

Menggambar Grafik 2D dengan EMT

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

Plot-Plot Dasar

Ini adalah fungsi-fungsi plot yang sangat umum. Terdapat koordinat-koordinat layar, yang mana selalu berupa selang dari 0 hingga 1024 dalam setiap sumbu, tidak peduli apakah layarnya yang berupa persegi atau tidak. Dan disini adalah koordinat-koordinat plot, yang mana dapat diatur dengan `setplot()`. Pemetaan antara koordinat-koordinat tergantung pada jendela layar. Sebagai contoh, bawaan `shrinkwindow()` akan mengatur jarak untuk sumbu label-label dan sebuah judul plot.

Dalam contoh, kami hanya akan menggambar beberapa garis-garis acak dalam warna-warna yang berbeda. Untuk lebih jelasnya dalam fungsi-fungsi tersebut, pelajari fungsi-fungsi utama dalam EMT.

```
>clg; // membersihkan layar
>window(0, 0, 1024, 1024); // menggunakan semua rentang dalam jendela
>setplot(0, 1, 0, 1); // atur koordinat-koordinat plot
>hold on; // mulai dengan mode overwrite
>n = 100; X = random(n, 2); Y = random(n,2); // membuat titik-titik acak
>colors = rgb(random(n), random(n), random(n)); // mendapatkan warna-warna acak
>loop 1 to n; color(colors[#]); plot(X[#], Y[#]); end; // membuat plot
>hold off; // mengakhiri mode overwrite
>insimg; // menampilkan gambar ke notebook
```



```
>reset;
```

Ini diperlukan untuk menahan gambar-gambar, karena perintah `plot()` akan membersihkan layar plot.

Untuk memebersihkan semuanya, kami menggunakan `reset()`.

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

Untuk contoh lainnya, kami menggambar sebuah plot sebagai sisipan dalam plot lainnya. Ini dapat dilakukan dengan mendefinisikan sebuah layar plot yang lebih kecil. Catatan bahwa layar ini tidak menyediakan ruang untuk label-label sumbu yang diluar dari layar plot. Kami juga dapat menambahkan beberapa margin untuk ini apabila diperlukan. Catatan bahwa kami menyimpan dan mengembalikan layar secara penuh, dan menahan plot sekarang sementara kami mensisipkan plot.

```
>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);
```

Sebuah plot dengan gambar-gambar yang banyak dapat diperoleh dengan cara yang sama. Terdapat alat fungsi `figure()` untuk melakukan ini.

Aspek Plot

Plot bawaan menggunakan sebuah layar plot persegi. Kamu dapat menggubahnya dengan fungsi `aspect()`. Jangan lupa untuk mengatur ulang `aspect` nanti. Kamu juga dapat mengubah bawaan dalam menu dengan "Set Aspect" untuk sebuah rasio `aspect` secara spesifik atau untuk ukuran sekarang dari layar gambar.

Tapi kamu dapat merubahnya juga untuk satu plot. Untuk melakukan ini, ukuran sekarang dari luas plot dapat berubah, dan layar diatur jadi label-label memiliki ruang yang cukup.

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

Fungsi `reset()` mengembalikan plot ke pengaturan bawaan termasuk rasio aspek.

Plot-plot 2D dalam Euler

EMT Math Toolbox memiliki plot-plot dalam 2D, dua-duanya untuk data dan fungsi-fungsi. EMT menggunakan fungsi `plot2d`. Fungsi ini dapat mem-plotkan fungsi-fungsi dan data.

Sangat mungkin untuk mem-plotkan dalam Maxima menggunakan Gnuplot atau dalam Python menggunakan Math Plot Lib.

Euler dapat mem-plot plot-plot 2D dari

- ekspresi,
- fungsi-fungsi, variabel-variabel, atau kurva-kurva yang terparameter.
- vektor-vektor dari nilai-nilai x - y .
- Titik-titik dalam bidang.
- Kurva-kurva implisit dengan level-level atau level regional.
- Fungsi kompleks

Gaya plot termasuk gaya-gaya yang bervariasi untuk garis-garis dan titik-titik, plot-plot batang dan plot-plot terasir.

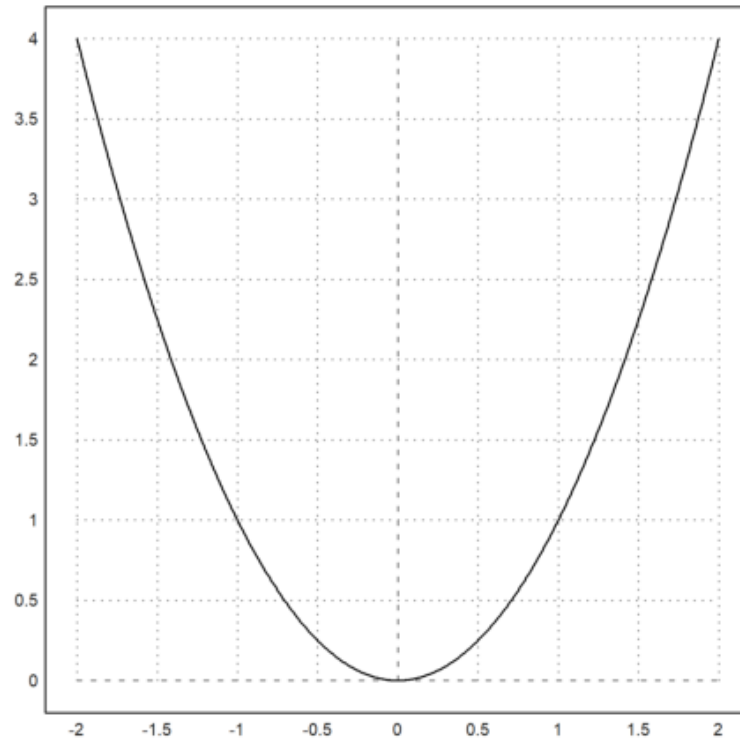
Plot Ekspresi atau Variabel

Ekspresi tunggal dalam "x" (seperti " $4*x^2$ ") atau nama sebuah fungsi (seperti "f") menghasilkan sebuah grafik dari fungsi.

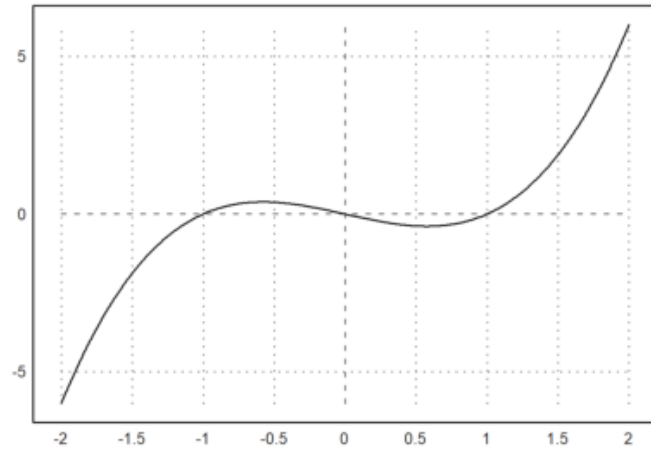
Ini adalah contoh umum, yang aman menggunakan selang bawaan dan himpunan-himpunan sesuai selang-y untuk menyesuaikan plot dari fungsi.

Catatan: Jika kamu mengakhiri sebuah baris perintah dengan sebuah titik dua ":", plot akan dimasukkan kedalam layar teks. Sebaliknya, tekan TAB untuk melihat plot jika layar plot tertutupi.

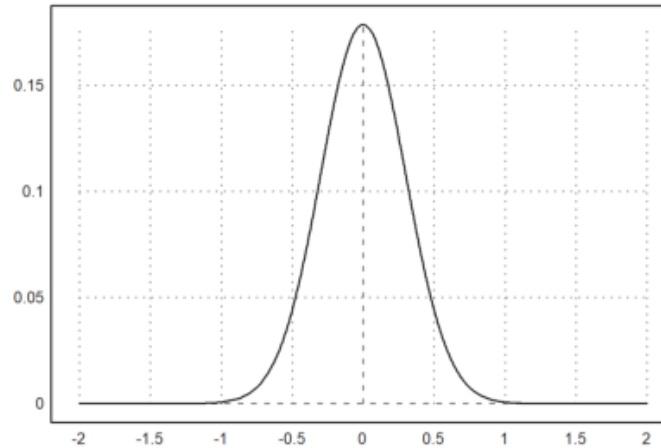
```
>plot2d("x^2"):
```



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



```
>a := 5.6; plot2d("exp(-a*x^2) / a"); insimg(30); // mengambilkan gambar hasil plot tertinggi setingg
```

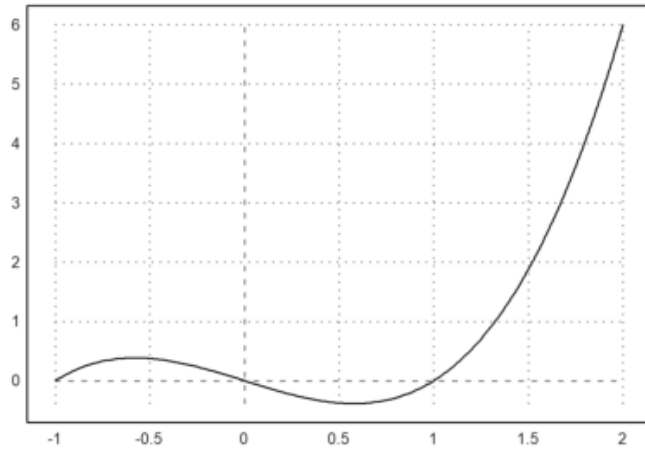



Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

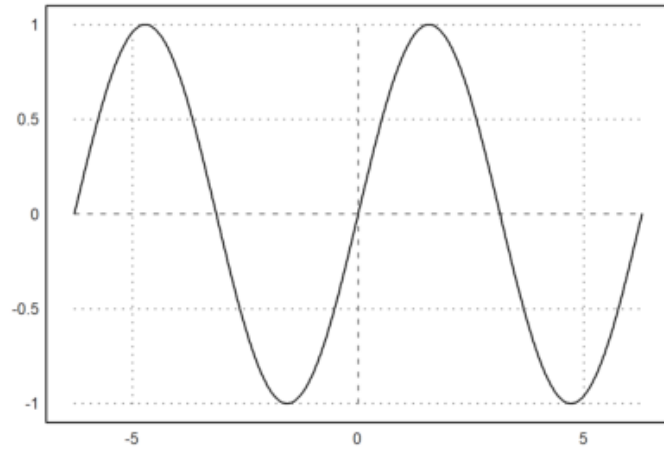
Selang dari plot diatur mengikuti parameter-parameter yang ditujukan

- a,b: x-range (bawaan -2,2)
- c,d: y-range (bawaan: skala dengan nilai-nilai)
- r: secara alternatif sebuah jari-jari mengelilingi plot lingkaran
- cx,cy: koordinat-koordinat dari tengah plot (bawaan 0,0)

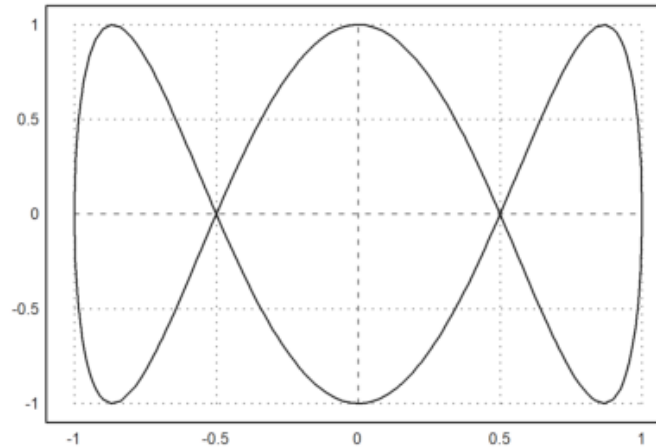
```
>plot2d("x^3-x",-1,2):
```



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



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



Sebuah alternatif untuk titik dua adalah perintah `insimg(lines)`, yang mana memasukkan plot-plot yang mempati sebuah angka yang spesifik dari baris-baris teks.

Dalam pilihan-pilihan, plot-plot dapat diatur untuk dimunculkan

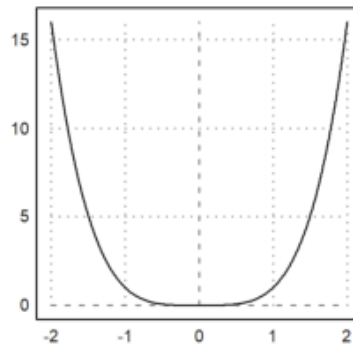
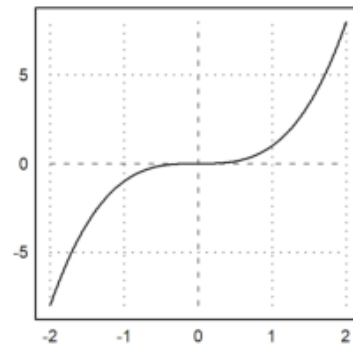
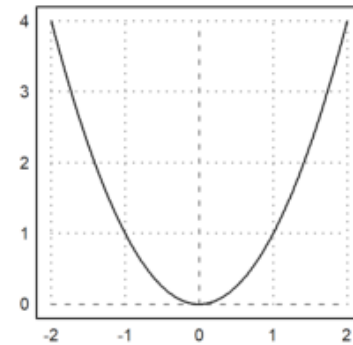
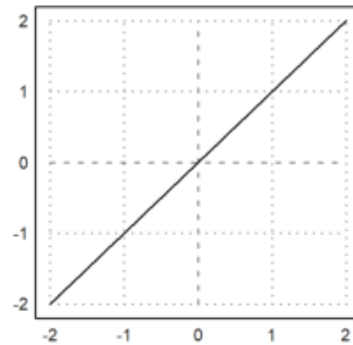
- dalam sebuah jendela yang dapat di ubah ukurannya secara terpisah,
- dalam jendela notebook.

Gaya-gaya lainnya yang dapat diperoleh dengan perintah-perintah plot yang spesifik

Dalam kasus yang lainnya, tekan tabulator untuk melihat plot, jika tersembunyi.

Untuk memisahkan jendela menjadi beberapa plot-plot, gunakan perintah `figure()`. Dalam contoh, kami membuat plot x^1 ke x^4 kedalam 4 bagian dari jendela. `figure(0)` mengatur ulang jendela bawaan.

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

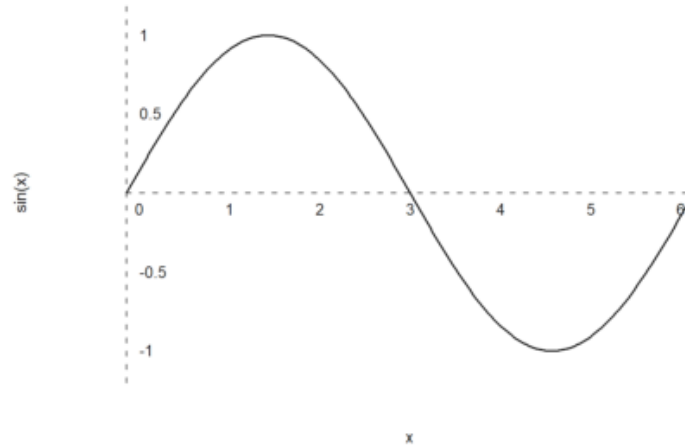


Jika dalam argumen-argumen untuk `plot2d()` adalah sebuah ekspresi yang diikuti oleh empat angka, ini angka-angka x - dan y -selang untuk plot.

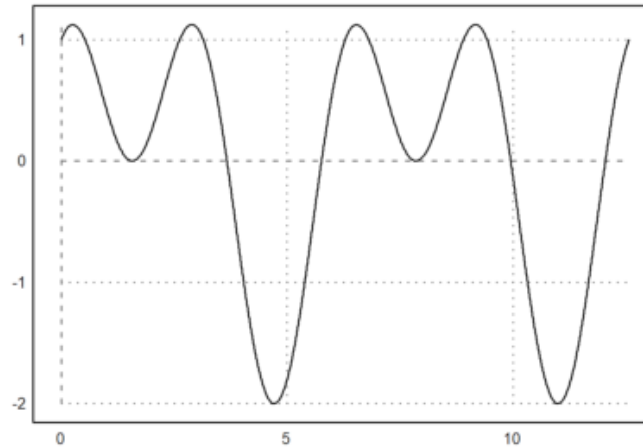
Sebagai alternatif, a , b , c , d dapat secara spesifik sebagai parameter-parameter yang ditujukan seperti $a = \dots$ dan lain-lain.

Seperti contoh berikut, kami mengubah gaya grid, menambah label-label, dan menggunakan label-label vertikan 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):
```

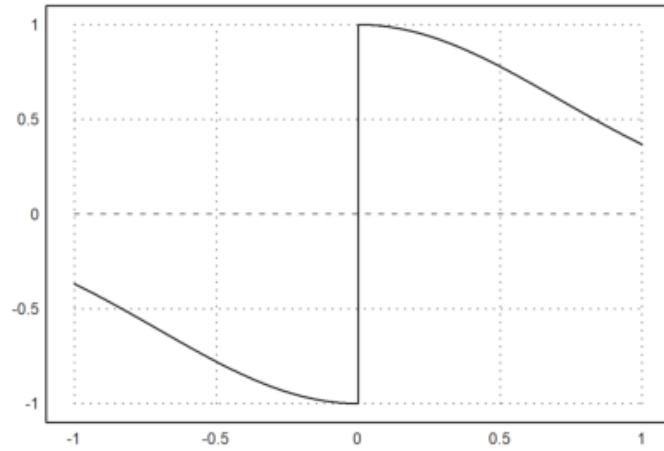


Gambar-gambar yang dihasilkan dengan memasukkan plot kedalam layar teks disimpan kedalam direktori seperti notebook, secara bawaan dalam sebuah sub-direktori bernama "images". Mereka juga menggunakan dengan ekspor HTML.

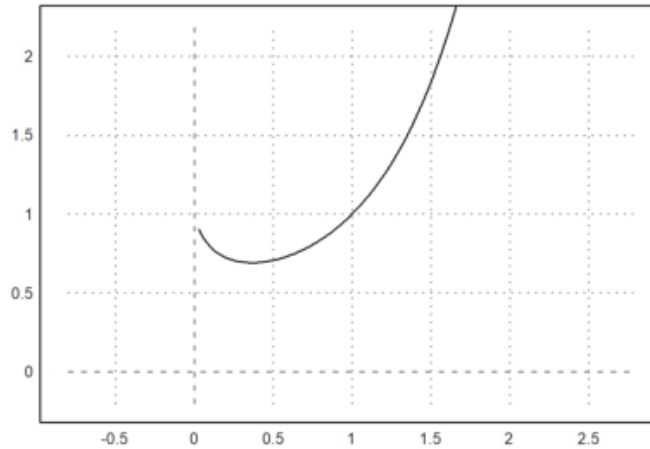
Kamu dapat secara mudah membuat tanda pada sebarang gambar dan mencetaknya kedalam clipboard dengan Ctrl + c. Tentu saja, kamu dapat juga mengekspor grafik-grafik sekarang dengan fungsi-fungsi dalam menu File.

Fungsi atau ekspresi dalam plot2d dapat dievaluasi secara adaptif. Untuk kecepatan yang lebih, matikan plot-plot adaptif dengan <adaptive dan spesifik angka dalam sub-interval dengan n = ... Ini seharusnya dibutuhkan dalam kasus yang sangat jarang.

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

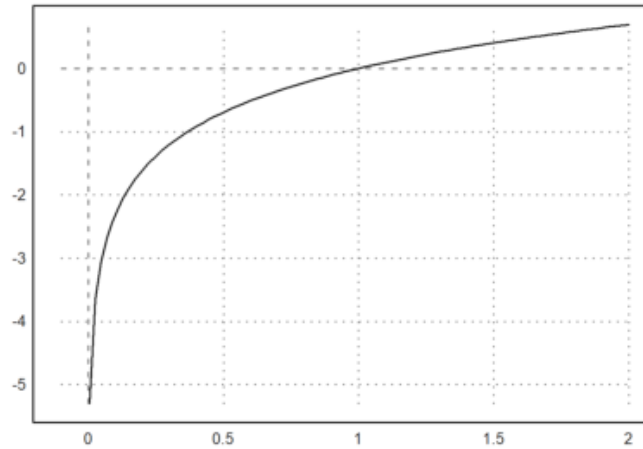


```
>plot2d("x^x", r = 1.2, cx = 1, cy = 1):
```

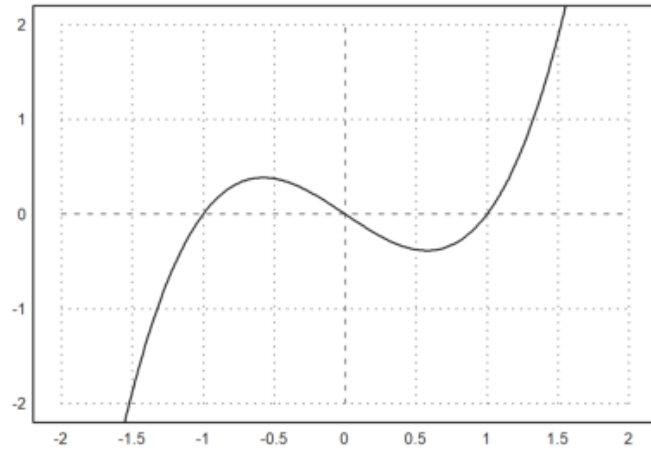
Catatan bahwa x^x tidak terdefinisi untuk $x \leq 0$. Fungsi `plot2d` mengetahui error ini, dan mulai membuat plot pada fungsi yang terdefinisi. Ini bekerja untuk semua fungsi yang mana mengembalikan NAN dari selang definisi.

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

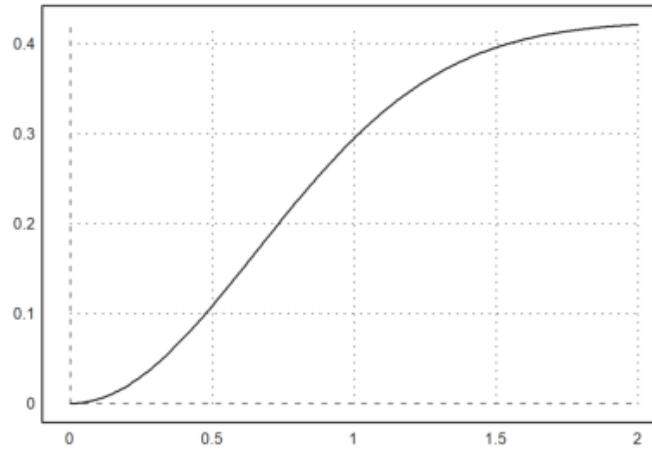


Parameter `square=true` (atau `>square`) memilih selang-y secara otomatis jadi bahwa hasil adah jendela kotak plot. Catatan bahwa secara bawaan, Euler menggunakan sebuah spasi kotak didalam layar plot.

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

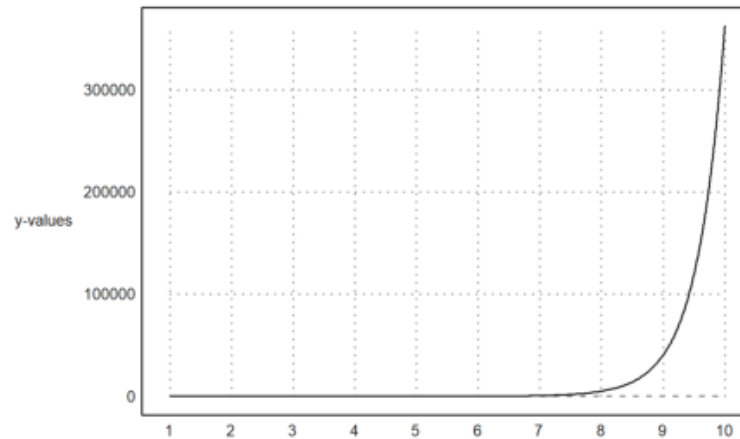


