TUGAS PROYEK
APLIKASI KOMPUTER

Disusun oleh:
Sherlyta Icha Nadiastuty
22305141013

PROGRAM STUDI MATEMATIKA
JURUSAN PENDIDIKAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA
2023

Daftar Isi

1	PENDAHULUAN DAN PENGENALAN CARA KERJA EMT	1
2	MENGGUNAKAN EMT UNTUK MENYELESAIKAN MASALAH ALJABAR	23
3	MENGGUNAKAN EMT UNTUK MENGGAMBAR GRAFIK 2D	111
4	MENGGUNAKAN EMT UNTUK MENGGAMBAR PLOT 3D	223
5	MENGGUNAKAN EMT UNTUK KALKULUS	291
6	MENGGUNAKAN EMT UNTUK GEOMETRI	401
7	MENGGUNAKAN EMT UNTUK STATISTIKA	489

BAB 1

PENDAHULUAN DAN PENGENALAN CARA KERJA EMT

Pendahuluan dan Pengenalan Cara Kerja EMT

Selamat datang! Ini adalah pengantar pertama ke Euler Math Toolbox (disingkat EMT atau Euler). EMT adalah sistem terintegrasi yang merupakan perpaduan kernel numerik Euler dan program komputer aljabar Maxima.

- Bagian numerik, GUI, dan komunikasi dengan Maxima telah dikembangkan oleh R. Grothmann, seorang profesor matematika di Universitas Eichstätt, Jerman. Banyak algoritma numerik dan pustaka software open source yang digunakan di dalamnya.

- Maxima adalah program open source yang matang dan sangat kaya untuk perhitungan simbolik dan aritmatika tak terbatas. Software ini dikelola oleh sekelompok pengembang di internet.

- Beberapa program lain (LaTeX, Povray, Tiny C Compiler, Python) dapat digunakan di Euler untuk memungkinkan perhitungan yang lebih cepat maupun tampilan atau grafik yang lebih baik.

Yang sedang Anda baca (jika dibaca di EMT) ini adalah berkas notebook di EMT. Notebook aslinya bawaan EMT (dalam bahasa Inggris) dapat dibuka melalui menu File, kemudian pilih "Open Tutorias and Example", lalu pilih file "00 First Steps.en". Perhatikan, file notebook EMT memiliki ekstensi ".en". Melalui notebook ini Anda akan belajar menggunakan software Euler untuk menyelesaikan berbagai masalah matematika.

Panduan ini ditulis dengan Euler dalam bentuk notebook Euler, yang berisi teks (deskriptif), baris-baris perintah, tampilan hasil perintah (numerik, ekspresi matematika, atau gambar/plot), dan gambar yang disisipkan dari file gambar.

Untuk menambah jendela EMT, Anda dapat menekan [F11]. EMT akan menampilkan jendela grafik di layar desktop Anda. Tekan [F11] lagi untuk kembali ke tata letak favorit Anda. Tata letak disimpan untuk sesi berikutnya.

Anda juga dapat menggunakan [Ctrl]+[G] untuk menyembunyikan jendela grafik. Selanjutnya Anda dapat beralih antara grafik dan teks dengan tombol [TAB].

Seperti yang Anda baca, notebook ini berisi tulisan (teks) berwarna hijau, yang dapat Anda edit dengan mengklik kanan teks atau tekan menu Edit -> Edit Comment atau tekan [F5], dan juga baris perintah EMT yang ditandai dengan ">" dan berwarna merah. Anda dapat menyisipkan baris perintah baru dengan cara menekan tiga tombol bersamaan: [Shift]+[Ctrl]+[Enter].

Komentar (Teks Uraian)

Komentar atau teks penjelasan dapat berisi beberapa "markup" dengan sintaks sebagai berikut.

```
- * Judul
- ** Sub-Judul
- latex: F (x) = \int_a^x f (t) \, dt
- mathjax: \frac{x^2-1}{x-1} = x + 1
- maxima: 'integrate(x^3,x) = integrate(x^3,x) + C
- http://www.euler-math-toolbox.de
- See: http://www.google.de | Google
- image: hati.png
- ---
```

Hasil sintaks-sintaks di atas (tanpa diawali tanda strip) adalah sebagai berikut.

Judul

Sub-Judul

$$F(x) = \int_a^x f(t) \, dt$$

$$\frac{x^2-1}{x-1} = x + 1$$

maxima: 'integrate(x^3,x) = integrate(x^3,x) + C
http://www.euler-math-toolbox.de
See: http://www.google.de | Google
image: hati.png

Gambar diambil dari folder images di tempat file notebook berada dan tidak dapat dibaca dari Web. Untuk "See:", tautan (URL)web lokal dapat digunakan.

Paragraf terdiri atas satu baris panjang di editor. Pergantian baris akan memulai baris baru. Paragraf harus dipisahkan dengan baris kosong.

```
>// baris perintah diawali dengan >, komentar (keterangan) diawali dengan /
```

Baris Perintah

Mari kita tunjukkan cara menggunakan EMT sebagai kalkulator yang sangat canggih.

EMT berorientasi pada baris perintah. Anda dapat menuliskan satu atau lebih perintah dalam satu baris perintah. Setiap perintah harus diakhiri dengan koma atau titik koma.

- Titik koma menyembunyikan output (hasil) dari perintah.
- Sebuah koma mencetak hasilnya.
- Setelah perintah terakhir, koma diasumsikan secara otomatis (boleh tidak ditulis).

Dalam contoh berikut, kita mendefinisikan variabel r yang diberi nilai 1,25. Output dari definisi ini adalah nilai variabel. Tetapi karena tanda titik koma, nilai ini tidak ditampilkan. Pada kedua perintah di belakangnya, hasil kedua perhitungan tersebut ditampilkan.

```
>r=1.25; pi*r^2, 2*pi*r
```

```
4.90873852123
7.85398163397
```

Latihan untuk Anda

- Sisipkan beberapa baris perintah baru
- Tulis perintah-perintah baru untuk melakukan suatu perhitungan yang Anda inginkan, boleh menggunakan variabel, boleh tanpa variabel.

```
>a=2; b=5; a+b
```

```
7
```

```
>1+2+3+4+5+6
```

21

```
>(1+2+3+4+5+6)/3
```

7

```
>x=12; y=10; x^2+y^2
```

244

```
>sqrt(625)
```

25

```
>x=50; y=46; z=72; 3*x+5*y-2*z
```

236

```
>15250+7500+10750+5000+21500
```

60000

```
>(-12)*9+75+(-256)
```

-289

```
>76.5+14.44+159
```

249.94

```
>1.2^2+0.5^2+5^2
```

```
26.69
```

Beberapa catatan yang harus Anda perhatikan tentang penulisan sintaks perintah EMT.

- Pastikan untuk menggunakan titik desimal, bukan koma desimal untuk bilangan!
- Gunakan * untuk perkalian dan ^ untuk eksponen (pangkat).
- Seperti biasa, * dan / bersifat lebih kuat daripada + atau -.
- ^ mengikat lebih kuat dari *, sehingga pi * r ^ 2 merupakan rumus luas lingkaran.
- Jika perlu, Anda harus menambahkan tanda kurung, seperti pada 2 ^ (2 ^ 3).

Perintah r = 1.25 adalah menyimpan nilai ke variabel di EMT. Anda juga dapat menulis r: = 1.25 jika mau. Anda dapat menggunakan spasi sesuka Anda.

Anda juga dapat mengakhiri baris perintah dengan komentar yang diawali dengan dua garis miring (//).

```
>r := 1.25 // Komentar: Menggunakan  := sebagai ganti =
```

```
1.25
```

Argumen atau input untuk fungsi ditulis di dalam tanda kurung.

```
>sin(45°), cos(pi), log(sqrt(E))
```

```
0.707106781187
-1
0.5
```

Seperti yang Anda lihat, fungsi trigonometri bekerja dengan radian, dan derajat dapat diubah dengan °. Jika keyboard Anda tidak memiliki karakter derajat tekan [F7], atau gunakan fungsi deg() untuk mengonversi.

EMT menyediakan banyak sekali fungsi dan operator matematika.Hampir semua fungsi matematika sudah tersedia di EMT. Anda dapat melihat daftar lengkap fungsi-fungsi matematika di EMT pada berkas Referensi (klik menu Help -> Reference)

Untuk membuat rangkaian komputasi lebih mudah, Anda dapat merujuk ke hasil sebelumnya dengan "%". Cara ini sebaiknya hanya digunakan untuk merujuk hasil perhitungan dalam baris perintah yang sama.

```
>$solve(x^2-x-1=0) // Sebenarnya "=0" tidak perlu ditulis
```

$$\left[x=\frac{1-\sqrt{5}}{2}, x=\frac{\sqrt{5}+1}{2}\right]$$

```
>(sqrt(5)+1)/2, %^2-%+1 // Memeriksa solusi x^2-x-1=0
```

```
1.61803398875
2
```

Latihan untuk Anda

- Buka berkas Reference dan baca fungsi-fungsi matematika yang tersedia di EMT.
- Sisipkan beberapa baris perintah baru.
- Lakukan contoh-contoh perhitungan menggunakan fungsi-fungsi matematika di EMT.

```
>abs(20)
```

```
20
```

```
>exp(10)
```

```
22026.4657948
```

```
>log10(100)
```

```
2
```

```
>cos(0°)
```

```
1
```

```
>sin(90°)+cos(60°)+tan(30°)
```

```
2.07735026919
```

```
>deg(atan(1))
```

```
45
```

```
>bin(10,0:10)
```

```
[1,  10,  45,  120,  210,  252,  210,  120,  45,  10,  1]
```

```
>fac(10)/fac(7)
```

```
720
```

```
>mod(7,2)
```

```
1
```

```
>x=9.7; ceil(x), floor(x)
```

```
10
9
```

```
>round(1.867,1)
```

```
1.9
```

```
>logbase(16,2)
```

```
4
```

```
>sin(90°)/2, sec(60°), cosec(45°), tan(45°), cot(30°)
```

```
0.5
2
1.41421356237
1
1.73205080757
```

```
>gcd(12,24), lcm(12,24)
```

```
12
24
```

```
>primes(50)
```

```
[2,  3,  5,  7,  11,  13,  17,  19,  23,  29,  31,  37,  41,  43,  47]
```

```
>sum(isprime(1:50))
```

```
15
```

```
>sum(primes(50))
```

```
328
```

```
>fraction factor(24)
```

```
[2,  2,  2,  3]
```

Satuan

EMT dapat mengubah unit satuan menjadi sistem standar internasional (SI). Tambahkan satuan di belakang angka untuk konversi sederhana.

```
>1miles  // 1 mil = 1609,344 m
```

```
1609.344
```

Beberapa satuan yang sudah dikenal di dalam EMT adalah sebagai berikut. Semua unit diakhiri dengan tanda dolar ($), namun boleh tidak perlu ditulis dengan mengaktifkan easyunits.

kilometer$:=1000;
km$:=kilometer$;
cm$:=0.01;
mm$:=0.001;
minute$:=60;
min$:=minute$;
minutes$:=minute$;
hour$:=60*minute$;
h$:=hour$;
hours$:=hour$;
day$:=24*hour$;
days$:=day$;
d$:=day$;
year$:=365.2425*day$;
years$:=year$;
y$:=year$;
inch$:=0.0254;
in$:=inch$;
feet$:=12*inch$;
foot$:=feet$;
ft$:=feet$;
yard$:=3*feet$;
yards$:=yard$;
yd$:=yard$;
mile$:=1760*yard$;
miles$:=mile$;
kg$:=1;
sec$:=1;
ha$:=10000;
Ar$:=100;
Tagwerk$:=3408;
Acre$:=4046.8564224;
pt$:=0.376mm;

Untuk konversi ke dan antar unit, EMT menggunakan operator khusus, yakni ->.

```
>4km -> miles, 4inch -> " mm"
```

```
2.48548476895
101.6 mm
```

Format Tampilan Nilai

Akurasi internal untuk nilai bilangan di EMT adalah standar IEEE, sekitar 16 digit desimal. Aslinya, EMT tidak mencetak semua digit suatu bilangan. Ini untuk menghemat tempat dan agar terlihat lebih baik. Untuk mengatrtamilan satu bilangan, operator berikut dapat digunakan.

```
>pi
```

```
3.14159265359
```

```
>longest pi
```

```
      3.141592653589793
```

```
>long pi
```

```
3.14159265359
```

```
>short pi
```

```
3.1416
```

```
>shortest pi
```

```
   3.1
```

```
>fraction pi
```

```
312689/99532
```

```
>short 1200*1.03^10, long E, longest pi
```

```
1612.7
2.71828182846
      3.141592653589793
```

Format aslinya untuk menampilkan nilai menggunakan sekitar 10 digit. Format tampilan nilai dapat diatur secara global atau hanya untuk satu nilai.

Anda dapat mengganti format tampilan bilangan untuk semua perintah selanjutnya. Untuk mengembalikan ke format aslinya dapat digunakan perintah "defformat" atau "reset".

```
>longestformat; pi, defformat; pi
```

```
3.141592653589793
3.14159265359
```

Kernel numerik EMT bekerja dengan bilangan titik mengambang (floating point) dalam presisi ganda IEEE (berbeda dengan bagian simbolik EMT). Hasil numerik dapat ditampilkan dalam bentuk pecahan.

```
>1/7+1/4, fraction %
```

```
0.392857142857
11/28
```

Perintah Multibaris

Perintah multi-baris membentang di beberapa baris yang terhubung dengan "..." di setiap akhir baris, kecuali baris terakhir. Untuk menghasilkan tanda pindah baris tersebut, gunakan tombol [Ctrl]+[Enter]. Ini akan menyambung perintah ke baris berikutnya dan menambahkan "..." di akhir baris sebelumnya. Untuk menggabungkan suatu baris ke baris sebelumnya, gunakan [Ctrl]+[Backspace].

Contoh perintah multi-baris berikut dapat dijalankan setiap kali kursor berada di salah satu barisnya. Ini juga menunjukkan bahwa ... harus berada di akhir suatu baris meskipun baris tersebut memuat komentar.

```
>a=4; b=15; c=2; // menyelesaikan a*x^2+b*x+c=0 secara manual ...
>D=sqrt(b^2/(a^2*4)-c/a); ...
>-b/(2*a) + D, ...
>-b/(2*a) - D
```

```
-0.138444501319
-3.61155549868
```

Menampilkan Daftar Variabe

Untuk menampilkan semua variabel yang sudah pernah Anda definisikan sebelumnya (dan dapat dilihat kembali nilainya), gunakan perintah "listvar".

```
>listvar
```

```
r                    1.25
a                    4
b                    15
c                    2
D                    1.73655549868123
```

Perintah listvar hanya menampilkan variabel buatan pengguna. Dimungkinkan untuk menampilkan variabel lain, dengan menambahkan string termuat di dalam nama variabel yang diinginkan.

Perlu Anda perhatikan, bahwa EMT membedakan huruf besar dan huruf kecil. Jadi variabel "d" berbeda dengan variabel "D".

Contoh berikut ini menampilkan semua unit yang diakhiri dengan "m" dengan mencari semua variabel yang berisi "m$".

```
>listvar m$
```

```
km$                  1000
cm$                  0.01
mm$                  0.001
nm$                  1853.24496
gram$                0.001
m$                   1
hquantum$            6.62606957e-34
atm$                 101325
```

Untuk menghapus variabel tanpa harus memulai ulang EMT gunakan perintah "remvalue".

```
>remvalue a,b,c,D
>D
```

```
Variable D not found!
Error in:
D ...
 ^
```

Menampilkan Panduan

Untuk mendapatkan panduan tentang penggunaan perintah atau fungsi di EMT, buka jendela panduan dengan menekan [F1] dan cari fungsinya. Anda juga dapat mengklik dua kali pada fungsi yang tertulis di baris perintah atau di teks untuk membuka jendela panduan.

Coba klik dua kali pada perintah "intrandom" berikut ini!

```
>intrandom(10,6)
```

```
[4,  2,  6,  2,  4,  2,  3,  2,  2,  6]
```

Di jendela panduan, Anda dapat mengklik kata apa saja untuk menemukan referensi atau fungsi.

Misalnya, coba klik kata "random" di jendela panduan. Kata tersebut boleh ada dalam teks atau di bagian "See:" pada panduan. Anda akan menemukan penjelasan fungsi "random", untuk menghasilkan bilangan acak berdistribusi uniform antara 0,0 dan 1,0. Dari panduan untuk "random" Anda dapat menampilkan panduan untuk fungsi "normal", dll.

```
>random(10)
```

```
[0.270906,  0.704419,  0.217693,  0.445363,  0.308411,  0.914541,  0.1935
0.463387,  0.095153,  0.595017]
```

```
>normal(10)
```

```
[-0.495418,  1.6463,  -0.390056,  -1.98151,  3.44132,  0.308178,  -0.7334
-0.526167,  1.10018,  0.108453]
```

Matriks dan Vektor

EMT merupakan suatu aplikasi matematika yang mengerti "bahasa matriks". Artinya, EMT menggunakan vektor dan matriks untuk perhitungan-perhitungan tingkat lanjut. Suatu vektor atau matriks dapat didefinisikan dengan tanda kurung siku. Elemen-elemennya dituliskan di dalam tanda kurung siku, antar elemen dalam satu baris dipisahkan oleh koma(,), antar baris dipisahkan oleh titik koma (;).

Vektor dan matriks dapat diberi nama seperti variabel biasa.

```
>v=[4,5,6,3,2,1]
```

```
[4,  5,  6,  3,  2,  1]
```

```
>A=[1,2,3;4,5,6;7,8,9]
```

```
           1              2              3
           4              5              6
           7              8              9
```

Karena EMT mengerti bahasa matriks, EMT memiliki kemampuan yang sangat canggih untuk melakukan perhitungan matematis untuk masalah-masalah aljabar linier, statistika, dan optimisasi.

Vektor juga dapat didefinisikan dengan menggunakan rentang nilai dengan interval tertentu menggunakan tanda titik dua (:),seperti contoh berikut ini.

```
>c=1:5
```

```
[1,  2,  3,  4,  5]
```

```
>w=0:0.1:1
```

```
[0,  0.1,  0.2,  0.3,  0.4,  0.5,  0.6,  0.7,  0.8,  0.9,  1]
```

```
>mean(w^2)
```

```
0.35
```

Bilangan Kompleks

EMT juga dapat menggunakan bilangan kompleks. Tersedia banyak fungsi untuk bilangan kompleks di EMT. Bilangan imaginer

$$i = \sqrt{-1}$$

dituliskan dengan huruf I (huruf besar I), namun akan ditampilkan dengan huruf i (i kecil).

```
re(x) : bagian riil pada bilangan kompleks x.
im(x) : bagian imaginer pada bilangan kompleks x.
complex(x) : mengubah bilangan riil x menjadi bilangan kompleks.
conj(x) : Konjugat untuk bilangan bilangan komplkes x.
arg(x) : argumen (sudut dalam radian) bilangan kompleks x.
real(x) : mengubah x menjadi bilangan riil.
```

Apabila bagian imaginer x terlalu besar, hasilnya akan menampilkan pesan kesalahan.

```
>sqrt(-1) // Error!
>sqrt(complex(-1))
```

```
>z=2+3*I, re(z), im(z), conj(z), arg(z), deg(arg(z)), deg(arctan(3/2))
```

```
2+3i
2
3
2-3i
0.982793723247
56.309932474
56.309932474
```

```
>deg(arg(I)) // 90°
```

```
90
```

```
>sqrt(-1)
```

```
Floating point error!
Error in sqrt
Error in:
sqrt(-1) ...
        ^
```

```
>sqrt(complex(-1))
```

```
0+1i
```

EMT selalu menganggap semua hasil perhitungan berupa bilangan riil dan tidak akan secara otomatis mengubah ke bilangan kompleks.

Jadi akar kuadrat -1 akan menghasilkan kesalahan, tetapi akar kuadrat kompleks didefinisikan untuk bidang koordinat dengan cara seperti biasa. Untuk mengubah bilangan riil menjadi kompleks, Anda dapat menambahkan 0i atau menggunakan fungsi "complex".

```
>complex(-1), sqrt(%)
```

```
-1+0i
0+1i
```

Matematika Simbolik

EMT dapat melakukan perhitungan matematika simbolis (eksak) dengan bantuan software Maxima. Software Maxima otomatis sudah terpasang di komputer Anda ketika Anda memasang EMT. Meskipun demikian, Anda dapat juga memasang software Maxima tersendiri (yang terpisah dengan instalasi Maxima di EMT).

Pengguna Maxima yang sudah mahir harus memperhatikan bahwa terdapat sedikit perbedaan dalam sintaks antara sintaks asli Maxima dan sintaks ekspresi simbolik di EMT.

Untuk melakukan perhitungan matematika simbolis di EMT, awali perintah Maxima dengan tanda "&". Setiap ekspresi yang dimulai dengan "&" adalah ekspresi simbolis dan dikerjakan oleh Maxima.

```
>&(a+b)^2
```

```
                                    2
                             (b + a)
```

```
>&expand((a+b)^2), &factor(x^2+5*x+6)
```

```
                          2            2
                         b  + 2 a b + a

                         (x + 2) (x + 3)
```

```
>&solve(a*x^2+b*x+c,x) // rumus abc
```

```
                    2                            2
        (- sqrt(b  - 4 a c)) - b         sqrt(b  - 4 a c) - b
   [x = ------------------------, x = --------------------]
                  2 a                           2 a
```

```
>&(a^2-b^2)/(a+b), &ratsimp(%) // ratsimp menyederhanakan bentuk pecahan
```

```
 2    2
a  - b
-------
 b + a
```

```
 a - b
```

```
>10! // nilai faktorial (modus EMT)
```

```
  3628800
```

```
>&10! //nilai faktorial (simbolik dengan Maxima)
```

```
3628800
```

Untuk menggunakan perintah Maxima secara langsung (seperti perintah pada layar Maxima) awali perintahnya dengan tanda "::" pada baris perintah EMT. Sintaks Maxima disesuaikan dengan sintaks EMT (disebut "modus kompatibilitas").

```
>factor(1000) // mencari semua faktor 1000 (EMT)
```

```
  [2,  2,  2,  5,  5,  5]
```

```
>:: factor(1000) // faktorisasi prima 1000 (dengan Maxima)
```

```
  3  3
 2  5
```

```
>:: factor(20!)
```

```
 18  8  4  2
2   3  5  7  11 13 17 19
```

Jika Anda sudah mahir menggunakan Maxima, Anda dapat menggunakan sintaks asli perintah Maxima dengan menggunakan tanda ":::" untuk mengawali setiap perintah Maxima di EMT. Perhatikan, harus ada spasi antara ":::" dan perintahnya.

```
>::: binomial(5,2); // nilai C(5,2)
```

```
                                   10
```

```
>::: binomial(m,4); // C(m,4)=m!/(4!(m-4)!)
```

```
                        (m - 3) (m - 2) (m - 1) m
                        -------------------------
                                   24
```

```
>::: trigexpand(cos(x+y)); // rumus cos(x+y)=cos(x) cos(y)-sin(x)sin(y)
```

```
                      cos(x) cos(y) - sin(x) sin(y)
```

```
>::: trigexpand(sin(x+y));
```

```
                      cos(x) sin(y) + sin(x) cos(y)
```

```
>::: trigsimp(((1-sin(x)^2)*cos(x))/cos(x)^2+tan(x)*sec(x)^2) //menyederhan
```

```
                                     4
                            sin(x) + cos (x)
                            ----------------
                                   3
                                cos (x)
```

Untuk menyimpan ekspresi simbolik ke dalam suatu variabel digunakan tanda "&=".

```
>p1 &= (x^3+1)/(x+1)
```

```
                                     3
                                    x  + 1
                                    ------
                                    x + 1
```

```
>&ratsimp(p1)
```

```
                                   2
                                  x  - x + 1
```

Untuk mensubstitusikan suatu nilai ke dalam variabel dapat digunakan perintah "with".

```
>&p1 with x=3 // (3^3+1)/(3+1)
```

```
                                      7
```

```
>&p1 with x=a+b, &ratsimp(%) //substitusi dengan variabel baru
```

```
                                        3
                                 (b + a)  + 1
                                 ------------
                                  b + a + 1

                             2                  2
                            b  + (2 a - 1) b + a  - a + 1
```

```
>&diff(p1,x) //turunan p1 terhadap x
```

```
                    2       3
                 3 x       x  + 1
                 ----- - --------
                 x + 1          2
                         (x + 1)
```

```
>&integrate(p1,x) // integral p1 terhadap x
```

```
                    3      2
                 2 x  - 3 x  + 6 x
                 -----------------
                         6
```

Tampilan Matematika Simbolik dengan LaTeX

Anda dapat menampilkan hasil perhitunagn simbolik secara lebih bagus menggunakan LaTeX. Untuk melakukan hal ini, tambahkan tanda dolar ($) di depan tanda & pada setiap perintah Maxima.
Perhatikan, hal ini hanya dapat menghasilkan tampilan yang diinginkan apabila komputer Anda sudah terpasang software LaTeX.

```
>$&(a+b)^2
>$&expand((a+b)^2), $&factor(x^2+5*x+6)
>$&solve(a*x^2+b*x+c,x) // rumus abc
>$&(a^2-b^2)/(a+b), $&ratsimp(%)
```

Selamat Belajar dan Berlatih!

Baik, itulah sekilas pengantar penggunaan software EMT. Masih banyak kemampuan EMT yang akan Anda pelajari dan praktikkan.

Sebagai latihan untuk memperlancar penggunaan perintah-perintah EMT yang sudah dijelaskan di atas, silakan Anda lakukan hal-hal sebagai berikut.

- Carilah soal-soal matematika dari buku-buku Matematika.
- Tambahkan beberapa baris perintah EMT pada notebook ini.
- Selesaikan soal-soal matematika tersebut dengan menggunakan EMT.

Pilih soal-soal yang sesuai dengan perintah-perintah yang sudah dijelaskan dan dicontohkan di atas.

```
>(-1765)+6786+(-540)/6-234
```

```
4697
```

```
>p=-12; q=9; r=24; 4*p*q-r
```

```
-456
```

```
>&sin(45°)+cos(90°)+sec(30°)
```

```
                      2          1
                   ------- + -------
                   sqrt(3)   sqrt(2)
```

```
>a=24; t=10; (a*t)/2 // luas segitiga
```

```
120
```

```
>a=1; b=(-3); c=(-10); ...
>D=sqrt(b^2/(a^2*4)-c/a); ...
>-b/(2*a) + D, ...
>-b/(2*a) - D
```

```
5
-2
```

```
>&2/5+13/10-6/3
```

```
                                    3
                                  - --
                                    10
```

```
>&integrate(x^2,x,0,2)
```

```
8
-
3
```

```
>&integrate(x^2+x,x,1,5)
```

```
160
---
 3
```

```
>&solve([a+2*b=8,a-3*b=(-7)],[a,b])
```

```
[[a = 2, b = 3]]
```

```
>A=[1,4,1;6,3,2;3,5,7]; 3*A
```

```
 3    12     3
18     9     6
 9    15    21
```

```
>p&=4*x^2+12*x-8
```

```
     2
4 x   + 12 x - 8
```

```
>&diff(p,x)
```

```
8 x + 12
```

BAB 2

MENGGUNAKAN EMT UNTUK MENYELESAIKAN MASALAH ALJABAR

EMT untuk Perhitungan Aljabar

Pada notebook ini Anda belajar menggunakan EMT untuk melakukan berbagai perhitungan terkait dengan materi atau topik dalam Aljabar. Kegiatan yang harus Anda lakukan adalah sebagai berikut:

- Membaca secara cermat dan teliti notebook ini;
- Menerjemahkan teks bahasa Inggris ke bahasa Indonesia;
- Mencoba contoh-contoh perhitungan (perintah EMT) dengan cara meng-ENTER setiap perintah EMT yang ada (pindahkan kursor ke baris perintah)
- Jika perlu Anda dapat memodifikasi perintah yang ada dan memberikan keterangan/penjelasan tambahan terkait hasilnya.
- Menyisipkan baris-baris perintah baru untuk mengerjakan soal-soal Aljabar dari file PDF yang saya berikan;
- Memberi catatan hasilnya.
- Jika perlu tuliskan soalnya pada teks notebook (menggunakan format LaTeX).
- Gunakan tampilan hasil semua perhitungan yang eksak atau simbolik dengan format LaTeX. (Seperti contoh-contoh pada notebook ini.)

Contoh pertama

Menyederhanakan bentuk aljabar:

$$6x^{-3}y^5 \times -7x^2y^{-9}$$

```
>$&6*x^(-3)*y^5*-7*x^2*y^(-9)
```

$$-\frac{42}{x\,y^4}$$

Menjabarkan:

$$(6x^{-3}+y^5)(-7x^2-y^{-9})$$

```
>$&showev('expand((6*x^(-3)+y^5)*(-7*x^2-y^(-9))))
```

$$expand\left(\left(-\frac{1}{y^9}-7\,x^2\right)\,\left(y^5+\frac{6}{x^3}\right)\right)=-7\,x^2\,y^5-\frac{1}{y^4}-\frac{6}{x^3\,y^9}-\frac{42}{x}$$

```
>&powerdisp: true
```

```
true
```

```
>$(1-x)^5|expand
```

$$1-5\,x+10\,x^2-10\,x^3+5\,x^4-x^5$$

```
>$expand((a+b)^3)
```

$$a^3+3\,a^2\,b+3\,a\,b^2+b^3$$

Baris perintah Euler terdiri dari satu atau beberapa perintah Euler diikuti dengan titik koma ";" atau koma ",". Titik koma mencegah pencetakan hasil. Koma setelah perintah terakhir dapat dihilangkan.

Baris perintah berikut hanya akan mencetak hasil ekspresi, bukan tugas atau perintah format.

```
>r:=2; h:=4; pi*r^2*h/3
```

```
16.7551608191
```

Perintah harus dipisahkan dengan tanda kosong. Baris perintah berikut mencetak dua hasilnya.

```
>pi*2*r*h, %+2*pi*r*h // Ingat tanda % menyatakan hasil perhitungan terakhi
```

```
50.2654824574
100.530964915
```

Baris perintah dieksekusi dalam urutan yang ditekan pengguna kembali. Jadi, Anda mendapatkan nilai baru setiap kali menjalankan baris kedua.

```
>x := 1;
>x := cos(x) // nilai cosinus (x dalam radian)
```

```
0.540302305868
```

```
>x := cos(x)
```

```
0.857553215846
```

Jika dua jalur dihubungkan dengan "..." kedua jalur akan selalu dijalankan secara bersamaan.

```
>x := 1.5; ...
>x := (x+2/x)/2, x := (x+2/x)/2, x := (x+2/x)/2,
```

```
1.41666666667
1.41421568627
1.41421356237
```

Ini juga cara yang baik untuk menyebarkan perintah panjang ke dua baris atau lebih. Anda dapat menekan Ctrl + Return untuk membagi garis menjadi dua pada posisi kursor saat ini, atau Ctlr + Back untuk menggabungkan garis.

Untuk melipat semua multi-garis tekan Ctrl + L. Kemudian garis berikutnya hanya akan terlihat, jika salah satunya memiliki fokus. Untuk melipat satu garis banyak, mulailah baris pertama dengan "% +".

```
>%+ x=4+5; ...
```

Garis yang dimulai dengan %% akan benar-benar tidak terlihat.

```
81
```

Euler mendukung loop dalam baris perintah, asalkan sesuai dengan satu baris atau beberapa baris. Dalam program, batasan ini tidak berlaku, tentu saja. Untuk informasi lebih lanjut, lihat pengantar berikut.

```
>x=1; for i=1 to 5; x := (x+2/x)/2, end; // menghitung akar 2
```

```
1.5
1.41666666667
1.41421568627
1.41421356237
1.41421356237
```

Tidak apa-apa menggunakan multi-garis. Pastikan baris diakhiri dengan "...".

```
>x := 1.5; // komentar diletakkan di sini sebelum ...
>ulang xnew:=(x+2/x)/2; until xnew~=x; ...
>   x := xnew; ...
>end; ...
>x,
```

```
Variable ulang not found!
Error in:
x := 1.5; ulang xnew:=(x+2/x)/2; until xnew~=x;    x := xnew;  ...
                ^
```

Struktur bersyarat juga berfungsi.

```
>if E^pi>pi^E; then "Berpikir begitu!", berakhir jika;
```

```
Berpikir begitu!
Variable berakhir not found!
Error in:
if E^pi>pi^E; then "Berpikir begitu!", berakhir jika; ...
                                                ^
```

Saat Anda menjalankan perintah, kursor dapat berada di posisi mana pun di baris perintah. Anda dapat kembali ke perintah sebelumnya atau melompat ke perintah berikutnya dengan tombol panah. Atau Anda dapat mengklik bagian komentar di atas perintah untuk membuka perintah.

Saat Anda menggerakkan kursor di sepanjang garis, pasangan buka dan tutup tanda kurung atau tanda kurung akan disorot. Juga, perhatikan baris statusnya. Setelah kurung buka dari fungsi sqrt (), baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan tombol kembali.

```
>sqrt(sin(10°)/cos(20°))
```

```
0.429875017772
```

Untuk melihat bantuan untuk perintah terbaru, buka jendela bantuan dengan F1. Di sana, Anda dapat memasukkan teks untuk dicari. Pada baris kosong, bantuan untuk jendela bantuan akan ditampilkan. Anda dapat menekan escape untuk menghapus garis, atau untuk menutup jendela bantuan.

Anda dapat mengklik dua kali pada perintah apa pun untuk membuka bantuan untuk perintah ini. Coba klik dua kali perintah exp di bawah ini di baris perintah.

```
>exp(log(2.5))
```

```
2.5
```

Anda juga dapat menyalin dan menempel di Euler. Gunakan Ctrl-C dan Ctrl-V untuk ini. Untuk menandai teks, seret mouse atau gunakan shift bersama dengan tombol kursor apa pun. Selain itu, Anda dapat menyalin tanda kurung yang disorot.

Sintaks Dasar

Euler mengetahui fungsi matematika biasa. Seperti yang Anda lihat di atas, fungsi trigonometri bekerja dalam radian atau derajat. Untuk mengonversi menjadi derajat, tambahkan simbol derajat (dengan tombol F7) ke nilainya, atau gunakan fungsi rad (x). Fungsi akar kuadrat disebut akar kuadrat di Euler. Tentu saja, x ^ (1/2) juga dimungkinkan.

Untuk menyetel variabel, gunakan "=" atau ": =". Demi kejelasan, pendahuluan ini menggunakan bentuk yang terakhir. Spasi tidak penting. Tapi ruang antar perintah diharapkan.

Beberapa perintah dalam satu baris dipisahkan dengan "," atau ";". Titik koma menekan keluaran dari perintah. Di akhir baris perintah, "," diasumsikan, jika ";" hilang.

```
>g:=9.81; t:=2.5; 1/2*g*t^2
```

```
30.65625
```

EMT menggunakan sintaks pemrograman untuk ekspresi. Memasuki

getah: e ^ 2 \ cdot \ left (\ frac {1} {3 + 4 \ log (0.6)} + \ frac {1} {7} \ kanan)

Anda harus mengatur tanda kurung yang benar dan menggunakan / untuk pecahan. Perhatikan tanda kurung yang disorot untuk bantuan. Perhatikan bahwa konstanta Euler e dinamai E dalam EMT.

```
>E^2*(1/(3+4*log(0.6))+1/7)
```

```
8.77908249441
```

Untuk menghitung ekspresi yang rumit seperti

lateks: \ kiri (\ frac {\ frac17 + \ frac18 + 2} {\ frac13 + \ frac12} \ kanan) ^ 2 \ pi

Anda harus memasukkannya dalam bentuk baris.

```
>((1/7 + 1/8 + 2) / (1/3 + 1/2))^2 * pi
```

```
23.2671801626
```

Tempatkan tanda kurung dengan hati-hati di sekitar sub-ekspresi yang perlu dihitung terlebih dahulu. EMT membantu Anda dengan menyoroti ekspresi bahwa kurung tutup selesai. Anda juga harus memasukkan nama "pi" untuk huruf Yunani pi.

Hasil dari perhitungan ini adalah bilangan floating point. Ini secara default dicetak dengan akurasi sekitar 12 digit. Di baris perintah berikut, kita juga belajar bagaimana kita bisa merujuk ke hasil sebelumnya dalam baris yang sama.

```
>1/3+1/7, fraction %
```

```
0.47619047619
10/21
```

Perintah Euler bisa berupa ekspresi atau perintah primitif. Ekspresi dibuat dari operator dan fungsi. Jika perlu, itu harus berisi tanda kurung untuk memaksa urutan eksekusi yang benar. Jika ragu, menetapkan braket adalah ide yang bagus. Perhatikan bahwa EMT menampilkan tanda kurung buka dan tutup saat mengedit baris perintah.

```
>(cos(pi/4)+1)^3*(sin(pi/4)+1)^2
```

```
14.4978445072
```

Operator numerik Euler termasuk

```
+ unary atau operator plus
- unary atau operator minus
*, /
. produk matriks
a ^ b pangkat untuk positif a atau bilangan bulat b (a ** b bekerja
```

juga)

```
n! operator faktorial
```

dan masih banyak lagi.
Berikut beberapa fungsi yang mungkin Anda perlukan. Masih banyak lagi.

```
sin, cos, tan, atan, asin, acos, rad, deg
log, exp, log10, sqrt, logbase
bin, logbin, logfac, mod, floor, ceil, round, abs, sign
konj, re, im, arg, konj, nyata, kompleks
beta, betai, gamma, complexgamma, ellrf, ellf, ellrd, elle
bitand, bitor, bitxor, bitnot
```

Beberapa perintah memiliki alias, mis. ln untuk log.

```
>ln(E^2), arctan(tan(0.5))
```

```
2
0.5
```

```
>sin(30°)
```

```
0.5
```

Pastikan untuk menggunakan tanda kurung (tanda kurung bulat), setiap kali ada keraguan tentang urutan eksekusi! Berikut ini tidak sama dengan (2 ^ 3) ^ 4, yang merupakan default untuk 2 ^ 3 ^ 4 di EMT (beberapa sistem numerik melakukannya dengan cara lain).

```
>2^3^4, (2^3)^4, 2^(3^4)
```

```
2.41785163923e+24
4096
2.41785163923e+24
```

Bilangan Nyata

Tipe data utama di Euler adalah bilangan real. Real direpresentasikan dalam format IEEE dengan akurasi sekitar 16 digit desimal.

```
>longest 1/3
```

```
      0.3333333333333333
```

Representasi ganda internal membutuhkan 8 byte.

```
>printdual(1/3)
```

```
1.0101010101010101010101010101010101010101010101010101*2^-2
```

```
>printhex(1/3)
```

```
5.5555555555554*16^-1
```

String

Sebuah string di Euler didefinisikan dengan "...".

```
>"Sebuah string bisa berisi apa saja."
```

```
Sebuah string bisa berisi apa saja.
```

String bisa digabungkan dengan | atau dengan +. Ini juga berfungsi dengan angka, yang diubah menjadi string dalam kasus itu.

```
>"The area of the circle with radius " + 2 + " cm is " + pi*4 + " cm^2."
```

```
The area of the circle with radius 2 cm is 12.5663706144 cm^2.
```

Fungsi cetak juga mengubah angka menjadi string. Ini bisa mengambil sejumlah digit dan sejumlah tempat (0 untuk output padat), dan secara optimal satu unit.

```
>"Golden Ratio : " + print((1+sqrt(5))/2,5,0)
```

```
Golden Ratio : 1.61803
```

Tidak ada string khusus, yang tidak dicetak. Itu dikembalikan oleh beberapa fungsi, ketika hasilnya tidak penting. (Ini dikembalikan secara otomatis, jika fungsi tidak memiliki pernyataan pengembalian.)

```
>none
```

Untuk mengonversi string menjadi angka, cukup evaluasi. Ini berfungsi untuk ekspresi juga (lihat di bawah).

```
>"1234.5"()
```

```
1234.5
```

Untuk mendefinisikan vektor string, gunakan notasi vektor [...].

```
>v:=["affe","charlie","bravo"]
```

```
affe
charlie
bravo
```

Vektor string kosong dilambangkan dengan [tidak ada]. Vektor string dapat digabungkan.

```
>w:=[none]; w|v|v
```

```
affe
charlie
bravo
affe
charlie
bravo
```

String dapat berisi karakter Unicode. Secara internal, string ini berisi kode UTF-8. Untuk menghasilkan string seperti itu, gunakan u "..." dan salah satu entitas HTML.

String unicode dapat digabungkan seperti string lainnya.

```
>u"&alpha; = " + 45 + u"&deg;" // pdfLaTeX mungkin gagal menampilkan secara
```

```
 = 45°
```

I

Di komentar, entitas yang sama seperti & alpha ;, & beta; dll. dapat digunakan. Ini mungkin alternatif cepat untuk Latex. (Lebih detail tentang komentar di bawah).

Ada beberapa fungsi untuk membuat atau menganalisis string unicode. Fungsi strtochar () akan mengenali string Unicode, dan menerjemahkannya dengan benar.

```
>v=strtochar(u"&Auml; is a German letter")
```

```
[196,  32,  105,  115,  32,  97,  32,  71,  101,  114,  109,  97,  110,
32,  108,  101,  116,  116,  101,  114]
```

Hasilnya adalah vektor bilangan Unicode. Fungsi kebalikannya adalah chartoutf ().

```
>v[1]=strtochar(u"&Uuml;")[1]; chartoutf(v)
```

```
Ü is a German letter
```

Fungsi utf () dapat menerjemahkan string dengan entitas dalam variabel menjadi string Unicode.

```
>s="We have &alpha;=&beta;."; utf(s) // pdfLaTeX mungkin gagal menampilkan
```

```
We have =.
```

Dimungkinkan juga untuk menggunakan entitas numerik.

```
>u"&#196;hnliches"
```

```
Ähnliches
```

Nilai Boolean

Nilai Boolean diwakili dengan 1 = true atau 0 = false di Euler. String dapat dibandingkan, seperti halnya angka.

```
>2<1, "apel"<"banana"
```

```
0
1
```

"dan" adalah operator "&&" dan "atau" adalah operator "||", seperti dalam bahasa C. (Kata "dan" dan "atau" hanya dapat digunakan dalam kondisi untuk "jika".)

```
>2<E && E<3
```

```
1
```

Operator Boolean mematuhi aturan bahasa matriks.

```
>(1:10)>5, nonzeros(%)
```

```
[0,  0,  0,  0,  0,  1,  1,  1,  1,  1]
[6,  7,  8,  9,  10]
```

Anda dapat menggunakan fungsi nonzeros () untuk mengekstrak elemen tertentu dari vektor. Dalam contoh, kami menggunakan isprime bersyarat (n).

```
>N=2|3:2:99 // N berisi elemen 2 dan bilangan2 ganjil dari 3 s.d. 99
```

```
[2,  3,  5,  7,  9,  11,  13,  15,  17,  19,  21,  23,  25,  27,  29,
31,  33,  35,  37,  39,  41,  43,  45,  47,  49,  51,  53,  55,  57,
59,  61,  63,  65,  67,  69,  71,  73,  75,  77,  79,  81,  83,  85,
87,  89,  91,  93,  95,  97,  99]
```

```
>N[nonzeros(isprime(N))] //pilih anggota2 N yang prima
```

```
[2,  3,  5,  7,  11,  13,  17,  19,  23,  29,  31,  37,  41,  43,  47,
53,  59,  61,  67,  71,  73,  79,  83,  89,  97]
```

Format Keluaran

Format keluaran default EMT mencetak 12 digit. Untuk memastikan bahwa kami melihat default, kami mengatur ulang format.

```
>defformat; pi
```

```
3.14159265359
```

Secara internal, EMT menggunakan standar IEEE untuk bilangan ganda dengan sekitar 16 digit desimal. Untuk melihat jumlah digit secara lengkap, gunakan perintah "longestformat", atau gunakan operator "longest" untuk menampilkan hasil dalam format terpanjang.

```
>longest pi
```

```
      3.141592653589793
```

Berikut adalah representasi heksadesimal internal dari bilangan ganda.

```
>printhex(pi)
```

```
3.243F6A8885A30*16^0
```

Format keluaran dapat diubah secara permanen dengan perintah format.

```
>format(12,5); 1/3, pi, sin(1)
```

```
    0.33333
    3.14159
    0.84147
```

Standarnya adalah format (12).

```
>format(12); 1/3
```

```
0.333333333333
```

Fungsi seperti "shortestformat", "shortformat", "longformat" bekerja untuk vektor dengan cara berikut.

```
>shortestformat; random(3,8)
```

```
    0.66     0.2    0.89    0.28    0.53    0.31    0.44     0.3
    0.28    0.88    0.27     0.7    0.22    0.45    0.31    0.91
    0.19    0.46   0.095     0.6    0.43    0.73    0.47    0.32
```

Format default untuk skalar adalah format (12). Tapi ini bisa diubah.

```
>setscalarformat(5); pi
```

```
3.1416
```

Fungsi "format terpanjang" mengatur format skalar juga.

```
>longestformat; pi
```

```
3.141592653589793
```

Sebagai referensi, berikut adalah daftar format keluaran terpenting.

```
format terpendek format pendek format panjang, format terpanjang
format (panjang, digit) format yang baik (panjang)
fracformat (panjang)
defformat
```

Akurasi internal EMT adalah sekitar 16 tempat desimal, yang merupakan standar IEEE. Angka disimpan dalam format internal ini.

Tetapi format keluaran EMT dapat diatur dengan cara yang fleksibel.

```
>longestformat; pi,
```

```
3.141592653589793
```

```
>format(10,5); pi
```

```
3.14159
```

Standarnya adalah defformat ().

```
>defformat; // default
```

Ada operator pendek yang hanya mencetak satu nilai. Operator "terpanjang" akan mencetak semua digit nomor yang valid.

```
>longest pi^2/2
```

```
      4.934802200544679
```

Ada juga operator singkat untuk mencetak hasil dalam format pecahan. Kami telah menggunakannya di atas.

```
>fraction 1+1/2+1/3+1/4
```

```
25/12
```

Karena format internal menggunakan cara biner untuk menyimpan angka, nilai 0.1 tidak akan direpresentasikan dengan tepat. Kesalahan bertambah sedikit, seperti yang Anda lihat dalam perhitungan berikut.

```
>longest 0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1-1
```

```
 -1.110223024625157e-16
```

Tetapi dengan "longformat" default Anda tidak akan melihat ini. Untuk kenyamanan, keluaran angka yang sangat kecil adalah 0.

```
>0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1-1
```

```
0
```

Ekspresi

String atau nama dapat digunakan untuk menyimpan ekspresi matematika, yang dapat dievaluasi oleh EMT. Untuk ini, gunakan tanda kurung setelah ekspresi. Jika Anda bermaksud menggunakan string sebagai ekspresi, gunakan konvensi untuk menamainya "fx" atau "fxy" dll. Ekspresi lebih diutamakan daripada fungsi.

Variabel global dapat digunakan dalam evaluasi.

```
>r:=2; fx:="pi*r^2"; longest fx()
```

```
      12.56637061435917
```

Parameter ditetapkan ke x, y, dan z dalam urutan itu. Parameter tambahan dapat ditambahkan menggunakan parameter yang ditetapkan.

```
>fx:="a*sin(x)^2"; fx(5,a=-1)
```

```
-0.919535764538
```

Perhatikan bahwa ekspresi akan selalu menggunakan variabel global, meskipun ada variabel dalam fungsi dengan nama yang sama. (Jika tidak, evaluasi ekspresi dalam fungsi dapat memiliki hasil yang sangat membingungkan bagi pengguna yang memanggil fungsi tersebut.)

```
>at:=4; function f(expr,x,at) := expr(x); ...
>f("at*x^2",3,5) // computes 4*3^2 not 5*3^2
```

```
36
```

Jika Anda ingin menggunakan nilai lain untuk "at" daripada nilai global, Anda perlu menambahkan "at = value".

```
>at:=4; function f(expr,x,a) := expr(x,at=a); ...
>f("at*x^2",3,5)
```

```
45
```

Sebagai referensi, kami berkomentar bahwa koleksi panggilan (dibahas di tempat lain) dapat berisi ekspresi. Jadi contoh diatas bisa kita buat sebagai berikut.

```
>at:=4; function f(expr,x) := expr(x); ...
>f({{"at*x^2",at=5}},3)
```

```
45
```

Ekspresi dalam x sering digunakan seperti halnya fungsi.
Perhatikan bahwa mendefinisikan fungsi dengan nama yang sama seperti ekspresi simbolik global akan menghapus variabel ini untuk menghindari kebingungan antara ekspresi simbolik dan fungsi.

```
>f &= 5*x;
>function f(x) := 6*x;
>f(2)
```

```
12
```

Dengan cara konvensi, ekspresi simbolik atau numerik harus diberi nama fx, fxy dll. Skema penamaan ini tidak boleh digunakan untuk fungsi.

```
>fx &= diff(x^x,x); $&fx
```

Bentuk ekspresi khusus memungkinkan variabel apa pun sebagai parameter tanpa nama untuk mengevaluasi ekspresi, tidak hanya "x", "y", dll. Untuk ini, mulailah ekspresi dengan "@ (variabel) ...".

```
>"@(a,b) a^2+b^2", %(4,5)
```

```
@(a,b) a^2+b^2
41
```

Hal ini memungkinkan untuk memanipulasi ekspresi dalam variabel lain untuk fungsi EMT yang membutuhkan ekspresi dalam "x".

Cara paling dasar untuk mendefinisikan fungsi sederhana adalah dengan menyimpan rumusnya dalam ekspresi simbolik atau numerik. Jika variabel utamanya adalah x, ekspresi tersebut dapat dievaluasi seperti fungsi.

Seperti yang Anda lihat pada contoh berikut, variabel global terlihat selama evaluasi.

```
>fx &= x^3-a*x;  ...
>a=1.2; fx(0.5)
```

```
-0.475
```

Semua variabel lain dalam ekspresi dapat ditentukan dalam evaluasi menggunakan parameter yang ditetapkan.

```
>fx(0.5,a=1.1)
```

```
-0.425
```

Ekspresi tidak perlu simbolis. Ini diperlukan, jika ekspresi berisi fungsi, yang hanya dikenal di kernel numerik, bukan di Maxima.

Matematika Simbolis

EMT melakukan matematika simbolis dengan bantuan Maxima. Untuk detailnya, mulailah dengan tutorial berikut, atau telusuri referensi untuk Maxima. Para ahli di Maxima harus memperhatikan bahwa ada perbedaan dalam sintaks antara sintaks asli dari Maxima dan sintaks default dari ekspresi simbolik di EMT.

Matematika simbolik terintegrasi mulus ke dalam Euler dengan &. Ekspresi apa pun yang dimulai dengan & adalah ekspresi simbolis. Itu dievaluasi dan dicetak oleh Maxima.

Pertama-tama, Maxima memiliki aritmatika "tak terbatas" yang dapat menangani angka yang sangat besar.

```
>$&44!
```

$$2658271574788448768043625811014615890319638528000000000$$

Dengan cara ini, Anda dapat menghitung hasil yang besar dengan tepat. Mari kita hitung getah: C (44,10) = \ frac {44!} {34! \ cdot 10!}

```
>$& 44!/(34!*10!) // nilai C(44,10)
```

$$2481256778$$

Tentu saja, Maxima memiliki fungsi yang lebih efisien untuk ini (seperti halnya bagian numerik EMT).

```
>$binomial(44,10) //menghitung C(44,10) menggunakan fungsi binomial()
```

$$2481256778$$

Untuk mempelajari lebih lanjut tentang fungsi tertentu, klik dua kali di atasnya. Misalnya, coba klik dua kali pada "& binomial" di baris perintah sebelumnya. Ini membuka dokumentasi Maxima yang disediakan oleh penulis program itu.

Anda akan belajar bahwa yang berikut ini juga berfungsi.

getah: C (x, 3) = \ frac {x!} {(x-3)! 3!} = \ frac {(x-2) (x-1) x} {6}

```
>$binomial(x,3) // C(x,3)
```

$$\frac{(x-2)\,(x-1)\,x}{6}$$

Jika Anda ingin mengganti x dengan nilai tertentu, gunakan "dengan".

```
>$&binomial(x,3) with x=10 // substitusi x=10 ke C(x,3)
```

$$120$$

Dengan begitu, Anda bisa menggunakan solusi persamaan di persamaan lain.

Ekspresi simbolik dicetak oleh Maxima dalam bentuk 2D. Alasannya adalah adanya bendera simbolis khusus dalam string tersebut.

Seperti yang akan Anda lihat pada contoh sebelumnya dan berikut, jika Anda menginstal LaTeX, Anda dapat mencetak ekspresi simbolik dengan Latex. Jika tidak, perintah berikut akan mengeluarkan pesan kesalahan.

Untuk mencetak ekspresi simbolik dengan LaTeX, gunakan $ infront dari & (atau Anda dapat menghilangkan &) sebelum perintah. Jangan menjalankan perintah Maxima dengan $, jika Anda belum menginstal LaTeX.

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

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

Ekspresi simbolik diurai oleh Euler. Jika Anda membutuhkan sintaks kompleks dalam satu ekspresi, Anda dapat mengapit ekspresi dalam "...". Menggunakan lebih dari sekedar ekspresi sederhana dimungkinkan, tetapi sangat tidak disarankan.

```
>&"v := 5; v^2"
```

```
25
```

Untuk kelengkapan, kami menyatakan bahwa ekspresi simbolik dapat digunakan dalam program, tetapi perlu diapit tanda kutip. Selain itu, jauh lebih efektif untuk memanggil Maxima pada waktu kompilasi jika memungkinkan.

```
>$&expand((1+x)^4), $&factor(diff(%,x)) // diff: turunan, factor: faktor
```

$$4\,\left(x+1\right)^3$$

Sekali lagi,% mengacu pada hasil sebelumnya.

Untuk mempermudah, kami menyimpan solusi ke variabel simbolik. Variabel simbolik didefinisikan dengan "& =".

```
>fx &= (x+1)/(x^4+1); $&fx
```

$$\frac{x+1}{x^4+1}$$

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

```
>$&factor(diff(fx,x))
```

$$\frac{-3\,x^4-4\,x^3+1}{\left(x^4+1\right)^2}$$

Masukan langsung dari perintah Maxima juga tersedia. Mulai baris perintah dengan "::". Sintaks Maxima disesuaikan dengan sintaks EMT (disebut "mode kompatibilitas").

```
>&factor(20!)
```

```
2432902008176640000
```

```
>::: factor(10!)
```

```
 8  4  2
2  3  5  7
```

```
>:: factor(20!)
```

```
 18  8  4  2
2   3  5  7  11 13 17 19
```

Jika Anda ahli dalam Maxima, Anda mungkin ingin menggunakan sintaks asli Maxima. Anda dapat melakukan ini dengan ":::".

```
>::: av:g$ av^2;
```

```
 2
g
```

```
>fx &= x^3*exp(x), $fx
```

```
 3  x
x  E
```

Variabel semacam itu dapat digunakan dalam ekspresi simbolik lainnya. Perhatikan, bahwa dalam perintah berikut, sisi kanan & = dievaluasi sebelum penugasan ke Fx.

```
>&(fx with x=5), $%, &float(%)
```

```
                                         5
                                    125 E


                               18551.64488782208
```

```
>fx(5)
```

```
18551.6448878
```

Untuk evaluasi ekspresi dengan nilai variabel tertentu, Anda dapat menggunakan operator "dengan".
Baris perintah berikut juga menunjukkan bahwa Maxima bisa mengevaluasi ekspresi secara numerik dengan float ().

```
>&(fx with x=10)-(fx with x=5), &float(%)
```

```
                                10          5
                          1000 E    - 125 E


                              2.20079141499189e+7
```

```
>$factor(diff(fx,x,2))
```

Untuk mendapatkan kode Latex untuk ekspresi, Anda dapat menggunakan perintah tex.

```
>tex(fx)
```

```
x^3\,e^{x}
```

Ekspresi simbolik dapat dievaluasi seperti ekspresi numerik.

```
>fx(0.5)
```

```
0.206090158838
```

Dalam ekspresi simbolik, ini tidak berfungsi, karena Maxima tidak mendukungnya. Sebagai gantinya, gunakan sintaks "with" (bentuk yang lebih bagus dari perintah at (...) Maxima).

```
>$&fx with x=1/2
```

Penugasan juga bisa bersifat simbolis.

```
>$&fx with x=1+t
```

Perintah memecahkan memecahkan ekspresi simbolik untuk variabel di Maxima. Hasilnya adalah vektor solusi.

```
>$&solve(x^2+x=4,x)
```

Bandingkan dengan perintah "selesaikan" numerik di Euler, yang membutuhkan nilai awal, dan secara opsional nilai target.

```
>solve("x^2+x",1,y=4)
```

```
1.56155281281
```

Nilai numerik dari solusi simbolik dapat dihitung dengan evaluasi hasil simbolik. Euler akan membaca tugas x = dll. Jika Anda tidak memerlukan hasil numerik untuk perhitungan lebih lanjut, Anda juga dapat membiarkan Maxima menemukan nilai numerik.

```
>sol &= solve(x^2+2*x=4,x); $&sol, sol(), $&float(sol)
```

```
[-3.23607,  1.23607]
```

Untuk mendapatkan solusi simbolik tertentu, seseorang dapat menggunakan "dengan" dan indeks.

```
>$&solve(x^2+x=1,x), x2 &= x with %[2]; $&x2
```

Untuk menyelesaikan sistem persamaan, gunakan vektor persamaan. Hasilnya adalah vektor solusi.

```
>sol &= solve([x+y=3,x^2+y^2=5],[x,y]); $&sol, $&x*y with sol[1]
```

Ekspresi simbolis dapat memiliki bendera, yang menunjukkan perlakuan khusus dalam Maxima. Beberapa flag juga dapat digunakan sebagai perintah, yang lainnya tidak. Bendera ditambahkan dengan "|" (bentuk yang lebih bagus dari "ev (..., flags)")

```
>$& diff((x^3-1)/(x+1),x) //turunan bentuk pecahan
>$& diff((x^3-1)/(x+1),x) | ratsimp //menyederhanakan pecahan
>$&factor(%)
```

Fungsi

Dalam EMT, fungsi adalah program yang ditentukan dengan perintah "fungsi". Ini bisa menjadi fungsi satu baris atau fungsi multiline.
Fungsi satu baris dapat berupa numerik atau simbolik. Fungsi satu baris numerik ditentukan oleh ": =".

```
>function f(x) := x*sqrt(x^2+1)
```

Untuk gambaran umum, kami menunjukkan semua kemungkinan definisi untuk fungsi satu baris. Sebuah fungsi dapat dievaluasi sama seperti fungsi Euler bawaan lainnya.

```
>f(2)
```

```
4.472135955
```

Fungsi ini akan bekerja untuk vektor juga, mengikuti bahasa matriks Euler, karena ekspresi yang digunakan dalam fungsi tersebut adalah vektorisasi.

```
>f(0:0.1:1)
```

```
[0,  0.100499,  0.203961,  0.313209,  0.430813,  0.559017,  0.699714,
0.854459,  1.0245,  1.21083,  1.41421]
```

Fungsi bisa diplot. Sebagai ganti ekspresi, kita hanya perlu memberikan nama fungsi. Berbeda dengan ekspresi simbolik atau numerik, nama fungsi harus diberikan dalam sebuah string.

```
>solve("f",1,y=1)
```

```
0.786151377757
```

Secara default, jika Anda perlu menimpa fungsi built-in, Anda harus menambahkan kata kunci "overwrite". Menimpa fungsi built-in berbahaya dan dapat menyebabkan masalah pada fungsi lain tergantung pada fungsinya.
Anda masih bisa memanggil fungsi built-in sebagai "_...", jika itu berfungsi di inti Euler.

```
>function overwrite sin (x) := _sin(x°) // redine sine in degrees
>sin(45)
```

```
0.707106781187
```

Lebih baik kita menghapus definisi ulang dosa ini.

```
>forget sin; sin(pi/4)
```

```
0.707106781187
```

Parameter Default

Fungsi numerik dapat memiliki parameter default.

```
>function f(x,a=1) := a*x^2
```

Menghilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

```
16
```

Menyetelnya menimpa nilai default.

```
>f(4,5)
```

```
80
```

Parameter yang ditetapkan juga menimpanya. Ini digunakan oleh banyak fungsi Euler seperti plot2d, plot3d.

```
>f(4,a=1)
```

```
16
```

Jika variabel bukan parameter, itu harus global. Fungsi satu baris dapat melihat variabel global.

```
>function f(x) := a*x^2
>a=6; f(2)
```

```
24
```

Tetapi parameter yang ditetapkan menggantikan nilai global.

Jika argumen tidak ada dalam daftar parameter yang ditentukan sebelumnya, itu harus dideklarasikan dengan ": ="!

```
>f(2,a:=5)
```

```
20
```

Fungsi simbolik didefinisikan dengan "& =". Mereka didefinisikan di Euler dan Maxima, dan bekerja di kedua dunia. Ekspresi yang menentukan dijalankan melalui Maxima sebelum definisi.

```
>function g(x) &= x^3-x*exp(-x); $&g(x)
```

$$x^3 - x\,e^{-x}$$

Fungsi simbolik dapat digunakan dalam ekspresi simbolik.

```
>$&diff(g(x),x), $&% with x=4/3
```

$$3\,x^2-e^{-x}+x\,e^{-x}$$

$$\frac{16}{3}+\frac{e^{-\frac{4}{3}}}{3}$$

Mereka juga dapat digunakan dalam ekspresi numerik. Tentu saja, ini hanya akan berfungsi jika EMT dapat menafsirkan semua yang ada di dalam fungsi tersebut.

```
>g(5+g(1))
```

```
178.635099908
```

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolik lainnya.

```
>function G(x) &= factor(integrate(g(x),x)); $&G(c) // integrate: menginteg
```

$$\frac{e^{-c}\,\left(4+4\,c+c^4\,e^{c}\right)}{4}$$

```
>solve(&g(x),0.5)
```

```
0.703467422498
```

Cara berikut juga berlaku, karena Euler menggunakan ekspresi simbolik dalam fungsi g, jika tidak menemukan variabel simbolis g, dan jika ada fungsi simbolik g.

```
>solve(&g,0.5)
```

```
0.703467422498
```

```
>function P(x,n) &= (2*x-1)^n; $&P(x,n)
```

$$\left(-1+2\,x\right)^{n}$$

```
>function Q(x,n) &= (x+2)^n; $&Q(x,n)
>$&P(x,4), $&expand(%)
```

$$(-1+2\,x)^4$$

$$1-8\,x+24\,x^2-32\,x^3+16\,x^4$$

```
>P(3,4)
```

```
625
```

```
>$&P(x,4)+ Q(x,3), $&expand(%)
```

$$(-1+2\,x)^4+Q\,(x,3)$$

$$1-8\,x+24\,x^2-32\,x^3+16\,x^4+Q\,(x,3)$$

```
>$&P(x,4)-Q(x,3), $&expand(%), $&factor(%)
```

$$(-1+2\,x)^4-Q\,(x,3)$$

$$1-8\,x+24\,x^2-32\,x^3+16\,x^4-Q\,(x,3)$$

$$1-8\,x+24\,x^2-32\,x^3+16\,x^4-Q\,(x,3)$$

```
>$&P(x,4)*Q(x,3), $&expand(%), $&factor(%)
```

$$(-1+2\,x)^4\;Q\,(x,3)$$

$$Q\,(x,3)-8\,x\,Q\,(x,3)+24\,x^2\,Q\,(x,3)-32\,x^3\,Q\,(x,3)+16\,x^4\,Q\,(x,3)$$

$$(-1+2\,x)^4\;Q\,(x,3)$$

```
>$&P(x,4)/Q(x,1), $&expand(%), $&factor(%)
```

$$\frac{(-1+2\,x)^4}{Q\,(x,1)}$$

$$\frac{1}{Q\,(x,1)}-\frac{8\,x}{Q\,(x,1)}+\frac{24\,x^2}{Q\,(x,1)}-\frac{32\,x^3}{Q\,(x,1)}+\frac{16\,x^4}{Q\,(x,1)}$$

$$\frac{(-1+2\,x)^4}{Q\,(x,1)}$$

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

$$-x+x^3$$

Dengan & = fungsinya adalah simbolik, dan dapat digunakan dalam ekspresi simbolik lainnya.

```
>$&integrate(f(x),x)
```

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

Dengan: = fungsinya adalah numerik. Contoh yang baik adalah seperti integral pasti
lateks: f (x) = \ int_1 ^ x t ^ t \, dt,
yang tidak dapat dievaluasi secara simbolis.
Jika kita mendefinisikan ulang fungsi dengan kata kunci "map", ini dapat digunakan untuk vektor x. Secara internal, fungsi ini dipanggil untuk semua nilai x satu kali, dan hasilnya disimpan dalam vektor.

```
>function map f(x) := integrate("x^x",1,x)
>f(0:0.5:2)
```

```
[-0.783431,  -0.410816,  0,  0.676863,  2.05045]
```

Fungsi dapat memiliki nilai default untuk parameter.

```
>function mylog (x,base=10) := ln(x)/ln(base);
```

Sekarang fungsi tersebut dapat dipanggil dengan atau tanpa parameter "base".

```
>mylog(100), mylog(2^6.7,2)
```

```
2
6.7
```

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

```
>mylog(E^2,base=E)
```

```
2
```

Seringkali, kami ingin menggunakan fungsi untuk vektor di satu tempat, dan untuk elemen individu di tempat lain. Ini dimungkinkan dengan parameter vektor.

```
>function f([a,b]) &= a^2+b^2-a*b+b; $&f(a,b), $&f(x,y)
```

Fungsi simbolik seperti itu dapat digunakan untuk variabel simbolik.
Tetapi fungsinya juga dapat digunakan untuk vektor numerik.

```
>v=[3,4]; f(v)
```

```
17
```

Ada juga fungsi simbolik murni, yang tidak dapat digunakan secara numerik.

```
>function lapl(expr,x,y) &&= diff(expr,x,2)+diff(expr,y,2)//turunan parsial
```

```
diff(expr, y, 2) + diff(expr, x, 2)
```

```
>$&realpart((x+I*y)^4), $&lapl(%,x,y)
```

Tetapi tentu saja, mereka dapat digunakan dalam ekspresi simbolik atau dalam definisi fungsi simbolik.

```
>function f(x,y) &= factor(lapl((x+y^2)^5,x,y)); $&f(x,y)
```

Untuk meringkas

- &= mendefinisikan fungsi simbolik,
- := mendefinisikan fungsi numerik,
- &&= mendefinisikan fungsi simbolik murni.

Memecahkan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolik.

Untuk menyelesaikan ekspresi sederhana dari satu variabel, kita dapat menggunakan fungsi Solving (). Diperlukan nilai awal untuk memulai pencarian. Secara internal, Solving () menggunakan metode garis potong.

```
>solve("x^2-2",1)
```

```
1.41421356237
```

Ini bekerja untuk ekspresi simbolik juga. Ambil fungsi berikut.

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

$$\left[x=-\sqrt{2},x=\sqrt{2}\right]$$

```
>$&solve(x^2-2,x)
```

$$\left[x=-\sqrt{2},x=\sqrt{2}\right]$$

```
>$&solve(a*x^2+b*x+c=0,x)
```

$$\left[x=\frac{-b-\sqrt{b^2-4\,a\,c}}{2\,a},x=\frac{-b+\sqrt{b^2-4\,a\,c}}{2\,a}\right]$$

```
>$&solve([a*x+b*y=c,d*x+e*y=f],[x,y])
```

$$\left[\left[x=-\frac{c\,(-5+e)}{b\,(-3+d)+a\,(5-e)},y=\frac{c\,(-3+d)}{b\,(-3+d)+a\,(5-e)}\right]\right]$$

```
>px &= 4*x^8+x^7-x^4-x; $&px
```

$$-x-x^4+x^7+4\,x^8$$

Sekarang kita mencari titik, di mana polinomialnya adalah 2. Dalam Solving (), nilai target default y = 0 dapat diubah dengan variabel yang ditetapkan.
Kami menggunakan y = 2 dan memeriksa dengan mengevaluasi polinomial pada hasil sebelumnya.

```
>solve(px,1,y=2), px(%)
```

```
0.966715594851
2
```

Memecahkan ekspresi simbolis dalam bentuk simbolik mengembalikan daftar solusi. Kami menggunakan penyelesaian pemecah simbolik () yang disediakan oleh Maxima.

```
>sol &= solve(x^2-x-1,x); $&sol
```

Cara termudah untuk mendapatkan nilai numerik adalah dengan mengevaluasi solusi secara numerik seperti ekspresi.

```
>longest sol()
```

```
      -0.6180339887498949       1.618033988749895
```

Untuk menggunakan solusi secara simbolis dalam ekspresi lain, cara termudah adalah "dengan".

```
>$&x^2 with sol[1], $&expand(x^2-x-1 with sol[2])
```

Sistem pemecahan persamaan secara simbolis dapat dilakukan dengan vektor persamaan dan penyelesaian pemecah simbolik (). Jawabannya adalah daftar persamaan.

```
>$&solve([x+y=2,x^3+2*y+x=4],[x,y])
```

Fungsi f () dapat melihat variabel global. Namun seringkali kita ingin menggunakan parameter lokal.

lateks: a ^ x-x ^ a = 0,1

dengan a = 3.

```
>function f(x,a) := x^a-a^x;
```

Salah satu cara untuk meneruskan parameter tambahan ke f () adalah dengan menggunakan daftar dengan nama fungsi dan parameternya (cara lainnya adalah parameter titik koma).

```
>solve({{"f",3}},2,y=0.1)
```

```
2.54116291558
```

Ini juga bekerja dengan ekspresi. Tapi kemudian, elemen daftar bernama harus digunakan. (Lebih lanjut tentang daftar di tutorial tentang sintaks EMT).

```
>solve({{"x^a-a^x",a=3}},2,y=0.1)
```

```
2.54116291558
```

Menyelesaikan Pertidaksamaan

Untuk menyelesaikan pertidaksamaan, EMT tidak akan dapat melakukannya, melainkan dengan bantuan Maxima, artinya secara eksak (simbolik). Perintah Maxima yang digunakan adalah fourier_elim(), yang harus dipanggil dengan perintah "load(fourier_elim)" terlebih dahulu.

```
>&load(fourier_elim)
```

```
        C:/Program Files/Euler x64/maxima/share/maxima/5.35.1/share/f\
ourier_elim/fourier_elim.lisp
```

```
>$&fourier_elim([x^2 - 1>0],[x]) // x^2-1 > 0
```

$$[1 < x] \vee [x < -1]$$

```
>$&fourier_elim([x^2 - 1<0],[x]) // x^2-1 < 0
```

$$[-1 < x, x < 1]$$

```
>$&fourier_elim([x^2 - 1 # 0],[x]) // x^-1 <> 0
```

$$[-1 < x, x < 1] \vee [1 < x] \vee [x < -1]$$

```
>$&fourier_elim([x # 6],[x])
```

$$[x < 6] \vee [6 < x]$$

```
>$&fourier_elim([x < 1, x > 1],[x]) // tidak memiliki penyelesaian
```

$$emptyset$$

```
>$&fourier_elim([minf < x, x < inf],[x]) // solusinya R
```

$$universalset$$

```
>$&fourier_elim([x^3 - 1 > 0],[x])
```

$$\left[1 < x, 1 + x + x^2 > 0\right] \vee \left[x < 1, -1 - x - x^2 > 0\right]$$

```
>$&fourier_elim([cos(x) < 1/2],[x]) // ??? gagal
```

$$[1 - 2\,\cos x > 0]$$

```
>$&fourier_elim([y-x < 5, x - y < 7, 10 < y],[x,y]) // sistem pertidaksamaa
```

$$[-5 + y < x, x < 7 + y, 10 < y]$$

```
>$&fourier_elim([y-x < 5, x - y < 7, 10 < y],[y,x])
```

$$[max\,(10, -7 + x) < y, y < 5 + x, 5 < x]$$

```
>$&fourier_elim((x + y < 5) and (x - y >8),[x,y])
```

$$\left[8 + y < x, x < 5 - y, y < -\frac{3}{2}\right]$$

```
>$&fourier_elim(((x + y < 5) and x < 1) or  (x - y >8),[x,y])
```

$$[8 + y < x] \vee [x < min\,(1, 5 - y)]$$

```
>&fourier_elim([max(x,y) > 6, x # 8, abs(y-1) > 12],[x,y])
```

```
         [6 < x, x < 8, y < - 11] or [8 < x, y < - 11]
  or [x < 8, 13 < y] or [x = y, 13 < y] or [8 < x, x < y, 13 < y]
  or [y < x, 13 < y]
```

```
>$&fourier_elim([(x+6)/(x-9) <= 6],[x])
```

$$[x = 12] \vee [12 < x] \vee [x < 9]$$

Bahasa Matriks

Dokumentasi inti EMT berisi diskusi terperinci tentang bahasa matriks Euler.

Vektor dan matriks dimasukkan dengan tanda kurung siku, elemen dipisahkan dengan koma, baris dipisahkan dengan titik koma.

```
>A=[1,2;3,4]
```

```
              1             2
              3             4
```

Produk matriks dilambangkan dengan titik.

```
>b=[3;4]
```

```
              3
              4
```

```
>b' // transpose b
```

```
  [3,   4]
```

```
>inv(A) //inverse A
```

```
             -2             1
            1.5          -0.5
```

```
>A.b //perkalian matriks
```

```
             11
             25
```

```
>A.inv(A)
```

```
              1             0
              0             1
```

Poin utama dari bahasa matriks adalah bahwa semua fungsi dan operator mengerjakan elemen untuk elemen.

```
>A.A
```

```
            7            10
           15            22
```

```
>A^2 //perpangkatan elemen2 A
```

```
            1             4
            9            16
```

```
>A.A.A
```

```
           37            54
           81           118
```

```
>power(A,3) //perpangkatan matriks
```

```
           37            54
           81           118
```

```
>A/A //pembagian elemen-elemen matriks yang seletak
```

```
            1             1
            1             1
```

```
>A/b //pembagian elemen2 A oleh elemen2 b kolom demi kolom (karena b vektor
```

```
     0.333333      0.666667
         0.75             1
```

```
>A\b // hasilkali invers A dan b, A^(-1)b
```

```
           -2
          2.5
```

```
>inv(A).b
```

```
-2
2.5
```

```
>A\A   //A^(-1)A
```

```
1              0
0              1
```

```
>inv(A).A
```

```
1              0
0              1
```

```
>A*A //perkalin elemen-elemen matriks seletak
```

```
1              4
9             16
```

Ini bukan hasil perkalian matriks, tetapi perkalian elemen dengan elemen. Hal yang sama juga berlaku untuk vektor.

```
>b^2 // perpangkatan elemen-elemen matriks/vektor
```

```
 9
16
```

Jika salah satu operan adalah vektor atau skalar, itu diperluas dengan cara alami.

```
>2*A
```

```
2              4
6              8
```

Misalnya, jika operan adalah vektor kolom, elemennya diterapkan ke semua baris A.

```
>[1,2]*A
```

```
              1             4
              3             8
```

Jika itu adalah vektor baris, itu diterapkan ke semua kolom A.

```
>A*[2,3]
```

```
              2             6
              6            12
```

Dapat dibayangkan perkalian ini seolah-olah vektor baris v telah diduplikasi untuk membentuk matriks dengan ukuran yang sama dengan A.

```
>dup([1,2],2) // dup: menduplikasi/menggandakan vektor [1,2] sebanyak 2 kal
```

```
              1             2
              1             2
```

```
>A*dup([1,2],2)
```

```
              1             4
              3             8
```

Ini juga berlaku untuk dua vektor di mana satu adalah vektor baris dan yang lainnya adalah vektor kolom. Kami menghitung i * j untuk i, j dari 1 sampai 5. Triknya adalah mengalikan 1: 5 dengan transposenya. Bahasa matriks Euler secara otomatis menghasilkan tabel nilai.

```
>(1:5)*(1:5)' // hasilkali elemen-elemen vektor baris dan vektor kolom
```

```
              1             2             3             4             5
              2             4             6             8            10
              3             6             9            12            15
              4             8            12            16            20
              5            10            15            20            25
```

Sekali lagi, ingatlah bahwa ini bukan hasil perkalian matriks!

```
>(1:5).(1:5)' // hasilkali vektor baris dan vektor kolom
```

```
55
```

```
>sum((1:5)*(1:5)) // sama hasilnya
```

```
55
```

Bahkan operator seperti <atau == bekerja dengan cara yang sama.

```
>(1:10)<6 // menguji elemen-elemen yang kurang dari 6
```

```
[1,  1,  1,  1,  1,  0,  0,  0,  0,  0]
```

Misalnya, kita dapat menghitung jumlah elemen yang memenuhi kondisi tertentu dengan fungsi sum ().

```
>sum((1:10)<6) // banyak elemen yang kurang dari 6
```

```
5
```

Euler memiliki operator perbandingan, seperti "==", yang memeriksa kesetaraan.

Kami mendapatkan vektor 0 dan 1, di mana 1 berarti benar.

```
>t=(1:10)^2; t==25 //menguji elemen2 t yang sama dengan 25 (hanya ada 1)
```

```
[0,  0,  0,  0,  1,  0,  0,  0,  0,  0]
```

Dari vektor seperti itu, "nonzeros" memilih elemen bukan nol.

Dalam hal ini, kami mendapatkan indeks dari semua elemen yang lebih besar dari 50.

```
>nonzeros(t>50) //indeks elemen2 t yang lebih besar daripada 50
```

```
[8,  9,  10]
```

Tentu saja, kita dapat menggunakan vektor indeks ini untuk mendapatkan nilai t yang sesuai.

```
>t[nonzeros(t>50)] //elemen2 t yang lebih besar daripada 50
```

```
[64,  81,  100]
```

Sebagai contoh, mari kita cari semua kuadrat dari angka 1 sampai 1000, yaitu 5 modulo 11 dan 3 modulo 13.

```
>t=1:1000; nonzeros(mod(t^2,11)==5 && mod(t^2,13)==3)
```

```
[4,  48,  95,  139,  147,  191,  238,  282,  290,  334,  381,  425,
433,  477,  524,  568,  576,  620,  667,  711,  719,  763,  810,  854,
862,  906,  953,  997]
```

EMT tidak sepenuhnya efektif untuk perhitungan integer. Ini menggunakan titik mengambang presisi ganda secara internal. Namun, seringkali ini sangat berguna.

Kita bisa memeriksa keutamaan. Mari kita cari tahu, berapa banyak kuadrat ditambah 1 yang merupakan bilangan prima.

```
>t=1:1000; length(nonzeros(isprime(t^2+1)))
```

```
112
```

Fungsi nonzeros () hanya berfungsi untuk vektor. Untuk matriks, ada mnonzeros ().

```
>seed(2); A=random(3,4)
```

```
     0.765761      0.401188      0.406347      0.267829
      0.13673      0.390567      0.495975      0.952814
     0.548138      0.006085      0.444255      0.539246
```

Ini mengembalikan indeks elemen, yang bukan nol.

```
>k=mnonzeros(A<0.4) //indeks elemen2 A yang kurang dari 0,4
```

```
            1             4
            2             1
            2             2
            3             2
```

Indeks ini dapat digunakan untuk mengatur elemen ke nilai tertentu.

```
>mset(A,k,0) //mengganti elemen2 suatu matriks pada indeks tertentu
```

```
       0.765761      0.401188      0.406347             0
              0             0      0.495975      0.952814
       0.548138             0      0.444255      0.539246
```

Fungsi mset () juga dapat menyetel elemen pada indeks ke entri beberapa matriks lainnya.

```
>mset(A,k,-random(size(A)))
```

```
       0.765761      0.401188      0.406347     -0.126917
      -0.122404     -0.691673      0.495975      0.952814
       0.548138     -0.483902      0.444255      0.539246
```

Dan dimungkinkan untuk mendapatkan elemen dalam vektor.

```
>mget(A,k)
```

```
  [0.267829,  0.13673,  0.390567,  0.006085]
```

Fungsi berguna lainnya adalah extrema, yang mengembalikan nilai minimal dan maksimal di setiap baris matriks dan posisinya.

```
>ex=extrema(A)
```

```
       0.267829             4      0.765761             1
        0.13673             1      0.952814             4
       0.006085             2      0.548138             1
```

Kita dapat menggunakan ini untuk mengekstrak nilai maksimal di setiap baris.

```
>ex[,3]'
```

```
  [0.765761,  0.952814,  0.548138]
```

Ini, tentu saja, sama dengan fungsi max ().

```
>max(A)'
```

```
  [0.765761,  0.952814,  0.548138]
```

Tetapi dengan mget (), kita dapat mengekstrak indeks dan menggunakan informasi ini untuk mengekstrak elemen pada posisi yang sama dari matriks lain.

```
>j=(1:rows(A))'|ex[,4], mget(-A,j)
```

```
             1             1
             2             4
             3             1
  [-0.765761,  -0.952814,  -0.548138]
```

Fungsi Matriks Lainnya (Building Matrix)

Untuk membangun matriks, kita dapat menumpuk satu matriks di atas matriks lainnya. Jika keduanya tidak memiliki jumlah kolom yang sama, kolom yang lebih pendek diisi dengan 0.

```
>v=1:3; v_v
```

```
             1             2             3
             1             2             3
```

Demikian juga, kita dapat melampirkan matriks ke sisi lain secara berdampingan, jika keduanya memiliki jumlah baris yang sama.

```
>A=random(3,4); A|v'
```

```
      0.032444     0.0534171      0.595713      0.564454             1
       0.83916      0.175552      0.396988       0.83514             2
     0.0257573      0.658585      0.629832      0.770895             3
```

Jika mereka tidak memiliki jumlah baris yang sama, matriks yang lebih pendek diisi dengan 0.

Ada pengecualian untuk aturan ini. Bilangan real yang melekat pada matriks akan digunakan sebagai kolom yang diisi dengan bilangan real tersebut.

```
>A|1
```

```
      0.032444     0.0534171      0.595713      0.564454             1
       0.83916      0.175552      0.396988       0.83514             1
     0.0257573      0.658585      0.629832      0.770895             1
```

Dimungkinkan untuk membuat matriks vektor baris dan kolom.

```
>[v;v]
```

```
              1             2             3
              1             2             3
```

```
>[v',v']
```

```
              1             1
              2             2
              3             3
```

Tujuan utamanya adalah untuk menafsirkan vektor ekspresi untuk vektor kolom.

```
>"[x,x^2]"(v')
```

```
              1             1
              2             4
              3             9
```

Untuk mendapatkan ukuran A, kita dapat menggunakan fungsi-fungsi berikut.

```
>C=zeros(2,4); rows(C), cols(C), size(C), length(C)
```

```
 2
 4
 [2,  4]
 4
```

Untuk vektor, ada panjang ().

```
>length(2:10)
```

```
 9
```

Ada banyak fungsi lain yang menghasilkan matriks.

```
>ones(2,2)
```

```
              1             1
              1             1
```

Ini juga dapat digunakan dengan satu parameter. Untuk mendapatkan vektor dengan angka selain 1, gunakan yang berikut ini.

```
>ones(5)*6
```

```
[6,  6,  6,  6,  6]
```

Juga matriks bilangan acak dapat dihasilkan dengan acak (distribusi seragam) atau normal (distribusi Gauß).

```
>random(2,2)
```

```
     0.66566      0.831835
       0.977      0.544258
```

Berikut adalah fungsi berguna lainnya, yang menyusun kembali elemen-elemen matriks menjadi matriks lain.

```
>redim(1:9,3,3) // menyusun elemen2 1, 2, 3, ..., 9 ke bentuk matriks 3x3
```

```
           1             2             3
           4             5             6
           7             8             9
```

Dengan fungsi berikut, kita dapat menggunakan this dan fungsi dup untuk menulis fungsi rep (), yang mengulang vektor sebanyak n kali.

```
>function rep(v,n) := redim(dup(v,n),1,n*cols(v))
```

Mari kita uji.

```
>rep(1:3,5)
```

```
[1,  2,  3,  1,  2,  3,  1,  2,  3,  1,  2,  3,  1,  2,  3]
```

Fungsi multdup () menduplikasi elemen vektor.

```
>multdup(1:3,5), multdup(1:3,[2,3,2])
```

```
[1,  1,  1,  1,  1,  2,  2,  2,  2,  2,  3,  3,  3,  3,  3]
[1,  1,  2,  2,  2,  3,  3]
```

Fungsi flipx () dan flipy () mengembalikan urutan baris atau kolom matriks. Yaitu, fungsi flipx () membalik secara horizontal.

```
>flipx(1:5) //membalik elemen2 vektor baris
```

```
[5,  4,  3,  2,  1]
```

Untuk rotasi, Euler memiliki rotleft () dan rotright ().

```
>rotleft(1:5) // memutar elemen2 vektor baris
```

```
[2,  3,  4,  5,  1]
```

Sebuah fungsi khusus adalah drop (v, i), yang menghilangkan elemen dengan indeks di i dari vektor v.

```
>drop(10:20,3)
```

```
[10,  11,  13,  14,  15,  16,  17,  18,  19,  20]
```

Perhatikan bahwa vektor i dalam drop (v, i) mengacu pada indeks elemen di v, bukan nilai elemen. Jika Anda ingin menghapus elemen, Anda harus menemukan elemennya terlebih dahulu. Indeks fungsi (v, x) dapat digunakan untuk mencari elemen x dalam vektor yang diurutkan v.

```
>v=primes(50), i=indexof(v,10:20), drop(v,i)
```

```
[2,  3,  5,  7,  11,  13,  17,  19,  23,  29,  31,  37,  41,  43,  47]
[0,  5,  0,  6,  0,  0,  0,  7,  0,  8,  0]
[2,  3,  5,  7,  23,  29,  31,  37,  41,  43,  47]
```

Seperti yang Anda lihat, tidak ada salahnya untuk menyertakan indeks di luar rentang (seperti 0), indeks ganda, atau indeks yang tidak disortir.

```
>drop(1:10,shuffle([0,0,5,5,7,12,12]))
```

```
[1,  2,  3,  4,  6,  8,  9,  10]
```

Ada beberapa fungsi khusus untuk mengatur diagonal atau untuk menghasilkan matriks diagonal.

Kami mulai dengan matriks identitas.

```
>A=id(5) // matriks identitas 5x5
```

```
           1           0           0           0           0
           0           1           0           0           0
           0           0           1           0           0
           0           0           0           1           0
           0           0           0           0           1
```

Kemudian kami mengatur diagonal bawah (-1) menjadi 1: 4.

```
>setdiag(A,-1,1:4) //mengganti diagonal di bawah diagonal utama
```

```
           1           0           0           0           0
           1           1           0           0           0
           0           2           1           0           0
           0           0           3           1           0
           0           0           0           4           1
```

Perhatikan bahwa kami tidak mengubah matriks A. Kami mendapatkan matriks baru sebagai hasil dari setdiag ().

Berikut adalah fungsi yang mengembalikan matriks tri-diagonal.

```
>function tridiag (n,a,b,c) := setdiag(setdiag(b*id(n),1,c),-1,a); ...
>tridiag(5,1,2,3)
```

```
           2           3           0           0           0
           1           2           3           0           0
           0           1           2           3           0
           0           0           1           2           3
           0           0           0           1           2
```

Diagonal matriks juga dapat diekstraksi dari matriks. Untuk mendemonstrasikan ini, kami merestrukturisasi vektor 1: 9 menjadi matriks 3x3.

```
>A=redim(1:9,3,3)
```

```
           1           2           3
           4           5           6
           7           8           9
```

Sekarang kita bisa mengekstrak diagonal.

```
>d=getdiag(A,0)
```

```
[1,  5,  9]
```

Misalnya. Kita dapat membagi matriks dengan diagonalnya. Bahasa matriks memperhatikan bahwa vektor kolom d diterapkan ke matriks baris demi baris.

```
>fraction A/d'
```

```
        1         2         3
      4/5         1       6/5
      7/9       8/9         1
```

Vektorisasi

Hampir semua fungsi di Euler bekerja untuk matriks dan input vektor juga, jika memungkinkan.

Misalnya, fungsi sqrt () menghitung akar kuadrat dari semua elemen vektor atau matriks.

```
>sqrt(1:3)
```

```
[1,  1.41421,  1.73205]
```

Jadi Anda dapat dengan mudah membuat tabel nilai. Ini adalah salah satu cara untuk memplot fungsi (alternatifnya menggunakan ekspresi).

```
>x=1:0.01:5; y=log(x)/x^2; // terlalu panjang untuk ditampikan
```

Dengan ini dan operator titik dua a: delta: b, vektor nilai fungsi dapat dibuat dengan mudah.

Dalam contoh berikut, kami menghasilkan vektor nilai t [i] dengan jarak 0,1 dari -1 hingga 1. Kemudian kami menghasilkan vektor nilai fungsi

lateks: s = t ^ 3-t

```
>t=-1:0.1:1; s=t^3-t
```

```
[0,  0.171,  0.288,  0.357,  0.384,  0.375,  0.336,  0.273,  0.192,
0.099,  0,  -0.099,  -0.192,  -0.273,  -0.336,  -0.375,  -0.384,
-0.357,  -0.288,  -0.171,  0]
```

EMT memperluas operator untuk skalar, vektor, dan matriks dengan cara yang jelas.

Misalnya, vektor kolom dikali vektor baris mengembang menjadi matriks, jika operator diterapkan. Berikut ini, v 'adalah vektor yang dialihkan (vektor kolom).

```
>shortest (1:5)*(1:5)'
```

```
      1         2         3         4         5
      2         4         6         8        10
      3         6         9        12        15
      4         8        12        16        20
      5        10        15        20        25
```

Perhatikan, ini sangat berbeda dari hasil perkalian matriks. Produk matriks dilambangkan dengan titik "." di EMT.

```
>(1:5).(1:5)'
```

```
 55
```

Secara default, vektor baris dicetak dalam format kompak.

```
>[1,2,3,4]
```

```
 [1,  2,  3,  4]
```

Untuk matriks, operator khusus. menunjukkan perkalian matriks, dan A 'menunjukkan transposing. Matriks 1x1 dapat digunakan seperti bilangan real.

```
>v:=[1,2]; v.v', %^2
```

```
 5
 25
```

Untuk mengubah urutan matriks, kami menggunakan apostrof.

```
>v=1:4; v'
```

```
            1
            2
            3
            4
```

Sehingga kita dapat menghitung matriks A dikali vektor b.

```
>A=[1,2,3,4;5,6,7,8]; A.v'
```

```
              30
              70
```

Perhatikan bahwa v masih merupakan vektor baris. Jadi v'.v berbeda dari v.v '.

```
>v'.v
```

```
               1             2             3             4
               2             4             6             8
               3             6             9            12
               4             8            12            16
```

v.v 'menghitung norma v kuadrat untuk vektor baris v. Hasilnya adalah vektor 1x1, yang bekerja seperti bilangan real.

```
>v.v'
```

```
    30
```

Ada juga norma fungsi (bersama dengan banyak fungsi lain dari Aljabar Linear).

```
>norm(v)^2
```

```
    30
```

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut adalah ringkasan aturannya.

- Fungsi yang diterapkan ke vektor atau matriks diterapkan ke setiap elemen.

- Seorang operator yang beroperasi pada dua matriks dengan ukuran yang sama diterapkan berpasangan ke elemen matriks.

- Jika kedua matriks memiliki dimensi yang berbeda, keduanya diperluas dengan cara yang masuk akal, sehingga memiliki ukuran yang sama.

Misalnya, nilai skalar dikalikan vektor mengalikan nilai dengan setiap elemen vektor. Atau matriks dikalikan dengan vektor (dengan *, bukan.) Memperluas vektor ke ukuran matriks dengan menduplikasinya.

Berikut ini adalah kasus sederhana dengan operator ^.

```
>[1,2,3]^2
```

```
[1,  4,  9]
```

Ini kasus yang lebih rumit. Vektor baris dikalikan kolom mengembang kelelahan dengan menduplikasi.

```
>v:=[1,2,3]; v*v'
```

```
            1             2             3
            2             4             6
            3             6             9
```

Perhatikan bahwa produk skalar menggunakan produk matriks, bukan *!

```
>v.v'
```

```
14
```

Ada banyak fungsi untuk matriks. Kami memberikan daftar singkat. Anda harus membaca dokumentasi untuk informasi lebih lanjut tentang perintah ini.

```
sum, prod menghitung jumlah dan produk dari baris
cumsum, cumprod melakukan hal yang sama secara kumulatif
menghitung nilai ekstrem dari setiap baris
extrema mengembalikan vektor dengan informasi ekstrem
diag (A, i) mengembalikan diagonal ke-i
setdiag (A, i, v) mengatur diagonal ke-i
id (n) matriks identitas
det (A) determinan
charpoly (A) polinomial karakteristik
eigenvalues ??(A) eigenvalues
```

```
>v*v, sum(v*v), cumsum(v*v)
```

```
[1,  4,  9]
14
[1,  5,  14]
```

Operator: menghasilkan vektor baris spasi yang sama, secara opsional dengan ukuran langkah.

```
>1:4, 1:2:10
```

```
[1,  2,  3,  4]
[1,  3,  5,  7,  9]
```

Untuk menggabungkan matriks dan vektor ada operator "|" dan "_".

```
>[1,2,3]|[4,5], [1,2,3]_1
```

```
[1,  2,  3,  4,  5]
              1              2              3
              1              1              1
```

Unsur-unsur matriks disebut dengan "A [i, j]".

```
>A:=[1,2,3;4,5,6;7,8,9]; A[2,3]
```

```
6
```

Untuk vektor baris atau kolom, v [i] adalah elemen ke-i dari vektor. Untuk matriks, ini mengembalikan baris ke-i lengkap dari matriks tersebut.

```
>v:=[2,4,6,8]; v[3], A[3]
```

```
6
[7,  8,  9]
```

Indeks juga dapat berupa vektor baris indeks. : menunjukkan semua indeks.

```
>v[1:2], A[:,2]
```

```
[2,  4]
              2
              5
              8
```

Bentuk singkat dari: adalah menghilangkan indeks sepenuhnya.

```
>A[,2:3]
```

```
              2              3
              5              6
              8              9
```

Untuk tujuan vektorisasi, elemen-elemen matriks dapat diakses seolah-olah mereka adalah vektor.

```
>A{4}
```

```
4
```

Matriks juga bisa diratakan, menggunakan fungsi redim (). Ini diimplementasikan dalam fungsi flatten ().

```
>redim(A,1,prod(size(A))), flatten(A)
```

```
[1,   2,   3,   4,   5,   6,   7,   8,   9]
[1,   2,   3,   4,   5,   6,   7,   8,   9]
```

Untuk menggunakan matriks untuk tabel, mari kita reset ke format default, dan menghitung tabel nilai sinus dan cosinus. Perhatikan bahwa sudut dalam radian secara default.

```
>defformat; w=0°:45°:360°; w=w'; deg(w)
```

```
            0
           45
           90
          135
          180
          225
          270
          315
          360
```

Sekarang kami menambahkan kolom ke matriks.

```
>M = deg(w)|w|cos(w)|sin(w)
```

```
            0             0             1             0
           45      0.785398      0.707107      0.707107
           90        1.5708             0             1
          135       2.35619     -0.707107      0.707107
          180       3.14159            -1             0
          225       3.92699     -0.707107     -0.707107
          270       4.71239             0            -1
          315       5.49779      0.707107     -0.707107
          360       6.28319             1             0
```

Dengan menggunakan bahasa matriks, kita dapat menghasilkan beberapa tabel dari beberapa fungsi sekaligus.

Dalam contoh berikut, kami menghitung t [j] ^ i untuk i dari 1 ke n. Kami mendapatkan matriks, di mana setiap baris adalah tabel t ^ i untuk satu i. Yaitu, matriks memiliki elemen lateks: a_ {i, j} = t_j ^ i, \ quad 1 \ le j \ le 101, \ quad 1 \ le i \ le n

Fungsi yang tidak bekerja untuk input vektor harus di-vectorisasi. Hal ini dapat dicapai dengan kata kunci "peta" dalam definisi fungsi. Kemudian fungsi tersebut akan dievaluasi untuk setiap elemen dari parameter vektor.

Integrasi numerik integ () hanya berfungsi untuk batas interval skalar. Jadi kita perlu melakukan vektorisasi.

```
>function map f(x) := integrate("x^x",1,x)
```

Kata kunci "map" memvektorisasi fungsi. Fungsi tersebut sekarang akan bekerja
untuk vektor angka.

```
>f([1:5])
```

```
[0,  2.05045,  13.7251,  113.336,  1241.03]
```

Sub-Matriks dan Elemen-Matriks

Untuk mengakses elemen matriks, gunakan notasi braket.

```
>A=[1,2,3;4,5,6;7,8,9], A[2,2]
```

```
            1             2             3
            4             5             6
            7             8             9
5
```

Kita dapat mengakses baris matriks yang lengkap.

```
>A[2]
```

```
[4,  5,  6]
```

Dalam kasus vektor baris atau kolom, ini mengembalikan elemen vektor.

```
>v=1:3; v[2]
```

```
2
```

Untuk memastikan, Anda mendapatkan baris pertama untuk matriks 1xn dan mxn, tentukan semua kolom menggunakan indeks kedua yang kosong.

```
>A[2,]
```

```
[4,  5,  6]
```

Jika indeks adalah vektor indeks, Euler akan mengembalikan baris yang sesuai dari matriks.

Di sini kami ingin baris pertama dan kedua dari A.

```
>A[[1,2]]
```

```
              1              2              3
              4              5              6
```

Kami bahkan dapat menyusun ulang A menggunakan vektor indeks. Tepatnya, kami tidak mengubah A di sini, tetapi menghitung versi A.

```
>A[[3,2,1]]
```

```
              7              8              9
              4              5              6
              1              2              3
```

Trik indeks juga bekerja dengan kolom.

Contoh ini memilih semua baris A dan kolom kedua dan ketiga.

```
>A[1:3,2:3]
```

```
              2              3
              5              6
              8              9
```

Untuk singkatan ":" menunjukkan semua indeks baris atau kolom.

```
>A[:,3]
```

```
                   3
                   6
                   9
```

Cara lainnya, biarkan indeks pertama kosong.

```
>A[,2:3]
```

```
                   2             3
                   5             6
                   8             9
```

Kita juga bisa mendapatkan baris terakhir A.

```
>A[-1]
```

```
 [7,  8,  9]
```

Sekarang mari kita ubah elemen A dengan menetapkan submatrix dari A ke beberapa nilai. Ini sebenarnya mengubah matriks A yang disimpan.

```
>A[1,1]=4
```

```
                   4             2             3
                   4             5             6
                   7             8             9
```

Kami juga dapat menetapkan nilai ke baris A.

```
>A[1]=[-1,-1,-1]
```

```
                  -1            -1            -1
                   4             5             6
                   7             8             9
```

Kami bahkan dapat menetapkan ke sub-matriks jika memiliki ukuran yang sesuai.

```
>A[1:2,1:2]=[5,6;7,8]
```

```
            5             6            -1
            7             8             6
            7             8             9
```

Selain itu, beberapa pintasan diperbolehkan.

```
>A[1:2,1:2]=0
```

```
            0             0            -1
            0             0             6
            7             8             9
```

Peringatan: Indeks di luar batas menampilkan matriks kosong, atau pesan kesalahan, tergantung pada pengaturan sistem. Standarnya adalah pesan kesalahan. Ingat, bagaimanapun, bahwa indeks negatif dapat digunakan untuk mengakses elemen matriks yang dihitung dari akhir.

```
>A[4]
```

```
Row index 4 out of bounds!
Error in:
A[4] ...
    ^
```

Menyortir dan Mengocok

Fungsi sort () mengurutkan vektor baris.

```
>sort([5,6,4,8,1,9])
```

```
[1,  4,  5,  6,  8,  9]
```

Seringkali perlu untuk mengetahui indeks dari vektor yang diurutkan dalam vektor asli. Ini dapat digunakan untuk menyusun ulang vektor lain dengan cara yang sama.

Mari kita mengacak vektor.

```
>v=shuffle(1:10)
```

```
[4,  5,  10,  6,  8,  9,  1,  7,  2,  3]
```

Indeks tersebut berisi urutan v.

```
>{vs,ind}=sort(v); v[ind]
```

```
  [1,  2,  3,  4,  5,  6,  7,  8,  9,  10]
```

Ini bekerja untuk vektor string juga.

```
>s=["a","d","e","a","aa","e"]
```

```
 a
 d
 e
 a
 aa
 e
```

```
>{ss,ind}=sort(s); ss
```

```
 a
 a
 aa
 d
 e
 e
```

Seperti yang Anda lihat, posisi entri ganda agak acak.

```
>ind
```

```
  [4,  1,  5,  2,  6,  3]
```

Fungsi unik mengembalikan daftar elemen unik vektor yang diurutkan.

```
>intrandom(1,10,10), unique(%)
```

```
  [4,  4,  9,  2,  6,  5,  10,  6,  5,  1]
  [1,  2,  4,  5,  6,  9,  10]
```

Ini bekerja untuk vektor string juga.

```
>unique(s)
```

```
a
aa
d
e
```

Aljabar linier

EMT memiliki banyak fungsi untuk menyelesaikan masalah sistem linier, sistem jarang, atau regresi.

Untuk sistem linier Ax = b, Anda dapat menggunakan algoritma Gauss, matriks invers atau fit linier. Operator A \ b menggunakan versi algoritma Gauss.

```
>A=[1,2;3,4]; b=[5;6]; A\b
```

```
         -4
        4.5
```

Untuk contoh lain, kami menghasilkan matriks 200x200 dan jumlah barisnya. Kemudian kita menyelesaikan Ax = b menggunakan matriks invers. Kami mengukur kesalahan sebagai deviasi maksimal semua elemen dari 1, yang tentu saja merupakan solusi yang tepat.

```
>A=normal(200,200); b=sum(A); longest totalmax(abs(inv(A).b-1))
```

```
  8.790745908981989e-13
```

Jika sistem tidak memiliki solusi, kesesuaian linier meminimalkan norma kesalahan Ax-b.

```
>A=[1,2,3;4,5,6;7,8,9]
```

```
          1             2             3
          4             5             6
          7             8             9
```

Determinan dari matriks ini adalah 0.

```
>det(A)
```

```
0
```

Matriks Simbolik

Maxima memiliki matriks simbolis. Tentu saja, Maxima dapat digunakan untuk soal-soal aljabar linier sederhana. Kita dapat mendefinisikan matriks untuk Euler dan Maxima dengan &: =, dan kemudian menggunakannya dalam ekspresi simbolik. Bentuk [...] biasa untuk mendefinisikan matriks dapat digunakan di Euler untuk mendefinisikan matriks simbolik.

```
>A &= [a,1,1;1,a,1;1,1,a]; $A
```

$$\begin{pmatrix}a & 1 & 1\\ 1 & a & 1\\ 1 & 1 & a\end{pmatrix}$$

```
>$&det(A), $&factor(%)
```

$$2-2\,a+a\,\left(-1+a^2\right)$$

$$\left(-1+a\right)^2\,\left(2+a\right)$$

```
>$&invert(A) with a=0
```

$$\begin{pmatrix}-\frac{1}{2} & \frac{1}{2} & \frac{1}{2}\\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2}\\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2}\end{pmatrix}$$

```
>A &= [1,a;b,2]; $A
```

$$\begin{pmatrix}1 & a\\ b & 2\end{pmatrix}$$

Seperti semua variabel simbolik, matriks ini dapat digunakan dalam ekspresi simbolik lainnya.

```
>$&det(A-x*ident(2)), $&solve(%,x)
```

$$-a\,b+\left(1-x\right)\,\left(2-x\right)$$

$$\left[x=\frac{3-\sqrt{1+4\,a\,b}}{2},x=\frac{3+\sqrt{1+4\,a\,b}}{2}\right]$$

Nilai eigen juga dapat dihitung secara otomatis. Hasilnya adalah vektor dengan dua vektor nilai eigen dan kelipatannya.

```
>$&eigenvalues([a,1;1,a])
```

$$[[-1+a, 1+a], [1, 1]]$$

Untuk mengekstrak vektor eigen tertentu, perlu pengindeksan yang cermat.

```
>$&eigenvectors([a,1;1,a]), &%[2][1][1]
```

```
[1, - 1]
```

Matriks simbolik dapat dievaluasi dalam Euler secara numerik seperti ekspresi simbolik lainnya.

```
>A(a=4,b=5)
```

```
1            4
5            2
```

Dalam ekspresi simbolik, gunakan dengan.

```
>$&A with [a=4,b=5]
```

$$\begin{pmatrix} 1 & 4 \\ 5 & 2 \end{pmatrix}$$

Akses ke baris matriks simbolik berfungsi seperti halnya dengan matriks numerik.

```
>$&A[1]
```

$$[1, a]$$

Ekspresi simbolis dapat berisi tugas. Dan itu mengubah matriks A.

```
>&A[1,1]:=t+1; $&A
```

$$\begin{pmatrix}1+t & a\\ b & 2\end{pmatrix}$$

Ada fungsi simbolik dalam Maxima untuk membuat vektor dan matriks. Untuk ini, lihat dokumentasi Maxima atau tutorial tentang Maxima di EMT.

```
>v &= makelist(1/(i+j),i,1,3); $v
```

$$\left[\frac{1}{1+j},\frac{1}{2+j},\frac{1}{3+j}\right]$$

```
>B &:= [1,2;3,4]; $B, $&invert(B)
```

$$\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}$$

$$\begin{pmatrix}-2 & 1\\ \frac{3}{2} & -\frac{1}{2}\end{pmatrix}$$

Hasilnya dapat dievaluasi secara numerik di Euler. Untuk informasi lebih lanjut tentang Maxima, lihat pengantar Maxima.

```
>$&invert(B)()
```

```
                 -2                 1
                1.5              -0.5
```

Euler juga memiliki fungsi xinv () yang kuat, yang membuat upaya lebih besar dan mendapatkan hasil yang lebih tepat.

Perhatikan, bahwa dengan &: = matriks B telah didefinisikan sebagai simbolik dalam ekspresi simbolik dan numerik dalam ekspresi numerik. Jadi kita bisa menggunakannya di sini.

```
>longest B.xinv(B)
```

```
                                  1                                  0
                                  0                                  1
```

Misalnya. nilai eigen dari A dapat dihitung secara numerik.

```
>A=[1,2,3;4,5,6;7,8,9]; real(eigenvalues(A))
```

```
[16.1168,  -1.11684,  0]
```

Atau secara simbolis. Lihat tutorial tentang Maxima untuk detailnya.

```
>$&eigenvalues(@A)
```

$$\left[\left[\frac{15-3\sqrt{33}}{2},\frac{15+3\sqrt{33}}{2},0\right],[1,1,1]\right]$$

Nilai Numerik dalam Ekspresi simbolis

Ekspresi simbolik hanyalah string yang mengandung ekspresi. Jika kita ingin mendefinisikan nilai untuk ekspresi simbolik dan ekspresi numerik, kita harus menggunakan "&: =".

```
>A &:= [1,pi;4,5]
```

```
             1       3.14159
             4             5
```

Masih terdapat perbedaan antara bentuk numerik dan simbolik. Saat mentransfer matriks ke bentuk simbolis, pendekatan pecahan untuk real akan digunakan.

```
>$&A
```

$$\begin{pmatrix}1 & \frac{1146408}{364913}\\ 4 & 5\end{pmatrix}$$

Untuk menghindari hal ini, ada fungsi "mxmset (variabel)".

```
>mxmset(A); $&A
```

$$\begin{pmatrix}1 & 3.141592653589793\\ 4 & 5\end{pmatrix}$$

Maxima juga dapat menghitung dengan angka floating point, dan bahkan dengan angka mengambang besar dengan 32 digit. Namun, evaluasinya jauh lebih lambat.

```
>$&bfloat(sqrt(2)), $&float(sqrt(2))
```

$$1.4142135623730950488016887242097_B \times 10^0$$

$$1.414213562373095$$

Ketepatan angka floating point besar dapat diubah.

```
>&fpprec:=100; &bfloat(pi)
```

```
        3.14159265358979323846264338327950288419716939937510582097494\
4592307816406286208998628034825342117068b0
```

Variabel numerik dapat digunakan dalam ekspresi simbolik apa pun yang menggunakan "@var".

Perhatikan bahwa ini hanya diperlukan, jika variabel telah ditentukan dengan ": =" atau "=" sebagai variabel numerik.

```
>B:=[1,pi;3,4]; $&det(@B)
```

$$-5.424777960769379$$

Demo - Suku Bunga

Di bawah ini, kami menggunakan Euler Math Toolbox (EMT) untuk menghitung suku bunga. Kami melakukannya secara numerik dan simbolis untuk menunjukkan kepada Anda bagaimana Euler dapat digunakan untuk memecahkan masalah kehidupan nyata.

Asumsikan Anda memiliki modal awal 5000 (katakanlah dalam dolar).

```
>K=5000
```

```
5000
```

Sekarang kami mengasumsikan tingkat bunga 3% per tahun. Mari kita tambahkan satu tingkat sederhana dan hitung hasilnya.

```
>K*1.03
```

```
5150
```

Euler akan memahami sintaks berikut juga.

```
>K+K*3%
```

```
5150
```

Tapi lebih mudah menggunakan faktornya

```
>q=1+3%, K*q
```

```
1.03
5150
```

Selama 10 tahun, kita cukup mengalikan faktor dan mendapatkan nilai akhir dengan suku bunga majemuk.

```
>K*q^10
```

```
6719.58189672
```

Untuk tujuan kami, kami dapat mengatur format menjadi 2 digit setelah titik desimal.

```
>format(12,2); K*q^10
```

```
    6719.58
```

Mari kita cetak yang dibulatkan menjadi 2 digit itu dalam kalimat lengkap.

```
>"Mulai dari " + K + "$ Anda mendapatkan " + round(K*q^10,2) + "$."
```

```
Mulai dari 5000$ Anda mendapatkan 6719.58$.
```

Bagaimana jika kita ingin mengetahui hasil antara tahun 1 sampai tahun 9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak perlu menulis loop, tetapi cukup masukkan

```
>K*q^(0:10)
```

```
Real 1 x 11 matrix

     5000.00     5150.00     5304.50     5463.64     ...
```

Bagaimana keajaiban ini bekerja? Pertama, ekspresi 0:10 mengembalikan vektor bilangan bulat.

```
>short 0:10
```

```
[0,  1,  2,  3,  4,  5,  6,  7,  8,  9,  10]
```

Kemudian semua operator dan fungsi di Euler dapat diterapkan ke elemen vektor untuk elemen. jadi

```
>short q^(0:10)
```

```
[1,  1.03,  1.0609,  1.0927,  1.1255,  1.1593,  1.1941,  1.2299,
1.2668,  1.3048,  1.3439]
```

adalah vektor faktor q ^ 0 hingga q ^ 10. Ini dikalikan dengan K, dan kita mendapatkan nilai vektor.

```
>VK=K*q^(0:10);
```

Tentu saja, cara realistis untuk menghitung suku bunga ini adalah dengan membulatkan ke sen terdekat setiap tahun. Mari kita tambahkan fungsi untuk ini.

```
>function oneyear (K) := round(K*q,2)
```

Mari kita bandingkan kedua hasil tersebut, dengan dan tanpa pembulatan.

```
>longest oneyear(1234.57), longest 1234.57*q
```

```
                  1271.61
                1271.6071
```

Sekarang tidak ada rumus sederhana untuk tahun ke-n, dan kita harus mengulang selama bertahun-tahun. Euler memberikan banyak solusi untuk ini.

Cara termudah adalah fungsi iterasi, yang mengulang fungsi tertentu beberapa kali.

```
>VKr=iterate("oneyear",5000,10)
```

```
  Real 1 x 11 matrix

      5000.00     5150.00     5304.50     5463.64     ...
```

Kami dapat mencetaknya dengan cara yang ramah, menggunakan format kami dengan tempat desimal tetap.

```
>VKr'
```

```
      5000.00
      5150.00
      5304.50
      5463.64
      5627.55
      5796.38
      5970.27
      6149.38
      6333.86
      6523.88
      6719.60
```

Untuk mendapatkan elemen tertentu dari vektor, kami menggunakan indeks dalam tanda kurung siku.

```
>VKr[2], VKr[1:3]
```

```
      5150.00
      5000.00     5150.00     5304.50
```

Anehnya, kita juga bisa menggunakan indeks vektor. Ingat bahwa 1: 3 menghasilkan vektor [1,2,3].

Mari kita bandingkan elemen terakhir dari nilai yang dibulatkan dengan nilai penuh.

```
>VKr[-1], VK[-1]
```

```
      6719.60
      6719.58
```

Perbedaannya sangat kecil.

Memecahkan Persamaan

Sekarang kita ambil fungsi yang lebih maju, yang menambahkan jumlah uang tertentu setiap tahun.

```
>function onepay (K) := K*q+R
```

Kami tidak harus menentukan q atau R untuk definisi fungsi. Hanya jika kita menjalankan perintah, kita harus menentukan nilai-nilai ini. Kami memilih R = 200.

```
>R=200; iterate("onepay",5000,10)
```

```
 Real 1 x 11 matrix

     5000.00     5350.00     5710.50     6081.82     ...
```

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?

```
>R=-200; iterate("onepay",5000,10)
```

```
 Real 1 x 11 matrix

     5000.00     4950.00     4898.50     4845.45     ...
```

Kami melihat bahwa uang berkurang. Jelas, jika kita hanya mendapatkan 150 bunga di tahun pertama, tetapi menghapus 200, kita kehilangan uang setiap tahun.

