Identitas

Nama : Salma Nadia Zulfa
NIM : 24030130005
Prodi : Pendidikan Matematika
Kelas : A

Mata Kuliah Aplikasi Komputer

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

Ada beberapa fungsi dasar dari plot. Terdapat koordinat layar (screen coordinates), yang selalu berkisar dari 0 hingga 1024 pada setiap sumbu, tidak peduli apakah layar berbentuk persegi atau tidak.
elain itu, ada juga koordinat plot (plot coordinates), yang dapat diatur dengan fungsi setplot().

Pemetaan antara kedua koordinat tersebut bergantung pada jendela plot (plot window) yang sedang digunakan. Misalnya, fungsi bawaan shrinkwindow() akan menyisakan ruang untuk label sumbu dan judul plot.

Dalam contoh, kita hanya menggambar beberapa garis acak dengan berbagai warna. Untuk detail lebih lanjut mengenai fungsi-fungsi ini, pelajari 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;
```

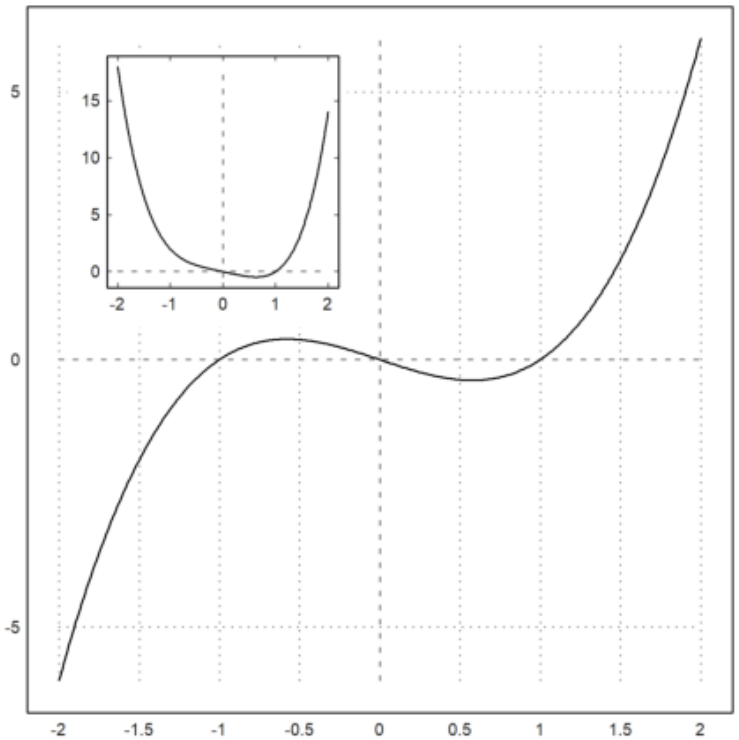
Kita perlu menahan (hold) grafik, karena perintah plot() akan menghapus jendela plot sebelumnya.

ntuk menghapus semua yang telah kita lakukan, gunakan perintah reset().
Untuk menampilkan gambar hasil plot di layar notebook, perintah plot2d() dapat diakhiri dengan titik dua (:).

ara lainnya adalah dengan mengakhiri perintah plot2d() dengan titik koma (;), kemudian menggunakan perintah insimg() untuk menampilkan hasil plot.

Sebagai contoh lain, kita dapat menggambar sebuah plot sebagai inset di dalam plot lain. Hal ini dilakukan dengan mendefinisikan jendela plot yang lebih kecil. Perlu diperhatikan bahwa jendela ini tidak menyediakan ruang untuk label sumbu di luar area plot. Oleh karena itu, kita harus menambahkan margin seperlunya.
elain itu, kita juga menyimpan dan mengembalikan jendela penuh, serta menahan plot yang sedang aktif ketika kita membuat inset plot.

```
>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);
```

Sebuah plot dengan beberapa gambar (multiple figures) dapat dibuat dengan cara yang sama. Untuk itu tersedia fungsi bantu figure().

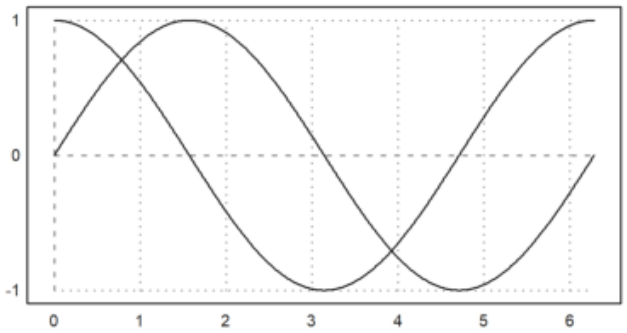
Aspek Plot

Secara default, plot menggunakan jendela berbentuk persegi. Anda dapat mengubahnya dengan fungsi aspect(). Jangan lupa untuk mereset kembali aspek tersebut setelah selesai.

elain itu, Anda juga dapat mengubah pengaturan default melalui menu "Set Aspect", baik ke rasio aspek tertentu maupun sesuai ukuran jendela grafik saat ini.

Namun, Anda juga bisa mengubah aspek ini hanya untuk satu plot saja. Dalam hal ini, ukuran area plot akan disesuaikan, dan jendela akan diatur agar label memiliki cukup ruang.

```
>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 memiliki kemampuan untuk membuat plot 2D, baik untuk data maupun fungsi. EMT menggunakan fungsi plot2d. Fungsi ini dapat digunakan untuk memplot fungsi maupun data.

Selain itu, plotting juga bisa dilakukan di Maxima menggunakan Gnuplot, atau di Python menggunakan Matplotlib.

Euler dapat membuat plot 2D untuk:

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

Gaya plot meliputi berbagai gaya untuk garis dan titik, plot batang (bar plots), dan plot arsiran (shaded plots).

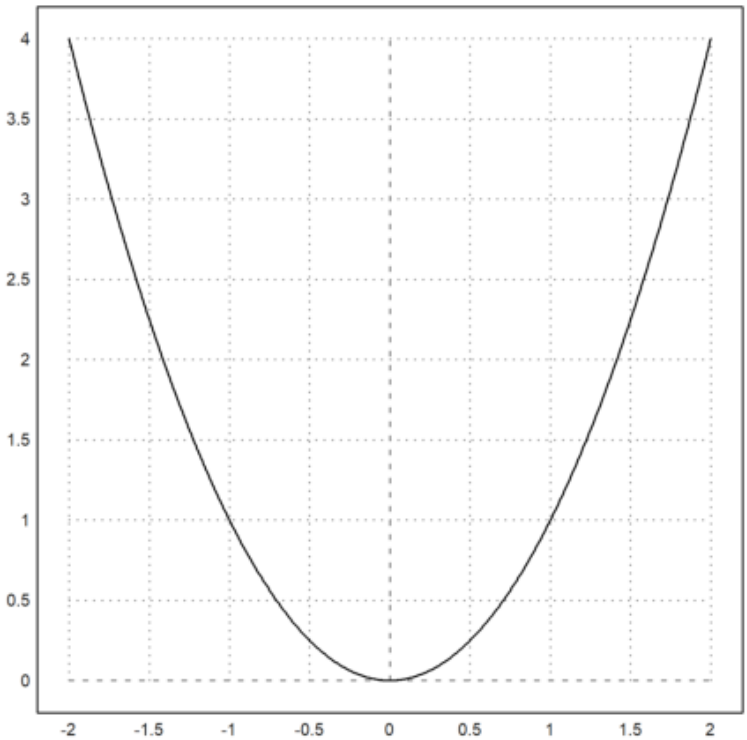
Plot Ekspresi atau Variabel

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

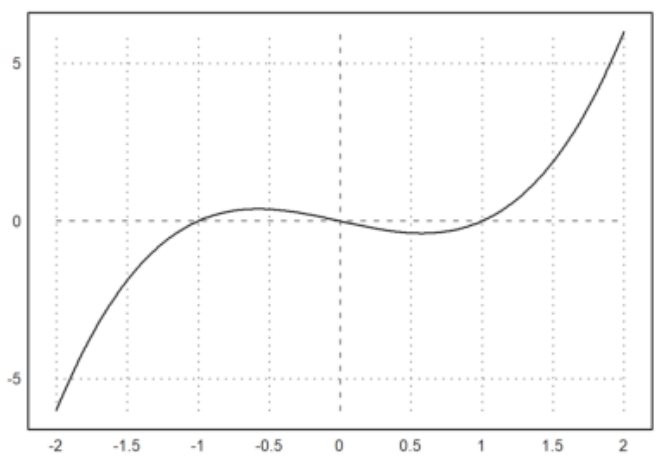
ontoh paling dasar menggunakan rentang default, dan secara otomatis menyesuaikan rentang y agar sesuai dengan grafik fungsi.

Catatan: Jika sebuah perintah diakhiri dengan titik dua (:), maka hasil plot akan ditampilkan di jendela teks. Jika tidak, tekan TAB untuk melihat plot jika jendela plot tertutup.

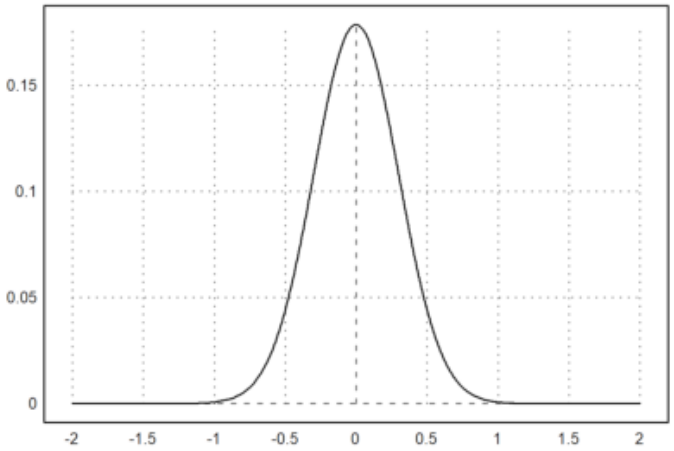
```
>plot2d("x^2"):
```




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



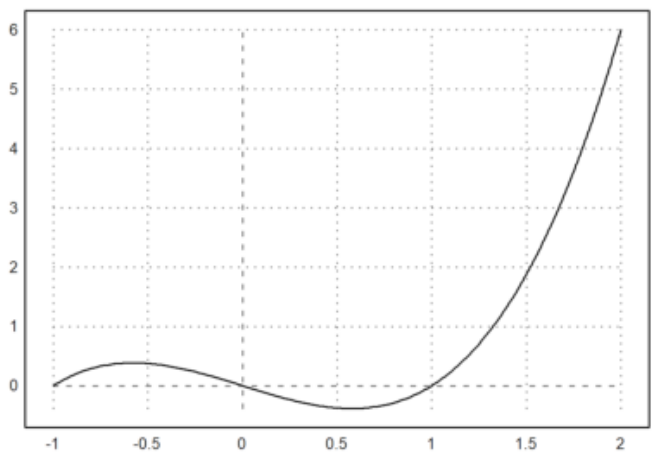

```
>a:=5.6; plot2d("exp(-a*x^2)/a"); insimg(30); // menampilkan gambar hasil plot setinggi 25
```



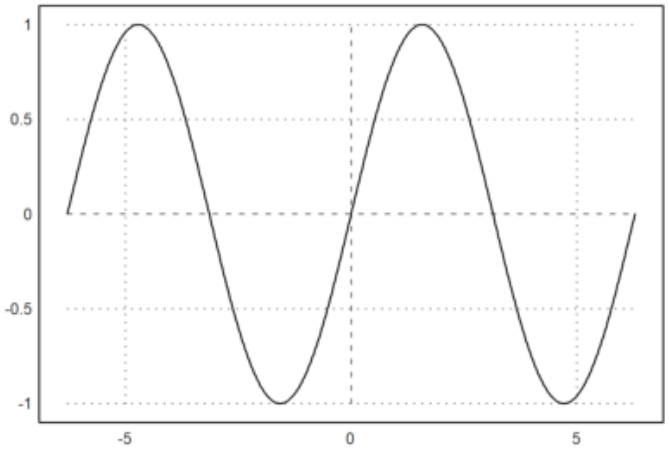
Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

Rentang plot dapat diatur dengan parameter berikut:
- a, b : rentang sumbu-x (default: -2 hingga 2)
- c, d : rentang sumbu-y (default: menyesuaikan dengan nilai)
- r : alternatifnya berupa jari-jari di sekitar pusat plot
- cx,cy : koordinat pusat plot (default: 0,0)

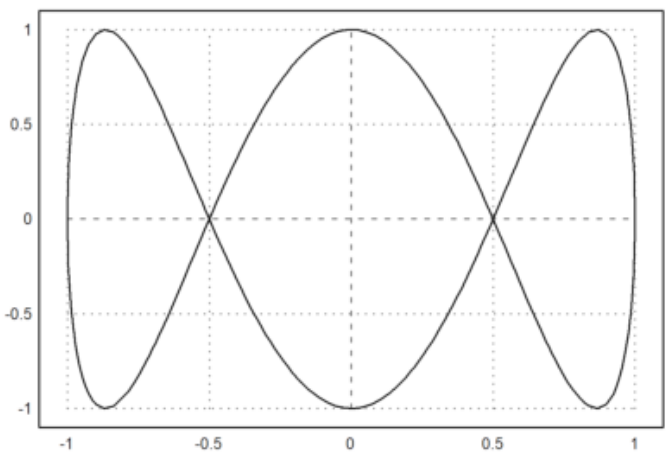
```
>plot2d("x^3-x",-1,2):
```




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




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



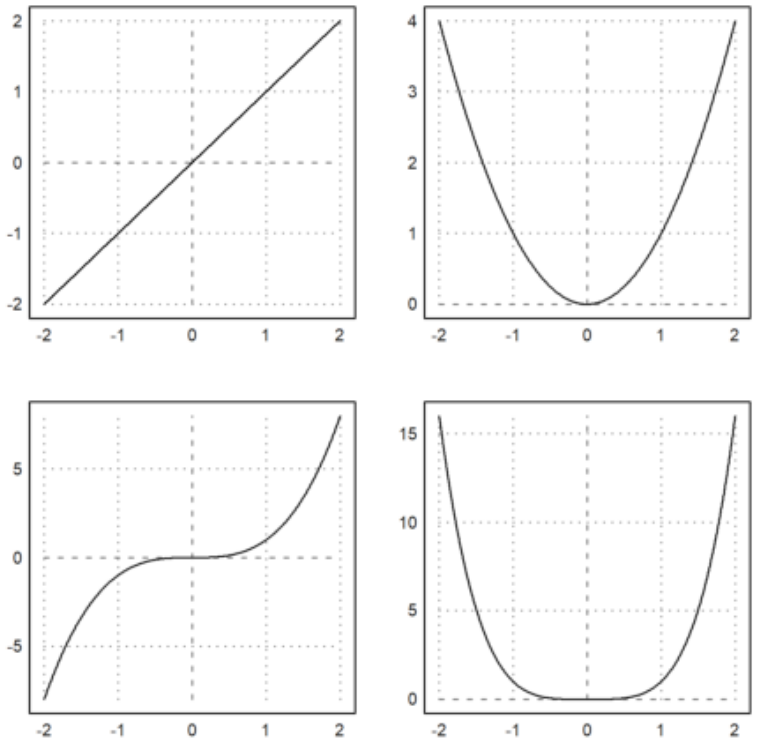
Alternatif lain dari penggunaan titik dua (:) adalah perintah insimg(lines), yang menyisipkan plot dengan ukuran sesuai jumlah baris teks yang ditentukan.

Pada opsi pengaturan, plot dapat ditampilkan:
- di jendela terpisah yang dapat diubah ukurannya,
- di dalam jendela notebook.

Lebih banyak gaya plot dapat dihasilkan dengan perintah plot yang spesifik.

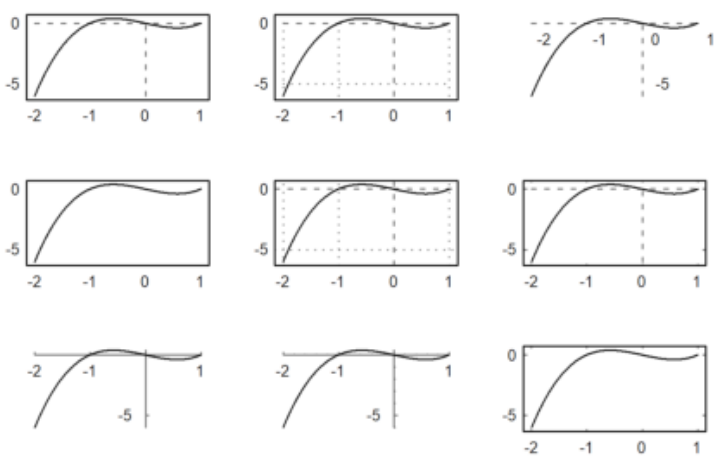
Dalam semua kasus, tekan tombol Tab untuk melihat plot jika tertutup.
Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure().
Sebagai contoh, kita dapat memplot x^1 hingga x^4 ke dalam 4 bagian jendela.
Perintah figure(0) akan mengembalikan ke jendela default.

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



Dalam plot2d(), terdapat gaya alternatif yang dapat digunakan dengan opsi grid=x.
Sebagai gambaran, berbagai gaya grid dapat ditampilkan dalam satu gambar (lihat penggunaan perintah figure() di bawah).
Gaya grid=0 tidak termasuk, karena gaya ini 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):
```

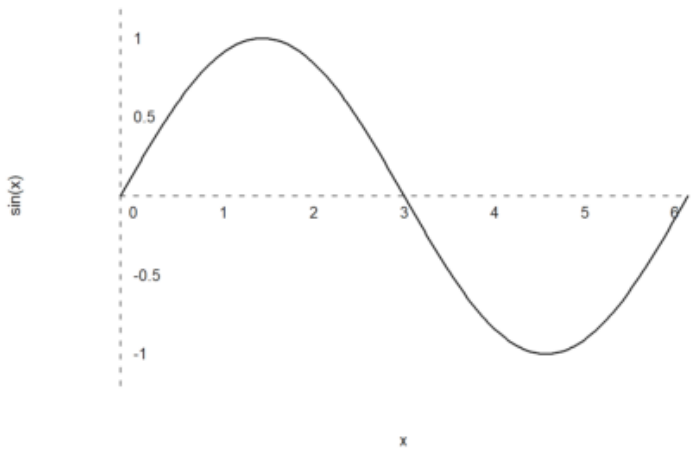


Jika argumen untuk plot2d() berupa sebuah ekspresi yang diikuti oleh empat angka, maka angka-angka tersebut adalah rentang x dan y untuk plot.

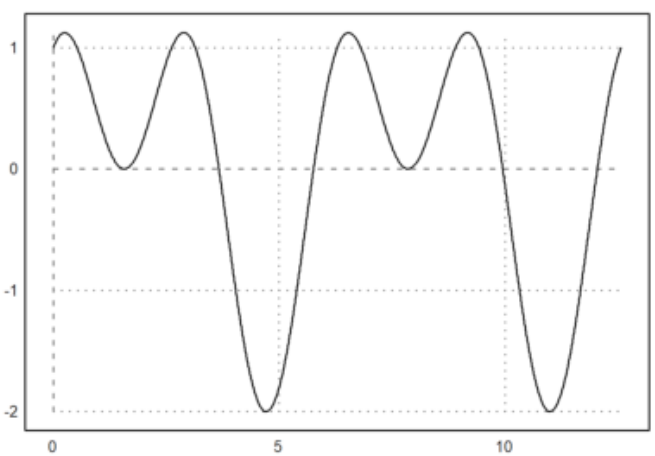
Sebagai alternatif, parameter a, b, c, d dapat ditentukan dengan cara assigned parameters seperti a=... dan seterusnya.

Pada contoh berikut, kita mengubah gaya grid, menambahkan label, 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):
```

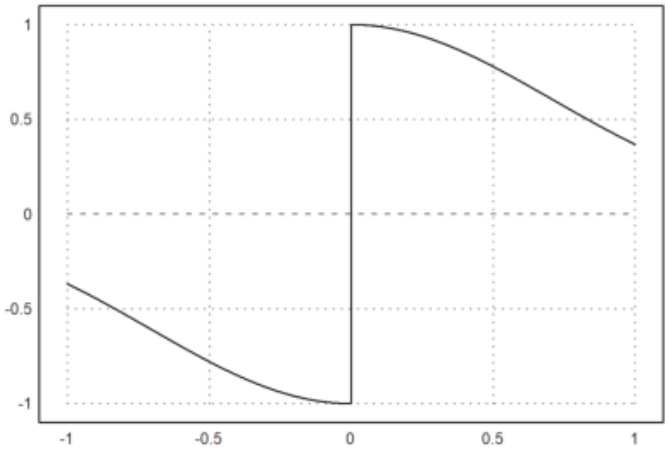


Gambar yang dihasilkan dengan menyisipkan plot ke dalam jendela teks secara otomatis disimpan di direktori yang sama dengan notebook, biasanya di dalam subdirektori bernama "images". Gambar-gambar ini juga digunakan pada saat melakukan ekspor HTML.

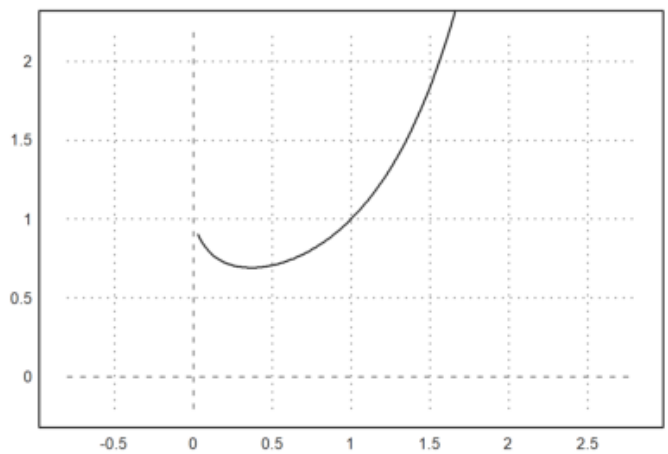
Anda dapat menandai gambar apa pun dan menyalinnya ke clipboard dengan Ctrl + C. Tentu saja, Anda juga dapat mengekspor grafik yang sedang aktif menggunakan fungsi pada menu File.

Fungsi atau ekspresi dalam plot2d dievaluasi secara adaptif. Untuk mendapatkan kecepatan lebih tinggi, mode adaptif dapat dimatikan dengan <adaptive, dan jumlah subinterval dapat ditentukan menggunakan n=.... Namun, pengaturan ini biasanya hanya diperlukan pada kasus-kasus tertentu saja.

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



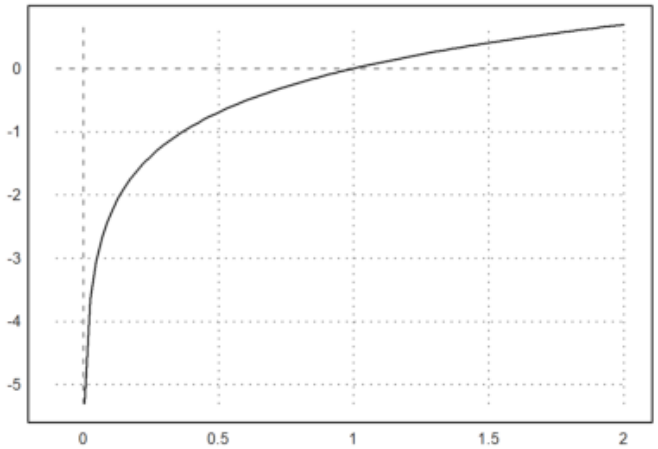

```
>plot2d("x^x",r=1.2,cx=1,cy=1):
```



Perlu diperhatikan bahwa x^x tidak terdefinisi untuk x<=0.
ungsi plot2d akan menangkap kesalahan ini, dan mulai membuat plot segera setelah fungsi tersebut terdefinisi.

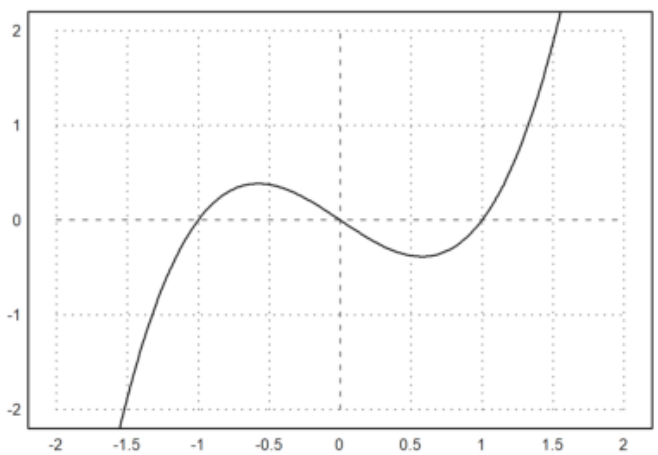
Hal ini berlaku untuk semua fungsi yang menghasilkan NAN (Not a Number) di luar daerah definisinya.

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

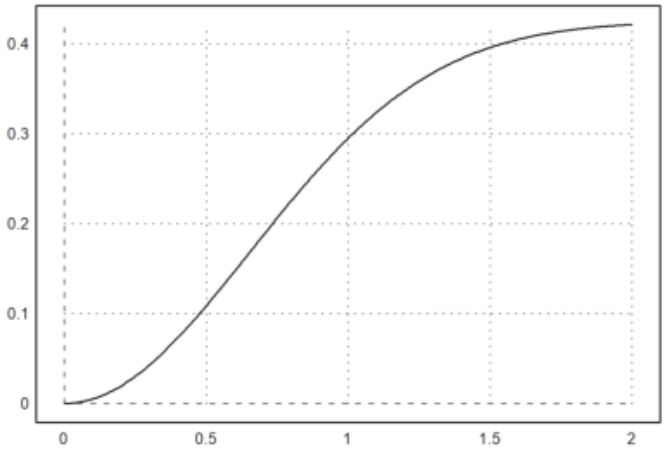


Parameter square=true (atau >square) akan secara otomatis menyesuaikan rentang sumbu-y sehingga hasilnya berupa jendela plot berbentuk persegi.
erlu diperhatikan bahwa secara default, Euler memang menggunakan ruang berbentuk persegi di dalam jendela plot.