```
>plot2d(''integrate("sin(x) *exp(-x^2)", 0, x)'', 0, 2): // plot integral
```



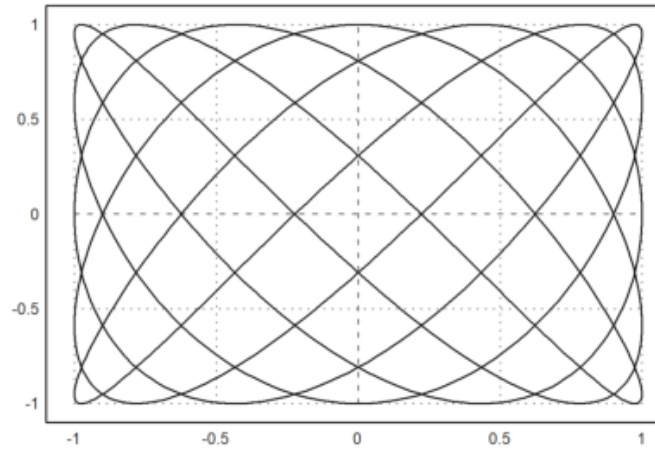
Jika kamu membutuhkan ruang yang lebih untuk label-y, panggil `shrinkwindow()` dengan parameter yang lebih kecil, atau atur sebuah nilai positif untuk "smaller" dalam `plot2d()`.

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

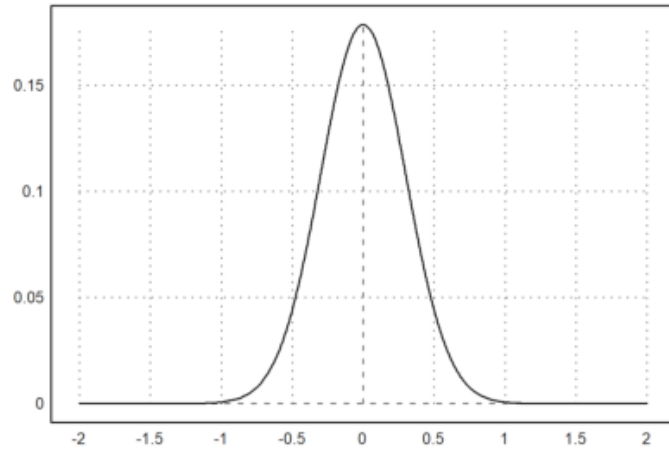


Ekspresi-ekspresi simbolik dapat juga digunakan, karena mereka disimpan sebagai ekspresi-ekspresi string yang sederhana.

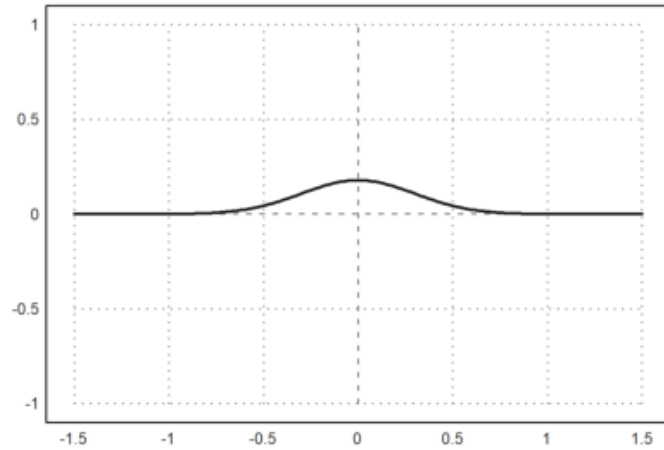
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



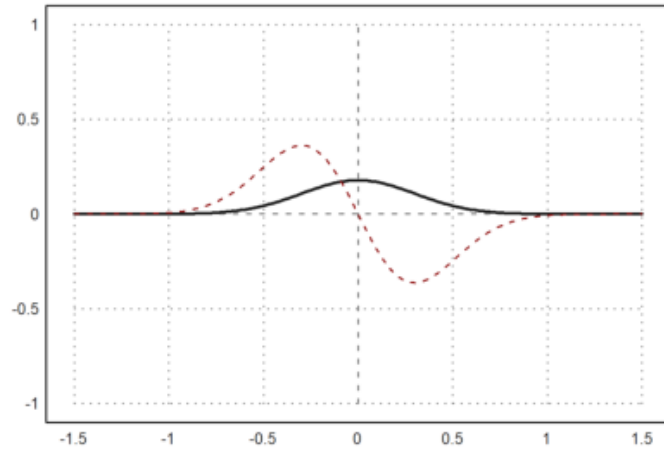
```
>a:=5.6; expr &= exp(-a*x^2)/a; // mendefinisikan ekspresi  
>plot2d(expr,-2,2): // membuat plot dari -2 ke 2
```



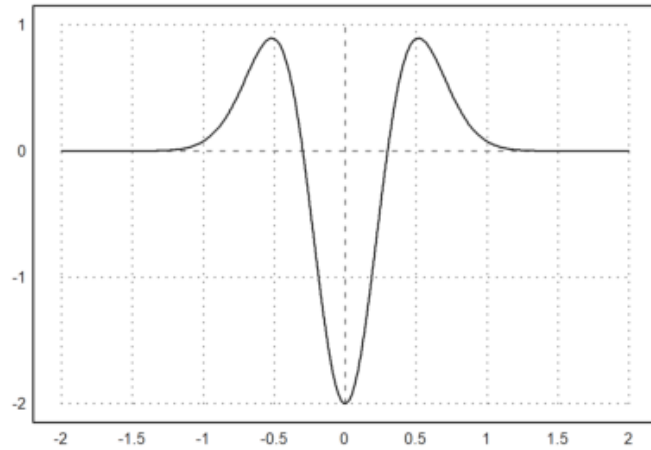
```
>plot2d(expr,r=1,thickness=2): // membuat plot dalam sebuah persegi sekitar (0,0)
```



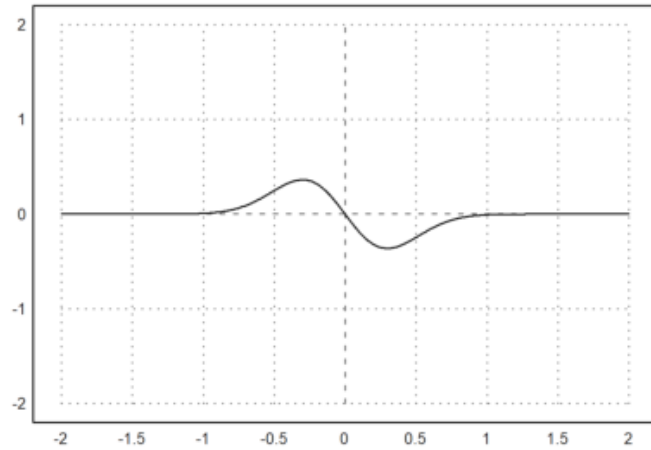
```
>plot2d(&diff(expr,x),>add,style="--",color=red): // menambahkan plot lainnya
```

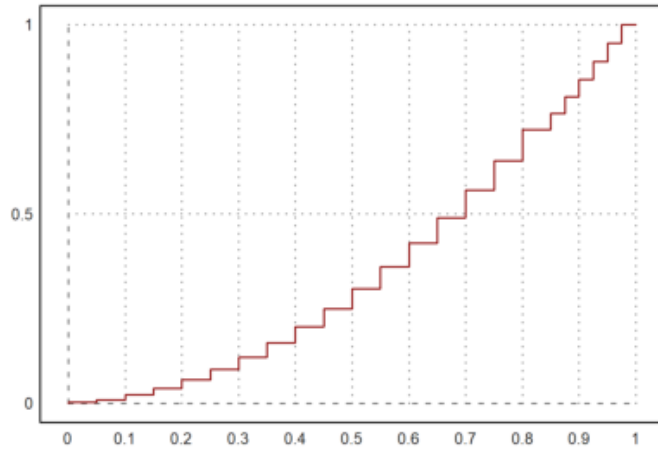
```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot dalam persegi panjang
```



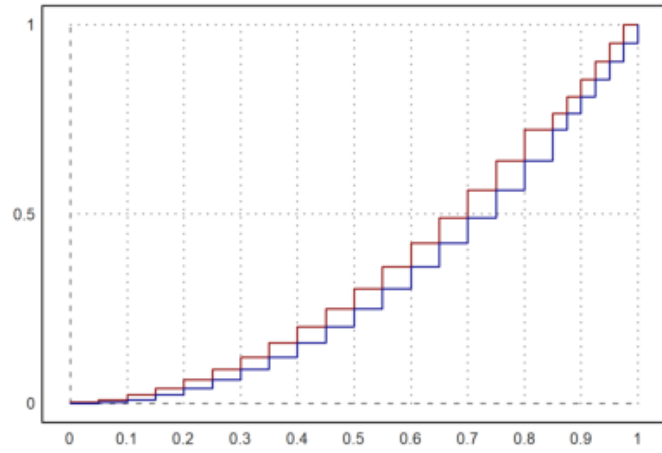
```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // tetap membuat plot pada persegi
```



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



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

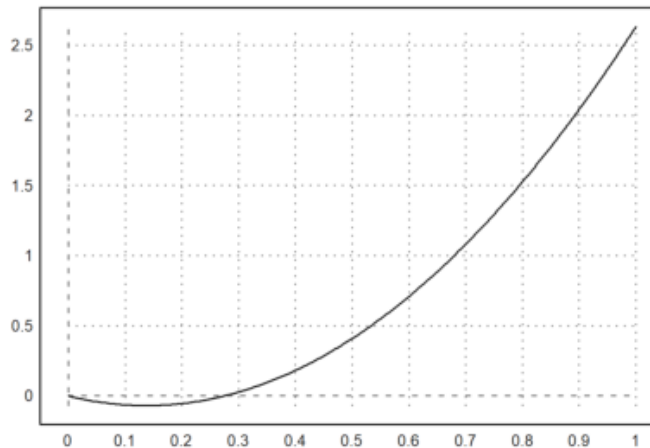


Fungsi-fungsi dalam satu Parameter

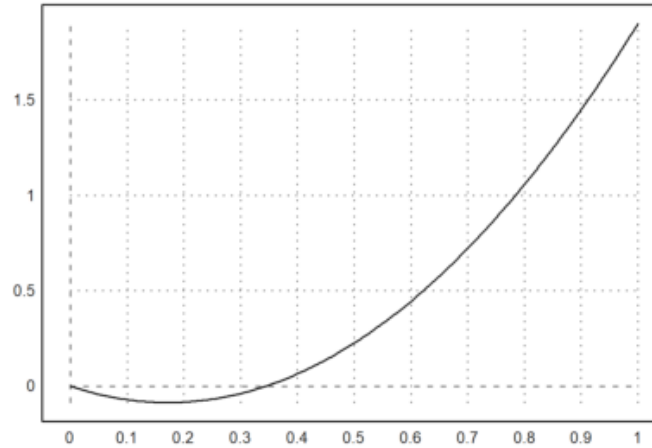
Hal yang paling penting dalam membuat fungsi plot adalah plot-plot planar adalah plot2d. Fungsi adalah pengimplementasian dalam bahasa Euler dalam berkas "plot.e", yang mana dimuat pada saat program dimulai.

Ini adalah beberapa contoh menggunakan sebuah fungsi. Seperti biasa dalam EMT, fungsi-fungsi itu bekerja untuk fungsi fungsi lainnya atau ekspresi-ekspresi, kamu dapat menambahkan parameter tambahan (disamping x) yang mana bukan variabel-variabel global untuk fungsi dengan parameter parameter titik dua atau dengan sebuah koleksi pemanggilan.

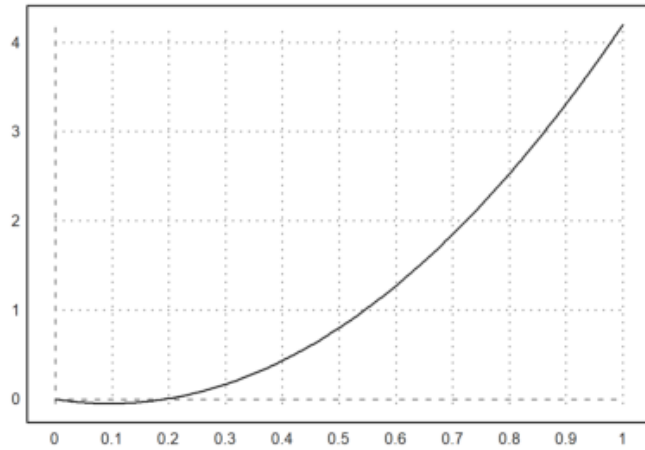
```
>function f(x, a) := x^2/a+a*x^2-x; // mendefinisikan sebuah fungsi  
>a = 0.3; plot2d("f", 0, 1; a): // plot fungsi f dengan a = 0.3
```



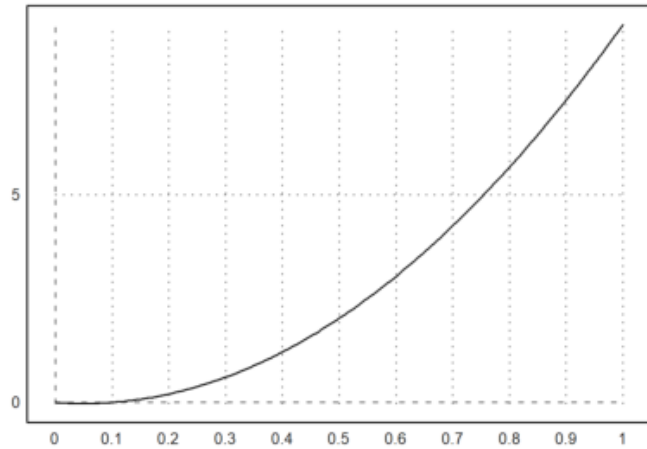
```
>plot2d("f", 0, 1; 0.4): // plot fungsi f dengan a = 0.4
```



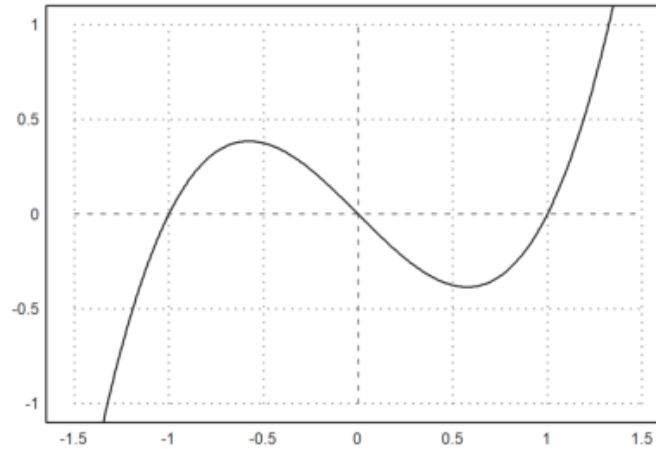
```
>plot2d({"f",0.2}), 0, 1): // plot dengan a = 0.2
```



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

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



Ini adalah rangkuman dari fungsi-fungsi yang diterima

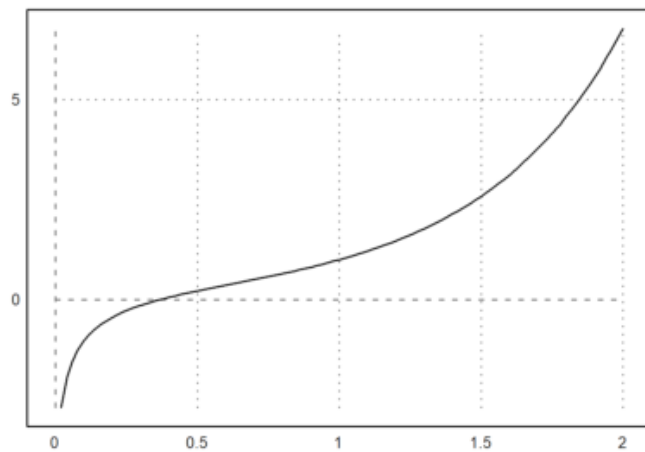
- ekspresi-ekspresi atau ekspresi-ekspresi simbolik dalam x
- fungsi-fungsi atau simbolik fungsi-fungsi dengan nama sebagai "f"
- fungsi-fungsi simbolik hanya dengan nama f

fungsi `plot2d()` juga menerima fungsi-fungsi simbolik. Untuk simbolik fungsi, nama bekerja secara sendiri.

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

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

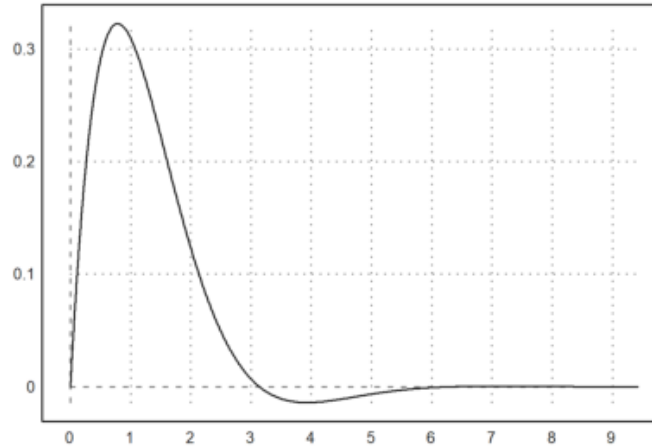
```
>plot2d(f, 0, 2):
```



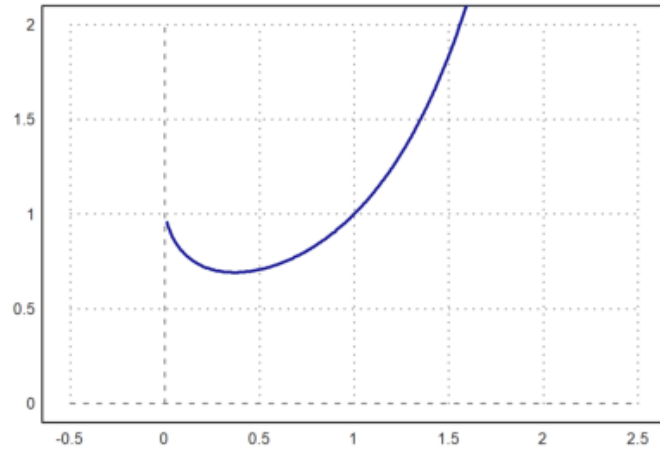
```
>expr &= sin(x) * exp(-x)
```

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

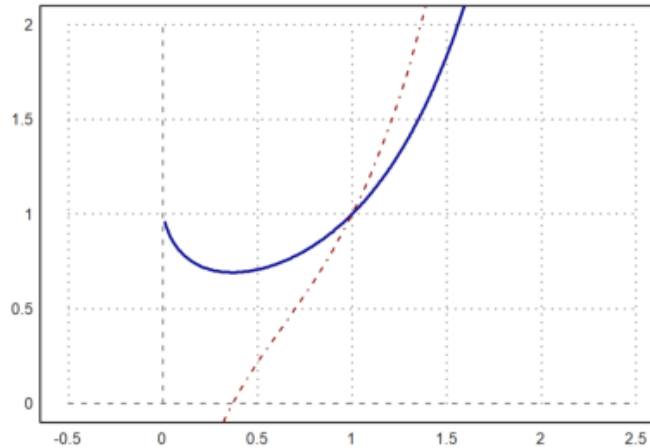
```
>plot2d(expr, 0, 3pi):
```



```
>function f(x) &= x^x;  
>plot2d(f, r = 1, cx = 1, cy = 1, color = blue, thickness = 2):
```



```
>plot2d(&diff(f(x), x), >add, color = red, style="-.-"):
```



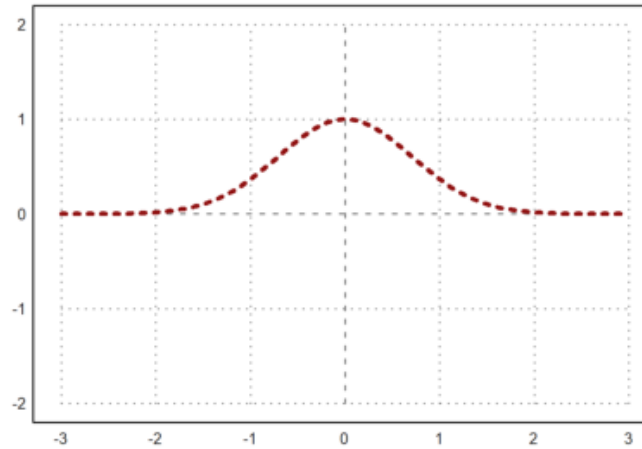
Untuk gaya garis disini memiliki banyak opsi.

- style = "...". Memilih dari "-", "--", "-.", ".", "-.", "-.-".
- color: Lihat dibawah untuk warna-warna.
- thickness: Bawaan adalah 1.

Warna warna dapat dipilih sebagai satu dari warna bawaan, atau sebagai sebuah warna RGB.

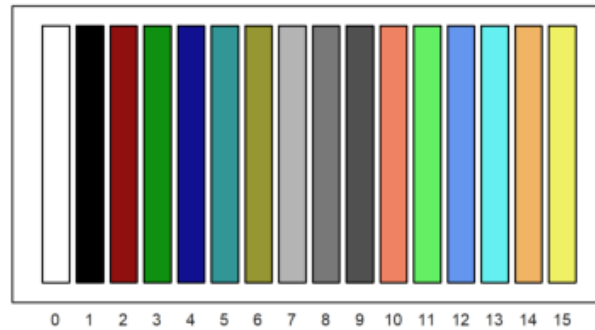
- 0..5: indeks warna bawaan.
- konstanta warna: white, black, red, green, blue, cyan, olive, lightgray, gray, darkgray, orange, lightgreen, turquoise, lightblue, lightorange, yellow.
- rgb(red, green, blue): parameter marameter adalah bilangan rill dalam [0, 1].

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



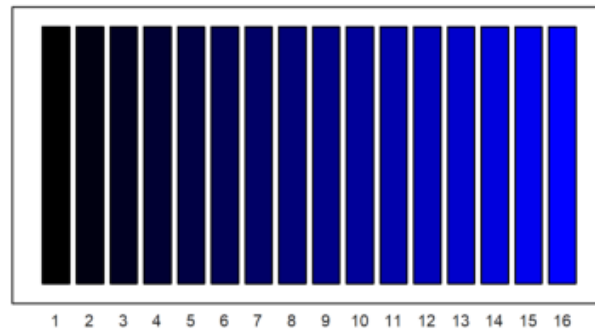
Ini adalah pandangan dari color yang telah terdefinisi dari EMT.

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



Tapi kamu dapat menggunakan warna apapun.

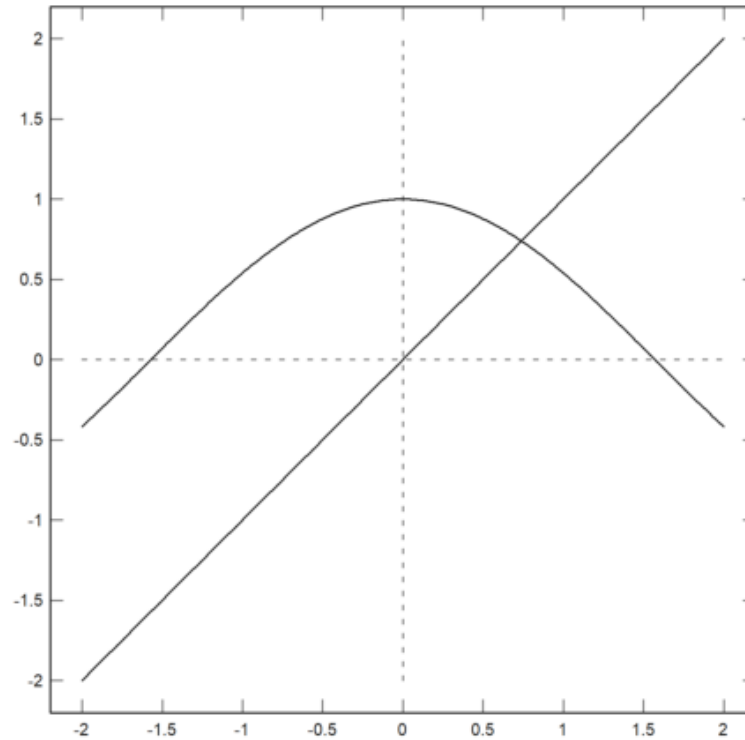
```
>columnsplot(ones(1, 16), grid = 0, color = rgb(0, 0, linspace(0, 1, 15))):
```



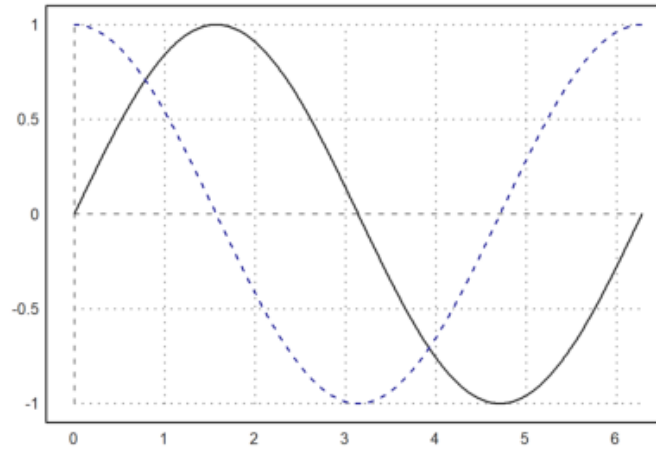
Menggambar Beberapa Kurva pada bidang koordinat yang sama

Membuat plot lebih dari satu fungsi (fungsi-fungsi yang banyak) kedalam satu jendela dapat dilakukan dengan cara-cara yang berbeda. Salah satu metode yang digunakan `>add` untuk beberapa pemanggilan ke `plot2d` dalam semua, tetapi pemanggilan pertama. Kami telah menggunakan fitur ini dalam contoh diatas.

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

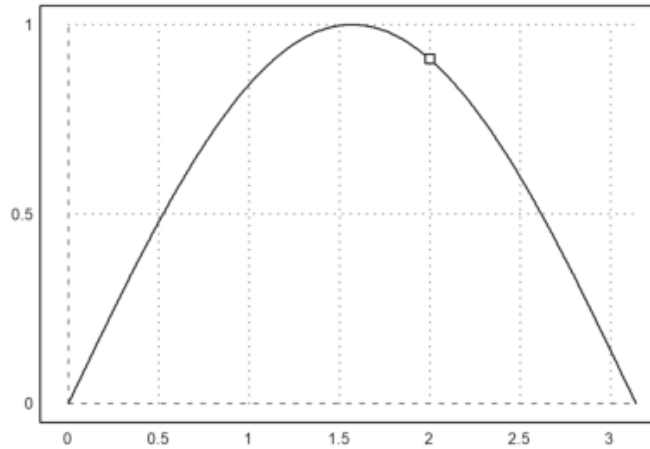


```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



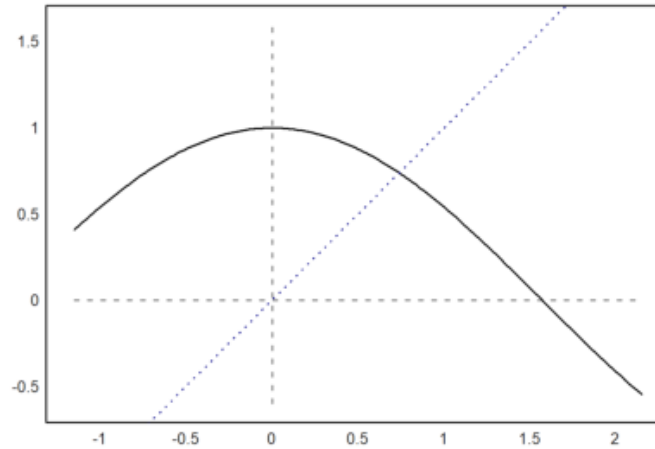
Salah satu kegunaan `>add` adalah untuk menambahkan titik pada kurva.

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

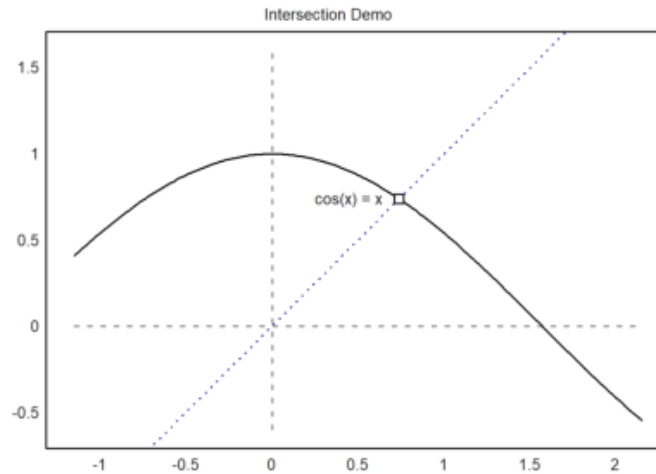


Kami menambahkan titik singgung dengan sebuah label (pada posisi "cl" untuk kiri tengah), dan memasukkan hasil kedalam notebook. Kami juga menambahkan sebuah judul kedalam plot.

```
>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):
```



Mengikuti demo, kami plot fungsi $\text{sinc}(x) = \sin(x) / x$ dan ke-8 dan ke-16 ekspansi Taylor. Kami menghitung ekspansi ini menggunakan Maxima melalui ekspresi simbolik.

Plot ini dibuat mengikuti perintah baris-banyak dengan tiga kali pemanggilan untuk `plot2d()`. Pengaturan kedua dan ketiga `>add`, yang mana membuat plot menggunakan rentang sebelumnya.

Kita menambahkan sebuah boks label untuk menjelaskan fungsi.

