

# Menggambar Grafik 2D dengan EMT

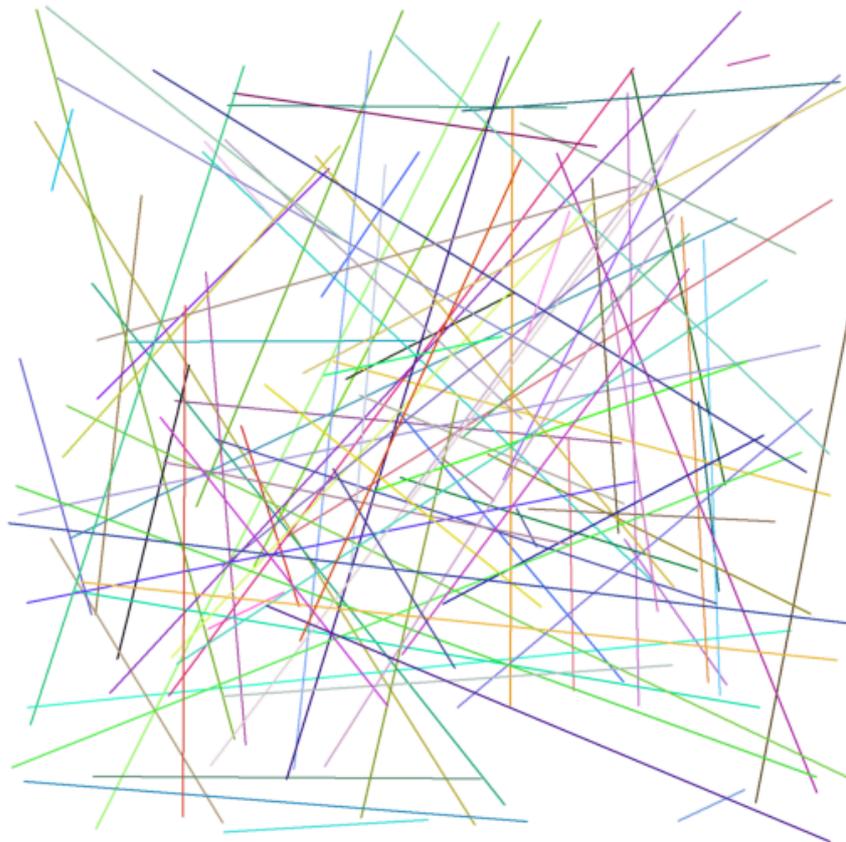
Notebook ini menjelaskan tentang cara menggambar berbagai kurva dan grafik 2D menggunakan software EMT. EMT menyediakan fungsi `plot2d()` untuk menggambar berbagai kurva dan grafik dua dimensi (2D).

## Plot Dasar

Ada beberapa fungsi dasar untuk membuat grafik. Terdapat koordinat layar, yang selalu berkisar dari 0 hingga 1024 pada setiap sumbu, terlepas dari apakah layar berbentuk persegi atau tidak. Selain itu, ada juga koordinat plot yang dapat diatur dengan `setplot()`. Pemetaan antara koordinat ini tergantung pada jendela plot yang sedang digunakan. Sebagai contoh, fungsi `shrinkwindow()` secara default meninggalkan ruang untuk label sumbu dan judul plot.

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

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



```
>reset;
```

catatan : dari hasil perintah tersebut akan berubah-ubah jika kita meng-enternya lagi, itu dikarenakan perintah yang tertulis adalah 'random' sehingga hasil yang diberikan akan selalu berubah-ubah.

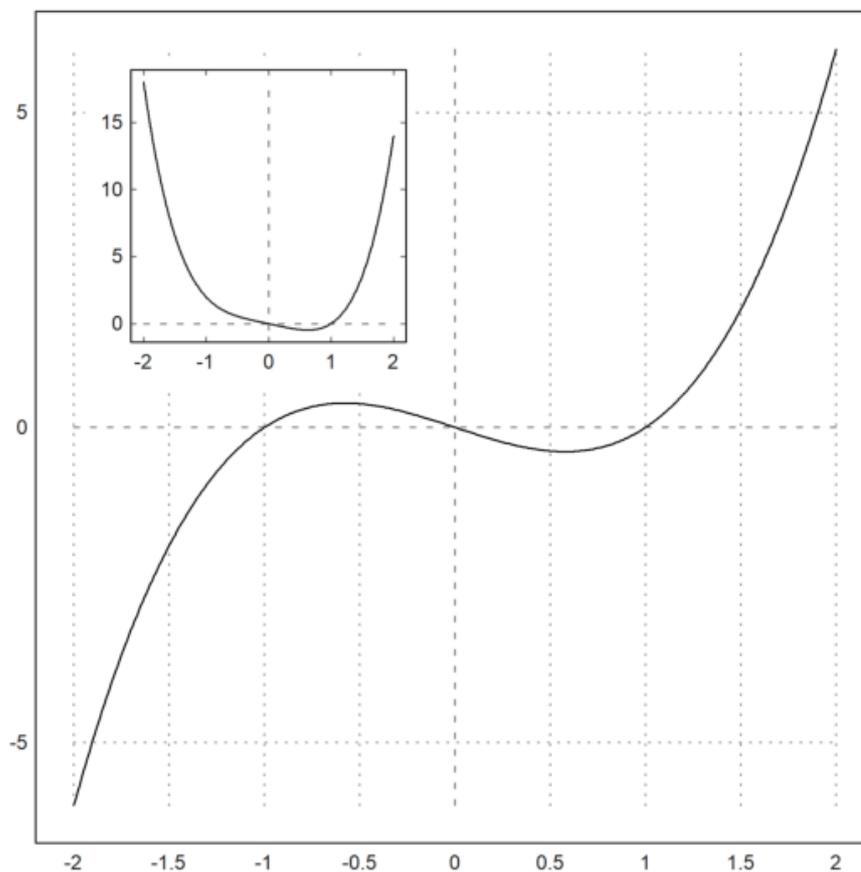
Untuk mempertahankan grafik yang telah digambar, perlu dilakukan, karena perintah `plot()` akan menghapus jendela plot sebelumnya.

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

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

Sebagai contoh lain, kita dapat menggambar grafik sebagai inset di dalam grafik lainnya. 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 sesuai kebutuhan. Selain itu, kita menyimpan dan mengembalikan jendela penuh, serta mempertahankan plot saat kita menggambar inset.

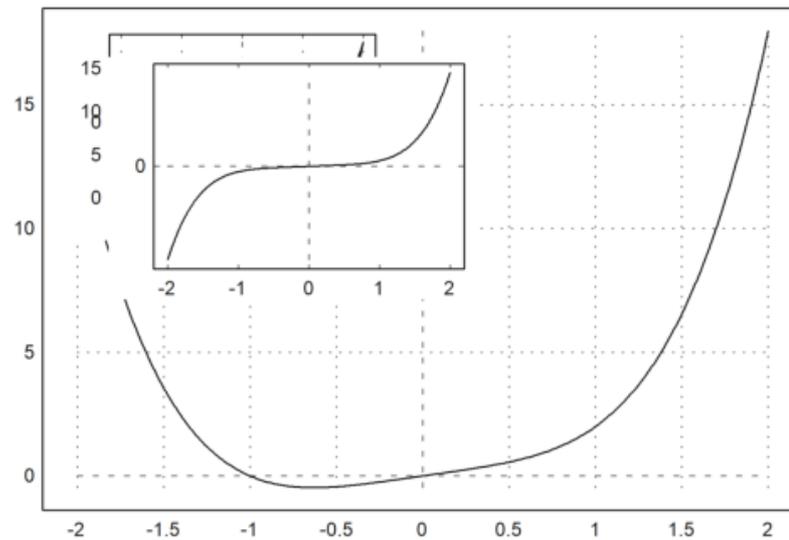
```
>plot2d("x^3-x");
>xw=200; yw=100; ww=300; hw=300;
>ow=window();
>window(xw,yw,xw+ww,yw+hw);
>hold on;
>barclear(xw-50,yw-10,ww+60,ww+60);
>plot2d("x^4-x",grid=6):
```



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

$$x^4 + x$$

```
>plot2d("x^4+x");
>xw=250; yw=150; ww=350; hw=350;
>ow=window();
>window(xw,yw,xw+ww,yw+hw);
>hold on;
>barclear(xw-50,yw-10,ww+60,ww+60);
>plot2d("x^5+x",grid=6):
```



catatan : pada penulisan perintah `plot2d` dituliskan dalam bentuk string dengan diapit oleh tanda petik(""), ukuran jendela grafik dapat diatur sesuai keinginan pada variabel `xw` dan `yw` (menetapkan posisi jendela grafik bagian kiri atas), `ww` dan `hw` (menetapkan lebar dan tinggi pada jendela grafik), perintah `'hold on'` berfungsi sebagai menjaga grafik yang sudah digambar agar tetap ada saat menambahkan grafik baru, perintah `'barclear'` untuk membersihkan area persegi di sekitar posisi dan ukuran yang telah ditentukan oleh variabel diatas. pada contoh ini menggunakan tanda titik dua untuk mengeluarkan hasil grafik tersebut.

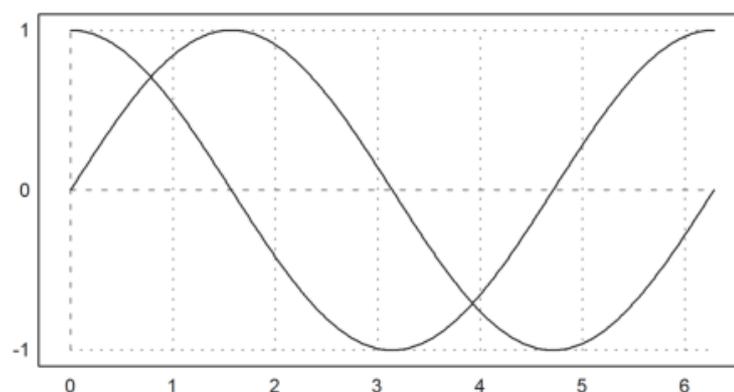
Plot dengan beberapa gambar dapat dicapai dengan cara yang sama. Ada fungsi utilitas `'figure()'` untuk ini.

## Aspek Plot

Plot default menggunakan jendela plot berbentuk persegi. Anda dapat mengubah ini dengan fungsi `'aspect()'`. Jangan lupa untuk mengatur ulang aspek setelah selesai. Anda juga dapat mengubah pengaturan default ini di menu dengan "Set Aspect" ke rasio aspek tertentu atau ke ukuran jendela grafik saat ini.

Namun, Anda juga bisa mengubahnya hanya untuk satu plot. Untuk melakukan ini, ukuran area plot saat ini diubah, dan jendela diatur sedemikian rupa sehingga label memiliki cukup ruang.

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

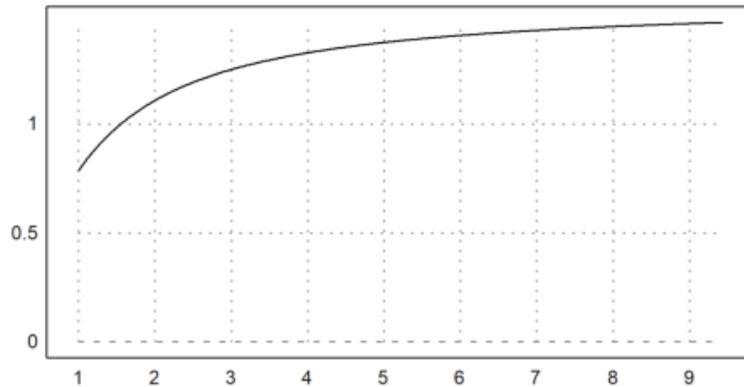


```
>aspect();
>reset;
>aspect(2);
```

$$f(x) = \text{atan}(x) \text{ dan } f(x) = \text{acos}(x)$$

dimana domain sumbu x antara 1 hingga  $3\pi$

```
>plot2d(["atan(x)", "acos(x)"], 1, 3pi):
```



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

catatan: pada contoh ini diberikan fungsi dan domain sumbu x yang kemudian fungsi tersebut digambarkan dalam satu plots/plots yang sama.

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

## Grafik 2D di Euler

EMT Math Toolbox memiliki kemampuan untuk membuat grafik 2D, baik untuk data maupun fungsi. EMT menggunakan fungsi `plot2d`. Fungsi ini dapat digunakan untuk memplot fungsi dan data.

Anda juga bisa memplot menggunakan Maxima dengan Gnuplot atau di Python dengan Matplotlib.

Euler dapat membuat plot 2D dari:

- Ekspresi
- Fungsi, variabel, atau kurva parametrik
- Vektor nilai x-y
- Sekumpulan titik di bidang
- Kurva implisit dengan level atau wilayah level
- Fungsi kompleks

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

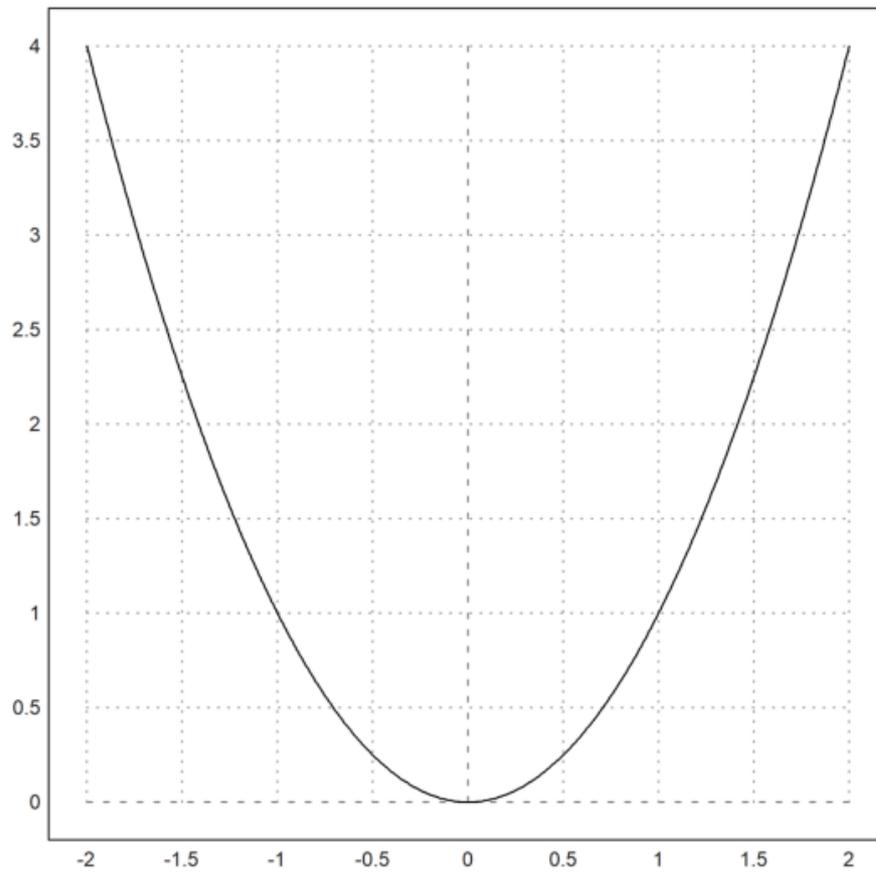
## Plot Ekspresi atau Variabel

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

Berikut adalah contoh paling dasar yang menggunakan rentang default dan mengatur rentang y yang sesuai agar grafik fungsi dapat ditampilkan dengan baik.

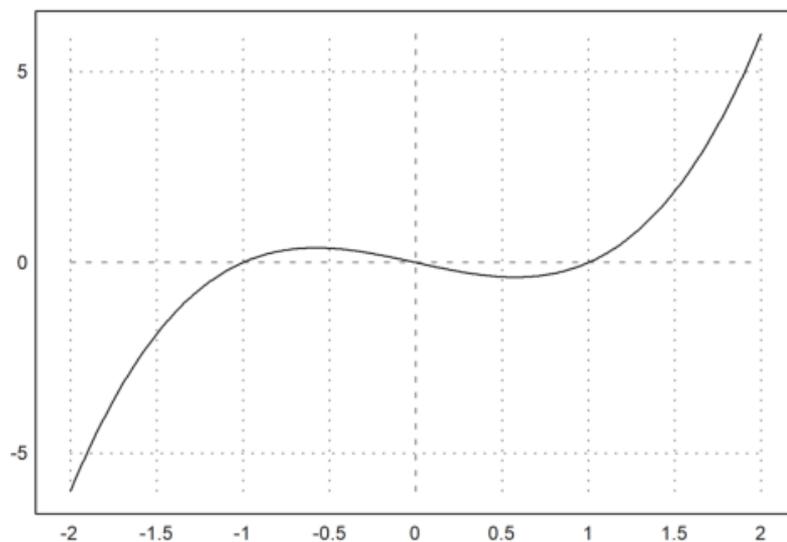
Catatan: Jika Anda mengakhiri baris perintah dengan titik dua (`:`), grafik akan dimasukkan ke dalam jendela teks. Jika tidak, tekan TAB untuk melihat grafik jika jendela plot tertutupi.

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



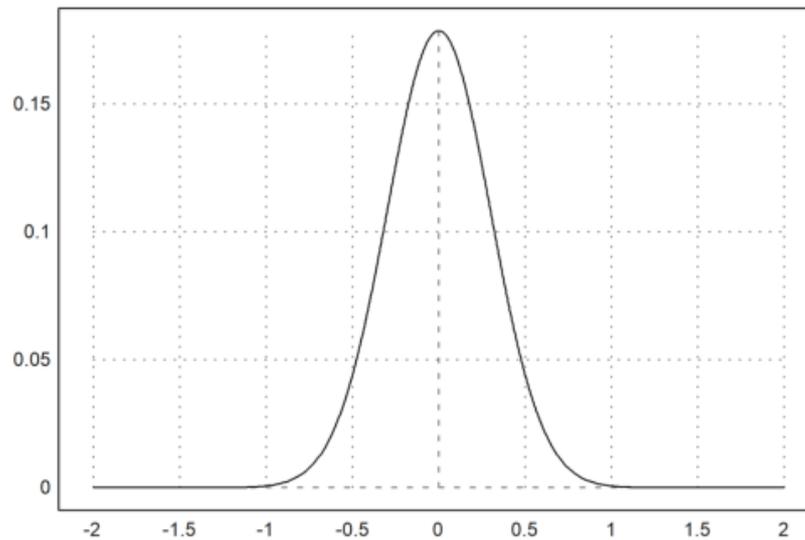
catatan: hasil dari grafik tersebut, berbentuk parabola simetris yang berpusat di (0,0) dengan sumbu simetri di sumbu y, pada sumbu ya selalu non-negatif karena kuadrat selalu bernilai positif atau nol. grafik ini memanjang ke atas semakin curam seiring dengan semakin jauhnya nilai x dari nol.

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



catatan: fungsi tersebut merupakan fungsi polinomial dengan derajat tiga, dan grafik berbentuk 'S', dimana grafiknya memotong sumbu x di  $(x=-1,0)$ ,  $(x=0,0)$ , dan  $(x=1,0)$ . fungsi ini akan terus naik untuk nilai x positif dan semakin turun untuk x negatif, dengan aspect ratio 1.5, dimana lebar grafik akan terlihat lebih lebar dibanding tingginya.

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



catatan: fungsi tersebut membentuk kurva lonceng yang berpusat di  $x=0$ . dengan  $a=5.6$ , kurva ini akan sangat sempit di sekitar titik  $x=0$  karena nilai  $a$  cukup besar

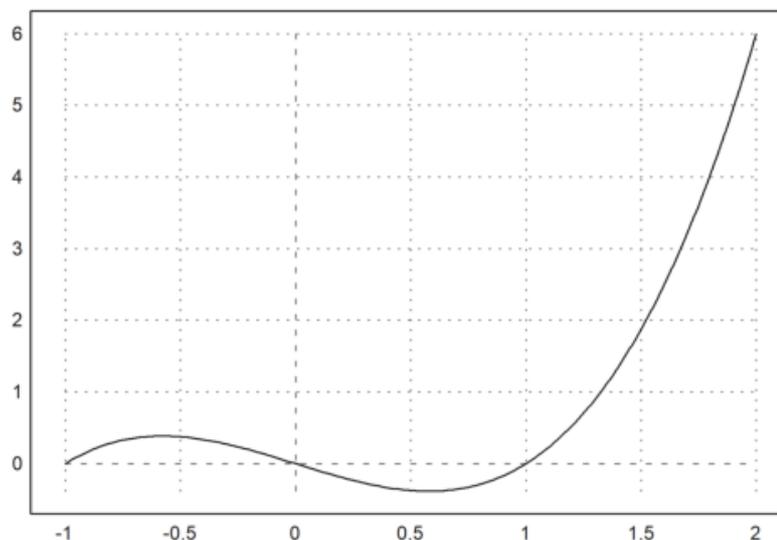
Dari beberapa contoh sebelumnya, Anda dapat melihat bahwa secara default, gambar plot menggunakan sumbu X dengan rentang nilai dari -2 hingga 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan batas-batas X (dan Y) di belakang ekspresi yang digambar.

Rentang plot diatur dengan parameter berikut:

- $a, b$ : rentang x (default: -2 hingga 2)
- $c, d$ : rentang y (default: menyesuaikan dengan nilai-nilai data)
- $r$ : sebagai alternatif, radius di sekitar pusat plot
- $cx, cy$ : koordinat pusat plot (default: 0,0)

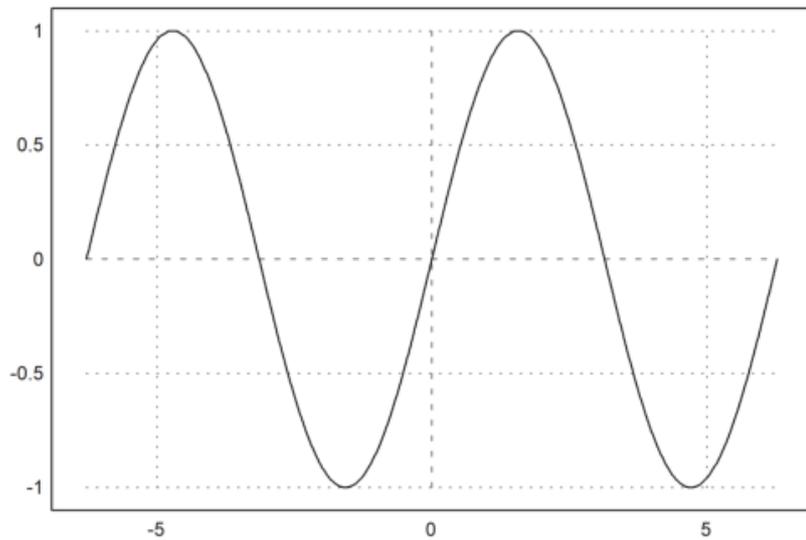
Dengan mengatur parameter ini, Anda bisa mengontrol tampilan plot untuk menyesuaikan kebutuhan Anda.

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



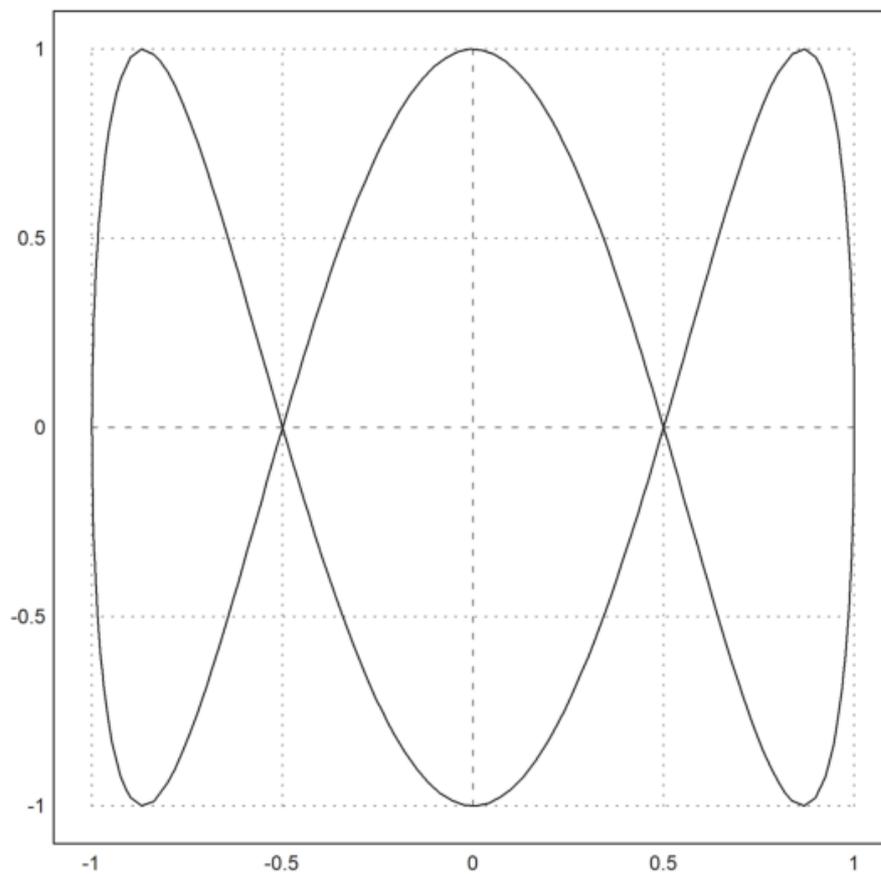
catatan: Hasil grafik pada fungsi tersebut menunjukkan tiga titik nol pada  $x=-1$ ,  $x=0$ , dan  $x=1$ , dengan grafik yang menurun antara -1 dan 0, kemudian naik antara 0 dan 1, selanjutnya naik dengan curam untuk  $x$  lebih dari 1.

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



catatan: plot ini menampilkan dua siklus penuh dari fungsi sinus, yakni satu siklus negatif dari  $-2\pi$  ke 0 dan satu siklus positif dari 0 ke  $2\pi$ .

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

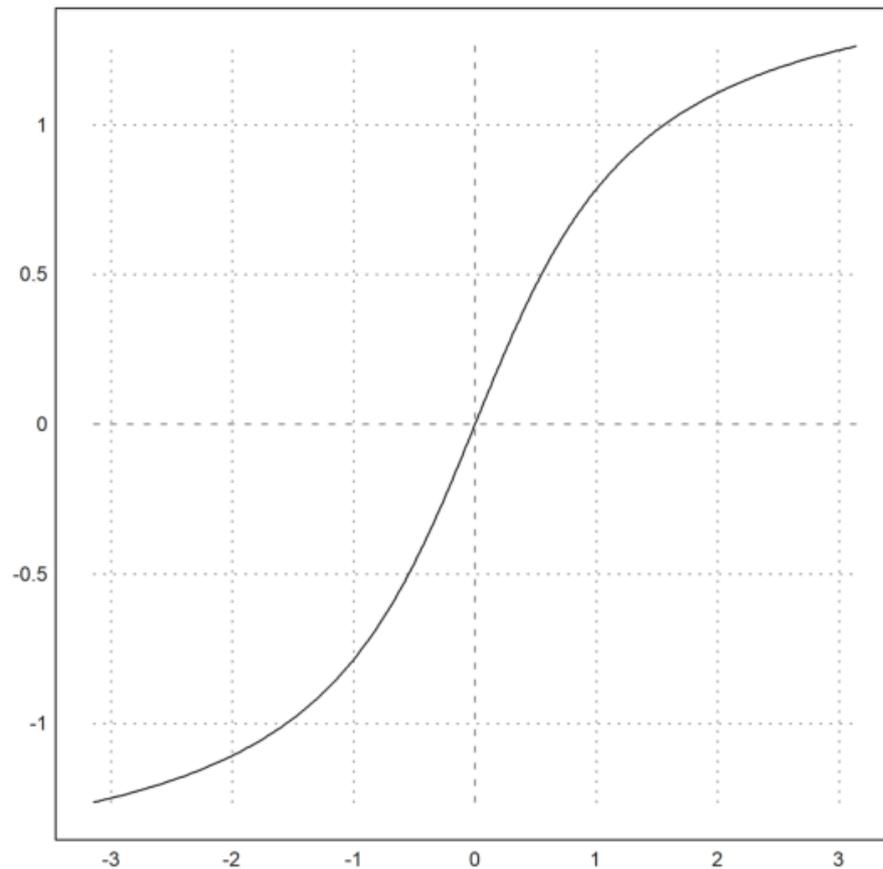


catatan: Plot ini menampilkan kedua fungsi di rentang yang sama, dengan  $\sin(3x)$  mengalami lebih banyak osilasi dibandingkan  $\cos(x)$  di interval  $[0, 2\pi]$ .

