Nama : Rofif Isyrof Nur Hafidz
Nim : 24030130030
Kelas: Pendidikan Mataematika C

Menggambar Grafik 2D dengan EMT

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

Plot Dasar

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

Pada contoh, kita hanya menggambar beberapa garis acak dalam berbagai warna. Untuk detail mengenai fungsi-fungsi ini, pelajari fungsi-fungsi inti dari EMT.

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



```
>reset;
```

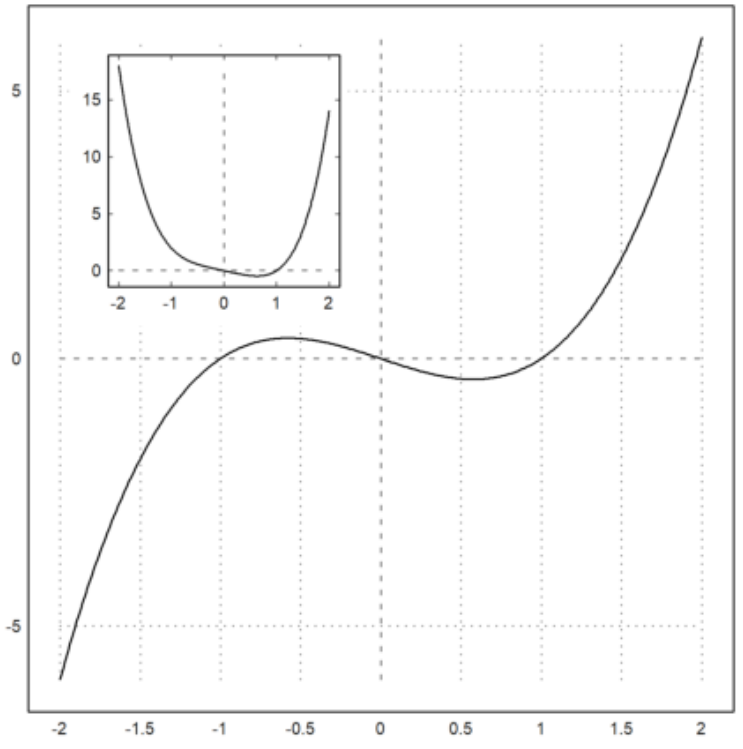
Anda harus menahan grafik, karena perintah plot() akan menghapus jendela plot.

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

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

Sebagai contoh lain, kita menggambar plot sebagai inset dalam plot lain. Hal 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 hal ini sesuai kebutuhan. Perhatikan bahwa kita menyimpan dan mengembalikan jendela penuh, dan menahan plot saat ini ketika kita membuat inset.

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




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

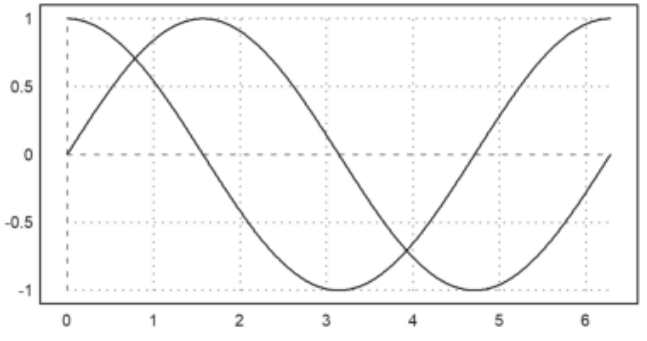
Plot dengan beberapa angka dicapai dengan cara yang sama. Ada fungsi utility figure() untuk ini.

Aspek Plot

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

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

```
>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 bentuk 2D, baik untuk data maupun fungsi. EMT menggunakan fungsi plot2d. Fungsi ini dapat memplot fungsi dan data.

Dimungkinkan untuk memplot di Maxima menggunakan Gnuplot atau di Python menggunakan Math Plot Lib.

Euler dapat memplot plot 2D dari

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

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

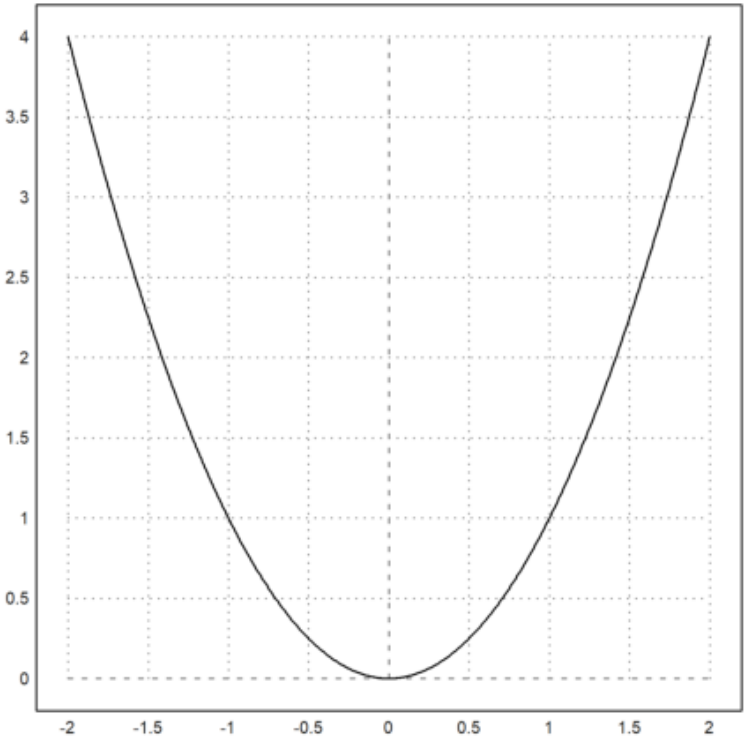
Plot Ekspresi atau Variabel

Ekspresi tunggal dalam ”x” (misalnya ”4*xˆ2”) atau nama fungsi (misalnya ”f”) menghasilkan grafik fungsi.

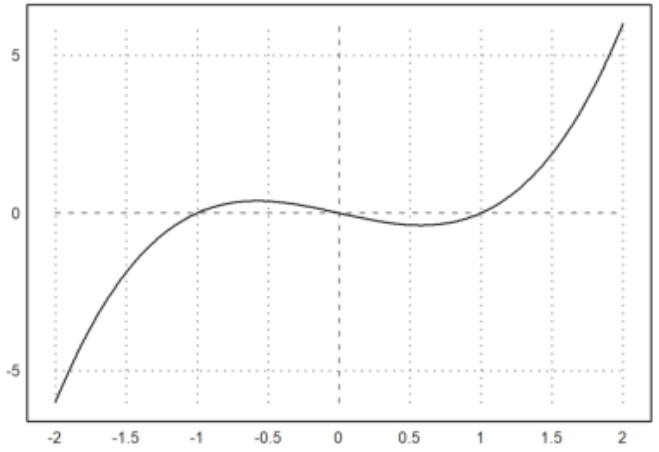
Berikut ini adalah contoh paling dasar, yang menggunakan rentang default dan menetapkan rentang y yang tepat agar sesuai dengan plot fungsi.

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

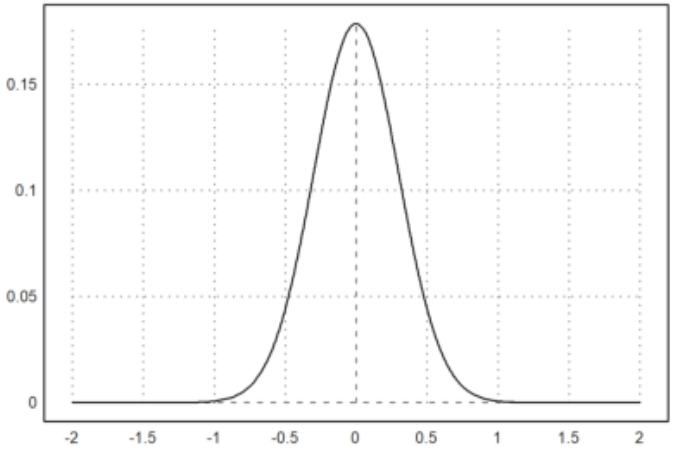
```
>plot2d("x^2"):
```




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




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

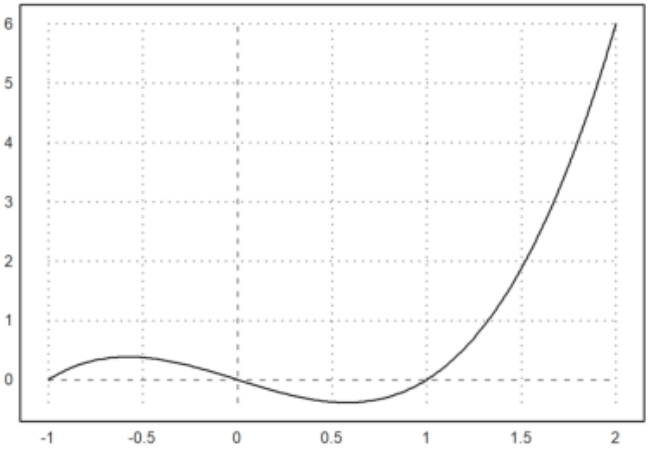


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

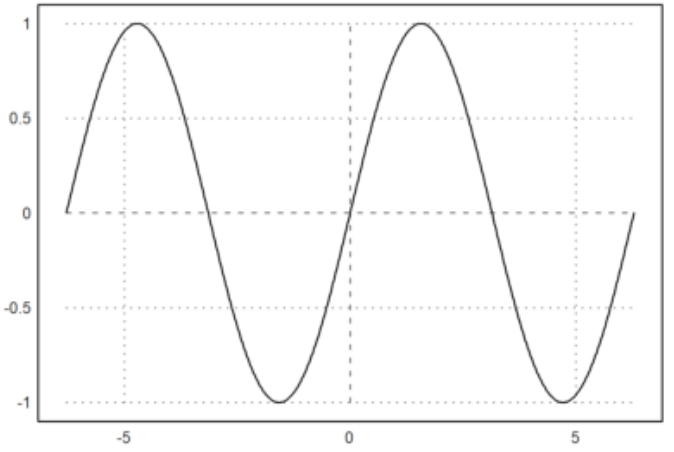
Rentang plot ditetapkan dengan parameter yang ditetapkan berikut ini

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

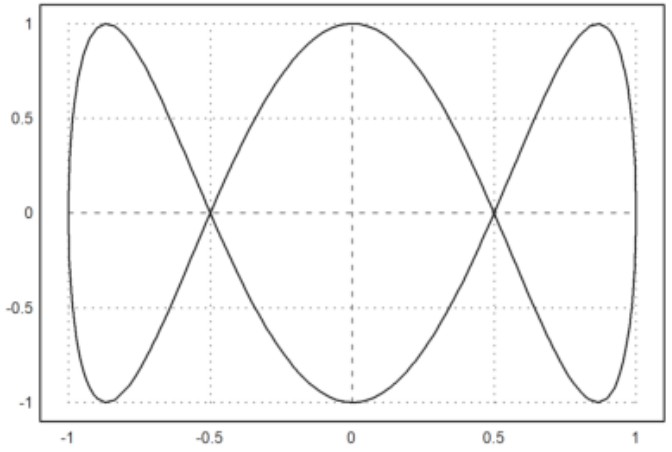
```
>plot2d("x^3-x",-1,2):
```




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

Dalam opsi, plot dapat diatur untuk muncul

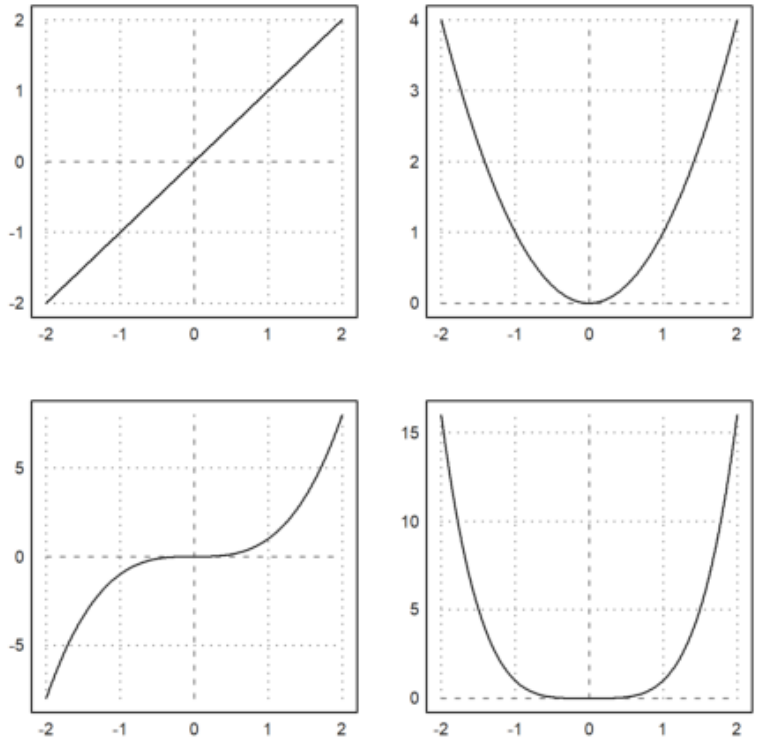
- di jendela terpisah yang dapat diubah ukurannya,
- di jendela buku catatan.

Lebih banyak gaya yang dapat dicapai dengan perintah plot tertentu.

Dalam hal apa pun, tekan tombol tabulator untuk melihat plot, jika disembunyikan.

Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure(). Pada contoh, kita memplot xˆ1 sampai xˆ4 ke dalam 4 bagian jendela. figure(0) akan mereset jendela default.

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



Alternatif untuk tanda titik dua adalah perintah insimg(lines), yang menyisipkan plot yang menempa

Dalam opsi, plot dapat diatur untuk muncul

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

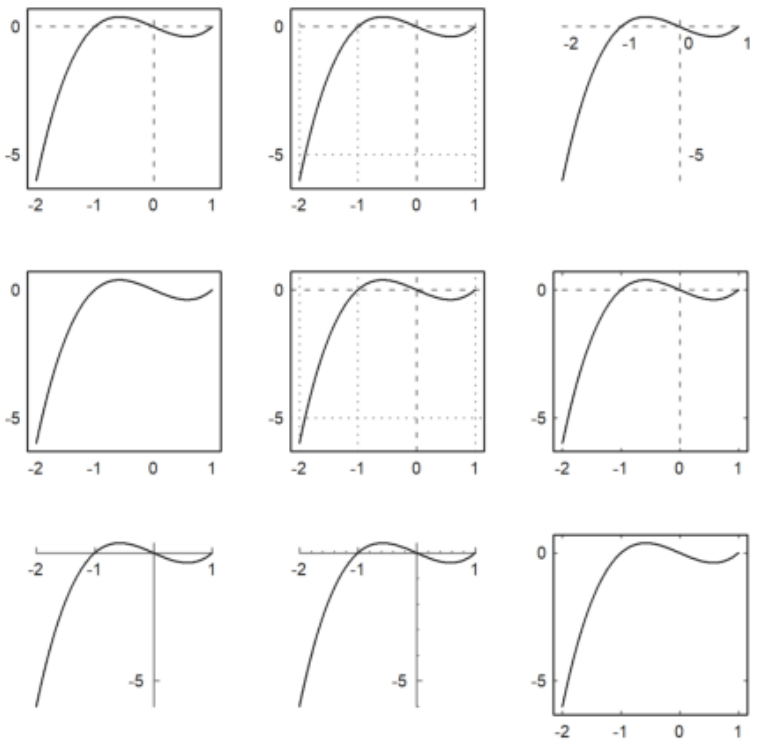
Lebih banyak gaya yang dapat dicapai dengan perintah plot tertentu.

```
Dalam hal apa pun, tekan tombol tabulator untuk melihat plot, jika disembunyikan.

Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure(). Pada contoh, kita memplot
```

Pada plot2d(), terdapat beberapa gaya alternatif yang tersedia dengan grid=x. Sebagai gambaran umum, kami menampilkan berbagai gaya grid dalam satu gambar (lihat di bawah ini untuk perintah figure()). Gaya grid=0 tidak disertakan. Gaya ini tidak menampilkan grid dan frame.

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

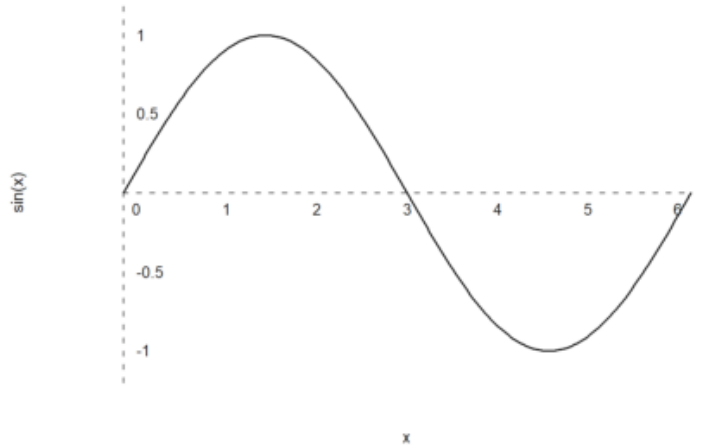


Jika argumen untuk plot2d() adalah sebuah ekspresi yang diikuti oleh empat angka, angka-angka ini adalah rentang x dan y untuk plot.

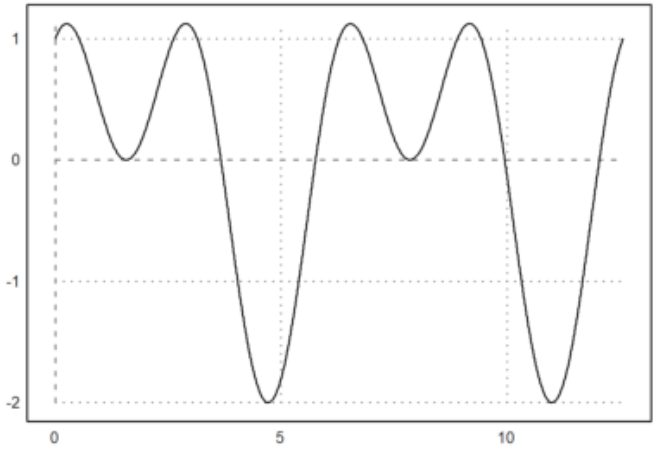
Sebagai alternatif, a, b, c, d dapat ditentukan sebagai parameter yang ditetapkan sebagai a=... dst.

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

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




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

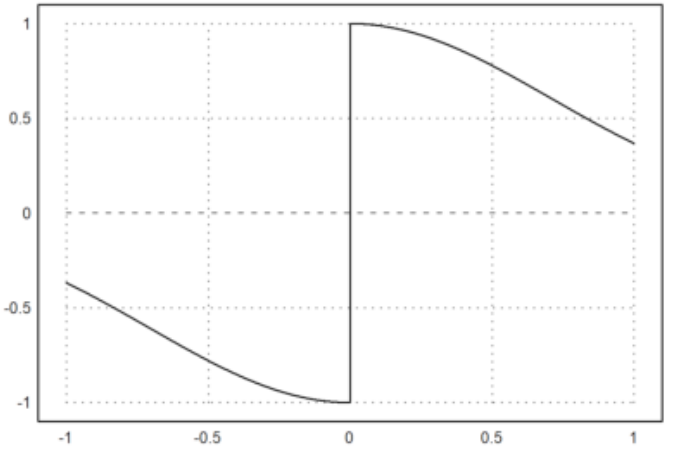


Gambar yang dihasilkan dengan menyisipkan plot ke dalam jendela teks disimpan dalam direktori yang sama dengan notebook, secara default dalam subdirektori bernama ”images”. Gambar-gambar tersebut juga digunakan oleh ekspor HTML.

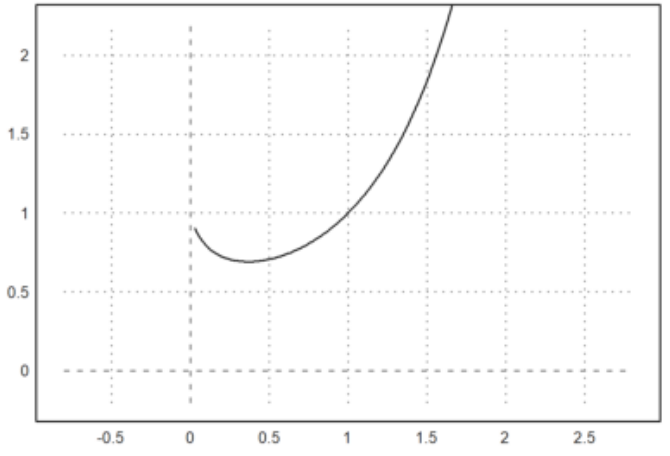
Anda cukup menandai gambar mana saja dan menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, Anda juga dapat mengekspor grafik saat ini dengan fungsi-fungsi pada menu File.

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

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

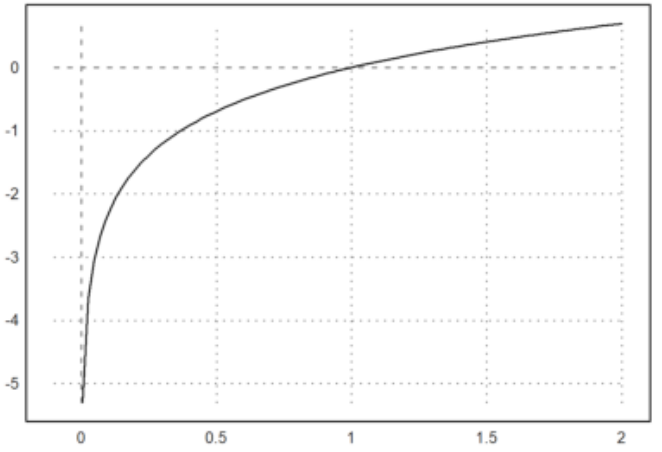



```
>plot2d("x^x",r=1.2,cx=1,cy=1):
```



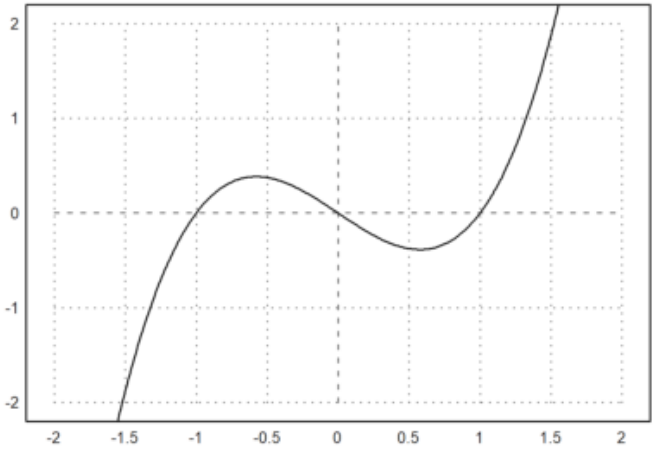
Perhatikan bahwa x^x tidak didefinisikan untuk x<=0. Fungsi plot2d menangkap kesalahan ini, dan mulai memplot segera setelah fungsi didefinisikan. Hal ini berlaku untuk semua fungsi yang mengembalikan NAN di luar jangkauan definisinya.

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

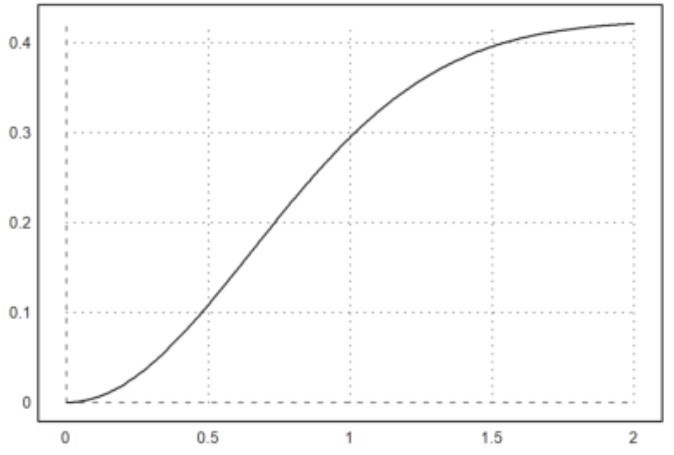


Parameter square=true (atau >square) memilih rentang y secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot.

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

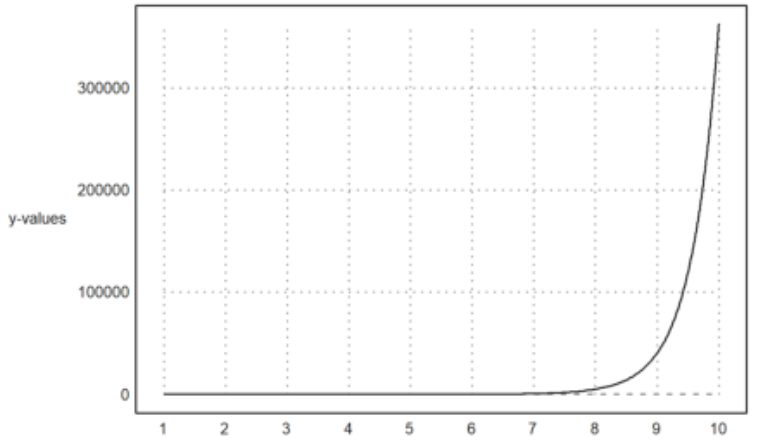



```
>plot2d(''integrate("sin(x)*exp(-x^2)",0,x)'',0,2): // plot integral
```



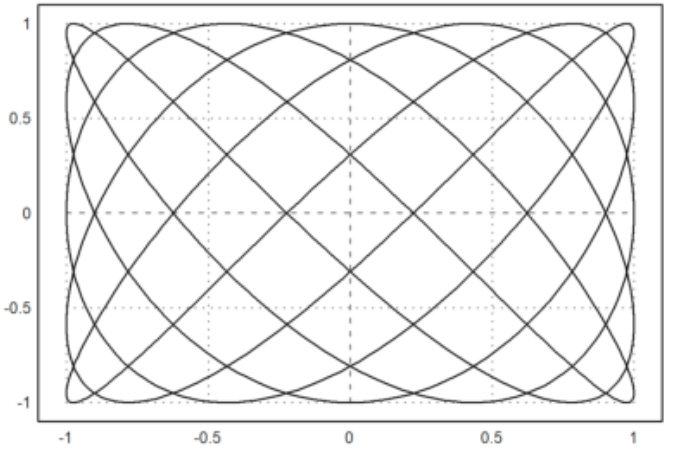
Jika Anda membutuhkan lebih banyak ruang untuk label-y, panggil shrinkwindow() dengan parameter lebih kecil, atau tetapkan nilai positif untuk ”lebih kecil” pada plot2d().

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

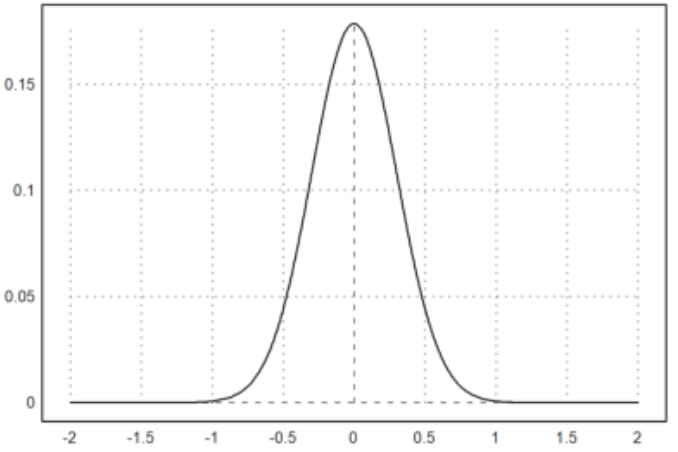


Ekspresi simbolik juga dapat digunakan, karena disimpan sebagai ekspresi string sederhana.

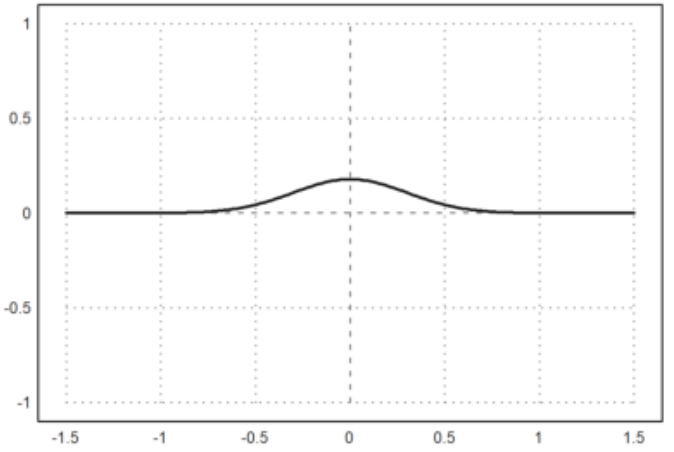
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



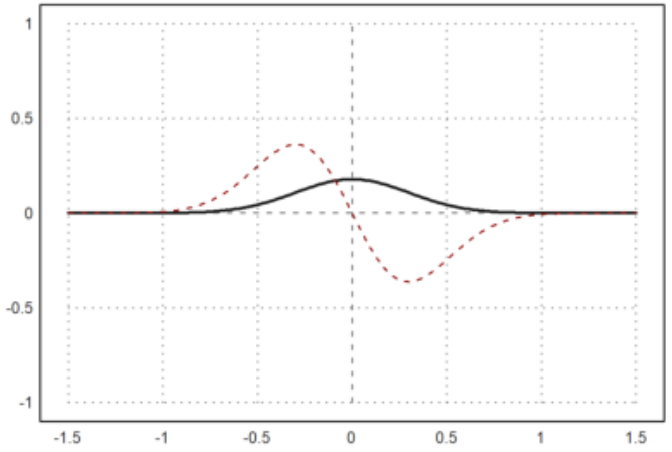

```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression
>plot2d(expr,-2,2): // plot from -2 to 2
```



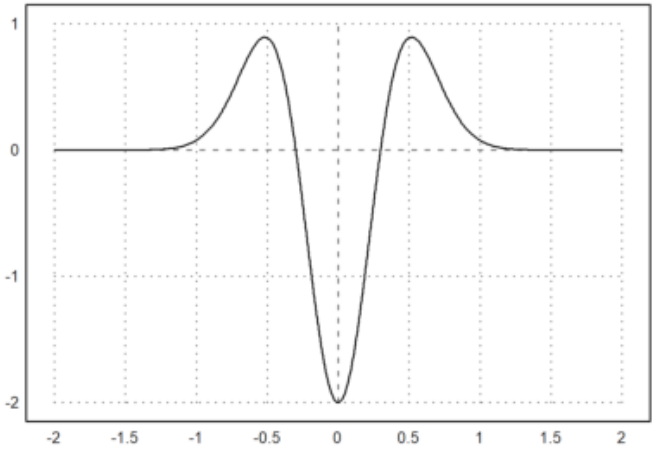

```
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)
```



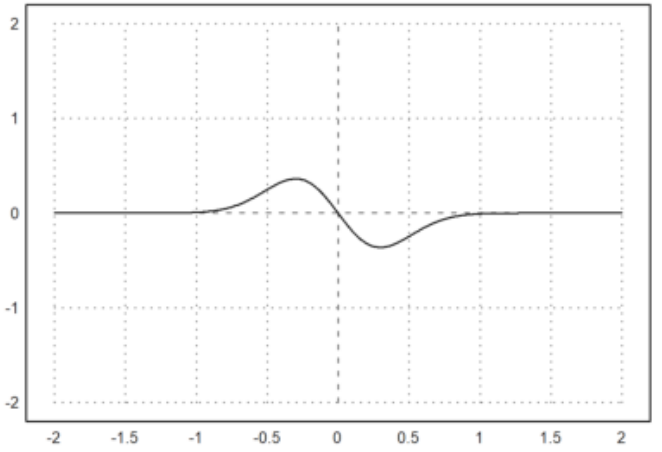

```
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot
```



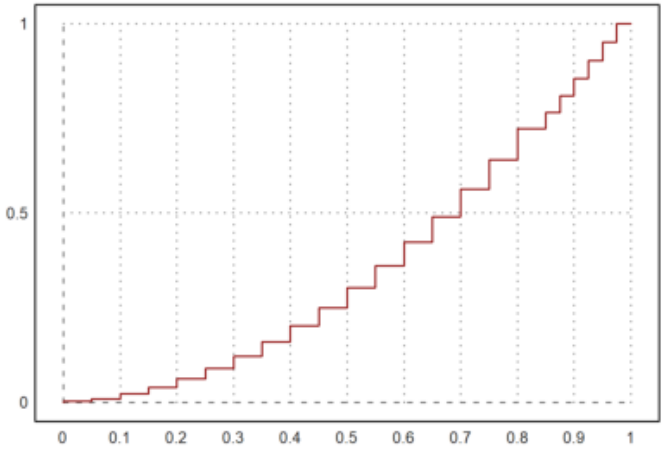

```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle
```



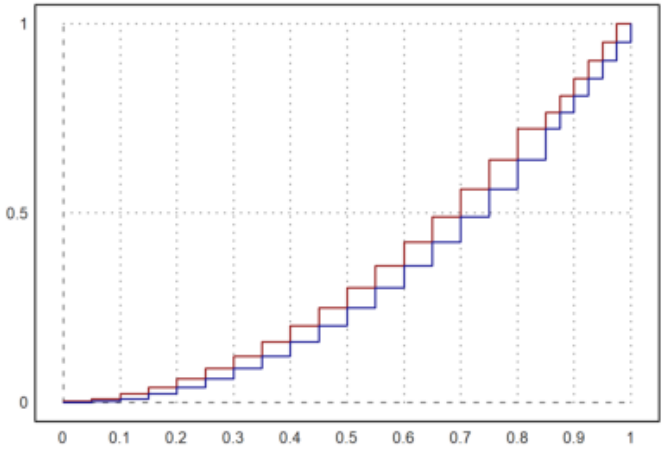