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

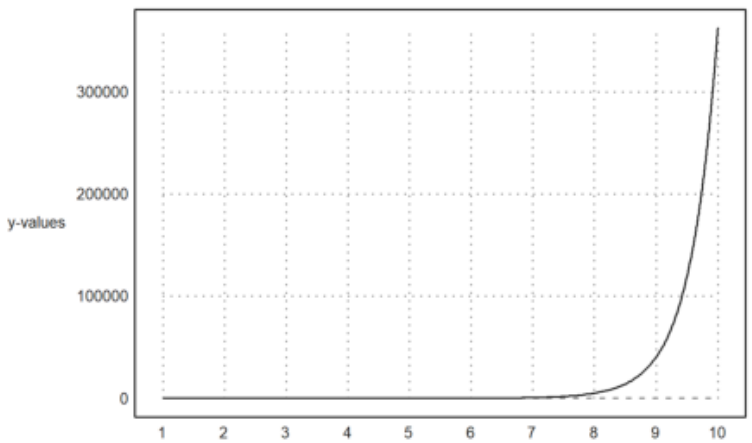



```
>plot2d(''integrate("sin(x)*exp(-x^2)",0,x)'',0,2): // plot integral
```



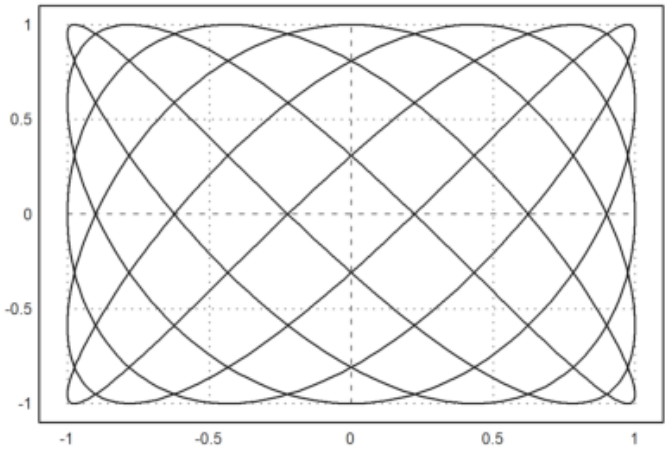
Jika Anda membutuhkan lebih banyak ruang untuk label sumbu-y, gunakan perintah shrinkwindow() dengan parameter smaller yang lebih kecil, atau tetapkan nilai positif untuk "smaller" di dalam plot2d().

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

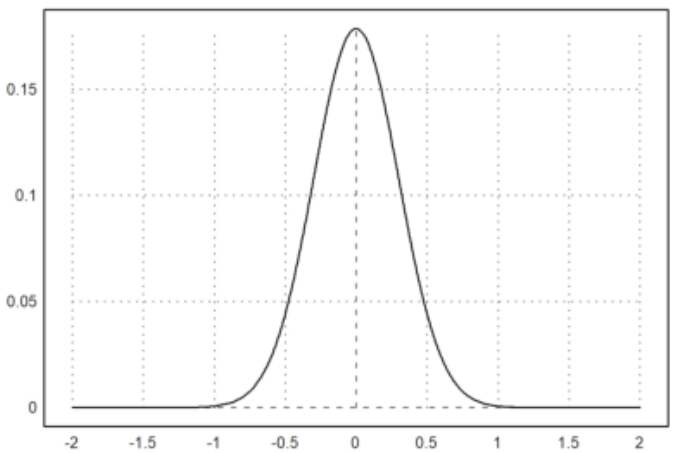


Ekspresi simbolik juga dapat digunakan, karena ekspresi tersebut disimpan sebagai string sederhana.

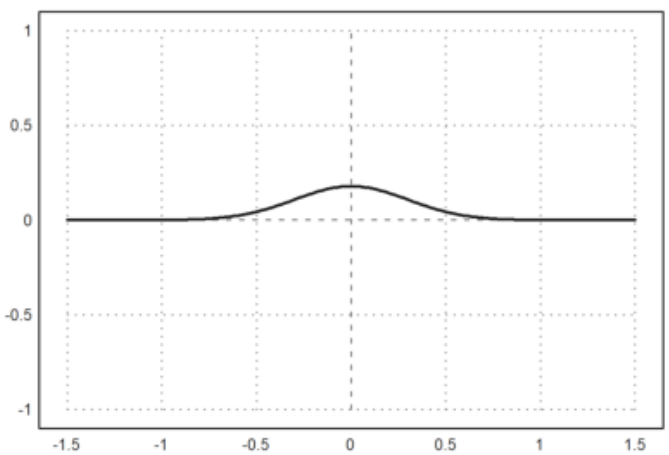
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



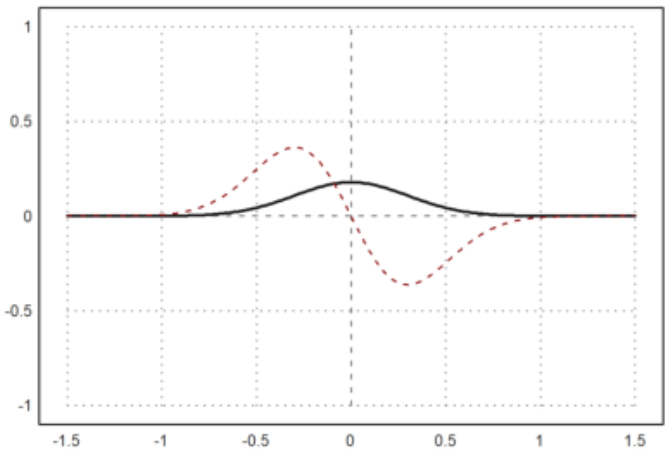

```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression
>plot2d(expr,-2,2): // plot from -2 to 2
```



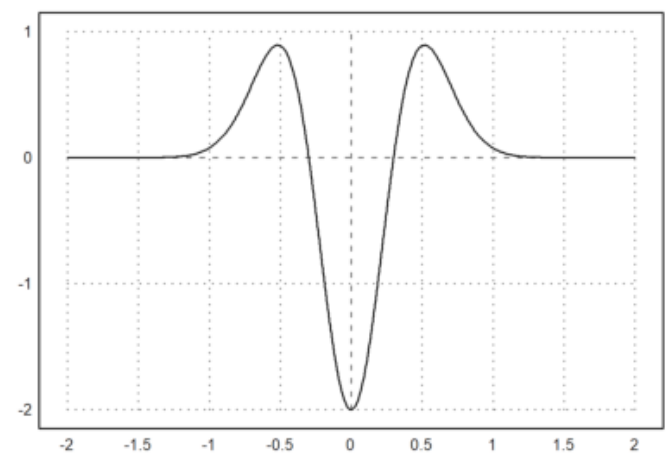

```
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)
```



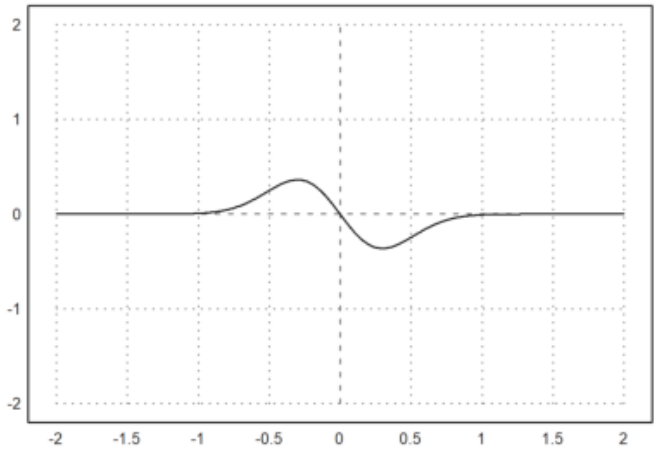
```
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot
```



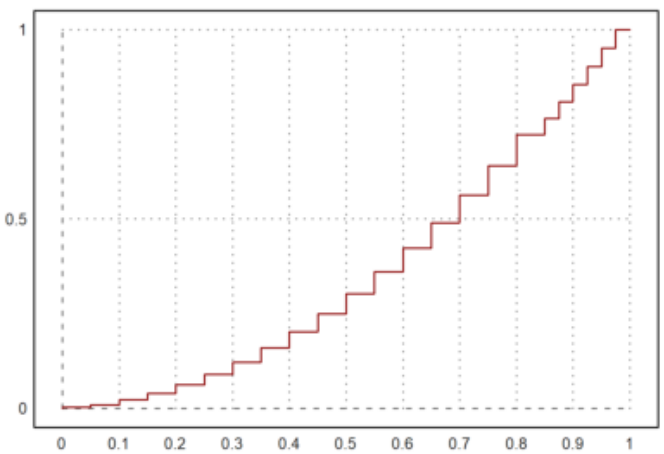

```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle
```



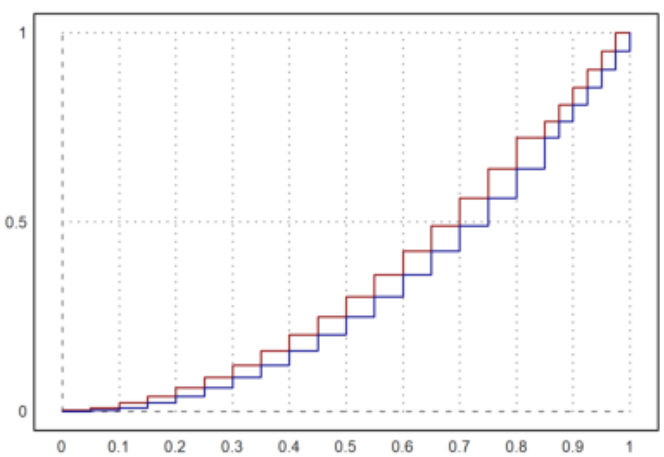

```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square
```




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




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

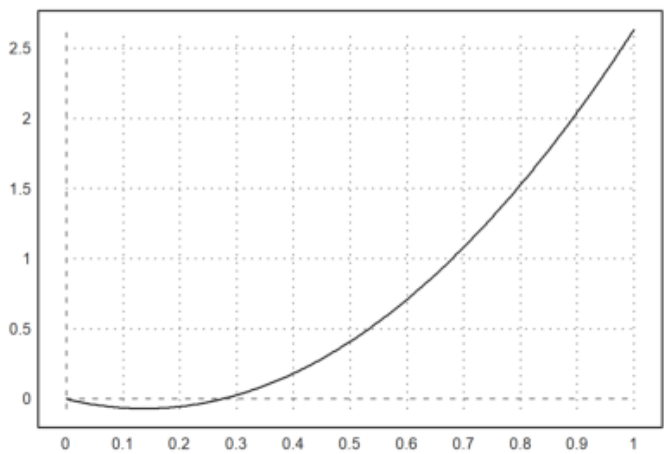


Fungsi dengan Satu Parameter

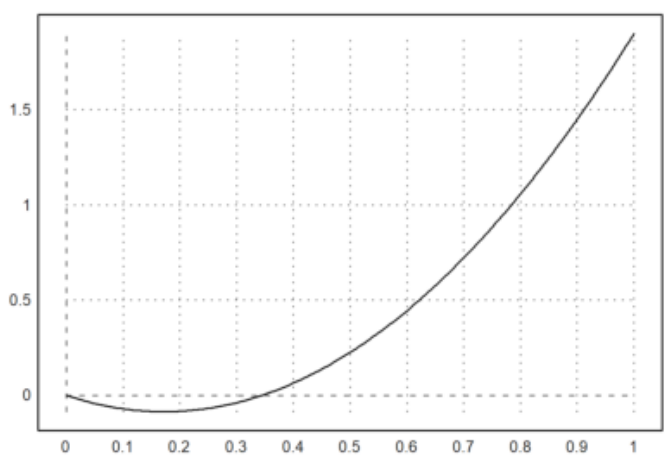
Fungsi plotting terpenting untuk plot bidang (planar plots) adalah plot2d().Fungsi ini diimplementasikan dalam bahasa Euler pada file "plot.e", yang dimuat saat program dijalankan.

Berikut adalah beberapa contoh penggunaan dengan sebuah fungsi.
eperti biasa di EMT, fungsi yang bekerja untuk fungsi lain atau ekspresi dapat diberikan parameter tambahan (selain x). Parameter ini bukan merupakan variabel global, dan dapat dilewatkan menggunakan parameter dengan titik koma (semicolon parameters) atau dengan call collection.

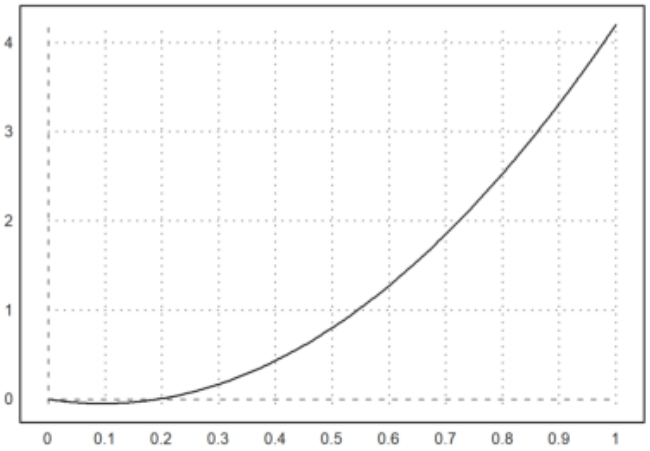
```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
>a=0.3; plot2d("f",0,1;a): // plot with a=0.3
```



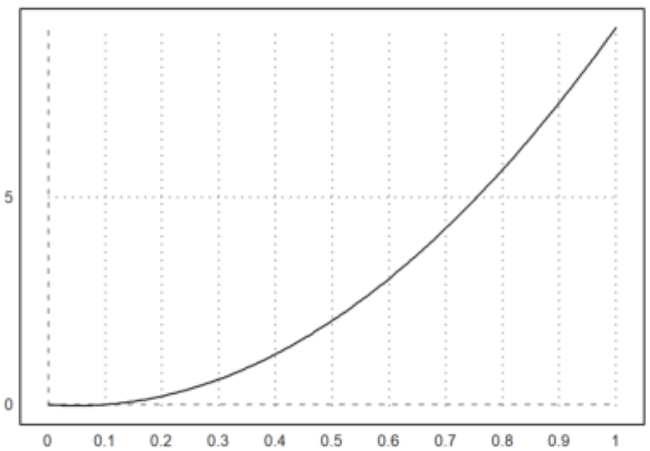

```
>plot2d("f",0,1;0.4): // plot with a=0.4
```



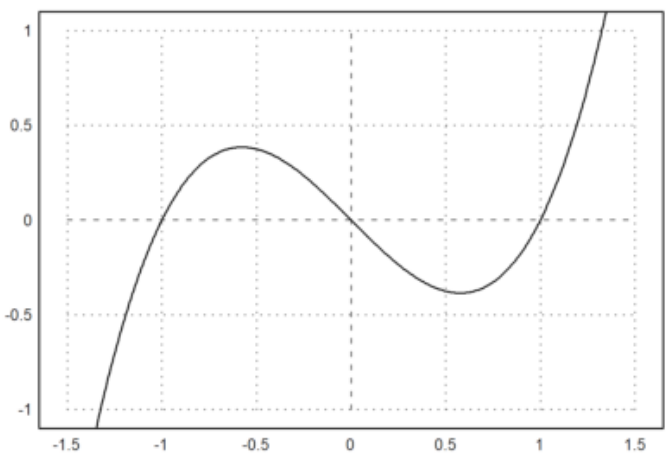

```
>plot2d({{"f",0.2}},0,1): // plot with a=0.2
```




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




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



Ringkasan Fungsi yang Dapat Diterima

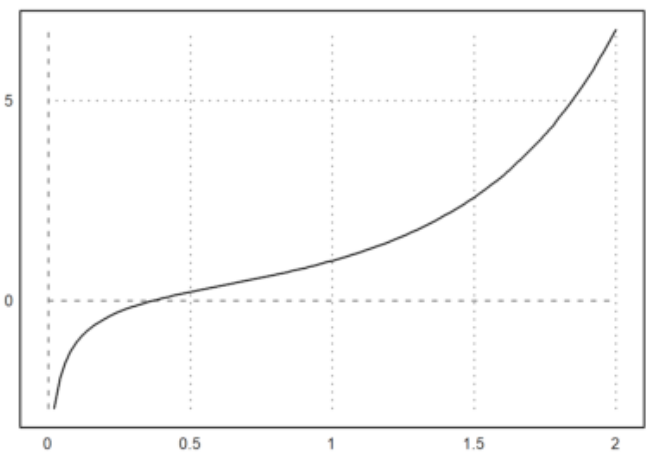
- Ekspresi atau ekspresi simbolik dalam variabel x
- Fungsi atau fungsi simbolik dengan menyebutkan namanya, misalnya "f"
- Fungsi simbolik cukup dengan menuliskan namanya saja, misalnya f

Fungsi plot2d() juga mendukung fungsi simbolik. Untuk fungsi simbolik, cukup dengan menuliskan namanya langsung.

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

$$x \quad x \left(\log(x) + 1\right)$$

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

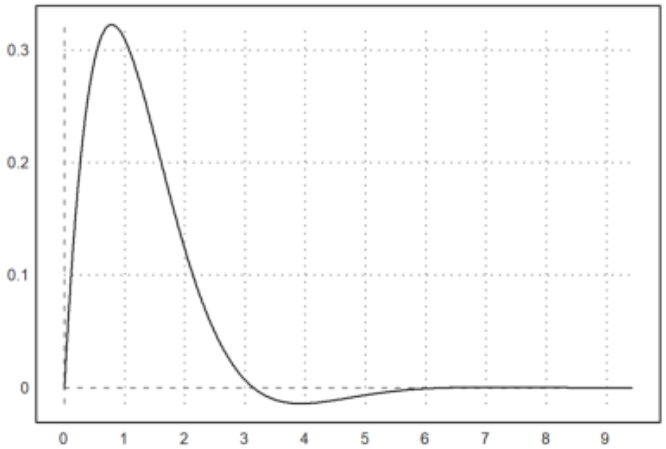


Tentu saja, untuk ekspresi atau ekspresi simbolik, cukup dengan menyebutkan nama variabelnya saja untuk dapat memplotnya.

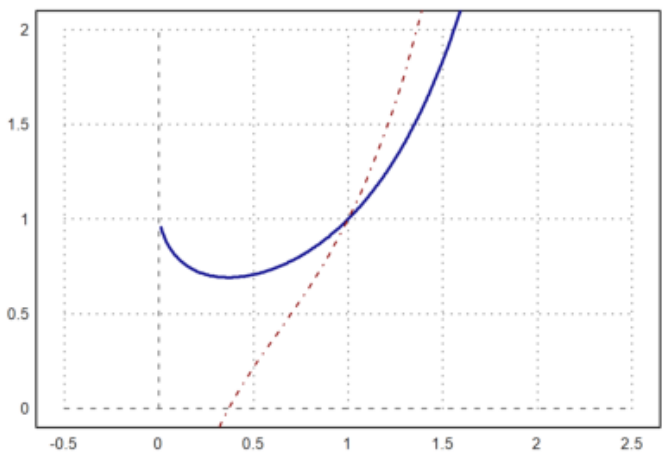
```
>expr &= sin(x)*exp(-x)
```

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

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

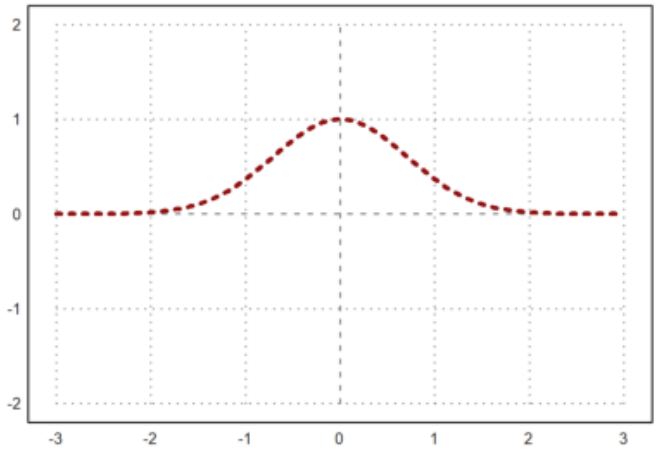


Gaya Garis (Line Style)

Terdapat berbagai opsi untuk gaya garis:
- style="..." : pilih dari "-", "–", "-.", ".", ".-.", "-.-".
- color : lihat daftar warna di bawah.
- thickness : ketebalan garis (default = 1).

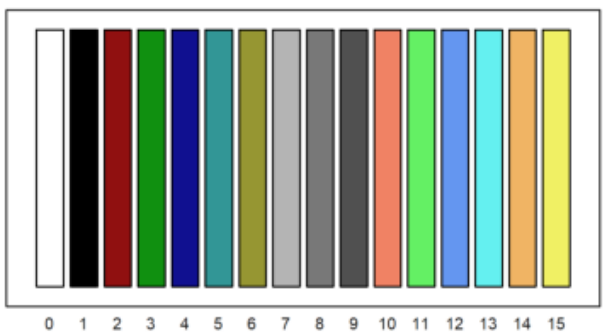
Warna dapat dipilih dari warna bawaan atau menggunakan warna RGB.
- 0..15 : indeks warna bawaan.
- konstanta warna: white, black, red, green, blue, cyan, olive, lightgray, gray, darkgray, orange, lightgreen, turquoise, lightblue, lightorange, yellow.
- rgb(red,green,blue) : parameter berupa bilangan real dalam rentang [0,1].

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



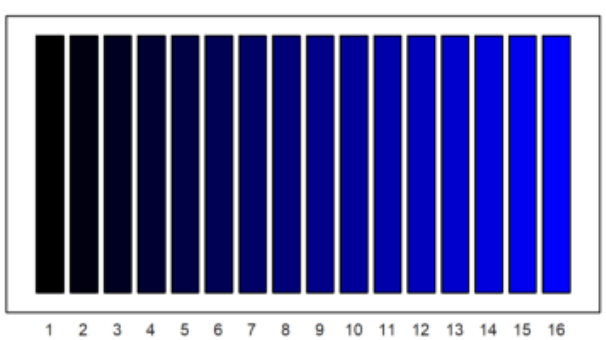
Berikut adalah tampilan warna bawaan (predefined colors) yang tersedia di EMT.

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



Namun, Anda dapat menggunakan warna apa pun.

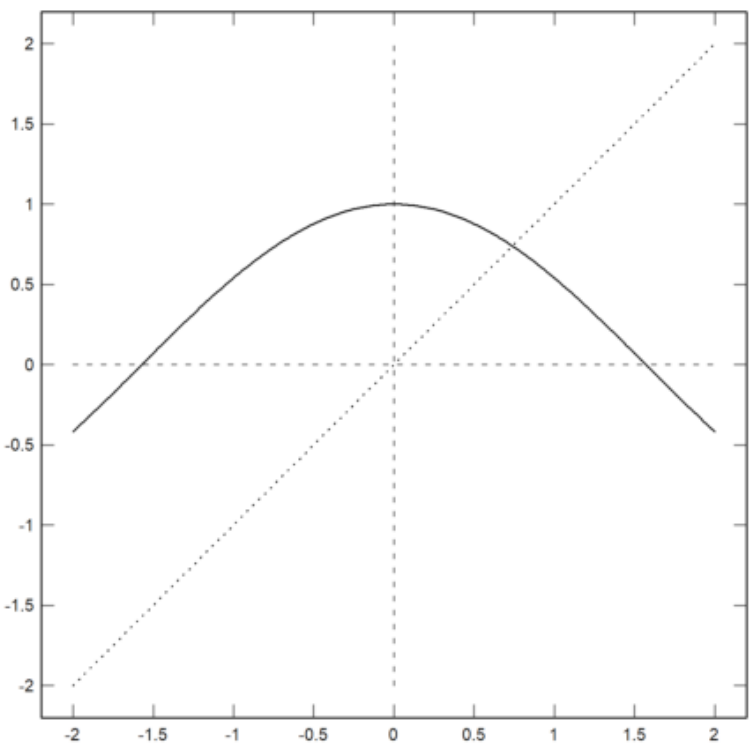
```
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```



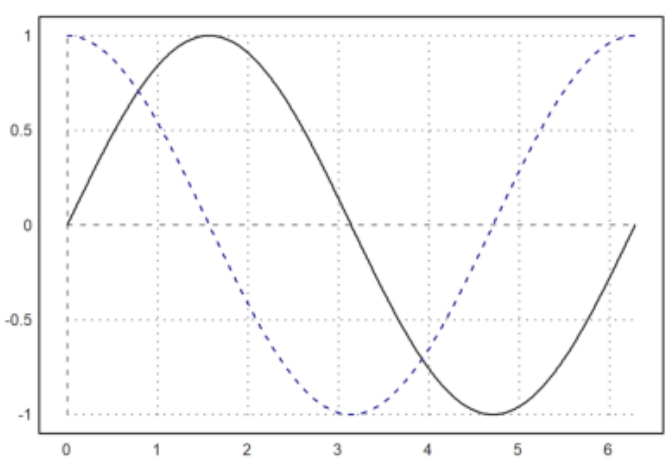
Menggambar Beberapa Kurva pada bidang koordinat yang sama

Untuk memplot lebih dari satu fungsi (multiple functions) dalam satu jendela dapat dilakukan dengan beberapa cara.
alah satu caranya adalah dengan menggunakan >add pada beberapa pemanggilan plot2d, kecuali pada pemanggilan pertama.
itur ini sebenarnya sudah kita gunakan pada contoh-contoh sebelumnya.

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

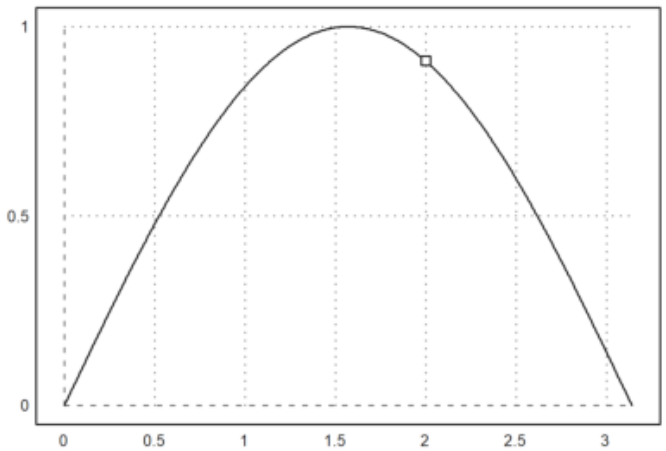



```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



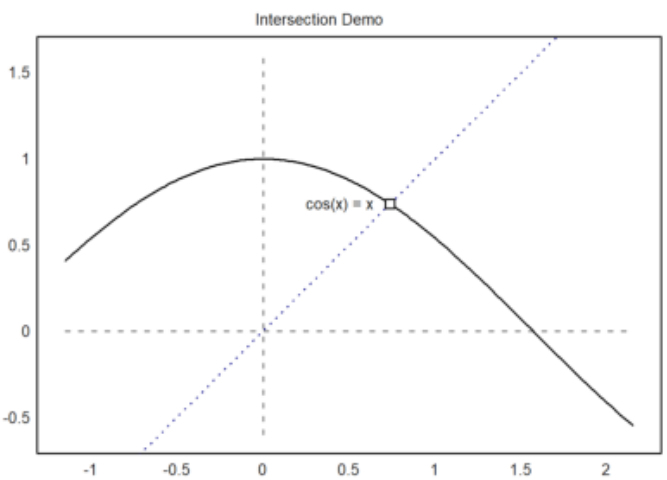
Salah satu kegunaan >add adalah untuk menambahkan titik pada kurva.