*arctan(x)*

pada interval  $[-\pi, \pi]$

```
>plot2d("arctan(x)", -pi, pi):
```



Sebagai alternatif dari titik dua, ada perintah ``insimg(lines)`` yang menyisipkan plot dengan menempati sejumlah baris teks yang ditentukan.

Dalam opsi, plot dapat diatur untuk muncul:

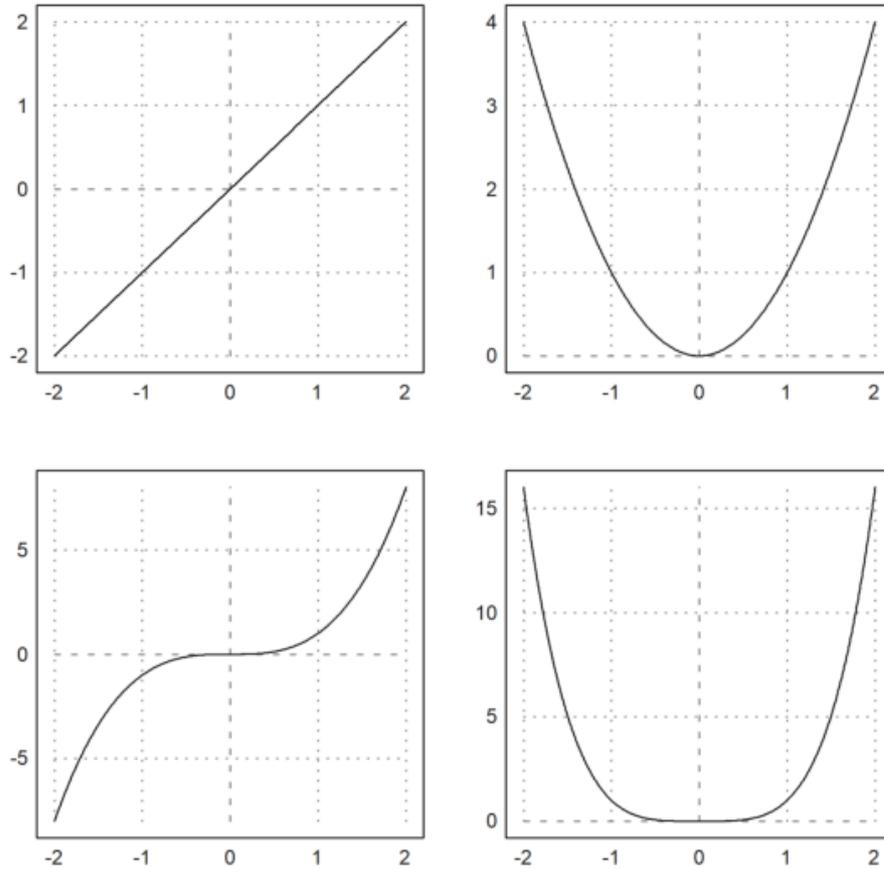
- di jendela terpisah yang dapat diubah ukurannya,
- di jendela notebook.

Lebih banyak gaya dapat dicapai dengan menggunakan perintah plot yang spesifik.

Jika grafik tersembunyi, tekan tombol tabulator untuk melihatnya.

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)`` mengembalikan jendela default.

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

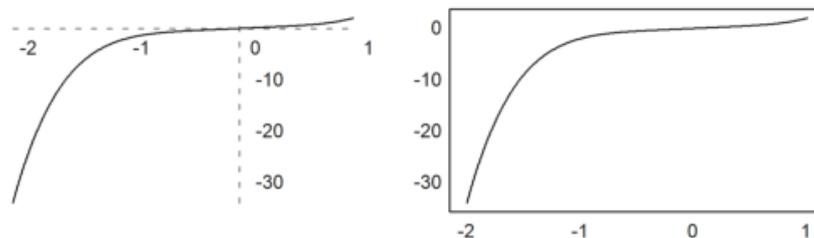


catatan: hasil dari perintah tersebut adalah jendela grafis yang dibagi menjadi 4 subplot dengan grid 2x2, disetiap subplot merupakan hasil dari fungsi yang berbeda-beda dari  $x^n$  dengan  $n=1$  sampai 4. setelah semua plot selesai, jendela grafis akan kembali ke mode default atau satu jendela.

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

Anda dapat mengganti nilai `x` dengan angka lain untuk melihat variasi gaya grid yang berbeda.

```
>figure(1,1); ...
for m=3 to 4; figure(m); plot2d("x^5+x",-2,1,grid=m); end; ...
figure(0):
```

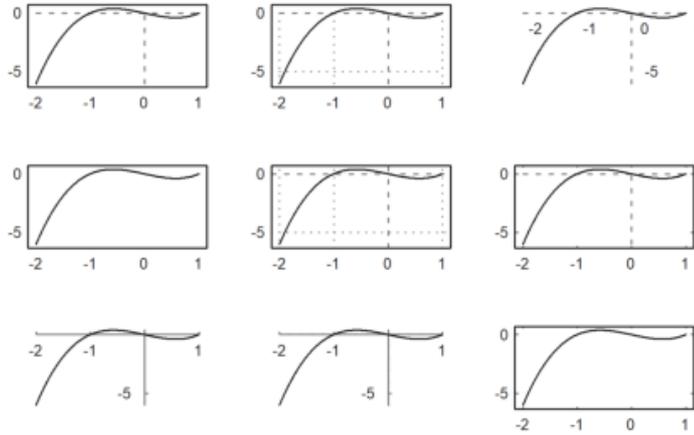


Illegal start value for loop!

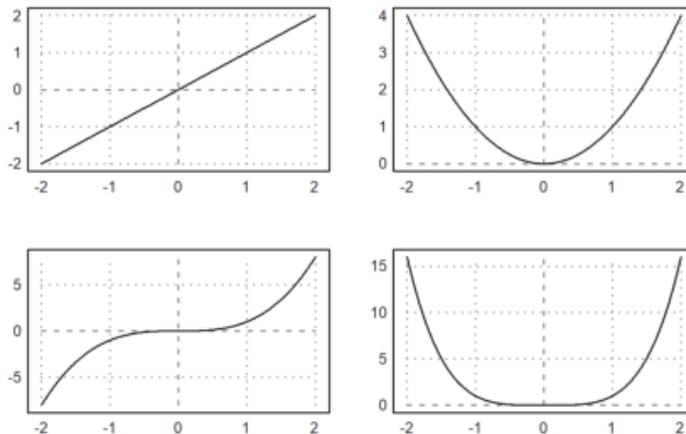
Error in:

```
figure(4,4); for k=:9; figure(k); plot2d("x^3-x",-2,1,grid=k); ...
```

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



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

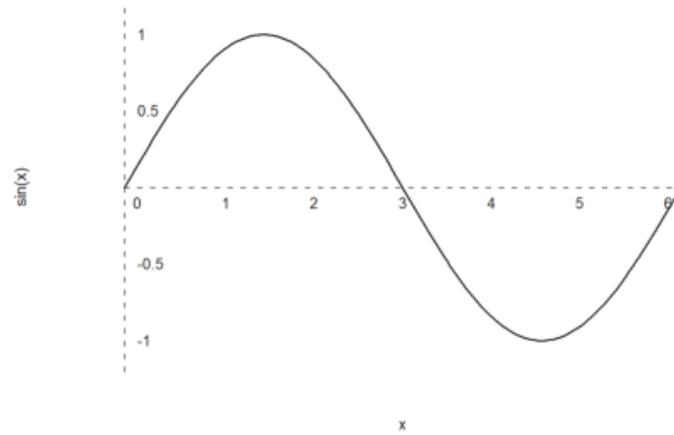


Jika argumen untuk `plot2d()` berupa ekspresi yang diikuti oleh empat angka, angka-angka tersebut adalah batas rentang x dan y untuk grafik.

Sebagai alternatif, nilai a, b, c, dan d dapat ditentukan sebagai parameter yang ditetapkan, seperti `a=...` dan seterusnya.

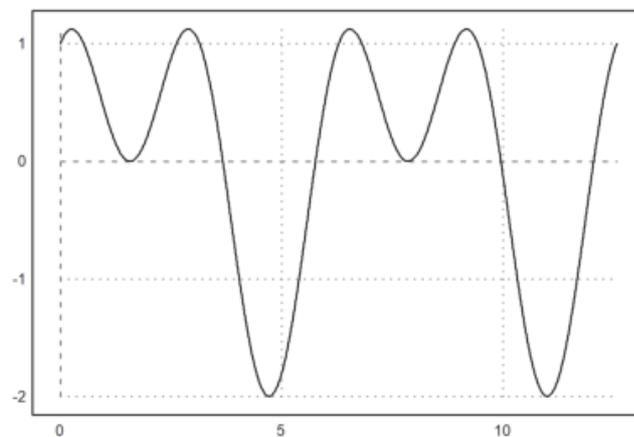
Dalam contoh berikut, kita mengubah gaya grid, menambahkan label, dan menggunakan label vertikal untuk sumbu y.

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



catatan: ini merupakan plot  $\sin(x)$  dari 0 hingga  $2\pi$  dengan grid, yang berfungsi untuk membantu pembacaan nilai-nilai penting pada sumbu. rasio yang diterapkan adalah 1.5 sehingga plot tersebut lebih lebar dari pada tinggi.

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



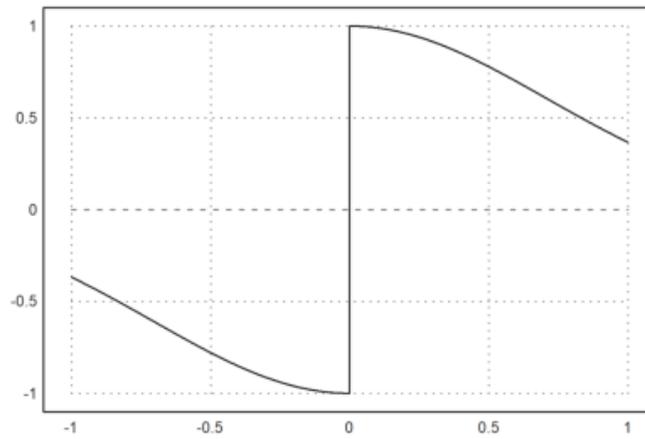
catatan: plot tersebut menampilkan grafik dari fungsi sinus dan kosinus yang saling berinterferensi satu sama lain sehingga menghasilkan pola gelombang.

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

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

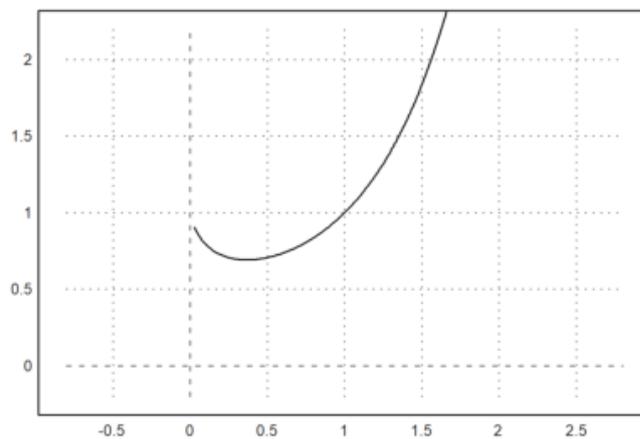
Fungsi atau ekspresi dalam `plot2d` dievaluasi secara adaptif. Untuk meningkatkan kecepatan, matikan plot adaptif dengan `` dan tentukan jumlah subinterval dengan `n=...`. Ini hanya perlu dilakukan dalam kasus yang jarang terjadi.

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



catatan: pada plot tersebut kurva  $x > 0$  akan berada di atas sumbu  $x$ , sedangkan kurva  $x < 0$  akan berada di bawah sumbu  $x$ , plot tersebut akan sangat halus dikarenakan penggunaan hingga 10.000 titik

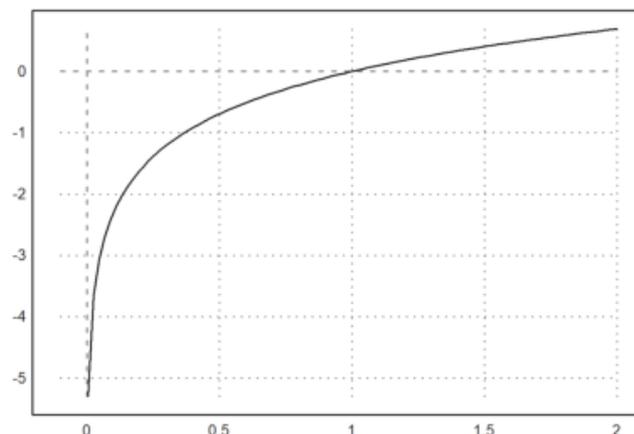
```
>plot2d("x^x", r=1.2, cx=1, cy=1):
```



catatan: plot tersebut menunjukkan fungsi  $x^x$  yang meningkat dengan cepat untuk  $x > 1$ . fungsi ini tidak terdefinisi untuk nilai  $x = 0$  sehingga hanya muncul di kuadran 1.

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

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



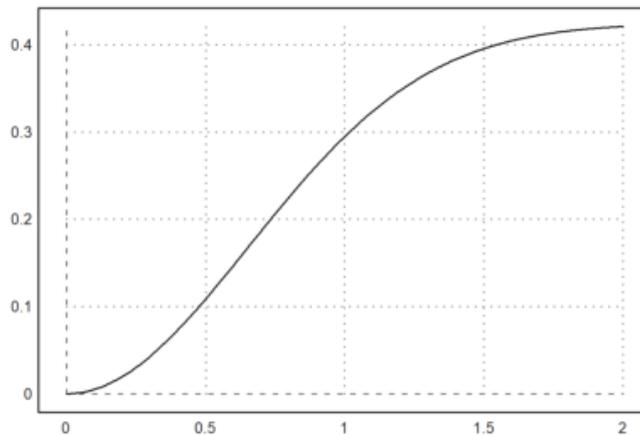
catatan: pada fungsi  $\log(x)$  hasil tidak terdefinisi di wilayah  $x < 0$ , sehingga tidak ada grafik di wilayah tersebut. hanya di wilayah  $x > 0$  yang memuat grafik.

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

```
plot2d("log(x)",-0.1,2):>plot2d("x^3-x",>square):
Function plot2d not found.
Try list ... to find functions!
Syntax error in expression, or unfinished expression!
Error in:
plot2d("log(x)",-0.1,2):>plot2d("x^3-x",>square): ...
^
```

catatan: fungsi tersebut terdiri menjadi 2 bagian, yakni ploty logaritma yang hanya akan menampilkan bagian positif untuk  $\log(x)$ . yang kedua adalah plot kubik yang menampilkan fungsi  $x^3-x$  yang memungkinkan untuk melihat karakteristik dan titik potong dengan lebih jelas.

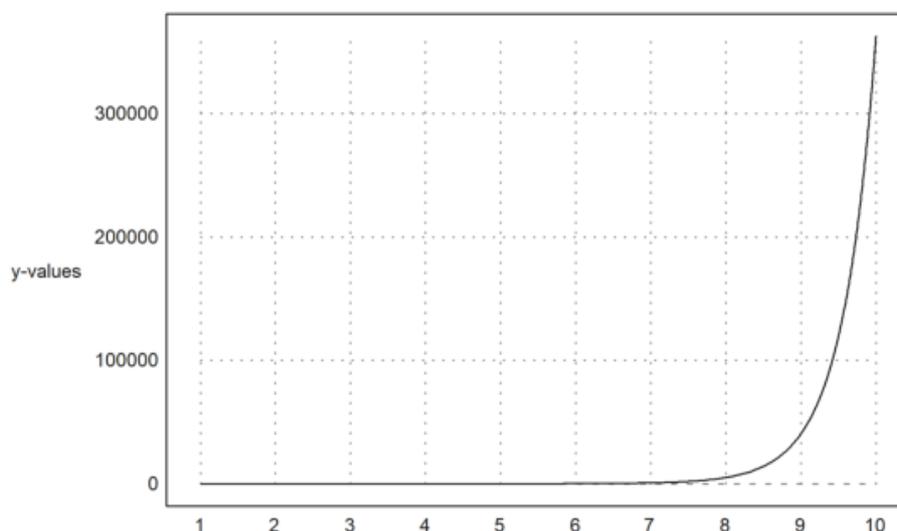
```
>plot2d('integrate("sin(x)*exp(-x^2)",0,x)',0,2): // plot integral
```



catatan: fungsi tersebut menghasilkan integral yang bergantung pada x, dengan rentang x dari 0 hingga 2.

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()`.

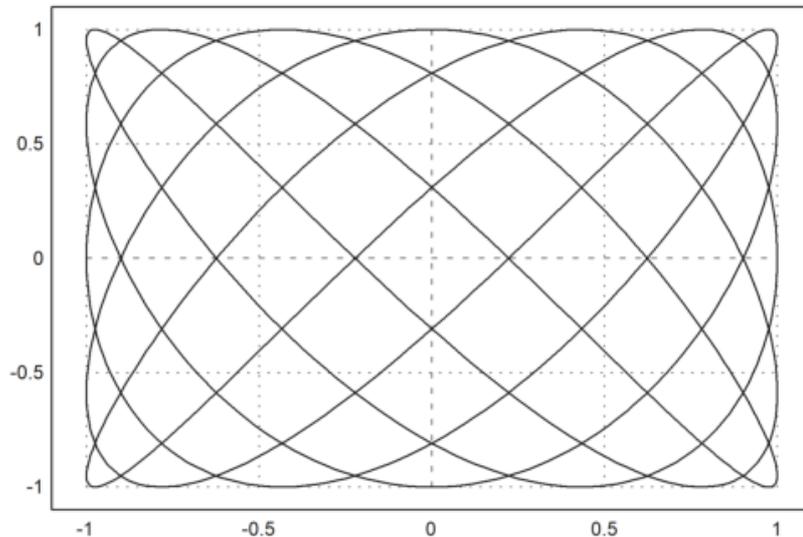
```
>plot2d("gamma(x)",1,10,yl="y-values",smaller=6,<vertical):
```



catatan: fungsi gamma merupakan perluasan dari faktorial ke bilangan real dan kompleks dengan rentang 1 sampai 10. dengan penambahan label sumbu-y memberikan penjelasan yang lebih informatif. dengan memperkecil ukuran jendela yakni smaller=6 untuk ruang sumbu y dan label.

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

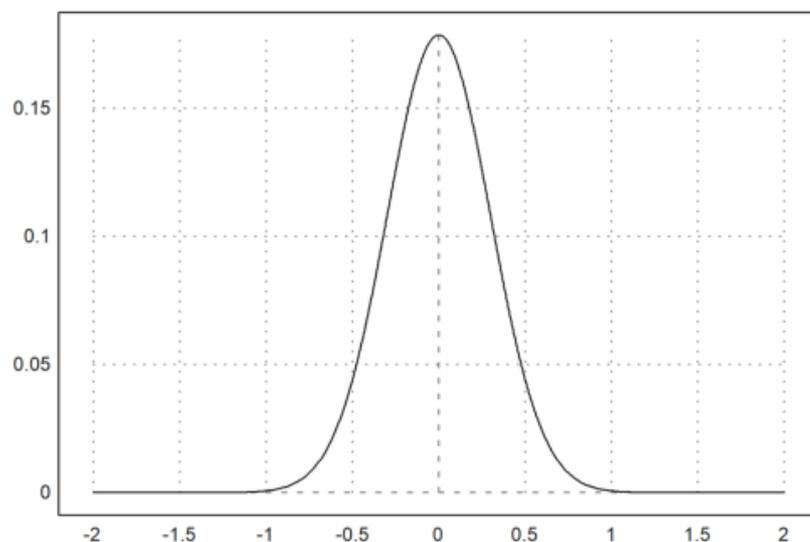
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



catatan: Grafik akan menampilkan kurva Lissajous yang terbentuk dari dua frekuensi sinusoidal yang berbeda. Jika frekuensi keduanya adalah bilangan bulat, pola yang dihasilkan akan memiliki simetri tertentu.

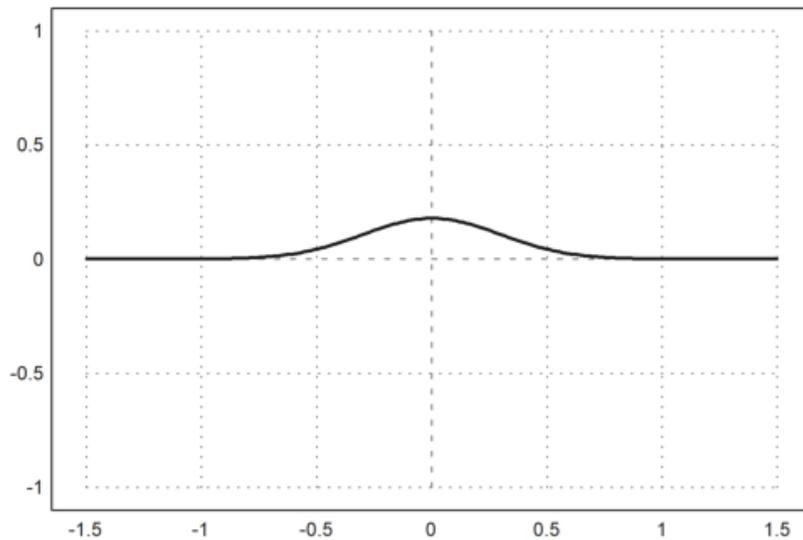
Ini adalah cara yang efektif untuk menggambarkan osilasi dua dimensi yang dipengaruhi oleh dua frekuensi berbeda.

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



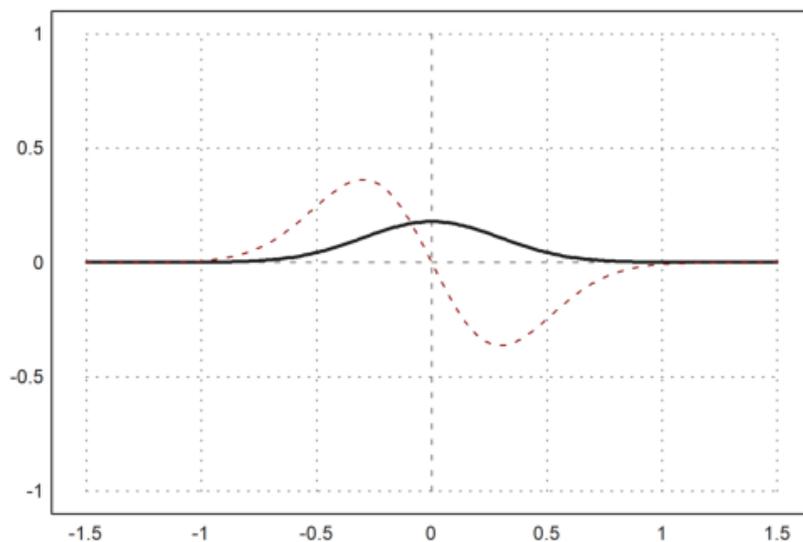
catatan: Grafik yang dihasilkan adalah kurva normal, yang berpusat di  $x=0$  dan mengecil secara eksponensial ke arah kiri dan kanan.

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



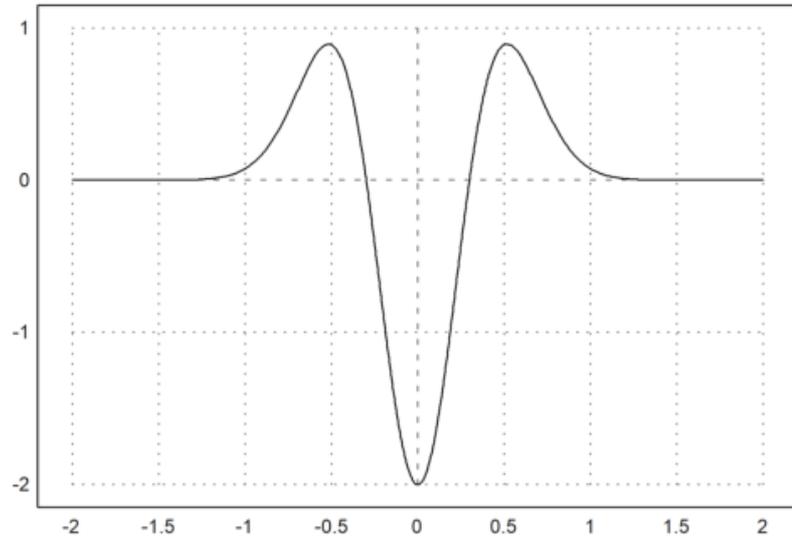
catatan: dengan perintah tersebut kurva normal digambarkan dengan ketebalan garis yang lebih besar.

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



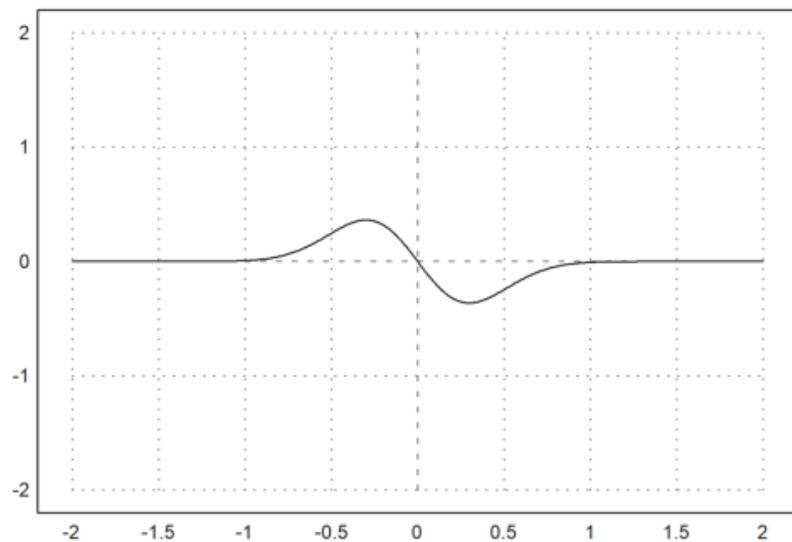
catatan:dengan perintah tersebut menghasilkan plot turunan dari kurva normal yang telah didefinisikan sebelumnya, dengan garis putus-putus berwarna merah.Jika sebelumnya ada plot fungsi yang sama, turunan akan ditampilkan di atasnya.

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

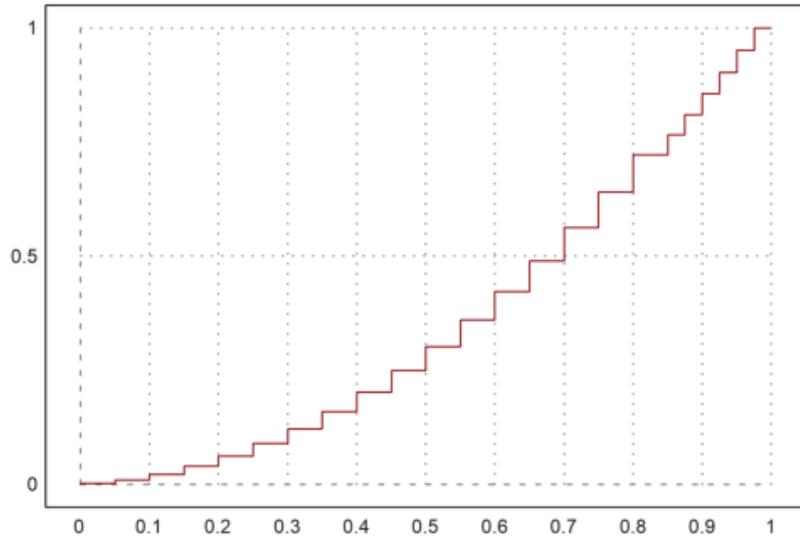


catatan:dengan perintah tersebut menghasilkan plot turunan dari kurva normal yang telah didefinisikan sebelumnya, dalam rentang x dari -2 hingga 2 dan y dari -2 hingga 1. Ini berguna untuk menganalisis perilaku fungsi dan sifat kekonveksan atau kekonkavan dari fungsi tersebut.