```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square
```




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




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

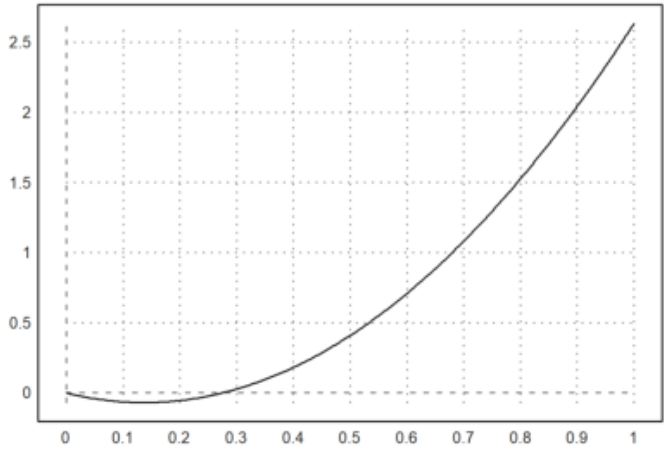

1
0.5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Fungsi dalam satu Parameter

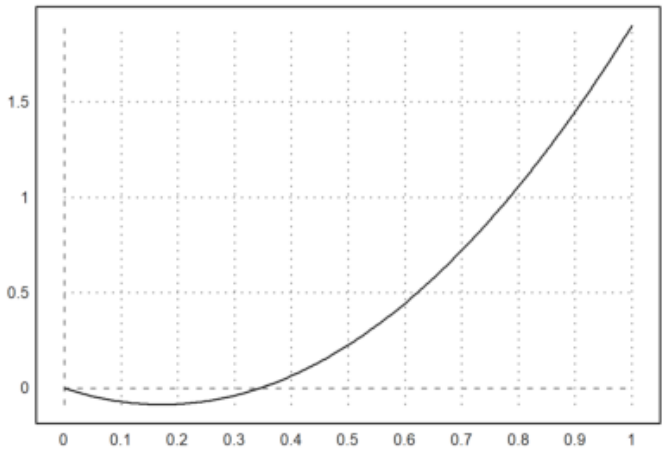
Fungsi plot yang paling penting untuk plot planar adalah plot2d(). Fungsi ini diimplementasikan dalam bahasa Euler dalam file ”plot.e”, yang dimuat pada awal program.

Berikut adalah beberapa contoh penggunaan fungsi. Seperti biasa dalam EMT, fungsi yang bekerja untuk fungsi atau ekspresi lain, Anda dapat mengoper parameter tambahan (selain x) yang bukan variabel global ke fungsi dengan parameter titik koma atau dengan koleksi panggilan.

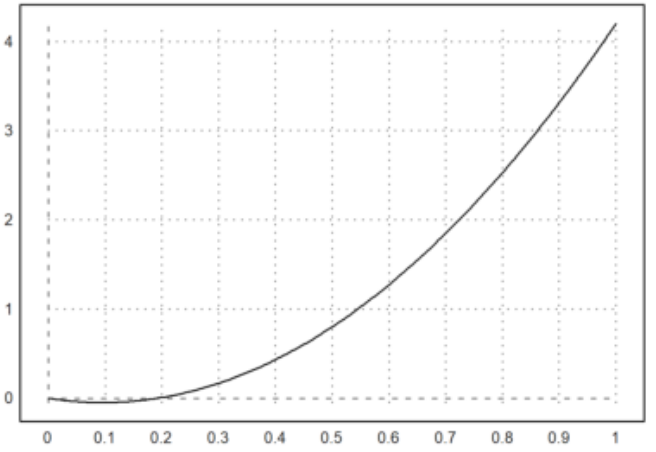
```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
>a=0.3; plot2d("f",0,1;a): // plot with a=0.3
```



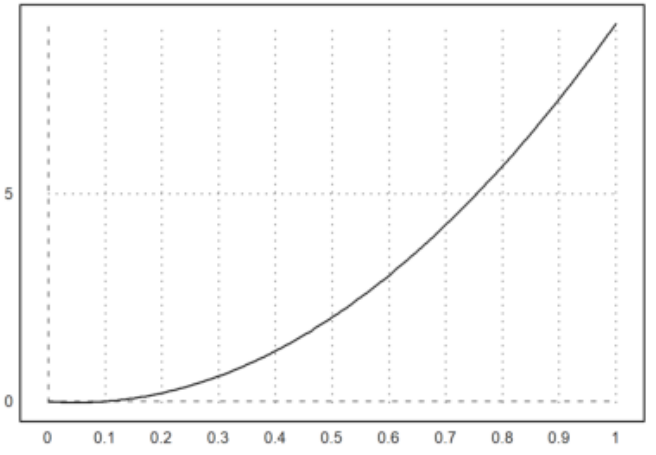
```
>plot2d("f",0,1;0.4): // plot with a=0.4
```



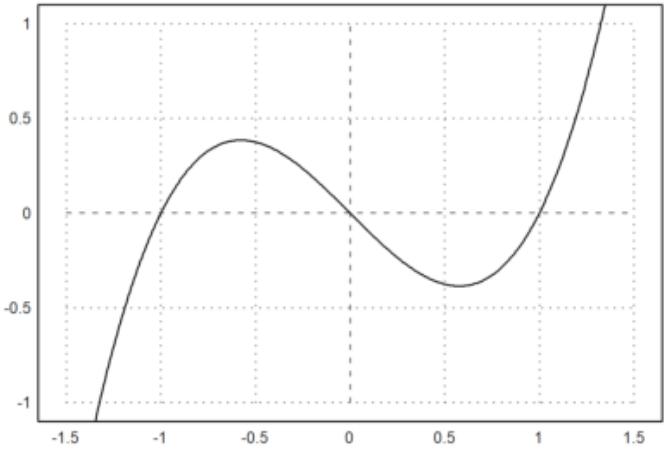

```
>plot2d({{"f",0.2}},0,1): // plot with a=0.2
```




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




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



Berikut ini adalah ringkasan dari fungsi yang diterima

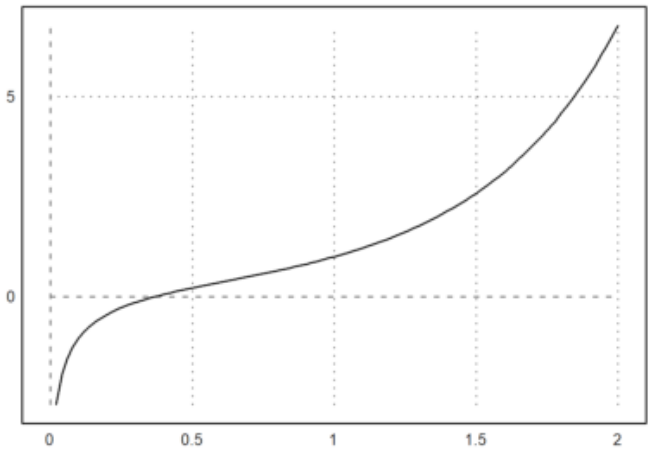
- ekspresi atau ekspresi simbolik dalam x
- fungsi atau fungsi simbolik dengan nama sebagai "f"
- fungsi-fungsi simbolik hanya dengan nama f

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

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

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

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

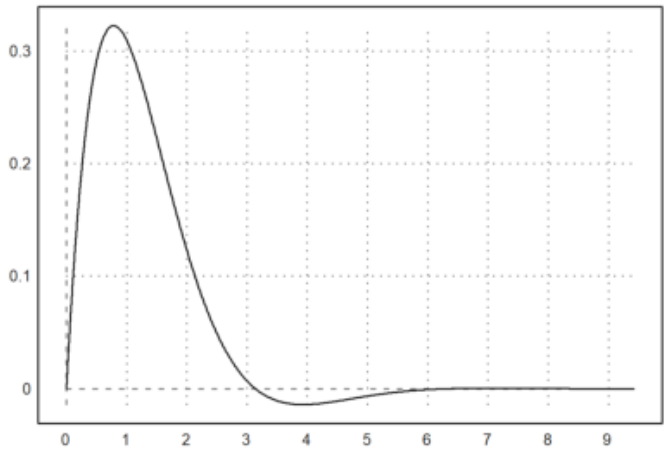


Tentu saja, untuk ekspresi atau ungkapan simbolik, nama variabel sudah cukup untuk memplotnya.

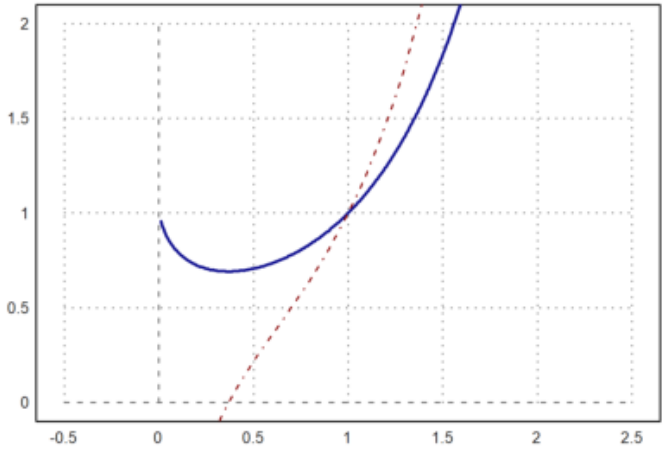
```
>expr &= sin(x)*exp(-x)
```

$$\sin(x)\, e^{-x}$$

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



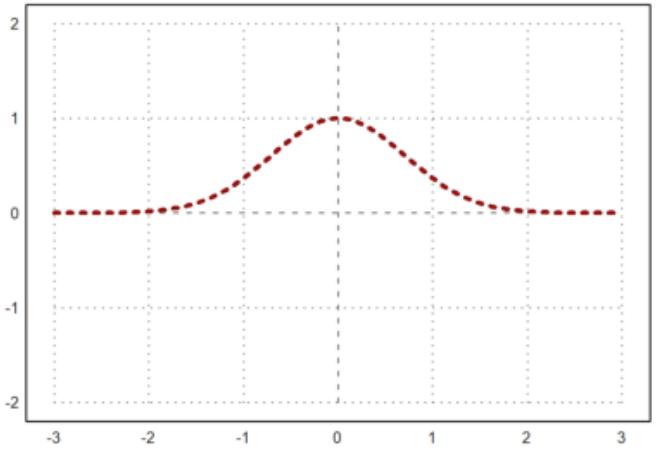
Untuk gaya garis, ada berbagai pilihan.

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

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

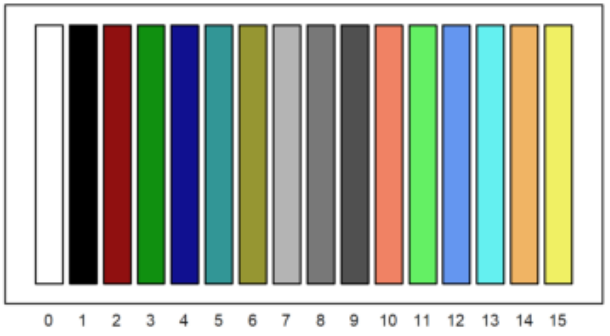
- 0..15: indeks warna default.
- konstanta warna: putih, hitam, merah, hijau, biru, cyan, zaitun, abu-abu muda, abu-abu, abu-abu tua, oranye, hijau muda, pirus, biru muda, oranye muda, kuning
- rgb (merah, hijau, biru): parameter adalah real dalam [0,1].

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



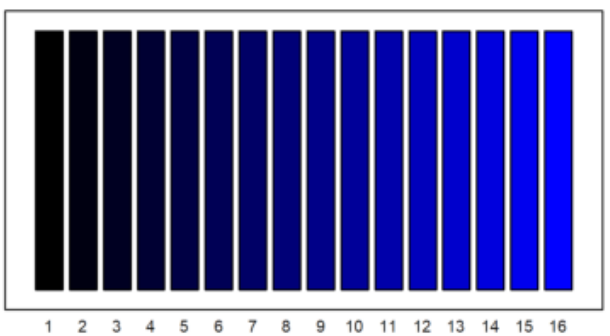
Berikut ini adalah pemandangan warna EMT yang sudah ditetapkan sebelumnya.

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



Tetapi Anda bisa menggunakan warna apa pun.

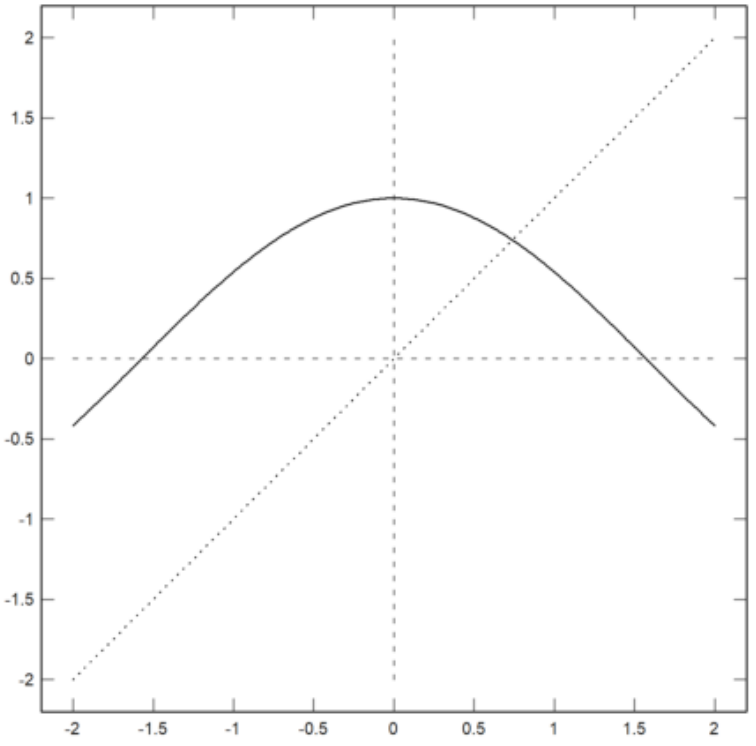
```
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```



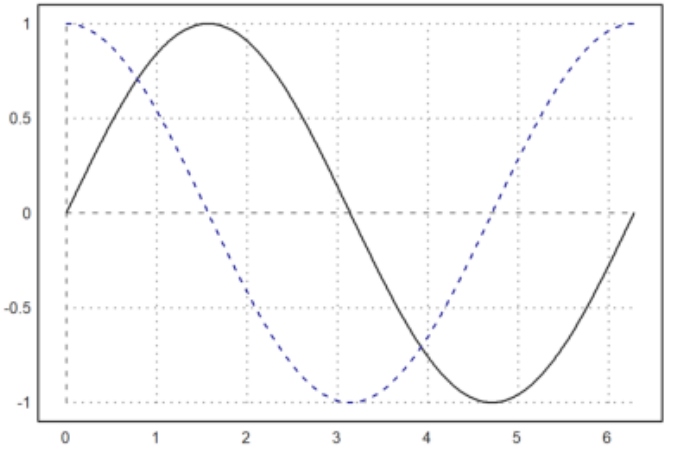
Menggambar beberapa kurva pada bidang koordinat yang sama

Memplot lebih dari satu fungsi (beberapa fungsi) ke dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu caranya adalah dengan menggunakan >add untuk beberapa pemanggilan ke plot2d secara bersamaan, kecuali pemanggilan pertama. Kita telah menggunakan fitur ini pada contoh di atas.

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

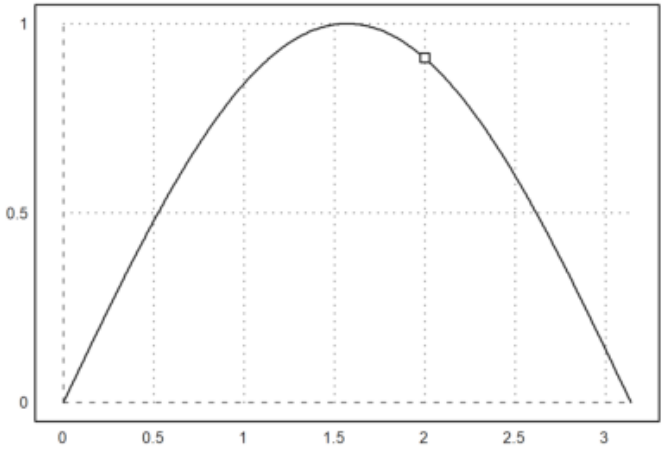



```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



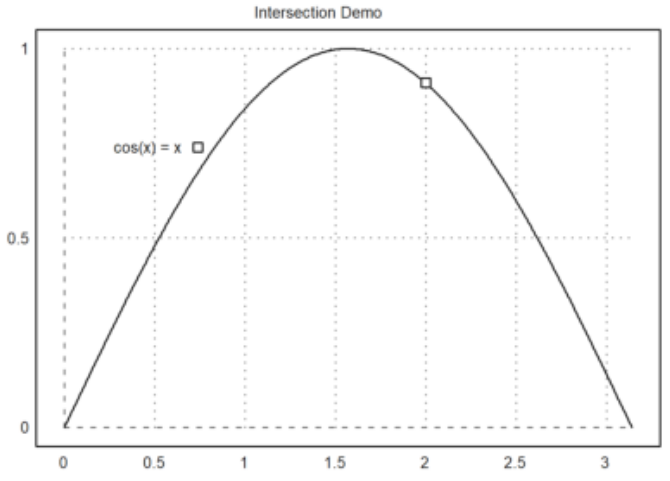
Salah satu kegunaan >add adalah untuk menambahkan titik pada kurva.

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



Kami menambahkan titik perpotongan dengan label (pada posisi "cl" untuk kiri tengah), dan menyisipkan hasilnya ke dalam buku catatan. 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):
```



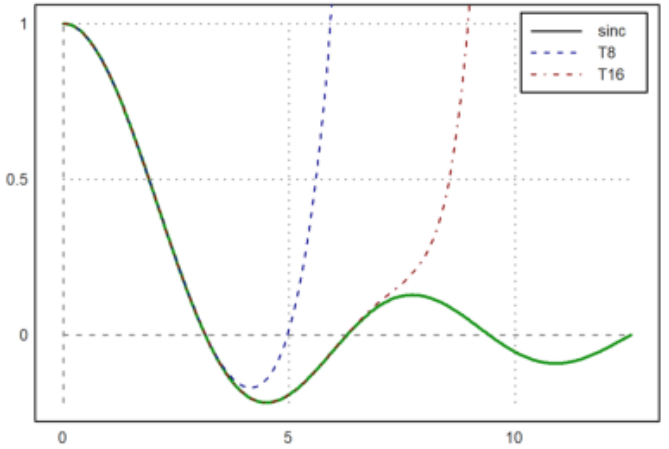
Dalam demo berikut ini, kami memplot fungsi sinc(x)=sin(x)/x dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung ekspansi ini menggunakan Maxima melalui ekspresi simbolik.
Plot ini dilakukan dalam perintah multi-baris berikut dengan tiga pemanggilan plot2d(). Perintah kedua dan ketiga memiliki set flag >add, yang membuat plot menggunakan rentang sebelumnya.

Kami menambahkan sebuah kotak label yang menjelaskan fungsi-fungsi tersebut.

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

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

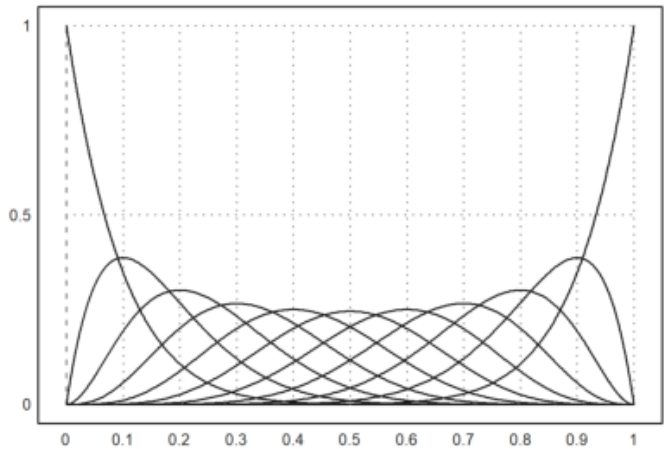
```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
>  plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
>  plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-.-"); ...
>  labelbox(["sinc","T8","T16"],styles=["-","--","-.-"], ...
>    colors=[black,blue,red]):
```



Pada contoh berikut, kami menghasilkan Polinomial Bernstein.

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

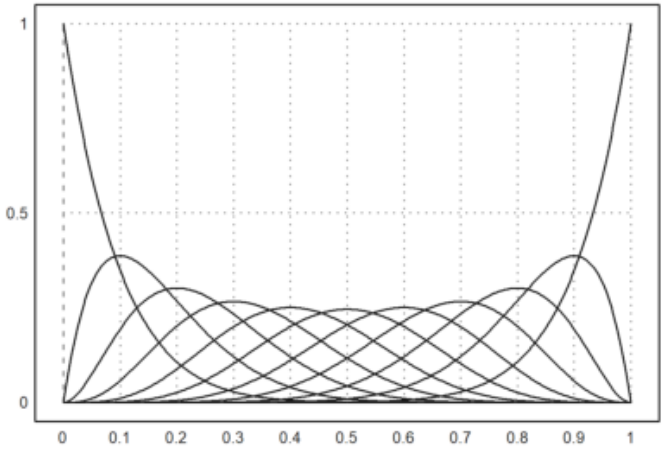
```
>plot2d("(1-x)^10",0,1); // plot first function
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;
>insimg;
```



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

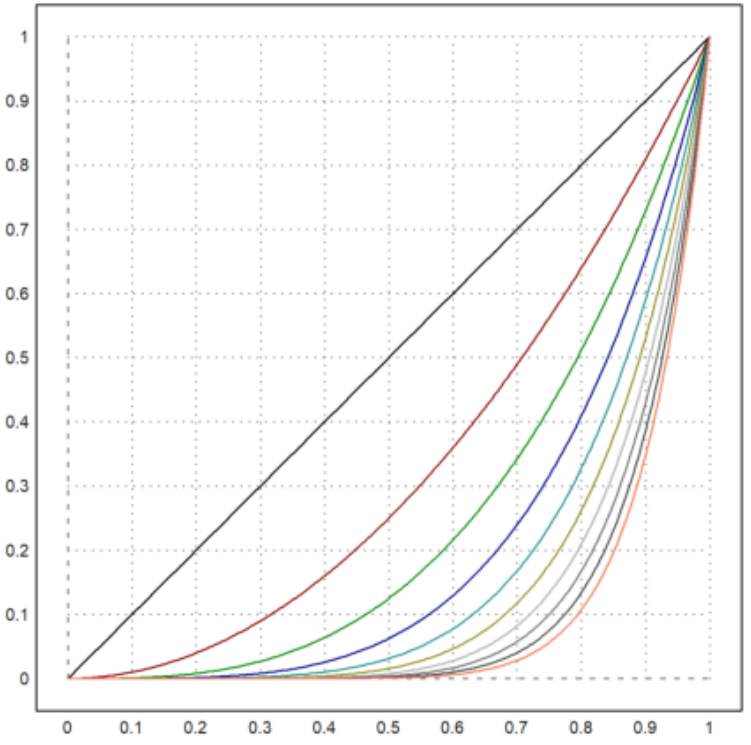
Kita membuat sebuah matriks nilai dengan satu Polinomial Bernstein di setiap baris. Untuk ini, kita cukup menggunakan vektor kolom i. Lihatlah pengantar tentang bahasa matriks untuk mempelajari lebih lanjut.

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



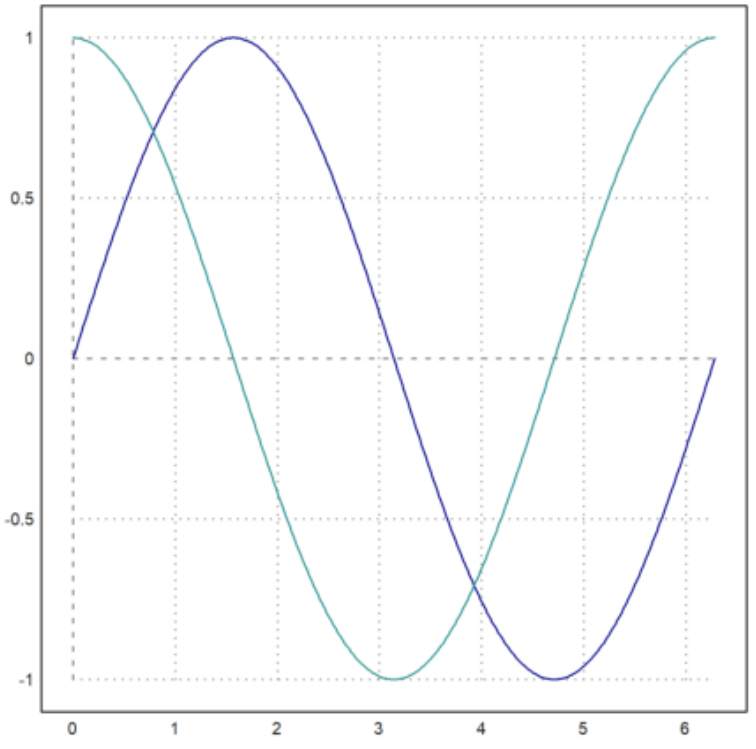
Perhatikan bahwa parameter warna dapat berupa vektor. Kemudian setiap warna digunakan untuk setiap baris matriks.

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

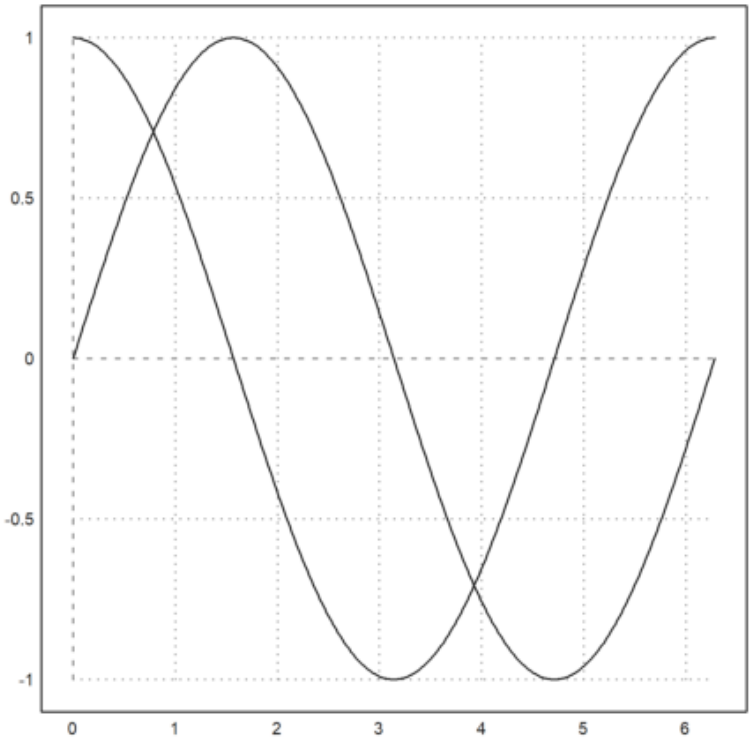


Metode lainnya adalah menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan larik warna, larik gaya, dan larik ketebalan dengan panjang yang sama.

```
>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 dengan menggunakan makelist() dan mxm2str().

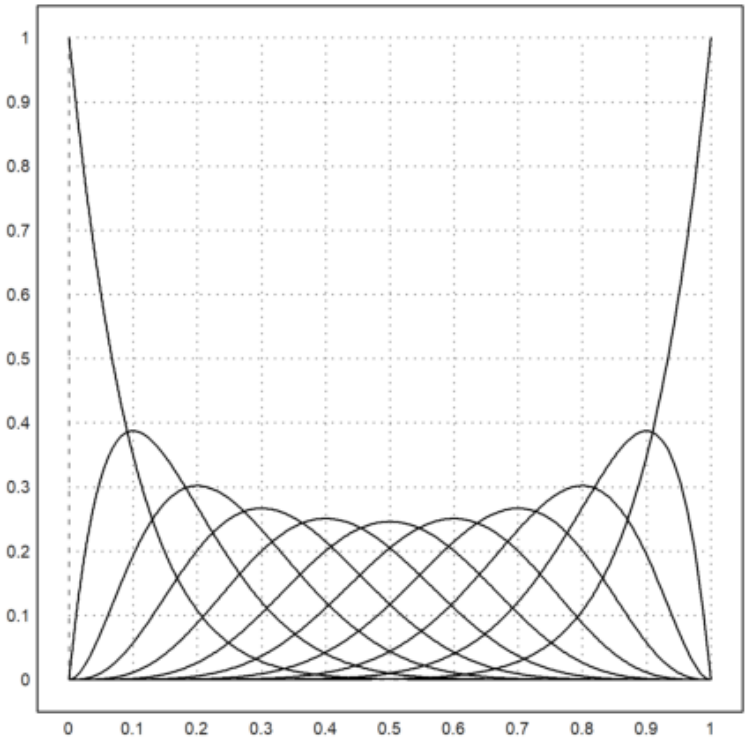
```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // make list
```

```
              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
```

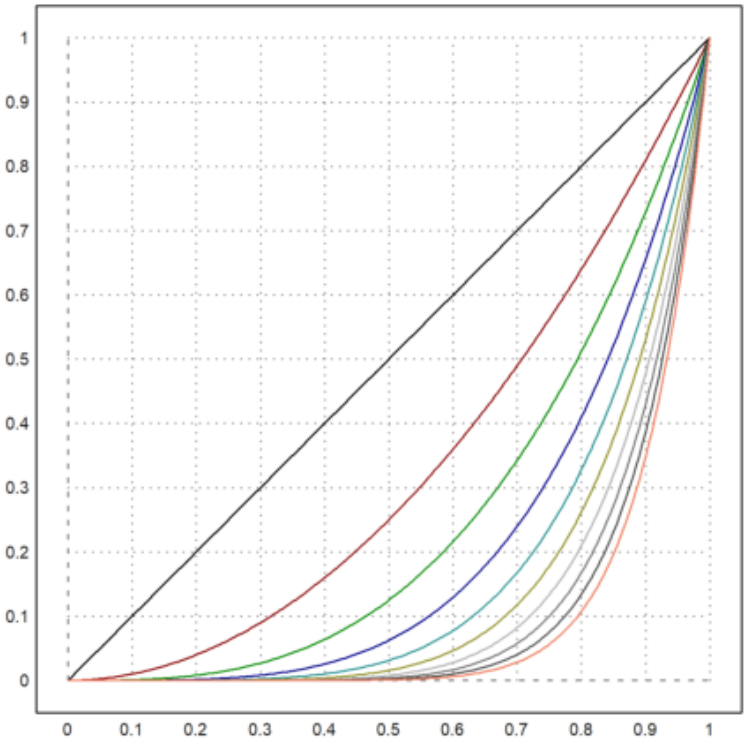


Alternatif lain adalah dengan menggunakan bahasa matriks Euler.

Jika sebuah ekspresi menghasilkan sebuah 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 sebuah larik warna ditambahkan, maka akan digunakan untuk setiap baris plot.

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

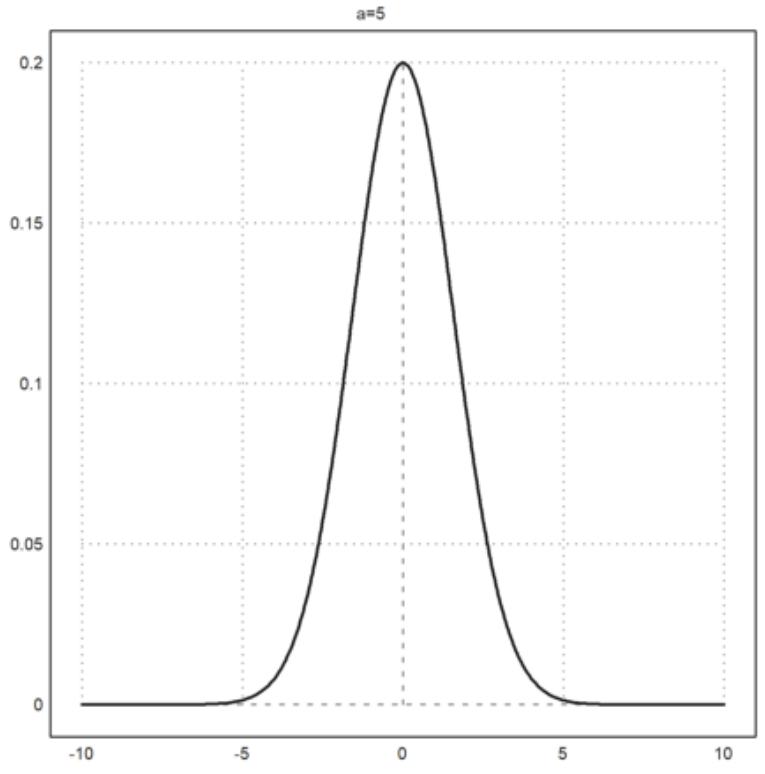


Ekspresi dan fungsi satu baris dapat melihat variabel global.

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

Berhati-hatilah untuk meletakkan semua parameter yang diberikan di akhir perintah plot2d. Pada contoh ini kita mengoper a=5 ke fungsi f, yang kita plot dari -10 ke 10.

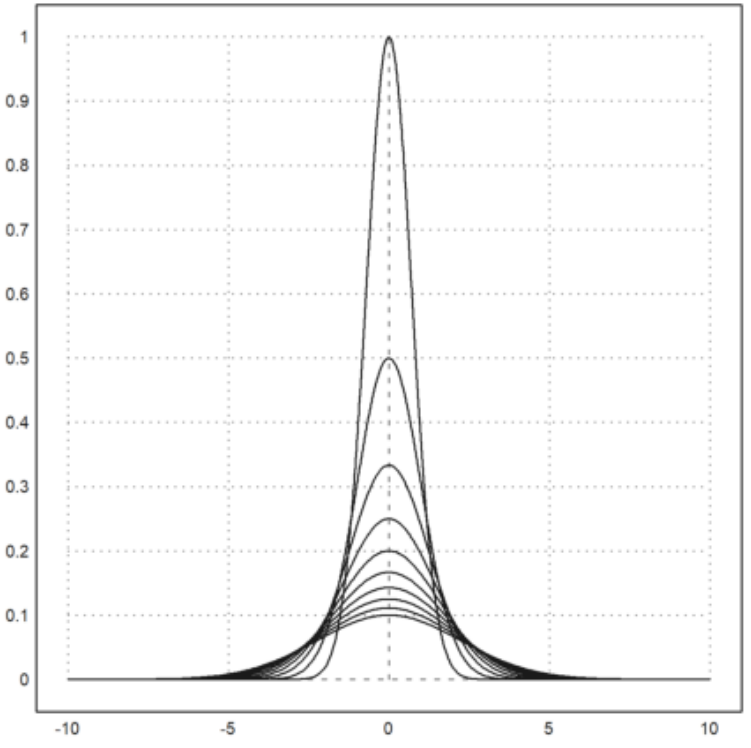
```
>function f(x,a) := 1/a*exp(-x^2/a); ...
>plot2d("f",-10,10;5,thickness=2,title="a=5"):
```



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

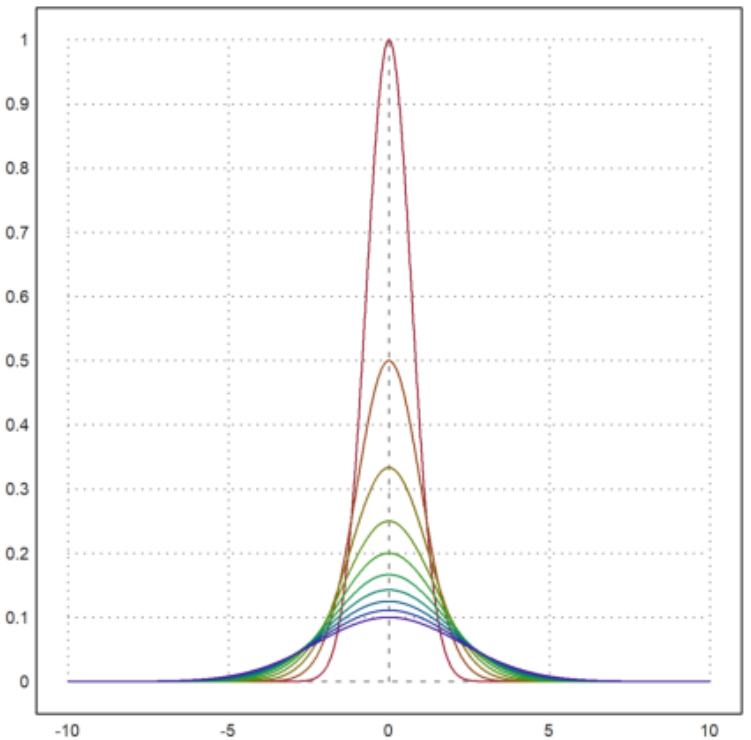
Pada contoh berikut, kita menggunakan perulangan untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman perulangan).

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



Kita dapat mencapai hasil yang sama dengan cara berikut menggunakan bahasa matriks EMT. Setiap baris dari matriks 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)):
```