Bagaimana kita bisa menentukan berapa tahun uang itu akan bertahan? Kami harus menulis loop untuk ini. Cara termudah adalah mengulanginya cukup lama.

```
>VKR=iterate("onepay",5000,50)
```

```
 Real 1 x 51 matrix

     5000.00     4950.00     4898.50     4845.45     ...
```

Dengan menggunakan bahasa matriks, kita dapat menentukan nilai negatif pertama dengan cara berikut.

```
>min(nonzeros(VKR<0))
```

```
48.00
```

Alasan untuk ini adalah bahwa nonzeros (VKR <0) mengembalikan vektor indeks i, di mana VKR [i] <0, dan min menghitung indeks minimal.

Karena vektor selalu dimulai dengan indeks 1, jawabannya adalah 47 tahun.

Fungsi iterate () memiliki satu trik lagi. Ini bisa mengambil kondisi akhir sebagai argumen. Maka itu akan mengembalikan nilai dan jumlah iterasi.

```
>{x,n}=iterate("onepay",5000,till="x<0"); x, n,
```

```
-19.83
 47.00
```

Mari kita coba menjawab pertanyaan yang lebih ambigu. Asumsikan kita tahu bahwa nilainya 0 setelah 50 tahun. Berapa tingkat bunganya?

Ini adalah pertanyaan yang hanya bisa dijawab secara numerik. Di bawah ini, kami akan mendapatkan rumus yang diperlukan. Maka Anda akan melihat bahwa tidak ada rumus mudah untuk suku bunga. Tetapi untuk saat ini, kami bertujuan untuk solusi numerik.

Langkah pertama adalah menentukan fungsi yang melakukan iterasi sebanyak n kali. Kami menambahkan semua parameter ke fungsi ini.

```
>function f(K,R,P,n) := iterate("x*(1+P/100)+R",K,n;P,R)[-1]
```

Iterasinya sama seperti di atas

lateks: x_ {n + 1} = x_n \ cdot \ kiri (1+ \ frac {P} {100} \ kanan) + R

Tapi kami lebih lama menggunakan nilai global R dalam ekspresi kami. Fungsi seperti iterate () memiliki trik khusus di Euler. Anda dapat meneruskan nilai variabel dalam ekspresi sebagai parameter titik koma. Dalam hal ini P dan R.

Apalagi kami hanya tertarik pada nilai terakhir. Jadi kami mengambil indeks [-1].

Mari kita coba tes.

```
>f(5000,-200,3,47)
```

```
-19.83
```

Sekarang kita bisa menyelesaikan masalah kita.

```
>solve("f(5000,-200,x,50)",3)
```

```
3.15
```

Rutin menyelesaikan memecahkan ekspresi = 0 untuk variabel x. Jawabannya adalah 3,15% per tahun. Kami mengambil nilai awal 3% untuk algoritme. Fungsi Solving () selalu membutuhkan nilai awal.

Kita dapat menggunakan fungsi yang sama untuk menjawab pertanyaan berikut: Berapa banyak yang dapat kita keluarkan per tahun sehingga modal awal habis setelah 20 tahun dengan asumsi tingkat bunga 3% per tahun.

```
>solve("f(5000,x,3,20)",-200)
```

```
-336.08
```

Perhatikan bahwa Anda tidak dapat menyelesaikan jumlah tahun, karena fungsi kami mengasumsikan n sebagai nilai integer.

Solusi Simbolis untuk Masalah Suku Bunga

Kita dapat menggunakan bagian simbolik Euler untuk mempelajari masalahnya. Pertama kita mendefinisikan fungsi onepay () secara simbolis.

```
>function op(K) &= K*q+R; $&op(K)
```

$$q\,K+R$$

Kami sekarang dapat mengulang ini.

```
>$&op(op(op(op(K)))), $&expand(%)
```

$$R+q\,\left(R+q\,\left(R+q\,\left(q\,K+R\right)\right)\right)$$

$$q^4\,K+R+q\,R+q^2\,R+q^3\,R$$

Kami melihat sebuah pola. Setelah n periode yang kita miliki

lateks: K_n = q ^ n K + R (1 + q + \ ldots + q ^ {n-1}) = q ^ n K + \ frac {q ^ n-1} {q-1} R

Rumusnya adalah rumus jumlah geometris, yang dikenal dengan Maxima.

```
>&sum(q^k,k,0,n-1); $& % = ev(%,simpsum)
```

$$\sum_{k=0}^{-1+n} q^k = \frac{-1+q^n}{-1+q}$$

Ini agak rumit. Jumlahnya dievaluasi dengan bendera "simpsum" untuk menguranginya menjadi hasil bagi.

Mari kita buat fungsi untuk ini.

```
>function fs(K,R,P,n) &= (1+P/100)^n*K + ((1+P/100)^n-1)/(P/100)*R; $&fs(K,
```

$$K\left(1+\frac{P}{100}\right)^n + \frac{100\,\left(-1+\left(1+\frac{P}{100}\right)^n\right)\,R}{P}$$

Fungsinya sama dengan fungsi f kita sebelumnya. Tapi itu lebih efektif.

```
>longest f(5000,-200,3,47), longest fs(5000,-200,3,47)
```

```
    -19.82504734650985
    -19.82504734652684
```

Sekarang kita dapat menggunakannya untuk menanyakan waktu n. Kapan modal kita habis? Tebakan awal kami adalah 30 tahun.

```
>solve("fs(5000,-330,3,x)",30)
```

```
     20.51
```

Jawaban ini mengatakan bahwa itu akan menjadi negatif setelah 21 tahun.

Kita juga dapat menggunakan sisi simbolik Euler untuk menghitung rumus pembayaran.

Asumsikan kita mendapat pinjaman sebesar K, dan membayar n pembayaran R (dimulai setelah tahun pertama) meninggalkan sisa utang Kn (pada saat pembayaran terakhir). Rumusnya jelas

```
>equ &= fs(K,R,P,n)=Kn; $&equ
```

$$K\,\left(1+\frac{P}{100}\right)^{n}+\frac{100\,\left(-1+\left(1+\frac{P}{100}\right)^{n}\right)\,R}{P}=Kn$$

Biasanya rumus ini diberikan dalam bentuk
getah: i = \ frac {P} {100}

```
>equ &= (equ with P=100*i); $&equ
```

$$\left(1+i\right)^{n}\,K+\frac{\left(-1+\left(1+i\right)^{n}\right)\,R}{i}=Kn$$

Kita bisa mencari nilai R secara simbolis.

```
>$&solve(equ,R)
```

$$\left[R=\frac{i\,Kn-i\,\left(1+i\right)^{n}\,K}{-1+\left(1+i\right)^{n}} \right]$$

Seperti yang Anda lihat dari rumusnya, fungsi ini mengembalikan kesalahan titik mengambang untuk i = 0. Euler tetap merencanakannya.
Tentu saja, kami memiliki batasan berikut.

```
>$&limit(R(5000,0,x,10),x,0)
```

$$\lim_{x\rightarrow 0}{R\left(5000 , 0 , x , 10\right)}$$

Jelas, tanpa bunga kita harus membayar kembali 10 bunga dari 500.
Persamaan ini juga bisa diselesaikan untuk n. Ini terlihat lebih bagus, jika kita menerapkan beberapa penyederhanaan padanya.

```
>fn &= solve(equ,n) | ratsimp; $&fn
```

$$\left[n=\frac{\log \left(\frac{i\,Kn+R}{i\,K+R}\right)}{\log \left(1+i\right)} \right]$$

Menjumlahkan dan mengurangkan bentuk-bentuk aljabar dilakukan dengan cara menggabungkan suku-suku yang sejenis.

Mengalikan dua bentuk aljabar dilakukan dengan cara mengalikan setiap suku dari kedua bentuk aljabar, kemudian menjumlahkan suku-suku sejenis dari hasil perkalian tersebut.

Pembagian dua monomial dilakukan dengan cara mengalikan hasil pembagian koefisien-koefisien numerik dan hasil pembagian variabel-variabel pada pembilang dan penyebut.

Contoh soal
Soal 1

$$(2x+3y+z-7)+(4x-2y-z+8)+(-3x+y-2z-4)$$

Penyelesaian :

```
>$&(2*x+3*y+z-7)+(4*x-2*y-z+8)+(-3*x+y-2*z-4)
```

$$-2\,z+2\,y+3\,x-3$$

Soal 2

$$(3x^2-2x-x^3+2)-(5x^2-8x-x^3+4)$$

Penyelesaian :

```
>$&(3*x^2-2*x-x^3+2)-(5*x^2-8*x-x^3+4)
```

$$-2\,x^2+6\,x-2$$

Soal 3

$$(m^{x-b}n^{x+b})^x(m^bn^{-b})^x$$

Penyelesaian :

```
>$&(m^(x-b)*n^(x+b))^x*(m^b*n^(-b))^x
```

$$\left(\frac{m^{b}}{n^{b}}\right)^{x}\,\left(m^{x-b}\,n^{x+b}\right)^{x}$$

Soal 4

$$\frac{a^2-4a+4}{a-2}$$

Penyelesaian :

```
>$&(a^2-4*a+4)/(a-2), $&ratsimp(%)
```

$$\frac{a^2-4\,a+4}{a-2}$$

$$a-2$$

Fungsi ratsimp digunakan untuk menyederhanakan bentuk pecahan.

Menjabarkan bentuk aljabar :

Untuk menjabarkan bentuk aljabar, gunakan fungsi expand().
Diberikan bentuk aljabar

$$(a+b)^3$$

```
>$&(a+b)^3
```

$$\left(b+a\right)^3$$

```
>$&expand((a+b)^3)
```

$$b^3+3\,a\,b^2+3\,a^2\,b+a^3$$

```
>&powerdisp: true
```

```
true
```

Gunakan fungsi ini agar variabelnya urut dari a-z

```
>$&expand((a+b)^3)
```

$$a^3+3\,a^2\,b+3\,a\,b^2+b^3$$

Soal 1

$$(2x+3y)^2$$

Penyelesaian :

```
>$&expand((2*x+3*y)^2)
```

$$4\,x^2+12\,x\,y+9\,y^2$$

Soal 2

$$(4x^2-5y)^2$$

Penyelesaian :

```
>$&expand((4*x^2-5*y)^2)
```

$$16\,x^4-40\,x^2\,y+25\,y^2$$

Soal 3

$$(y-2)(y+2)(y^2+4)$$

Penyelesaian :

```
>$&expand((y-2)*(y+2)*(y^2+4))
```

$$-16 + y^4$$

Memfaktorkan Bentuk Aljabar :

Dalam memfaktorkan bentuk aljabar, gunakan fungsi factor().
Gunakan fungsi powerdisp: false agar penulisan variabelnya di depan.

```
>&powerdisp: false
```

```
false
```

Contoh soal
Soal 1

$$m^2 - 9n^2$$

Penyelesaian :

```
>$&factor(m^2-9*n^2)
```

$$(m - 3\,n)\,(3\,n + m)$$

Soal 2

$$t^2 + 8t + 15$$

Penyelesaian :

```
>$&factor(t^2+8*t+15)
```

$$(t + 3)\,(t + 5)$$

Soal 3

$$z^2 - 81$$

Penyelesaian :

```
>$&factor(z^2-81)
```

$$(z-9)\ (z+9)$$

Soal 4

$$p^3 - 2p^2 - 9p + 18$$

Penyelesaian :

```
>$&factor(p^3-2*p^2-9*p+18)
```

$$(p-3)\ (p-2)\ (p+3)$$

Operasi dan Fungsi Matematika

Operasi aljabar meliputi :
1. Penjumlahan
Suku-suku yang dapat dijumlahkan dalam bentuk aljabar adalah suku-suku yang sejenis. Penjumlahan bentuk aljabar dapat dilakukan dengan menjumlahkan koefisien dengan koefisien maupun konstanta dengan konstanta pada suku yang sejenis tanpa merubah variabel.
2. Pengurangan
Pennguranga bentuk ini dapat dilakukan dengan mengurangkan koefisien dengan koefisien maupun konstanta dengan konstanta pada suku yang sejenis tanpa merubah variabel.
3. Perkalian
Perkalian pada bentuk aljabar dapat diselesaikan dengan cara distributif.
Misalkan diberikan bentuk aljabar

$$(x+a)(x+b)$$

$$x \times x + b \times x + a \times x + a \times b$$

$$x^2 + ax + bx + ab$$

4. Pembagian
Pembagian aljabar biasanya diselesaikan dalam bentuk pecahan, di mana penyelesaiannya dilakukan dengan membangi koefisien dan variabel masing-masing.
Contoh soal
soal 1

$$(2x^4 - 3x^2 + 7x) - (5x^3 + 2x^2 - 3x + 5)$$

Penyelesaian :

```
>$&(2*x^4-3*x^2-7*x)-(5*x^3+2*x^2-3*x+5)
```

$$2\,x^4 - 5\,x^3 - 5\,x^2 - 4\,x - 5$$

Soal 2

$$(2x^2 + 12x - 11) + (6x^2 - 2x + 4)$$

Penyelesaian :

```
>$&(2*x^2+12*x-11)+(6*x^2-2*x+4)
```

$$8\,x^2 + 10\,x - 7$$

Soal 3

$$(3x + 5y)(3x - 5y)$$

Penyelesaian :

```
>$&expand((3*x+5*y)*(3*x-5*y))
```

$$9\,x^2 - 25\,y^2$$

Soal 4

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

Penyelesaian :

```
>$&ratsimp((x^2-4)/(x^2-4*x+4))
```

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

Soal 5

$$(5x+2y)^2$$

Penyelesaian :

```
>$&expand((5*x+2*y)^2)
```

$$25\,x^2+20\,x\,y+4\,y^2$$

Fungsi
"function" digunakan untuk mendefinisikan sebuah fungsi.
Fungsu baris numerik ditentukan oleh ":=".

```
>function f(x) := x^2+3*x-28
>f(2)
```

```
-18
```

Turunan

```
>$&diff(x^2+5*x-25,x)
```

$$2\,x+5$$

Integral

```
>$&integrate(2*x^3+4*x-12,x)
```

$$\frac{x^4}{2}+2\,x^2-12\,x$$

Bilangan kompleks

Bilangan kompleks merupakan bilangan yang terdiri atas bagian riil dan bagian imajiner.

Secara umum, bilangan kompleks dilambangkan dengan a + ib, dengan a dan b merupakan bilangan real.

Adapun yang menyebabkan bilangan tersebut menjadi bilangan kompleks yaitu keberadaan "i" atau dapat disebut sebagai bilangan imajiner.

Pada bilangan kompleks berbentuk a + ib, bagian "a" merupakan bagian real dan "ib" merupakan bagian imajiner.

$$i=\sqrt{-1}$$

Operasi bilangan kompleks

1. Penjumlahan

$$(-5+3i)+(7+8i)$$

Penyelesaian :

```
>$&(-5+3*I)+(7+8*I)
```

$$2+11\,i$$

$$(12+3i)+(-8+5i)$$

Penyelesaian :

```
>$&(12+3*I)+(-8+5*I)
```

$$4+8\,i$$

Dalam operasi penjumlahan bilangan kompleks, penjumlahan dilakukan dengan mengelompokkan bagian riil dan bagian imajinernya lalu dijumlahkan masing-masing.

2. Pengurangan

$$(10+7i)-(5+3i)$$

Penyelesaian :

```
>$&(10+7*I)-(5+3*I)
```

$$5+4\,i$$

$$(4-9i)-(2+3i)$$

Penyelesaian :

```
>$&(4-9*I)-(2+3*I)
```

$$2-12\,i$$

Sama dengan operasi penjumlahan, pada operasi pengurangan bilangan kompleks dilakukan dengan mengelompokkan bagian riil dan imajinernya kemudian dilakukan pengurangan pada masing-masing bagian.

3. Perkalian

$$7i(2-5i)$$

Penyelesaian :

```
>$&expand((7*I)*(2-5*I))
```

$$35+14\,i$$

Operasi perkalian pada bilangan kompleks memiliki sifat distributif perkalian seperti halnya pada bilangan real. Apabila faktor yang dikalikan ada lebih dari dua faktor, maka setiap faktor dikali secara bertahap.

$$7i(2-5i) = 7i \times 2 + 7i \times (-5i) = 14i + (-35i^2) = 14i + (-35 \times -1) = 14i + 35$$

Ingat bahwa

$$i = \sqrt{-1}$$

$$i^2 = -1$$

4. Pembagian
Pembagian bilangan kompleks baru dapat dilakukan apabila penyebutnya berupa bilangan real. Oleh karena itu apabila persoalan pembagian bilangan kompleks memiliki penyebut bilangan kompleks, penyebutnya harus dikali menjadi bilangan kompleks konjugatnya agar berubah menjadi bilangan real.

Perhitungan dengan fungsi buatan sendiri

Soal 1
Akan dicari faktor dan penyelesaian dari persamaan berikut.

$$x^2 + 10x + 16$$

Penyelesaian :

```
>$&(x^2+10*x+16), $&factor(%), $&solve(%)
```

$$16 + 10\,x + x^2$$

$$(2 + x)\,(8 + x)$$

$$[x = -8, x = -2]$$

Soal 2
Diketahui :

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

$$g(x) = x - 3$$

Akan dicari g(f(x)) dan f(g(x))
Penyelesaian :

```
>function f(x) &= 2*x+5
```

```
2 x + 5
```

```
>function g(x) &= x-3
```

```
x - 3
```

```
>$&g(f(x))
```

$$2\,x+2$$

```
>$&f(g(x)), $&expand(%)
```

$$2\,\left(x-3\right)+5$$

$$2\,x-1$$

Soal 3

```
>function h(x) &= 3*x^2+8*x-3
```

```
                           2
                        3 x  + 8 x - 3
```

```
>$&h(3)
```

$$48$$

Menyelesaikan persamaan dan sistem persamaan

Dalam menyelesaikan suatu persamaan, gunakan fungsi solve().
Contoh soal
Soal 1

$$x^2 - 36 = 0$$

Penyelesaian :

```
>$&solve(x^2-36=0)
```

$$[x = -6, x = 6]$$

Soal 2

$$y^2 - 4y - 45 = 0$$

Penyelesaian :

```
>$&solve(y^2-4*y-45=0)
```

$$[y = 9, y = -5]$$

Soal 3

$$x^2 - 4x - 32 = 0$$

Penyelesaian :

```
>$&solve(x^2-4*x-32=0)
```

$$[x = -4, x = 8]$$

Soal 4

$$7(3x + 6) = 11 - (x + 2)$$

Penyelesaian :

```
>$&solve(7*(3*x+6)=11-(x+2))
```

$$\left[x = -\frac{3}{2}\right]$$

Soal 5

$$5t^2 - 8t = 3$$

Penyelesaian :

```
>$&solve(5*t^2-8*t=3)
```

$$\left[t = \frac{4 - \sqrt{31}}{5}, t = \frac{4 + \sqrt{31}}{5}\right]$$

Menyelesaikan pertidaksamaan sistem pertidaksamaan

Dalam menyelesaikan pertidaksamaan, gunakan fungsi fourier_elim().
Contoh soal
soal 1

$$|2y-3|>=1-y+5$$

Penyelesaian :

```
>$&fourier_elim([2*y-3>=1-y+5],[y])
```

$$[y=3]\vee[3<y]$$

Soal 2

$$(x-2)(x+5)>x(x-3)$$

Penyelesaian :

```
>$&fourier_elim([(x-2)*(x+5)>x*(x-3)],[x])
```

$$\left[\frac{5}{3}<x\right]$$

Soal 3

$$|x+6|>=7$$

Penyelesaian :

```
>$&fourier_elim([x+6>=7,x+6<=-7],[x])
```

$$emptyset$$

Jadi, pertidaksamaan tersebut tidak mempunyai penyelesaian.
Soal 4

$$|x+5|>2$$

Penyelesaian:

```
>$&fourier_elim([x+5>2,x+5<(-2)],[x])
```

$$emptyset$$

Soal 5

$$|2x-1|<5$$

Penyelesaian :

```
>$&fourier_elim([2*x-1<5,2*x-1>-5],[x])
```

$$[-2<x,x<3]$$

Manipuasi perhitungan dengan matriks vektor

Soal 1

```
>$M = [3,2,4;1,6,4;-2,8,1]
```

$$M=\begin{pmatrix}3 & 2 & 4\\ 1 & 6 & 4\\ -2 & 8 & 1\end{pmatrix}$$

```
>inv(M)
```

```
          1.625          -1.875                1
         0.5625         -0.6875              0.5
          -1.25            1.75               -1
```

```
>det(M)
```

```
  -16
```

Untuk mencari invers suatu matriks, gunakan fungsi inv(), sedangkan untuk mencari determinan, gunakan fungsi det()

Soal 2

```
>N &= [9,6;2,4]; $N
```

$$\begin{pmatrix}9 & 6\\ 2 & 4\end{pmatrix}$$

```
>O &= [3,8;-12,8]; $O
```

$$\begin{pmatrix}3 & 8\\ -12 & 8\end{pmatrix}$$

```
>$&(N) + (O)
```

$$\begin{pmatrix}12 & 14\\ -10 & 12\end{pmatrix}$$

BAB 3

MENGGUNAKAN EMT UNTUK MENGGAMBAR GRAFIK 2D

Menggambar Grafik 2D dengan EMT

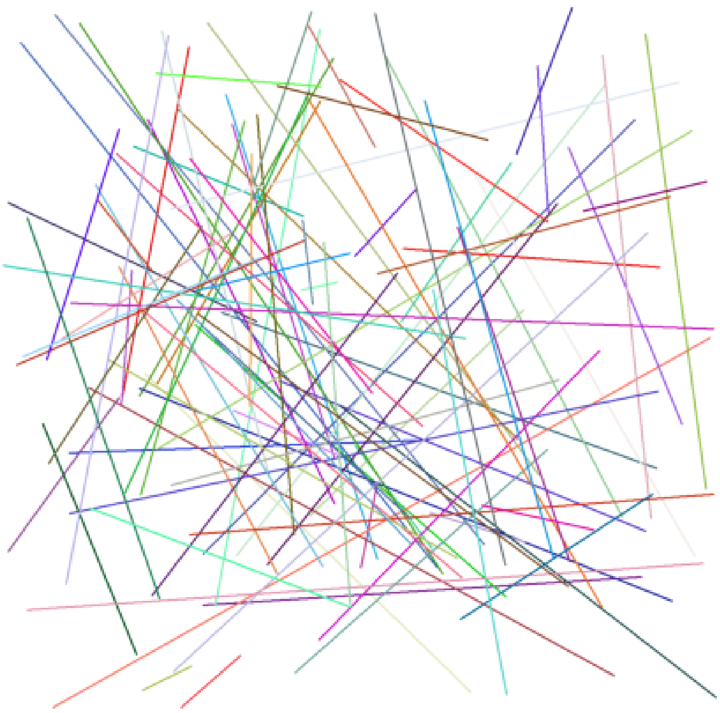
Notebook ini menjelaskan tentang cara menggambar berbagaikurva dan grafik 2D dengan software EMT. EMT menyediakan fungsi plot2d() untuk menggambar berbagai kurva dan grafik dua dimensi (2D).

Plot Dasar

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

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

```
>clg; // membersihkan layar
>window(0,0,1024,1024); // gunakan semua jendela
>setplot(0,1,0,1); // mengatur koordinat plot
>hold on; // mulai mode timpa
>n=100; X=random(n,2); Y=random(n,2);  // mendapatkan poin acak
>colors=rgb(random(n),random(n),random(n)); // mendapatkan warna acak
>loop 1 to n; color(colors[#]); plot(X[#],Y[#]); end; // plot
>hold off; // akhiri mode timpa
>insimg; // sisipkan ke notebook
```



```
>reset;
```

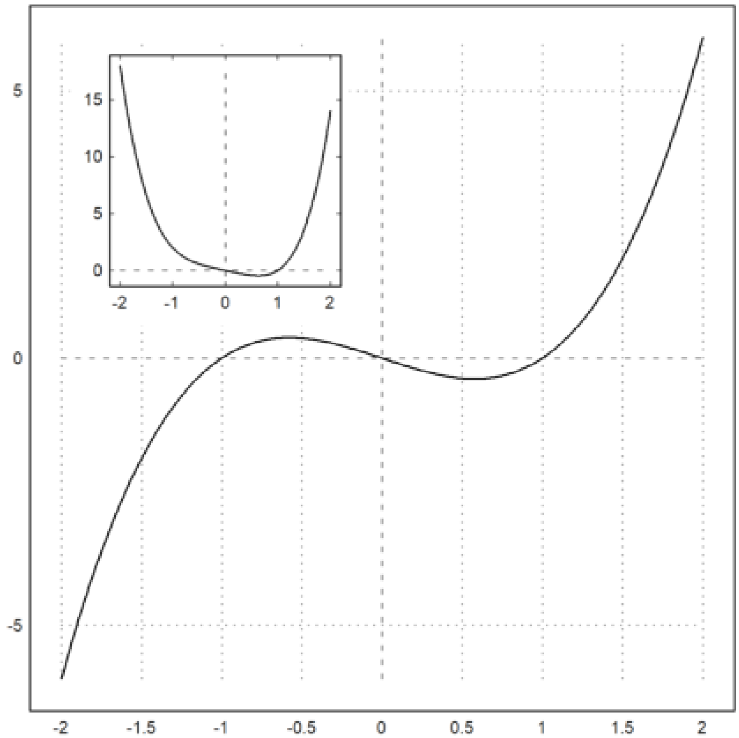
Anda harus menahan grafik, karena perintah plot() akan menghapus jendela plot.

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 gunakan 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 sementara 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);
```

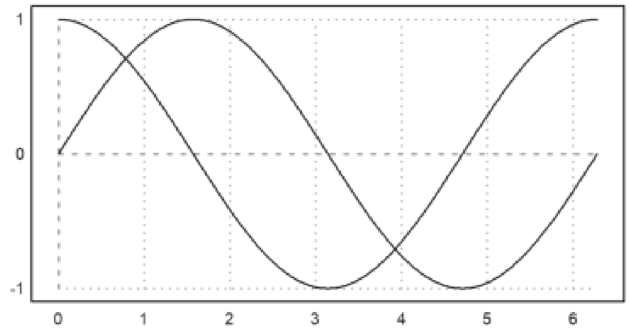
Plot dengan beberapa angka dicapai dengan cara yang sama. Ada fungsi utility figure() untuk ini.

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 grafis saat ini.

Tetapi Anda juga dapat mengubahnya untuk satu plot. Untuk melakukan 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() memulihkan default plot, termasuk rasio aspek.

2D Plots 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.

Hal ini memungkinkan untuk memplot di Maxima menggunakan Gnuplot atau di Python menggunakan Math Plot Lib.

Euler dapat memplot plot 2D dari
- ekspresi
- fungsi, variabel, atau kurva yang diparameterkan,
- vektor nilai x-y,
- awan titik-titik di dalam pesawat,
- 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*x^2") 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.

1. Fungsi Linear
Fungsi Linear merupakan suatu fungsi yang membentuk grafik secara garis lurus. Fungsi linear mengandung satu variabel dengan pangkat tertinggi satu.
Bentuk umum :

$$ax + b$$

Contoh :

$$2x + 3$$

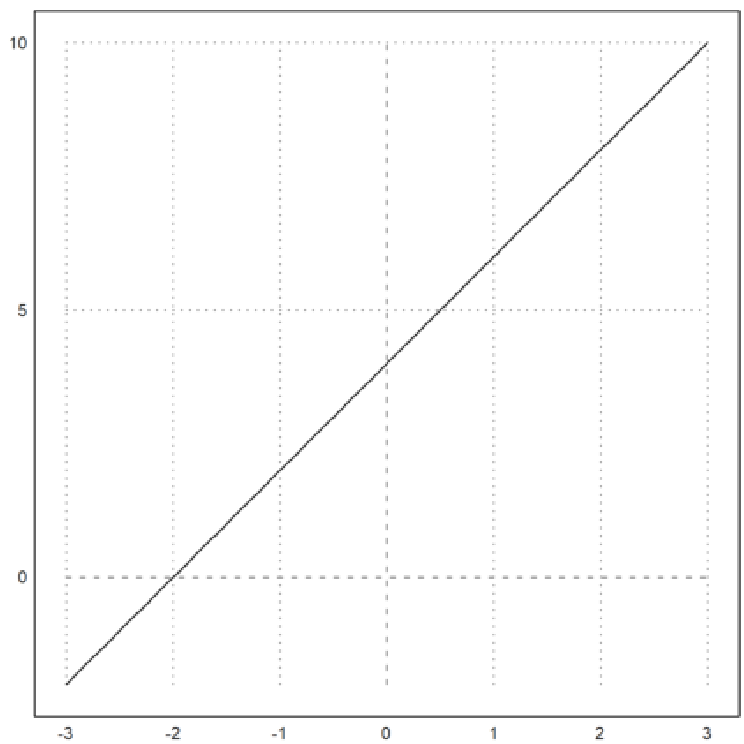
$$4x$$

$$8x - 7$$

Akan digambar grafik fungsi linear yaitu

$$2x + 4$$

```
>aspect(1,1); plot2d("2*x+4",-3,3): // menampilan grafik dengan rasio 1:1 y
```



Jika dilihat dari gambar di atas, fungsi tersebut berpotongan dengan sumbu x di titik (-2,0) dan sumbu y di titkik (0,4).

Secara manual,suatu grafik akan berpotongan di sumbu x ketika y=0, sedangkan berpotongan di sumbu y ketika x=0.

$$y = 0$$

$$2x + 4 = 0$$

$$2x = -4$$

$$x = -2$$

$$x = 0$$

$$2(0) + 4 = y$$

$$y = 4$$

Sehingga diperoleh titik potong sumbu x yaitu titik (-2,0) dan titik potong sumbu y di titik (0,4).

---

2. Fungsi Kuadrat
Fungsi kuadrat adalah suatu fungsi yang memiliki bentuk umum

$$ax^2 + bx + c$$

dengan a tidak boleh nol

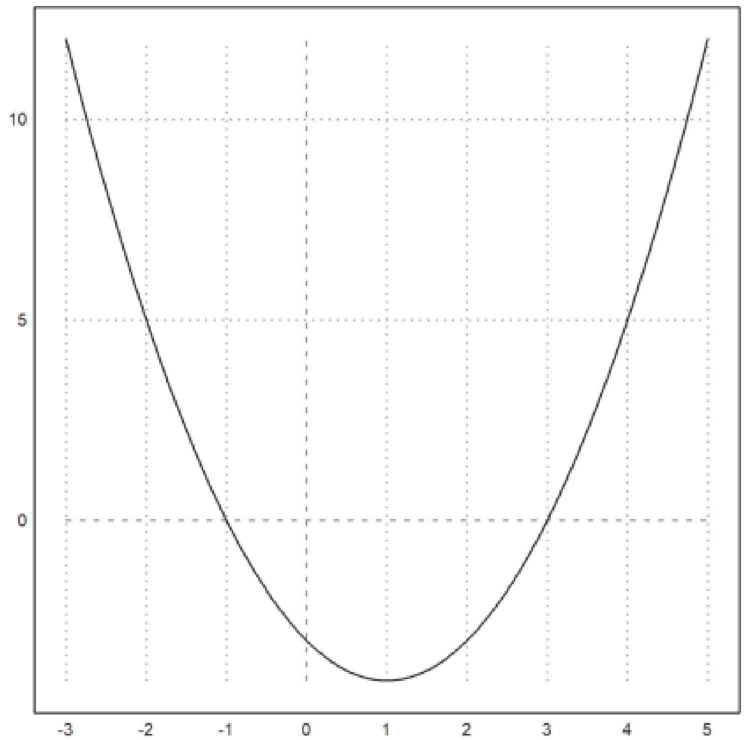
Contoh:

$$2x^2 + 1$$

$$3x^2 + 6x - 3$$

Selanjutnya akan digambarkan grafik fungsi kuadrat dengan EMT

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



Secara manual, dalam menggambar grafik fungsi kuadrat kita mencari terlebih dahulu titik puncak dan titik potong sumbu x serta sumbu y.
- Titik potong

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

$$x = \frac{-(-2)}{2(1)}$$

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

$$x = 1$$

Substitusi x=1 ke dalam persamaannya

$$y = (1)^2 - 2(1) - 3$$

$$y = 1 + -2 - 3$$

$$y = -4$$

Jadi, diperoleh titik potongnya yaitu pada titik (1,-4)

- Titik potong sumbu x dan sumbu y
Titik potong sumbu x (y=0)

```
>$&(x^2-2*x-3), $&factor(x^2-2*x-3), $&solve(x^2-2*x-3)
```

$$[x = 3, x = -1]$$

Jadi, diperoleh titik potong sumbu x yaitu di titik (3,0) dan (-1,0)

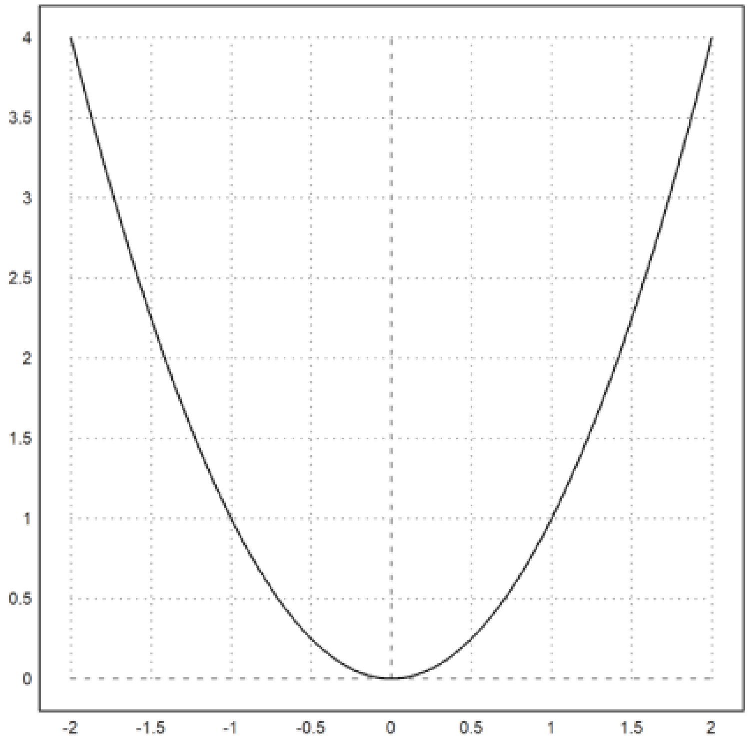
Titik potong sumbu y (x=0)

$$y = 0^2 - 2(0) - 3$$

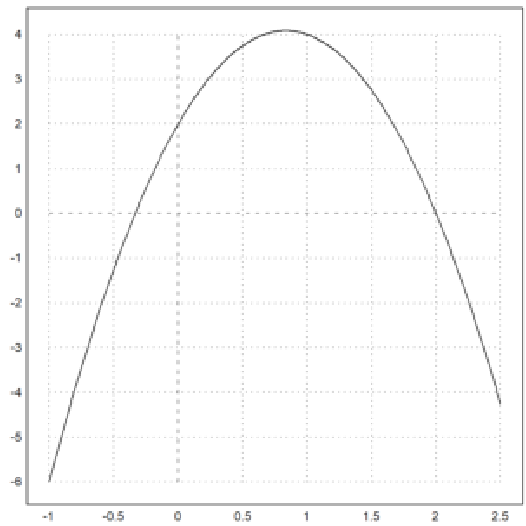
$$y = -3$$

Jadi, titik potong sumbu y yaitu di titik (0,-3)

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



```
>plot2d("-3*x^2+5*x+2",-1,2.5); insimg(20); // menampilkan grafik setinggi
```



3. Grafik Fungsi Polinomial
Fungsi Polinomial dalam variabel x adalah suatu fungsi yang memiliki bentuk umum

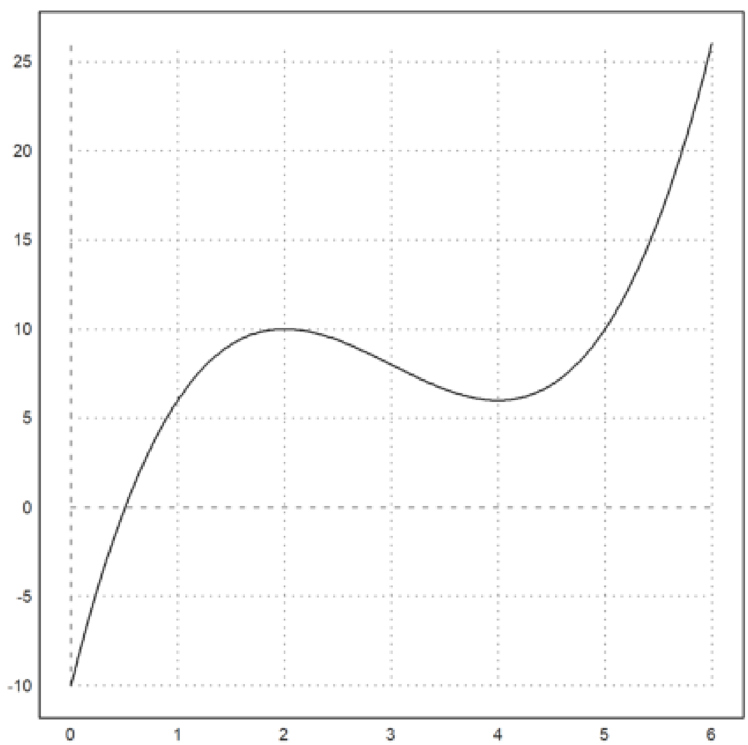
$$P(x) = a_n x^n + a_{(n-1)} x^{(n-1)} + a_{(n-2)} x^{(n-2)} + ... + a_1 x + a_0$$

dengan a adalah bilangan real, n adalah bilangan cacah, dan
a_n tidak sama dengan nol

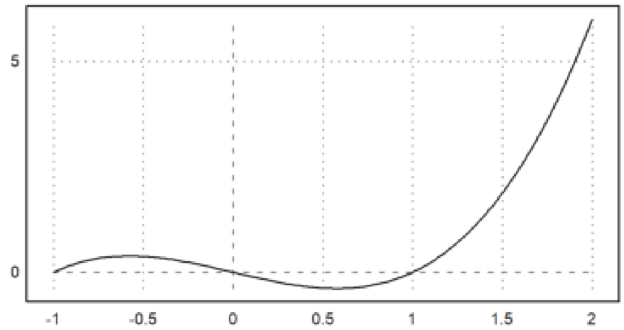
Langkah- Langkah melukis Grafik Fungsi polinom
1. Menentukan titik potong dengan sumbu X dan sumbu Y (jika mudah ditentukan)
2. Menentukan interval fungsi naik dan fungsi turun serta titik-titik stasionernya
3. Menentukan Interval cekung atas dan cekung bawah fungsi serta titik beloknya
4. Melukis sketsa grafik

Sdangkan untuk menggambar grafik fungsi polinom dengan EMT menggunakan plot2d()

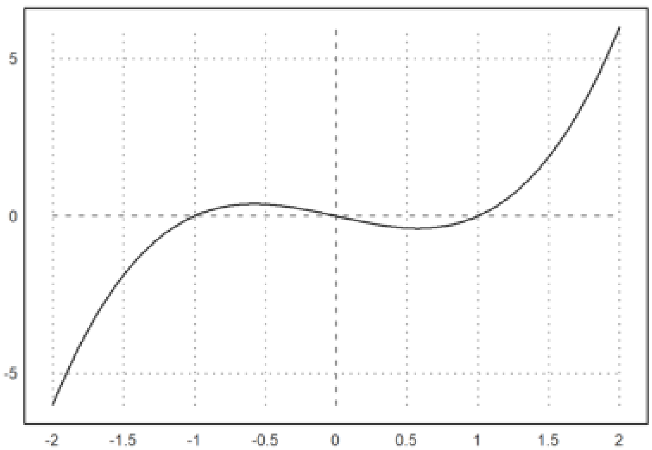
```
>plot2d("x^3-9*x^2+24*x-10", 0,6):
```




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




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



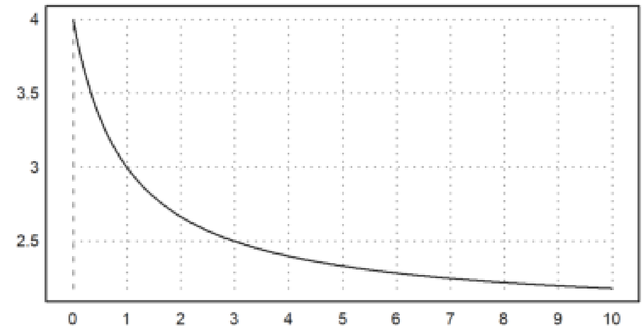
---

4. Fungsi Rasional
Fungsi rasional adalah fungsi yang memiliki bentuk

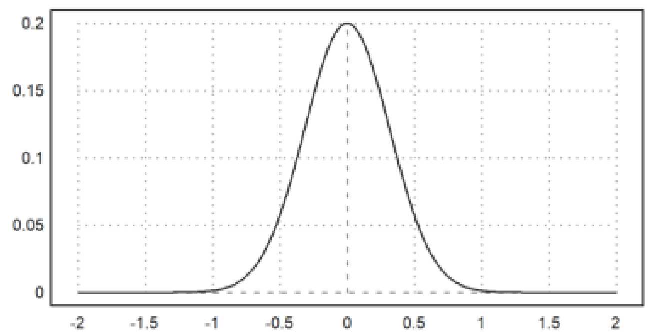
$$P(x) = \frac{a(x)}{b(x)}$$

dengan a dan b merupakan polinomial dan b(x) tidak sama dengan nol.
Domain dari P(x) merupakan seluruh bilangan real, kecuali pembuat nol dari b.

```
>aspect(2); plot2d("((2*x)+4)/(x+1)", 0,10):
```




```
>a:=5; plot2d("exp(-a*x^2)/a"):
```



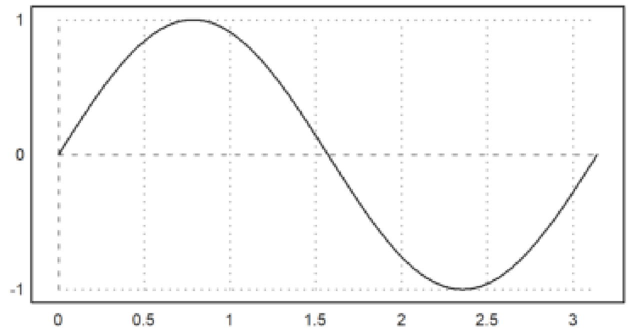
Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

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 adalah radius di sekitar pusat plot
- cx, cy: koordinat pusat plot (standar 0,0)

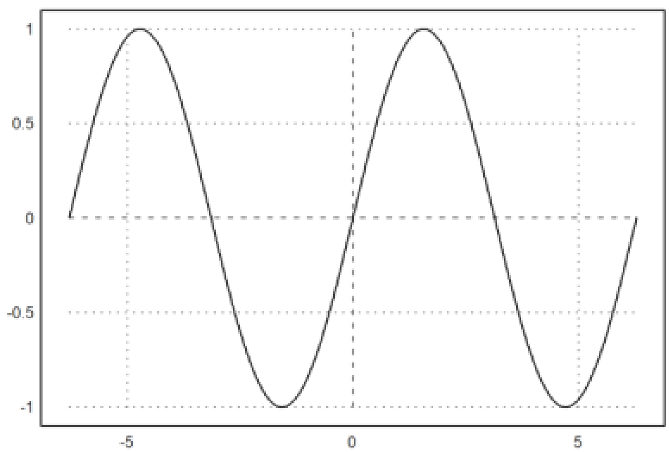
---

5. Fungsi Trigonometri
Fungsi trigonometri merupakan suatu fungsi yang melibatkan bentuk trigonometri, yaitu sinus, cosinus, tangen, cotangen, sec, cosec.
Contoh :
Akan digambarkan grafik fungsi sin2x menggunakan EMT

```
>aspect(2); plot2d("sin(2*x)", 0,pi):
```

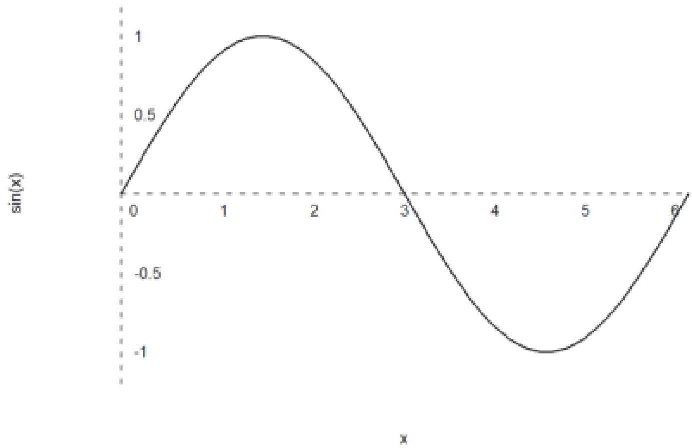


```
>plot2d("sin(x)",-2*pi,2*pi): // plot sin(x) pada interval [-2pi, 2pi]
```

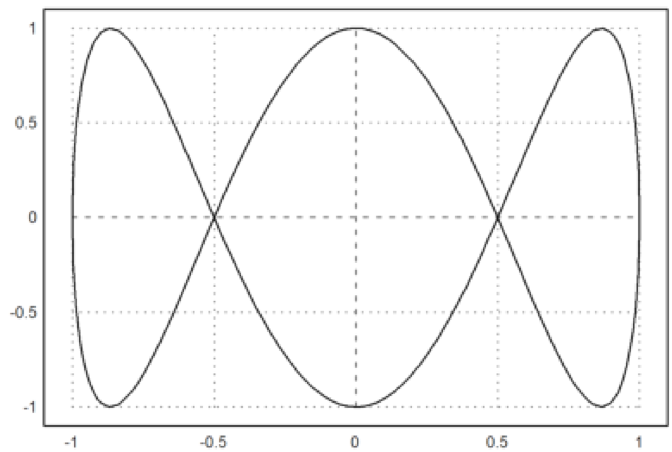


Pada contoh berikut, kita mengubah gaya kisi, menambahkan label, dan menggunakan label vertikal untuk sumbu y.

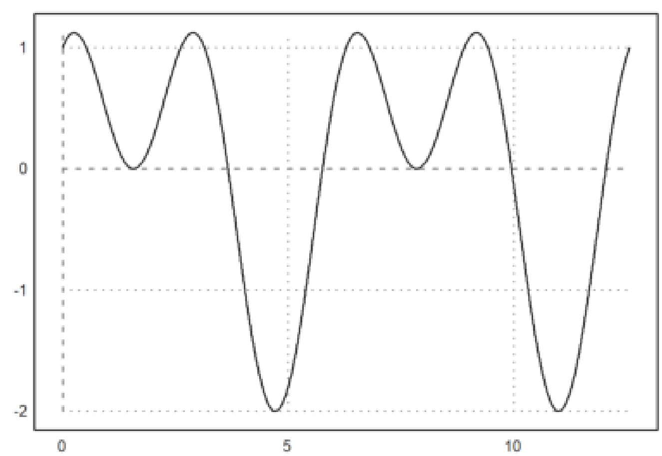
```
>aspect(1.5); plot2d("sin(x)",0,2pi,-1.2,1.2,grid=3,xl="x",yl="sin(x)"):
```




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




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



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

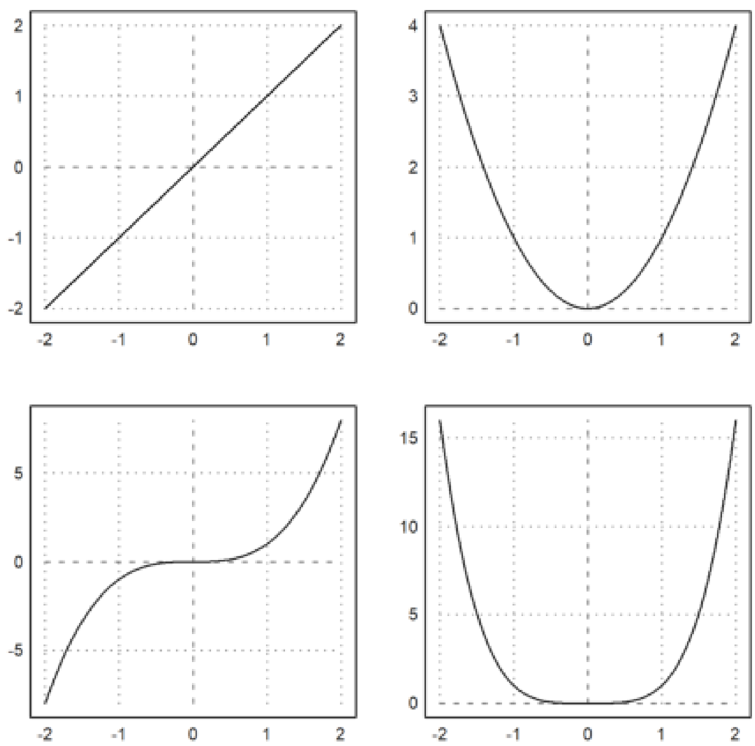
- dalam jendela terpisah yang dapat diubah ukurannya,
- di jendela buku catatan.

Lebih banyak gaya dapat dicapai dengan perintah plot tertentu.

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

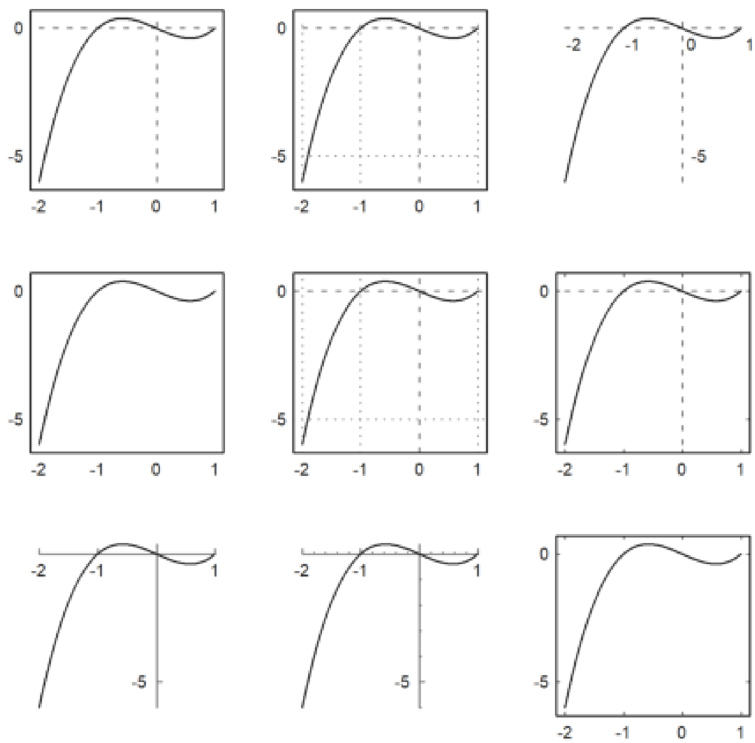
Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure(). Pada contoh, kita memplot x^1 hingga x^4 menjadi 4 bagian jendela. figure(0) mengatur ulang jendela default.

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



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):
```

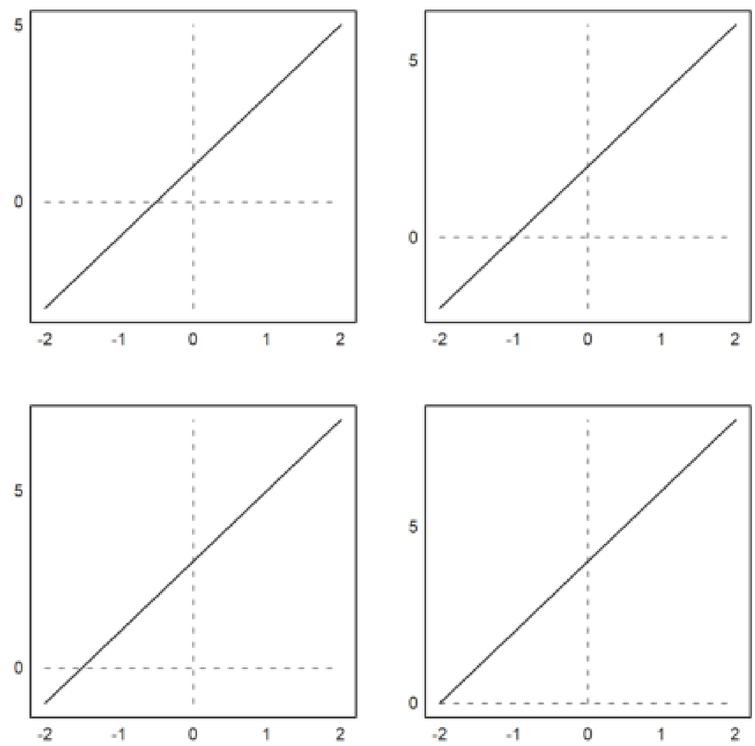


Jika argumen untuk plot2d() adalah sebuah ekspresi yang diikuti oleh empat angka, angka-angka ini adalah rentang x dan y untuk plot.

Atau, a, b, c, d dapat ditetapkan sebagai parameter yang ditetapkan sebagai a=... dst.

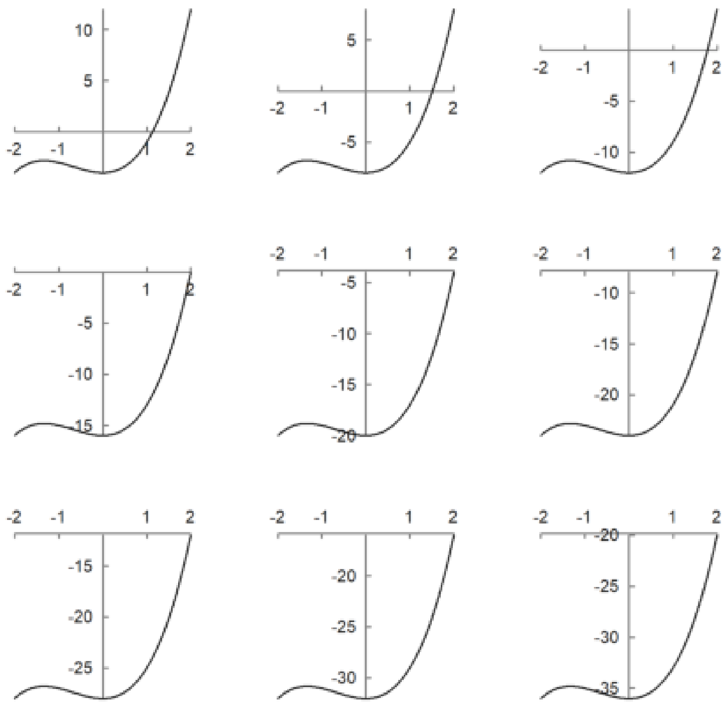
Contoh-contoh lain :

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



Gambar diatas menunjukkan grafik-grafik fungsi 2x+n, dimana n = 1,2,3,4, dengan tipe grid 1.

```
>figure(3,3);...
>for n=1 to 9; figure(n); plot2d("x^3+2*x^2-4*n", grid=7); end; ...
>figure(0):
```



Gambar diatas menunjukkan grafik-grafik fungsi

$$x^3 + 2x^2 - 4n = 0$$

dimana n = 1 sampai 9 dengan tipe grid 7 yaitu hanya menampilkan sumbu dan angka-angka pada setiap sumbu.

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

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

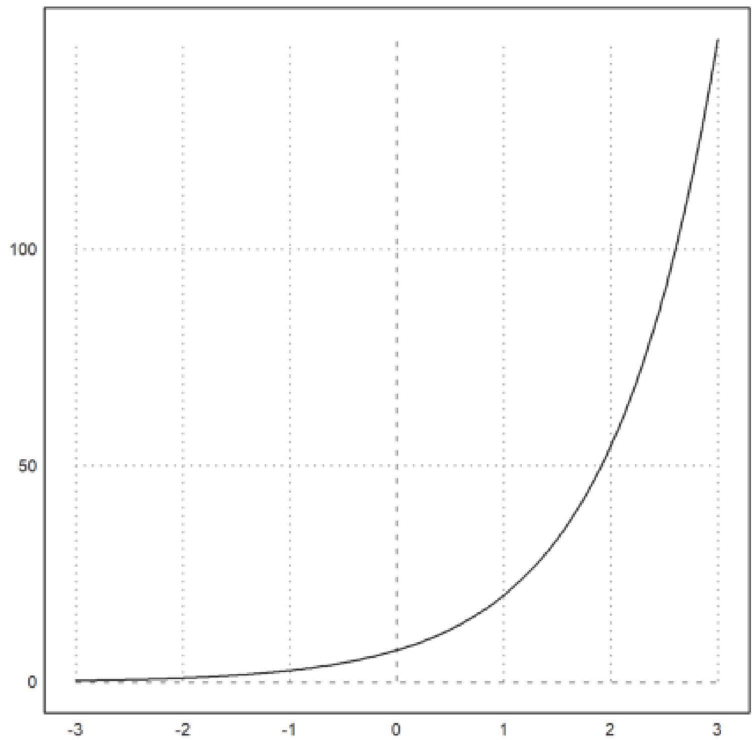
Fungsi atau ekspresi dalam plot2d dievaluasi secara adaptif. Untuk kecepatan yang lebih tinggi, matikan plot adaptif dengan <adaptive dan tentukan jumlah subinterval dengan n=... Hal ini hanya diperlukan pada kasus- kasus yang jarang terjadi.

---

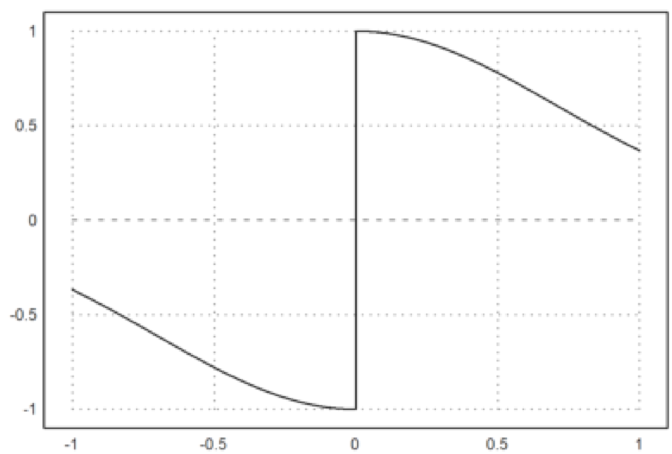
6. Fungsi Eksponensial
Fungsi eksponensial adalah fungsi yang memetakan anggota setiap x bilangan real ke f(x)=a^x dengan a=/=1 dan a>0.

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

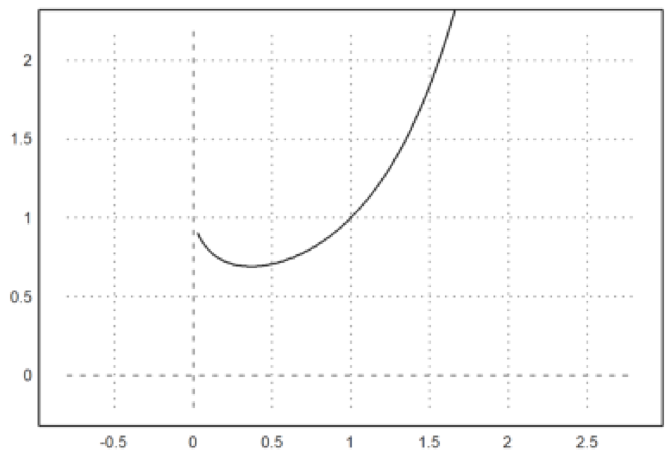
```
>plot2d("exp(x+2)", -3,3):
```




```
>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.

---

7. Fungsi Logaritma
Fungsi Logaritma Sederhana
Bentuk umum:

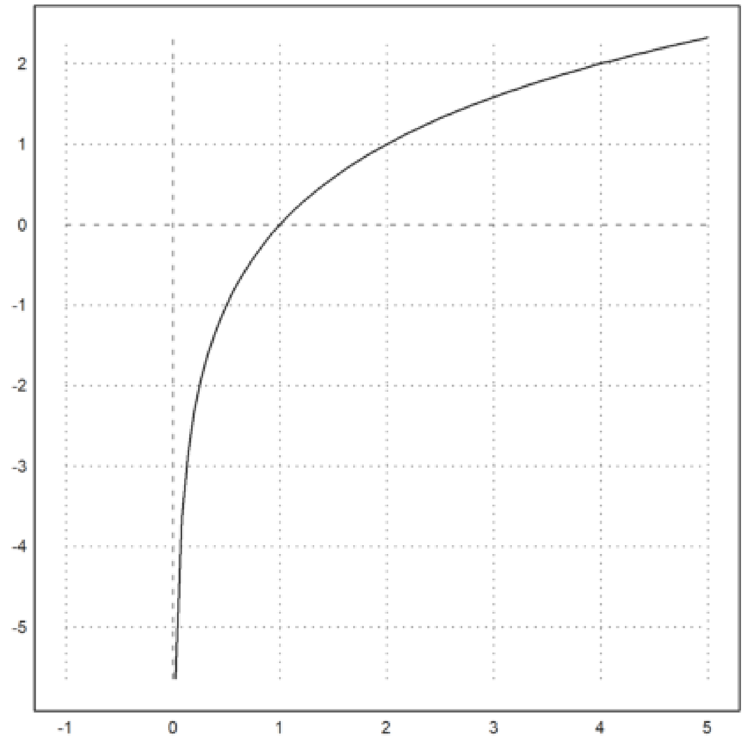
$$f(x) =^a logx$$

dengan a>0, a tidak sama dengan 1, dan x>0 serta x adalah variabel bebasnya.

Grfaik fungsi logaritma bergantung dari nilai basisnya. Jika a>1, maka grafiknya naik, sedangkan jika 0<a<1, maka grafiknya turun.

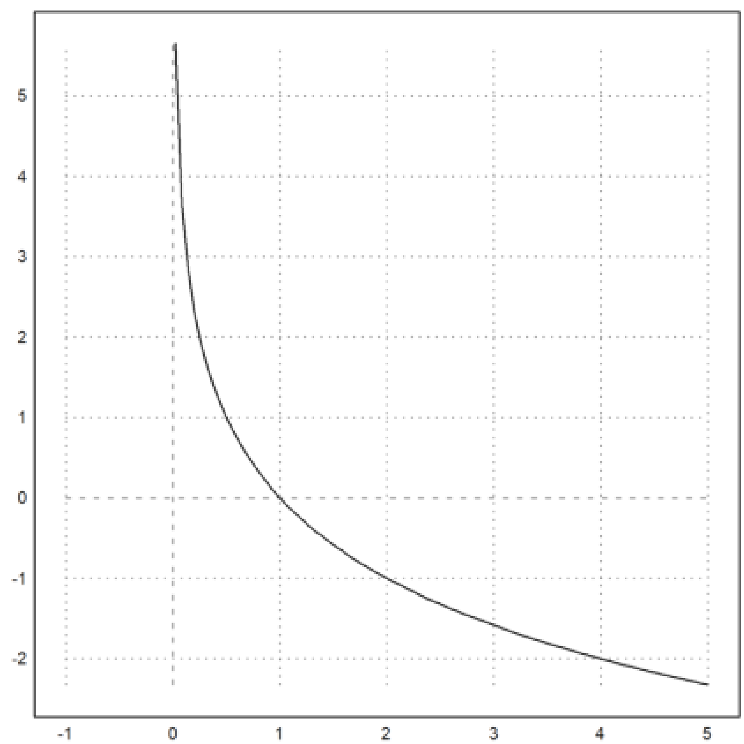
Contoh:

```
>plot2d("logbase(x,2)", -1,5):
```



Grafik naik

```
>plot2d("logbase(x,1/2)", -1,5):
```

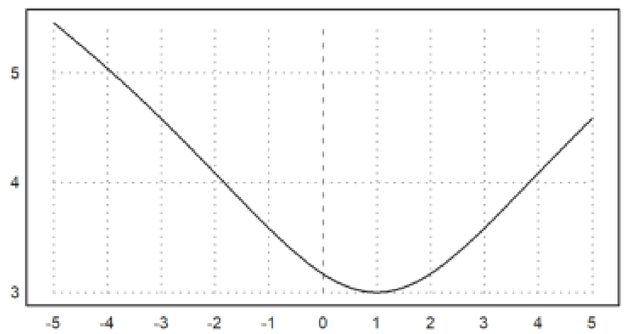


Grafik turun

Selanjutkan akan digambarkan grafik fungsi logaritma dari

$$^2log(x^2-2x+9)$$

```
>aspect(2); plot2d("logbase((x^2-2*x+9),2)", -5,5):
```



Dilihat dari gambar di atas dapat kita tentuntakn untuk titik minimumnya yaitu pada titik (1,3) dengan nilai minimum 3.

Secara manual, nilai minimumnya dapat kita cari dengan cara sebagai berikut.
- Nilai basisnya 2 (lebih dari 1), sehingga fungsi tersebut minimun ketika x^2-2x+9 juga minimum. Nilai minimum x^2-2x+9 diperoleh ketika

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

$$x = \frac{-(-2)}{2(1)}$$

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

$$x = 1$$

- Menentukan nilai minimum fungsi logaritma
substitusi x=1

$$minimum =^2 log(1^2 - 2(1) + 9)$$

$$=^2 log(8)$$

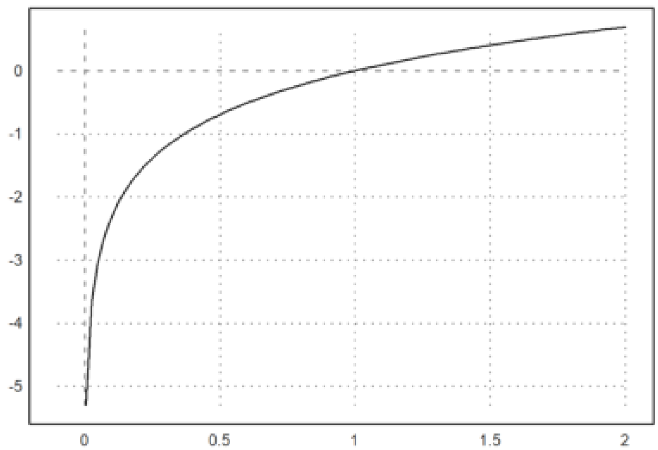
$$=^2 log(2^3)$$

$$= 3.^2 log 2$$

$$= 3.1 = 3$$

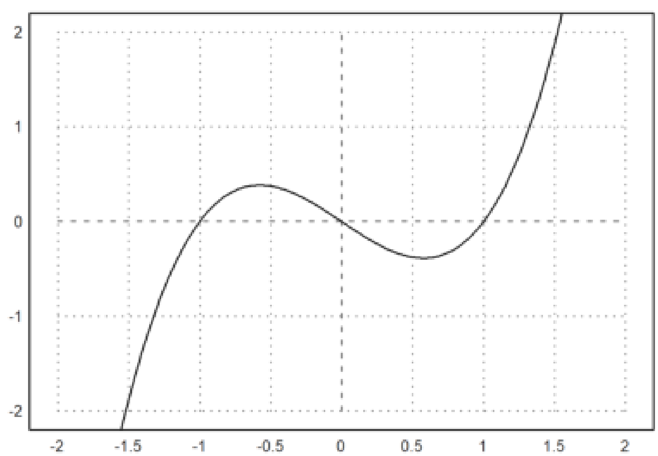
Jadi, nilai minimumnya adalah 3.

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

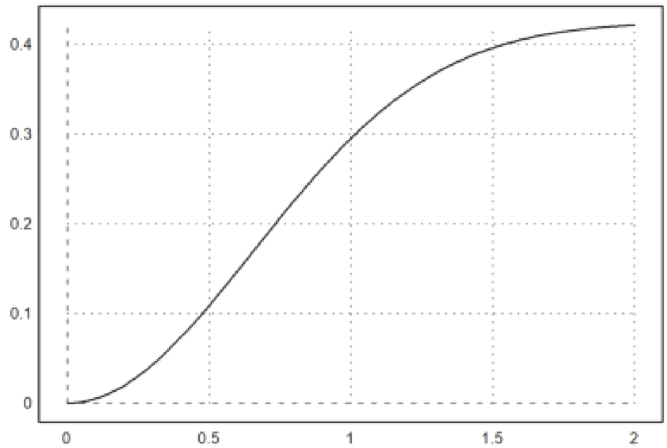


Parameter square=true (atau >square) memilih rentang y secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot.

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

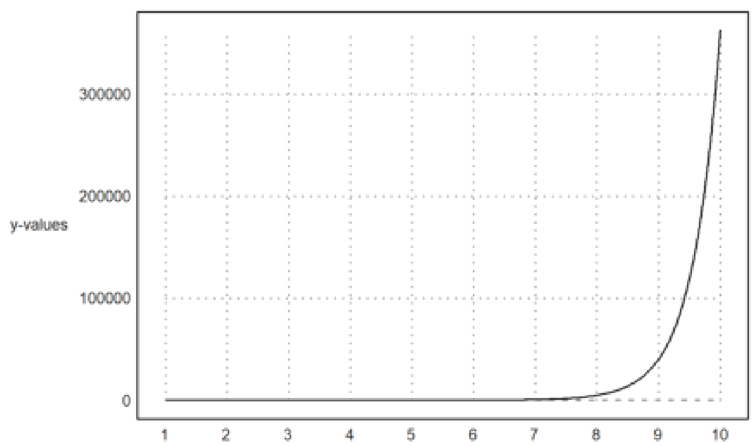


```
>plot2d(''integrate("sin(x)*exp(-x^2)",0,x)'',0,2): // plot integral
```



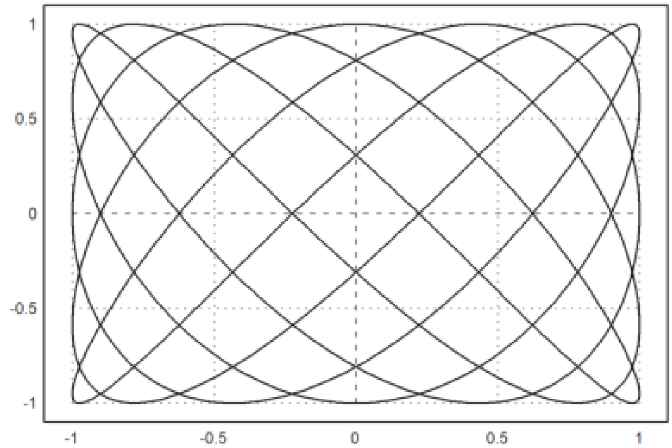
Jika Anda membutuhkan lebih banyak ruang untuk label-y, panggil shrinkwindow() dengan parameter lebih kecil, atau tetapkan nilai positif untuk "lebih kecil" pada plot2d().

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

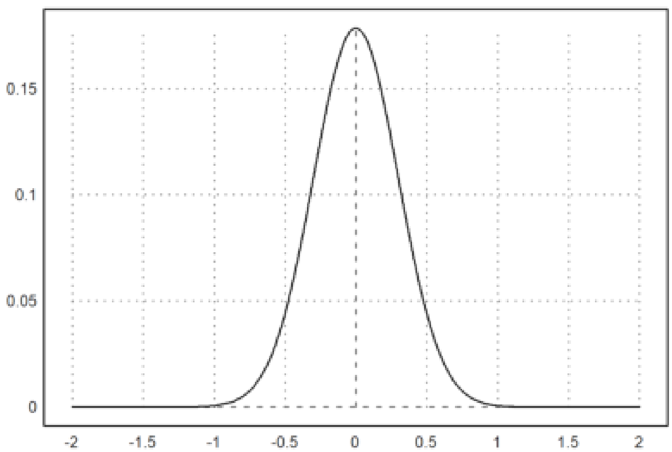


Ekspresi simbolik juga dapat digunakan, karena disimpan sebagai ekspresi string sederhana.

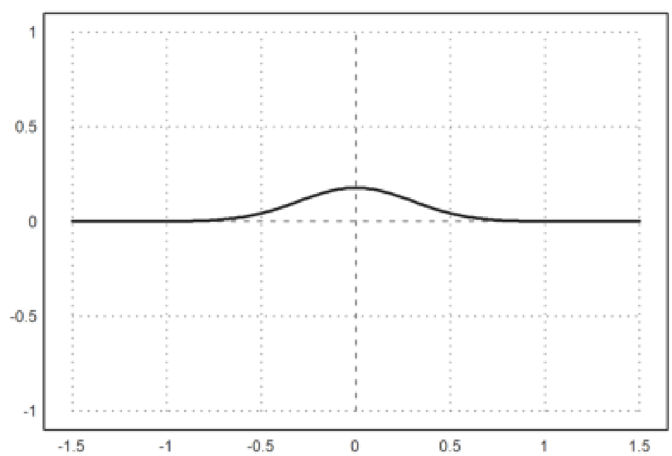
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



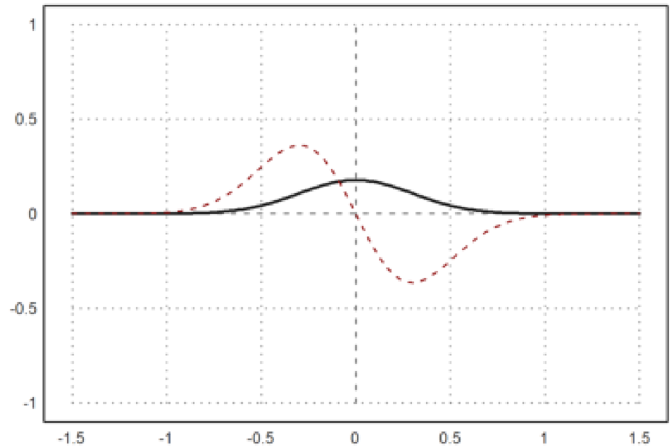

```
>a:=5.6; expr &= exp(-a*x^2)/a; //mendefinisikan ekspresi
>plot2d(expr,-2,2): // plot dari -2 to 2
```



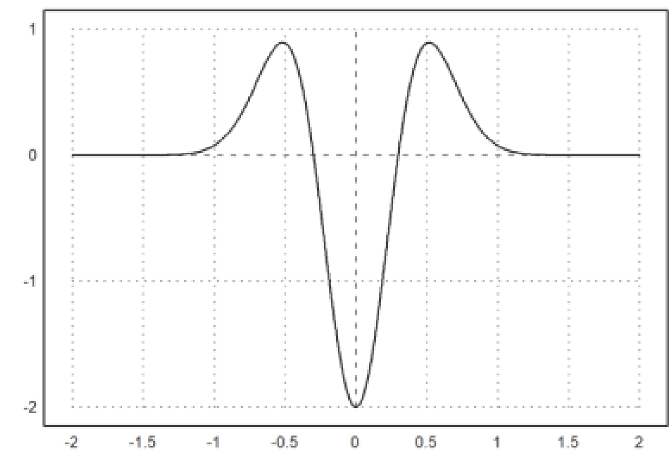

```
>plot2d(expr,r=1,thickness=2): // plot dalam kotak di sekitar (0,0)
```



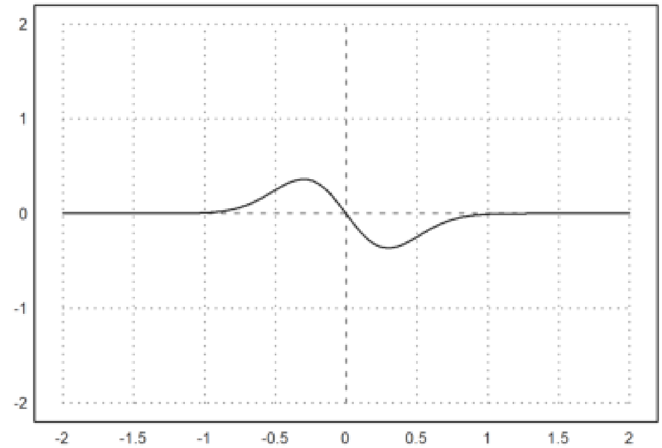
```
>plot2d(&diff(expr,x),>add,style="--",color=red): // tambahkan plot lain
```



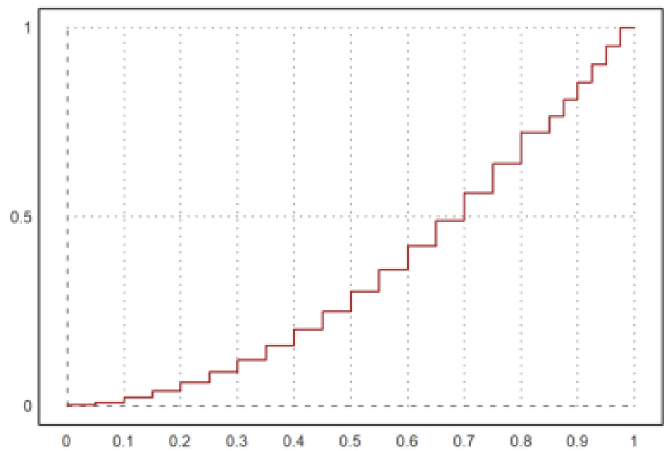

```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot dalam persegi panjang
```



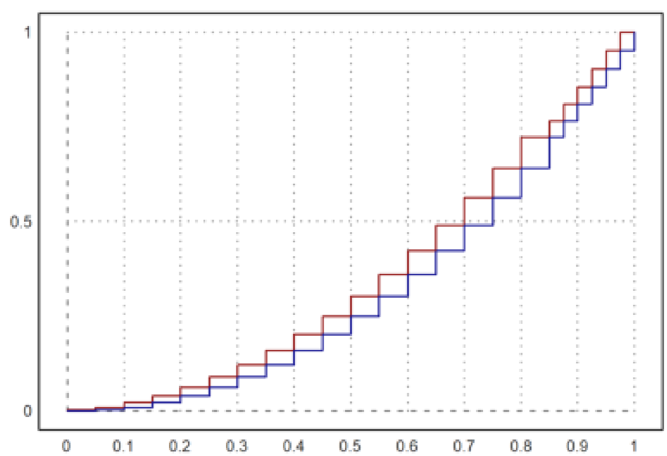

```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // pertahankan plot tetap persegi
```




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



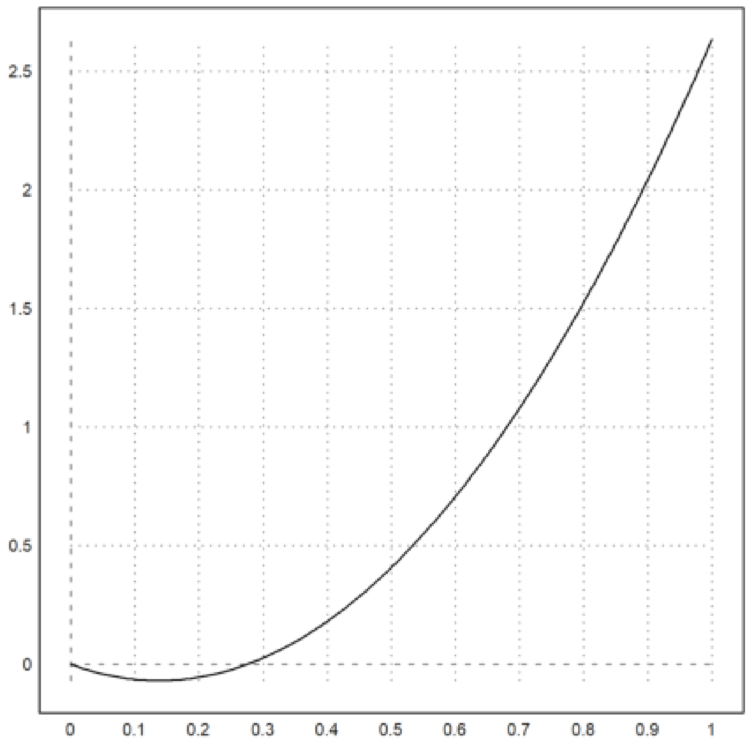

```
>plot2d("x^2",>add,steps=2,color=blue,n=10):
```



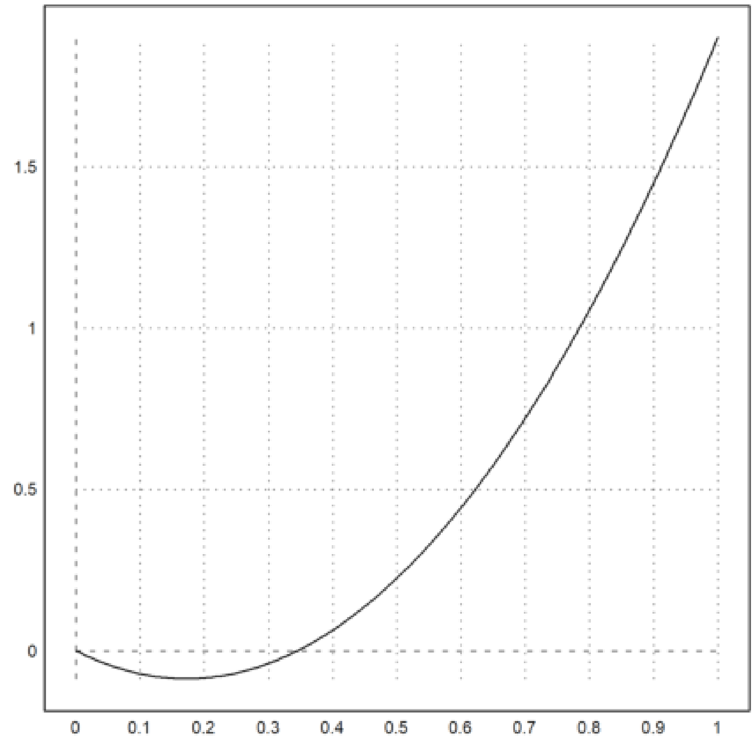
Fungsi plot yang paling penting untuk plot planar adalah plot2d(). Fungsi ini diimplementasikan dalam bahasa Euler dalam file "plot.e", yang dimuat pada awal program.

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

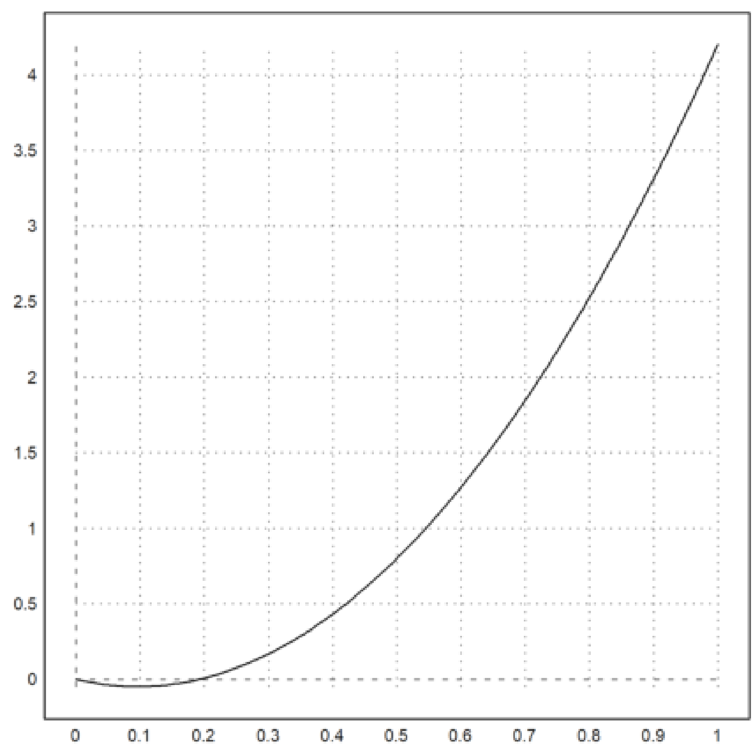
```
>function f(x,a) := x^2/a+a*x^2-x; // mendefinisikan sebuah fungsi
>a=0.3; plot2d("f",0,1;a): // plot dengan a=0.3
```



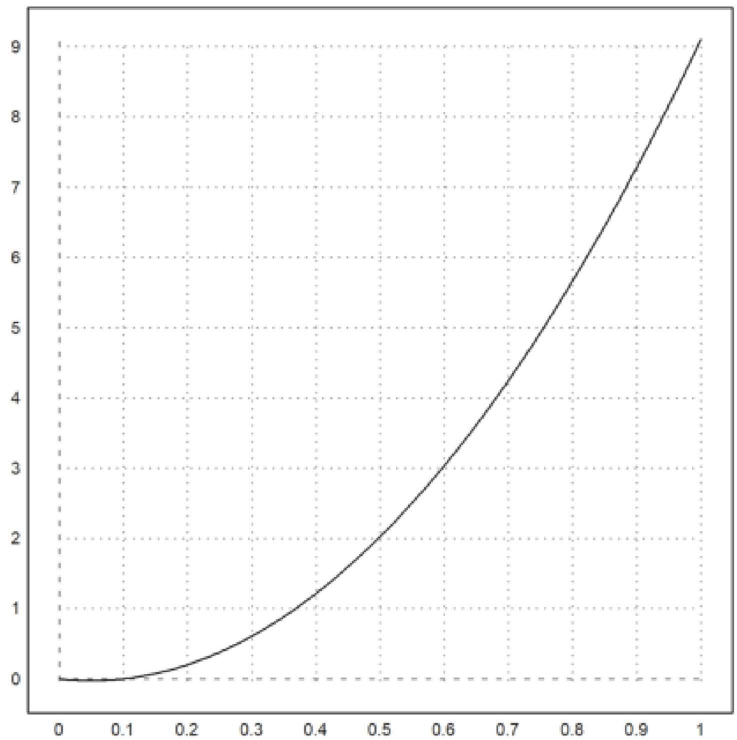

```
>plot2d("f",0,1;0.4): // plot dengan a=0.4
```



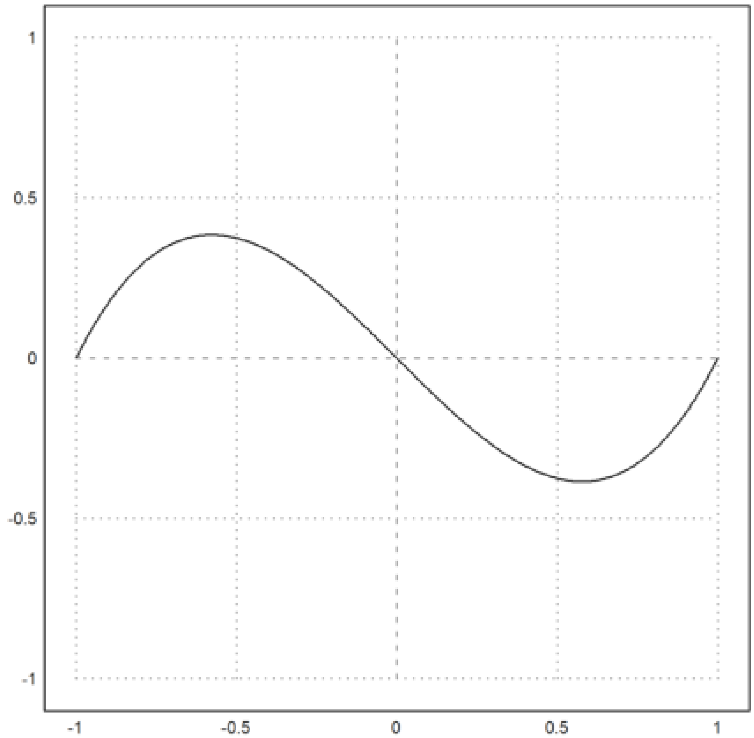

```
>plot2d({{"f",0.2}},0,1): // plot dengan a=0.2
```



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



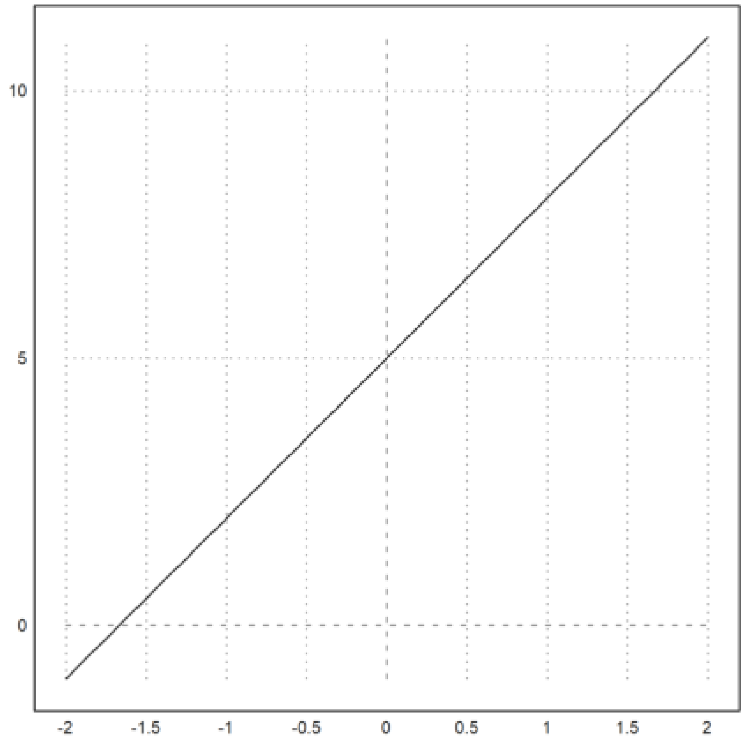

```
>function f(x) := x^3-x; ...
>plot2d("f",r=1):
```



Contoh-contoh lain :
Akan digambarkan grafik dari fungsi

$$a(x) = 3x + 5$$

```
>function a(x):= 3*x+5 // mendefinisikan fungsi a
>aspect(1); plot2d("a"): // menggambar grafik fungsi a
```



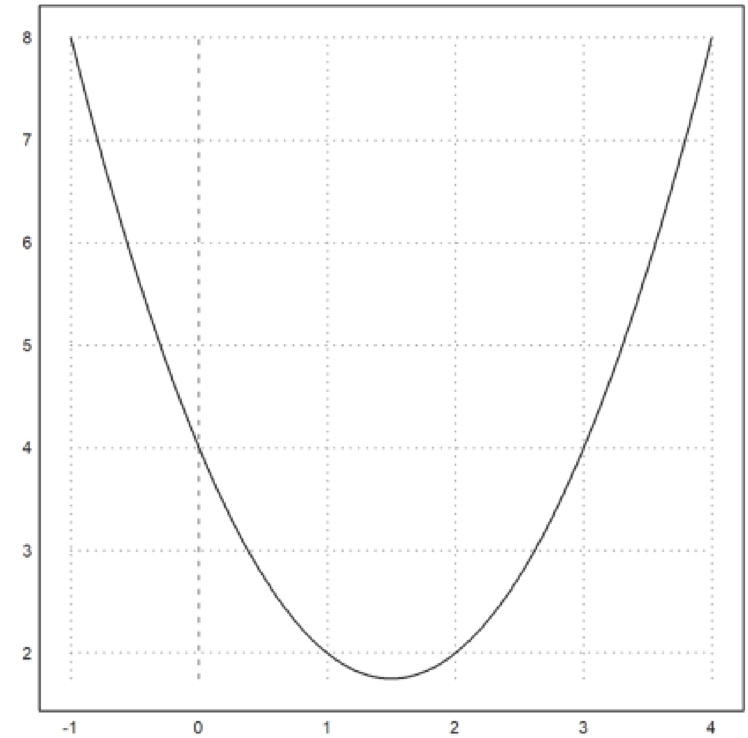
Untuk menggambar fungsi yang berbeda, definisikan fungsi dengan variabel yang berbeda (variabel tidak boleh sama).

Akan digambarkan grafik fungsi lain yaitu

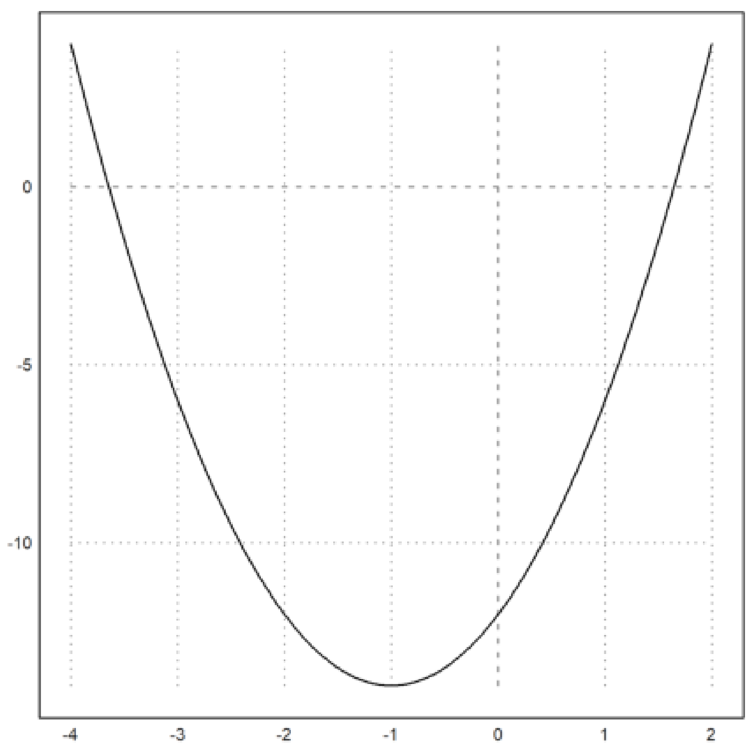
$$b(x) = x^2 - 3x + 4$$

dengan domain x yaitu -1 sampai 4

```
>function b(x):=x^2-3*x+4 // mendefinisikan fungsi b
>plot2d("b", -1,4): // menggambarkan grafik fungsi b
```



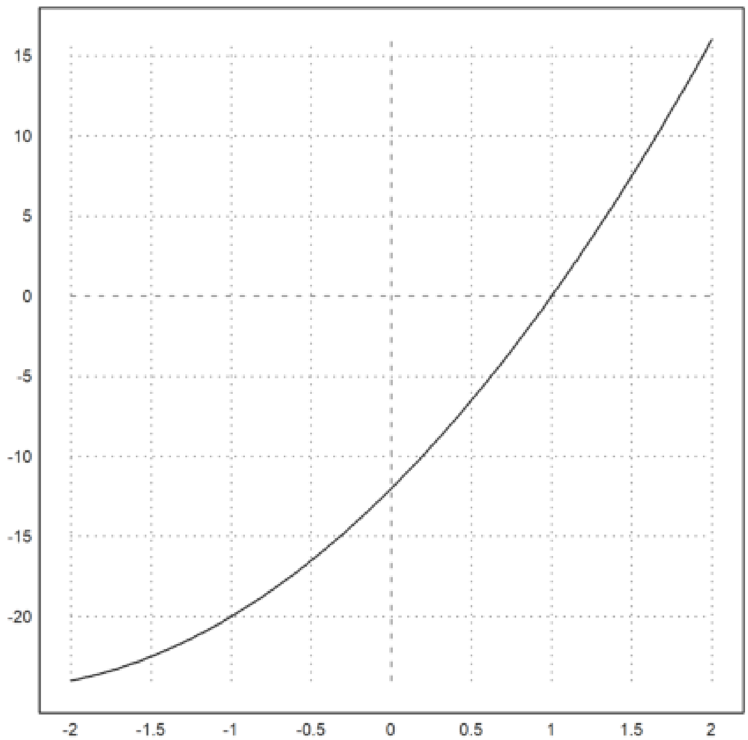

```
>function c(x,n):= 2*x^2+n*x-12
>n=4; plot2d("c", -4,2; n):
```



Grafik di atas menampilkan grafik fungsi c(x,n) untuk x dari -4 sampai 2 dan n = 4.

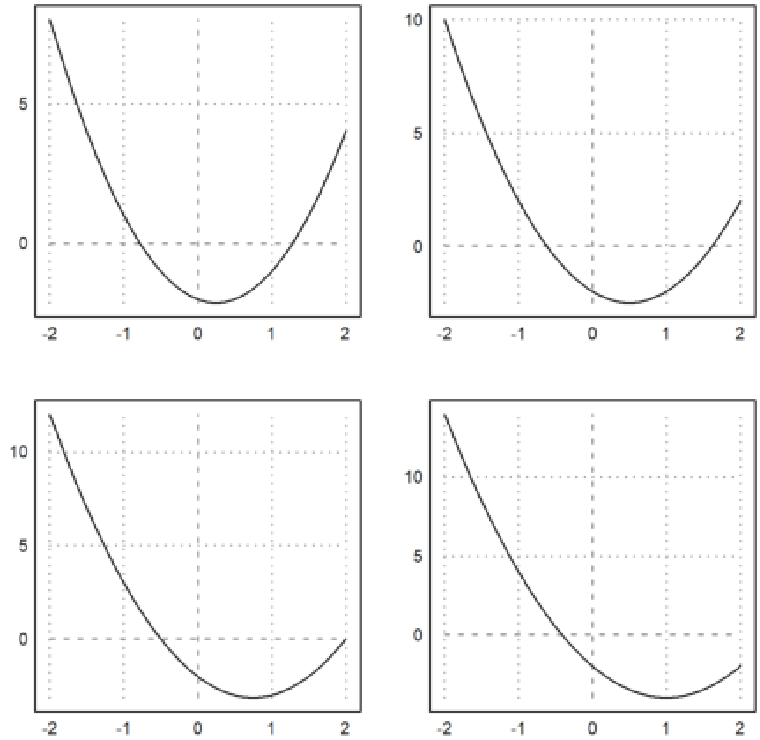
Kita bisa mengganti nilai n dengan bilangan lain, misalnya akan kita cari grafik fungsi c tersebut untuk n = 10.

```
>plot2d("c";10):
```



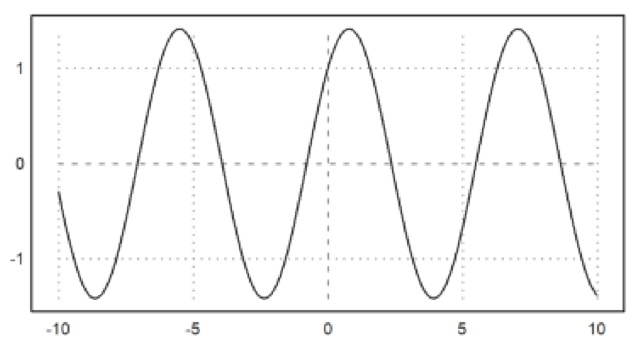
Selanjutnya, akan digambarkan beberapa kurva sekaligus untuk fungsi yang rumusnya disimpan dalam variabel ekspresi.

```
>function d(x):= 2*x^2-k*x-2
>figure(2,2); ...
>aspect(1); for k=1 to 4; figure(k); plot2d("d", -2,2); end; ...
>figure(0):
```

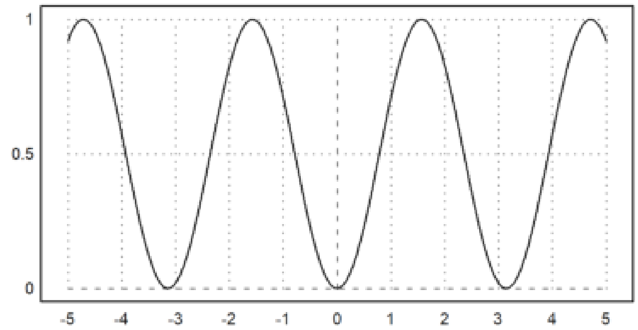


Keempat grafik diatas menggambarkan grafik fungsi d untuk k = 1 sampai 4.

```
>function m(x) := sin(x)+cos(x) // mendefinisikan fungsi m
>aspect(2); plot2d("m", -10,10): // menggambar grafik fungsi m
```



```
>function n(x) := sin(x)^2
>aspect(2); plot2d("n", -5,5):
```



Berikut ini adalah ringkasan dari fungsi yang diterima

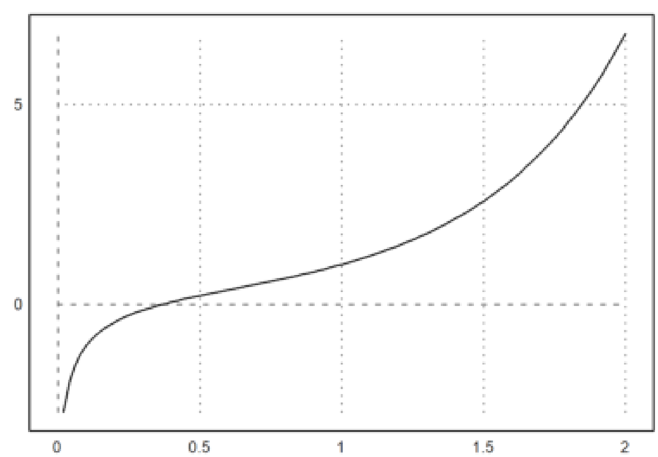
- ekspresi atau ekspresi simbolik dalam x
- fungsi atau fungsi simbolis dengan nama sebagai "f"
- fungsi simbolik hanya dengan nama f

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):
```

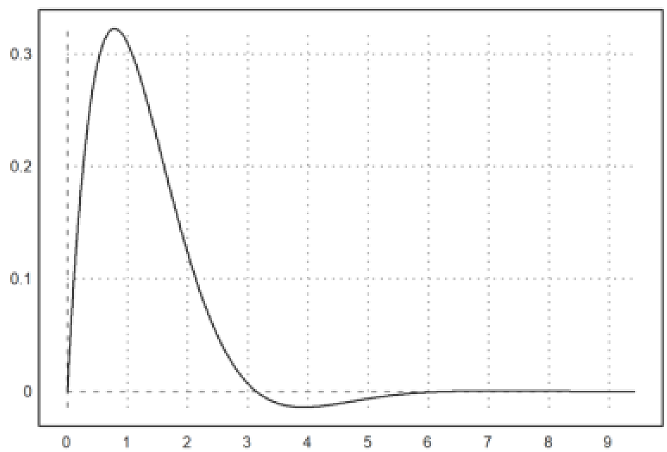


Tentu saja, untuk ekspresi atau ungkapan simbolik, nama variabel sudah cukup untuk memplotnya.

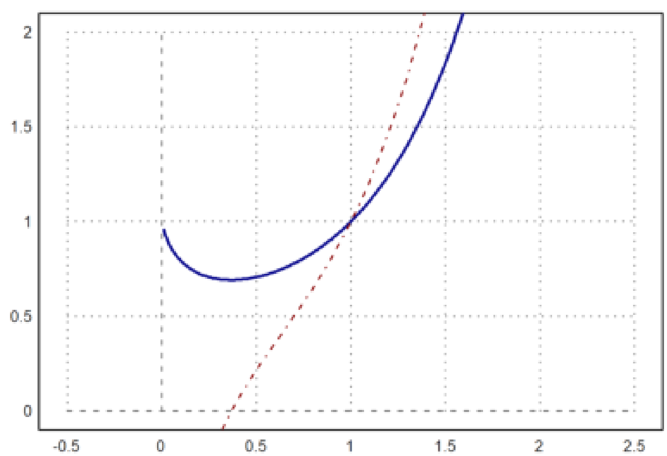
```
>expr &= sin(x)*exp(-x)
```

```
                                 - x
                                E     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="-.-"):
```



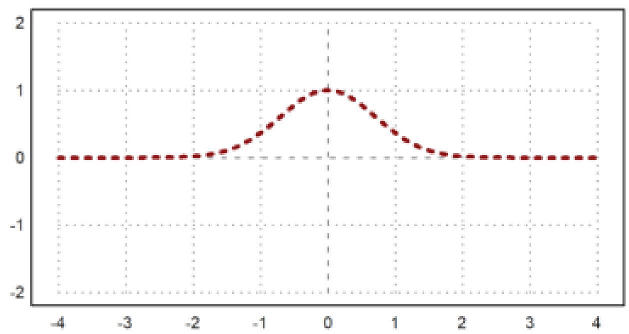
Untuk gaya garis, terdapat berbagai opsi.

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

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

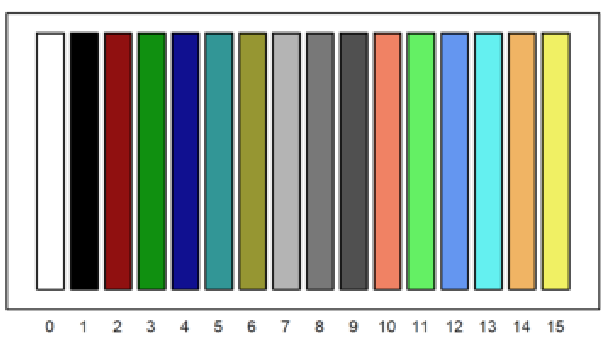
- 0..15: indeks warna default.
- konstanta warna: white, black, red, green, blue, cyan, olive, lightgray, gray, darkgray, orange, lightgreen, turquoise, lightblue, lightorange, yellow
- rgb (red, green, blue): parameter dalam bentuk real dalam [0,1].

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



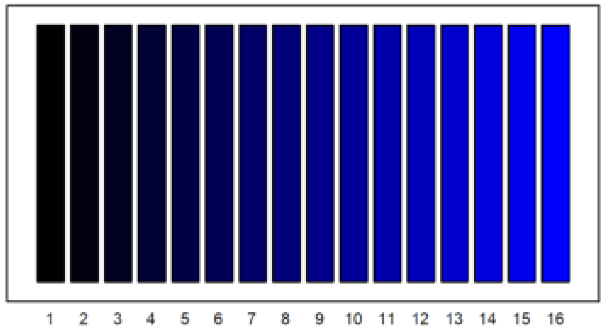
Berikut ini adalah pemandangan warna EMT yang sudah ditetapkan sebelumnya.

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



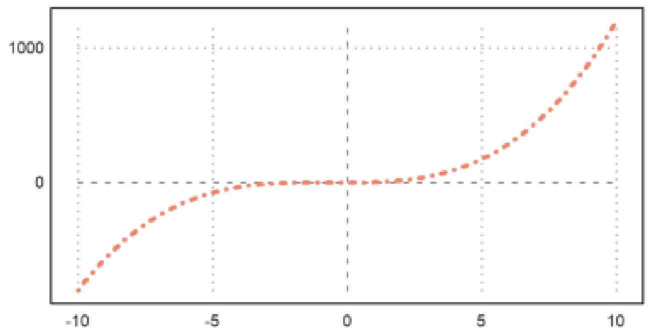
Tetapi Anda bisa menggunakan warna apa pun.

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



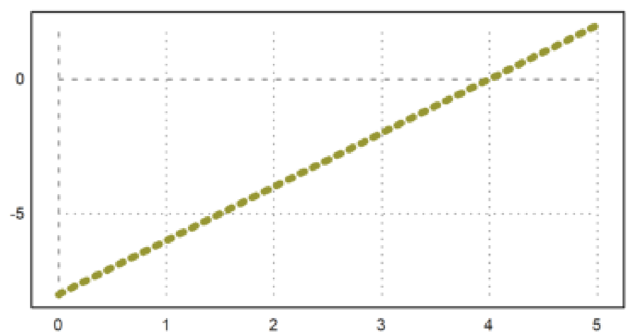
Contoh-contoh lain :

```
>function f(x) := x^3+2*x^2
>plot2d("f",-10,10, color=orange, thickness=3, style="-."):
```



Gambar di atas menenjukkan grafik fungsi f(x) dengan domain -10 sampai 10 dan garis berwarna orange dengan ketebalan 3 serta bergaya "-.".

```
>function g(x) := 2*x-8
>plot2d("g", 0,5, color=olive, thickness=5, style="--"):
```



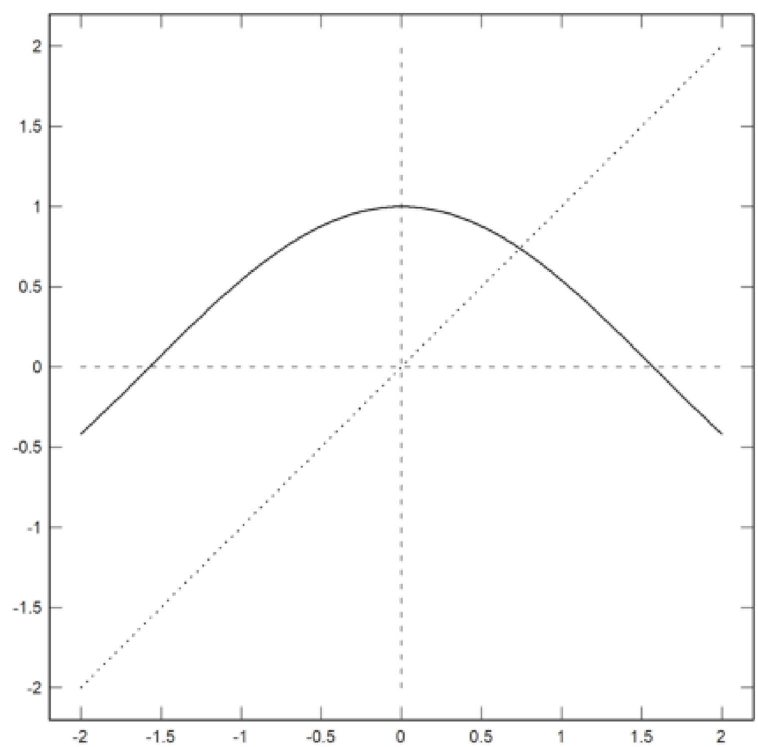
Gambar di atas menenjukkan grafik fungsi g(x) dengan domain 0 sampai 5 dan garis berwarna olive dengan ketebalan 5 serta bergaya "–".

Jadi, di EMT garis dapat kita atur warna, ketebalan, serta gayanya sesuai keinginan.

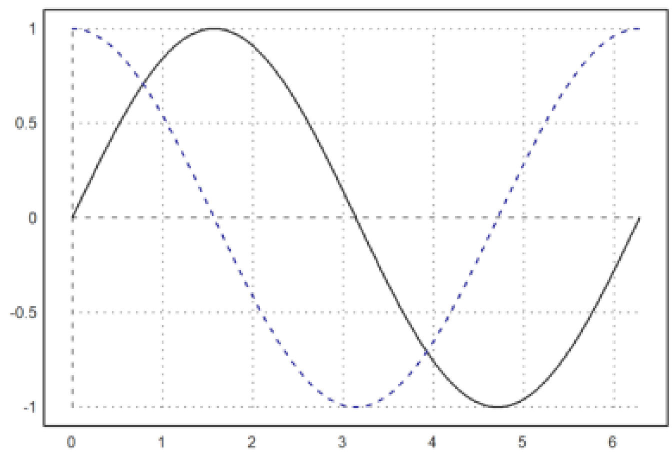
Menggambar Beberapa Kurva pada bidang koordinat yang sama

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):
```

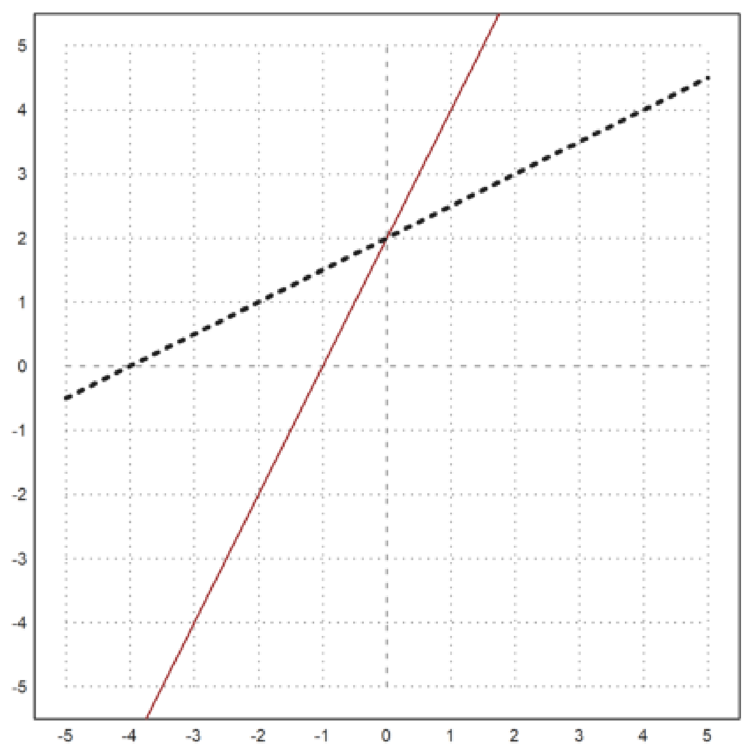



```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--"
```



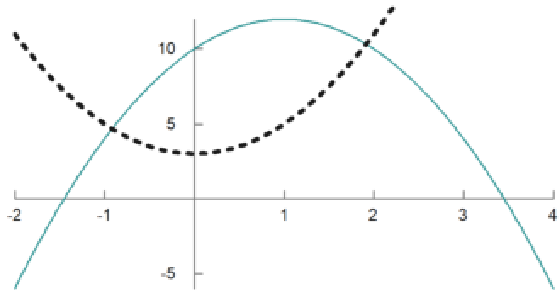
Contoh-contoh lain :

```
>aspect(); plot2d("2*x+2", r=5, color=red, grid=2); plot2d("1/2*x+2", thick
```

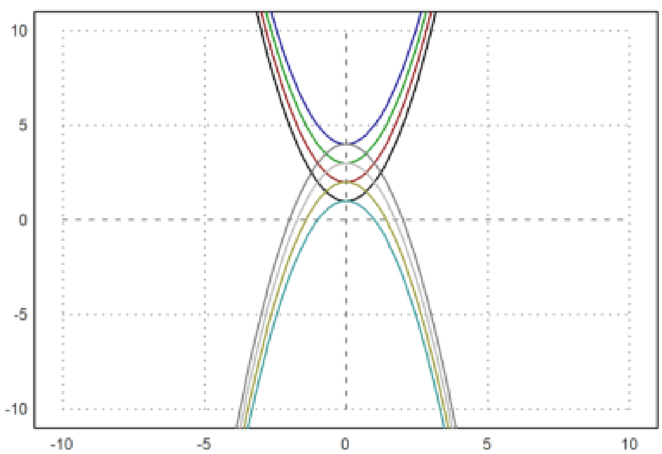


Gambar di atas menggambarkan dua buah garis pada satu bidang koordinat. Dapat dilihat bahwa kedua garis tersebut saling berpotongan.