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

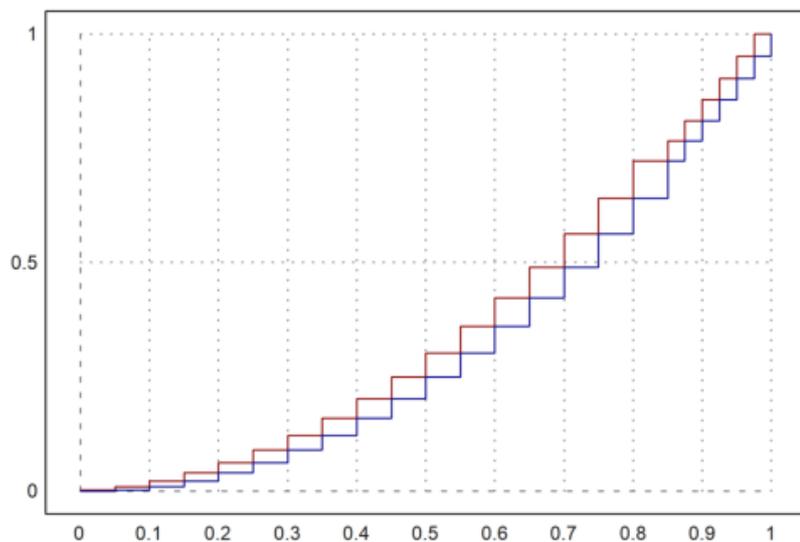


```
>plot2d("x^2",0,1,steps=1,color=red,n=10):
```



catatan: Perintah ini akan menghasilkan plot dari fungsi kuadrat  $x^2$  pada interval  $[0, 1]$  dengan garis berwarna merah dan evaluasi yang halus berkat pengaturan langkah yang rendah.

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



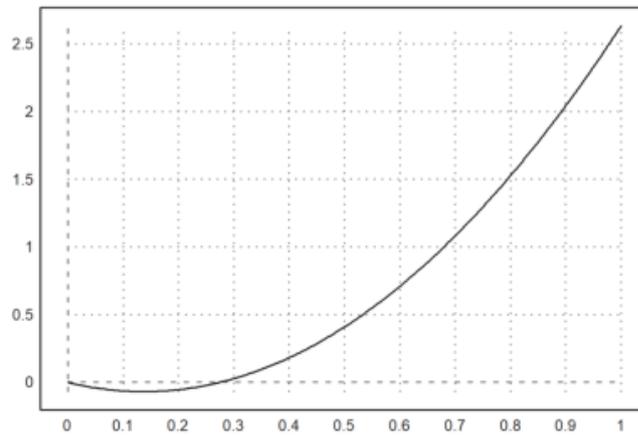
catatan: Perintah ini akan menghasilkan plot dari fungsi kuadrat  $x^2$  yang ditambahkan ke plot sebelumnya (jika ada) dengan garis berwarna biru dan evaluasi yang dilakukan dengan langkah 2, menghasilkan tampilan yang lebih kasar.

## Fungsi dengan Satu Parameter

Fungsi plotting yang paling penting untuk grafik planar adalah `plot2d()`. Fungsi ini diimplementasikan dalam bahasa Euler di dalam file "plot.e", yang dimuat pada awal program.

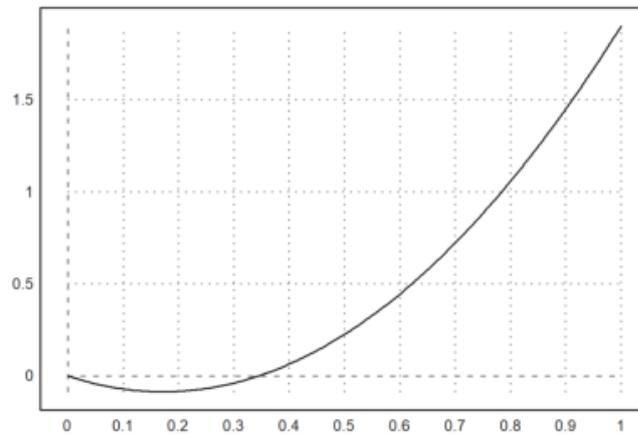
Berikut beberapa contoh menggunakan sebuah fungsi. Seperti biasa di EMT, fungsi yang bekerja untuk fungsi atau ekspresi lain dapat menerima parameter tambahan (selain  $x$ ) yang bukan variabel global ke dalam fungsi, baik dengan parameter titik koma atau dengan koleksi panggilan.

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



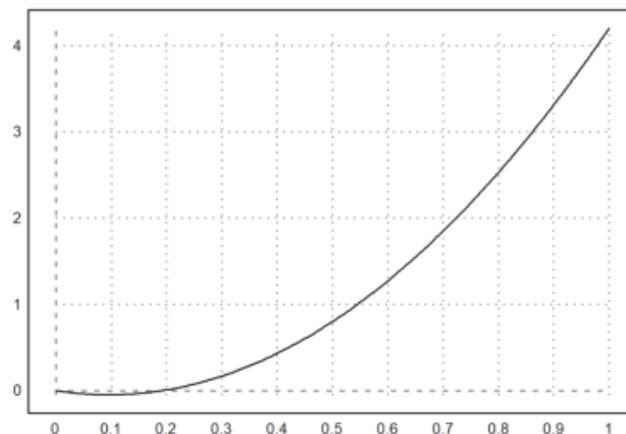
catatan: Perintah ini akan menghasilkan plot dari fungsi  $f(x,0.3)$  di rentang dari 0 hingga 1. Fungsi tersebut akan ditampilkan dengan menggunakan nilai yang telah ditetapkan.

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

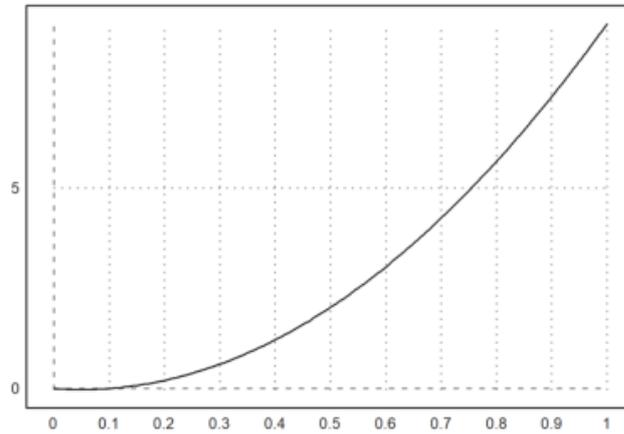


catatan: perintah ini menghasilkan plot dari fungsi  $f(x,0.4)$  dengan rentang 0 hingga 1 dengan a diset ke 0.4

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

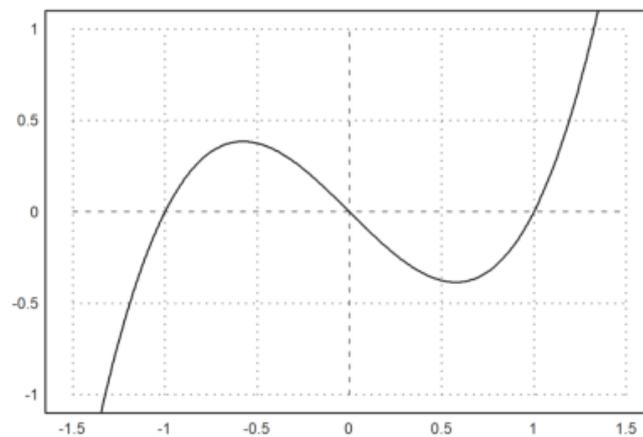


```
>plot2d({"f(x,b)",b=0.1},0,1): // plot with 0.1
```



catatan: Perintah ini akan menghasilkan plot dari fungsi  $(x,0.1)$  dalam rentang dari 0 hingga 1. Dengan parameter diset ke 0.1.

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



catatan: Plot ini akan menunjukkan grafik dari fungsi  $(x)=x^3-x$  dengan skala tertentu, berdasarkan nilai  $r$ , yakni 1.

Berikut adalah ringkasan fungsi yang diterima:

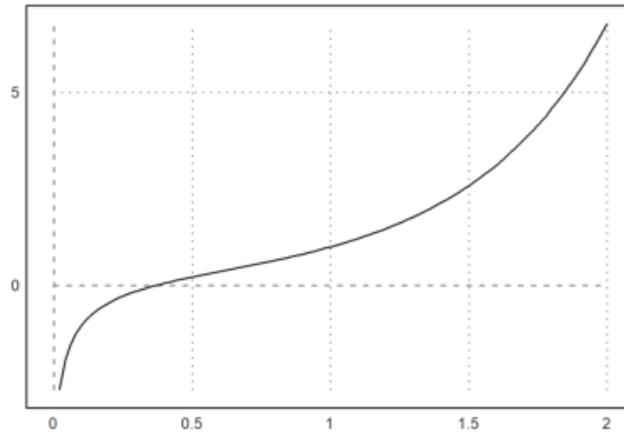
- Ekspresi atau ekspresi simbolik dalam  $x$
- Fungsi atau fungsi simbolik dengan nama seperti "f"
- Fungsi simbolik hanya dengan nama  $f$

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

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

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

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



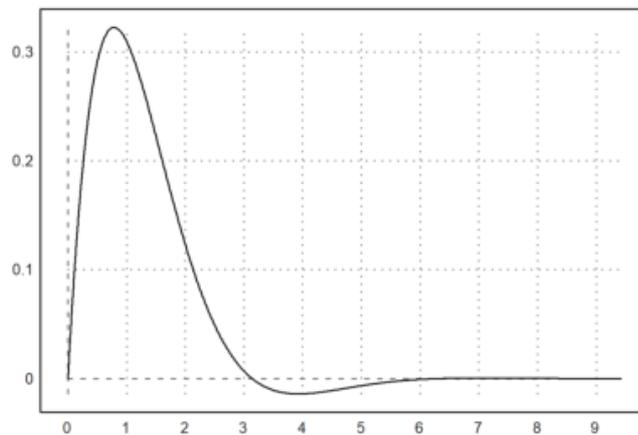
catatan: grafik ini menunjukkan nilai dari turunan fungsi tersebut dengan rentang x dari 0 hingga 2.

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

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

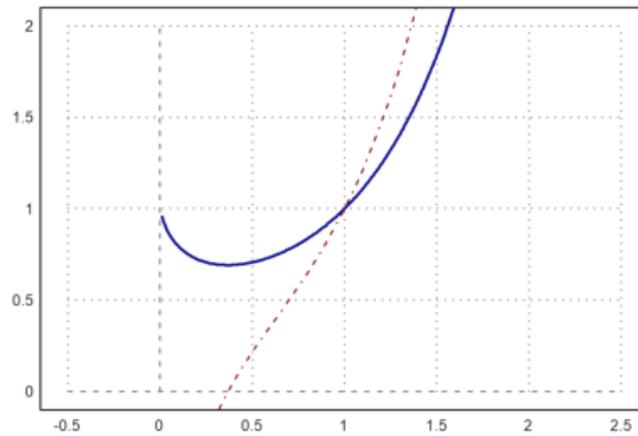
$$E^{-x} \sin(x)$$

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



catatan: Grafik yang dihasilkan akan menunjukkan gelombang sinus yang berkurang amplitudonya seiring dengan bertambahnya nilai x, karena faktor  $E^{-x}$  mengurangi nilai fungsi sinus di rentang tersebut.

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



catatan: Grafik yang dihasilkan akan menampilkan kurva  $x$  dalam warna biru dan turunan dari fungsi tersebut dalam warna merah dengan gaya garis putus-putus

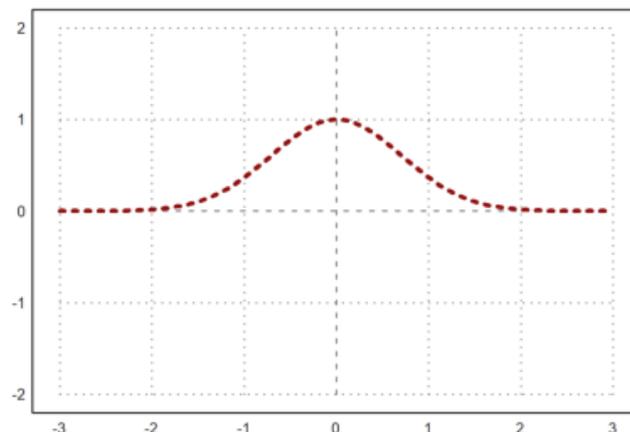
Untuk gaya garis, terdapat berbagai opsi:

- ``style="..."``: Pilih dari "-", "--", "-.", ".", "-.-", "-.-.-".
- ``color``: Lihat di bawah untuk pilihan warna.
- ``thickness``: Default adalah 1.

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

- 0..15: indeks warna default.
- Konstanta warna: putih, hitam, merah, hijau, biru, cyan, zaitun, abu-abu terang, abu-abu, abu-abu gelap, oranye, hijau muda, turquoise, biru muda, oranye muda, kuning.
- ``rgb(red, green, blue)``: parameter adalah bilangan riil dalam rentang [0,1].

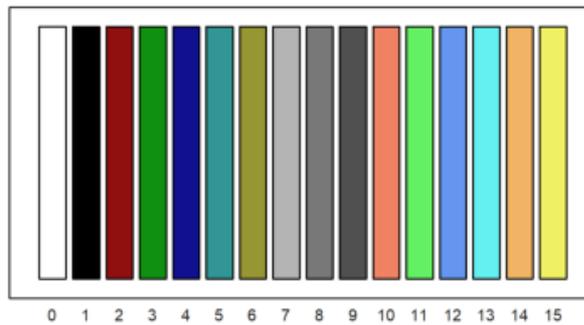
```
>plot2d("exp(-x^2)", r=2, color=red, thickness=3, style="--"):
```



catatan: Grafik yang dihasilkan akan menampilkan kurva  $e^{-x^2}$  dalam warna merah, dengan garis putus-putus dan ketebalan yang cukup besar,

Berikut adalah tampilan warna yang sudah ditentukan di EMT.

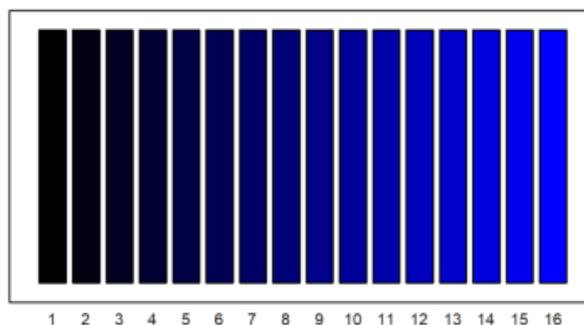
```
>aspect(2); columnplot(ones(1,16), lab=0:15, grid=0, color=0:15):
```



catatan: Hasil dari perintah ini adalah plot kolom yang memiliki 16 kolom, masing-masing diberi label dari 0 hingga 15. Setiap kolom akan berwarna berbeda, dan tidak ada garis grid di belakang.

Namun, Anda dapat menggunakan warna apa pun.

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

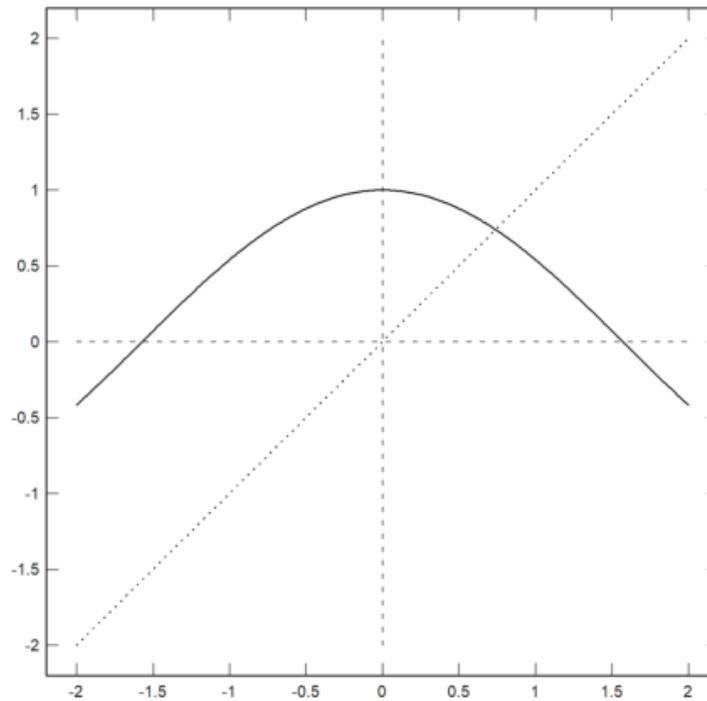


catatan: Hasil dari perintah ini adalah plot kolom dengan 16 kolom, di mana setiap kolom akan memiliki warna biru yang semakin cerah dari kiri ke kanan. Tidak ada garis grid di belakang.

## Menggambar Beberapa Kurva pada Bidang Koordinat yang Sama

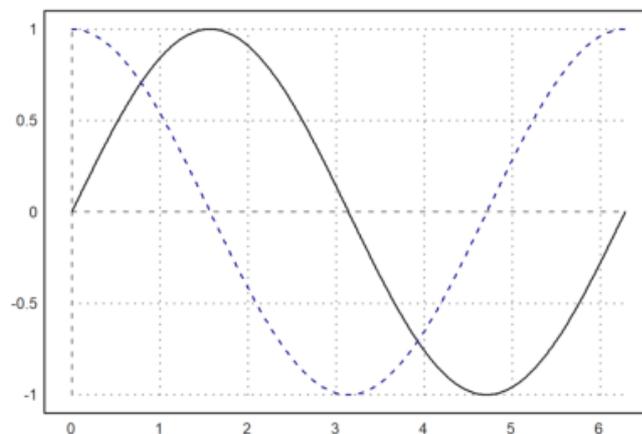
Plot lebih dari satu fungsi (fungsi ganda) dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu metodenya adalah menggunakan `>add` untuk beberapa panggilan ke `plot2d`, kecuali pada panggilan pertama. Kita sudah menggunakan fitur ini dalam contoh-contoh di atas.

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



catatan: Hasil dari perintah ini adalah sebuah grafik yang menampilkan fungsi  $\cos(x)$  dan fungsi linear  $x$  di dalam satu plot. Fungsi  $\cos(x)$  akan digambar sebagai garis kontinyu, sedangkan fungsi  $x$  akan ditampilkan sebagai titik-titik pada grafik, memberikan perbandingan visual antara keduanya. Dengan pengaturan grid, pengguna dapat lebih mudah menganalisis nilai-nilai pada grafik.

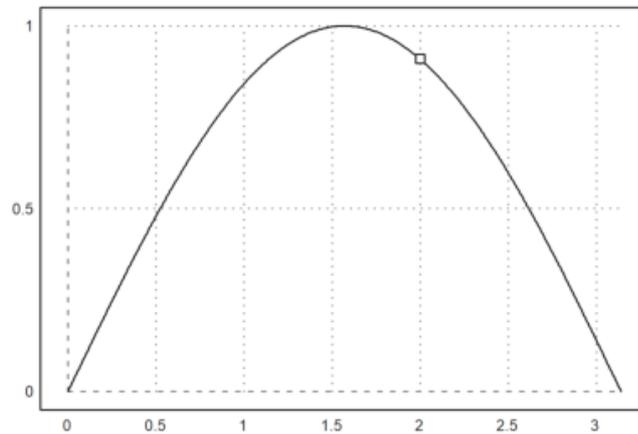
```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



catatan: Hasil dari perintah ini adalah sebuah grafik yang menampilkan kedua fungsi  $\sin(x)$  dan  $\cos(x)$  dalam satu plot. Grafik  $\sin(x)$  akan terlihat sebagai garis kontinyu, sedangkan grafik  $\cos(x)$  akan terlihat sebagai garis putus-putus berwarna biru. Aspek rasio yang diatur menjadi 1.5 memberikan tampilan yang lebih lebar, membantu dalam visualisasi hubungan antara kedua fungsi tersebut

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

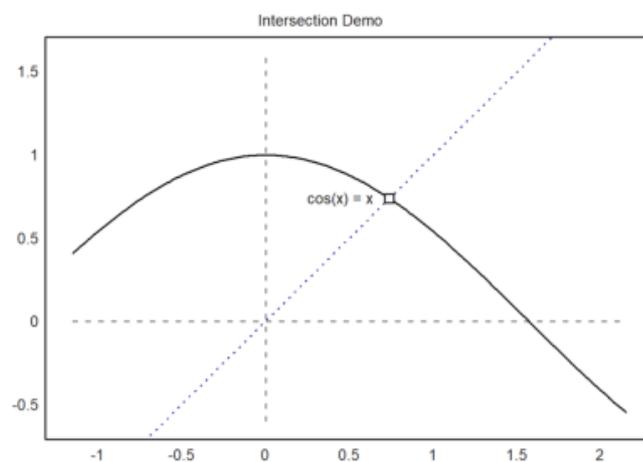
```
>plot2d("sin(x)",0,pi); plot2d(2,sin(2),>points,>add):
```



catatan: Hasil dari perintah ini adalah grafik  $\sin(x)$  untuk interval 0 hingga  $\pi$  dengan sebuah titik ditambahkan pada koordinat  $(2, \sin(2))$ . Titik ini akan terlihat jelas pada grafik sebagai titik terpisah dari garis sinus, memberikan visualisasi yang jelas tentang nilai  $\sin$  pada  $x = 2$ .

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

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



catatan: Hasil dari perintah ini adalah plot dari dua fungsi  $\cos(x)$  dan  $x$  dengan titik interseksi ditandai. Judul "Intersection Demo" menunjukkan bahwa kita sedang mendemonstrasikan titik di mana kedua fungsi ini bertemu. Label pada titik interseksi memberikan konteks tentang apa yang ditunjukkan oleh grafik tersebut.

Dalam demo berikut, kita memplot fungsi  $\text{sinc}(x) = \frac{\sin(x)}{x}$  dan perluasan Taylor ke-8 dan ke-16. Kita menghitung perluasan ini menggunakan Maxima melalui ekspresi simbolik.

Plot ini dilakukan dalam perintah multiline berikut dengan tiga panggilan ke `plot2d()`. Panggilan kedua dan ketiga memiliki bendera `>add` yang diatur, sehingga plot tersebut menggunakan rentang yang sebelumnya.

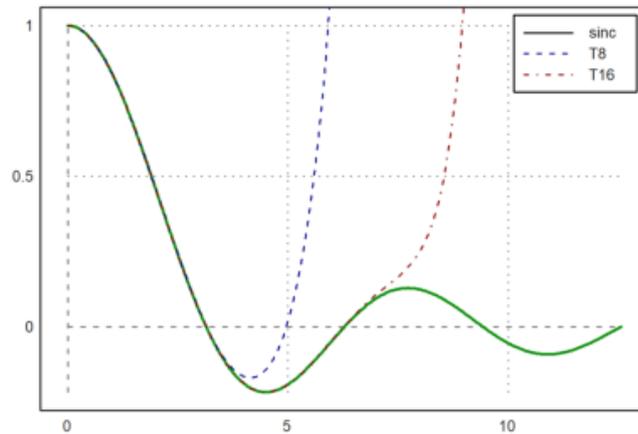
Kita juga menambahkan kotak label yang menjelaskan fungsi-fungsi tersebut.

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

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

```
>plot2d("sinc(x)", 0, 4pi, color=green, thickness=2); ...
plot2d(&taylor(sin(x)/x, x, 0, 8), >add, color=blue, style="--"); ...
```

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

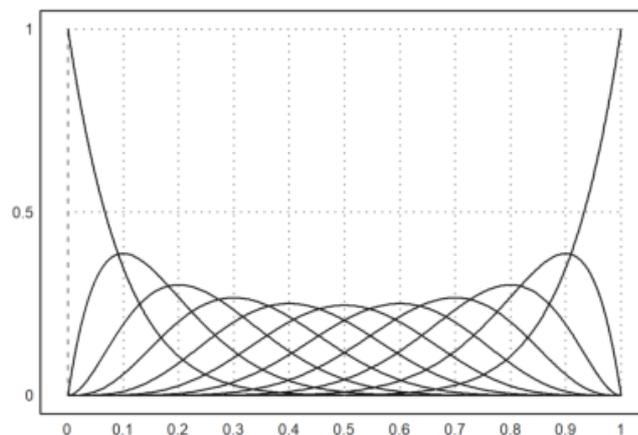


catatan :Hasil dari perintah ini adalah plot yang menampilkan fungsi  $\text{sinc}(x)$  dengan dua deret Taylor (derajat 8 dan 16). Kotak label di sudut plot menjelaskan kurva mana yang merepresentasikan fungsi asli dan kurva mana yang merupakan pendekatan deret Taylor. Ini memberikan gambaran visual yang jelas tentang seberapa baik deret Taylor mendekati fungsi asli dalam rentang yang diberikan.

dalam contoh berikut, kita menghasilkan Polinom Bernstein.

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

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



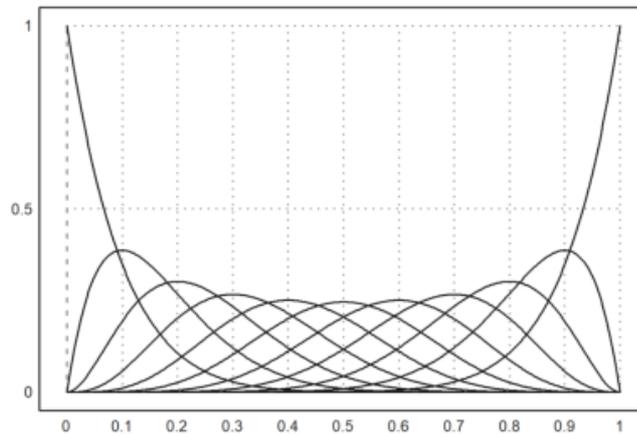
catatan: Hasil dari perintah ini adalah grafik yang menampilkan fungsi  $(1-x)^{10}$  dan sepuluh fungsi Bernstein yang terkait. Ini akan memberikan gambaran yang jelas tentang bagaimana fungsi-fungsi tersebut berkontribusi dalam mendekati bentuk polinomial dan bagaimana mereka berinteraksi satu sama lain di dalam rentang yang ditentukan.

Metode kedua adalah menggunakan sepasang matriks nilai  $x$  dan matriks nilai  $y$  dengan ukuran yang sama.

Kita menghasilkan matriks nilai dengan satu Polinom Bernstein di setiap baris. Untuk ini, kita cukup menggunakan vektor kolom dari  $i$ . Silakan lihat pengantar tentang bahasa matriks untuk mempelajari lebih lanjut tentang detailnya.