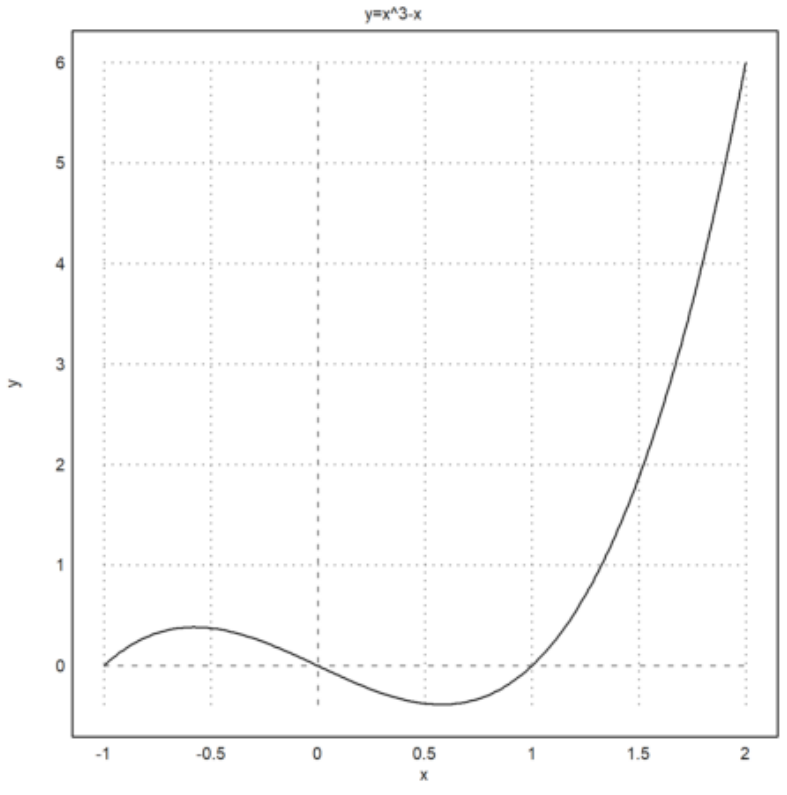
Label Teks

Dekorasi sederhana dapat berupa

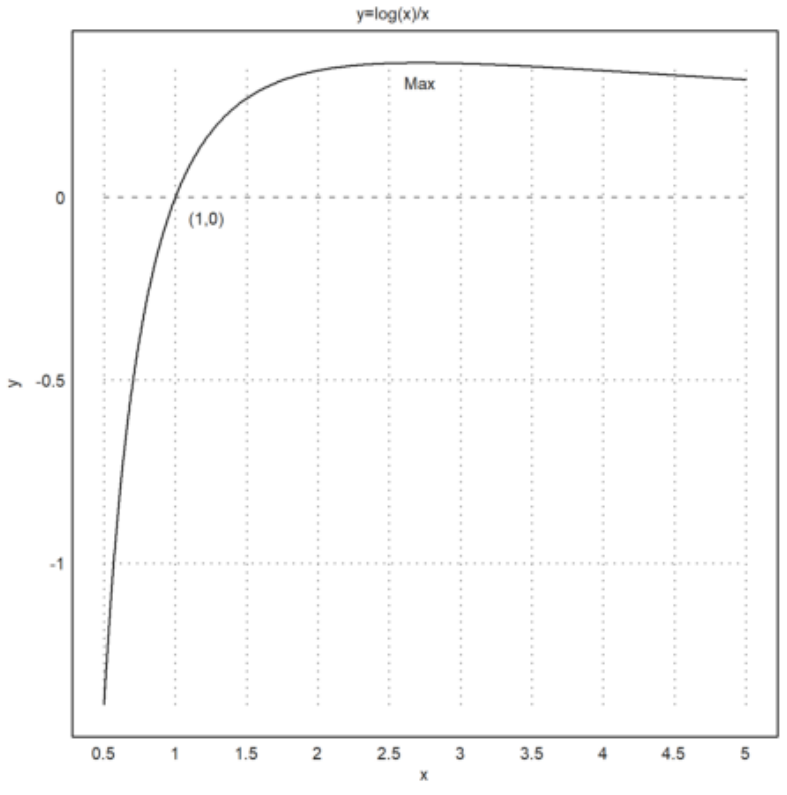
- sebuah judul dengan title=”...”
- label x dan y dengan xl=”...”, yl=”...”
- label teks lain dengan label(”...”,x,y)

Perintah label akan memplotkan ke dalam plot saat ini pada koordinat plot (x,y). Perintah ini dapat menerima sebuah argumen posisi.

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



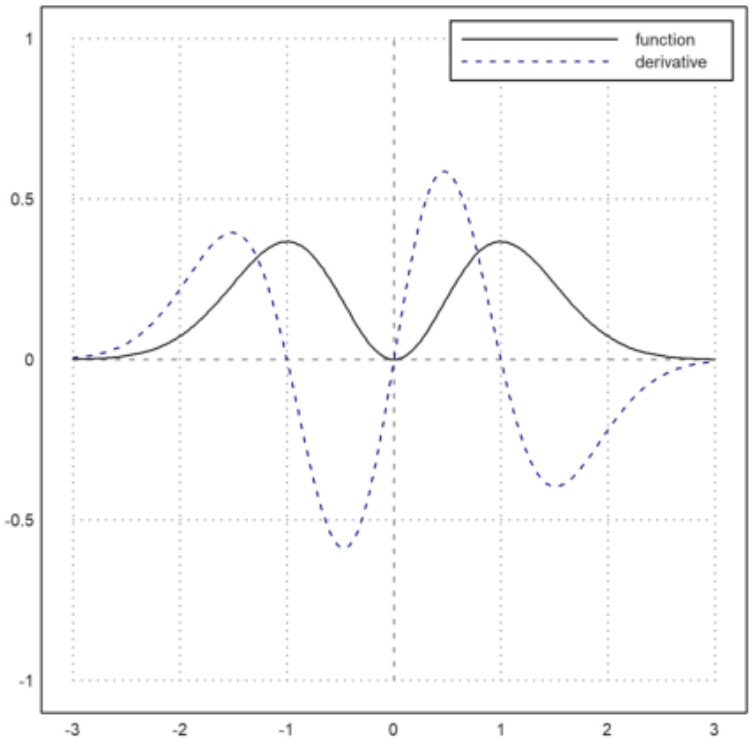

```
>expr := "log(x)/x"; ...
>  plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
>  label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```



Ada juga fungsi labelbox(), yang dapat menampilkan fungsi dan teks. Fungsi ini membutuhkan 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):
```

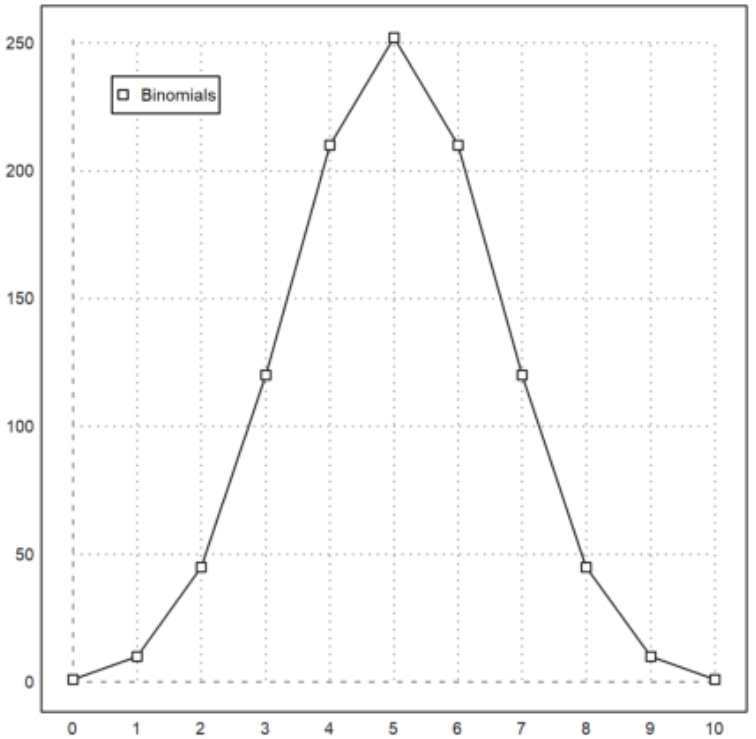


Kotak tersebut berlabuh di kanan atas secara default, tetapi >kiri menambatkannya di kiri atas. Anda dapat memindahkannya ke tempat mana pun yang Anda suka. Posisi jangkar adalah sudut kanan atas kotak, dan angkanya adalah pecahan dari ukuran jendela grafik. Lebarnya otomatis.

Untuk plot titik, kotak label juga dapat digunakan. Tambahkan parameter >titik, atau vektor bendera, satu untuk setiap label.

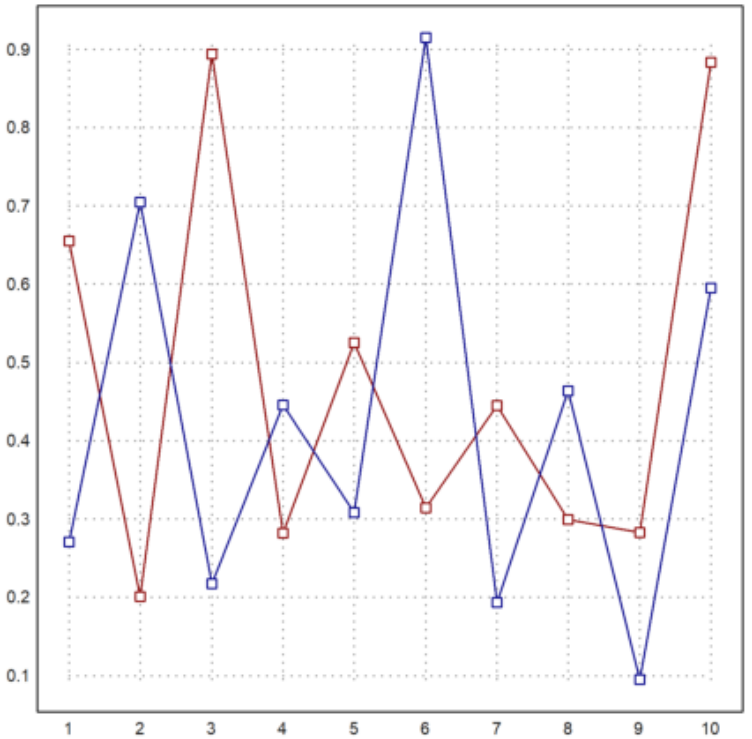
Pada contoh berikut, hanya ada satu fungsi. Jadi kita dapat menggunakan string dan bukan vektor dari string. Kita mengatur warna teks menjadi hitam untuk contoh ini.

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



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

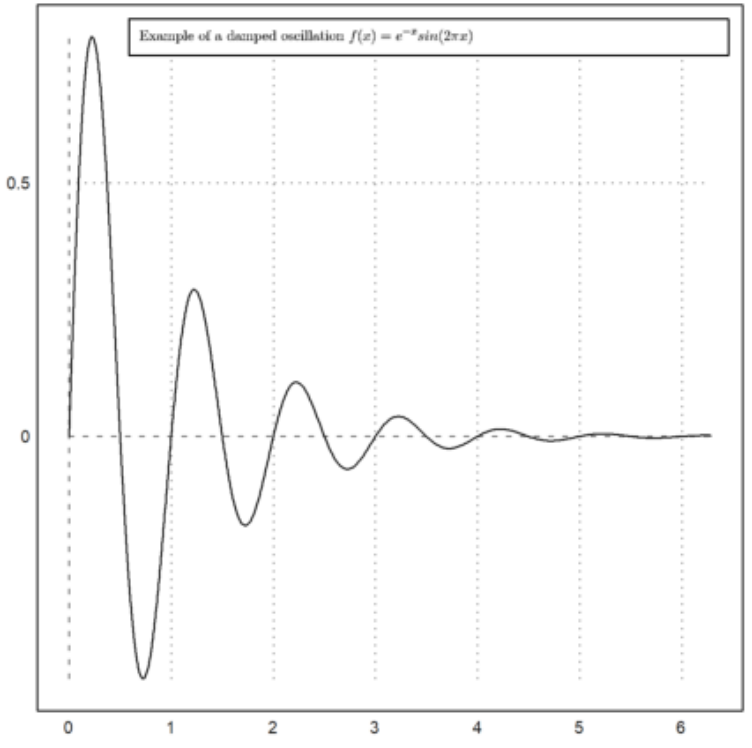
```
>statplot(1:10,random(2,10),color=[red,blue]):
```



Fitur yang serupa adalah fungsi textbox().

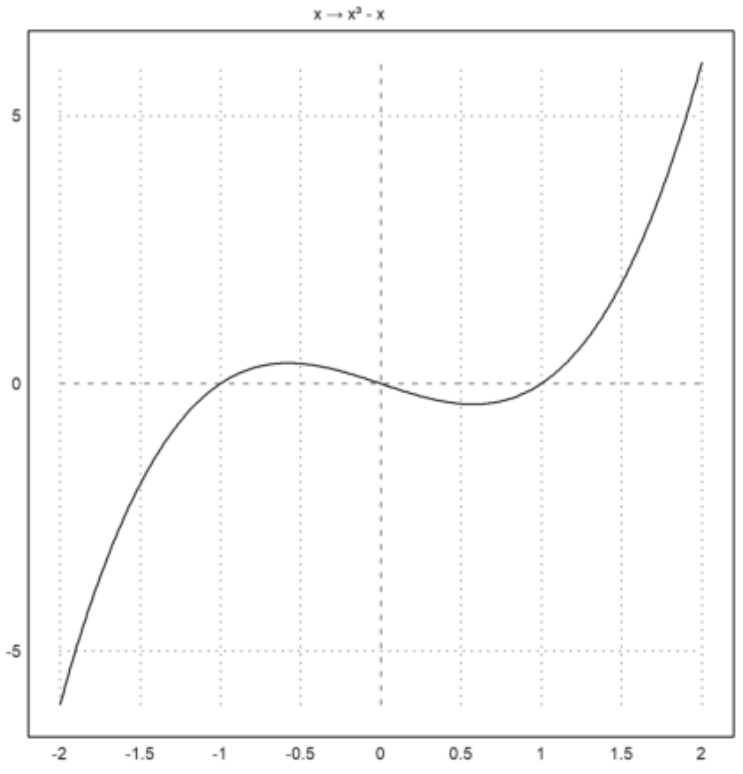
Lebarnya secara default adalah lebar maksimal baris teks. Tetapi dapat diatur oleh pengguna juga.

```
>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):
```



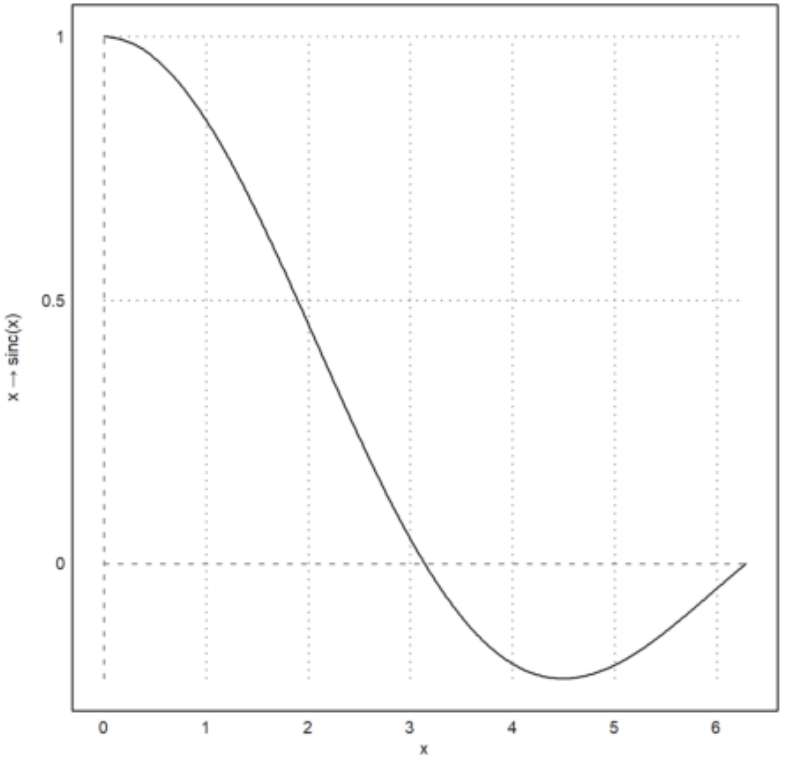
Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaks EMT untuk mengetahui lebih lanjut tentang string Unicode).

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



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

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

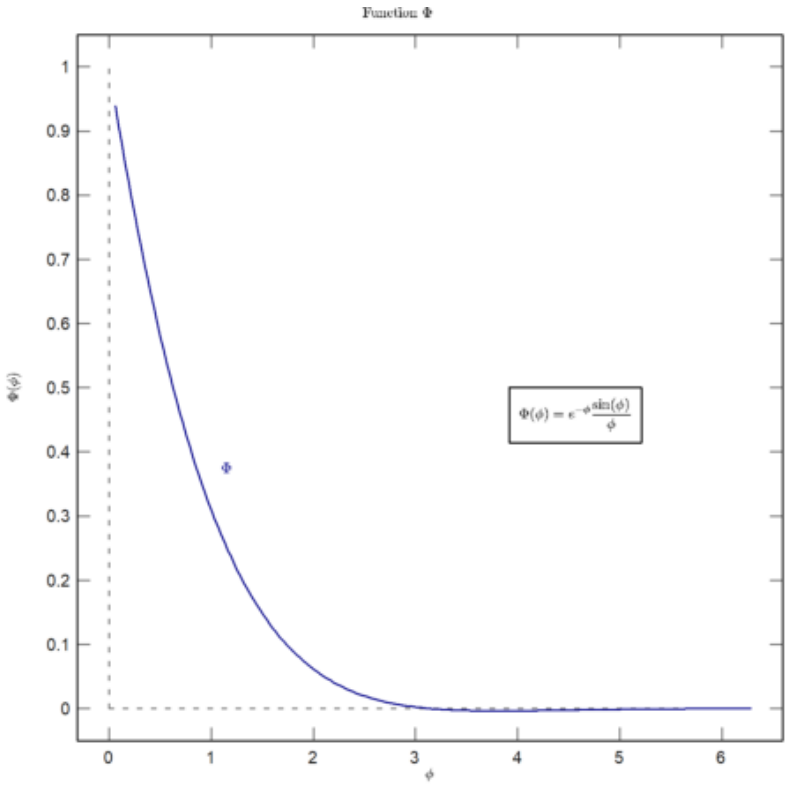

1
0.5
0
x → sinc(x)
0
1
2
3
4
5
6
x

Anda juga dapat memplot formula LaTeX jika Anda telah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Jalur ke binari "lateks" dan "dvipng" harus berada di jalur sistem, atau Anda harus mengatur LaTeX di menu opsi.

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

Pada plot berikut ini, kita menggunakan LaTeX untuk label x dan y, sebuah label, kotak label dan judul plot.

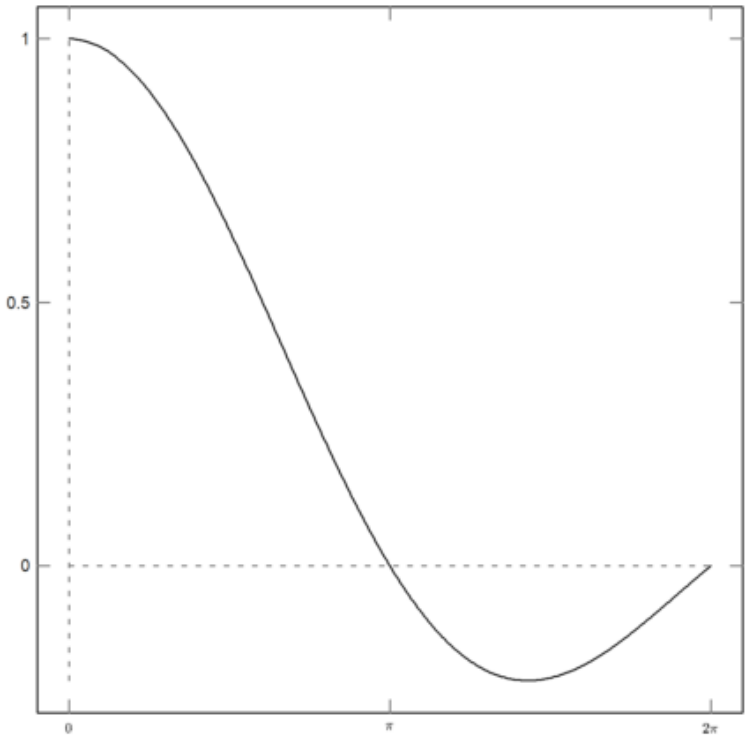
```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
>  title=latex("\text{Function $\Phi$}"), ...
>  xl=latex("\phi"),yl=latex("\Phi(\phi)")); ...
>textbox( ...
>  latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}"),x=0.8,y=0.5); ...
>label(latex("\Phi",color=blue),1,0.4):
```



Seringkali, kita menginginkan spasi dan label teks yang tidak sesuai pada sumbu x. Kita dapat menggunakan xaxis() dan yaxis() seperti yang akan kita tunjukkan nanti.

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

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



Tentu saja, fungsi juga dapat digunakan.

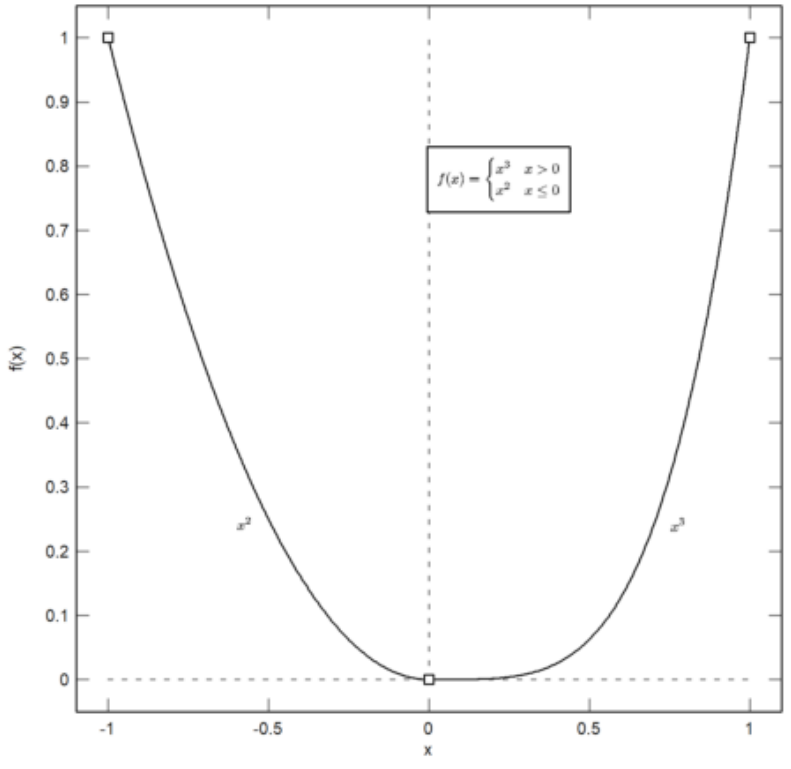
```
>function map f(x) ...
```

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

Parameter "map" membantu menggunakan fungsi untuk vektor. Untuk plot, hal ini tidak diperlukan. Tetapi untuk mendemonstrasikan bahwa vektorisasi berguna, kami menambahkan beberapa titik kunci pada plot pada x=-1, x=0 dan x=1.

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

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


$f(x) = \begin{cases} x^3 & x > 0 \\ x^2 & x \leq 0 \end{cases}$
f(x)
x
x^2
x^3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.5
0
0.5
1

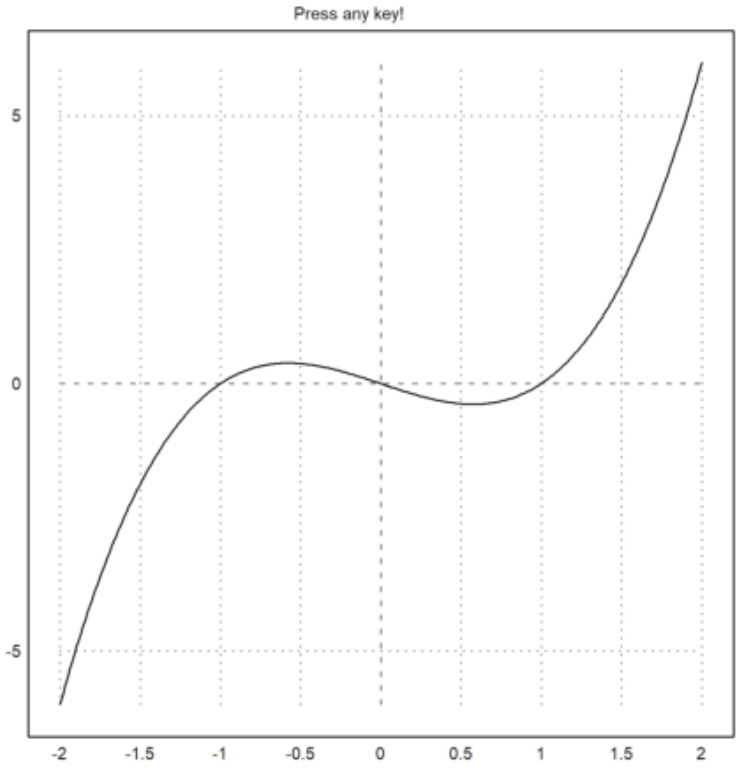
Ketika memplot fungsi atau ekspresi, parameter >user memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kursor atau mouse. Pengguna dapat

- memperbesar dengan + atau -
- memindahkan plot dengan tombol kursor
- memilih jendela plot dengan mouse
- mengatur ulang tampilan dengan spasi
- keluar dengan return

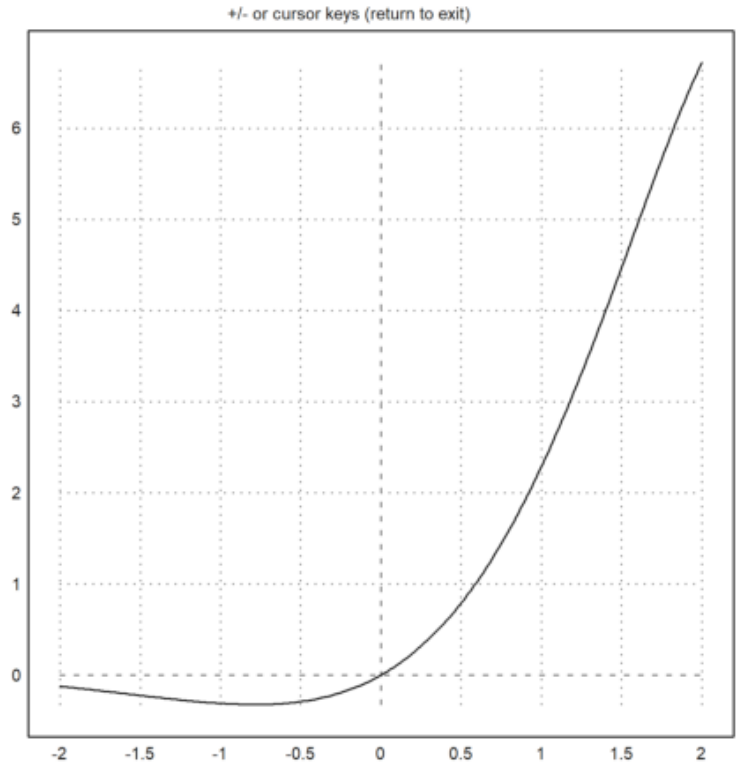
Tombol spasi akan mengatur ulang plot ke jendela plot awal.

Ketika memplot data, bendera >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 peristiwa mouse atau keyboard. Fungsi ini melaporkan mouse ke bawah, mouse bergerak atau mouse ke atas, dan penekanan tombol. Fungsi dragpoints() memanfaatkan hal ini, dan mengizinkan pengguna untuk menyeret titik manapun di dalam plot.

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

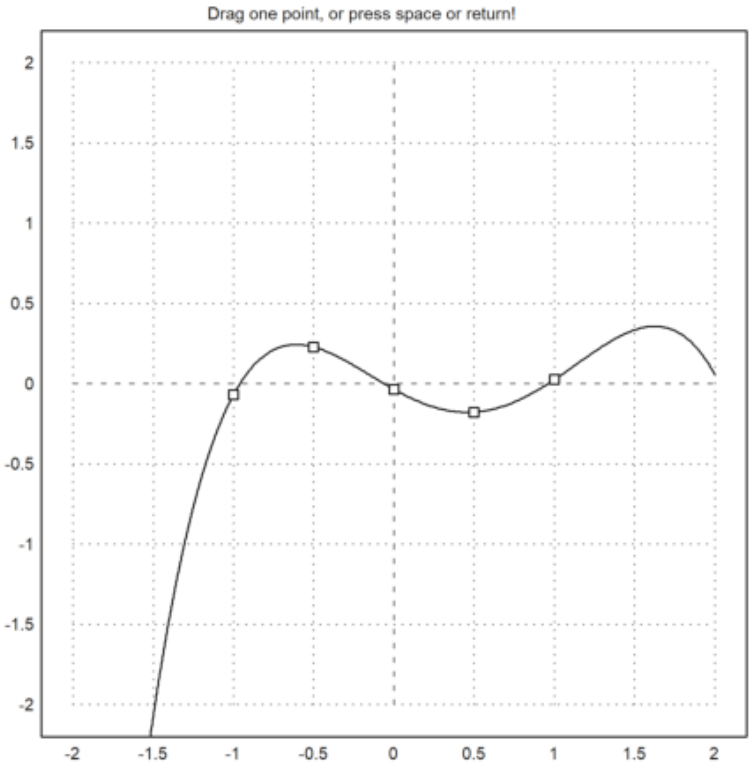
```
>function plotf(xp,yp,select) ...
```

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

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

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

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



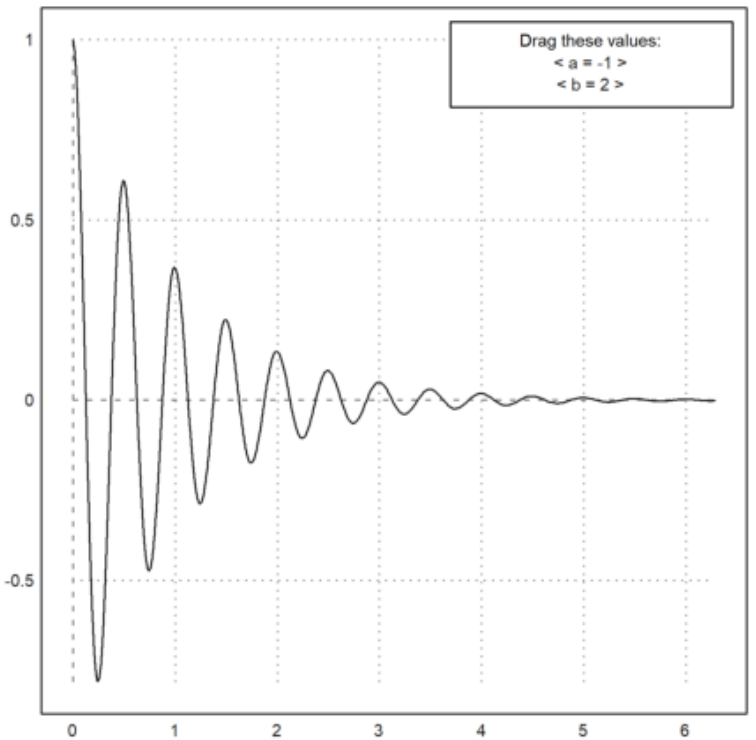
Ada juga fungsi yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.