```
>aspect(2); plot2d("-2*x^2+4*x+10", -2,4, color=cyan, grid=7); plot2d("2*x^
```

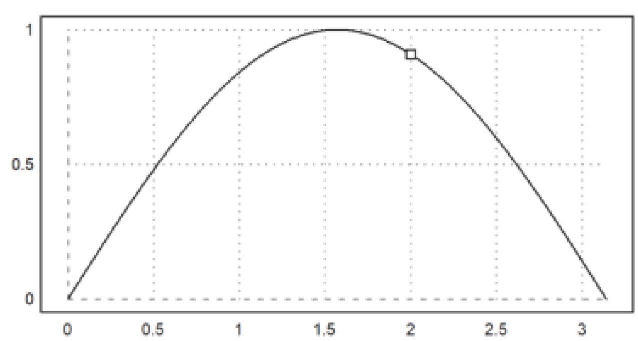



```
>aspect(1.5); plot2d(["x^2+1","x^2+2","x^2+3","x^2+4"],a=-10,b=10,c=-10,d=1
>plot2d(["-x^2+1","-x^2+2","-x^2+3","-x^2+4"],a=-10,b=10,c=-10,d=10,color=5
```

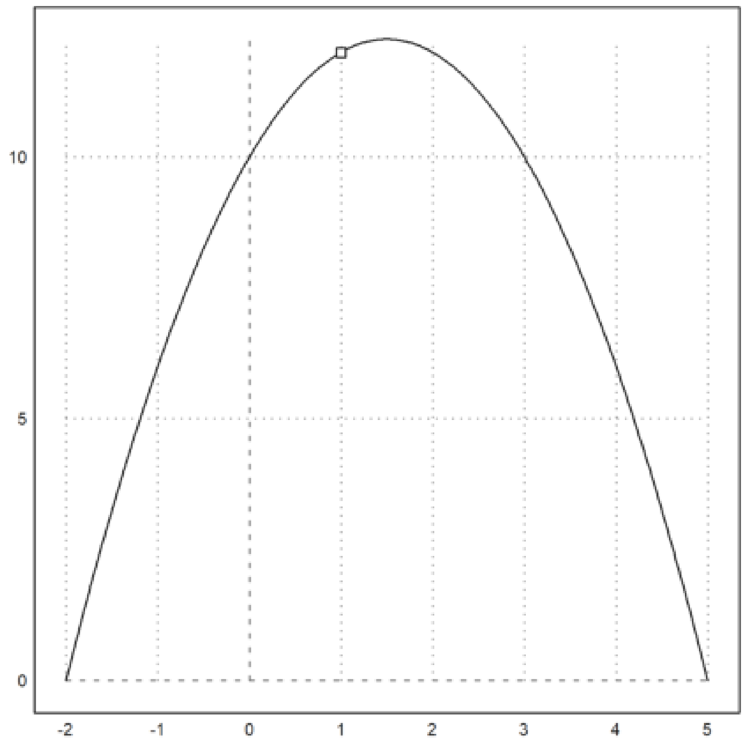


Salah satu kegunaan >add adalah untuk menambahkan titik pada kurva.

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

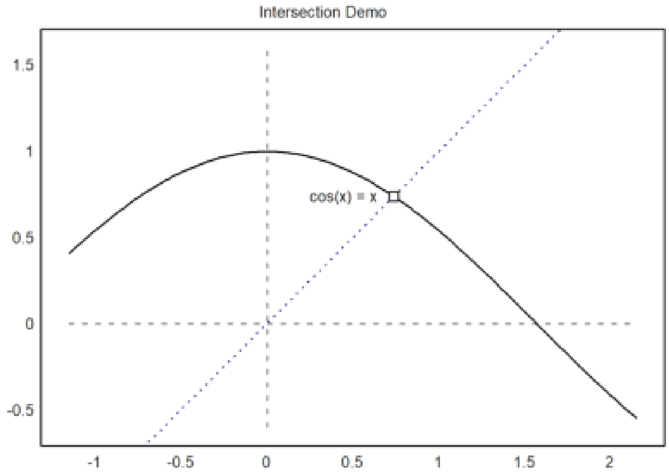


```
>aspect(1); plot2d("-x^2+3*x+10", -2,5); plot2d(12,>points,>add):
```



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):
```



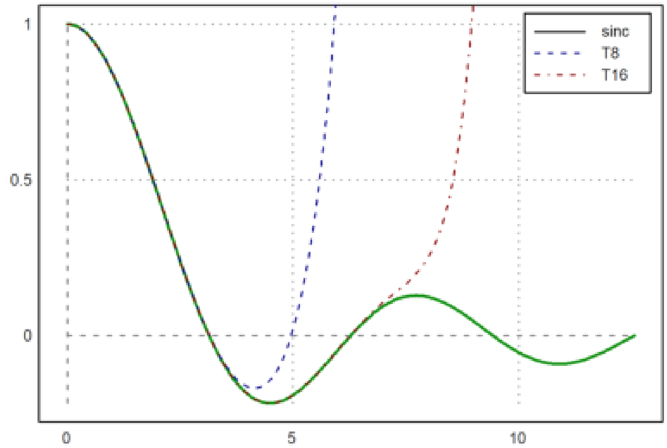
Dalam demo berikut ini, kami memplot fungsi sinc(x)=sin(x)/x dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung ekspansi ini menggunakan Maxima melalui ekspresi simbolik.
Plot ini dilakukan dalam perintah multi-baris berikut ini dengan tiga kali pemanggilan plot2d(). Pemanggilan kedua dan ketiga memiliki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

Kami menambahkan kotak label yang menjelaskan fungsinya.

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

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

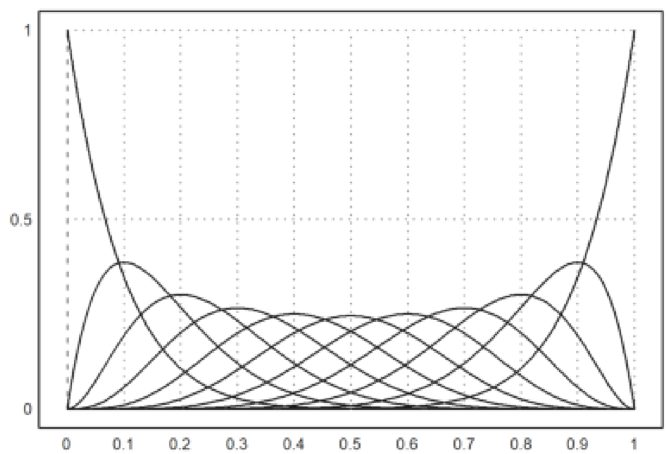
```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
>  plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
>  plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-.-"); ...
>  labelbox(["sinc","T8","T16"],styles=["-","--","-.-"], ...
>    colors=[black,blue,red]):
```



Pada contoh berikut, kami menghasilkan Polinomial Bernstein.

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

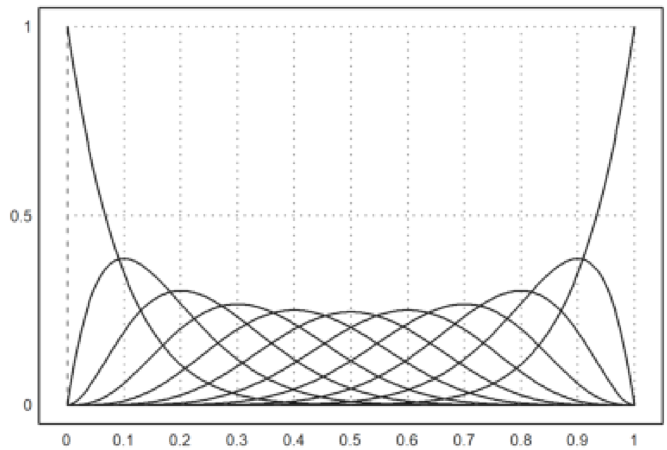
```
>plot2d("(1-x)^10",0,1); // plot fungsi pertama
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;
>insimg;
```



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

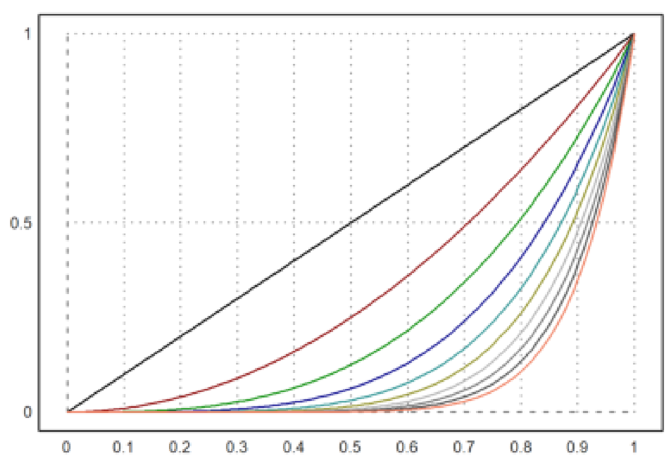
Kita membuat sebuah matriks nilai dengan satu Bernstein-Polynomial di setiap baris. Untuk ini, kita cukup menggunakan vektor kolom i. Lihatlah pengantar tentang bahasa matriks untuk mempelajari lebih lanjut.

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



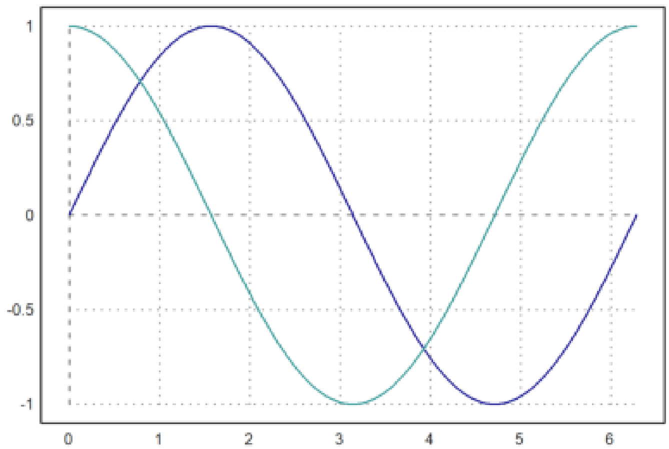
Perhatikan bahwa parameter warna dapat berupa vektor. Kemudian setiap warna digunakan untuk setiap baris matriks.

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

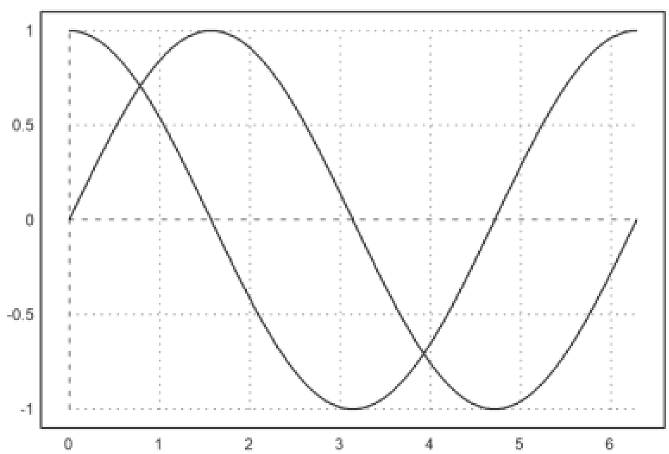


Metode lainnya adalah menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan larik warna, larik gaya, dan larik ketebalan dengan panjang yang sama.

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




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



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
```

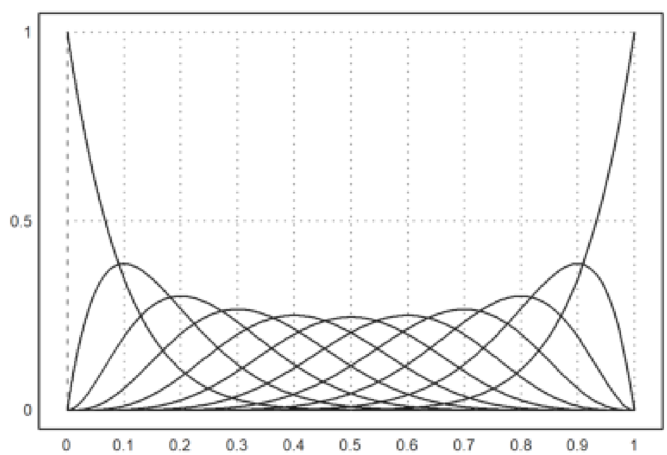
```
              10                 9                   8  2                  7  3
     [(1 - x)  , 10 (1 - x)  x, 45 (1 - x)  x , 120 (1 - x)  x ,
          6  4                  5  5                4  6                 3  7
```

```
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
```

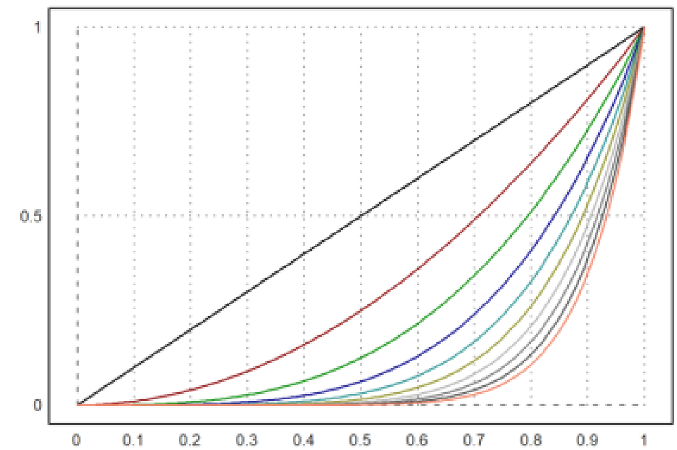


Alternatif lain adalah dengan menggunakan bahasa matriks Euler.

Jika sebuah ekspresi menghasilkan 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):
```

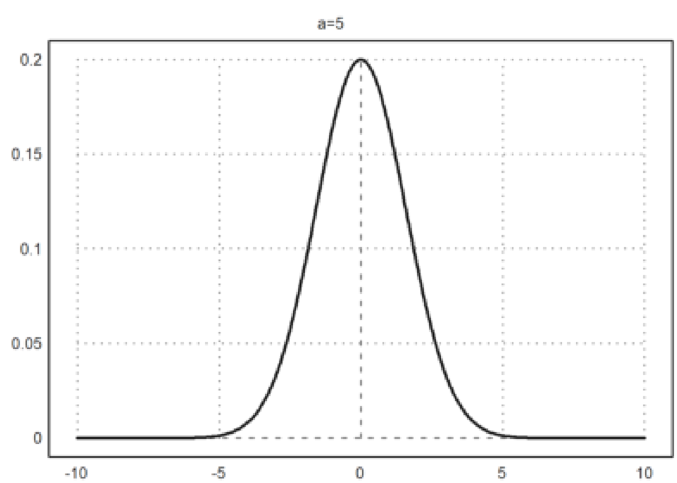


Ekspresi dan fungsi satu baris dapat melihat variabel global.

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

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

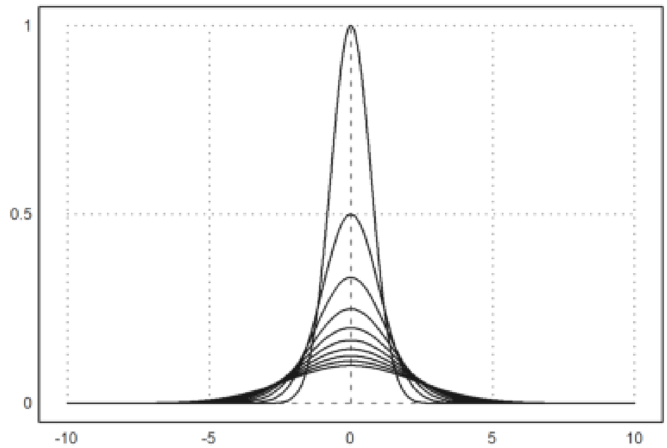
```
>function f(x,a) := 1/a*exp(-x^2/a); ...
>plot2d("f",-10,10;5,thickness=2,title="a=5"):
```



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

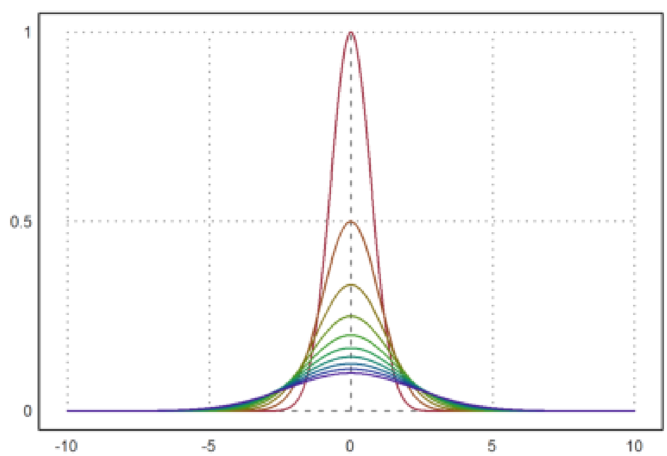
Pada contoh berikut ini, kita menggunakan loop untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman untuk loop).

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



Kita 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)):
```

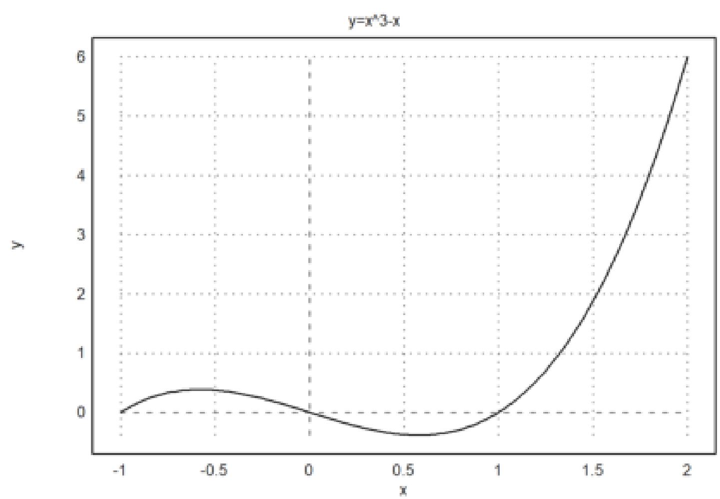


Dekorasi sederhana dapat berupa

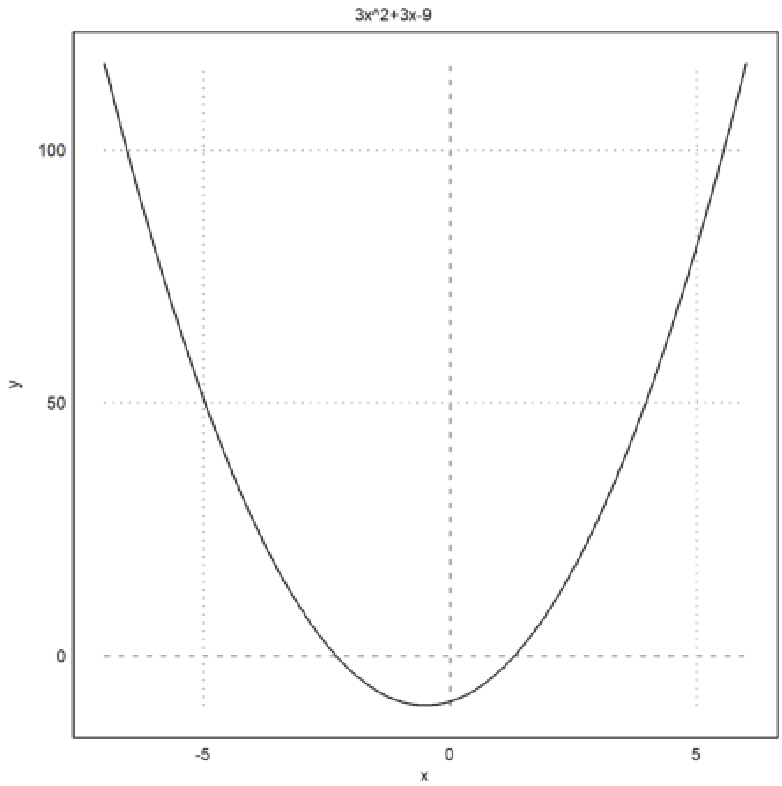
- judul dengan title = "..."
- Label x dan y dengan xl="...", yl="..."
- label teks lain dengan label("...",x,y)

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

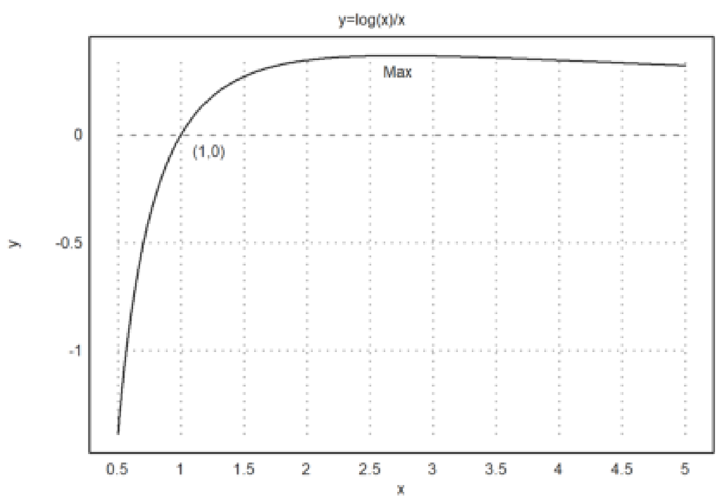
```
>plot2d("x^3-x",-1,2,title="y=x^3-x",yl="y",xl="x"):
```




```
>aspect(1); plot2d("3*x^2+3*x-9",-7,6,title="3x^2+3x-9",yl="y",xl="x"):
```

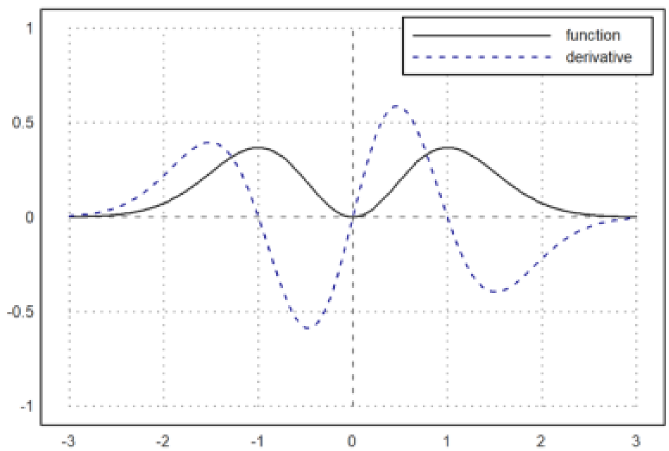



```
>expr := "log(x)/x"; ...
>  plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
>  label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```



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):
```

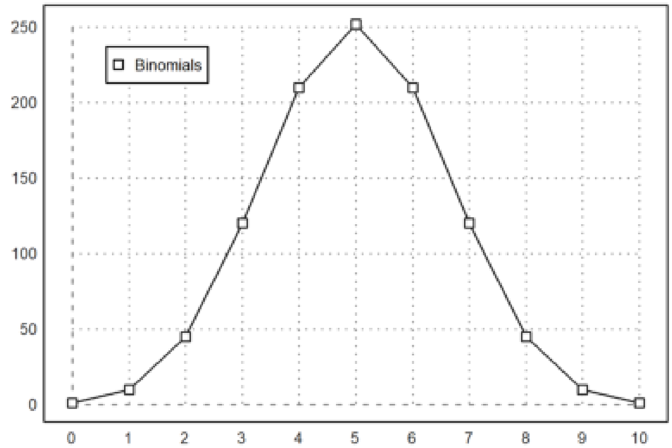


Kotak tersebut berlabuh di kanan atas secara default, tetapi >kiri menambatkannya di kiri atas. Anda dapat memindahkannya ke tempat mana pun yang Anda suka. Posisi jangkar adalah sudut kanan atas kotak, dan angkanya adalah pecahan dari ukuran jendela grafik. Lebarnya adalah otomatis.

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

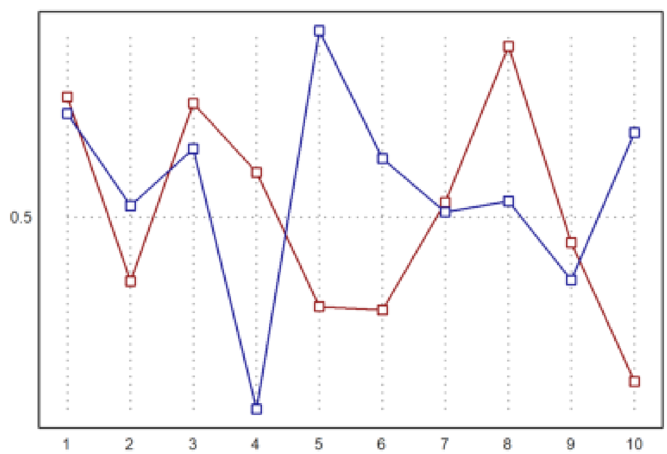
Pada contoh berikut ini, hanya ada satu fungsi. Jadi kita dapat menggunakan string dan bukan vektor string. Kami mengatur warna teks menjadi hitam untuk contoh ini.

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



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

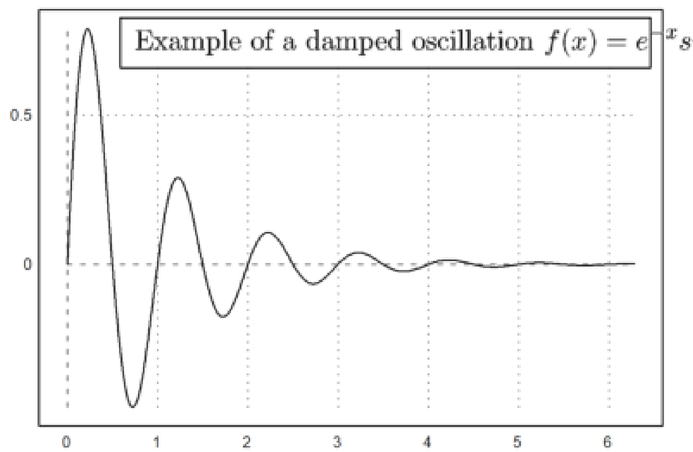
```
>statplot(1:10,random(2,10),color=[red,blue]):
```



Fitur yang serupa adalah fungsi textbox().

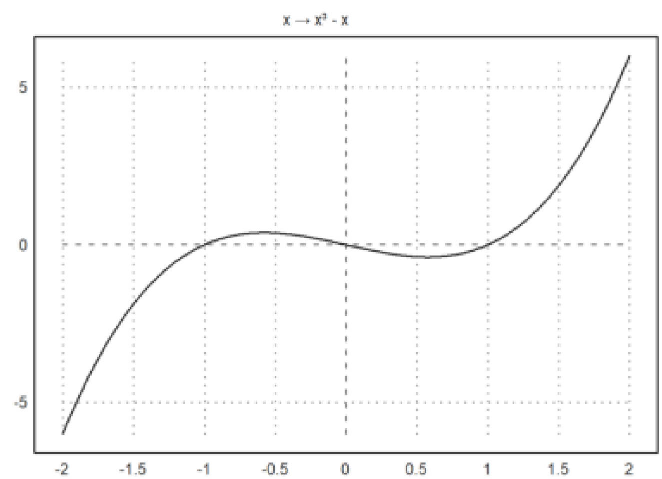
Lebarnya secara default adalah lebar maksimal baris teks. Tetapi, ini juga dapat diatur oleh pengguna.

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



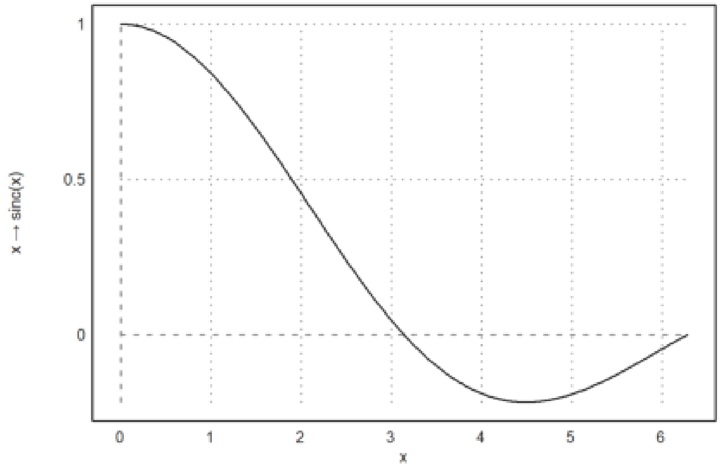
Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaks EMT untuk mengetahui lebih lanjut tentang string Unicode).

```
>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):
```



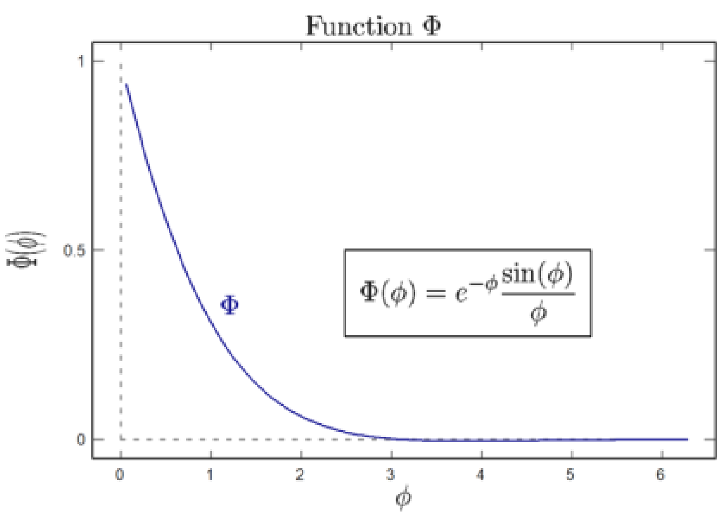
LaTeX

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.

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

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

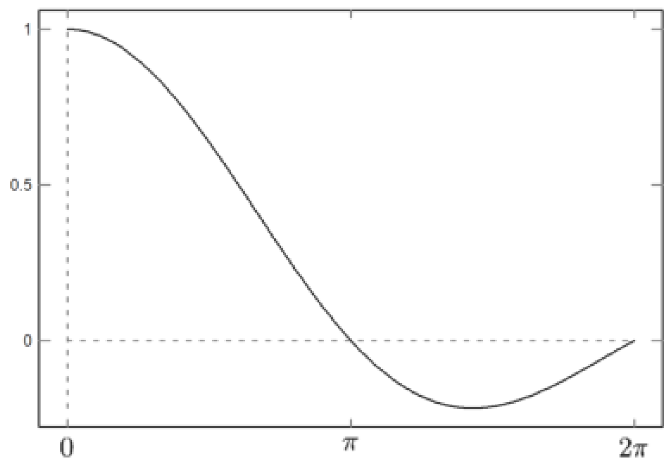
```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
>  title=latex("\text{Function $\Phi$}"), ...
>  xl=latex("\phi"),yl=latex("\Phi(\phi)")); ...
>textbox( ...
>  latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}"),x=0.8,y=0.5); ...
>label(latex("\Phi",color=blue),1,0.4):
```



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.

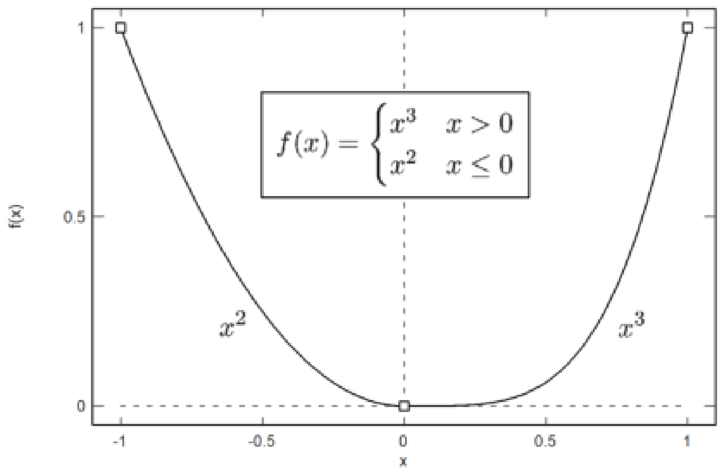
```
>function map f(x) ...
```

```
  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 menunjukkan 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. Kami menggunakannya untuk dua label dan sebuah kotak teks. Tentu saja, Anda hanya dapat menggunakan LaTeX jika Anda telah menginstal LaTeX dengan benar.

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



Interaksi Pengguna

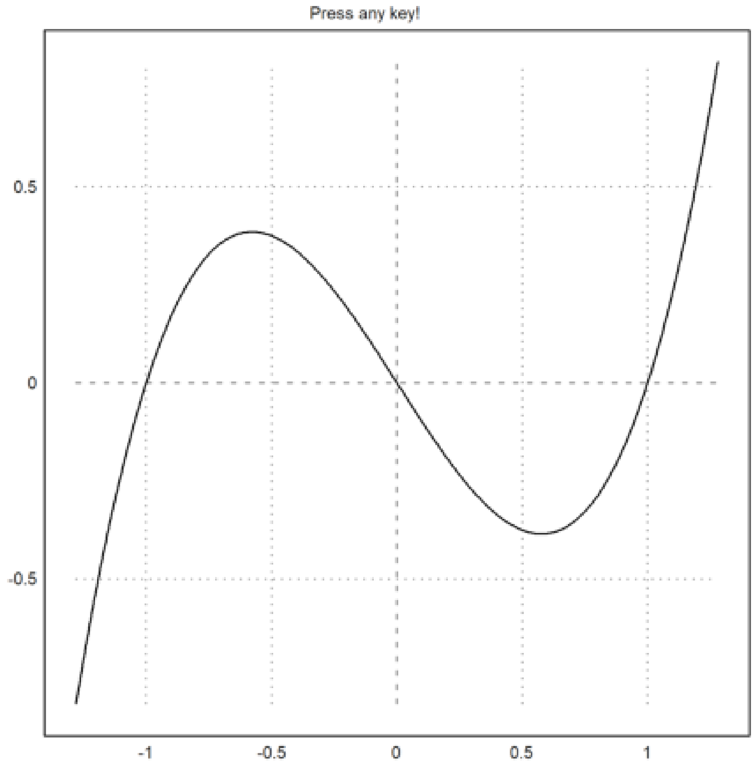
Ketika memplot fungsi atau ekspresi, parameter >user memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kursor atau mouse. Pengguna dapat

- zoom dengan + atau -
- memindahkan plot dengan tombol kursor
- pilih jendela plot dengan mouse
- mengatur ulang tampilan dengan spasi
- keluar dengan kembali

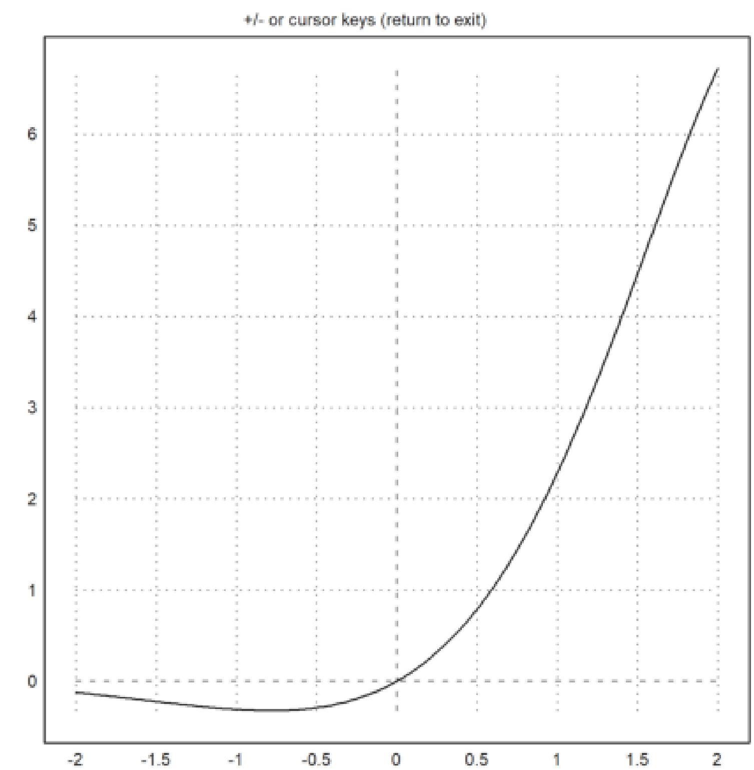
Tombol spasi akan mengatur ulang plot ke jendela plot asli.

Saat 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 mengenai 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 dalam 5 titik dengan polinomial. Fungsi ini harus memplot ke dalam area plot yang tetap.

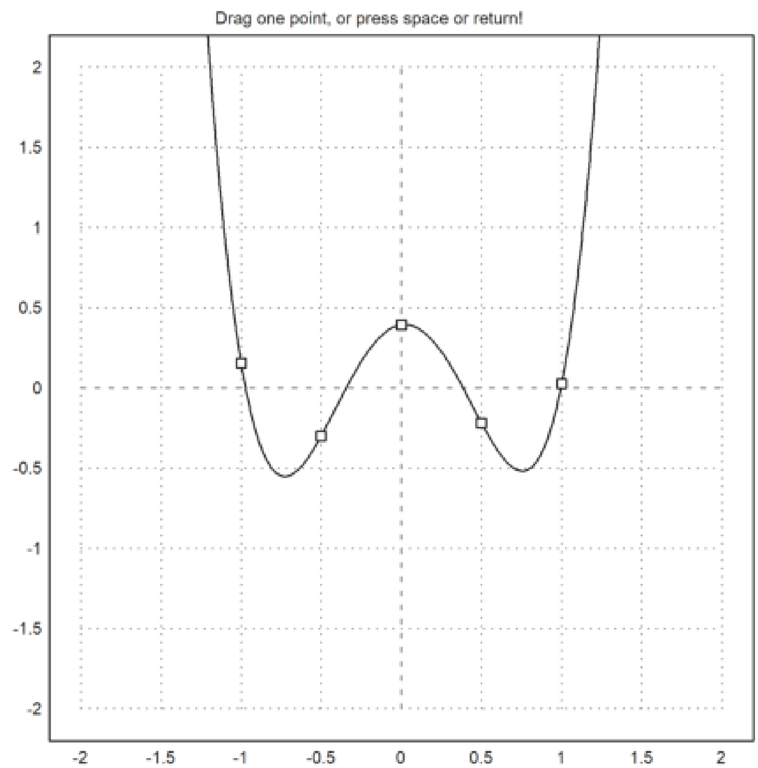
```
>function plotf(xp,yp,select) ...
```

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

Perhatikan parameter titik koma 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):
```



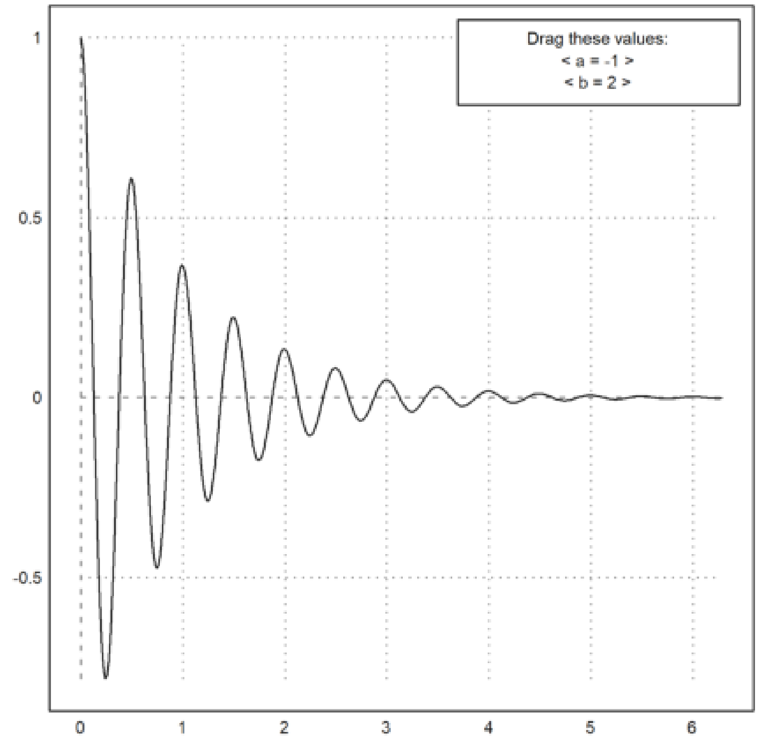
Ada juga fungsi yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.

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):
```



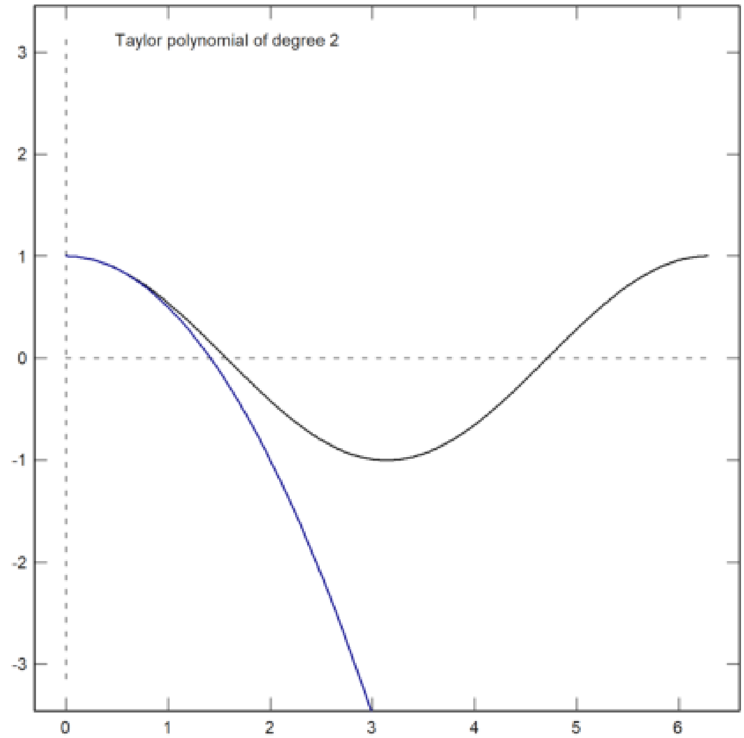
Anda dapat membatasi nilai yang diseret menjadi bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor dengan derajat n ke fungsi kosinus.

```
>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 ...
```

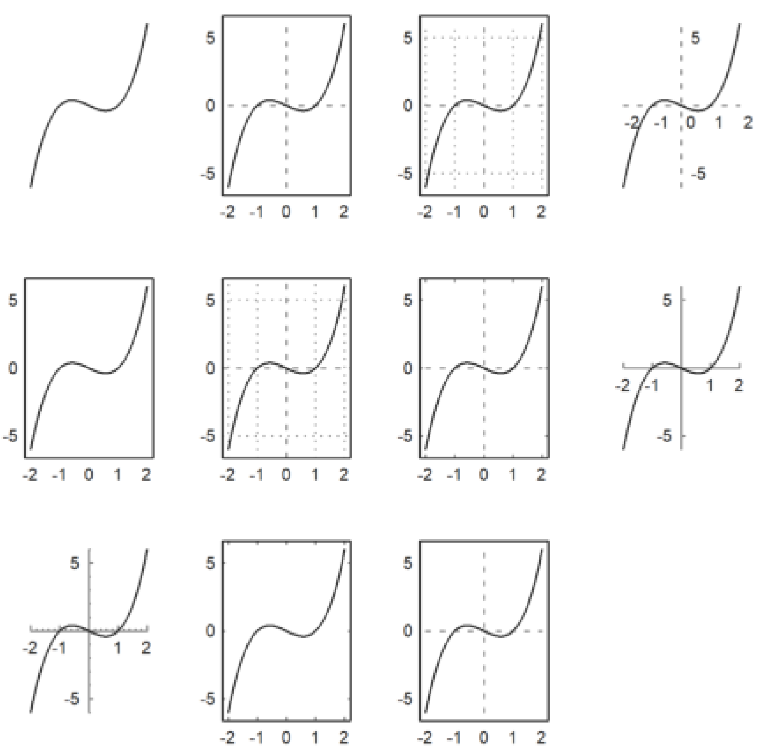
```
    plot2d(none,r=1,title="Drag with the mouse, or press any key!");
    start=0;
    repeat
      {flag,m,time}=mousedrag();
      if flag==0 then return; endif;
      if flag==2 then
        hold on; mark(m[1],m[2]); hold off;
      endif;
    end
  endfunction
```

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

Gaya Plot 2D

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

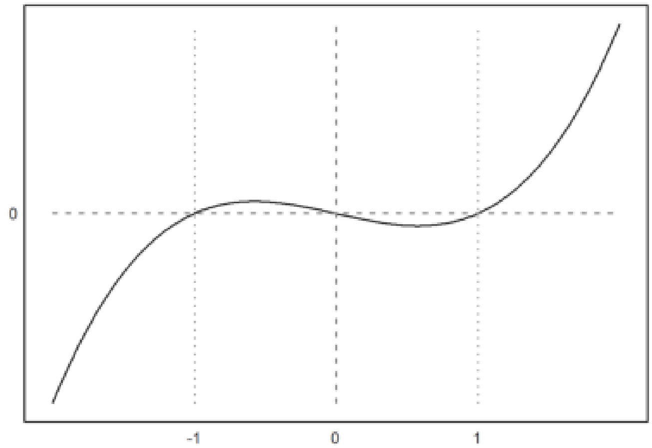
```
>aspect();
>figure(3,4); ...
> figure(1); plot2d("x^3-x",grid=0); ... // tidak ada grid, bingkai, atau s
> figure(2); plot2d("x^3-x",grid=1); ... // sumbu x-y
> figure(3); plot2d("x^3-x",grid=2); ... // kutu default
> figure(4); plot2d("x^3-x",grid=3); ... // Sumbu x-y dengan label di dalam
> figure(5); plot2d("x^3-x",grid=4); ... // tidak ada kutu, hanya label
> figure(6); plot2d("x^3-x",grid=5); ... // default, tetapi tidak ada margi
> figure(7); plot2d("x^3-x",grid=6); ... // sumbu saja
> figure(8); plot2d("x^3-x",grid=7); ... // sumbu saja, centang pada sumbu
> figure(9); plot2d("x^3-x",grid=8); ... // sumbu saja, kutu yang lebih hal
> figure(10); plot2d("x^3-x",grid=9); ... // default, kutu kecil didalamnya
> figure(11); plot2d("x^3-x",grid=10); ...// tidak ada kutu,hanya sumbu
> figure(0):
```



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

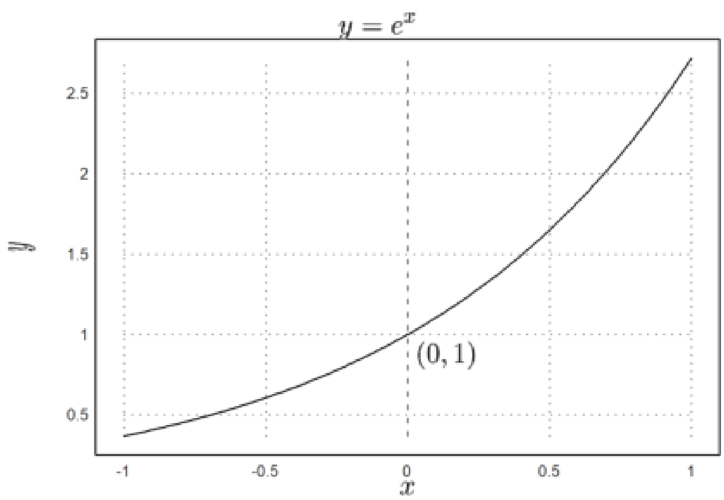
Jika Anda menginginkan tanda centang Anda sendiri, Anda dapat menggunakan style=0, dan menambahkan semuanya nanti.

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



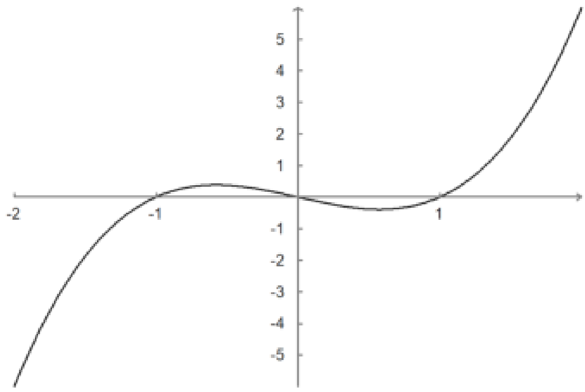
Untuk judul plot dan label sumbu, lihat contoh berikut.

```
>plot2d("exp(x)",-1,1);
>textcolor(black); // mengatur warna teks
>title(latex("y=e^x")); // judul di atas plot
>xlabel(latex("x")); // "x" for x-axis
>ylabel(latex("y"),>vertical); // vertikal "y" untuk sumbu-y
>label(latex("(0,1)"),0,1,color=blue): // memberi label sebuah titik
```



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="->"):
```

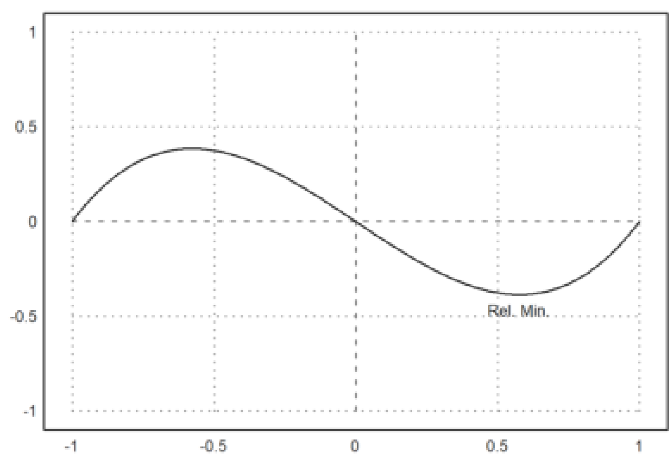


Teks pada plot dapat diatur dengan label(). Pada contoh berikut ini, "lc" berarti lower center. Ini mengatur posisi label relatif terhadap koordinat plot.

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

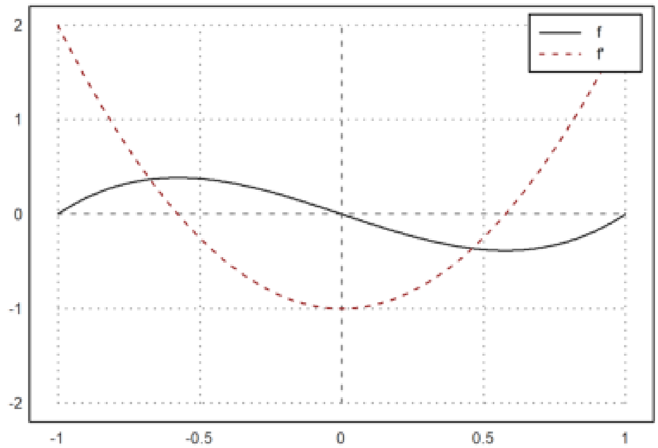
```
                                 3
                                x  - x
```

```
>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
```

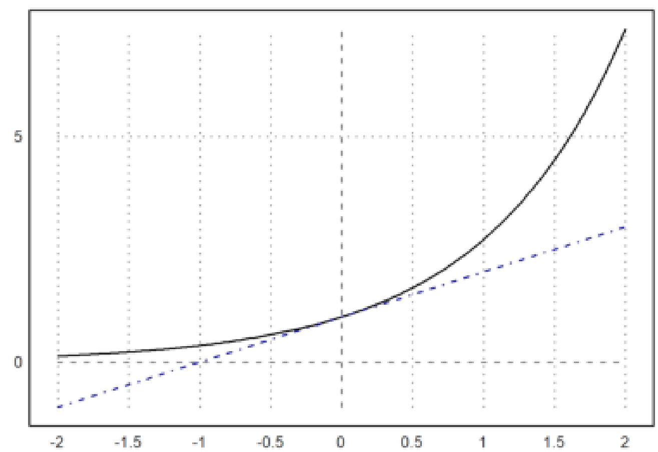


Terdapat juga kotak teks.

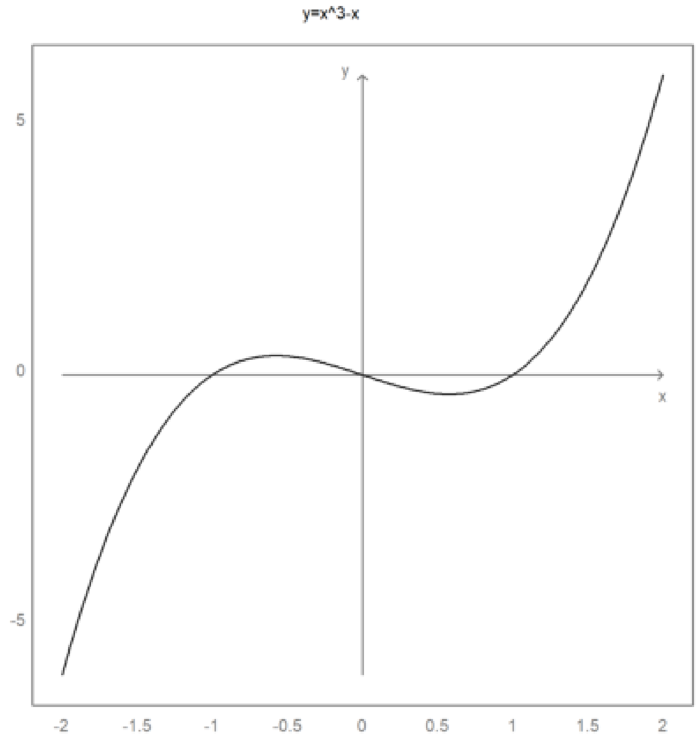
```
>plot2d(&f(x),-1,1,-2,2); // fungsi
>plot2d(&diff(f(x),x),>add,style="--",color=red); // turunan
>labelbox(["f","f'"],["-","--"],[black,red]): //kotak label
```




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



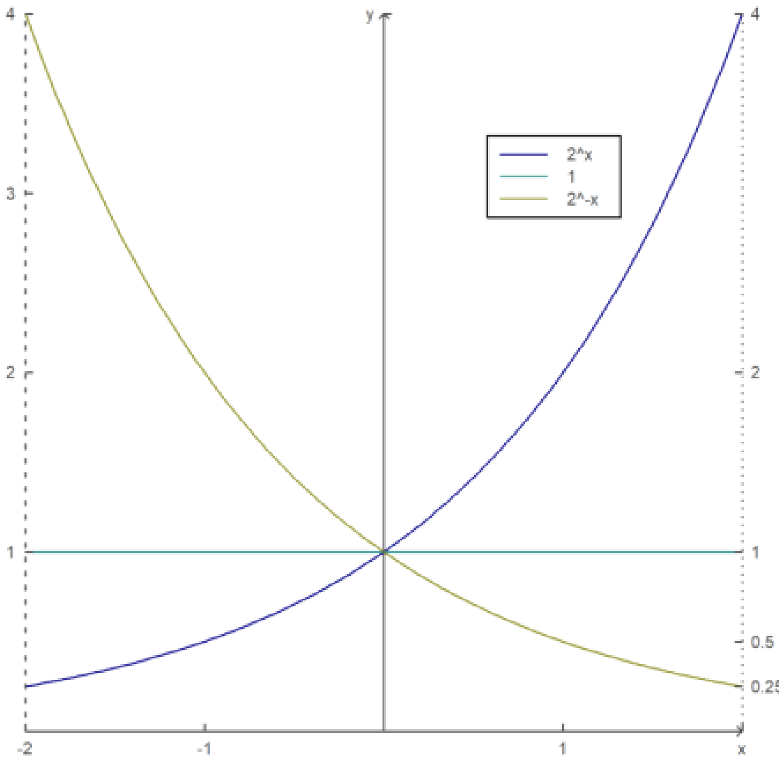

```
>gridstyle("->",color=gray,textcolor=gray,framecolor=gray);  ...
> plot2d("x^3-x",grid=1);   ...
> settitle("y=x^3-x",color=black); ...
> label("x",2,0,pos="bc",color=gray);  ...
> label("y",0,6,pos="cl",color=gray); ...
> reset():
```



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

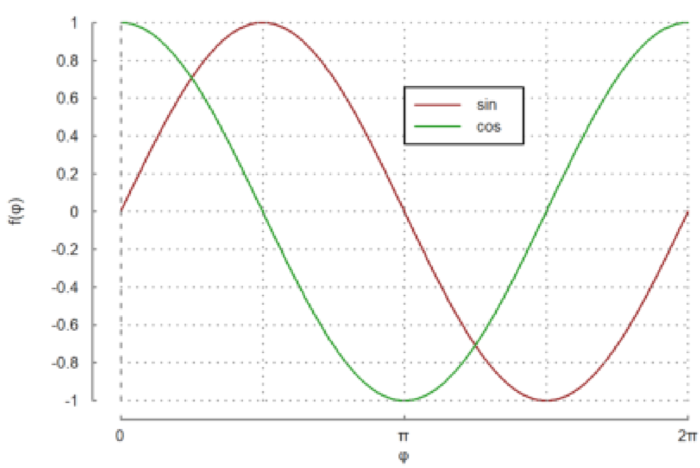
Perintah fullwindow() memperluas jendela plot karena kita tidak lagi membutuhkan tempat untuk label di luar jendela plot. Gunakan shrinkwindow() atau reset() untuk mengatur ulang ke default.

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



Berikut 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;)"):
```



Memplod 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.

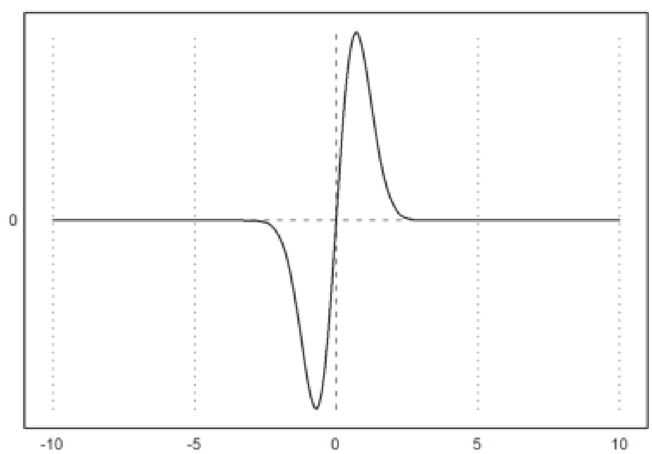
Memplot ekspresi hanyalah singkatan dari plot data. Untuk plot data, Anda memerlukan satu atau lebih baris nilai x, dan satu atau lebih baris nilai y. Dari rentang dan nilai x, fungsi plot2d akan menghitung data untuk diplot, secara default dengan evaluasi adaptif dari fungsi tersebut. Untuk plot titik,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

Tetapi 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):
```

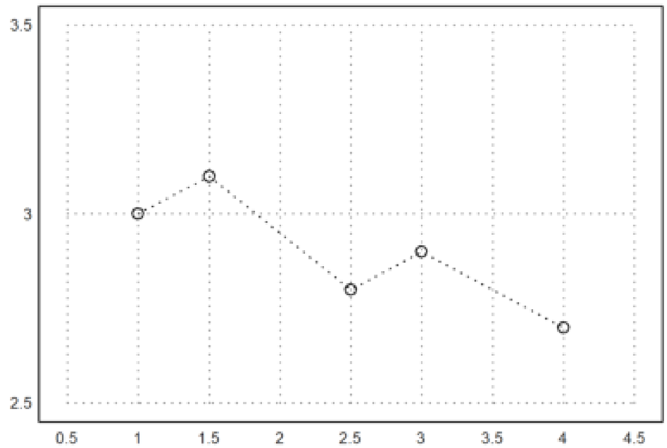


Data juga dapat diplot sebagai titik. Gunakan poin=true untuk ini. Plot ini berfungsi seperti poligon, namun hanya menggambar sudut-sudutnya saja.

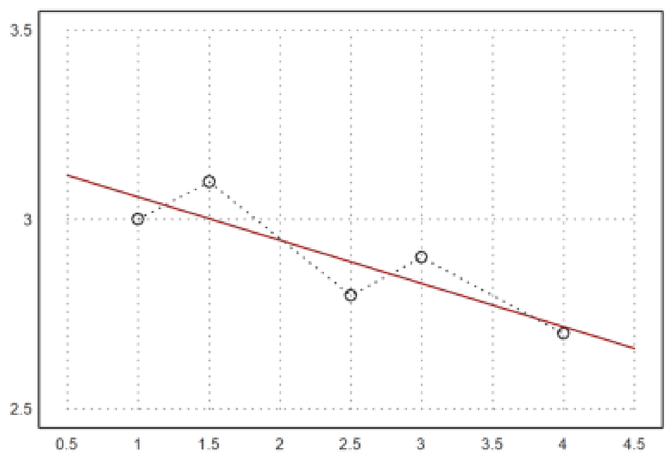
- style = "...": Pilih dari "[]", "<>", "o", ".", "..", "+", "*", "[]", "<>", "o", "..", "", "|".

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

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




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



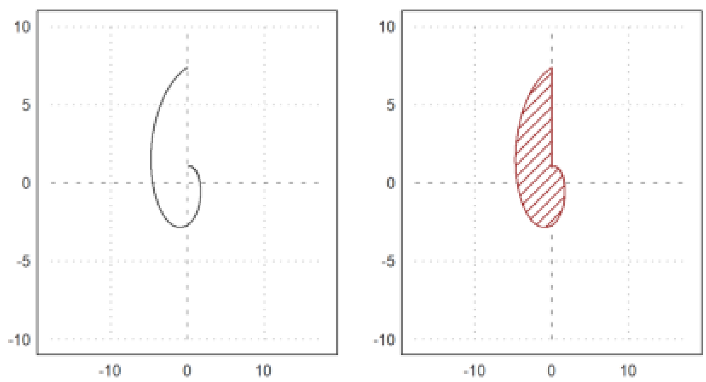
Menggambar Daerah Yang Dibatasi Kurva

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

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

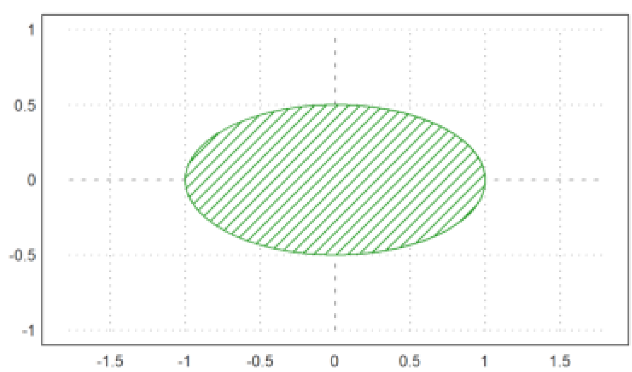
Warna isian ditentukan oleh argumen "fillcolor", dan pada opsional <outline mencegah menggambar garis batas untuk semua gaya kecuali yang default.

```
>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); // plot kurva
>figure(2); plot2d(x,y,r=10,>filled,style="/",fillcolor=red); // mengisi ku
>figure(0):
```

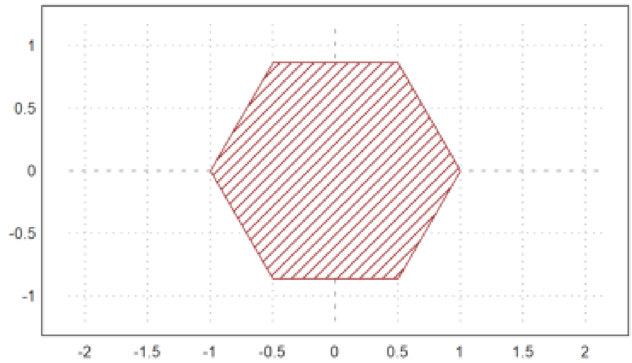


Pada contoh berikut ini, kami memplot elips yang terisi dan dua segi enam yang terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian yang berbeda.

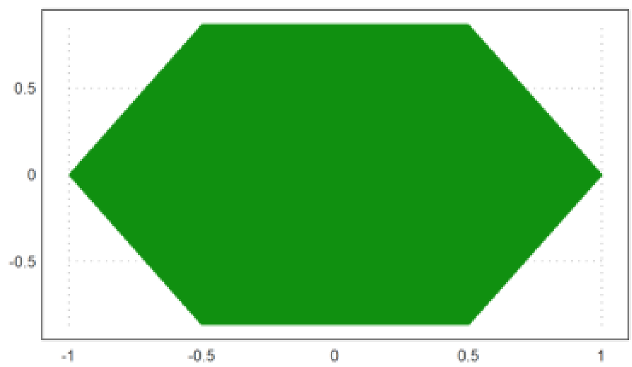
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```



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

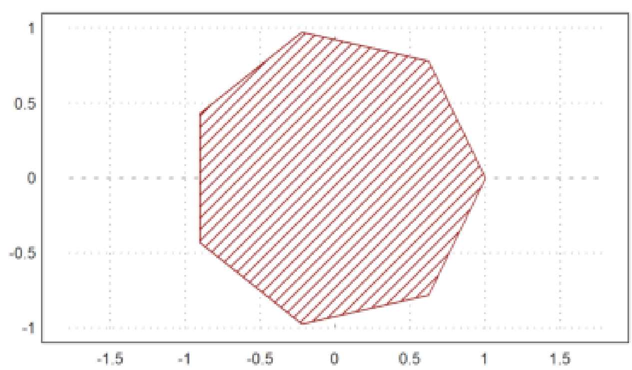



```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



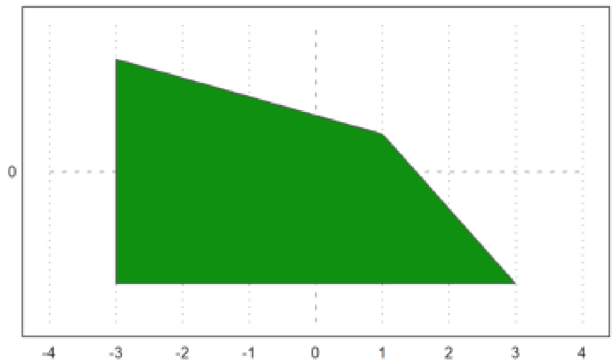
Contoh lainnya adalah septagon, yang kita buat dengan 7 titik pada lingkaran satuan.

```
>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):
```

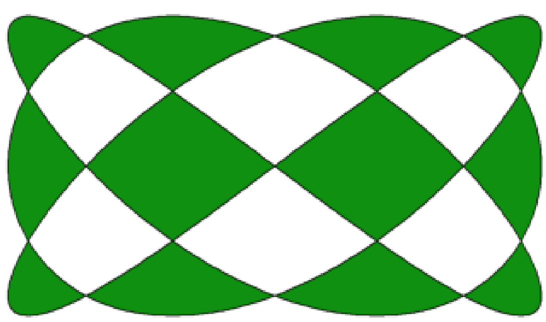


Poin utama dari bahasa matriks adalah bahwa bahasa ini memungkinkan untuk menghasilkan tabel fungsi dengan mudah.

```
>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, hal ini memberikan hasil yang bagus karena aturan penggulungan, yang digunakan untuk isi.

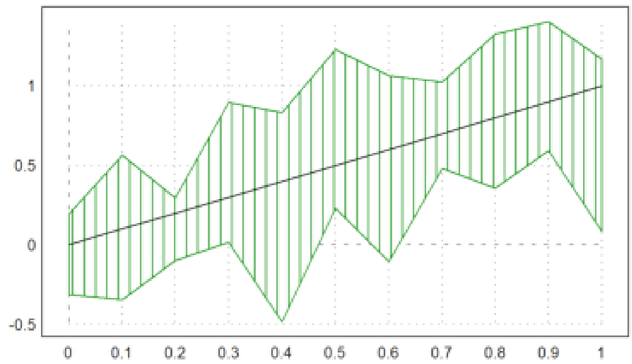
```
>plot2d(x,y,<grid,<frame,>filled):
```



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

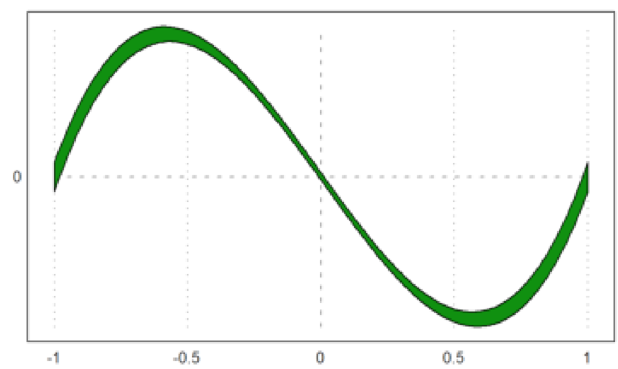
Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tetapi juga dapat digunakan untuk memplot kesalahan statistik.

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



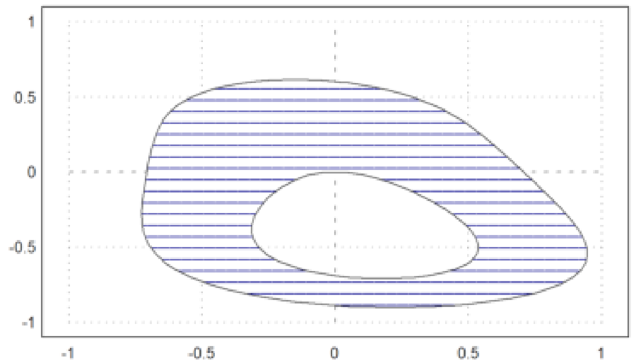
Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka plot2d akan memplot rentang interval yang terisi di bidang, gaya isian sama dengan gaya poligon.

```
>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.

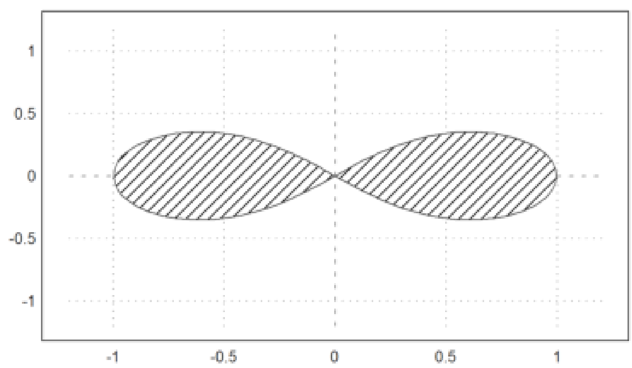
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



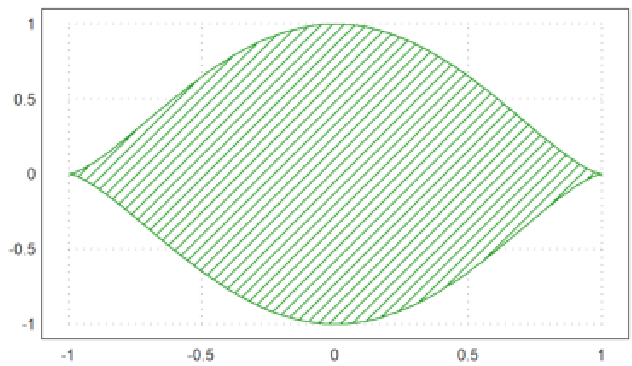
Kita juga dapat mengisi rentang nilai seperti

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

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



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



Grafik Fungsi Parametrik

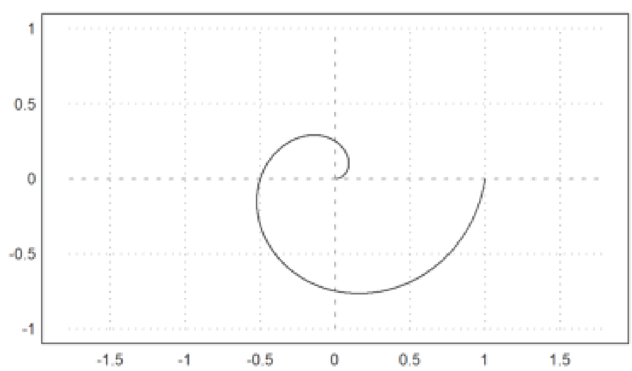
Nilai x tidak perlu diurutkan. (x,y) hanya menggambarkan sebuah kurva. Jika x diurutkan, kurva tersebut adalah grafik dari sebuah fungsi.

Dalam contoh berikut, kami memplot spiral

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

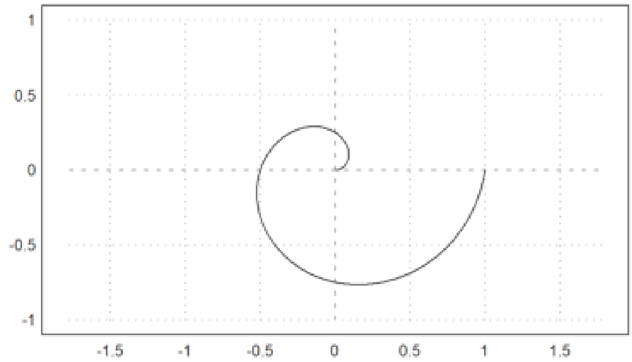
Kita perlu menggunakan sangat banyak titik untuk tampilan yang halus atau fungsi adaptive() untuk mengevaluasi ekspresi (lihat fungsi adaptive() untuk lebih jelasnya).

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

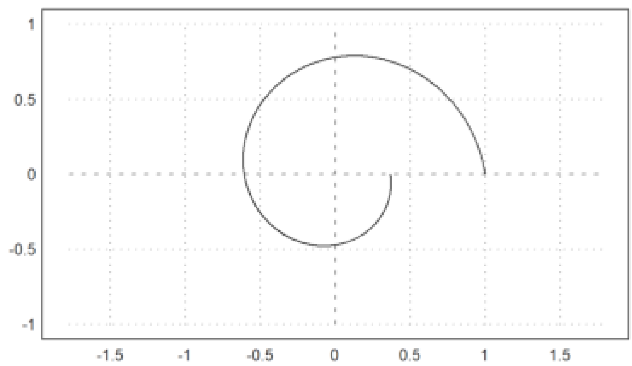


Sebagai alternatif, Anda dapat menggunakan dua ekspresi untuk kurva. Berikut ini memplot kurva yang sama seperti di atas.

```
>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):
```



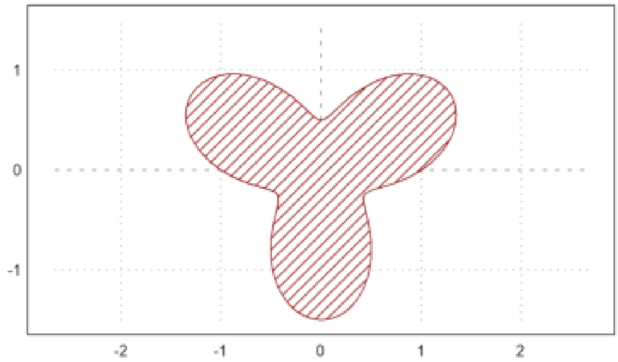
Dalam contoh berikut, kami memplot kurva

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

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):
```

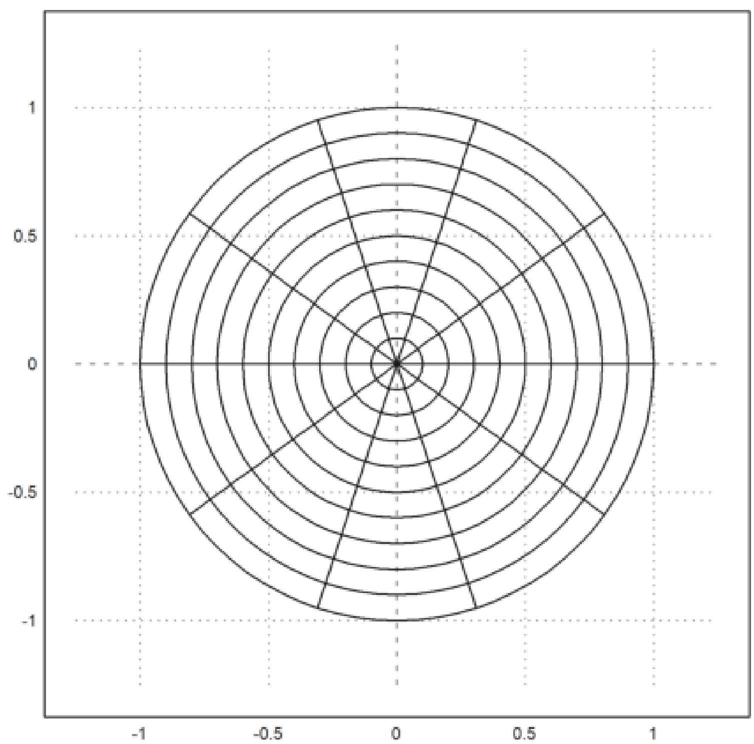


Larik bilangan kompleks juga dapat diplot. Kemudian titik-titik kisi akan dihubungkan. Jika sejumlah garis kisi ditentukan (atau vektor 1x2 garis kisi) pada argumen cgrid, hanya garis-garis kisi tersebut yang akan terlihat.

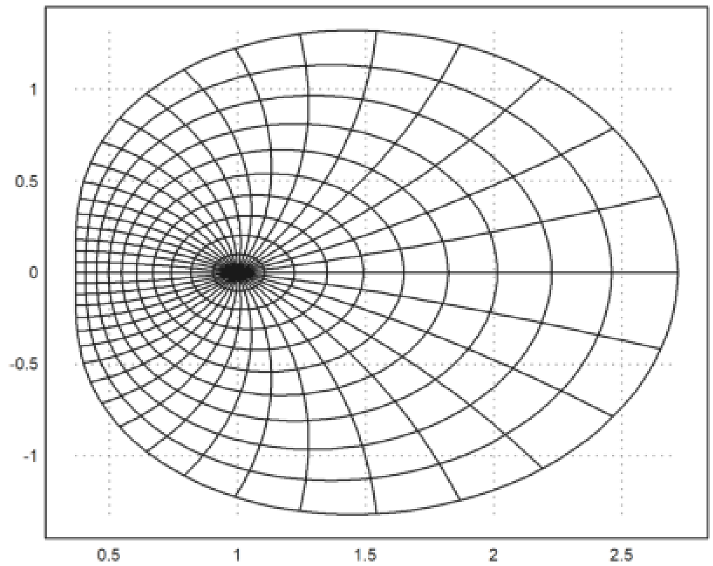
Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi-kisi pada bidang kompleks.

Pada contoh berikut, kami memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva kisi-kisi.

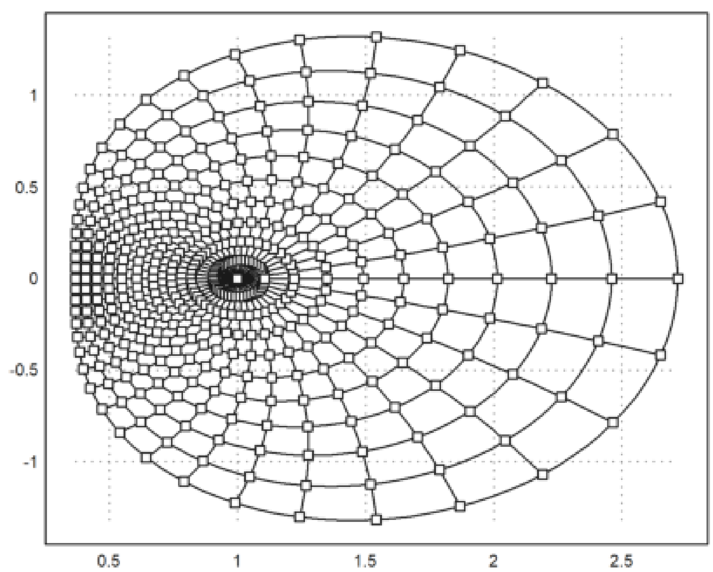
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```



```
>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):
```

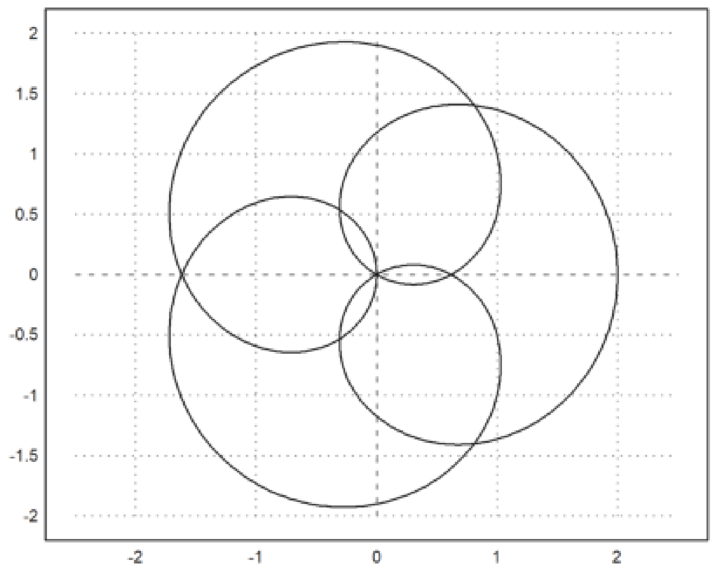


Vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian riil dan bagian imajiner.

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):
```



Plot Statistika

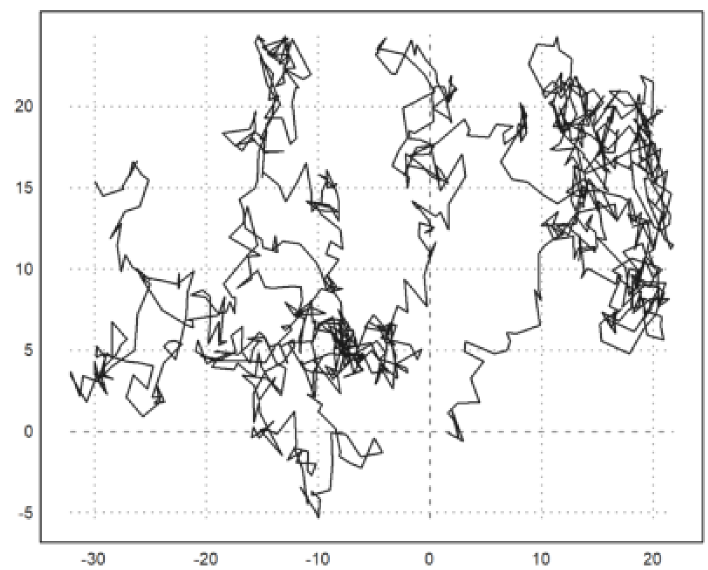
Terdapat banyak fungsi yang dikhususkan pada plot statistik. Salah satu plot yang sering digunakan adalah plot kolom.

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

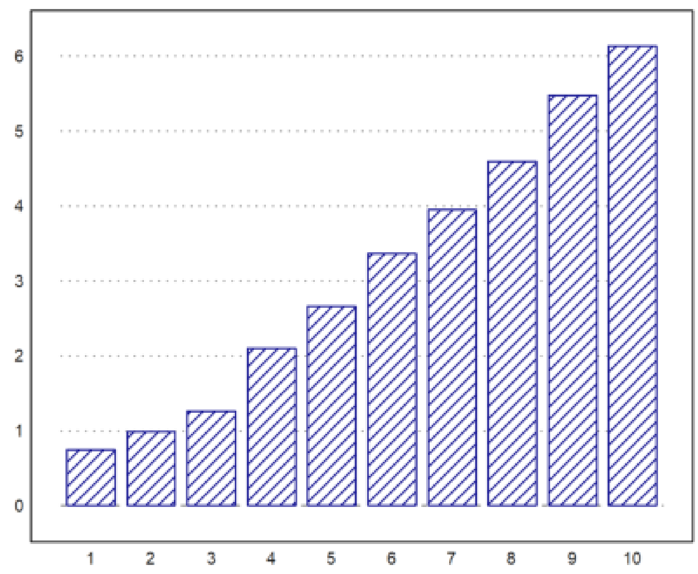
```
>plot2d(cumsum(randnormal(1,1000))):
```

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

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

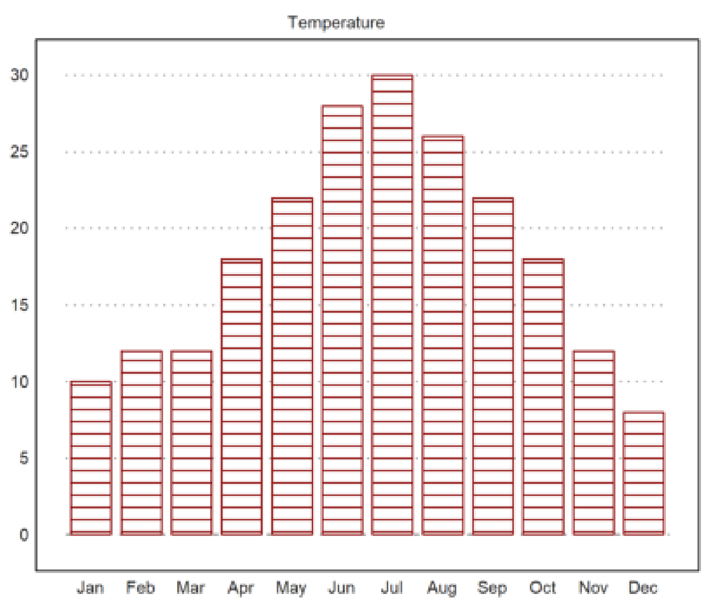



```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

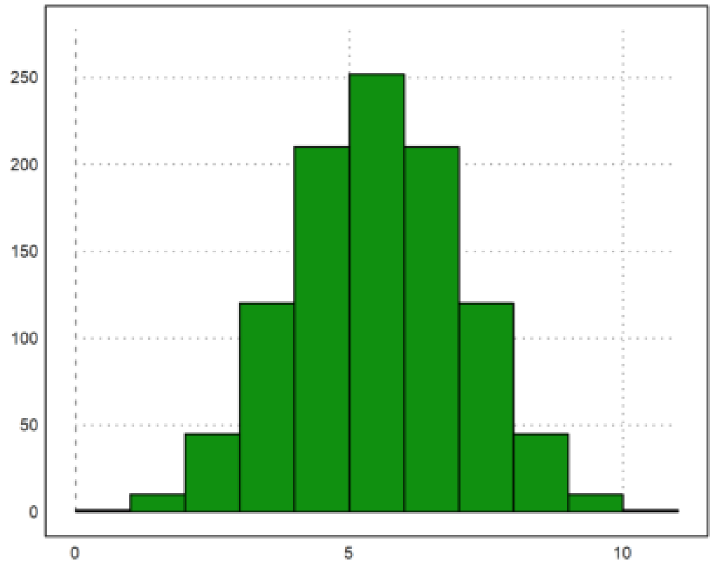


Ini juga dapat menampilkan string sebagai label.

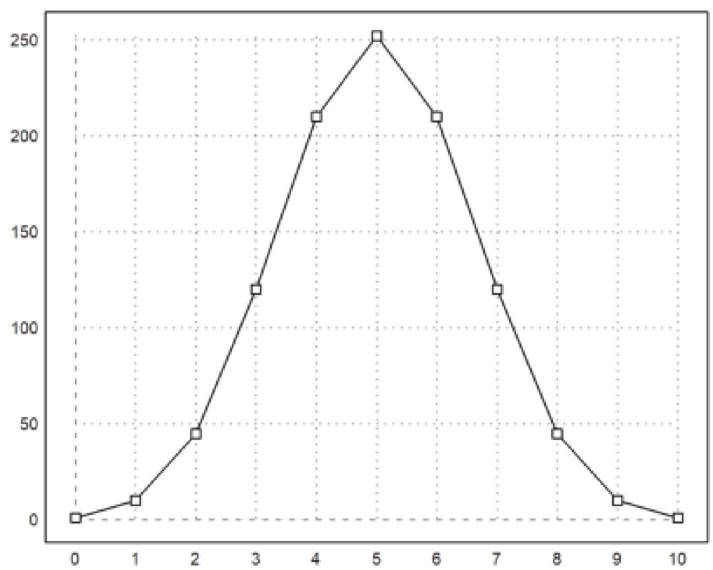
```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
>  "Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnsplot(values,lab=months,color=red,style="-");
>title("Temperature"):
```



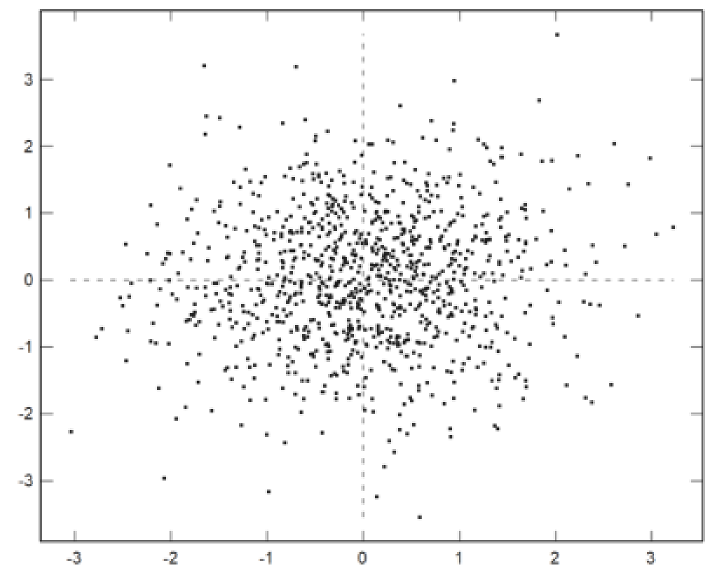

```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```



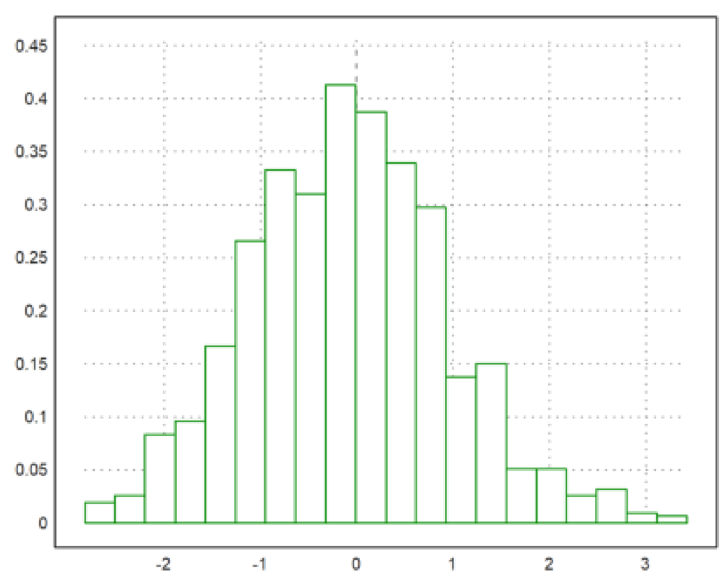

```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



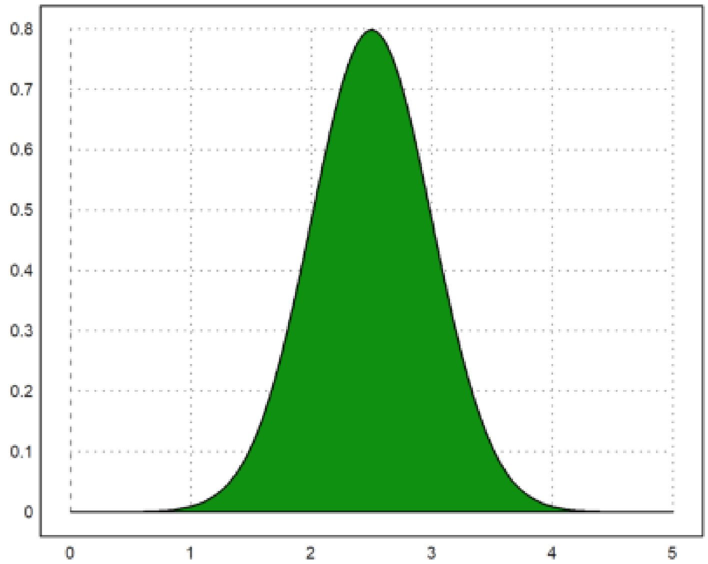

```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```




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

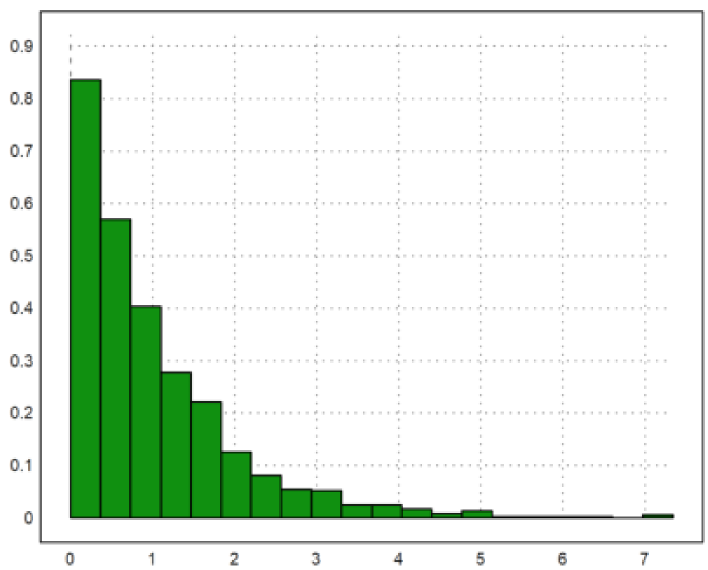



```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



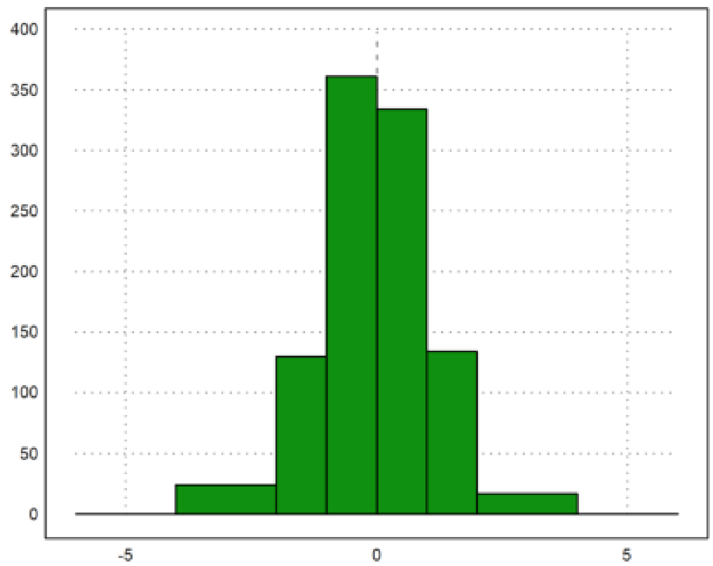
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan distribution=n dengan plot2d.

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



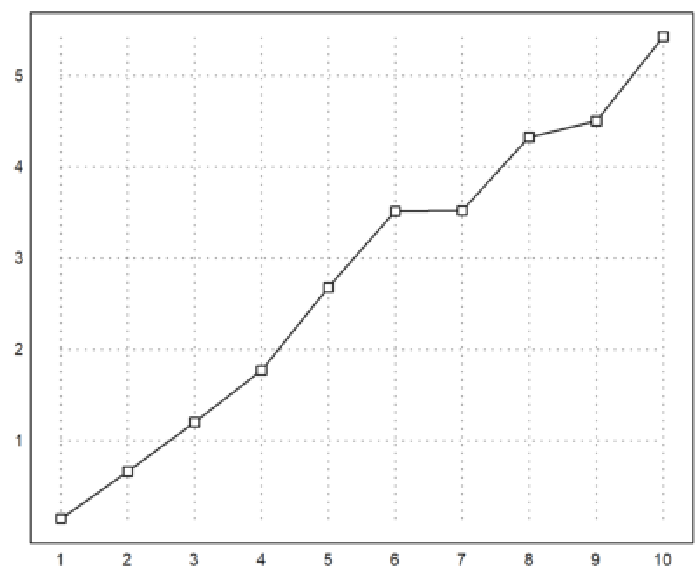
Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan >bar di plot3d, atau dengan plot kolom.

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

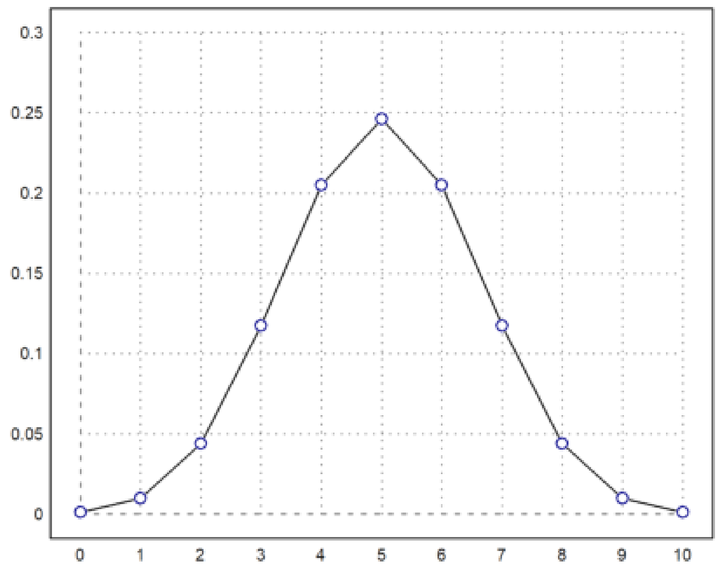


Fungsi statplot() menetapkan gaya dengan string sederhana.

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



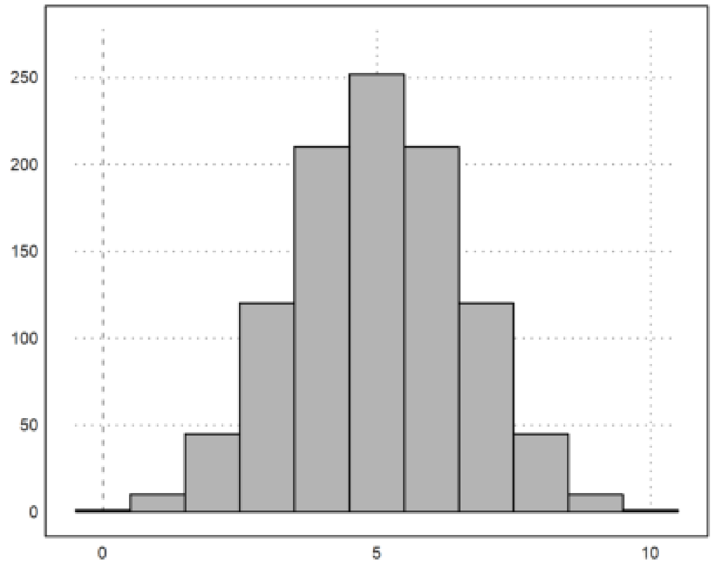
```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



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.

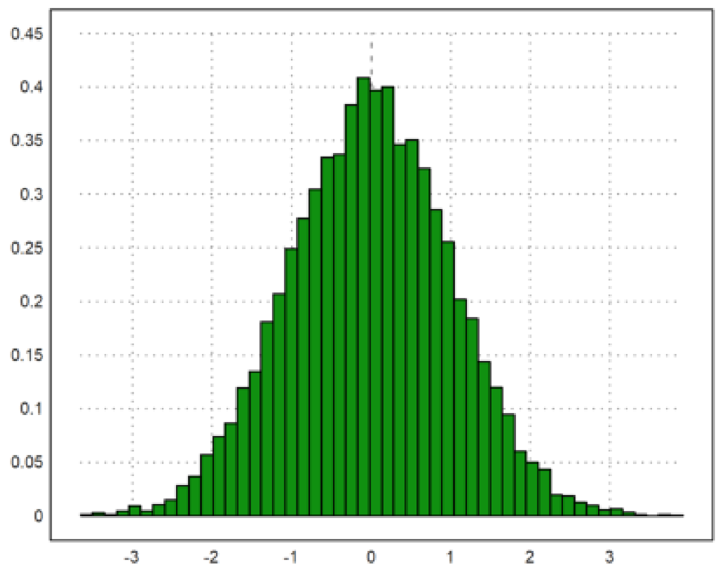
Gaya isian dapat digunakan seperti di atas.

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

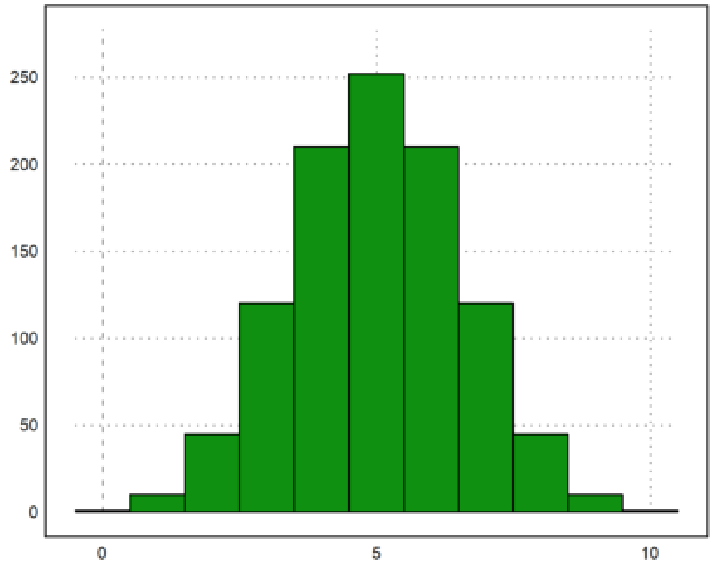


Data untuk plot batang (batang = 1) dan histogram (histogram = 1) dapat diberikan secara eksplisit dalam xv dan yv, atau dapat dihitung dari distribusi empiris dalam xv dengan >distribusi (atau distribusi = n). Histogram dari nilai xv akan dihitung secara otomatis dengan >histogram. Jika >even ditentukan,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

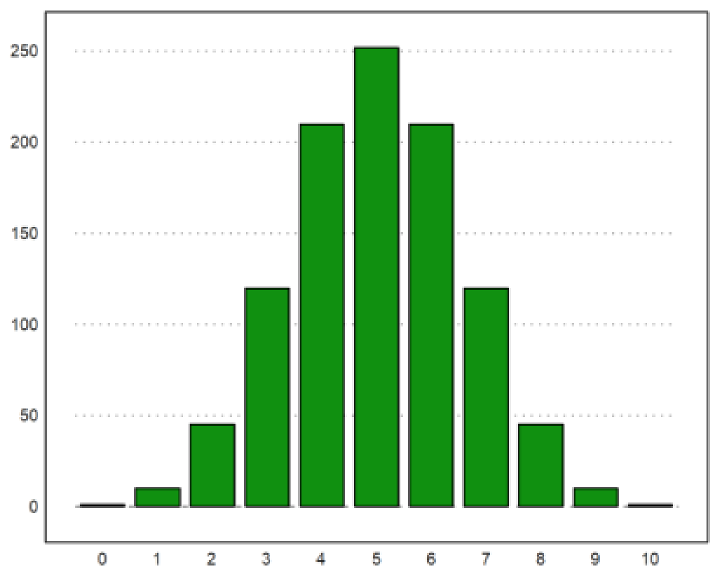
```
>plot2d(normal(10000),distribution=50):
```



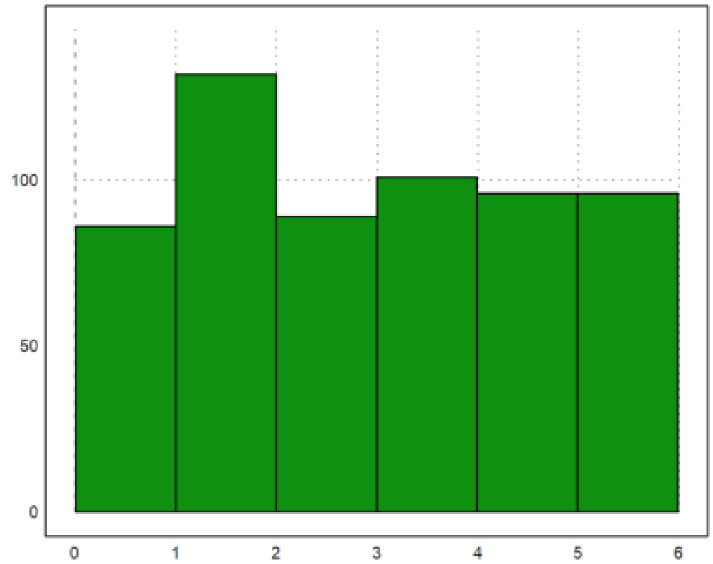

```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```




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

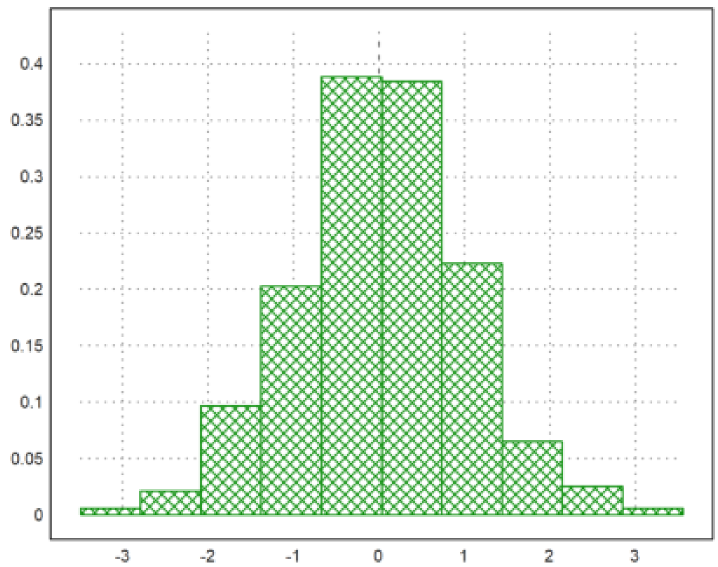



```
>plot2d(random(600)*6,histogram=6):
```



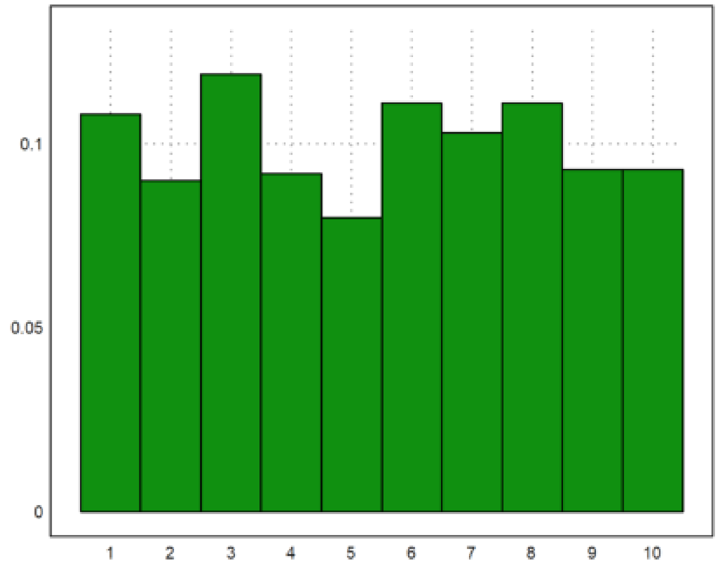
Untuk distribusi, ada parameter distribution=n, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.

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



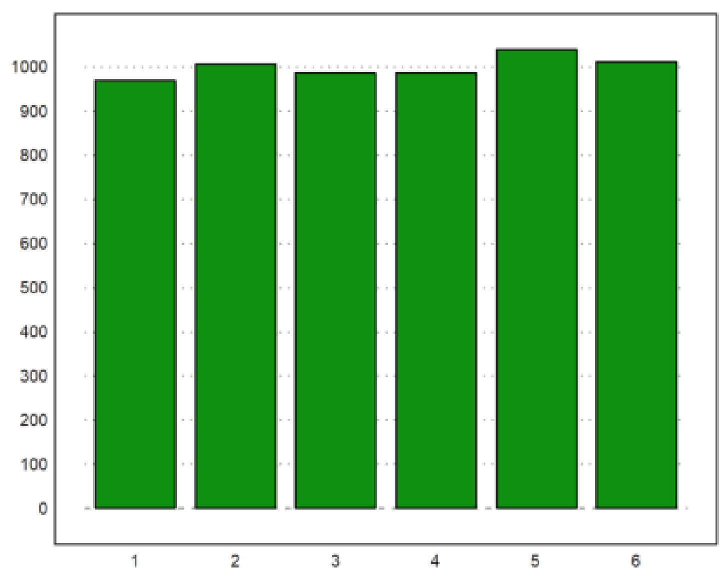
Dengan parameter even=true, ini akan menggunakan interval bilangan bulat.

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

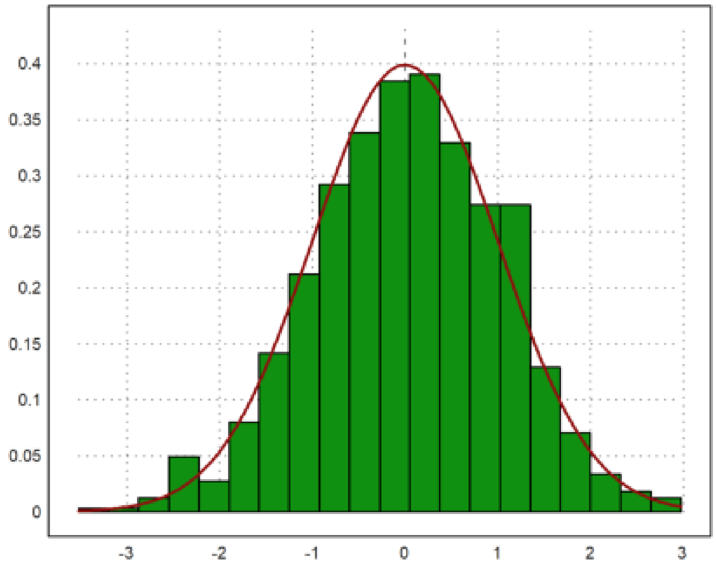


Perhatikan bahwa ada banyak plot statistik yang mungkin berguna. Lihatlah tutorial tentang statistik.

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

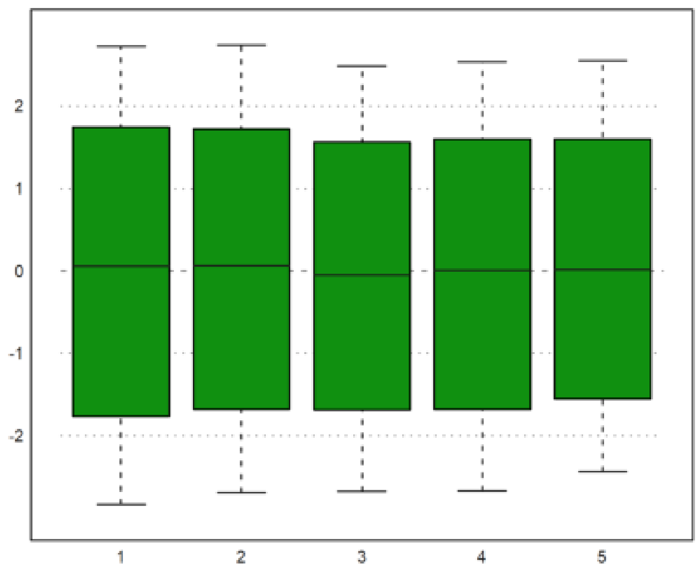



```
>plot2d(normal(1,1000),>distribution); ...
>  plot2d("qnormal(x)",color=red,thickness=2,>add):
```



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)):
```



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 nilai 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 sebuah matriks nilai.

