

Nama : Novita Indriyati  
NIM : 23030130020  
Kelas : Pendidikan Matematika A

## Menggambar Grafik 2D dengan EMT

---

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

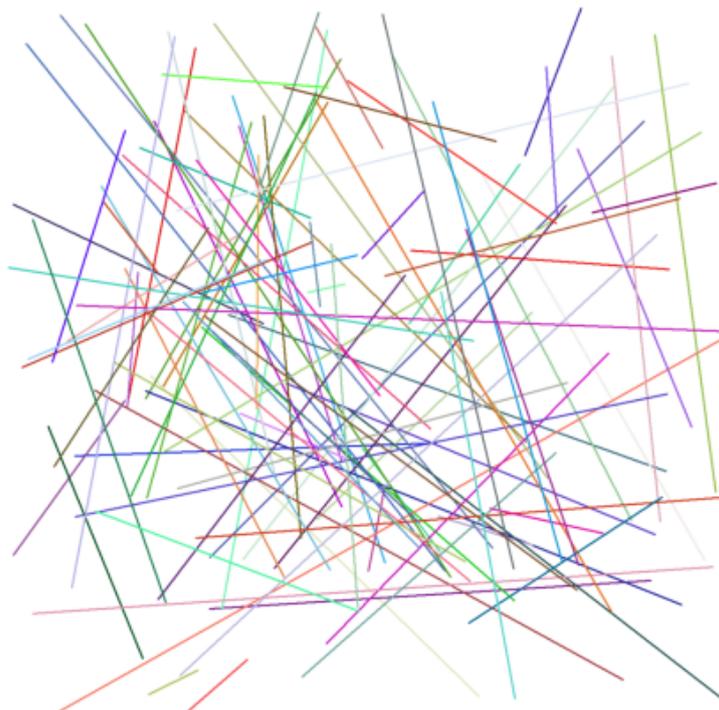
### Plot Dasar

---

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

Dalam contoh, kita hanya menggambar beberapa garis acak dalam berbagai warna. Untuk detail tentang 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;
```

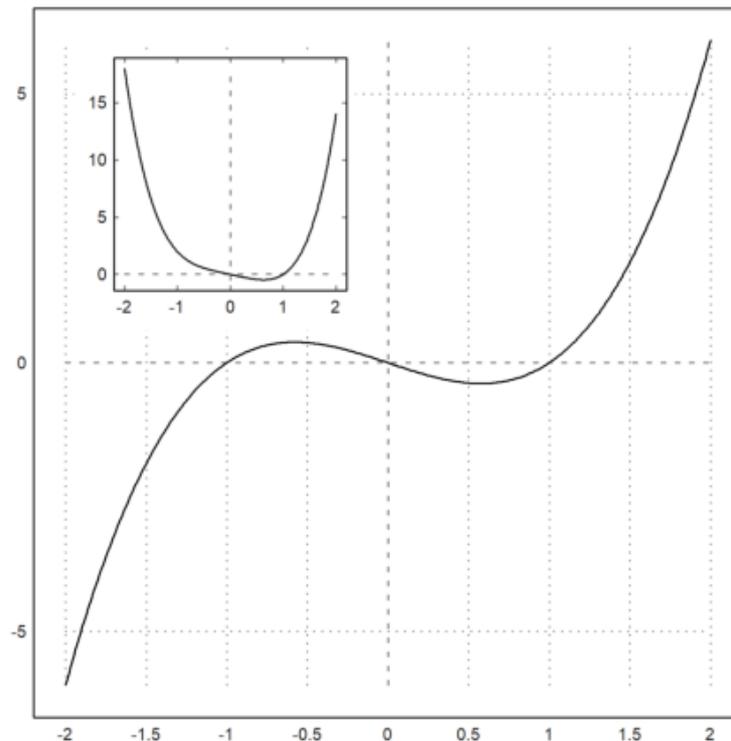
Grafik perlu ditahan, karena perintah `plot()` akan menghapus jendela plot.

Untuk menghapus semua yang kami lakukan, kami 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, kami menggambar plot sebagai sisipan di plot lain. Ini dilakukan dengan mendefinisikan jendela plot yang lebih kecil. Perhatikan bahwa jendela ini tidak menyediakan ruang untuk label sumbu di luar jendela plot. Kita harus menambahkan beberapa margin untuk ini sesuai kebutuhan. Perhatikan bahwa kami menyimpan dan memulihkan jendela penuh, dan menahan plot saat ini saat kami memplot 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);
```

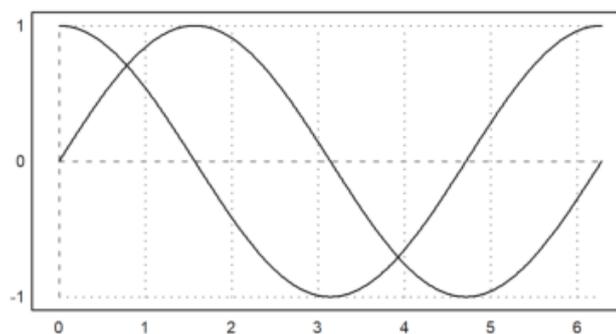
Plot dengan banyak angka dicapai dengan cara yang sama. Ada fungsi `figure()` utilitas untuk ini.

## Aspek Plot

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 jendela grafis saat ini.

Tetapi Anda juga dapat mengubahnya untuk satu plot. Untuk ini, ukuran area 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 default plot termasuk rasio aspek.

Plot 2D di Euler

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

Dimungkinkan untuk membuat plot di Maxima menggunakan Gnuplot atau dengan Python menggunakan Math Plot Lib.

Euler dapat memplot plot 2D dari

- ekspresi
- fungsi, variabel, atau kurva parameter,
- vektor nilai x-y,
- awan titik di pesawat,
- kurva implisit dengan level atau wilayah level.
- Fungsi kompleks

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

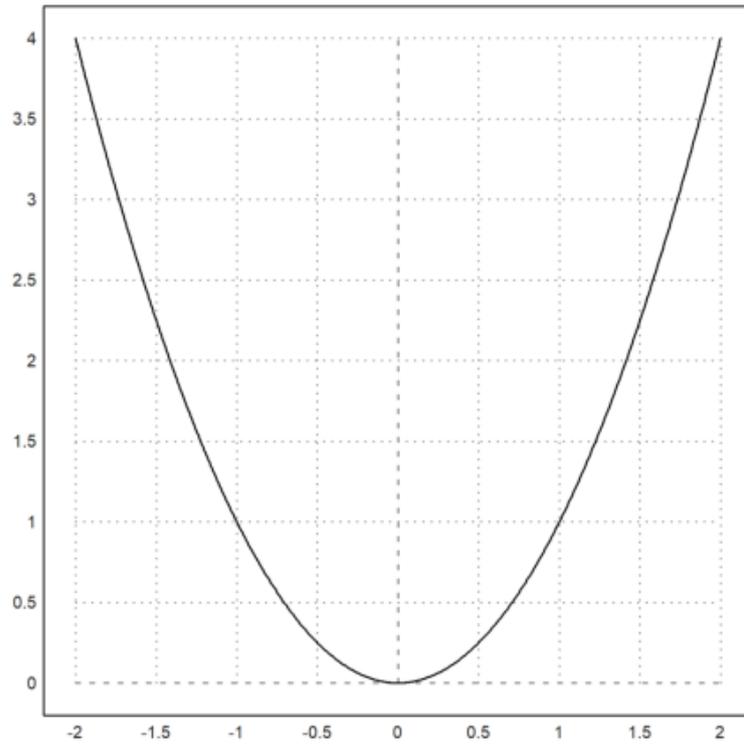
Plot Ekspresi atau Variabel

Ekspresi tunggal dalam "x" (mis. "4\*x^2") atau nama fungsi (mis. "f") menghasilkan grafik fungsi.

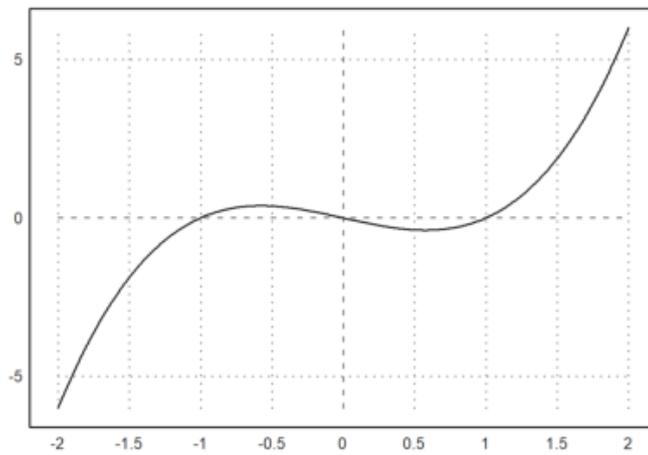
Berikut adalah contoh paling dasar, yang menggunakan rentang default dan menetapkan rentang y yang tepat agar sesuai dengan plot fungsi.

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

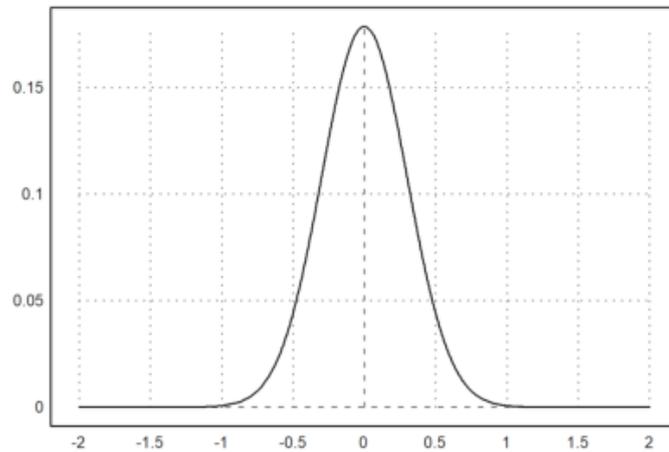
```
>plot2d("x^2"):
```



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



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

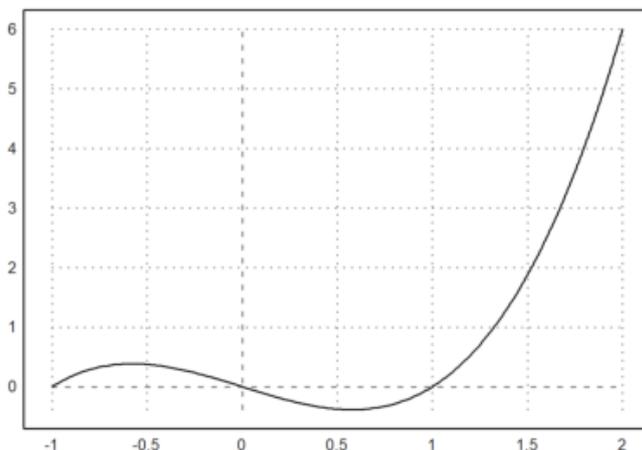


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa Gambaran gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai batas X (dan Y) di belakang ekspresi yang digambar.

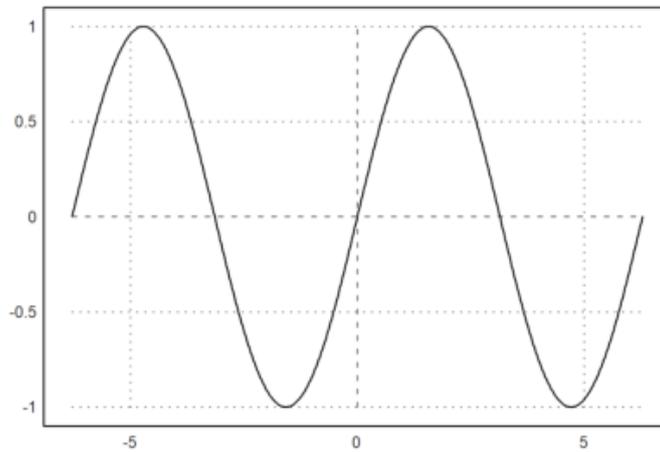
Rentang plot diatur dengan parameter yang ditetapkan berikut:

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

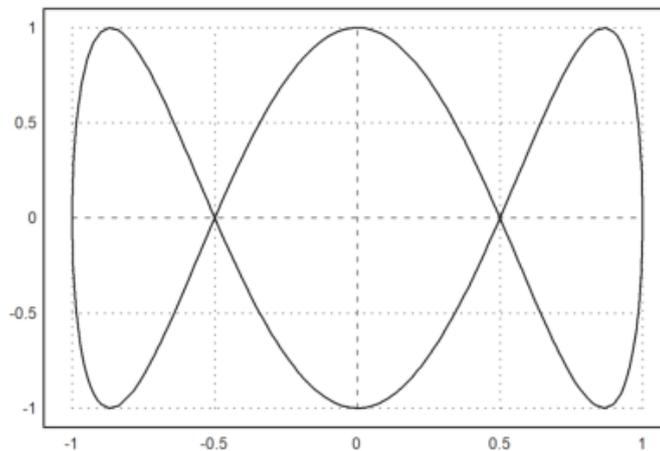
```
>plot2d("x^3-x",-1,2):
```



```
>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 untuk titik dua adalah perintah `insimg(baris)`, yang menyisipkan plot yang menempati sejumlah baris teks tertentu.

Dalam opsi, plot dapat diatur untuk muncul

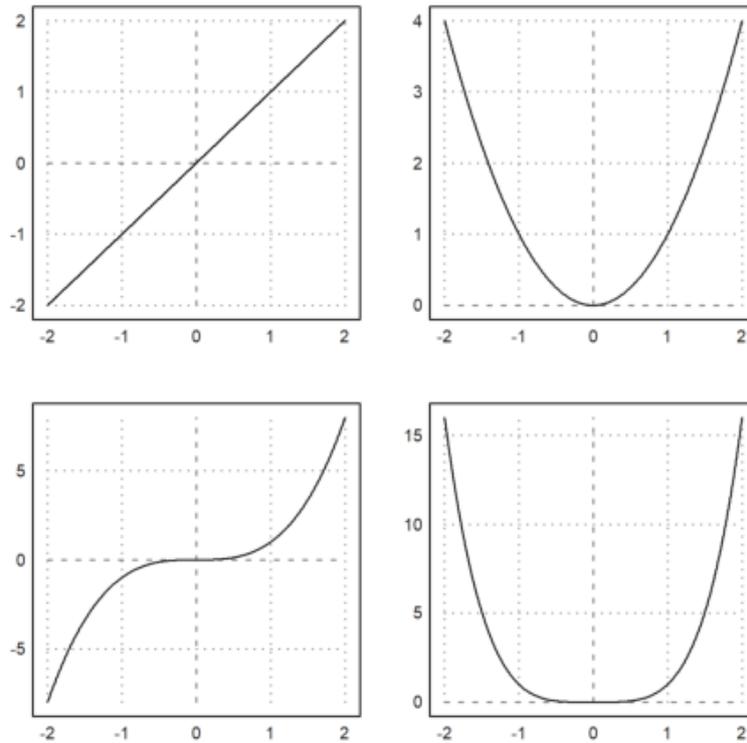
- di jendela terpisah yang dapat diubah ukurannya,
- di jendela buku catatan.

Lebih banyak gaya dapat dicapai dengan perintah plot tertentu.

Bagaimanapun, tekan tombol tabulator untuk melihat plot, jika disembunyikan.

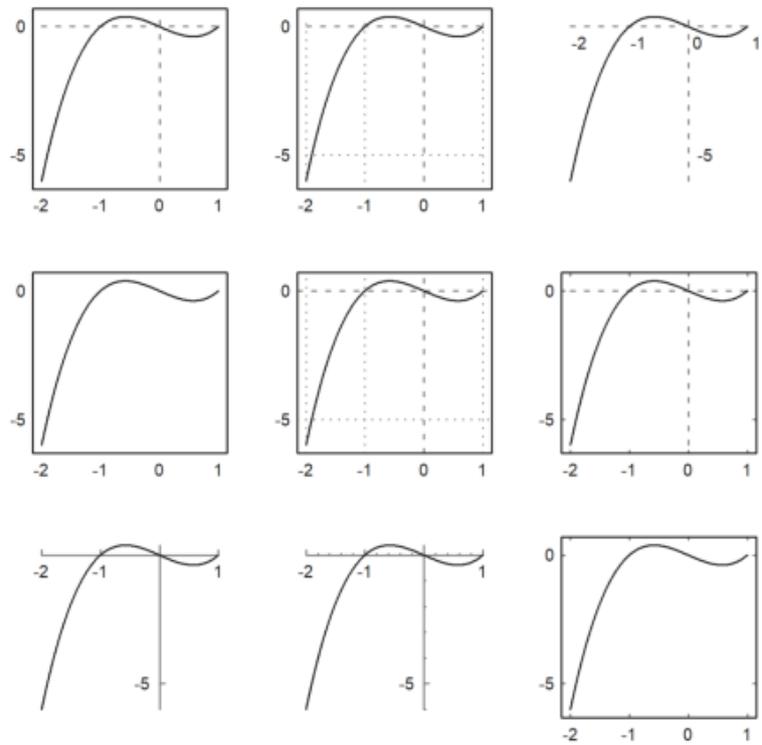
Untuk membagi jendela menjadi beberapa plot, gunakan perintah `figure()`. Dalam contoh, kami memplot  $x^1$  hingga  $x^4$  menjadi 4 bagian jendela. `figure(0)` mengatur ulang jendela default.

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



Di `plot2d()`, ada gaya alternatif yang tersedia dengan `grid=x`. Untuk gambaran umum, kami menunjukkan berbagai gaya kisi dalam satu gambar (lihat di bawah untuk perintah `figure()`). Gaya `kisi=0` tidak disertakan. Ini menunjukkan tidak ada grid dan tidak ada bingkai.

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

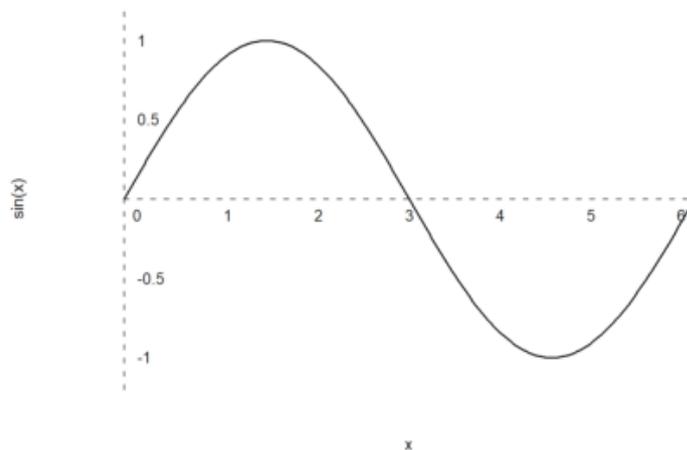


Jika argumen ke `plot2d()` adalah ekspresi yang diikuti oleh empat angka, angka-angka ini adalah rentang  $x$  dan  $y$  untuk plot.

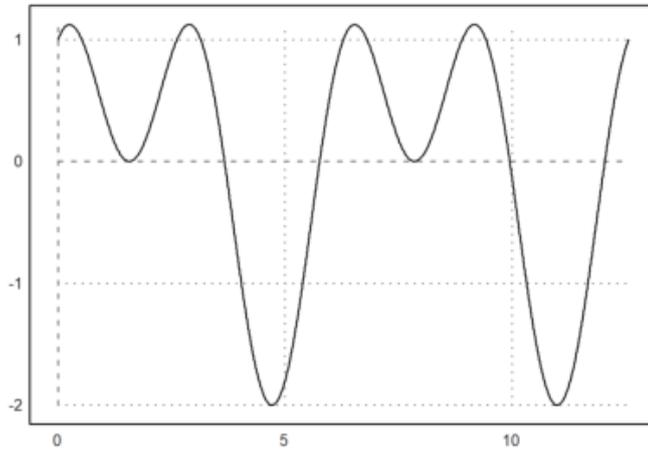
Atau,  $a$ ,  $b$ ,  $c$ ,  $d$  dapat ditentukan sebagai parameter yang ditetapkan sebagai  $a=...$  dll.

Dalam contoh berikut, kita mengubah gaya kisi, 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);
```

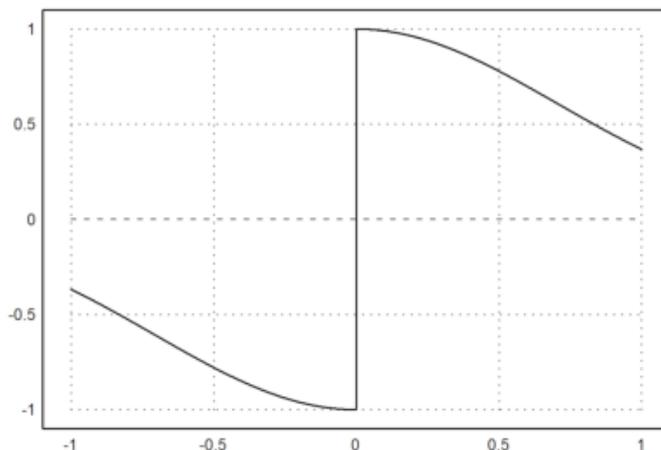


Gambar yang dihasilkan dengan memasukkan plot ke dalam jendela teks disimpan di direktori yang sama dengan buku catatan, secara default di subdirektori bernama "gambar". Mereka juga digunakan oleh ekspor HTML.

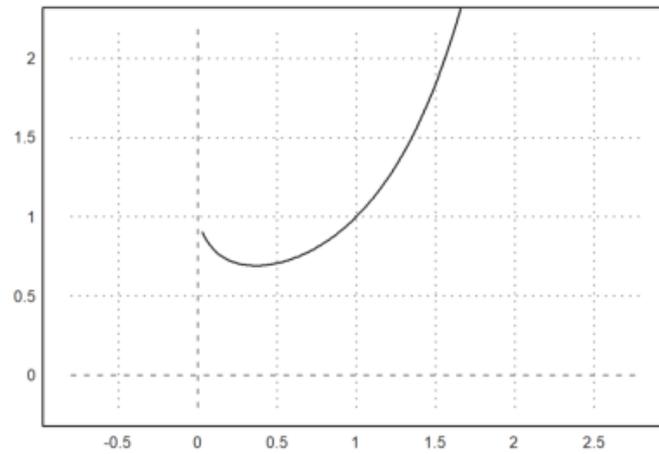
Anda cukup menandai gambar apa saja dan menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, Anda juga dapat mengekspor grafik saat ini dengan fungsi di menu File.

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

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

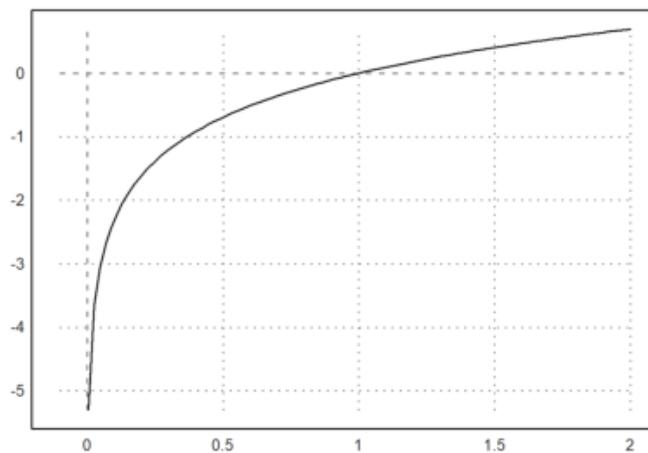


```
>plot2d("x^x",r=1.2,cx=1,cy=1):
```



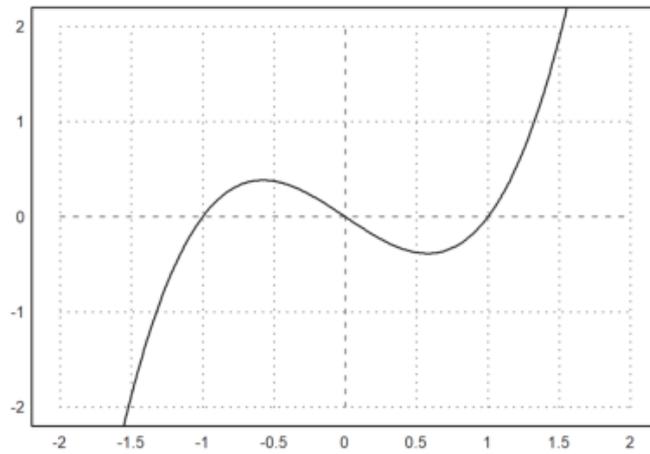
Perhatikan bahwa  $x^x$  tidak didefinisikan untuk  $x \leq 0$ . Fungsi `plot2d` menangkap kesalahan ini, dan mulai merencanakan segera setelah fungsi didefinisikan. Ini berfungsi untuk semua fungsi yang mengembalikan NAN keluar dari jangkauan definisinya.

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

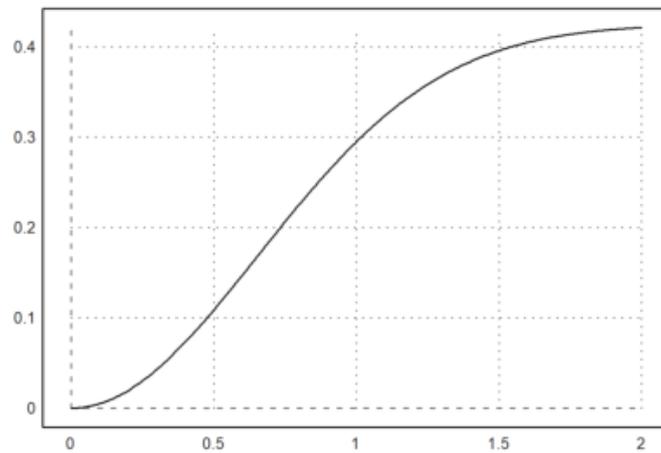


Parameter `square=true` (atau `>square`) memilih y-range secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot.

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

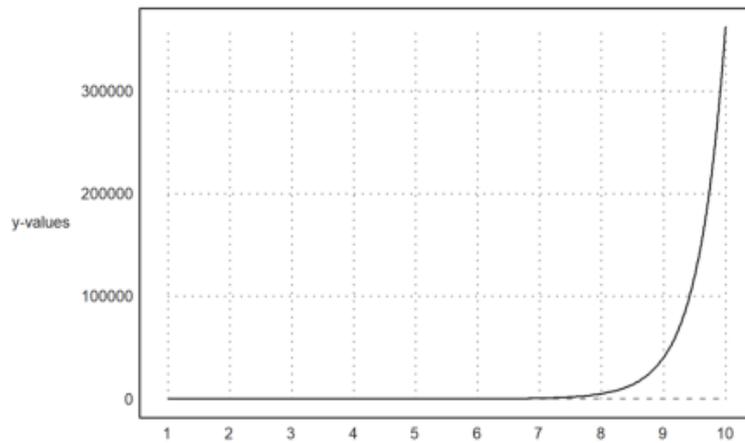


```
>plot2d('integrate("sin(x)*exp(-x^2)",0,x)',0,2): // plot integral
```



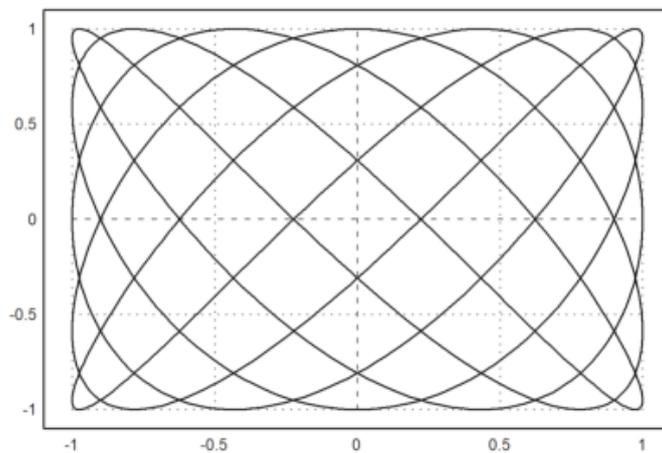
Jika Anda membutuhkan lebih banyak ruang untuk label-y, panggil `shrinkwindow()` dengan parameter yang lebih kecil, atau tetapkan nilai positif untuk "lebih kecil" di `plot2d()`.

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

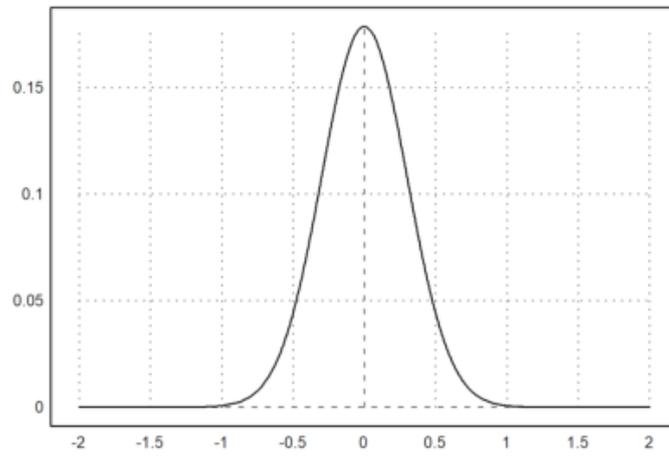


Ekspresi simbolik juga dapat digunakan, karena disimpan sebagai ekspresi string sederhana.

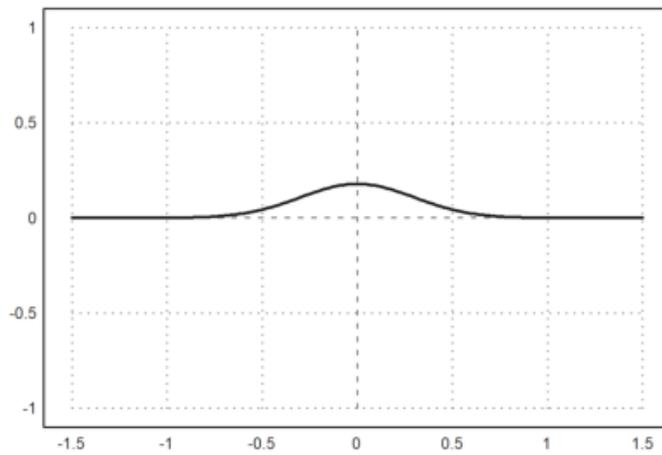
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



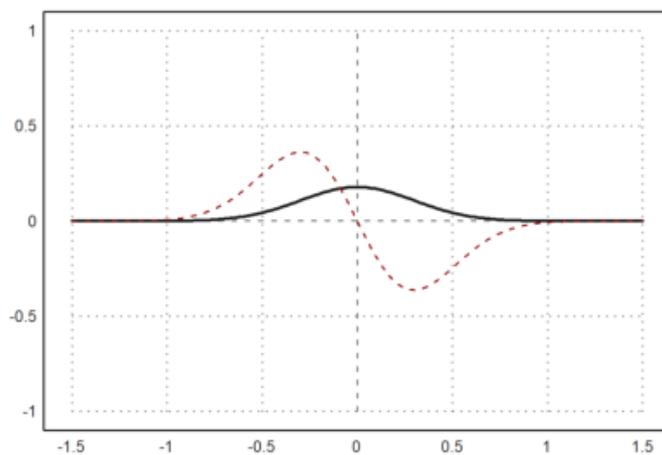
```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression
>plot2d(expr,-2,2): // plot from -2 to 2
```



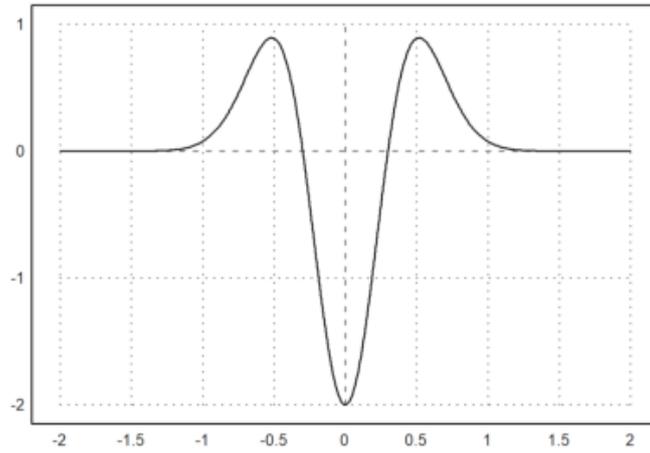
```
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)
```



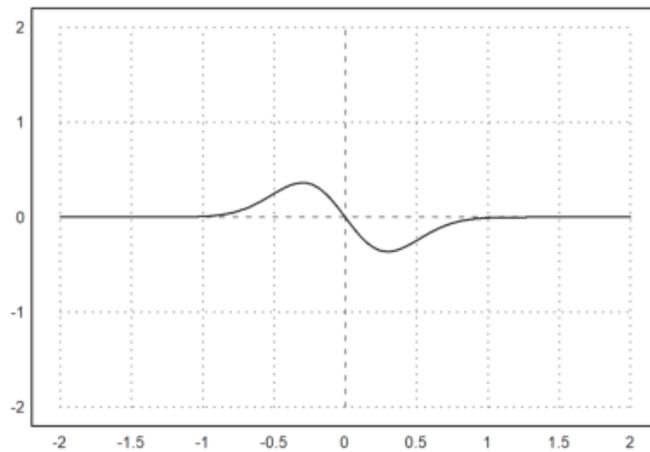
```
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot
```



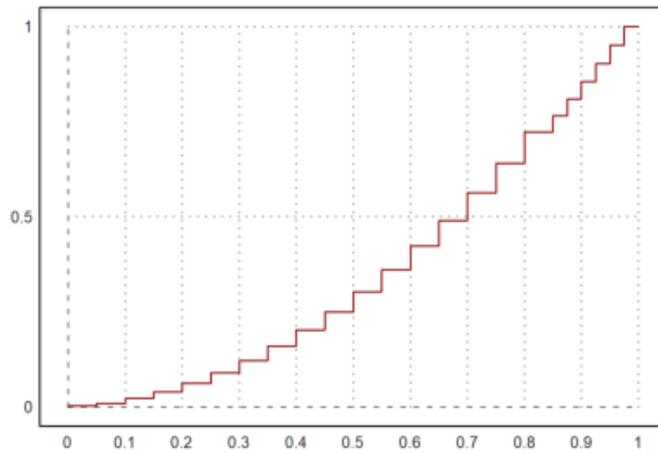
```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle
```



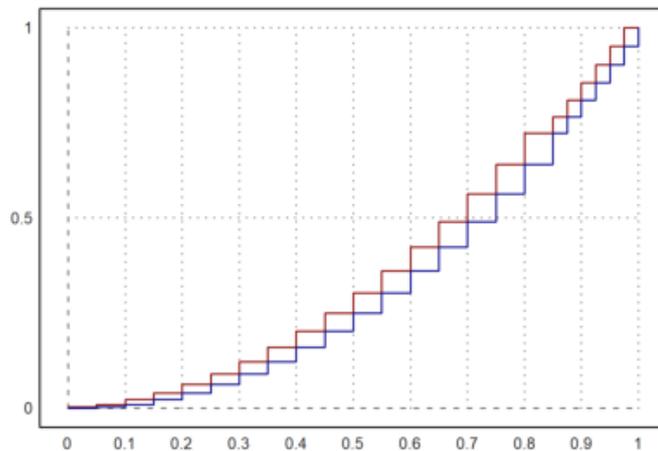
```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square
```



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



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

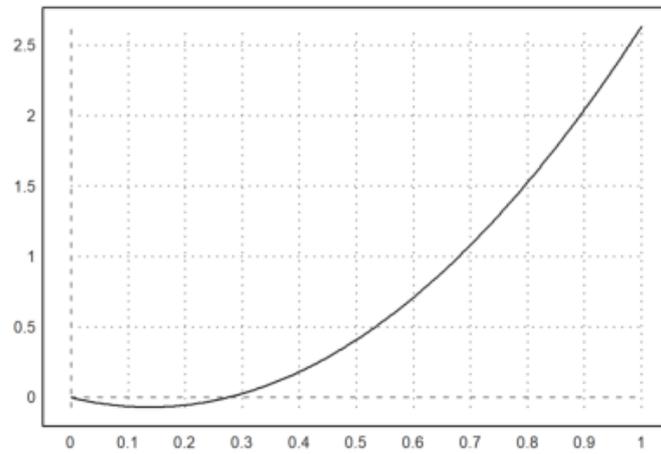


## Fungsi dalam satu Parameter

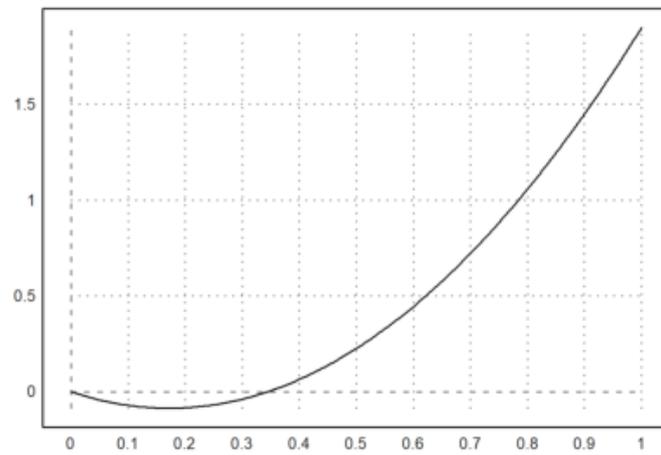
Fungsi plot yang paling penting untuk plot planar adalah `plot2d()`. Fungsi ini diimplementasikan dalam bahasa Euler dalam file "plot.e", yang dimuat di awal program.

Berikut adalah beberapa contoh menggunakan fungsi. Seperti biasa di EMT, fungsi yang berfungsi untuk fungsi atau ekspresi lain, Anda dapat meneruskan parameter tambahan (selain  $x$ ) yang bukan variabel global ke fungsi dengan parameter titik koma atau dengan koleksi panggilan.

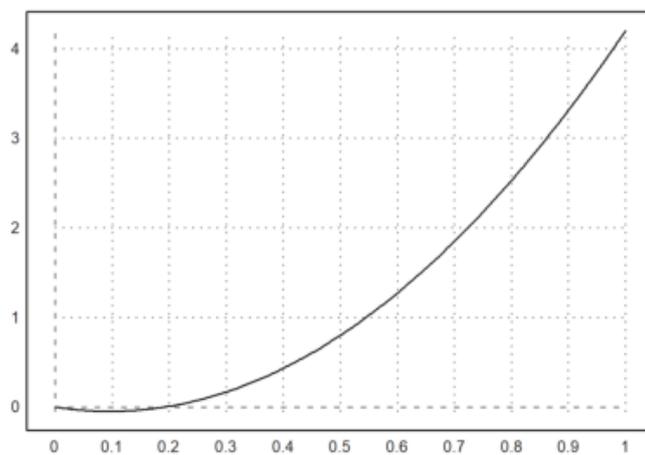
```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
>a=0.3; plot2d("f",0,1;a): // plot with a=0.3
```



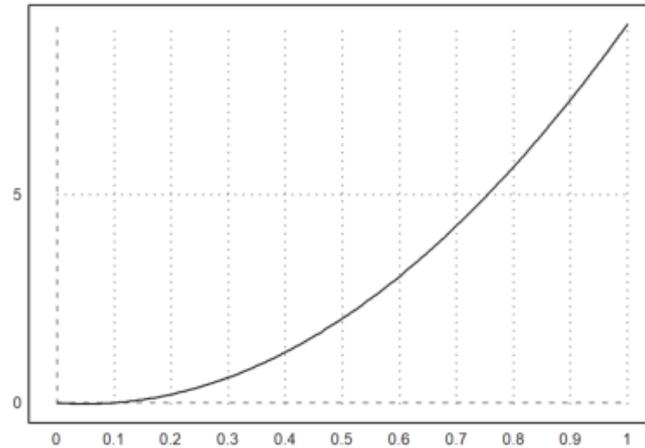
```
>plot2d("f",0,1;0.4): // plot with a=0.4
```



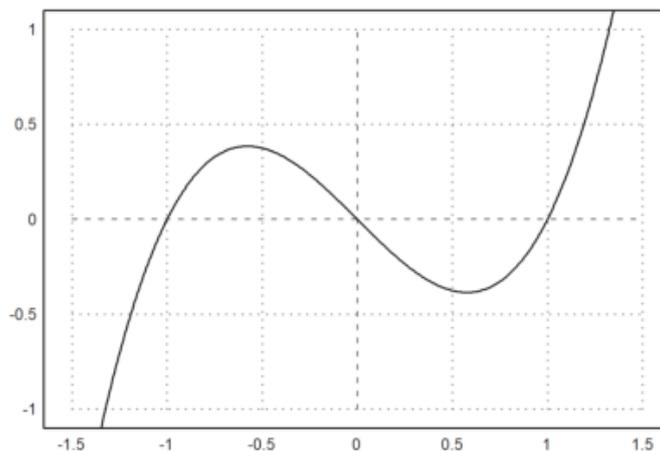
```
>plot2d({"f",0.2},0,1): // plot with a=0.2
```



```
>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 dari fungsi yang diterima

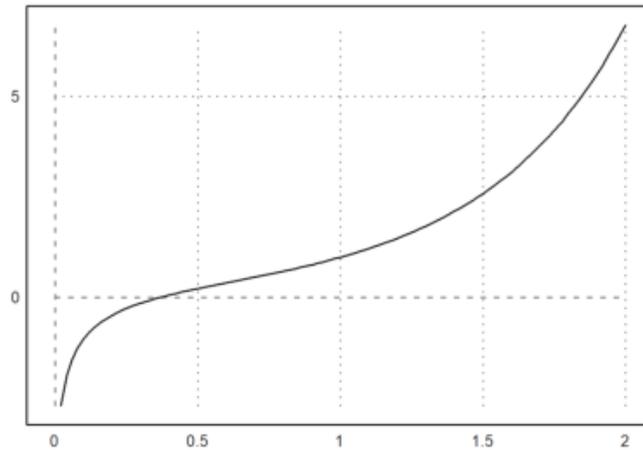
- ekspresi atau ekspresi simbolik dalam x
- fungsi atau fungsi simbolis dengan nama sebagai "f"
- fungsi simbolis hanya dengan nama f

Fungsi plot2d() juga menerima fungsi simbolis. Untuk fungsi simbolis, nama saja yang berfungsi.

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

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

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

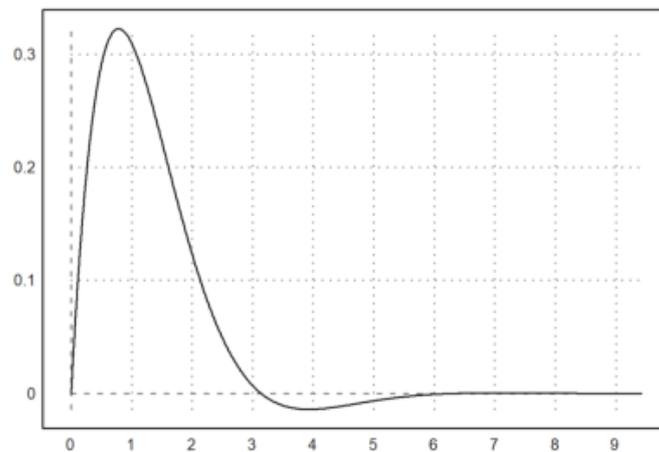


Tentu saja, untuk ekspresi atau ekspresi simbolik, nama variabel sudah cukup untuk memplotnya.

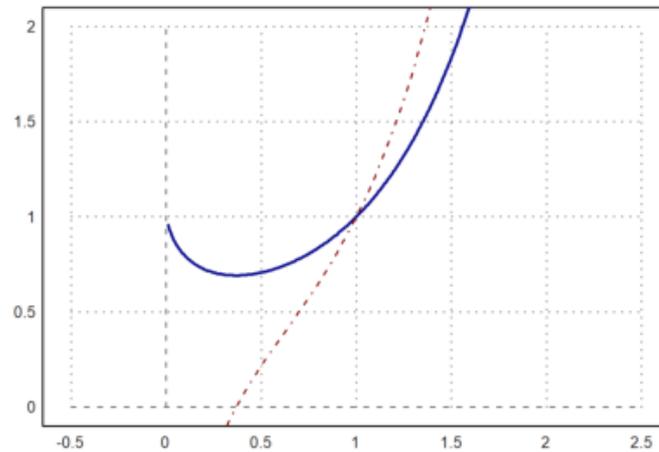
```
>expr &= sin(x)*exp(-x)
```

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

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



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



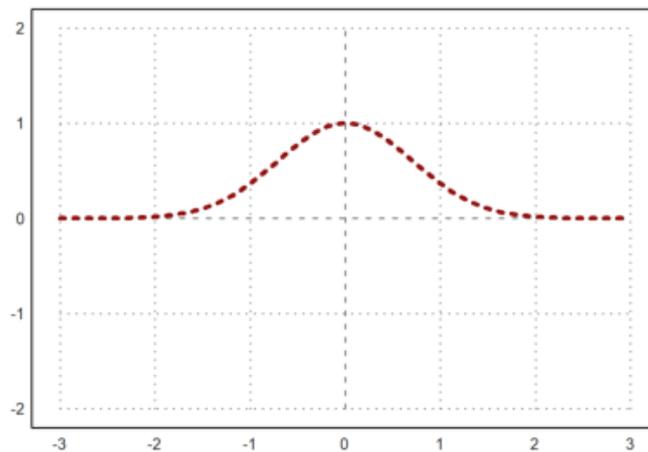
Untuk gaya garis ada berbagai pilihan.

- gaya="...". Pilih dari "-", "--", "-.", ":", ":-", "-.-".
- warna: Lihat di bawah untuk warna.
- ketebalan: Default adalah 1.

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

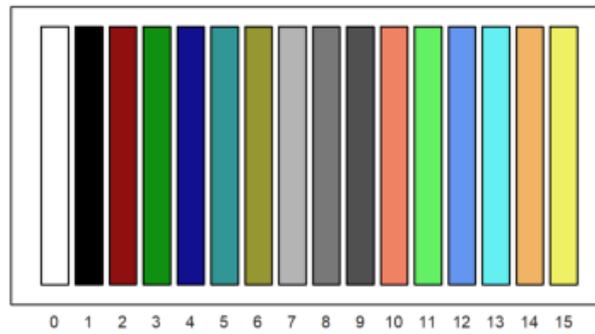
- 0.15: indeks warna default.
- konstanta warna: putih, hitam, merah, hijau, biru, cyan, zaitun, abu-abu muda, abu-abu, abu-abu tua, oranye, hijau muda, pirus, biru muda, oranye terang, kuning
- rgb(merah, hijau, biru): parameter adalah real dalam [0,1].

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



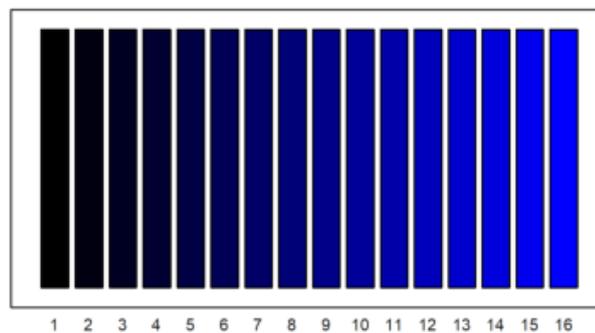
Berikut adalah tampilan warna EMT yang telah ditentukan sebelumnya.

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



But you can use any color.

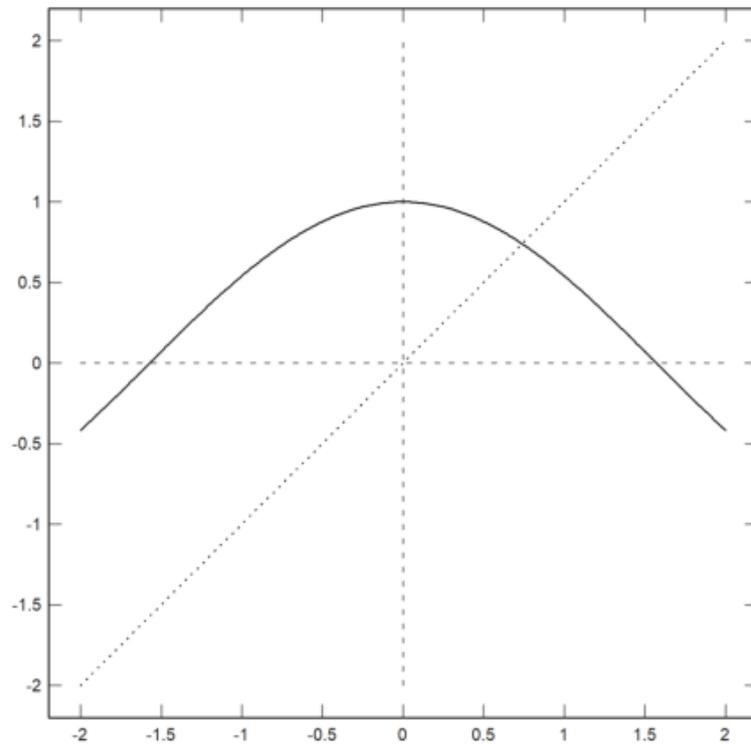
```
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```



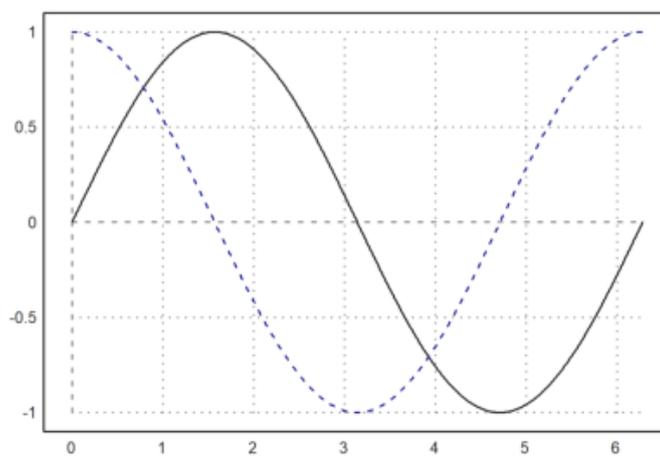
## Menggambar Beberapa Kurva pada bidang koordinat yang sama

Plot lebih dari satu fungsi (multiple function) ke dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu metode menggunakan `>add` untuk beberapa panggilan ke `plot2d` secara keseluruhan, tetapi panggilan pertama. Kami telah menggunakan fitur ini dalam contoh di atas.