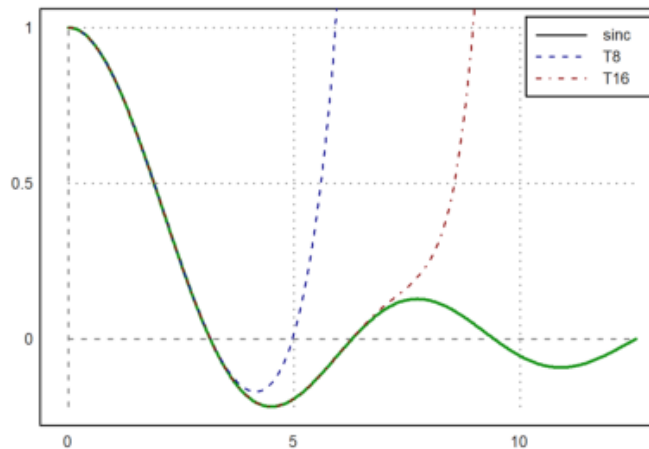
```
>$taylor(sin(x) / x, x, 0, 4)
```

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

```

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

```



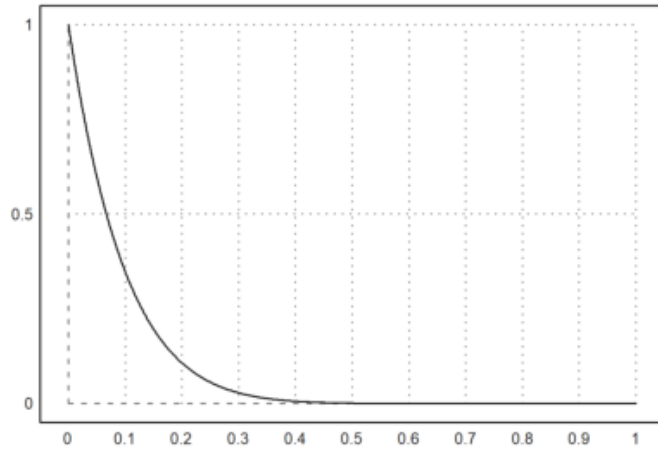
Contoh berikut, kami membuat Bernstein-Polynomials.

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

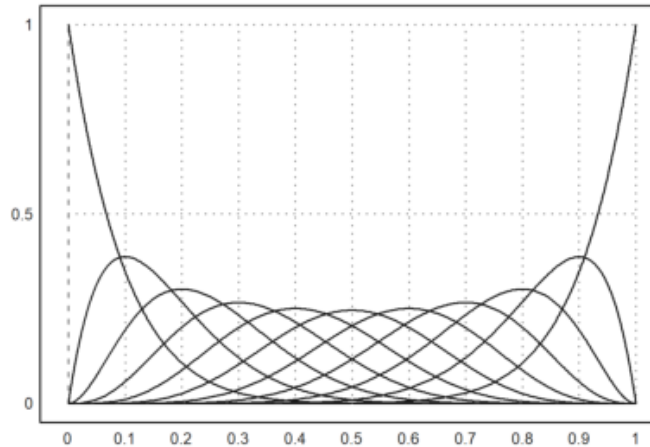
```

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

```



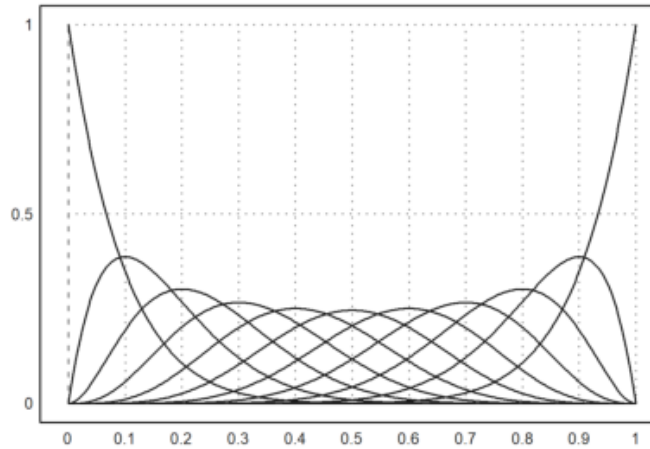
```
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;  
>insimg;
```

Metode kedua menggunakan sepasang dari sebuah matriks dari nilai-x dan sebuah matriks dari nilai-y dari ukuran yang sama.

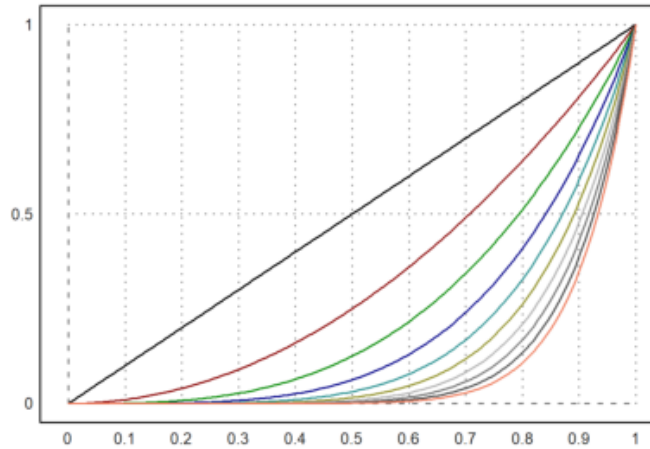
Kami menghasilkan sebuah matriks dari nilai-nilai dengan satu Bernstein-Polynomial dalam setiap bari. Untuk ini, kami mempersingkat menggunakan sebuah vektor kolom dari i. Lihat kembali ke pengantar tentang bahasa matriks untuk mempelajari lebih rinci.

```
>x = linspace(0, 1, 500);
>n = 10; k = (0:n)'; // n adalah vektor baris, k adalah kolom vektor
>y = bin(n, k)*x^k*(1-x)^(n-k); // y adalah sebuah matriks lalu
>plot2d(x, y):
```



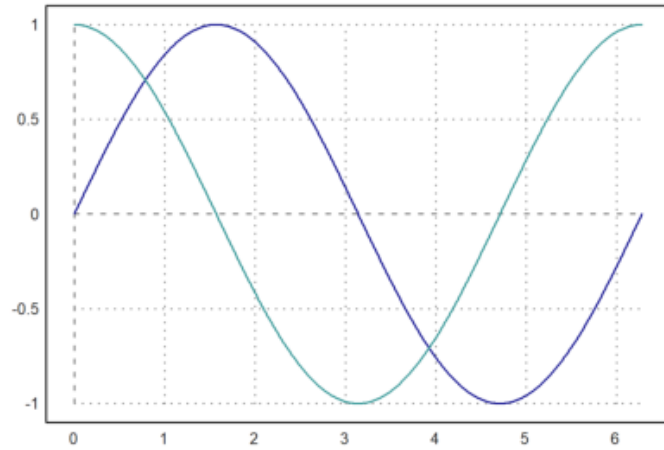
Catatan bahwa parameter warna dapat berupa sebuah vektor. Lalu setiap warna dapat digunakan untuk setiap baris dari matriks.

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

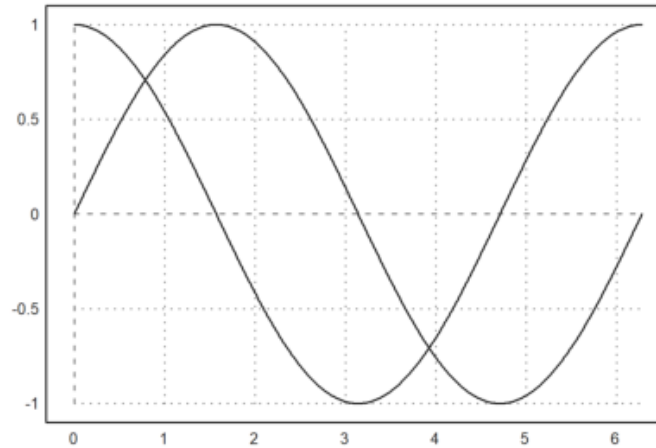


Metode lain adalah menggunakan vektor ekspresi (string). Kamu dapat menggunakan sebuah array warna, sebuah array dari gaya, dan sebuah array dari ketebalan dari panjang yang sama.

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



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



Kami dapat mendapatkan sebuah vektor dari Maxima menggunakan makelist() dan mxm2str().

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

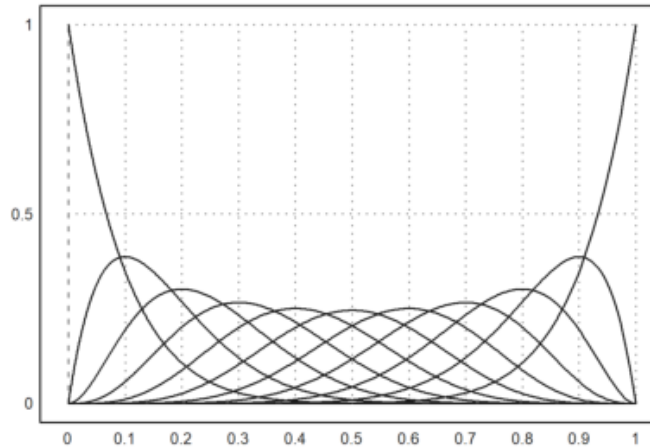
```

      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) // mendapatkan sebuah vektor dari string dari vektor simbolik
```

```
(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): // fungsi-fungsi plot
```

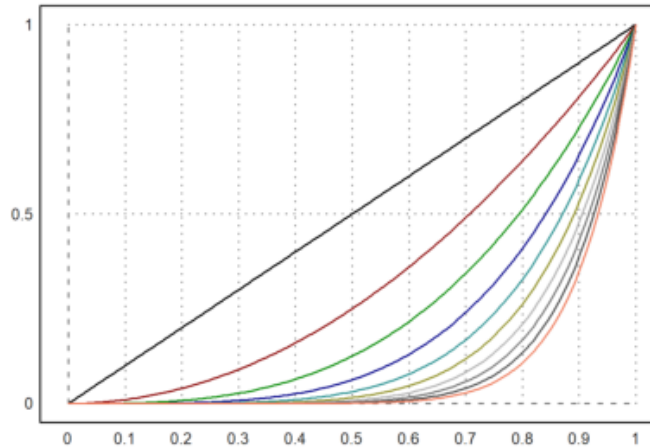


Alternatif lainnya adalah menggunakan bahasa matriks dari Euler.

Jika sebuah ekspresi menghasilkan sebuah matriks dari fungsi, dengan satu fungsi dalam setiap baris, semua fungsi-fungsi ini akan di plotkan menjadi satu plot.

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

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

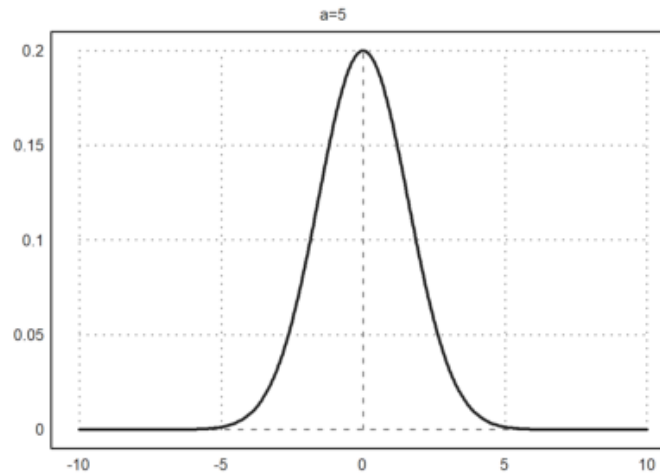


Ekspresi-ekspresi dan fungsi-fungsi satu-baris dapat dilihat variabel global.

Jika kamu tidak menggunakan variabel global, kamu membutuhkan untuk menggunakan sebuah fungsi dengan parameter ekstra, dan memasukkan parameter ini seperti sebuah parameter titik dua.

Berhati-hati, untuk mengambil semua parameter yang terpasang ke akhir dari perintah plot2d. Dalam contoh kami memasukkan $a = 5$ ke fungsi f , yang mana kami plot dari -10 ke 10.

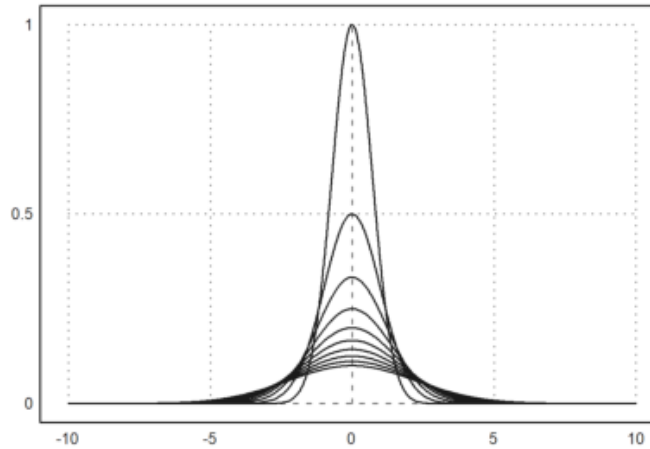
```
>function f(x,a) := 1/a *exp(-x^2/a); ...
>plot2d("f",-10,10;5,thickness=2,title="a=5"):
```

Sebagai gantinya, gunakan sebuah collection dengan nama fungsi dan semua parameter-parameter tambahan. Daftar spesial ini disebut collection dan ini adalah cara yang disarankan untuk memasukkan argumen kedalam sebuah fungsi yang mana itu sendiri dimasukkan sebagai sebuah argumen kedalam fungsi lainnya.

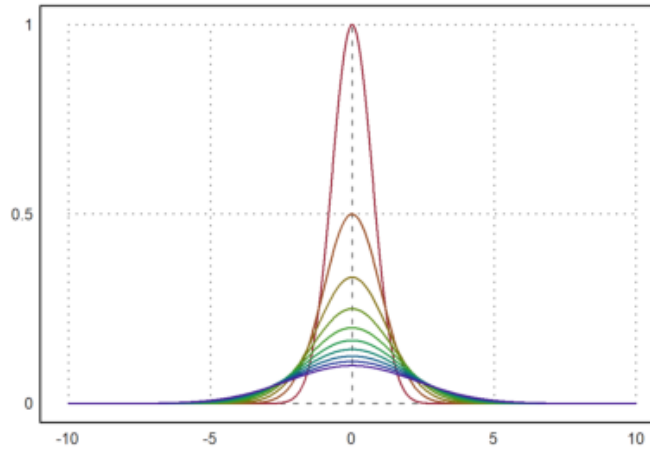
Contoh berikut, kami menggunakan sebuah iterasi untuk membuat plot beberapa fungsi (lihat tutorial tentang pemrograman untuk iterasi).

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



Kami dapat menghasilkan hasil yang sama dengan menggunakan cara seperti menggunakan bahasa matriks EMT. Setiap baris dari matriks $f(x, a)$ adalah satu fungsi. Terlebih, kita dapat mengatur warna untuk setiap baris dari matriks. Untuk penjelasan klik dua kali ke fungsi `getspectral()`.

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



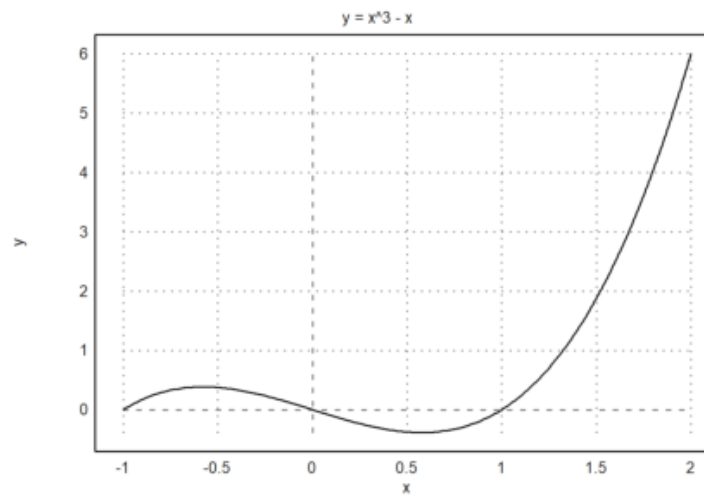
Label Teks

Dekorasi sederhana dapat berupa

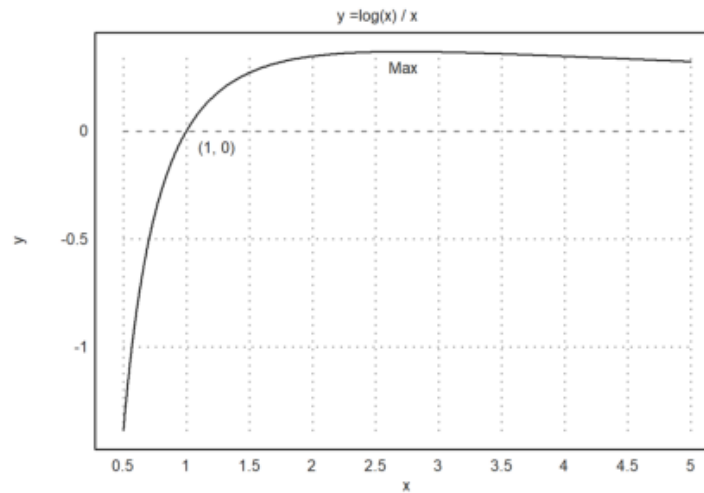
- sebuah judul dengan `title="..."`
- label-x dan label-y dengan `xl="..."`, `yl="..."`
- teks label lainnya dengan `label("...", x, y)`

perintah `label` akan memplot ke plot sekarang sebagai koordinat plot (x, y). Ini dapat mengambil argumen posisi.

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

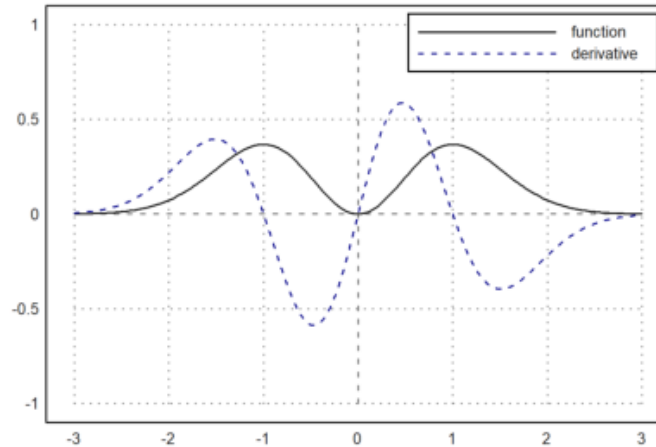


```
>expr := "log(x) / x";...  
>plot2d(expr, 0.5, 5, title = "y =" +expr, xl = "x", yl = "y");...  
>label("(1, 0)", 1, 0); label("Max", E, expr(E), pos="1c");
```



Ini juga fungsi `labelbox()`, yang mana dapat memperlihatkan fungsi-fungsi dan sebuah teks. Dengan mengambil vektor dari string dan warna, satu dari 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):
```

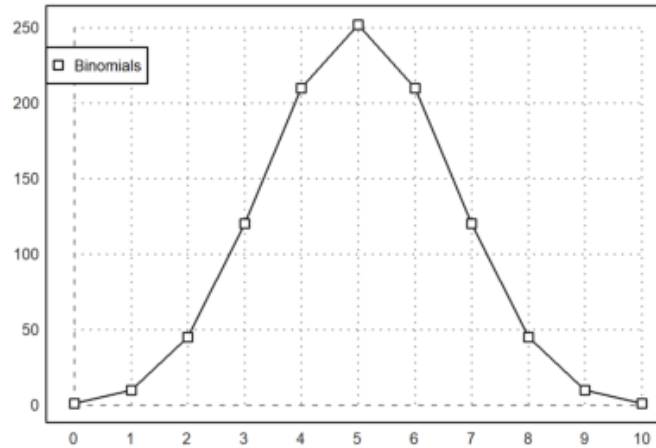


Box tersebut diposisikan pada kanan atas secara bawaan, tetapi posisi `>left` membuatnya pada kiri atas. Kamu dapat mengubahnya sebarang tempat yang kamu sukai. Posisi terdapat pada pojok kanan atas dari box, dan angka-angkanya adalah pecahan dari ukuran jendela. Lebar secara otomatis diatur.

Untuk titik-titik plot, label box juga demikian. Tambahkan parameter `>points`, atau sebuah vektor dari pernyataan, satu untuk setiap label.

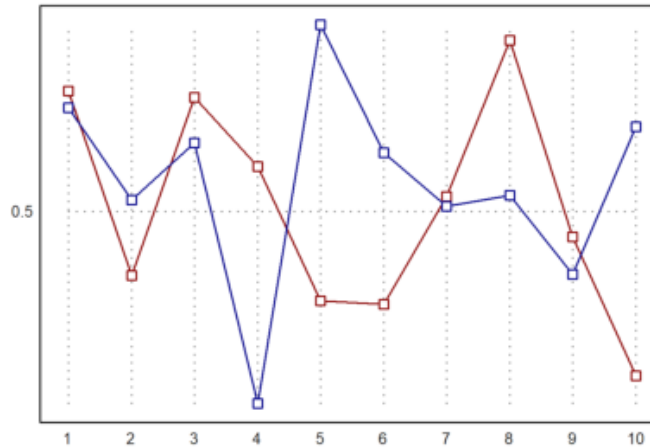
Seperti contoh yang ditunjukkan, ini hanya satu fungsi. Jadi kami dapat menggunakan strings daripada vektor string. Kami mengatur warna teks ke hitam untuk contoh ini.

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



Gaya dari plot ini juga tersedia dalam `statplot()`. Seperti dalam `plot2d()` warna-warna dapat diatur untuk setiap baris dari plot. Ini adalah plot-plot spesial untuk tujuan statistika (lihat tutorial tentang statistika).

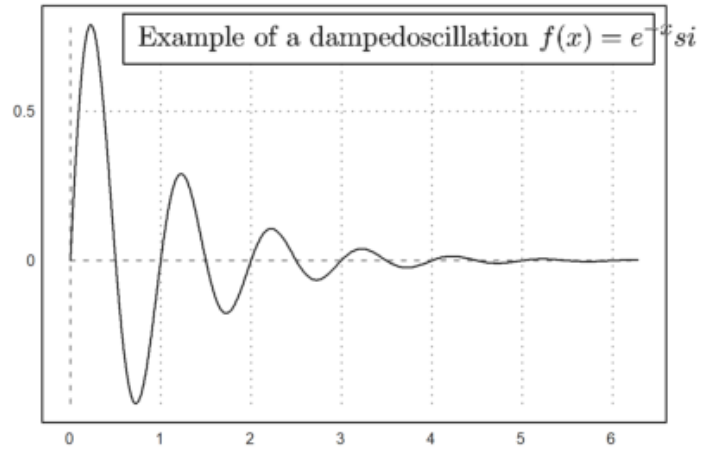
```
>statplot(1: 10, random(2, 10), color = [red, blue]):
```



Fitur yang sama dari fungsi `textbox()`.

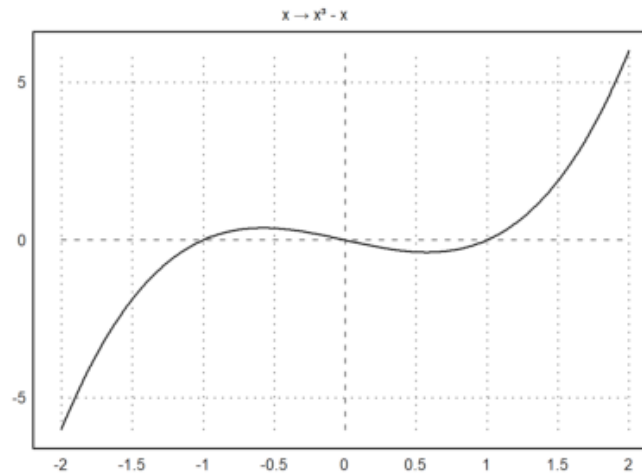
Secara bawaan lebar diatur setara dengan lebar maksimal dari lebar baris teks. Tapi ini dapat diatur oleh pengguna juga.

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

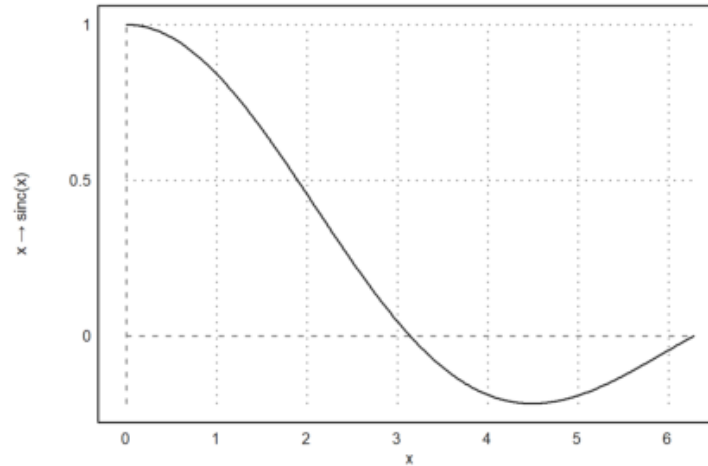
Label-label teks, judul, label box dan teks lainnya dapat berupa string Unicode (lihat sintak dari EMT untuk lebih lanjut tentang string Unicode).

```
>plot2d("x^3 - x", title = u"x &rarr; x3 - x"):
```



Label-label dalam sumbu x dan y dapat secara vertikal, mengikuti sumbunya.

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



LaTeX

Kamu juga dapat mem-plot formula LaTeX jika kamu telah menginstall sistem LaTeX. Saya merekomendasikan MiKTeX. Jalur ke binari "latex" dan "dvi2png" harus ada dalam jalur sistem, taua kamu telah mengatur LaTeX dalam menu options.

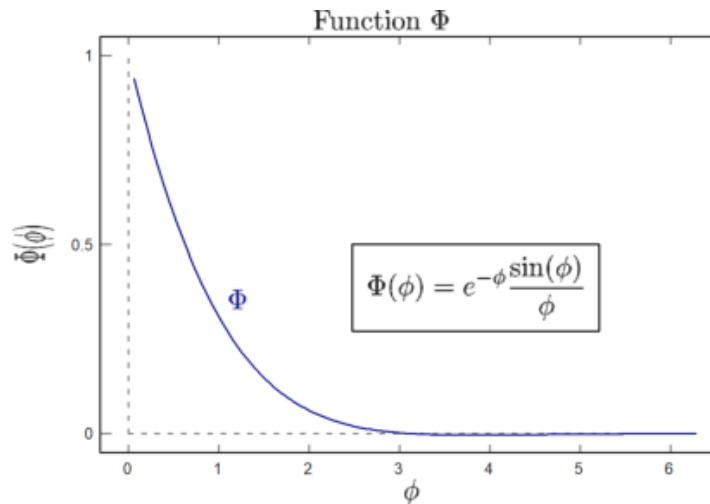
Catatan, bahwa pengubah LaTeX itu lambat. Jika kamu ingin LaTeX dalam plot yang memiliki animai, kamu harus memanggil `latex()` sebelum iterasi pertama dan gunakan hasilnya (sebuah gambar dalam sebuah matriks RGB).

Seperti plot berikut, kita menggunakan LaTeX untuk label x dan label y, sebuah label, sebuah label box 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 $\Phi$}"), ...
>x1 = latex("\phi"), y1 = 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):

```



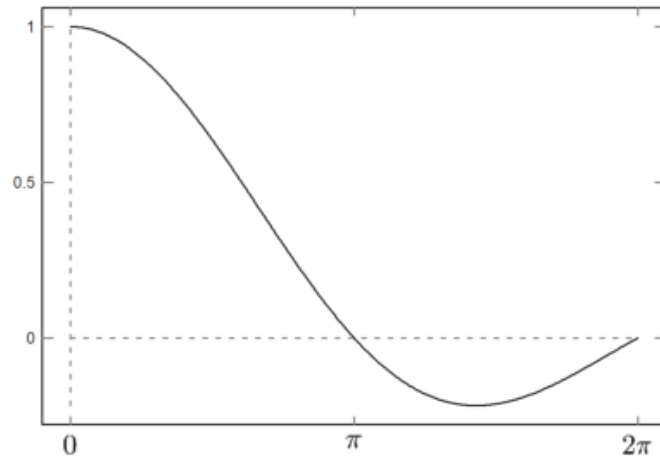
Terkadang, kita ingin spasi yang non-conformal dan label teks dalam sumbu-x. Kita dapat menggunakan `xaxis()` dan `yaxis()` seperti yang akan ditunjukkan kemudian.