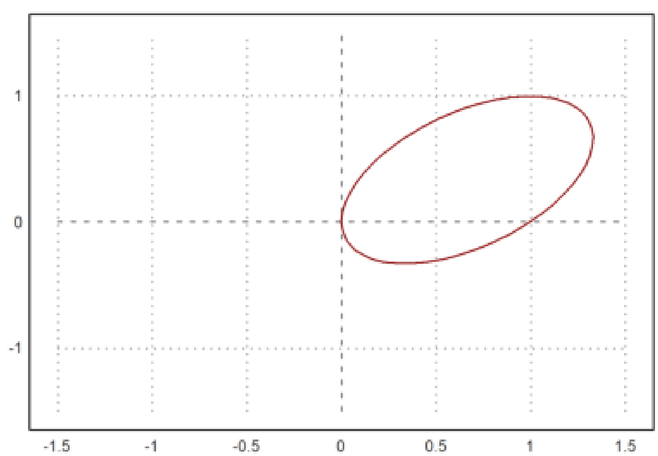
Euler dapat menandai garis level

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

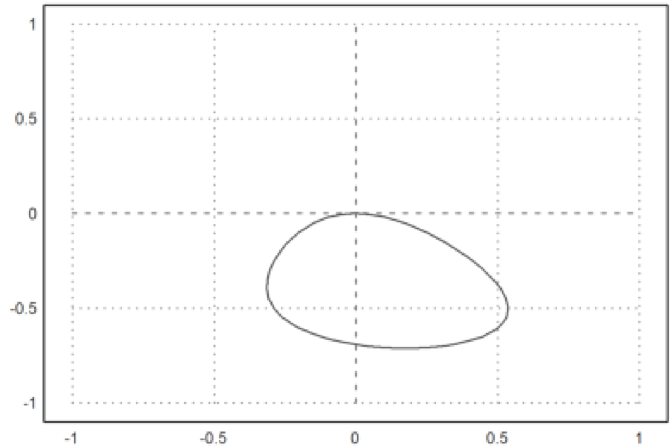
dari fungsi apa pun.

Untuk menggambar himpunan f(x,y)=c untuk satu atau lebih konstanta c, Anda dapat 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.

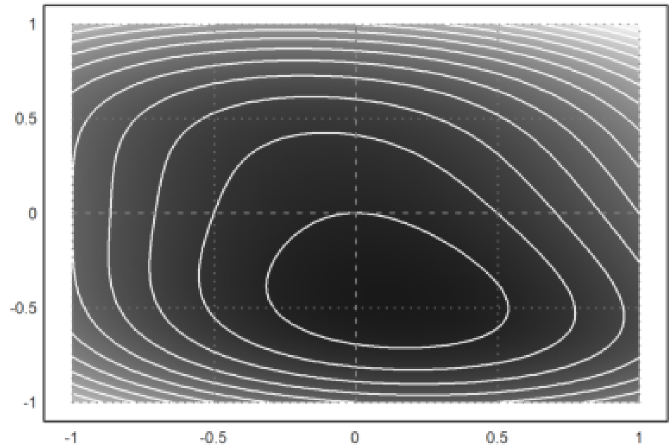
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```



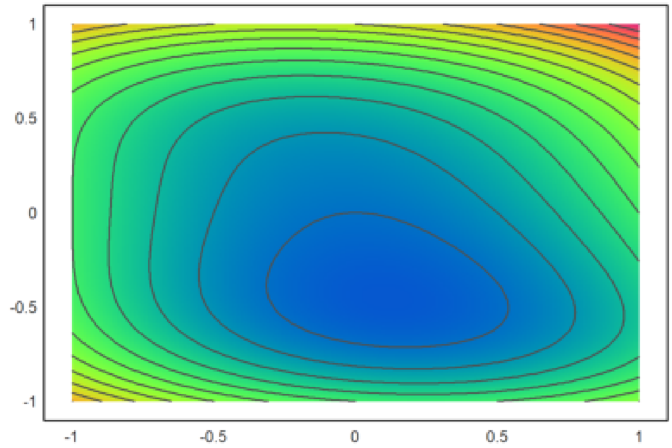

```
>expr := "2*x^2+x*y+3*y^4+y"; // mendefinisikan ekspresi f(x,y)
>plot2d(expr,level=0): //solusi dari f(x,y)=0
```




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

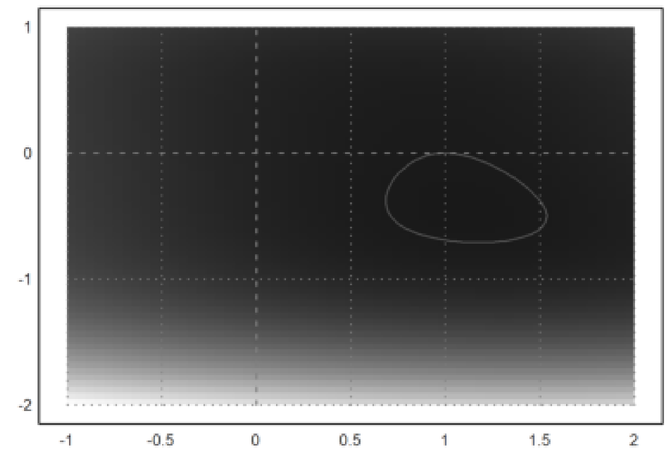



```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // lebih bagus
```

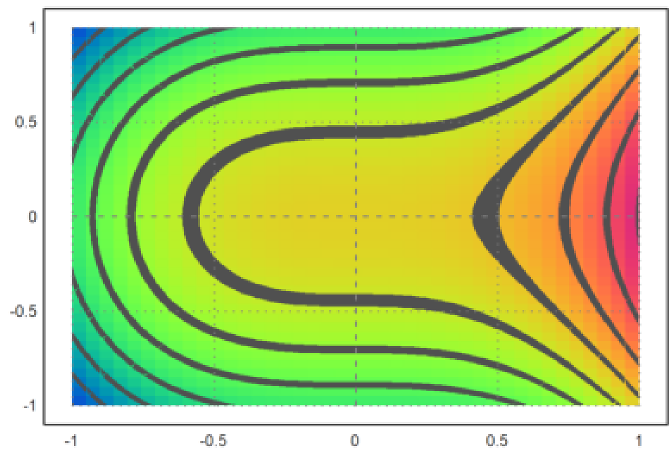


Hal ini juga berlaku untuk plot data. Tetapi Anda harus menentukan rentang untuk label sumbu.

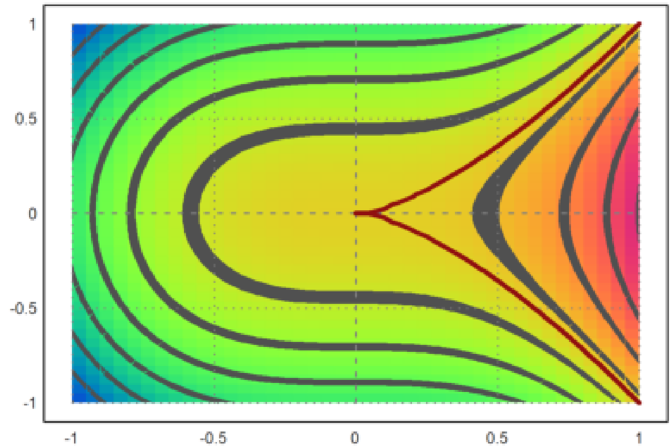
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



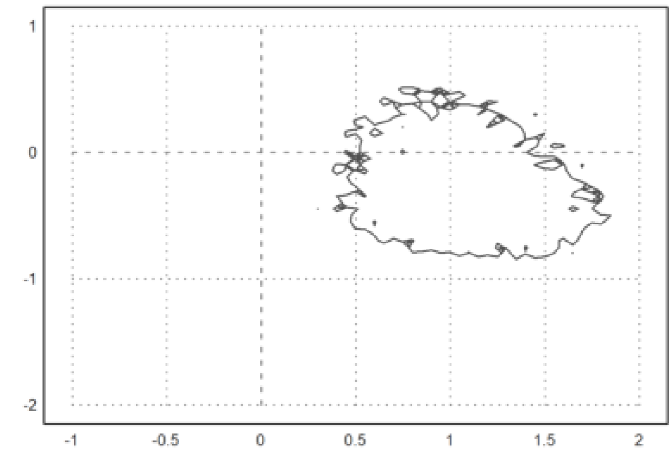

```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



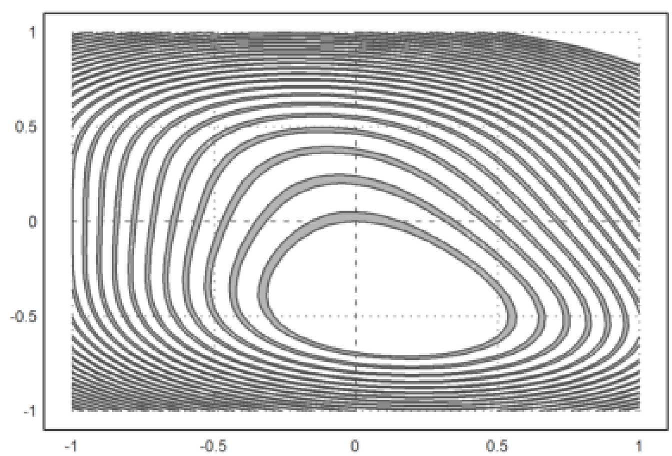

```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



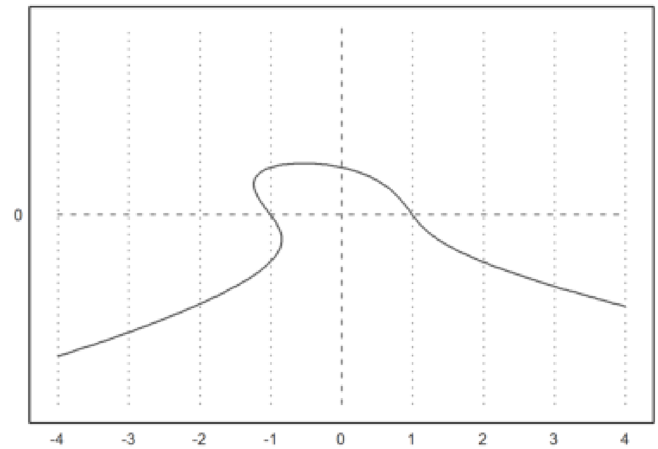

```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



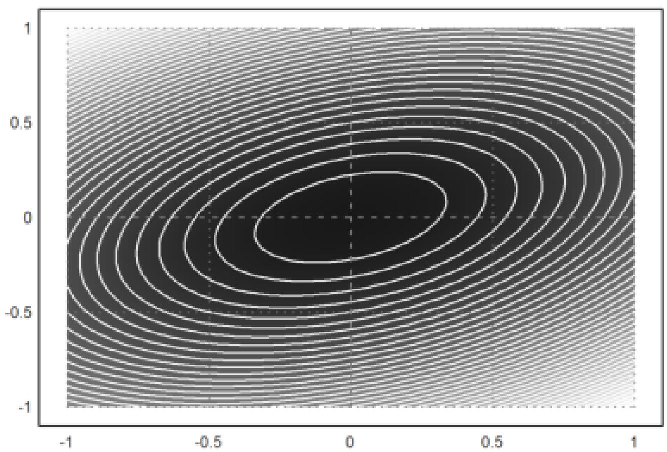

```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```



```
>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):
```



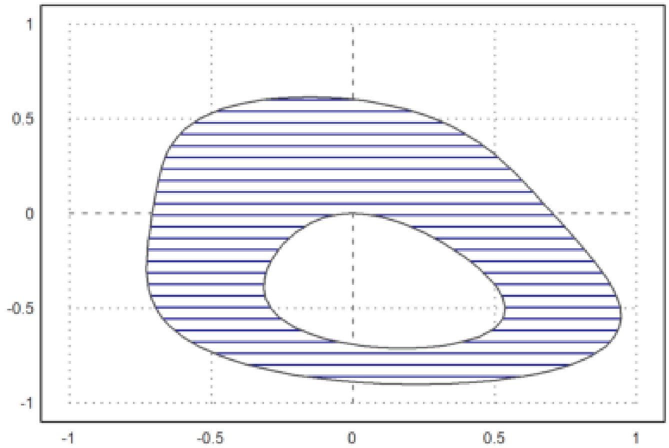
Dimungkinkan juga untuk mengisi set

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

dengan rentang level.

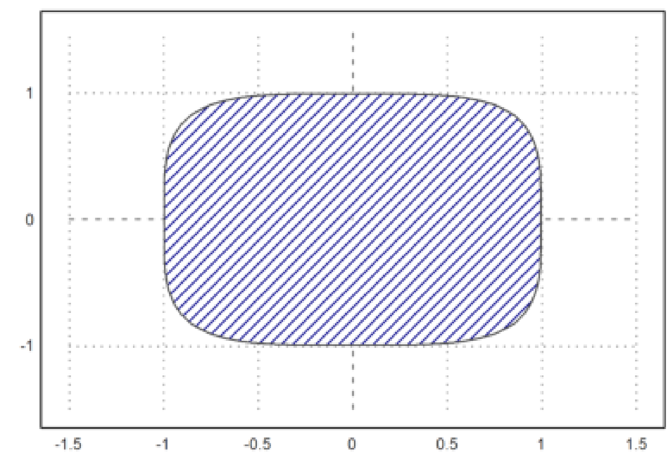
Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

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

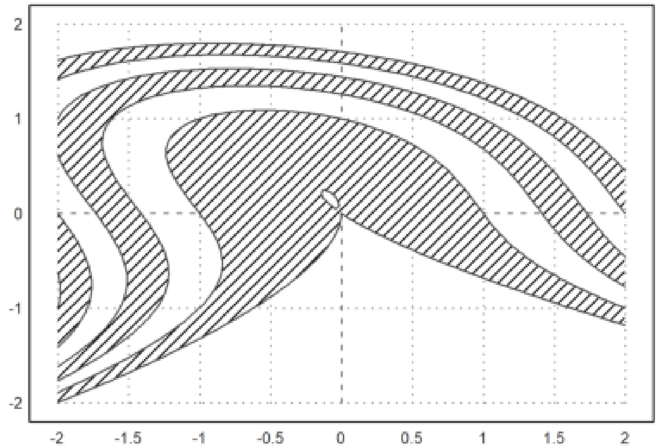


Plot implisit juga dapat menunjukkan rentang level. Maka level harus berupa matriks 2xn interval level, di mana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval. Sebagai alternatif, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level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

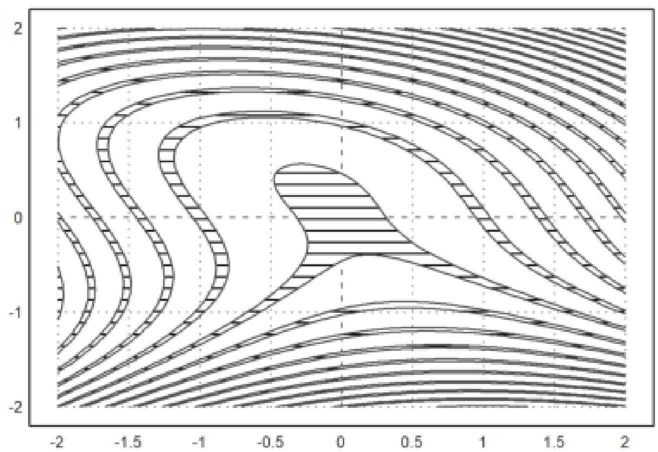
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



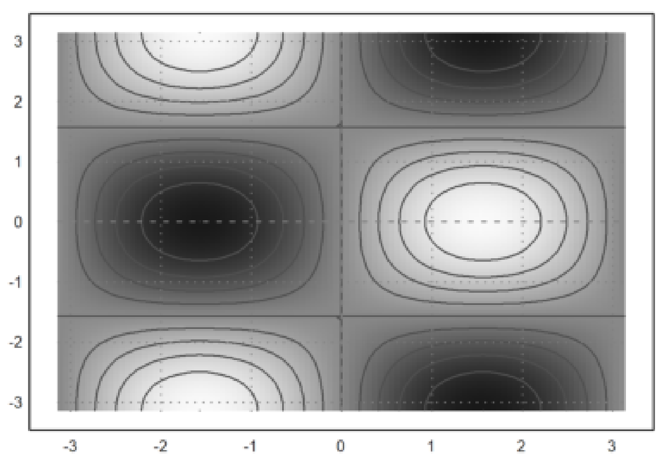

```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```




```
>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):
```

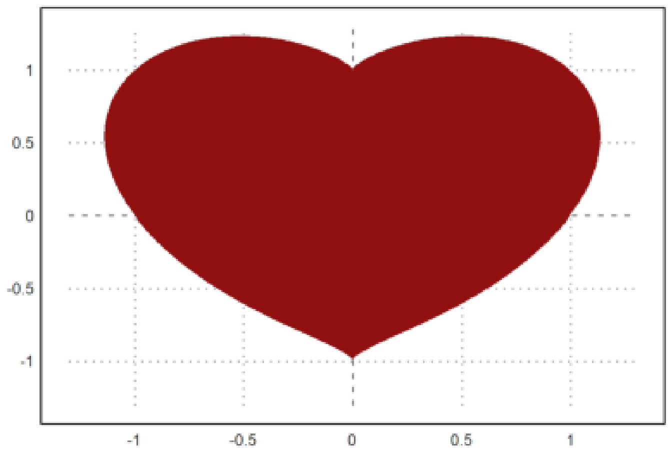


Anda juga dapat menandai suatu wilayah

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

Hal ini dilakukan dengan menambahkan level dengan dua baris.

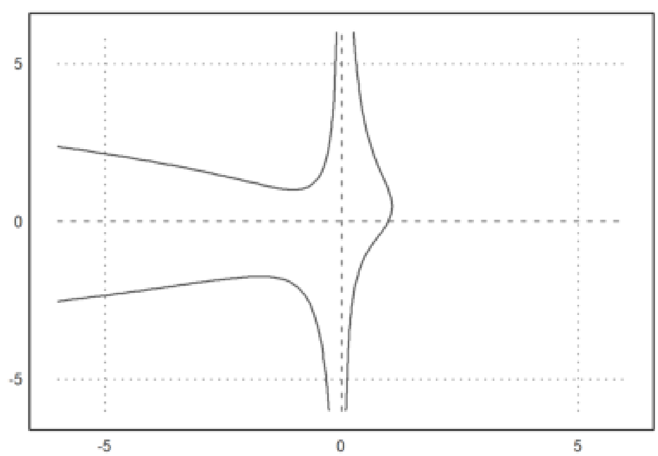
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
>  style="#",color=red,<outline, ...
>  level=[-2;0],n=100):
```



Dimungkinkan untuk menentukan level tertentu. Misalnya, 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):
```



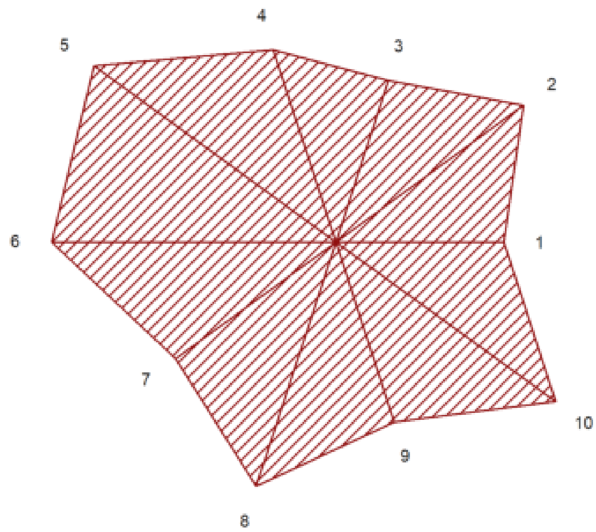
```
>function starplot1 (v, style="/", color=green, lab=none) ...
```

```
  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 merencanakan sesuatu yang tidak dapat dilakukan oleh plot2d, tetapi hampir.

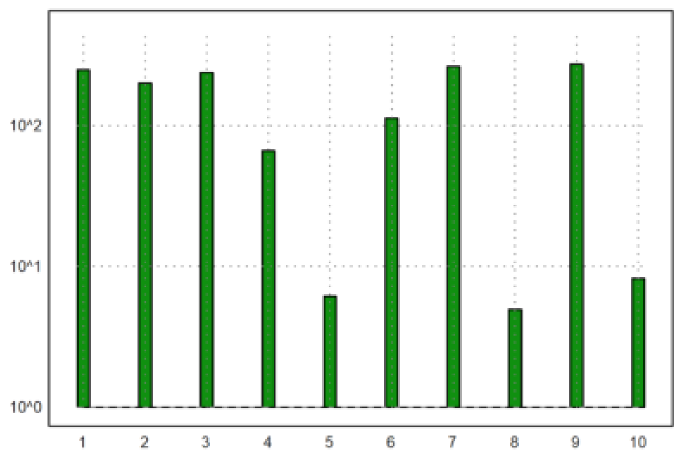
Pada fungsi berikut, kita melakukan plot impuls logaritmik. plot2d dapat melakukan plot logaritmik, tetapi tidak untuk batang impuls.

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

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

Mari kita uji dengan nilai yang terdistribusi secara eksponensial.

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



Mari kita 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 grafis. Kalau tidak, penggambaran ulang 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
```

Tekan sembarang tombol untuk menghentikan animasi ini.

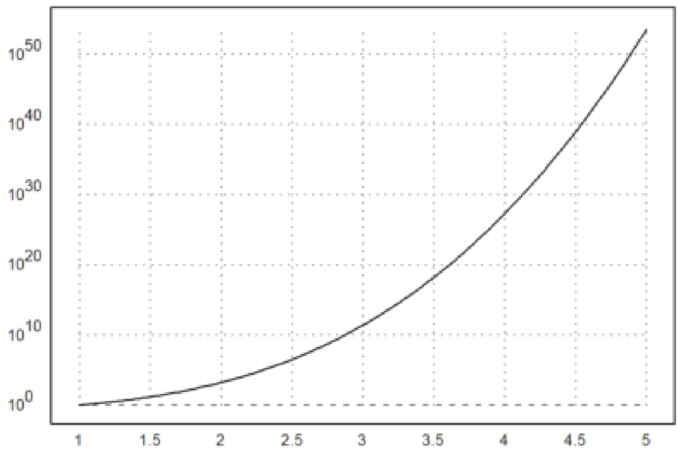
```
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

Plot Logaritmik

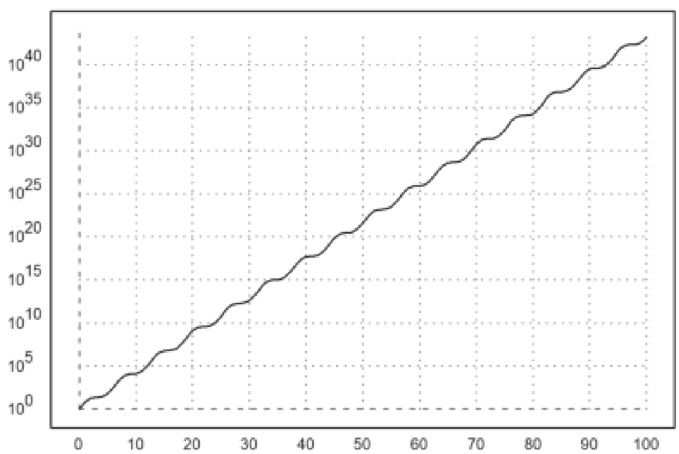
EMT menggunakan parameter "logplot" untuk skala logaritmik.
Plot logaritmik dapat diplot menggunakan skala logaritmik dalam y dengan logplot = 1, atau menggunakan skala logaritmik dalam x dan y dengan logplot = 2, atau dalam x dengan logplot = 3.

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

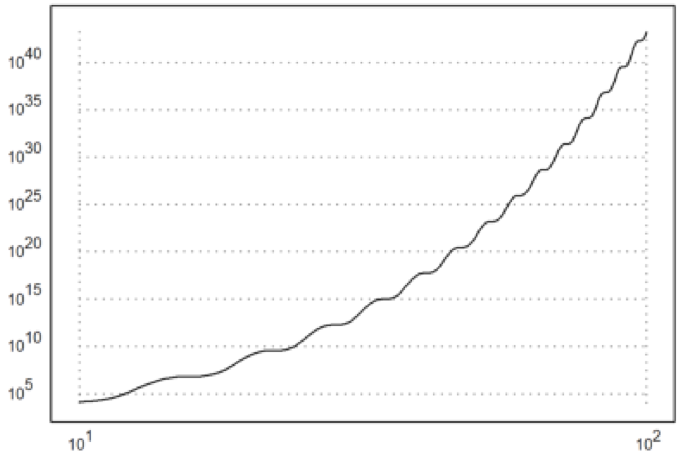
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



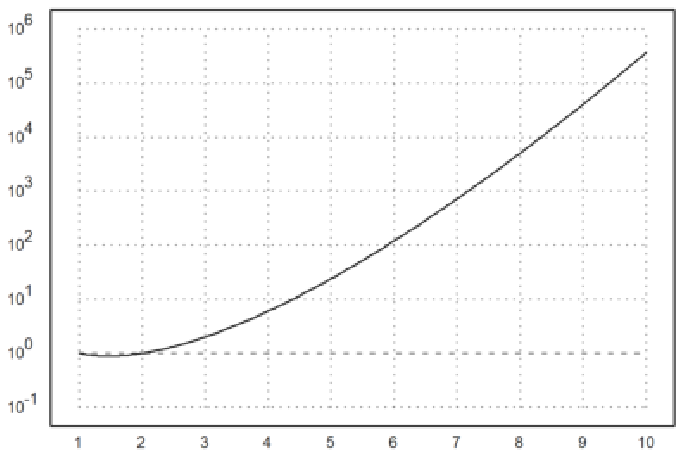

```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



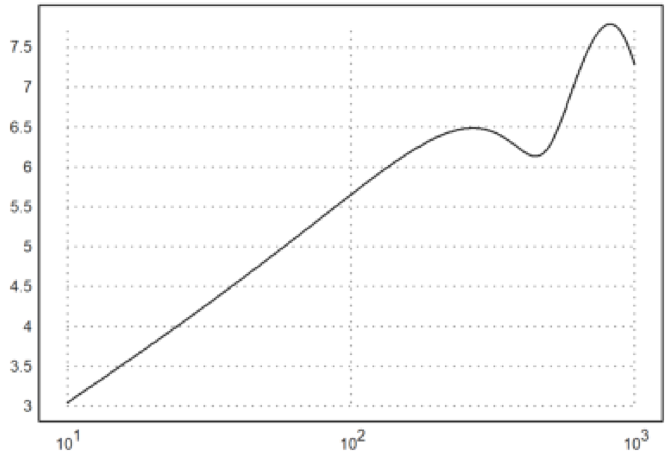
```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```




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

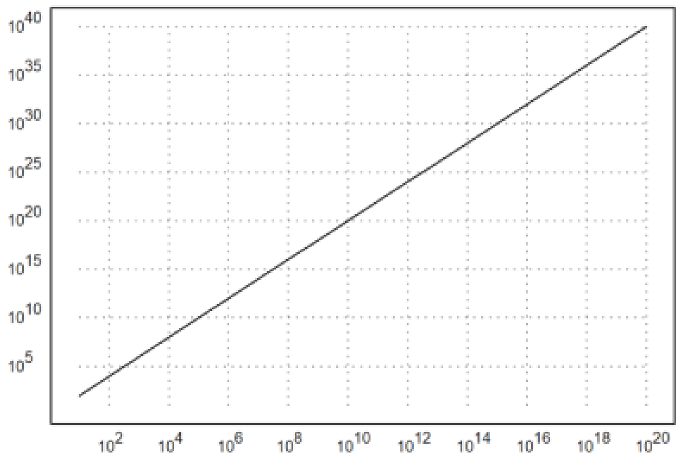



```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Hal ini juga berlaku pada plot data.

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



Rujukan Lengkap Fungsi plot2d()

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

Multipurpose plot function for plots in the plane (2D plots). This function can do plots of functions of one variables, data plots, curves in the plane, bar plots, grids of complex numbers, and implicit plots of functions of two variables.

Parameters
x,y : equations, functions or data vectors
a,b,c,d : Plot area (default a=-2,b=2)
r : if r is set, then a=cx-r, b=cx+r, c=cy-r, d=cy+r

```
r can be a vector [rx,ry] or a vector [rx1,rx2,ry1,ry2].
```

xmin,xmax : range of the parameter for curves
auto : Determine y-range automatically (default)
square : if true, try to keep square x-y-ranges
n : number of intervals (default is adaptive)
grid : 0 = no grid and labels,

```
1 = axis only,
2 = normal grid (see below for the number of grid lines)
3 = inside axis
4 = no grid
5 = full grid including margin
6 = ticks at the frame
7 = axis only
8 = axis only, sub-ticks
```

frame : 0 = no frame
framecolor: color of the frame and the grid
margin : number between 0 and 0.4 for the margin around the plot
color : Color of curves. If this is a vector of colors,

```
it will be used for each row of a matrix of plots. In the case
point plots, it should be a column vector. If a row vector or a
full matrix of colors is used for point plots, it will be used
each data point.
```

thickness : line thickness for curves

```
This value can be smaller than 1 for very thin lines.
```

style : Plot style for lines, markers, and fills.

```
For points use
"[]", "<>", ".", "..", "...",
"*", "+", "|", "-", "o"
"[]#", "<>#", "o#" (filled shapes)
"[]w", "<>w", "ow" (non-transparent)
For lines use
"-", "--", "-.", ".", ".-.", "-.-", "->"
For filled polygons or bar plots use
"#", "#O", "O", "/", "\", "\/",
"+", "|", "-", "t"
```

points : plot single points instead of line segments
addpoints : if true, plots line segments and points
add : add the plot to the existing plot
user : enable user interaction for functions
delta : step size for user interaction
bar : bar plot (x are the interval bounds, y the interval values)
histogram : plots the frequencies of x in n subintervals
distribution=n : plots the distribution of x with n subintervals
even : use inter values for automatic histograms.
steps : plots the function as a step function (steps=1,2)
adaptive : use adaptive plots (n is the minimal number of steps)
level : plot level lines of an implicit function of two variables
outline : draws boundary of level ranges.
If the level value is a 2xn matrix, ranges of levels will be drawn
in the color using the given fill style. If outline is true, it
will be drawn in the contour color. Using this feature, regions of
f(x,y) between limits can be marked.
hue : add hue color to the level plot to indicate the function

```
value
```

contour : Use level plot with automatic levels
nc : number of automatic level lines
title : plot title (default "")
xl, yl : labels for the x- and y-axis
smaller : if >0, there will be more space to the left for labels.
vertical :

```
Turns vertical labels on or off. This changes the global variable
verticallabels locally for one plot. The value 1 sets only vertical
text, the value 2 uses vertical numerical labels on the y axis.
```

filled : fill the plot of a curve
fillcolor : fill color for bar and filled curves
outline : boundary for filled polygons
logplot : set logarithmic plots

```
1 = logplot in y,
2 = logplot in xy,
3 = logplot in x
```

own :

```
A string, which points to an own plot routine. With >user, you get
the same user interaction as in plot2d. The range will be set
before each call to your function.
```

maps : map expressions (0 is faster), functions are always mapped.
contourcolor : color of contour lines
contourwidth : width of contour lines
clipping : toggles the clipping (default is true)
title :

```
This can be used to describe the plot. The title will appear above
the plot. Moreover, a label for the x and y axis can be added with
xl="string" or yl="string". Other labels can be added with the
functions label() or labelbox(). The title can be a unicode
string or an image of a Latex formula.
```

cgrid :

```
Determines the number of grid lines for plots of complex grids.
Should be a divisor of the the matrix size minus 1 (number of
subintervals). cgrid can be a vector [cx,cy].
```

Overview

The function can plot

- expressions, call collections or functions of one variable,
- parametric curves,
- x data against y data,
- implicit functions,
- bar plots,
- complex grids,
- polygons.

If a function or expression for xv is given, plot2d() will compute values in the given range using the function or expression. The expression must be an expression in the variable x. The range must be defined in the parameters a and b unless the default range should be used. The y-range will be computed automatically, unless c and d are specified, or a radius r, which yields the range
r,r
for x and y. For plots of functions, plot2d will use an adaptive evaluation of the function by default. To speed up the plot for complicated functions, switch this off with <adaptive, and optionally decrease the number of intervals n. Moreover, plot2d() will by default use mapping. I.e., it will compute the plot element for element. If your expression or your functions can handle a vector x, you can switch that off with <maps for faster evaluation.

Note that adaptive plots are always computed element for element. If functions or expressions for both xv and for yv are specified, plot2d() will compute a curve with the xv values as x-coordinates and the yv values as y-coordinates. In this case, a range should be defined for the parameter using xmin, xmax. Expressions contained in strings must always be expressions in the parameter variable x.

BAB 4

MENGGUNAKAN EMT UNTUK MENGGAMBAR PLOT 3D

Menggambar Plot 3D dengan EMT

Ini adalah pengenalan plot 3D di Euler. Kita memerlukan plot 3D untuk memvisualisasikan fungsi dari dua variabel.

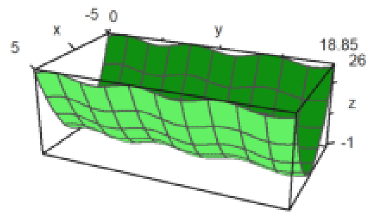
Euler menggambar fungsi-fungsi tersebut dengan menggunakan algoritme pengurutan untuk menyembunyikan bagian-bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Standarnya adalah dari kuadran x-y positif ke arah asal x=y=z=0, tetapi sudut=0° terlihat dari arah sumbu-y. Sudut pandang dan ketinggian dapat diubah.

Euler dapat merencanakan

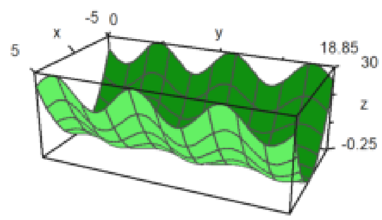
- permukaan dengan bayangan dan garis level atau rentang level,
- awan titik-titik,
- kurva parametrik,
- permukaan implisit.

Plot 3D dari sebuah fungsi menggunakan plot3d. Cara termudah adalah dengan memplot ekspresi dalam x dan y. Parameter r mengatur rentang plot di sekitar (0,0).

```
>aspect(1.5); plot3d("x^2+sin(y)",-5,5,0,6*pi):
```




```
>plot3d("x^2+x*sin(y)",-5,5,0,6*pi):
```



Silakan lakukan modifikasi agar gambar "talang bergelombang" tersebut tidak lurus melainkan melengkung/melingkar, baik melingkar secara mendatar maupun melingkar turun/naik (seperti papan peluncur pada kolam renang. Temukan rumusnya.

Fungsi dari Dua Variabel

Untuk grafik fungsi, gunakan

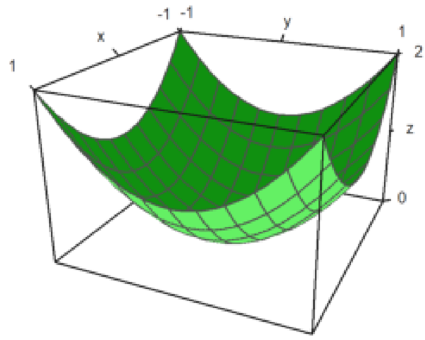
- ekspresi sederhana dalam x dan y,
- nama fungsi dari dua variabell
- atau matriks data.

Standarnya adalah kisi-kisi kawat yang terisi dengan warna yang berbeda pada kedua sisinya. Perhatikan, bahwa jumlah interval kisi-kisi default adalah 10, tetapi plot menggunakan jumlah default 40x40 persegi panjang untuk membangun permukaan. Hal ini dapat diubah.

- n = 40, n = [40,40]: jumlah garis kisi di setiap arah
- grid=10, grid=[10,10]: jumlah garis kisi di setiap arah.

Kami menggunakan default n=40 dan grid=10.

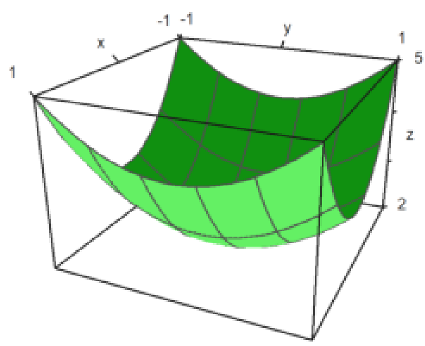
```
>plot3d("x^2+y^2"):
```



Contoh lain:
Akan digambarkan grafik

$$2x^2 + y^2 + 2$$

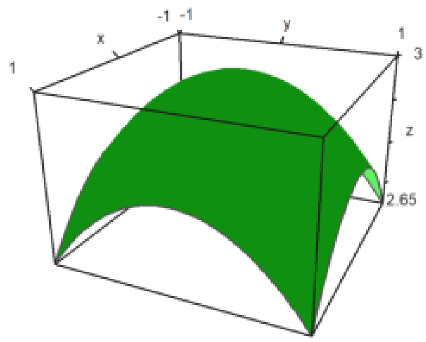
```
>plot3d("2*x^2+y^2+2", n=50, grid=5):
```



Gambar di atas menampilkan grafik fungsi dengan n = 50 dan grid = 5.
Akan digambarkan grafik fungsi

$$x^2 + y^2 - 9$$

```
>plot3d("(9-x^2-y^2)^(1/2)", n=40, grid=1):
```



Secara manual, cara menggambar grafik fungsi di 3D yaitu
Kita misalkan

$$z = \sqrt{16 - (x^2 + y^2)}$$

dan

$$z \geq 0$$

Jika kedua ruas dikuadratkan dan sederhanakan, maka akan diperoleh

$$x^2 + y^2 + z^2 = 9$$

yang merupakan persamaan sebuah bola.
Selanjutnya akan dicari jejak pada bidang koordinat
-bidang XOY(z=0):

$$x^2 + y^2 = 9$$

berupa lingkaran dengan pusat(0,0) dan jari-jari 3
-bidang YOZ(x=0)

$$y^2 + z^2 = 9$$

berupa lingkaran dengan pusat(0,0) dan jari-jari 3
-bidang XOZ(y=0)

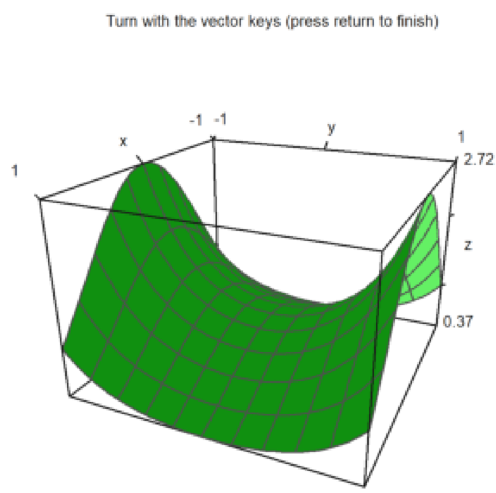
$$x^2 + z^2 = 9$$

berupa lingkaran dengan pusat(0,0) dan jari-jari 3

Interaksi pengguna dapat dilakukan dengan parameter >user. Pengguna dapat menekan tombol berikut ini.

- kiri, kanan, atas, bawah: putar sudut pandang
- +,-: memperbesar atau memperkecil
- a: menghasilkan anaglyph (lihat di bawah)
- l: sakelar untuk memutar sumber cahaya (lihat di bawah)
- spasi: setel ulang ke default
- kembali: mengakhiri interaksi

```
>plot3d("exp(-x^2+y^2)",>user, ...
>title="Turn with the vector keys (press return to finish)"):
```



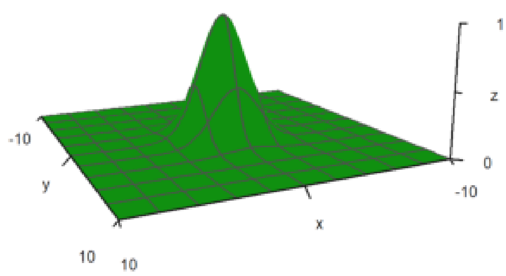
Rentang plot untuk fungsi dapat ditentukan dengan

- a, b: rentang x
- c, d: rentang y
- r: bujur sangkar simetris di sekitar (0,0).
- n: jumlah subinterval untuk plot.

Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

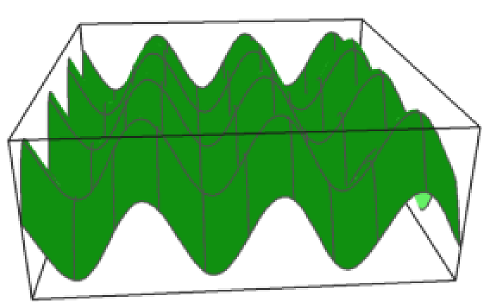
fscale: skala ke nilai fungsi (defaultnya adalah <fscale).
skala: angka atau vektor 1x2 untuk menskalakan ke arah x dan y.
frame: jenis bingkai (default 1).

```
>plot3d("exp(-(x^2+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3,>user):
```



Contoh lain:

```
>plot3d("sin(x)+sin(y)",r=10,n=80,fscale=4,scale=1,frame=2,>user):
```



Tampilan dapat diubah dengan berbagai cara.

- jarak: jarak pandang ke plot.
- zoom: nilai zoom.
- sudut: sudut ke sumbu y negatif dalam radian.
- height: ketinggian tampilan dalam radian.

Nilai default dapat diperiksa atau diubah dengan fungsi view(). Fungsi ini mengembalikan parameter sesuai urutan di atas.

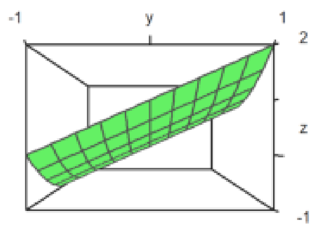
```
>view
```

```
  [5,   2.6,   2,   0.4]
```

Jarak yang lebih dekat membutuhkan zoom yang lebih sedikit. Efeknya lebih seperti lensa sudut lebar.

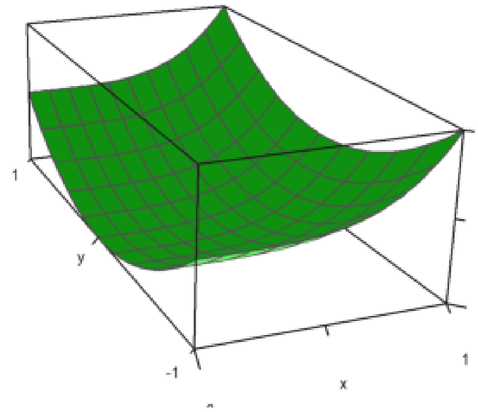
Pada contoh berikut ini, sudut = 0 dan tinggi = 0 terlihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

```
>plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0):
```



Plot terlihat selalu ke bagian tengah kubus plot. Anda dapat memindahkan bagian tengah dengan parameter center.

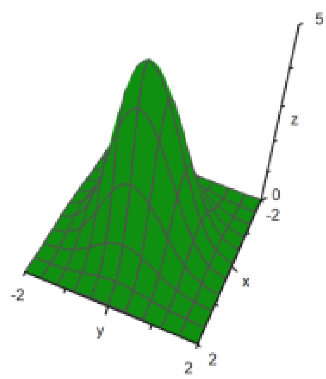
```
>plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...
>  center=[0.4,0,0],zoom=5):
```



Plot diskalakan agar sesuai dengan kubus satuan untuk dilihat. Jadi, tidak perlu mengubah jarak atau melakukan zoom, tergantung pada ukuran plot. Namun demikian, label mengacu ke ukuran yang sesungguhnya.

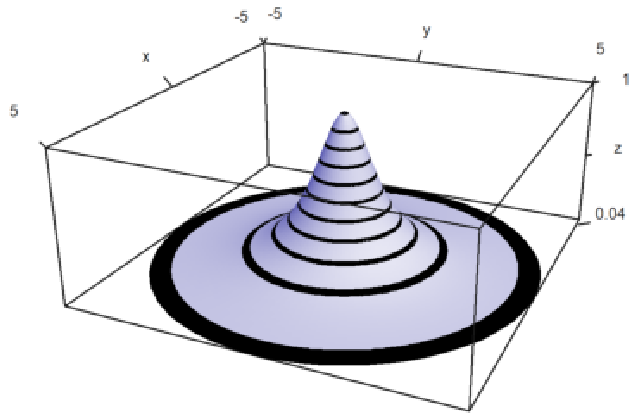
Jika Anda menonaktifkannya dengan scale=false, Anda harus berhati-hati, agar plot tetap muat ke dalam jendela plotting, dengan mengubah jarak pandang atau zoom, dan memindahkan bagian tengahnya.

```
>plot3d("5*exp(-x^2-y^2)",r=2,<fscale,<scale,distance=13,height=50°, ...
>  center=[0,0,-2],frame=3):
```

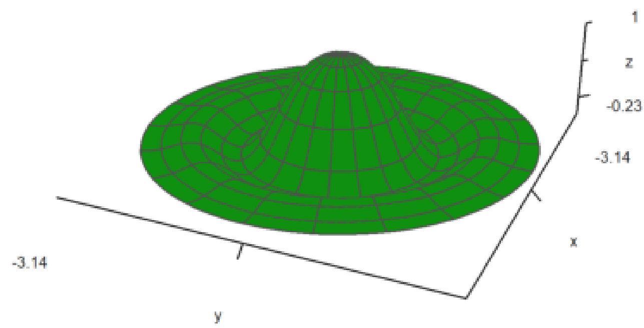


Plot polar juga tersedia. Parameter polar=true menggambar plot polar. Fungsi harus tetap merupakan fungsi dari x dan y. Parameter "fscale" menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi akan diskalakan agar sesuai dengan kubus.

```
>plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=blue):
```



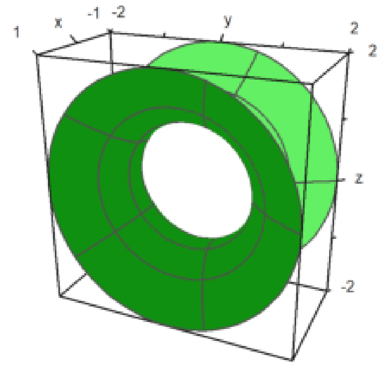

```
>function f(r) := exp(-r/2)*cos(r); ...
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4):
```



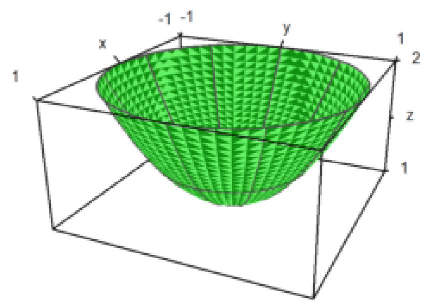
Parameter rotate memutar fungsi dalam x di sekitar sumbu x.

- rotate = 1: Menggunakan sumbu x
- rotate = 2: Menggunakan sumbu z

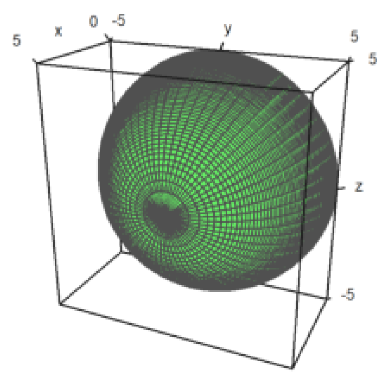
```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



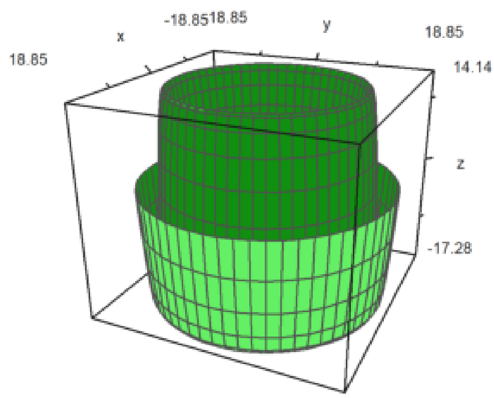

```
>plot3d("x^2+1",a=-1,b=1,rotate=2,grid=5):
```




```
>plot3d("sqrt(25-x^2)",a=0,b=5,rotate=1):
```

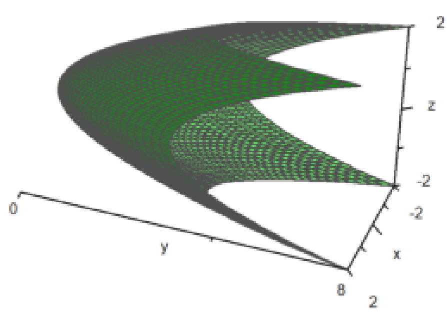


```
>plot3d("x*sin(x)",a=0,b=6pi,rotate=2):
```



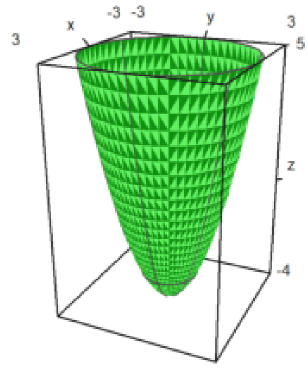
Berikut ini adalah plot dengan tiga fungsi.

```
>plot3d("x","x^2+y^2","y",r=2,zoom=3.5,frame=3):
```

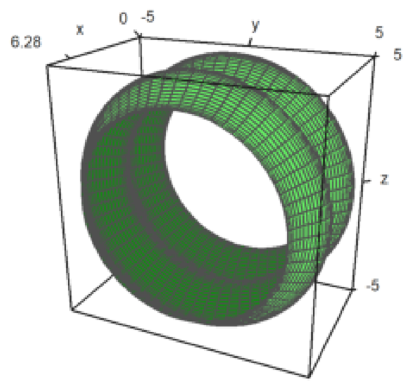


Contoh lain:

```
>plot3d("x^2-4", a=-3, b=3, rotate=2, grid=3):
```

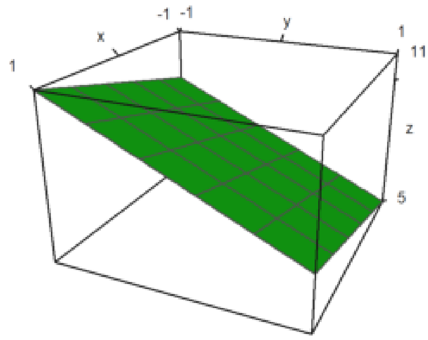



```
>plot3d("sin(x)^2+4", a=0, b=2pi, rotate=1):
```

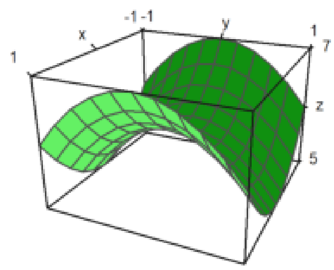


Contoh jika fungsi disimpan dalam variabel ekspresi

```
>function f(x,y):=x-2*y+8
>plot3d("f(x,y)", grid=5):
```




```
>function g(x,y):=x^2-y^2+6
>plot3d("g(x,y)", grid=10, n=10, zoom=2):
```



Plot Kontur

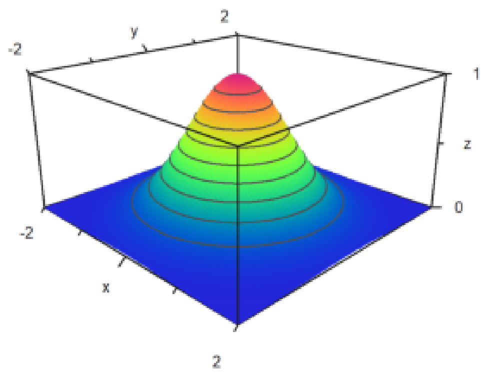
Untuk plot, Euler menambahkan garis kisi-kisi. Sebagai gantinya, dimungkinkan untuk menggunakan garis level dan rona satu warna atau rona berwarna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan bayangan. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah / cyan.

- > hue: Mengaktifkan bayangan cahaya, bukan kabel.
- >contour: Memplot garis kontur otomatis pada plot.
- level=... (atau level): Vektor nilai untuk garis kontur.

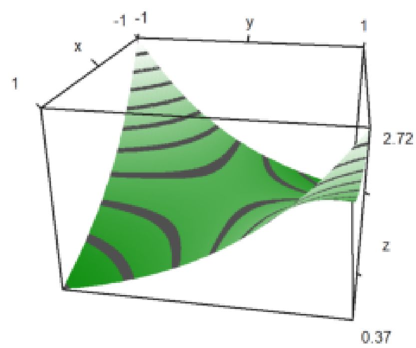
Standarnya adalah level = "auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat dalam plot, level sebenarnya adalah kisaran level.

Gaya default dapat diubah. Untuk plot kontur berikut ini, kami menggunakan grid yang lebih halus untuk titik-titik 100x100, skala fungsi dan plot, dan menggunakan sudut pandang yang berbeda.

```
>plot3d("exp(-x^2-y^2)",r=2,n=100,level="thin", ...
> >contour,>spectral,fscale=1,scale=1.1,angle=45°,height=20°):
```




```
>plot3d("exp(x*y)",angle=100°,>contour,color=green):
```

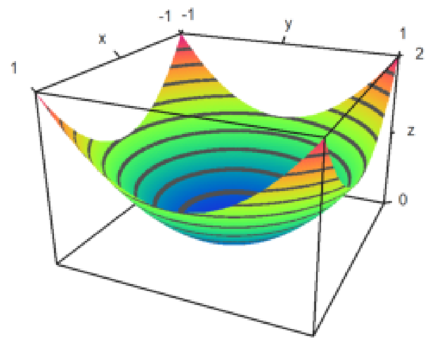


Bayangan default menggunakan warna abu-abu. Tetapi, kisaran warna spektral juga tersedia.

- >spectral: Menggunakan skema spektral default
- color =...: Menggunakan warna khusus atau skema spektral

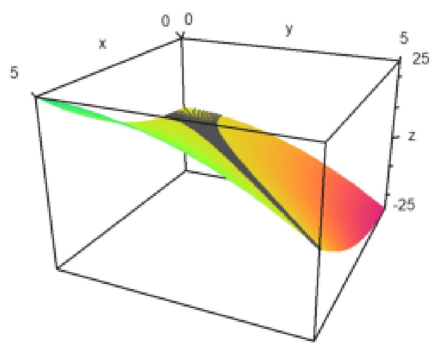
Untuk plot berikut ini, kami menggunakan skema spektral default dan menambah jumlah titik untuk mendapatkan tampilan yang sangat mulus.

```
>plot3d("x^2+y^2",>spectral,>contour,n=100):
```



Alih-alih garis level otomatis, kita juga dapat menetapkan nilai garis level. Hal ini akan menghasilkan garis level yang tipis, alih-alih rentang level.

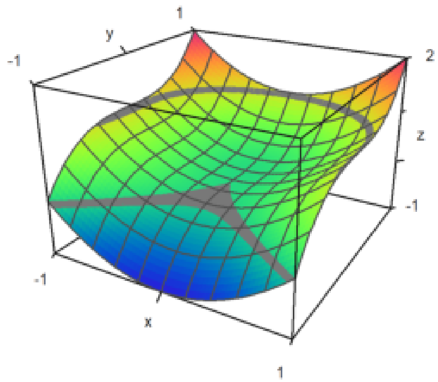
```
>plot3d("x^2-y^2",0,5,0,5,level=-1:0.1:1,color=redgreen):
```



Pada plot berikut, kami menggunakan dua pita level yang sangat luas dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas-batas level sebagai kolom.

Selain itu, kami menghamparkan kisi-kisi dengan 10 interval di setiap arah.

```
>plot3d("x^2+y^3",level=[-0.1,0.9;0,1], ...
>  >spectral,angle=30°,grid=10,contourcolor=gray):
```

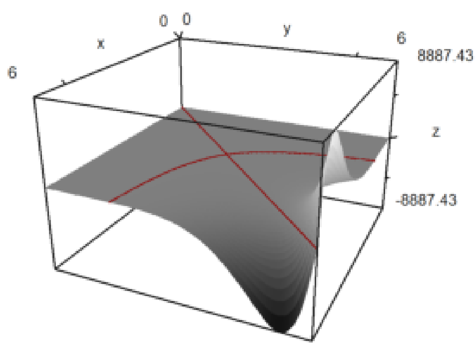


Dalam contoh berikut, kami memplot himpunan, di mana

$$f(x, y) = x^y - y^x = 0$$

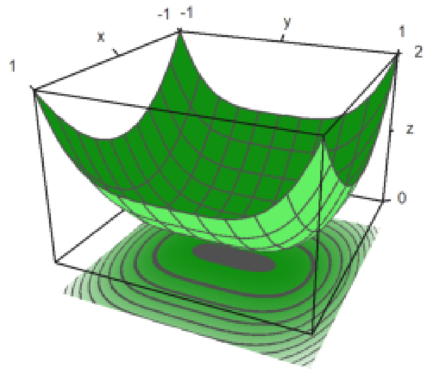
Kami menggunakan satu garis tipis untuk garis level.

```
>plot3d("x^y-y^x",level=0,a=0,b=6,c=0,d=6,contourcolor=red,n=100):
```



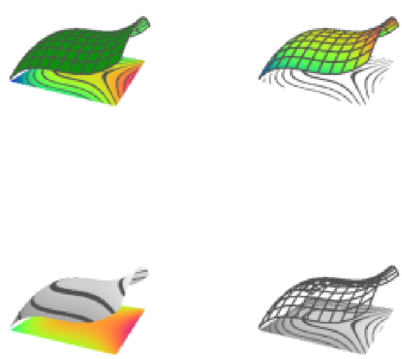
Dimungkinkan untuk menampilkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan.

```
>plot3d("x^2+y^4",>cp,cpcolor=green,cpdelta=0.2):
```



Berikut ini beberapa gaya lainnya. Kami selalu mematikan bingkai, dan menggunakan berbagai skema warna untuk plot dan kisi-kisi.

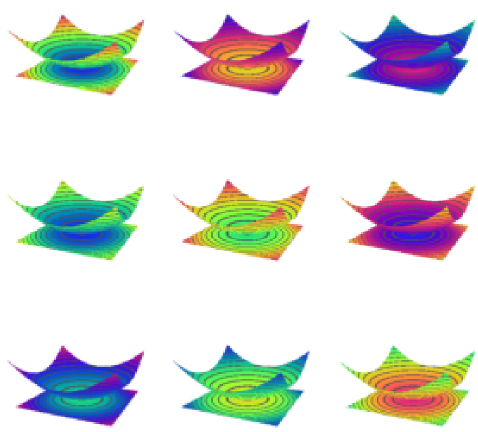
```
>figure(2,2); ...
>expr="y^3-x^2"; ...
>figure(1);  ...
>  plot3d(expr,<frame,>cp,cpcolor=spectral); ...
>figure(2);  ...
>  plot3d(expr,<frame,>spectral,grid=10,cp=2); ...
>figure(3);  ...
>  plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...
>figure(4);  ...
>  plot3d(expr,<frame,>hue,grid=10,>transparent,>cp,cpcolor=gray); ...
>figure(0):
```



Ada beberapa skema spektral lainnya, yang diberi nomor dari 1 hingga 9. Tetapi Anda juga dapat menggunakan color=value, di mana value

- spektral: untuk rentang dari biru ke merah
- putih: untuk rentang yang lebih redup
- kuningbiru, ungu-hijau, biru-kuning, hijau-merah
- biru-kuning, hijau-ungu, kuning-biru, merah-hijau

```
>figure(3,3); ...
>for i=1:9;  ...
>  figure(i); plot3d("x^2+y^2",spectral=i,>contour,>cp,<frame,zoom=4);  ...
>end; ...
>figure(0):
```

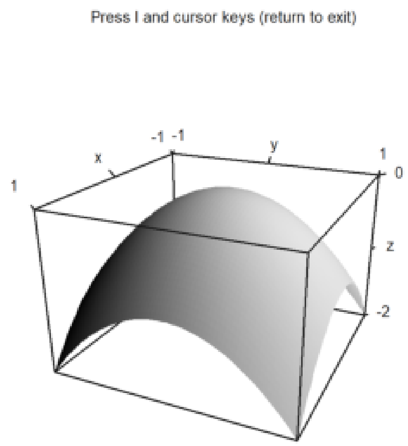


Sumber cahaya dapat diubah dengan l dan tombol kursor selama interaksi pengguna. Ini juga dapat ditetapkan dengan parameter.

- cahaya: arah untuk cahaya
- amb: cahaya sekitar antara 0 dan 1

Perhatikan, bahwa program ini tidak membuat perbedaan di antara sisi-sisi plot. Tidak ada bayangan. Untuk ini, Anda memerlukan Povray.

```
>plot3d("-x^2-y^2", ...
>  hue=true,light=[0,1,1],amb=0,user=true, ...
>  title="Press l and cursor keys (return to exit)"):
```

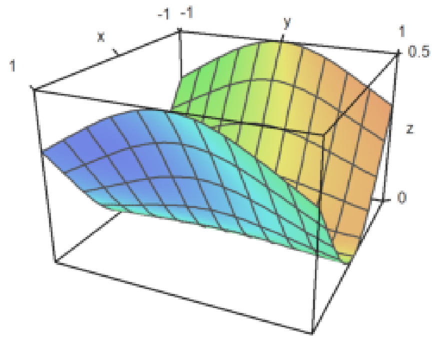


Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah.

```
>plot3d("-x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...
>  zoom=3,contourcolor=red,level=-2:0.1:1,dl=0.01):
```

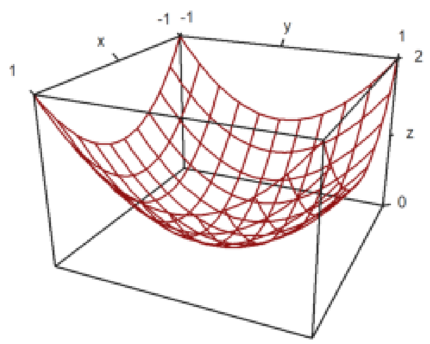
Warna 0 memberikan efek pelangi yang istimewa.

```
>plot3d("x^2/(x^2+y^2+1)",color=0,hue=true,grid=10):
```



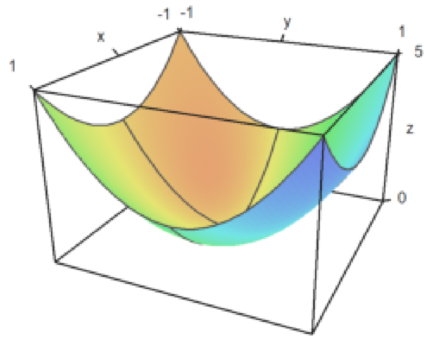
Permukaannya juga bisa transparan.

```
>plot3d("x^2+y^2",>transparent,grid=10,wirecolor=red):
```

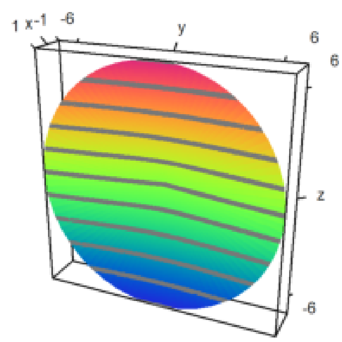


Contoh lain:

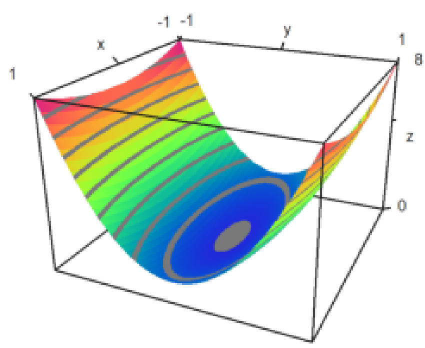
```
>plot3d("2*x^2+3*y^2",color=0,hue=true,grid=2):
```




```
>plot3d("x^2+5*x", >spectral, >contour, n=100, rotate=1):
```




```
>plot3d("x^2+7*y^2", >spectral, >contour, n=50):
```



Ada juga plot implisit dalam tiga dimensi. Euler menghasilkan potongan melalui objek. Fitur plot3d termasuk plot implisit. Plot-plot ini menunjukkan himpunan nol dari sebuah fungsi dalam tiga variabel.
Solusi dari

$$f(x, y, z) = 0$$

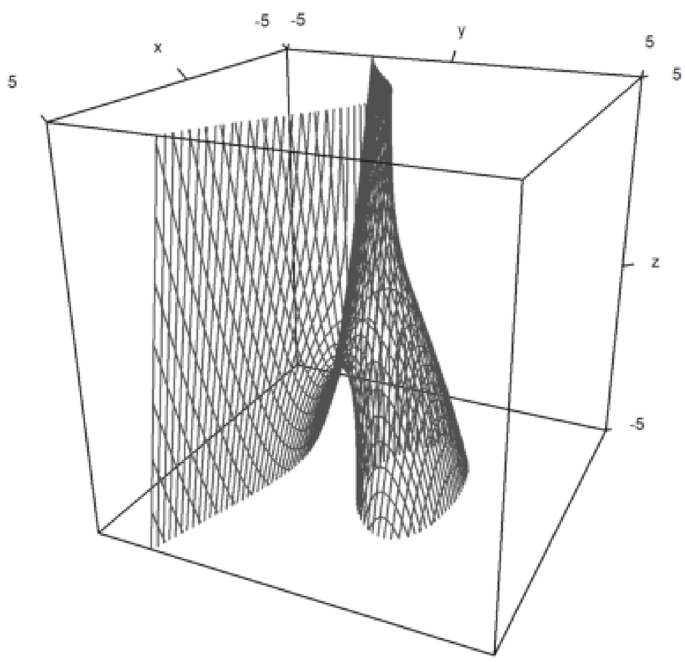
dapat divisualisasikan dalam potongan yang sejajar dengan bidang x-y, bidang x-z dan bidang y-z.

- implisit = 1: potong sejajar dengan bidang y-z
- implicit=2: potong sejajar dengan bidang x-z
- implicit = 4: potong sejajar dengan bidang x-y

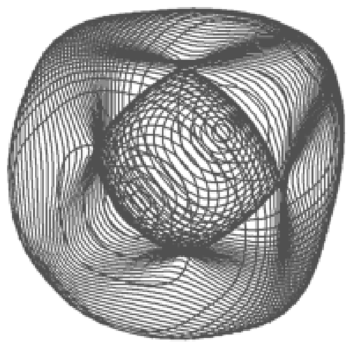
Tambahkan nilai-nilai ini, jika Anda mau. Dalam contoh, kami memplot

$$M = \{(x, y, z) : x^2 + y^3 + zy = 1\}$$

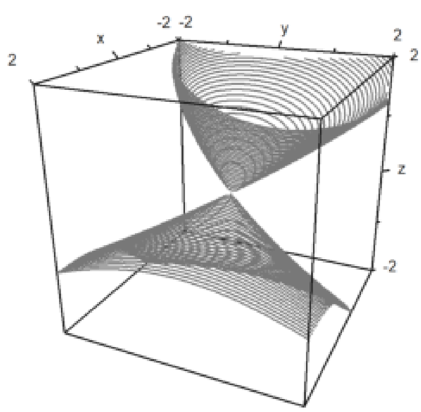
```
>plot3d("x^2+y^3+z*y-1",r=5,implicit=3):
```



```
>c=1; d=1;
>plot3d("((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-
```

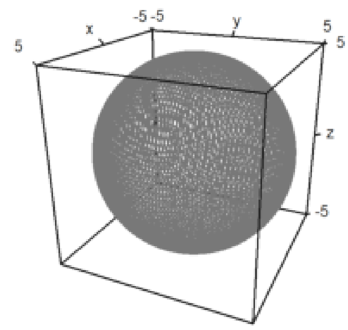


```
>plot3d("x^2+y^2+4*x*z+z^3",>implicit,r=2,zoom=2.5):
```

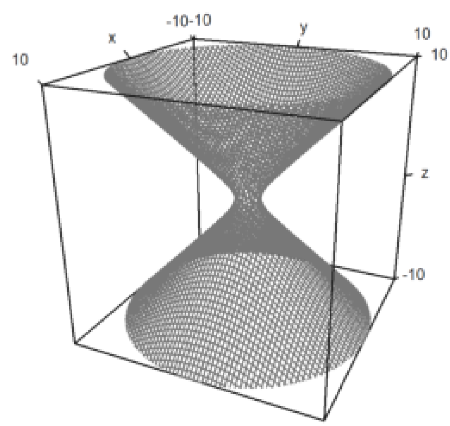


Contoh lain:

```
>plot3d("x^2+y^2+z^2-25",r=5,implicit=3, zoom=2):
```




```
>function h(x,y,z):=x^2+y^2-z^2-1
>plot3d("h(x,y,z)", r=10, implicit=3):
```



Memplot Data 3D

Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, Anda perlu menyediakan matriks nilai x, y, dan z, atau tiga fungsi atau ekspresi fx(x,y), fy(x,y), fz(x,y).

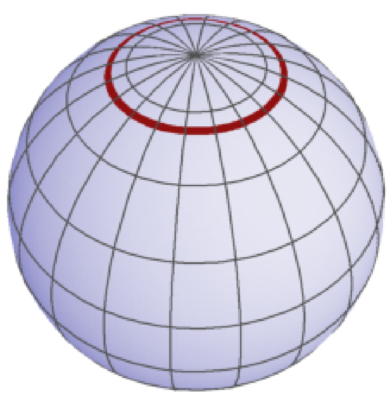
$$\gamma(t,s) = (x(t,s), y(t,s), z(t,s))$$

Karena x, y, z adalah matriks, kita asumsikan bahwa (t, s) berjalan melalui kisi-kisi persegi. Hasilnya, Anda dapat memplot gambar persegi panjang dalam ruang.

Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

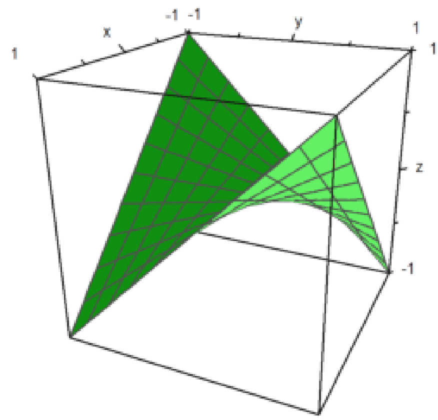
Pada contoh berikut, kita menggunakan vektor nilai t dan vektor kolom nilai s untuk memparameterkan permukaan bola. Pada gambar kita dapat menandai daerah, dalam kasus kita daerah kutub.

```
>t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ...
>x=cos(s)*cos(t); y=cos(s)*sin(t); z=sin(s); ...
>plot3d(x,y,z,>hue, ...
>color=blue,<frame,grid=[10,20], ...
>values=s,contourcolor=red,level=[90°-24°;90°-22°], ...
>scale=1.4,height=50°):
```



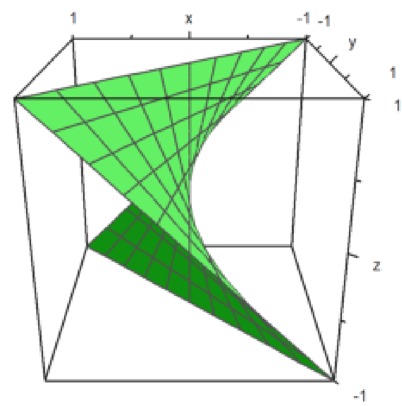
Berikut ini adalah contoh, yang merupakan grafik suatu fungsi.

```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):
```



Namun demikian, kita bisa membuat segala macam permukaan. Berikut ini adalah permukaan yang sama dengan suatu fungsi

```
>plot3d(t*s,t,s,angle=180°,grid=10):
```



Dengan lebih banyak upaya, kita bisa menghasilkan banyak permukaan.

Dalam contoh berikut ini, kami membuat tampilan berbayang dari bola yang terdistorsi. Koordinat biasa untuk bola adalah

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

dengan

$$0 \le t \le 2\pi, \quad \frac{-\pi}{2} \le s \le \frac{\pi}{2}.$$

Kami mengurangi hal ini dengan faktor

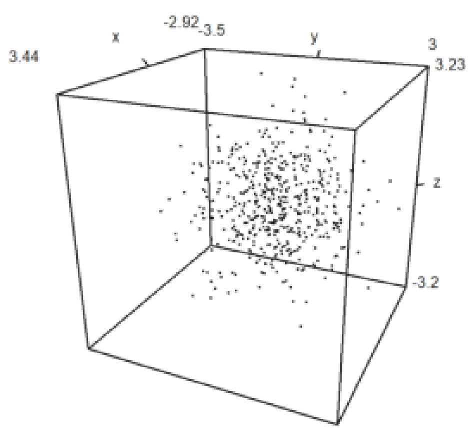
$$d(t,s) = \frac{\cos(4t) + \cos(8s)}{4}.$$

```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
>d=1+0.2*(cos(4*t)+cos(8*s)); ...
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
>  light=[1,0,1],frame=0,zoom=5):
```

Tentu saja, awan titik juga dimungkinkan. Untuk memplot data titik dalam ruang, kita memerlukan tiga vektor untuk koordinat titik.

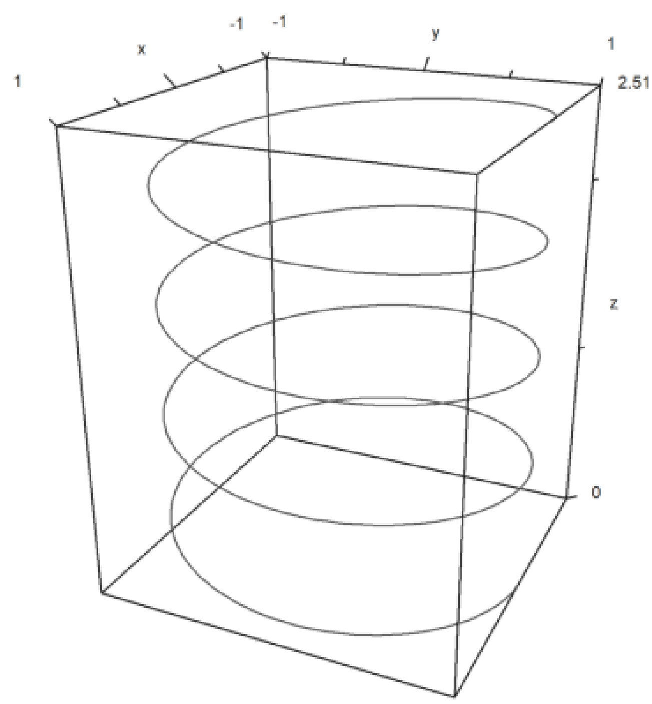
Gaya sama seperti di plot2d dengan poin=true;

```
>n=500;  ...
>  plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```

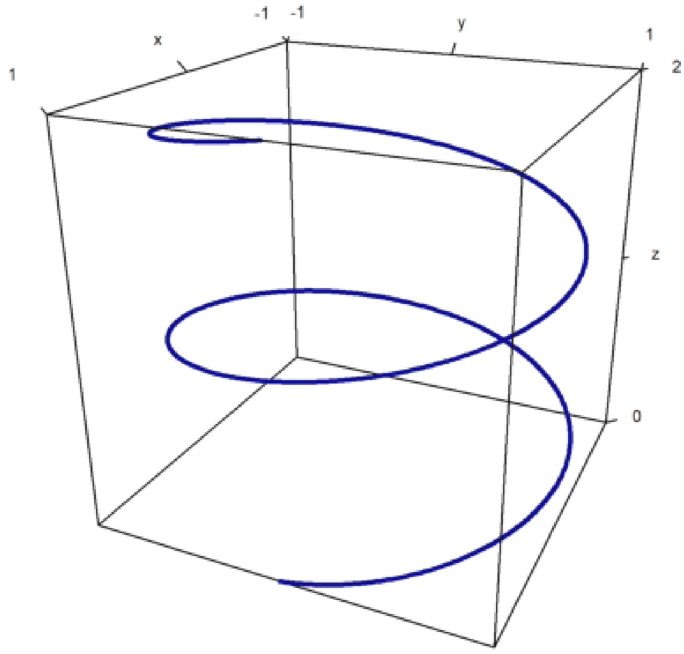


Anda juga dapat memplot kurva dalam bentuk 3D. Dalam hal ini, akan lebih mudah untuk menghitung titik- titik kurva. Untuk kurva pada bidang, kami menggunakan urutan koordinat dan parameter wire = true.

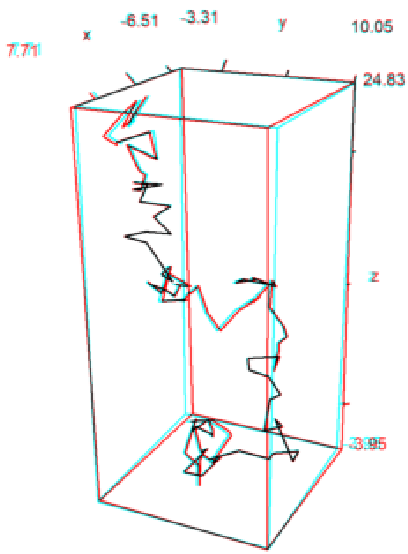
```
>t=linspace(0,8pi,500); ...
>plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
```




```
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...
>linewidth=3,wirecolor=blue):
```

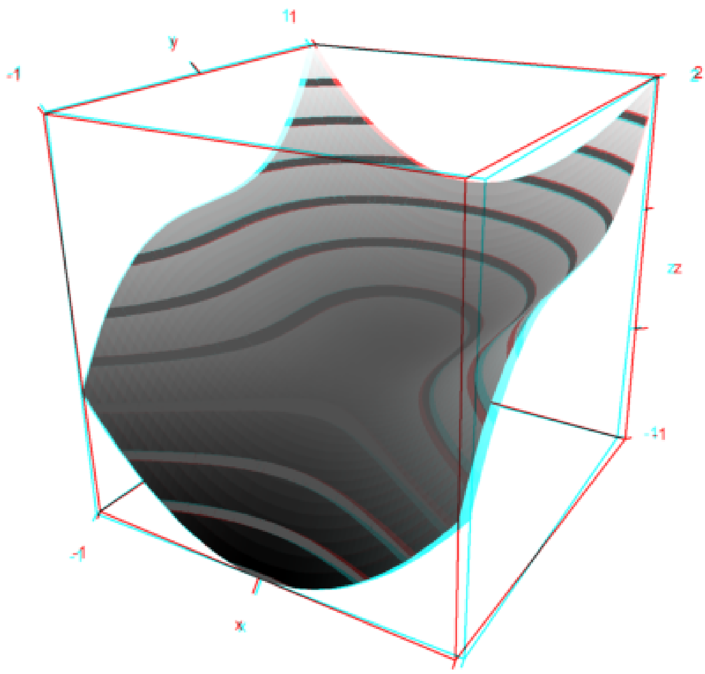



```
>X=cumsum(normal(3,100)); ...
> plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```



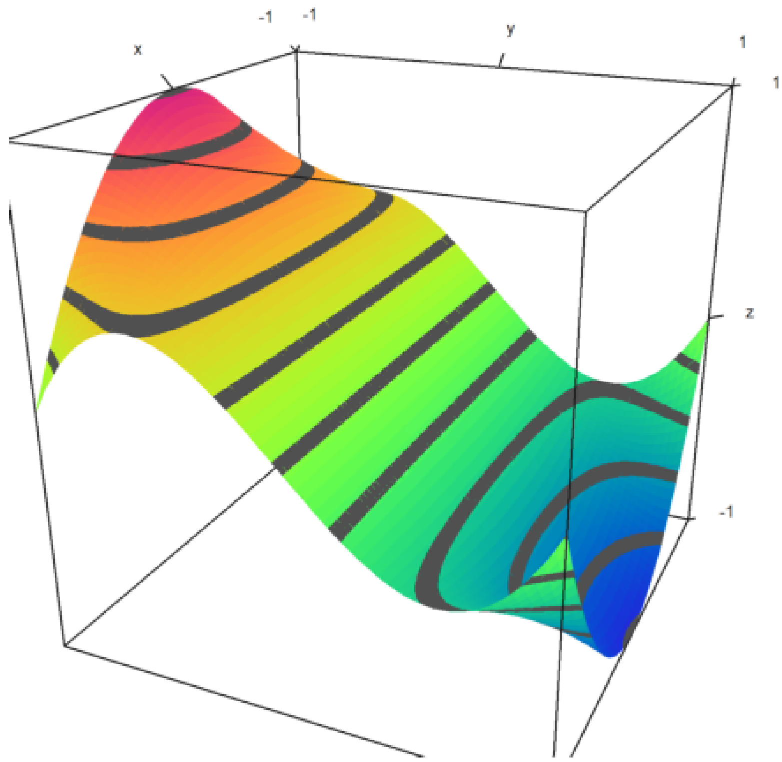
EMT juga dapat membuat plot dalam mode anaglyph. Untuk melihat plot semacam itu, Anda memerlukan kacamata merah/cyan.

```
> plot3d("x^2+y^3",>anaglyph,>contour,angle=30°):
```



Sering kali, skema warna spektral digunakan untuk plot. Hal ini menekankan ketinggian fungsi.

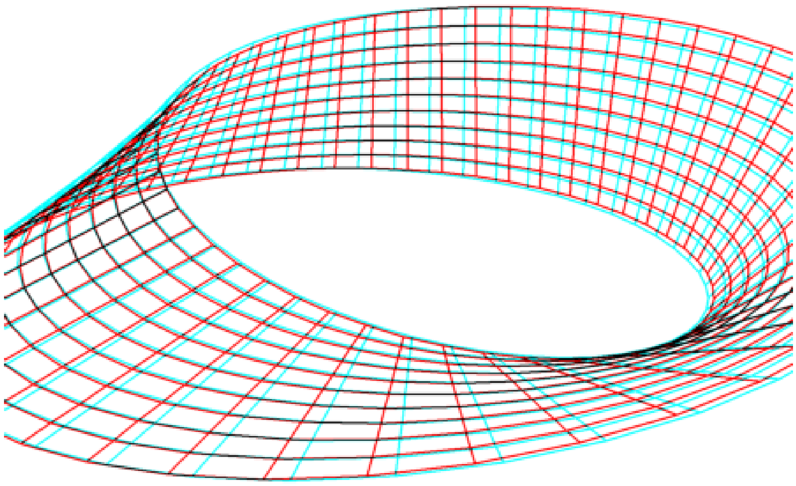
```
>plot3d("x^2*y^3-y",>spectral,>contour,zoom=3.2):
```



Euler juga dapat memplot permukaan yang diparameterkan, apabila parameternya adalah nilai x, y, dan z dari gambar kisi-kisi persegi panjang di dalam ruang.

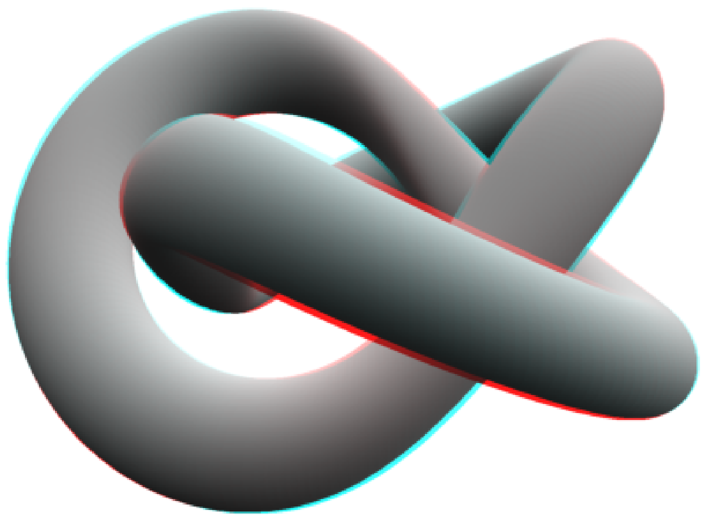
Untuk demo berikut ini, kita akan menyiapkan parameter u dan v, dan menghasilkan koordinat ruang dari parameter ini.

```
>u=linspace(-1,1,10); v=linspace(0,2*pi,50)'; ...
>X=(3+u*cos(v/2))*cos(v); Y=(3+u*cos(v/2))*sin(v); Z=u*sin(v/2); ...
>plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3):
```



Berikut ini contoh yang lebih rumit, yang tampak megah dengan kacamata merah/cyan.

```
>u:=linspace(-pi,pi,160); v:=linspace(-pi,pi,400)';  ...
>x:=(4*(1+.25*sin(3*v))+cos(u))*cos(2*v); ...
>y:=(4*(1+.25*sin(3*v))+cos(u))*sin(2*v); ...
> z=sin(u)+2*cos(3*v); ...
>plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=2.8,>anaglyph):
```



Plot Statistika

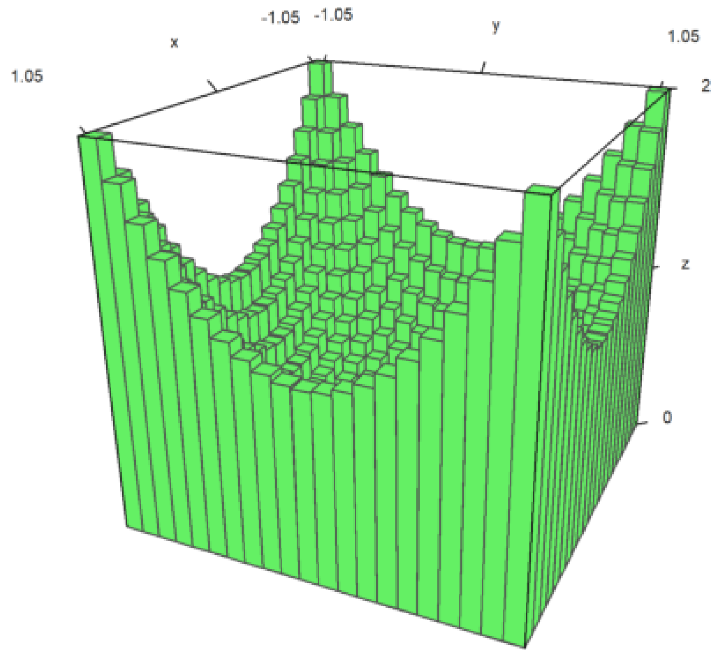
Petak batang juga dimungkinkan. Untuk ini, kita harus menyediakan

- x: vektor baris dengan n+1 elemen
- y: vektor kolom dengan n+1 elemen
- z: matriks nilai nxn.

z dapat lebih besar, tetapi hanya nilai nxn yang akan digunakan.

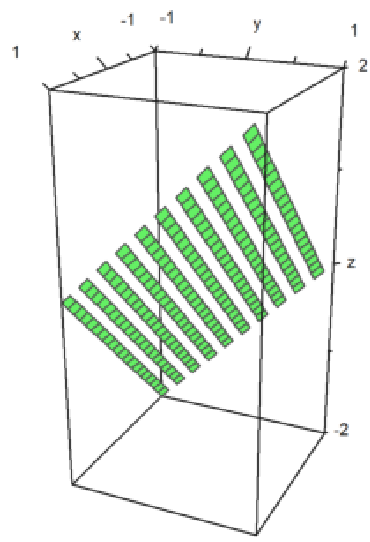
Dalam contoh, pertama-tama kita menghitung nilainya. Kemudian kita menyesuaikan x dan y, sehingga vektor berpusat pada nilai yang digunakan.

```
>x=-1:0.1:1; y=x'; z=x^2+y^2; ...
>xa=(x|1.1)-0.05; ya=(y_1.1)-0.05; ...
>plot3d(xa,ya,z,bar=true):
```



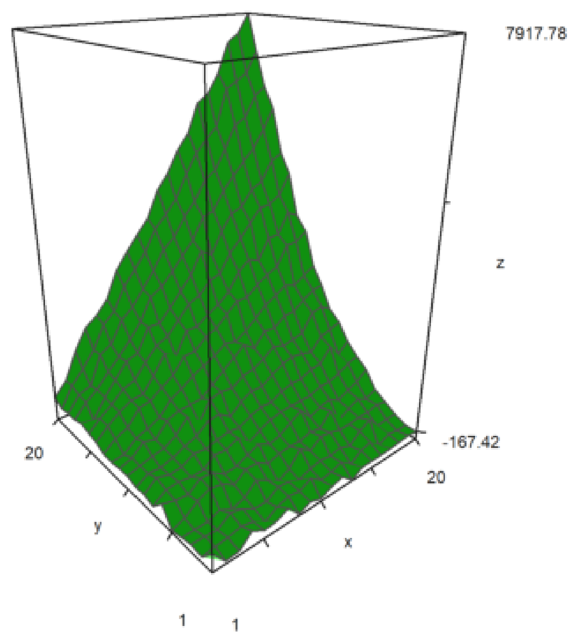
Hal ini memungkinkan untuk membagi plot permukaan menjadi dua bagian atau lebih.

```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...
>plot3d(x,y,z,disconnect=2:2:20):
```

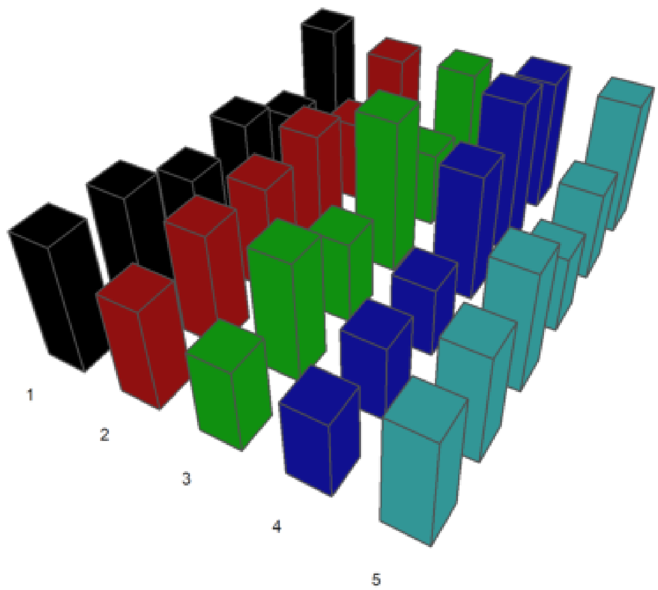


Jika memuat atau menghasilkan matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke [-1,1] dengan scale(M), atau menskalakan matriks dengan >zscale. Hal ini dapat dikombinasikan dengan faktor penskalaan individual yang diterapkan sebagai tambahan.

```
>i=1:20; j=i'; ...
>plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8
```

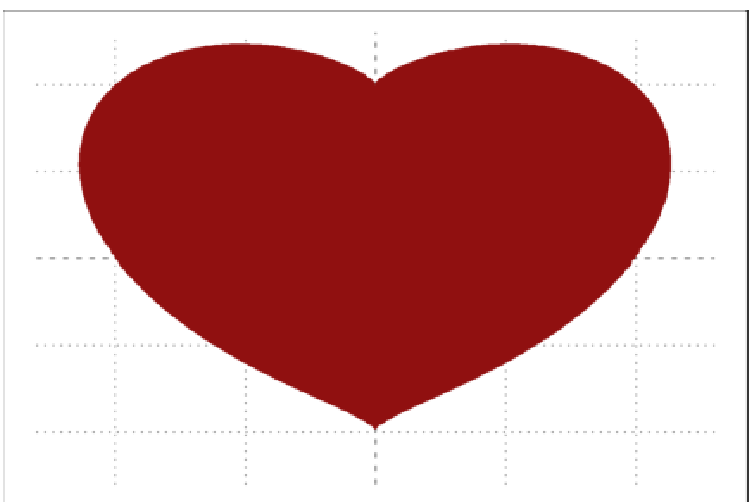



```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
>loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
>columnsplot3d(v',scols=1:5,ccols=[1:5]):
```



Permukaan Benda Putar

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



```
>ekspresi &= (x^2+y^2-1)^3-x^2*y^3; $ekspresi
```

$$\left(y^2+x^2-1\right)^3-x^2\,y^3$$

Kami ingin memutar kurva jantung di sekitar sumbu y. Inilah ekspresi yang mendefinisikan jantung:

$$f(x,y)=(x^2+y^2-1)^3-x^2.y^3.$$

Selanjutnya kami menetapkan

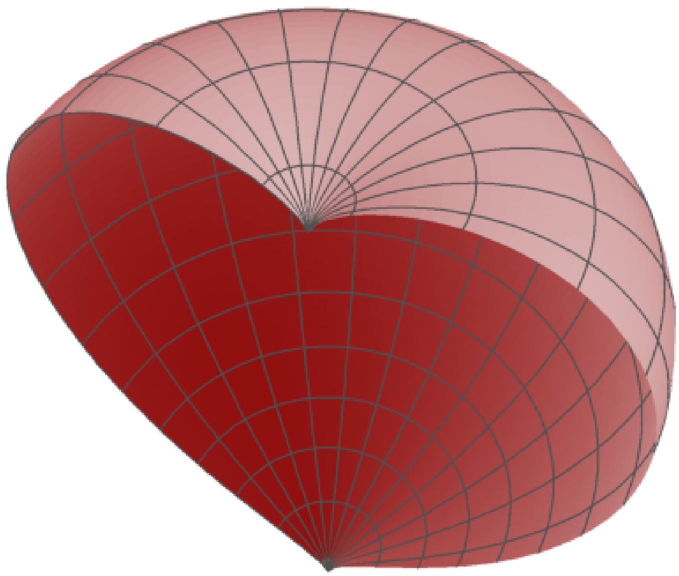
$$x=r.cos(a),\quad y=r.sin(a).$$

```
>function fr(r,a) &= ekspresi with [x=r*cos(a),y=r*sin(a)] | trigreduce; $f
```

$$\left(r^2-1\right)^3+\frac{\left(\sin\left(5\,a\right)-\sin\left(3\,a\right)-2\,\sin a\right)\,r^5}{16}$$

Hal ini memungkinkan untuk mendefinisikan fungsi numerik, yang menyelesaikan untuk r, jika a diberikan. Dengan fungsi tersebut kita dapat memplotkan jantung yang diputar sebagai permukaan parametrik.

```
>function map f(a) := bisect("fr",0,2;a); ...
>t=linspace(-pi/2,pi/2,100); r=f(t);  ...
>s=linspace(pi,2pi,100)'; ...
>plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...
>>hue,<frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):
```

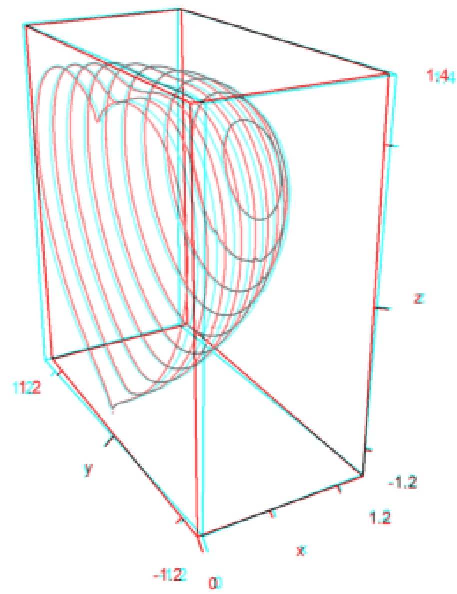


Berikut ini adalah plot 3D dari gambar di atas yang diputar mengelilingi sumbu-z. Kami mendefinisikan fungsi, yang menggambarkan objek.

```
>function f(x,y,z) ...
```

```
  r=x^2+y^2;
  return (r+z^2-1)^3-r*z^3;
   endfunction
```

```
>plot3d("f(x,y,z)", ...
>xmin=0,xmax=1.2,ymin=-1.2,ymax=1.2,zmin=-1.2,zmax=1.4, ...
>implicit=1,angle=-30°,zoom=2.5,n=[10,100,60],>anaglyph):
```



Plot 3D Khusus

Fungsi plot3d memang bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Di samping rutinitas yang lebih mendasar, Anda bisa mendapatkan plot berbingkai dari objek apa pun yang Anda sukai.

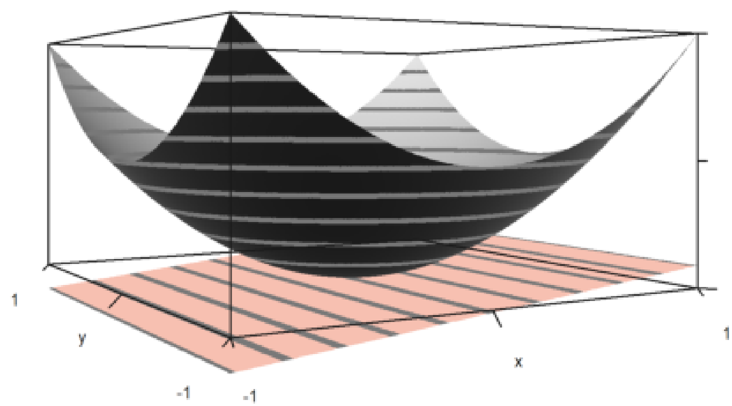
Meskipun Euler bukan program 3D, namun dapat menggabungkan beberapa objek dasar. Kami mencoba memvisualisasikan parabola dan garis singgungnya.

```
>function myplot ...
```

```
    y=-1:0.01:1; x=(-1:0.01:1)';
    plot3d(x,y,0.2*(x-0.1)/2,<scale,<frame,>hue, ..
      hues=0.5,>contour,color=orange);
    h=holding(1);
    plot3d(x,y,(x^2+y^2)/2,<scale,<frame,>contour,>hue);
    holding(h);
  endfunction
```

Sekarang framedplot() menyediakan frame, dan mengatur tampilan.

```
>framedplot("myplot",[-1,1,-1,1,0,1],height=0,angle=-30°, ...
>  center=[0,0,-0.7],zoom=3):
```

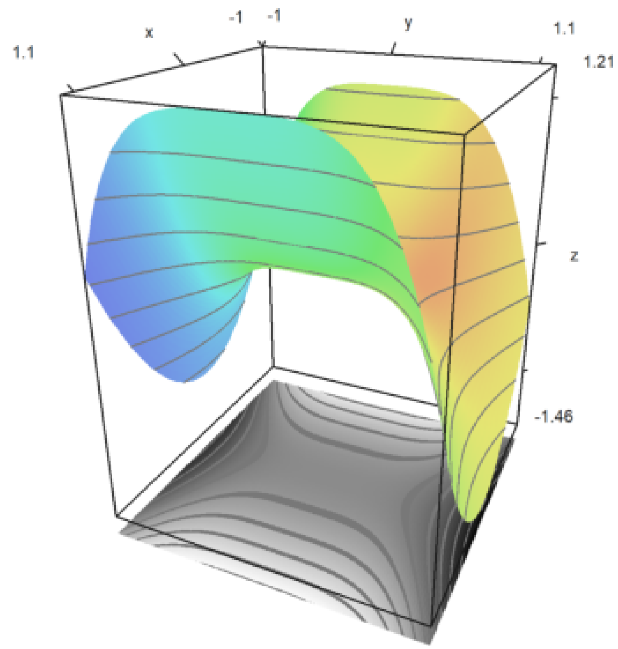


Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa plot3d() mengatur jendela ke fullwindow() secara default, namun plotcontourplane() mengasumsikannya.

```
>x=-1:0.02:1.1; y=x'; z=x^2-y^4;
>function myplot (x,y,z) ...
```

```
    zoom(2);
    wi=fullwindow();
    plotcontourplane(x,y,z,level="auto",<scale);
    plot3d(x,y,z,>hue,<scale,>add,color=white,level="thin");
    window(wi);
    reset();
  endfunction
```

```
>myplot(x,y,z):
```



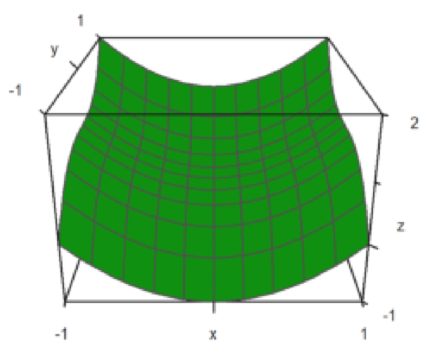
Animasi

Euler dapat menggunakan frame untuk melakukan pra-komputasi animasi.

Salah satu fungsi yang memanfaatkan teknik ini adalah rotate. Fungsi ini dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi ini memanggil addpage() untuk setiap plot baru. Terakhir, fungsi ini menganimasikan plot-plot tersebut.

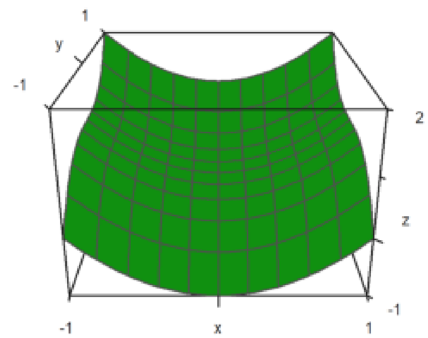
Silakan pelajari sumber rotasi untuk mengetahui detail selengkapnya.

```
>function testplot () := plot3d("x^2+y^3"); ...
>rotate("testplot"); testplot():
```



Contoh lain:

```
>function testplot2 () := plot3d("8*x^2+3*y^3");...
>rotate("testplot2"); testplot():
```



Menggambar Povray

Dengan bantuan file Euler povray.e, Euler dapat menghasilkan file Povray. Hasilnya sangat bagus untuk dilihat.

Anda perlu menginstal Povray (32bit atau 64bit) dari http://www.povray.org/, dan meletakkan sub- direktori "bin" dari Povray ke dalam jalur lingkungan, atau mengatur variabel "defaultpovray" dengan jalur lengkap yang mengarah ke "pvengine.exe".

Antarmuka Povray dari Euler menghasilkan file Povray di direktori home pengguna, dan memanggil Povray untuk mengurai file-file ini. Nama file default adalah current.pov, dan direktori defaultnya adalah eulerhome(), biasanya c:\Users\Username\Euler. Povray menghasilkan sebuah file PNG, yang dapat dimuat oleh Euler ke dalam notebook. Untuk membersihkan berkas-berkas ini, gunakan povclear().

Fungsi pov3d memiliki semangat yang sama dengan plot3d. Fungsi ini dapat menghasilkan grafik fungsi f(x,y), atau permukaan dengan koordinat X,Y,Z dalam matriks, termasuk garis level opsional. Fungsi ini memulai raytracer secara otomatis, dan memuat adegan ke dalam notebook Euler.

Selain pov3d(), ada banyak fungsi yang menghasilkan objek Povray. Fungsi-fungsi ini mengembalikan string, yang berisi kode Povray untuk objek. Untuk menggunakan fungsi-fungsi ini, mulai file Povray dengan povstart(). Kemudian gunakan writeln(...) untuk menulis objek ke file scene. Terakhir, akhiri file dengan povend(). Secara default, raytracer akan dimulai, dan PNG akan dimasukkan ke dalam notebook Euler.

Fungsi objek memiliki parameter yang disebut "look", yang membutuhkan string dengan kode Povray untuk tekstur dan hasil akhir objek. Fungsi povlook() dapat digunakan untuk menghasilkan string ini. Fungsi ini memiliki parameter untuk warna, transparansi, Phong Shading, dll.

Perhatikan bahwa alam semesta Povray memiliki sistem koordinat lain.
Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi, Anda dapat terus berpikir dalam sistem koordinat Euler dengan z menunjuk vertikal ke atas, dan sumbu x, y, z di tangan kanan.
Anda perlu memuat file povray.

```
>load povray;
```

Pastikan direktori bin povray berada di dalam path. Jika tidak, edit variabel berikut sehingga berisi jalur ke povray yang dapat dieksekusi.

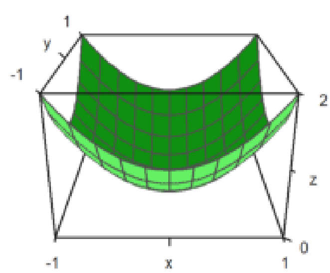
```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

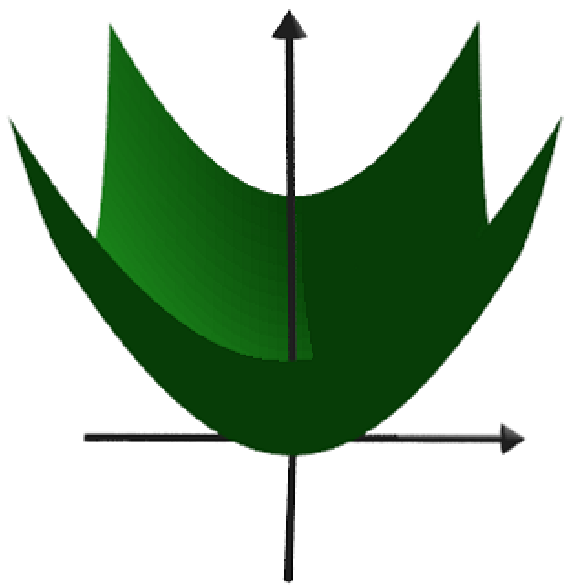
Untuk kesan pertama, kita plot sebuah fungsi sederhana. Perintah berikut ini menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk melacak sinar pada file ini.

Jika Anda menjalankan perintah berikut, GUI Povray akan terbuka, menjalankan file, dan menutup secara otomatis. Karena alasan keamanan, Anda akan ditanya apakah Anda ingin mengizinkan file exe untuk dijalankan. Anda dapat menekan cancel untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK pada jendela Povray untuk mengetahui dialog awal Povray.

```
>plot3d("x^2+y^2",zoom=2):
```

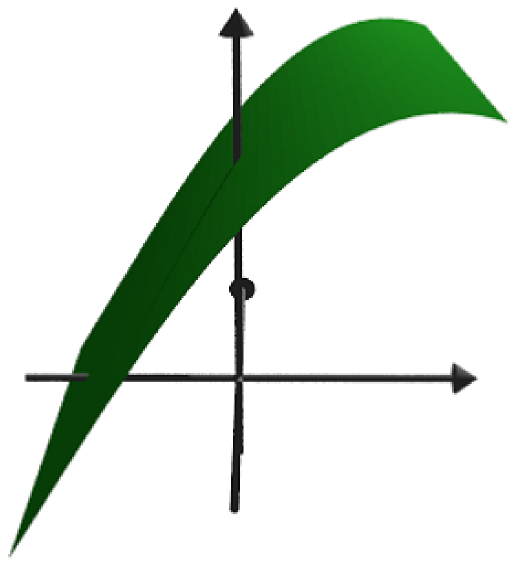


```
>pov3d("x^2+y^2",zoom=3);
```



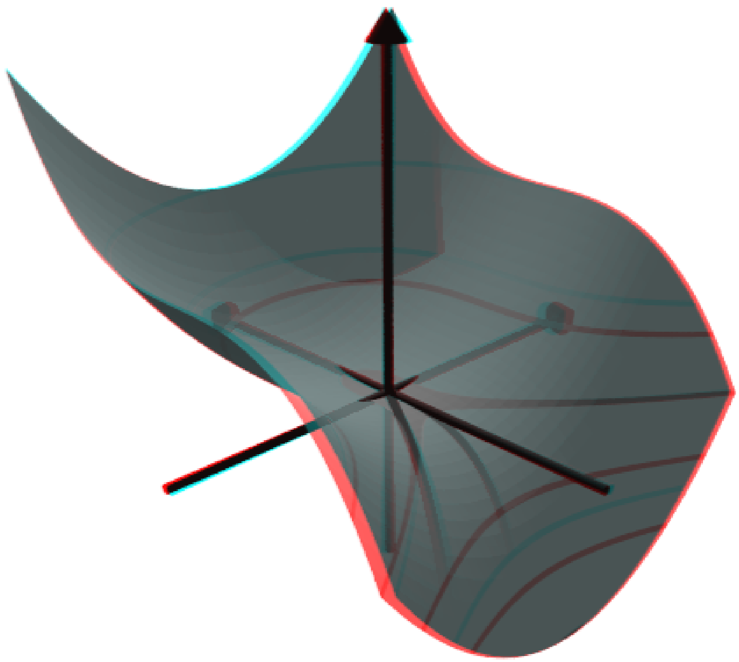
Contoh lain:

```
>pov3d("sin(x)+cos(x)", zoom=3);
```



Kita dapat membuat fungsi menjadi transparan dan menambahkan hasil akhir lainnya. Kita juga dapat menambahkan garis level ke plot fungsi.

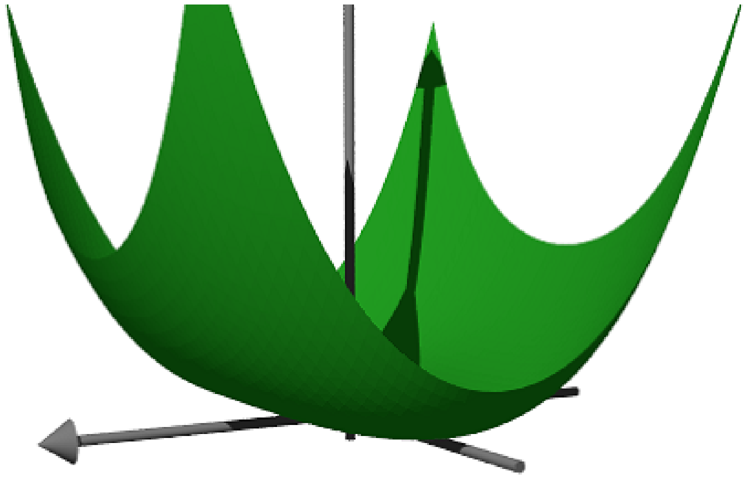
```
>pov3d("x^2+y^3",axiscolor=red,angle=-45°,>anaglyph, ...
>  look=povlook(cyan,0.2),level=-1:0.5:1,zoom=3.8);
```



Terkadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi dengan tangan.

Kami memplot kumpulan titik pada bidang kompleks, di mana hasil kali jarak ke 1 dan -1 sama dengan 1.

```
>pov3d("((x-1)^2+y^2)*((x+1)^2+y^2)/40",r=2, ...
>  angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=10°,n=50, ...
>  <fscale,zoom=3.8);
```

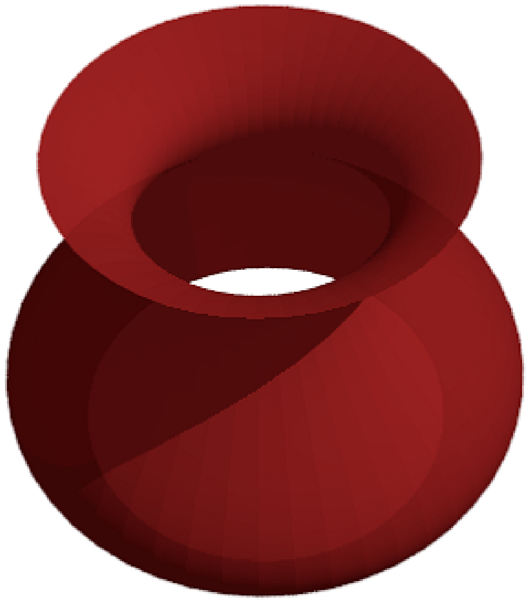


Plot dengan Koordinat

Alih-alih menggunakan fungsi, kita dapat membuat plot dengan koordinat. Seperti pada plot3d, kita membutuhkan tiga matriks untuk mendefinisikan objek.

Pada contoh, kita memutar sebuah fungsi pada sumbu z.

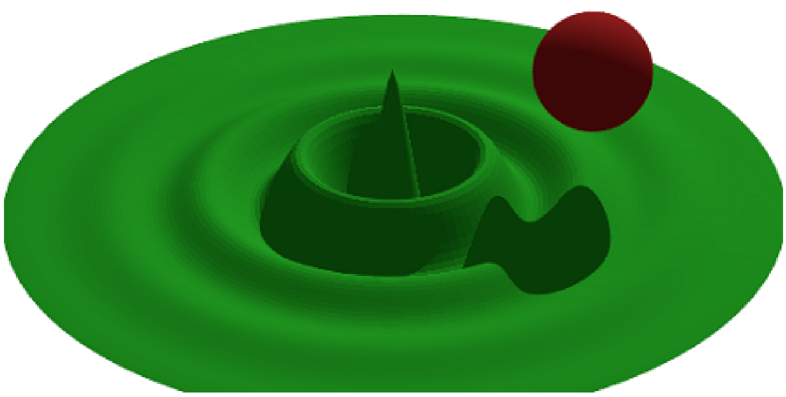
```
>function f(x) := x^3-x+1; ...
>x=-1:0.01:1; t=linspace(0,2pi,50)'; ...
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
>pov3d(X,Y,Z,angle=40°,look=povlook(red,0.1),height=50°,axis=0,zoom=4,light
```



Pada contoh berikut, kita memplot gelombang teredam. Kami menghasilkan gelombang dengan bahasa matriks Euler.

Kami juga menunjukkan, bagaimana objek tambahan dapat ditambahkan ke adegan pov3d. Untuk pembuatan objek, lihat contoh berikut. Perhatikan bahwa plot3d menskalakan plot, sehingga sesuai dengan kubus satuan.

```
>r=linspace(0,1,80); phi=linspace(0,2pi,80)'; ...
>x=r*cos(phi); y=r*sin(phi); z=exp(-5*r)*cos(8*pi*r)/3;  ...
>pov3d(x,y,z,zoom=6,axis=0,height=30°,add=povsphere([0.5,0,0.25],0.15,povlo
>  w=500,h=300);
```



Dengan metode bayangan canggih Povray, hanya sedikit titik yang bisa menghasilkan permukaan yang sangat halus. Hanya pada batas-batas dan bayangan, trik ini bisa terlihat jelas.

Untuk itu, kita perlu menambahkan vektor normal di setiap titik matriks.

```
>Z &= x^2*y^3
```

```
 2  3
x  y
```

Persamaan permukaannya adalah [x,y,Z]. Kami menghitung dua turunan terhadap x dan y dari persamaan ini dan mengambil hasil perkalian silang sebagai normal.

```
>dx &= diff([x,y,Z],x); dy &= diff([x,y,Z],y);
```

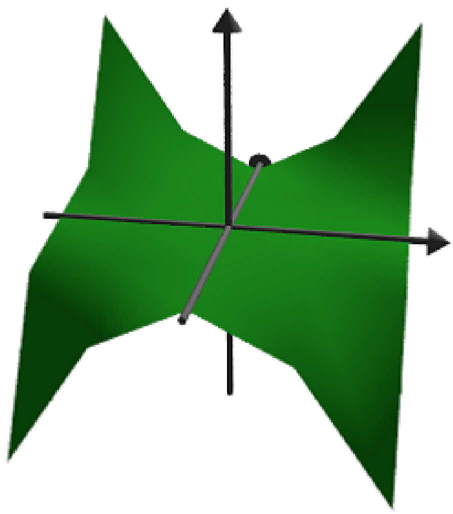
Kami mendefinisikan normal sebagai hasil kali silang dari turunan ini, dan mendefinisikan fungsi koordinat.