```
>x=linspace(0,1,500);
>n=10; k=(0:n)'; // n is row vector, k is column vector
```

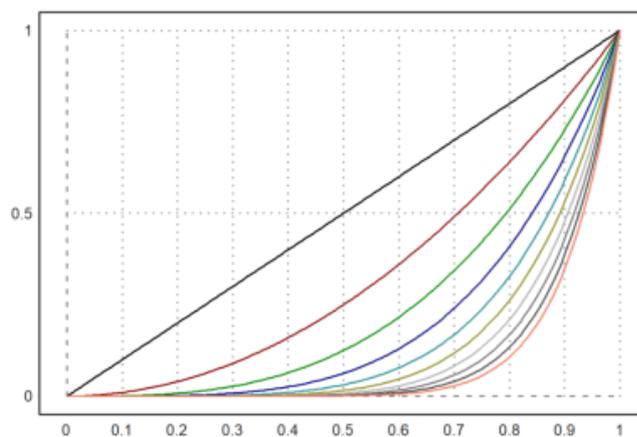
```
>y=bin(n,k)*x^k*(1-x)^(n-k); // y is a matrix then
>plot2d(x,y):
```



catatan: Hasil dari perintah ini adalah grafik yang menunjukkan polinomial Bernstein untuk  $n=10$  dalam rentang dari 0 hingga 1. Setiap fungsi akan memiliki bentuk yang khas, memperlihatkan bagaimana polinomial Bernstein berkonvergensi dan saling berinteraksi dalam rentang tersebut.

Perhatikan bahwa parameter warna dapat berupa vektor. Dalam hal ini, setiap warna akan digunakan untuk setiap baris matriks.

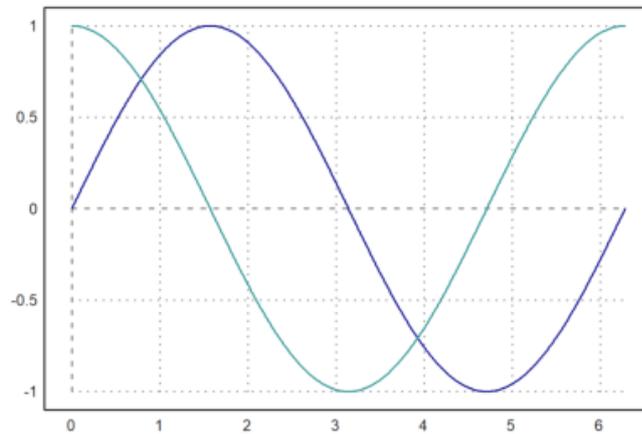
```
>x=linspace(0,1,200); y=x^(1:10)'; plot2d(x,y,color=1:10):
```



catatan: Hasil dari perintah ini adalah grafik yang menunjukkan kurva untuk  $y = x^1$ ,  $y = x^2$ , hingga  $y = x^{10}$  dalam rentang  $x$  dari 0 hingga 1. Setiap kurva akan memiliki warna yang berbeda, memberikan visualisasi yang jelas tentang bagaimana fungsi ini berperilaku dalam interval tersebut.

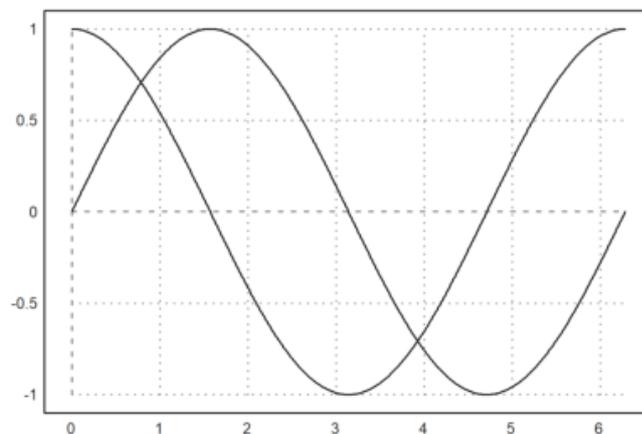
Metode lainnya adalah dengan menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan array warna, array gaya, dan array ketebalan dengan panjang yang sama.

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



catatan: Hasil dari perintah ini adalah grafik yang menampilkan kedua kurva  $\sin(x)$  dan  $\cos(x)$  dalam rentang  $x$  dari 0 hingga  $2\pi$ , dengan warna yang berbeda untuk masing-masing fungsi. Ini memungkinkan perbandingan visual yang jelas antara dua fungsi trigonometri tersebut.

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



catatan: Hasil dari perintah ini adalah grafik yang menampilkan kedua fungsi  $\sin(x)$  dan  $\cos(x)$  dalam rentang  $x$  dari 0 hingga  $2\pi$ . Keduanya akan digambar pada plot yang sama, memungkinkan perbandingan visual antara kedua fungsi trigonometri tersebut.

Kita dapat mendapatkan vektor seperti itu dari Maxima menggunakan ``makelist()`` dan ``mxm2str()``.

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

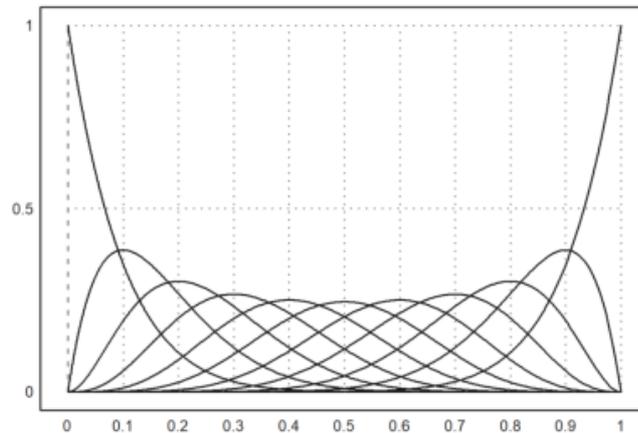
$$\left[ (1-x)^{10}, 10(1-x)^9x, 45(1-x)^8x^2, 120(1-x)^7x^3, 210(1-x)^6x^4, 252(1-x)^5x^5, 210(1-x)^4x^6, 120(1-x)^3x^7, 45(1-x)^2x^8, 10(1-x)x^9, x^{10} \right]$$

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

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

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

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



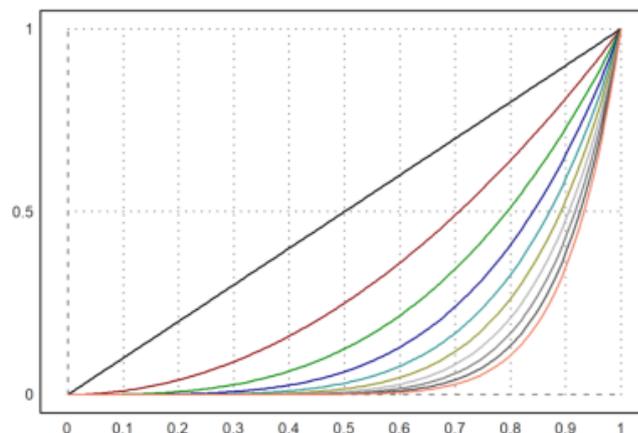
catatan: Perintah ini akan menghasilkan plot dari semua fungsi yang terdapat dalam `v` di sepanjang sumbu x dari 0 hingga 1. Pastikan bahwa `v` telah didefinisikan sebelumnya sebagai vektor yang berisi fungsi yang valid untuk di-plot.

Alternatif lainnya adalah menggunakan bahasa matriks dari Euler.

Jika sebuah ekspresi menghasilkan matriks fungsi, dengan satu fungsi di setiap baris, semua fungsi ini akan dipplot dalam satu grafik.

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

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



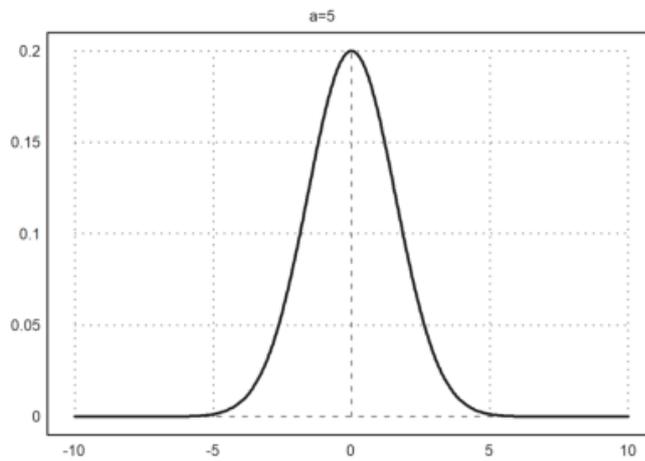
catatan: Perintah ini akan menghasilkan grafik yang menampilkan fungsi  $x^1$ ,  $x^2$ , ...,  $x^{10}$  dalam rentang x dari 0 hingga 1, dengan setiap kurva memiliki warna yang berbeda.

Ekspresi dan fungsi satu-lini dapat mengakses variabel global.

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

Pastikan untuk menempatkan semua parameter yang ditetapkan di akhir perintah `plot2d`. Dalam contoh ini, kita melewati  $a=5$  ke fungsi  $f$ , yang 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"):
```

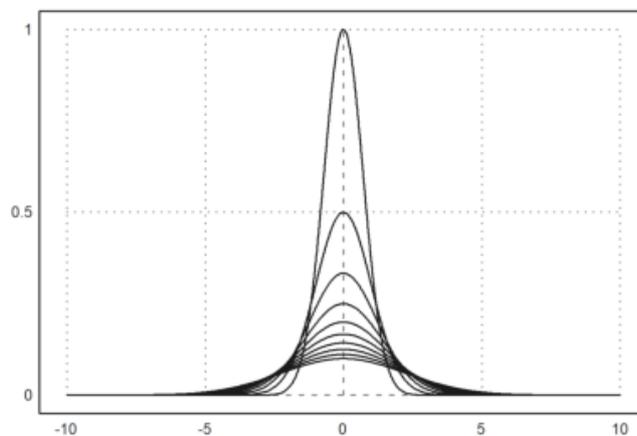


catatan: Perintah ini akan menghasilkan grafik dari fungsi Gaussian dengan parameter  $a = 5$ , menunjukkan bentuk fungsi dalam rentang  $x$  dari  $-10$  hingga  $10$ .

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

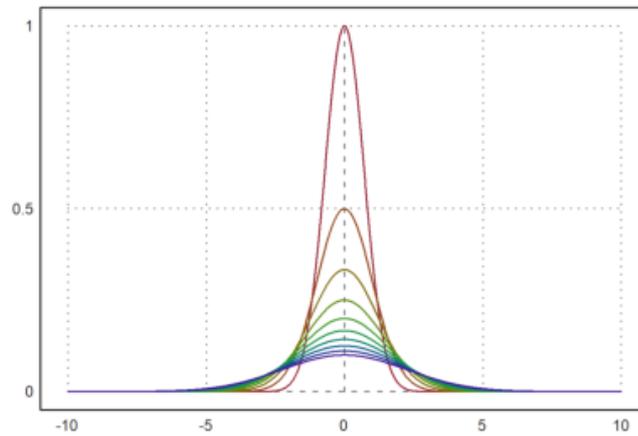
Dalam contoh berikut, kita menggunakan loop untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman untuk loop).

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



Kita bisa mencapai hasil yang sama dengan cara berikut menggunakan bahasa matriks dari EMT. Setiap baris dari matriks  $f(x,a)$  adalah satu fungsi. Selain itu, kita dapat mengatur warna untuk setiap baris matriks. Klik dua kali pada fungsi `getspectral()` untuk penjelasan lebih lanjut.

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



catatan: Perintah ini akan menghasilkan grafik dari fungsi  $f$  yang diukur di seluruh rentang  $x$  dari -10 hingga 10, dengan 10 kurva berbeda untuk nilai  $a$  dari 1 hingga 10, masing-masing dengan warna yang berbeda berdasarkan spektrum.

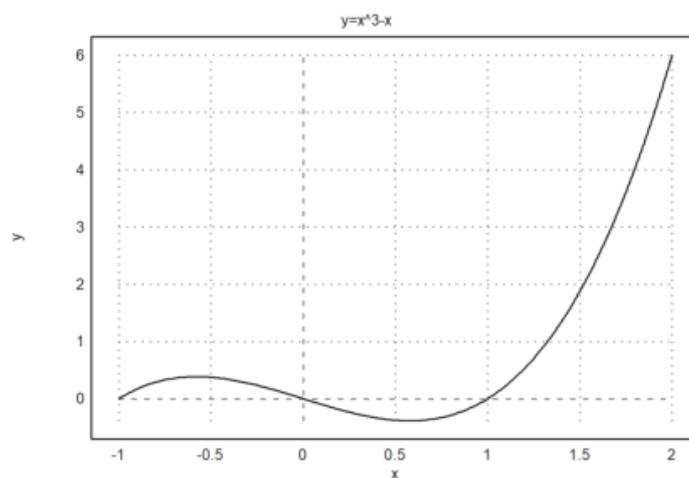
## Label Teks

Dekorasi sederhana dapat berupa:

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

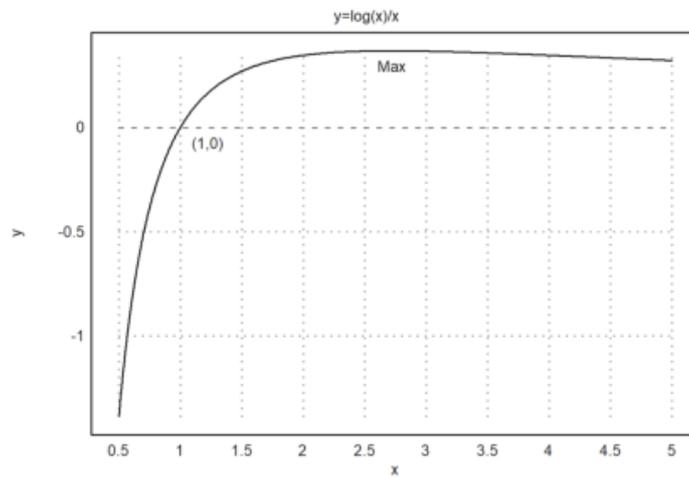
Perintah ``label`` akan memplot ke grafik saat ini pada koordinat `plot(x,y)`. Ini dapat menerima argumen posisi.

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



catatan: Perintah ini akan menghasilkan grafik dari fungsi  $y = x^3 - x$  dengan rentang yang ditentukan, lengkap dengan judul dan label sumbu yang jelas. Grafik ini akan menunjukkan bentuk fungsi kubik, termasuk titik belok dan titik potong sumbu.

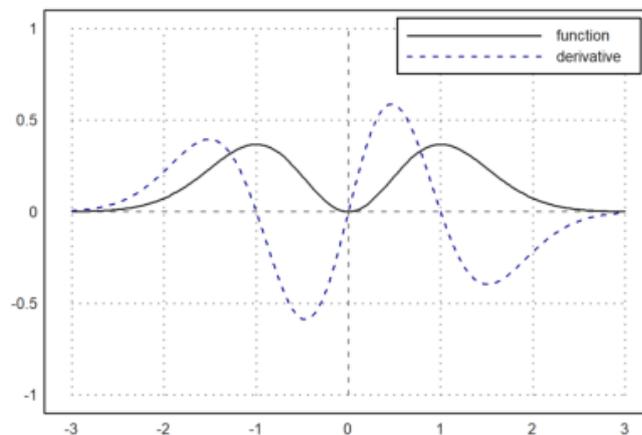
```
>expr := "log(x)/x"; ...
plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```



catatan: Hasil dari perintah ini akan menunjukkan grafik fungsi  $y = \frac{\log(x)}{x}$  dengan titik maksimum yang ditandai di lokasi  $(E, \frac{\log(E)}{E})$  dan titik lainnya di  $(1,0)$ . Label akan memberikan informasi yang jelas tentang titik-titik penting pada grafik.

Ada juga fungsi `labelbox()`, yang dapat menampilkan fungsi-fungsi dan teks. 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):
```



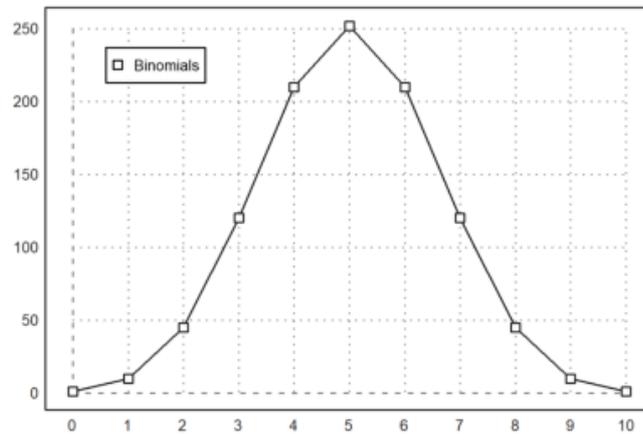
catatan: Hasil dari perintah ini akan menampilkan grafik dari  $f(x) = x^2 e^{-x^2}$  beserta grafik turunan fungsi tersebut. Label yang ditambahkan akan membantu membedakan antara grafik fungsi dan grafik turunan, memberikan konteks yang jelas bagi pemirsa.

Kotak label secara default terikat di sudut kanan atas, tetapi `>left` mengikatnya di sudut kiri atas. Anda dapat memindahkannya ke posisi mana pun yang Anda inginkan. Posisi jangkar adalah sudut kanan atas kotak, dan angka-angka adalah fraksi dari ukuran jendela grafik. Lebar kotak diatur secara otomatis.

Untuk plot titik, kotak label juga berfungsi. Tambahkan parameter `>points`, atau vektor bendera, satu untuk setiap label.

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

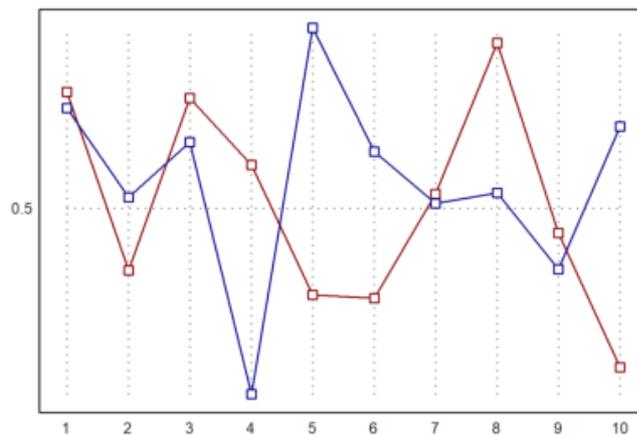
```
>n=10; plot2d(0:n,bin(n,0:n),>addpoints); ...
labelbox("Binomials",styles="[]",>points,x=0.1,y=0.1, ...
tcolor=black,>left):
```



catatan: Hasil dari perintah ini akan menampilkan grafik dari koefisien binomial  $\binom{n}{k}$  untuk  $k = 0, 1, 2, \dots, 10$ . Kotak label akan muncul di sudut grafik, memberikan informasi tentang isi grafik, yaitu binomial.

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

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

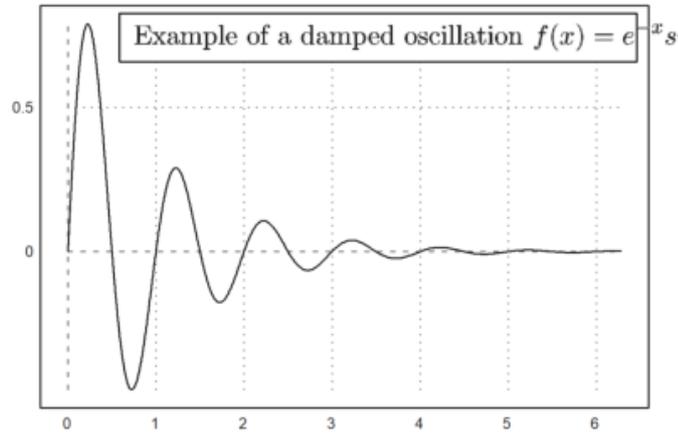


catatan: Hasilnya adalah grafik yang memplot dua set data acak dengan rentang sumbu x dari 1 hingga 10. Baris pertama akan diplot dengan warna merah. Baris kedua akan diplot dengan warna biru.

Fitur serupa adalah fungsi `textbox()`.

Lebar kotak teks secara default adalah lebar maksimal dari baris teks. Namun, lebar ini juga dapat diatur oleh pengguna.

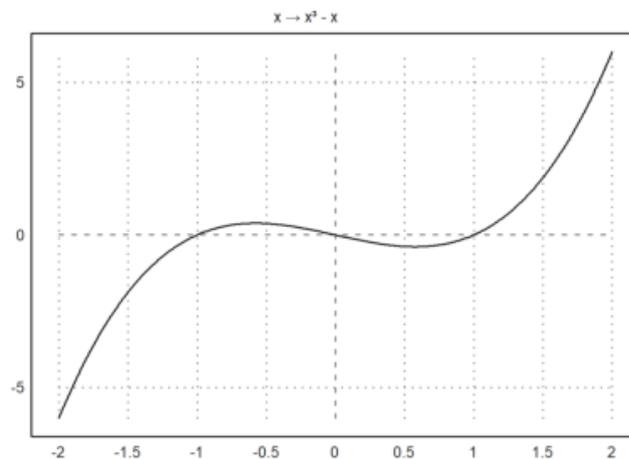
```
>function f(x) &= exp(-x)*sin(2*pi*x); ...
plot2d("f(x)",0,2pi); ...
textbox(latex("\text{Example of a damped oscillation}\ f(x)=e^{-x}\sin(2\pi x)"),w=0.85):
```



catatan: Grafik yang dihasilkan adalah plot osilasi teredam  $f(x)$ , yang mulai dengan amplitudo tinggi dan secara bertahap berkurang menuju nol. Teks yang ditambahkan akan berisi keterangan yang menjelaskan grafik tersebut sebagai "Example of a damped oscillation" dengan fungsi matematisnya dalam format LaTeX.

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

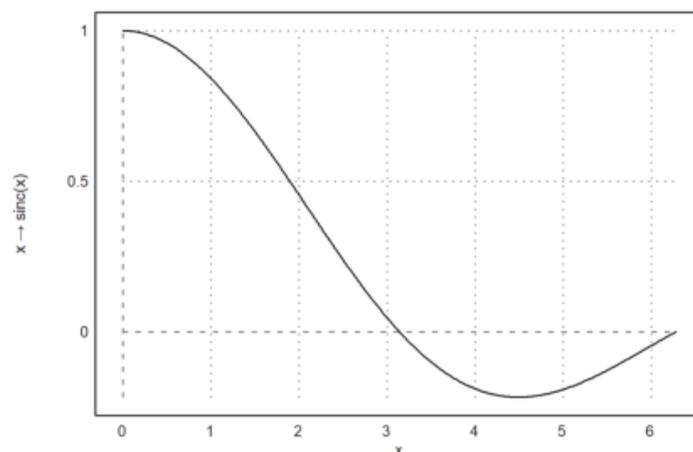
```
>plot2d("x^3-x",title="x \rightarrow x^3 - x"):
```



catatan: Grafik fungsi  $x^3 - x$  akan diplot, dan judul grafiknya adalah  $x \rightarrow x^3 - x$ , dengan panah kanan dan eksponen 3 ditampilkan menggunakan Unicode.

Label pada sumbu x dan y dapat ditampilkan secara vertikal, begitu juga dengan sumbu itu sendiri.

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



catatan: Plot grafik  $\text{sinc}(x)$  dari 0 hingga  $2\pi$  dengan sumbu-x berlabel "x" dan sumbu-y menampilkan "x ->  $\text{sinc}(x)$ " dalam format vertikal.

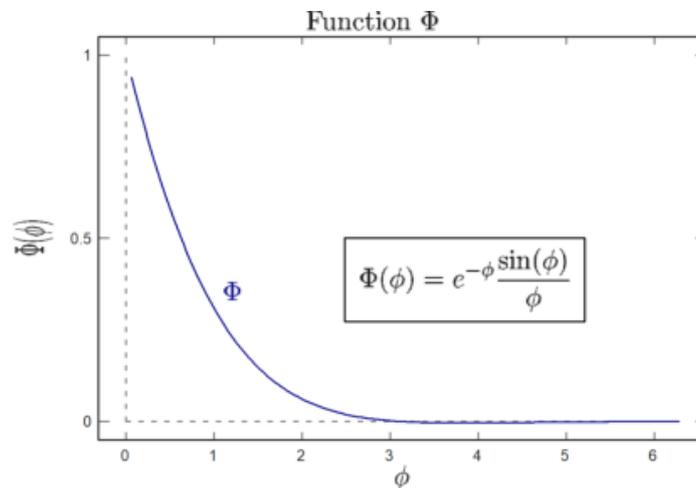
## LaTeX

Anda juga dapat memplot rumus LaTeX jika Anda telah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Jalur ke binari "latex" dan "dvi2png" harus ada dalam jalur sistem, atau Anda harus mengatur LaTeX di menu opsi.

Perhatikan bahwa parsing LaTeX cukup lambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, Anda harus memanggil ``latex()`` sekali sebelum loop dan menggunakan 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.

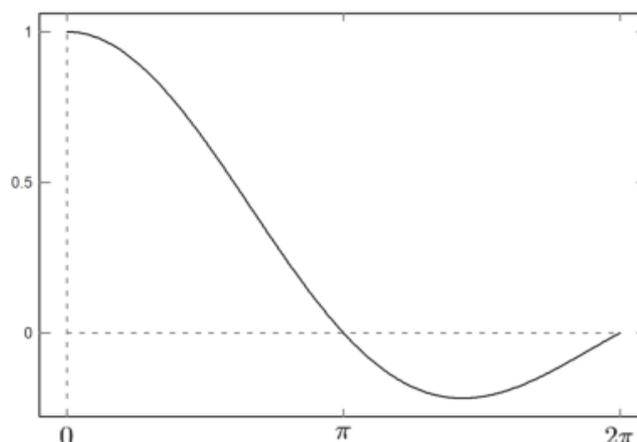
```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
  title=latex("\text{Function }\Phi$"), ...
  xl=latex("\phi"),yl=latex("\Phi(\phi)")); ...
textbox( ...
  latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}"),x=0.8,y=0.5); ...
label(latex("\Phi",color=blue),1,0.4):
```



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

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

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



catatan: Grafik  $\text{sinc}(x)$  ditampilkan dari 0 hingga  $2\pi$  dengan grid yang telah disesuaikan. Label sumbu-x diatur menggunakan simbol  $p$  dalam format LaTeX, dan grid pada sumbu-x serta sumbu-y diatur dengan interval khusus yang memberikan kontrol lebih detail atas tampilan grid dan ticks.

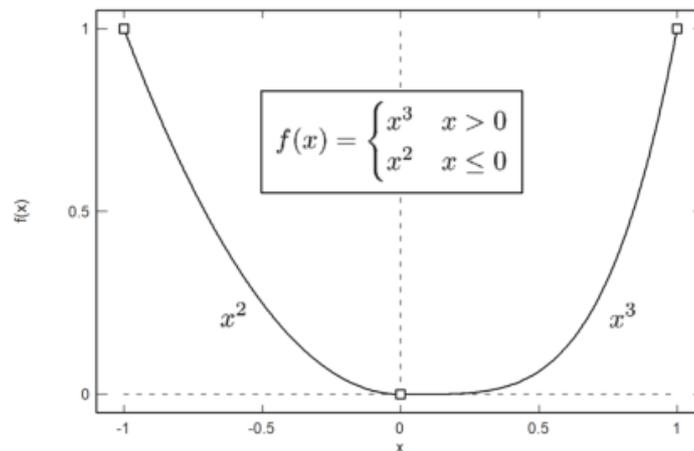
Tentu saja, fungsi juga dapat digunakan.

```
>function map f(x) ...
  if x>0 then return x^4
  else return x^2
endif
endfunction
```

Parameter "map" membantu menggunakan fungsi untuk vektor. Untuk plot, ini mungkin tidak diperlukan. Namun, untuk menunjukkan bahwa vektorisasi itu berguna, kita menambahkan beberapa titik kunci ke plot pada  $x=-1$ ,  $x=0$ , dan  $x=1$ .

Dalam plot berikut, kita juga memasukkan beberapa kode LaTeX. Kita menggunakannya untuk dua label dan sebuah kotak teks. Tentu saja, Anda hanya dapat menggunakan LaTeX jika Anda telah menginstal LaTeX dengan benar.

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



## Interaksi Pengguna

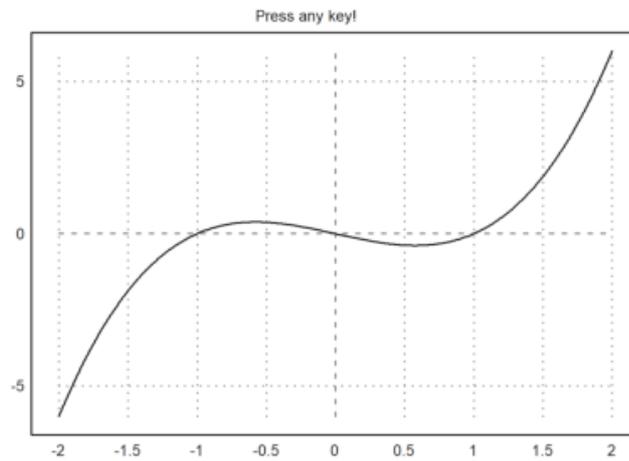
Saat memplot fungsi atau ekspresi, parameter `>user` memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kursor atau mouse. Pengguna dapat:

- Memperbesar dengan + atau -
- Menggeser plot dengan tombol kursor
- Memilih jendela plot dengan mouse
- Mengatur ulang tampilan dengan tombol spasi
- Keluar dengan tombol return

Tombol spasi akan mengatur ulang plot ke jendela plot yang asli.