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

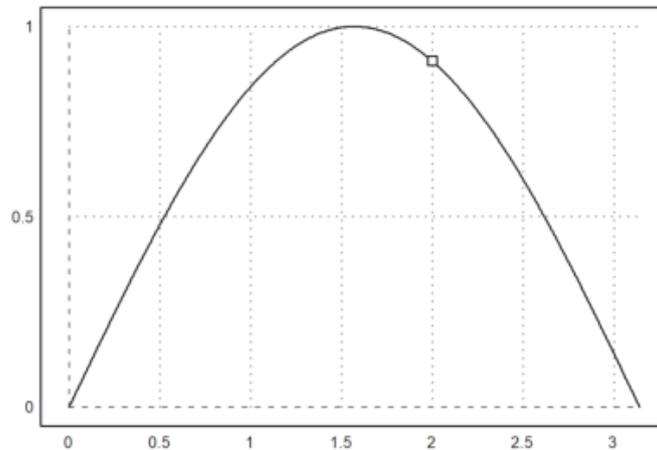


```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



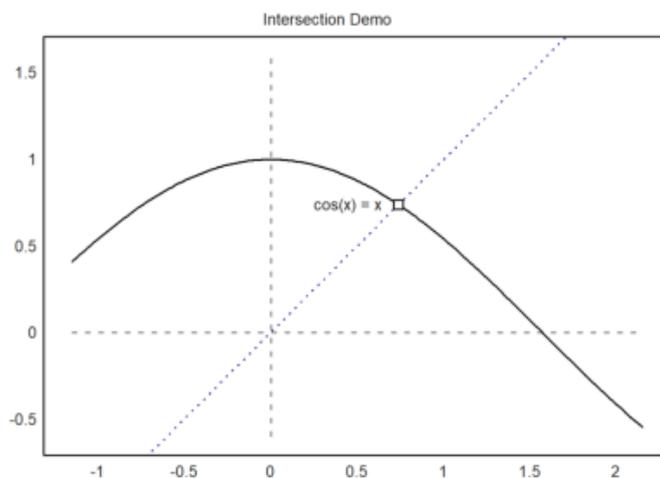
Salah satu kegunaan `>add` adalah untuk menambahkan titik pada kurva.

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



Kami menambahkan titik persimpangan dengan label (pada posisi "cl" untuk kiri tengah), dan memasukkan hasilnya ke dalam notebook. Kami juga menambahkan judul ke plot.

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



Dalam demo berikut, kami memplot fungsi sinc(x)=sin(x)/x dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung ekspansi ini menggunakan Maxima melalui ekspresi simbolis.

Plot ini dilakukan dalam perintah multi-baris berikut dengan tiga panggilan ke plot2d(). Yang kedua dan yang ketiga memiliki set flag >add, yang membuat plot menggunakan rentang sebelumnya.

Kami menambahkan kotak label yang menjelaskan fungsi.

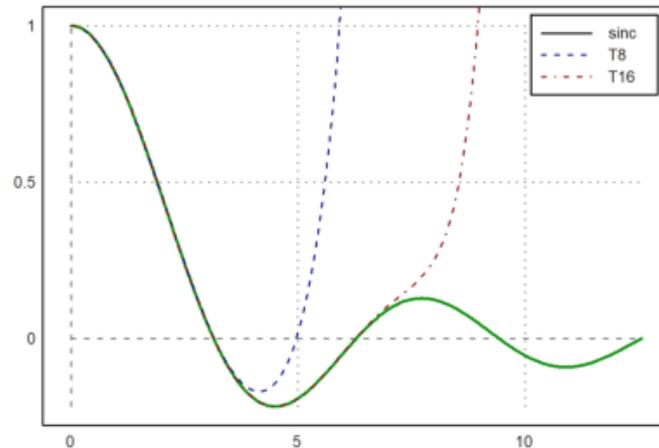
```
>$taylor(sin(x)/x,x,0,4)
```

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

```

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

```



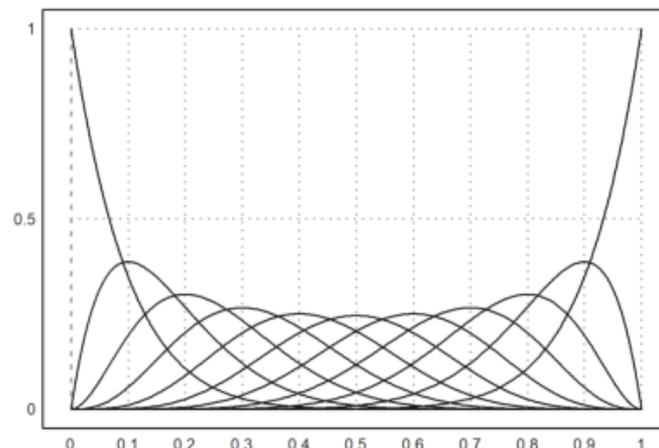
Dalam contoh berikut, kami menghasilkan Bernstein-Polinomial.

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

```

>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;

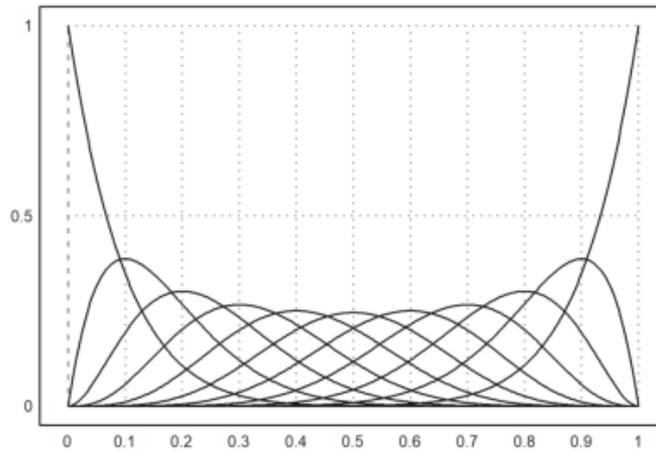
```



Metode kedua menggunakan pasangan matriks nilai-x dan matriks nilai-y yang berukuran sama.

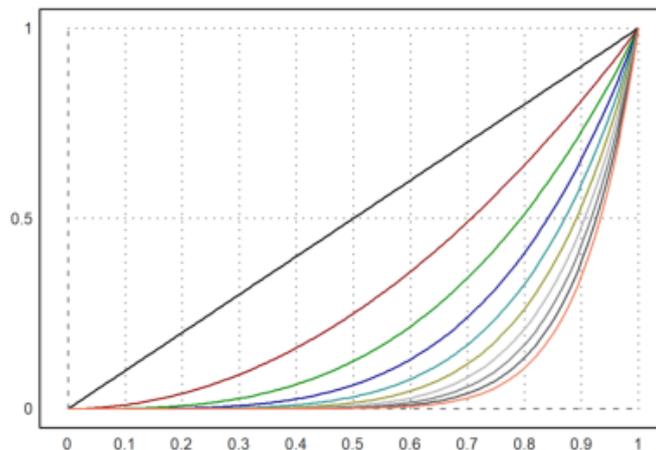
Kami menghasilkan matriks nilai dengan satu Polinomial Bernstein di setiap baris. Untuk ini, kita cukup menggunakan vektor kolom i. Lihat pengantar tentang bahasa matriks untuk mempelajari lebih detail.

```
>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):
```



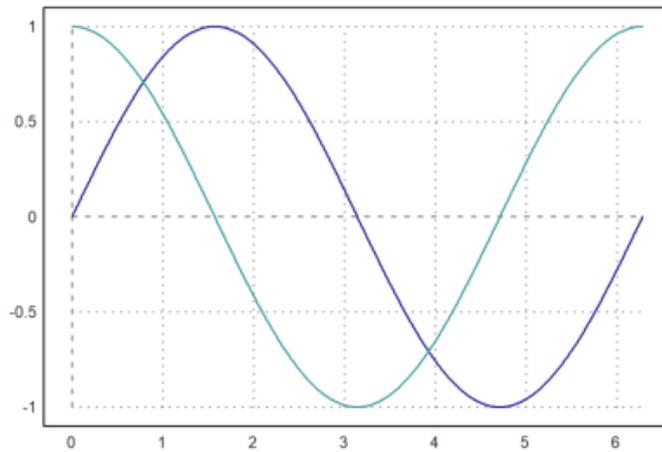
Perhatikan bahwa parameter warna dapat berupa vektor. Kemudian setiap warna digunakan untuk setiap baris matriks.

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

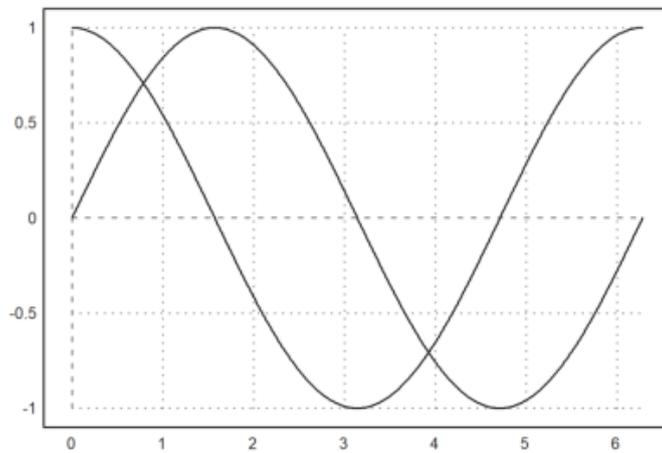


Metode lain adalah menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan larik warna, larik gaya, dan larik ketebalan dengan panjang yang sama.

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



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



Kita bisa 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
```

```

      10          9          8 2          7 3
      [(1 - x) , 10 (1 - x) x, 45 (1 - x) x , 120 (1 - x) x ,
      6 4          5 5          4 6          3 7
210 (1 - x) x , 252 (1 - x) x , 210 (1 - x) x , 120 (1 - x) x ,
      2 8          9 10
45 (1 - x) x , 10 (1 - x) x , x ]
```

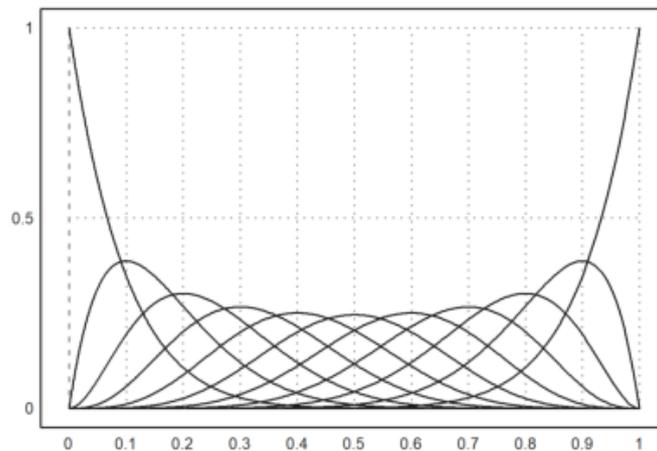
```
>mxm2str(v) // get a vector of strings from the symbolic vector
```

```

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

```

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

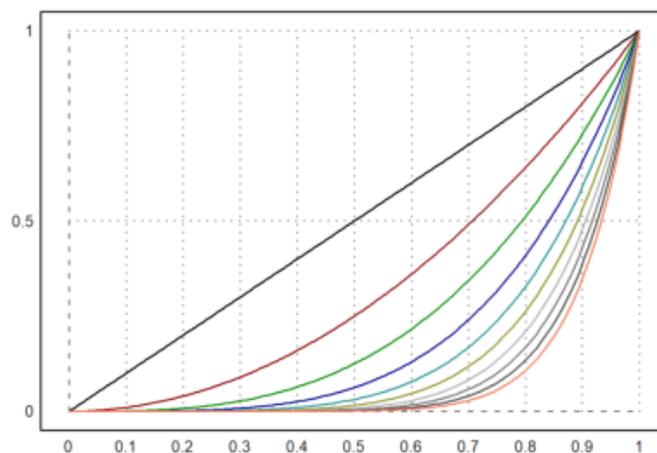


Alternatif lain adalah dengan menggunakan bahasa matriks Euler.

Jika ekspresi menghasilkan matriks fungsi, dengan satu fungsi di setiap baris, semua fungsi ini akan diplot ke dalam satu plot.

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

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

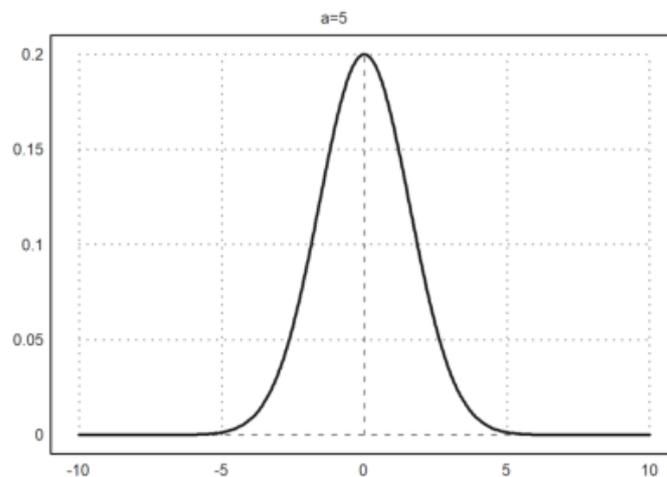


Ekspresi dan fungsi satu baris dapat melihat variabel global.

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

Berhati-hatilah, untuk meletakkan semua parameter yang ditetapkan di akhir perintah plot2d. Dalam contoh kita meneruskan a=5 ke fungsi f, yang kita plot dari -10 hingga 10.

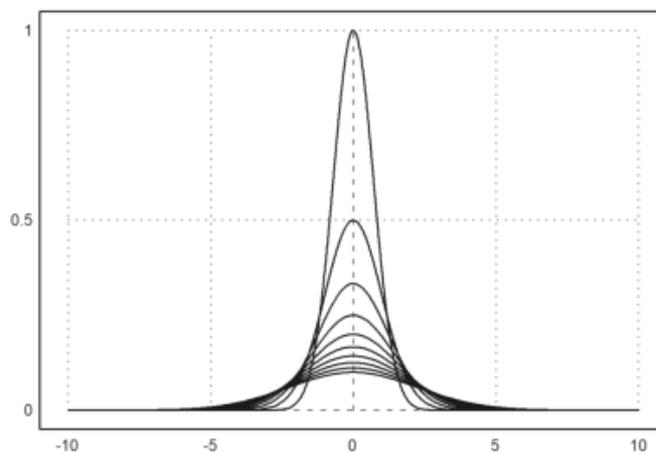
```
>function f(x,a) := 1/a*exp(-x^2/a); ...  
>plot2d("f",-10,10;5,thickness=2,title="a=5"):
```



Atau, gunakan koleksi dengan nama fungsi dan semua parameter tambahan. Daftar khusus ini disebut koleksi panggilan, dan itu adalah cara yang lebih disukai untuk meneruskan argumen ke fungsi yang dengan sendirinya diteruskan sebagai argumen ke fungsi lain.

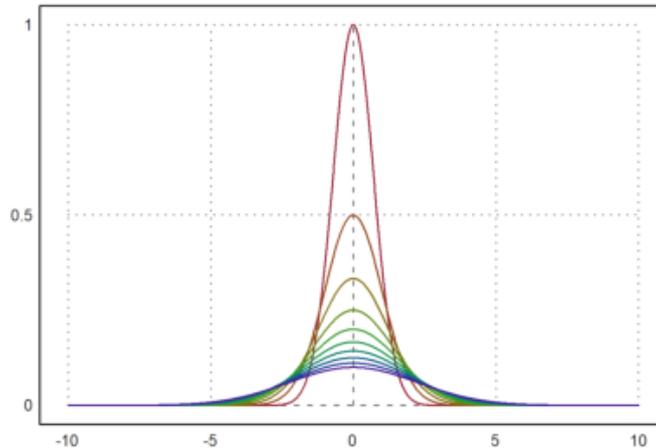
Dalam contoh berikut, kami menggunakan loop untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman untuk loop).

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



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

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



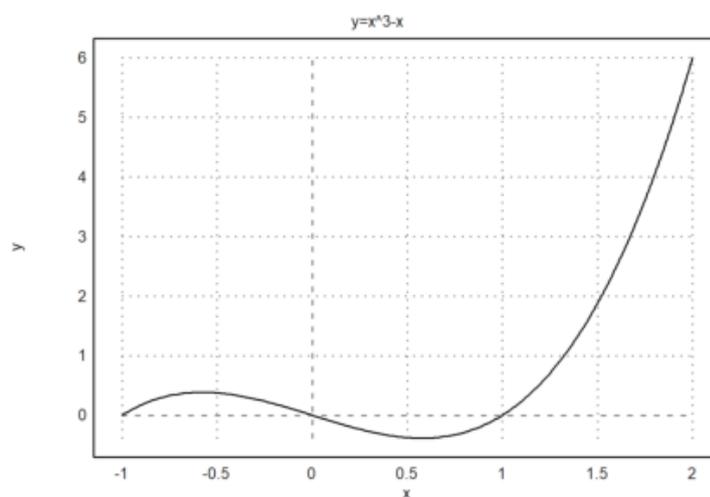
## Label Teks

Dekorasi sederhana bisa

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

Perintah label akan memplot ke dalam plot saat ini pada koordinat plot  $(x,y)$ . Itu bisa mengambil argumen posisi.

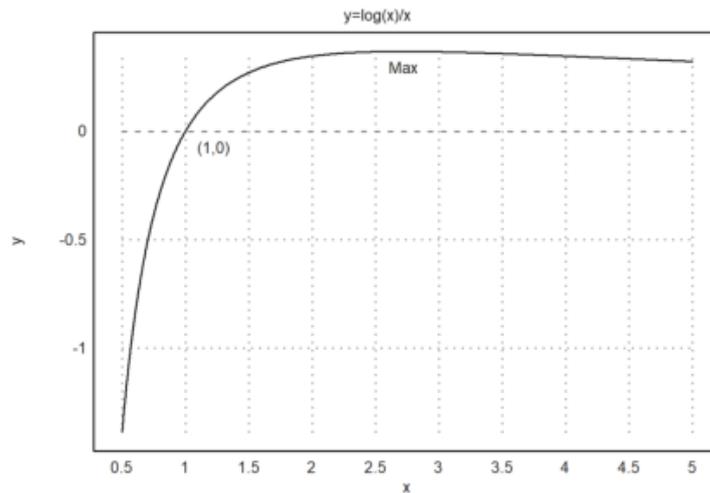
```
>plot2d("x^3-x",-1,2,title="y=x^3-x",yl="y",xl="x"):
```



```

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

```

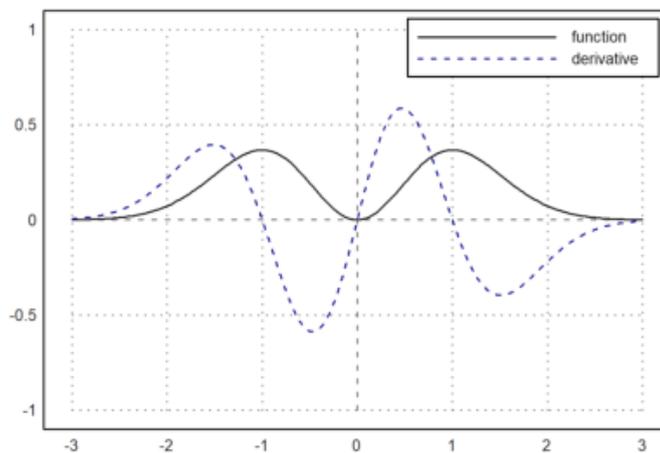


Ada juga fungsi `labelbox()`, yang dapat menampilkan fungsi dan teks. Dibutuhkan vektor string dan warna, satu item untuk setiap fungsi.

```

>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):

```

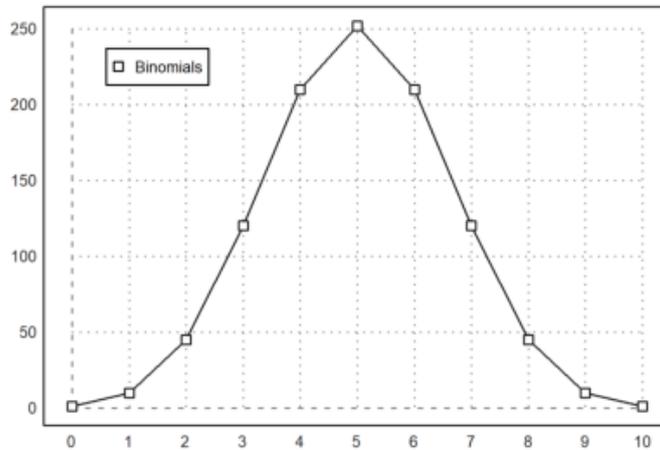


Kotak ditambahkan di kanan atas secara default, tetapi > kiri menambatkannya di kiri atas. Anda dapat memindahkannya ke tempat yang Anda sukai. Posisi jangkar adalah sudut kanan atas kotak, dan angkanya adalah pecahan dari ukuran jendela grafik. Lebarinya otomatis.

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

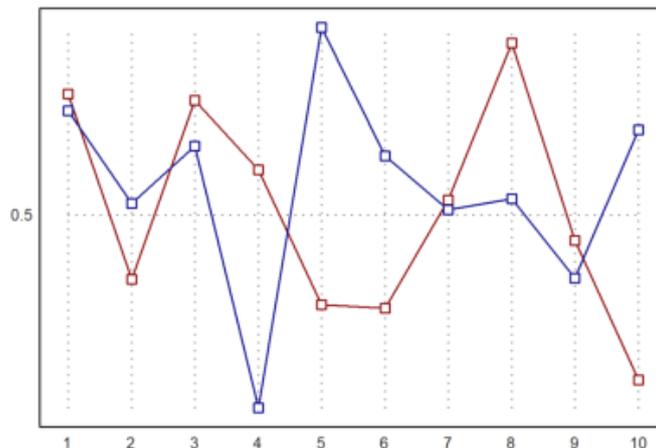
Dalam contoh berikut, hanya ada satu fungsi. Jadi kita bisa menggunakan string sebagai pengganti vektor string. Kami mengatur warna teks menjadi hitam untuk contoh ini.

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



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

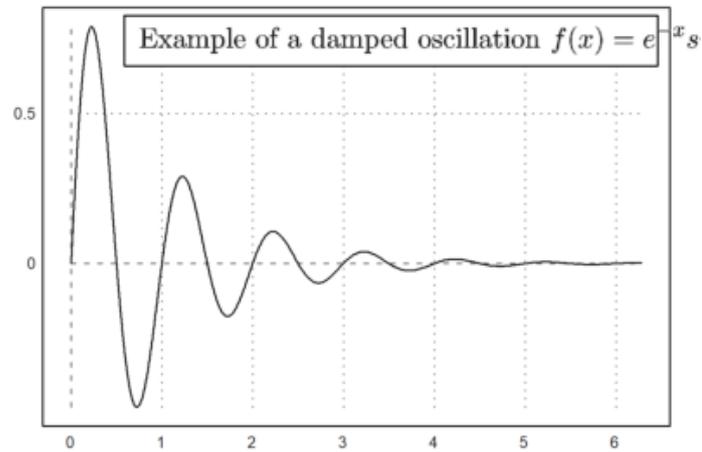
```
>statplot(1:10,random(2,10),color=[red,blue]):
```



Fitur serupa adalah fungsi `textbox()`.

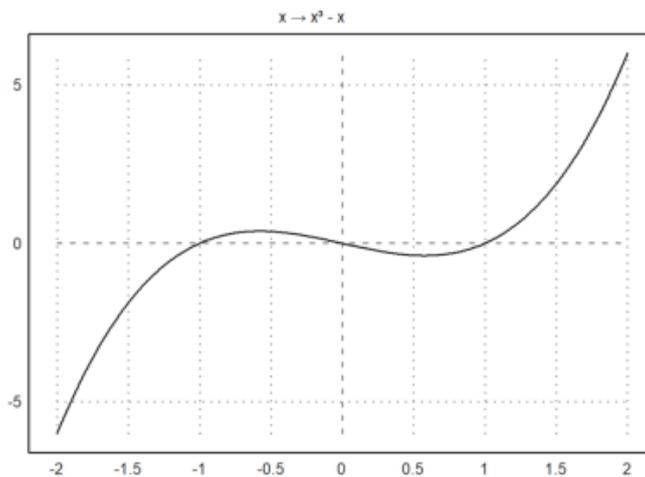
Lebar secara default adalah lebar maksimal dari baris teks. Tapi itu bisa diatur oleh pengguna juga.

```
>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):
```



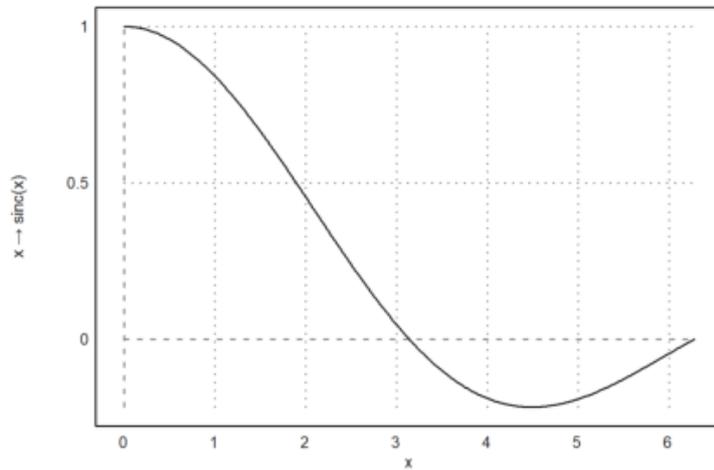
Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaks EMT untuk mengetahui lebih lanjut tentang string Unicode).

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



Label pada sumbu x dan y bisa vertikal, begitu juga sumbunya.

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



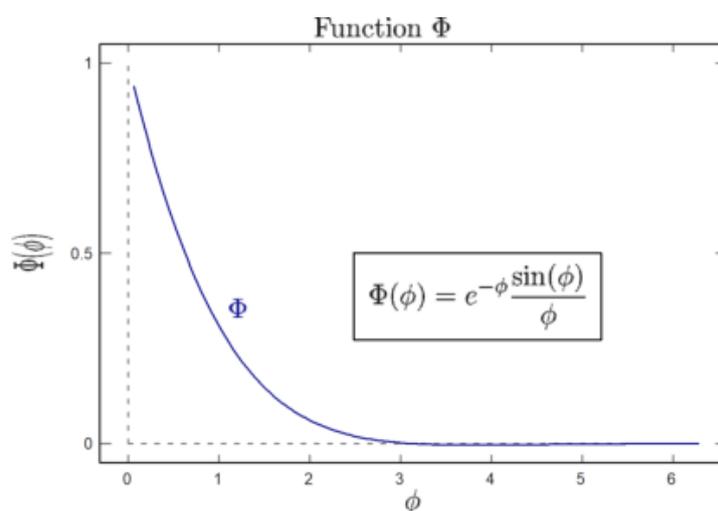
## LaTeX

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

Perhatikan, bahwa penguraian LaTeX lambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, Anda harus memanggil latex() sebelum loop sekali dan menggunakan hasilnya (gambar dalam matriks RGB).

Dalam plot berikut, kami menggunakan LaTeX untuk label x dan y, label, kotak label, dan judul plot.

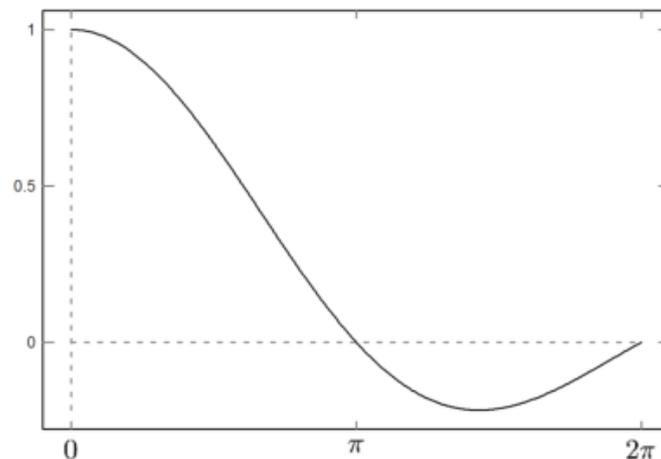
```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
> title=latex("\text{Function }\Phi$"), ...
> xl=latex("\phi"),yl=latex("\Phi(\phi)")); ...
>textbox( ...
> latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}"),x=0.8,y=0.5); ...
>label(latex("\Phi",color=blue),1,0.4):
```



Seringkali, kami menginginkan spasi dan label teks non-konformal pada sumbu x. Kita dapat menggunakan `xaxis()` dan `yaxis()` seperti yang akan kita tunjukkan nanti.

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

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



Tentu saja, fungsi juga dapat digunakan.

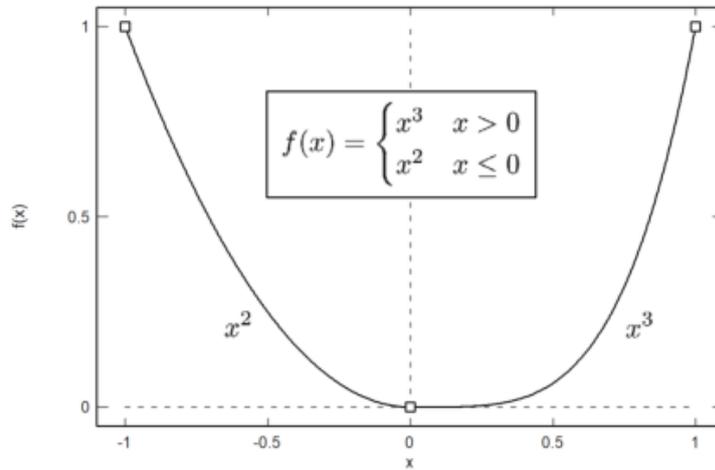
```
>function map f(x) ...
```

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

Parameter "peta" membantu menggunakan fungsi untuk vektor. Untuk plot, itu tidak perlu. Tetapi untuk mendemonstrasikan vektorisasi itu berguna, kami menambahkan beberapa poin kunci ke plot di  $x=-1$ ,  $x=0$  dan  $x=1$ .

Pada plot berikut, kami juga memasukkan beberapa kode LaTeX. Kami menggunakannya untuk dua label dan kotak teks. Tentu saja, Anda hanya akan 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

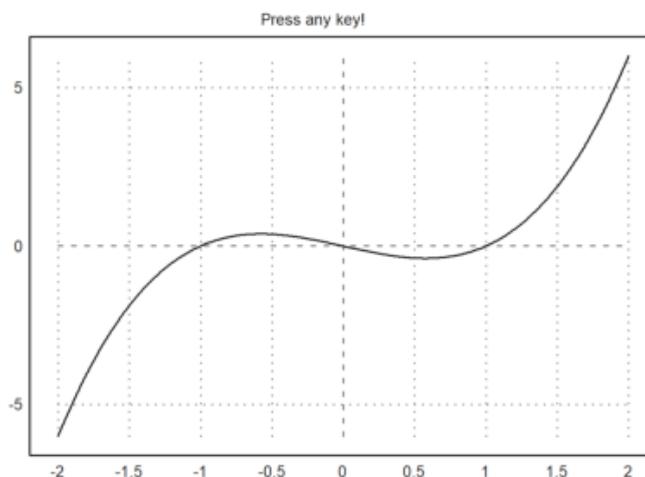
Saat memplot fungsi atau ekspresi, parameter `>user` memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kursor atau mouse. Pengguna dapat

- perbesar dengan + atau -
- pindahkan plot dengan tombol kursor
- pilih jendela plot dengan mouse
- atur ulang tampilan dengan spasi
- keluar dengan kembali

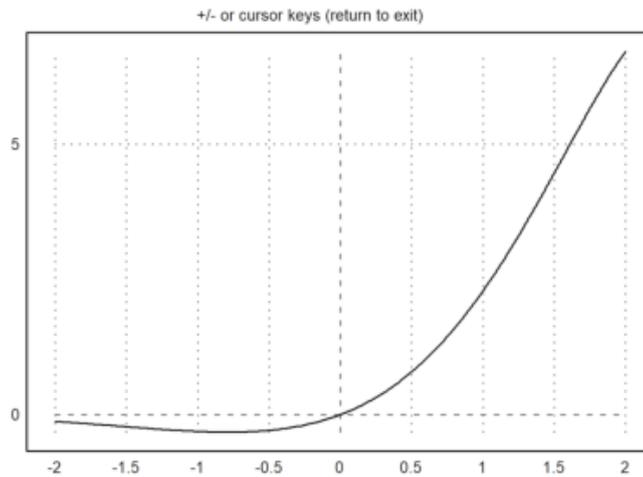
Tombol spasi akan mengatur ulang plot ke jendela plot asli.

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

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



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



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

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

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

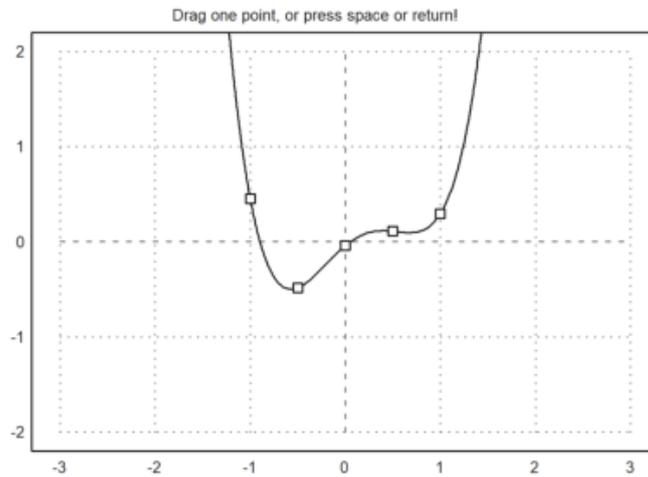
```
>function plotf(xp,yp,select) ...

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

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

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

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



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

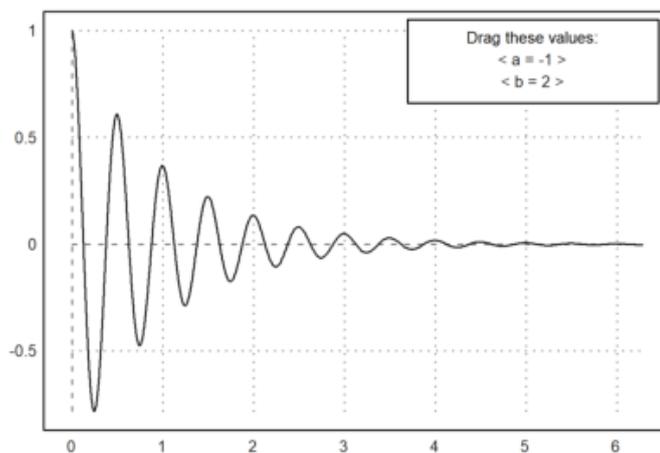
Pertama kita membutuhkan fungsi plot.

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

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

Ada slider interaktif, yang dapat mengatur nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

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



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

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

```

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

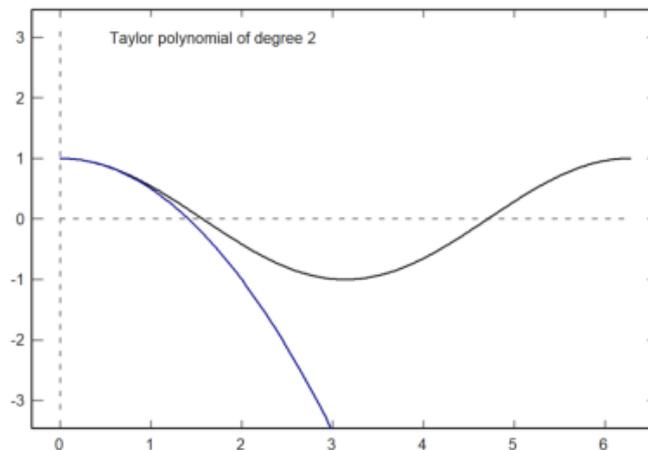
```

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

```

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

```



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

```

>function dragtest ...

```

```

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

```

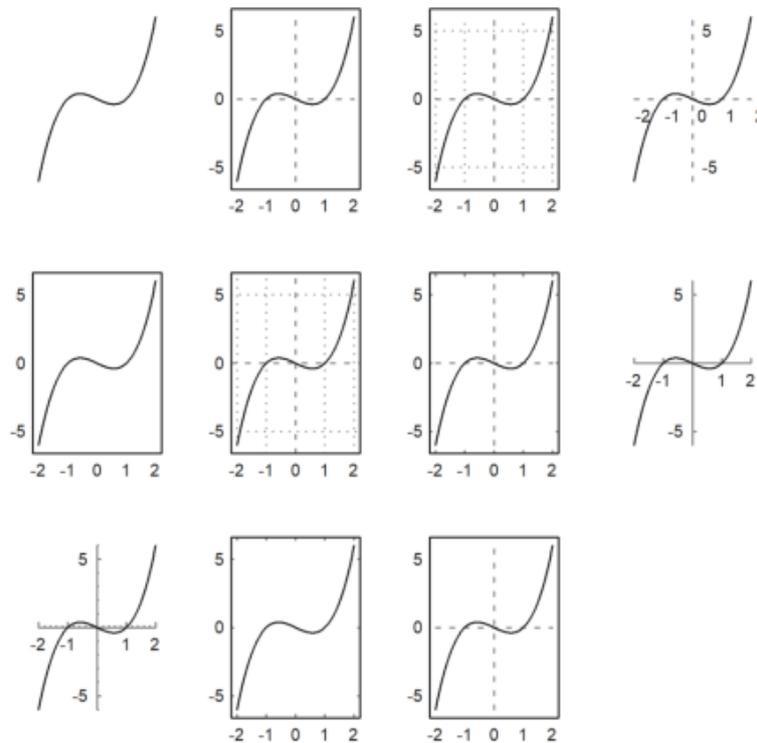
```

>dragtest // lihat hasilnya dan cobalah lakukan!

```

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

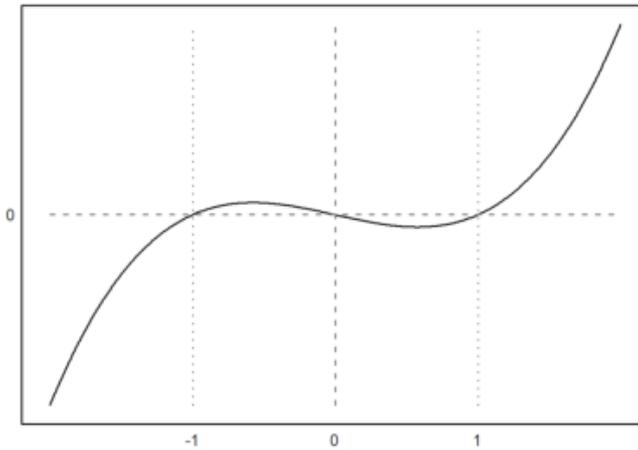
```
>aspect();
>figure(3,4); ...
> figure(1); plot2d("x^3-x",grid=0); ... // no grid, frame or axis
> figure(2); plot2d("x^3-x",grid=1); ... // x-y-axis
> figure(3); plot2d("x^3-x",grid=2); ... // default ticks
> figure(4); plot2d("x^3-x",grid=3); ... // x-y- axis with labels inside
> figure(5); plot2d("x^3-x",grid=4);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



Parameter <frame mematikan frame, dan framecolor=blue mengatur frame ke warna biru.