```
>N &= crossproduct(dx,dy); NX &= N[1]; NY &= N[2]; NZ &= N[3]; N,
```

```
        3         2  2
[- 2 x y , - 3 x  y , 1]
```

Kami hanya menggunakan 25 poin.

```
>x=-1:0.5:1; y=x';
>pov3d(x,y,Z(x,y),angle=10°, ...
>  xv=NX(x,y),yv=NY(x,y),zv=NZ(x,y),<shadow);
```



Berikut ini adalah simpul Trefoil yang dibuat oleh A. Busser di Povray. Ada versi yang lebih baik dari ini dalam contoh.

See: Examples\Trefoil Knot | Trefoil Knot

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kami menambahkan vektor normal di sini. Kami menggunakan Maxima untuk menghitung normal untuk kami. Pertama, tiga fungsi untuk koordinat sebagai ekspresi simbolis.

```
>X &= ((4+sin(3*y))+cos(x))*cos(2*y); ...
>Y &= ((4+sin(3*y))+cos(x))*sin(2*y); ...
>Z &= sin(x)+2*cos(3*y);
```

Kemudian dua vektor turunan terhadap x dan y.

```
>dx &= diff([X,Y,Z],x); dy &= diff([X,Y,Z],y);
```

Sekarang yang normal, yang merupakan produk silang dari dua turunan.

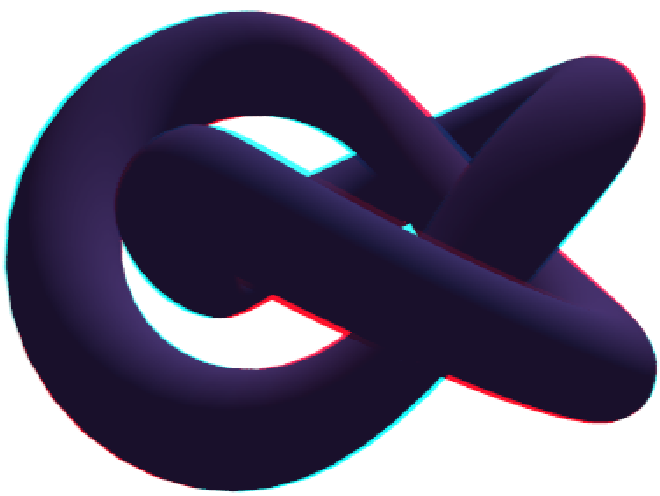
```
>dn &= crossproduct(dx,dy);
```

Kami sekarang mengevaluasi semua ini secara numerik.

```
>x:=linspace(-%pi,%pi,40); y:=linspace(-%pi,%pi,100)';
```

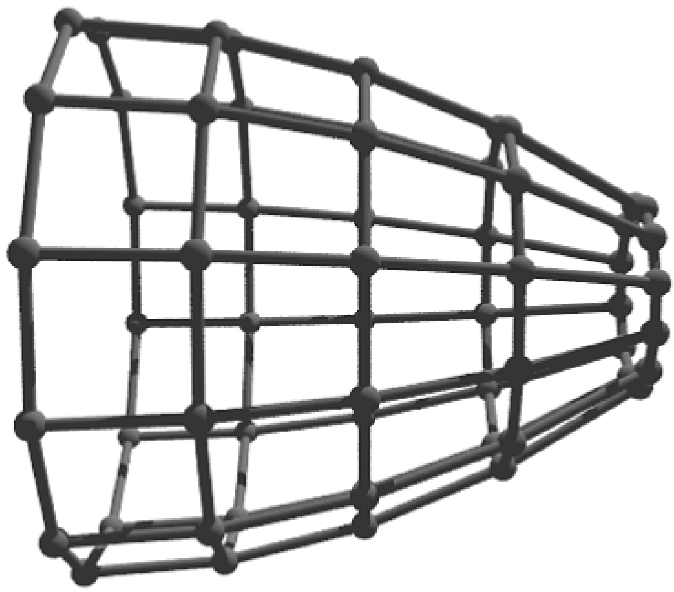
Vektor normal adalah evaluasi dari ekspresi simbolik dn[i] untuk i=1,2,3. Sintaks untuk ini adalah &"expres- sion"(parameter). Ini adalah sebuah alternatif dari metode pada contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik NX, NY, NZ terlebih dahulu.

```
>pov3d(X(x,y),Y(x,y),Z(x,y),>anaglyph,axis=0,zoom=5,w=450,h=350, ...
>  <shadow,look=povlook(blue), ...
>  xv=&"dn[1]"(x,y), yv=&"dn[2]"(x,y), zv=&"dn[3]"(x,y));
```



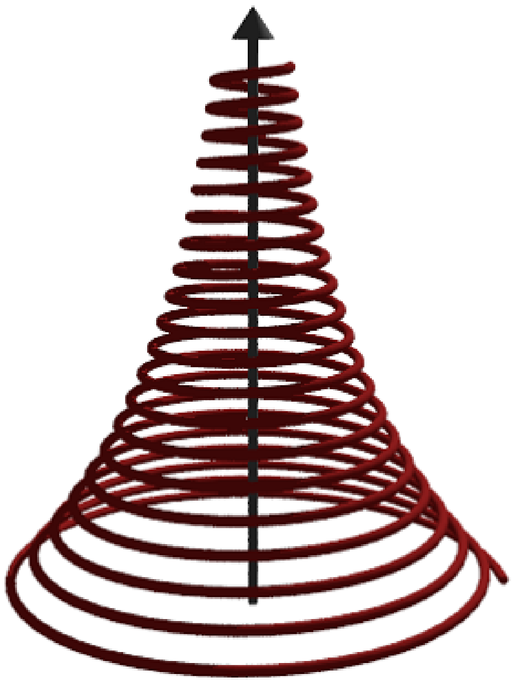
Kami juga dapat menghasilkan kisi-kisi dalam bentuk 3D.

```
>povstart(zoom=4); ...
>x=-1:0.5:1; r=1-(x+1)^2/6; ...
>t=(0°:30°:360°)'; y=r*cos(t); z=r*sin(t); ...
>writeln(povgrid(x,y,z,d=0.02,dballs=0.05)); ...
>povend();
```



Dengan povgrid(), kurva dapat dibuat.

```
>povstart(center=[0,0,1],zoom=3.6); ...
>t=linspace(0,2,1000); r=exp(-t); ...
>x=cos(2*pi*10*t)*r; y=sin(2*pi*10*t)*r; z=t; ...
>writeln(povgrid(x,y,z,povlook(red))); ...
>writeAxis(0,2,axis=3); ...
>povend();
```



Objek Povray

Di atas, kami menggunakan pov3d untuk memplot permukaan. Antarmuka povray di Euler juga dapat menghasilkan objek Povray. Objek-objek ini disimpan sebagai string di Euler, dan perlu ditulis ke file Povray.

Kita memulai output dengan povstart().

```
>povstart(zoom=4);
```

Pertama, kita mendefinisikan tiga silinder, dan menyimpannya dalam string di Euler.

Fungsi povx() dll. hanya mengembalikan vektor [1,0,0], yang dapat digunakan sebagai gantinya.

```
>c1=povcylinder(-povx,povx,1,povlook(red)); ...
>c2=povcylinder(-povy,povy,1,povlook(yellow)); ...
>c3=povcylinder(-povz,povz,1,povlook(blue)); ...
```

String berisi kode Povray, yang tidak perlu kita pahami pada saat itu.

```
>c2
```

```
 cylinder { <0,0,-1>, <0,0,1>, 1
  texture { pigment { color rgb <0.941176,0.941176,0.392157> }  }
  finish { ambient 0.2 }
  }
```

Seperti yang Anda lihat, kami menambahkan tekstur ke objek dalam tiga warna berbeda.

Hal itu dilakukan dengan povlook(), yang mengembalikan sebuah string dengan kode Povray yang relevan. Kita dapat menggunakan warna default Euler, atau menentukan warna kita sendiri. Kita juga dapat menambahkan transparansi, atau mengubah cahaya sekitar.

```
>povlook(rgb(0.1,0.2,0.3),0.1,0.5)
```

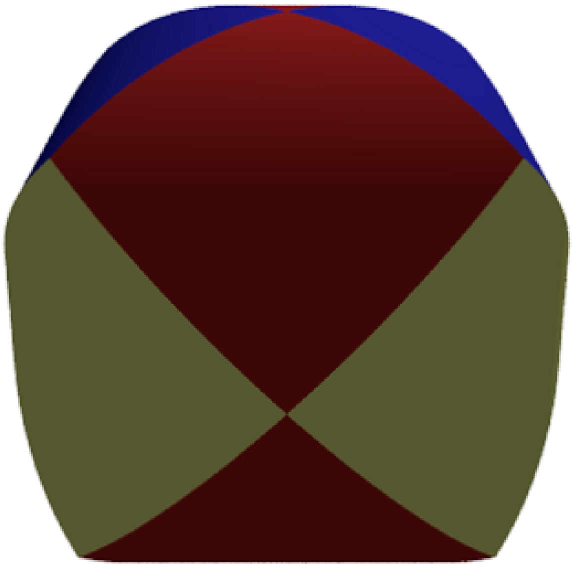
```
  texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> }  }
  finish { ambient 0.5 }
```

Sekarang kita mendefinisikan objek perpotongan, dan menulis hasilnya ke file.

```
>writeln(povintersection([c1,c2,c3]));
```

Perpotongan tiga silinder sulit dibayangkan, jika Anda belum pernah melihatnya.

```
>povend;
```



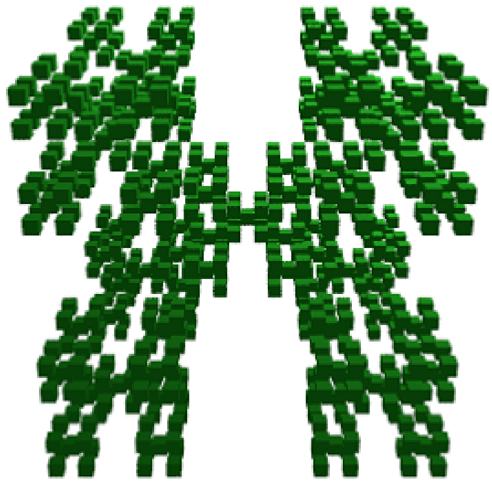
Fungsi-fungsi berikut ini menghasilkan fraktal secara rekursif.

Fungsi pertama menunjukkan, bagaimana Euler menangani objek Povray sederhana. Fungsi povbox() mengembalikan sebuah string, yang berisi koordinat kotak, tekstur dan hasil akhir.

```
>function onebox(x,y,z,d) := povbox([x,y,z],[x+d,y+d,z+d],povlook());
>function fractal (x,y,z,h,n) ...
```

```
  if n==1 then writeln(onebox(x,y,z,h));
  else
    h=h/3;
    fractal(x,y,z,h,n-1);
    fractal(x+2*h,y,z,h,n-1);
    fractal(x,y+2*h,z,h,n-1);
    fractal(x,y,z+2*h,h,n-1);
    fractal(x+2*h,y+2*h,z,h,n-1);
    fractal(x+2*h,y,z+2*h,h,n-1);
    fractal(x,y+2*h,z+2*h,h,n-1);
    fractal(x+2*h,y+2*h,z+2*h,h,n-1);
    fractal(x+h,y+h,z+h,h,n-1);
  endif;
endfunction
```

```
>povstart(fade=10,<shadow);
>fractal(-1,-1,-1,2,4);
>povend();
```



Perbedaan memungkinkan pemotongan satu objek dari objek lainnya. Seperti persimpangan, ada bagian dari objek CSG Povray.

```
>povstart(light=[5,-5,5],fade=10);
```

Untuk demonstrasi ini, kita mendefinisikan sebuah objek di Povray, alih-alih menggunakan string di Euler. Definisi akan langsung dituliskan ke file.

Koordinat kotak -1 berarti [-1,-1,-1].

```
>povdefine("mycube",povbox(-1,1));
```

Kita dapat menggunakan objek ini di povobject(), yang mengembalikan sebuah string seperti biasa.

```
>c1=povobject("mycube",povlook(red));
```

Kami menghasilkan kubus kedua, dan memutar serta menskalakannya sedikit.

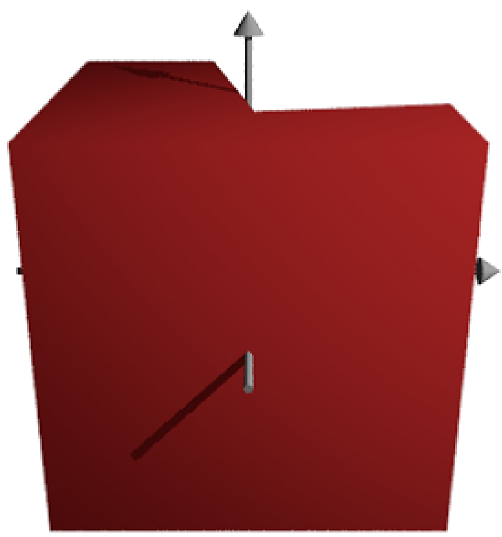
```
>c2=povobject("mycube",povlook(yellow),translate=[1,1,1], ...
>  rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Kemudian kita ambil selisih dari kedua objek tersebut.

```
>writeln(povdifference(c1,c2));
```

Sekarang tambahkan tiga sumbu.

```
>writeAxis(-1.2,1.2,axis=1); ...
>writeAxis(-1.2,1.2,axis=2); ...
>writeAxis(-1.2,1.2,axis=4); ...
>povend();
```



Fungsi Implisit

Povray dapat memplot himpunan di mana f(x,y,z)=0, seperti parameter implisit pada plot3d. Namun, hasilnya terlihat jauh lebih baik.

Sintaks untuk fungsi-fungsi tersebut sedikit berbeda. Anda tidak dapat menggunakan output dari ekspresi Maxima atau Euler.

$$((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2)=d$$

```
>povstart(angle=70°,height=50°,zoom=4);
>c=0.1; d=0.1; ...
>writeln(povsurface("(pow(pow(x,2)+pow(y,2)-pow(c,2),2)+pow(pow(z,2)-1,2))*
>povend();
```

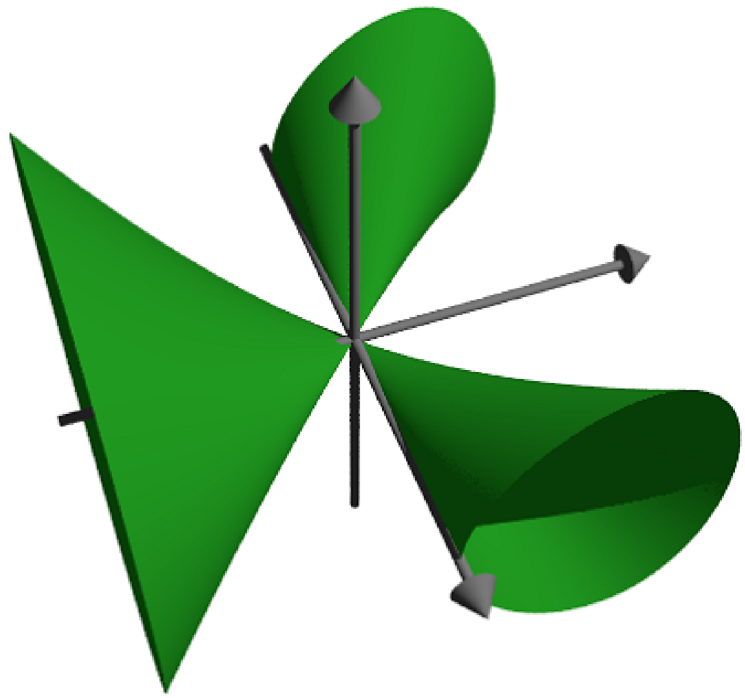
```
Error : Povray error!

Error generated by error() command

povray:
    error("Povray error!");
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

```
>povstart(angle=25°,height=10°);
>writeln(povsurface("pow(x,2)+pow(y,2)*pow(z,2)-1",povlook(blue),povbox(-2,
>povend();
```

```
>povstart(angle=70°,height=50°,zoom=4);
```

Membuat permukaan implisit. Perhatikan sintaks yang berbeda dalam ekspresi.

```
>writeln(povsurface("pow(x,2)*y-pow(y,3)-pow(z,2)",povlook(green))); ...
>writeAxes(); ...
>povend();
```



Objek Jaring

Dalam contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambarnya dengan informasi tambahan.

Kami ingin memaksimalkan xy di bawah kondisi x+y = 1 dan mendemonstrasikan sentuhan tangensial dari garis level.

```
>povstart(angle=-10°,center=[0.5,0.5,0.5],zoom=7);
```

Kita tidak dapat menyimpan objek dalam sebuah string seperti sebelumnya, karena ukurannya terlalu besar. Jadi kita mendefinisikan objek dalam file Povray menggunakan declare. Fungsi povtriangle() melakukan hal ini secara otomatis. Fungsi ini dapat menerima vektor normal seperti halnya pov3d().

Berikut ini mendefinisikan objek mesh, dan langsung menuliskannya ke dalam file.

```
>x=0:0.02:1; y=x'; z=x*y; vx=-y; vy=-x; vz=1;
>mesh=povtriangles(x,y,z,"",vx,vy,vz);
```

Sekarang kita tentukan dua cakram, yang akan berpotongan dengan permukaan.

```
>cl=povdisc([0.5,0.5,0],[1,1,0],2); ...
>ll=povdisc([0,0,1/4],[0,0,1],2);
```

Tuliskan permukaan dikurangi kedua cakram.

```
>writeln(povdifference(mesh,povunion([cl,ll]),povlook(green)));
```

Tuliskan kedua perpotongan tersebut.

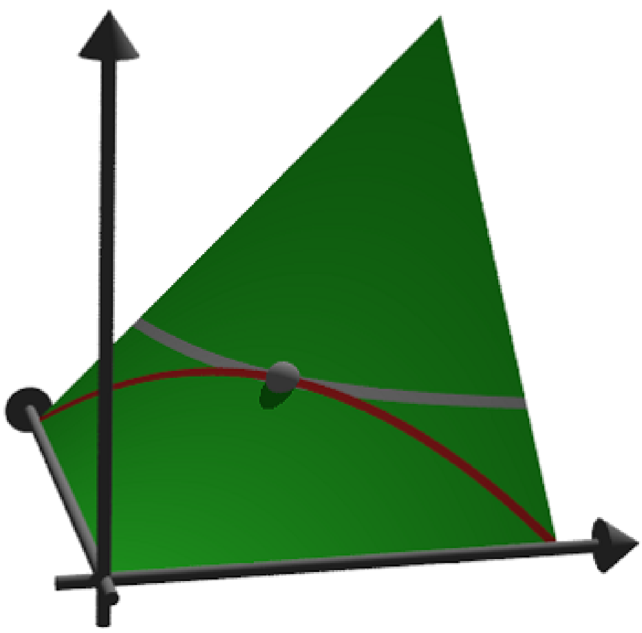
```
>writeln(povintersection([mesh,cl],povlook(red))); ...
>writeln(povintersection([mesh,ll],povlook(gray)));
```

Tulislah satu titik secara maksimal.

```
>writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2*defaultpointsize));
```

Tambahkan sumbu dan selesaikan.

```
>writeAxes(0,1,0,1,0,1,d=0.015); ...
>povend();
```

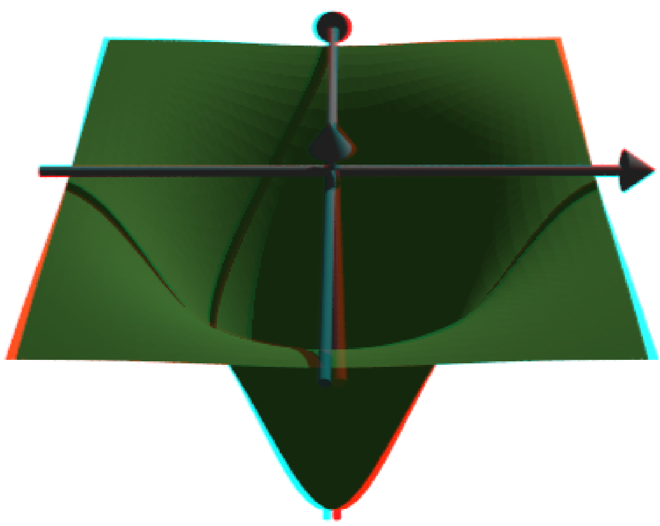


Untuk menghasilkan anaglyph untuk kacamata merah/cyan, Povray harus dijalankan dua kali dari posisi kamera yang berbeda. Ini menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi loadanaglyph().

Tentu saja, Anda memerlukan kacamata merah/cyan untuk melihat contoh berikut ini dengan benar.

Fungsi pov3d() memiliki saklar sederhana untuk menghasilkan anaglyph.

```
>pov3d("-exp(-x^2-y^2)/2",r=2,height=45°,>anaglyph, ...
>   center=[0,0,0.5],zoom=3.5);
```



Jika Anda membuat scene dengan objek, Anda harus menempatkan pembuatan scene ke dalam fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter anaglyph.

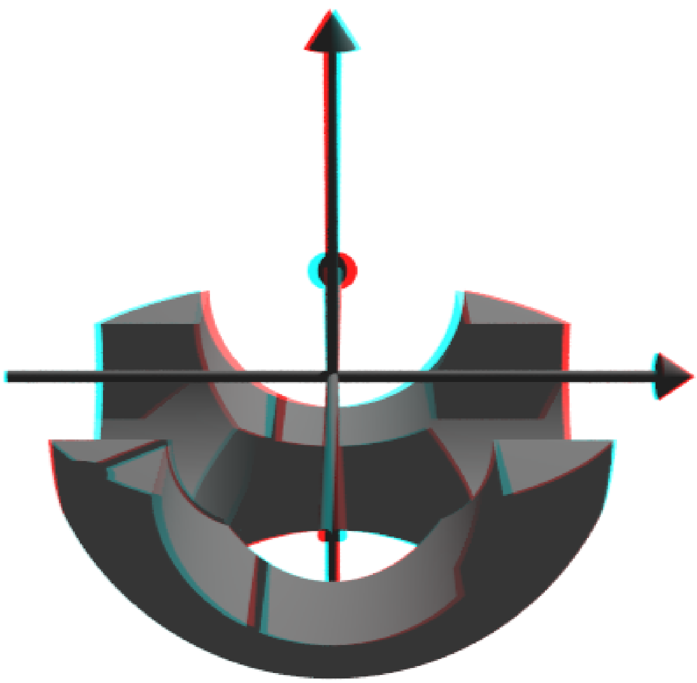
```
>function myscene ...
```

```
    s=povsphere(povc,1);
    cl=povcylinder(-povz,povz,0.5);
    clx=povobject(cl,rotate=xrotate(90°));
```

```
    cly=povobject(cl,rotate=yrotate(90°));
    c=povbox([-1,-1,0],1);
    un=povunion([cl,clx,cly,c]);
    obj=povdifference(s,un,povlook(red));
    writeln(obj);
    writeAxes();
  endfunction
```

Fungsi povanaglyph() melakukan semua ini. Parameter-parameternya seperti pada povstart() dan povend() yang digabungkan.

```
>povanaglyph("myscene",zoom=4.5);
```



Mendeskripsikan Objek Sendiri

Antarmuka povray Euler berisi banyak sekali objek. Tetapi Anda tidak dibatasi pada objek-objek tersebut. Anda dapat membuat objek sendiri, yang menggabungkan objek lain, atau objek yang benar-benar baru.

Kami mendemonstrasikan sebuah torus. Perintah Povray untuk ini adalah "torus". Jadi kita mengembalikan sebuah string dengan perintah ini dan parameternya. Perhatikan bahwa torus selalu berpusat di titik asal.

```
>function povdonat (r1,r2,look="") ...
```

```
    return "torus {"+r1+","+r2+look+"}";
  endfunction
```

Inilah torus pertama kami.

```
>t1=povdonat(0.8,0.2)
```

```
  torus {0.8,0.2}
```

Mari kita gunakan objek ini untuk membuat torus kedua, diterjemahkan dan diputar.

```
>t2=povobject(t1,rotate=xrotate(90°),translate=[0.8,0,0])
```

```
  object { torus {0.8,0.2}
   rotate 90 *x
   translate <0.8,0,0>
   }
```

Sekarang, kita tempatkan semua benda ini ke dalam suatu pemandangan. Untuk tampilannya, kami menggunakan Phong Shading.

```
>povstart(center=[0.4,0,0],angle=0°,zoom=3.8,aspect=1.5); ...
>writeln(povobject(t1,povlook(green,phong=1))); ...
>writeln(povobject(t2,povlook(green,phong=1))); ...
```

```
>povend();
```

memanggil program Povray. Namun, jika terjadi kesalahan, program ini tidak menampilkan kesalahan. Oleh karena itu, Anda harus menggunakan

```
>povend(<exit);
```

jika ada yang tidak berhasil. Ini akan membiarkan jendela Povray terbuka.

```
>povend(h=320,w=480);
```

Berikut adalah contoh yang lebih rumit. Kami menyelesaikan

$$Ax \leq b, \quad x \geq 0, \quad c.x \to \text{Max.}$$

dan menunjukkan titik-titik yang layak dan optimal dalam plot 3D.

```
>A=[10,8,4;5,6,8;6,3,2;9,5,6];
>b=[10,10,10,10]';
>c=[1,1,1];
```

Pertama, mari kita periksa, apakah contoh ini memiliki solusi atau tidak.

```
>x=simplex(A,b,c,>max,>check)'
```

```
[0,   1,   0.5]
```

Ya, benar.
Selanjutnya kita mendefinisikan dua objek. Yang pertama adalah pesawat

$$a \cdot x \leq b$$

```
>function oneplane (a,b,look="") ...
```

```
    return povplane(a,b,look)
  endfunction
```

Kemudian kita mendefinisikan perpotongan semua setengah ruang dan kubus.

```
>function adm (A, b, r, look="") ...
```

```
    ol=[];
    loop 1 to rows(A); ol=ol|oneplane(A[#],b[#]); end;
    ol=ol|povbox([0,0,0],[r,r,r]);
    return povintersection(ol,look);
  endfunction
```

Sekarang, kita bisa merencanakan adegan tersebut.

```
>povstart(angle=120°,center=[0.5,0.5,0.5],zoom=3.5); ...
>writeln(adm(A,b,2,povlook(green,0.4))); ...
>writeAxes(0,1.3,0,1.6,0,1.5); ...
```

Berikut ini adalah lingkaran di sekeliling optimal.

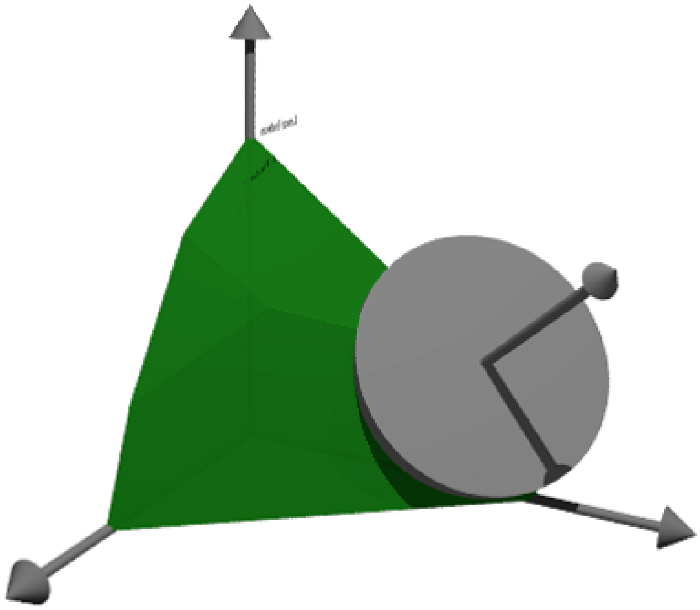
```
>writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ...
>  povlook(red,0.9)));
```

Dan kesalahan pada arah yang optimal.

```
>writeln(povarrow(x,c*0.5,povlook(red)));
```

Kami menambahkan teks ke layar. Teks hanyalah sebuah objek 3D. Kita perlu menempatkan dan memutarnya sesuai dengan pandangan kita.

```
>writeln(povtext("Linear Problem",[0,0.2,1.3],size=0.05,rotate=5°)); ...
>povend();
```



Contoh Lain

Anda dapat menemukan beberapa contoh lain untuk Povray di Euler dalam file berikut.

See: Examples/Dandelin Spheres
See: Examples/Donat Math
See: Examples/Trefoil Knot
See: Examples/Optimization by Affine Scaling

BAB 5

MENGGUNAKAN EMT UNTUK KALKULUS

Kalkulus dengan EMT

Materi Kalkulus mencakup di antaranya: - Fungsi (fungsi aljabar, trigonometri, eksponensial, logaritma, komposisi fungsi)
- Limit Fungsi,
- Turunan Fungsi,
- Integral Tak Tentu,
- Integral Tentu dan Aplikasinya,
- Barisan dan Deret (kekonvergenan barisan dan deret).

EMT (bersama Maxima) dapat digunakan untuk melakukan semua perhitungan di dalam kalkulus, baik secara numerik maupun analitik (eksak).

Mendefinisikan Fungsi

Terdapat beberapa cara mendefinisikan fungsi pada EMT, yakni: - Menggunakan format nama_fungsi := rumus fungsi (untuk fungsi numerik),
- Menggunakan format nama_fungsi &= rumus fungsi (untuk fungsi simbolik, namun dapat dihitung secara numerik),
- Menggunakan format nama_fungsi &&= rumus fungsi (untuk fungsi simbolik murni, tidak dapat dihitung langsung),
- Fungsi sebagai program EMT.

Setiap format harus diawali dengan perintah function (bukan sebagai ekspresi). Berikut adalah adalah beberapa contoh cara mendefinisikan fungsi:

$$f(x) = 2x^2 + e^{\sin(x)}.$$

```
>function f(x) := 2*x^2+exp(sin(x)) // fungsi numerik
>f(0), f(1), f(pi)
```

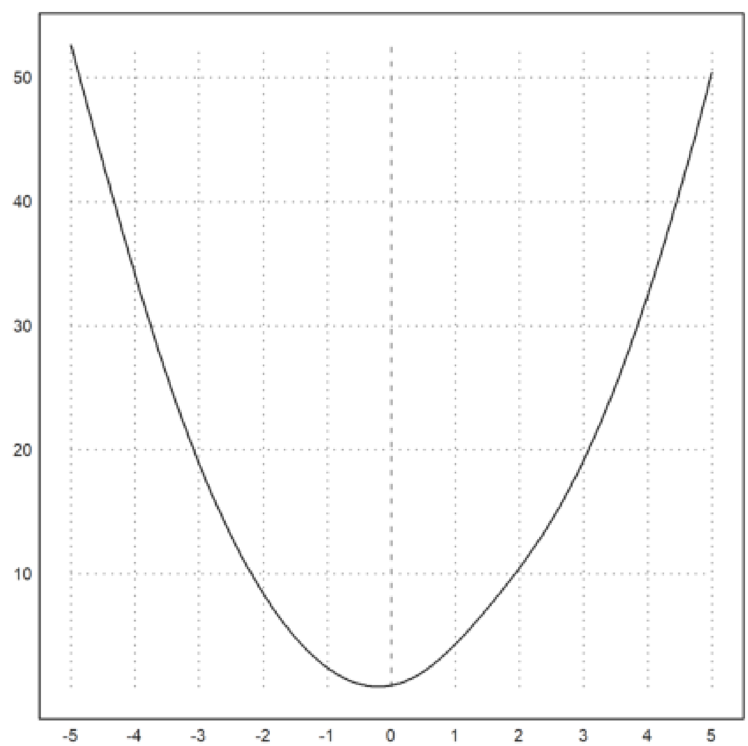
```
1
4.31977682472
20.7392088022
```

```
>f(a) // tidak dapat dihitung nilainya
```

```
Variable or function a not found.
Error in:
f(a) // tidak dapat dihitung nilainya ...
   ^
```

Silakan Anda plot kurva fungsi di atas!

```
>plot2d("f(x)",-5,5):
```



Berikutnya kita definisikan fungsi:

$$g(x) = \frac{\sqrt{x^2 - 3x}}{x + 1}.$$

```
>function g(x) := sqrt(x^2-3*x)/(x+1)
>g(3)
```

```
0
```

```
>g(0)
```

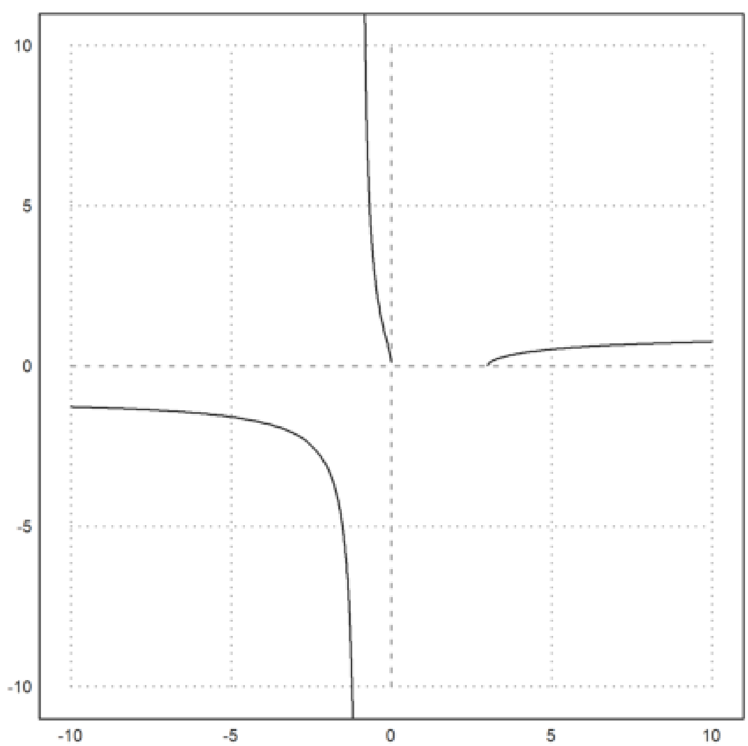
```
0
```

```
>g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik
```

```
Floating point error!
Error in sqrt
Try "trace errors" to inspect local variables after errors.
g:
    useglobal; return sqrt(x^2-3*x)/(x+1)
Error in:
g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik ...
    ^
```

Silakan Anda plot kurva fungsi di atas!

```
>plot2d("g(x)",-10,10,-10,10):
```



```
>f(g(5)) // komposisi fungsi
```

```
2.20920171961
```

```
>g(f(5))
```

```
0.950898070639
```

```
>function h(x) := f(g(x)) // definisi komposisi fungsi
>h(5) // sama dengan f(g(5))
```

```
2.20920171961
```

Silakan Anda plot kurva fungsi komposisi fungsi f dan g:

$$h(x) = f(g(x))$$

dan

$$u(x) = g(f(x))$$

bersama-sama kurva fungsi f dan g dalam satu bidang koordinat.

```
>function f(x) &= 2*x^2+exp(sin(x))
```

```
 sin(x)      2
E       + 2 x
```

```
>function g(x) &= sqrt(x^2-3*x)/(x+1)
```

```
      2
sqrt(x  - 3 x)
--------------
    x + 1
```

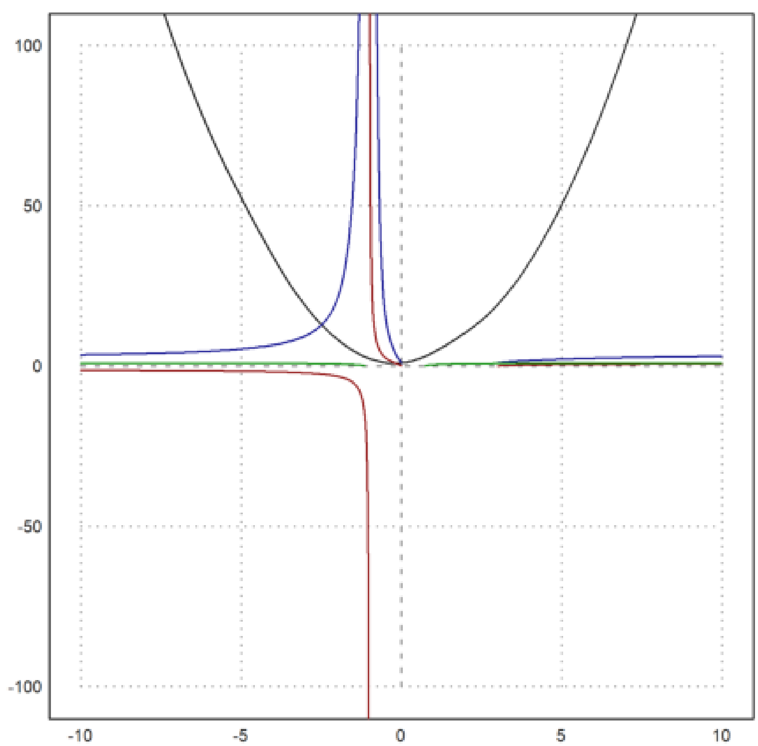
```
>function h(x) &= f(g(x))
```

```
                   2
             sqrt(x   - 3 x)
         sin(--------------)         2
                 x + 1          2 (x   - 3 x)
        E                     + ------------
                                         2
                                  (x + 1)
```

```
>function u(x) &= g(f(x))
```

```
           sin(x)       2 2         sin(x)       2
     sqrt((E        + 2 x )  - 3 (E        + 2 x ))
     ----------------------------------------------
                     sin(x)       2
                    E       + 2 x  + 1
```

```
>aspect(1); plot2d("f(x)",-10,10,-100,100); plot2d("g(x)",color=red,>add);
```



```
>f(0:10) // nilai-nilai f(0), f(1), f(2), ..., f(10)
```

```
[1,  4.31978,  10.4826,  19.1516,  32.4692,  50.3833,  72.7562,
99.929,  130.69,  163.51,  200.58]
```

```
>fmap(0:10) // sama dengan f(0:10), berlaku untuk semua fungsi
```

```
[1,  4.31978,  10.4826,  19.1516,  32.4692,  50.3833,  72.7562,
99.929,  130.69,  163.51,  200.58]
```

```
>gmap(200:210)
```

```
[0.987534,  0.987596,  0.987657,  0.987718,  0.987778,  0.987837,
0.987896,  0.987954,  0.988012,  0.988069,  0.988126]
```

Misalkan kita akan mendefinisikan fungsi

$$f(x) = \begin{cases} x^3 & x > 0 \\ x^2 & x \leq 0. \end{cases}$$

Fungsi tersebut tidak dapat didefinisikan sebagai fungsi numerik secara "inline" menggunakan format :=, melainkan didefinisikan sebagai program. Perhatikan, kata "map" digunakan agar fungsi dapat menerima vektor sebagai input, dan hasilnya berupa vektor. Jika tanpa kata "map" fungsinya hanya dapat menerima input satu nilai.

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

```
  if x>0 then return x^3
  else return x^2
  endif;
endfunction
```

```
>f(1)
```

```
1
```

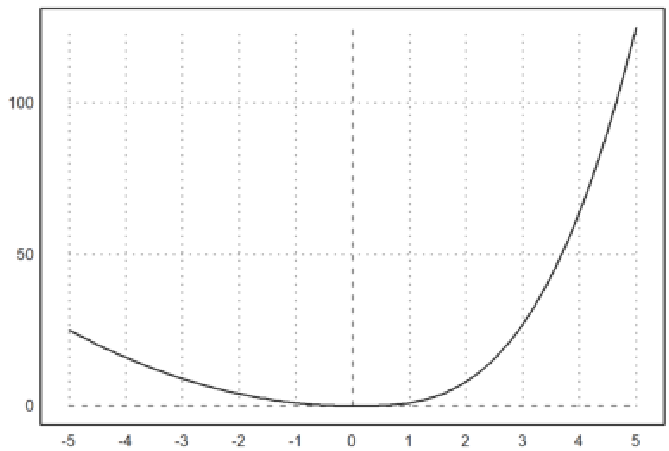
```
>f(-2)
```

```
4
```

```
>f(-5:5)
```

```
[25,  16,  9,  4,  1,  0,  1,  8,  27,  64,  125]
```

```
>aspect(1.5); plot2d("f(x)",-5,5):
```




```
>function f(x) &= 2*E^x // fungsi simbolik
```

```
                             x
                          2 E
```

```
>$f(a) // nilai fungsi secara simbolik
```

$$2\,e^{a}$$

```
>f(E) // nilai fungsi berupa bilangan desimal
```

```
30.308524483
```

```
>$f(E), $float(%)
```

30.30852448295852

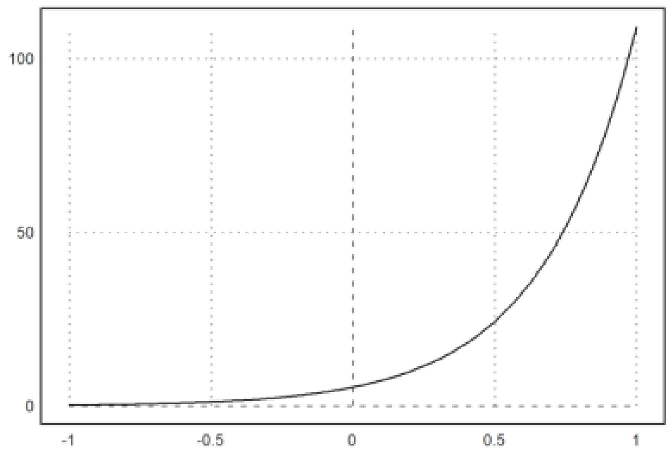
```
>function g(x) &= 3*x+1
```

```
                          3 x + 1
```

```
>function h(x) &= f(g(x)) // komposisi fungsi
```

```
                            3 x + 1
                         2 E
```

```
>plot2d("h(x)",-1,1):
```



Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan fungsi-fungsi tersebut dan komposisinya di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung beberapa nilainya, baik untuk satu nilai maupun vektor. Gambar grafik fungsi-fungsi tersebut dan komposisi-komposisi 2 fungsi.

Juga, carilah fungsi beberapa (dua) variabel. Lakukan hal sama seperti di atas.

Soal 1

$$m(x) = \sqrt{x^2 - 1}$$

$$n(x) = \frac{2}{x}$$

```
>function m(x) &= (x^2-1)^(1/2)
```

```
                             2
                        sqrt(x  - 1)
```

```
>function n(x) &= 2/x
```

```
                            2
                            -
                            x
```

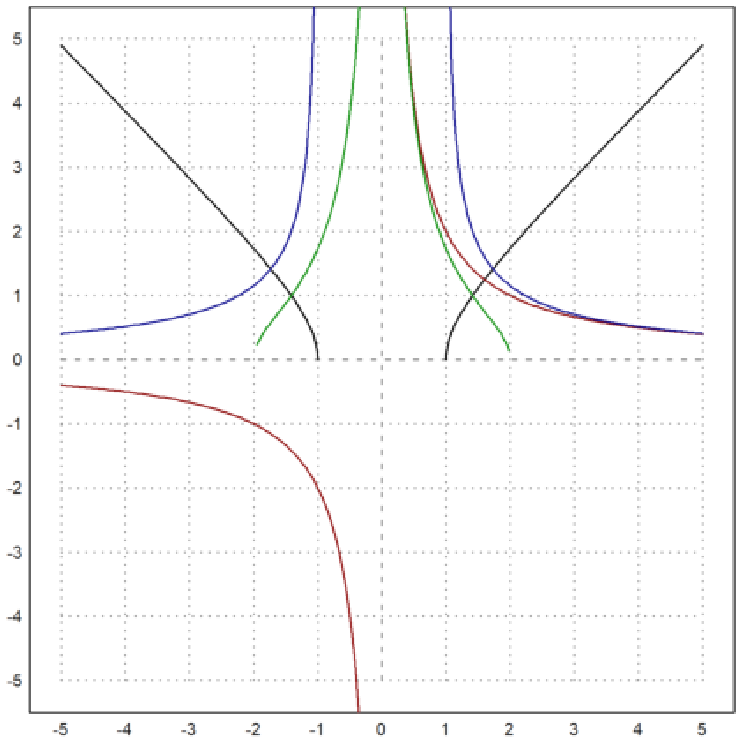
```
>function p(x) &= m(n(x))
```

```
                            4
                        sqrt(-- - 1)
                             2
                            x
```

```
>function q(x) &= n(m(x))
```

```
      2
------------
         2
sqrt(x   - 1)
```

```
>plot2d(["m(x)","n(x)","p(x)","q(x)"],color=[1:4],a=-5,b=5,c=-5,d=5):
```



Soal 2

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

```
>function f(x) := x^3-2*x
>f(1)
```

```
-1
```

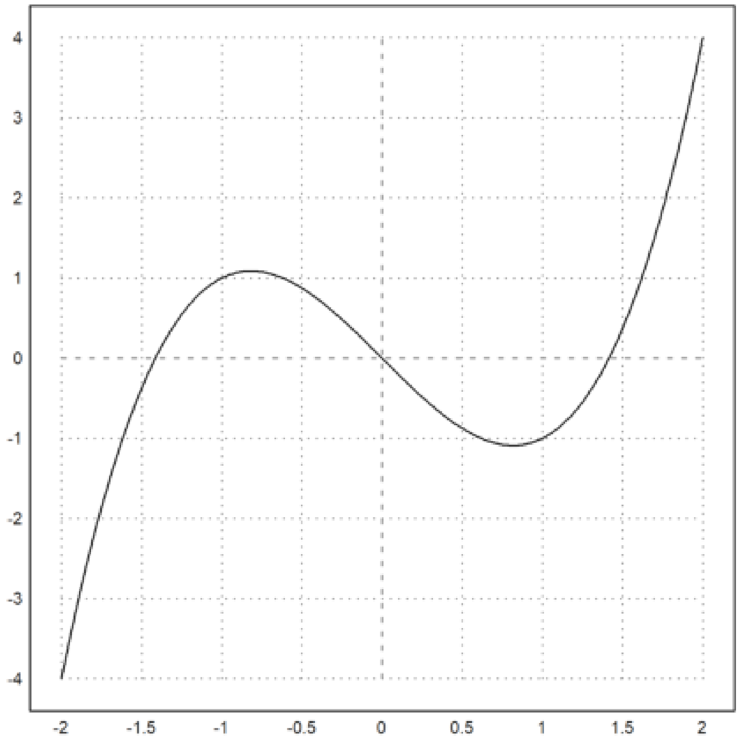
```
>f(5)
```

```
115
```

```
>f(-5:5)
```

```
[-115,   -56,   -21,   -4,   1,   0,   -1,   4,   21,   56,   115]
```

```
>aspect(1); plot2d("f(x)"):
```




```
>function g(x) := -x^3+5*x
>function h(x) := f(g(x))
>h(2)
```

```
4
```

Soal 3

$$k(x) = 3sin(x)$$

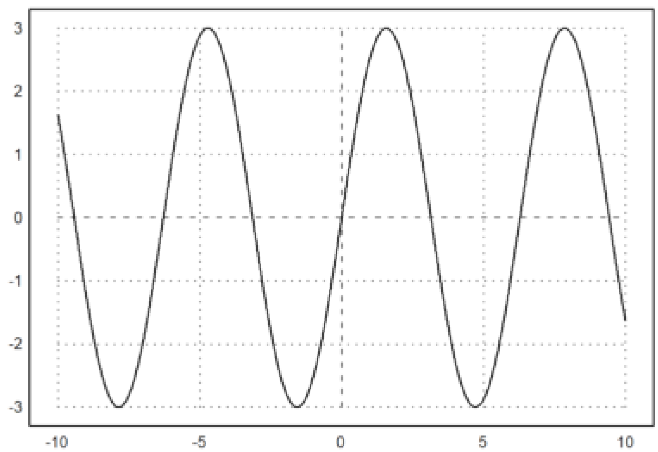
```
>function k(x) := 3*sin(x)
>k(pi), k(pi/2), k(pi/3)
```

```
0
3
2.59807621135
```

```
>k(pi/2:2pi)
```

```
[3,  1.62091,  -1.24844,  -2.96998,  -1.96093]
```

```
>aspect(1.5); plot2d("k(x)",-10,10):
```



Soal 4

$$e^2 + 3x$$

```
>function l(x) &= E^2+3*x
```

```
       2
3 x + E
```

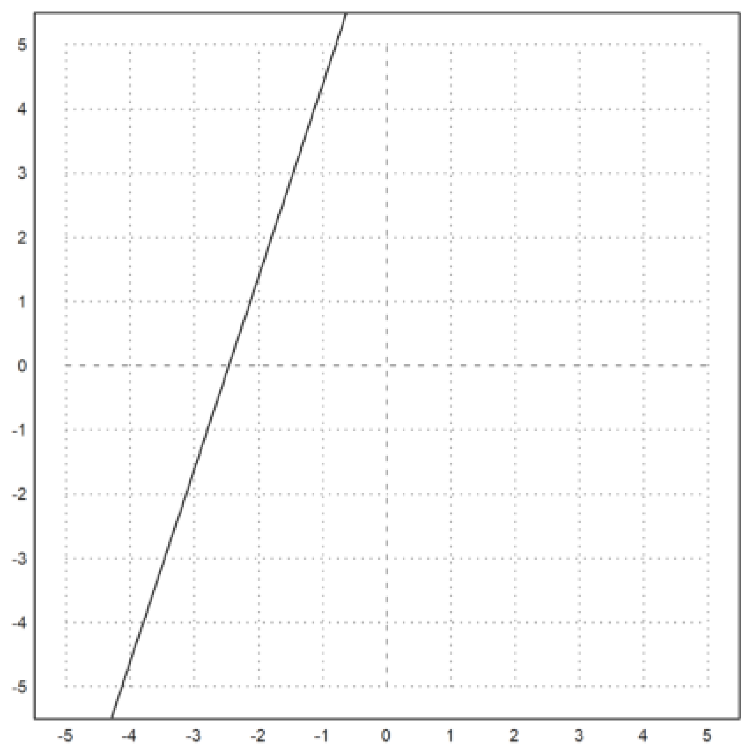
```
>$l(5), $l(9)
```

$$e^2 + 27$$

```
>$l(10), $float(%)
```

$$37.38905609893065$$

```
>plot2d("l(x)", -5,5,-5,5):
```



Soal 5

$$r(x) = \begin{cases} -x^2 + 4 & x \leq 1 \\ 3x & x > 1. \end{cases}$$

```
>function map r(x)
```

```
    if x<=1 then return -x^2+4
    else x>1 return 3*x
    endif;
  endfunction
```

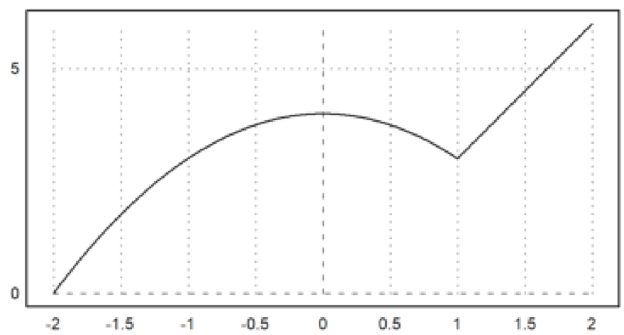
```
>r(-2), r(4)
```

```
0
12
```

```
>r(-10:10)
```

```
[-96,  -77,  -60,  -45,  -32,  -21,  -12,  -5,  0,  3,  4,  3,  6,  9,
12,  15,  18,  21,  24,  27,  30]
```

```
>aspect(2); plot2d("r(x)"):
```



Menghitung Limit

Perhitungan limit pada EMT dapat dilakukan dengan menggunakan fungsi Maxima, yakni "limit". Fungsi "limit" dapat digunakan untuk menghitung limit fungsi dalam bentuk ekspresi maupun fungsi yang sudah didefinisikan sebelumnya. Nilai limit dapat dihitung pada sebarang nilai atau pada tak hingga (-inf, minf, dan inf). Limit kiri dan limit kanan juga dapat dihitung, dengan cara memberi opsi "plus" atau "minus". Hasil limit dapat berupa nilai, "und" (tak definisi), "ind" (tak tentu namun terbatas), "infinity" (kompleks tak hingga).

Perhatikan beberapa contoh berikut. Perhatikan cara menampilkan perhitungan secara lengkap, tidak hanya menampilkan hasilnya saja.

```
>$showev('limit(sqrt(x^2-3*x)/(x+1),x,inf))
```

$$\lim_{x\rightarrow \infty }{\frac{\sqrt{x^2-3\,x}}{x+1}}=1$$

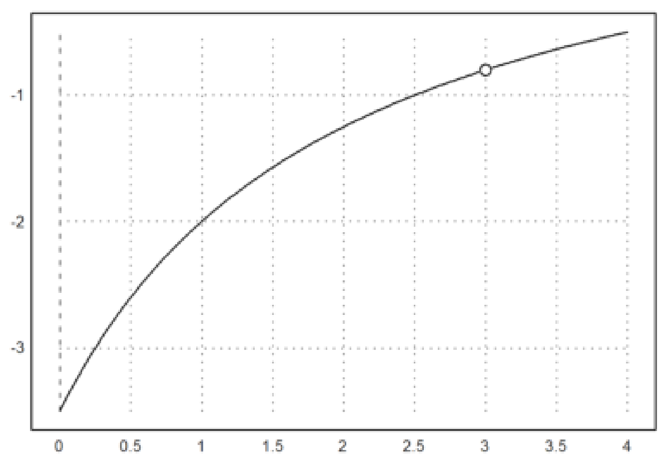
```
>$limit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3)
```

$$-\frac{4}{5}$$

$$\lim_{x\rightarrow 3}{\frac{x^3-13\,x^2+51\,x-63}{x^3-4\,x^2-3\,x+18}}=-\frac{4}{5}$$

Fungsi tersebut diskontinu di titik x=3. Berikut adalah grafik fungsinya.

```
>aspect(1.5); plot2d("(x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18)",0,4); plot2d
```



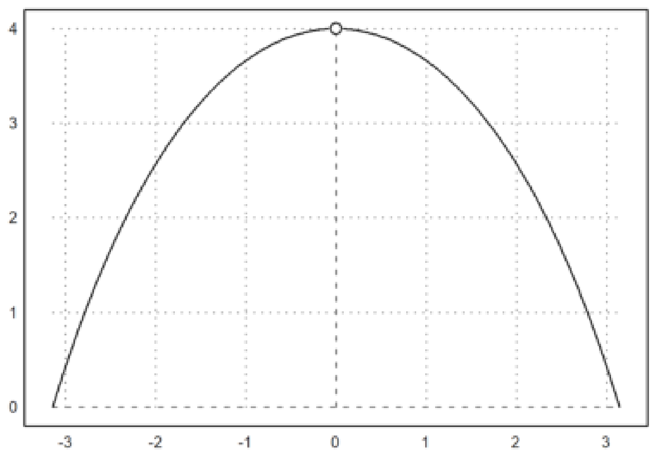
```
>$limit(2*x*sin(x)/(1-cos(x)),x,0)
```

$$4$$

$$2\left(\lim_{x\to 0}{\frac{x\,\sin x}{1-\cos x}}\right)=4$$

Fungsi tersebut diskontinu di titik x=0. Berikut adalah grafik fungsinya.

```
>plot2d("2*x*sin(x)/(1-cos(x))",-pi,pi); plot2d(0,4,>points,style="ow",>add
```



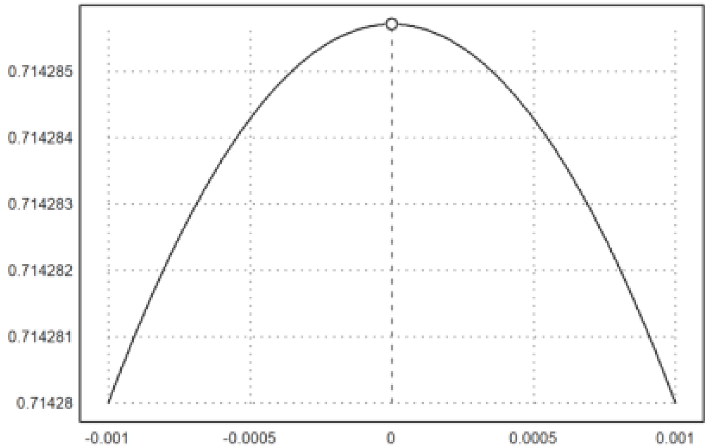

```
>$limit(cot(7*h)/cot(5*h),h,0)
```

$$\frac{5}{7}$$

$$\lim_{h\to 0}{\frac{\cot \left(7\,h\right)}{\cot \left(5\,h\right)}}=\frac{5}{7}$$

Fungsi tersebut juga diskontinu (karena tidak terdefinisi) di x=0. Berikut adalah grafiknya.

```
>plot2d("cot(7*x)/cot(5*x)",-0.001,0.001); plot2d(0,5/7,>points,style="ow",
```

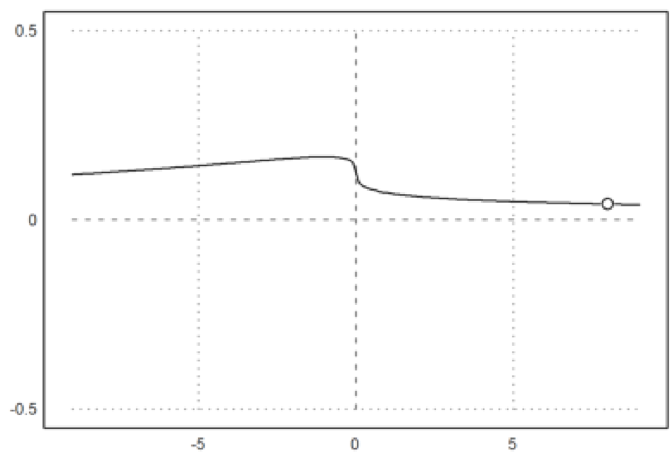



```
>$showev('limit(((x/8)^(1/3)-1)/(x-8),x,8))
```

$$\lim_{x\rightarrow 8}{\frac{\frac{x^{\frac{1}{3}}}{2}-1}{x-8}}=\frac{1}{24}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>aspect(1.5); plot2d("((x/8)^(1/3)-1)/(x-8)",-9,9,-0.5,0.5); plot2d(8,1/24,
```

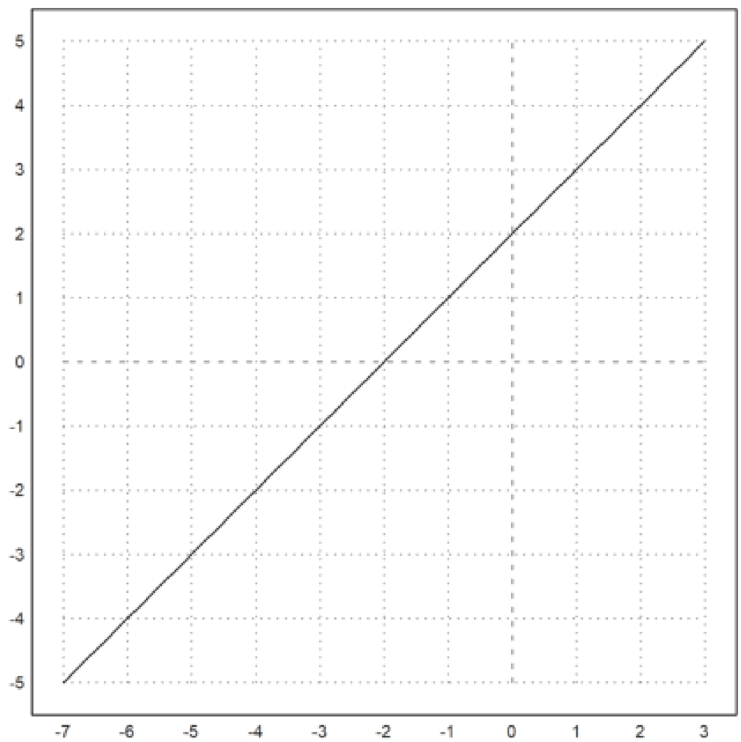



```
>$showev('limit(1/(2*x-1),x,0))
```

$$\lim_{x\rightarrow 0}{\frac{1}{2\,x-1}}=-1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>aspect(1); plot2d("(x^2-3*x-10)/(x-5)",-7,3,-5,5):
```

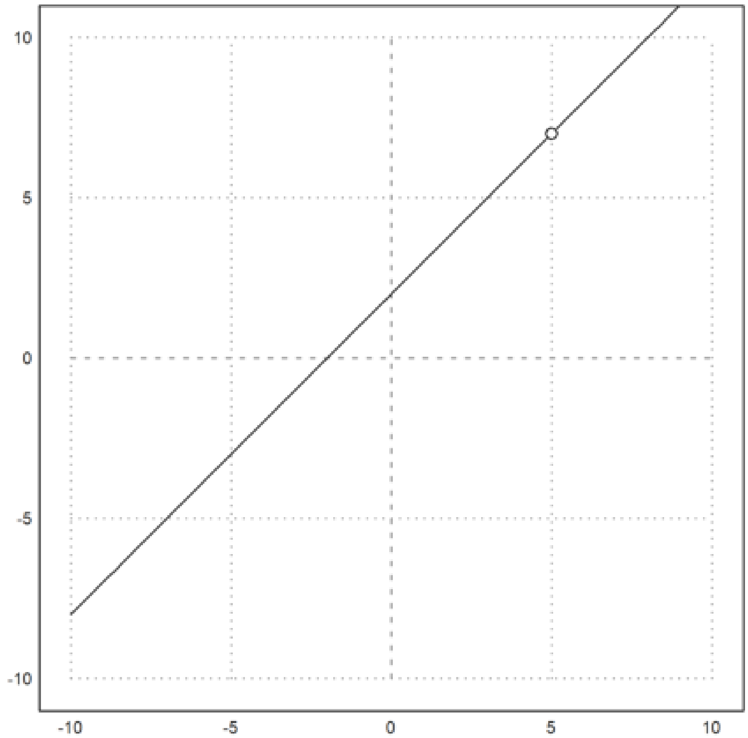



```
>$showev('limit((x^2-3*x-10)/(x-5),x,5))
```

$$\lim_{x\rightarrow 5}{\frac{x^2-3\,x-10}{x-5}}=7$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(x^2-3*x-10)/(x-5)",-10,10,-10,10); plot2d(5,7,>points,style="ow",
```

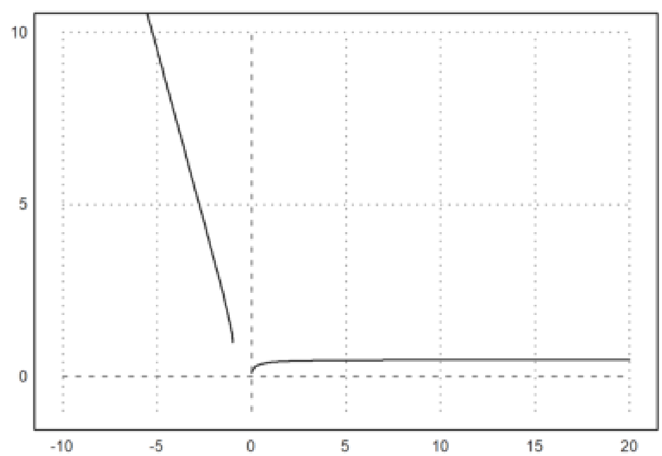



```
>$showev('limit(sqrt(x^2+x)-x,x,inf))
```

$$\lim_{x\to\infty} \sqrt{x^2+x} - x = \frac{1}{2}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>aspect(1.5); plot2d("(sqrt(x^2+x))-x",-10,20,-1,10):
```



```
>$showev('limit(abs(x-1)/(x-1),x,1,minus))
```

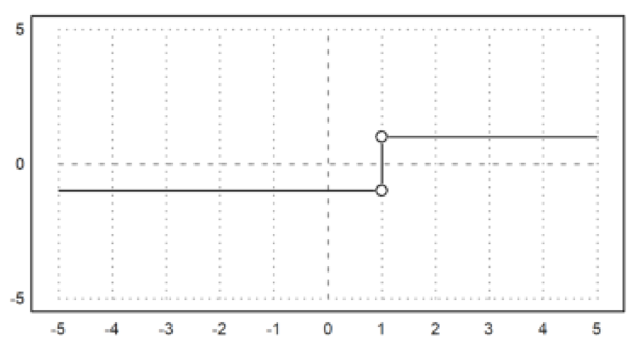
$$\lim_{x\uparrow 1}\frac{|x-1|}{x-1}=-1$$

Hitung limit di atas untuk x menuju 1 dari kanan.
Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit(abs(x-1)/(x-1),x,1,plus))
```

$$\lim_{x\downarrow 1}\frac{|x-1|}{x-1}=1$$

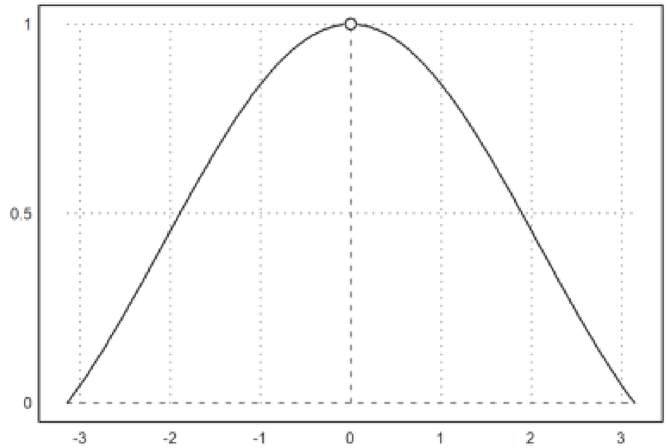
```
>aspect(2); plot2d("(abs(x-1))/(x-1)",-5,5,-5,5); plot2d(1,1,>points,style=
```




```
>$showev('limit(sin(x)/x,x,0))
```

$$\lim_{x\to 0}\frac{\sin x}{x}=1$$

```
>plot2d("sin(x)/x",-pi,pi); plot2d(0,1,>points,style="ow",>add):
```

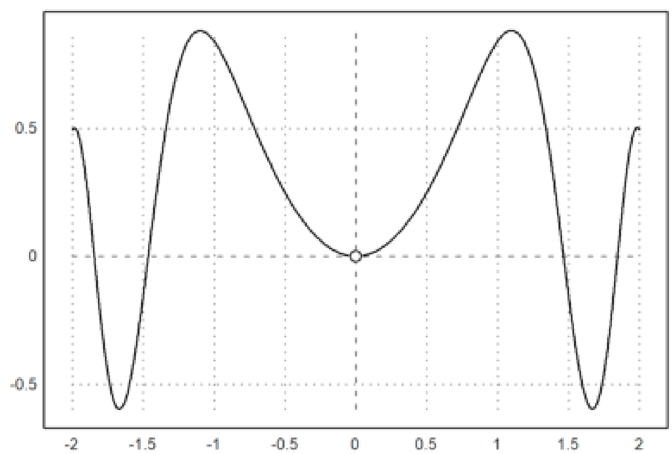


```
>$showev('limit(sin(x^3)/x,x,0))
```

$$\lim_{x\to 0}{\frac{\sin x^3}{x}}=0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>aspect(1.5); plot2d("(sin(x^3))/x"); plot2d(0,0,>points,style="ow",>add):
```



```
>$showev('limit(log(x), x, inf))
```

$$\lim_{x\to \infty }\log x=\infty$$

```
>$showev('limit((-2)^x, x, inf))
```

$$\lim_{x\rightarrow\infty}{\left(-2\right)^{x}}=\mathit{infinity}$$

```
>$showev('limit(t-sqrt(2-t),t,2,minus))
```

$$\lim_{t\uparrow 2}{t-\sqrt{2-t}}=2$$

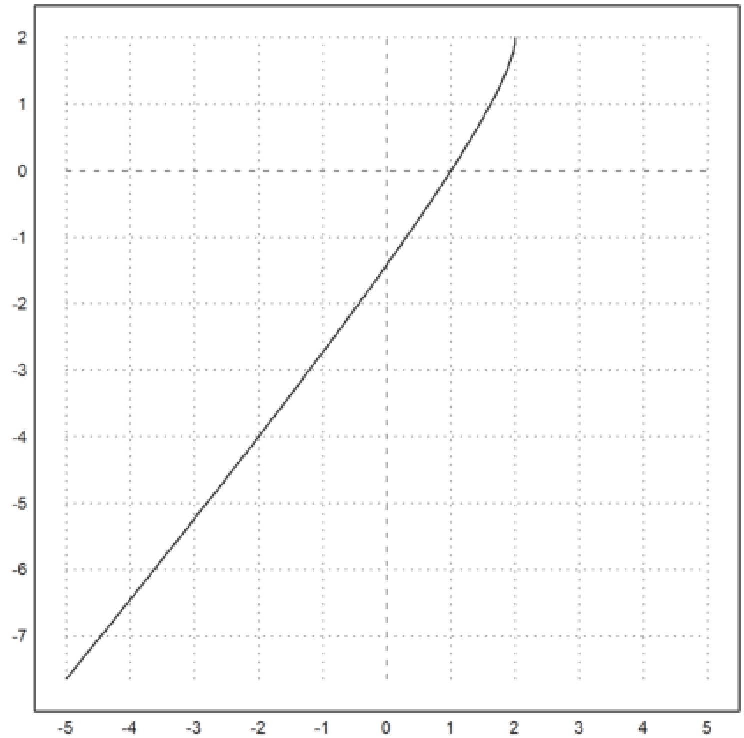
```
>$showev('limit(t-sqrt(2-t),t,2,plus))
```

$$\lim_{t\downarrow 2}{t-\sqrt{2-t}}=2$$

```
>$showev('limit(t-sqrt(2-t),t,5,plus)) // Perhatikan hasilnya
```

$$\lim_{t\downarrow 5}{t-\sqrt{2-t}}=5-\sqrt{3}\,i$$

```
>aspect(1); plot2d("x-sqrt(2-x)",-5,5):
```

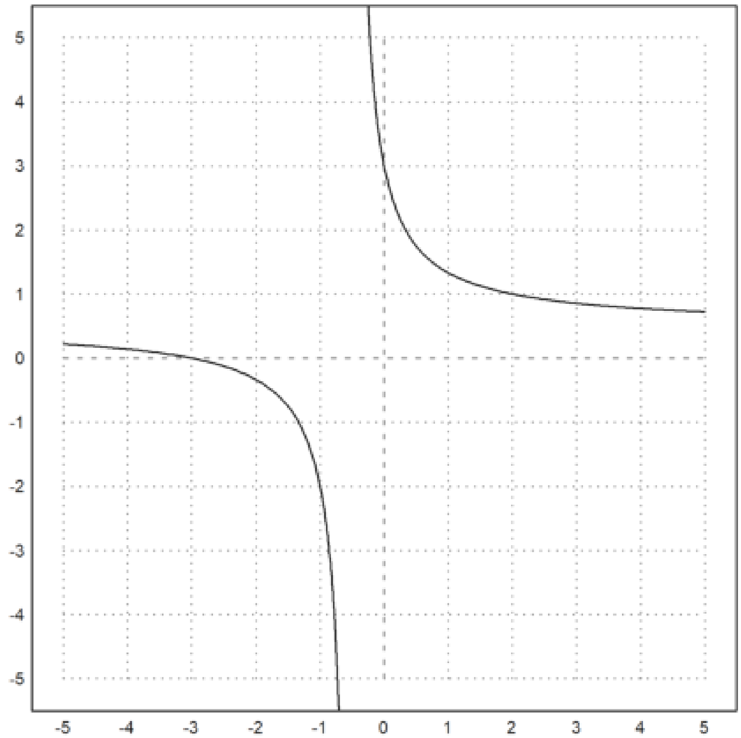



```
>$showev('limit((x^2-9)/(2*x^2-5*x-3),x,3))
```

$$\lim_{x\to 3}\frac{x^2-9}{2\,x^2-5\,x-3}=\frac{6}{7}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>aspect(1); plot2d("(x^2-9)/(2*x^2-5*x-3)",-5,5,-5,5):
```

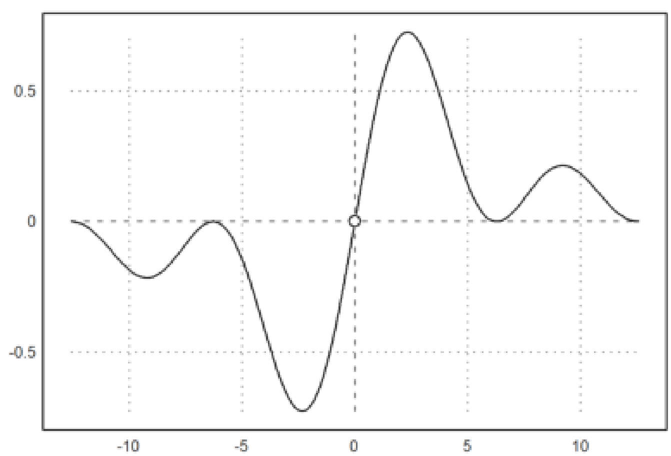



```
>$showev('limit((1-cos(x))/x,x,0))
```

$$\lim_{x\to 0}\frac{1-\cos x}{x}=0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(1-cos(x))/x",-4pi,4pi); plot2d(0,0,>points,style="ow",>add):
```

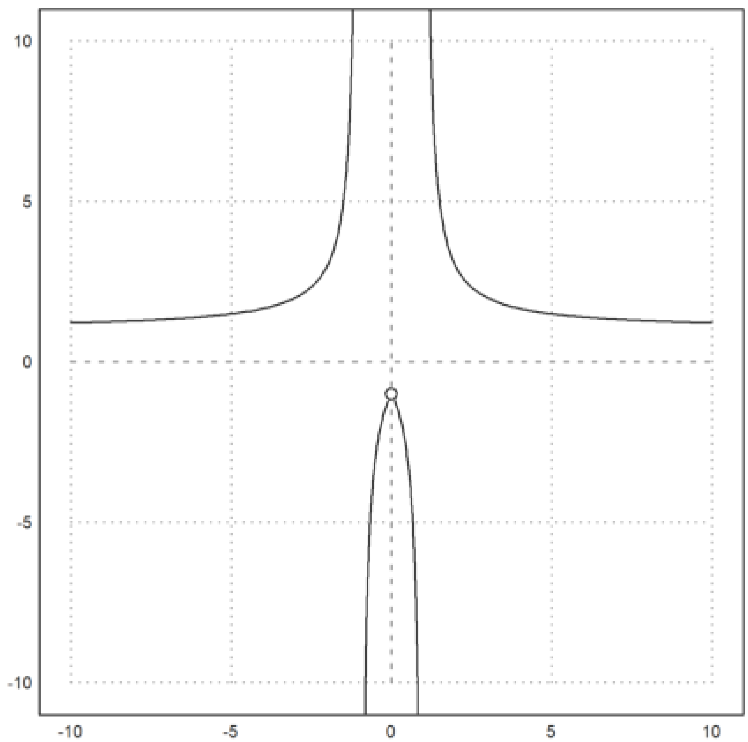


```
>$showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
```

$$\lim_{x\to 0} \frac{|x| + x^2}{x^2 - |x|} = -1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

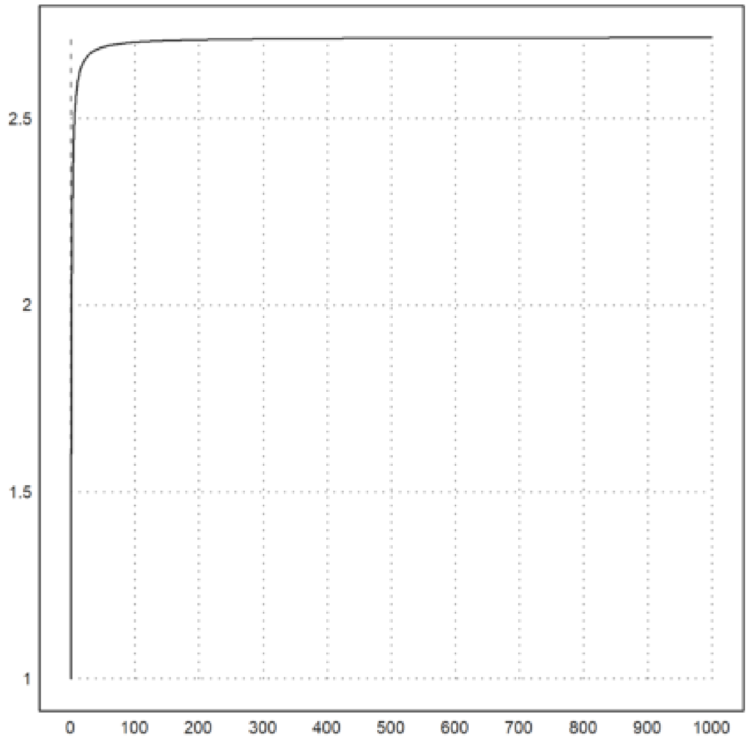
```
>aspect(1); plot2d("(x^2+abs(x))/(x^2-abs(x))",-10,10,-10,10); plot2d(0,-1,
```




```
>$showev('limit((1+1/x)^x,x,inf))
```

$$\lim_{x\to\infty} \left(\frac{1}{x} + 1\right)^x = e$$

```
>plot2d("(1+1/x)^x",0,1000):
```




```
>$showev('limit((1+k/x)^x,x,inf))
```

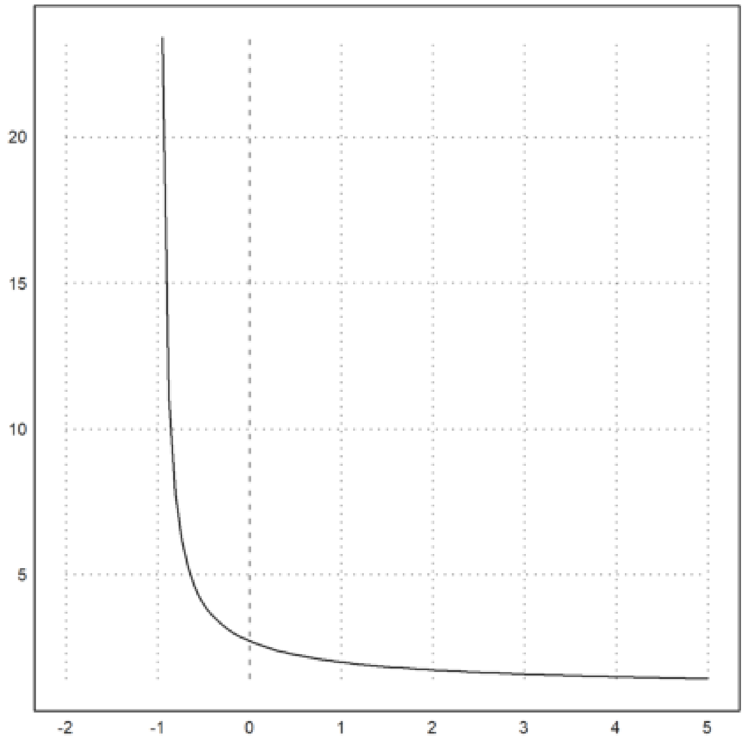
$$\lim_{x\to\infty}\left(\frac{k}{x}+1\right)^{x}=e^{k}$$

```
>$showev('limit((1+x)^(1/x),x,0))
```

$$\lim_{x\to 0}\left(x+1\right)^{\frac{1}{x}}=e$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(1+x)^(1/x)",-2,5):
```




```
>$showev('limit((x/(x+k))^x,x,inf))
```

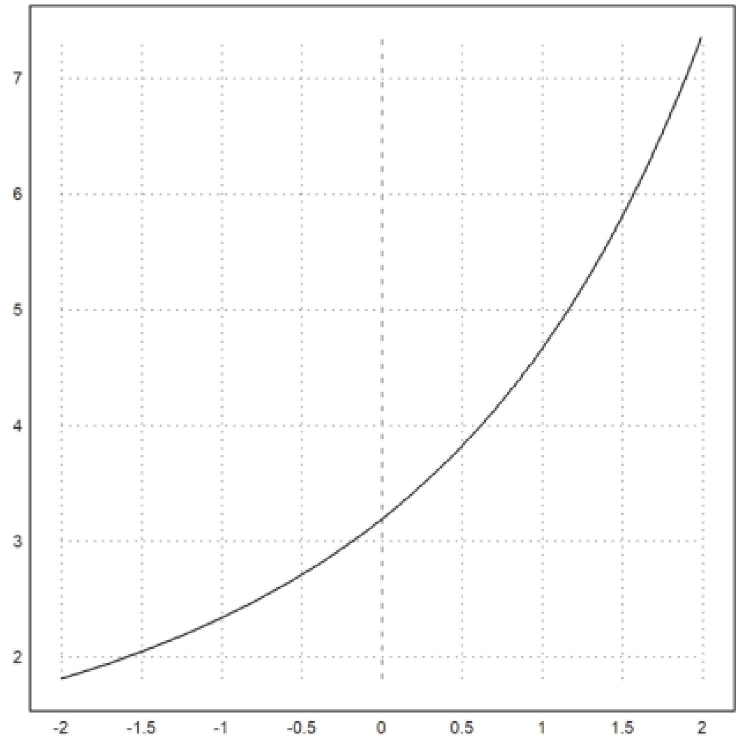
$$\lim_{x\to\infty}\left(\frac{x}{x+k}\right)^{x}=e^{-k}$$

```
>$showev('limit((E^x-E^2)/(x-2),x,2))
```

$$\lim_{x\to 2}\frac{e^{x}-e^{2}}{x-2}=e^{2}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(E^x-E^2)/(x-2)"):
```



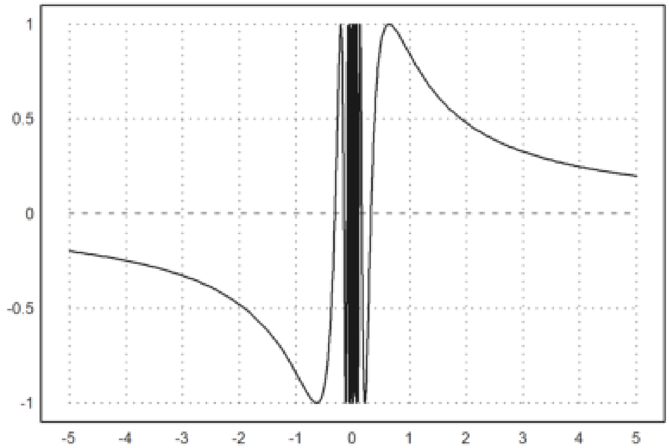

```
>$showev('limit(sin(1/x),x,0))
```

$$\lim_{x \to 0} \sin \left(\frac{1}{x} \right) = ind$$

```
>$showev('limit(sin(1/x),x,inf))
```

$$\lim_{x \to \infty} \sin \left(\frac{1}{x} \right) = 0$$

```
>aspect(1.5); plot2d("sin(1/x)",-5,5,-1,1):
```



Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung nilai limit fungsi tersebut di beberapa nilai dan di tak hingga. Gambar grafik fungsi tersebut untuk mengkonfirmasi nilai-nilai limit tersebut.

Soal 1

$$\lim_{x\to n} x^2 - 3x + 1$$

$$n = -2$$

```
>$showev('limit(x^2-3*x+1,x,-2))
```

$$\lim_{x\to -2} x^2 - 3\,x + 1 = 11$$

$$n = 6$$

```
>$showev('limit(x^2-3*x+1,x,6))
```

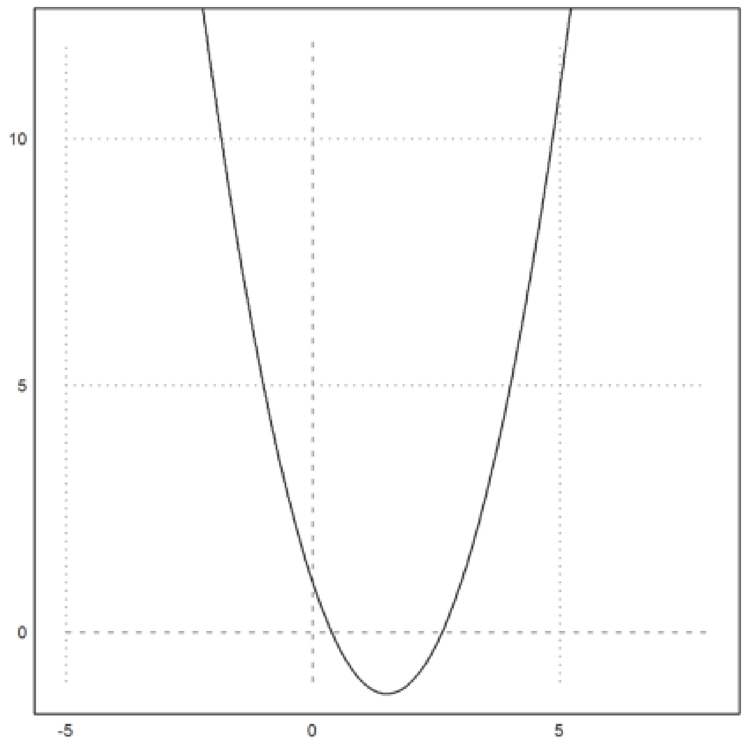
$$\lim_{x\to 6} x^2 - 3\,x + 1 = 19$$

$$n = \infty$$

```
>$showev('limit(x^2-3*x+1,x,inf))
```

$$\lim_{x\to\infty} x^2 - 3\,x + 1 = \infty$$

```
>plot2d("x^2-3*x+1",-5,8,-1,12):
```



Soal 2

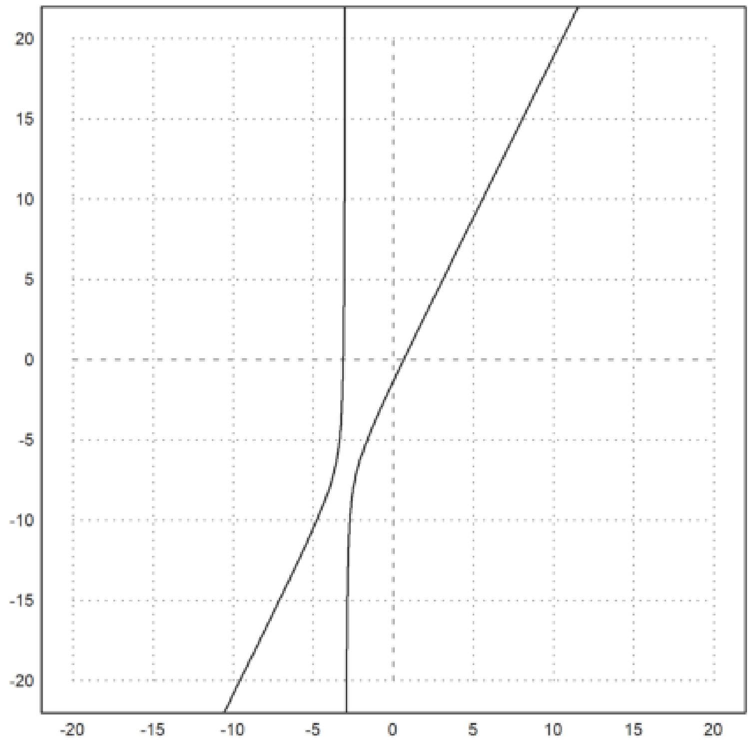
```
>$showev('limit((2*x^2+5*x-4)/(x+3),x,2))
```

$$\lim_{x\to 2} \frac{2\,x^2 + 5\,x - 4}{x + 3} = \frac{14}{5}$$

```
>$showev('limit((2*x^2+5*x-4)/(x+3),x,inf))
```

$$\lim_{x\to\infty} \frac{2\,x^2 + 5\,x - 4}{x + 3} = \infty$$

```
>plot2d("(2*x^2+5*x-4)/(x+3)",-20,20,-20,20):
```



Soal 3

```
>$showev('limit((x-1)^(1/2)/(2*x+1),x,1,minus))
```

$$\lim_{x\uparrow 1}\frac{\sqrt{x-1}}{2\,x+1}=0$$

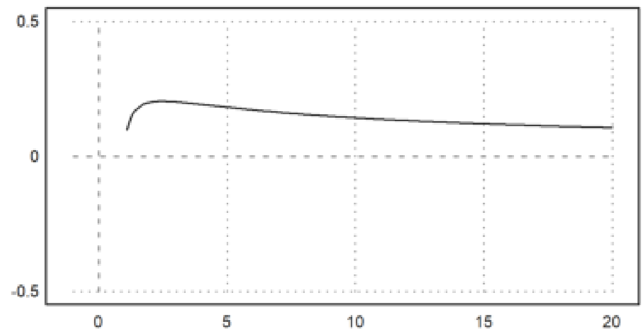
```
>$showev('limit((x-1)^(1/2)/(2*x+1),x,1,plus))
```

$$\lim_{x\downarrow 1}\frac{\sqrt{x-1}}{2\,x+1}=0$$

```
>$showev('limit((x-1)^(1/2)/(2*x+1),x,1))
```

$$\lim_{x\rightarrow 1}\frac{\sqrt{x-1}}{2\,x+1}=0$$

```
>aspect(2); plot2d("(x-1)^(1/2)/(2*x+1)",-1,20,-0.5,0.5):
```

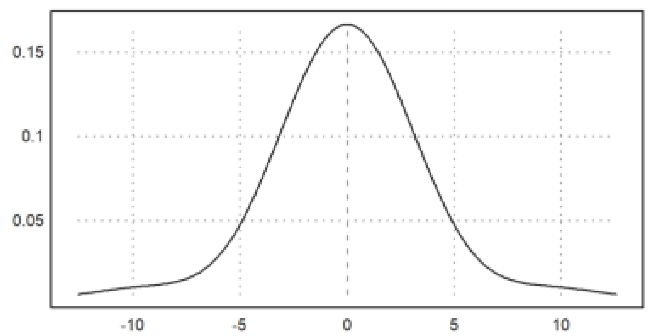


Soal 4

```
>$showev('limit((x-sin(x))/(x^3),x,0))
```

$$\lim_{x \to 0} \frac{x - \sin x}{x^3} = \frac{1}{6}$$

```
>plot2d("(x-sin(x))/(x^3)",-4pi,4pi):
```



Soal 5

```
>function p(x)&=2*x
```

```
2 x
```

```
>function map p(x)
```

```
    if x<2 then return x+3
    else x>=2 return x^2+1
    endif;
  endfunction
```

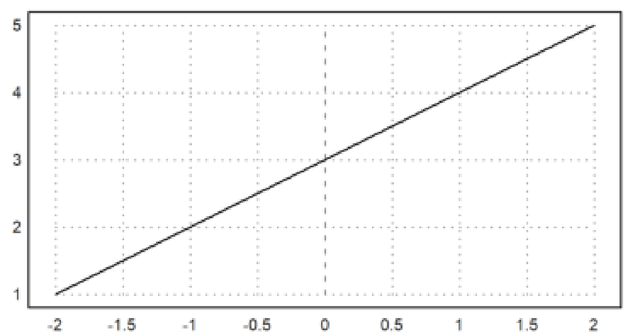
```
>p(2)
```

```
  5
```

```
>$showev('limit(x^2+1,x,2))
```

$$\lim_{x\to 2} x^2+1=5$$

```
>plot2d("p(x)"):
```

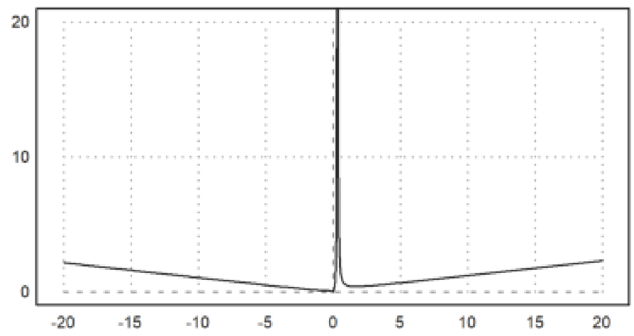


Soal 6

```
>$showev('limit(((x^2+1)*abs(x))/(3*x-1)^2,x,2,plus))
```

$$\lim_{x\downarrow 2} \frac{(x^2+1)\;|x|}{(3\,x-1)^2}=\frac{2}{5}$$

```
>aspect(2); plot2d("((x^2+1)*abs(x))/(3*x-1)^2",-20,20,0,20):
```



Turunan Fungsi

Definisi turunan:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).

```
>$showev('limit(((x+h)^2-x^2)/h,h,0)) // turunan x^2
```

$$\lim_{h \to 0} \frac{(x+h)^2 - x^2}{h} = 2\,x$$

```
>p &= expand((x+h)^2-x^2)|simplify; $p //pembilang dijabarkan dan disederha
```

$$2\,h\,x + h^2$$

```
>q &=ratsimp(p/h); $q // ekspresi yang akan dihitung limitnya disederhanaka
```

$$2\,x+h$$

```
>$limit(q,h,0) // nilai limit sebagai turunan
```

$$2\,x$$

```
>$showev('limit(((x+h)^n-x^n)/h,h,0)) // turunan x^n
```

$$\lim_{h\rightarrow 0}{\frac{\left(x+h\right)^{n}-x^{n}}{h}}=n\,x^{n-1}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan (x+h)^n dengan menggunakan teorema binomial.

Penjelasan:

$$f'(x)=\lim_{h\to 0}\frac{(x+h)^n-x^n}{h}$$

$$=\lim_{h\to 0}\frac{x^n+nx^{n-1}h+\frac{n(n-1)}{2}x^{n-2}h^2+...+nxh^{n-1}+h^n-x^n}{h}$$

$$=\lim_{h\to 0}\frac{h\left[nx^{n-1}+\frac{n(n-1)}{2}x^{n-2}h+...+nxh^{n-2}+h^{n-1}\right]}{h}$$

$$=\lim_{h\to 0}nx^{n-1}+\frac{n(n-1)}{2}x^{n-2}h+...+nxh^{n-2}+h^{n-1}$$

$$f'(x)=nx^{n-1}$$

```
>$showev('limit((sin(x+h)-sin(x))/h,h,0)) // turunan sin(x)
```

$$\lim_{h\rightarrow 0}{\frac{\sin \left(x+h\right)-\sin x}{h}}=\cos x$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan sin(x+h) dengan menggunakan rumus jumlah dua sudut.

Penjelasan:

$$D_x(sin\ x) = \lim_{h\to 0} \frac{sin(x+h) - sin\ x}{h}$$

$$= \lim_{h\to 0} \frac{sin\ x\ cos\ h + cos\ x\ sin\ h - sin\ x}{h}$$

$$= \lim_{h\to 0}(-sin\ x\frac{1-cos\ h}{h} + cos\ x\frac{sin\ h}{h})$$

$$= (-sin\ x)\left[\lim_{h\to 0}\frac{1-cos\ h}{h}\right] + (cos\ x)\left[\lim_{h\to 0}\frac{sin\ h}{h}\right]$$

$$= (-sin\ x)\cdot 0 + (cos\ x)\cdot 1$$

$$= cos\ x$$

```
>$showev('limit((log(x+h)-log(x))/h,h,0)) // turunan log(x)
```

$$\lim_{h\to 0}\frac{\log(x+h)-\log x}{h} = \frac{1}{x}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, gunakan sifat-sifat logaritma dan hasil limit pada bagian sebelumnya di atas.

Penjelasan:

$$D_x log(x) = \lim_{h\to 0}\frac{log(x+h) - log\ x}{h}$$

$$= \lim_{h\to 0}\frac{log\left(\frac{x+h}{x}\right)}{h}$$

$$= \lim_{h\to 0}\frac{1}{h}\left(log\left(1+\frac{h}{x}\right)\right)$$

$$= \lim_{h\to 0}\left(log\left(1+\frac{h}{x}\right)^{\frac{1}{h}}\right)$$

$$= log\left(\lim_{h\to 0}\left(1+\frac{h}{x}\right)^{\frac{1}{h}}\right)$$

Dengan

$$\lim_{x\to 0}(1+x)^{\frac{1}{x}} = e$$

maka

$$log\left(\lim_{h\to 0}\left(1+\frac{h}{x}\right)^{\frac{1}{h}}\right)^{\frac{h}{x}\cdot\frac{1}{h}}$$

$$= log\ e^{\frac{1}{x}}$$

$$= \frac{1}{x} log\ e$$

$$= \frac{1}{x}\cdot 1$$

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

```
>$showev('limit((1/(x+h)-1/x)/h,h,0)) // turunan 1/x
```

$$\lim_{h\to 0}\frac{\frac{1}{x+h}-\frac{1}{x}}{h} = -\frac{1}{x^2}$$

```
>$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x
```

```
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Maxima is asking
Acceptable answers are: yes, y, Y, no, n, N, unknown, uk
Is x an integer?

Use assume!
```

```
Error in:
 $showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x ...
                                             ^
```

Maxima bermasalah dengan limit:

$$\lim_{h\to 0} \frac{e^{x+h}-e^x}{h}.$$

Oleh karena itu diperlukan trik khusus agar hasilnya benar.

```
>$showev('limit((E^h-1)/h,h,0))
```

$$\lim_{h\to 0} \frac{e^h-1}{h} = 1$$

```
>$showev('factor(E^(x+h)-E^x))
```

$$\mathit{factor}\left(e^{x+h}-e^x\right) = \left(e^h-1\right)\, e^x$$

```
>$showev('limit(factor((E^(x+h)-E^x)/h),h,0)) // turunan f(x)=e^x
```

$$\left(\lim_{h\to 0} \frac{e^h-1}{h}\right)\, e^x = e^x$$

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

```
                                   x
                                  x
```

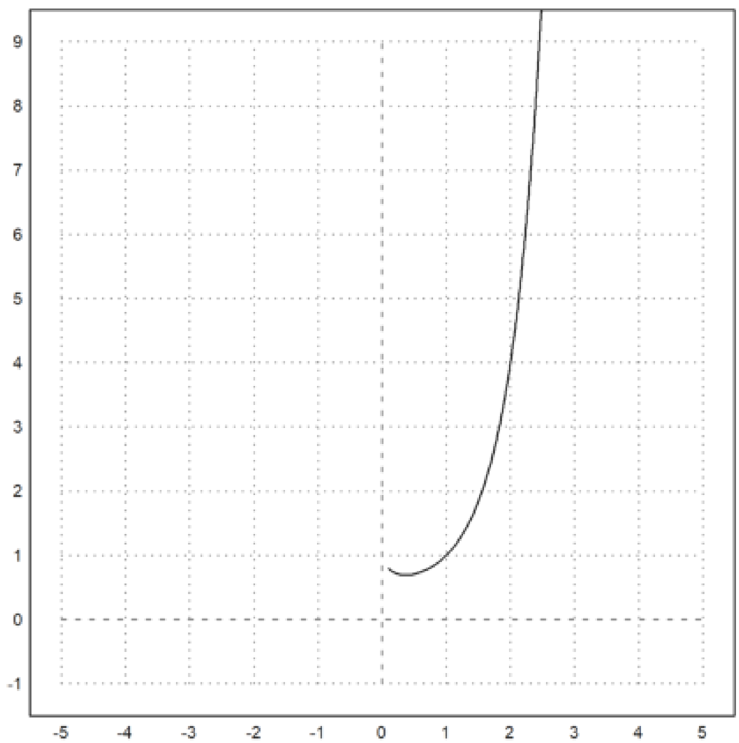
```
>$showev('limit(f(x),x,0))
```

$$\lim_{x\to 0} x^x = 1$$

Silakan Anda gambar kurva

$$y = x^x.$$

```
>aspect(1); plot2d("x^x",-5,5,-1,9):
```



```
>$showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h\to 0}\frac{(x+h)^{x+h}-x^x}{h} = \mathit{infinity}$$

Di sini Maxima juga bermasalah terkait limit:

$$\lim_{h\to 0}\frac{(x+h)^{x+h}-x^x}{h}.$$

Dalam hal ini diperlukan asumsi nilai x.

```
>&assume(x>0); $showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h\to 0}\frac{(x+h)^{x+h}-x^x}{h}=x^x\left(\log x+1\right)$$

```
>&forget(x>0) // jangan lupa, lupakan asumsi untuk kembali ke semula
```

```
[x > 0]
```

```
>&forget(x<0)
```

```
[x < 0]
```

```
>&facts()
```

```
[]
```

```
>$showev('limit((asin(x+h)-asin(x))/h,h,0)) // turunan arcsin(x)
```

$$\lim_{h\to 0}\frac{\arcsin\left(x+h\right)-\arcsin x}{h}=\frac{1}{\sqrt{1-x^2}}$$

```
>$showev('limit((tan(x+h)-tan(x))/h,h,0)) // turunan tan(x)
```

$$\lim_{h\to 0}\frac{\tan\left(x+h\right)-\tan x}{h}=\frac{1}{\cos^2 x}$$

```
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
```

```
sinh(x)
```

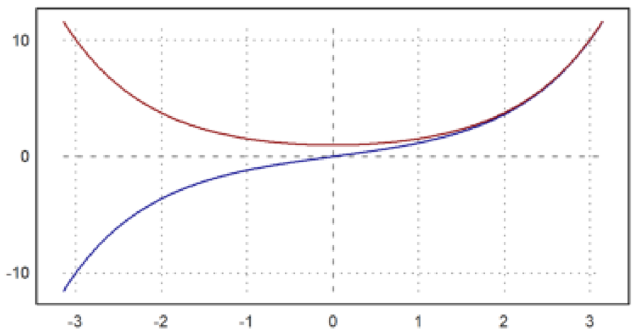
```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

$$\frac{e^{-x}\,\left(e^{2\,x}+1\right)}{2}$$

Hasilnya adalah cosh(x), karena

$$\frac{e^x+e^{-x}}{2}=\cosh(x).$$

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]):
```




```
>function f(x) &= sin(3*x^5+7)^2
```

```
                                 2      5
                              sin  (3 x  + 7)
```

```
>diff(f,3), diffc(f,3)
```

```
1198.32948904
1198.72863721
```

Apakah perbedaan diff dan diffc?

```
>$showev('diff(f(x),x))
```

$$\frac{d}{d\,x}\,\sin^2\left(3\,x^5+7\right)=30\,x^4\,\cos\left(3\,x^5+7\right)\,\sin\left(3\,x^5+7\right)$$

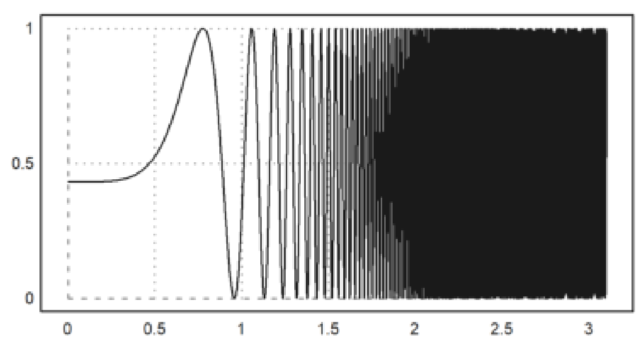
```
>$% with x=3
```

$$\%at\left(\frac{d}{d\,x}\,\sin^2\left(3\,x^5+7\right),x=3\right)=2430\,\cos 736\,\sin 736$$

```
>$float(%)
```

$$\%at\left(\frac{d^{1.0}}{d\,x^{1.0}}\,\sin^2\left(3.0\,x^5+7.0\right),x=3.0\right)=1198.728637211748$$

```
>plot2d(f,0,3.1):
```




```
>function f(x) &=5*cos(2*x)-2*x*sin(2*x) // mendifinisikan fungsi f
```

```
5 cos(2 x) - 2 x sin(2 x)
```

```
>function df(x) &=diff(f(x),x) // fd(x) = f'(x)
```

```
- 12 sin(2 x) - 4 x cos(2 x)
```

```
>$'f(1)=f(1), $float(f(1)), $'f(2)=f(2), $float(f(2)) // nilai f(1) dan f(2
```

$$-0.2410081230863468$$

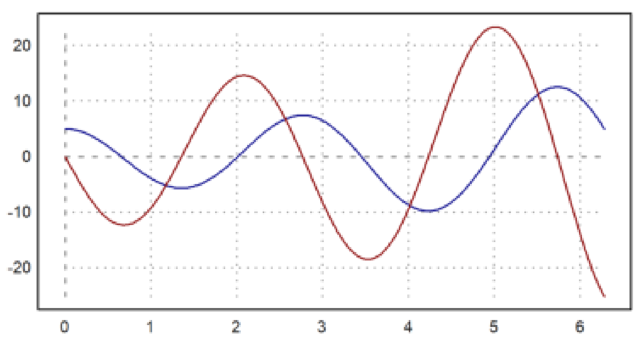
```
>xp=solve("df(x)",1,2,0) // solusi f'(x)=0 pada interval [1, 2]
```

```
1.35822987384
```

```
>df(xp), f(xp) // cek bahwa f'(xp)=0 dan nilai ekstrim di titik tersebut
```

```
0
-5.67530133759
```

```
>plot2d(["f(x)","df(x)"],0,2*pi,color=[blue,red]): //grafik fungsi dan turu
```



Perhatikan titik-titik "puncak" grafik y=f(x) dan nilai turunan pada saat grafik fungsinya mencapai titik "puncak" tersebut.

Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, tentukan turunannya dengan menggunakan definisi turunan (limit), menggunakan perintah diff, dan secara manual (langkah demi langkah yang dihitung dengan Maxima) seperti contoh-contoh di atas. Gambar grafik fungsi asli dan fungsi turunannya pada sumbu koordinat yang sama.

Soal 1

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

```
>function f(x) &= x^3-2*x; $f(x)
```

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

```
>$showev('limit((f(x+h)-f(x))/h,h,0))
```

$$\lim_{h\rightarrow 0}{\frac{\left(x+h\right)^3-x^3-2\,\left(x+h\right)+2\,x}{h}} = 3\,x^2-2$$

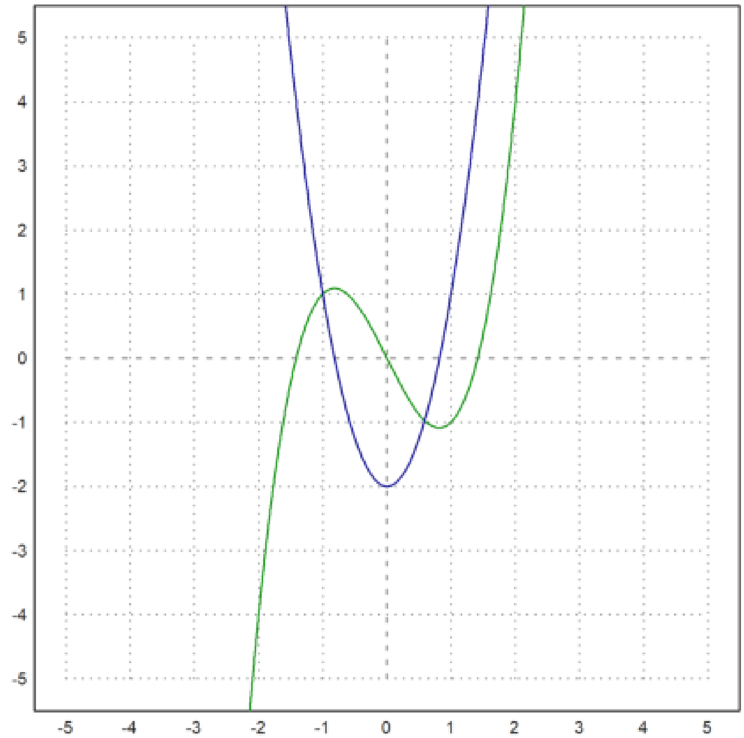
```
>$showev('diff(f(x),x))
```

$$\frac{d}{d\,x}\,\left(x^3-2\,x\right) = 3\,x^2-2$$

```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x)
```

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

```
>plot2d(["f(x)","df(x)"],-5,5,-5,5,color=[green,blue]):
```



Soal 2

$$g(x) = 6sinxcosx$$

```
>function g(x) &= 6*sin(x)*cos(x); $g(x)
```

$$6\ \cos x\ \sin x$$

```
>$showev('limit((g(x+h)-g(x))/h,h,0))
```

$$\lim_{h\rightarrow 0}{\frac{6\ \cos \left(x+h\right)\ \sin \left(x+h\right)-6\ \cos x\ \sin x}{h}}=-6\ \left(\sin ^2x-\cos ^2x\right)$$

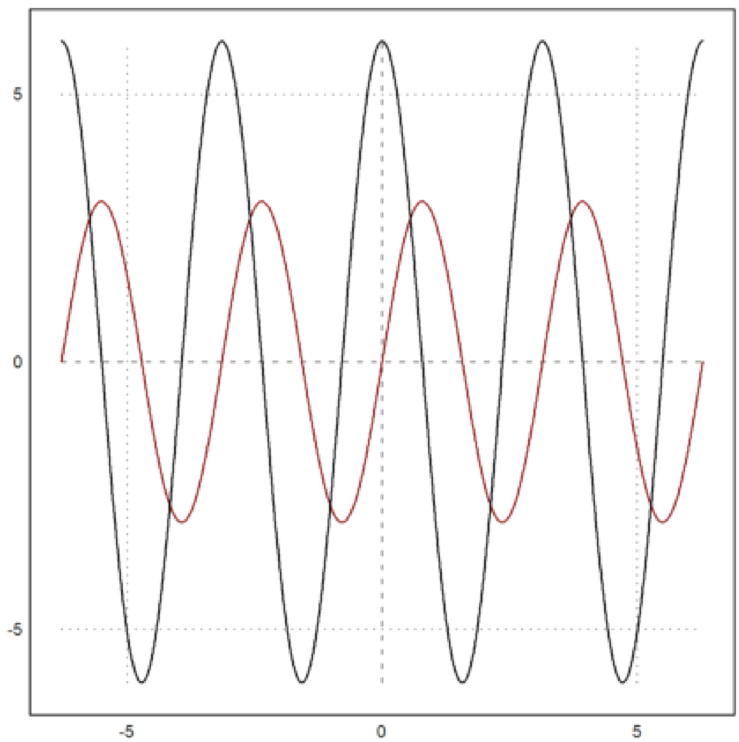
```
>$showev('diff(g(x),x))
```

$$\frac{d}{d\,x}\,\left(6\ \cos x\ \sin x\right)=6\ \cos ^2x-6\ \sin ^2x$$

```
>function dg(x) &= limit((g(x+h)-g(x))/h,h,0); $dg(x)
```

$$-6\,\left(\sin^2 x-\cos^2 x\right)$$

```
>plot2d(["g(x)","dg(x)"],-2pi,2pi,color=[red,black]):
```



Soal 3

$$k(x)=\frac{2x^2-3x+1}{2x+1}$$

```
>function k(x) &= (2*x^2-3*x+1)/(2*x+1); $k(x)
```

$$\frac{2\,x^2-3\,x+1}{2\,x+1}$$

```
>$showev('limit((k(x+h)-k(x))/h,h,0))
```

$$\lim_{h\to 0}\frac{\frac{2\,(x+h)^2-3\,(x+h)+1}{2\,(x+h)+1}-\frac{2\,x^2-3\,x+1}{2\,x+1}}{h}=\frac{4\,x^2+4\,x-5}{4\,x^2+4\,x+1}$$

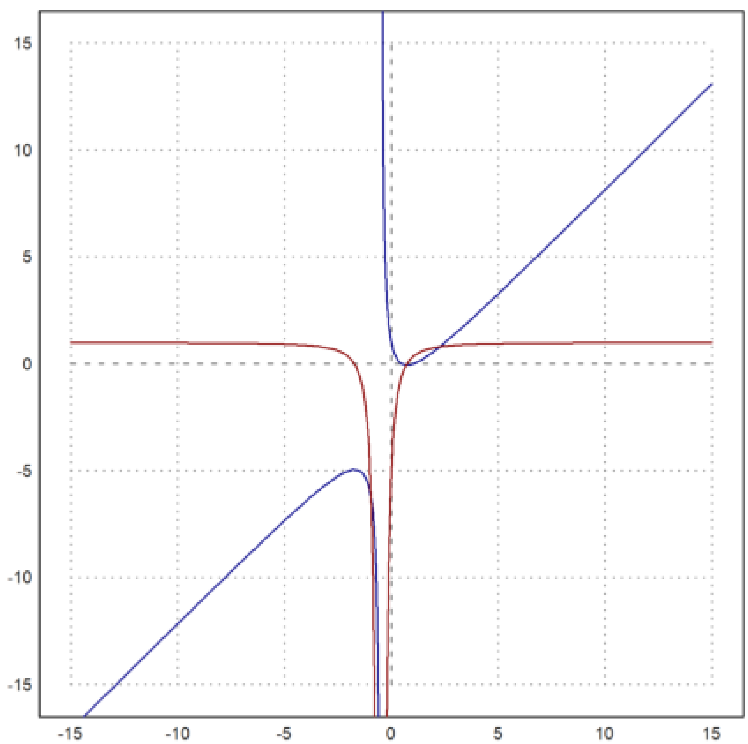
```
>$showev('diff(k(x),x))
```

$$\frac{d}{d\,x}\left(\frac{2\,x^2-3\,x+1}{2\,x+1}\right)=\frac{4\,x-3}{2\,x+1}-\frac{2\,\left(2\,x^2-3\,x+1\right)}{\left(2\,x+1\right)^2}$$

```
>function dk(x) &= limit((k(x+h)-k(x))/h,h,0); $dk(x)
```

$$\frac{4\,x^2+4\,x-5}{4\,x^2+4\,x+1}$$

```
>plot2d(["k(x)","dk(x)"],-15,15,-15,15,color=[blue,red]):
```



Soal 4

$$n(x) = \frac{1}{4x^2 - 3x + 9}$$

```
>function n(x) &= 1/(4*x^2-3*x+9); $n(x)
```

$$\frac{1}{4\,x^2-3\,x+9}$$

```
>$showev('limit((n(x+h)-n(x))/h,h,0))
```

$$\lim_{h\rightarrow 0}{\frac{\frac{1}{4\,(x+h)^2-3\,(x+h)+9}-\frac{1}{4\,x^2-3\,x+9}}{h}}=\frac{3-8\,x}{16\,x^4-24\,x^3+81\,x^2-54\,x+81}$$

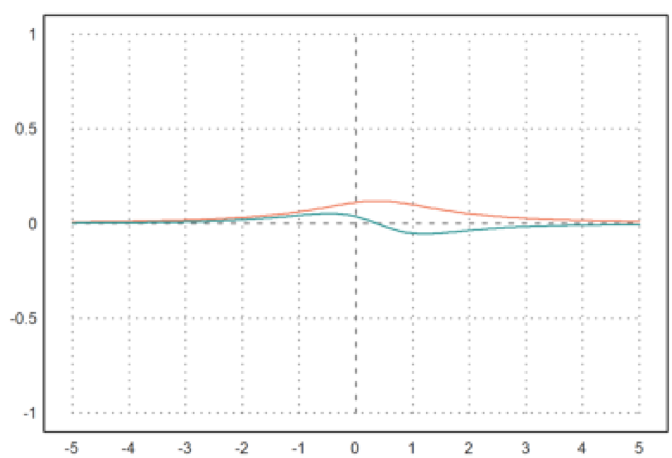
```
>$showev('diff(n(x),x))
```

$$\frac{d}{d\,x}\,\left(\frac{1}{4\,x^2-3\,x+9}\right)=\frac{3-8\,x}{\left(4\,x^2-3\,x+9\right)^2}$$

```
>function dn(x) &= limit((n(x+h)-n(x))/h,h,0); $dn(x)
```

$$\frac{3-8\,x}{16\,x^4-24\,x^3+81\,x^2-54\,x+81}$$

```
>aspect(1.5); plot2d(["n(x)","dn(x)"],-5,5,-1,1,color=[orange,cyan]):
```



Soal 5

$$p(x) = x^2 e^x$$

```
>function p(x) &= x^2*E^x; $p(x)
```

$$x^2\, e^x$$

```
>$showev('limit((p(x+h)-p(x))/h,h,0))
```

```
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Maxima is asking
Acceptable answers are: yes, y, Y, no, n, N, unknown, uk
Is x an integer?