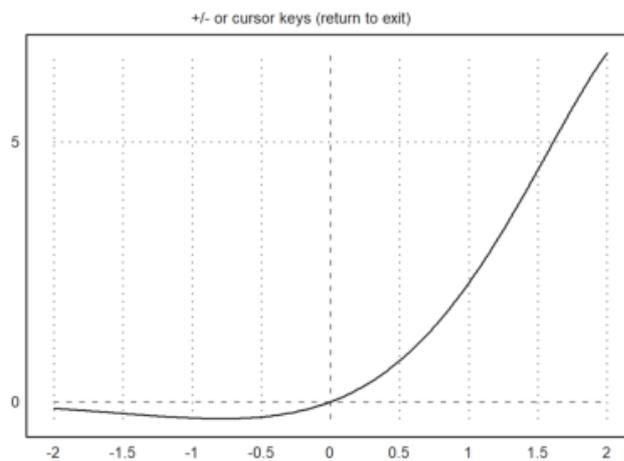
Saat memplot data, bendera `>user` akan menunggu tekanan tombol.

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



catatan: Plot dari fungsi  $x^3 - x$  ditampilkan dengan  $a = 1$ , dan pengguna diminta untuk menekan tombol atau melakukan interaksi lain sesuai instruksi di layar.

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



catatan: Grafik dari fungsi  $e^x \sin(x)$  ditampilkan dengan instruksi yang jelas untuk berinteraksi, memberikan pengguna fleksibilitas untuk menjelajahi grafik secara langsung.

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

Fungsi bawaan ``mousedrag()`` menunggu peristiwa mouse atau keyboard. Fungsi ini melaporkan mouse down, mouse moved, atau mouse up, serta tekanan tombol. Fungsi ``dragpoints()`` memanfaatkan ini, dan memungkinkan pengguna untuk menyeret titik mana pun dalam plot.

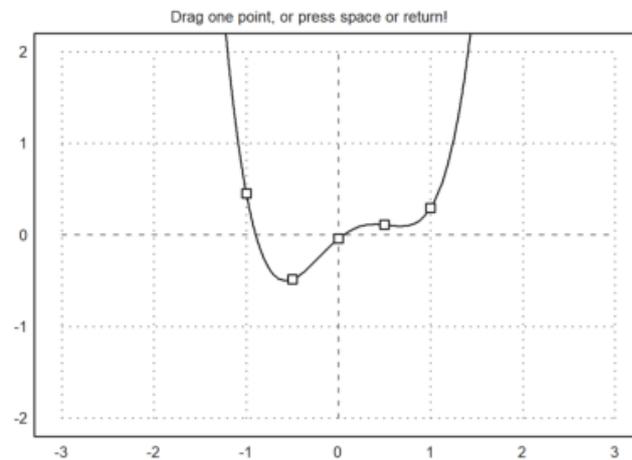
Kita perlu fungsi plot terlebih dahulu. Sebagai contoh, kita melakukan interpolasi pada 5 titik dengan polinomial. Fungsi ini harus memplot ke dalam area plot yang tetap.

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

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

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

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



catatan: Grafik akan menampilkan titik-titik yang dapat di-drag di sepanjang sumbu y, memungkinkan pengguna untuk berinteraksi secara langsung dengan grafik dan mengubah posisi titik-titik tersebut.

Ada juga fungsi yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna untuk menyesuaikan parameter tersebut.

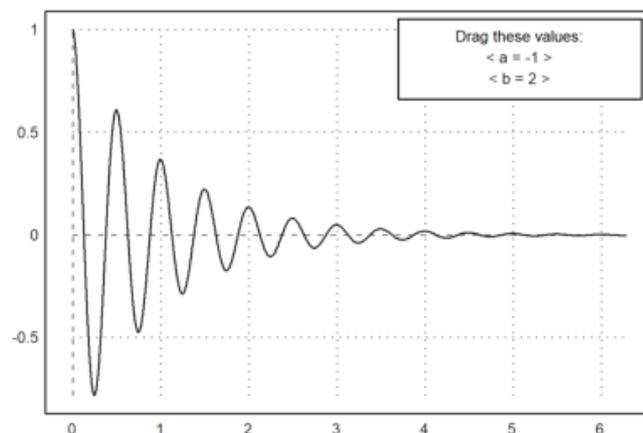
Pertama, kita memerlukan fungsi plot tersebut.

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

Kemudian kita memerlukan nama untuk parameter, nilai awal, dan matriks nx2 untuk rentang, serta baris judul secara opsional.

Terdapat slider interaktif yang dapat digunakan pengguna untuk mengatur nilai. Fungsi `dragvalues()` menyediakan fitur ini.

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



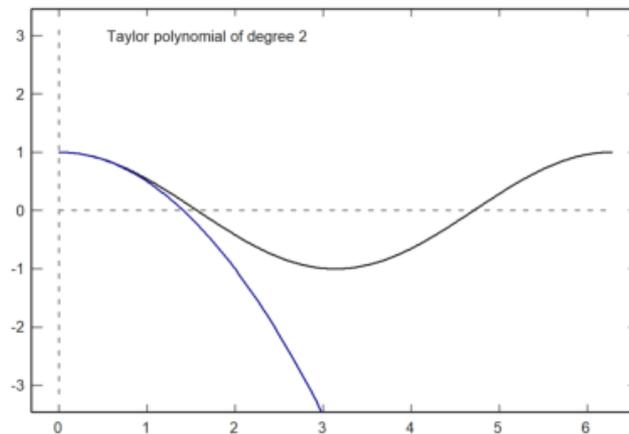
catatan: Pengguna akan melihat slider untuk kedua parameter, memungkinkan mereka untuk dengan mudah menyesuaikan nilai dan langsung melihat efeknya pada grafik yang ditampilkan.

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

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

Sekarang kita mengizinkan derajat  $n$  bervariasi dari 0 hingga 20 dalam 20 langkah. Hasil dari `dragvalues()` digunakan untuk memplot sketsa dengan  $n$  ini, dan untuk menyisipkan plot ke dalam notebook.

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



catatan: Pengguna akan melihat slider untuk `degree` dan dapat mengubah nilainya. Fungsi `plotf` kemudian akan diperbarui secara otomatis untuk mencerminkan nilai baru yang dipilih, memungkinkan eksplorasi yang interaktif terhadap perilaku fungsi.

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

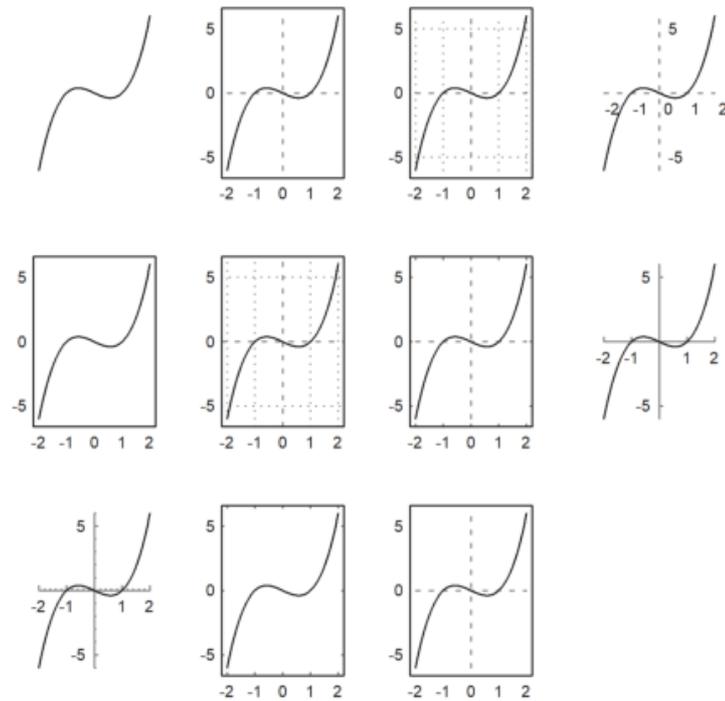
```
>function dragtest ...
  plot2d(none,r=1,title="Drag with the mouse, or press any key!");
  start=0;
  repeat
    {flag,m,time}=mousedrag();
    if flag==0 then return; endif;
    if flag==2 then
      hold on; mark(m[1],m[2]); hold off;
    endif;
  end
endfunction
```

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

## Gaya Plot 2D

Secara default, EMT menghitung tanda sumbu otomatis dan menambahkan label ke setiap tanda. Ini dapat diubah dengan parameter `grid`. Gaya default dari 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):
```

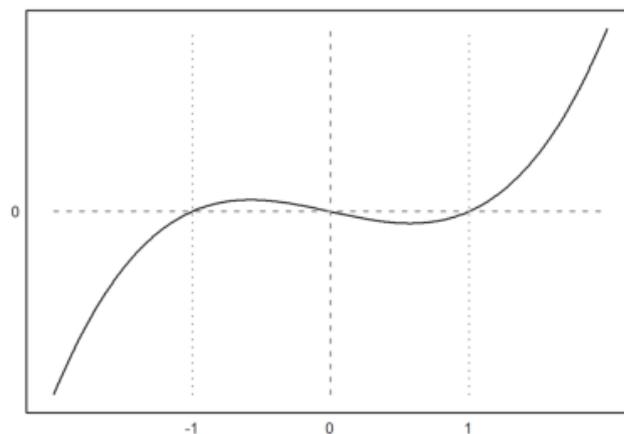


catatan: Dengan menggunakan variasi grid dan pengaturan, Anda dapat menciptakan grafik yang memiliki tingkat detail dan informasi yang berbeda. Ini sangat berguna untuk tujuan presentasi atau analisis, tergantung pada apa yang ingin Anda soroti dalam grafik.

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

Jika Anda ingin tanda sumbu Anda sendiri, Anda dapat menggunakan `style=0`, dan menambahkan semuanya kemudian.

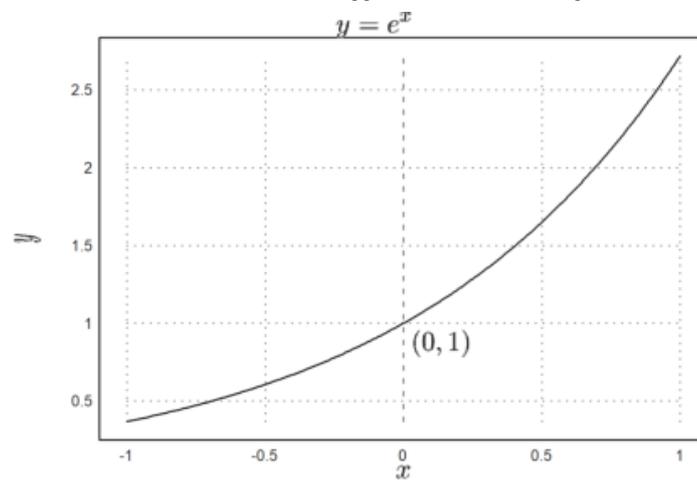
```
>aspect(1.5);
>plot2d("x^3-x",grid=0); // plot
>frame; xgrid([-1,0,1]); ygrid(0): // add frame and grid
```



catatan: dengan grafik tersebut kita mendapatkan grafik yang jelas dari fungsi  $x^3 - x$  dengan bingkai dan grid yang membantu menandai posisi penting pada sumbu. Ini memungkinkan analisis yang lebih mudah terhadap grafik, terutama di sekitar titik potong sumbu atau titik ekstrem.

Untuk judul plot dan label sumbu, lihat contoh berikut.

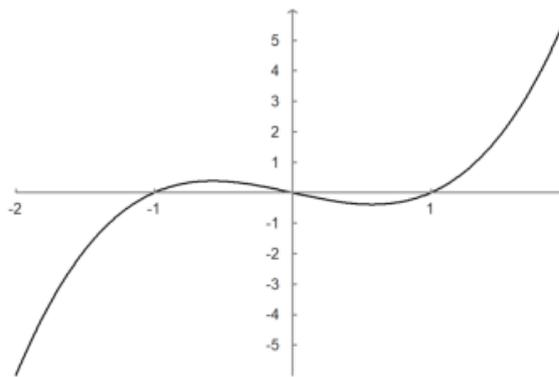
```
>plot2d("exp(x)",-1,1);
>textcolor(black); // set the text color to black
>title(latex("y=e^x")); // title above the plot
>xlabel(latex("x")); // "x" for x-axis
>ylabel(latex("y"),>vertical); // vertical "y" for y-axis
>label(latex("(0,1)"),0,1,color=blue): // label a point
```



catatan: grafik tersebut merupakan representasi visual yang jelas dari fungsi eksponen, dengan adanya elemen tambahan yang membantu menginterpretasikan data dengan baik.

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

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



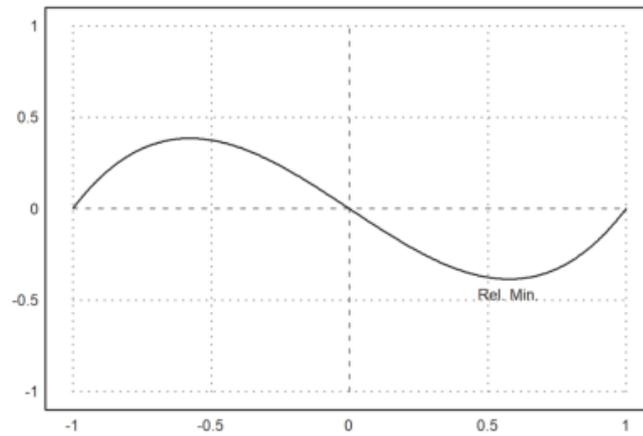
catatan: dari grafik tersebut tanpa adanya grid(bersih) dapat menjadikannya lebih fokus terhadap fungsi  $x^3-x$  dengan sumbu yang jelas.

Teks pada plot dapat diatur dengan menggunakan ``label()``. Dalam contoh berikut, "lc" berarti lower center, yang mengatur posisi label relatif terhadap koordinat plot.

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

$$x^3 - x$$

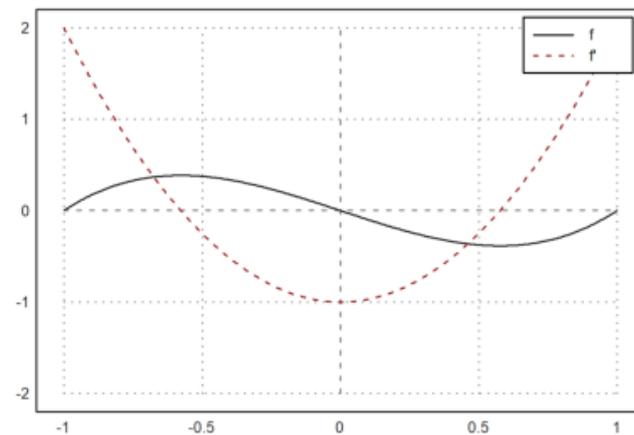
```
>plot2d(f, -1, 1, >square);
>x0=fmin(f, 0, 1); // compute point of minimum
>label("Rel. Min.", x0, f(x0), pos="lc"); // add a label there
```



catatan: grafik menunjukkan fungsi dengan titik minimum relatif yang jelas ditandai, memudahkan dalam menganalisis dan mengidentifikasi fungsi minimum tersebut.

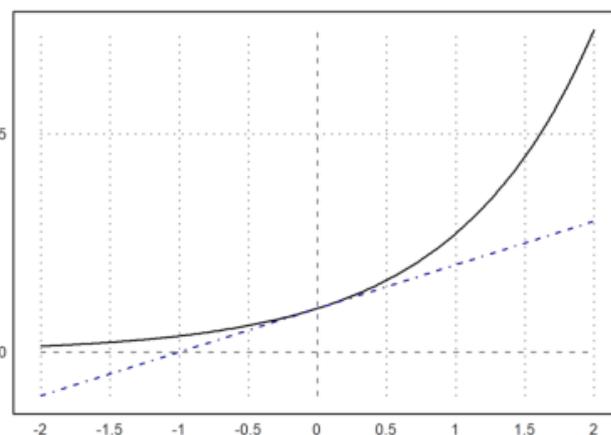
Ada juga kotak teks.

```
>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
```



catatan: hasil dari grafik menunjukkan fungsi dan turunannya dimana turunannya digambarkan dalam bentuk garis putus-putus dengan warna merah.

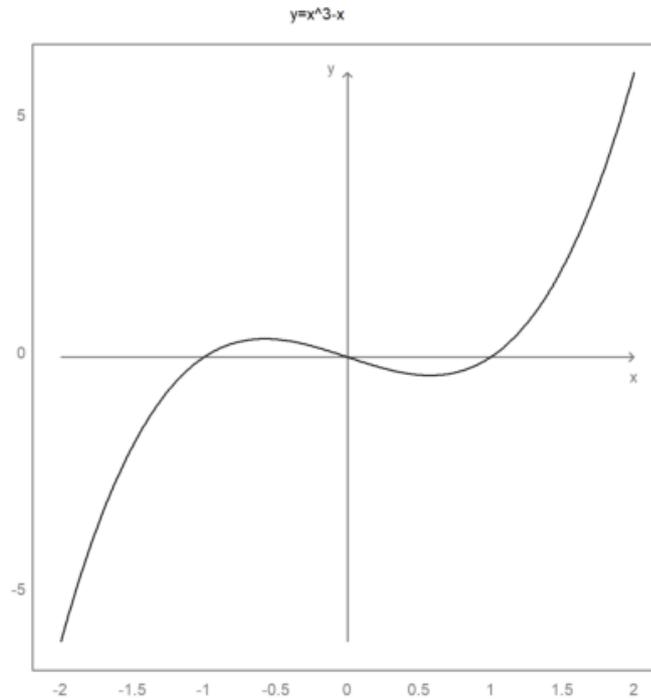
```
>plot2d(["exp(x)", "1+x"],color=[black,blue],style=["-", "-.-"]):
```



catatan: grafik menunjukkan kedua fungsi dalam satu plot, dimana fungsi  $e^x$  digambarkan dengan garis berwarna hitam, sedangkan fungsi  $1+x$  digambarkan dengan garis lurus putus-putus berwarna

biru. hal ini menunjukkan bahwa eksponensial mengalami pertumbuhan yang lebih cepat dibandingkan fungsi linear

```
>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():
```

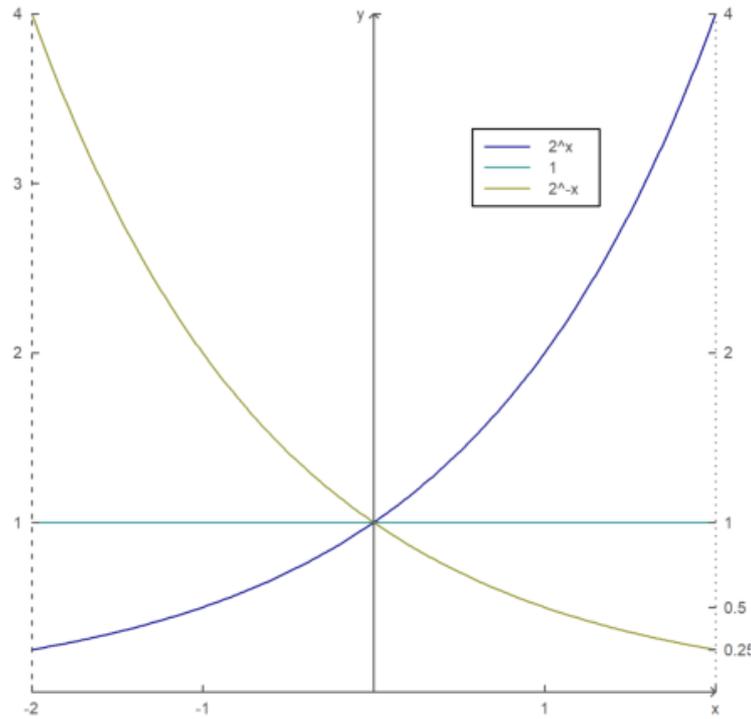


catatan: Grafik ini menampilkan fungsi  $y=x^3-x$  dengan grid dan label yang disesuaikan untuk memberikan tampilan yang bersih. Penggunaan warna abu-abu untuk elemen grid dan label memberikan kontras yang halus terhadap grafik utama, yang membuatnya lebih mudah dibaca tanpa terlalu mencolok.

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

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

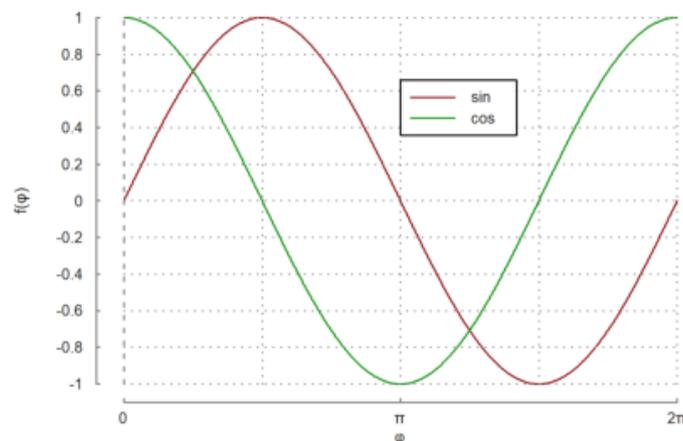
```
>fullwindow; ...
gridstyle(color=darkgray,textcolor=darkgray); ...
plot2d(["2^x","1","2^(-x)"],a=-2,b=2,c=0,d=4,<grid,color=4:6,<frame); ...
xaxis(0,-2:1,style="->"); xaxis(0,2,"x",<axis); ...
yaxis(0,4,"y",style="->"); ...
yaxis(-2,1:4,>left); ...
yaxis(2,2^(-2:2),style=".",<left); ...
labelbox(["2^x","1","2^(-x)"],colors=4:6,x=0.8,y=0.2);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



catatan: Grafik ini menunjukkan perilaku dari fungsi eksponensial positif dan negatif di rentang yang ditentukan. Dengan pengaturan sumbu yang jelas dan penggunaan warna yang berbeda, grafik ini memudahkan pemahaman bagaimana fungsi-fungsi tersebut berinteraksi. Penggunaan grid dan label yang terorganisir juga meningkatkan keterbacaan grafik, membuatnya efektif untuk analisis atau presentasi.

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

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



catatan: Grafik tersebut menampilkan dua fungsi trigonometri: sinus ( $\sin(x)$ ) dan kosinus ( $\cos(x)$ ) dalam interval dari 0 hingga  $2\pi$ .

- Sinus (merah): Memiliki puncak pada  $\pi/2$  dan nilai minimum pada  $3\pi/2$ , berosilasi antara -1 dan 1.
- Kosinus (hijau): Memulai dari 1 saat  $x = 0$ , dengan puncak pada 0 dan  $2\pi$ , serta minimum pada  $\pi$ .

Grafik dilengkapi dengan grid dan label yang jelas, membantu dalam memahami hubungan antara kedua fungsi. Sumbu x diberi penanda untuk 0,  $\pi$ , dan  $2\pi$ , sedangkan sumbu y mencakup rentang -0.1

hingga 1, sesuai dengan nilai fungsi. Ini memberikan gambaran yang jelas tentang perilaku kedua fungsi dalam satu periode.

## Mempplot Data 2D

Jika  $x$  dan  $y$  adalah vektor data, data ini akan digunakan sebagai koordinat  $x$  dan  $y$  dari sebuah kurva. Dalam hal ini,  $a$ ,  $b$ ,  $c$ , dan  $d$ , atau radius  $r$  dapat ditentukan, atau jendela plot akan menyesuaikan secara otomatis dengan data. Sebagai alternatif, `>square` dapat diatur untuk menjaga rasio aspek kuadrat.

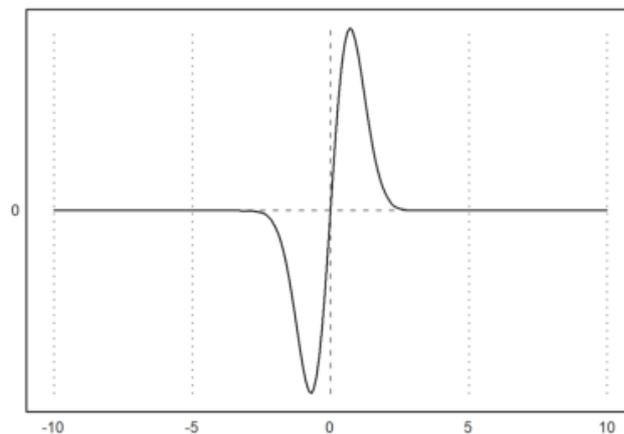
Mempplot sebuah ekspresi hanyalah singkatan untuk plot data. Untuk plot data, Anda memerlukan satu atau lebih baris nilai  $x$ , dan satu atau lebih baris nilai  $y$ . Dari rentang dan nilai  $x$ , fungsi `plot2d` akan menghitung data untuk dipplot, secara default dengan evaluasi adaptif fungsi. Untuk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

Namun, Anda juga dapat memasukkan data secara langsung.

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

Berikut adalah contoh dengan satu baris untuk  $x$  dan  $y$ .

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



catatan: Grafik dari fungsi  $y = e^{-x^2}x$  menunjukkan perilaku sebagai berikut:

- Bentuk: Fungsi ini memiliki bentuk simetris terhadap sumbu  $y$ , dengan nilai positif untuk  $x > 0$  dan negatif untuk  $x < 0$ .
- Nilai Maksimum: Mencapai nilai maksimum di sekitar  $x = 1$  sebelum menurun kembali.
- Nilai Dekat Nol: Di dekat  $x = 0$ , fungsi mendekati nol, dan saat  $|x|$  semakin besar, fungsi juga mendekati nol dengan cepat karena faktor  $e^{-x^2}$  yang mendominasi.
- Perilaku Asimptotik: Pada kedua ujung grafik (untuk  $x$  sangat positif atau negatif), nilai  $y$  mendekati nol.

Secara keseluruhan, grafik ini menunjukkan interaksi antara fungsi eksponensial yang menurun dan fungsi linear yang menghasilkan karakteristik unik dari grafik.

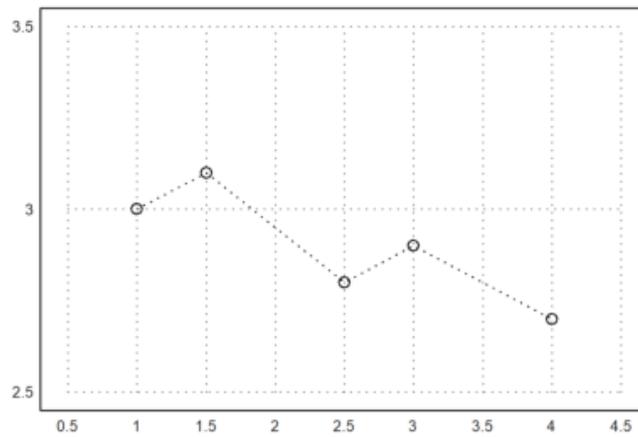
Data juga dapat dipplot sebagai titik. Gunakan `points=true` untuk ini. Plot bekerja seperti poligon, tetapi hanya menggambar sudut-sudutnya.

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

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



catatan: Grafik menunjukkan data titik dengan koordinat (1, 3), (1.5, 3.1), (2.5, 2.8), (3, 2.9), (4, 2.7).

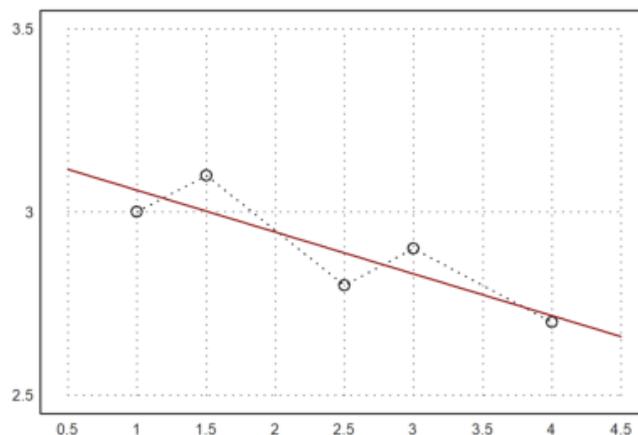
1.Pola Umum: Nilai y sedikit meningkat dari 3 pada  $x = 1$  ke 3.1 pada  $x = 1.5$ , kemudian menurun perlahan hingga 2.7 pada  $x = 4$ .

2.Visualisasi: Titik data ditandai dengan simbol "o", memberikan fokus pada posisi setiap titik dalam grafik.

3.Rentang Sumbu: Sumbu x dari 0.5 hingga 4.5 dan sumbu y dari 2.5 hingga 3.5 membantu memperjelas tren data.

Grafik ini efektif untuk menunjukkan pola perubahan dalam data yang diberikan.

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



catatan: Grafik menunjukkan data titik  $\{(1, 3), (1.5, 3.1), (2.5, 2.8), (3, 2.9), (4, 2.7)\}$  serta garis regresi linier yang ditambahkan dalam warna merah.

