Menggambar Grafik 2D dengan EMT

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

Identitas

Nama : Intan Nur Setyarini
NIM : 24030130001
Kelas: Pendidikan Matematika

Plot Dasar

Ada fungsi dasar yang sangat sederhana untuk membuat plot. Terdapat koordinat layar (screen coordinates), yang selalu memiliki rentang dari 0 sampai 1024 pada setiap sumbu, terlepas apakah layar berbentuk persegi atau tidak. Selain itu, terdapat koordinat plot (plot coordinates), yang dapat diatur dengan setplot(). Pemetaan antara koordinat layar dan koordinat plot bergantung pada jendela plot yang sedang digunakan. Sebagai contoh, fungsi shrinkwindow() secara bawaan akan menyisakan ruang untuk label sumbu dan judul grafik.

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; // bersihkan layar
>window(0,0,1024,1024); // gunakan seluruh jendela
>setplot(0,1,0,1); // atur koordinat plot
>hold on; // mulai mode  overwrite
>n=100; X=random(n,2); Y=random(n,2);  // ambil titik acak
>colors=rgb(random(n),random(n),random(n)); // ambil warna acak
>loop 1 to n; color(colors[#]); plot(X[#],Y[#]); end; // buat plot tiap titik
>hold off; // akhiri mode overwrite
>insimg; // sisipkan ke dalam notebook
```

```
>reset;
```

Program EMT tersebut diawali dengan membersihkan layar menggunakan perintah clg, kemudian seluruh jendela digunakan sebagai kanvas dengan window(0,0,1024,1024).
Setelah itu, sistem koordinat diatur dari (0,0) sampai (1,1) melalui setplot(0,1,0,1), lalu mode gambar bertumpuk diaktifkan dengan hold on agar setiap objek yang digambar tidak saling menghapus.
Selanjutnya, dibuat 100 titik acak dengan X dan Y sebagai koordinatnya, serta 100 warna acak yang dihasilkan melalui fungsi rgb. Fungsi plot(X[], Y[]) di EMT bukan hanya menggambar satu titik, tetapi bisa membuat garis yang terhubung sesuai koordinat. Titik-titik tersebut kemudian digambar satu per satu dalam perulangan, di mana setiap titik diberi warna yang berbeda sesuai daftar warna acak.
Setelah seluruh titik selesai digambar, mode tumpang tindih dinonaktifkan dengan hold off, dan akhirnya hasil plot disisipkan ke dalam notebook menggunakan perintah insimg.
perintah reset untuk Mengembalikan semua pengaturan (jendela, koordinat, variabel, warna, dll.) ke kondisi awal default EMT.

Sangat diperlukan untuk hold grafik, karena perintah plot() akan menghapus jendela plot.

Untuk menghapus semua yang telah kita buat, kita gunakan reset().

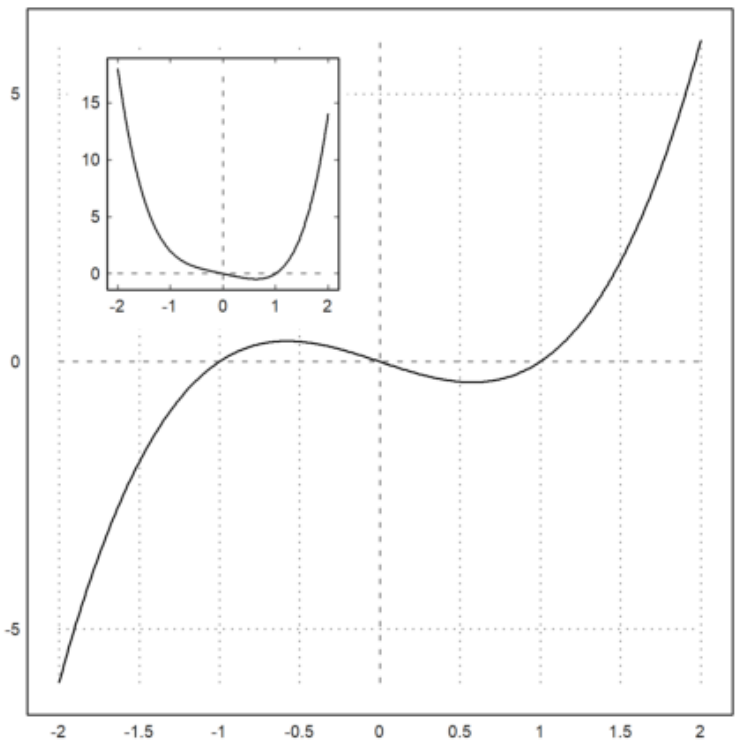
Untuk menampilkan gambar hasil plot di layar notebook, perintah plot2d() dapat diakhiri dengan titik dua (:). Cara lain adalah perintah plot2d() diakhiri dengan titik koma (;), kemudian menggunakan perintah insimg() untuk menampilkan gambar hasil plot.

Untuk contoh lain, kita menggambar sebuah plot sebagai inset di dalam plot yang lain. Hal ini dilakukan dengan mendefinisikan jendela plot yang lebih kecil. Perlu diperhatikan bahwa jendela ini tidak menyediakan ruang untuk label sumbu di luar jendela plot. Kita harus menambahkan margin seperlunya untuk hal ini. Perlu dicatat juga bahwa kita menyimpan dan memulihkan jendela penuh, serta menahan plot yang sedang aktif ketika kita menggambar inset.

```
>plot2d("x^3-x"); // Menampilkan grafik fungsi y=x^
```

```
3
-x pada jendela plot utama.
```

```
>xw=200; yw=100; ww=300; hw=300; // Mendefinisikan ukuran dan posisi inset plot
>ow=window(); // Menyimpan pengaturan jendela saat ini
>window(xw,yw,xw+ww,yw+hw); // Mengubah jendela aktif menjadi area kecil (inset)
>hold on; // Mengaktifkan mode overwrite, supaya grafik inset tidak menghapus grafik utama
>barclear(xw-50,yw-10,ww+60,ww+60); // Membersihkan area di sekitar inset dengan kotak kos
>plot2d("x^4-x",grid=6): // Menggambar grafik fungsi
```




```
>hold off; // Menutup mode hold
>window(ow); // Mengembalikan jendela ke pengaturan awal
```

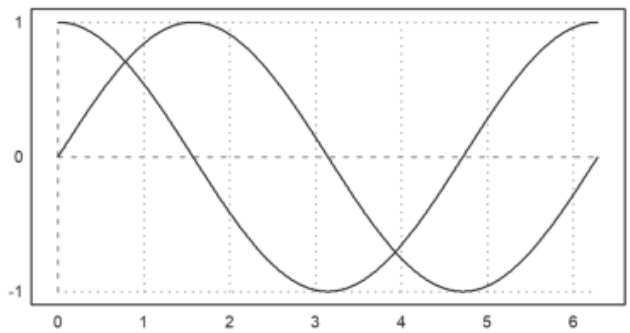
Sebuah plot dengan beberapa gambar dapat dibuat dengan cara yang sama. Terdapat fungsi utilitas figure() untuk tujuan ini. Fungsi figure() disebut fungsi utilitas karena membantu kita membuat beberapa gambar (multiple figures) dengan cara yang sederhana, tanpa perlu mengatur ulang jendela manual setiap kali.

Aspek Plot

Plot bawaan menggunakan jendela plot berbentuk persegi. Kamu dapat mengubahnya dengan fungsi aspect(). Jangan lupa untuk mengatur ulang (reset) aspek tersebut setelahnya. Kamu juga bisa mengubah pengaturan bawaan ini di menu dengan memilih "Set Aspect", baik ke rasio aspek tertentu maupun ke ukuran saat ini dari jendela grafik.

Namun, kamu juga bisa mengubah aspek hanya untuk satu plot saja. Untuk hal ini, ukuran area plot saat ini akan diubah, dan jendela akan diatur agar label memiliki ruang yang cukup.

```
>aspect(2); // rasio panjang dan lebar 2:1 // mengatur rasio aspek (aspect ratio)
>plot2d(["sin(x)","cos(x)"],0,2pi): // Menggambar dua fungsi dengan rentang 0-2phi
```




```
>aspect(); // mengembalikan rasio aspek
>reset;
```

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

2D Plots di Euler

EMT Math Toolbox memiliki plot dalam 2D, baik untuk data maupun fungsi. EMT menggunakan fungsi plot2d. Fungsi ini dapat digunakan untuk memplot fungsi maupun data.

Dimungkinkan untuk membuat plot di Maxima menggunakan Gnuplot atau di Python menggunakan Matplotlib.

Euler dapat membuat plot 2D dari

- ekspresi: misalnya plot2d("4*x^2") untuk grafik parabola.
- fungsi, variabel, atau kurva terparametrisasi: misalnya menggambar fungsi yang didefinisikan sendiri atau kurva berdasarkan parameter.
- vektor nilai x-y: jika punya data berpasangan (x,y), bisa diplot langsung sebagai grafik.
- sekumpulan titik (clouds of points) pada bidang: ntuk data acak atau scatter plot.
- kurva implisit dengan level atau daerah level: misalnya grafik level (contour plot) atau daerah level dari persamaan dua variabel.
- fungsi kompleks: visualisasi fungsi bilangan kompleks dalam bidang 2D.

Gaya plot mencakup berbagai gaya untuk garis dan titik, plot batang, serta plot berarsir. plot2d mendukung berbagai gaya tampilan seperti gaya garis dan titik, bar plot, atau plot arsir.

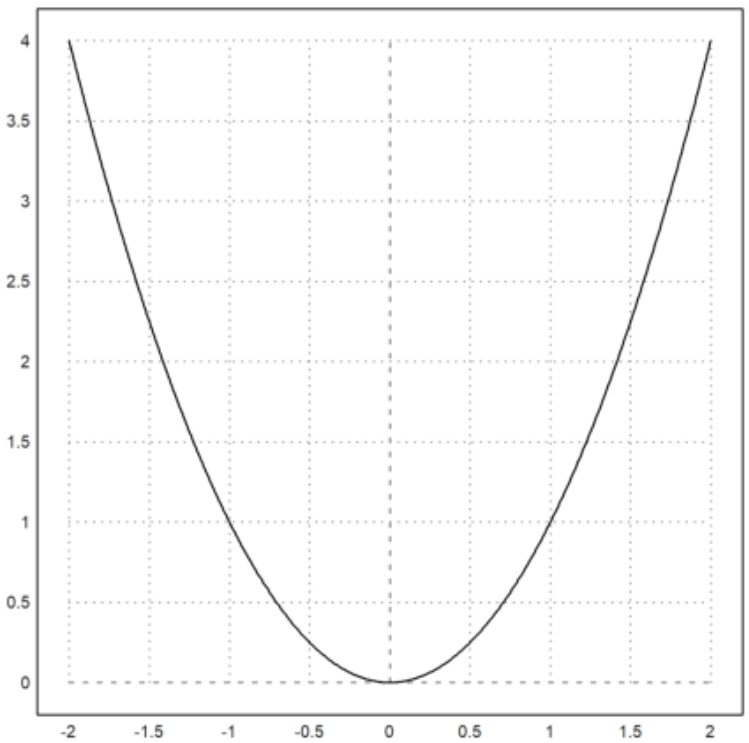
Plot Ekspresi atau Variabel

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

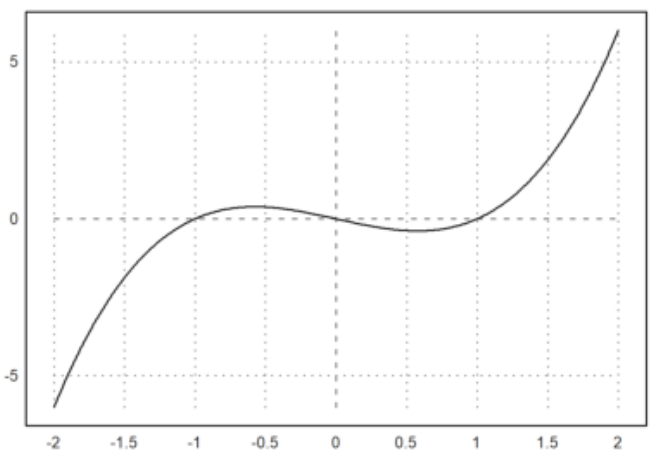
Berikut adalah contoh paling dasar, yang menggunakan rentang bawaan (default range) dan mengatur rentang y yang sesuai agar grafik fungsi dapat ditampilkan dengan benar.

Catatan: Jika sebuah baris perintah diakhiri dengan tanda titik dua :, maka plot akan disisipkan ke dalam jendela teks. Jika tidak, tekan TAB untuk melihat plot apabila jendela plot tertutup.

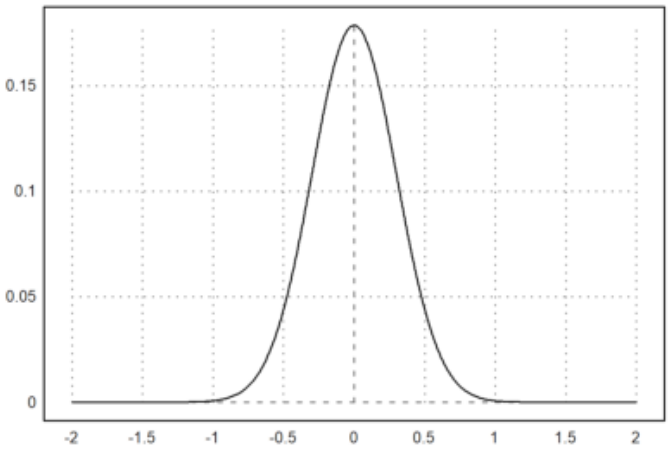
```
>plot2d("x^2"):
```




```
>aspect(1.5); plot2d("x^3-x"): // aspect digunakan untuk rasio panjang dan lebar
```




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

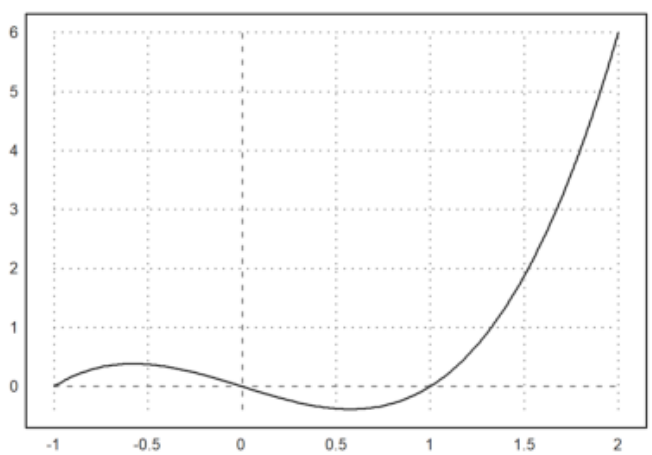


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

Rentang plot diatur dengan parameter berikut:

- a,b: rentang sumbu-x (default: -2 sampai 2)
- c,d: rentang sumbu-y (default: diskalakan sesuai nilai fungsi)
- r: sebagai alternatif, jari-jari di sekitar pusat plot
- cx,cy: koordinat pusat plot (default: 0,0)

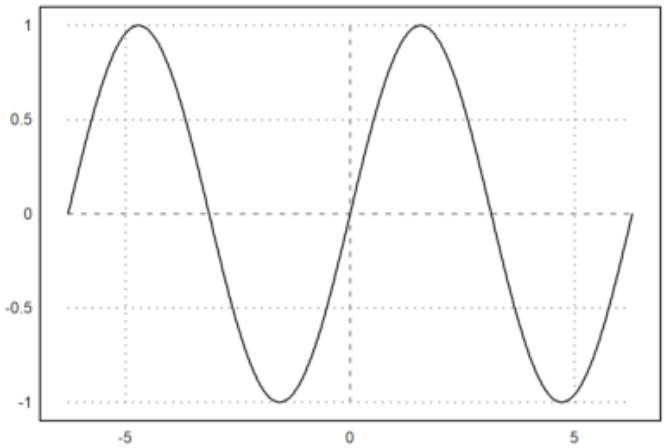
```
>plot2d("x^3-x",-1,2):
```



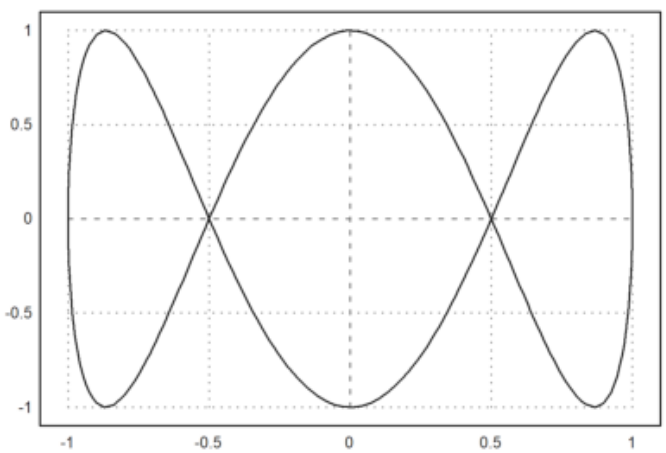
Rentang sumbu x pada grafik tersebut dibatasi dari -1 sampai 2.

Rentang y otomatis disesuaikan agar grafik terlihat jelas.

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




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



Perintah ini tersebut memplot kurva parametrik.
Sumbu x diberikan oleh cos(x). Sumbu y diberikan oleh sin(3x). Parameter x bervariasi dari 0 sampai pi.

Hasilnya adalah kurva khusus (mirip dengan kurva Lissajous) karena melibatkan kombinasi cosinus dan sinus dengan frekuensi berbeda.

Insimg(lines) digunakan untuk menyisipkan plot dengan ukuran tertentu yang menempati sejumlah baris teks.

Dalam opsi, plot dapat diatur untuk muncul:

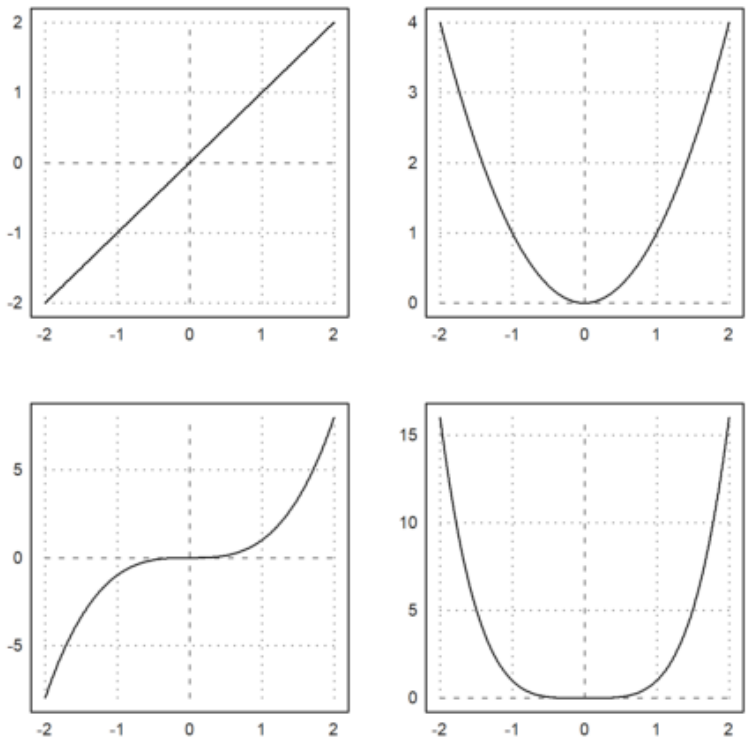
- di jendela terpisah yang bisa diubah ukurannya
- Di jendela notebook.

Lebih banyak gaya dapat dicapai dengan perintah plot tertentu.

Bagaimanapun juga, tekan tombol tabulator untuk melihat plot jika plot tersebut tersembunyi.

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

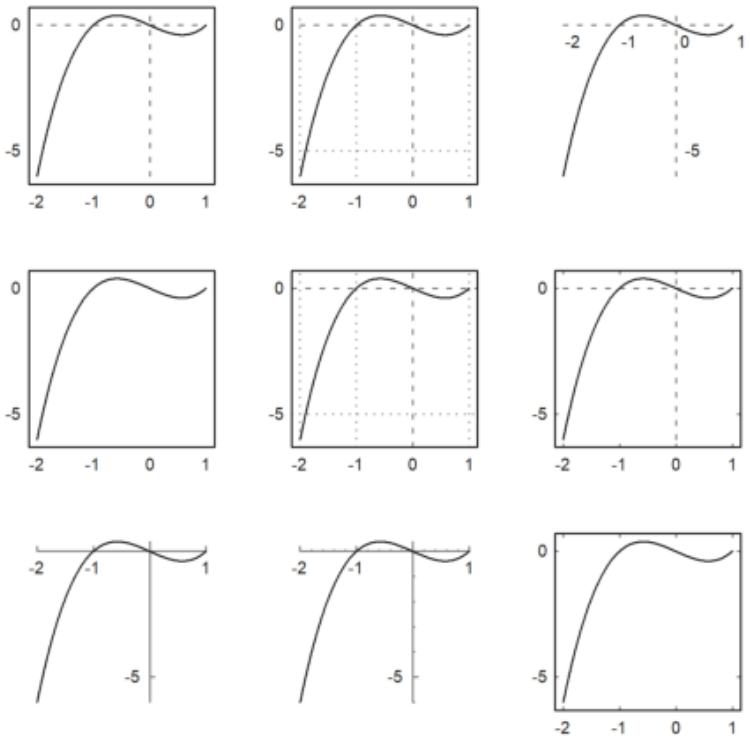
```
>reset;
>figure(2,2); ...
>for n=1 to 4; figure(n); plot2d("x^"+n); end; ...
>figure(0): // mengembalikan grafik ke default
```



4 grafik berbeda secara berdampingan dalam satu tampilan karena perintah figure (2,2).
Terdapat looping dari n=1 sampai 4 dengan fungsi y=x^n sesuai urutan kotak. Sehingga kotak 1 menggambarkan fungsi y=x^1, kotak 2 menggambar fungsi y=x^2, dst.

Dalam plot2d(), terdapat gaya alternatif yang tersedia dengan grid = x. Untuk gambaran umum, kita dapat menampilkan berbagai gaya grid dalam satu gambar (lihat di bawah untuk perintah figure()).
aya grid = 0 tidak disertakan. Gaya ini menampilkan tanpa grid dan tanpa bingkai.

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



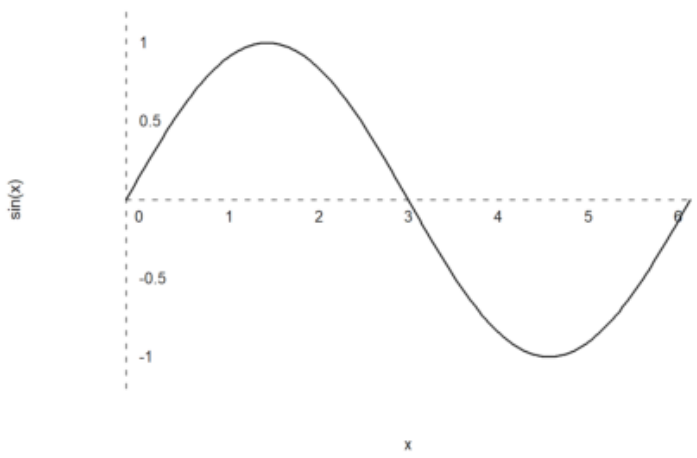
Grafik tersebut menunjukkan fungsi y=x^3-3 dengan rentang x dari -2 sampai 1 dan parameter grid = 2. Jadi, setiap subplot memakai gaya grid berbeda sesuai nomor preset. Gaya grid ini sudah tersedia secara default di EMT.

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

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

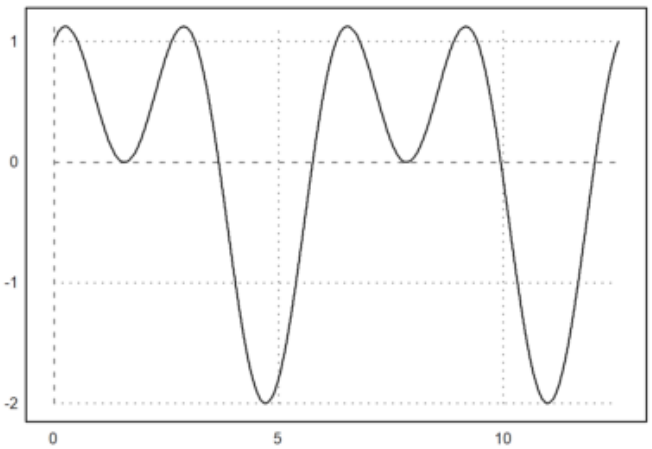
Dalam contoh berikut, kita mengubah gaya grid, menambahkan label, dan menggunakan label vertikal untuk sumbu-y.

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



Grafik tersebut menunjukkan grafik fungsi y=sin(x) dengan rentang x dari 0 sampai 2phi dan rentang y dari -1.2 sampai 1.2. Perintah grid 3 untuk memilih tampilan gaya grid nomor 3. Perintah xl memberi label pada sumbu x dan yl memberi label pada sumbu y.

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



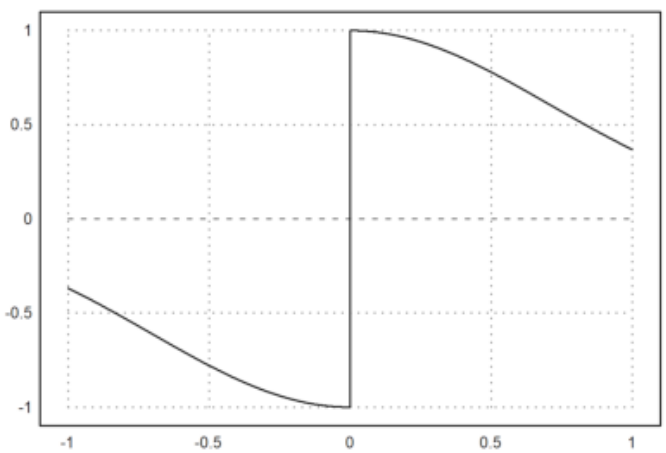
Grafik tersebut menunjukkan fungsi y=sin(x)+cos(2x) dengan rentang x dari 0 sampai 4phi sehingga terlihat gabungan gelombang sin dan cos dengan periode berbeda. sin(x) peridode 2phi dan cos(2x) periode phi.

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

Anda bisa dengan mudah menandai gambar apa pun dan menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, Anda juga bisa mengekspor grafik saat ini menggunakan fungsi yang ada di menu File.

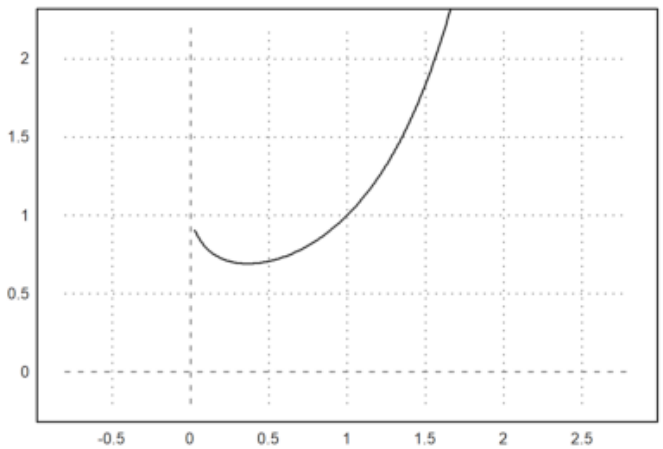
Fungsi atau ekspresi dalam plot2d dievaluasi secara adaptif. Untuk mempercepat, Anda dapat mematikan plot adaptif dengan <adaptive dan menentukan jumlah subinterval dengan n=.... Namun, hal ini biasanya hanya diperlukan dalam kasus-kasus tertentu saja.

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



Grafik tersebut menunjukkan fungsi f(x)=sign(x).e^-x^2
Grafik memperlihatkan simetri kiri-kanan, dengan lompatan kecil di titik
=0 karena fungsi sign. n=10000 artinya Euler akan membagi interval jadi 10.000 titik, supaya kurva lebih halus.

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

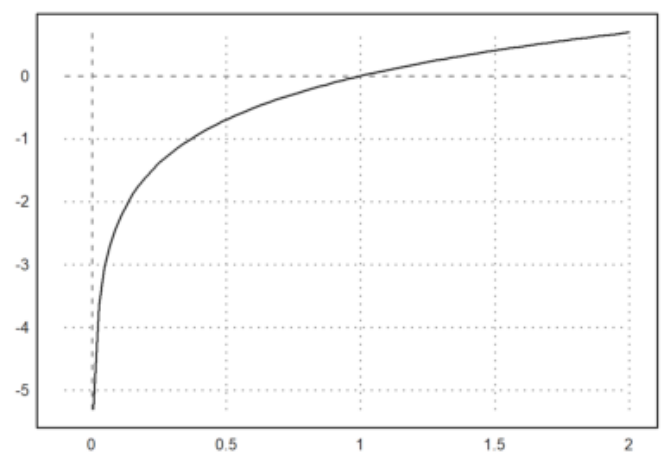


Grafik tersebut menunjukkan fungsi

$$f(x) = x^x$$

Fungsi ini terdefinisi nyata untuk x>0 dan memiliki minimum x=1/e.
r=1.2 artinya radius sekitar pusat grafik = 1.2.
cx=1, cy=1 artinya pusat grafik berada di koordinat (1,1).
Catatan bahwa x^x tidak terdefinisi untuk
<= 0. Fungsi plot2d menangkap kesalahan ini, dan mulai memplot segera setelah fungsi tersebut terdefinisi.
Hal ini berlaku untuk semua fungsi yang menghasilkan NAN (Not A Number) di luar daerah definisinya.

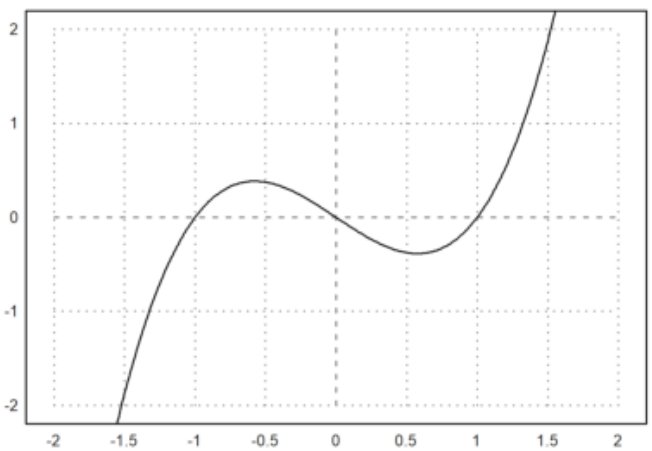
```
>plot2d("log(x)",-0.1,2):
```



Pada grafik tersebut terlihat bagian interval [-0.1,0) tidak terdefinisi dan menghasilkan NAN sehingga grafik muncul setelah x>0.

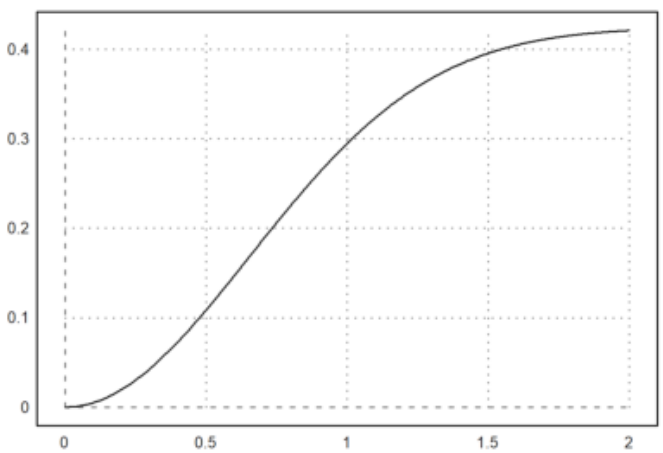
Parameter square=true (atau >square) secara otomatis memilih rentang y sehingga hasilnya berupa jendela plot berbentuk persegi. Perlu dicatat bahwa secara default, Euler menggunakan ruang berbentuk persegi di dalam jendela plot.

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



Penggunaan perintah square menjadikan grafik tersebut proporsional dan tidak terdistorsi (miasalnya terlalu tinggi atau gepeng).

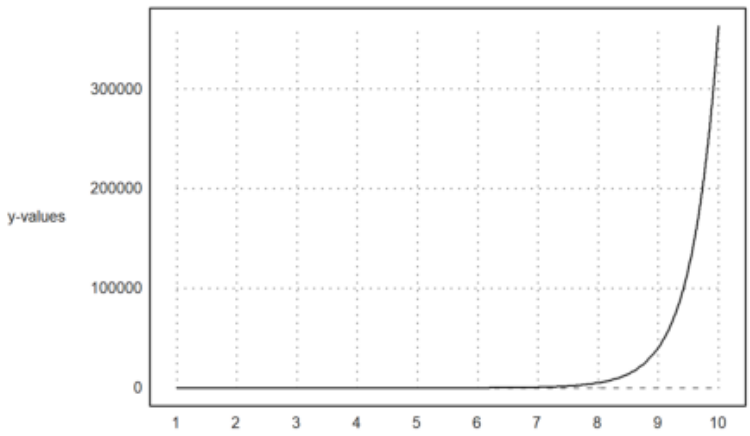
```
>plot2d(''integrate("sin(x)*exp(-x^2)",0,x)'',0,2): // plot integral
```



Karena integrand sin(x)e^(-x^2) berosilasi tetapi mengecil (karena faktor e^(-x^2) mengecil cepat), grafik integralnya akan
aik-turun kecil tapi cenderung menuju nilai mendekati konstan.

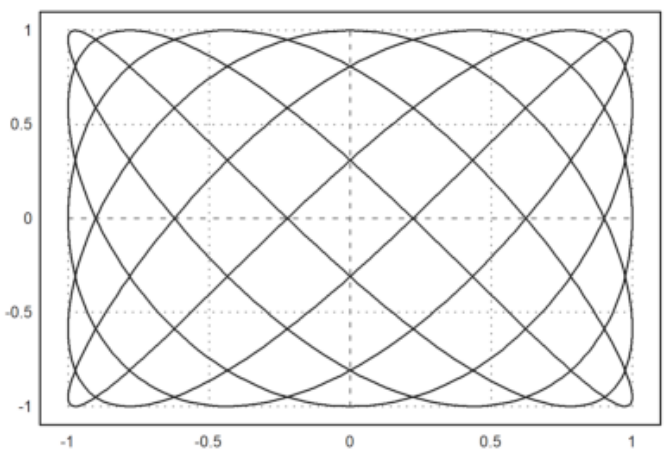
Jika Anda memerlukan lebih banyak ruang untuk label sumbu-y,panggil shrinkwindow() dengan parameter yang lebih kecil, atau atur nilai positif untuk "smaller" di dalam plot2d().

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



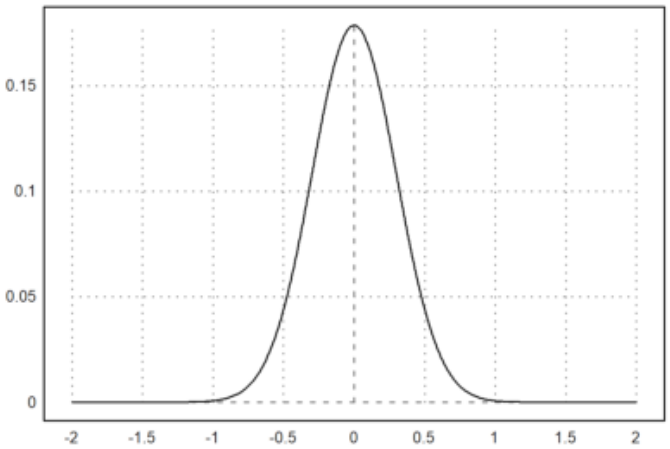
Fungsi yang dipanggil adalah gamma function.
Fungsi gamma adalah generalisasi dari faktorial dan tumbuh sangat cepat (eksponensial) seiring x membesar. Perintah smaller=6 memberi lebih banyak ruang untuk label sumbu, sehingga label tidak terpotong oleh batas jendela grafik dan perintah <vertical membuat label pada sumbu-y ditulis secara vertikal, bukan horizontal.
Ekspresi simbolik juga dapat digunakan, karena ekspresi tersebut disimpan sebagai ekspresi string sederhana.

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

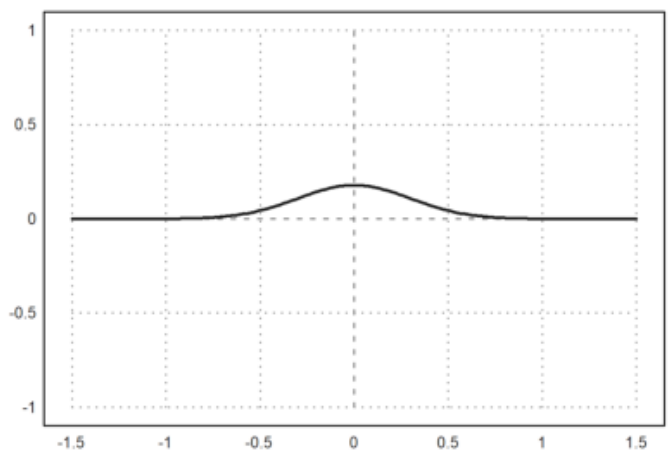


Kurva tersebut merupakan kurva Lissajous. Fungsi yang mungkin dipakai untuk menghasilkan kurva tersebut adalah x(t)=sin(5t), y(t)=cos(7t), dengan t anggota [0,2phi].