Use assume!
Error in:
 $showev('limit((p(x+h)-p(x))/h,h,0)) ...
                                    ^
```

```
>$showev('limit((E^h-1)/h,h,0))
```

$$\lim_{h\rightarrow 0}{\frac{e^{h}-1}{h}}=1$$

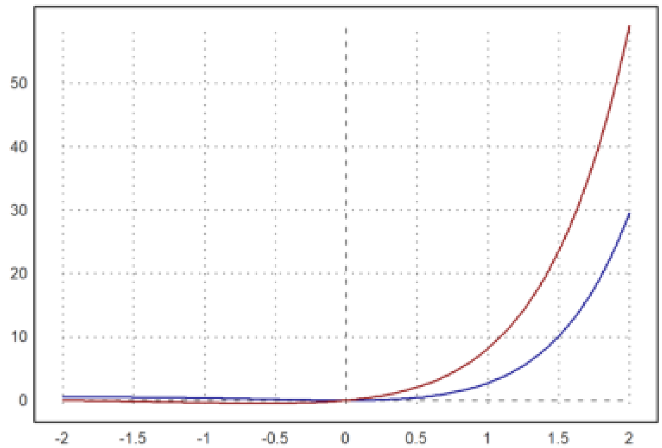
```
>$factor((x+h)^2*E^(x+h)-(x^2*E^x))
```

$$\left(e^{h}\,x^2-x^2+2\,h\,e^{h}\,x+h^2\,e^{h}\right)\,e^{x}$$

```
>$showev('limit(factor(((x+h)^2*E^(x+h)-(x^2*E^x))/h),h,0))
```

$$e^{x}\,\left(\lim_{h\rightarrow 0}{\frac{e^{h}\,x^2-x^2+2\,h\,e^{h}\,x+h^2\,e^{h}}{h}}\right)=\left(x^2+2\,x\right)\,e^{x}$$

```
>plot2d(["p(x)","E^x*(x^2+2x)"], color=[blue,red]):
```




```
>reset();
```

Integral

EMT dapat digunakan untuk menghitung integral, baik integral tak tentu maupun integral tentu. Untuk integral tak tentu (simbolik) sudah tentu EMT menggunakan Maxima, sedangkan untuk perhitungan integral tentu EMT sudah menyediakan beberapa fungsi yang mengimplementasikan algoritma kuadratur (perhitungan integral tentu menggunakan metode numerik).

Pada notebook ini akan ditunjukkan perhitungan integral tentu dengan menggunakan Teorema Dasar Kalkulus:

$$\int_a^b f(x)\,dx = F(b)-F(a), \quad \text{dengan } F'(x)=f(x).$$

Fungsi untuk menentukan integral adalah integrate. Fungsi ini dapat digunakan untuk menentukan, baik integral tentu maupun tak tentu (jika fungsinya memiliki antiderivatif). Untuk perhitungan integral tentu fungsi integrate menggunakan metode numerik (kecuali fungsinya tidak integrabel, kita tidak akan menggunakan metode ini).

```
>$showev('integrate(x^n,x))
```

```
Answering "Is n equal to -1?" with "no"
```

$$\int x^n\,dx = \frac{x^{n+1}}{n+1}$$

```
>$showev('integrate(1/(1+x),x))
```

$$\int \frac{1}{x+1}\,dx = \log\left(x+1\right)$$

```
>$showev('integrate(1/(1+x^2),x))
```

$$\int \frac{1}{x^2+1}\,dx = \arctan x$$

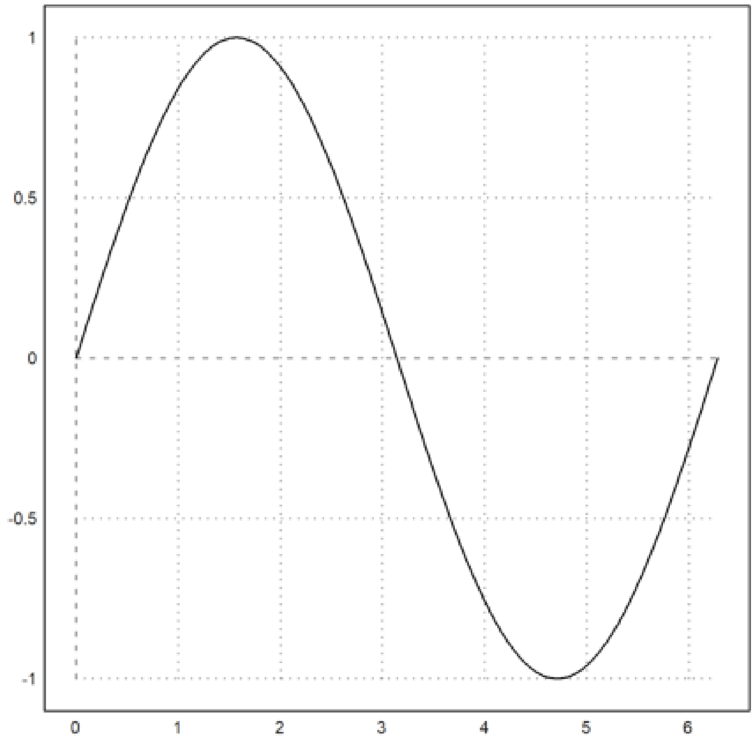
```
>$showev('integrate(1/sqrt(1-x^2),x))
```

$$\int \frac{1}{\sqrt{1-x^2}}\,dx = \arcsin x$$

```
>$showev('integrate(sin(x),x,0,pi))
```

$$\int_0^{\pi} \sin x\,dx = 2$$

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




```
>$showev('integrate(sin(x),x,a,b))
```

$$\int_{a}^{b}{\sin x\;dx}=\cos a-\cos b$$

```
>$showev('integrate(x^n,x,a,b))
```

```
Answering "Is n positive, negative or zero?" with "positive"
```

$$\int_{a}^{b}{x^{n}\;dx}=\frac{b^{n+1}}{n+1}-\frac{a^{n+1}}{n+1}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x))
```

$$\int {x^2\,\sqrt{2\,x+1}}{\;dx}=\frac{\left(2\,x+1\right)^{\frac{7}{2}}}{28}-\frac{\left(2\,x+1\right)^{\frac{5}{2}}}{10}+\frac{\left(2\,x+1\right)^{\frac{3}{2}}}{12}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x,0,2))
```

$$\int_{0}^{2}x^2\,\sqrt{2\,x+1}\,dx=\frac{2\,5^{\frac{5}{2}}}{21}-\frac{2}{105}$$

```
>$ratsimp(%)
```

$$\int_{0}^{2}x^2\,\sqrt{2\,x+1}\,dx=\frac{2\,5^{\frac{7}{2}}-2}{105}$$

```
>$showev('integrate((sin(sqrt(x)+a)*E^sqrt(x))/sqrt(x),x,0,pi^2))
```

$$\int_{0}^{\pi^2}\frac{\sin\left(\sqrt{x}+a\right)\,e^{\sqrt{x}}}{\sqrt{x}}\,dx=\left(-e^{\pi}-1\right)\,\sin a+\left(e^{\pi}+1\right)\,\cos a$$

```
>$factor(%)
```

$$\int_{0}^{\pi^2}\frac{\sin\left(\sqrt{x}+a\right)\,e^{\sqrt{x}}}{\sqrt{x}}\,dx=\left(-e^{\pi}-1\right)\,\left(\sin a-\cos a\right)$$

```
>function map f(x) &= E^(-x^2)
```

```
                                         2
                                      - x
                                     E
```

```
>$showev('integrate(f(x),x))
```

$$\int e^{-x^2}\,dx=\frac{\sqrt{\pi}\,\mathrm{erf}\left(x\right)}{2}$$

Fungsi f tidak memiliki antiturunan, integralnya masih memuat integral lain.

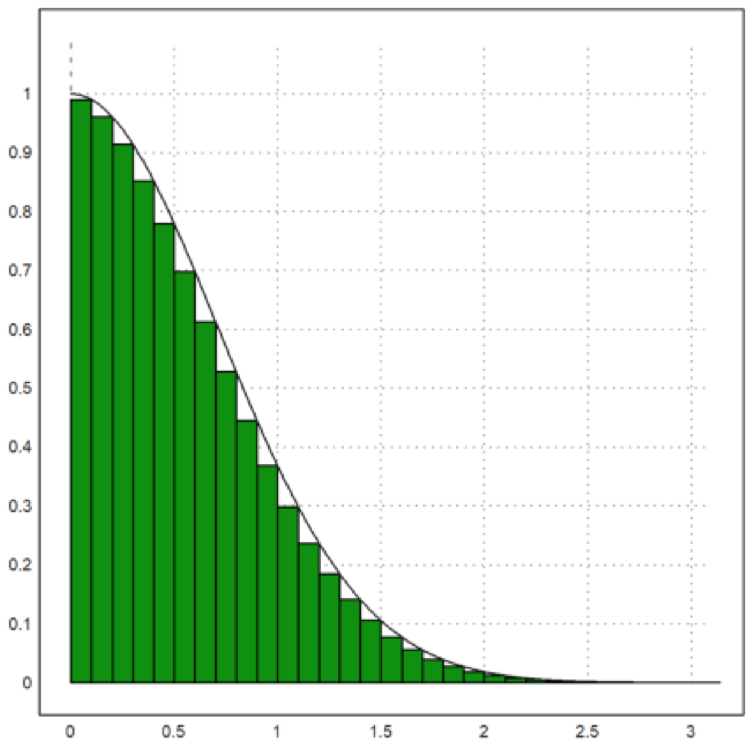
$$erf(x) = \int \frac{e^{-x^2}}{\sqrt{\pi}}\, dx.$$

Kita tidak dapat menggunakan teorema Dasar kalkulus untuk menghitung integral tentu fungsi tersebut jika semua batasnya berhingga. Dalam hal ini dapat digunakan metode numerik (rumus kuadratur).

Misalkan kita akan menghitung:

$$\int_0^{\pi} e^{-x^2}\, dx$$

```
>x=0:0.1:pi-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,pi,>add):
```



Integral tentu

$$\int_0^{\pi} f(x)\, dx$$

dapat dihampiri dengan jumlah luas persegi-persegi panjang di bawah kurva y=f(x) tersebut. Langkah-langkahnya adalah sebagai berikut.

```
>t &= makelist(a,a,0,pi-0.1,0.1); // t sebagai list untuk menyimpan nilai-n
>fx &= makelist(f(t[i]+0.1),i,1,length(t)); // simpan nilai-nilai f(x)
>// jangan menggunakan x sebagai list, kecuali Anda pakar Maxima!
```

Hasilnya adalah:

$$\int_{0}^{\pi}e^{-x^2}\,dx=0.8362196102528469$$

Jumlah tersebut diperoleh dari hasil kali lebar sub-subinterval (=0.1) dan jumlah nilai-nilai f(x) untuk x = 0.1, 0.2, 0.3, ..., 3.2.

```
>0.1*sum(f(x+0.1)) // cek langsung dengan perhitungan numerik EMT
```

```
0.836219610253
```

Untuk mendapatkan nilai integral tentu yang mendekati nilai sebenarnya, lebar sub-intervalnya dapat diperkecil lagi, sehingga daerah di bawah kurva tertutup semuanya, misalnya dapat digunakan lebar subinterval 0.001. (Silakan dicoba!)

Meskipun Maxima tidak dapat menghitung integral tentu fungsi tersebut untuk batas-batas yang berhingga, namun integral tersebut dapat dihitung secara eksak jika batas-batasnya tak hingga. Ini adalah salah satu keajaiban di dalam matematika, yang terbatas tidak dapat dihitung secara eksak, namun yang tak hingga malah dapat dihitung secara eksak.

```
>$showev('integrate(f(x),x,0,inf))
```

$$\int_{0}^{\infty}e^{-x^2}\,dx=\frac{\sqrt{\pi}}{2}$$

Tunjukkan kebenaran hasil di atas!

Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

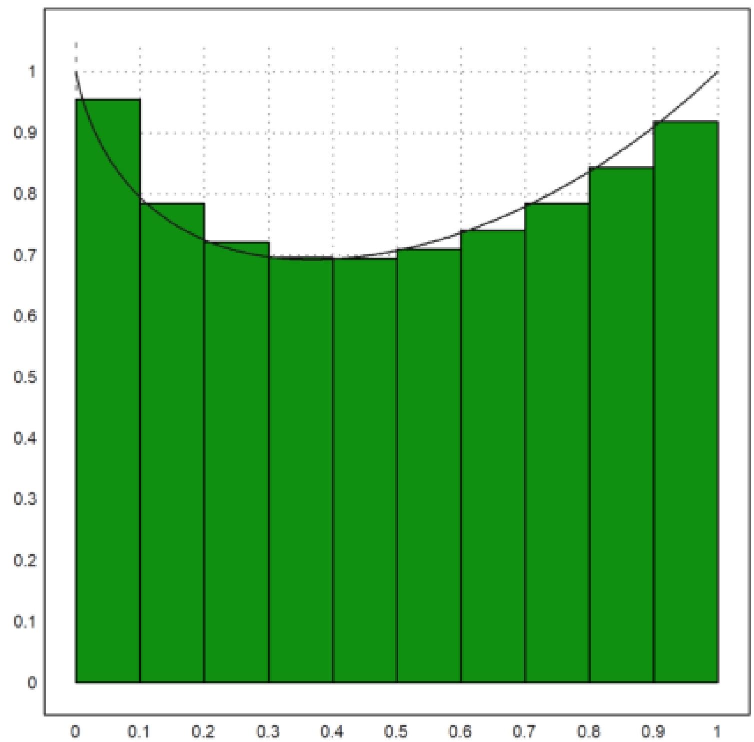
```
>function f(x) &= x^x
```

```
 x
x
```

```
>$showev('integrate(f(x),x,0,1))
```

$$\int_{0}^{1}{x^{x}\;dx}=\int_{0}^{1}{x^{x}\;dx}$$

```
>x=0:0.1:1-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```



Maxima gagal menghitung integral tentu tersebut secara langsung menggunakan perintah integrate. Berikut kita lakukan seperti contoh sebelumnya untuk mendapat hasil atau pendekatan nilai integral tentu tersebut.

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
```

maxima: 'integrate(f(x),x,0,1) = 0.01*sum(fx[i],i,1,length(fx))

Apakah hasil tersebut cukup baik? perhatikan gambarnya.

```
>function f(x) &= sin(3*x^5+7)^2
```

```
                         2     5
                      sin  (3 x  + 7)
```

```
>integrate(f,0,1)
```

```
0.542581176074
```

```
>&showev('integrate(f(x),x,0,1))
```

```
        1                           1              pi
        /                     gamma(-) sin(14) sin(--)
        [      2    5               5              10
        I   sin  (3 x  + 7) dx = ------------------------
        ]                                 1/5
        /                               10 6
        0
       4/5                 1           4/5                1
 - (((6    gamma_incomplete(-, 6 I) + 6    gamma_incomplete(-, - 6 I))
                           5                              5
            4/5                    1
 sin(14) + (6    I gamma_incomplete(-, 6 I)
                                   5
    4/5                    1                       pi
 - 6    I gamma_incomplete(-, - 6 I)) cos(14)) sin(--) - 60)/120
                           5                       10
```

```
>&float(%)
```

```
        1.0
        /
        [        2       5
        I     sin  (3.0 x  + 7.0) dx =
        ]
```

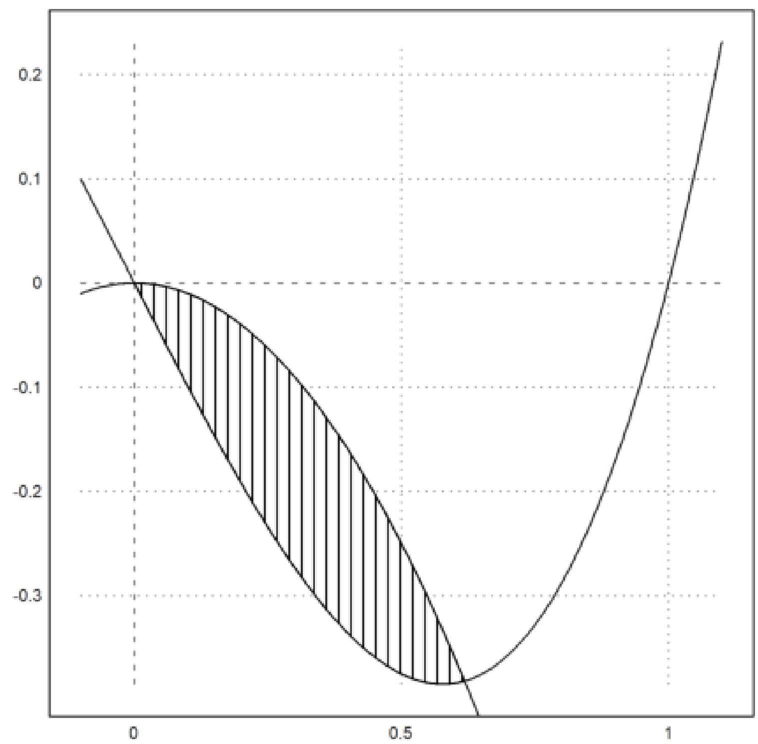
```
        /
         0.0
0.09820784258795788 - 0.008333333333333333
 (0.3090169943749474 (0.1367372182078336
 (4.192962712629476 I gamma__incomplete(0.2, 6.0 I)
 - 4.192962712629476 I gamma__incomplete(0.2, - 6.0 I))
 + 0.9906073556948704 (4.192962712629476 gamma__incomplete(0.2, 6.0 I)
 + 4.192962712629476 gamma__incomplete(0.2, - 6.0 I))) - 60.0)
```

```
>$showev('integrate(x*exp(-x),x,0,1)) // Integral tentu (eksak)
```

$$\int_{0}^{1}{x\,e^ {- x }\;dx}=1-2\,e^ {- 1 }$$

Aplikasi Integral Tentu

```
>plot2d("x^3-x",-0.1,1.1); plot2d("-x^2",>add);  ...
>b=solve("x^3-x+x^2",0.5); x=linspace(0,b,200); xi=flipx(x); ...
>plot2d(x|xi,x^3-x|-xi^2,>filled,style="|",fillcolor=1,>add): // Plot daera
```



```
>a=solve("x^3-x+x^2",0), b=solve("x^3-x+x^2",1) // absis titik-titik potong
```

```
0
0.61803398875
```

```
>integrate("(-x^2)-(x^3-x)",a,b) // luas daerah yang diarsir
```

```
0.0758191713542
```

Hasil tersebut akan kita bandingkan dengan perhitungan secara analitik.

```
>a &= solve((-x^2)-(x^3-x),x); $a // menentukan absis titik potong kedua ku
```

$$\left[x=\frac{-\sqrt{5}-1}{2},x=\frac{\sqrt{5}-1}{2},x=0\right]$$

```
>$showev('integrate(-x^2-x^3+x,x,0,(sqrt(5)-1)/2)) // Nilai integral secara
```

$$\int_{0}^{\frac{\sqrt{5}-1}{2}}{-x^3-x^2+x\;dx}=\frac{13-5^{\frac{3}{2}}}{24}$$

```
>$float(%)
```

$$\int_{0.0}^{0.6180339887498949}{-1.0\,x^3-1.0\,x^2+x\;dx}=0.07581917135421037$$

Panjang Kurva

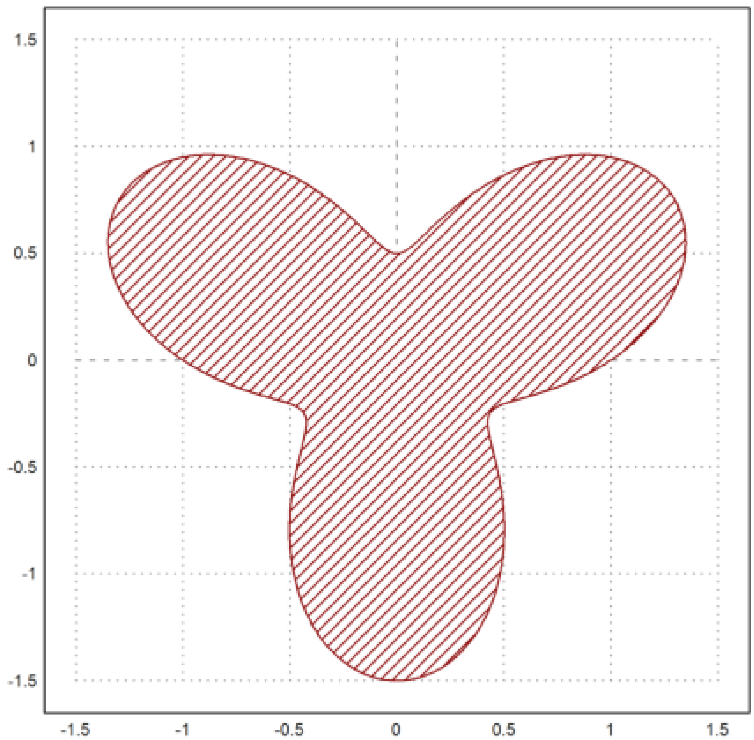
Hitunglah panjang kurva berikut ini dan luas daerah di dalam kurva tersebut.

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

dengan

$$r(t)=1+\frac{\sin(3t)}{2},\quad 0\le t\le 2\pi.$$

```
>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): // Kita gambar kurvanya
```




```
>function r(t) &= 1+sin(3*t)/2; $'r(t)=r(t)
```

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

```
>function fx(t) &= r(t)*cos(t); $'fx(t)=fx(t)
```

$$fx\left(t\right)=\cos t\,\left(\frac{\sin\left(3\,t\right)}{2}+1\right)$$

```
>function fy(t) &= r(t)*sin(t); $'fy(t)=fy(t)
```

$$fy\left(t\right)=\sin t\,\left(\frac{\sin\left(3\,t\right)}{2}+1\right)$$

```
>function ds(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2)))
```

$$ds\left(t\right)=\frac{\sqrt{4\,\cos \left(6\,t\right)+4\,\sin \left(3\,t\right)+9}}{2}$$

```
>$integrate(ds(x),x,0,2*pi) //panjang (keliling) kurva
```

$$\frac{\int_{0}^{2\,\pi}{\sqrt{4\,\cos \left(6\,x\right)+4\,\sin \left(3\,x\right)+9}\;dx}}{2}$$

Maxima gagal melakukan perhitungan eksak integral tersebut.

Berikut kita hitung integralnya secara umerik dengan perintah EMT.

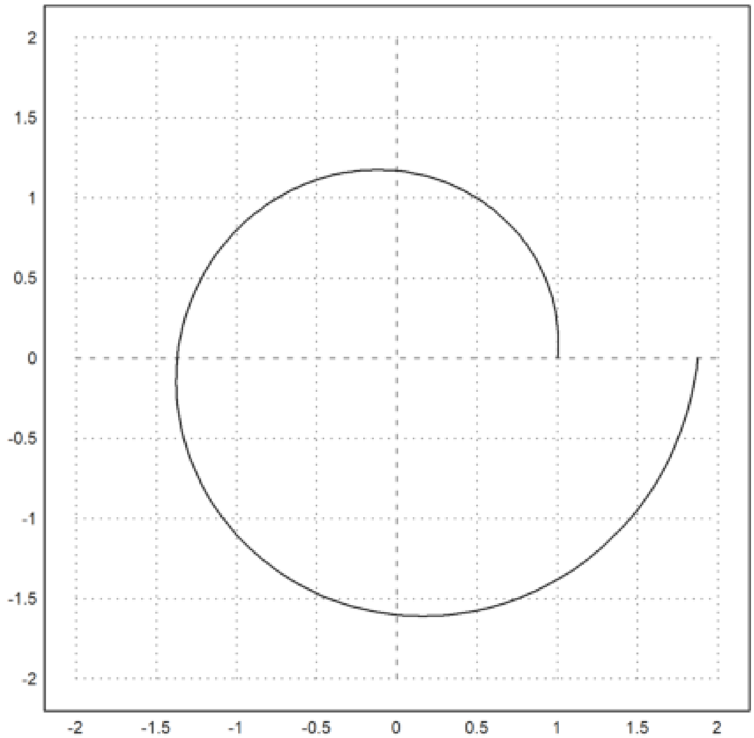
```
>integrate("ds(x)",0,2*pi)
```

```
9.0749467823
```

Spiral Logaritmik

$$x=e^{ax}\cos x,\;y=e^{ax}\sin x.$$

```
>a=0.1; plot2d("exp(a*x)*cos(x)","exp(a*x)*sin(x)",r=2,xmin=0,xmax=2*pi):
```




```
>&kill(a) // hapus expresi a
```

done

```
>function fx(t) &= exp(a*t)*cos(t); $'fx(t)=fx(t)
```

$$fx\left(t\right)=e^{a\,t}\,\cos t$$

```
>function fy(t) &= exp(a*t)*sin(t); $'fy(t)=fy(t)
```

$$fy\left(t\right)=e^{a\,t}\,\sin t$$

```
>function df(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2)))
```

$$df\left(t\right)=\sqrt{a^2+1}\,e^{a\,t}$$

```
>S &=integrate(df(t),t,0,2*%pi); $S // panjang kurva (spiral)
```

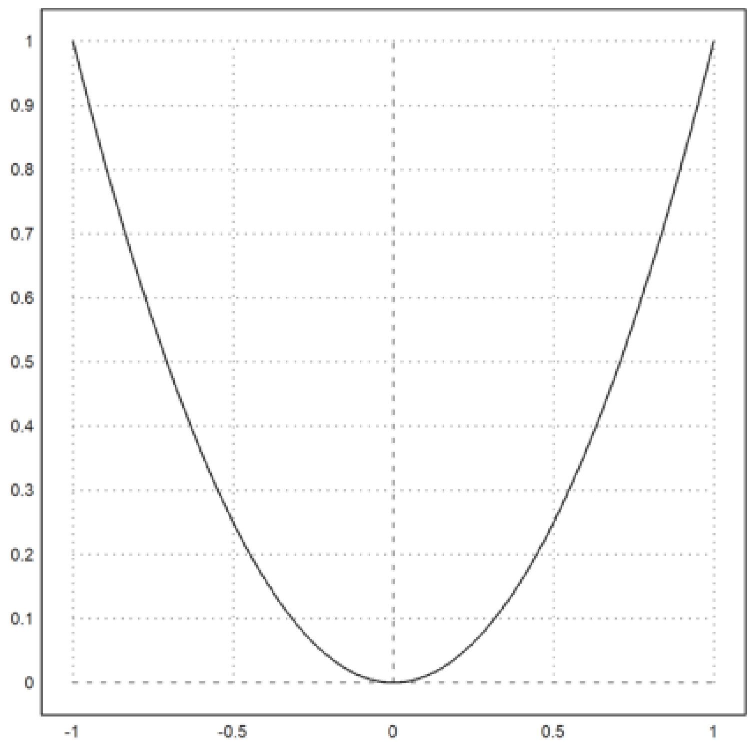
$$\sqrt{a^2+1}\,\left(\frac{e^{2\,\pi\,a}}{a}-\frac{1}{a}\right)$$

```
>S(a=0.1) // Panjang kurva untuk a=0.1
```

```
8.78817491636
```

Berikut adalah contoh menghitung panjang parabola.

```
>plot2d("x^2",xmin=-1,xmax=1):
```



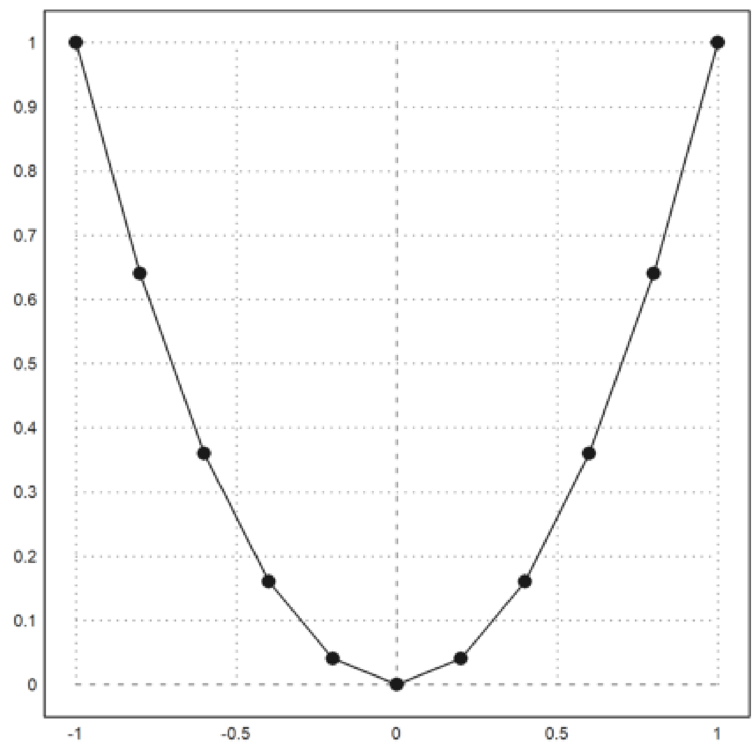

```
>$showev('integrate(sqrt(1+diff(x^2,x)^2),x,-1,1))
```

$$\int_{-1}^{1}{\sqrt{4\,x^2+1}\;dx}=\frac{\operatorname{asinh}2+2\,\sqrt{5}}{2}$$

```
>$float(%)
```

$$\int_{-1.0}^{1.0}\sqrt{4.0\,x^2+1.0}\,dx=2.957885715089195$$

```
>x=-1:0.2:1; y=x^2; plot2d(x,y);  ...
>  plot2d(x,y,points=1,style="o#",add=1):
```



Panjang tersebut dapat dihampiri dengan menggunakan jumlah panjang ruas-ruas garis yang menghubungkan titik-titik pada parabola tersebut.

```
>i=1:cols(x)-1; sum(sqrt((x[i+1]-x[i])^2+(y[i+1]-y[i])^2))
```

```
2.95191957027
```

Hasilnya mendekati panjang yang dihitung secara eksak. Untuk mendapatkan hampiran yang cukup akurat, jarak antar titik dapat diperkecil, misalnya 0.1, 0.05, 0.01, dan seterusnya. Cobalah Anda ulangi perhitungannya dengan nilai-nilai tersebut.

Koordinat Kartesius

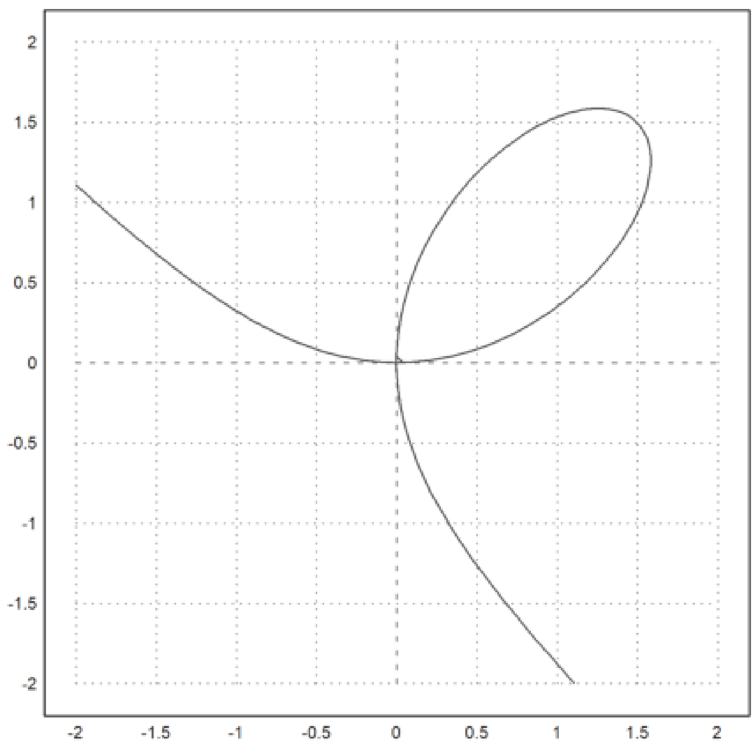
Berikut diberikan contoh perhitungan panjang kurva menggunakan koordinat Kartesius. Kita akan hitung panjang kurva dengan persamaan implisit:

$$x^3 + y^3 - 3xy = 0.$$

```
>z &= x^3+y^3-3*x*y; $z
```

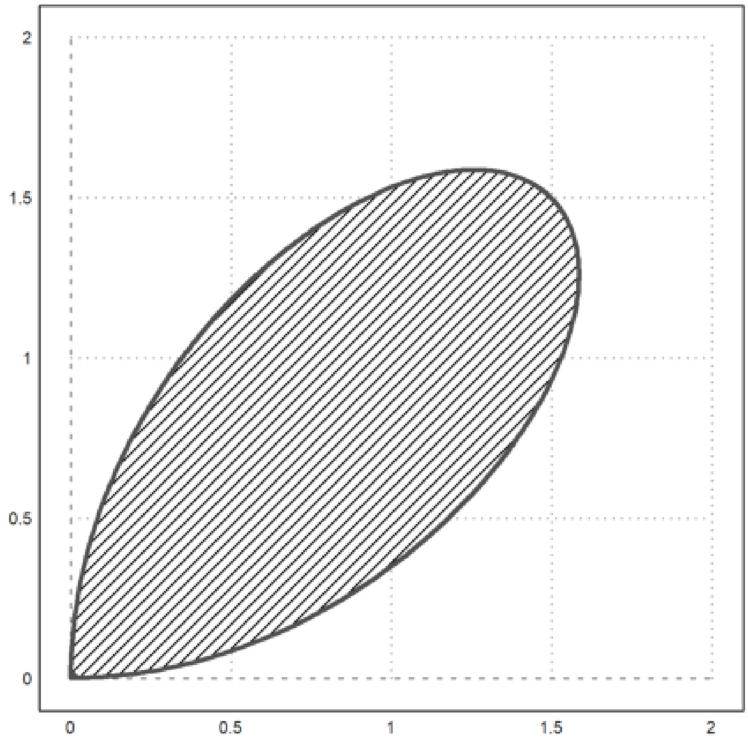
$$y^3 - 3\,x\,y + x^3$$

```
>plot2d(z,r=2,level=0,n=100):
```



Kita tertarik pada kurva di kuadran pertama.

```
>plot2d(z,a=0,b=2,c=0,d=2,level=[-10;0],n=100,contourwidth=3,style="/"):
```



Kita selesaikan persamaannya untuk x.

```
>$z with y=l*x, sol &= solve(%,x); $sol
```

$$\left[x=\frac{3\,l}{l^3+1} , x=0\right]$$

$$\left[x=\frac{3\,l}{l^3+1} , x=0\right]$$

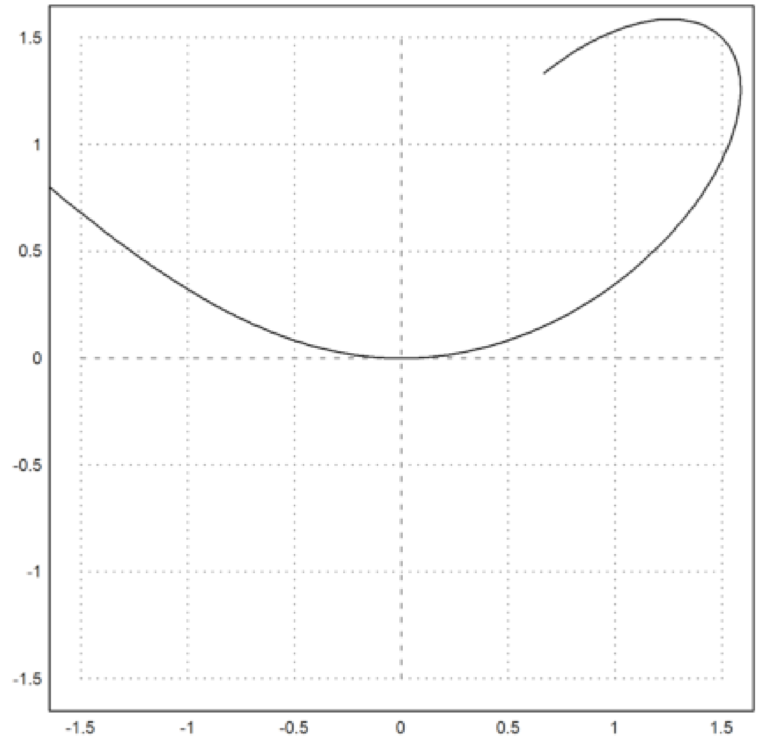
Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

```
>function f(l) &= rhs(sol[1]); $'f(l)=f(l)
```

$$f\left(l\right)=\frac{3\,l}{l^3+1}$$

Fungsi tersebut juga dapat digunaka untuk menggambar kurvanya. Ingat, bahwa fungsi tersebut adalah nilai x dan nilai y=l*x, yakni x=f(l) dan y=l*f(l).

```
>plot2d(&f(x),&x*f(x),xmin=-0.5,xmax=2,a=0,b=2,c=0,d=2,r=1.5):
```



Elemen panjang kurva adalah:

$$ds = \sqrt{f'(l)^2 + (lf'(l) + f(l))^2}.$$

```
>function ds(l) &= ratsimp(sqrt(diff(f(l),l)^2+diff(l*f(l),l)^2)); $'ds(l)=
```

$$ds\left(l\right)=\frac{\sqrt{9\,l^8+36\,l^6-36\,l^5-36\,l^3+36\,l^2+9}}{\sqrt{l^{12}+4\,l^9+6\,l^6+4\,l^3+1}}$$

```
>$integrate(ds(l),l,0,1)
```

$$\int_{0}^{1}{\frac{\sqrt{9\,l^8+36\,l^6-36\,l^5-36\,l^3+36\,l^2+9}}{\sqrt{l^{12}+4\,l^9+6\,l^6+4\,l^3+1}}\;dl}$$

Integral tersebut tidak dapat dihitung secara eksak menggunakan Maxima. Kita hitung integral etrsebut secara numerik dengan Euler. Karena kurva simetris, kita hitung untuk nilai variabel integrasi dari 0 sampai 1, kemudian hasilnya dikalikan 2.

```
>2*integrate("ds(x)",0,1)
```

```
4.91748872168
```

```
>2*romberg(&ds(x),0,1)// perintah Euler lain untuk menghitung nilai hampira
```

```
4.91748872168
```

Perhitungan di datas dapat dilakukan untuk sebarang fungsi x dan y dengan mendefinisikan fungsi EMT, misalnya kita beri nama panjangkurva. Fungsi ini selalu memanggil Maxima untuk menurunkan fungsi yang diberikan.

```
>function panjangkurva(fx,fy,a,b) ...
```

```
ds=mxm("sqrt(diff(@fx,x)^2+diff(@fy,x)^2)");
return romberg(ds,a,b);
endfunction
```

```
>panjangkurva("x","x^2",-1,1) // cek untuk menghitung panjang kurva parabol
```

```
2.95788571509
```

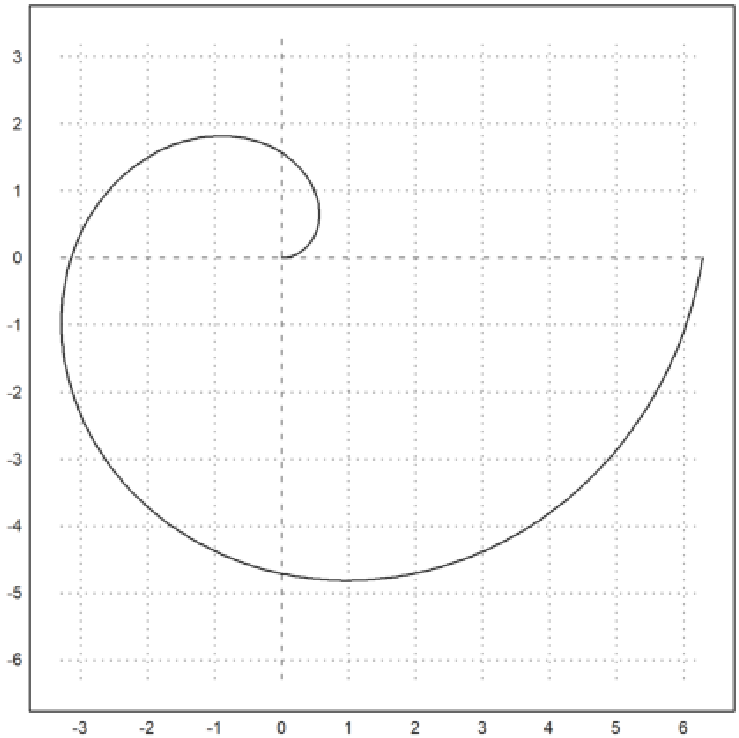
Bandingkan dengan nilai eksak di atas.

```
>2*panjangkurva(mxm("f(x)"),mxm("x*f(x)"),0,1) // cek contoh terakhir, band
```

```
4.91748872168
```

Kita hitung panjang spiral Archimides berikut ini dengan fungsi tersebut.

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




```
>panjangkurva("x*cos(x)","x*sin(x)",0,2*pi)
```

```
21.2562941482
```

Berikut kita definisikan fungsi yang sama namun dengan Maxima, untuk perhitungan eksak.

```
>&kill(ds,x,fx,fy)
```

```
done
```

```
>function ds(fx,fy) &&= sqrt(diff(fx,x)^2+diff(fy,x)^2)
```

```
                    2                2
sqrt(diff  (fy, x) + diff  (fx, x))
```

```
>sol &= ds(x*cos(x),x*sin(x)); $sol // Kita gunakan untuk menghitung panjan
```

$$\sqrt{\left(\cos x-x\,\sin x\right)^2+\left(\sin x+x\,\cos x\right)^2}$$

```
>$sol | trigreduce | expand, $integrate(%,x,0,2*pi), %()
```

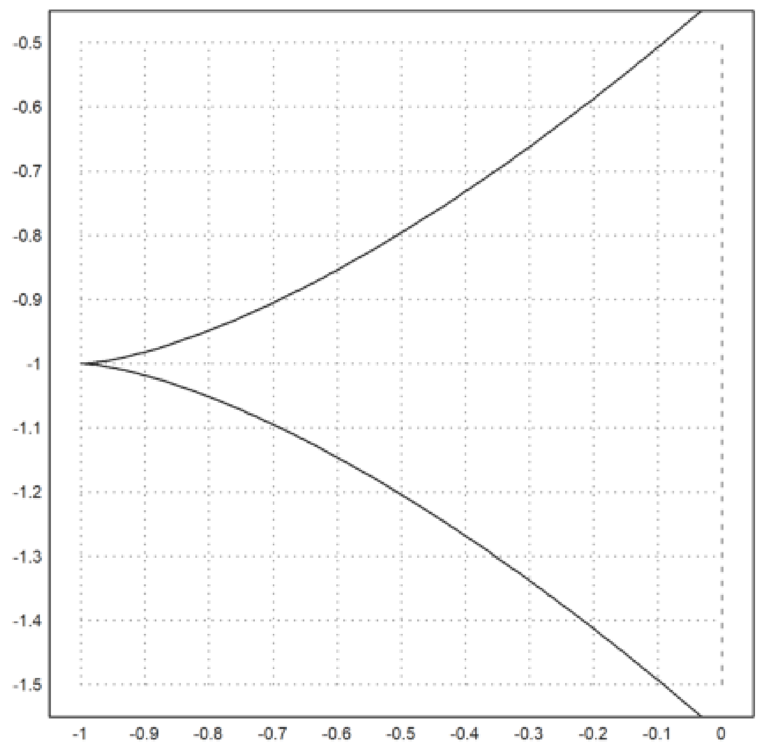
$$\frac{\mathrm{asinh}\;\left(2\,\pi\right)+2\,\pi\,\sqrt{4\,\pi^2+1}}{2}$$

$$\frac{\mathrm{asinh}\;\left(2\,\pi\right)+2\,\pi\,\sqrt{4\,\pi^2+1}}{2}$$

```
21.2562941482
```

Hasilnya sama dengan perhitungan menggunakan fungsi EMT.
Berikut adalah contoh lain penggunaan fungsi Maxima tersebut.

```
>plot2d("3*x^2-1","3*x^3-1",xmin=-1/sqrt(3),xmax=1/sqrt(3),square=1):
```



```
>sol &= radcan(ds(3*x^2-1,3*x^3-1)); $sol
```

$$3\,x\,\sqrt{9\,x^2+4}$$

```
>$showev('integrate(sol,x,0,1/sqrt(3))), $2*float(%) // panjang kurva di at
```

$$6.0\int_{0.0}^{0.5773502691896258}{x\,\sqrt{9.0\,x^2+4.0}\;dx}=2.337835372767141$$

$$6.0\int_{0.0}^{0.5773502691896258}{x\,\sqrt{9.0\,x^2+4.0}\;dx}=2.337835372767141$$

Sikloid

Berikut kita akan menghitung panjang kurva lintasan (sikloid) suatu titik pada lingkaran yang berputar ke kanan pada permukaan datar. Misalkan jari-jari lingkaran tersebut adalah r. Posisi titik pusat lingkaran pada saat t adalah:

$$(rt, r).$$

Misalkan posisi titik pada lingkaran tersebut mula-mula (0,0) dan posisinya pada saat t adalah:

$$(r(t-\sin(t)), r(1-\cos(t))).$$

Berikut kita plot lintasan tersebut dan beberapa posisi lingkaran ketika t=0, t=pi/2, t=r*pi.

```
>x &= r*(t-sin(t))
```

```
                           r (t - sin(t))
```

```
>y &= r*(1-cos(t))
```

```
                           r (1 - cos(t))
```

Berikut kita gambar sikloid untuk r=1.

```
>ex &= x-sin(x); ey &= 1-cos(x); aspect(1);
>plot2d(ex,ey,xmin=0,xmax=4pi,square=1); ...
>  plot2d("2+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ...
>  plot2d([2,ex(2)],[1,ey(2)],color=red,>add); ...
>  plot2d(ex(2),ey(2),>points,>add,color=red); ...
>  plot2d("2pi+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ...
>  plot2d([2pi,ex(2pi)],[1,ey(2pi)],color=red,>add);   ...
>  plot2d(ex(2pi),ey(2pi),>points,>add,color=red):
```

```
Variable or function t not found.
Error in expression: r*(t-sin(t))-sin(r*(t-sin(t)))
adaptiveeval:
    sx=f$(t;args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    dw/n,dw/n^2,dw/n;args());
```

Berikut dihitung panjang lintasan untuk 1 putaran penuh. (Jangan salah menduga bahwa panjang lintasan 1 putaran penuh sama dengan keliling lingkaran!)

```
>ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds) // elemen
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds ...
                                             ^
```

```
>ds &= trigsimp(ds); $ds
>$showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid satu putaran pe
>integrate(mxm("ds"),0,2*pi) // hitung secara numerik
```

```
8
```

```
>romberg(mxm("ds"),0,2*pi) // cara lain hitung secara numerik
```

```
8
```

Perhatikan, seperti terlihat pada gambar, panjang sikloid lebih besar daripada keliling lingkarannya, yakni:

$$2\pi.$$

Kurvatur (Kelengkungan) Kurva

image: Osculating.png

Aslinya, kelengkungan kurva diferensiabel (yakni, kurva mulus yang tidak lancip) di titik P didefinisikan melalui lingkaran oskulasi (yaitu, lingkaran yang melalui titik P dan terbaik memperkirakan, paling banyak menyinggung kurva di sekitar P). Pusat dan radius kelengkungan kurva di P adalah pusat dan radius lingkaran oskulasi. Kelengkungan adalah kebalikan dari radius kelengkungan:

$$\kappa = \frac{1}{R}$$

dengan R adalah radius kelengkungan. (Setiap lingkaran memiliki kelengkungan ini pada setiap titiknya, dapat diartikan, setiap lingkaran berputar 2pi sejauh 2piR.)

Definisi ini sulit dimanipulasi dan dinyatakan ke dalam rumus untuk kurva umum. Oleh karena itu digunakan definisi lain yang ekivalen.

Definisi Kurvatur dengan Fungsi Parametrik Panjang Kurva

Setiap kurva diferensiabel dapat dinyatakan dengan persamaan parametrik terhadap panjang kurva s:

$$\gamma(s) = (x(s),\ y(s)),$$

dengan x dan y adalah fungsi riil yang diferensiabel, yang memenuhi:

$$\|\gamma'(s)\| = \sqrt{x'(s)^2 + y'(s)^2} = 1.$$

Ini berarti bahwa vektor singgung

$$\mathbf{T}(s) = (x'(s),\ y'(s))$$

memiliki norm 1 dan merupakan vektor singgung satuan.

Apabila kurvanya memiliki turunan kedua, artinya turunan kedua x dan y ada, maka T′(s) ada. Vektor ini merupakan normal kurva yang arahnya menuju pusat kurvatur, norm-nya merupakan nilai kurvatur (kelengkungan):

$$\begin{aligned} \mathbf{T}(s) &= \gamma'(s), \\ \mathbf{T}^2(s) &= 1 \text{ (konstanta)} \Rightarrow \mathbf{T}'(s) \cdot \mathbf{T}(s) = 0 \\ \kappa(s) &= \|\mathbf{T}'(s)\| = \|\gamma''(s)\| = \sqrt{x''(s)^2 + y''(s)^2}. \end{aligned}$$

Nilai

$$R(s) = \frac{1}{\kappa(s)}$$

disebut jari-jari (radius) kelengkungan kurva.

Bilangan riil

$$k(s) = \pm\kappa(s)$$

disebut nilai kelengkungan bertanda.

Contoh:

Akan ditentukan kurvatur lingkaran

$$x = r\cos t, \; y = r\sin t.$$

```
>fx &= r*cos(t); fy &=r*sin(t);
>&assume(t>0,r>0); s &=integrate(sqrt(diff(fx,t)^2+diff(fy,t)^2),t,0,t); s
```

```
r t
```

```
>&kill(s); fx &= r*cos(s/r); fy &=r*sin(s/r); // definisi ulang persamaan p
>k &= trigsimp(sqrt(diff(fx,s,2)^2+diff(fy,s,2)^2)); $k // nilai kurvatur l
```

$$\frac{1}{r}$$

Untuk representasi parametrik umum, misalkan

$$x = x(t), \; y = y(t)$$

merupakan persamaan parametrik untuk kurva bidang yang terdiferensialkan dua kali. Kurvatur untuk kurva tersebut didefinisikan sebagai

$$\begin{aligned} \kappa &= \frac{d\phi}{ds} = \frac{\frac{d\phi}{dt}}{\frac{ds}{dt}} \quad (\phi \text{ adalah sudut kemiringan garis singgung dan } s \text{ adalah panjang kurva}) \\ &= \frac{\frac{d\phi}{dt}}{\sqrt{(\frac{dx}{dt})^2 + (\frac{dy}{dt})^2}} = \frac{\frac{d\phi}{dt}}{\sqrt{x'(t)^2 + y'(t)^2}}. \end{aligned}$$

Selanjutnya, pembilang pada persamaan di atas dapat dicari sebagai berikut.

$$\sec^2\phi\frac{d\phi}{dt} = \frac{d}{dt}(\tan\phi) = \frac{d}{dt}\left(\frac{dy}{dx}\right) = \frac{d}{dt}\left(\frac{dy/dt}{dx/dt}\right) = \frac{d}{dt}\left(\frac{y'(t)}{x'(t)}\right) = \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2}.$$

$$\begin{aligned}\frac{d\phi}{dt} &= \frac{1}{\sec^2\phi}\frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2}\\ &= \frac{1}{1+\tan^2\phi}\frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2}\\ &= \frac{1}{1+\left(\frac{y'(t)}{x'(t)}\right)^2}\frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2}\\ &= \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2 + y'(t)^2}.\end{aligned}$$

Jadi, rumus kurvatur untuk kurva parametrik

$$x = x(t),\ y = y(t)$$

adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{\left(x'(t)^2 + y'(t)^2\right)^{3/2}}.$$

Jika kurvanya dinyatakan dengan persamaan parametrik pada koordinat kutub

$$x = r(\theta)\cos\theta,\ y = r(\theta)\sin\theta,$$

maka rumus kurvaturnya adalah

$$\kappa(\theta) = \frac{r(\theta)^2 + 2r'(\theta)^2 - r(\theta)r''(\theta)}{\left(r'(\theta)^2 + r'(\theta)^2\right)^{3/2}}.$$

(Silakan Anda turunkan rumus tersebut!)

Contoh:

Lingkaran dengan pusat (0,0) dan jari-jari r dapat dinyatakan dengan persamaan parametrik

$$x = r\cos t,\ y = r\sin t.$$

Nilai kelengkungan lingkaran tersebut adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{\left(x'(t)^2 + y'(t)^2\right)^{3/2}} = \frac{r^2}{r^3} = \frac{1}{r}.$$

Hasil cocok dengan definisi kurvatur suatu kelengkungan.

Kurva

$$y = f(x)$$

dapat dinyatakan ke dalam persamaan parametrik

$$x = t,\ y = f(t),\ \text{ dengan } x'(t) = 1,\ x''(t) = 0,$$

sehingga kurvaturnya adalah

$$\kappa(t) = \frac{y''(t)}{(1 + y'(t)^2)^{3/2}}.$$

Contoh:
Akan ditentukan kurvatur parabola

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

```
>function f(x) &= a*x^2+b*x+c; $y=f(x)
```

$$y = a\,x^2 + b\,x + c$$

```
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) //
```

$$k\left(x\right) = \frac{2\,a}{\left(\left(2\,a\,x + b\right)^2 + 1\right)^{\frac{3}{2}}}$$

```
>function f(x) &= x^2+x+1; $y=f(x) // akan kita plot kelengkungan parabola
```

$$y = x^2 + x + 1$$

```
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) //
```

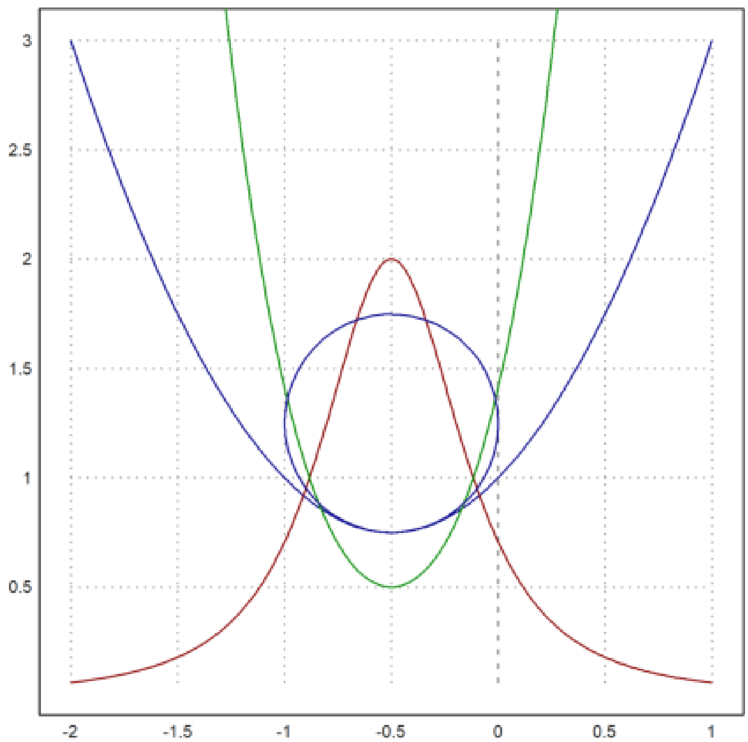
$$k\left(x\right) = \frac{2}{\left(\left(2\,x + 1\right)^2 + 1\right)^{\frac{3}{2}}}$$

Berikut kita gambar parabola tersebut beserta kurva kelengkungan, kurva jari-jari kelengkungan dan salah satu lingkaran oskulasi di titik puncak parabola. Perhatikan, puncak parabola dan jari-jari lingkaran oskulasi di puncak parabola adalah

$$(-1/2, 3/4),\ 1/k(2) = 1/2,$$

sehingga pusat lingkaran oskulasi adalah (-1/2, 5/4).

```
>plot2d(["f(x)", "k(x)"],-2,1, color=[blue,red]); plot2d("1/k(x)",-1.5,1,co
>plot2d("-1/2+1/k(-1/2)*cos(x)","5/4+1/k(-1/2)*sin(x)",xmin=0,xmax=2pi,>add
```



Untuk kurva yang dinyatakan dengan fungsi implisit

$$F(x, y) = 0$$

dengan turunan-turunan parsial

$$F_x = \frac{\partial F}{\partial x},\ F_y = \frac{\partial F}{\partial y},\ F_{xy} = \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial x}\right),\ F_{xx} = \frac{\partial}{\partial x}\left(\frac{\partial F}{\partial x}\right),\ F_{yy} = \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial y}\right),$$

berlaku

$$F_x dx + F_y dy = 0 \text{ atau } \frac{dy}{dx} = -\frac{F_x}{F_y},$$

sehingga kurvaturnya adalah

$$\kappa = \frac{F_y^2 F_{xx} - 2F_x F_y F_{xy} + F_x^2 F_{yy}}{\left(F_x^2 + F_y^2\right)^{3/2}}.$$

(Silakan Anda turunkan sendiri!)

Contoh 1:
Parabola

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

dapat dinyatakan ke dalam persamaan implisit

$$ax^2 + bx + c - y = 0.$$

```
>function F(x,y) &=a*x^2+b*x+c-y; $F(x,y)
```

$$-y + a\,x^2 + b\,x + c$$

```
>Fx &= diff(F(x,y),x), Fxx &=diff(F(x,y),x,2), Fy &=diff(F(x,y),y), Fxy &=d
```

```
2 a x + b

2 a

- 1

0

0
```

```
>function k(x) &= (Fy^2*Fxx-2*Fx*Fy*Fxy+Fx^2*Fyy)/(Fx^2+Fy^2)^(3/2); $'k(x)
```

$$k\left(x\right) = \frac{2\,a}{\left(\left(2\,a\,x + b\right)^2 + 1\right)^{\frac{3}{2}}}$$

Hasilnya sama dengan sebelumnya yang menggunakan persamaan parabola biasa.

Latihan

- Bukalah buku Kalkulus.
- Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi).
- Untuk setiap fungsi, tentukan anti turunannya (jika ada), hitunglah integral tentu dengan batas-batas yang menarik (Anda tentukan sendiri), seperti contoh-contoh tersebut.
- Lakukan hal yang sama untuk fungsi-fungsi yang tidak dapat diintegralkan (cari sedikitnya 3 fungsi).
- Gambar grafik fungsi dan daerah integrasinya pada sumbu koordinat yang sama.
- Gunakan integral tentu untuk mencari luas daerah yang dibatasi oleh dua kurva yang berpotongan di dua titik. (Cari dan gambar kedua kurva dan arsir (warnai) daerah yang dibatasi oleh keduanya.)
- Gunakan integral tentu untuk menghitung volume benda putar kurva y= f(x) yang diputar mengelilingi sumbu x dari x=a sampai x=b, yakni

$$V = \int_a^b \pi (f(x)^2\, dx.$$

(Pilih fungsinya dan gambar kurva dan benda putar yang dihasilkan. Anda dapat mencari contoh-contoh bagaimana cara menggambar benda hasil perputaran suatu kurva.)
- Gunakan integral tentu untuk menghitung panjang kurva y=f(x) dari x=a sampai x=b dengan menggunakan rumus:

$$S = \int_a^b \sqrt{1 + (f'(x))^2}\, dx.$$

(Pilih fungsi dan gambar kurvanya.)
- Apabila fungsi dinyatakan dalam koordinat kutub x=f(r,t), y=g(r,t), r=h(t), x=a bersesuaian dengan t=t0 dan x=b bersesuaian dengan t=t1, maka rumus di atas akan menjadi:

$$S = \int_{t_0}^{t_1} \sqrt{x'(t)^2 + y'(t)^2}\, dt.$$

- Pilih beberapa kurva menarik (selain lingkaran dan parabola) dari buku kalkulus. Nyatakan setiap kurva tersebut dalam bentuk:

```
a. koordinat Kartesius (persamaan y=f(x))
b. koordinat kutub ( r=r(theta))
c. persamaan parametrik x=x(t), y=y(t)
d. persamaan implit F(x,y)=0
```

- Tentukan kurvatur masing-masing kurva dengan menggunakan keempat representasi tersebut (hasilnya harus sama).
- Gambarlah kurva asli, kurva kurvatur, kurva jari-jari lingkaran oskulasi, dan salah satu lingkaran oskulasinya.

Soal 1

$$f(x) = 18x^8 - 25x^4 + 3x^2$$

```
>function f(x) &= 18*x^8-25*x^4+3*x^2; $f(x)
```

$$18\,x^8-25\,x^4+3\,x^2$$

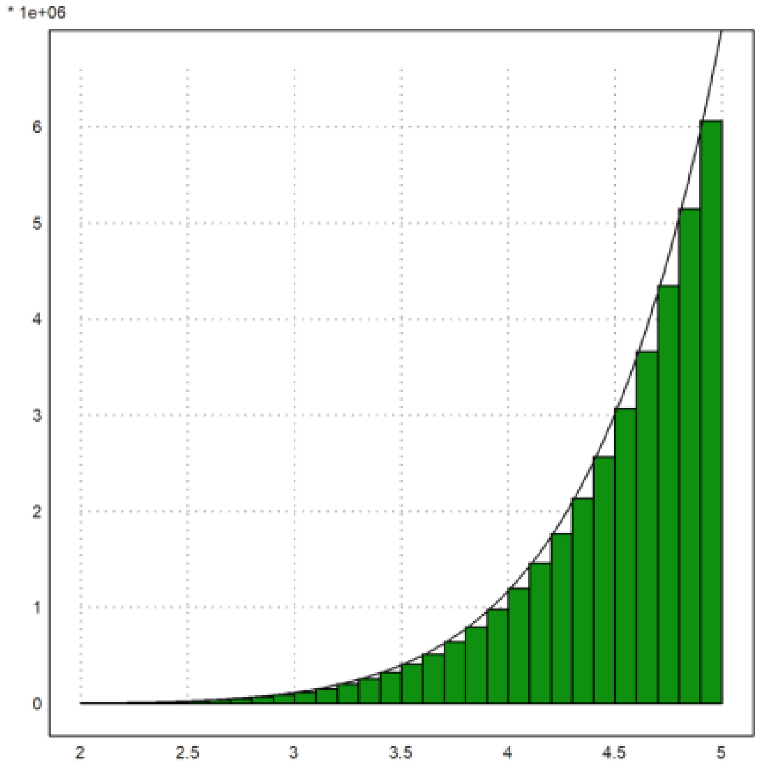
```
>$showev('integrate(18*x^8-25*x^4+3*x^2,x)) // integral tak tentu
```

$$\int {18\,x^8-25\,x^4+3\,x^2}{\;dx}=2\,x^9-5\,x^5+x^3$$

```
>$showev('integrate(18*x^8-25*x^4+3*x^2,x,2,5)) // integral tentu
```

$$\int_{2}^{5}{18\,x^8-25\,x^4+3\,x^2\;dx}=3889878$$

```
>x=2:0.1:5-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",2,5,>add):
```



Soal 2

$$g(x) = 3x^4(2x^5 + 9)^3$$

```
>function g(x) &= 3*x^4*(2*x^5+9)^3; $g(x)
```

$$3\,x^4\,\left(2\,x^5+9\right)^3$$

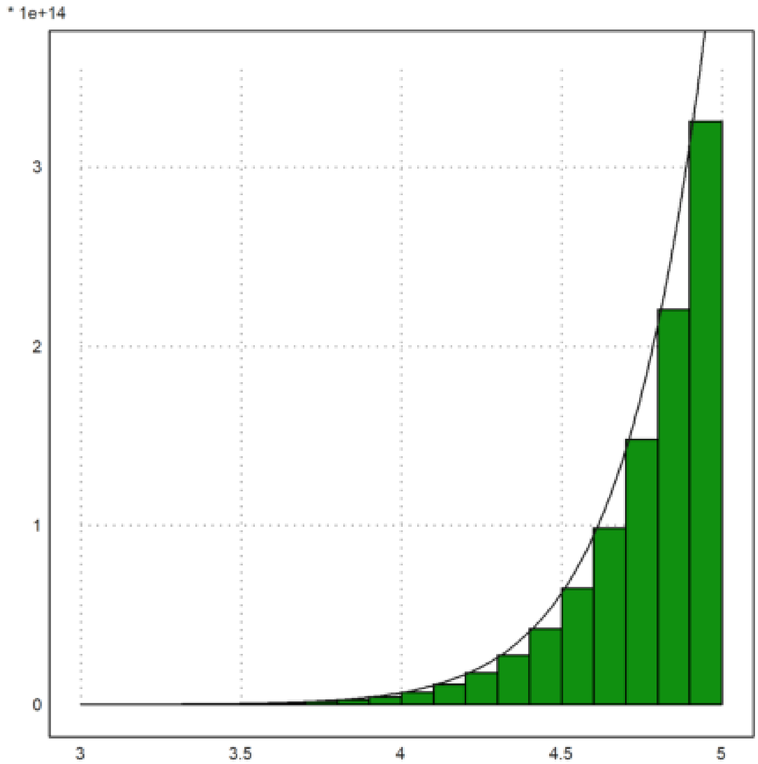
```
>$showev('integrate(g(x),x)) // integral tak tentu
```

$$3\,\int {x^4\,\left(2\,x^5+9\right)^3}{\;dx}=\frac{3\,\left(2\,x^5+9\right)^4}{40}$$

```
>$showev('integrate(g(x),x,3,5)) // integral tentu
```

$$3\,\int_{3}^{5}{x^4\,\left(2\,x^5+9\right)^3\;dx}=\frac{575485100289726}{5}$$

```
>x=3:0.1:5-0.01; plot2d(x,g(x+0.01),>bar); plot2d("g(x)",3,5,>add):
```




```
>function k(x) &= 2*x+sin(x); $k(x)
```

$$\sin x + 2\,x$$

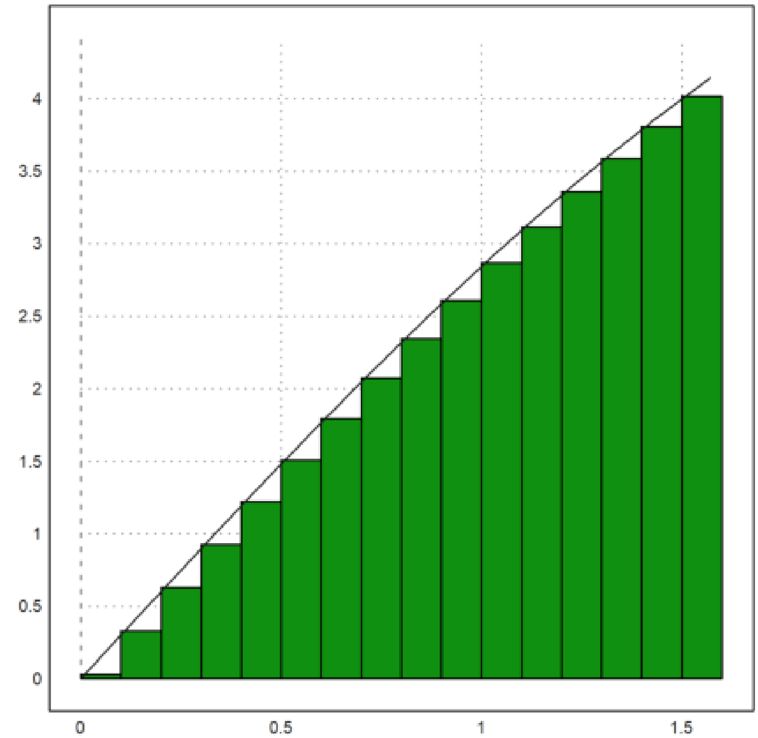
```
>$showev('integrate(k(x),x)) // integral tak tentu
```

$$\int \sin x + 2\,x\;dx = x^2 - \cos x$$

```
>$showev('integrate(k(x),x,0,pi/2)) // integral tentu
```

$$\int_0^{\frac{\pi}{2}} \sin x + 2\,x\;dx = \frac{\pi^2 + 4}{4}$$

```
>x=0:0.1:pi/2-0.01; plot2d(x,k(x+0.01),>bar); plot2d("k(x)",0,pi/2,>add):
```



Soal 4

$$l(x) = (sinx + cosx)^2$$

```
>function l(x) &= (sin(x)+cos(x))^2; $l(x)
```

$$(\sin x + \cos x)^2$$

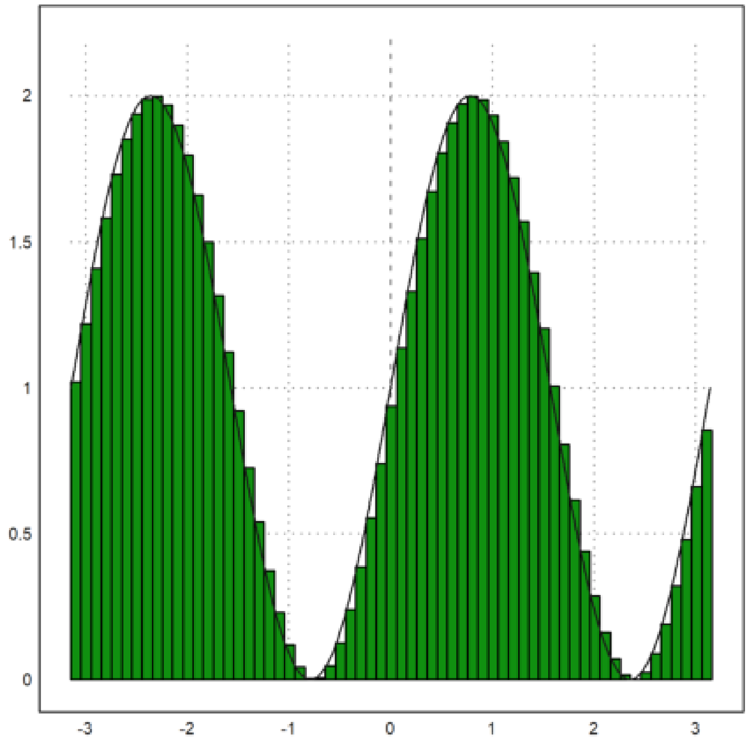
```
>$showev('integrate(l(x),x)) // integral tak tentu
```

$$\int (\sin x + \cos x)^2\, dx = \frac{\frac{\sin(2\,x)}{2} + x}{2} + \frac{x - \frac{\sin(2\,x)}{2}}{2} - \cos^2 x$$

```
>$showev('integrate(l(x),x,-pi,pi)) // integral tentu
```

$$\int_{-\pi}^{\pi}{\left(\sin x+\cos x\right)^2\;dx}=2\,\pi$$

```
>x=-pi:0.1:pi-0.01; plot2d(x,l(x+0.01),>bar); plot2d("l(x)",-pi,pi,>add):
```



Soal 5

$$\int \frac{x+2}{(x^2+4x+1)^2}\,dx \; dan \; \int_0^2 \frac{2+x}{x^2+4x+1}\,dx$$

```
>function m(x) &= (x+2)/((x^2+4*x+1)^2); $m(x)
```

$$\frac{x+2}{\left(x^2+4\,x+1\right)^2}$$

```
>$showev('integrate(m(x),x)) // integral tak tentu
```

$$\int \frac{x+2}{\left(x^2+4\,x+1\right)^2}\,dx=-\frac{1}{2\,\left(x^2+4\,x+1\right)}$$

```
>$showev('integrate(m(x),x,0,1)) // integral tentu
```

$$\int_{0}^{1}{\frac{x+2}{\left(x^2+4\,x+1\right)^2}\,dx}=\frac{5}{12}$$

Soal 6

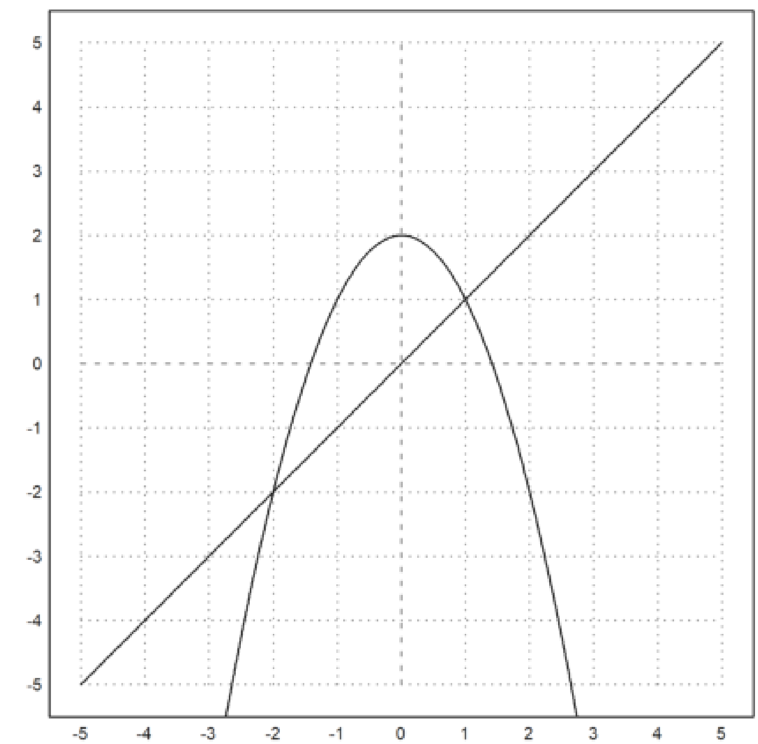
```
>function p(x) &= 2-x^2
```

```
                                     2
                                2 - x
```

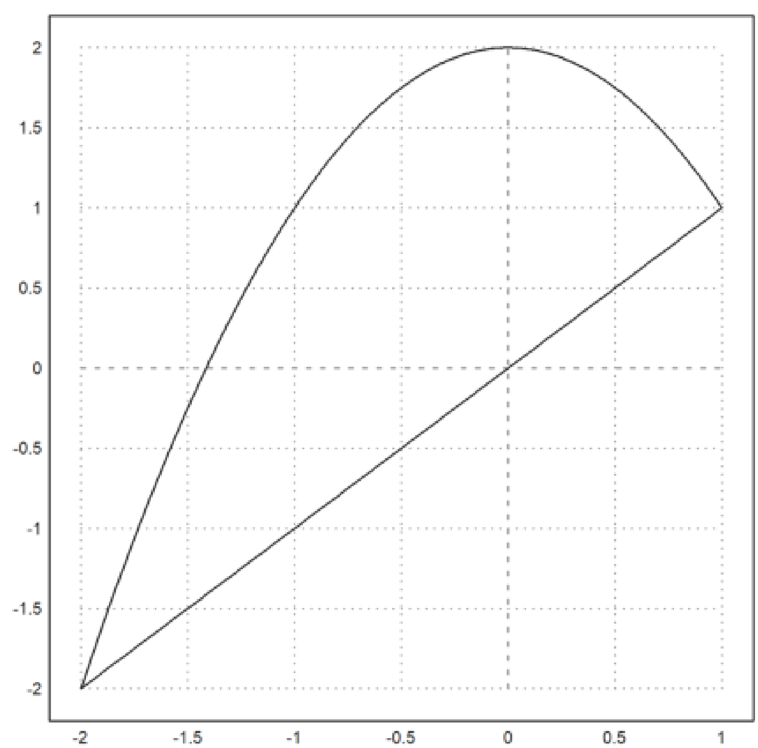
```
>function q(x) &= x
```

```
                                  x
```

```
>plot2d(["p(x)","q(x)"],-5,5,-5,5):
```



```
>plot2d(["p(x)","q(x)"],xmin=-2,xmax=1,>filled,style="/"):
```



```
>function r(x) &= p(x)-q(x)
```

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

```
>$showev('integrate(r(x),x,-2,1))
```

$$\int_{-2}^{1}{-x^2-x+2\;dx}=\frac{9}{2}$$

Jadi, luas daerah yang dibatasi dua kurva tersebut adalah 9/2.

Contoh fungsi yang tidak bisa diintegralkan

1.

$$\int \left|\frac{sin\ x}{x}\right|\,dx$$

```
>function a(x) &= abs(sin(x)/x); $a(x)
```

$$\frac{\left|\sin x\right|}{\left|x\right|}$$

```
>$showev('integrate(a(x),x)) // integral tak tentu
```

$$\int {\frac{\left|\sin x\right|}{\left|x\right|}}{\;dx}=\int {\frac{\left|\sin x\right|}{\left|x\right|}}{\;dx}$$

Barisan dan Deret

(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh pengguanaan EMT dan Maxima untuk menghitung limit barisan, rumus jumlah parsial suatu deret, jumlah tak hingga suatu deret konvergen, dan sebagainya. Anda dapat mengeksplor contoh-contoh di EMT atau perbagai panduan penggunaan Maxima di software Maxima atau dari Internet.)

Barisan dapat didefinisikan dengan beberapa cara di dalam EMT, di antaranya:

- dengan cara yang sama seperti mendefinisikan vektor dengan elemen-elemen beraturan (menggunakan titik dua ":");

- menggunakan perintah "sequence" dan rumus barisan (suku ke -n);
- menggunakan perintah "iterate" atau "niterate";
- menggunakan fungsi Maxima "create_list" atau "makelist" untuk menghasilkan barisan simbolik;
- menggunakan fungsi biasa yang inputnya vektor atau barisan;
- menggunakan fungsi rekursif.

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni:

- sum: menghitung jumlah semua elemen suatu barisan
- cumsum: jumlah kumulatif suatu barisan
- differences: selisih antar elemen-elemen berturutan

EMT juga dapat digunakan untuk menghitung jumlah deret berhingga maupun deret tak hingga, dengan menggunakan perintah (fungsi) "sum". Perhitungan dapat dilakukan secara numerik maupun simbolik dan eksak.

Berikut adalah beberapa contoh perhitungan barisan dan deret menggunakan EMT.

```
>1:10 // barisan sederhana
```

```
[1,  2,  3,  4,  5,  6,  7,  8,  9,  10]
```

```
>1:2:30
```

```
[1,  3,  5,  7,  9,  11,  13,  15,  17,  19,  21,  23,  25,  27,  29]
```

Iterasi dan Barisan

EMT menyediakan fungsi iterate("g(x)", x0, n) untuk melakukan iterasi

$$x_{k+1} = g(x_k),\ x_0 = x_0, k = 1, 2, 3, ..., n.$$

Berikut ini disajikan contoh-contoh penggunaan iterasi dan rekursi dengan EMT. Contoh pertama menunjukkan pertumbuhan dari nilai awal 1000 dengan laju pertambahan 5%, selama 10 periode.

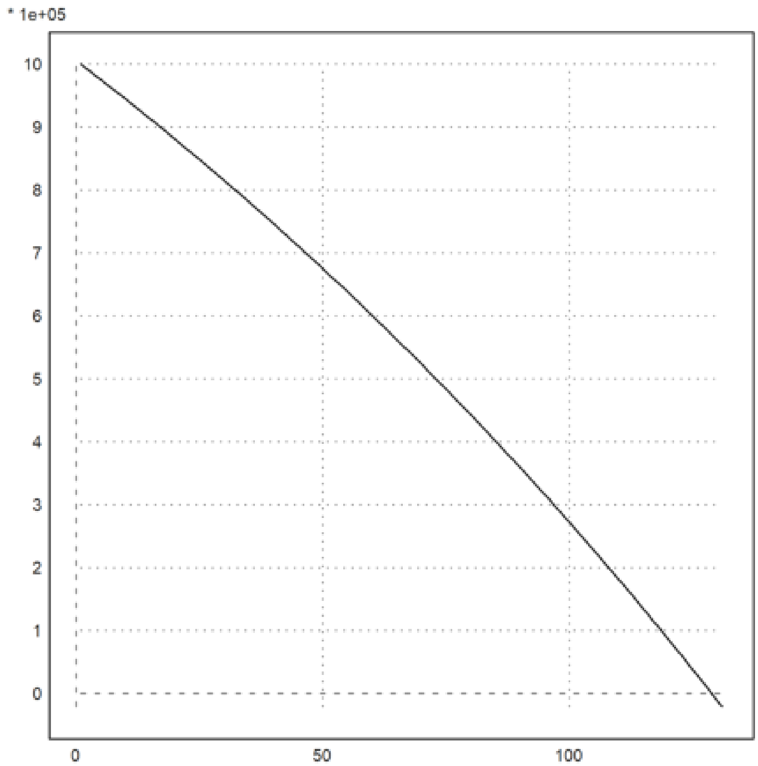
```
>q=1.05; iterate("x*q",1000,n=10)'
```

```
        1000
        1050
      1102.5
     1157.63
     1215.51
     1276.28
      1340.1
```

```
 1407.1
1477.46
1551.33
1628.89
```

Contoh berikutnya memperlihatkan bahaya menabung di bank pada masa sekarang! Dengan bunga tabungan sebesar 6% per tahun atau 0.5% per bulan dipotong pajak 20%, dan biaya administrasi 10000 per bulan, tabungan sebesar 1 juta tanpa diambil selama sekitar 10 tahunan akan habis diambil oleh bank!

```
>r=0.005; plot2d(iterate("(1+0.8*r)*x-10000",1000000,n=130)):
```



Silakan Anda coba-coba, dengan tabungan minimal berapa agar tidak akan habis diambil oleh bank dengan ketentuan bunga dan biaya administrasi seperti di atas.

Berikut adalah perhitungan minimal tabungan agar aman di bank dengan bunga sebesar r dan biaya administrasi a, pajak bunga 20%.

```
>$solve(0.8*r*A-a,A), $% with [r=0.005, a=10]
```

$$[A = 2500.0]$$

Berikut didefinisikan fungsi untuk menghitung saldo tabungan, kemudian dilakukan iterasi.

```
>function saldo(x,r,a) := round((1+0.8*r)*x-a,2);
>iterate({{"saldo",0.005,10}},1000,n=6)
```

```
[1000,  994,  987.98,  981.93,  975.86,  969.76,  963.64]
```

```
>iterate({{"saldo",0.005,10}},2000,n=6)
```

```
[2000,  1998,  1995.99,  1993.97,  1991.95,  1989.92,  1987.88]
```

```
>iterate({{"saldo",0.005,10}},2500,n=6)
```

```
[2500,  2500,  2500,  2500,  2500,  2500,  2500]
```

Tabungan senilai 2,5 juta akan aman dan tidak akan berubah nilai (jika tidak ada penarikan), sedangkan jika tabungan awal kurang dari 2,5 juta, lama kelamaan akan berkurang meskipun tidak pernah dilakukan penarikan uang tabungan.

```
>iterate({{"saldo",0.005,10}},3000,n=6)
```

```
[3000,  3002,  3004.01,  3006.03,  3008.05,  3010.08,  3012.12]
```

Tabungan yang lebih dari 2,5 juta baru akan bertambah jika tidak ada penarikan.

Untuk barisan yang lebih kompleks dapat digunakan fungsi "sequence()". Fungsi ini menghitung nilai-nilai x[n] dari semua nilai sebelumnya, x[1],...,x[n-1] yang diketahui.
Berikut adalah contoh barisan Fibonacci.

$$x_n = x_{n-1} + x_{n-2}, \quad x_1 = 1, \quad x_2 = 1$$

```
>sequence("x[n-1]+x[n-2]",[1,1],15)
```

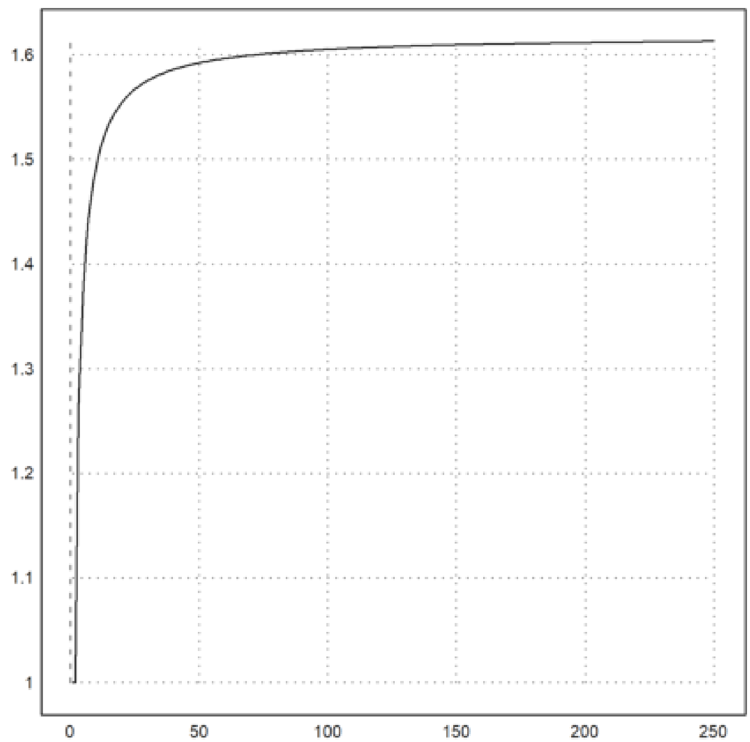
```
[1,  1,  2,  3,  5,  8,  13,  21,  34,  55,  89,  144,  233,  377,  610]
```

Barisan Fibonacci memiliki banyak sifat menarik, salah satunya adalah akar pangkat ke-n suku ke-n akan konvergen ke pecahan emas:

```
>$'(1+sqrt(5))/2=float((1+sqrt(5))/2)
```

$$\frac{\sqrt{5}+1}{2} = 1.618033988749895$$

```
>plot2d(sequence("x[n-1]+x[n-2]",[1,1],250)^(1/(1:250))):
```



Barisan yang sama juga dapat dihasilkan dengan menggunakan loop.

```
>x=ones(500); for k=3 to 500; x[k]=x[k-1]+x[k-2]; end;
```

Rekursi dapat dilakukan dengan menggunakan rumus yang tergantung pada semua elemen sebelumnya. Pada contoh berikut, elemen ke-n merupakan jumlah (n-1) elemen sebelumnya, dimulai dengan 1 (elemen ke-1). Jelas, nilai elemen ke-n adalah 2^(n-2), untuk n=2, 4, 5, ...

```
>sequence("sum(x)",1,10)
```

```
[1,  1,  2,  4,  8,  16,  32,  64,  128,  256]
```

Selain menggunakan ekspresi dalam x dan n, kita juga dapat menggunakan fungsi.
Pada contoh berikut, digunakan iterasi

$$x_n = A \cdot x_{n-1},$$

dengan A suatu matriks 2x2, dan setiap x[n] merupakan matriks/vektor 2x1.

```
>A=[1,1;1,2]; function suku(x,n) := A.x[,n-1]
>sequence("suku",[1;1],6)
```

```
Real 2 x 6 matrix

            1             2             5            13       ...
            1             3             8            21       ...
```

Hasil yang sama juga dapat diperoleh dengan menggunakan fungsi perpangkatan matriks "matrixpower()". Cara ini lebih cepat, karena hanya menggunakan perkalian matriks sebanyak log_2(n).

$$x_n = A.x_{n-1} = A^2.x_{n-2} = A^3.x_{n-3} = ... = A^{n-1}.x_1.$$

```
>sequence("matrixpower(A,n).[1;1]",1,6)
```

```
Real 2 x 6 matrix

            1             5            13            34       ...
            1             8            21            55       ...
```

Spiral Theodorus

image: Spiral_of_Theodorus.png
Spiral Theodorus (spiral segitiga siku-siku) dapat digambar secara rekursif. Rumus rekursifnya adalah:

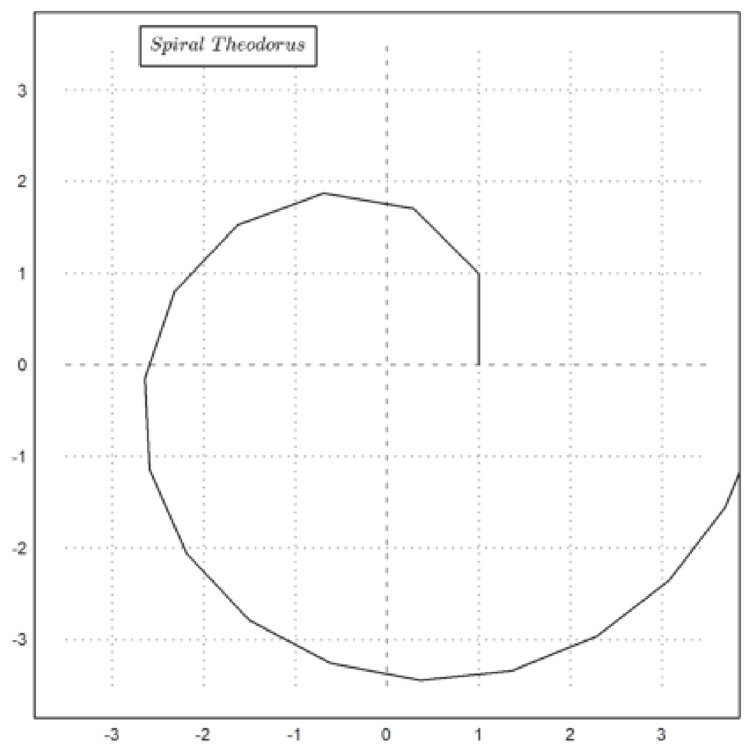
$$x_n = \left(1 + \frac{i}{\sqrt{n-1}}\right) x_{n-1}, \quad x_1 = 1,$$

yang menghasilkan barisan bilangan kompleks.

```
>function g(n) := 1+I/sqrt(n)
```

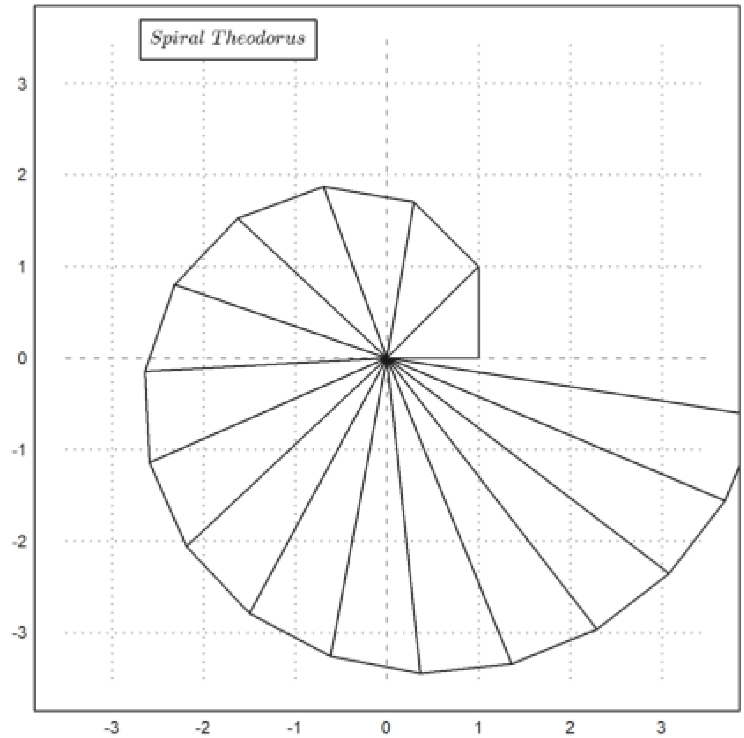
Rekursinya dapat dijalankan sebanyak 17 untuk menghasilkan barisan 17 bilangan kompleks, kemudian digambar bilangan-bilangan kompleksnya.

```
>x=sequence("g(n-1)*x[n-1]",1,17); plot2d(x,r=3.5); textbox(latex("Spiral\
```



Selanjutnya dihubungan titik 0 dengan titik-titik kompleks tersebut menggunakan loop.

```
>for i=1:cols(x); plot2d([0,x[i]],>add); end:
```




```
>
```

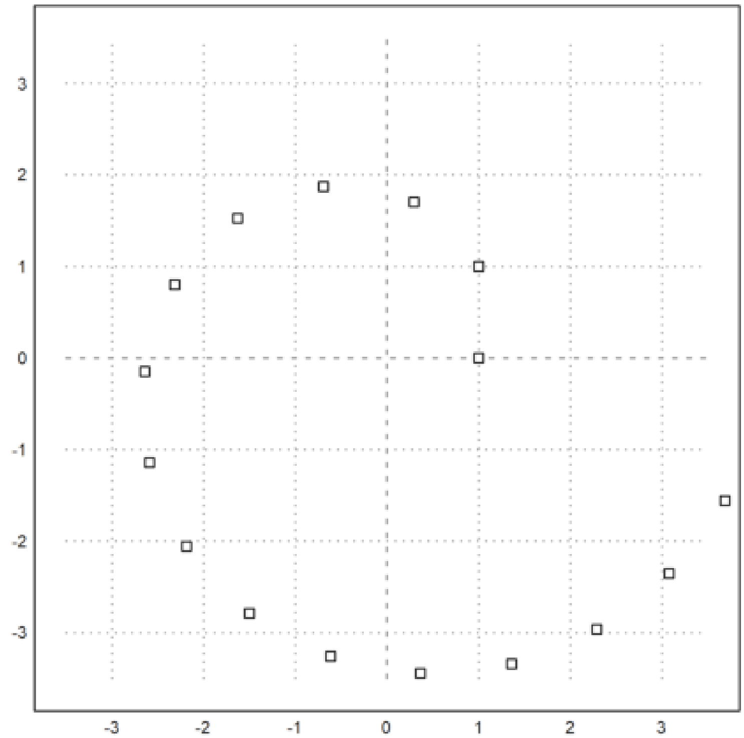
Spiral tersebut juga dapat didefinisikan menggunakan fungsi rekursif, yang tidak memmerlukan indeks dan bilangan kompleks. Dalam hal ini diigunakan vektor kolom pada bidang.

```
>function gstep (v) ...
```

```
  w=[-v[2];v[1]];
  return v+w/norm(w);
  endfunction
```

Jika dilakukan iterasi 16 kali dimulai dari [1;0] akan didapatkan matriks yang memuat vektor-vektor dari setiap iterasi.

```
>x=iterate("gstep",[1;0],16); plot2d(x[1],x[2],r=3.5,>points):
```



Kekonvergenan

Terkadang kita ingin melakukan iterasi sampai konvergen. Apabila iterasinya tidak konvergen setelah ditunggu lama, Anda dapat menghentikannya dengan menekan tombol [ESC].

```
>iterate("cos(x)",1) // iterasi x(n+1)=cos(x(n)), dengan x(0)=1.
```

```
0.739085133216
```

Iterasi tersebut konvergen ke penyelesaian persamaan

$$x = \cos(x).$$

Iterasi ini juga dapat dilakukan pada interval, hasilnya adalah barisan interval yang memuat akar tersebut.

```
>hasil := iterate("cos(x)",~1,2~) //iterasi x(n+1)=cos(x(n)), dengan interv
```

```
~0.739085133211,0.7390851332133~
```

Jika interval hasil tersebut sedikit diperlebar, akan terlihat bahwa interval tersebut memuat akar persamaan x=cos(x).

```
>h=expand(hasil,100), cos(h) << h
```

```
~0.73908513309,0.73908513333~
1
```

Iterasi juga dapat digunakan pada fungsi yang didefinisikan.

```
>function f(x) := (x+2/x)/2
```

Iterasi x(n+1)=f(x(n)) akan konvergen ke akar kuadrat 2.

```
>iterate("f",2), sqrt(2)
```

```
1.41421356237
1.41421356237
```

Jika pada perintah iterate diberikan tambahan parameter n, maka hasil iterasinya akan ditampilkan mulai dari iterasi pertama sampai ke-n.

```
>iterate("f",2,5)
```

```
[2,  1.5,  1.41667,  1.41422,  1.41421,  1.41421]
```

Untuk iterasi ini tidak dapat dilakukan terhadap interval.

```
>niterate("f",~1,2~,5)
```

```
[ ~1,2~,  ~1,2~,  ~1,2~,  ~1,2~,  ~1,2~,  ~1,2~ ]
```

Perhatikan, hasil iterasinya sama dengan interval awal. Alasannya adalah perhitungan dengan interval bersifat terlalu longgar. Untuk meingkatkan perhitungan pada ekspresi dapat digunakan pembagian intervalnya, menggunakan fungsi ieval().

```
>function s(x) := ieval("(x+2/x)/2",x,10)
```

Selanjutnya dapat dilakukan iterasi hingga diperoleh hasil optimal, dan intervalnya tidak semakin mengecil. Hasilnya berupa interval yang memuat akar persamaan:

$$x = \frac{1}{2}\left(x + \frac{2}{x}\right).$$

Satu-satunya solusi adalah

$$x = \sqrt{2}.$$

```
>iterate("s",~1,2~)
```

```
~1.41421356236,1.41421356239~
```

Fungsi "iterate()" juga dapat bekerja pada vektor. Berikut adalah contoh fungsi vektor, yang menghasilkan rata-rata aritmetika dan rata-rata geometri.

$$(a_{n+1}, b_{n+1}) = \left(\frac{a_n + b_n}{2}, \sqrt{a_n b_n}\right)$$

Iterasi ke-n disimpan pada vektor kolom x[n].

```
>function g(x) := [(x[1]+x[2])/2;sqrt(x[1]*x[2])]
```

Iterasi dengan menggunakan fungsi tersebut akan konvergen ke rata-rata aritmetika dan geometri dari nilai-nilai awal.

```
>iterate("g",[1;5])
```

```
      2.60401
      2.60401
```

Hasil tersebut konvergen agak cepat, seperti kita cek sebagai berikut.

```
>iterate("g",[1;5],4)
```

```
            1             3       2.61803       2.60403       2.60401
            5       2.23607       2.59002       2.60399       2.60401
```

Iterasi pada interval dapat dilakukan dan stabil, namun tidak menunjukkan bahwa limitnya pada batas-batas yang dihitung.

```
>iterate("g",[~1~;~5~],4)
```

```
Interval 2 x 5 matrix

~0.99999999999999778,1.0000000000000022~      ...
~4.9999999999999911,5.0000000000000089~      ...
```

Iterasi berikut konvergen sangat lambat.

$$x_{n+1} = \sqrt{x_n}.$$

```
>iterate("sqrt(x)",2,10)
```

```
[2,  1.41421,  1.18921,  1.09051,  1.04427,  1.0219,  1.01089,
1.00543,  1.00271,  1.00135,  1.00068]
```

Kekonvergenan iterasi tersebut dapat dipercepatdengan percepatan Steffenson:

```
>steffenson("sqrt(x)",2,10)
```

```
[1.04888,  1.00028,  1,  1]
```

Iterasi menggunakan Loop yang ditulis Langsung

Berikut adalah beberapa contoh penggunaan loop untuk melakukan iterasi yang ditulis langsung pada baris perintah.

```
>x=2; repeat x=(x+2/x)/2; until x^2~=2; end; x,
```

```
1.41421356237
```

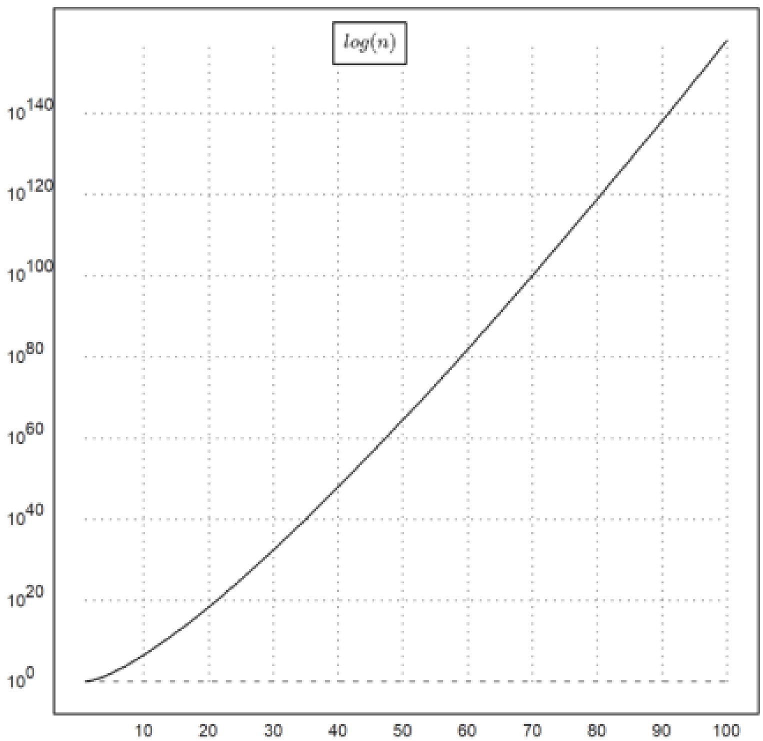
Penggabungan matriks menggunakan tanda "|" dapat digunakan untuk menyimpan semua hasil iterasi.

```
>v=[1]; for i=2 to 8; v=v|(v[i-1]*i); end; v,
```

```
[1,  2,  6,  24,  120,  720,  5040,  40320]
```

hasil iterasi juga dapat disimpan pada vektor yang sudah ada.

```
>v=ones(1,100); for i=2 to cols(v); v[i]=v[i-1]*i; end; ...
>plot2d(v,logplot=1); textbox(latex(&log(n)),x=0.5):
```




```
>A =[0.5,0.2;0.7,0.1]; b=[2;2]; ...
>x=[1;1]; repeat xnew=A.x-b; until all(xnew~=x); x=xnew; end; ...
>x,
```

```
       -7.09677
       -7.74194
```

Iterasi di dalam Fungsi

Fungsi atau program juga dapat menggunakan iterasi dan dapat digunakan untuk melakukan iterasi. Berikut adalah beberapa contoh iterasi di dalam fungsi.

Contoh berikut adalah suatu fungsi untuk menghitung berapa lama suatu iterasi konvergen. Nilai fungsi tersebut adalah hasil akhir iterasi dan banyak iterasi sampai konvergen.

```
>function map hiter(f$,x0) ...
```

```
x=x0;
maxiter=0;
repeat
  xnew=f$(x);
  maxiter=maxiter+1;
  until xnew~=x;
  x=xnew;
end;
return maxiter;
endfunction
```

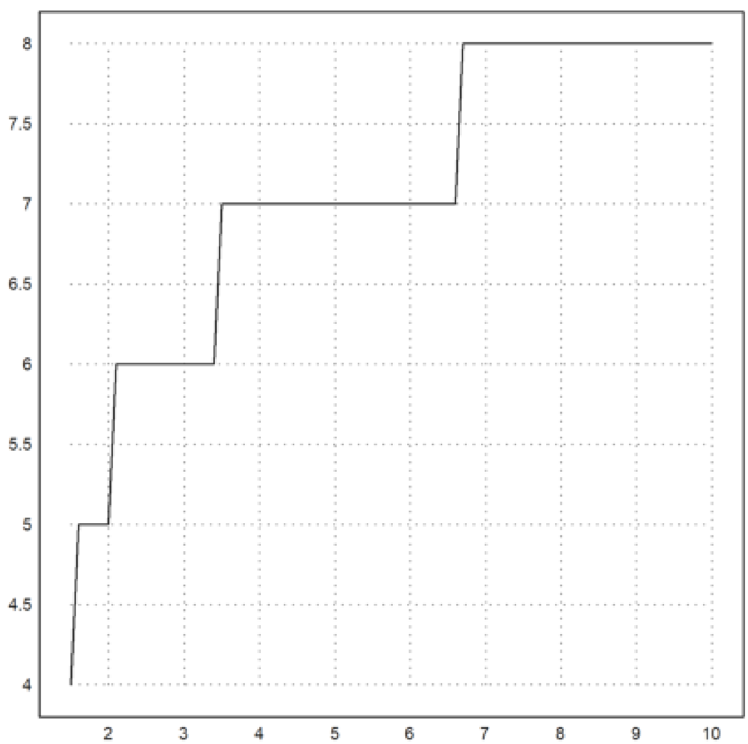
Misalnya, berikut adalah iterasi untuk mendapatkan hampiran akar kuadrat 2, cukup cepat, konvergen pada iterasi ke-5, jika dimulai dari hampiran awal 2.

```
>hiter("(x+2/x)/2",2)
```

```
5
```

Karena fungsinya didefinisikan menggunakan "map". maka nilai awalnya dapat berupa vektor.

```
>x=1.5:0.1:10; hasil=hiter("(x+2/x)/2",x); ...
>  plot2d(x,hasil):
```



Dari gambar di atas terlihat bahwa kekonvergenan iterasinya semakin lambat, untuk nilai awal semakin besar, namun penambahnnya tidak kontinu. Kita dapat menemukan kapan maksimum iterasinya bertambah.

```
>hasil[1:10]
```

```
[4,  5,  5,  5,  5,  5,  6,  6,  6,  6]
```

```
>x[nonzeros(differences(hasil))]
```

```
[1.5,  2,  3.4,  6.6]
```

maksimum iterasi sampai konvergen meningkat pada saat nilai awalnya 1.5, 2, 3.4, dan 6.6. Contoh berikutnya adalah metode Newton pada polinomial kompleks berderajat 3.

```
>p &= x^3-1; newton &= x-p/diff(p,x); $newton
```

$$x - \frac{x^3 - 1}{3\,x^2}$$

Selanjutnya didefinisikan fungsi untuk melakukan iterasi (aslinya 10 kali).

```
>function iterasi(f$,x,n=10) ...
```

```
loop 1 to n; x=f$(x); end;
return x;
endfunction
```

Kita mulai dengan menentukan titik-titik grid pada bidang kompleksnya.

```
>r=1.5; x=linspace(-r,r,501); Z=x+I*x'; W=iterasi(newton,Z);
```

Berikut adalah akar-akar polinomial di atas.

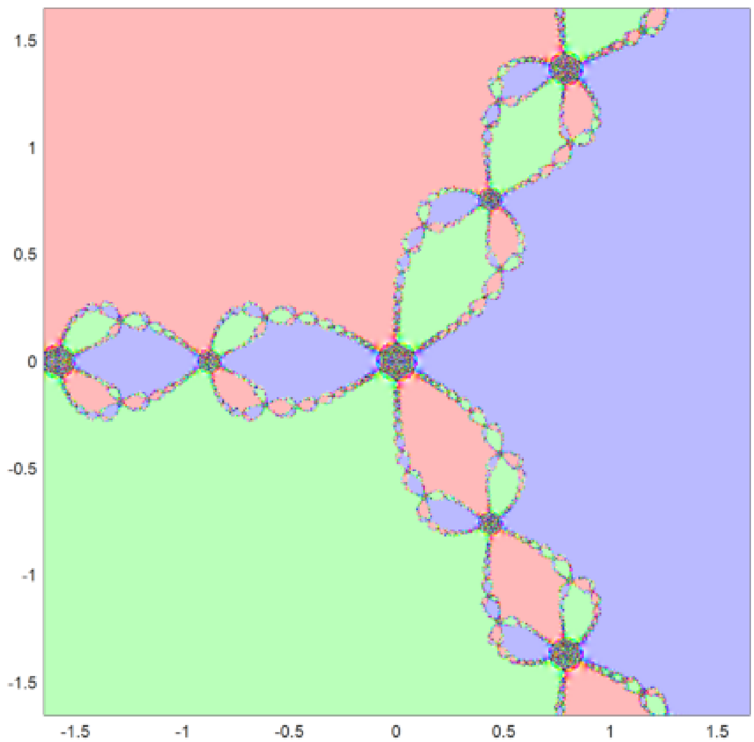
```
>z=&solve(p)()
```

```
[ -0.5+0.866025i,  -0.5-0.866025i,  1+0i  ]
```

Untuk menggambar hasil iterasinya, dihitung jarak dari hasil iterasi ke-10 ke masing-masing akar, kemudian digunakan untuk menghitung warna yang akan digambar, yang menunjukkan limit untuk masing-masing nilai awal.

Fungsi plotrgb() menggunakan jendela gambar terkini untuk menggambar warna RGB sebagai matriks.

```
>C=rgb(max(abs(W-z[1]),1),max(abs(W-z[2]),1),max(abs(W-z[3]),1)); ...
>  plot2d(none,-r,r,-r,r); plotrgb(C):
```



Iterasi Simbolik

Seperti sudah dibahas sebelumnya, untuk menghasilkan barisan ekspresi simbolik dengan Maxima dapat digunakan fungsi makelist().

```
>&powerdisp:true // untuk menampilkan deret pangkat mulai dari suku berpang
```

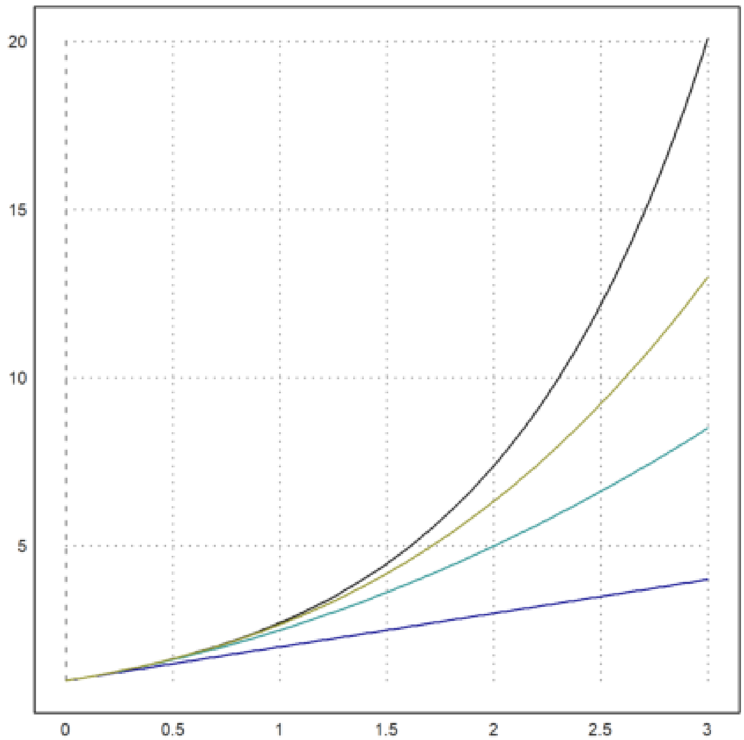
```
true
```

```
>deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // barisan deret Tay
```

$$\left[1+x,1+x+\frac{x^2}{2},1+x+\frac{x^2}{2}+\frac{x^3}{6}\right]$$

Untuk mengubah barisan deret tersebut menjadi vektor string di EMT digunakan fungsi mxm2str(). Selanjutnya, vektor string/ekspresi hasilnya dapat digambar seperti menggambar vektor eskpresi pada EMT.

