsi Piecewise
f(x)
x
4
3.5
3
2.5
2
1.5
1
0.5
0
-2
-1.5
-1
-0.5
0
0.5
1