Pertama, kita memerlukan 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, dan secara opsional, sebuah garis judul.
Terdapat 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):
```



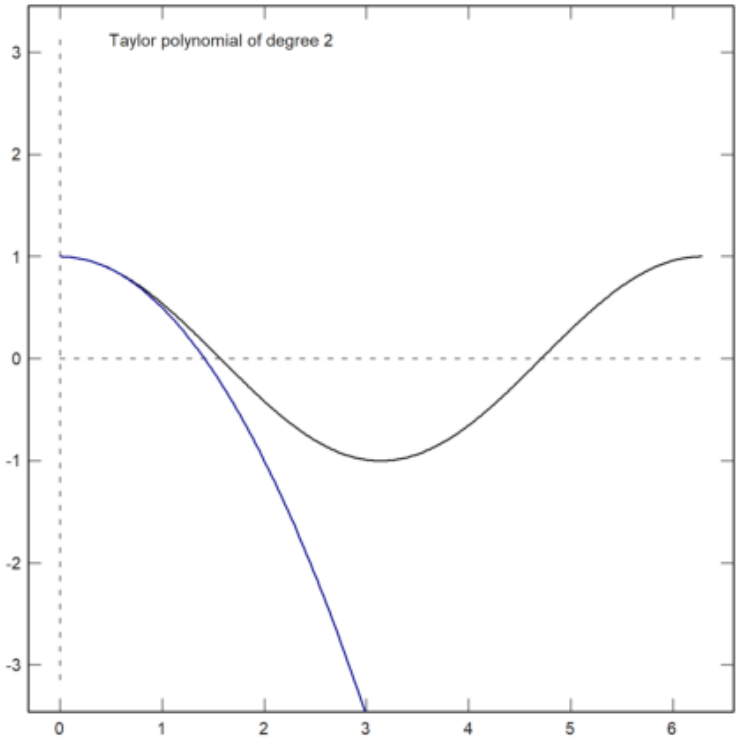
Anda dapat membatasi nilai yang diseret menjadi bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor dengan derajat n ke fungsi kosinus.

```
>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 kita membiarkan derajat n bervariasi dari 0 sampai 20 dalam 20 stop. Hasil dari dragvalues() digunakan untuk memplot sketsa dengan n ini, dan untuk menyisipkan 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 peragaan sederhana dari fungsi ini. Pengguna dapat menggambar di atas jendela plot, meninggalkan jejak titik.

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

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

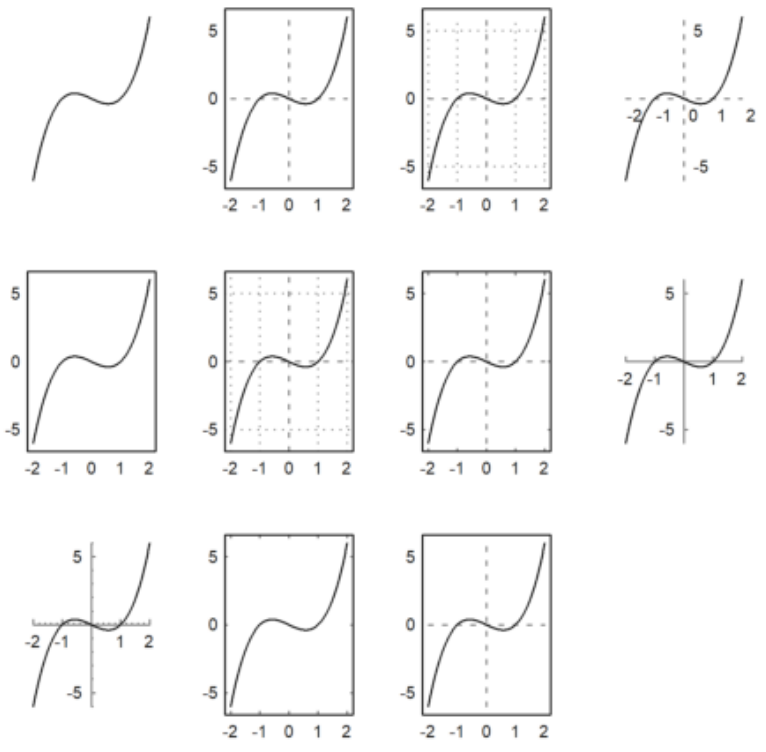
```
>dragtest // lihat hasilnya dan cobalah lakukan!
```

Gaya Plot 2D

Secara default, EMT menghitung tanda sumbu otomatis dan menambahkan label pada setiap tanda. Hal 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
```

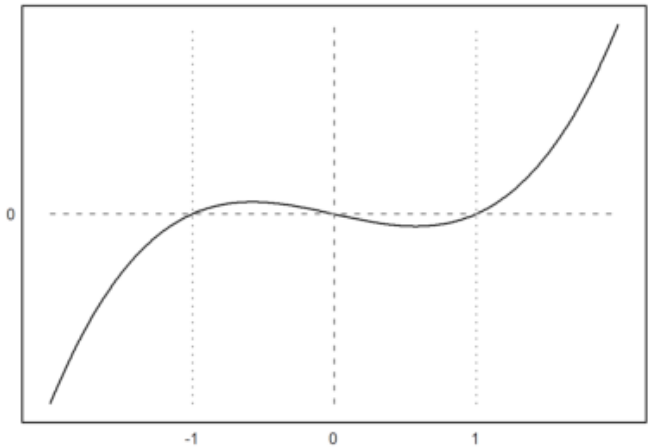
```
> figure(5); plot2d("x^3-x",grid=4); ... // no ticks, only labels
> figure(6); plot2d("x^3-x",grid=5); ... // default, but no margin
> figure(7); plot2d("x^3-x",grid=6); ... // axes only
> figure(8); plot2d("x^3-x",grid=7); ... // axes only, ticks at axis
> figure(9); plot2d("x^3-x",grid=8); ... // 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 bingkai, dan framecolor=blue menetapkan bingkai ke warna biru.

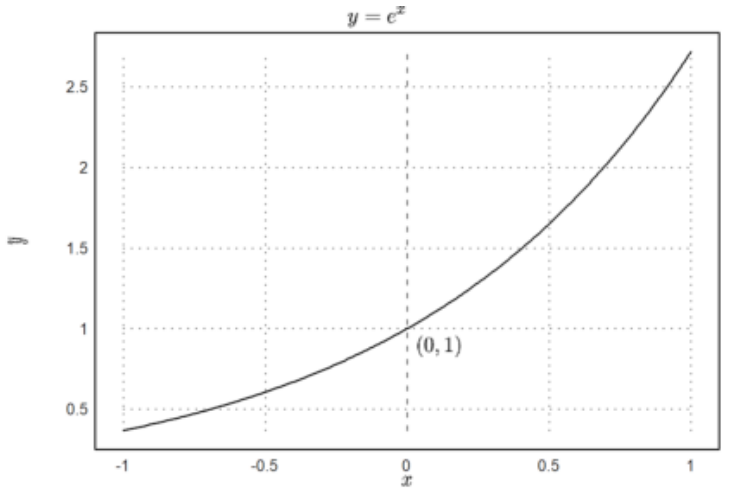
Jika Anda menginginkan tanda centang Anda sendiri, Anda dapat menggunakan style=0, dan menambahkan semuanya nanti.

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



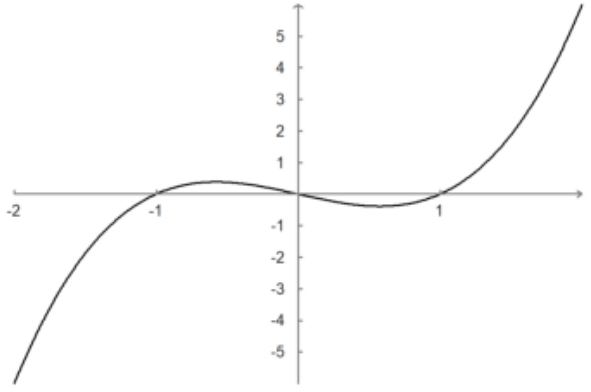
Untuk judul plot dan label sumbu, lihat contoh berikut.

```
>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 sumbu x() dan sumbu y().

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

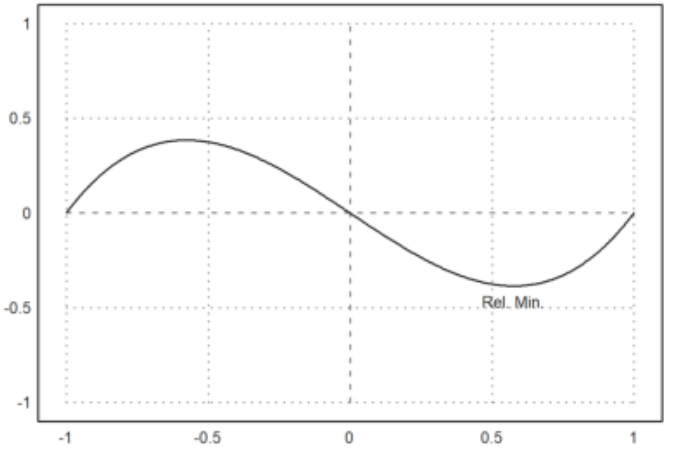


Teks pada plot dapat diatur dengan label(). Pada contoh berikut ini, ”lc” berarti lower center. Ini mengatur posisi label relatif terhadap koordinat plot.

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

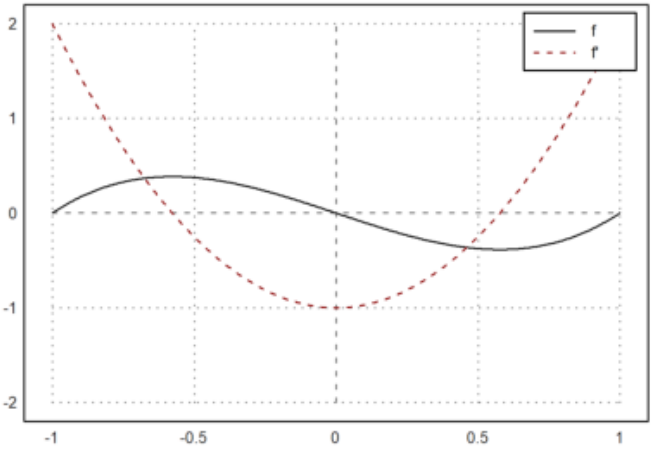
$$x^3 - x$$

```
>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
```

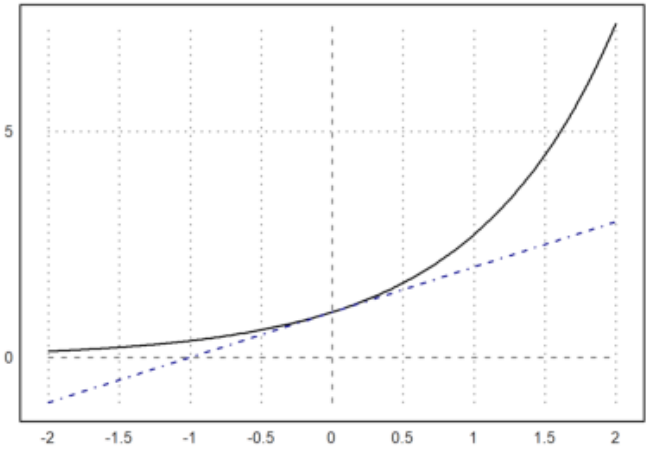


Terdapat juga kotak teks.

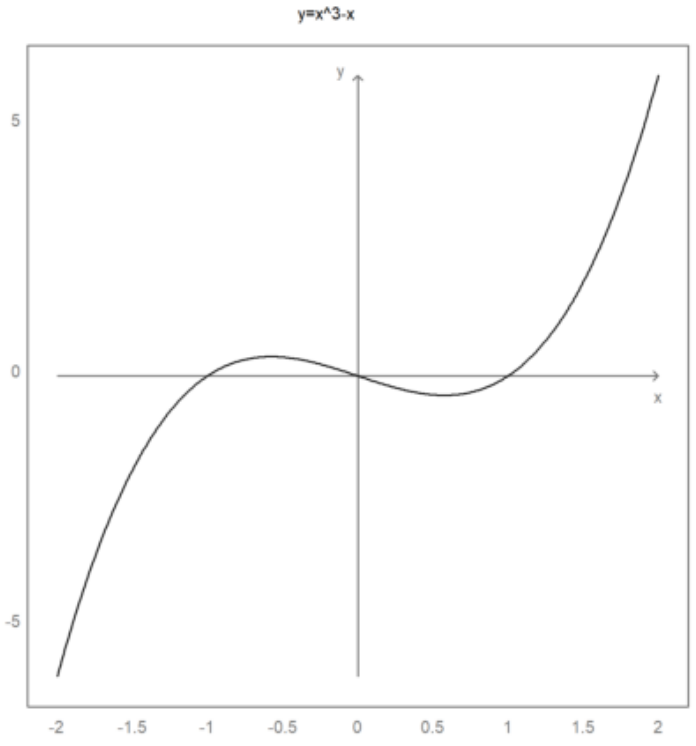
```
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // derivative
>labelbox(["f","f'"],["-","--"],[black,red]): // label box
```




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



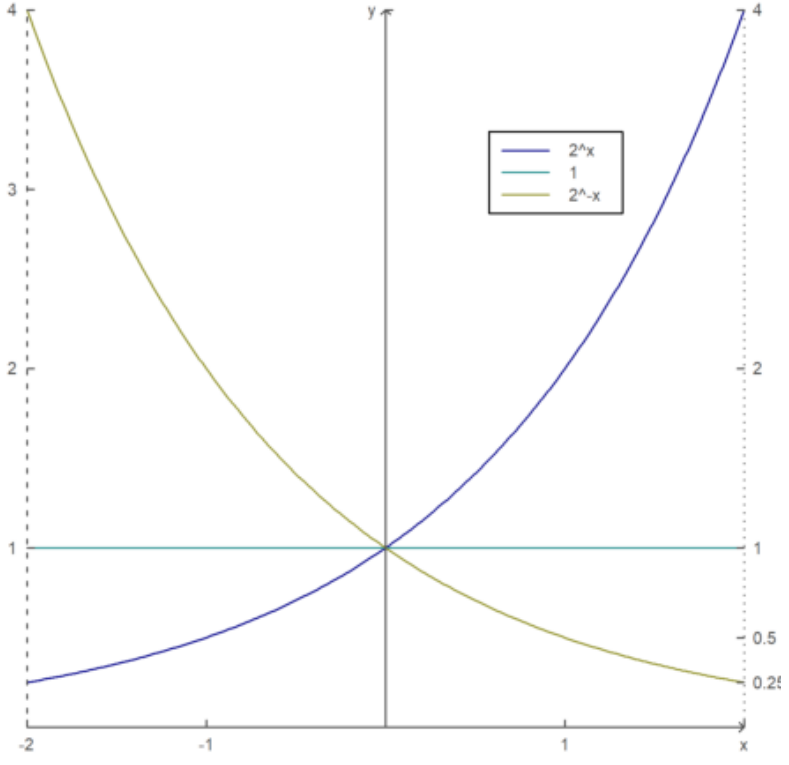

```
>gridstyle("->",color=gray,textcolor=gray,framecolor=gray);  ...
> plot2d("x^3-x",grid=1);   ...
> settitle("y=x^3-x",color=black); ...
> label("x",2,0,pos="bc",color=gray);  ...
> label("y",0,6,pos="cl",color=gray); ...
> reset():
```



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

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

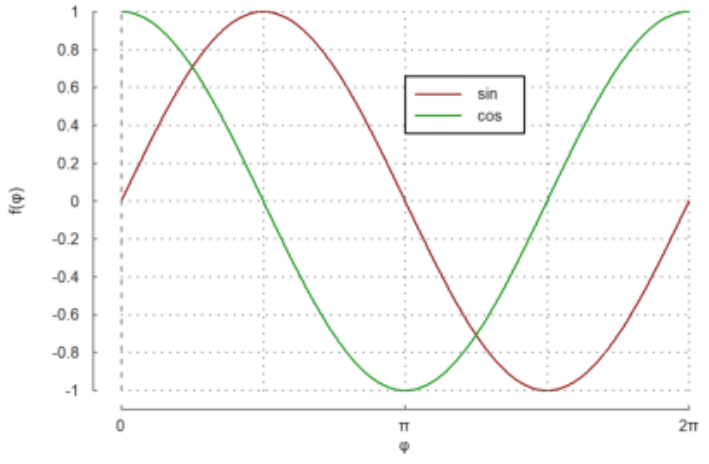
```
>fullwindow; ...
> gridstyle(color=darkgray,textcolor=darkgray); ...
> plot2d(["2^x","1","2^(-x)"],a=-2,b=2,c=0,d=4,<grid,color=4:6,<frame); ...
> xaxis(0,-2:1,style="->"); xaxis(0,2,"x",<axis); ...
> yaxis(0,4,"y",style="->"); ...
> yaxis(-2,1:4,>left); ...
> yaxis(2,2^(-2:2),style=".",<left); ...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



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

```
>aspect(1.5);
>plot2d(["sin(x)","cos(x)"],0,2pi,color=[red,green],<grid,<frame); ...
> xaxis(-1.1,(0:2)*pi,xt=["0",u"&pi;",u"2&pi;"],style="-",>ticks,>zero);  ...
> xgrid((0:0.5:2)*pi,<ticks); ...
```

```
> yaxis(-0.1*pi,-1:0.2:1,style="-",>zero,>grid); ...
> labelbox(["sin","cos"],colors=[red,green],x=0.5,y=0.2,>left); ...
> xlabel(u"&phi;"); ylabel(u"f(&phi;)"):
```



Memplot Data 2D

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

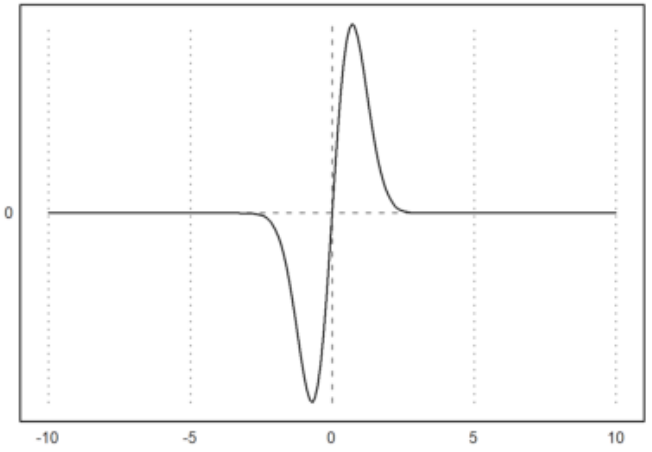
Memplot ekspresi hanyalah singkatan untuk plot data. Untuk plot data, Anda memerlukan satu atau beberapa baris nilai x, dan satu atau beberapa baris nilai y. Dari rentang dan nilai x, fungsi plot2d akan menghitung data untuk diplot, secara default dengan evaluasi adaptif dari fungsi tersebut. Untuk plot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

Namun Anda dapat memasukkan data secara langsung.

- Gunakan vektor baris untuk x dan y untuk satu fungsi.
- Matriks untuk x dan y diplot baris demi baris.

Berikut adalah contoh dengan satu baris untuk x dan y.

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

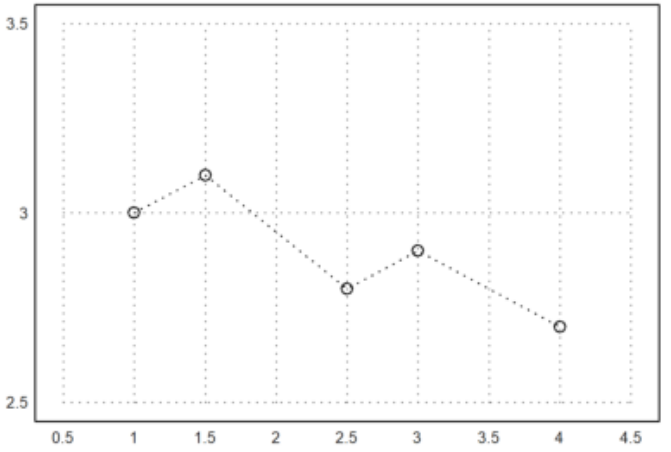


Data juga dapat diplot sebagai titik. Gunakan poin=true untuk ini. Plot ini bekerja seperti poligon, namun hanya menggambar sudut-sudutnya saja.

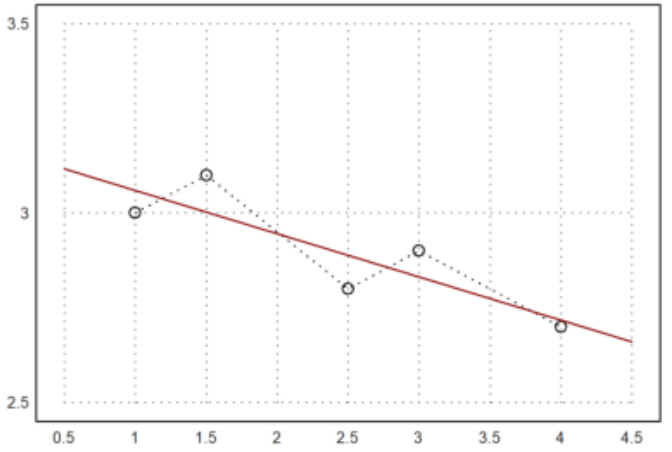
- style = "...": Pilih dari "[]", "<>", "o", ".", "..", "+", "*", "[] ", "<>", "o", "..", "", "|".

Untuk memplot kumpulan titik, gunakan >titik. Jika warna adalah sebuah vektor warna, setiap titik mendapatkan warna yang berbeda. Untuk sebuah matriks koordinat dan vektor kolom, warna berlaku pada baris-baris matriks.
Parameter >addpoints menambahkan titik-titik pada segmen garis untuk plot data.

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




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


3.5
3
2.5
0.5
1
1.5
2
2.5
3
3.5
4
4.5

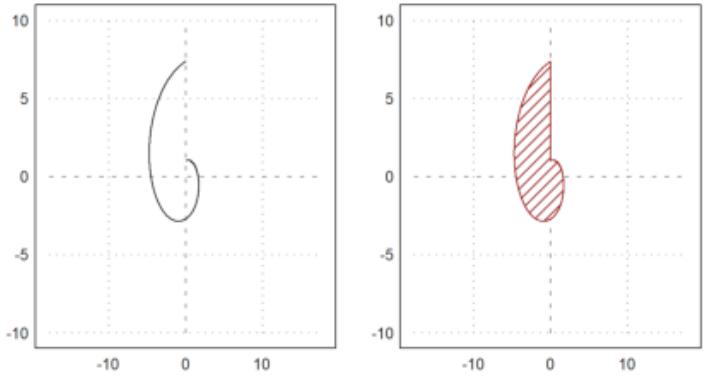
Menggambar Daerah Yang Dibatasi Kurva

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

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

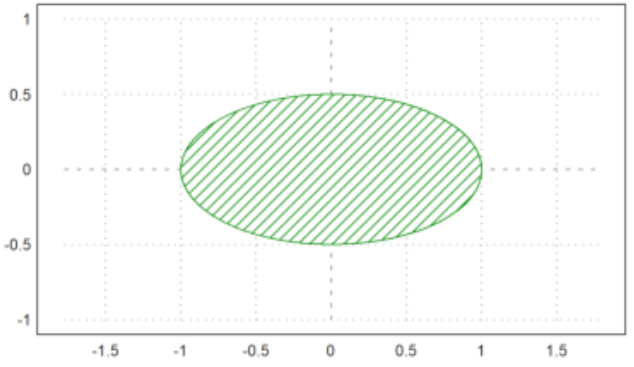
Warna isian ditentukan oleh argumen "fillcolor", dan pada pilihan <outline mencegah menggambar batas untuk semua gaya kecuali gaya default.

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

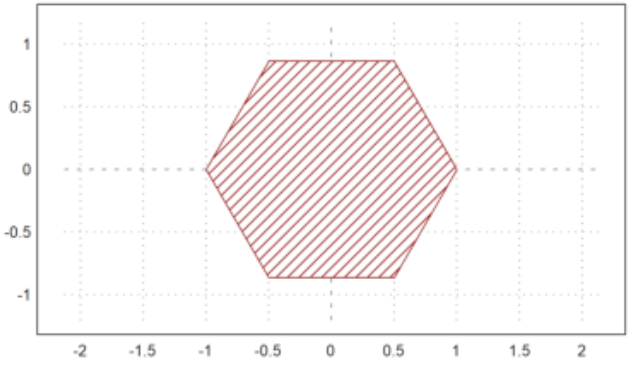


Pada contoh berikut ini, kami memplot elips terisi dan dua segi enam terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian yang berbeda.

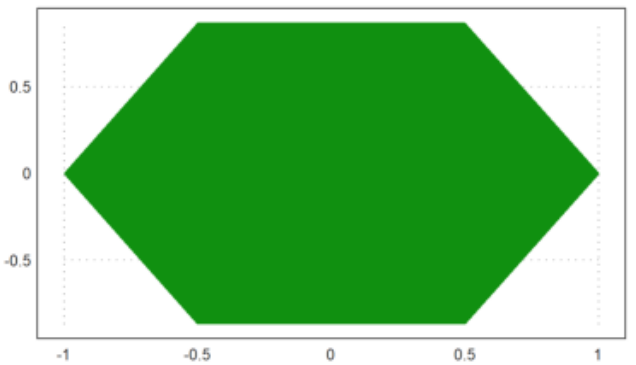
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```




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

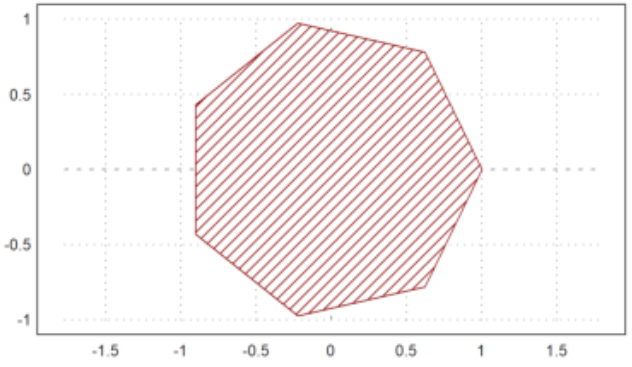



```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



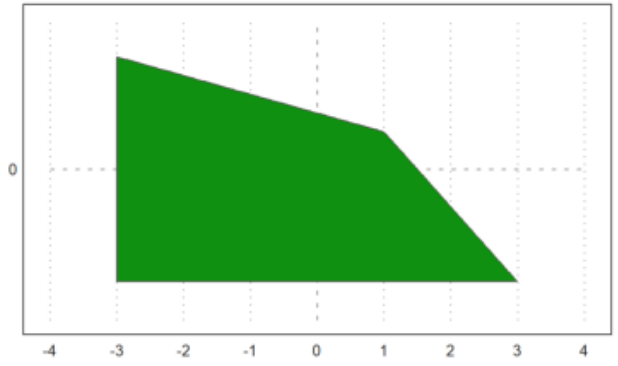
Contoh lainnya adalah septagon, yang kita buat dengan 7 titik pada lingkaran satuan.

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



Berikut ini adalah himpunan nilai maksimal dari empat kondisi linier yang kurang dari atau sama dengan 3. Ini adalah A[k].v<=3 untuk semua barisan A. Untuk mendapatkan sudut-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):
```

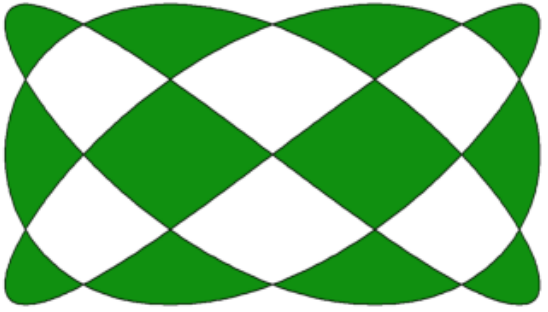


Poin utama dari bahasa matriks adalah bahwa bahasa ini memungkinkan untuk menghasilkan tabel fungsi dengan mudah.

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

Kita sekarang memiliki vektor nilai x dan y. plot2d() dapat memplot nilai-nilai ini sebagai sebuah kurva yang menghubungkan titik-titik. Plot dapat diisi. Dalam kasus ini ini memberikan hasil yang bagus karena aturan penggulungan, yang digunakan untuk pengisian.

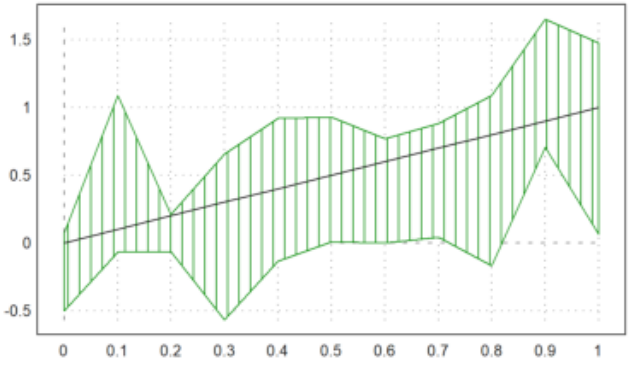
```
>plot2d(x,y,<grid,<frame,>filled):
```



Vektor interval diplot terhadap nilai x sebagai wilayah yang terisi antara nilai bawah dan atas interval.

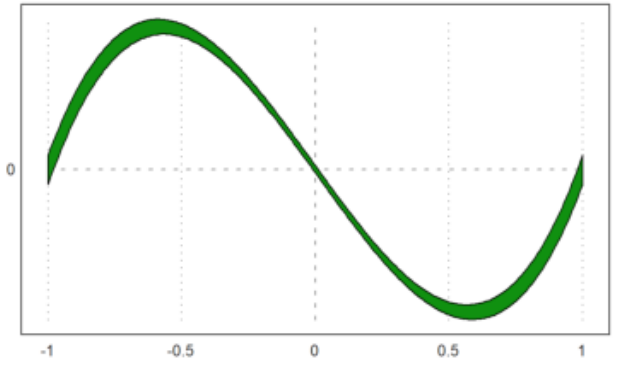
Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga dapat digunakan untuk memplot kesalahan statistik.

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



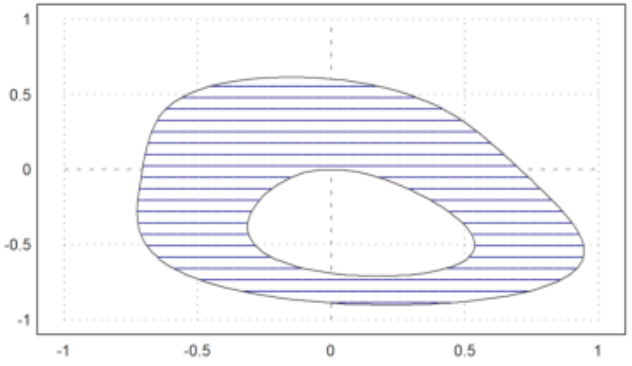
Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka plot2d akan memplot rentang interval yang terisi pada bidang, gaya isian sama dengan gaya poligon.

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



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.

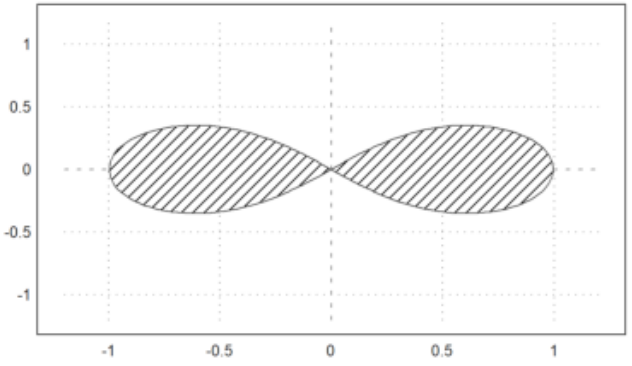
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



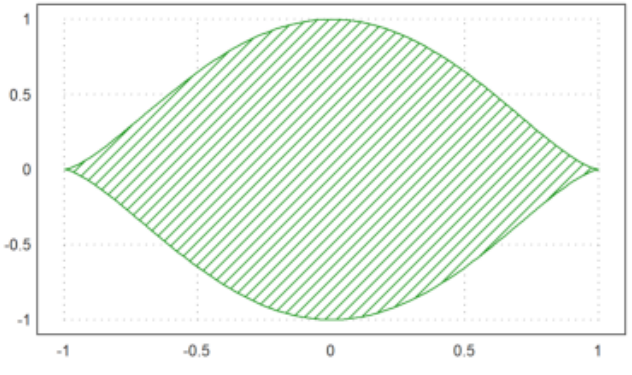
Kita juga dapat mengisi rentang nilai seperti

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

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




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



Grafik Fungsi Parametrik

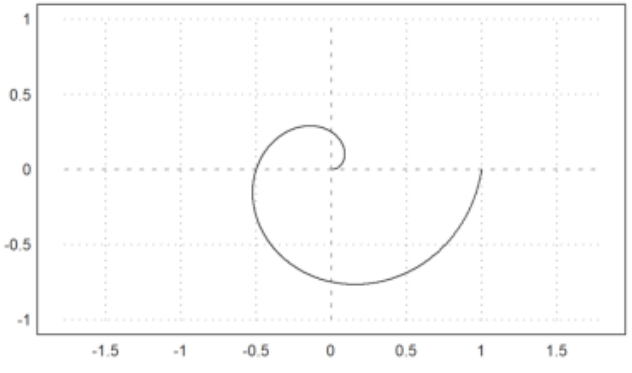
Nilai x tidak perlu diurutkan. (x,y) hanya menggambarkan sebuah kurva. Jika x diurutkan, kurva tersebut adalah grafik fungsi.