```
>a:=5.6; expr &= exp(-a*x^2)/a; // mendefinisikan ekspresi
>plot2d(expr,-2,2): // memplot pada interval -2 sampai 2
```

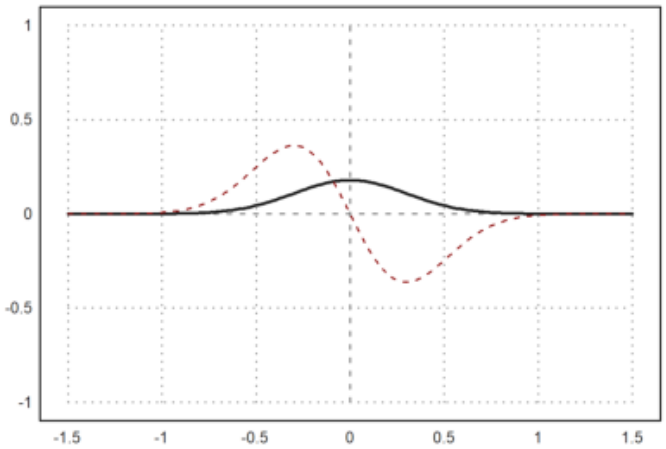



```
>plot2d(expr,r=1,thickness=2): // memplot gambar di sekitar (0,0)
```



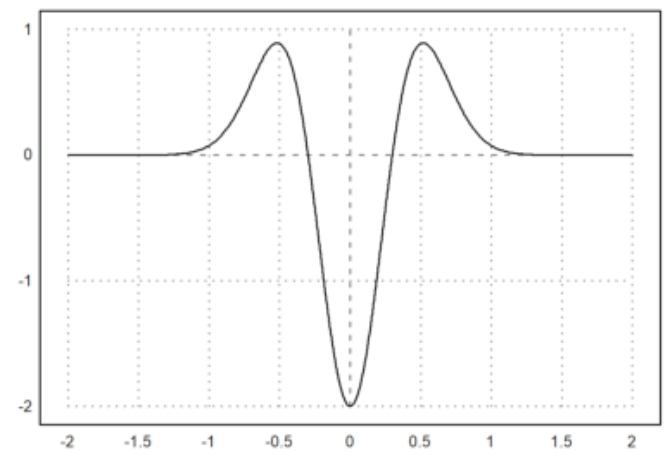
Plot diatur dengan radius r=1, jadi jendela grafik berupa persegi di sekitar titik pusat (0,0). Perintah thickness=2 mempertebal garis grafik.

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



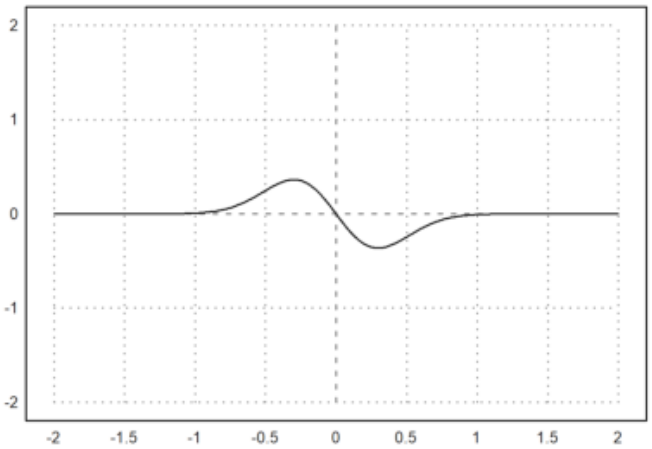
Kurva tersebut menggambarkan turunan fungsi pertama f(x)=(e^(-ax^2))/a sehingga f'(x)=(-2axe^9-ax^2))/a yang digambarkan dengan garis putus-putus.

```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // memplot dalam jendela persegi panjang
```



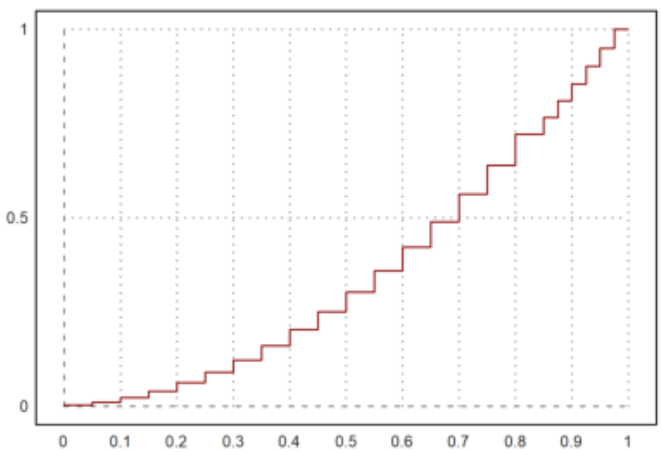
Kurva tersebut menggambarkan turunan kedua dari fungsi dengan interval x dari -2 sampai 2 dan interval y dari -2 sampai 1. Interval tersebut menjadikan tampilan persegi panjang.

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



Perintah square digunakan agar tampilan kurva proporsi dengan skala sumbu x serta y sama dan tidak terlihat melebar atau gepeng.

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

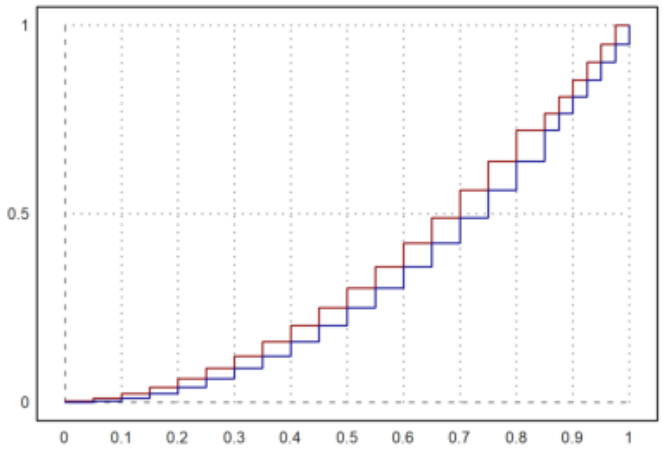


Kurva tersebut menggambarkan fungsi

$$f(x) = x^2$$

dengan interval [0,1] yang dibagi menjadi 10 titik x (n=10) sehingga ada 11 titik total. Perintah Steps=1 menggambarkan garis lurus antar titik sehingga kurva parabola hanya didekati dengan segmen garis.

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



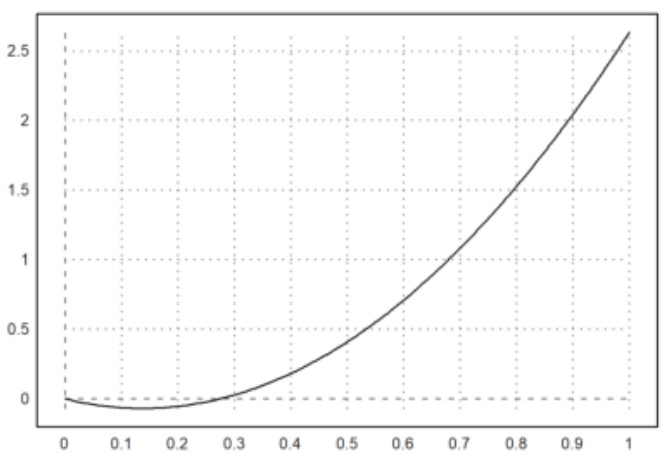
Perintah >add menambahkan kurva halus berwarna biru ke kurva sebelumnya.

Fungsi dengan Satu Parameter

Fungsi plot yang paling penting untuk grafik bidang datar adalah plot2d(). Fungsi ini diimplementasikan dalam bahasa Euler pada file "plot.e", yang dimuat saat awal program dijalankan

Berikut beberapa contoh penggunaan fungsi. Seperti biasa dalam EMT, fungsi yang bekerja untuk fungsi lain atau ekspresi dapat diberikan parameter tambahan (selain x), yang bukan merupakan variabel global, ke dalam fungsi dengan parameter titik koma atau dengan call collection.

```
>function f(x,a) := x^2/a+a*x^2-x; // mendefinisikan fungsi dg parameter a
>a=0.3; plot2d("f",0,1;a): // memplot a=0.3
```

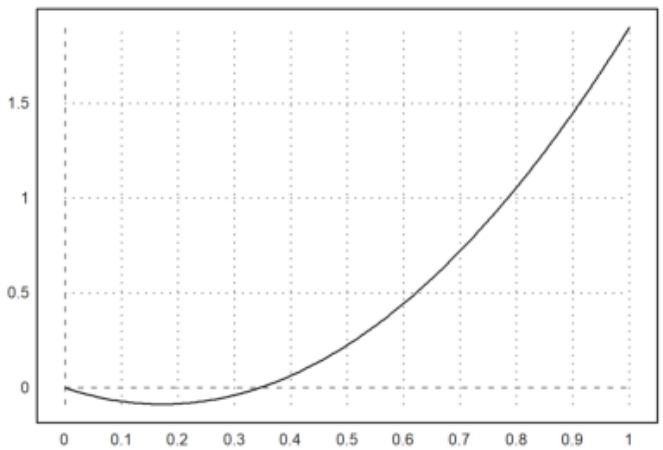


Kurva tersebut menggambarkan fungsi yang sudah didefinisikan dengan nilai parameter a=0.3 sehingga fungsi menjadi

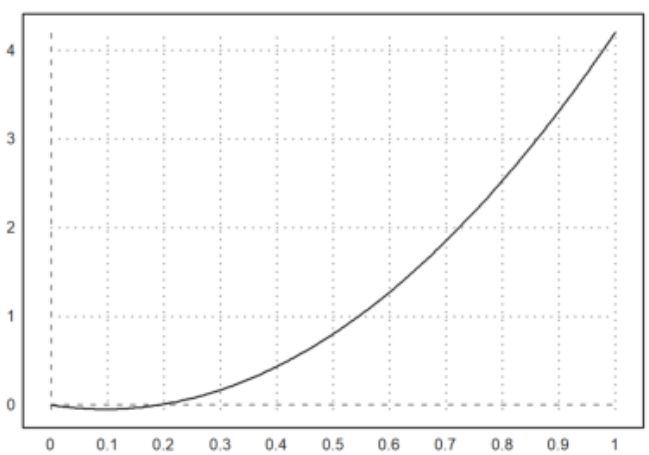
$$f(x) = 3.633x^2 - x$$

Kurva tersebut berbentuk parabole ke atas karena a>0 dan kurva berada pada interval x dari o sampai 1.

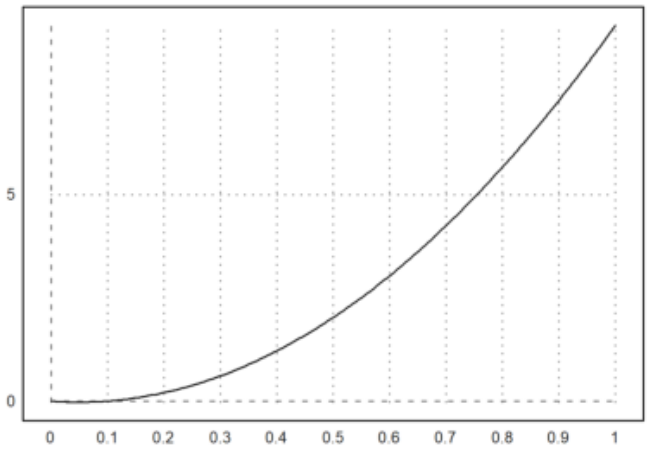
```
>plot2d("f",0,1;0.4): // plot dg a=0.4
```



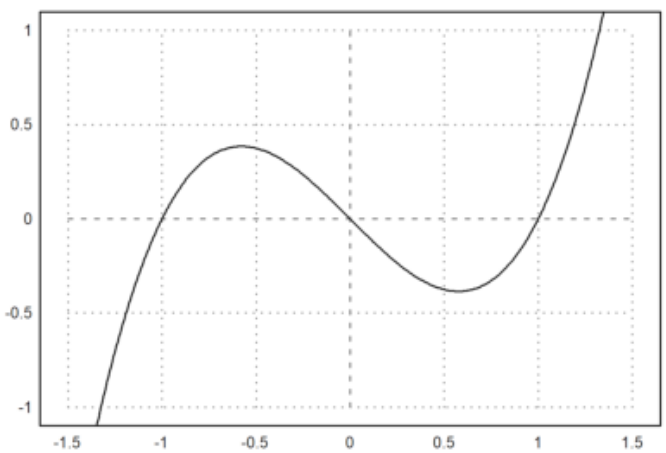

```
>plot2d({{"f",0.2}},0,1): // plot dg a=0.2
```




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




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



Kurva tersebut menggambarkan kurva kubik dengan fungsi

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

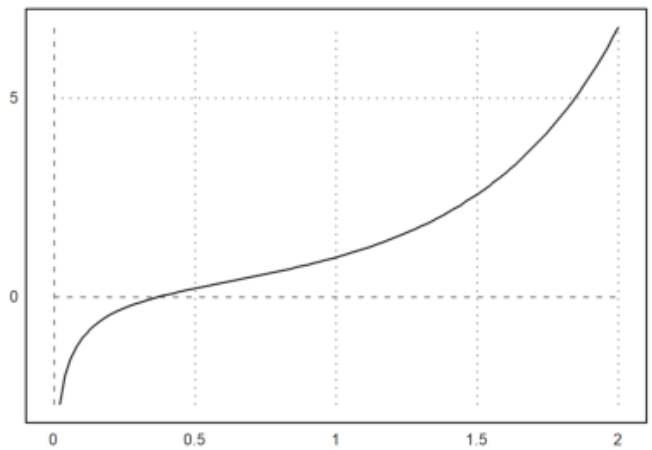
Parameter r=1 berfungsi untuk memberi jangkauan atau interval pada sumbuu-x [-1,1] dan sumbu-y [-1,1].
Berikut ringkasan fungsi yang diterima:
- ekspresi atau ekspresi simbolik dalam variabel x
- fungsi atau fungsi simbolik dengan nama seperti "f"
- fungsi simbolik cukup dengan menyebutkan namanya saja, misalnya f

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

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

```
                x
               x  (log(x) + 1)
```

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



Kurva tersebut menggambarkan kurva turunan dari fungsi f(x)=x^x dterhadap x sehingga

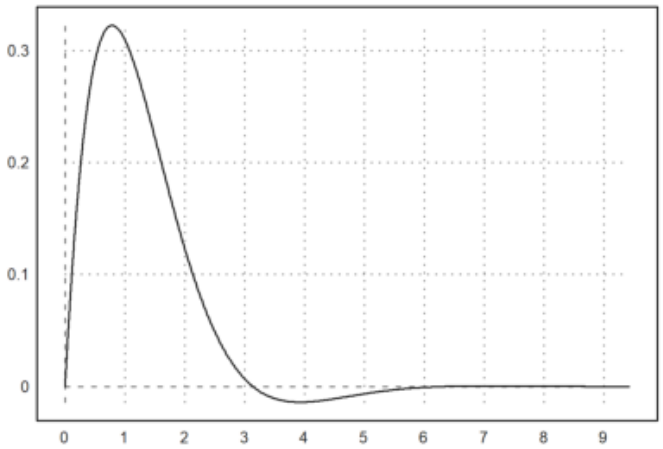
$$f`(x) = x^x(ln(x) + 1)$$

dengan interval sumbu-x [0-2]. Kurva tersebut tidak terdefinisi untuk x<= 0
Tentu saja, untuk ekspresi atau ekspresi simbolik, cukup menggunakan nama variabel saja untuk memplotnya.

```
>expr &= sin(x)*exp(-x) // mendefinisikan ekspresi simbolik
```

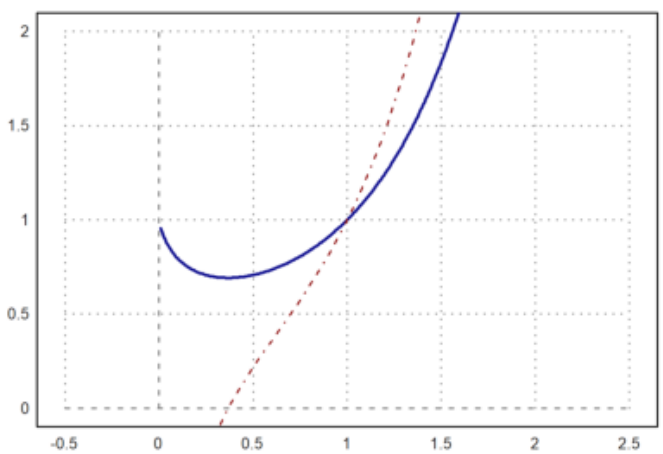
```
                 - x
               E    sin(x)
```

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



Kurva tersebut menggambarkan fungsi f(x)=sin(x)e^-x pada interval x dari 0 sampai 3phi. Jadi, kurva tersebut menampilkan kurva yang amplitudonya mengecil seiring bertambahnya x. Hal itu karena faktor e yang mengecil.

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



Kurva tersebut menunjukkan fungsi

$$f(x) = x^x$$

berwarna biru dengan ketebalan 2 yang di tambahkan dengan perintah >add untuk menggambarkan kurva dari fungsi turunan dari fungsi awal. Kurva turunan digambar dengan warna merah dan garis putus-putus.

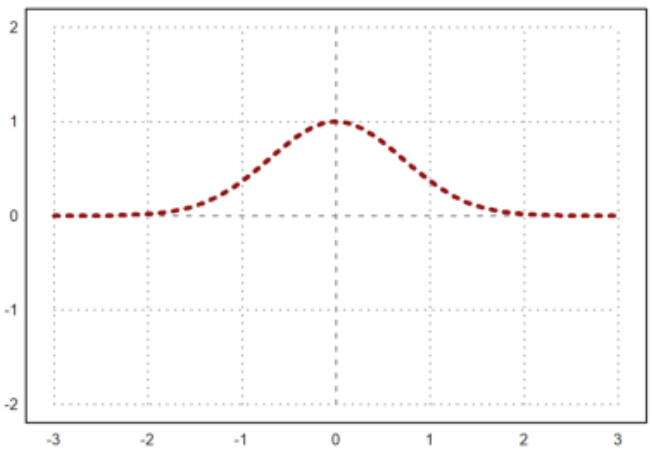
Untuk gaya garis (line style) tersedia berbagai opsi:

- style="...". Pilih dari "-", "–", "-.", ".", ".-.", "-.-".
- waena: lihat daftar warna di bawah.
- thickness: ketebalan garis (default 1).

Pilihan warna bisa menggunakan warna bawaan (default colors) atau ditentukan dengan model RGB.

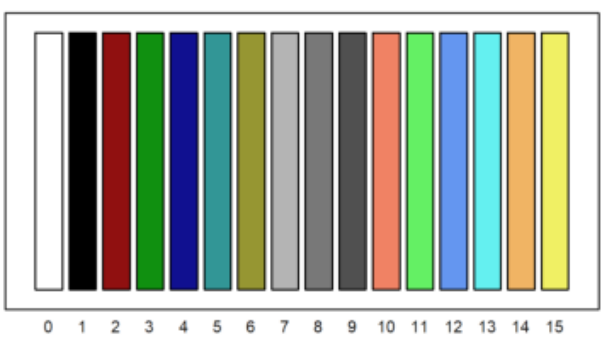
- 0..15: indeks warna bawaan.
- Konstanta warna: white, black, red, green, blue, cyan, olive, lightgray, gray, darkgray, orange, lightgreen, turquoise, lightblue, lightorange, yellow
- rgb(red,green,blue): parameter berupa bilangan real dalam interval[0,1].

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



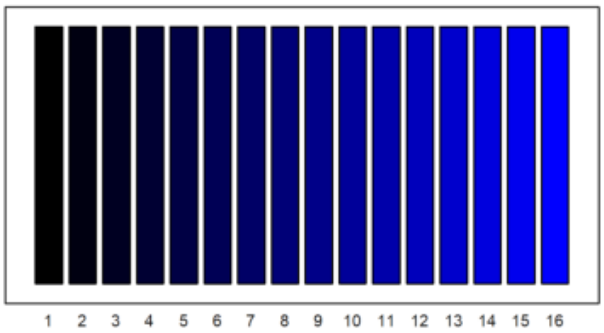
Berikut adalah tampilan dari warna-warna yang telah ditentukan sebelumnya (predefined colors) di EMT.

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



Tetapi Anda dapat menggunakan warna apa pun.

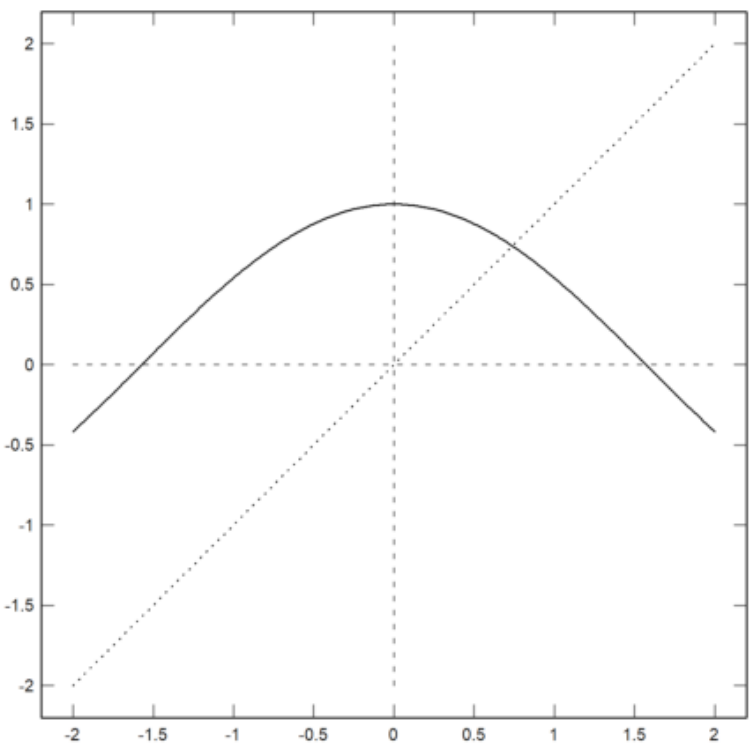
```
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```



Gambar Beberapa Kurva pada bidang koordinat yg sama

Memplot lebih dari satu fungsi (multiple functions) dalam satu jendela dapat dilakukan dengan beberapa cara. Salah satu metodenya adalah menggunakan >add untuk beberapa pemanggilan plot2d, kecuali pada pemanggilan pertama. Fitur ini sudah kita gunakan pada contoh-contoh sebelumnya.

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



Kurva tersebut menggambarkan fungsi

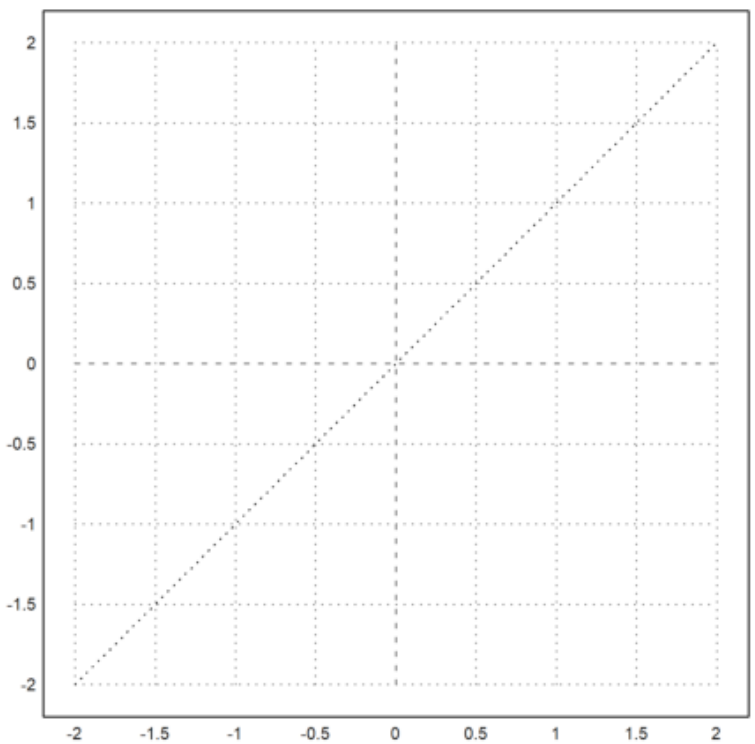
$$f(x) = cos(x)$$

dengan interval sumbu-x dan sumbu-y [-2,2]. Kemudian kurva tersebut ditambah fungsi

$$g(x) = x$$

dengan perintah >add. Fungsi kedua digambar dengan style "."