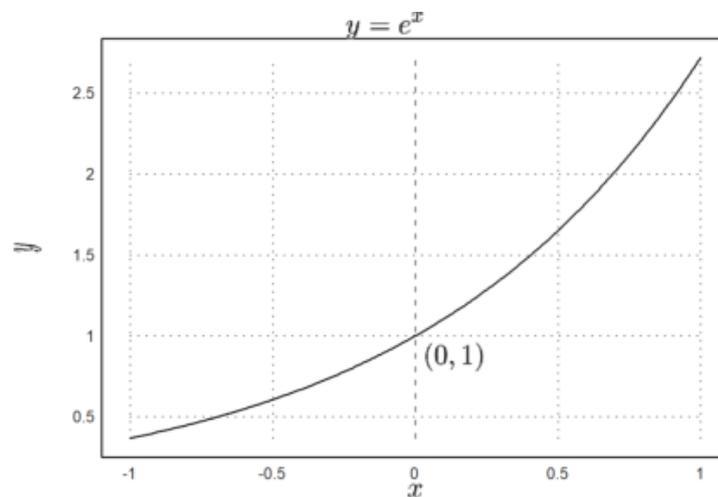
Jika Anda ingin centang sendiri, Anda dapat menggunakan style=0, dan menambahkan semuanya nanti.

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



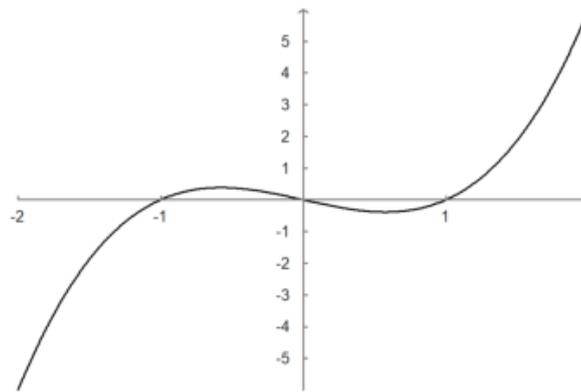
Untuk judul plot dan label sumbu, lihat contoh berikut.

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



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

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

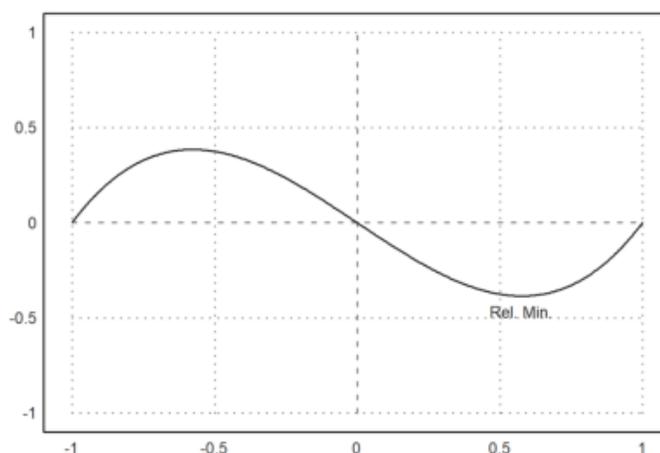


Teks pada plot dapat diatur dengan `label()`. Dalam contoh berikut, "lc" berarti tengah bawah. Ini mengatur posisi label relatif terhadap koordinat plot.

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

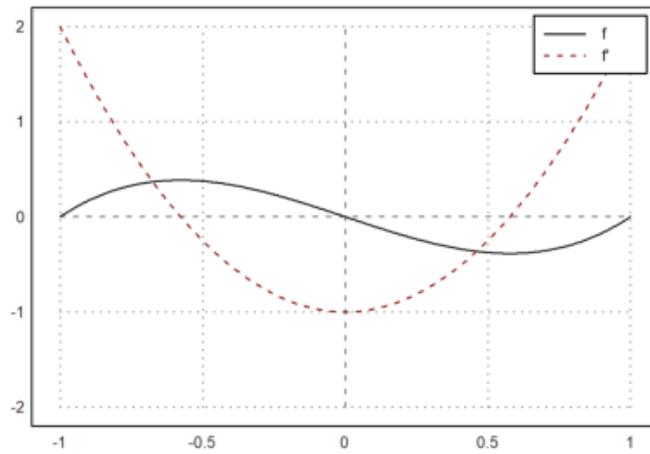
$$x^3 - x$$

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

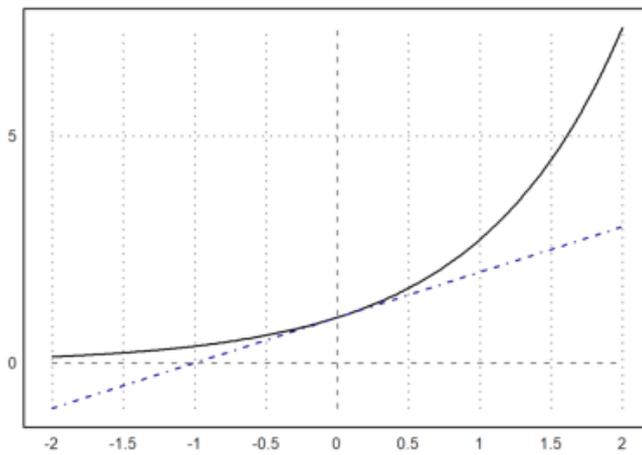


Ada juga kotak teks.

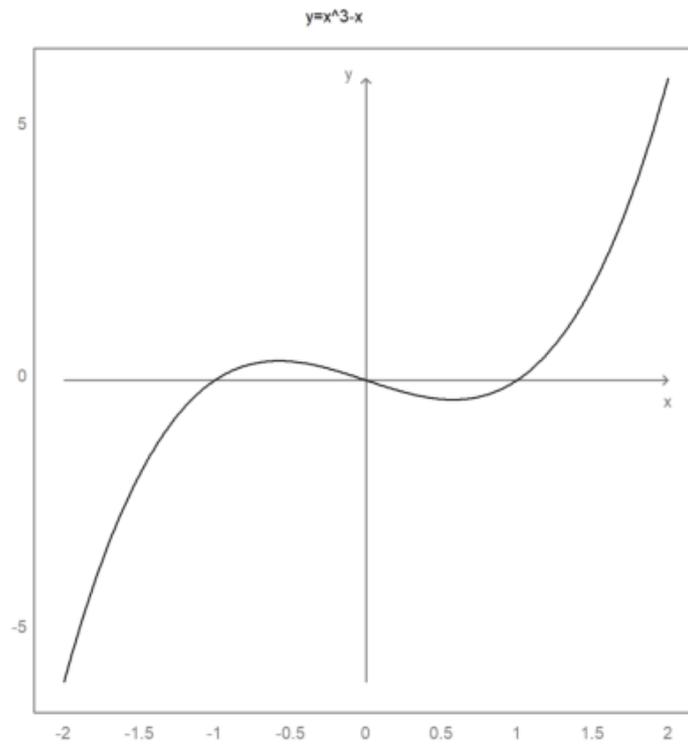
```
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // derivative
>labelbox(["f","f'"],["-","--"],[black,red]): // label box
```



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



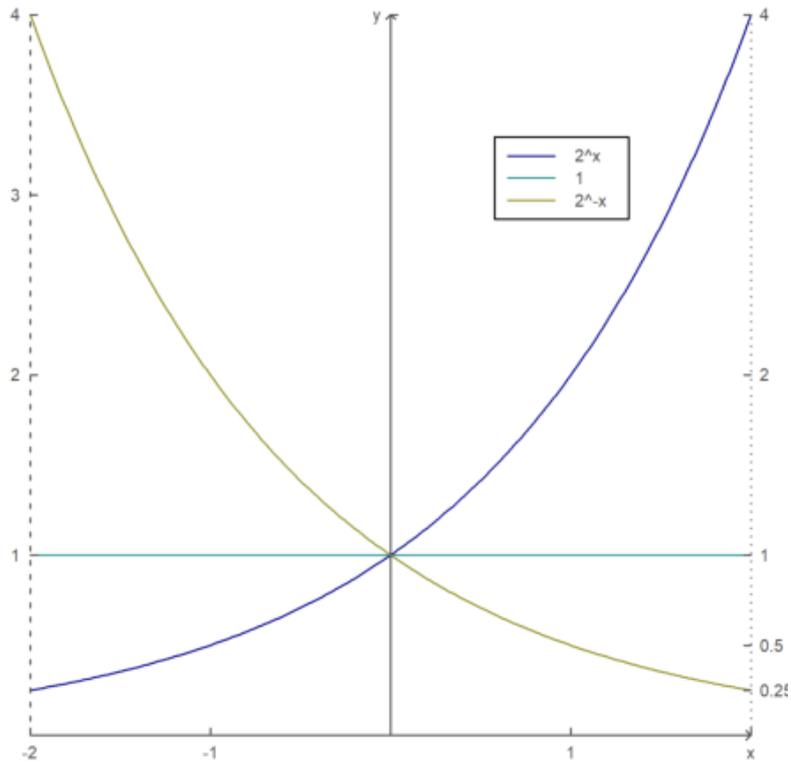
```
>gridstyle("->", color=gray, textcolor=gray, framecolor=gray); ...
> plot2d("x^3-x", grid=1); ...
> settitle("y=x^3-x", color=black); ...
> label("x", 2, 0, pos="bc", color=gray); ...
> label("y", 0, 6, pos="cl", color=gray); ...
> reset():
```



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

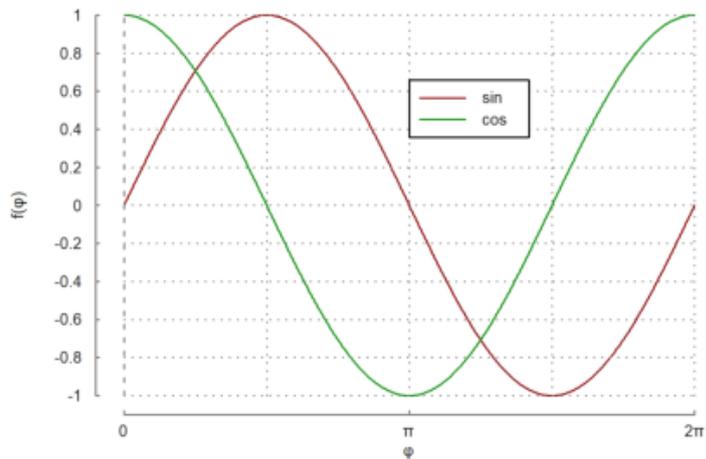
Perintah `fullwindow()` memperluas jendela plot karena kita tidak lagi membutuhkan tempat untuk label di luar jendela plot. Gunakan `shrinkwindow()` atau `reset()` untuk mengatur ulang ke default.

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



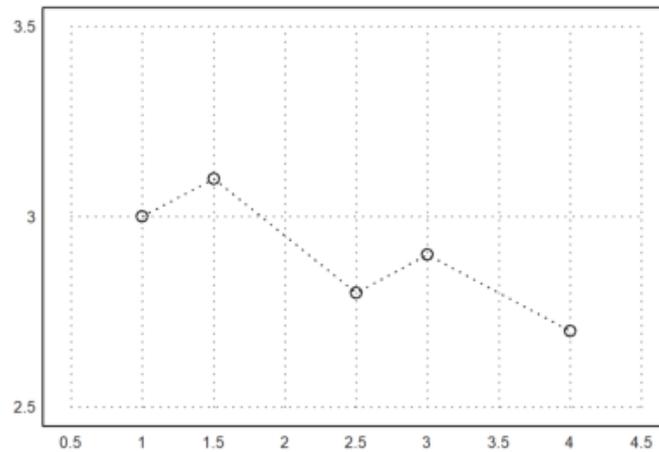
Berikut adalah contoh lain, di mana string Unicode digunakan dan sumbu di luar area plot.

```
>aspect(1.5);
>plot2d(["sin(x)", "cos(x)"], 0, 2pi, color=[red, green], <grid, <frame); ...
> xaxis(-1.1, (0:2)*pi, xt=["0", u"π", u"2π"], 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"φ"); ylabel(u"f(φ)");
```

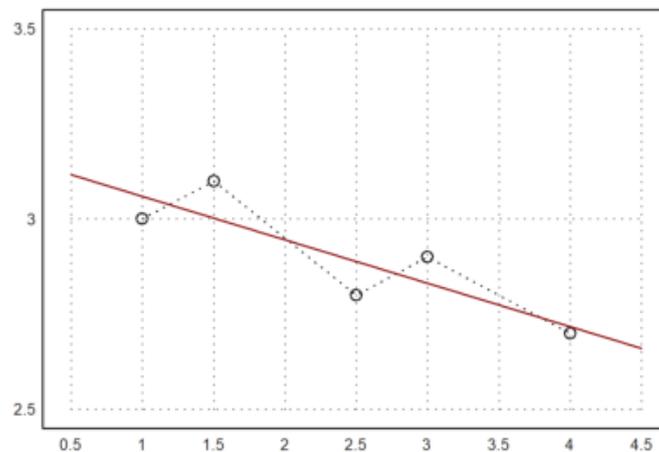


## Merencanakan Data 2D





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



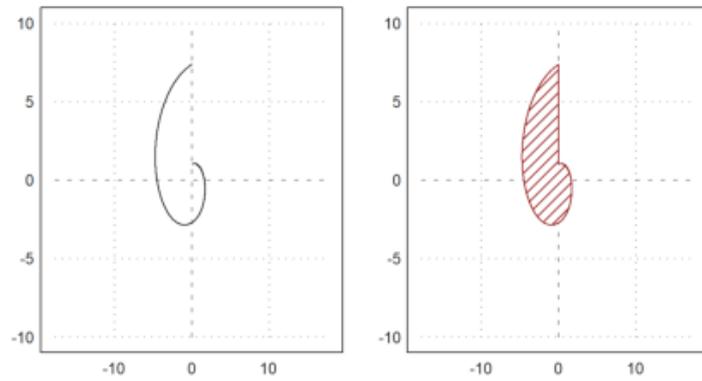
## Menggambar Daerah Yang Dibatasi Kurva

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

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

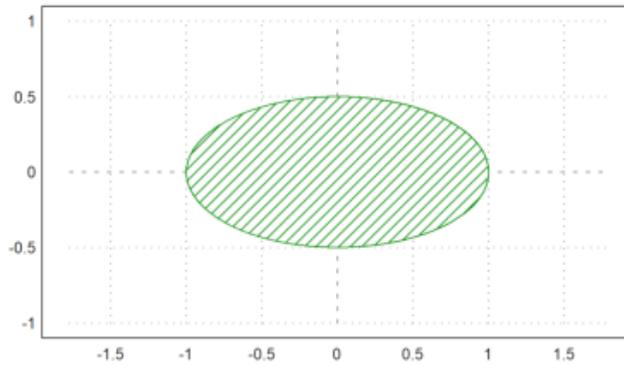
Warna isian ditentukan oleh argumen "fillcolor", dan pada <outline opsional mencegah menggambar batas untuk semua gaya kecuali yang default.

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

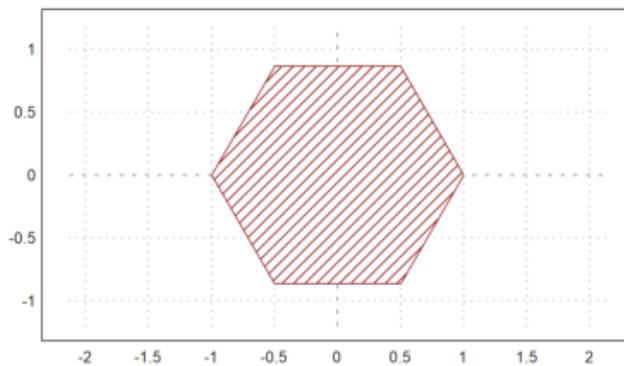


Dalam contoh berikut kami memplot elips terisi dan dua segi enam terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian berbeda.

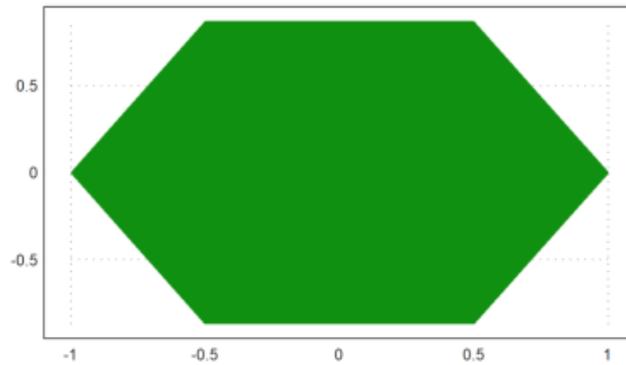
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```



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

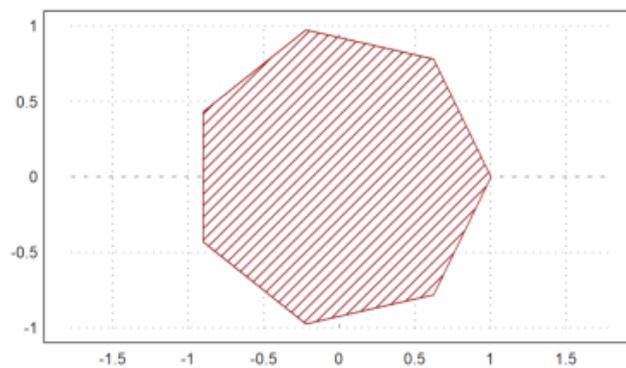


```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#") :
```



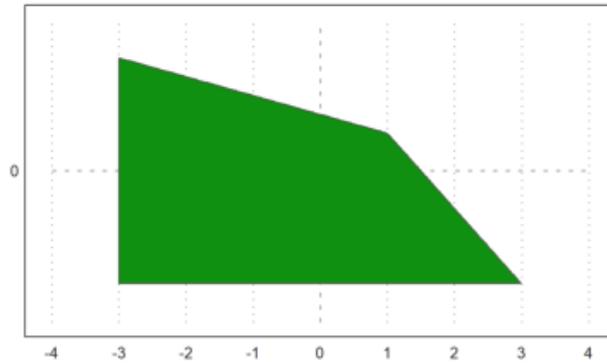
Contoh lainnya adalah segi empat, yang kita buat dengan 7 titik pada lingkaran satuan.

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



Berikut ini adalah himpunan nilai maksimal dari empat kondisi linier yang kurang dari atau sama dengan 3. Ini adalah  $A[k].v \leq 3$  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) :
```

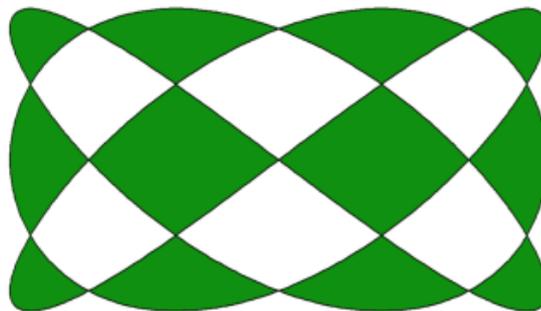


Poin utama dari bahasa matriks adalah memungkinkan untuk menghasilkan tabel fungsi dengan mudah.

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

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

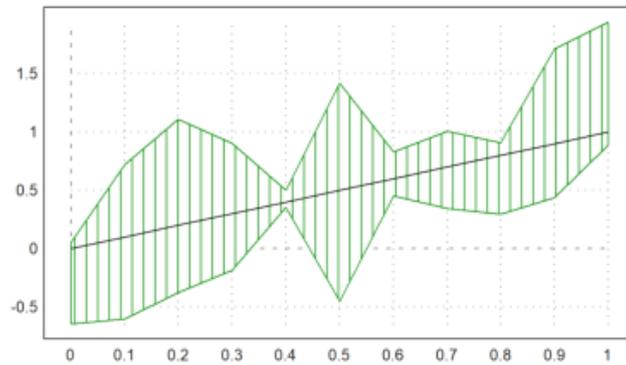
```
>plot2d(x,y,<grid,<frame,>filled):
```



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

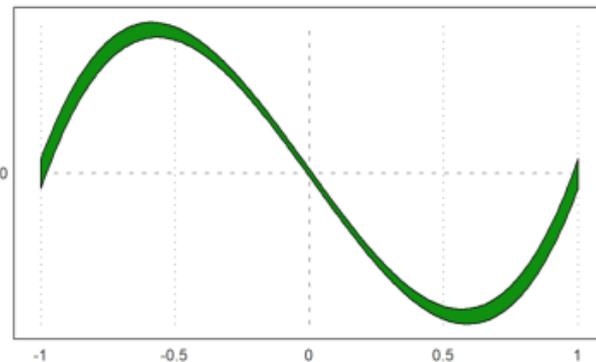
Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga digunakan untuk memplot kesalahan statistik.

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



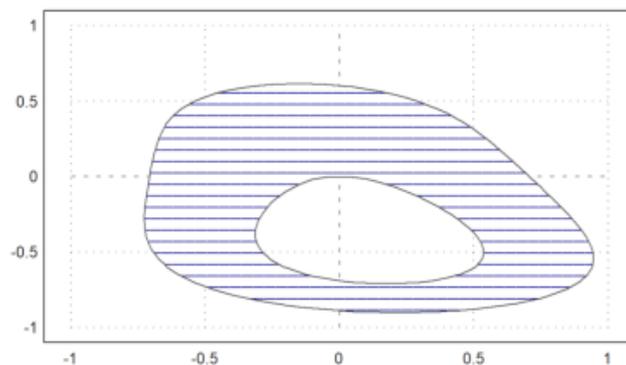
Jika  $x$  adalah vektor yang diurutkan, dan  $y$  adalah vektor interval, maka `plot2d` akan memplot rentang interval yang terisi dalam bidang. Gaya isian sama dengan gaya poligon.

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



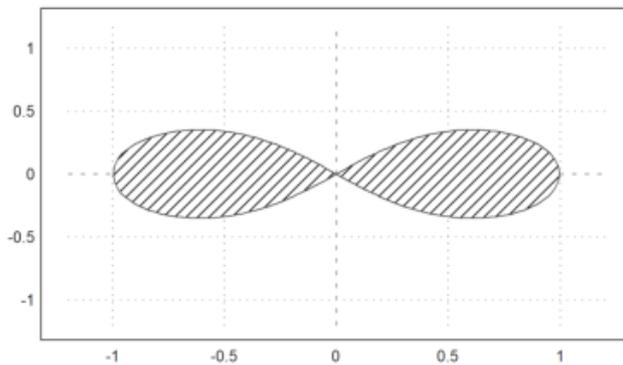
Jika  $x$  adalah vektor yang diurutkan, dan  $y$  adalah vektor interval, maka `plot2d` akan memplot rentang interval yang terisi dalam bidang. Gaya isian sama dengan gaya poligon.

```
>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
```

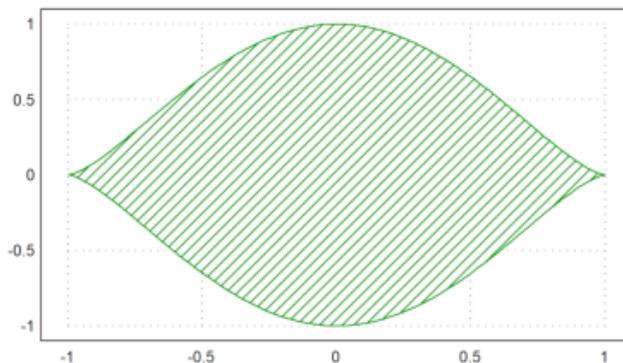


Kami juga dapat mengisi rentang nilai seperti  
lateks:  $-1 \leq (x^2+y^2)^2 - x^2 + y^2 \leq 0$ .

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



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



## Grafik Fungsi Parametrik

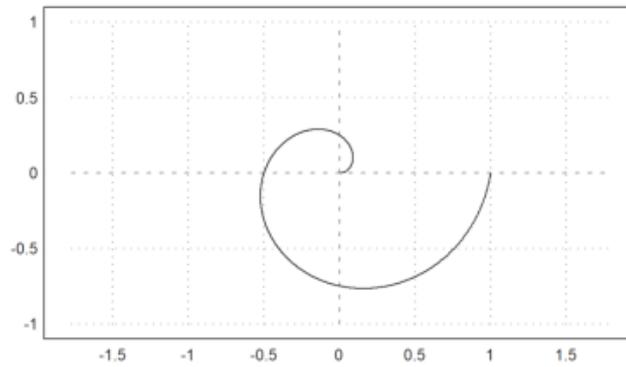
Nilai-x tidak perlu diurutkan. (x,y) hanya menggambarkan kurva. Jika x diurutkan, kurva tersebut merupakan grafik fungsi.

Dalam contoh berikut, kami memplot spiral

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

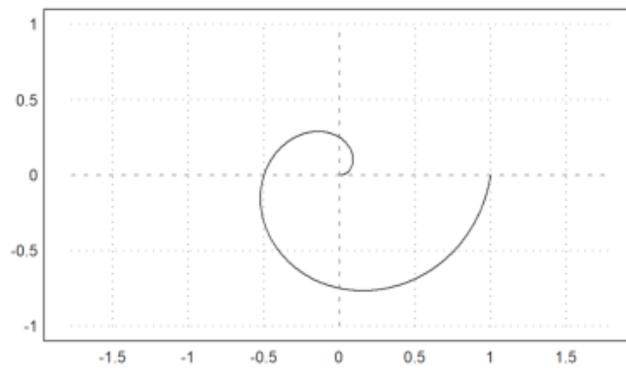
Kita perlu menggunakan banyak titik untuk tampilan yang halus atau fungsi adaptif() untuk mengevaluasi ekspresi (lihat fungsi adaptif() untuk lebih jelasnya).

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

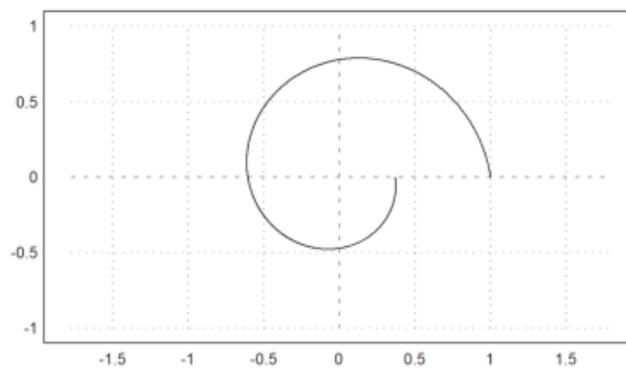


Atau, dimungkinkan untuk menggunakan dua ekspresi untuk kurva. Berikut ini plot kurva yang sama seperti di atas.

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



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



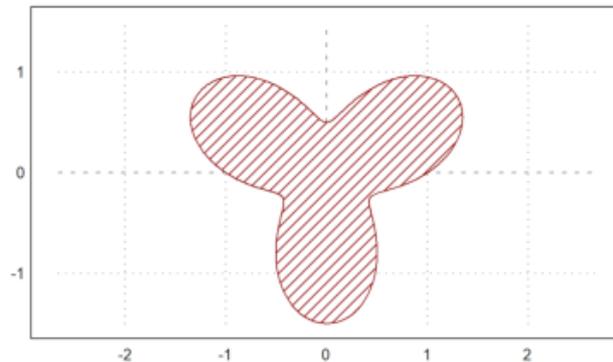
Dalam contoh berikutnya, kami memplot kurva

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

dengan

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

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



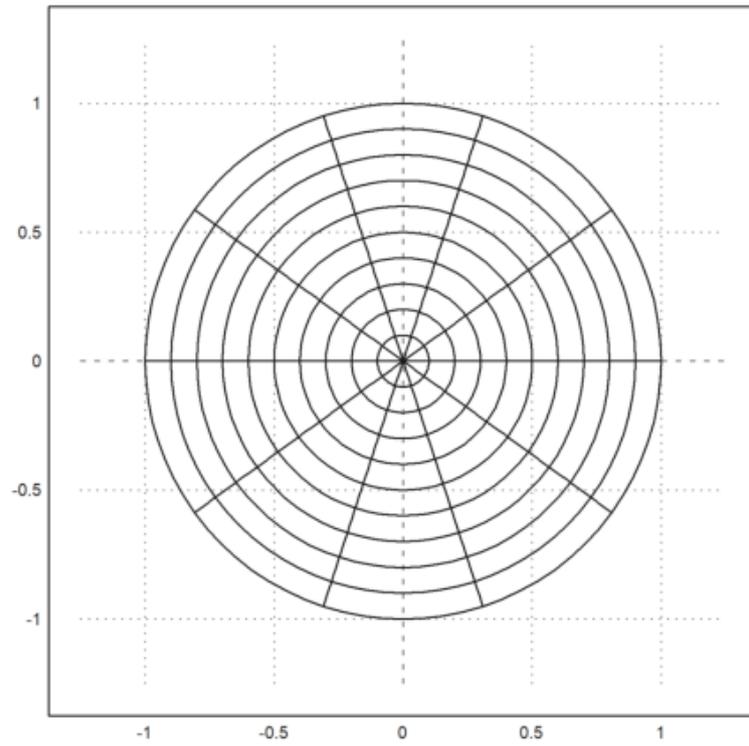
## Menggambar Grafik Bilangan Kompleks

Array bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan terhubung. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi  $1 \times 2$ ) dalam argumen `cgrid`, hanya garis kisi tersebut yang terlihat.

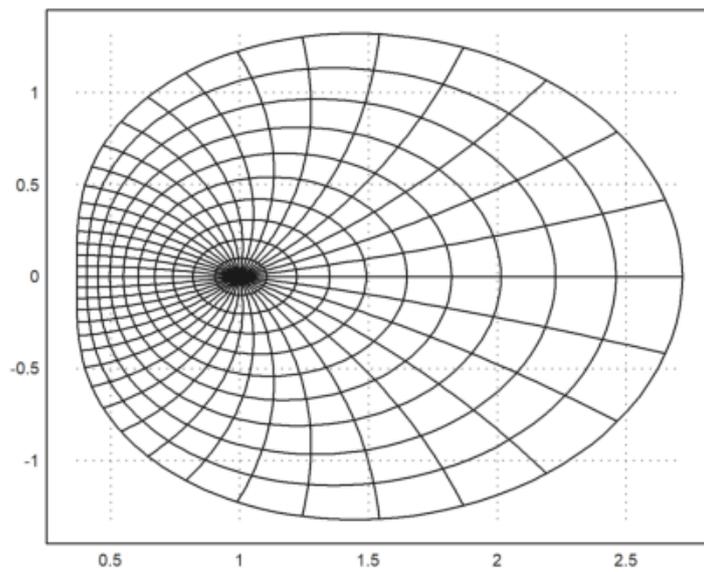
Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi di bidang kompleks.

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

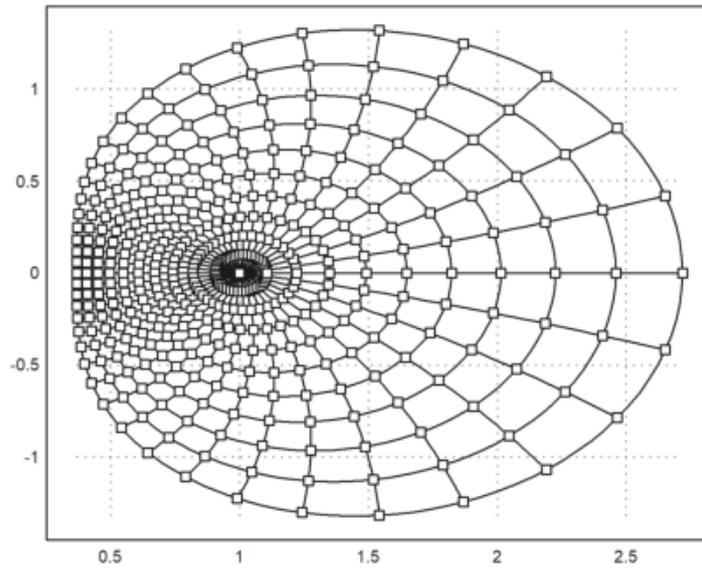
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...  
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```



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



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

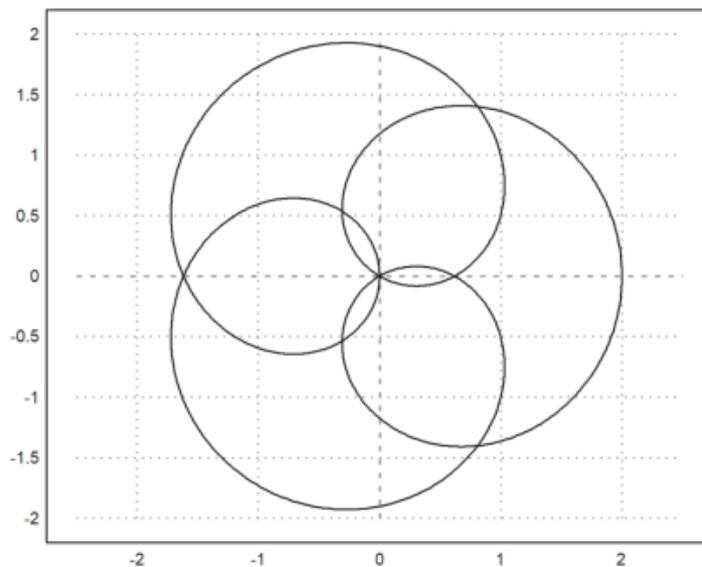


Sebuah vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian real dan bagian imajiner.

Dalam contoh, kami memplot lingkaran satuan dengan

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

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



**Plot Statistik**

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

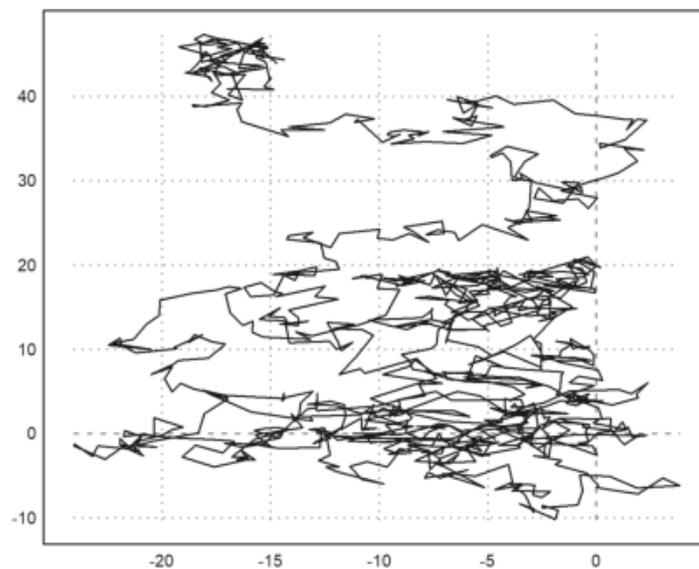
Jumlah kumulatif dari nilai terdistribusi 0-1-normal menghasilkan jalan acak.

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

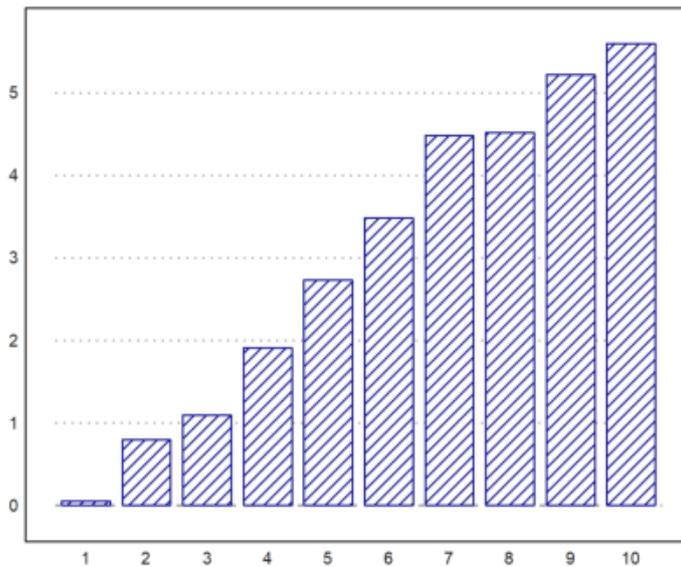


Menggunakan dua baris menunjukkan jalan dalam dua dimensi.

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

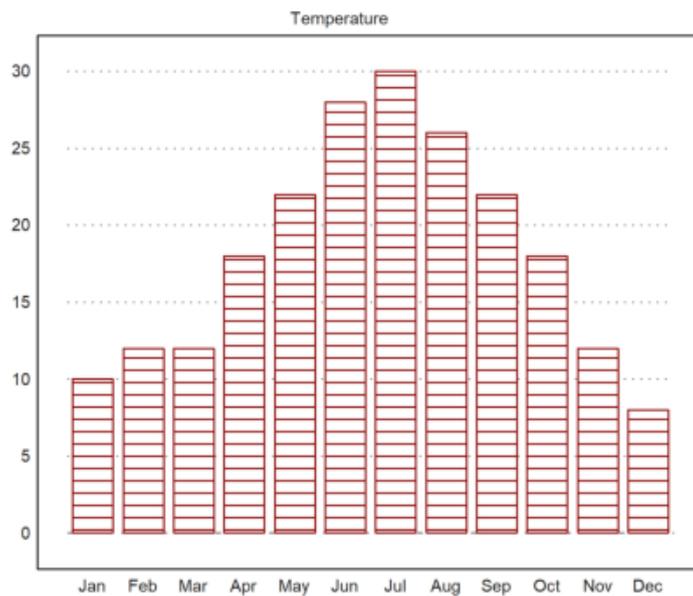


```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

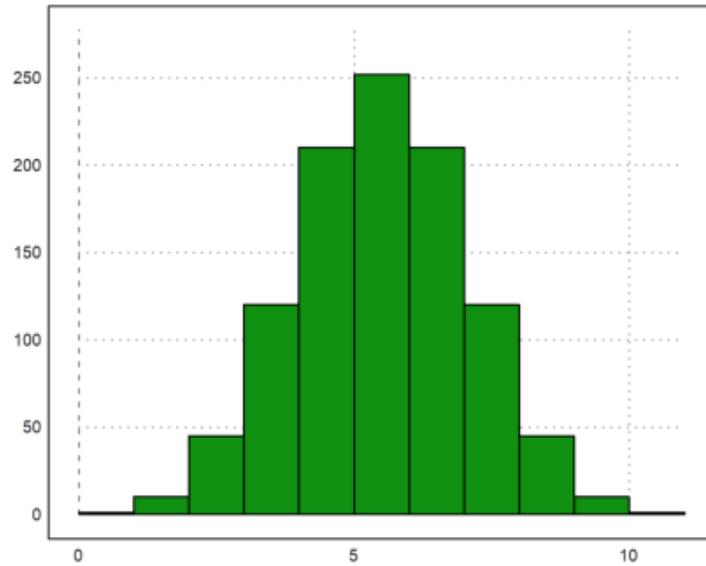


Itu juga dapat menampilkan string sebagai label.