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



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

```
>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 ke-8 dan ke-16. Ekspansi ini dihitung menggunakan Maxima melalui ekspresi simbolik.

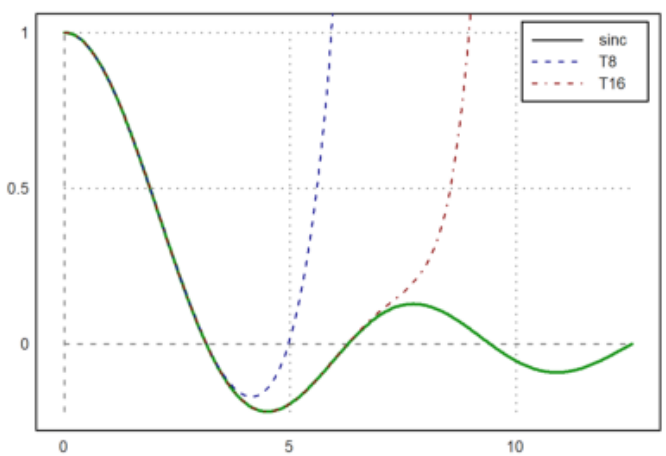
Plot ini dibuat dengan perintah multi-baris yang memanggil plot2d() sebanyak tiga kali.
ada pemanggilan kedua dan ketiga, digunakan flag >add, sehingga plot tersebut menggunakan rentang yang sama dengan plot sebelumnya.

Kita juga menambahkan kotak label untuk menjelaskan fungsi-fungsi yang ditampilkan.

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

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

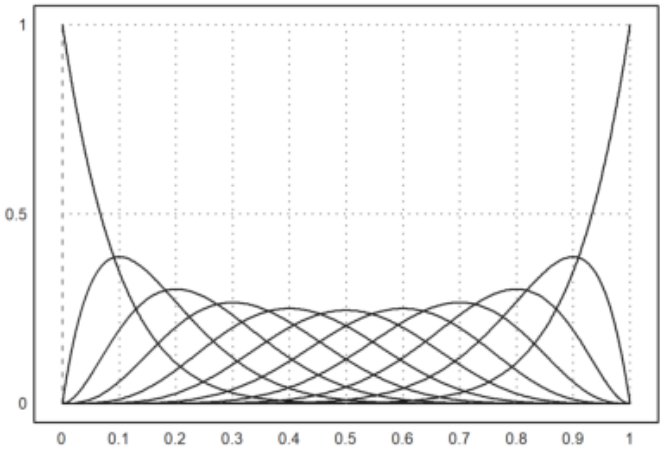
```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
>  plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
>  plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-.-"); ...
>  labelbox(["sinc","T8","T16"],styles=["-","--","-.-"], ...
>    colors=[black,blue,red]):
```



Pada contoh berikut, kita akan menghasilkan Polinomial Bernstein.

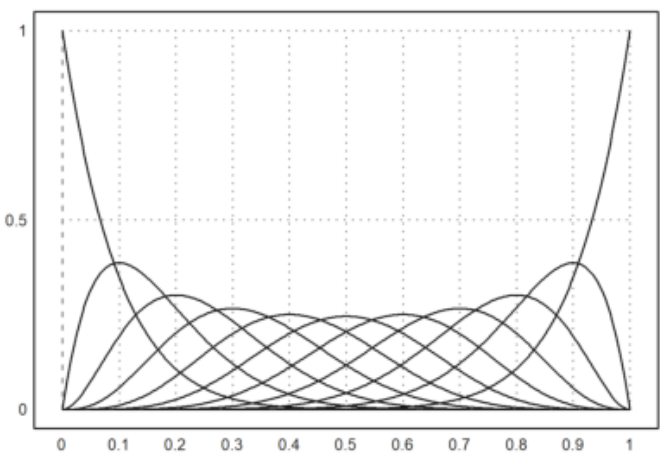
$$B_i(x)=\binom{n}{i}x^i(1-x)^{n-i}$$

```
>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;
```



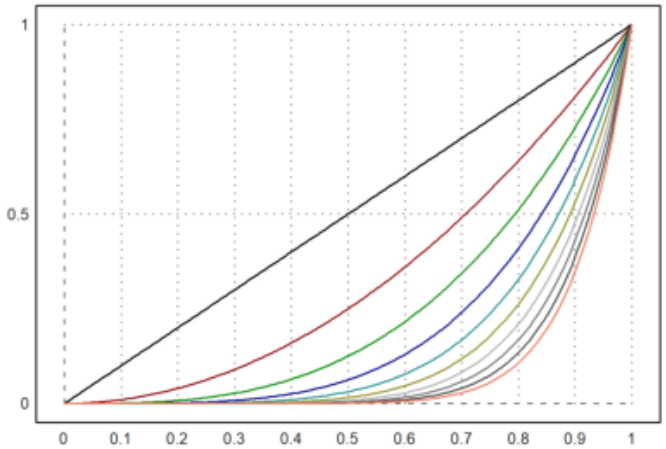
Metode kedua adalah dengan menggunakan sepasang matriks nilai x dan matriks nilai y dengan ukuran yang sama.
Kita membangkitkan sebuah matriks nilai dengan satu Polinomial Bernstein di setiap baris.
ntuk ini, kita cukup menggunakan vektor kolom i.
Untuk penjelasan lebih detail, silakan lihat kembali bagian pengantar bahasa matriks.

```
>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):
```



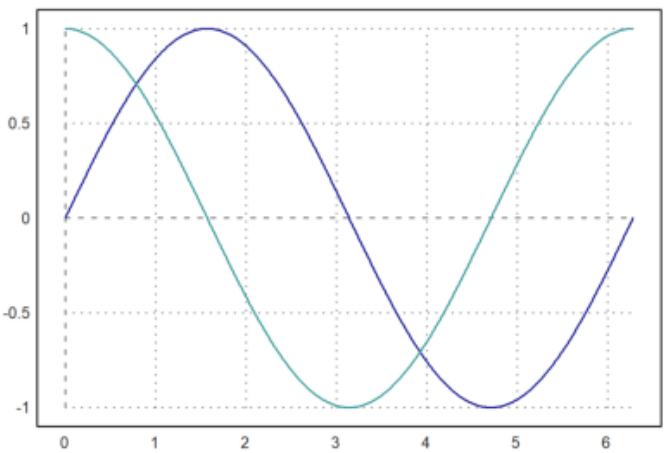
Perlu diperhatikan bahwa parameter color dapat berupa sebuah vektor.
Dengan demikian, setiap warna akan digunakan untuk setiap baris matriks.

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

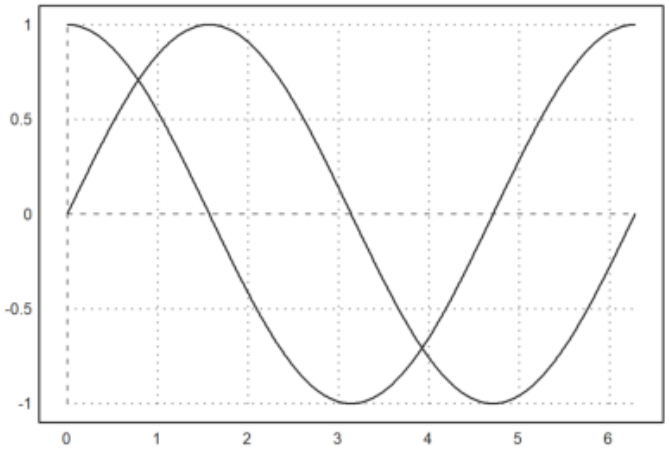


Metode lain adalah dengan menggunakan vektor ekspresi (string).
engan cara ini, Anda dapat menggunakan array warna, array gaya garis, dan array ketebalan dengan panjang yang sama.

```
>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 fungsi makelist() dan mxm2str().

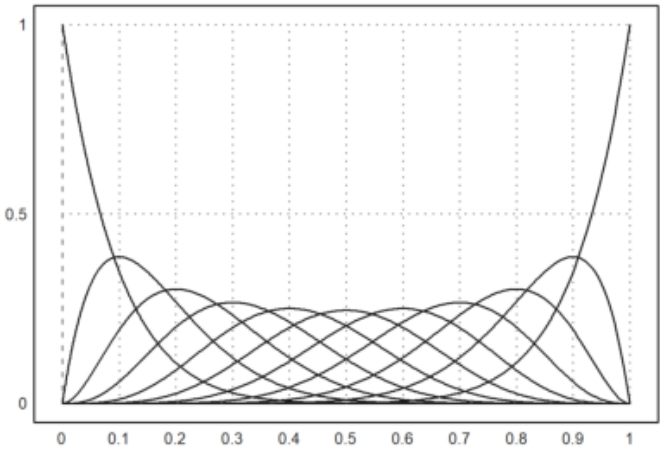
```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // make list
```

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

```
>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
```

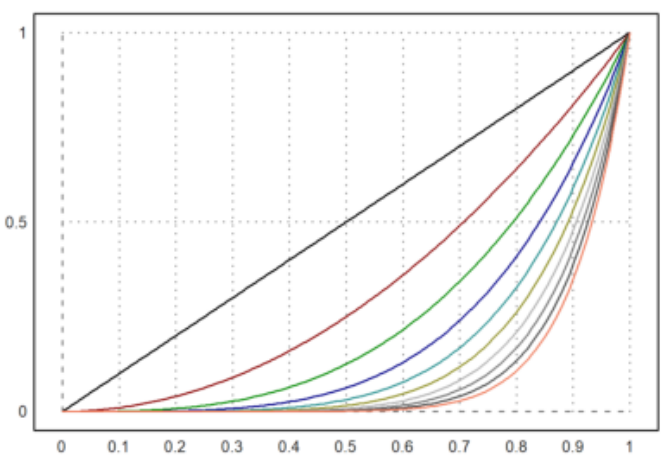
```
>plot2d(mxm2str(v),0,1): // plot functions
```



Alternatif lain adalah dengan menggunakan bahasa matriks pada Euler.
Jika sebuah ekspresi menghasilkan matriks fungsi, dengan satu fungsi di setiap baris, maka semua fungsi tersebut akan dipetakan ke dalam satu plot.

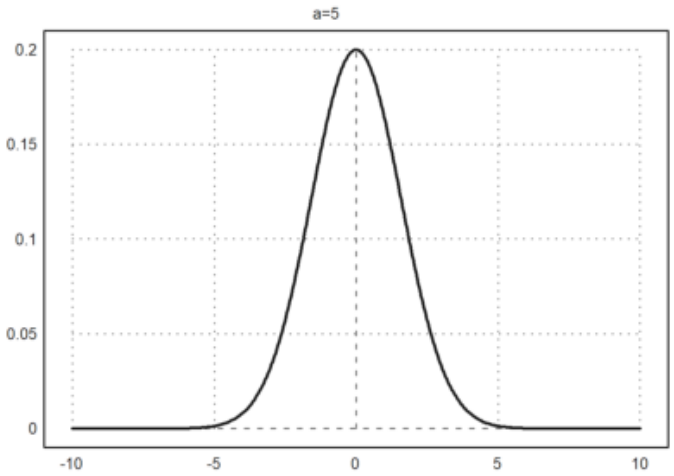
Untuk hal ini, gunakan vektor parameter dalam bentuk vektor kolom. Jika ditambahkan array warna, maka warna tersebut akan digunakan untuk setiap baris pada plot.

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



Ekspresi dan fungsi satu-baris (one-line functions) dapat melihat variabel global.
Jika Anda tidak bisa menggunakan variabel global, maka Anda perlu membuat fungsi dengan parameter tambahan, lalu mengoper parameter tersebut sebagai parameter titik koma (semicolon parameter).
Perhatikan bahwa semua parameter terikat (assigned parameters) harus diletakkan di bagian akhir perintah plot2d.
ebagai contoh, kita memberikan a=5 ke fungsi f, yang kemudian diplot dari -10 hingga 10.

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

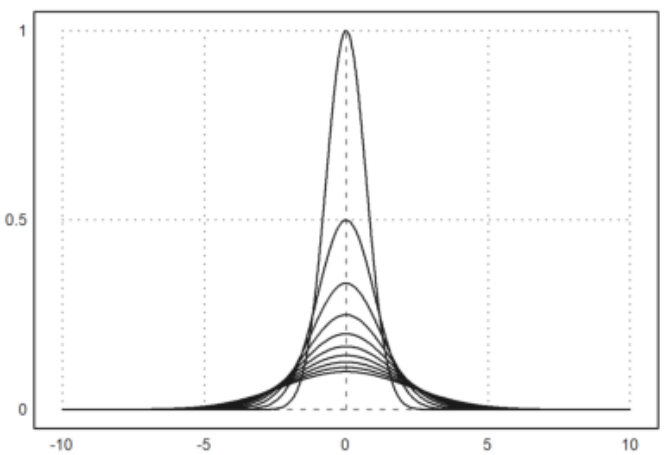


Sebagai alternatif, gunakan collection berisi nama fungsi dan semua parameter tambahan.

Daftar khusus ini disebut call collections, dan cara ini merupakan metode yang lebih disarankan untuk mengoper argumen ke sebuah fungsi yang juga diteruskan sebagai argumen ke fungsi lain.

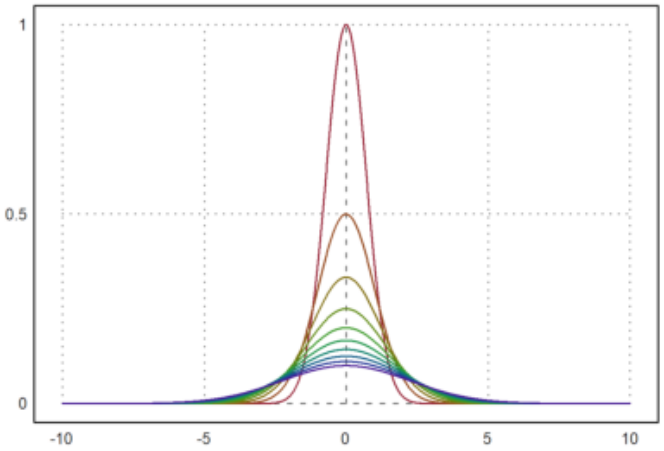
Pada contoh berikut, kita menggunakan loop untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman dengan loop).

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



Kita dapat mencapai hasil yang sama dengan cara berikut, menggunakan bahasa matriks EMT.
Setiap baris dari matriks
(x,a) merupakan satu fungsi.
Selain itu, kita juga dapat mengatur warna untuk setiap baris matriks.
Klik ganda pada fungsi getspectral() untuk penjelasan lebih lanjut.

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

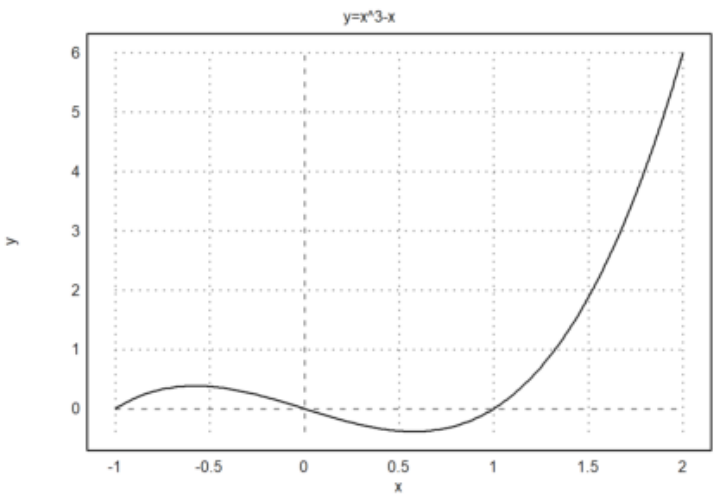


Text Labels

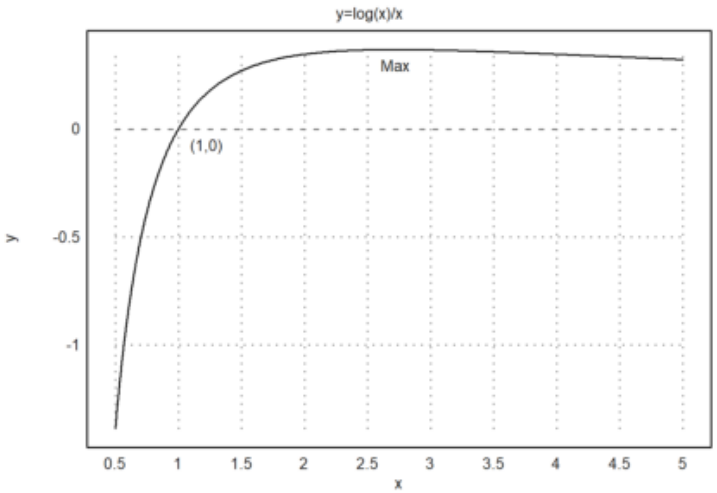
Dekorasi sederhana dapat berupa:
- judul dengan title="..."
- label sumbu-x dan sumbu-y dengan xl="...", yl="..."
- teks label lain dengan label("...", x, y)

Perintah label akan menambahkan teks ke dalam plot aktif pada koordinat plot (x,y).
Label ini juga dapat diberi posisi tertentu.

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

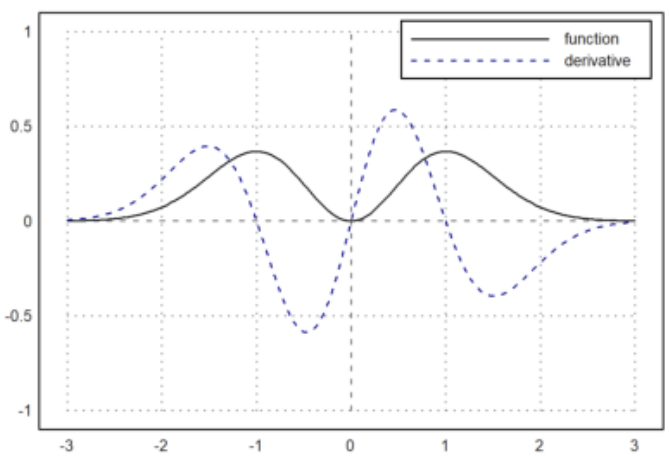



```
>expr := "log(x)/x"; ...
>  plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
>  label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```



Terdapat juga fungsi labelbox(), yang dapat menampilkan daftar fungsi beserta teksnya.
ungsi ini menerima vektor string dan vektor warna, masing-masing satu 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):
```

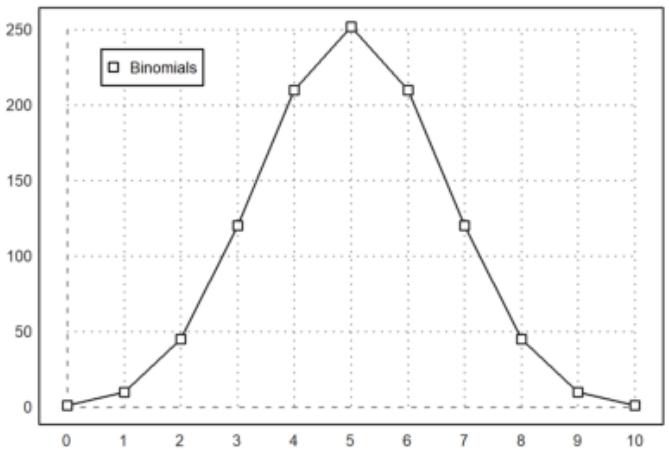


Kotak label secara default berada di sudut kanan atas, tetapi dengan opsi >left kotak dapat diposisikan di sudut kiri atas.
nda juga bisa memindahkannya ke posisi mana pun sesuai keinginan. Titik jangkar (anchor position) adalah sudut kanan atas kotak, dan nilai koordinatnya berupa pecahan dari ukuran jendela grafik. Lebarnya akan disesuaikan secara otomatis.

Untuk plot titik (point plots), fungsi kotak label juga dapat digunakan.
ambahkan parameter >points, atau gunakan vektor flag, satu untuk setiap label.

Pada contoh berikut, hanya ada satu fungsi. Karena itu, kita bisa menggunakan string tunggal sebagai ganti vektor string.
alam contoh ini, warna teks diatur menjadi hitam.

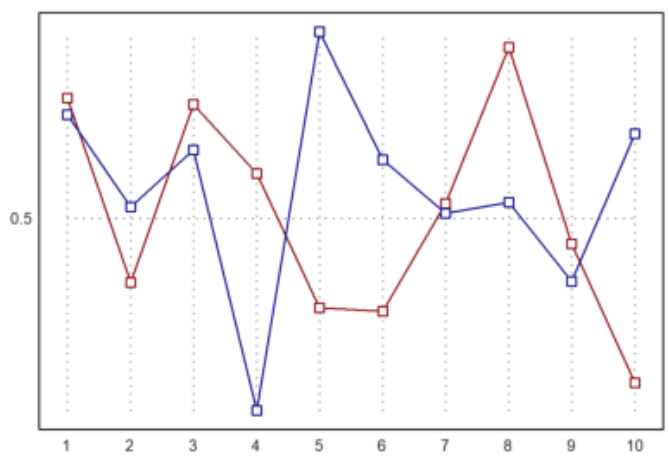
```
>n=10; plot2d(0:n,bin(n,0:n),>addpoints); ...
>labelbox("Binomials",styles="[]",>points,x=0.1,y=0.1, ...
>tcolor=black,>left):
```



Gaya plot seperti ini juga tersedia dalam statplot(). Sama seperti pada plot2d(), warna dapat diatur untuk setiap baris plot.

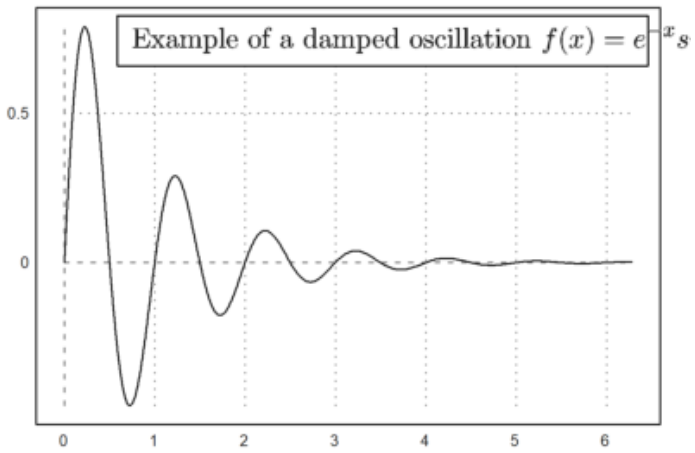
Selain itu, terdapat lebih banyak jenis plot khusus untuk keperluan statistik (lihat tutorial tentang statistik).

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



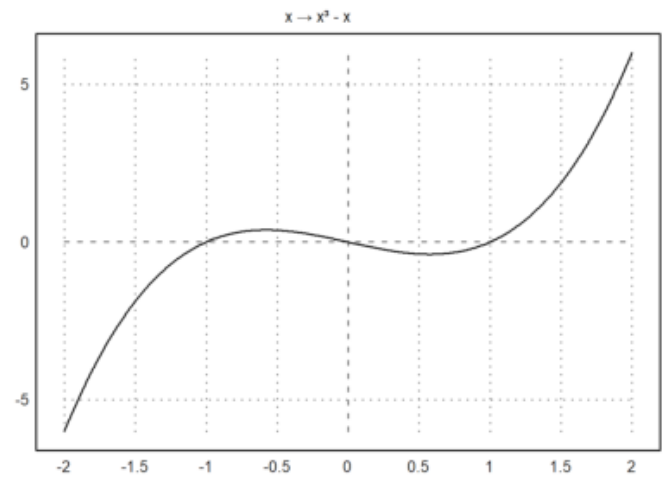
Fitur serupa adalah fungsi textbox().
Secara default, lebar kotak teks ditentukan oleh lebar maksimum baris teks.
Namun, lebar ini juga dapat diatur secara manual oleh pengguna.

```
>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):
```



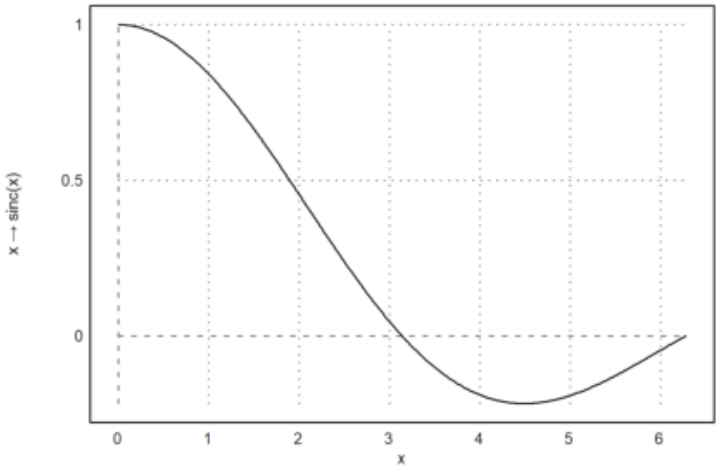
Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode
lihat sintaks EMT untuk penjelasan lebih lanjut tentang string Unicode).

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



Label pada sumbu-x dan sumbu-y dapat ditampilkan secara vertikal, begitu juga dengan sumbunya.

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



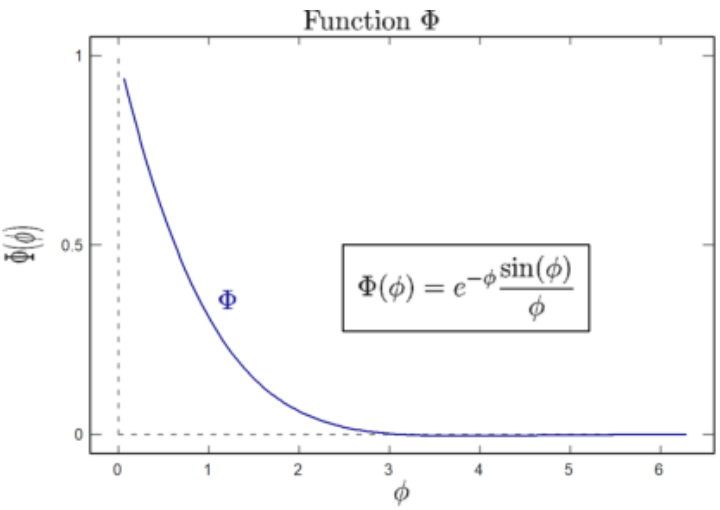
LaTeX

Anda juga dapat menampilkan rumus LaTeX jika sudah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Jalur ke file biner "latex" dan "dvipng" harus ada di system path, atau Anda perlu mengatur LaTeX melalui menu options.

