# Menggambar Grafik 2D dengan EMT

CHOIRUL ULY NUCHA
24030130023

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

## Basic Plots

Ada fungsi-fungsi dasar dari plot. Ada koordinat layar, yang selalu berkisar dari 0 hingga 1024 pada setiap sumbu, tidak peduli apakah layarnya berbentuk persegi atau tidak. Dan ada koordinat plot, yang dapat diatur dengan setplot(). Pemetaan antara koordinat tergantung pada jendela plot saat ini. Misalnya, shrinkwindow() default memberikan ruang untuk label sumbu dan judul plot.

Dalam contoh ini, kami hanya menggambar beberapa garis acak dengan berbagai warna. Untuk rincian tentang fungsi-fungsi ini, pelajari fungsi inti dari EMT.

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



```
>reset;
```

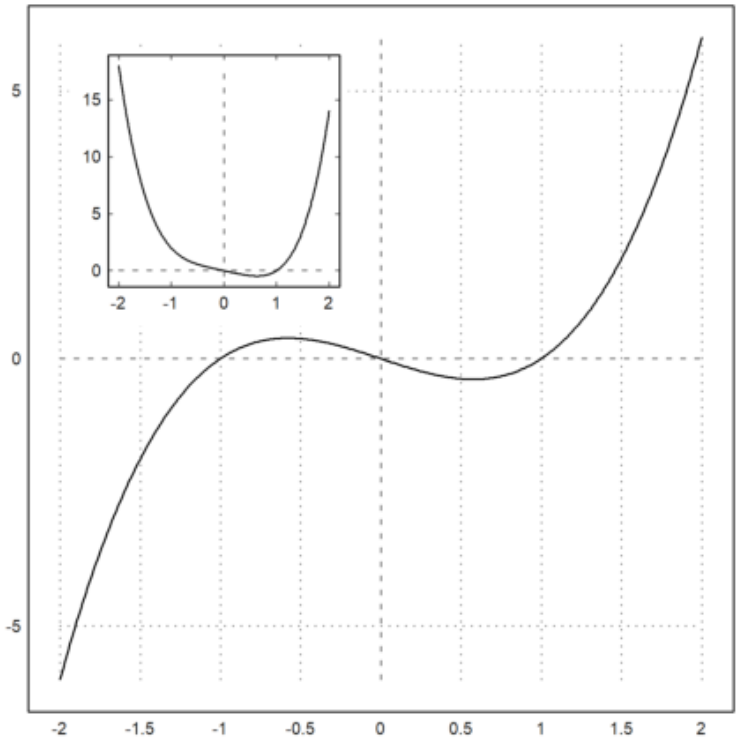
Penting untuk menahan grafik, karena perintah plot() akan membersihkan jendela plot.

Untuk membersihkan segala yang kita lakukan, kita menggunakan reset().

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

Untuk contoh lain, kita menggambar sebuah plot sebagai inset di dalam plot lainnya. Ini dilakukan dengan mendefinisikan jendela plot yang lebih kecil. Perhatikan bahwa jendela ini tidak memberikan ruang untuk label sumbu di luar jendela plot. Kita harus menambahkan sedikit margin untuk ini sesuai kebutuhan. Perhatikan bahwa kita menyimpan dan mengembalikan jendela penuh, dan menahan 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);
```

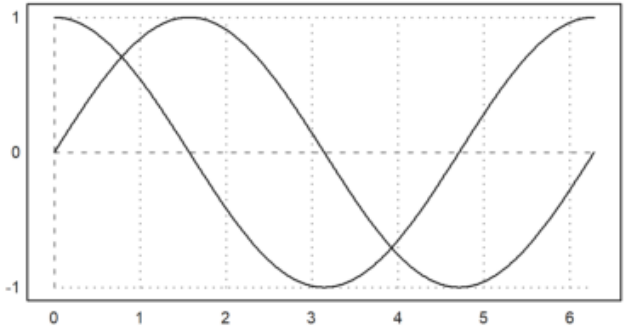
Plot dengan beberapa gambar dicapai dengan cara yang sama. Ada fungsi utilitas figure() untuk ini.

## Plot Aspect

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

Tetapi Anda juga dapat mengubahnya untuk satu plot. Untuk ini, ukuran wilayah plot saat ini diubah, dan jendela diatur sehingga label memiliki cukup ruang.

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




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

Fungsi reset() mengembalikan pengaturan default plot termasuk rasio aspek

# 2D Plots in Euler

Toolkit Matematika EMT memiliki plot dalam 2D, baik untuk data maupun fungsi. EMT menggunakan fungsi plot2d.
Fungsi ini dapat memplot fungsi dan data.Memungkinkan untuk memplot di Maxima menggunakan Gnuplot atau di Python menggunakan Math Plot Lib.
Euler dapat memplot plot 2D dari

- ekspresi
- fungsi, variabel, atau kurva terparameter,
- vektor nilai x-y,
- kumpulan titik di bidang,
- kurva implisit dengan level atau daerah level.
- Fungsi kompleks

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

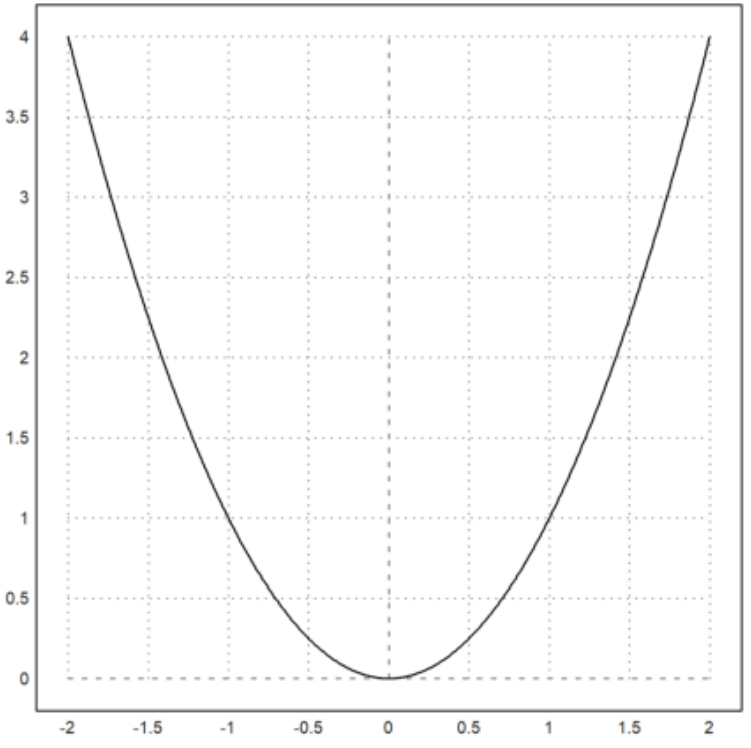
## Plot Ekspresi atau Variabel

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

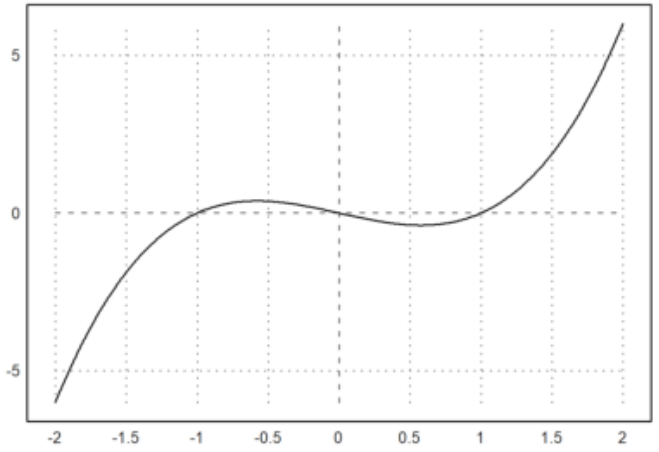
Berikut adalah contoh paling dasar, yang menggunakan rentang default dan mengatur rentang y yang tepat agar sesuai dengan plot fungsi.

Catatan: Jika Anda mengakhiri baris perintah dengan tanda titik dua ":", plot akan dimasukkan ke dalam jendela teks. Jika tidak, tekan TAB untuk melihat plot jika jendela plot tertutup.

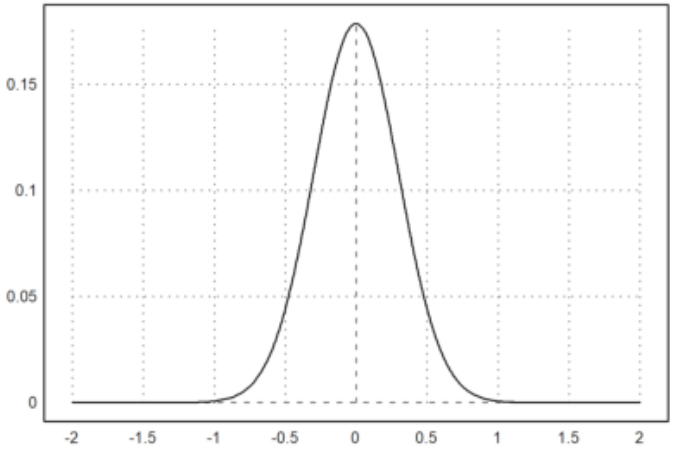
```
>plot2d("x^2"):
```




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



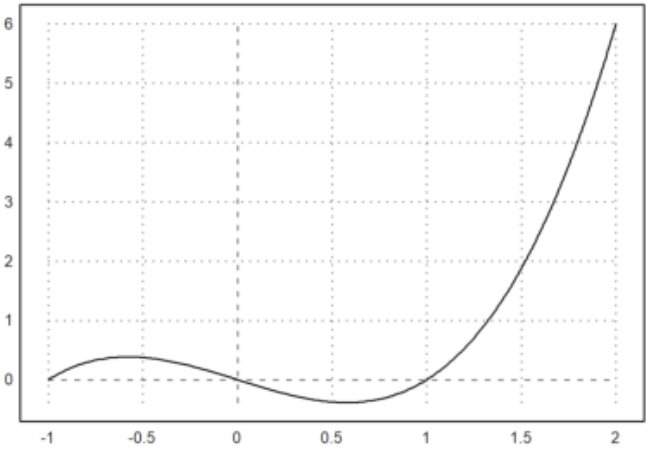