1.Pola Data: Titik-titik menunjukkan bahwa nilai y meningkat sedikit pada awalnya, lalu menurun.

2.Garis Regresi: Garis merah mewakili tren umum dari data, menunjukkan hubungan linier antara x dan y.

3.Visualisasi: Kombinasi titik dan garis regresi memudahkan pemahaman tentang kecenderungan data.

Garis regresi membantu mengidentifikasi pola linier yang mungkin ada dalam data.

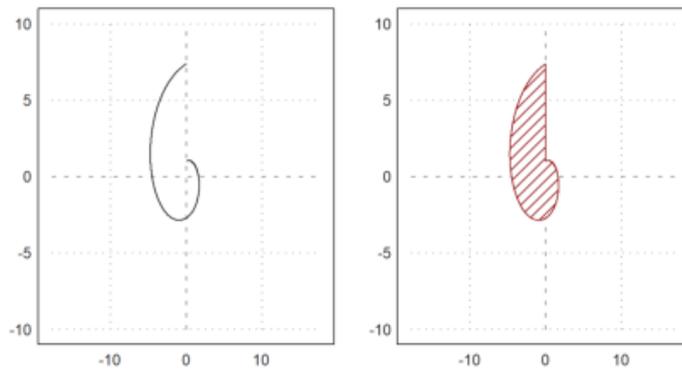
# Menggambar Daerah yang Dibatasi Kurva

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

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

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

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



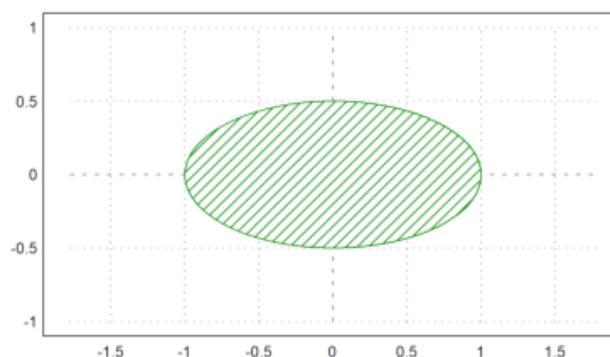
catatan: Grafik menunjukkan kurva parametrik yang didefinisikan oleh  $x(t) = \sin(t) \cdot e^{t/\pi}$  dan  $y(t) = \cos(t) \cdot e^{t/\pi}$  untuk  $t$  dari 0 hingga  $2\pi$ .

1. Bentuk Kurva: Kurva berputar dan mengembang seiring bertambahnya  $t$  karena faktor eksponensial.
2. Plot Kurva: Grafik pertama menampilkan kurva dengan radius 10, memberikan gambaran jelas tentang bentuknya.
3. Pengisian: Grafik kedua mengisi area di bawah kurva dengan warna merah, menyoroti bentuk yang dihasilkan.
4. Rasio Aspek: Aspek 16:9 membuat tampilan grafik lebih lebar dan estetik.

Grafik ini secara efektif menunjukkan sifat dinamis dan area yang dibentuk oleh kurva parametrik.

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

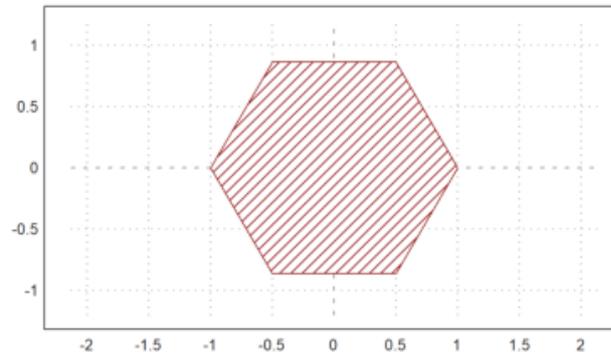
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```



cacatan: Grafik menunjukkan hubungan antara fungsi  $y = \sin(x)$  dan  $y = 0.5 \cdot \cos(x)$  untuk  $x$  dari 0 hingga  $2\pi$ .

1. Bentuk Kurva: Garis sinus bergerak antara -1 dan 1, sedangkan garis kosinus dikalikan 0.5, sehingga bergerak antara -0.5 dan 0.5.
  2. Pengisian Area: Area di bawah kurva  $\sin(x)$  diisi, menonjolkan bentuk yang dihasilkan.
  3. Rentang: Grafik mencakup satu periode penuh dari fungsi sinus dan kosinus.
- Grafik ini efektif dalam menggambarkan pola interaksi antara kedua fungsi secara visual.

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

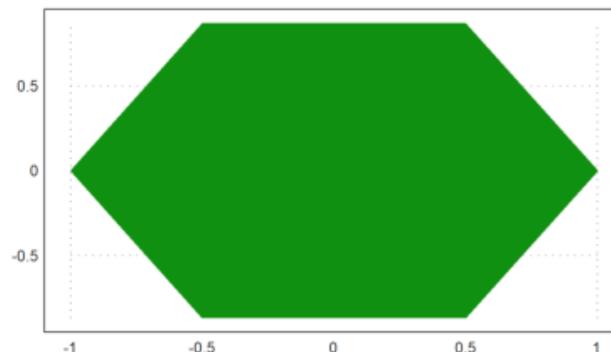


cacatan: Grafik menunjukkan lingkaran unit yang dibentuk oleh titik-titik pada fungsi  $x = \cos(t)$  dan  $y = \sin(t)$  untuk  $t$  dari 0 hingga  $2\pi$  dengan 6 titik.

1. Bentuk: Lingkaran unit dengan titik-titik pada posisi penting di sepanjang lingkaran.
2. Pengisian: Area di dalam lingkaran diisi dengan warna merah, menonjolkan bentuknya.
3. Radius: Radius ditetapkan pada 1.2, memberikan tampilan yang lebih tebal.

Grafik ini efektif dalam menggambarkan lingkaran unit dan area yang dibentuk oleh titik-titik tersebut.

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



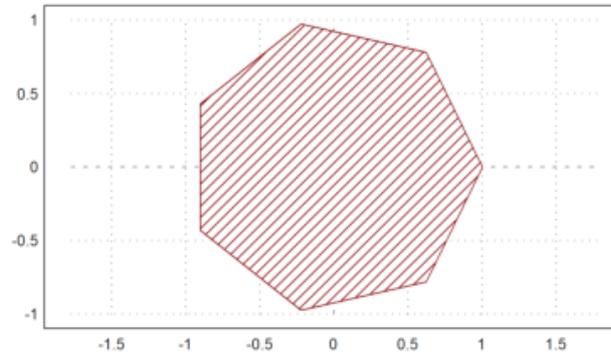
cacatan: Grafik menunjukkan lingkaran unit yang dibentuk oleh titik-titik pada fungsi  $x = \cos(t)$  dan  $y = \sin(t)$  untuk  $t$  dari 0 hingga  $2\pi$  dengan 6 titik

1. Bentuk: Titik-titik terdistribusi merata di sepanjang lingkaran unit.
2. Gaya: Garis ditampilkan dengan gaya `style="#"`, memberikan tampilan yang lebih tebal dan jelas tanpa mengisi area di dalam lingkaran.

Grafik ini secara efektif menggambarkan posisi titik pada lingkaran unit.

Contoh lainnya adalah septagon, yang kita buat dengan 7 titik di lingkaran satuan.

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



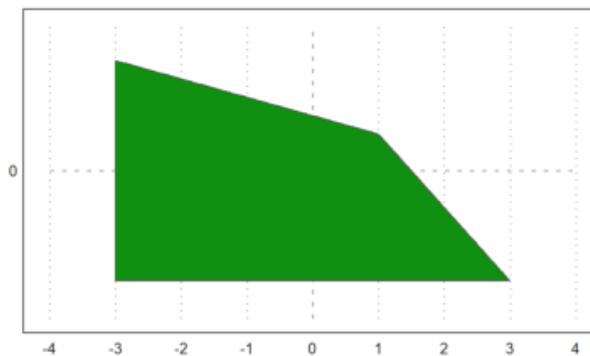
catatan: Grafik menunjukkan lingkaran unit yang dibentuk oleh titik-titik pada fungsi  $x = \cos(t)$  dan  $y = \sin(t)$  untuk  $t$  dari 0 hingga  $2\pi$  dengan 7 titik.

1. Bentuk: Titik-titik terdistribusi merata di sekitar lingkaran unit.
2. Pengisian: Area di dalam lingkaran diisi dengan warna merah, menonjolkan bentuknya.

Grafik ini secara efektif menggambarkan lingkaran unit dan area yang dibentuk oleh titik-titik tersebut.

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

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



catatan: Grafik menggambarkan fungsi  $f(x, y) = \max([x, y] \cdot A')$ , di mana  $A$  adalah matriks yang menentukan kombinasi  $x$  dan  $y$ .

1. Fungsi Maksimum: Grafik menunjukkan bagaimana nilai maksimum dari operasi matriks bervariasi di ruang  $x$  dan  $y$ .
2. Kontur: Level kontur ditampilkan antara 0 dan 3 dalam warna hijau, menunjukkan variasi nilai fungsi dalam wilayah tersebut.
3. Bentuk Permukaan: Garis-garis kontur membantu memahami perubahan nilai maksimum dari fungsi di seluruh bidang.

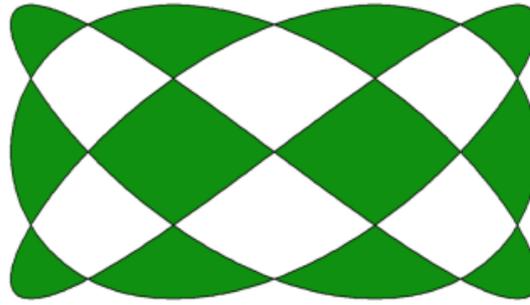
Grafik ini memberikan gambaran visual tentang perubahan nilai maksimum fungsi seiring dengan perubahan  $x$  dan  $y$ .

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

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

Sekarang kita memiliki vektor  $x$  dan  $y$  dari nilai-nilai. Fungsi `plot2d()` dapat memplot nilai-nilai ini sebagai kurva yang menghubungkan titik-titik. Plot ini juga dapat diisi. Dalam hal ini, ini menghasilkan hasil yang baik berkat aturan winding, yang digunakan untuk pengisian.

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



catatan: Grafik menunjukkan kurva Lissajous yang didefinisikan oleh  $x = \cos(3t)$  dan  $y = \sin(4t)$  untuk  $t$  dari 0 hingga  $2\pi$ .

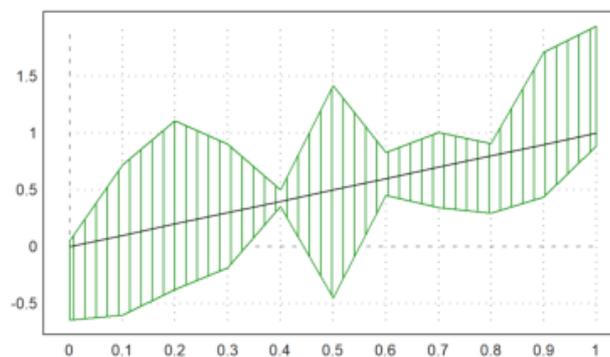
1. Bentuk: Kurva menampilkan pola simetris dan terjalin karena perbedaan frekuensi (3 dan 4) antara fungsi sinus dan kosinus.
2. Pengisian Area: Area di bawah kurva diisi, memperjelas bentuk kurva.
3. Grid dan Frame: Grid membantu dalam pelacakan titik, dan bingkai memberikan batas grafik.

Grafik ini menggambarkan pola Lissajous dengan tampilan yang jelas dan terstruktur.

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

Ini bisa berguna untuk memplot kesalahan dalam perhitungan. Namun, ini juga bisa 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):
```



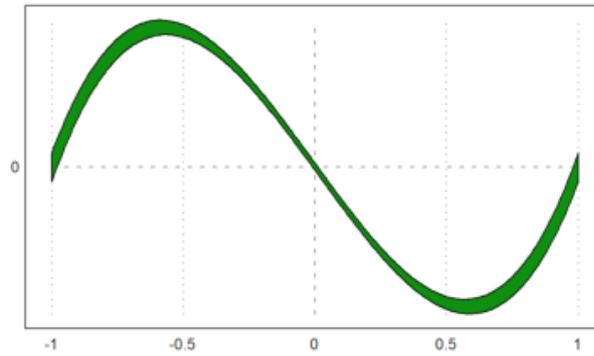
catatan: Grafik menunjukkan variasi acak di sekitar garis lurus  $y = t$  untuk  $t$  dalam rentang 0 hingga 1.

1. Error Bars: Interval acak vertikal (seperti error bars) digambarkan di sekitar setiap titik  $t$ , menunjukkan fluktuasi.
2. Garis Lurus: Garis lurus  $y = t$  ditambahkan untuk menunjukkan hubungan ideal.

Grafik ini memperlihatkan bagaimana nilai acak menyebar di sekitar garis lurus  $y = t$ .

Jika  $x$  adalah vektor yang terurut, dan  $y$  adalah vektor interval, maka `plot2d` akan memplot rentang terisi dari interval di bidang. Gaya pengisian yang digunakan sama dengan gaya poligon.

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



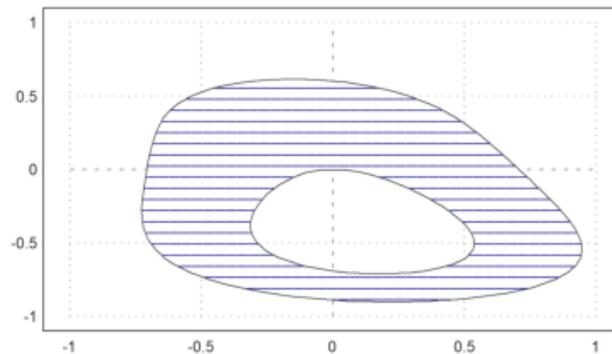
cacatan: Grafik menunjukkan fungsi kubik  $y = x^3 - x$ , di mana  $x$  bervariasi sedikit secara acak di sekitar  $t$ .

1. Fungsi Kubik: Grafik memiliki bentuk simetris seperti huruf "S", dengan titik-titik di sekitar  $t = -1$ ,  $t = 0$ , dan  $t = 1$ .
2. Variasi Acak: Nilai  $x$  dipilih dalam interval kecil di sekitar  $t$ , memberikan sedikit distorsi pada kurva.

Grafik ini menggambarkan bentuk fungsi kubik dengan sedikit variasi acak di sekitar nilai-nilai  $t$ .

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

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



cacatan: Grafik menunjukkan kontur dari fungsi  $f(x, y) = 2x^2 + xy + 3y^4 + y$  pada level antara 0 dan 1.

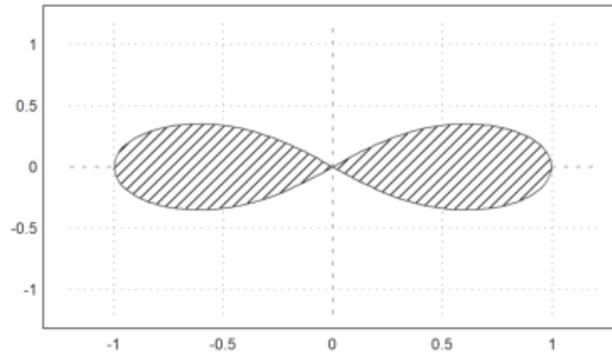
1. Fungsi: Fungsi terdiri dari komponen kuadrat, pangkat empat, dan linear dari  $x$  dan  $y$ .
2. Kontur: Garis kontur mewakili nilai  $f(x, y)$  yang berada di antara 0 dan 1.
3. Warna: Kontur digambar dengan garis biru.

Grafik ini memperlihatkan bagaimana fungsi  $f(x, y)$  berperilaku di area tertentu berdasarkan level kontur.

kita juga dapat mengisi rentang nilai seperti

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

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

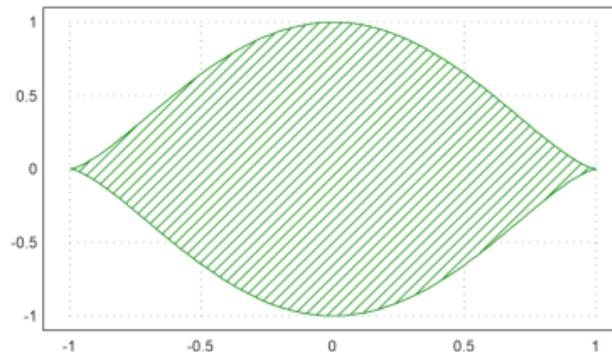


cacatan: Grafik menunjukkan kontur dari fungsi  $f(x, y) = (x^2 + y^2)^2 - x^2 + y^2$  pada level antara -1 dan 0.

- 1.Fungsi: Fungsi melibatkan komponen kuadrat yang menciptakan bentuk kompleks.
- 2.Kontur: Garis kontur menunjukkan area di mana nilai fungsi berada di antara -1 dan 0.
- 3.Gaya: Kontur digambar dengan gaya garis miring `style="/"`.

Grafik ini memberikan pandangan tentang perilaku fungsi dalam rentang nilai tertentu.

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



cacatan: Grafik menunjukkan area yang dibentuk oleh fungsi  $y = \cos(x)$  dan  $y = \sin^3(x)$  dalam rentang x dari 0 hingga  $2\pi$ .

- 1.Fungsi:  $y = \cos(x)$  berosilasi antara -1 dan 1, sementara  $y = \sin^3(x)$  tetap positif dan lebih kecil dari 1.
- 2.Pengisian Area: Area antara kedua kurva diisi, memberikan visualisasi jelas tentang perbedaan fungsi.

Grafik ini menggambarkan interaksi antara  $\cos(x)$  dan  $\sin^3(x)$  serta area yang dibentuk di antara keduanya.

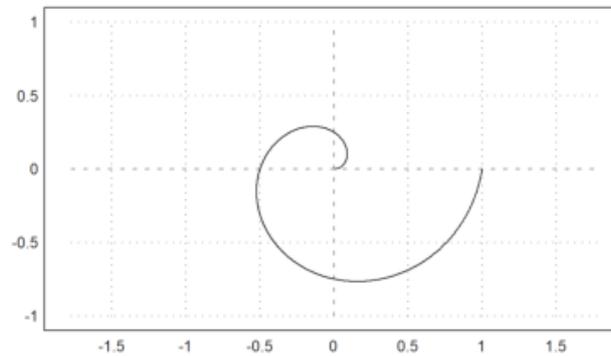
## Grafik Fungsi Parametrik

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

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

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

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



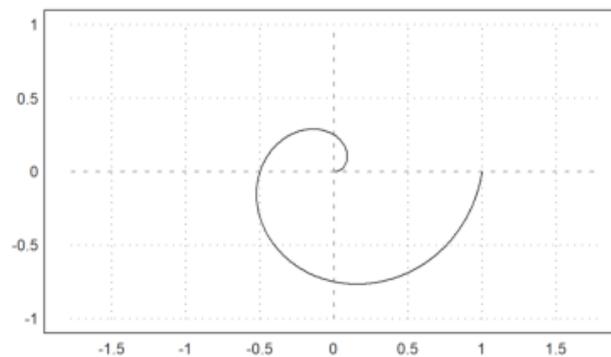
catatan: Grafik menunjukkan kurva parametrik yang didefinisikan oleh  $x = t \cos(2\pi t)$  dan  $y = t \sin(2\pi t)$  untuk  $t$  dari 0 hingga 1.

1. Bentuk: Kurva membentuk spiral yang berkembang, dengan osilasi yang dihasilkan dari frekuensi  $2\pi t$ .
2. Rentang: Menggambarkan satu putaran penuh dengan jarak dari pusat yang meningkat seiring bertambahnya  $t$ .

Grafik ini menampilkan visualisasi spiral yang menarik berdasarkan fungsi parametrik.

Sebagai alternatif, kita juga dapat menggunakan dua ekspresi untuk kurva. Berikut ini memplot kurva yang sama seperti di atas.

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

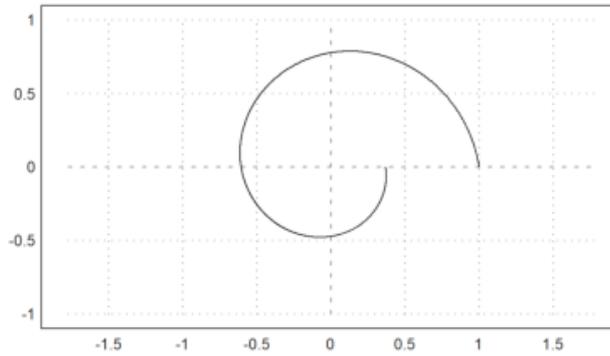


catatan: Grafik menunjukkan kurva parametrik yang didefinisikan oleh  $x = x \cos(2\pi x)$  dan  $y = x \sin(2\pi x)$  untuk  $x$  dari 0 hingga 1.

1. Bentuk: Kurva membentuk pola spiral yang berkembang dengan osilasi akibat fungsi sinus dan kosinus.
2. Rentang: Mencakup nilai  $x$  dari 0 hingga 1, menghasilkan satu putaran penuh.

Grafik ini menampilkan visualisasi spiral yang menarik berdasarkan fungsi parametrik.

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



catatan: Grafik menunjukkan kurva spiral yang mengecil, didefinisikan oleh  $x = e^{-t} \cos(2\pi t)$  dan  $y = e^{-t} \sin(2\pi t)$  untuk  $t$  dari 0 hingga 1.

1. Bentuk: Spiral mengecil karena radius  $r = e^{-t}$  berkurang secara eksponensial saat  $t$  bertambah.
2. Frekuensi: Menghasilkan beberapa putaran saat  $t$  bergerak dari 0 ke 1.

Grafik ini menampilkan spiral eksponensial yang menarik, dengan interaksi antara fungsi sinus dan kosinus.

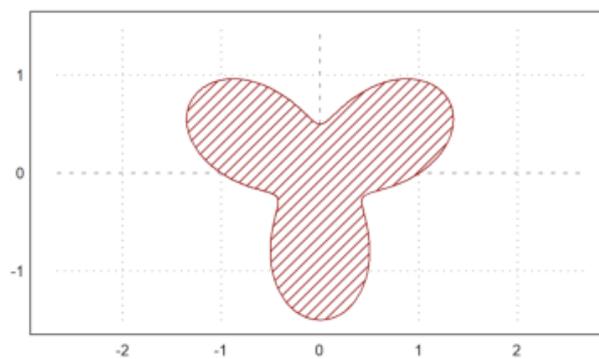
Dalam contoh berikut, kita memplot kurva

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

dengan

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

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



catatan: Grafik menunjukkan pola bunga yang dihasilkan dari fungsi polar  $r = 1 + 1/2 \sin(3t)$  untuk  $t$  dari 0 hingga  $2\pi$ .

1. Bentuk: Fungsi ini menciptakan tiga "kelopak" karena osilasi  $\sin(3t)$ .
2. Pengisian: Area di dalam kurva diisi dengan warna merah, menonjolkan bentuknya.

Grafik ini memberikan visualisasi yang menarik dari pola bunga berdasarkan fungsi polar.

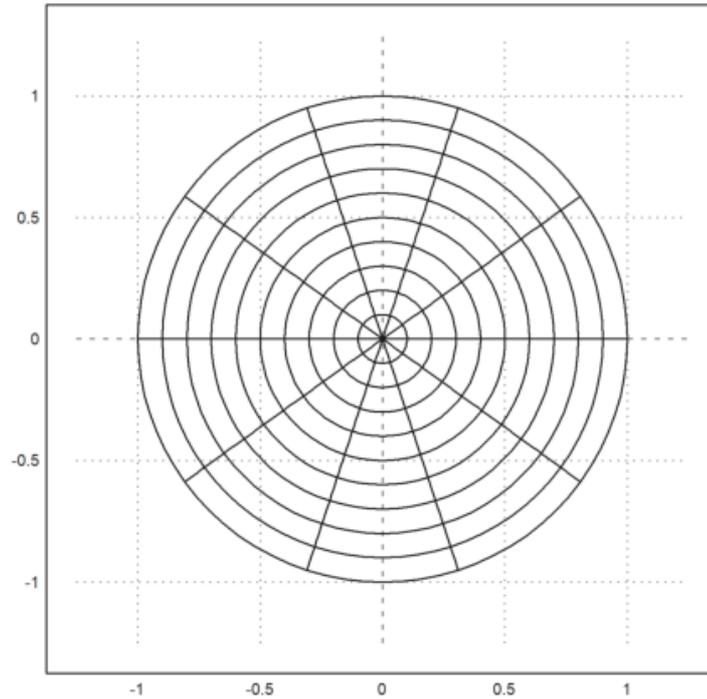
## Menggambar Grafik Bilangan Kompleks

Sebuah array bilangan kompleks juga dapat dipetakan. Titik-titik grid akan dihubungkan. Jika sejumlah garis grid ditentukan (atau vektor 1x2 dari garis grid) dalam argumen `cgrid`, hanya garis grid tersebut yang akan terlihat.

Sebuah matriks bilangan kompleks secara otomatis akan dipetakan sebagai grid di bidang kompleks.

Dalam contoh berikut, kita memplot citra dari 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):
```

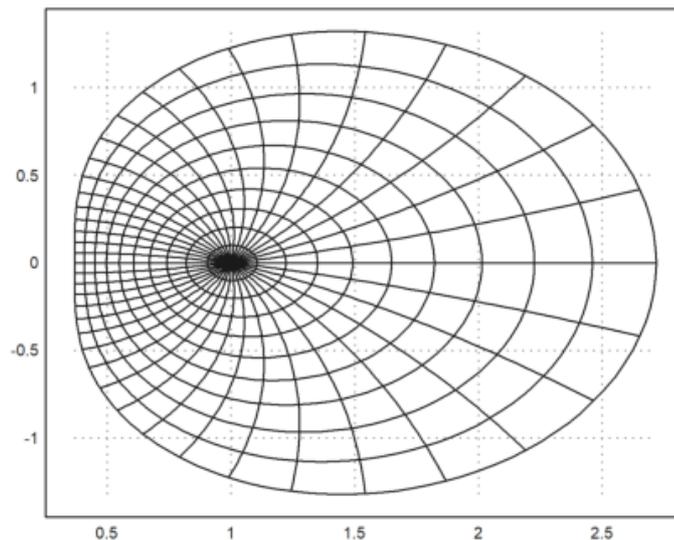


catatan Grafik menunjukkan lingkaran dalam bidang kompleks, didefinisikan oleh  $z = r e^{i a}$  dengan  $r$  dari 0 hingga 1 dan  $a$  dari 0 hingga  $2\pi$ .

1. Bentuk: Menghasilkan lingkaran dengan jari-jari bervariasi dari 0 hingga 1.
2. Koordinat Polar: Menunjukkan hubungan antara radius dan sudut.

Grafik ini memberikan visualisasi yang jelas dari lingkaran dalam bidang kompleks.

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

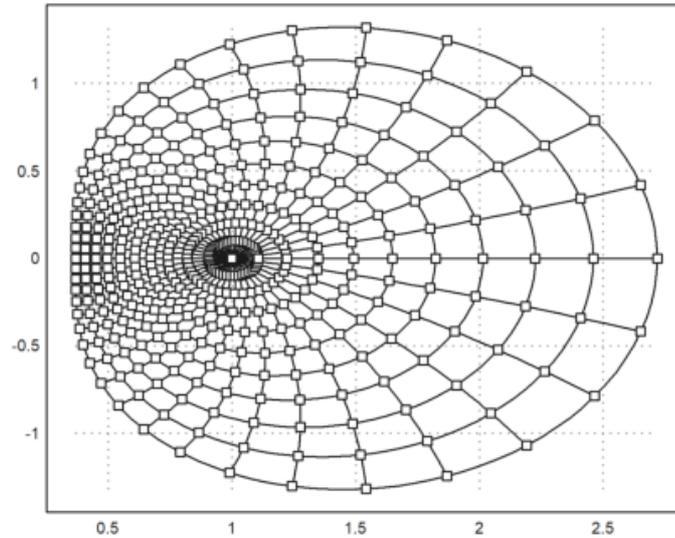


catatan: Grafik menunjukkan representasi fungsi eksponensial kompleks  $e^z$  dengan  $z = r e^{i a}$  untuk  $r$  dari 0 hingga 1 dan  $a$  dari 0 hingga  $2\pi$ .