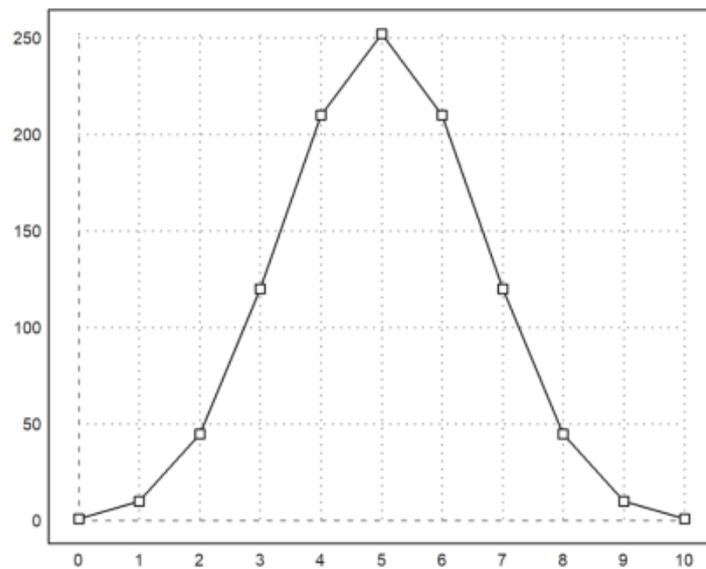
```
>months=["Jan", "Feb", "Mar", "Apr", "May", "Jun", ...  
> "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"];  
>values=[10,12,12,18,22,28,30,26,22,18,12,8];  
>columnsplot(values,lab=months,color=red,style="-");  
>title("Temperature"):
```



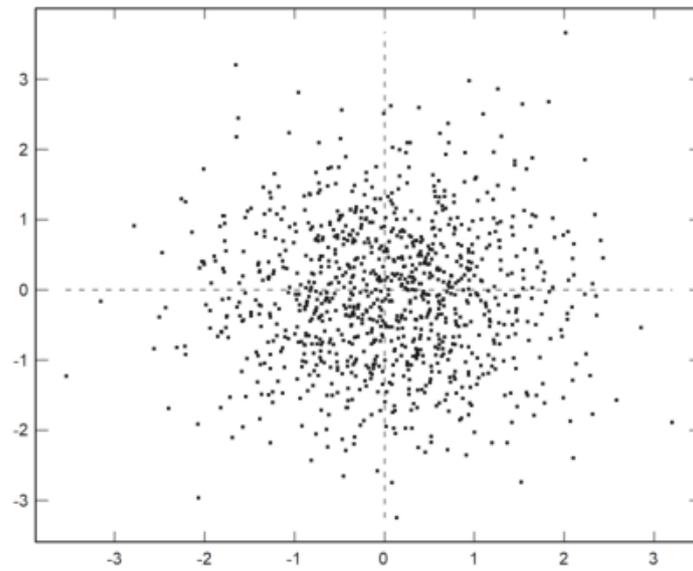
```
>k=0:10;  
>plot2d(k,bin(10,k),>bar):
```



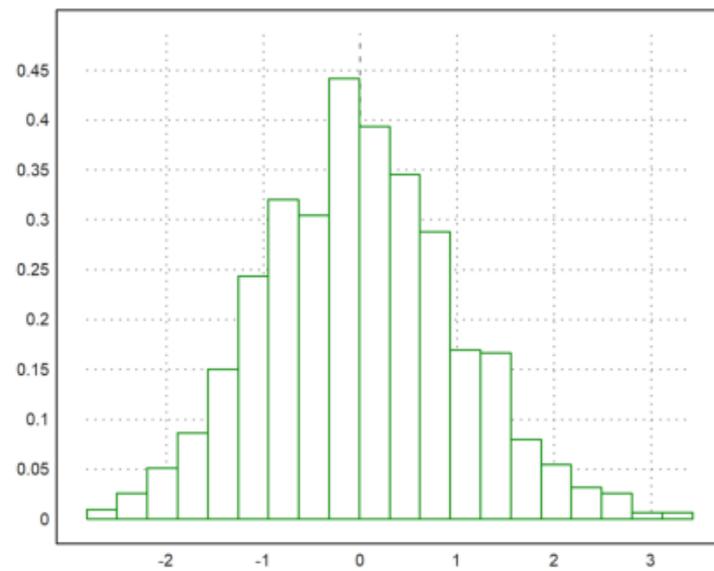
```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



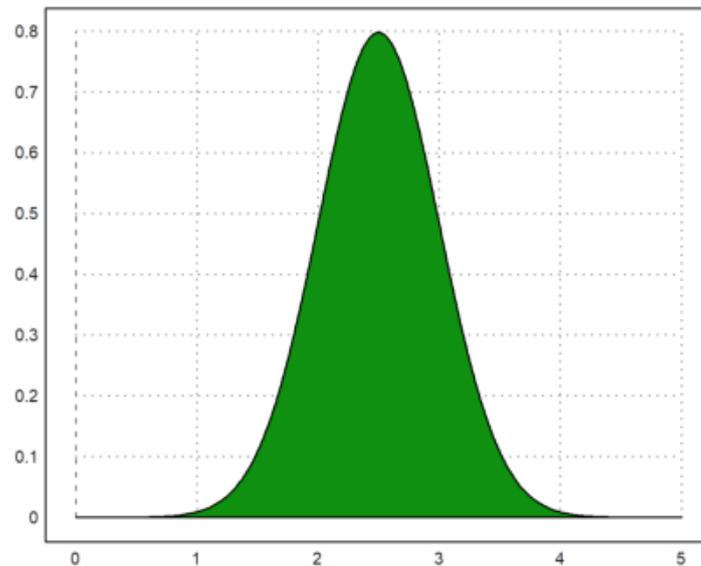
```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```



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

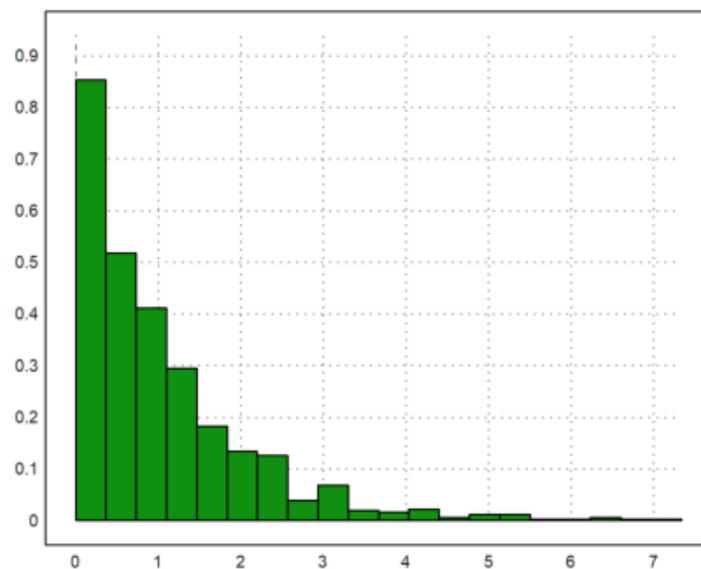


```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



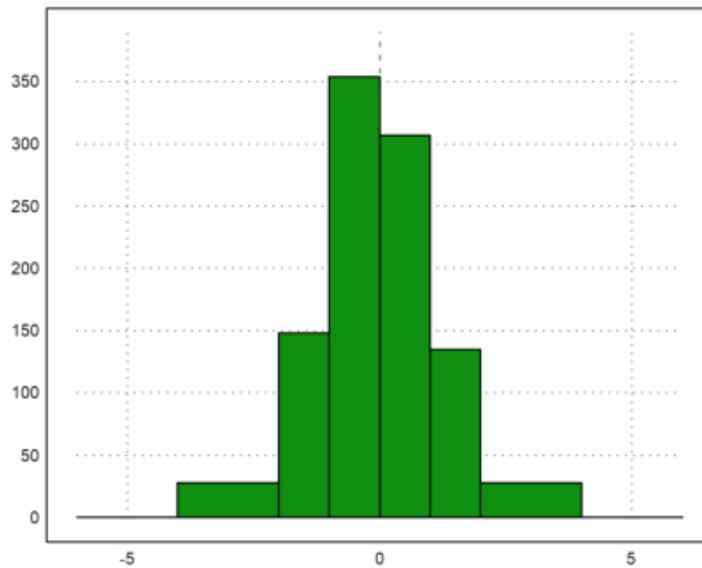
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan `distribution=n` dengan `plot2d`.

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



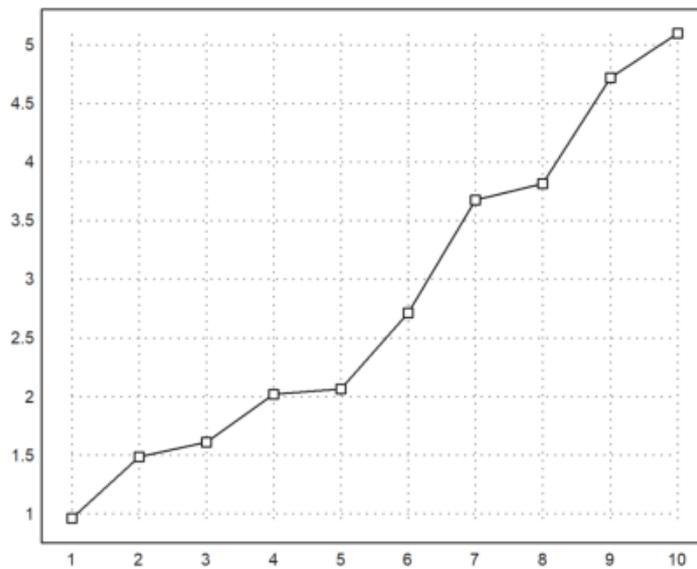
Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan `>bar` di `plot3d`, atau dengan `plot` kolom.

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

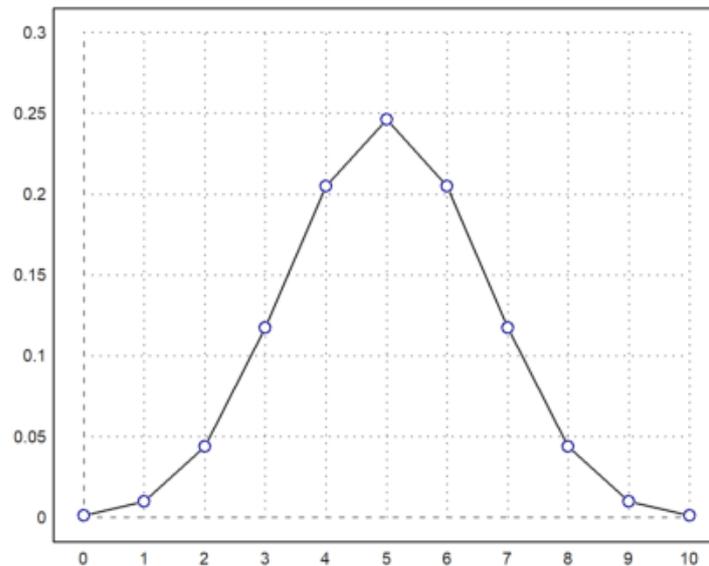


Fungsi `statplot()` menyetel gaya dengan string sederhana.

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



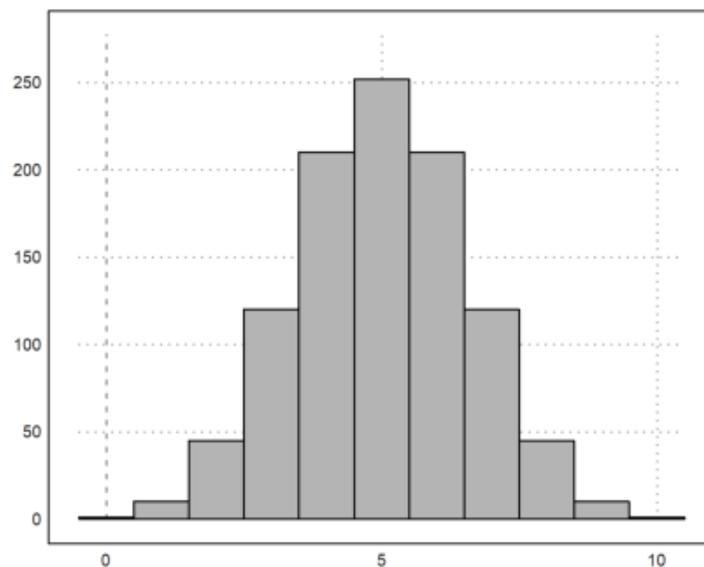
```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue) :
```



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

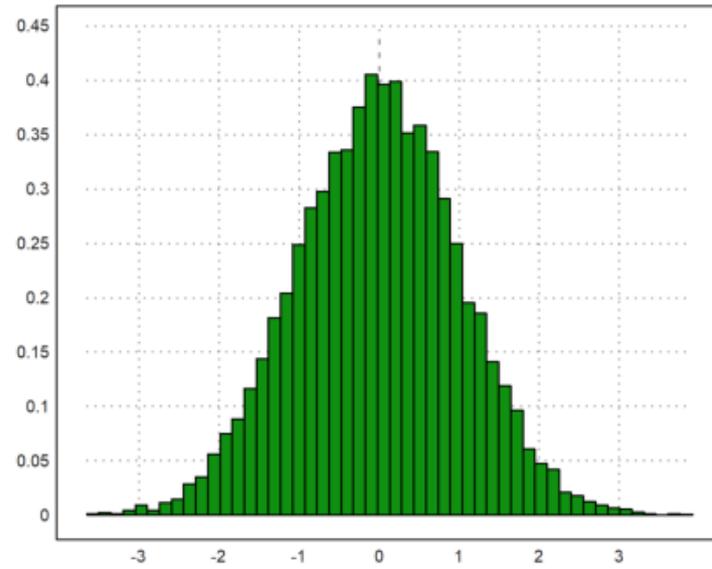
Gaya isian dapat digunakan seperti di atas.

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

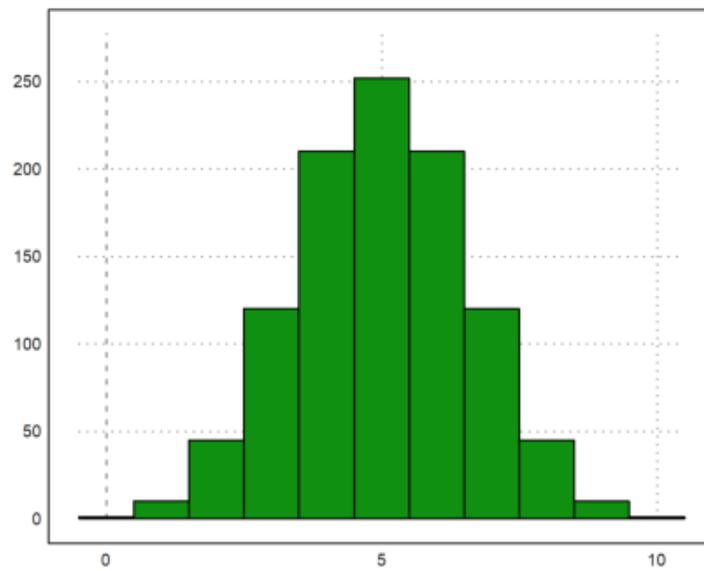


Data untuk plot batang (`bar=1`) dan histogram (`histogram=1`) dapat dinyatakan secara eksplisit dalam `xv` dan `yv`, atau dapat dihitung dari distribusi empiris dalam `xv` dengan `>distribusi` (atau `distribusi=n`). Histogram nilai `xv` akan dihitung secara otomatis dengan `>histogram`. Jika `>genap`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

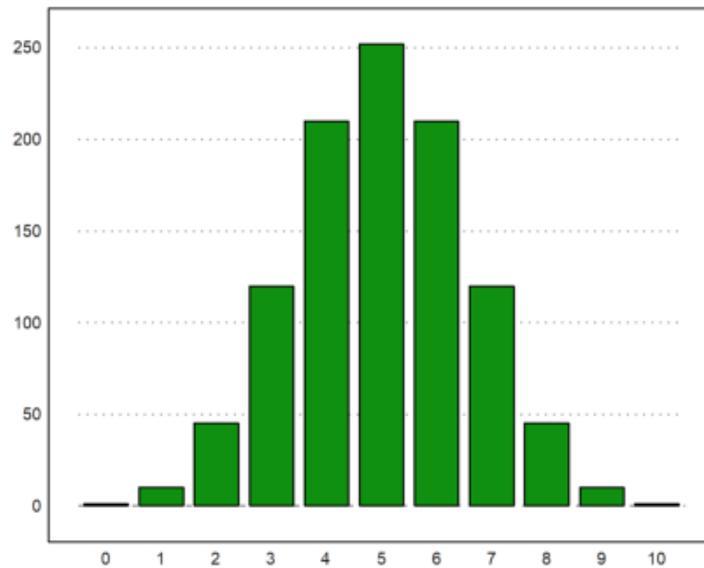
```
>plot2d(normal(10000),distribution=50):
```



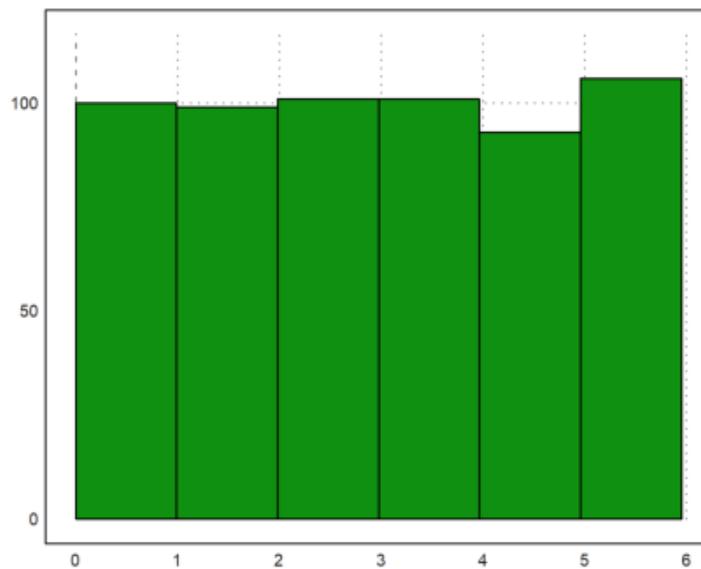
```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```



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

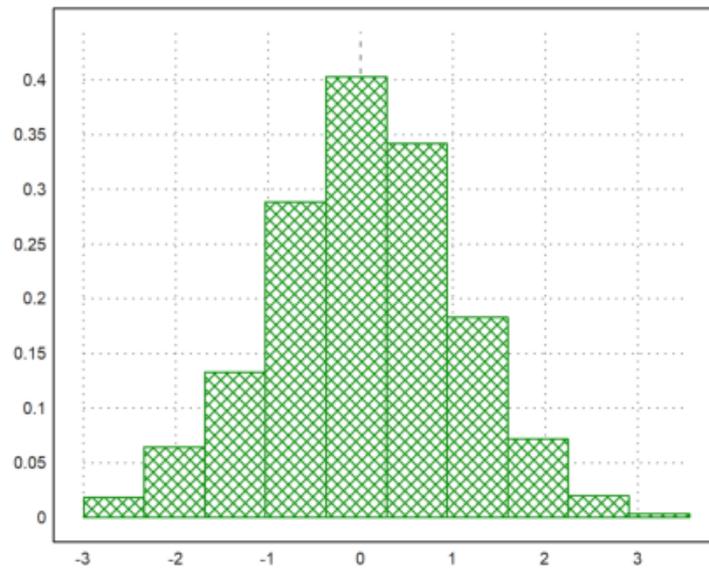


```
>plot2d(random(600)*6,histogram=6):
```



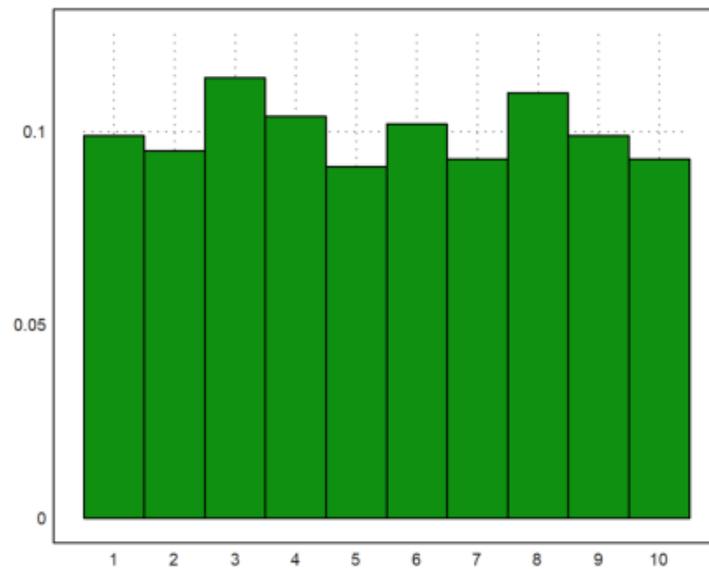
Untuk distribusi, ada parameter `distribution=n`, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan `n` sub-interval.

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



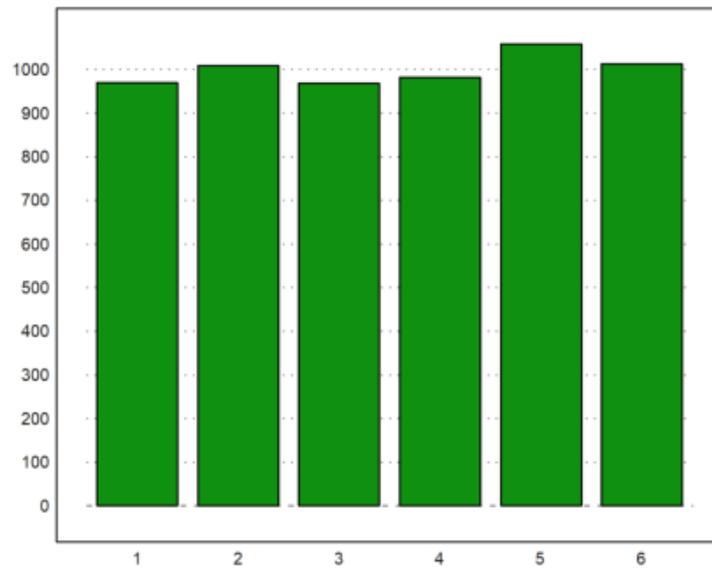
Dengan parameter `even=true`, ini akan menggunakan interval integer.

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

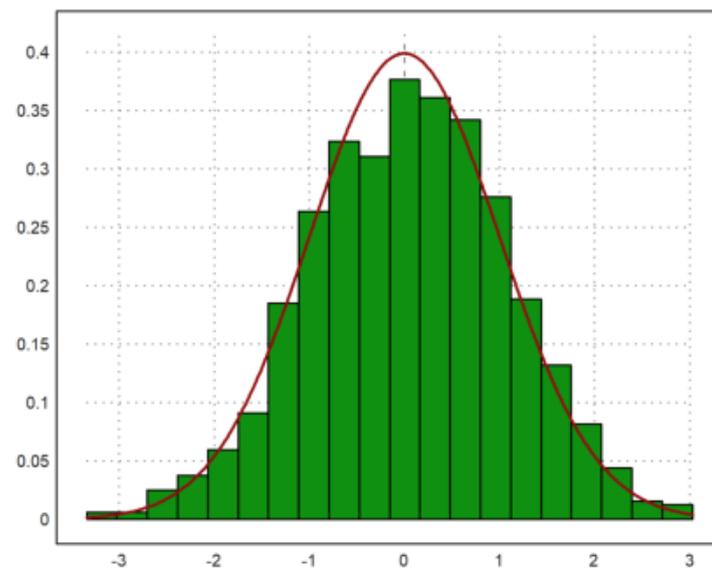


Perhatikan bahwa ada banyak plot statistik, yang mungkin berguna. Silahkan lihat tutorial tentang statistik.

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

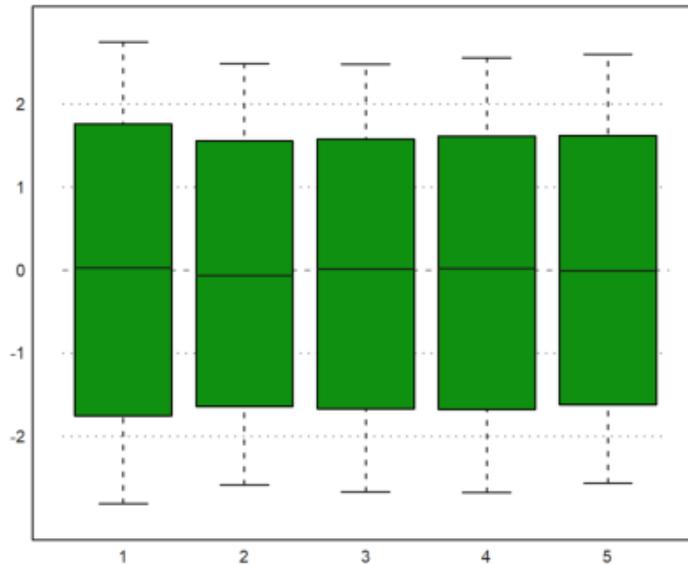


```
>plot2d(normal(1,1000),>distribution); ...
> plot2d("qnormal(x)",color=red,thickness=2,>add):
```



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

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



## Fungsi Implisit

Plot implisit menunjukkan garis level yang menyelesaikan  $f(x,y)=\text{level}$ , di mana "level" dapat berupa nilai tunggal atau vektor nilai. Jika  $\text{level}=\text{"auto"}$ , akan ada garis level  $n_c$ , yang akan menyebar antara fungsi minimum dan maksimum secara merata. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan  $\text{>hue}$  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.

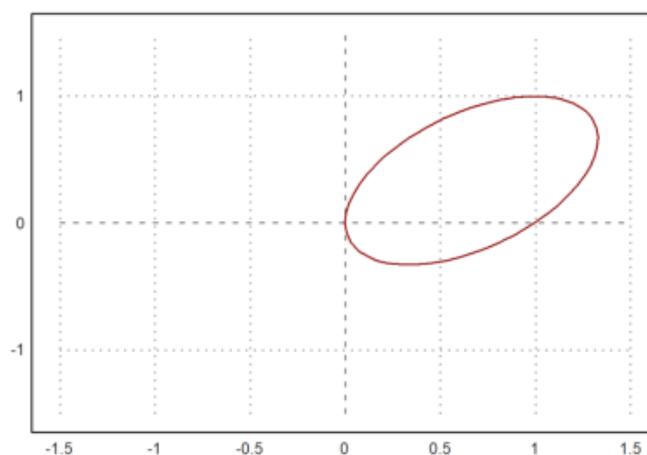
Euler dapat menandai garis level

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

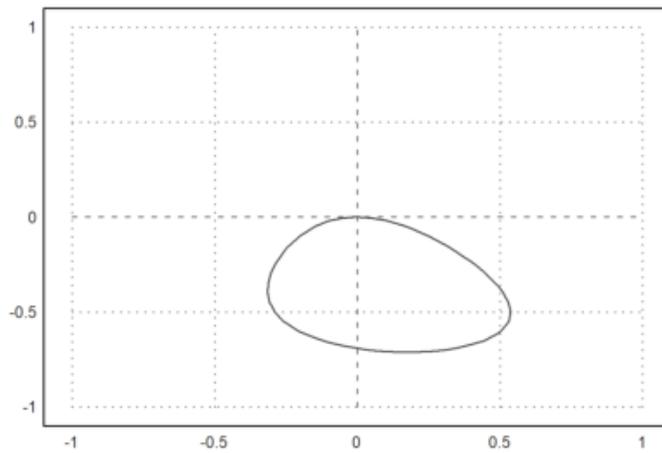
dari fungsi apapun.

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

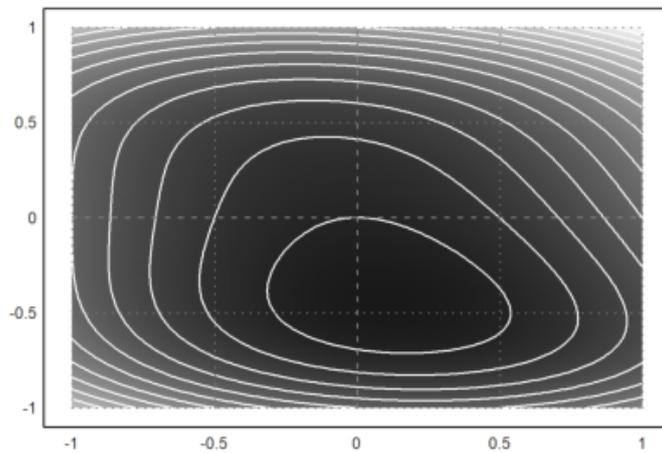
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x", r=1.5, level=0, contourcolor=red):
```



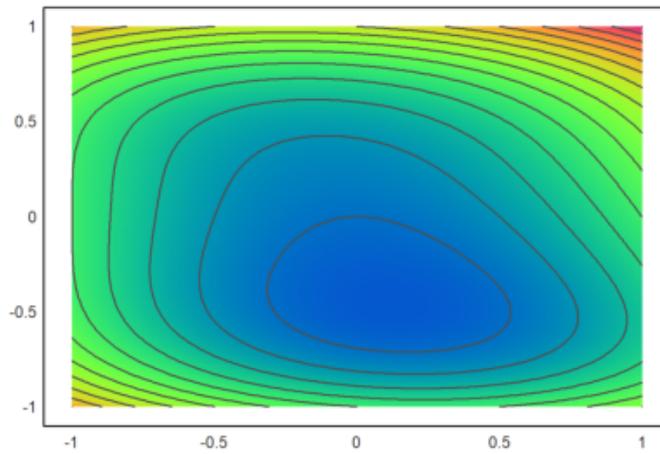
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0): // Solutions of f(x,y)=0
```



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

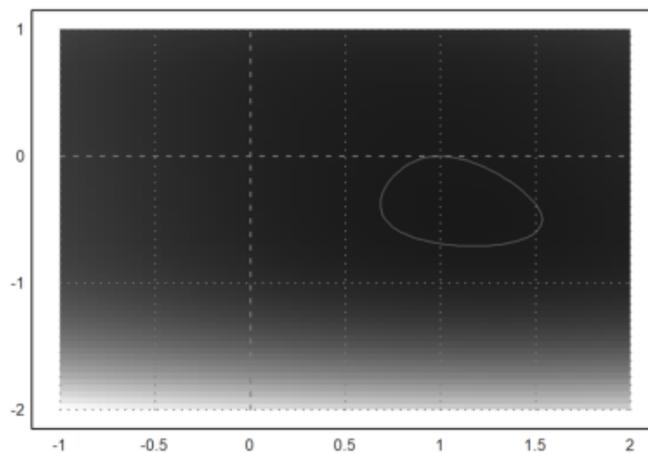


```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
```

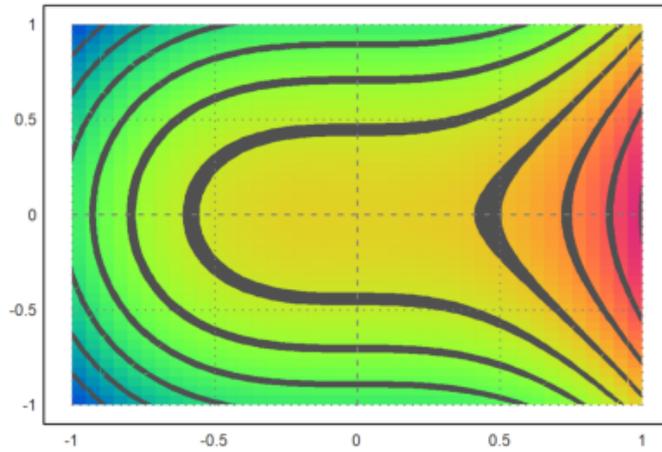


Ini berfungsi untuk plot data juga. Tetapi Anda harus menentukan rentangnya untuk label sumbu.

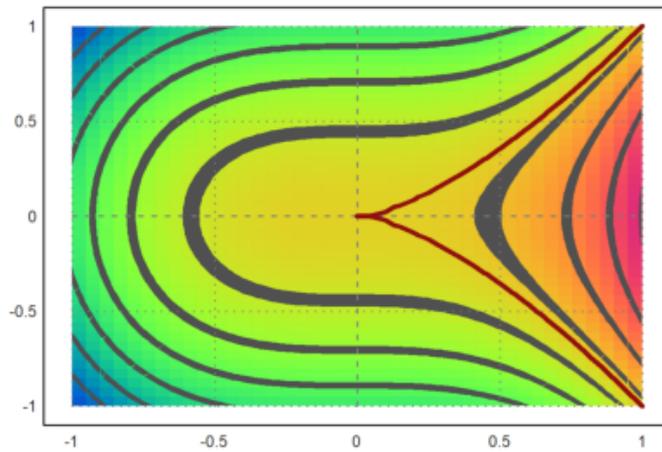
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



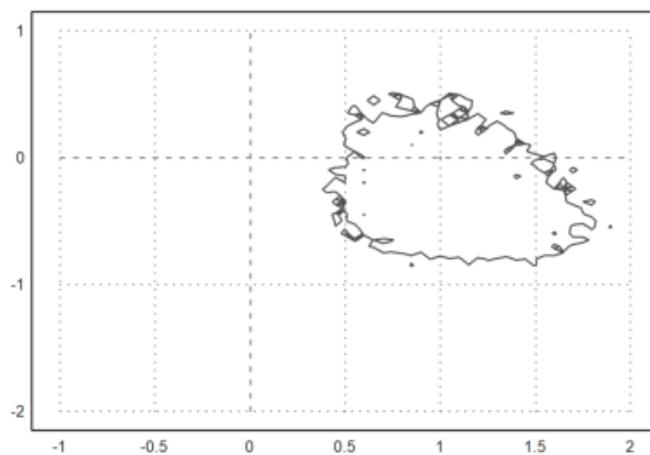
```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



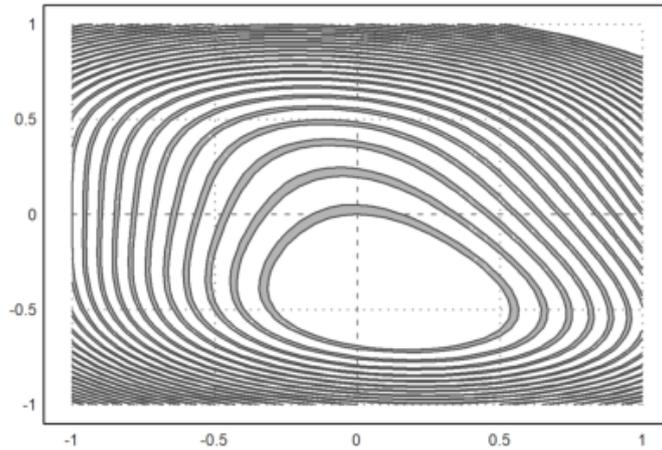
```
>plot2d("x^3-y^2", level=0, contourwidth=3, >add, contourcolor=red) :
```



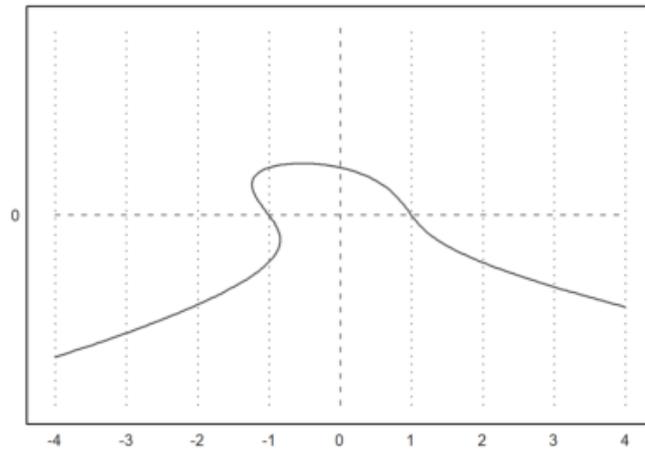
```
>z=z+normal(size(z))*0.2;
>plot2d(z, level=0.5, a=-1, b=2, c=-2, d=1) :
```



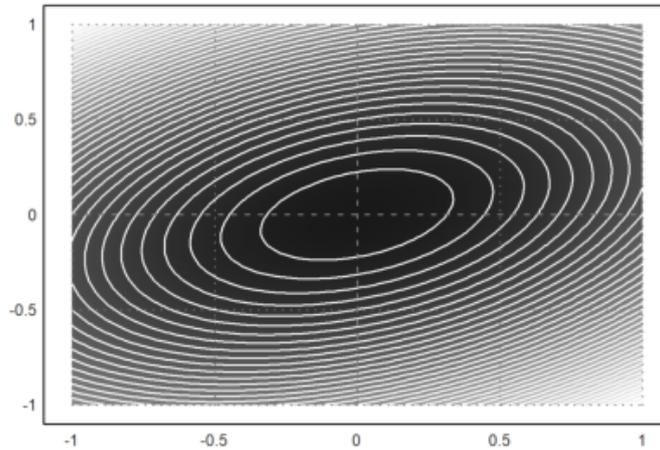
```
>plot2d(expr, level=[0:0.2:5;0.05:0.2:5.05], color=lightgray):
```



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



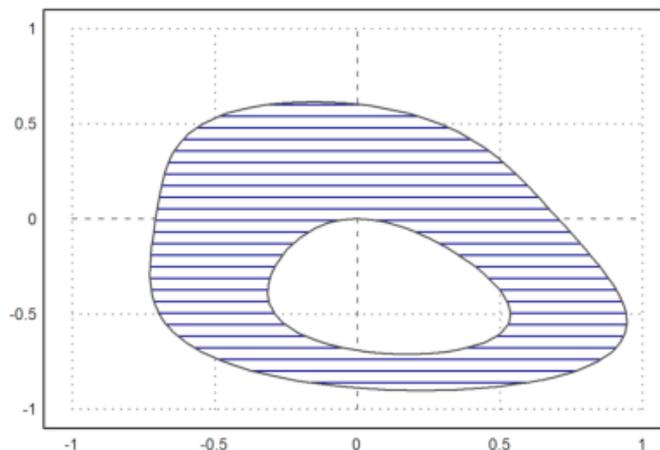
```
>plot2d("x^2+2*y^2-x*y", level=0:0.1:10, n=100, contourcolor=white, >hue):
```



Juga dimungkinkan untuk mengisi set  
 latex:  $a \leq f(x,y) \leq b$   
 dengan rentang tingkat.

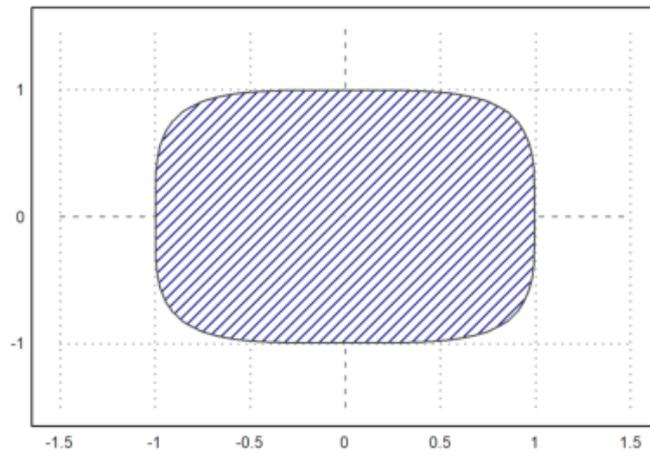
Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

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

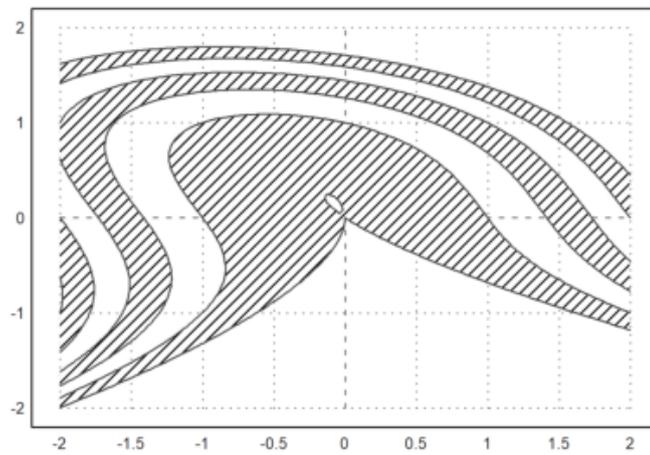


Plot implisit juga dapat menunjukkan rentang level. Kemudian level harus berupa matriks 2xn dari interval level, di mana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval. Atau, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level ke interval.

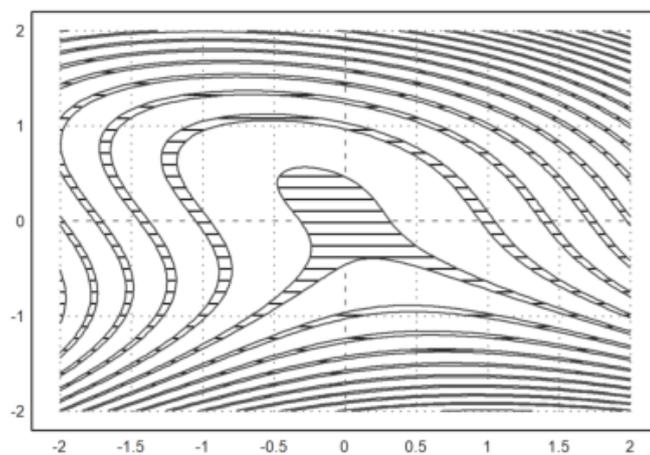
```
>plot2d("x^4+y^4", r=1.5, level=[0;1], color=blue, style="/") :
```



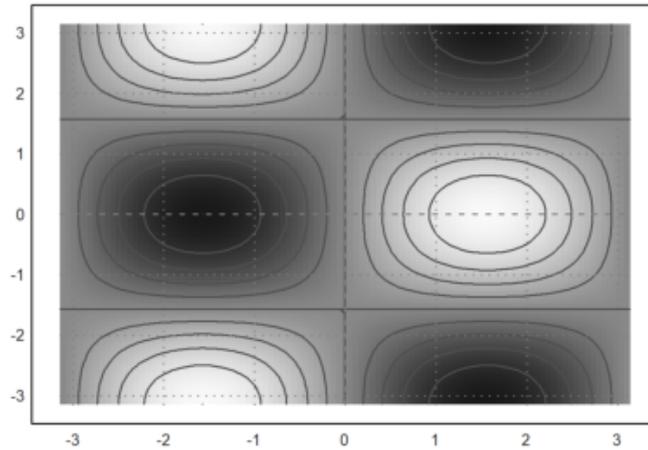
```
>plot2d("x^2+y^3+x*y", level=[0, 2, 4; 1, 3, 5], style="/", r=2, n=100) :
```



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



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



Dimungkinkan juga untuk menandai suatu wilayah

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

Ini dilakukan dengan menambahkan level dengan dua baris.

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

Illegal parameter after named parameter!

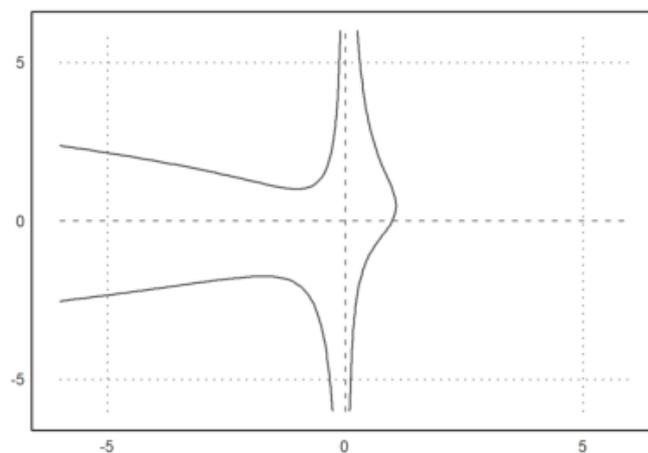
Error in:

```
... yle="#",color=red,<outline, level=[-2;0],n=100): ...
^
```

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

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

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



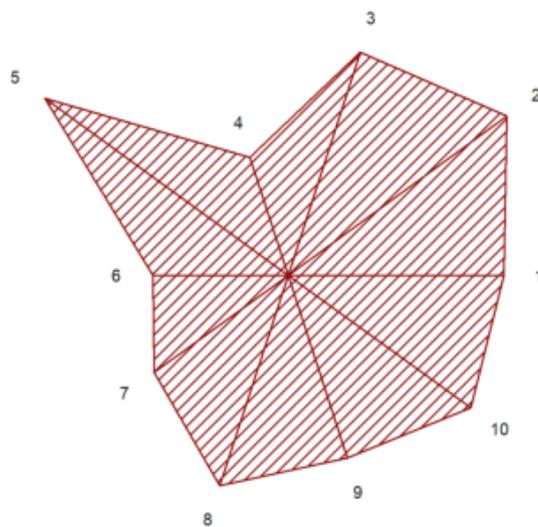
```
>function starplot1 (v, style="/", color=green, lab=none) ...
```

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

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

Kami memanggil reset sebelum kami menguji plot ini untuk mengembalikan default grafis. Ini tidak perlu, jika Anda yakin plot Anda berhasil.

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

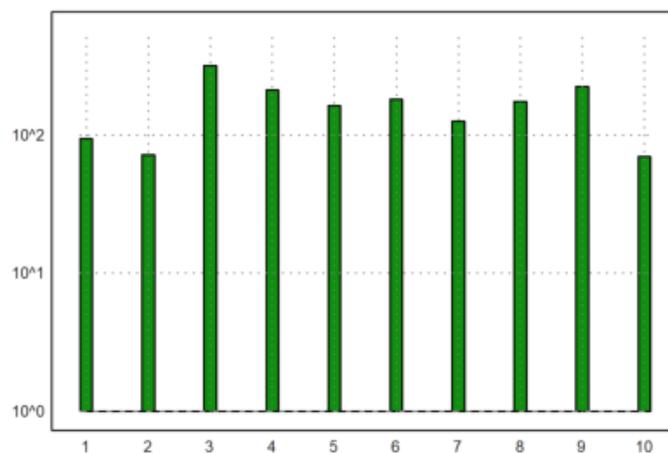


Terkadang, Anda mungkin ingin merencanakan sesuatu yang tidak dapat dilakukan plot2d, tetapi hampir. Dalam fungsi berikut, kami melakukan plot impuls logaritmik. plot2d dapat melakukan plot logaritmik, tetapi tidak untuk batang impuls.

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

Mari kita uji dengan nilai yang terdistribusi secara eksponensial.

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



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

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

```
>function animliss (n,m) ...  
  
    t=linspace(0,2pi,500);  
    f=0;  
    c=framecolor(0);
```

```

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

```

Tekan sembarang tombol untuk menghentikan animasi ini.

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

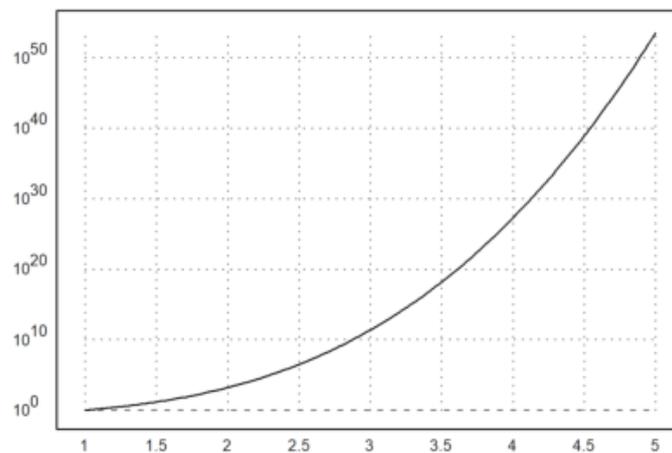
## Plot Logaritmik

EMT menggunakan parameter "logplot" untuk skala logaritmik.

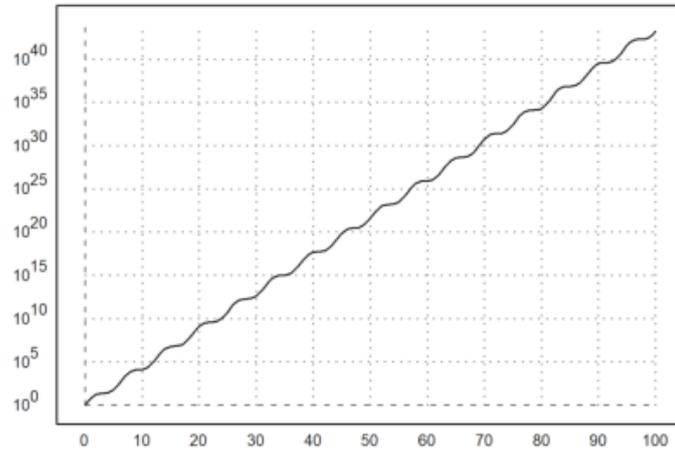
Plot logaritma dapat diplot baik menggunakan skala logaritma dalam y dengan logplot=1, atau menggunakan skala logaritma dalam x dan y dengan logplot=2, atau dalam x dengan logplot=3.

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

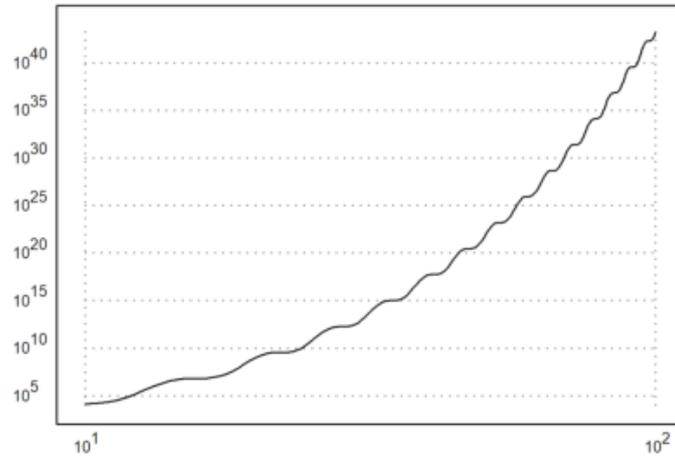
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



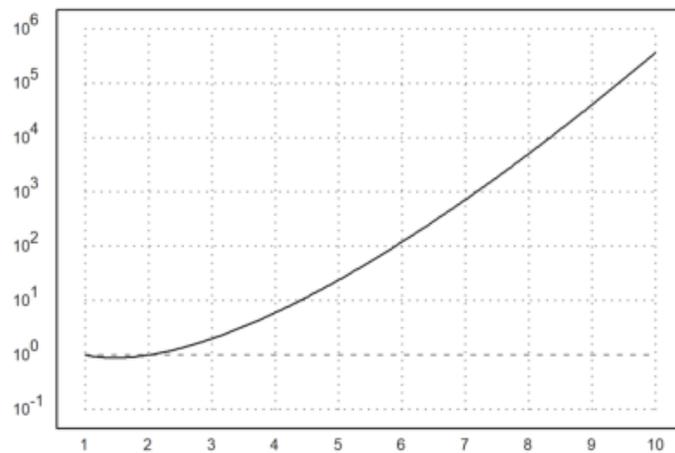
```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



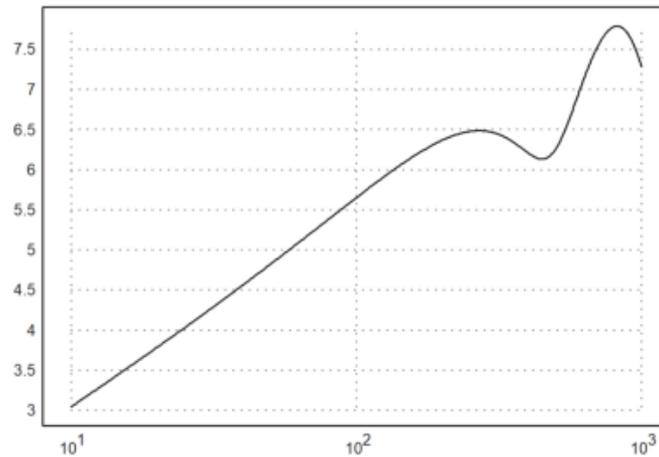
```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```



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

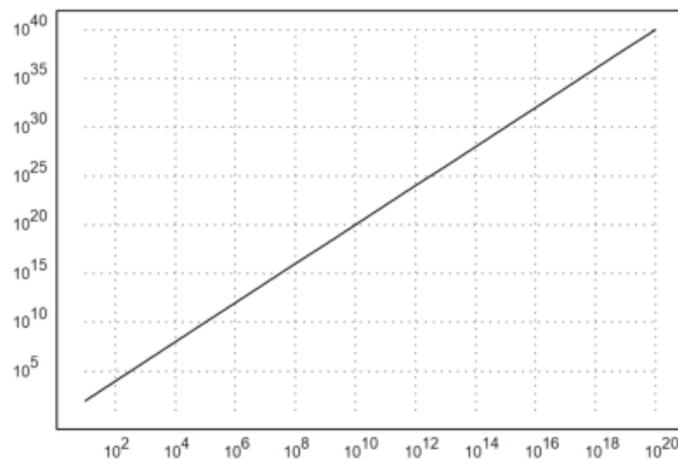


```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Ini juga berfungsi dengan plot data.