```
>a:=5.6; plot2d("exp(-a*x^2)/a"); insimg(30); // menampilkan gambar hasil plot setinggi 25 baris
```



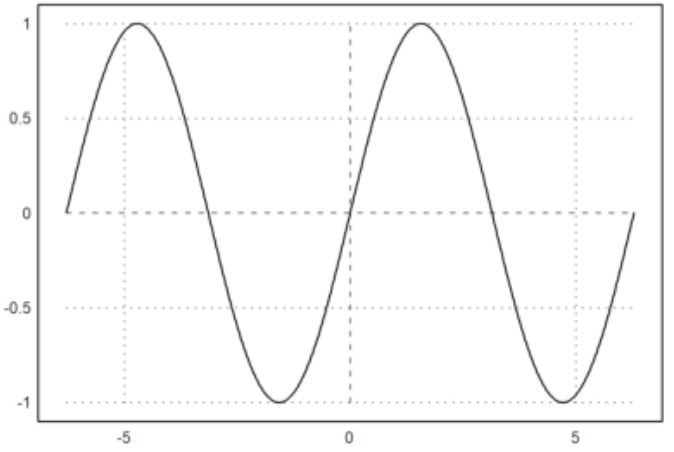

Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

Rentang plot disetel dengan parameter yang ditugaskan sebagai berikut
- a,b: rentang x (default -2,2)
- c,d: rentang y (default: skala dengan nilai)
- r: alternatif radius di sekitar pusat plot
- cx,cy: koordinat pusat plot (default 0,0)

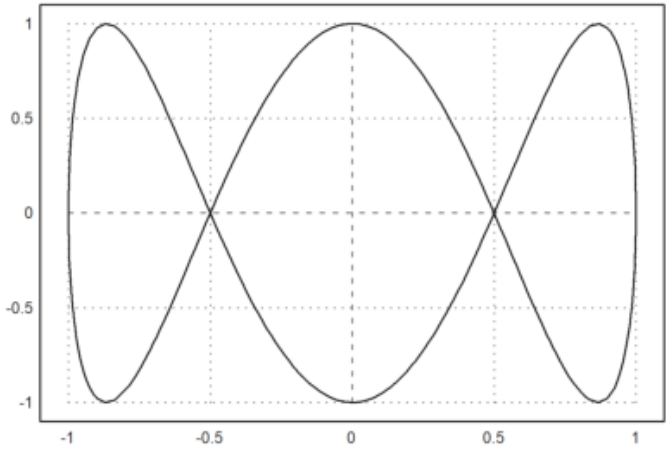
```
>plot2d("x^3-x",-1,2):
```




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




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



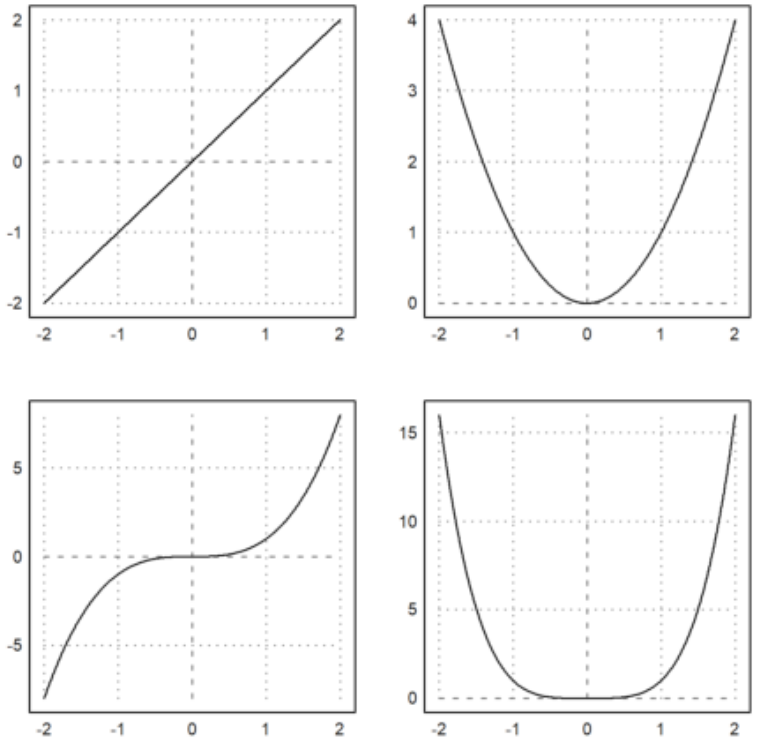

Alternatif dari tanda titik dua adalah perintah insimg(lines), yang menyisipkan grafik yang menempati sejumlah baris teks tertentu.

Dalam opsi, grafik dapat diatur untuk tampil

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

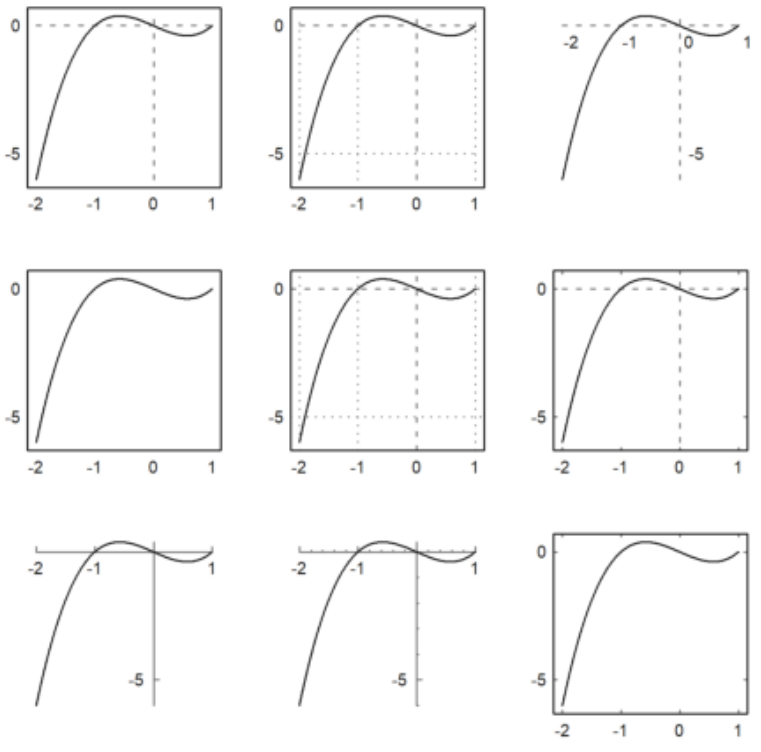
Lebih banyak gaya dapat dicapai dengan perintah grafik khusus.
Dalam kasus apa pun, tekan tombol tabulator untuk melihat grafik jika tersembunyi.Untuk membagi jendela menjadi beberapa grafik, gunakan perintah figure(). Pada contoh, kita menggambar hingga ke dalam 4 bagian jendela. figure(0) mengatur ulang jendela ke default.

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




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

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

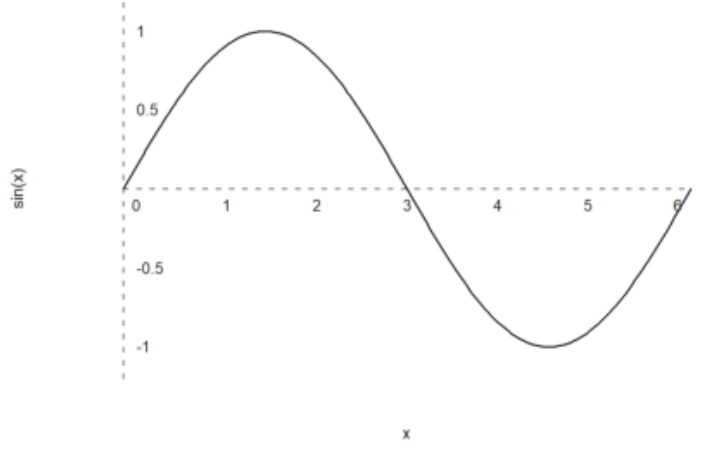


Jika argumen pada plot2d() berupa sebuah ekspresi yang diikuti oleh empat angka, maka angka-angka tersebut merupakan rentang x dan y untuk grafik.

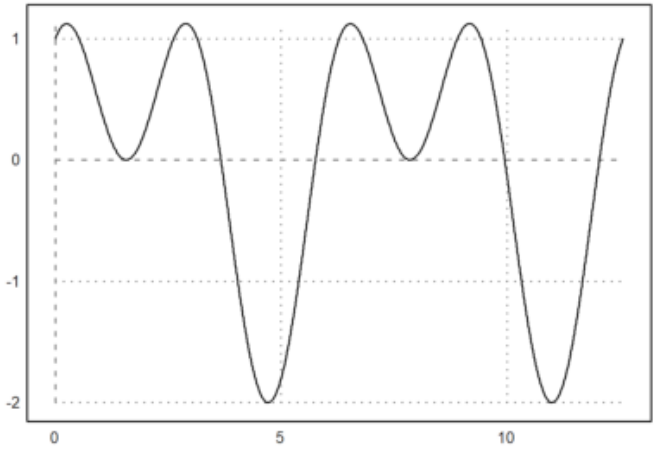
Sebagai alternatif, a, b, c, d dapat ditentukan sebagai parameter yang diberi nilai, misalnya a=... dan seterusnya.

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

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




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

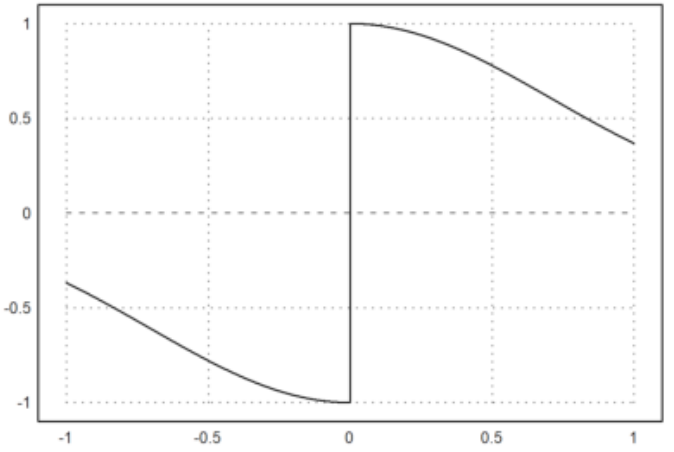



Gambar yang dihasilkan dengan menyisipkan grafik ke dalam jendela teks disimpan di direktori yang sama dengan notebook, secara default dalam subdirektori bernama ”images”. Gambar-gambar tersebut juga digunakan dalam ekspor HTML.

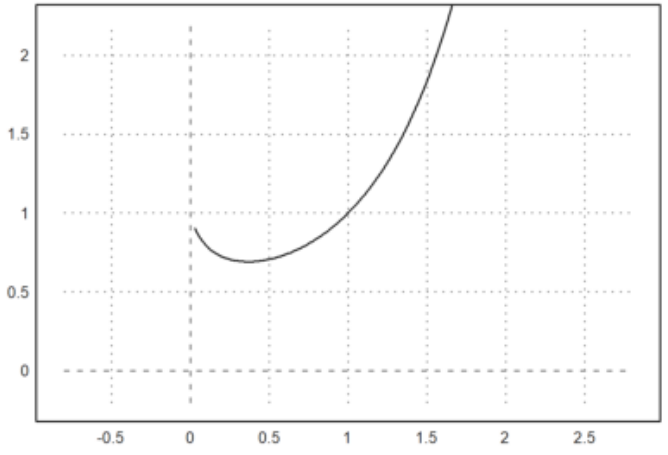
Anda dapat dengan mudah menandai gambar apa pun dan menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, Anda juga dapat mengekspor grafik yang sedang aktif menggunakan fungsi di menu File.

Fungsi atau ekspresi dalam plot2d dievaluasi secara adaptif. Untuk mendapatkan kecepatan lebih tinggi, matikan grafik adaptif dengan <adaptive dan tentukan jumlah subinterval dengan n=.... Hal ini biasanya hanya diperlukan pada kasus-kasus tertentu saja.

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

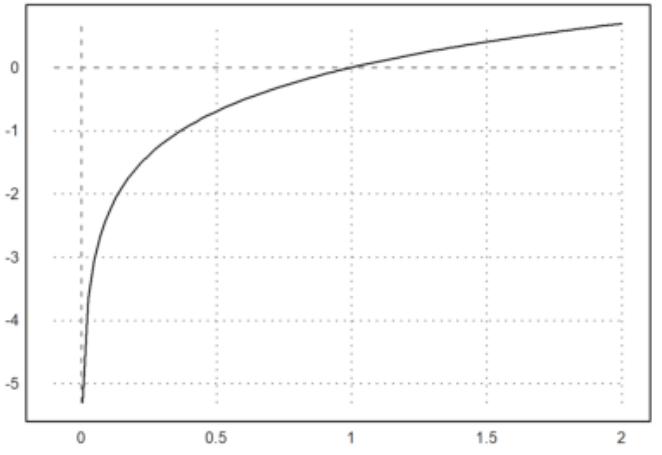