```
>plot2d("exp(x)",0,3); // plot fungsi aslinya, e^x
>plot2d(mxm2str("deret"),>add,color=4:6): // plot ketiga deret taylor hampi
```

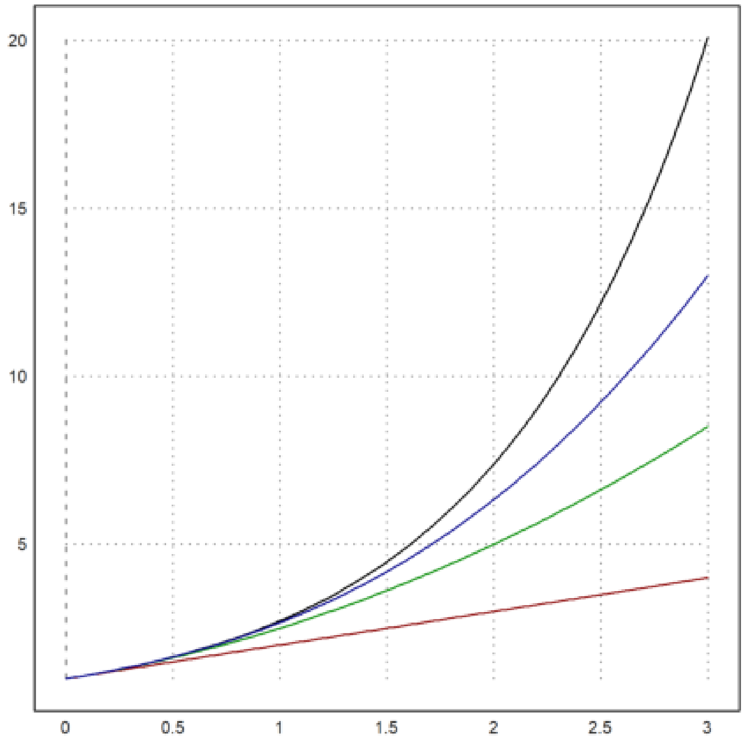


Selain cara di atas dapat juga dengan cara menggunakan indeks pada vektor/list yang dihasilkan.

```
>$deret[3]
```

$$1+x+\frac{x^2}{2}+\frac{x^3}{6}$$

```
>plot2d(["exp(x)",&deret[1],&deret[2],&deret[3]],0,3,color=1:4):
```



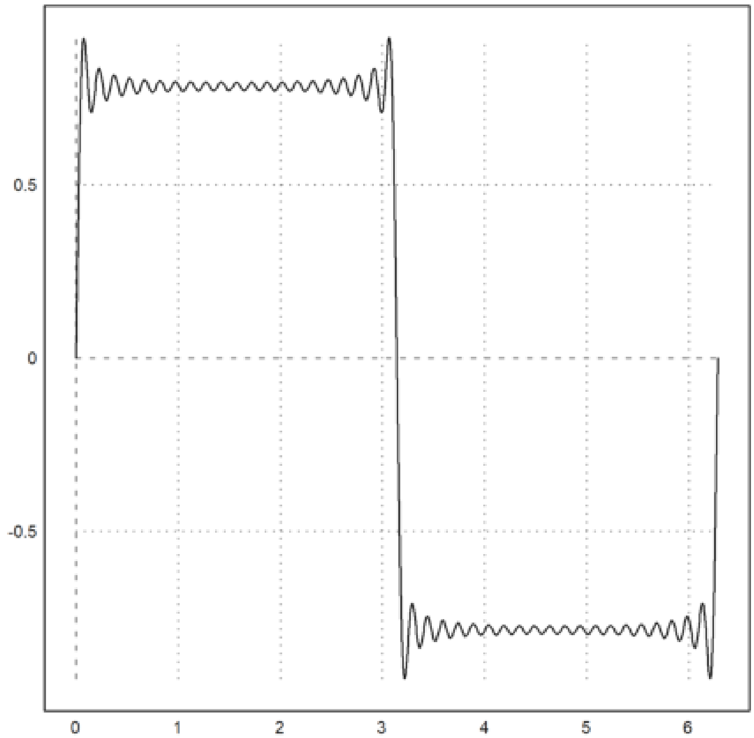

```
>$sum(sin(k*x)/k,k,1,5)
```

$$\sin x+\frac{\sin \left(2\,x\right)}{2}+\frac{\sin \left(3\,x\right)}{3}+\frac{\sin \left(4\,x\right)}{4}+\frac{\sin \left(5\,x\right)}{5}$$

Berikut adalah cara menggambar kurva

$$y=\sin(x)+\frac{\sin 3x}{3}+\frac{\sin 5x}{5}+\ldots.$$

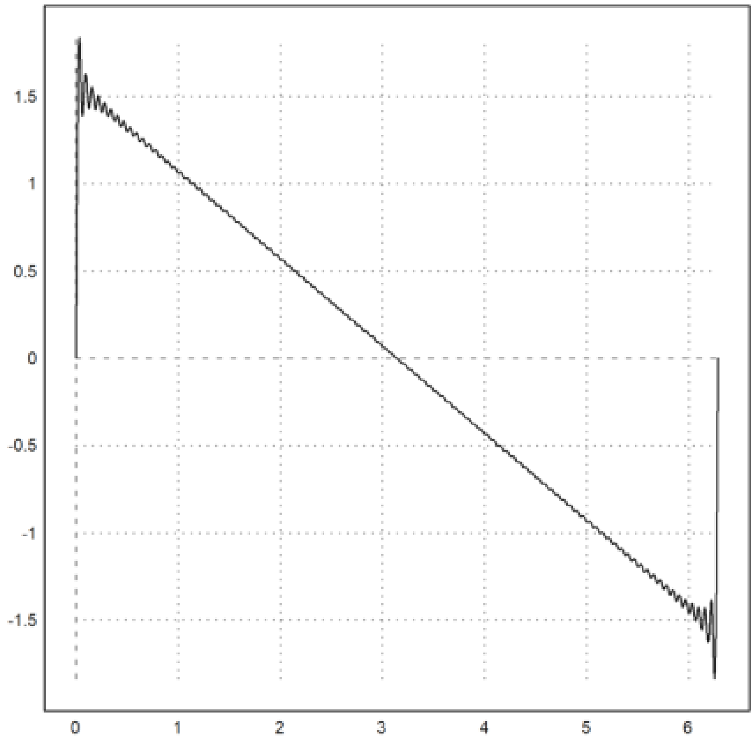
```
>plot2d(&sum(sin((2*k+1)*x)/(2*k+1),k,0,20),0,2pi):
```



Hal serupa juga dapat dilakukan dengan menggunakan matriks, misalkan kita akan menggambar kurva

$$y = \sum_{k=1}^{100} \frac{\sin(kx)}{k}, \quad 0 \le x \le 2\pi.$$

```
>x=linspace(0,2pi,1000); k=1:100; y=sum(sin(k*x')/k)'; plot2d(x,y):
```



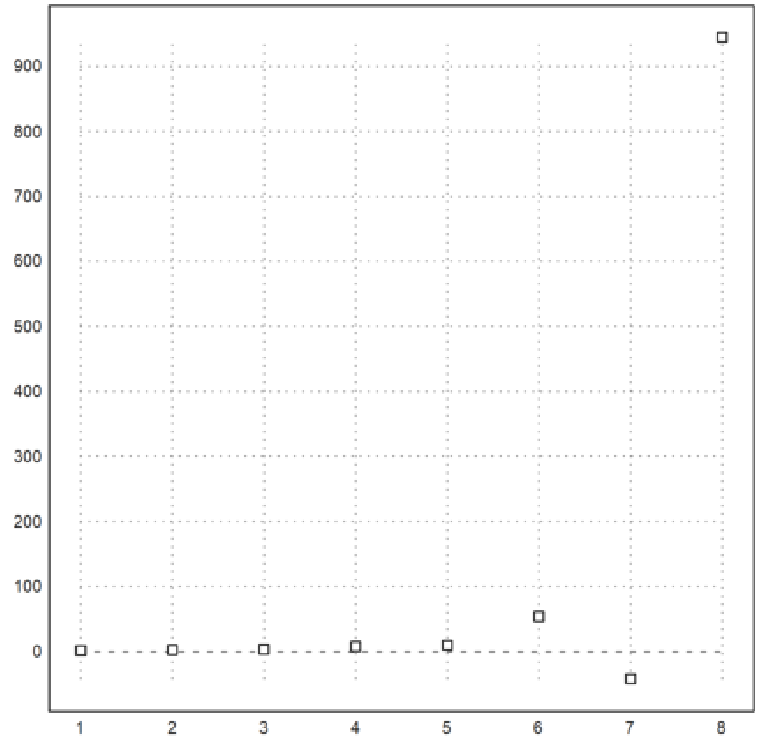
Tabel Fungsi

Terdapat cara menarik untuk menghasilkan barisan dengan ekspresi Maxima. Perintah mxmtable() berguna untuk menampilkan dan menggambar barisan dan menghasilkan barisan sebagai vektor kolom.

Sebagai contoh berikut adalah barisan turunan ke-n x^x di x=1.

```
>mxmtable("diffat(x^x,x=1,n)","n",1,8,frac=1);
```

```
      1
      2
      3
      8
     10
     54
    -42
    944
```




```
>$'sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n),simpsum=true)) // simpsum:me
```

$$\sum_{k=1}^{n} k = \frac{n\,(1+n)}{2}$$

```
>$'sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf),simpsum=
```

$$\sum_{k=0}^{\infty} \frac{1}{k+3^k} = \sum_{k=0}^{\infty} \frac{1}{k+3^k}$$

Di sini masih gagal, hasilnya tidak dihitung.

```
>$'sum(1/x^2, x, 1, inf)= ev(sum(1/x^2, x, 1, inf),simpsum=true) // ev: men
```

$$\sum_{x=1}^{\infty} \frac{1}{x^2} = \frac{\pi^2}{6}$$

```
>$'sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf),si
```

$$\sum_{k=1}^{\infty}{\frac{\left(-1\right)^{-1+k}}{k}}=-\sum_{x=1}^{\infty}{\frac{\left(-1\right)^{x}}{x}}$$

Di sini masih gagal, hasilnya tidak dihitung.

```
>$'sum((-1)^k/(2*k-1), k, 1, inf) = factor(ev(sum((-1)^k/(2*k-1), k, 1, inf
```

$$\sum_{k=1}^{\infty}{\frac{\left(-1\right)^{k}}{-1+2\,k}}=\sum_{k=1}^{\infty}{\frac{\left(-1\right)^{k}}{-1+2\,k}}$$

```
>$ev(sum(1/n!, n, 0, inf),simpsum=true)
```

$$\sum_{n=0}^{\infty}{\frac{1}{n!}}$$

Di sini masih gagal, hasilnya tidak dihitung, harusnya hasilnya e.

```
>&assume(abs(x)<1); $'sum(a*x^k, k, 0, inf)=ev(sum(a*x^k, k, 0, inf),simpsu
```

$$a\,\sum_{k=0}^{\infty}{x^{k}}=\frac{a}{1-x}$$

Deret geometri tak hingga, dengan asumsi rasional antara -1 dan 1.

```
>$'sum(x^k/k!,k,0,inf)=ev(sum(x^k/k!,k,0,inf),simpsum=true)
```

$$\sum_{k=0}^{\infty}{\frac{x^{k}}{k!}}=\sum_{k=0}^{\infty}{\frac{x^{k}}{k!}}$$

```
>$limit(sum(x^k/k!,k,0,n),n,inf)
```

$$\lim_{n\rightarrow \infty }{\sum_{k=0}^{n}{\frac{x^{k}}{k!}}}$$

```
>function d(n) &= sum(1/(k^2-k),k,2,n); $'d(n)=d(n)
```

$$d\left(n\right)=\sum_{k=2}^{n}{\frac{1}{-k+k^2}}$$

```
>$d(10)=ev(d(10),simpsum=true)
```

$$\sum_{k=2}^{10}{\frac{1}{-k+k^2}}=\frac{9}{10}$$

```
>$d(100)=ev(d(100),simpsum=true)
```

$$\sum_{k=2}^{100}{\frac{1}{-k+k^2}}=\frac{99}{100}$$

Deret Taylor

Deret Taylor suatu fungsi f yang diferensiabel sampai tak hingga di sekitar x=a adalah:

$$f(x)=\sum_{k=0}^{\infty}\frac{(x-a)^k f^{(k)}(a)}{k!}.$$

```
>$'e^x =taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x=0, sampai su
```

$$e^{x}=1+x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\frac{x^5}{120}+\frac{x^6}{720}+\frac{x^7}{5040}+\frac{x^8}{40320}+\frac{x^9}{362880}+\frac{x^{10}}{3628800}$$

```
>$'log(x)=taylor(log(x),x,1,10)// deret log(x) di sekitar x=1
```

$$\log x=-1-\frac{\left(-1+x\right)^2}{2}+\frac{\left(-1+x\right)^3}{3}-\frac{\left(-1+x\right)^4}{4}+\frac{\left(-1+x\right)^5}{5}-\frac{\left(-1+x\right)^6}{6}+\frac{\left(-1+x\right)^7}{7}-\frac{\left(-1+x\right)^8}{8}+\frac{\left(-1+x\right.}{9}$$

BAB 6

MENGGUNAKAN EMT UNTUK GEOMETRI

article

eumat

Tugas Aplikasi Komputer Pekan 11-12

Nama : Sherlyta Icha Nadiastuty
NIM : 22305141013
Kelas : Matematika B

Visualisasi dan Perhitungan Geometri dengan EMT

Euler menyediakan beberapa fungsi untuk melakukan visualisasi dan perhitungan geometri, baik secara numerik maupun analitik (seperti biasanya tentunya, menggunakan Maxima). Fungsi-fungsi untuk visualisasi dan perhitungan geometeri tersebut disimpan di dalam file program "geometry.e", sehingga file tersebut harus dipanggil sebelum menggunakan fungsi-fungsi atau perintah-perintah untuk geometri.

```
>load geometry
```

```
Numerical and symbolic geometry.
```

Fungsi-fungsi Geometri

Fungsi-fungsi untuk Menggambar Objek Geometri:

```
defaultd:=textheight()*1.5: nilai asli untuk parameter d
setPlotrange(x1,x2,y1,y2): menentukan rentang x dan y pada bidang koordin
setPlotRange(r): pusat bidang koordinat (0,0) dan batas-batas sumbu-x dan
plotPoint (P, "P"): menggambar titik P dan diberi label "P"
plotSegment (A,B, "AB", d): menggambar ruas garis AB, diberi label "AB" s
plotLine (g, "g", d): menggambar garis g diberi label "g" sejauh d
plotCircle (c,"c",v,d): Menggambar lingkaran c dan diberi label "c"
plotLabel (label, P, V, d): menuliskan label pada posisi P
```

Fungsi-fungsi Geometri Analitik (numerik maupun simbolik):

```
turn(v, phi): memutar vektor v sejauh phi
turnLeft(v):   memutar vektor v ke kiri
turnRight(v):  memutar vektor v ke kanan
normalize(v): normal vektor v
crossProduct(v, w): hasil kali silang vektorv dan w.
lineThrough(A, B): garis melalui A dan B, hasilnya [a,b,c] sdh. ax+by=c.
lineWithDirection(A,v): garis melalui A searah vektor v
getLineDirection(g): vektor arah (gradien) garis g
getNormal(g): vektor normal (tegak lurus) garis g
getPointOnLine(g):  titik pada garis g
perpendicular(A, g):  garis melalui A tegak lurus garis g
parallel (A, g):  garis melalui A sejajar garis g
lineIntersection(g, h):  titik potong garis g dan h
projectToLine(A, g):   proyeksi titik A pada garis g
distance(A, B):  jarak titik A dan B
distanceSquared(A, B):  kuadrat jarak A dan B
quadrance(A, B): kuadrat jarak A dan B
areaTriangle(A, B, C):  luas segitiga ABC
computeAngle(A, B, C):   besar sudut <ABC
angleBisector(A, B, C): garis bagi sudut <ABC
circleWithCenter (A, r): lingkaran dengan pusat A dan jari-jari r
getCircleCenter(c):  pusat lingkaran c
getCircleRadius(c):  jari-jari lingkaran c
circleThrough(A,B,C):  lingkaran melalui A, B, C
middlePerpendicular(A, B): titik tengah AB
lineCircleIntersections(g, c): titik potong garis g dan lingkran c
circleCircleIntersections (c1, c2):  titik potong lingkaran c1 dan c2
planeThrough(A, B, C):  bidang melalui titik A, B, C
```

Fungsi-fungsi Khusus Untuk Geometri Simbolik:

```
getLineEquation (g,x,y): persamaan garis g dinyatakan dalam x dan y
getHesseForm (g,x,y,A): bentuk Hesse garis g dinyatakan dalam x dan y den
sisi positif (kanan/atas) garis
quad(A,B): kuadrat jarak AB
```

```
spread(a,b,c): Spread segitiga dengan panjang sisi-sisi a,b,c, yakni sin(
alpha sudut yang menghadap sisi a.
crosslaw(a,b,c,sa): persamaan 3 quads dan 1 spread pada segitiga dengan p
triplespread(sa,sb,sc): persamaan 3 spread sa,sb,sc yang memebntuk suatu
doublespread(sa): Spread sudut rangkap Spread 2*phi, dengan sa=sin(phi)^2
```

Contoh 1: Luas, Lingkaran Luar, Lingkaran Dalam Segitiga

Untuk menggambar objek-objek geometri, langkah pertama adalah menentukan rentang sumbu-sumbu koordinat. Semua objek geometri akan digambar pada satu bidang koordinat, sampai didefinisikan bidang koordinat yang baru.

```
>setPlotRange(-0.5,2.5,-0.5,2.5); // mendefinisikan bidang koordinat baru
```

Sekarang, tetapkan tiga titik dan plotkan.

```
>A=[1,0]; plotPoint(A,"A"); // definisi dan gambar tiga titik
>B=[0,1]; plotPoint(B,"B");
>C=[2,2]; plotPoint(C,"C");
```

Kemudian tiga segmen.

```
>plotSegment(A,B,"c"); // c=AB
>plotSegment(B,C,"a"); // a=BC
>plotSegment(A,C,"b"); // b=AC
```

Fungsi geometri mencakup fungsi untuk membuat garis dan lingkaran. Format untuk garis adalah [a,b,c], yang merepresentasikan garis dengan persamaan ax+by=c.

```
>lineThrough(B,C) // garis yang melalui B dan C
```

```
[-1,  2,  2]
```

Hitung garis tegak lurus yang melalui A pada BC.

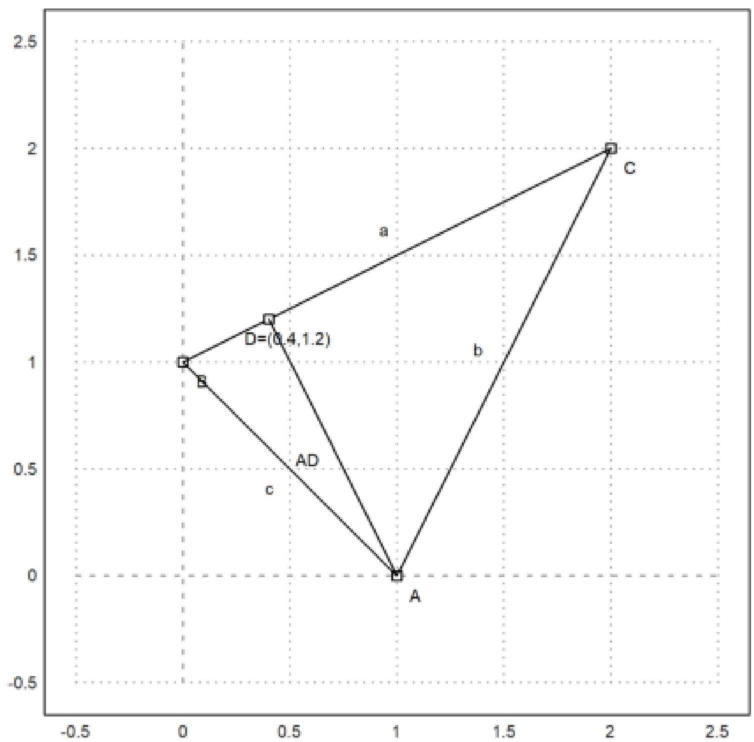
```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

Dan persimpangannya dengan BC.

```
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
```

Plot itu.

```
>plotPoint(D,value=1); // koordinat D ditampilkan
>aspect(1); plotSegment(A,D): // tampilkan semua gambar hasil plot...()
```



Hitung luas ABC:

$$L_{\triangle ABC}=\frac{1}{2}AD.BC.$$

```
>norm(A-D)*norm(B-C)/2 // AD=norm(A-D), BC=norm(B-C)
```

```
1.5
```

Bandingkan dengan rumus determinan.

```
>areaTriangle(A,B,C) // hitung luas segitiga langusng dengan fungsi
```

    1.5

Cara lain menghitung luas segitigas ABC:

```
>distance(A,D)*distance(B,C)/2
```

    1.5

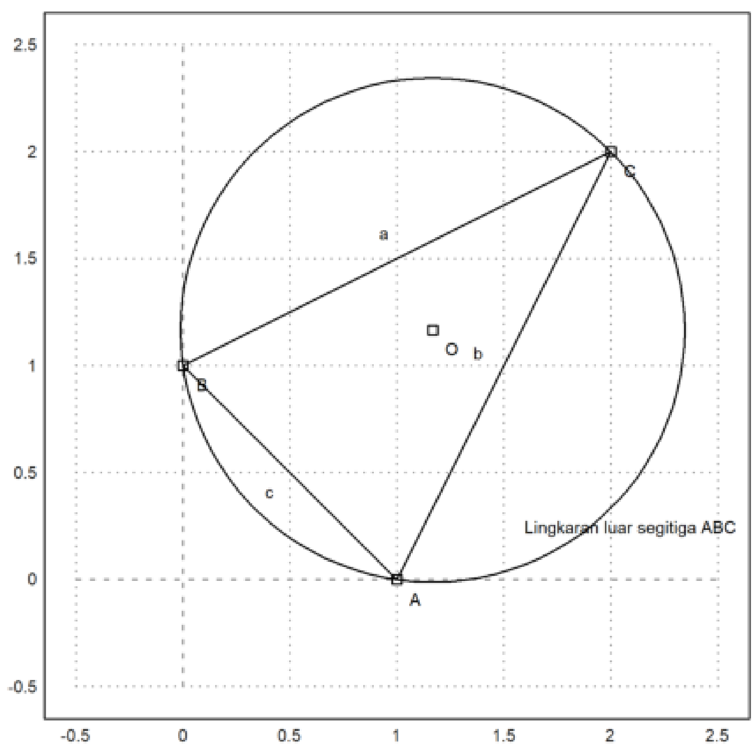
Sudut pada C.

```
>degprint(computeAngle(B,C,A))
```

    36°52'11.63''

Sekarang, lingkarilah segitiga tersebut.

```
>c=circleThrough(A,B,C); // lingkaran luar segitiga ABC
>R=getCircleRadius(c); // jari2 lingkaran luar
>O=getCircleCenter(c); // titik pusat lingkaran c
>plotPoint(O,"O"); // gambar titik "O"
>plotCircle(c,"Lingkaran luar segitiga ABC"):
```



Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.

```
>O, R
```

```
[1.16667,  1.16667]
1.17851130198
```

Sekarang akan digambar lingkaran dalam segitiga ABC. Titik pusat lingkaran dalam adalah titik potong garis-garis bagi sudut.

```
>l=angleBisector(A,C,B); // garis bagi <ACB
>g=angleBisector(C,A,B); // garis bagi <CAB
>P=lineIntersection(l,g) // titik potong kedua garis bagi sudut
```

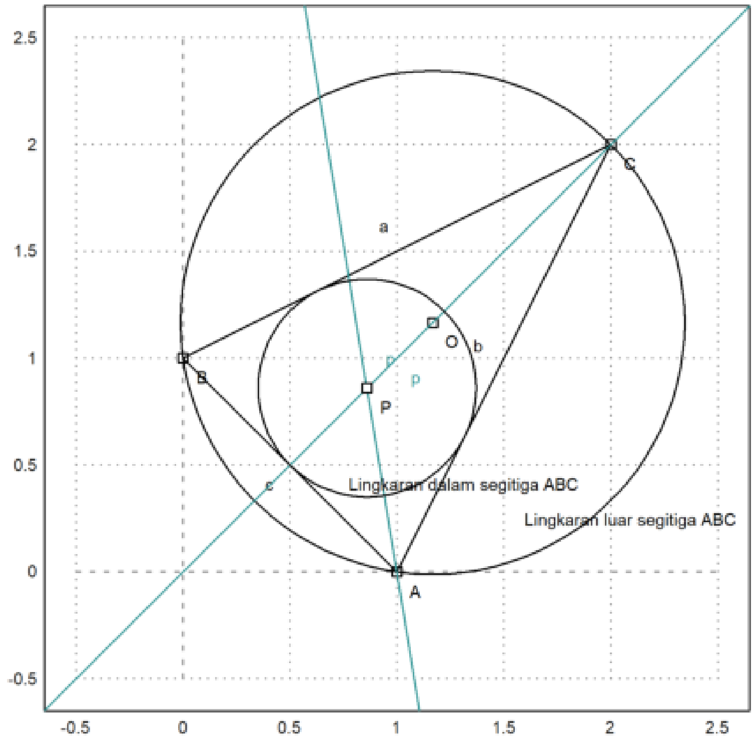
```
[0.86038,  0.86038]
```

Tambahkan semuanya ke plot.

```
>color(5); plotLine(l); plotLine(g); color(1); // gambar kedua garis bagi s
>plotPoint(P,"P"); // gambar titik potongnya
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam
```

```
0.509653732104
```

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"): // gamba
```



Latihan

1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.
Jawab :

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC");
>K=lineCircleIntersections(lineThrough(A,B),circleWithCenter(P,r))
```

```
[0.5,  0.5]
```

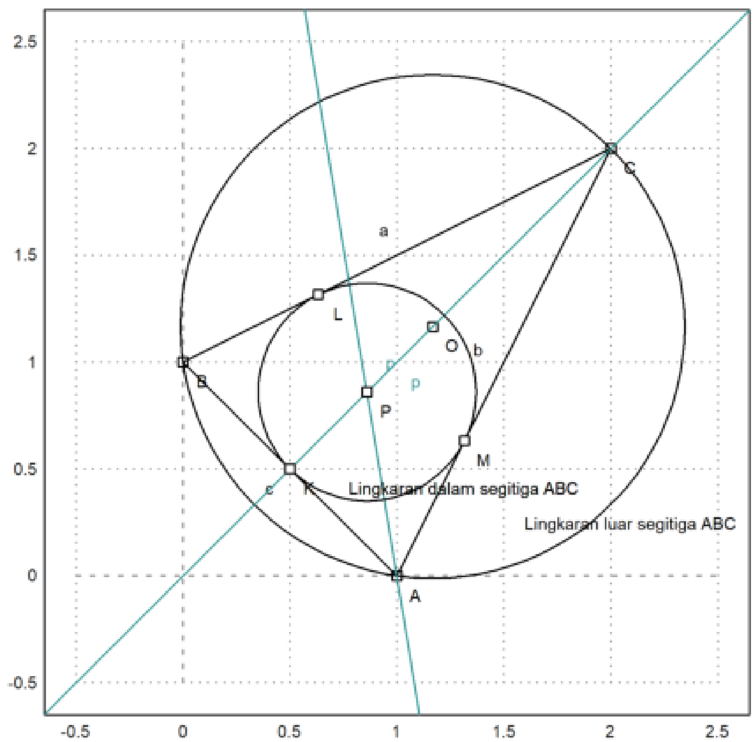
```
>L=lineCircleIntersections(lineThrough(C,B),circleWithCenter(P,r))
```

```
[0.632456,  1.31623]
```

```
>M=lineCircleIntersections(lineThrough(C,A),circleWithCenter(P,r))
```

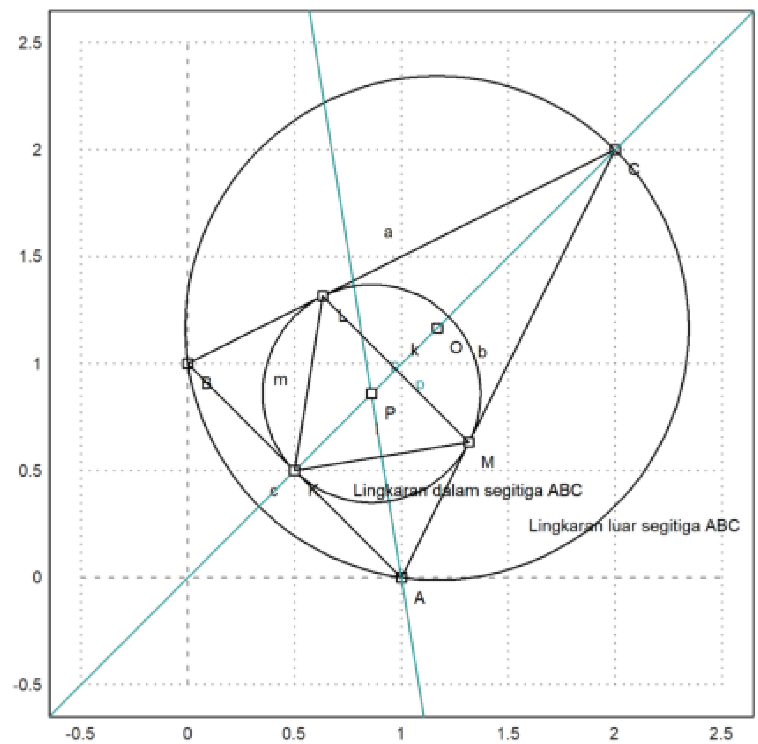
```
[1.31623,  0.632456]
```

```
>plotPoint(K,"K"); plotPoint(L,"L"); plotPoint(M,"M"):
```



2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut. Merupakan segitiga apakah itu?
Jawab :

```
>plotSegment(K,L,"m");
>plotSegment(L,M,"k");
>plotSegment(K,M,"l"):
```




```
>m=distance(K,L) // panjang sisi KL
```

    0.826905214631

```
>k=distance(L,M) // panjang sisi LM
```

    0.966999966873

```
>l=distance(K,M) // panjang sisi KM
```

    0.826905214631

Karena panjang sisi KL dan KM sama, maka dapat disimpulkan bahwa segitiga tersebut berbentuk segitiga sama kaki.

3. Hitung luas segitiga tersebut.
Jawab :

```
>h=sqrt(m^2-(k/2)^2) // tinggi segitiga
```

    0.67082039325

```
>k*h/2 // luas segitiga
```

```
0.324341649025
```

Jadi, lua ssegitiga adalah 0,324.

4. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.
Jawab :

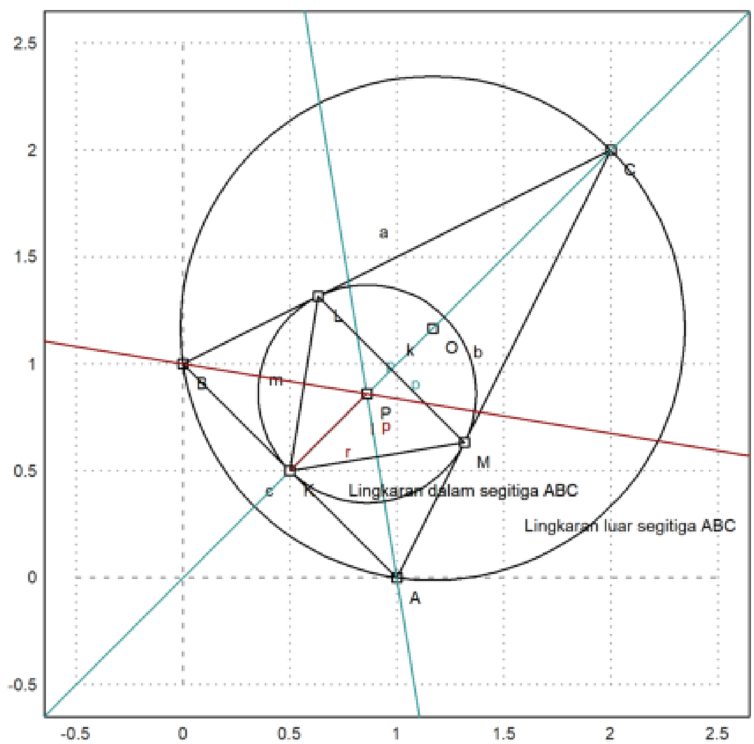
```
>P, r
```

```
[0.86038,  0.86038]
0.509653732104
```

```
>n=angleBisector(A,B,C)
```

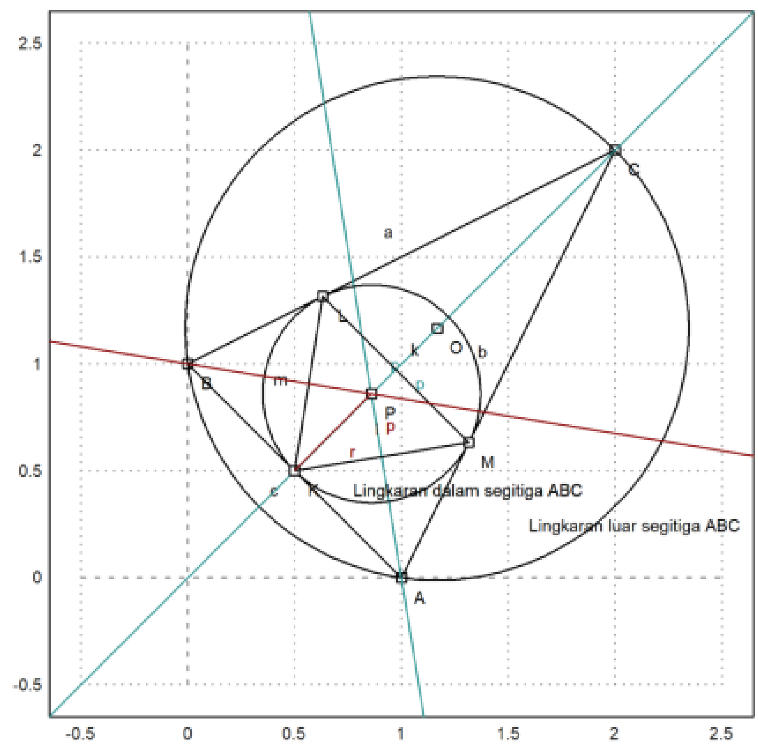
```
[-0.264911,  -1.63246,  -1.63246]
```

```
>color(2); plotLine(n):
```



5. Gambar jari-jari lingkaran dalam.
Jawab :

```
>plotSegment(P,K,"r"):
```



Contoh 2: Geometri Smbolik

Kita dapat menghitung geometri eksak dan simbolik menggunakan Maxima.

File geometri.e menyediakan fungsi yang sama (dan lebih banyak lagi) di Maxima. Namun, kita dapat menggunakan komputasi simbolik sekarang.

```
>A &= [1,0]; B &= [0,1]; C &= [2,2]; // menentukan tiga titik A, B, C
```

Fungsi untuk garis dan lingkaran bekerja seperti fungsi Euler, tetapi menyediakan komputasi simbolis.

```
>c &= lineThrough(B,C) // c=BC
```

$$[-1, 2, 2]$$

Kita bisa mendapatkan persamaan untuk sebuah garis dengan mudah.

```
>$getLineEquation(c,x,y), $solve(%,y) | expand // persamaan garis c
```

$$\left[y=\frac{x}{2}+1\right]$$

$$\left[y=\frac{x}{2}+1\right]$$

```
>$getLineEquation(lineThrough([x1,y1],[x2,y2]),x,y), $solve(%,y) // persama
```

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

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

```
>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A
```

$$\left(x_1-1\right)\,y-x\,y_1=-y_1$$

```
>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC
```

```
[2, 1, 2]
```

```
>Q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
```

```
  2   6
[-, -]
  5   5
```

```
>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
```

$$\left[\frac{2}{5}, \frac{6}{5}\right]$$

```
>$distance(A,Q) // jarak AQ
```

$$\frac{3}{\sqrt{5}}$$

```
>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran m
```

$$\left[\frac{7}{6}, \frac{7}{6}, \frac{5}{3\sqrt{2}}\right]$$

```
>r&=getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
```

$$1.178511301977579$$

```
>$computeAngle(A,C,B) // nilai <ACB
```

$$\arccos\left(\frac{4}{5}\right)$$

```
>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis
```

$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // ti
```

$$\left[\frac{\sqrt{2}\,\sqrt{5}+2}{6},\frac{\sqrt{2}\,\sqrt{5}+2}{6}\right]$$

```
>P() // hasilnya sama dengan perhitungan sebelumnya
```

```
[0.86038,  0.86038]
```

Garis dan Lingkaran yang Berpotongan

Tentu saja, kita juga bisa memotong garis dengan lingkaran, dan lingkaran dengan lingkaran.

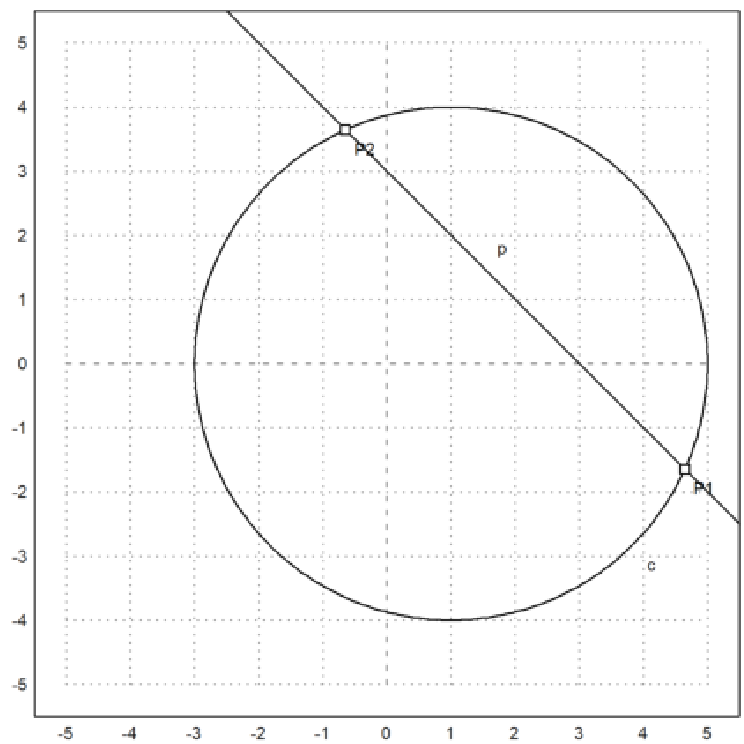
```
>A &:= [1,0]; c=circleWithCenter(A,4);
>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C);
>setPlotRange(5); plotCircle(c); plotLine(l);
```

Perpotongan garis dengan lingkaran menghasilkan dua titik dan jumlah titik perpotongan.

```
>{P1,P2,f}=lineCircleIntersections(l,c);
>P1, P2, f
```

```
[4.64575,  -1.64575]
[-0.645751,  3.64575]
2
```

```
>plotPoint(P1); plotPoint(P2):
```



Hal yang sama pada Maxima.

```
>c &= circleWithCenter(A,4) // lingkaran dengan pusat A jari-jari 4
```

[1, 0, 4]

```
>l &= lineThrough(B,C) // garis l melalui B dan C
```

[1, 1, 3]

```
>$lineCircleIntersections(l,c) | radcan, // titik potong lingkaran c dan ga
```

$$\left[\left[\sqrt{7}+2,1-\sqrt{7}\right],\left[2-\sqrt{7},\sqrt{7}+1\right]\right]$$

Akan ditunjukkan bahwa sudut-sudut yang menghadap bsuusr yang sama adalah sama besar.

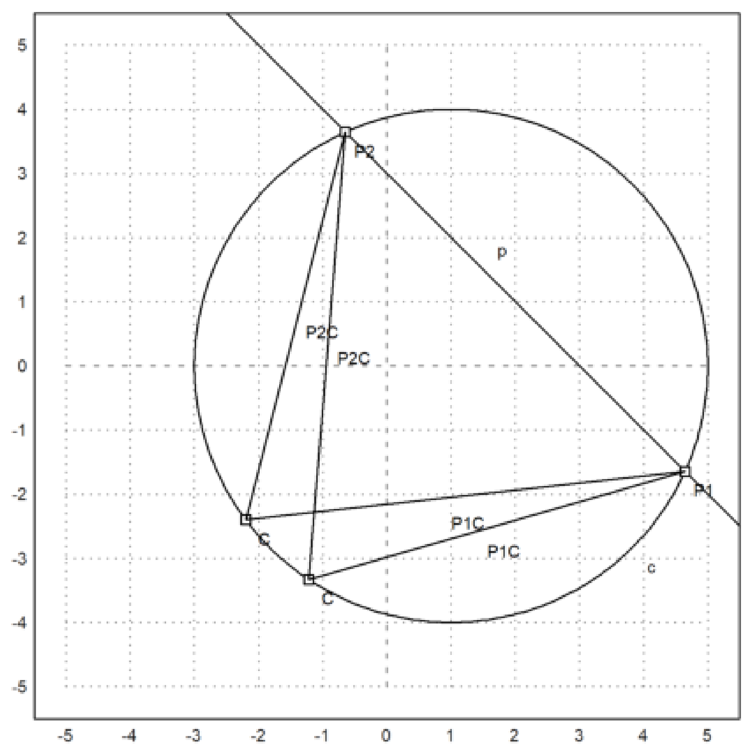
```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,
>degprint(computeAngle(P1,C,P2))
```

```
69°17'42.68''
```

```
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,
>degprint(computeAngle(P1,C,P2))
```

```
69°17'42.68''
```

```
>insimg;
```

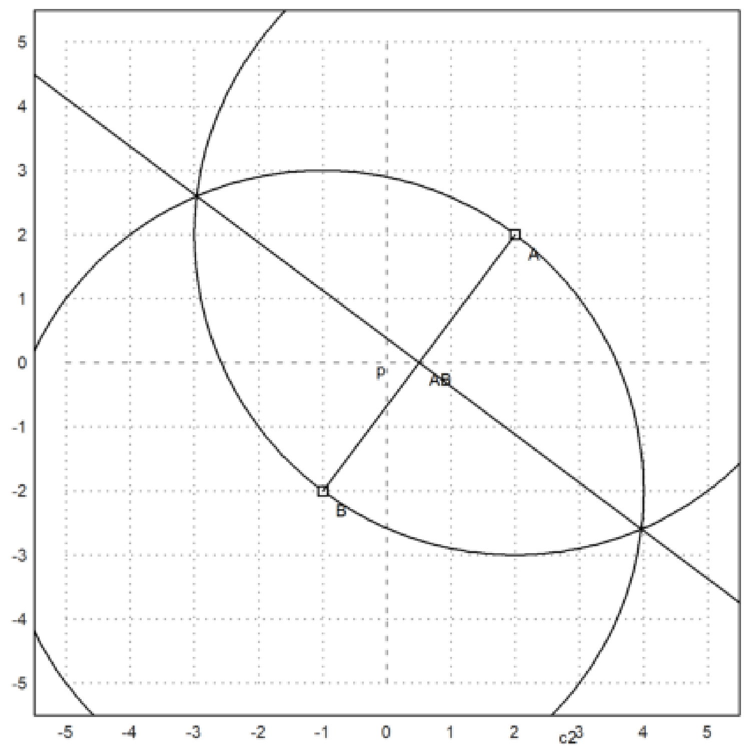


Garis Sumbu

Berikut adalah langkah-langkah menggambar garis sumbu ruas garis AB:

1. Gambar lingkaran dengan pusat A melalui B.
2. Gambar lingkaran dengan pusat B melalui A.
3. Tarik garis melallui kedua titik potong kedua lingkaran tersebut. Garis ini merupakan garis sumbu (melalui titik tengah dan tegak lurus) AB.

```
>A=[2,2]; B=[-1,-2];
>c1=circleWithCenter(A,distance(A,B));
>c2=circleWithCenter(B,distance(A,B));
>{P1,P2,f}=circleCircleIntersections(c1,c2);
>l=lineThrough(P1,P2);
>setPlotRange(5); plotCircle(c1); plotCircle(c2);
>plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l):
```



Selanjutnya, kami melakukan hal yang sama di Maxima dengan koordinat umum.

```
>A &= [a1,a2]; B &= [b1,b2];
>c1 &= circleWithCenter(A,distance(A,B));
>c2 &= circleWithCenter(B,distance(A,B));
>P &= circleCircleIntersections(c1,c2); P1 &= P[1]; P2 &= P[2];
```

Persamaan untuk persimpangan cukup rumit. Tetapi kita dapat menyederhanakannya, jika kita menyelesaikan untuk y.

```
>g &= getLineEquation(lineThrough(P1,P2),x,y);
>$solve(g,y)
```

$$\left[y=\frac{-\left(2\,b_1-2\,a_1\right)\,x+{b_2}^2+{b_1}^2-{a_2}^2-{a_1}^2}{2\,b_2-2\,a_2}\right]$$

Ini memang sama dengan tegak lurus tengah, yang dihitung dengan cara yang sama sekali berbeda.

```
>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
```

$$\left[y=\frac{-\left(2\,b_1-2\,a_1\right)\,x+{b_2}^2+{b_1}^2-{a_2}^2-{a_1}^2}{2\,b_2-2\,a_2}\right]$$

```
>h &=getLineEquation(lineThrough(A,B),x,y);
>$solve(h,y)
```

$$\left[y=\frac{\left(b_2-a_2\right)\,x-a_1\,b_2+a_2\,b_1}{b_1-a_1}\right]$$

Perhatikan hasil kali gradien garis g dan h adalah:

$$\frac{-(b_1-a_1)}{(b_2-a_2)}\times\frac{(b_2-a_2)}{(b_1-a_1)}=-1.$$

Artinya kedua garis tegak lurus.

Contoh 3: Rumus Heron

Rumus Heron menyatakan bahwa luas segitiga dengan panjang sisi-sisi a, b dan c adalah:

$$L=\sqrt{s(s-a)(s-b)(s-c)}\quad \text{dengan } s=(a+b+c)/2,$$

atau bisa ditulis dalam bentuk lain:

$$L=\frac{1}{4}\sqrt{(a+b+c)(b+c-a)(a+c-b)(a+b-c)}$$

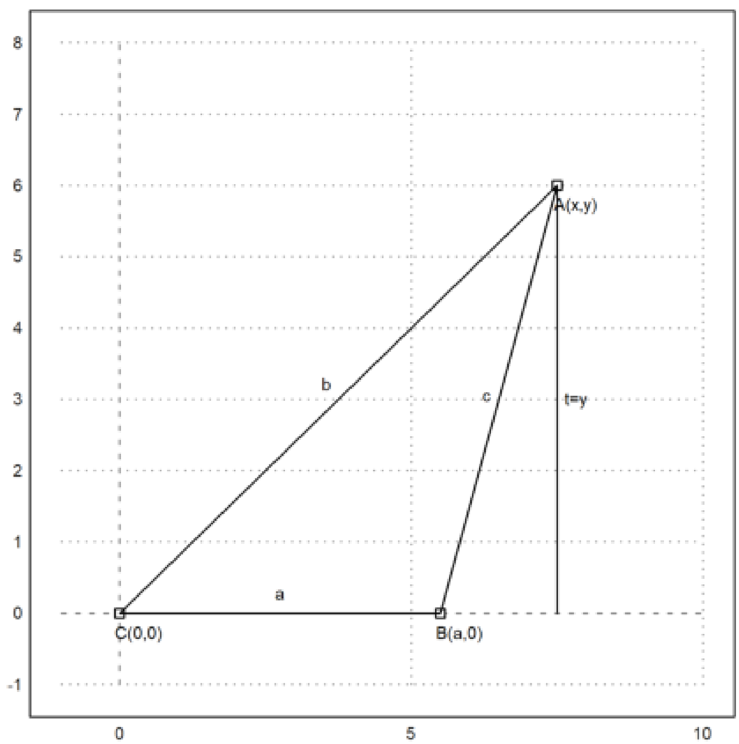
Untuk membuktikan hal ini kita misalkan C(0,0), B(a,0) dan A(x,y), b=AC, c=AB. Luas segitiga ABC adalah

$$L_{\triangle ABC} = \frac{1}{2} a \times y.$$

Nilai y didapat dengan menyelesaikan sistem persamaan:

$$x^2 + y^2 = b^2, \quad (x-a)^2 + y^2 = c^2.$$

```
>setPlotRange(-1,10,-1,8); plotPoint([0,0], "C(0,0)"); plotPoint([5.5,0], "
> plotPoint([7.5,6], "A(x,y)");
>plotSegment([0,0],[5.5,0], "a",25); plotSegment([5.5,0],[7.5,6],"c",15);
>plotSegment([0,0],[7.5,6],"b",25);
>plotSegment([7.5,6],[7.5,0],"t=y",25):
```




```
>&assume(a>0); sol &= solve([x^2+y^2=b^2,(x-a)^2+y^2=c^2],[x,y])
```

```
                2     2     2
             - c  + b  + a
       [[x = --------------, y =
```

```
                       2 a
              4        2   2       2   2     4       2   2     4
   sqrt(- c   + 2 b   c   + 2 a   c   - b   + 2 a   b   - a )
 - --------------------------------------------------],
                              2 a
            2     2     2
         - c   + b   + a
 [x = --------------, y = 
              2 a
            4        2   2       2   2     4       2   2     4
 sqrt(- c   + 2 b   c   + 2 a   c   - b   + 2 a   b   - a )
 --------------------------------------------------]]
                            2 a
```

Ekstrak solusi y.

```
>ysol &= y with sol[2][2]; $'y=sqrt(factor(ysol^2))
```

$$y=\frac{\sqrt{\left(-c+b+a\right)\,\left(c-b+a\right)\,\left(c+b-a\right)\,\left(c+b+a\right)}}{2\,a}$$

Kita mendapatkan rumus Heron.

```
>function H(a,b,c) &= sqrt(factor((ysol*a/2)^2)); $'H(a,b,c)=H(a,b,c)
```

$$H\left(a,b,c\right)=\frac{\sqrt{\left(-c+b+a\right)\,\left(c-b+a\right)\,\left(c+b-a\right)\,\left(c+b+a\right)}}{4}$$

```
>$'Luas=H(2,5,6) // luas segitiga dengan panjang sisi-sisi 2, 5, 6
```

$$Luas=\frac{3\,\sqrt{39}}{4}$$

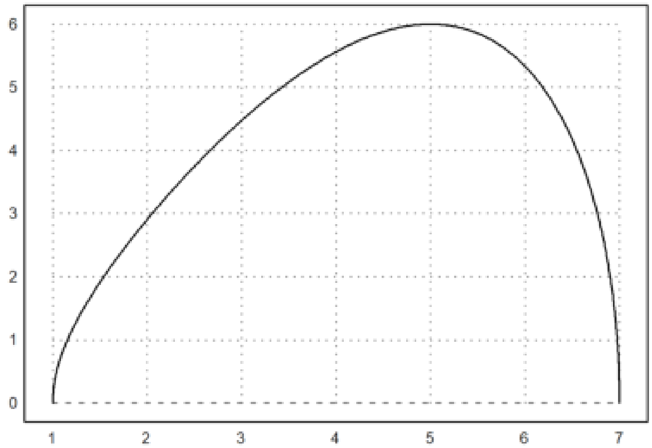
Tentu saja, setiap segitiga persegi panjang adalah kasus yang terkenal.

```
>H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5
```

```
6
```

Dan juga jelas, bahwa ini adalah segitiga dengan luas maksimal dan kedua sisi 3 dan 4.

```
>aspect (1.5); plot2d(&H(3,4,x),1,7): // Kurva luas segitiga sengan panjang
```



Kasus umum juga berfungsi.

```
>$solve(diff(H(a,b,c)^2,c)=0,c)
```

$$\left[c=-\sqrt{b^2+a^2}, c=\sqrt{b^2+a^2}, c=0\right]$$

Sekarang mari kita cari himpunan semua titik di mana b+c=d untuk beberapa konstanta d. Diketahui bahwa ini adalah elips.

```
>s1 &= subst(d-c,b,sol[2]); $s1
```

$$\left[x=\frac{(d-c)^2-c^2+a^2}{2\,a}, y=\frac{\sqrt{-(d-c)^4+2\,c^2\,(d-c)^2+2\,a^2\,(d-c)^2-c^4+2\,a^2\,c^2-a^4}}{2\,a}\right]$$

Dan buat fungsi ini.

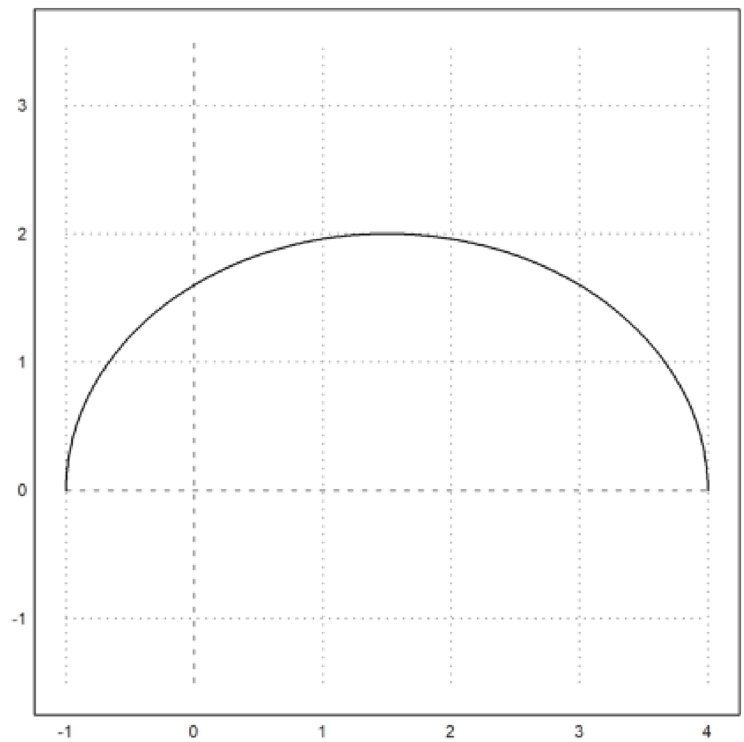
```
>function fx(a,c,d) &= rhs(s1[1]); $fx(a,c,d), function fy(a,c,d) &= rhs(s1
```

$$\frac{\sqrt{-(d-c)^4+2\,c^2\,(d-c)^2+2\,a^2\,(d-c)^2-c^4+2\,a^2\,c^2-a^4}}{2\,a}$$

$$\frac{\sqrt{-(d-c)^4+2\,c^2\,(d-c)^2+2\,a^2\,(d-c)^2-c^4+2\,a^2\,c^2-a^4}}{2\,a}$$

Sekarang kita bisa menggambar setnya. Sisi b bervariasi dari 1 hingga 4. Diketahui bahwa kita mendapatkan elips.

```
>aspect(1); plot2d(&fx(3,x,5),&fy(3,x,5),xmin=1,xmax=4,square=1):
```



Kita dapat memeriksa persamaan umum untuk elips ini, yaitu.

$$\frac{(x - x_m)^2}{u^2} + \frac{(y - y_m)}{v^2} = 1,$$

di mana (xm,ym) adalah pusat, dan u dan v adalah setengah sumbu.

```
>$ratsimp((fx(a,c,d)-a/2)^2/u^2+fy(a,c,d)^2/v^2 with [u=d/2,v=sqrt(d^2-a^2)
```

$$1$$

Kita lihat bahwa tinggi dan luas segitiga adalah maksimal untuk x=0.
Jadi luas segitiga dengan a+b+c=d maksimal jika segitiga sama sisi.
Kami ingin menurunkan ini secara analitis.

```
>eqns &= [diff(H(a,b,d-(a+b))^2,a)=0,diff(H(a,b,d-(a+b))^2,b)=0]; $eqns
```

$$\left[\frac{d\,\left(d-2\,a\right)\,\left(d-2\,b\right)}{8}-\frac{\left(-d+2\,b+2\,a\right)\,d\,\left(d-2\,b\right)}{8}=0,\frac{d\,\left(d-2\,a\right)\,\left(d-2\,b\right)}{8}-\frac{\left(-d+2\,b+2\,a\right)\,d\,\left(d-2\right.}{8}\right.$$

Kami mendapatkan beberapa minima, yang termasuk dalam segitiga dengan satu sisi 0, dan solusinya a=b=c=d/3.

```
>$solve(eqns,[a,b])
```

$$\left[\left[a=\frac{d}{3},b=\frac{d}{3}\right],\left[a=0,b=\frac{d}{2}\right],\left[a=\frac{d}{2},b=0\right],\left[a=\frac{d}{2},b=\frac{d}{2}\right]\right]$$

Ada juga metode Lagrange, memaksimalkan H(a,b,c)^2 terhadap a+b+d=d.

```
>&solve([diff(H(a,b,c)^2,a)=la,diff(H(a,b,c)^2,b)=la, ...
>   diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la])
```

```
                         d       d
           [[a = 0, b = -, c = -, la = 0], 
                         2       2
       d                 d                 d       d
   [a = -, b = 0, c = -, la = 0], [a = -, b = -, c = 0, la = 0], 
       2                 2                 2       2
                                    3
       d       d       d           d
   [a = -, b = -, c = -, la = ---]]
       3       3       3          108
```

Kita bisa membuat plot situasinya

Pertama-tama atur poin di Maxima.

```
>A &= at([x,y],sol[2]); $A
```

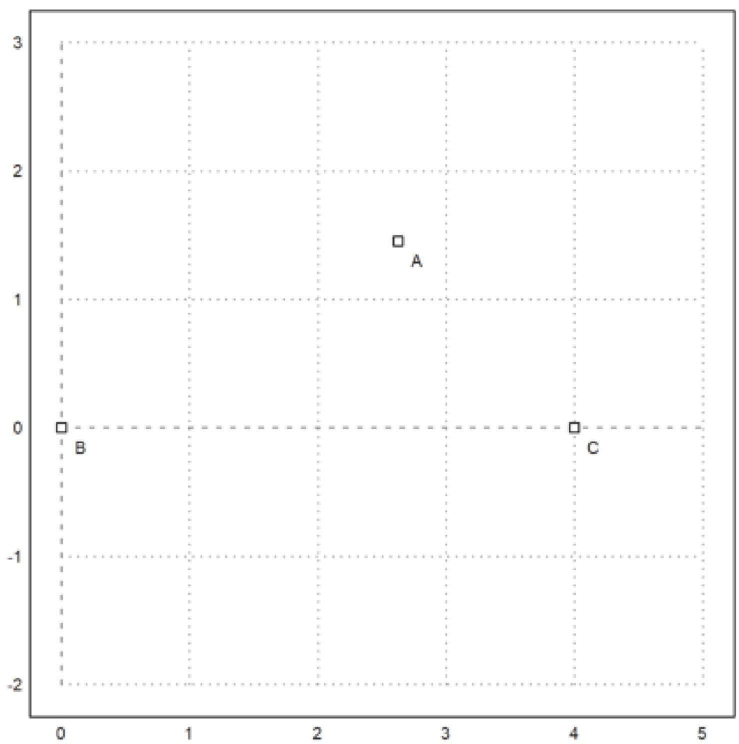
$$\left[\frac{-c^2+b^2+a^2}{2\,a},\frac{\sqrt{-c^4+2\,b^2\,c^2+2\,a^2\,c^2-b^4+2\,a^2\,b^2-a^4}}{2\,a}\right]$$

```
>B &= [0,0]; $B, C &= [a,0]; $C
```

$$[a,0]$$

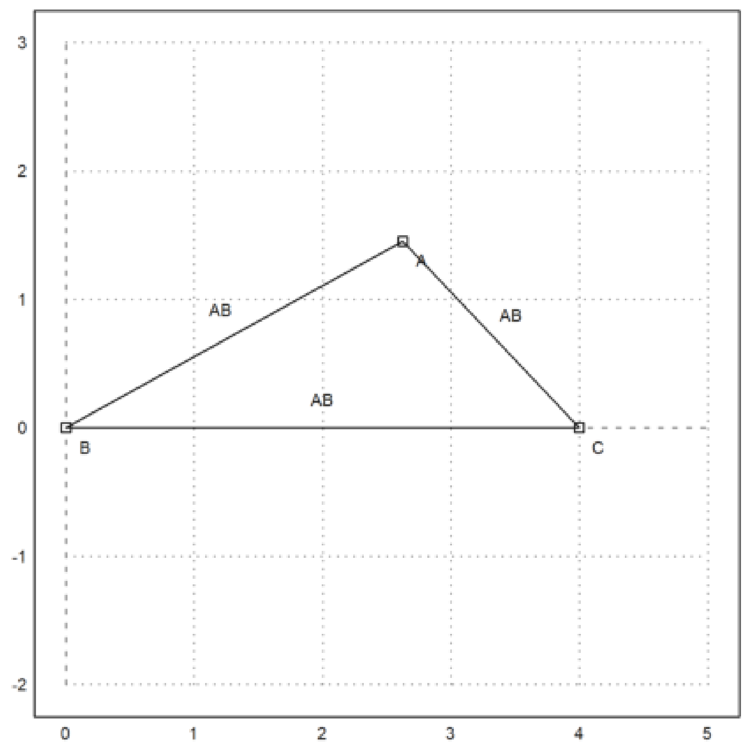
Kemudian atur rentang plot, dan plot titik-titiknya.

```
>setPlotRange(0,5,-2,3); ...
>a=4; b=3; c=2; ...
>plotPoint(mxmeval("B"),"B"); plotPoint(mxmeval("C"),"C"); ...
>plotPoint(mxmeval("A"),"A"):
```



Plot segmen.

```
>plotSegment(mxmeval("A"),mxmeval("C")); ...
>plotSegment(mxmeval("B"),mxmeval("C")); ...
>plotSegment(mxmeval("B"),mxmeval("A")):
```



Hitung tegak lurus tengah di Maxima.

```
>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
```

Dan pusat lingkaran.

```
>U &= lineIntersection(h,g);
```

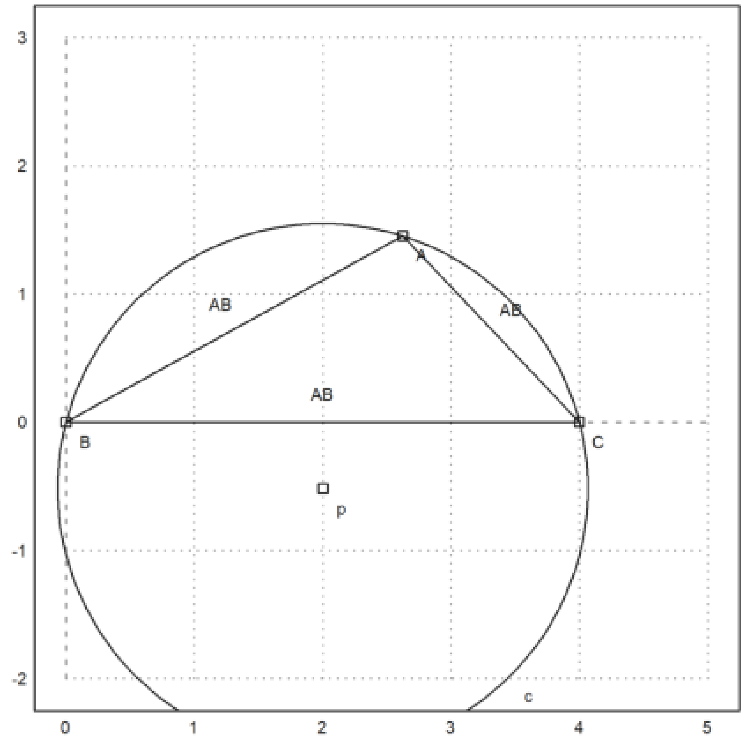
Kami mendapatkan rumus untuk jari-jari lingkaran.

```
>&assume(a>0,b>0,c>0); $distance(U,B) | radcan
```

$$\frac{i\,a\,b\,c}{\sqrt{c-b-a}\,\sqrt{c-b+a}\,\sqrt{c+b-a}\,\sqrt{c+b+a}}$$

Mari kita tambahkan ini ke plot.

```
>plotPoint(U()); ...
>plotCircle(circleWithCenter(mxmeval("U"),mxmeval("distance(U,C)"))):
```



Menggunakan geometri, kami memperoleh rumus sederhana

$$\frac{a}{\sin(\alpha)} = 2r$$

untuk radiusnya. Kita bisa cek, apakah ini benar dengan Maxima. Maxima akan memfaktorkan ini hanya jika kita kuadratkan.

```
>$c^2/sin(computeAngle(A,B,C))^2  | factor
```

$$-\frac{4\,a^2\,b^2\,c^2}{(c-b-a)\,(c-b+a)\,(c+b-a)\,(c+b+a)}$$

Contoh 4: Garis Euler dan Parabola

Garis Euler adalah garis yang ditentukan dari sembarang segitiga yang tidak sama sisi. Ini adalah garis tengah segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk orthocenter, circumcenter, centroid, titik Exeter dan pusat lingkaran sembilan titik segitiga.

Untuk demonstrasi, kami menghitung dan memplot garis Euler dalam sebuah segitiga.

Pertama, kita mendefinisikan sudut-sudut segitiga di Euler. Kami menggunakan definisi, yang terlihat dalam ekspresi simbolis.

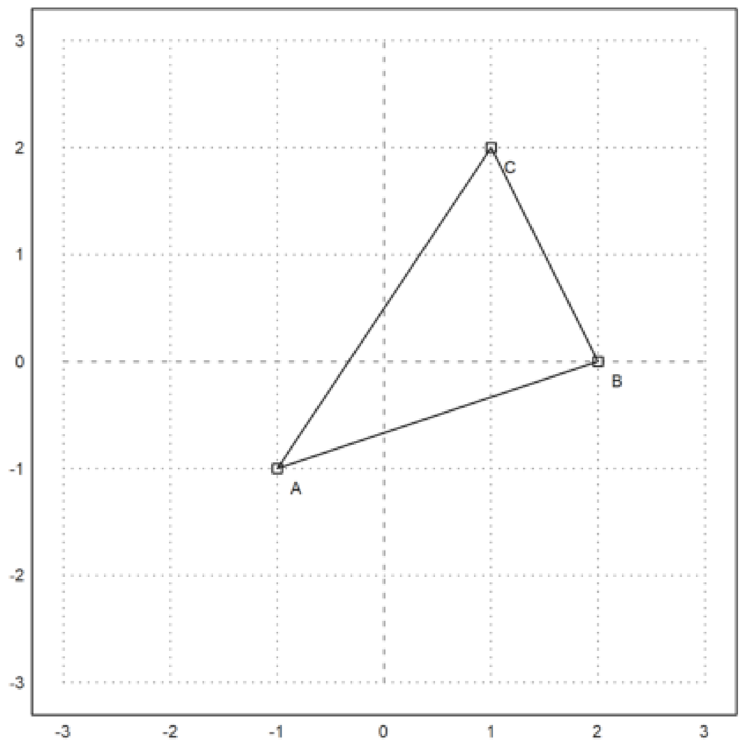
```
>A::=[-1,-1]; B::=[2,0]; C::=[1,2];
```

Untuk memplot objek geometris, kami menyiapkan area plot, dan menambahkan titik ke sana. Semua plot objek geometris ditambahkan ke plot saat ini.

```
>setPlotRange(3); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
```

Kita juga bisa menambahkan sisi segitiga.

```
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,""):
```



Berikut adalah luas segitiga, menggunakan rumus determinan. Tentu saja, kita harus mengambil nilai absolut dari hasil ini.

```
>$areaTriangle(A,B,C)
```

$$-\frac{7}{2}$$

Kita dapat menghitung koefisien sisi c.

```
>c &= lineThrough(A,B)
```

```
[- 1, 3, - 2]
```

Dan juga dapatkan rumus untuk baris ini.

```
>$getLineEquation(c,x,y)
```

$$3\,y-x=-2$$

Untuk bentuk Hesse, kita perlu menentukan sebuah titik, sehingga titik tersebut berada di sisi positif dari bentuk Hesse. Memasukkan titik menghasilkan jarak positif ke garis.

```
>$getHesseForm(c,x,y,C), $at(%,[x=C[1],y=C[2]])
```

$$\frac{7}{\sqrt{10}}$$

$$\frac{7}{\sqrt{10}}$$

Sekarang kita hitung lingkaran luar ABC.

```
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
```

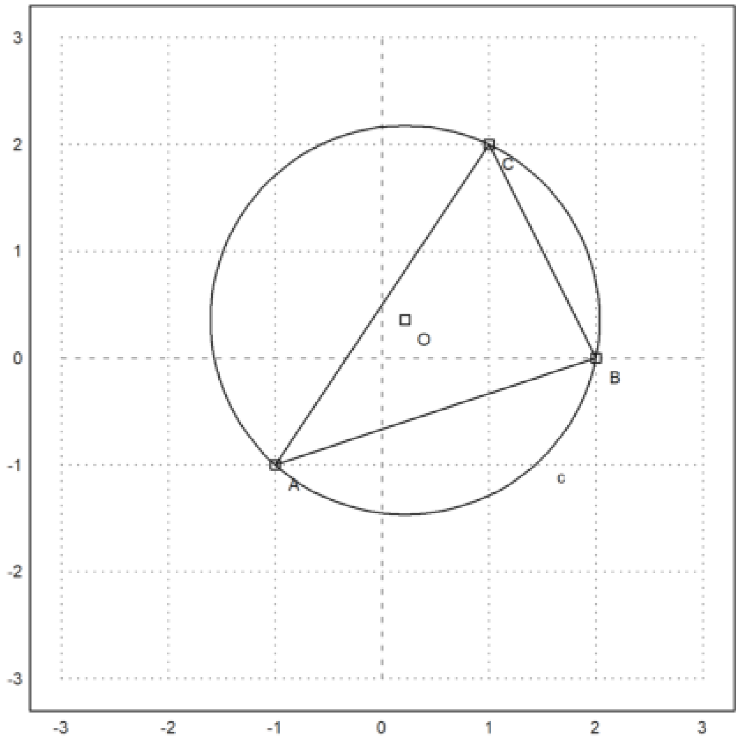
$$\left(y-\frac{5}{14}\right)^2+\left(x-\frac{3}{14}\right)^2=\frac{325}{98}$$

```
>O &= getCircleCenter(LL); $O
```

$$\left[\frac{3}{14},\frac{5}{14}\right]$$

Gambarkan lingkaran dan pusatnya. Cu dan U adalah simbolis. Kami mengevaluasi ekspresi ini untuk Euler.

```
>plotCircle(LL()); plotPoint(O(),"O"):
```



Kita dapat menghitung perpotongan ketinggian di ABC (orthocenter) secara numerik dengan perintah berikut.

```
>H &= lineIntersection(perpendicular(A,lineThrough(C,B)),...
>  perpendicular(B,lineThrough(A,C))); $H
```

$$\left[\frac{11}{7}, \frac{2}{7}\right]$$

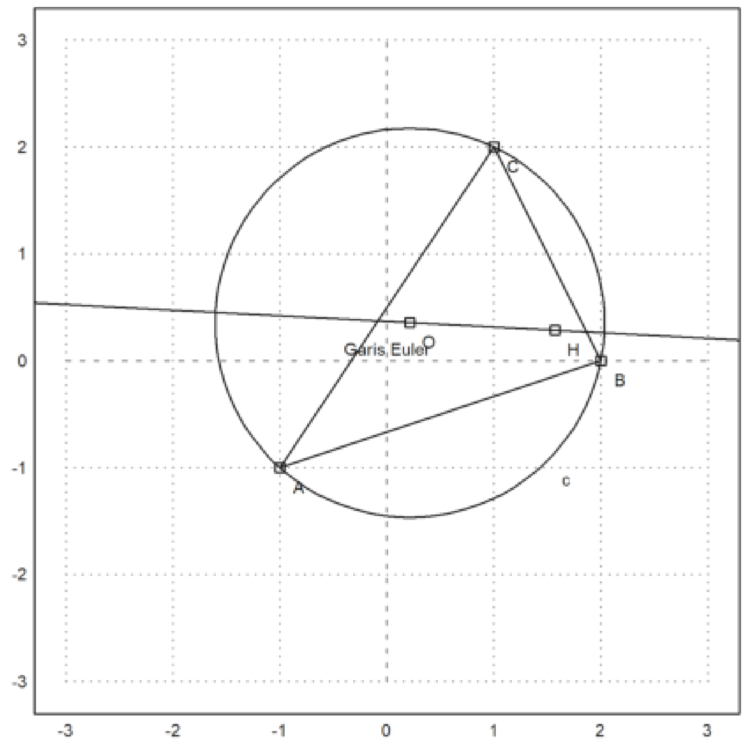
Sekarang kita dapat menghitung garis Euler dari segitiga.

```
>el &= lineThrough(H,O); $getLineEquation(el,x,y)
```

$$-\frac{19\,y}{14} - \frac{x}{14} = -\frac{1}{2}$$

Tambahkan ke plot kami.

```
>plotPoint(H(),"H"); plotLine(el(),"Garis Euler"):
```

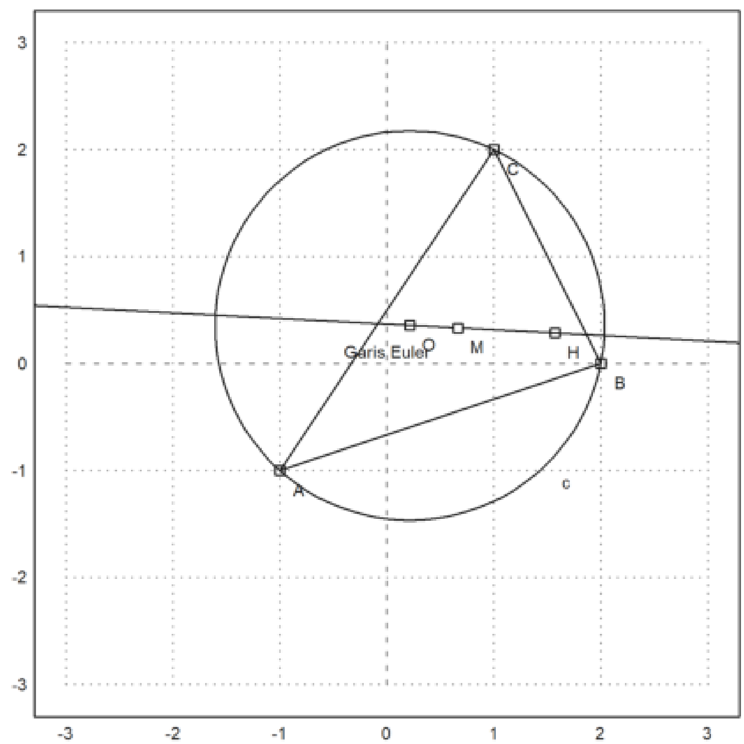


Pusat gravitasi harus berada di garis ini.

```
>M &= (A+B+C)/3; $getLineEquation(el,x,y) with [x=M[1],y=M[2]]
```

$$-\frac{1}{2} = -\frac{1}{2}$$

```
>plotPoint(M(),"M"): // titik berat
```



Teorinya memberitahu kita MH=2*MO. Kita perlu menyederhanakan dengan radcan untuk mencapai ini.

```
>$distance(M,H)/distance(M,O)|radcan
```

$$2$$

Fungsi termasuk fungsi untuk sudut juga.

```
>$computeAngle(A,C,B), degprint(%())
```

$$\arccos\left(\frac{4}{\sqrt{5}\,\sqrt{13}}\right)$$

```
60°15'18.43''
```

Persamaan untuk pusat incircle tidak terlalu bagus.

```
>Q &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))|radcan; $
```

$$\left[\frac{\left(2^{\frac{3}{2}}+1\right)\,\sqrt{5}\,\sqrt{13}-15\,\sqrt{2}+3}{14},\frac{\left(\sqrt{2}-3\right)\,\sqrt{5}\,\sqrt{13}+5\,2^{\frac{3}{2}}+5}{14}\right]$$

Mari kita hitung juga ekspresi untuk jari-jari lingkaran yang tertulis.

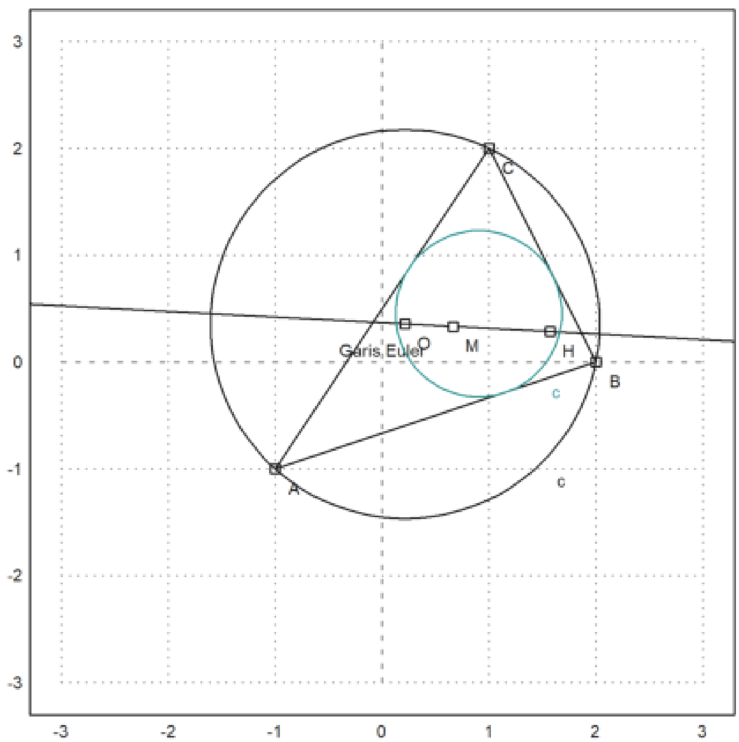
```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r
```

$$\frac{\sqrt{\left(-41\,\sqrt{2}-31\right)\,\sqrt{5}\,\sqrt{13}+115\,\sqrt{2}+614}}{7\,\sqrt{2}}$$

```
>LD &=  circleWithCenter(Q,r); // Lingkaran dalam
```

Mari kita tambahkan ini ke plot.

```
>color(5); plotCircle(LD()):
```



Parabola

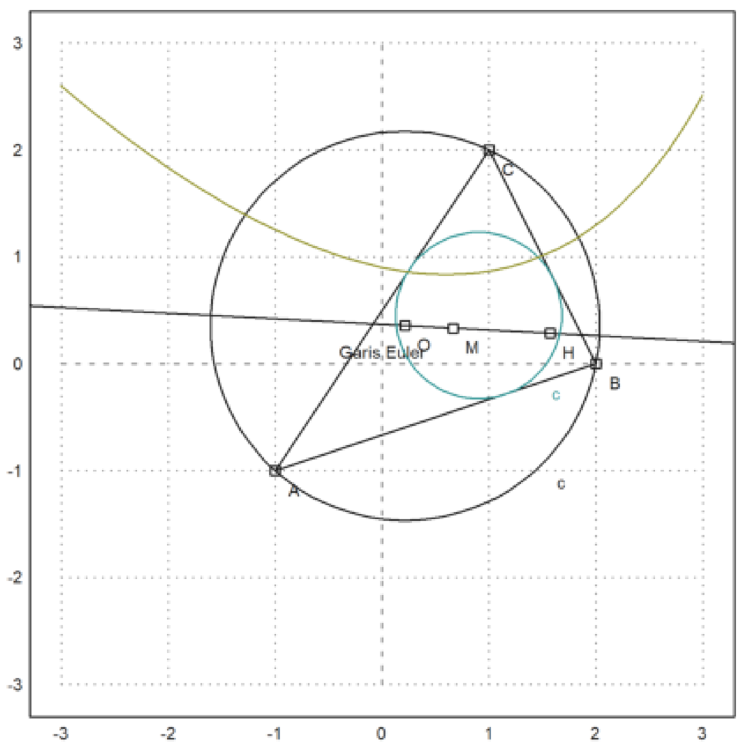
Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
```

$$\frac{3\,y-x+2}{\sqrt{10}}-\sqrt{\left(2-y\right)^2+\left(1-x\right)^2}=0$$

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

```
>plot2d(p,level=0,add=1,contourcolor=6):
```



Ini seharusnya menjadi beberapa fungsi, tetapi pemecah default Maxima hanya dapat menemukan solusinya, jika kita kuadratkan persamaannya. Akibatnya, kami mendapatkan solusi palsu.

```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-distance([x,y],C)^2,y
```

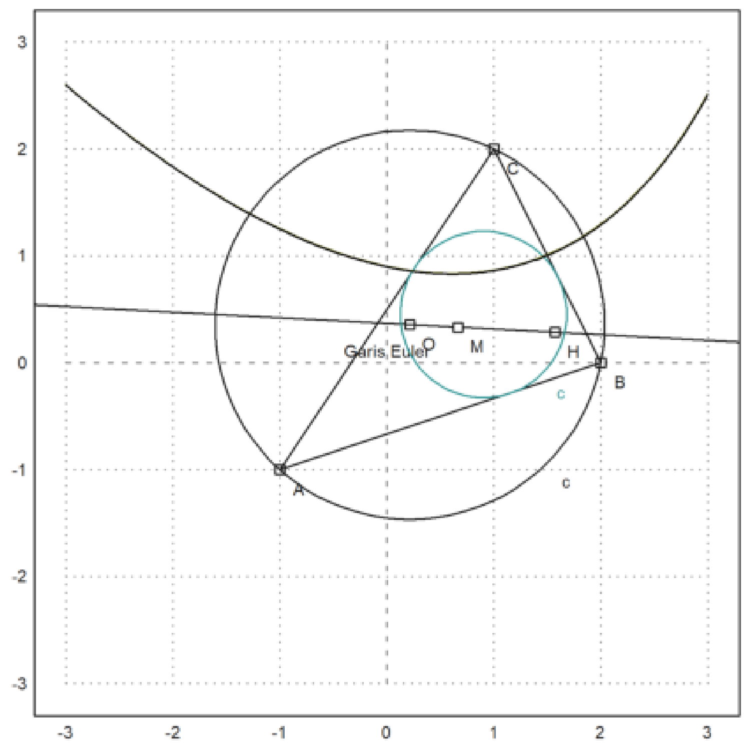
```
[y = - 3 x - sqrt(70) sqrt(9 - 2 x) + 26,
                              y = - 3 x + sqrt(70) sqrt(9 - 2 x) + 26]
```

Solusi pertama adalah

$$akar_1$$

Menambahkan solusi pertama ke plot menunjukkan, bahwa itu memang jalan yang kita cari. Teorinya memberi tahu kita bahwa itu adalah parabola yang diputar.

```
>plot2d(&rhs(akar[1]),add=1):
```




```
>function g(x) &= rhs(akar[1]); $'g(x)= g(x)// fungsi yang mendefinisikan k
```

$$g\left(x\right)=-3\,x-\sqrt{70}\,\sqrt{9-2\,x}+26$$

```
>T &=[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut
>dTC &= distance(T,C); $fullratsimp(dTC), $float(%) // jarak T ke C
```

$$2.135605779339061$$

```
>U &= projectToLine(T,lineThrough(A,B)); $U // proyeksi T pada garis AB
```

$$\left[\frac{80-3\sqrt{11}\sqrt{70}}{10},\frac{20-\sqrt{11}\sqrt{70}}{10}\right]$$

```
>dU2AB &= distance(T,U); $fullratsimp(dU2AB), $float(%) // jatak T ke AB
```

$$2.135605779339061$$

Ternyata jarak T ke C sama dengan jarak T ke AB. Coba Anda pilih titik T yang lain dan ulangi perhitungan-perhitungan di atas untuk menunjukkan bahwa hasilnya juga sama.

Contoh 5: Trigonometri Rasional

Ini terinspirasi dari ceramah N.J.Wildberger. Dalam bukunya "Divine Proportions", Wildberger mengusulkan untuk mengganti pengertian klasik tentang jarak dan sudut dengan kuadrat dan penyebaran. Dengan menggunakan ini, memang mungkin untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional".

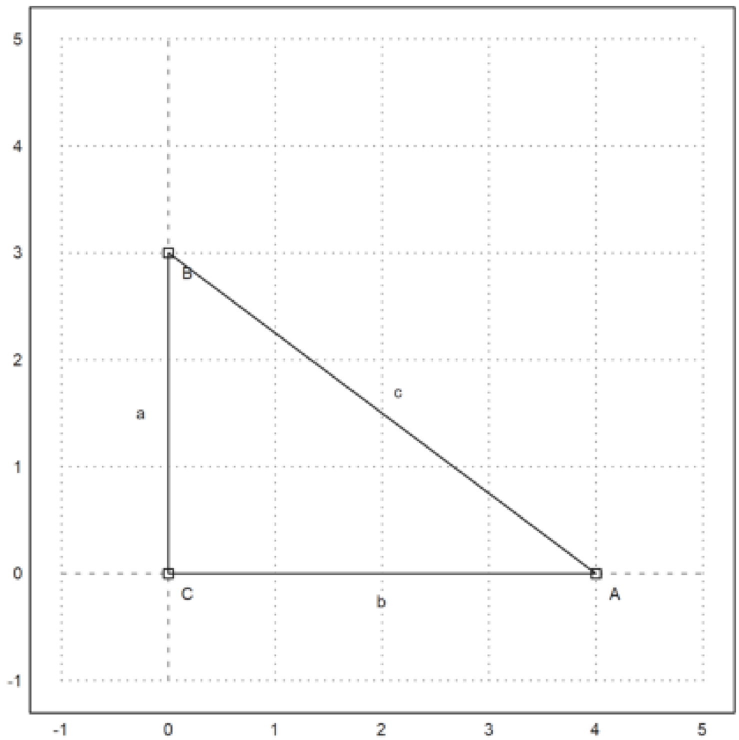
Berikut ini, saya memperkenalkan konsep, dan memecahkan beberapa masalah. Saya menggunakan perhitungan simbolik Maxima di sini, yang menyembunyikan keuntungan utama dari trigonometri rasional bahwa perhitungan hanya dapat dilakukan dengan kertas dan pensil. Anda diundang untuk memeriksa hasil tanpa komputer.

Intinya adalah bahwa perhitungan rasional simbolis sering kali menghasilkan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang hanya mengevaluasi perkiraan numerik.

```
>load geometry;
```

Untuk pengenalan pertama, kami menggunakan segitiga persegi panjang dengan proporsi Mesir terkenal 3, 4 dan 5. Perintah berikut adalah perintah Euler untuk merencanakan geometri bidang yang terdapat dalam file Euler "geometry.e".

```
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...
>setPlotRange(-1,5,-1,5); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg(30);
```



Tentu saja,

$$\sin(w_a) = \frac{a}{c},$$

di mana wa adalah sudut di A. Cara biasa untuk menghitung sudut ini, adalah dengan mengambil kebalikan dari fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicerna, yang hanya dapat dicetak kira-kira.

```
>wa := arcsin(3/5); degprint(wa)
```

```
36°52'11.63''
```

Trigonometri rasional mencoba menghindari hal ini.

Gagasan pertama trigonometri rasional adalah kuadran, yang menggantikan jarak. Sebenarnya, itu hanya jarak kuadrat. Berikut ini, a, b, dan c menunjukkan kuadrat dari sisi-sisinya.

Teorema Pythogoras menjadi a+b=c.

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
```

```
25 = 25
```

Pengertian kedua dari trigonometri rasional adalah penyebaran. Spread mengukur pembukaan antar baris. Ini adalah 0, jika garis-garisnya sejajar, dan 1, jika garis-garisnya persegi panjang. Ini adalah kuadrat sinus sudut antara dua garis.

Penyebaran garis AB dan AC pada gambar di atas didefinisikan sebagai

$$s_a = \sin(\alpha)^2 = \frac{a}{c},$$

di mana a dan c adalah kuadrat dari sembarang segitiga siku-siku dengan salah satu sudut di A.

```
>sa &= a/c; $sa
```

$$\frac{9}{25}$$

Ini lebih mudah dihitung daripada sudut, tentu saja. Tetapi Anda kehilangan properti bahwa sudut dapat ditambahkan dengan mudah.

Tentu saja, kita dapat mengonversi nilai perkiraan untuk sudut wa menjadi sprad, dan mencetaknya sebagai pecahan.

```
>fracprint(sin(wa)^2)
```

```
9/25
```

Hukum kosinus trgonometri klasik diterjemahkan menjadi "hukum silang" berikut.

$$(c + b - a)^2 = 4bc\,(1 - s_a)$$

Di sini a, b, dan c adalah kuadrat dari sisi-sisi segitiga, dan sa adalah penyebaran sudut A. Sisi a, seperti biasa, berhadapan dengan sudut A.

Hukum ini diimplementasikan dalam file geometri.e yang kami muat ke Euler.

```
>$crosslaw(aa,bb,cc,saa)
```

$$(cc + bb - aa)^2 = 4\,bb\,cc\,(1 - saa)$$

Dalam kasus kami, kami mendapatkan

```
>$crosslaw(a,b,c,sa)
```

$$1024 = 1024$$

Mari kita gunakan crosslaw ini untuk mencari spread di A. Untuk melakukan ini, kita buat crosslaw untuk kuadran a, b, dan c, dan selesaikan untuk spread yang tidak diketahui sa.

Anda dapat melakukannya dengan tangan dengan mudah, tetapi saya menggunakan Maxima. Tentu saja, kami mendapatkan hasilnya, kami sudah memilikinya.

```
>$crosslaw(a,b,c,x), $solve(%,x)
```

$$\left[x = \frac{9}{25}\right]$$

$$\left[x = \frac{9}{25}\right]$$

Kita sudah tahu ini. Definisi spread adalah kasus khusus dari crosslaw.

Kita juga dapat menyelesaikan ini untuk umum a,b,c. Hasilnya adalah rumus yang menghitung penyebaran sudut segitiga yang diberikan kuadrat dari ketiga sisinya.

```
>$solve(crosslaw(aa,bb,cc,x),x)
```

$$\left[x = \frac{-cc^2 - (-2\,bb - 2\,aa)\,cc - bb^2 + 2\,aa\,bb - aa^2}{4\,bb\,cc}\right]$$

Kita bisa membuat fungsi dari hasilnya. Fungsi seperti itu sudah didefinisikan dalam file geometri.e dari Euler.

```
>$spread(a,b,c)
```

$$\frac{9}{25}$$

Sebagai contoh, kita dapat menggunakannya untuk menghitung sudut segitiga dengan sisi

$$a, \quad a, \quad \frac{4a}{7}$$

Hasilnya rasional, yang tidak begitu mudah didapat jika kita menggunakan trigonometri klasik.

```
>$spread(a,a,4*a/7)
```

$$\frac{6}{7}$$

Ini adalah sudut dalam derajat.

```
>degprint(arcsin(sqrt(6/7)))
```

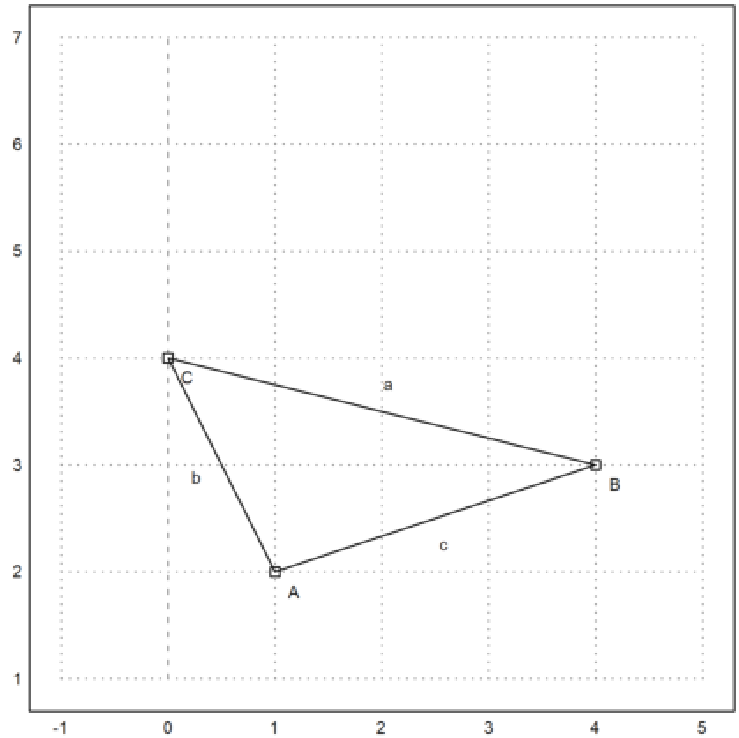
```
67°47'32.44''
```

Contoh lain

Sekarang, mari kita coba contoh yang lebih maju.

Kami mengatur tiga sudut segitiga sebagai berikut.

```
>A&:=[1,2]; B&:=[4,3]; C&:=[0,4]; ...
>setPlotRange(-1,5,1,7); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg;
```



Menggunakan Pythogoras, mudah untuk menghitung jarak antara dua titik. Saya pertama kali menggunakan jarak fungsi file Euler untuk geometri. Jarak fungsi menggunakan geometri klasik.

```
>$distance(A,B)
```

$$\sqrt{10}$$

Euler juga mengandung fungsi untuk kuadran antara dua titik.

Dalam contoh berikut, karena c+b bukan a, maka segitiga itu bukan persegi panjang.

```
>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); $a,
```

17

Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle menggunakan metode biasa berdasarkan hasil kali titik dua vektor. Hasilnya adalah beberapa pendekatan floating point.

$$A = < 1, 2 > \quad B = < 4, 3 >, \quad C = < 0, 4 >$$

$$\mathbf{a} = C - B = < -4, 1 >, \quad \mathbf{c} = A - B = < -3, -1 >, \quad \beta = \angle ABC$$

$$\mathbf{a}.\mathbf{c} = |\mathbf{a}|.|\mathbf{c}| \cos\beta$$

$$\cos\angle ABC = \cos\beta = \frac{\mathbf{a}.\mathbf{c}}{|\mathbf{a}|.|\mathbf{c}|} = \frac{12-1}{\sqrt{17}\sqrt{10}} = \frac{11}{\sqrt{17}\sqrt{10}}$$

```
>wb &= computeAngle(A,B,C); $wb, $(wb/pi*180)()
```

$$\arccos\left(\frac{11}{\sqrt{10}\,\sqrt{17}}\right)$$

```
32.4711922908
```

Dengan menggunakan pensil dan kertas, kita dapat melakukan hal yang sama dengan hukum silang. Kami memasukkan kuadran a, b, dan c ke dalam hukum silang dan menyelesaikan x.

```
>$crosslaw(a,b,c,x), $solve(%,x), //(b+c-a)^=4b.c(1-x)
```

$$\left[x = \frac{49}{50}\right]$$

$$\left[x = \frac{49}{50}\right]$$

Yaitu, apa yang dilakukan oleh penyebaran fungsi yang didefinisikan dalam "geometry.e".

```
>sb &= spread(b,a,c); $sb
```

$$\frac{49}{170}$$

Maxima mendapatkan hasil yang sama menggunakan trigonometri biasa, jika kita memaksanya. Itu menyelesaikan istilah sin(arccos(...)) menjadi hasil pecahan. Sebagian besar siswa tidak dapat melakukan ini.

```
>$sin(computeAngle(A,B,C))^2
```

$$\frac{49}{170}$$

Setelah kita memiliki spread di B, kita dapat menghitung tinggi ha di sisi a. Ingat bahwa

$$s_b = \frac{h_a}{c}$$

Menurut definisi.

```
>ha &= c*sb; $ha
```

$$\frac{49}{17}$$

Gambar berikut telah dihasilkan dengan program geometri C.a.R., yang dapat menggambar kuadrat dan menyebar.

gambar: (20) Rational_Geometry_CaR.png

Menurut definisi, panjang ha adalah akar kuadrat dari kuadratnya.

```
>$sqrt(ha)
```

$$\frac{7}{\sqrt{17}}$$

Sekarang kita dapat menghitung luas segitiga. Jangan lupa, bahwa kita berhadapan dengan kuadrat!

```
>$sqrt(ha)*sqrt(a)/2
```

$$\frac{7}{2}$$

Rumus determinan biasa menghasilkan hasil yang sama.

```
>$areaTriangle(B,A,C)
```

$$\frac{7}{2}$$

Rumus Heron

Sekarang, mari kita selesaikan masalah ini secara umum!

```
>&remvalue(a,b,c,sb,ha);
```

Pertama kita hitung spread di B untuk segitiga dengan sisi a, b, dan c. Kemudian kita menghitung luas kuadrat ("quadrea"?), faktorkan dengan Maxima, dan kita mendapatkan rumus Heron yang terkenal.

Memang, ini sulit dilakukan dengan pensil dan kertas.

```
>$spread(b^2,c^2,a^2), $factor(%*c^2*a^2/4)
```

$$\frac{(-c+b+a)\,(c-b+a)\,(c+b-a)\,(c+b+a)}{16}$$

$$\frac{(-c+b+a)\,(c-b+a)\,(c+b-a)\,(c+b+a)}{16}$$

Aturan Triple Spread

Kerugian dari spread adalah mereka tidak lagi hanya menambahkan sudut yang sama.

Namun, tiga spread dari sebuah segitiga memenuhi aturan "triple spread" berikut.

```
>&remvalue(sa,sb,sc); $triplespread(sa,sb,sc)
```

$$(sc+sb+sa)^2=2\,\left(sc^2+sb^2+sa^2\right)+4\,sa\,sb\,sc$$

Aturan ini berlaku untuk setiap tiga sudut yang menambah 180 °.

$$\alpha+\beta+\gamma=\pi$$

Sejak menyebar

$$\alpha, \pi - \alpha$$

sama, aturan triple spread juga benar, jika

$$\alpha + \beta = \gamma$$

Karena penyebaran sudut negatif adalah sama, aturan penyebaran rangkap tiga juga berlaku, jika

$$\alpha + \beta + \gamma = 0$$

Misalnya, kita dapat menghitung penyebaran sudut 60°. Ini 3/4. Persamaan memiliki solusi kedua, bagaimanapun, di mana semua spread adalah 0.

```
>$solve(triplespread(x,x,x),x)
```

$$\left[x=\frac{3}{4} , x=0\right]$$

Sebaran 90° jelas 1. Jika dua sudut dijumlahkan menjadi 90°, sebarannya menyelesaikan persamaan sebaran rangkap tiga dengan a,b,1. Dengan perhitungan berikut kita mendapatkan a+b=1.

```
>$triplespread(x,y,1), $solve(%,x)
```

$$\left[x=1-y\right]$$

Karena sebaran 180°-t sama dengan sebaran t, rumus sebaran rangkap tiga juga berlaku, jika satu sudut adalah jumlah atau selisih dua sudut lainnya.

Jadi kita dapat menemukan penyebaran sudut berlipat ganda. Perhatikan bahwa ada dua solusi lagi. Kami membuat ini fungsi.

```
>$solve(triplespread(a,a,x),x), function doublespread(a) &= factor(rhs(%[1]
```

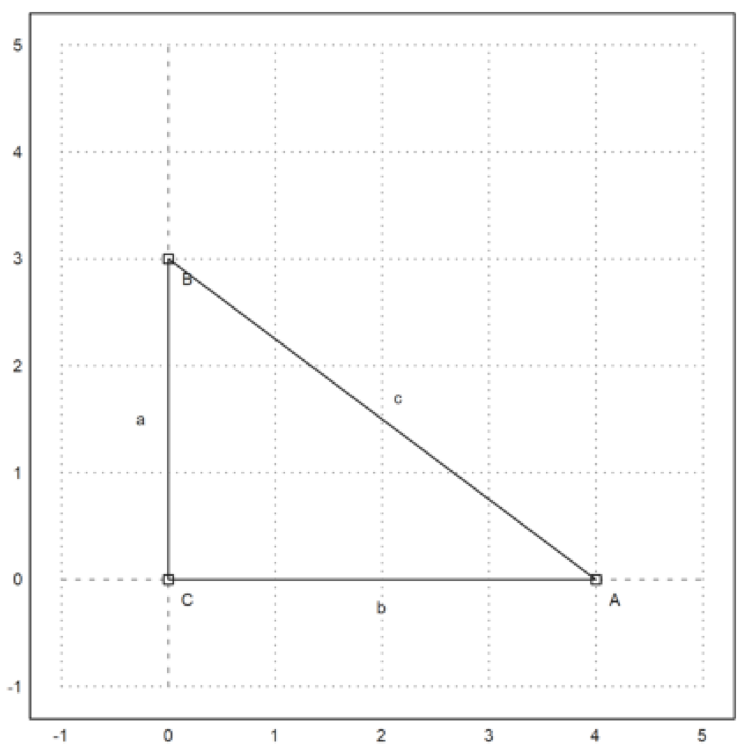
$$\left[x=4\,a-4\,a^2 , x=0\right]$$

```
                              - 4 (a - 1) a
```

Pembagi Sudut

Ini situasinya, kita sudah tahu.

```
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...
>setPlotRange(-1,5,-1,5); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg;
```



Mari kita hitung panjang garis bagi sudut di A. Tetapi kita ingin menyelesaikannya untuk umum a,b,c.

```
>&remvalue(a,b,c);
```

Jadi pertama-tama kita hitung penyebaran sudut yang dibagi dua di A, dengan menggunakan rumus sebaran rangkap tiga.

Masalah dengan rumus ini muncul lagi. Ini memiliki dua solusi. Kita harus memilih yang benar. Solusi lainnya mengacu pada sudut terbelah 180 °-wa.

```
>$triplespread(x,x,a/(a+b)), $solve(%,x), sa2 &= rhs(%[1]); $sa2
```

$$\frac{-\sqrt{b}\,\sqrt{b+a}+b+a}{2\,b+2\,a}$$

$$\left[x=\frac{-\sqrt{b}\,\sqrt{b+a}+b+a}{2\,b+2\,a} , x=\frac{\sqrt{b}\,\sqrt{b+a}+b+a}{2\,b+2\,a}\right]$$

$$\frac{-\sqrt{b}\,\sqrt{b+a}+b+a}{2\,b+2\,a}$$

Mari kita periksa persegi panjang Mesir.

```
>$sa2 with [a=3^2,b=4^2]
```

$$\frac{1}{10}$$

Kami dapat mencetak sudut dalam Euler, setelah mentransfer penyebaran ke radian.

```
>wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

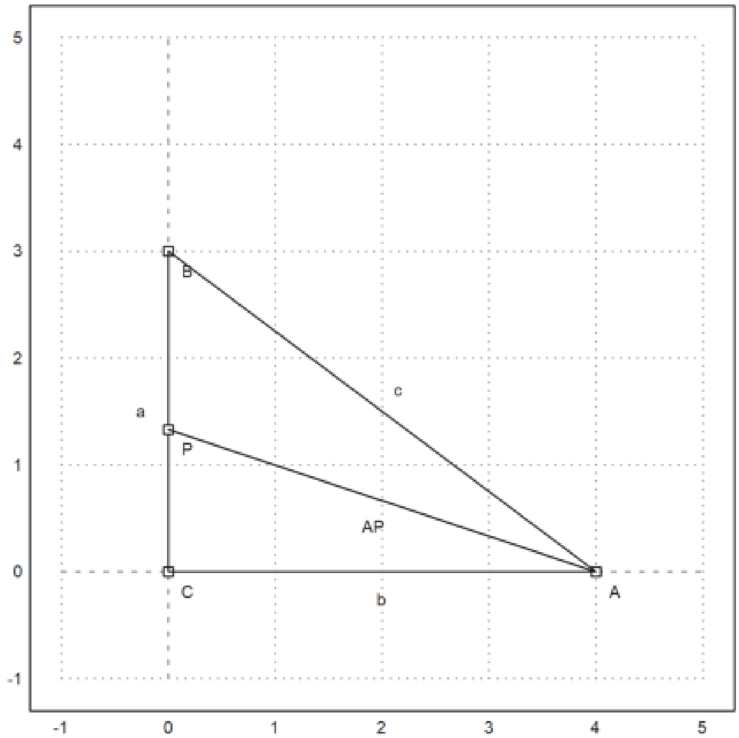
```
18°26'5.82''
```

Titik P adalah perpotongan garis bagi sudut dengan sumbu y.

```
>P := [0,tan(wa2)*4]
```

```
[0,  1.33333]
```

```
>plotPoint(P,"P"); plotSegment(A,P):
```



Mari kita periksa sudut dalam contoh spesifik kita.

```
>computeAngle(C,A,P), computeAngle(P,A,B)
```

```
0.321750554397
0.321750554397
```

Sekarang kita hitung panjang garis bagi AP.

Kami menggunakan teorema sinus dalam segitiga APC. Teorema ini menyatakan bahwa

$$\frac{BC}{\sin(w_a)} = \frac{AC}{\sin(w_b)} = \frac{AB}{\sin(w_c)}$$

berlaku dalam segitiga apa pun. Kuadratkan, itu diterjemahkan ke dalam apa yang disebut "hukum penyebaran"

$$\frac{a}{s_a} = \frac{b}{s_b} = \frac{c}{s_b}$$

di mana a,b,c menunjukkan qudrances.

Karena spread CPA adalah 1-sa2, kita dapatkan darinya bisa/1=b/(1-sa2) dan dapat menghitung bisa (kuadran dari garis-bagi sudut).

```
>&factor(ratsimp(b/(1-sa2))); bisa &= %; $bisa
```

$$\frac{2\,b\,\left(b+a\right)}{\sqrt{b}\,\sqrt{b+a}+b+a}$$

Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

```
>sqrt(mxmeval("at(bisa,[a=3^2,b=4^2])")), distance(A,P)
```

```
4.21637021356
4.21637021356
```

Kita juga dapat menghitung P menggunakan rumus spread.

```
>py&=factor(ratsimp(sa2*bisa)); $py
```

$$-\frac{b\,\left(\sqrt{b}\,\sqrt{b+a}-b-a\right)}{\sqrt{b}\,\sqrt{b+a}+b+a}$$

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

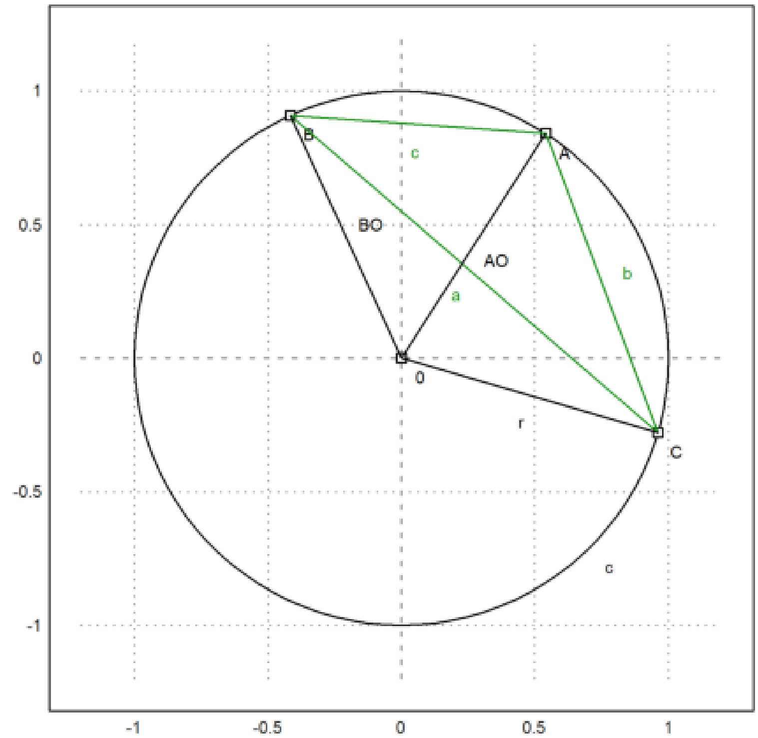
```
>sqrt(mxmeval("at(py,[a=3^2,b=4^2])"))
```

```
1.33333333333
```

Sudut Akord

Perhatikan situasi berikut.

```
>setPlotRange(1.2); ...
>color(1); plotCircle(circleWithCenter([0,0],1)); ...
>A:=[cos(1),sin(1)]; B:=[cos(2),sin(2)]; C:=[cos(6),sin(6)]; ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>color(3); plotSegment(A,B,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a")
>color(1); O:=[0,0];  plotPoint(O,"0"); ...
>plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ...
>insimg;
```



Kita dapat menggunakan Maxima untuk menyelesaikan rumus penyebaran rangkap tiga untuk sudut-sudut di pusat O untuk r. Jadi kita mendapatkan rumus untuk jari-jari kuadrat dari pericircle dalam hal kuadrat dari sisi.

Kali ini, Maxima menghasilkan beberapa nol kompleks, yang kita abaikan.

```
>&remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru
>rabc &= rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),
```

$$-\frac{a\,b\,c}{c^2-2\,b\,c+a\,\left(-2\,c-2\,b\right)+b^2+a^2}$$

Kita dapat menjadikannya sebagai fungsi Euler.

```
>function periradius(a,b,c) &= rabc;
```

Mari kita periksa hasilnya untuk poin A,B,C.

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Jari-jarinya memang 1.

```
>periradius(a,b,c)
```

```
1
```

Faktanya, spread CBA hanya bergantung pada b dan c. Ini adalah teorema sudut chord.

```
>$spread(b,a,c)*rabc | ratsimp
```

$$\frac{b}{4}$$

Sebenarnya spreadnya adalah b/(4r), dan kita melihat bahwa sudut chord dari chord b adalah setengah dari sudut pusat.

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

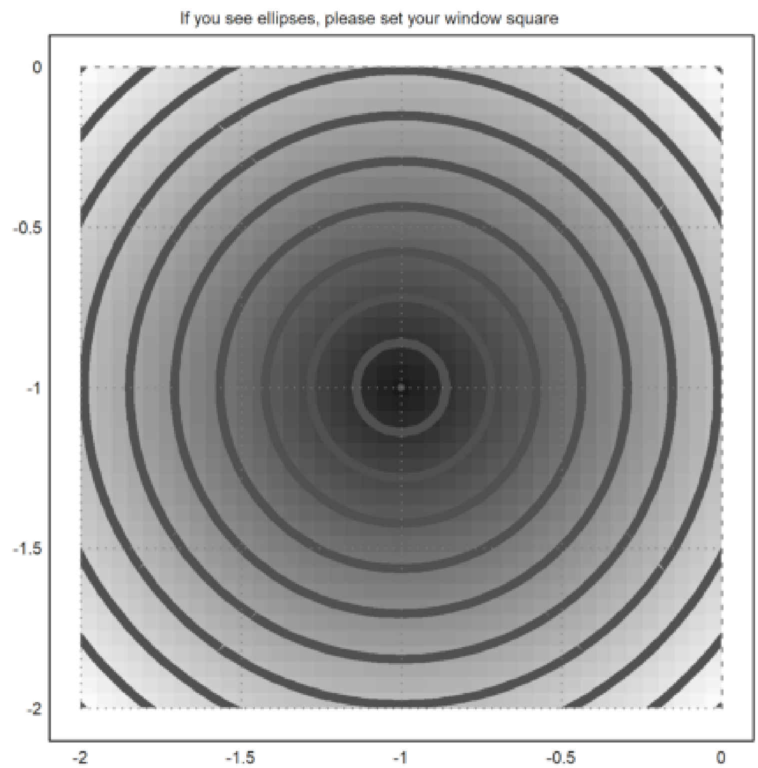
$$0$$

Contoh 6: Jarak Minimal pada Bidang

Catatan awal

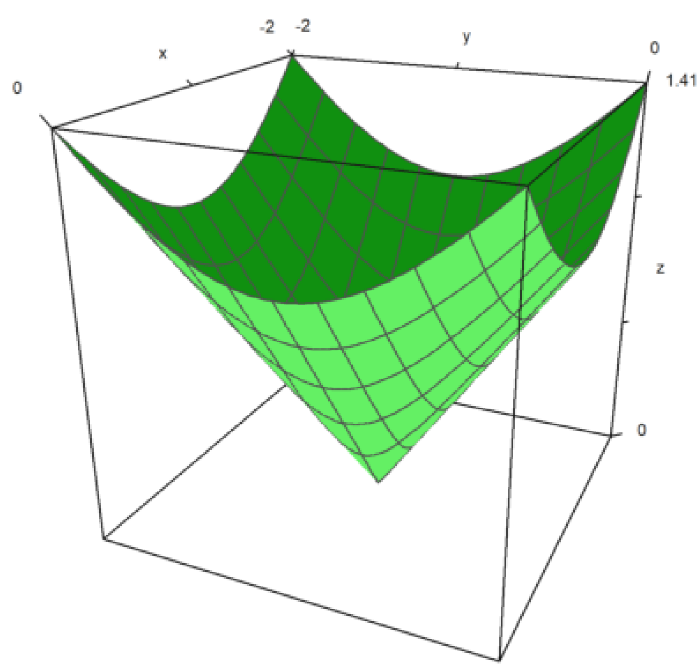
Fungsi yang, ke titik M di bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis level yang agak sederhana: lingkaran berpusat di A.

```
>&remvalue();
>A=[-1,-1];
>function d1(x,y):=sqrt((x-A[1])^2+(y-A[2])^2)
>fcontour("d1",xmin=-2,xmax=0,ymin=-2,ymax=0,hue=1, ...
>title="If you see ellipses, please set your window square"):
```



dan grafiknya juga agak sederhana: bagian atas kerucut:

```
>plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):
```

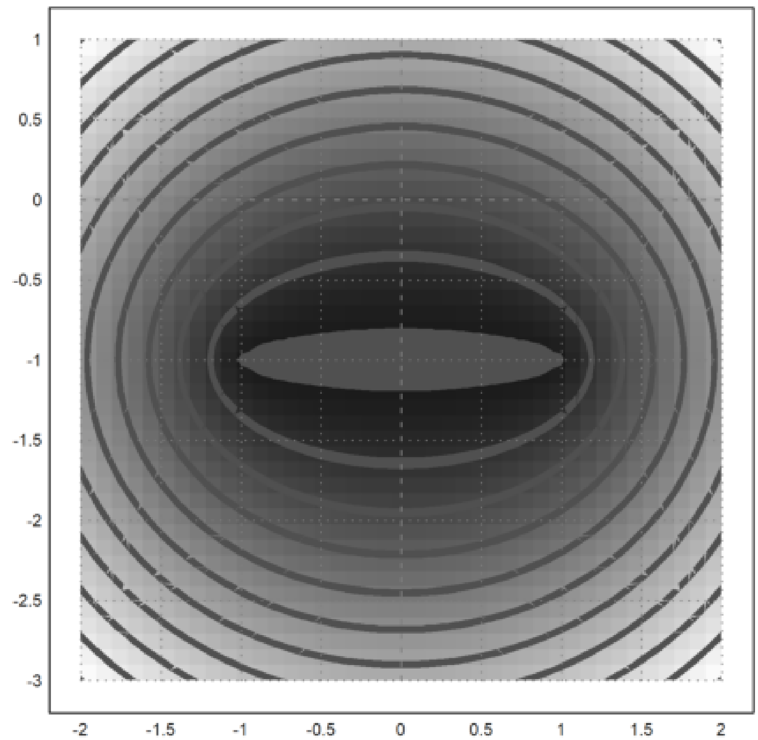


Tentu saja minimal 0 dicapai di A.