Pada contoh berikut, kita memplot spiral

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

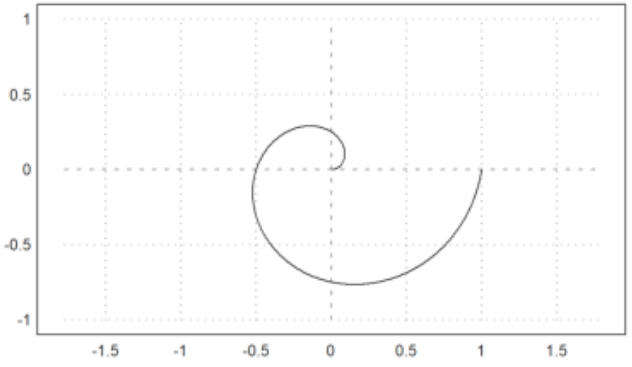
Kita mungkin perlu menggunakan sangat banyak titik untuk tampilan yang halus atau fungsi adaptive() untuk mengevaluasi ekspresi (lihat fungsi adaptive() untuk lebih jelasnya).

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

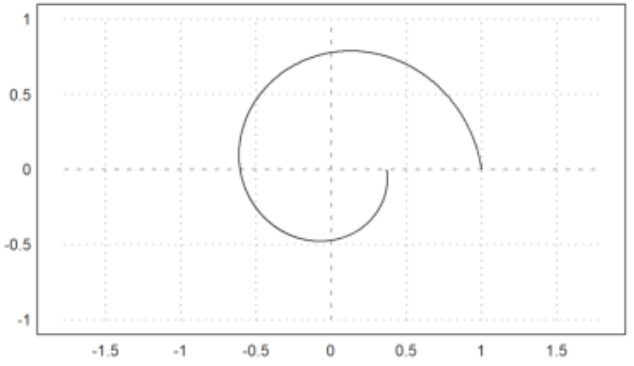


Sebagai alternatif, Anda dapat menggunakan dua ekspresi untuk kurva. Berikut ini adalah plot kurva yang sama seperti di atas.

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




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



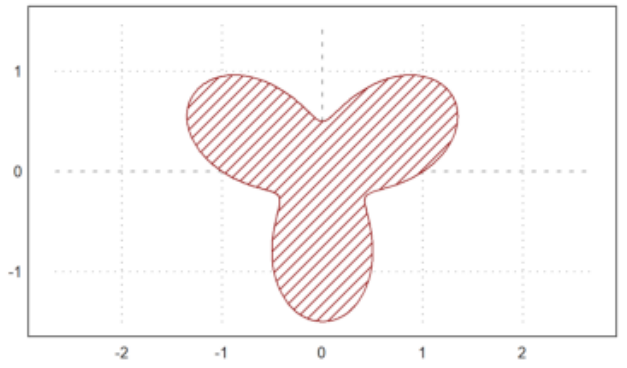
Dalam contoh berikut, kami memplot kurva

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

dengan

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

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


1
0
-1
-2
-1
0
1
2

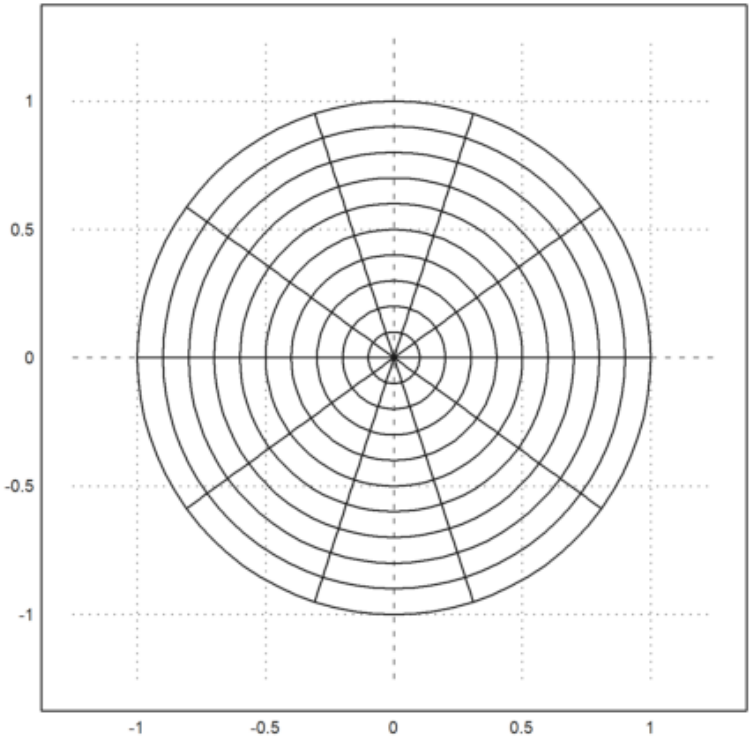
Menggambar Grafik Bilangan Kompleks

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

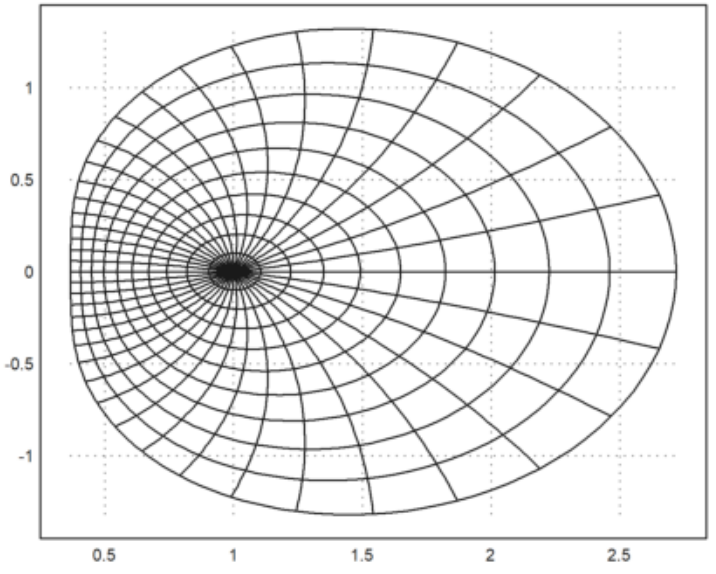
Matriks bilangan kompleks akan secara otomatis diplot sebagai sebuah grid pada bidang kompleks.

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

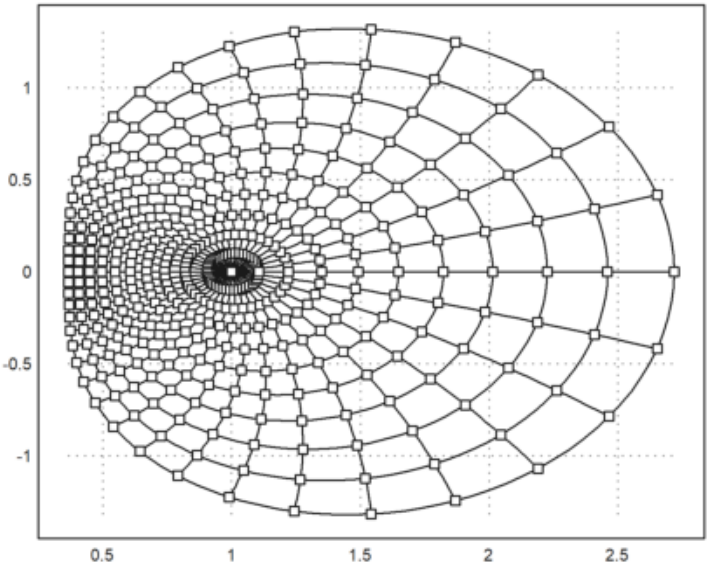
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```




```
>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):
```

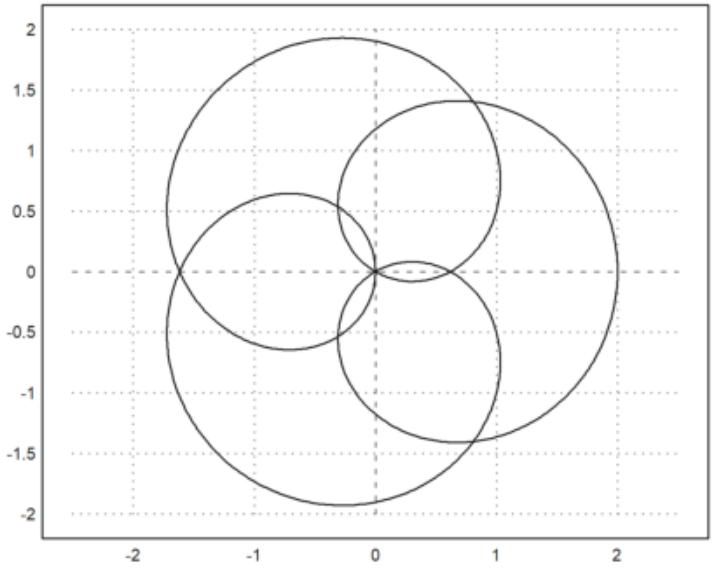


Vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian nyata dan bagian imajiner.

Dalam contoh, kami memplot lingkaran satuan dengan

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

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


2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1
0
1
2

Plot Statistik

Terdapat banyak fungsi yang dikhususkan untuk plot statistik. Salah satu plot yang sering digunakan adalah plot kolom.

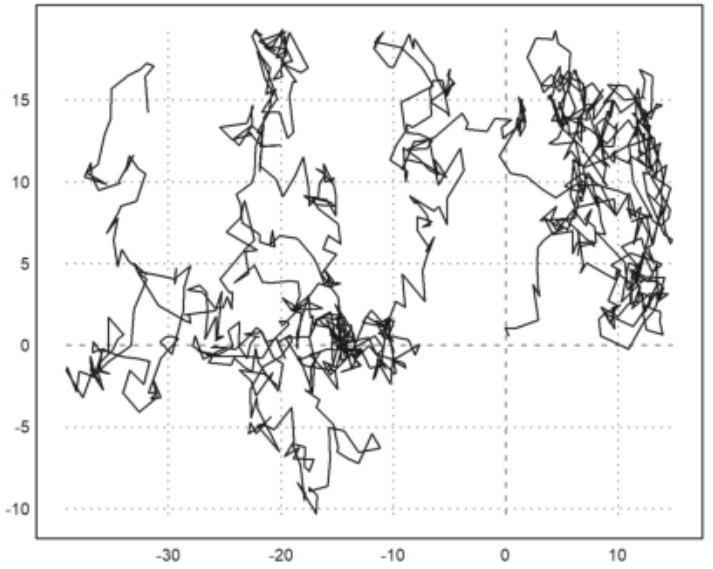
Jumlah kumulatif dari nilai berdistribusi normal 0-1 menghasilkan jalan acak.

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

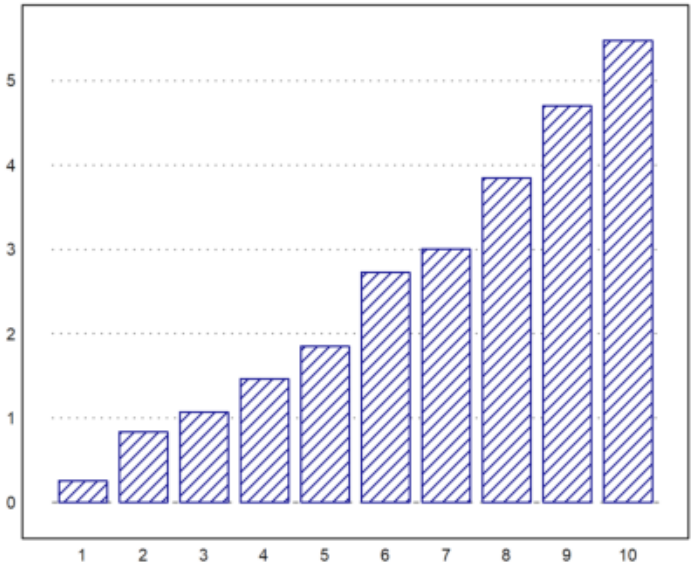


Dengan menggunakan dua baris, ini menunjukkan jalan kaki dalam dua dimensi.

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

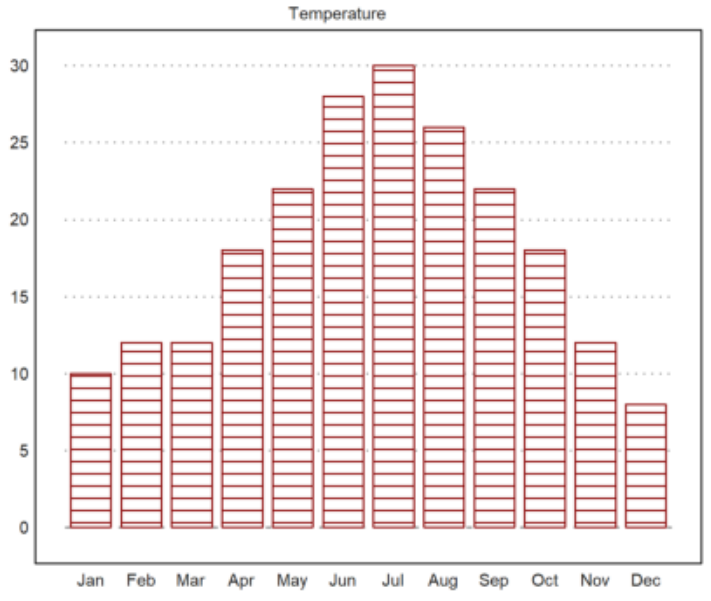



```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

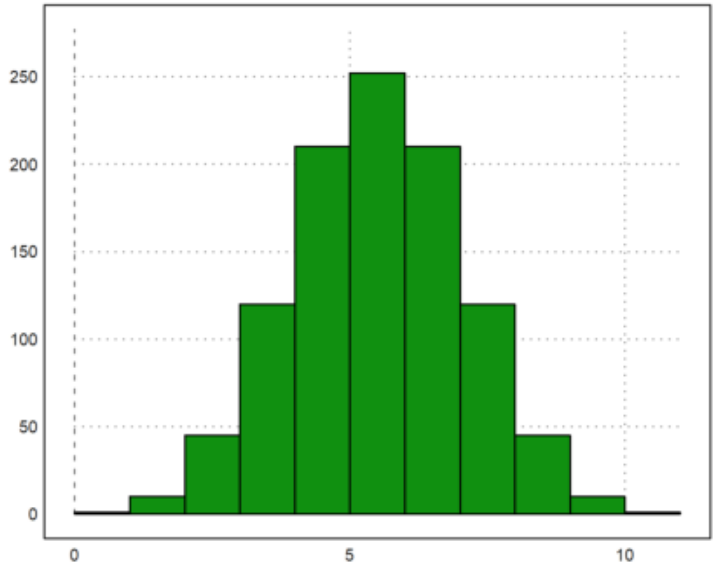


Ini juga dapat menampilkan string sebagai label.

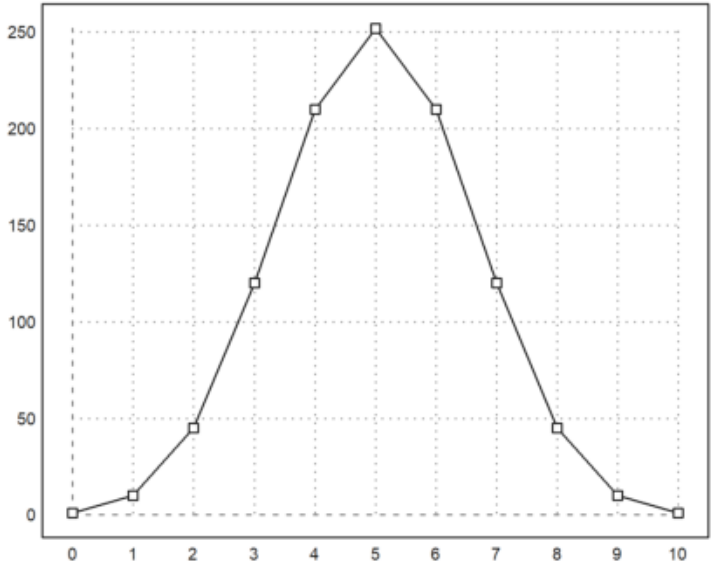
```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
>  "Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnsplot(values,lab=months,color=red,style="-");
>title("Temperature"):
```



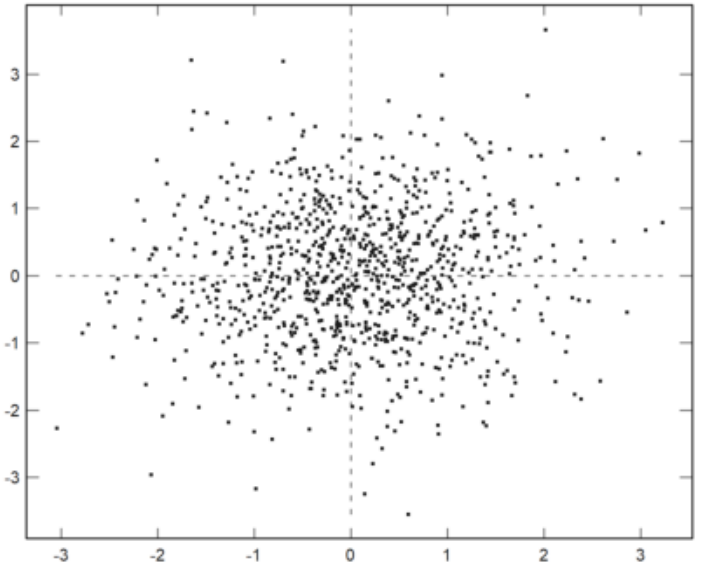

```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```



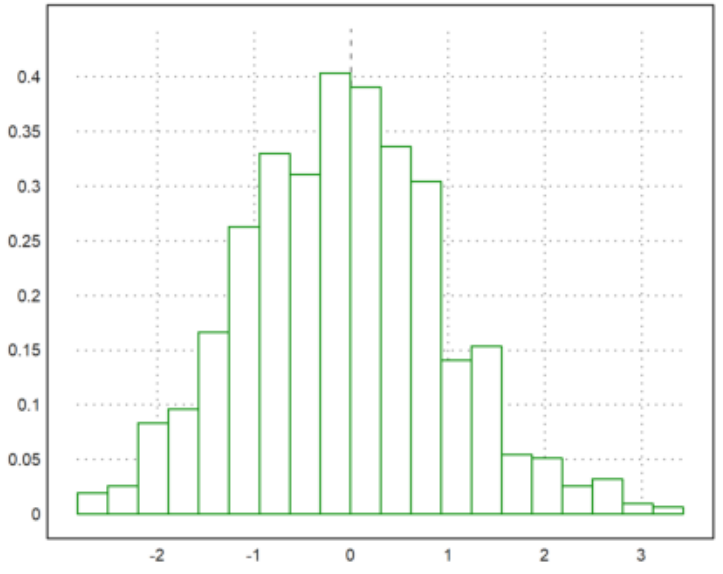

```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



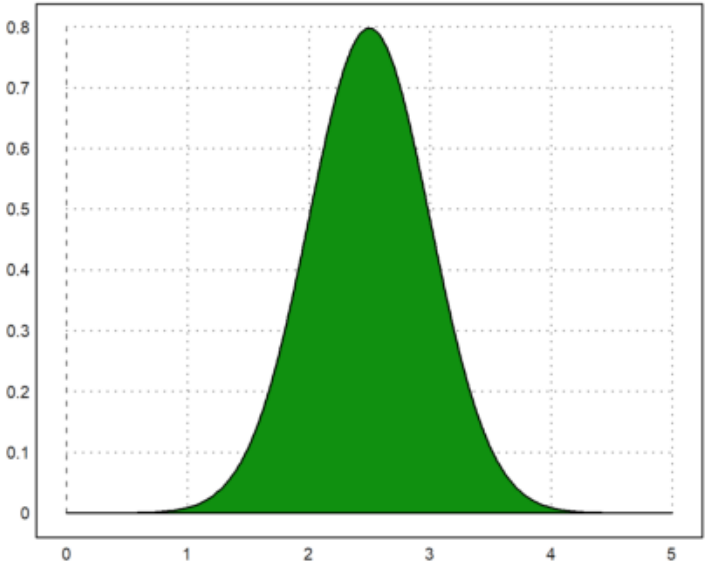

```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```




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

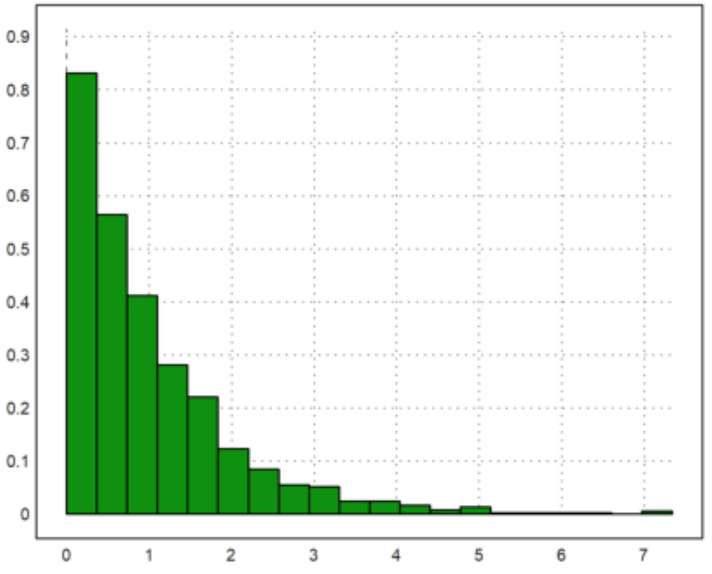



```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



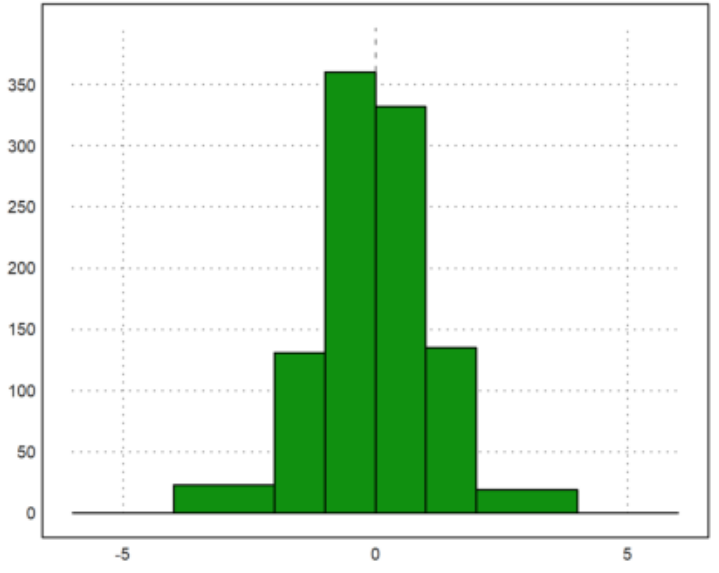
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan distribution=n dengan plot2d.

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



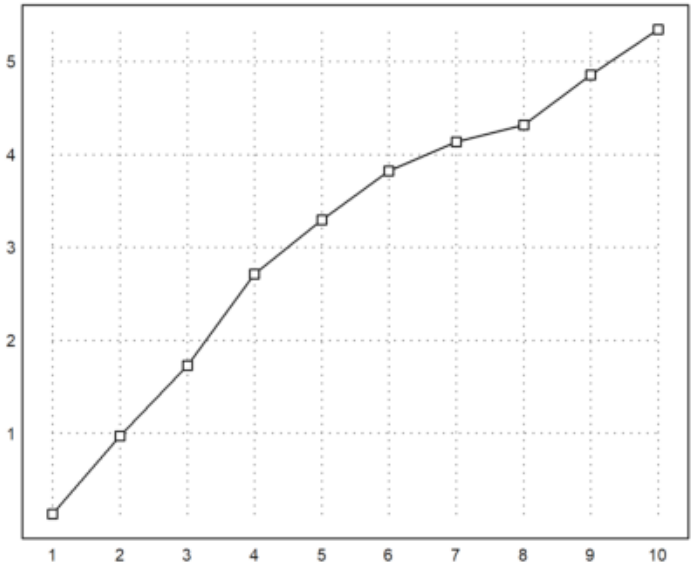
Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan >bar di plot3d, atau dengan plot kolom.

```
>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):
```

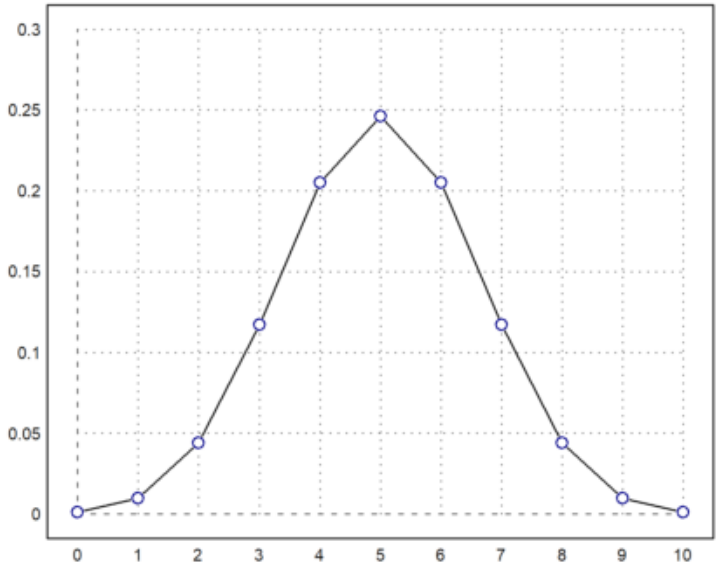


Fungsi statplot() menetapkan gaya dengan string sederhana.

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



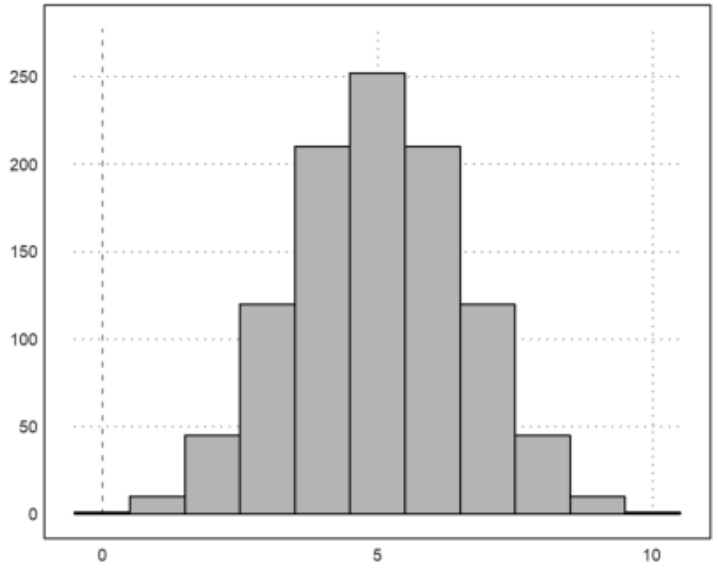

```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



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

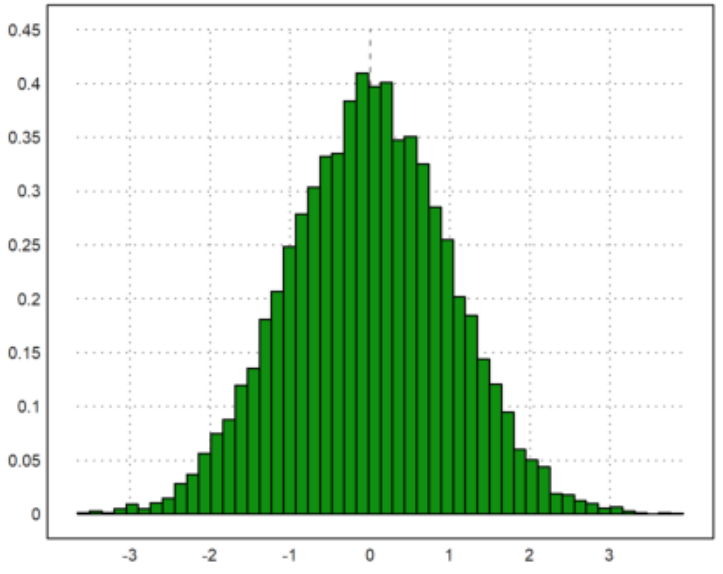
Gaya isian dapat digunakan seperti di atas.

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

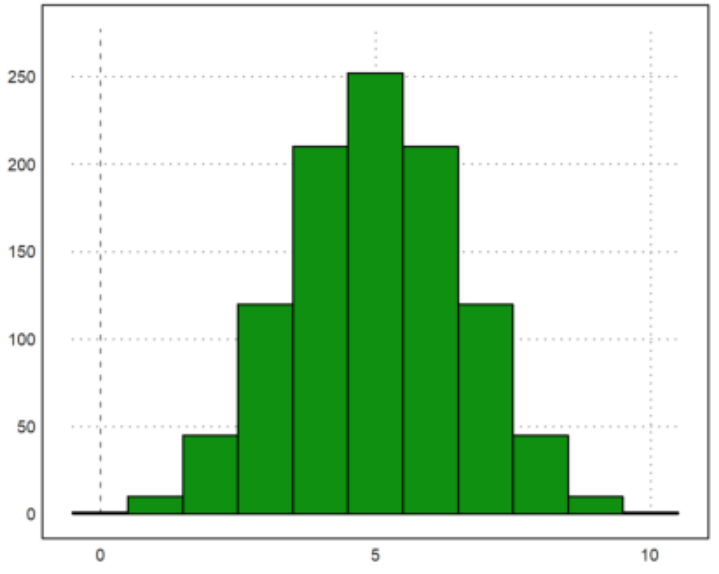


Data untuk plot batang (bar = 1) dan histogram (histogram = 1) dapat diberikan secara eksplisit dalam xv dan yv, atau dapat dihitung dari distribusi empiris dalam xv dengan >distribusi (atau distribusi = n). Histogram dari nilai xv akan dihitung secara otomatis dengan >histogram. Jika >even ditentukan,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

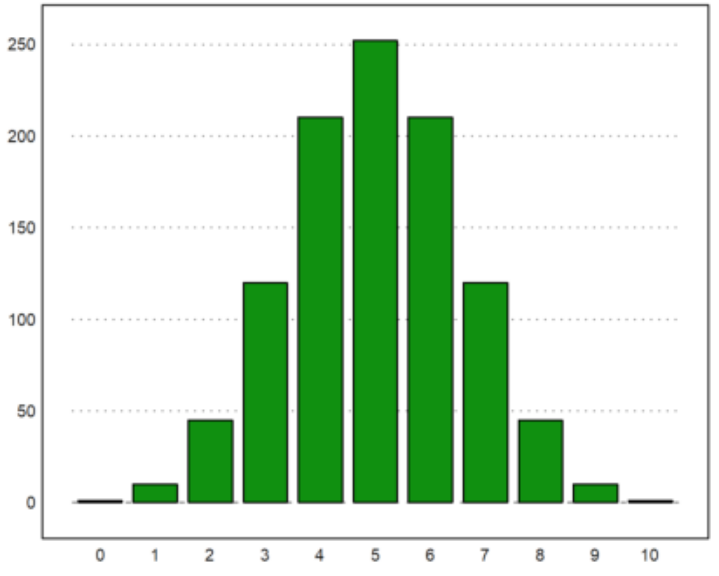
```
>plot2d(normal(10000),distribution=50):
```



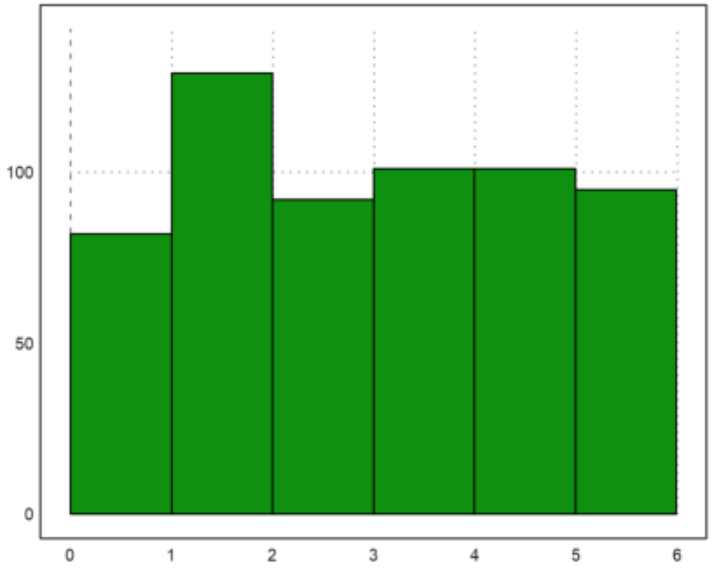

```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```




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

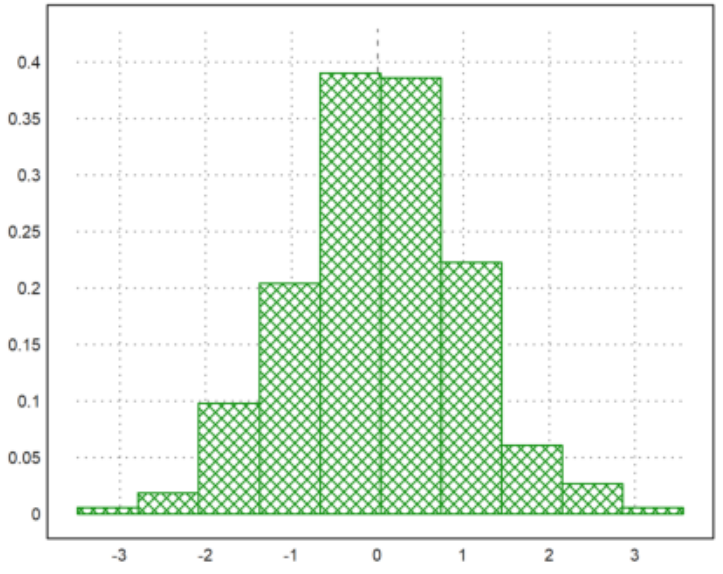