```
>plot2d("x^x",r=1.2,cx=1,cy=1):
```



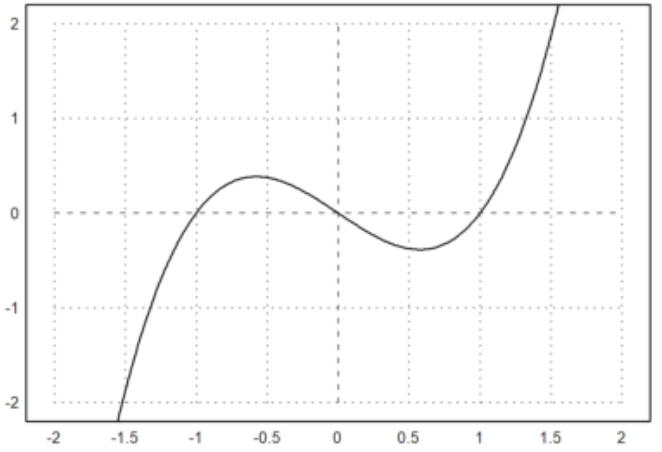

Perhatikan bahwa tidak terdefinisi untuk =0. Fungsi plot2d akan menangkap kesalahan ini, dan mulai menggambar grafik segera setelah fungsi tersebut terdefinisi. Hal ini berlaku untuk semua fungsi yang menghasilkan NAN di luar daerah definisinya.

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

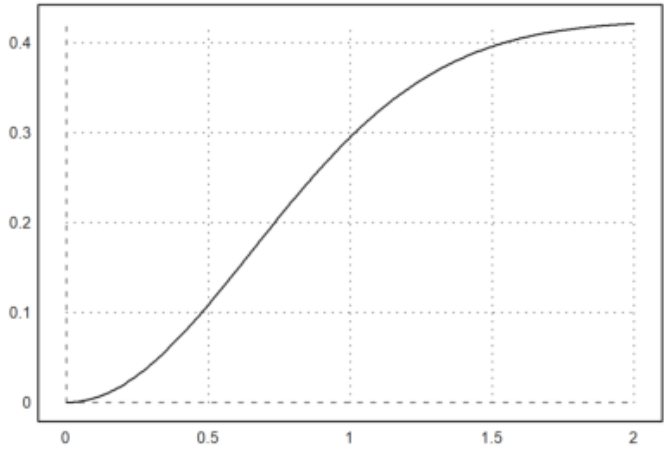



Parameter square=true (atau >square) akan secara otomatis memilih rentang y sehingga hasilnya berupa jendela grafik berbentuk persegi. Perlu dicatat bahwa secara default, Euler menggunakan ruang persegi di dalam jendela grafik.

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

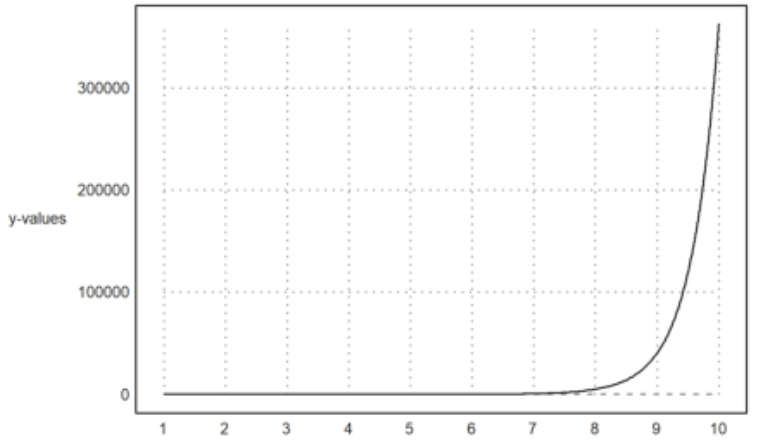



```
>plot2d(''integrate("sin(x)*exp(-x^2)",0,x)'',0,2): // plot integral
```



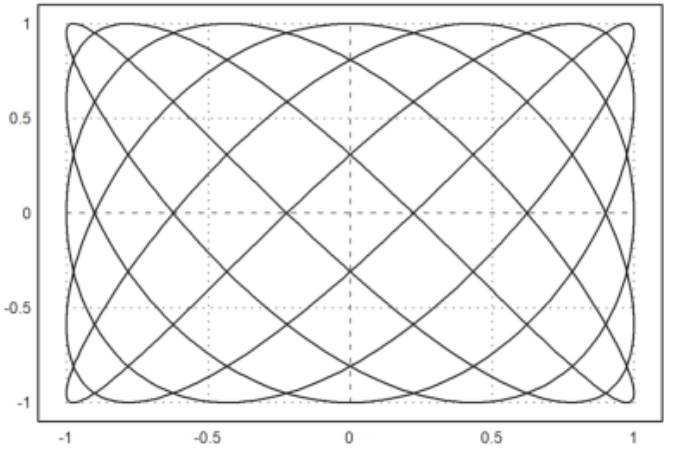

Jika Anda membutuhkan lebih banyak ruang untuk label sumbu y, panggil shrinkwindow() dengan parameter smaller, atau atur nilai positif untuk ”smaller” di dalam plot2d().

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

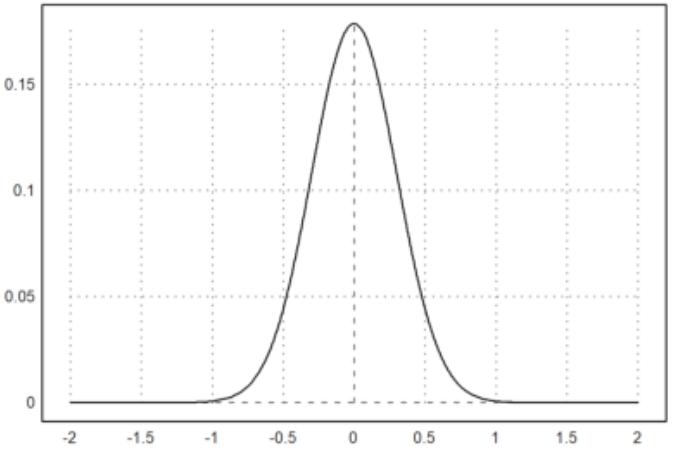



Ekspresi simbolik juga dapat digunakan, karena disimpan sebagai ekspresi string sederhana.

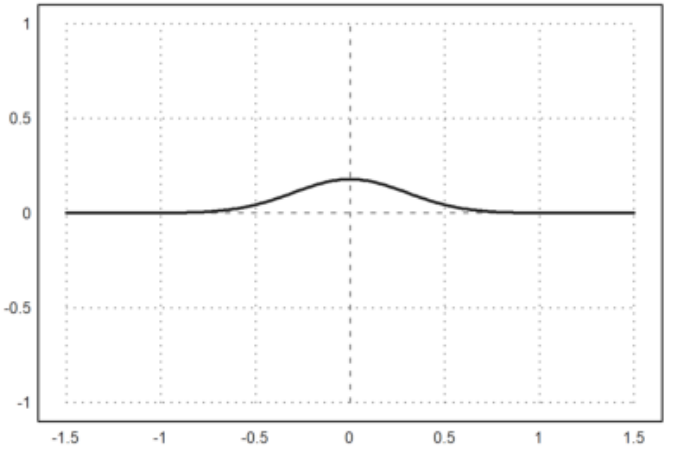
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



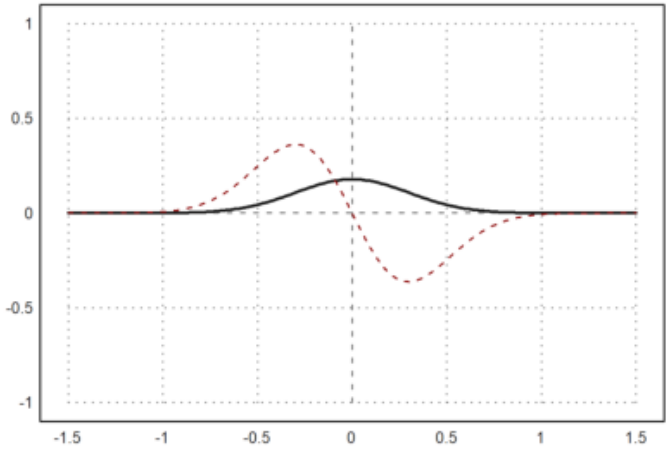

```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression
>plot2d(expr,-2,2): // plot from -2 to 2
```



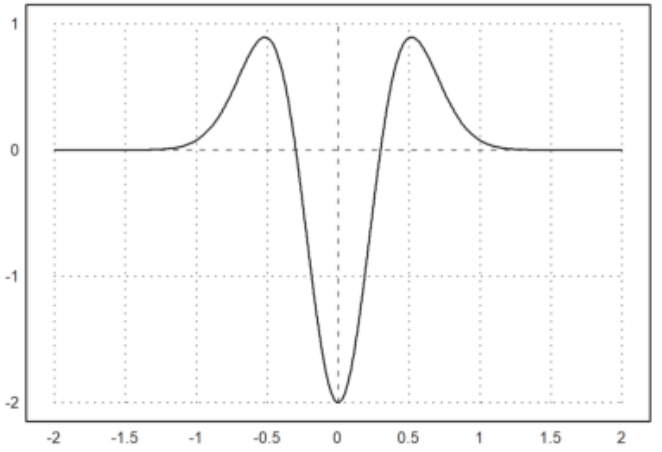

```
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)
```



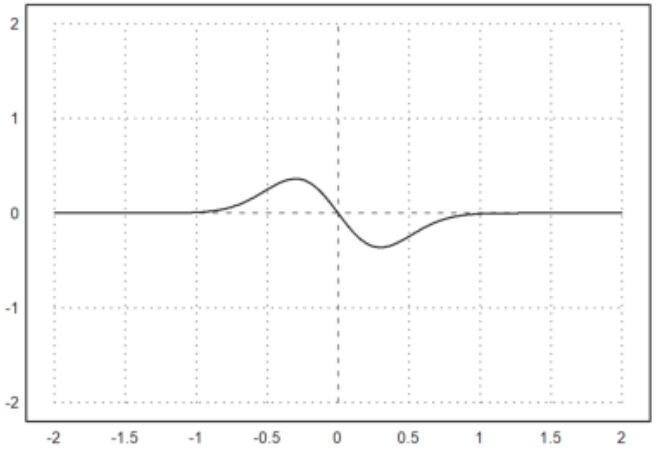

```
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot
```



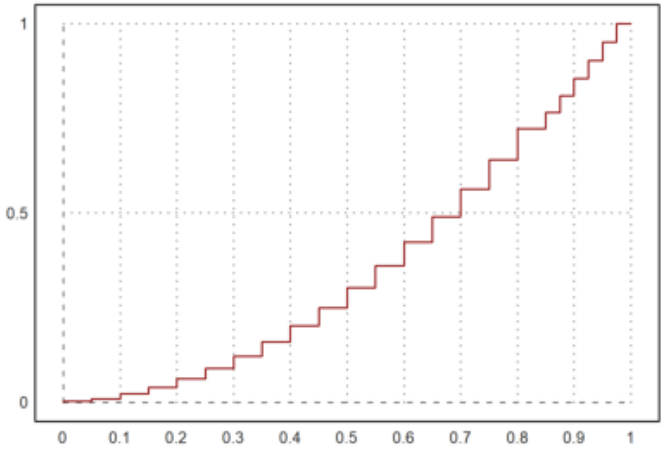

```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle
```



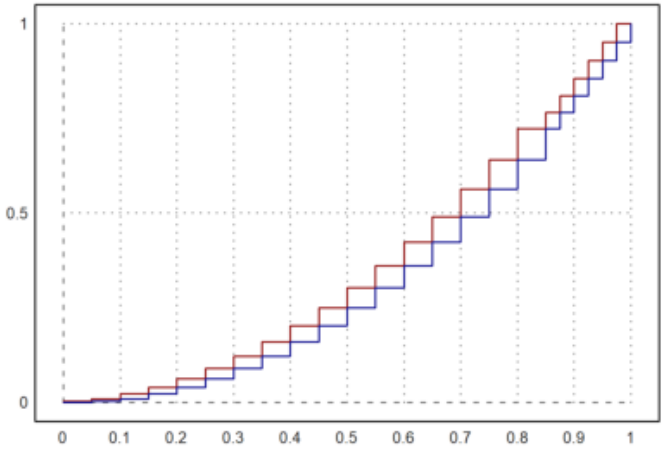