Perlu diperhatikan bahwa proses parsing LaTeX cukup lambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, sebaiknya panggil fungsi latex() sekali sebelum loop dan gunakan hasilnya (berupa gambar dalam bentuk matriks RGB).

Pada contoh plot berikut, kita menggunakan LaTeX untuk label sumbu-x dan sumbu-y, sebuah label, sebuah kotak label (label box), dan juga 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):
```

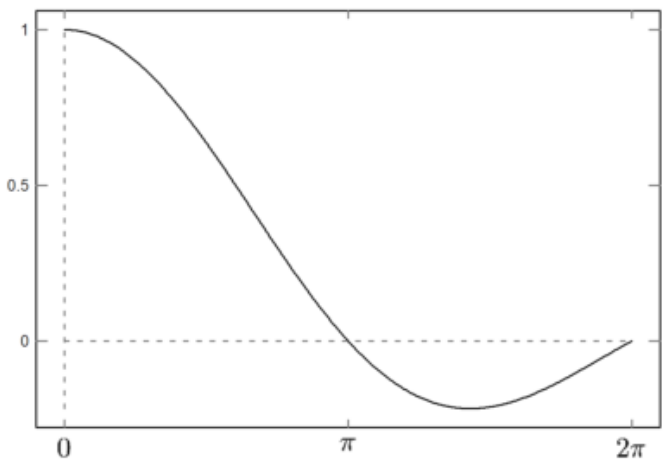


Sering kali, kita menginginkan jarak yang tidak konformal (non-conformal spacing) dan label teks pada sumbu-x. Untuk itu, kita dapat menggunakan xaxis() dan yaxis() sebagaimana akan ditunjukkan nanti.

Cara paling mudah adalah membuat plot kosong dengan bingkai menggunakan grid=4, kemudian menambahkan garis kisi dengan ygrid() dan xgrid().

Pada contoh berikut, kita menggunakan tiga string LaTeX untuk label pada sumbu-x dengan xtick().

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



Tentu saja, fungsi juga dapat digunakan.

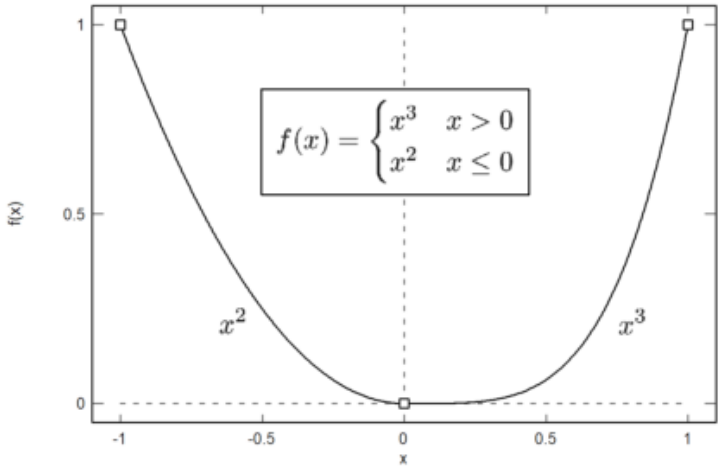
```
>function map f(x) ...
```

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

Parameter "map" membantu penggunaan fungsi untuk vektor. Untuk kebutuhan plot, sebenarnya hal ini tidaklah diperlukan. 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. Kode tersebut digunakan untuk dua label dan sebuah kotak teks. Tentu saja, Anda hanya dapat menggunakan LaTeX jika LaTeX 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):
```



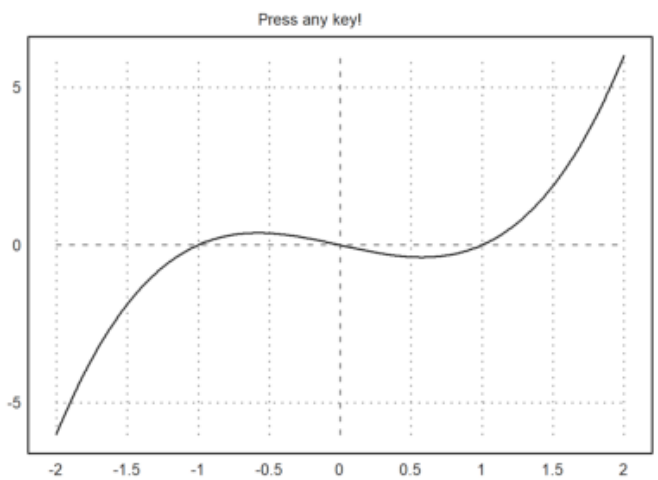
Interaksi Pengguna

Saat memplot sebuah fungsi atau ekspresi, parameter >user memungkinkan pengguna untuk melakukan zoom dan menggeser plot menggunakan tombol kursor atau mouse.

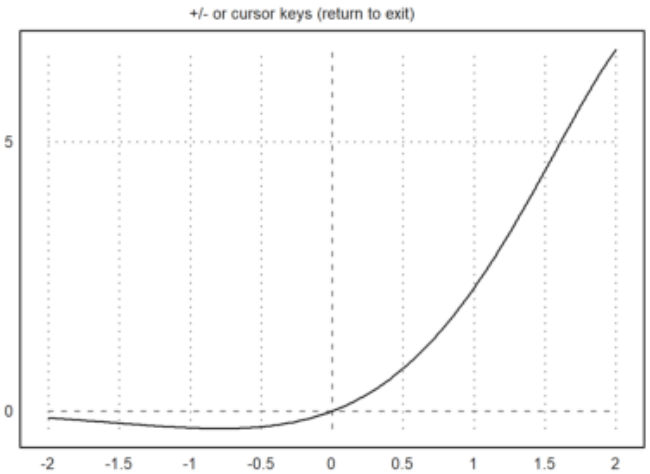
Pengguna dapat:
- melakukan zoom dengan tombol + atau -
- menggeser plot dengan tombol kursor
- memilih jendela plot dengan mouse
- mereset tampilan dengan tombol spasi
- keluar dengan tombol return (Enter)

Tombol spasi akan mengembalikan plot ke jendela plot aslinya.
aat 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)"):
```



Contoh berikut menunjukkan cara lanjutan untuk melakukan interaksi pengguna (lihat tutorial tentang pemrograman untuk detail lebih lanjut). Fungsi bawaan mousedrag() akan menunggu peristiwa dari mouse atau keyboard.

Fungsi ini dapat melaporkan:
klik/tahan mouse (mouse down)
pergerakan mouse (mouse moved)
pelepasan mouse (mouse up)
serta penekanan tombol (key presses).

Fungsi dragpoints() memanfaatkan hal ini, dan memungkinkan pengguna untuk menggeser (drag) titik apa pun di dalam plot.
Pertama-tama, kita membutuhkan fungsi plot. Sebagai contoh, kita akan melakukan interpolasi polinomial pada 5 titik. Fungsi ini harus memplot ke dalam area plot yang tetap (fixed plot area).

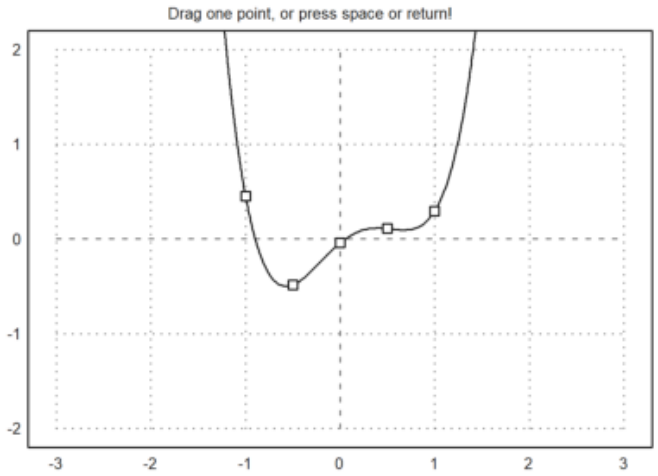
```
>function plotf(xp,yp,select) ...
```

```
    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 (semicolon parameters) pada plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp().
Tanpa ini, kita harus menulis fungsi plotinterp() terlebih dahulu, dengan mengakses nilai secara global.

Sekarang kita akan menghasilkan beberapa nilai acak, lalu membiarkan pengguna menggeser (drag) titik-titik tersebut.

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

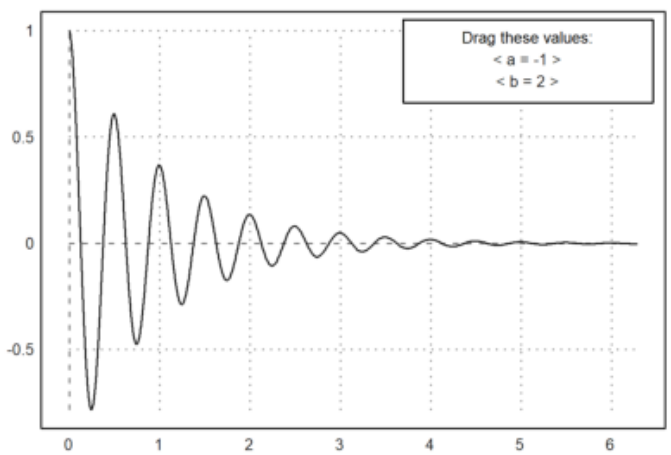


Ada juga sebuah fungsi yang dapat memplot fungsi lain yang bergantung pada vektor parameter, serta memungkinkan pengguna untuk menyesuaikan parameter-parameter tersebut.
Pertama, kita memerlukan fungsi plot terlebih dahulu.

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

Kemudian, kita memerlukan nama-nama parameter, nilai awal, dan sebuah matriks n×2 yang berisi rentang nilai (range) untuk setiap parameter, serta secara opsional bisa ditambahkan baris judul.
Terdapat slider interaktif yang memungkinkan pengguna mengatur nilai parameter secara langsung. Fungsi dragvalues() menyediakan fasilitas ini.

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



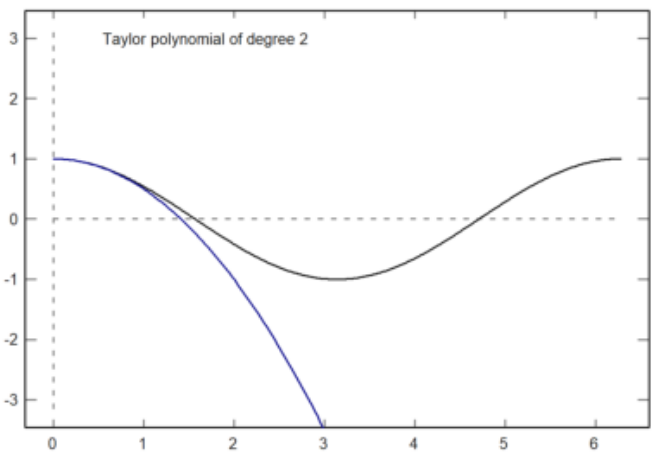
Dimungkinkan untuk membatasi nilai yang digeser (dragged values) hanya pada bilangan bulat.
Sebagai contoh, kita menulis sebuah fungsi plot yang memplot polinomial Taylor berderajat n terhadap fungsi kosinus.

```
>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 kita membiarkan derajat nbervariasi dari 0 hingga 20 dalam 20 langkah.
Hasil dari dragvalues() digunakan untuk memplot sketsa dengan nilai n tersebut, dan untuk menyisipkan plot ke dalam notebook.

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



Berikut ini adalah demonstrasi sederhana dari fungsi tersebut.
engguna dapat menggambar di atas jendela plot, sehingga meninggalkan jejak titik.

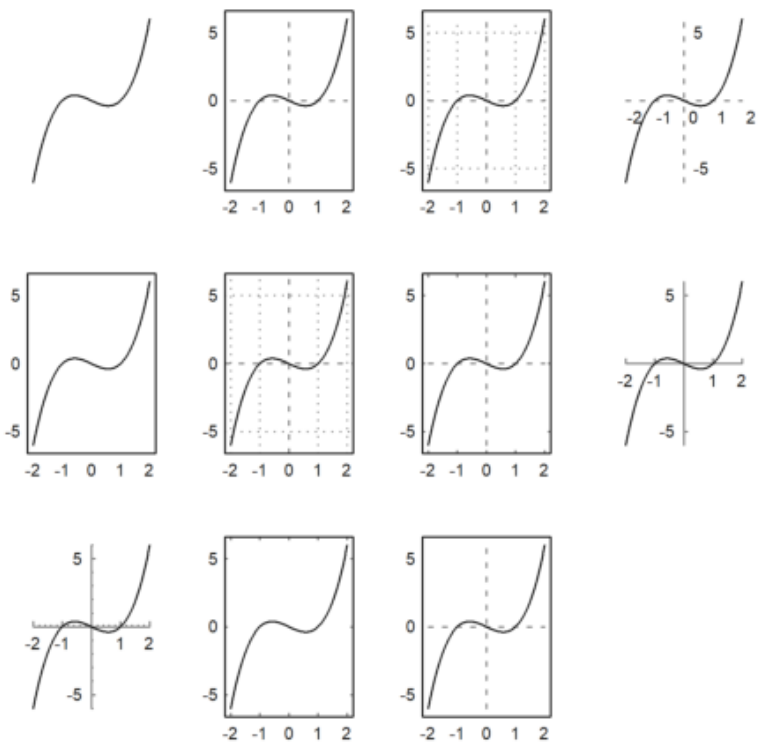
```
>function dragtest ...
```

```
    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!
```

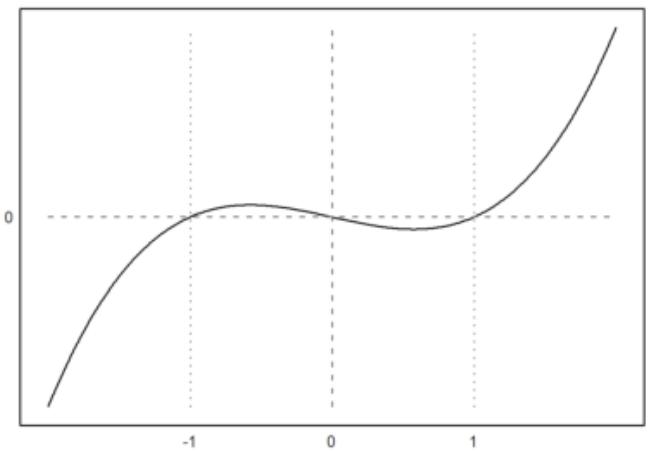
Secara default, EMT menghitung secara otomatis tanda-tanda sumbu (axis ticks) dan menambahkan label pada setiap tanda.
Hal ini dapat diubah menggunakan parameter grid.
Gaya default untuk sumbu dan label dapat dimodifikasi.
Selain itu, label dan judul juga dapat ditambahkan secara manual.
ntuk mengembalikan ke gaya default, gunakan perintah 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):
```



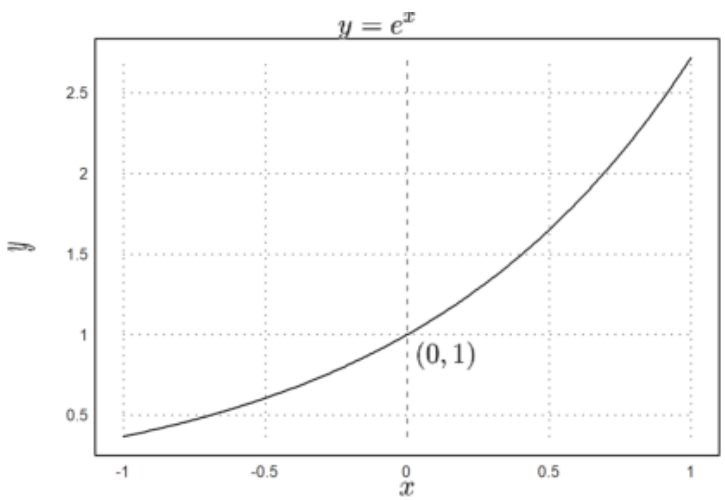
Parameter <frame digunakan untuk mematikan bingkai, dan framecolor=blue mengatur bingkai menjadi warna biru.
Jika Anda ingin menggunakan tanda sumbu (ticks) sendiri, Anda dapat menggunakan style=0, lalu menambahkan semua elemen lainnya secara manual.

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



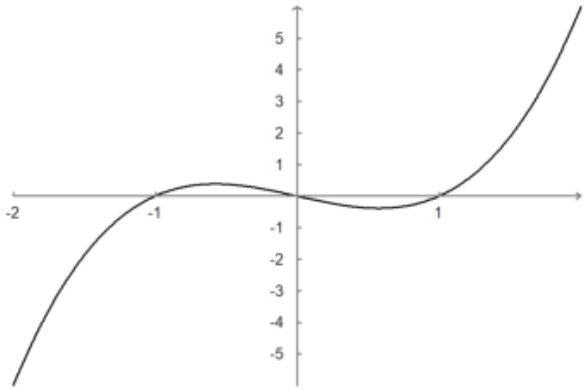
Untuk judul plot dan label sumbu, lihat contoh berikut.

```
>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 dapat 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="->"):
```

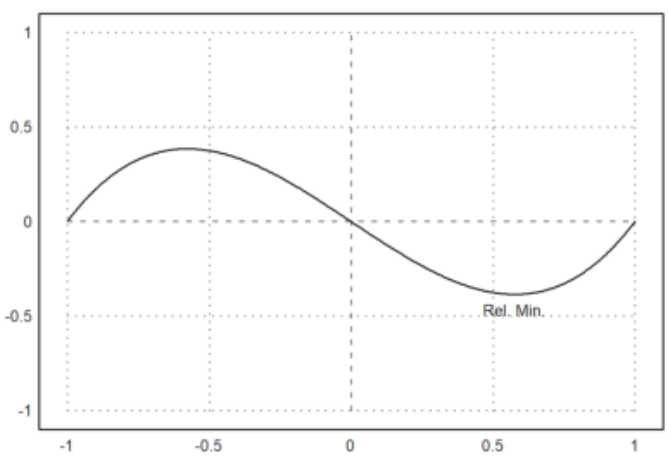


Teks pada plot dapat ditambahkan menggunakan label().
ada contoh berikut, "lc" berarti lower center.
psi ini menentukan posisi label relatif terhadap koordinat plot.

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

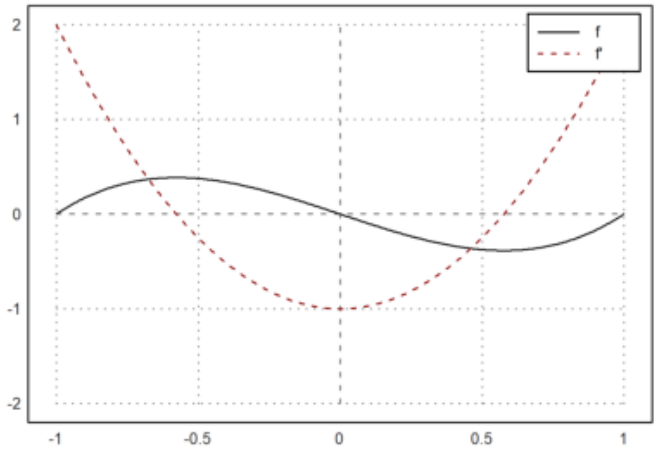
$$x^3 - x$$

```
>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
```

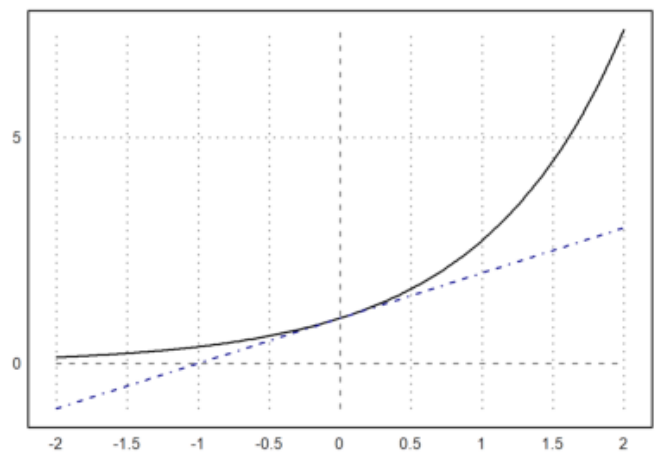


Selain itu, terdapat juga kotak teks (text boxes).

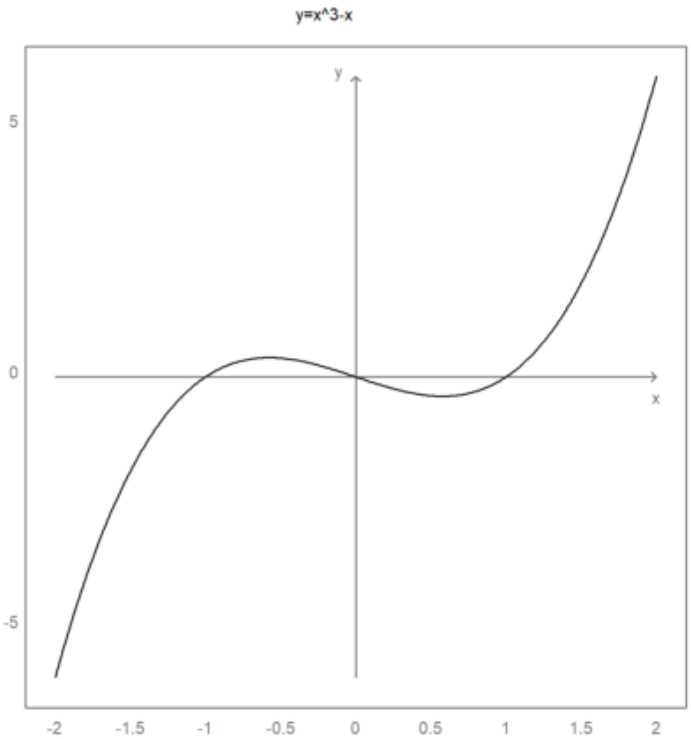
```
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // derivative
>labelbox(["f","f'"],["-","--"],[black,red]): // label box
```




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

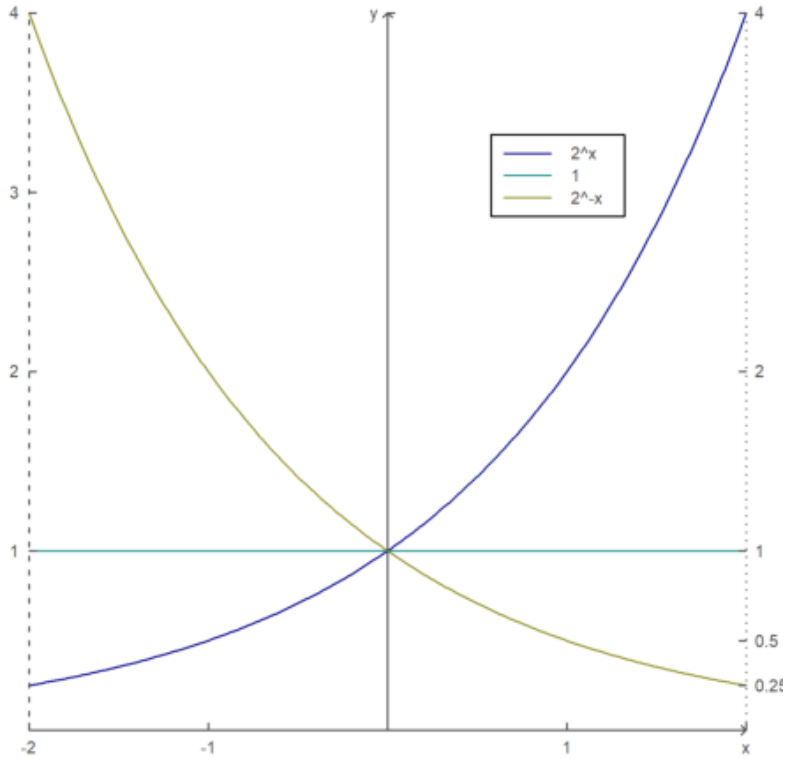



```
>gridstyle("->",color=gray,textcolor=gray,framecolor=gray);  ...
> plot2d("x^3-x",grid=1);   ...
> settitle("y=x^3-x",color=black); ...
> label("x",2,0,pos="bc",color=gray);  ...
> label("y",0,6,pos="cl",color=gray); ...
> reset():
```



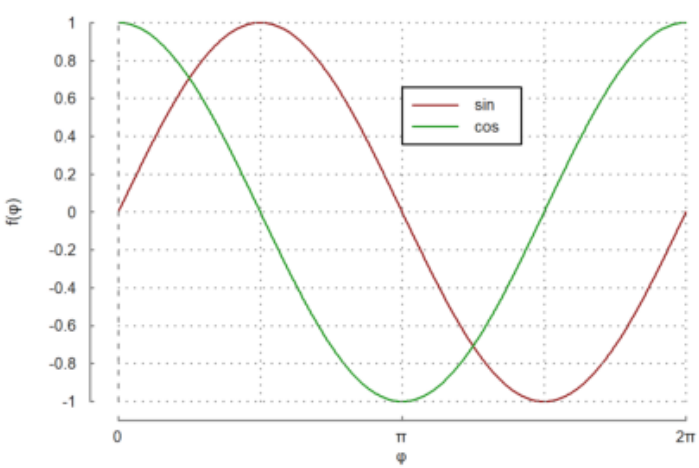
Untuk kontrol yang lebih tinggi, sumbu-x dan sumbu-y dapat digambar secara manual.
Perintah fullwindow() memperluas jendela plot, karena kita tidak lagi membutuhkan ruang untuk label di luar jendela plot.
Gunakan shrinkwindow() atau reset() untuk mengembalikan pengaturan ke default.