```
>plot2d(random(600)*6,histogram=6):
```



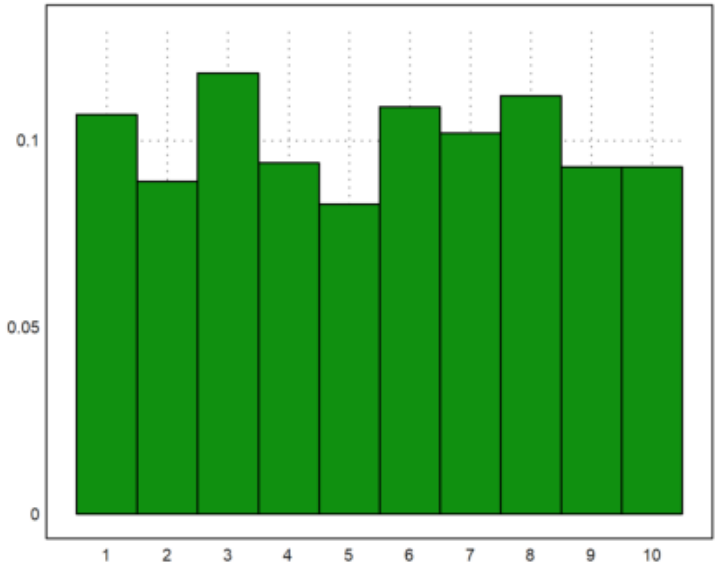
Untuk distribusi, ada parameter distribution=n, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.

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



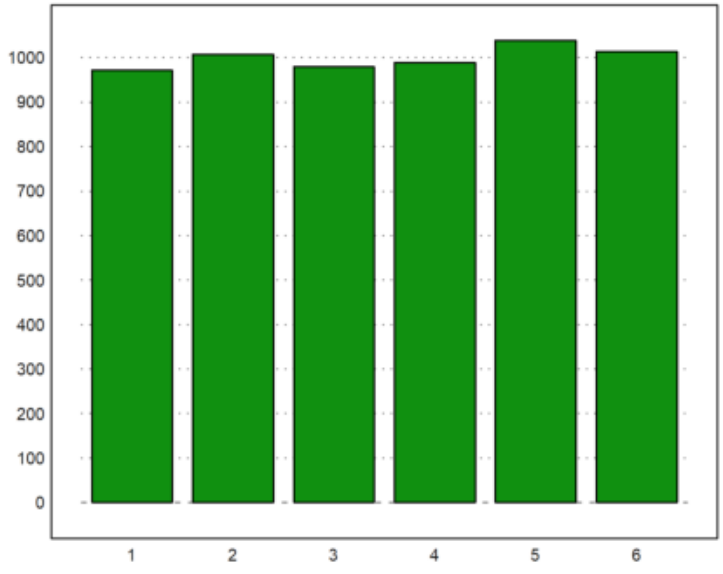
Dengan parameter even=true, ini akan menggunakan interval bilangan bulat.

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

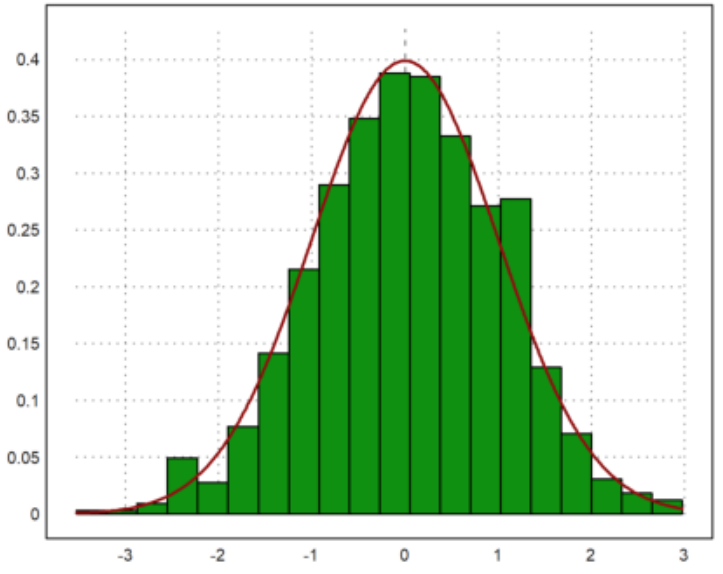


Perhatikan bahwa ada banyak plot statistik yang mungkin berguna. Lihatlah tutorial tentang statistik.

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

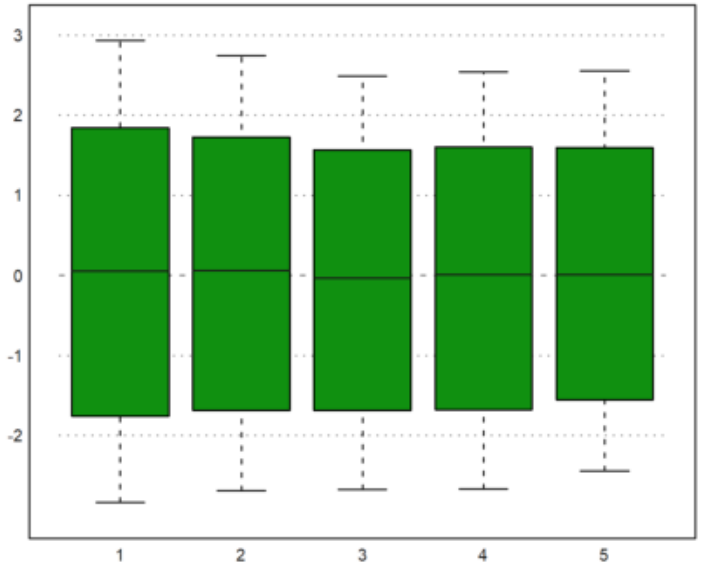



```
>plot2d(normal(1,1000),>distribution); ...
>  plot2d("qnormal(x)",color=red,thickness=2,>add):
```



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

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


3
2
1
0
-1
-2
1
2
3
4
5

Fungsi Implisit

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

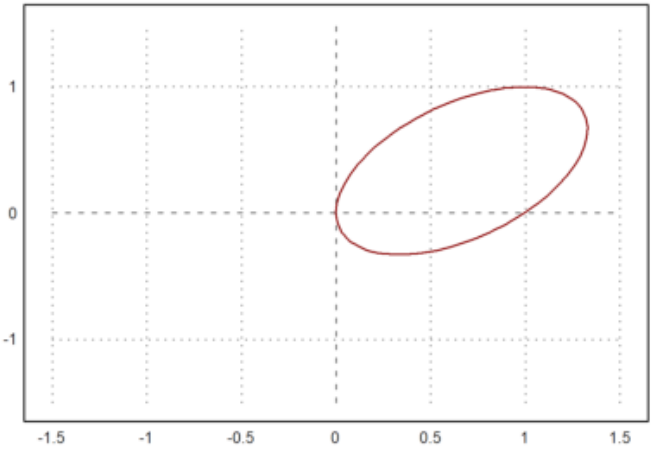
Euler dapat menandai garis level

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

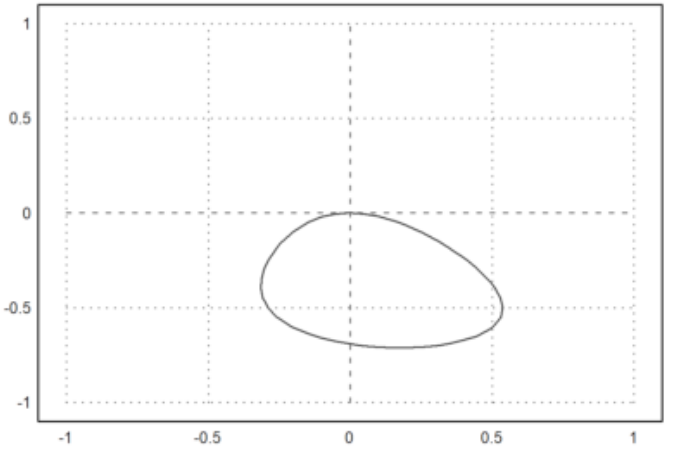
dari fungsi apa pun.

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

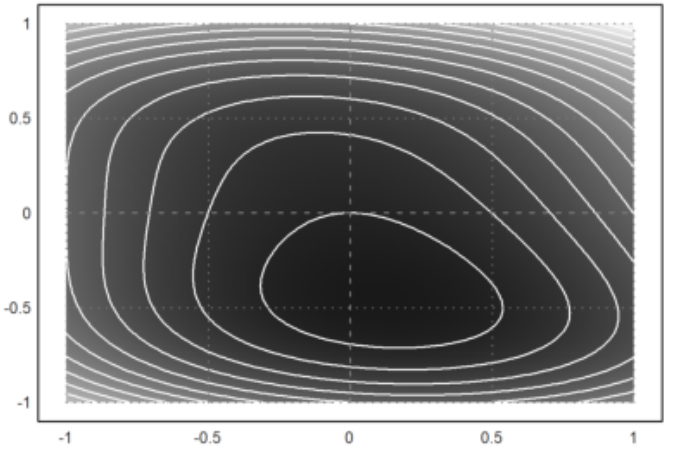
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```



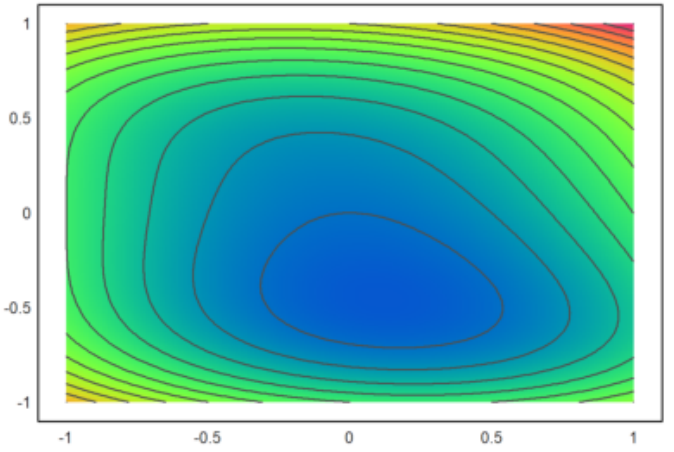

```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0): // Solutions of f(x,y)=0
```




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

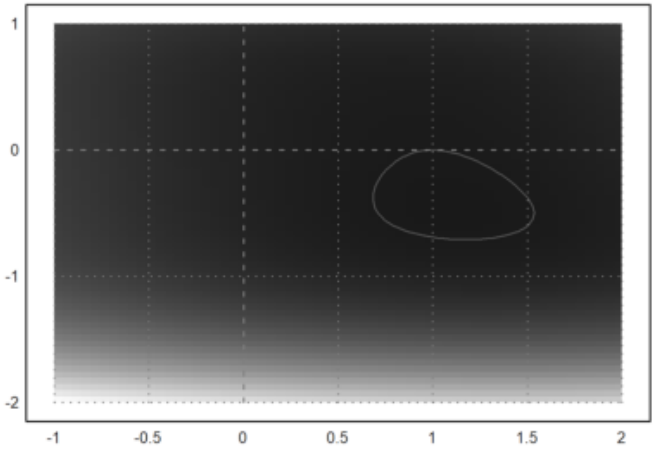



```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
```

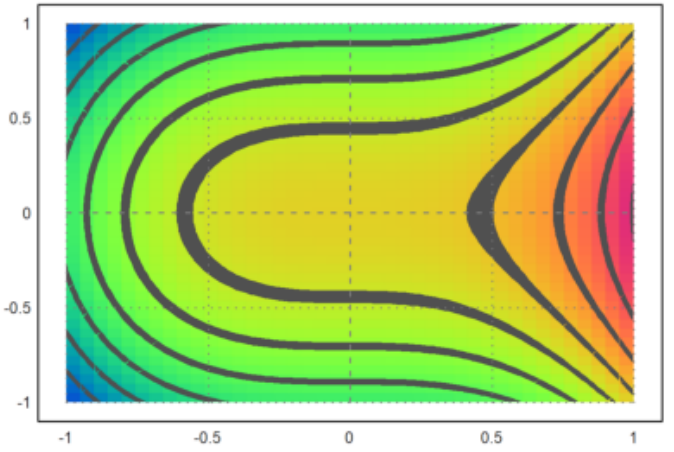


Hal ini juga berlaku untuk plot data. Tetapi Anda harus menentukan rentang untuk label sumbu.

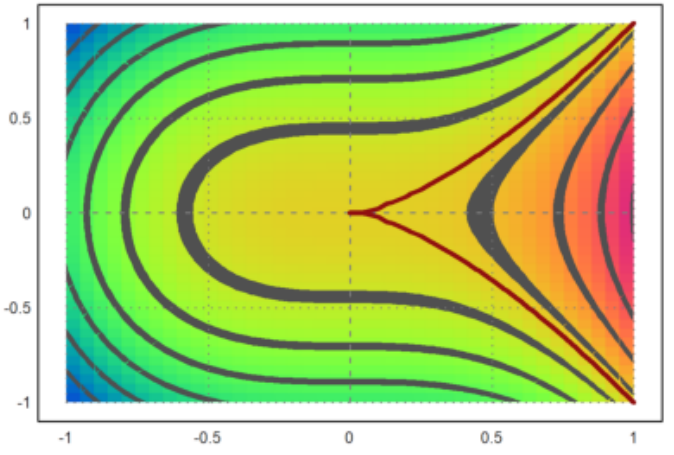
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



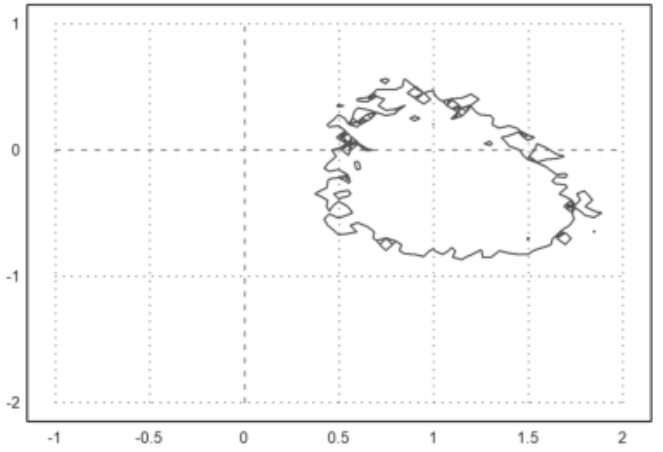

```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



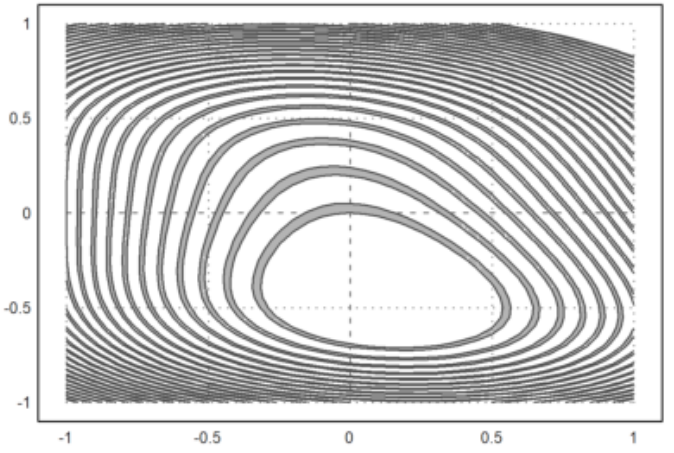

```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



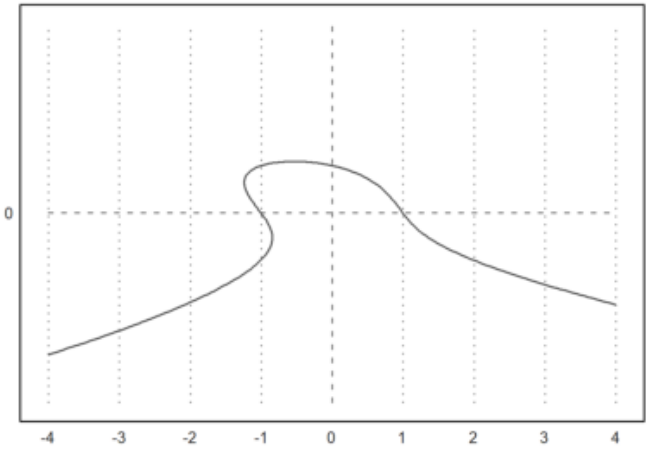

```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



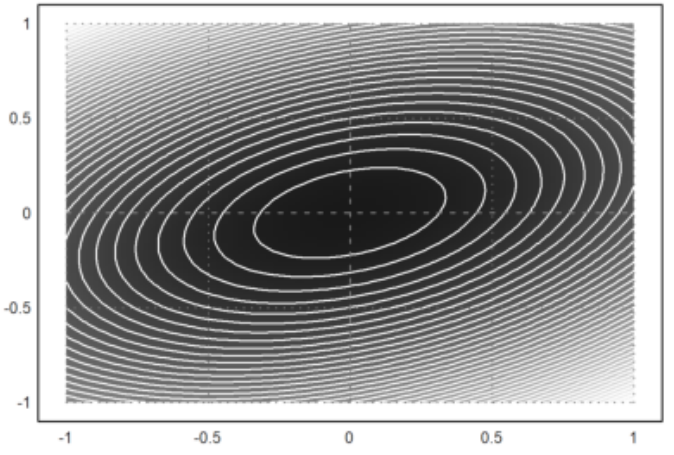

```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```




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




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



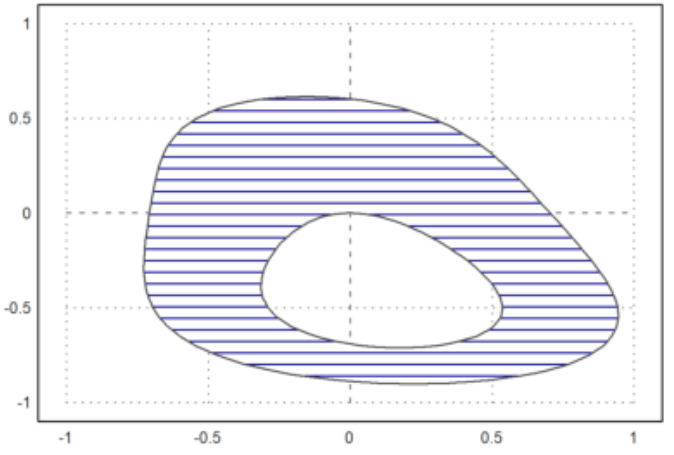
Dimungkinkan juga untuk mengisi set

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

dengan rentang level.

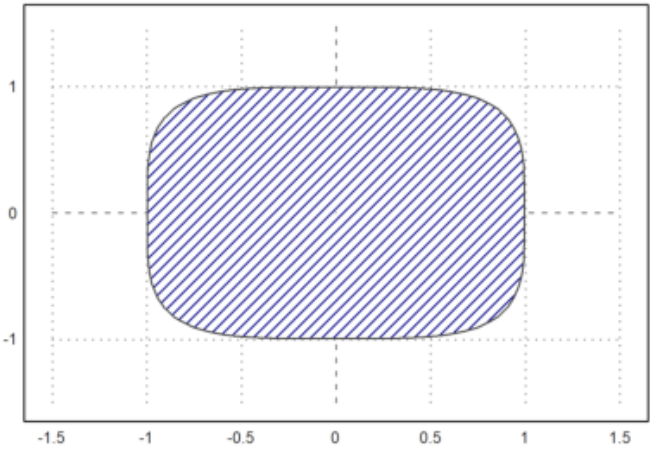
Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

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

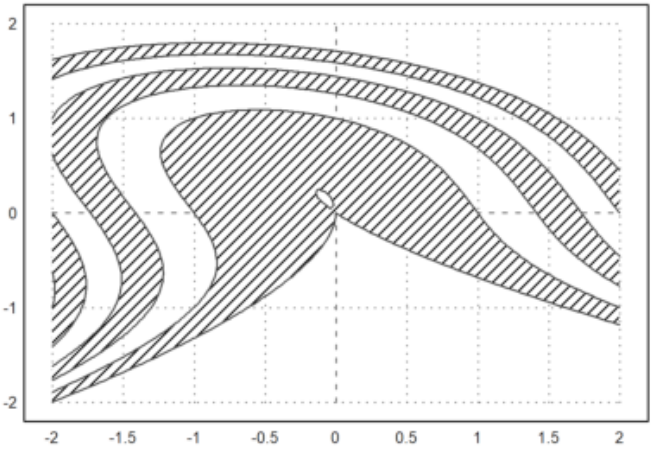


Plot implisit juga dapat menunjukkan rentang level. Maka level harus berupa matriks 2xn interval level, di mana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval. Sebagai alternatif, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level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

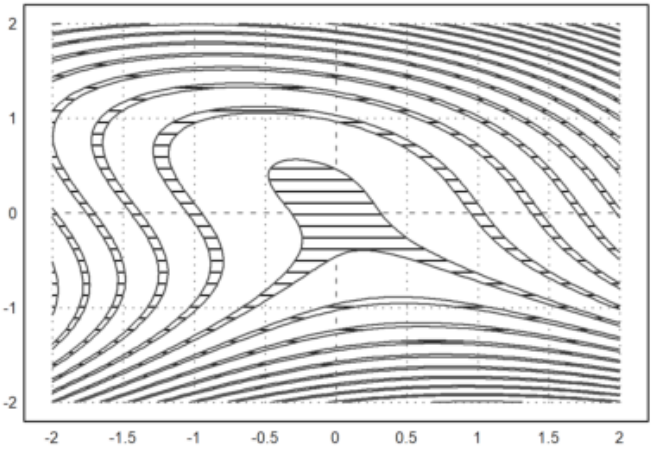
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



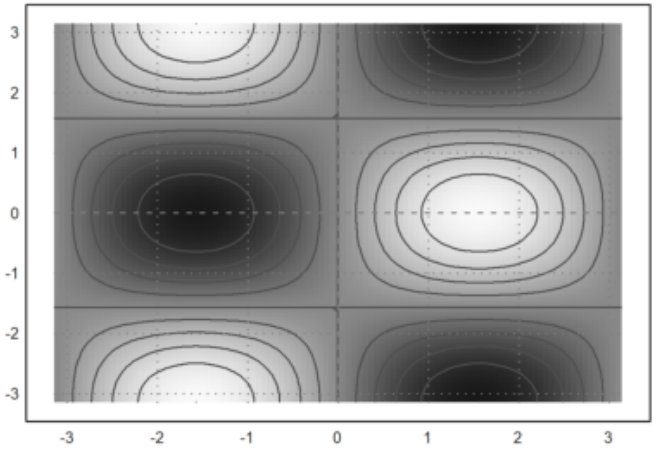

```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```




```
>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):
```

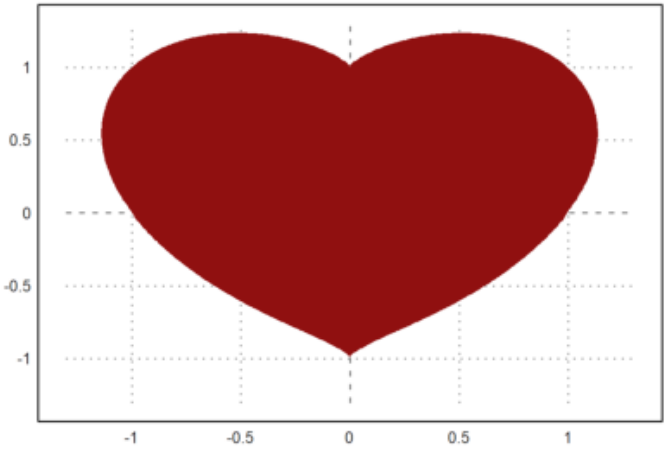


Anda juga dapat menandai suatu wilayah

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

Hal ini dilakukan dengan menambahkan level dengan dua baris.

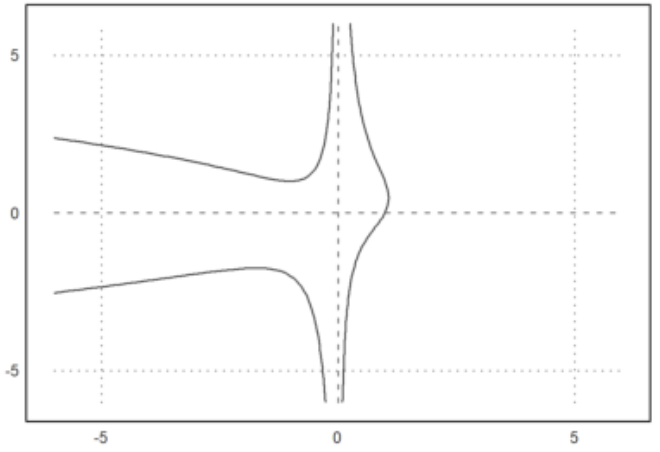
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
>  style="#",color=red,<outline, ...
>  level=[-2;0],n=100):
```



Dimungkinkan untuk menentukan level tertentu. Sebagai contoh, kita dapat memplot solusi dari persamaan seperti

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

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




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

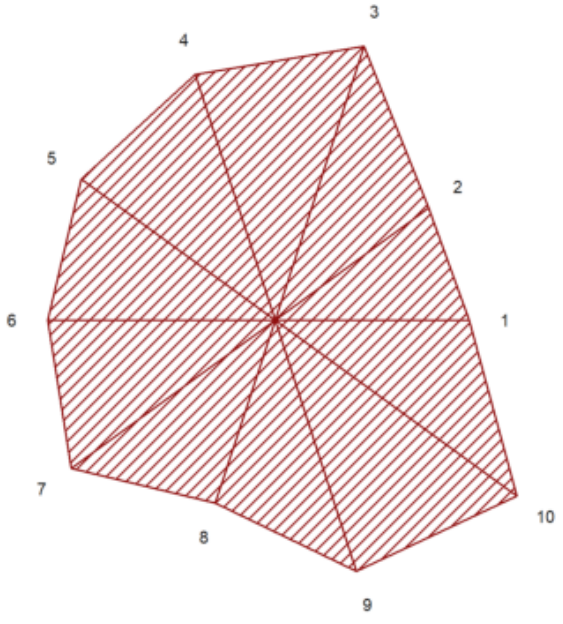
```
if !holding() then clg; endif;
w=window(); window(0,0,1024,1024);
h=holding(1);
r=max(abs(v))*1.2;
setplot(-r,r,-r,r);
n=cols(v); t=linspace(0,2pi,n);
v=v|v[1]; c=v*cos(t); s=v*sin(t);
cl=barcolor(color); st=barstyle(style);
loop 1 to n
  polygon([0,c[#],c[#+1]],[0,s[#],s[#+1]],1);
  if lab!=none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

```
    endif;
  end;
  barcolor(cl); barstyle(st);
  holding(h);
  window(w);
endfunction
```

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

Kami memanggil reset sebelum kami menguji plot ini untuk mengembalikan default grafis. Hal ini tidak perlu dilakukan, jika Anda yakin bahwa plot Anda berfungsi.

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



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

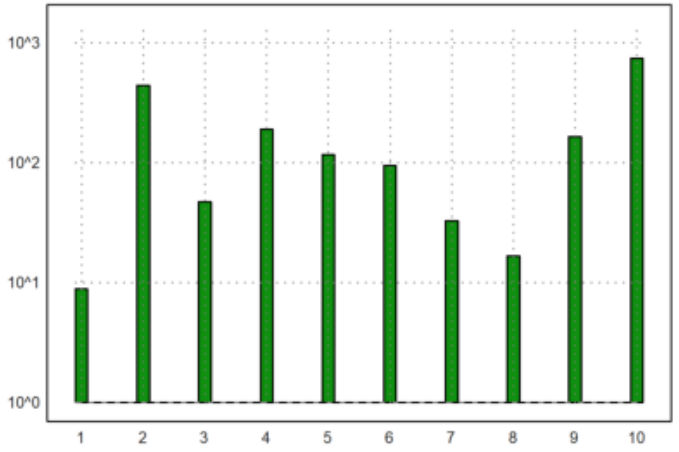
Pada fungsi berikut ini, kita akan membuat plot impuls logaritmik. plot2d dapat membuat plot logaritmik, tetapi tidak untuk batang impuls.

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

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

Mari kita uji dengan nilai yang terdistribusi secara eksponensial.

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



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

Fungsi wait(0) memaksa plot untuk muncul pada jendela grafik. Jika tidak, penggambaran ulang akan dilakukan dalam interval waktu yang jarang.

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

```
  t=linspace(0,2pi,500);
  f=0;
  c=framecolor(0);
  l=linewidth(2);
  setplot(-1,1,-1,1);
  repeat
    clg;
```

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

Press any key to stop this animation.

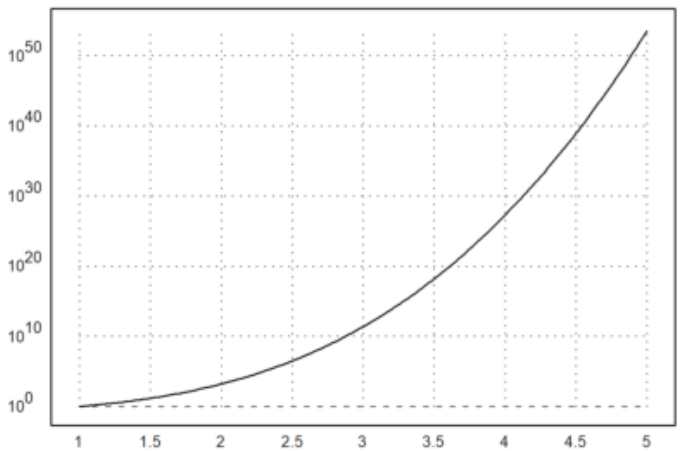
```
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

Plot Logaritmik

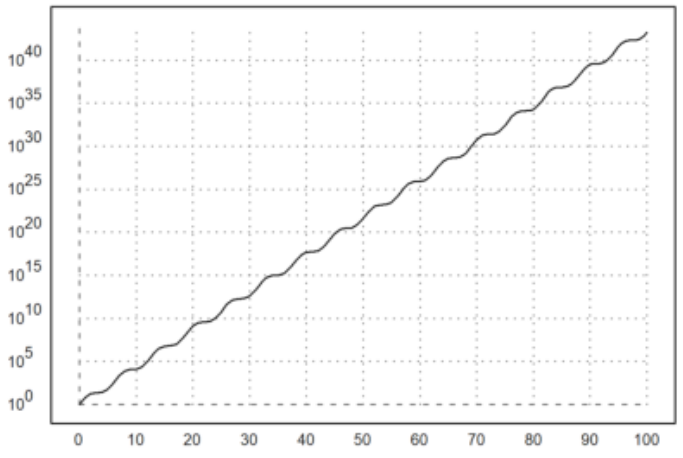
EMT menggunakan parameter ”logplot” untuk skala logaritmik.
Plot logaritmik dapat diplot menggunakan skala logaritmik dalam y dengan logplot = 1, atau menggunakan skala logaritmik dalam x dan y dengan logplot = 2, atau dalam x dengan logplot = 3.

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

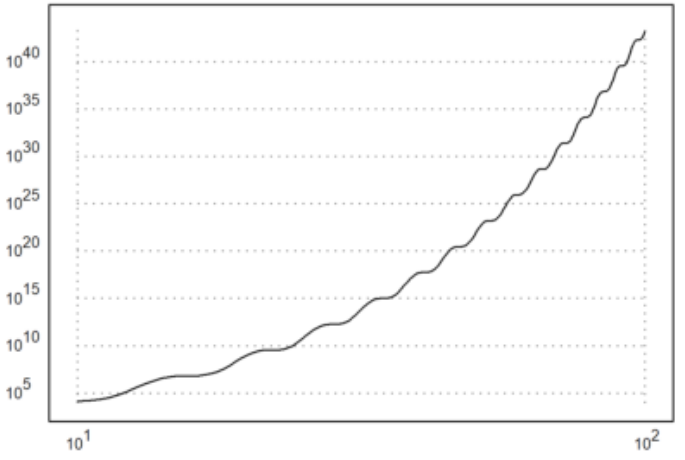
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



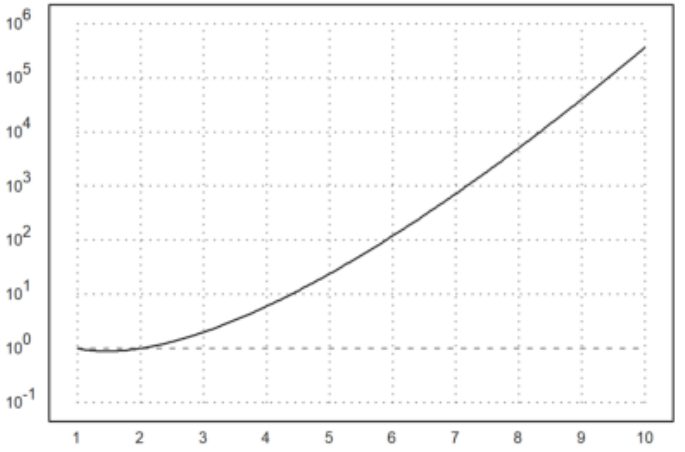
```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



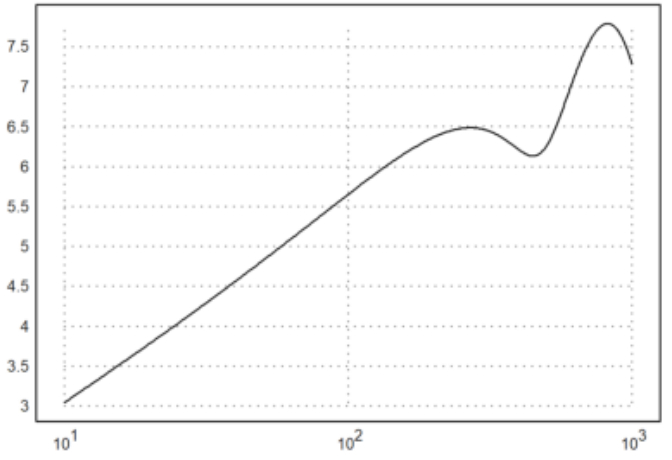

```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```




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

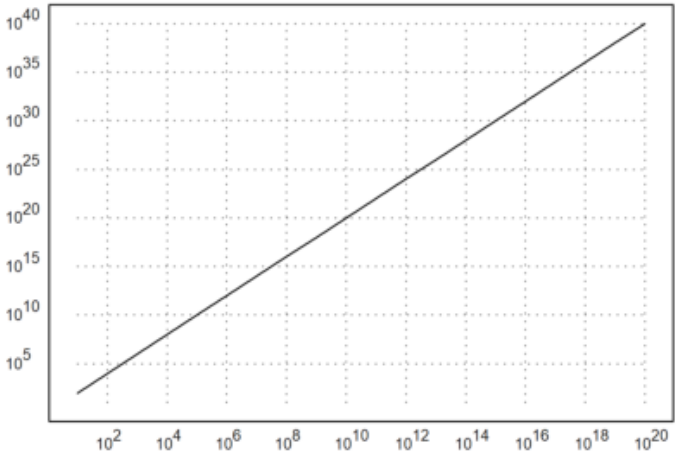



```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Hal ini juga bisa dilakukan dengan plot data.