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



Kurva tersebut hanya menggambarkan fungsi

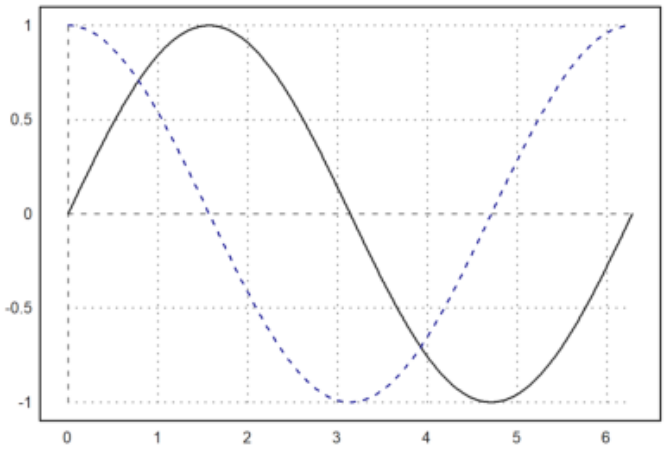
$$g(x) = x$$

saja tanpa menggambar kurva fungsi

$$f(x) = cos(x)$$

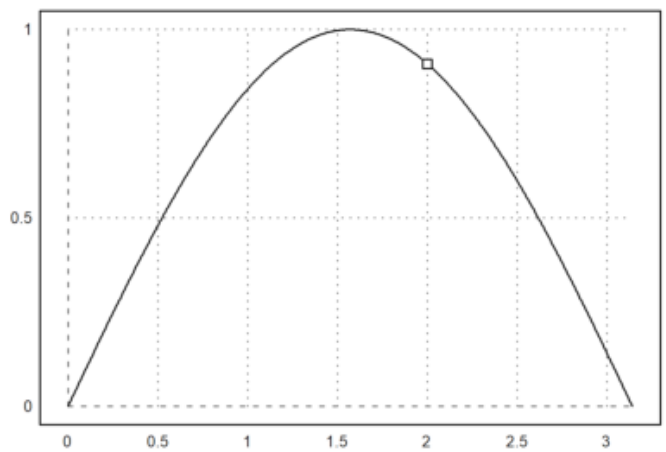
karena tidak ditambah perintah >add.

```
>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.

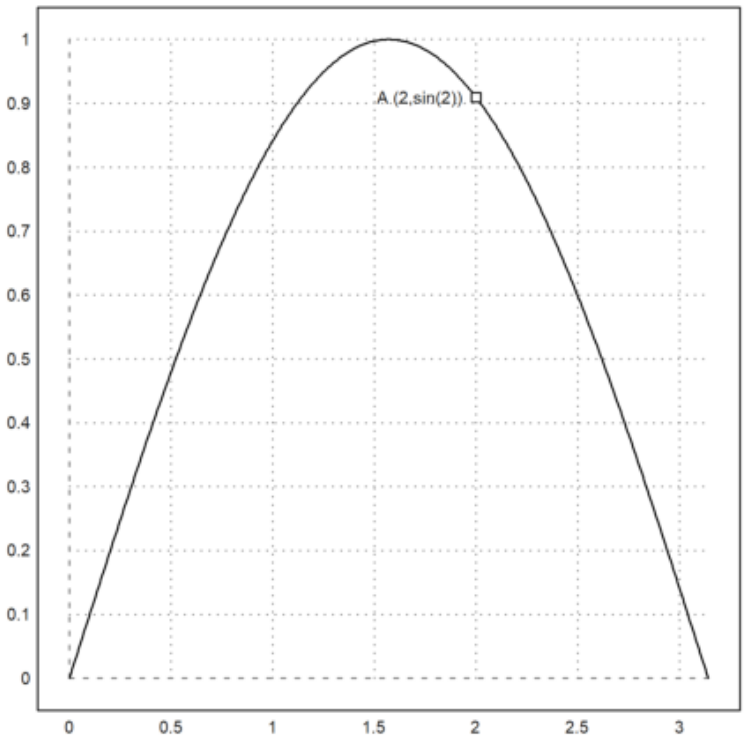
```
>plot2d("sin(x)",0,pi); plot2d(2,sin(2),>points,>add):
```



Kurva tersebut menggambarkan fungsi sin (x) dari x=0 sampai phi. Kemudian ditambahkan satu titik ke kurva dengan perintah >points, >add. Titik yang diplot adalah (2,sin(2)) atau (2,0.9093).

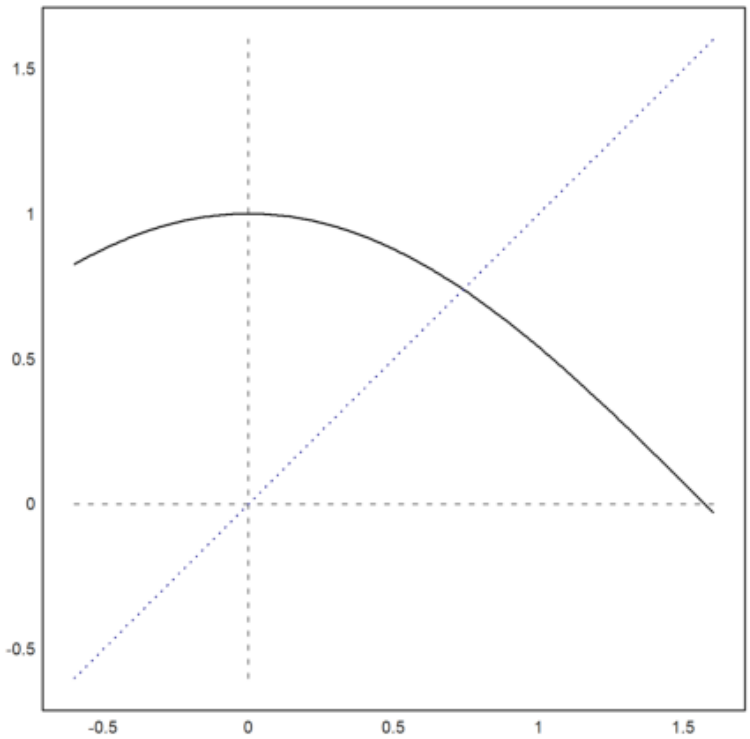
Kita menambahkan titik potong dengan sebuah label (pada posisi "cl" yaitu center left), dan menyisipkan hasilnya ke dalam notebook. Kita juga menambahkan sebuah judul pada plot tersebut.

```
>plot2d("sin(x)",0,pi); plot2d(2,sin(2),>points,>add); ...
>label("A (2,sin(2))",2,sin(2),pos="cl"):
```



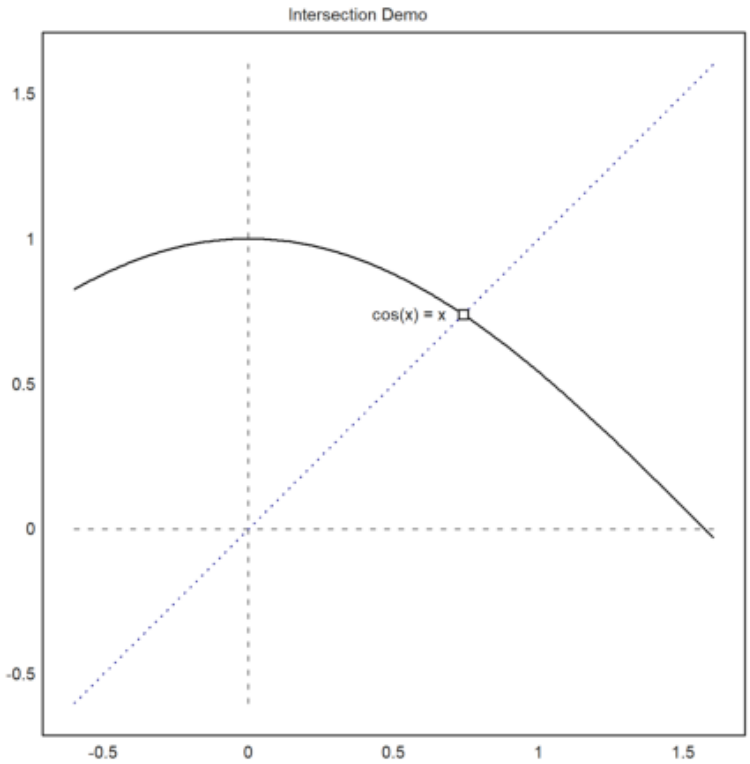
Perintah label memberikan nama pada titik yang dipilih dengan posisi cl (center left). Berikut posisi yang dapat digunakan untuk posisi label titik:
- cr (center right)
- tl (top left)
- tr (top right)
- bl (bottom left)
- br (bottom right)

```
>plot2d(["cos(x)","x"],r=1.1,cx=0.5,cy=0.5, ...
>  color=[black,blue],style=["-","."], ...
>  grid=1):
```



Perintah tersebut menampilkan kurva dari 2 fungsi yaitu fungsi f(x)=cos(x) dan f(x)=x. Style dari fungsi cos adalah garis sedangkan style fungsi f(x)=y adalah titik-titik

```
>x0=solve("cos(x)-x",1);  ...
>  plot2d(x0,x0,>points,>add,title="Intersection Demo");  ...
>  label("cos(x) = x",x0,x0,pos="cl",offset=20):
```



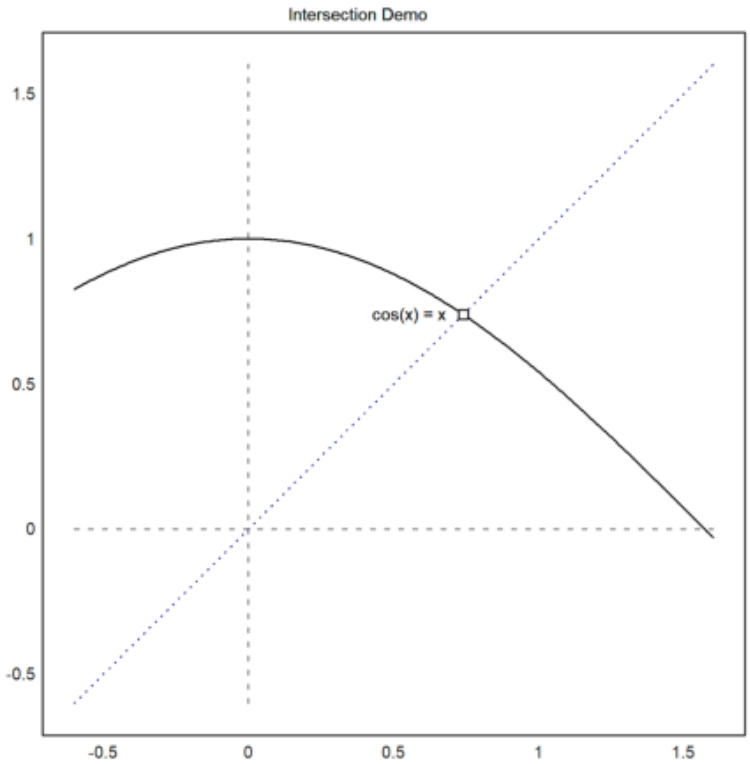
Perintah tersebut untuk menentukan titik potong kurva dari 2 fungsi tersebut atau mencari solusi dari persamaan cos(x)-x=0 dengan tebakan awal 1.
Kemudian, titik potong tersebut diberi label dan memberikan judul pada plot. Label pada titik digeser sebesar 20 piksel agar tidak menutupi titik.

```
>x0=solve("cos(x)-x");  ...
>  plot2d(x0,x0,>points,>add,title="Intersection Demo");  ...
>  label("cos(x) = x",x0,x0,pos="cl",offset=20):
```

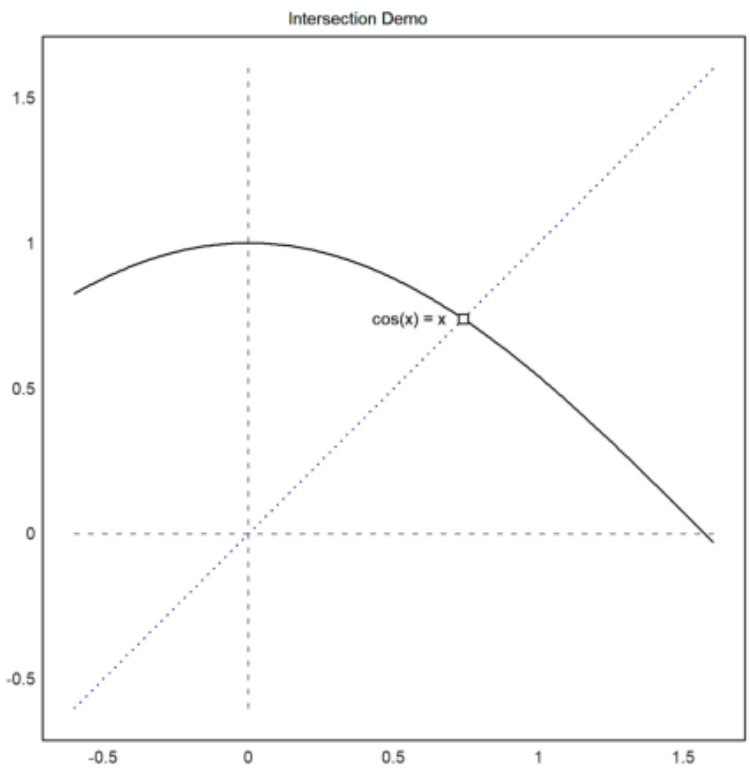
```
  Function solve needs at least 2 arguments!
  Use: solve (f$: call, a: scalar complex {, b: none, y: number, eps: none})
  Error in:
  x0=solve("cos(x)-x");    plot2d(x0,x0,>points,>add,title="Inte ...
                      ^
```

Perintah tersebut eror karena tidak menyertakan angka 1 pada perintah solve. Hal ini karena EMT memerlukan nilai awal (numerik) untuk memulai proses. tanpa angka 1, maka program tidak tahu dari mana harus memulai pencarian akar, sehingga muncul error.

```
>x0=solve("cos(x)-x",4);  ...
>  plot2d(x0,x0,>points,>add,title="Intersection Demo");  ...
>  label("cos(x) = x",x0,x0,pos="cl",offset=20):
```




```
>x0=solve("cos(x)-x",2);  ...
>  plot2d(x0,x0,>points,>add,title="Intersection Demo");  ...
>  label("cos(x) = x",x0,x0,pos="cl",offset=20):
```



Berapapun angka yang digunakan untuk tebakan awal tetap bisa digunakan dan solusi persamaan tetap sama. Pengisian angka digunakan agar EMT bisa mengenali numerik tebakan awalnya.

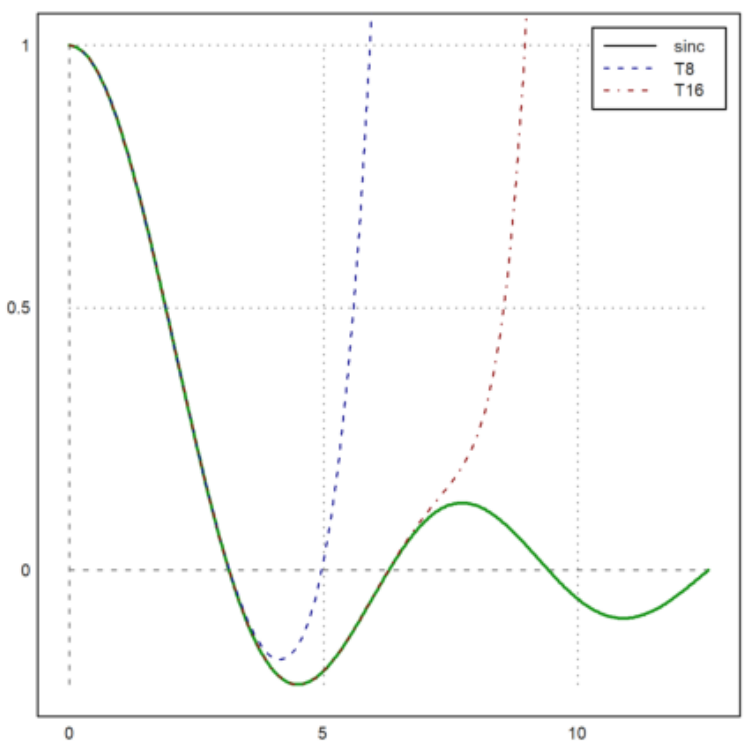
Dalam demo berikut, kita menggambar fungsi sinc(x) = sin(x)/x serta ekspansi Taylor orde ke-8 dan ke-16.

Ekspansi ini dihitung menggunakan Maxima melalui ekspresi simbolik.
Plot ini dibuat dengan perintah multi-baris yang berisi tiga kali pemanggilan plot2d().
emanggilan kedua dan ketiga menggunakan flag >add, yang membuat plot mengikuti rentang (range) dari plot sebelumnya.
Kita juga menambahkan kotak label untuk menjelaskan fungsi-fungsi tersebut.

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

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

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

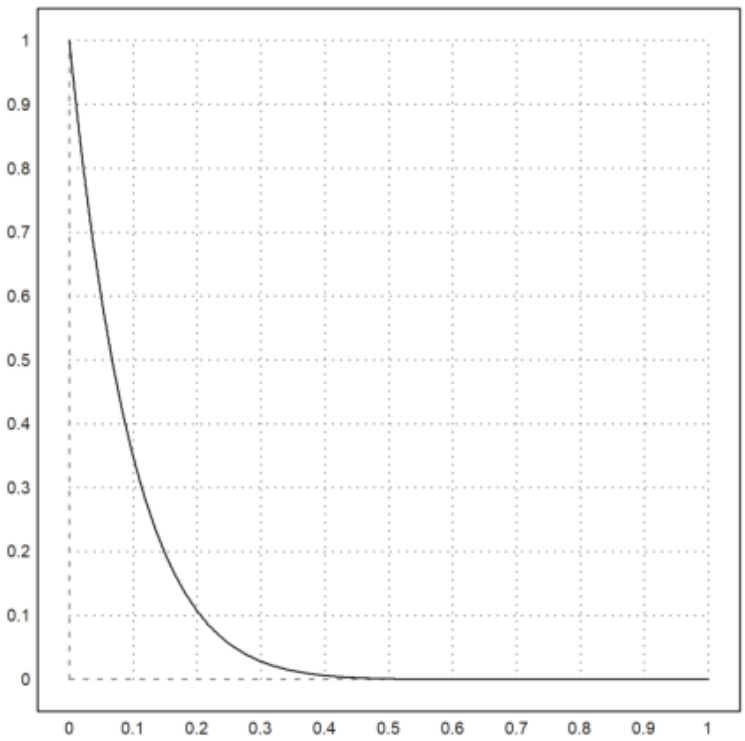


Kurva tersebut menggambarkan kurva fungsi asli sinc(x)=sin(x)/x dengan warna hijau. Kemudian Menggambar kurva hasil deret taylor hingga orde/turunan 8 dengan warna biru dan style garis putus-putus serta menggambar deter taylor hingga orde 16 dengan warna merah dan style garis putus-titik. Deret Taylor digunakan untuk mendekati suatu fungsi dengan polinomial tak hingga di sekitar titik (biasanya x=0). Kurva menunjukkan bahwa semakin besar ordenya semakin mendekati kurva asli.
Labelbok digunakan untuk memberi keterangan garis yang ada.

Dalam contoh berikut, kita menghasilkan Polinomial Bernstein.

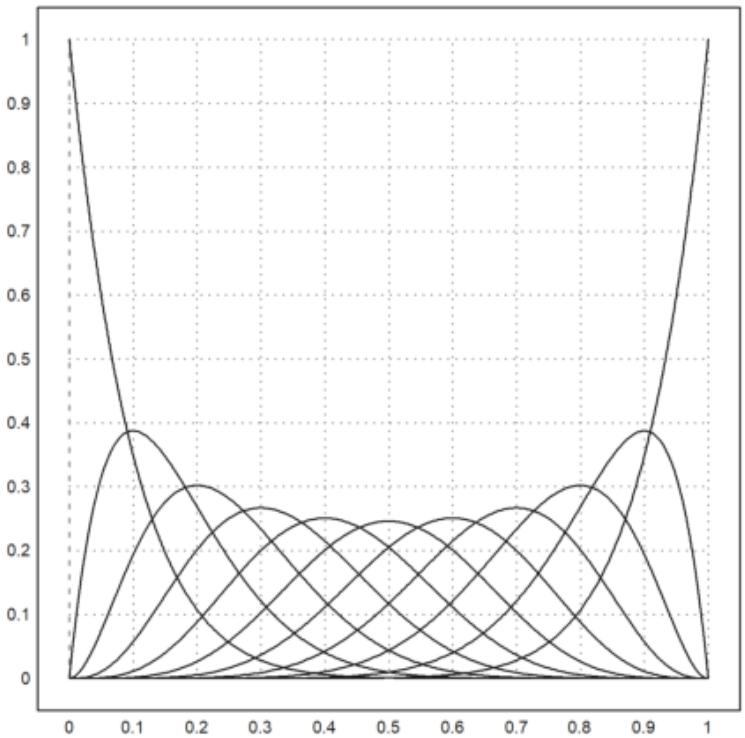
$$B_i(x) = \binom{n}{i} x^i (1-x)^{n-i}$$

```
>plot2d("(1-x)^10",0,1): // menggambar fungsi pertama
```



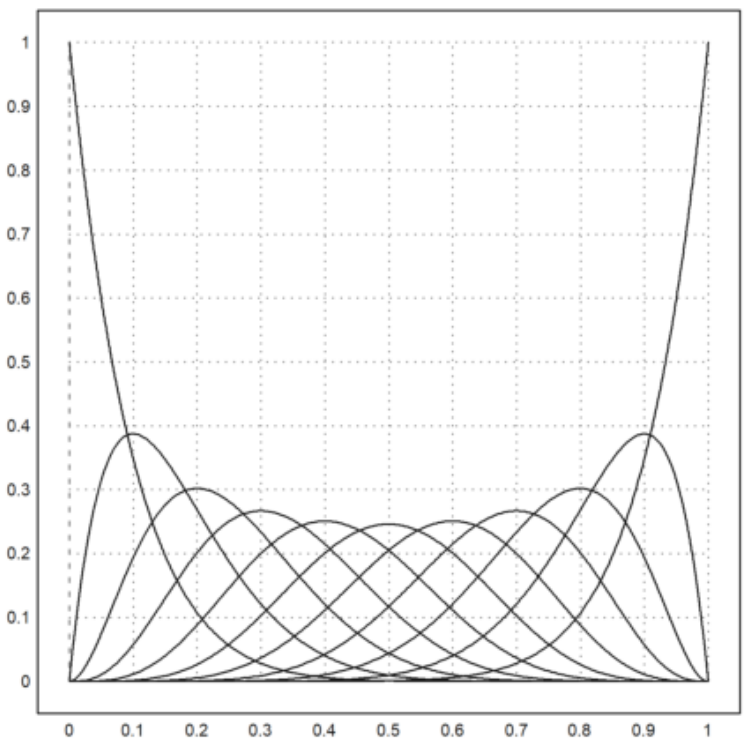
Kurva tersebut menunjukkan fungsi dasar (1-x)^10 dan dianggap kasus khusus dari distribusi binomial ketika i=0.

```
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end:
```



Kurva tersebut menunjukkan distribusi binomial (n=10) sebagai fungsi dari probabilitas sukses x. Tinggi ekstrem terletak di x=0 dan x=1. Kurva tersebut juga menunjukkan probabilitas jumlah sukses menengah saat x=0.5.

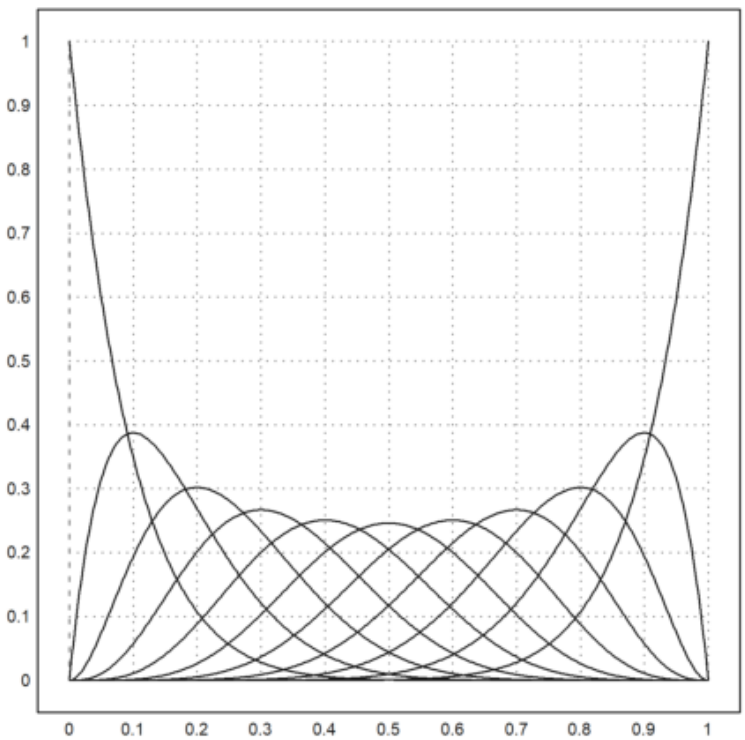
```
>insimg
```



Metode kedua menggunakan sepasang matriks nilai x dan matriks nilai y yang memiliki ukuran sama.

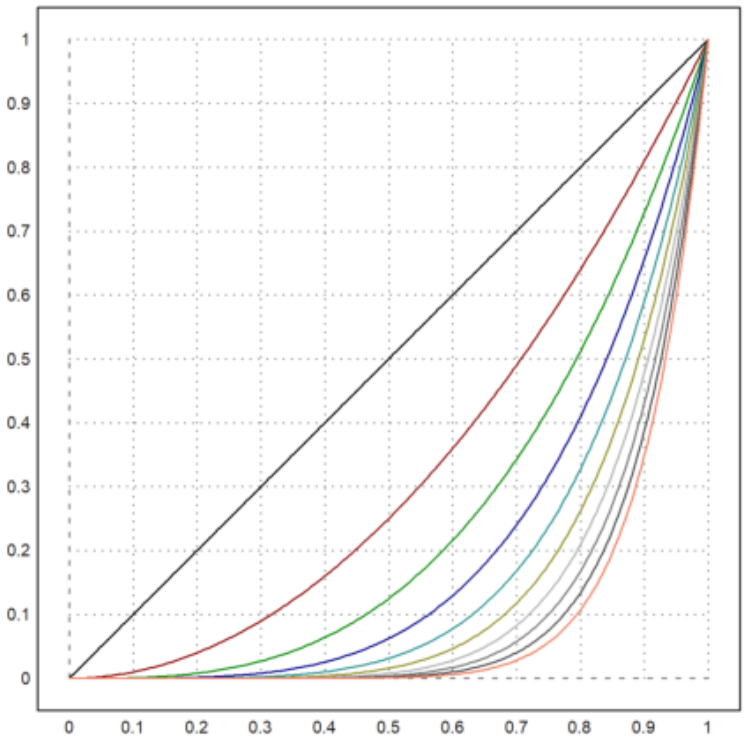
Kita menghasilkan sebuah matriks nilai dengan satu polinomial Bernstein di setiap baris. Untuk ini, kita cukup menggunakan vektor kolom dari i. Lihatlah bagian pengantar tentang bahasa matriks untuk mempelajari lebih banyak detail.