Cara termudah untuk melakukan sebuah plot yang tidak ada dengan sebuah frame menggunakan `grid = 4` dan lalu tambahkan grid dengan `ygrid()` dan `xgrid()`. Seperti contoh berikut, kita gunakan tiga string LaTeX untuk label dalam 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-fungsi dapat juga digunakan.

```

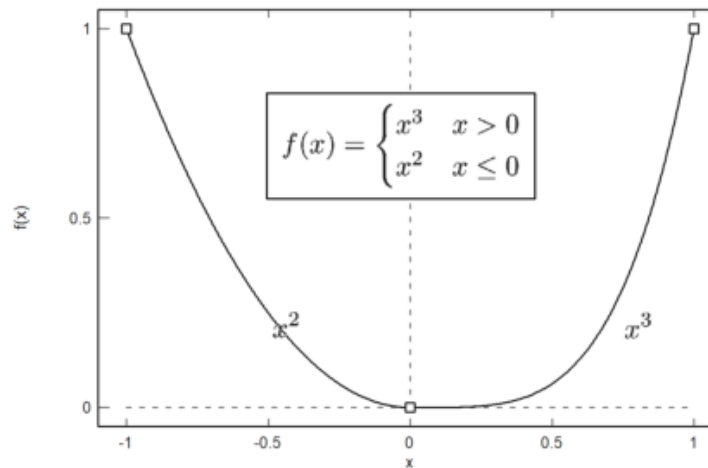
>function map f(x) ...

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

```

Parameter "map" membantu untuk menggunakan fungsi dari vektor. Untuk membuat plot, ini tidak akan digunakan. Tetapi untuk demonstrasi bahwa vektorisasi sangat berguna, kita tambahkan beberapa titik kedalam plot pada $x = -1$, $x = 0$ dan $x = 1$. Seperti plot berikutm kita juga memasukkan beberapa code LaTeX. Kami gunakan ini untuk dua label dan sebuah teks box. Tentu saja, kamu hanya ingin menggunakan LaTeX jika kamu telah menginstal LaTeX secara benar.

```
>plot2d("f", -1, 1, x1 = "x", y1 = "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 = "11");...
>textbox(...
>latex("f(x) = \begin{cases} x^3 & x > 0 \\ x^2 & x \le 0 \end{cases}"), ...
>x = 0.7, y = 0.2):
```



Ketika membuat plot sebuah fungsi atau sebuah ekspresi, parameter `>user` memperbolehkan pengguna untuk memperbesar dan memindahkan plot dengan kursor atau mouse.

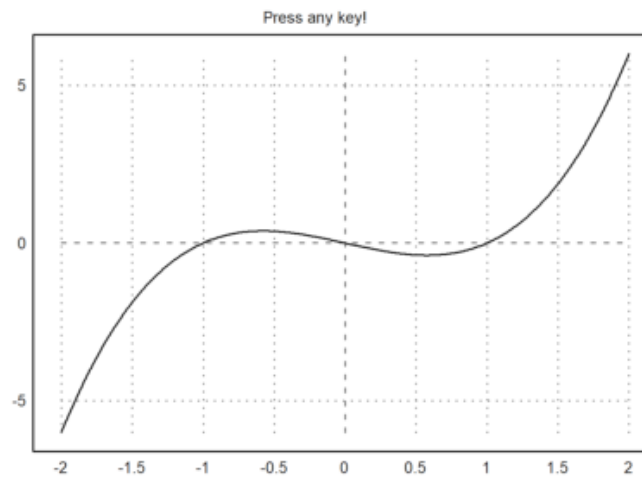
Pengguna dapat

- memperbesar dengan `+` or `-`
- memindahkan plot dengan kursor
- memilih jendela sebuah plot dengan mouse
- mengatur ulang pandangan dengan spasi
- keluar dengan `return`

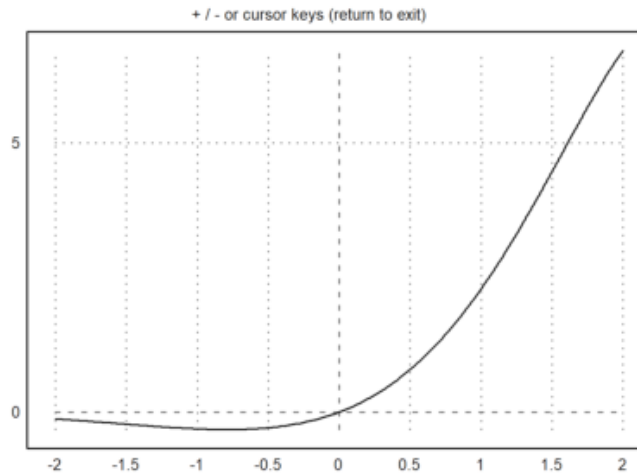
Spasi akan mengatur ulang plot menjadi jendela plot asli.

Ketika membuat plot sebuah data, `>user` parameter akan secara sederhana menunggu tombol diketikkan.

```
>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)":
```

Demonstrasi berikut sebuah cara lanjutan dari interaksi pengguna (lihat tutorial tentang pemrograman untuk detailnya).

Fungsi yang sudah ada `mousedrag()` menunggu mouse atau pergerakan keyboard. ini akan melaporkan mouse yang bergerak dan keyboard tertekan. fungsi `dragpoints()` menggunakan ini, dan memungkinkan pengguna menyeret sebarang titik dalam sebuah plot.

Kita butuh sebuah fungsi plot terlebih dahulu. Sebagai contoh, kita menginterpolasi dalam 5 titik oleh sebuah polinomial. Fungsi akan memplot sampai sebuah luasan 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
```

```

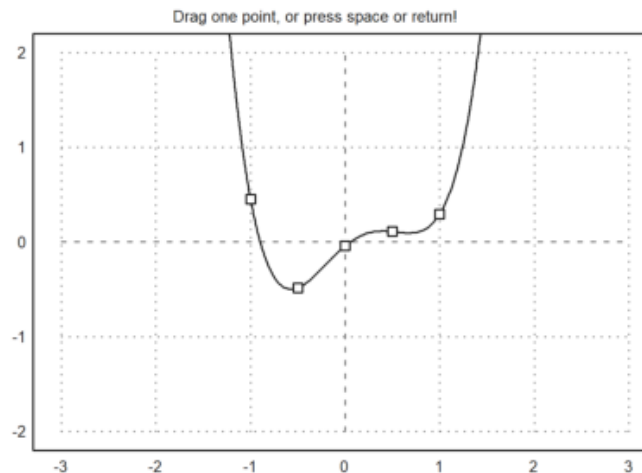
    plot2d(xp[select], yp[select], color = red, >points, >add);
endif;
title("Drag one point, or press space or return!");
endfunction

```

Catatan paramter semicolon dalam plot2d(d dan xp), yang mana dimasukkan ke evaluasi dari fungsi interp(). Tanpa ini, kita harus menuliskan sebuah fungsi plotinterp() terlebih dahulu, mengakses nilai secara global.

Sekarang kita menghasilkan beberapa nilai acak dan biarkan pengguna menggeser titik-titiknya.

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



Ini juga sebuah fungsi, yang mana membuat plot fungsi lainnya tergantung dalam sebuah vektor dari parameter, dan membiarkan pengguna menyesuaikan parameter-parameternya.

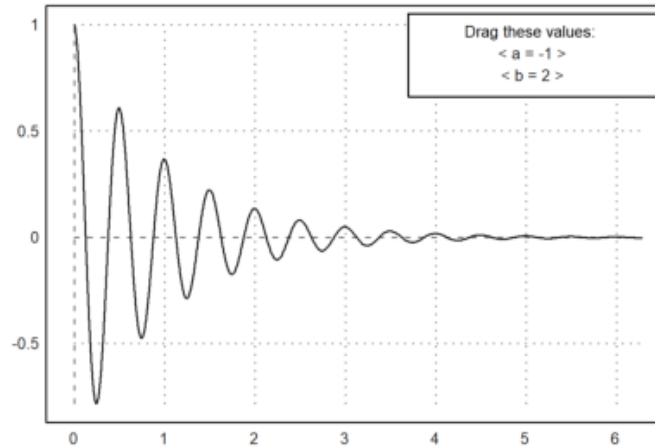
Pertama-tama kita butuh fungsi plot.

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

Lalu kita butuhkan nama untuk parameter-parameter, nilai awal dan sebuah matriks nx2 dari selang, secara pilihan sebuah baris judul.

Ini adalah slider yang interaktif, dimana dapat merubah nilai oleh pengguna. Fungsi dragvalues() tersedia untuk hal ini.

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



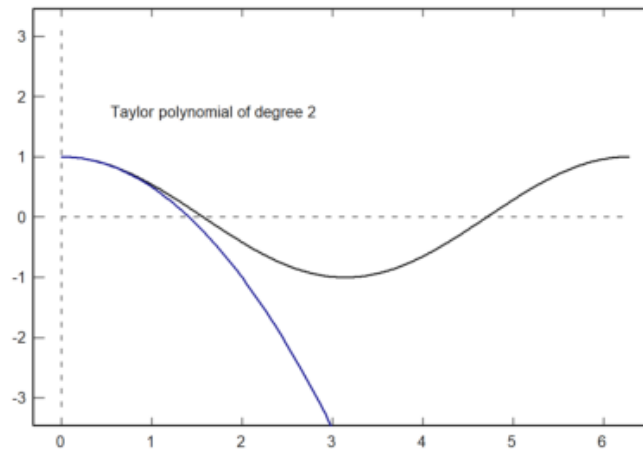
Ini memungkinkan untuk membatasi nilai yang dapat digerakkan hanya bilangan bulat. Sebagai contoh, kita menulis sebuah fungsi plot, yang mana membuat plot sebuah polinomial Taylor berderajat n ke fungsi cosinus.

```
>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, 0.2, style = "t", >left);
endfunction
```

Sekarang kita bisa membuat derajat n menjadi bervariasi dari 0 ke 20 dengan 20 pemberhentian. Hasil dari `dragvalues()` digunakan untuk membuat plot sketsa dengan n , dan memasukkan plot kedalam notebook.

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



Demonstrasi sederhana ini dari sebuah fungsi. Pengguna dapat menggambar diatas jendela plot, meninggalkan titik-titik lintasan.

```
>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
```

Gaya-Gaya Plot 2D

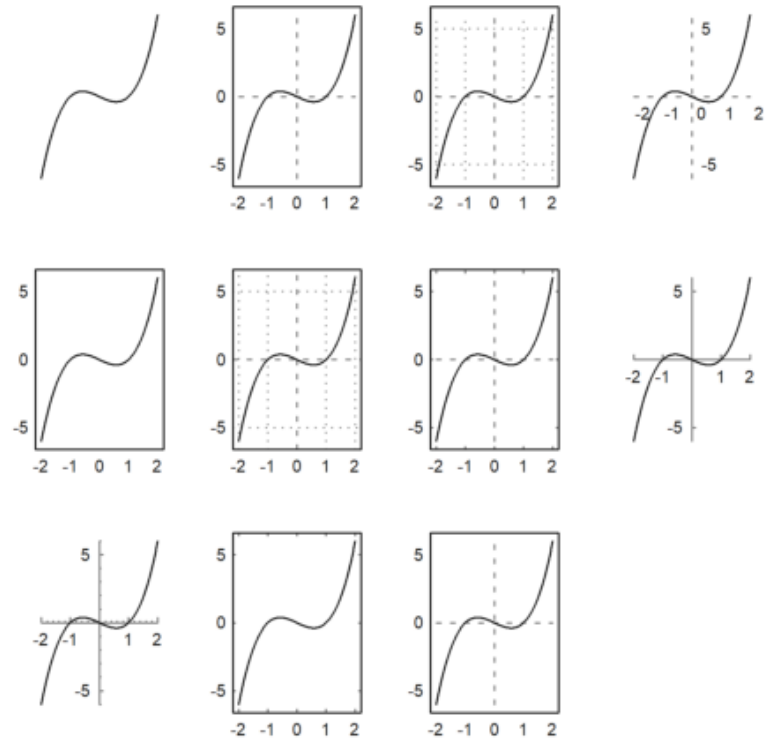
Secara bawaan, EMT secara otomatis menghitung titik sumbu dan menambahkan label ke titiknya. Ini dapat diubah dengan parameter grid. Gaya bawaan dari sumbu dan labelnya dapat diubah. Sebagai tambahan, label dan judul dapat ditambahkan secara manual. Untuk mengatur ulang gaya bawaan, gunakan `reset()`.

```

>aspect();
>figure(3, 4);...
>figure(1); plot2d("x^3-x", grid=0);... // tidak ada garis bantu, rangka atau sumbu
>figure(2); plot2d("x^3-x", grid=1);... // sumbu-x-y
>figure(3); plot2d("x^3-x", grid=2);... // titik bantu bawaan
>figure(4); plot2d("x^3-x", grid=3);... // sumbu x-y dengan label didalam

```

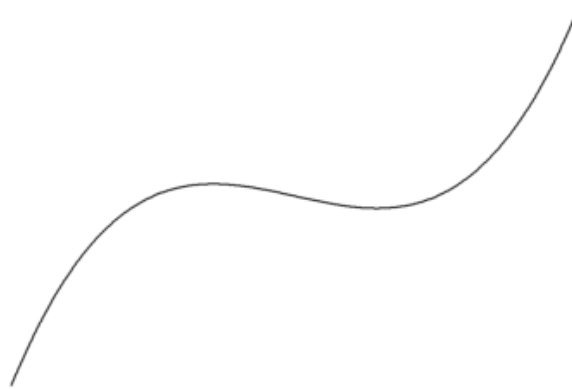
```
>figure(5); plot2d("x^3-x", grid=4);... // tidak ada titik bantu, hanya label  
>figure(6); plot2d("x^3-x", grid=5);... // bawaan, tetapi tidak ada margin  
>figure(7); plot2d("x^3-x", grid=6);... // hanya sumbu  
>figure(8); plot2d("x^3-x", grid=7);... // hanya sumbu, titik bantu pada sumbu  
>figure(9); plot2d("x^3-x", grid=8);... // hanya sumbu, titik bantu lebih mulus pada sumbu  
>figure(10); plot2d("x^3-x", grid=9);... // bawaaan, titik bantu kecil didalam  
>figure(11); plot2d("x^3-x", grid=10);... // tidak ada titik bantu, hanya sumbu  
>figure(0):
```



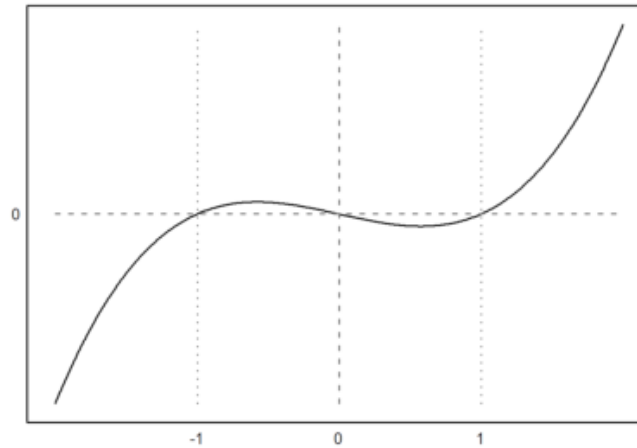
Parameter `<frame` membuat frame tidak ada, dan `facecolor = blue` mengubah frame menjadi berwarna biru.

Jika kamu ingin titik-titik kamu, kamu dapat menggunakan `style = 0`, dan tambahkan kesemuanya nanti.


```
>aspect(1.5);  
>plot2d("x^3 - x", grid = 0): // plot
```

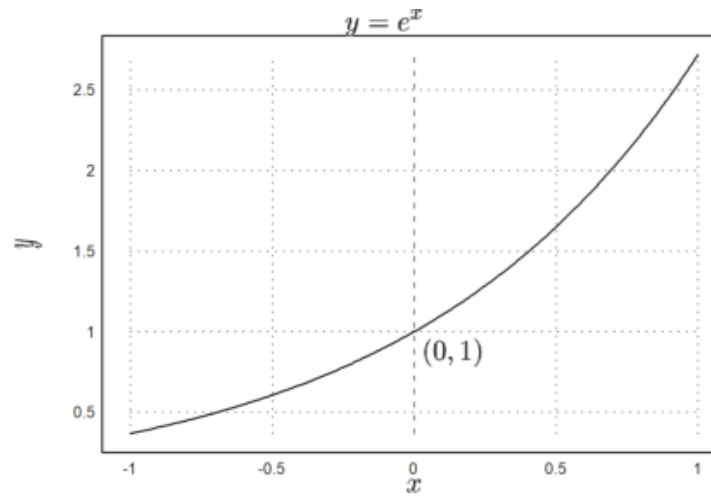


```
>frame; xgrid([-1, 0, 1]); ygrid(0): // menambahkan kerangka dan garis bantu
```



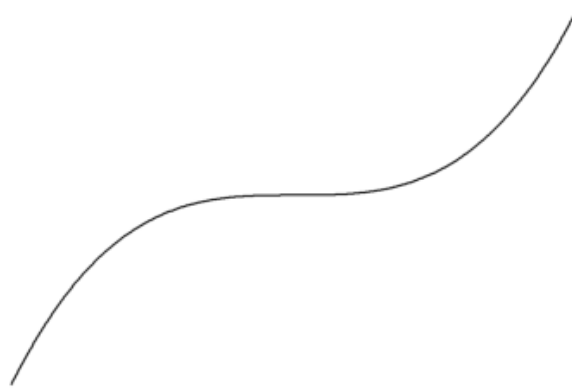
Untuk judul plot dan sumbu label, perhatikan contoh berikut

```
>plot2d("exp(x)", -1, 1);
>textcolor(black); // atur warna teks menjadi hitam
>title(latex("y = e^x")); // judul diatas plot
>xlabel(latex("x")); // "x" untuk sumbu-x
>ylabel(latex("y"), >vertical); // "y" secara vertikal untuk sumbu-y
>label(latex("(0, 1)"), 0, 1, color = blue); // label sebuah titik
```

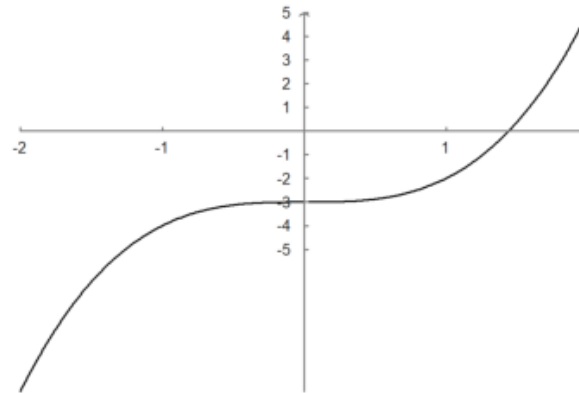


Sumbu-sumbu dapat digambarkan secara terpisah dengan `xaxis()` dan `yaxis()`.

```
>plot2d("x^3 - 3", <grid, <frame):
```



```
>xaxis(0, xx = -2:1, style = "->"); yaxis(0, yy = -5:5, style = "->"):
```

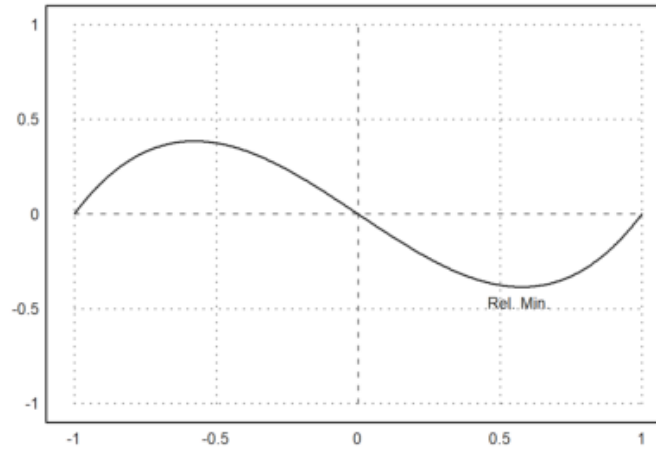


Teks dalam plot dapat diatur dengan `label()`. Seperti contoh, "lc" berarti tengah bawah. Ini mengatur posisi dari label secara relatif ke dalam koordinat plot.

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

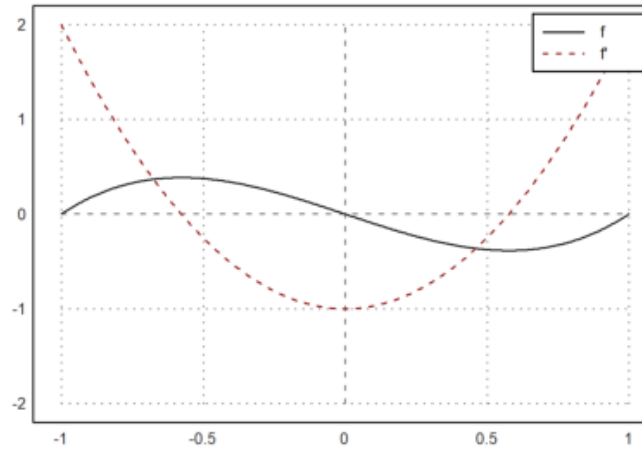
$$x^3 - x$$

```
>plot2d(f, -1, 1, >square);
>x0 = fmin(f, 0, 1); // menghitung titik minimum
>label("Rel. Min. ", x0, f(x0), pos = "lc"): // menambahkan sebuah label disini
```

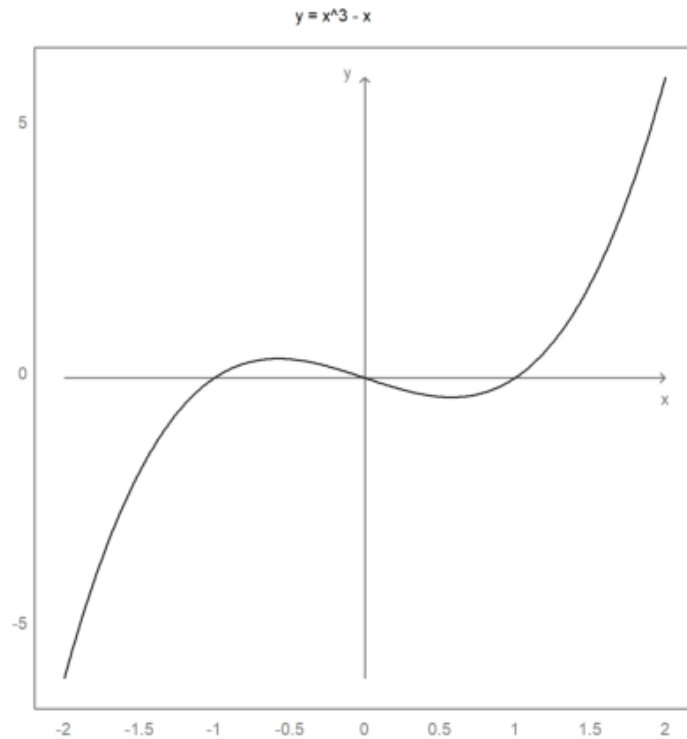


Ini juga teks box.

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



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

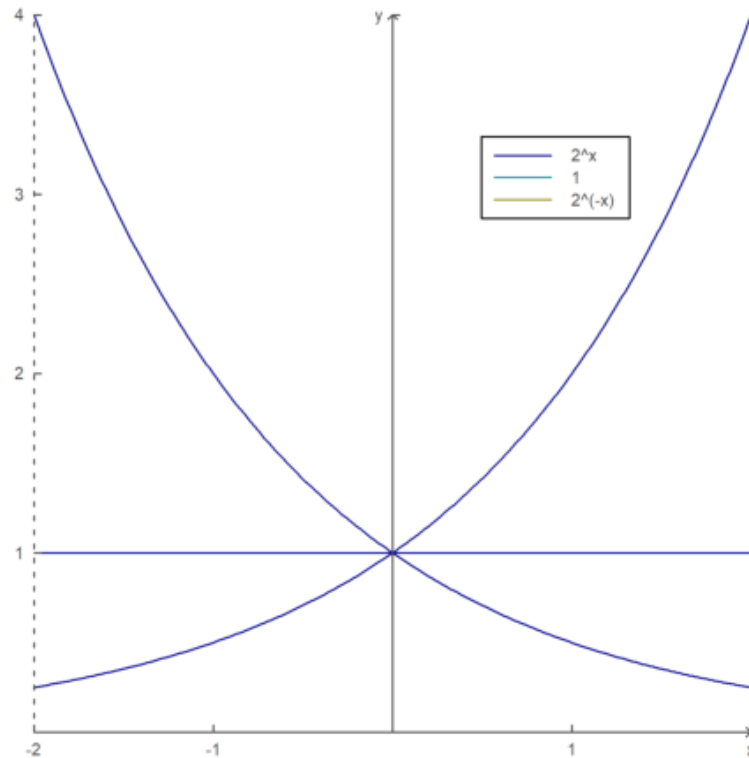


Untuk kontrol lebih, sumbu-x dan sumbu-y dapat dilakukan secara manual. Perintah `fullwindow()` menyebarkan jendela plot karena kita tidak membutuhkan tempat untuk label luar pada judul plot. Gunakan `shrinkwindow()` atau `reset()` untuk mengatur ulang ke pengaturan bawaan.


```

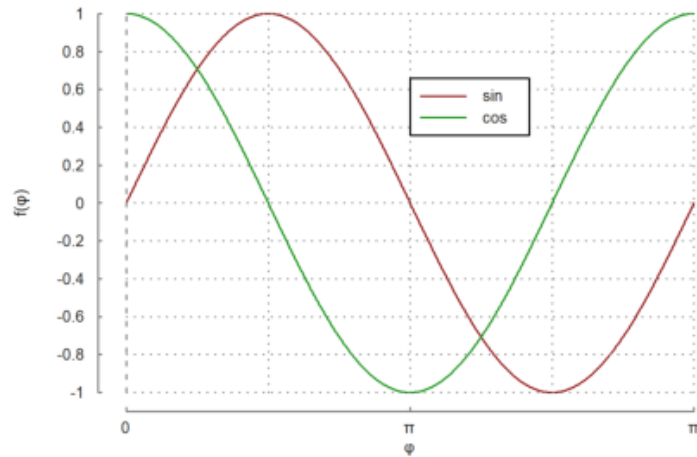
>fullwindow; ...
>gridstyle(color = darkgray, textcolor = darkgray); ...
>plot2d(["2^x", "1", "2^(-x)"], a=-2, b=2, c=0, d=4, <grid, color=4.6, <frame);...
>xaxis(0, -2:1, style="->"); xaxis(0, 2, "x", <axis); ...
>yaxis(0, 4, "y", style="->"); yaxis(-2, 1:4, >left); ...
>labelbox(["2^x", "1", "2^(-x)"], colors=4:6, x=0.8, y=0.2);...
>reset:

```



Ini adalah contoh lainnya, dimana string Unicode digunakan dan sumbu diluar luasan 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"&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;)"):
```



Memploting data 2D

Jika x dan y adalah vektor-vektor data, data ini akan digunakan sebagai koordinat x dan y dari kurva. Dalam kasus ini, a , b , c dan d atau sebuah radius r dapat dispesifikkan atau jendela plot akan menyesuaikan data secara otomatis.