```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square
```




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




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

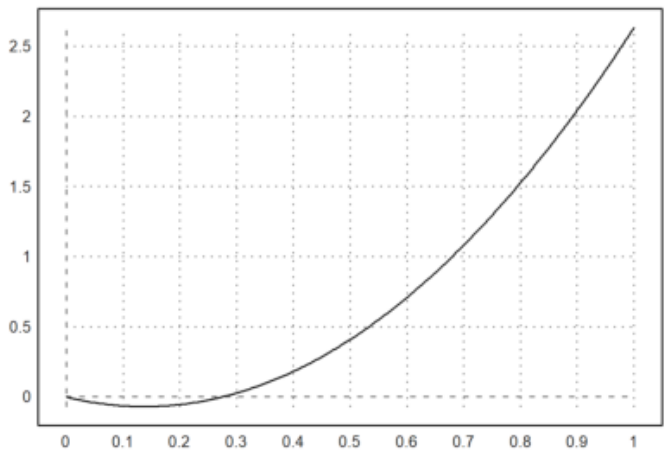

1
0.5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

# Functions in one Parameter

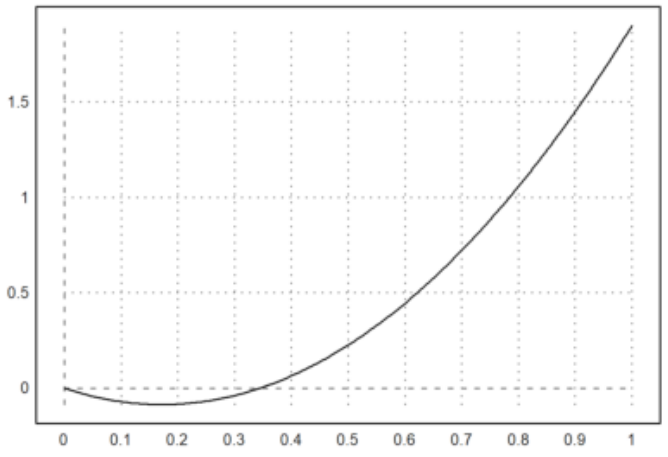
Fungsi penggambaran paling penting untuk grafik bidang adalah plot2d(). Fungsi ini diimplementasikan dalam bahasa Euler pada file ”plot.e”, yang dimuat saat program dijalankan.

Berikut beberapa contoh penggunaan dengan sebuah fungsi. Seperti biasa dalam EMT, fungsi-fungsi yang bekerja untuk fungsi lain atau ekspresi dapat diberikan parameter tambahan (selain x) yang bukan variabel global, dengan menggunakan parameter titik koma atau dengan pemanggilan koleksi.

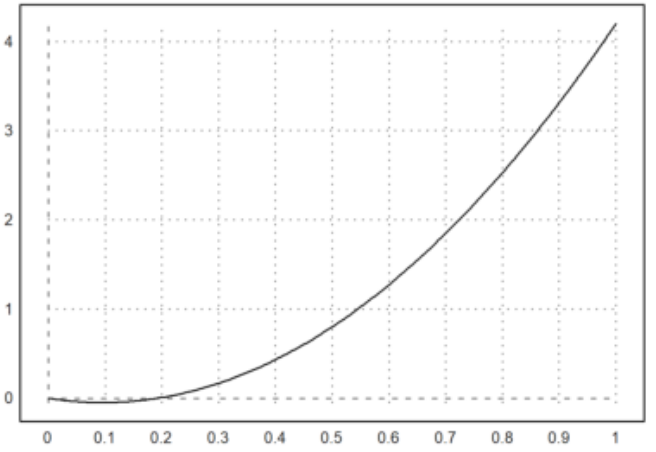
```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
>a=0.3; plot2d("f",0,1;a): // plot with a=0.3
```



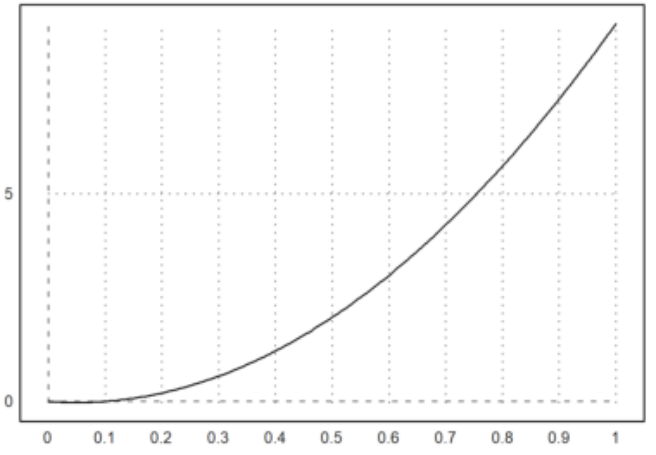
```
>plot2d("f",0,1;0.4): // plot with a=0.4
```



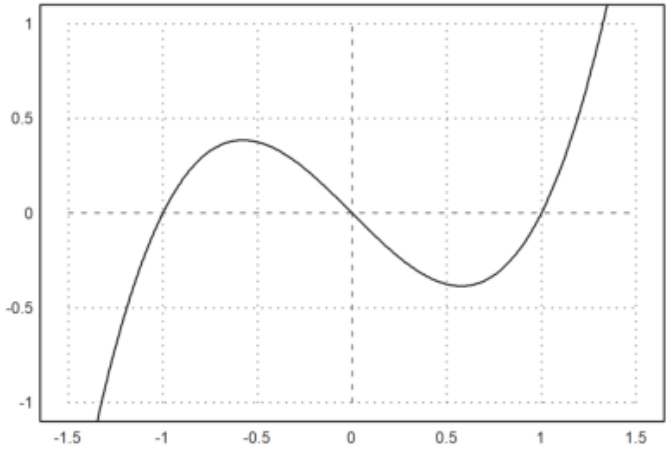

```
>plot2d({{"f",0.2}},0,1): // plot with a=0.2
```




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




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




Berikut adalah ringkasan fungsi yang dapat diterima:

ekspresi atau ekspresi simbolik dalam x

fungsi atau fungsi simbolik dengan nama seperti "f"

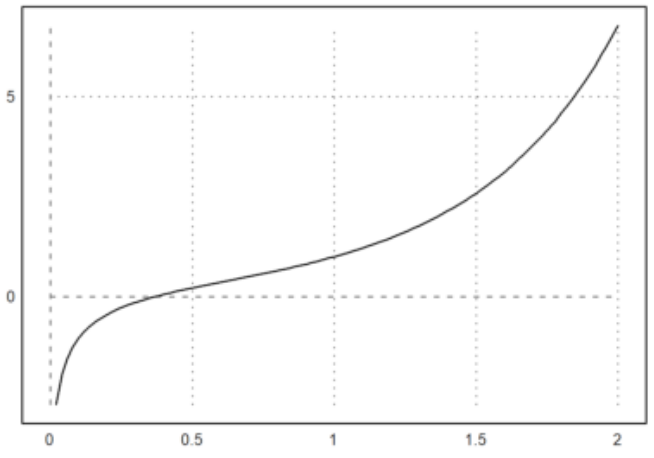
fungsi simbolik hanya dengan menyebutkan nama f

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

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

$$x^{x} \, (\log(x) + 1)$$

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

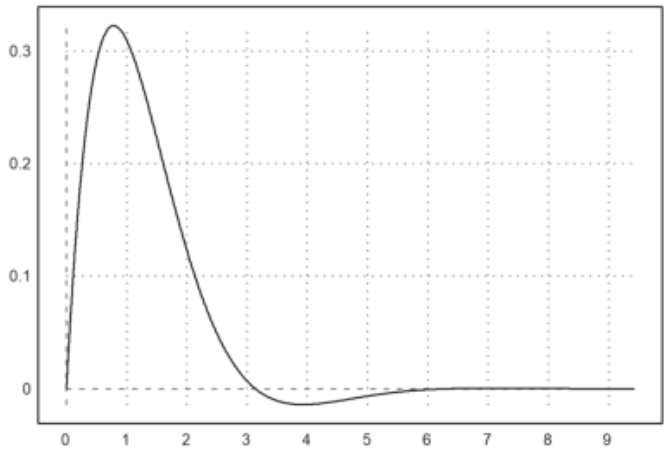



Tentu saja, untuk ekspresi atau ekspresi simbolik, nama variabel saja sudah cukup untuk menggambarkannya.

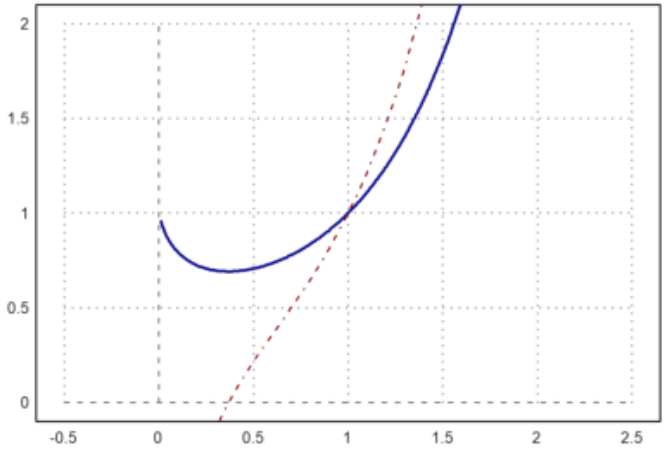
```
>expr &= sin(x)*exp(-x)
```

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

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




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



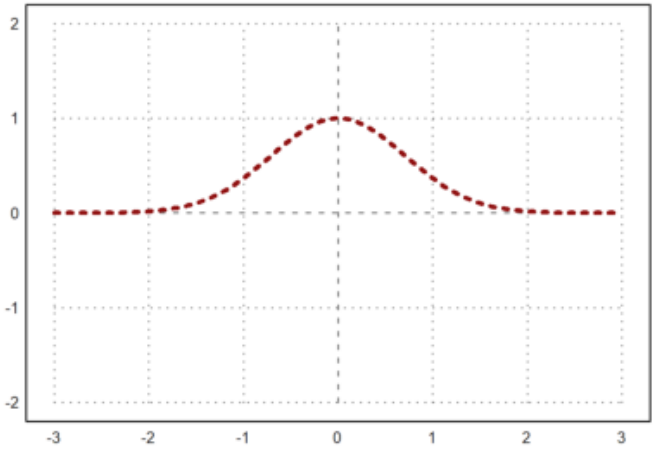

For the line style there are various options.

- style="...". Pilih dari "-", "–", "-.", ".", ".-.", "-.-".
- color: Lihat daftar warna di bawah.
- thickness: Nilai default adalah 1.

Warna dapat dipilih dari warna default atau menggunakan warna RGB.

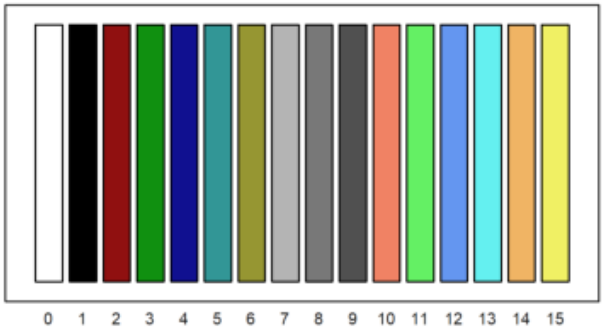
- 0..15: indeks warna default.
- color constants: white, black, red, green, blue, cyan, olive, lightgray, gray, darkgray, orange, lightgreen, turquoise, lightblue, lightorange, yellow
- rgb(red,green,blue): parameternya berupa bilangan real dalam interval [0,1].

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



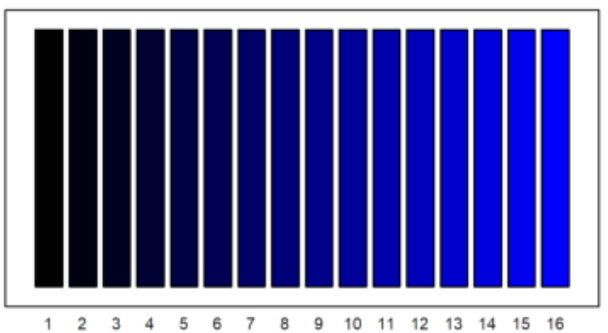

Berikut adalah tampilan warna-warna yang telah didefinisikan sebelumnya di EMT.

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




But you can use any color.

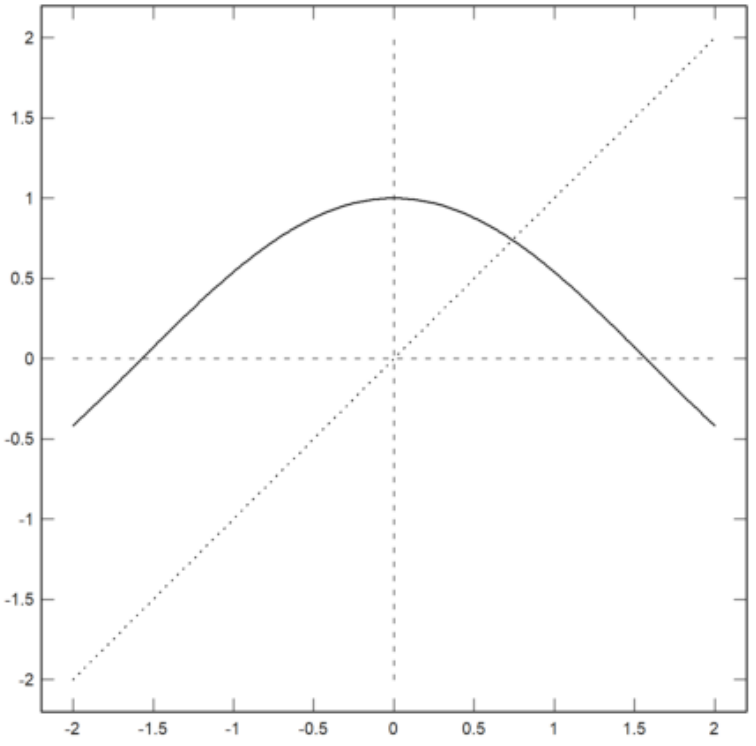
```
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```