```
>x=linspace(0,1,500); // membuat vektor x dari 0 sampai 1 dg 500 titik
>n=10; k=(0:n)'; // n adl vektor baris, k adl vektor kolom
>y=bin(n,k)*x^k*(1-x)^(n-k); // y is a matrix then
>plot2d(x,y):
```



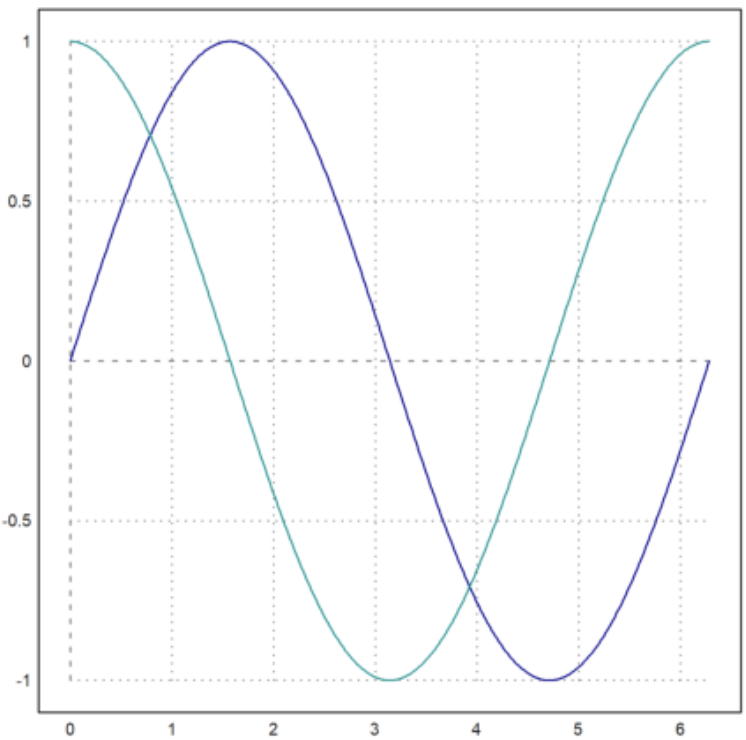
Perlu dicatat bahwa parameter color dapat berupa sebuah vektor. Dengan demikian, setiap warna akan digunakan untuk setiap baris dari matriks.

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

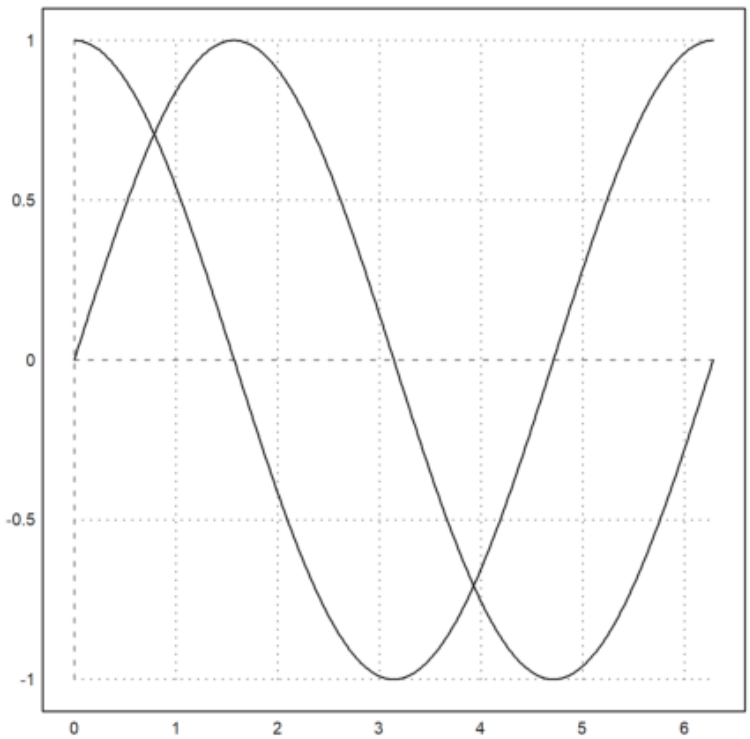


Metode lain adalah dengan menggunakan sebuah vektor berisi ekspresi (string). Dengan cara ini, Anda dapat menggunakan array warna, array gaya, dan array ketebalan dengan panjang yang sama.

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



```
>plot2d(["sin(x)","cos(x)"],0,2pi): // Menggambar ekspresi vektor
```



Kita dapat memperoleh vektor seperti itu dari Maxima dengan menggunakan makelist() dan mxm2str()

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

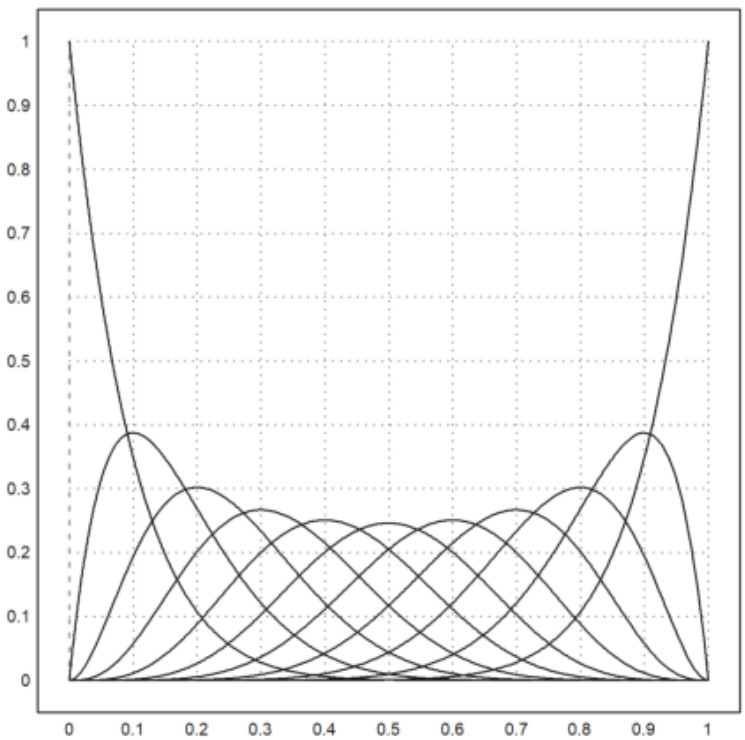
$$\left[(1-x)^{10},\ 10\,(1-x)^{9}\,x,\ 45\,(1-x)^{8}\,x^{2},\ 120\,(1-x)^{7}\,x^{3},\ 210\,(1-x)^{6}\,x^{4},\ 252\,(1-x)^{5}\,x^{5},\ 210\,(1-x)^{4}\,x^{6},\ 120\,(1-x)^{3}\,x^{7},\ 45\,(1-x)^{2}\,x^{8},\ 10\,(1-x)\,x^{9},\ x^{10}\right]$$

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

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

```
10*(1-x)*x^9
x^10
```

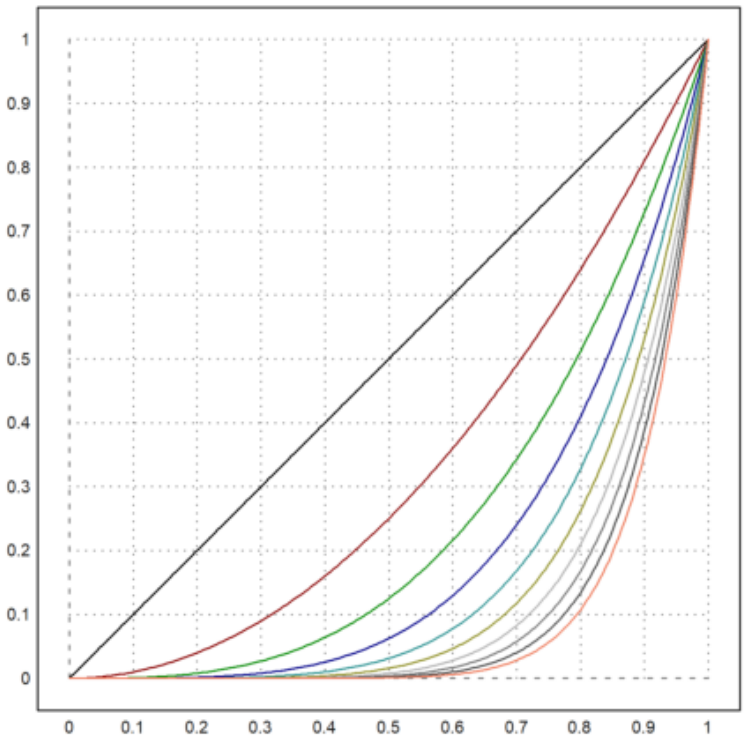
```
>plot2d(mxm2str(v),0,1): // menggambar fungsi
```



Alternatif lain adalah menggunakan bahasa matriks dari Euler.

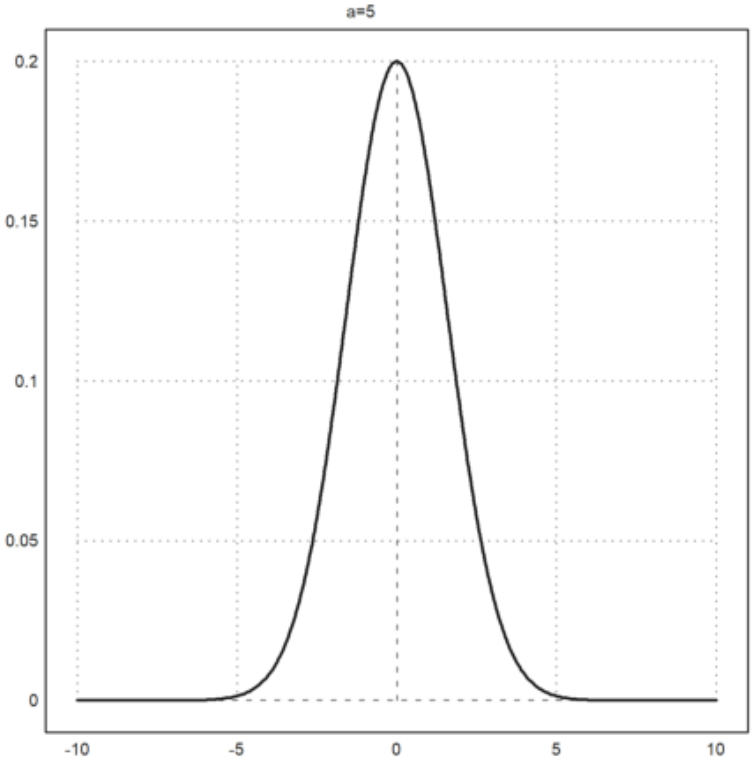
Jika suatu ekspresi menghasilkan sebuah matriks fungsi, dengan satu fungsi pada setiap baris, maka semua fungsi tersebut akan dipetakan ke dalam satu plot.
Untuk itu, gunakan vektor parameter dalam bentuk vektor kolom. Jika ditambahkan array warna, maka array tersebut akan digunakan untuk setiap baris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

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



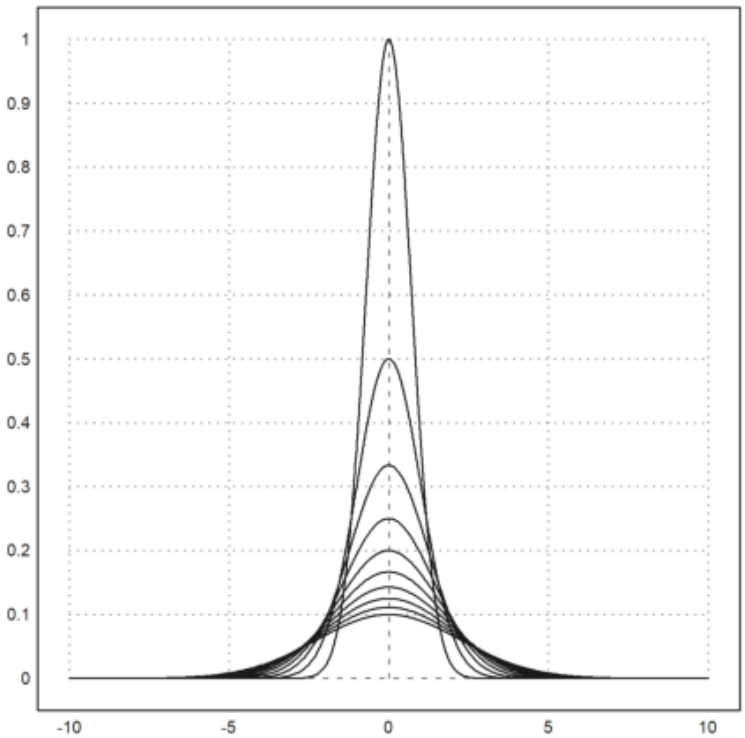
Ekspresi dan fungsi satu baris dapat melihat variabel global.
Jika Anda tidak dapat menggunakan variabel global, Anda perlu menggunakan fungsi dengan parameter tambahan, dan meneruskan parameter tersebut sebagai parameter titik koma. Perhatikan untuk meletakkan semua parameter yang diberikan di bagian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

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



Sebagai alternatif, gunakan sebuah koleksi berisi nama fungsi dan semua parameter tambahan. Daftar khusus ini disebut call collection, dan merupakan cara yang lebih disarankan untuk meneruskan argumen ke sebuah fungsi yang dirinya sendiri diteruskan sebagai argumen ke fungsi lain.
Pada contoh berikut, kita menggunakan sebuah loop untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman untuk loops).

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



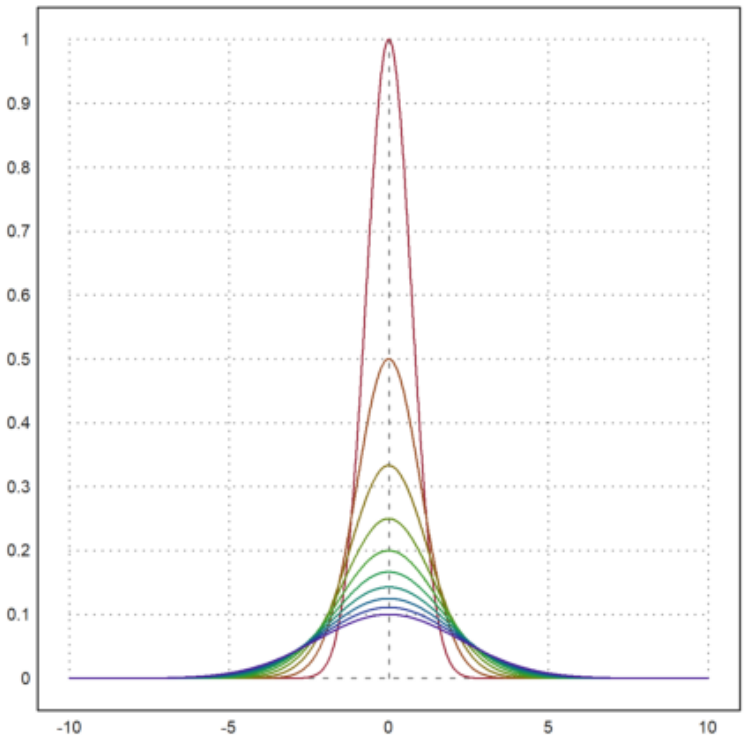
Kurva tersebut menunjukkan kurva fungsi yang sudah didefinisikan sebelumnya. Fungsi tersebut adalah

$$f(x,a) = e^{-a*x^2}/a$$

Dengan Interval x=[-10,10] dan parameter a dari 1 sampai 10.

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

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



Perintah getspectral() untuk mengjasilkan warna spektral untuk nilai x dengan rentang 0 = x = 1. Skema warna ini Dari biru (x = 0) ke merah (x = 1). Jadi, fungsi ini digunakan untuk mengonversi nilai numerik menjadi warna spektral dalam visualisasi data.

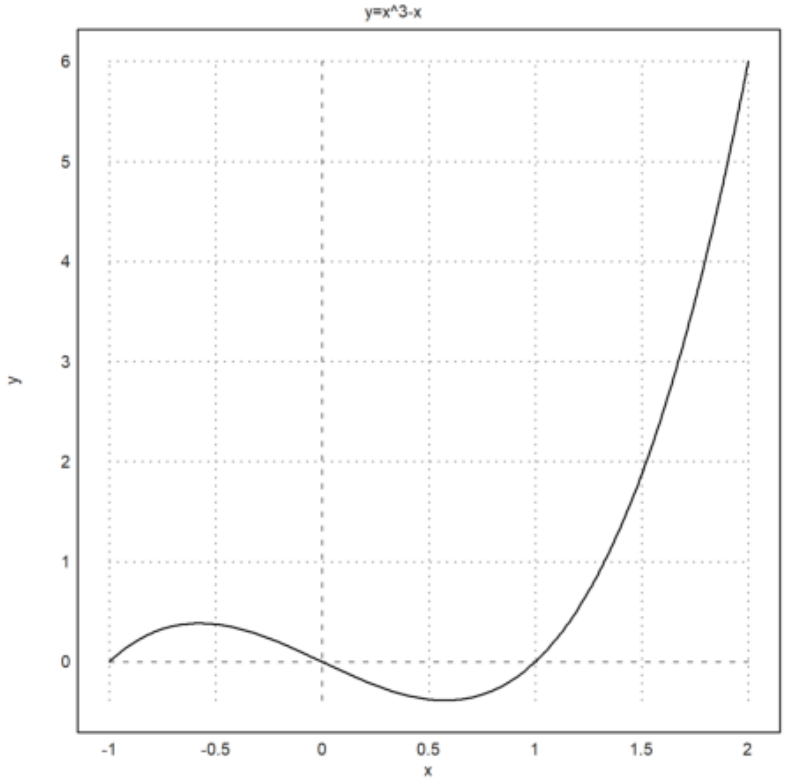
Label Text

Dekorasi sederhana dapat berupa:

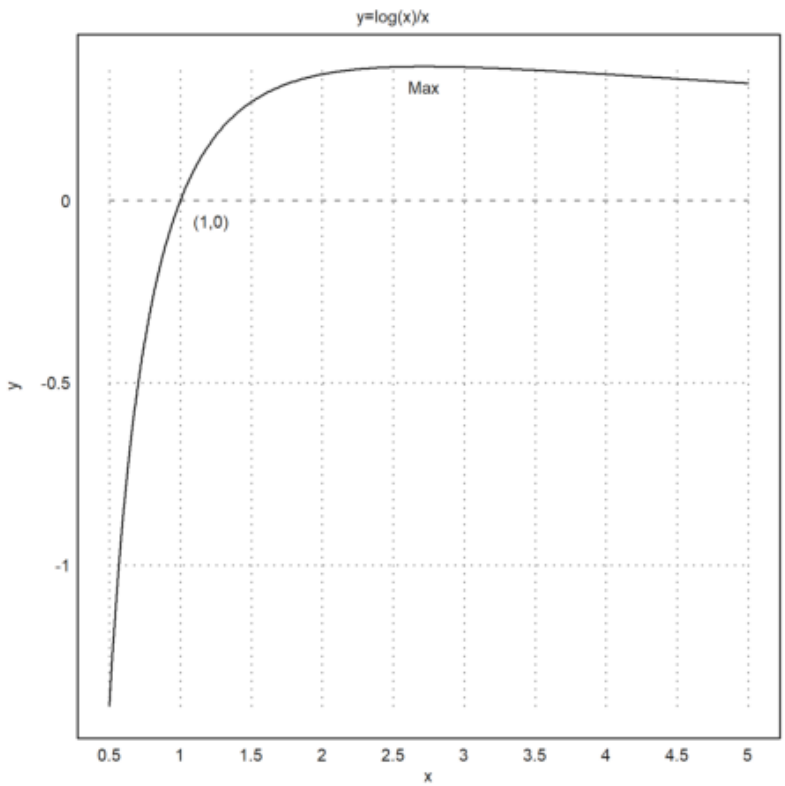
- Judul dengan title="..."
- Label sumbu x dan y dengan xl="...", yl="..."
- Label teks tambahan dengan label("...", x, y)

Perintah label akan menampilkan teks pada plot saat ini di koordinat plot (x, y).
erintah ini juga dapat menerima argumen posisi.

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

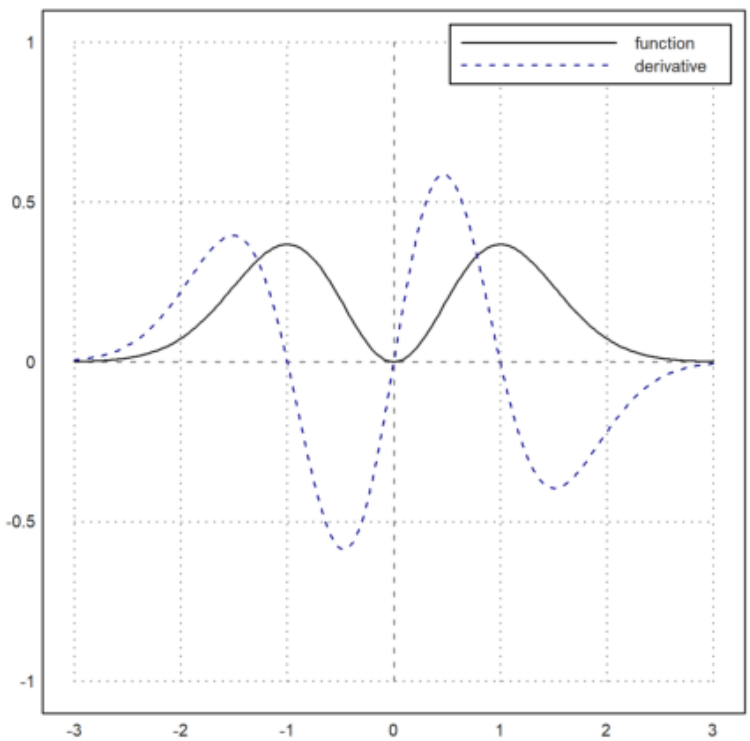



```
>expr := "log(x)/x"; ...
>  plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
>  label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```



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

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



Kurva tersebut menunjukkan kurva fungsi

$$f(x) = x^2 e^{-x^2}$$

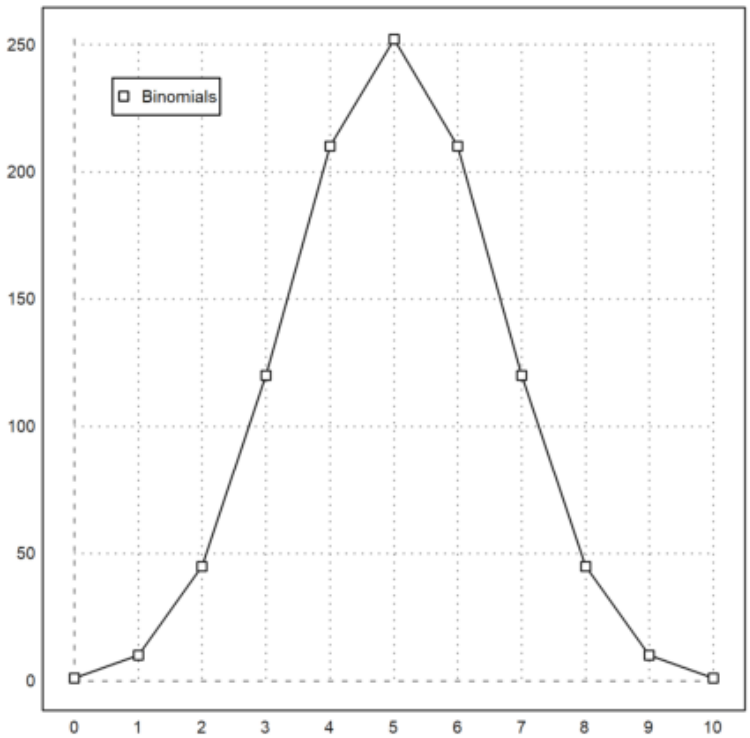
serta turunan pertamanya. Fungsi pertama digambar dalam kurva hitam dan fungsi turunan digambar dalam kurva biru dan style garis putus-putus.
Labelbox ditambahkan untuk memberi keterangan kurva fungsi pertama dan fungsi turunan dengan lebar kotak 0.4

Kotak legend secara default terpasang di kanan atas, tetapi dengan >left kotak akan terpasang di kiri atas. Kamu bisa memindahkannya ke posisi mana pun. Posisi anchor adalah sudut kanan atas kotak Dan angka-angka tersebut adalah pecahan dari ukuran jendela grafik. Lebar kotak diatur secara otomatis.

Untuk plot titik, kotak legend juga bisa digunakan. Tambahkan parameter >points, atau sebuah vektor flag, satu untuk setiap label.

Dalam contoh berikut, hanya ada satu fungsi, jadi kita bisa menggunakan string tunggal bukan vektor string. Warna teks pada contoh ini diatur menjadi hitam.

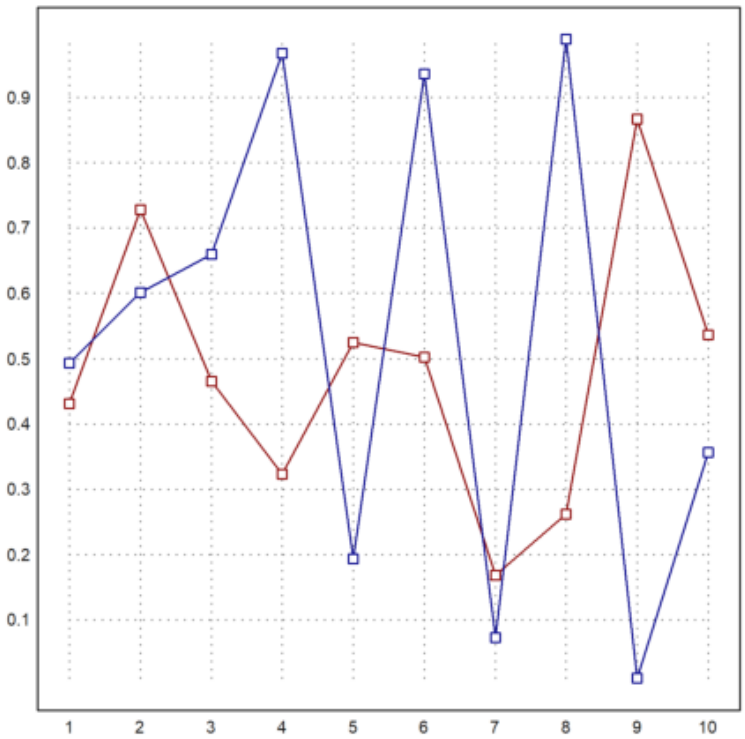
```
>n=10; plot2d(0:n,bin(n,0:n),>addpoints); ...
>labelbox("Binomials",styles="[]",>points,x=0.1,y=0.1, ...
>tcolor=black,>left):
```



Grafik tersebut menggambarkan plot binomial untuk n=10 sebagai titik serta menambahkan labelbox dengan perintah labelbox() yang terletak di kiri atas. Gaya plot berupa titik dan ditunjukkan pada labelbox. Perintah x dan y digunakan untuk mengatur posisi labelbox sebagai pecahan dari jendela grafik, 0.1 = 10% dari lebar/tinggi.

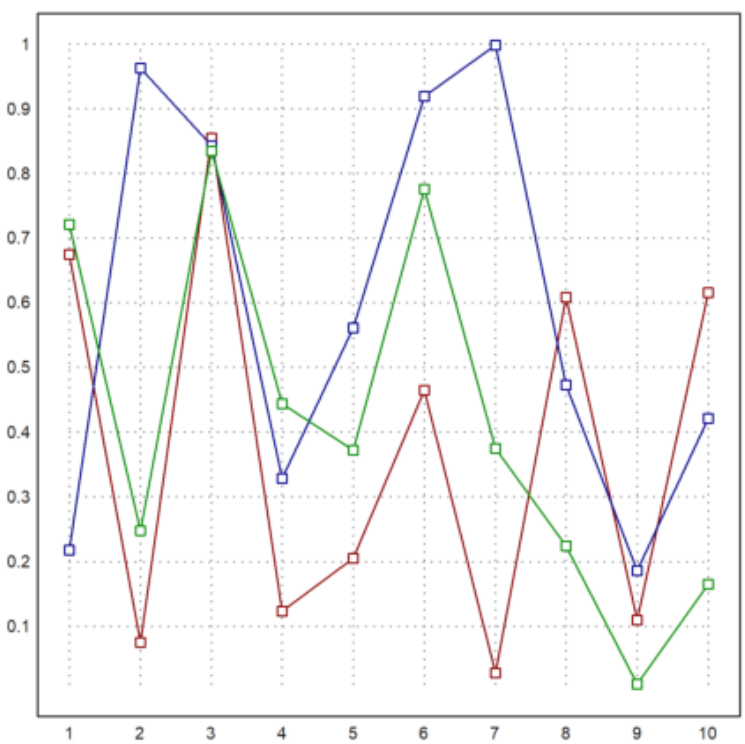
Gaya plot ini juga tersedia di statplot(). Seperti pada plot2d(), warna dapat diatur untuk setiap baris plot. Terdapat juga plot khusus lain untuk tujuan statistik (lihat tutorial tentang statistik).

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



Grafik tersebut menunjukkan plot dua set data acak dengan vektor x dari 1 sampai 10.

```
>statplot(1:10,random(3,10),color=[red,blue,green]):
```

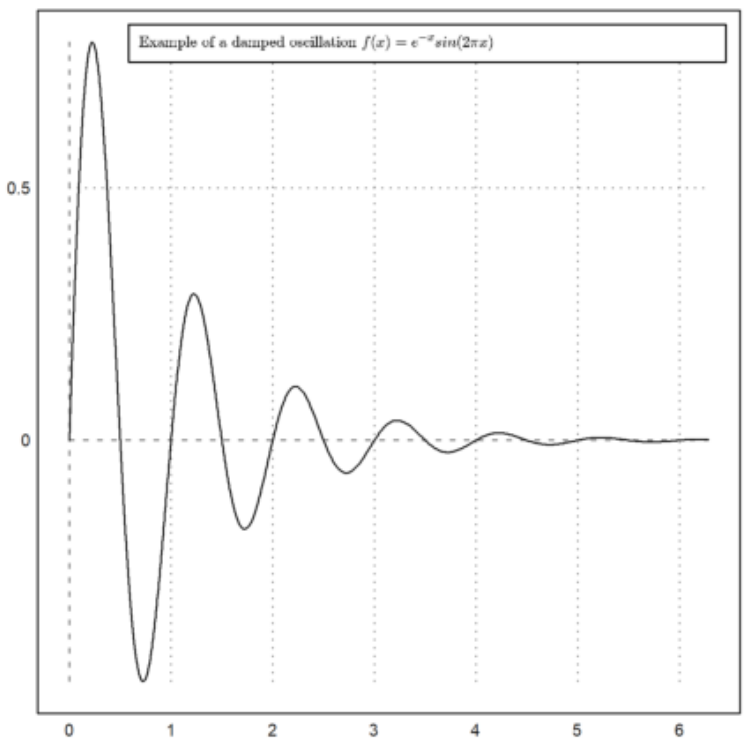


Grafik tersebut menunjukkan plot tiga set data acak dengan vektor x dari 1 sampai 10.

Fitur serupa adalah fungsi textbox().

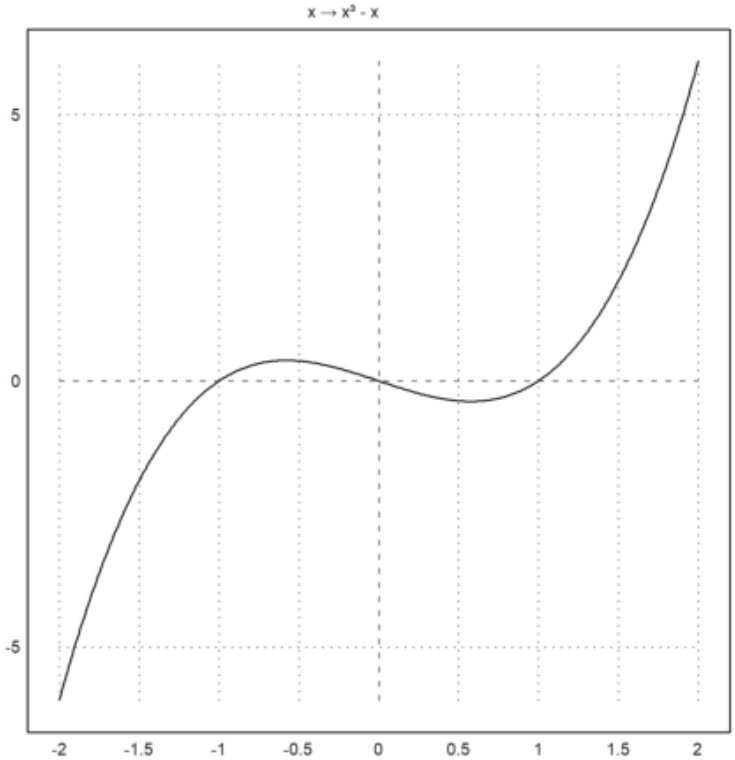
Lebarnya secara default adalah lebar maksimum dari baris-baris teks. Namun, lebar ini juga bisa diatur sendiri oleh pengguna.