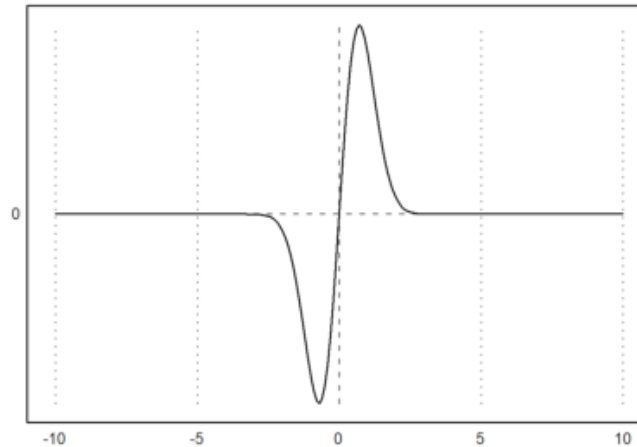
Membuat plot ekspresi hanya sebuah singkatan untuk plot data. Untuk plot data, kamu membutuhkan satu atau lebih barisan dari nilai- x dan satu atau lebih barisan dari nilai- y . Dari rentang dan nilai- x fungsi `plot2d` akan menghitung data untuk plot, secara bawaan dengan evaluasi adaptasi dari fungsi. Untuk plot-plot titik gunakan `>points`, untuk campuran garis dan titik gunakan `>addpoints`.

Tetapi kamu dapat memasukkan data secara langsung:

- Gunakan vektor baris untuk x dan y untuk satu fungsi.
- Matriks-matriks untuk x dan y diplotkan garis dengan garis.

Ini adalah contoh untuk satu baris untuk x dan y .

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



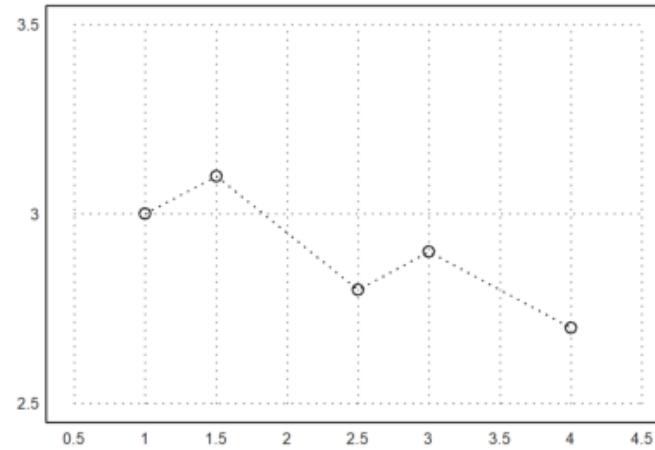
Data juga dapat diplotkan sebagai titik-titik. Gunakan `points=true` untuk melakukan ini. Plot akan bekerja seperti poligon, tetapi menggambar hanya sudut-sudutnya.

- `style = "...`: pilih dari "`[]`", "`<>`", "`o`", "`.`", "`..`", "`+`", "`[]`", "`<>`", "`o`", "`..`", "`''`", "`|`".

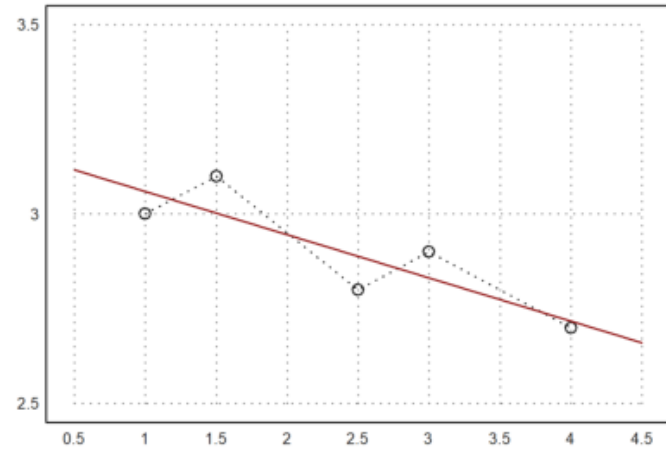
Untuk barisan plot dari titik-titik gunakan `>points`. Jika warna adalah sebuah vektor warna, setiap titik akan memiliki warna yang berbeda. Untuk sebuah matriks koordinat dan vektor kolom, warna diaplikasikan ke barisan dari matriks.

Parameter `>addpoint` menambahkan titik ke segmen bari 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="."); // garis
>plot2d(xdata, ydata, >points, >add, style="o"): // menambahkan titik
```



```
>p = polyfit(xdata, ydata, 1); // mendapatkan garis regresi  
>plot2d("polyval(p, x)", >add, color=red): // menambah plot garis
```



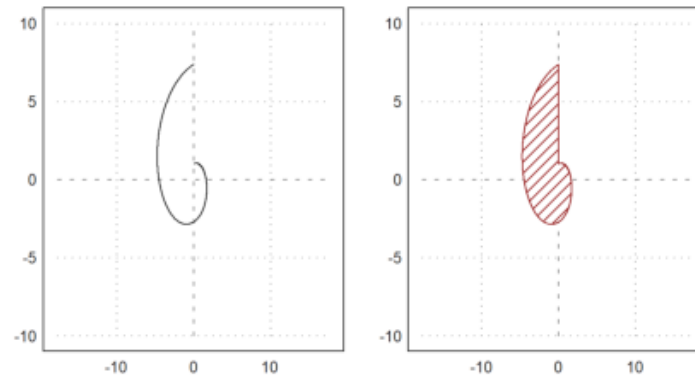
Menggambar Daerah yang dibatasi Kurva

Plot-plot data memang benar-benar poligon. Kita dapat juga membuat plot kurva atau kurva yang berisi.

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

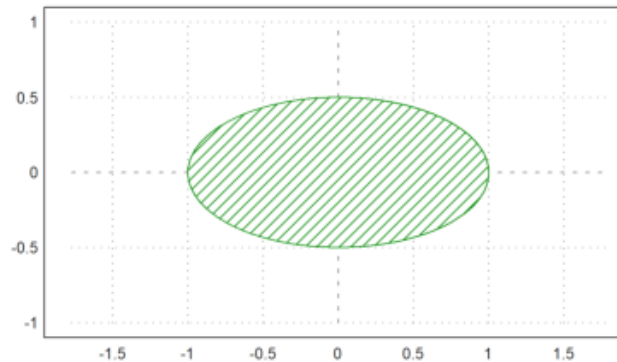
Warna isian dapat ditentukan dengan argumen "fillcolor" dan secara pilihan <outline mencegah membuat batasan dari setiap gaya tetapi salah satunya bawaan.

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

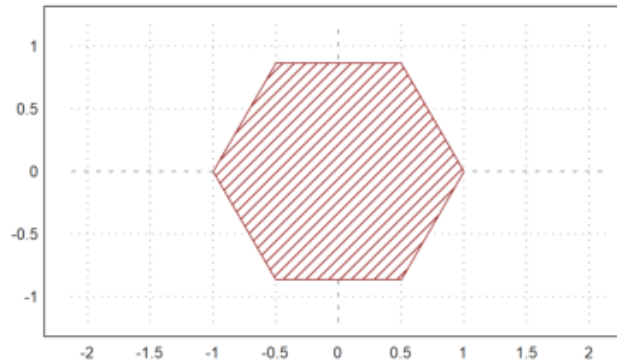


Contoh berikut kami membuat plot sebuah ellips yang terisi dan dua segidelapan yang terisi menggunakan kurva tertutup dengan 6 titik-titik dengan gaya isian yang berbeda.

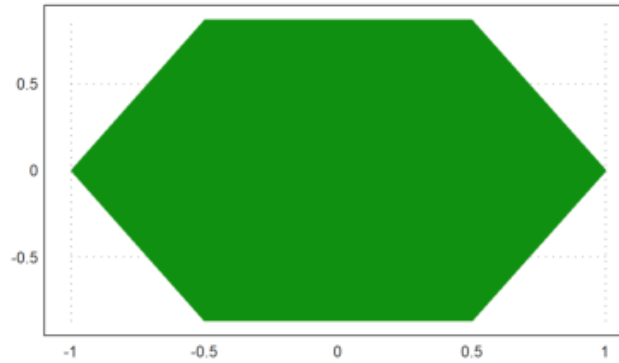
```
>x = linspace(0, 2pi, 1000); plot2d(sin(x), cos(x) * 0.5, r=1, >filled, style="/"):
```




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

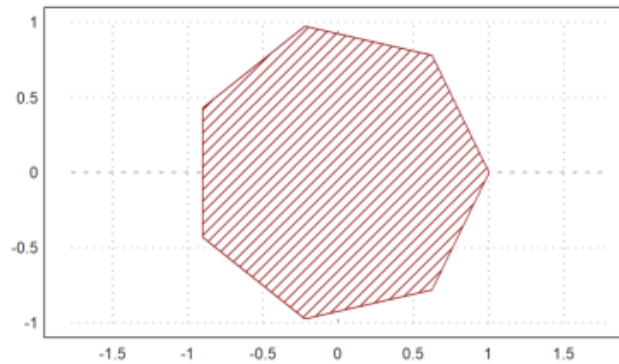


```
>t = linspace(0, 2pi, 6); plot2d(cos(t), sin(t), >filled, style="#"):
```



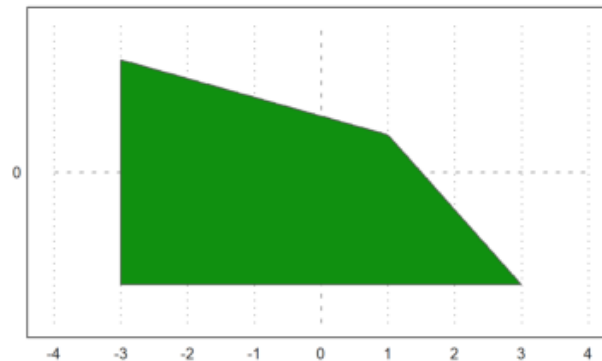
Contoh lainnya adalah sebuah segi-7, yang mana kami membuat 7 titik-titik dalam lingkaran.

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



Contoh ini adalah barisan dari nilai maksimal dari empat kondisi linear kurang atau sama dengan 3. Ini adalah $A[k].v \leq 3$ untuk semua barisan A. Untuk membuat sudut yang bagus, kita menggunakan n dengan relatif yang 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):
```

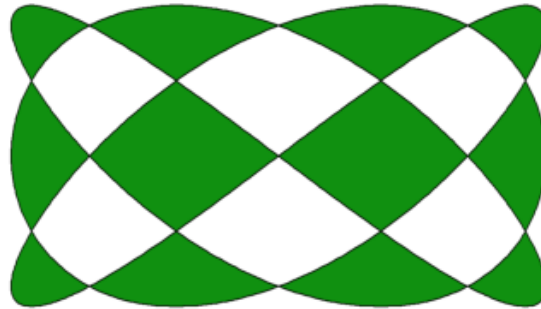


Titik utama dari bahasa matriks adalah membiarkannya untuk membuat tabel dari fungsi secara mudah.

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

Kami sekarang memiliki vektor x dan y dari nilai-nilai. `plot2d()` dapat membuat plot nilai-nilai ini sebagai sebuah kurva yang terkoneksi dengan titik-titik. Plot dapat diisi. Dalam kasus ini menghasilkan hasil yang bagus karena aturan berliku, yang mana digunakan untuk pengisian.

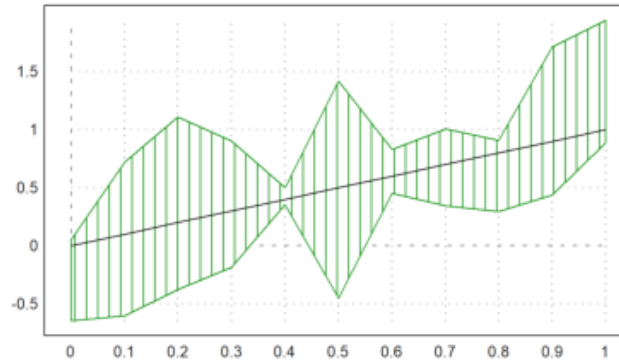
```
>plot2d(x, y, <grid, <frame, >filled):
```



Sebuah vektor dari interval-interval diplotkan dengan nilai-nilai x sebagai wilayah antara batas bawah dan batas atas dari interval.

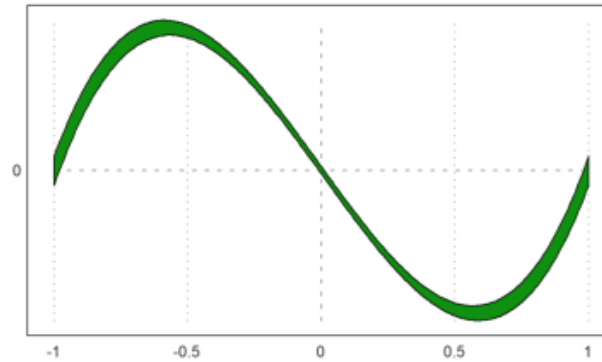
Ini dapat berguna untuk membuat galat plot dari komputasi. Tapi ini dapat juga digunakan untuk galat statistik.

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



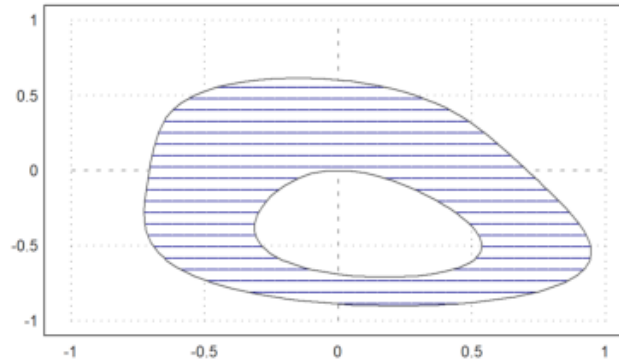
Jika x adalah vektor yang terurut dan y adalah interval vektor, lalu `plot2d` akan diplotkan pada rentang yang terisi dari interval dalam bidang. Gaya-gaya pengisian sama seperti gaya-gaya dari poligon.

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



Sangat mungkin untuk mengisi nilai nilai untuk sebuah fungsi speifik. Untuk ini, level harus sebuah matriks 2xn. Baris pertama adalah batas bawah dan baris kedua mencakup batas atas.

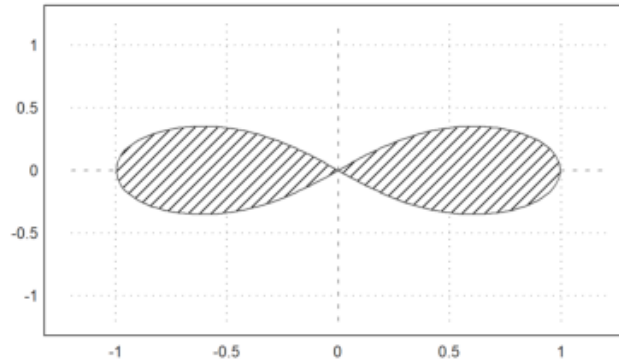
```
>expr := "2*x^2+x*y+3*y^4+y"; // mendefinisikan ekspresi f(x, y)
>plot2d(expr, level=[0; 1], style="-", color=blue): // 0 <= f(x, y) <= 1
```



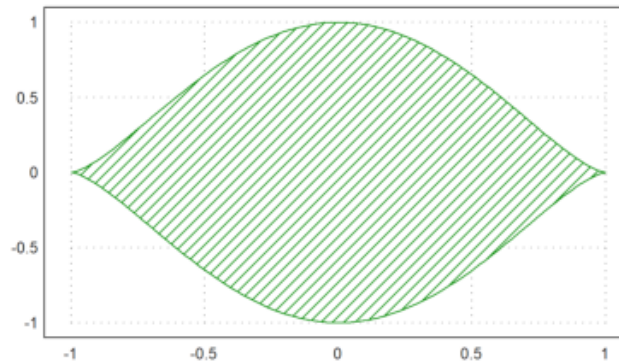
Kami dapat juga mengisi rentang dengan nilai seperti

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

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



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



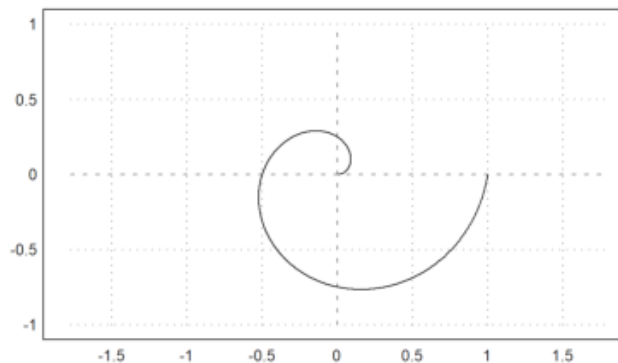
Grafik Fungsi Parametrik

Nilai x butuh tidak diurutkan. (x, y) secara sederhana menggambarkan sebuah kurva. Jika x terurut, kurva adalah sebuah grafik dari sebuah fungsi. Contoh ini, kami akan membuat plot spiral

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

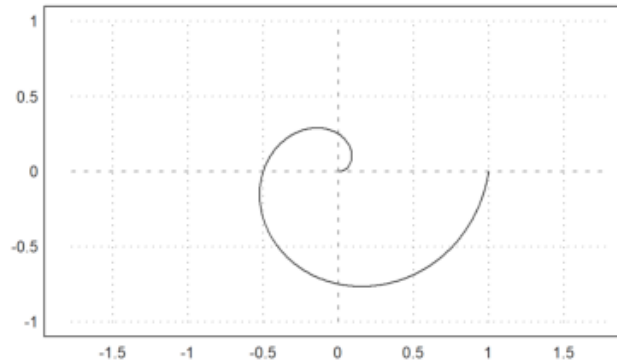
Kami juga membutuhkan banyak titik-titik untuk memperhalus atau fungsi `adaptive()` untuk mengevaluasi ekspresi (lihat fungsi `adaptive()` untuk lebih detail).

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

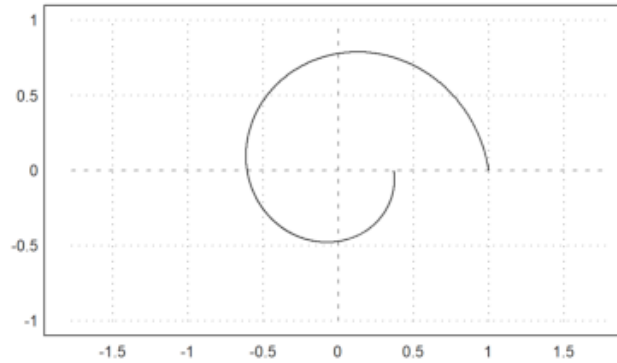


Sebagai alternatif, yang memungkinkan untuk menggunakan 2 ekspresi untuk kurva. Contoh plot ini sama seperti kurva diatas.

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



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



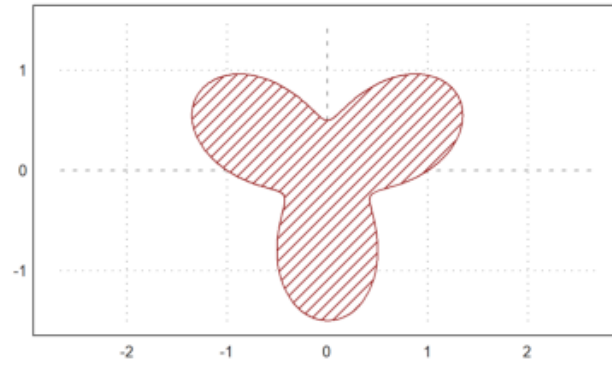
Contoh selanjutnya, kami membuat plot dari kurva

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

dengan

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

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



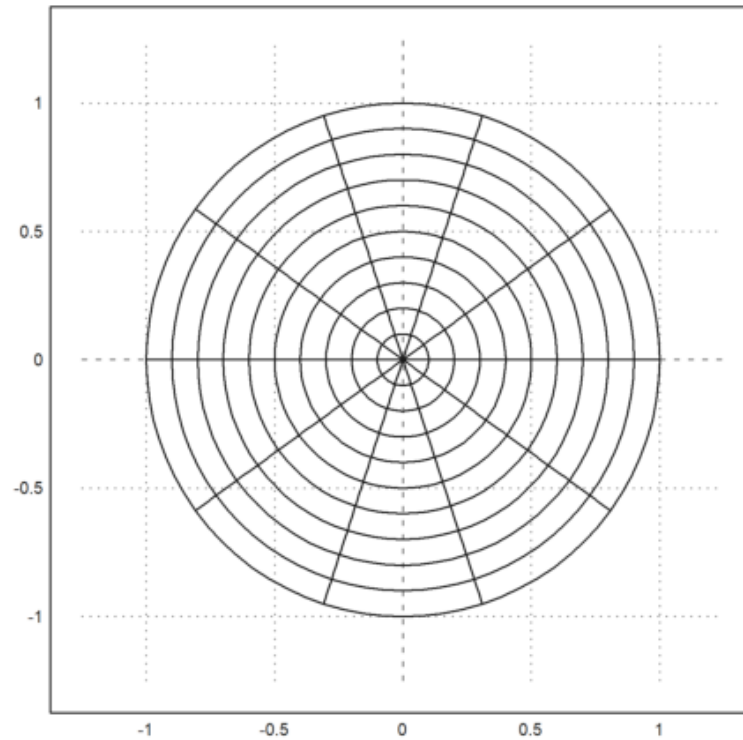
Menggambar Grafik Bilangan Kompleks

Sebuah array dari angka-angka kompleks juga dapat dibuat plot. Lalu titik-titik batas akan terkoneksi. Jika sebuah angka dari batas garis yang dispesifikasikan (atau sebuah 1x2 vektor dari batas garis) dalam sebuah argumen cgrid hanya terlihat baris batas ini.

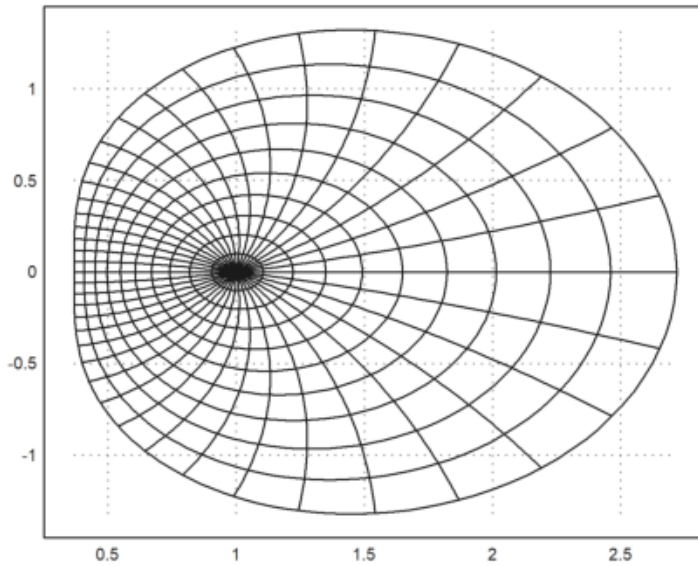
Sebuah matriks dari angka-angka kompleks akan secara otomatis diplotkan sebagai sebuah batas dari bidang kompleks.

Sebagai contoh, kami membuat plot gambar dengan lingkaran unit dari fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva batasnya.

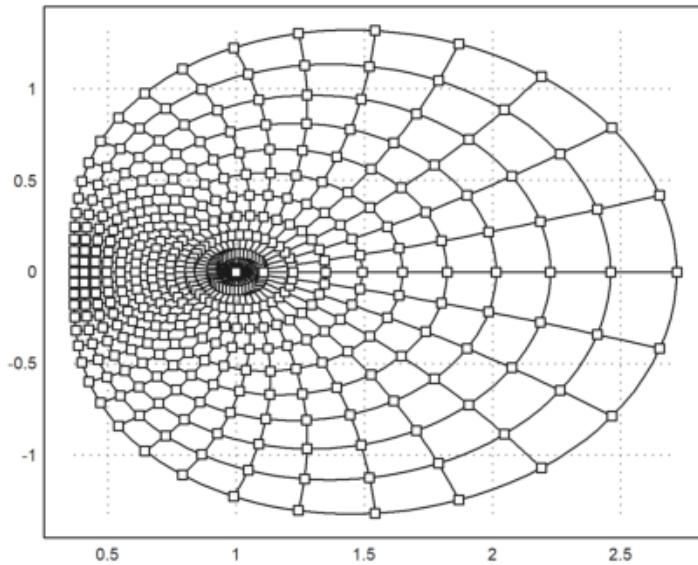
```
>aspect(); r = linspace(0, 1, 50); a = linspace(0, 2pi, 80)'; z = r*exp(I*a); ...  
>plot2d(z, a=-1.25, b=1.25, c=-1.25, d=1.25, cgrid=10):
```



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



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

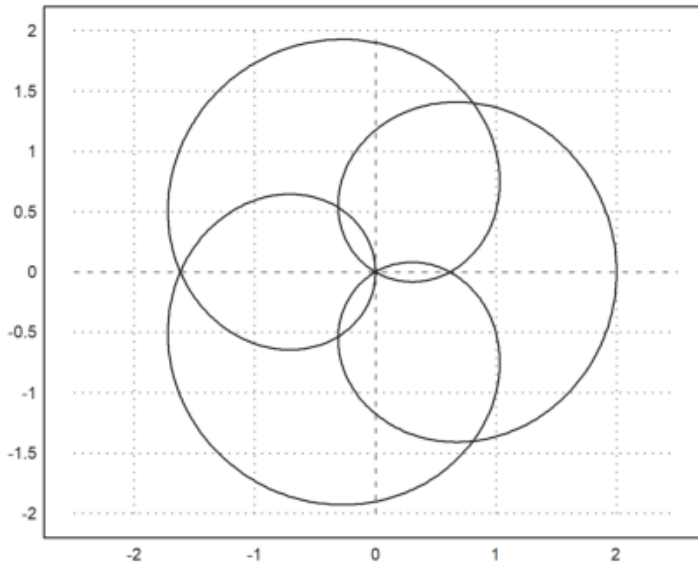


Sebuah vektor dari angka-angka kompleks secara otomatis terplotkan sebagai sebuah kurva dalam bidang kompleks dengan bagian riil dan bagian imajiner.