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



Langsung

Ekspresi adalah gabungan suku-suku yang digabungkan dengan menggunakan operasi matematika seperti pengurangan, penjumlahan, perkalian, dan pembagian. Istilah-istilah yang terlibat dalam ekspresi dalam matematika adalah:

- Konstanta: nilai numerik yang tetap.
- Variabel: simbol yang tidak memiliki nilai tetap.
- Suku: Suku dapat berupa konstanta tunggal, variabel tunggal, atau kombinasi variabel dan konstanta yang digabungkan dengan perkalian atau pembagian.
- Koefisien: angka yang dikalikan dengan variabel dalam suatu ekspresi.

**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, vertikal, lebih kecil, dl, niveau, level)
```

Fungsi plot serbaguna untuk plot dalam bidang (plot 2D). Fungsi ini dapat melakukan plot fungsi satu variabel, plot data, kurva dalam bidang, plot batang, kisi-kisi bilangan kompleks, dan plot implisit fungsi dua variabel.

Parameter

x, y : persamaan, fungsi, atau vektor data

a, b, c, d : Area plot (default a = -2, b = 2)

r : jika r ditetapkan, maka  $a = cx-r$ ,  $b = cx + r$ ,  $c = cy-r$ ,  $d = cy + r$

r dapat berupa vektor [rx,ry] atau vektor

[rx1,rx2,ry1,ry2].

xmin, xmax : rentang parameter untuk kurva

auto : Menentukan rentang y secara otomatis (default)

square : jika benar, cobalah untuk mempertahankan rentang x-y persegi

n : jumlah interval (standarnya adalah adaptif)

kisi-kisi: 0 = tidak ada kisi-kisi dan label,

1 = hanya sumbu,

2 = kisi-kisi normal (lihat di bawah ini untuk mengetahui

jumlah garis kisi-kisi)

3 = sumbu dalam

4 = tidak ada kisi-kisi

5 = kisi-kisi penuh termasuk margin

6 = tanda centang pada bingkai

7 = sumbu saja

8 = sumbu saja, sub-kutu

frame: 0 = tidak ada bingkai  
framecolor: warna dari frame dan grid  
margin : angka antara 0 dan 0,4 untuk margin di sekitar plot  
color : Warna kurva. Jika ini adalah sebuah vektor warna,

vektor ini akan digunakan untuk setiap baris pada matriks

plot. Dalam kasus

plot titik, ini harus berupa vektor kolom. Jika vektor

baris atau

matriks penuh warna digunakan untuk plot titik, itu akan

digunakan untuk

setiap titik data.

thickness : ketebalan garis untuk kurva

Nilai ini dapat lebih kecil dari 1 untuk garis yang sangat

tipis.

style : Gaya plot untuk garis, penanda, dan isian.

```
Untuk titik-titik, gunakan
"[]", "<>", \'.', \'.\'', \'....\'',
"*", \'+', \'|', \'-', "o"
"[#]", "<>#", 'o#' (bentuk terisi)
"[w]", "<>w", 'ow' (tidak transparan)
Untuk penggunaan garis
"--", "--", \'.', \'.\'', \'.-\'', \'.-\'', "-->"
Untuk poligon terisi atau plot batang, gunakan
"#", '#O', 'O', \'/', \\'', '\\\'',
"+", \'|', \'-', "t"
```

points : memplot titik-titik tunggal, bukan segmen garis

addpoints : jika benar, memplot segmen garis dan titik

add : menambahkan plot ke plot yang sudah ada

user : mengaktifkan interaksi pengguna untuk fungsi-fungsi

delta : ukuran langkah untuk interaksi pengguna

bar : plot batang (x adalah batas interval, y adalah nilai interval)

histogram : memplot frekuensi x dalam n subinterval

distribution=n : memplot distribusi x dengan n subinterval

genap : menggunakan nilai antar untuk histogram otomatis.

steps : memplot fungsi sebagai fungsi langkah (langkah = 1,2)

adaptive : menggunakan plot adaptif (n adalah jumlah langkah minimal)

level : memplot garis level dari fungsi implisit dari dua variabel

outline : menggambar batas rentang level.

Jika nilai level adalah matriks  $2 \times n$ , rentang level akan digambar dalam warna menggunakan gaya isian yang diberikan. Jika garis besar benar, itu akan digambar dalam warna kontur. Dengan menggunakan fitur ini, wilayah  $f(x,y)$  di antara batas-batas dapat ditandai.

hue : menambahkan warna rona ke plot level untuk menunjukkan fungsi

nilai

kontur: Gunakan plot level dengan level otomatis

nc : jumlah garis level otomatis

title : judul plot (default "")

xl, yl : label untuk sumbu x dan y

smaller : jika  $>0$ , akan ada lebih banyak ruang di sebelah kiri untuk label.

vertikal:

Mengaktifkan atau menonaktifkan label vertikal. Hal ini mengubah

variabel global

label vertikal secara lokal untuk satu plot. Nilai 1 hanya

menetapkan vertikal

vertikal, nilai 2 menggunakan label numerik vertikal pada sumbu y.

filled : mengisi plot kurva

fillcolor : warna isian untuk batang dan kurva yang terisi

outline : batas untuk poligon yang terisi

logplot : mengatur plot logaritmik

1 = logplot dalam y,  
2 = logplot dalam xy,  
3 = logplot dalam x

own :

Sebuah string, yang mengarah ke rutinitas plot sendiri. Dengan

>user, Anda mendapatkan

interaksi pengguna yang sama seperti pada plot2d. Jangkauan akan

diatur

sebelum setiap pemanggilan ke fungsi Anda.

maps : ekspresi pemetaan (0 lebih cepat), fungsi selalu dipetakan.  
contourcolor : warna garis kontur  
contourwidth : lebar garis kontur  
clipping : mengaktifkan kliping (defaultnya adalah true)  
title :

Ini dapat digunakan untuk mendeskripsikan plot. Judul akan muncul di

atas

plot. Selain itu, label untuk sumbu x dan y dapat ditambahkan dengan  
xl = "string" atau yl = "string". Label lain dapat ditambahkan

dengan fungsi

fungsi label() atau labelbox(). Judul dapat berupa sebuah unicode  
unicode atau sebuah gambar dari formula Latex.

cgrid :

Menentukan jumlah garis kisi untuk plot kisi-kisi yang kompleks.  
Harus berupa pembagi dari ukuran matriks dikurangi 1 (jumlah  
subinterval). cgrid dapat berupa vektor [cx, cy].

Ikhtisar

Fungsi ini dapat memplot

- ekspresi, memanggil koleksi atau fungsi dari satu variabel,
- kurva parametrik,
- data x terhadap data y,
- fungsi implisit,
- plot batang,
- kisi-kisi yang kompleks,
- poligon.

Jika sebuah fungsi atau ekspresi untuk xv diberikan, plot2d() akan menghitung  
nilai dalam rentang yang diberikan menggunakan fungsi atau ekspresi tersebut. Ekspresi  
haruslah sebuah ekspresi dalam variabel x. Kisarannya harus  
harus didefinisikan dalam parameter a dan b kecuali jika rentang default  
harus digunakan. Rentang y akan dihitung secara otomatis,  
kecuali jika c dan d ditentukan, atau radius r, yang menghasilkan rentang  
r,r

untuk x dan y. Untuk plot fungsi, plot2d akan menggunakan  
evaluasi adaptif dari fungsi secara default. Untuk mempercepat proses  
plot untuk fungsi yang rumit, matikan ini dengan <adaptive, dan  
secara opsional mengurangi jumlah interval n. Selain itu, plot2d()  
secara default akan menggunakan pemetaan. Yaitu, ia akan menghitung elemen plot  
untuk elemen. Jika ekspresi Anda atau fungsi-fungsi Anda dapat menangani sebuah  
vektor x, Anda bisa menonaktifkannya dengan <maps untuk evaluasi yang lebih cepat.

Perhatikan bahwa plot adaptif selalu dihitung elemen per elemen.  
Jika fungsi atau ekspresi untuk xv dan yv ditentukan,

plot2d() akan menghitung kurva dengan nilai xv sebagai koordinat x dan nilai yv sebagai koordinat y. Dalam kasus ini, sebuah rentang harus didefinisikan untuk parameter menggunakan xmin, xmax. Ekspresi yang terkandung dalam string harus selalu berupa ekspresi dalam variabel parameter x.

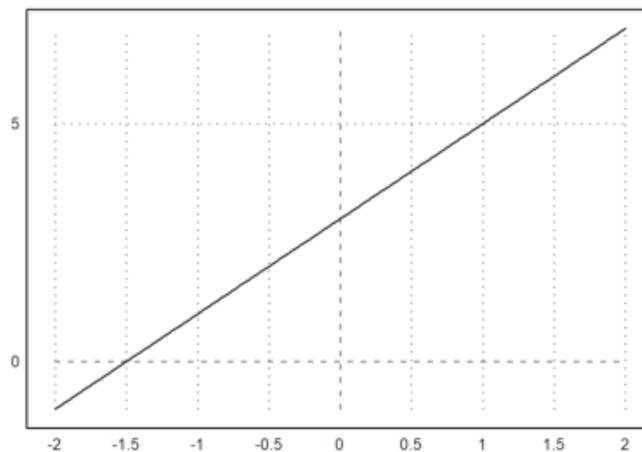
## Uraian Materi (Teori)

Ekspresi linier adalah ekspresi aljabar yang mengandung konstanta dan variabel yang dipangkatkan 1. Misalnya:

$$2x + 3$$

Gunakan plot2d pada baris perintah untuk membuat grafik  $2x+3$ .

```
>plot2d("2*x+3",-2,2): // plot x dari -2 sampai 2
```



Untuk mencari titik potong dengan sumbu x dan sumbu y, gunakan  $x=0$  untuk mencari titik perpotongan dengan sumbu y dan gunakan  $y=0$  untuk mencari perpotongan sumbu x.

$$2x + 3 = 2(0) + 3 = 3$$

untuk  $y=0$

$$2x + 3 = 0$$

$$2x = -3$$

$$x = \frac{-3}{2}$$

## Ekspresi Kuadrat

$$ax^2 + bx + c$$

nilai a tidak boleh sama dengan nol karena saat  $a=0$ , maka ini akan menghasilkan ekspresi linier.

Contoh dari ekspresi kuadrat adalah:

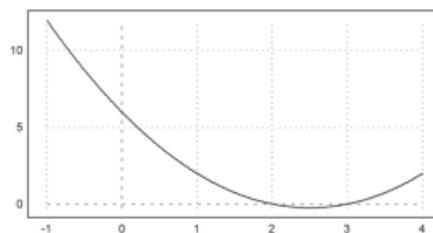
$$x^2 - 5x + 6$$

Untuk membuat grafik dengan ekspresi tersebut dapat diperoleh dengan merepresentasikan sebagai fungsi kuadrat atau:

$$y = ax^2 + bx + c$$

untuk membuat grafik tersebut di EMT, gunakan "plot 2d(...)"

```
>aspect(6,3); plot2d("x^2-5x+6",-1,4); insimg(10); //
```



mencari titik potong grafik tersebut pada sumbu x dan sumbu y, substitusikan  $x=0$  untuk mencari titik potong antara sumbu y dan substitusikan  $y=0$  untuk mencari titik potong antara sumbu x.

Untuk  $x=0$

$$x^2 - 5x + 6 = y$$

$$0^2 - 5(0) + 6 = y$$

$$y = 6$$

untuk  $y=0$

$$x^2 - 5x + 6 = 0$$

$$(x - 3)(x - 2) = 0$$

$$x_1 = 3; x_2 = 2$$

Untuk mencari titik ekstrim pada sumbu y, kita perlu mencari nilai determinan dari fungsi kuadrat tersebut dengan menggunakan EMT.

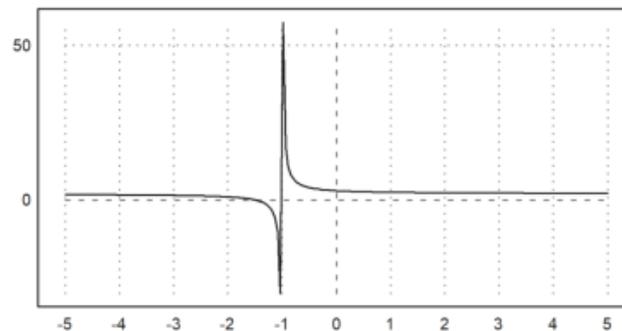
```
>a=1; b=-5; c=6; d=b^2-4*a*c //
```

1

```
>$ (2*x+3)/(x+1)
```

$$\frac{2x+3}{x+1}$$

```
>x1 = linspace(-5,-0.1, 100);  
>x2 = linspace(0.1, 5, 100);  
>x = [x1, x2];  
>y = (2*x+3)/(x+1);  
>plot2d(x,y):
```



Ekspresi rasional menunjukkan rasio dua polinomial, artinya pembilang dan penyebutnya adalah polinomial. Sama seperti pecahan, ini juga merupakan rasio ekspresi aljabar, yang terdiri dari variabel yang tidak diketahui. Contoh dari ekspresi rasional yaitu:

$$\frac{2x+3}{x+1}$$

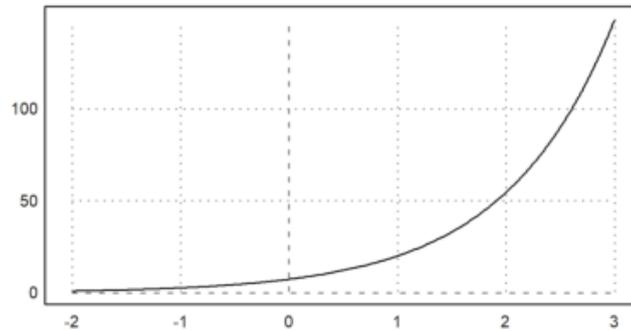
Untuk menggambar grafik fungsi logaritma, gunakan "plot2d(log10(expr))" atau "plot2d(logbase(x,base))".

Untuk menggambar ekspresi eksponensial, gunakan plot2d(exp) pada baris perintah.

```
>$ exp(x+2)
```

$$e^{x+2}$$

```
>plot2d("exp(x+2)", -2, 3) :
```



Beberapa jenis fungsi matematika yang termasuk contoh dari grafik fungsi satu variabel, di antaranya :

1. Fungsi Linear

Grafik yang dihasilkan berupa garis lurus. Contoh fungsi :

$$y = 4x + 7$$

2. Fungsi Kuadrat

Grafik yang dihasilkan berbentuk parabola. Contoh fungsi :

$$y = x^2 + 4$$

3. Fungsi Trigonometri

Grafik yang dihasilkan berupa gelombang. Contoh fungsi :

$$y = 2\cos x$$

4. Fungsi Eksponensial. Contoh fungsi :

$$y = 3^x$$

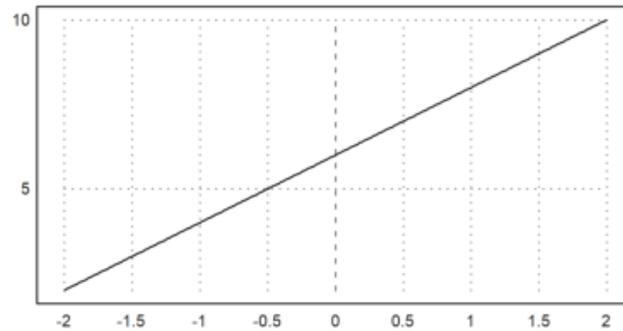
5. Fungsi Logaritmik. Contoh fungsi :

$$y = \log(x)$$

---

**contoh soal**

```
>function a(x) := 2x+6
>plot2d("a"):
```



Langkah pertama yang dilakukan untuk menyimpan suatu fungsi yaitu menentukan variabel yang akan digunakan (variabel hanya dapat digunakan satu kali). Pada contoh di atas, fungsi disimpan dalam variabel a. Memotong sumbu x(y=0)

$$0 = 2x + 6$$

$$x = \frac{6}{2}$$

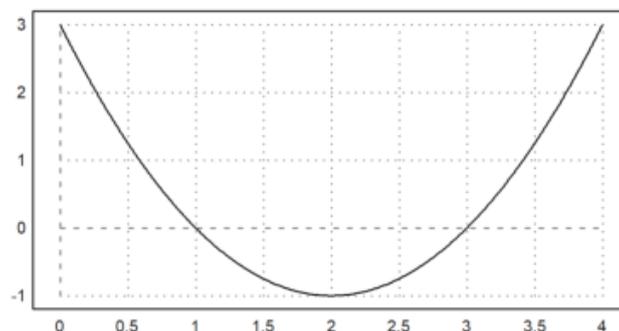
$$x = 3$$

Memotong sumbu y(x=0)

$$y = 2(0) + 6$$

$$y = 6$$

```
>function b(x) := x^2-4x+3
>aspect(2); plot2d("b",0,4):
```



Dalam euler math toolbox cara mendefinisikan fungsi menggunakan sintak function. untuk mendefinisikan fungsi numerik menggunakan tanda ":="

Fungsi numerik menjelaskan cara bilangan dalam domain berhubungan dengan bilangan sebagai kodomain, biasanya diberikan dalam bentuk rumus matematik(persamaan) atau aturan yang memetakan setiap domain kedalam kodomain yang sesuai. contoh:

$$f(x) = 2x + 1$$

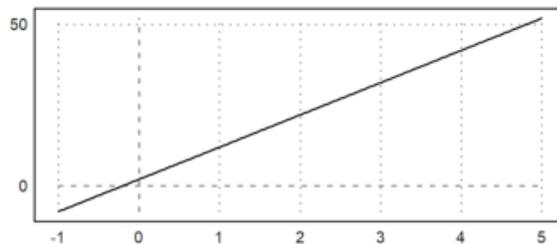
(x)(variabel dependen) adalah fungsi yang memetakan setiap nilai x(variabel independen)kedalam nilai  $2x+1$ . Terdapat berbagai jenis fungsi yang termasuk ke dalam fungsi numerik, diantaranya:

. Fungsi linier dengan bentuk umum

Bentuk grafiknya adalah garis

$$f(x) = ax + b$$

```
>reset;  
>function f(x) := 10x+2  
>aspect(2.5), plot2d("f(x)",-1,5):
```



titik potong:

$$f(x) = 10x + 2$$

$$f(x) = 10 * 0 + 2$$

$$f(x) = 2$$

$$0 = 10x + 2$$

$$x = -\frac{1}{5}$$

CONTOH 2

$$f(x) = \frac{3x - 4}{x + 2}$$

. Fungsi kuadrat dengan bentuk umum

Bentuk grafik adalah parabola

$$f(x) = ax^2 + bx + c$$

. Fungsi eksponensial dengan bentuk umum

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

. Fungsi logaritma dengan bentuk umum

$$f(x) = \log_a(x)$$

. Fungsi trigonometri dengan bentuk umum

$$f(x) = \sin(x), f(x) = \cos(x)$$

dan lain lain.

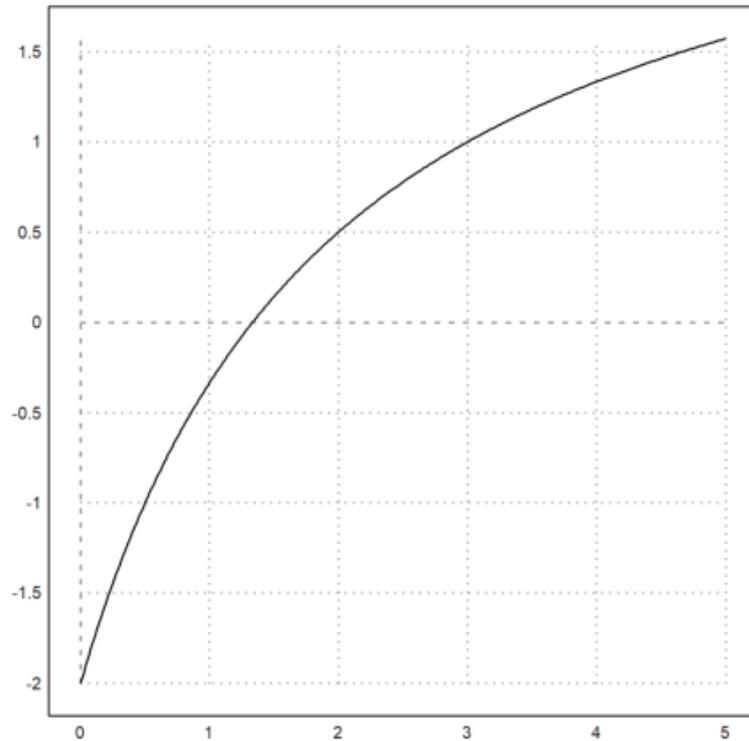
**Contoh soal**

---

CONTOH 1

$$10x + 2$$

```
> reset;
>function f(x)=(3*x-4)/(x+2) //mendefinisikan fungsi
>plot2d("f(x)",0,5): //menggambar fungsi ke bidang kartesius xy
```



Mengambar fungsi ke bidang kartesius xy dengan memberikan ukuran atau rasio plot yaitu 2.5 atau setara dengan x:y (5:2).

titik potong sumbu y

$$\frac{3x - 4}{x + 2}$$

$$\frac{3 * 0 - 4}{0 + 2}$$

$$\frac{-4}{2}$$

$$-2$$

titik potong sumbu x

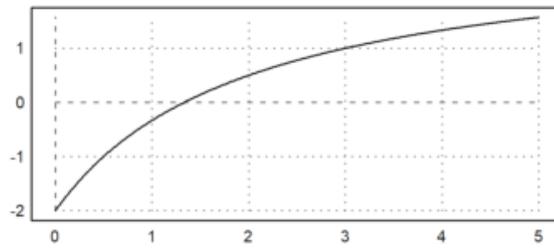
$$\frac{3x - 4}{x + 2} = 0$$

$$3x - 4 = 0$$

$$3x = 4$$

$$x = \frac{4}{3}$$

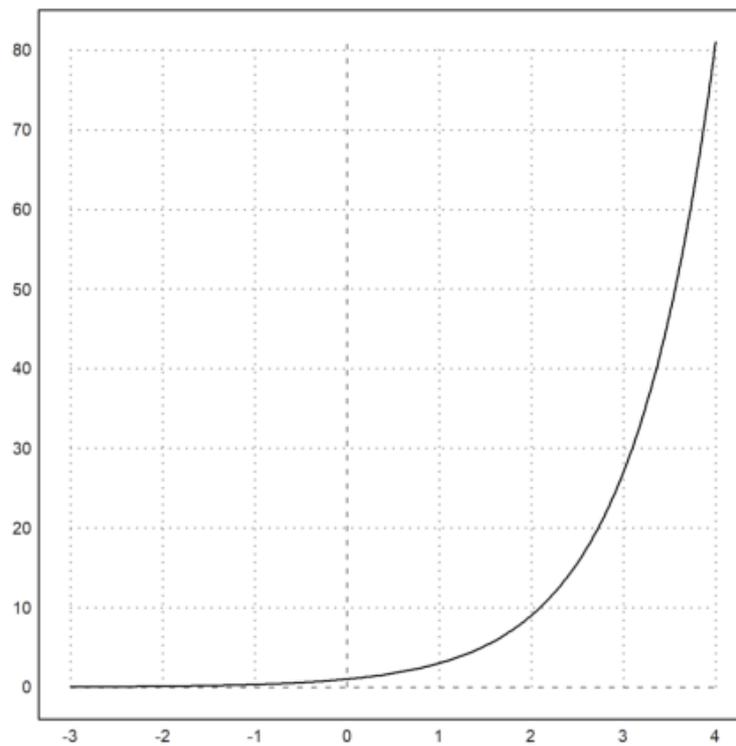
```
>aspect(2.5), plot2d("f(x)",0,5):
```



CONTOH 3

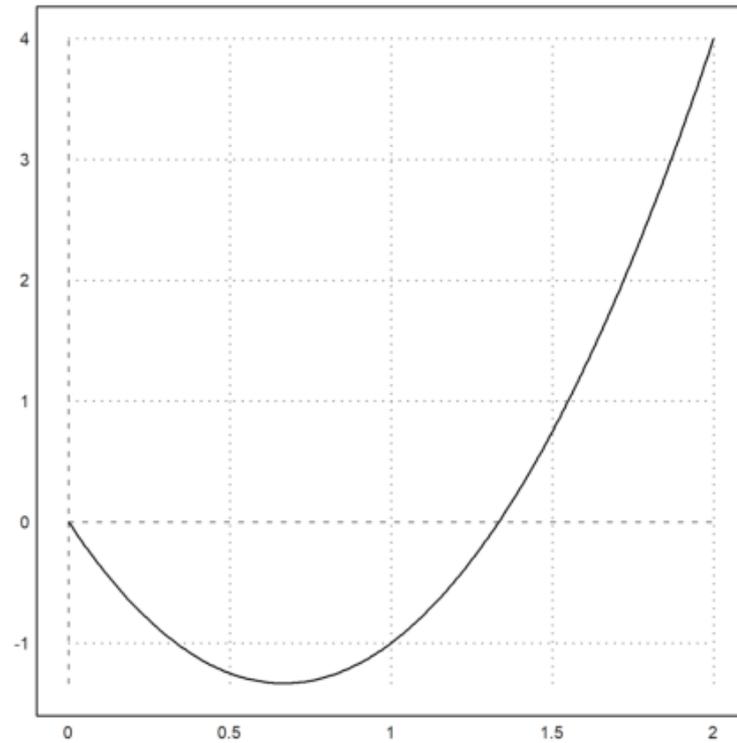
$$f(x) = 3^x$$

```
>reset; //Mereset plot sehingga plot yang ditampilkan hanya fungsi selanjutnya
>function f(x) := 3^x
>plot2d("f(x)", -3, 4):
```



Untuk membuat plot fungsi turunan, kita dapat menambahkan `&diff()` beserta fungsi yang akan diturunkan setelah perintah `plot2d()`.

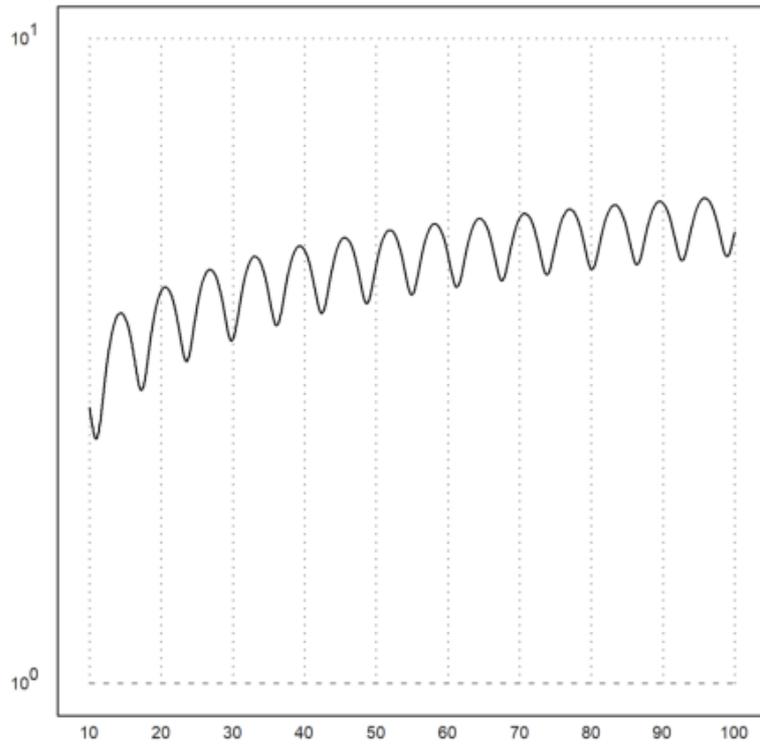
```
>reset();
>function g(x) &=x^3-2*x^2;
>plot2d(&diff(g(x), x), 0, 2):
```



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

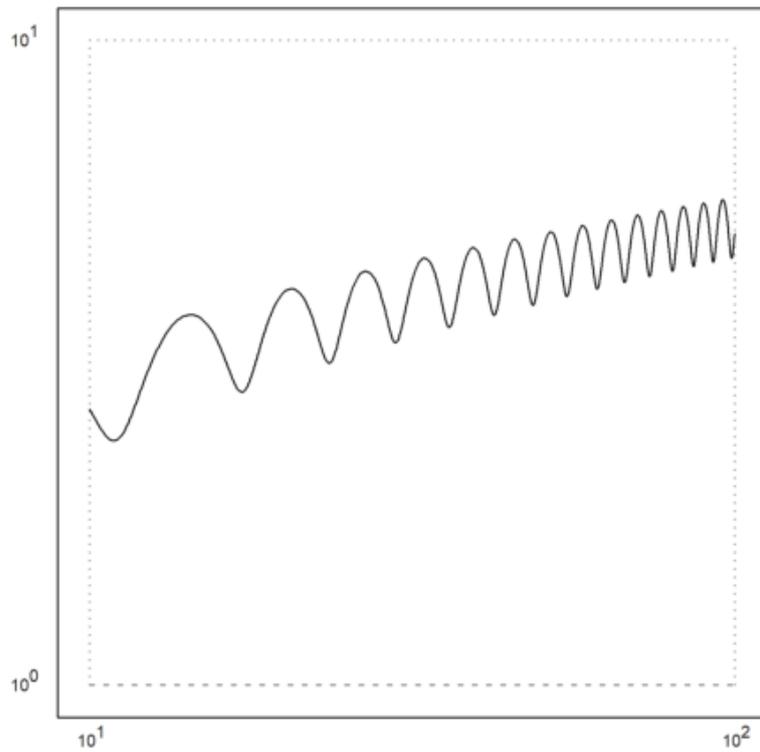
- logplot=1: y-logarithmic
- logplot=2: x-y-logarithmic
- logplot=3: x-logarithmic

```
>reset();
>function f(x) &= log(x*(2+sin(x)));
>plot2d(&f(x),10,100,logplot=1):
```



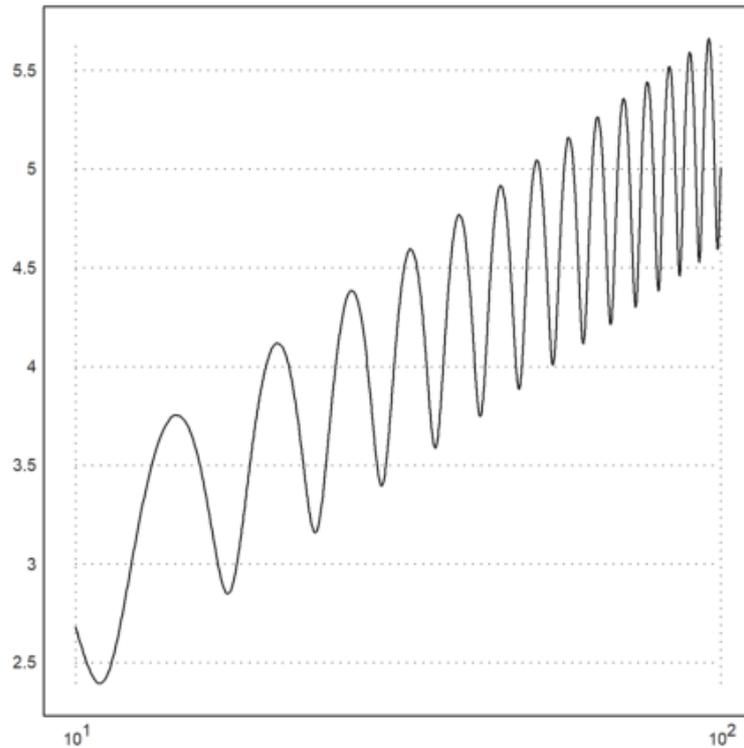
Untuk logplot=1, rentang sumbu y dituliskan dalam bentuk 10 pangkat n.

```
>plot2d(&f(x),10,100,logplot=2) :
```



Untuk logplot=2, rentang sumbu x dan y dituliskan dalam bentuk 10 pangkat n.

```
>plot2d(&f(x),10,100,logplot=3):
```



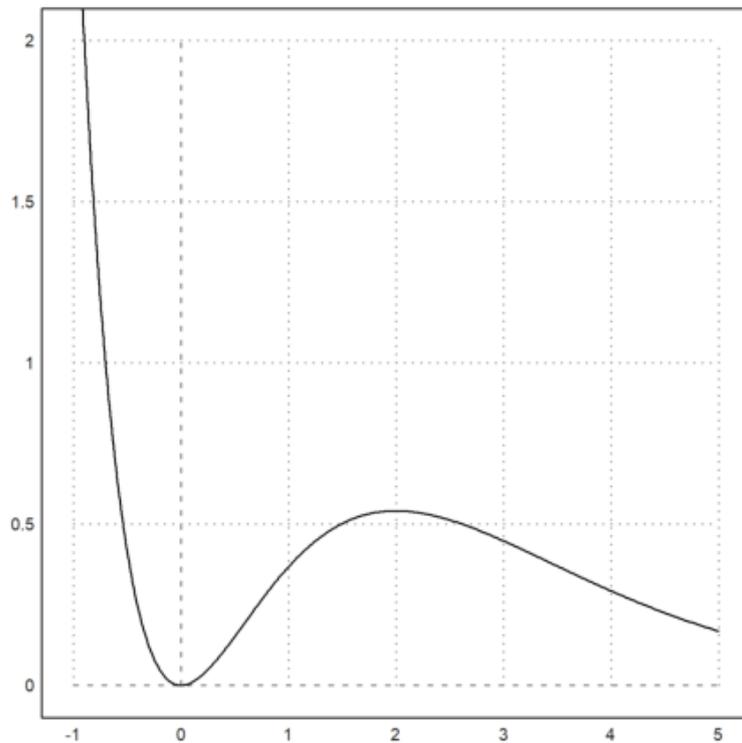
Sementara logplot=3, rentang sumbu x dituliskan dalam bentuk 10 pangkat n.

Fungsi eksponensial memiliki bentuk umum

$$e^x$$

Dalam baris perintah EMT, kita dapat menuliskannya menjadi "exp(x)"

```
>reset();  
>function g(x) &= x^2*exp(-x);  
>plot2d(&g(x),a=-1,b=5,c=0,d=2):
```



## Contoh Soal

1. Gambarlah grafik fungsi  $f(x)$  berikut dengan  $-20 < x < 20$

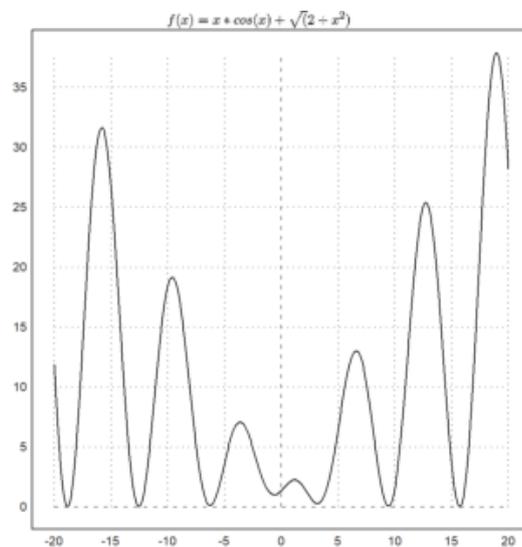
$$f(x) = x \cos(x) + \sqrt{2 + x^2}$$

Penyelesaian:

```
>reset(); // mereset ke format default grafik
>function f(x) &= x*cos(x) + sqrt(2+x^2) // mendefinisikan fungsi simbolik
```

$$x \cos(x) + \sqrt{x^2 + 2}$$

```
>plot2d(&f(x), -20, 20, title=latex("f(x) = x*cos(x) + \sqrt{2+x^2}")); insimg(20)
```



Terbentuk grafik fungsi  $f(x)$  dengan rentang  $x$  dari -20 sampai 20. Fungsi `insimg(20)` untuk mengatur besar kecilnya plot yang ditampilkan. Jika tidak menggunakan `insimg`, otomatis tampilan plot menggunakan default `insimg(30)` yang lebih besar dari tampilan plot di atas.

2. Gambarkan grafik fungsi  $h(x)$  berikut beserta turunannya

$$h(x) = 2x + x^3$$

Penyelesaian:

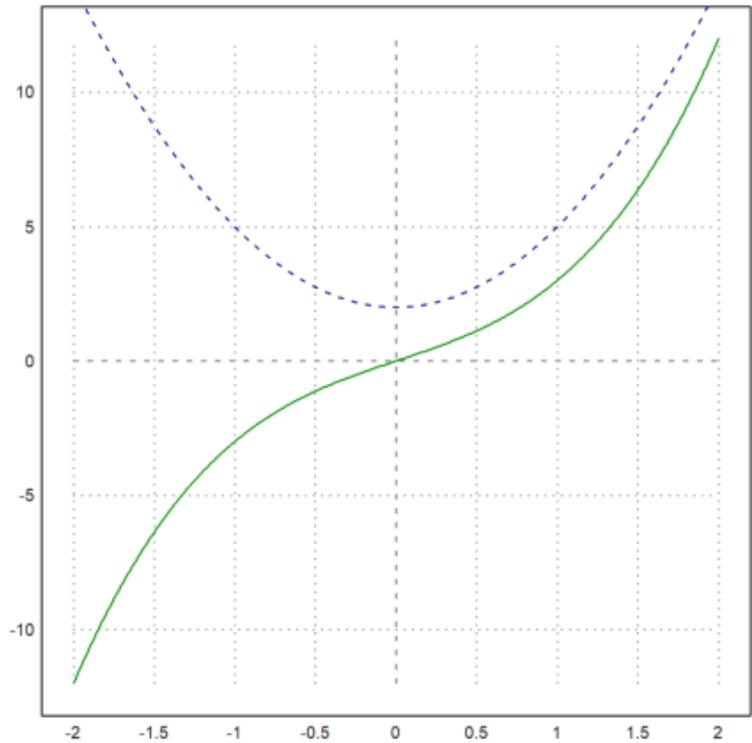
```
>reset(); //mereset format default grafik
>function h(x)&= 2*x+x^3 // mendefinisikan fungsi h(x)
```

$$x^3 + 2x$$

```
>&h(x) // memanggil fungsi h(x)
```

$$x^3 + 2x$$

```
>plot2d(&h(x),color=green); // grafik fungsi h(x)
>plot2d(&diff(h(x),x),>add,color=blue,style="--"): // menambahkan grafik fungsi
```



3. Gambarlah kedua fungsi berikut dan tentukan titik potong kedua grafik tersebut.

$$f(x) = \frac{x^3 - 9x^2 + 2}{3}, 0 \leq x \leq 10$$

$$g(x) = \frac{5}{3}x + \frac{2}{3}, 0 \leq x \leq 10$$

Penyelesaian:

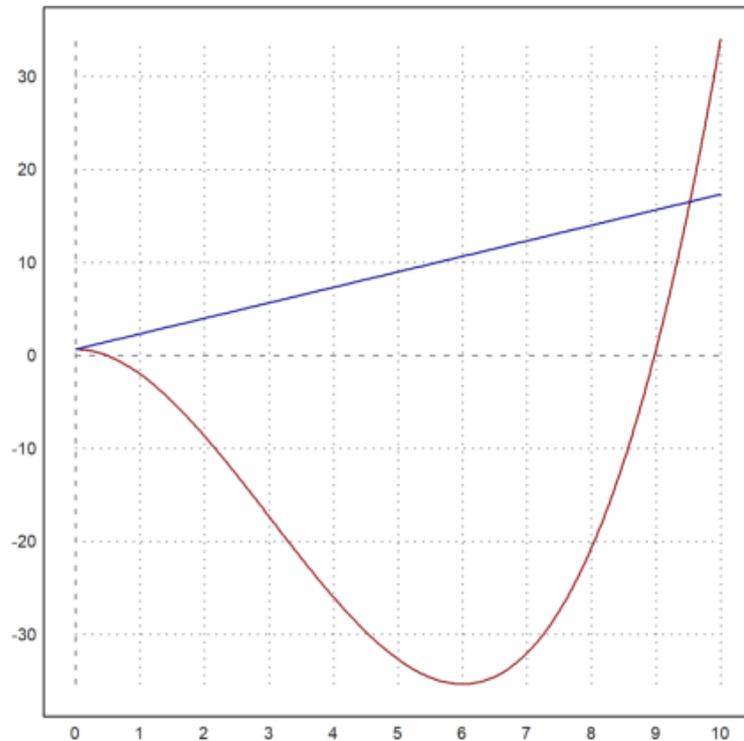
```
>reset();
>function f(x) &= (x^3-9*x^2+2)/3 // mendefinisikan fungsi f(x);
```

$$\frac{x^3 - 9x^2 + 2}{3}$$

```
>function g(x) &= 5/3*x+2/3 // mendefinisikan fungsi g(x)
```

$$\frac{5x}{3} + \frac{2}{3}$$

```
>plot2d([&f(x), &g(x)], 0, 10, color=[red, blue]) :
```



Dari grafik di atas, kita dapat membuat grafik fungsi  $f(x)$  dan  $g(x)$  sekaligus dalam satu plot dengan menggunakan kurung siku `[]` setelah perintah `plot2d` dan kedua fungsi dipisahkan oleh koma. Begitu juga dengan pewarnaan kedua grafik, kita dapat melakukannya demikian.

Grafik di atas terlihat ada perpotongan di titik  $x=0$  dan di sekitar titik  $x=9.5$ . Untuk lebih tepatnya, kita dapat menentukan titik perpotongan tersebut.

- Mencari titik potong pertama

```
>x1 = 0; g(x1), f(x1)
```

```
0.6666666666667
```

```
0.6666666666667
```

Sehingga titik potong pertama di titik  $(0, 0.667)$ .

- Mencari titik potong yang kedua