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



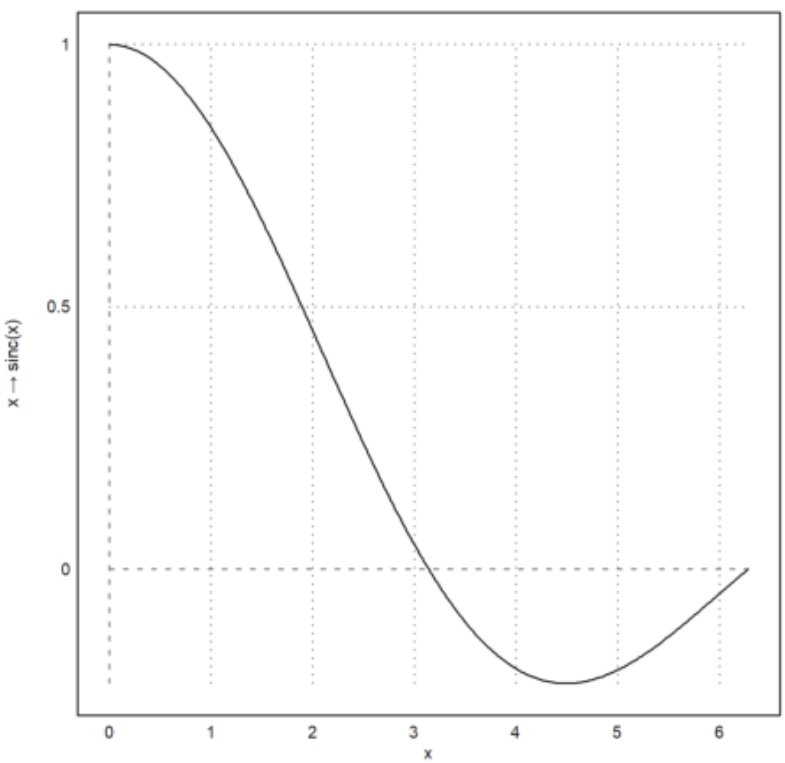
Label teks, judul, kotak label, maupun teks lainnya dapat dituliskan dengan string Unicode (lihat dokumentasi EMT untuk detail penggunaannya).
ara default adalah lebar maksimum dari baris-baris teks. Namun, lebar ini juga bisa diatur sendiri oleh pengguna.

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



Selain itu, label sumbu x maupun y dapat dibuat vertikal, begitu juga orientasi sumbunya.

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

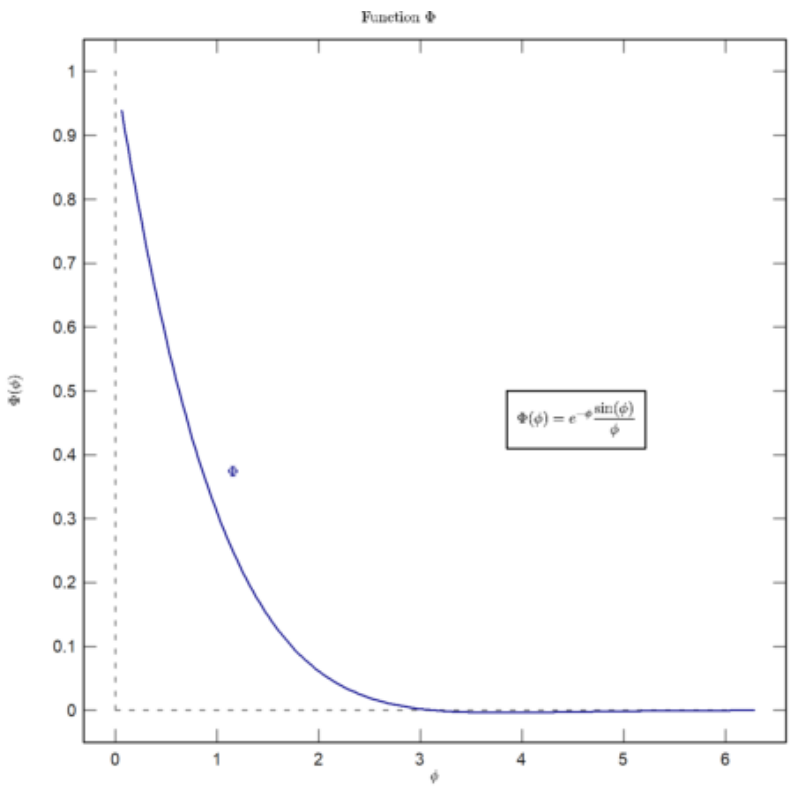


Anda juga bisa memplot formula LaTeX jika Anda telah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Path ke file biner “latex” dan “dvipng” harus ada di sistem path, atau Anda perlu mengatur LaTeX di menu opsi.

Perlu dicatat, parsing LaTeX itu lambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, sebaiknya panggil latex() sekali sebelum loop dan gunakan hasilnya (sebuah gambar dalam matriks RGB).

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

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



Kurva tersebut menunjukkan kurva fungsi

$$\Phi(\phi) = e^{-\phi}\frac{\sin(\phi)}{\phi}$$

dengan rentang x=[0,2phi] dan y=[0,1] dan warna kurva biru. Kurva tersebut diberi judul dengan style latex. Serta label sumbu-x dan sumbu-y. Perbedaan menulis lambang

$$\Phi$$

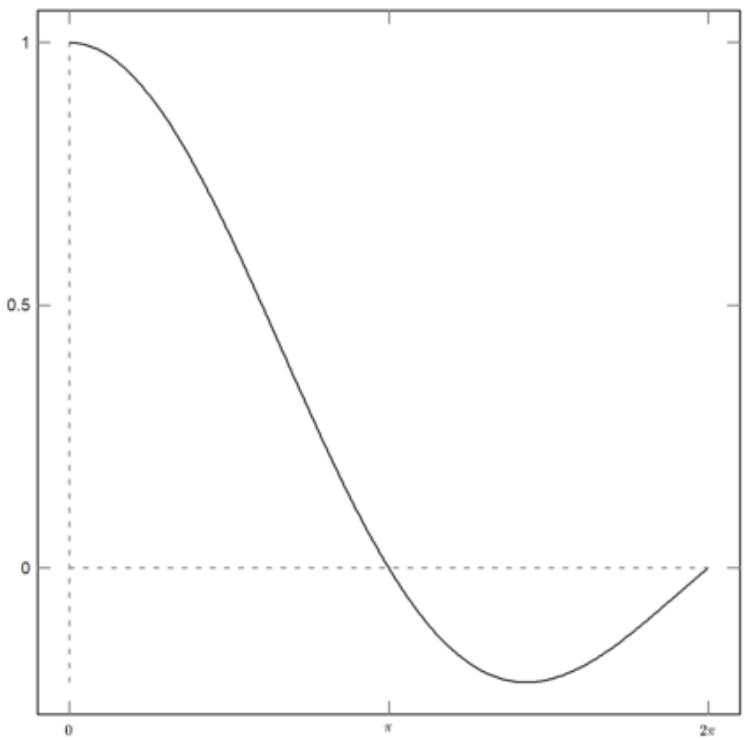
$$\phi$$

adalah huruf kapital untuk huruf P

Seringkali, kita ingin jarak tidak konformal dan label teks pada sumbu x. Kita bisa menggunakan xaxis() dan yaxis(), seperti yang akan ditunjukkan nanti.

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

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



Perintah xtick() digunakan untuk memberikan label pada setiap interval sumbu x yang dibagi dalam 3 string. >ticks perlu ditambahkan pada plot dasar sebagai Placeholder untuk pengaturan ticks khusus yang akan diatur kemudian. >ticks juga ditambahkan pada xgrid() karena pengaturan ticks custom akan diisi oleh xticks.

Jelas, fungsi juga dapat digunakan.

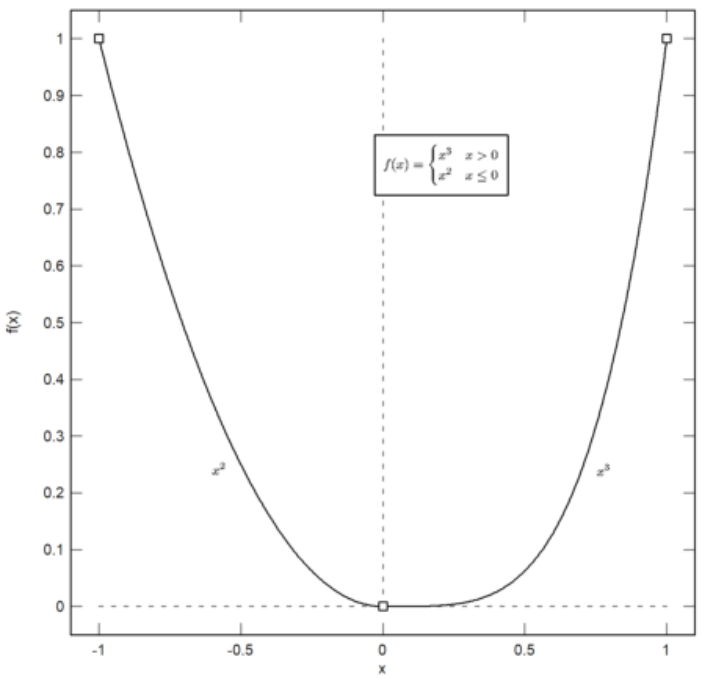
```
>function map f(x) ...
```

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

Parameter "map" membantu fungsi dapat digunakan untuk vektor. Untuk plot saja, sebenarnya tidak wajib. Tetapi untuk menunjukkan bahwa vektorisasi berguna, kita menambahkan beberapa titik kunci pada plot di x = -1, x = 0, dan x = 1.

Dalam plot berikut, kita juga memasukkan beberapa kode LaTeX. LaTeX digunakan untuk dua label dan sebuah kotak teks. Tentu saja, Anda hanya bisa menggunakan LaTeX jika LaTeX telah terinstal dengan benar di sistem Anda.

```
>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

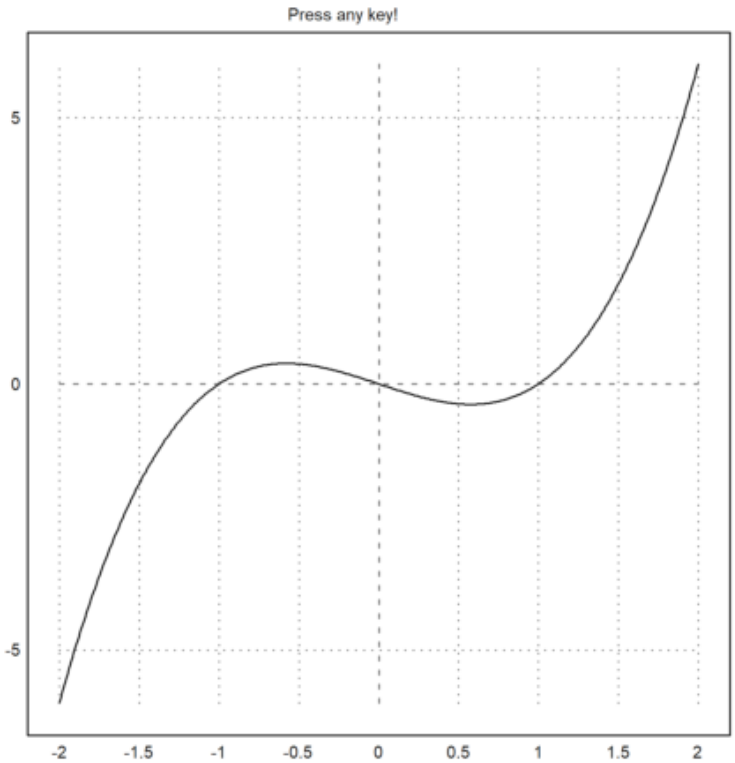
Saat memplot sebuah fungsi atau ekspresi, parameter >user memungkinkan pengguna untuk zoom dan menggeser plot dengan tombol panah atau mouse. Pengguna dapat:

- zoom dengan tombol + atau -
- menggeser plot dengan tombol panah (cursor keys)
- memilih jendela plot dengan mouse
- mereset tampilan dengan tombol spasi
- keluar dengan tombol Enter (return)

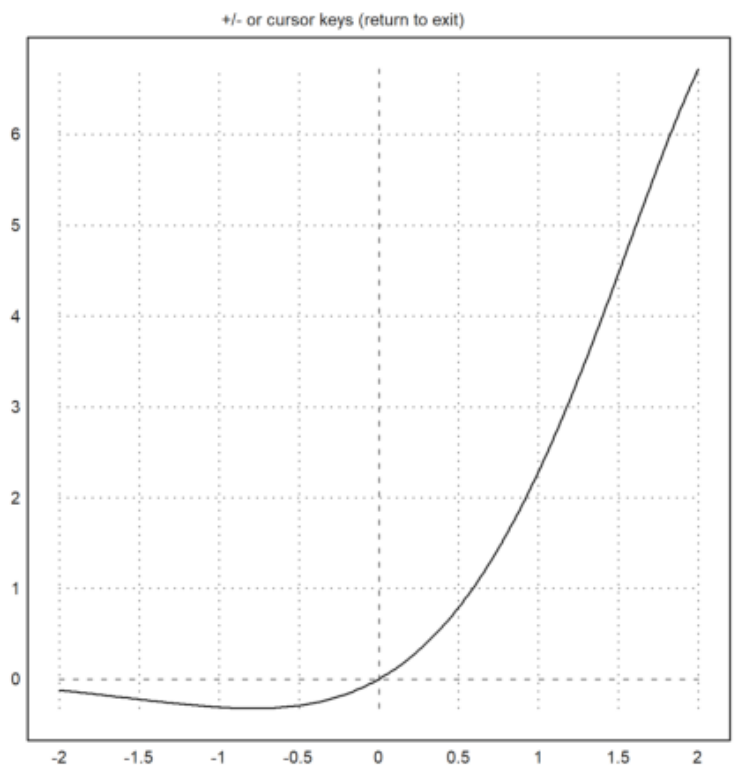
Tombol spasi akan mengembalikan plot ke jendela tampilan asli.

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

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




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



Berikut ini menunjukkan cara interaksi pengguna tingkat lanjut (lihat tutorial tentang pemrograman untuk detailnya).

Fungsi bawaan mousedrag() menunggu event mouse atau keyboard. Ini melaporkan mouse ke bawah, mouse dipindahkan atau mouse ke atas, dan penekanan tombol. Fungsi dragpoints() memanfaatkan ini, dan memungkinkan pengguna menyeret titik mana pun dalam plot.

Kita membutuhkan fungsi plot terlebih dahulu. Sebagai contoh, kita interpolasi dalam 5 titik dengan polinomial. Fungsi harus diplot ke area plot tetap.

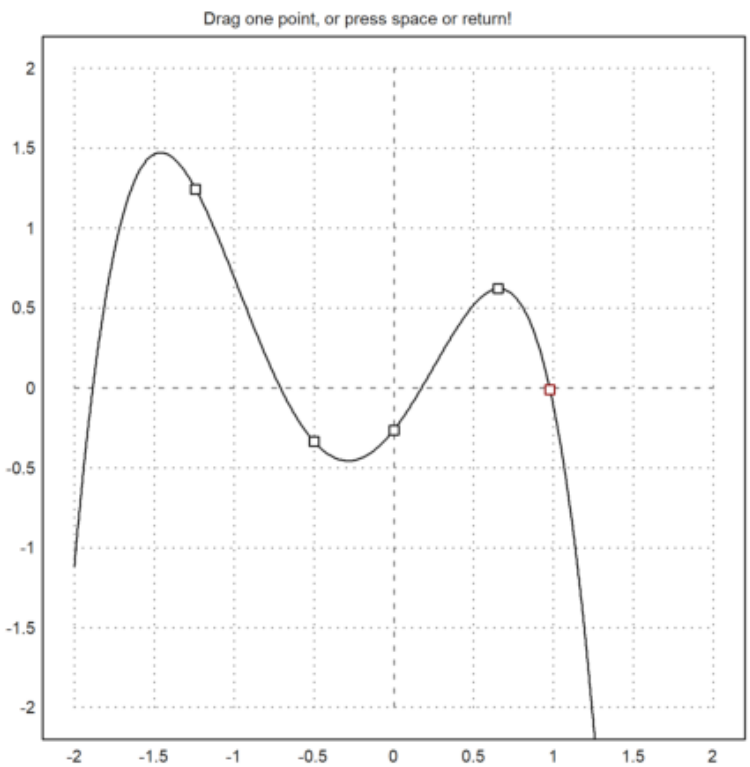
```
>function plotf(xp,yp,select) ...
```

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

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

Sekarang kita menghasilkan beberapa nilai acak, dan membiarkan pengguna menyeret poin.

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

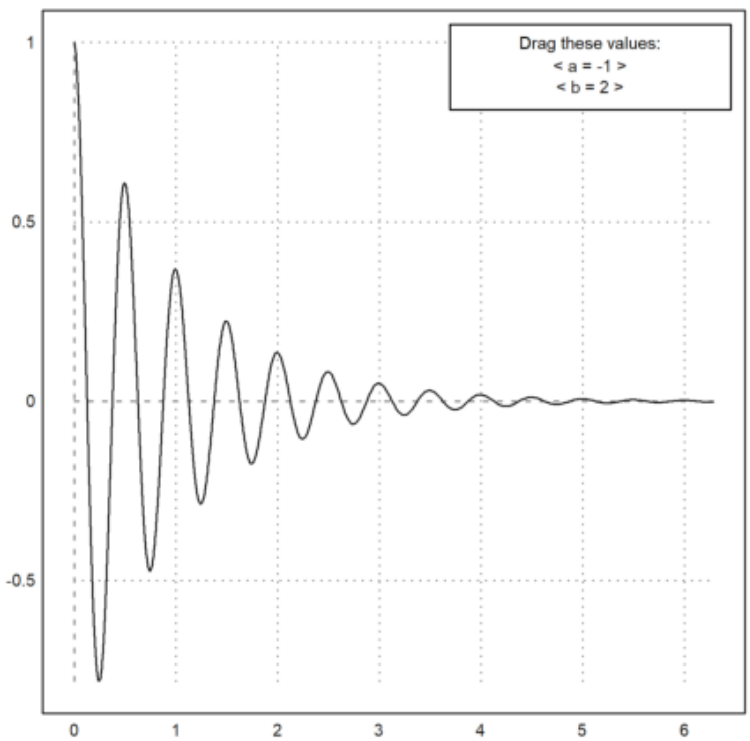


Ada juga fungsi, yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.
Pertama kita membutuhkan fungsi plot.

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

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

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



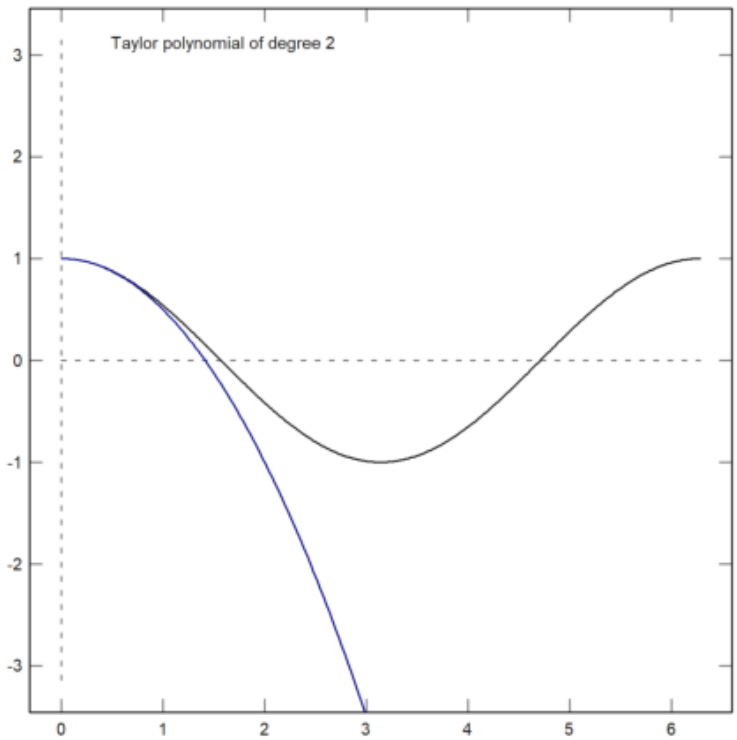
Dimungkinkan untuk membatasi nilai yang diseret ke bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor derajat n ke fungsi kosinus. Fungsi dragvalues() menyediakan ini.

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

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

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

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



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

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

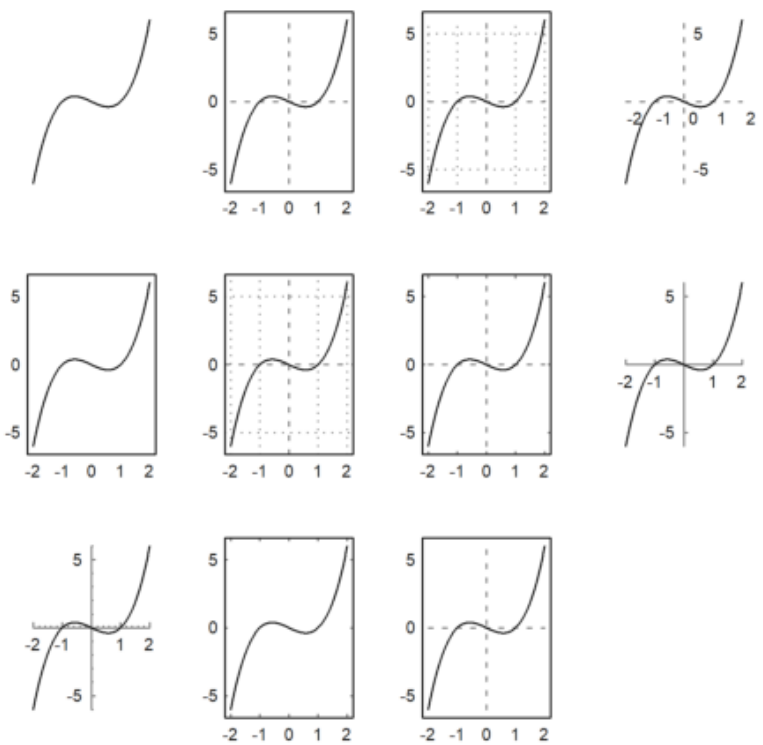
```
    plot2d(none,r=1,title="Drag with the mouse, or press any key!");
    start=0;
    repeat
      {flag,m,time}=mousedrag();
      if flag==0 then return; endif;
      if flag==2 then
        hold on; mark(m[1],m[2]); hold off;
      endif;
    end
  endfunction
```

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

Gaya Plot 2D

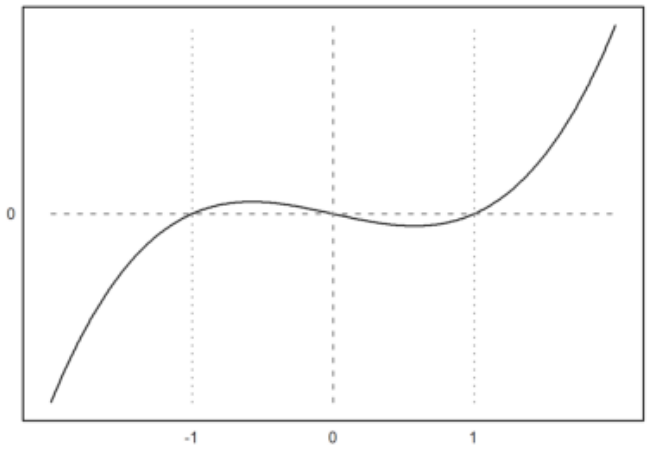
Secara default, EMT menghitung tick sumbu otomatis dan menambahkan label ke setiap tick. Ini dapat diubah dengan parameter grid. Gaya default sumbu dan label dapat dimodifikasi. Selain itu, label dan judul dapat ditambahkan secara manual. Untuk mengatur ulang ke gaya default, gunakan reset().

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



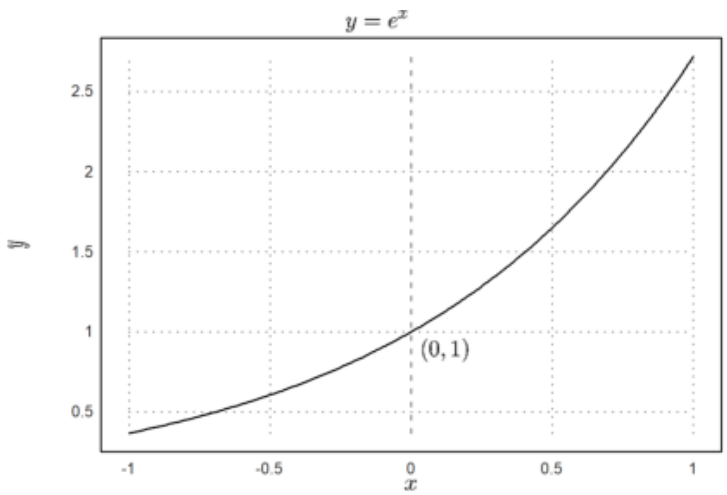
Parameter <frame mematikan frame, dan framecolor=blue mengatur frame ke warna biru.
Jika Anda ingin centang sendiri, Anda dapat menggunakan style=0, dan menambahkan semuanya nanti.

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



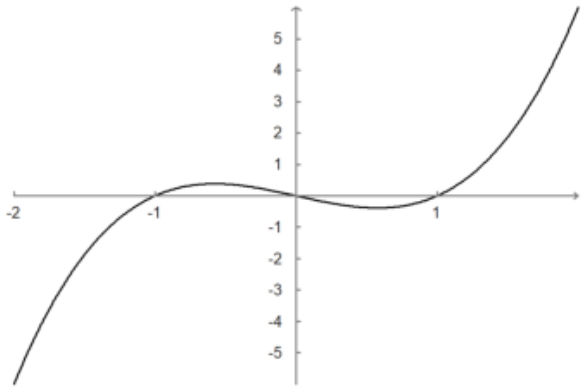
Untuk judul plot dan label sumbu, lihat contoh berikut.

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



Sumbu dapat digambar secara terpisah dengan xaxis() dan yaxis().

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

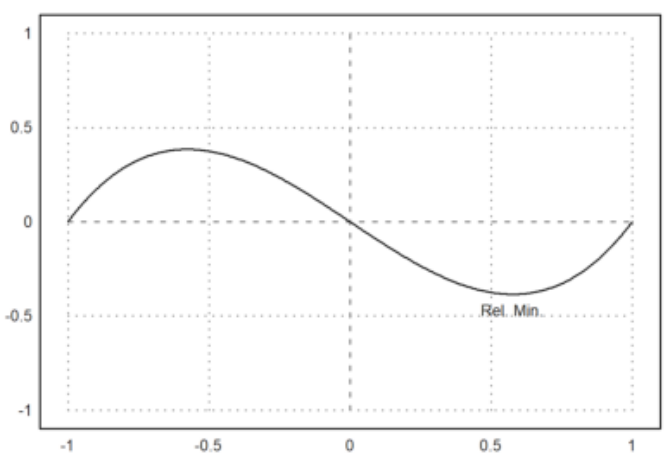


Teks pada plot dapat diatur dengan label(). Dalam contoh berikut, "lc" berarti tengah bawah. Ini mengatur posisi label relatif terhadap koordinat plot

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

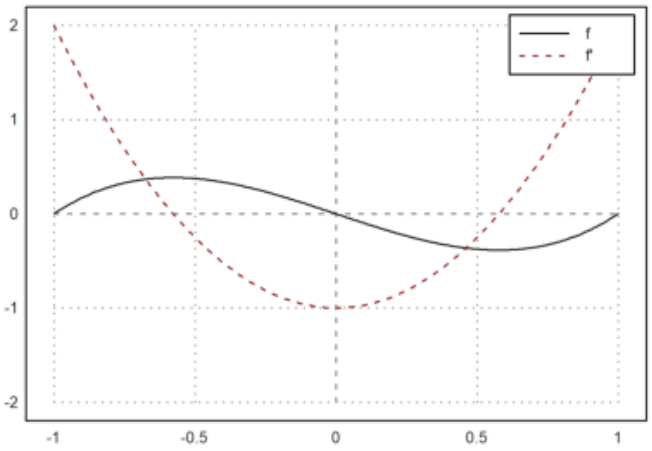
$$x^3 - x$$

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

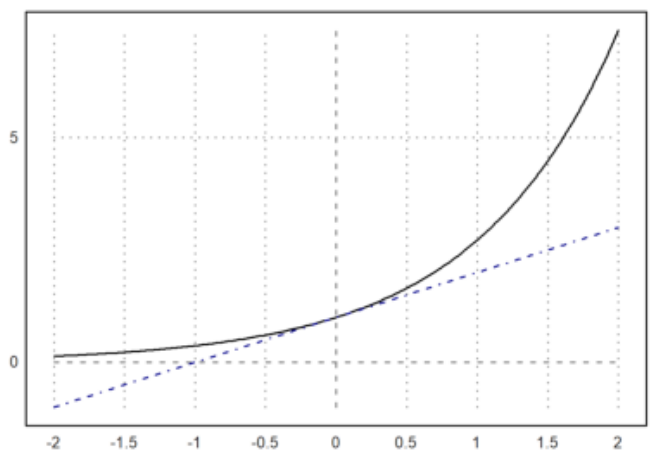


Ada juga kotak teks.

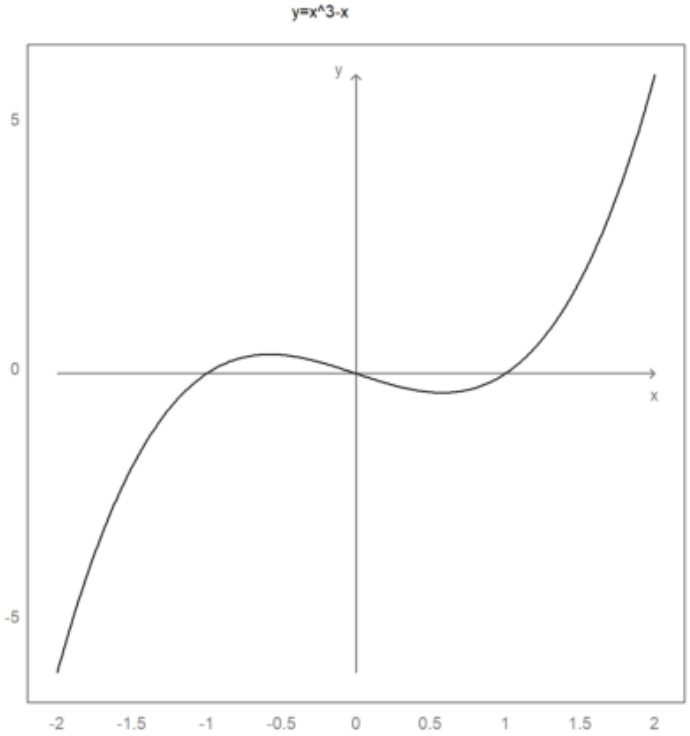
```
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // derivative
>labelbox(["f","f'"],["-","--"],[black,red]): // label box
```




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

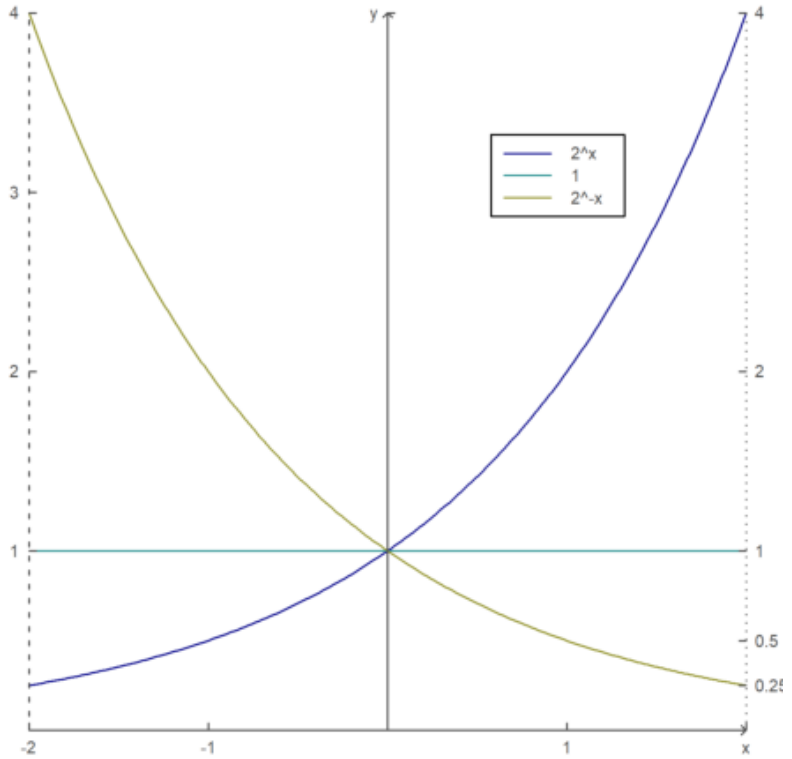