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

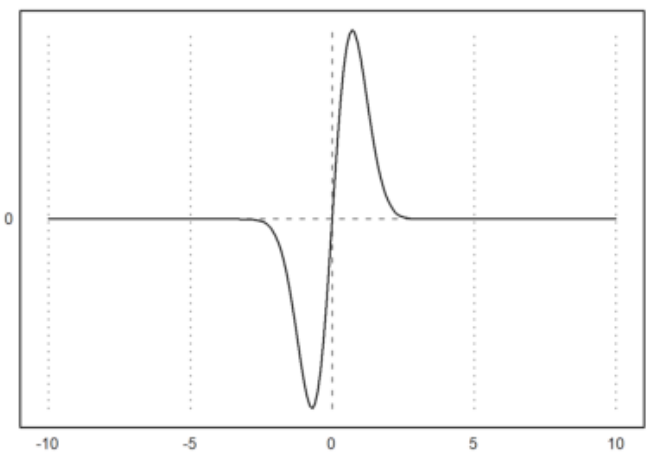
```
>aspect(1.5);
>plot2d(["sin(x)","cos(x)"],0,2pi,color=[red,green],<grid,<frame); ...
> xaxis(-1.1,(0:2)*pi,xt=["0",u"&pi;",u"2&pi;"],style="-",>ticks,>zero);  ...
> xgrid((0:0.5:2)*pi,<ticks); ...
> yaxis(-0.1*pi,-1:0.2:1,style="-",>zero,>grid); ...
> labelbox(["sin","cos"],colors=[red,green],x=0.5,y=0.2,>left);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



Plotting 2D Data

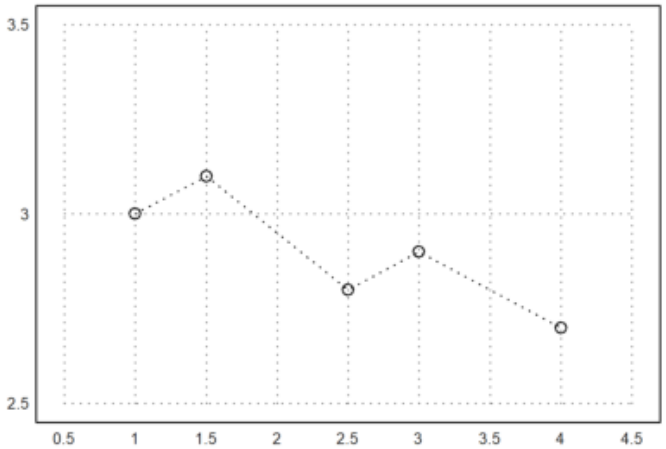
Jika x dan y adalah vektor data, data ini akan digunakan sebagai koordinat x dan y dari sebuah kurva.
Dalam hal ini, a, b, c, dan d, atau radius r dapat ditentukan, atau jendela plot akan menyesuaikan secara otomatis dengan data.
ebagai alternatif, opsi >square dapat digunakan untuk mempertahankan rasio aspek persegi.
Memplot sebuah ekspresi sebenarnya hanyalah singkatan untuk plot data.
ntuk plot data, Anda memerlukan satu atau lebih baris nilai x, dan satu atau lebih baris nilai y.
Berdasarkan rentang dan nilai x, fungsi plot2d akan menghitung data yang akan diplot, secara default dengan evaluasi adaptif dari fungsi tersebut.
- Untuk plot titik, gunakan >points
- Untuk campuran garis dan titik, gunakan >addpoints
Namun, Anda juga dapat memasukkan data secara langsung:
- Gunakan vektor baris untuk x dan y untuk satu fungsi
- Gunakan matriks untuk x dan y, yang akan 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):
```

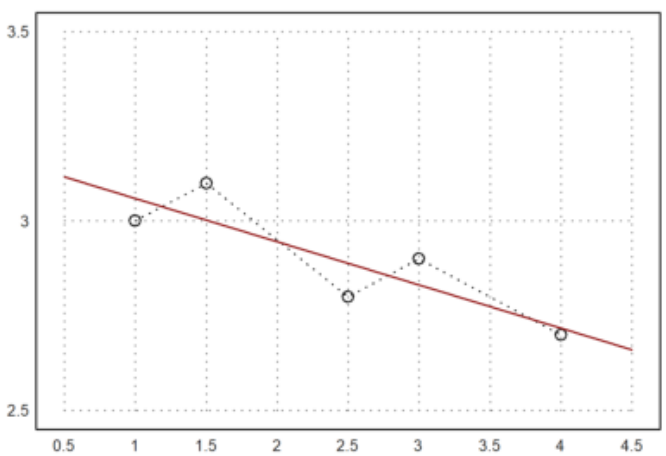


Data juga dapat diplot sebagai titik. Gunakan points=true untuk ini.
Plot ini bekerja seperti poligon, tetapi hanya menggambar titik-titik sudutnya.
- style="...": Pilih dari "[]", "<>", "o", ".", "..", "+", "*", "[]", "<>", "o", "..", "", "|".
Untuk memplot sekumpulan titik, gunakan >points.
Jika color berupa vektor warna, setiap titik akan memiliki warna yang berbeda.
Untuk matriks koordinat dan vektor kolom, warna akan diterapkan pada baris-baris matriks.
Parameter >addpoints menambahkan titik ke segmen garis pada 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
```



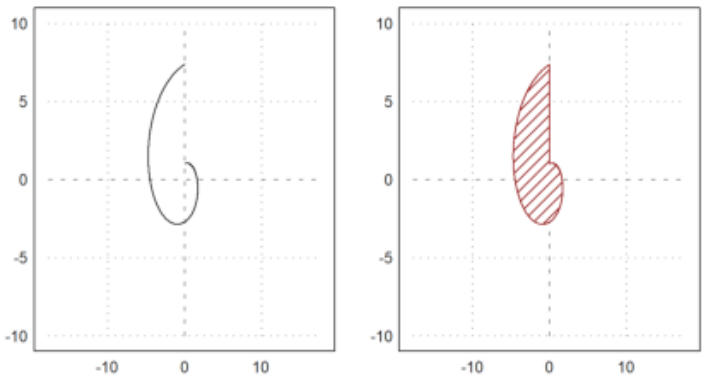
Menggambar Daerah Yang Dibatasi Kurva

Plot data sebenarnya adalah poligon. Kita juga dapat memplot kurva atau kurva yang diisi (filled curves).

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

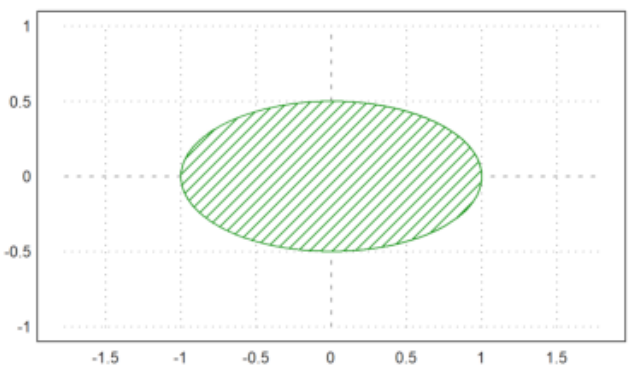
Warna isi ditentukan oleh argumen fillcolor, dan opsi opsional <outline mencegah menggambar garis batas untuk semua gaya kecuali gaya default.

```
>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):
```

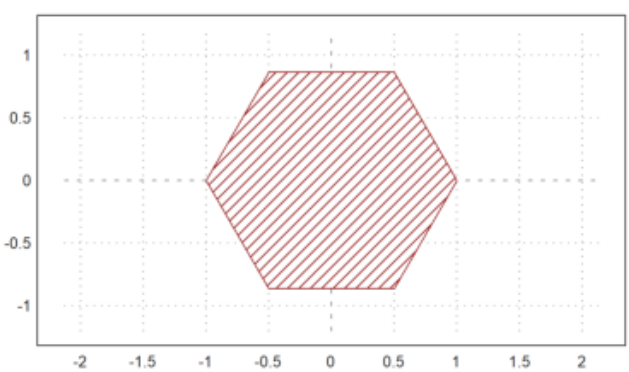


Pada contoh berikut, kita memplot elips yang diisi dan dua heksagon yang diisi menggunakan kurva tertutup dengan 6 titik, masing-masing dengan gaya isi (fill style) yang berbeda.

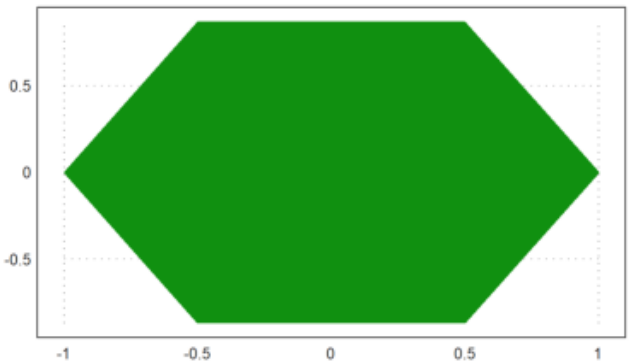
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```




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

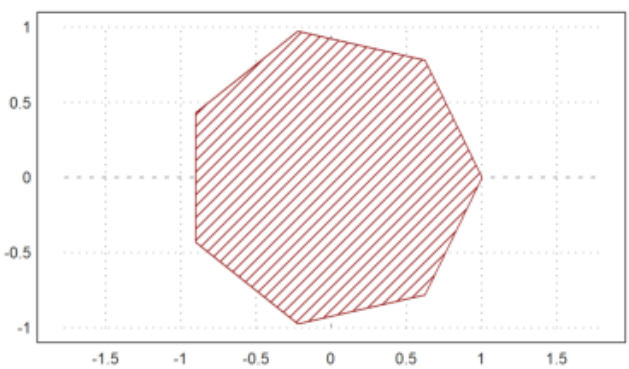


```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



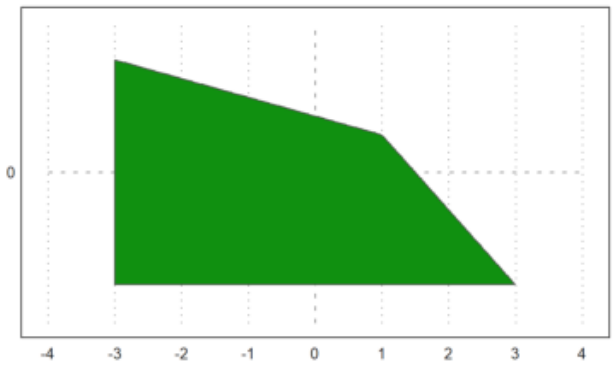
Contoh lain adalah septagon, yang kita buat menggunakan 7 titik pada lingkaran satuan (unit circle).

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



Berikut ini adalah himpunan nilai maksimum dari empat kondisi linear yang kurang dari atau sama dengan 3.
Ini adalah A[k]·v <= 3 untuk semua baris pada A.
Untuk mendapatkan titik 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):
```

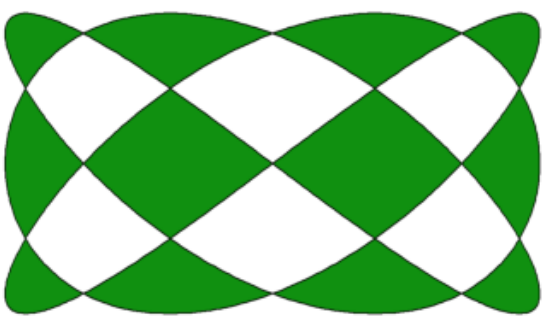


Poin utama dari bahasa matriks adalah bahwa bahasa ini memungkinkan kita untuk membuat tabel fungsi dengan mudah.

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

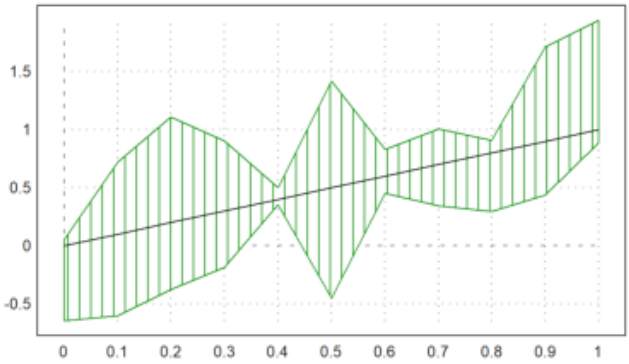
Sekarang kita memiliki vektor x dan y dari nilai-nilai tertentu.
Fungsi plot2d() dapat memplot nilai-nilai ini sebagai kurva yang menghubungkan titik-titik.
Plot ini juga dapat diisi (filled).
Dalam hal ini, hasilnya akan rapi karena menggunakan winding rule, yang diterapkan untuk pengisian (fill).

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



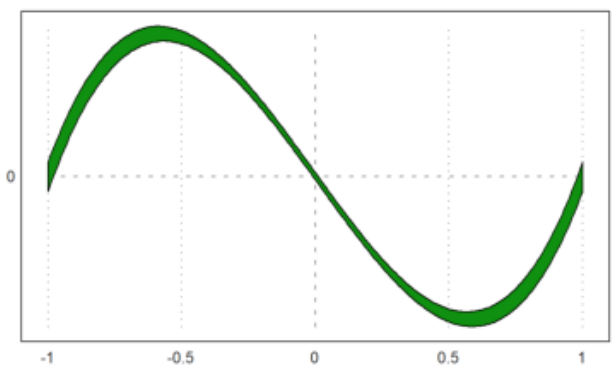
Vektor interval dapat diplot terhadap nilai x sebagai daerah terisi (filled region) di antara nilai bawah (lower) dan atas (upper) dari interval tersebut.
Ini berguna, misalnya, untuk memvisualisasikan kesalahan perhitungan (computation errors).
Selain itu, dapat juga digunakan untuk menampilkan kesalahan statistik (statistical errors).

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



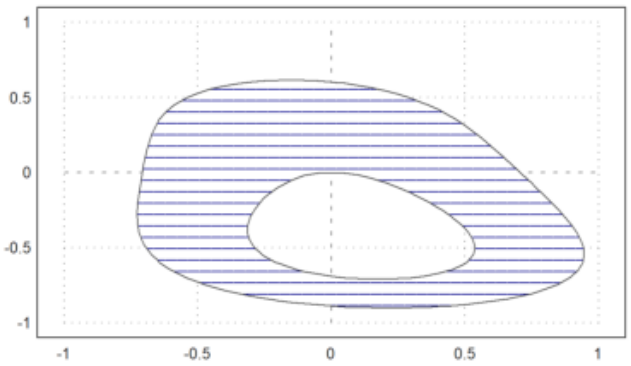
Jika x adalah vektor yang sudah diurutkan, dan y adalah vektor interval, maka plot2d akan memplot range terisi (filled ranges) dari interval tersebut di bidang koordinat.
Gaya pengisian (fill styles) sama dengan gaya poligon.

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



Dimungkinkan untuk mengisi (fill) daerah nilai untuk fungsi tertentu.
Untuk ini, level harus berupa matriks 2×n.
- Baris pertama berisi batas bawah (lower bounds)
- Baris kedua berisi batas atas (upper bounds)

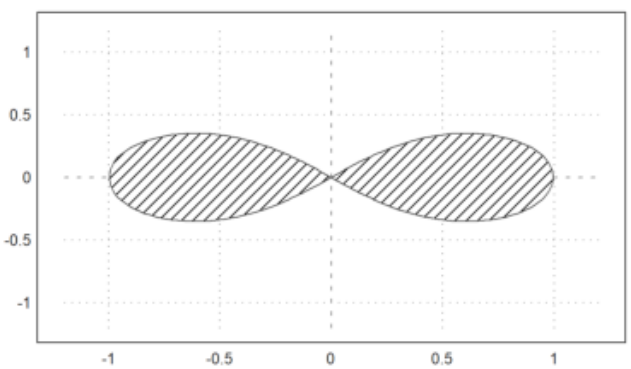
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



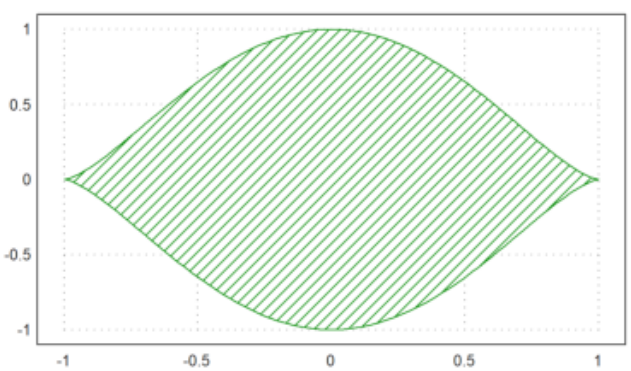
Kita juga dapat mengisi (fill) rentang nilai seperti berikut:

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

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

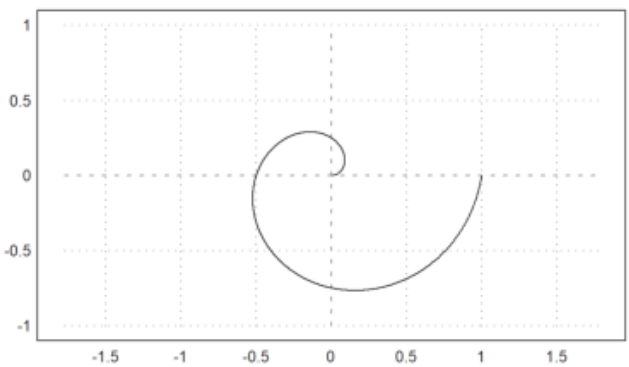


Nilai x tidak perlu diurutkan. Pasangan(x,y) hanya menggambarkan sebuah kurva.
Jika x diurutkan, kurva tersebut merupakan grafik dari sebuah fungsi.
Pada contoh berikut, kita memplot spiral

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

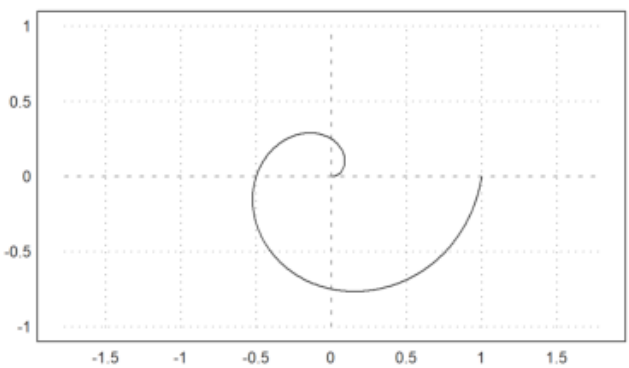
Kita perlu menggunakan banyak titik agar tampilan kurva halus, atau menggunakan fungsi adaptive() untuk mengevaluasi ekspresi (lihat fungsi adaptive() untuk penjelasan lebih detail).

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

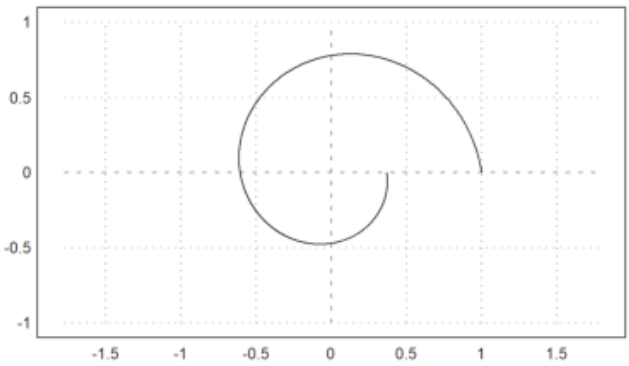


Sebagai alternatif, dimungkinkan untuk menggunakan dua ekspresi untuk kurva.
Berikut ini memplot kurva yang sama seperti pada contoh sebelumnya.

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




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



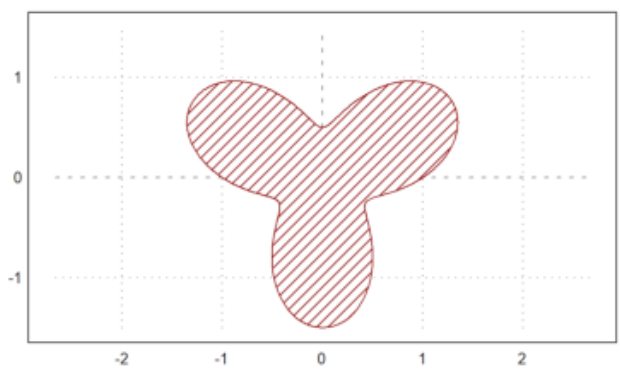
Pada contoh berikut, kita memplot kurva

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

dengan

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

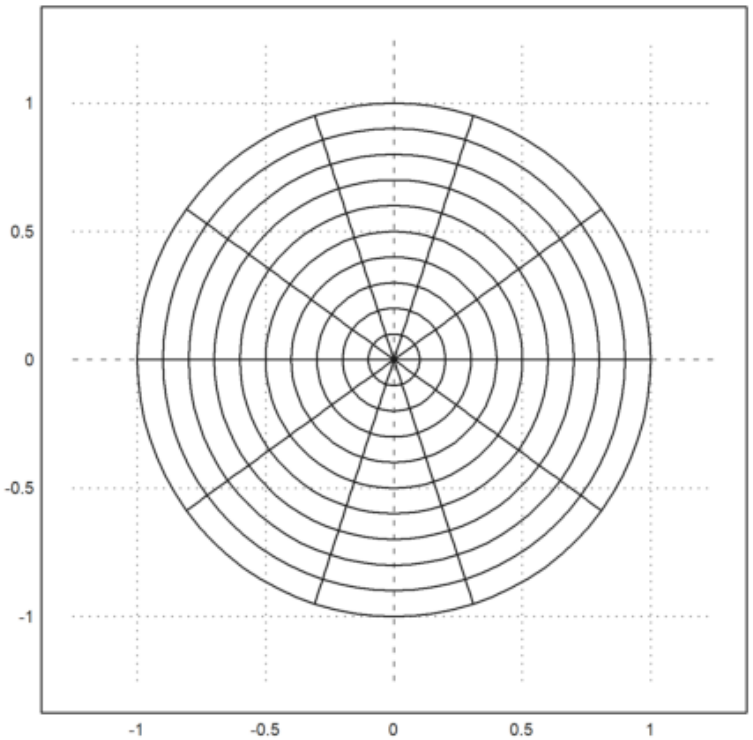
```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
>plot2d(x,y,>filled,fillcolor=red,style="/",r=1.5):
```



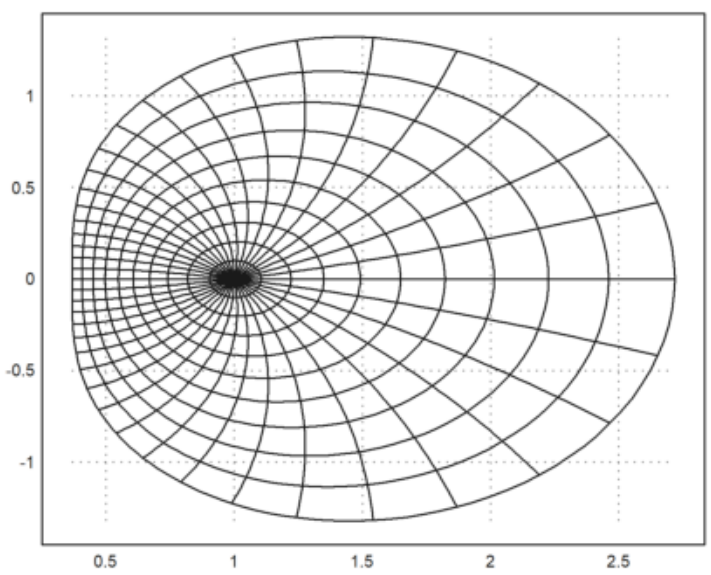
Menggambar Grafik Bilangan Kompleks

An array of complex numbers can also be plotted. Then the grid points will be connected. If a number of grid lines is specified (or a 1x2 vector of grid lines) in the argument cgrid only those grid lines are visible.
A matrix of complex numbers will automatically plot as a grid in the complex plane.
In the following example, we plot the image of the unit circle under the exponential function. The cgrid parameter hides some of the grid curves.

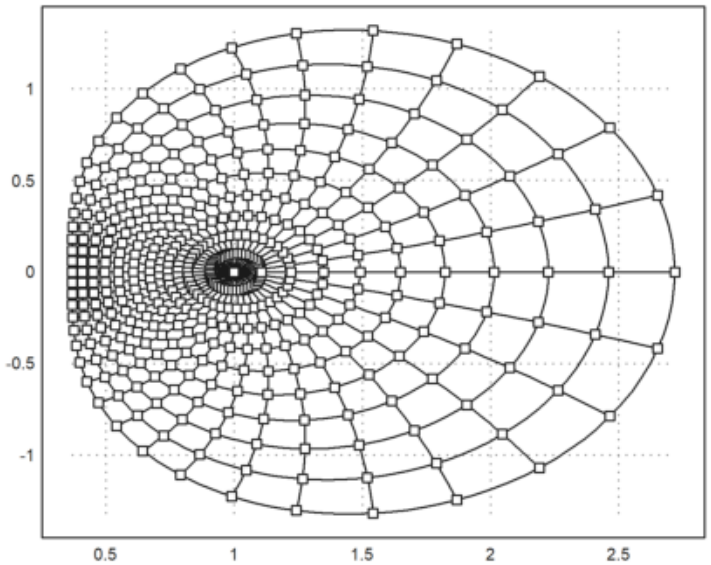
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```




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



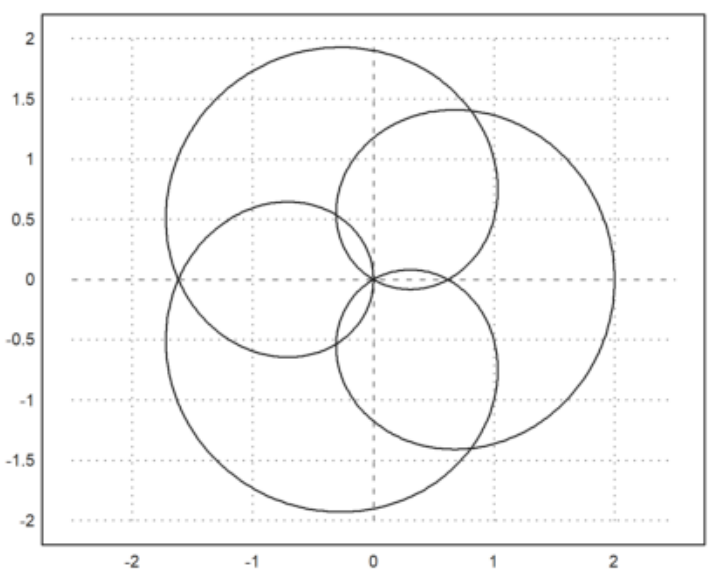