Contoh berikut, kami membuat plot dengan unit lingkaran dengan

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

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

Plot-Plot Statistik

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

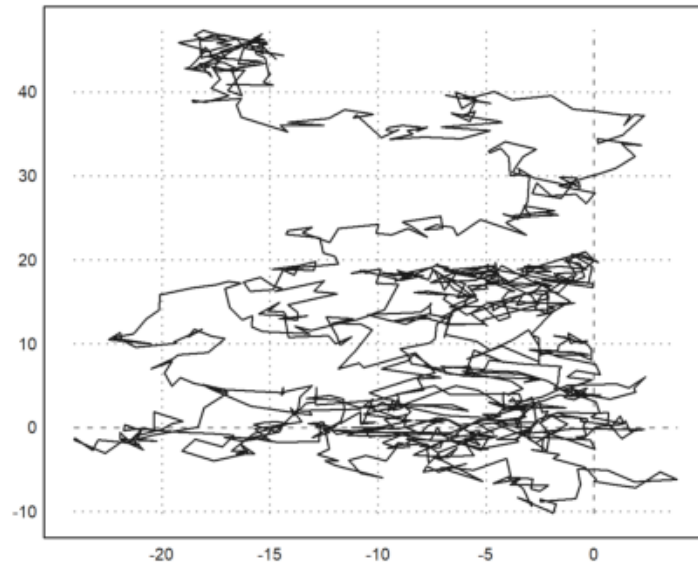
Sebuah jumlahan kumulatif dari sebuah 0-1 distribusi nilai normal menghasilkan pergerakan secara acak.

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

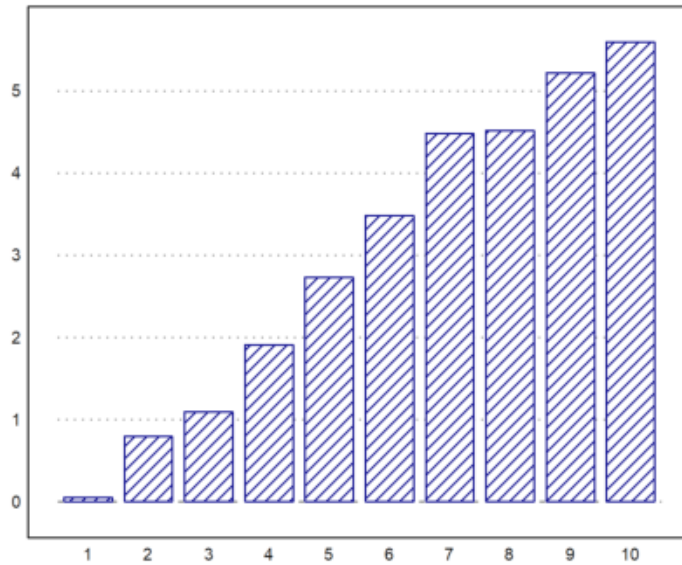


Menggunakan dua baris untuk menunjukkan dua dimensi.

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

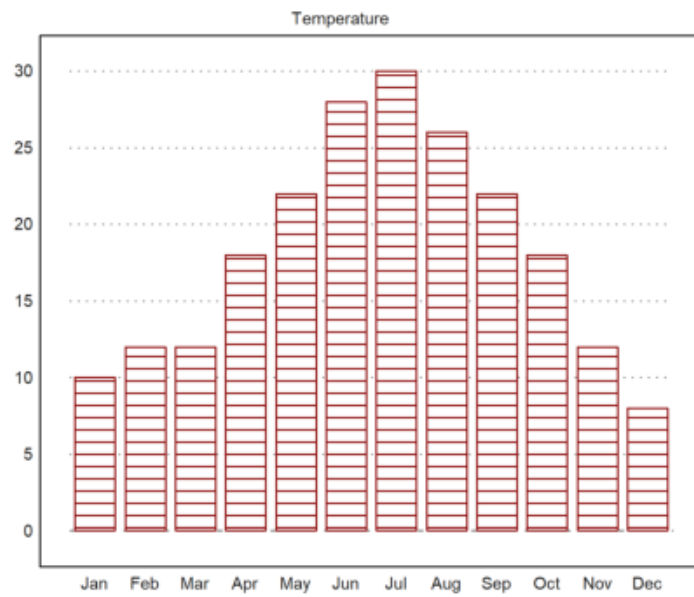


```
>columnspot(cumsum(random(10)), style="/", color=blue):
```

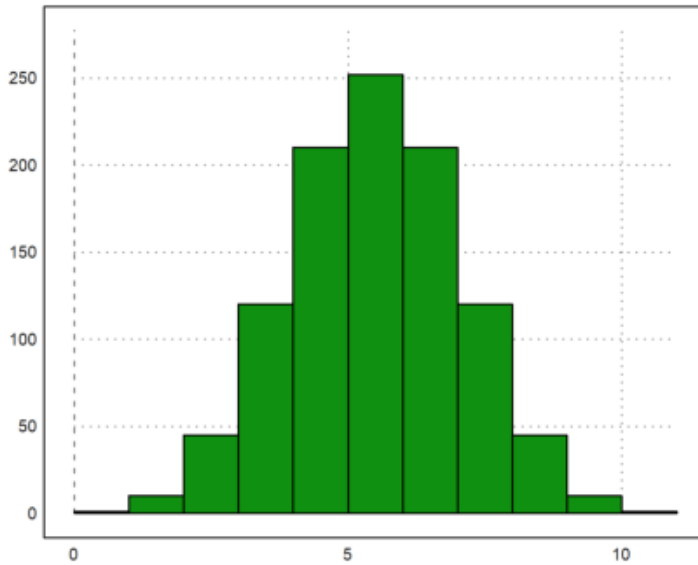


Ini dapat juga menunjukkan string sebagai label.

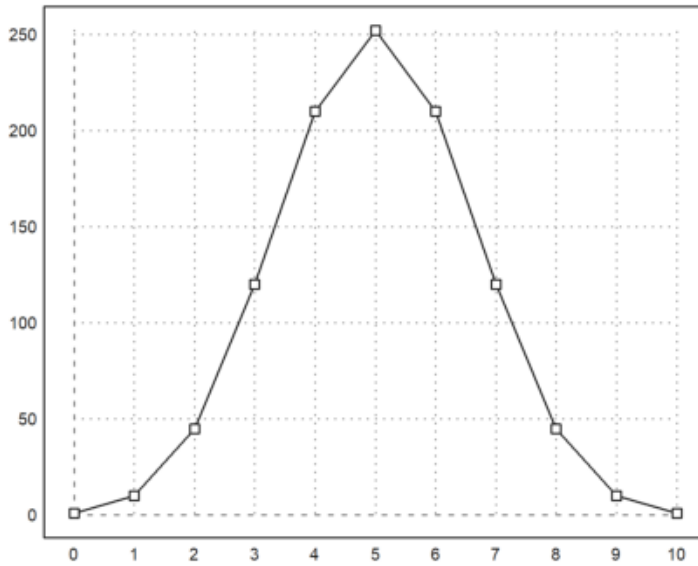
```
>months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", ...  
>"Jul", "Aug", "Sep", "Oct", "Nov", "Dec"];  
>values = [10, 12, 12, 18, 22, 28, 30, 26, 22, 18, 12, 8];  
>columnplot(values, lab=months, color=red, style="-");  
>title("Temperature"):
```



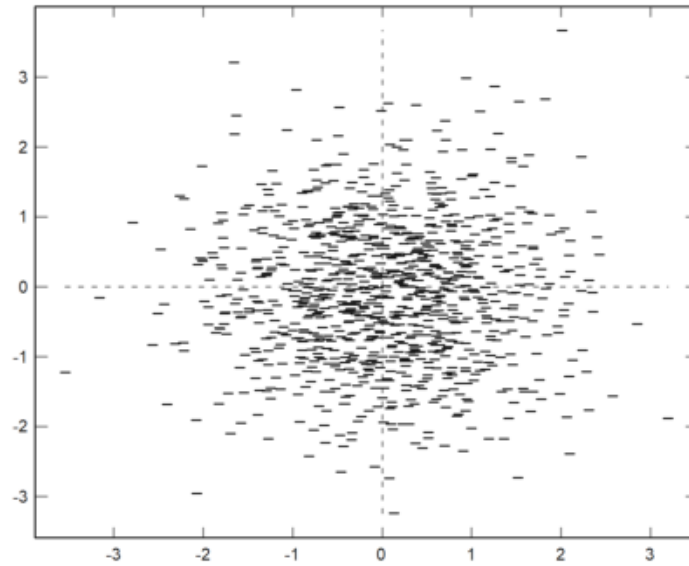
```
>k = 0:10;  
>plot2d(k, bin(10, k), >bar):
```



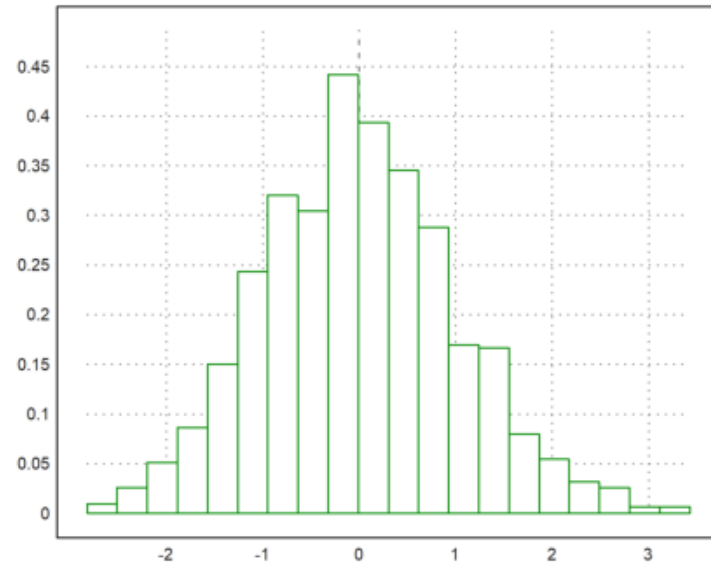
```
>plot2d(k, bin(10, k)); plot2d(k, bin(10, k), >points, >add):
```



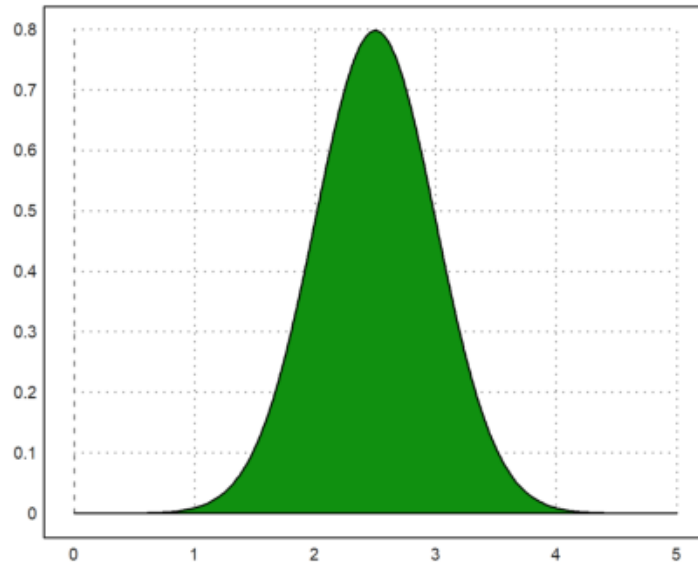
```
>plot2d(normal(1000), normal(1000), >points, grid=6, style="-"):
```



```
>plot2d(normal(1, 1000), >distribution, style="0"):
```

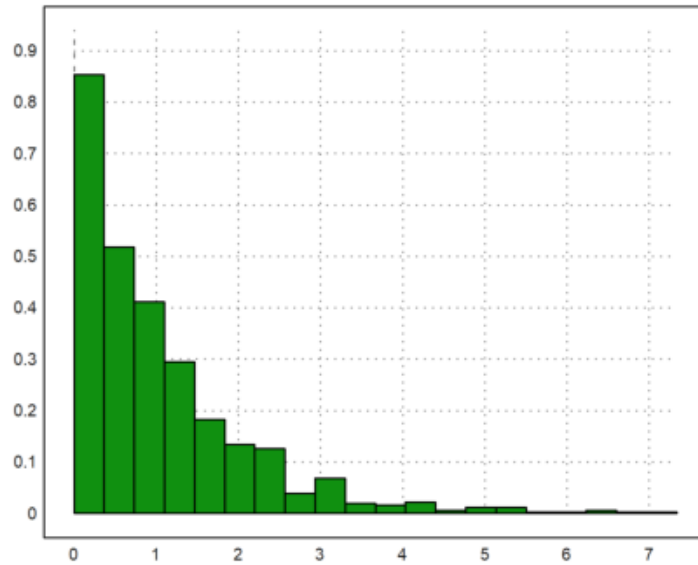



```
>plot2d("qnormal", 0, 5; 2.5, 0.5, >filled):
```



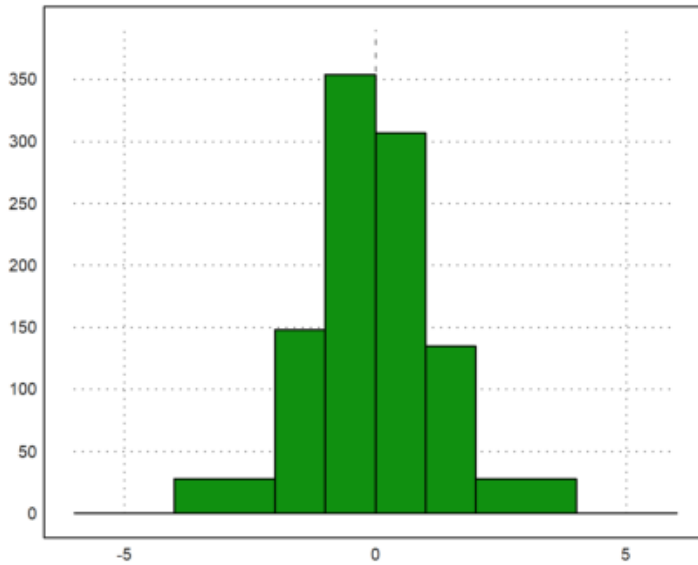
Untuk plot sebuah distribusi eksperimen statistik, kamu juga dapat menggunakan `distribution=n` dengan `plot2d`.

```
>w = randexponential(1, 1000); // distribusi eksponensial  
>plot2d(w, >distribution): // atau distribution=n dengan interval n
```



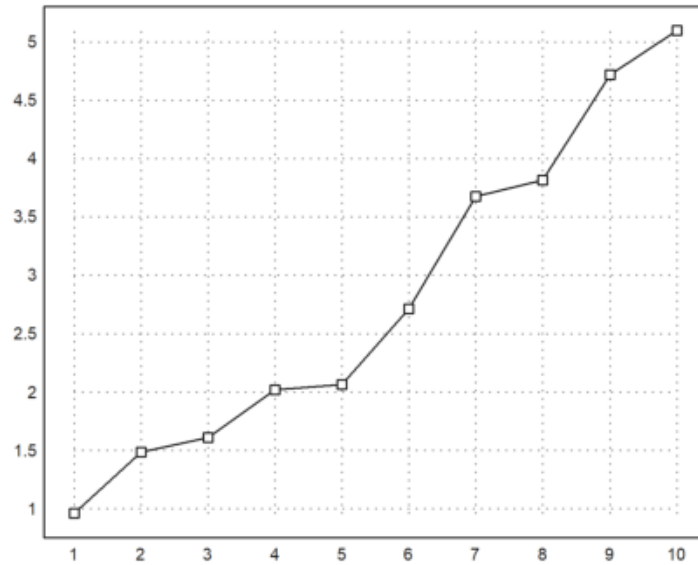
Atau kamu juga menghitung bentuk distribusi data dan plot hasil dengan `>bar` dalam `plot3d`, atau dengan sebuah plot kolom.

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

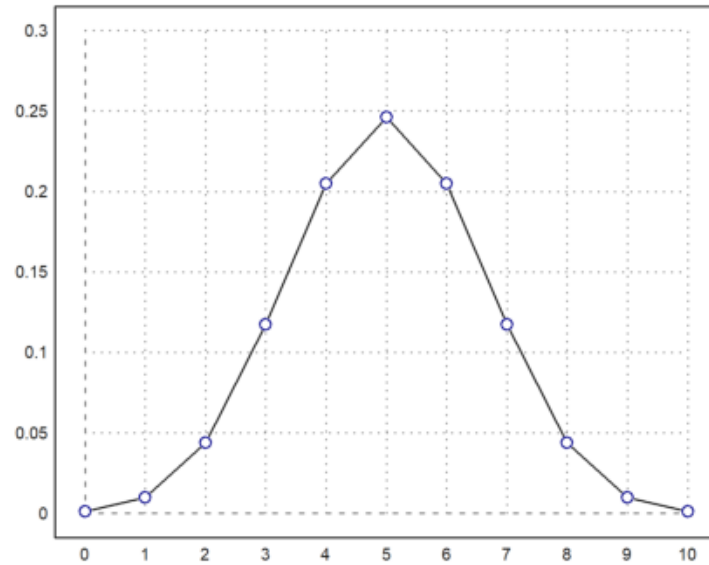


fungsi `statplot()` mengatur gaya dengan string yang sederhana.

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



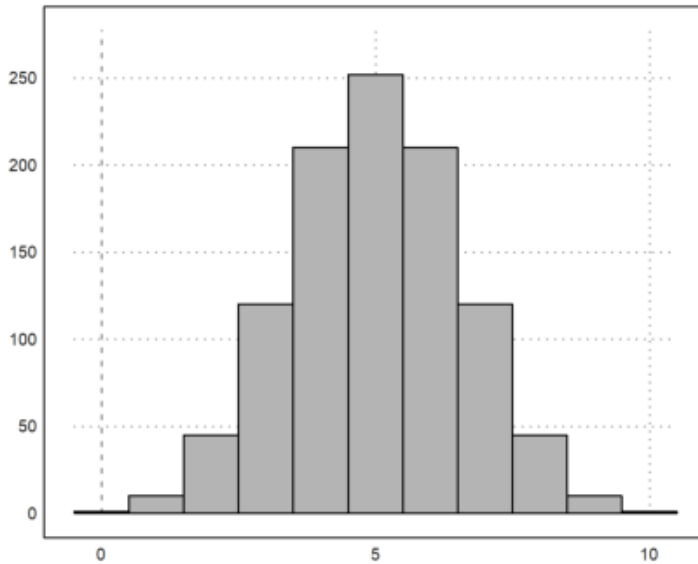
```
>n = 10; i = 0:n; ...  
>plot2d(i, bin(n, i) / 2^n, a=0, b=10, c=0, d=0.3);...  
>plot2d(i, bin(n, i) / 2^n, >points, >add, style="ow", color=blue):
```



Selebihnya, data dapat diplotkan sebagai baris. Dalam kasus ini, x akan diurutkan dan satu elemen lebih panjang daripada y. Baris akan memanjang dari $x[i]$ ke $x[i+1]$ dengan nilai $y[i]$. Jika x memiliki ukuran yang sama seperti y, akan memanjang dengan satu elemen dengan spasi dibagian akhir.

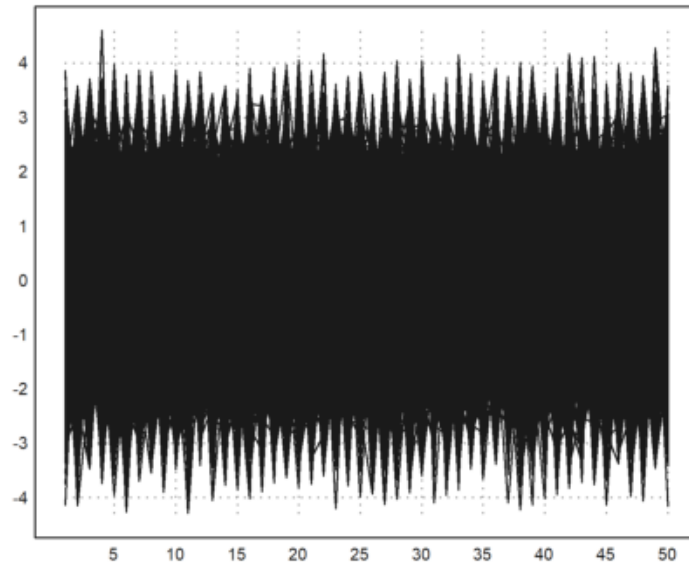
Gaya isian akan digunakan seperti diatas.

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

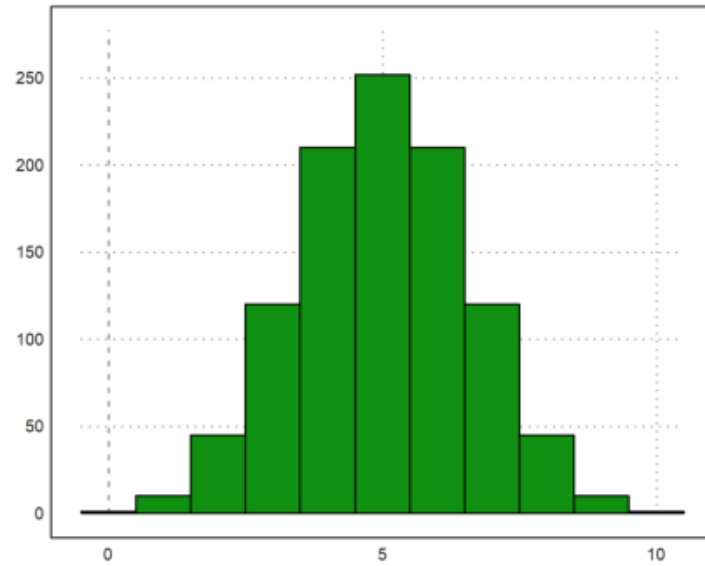


Data untuk plot batang (`bar = 1`) dan histogram (`histogram = 1`) dapat secara eksplisit diberikan dalam `xv` dan `yv` atau dapat dihitung dari sebuah distribusi empiris dalam `xv` dengan `>distribution` (or `distribution = n`). Histogram dari nilai `xv` dapat dihitung secara otomatis dengan `>histogram`.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

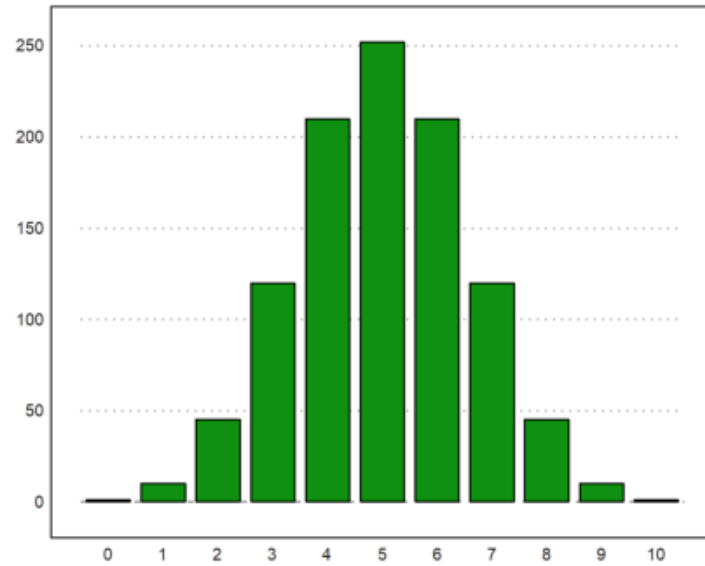
```
>plot2d(normal(10000, distribution=50)):
```



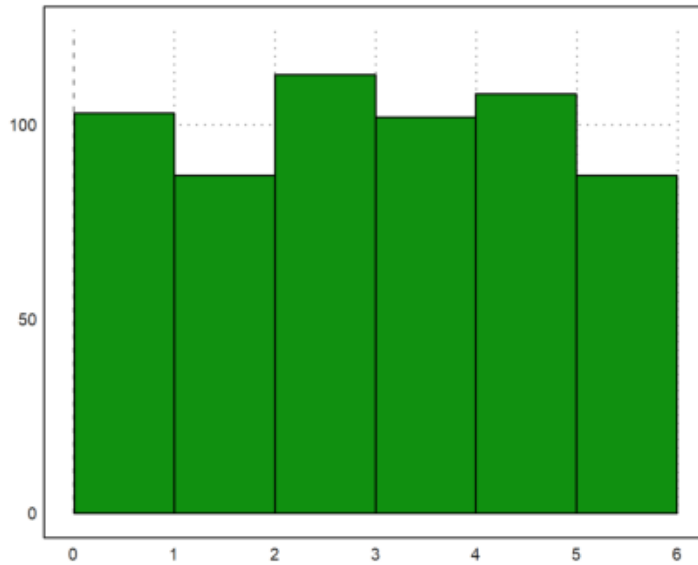
```
>k = 0:10; m = bin(10, k); x = (0:11)-0.5; plot2d(x, m, >bar):
```

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

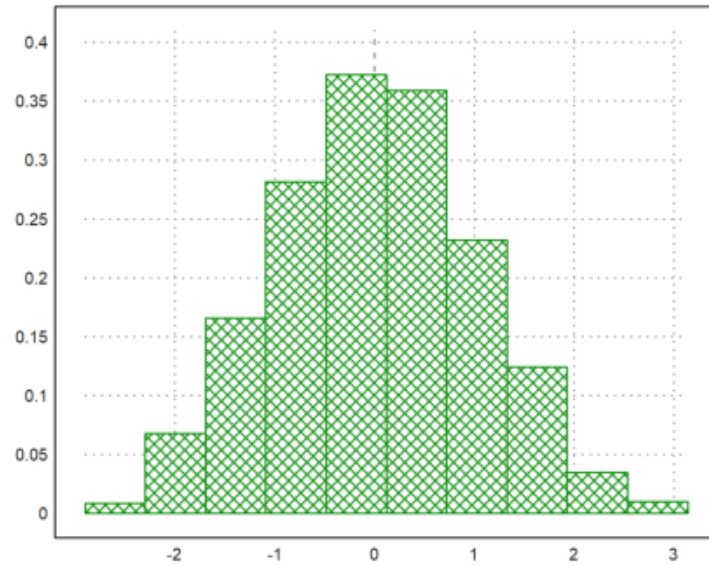


```
>plot2d(random(600) * 6, histogram=6):
```



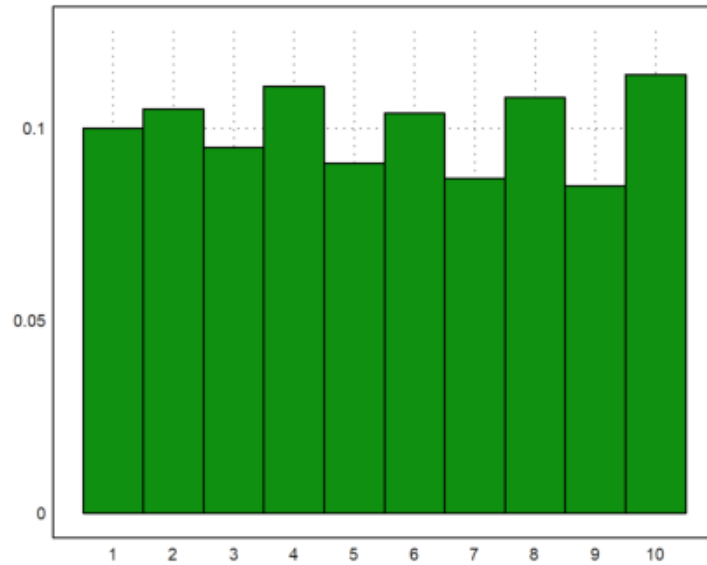
Untuk distribusi, parameter distribusi = n ini, yang mana menghitung nilai-nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.