```
>gridstyle("->",color=gray,textcolor=gray,framecolor=gray);  ...
> plot2d("x^3-x",grid=1);   ...
> settitle("y=x^3-x",color=black); ...
> label("x",2,0,pos="bc",color=gray);  ...
> label("y",0,6,pos="cl",color=gray); ...
> reset():
```



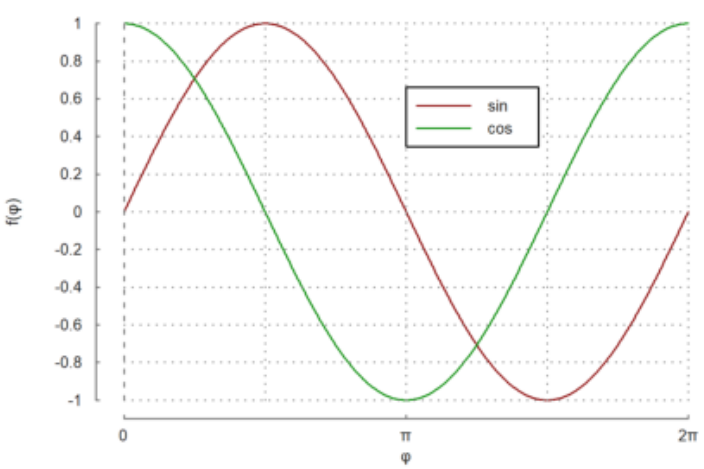
Untuk kontrol lebih, sumbu x dan sumbu y dapat dilakukan secara manual.
Perintah fullwindow() memperluas jendela plot karena kita tidak lagi membutuhkan tempat untuk label di luar jendela plot. Gunakan shrinkwindow() atau reset() untuk mengatur ulang ke default.

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



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

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



Plotting Data 2D

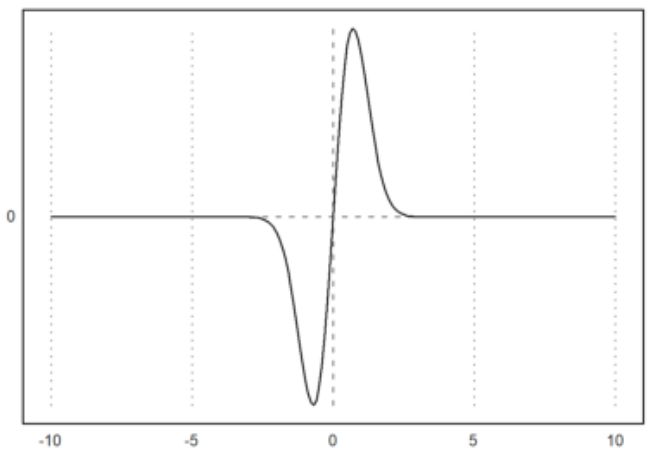
Jika x dan y adalah vektor data, data ini akan digunakan sebagai koordinat x dan y dari suatu kurva. Dalam hal ini, a, b, c, dan d, atau radius r dapat ditentukan, atau jendela plot akan menyesuaikan secara otomatis dengan data. Atau, >persegi dapat diatur untuk menjaga rasio aspek persegi.

Memplot ekspresi hanyalah singkatan untuk plot data. Untuk plot data, Anda memerlukan satu atau beberapa baris nilai x, dan satu atau beberapa baris nilai y. Dari rentang dan nilai-x, fungsi plot2d akan menghitung data yang akan diplot, secara default dengan evaluasi fungsi yang adaptif. Untuk plot titik gunakan ">titik", untuk garis campuran dan titik gunakan ">tambahan".

Tapi Anda bisa memasukkan data secara langsung.
- Gunakan vektor baris untuk x dan y untuk satu fungsi.
- Matriks untuk x dan y diplot baris demi baris.

Berikut adalah contoh dengan satu baris untuk x dan y.

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



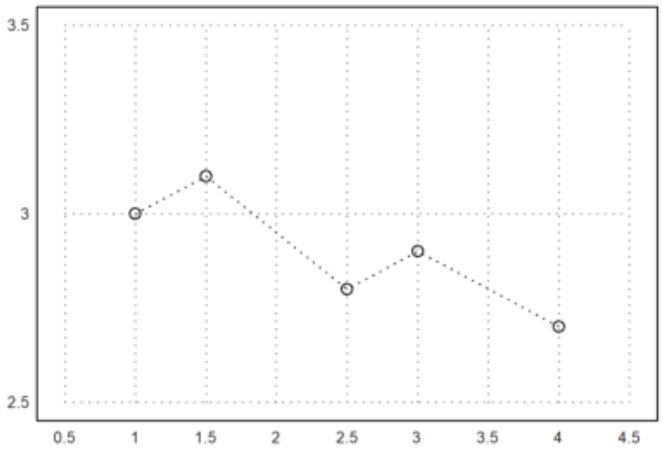
Data juga dapat diplot sebagai titik. Gunakan poin=true untuk ini. Plotnya bekerja seperti poligon, tetapi hanya menggambar sudut-sudutnya.

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

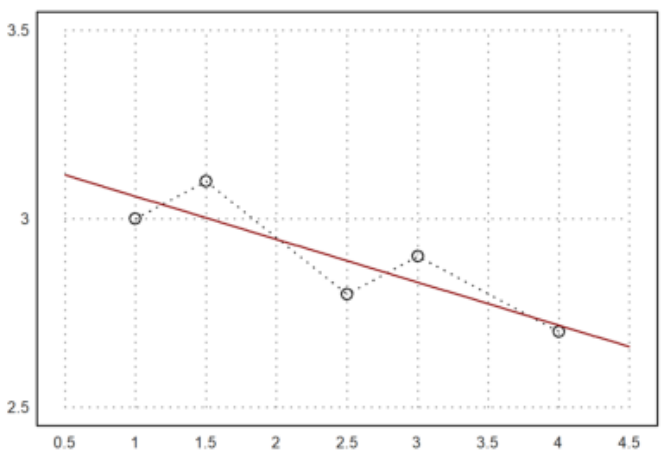
Untuk memplot set poin gunakan >points. Jika warna adalah vektor warna, setiap titik mendapat warna yang berbeda. Untuk matriks koordinat dan vektor kolom, warna berlaku untuk baris matriks.

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

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




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



Menggambar Daerah Yang Dibatasi Kurva

Plot data benar-benar poligon. Kita juga dapat memplot kurva atau kurva terisi.

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

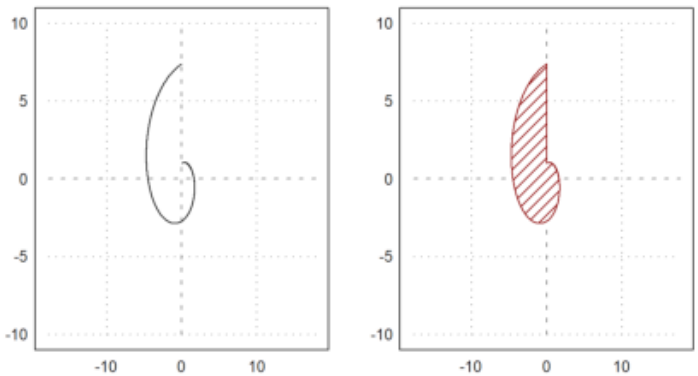
Warna isian ditentukan oleh argumen "fillcolor", dan pada <outline opsional mencegah menggambar batas untuk semua gaya kecuali yang default.

Parameter >addpoints menambahkan titik ke segmen garis untuk plot data.
ot2d akan menghitung data yang akan diplot, secara default dengan evaluasi fungsi yang adaptif. Untuk plot titik gunakan ">titik", untuk garis campuran dan titik gunakan ">tambahan".

Tapi Anda bisa memasukkan data secara langsung.
- Gunakan vektor baris untuk x dan y untuk satu fungsi.
- Matriks untuk x dan y diplot baris demi baris.

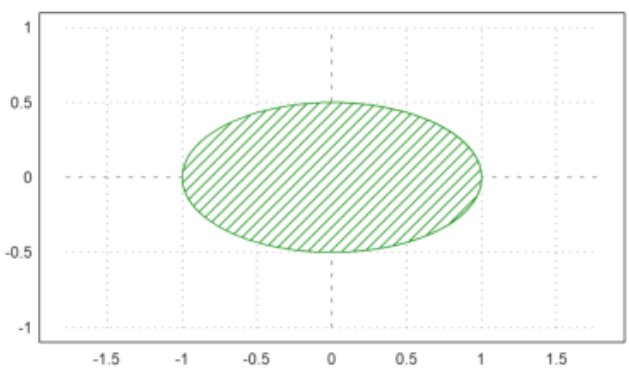
Berikut adalah contoh dengan satu baris untuk x dan y.

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

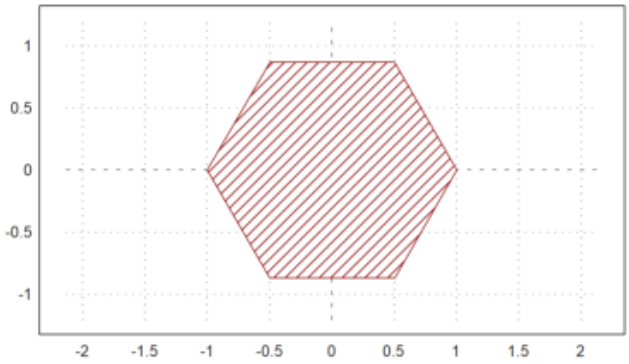


Dalam contoh berikut kami memplot elips terisi dan dua segi enam terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian berbeda.

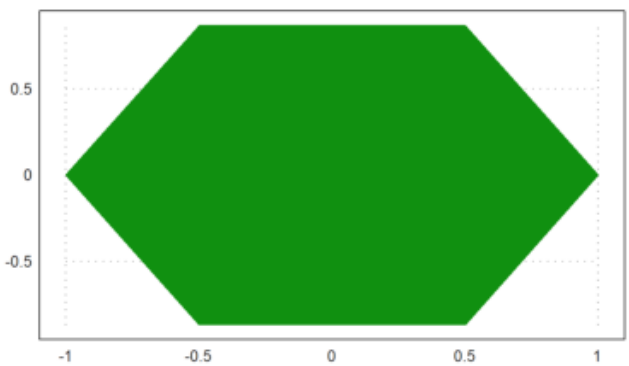
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```




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

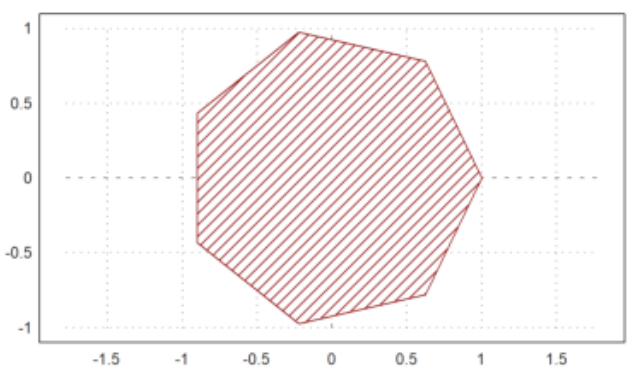



```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



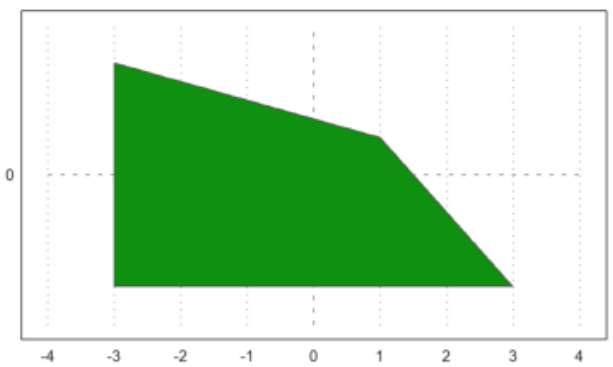
Contoh lainnya adalah segi empat, yang kita buat dengan 7 titik pada lingkaran satuan.

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



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

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

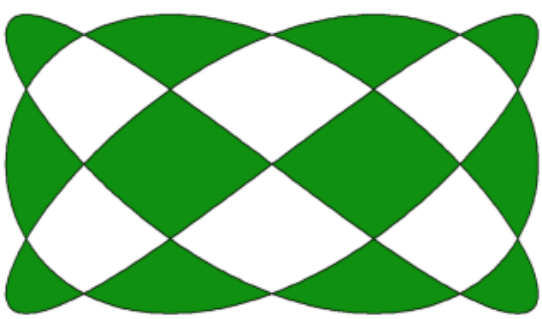


Poin utama dari bahasa matriks adalah memungkinkan untuk menghasilkan tabel fungsi dengan mudah.

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

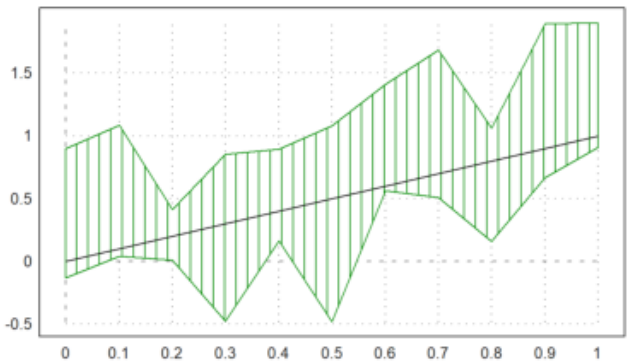
Kami sekarang memiliki vektor x dan y nilai. plot2d() dapat memplot nilai-nilai ini sebagai kurva yang menghubungkan titik-titik. Plotnya bisa diisi. Pada kasus ini ini menghasilkan hasil yang bagus karena aturan lilitan, yang digunakan untuk isi.

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



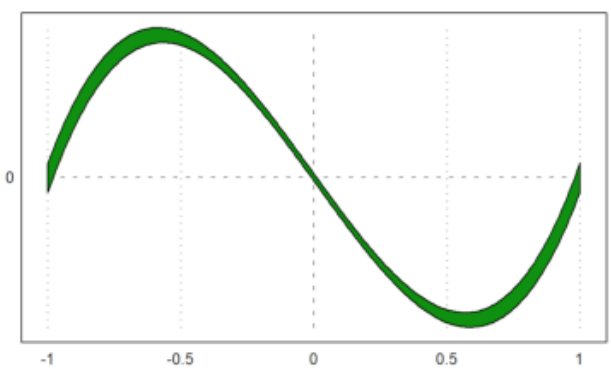
Sebuah vektor interval diplot terhadap nilai x sebagai daerah terisi antara nilai interval bawah dan atas.
Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga digunakan untuk memplot kesalahan statistik

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



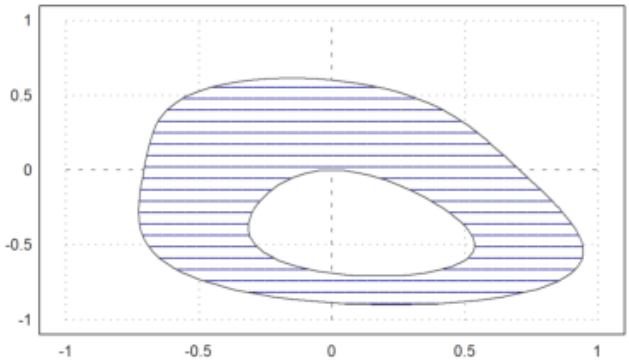
Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka plot2d akan memplot rentang interval yang terisi dalam bidang. Gaya isian sama dengan gaya poligon.

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



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

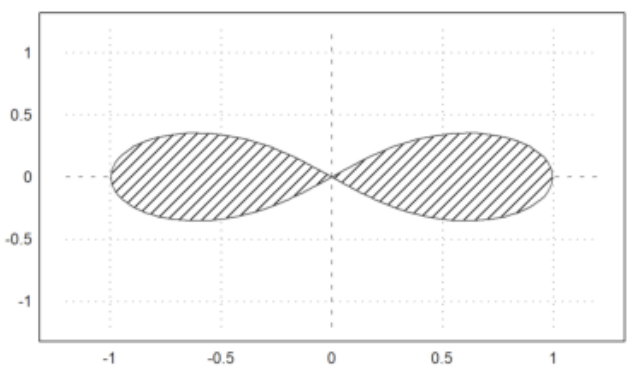
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



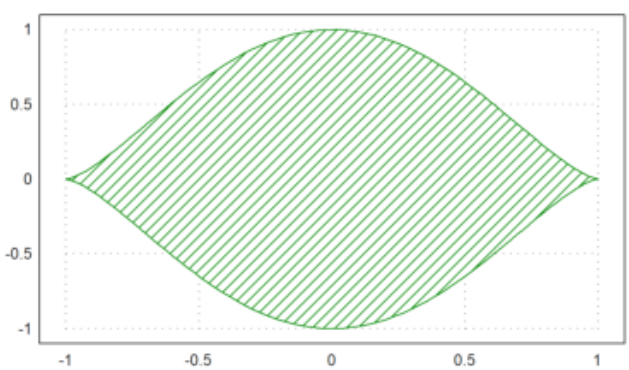
We can also fill ranges of values like

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

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




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

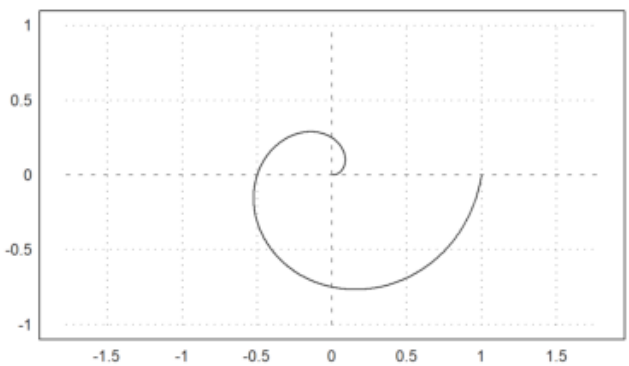


Nilai-x tidak perlu diurutkan. (x,y) hanya menggambarkan kurva. Jika x diurutkan, kurva tersebut merupakan grafik fungsi.
Dalam contoh berikut, kami memplot spiral.

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

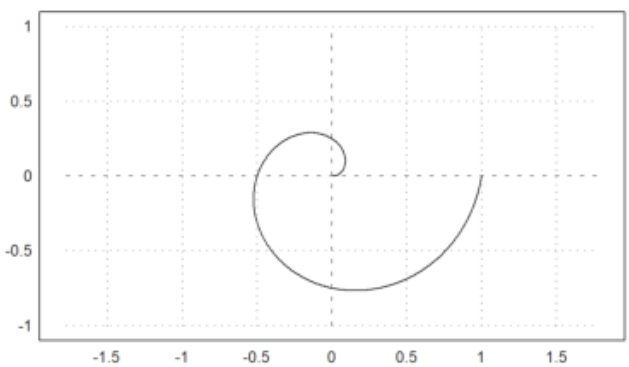
Kita perlu menggunakan banyak titik untuk tampilan yang halus atau fungsi adaptif() untuk mengevaluasi ekspresi (lihat fungsi adaptif() untuk lebih jelasnya).

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

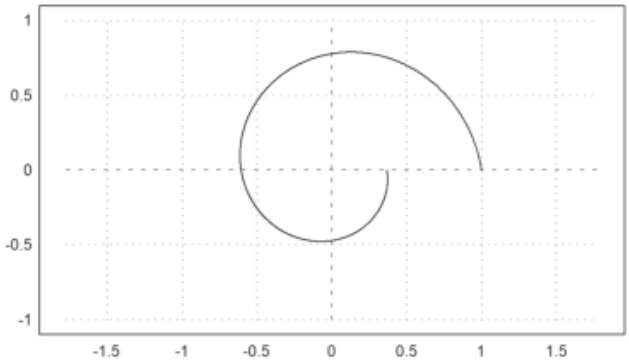


Atau, dimungkinkan untuk menggunakan dua ekspresi untuk kurva. Berikut ini plot kurva yang sama seperti di atas.

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




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



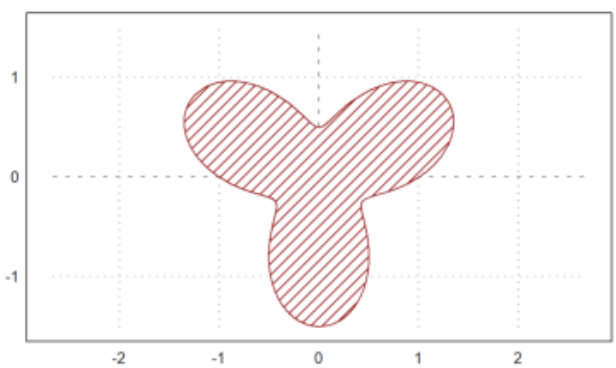
Contoh selanjutnya, kita akan menggambar kurva

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

dengan

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

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



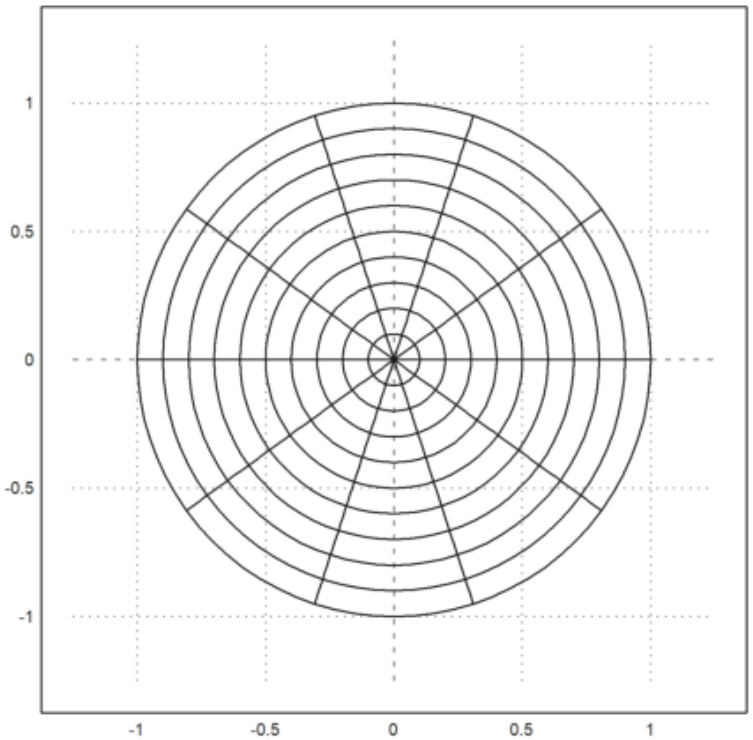
Menggambar Grafik Bilangan Kompleks

Array bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan terhubung. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi 1x2) dalam argumen cgrid, hanya garis kisi tersebut yang terlihat.

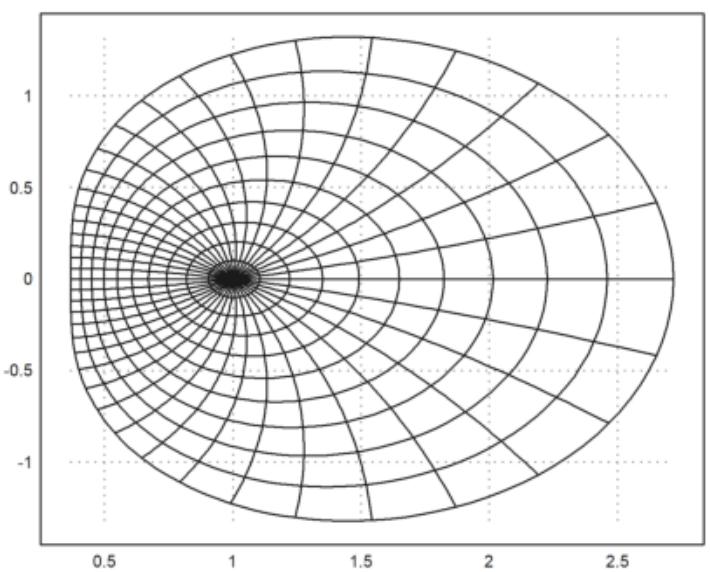
Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi di bidang kompleks.

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

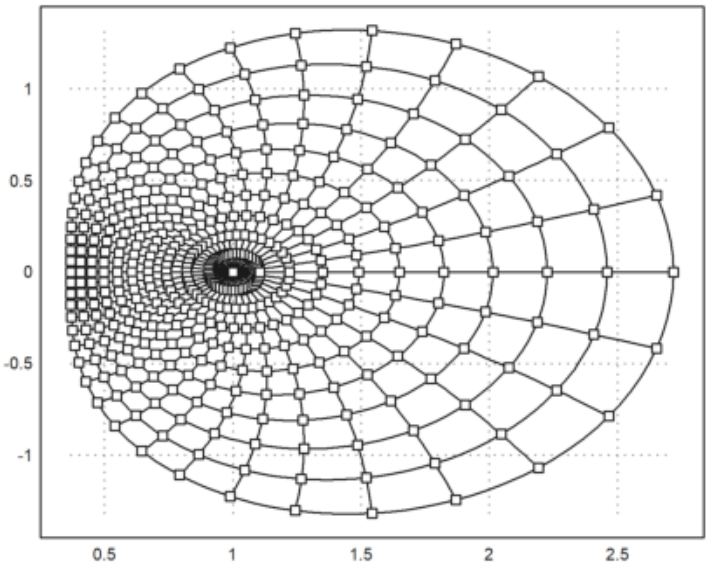
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```




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




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

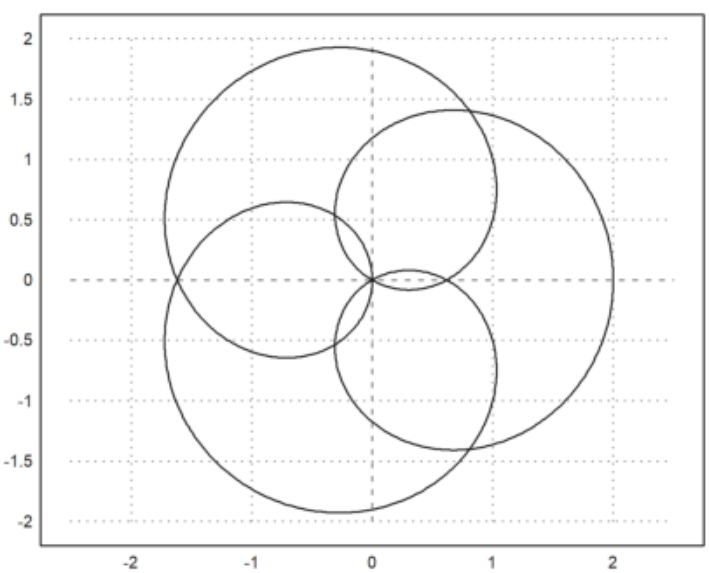


Sebuah vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian real dan bagian imajiner.

Dalam contoh, kami memplot lingkaran satuan dengan lateks:

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

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



Plot Statistika

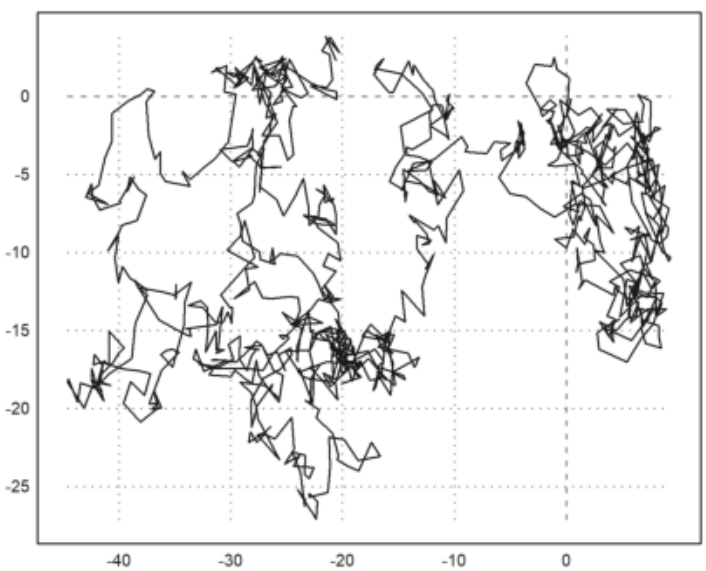
Ada banyak fungsi yang dikhususkan pada plot statistik. Salah satu plot yang sering digunakan adalah plot kolom.