1. Bentuk: Menciptakan pola spiral atau gelombang yang kompleks.
2. Koordinat Polar: Titik dihasilkan dari kombinasi radius dan sudut.

Grafik ini memberikan visualisasi menarik tentang perilaku fungsi eksponensial dalam bidang kompleks.

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

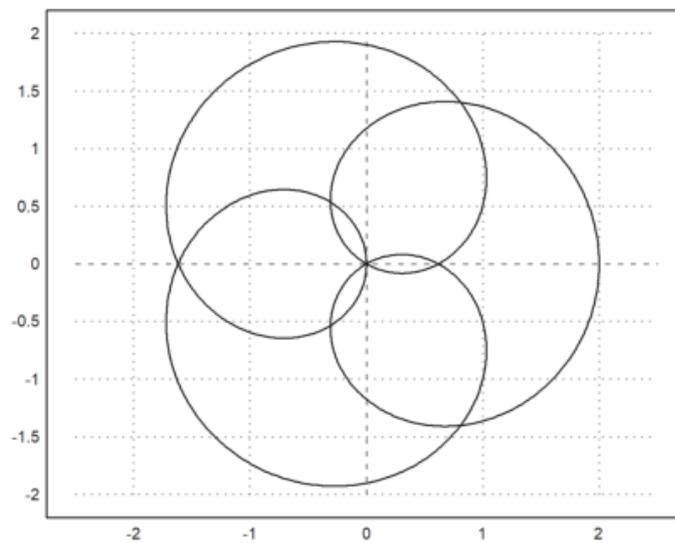


ebuah vektor bilangan kompleks secara otomatis digambarkan sebagai kurva di bidang kompleks dengan bagian real dan bagian imajiner.

Dalam contoh ini, kita menggambarkan lingkaran satuan dengan

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

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



## Plot Statistik

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

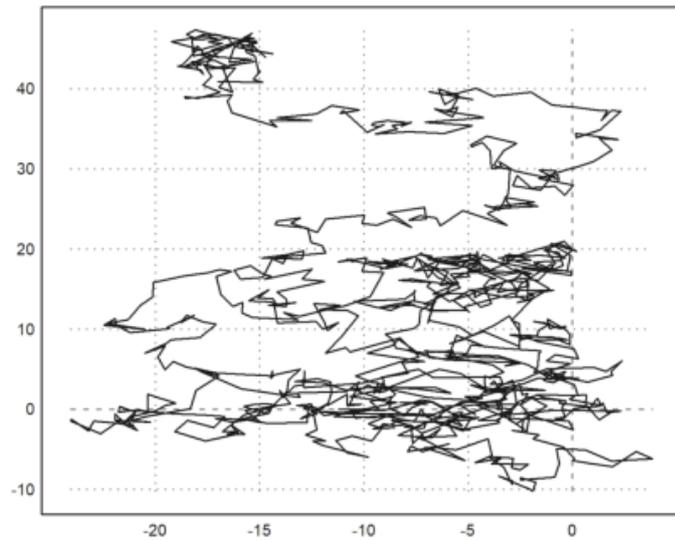
Jumlah kumulatif dari nilai yang terdistribusi normal 0-1 menghasilkan sebuah langkah acak.

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

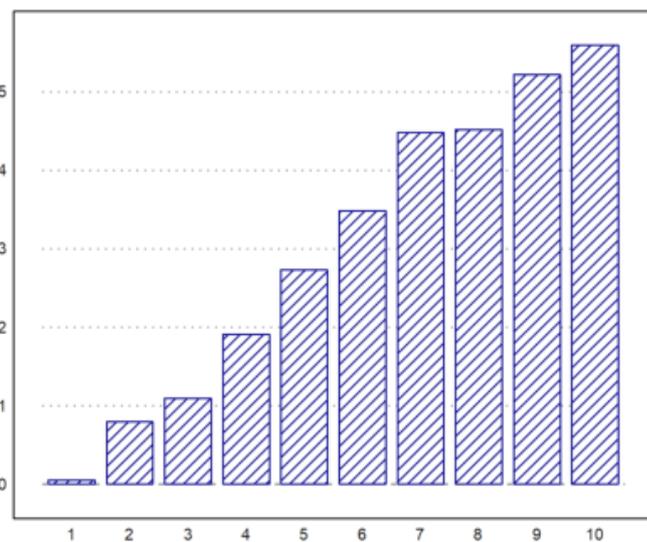


Menggunakan dua baris menunjukkan langkah dalam dua dimensi.

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

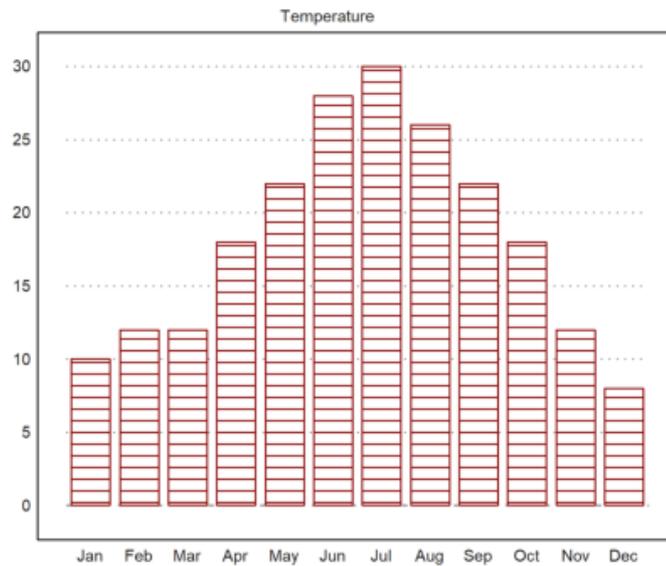


```
>columnplot(cumsum(random(10)),style="/",color=blue):
```

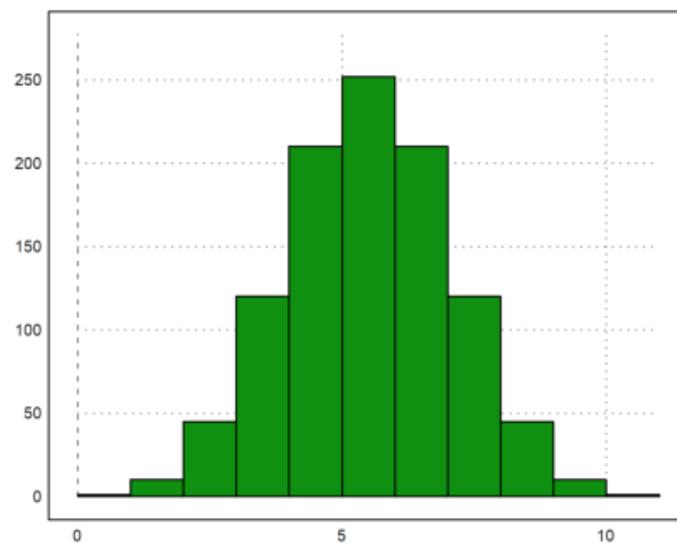


Ini juga dapat menampilkan string sebagai label.

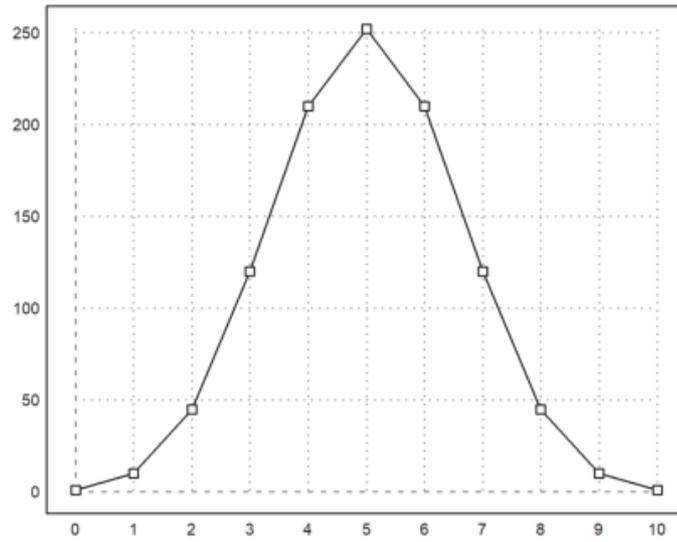
```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
"Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnplot(values,lab=months,color=red,style="-");
>title("Temperature"):
```



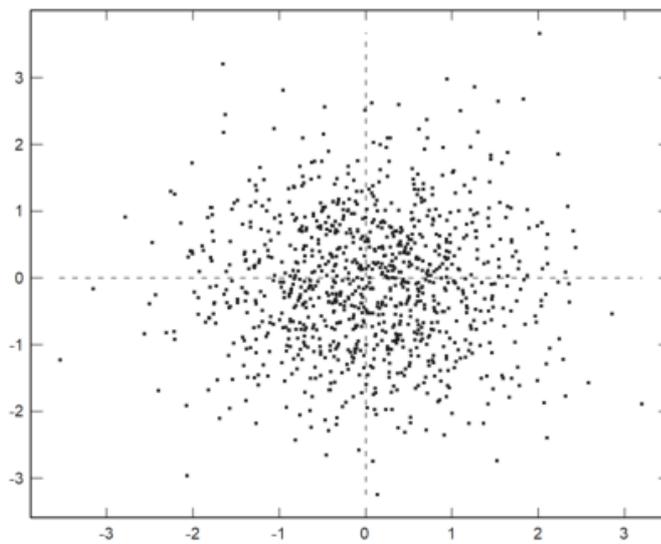
```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```



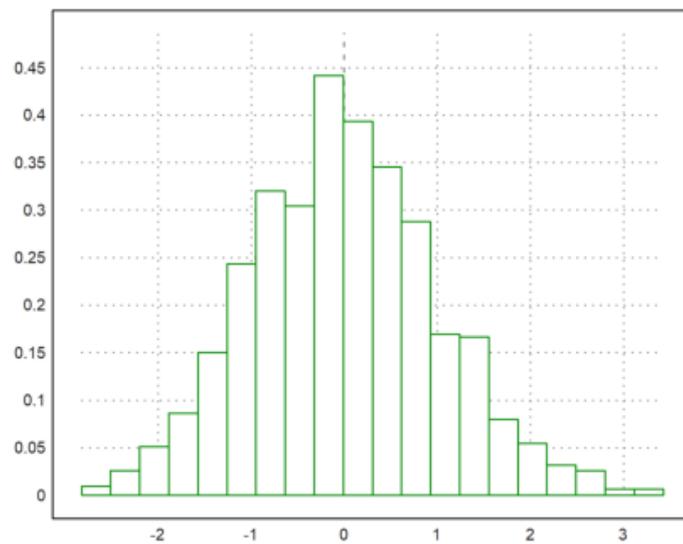
```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



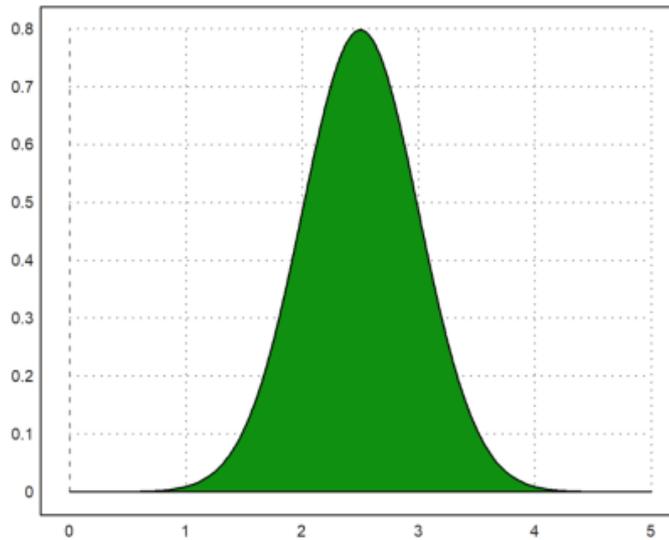
```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```



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

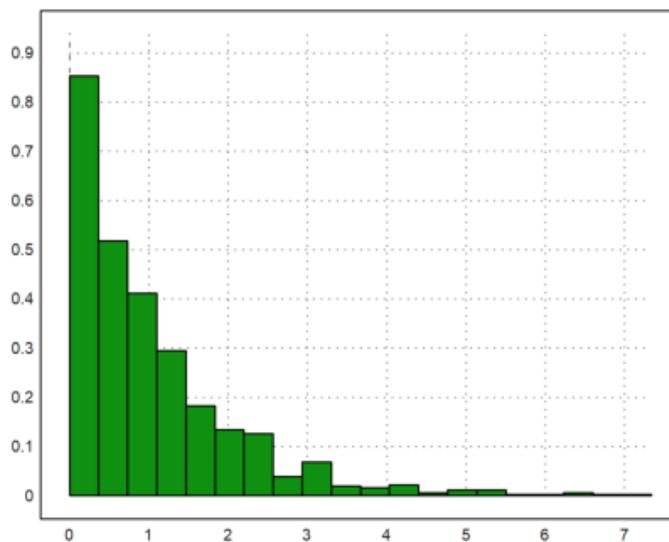


```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



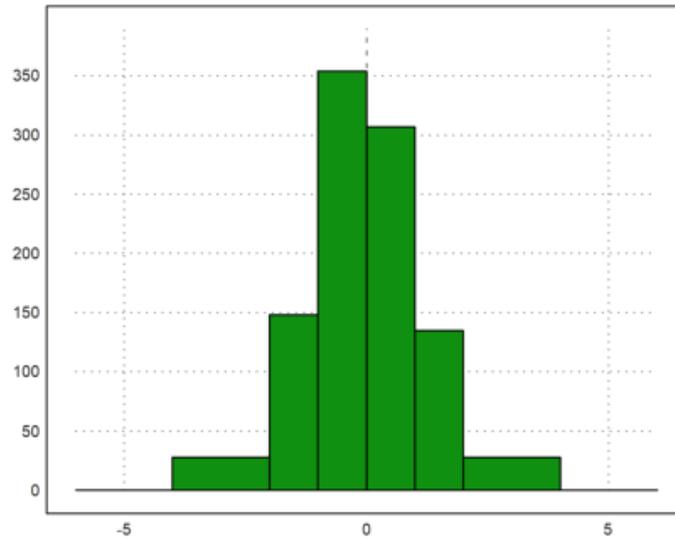
Untuk memplot distribusi statistik eksperimen, Anda dapat menggunakan `distribution=n` dengan `plot2d`.

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



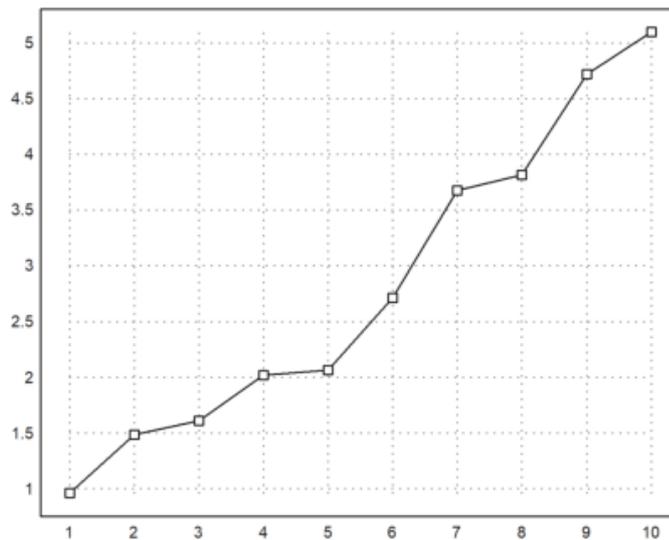
Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan `>bar` di `plot3d`, atau dengan plot kolom.

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

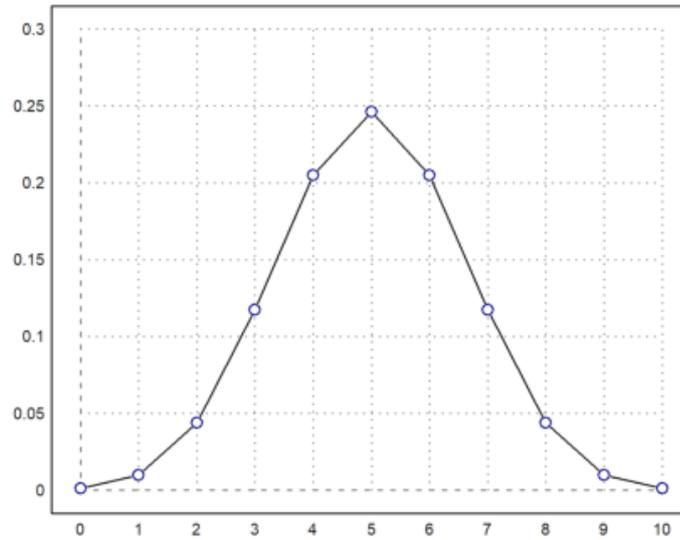


Fungsi `statplot()` mengatur gaya dengan menggunakan sebuah string sederhana.

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



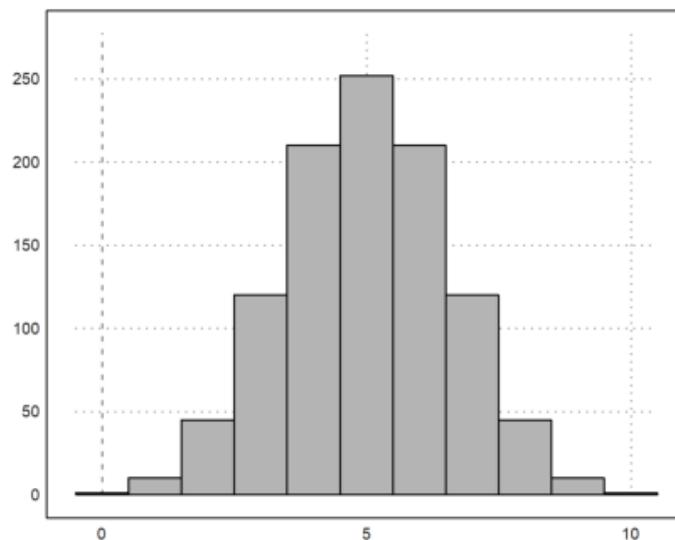
```
>n=10; i=0:n; ...
plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



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

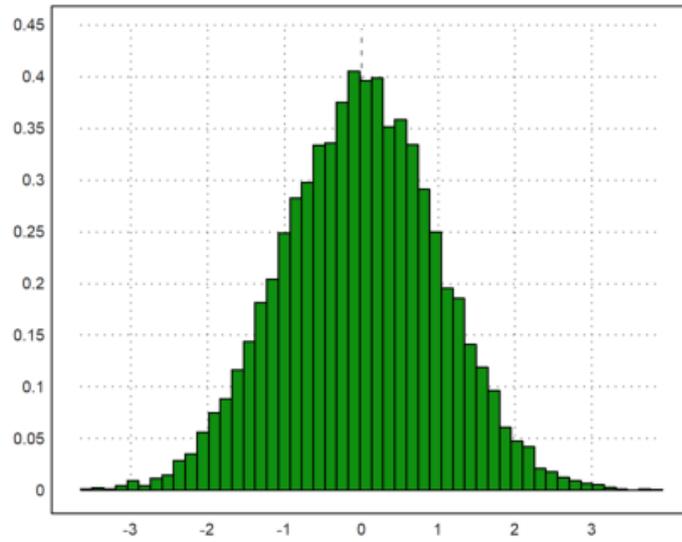
Gaya pengisian dapat digunakan seperti di atas.

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

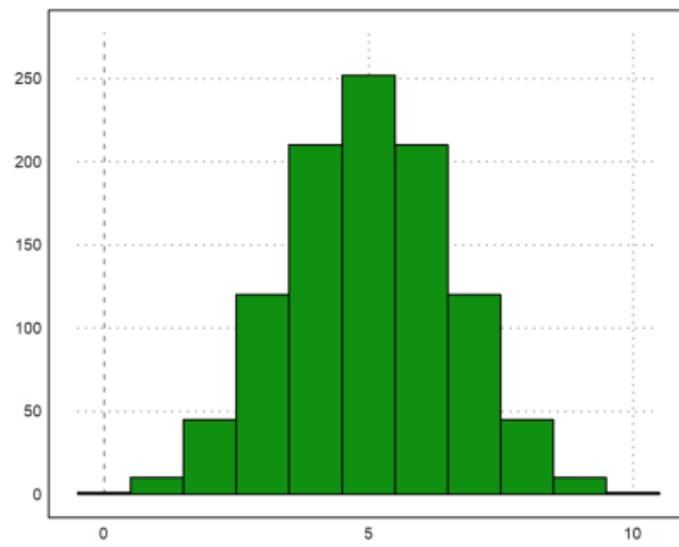


Data untuk plot batang (`bar=1`) dan histogram (`histogram=1`) dapat diberikan secara eksplisit dalam `xv` dan `yv`, atau dapat dihitung dari distribusi empiris dalam `xv` dengan `>distribution` (atau `distribution=n`). Histogram dari nilai `xv` akan dihitung secara otomatis dengan `>histogram`. Jika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

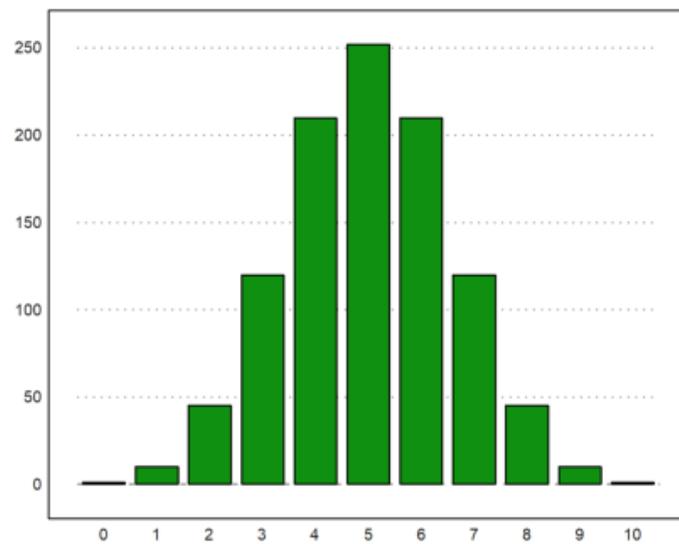
```
>plot2d(normal(10000),distribution=50):
```



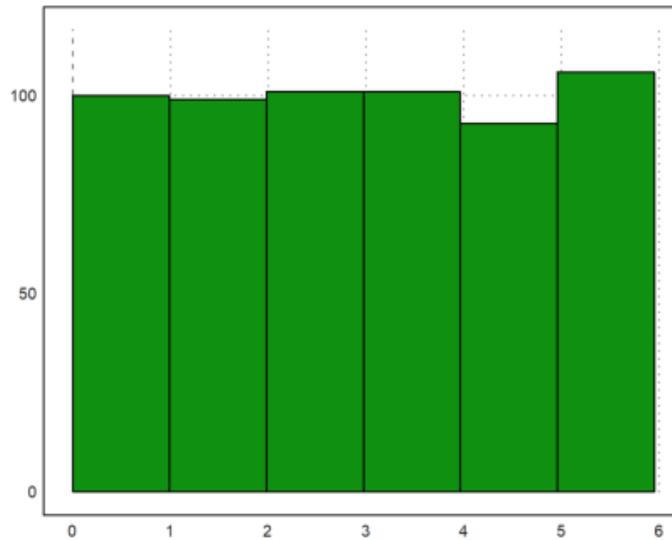
```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar) :
```



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

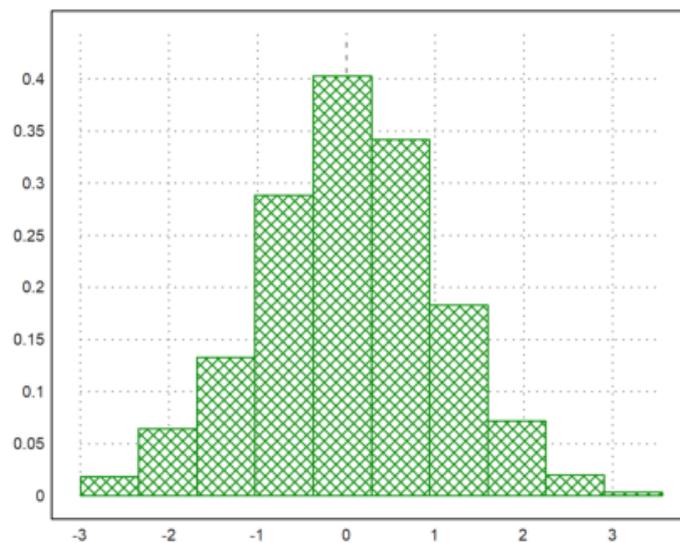


```
>plot2d(random(600)*6,histogram=6):
```



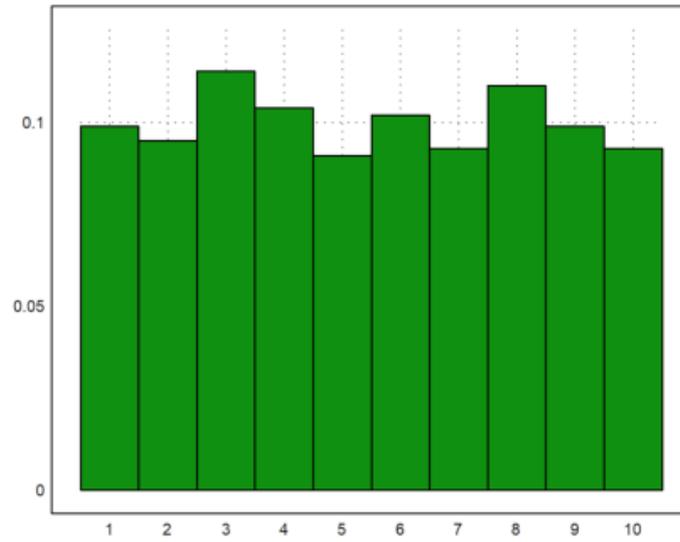
Untuk distribusi, ada parameter ``distribution=n``, yang secara otomatis menghitung nilai dan mencetak distribusi relatif dengan `n` sub-interval.

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



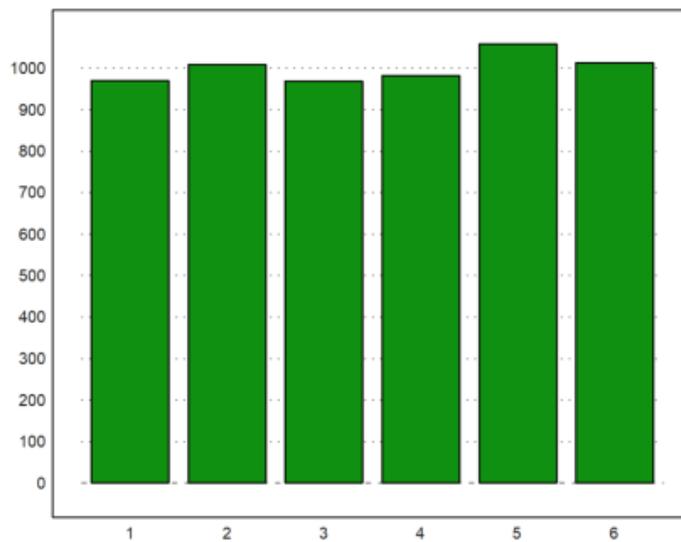
Dengan parameter ``even=true``, ini akan menggunakan interval bilangan bulat.

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

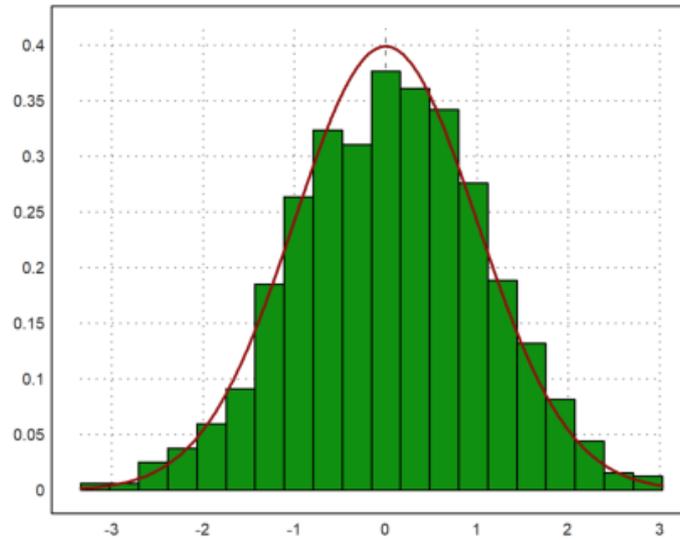


Perhatikan bahwa ada banyak plot statistik yang mungkin berguna. Silakan lihat tutorial tentang statistik.

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

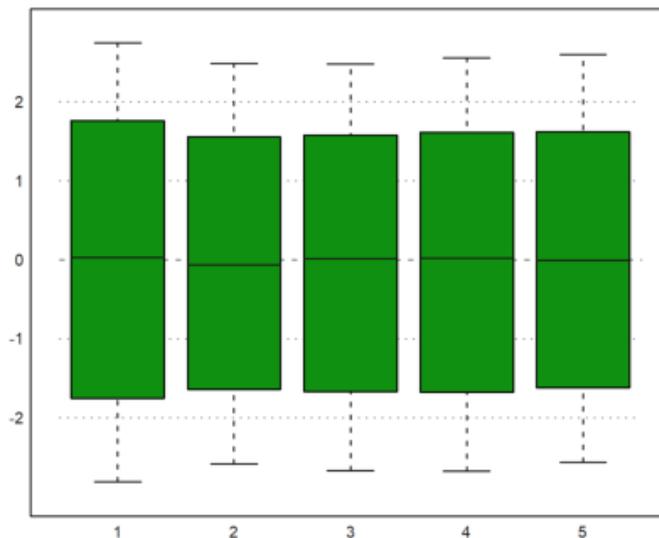