```
>r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=r*exp(I*a);
>plot2d(exp(z),>points,>add):
```



Vektor bilangan kompleks secara otomatis diplot sebagai kurva di bidang kompleks, menggunakan bagian real dan bagian imajiner.
Pada contoh ini, kita memplot lingkaran satuan (unit circle) dengan

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

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



Plot Statistik

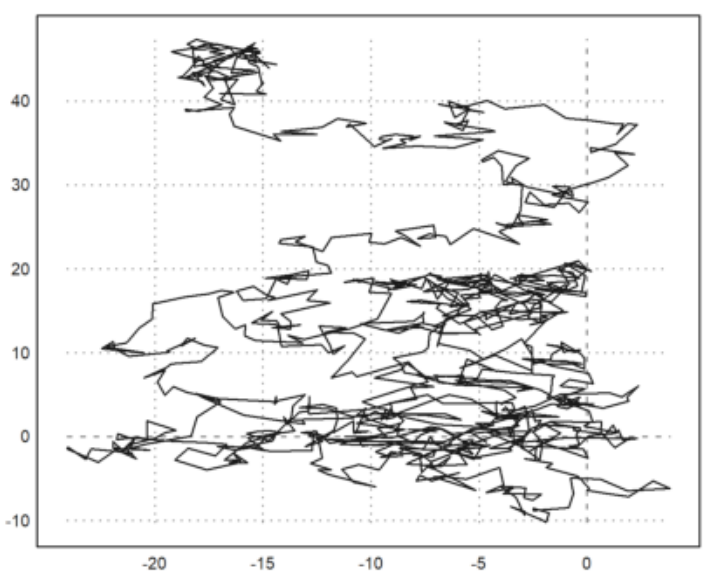
Terdapat banyak fungsi yang khusus untuk plot statistik.
Salah satu plot yang sering digunakan adalah plot kolom (column plot).
Jumlah kumulatif (cumulative sum) dari nilai yang terdistribusi normal 0-1 menghasilkan sebuah random walk.

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

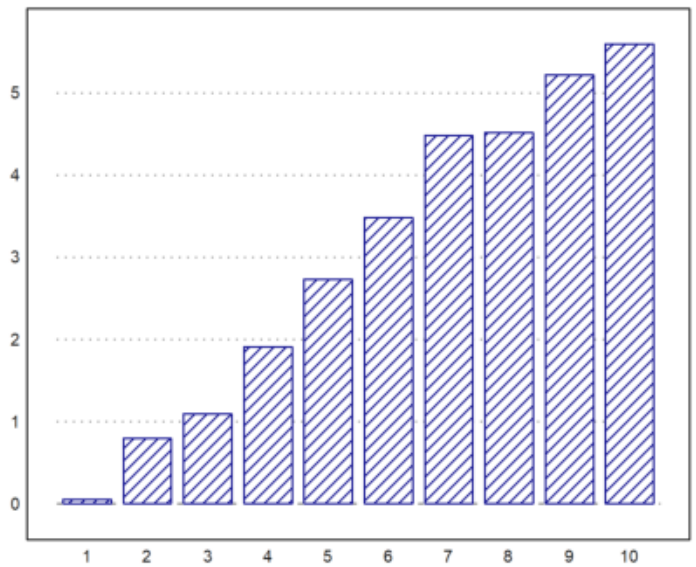


Menggunakan dua baris menampilkan random walk dalam dua dimensi.

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

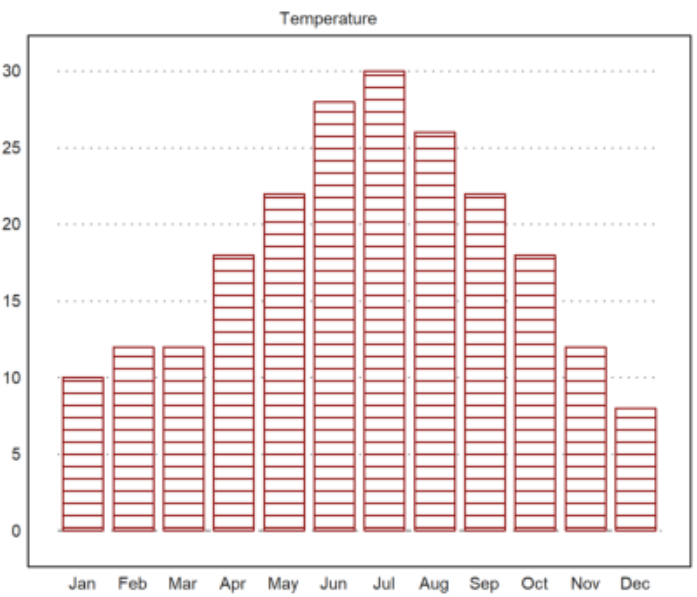


```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

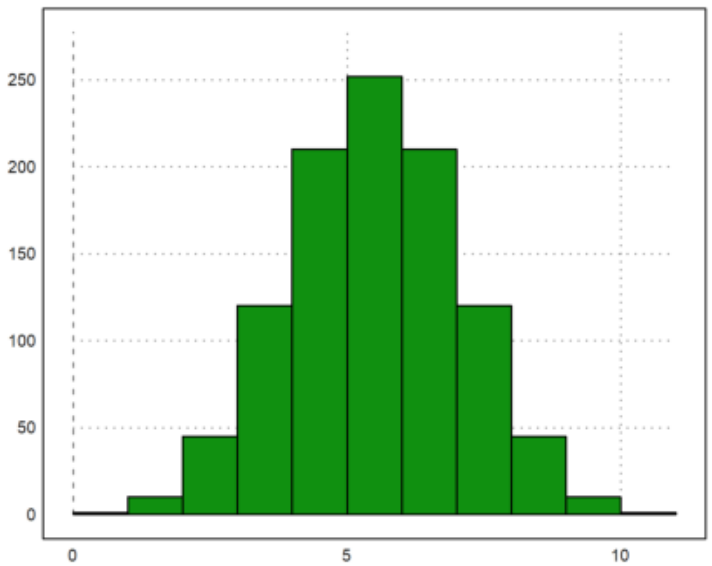


Plot ini juga dapat menampilkan string sebagai label.

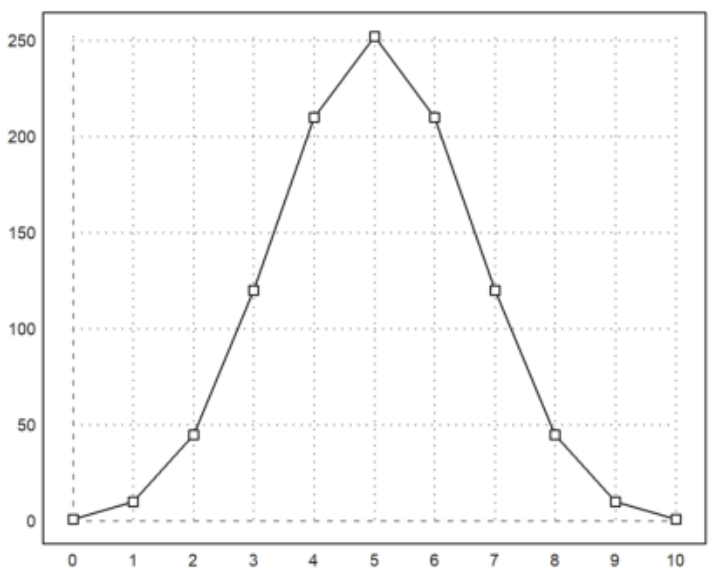
```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
>  "Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnsplot(values,lab=months,color=red,style="-");
>title("Temperature"):
```



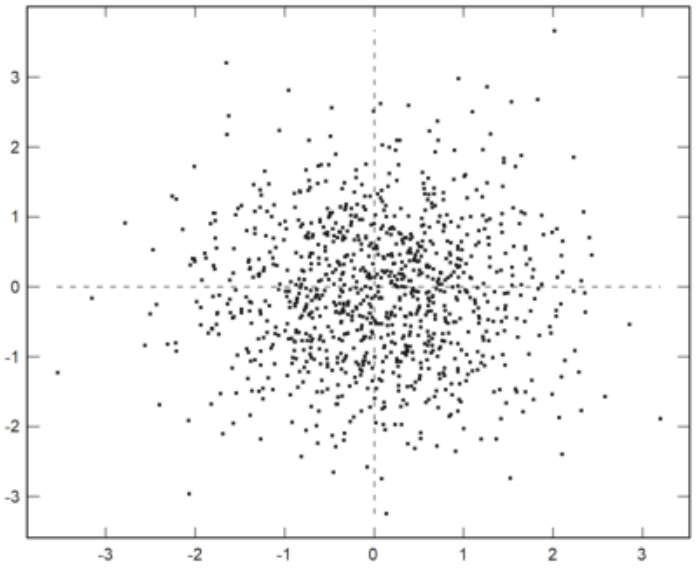
```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```



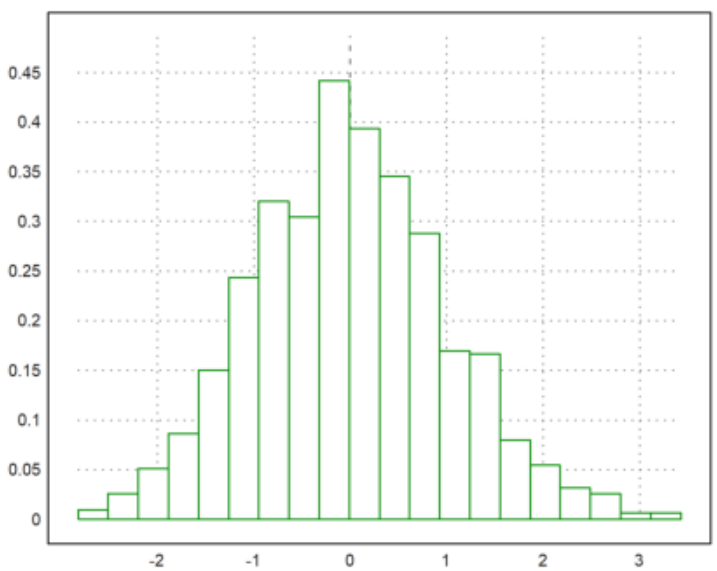

```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



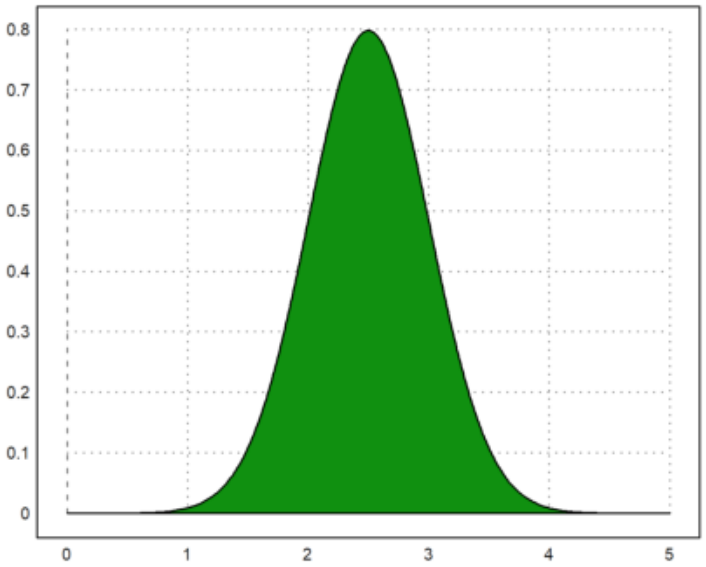

```
>plot2d(normal(1000),normal(1000),>points,grid=6,style="..");
```




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

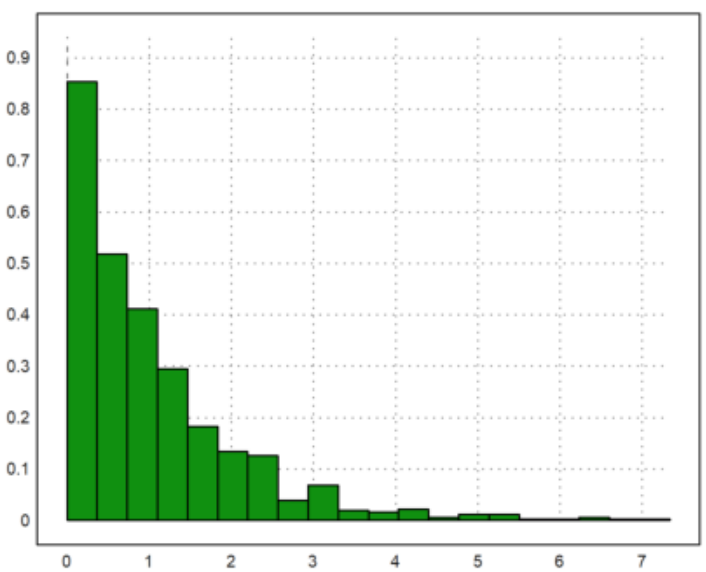



```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



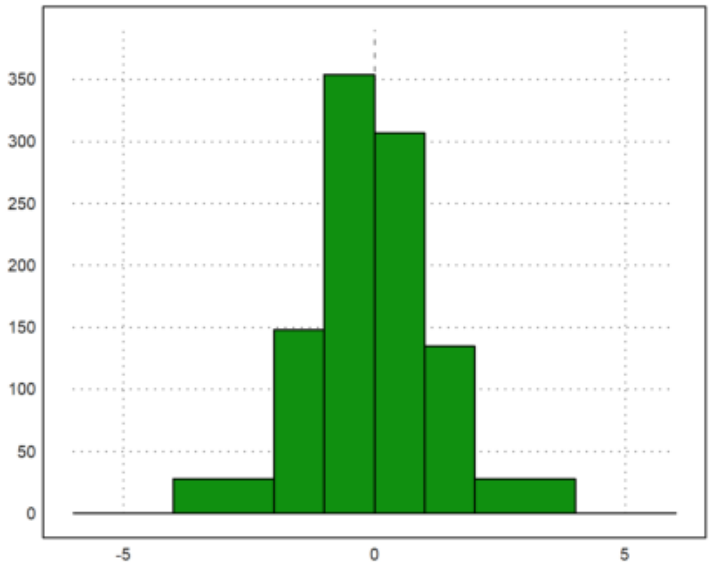
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan distribution=n dengan plot2d.

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



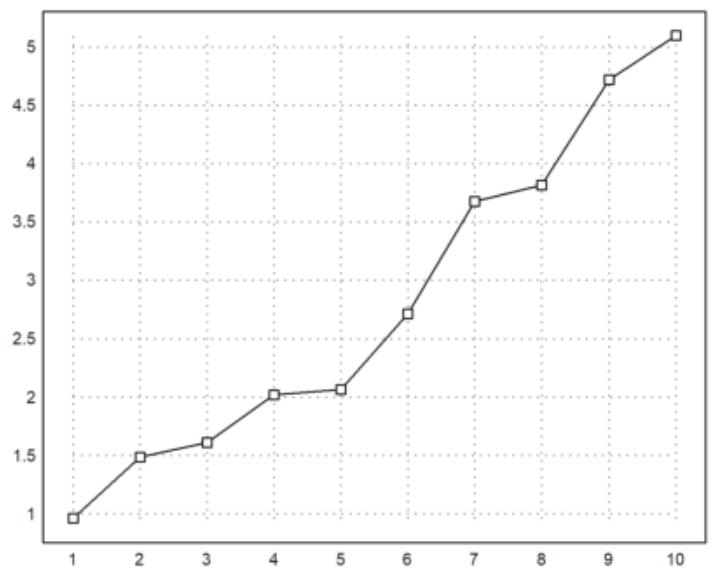
Atau, Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan >bar di plot3d, atau menggunakan plot kolom (column plot).

```
>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):
```

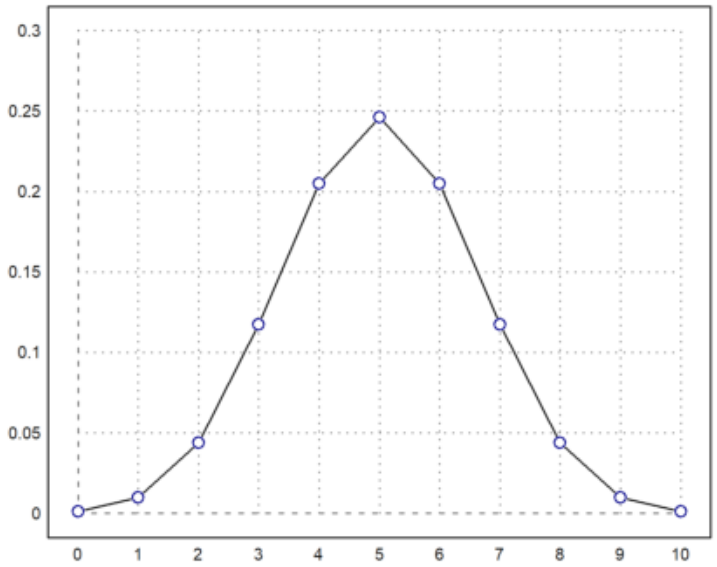


Fungsi statplot() mengatur gaya plot menggunakan string sederhana.

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

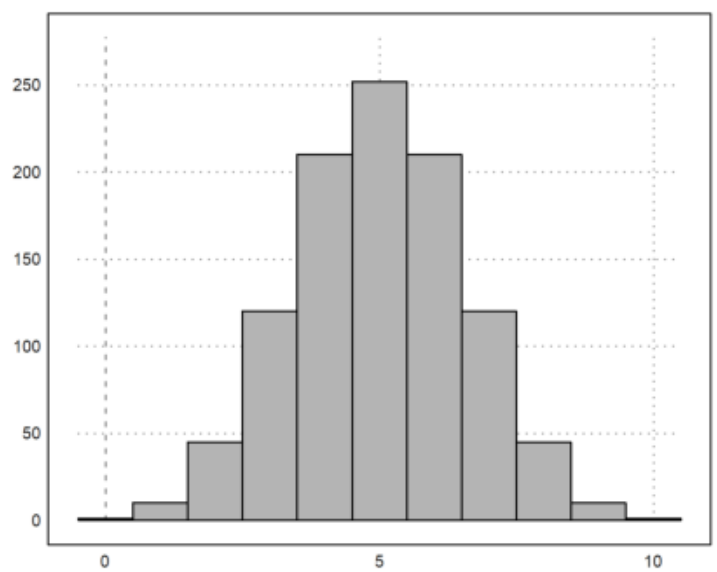



```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



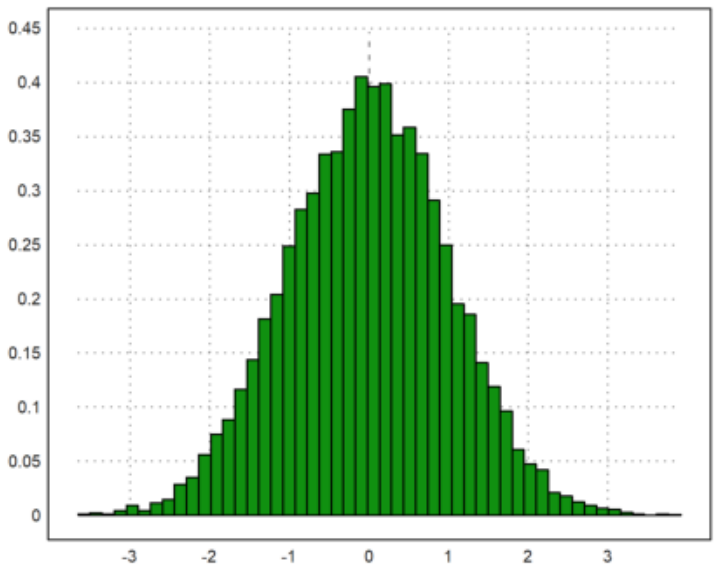
Selain itu, data juga dapat diplot sebagai batang (bars).
Dalam hal ini, x harus diurutkan dan memiliki satu elemen lebih banyak daripada y.
Batang akan membentang dari x[i] ke x[i+1] dengan nilai y[i].
Jika x memiliki ukuran yang sama dengan y, maka x akan diperluas satu elemen dengan jarak terakhir.
Gaya pengisian (fill styles) dapat digunakan seperti yang dijelaskan sebelumnya.

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

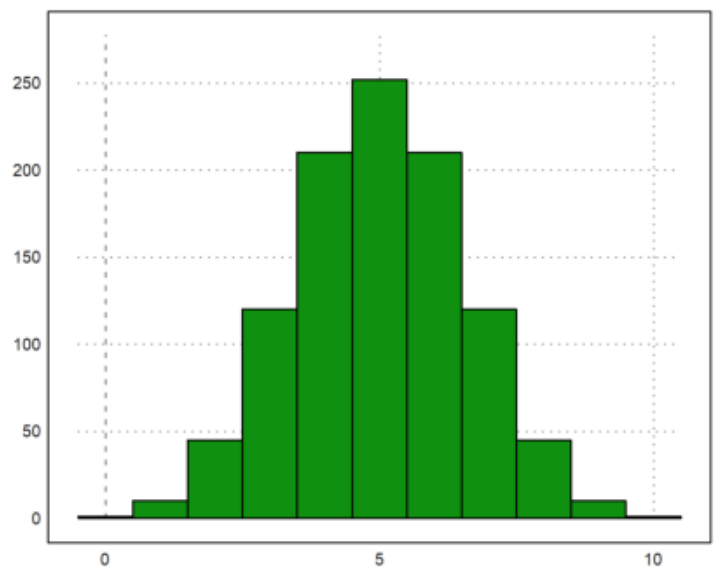


Data untuk plot batang (bar=1) dan histogram (histogram=1) dapat:
- diberikan secara eksplisit dalam xv dan yv, atau
- dihitung dari distribusi empiris dalam xv dengan >distribution (atau distribution=n).
Histogram dari nilai xv akan dihitung secara otomatis dengan >histogram.
Jika >even ditentukan,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

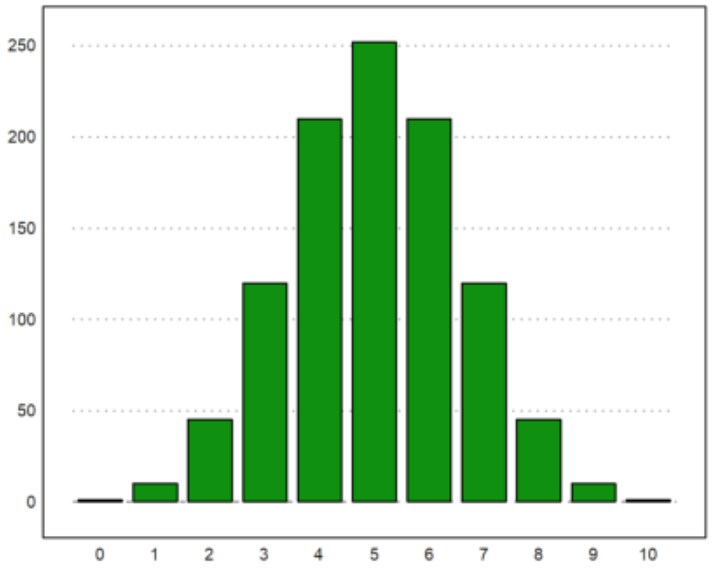
```
>plot2d(normal(10000),distribution=50):
```



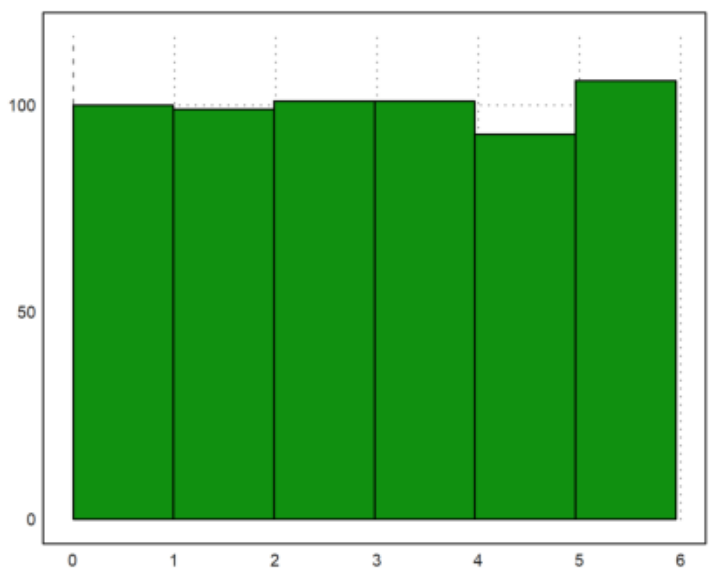

```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```




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

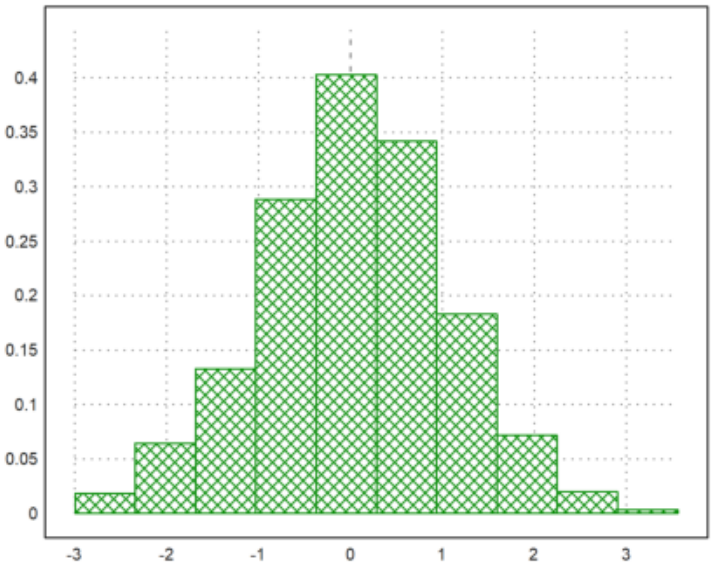