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

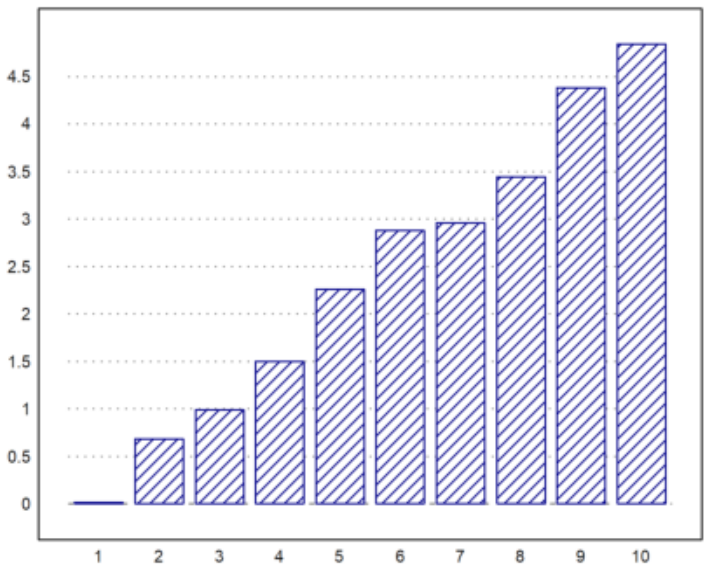
```
>plot2d(cumsum(randnormal(1,1000))):
```

Menggunakan dua baris menunjukkan jalan dalam dua dimensi.

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

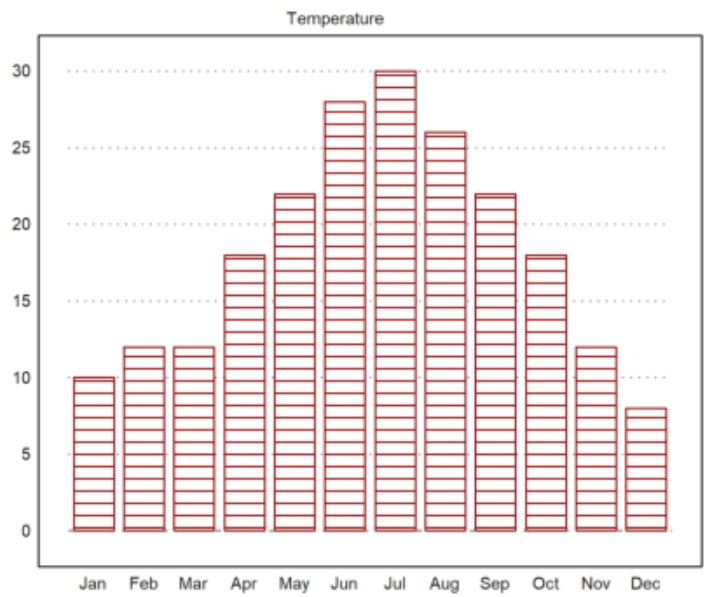


```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

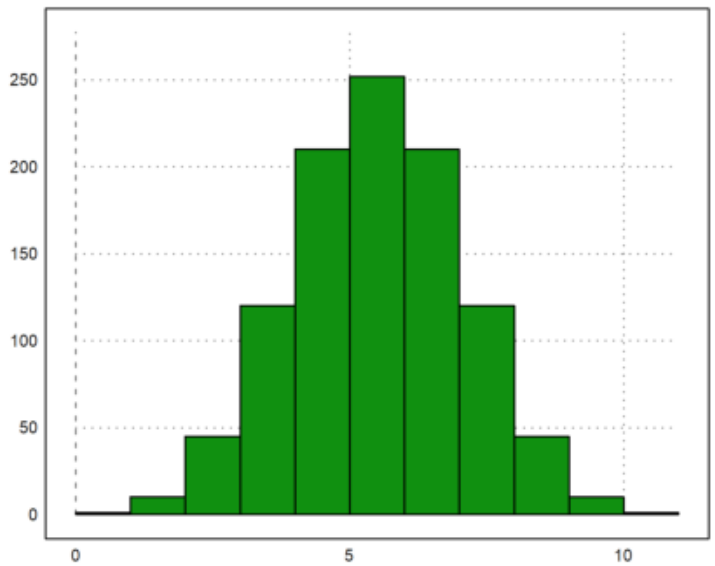


Itu juga dapat menampilkan string sebagai label

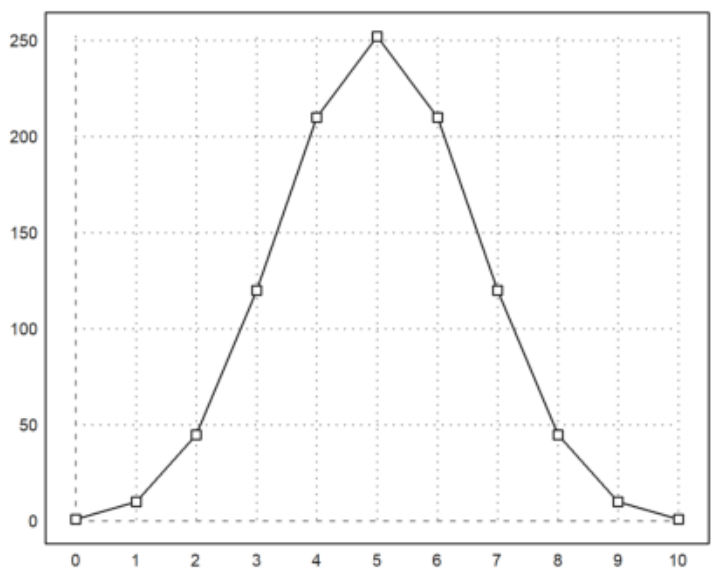
```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
>  "Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnsplot(values,lab=months,color=red,style="-");
>title("Temperature"):
```



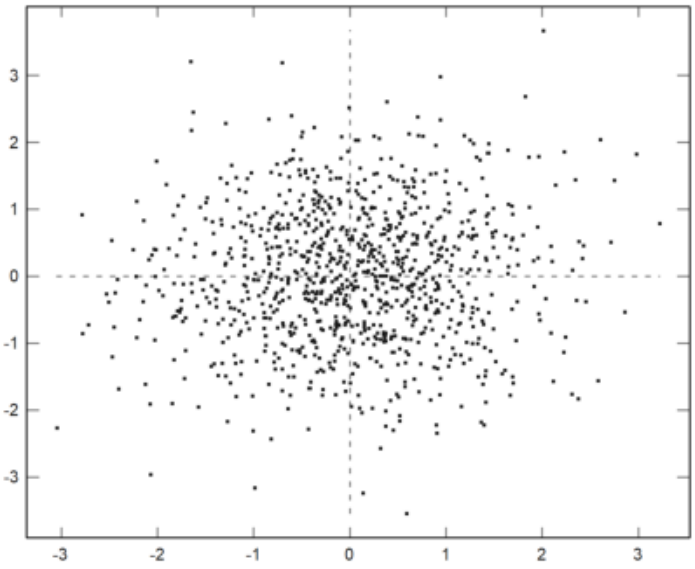
```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```



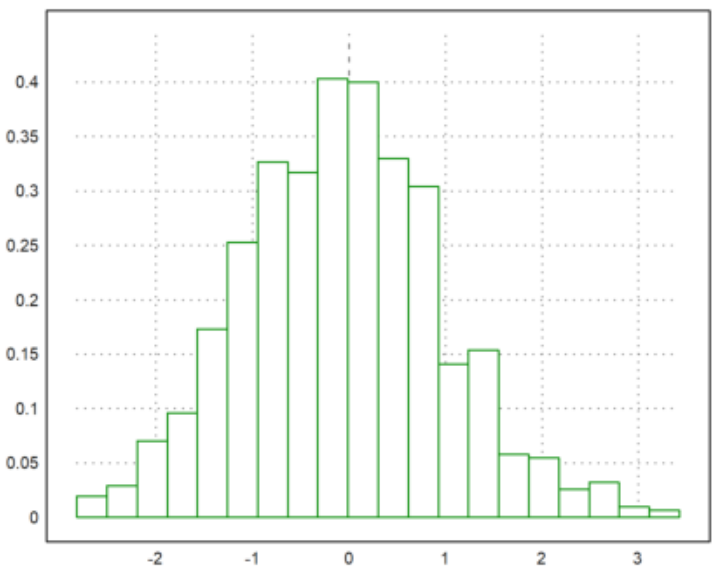

```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



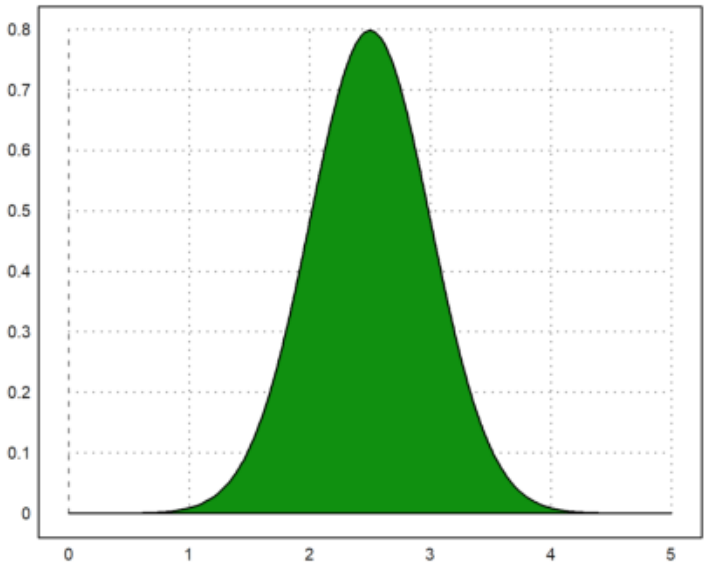

```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```




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

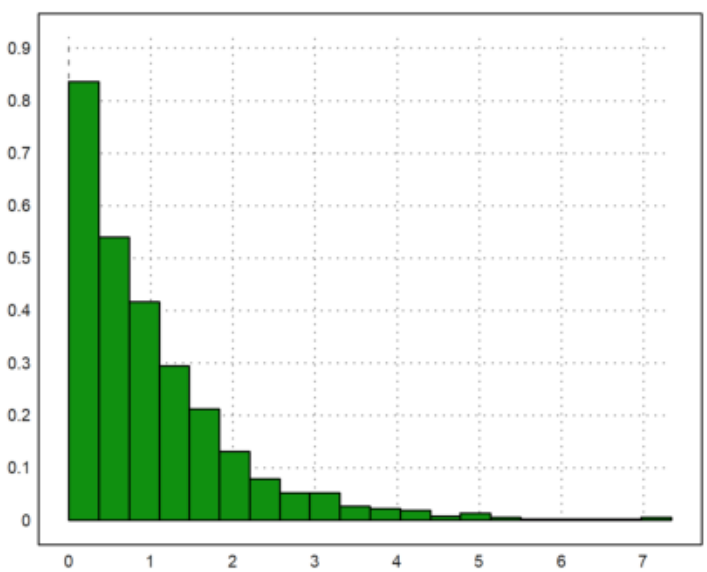



```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



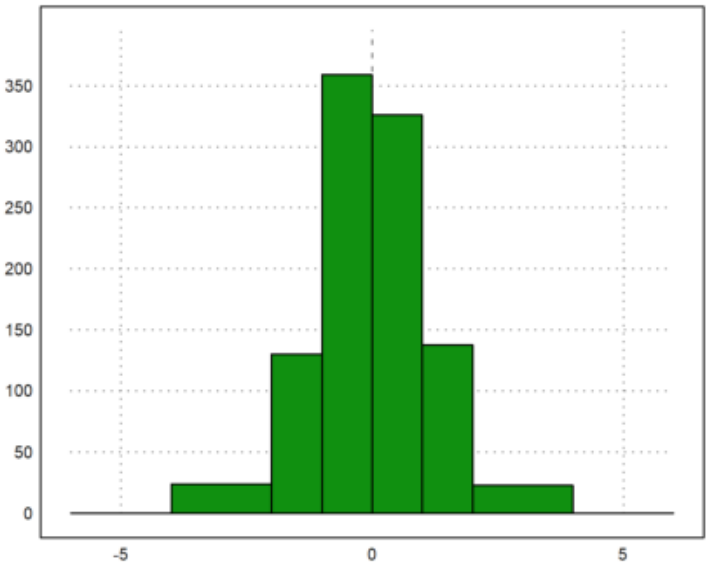
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan distribution=n dengan plot2d.

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



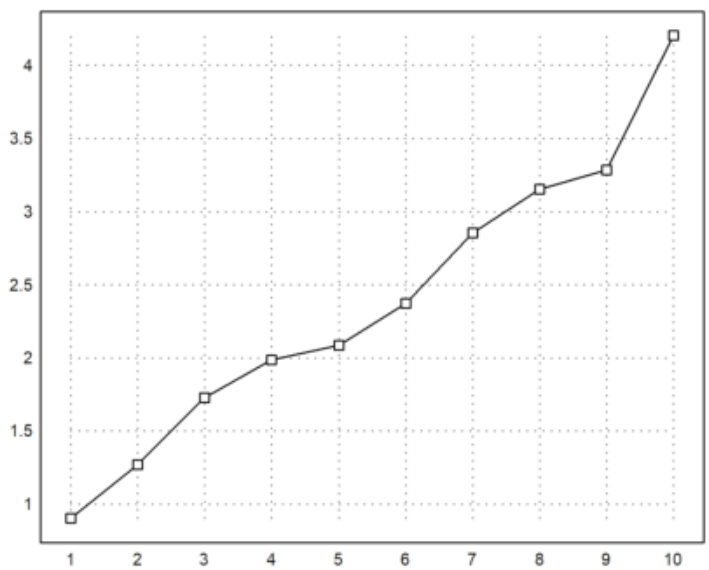
Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan >bar di plot3d, atau dengan plot kolom.

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

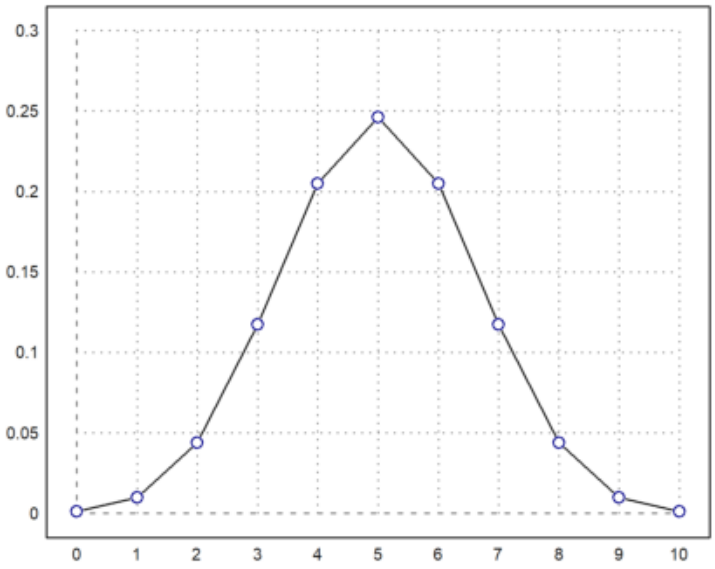


Fungsi statplot() menyetel gaya dengan string sederhana.

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



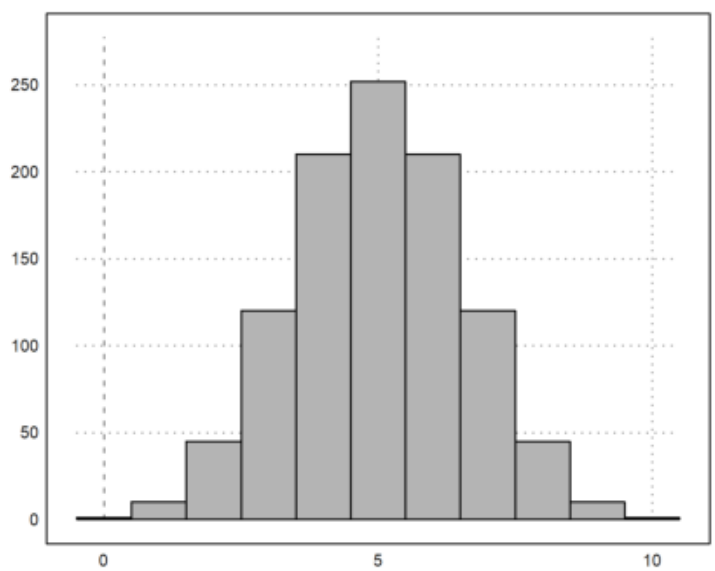

```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



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

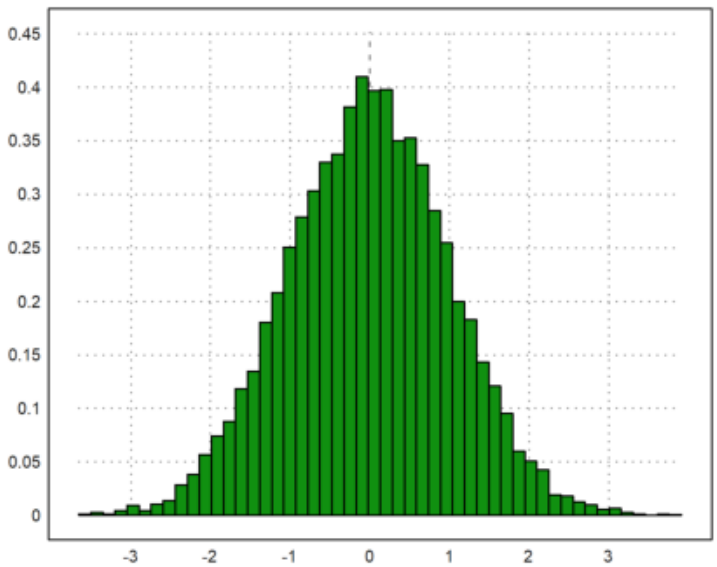
Gaya isian dapat digunakan seperti di atas.

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

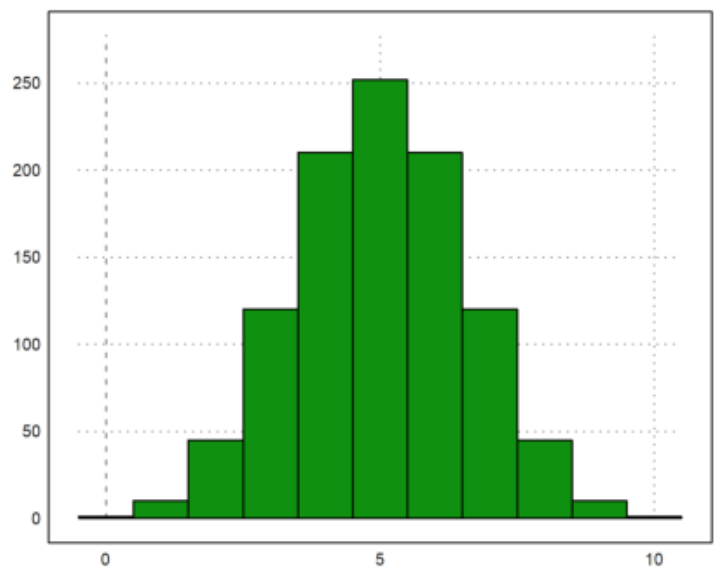


Data untuk plot batang (bar=1) dan histogram (histogram=1) dapat dinyatakan secara eksplisit dalam xv dan yv, atau dapat dihitung dari distribusi empiris dalam xv dengan >distribusi (atau distribusi=n). Histogram nilai xv akan dihitung secara otomatis dengan >histogram. Jika >genap ditentukan, nilai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

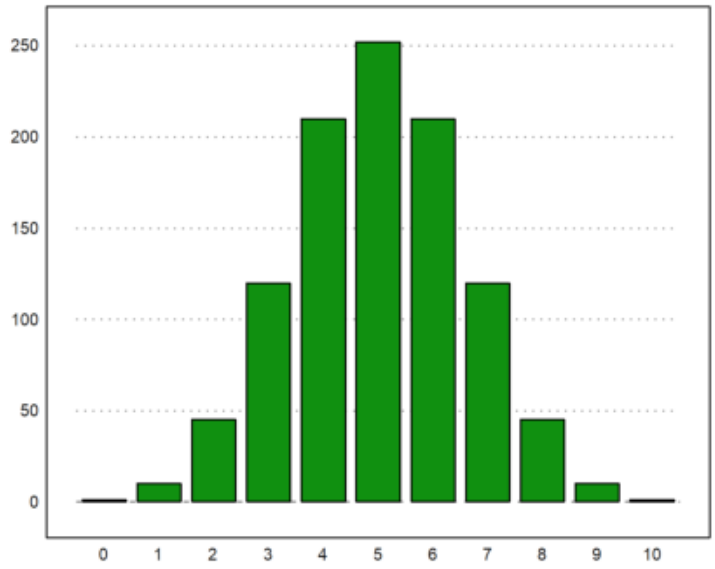
```
>plot2d(normal(10000),distribution=50):
```



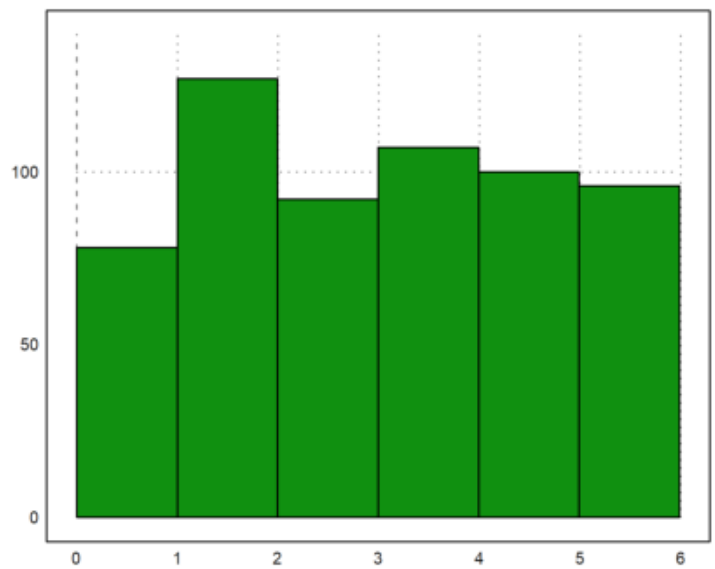

```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```




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

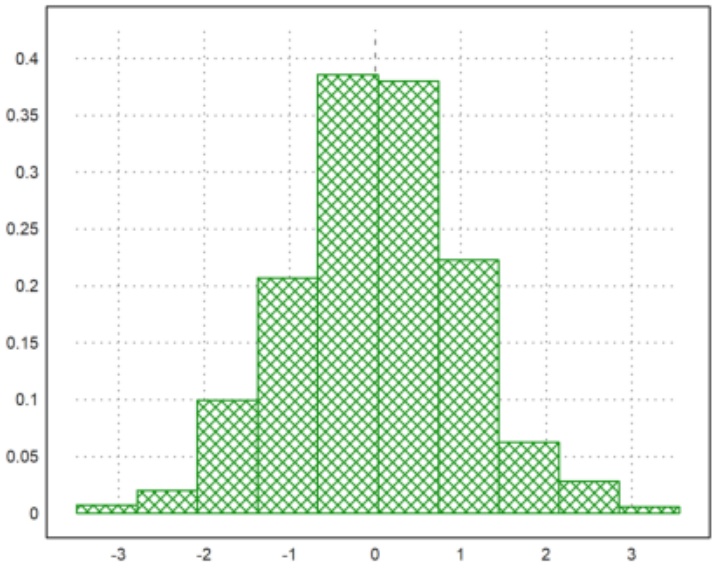



```
>plot2d(random(600)*6,histogram=6):
```



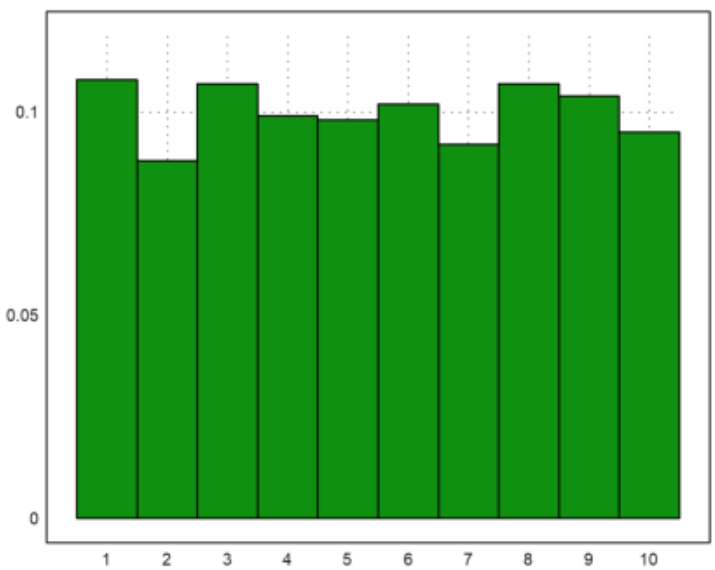
Untuk distribusi, ada parameter distribusi=n, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.
intervals.

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



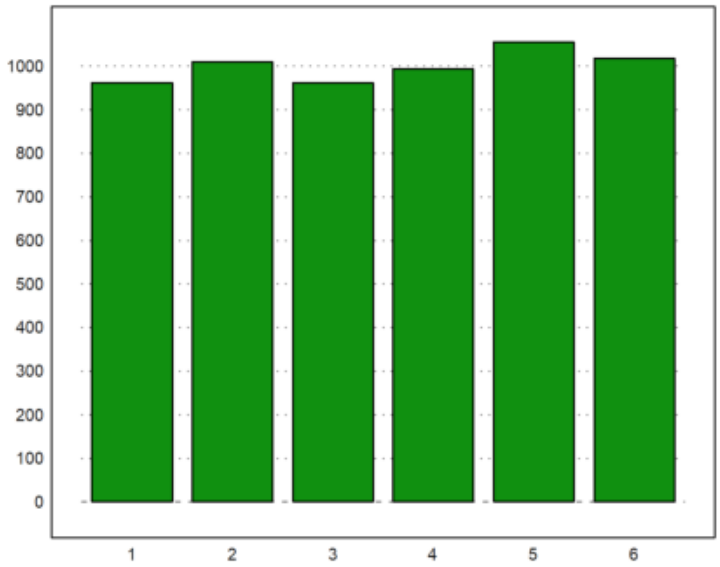
Dengan parameter even=true, ini akan menggunakan interval integer.

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

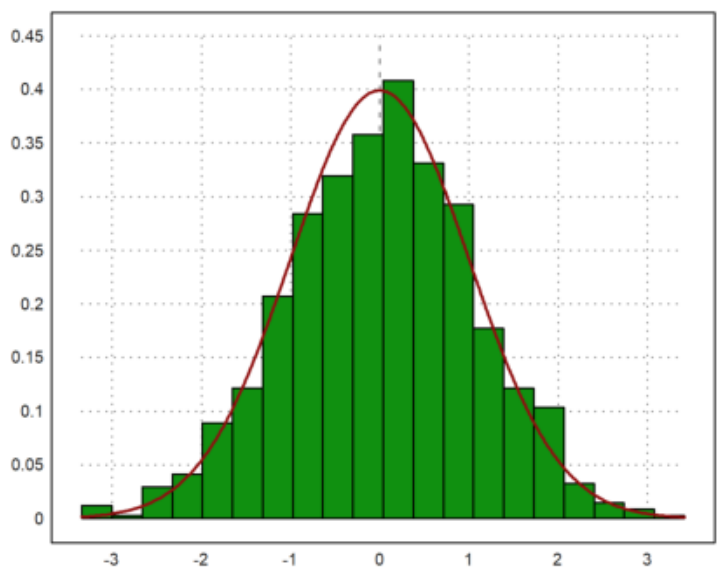


Perhatikan bahwa ada banyak plot statistik, yang mungkin berguna. Silahkan lihat tutorial tentang statistik.

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

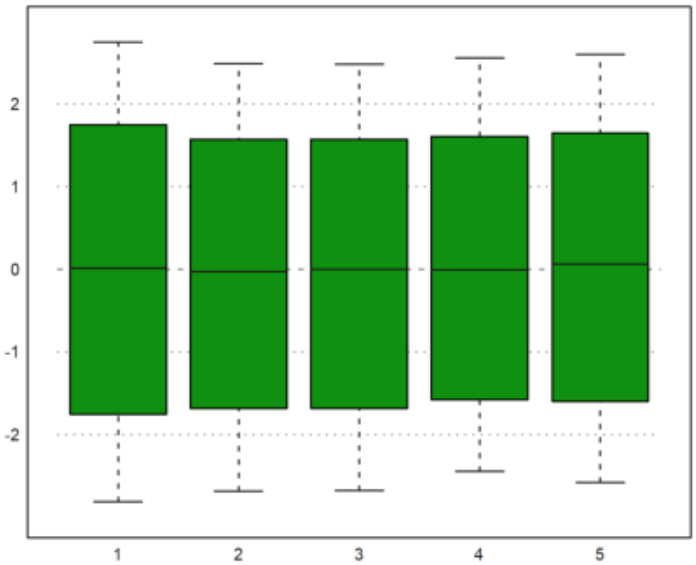



```
>plot2d(normal(1,1000),>distribution); ...
>  plot2d("qnormal(x)",color=red,thickness=2,>add):
```



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

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



Fungsi Implisit

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

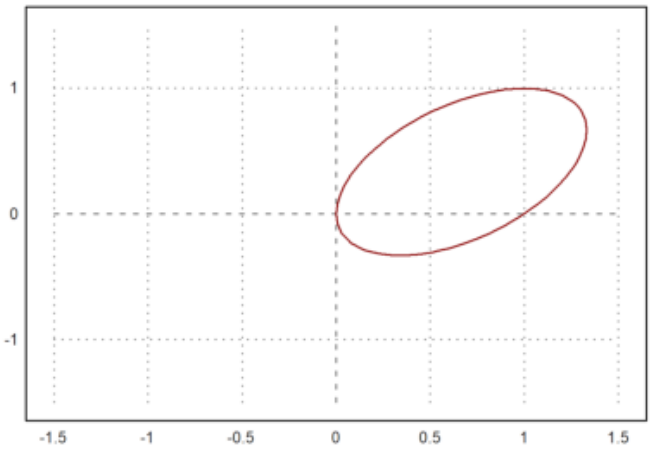
Euler dapat menandai garis level

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

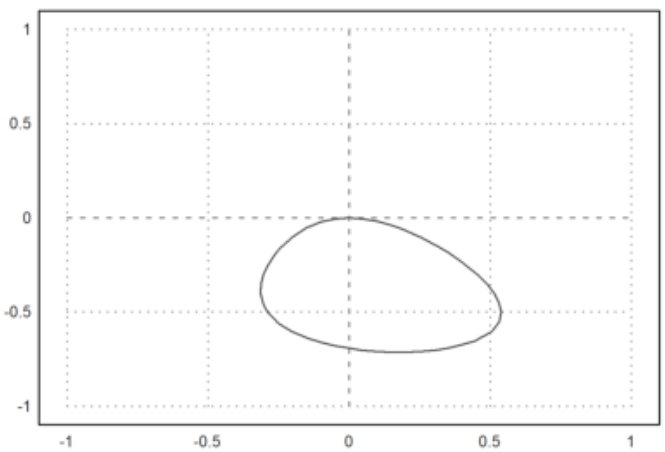
dari fungsi apapun.

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

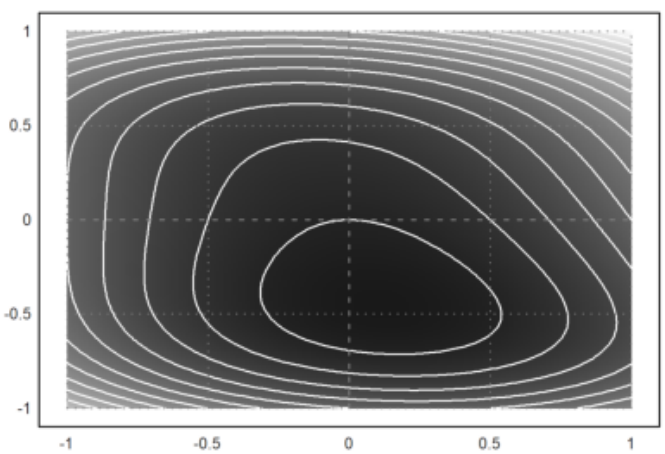
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```



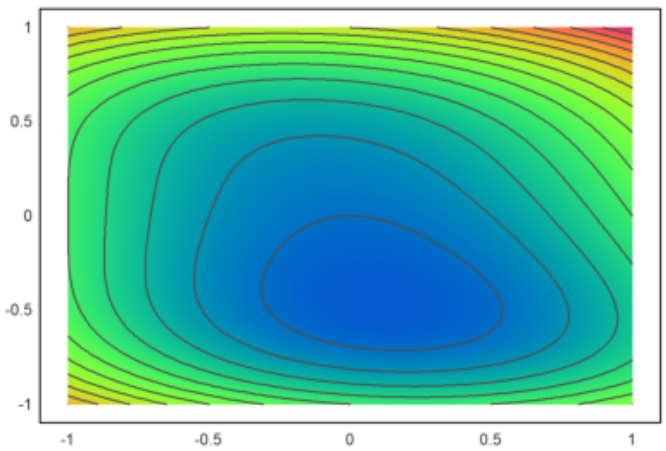