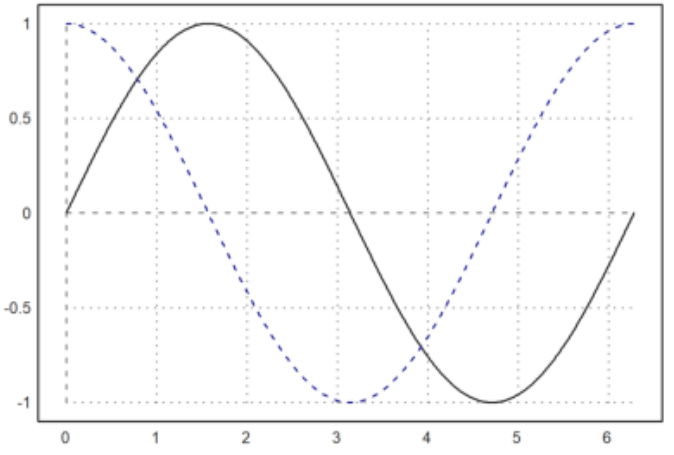
## Menggambar Beberapa Kurva pada bidang koordinat yang sama

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

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

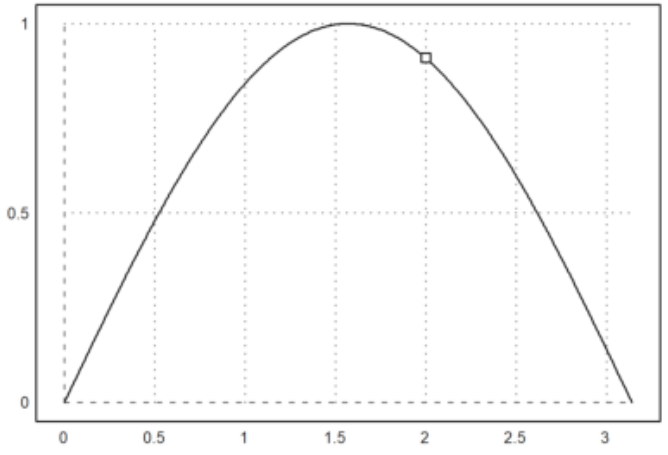



```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



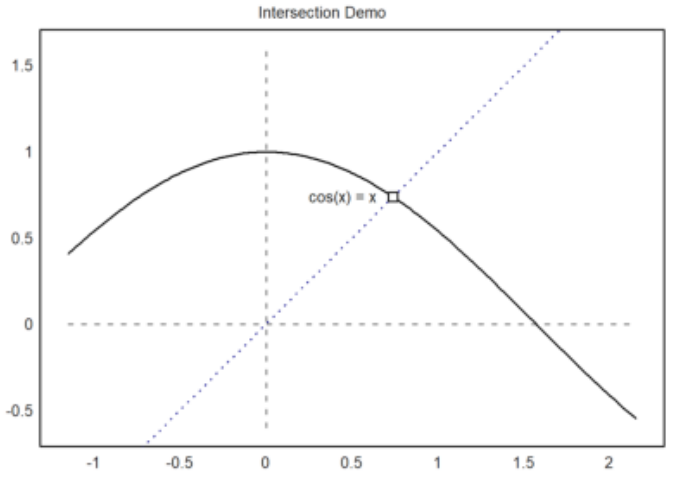

Salah satu kegunaan >add adalah untuk menambahkan titik pada kurva.

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




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

```
>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):
```



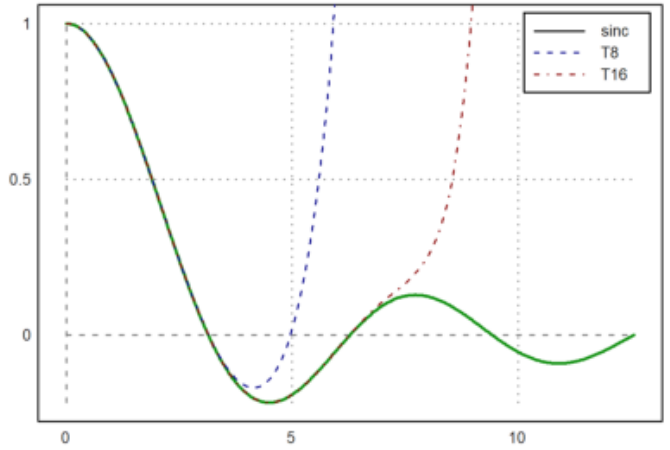

Pada demo berikut, kita menggambar fungsi sinc(x) = sin(x)/x beserta perluasan Taylor ke-8 dan ke-16. Perluasan ini dihitung menggunakan Maxima melalui ekspresi simbolik.
Grafik ini dibuat dengan perintah multi-baris yang berisi tiga pemanggilan plot2d(). Pemanggilan kedua dan ketiga menggunakan tanda >add, sehingga grafik menggunakan rentang yang sama dengan pemanggilan sebelumnya.

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

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

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

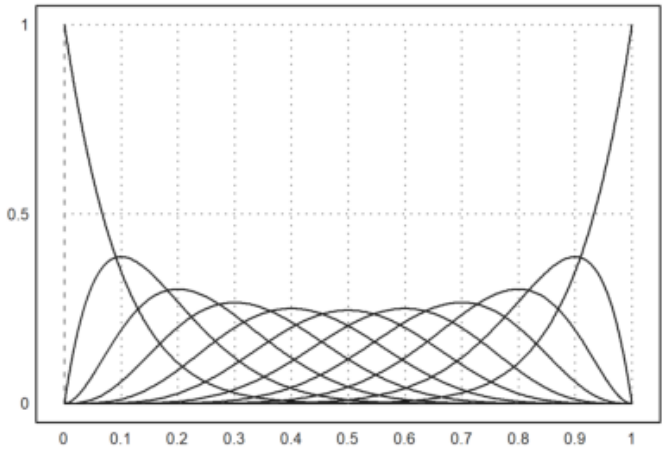
```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
>  plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
>  plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-.-"); ...
>  labelbox(["sinc","T8","T16"],styles=["-","--","-.-"], ...
>    colors=[black,blue,red]):
```




Pada contoh berikut, kita menghasilkan Polinomial Bernstein.

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

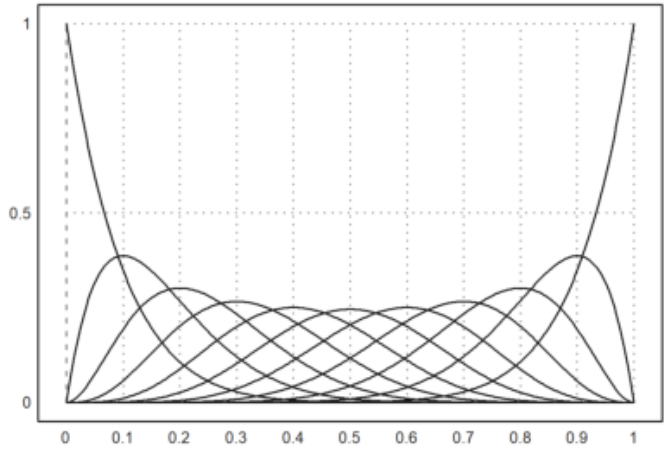
```
>plot2d("(1-x)^10",0,1); // plot first function
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;
>insimg;
```




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

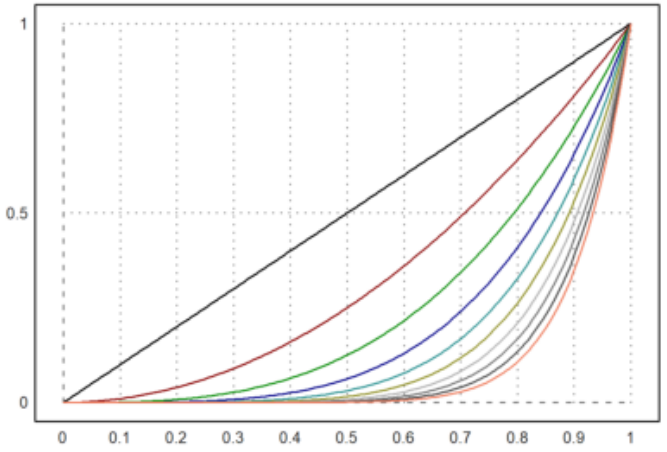
Kita membangkitkan sebuah matriks nilai dengan satu Polinomial Bernstein pada setiap barisnya. Untuk itu, kita cukup menggunakan sebuah vektor kolom dari i. Silakan lihat pada bagian pengantar tentang bahasa matriks untuk mempelajari detail lebih lanjut.