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



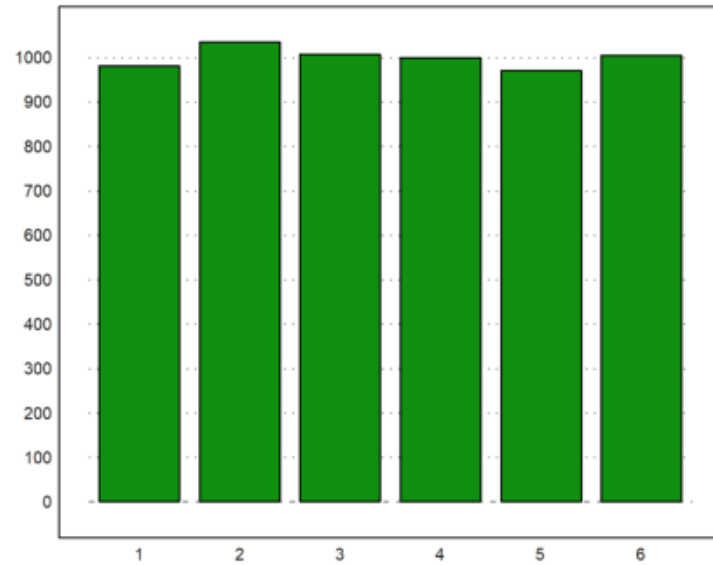
Dengan parameter `even=true`, ini akan menggunakan interval bilangan bulat.

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

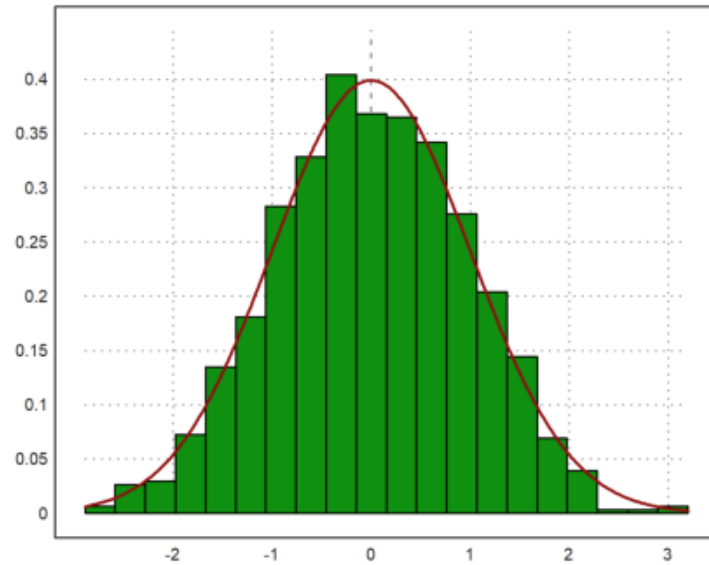


Catat bahwa terdapat banyak plot statistik, yang mana mungkin akan berguna. Lihatlah tutorial tentang statistik.

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

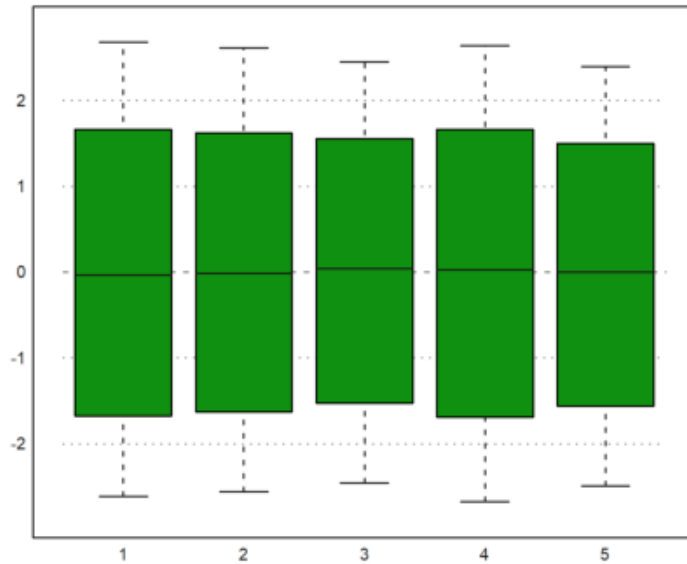


```
>plot2d(normal(1, 1000), >distribution); ...  
>plot2d("qnormal(x)", color=red, thickness=2, >add):
```



Ini juga terdapat banyak plot spesial untuk statistik. Sebuah boxplot memperlihatkan kuartil dari distribusi ini dan banyak luaran. Dengan definisi, luaran dalam sebuah boxplot adalah data yang melebihi 1.5 kali dari tengahan 50% rentang dari plot.

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



Fungsi-Fungsi Implisit

Plot implisit menunjukkan barisan level dengan menyelesaikan $f(x, y) = \text{level}$, dimana "level" dapat berupa nilai tunggal atau sebuah vektor dari nilai. Jika `level="auto"`, ini akan menjadi `nc` baris level, yang mana akan menyebar diantara nilai minimum dan maksimum dari fungsi. Gelap atau terangnya warna dapat ditambahkan dengan `>huv` untuk menunjukkan nilai dari fungsi. Untuk fungsi implisit, `xv` harus menjadi sebuah fungsi dari parameter `x` dan `y`, atau, secara alternatif, `xv` dapat menjadi sebuah matriks dari nilai.

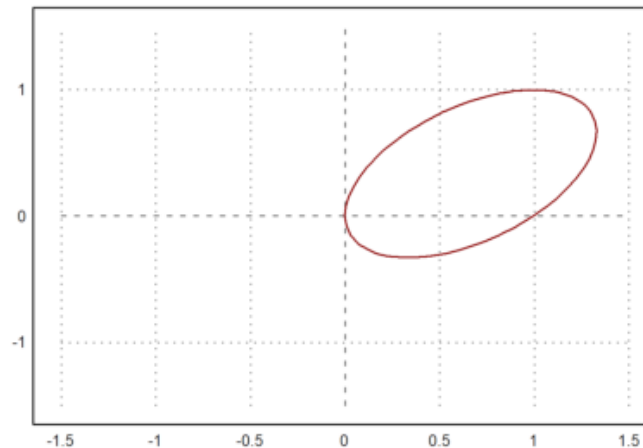
Euler dapat menandai garis level

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

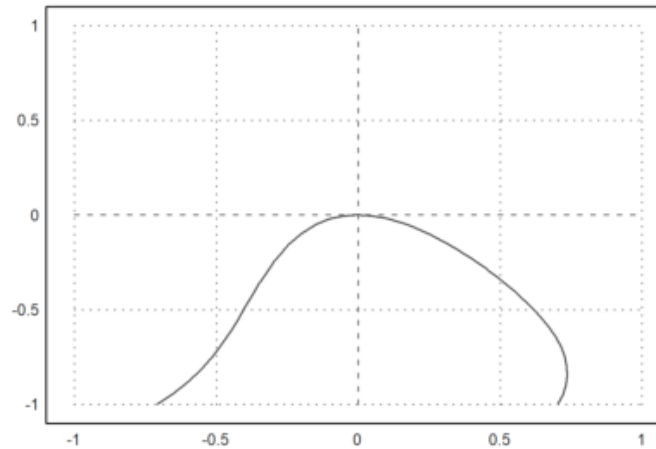
dari sebarang fungsi

Untuk menggambar himpunan $f(x, y) = c$ untuk satu atau lebih konstan c kamu dapat menggunakan `plot2d()` dengan ini plot implisit dalam bidang. Parameter untuk c adalah `level = c`, dimana c dapat berupa vektor dari baris `level`. Dengan tambahan, sebuah skema warna dapat digambarkan dalam latar belakang untuk mengindikasikan nilai dari fungsi untuk setiap titik dalam plot. Parameter "`n`" menentukan kemulusan dari plot.

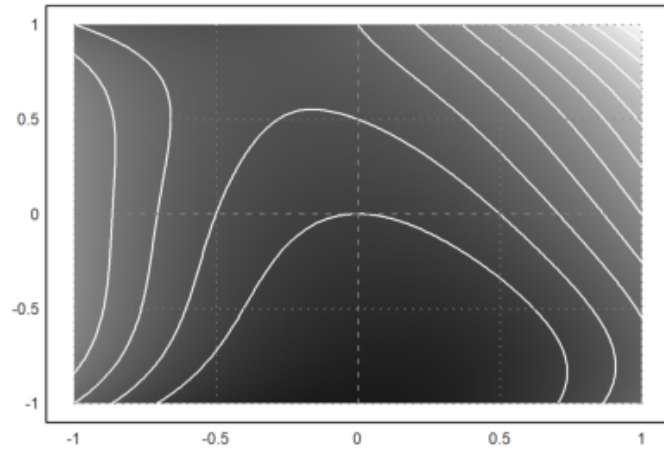
```
>aspect(1.5);
>plot2d("x^2 + y^2 - x*y - x", r=1.5, level=0, contourcolor=red):
```



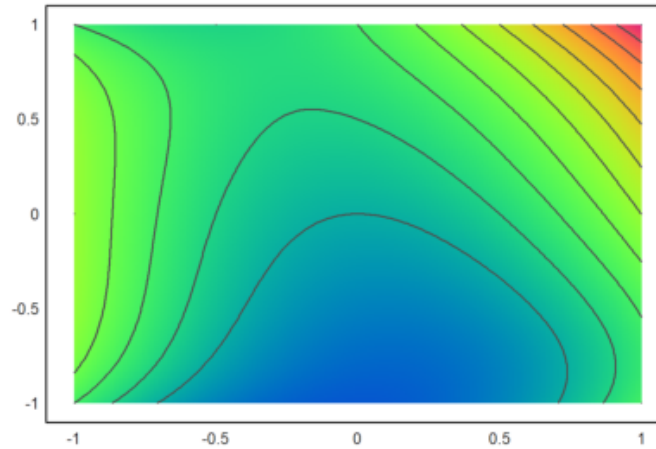
```
>expr := "2*x^2 + x*y + x*y^4 + y"; // mendefinisikan sebuah ekspresi f(x, y)
>plot2d(expr, level=0): // solusi dari f(x, y) = 0
```



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

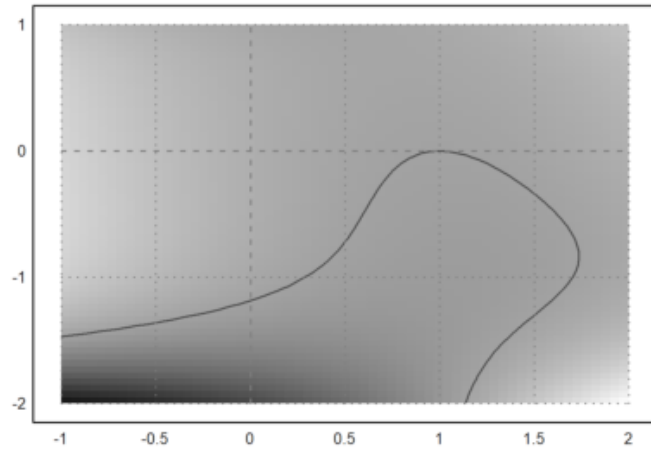


```
>plot2d(expr, level=0:0.5:20, >hue, >spectral, n=200, grid=4): // lebih bagus
```

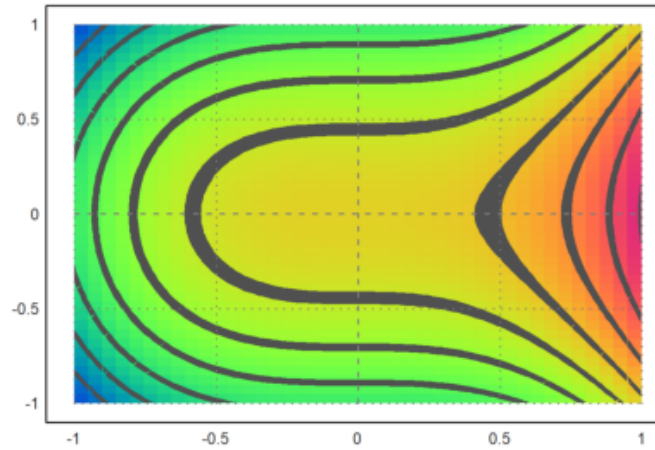


Ini bekerja untuk data plot juga. Tetapi kamu harus menspesifikasikan rentang untuk sumbu-sumbu label.

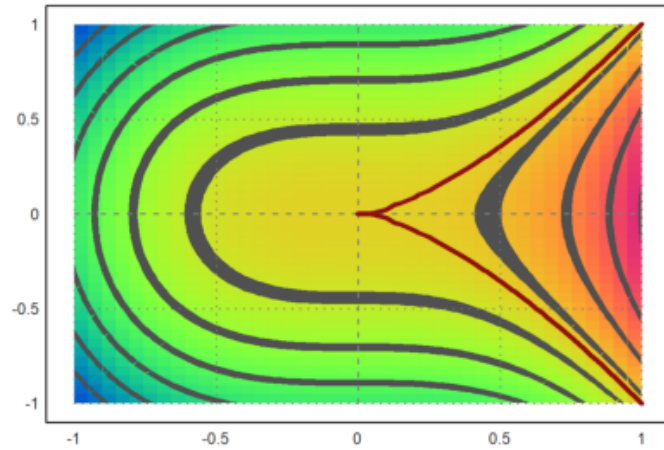
```
>x = -2:0.05:1; y = x'; z = expr(x, y);  
>plot2d(z, level=0, a=-1, b=2, c=-2, d=1, >hue):
```



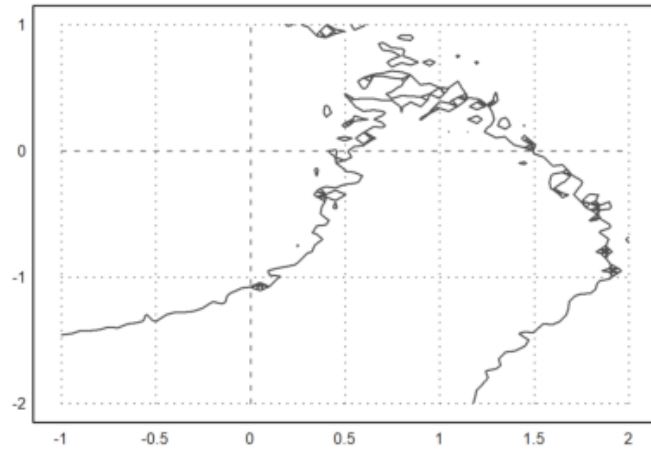
```
>plot2d("x^3-y^2", >contour, >hue, >spectral):
```



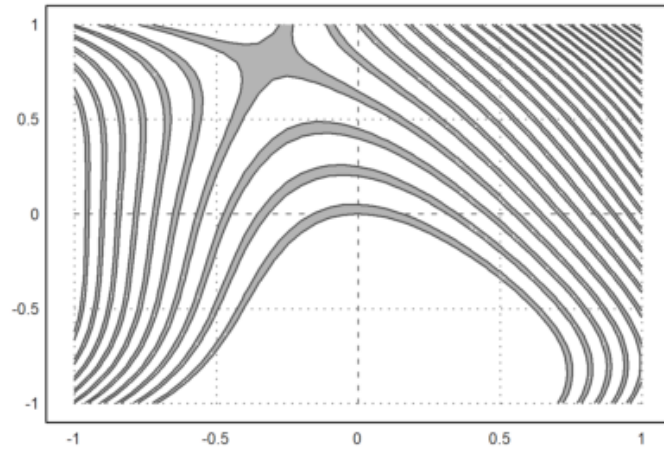
```
>plot2d("x^3-y^2", level=0, contourwidth=3, >add, contourcolor=red):
```



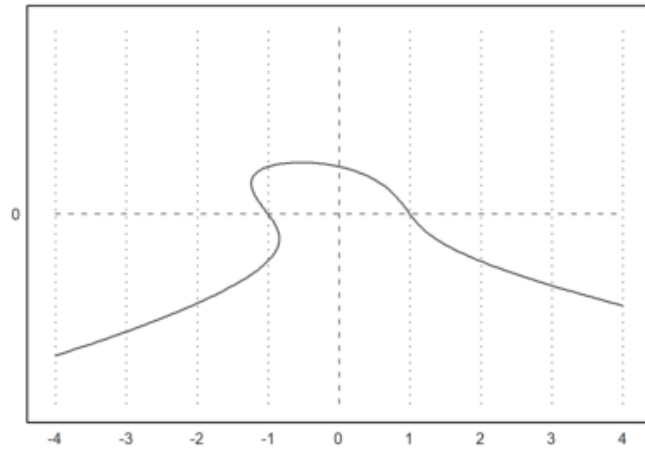
```
>z = z + normal(size(z)) * 0.2;  
>plot2d(z, level=0.5, a=-1, b=2, c=-2, d=1):
```



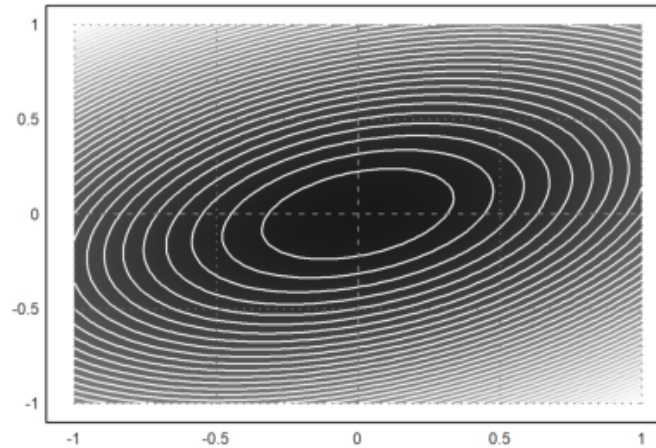
```
>plot2d(expr, level=[0:0.2:5; 0.05:0.2:5.05], color=lightgray):
```

```
>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):
```



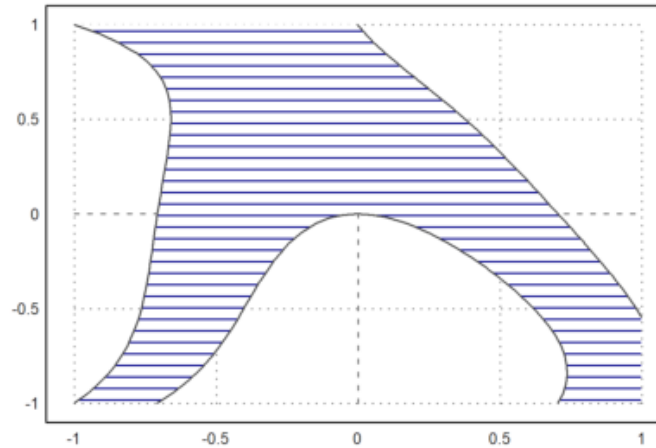
Ini juga memungkinkan untuk mengisi himpunan

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

dengan sebuah rentang level.

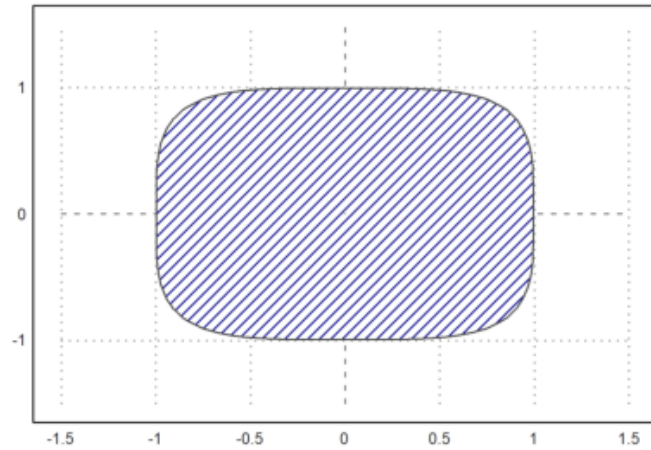
Ini memungkinkan untuk mengisi wilayah dari nilai-nilai untuk fungsi tertentu. Untuk ini, level haruslah sebuah matriks 2xn. Baris pertama adalah batasan bawah dan baris kedua mengandung batasan atas.

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

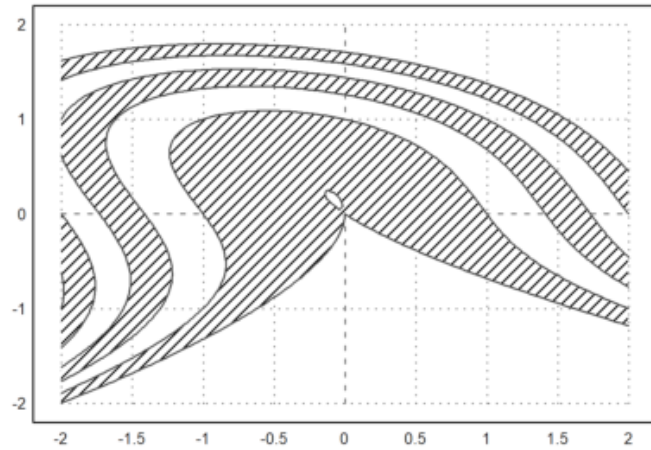


Plot implisit juga dapat memperlihatkan rentang dari level. Lalu level haruslah sebuah matriks 2xn dari interval level, dimana baris pertama mengandung awal dan baris kedua adalah akhir dari setiap interval. Sebagai gantinya, sebuah baris vektor dapat digunakan untuk level, dan sebuah dl parameter meluaskan nilai level untuk interval.

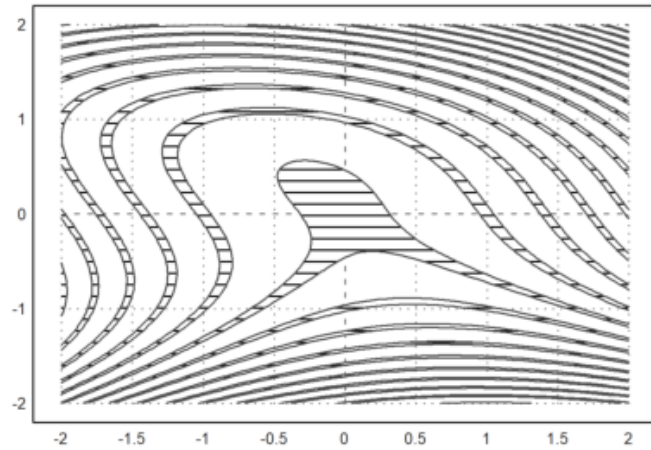
```
>plot2d("x^4 + y^4", r=1.5, level=[0;1], color=blue, style="/"):
```



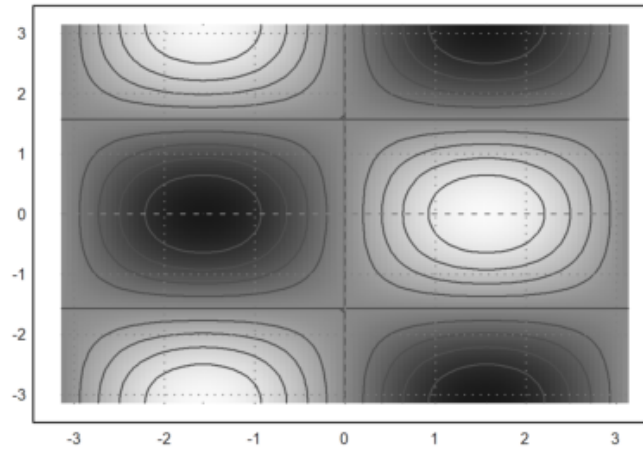
```
>plot2d("x^2 + y^3 + x*y", level=[0, 2, 4; 1, 3, 5], style="/", r=2, n=100):
```



```
>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):
```

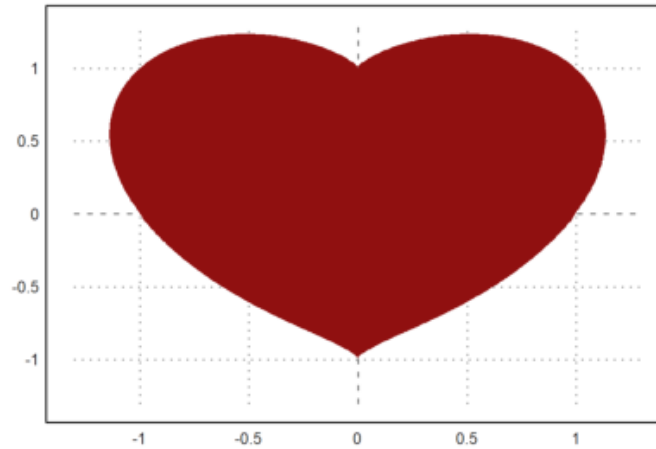


Ini juga memungkinkan untuk menandai sebuah wilayah

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

ini dapat dilakukan dengan menambahkan sebuah level dengan 2 baris.

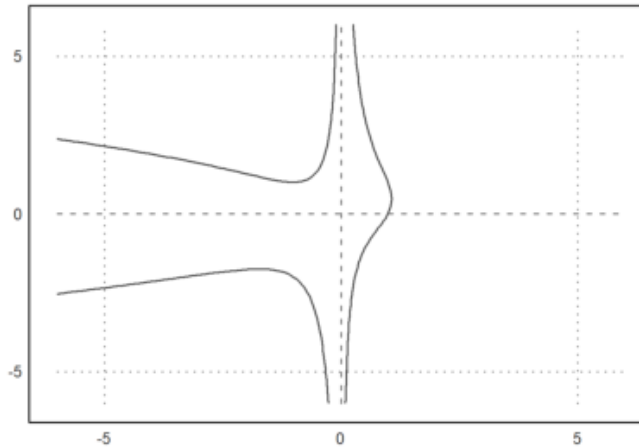
```
>plot2d("(x^2 + y^2 - 1)^3 - x^2*y^3", r=1.3,...
>style="#", color=red, <outline, ...
>level=[-2; 0], n=100):
```

Ini juga memungkinkan untuk secara level spesifik. Sebagai contoh, kami akan membuat plot solusi dari sebuah persamaan seperti

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

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



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

```

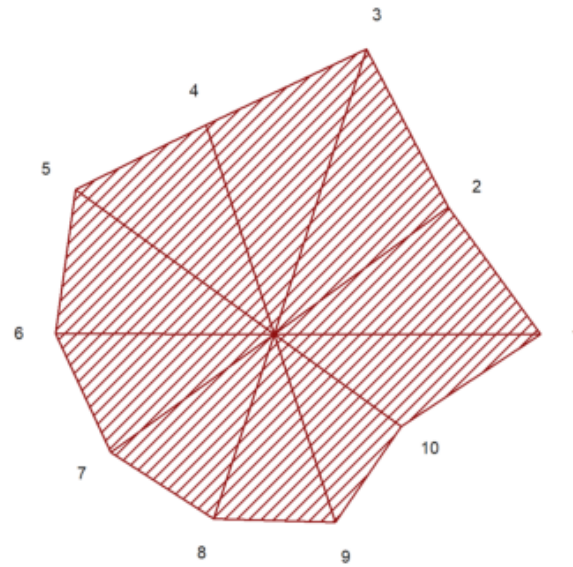
if !holding() then clg; endif;
w=window(); window(0,0,1024,1024);
h=holding(1);
r=max(abs(v))*1.2;
setplot(-r,r,-r,r);
n=cols(v); t=linspace(0,2pi,n);
v=v|v[1]; c=v*cos(t); s=v*sin(t);
cl=barcolor(color); st=barstyle(style);
loop 1 to n
  polygon([0,c[#],c[#+1]], [0,s[#],s[#+1]],1);
  if lab!=none then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

```

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

Tidak ada titik-titik atau batasan sumbu disini. Selebihnya, kami gunakan dengan jendela penuh untuk plot. Kami memanggil reset sebelumnya, kami test plot ini untuk mengatur ulang grafik menjadi seperti bawaan. Ini tidak dibutuhkan, jika kamu memastikan bahwa plot kamu berkerja.