```
>plot2d(normal(1,1000),>distribution); ...
  plot2d("qnormal(x)",color=red,thickness=2,>add):
```



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

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



## Fungsi Implisit

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

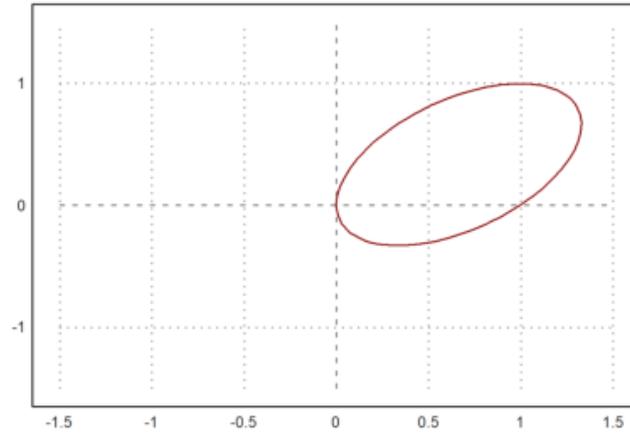
Euler dapat menandai garis level

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

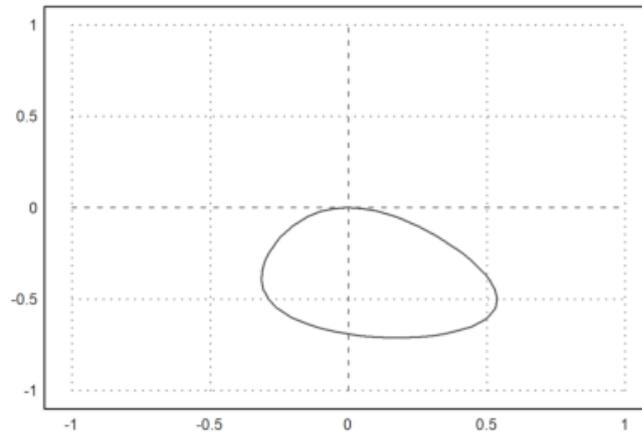
dari fungsi manapun.

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

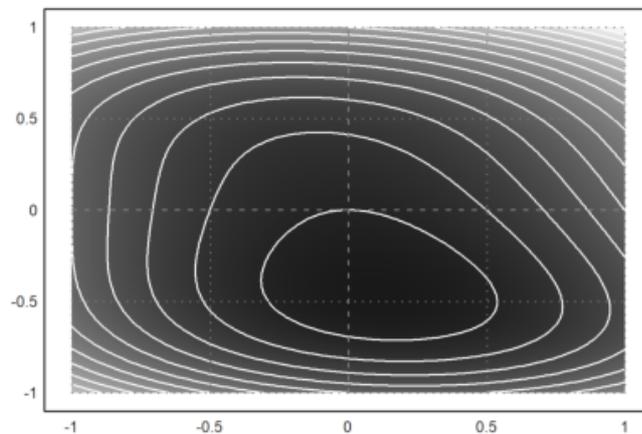
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x", r=1.5, level=0, contourcolor=red):
```



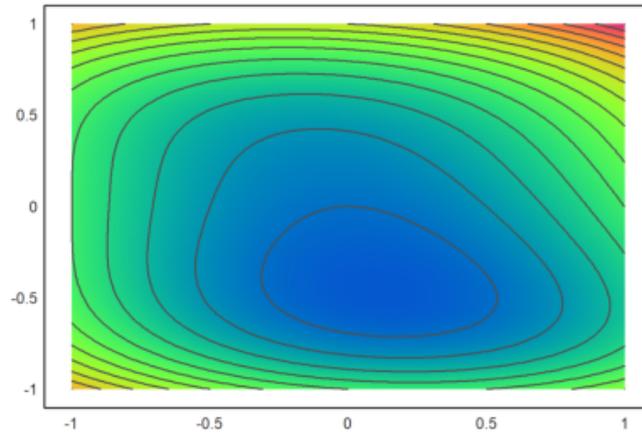
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr, level=0): // Solutions of f(x,y)=0
```



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

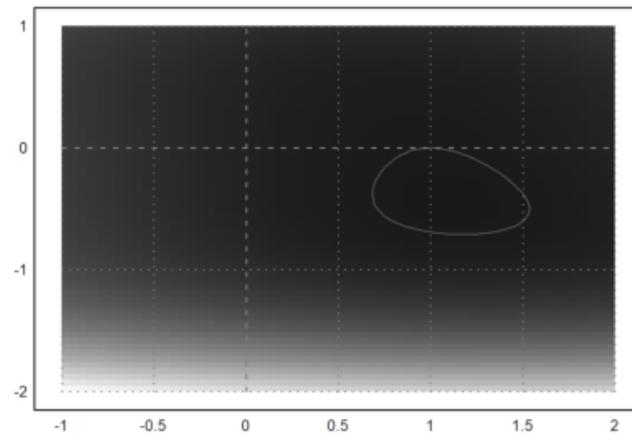


```
>plot2d(expr, level=0:0.5:20, >hue, >spectral, n=200, grid=4): // nicer
```

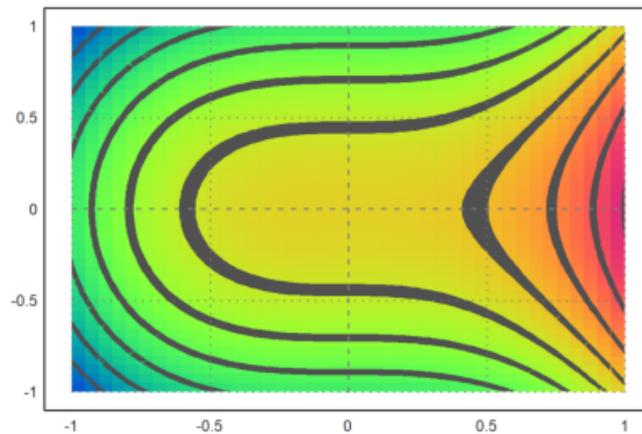


This works for data plots too. But you will have to specify the ranges for the axis labels.

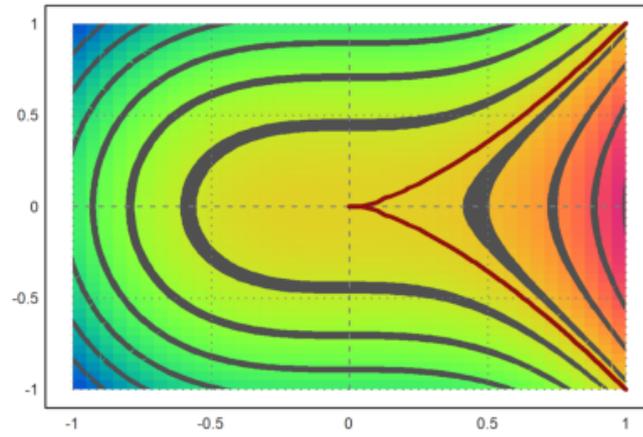
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



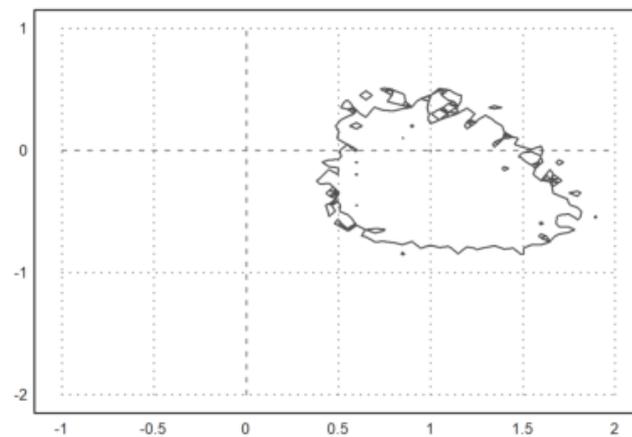
```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



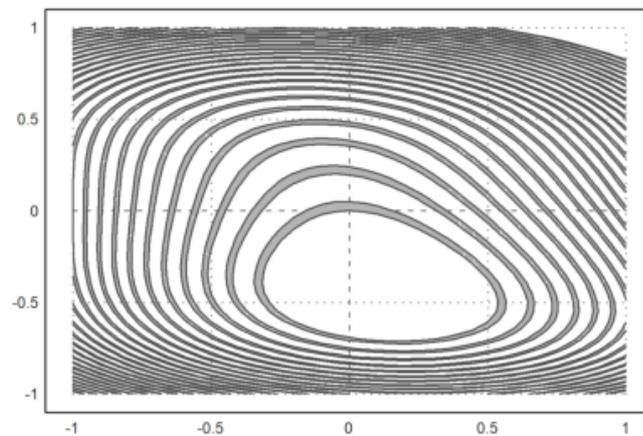
```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



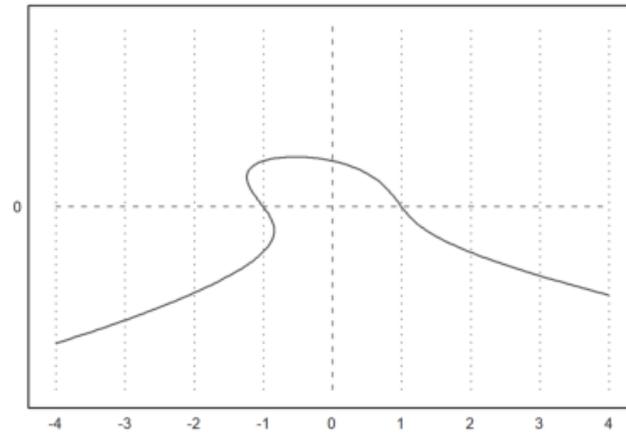
```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



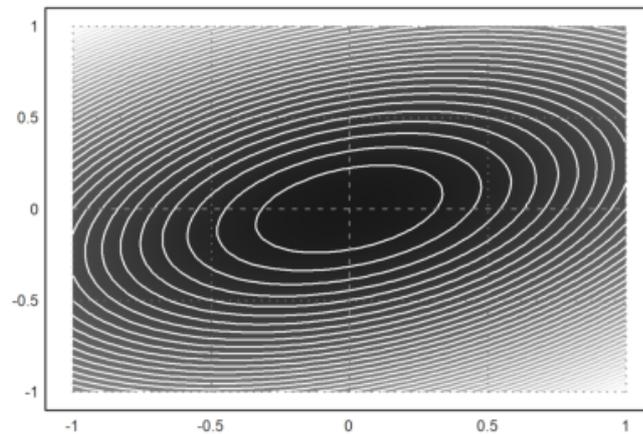
```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```



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



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



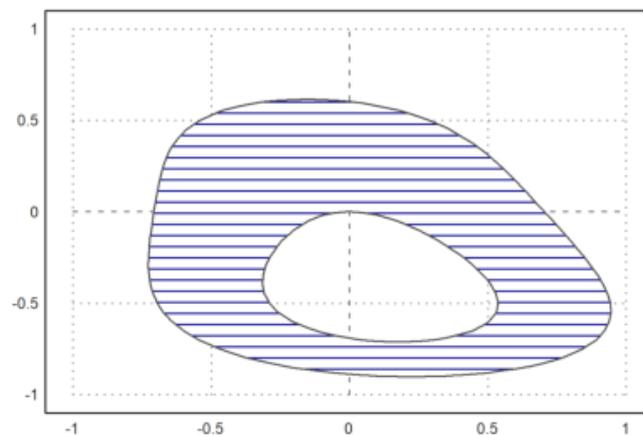
Juga dimungkinkan untuk mengisi himpunan

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

dengan rentang level.

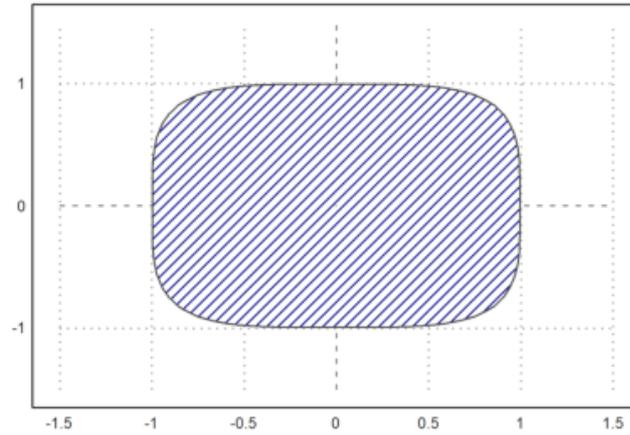
Anda dapat mengisi daerah nilai untuk fungsi tertentu. Untuk ini, `level` harus berupa matriks 2xn. Baris pertama adalah batas bawah, dan baris kedua berisi batas atas.

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

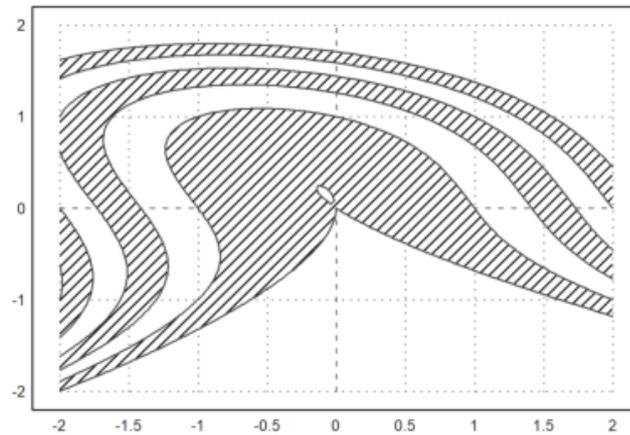


Plot implisit juga dapat menunjukkan rentang level. Dalam hal ini, `level` harus berupa matriks 2xn dari interval level, di mana baris pertama berisi awal setiap interval dan baris kedua berisi akhir setiap interval. Sebagai alternatif, vektor baris sederhana dapat digunakan untuk `level`, dan parameter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

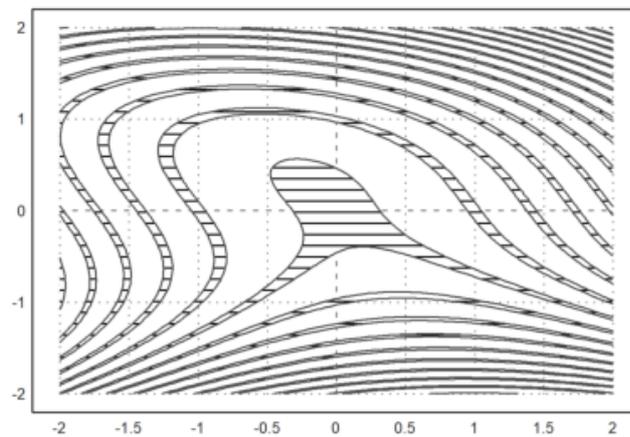
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



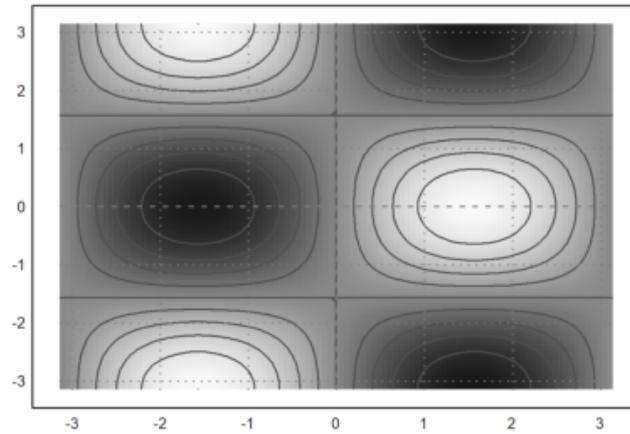
```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```



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



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

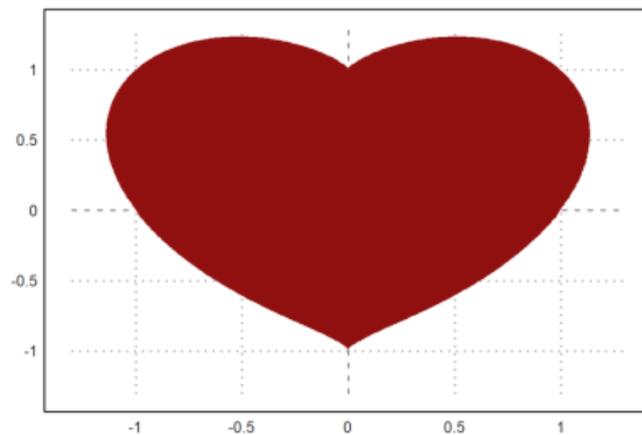


Juga dimungkinkan untuk menandai sebuah daerah

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

Ini dilakukan dengan menambahkan level yang terdiri dari dua baris.

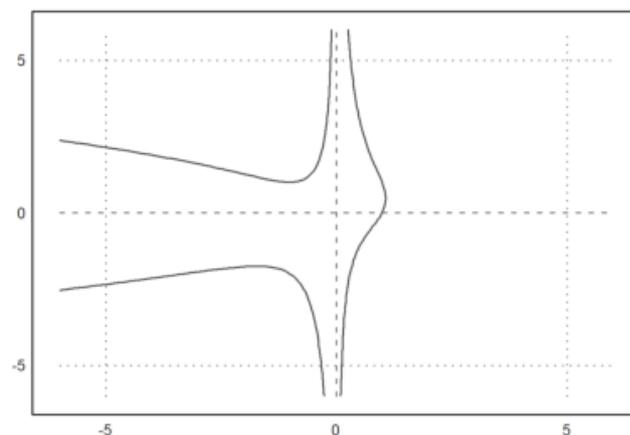
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
  style="#",color=red,<outline, ...
  level=[-2;0],n=100):
```



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

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

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



```

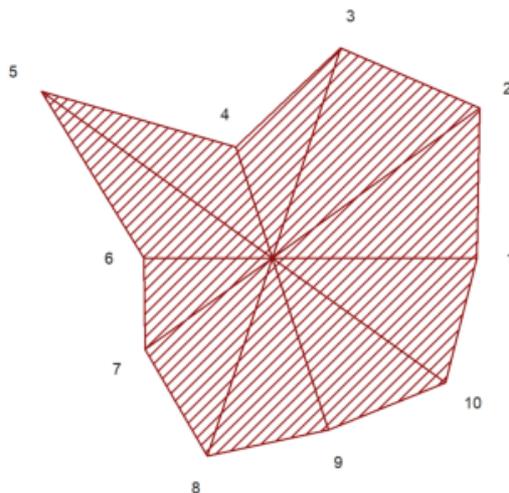
>function starplot1 (v, style="/", color=green, lab=none) ...
  if !holding() then clg; endif;
  w=window(); window(0,0,1024,1024);
  h=holding(1);
  r=max(abs(v))*1.2;
  setplot(-r,r,-r,r);
  n=cols(v); t=linspace(0,2pi,n);
  v=v|v[1]; c=v*cos(t); s=v*sin(t);
  cl=barcolor(color); st=barstyle(style);
  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 grid atau tanda sumbu di sini. Selain itu, kita menggunakan seluruh jendela untuk plot.

Kita memanggil `reset` sebelum menguji plot ini untuk mengembalikan pengaturan grafis ke default. Ini tidak diperlukan jika Anda yakin bahwa plot Anda berfungsi dengan baik.

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



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

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

```

>function logimpulseplot1 (x,y) ...
  {x0,y0}=makeimpulse(x,log(y)/log(10));
  plot2d(x0,y0,>bar,grid=0);
  h=holding(1);
  frame();
  xgrid(ticks(x));
  p=plot();
  for i=-10 to 10;
    if i<=p[4] and i>=p[3] then

```

```

        ygrid(i,yt="10^"+i);
    endif;
end;
holding(h);
endfunction

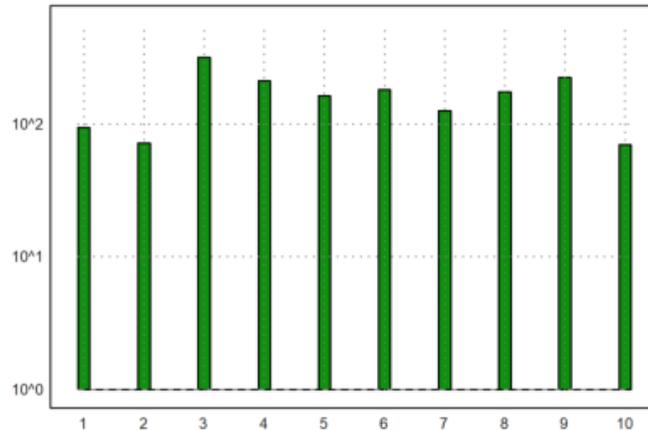
```

Mari kita uji dengan nilai yang terdistribusi secara eksponensial.

```

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

```



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

Fungsi `wait(0)` memaksa plot untuk muncul di jendela grafis. Jika tidak, penggambaran ulang akan 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

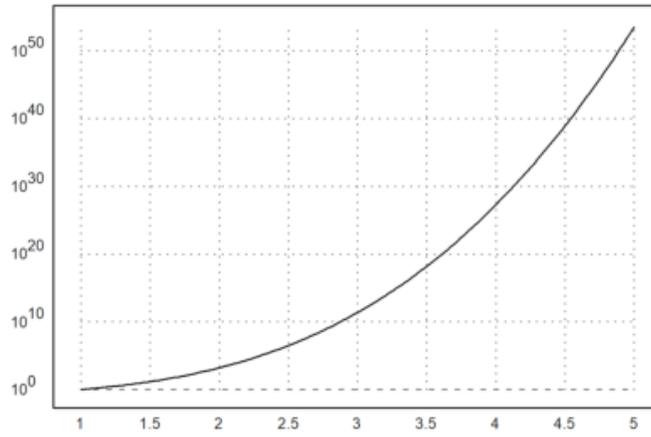
EMT menggunakan parameter `logplot` untuk skala logaritmik. Plot logaritmik dapat digambarkan dengan menggunakan skala logaritmik pada sumbu y, x, atau keduanya.

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

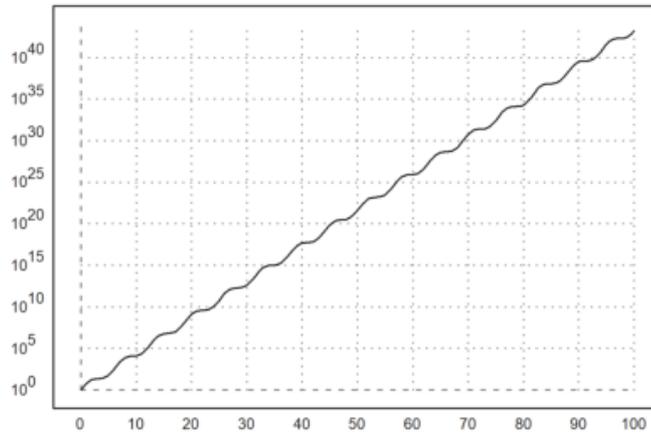
```

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

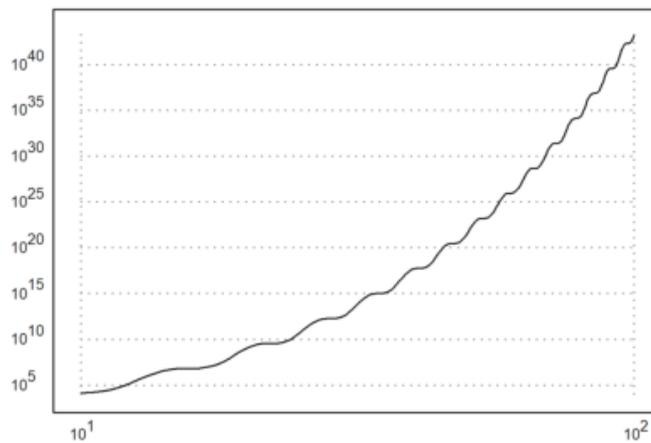
```



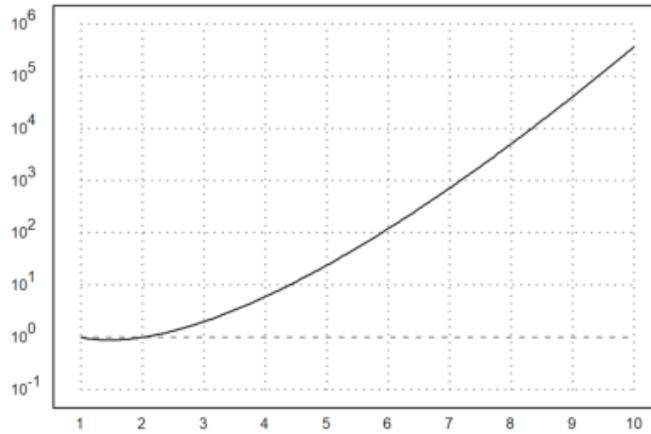
```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



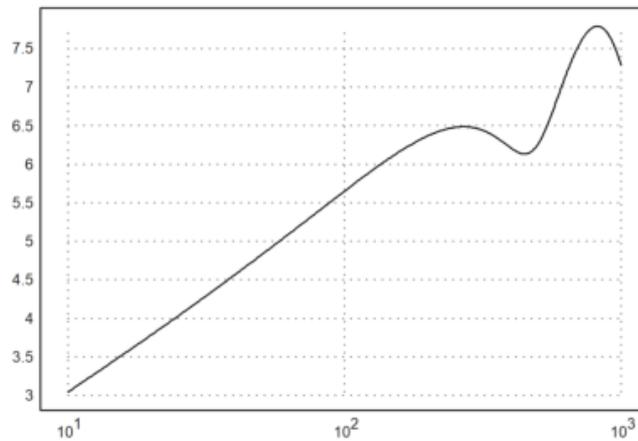
```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```



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

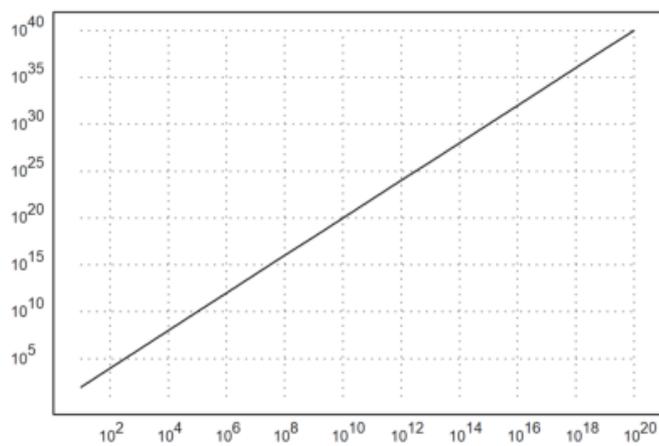


```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Plot logaritmik 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)
```

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
"#", "#0", "0", "/", "\", "\\",
"+", "|", "-", "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 [-2,2] 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.