```
>plot2d(random(600)*6,histogram=6):
```



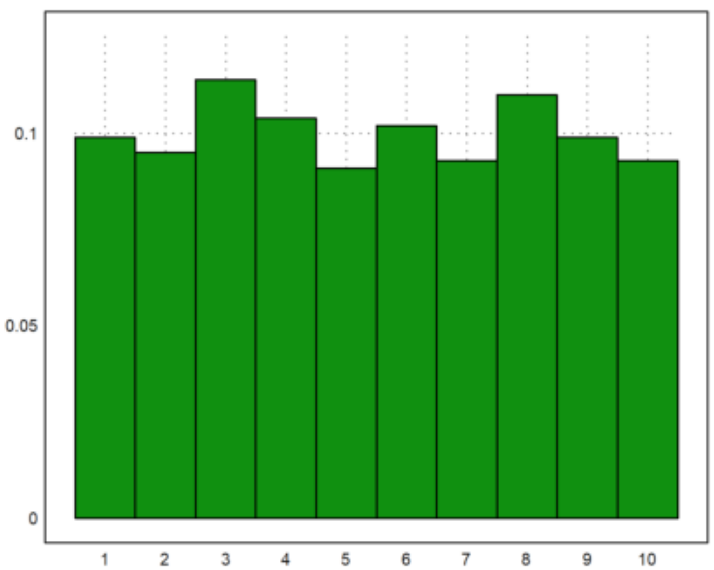
Untuk distribusi, terdapat parameter distribution=n, yang secara otomatis menghitung nilai dan menampilkan distribusi relatif dengan n sub-interval.

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



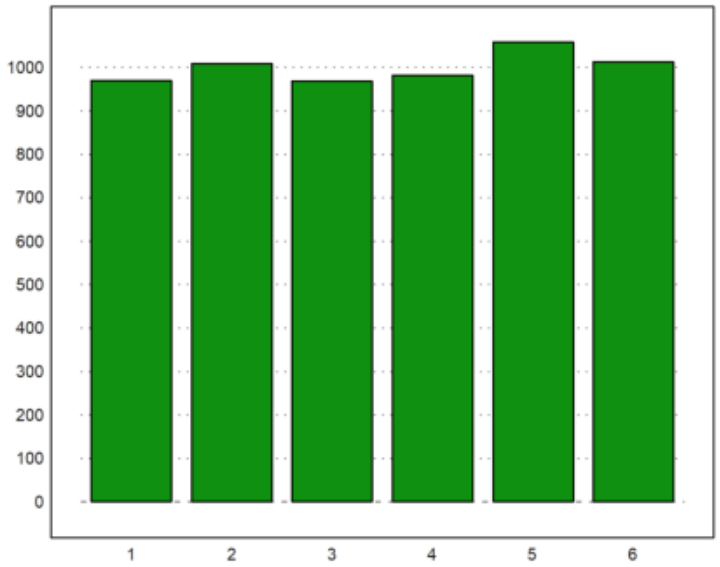
Dengan parameter even=true, interval yang digunakan akan berupa bilangan bulat (integer intervals).

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

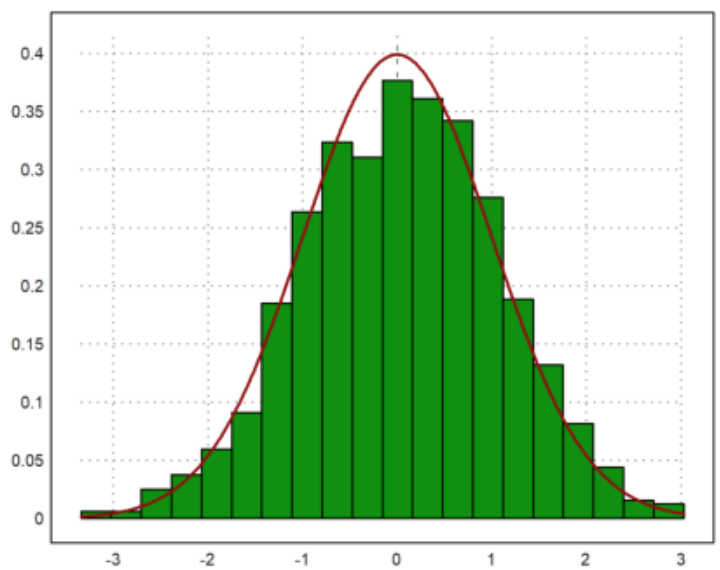


Perlu dicatat bahwa terdapat banyak plot statistik yang mungkin berguna.
Silakan lihat tutorial tentang statistik untuk informasi lebih lanjut.

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

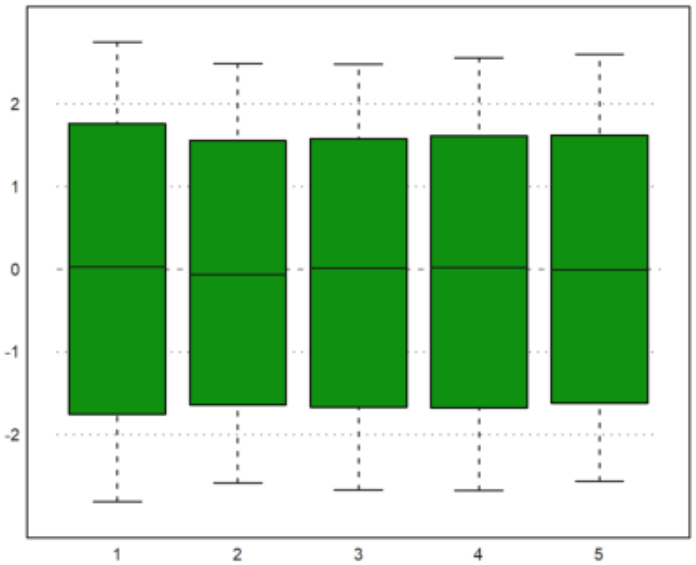



```
>plot2d(normal(1,1000),>distribution); ...
>  plot2d("qnormal(x)",color=red,thickness=2,>add):
```



Terdapat juga banyak plot khusus untuk statistik.
Boxplot menampilkan kuartil-kuartil dari distribusi tersebut dan banyak outlier.
Berdasarkan definisi, outlier dalam boxplot adalah data yang melebihi 1,5 kali rentang 50% tengah (middle 50% range) dari plot.

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



Fungsi Implisit

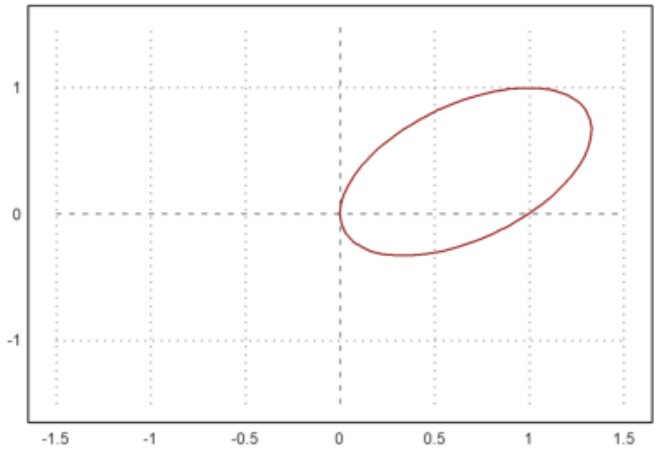
Plot implisit menampilkan garis level (level lines) yang menyelesaikan
(x,y)=level, di mana level dapat berupa satu nilai atau vektor nilai.
Jika level="auto", akan dibuat nc garis level, yang tersebar merata antara nilai minimum dan maksimum dari fungsi.
Warna lebih gelap atau lebih terang dapat ditambahkan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

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

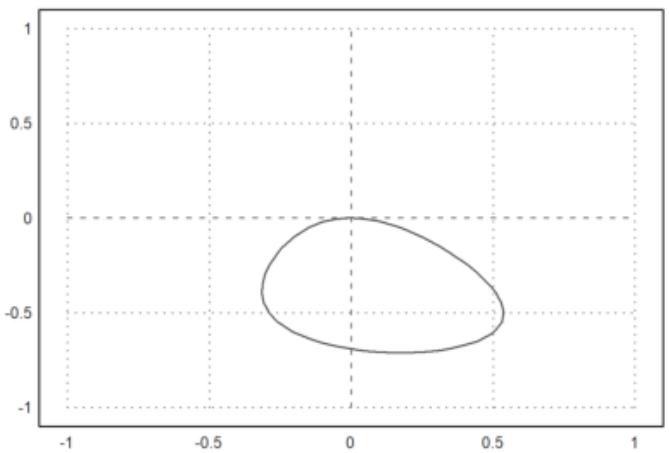
dari fungsi apapun.

Untuk menggambar himpunan f(x,y)=c untuk satu atau lebih konstanta c, Anda dapat menggunakan plot2d() dengan plot implisit di bidang koordinat. Parameter untuk c adalah level=c, di mana c bisa berupa vektor garis level.
Selain itu, skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi pada setiap titik di plot.
arameter n menentukan ketelitian (fineness) dari plot.

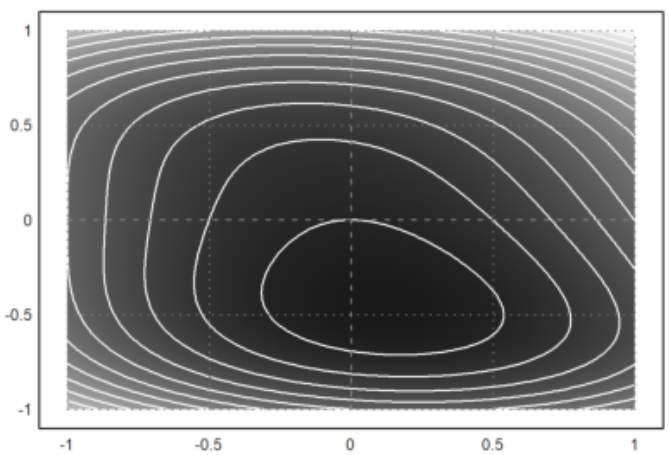
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```



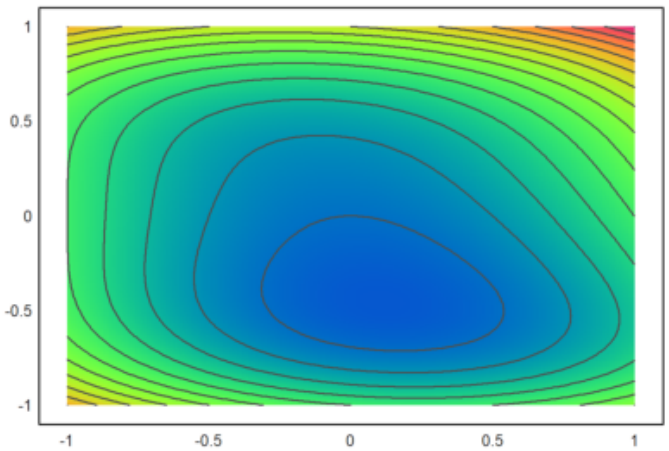

```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0): // Solutions of f(x,y)=0
```




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

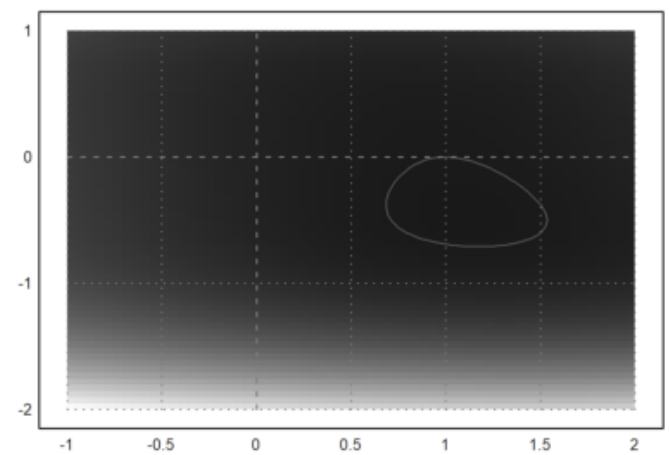


```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
```

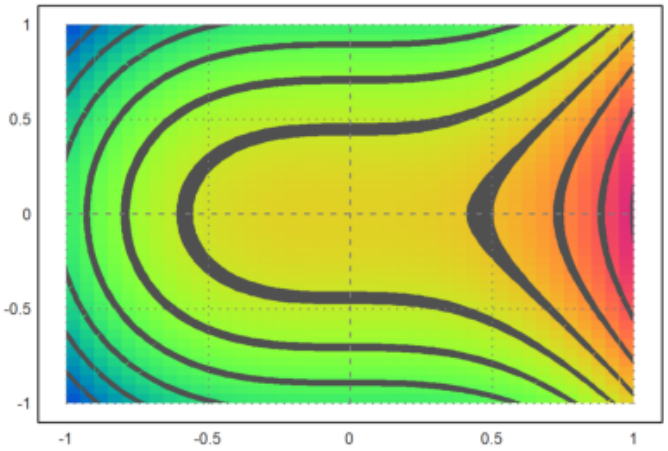


Hal ini juga berlaku untuk plot data.
Namun, Anda harus menentukan rentang (ranges) untuk label sumbu.

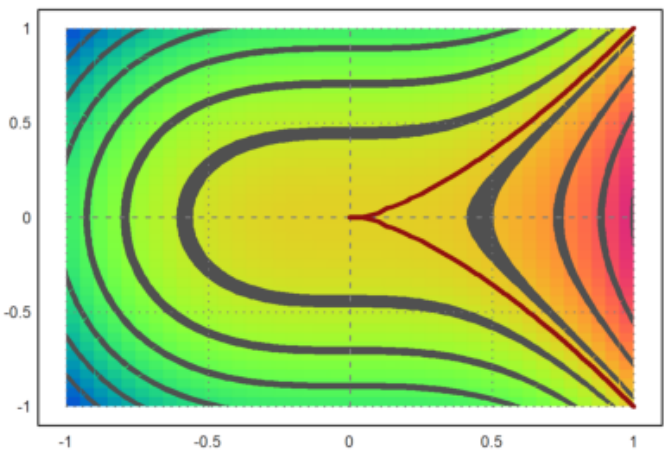
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



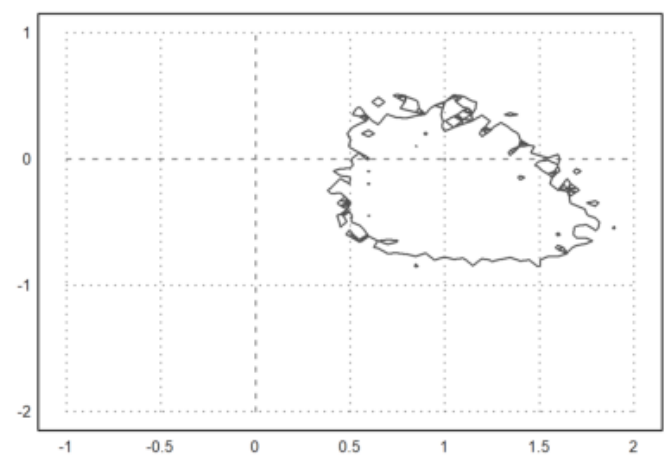

```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



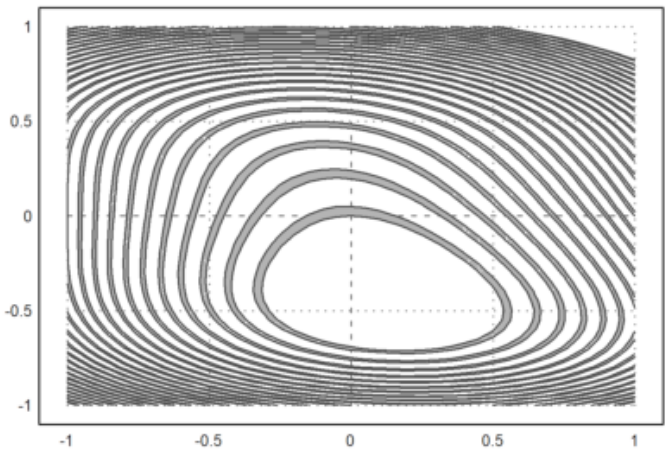

```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



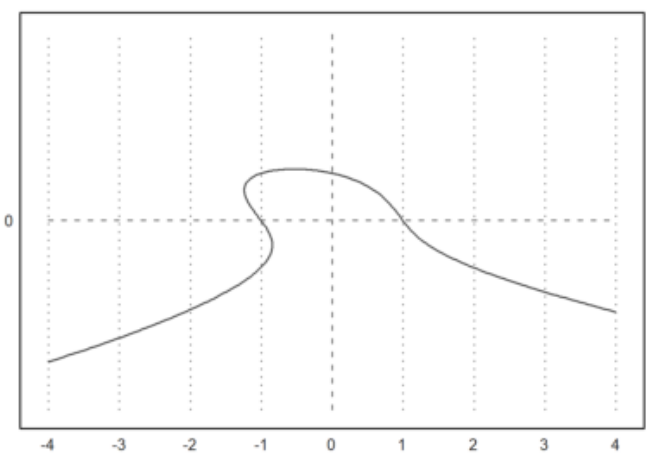

```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



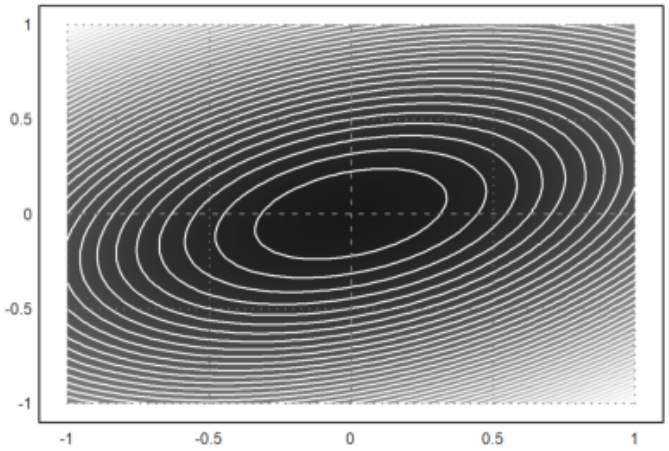
```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```




```
>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):
```

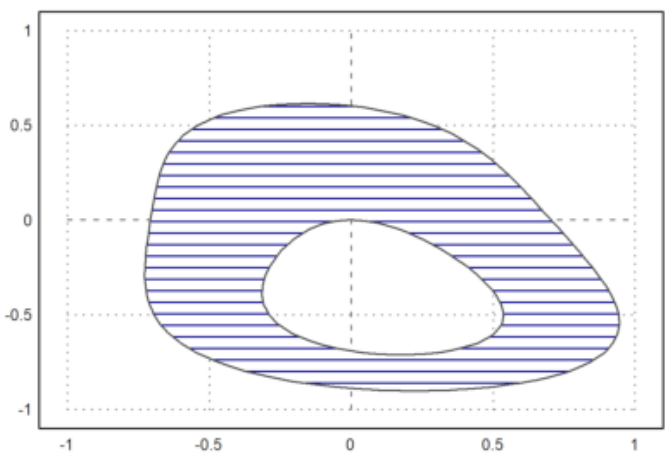


Dimungkinkan juga untuk mengisi (fill) himpunan

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

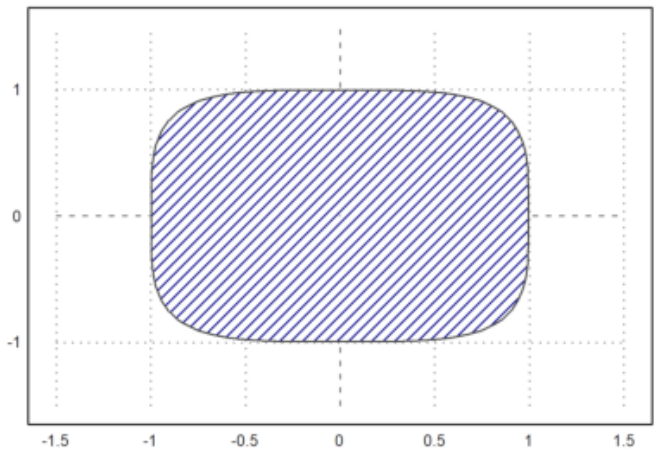
dengan rentang level (level range).
Dimungkinkan untuk mengisi daerah nilai untuk suatu fungsi tertentu.
Untuk ini, level harus berupa matriks 2×n:
- Baris pertama berisi batas bawah (lower bounds)
- Baris kedua berisi batas atas (upper bounds)

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

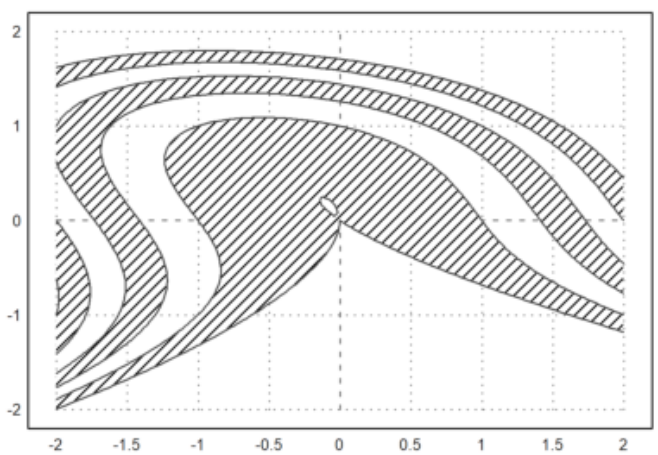


Plot implisit juga dapat menampilkan rentang level.
Dalam hal ini, level harus berupa matriks 2×n dari interval level, di mana:
- Baris pertama berisi nilai awal (start) dari setiap interval
- Baris kedua berisi nilai akhir (end) dari setiap interval
Sebagai alternatif, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level menjadi interval.

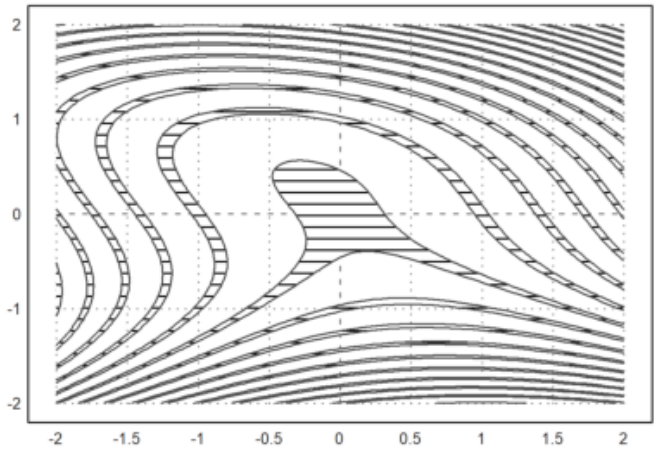
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



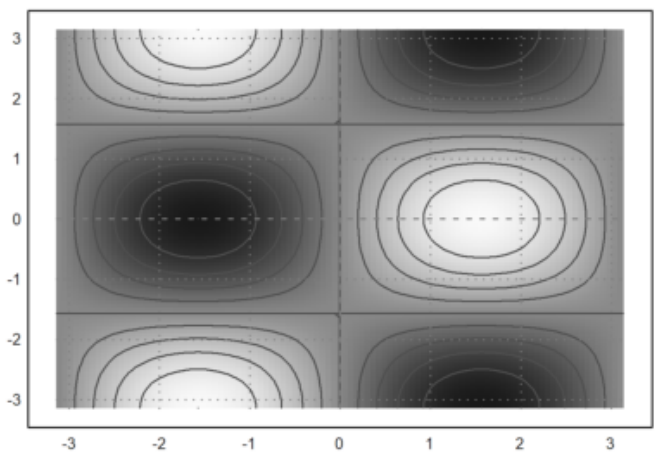

```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```




```
>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):
```

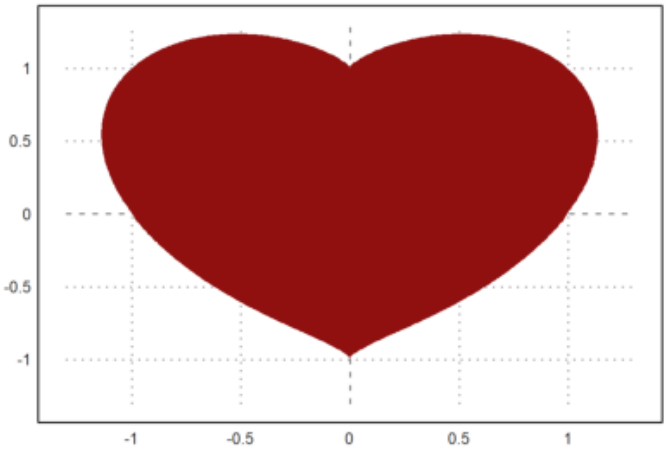


Dimungkinkan juga untuk menandai suatu daerah

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

Hal ini dilakukan dengan menambahkan level yang memiliki dua baris.

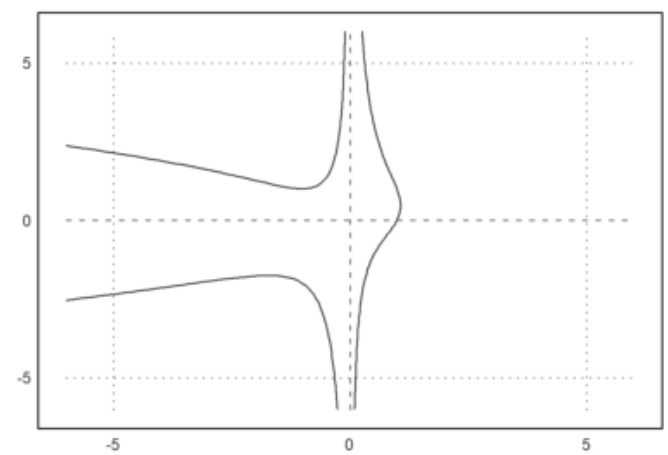
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
>  style="#",color=red,<outline, ...
>  level=[-2;0],n=100):
```



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):
```



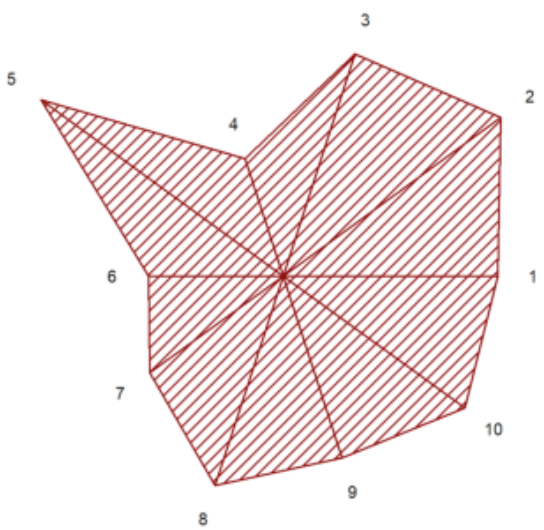

```
>function starplot1 (v, style="/", color=green, lab=none) ...
```

```
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 (axis ticks).
Selain itu, kita menggunakan jendela penuh (full window) untuk plot.
Kita memanggil reset sebelum menguji plot ini untuk mengembalikan pengaturan grafik ke default.
Hal ini tidak diperlukan jika Anda yakin plot Anda sudah bekerja dengan benar.

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



Terkadang, Anda mungkin ingin memplot sesuatu yang plot2d hampir bisa lakukan, tetapi tidak sepenuhnya.
Pada fungsi berikut, kita membuat plot impuls logaritmik (logarithmic impulse plot).
plot2d dapat membuat plot logaritmik, tetapi tidak untuk batang impuls (impulse bars).

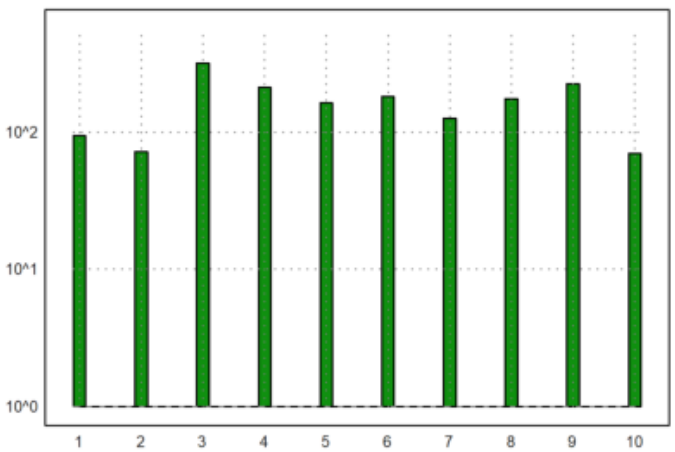
```
>function logimpulseplot1 (x,y) ...
```

```
  {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 (exponentially distributed values).