```
>x=linspace(0,1,500);
>n=10; k=(0:n)'; // n is row vector, k is column vector
>y=bin(n,k)*x^k*(1-x)^(n-k); // y is a matrix then
>plot2d(x,y):
```



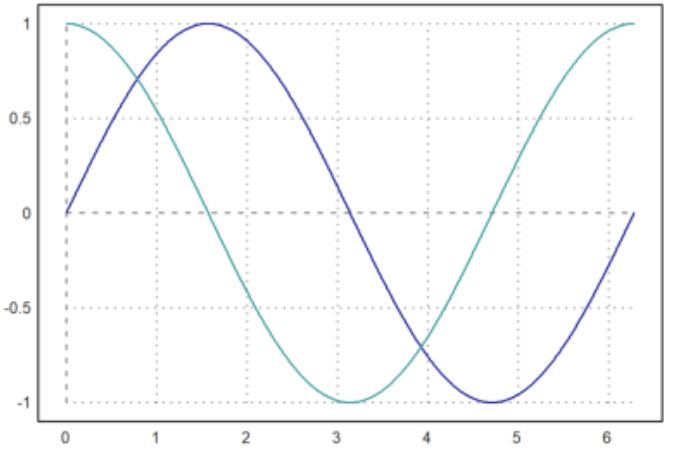

perlu dicatat bahwa parameter color dapat berupa sebuah vektor. Dengan demikian, setiap warna akan digunakan untuk setiap baris matriks.

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

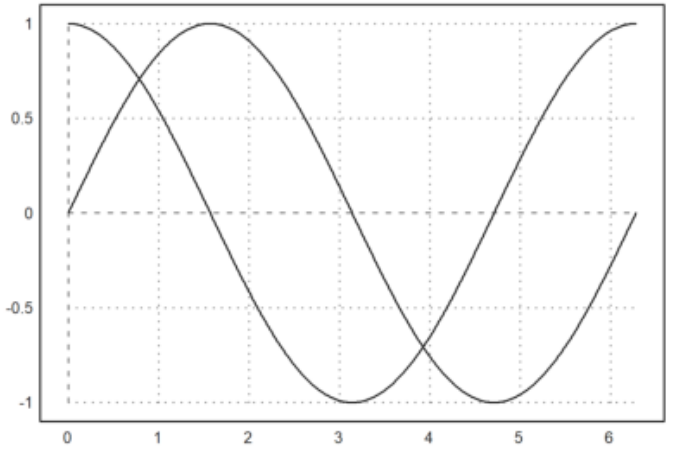



Another method is using a vector of expressions (strings). You can then use a color array, an array of styles, and an array of thicknesses of the same length.

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




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




We can get such a vector from Maxima using makelist() and mxm2str().

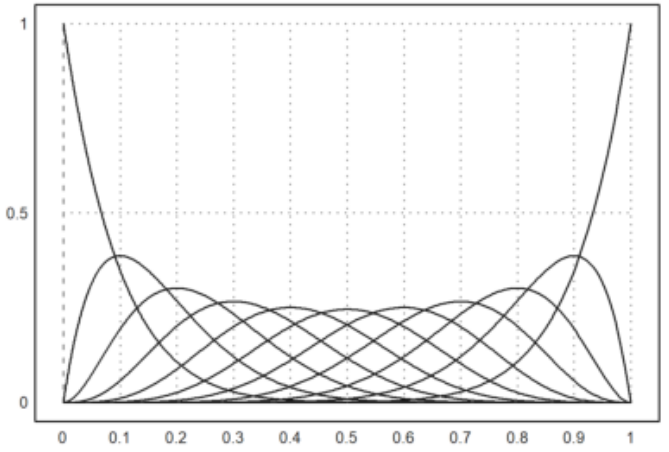
```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // make list
```

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

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

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

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

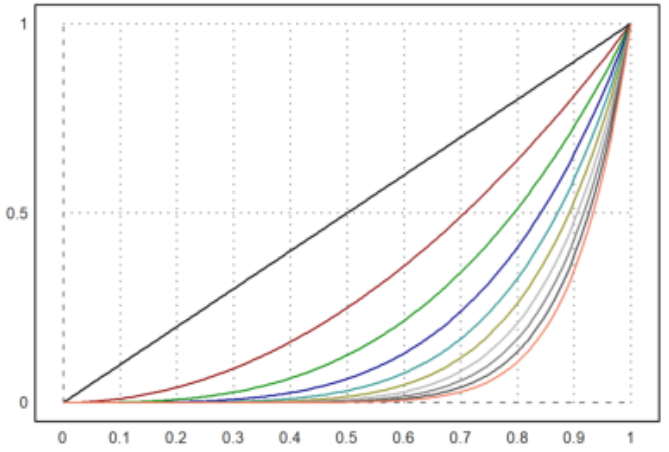



Alternatif lain adalah menggunakan bahasa matriks dari Euler.

Jika suatu ekspresi menghasilkan matriks fungsi, dengan satu fungsi pada setiap baris, maka semua fungsi tersebut akan digambar dalam satu grafik.

Untuk itu, gunakan vektor parameter dalam bentuk vektor kolom. Jika ditambahkan sebuah array warna, maka warna tersebut akan digunakan untuk setiap baris grafik.

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

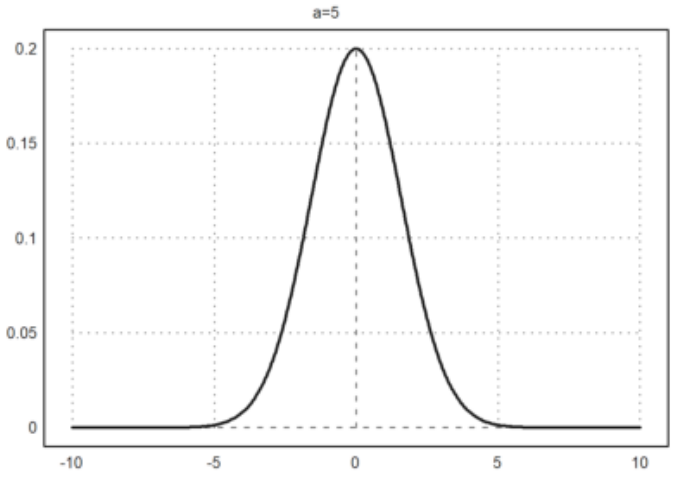



Ekspresi dan fungsi satu baris dapat mengakses variabel global.

Jika Anda tidak dapat menggunakan variabel global, Anda perlu membuat fungsi dengan parameter tambahan, lalu meneruskan parameter tersebut sebagai parameter titik koma.

Perhatikan bahwa semua parameter yang diberi nilai harus diletakkan di bagian akhir perintah plot2d. Pada contoh, kita memberikan a=5 ke fungsi f, yang kemudian digambarkan pada interval -10 hingga 10.

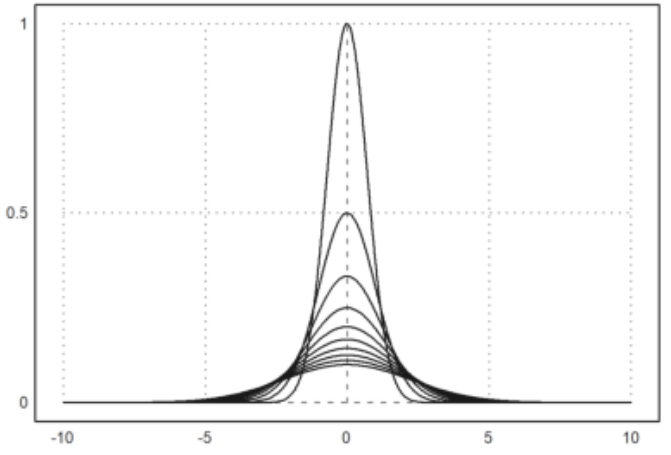
```
>function f(x,a) := 1/a*exp(-x^2/a); ...
>plot2d("f",-10,10;5,thickness=2,title="a=5"):
```




Sebagai alternatif, gunakan sebuah koleksi yang berisi nama fungsi beserta semua parameter tambahannya. Daftar khusus ini disebut call collections, dan ini merupakan cara yang lebih disarankan untuk meneruskan argumen ke suatu fungsi yang dirinya sendiri dipanggil sebagai argumen untuk fungsi lain.

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

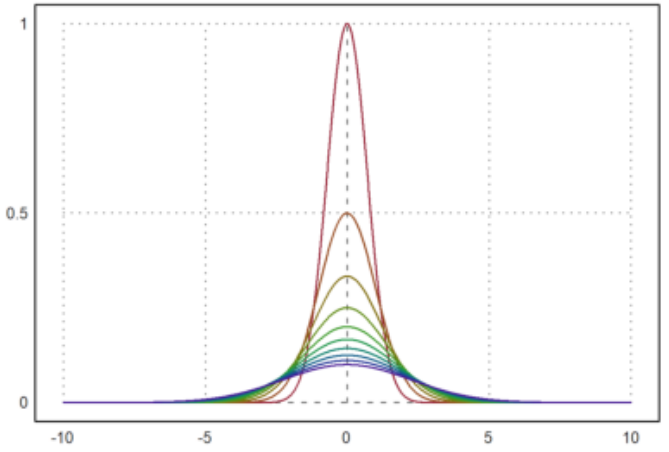
```
>plot2d({{"f",1}},-10,10); ...
>for a=2:10; plot2d({{"f",a}},>add); end:
```




Kita dapat memperoleh hasil yang sama dengan cara berikut menggunakan bahasa matriks EMT. Setiap baris dari matriks
(x,a) merupakan satu fungsi.

Selain itu, kita juga dapat mengatur warna untuk setiap baris matriks.lick on the function getspectral() for an explanation.

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




## Text Labels

Keterangan dapat berupa:

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

Perintah label akan menampilkan teks pada grafik yang sedang aktif di koordinat grafik (x,y). Perintah ini juga dapat menerima argumen posisi.

```
>plot2d("x^3-x",-1,2,title="y=x^3-x",yl="y",xl="x"):
>expr := "log(x)/x"; ...
>  plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
>  label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```

Terdapat juga fungsi labelbox(), yang dapat menampilkan fungsi-fungsi beserta sebuah teks. Fungsi ini menerima vektor string dan warna, masing-masing satu untuk setiap fungsi.

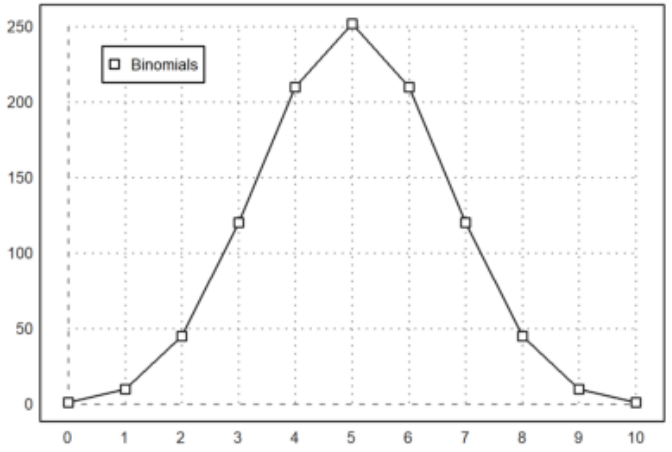
```
>function f(x) &= x^2*exp(-x^2);  ...
>plot2d(&f(x),a=-3,b=3,c=-1,d=1);  ...
>plot2d(&diff(f(x),x),>add,color=blue,style="--"); ...
>labelbox(["function","derivative"],styles=["-","--"], ...
>   colors=[black,blue],w=0.4):
```

Kotak label secara default ditempatkan pada bagian kanan atas, tetapi dengan >left kotak akan ditempatkan di bagian kiri atas. Anda dapat memindahkannya ke posisi mana pun yang diinginkan. Posisi anchor adalah sudut kanan atas kotak, dan nilainya berupa pecahan dari ukuran jendela grafik. Lebarnya diatur secara otomatis.

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

Pada contoh berikut, hanya ada satu fungsi. Jadi kita dapat menggunakan string tunggal, bukan vektor string. Untuk contoh ini, kita mengatur warna teks menjadi hitam.