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

```
Mengaktifkan atau menonaktifkan label vertikal. Hal ini mengubah
```

variabel global

```
verticallabels 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* Rujukan Lengkap Fungsi plot2d()
```

```
function plot2d (xv, yv, btest, a, b, c, d, xmin, xmax, r, n, ..
logplot, grid, frame, framecolor, square, color, thickness, style,
```

..

```
auto, add, user, delta, points, addpoints, pointstyle, bar,
```

histogram, ..

```
distribution, even, steps, own, adaptive, hue, level, contour, ..
nc, filled, fillcolor, outline, title, xl, yl, maps, contourcolor,
```

..

contourwidth, ticks, margin, clipping, cx, cy, insimg, spectral, ..
cgrid, vertical, smaller, dl, niveau, levels)

Fungsi plot serbaguna untuk plot pada 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 dari 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)
grid : 0 = tidak ada kisi dan label,

```
1 = hanya sumbu saja,
2 = kisi normal (lihat di bawah ini untuk jumlah garis
```

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 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
even : menggunakan nilai antar untuk histogram otomatis.
steps : memplot fungsi sebagai fungsi langkah (langkah = 1,2)

sendiri :

```
  Sebuah string, yang menunjuk 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].

 Gambaran Umum

 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. Jangkauan 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.

materi

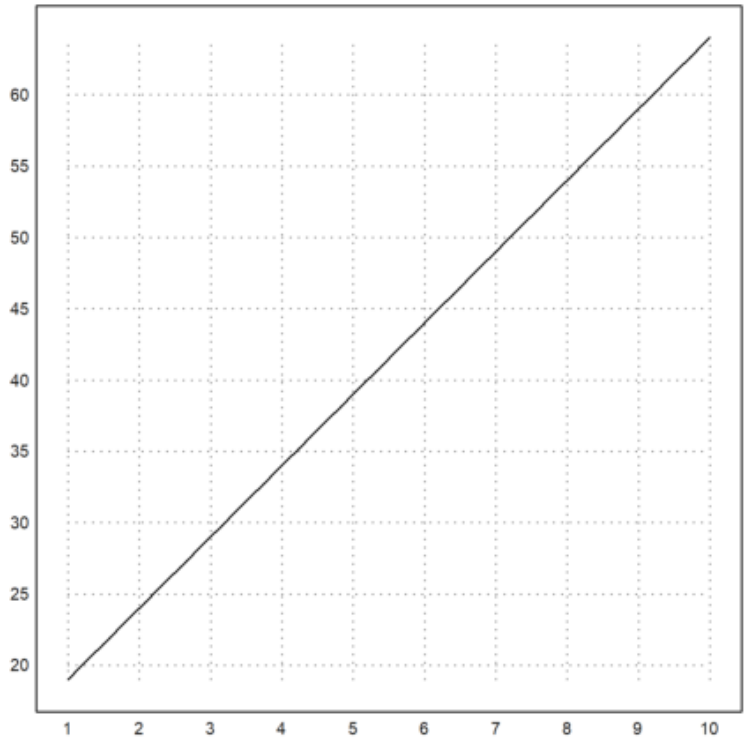
Menggambar Grafik Fungsi Satu Variabel

Grafik Fungsi Satu Variabel dalam Bentuk Ekspresi Langsung

Dalam menggambar grafik dalam bentuk ekspresi langsung, kita dapat langsung memasukkan ekspresi fungsi ke dalam perintah plot2d().

Sintaks:
plot2d(”ekspresi_fungsi”, rentang_x_min, rentang_x_max)

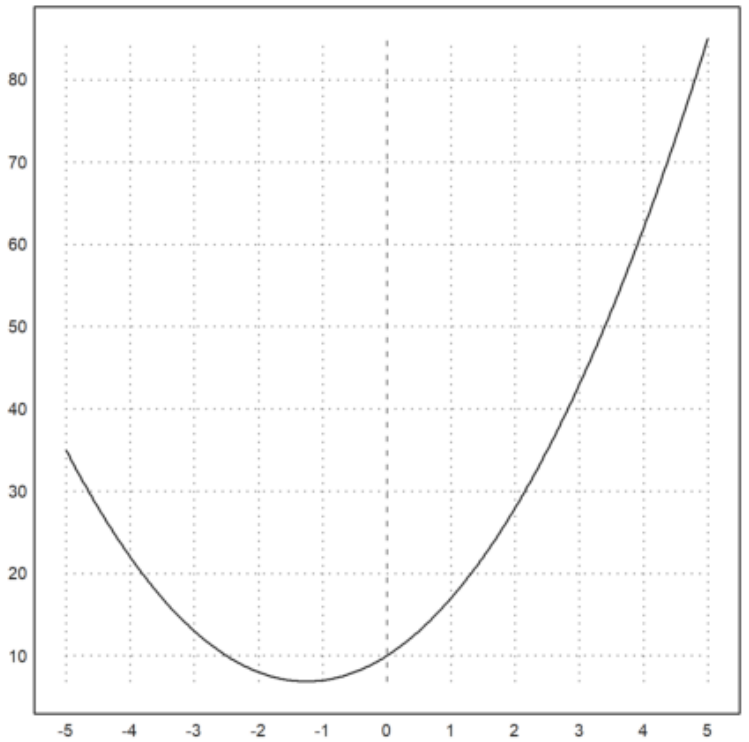
```
>//Penggunaan plot 2D pada fungsi linear
>plot2d("5x+14",1, 10)
>insimg
```



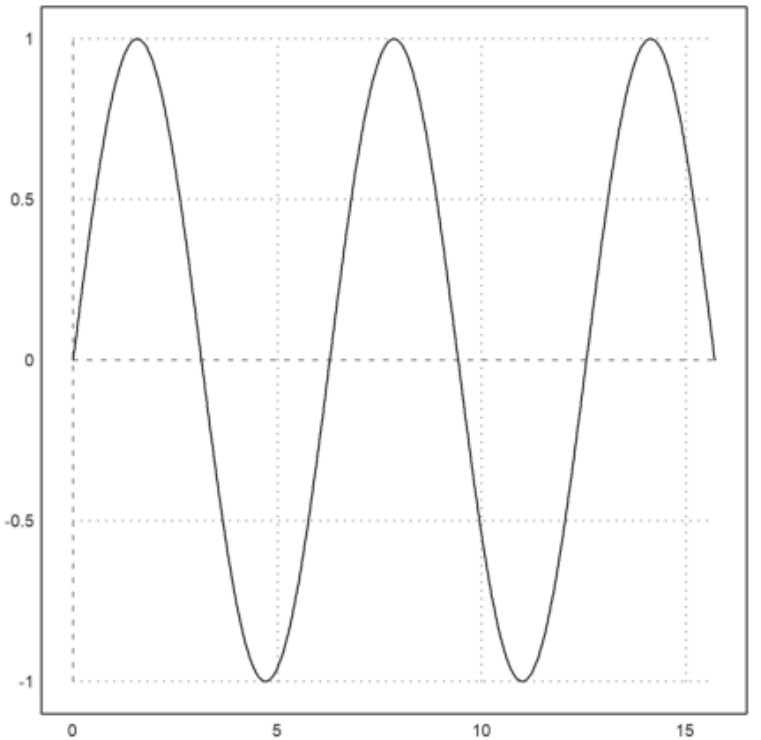
Tampilan plot akan menyesuaikan interval fungsi yang sudah diinputkan sebelumnya sehingga setiap plot memiliki sumbu x sepanjang intervalnya

Fungsi insimg digunakan untuk membuat plot bisa divisualisasikan di halaman utama EMT, tidak hanya di halaman grafik saja.

```
>
>//Penggunaan plot 2D pada fungsi kuadrat
>plot2d("2*x^2+5*x+10", -5, 5)
> insimg
```

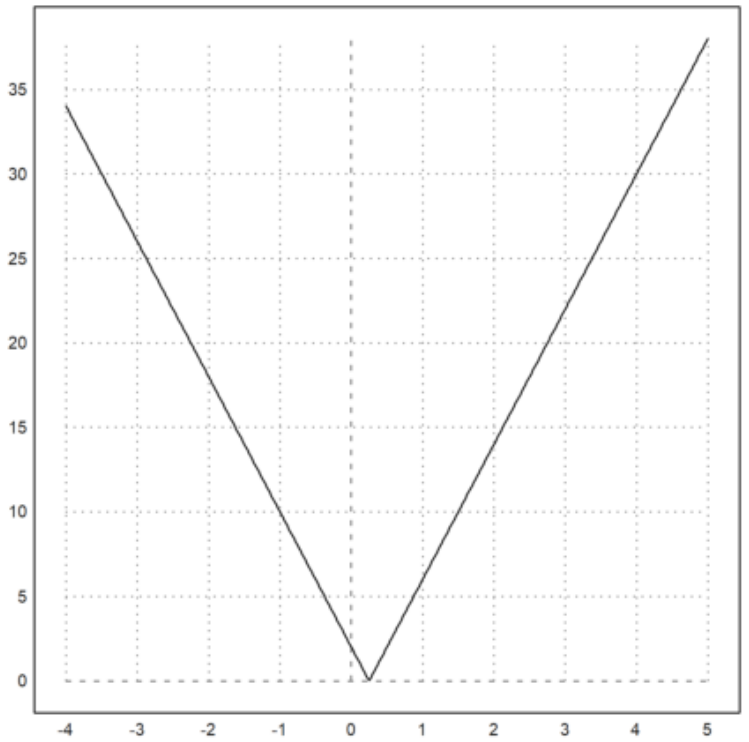


```
>
>//Penggunaan plot 2D pada fungsi trigonometri
>plot2d("sin(x)", 0, 5*pi)
> insimg
```



Dalam kurva fungsi trigonometri, sumbu x yang tertera adalah nilai dari pi, bukan dalam bentuk pi atau kelipatannya

```
>
>//Penggunaan plot 2D pada fungsi nilai mutlak
> plot2d("abs(8x-2)",-4,5)
>insimg
```



Dalam penulisan fungsi nilai mutlak dalam sintaks, tidak menggunakan simbol 2 garis tegak, melainkan menggunakan abs atau absolute.

```
>clg
```

Fungsi clg digunakan untuk menghapus plot pada halaman grafik atau biasa disebut dengan clearscreen

```
>plot2d("x^3-x");
>hold on;
>plot2d("x^4-x")
>clg
```

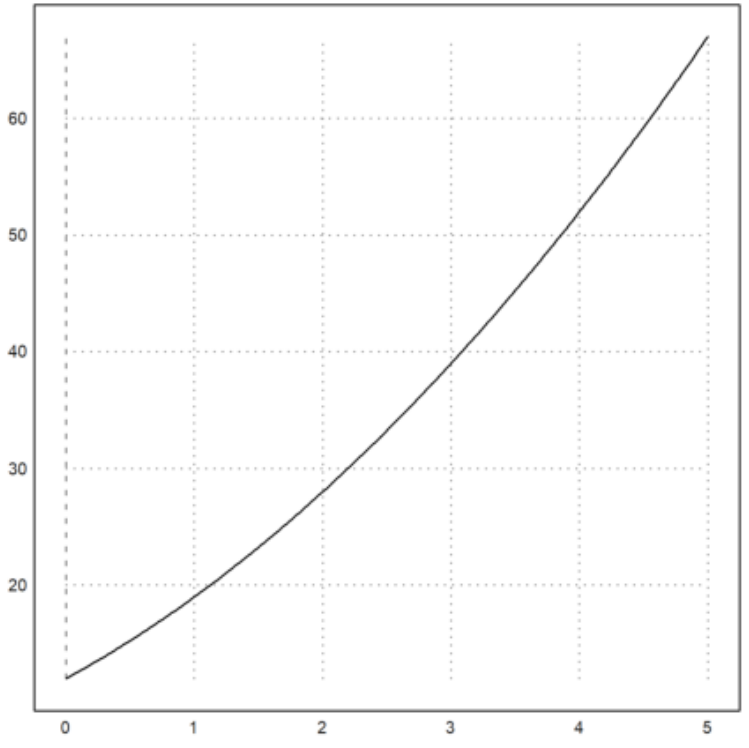
Menggambar grafik fungsi yang disimpan dalam variabel ekspresi

Dalam menggambar grafik satu variabel yang rumusnya disimpan dalam bentuk variabel ekspresi, kita dapat mendefinisikan terlebih dahulu variabel ekspresinya baru menggunakan plot2d.

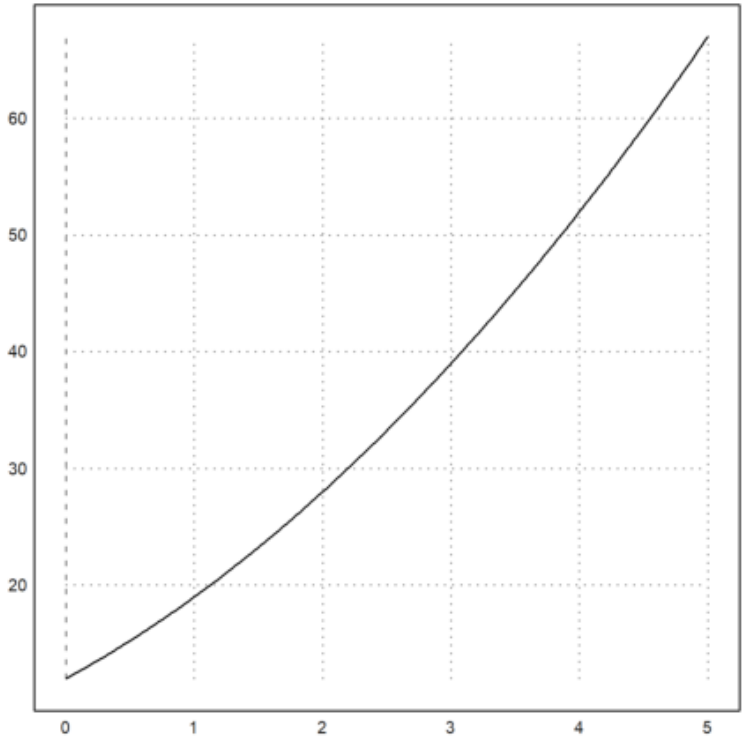
Langkah-langkah:
1. Mendefinisikan ekspresi fungsi ke dalam suatu nama variabel
2. Memasukkan nama variabel beserta interval ke dalam plot 2d

Sintaks:
function nama_variabel := ekspresi_fungsi
plot2d(”nama_variabel”, rentang_x_min, rentang_x_max)

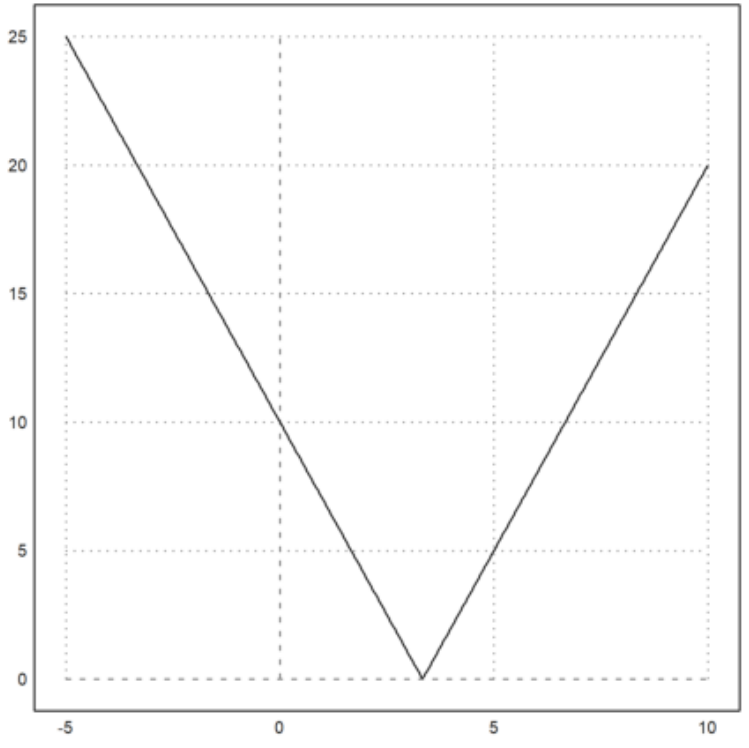
```
>function h(x):= x^2 + 6*x + 12
>plot2d("h", 0, 5)
> insimg
```




```
>function m(x):= sin(x)+tan(x)
>plot2d("m", 0, pi)
>insimg
```




```
>function n(x):= abs(-3x + 10)
>plot2d("n", -5, 10)
> insimg
```


25
20
15
10
5
0
-5
0
5
10

Fungsi Numerik

Fungsi satu variabel yang didefinisikan sebagai fungsi numerik adalah fungsi dengan satu variabel bebas yang dirumuskan khusus untuk perhitungan angka, sehingga bisa dievaluasi, digrafikkan, dan diolah secara numerik.

Langkah Menggambar Grafik

Untuk mendefinisikan sebuah fungsi yaitu dengan menggunakan
function f(x):=(), penggunaan :=() mendefinisikan bahwa fungsi tersebut adalah fungsi numerik.

Contoh:

$$f(x) = x^2 - 4x + 3$$

-Gambarkan grafik fungsi f(x) pada interval [-2,6]

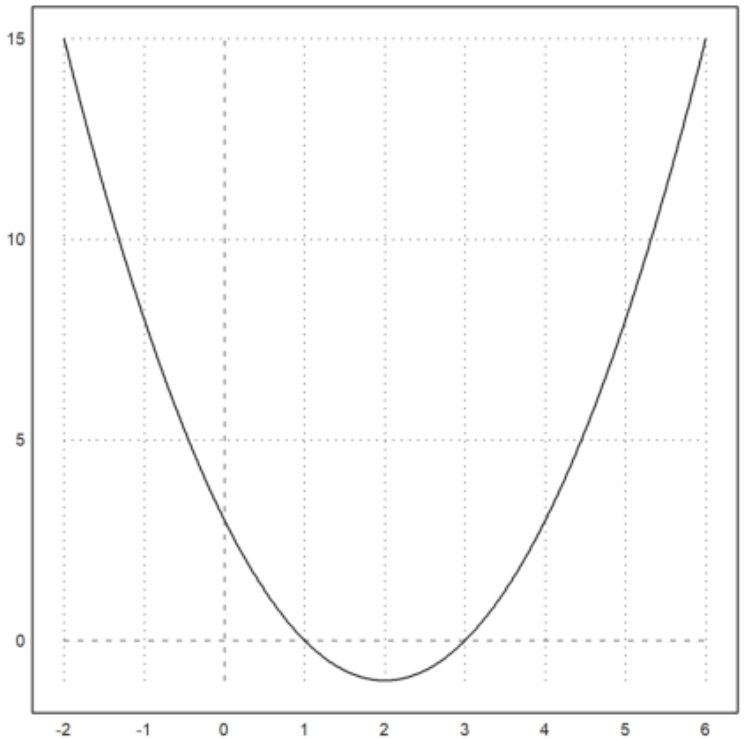
Langkah:

`-definisikan fungsi`

```
>function f(x):=(x^2-4*x+3)
```

-gambar grafik dengan memasukkan interval

```
>plot2d("f",-2,6):
```



untuk fungsi yang berfungsi untuk fungsi atau ekspresi lain, dapat meneruskan parameter tambahan (selain x) yang bukan variabel global ke fungsi tersebut dengan parameter titik koma atau dengan koleksi pemanggilan.

Contoh:

$$f(x) = ax^2 + \frac{x^2}{a} - x$$

-Diketahu a=4,gambarkan grafik fungsi f(x) pada interval [0,1]

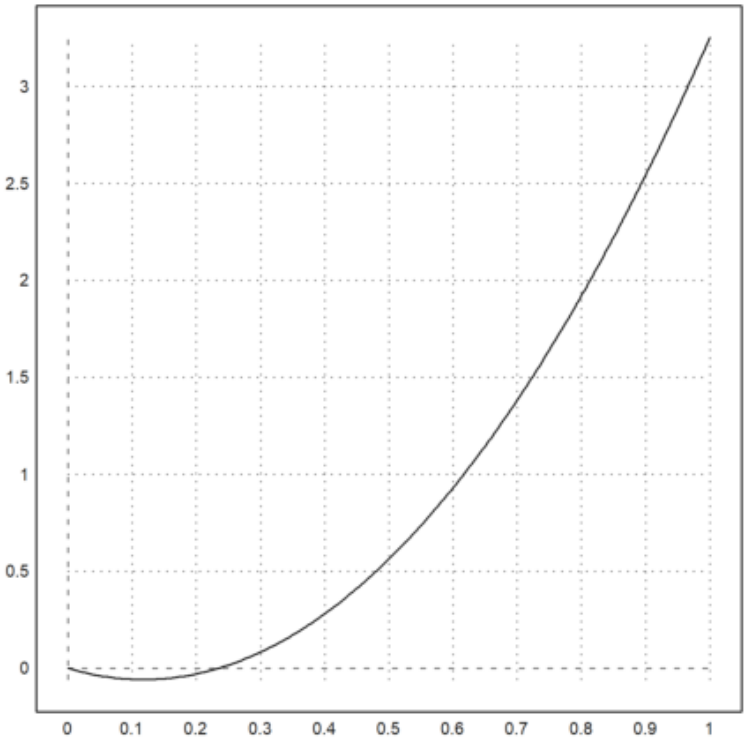
Langkah:

-definisikan fungsi

```
>function f(x,a):=(a*x^2+x^2/a-x)
```

-gambar grafik dengan memasukkan nilai a dan interval

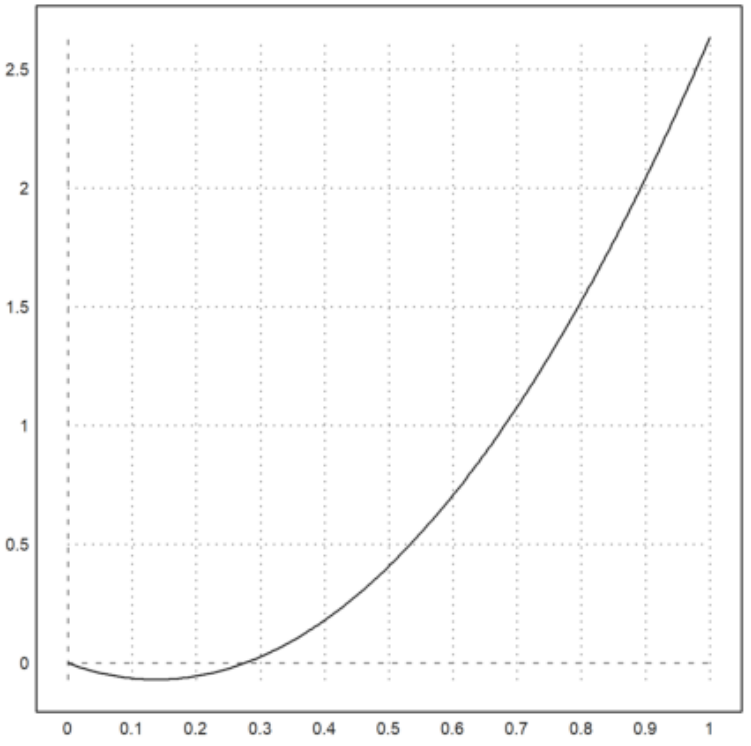
```
>a=4; plot2d("f",0,1;a):
```



Dapat juga menggunakan pemanggilan lain

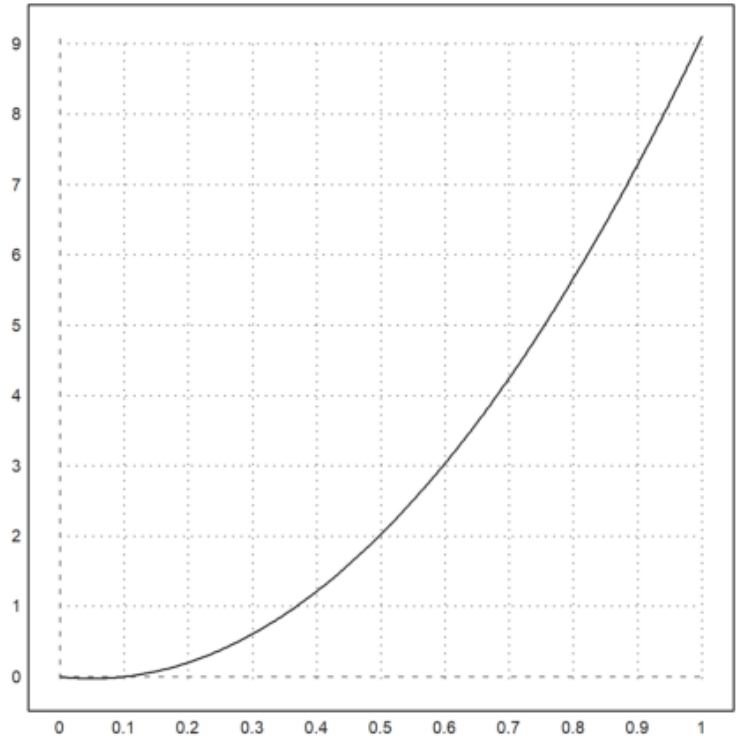
-ambil contoh nilai a=0,3

```
>plot2d("f",0,1;0.3):
```



-ambil contoh nilai a=0,1

```
>plot2d({{"f(x,a)",a=0.1}},0,1):
```


9
8
7
6
5
4
3
2
1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Fungsi satu variabel yang didefinisikan sebagai fungsi simbolik adalah fungsi dengan satu variabel yang ditulis sebagai ekspresi simbolik, sehingga bisa dimanipulasi aljabarnya (turunan, integral, faktorisasi)

Langkah Menggambar Grafik

Untuk mendefinisikan sebuah fungsi yaitu dengan menggunakan
function f(x)&=()atau menggunakan expr &=(), penggunaan &=() mendefinisikan bahwa fungsi tersebut adalah fungsi simbolik.

Contoh:

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

pada interval $0 \leq x \leq 10$

1.gambarlah grafik fungsi f(x)tersebut

Langkah:

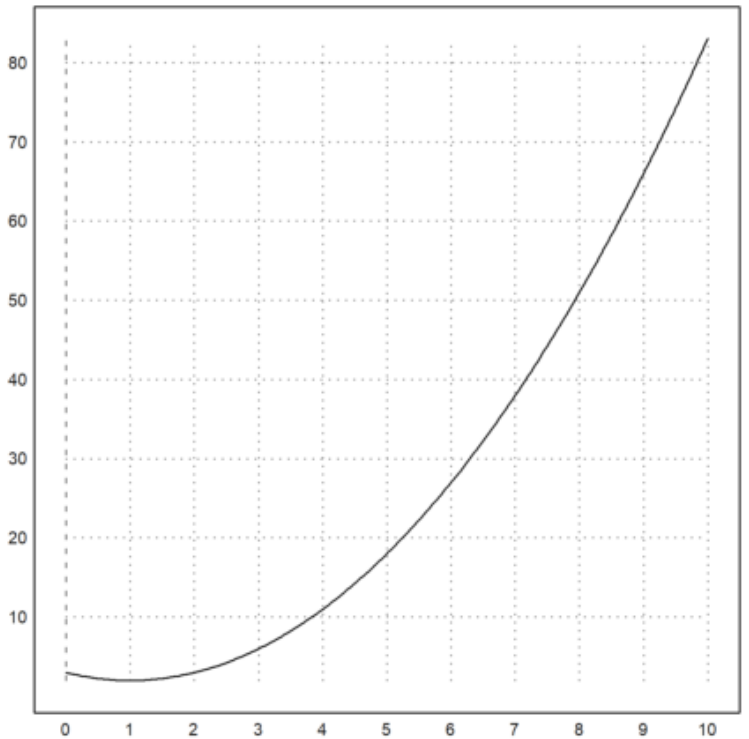
-mendefinisikan fungsi f(x)

```
>expr &= x^2-2*x+3
```

```
  2
x  - 2 x + 3
```

-gambar grafik dengan memasukkan fungsi dan interval

```
>plot2d (expr,0,10):
```



2. Tentukan f'(x) dan gambarkan grafik dengan interval yang sama

Langkah:

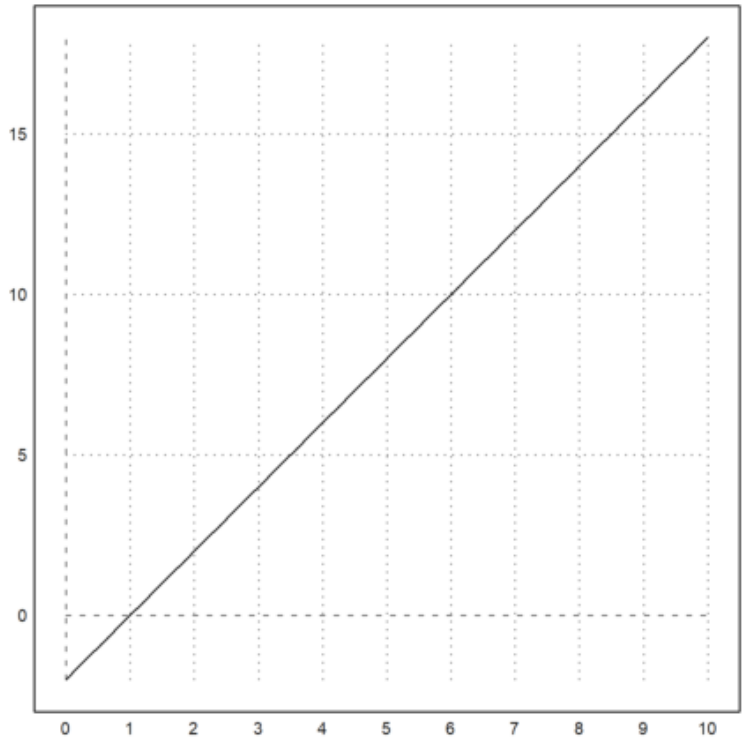
-menentukan nilai f'(x)

```
>expr &= diff (x^2-2*x+3,x)
```

$$2\,x - 2$$

-gambar grafik dengan memasukkan fungsi dan interval

```
>plot2d (expr,0,10):
```



3.Tentukan nilai dari

$$\int f(x)dx$$

dan gambarkan grafik dari dengan interval yang sama

Langkah:

-menentukan nilai

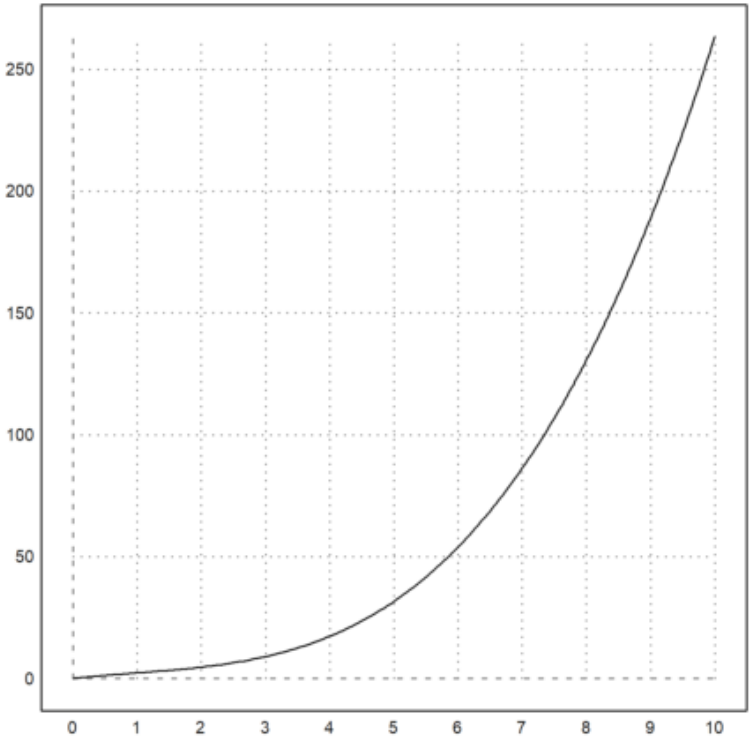
$$\int f(x)dx$$