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



Mari kita membuat animasi kurva 2D menggunakan direct plots. Perintah(x,y) secara sederhana memplot kurva ke jendela plot. setplot(a,b,c,d) mengatur ukuran jendela plot.

Fungsi wait(0) memaksa plot muncul segera di jendela grafik.
Jika tidak, redraw terjadi hanya pada interval waktu tertentu (sparse time intervals).

```
>function animliss (n,m) ...
```

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

Tekan tombol apa saja untuk menghentikan animasi ini.

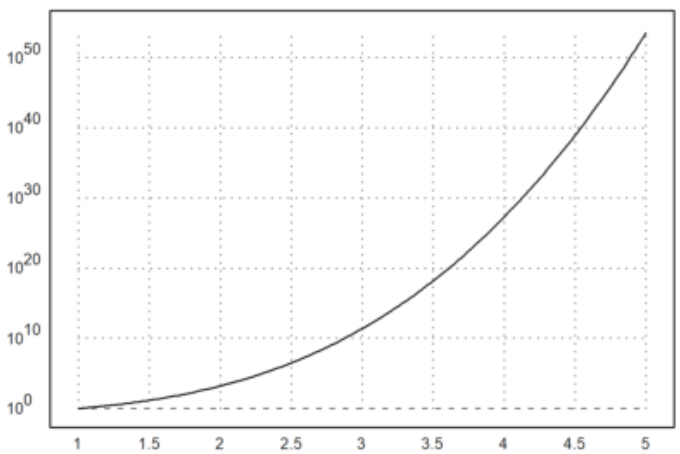
```
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

Plot Logaritmik

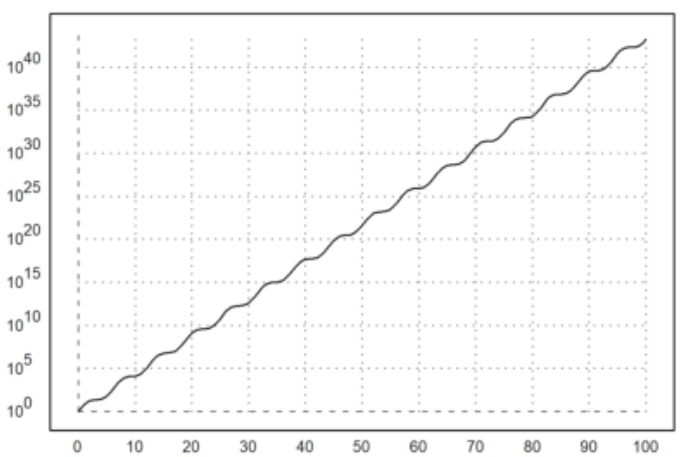
EMT menggunakan parameter logplot untuk skala logaritmik.
Plot logaritmik dapat dibuat dengan:
logplot=1 : skala logaritmik pada sumbu y
logplot=2 : skala logaritmik pada sumbu x dan y
logplot=3 : skala logaritmik pada sumbu x

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

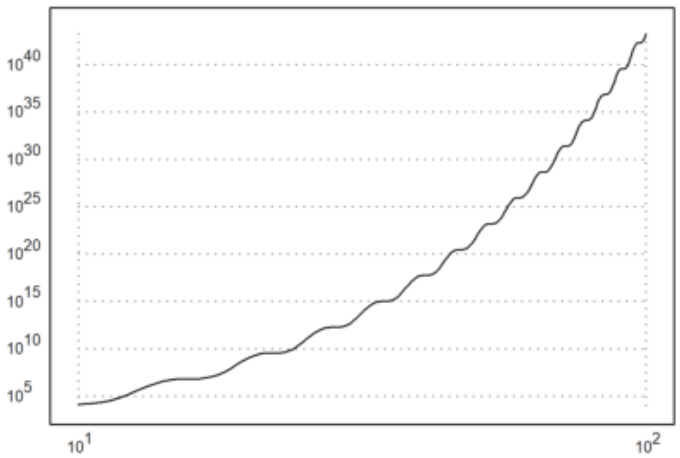
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



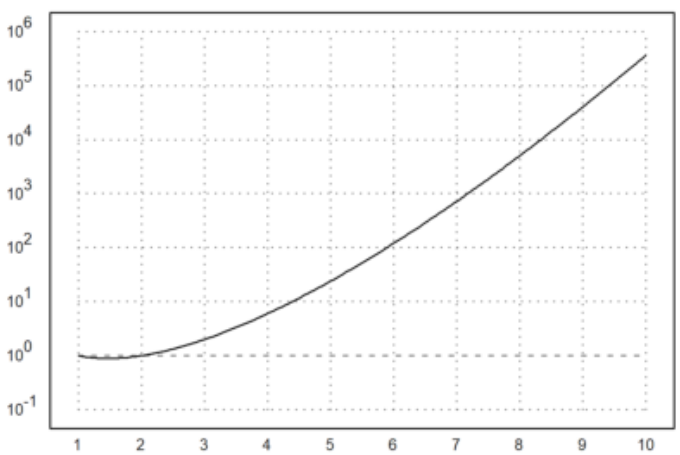

```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



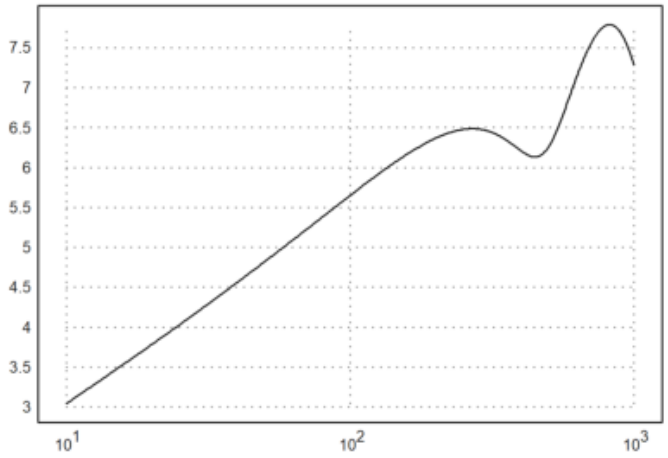
```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```




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

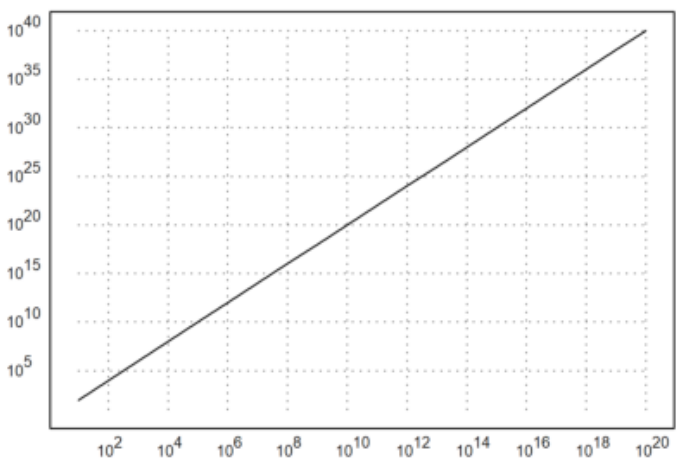



```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Hal ini juga berlaku untuk plot data.

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



CATATAN

Konsep dasar (teori singkat)

Aljabar: grafik fungsi polinomial (y=ax^2+bx+c) berbentuk parabola; derajat menentukan jumlah puncak/nilai stasioner (turunan nol) dan perilaku ujung (end behavior). Intersep dan akar (nilai x saat y=0) dapat ditentukan secara aljabar.

Kalkulus: turunan pertama f′(x) menunjukkan kemiringan dan titik kritis (maks/min lokal), turunan kedua f″(x) menentukan kelengkungan. Integral menentukan area di bawah kurva (bisa divisualisasikan sebagai wilayah terisi/filled).

Trigonometri: fungsi periodik seperti
inx,cosx memiliki amplitude, frekuensi, dan fase; perlu tampilan interval yang cukup (beberapa periode) untuk melihat pola.

Fungsi implisit & level sets: solusi f(x,y)=c menggambarkan kurva / batas yang mungkin tidak dapat dituliskan sebagai y=g(x).

Kurva parametrik: bentuk(x(t),y(t)) berguna untuk lintasan, spirals, Lissajous; parameter memberi orientasi dan kecepatan penelusuran.

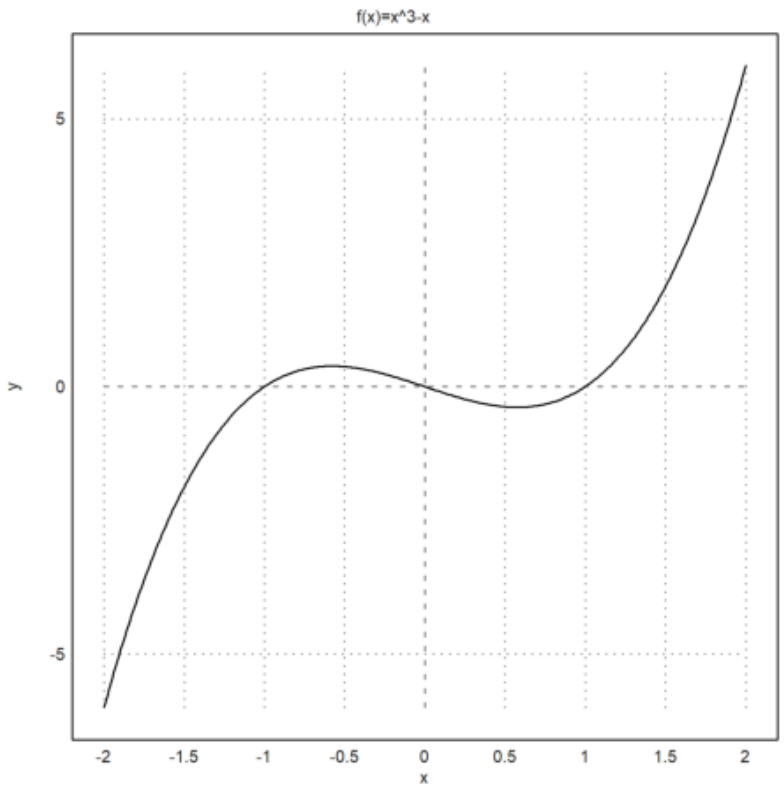
Statistika (plot distribusi): histogram / bar plot memperlihatkan sebaran frekuensi; boxplot menampilkan kuartil dan outlier.

Soal
"Gambar grafik fungsi

$$(f(x) = x^3 - x)$$

Tentukan dan tandai titik stasioner serta jelaskan jenisnya."

```
>reset;
>plot2d("x^3-x",-2,2,grid=2,title="f(x)=x^3-x",xl="x",yl="y"):
```



Grafik menunjukkan sumbu-sumbu dan kurva kubik yang melengkung pada dua sisi; titik stasioner ditemukan dari f'(x)=3x^2-1=0–>x=±1/3.

Tambahkan plot fungsi turunan plot2d(&diff("x^3-x",x), >add, style="–", color=red) untuk melihat tandingan tanda perubahan kemiringan; di titik dengan perubahan tanda dari + ke - ada maksimum lokal, dan sebaliknya.

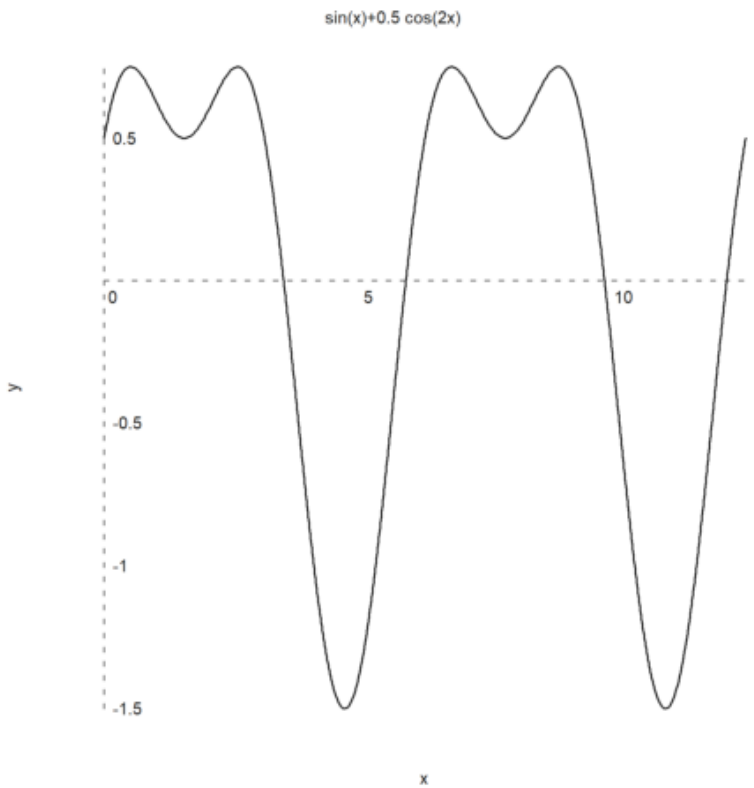
Interpretasi: titik stasioner yang berganti tanda turunan adalah maksimum/minimum lokal; lihat nilai f" untuk mengklasifikasikan

Soal
"Plot fungsi

$$(y = sin(x) + 0.5cos(2x))padainterval([0, 4pi]).$$

Jelaskan periode dan puncak."

```
>plot2d("sin(x)+0.5*cos(2*x)",0,4*pi,grid=3,title="sin(x)+0.5 cos(2x)",xl="x",yl="y"):
```



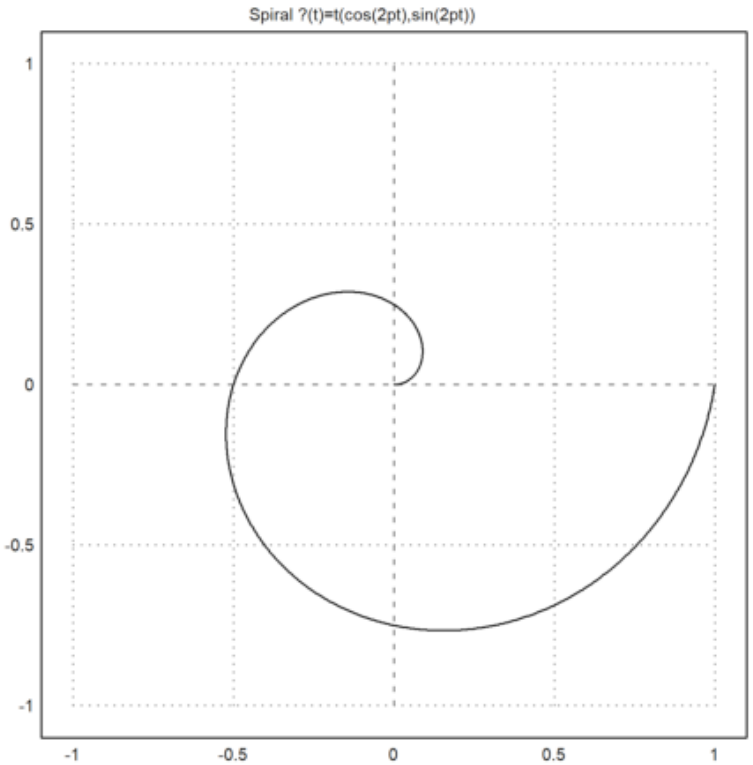
Anda akan melihat gelombang gabungan: komponen sinx (periode 2phi) dan cos2x (periode phi). Hasilnya pola berulang tetapi tidak simetris sederhana—modulasi amplitude oleh komponen kedua.
Untuk analisis puncak, cari akar turunan numerik atau titik maksimum lokal dengan fmax/fmin atau solve.

Soal
"Gambar spiral

$$(gamma(t) = t(cos(2pit), sin(2pit)))untuktanggota[0, 1].$$

Apa bentuknya?"

```
>t=linspace(0,1,1000);
>x = t*cos(2*pi*t);
>y = t*sin(2*pi*t);
>plot2d(x,y,r=1,title="Spiral ?(t)=t(cos(2pt),sin(2pt))"):
```



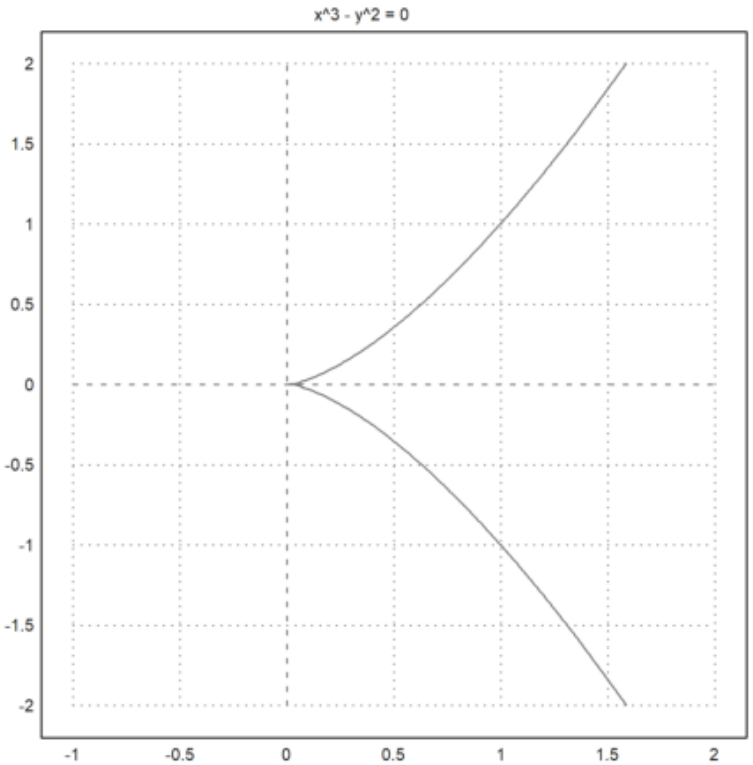
Hasilnya spiral yang mulai di pusat (0,0) dan memanjang keluar, setiap putaran sedikit lebih jauh dari pusat (radius t).
Karena t bertambah linier, titik-titik plot mengikuti arah searah jarum jam/bertahap sesuai tanda pada fungsi trig.

Soal
"Plot kurva implisit

$$(x^3 - y^2 = 0) yaitu (y = pmx^{3/2}) untuk x = 0)$$

dan jelaskan komponen grafiknya."

```
>plot2d("x^3-y^2",level=0,a=-1,b=2,c=-2,d=2,n=200,grid=2,title="x^3 - y^2 = 0"):
```



Plot akan menunjukkan dua cabang (atas dan bawah) untuk x=0; karena bentuk y^2=x^3 untuk x<0 tidak ada real y (bergantung pada pangkat).

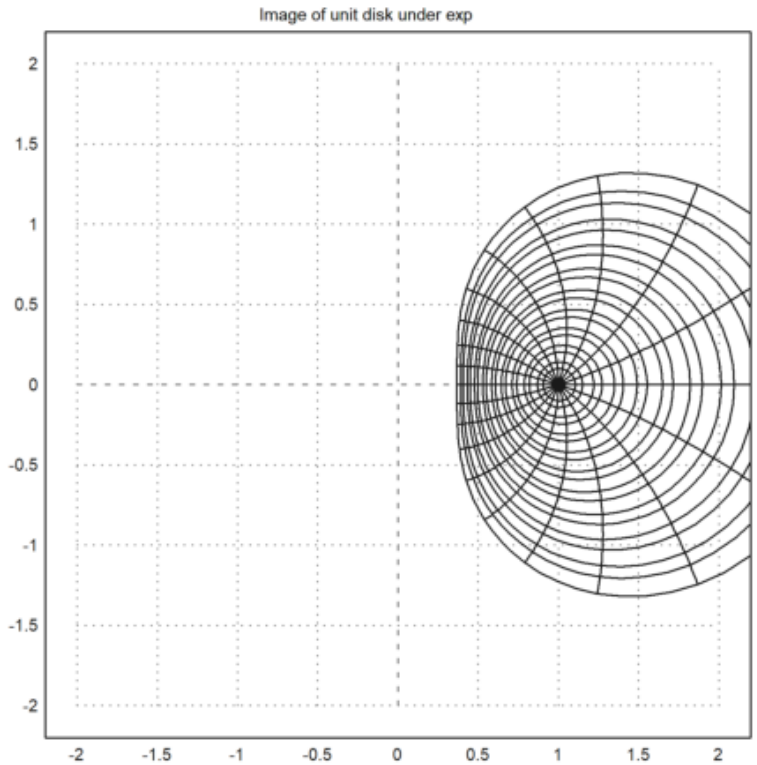
Kerapatan sampling n=200 membantu menangkap bentuk halus; gunakan >hue atau contourcolor untuk memperjelas jika perlu.

Soal
"Plot gambar dari peta

$$z <-> e^z$$

pada kisi disk unit (plot real vs imag). Apa yang terlihat?"

```
>r=linspace(0,1,50); a=linspace(0,2*pi,80)';
>z = r*exp(I*a);
>plot2d(exp(z), cgrid=[20,20], r=2, title="Image of unit disk under exp"):
```



Fungsi eksponensial memetakan kotak/bundel complex ke lingkaran/radial pattern; unit disk menjadi area di kompleks yang memanjang, menampilkan garis-garis konformal. cgrid mengontrol grid visibility.

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)
```

Multipurpose plot function for plots in the plane (2D plots). This function can do plots of functions of one variables, data plots, curves in the plane, bar plots, grids of complex numbers, and implicit plots of functions of two variables.

Parameters
x,y : equations, functions or data vectors
a,b,c,d : Plot area (default a=-2,b=2)
r : if r is set, then a=cx-r, b=cx+r, c=cy-r, d=cy+r

```
r can be a vector [rx,ry] or a vector [rx1,rx2,ry1,ry2].
```

xmin,xmax : range of the parameter for curves
auto : Determine y-range automatically (default)
square : if true, try to keep square x-y-ranges
n : number of intervals (default is adaptive)
grid : 0 = no grid and labels,

```
1 = axis only,
2 = normal grid (see below for the number of grid lines)
3 = inside axis
4 = no grid
5 = full grid including margin
6 = ticks at the frame
7 = axis only
8 = axis only, sub-ticks
```

frame : 0 = no frame
framecolor: color of the frame and the grid
margin : number between 0 and 0.4 for the margin around the plot
color : Color of curves. If this is a vector of colors,

```
it will be used for each row of a matrix of plots. In the case of
point plots, it should be a column vector. If a row vector or a
full matrix of colors is used for point plots, it will be used for
each data point.
```

thickness : line thickness for curves

```
This value can be smaller than 1 for very thin lines.
```

style : Plot style for lines, markers, and fills.

```
For points use
"[]", "<>", ".", "..", "...",
"*", "+", "|", "-", "o"
"[]#", "<>#", "o#" (filled shapes)
"[]w", "<>w", "ow" (non-transparent)
For lines use
"-", "--", "-.", ".", ".-.", "-.-", "->"
For filled polygons or bar plots use
"#", "#O", "O", "/", "\", "\/",
"+", "|", "-", "t"
```

points : plot single points instead of line segments
addpoints : if true, plots line segments and points
add : add the plot to the existing plot
user : enable user interaction for functions
delta : step size for user interaction
bar : bar plot (x are the interval bounds, y the interval values)
histogram : plots the frequencies of x in n subintervals
distribution=n : plots the distribution of x with n subintervals
even : use inter values for automatic histograms.
steps : plots the function as a step function (steps=1,2)
adaptive : use adaptive plots (n is the minimal number of steps)
level : plot level lines of an implicit function of two variables
outline : draws boundary of level ranges.
If the level value is a 2xn matrix, ranges of levels will be drawn
in the color using the given fill style. If outline is true, it
will be drawn in the contour color. Using this feature, regions of

f(x,y) between limits can be marked.
hue : add hue color to the level plot to indicate the function

```
            value
```

contour : Use level plot with automatic levels
nc : number of automatic level lines
title : plot title (default "")
xl, yl : labels for the x- and y-axis
smaller : if >0, there will be more space to the left for labels.
vertical :

```
  Turns vertical labels on or off. This changes the global variable
  verticallabels locally for one plot. The value 1 sets only vertical
  text, the value 2 uses vertical numerical labels on the y axis.
```

filled : fill the plot of a curve
fillcolor : fill color for bar and filled curves
outline : boundary for filled polygons
logplot : set logarithmic plots

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

own :

```
  A string, which points to an own plot routine. With >user, you get
  the same user interaction as in plot2d. The range will be set
  before each call to your function.
```

maps : map expressions (0 is faster), functions are always mapped.
contourcolor : color of contour lines
contourwidth : width of contour lines
clipping : toggles the clipping (default is true)
title :

```
  This can be used to describe the plot. The title will appear above
  the plot. Moreover, a label for the x and y axis can be added with
  xl="string" or yl="string". Other labels can be added with the
  functions label() or labelbox(). The title can be a unicode
  string or an image of a Latex formula.
```

cgrid :

```
  Determines the number of grid lines for plots of complex grids.
  Should be a divisor of the the matrix size minus 1 (number of
  subintervals). cgrid can be a vector [cx,cy].
```

Overview

The function can plot

- expressions, call collections or functions of one variable,
- parametric curves,
- x data against y data,
- implicit functions,
- bar plots,
- complex grids,
- polygons.

If a function or expression for xv is given, plot2d() will compute values in the given range using the function or expression. The expression must be an expression in the variable x. The range must be defined in the parameters a and b unless the default range should be used. The y-range will be computed automatically, unless c and d are specified, or a radius r, which yields the range r,r for x and y. For plots of functions, plot2d will use an adaptive evaluation of the function by default. To speed up the plot for complicated functions, switch this off with <adaptive, and optionally decrease the number of intervals n. Moreover, plot2d() will by default use mapping. I.e., it will compute the plot element for element. If your expression or your functions can handle a vector x, you can switch that off with <maps for faster evaluation.

Note that adaptive plots are always computed element for element. If functions or expressions for both xv and for yv are specified, plot2d() will compute a curve with the xv values as x-coordinates and the yv values as y-coordinates. In this case, a range should be defined for the parameter using xmin, xmax. Expressions contained in strings must always be expressions in the parameter variable x.