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



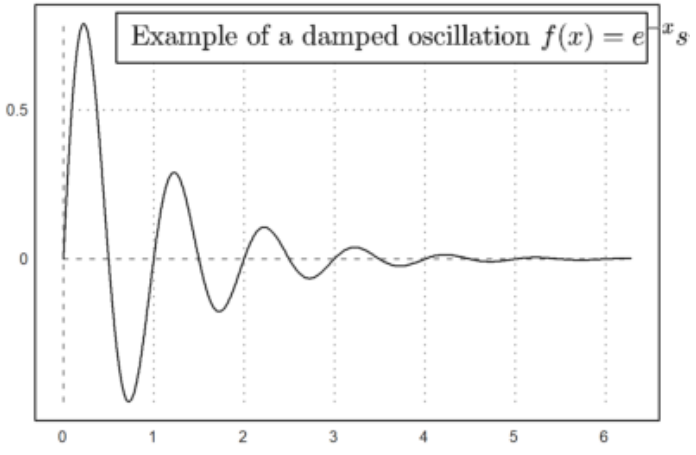

Gaya grafik ini juga tersedia di statplot(). Seperti pada plot2d(), warna dapat diatur untuk setiap baris grafik. Terdapat lebih banyak grafik khusus untuk keperluan statistik (lihat tutorial tentang statistika).

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

Fitur serupa adalah fungsi textbox().

Lebar kotak secara default mengikuti lebar maksimum dari baris teks, tetapi 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):
```




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=u"x &rarr; x&sup3; - x"):
```

Label pada sumbu-x dan sumbu-y dapat ditempatkan secara vertikal, sama seperti sumbunya.

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

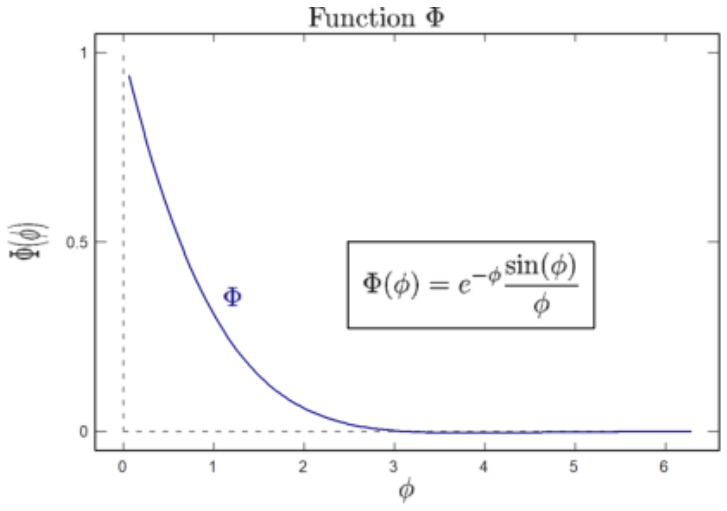
## LaTeX

Anda juga dapat menggambar formula LaTeX jika sistem LaTeX telah terpasang. Saya menyarankan MiKTeX. Jalur menuju file biner "latex" dan "dvipng" harus ada di path sistem, atau Anda harus mengatur LaTeX melalui menu opsi.

Perlu diperhatikan, parsing LaTeX berjalan lambat. Jika ingin menggunakan LaTeX pada grafik animasi, sebaiknya panggil latex() sekali sebelum loop dan gunakan hasilnya (sebuah gambar dalam matriks RGB).

Pada grafik berikut, kita menggunakan LaTeX untuk label sumbu-x dan sumbu-y, sebuah label, kotak label, dan judul grafik.

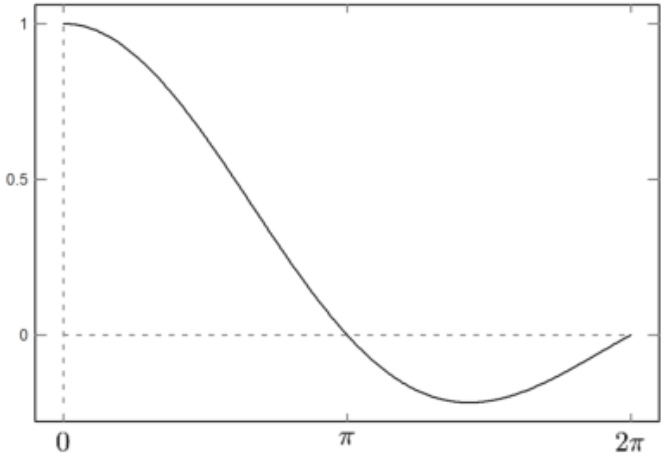
```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
>  title=latex("\text{Function $\Phi$}"), ...
>  xl=latex("\phi"),yl=latex("\Phi(\phi)")); ...
>textbox( ...
>  latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}"),x=0.8,y=0.5); ...
>label(latex("\Phi",color=blue),1,0.4):
```




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

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

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




Of course, functions can also be used.

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

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

Parameter "map" berguna untuk menggunakan fungsi pada vektor. Untuk grafik, hal ini sebenarnya tidak diperlukan. Namun, untuk menunjukkan bahwa vektorisasi berguna, kita menambahkan beberapa titik kunci pada grafik di x=-1, x=0 and x=1.

Pada grafik berikut, kita juga memasukkan beberapa kode LaTeX. Kode ini digunakan untuk dua label dan sebuah kotak teks. Tentu saja, Anda hanya dapat menggunakan LaTeX jika sistem LaTeX telah terpasang dengan benar.

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

## User Interaction

Saat menggambar sebuah fungsi atau ekspresi, parameter >user memungkinkan pengguna untuk zoom dan menggeser grafik menggunakan tombol kursor atau mouse. Pengguna dapat:

- zoom with + or -
- move the plot with the cursor keys
- select a plot window with the mouse
- reset the view with space
- exit with return

Tombol spasi akan mengembalikan grafik ke jendela plot awal.

Saat menggambar data, tanda >user hanya akan menunggu input dari keyboard.

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

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

Berikut ini menunjukkan cara lanjutan untuk interaksi pengguna (lihat tutorial tentang pemrograman untuk detail lebih lanjut).
Fungsi bawaan mousedrag() menunggu event dari mouse atau keyboard. Fungsi ini melaporkan klik mouse, pergerakan mouse, pelepasan mouse, dan penekanan tombol. Fungsi dragpoints() memanfaatkan ini, memungkinkan pengguna untuk menggeser titik mana pun pada grafik.
Kita memerlukan fungsi plot terlebih dahulu. Sebagai contoh, kita melakukan interpolasi pada 5 titik menggunakan sebuah polinomial. Fungsi ini harus menggambar pada area plot tetap.

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

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

Perhatikan parameter titik koma pada plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp(). Tanpa ini, kita harus menulis fungsi plotinterp() terlebih dahulu yang mengakses nilai secara global. Sekarang kita menghasilkan beberapa nilai acak, dan membiarkan pengguna menggeser titik-titik tersebut.

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

Terdapat juga sebuah fungsi yang menggambar fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter-parameter tersebut.

Pertama, kita membutuhkan fungsi plot.

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

Selanjutnya, kita membutuhkan nama-nama parameter, nilai awal, dan sebuah matriks nx2 untuk rentang nilai, serta opsional sebuah baris judul.

Terdapat slider interaktif yang memungkinkan pengguna mengatur nilai-nilai tersebut. Fungsi dragvalues() menyediakan fitur ini.

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

Dimungkinkan untuk membatasi nilai yang digeser menjadi bilangan bulat. Sebagai contoh, kita menulis sebuah fungsi plot yang menggambar 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 membiarkan derajat n bervariasi dari 0 hingga 20 dalam 20 langkah. Hasil dari dragvalues() digunakan untuk menggambar sketsa dengan nilai n tersebut, dan menyisipkan grafik ke dalam notebook.

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

Berikut adalah demonstrasi sederhana dari fungsi tersebut. Pengguna dapat menggambar di atas jendela grafik, 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!
```

## Styles of 2D Plots

Secara default, EMT menghitung tanda sumbu (axis ticks) secara otomatis dan menambahkan label pada setiap tanda. Hal ini dapat diubah dengan parameter grid. Gaya default dari sumbu dan label dapat dimodifikasi. Selain itu, label dan judul juga dapat ditambahkan secara manual. Untuk mengembalikan ke gaya default, gunakan reset().

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

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

Jika Anda ingin menggunakan tanda sumbu sendiri, gunakan style=0, lalu tambahkan semuanya kemudian.

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

FUntuk judul grafik 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
```

The axis can be drawn separately with xaxis() and yaxis().

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

yeks pada grafik dapat diatur dengan label(). Pada contoh berikut, ”lc” berarti lower center, yang mengatur posisi label relatif terhadap koordinat grafik.

```
>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
```

There are also text boxes.

```
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // derivative
>labelbox(["f","f'"],["-","--"],[black,red]): // label box
>plot2d(["exp(x)","1+x"],color=[black,blue],style=["-","-."]):
>gridstyle("->",color=gray,textcolor=gray,framecolor=gray);  ...
> plot2d("x^3-x",grid=1);   ...
> settitle("y=x^3-x",color=black); ...
> label("x",2,0,pos="bc",color=gray);  ...
> label("y",0,6,pos="cl",color=gray); ...
> reset():
```

Untuk kontrol lebih lanjut, sumbu-x dan sumbu-y dapat diatur secara manual.

Perintah fullwindow() memperluas jendela grafik karena kita tidak lagi memerlukan ruang untuk label di luar jendela grafik. Gunakan shrinkwindow() atau reset() untuk mengembalikan 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:
```

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

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

# Plotting 2D Data

Jika x dan y adalah vektor data, data ini akan digunakan sebagai koordinat x dan y dari sebuah kurva. Dalam kasus ini, a, b, c, dan d, atau radius r dapat ditentukan, atau jendela grafik akan menyesuaikan secara otomatis dengan data. Sebagai alternatif, >square dapat digunakan untuk mempertahankan rasio aspek persegi.
Menggambar sebuah ekspresi hanyalah singkatan dari grafik data. Untuk grafik data, Anda membutuhkan satu atau lebih baris nilai x, dan satu atau lebih baris nilai y. Dari rentang dan nilai x, fungsi plot2d akan menghitung data yang akan digambar, secara default dengan evaluasi adaptif. Untuk grafik titik gunakan ">points", untuk garis campur titik gunakan ">addpoints".

Namun, Anda juga dapat memasukkan data secara langsung:

-Gunakan vektor baris untuk x dan y untuk satu fungsi.

unakan matriks untuk x dan y agar digambar baris demi baris.

Berikut adalah contoh dengan satu baris untuk x dan y.

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

Data juga dapat digambar sebagai titik. Gunakan points=true untuk ini. Grafik bekerja seperti poligon, tetapi hanya menggambar titik sudut saja.

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

Untuk menggambar sekumpulan titik, gunakan >points. Jika color berupa vektor warna, setiap titik akan mendapatkan warna berbeda. Untuk matriks koordinat dan vektor kolom, warna diterapkan pada baris-baris matriks.
Parameter >addpoints menambahkan titik pada segmen garis untuk grafik data.

```
>xdata=[1,1.5,2.5,3,4]; ydata=[3,3.1,2.8,2.9,2.7]; // data
>plot2d(xdata,ydata,a=0.5,b=4.5,c=2.5,d=3.5,style="."); // lines
>plot2d(xdata,ydata,>points,>add,style="o"): // add points
>p=polyfit(xdata,ydata,1); // get regression line
>plot2d("polyval(p,x)",>add,color=red): // add plot of line
```

# Menggambar Daerah Yang Dibatasi Kurva

Grafik data sebenarnya adalah poligon. Kita juga dapat menggambar kurva atau kurva terisi.

- filled=true fills the plot.
- style="...": Select from "", "/", "\", "\/".
- fillcolor: See above for available colors.

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

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

Pada contoh berikut, kita menggambar sebuah elips terisi dan dua heksagon terisi menggunakan kurva tertutup dengan 6 titik, masing-masing dengan gaya isian berbeda.

```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
>t=linspace(0,2pi,6); ...
>plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1.2):
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```

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

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

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

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

Inti utama dari bahasa matriks adalah memungkinkan pembuatan tabel fungsi dengan mudah.