```
>x2 = solve("g(x)-f(x)", 10)
```

```
9.52493781056
```

```
>(g(x2))
```

```
16.5415630176
```

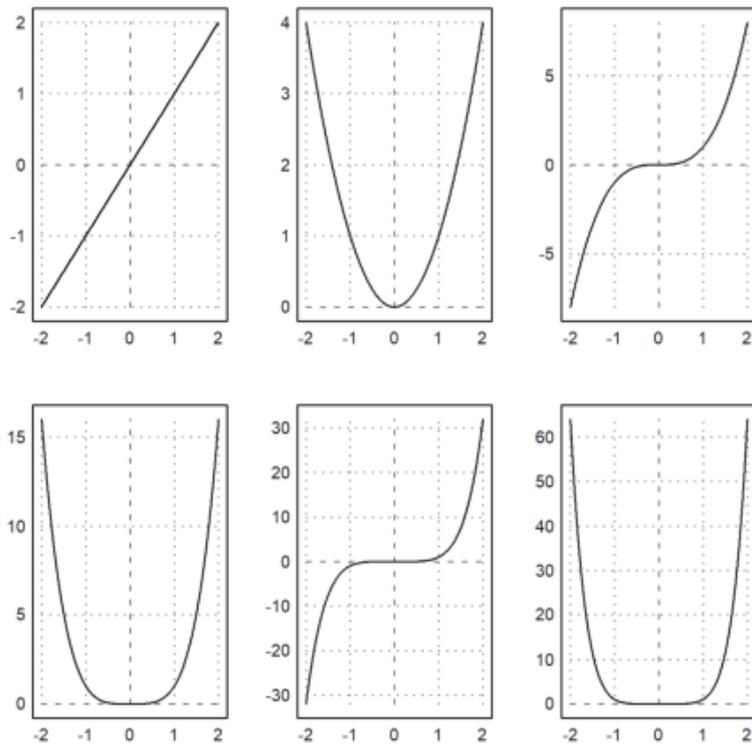
```
>(f(x2))
```

16.5415630176

Gambarkan plot fungsi berikut:

$$x^n, 1 \leq n \leq 6$$

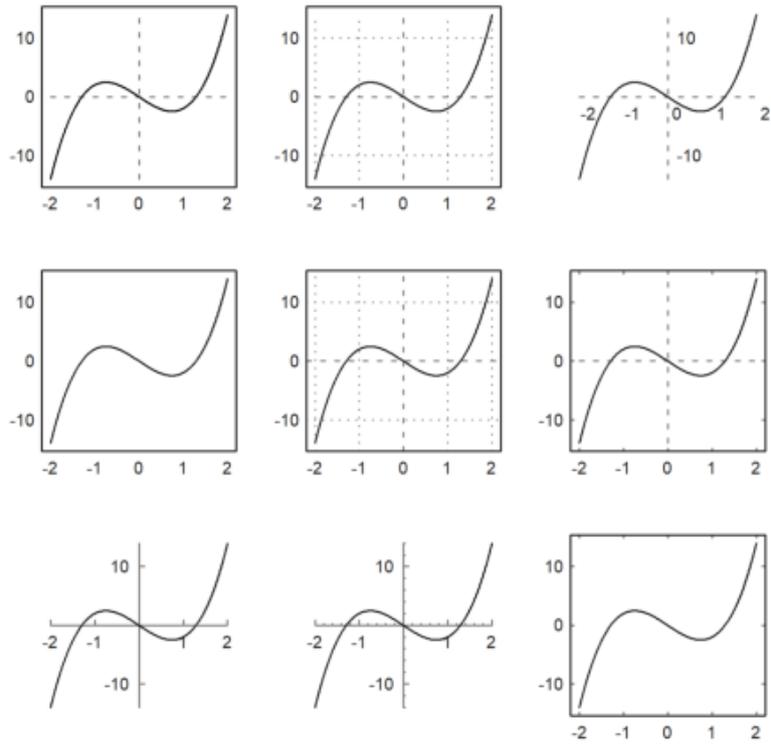
```
>reset; ...  
>figure(2,3); ...  
>for n=1 to 6; figure(n); plot2d("x^"+n); end; ...  
>figure(0):
```



Gambarkan plot fungsi berikut:

$$3x^3 - 5x, -2 \leq x \leq 2$$

```
>reset; ...  
>figure(3,3); ...  
>for k=1 to 9; figure(k); plot2d("3x^3-5x",-2,2,grid=k); end; ...  
>figure(0):
```



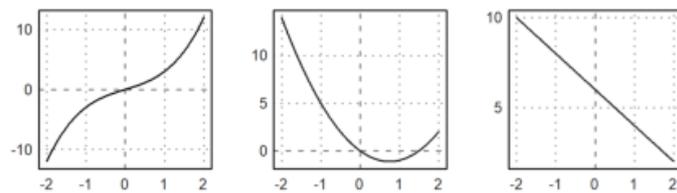
Gambarkan fungsi plot berikut ini:

$$x^3 + 2x$$

$$2x^2 - 3x$$

$$-2x + 6$$

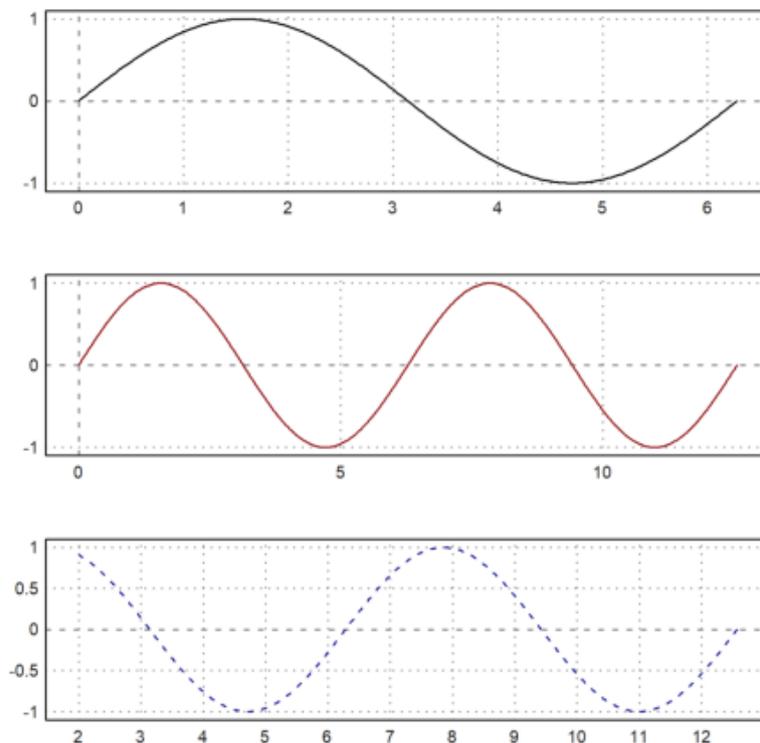
```
>reset; ...
>aspect(3,1); ...
>figure(1,3); ...
>figure(1); plot2d("x^3+2x", grid=5); ...
>figure(2); plot2d("2x^2-3x", grid=5); ...
>figure(3); plot2d("-2x+6", grid=5); ...
>figure(0):
```



Gambarkan fungsi plot berikut ini:

$$\sin x$$

```
>reset; ...  
>figure(3,1); ...  
>figure(1); plot2d("sin(x)",0,2pi, grid=5); ...  
>figure(2); plot2d("sin(x)",0,4pi,grid=5, color=red); ...  
>figure(3); plot2d("sin(x)",2,4pi,grid=5, color=blue, style="--"); ...  
>figure(0):
```

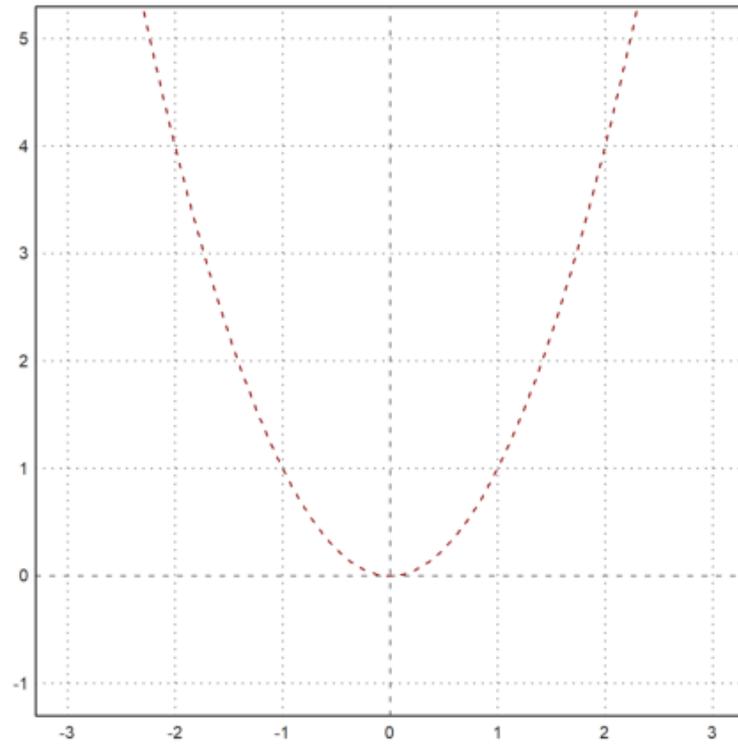


Buat dalam satu bidang fungsi

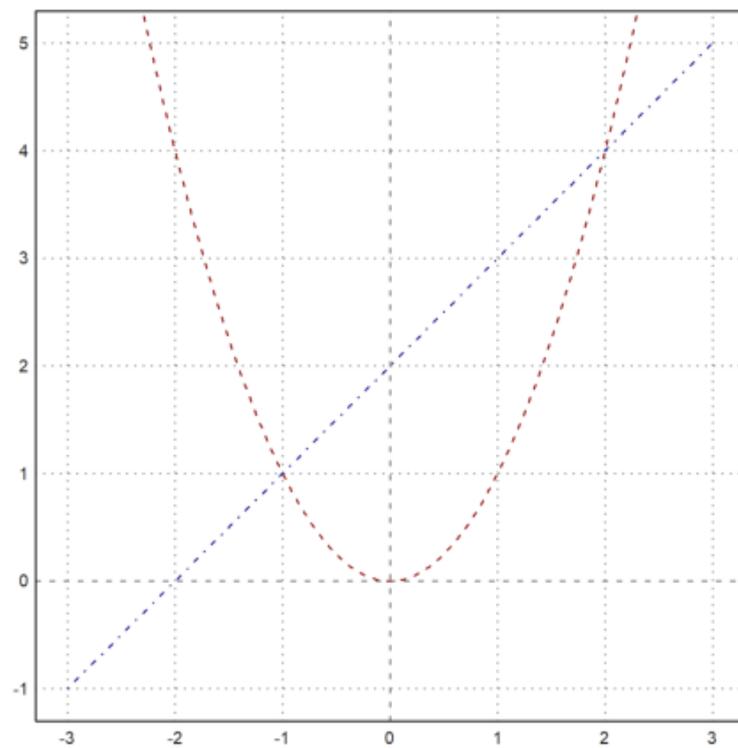
$$x^2$$

$$x + 2$$

```
>reset; ...  
>plot2d("x^2",a=-3,b=3,c=-1,d=5,grid=5,color=red,style="--"); ...  
>insimg(30);
```



```
>plot2d("x+2", >add, color=blue, style="-.") :
```



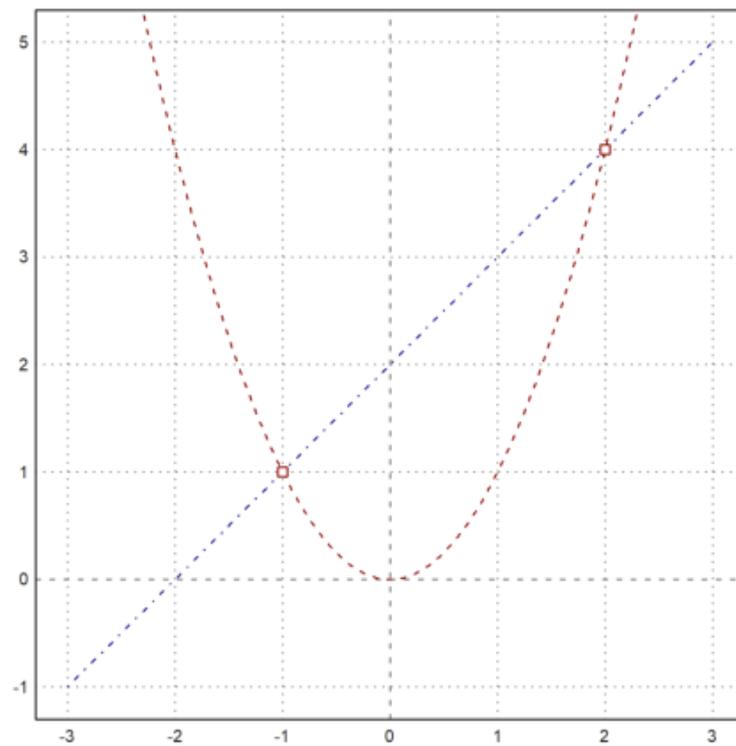
Mencari titik potong:

```
>$ solve(x^2=x+2, x)
```

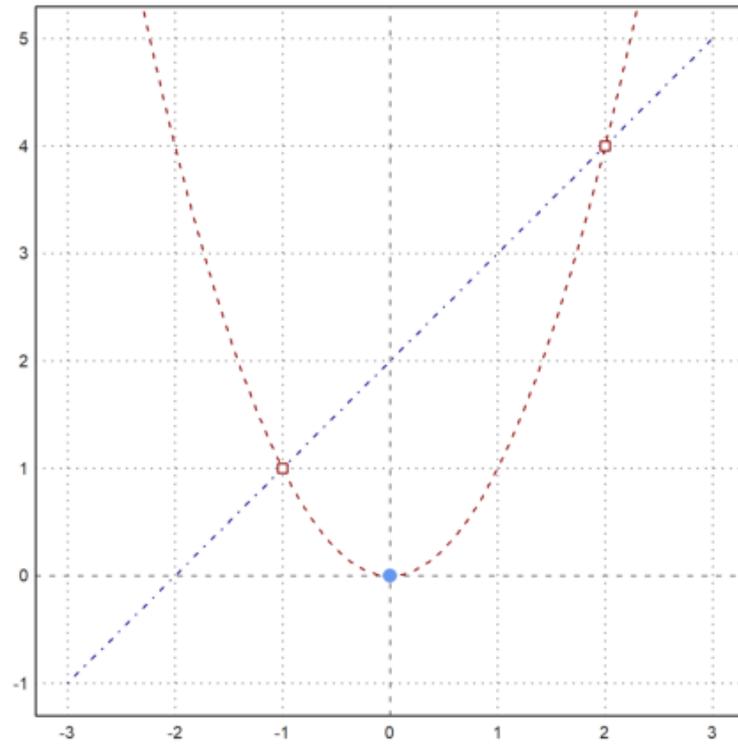
$[x = 2, x = -1]$

Menambahkan titik potong ke grafik

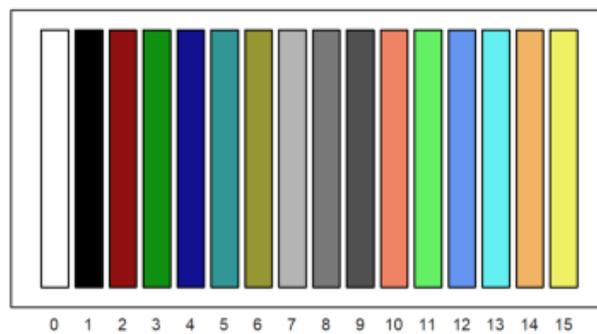
```
>function f(x):=x+2; ...  
>plot2d((2),f(2),>points,>add,color=red); ...  
>plot2d((-1),f(-1),>points,>add,color=red):
```



```
>a=1; b=0; c=0; D=b^2-4*a*c; // ax^2+bx+c untuk x^2 ...  
>plot2d((-b/2*a), (-D/4*a),>points,style="o#",>add,color=12):
```

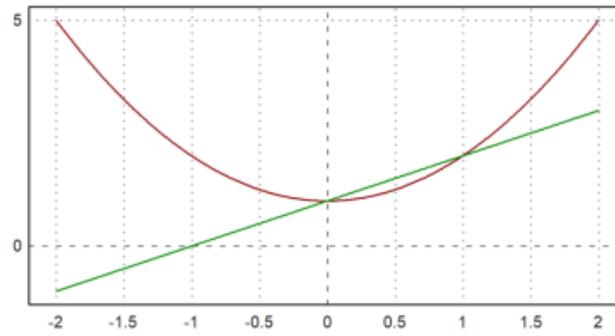


```
>aspect(2); columnsplot(ones(1,16),lab=0:15,grid=0,color=0:15):
```



Ada yang lebih simple daripada >add dan add=true yaitu dengan ditambahkan kedalam kurung siku seperti berikut.

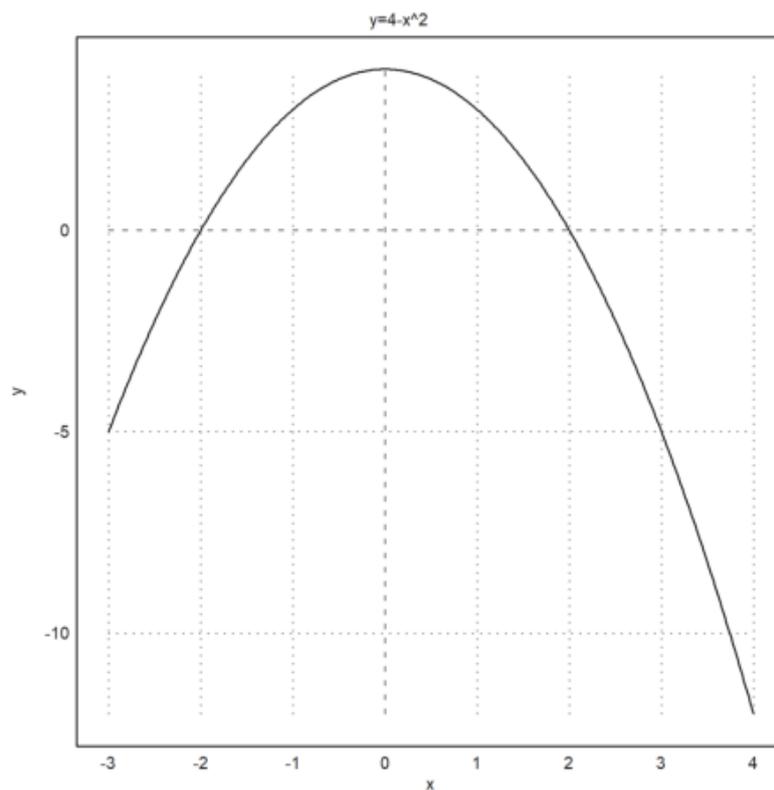
```
>plot2d(["x^2+1", "x+1"],color=2:3,grid=5):
```



Gambarkan kurva dengan fungsi:

$$y = 4 - x^2$$

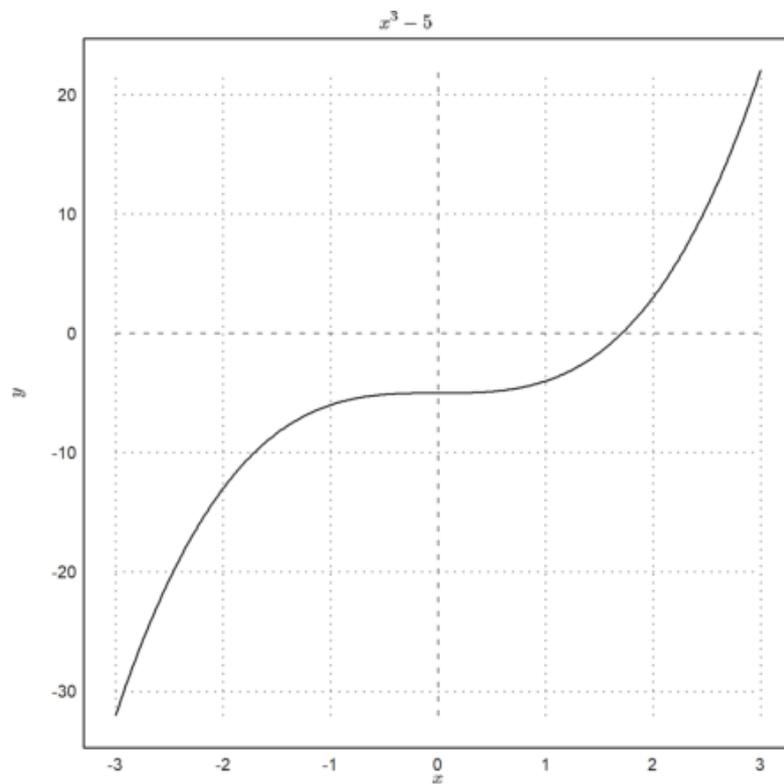
```
>aspect(1); ...
>plot2d("4-x^2", -3, 4, title="y=4-x^2", xl="x", yl="y") :
```



Gambarkan plot fungsi

$$y = x^3 - 5$$

```
>plot2d("x^3-5", -3, 3, title=latex("x^3-5"), xl=latex("x"), yl=latex("y")) :
```



Gambarkan plot fungsi

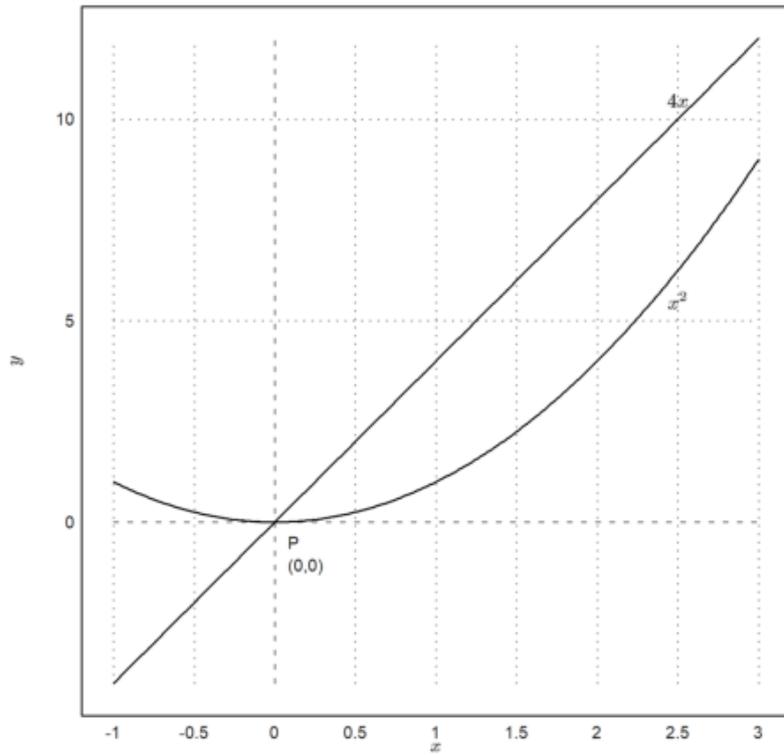
$$y = 4x$$

dan

$$y = x^2$$

dengan interval [-1,3].

```
>reset; ...
>plot2d(["4x", "x^2"], -1, 3, xl=latex("x"), yl=latex("y")); ...
>label(["P", "(0,0)"], 0, 0); ...
>label(latex("4x"), 2.5, 10, pos="uc"); ...
>label(latex("x^2"), 2.5, 6, pos="lc");
```



Gambarkan plot fungsi

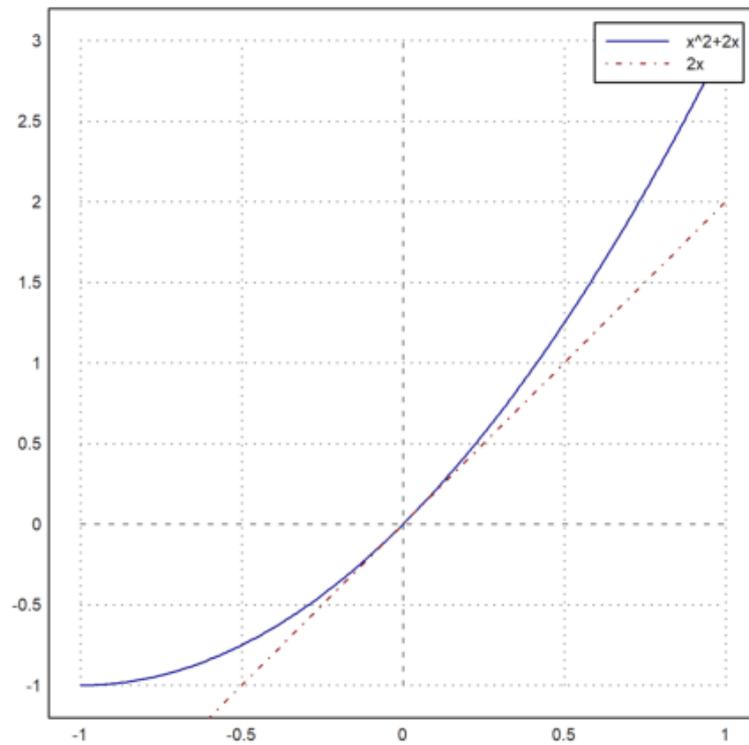
$$f(x) = x^2 + 2x$$

dan

$$g(x) = 2x$$

lalu diberikan keterangan kurvanya (legend)

```
>plot2d("x^2+2*x",-1,1,color=blue); ...
>plot2d("2*x",color=red,style="-.",>add); ...
>labelbox(["x^2+2x","2x"],colors=[blue,red],styles=["-","-."]):
```



Gambarkan plot fungsi

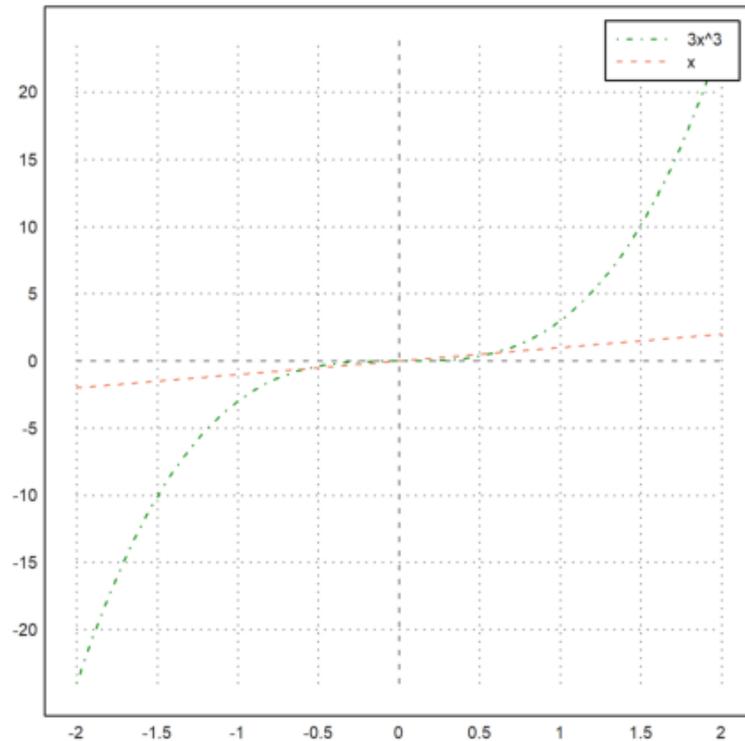
$$f(x) = 3x^2$$

dan

$$g(x) = x$$

lalu berikan keterangan kurjanya

```
>plot2d(["3*x^3","x"], color=[green, orange], style=[".-","--"]); ...
>labelbox(["3x^3","x"], colors=[green, orange], styles=[".-","--"]):
```



### 1. Mengatur Ukuran Gambar

Mengatur ukuran gambar dapat diatur dengan menyesuaikan aspek rasio yang sesuai dengan data yang disajikan.

### 2. Mengatur Format (Style)

Format (Style) digunakan untuk menentukan bentuk seperti apa yang akan muncul untuk garis kurva. Ada beberapa pilihan antara lain sebagai berikut.

1. "-"
2. "\_"
3. "-."
4. "."
5. "-."
6. "-.-"

Untuk menggunakan style tersebut dapat menggunakan perintah `style="..."`

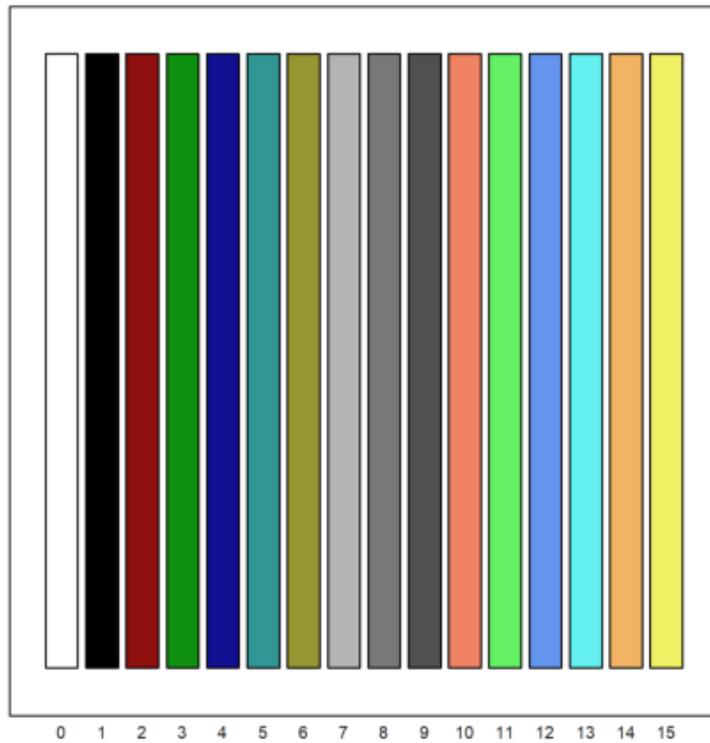
### 3. Mengatur Warna Kurva

EMT memiliki indeks warna dengan skala 0-15.

Konstanta warna : putih, hitam, merah, hijau, biru, cyan, zaitun, abu-abu muda, abu-abu tua, oranye, hijau muda, pirus, biru muda, oranye terang, kuning.

Berikut warna konstan yang ada di EMT

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



Contoh:

1. Buatlah kurva dengan fungsi:

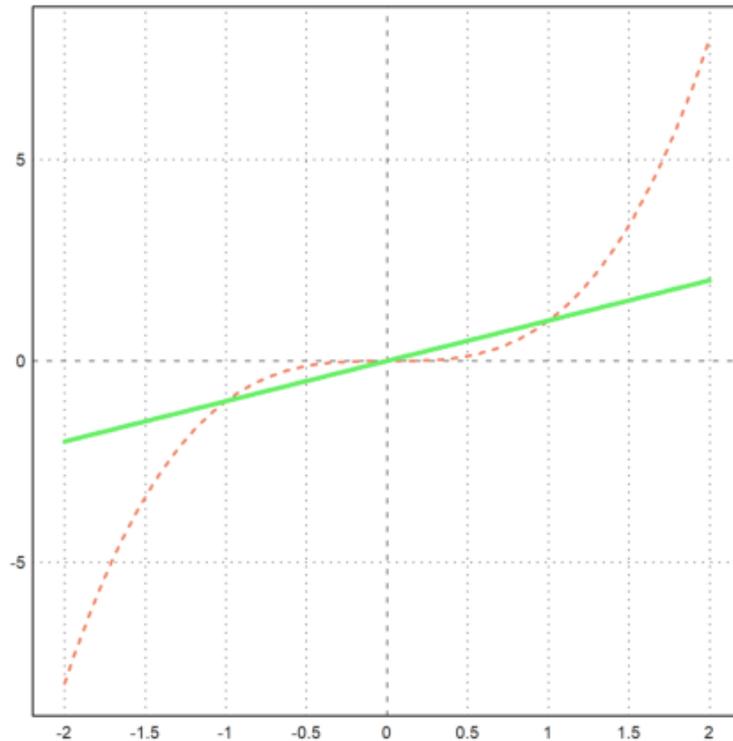
$$y = x^3$$

dan

$$g = x$$

Dengan ukuran aspect (1), style "--" dan "-", dan warna kurva 10:11!

```
>reset;...
>aspect(1);...
>plot2d(["x^3","x"], style=["--","-"], color=10:11, grid=5, thickness=[2:3]):
```



1. reset;

-digunakan sebagai awalan

-untuk menghapus grafik yang telah ada sebelumnya.

2. aspect(1)

-digunakan untuk mengatur rasio aspek dari jendela grafik

-aspect(1) akan menghasilkan plot dengan perbandingan panjang sumbu x dan y berturut-turut 1:1

3. plot2d(["x^3","x"], style=["-","-"], color=10:11, grid=5, thickness=[2:3]):

-digunakan untuk menggabungkan plot fungsi  $y=x^3$  dan  $g=x$

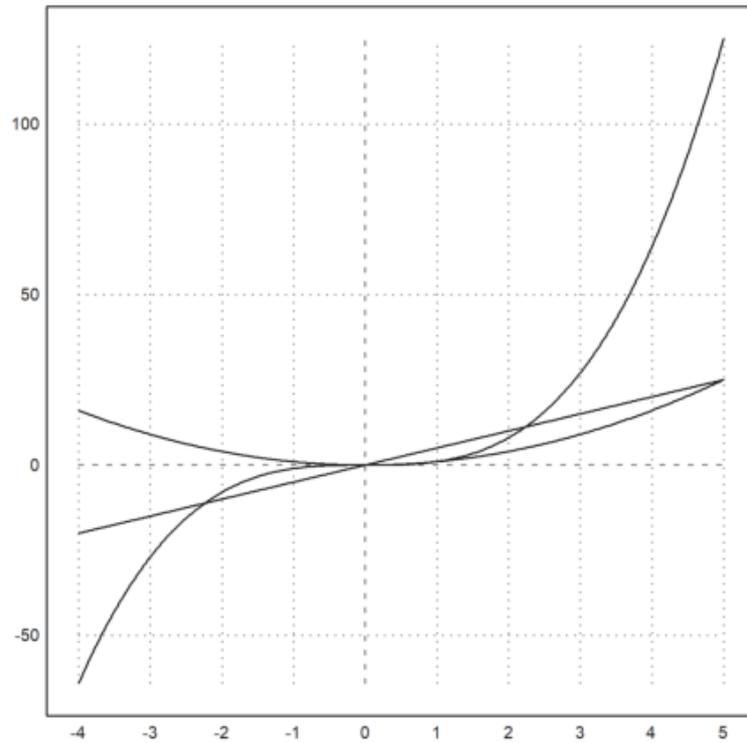
-dengan style "-" untuk fungsi y dan "-" untuk fungsi g

-color=10:11 untuk menentukan warna yang sesuai dengan no warna konstan EMT

-thickness ketebalan grafik

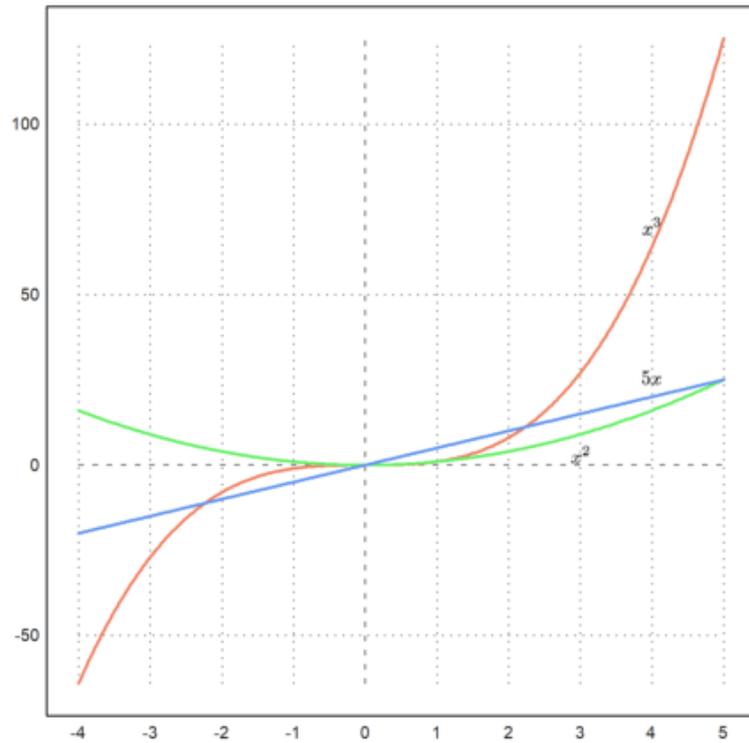
Cara menggambar sekumpulan kurva dalam satu perintah plot 2d

```
>plot2d(["x^3", "x^2", "5x"], -4, 5) :
```



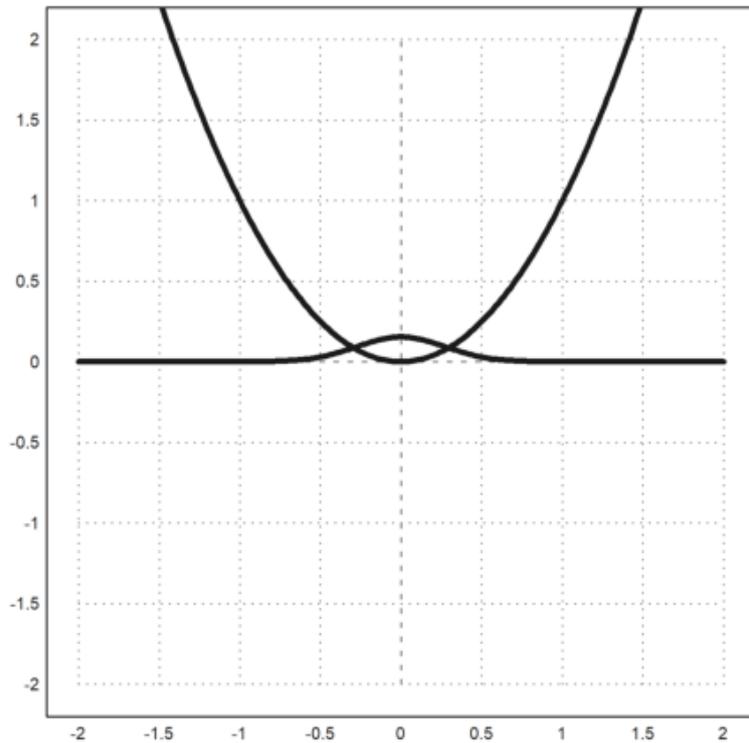
Dalam perintah di atas menggunakan [] untuk menggambar sekumpulan kurva, dalam contoh di atas menggambar 3 kurva dengan batas berada pada rentang -4 sampai 5. Ketiga kurva berpotongan di  $x=0$ .

```
>plot2d(["x^3","x^2","5x"],-4,5,color=[10:12], thickness=2:2:2);...
>label(latex("x^3"),4,64,pos="uc");...
>label(latex("x^2"),3,9,pos="lc");...
>label(latex("5x"),4,20,pos="uc"):
```



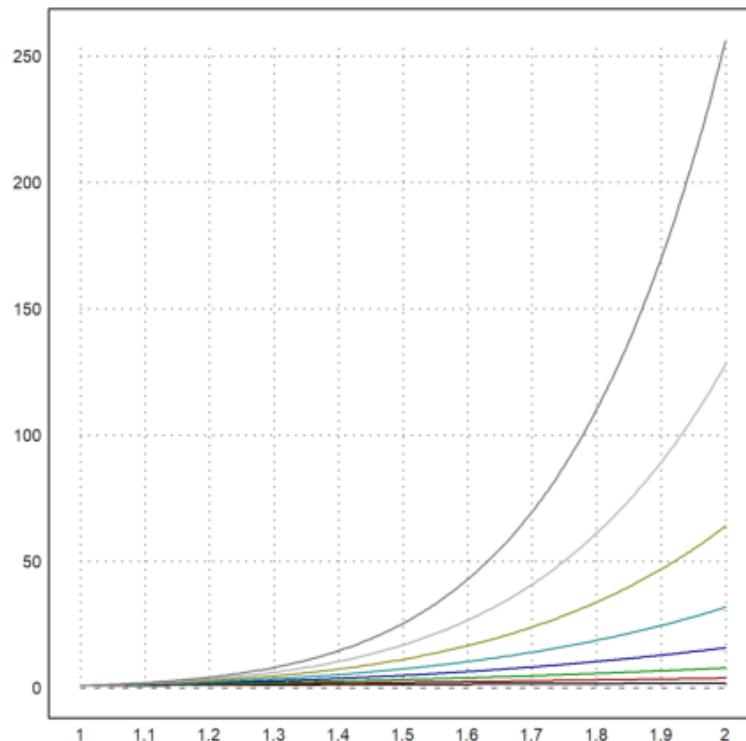
Unruk memudahkan dalam membedakan kurva maka diberikan warna dengan warna oranye= $10(x^3)$ , hijau= $11(x^2)$ , dan biru= $12(5x)$ . serta diberikan label nama

```
>a:=6.5; f&=exp(-a*x^2)/a; g&=x^2;
>plot2d([f,g],r=2,thickness=4):
```



selain langsung dituliskan setelah perintah plot2d, bisa juga didefinisikan terlebih dahulu fungsinya lalu dari definisi tersebut dijalankan pada perintah seperti cara sebelumnya menggunakan []. parameter r untuk mengatur rentang plot dan parameter thickness untuk mengatur ketebalan garis grafik.