```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0): // Solutions of f(x,y)=0
```




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

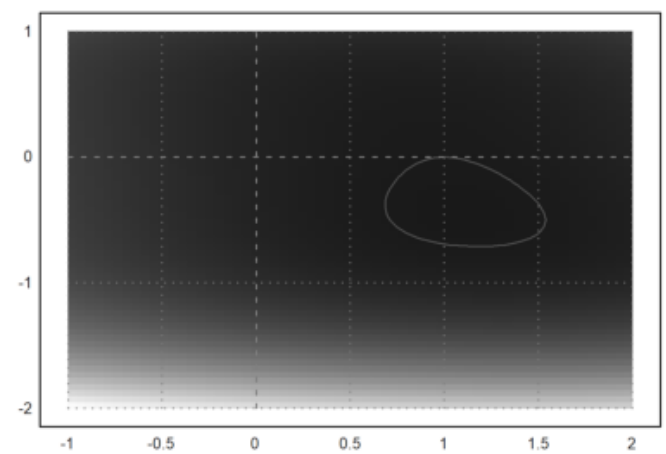


```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
```

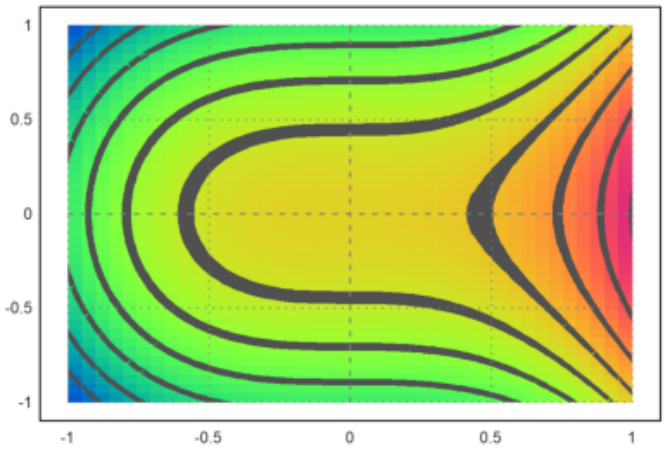


Ini berfungsi untuk plot data juga. Tetapi Anda harus menentukan rentangnya untuk label sumbu

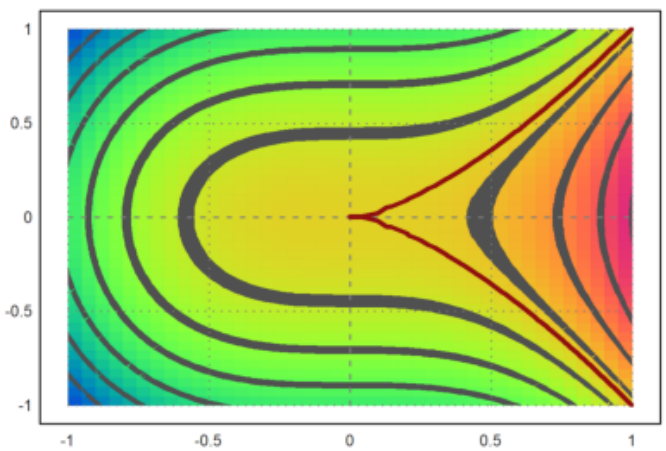
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



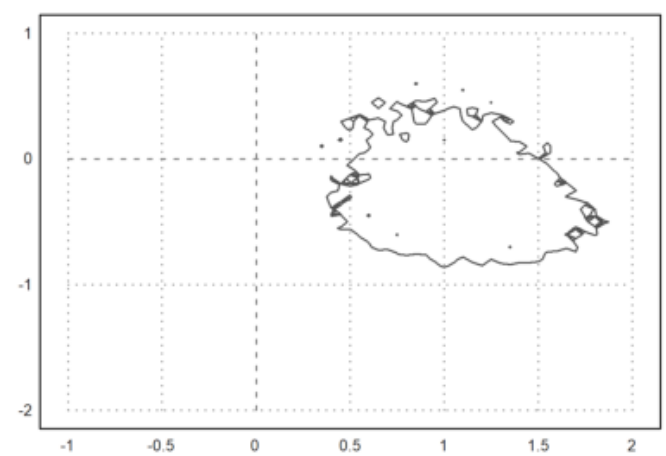

```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



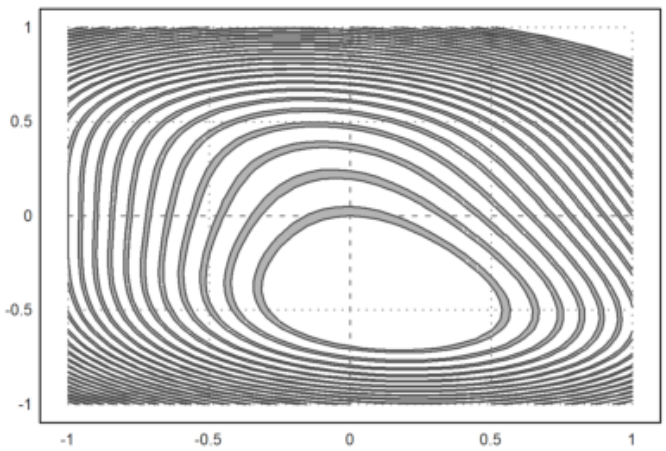

```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



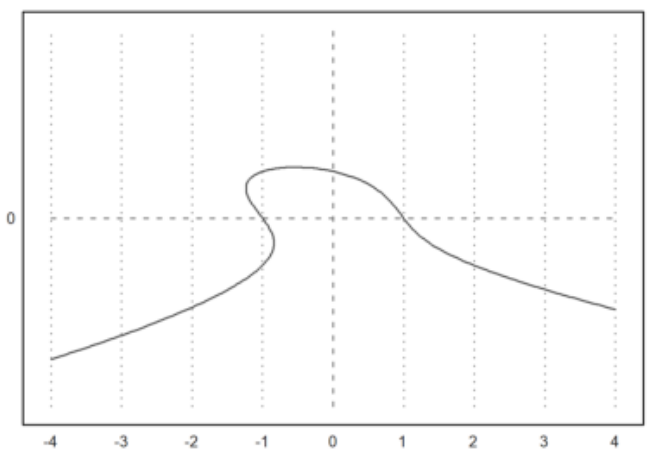

```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



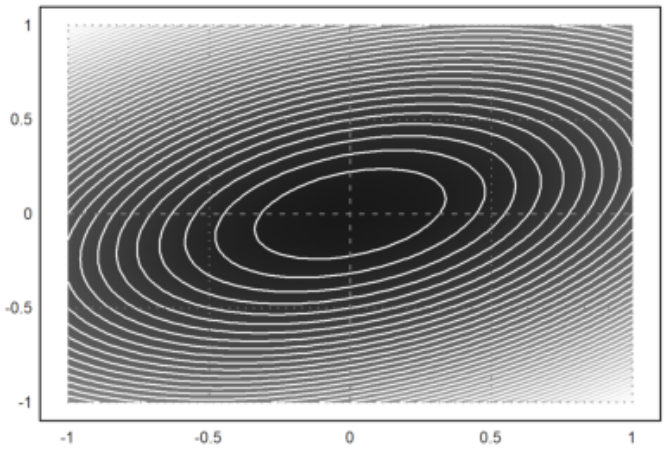
```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```




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




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

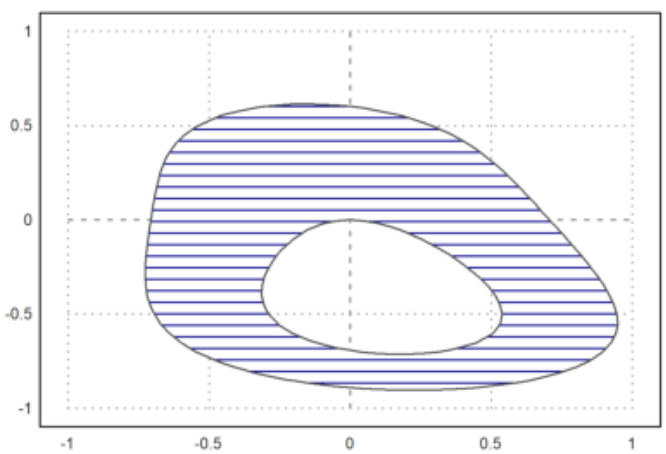


Juga dimungkinkan untuk mengisi set

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

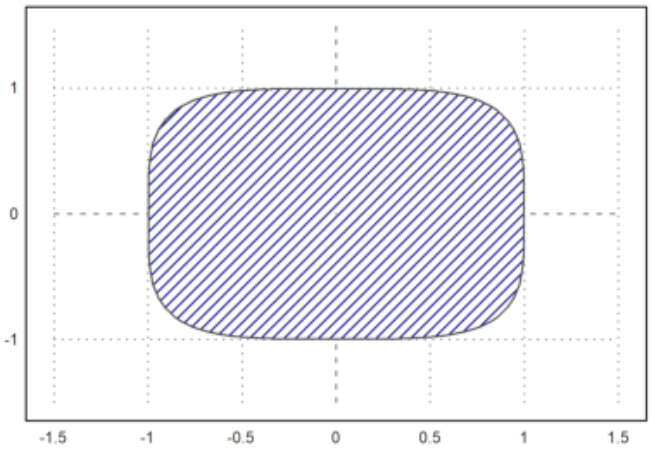
Dengan rentang tingkat.
Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

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

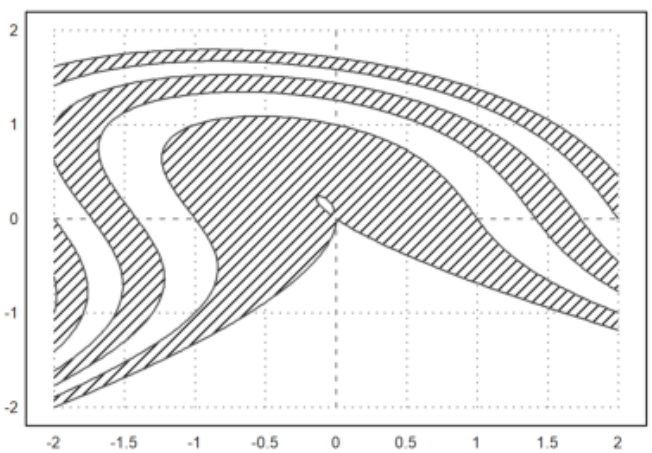


Plot implisit juga dapat menunjukkan rentang level. Kemudian level harus berupa matriks 2xn dari interval level, di mana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval. Atau, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level ke interval.

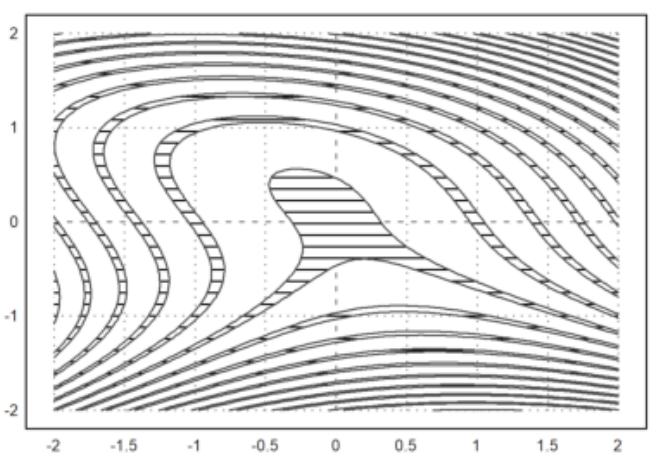
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



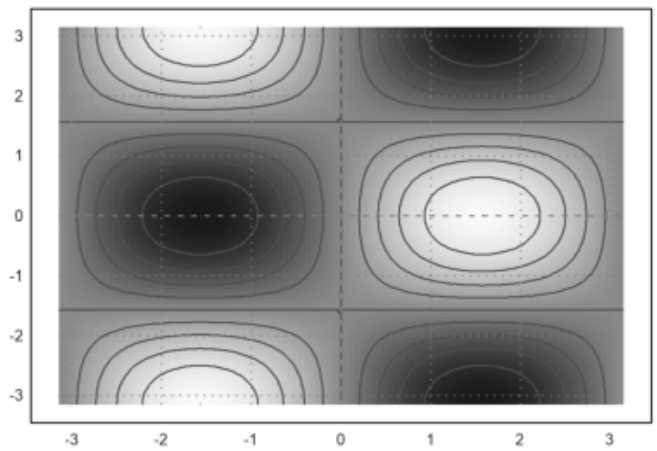

```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```




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



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

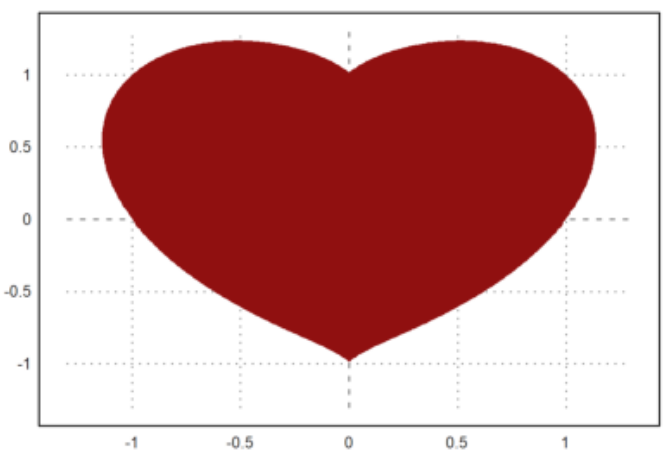


Dimungkinkan juga untuk menandai suatu wilayah

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

Ini dilakukan dengan menambahkan level dengan dua baris.

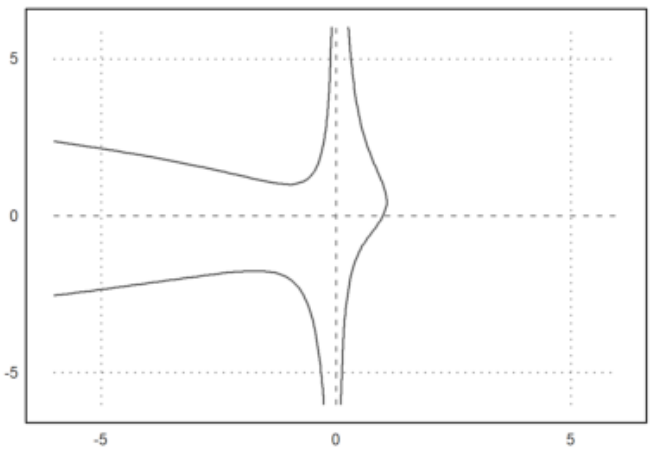
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
>  style="#",color=red,<outline, ...
>  level=[-2;0],n=100):
```



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

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

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



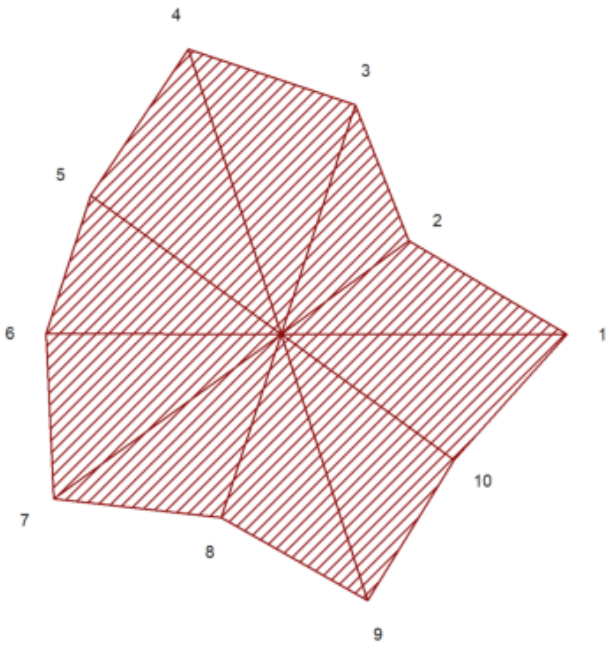

```
>function starplot1 (v, style="/", color=green, lab=none) ...
```

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

Tidak ada kotak atau sumbu kutu di sini. Selain itu, kami menggunakan jendela penuh untuk plot.

Kami memanggil reset sebelum kami menguji plot ini untuk mengembalikan default grafis. Ini tidak perlu, jika Anda yakin plot Anda berhasil.
an untuk level, dan parameter dl memperluas nilai level ke interval.

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



Terkadang, Anda mungkin ingin merencanakan sesuatu yang tidak dapat dilakukan plot2d, tetapi hampir.

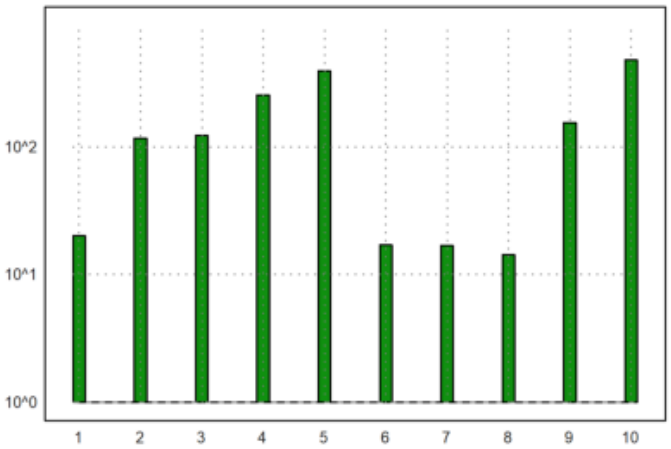
Dalam fungsi berikut, kami melakukan plot impuls logaritmik. plot2d dapat melakukan plot logaritmik, tetapi tidak untuk batang impuls.
an untuk level, dan parameter dl memperluas nilai level ke interval.

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

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

Mari kita uji dengan nilai yang terdistribusi secara eksponensial.

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



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

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

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

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

Tekan sembarang tombol untuk menghentikan animasi ini.

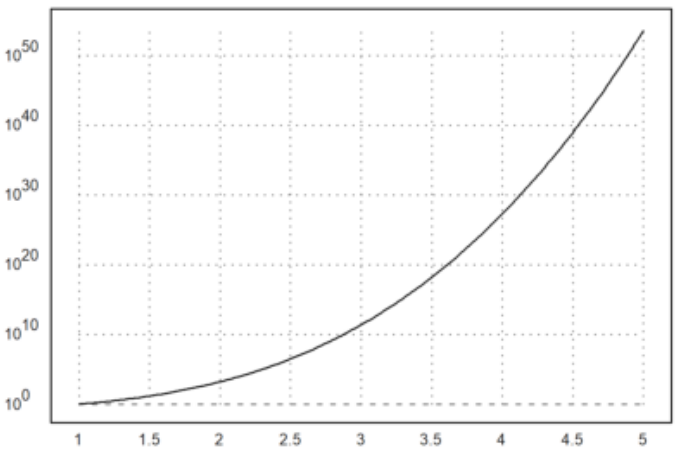
```
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

Plot Logaritmik

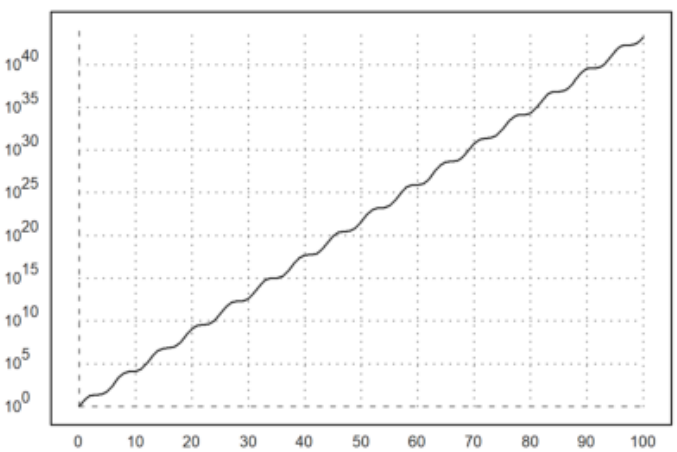
EMT menggunakan parameter "logplot" untuk skala logaritmik.
Plot logaritma dapat diplot baik menggunakan skala logaritma dalam y dengan logplot=1, atau menggunakan skala logaritma dalam x dan y dengan logplot=2, atau dalam x dengan logplot=3.

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

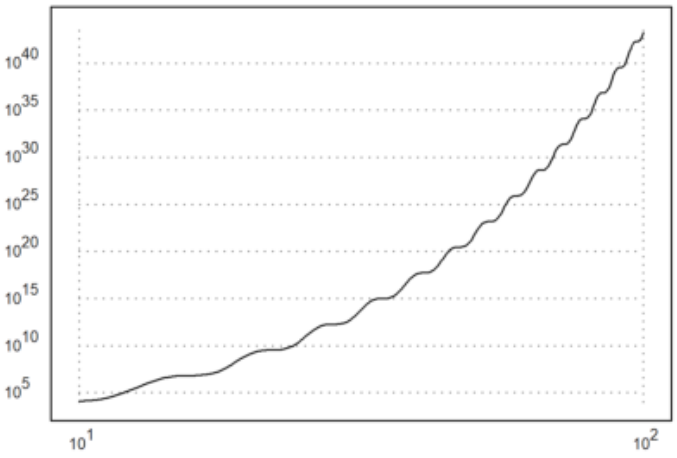
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



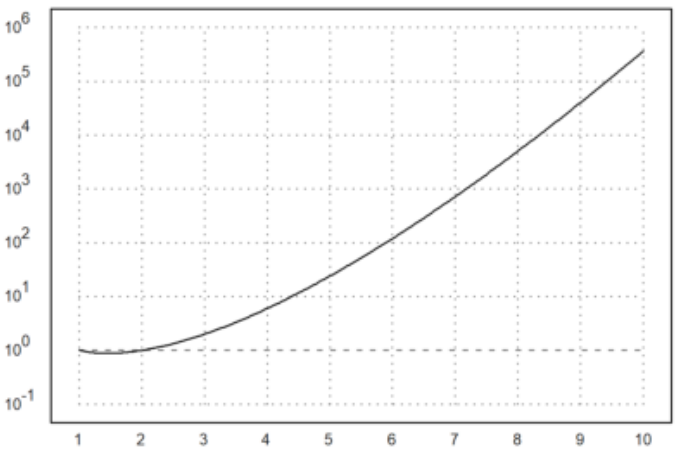

```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



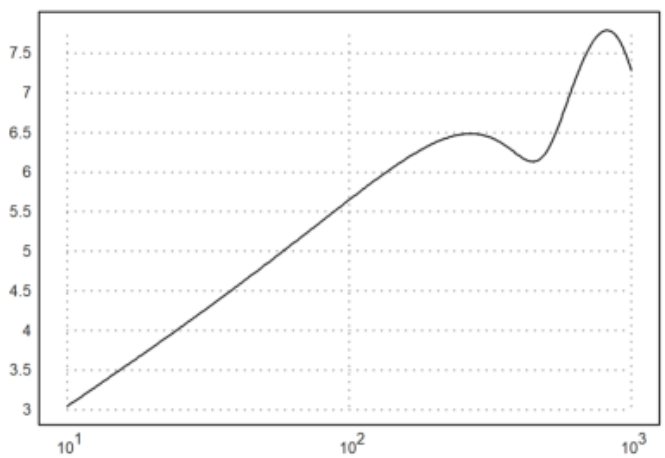

```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```




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

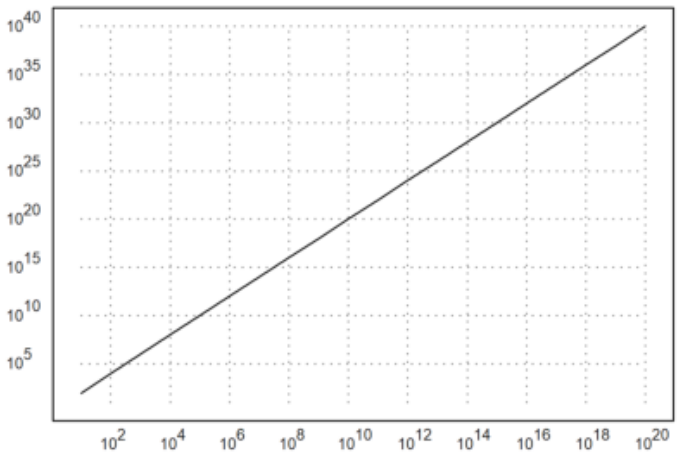



```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Ini juga berfungsi dengan plot data

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



Rujukan Lengkap Fungsi plot2d()

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

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

Parameter

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

square :jika true, mencoba menjaga skala sumbu x-y tetap sama (persegi)
n : jumlah interval (default: adaptif)
grid :
1=hanya sumbu
2=grid normal
3=sumbu di dalam
4=tanpa grid
5=grid penuh termasuk margin
6=tanda pada bingkai
7=hanya sumbu
8=sumbu dengan sub-ticks
0=tanpa grid dan label
frame :0= tanpa bingkai
framecolor :warna bingkai dan grid

margin :nilai antara 0–0.4 untuk margin sekitar plot
color :warna kurva
thickness :ketebalan garis (bisa <1 untuk garis sangat tipis)
style : gaya plot untuk garis, marker, atau isian
titik = "[]", " <>", ".", ".."", "..."", "*", "+", "|", "-", "o", "" (isian), "w" (non-transparan)
garis = "-", "–", "-.", ".", ".-.", "-.-", "->"
poligon isi/bar = "", "O", "/", "\", "\ /", "+", "|", "-", "t"
points :menampilkan titik tunggal, bukan segmen garis
addpoints : jika true, tampilkan titik dan garis

add :menambahkan plot pada plot yang sudah ada
user :mengaktifkan interaksi pengguna untuk fungsi
delta :ukuran langkah untuk interaksi pengguna
bar : diagram batang (x = batas interval, y = nilai)
histogram : frekuensi data x dalam n subinterval
distribution=n:distribusi x dengan n subinterval
even :gunakan nilai antar interval untuk histogram otomatis
steps :menampilkan fungsi sebagai fungsi tangga (steps=1,2)
adaptive :plot adaptif (n = jumlah langkah minimum)

evel :menampilkan garis level fungsi implisit 2 variabel

outline :menggambar batas dari area level

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

hue :memberi warna gradien (hue) pada plot level sesuai nilai fungsi
contour :menampilkan garis level otomatis
nc :jumlah garis level otomatis
title :judul plot (default "")
xl, yl :label sumbu x dan y
samller :jika >0, memberi ruang lebih untuk label di sebelah kiri

ertical :mengatur label vertikal. 1(teks vertikal), 2(label angka vertikal pada sumbu-y)
filled :mengisi area plot kurva
fillcolor :warna isi untuk bar/kurva isi

ogplot :menentukan jenis skala logaritmik. 1(logaritmik pada sumbu y), 2(logaritmik pada sumbu x dan y), 3(logaritmik pada sumbu x).
own :string yang menunjuk ke rutin plot buatan sendiri
maps :memetakan ekspresi (0 lebih cepat), fungsi selalu dipetakan
contourcolor:warna garis kontur
contourwidth:ketebalan garis kontur
slipping :menyalakan/mematikan pemotongan (default aktif)
title :digunakan untuk mendeskripsikan gambar. Judul akan terlihat diawas gambar. Judul lain juga bisa ditambahkan dengan perintah label() atau labelbox(). Judul juga bisa ditulis dalam LaTex.
cgrid :jumlah garis grid untuk plot bilangan kompleks sebaiknya kelipatan ukuran matriks - 1 bisa berupa [cx,cy]

Overview

Fungsi plot2d() dapat digunakan untuk menggambar:
- ekspresi, koleksi panggilan, atau fungsi satu variabel,
- kurva parametik,
- data x terhadap y,
- fungsi implisit,
- diagram batang,
- grid kompleks,
- poligon.

Jika diberikan ekspresi atau fungsi untuk xv, maka plot2d() akan menghitung nilainya dalam rentang yang ditentukan (a dan b, default [-2,2]). Rentang-y dihitung otomatis kecuali c dan d ditentukan, atau r digunakan

(memberikan area [-r,r]).
Secara default, plot2d() memakai evaluasi adaptif. Untuk mempercepat fungsi kompleks, bisa dimatikan dengan <adaptive dan jumlah interval n dikurangi. Selain itu, plot2d() secara default menggunakan mapping (menghitung elemen per elemen). Jika fungsi bisa menerima vektor x sekaligus, gunakan <maps agar lebih cepat.
Jika xv dan yv diberikan, maka plot2d() akan menggambar kurva dengan koordinat (xv,yv). Dalam hal ini, perlu didefinisikan rentang parameter dengan xmin, xmax. Ekspresi string harus berbentuk fungsi dari variabel parameter x.

```
>
```

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.