Dua poin

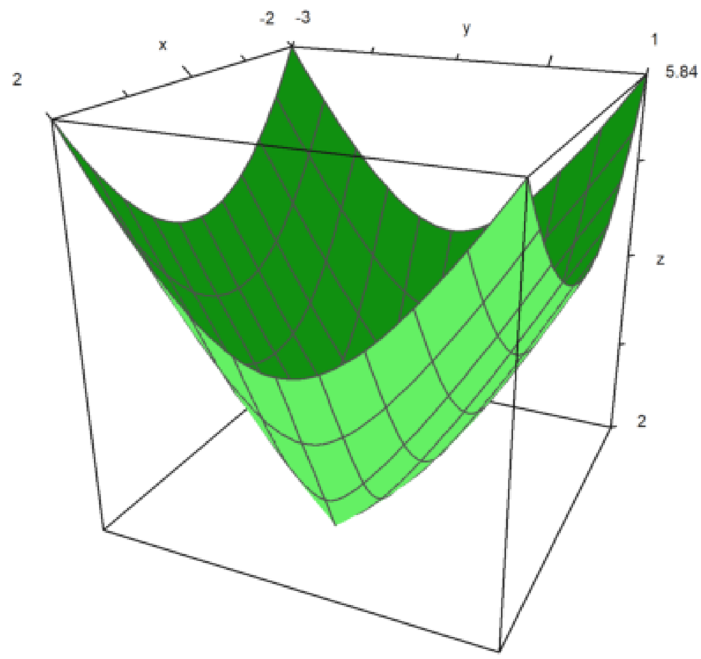
Sekarang kita lihat fungsi MA+MB dimana A dan B adalah dua titik (tetap). Ini adalah "fakta yang diketahui" bahwa kurva level adalah elips, titik fokusnya adalah A dan B; kecuali untuk AB minimum yang konstan pada segmen [AB]:

```
>B=[1,-1];
>function d2(x,y):=d1(x,y)+sqrt((x-B[1])^2+(y-B[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```



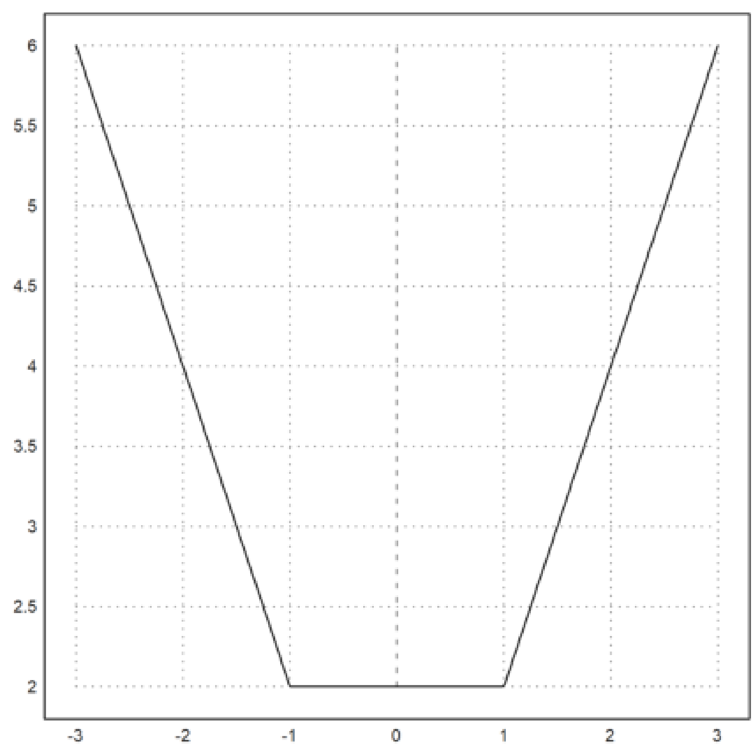
Grafiknya lebih menarik:

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



Pembatasan garis (AB) lebih terkenal:

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```

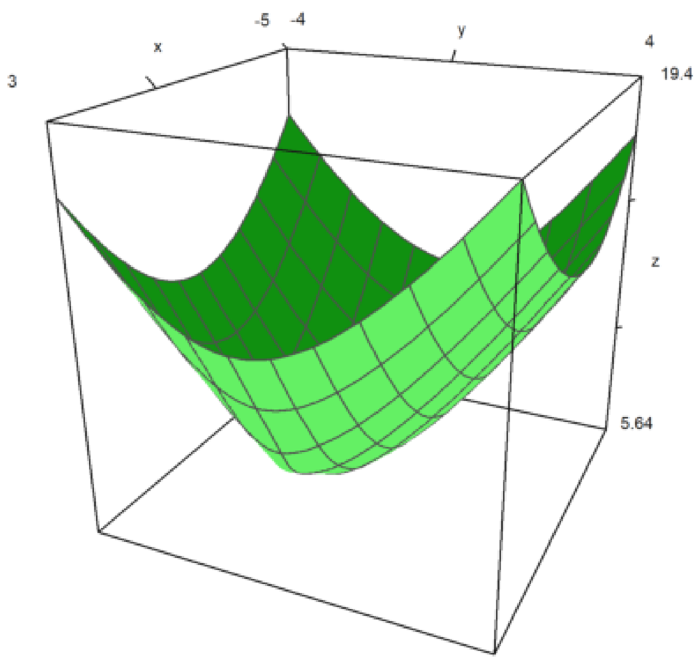


Sekarang hal-hal yang kurang sederhana: Ini sedikit kurang terkenal bahwa MA+MB+MC mencapai minimum pada satu titik pesawat tetapi untuk menentukan itu kurang sederhana:

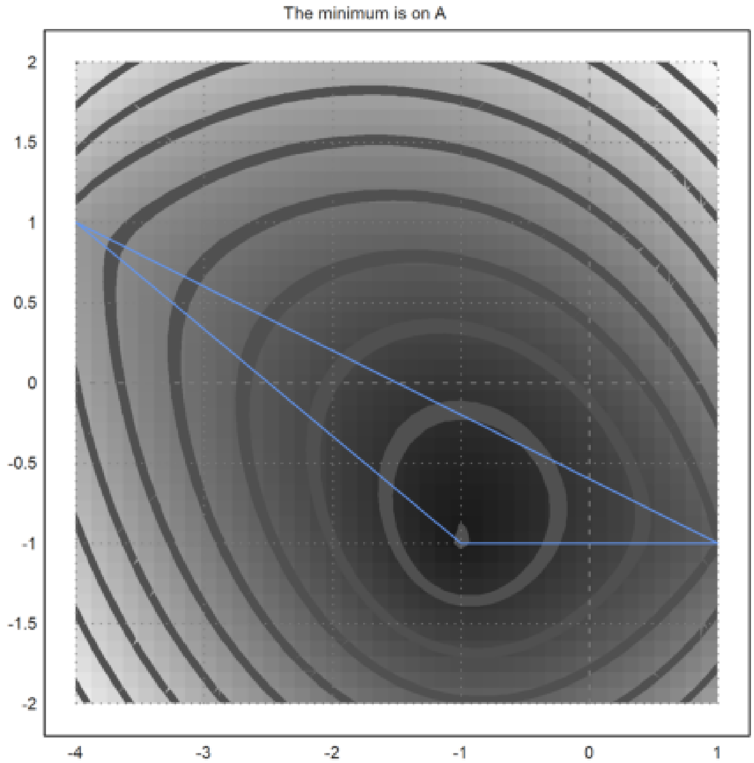
1) Jika salah satu sudut segitiga ABC lebih dari 120° (katakanlah di A), maka minimum dicapai pada titik ini (misalnya AB+AC).

Contoh:

```
>C=[-4,1];
>function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2+(y-C[2])^2)
>plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);
>insimg;
```

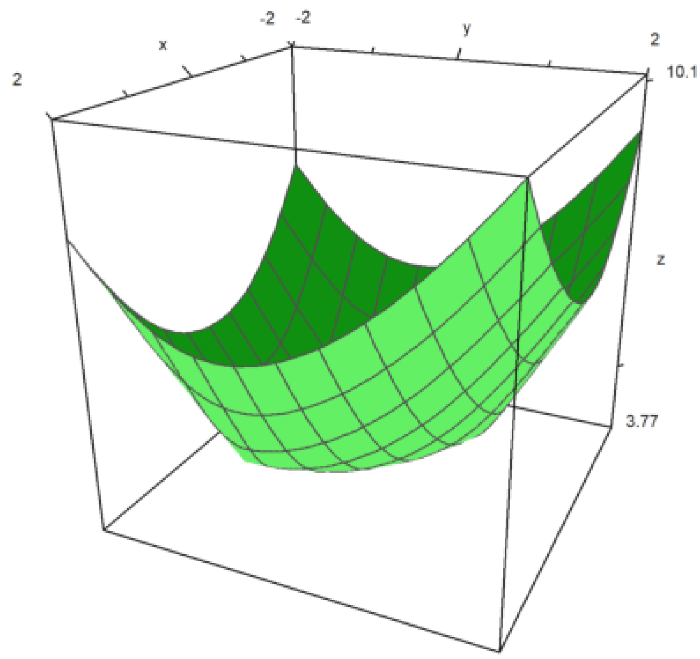



```
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

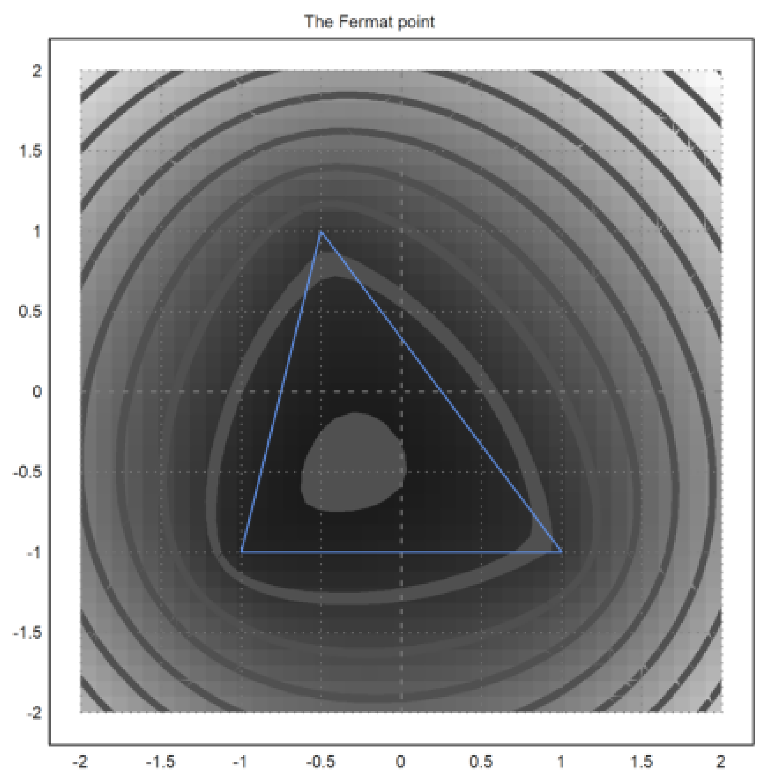


2) Tetapi jika semua sudut segitiga ABC kurang dari 120 °, minimumnya adalah pada titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (maka masing-masing 120 °):

```
>C=[-0.5,1];
>plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2):
```




```
>fcontour("d3",xmin=-2,xmax=2,ymin=-2,ymax=2,hue=1,title="The Fermat point"
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```



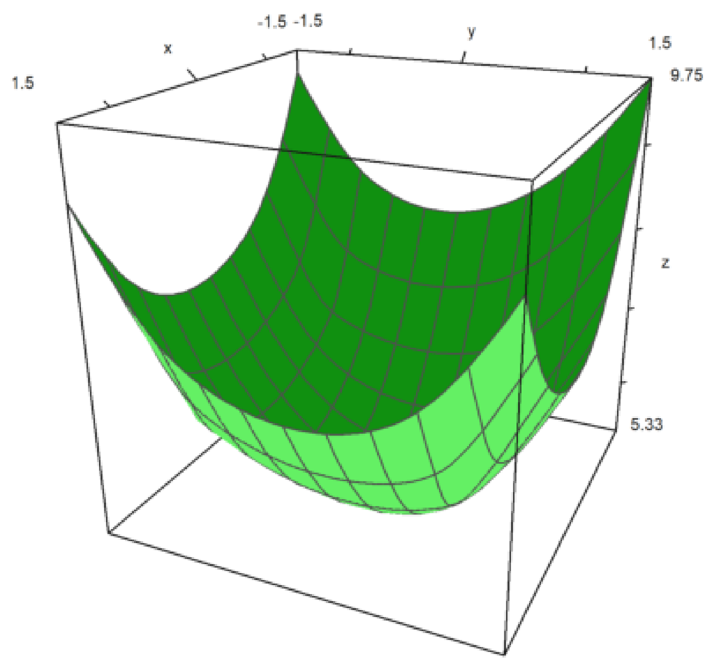
Merupakan kegiatan yang menarik untuk mewujudkan gambar di atas dengan perangkat lunak geometri; misalnya, saya tahu soft yang ditulis di Jawa yang memiliki instruksi "garis kontur" ...

Semua ini di atas telah ditemukan oleh seorang hakim Perancis bernama Pierre de Fermat; dia menulis surat kepada dilettants lain seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di pajak penghasilan. Jadi titik unik F sedemikian rupa sehingga FA+FB+FC minimal, disebut titik Fermat segitiga. Tetapi tampaknya beberapa tahun sebelumnya, Torriccelli Italia telah menemukan titik ini sebelum Fermat melakukannya! Bagaimanapun tradisinya adalah mencatat poin ini F...

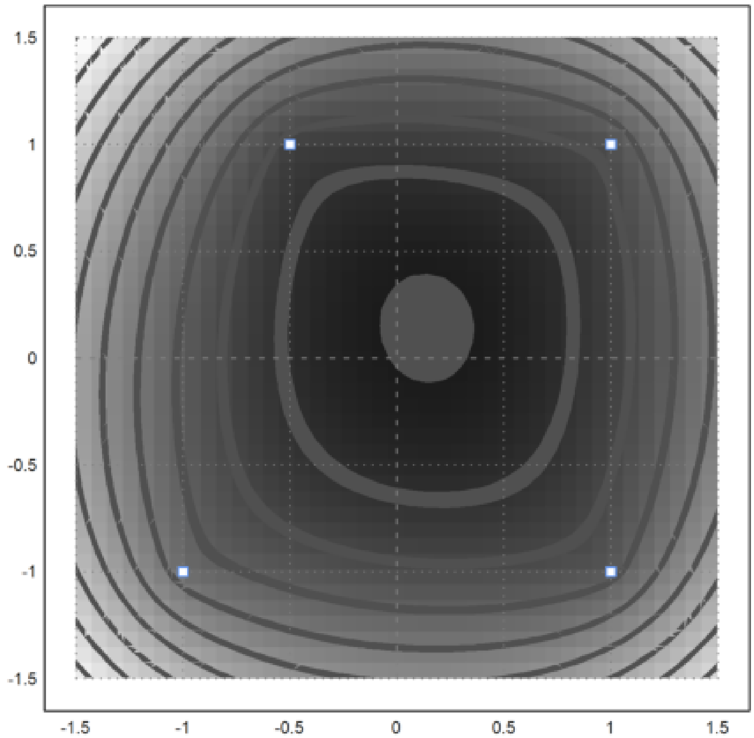
Empat poin

Langkah selanjutnya adalah menambahkan 4 titik D dan mencoba meminimalkan MA+MB+MC+MD; katakan bahwa Anda adalah operator TV kabel dan ingin mencari di bidang mana Anda harus meletakkan antena sehingga Anda dapat memberi makan empat desa dan menggunakan panjang kabel sesedikit mungkin!

```
>D=[1,1];
>function d4(x,y):=d3(x,y)+sqrt((x-D[1])^2+(y-D[2])^2)
>plot3d("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5):
```



```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],points=1,add=1,color=12);
>insimg;
```



Masih ada minimum dan tidak tercapai di salah satu simpul A, B, C atau D:

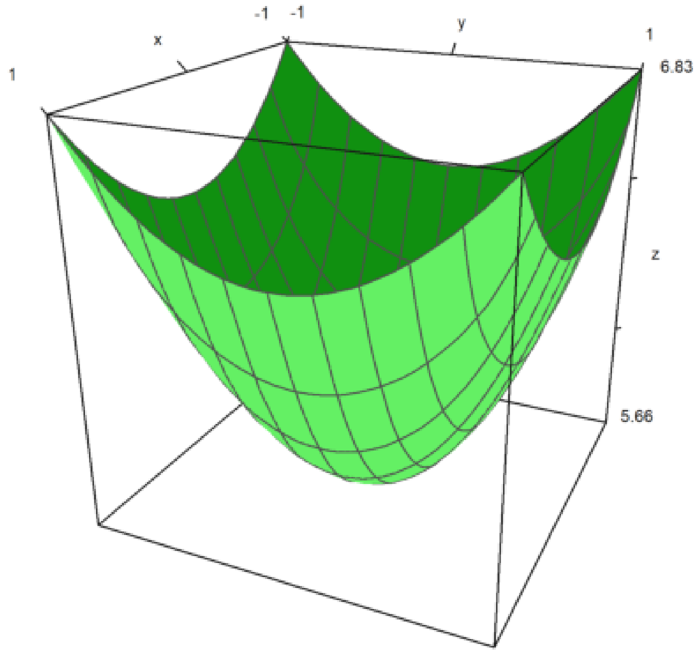
```
>function f(x):=d4(x[1],x[2])
>neldermin("f",[0.2,0.2])
```

```
  [0.142858,   0.142857]
```

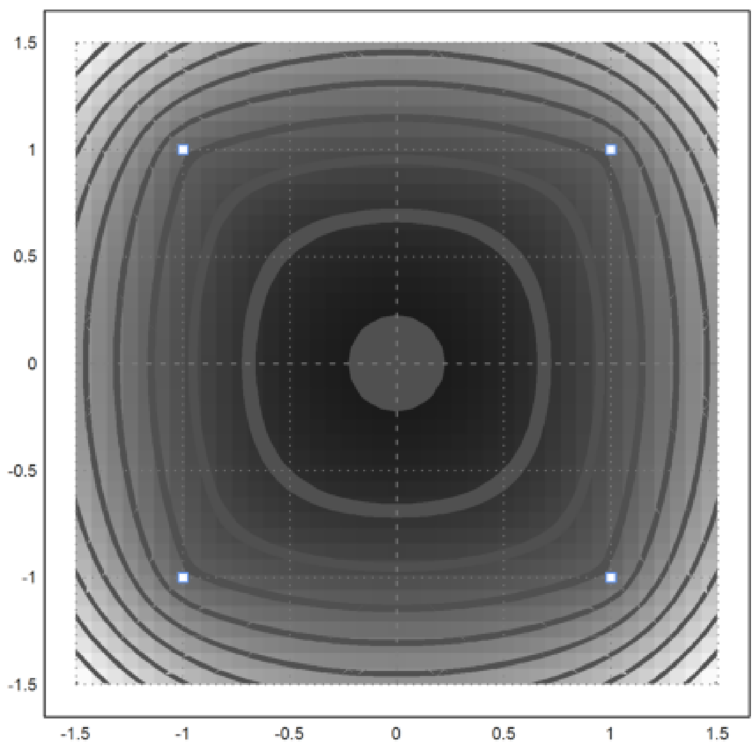
Tampaknya dalam kasus ini, koordinat titik optimal adalah rasional atau mendekati rasional...

Sekarang ABCD adalah persegi, kami berharap bahwa titik optimal akan menjadi pusat ABCD:

```
>C=[-1,1];
>plot3d("d4",xmin=-1,xmax=1,ymin=-1,ymax=1):
```




```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],add=1,color=12,points=1);
>insimg;
```



Contoh 7: Bola Dandelin dengan Povray

Anda dapat menjalankan demonstrasi ini, jika Anda telah menginstal Povray, dan pvengine.exe di jalur program.

Pertama kita hitung jari-jari bola.

Jika Anda melihat gambar di bawah, Anda melihat bahwa kita membutuhkan dua lingkaran yang menyentuh dua garis yang membentuk kerucut, dan satu garis yang membentuk bidang yang memotong kerucut.

Kami menggunakan file geometri.e dari Euler untuk ini.

```
>load geometry;
```

Pertama dua garis yang membentuk kerucut.

```
>g1 &= lineThrough([0,0],[1,a])
```

$$[-a, 1, 0]$$

```
>g2 &= lineThrough([0,0],[-1,a])
```

$$[-a, -1, 0]$$

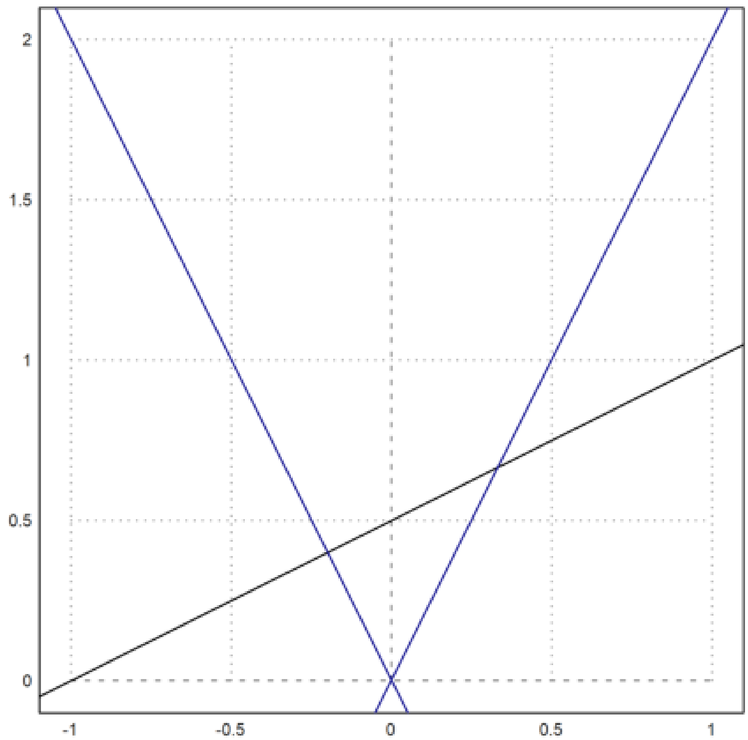
Kemudian baris ketiga.

```
>g &= lineThrough([-1,0],[1,1])
```

$$[-1, 2, 1]$$

Kami merencanakan semuanya sejauh ini.

```
>setPlotRange(-1,1,0,2);
>color(black); plotLine(g(),"")
>a:=2; color(blue); plotLine(g1(),""), plotLine(g2(),""):
```



Sekarang kita ambil titik umum pada sumbu y.

```
>P &= [0,u]
```

```
[0, u]
```

Hitung jarak ke g1.

```
>d1 &= distance(P,projectToLine(P,g1)); $d1
```

$$\sqrt{\left(\frac{a^2\,u}{a^2+1}-u\right)^2+\frac{a^2\,u^2}{\left(a^2+1\right)^2}}$$

Hitung jarak ke g.

```
>d &= distance(P,projectToLine(P,g)); $d
```

$$\sqrt{\left(\frac{u+2}{5}-u\right)^2+\frac{\left(2\,u-1\right)^2}{25}}$$

Dan temukan pusat kedua lingkaran yang jaraknya sama.

```
>sol &= solve(d1^2=d^2,u); $sol
```

$$\left[u=\frac{-\sqrt{5}\,\sqrt{a^2+1}+2\,a^2+2}{4\,a^2-1} , u=\frac{\sqrt{5}\,\sqrt{a^2+1}+2\,a^2+2}{4\,a^2-1}\right]$$

Ada dua solusi.

Kami mengevaluasi solusi simbolis, dan menemukan kedua pusat, dan kedua jarak.

```
>u := sol()
```

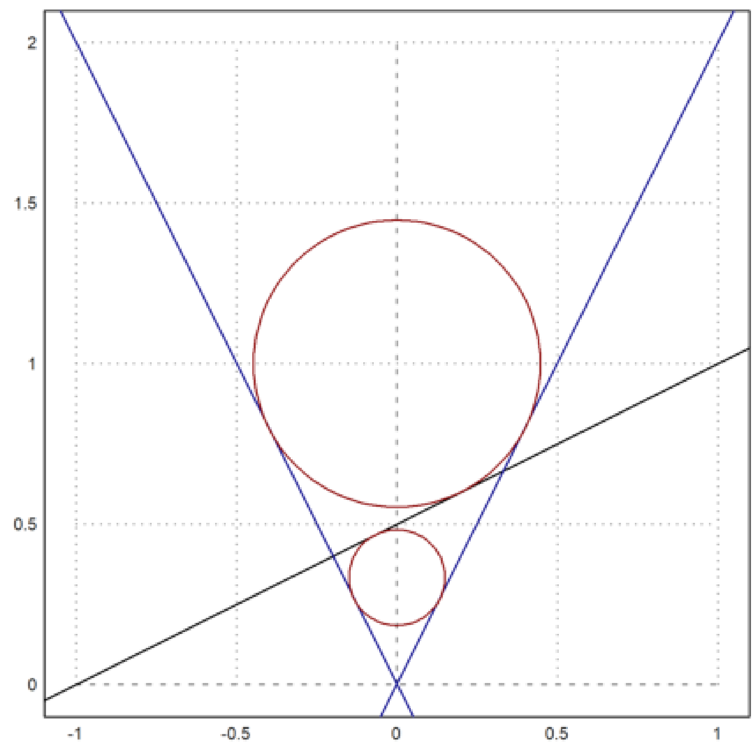
```
[0.333333,  1]
```

```
>dd := d()
```

```
[0.149071,  0.447214]
```

Plot lingkaran ke dalam gambar.

```
>color(red);
>plotCircle(circleWithCenter([0,u[1]],dd[1]),"");
>plotCircle(circleWithCenter([0,u[2]],dd[2]),"");
>insimg;
```



Plot dengan Povray

Selanjutnya kami merencanakan semuanya dengan Povray. Perhatikan bahwa Anda mengubah perintah apa pun dalam urutan perintah Povray berikut, dan menjalankan kembali semua perintah dengan Shift-Return.
Pertama kita memuat fungsi povray.

```
>load povray;
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

Kami mengatur adegan dengan tepat.

```
>povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Selanjutnya kita menulis dua bidang ke file Povray.

```
>writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
>writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
```

Dan kerucutnya, transparan.

```
>writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
```

Kami menghasilkan bidang terbatas pada kerucut.

```
>gp=g();
>pc=povcone([0,0,0],0,[0,0,a],1,"");
>vp=[gp[1],0,gp[2]]; dp=gp[3];
>writeln(povplane(vp,dp,povlook(blue,0.5),pc));
```

Sekarang kita menghasilkan dua titik pada lingkaran, di mana bola menyentuh kerucut.

```
>function turnz(v) := return [-v[2],v[1],v[3]]
>P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
>writeln(povpoint(P1,povlook(yellow)));
>P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
>writeln(povpoint(P2,povlook(yellow)));
```

Kemudian kami menghasilkan dua titik di mana bola menyentuh bidang. Ini adalah fokus dari elips.

```
>P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
>writeln(povpoint(P3,povlook(yellow)));
>P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
>writeln(povpoint(P4,povlook(yellow)));
```

Selanjutnya kita hitung perpotongan P1P2 dengan bidang.

```
>t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
>writeln(povpoint(P5,povlook(yellow)));
```

Kami menghubungkan titik-titik dengan segmen garis.

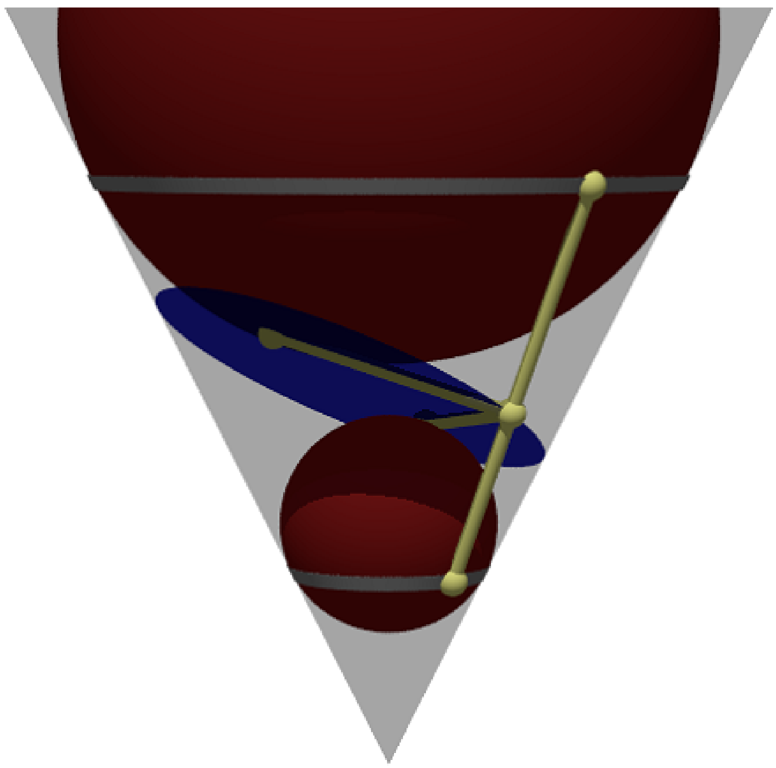
```
>writeln(povsegment(P1,P2,povlook(yellow)));
>writeln(povsegment(P5,P3,povlook(yellow)));
>writeln(povsegment(P5,P4,povlook(yellow)));
```

Sekarang kita menghasilkan pita abu-abu, di mana bola menyentuh kerucut.

```
>pcw=povcone([0,0,0],0,[0,0,a],1.01);
>pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize
>writeln(povintersection([pcw,pc1],povlook(gray)));
>pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize
>writeln(povintersection([pcw,pc2],povlook(gray)));
```

Mulai program Povray.

```
>povend();
```



Untuk mendapatkan Anaglyph ini kita perlu memasukkan semuanya ke dalam fungsi scene. Fungsi ini akan digunakan dua kali kemudian.

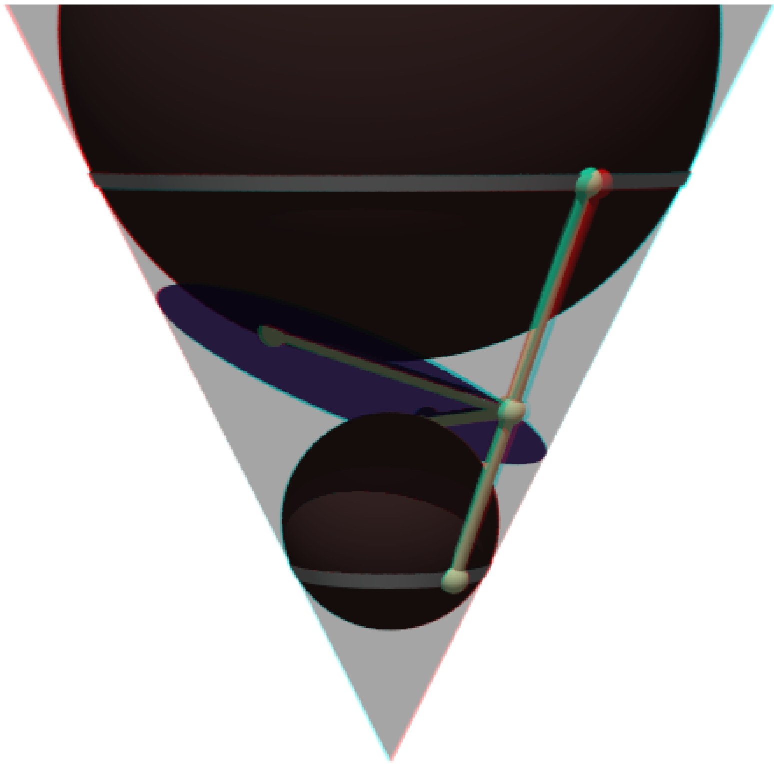
```
>function scene () ...
```

```
  global a,u,dd,g,g1,defaultpointsize;
  writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
  writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
```

```
writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
gp=g();
pc=povcone([0,0,0],0,[0,0,a],1,"");
vp=[gp[1],0,gp[2]]; dp=gp[3];
writeln(povplane(vp,dp,povlook(blue,0.5),pc));
P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
writeln(povpoint(P1,povlook(yellow)));
P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
writeln(povpoint(P2,povlook(yellow)));
P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
writeln(povpoint(P3,povlook(yellow)));
P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
writeln(povpoint(P4,povlook(yellow)));
t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
writeln(povpoint(P5,povlook(yellow)));
writeln(povsegment(P1,P2,povlook(yellow)));
writeln(povsegment(P5,P3,povlook(yellow)));
writeln(povsegment(P5,P4,povlook(yellow)));
pcw=povcone([0,0,0],0,[0,0,a],1.01);
pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsiz
writeln(povintersection([pcw,pc1],povlook(gray)));
pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsiz
writeln(povintersection([pcw,pc2],povlook(gray)));
endfunction
```

Anda membutuhkan kacamata merah/sian untuk menghargai efek berikut.

```
>povanaglyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```



Contoh 8: Geometri Bumi

Dalam buku catatan ini, kami ingin melakukan beberapa perhitungan sferis. Fungsi-fungsi tersebut terdapat dalam file "spherical.e" di folder contoh. Kita perlu memuat file itu terlebih dahulu.

```
>load "spherical.e";
```

Untuk memasukkan posisi geografis, kami menggunakan vektor dengan dua koordinat dalam radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut koordinat Kampus FMIPA UNY.

```
>FMIPA=[rad(-7,-46.467),rad(110,23.05)]
```

```
[-0.13569,  1.92657]
```

Anda dapat mencetak posisi ini dengan sposprint (cetak posisi spherical).

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY
```

```
S 7°46.467' E 110°23.050'
```

Mari kita tambahkan dua kota lagi, Solo dan Semarang.

```
>Solo=[rad(-7,-34.333),rad(110,49.683)]; Semarang=[rad(-6,-59.05),rad(110,2
>sposprint(Solo), sposprint(Semarang),
```

```
S 7°34.333' E 110°49.683'
S 6°59.050' E 110°24.533'
```

Pertama kita menghitung vektor dari satu ke yang lain pada bola ideal. Vektor ini [pos,jarak] dalam radian. Untuk menghitung jarak di bumi, kita kalikan dengan jari-jari bumi pada garis lintang 7°.

```
>br=svector(FMIPA,Solo); degprint(br[1]), br[2]*rearth(7°)->km // perkiraan
```

```
65°20'26.60''
53.8945384608
```

Ini adalah perkiraan yang baik. Rutinitas berikut menggunakan perkiraan yang lebih baik. Pada jarak yang begitu pendek hasilnya hampir sama.

```
>esdist(FMIPA,Semarang)->" km" // perkiraan jarak FMIPA-Semarang
```

```
Commands must be separated by semicolon or comma!
Found:  // perkiraan jarak FMIPA-Semarang (character 32)
You can disable this in the Options menu.
Error in:
esdist(FMIPA,Semarang)->" km" // perkiraan jarak FMIPA-Semaran ...
                             ^
```

Ada fungsi untuk heading, dengan mempertimbangkan bentuk elips bumi. Sekali lagi, kami mencetak dengan cara yang canggih.

```
>sdegprint(esdir(FMIPA,Solo))
```

```
     65.34°
```

Sudut segitiga melebihi 180° pada bola.

```
>asum=sangle(Solo,FMIPA,Semarang)+sangle(FMIPA,Solo,Semarang)+sangle(FMIPA,
```

```
180°0'10.77''
```

Ini dapat digunakan untuk menghitung luas segitiga. Catatan: Untuk segitiga kecil, ini tidak akurat karena kesalahan pengurangan dalam asum-pi.

```
>(asum-pi)*rearth(48°)^2->" km^2" // perkiraan luas segitiga FMIPA-Solo-Sem
```

```
Commands must be separated by semicolon or comma!
Found:  // perkiraan luas segitiga FMIPA-Solo-Semarang (character 32)
You can disable this in the Options menu.
Error in:
(asum-pi)*rearth(48°)^2->" km^2" // perkiraan luas segitiga FM ...
                                ^
```

Ada fungsi untuk ini, yang menggunakan garis lintang rata-rata segitiga untuk menghitung jari-jari bumi, dan menangani kesalahan pembulatan untuk segitiga yang sangat kecil.

```
>esarea(Solo,FMIPA,Semarang)->" km^2", //perkiraan yang sama dengan fungsi
```

```
2123.64310526 km^2
```

Kita juga dapat menambahkan vektor ke posisi. Sebuah vektor berisi heading dan jarak, keduanya dalam radian. Untuk mendapatkan vektor, kami menggunakan vektor. Untuk menambahkan vektor ke posisi, kami menggunakan vektor sadd.

```
>v=svector(FMIPA,Solo); sposprint(saddvector(FMIPA,v)), sposprint(Solo),
```

```
S 7°34.333' E 110°49.683'
S 7°34.333' E 110°49.683'
```

Fungsi-fungsi ini mengasumsikan bola yang ideal. Hal yang sama di bumi.

```
>sposprint(esadd(FMIPA,esdir(FMIPA,Solo),esdist(FMIPA,Solo))), sposprint(So
```

```
S 7°34.333' E 110°49.683'
S 7°34.333' E 110°49.683'
```

Mari kita beralih ke contoh yang lebih besar, Tugu Jogja dan Monas Jakarta (menggunakan Google Earth untuk mencari koordinatnya).

```
>Tugu=[-7.7833°,110.3661°]; Monas=[-6.175°,106.811944°];
>sposprint(Tugu), sposprint(Monas)
```

```
S 7°46.998' E 110°21.966'
S 6°10.500' E 106°48.717'
```

Menurut Google Earth, jaraknya adalah 429,66 km. Kami mendapatkan pendekatan yang baik.

```
>esdist(Tugu,Monas)->" km" // perkiraan jarak Tugu Jogja - Monas Jakarta
```

```
Commands must be separated by semicolon or comma!
Found:  // perkiraan jarak Tugu Jogja - Monas Jakarta (character 32)
You can disable this in the Options menu.
Error in:
esdist(Tugu,Monas)->" km" // perkiraan jarak Tugu Jogja - Mona ...
                         ^
```

Judulnya sama dengan judul yang dihitung di Google Earth.

```
>degprint(esdir(Tugu,Monas))
```

```
294°17’2.85’’
```

Namun, kita tidak lagi mendapatkan posisi target yang tepat, jika kita menambahkan heading dan jarak ke posisi semula. Hal ini terjadi, karena kita tidak menghitung fungsi invers secara tepat, tetapi mengambil perkiraan jari-jari bumi di sepanjang jalan.

```
>sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

```
S 6°10.500’ E 106°48.717’
```

Namun, kesalahannya tidak besar.

```
>sposprint(Monas),
```

```
S 6°10.500’ E 106°48.717’
```

Tentu kita tidak bisa berlayar dengan tujuan yang sama dari satu tujuan ke tujuan lainnya, jika kita ingin menempuh jalur terpendek. Bayangkan, Anda terbang NE mulai dari titik mana pun di bumi. Kemudian Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti heading yang konstan!

Perhitungan berikut menunjukkan bahwa kami jauh dari tujuan yang benar, jika kami menggunakan pos yang sama selama perjalanan kami.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Sekarang kita tambahkan 10 kali sepersepuluh dari jarak, menggunakan pos ke Monas, kita sampai di Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya jauh.

```
>sposprint(p), skmprint(esdist(p,Monas))
```

```
 S 6°11.250' E 106°48.372'
      1.529km
```

Sebagai contoh lain, mari kita ambil dua titik di bumi pada garis lintang yang sama.

```
>P1=[30°,10°]; P2=[30°,50°];
```

Jalur terpendek dari P1 ke P2 bukanlah lingkaran garis lintang 30°, melainkan jalur terpendek yang dimulai 10° lebih jauh ke utara di P1.

```
>sdegprint(esdir(P1,P2))
```

```
      79.69°
```

Tapi, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi kita harus menyesuaikan arah kita di sepanjang jalan. Untuk tujuan kasar, kami menyesuaikannya pada 1/10 dari total jarak.

```
>p=P1;  dist=esdist(P1,P2); ...
>  loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10); e
```

```
      79.69°
      81.67°
      83.71°
      85.78°
      87.89°
      90.00°
      92.12°
      94.22°
      96.29°
      98.33°
```

Jaraknya tidak tepat, karena kita akan menambahkan sedikit kesalahan, jika kita mengikuti heading yang sama terlalu lama.

```
>skmprint(esdist(p,P2))
```

```
      0.203km
```

Kami mendapatkan perkiraan yang baik, jika kami menyesuaikan pos setelah setiap 1/100 dari total jarak dari Tugu ke Monas.

```
>p=Tugu; dist=esdist(Tugu,Monas); ...
>  loop 1 to 100; p=esadd(p,esdir(p,Monas),dist/100); end;
>skmprint(esdist(p,Monas))
```

```
      0.000km
```

Untuk keperluan navigasi, kita bisa mendapatkan urutan posisi GPS di sepanjang lingkaran besar menuju Monas dengan fungsi navigasi.

```
>load spherical; v=navigate(Tugu,Monas,10); ...
>  loop 1 to rows(v); sposprint(v[#]), end;
```

```
 S 7°46.998' E 110°21.966'
 S 7°37.422' E 110°0.573'
 S 7°27.829' E 109°39.196'
 S 7°18.219' E 109°17.834'
 S 7°8.592' E 108°56.488'
 S 6°58.948' E 108°35.157'
 S 6°49.289' E 108°13.841'
 S 6°39.614' E 107°52.539'
 S 6°29.924' E 107°31.251'
 S 6°20.219' E 107°9.977'
 S 6°10.500' E 106°48.717'
```

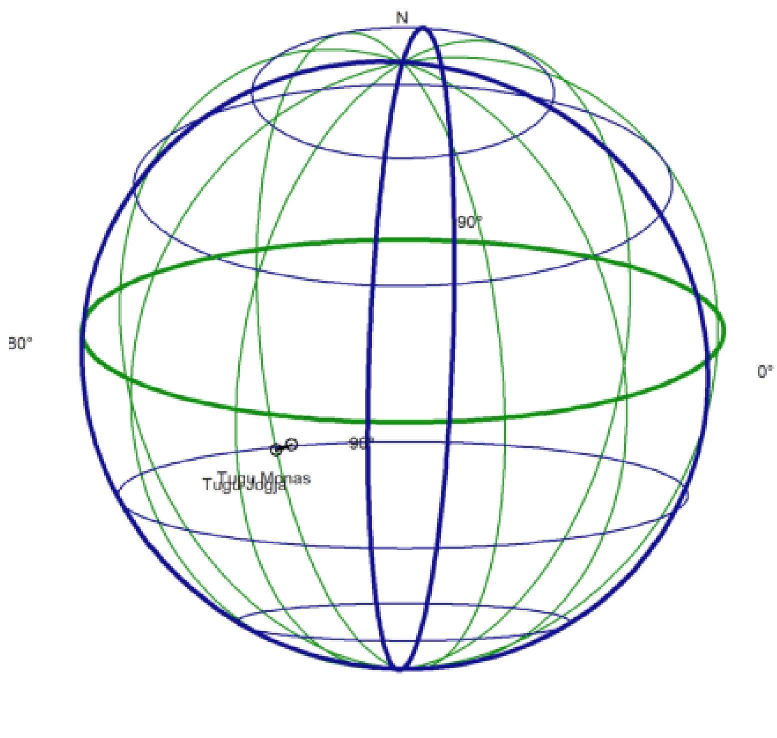
Kami menulis sebuah fungsi, yang memplot bumi, dua posisi, dan posisi di antaranya.

```
>function testplot ...
```

```
 useglobal;
 plotearth;
 plotpos(Tugu,"Tugu Jogja"); plotpos(Monas,"Tugu Monas");
 plotposline(v);
 endfunction
```

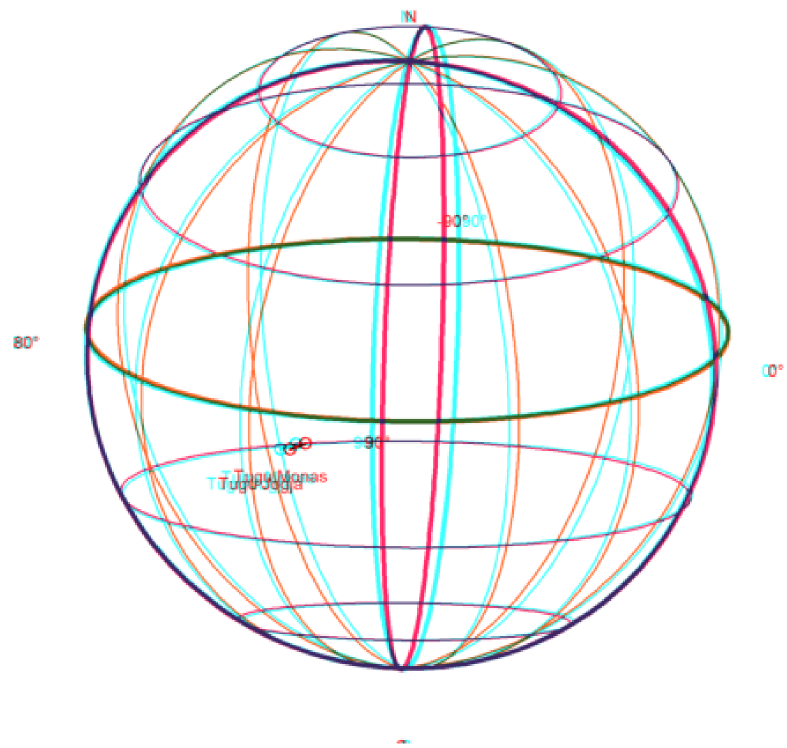
Sekarang plot semuanya.

```
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=4):
```



Atau gunakan plot3d untuk mendapatkan tampilan anaglyph. Ini terlihat sangat bagus dengan kacamata merah/sian.

```
>plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4):
```



```
>reset;
```

Latihan

1. Gambarlah segi-n beraturan jika diketahui titik pusat O, n, dan jarak titik pusat ke titik-titik sudut segi-n tersebut (jari-jari lingkaran luar segi-n), r.

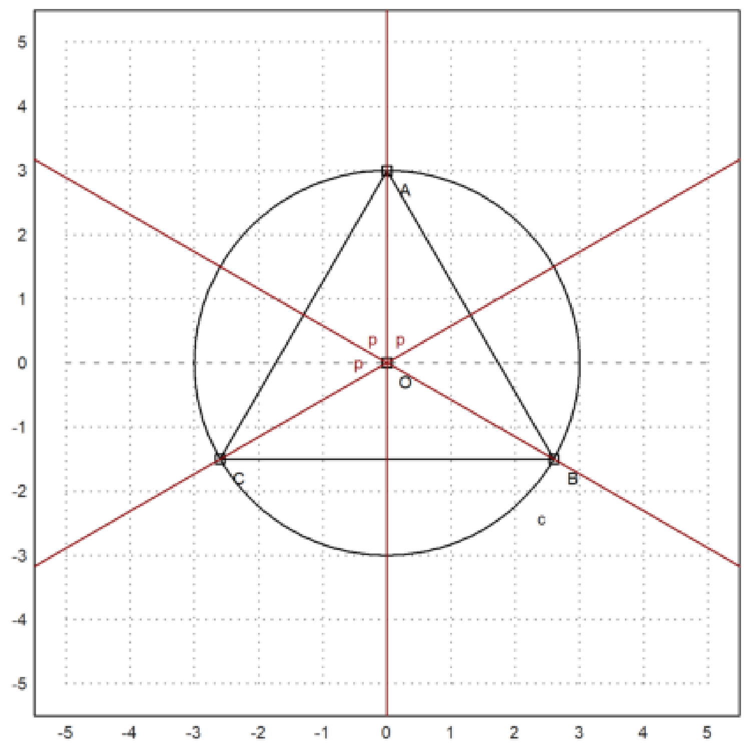
Petunjuk:

- Besar sudut pusat yang menghadap masing-masing sisi segi-n adalah (360/n).
- Titik-titik sudut segi-n merupakan perpotongan lingkaran luar segi-n dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar kelipatan (360/n).
- Untuk n ganjil, pilih salah satu titik sudut adalah di atas.
- Untuk n genap, pilih 2 titik di kanan dan kiri lurus dengan titik pusat.
- Anda dapat menggambar segi-3, 4, 5, 6, 7, dst beraturan.

Jawab :
a) Segitiga

```
>setPlotRange(5);
>O=[0,0]; plotPoint(O,"O")
>A=[0,3]; plotPoint(A,"A");
>B=turn(A,-120°); plotPoint(B,"B");
>C=turn(A,120°); plotPoint(C,"C");
```

```
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(A,C,"");
>c=circleThrough(A,B,C);
>R=getCircleRadius(c);
>plotCircle(c);
>k=angleBisector(C,A,B);
>l=angleBisector(A,C,B);
>m=angleBisector(A,B,C);
>color(red); plotLine(l); plotLine(k); plotLine(m):
```

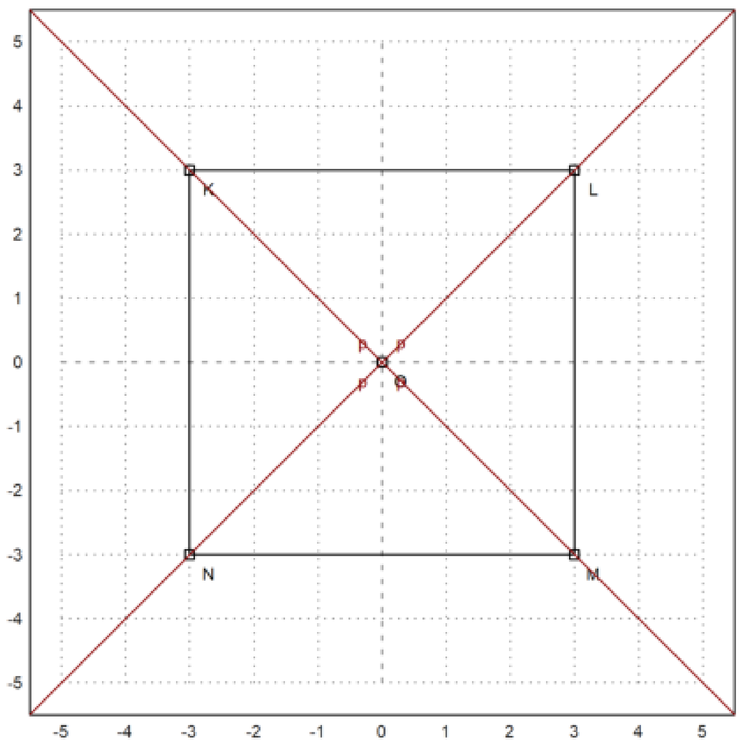



```
>reset;
```

b) Persegi

```
>setPlotRange(5);
>O=[0,0]; plotPoint(O,"O")
>K=[-3,3]; plotPoint(K,"K");
>L=turn(K,-90°); plotPoint(L,"L");
>M=turn(K,-180°); plotPoint(M,"M");
>N=turn(K,-270°); plotPoint(N,"N");
>plotSegment(K,L,""); plotSegment(L,M,""); plotSegment(M,N,""); plotSegment
```

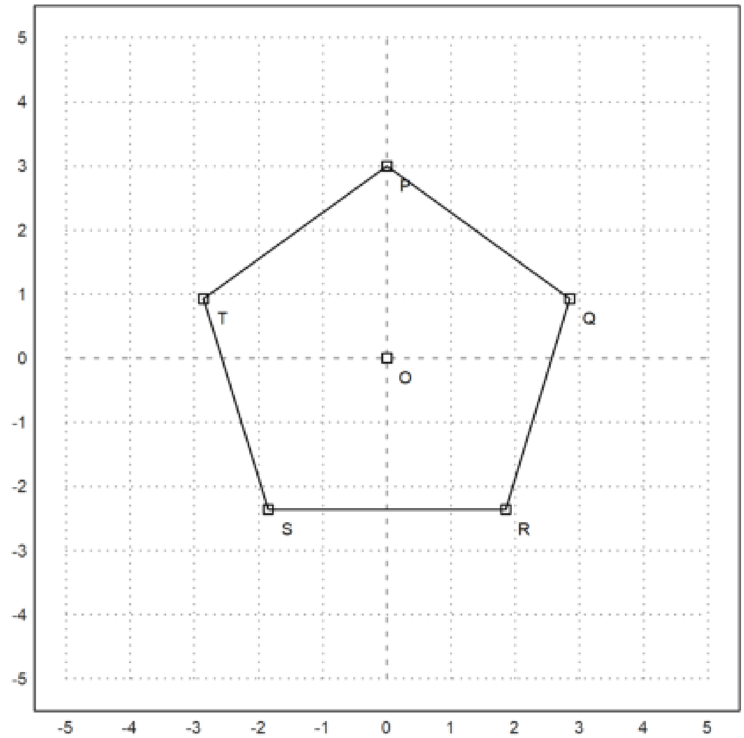
```
>k=angleBisector(K,L,M);
>l=angleBisector(L,M,N);
>m=angleBisector(M,N,K);
>n=angleBisector(N,K,L);
>color(red); plotLine(l); plotLine(k); plotLine(m); plotLine(n):
```




```
>reset;
```

c) Segilima

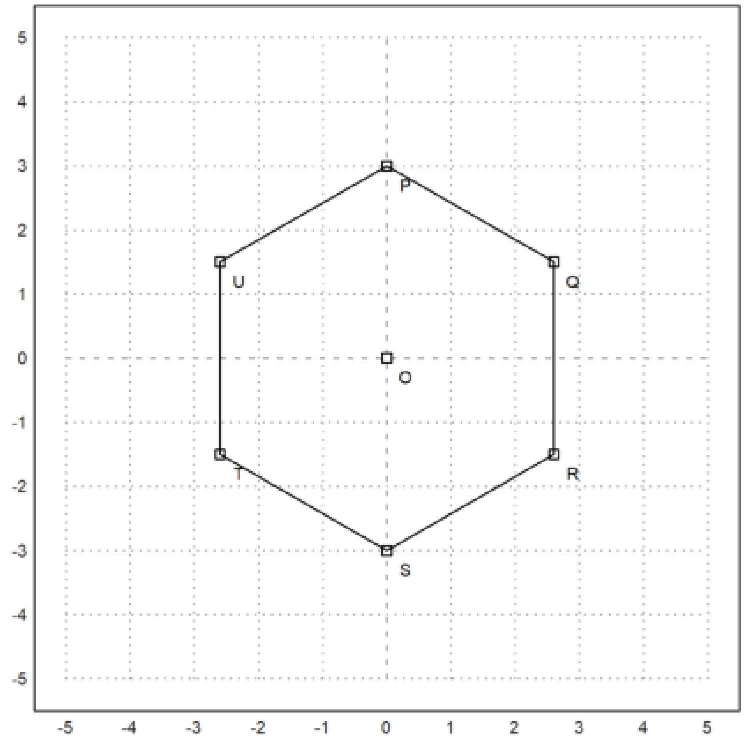
```
>setPlotRange(5);
>O=[0,0]; plotPoint(O,"O")
>P=[0,3]; plotPoint(P,"P");
>Q=turn(P,-72°); plotPoint(Q,"Q");
>R=turn(P,-142°); plotPoint(R,"R");
>S=turn(P,142°); plotPoint(S,"S");
>T=turn(P,72°); plotPoint(T,"T");
>plotSegment(P,Q,""); plotSegment(Q,R,""); plotSegment(R,S,""); plotSegment
```




```
>reset;
```

d) Segienam

```
>setPlotRange(5);
>O=[0,0]; plotPoint(O,"O")
>P=[0,3]; plotPoint(P,"P");
>Q=turn(P,-60°); plotPoint(Q,"Q");
>R=turn(P,-120°); plotPoint(R,"R");
>S=turn(P,-180°); plotPoint(S,"S");
>T=turn(P,-240°); plotPoint(T,"T");
>U=turn(P,-300°); plotPoint(U,"U");
>plotSegment(P,Q,""); plotSegment(Q,R,""); plotSegment(R,S,""); plotSegment
```




```
>reset;
```

2. Gambarlah suatu parabola yang melalui 3 titik yang diketahui.

Petunjuk:
- Misalkan persamaan parabolanya y= ax^2+bx+c.
- Substitusikan koordinat titik-titik yang diketahui ke persamaan tersebut.
- Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai a, b, c.

Jawab :

```
>function f(x) &= a*x^2+b*x+c;
>setPlotRange(5);
>P=[0,4]; Q=[1,5]; R=[3,1];
>plotPoint(P,"P"); plotPoint(Q,"Q"); plotPoint(R,"R");
>&powerdisp:true;
>&f(0)=4
```

$$c = 4$$

```
>&f(1)=5
```

a + b + c = 5

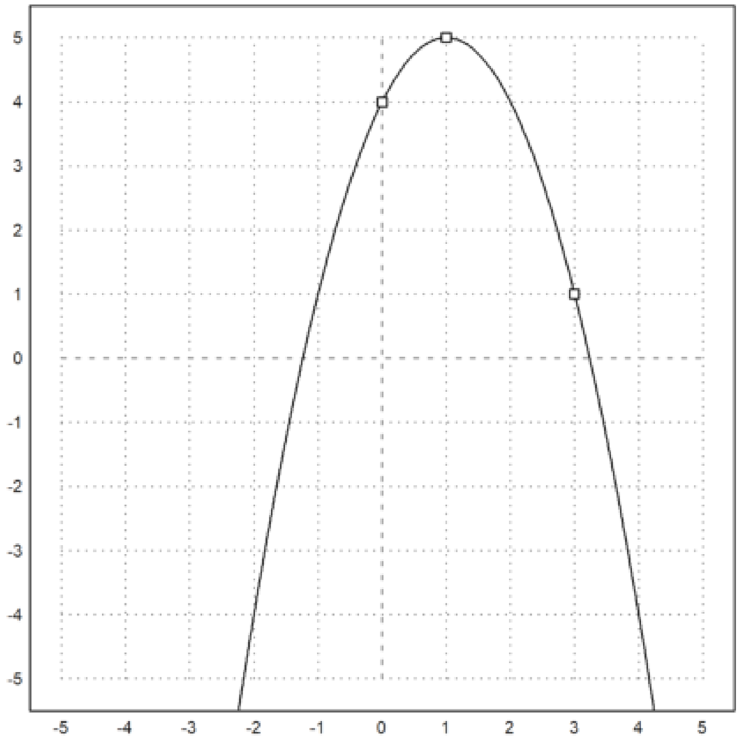
```
>&f(3)=1
```

9 a + 3 b + c = 1

```
>&solve([c=4,a+b+c=5,9*a+3*b+c=1],[a,b,c])
```

[[a = - 1, b = 2, c = 4]]

```
>plot2d("-1*x^2+2*x+4",-5,5,-5,5); plot2d(0,4,>add,>points); plot2d(1,5,>ad
```



3. Gambarlah suatu segi-4 yang diketahui keempat titik sudutnya, misalnya A, B, C, D.

```
- Tentukan apakah segi-4 tersebut merupakan segi-4 garis singgung
```

(sisinya-sisintya merupakan garis singgung lingkaran yang sama yakni lingkaran dalam segi-4 tersebut).

```
- Suatu segi-4 merupakan segi-4 garis singgung apabila keempat
```

garis bagi sudutnya bertemu di satu titik.

```
- Jika segi-4 tersebut merupakan segi-4 garis singgung, gambar
```

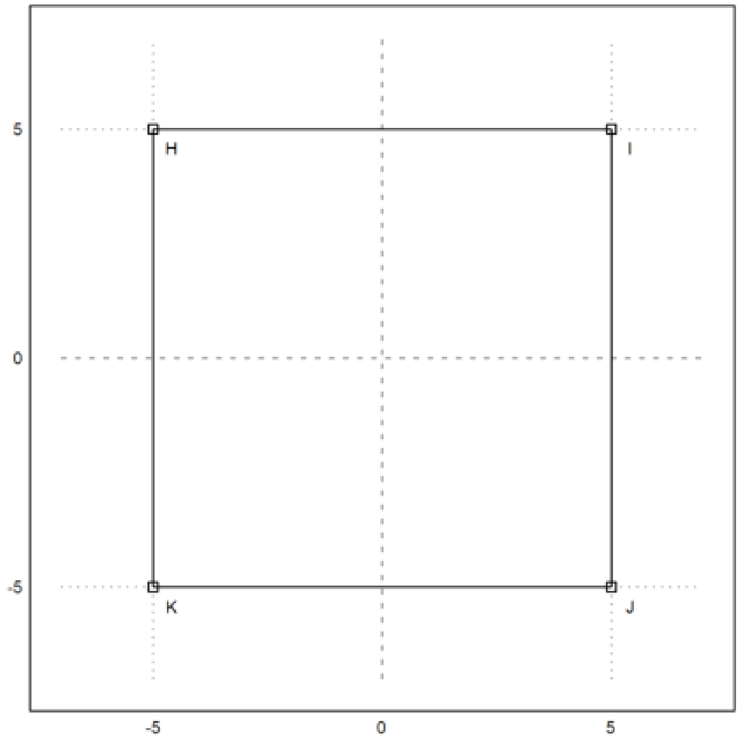
lingkaran dalamnya.

```
- Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis
```

singgung apabila hasil kali panjang sisi-sisi yang berhadapan sama.

Jawab :

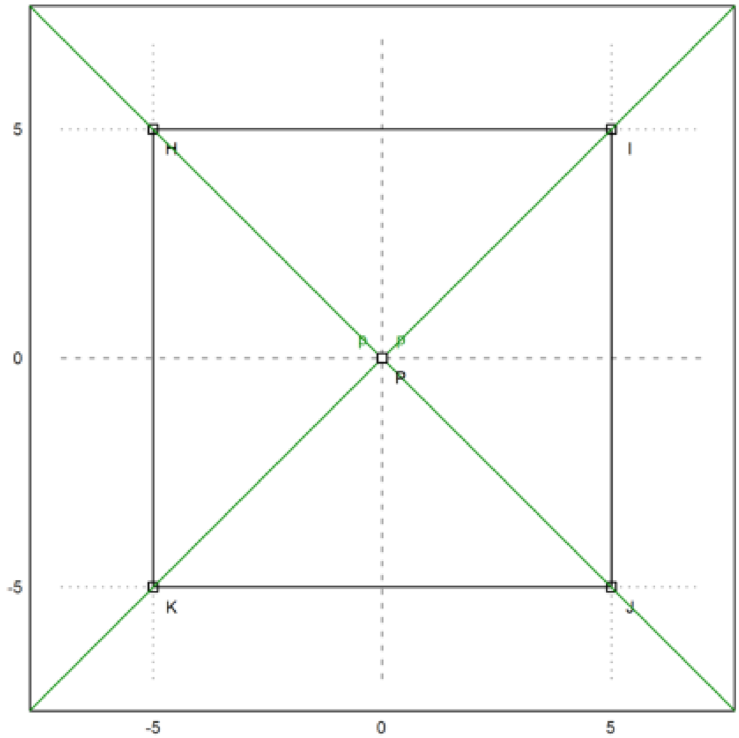
```
>setPlotRange(7);
>H=[-5,5]; I=[5,5]; J=[5,-5]; K=[-5,-5];
>plotPoint(H,"H"); plotPoint(I,"I"); plotPoint(J,"J"); plotPoint(K,"K");
>plotSegment(H,I,""); plotSegment(I,J,""); plotSegment(J,K,""); plotSegment
```




```
>l=angleBisector(I,J,K);
>m=angleBisector(H,I,J);
>P=lineIntersection(l,m)
```

```
[0,   0]
```

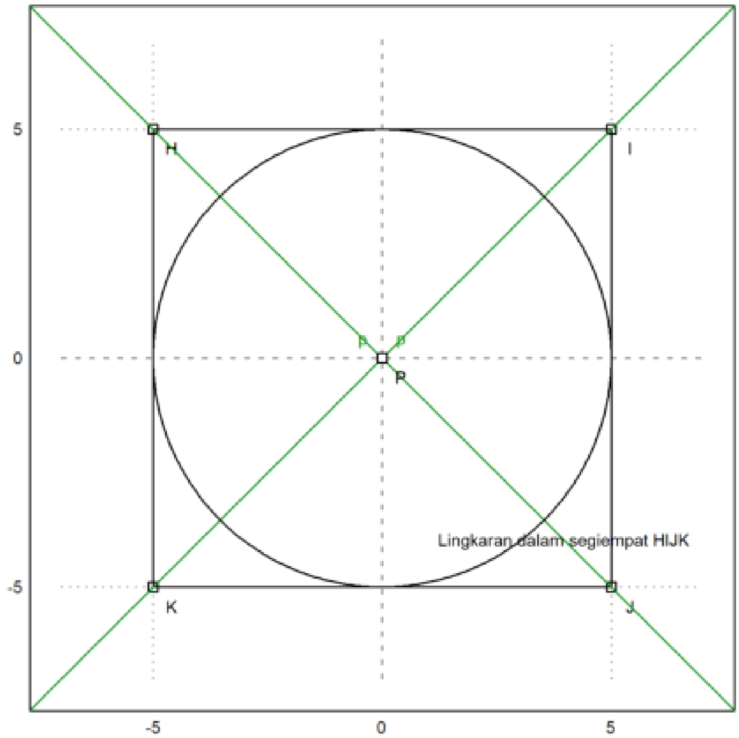
```
>color(green); plotLine(l); plotLine(m); color(1);
>plotPoint(P,"P"):
```




```
>r=norm(P-projectToLine(P,lineThrough(H,I)))
```

5

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segiempat HIJK"):
```




```
>HI=norm(H-I) // panjang sisi HI
```

```
10
```

```
>IJ=norm(I-J) // panjang sisi IJ
```

```
10
```

```
>JK=norm(J-K) // panjang sisi JK
```

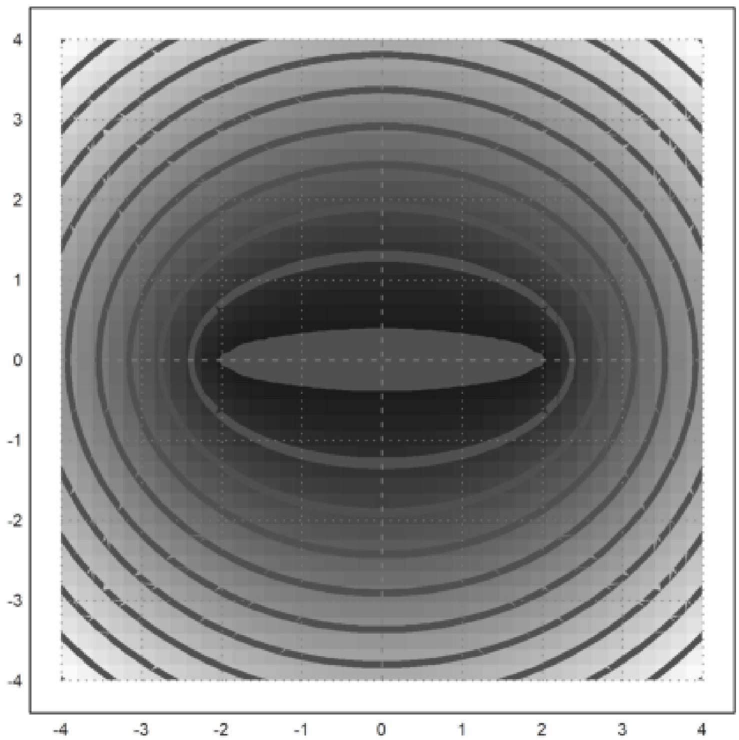
```
10
```

```
>HK=norm(H-K) // panjang sisi HK
```

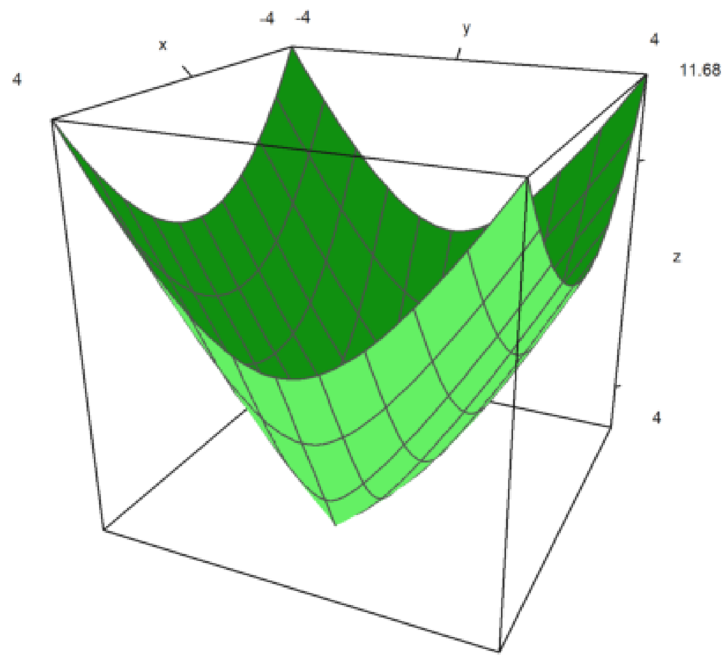
```
10
```

4. Gambarlah suatu ellips jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang jumlah jarak ke P dan ke Q selalu sama (konstan).
Jawab :

```
>P=[-2,0];
>Q=[2,0];
>function d1(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)
>function d2(x,y):=d1(x,y)+sqrt((x-Q[1])^2+(y-Q[2])^2)
>fcontour("d2",-4,4,-4,4,hue=1):
```



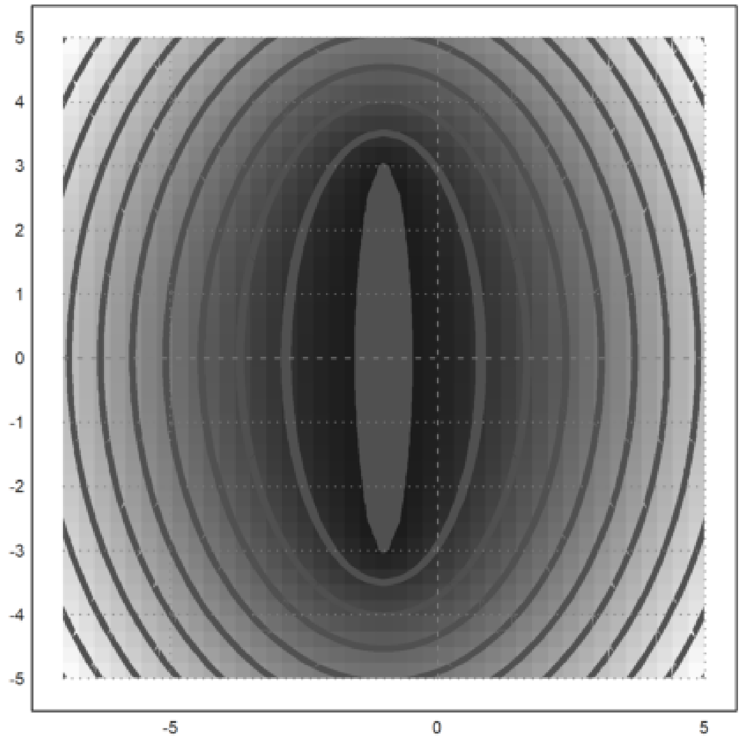

```
>plot3d("d2",-4,4,-4,4):
```



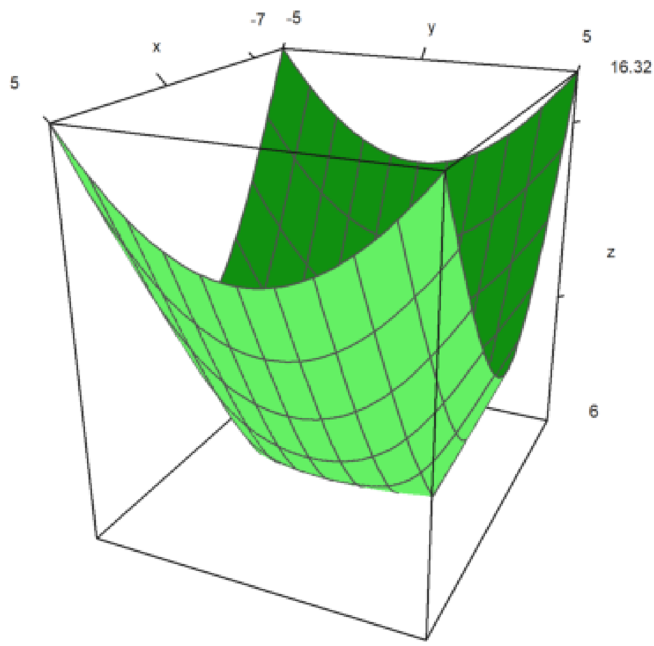

```
>reset;
```

5. Gambarlah suatu hiperbola jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu sama (konstan). Jawab :

```
>P=[-1,3];
>Q=[1,3];
>function d1(x,y):=sqrt((x-p[1])^2+(y-p[2])^2)
>function d2(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)+sqrt((x+Q[1])^2+(y+Q[2])^2)
>fcontour("d2",-7,5,-5,5,hue=1):
```




```
>plot3d("d2",-7,5,-5,5):
```



```
>reset;
```

BAB 7

MENGGUNAKAN EMT UNTUK STATISTIKA

Diagram titik

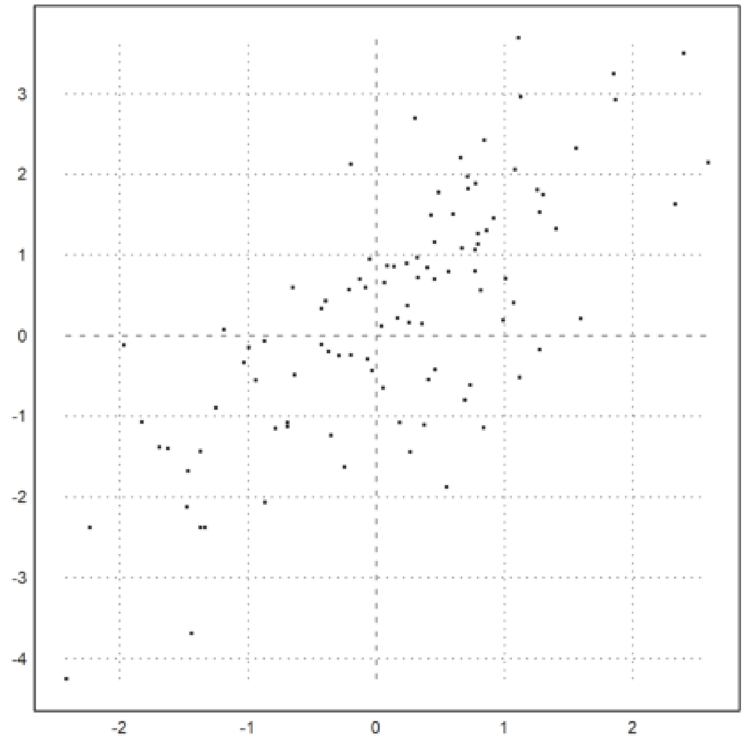
Diagram titik atau disebut juga sebagai scatter plot, adalah jenis diagram statistik yang menggunakan titik-titik untuk merepresentasikan nilai dari dua variabel yang berbeda. Setiap titik dalam diagram pencar mewakili satu pengamatan atau data dengan nilai-nilai yang sesuai untuk kedua variabel tersebut.

Scatter plot sangat berguna untuk menemukan pola atau hubungan antara dua variabel, serta untuk mengevaluasi distribusi data. Dalam scatter plot, sumbu horizontal umumnya digunakan untuk variabel independen (bebas), sedangkan sumbu vertikal digunakan untuk variabel dependen (bergantung). Dengan mengamati pola penyebaran titik-titik, kita mendapatkan wawasan tentang apakah ada korelasi antara dua variabel dan jenis korelasi apa yang mungkin ada (positif, negatif, atau tidak ada korelasi).

Berikut contoh menggambar diagram titik di EMT

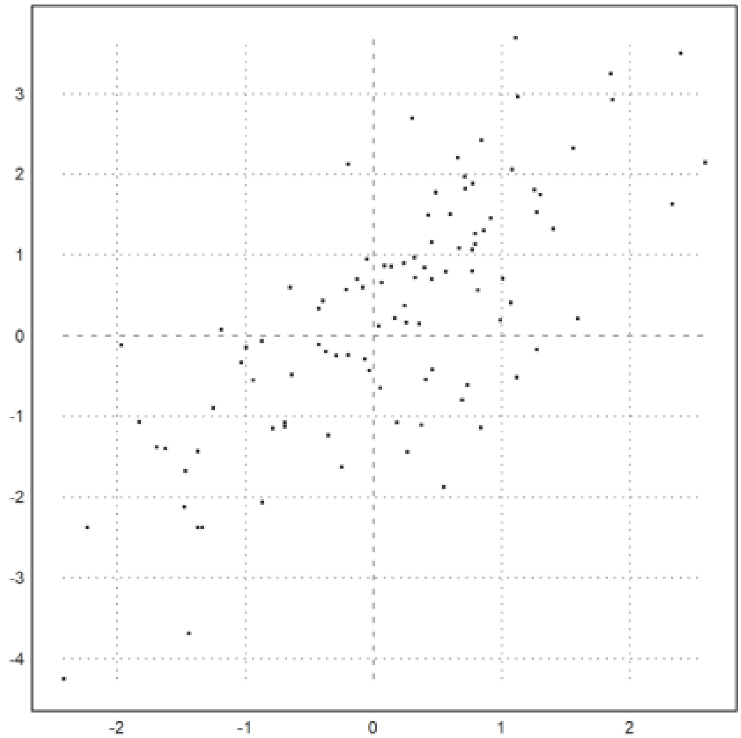
Contoh 1
Pada contoh ini, kita akan menggambar diagram titik dengan menggunakan fungsi plot2d().

```
>x=normal(1,100);
>plot2d(x,x+rotright(x),>points,style=".."):
```



Terdapat banyak style titik yang dapat kita dunakan, yaitu:
"[]", "<>", ".", "..", "...","*", "+", "|", "-", "o",
"[]", "<>", "o" (bentuk terisi)
"[]w", "<>w", "ow" (tidak transparan) Selanjutnya, akan kita coba gambarkan diagram tersebut menggunakan fungsi statplot(), pilih plottype="p" (karena kita akan menggambar plot titik).

```
>statplot(x,x+rotright(x),plottype="p",pstyle=".."):
```



Contoh 2
Pada contoh ini, akan kita gambarkan diagram titik menggunakan fungsi scatterplot().

```
>{MS,hd}:=readtable("table1.dat",tok2:=["m","f"]);
>writetable(MS,labc=hd,tok2:=["m","f"]);
```

Person	Sex	Age	Mother	Father	Siblings
1	m	29	58	61	1
2	f	26	53	54	2
3	m	24	49	55	1
4	f	25	56	63	3
5	f	25	49	53	0
6	f	23	55	55	2
7	m	23	48	54	2
8	m	27	56	58	1
9	m	25	57	59	1
10	m	24	50	54	1
11	f	26	61	65	1
12	m	24	50	52	1
13	m	29	54	56	1
14	m	28	48	51	2
15	f	23	52	52	1
16	m	24	45	57	1
17	f	24	59	63	0

18	f	23	52	55	1
19	m	24	54	61	2
20	f	23	54	55	1

```
>scatterplots(tablecol(MS,3:5),hd[3:5]):
```

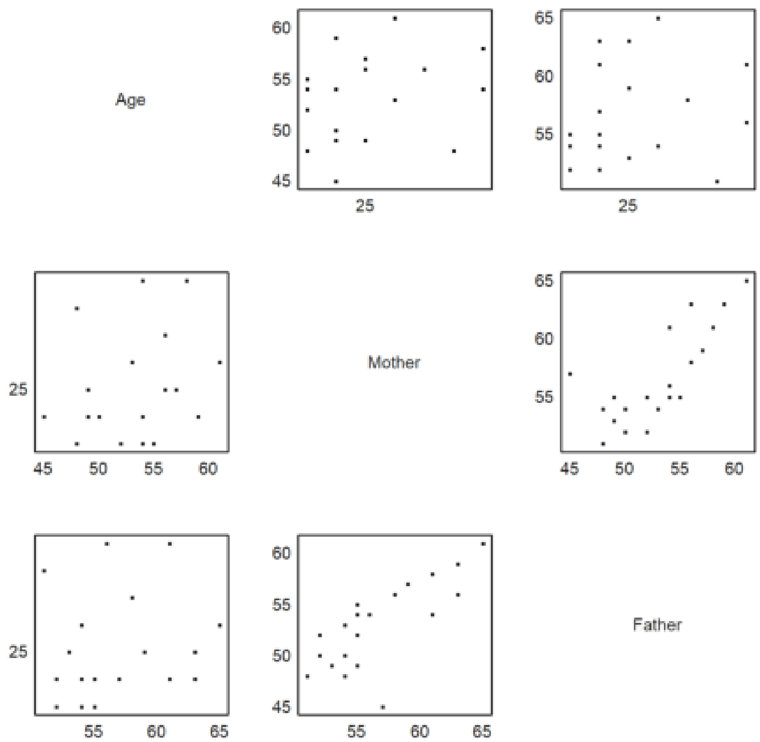


Diagram Garis

Diagram garis adalah penyajian data yang digunakan untuk menggambarkan suatu keadaan berupa data berkala atau berkelanjutan. Selain itu, diagram ini juga bisa dikatakan berhubungan dengan kurun waktu dan untuk menunjukkan perkembangan suatu keadaan. Diagram ini sangat tepat untuk menyajikan data untuk mengetahui kecenderungan kelakuan atau tren, seperti produksi minyak tiap tahun, jumlah kelahiran tiap tahun, jumlah produksi tiap jam, dan lain-lain.

Dalam diagram garis, terdapat sumbu vertikal (sumbu y) untuk menunjukkan frekuensi dan sumbu horizontal (sumbu x) untuk menunjukkan variabel tertentu.

Berikut contoh menggambar diagram garis di EMT

Contoh 1
Akan digambarkan diagram garis data banyaknya pelanggan di toko A tahun 2015-2023.

Kita deskripsikan terlebih dahulu matriks x dan y, kemudian akan kita buat tabel datanya.

```
>x=[2015,2016,2017,2018,2019,2020,2021,2022,2023]; y=[600,500,900,1000,800,
>writetable(x'|y',labc=["Tahun","Banyak Pelanggan"])
```

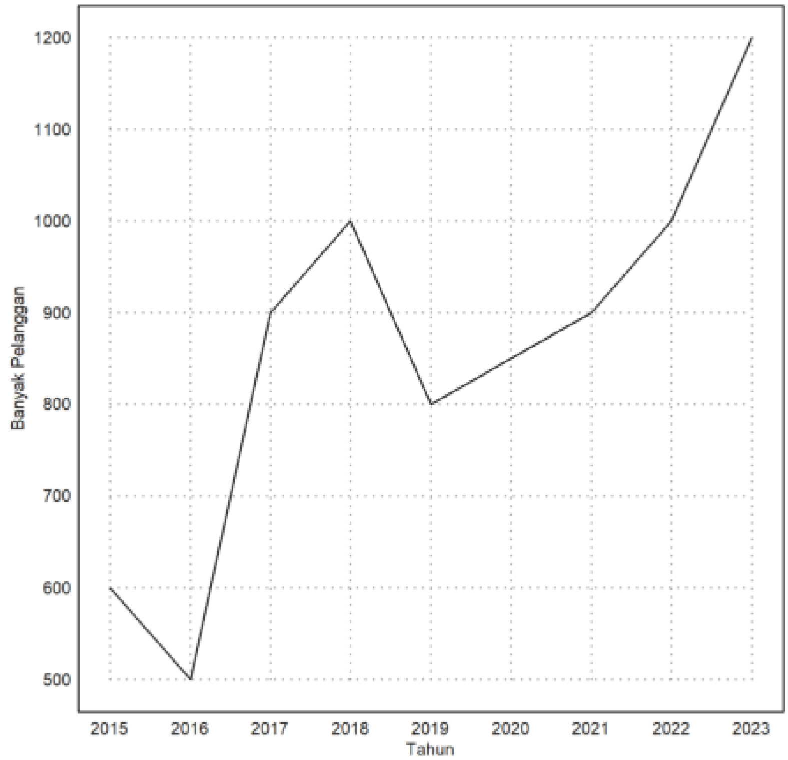
```
 Tahun Banyak Pelanggan
  2015              600
  2016              500
  2017              900
  2018             1000
  2019              800
  2020              850
  2021              900
  2022             1000
  2023             1200
```

Selanjutnya, akan digambarkan diagram garis dengan menggunakan fungsi statplot, dengan format: statplot (x, y, plottype="l", lstyle="-", xl="", yl="", color=none, vertical=0)

x : data untuk sumbu x
y : data untuk sumbu y
plotstyle : "l" (kita pilih style "l" karena berupa plot garis)
lstyle : style garis
xl : label sumbu x
yl : label sumbu y
color : warna garis
vertikal : vertikal

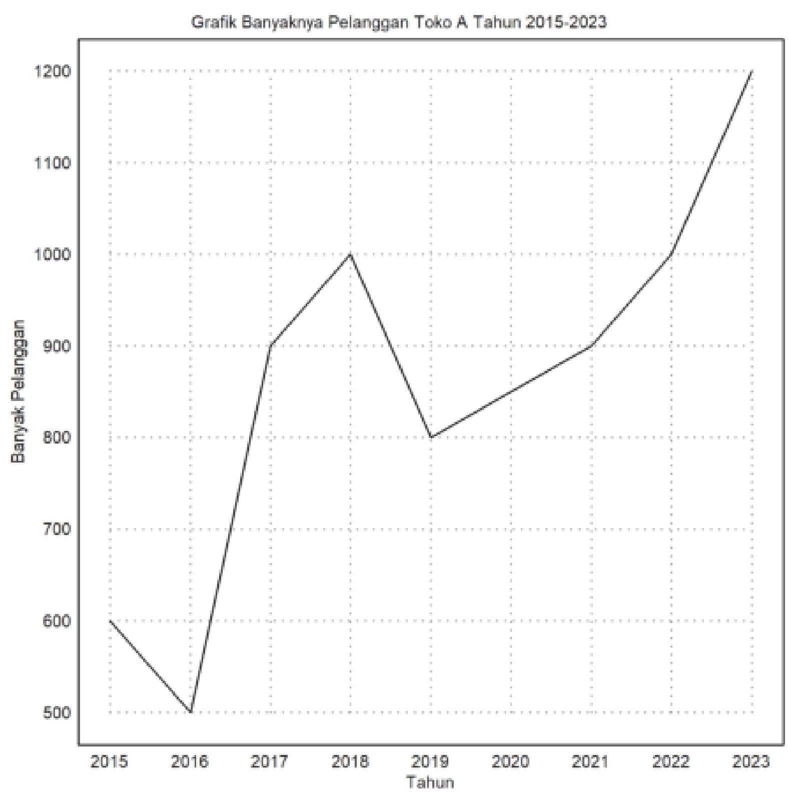
Style garis:
"-", "–", "-.", ".", ".-.", "-.-", "->"

```
>statplot(x,y,plottype="l",lstyle="-",xl="Tahun",yl="Banyak Pelanggan",vert
```



Kita juga bisa menambahkan judul pada grafik dengan menggunakan fungsi title().

```
>title("Grafik Banyaknya Pelanggan Toko A Tahun 2015-2023"):
```

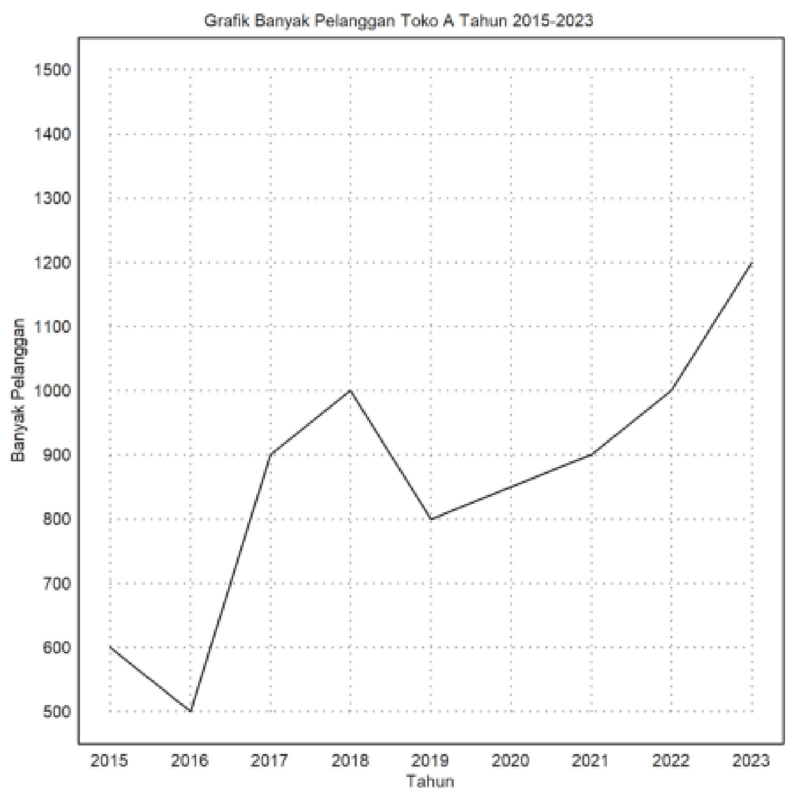


Menyajikan data dalam bentuk diagram dapat memudahkan pembaca untuk memahami data yang disajikan.

Dari diagram garis diatas, dapat kita peroleh informasi bahwa:
Banyaknya pelanggan di toko A tidak tetap (naik-turun)untuk setiap tahunnya.
Banyaknya pelanggan paling sedikit pada tahun 2016 yaitu sebanyak 500 pelanggan, sedangkan banyak pelanggan paling banyak pada tahun 2023 yaitu sebanyak 1200 pelanggan.

Selain menggunakan fungsi statplot, kita juga dapat menggambar diagram garis menggunakan fungsi plot2d() seperti yang sudah pernah kita pelajari sebelumnya, yaitu sebagai berikut.

```
>plot2d(x,y,a=2015,b=2023,c=500,d=1500,style="_",xl="Tahun",yl="Banyak Pela
```



a dan b : batas untuk sumbu x
c dan d : batas untuk sumbu y
style : gaya garis
xl : label untuk sumbu x
yl : label untuk sumbu y

Selain kita dapat menggambarkan diagram garis saja atau diagram titik saja, kita juga dapat menggambarkan diagram keduanya.

Contoh 2

Akan kita gambar diagram titik dan garis data hasil pemilu Jerman dari tahun 1990 sampai 2013, diukur dalam kursi.

```
>BW := [ ...
>1990,662,319,239,79,8,17; ...
>1994,672,294,252,47,49,30; ...
>1998,669,245,298,43,47,36; ...
>2002,603,248,251,47,55,2; ...
>2005,614,226,222,61,51,54; ...
>2009,622,239,146,93,68,76; ...
>2013,631,311,193,0,63,64];
>P:=["CDU/CSU","SPD","FDP","Gr","Li"];
>BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
>writetable(BT*100,wc=6,dc=0,>fixed,labc=P,labr=YT)
```

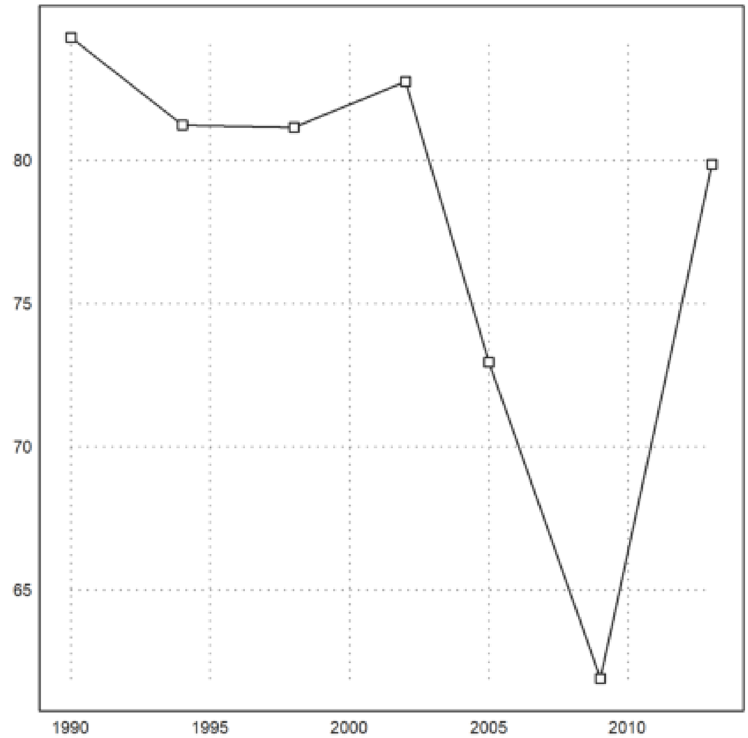
```
          CDU/CSU    SPD    FDP     Gr     Li
    1990       48     36     12      1      3
    1994       44     38      7      7      4
    1998       37     45      6      7      5
    2002       41     42      8      9      0
    2005       37     36     10      8      9
    2009       38     23     15     11     12
    2013       49     31      0     10     10
```

```
>BT1:=(BT.[1;1;0;0;0])'*100
```

```
[84.29,  81.25,  81.1659,  82.7529,  72.9642,  61.8971,  79.8732]
```

Akan kita gambarkan plot statistik sederhana, yaitu plot titik dan garis secara bersamaan dengan menggunakan fungsi statplot dan pilih plottype="b".

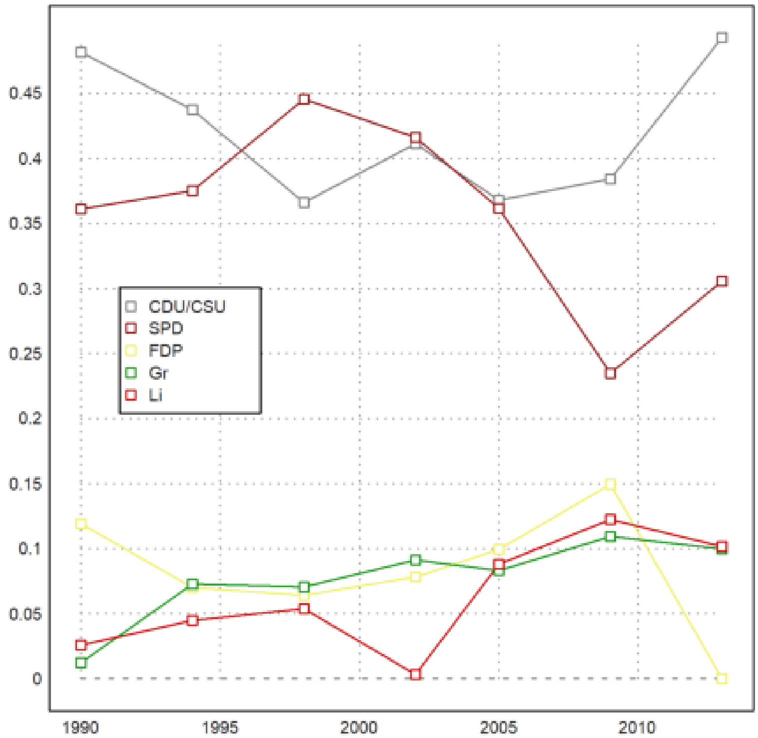
```
>statplot(YT,BT1,"b"):
```




```
>CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];
```

Untuk menggabungkan deretan data statistik dalam satu plot, dapat kita digunakan fungsi dataplot().

```
>J:=BW[,1]'; DP:=BW[,3:7]'; ...
>dataplot(YT,BT',color=CP);  ...
>labelbox(P,colors=CP,styles="[]",>points,w=0.2,x=0.3,y=0.4):
```

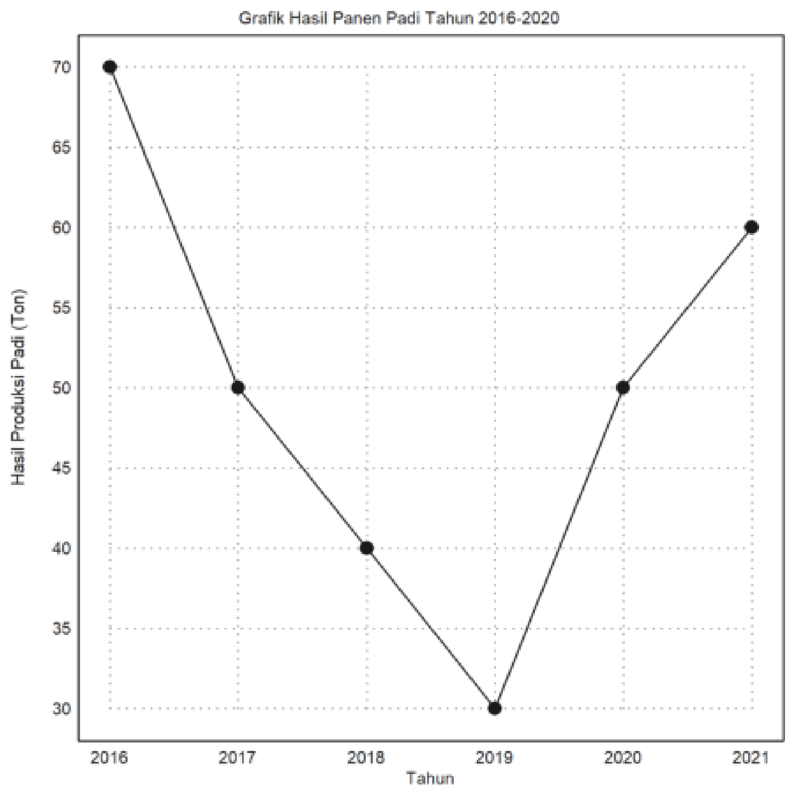


Contoh 3
Akan digambarkan diagram titik dan garis dari data hasil panen padi pada tahun 2016 sampai 2020.

```
>T=[2016,2017,2018,2019,2020,2021]; P=[70,50,40,30,50,60];
>writetable(T'|P',labc=["Tahun","Hasil Panen Padi (Ton)"])
```

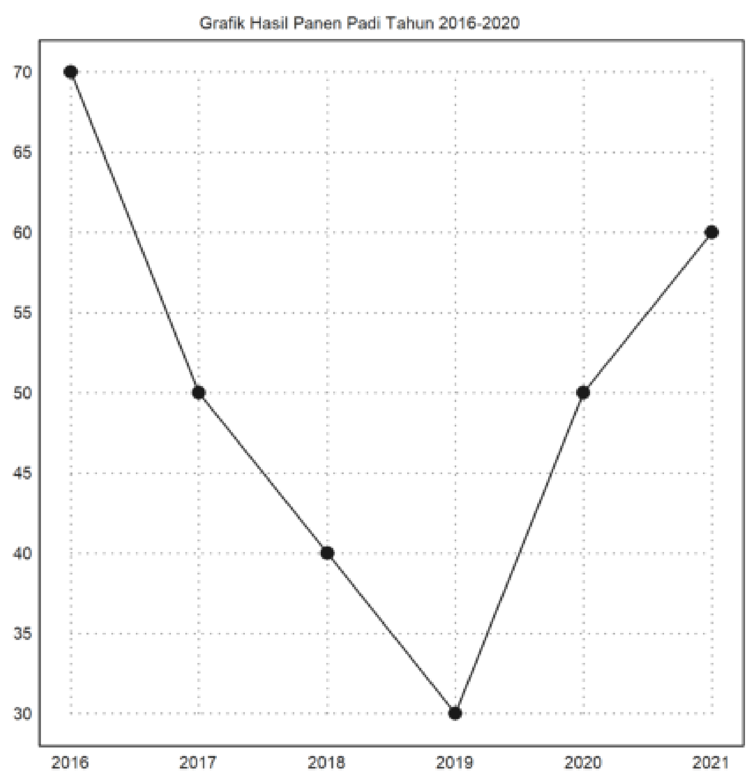
```
 Tahun Hasil Panen Padi (Ton)
  2016                     70
  2017                     50
  2018                     40
  2019                     30
  2020                     50
  2021                     60
```

```
>statplot(T,P,"b",pstyle="o#",lstyle="-",xl="Tahun",yl="Hasil Panen Padi (T
>title("Grafik Hasil Panen Padi Tahun 2016-2020"):
```



Kita juga menggambar diagram titik dan garis secara bersama dengan menggunakan fungsi plot2d(), yaitu sebagai berikut.

```
>plot2d(T,P,2016,2020); plot2d(T,P,>points,style="o#",>add); title("Grafik
```



Dari grafik di atas, dapat dengan mudah kita ketahui bahwa panen padi paling banyak yaitu pada tahun 2016 (70 ton) dan paling sedikit yaitu pada tahun 2019 (30 ton).

Kurva Regresi

Kurva regresi adalah representasi grafis dari model regresi yang digunakan untuk memodelkan hubungan antara satu atau lebih variabel independen (biasanya dilambangkan sebagai (X) dan variabel dependen (Y). Kurva regresi ini mencoba untuk menunjukkan pola atau tren dalam data dan memungkinkan kita untuk membuat prediksi atau estimasi berdasarkan model tersebut.

Secara umum, terdapat dua jenis kurva regresi yang umum digunakan yaitu regresi linier dan regresi non-linier. Regresi Linier adalah garis lurus yang digunakan untuk memodelkan hubungan antara variabel independen (X) dan variabel dependen (Y).

Persamaan regresi linier umumnya ditulis sebagai

$$Y = mX + b$$

di mana m adalah kemiringan (slope) dan b adalah perpotongan sumbu-y (intercept).

Regresi linier dapat dilakukan dengan fungsi polyfit () atau berbagai fungsi fit. Sebagai permulaan kita menemukan garis regresi untuk data univariat dengan polyfit(x, y, 1).

Berikut contoh menggambar kurva regresi di EMT

Contoh 1

```
>x=1:10; y=[2,3,1,5,6,3,7,8,9,8]; writetable(x'|y',labc=["x","y"])
```

```
          x         y
          1         2
          2         3
          3         1
          4         5
          5         6
          6         3
          7         7
          8         8
          9         9
         10         8
```

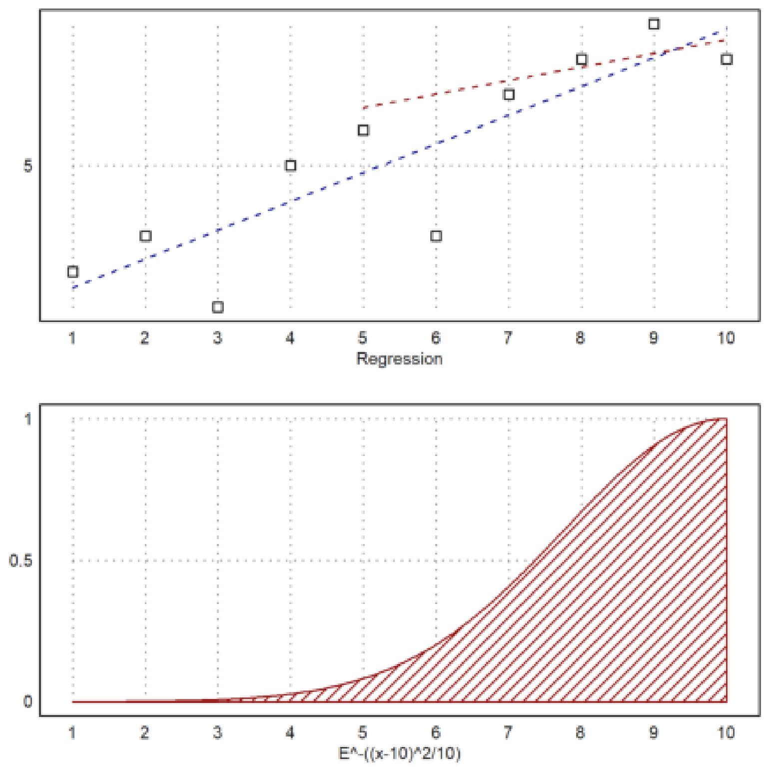
```
>p=polyfit(x,y,1)
```

```
[0.733333,  0.812121]
```

```
>w &= "exp(-(x-10)^2/10)"; pw=polyfit(x,y,1,w=w(x))
```

```
[4.71566,  0.38319]
```

```
>figure(2,1); ...
>figure(1); statplot(x,y,"p",xl="Regression"); ...
>  plot2d("evalpoly(x,p)",>add,color=blue,style="--"); ...
>  plot2d("evalpoly(x,pw)",5,10,>add,color=red,style="--"); ...
>figure(2); plot2d(w,1,10,>filled,style="/",fillcolor=red,xl=w); ...
>figure(0):
```

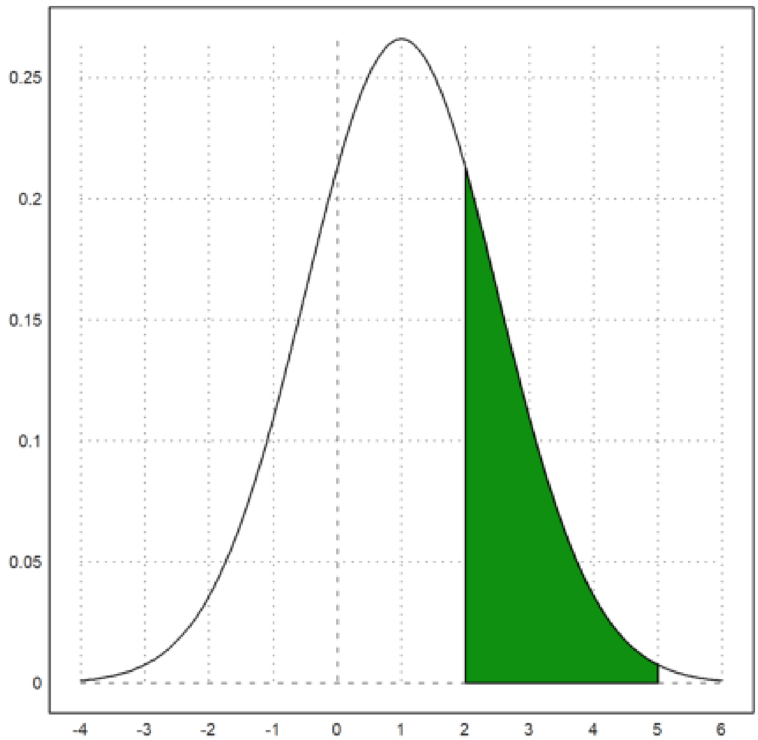


Kurva Fungsi Kerapatan Probabilitas

Fungsi kerapatan/kepadatan probabilitas adalah fungsi yang memberikan kemungkinan bahwa nilai suatu variabel acak akan berada di antara rentang nilai tertentu.

Grafik fungsi kepadatan probabilitas berbentuk kurva lonceng. Area yang terletak di antara dua nilai tertentu memberikan probabilitas hasil observasi yang ditentukan. Berikut contoh kurva fungsi kepadatan probabolitas.

```
>plot2d("qnormal(x,1,1.5)",-4,6);
>plot2d("qnormal(x,1,1.5)",a=2,b=5,>add,>filled):
```



Probabilitas variabel acak x yang terletak antara 2 dan 5 memenuhi
P(2<X<5)= luas daerah hijau

Kurva Distribusi Kumulatif

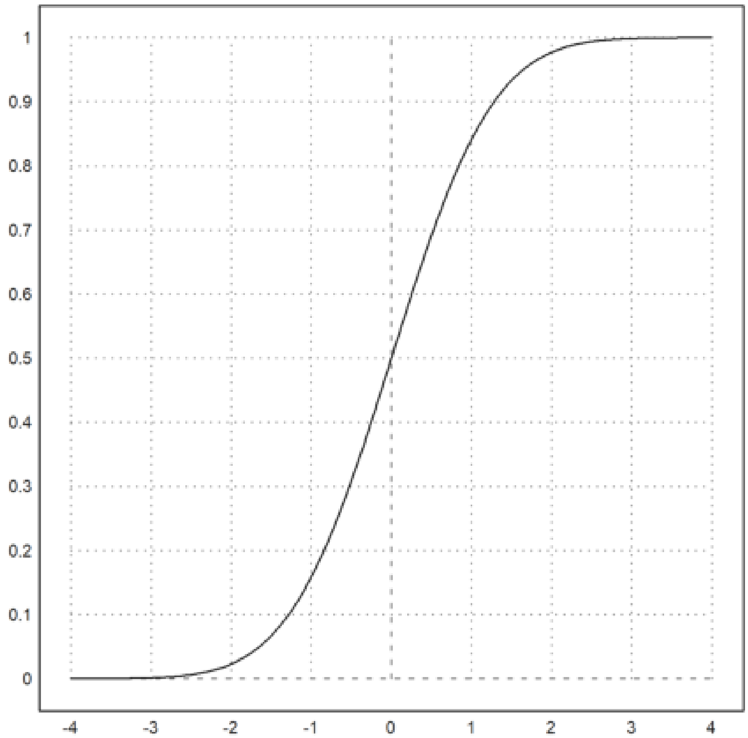
Kurva distribusi kumulatif (CDF) adalah representasi kumulatif dari fungsi distribusi probabilitas suatu variabel acak. CDF memberikan probabilitas bahwa variabel acak tersebut kurang dari atau sama dengan suatu nilai tertentu.

Seringkali, grafik CDF disajikan dalam bentuk kurva monotonik yang terus meningkat, dan ini memberikan gambaran visual yang baik tentang distribusi probabilitas variabel acak.

Berikut menggambar kurva distribusi kumulatif di EMT

Contoh 1

```
>plot2d("normaldis",-4,4):
```



Dapat kita lihat dalam kurva CDF kontinu di atas dibagi menjadi 3 bagian, yaitu:
1. Bernilai 0 untuk x kurang dari batas bawah daerah rentang.
2. Merupakan fungsi monoton naik pada daerah rentang.
3. Bernilai konstan 1 untuk x lebih dari batas atas daerah rentangnya.

Contoh 2
Diberikan variabel acak dengan PDF sebagai berikut:

$$f(x) = \begin{cases} \frac{6}{5}(x^2 + x) & 0 \leq x \leq 1 \\ 0 & \text{x yang lain.} \end{cases}$$

Untuk menggambar grafik CDF-nya, pertama kita cari terlebih dahulu CDF dari fungsi tersebut.

Untuk x pada interval

$$(-\infty, x)$$

$$F(x) = \int_{-\infty}^{x} f(x)\, dx$$

$$F(x) = \int_{-\infty}^{x} 0\, dx$$

$$F(x) = 0$$

Untuk x pada interval [0,1]

$$F(x) = \int_{-\infty}^{x} f(x)\,dx$$

$$F(x) = \int_{-\infty}^{0} f(x)\,dx + \int_{0}^{x} f(x)\,dx$$

$$F(x) = \int_{-\infty}^{0} 0\,dx + \int_{0}^{x} \frac{6}{5}(x^2 + x)\,dx$$

$$F(x) = 0 + \frac{6}{5}(\frac{x^3}{3} + \frac{x^2}{2})$$

$$F(x) = \frac{6}{5}(\frac{x^3}{3} + \frac{x^2}{2})$$

Untuk x pada interval

$$(1, \infty)$$

$$F(x) = \int_{-\infty}^{x} f(x)\,dx$$

$$F(x) = \int_{-\infty}^{0} f(x)\,dx + \int_{0}^{1} f(x)\,dx + \int_{1}^{x} f(x)\,dx$$

$$F(x) = \int_{-\infty}^{0} 0\,dx + \int_{0}^{1} \frac{6}{5}(x^2 + x)\,dx + \int_{1}^{\infty} 0\,dx$$

$$F(x) = 0 + \frac{6}{5}(\frac{1^3}{3} + \frac{1^2}{2}) + 0$$

$$F(x) = \frac{6}{5}\frac{5}{6} = 1$$

Sehingga, diperoleh CDF :

$$F(x) = \begin{cases} 0 & x < 0 \\ \frac{6}{5}(\frac{x^3}{3} + \frac{x^2}{2}) & 0 \leq x \leq 1 \\ 1 & x > 1. \end{cases}$$

Selanjutnya, akan kita gambarkan grafik CDF tersebut di EMT.
Pertama, kita definisikan terlebih dahulu fungsi f(x) dengan menggunakan fungsi function map.

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

```
    if x<0 then return 0
    else if x>=0 and x<=1 then return (6/5)*((x^3/3)+(x^2/2))
    else return 1
    endif;
  endfunction
```

Kemudian, kita akan menggambar grafik fungsi di atas dengan menggunakan fungsi plot2d() pada interval x dari -1 sampai 2.

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