```
>n=(1:8)'; plot2d("x^n",1,2,color=1:8):
```



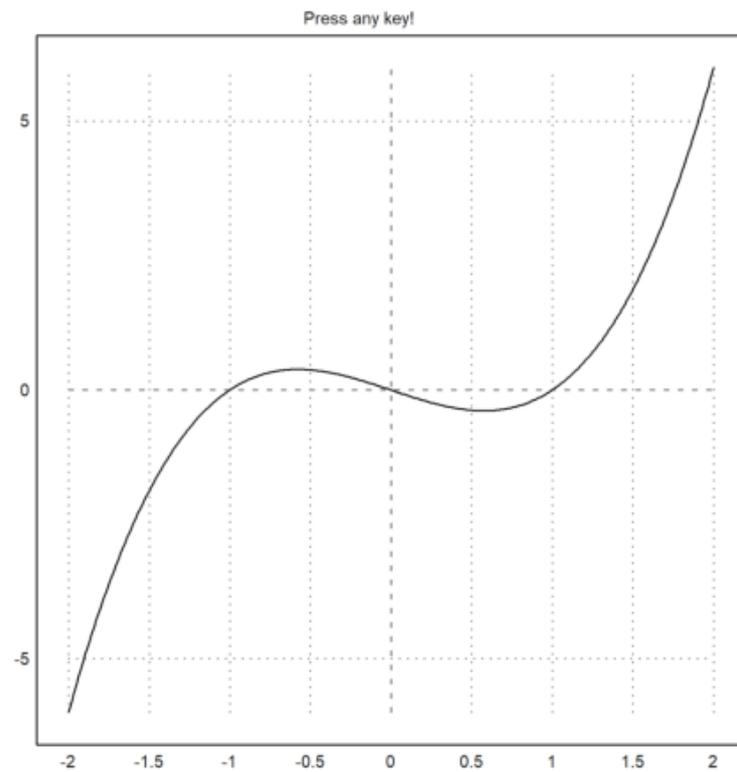
menggambar banyak kurva dengan satu fungsi diketahui berurutan.

Saat memplot suatu fungsi atau ekspresi, pengguna memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kursor atau mouse.

Di Euler Math Toolbox kita dapat menggunakan perintah >user untuk memperbesar dan memperkecil kurva, memindahkan memilih jendela kurva dengan mouse, mengembalikan ke bentuk semula dengan spasi, dan keluar dengan enter

- perbesar dengan + atau -
- pindahkan plot dengan tombol kursor
- pilih jendela plot dengan mouse
- atur ulang tampilan dengan spasi
- keluar dengan kembali

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



contoh soal

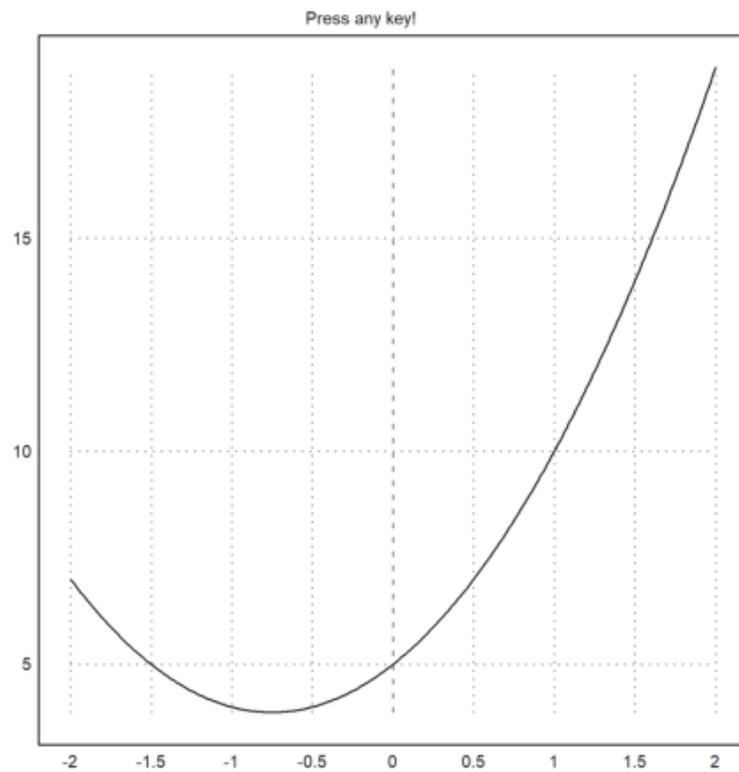
---

1. Gambarkan grafik interaktif dengan fungsi:

$$f(x) = 2x^2 + 3x + 5$$

dengan  $x=2$

```
>plot2d({"2*x^2+3*x+5",x=2},>user,title="Press any key!"):
```



2. Gambarkan grafik dibawah ini!

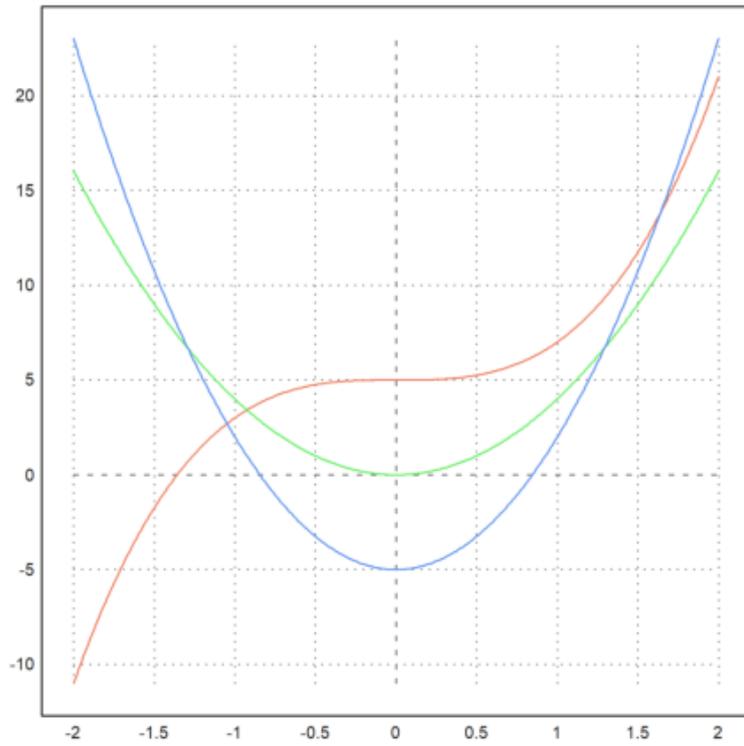
$$f(x) = 2x^3 + 5$$

$$g(x) = 4x^2$$

$$h(x) = 7x^2 - 5$$

dengan setiap fungsi memiliki warna yang berbeda-beda

```
>plot2d(["2*x^3+5", "4*x^2", "7*x^2-5"], -4:5, color=[10:12]):
```

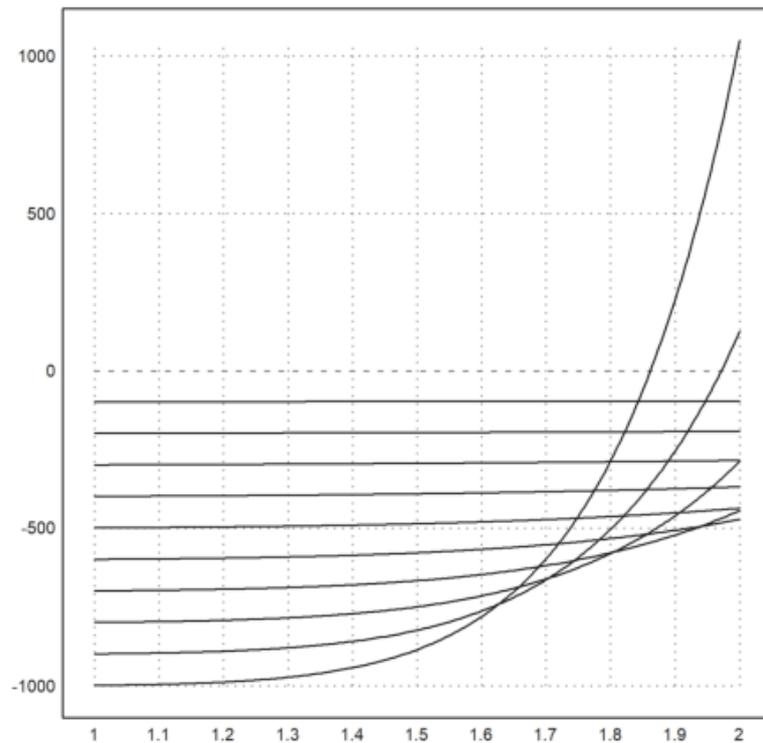


3. Gambarlah grafik :

$$f(x) = 2x^a - 100a$$

dengan  $a=1:10$

```
>a=(1:10)'; plot2d("2*x^a-100a",1,2):
```



Salah satu tujuan utama adalah membuat sketsa kurva yang diberikan dalam bentuk persamaan parametrik. Metode yang paling sederhana adalah mencari hubungan antara  $x$  dan  $y$ ; yang biasanya dilakukan dengan melakukan substitusi parameter  $t$ .

$$> t=(-1:2); x=2*t-4$$

$$[-6, -4, -2, 0]$$

$$> t=(-1:2); y=t^2+1$$

$$[2, 1, 2, 5]$$

Dari  $x = 2t - 4$ ; diperoleh

$$t = \frac{x+4}{2}, \text{ maka}$$

$$y = 4t^2 + 1 = 4\left(\frac{x+4}{2}\right)^2 + 1 = (x+4)^2 + 1 = x^2 + 8x + 17$$

Jadi, kurva tersebut merupakan bagian dari parabola

$$y = (x+4)^2 + 1, -1 \leq t \leq 2, \text{ maka :}$$

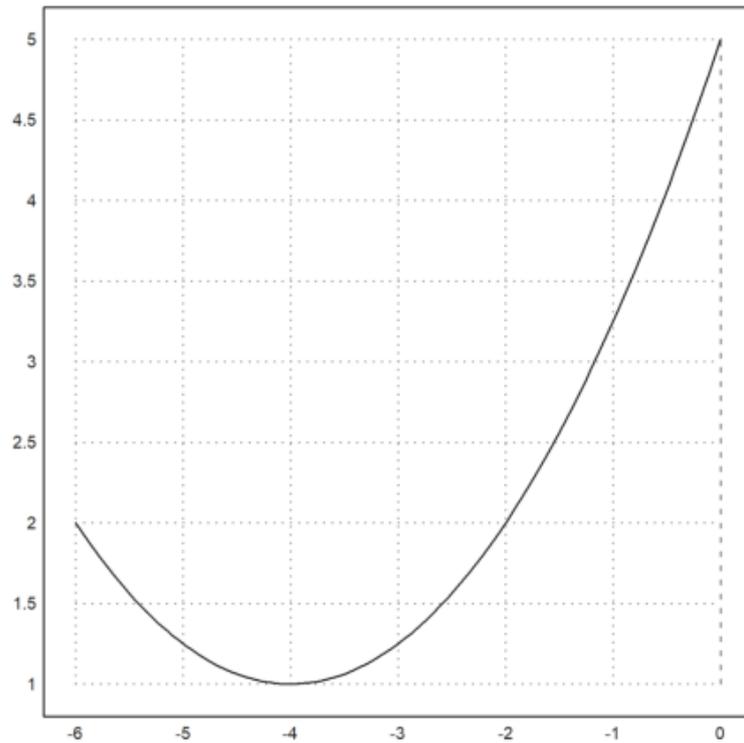
$$-2 \leq 2t \leq 4$$

$$-6 \leq 2t - 4 \leq 0,$$

$$-6 \leq x \leq 0$$

Jadi, domain grafik adalah interval  $[-2, 0]$ .

```
> t=linspace(-1,2,100); x=2*t-4; y=t^2+1;  
> plot2d(x,y,grid=2):
```



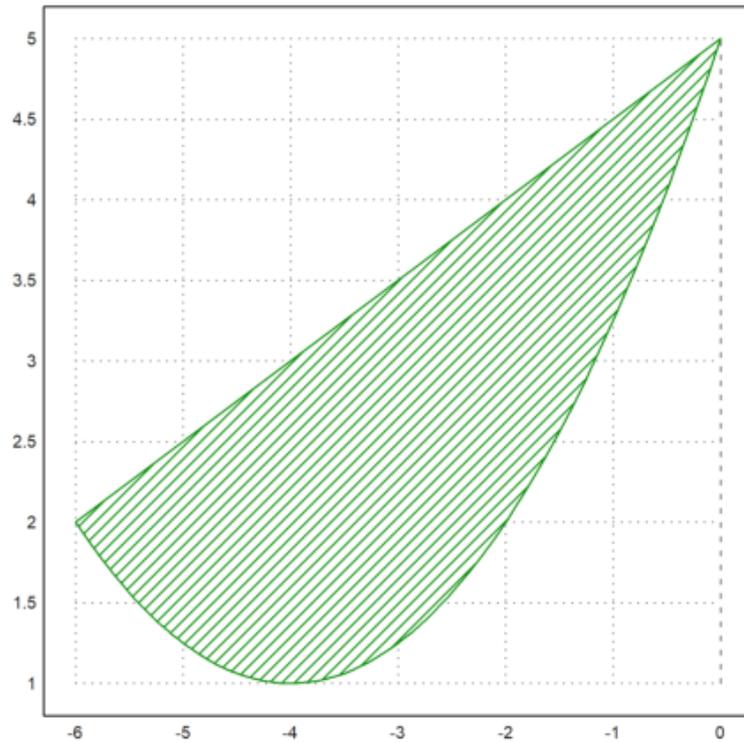
`linspace(a,b,n)`

Pada baris perintah kita menggunakan `linspace` untuk memudahkan batasan pada parameter `t`. `n` adalah nilai spasi linear dalam `[a,b]` yang termasuk `a` dan `b`.

Sekarang grafik fungsi parameter tersebut memiliki nilai vektor `x` dan `y`. `Plot2d()` juga bisa membuat kurva dengan menghubungkan titik-titik yang ada di grafik. Selain itu juga plot tersebut dapat diisi dengan baris perintah `>filled`.

Sehingga grafik fungsi yang ditampilkan bagus karena kita menambahkan arsiran pada plot yang dihasilkan seperti berikut:

```
> t=linspace(-1,2,100); x=2*t-4; y=t^2+1;  
> plot2d(x,y,grid=2,>filled,style="/"):
```



Titik stasioner dari garis singgung horizontal dapat dicari dengan menyelesaikan:

$$\frac{dy}{dt} = 0, y = t^2 + 1$$

$$2t = 0$$

$$t = 0$$

Artinya ketika  $t=0$ , maka posisi titik tersebut adalah titik stasioner untuk garis singgung horizontal. Jika  $t=0$ , maka  $(0,1)$  adalah titik yang dimaksud. Oleh karena itu, garis singgung horizontalnya adalah  $y=1$ .

Titik stasioner dari garis singgung vertikal dapat dicari dengan menyelesaikan:

$$\frac{dx}{dt} = 0, x = 2t - 4$$

$$\frac{dx}{dt} = \frac{d}{dt}(2t - 4) = 2 \frac{d}{dt}(t) - 0 = 2$$

Kemudian,  $dx/dt$  sama dengan 0 untuk mencari titik stasionernya:

$$2=0$$

```
> reset ();
```

Contoh:

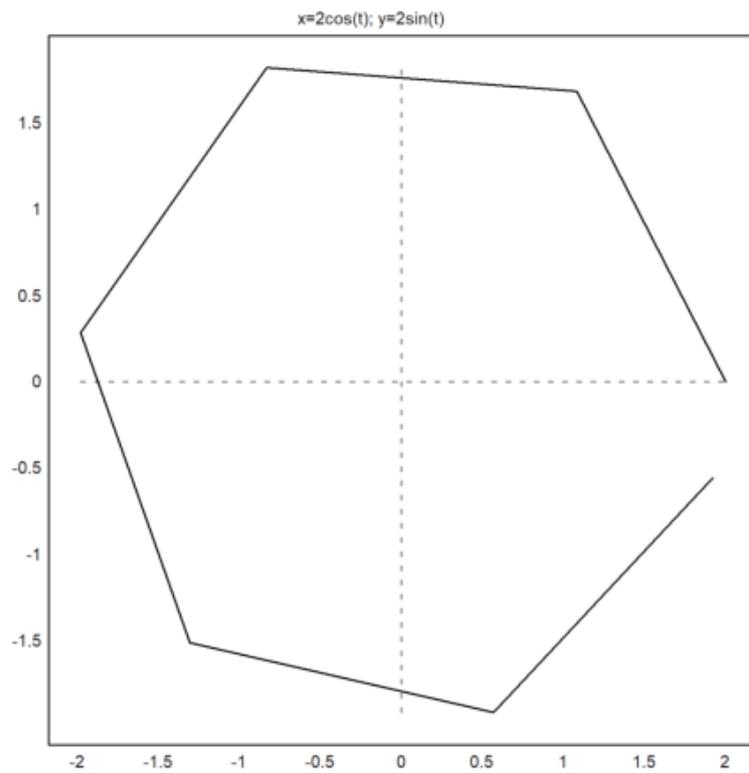
Buat sketsa kurva yang dideskripsikan oleh persamaan parametrik berikut:

$$x = 2\cos t$$

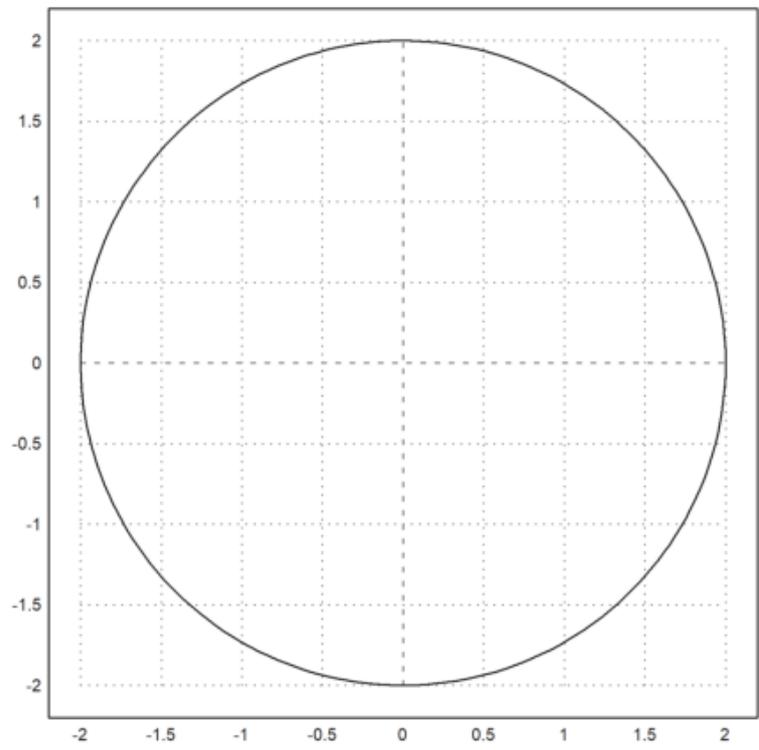
$$y = 2\sin t$$

$$0 \leq t \leq 2\pi$$

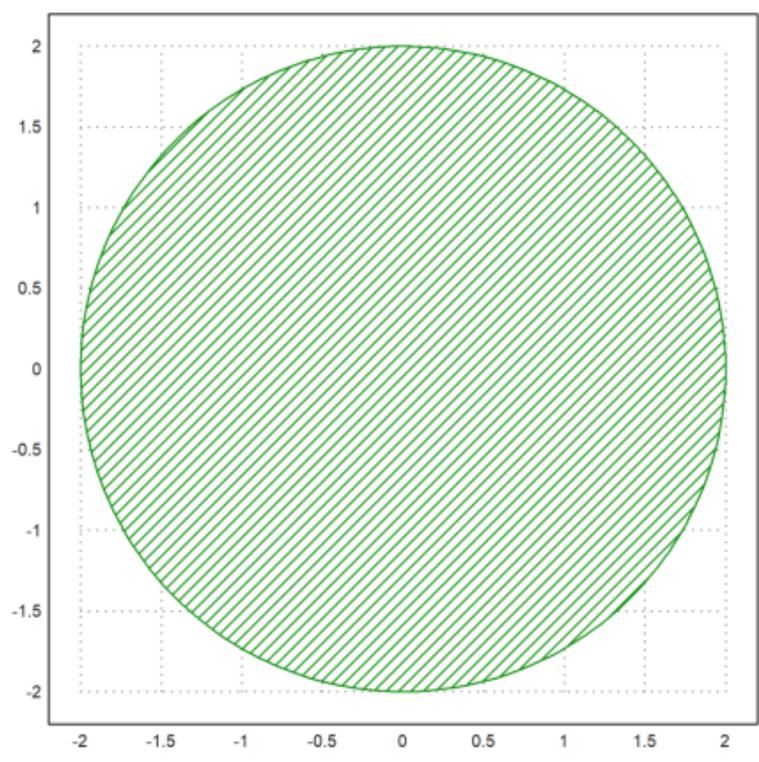
```
> t=(0:2pi); x=2*cos(t);  
> t=(0:2pi); y=2*sin(t);  
> t=linspace(0,2pi,100);  
> plot2d(x,y,grid=1, title="x=2cos(t); y=2sin(t)");
```



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



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



Garis singgung persamaan parametrik:

Secara umum, gradien garis singgung untuk persamaan parametrik adalah:

$$\frac{dy}{dx} = \frac{\frac{dy}{dt}}{\frac{dx}{dt}}$$

Sehingga untuk persamaan parametrik berikut:

$$x = 2\cos t$$

$$y = 2\sin t$$

maka:

$$\frac{dy}{dx} = \frac{\frac{dy}{dt}}{\frac{dx}{dt}} = \frac{2\cos t}{-2\sin t} = -\cot t$$

Ketika  $t = \pi/6$ , maka diperoleh titik  $(\pi/6, 1)$  lalu garis singgung di titik tersebut memiliki gradien:

$$m = -\cot(\pi/6) = \sqrt{3}$$

Karena kurva parametrik dilalui titik  $(\pi/6, 1)$  maka garis singgung di titik tersebut memiliki gradien 0, sehingga:

$$(y - 1) = \sqrt{3}\left(x - \frac{1}{2}\sqrt{3}\right)$$

$$y = \sqrt{3}\left(x - \frac{1}{2}\sqrt{3}\right) + 1$$

$$y = \sqrt{3}x - \frac{3}{2} + 1$$

$$y = \sqrt{3}x - \frac{1}{2}$$

```
> reset();
```

Seperti halnya menggambar suatu persamaan, persamaan parametrik dapat digambarkan dengan mencacah nilai dari variabel  $x$  dan variabel  $y$ . Tentu, nilai dari dua variabel tersebut diperoleh dengan mensubstitusikan beberapa nilai dari parameternya dahulu.

Contoh:

Gambar sketsa dari grafik

$$x = 4\cos(t)$$

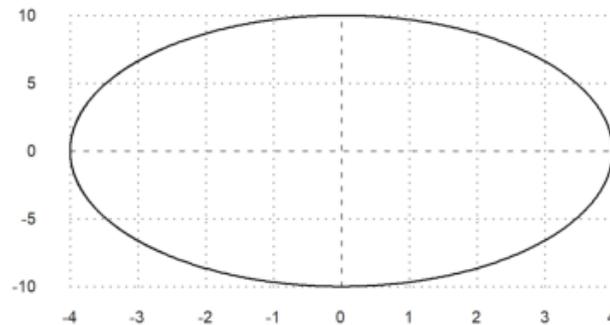
$$y = 10\sin(t)$$

$$0 \leq t \leq 2\pi$$

```

> t=(0:2pi); x=4*cos(t);
> t=(0:2pi); y=10*sin(t);
> t=linspace(-4,4,1000); x=4*cos(t); y=10*sin(t);
> aspect(2); plot2d(x,y,<square,<frame,<filled,style="[]#" ):

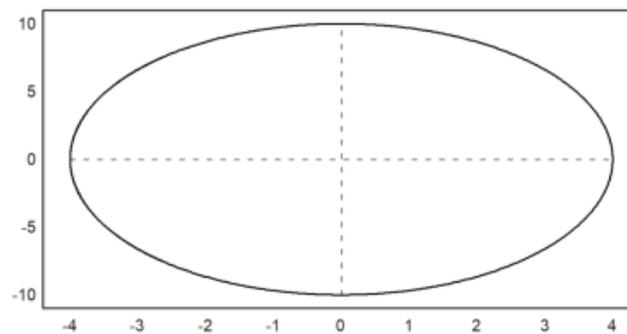
```



```

> reset();
> a=4;
> b=10;
> x= a*cos(t);
> y=b*sin(t);
> aspect(2); t=linspace(-4,4,1000); plot2d(x,y,grid=1):

```



Cycloid atau sikloid adalah kurva yang dihasilkan oleh suatu titik pada keliling lingkaran yang menggelinding sepanjang garis lurus . Jika  $r$  adalah jari-jari lingkaran dan  $\theta$  adalah perpindahan sudut lingkaran, maka persamaan kutub kurvanya adalah

$$x = r(\theta - \sin\theta)$$

$$y = r(1 - \cos\theta)$$

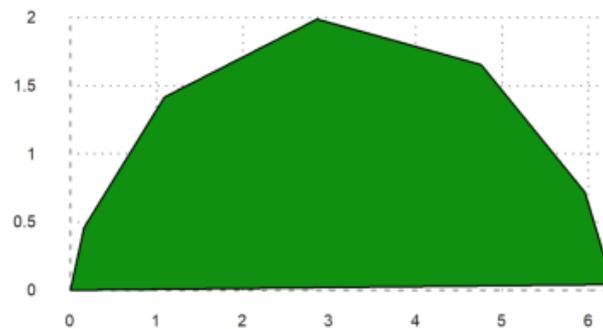
Sehingga kita akan menggambar grafik fungsi persamaan parametrik sikloid:

$$x = 2(\theta - \sin\theta)$$

$$y = 2(1 - \cos\theta)$$

$$0 \leq \theta \leq 6\pi$$

```
> t=(0:2*pi); x=t-sin(t);  
> t=(0:2*pi); y=1-cos(t);  
> t=linspace(0,6*pi,100); r=2;  
> aspect(2); plot2d(x,y,<square,<frame,>filled,style="[]#") :
```



Untuk mencari luas satu area sikloid dapat dicari dengan menggunakan:

$$3\pi a^2$$

Dengan persamaan sikloid:

$$x = r(\theta - \sin\theta)$$

$$y = r(1 - \cos\theta)$$

Maka luas satu area sikloid adalah:

$$3\pi(2)^2$$

$$3\pi(4)$$

$$12\pi$$

Sehingga luas satu lengkungan sikloid adalah  $12\pi$ .

Untuk panjang busur satu area sikloid adalah:

$$8a$$

$$8(2)$$

$$16$$

Sehingga panjang busur untuk satu area sikloid adalah 16.

Di euler sendiri, fungsi implisit haruslah berupa fungsi atau ekspresi yang parameternya adalah  $x$  dan  $y$ .

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

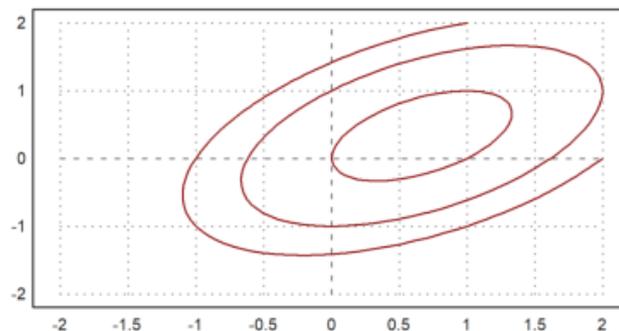
Untuk menggambar himpunan  $f(x,y)=c$  untuk satu atau lebih konstanta  $c$ , dapat menggunakan "plot2d()".

Fungsi implisit juga dapat diisi dengan persamaan tingkat

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

Untuk fungsi ini harus berupa matriks  $2 \times n$  dimana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval.

```
> aspect(2)
> plot2d("x^2+y^2-x*y-x", r=2, level=[0,1,2], contourcolor=red):
```



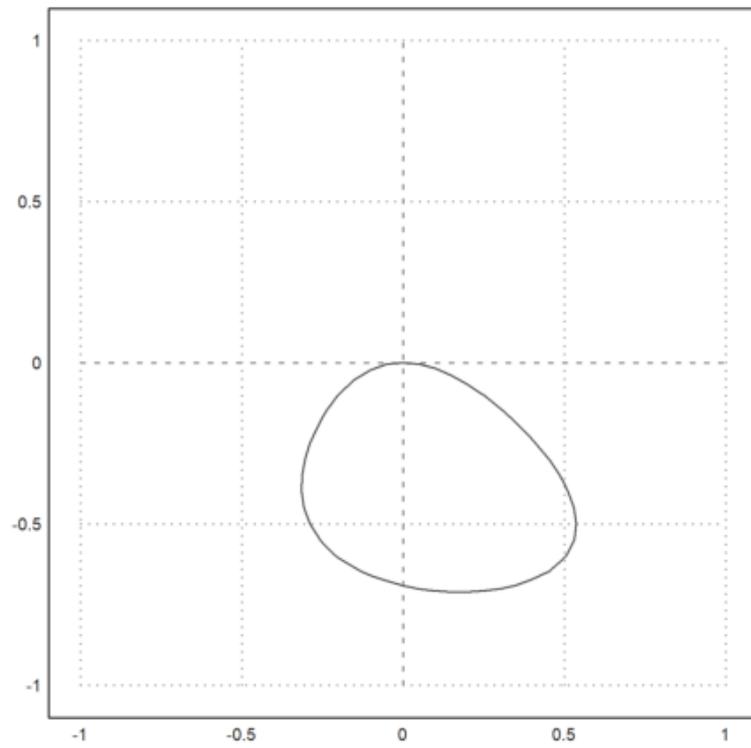
$r$  merupakan batas sumbu  $x$  dan  $y$  yang berkisar dari -2 hingga 2.

level adalah daftar nilai-nilai level yang ingin ditampilkan dalam grafik kontur (harus lebih dari 0).

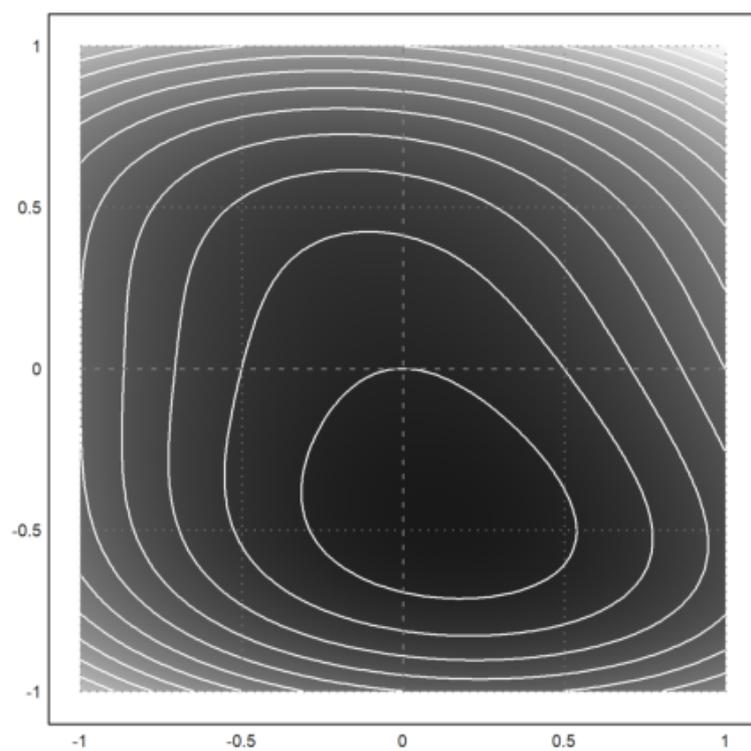
untuk lebih dari 1, penulisannya = [0,1,2,...]

jadi semakin besar levelnya, semakin besar konturnya

```
> reset;
> expr:="2*x^2+x*y+3*y^4+y"; //mendefinisikan ekspresi f(x,y)
> plot2d(expr, level=0):
```



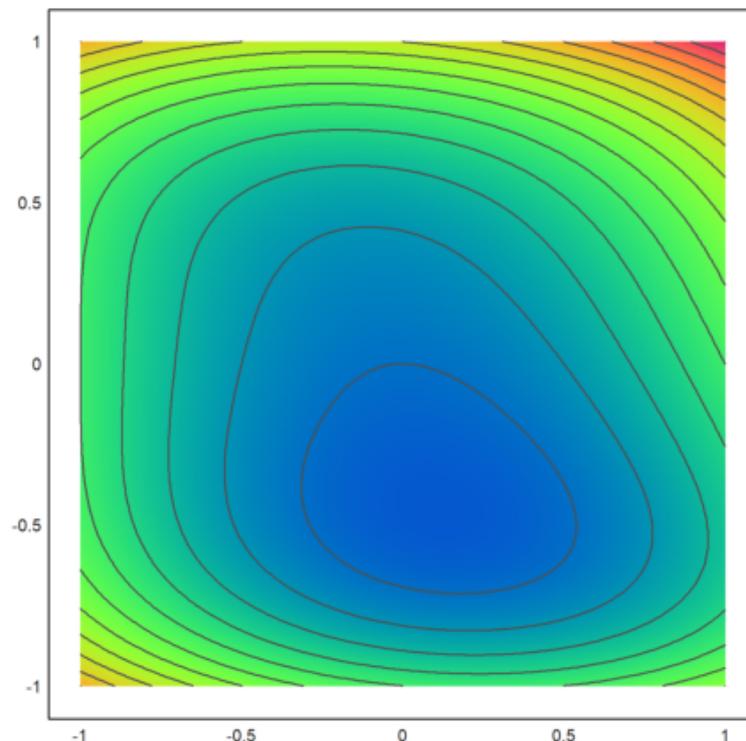
```
> plot2d(expr, level=0:0.5:20, >hue, contourcolor=white, n=200):
```



garis kontur untuk nilai level dari 0 hingga 20 dengan selang 0.5.

>hue berarti level yang rendah memiliki warna yang lebih pekat daripada level yang lebih tinggi

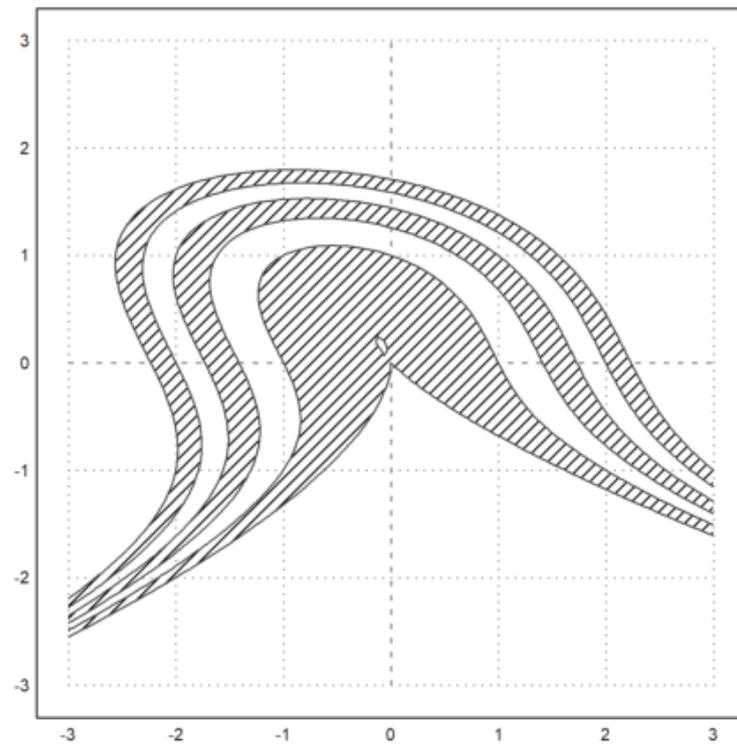
```
> plot2d(expr, level=0:0.5:20, >hue, >spectral, n=200, grid=4):
```



>spectral bertujuan untuk mengubah warna contour agar tampak seperti spektrum (kuning, orange, merah, hijau, biru, dan ungu).

grid berfungsi untuk mengatur jumlah grid(garis bantu) pada plot. grid=4 menunjukkan garis bantu pada plot dengan interval sebanyak 4 satuan. Garis grid ini membantu dalam memahami skala plot dan memudahkan untuk mengidentifikasi nilai pada sumbu x dan y.

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



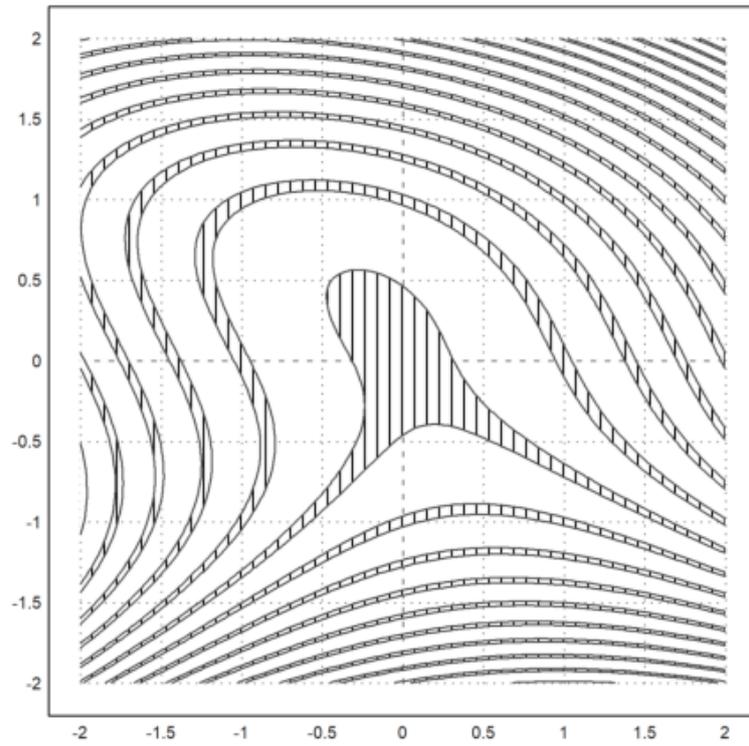
`level[a,b,c;d,e,f]`

`a,b,c` sebagai batas bawah

`d,e,f` sebagai batas atas

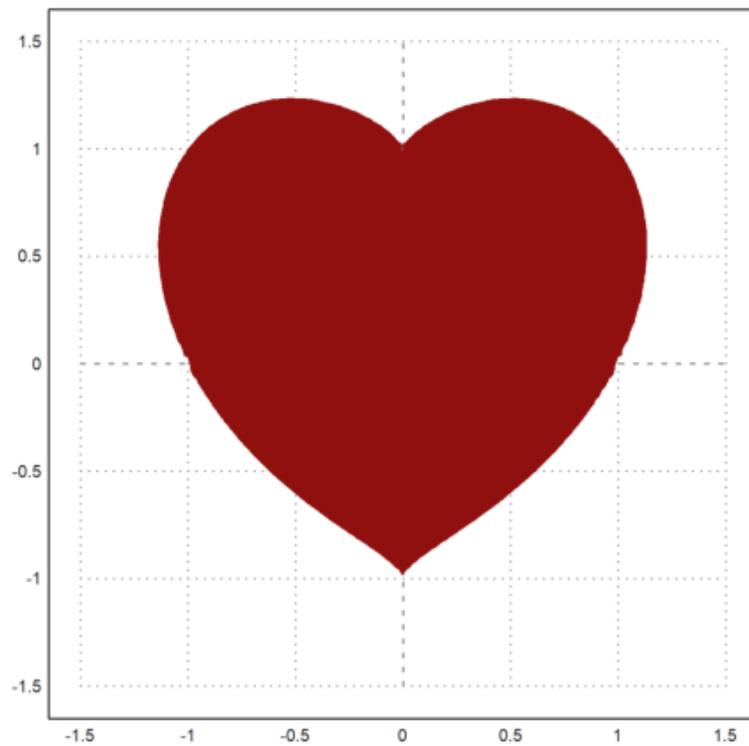
pada saat memberikan garis dalam plot kontur.

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



d1 adalah tingkat perubahan dalam kontur adalah sebesar 0.1

```
>plot2d("(x^2+y^2-1)^3-x^2*y^3", r=1.5, ...
>style=".", color=red, <outline, ...
>level=[-2;0],n=100):
```



<outline: menampilkan garis batas luar dari objek yang akan diplot

fungsi kompleks adalah suatu ekspresi dalam matematika yang melibatkan bilangan real dan bilangan imajiner. Fungsi kompleks dengan variabel kompleks  $z$  dinyatakan oleh  $w = f(z)$  dengan  $z = x + iy$ , yang dapat dinyatakan dengan

$$w(z) = u(x, y) + v(x, y)i$$

dimana  $u(x,y)$  adalah bilangan real dan  $v(x,y)$  bilangan imajiner.

fungsi kompleks dapat divisualisasikan dalam suatu kurva untuk melihat bagaimana pola / bentuk dari fungsi tersebut.

Penulisan  $z = x + iy$  dengan  $x$  dan  $y$  adalah bilangan real, yaitu :

```
> z = 4 + I*6
```

```
4+6i
```

```
> aspect(1); r=linspace(0,5,10); a=linspace(0,2pi,20)'; z=r*exp(I*a);
```

Persamaan

$$z = r \times \exp^{ia}$$

adalah bentuk umum dari representasi bilangan kompleks eksponensial bentuk polar.

$z$  = fungsi yang akan digambarkan

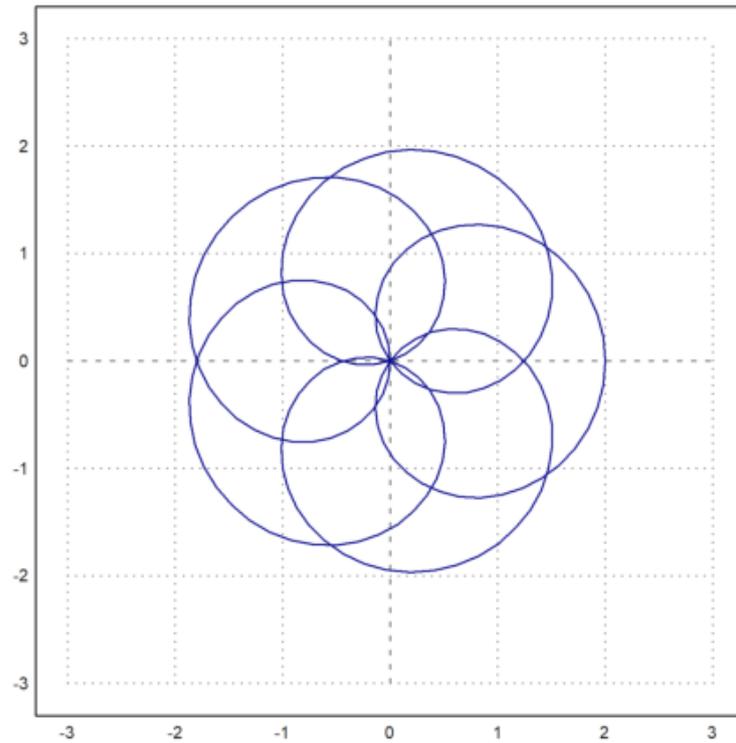
$r$  = jarak dari titik kompleks ke titik asal

$a$  = sudut argumen (antara sumbu real positif dan garis yang menghubungkan titik tersebut dengan pusat koordinat dalam arah berlawanan jarum jam)

$cgrid$  = tampilan grid pada plot yang dapat diubah sesuai kebutuhan / keinginan. Jika  $cgrid=0$  maka grid akan disembunyikan

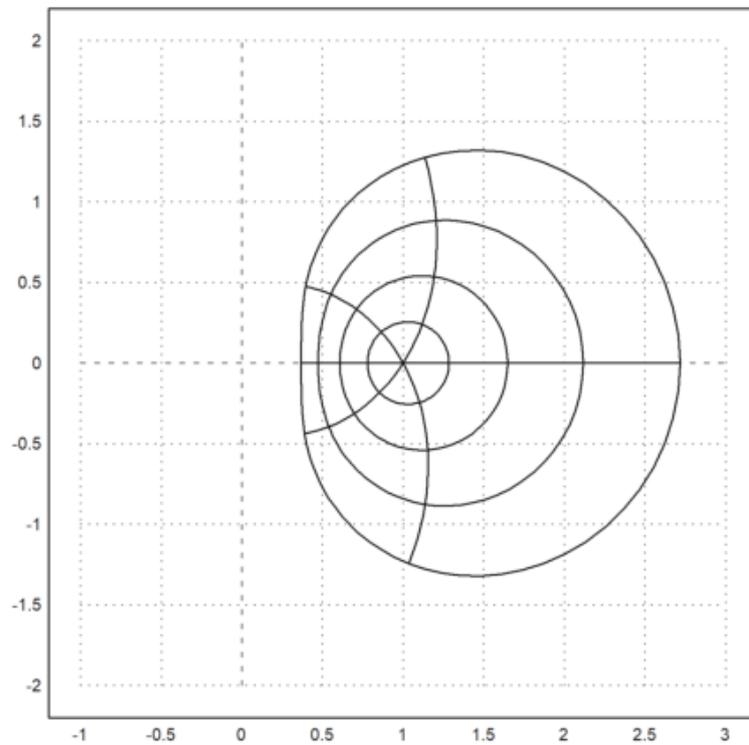
$r$  dan  $a$  disini adalah suatu array. Untuk  $r$  yaitu array dari 0 sampai 5 yang berisi 20 elemen.

```
> aspect(1); n=linspace(0,2pi,200);  
> plot2d(exp(I*n)+exp(6*I*n),-3,3,-3,3, color=blue):
```



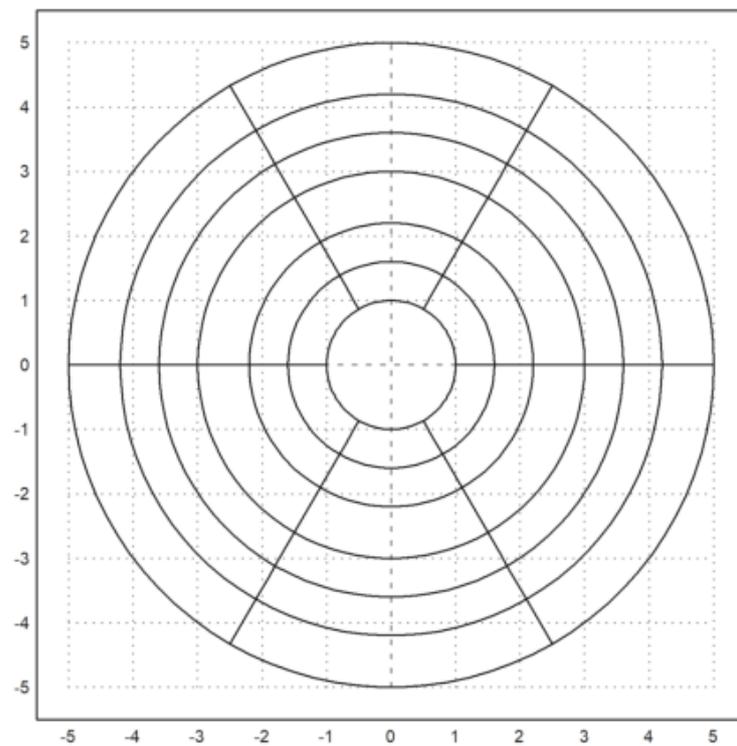
Contoh di atas merupakan penjumlahan dua fungsi eksponensial dengan  $n$  yang merupakan suatu array kompleks dari 0 sampai  $2\pi$  hingga 200 titik sehingga dapat membentuk beberapa lingkaran. Gambar di atas menggunakan aspect 1 dengan rentang sumbu  $x$  dan  $y$  dari -3 sampai 3.

```
> aspect(1); r=linspace(0,1,20); a=linspace(0,2pi,100)'; z=r*exp(I*a);
> plot2d(exp(z),-1,3,-2,2,cgrid=[6,4]):
```



Gambar di atas menunjukkan hasil dari fungsi eksponensial  $z$  dengan  $r$  adalah array 1 sampai 5 yang berisi 10 elemen dan jumlah cgrid yang berbeda.