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

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

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

Sebuah vektor interval dapat digambar terhadap nilai-nilai x sebagai daerah terisi antara nilai bawah dan atas dari interval tersebut.
Ini berguna untuk menggambar kesalahan perhitungan, tetapi juga dapat digunakan untuk menampilkan kesalahan statistik.

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

Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka plot2d akan menggambar rentang terisi dari interval tersebut di bidang grafik. Gaya isian sama dengan gaya poligon.

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

Dimungkinkan untuk mengisi daerah nilai untuk sebuah fungsi tertentu. Untuk ini, level harus berupa matriks 2×n, di mana 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
```

We can also fill ranges of values like

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

```
>plot2d("(x^2+y^2)^2-x^2+y^2",r=1.2,level=[-1;0],style="/"):
>plot2d("cos(x)","sin(x)^3",xmin=0,xmax=2pi,>filled,style="/"):
```

# Grafik Fungsi Parametrik

Nilai x tidak perlu diurutkan. Pasangan

?

?

x,y) hanya menggambarkan sebuah kurva. Jika x diurutkan, kurva tersebut merupakan grafik dari sebuah fungsi.
Pada contoh berikut, kita menggambar spiral:

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

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

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

Sebagai alternatif, dimungkinkan menggunakan dua ekspresi untuk kurva. Contoh berikut menggambar kurva yang sama seperti sebelumnya.

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

Pada contoh berikut, kita menggambar kurva

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

dengan

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

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

# Menggambar Grafik Bilangan Kompleks

Sebuah array bilangan kompleks juga dapat digambar. Titik-titik grid akan dihubungkan. Jika sejumlah garis grid ditentukan (atau vektor $1\times2$ dari garis grid) pada argumen cgrid, hanya garis-garis tersebut yang terlihat.
Sebuah matriks bilangan kompleks secara otomatis akan digambar sebagai grid di bidang kompleks.
Pada contoh berikut, kita menggambar citra lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
>aspect(1.25); r=linspace(0,1,50); a=linspace(0,2pi,200)'; z=r*exp(I*a);
>plot2d(exp(z),cgrid=[40,10]):
>r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=r*exp(I*a);
>plot2d(exp(z),>points,>add):
```

Sebuah vektor bilangan kompleks secara otomatis digambar sebagai kurva di bidang kompleks dengan bagian nyata dan bagian imajiner.
Pada contoh ini, kita menggambar lingkaran satuan dengan:

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

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

# Statistical Plots

Terdapat banyak fungsi yang khusus untuk grafik statistik. Salah satu grafik yang sering digunakan adalah grafik kolom.
Jumlah kumulatif dari nilai yang terdistribusi normal 0-1 menghasilkan random walk.

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

Menggunakan dua baris menunjukkan pergerakan (walk) dalam dua dimensi.

```
>X=cumsum(randnormal(2,1000)); plot2d(X[1],X[2]):
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

It can also show strings as labels.

```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
>  "Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnsplot(values,lab=months,color=red,style="-");
>title("Temperature"):
>k=0:10;
>plot2d(k,bin(10,k),>bar):
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```

```
>plot2d(normal(1000),normal(1000),>points,grid=6,style="..");
>plot2d(normal(1,1000),>distribution,style="O");
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```

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

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

Atau, Anda dapat menghitung distribusi dari data dan menggambar hasilnya dengan >bar pada plot3d, atau menggunakan grafik 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 grafik menggunakan string sederhana.

```
>statplot(1:10,cumsum(random(10)),"b"):
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```

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

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

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

```
>plot2d(normal(10000),distribution=50):
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
>columnsplot(m,k):
>plot2d(random(600)*6,histogram=6):
```

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

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

With the parameter even=true, this will use integer intervals.

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

Perlu dicatat bahwa terdapat banyak grafik statistik yang mungkin berguna. Lihatlah tutorial tentang statistik untuk informasi lebih lanjut.

```
>columnsplot(getmultiplicities(1:6,intrandom(1,6000,6))):
>plot2d(normal(1,1000),>distribution); ...
>  plot2d("qnormal(x)",color=red,thickness=2,>add):
```

Terdapat juga banyak grafik khusus untuk statistik. Sebuah boxplot menunjukkan kuartil dari distribusi tersebut serta banyak outlier. Menurut definisi, outlier pada boxplot adalah data yang melebihi 1,5 kali rentang 50% tengah dari plot.

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

# Implicit Functions

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

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

of any function.Untuk menggambar himpunan (x,y)=c untuk satu atau lebih konstanta, gunakan plot2d() dengan plot implisit di bidang. Parameter untuk adalah level=c, di mana c dapat berupa vektor garis level. Selain itu, skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi pada setiap titik di plot. Parameter ”n” menentukan ketelitian plot.

```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0): // Solutions of f(x,y)=0
>plot2d(expr,level=0:0.5:20,>hue,contourcolor=white,n=200): // nice
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
```

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

```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
>plot2d("x^3-y^2",>contour,>hue,>spectral):
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
>plot2d("x^2+y^3+x*y",level=1,r=4,n=100):
>plot2d("x^2+2*y^2-x*y",level=0:0.1:10,n=100,contourcolor=white,>hue):
```

It is also possible to fill the set

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

Dengan rentang level.

Dimungkinkan untuk mengisi daerah nilai untuk sebuah fungsi tertentu. Untuk ini, level harus berupa matriks 2×n, di mana baris pertama berisi batas bawah dan baris kedua berisi batas atas.

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

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

```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
>plot2d("x^2+y^3+x*y",level=-10:20,r=2,style="-",dl=0.1,n=100):
>plot2d("sin(x)*cos(y)",r=pi,>hue,>levels,n=100):
```

It is also possible to mark a region

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

This is done by adding a level with two rows.

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

It is possible to specify a specific level. E.g., we can plot the solution of an equation like

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

```
>plot2d("x^3-x*y+x^2*y^2",r=6,level=1,n=100):
>function starplot1 (v, style="/", color=green, lab=none) ...
```

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

Di sini tidak ada grid atau tanda sumbu. Selain itu, kita menggunakan seluruh jendela untuk plot. Kita memanggil reset sebelum menguji plot ini untuk mengembalikan pengaturan grafik ke default. Hal ini tidak diperlukan jika Anda yakin plot Anda sudah bekerja dengan benar.

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

Terkadang, Anda mungkin ingin menggambar sesuatu yang plot2d hampir bisa lakukan, tapi tidak sepenuhnya.
Pada fungsi berikut, kita membuat plot impuls logaritmik. plot2d bisa membuat grafik 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
```

Let us test it with exponentially distributed values.

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

Let us animate a 2D curve using direct plots. The plot(x,y) command simply plots a curve into the plot window. setplot(a,b,c,d) sets this window.

The wait(0) function forces the plot to appear on the graphics windows. Otherwise, the redraw takes place in sparse time intervals.

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

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

Press any key to stop this animation.

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

# Logarithmic Plots

EMT menggunakan parameter ”logplot” untuk skala logaritmik.
lot logaritmik dapat dibuat dengan:

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

```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
>plot2d("exp(x+sin(x))",0,100,logplot=1):
>plot2d("exp(x+sin(x))",10,100,logplot=2):
>plot2d("gamma(x)",1,10,logplot=1):
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```

This does also work with data plots.

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

1. Fungsi dalam Satu Parameter

plot2d() tidak hanya dipakai untuk fungsi sederhana
y=f(x), tetapi juga bisa dipakai untuk fungsi dengan parameter tambahan.

Misalnya fungsi

```
>$f(x,y)=x^3 + ax^2 - ax
```

$$f\left(x,y\right)=x^3+ax^2-ax$$

memerlukan input x dan juga parameter a.

Di EMT, parameter tambahan bisa diberikan langsung di pemanggilan fungsi.

```
>clg;
>function f(x,a):= x^3 + a*x^2 - a*x
>plot2d("f",-5,5;3);
>
```

- function f(x,a) mendefinisikan fungsi dengan parameter tambahan.
- plot2d(”f’, -5, 5; 3) menggambar grafik fungsi dari -5 sampai 5 dengan nilai a=3.
- Tanda ; 3 berarti parameter tambahan yang dipakai adalah a=3.

Nilai a bisa diganti untuk melihat bagaimana bentuk kurva berubah.
Misalnya:

```
>clg;
>hold on;
>plot2d("f", -5, 5; 2, color=red);
>plot2d("f", -5, 5; 4, color=blue);
```

1. Menggambar Grafik Fungsi Satu Variabel dalam Bentuk Ekspresi Langsung

.

Fungsi digambar dengan langsung menuliskan ekspresinya dalam plot2d().

```
>plot2d("sin(x)", -2pi, 2pi);
```

2. Menggambar Grafik Fungsi Satu Variabel yang Rumusnya Disimpan dalam Variabel Ekspresi

- Ekspresi disimpan dalam variabel, lalu dipanggil.

```
>expr:="x^2 - 2*x + 1";
>plot2d(expr, -5, 5);
```

3. Menggambar Grafik Fungsi Satu Variabel yang Fungsinya Didefinisikan sebagai Fungsi Numerik

Definisi fungsi menggunakan function ... := ....

```
>function f(x) :=x^3 - 2*x;
>plot2d("f", -5, 5);
```

4. Menggambar Beberapa Kurva Sekaligus

Beberapa fungsi bisa digambar dalam satu perintah.

```
> plot2d(["sin(x)","cos(x)"], -2pi, 2pi);
```

5. Mengatur Ukuran Gambar, Format (Style), dan Warna Kurva

Gunakan parameter style, color, thickness.

```
>plot2d("x^2", -5, 5, color=green, style="--", thickness=2);
```

# CONTOH PENYELESAIAN MASALAH PLOT 2D

ˆ

1. Gambarkan grafik fungsi kuadrat f(x)=xˆ2-4x+3 pada interval [-1,5].

```
>clg;
>plot2d("xˆ2-4*x+3", -1, 5, color=blue, thickness=2);
>xlabel("x"); ylabel("f(x)");
```

Grafik berbentuk parabola terbuka ke atas, memotong sumbu-x di
x=1 dan x=3, serta memiliki titik minimum di (2,-1).

2. Gambarkan grafik fungsi f(x)=eˆx beserta garis singgung di titik x=0.

```
>clg; hold on;
>plot2d("exp(x)", -2, 2, color=blue, thickness=2);
>plot2d("1+x", -2, 2, color=red, style="--");
>xlabel("x"); ylabel("y");
```

Kurva eˆ selalu positif dan meningkat cepat. Garis singgung di(0,1) adalah =1+x, ditampilkan dengan garis merah putus-putus.

3. gambarkan grafik fungsi y=sin(x) dan y=cos(x) pada interval [-2\pi,2\pi].

```
>clg; hold on;
>plot2d("sin(x)", -2pi, 2pi, color=blue);
>plot2d("cos(x)", -2pi, 2pi, color=red);
>xlabel("x"); ylabel("y");
```

Grafik sinus (biru) dan cosinus (merah) adalah gelombang periodik dengan periode 2pi. Terlihat bahwa cos(x) adalah pergeseran sin(x) sejauh pi/2

4. Gambarkan grafik fungsi \; f(x)=\frac{1}{x} \; pada interval \; [-5,5], \; x \neq 0.

```
>clg;
>plot2d("1/x", -5, -0.1, color=blue);
>plot2d("1/x", 0.1, 5, color=blue);
>xlabel("x"); ylabel("f(x)");
```

Fungsi f(x)=1/x memiliki asimtot tegak di x=0 dan asimtot datar di y=0. Grafik terbagi menjadi dua cabang, satu di kuadran I dan satu di kuadran III.

5. Gambarkan lingkaran dengan persamaan xˆ2+yˆ2=9.

```
>n = 200;
>t = linspace(0,2pi,n);
>X = 3*cos(t);
>Y = 3*sin(t);
>clg;
>plot2d(X,Y,thickness=2,color=blue);
```