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



Terkadang, kamu ingin untuk membuat plot sesuatu dengan plot2d tidak dapat melakukannya, tetapi hampir. Contoh fungsi berikut, kami melakukan plot impuls logaritmik. plot2d juga dapat mem 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

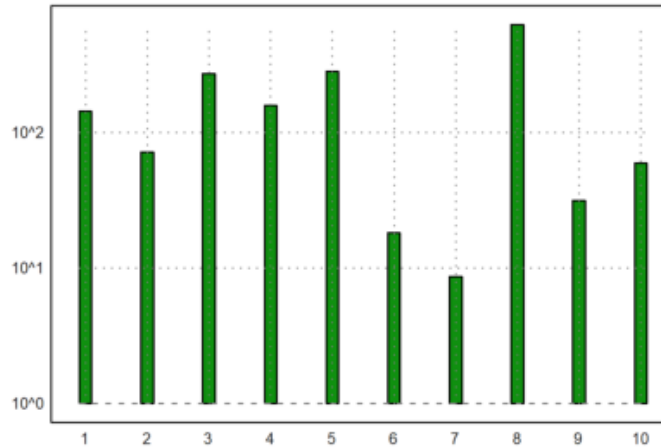
```

Mari kita tes ini dengan nilai yang terdistribusi secara eksponensial.

```

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

```



Mari kita animasikan sebuah kurva 2D menggunakan plot langsung. Perintah `plot(x, y)` secara sederhana membuat sebuah plot menjadi jendela plot. `setplot(a, b, c, d)` mengatur jendela ini.

Fungsi `wait(0)` memaksa plot untuk muncul dalam gambar windows. Sebaliknya, menggambar ulang mengambil tempat interval nilai untuk membuatnya.

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

```

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

```

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

Tekan tombol apapun untuk menghentikan animasi ini.

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

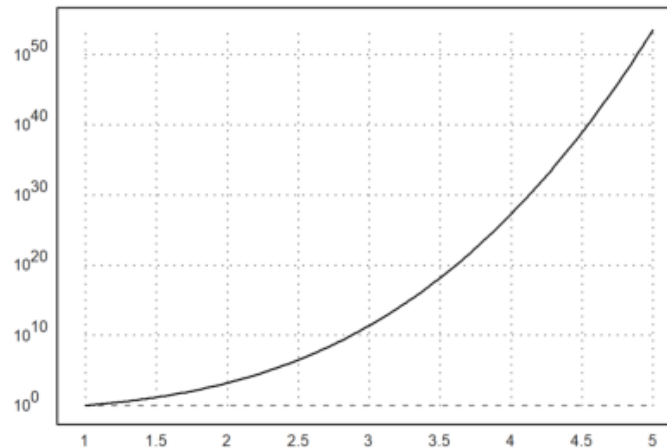
Plot-Plot Logaritmik

EMT menggunakan parameter "logplot" untuk skala logaritmik.

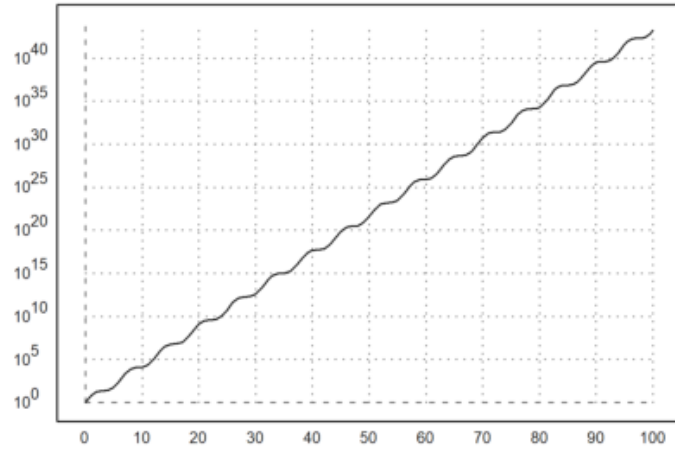
Plot logaritmik dapat diplotkan dengan menggunakan skala logaritmik dalam y dengan logplot=1 atau menggunakan skala logaritmik dalam x dan y dengan logplot=2 atau dalam x dengan logplot=3.

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

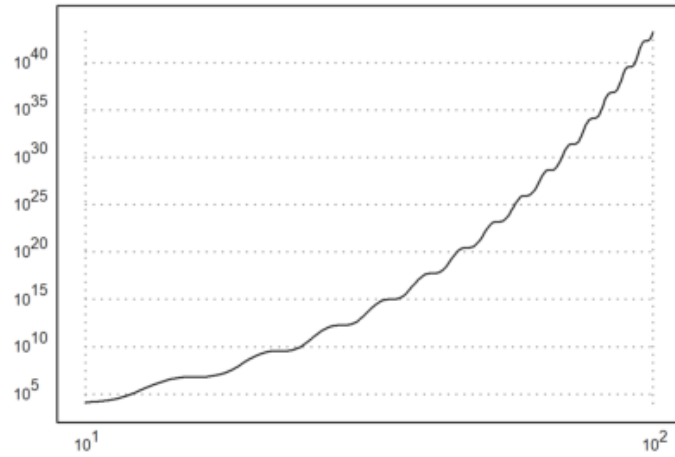
```
>plot2d("exp(x^3-x)*x^2", 1, 5, logplot=1):
```



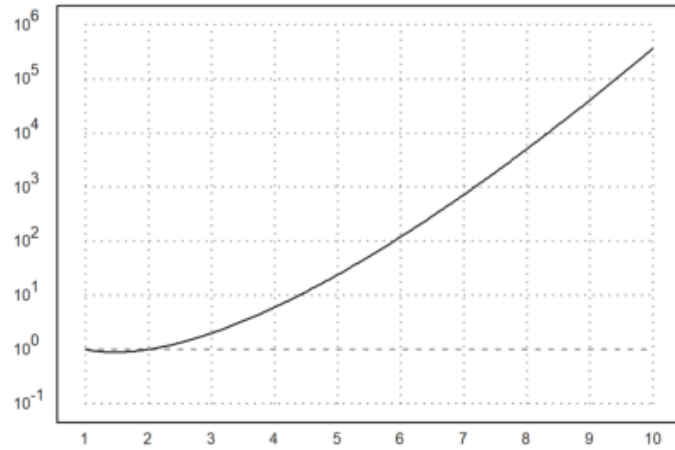

```
>plot2d("exp(x + sin(x))", 0, 100, logplot=1):
```



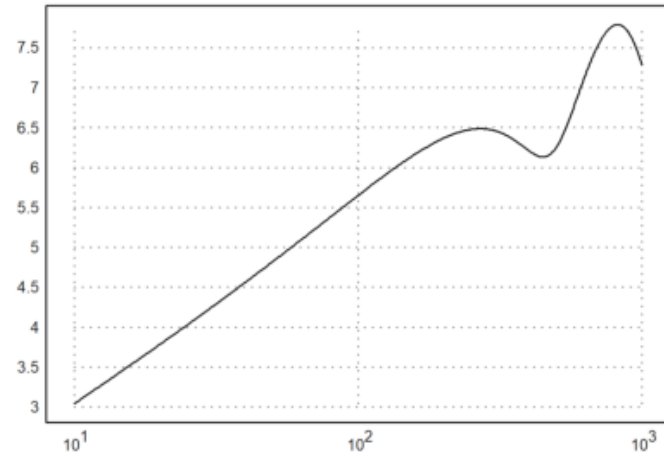
```
>plot2d("exp(x + sin(x))", 10, 100, logplot=2):
```



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

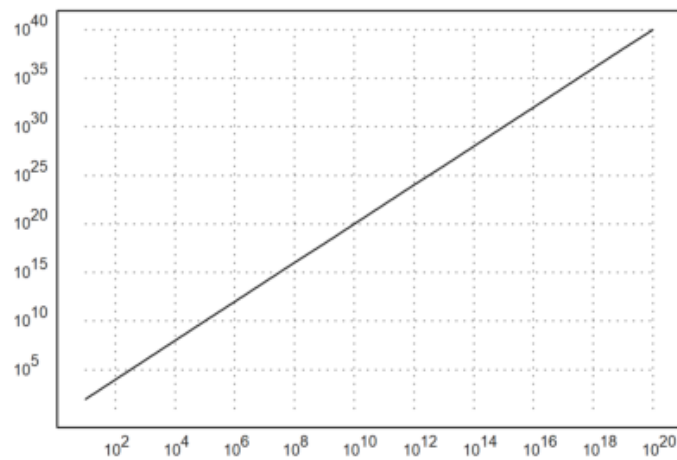


```
>plot2d("log(x * (2 + sin(x / 100)))", 10, 1000, logplot=3):
```



Ini juga dapat bekerja dengan plot-plot data.

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



Rujukan Lengkap Fungsi plot2d()

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

fungsi plot serbaguna untuk membuat plot dalam bidang (plot 2D). Fungsi ini dapat melakukan plotting fungsi satu variabel, plot data, kurva bidang, plot garis, batas-batas angka kompleks, dan plot implisit dari fungsi dua variabel.

Parameter

x, y	: persamaan, fungsi atau vektor data
a, b, c, d	: luasan plot (secara bawaan a = -2, b = 2)
r	: jika r diatur, maka a=cx-r, b=cx+r, c=cy-r, d=cy+r r dapat berupa sebuah vektor [rx, ry] atau sebuah vektor [rx1, rx2, ry1, ry2]
xmin,xmax	: selang dari parameter untuk kurva
auto	: Menentukan selang-y secara otomatis (bawaan)
square	: jika true, akan mempertahankan kotak selang-x-y
n	: jumlah rentang interval (adaptif secara default)
grid	: 0 = tidak mempunyai batas dan label, 1 = hanya sumbu, 2 = batas normal(lihat bawah untuk jumlah baris batasan) 3 = didalam sumbu 4 = tidak ada batas 5 = batas penuh termasuk margin 6 = titik bantu pada frame 7 = hanya sumbu

8 = hanya sumbu, sub-titik bantu
 frame : 0 = tidak ada frame
 framecolor : warna dari frame dan batas
 margin : angka antara 0 dan 0.4 untuk margin sekeliling plot
 color : Warna kurva. Jika ini adalah sebuah vektor warna, ini akan digunakan pada setiap baris dari sebuah plot matriks.
 Dalam kasus ini titik-titik plot, seharusnya sebuah vektor kolom. Jika sebuah baris vektor atau sebuah matriks warna lengkap yang digunakan untuk titik plot, ini akan digunakan untuk setiap titik data.
 thickness : ketebalan garis dari kurva
 Nilai ini dapat lebih kecil dari 1 untuk garis yang sangat tipis.
 style : Gaya plot untuk garis, tanda dan isian.
 Untuk titik gunakan
 "[]", "<>", ".", "..", "...",
 "*", "+", "|", "-", "o"
 "[]#", "<>#", "o#" (bentuk isian)
 "[]w", "<>w", "ow" (tidak-transparan)
 Untuk penggunaan garis
 "-", "--", "-.", ".", "-.-", "-.-", "->"
 Untuk penggunaan isian poligon atau plot garis
 "#", "#0", "0", "/", "\", "\/",
 "+", "|", "-", "t"
 points : membuat plot titik tunggal daripada baris segmen
 addpoints : Jika true, segmen garis plot dan titik
 add : menambahkan plot kedalam plot yang telah ada
 user : membuat pengguna untuk berinteraksi pada fungsi
 delta : ukuran langkah untuk interaksi pengguna
 bar : plot batang (x adalah batas interval, y nilai interval)
 histogram : plot frekuensi dari x dalam subinterval n
 distribution=n : plot distribusi dari x dalam subinterval n
 even : gunakan dalam nilai untuk histogram otomatis
 steps : plot fungsi sebagai sebuah fungsi langkah (steps=1,2)

adaptive : penggunaan plot adaptif (n adalah jumlah minimal langkah)
level : plot garis level dari sebuah fungsi implisit dua variabel
outline : menggambar batasan dari selang level

Jika nilai level adalah sebuah matriks 2xn, selang dari level akan

digambarkan dalam warna menggunakan gaya isian. Jika luaran bernilai true, ini akan digambarkan sebagai warna kontur. Gunakan fitur ini, wilayah dari $f(x, y)$ diantara limit dapat dicatat.

hue : tambahkan warna hue untuk plot level untuk
mengindikasikan nilai fungsi
contour : Gunakan plot level dengan level otomatis
nc : Banyaknya baris level otomatis
title : Judul plot (bawaan "")
xl, yl : label untuk sumbu-x dan sumbu-y
smaller : jika > 0, akan memberi ruang lebih ke kiri untuk label
vertical :
Mengubah arah label secara vertikal. Ini mengubah variabel global
verticallabel secara local pada satu plot. Nilai 1 mengatur hanya
teks vertikal, nilai 2 digunakan untuk label numerik vertical pada
sumbu y.
filled : mengisi plot dari sebuah kurva
fillcolor : mengisi warna untuk batang dan kurva yang terisi
outline : batasan untuk poligon yang terisi
logplot : mengatur plot logaritmik
1 = plot logartimik dalam y,
2 = plot logaritmik dalam xy,
3 = plot logaritmik dalam x
own :
Sebuah string, yang mana titik-titik ke sebuah rutin plot sendiri.
Dengan >user, kamu akan mendapatkan interaksi pengguna seperti

dalam plot2d. Rentang akan diatur sebelum tiap-tiap panggilan pada fungsi.

maps : ekspresi peta (0 tercepat), fungsi selalu terpetakan

contourcolor : warna dari garis kontur

contourwidth : tebal dari garis kontur

clipping : mengalihkan cetakan (bawaan adalah true)

title :

Ini dapat digunakan untuk mendeskripsikan plot. Judul akan muncul diatas plot. Terlebih, sebuah label untuk sumbu x dan y akan ditambahkan dengan `xl="string"` atau `yl="string"`. Label lainnya akan ditambahkan dengan fungsi `label()` atau `labelbox()`. Judul dapat berupa sebuah string unicode atau sebuah gambar dari sebuah formula Latex.

cgrid :

Menentukan banyaknya baris batasan untuk plot dari batasan kompleks. Seharusnya sebuah pembagi dari ukuran matriks dikurangi 1 (banyak nya sub-interval). `cgrid` dapat berupa sebuah vektor `[cx, cy]`.

Pratinjau

Fungsi ini dapat membuat plot

- Ekspresi, pemanggilan collection atau fungsi dari satu variabel,
- Kurva parametrik,
- data x beserta data y,
- fungsi implisit,
- plot batang,
- batasan kompleks,
- poligon.

Jika sebuah fungsi atau ekspresi untuk xv diberikan, `plot2d` akan menghitung nilai yang rentang berikan menggunakan fungsi atau ekspresi. Ekspresi haruslah sebuah ekspresi dalam variabel x . Selang haruslah didefinisikan dalam parameter a dan b kecuali memiliki rentang bawaan $[-2, 2]$ akan digunakan. Selang- y akan secara otomatis dihitung, kecuali c dan d dispesifikasikan, atau sebuah jari jari r , yang mana menghasilkan selang $[-r, r]$ untuk x dan y . Untuk plot dari fungsi, `plot2d` akan menggunakan evaluasi adaptif dari fungsi secara bawaan. Untuk mempercepat membuat plot pada fungsi yang rumit, matikan `<adaptive ini`, dan secara pilihan mengurangi banyaknya interval n . Selebihnya, `plot2d()` akan secara bawaan menggunakan pemetaan. Dengan kata lain, ini akan menghitung elemen plot untuk elemen. Jika ekspresi kamu atau fungsi kamu dapat dipegang oleh vektor x , kamu dapat mengubah `<maps` agar tidak aktif untuk evaluasi yang lebih cepat.

Catatan bahwa plot adaptif selalu menghitung elemen untuk elemen. Jika fungsi atau ekspresi untuk keduanya xv dan untuk yv dispesifikasikan, `plot2d()` akan dihitung sebuah kurva dengan nilai xv sebagai koordinat- x dan nilai yv sebagai koordinat- y . Dalam kasus ini, selang harusnya didefinisikan untuk parameter menggunakan $xmin$, $xmax$. Ekspresi yang dimuat dalam string harus selalu berupa ekspresi dengan parameter variabel x .

Beberapa Materi untuk Fungsi dan Plot

Dalam matematika plot tidak dapat dipisahkan dengan sebuah fungsi. Karena plot itu sendiri merupakan gambaran/sketsa yang dihasilkan oleh fungsi tersebut. Dalam kasus ini, EMT menyediakan fungsi pemrograman yaitu `plot2d` yang mana dapat membuat plot untuk beberapa kondisi dalam suatu bidang dua dimensi. Ini berarti fungsi matematika yang akan kita pakai adalah fungsi untuk satu variabel, sederhananya $f(x) = y$.

Dalam bahasan selanjutnya, saya akan membahas beberapa fungsi yang ada di matematika, seperti:

- fungsi linear
- fungsi kuadrat
- fungsi polinomial
- fungsi eksponensial
- fungsi logaritmik
- fungsi trigonometri

Kemudian akan membahas fungsi khusus seperti fungsi tangga, fungsi atap, dan fungsi absolut.

Selanjutnya, akan dibahas beberapa plot dalam statistika, seperti plot batang, plot pencar, dan plot distribusi normal

Terakhir disuguhkan beberapa hal yang bisa dilakukan menggunakan EMT ini dengan sedikit bermain.

Fungsi Linear

Dalam matematika, sebuah fungsi dapat dikatakan linear apabila membentuk sebuah garis lurus. Dimana fungsi tersebut merupakan fungsi polinomial yang variabel-variabelnya memiliki pangkat tertinggi adalah 1.

Bentuk umum fungsi linear adalah

$$f(x) = y = mx + c$$

dengan:

- y adalah variabel tak bebas
- m adalah gradien
- x adalah variabel bebas
- c sebarang bilangan

Contoh:

1. Buatlah plot dengan garis lurus yang melalui titik (4, 2) dan (2, 6)

Penyelesaian:

Dapat menggunakan persamaan umum dari pencarian gradien, yaitu

$$\frac{(y - y_1)}{(y_2 - y_1)} = \frac{(x - x_1)}{(x_2 - x_1)}$$

dengan mengubah menjadi persamaan $y = \dots$

```
>persamaanY &= solve((y - y1) / (y2 - y1) = (x - x1) / (x2 - x1), y)
```

$$[y = \frac{- (x_1 - x) y_2 - (x - x_2) y_1}{x_2 - x_1}]$$

Kemudian masukkan titik titiknya, misalkan

$$y_1 = 2, y_2 = 6, x_1 = 4, x_2 = 2$$

```
>&persamaanY with [y1 = 2, y2 = 6, x1 = 4, x2 = 2]
```

$$[y = \frac{2(x - 2) + 6(4 - x)}{2}]$$

Kemudian ubah menjadi ekspresi program

```
>ekspresiPersamaanY = "(2 * (x - 2) + 6 * (4 - x)) / 2"
```

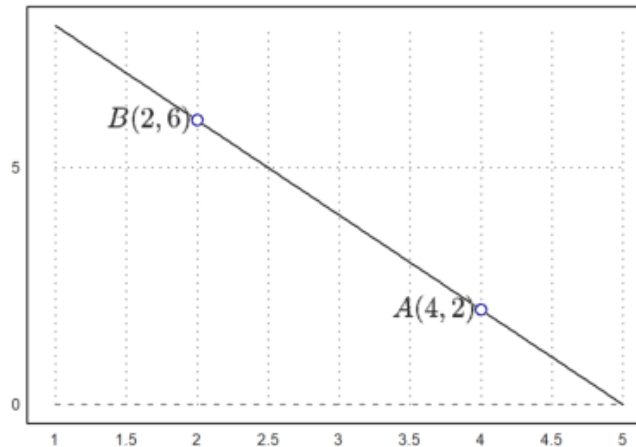
$$(2 * (x - 2) + 6 * (4 - x)) / 2$$

Terakhir adalah buat plotnya dengan memastikan titik-titik yang dilaluinya.

```

>y1 = 2;...
>y2 = 6;
>x1 = solve(ekspresiPersamaanY, 0, 1, y1); ...
>x2 = solve(ekspresiPersamaanY, 0, 1, y2); ...
>plot2d(ekspresiPersamaanY, 1, 5); ...
>plot2d(x1, y1, >points, >add, color=blue, style="ow");...
>label(latex("A(4, 2)"), x1, y1, pos = "cl", offset=10); ...
>plot2d(x2, y2, >points, >add, color=blue, style="ow");...
>label(latex("B(2, 6)"), x2, y2, pos = "cl", offset=10):

```



Jadi, persamaan garis lurus yang melalui titik (4, 2) dan (2, 6) adalah

$$y = \frac{2(x - 2) + 6(4 - x)}{2}$$

dengan plot seperti diatas

2. Buatlah plot persamaan grafik fungsi linear melalui titik (2, 4) dengan gradien 2.

Pembahasan:

Menggunakan rumus

$$y - y_1 = m(x - x_1)$$

dengan:

$$y_1 = 4, m = 2, x_1 = 2$$

```
>persamaan &= "y - y1 = m*(x - x1)"
```

$$y - y_1 = m (x - x_1)$$

```
>persamaanY &= solve(persamaan, y)
```

$$[y = y_1 - m x_1 + m x]$$

```
>&persamaanY with [m = 2, y1 = 4, x1 = 2]
```

$$[y = 2 \ x]$$

```
>ekspresiPersamaanY = "y = 2*x";  
>y1 = 4;  
>x1 = solve(ekspresiPersamaanY, 0, 1, y1);...
```

Error in Evaluate, superfluous characters found.

Error in expression: $y = 2 \cdot x$

secant:

$y_0 = f(x_0, \text{args}()) - y$; $y_1 = f(x_1, \text{args}()) - y$;

Try "trace errors" to inspect local variables after errors.

solve:

if eps then return secant(f\$,a,b,y;args(),eps=eps);

```
>plot2d(ekspresiPersamaanY, 0, 6); ...  
>plot2d(x1, y1, >points, >add, color=blue, style="ow");...  
>label(latex("A(2, 4)"), x1, y1, pos="c1", offset=10):
```

Jadi persamaan garis lurus bergradien 2 yang melalui (2, 4) adalah

$$y = 2x$$

dengan plot seperti diatas.

Fungsi Kuadrat

Biasanya fungsi kuadrat ini berupa parabola apabila di gambarkan. Definisi umum dari fungsi ini merupakan fungsi polinomial dengan pangkat tertinggi variabelnya adalah 2. Bentuk umumnya dinotasikan sebagai berikut

$$f(x) = ax^2 + bx + c, a \neq 0$$

Contoh soal

Buatlah grafik fungsi

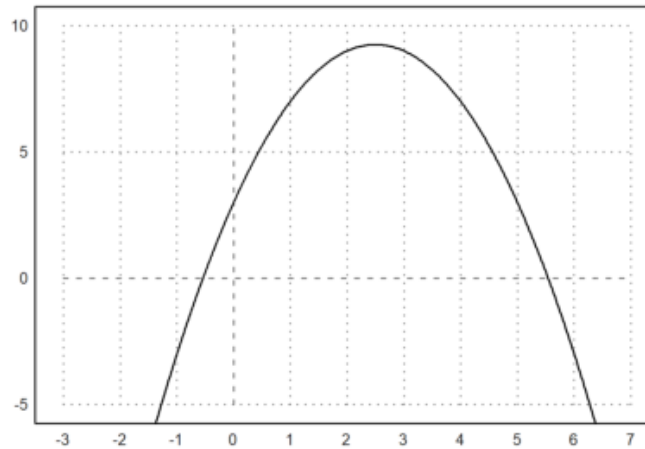
$$f(x) = -x^2 + 5x + 3$$

dalam sistem koordinat. Temukan titik puncak, titik potong terhadap sumbu-x dan sumbu-y.

Penyelesaian:

- Grafik fungsi

```
>function f(x) &= -(x)^2 + 5*x + 3;  
>plot2d(f, -3, 7, -5, 10):
```



- Titik puncak

Dapat dihitung dengan

$$\left(\frac{-b}{2a}, f\left(\frac{-b}{2a}\right)\right)$$

dengan:

$$a = -1, b = 5, c = 3$$

```
>titikPuncak &= "-b/2*a"
```

$$- \frac{a \ b}{2}$$

```
>titikPuncakX := &titikPuncak with [b = 5, a = -1]
```

$$\frac{5}{2}$$

```
>titikPuncakY = f(5/2)
```

Jadi titik puncak dari fungsi kuadrat tersebut adalah

$$(2.5, 9.25)$$

- Titik Potong sumbu-x

```
>ekspresiPersamaan &= "y = -x^2 + 5*x + 3";  
>titikPotongY := &solve(ekspresiPersamaan with y = 0, x)
```

$$[x = \frac{5 - \sqrt{37}}{2}, x = \frac{\sqrt{37} + 5}{2}]$$

Jadi diperoleh titik potong persamaan kuadrat tersebut terhadap sumbu-x adalah

$$(\frac{5 - \sqrt{37}}{2}, 0), (\frac{\sqrt{37} + 5}{2}, 0)$$

- Titik potong sumbu-y

```
>$&solve(ekspresiPersamaan with x = 0, y)
```

$$[y = 3]$$

Jadi diperoleh titik potong persamaan kuadrat tersebut terhadap sumbu-y adalah

$$(0, 3)$$

Funsi Polinomial

fungsi polinomial adalah sebuah fungsi yang dapat ditulis dengan bentuk umum:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

untuk n adalah bilangan bulat tidak negatif, dengan koefisiennya

$$a_0, a_1, \dots$$

adalah bilangan riil, dan

$$a_n \neq 0$$

Soal:

Buatlah plot dari fungsi

$$f(x) = x^3 - 4x^2 + 6x - 1$$

penyelesaian:

```
>function f(x) := x^3 - 4*x^2 + 6*x - 1;...  
>plot2d(&f, -2, 5, -5, 10):
```