```
> aspect(1); r=linspace(1,5,20); a=linspace(0,2pi,90)'; z=r*(cos(a) + I*sin(a));
> plot2d(z,-5,5,-5,5, cgrid=6):
```



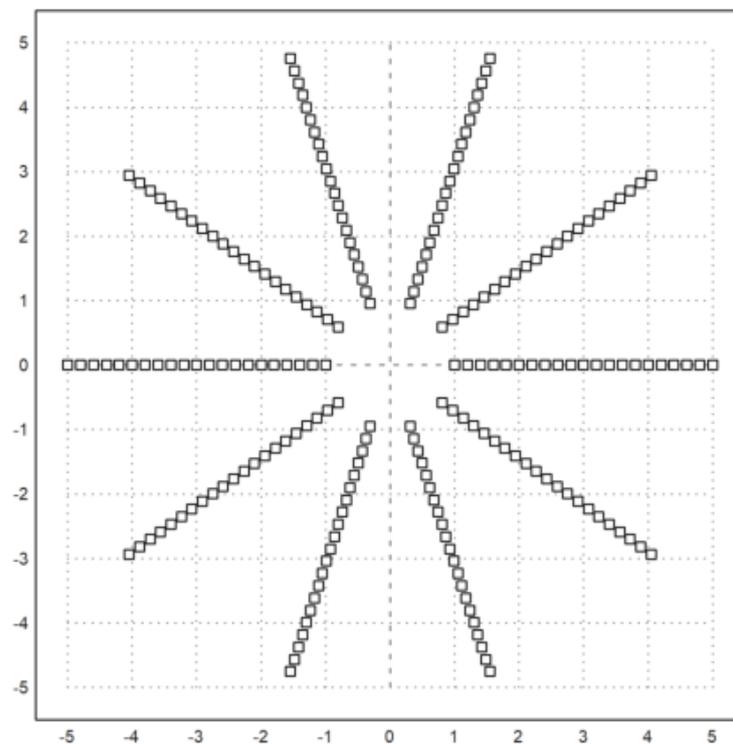
Perintah di atas merupakan eksponensial kompleks yang melibatkan fungsi trigonometri dengan r yaitu array dari 1 sampai 5 yang berisi 20 elemen di dalamnya

Bilangan kompleks  $z = x + yi$  dapat ditulis dalam bentuk polar menjadi

$$z = r \times (\cos(\Theta)) + i \times \sin(\Theta)$$

dimana r adalah modulus dari z

```
> aspect(1); r=linspace(1,5,20); a=linspace(0,2pi,10)'; z=r*(cos(a) + I*sin(a));  
> plot2d(z,-5,5,-5,5, cgrid=6, >points):
```



Dalam EMT ada beberapa perintah yang digunakan untuk menggambar daerah yang dibatasi oleh beberapa kurva, di antaranya yaitu:

- plot2d

Digunakan untuk melakukan plotting.

- filled=true

Digunakan untuk memberikan isian/arsiran pada daerah/area di bawah kurva saat plotting.

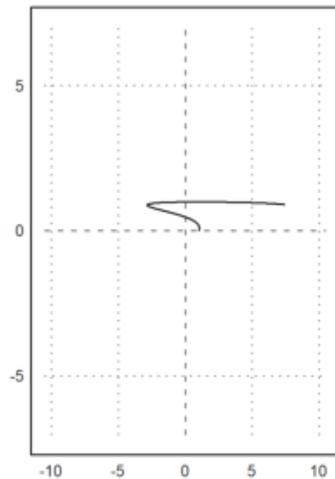
- style="..."

Digunakan untuk memilih gaya kurva yang akan digunakan saat plotting. Anda dapat memilih dari beberapa gaya, seperti "", "/", "\", atau "-". Dan hal ini mempengaruhi tampilan daerah kurva yang terbentuk.

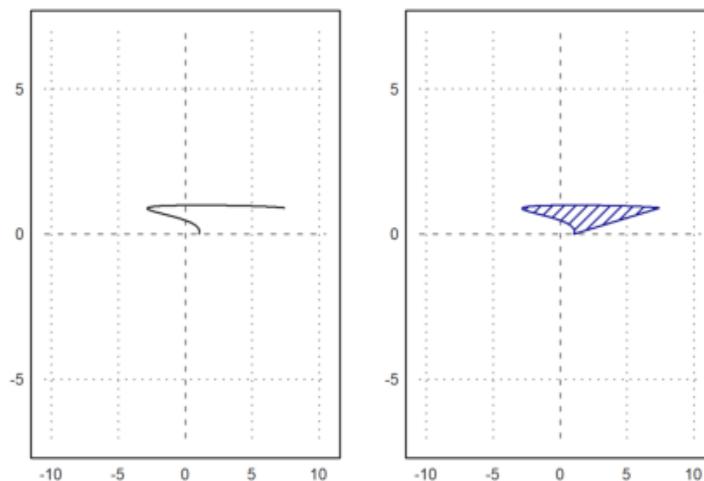
- fillcolor

Digunakan untuk menentukan warna isian yang akan digunakan untuk mengisi area di bawah kurva.

```
>t=linspace(0, 2pi, 1000); // parameter untuk kurva  
>x=cos(t)*exp(t/pi); y=sin(t/pi); // x(t) dan y(t)  
>figure(1,2); aspect(3/2)  
>figure(1); plot2d(x,y,r=7): // plot kurva
```



```
>figure(2); plot2d(x,y,r=7,>filled,style="/",fillcolor=blue); // mengisi kurva
>figure(0) :
```



#### Penjelasan

-  $t = \text{linspace}(0, 2\pi, 1000)$ ;

Pada langkah pertama yaitu mendefinisikan parameter  $t$  sebagai serangkaian 1000 titik antara 0 dan  $2\pi$ . Parameter  $t$  ini akan digunakan sebagai parameter untuk menggambar kurva.

-  $\cos(t) \cdot \exp(t/\pi)$ ;  $y = \sin(t) \cdot \exp(t/\pi)$ ;

Didefinisikan dua vektor  $x$  dan  $y$  yang merupakan koordinat  $x$  dan  $y$  dari kurva yang akan digambar.

Fungsi :

$$\cos(t) * \exp(t/\pi)$$

digunakan untuk menghitung komponen  $x(x(t))$ , dan

$$\sin(t) * \exp(t/\pi)$$

digunakan untuk menghitung komponen  $y$  ( $y(t)$ ) dari kurva.

```
- figure(1,2); aspect(3/2)
```

Perintah ini digunakan untuk mengatur tampilan gambar. Perintah `figure(1,2)` digunakan membuat dua gambar (1 dan 2) dalam satu jendela gambar. Dan perintah `aspect(3/2)` mengatur rasio aspek gambar menjadi 3:2, yang mempengaruhi bentuk dan ukuran gambar yang akan digambar.

```
- figure(1); plot2d(x,y,r=7);
```

Perintah ini memilih gambar pertama (1) dan menggunakan perintah `plot2d` untuk menggambar kurva yang dihitung sebelumnya. Parameter `r=10` mengatur lebar garis plot. Ini menghasilkan kurva tanpa adanya isi atau arsiran di dalamnya.

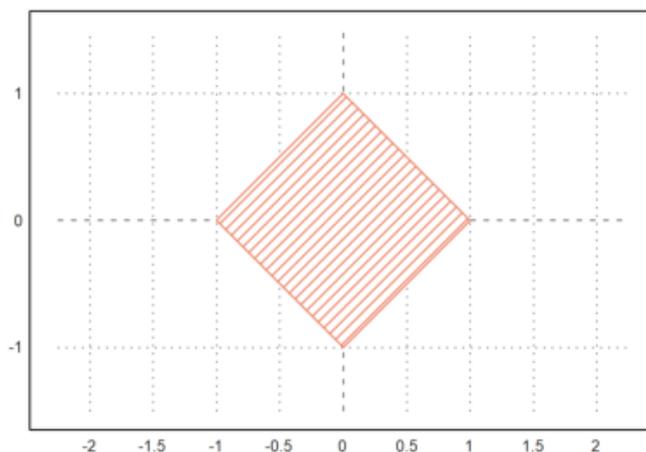
```
- figure(2); plot2d(x,y,r=7,>filled,style="/",fillcolor=blue);
```

Selanjutnya pada perintah ini beralih ke gambar kedua (2) dan menggunakan perintah `plot2d` lagi untuk menggambar kurva yang sama dengan pengisian area di bawahnya. Perintah `>filled` digunakan untuk mengisi area di bawah kurva, `style="/"` digunakan untuk mengatur gaya garis menjadi garis miring, dan `fillcolor=blue` digunakan untuk mengatur warna isian menjadi biru.

```
- figure(0);
```

Baris perintah ini digunakan untuk mengakhiri gambar dan kembali ke tampilan biasa tanpa gambar. Perintah ini berfungsi untuk menyelesaikan proses penggambaran.

```
>t=linspace(0,2pi,4);  
>plot2d(cos(t),sin(t),>filled, style="/",fillcolor=orange,r=1.5):
```

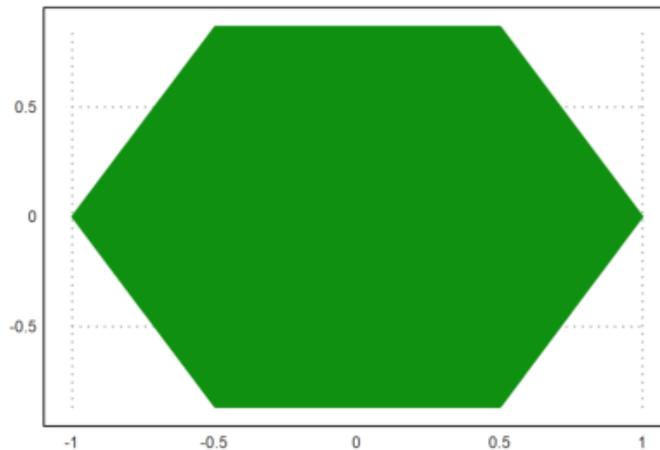


Penjelasan:

```
- t=linspace(0,2pi,4);
```

Pada perintah ini, kita definisikan vektor  $t$  dengan menggunakan perintah `linspace`. `linspace` digunakan untuk membuat vektor dengan 4 titik yang terletak secara merata antara 0 dan  $2\pi$ . Dalam konteks ini, vektor  $t$  akan digunakan sebagai parameter saat menggambar kurva.

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

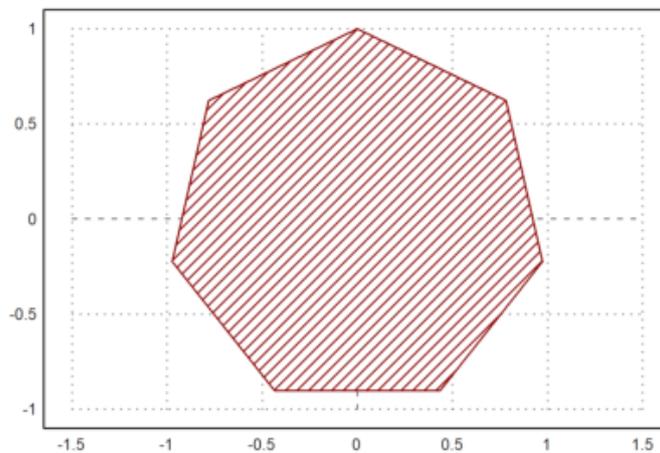


Penjelasan:

```
- t=linspace(0,2pi,6);
```

Pada perintah ini, kita definisikan vektor  $t$  dengan menggunakan perintah `linspace`. `Linspace` digunakan untuk membuat vektor dengan 6 titik yang terletak secara merata antara 0 dan  $2\pi$ . Dalam konteks ini, vektor  $t$  akan digunakan sebagai parameter saat menggambar kurva.

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

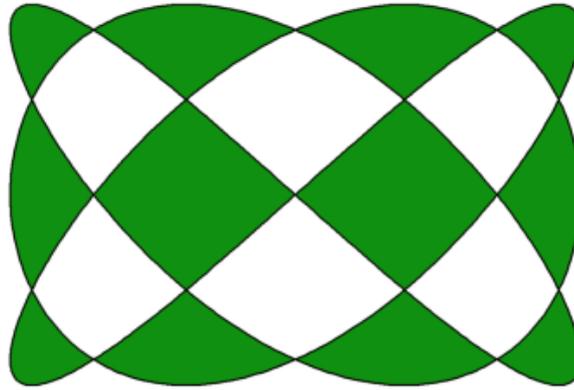


Penjelasan:

```
- t=linspace(0,2pi,7):
```

Fungsi `linspace` digunakan untuk membuat array berisi sejumlah nilai yang merata dalam rentang tertentu. Dalam hal ini, rentangnya adalah dari 0 hingga  $2\pi$  (dua kali nilai  $\pi$ ) dan sebanyak 7 titik akan dihasilkan. Ini akan digunakan sebagai sudut dalam koordinat polar untuk menggambarkan data.

```
>t=linspace(0,2pi,1000); x=cos(3*t); y=sin(4*t);
>plot2d(x,y,<grid,<frame,>filled):
```



Penjelasan:

- `t = linspace(0, 2*pi, 1000);`

Ini adalah perintah untuk membuat vektor `t` yang berisi 1000 nilai yang merata terdistribusi antara 0 hingga  $2\pi$ . Vektor `t` ini akan digunakan sebagai parameter waktu atau sudut dalam parameterisasi lingkaran. `linspace(0, 2*pi, 1000)` membuat 1000 titik antara 0 hingga  $2\pi$ , memberikan sudut-sudut yang merata di sepanjang satu putaran lingkaran.

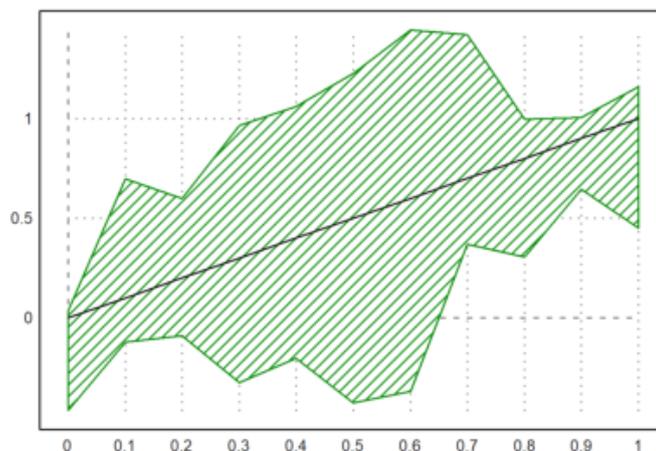
- `x = cos(3*t); y = sin(4*t);`

Ini adalah perintah untuk menghitung vektor `x` dan `y` yang menggambarkan lintasan dalam koordinat polar.

> `x = cos(3*t);` menghitung nilai `x` sebagai hasil dari fungsi kosinus dari 3 kali nilai `t`. Ini akan menghasilkan osilasi yang lebih cepat pada sumbu `x`.

> `y = sin(4*t);` menghitung nilai `y` sebagai hasil dari fungsi sinus dari 4 kali nilai `t`. Ini akan menghasilkan osilasi yang lebih cepat pada sumbu `y`.

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



Penjelasan:

- `t = 0:0.1:1;`

Ini adalah perintah untuk membuat vektor `t` yang berisi nilai-nilai dari 0 hingga 1 dengan interval 0.1. Hasilnya adalah vektor `[0, 0.1, 0.2, 0.3, ..., 0.9, 1]`.

```
- plot2d(t, interval(t - random(size(t)), t + random(size(t))), style="|");
```

Ini adalah perintah untuk membuat plot pertama. Rincian perintah ini adalah sebagai berikut:

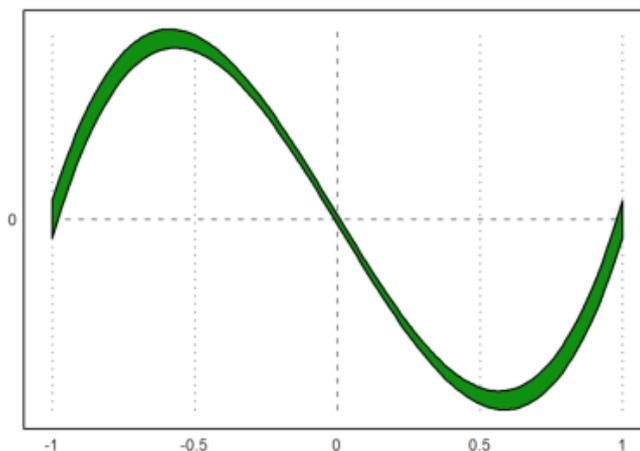
> interval(t - random(size(t)), t + random(size(t))) adalah interval yang digunakan untuk menggambar "garis" pada plot. Setiap titik pada sumbu x (t) akan dihubungkan oleh dua garis vertikal yang dibuat secara acak di sekitar titik tersebut menggunakan random(size(t)). Hasilnya adalah plot dengan garis-garis vertikal yang mewakili interval acak di sekitar setiap titik pada sumbu x.

> style="|" mengatur gaya plot menjadi garis vertikal ("|").

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

```
[ ~-0.04,0.04~, ~-0.02,0.059~, ~-0.0003,0.077~, ~0.018,0.096~,  
~0.0373,0.113~, ~0.0552,0.13~, ~0.0726,0.146~, ~0.0894,0.162~,  
~0.105,0.177~, ~0.121,0.191~, ~0.136,0.206~, ~0.151,0.219~,  
~0.165,0.232~, ~0.178,0.244~, ~0.191,0.256~, ~0.203,0.268~,  
~0.215,0.279~, ~0.227,0.289~, ~0.238,0.299~, ~0.248,0.308~,  
~0.258,0.317~, ~0.268,0.326~, ~0.276,0.334~, ~0.285,0.342~,  
~0.293,0.349~, ~0.301,0.355~, ~0.308,0.361~, ~0.314,0.367~,  
~0.32,0.373~, ~0.326,0.377~, ~0.332,0.382~, ~0.337,0.386~,  
~0.341,0.39~, ~0.345,0.393~, ~0.349,0.396~, ~0.352,0.398~,  
~0.355,0.4~, ~0.357,0.402~, ~0.359,0.404~, ~0.361,0.404~,  
~0.363,0.405~, ~0.364,0.405~, ~0.364,0.405~, ~0.364,0.405~,  
~0.364,0.404~, ~0.364,0.403~, ~0.363,0.402~, ~0.362,0.4~,  
~0.361,0.398~, ~0.359,0.395~, ~0.357,0.393~, ~0.355,0.39~,  
~0.352,0.387~, ~0.349,0.383~, ~0.346,0.379~, ~0.342,0.375~,  
~0.338,0.371~, ~0.334,0.366~, ~0.33,0.362~, ~0.325,0.356~,  
~0.321,0.351~, ~0.316,0.346~, ~0.31,0.34~, ~0.305,0.334~,  
~0.299,0.328~, ~0.293,0.321~, ~0.287,0.315~, ~0.28,0.308~,  
~0.274,0.301~, ~0.267,0.293~, ~0.26,0.286~, ~0.253,0.279~,  
~0.245,0.271~, ~0.238,0.263~, ~0.23,0.255~, ~0.222,0.247~,  
~0.214,0.238~, ~0.206,0.23~, ~0.197,0.221~, ~0.189,0.212~,  
... ]
```

```
>plot2d(t,y):
```



Penjelasan:

```
- t = -1:0.01:1;
```

Ini adalah perintah untuk membuat vektor t yang berisi nilai-nilai dari -1 hingga 1 dengan interval 0.01. Hasilnya adalah vektor t yang berisi nilai-nilai seperti [-1, -0.99, -0.98, ..., 0.99, 1]. Vektor t ini akan digunakan sebagai sumbu x pada plot.

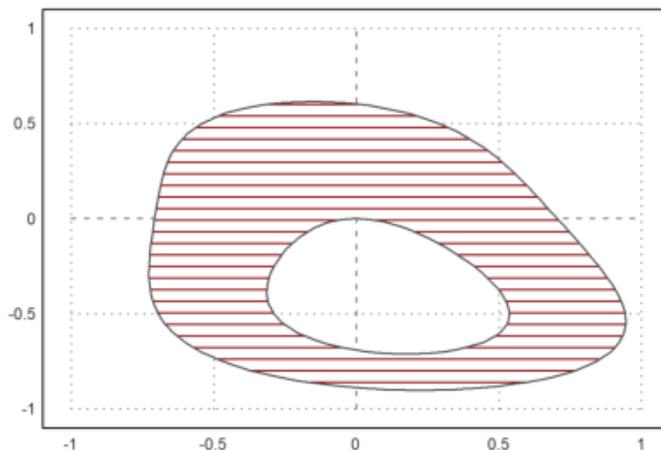
```
- x = t - 0.01, t + 0.01 ;
```

Ini adalah perintah yang menghitung vektor x. Tanda digunakan di sini untuk mendefinisikan dua interval, yaitu [ t - 0.01, t + 0.01 ]. Ini menghasilkan vektor x yang memiliki dua interval, satu yang kurang dari t - 0.01 dan satu yang lebih dari t + 0.01.

```
- y = x^3 - x;
```

Ini adalah perintah yang menghitung vektor y sebagai fungsi dari x. Fungsi ini menghitung nilai y dengan memasukkan setiap nilai x ke dalam rumus  $x^3 - x$ .

```
>expr := "2*x^2+x*y+3*y^4+y"; // mendefinisikan sebuah ekspresi f(x,y)
>plot2d(expr,level=[0;1],style="-",color=red): // 0 <= f(x,y) <= 1
```



Penjelasan:

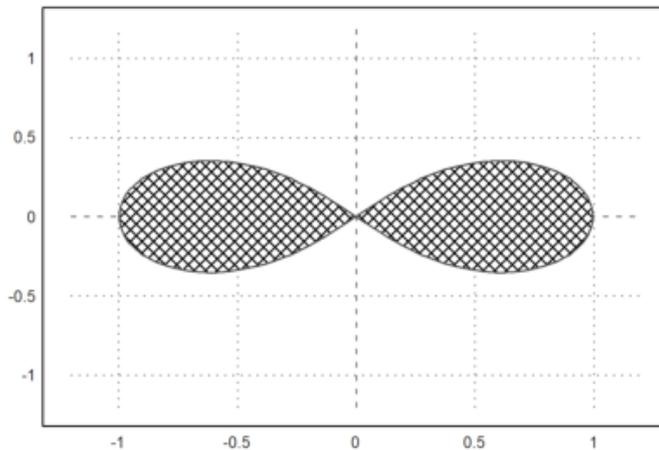
```
- expr := "2*x^2+x*y+3*y^4+y";
```

Ini adalah perintah untuk mendefinisikan ekspresi matematika yang disimpan dalam variabel expr. Ekspresi ini merupakan suatu fungsi  $f(x, y)$  yang tergantung pada dua variabel, yaitu x dan y. Ekspresi ini memiliki bentuk matematika yang terdiri dari berbagai suku, seperti kuadrat dari x, perkalian  $x*y$ , kuadrat dari y, dan lainnya.

Kita juga dapat mengisi rentang nilai seperti berikut:

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

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



Penjelasan:

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

Ini adalah perintah untuk membuat plot dari fungsi matematika yang didefinisikan dalam bentuk string:

`"(x^2+y^2)^2-x^2+y^2"`. Fungsi ini tergantung pada dua variabel, yaitu  $x$  dan  $y$ .

`(x^2+y^2)^2-x^2+y^2` adalah rumus dari fungsi matematika yang akan diplotkan.

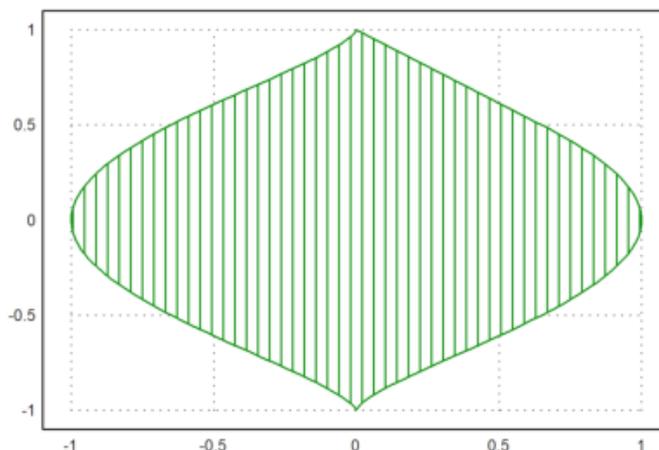
- `r=1.2` mengatur rentang (range) plot untuk kedua sumbu  $x$  dan  $y$ . Dalam hal ini, rentangnya adalah  $[-1.2, 1.2]$ , yang berarti plot akan berada dalam wilayah ini.

- `level=[-1;0]` mengatur tingkat kontur (contour levels) yang akan digunakan dalam plot. Dalam hal ini, ada dua tingkat kontur:  $-1$  dan  $0$ . Ini akan menentukan wilayah kontur dalam plot.

- `style="/\"` mengatur gaya plot menjadi garis berpotongan (`"/\"`).

Ini akan menghasilkan plot dengan garis-garis berpotongan yang menggambarkan kontur fungsi. Di sini menggunakan kombinasi dua simbol yaitu slash dan backslash (`"/\"`) untuk membuat garis-garis yang berpotongan.

```
>plot2d("sin(x)^3", "cos(x)", xmin=2*pi, >filled, style="|"):
```



Penjelasan:

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

Ini adalah perintah untuk membuat plot dari dua fungsi matematika, yaitu  $\sin(x)^3$  dan  $\cos(x)$ , dalam satu plot yang sama. Berikut adalah rincian perintah ini:

- "sin(x)^3" adalah ekspresi pertama yang akan diplotkan. Ini adalah fungsi trigonometri sin(x) yang dipangkatkan tiga. Fungsi ini tergantung pada variabel x.
- "cos(x)" adalah ekspresi kedua yang akan diplotkan. Ini adalah fungsi trigonometri cos(x). Fungsi ini juga tergantung pada variabel x.
- xmin=0 dan xmax=2\*pi mengatur rentang (range) plot untuk sumbu x dari 0 hingga 2pi. Ini adalah rentang yang akan ditampilkan dalam plot.
- >filled mengisi area di bawah kurva fungsi dengan warna, sehingga area di bawah kurva fungsi akan diisi dengan warna.
- style="|" mengatur gaya plot menjadi garis vertikal ("|"). Ini akan menghasilkan plot dengan garis-garis vertikal.

## Contoh-Contoh Soal

---

1. Gambarkan plot fungsi berikut ini

$$6x^2 + 5y^2 + 2xy + 4x + 3y$$

dengan interval

$$0 \leq f(x, y) \leq 1.5$$

2. Gambarkan plot fungsi berikut:

$$2x^2 + 3x + 1$$

$$2x + 6$$

dengan

$$xmin = 0$$

$$xmax = 10$$

3. Gambarkan plot fungsi berikut:

$$3x^2 + 3x + 1$$

$$2x + 6$$

$$-3x + 2$$

dengan

$$xmin = -10$$

xmax= 0

Poligon atau segi banyak adalah bangun datar yang digambarkan dengan jumlah terhingga dari garis lurus yang terhubung, sehingga membentuk sebuah rantai poligonal (atau sirkuit poligonal) yang tertutup.

Segi-n adalah sebuah poligon yang mempunyai n sisi, contohnya, segi-3(segitiga). Segi banyak memiliki paling sedikit tiga sisi.

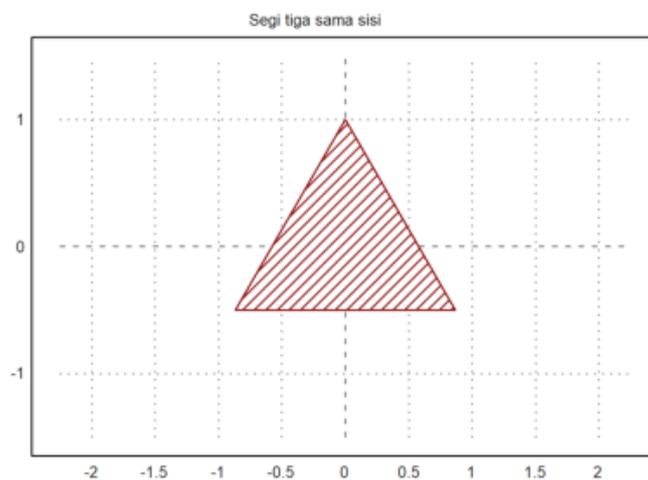
Terdapat dua macam segi banyak, yaitu:

1. Segi banyak beraturan
2. Segi banyak tidak beraturan

Contoh dari segi banyak beraturan adalah bangun segi tiga sama sisi, persegi, segi lima beraturan, segi enam beraturan, dst.

Contoh gambar

```
>t=linspace(0,2pi,3);...  
>plot2d(sin(t),cos(t),title="Segi tiga sama sisi",>filled,style="/",fillcolor=red,r=1.5):
```



Plot di atas membentuk segi tiga sama sisi.

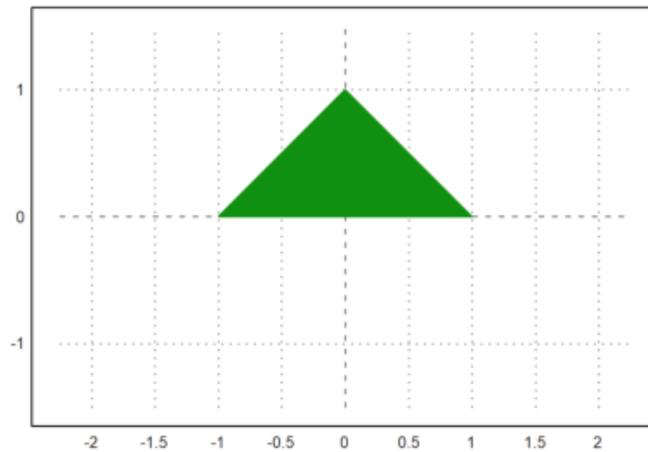
Mengubah style plot:

- filled=true mengisi plot.
- style = "...": Pilih dari "", "O", "o", "/", "\\", "\/", "+", "|", "-", "t". (style untuk isian poligon atau plot bar)
- fillcolor: dapat dilihat di bawah untuk warna yang tersedia.

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

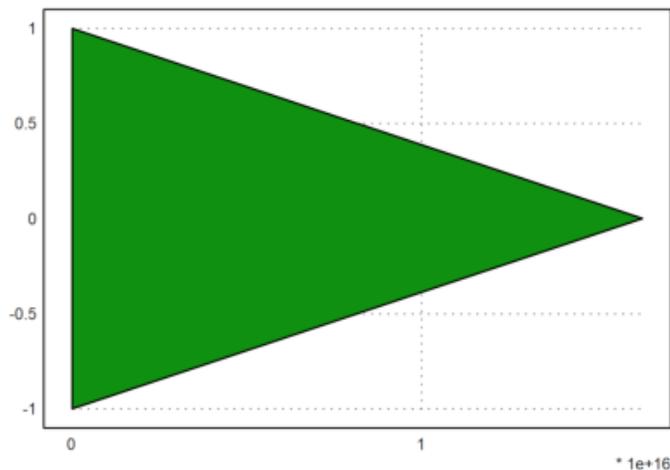
- 0..15: indeks warna default.
- konstanta warna: white, black, red, green, blue, cyan, olive, lightgray, gray, darkgray, orange, lightgreen, turquoise, lightblue, lightorange, yellow
- rgb (red,green,blue): parameter dalam bentuk real dalam [0,1].

```
>t=linspace(0,pi,2);...  
>plot2d(cos(t),sin(t),>filled,style="#",r=1.5):
```



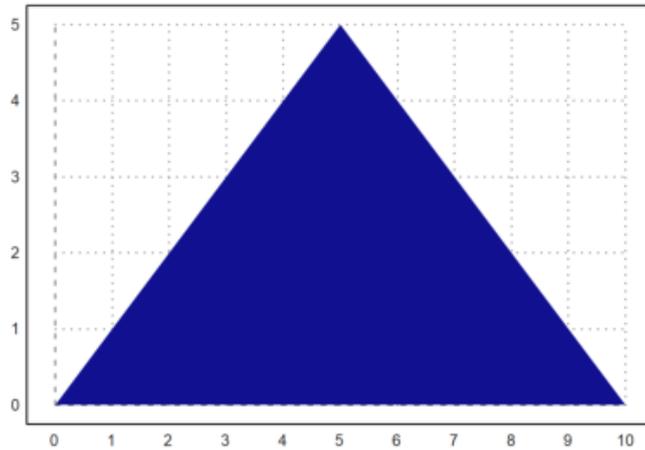
Dengan parameter  $t=\text{linspace}(0,\pi,2)$  menunjukkan bahwa terbentuk 2 titik dalam setengah lingkaran (karena batas  $t$  dari 0 hingga  $\pi$ ) sehingga terbentuk segitiga sama kaki. Namun, segitiga sama kaki bukan termasuk segi banyak beraturan karena hanya 2 sisi yang sama panjang.

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



Dihasilkan bentuk segitiga, namun bukan sama sisi melainkan sama kaki.

```
>plot2d([0,5,10],[0,5,0],>filled,fillcolor=blue,<outline):
```



Untuk menggambar segi banyak dapat juga menggunakan titik-titik koordinat yang saling dihubungkan. Plot di atas dengan menghubungkan titik  $(0,0)$ ,  $(5,5)$ ,  $(10,0)$  maka terbentuk segi tiga sama kaki. Namun bangun ini tidak termasuk segi banyak beraturan karena ada sisi yang panjangnya berbeda.

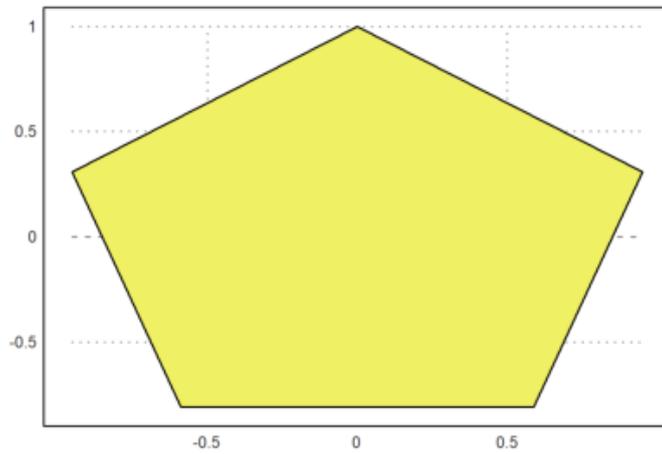
Untuk menghilangkan outline, kita dapat menambahkan "<outline" atau menggunakan `style="O"` dengan isian tanpa outline.

```
>plot2d([1, 3, 3, 1], [1, 1, 3, 3], >filled, fillcolor=blue, <outline, grid=0) :
```



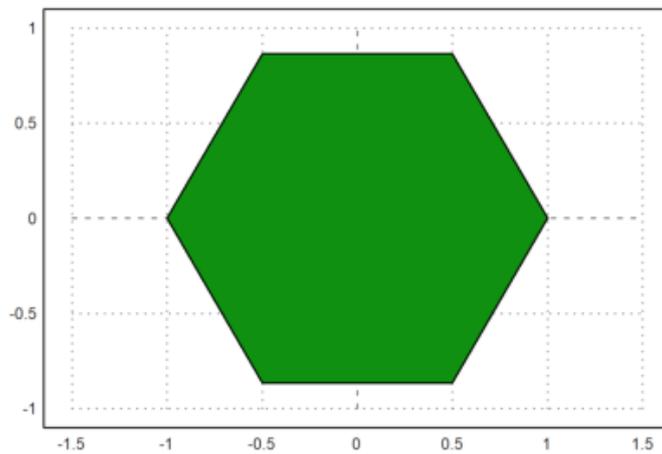
Dengan menghubungkan titik-titik  $(1,1)$ ,  $(3,1)$ ,  $(3,3)$ ,  $(1,3)$  maka terbentuk persegi. Pada plot di atas, `grid=0` membuat plot tidak menampilkan grid koordinat.

```
>t=linspace(0, 2pi, 5); ...
>plot2d(sin(t), cos(t), >filled, fillcolor=yellow) :
```



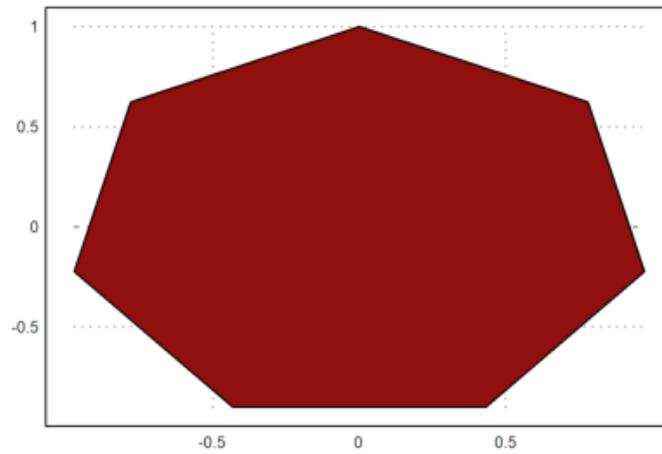
Terbentuk segi lima sama sisi.

```
>t=linspace(0,2pi,6);...
>plot2d(cos(t),sin(t),>filled,r=1):
```



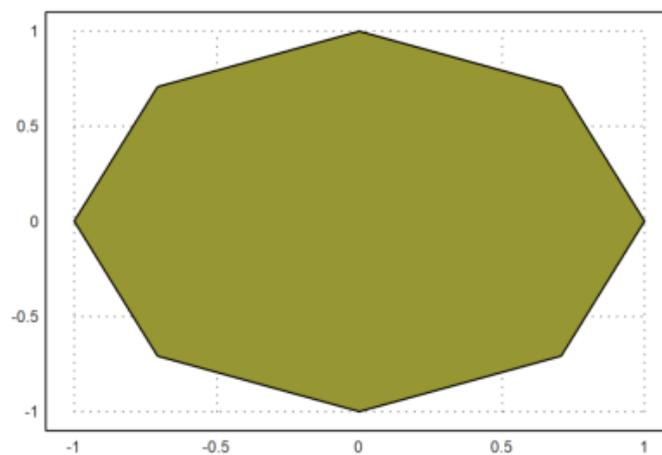
Terbentuk segi enam sama sisi.

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



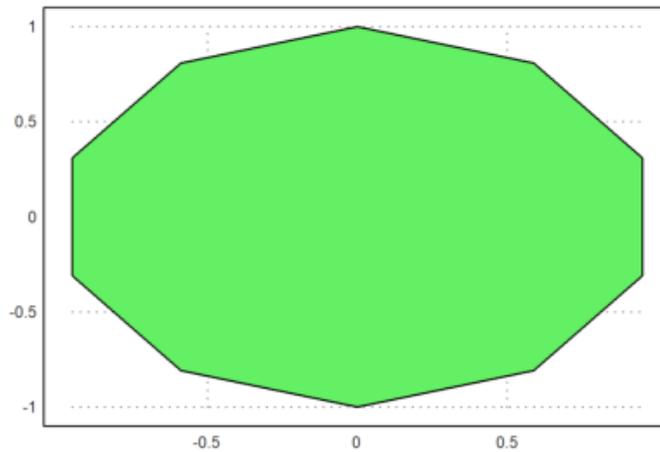
Terbentuk segi tujuh sama sisi.

```
>t=linspace(0,2pi,8);...
>plot2d(sin(t),cos(t),>filled,fillcolor=olive):
```



Terbentuk segi depalan sama sisi.

```
>t=linspace(0,2pi,10);...
>plot2d(sin(t),cos(t),>filled,fillcolor=lightgreen):
```

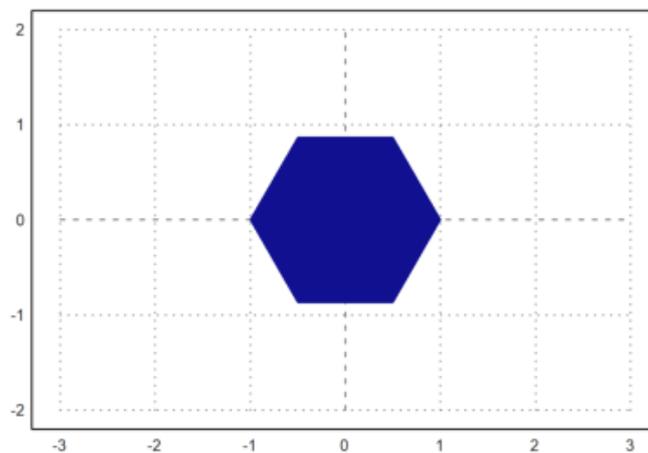


Terbentuk segi sepuluh sama sisi.

Parameter  $t$  dengan  $(0, 2\pi, n)$  dan fungsi  $\sin(t), \cos(t)$  akan menghasilkan segi banyak dengan banyak sisi  $n$  dan panjang sisi serta besar sudutnya sama.

Cara lain menggambar segi enam:

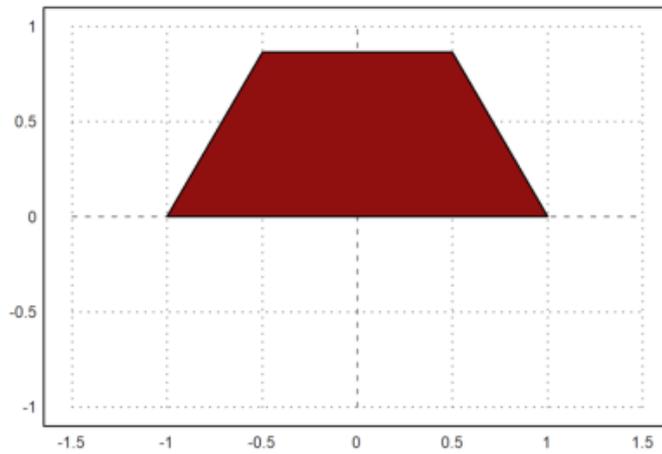
```
>t=linspace(0,3pi,9);...
>plot2d(cos(t),sin(t),>filled,style="#",fillcolor=blue,r=2):
```



$r$  diartikan sebagai jarak sumbu  $x$  positif, sumbu  $x$  negatif, sumbu  $y$  positif, dan sumbu  $y$  negatif dari titik pusat sejauh  $r$ .

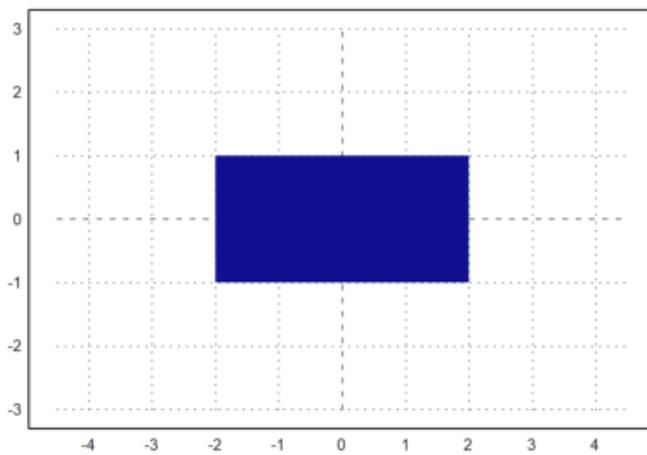
Contoh: segitiga sembarang, persegi panjang, trapesium, dan sebagainya.

```
>t=linspace(0,pi,3);...
>plot2d(cos(t),sin(t),>filled,fillcolor=red,r=1):
```



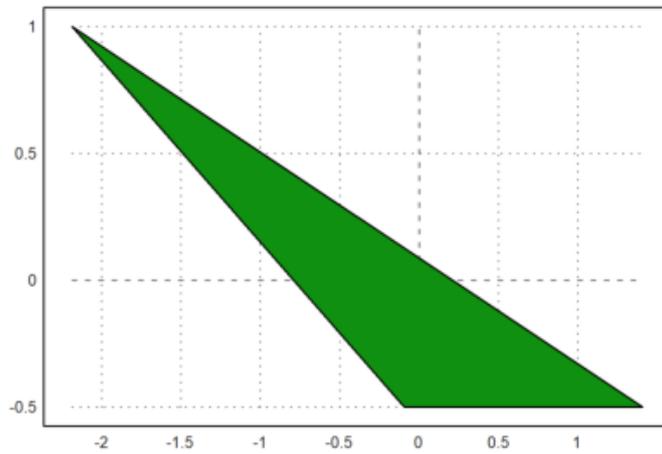
Terbentuk trapesium sama kaki.

```
>plot2d([-2,2,2,-2],[-1,-1,1,1],>filled,fillcolor=blue,<outline,r=3):
```



Dengan menghubungkan titik titik  $(-2,-1)$ ,  $(2,-1)$ ,  $(2,1)$ ,  $(-2,1)$  terbentuk bangun persegi panjang.

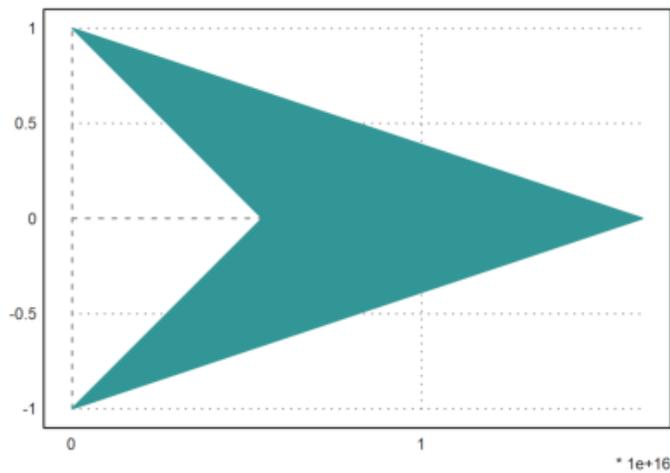
```
>t=linspace(0,2pi,3);...
>plot2d(tan(t+2),cos(t),>filled,fillcolor=green):
```



Terbentuk segitiga sembarang.

Contoh lain segi banyak tidak beraturan yaitu:

```
>t=linspace(0,2pi,4);...
>plot2d(tan(t),cos(t),>filled,style="#",fillcolor=cyan):
```



Contoh gambar di atas adalah kurva fungsi kompleks yang sama dengan sebelumnya, namun ada penambahan sintaks `<points` ke dalam `plot2d`, sehingga gambar yang dihasilkan berupa titik-titik.