```
>expr &= integrate (x^2-2*x+3,x)
```

```
 3
x     2
-- - x  + 3 x
3
```

-gambar grafik dengan memasukkan fungsi dan interval

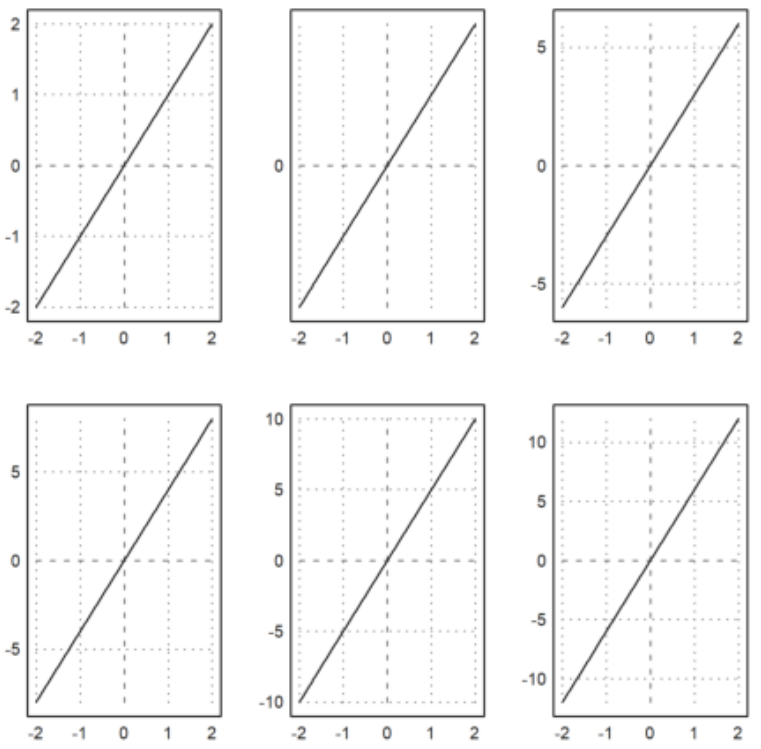
```
>plot2d (expr,0,10):
```



Menggambar Beberapa Kurva Sekaligus

Untuk menggambarkan beberapa kurva sekaligus dapat dilakukan dengan membuat jendela grafik menjadi beberapa bagian. akan menampilkan beberapa grafik yang terpisah dalam satu jendela.

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



Penjelasan dari plot fungsi

$$x * n, 1 <= n <= 6$$

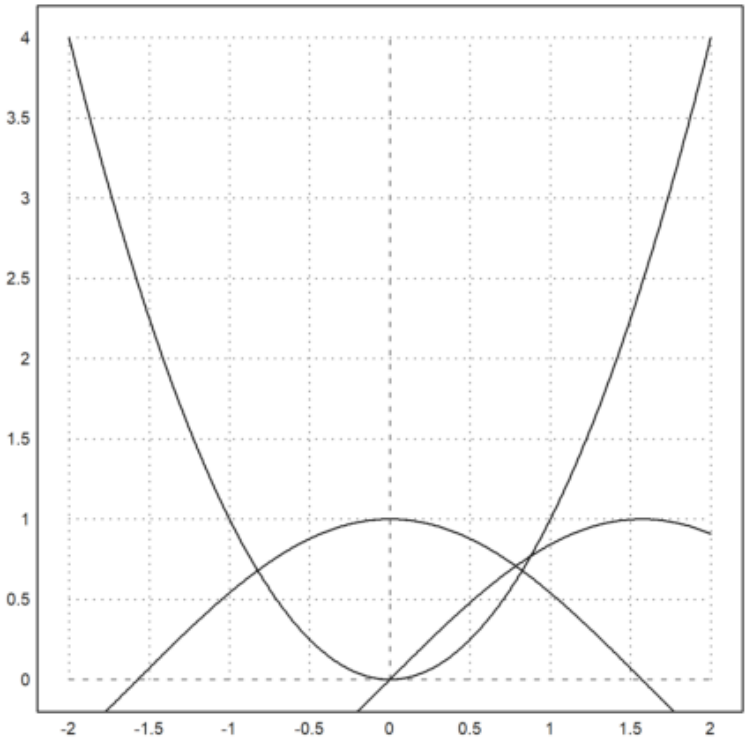
-figure(2x3)
perintah ini digunakan untuk membuat jendela grafik ukuran 2x3, akan ada enam jendela baru
-for n=1 to 4
perintah ini digunakan untuk pengulangan atau looping sebanyak 6 kali, dari 1 hingga 6
-figure(n)
perintah ini digunakan untuk beralih dari jendela grafik satu ke lainnya
-perintah plot2d("x*"+n)
untuk membuat plot fungsi matematika. dalam hal ini, n adalah variabel yang sedang diulang, atau membuat plot x*1, x*2, x*3, x*4, x*5, x*6 dalam jendela grafik yang sesuai
-end
digunakan untuk mengakhiri looping
-figure(0)
perintah ini digunakan untuk kembali ke jendela grafik utama

Menggambar Beberapa Kurva pada Bidang Koordinat yang sama

menggambarkan plot lebih dari satu fungsi je dalam satu jendela yang sama dapat dilakukan dengan mengguanakan >add untuk beberapa panggilan ke plot2d secara keseluruhan, kecuali panggilan pertama.

menggambar kurva parabola y=xˆ2, kurva sinus, dan kurva kosinus dalam koordinat yang sama

```
>plot2d("x^2",-2,2);
>plot2d("sin(x)",-2,2,>add);
>plot2d("cos(x)",-2,2,>add):
```



Perintah pertama
>plot2d(”xˆ2”,-2,2);
baris ini akan membuat plot pertama yaitu menampilkan kurva parabola y=xˆ2

Perintah kedua
>plot2d(”sin(x)”,-2,2,>add);
Baris ini akan menambahkan kurva sinus ke plot yang sudah berisi parabola, >add disini berfungsi untuk menambahkan kurva sinus ke plot yang sudah ada

perintah ketiga
>plot2d(”cos(x)”,-2,2,>add);
Baris ini akan menambahkan kurva kosinus ke plot yang sudah berisi parabola dan kurva sinus. >add disini berfungsi untuk menambahkan kurva kosinus ke plot yang sudah ada

```
>clg
```

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

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

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

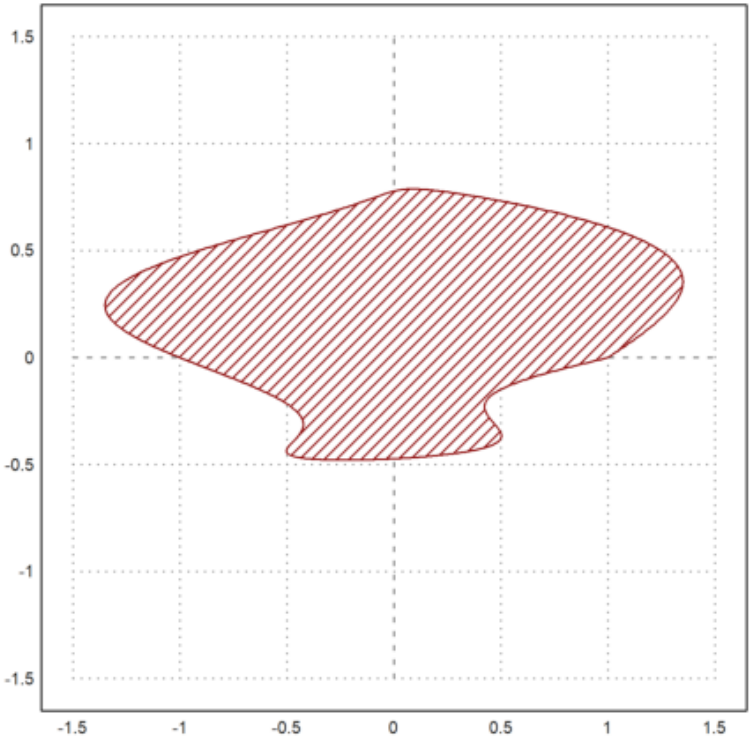
```
>t=linspace(0,1,1000);
>plot2d(t*cos(2*pi*t),(t*sin(2*pi*t)),r=1);
```

Atau, dimungkinkan untuk menggunakan dua ekspresi untuk kurva

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

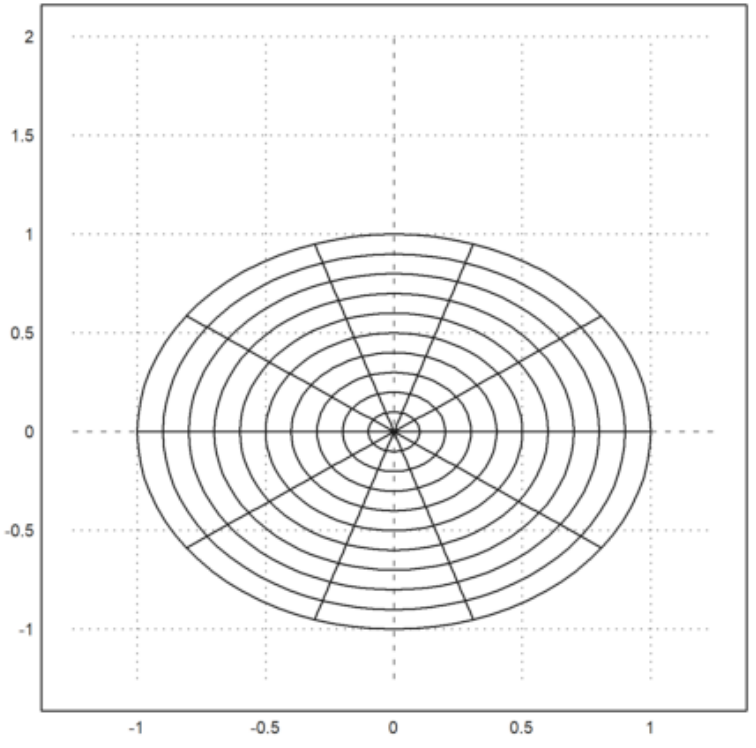
Dalam contoh berikutnya, kami memplot kurva
lateks:\gamma(t)=(r(t)\cos(t),r(t)\sin(t))
dengan
lateks:r(t)=1+\dfrac{\sin(3t)}{2}.

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

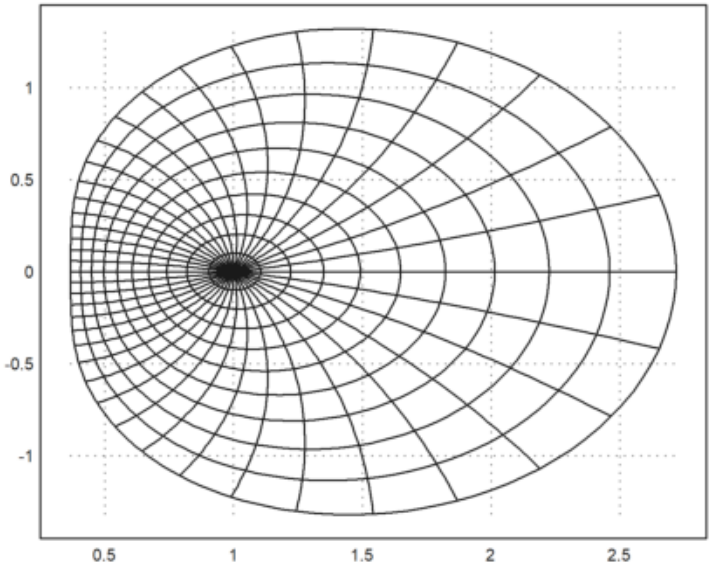

1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5

Array bilangan kompleks juga dapat diplot yang kemudian titik-titik grid akan terhubung. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi 1x2) dalam argumen cgrid, hanya garis kisi tersebut yang terlihat.
Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi dibidang kompleks.
Dalam contoh berikut, kami memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

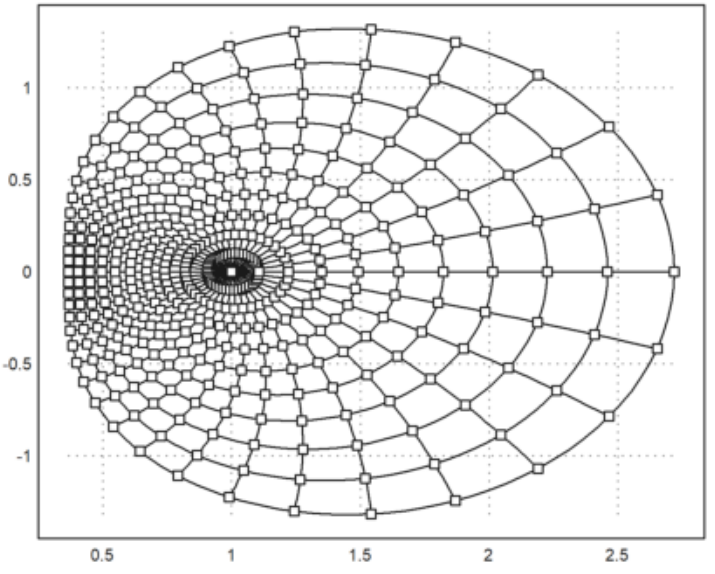
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);
>plot2d(z,a=-1.25,b=1.25,c=-1.25,cgrid=10):
```




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

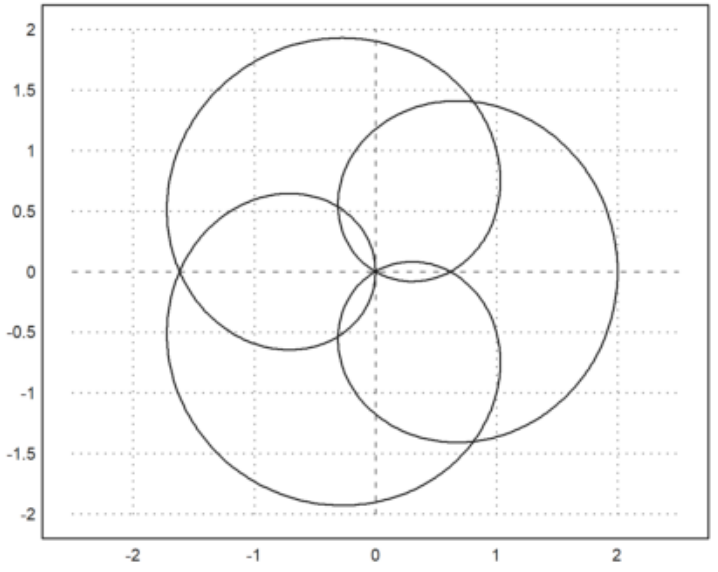



```
>r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=r*exp(I*a);
>plot2d(exp(z),>points,>add):
```



Sebuah vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian imajiner. Dalam contoh, kami memplot lingkaran satuan dengan latex:\gamma(t)=eˆ{it}

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



Menggambar Segibanyak

Fungsi perintah yang digunakan untuk menggambar segi banyak atau poligon, yaitu.

x=linspace(2,2pi,n);plot2d(cos(x),sin(x),r=1,>filled,style="..."):

atau dengan fungsi perintah berikut

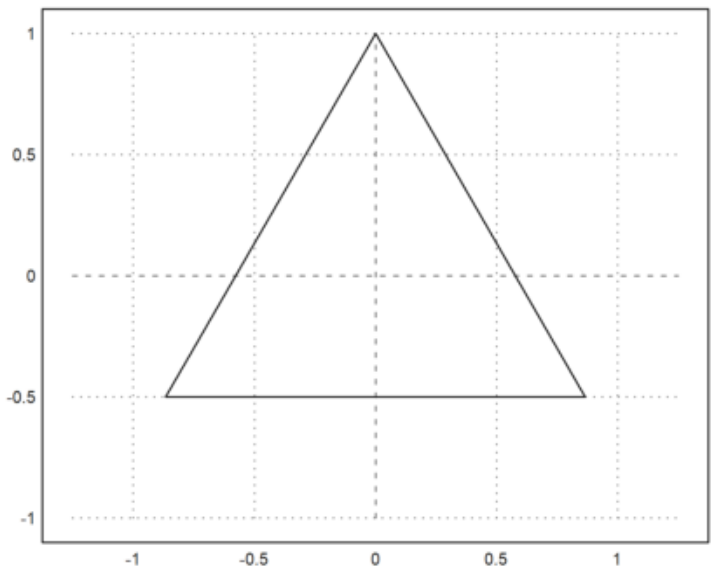
x=linspace(0,2pi,n);plot2d(sin(x),cos(x),r=1,>filled,styles="...",fillcolor=red):

Keterangan:
- filled=true untuk mengisi plot

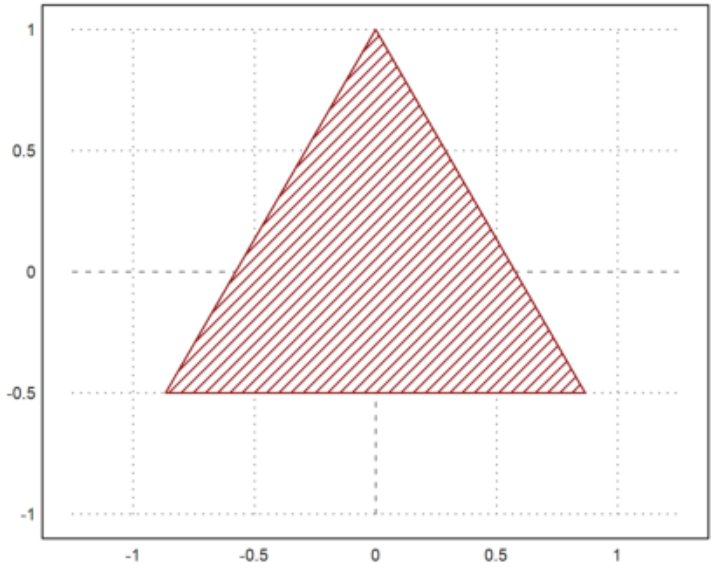
- styles"..." untuk memilih gaya plot, dapat "/","\" atau lainnya.
- fillcolor: untuk memberikan warna

A. Menggambar Segitiga

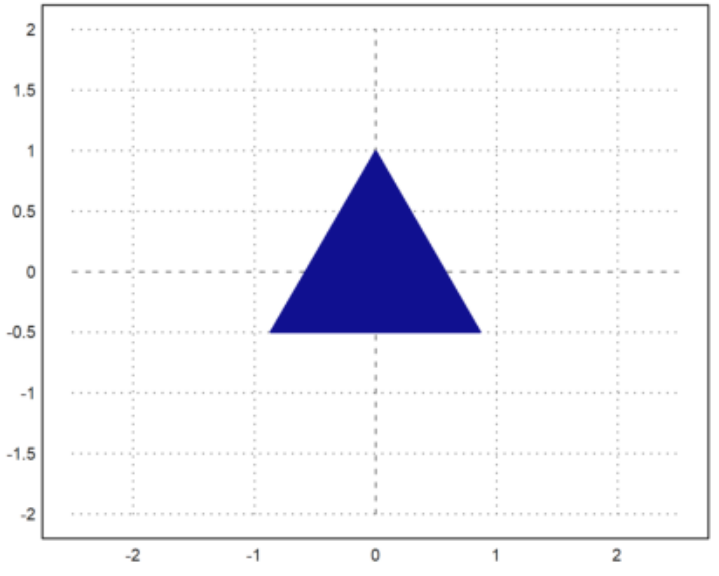
```
>x=linspace(0,2pi,3);...
>plot2d(sin(x),cos(x),r=1):
```




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

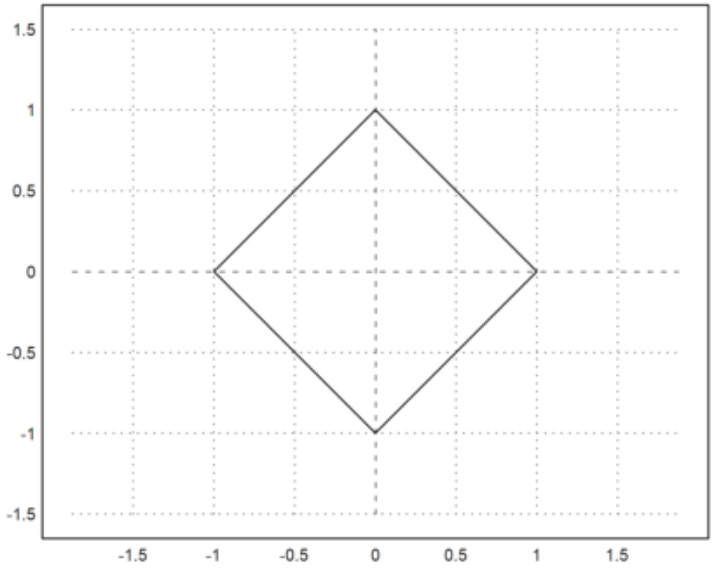



```
>x=linspace(0,2pi,3);...
>plot2d(sin(x),cos(x),>filled,style="#",fillcolor=blue,r=2):
```

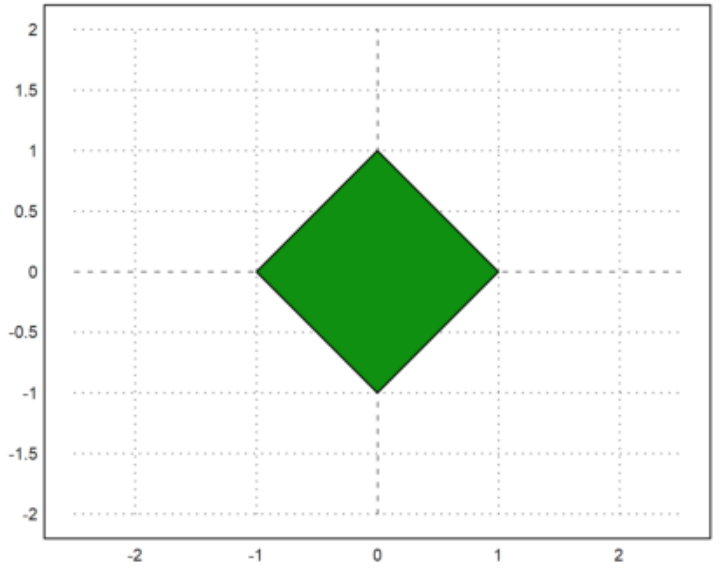


B. Menggambar Segiempat

```
>x=linspace(0,2pi,4);...
>plot2d(cos(x),sin(x),r=1.5):
```

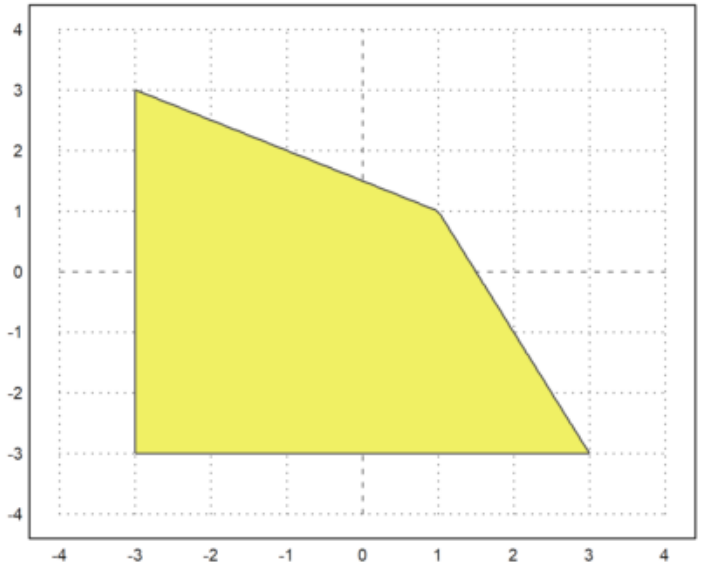



```
>x=linspace(0,2pi,4);
>plot2d(cos(x),sin(x),r=2,>filled,outline=1):
```



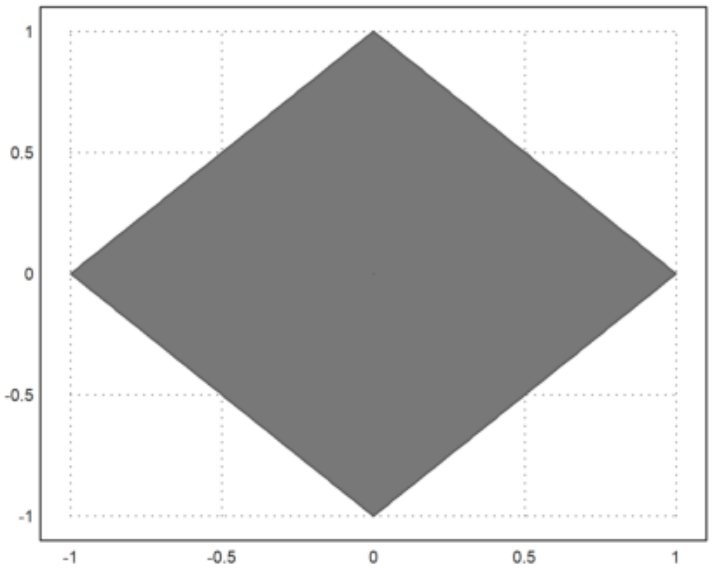
Selain menggubakan fungsi tersebut, dapat juga menggunakan fungsi maksimum. Fungsi maksimum digunakan untuk menggabungkan semua batasan linear (garis-garis batas) menjadi satu fungsi tunggal yang dapat diplot, sehingga menciptakan bentuk poligon.

```
>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=yellow,n=111):
```



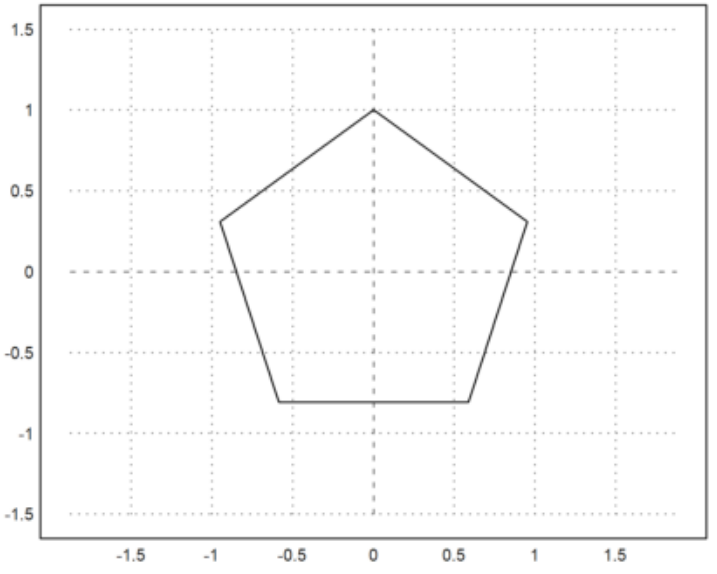
Keterangan:
- A adalah matriks A. Setiap garis dalam matriks ini adalah vektor normal dari garis-garis yang membentuk sisi poligon. Vektor ini mewakili koefisien x dan y dari setiap pertidaksamaan.
- Fungsi f(x,y) menghitung nilai maksimum dari serangkaian ekpresi linear.
- [x,y].A' melakukan perkalian matriks yang setara dengan menghitung nilai ax+by untuk setiap baris di matriks A.
- r=4 menetapkan rentang koordinat x dan y
- level=[0;3] merupakan parameter yang menentukan rentang nilai fungsi yang akan diwarnai.
- n adalah parameter yang menentukan jumlah titik data yang digunakan untuk menggambar plot. Jika n kecil, jarak antar titik yang dihitung menjadi terlalu jauh, akibatnya sudut-sudut poligon yang seharusnya tajam menjadi terlihat tumpul.

```
>A=[1,1;-1,1;-1,-1;1,-1];
>function f(x,y) :=max([x,y].A');
>plot2d("f",r=1,level=[0;1],color=gray,n=90):
```



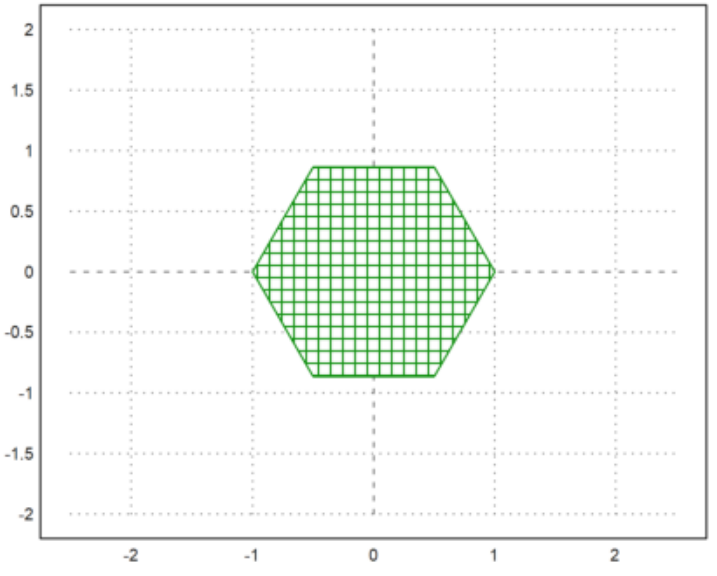
C. Menggambar Segilima

```
>t=linspace(0,2pi,5); plot2d(sin(t),cos(t),r=1.5):
```



D. Menggambar Heksagon

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



E. Menggambar Dekagon

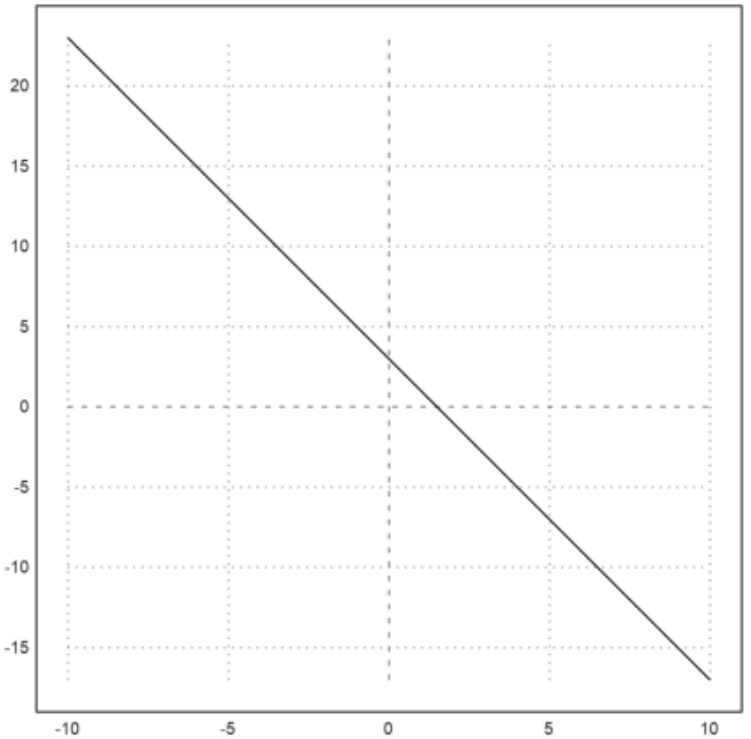
```
>t=linspace(0,2pi,10);
```

soal soal

1. sketsakan grafik

$$y = x2 - 4x + 3$$

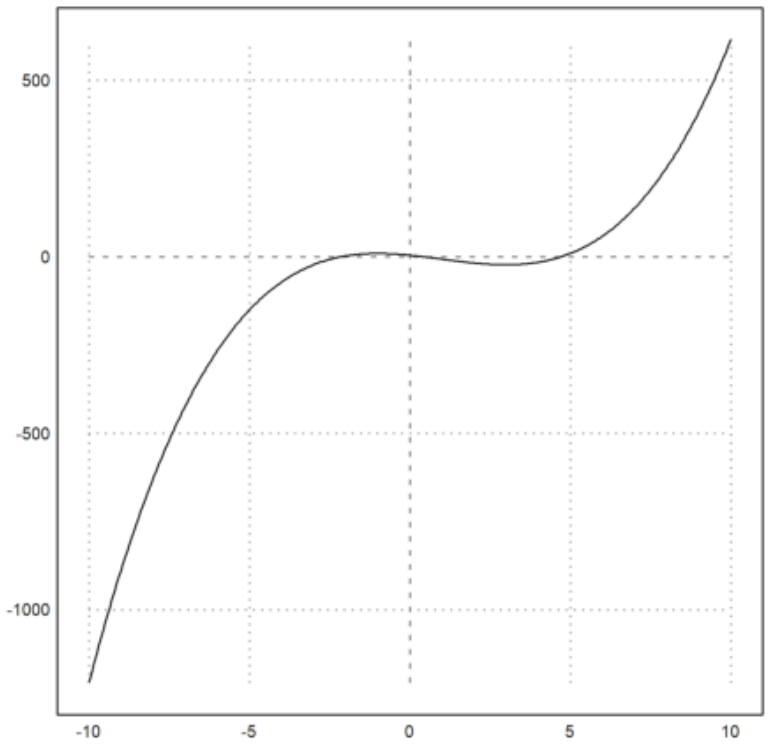
```
>plot2d("x*2-4*x+3",-10,10):
```



2. sketsakan Fungsi kubik

$$x^3 - 3x^2 - 9x + 5$$

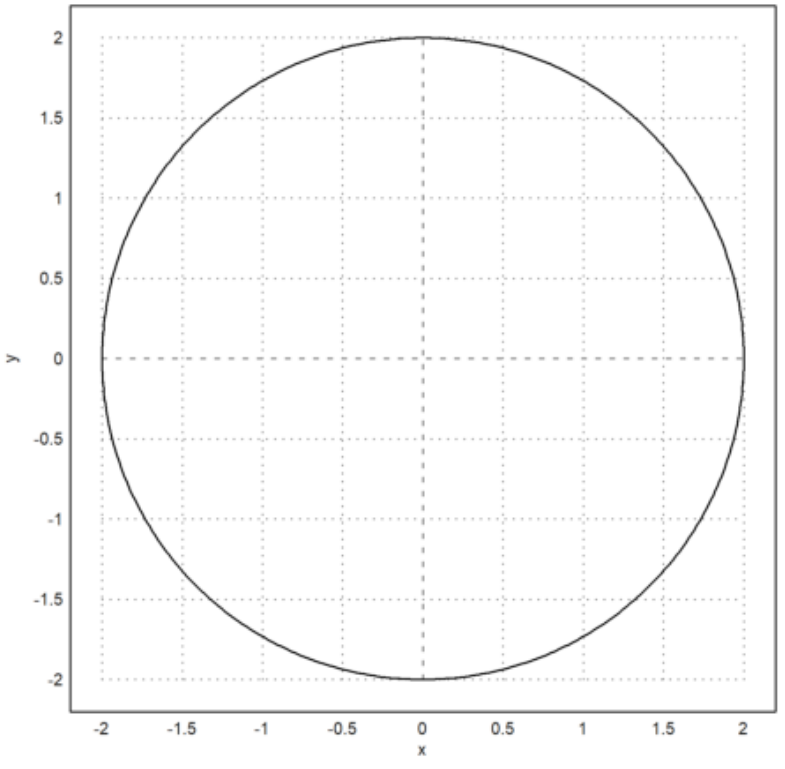
```
>plot2d("x^3-3*x^2-9*x+5",-10,10):
```



3. sketsakan

$$x = 2cost, y = 2sint, t \in [0, 2p]$$

```
>t = linspace(0, 2*pi, 400);
>xp = 2*cos(t);
>yp = 2*sin(t);
>plot2d(xp, yp); hold on;
>xlabel("x"); ylabel("y");
>insimg
```


2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
y
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
x